Bilim ve Gelecek Kitaplığı

50 SORUDA

yapay zekâ

Cem Say



Kitap Dizisi 17

7. BASKI

50 SORUDA yapay zekâ | Cem Say | 17

Bilim ve Gelecek Kitaplığı - 59

50 Soruda Kitap Dizisi - 17
50 Soruda Yapay Zekâ

Cem Say

© Bu kitabın yayın hakları
7 Renk Basım Yayım ve Filmcilik Ltd. Şti.'ne aittir.

Birinci Baskı: Ekim 2018
Yedinci Baskı: Aralık 2018

ISBN: 978-605-5888-58-9

Yayıma hazırlayan: Nalân Mahsereci
Sayfa tasarımı: Baha Okar
Kapak figürü çizimi: Aslı Say

Baskı: Berdan Matbaası
Davutpaşa Cad. Güven Sanayi Sitesi
C Blok No: 215-216
Topkapı / İstanbul Sertifika No: 12491
Tel: 0212.613 12 11

7 Renk Basım Yayın ve Filmcilik Ltd. Şti
Moda C. Zuhal Sk. No: 9/1, Kadıköy-İstanbul
Tel: 0216.349 71 72
http://www.bilimvegelecek.com.tr
e-mail: bilgi@bilimvegelecek.com.tr



yapay zekâ

Cem Say

İÇİNDEKİLER

Önsöz 7

1. Bölüm

YAPAY ZEKÂNIN TOHUMLARI 11

1) Bilgisayarlar her şeyi yapabilir mi? İnsanların yapabilip makinelerin yapamayacağı şeyler var mıdır? **11**
2) Düşünen bir makine yapılabilir mi? **15**
3) Sadece 0 ve 1 her şeye yeter mi? **18**
4) Matematikçiler çelişkiyi sever mi? **21**
5) Matematik sağlama bağlanabilir mi? **23**
6) Gödel neden öldü? **25**
7) Alan Turing kimdir? **28**
8) Turing makinesi nedir? **30**
9) Evrensellik nedir? **34**
10) Çözülemez problemler var mıdır? **36**

2. Bölüm

BEYİNLER VE DİĞER BİLGİSAYARLAR 41

11) Bilgisayar nedir? **41**
12) Doğanın programlama dili nedir? **45**
13) Yaşamın programlama dili nedir? **47**
14) Beyin nasıl bir bilgisayardır? **51**
15) İnsanların programlama dili nedir? **54**
16) Enformasyon nedir? **57**
17) Çarpma toplamadan zor mudur? **61**
18) Hesaplama karmaşıklığı nedir? **65**
19) "P = NP?" sorusu nedir, ne anlama gelir? **70**
20) Zar atmak işe yarar mı? **74**
21) Kuantum bilgisayarı nedir, ne işe yarar? **77**

3. Bölüm

YAPAY ZEKÂNIN DOĞUŞU 81

22) Yapay zekâ nedir? **81**
23) Turing testi nedir? **83**
24) "Yapay zekâ" adını kim koydu? **85**
25) "Yapay zekâ mevsimleri" nedir? **86**

4. Bölüm

YAPAY ZEKÂ NELER YAPAR, NASIL ÇALIŞIR? 93

26) "Eski moda yapay zekâ" nedir, nasıl çalışır? **93**
27) Evrimsel programlama nedir? **97**
28) Sinir ağları nasıl çalışır? **98**
29) Derin öğrenme nedir? **103**
30) Bilgisayar buluş yapabilir mi? **106**
31) Bilgisayar sanat yapabilir mi? **109**
32) Bilgisayarlar avukatlık yapabilir mi? **112**
33) Kasparov'u nasıl yendik? **115**
34) AlphaGo dünya go şampiyonunu nasıl yendi? **117**
35) Kendi kendini süren otomobiller nasıl çalışır? **120**
36) Sohbet programları nasıl çalışır? **123**
37) Bilgisayarlar insan dillerini nasıl anlar? **127**
38) Bilgisayarlar nasıl çeviri yapar? **133**
39) Google nasıl gezegen keşfetti? **137**
40) Bilgisayarlar bizi bizden iyi tanıyabilir mi? **139**
41) Robotlar askere alınsın mı? **143**
42) Yapay zekâ doktorluk yapar mı? **145**

5. Bölüm

YAPAY ZEKÂNIN GELECEĞİ 149

43) Yapay zekâ yanlış yapar mı? **149**
44) Yapay zekâ kullanımının zararları nelerdir? **152**
45) Robotlar âşık olmalı mı? **156**
46) Robotlar âşık olabilir mi? **159**
47) Çince Odası nedir? **163**
48) Gödel'in eksiklik teoremi yapay zekâyı olanaksız kılar mı? **165**
49) Yapay zekâ dünyayı ele geçirip hepimizi yok edecek mi? **166**
50) İnsan zekâsının bir geleceği var mı? **171**

OKUMA ÖNERİLERİ 176

DİZİN 177

Önsöz

Yapay zekâ çok ilginç bir kavram. Makinelerin "düşünce gerektiren" işleri yapabilmesi insanlarda karışık duygular yaratıyor. Düşünme yeteneğinin insanlara mahsus bir özellik olduğuna inananlarımız bunu önce tuhaf, sonra da bir ölçüde rahatsız edici buluyorlar. Tartışma insanların nasıl olup da düşünebildiğine, bu "düşünce" denen şeyin tam olarak ne olduğuna, başka konularda çok başarılı olan bilimin bu soruyu da yanıtlamasının mümkün olup olmadığına uzanıyor.

Biz tartışaduralım, makinelerin yükselişi sürüyor. Kol gücünde bizi geride bırakmalarına alışmış, hatta bunu sevinilecek bir gelişme olarak kabul etmiştik, ama şimdi sırada beyin gücü var gibi görünüyor. Neredeyse her gün bilgisayarların performansının bir "beyaz yaka" işinde daha insanlarınkini geçtiğine dair haberler karşımıza çıkıyor. Bu işin sonu nereye varır?

Yapay zekâ sistemleri nasıl çalışıyor? Makineler nasıl düşünebiliyor? Her yaptıklarını onlara biz insanlar öğretiyorsak nasıl oluyor da bazen hiçbir insanın bilmediği şeyleri keşfedebiliyorlar? Yapamayacakları bir şey var mı?

Elinizdeki kitapta bu saydıklarım gibi 50 temel soru çerçevesinde yapay zekâ fikrinin tarihçesini, bilimsel altyapısını, bu konuda yapılan çalışmaları, ortaya çıkan ürünlerin nasıl çalıştığını, neleri yapabildiklerini, neleri

henüz yapamadıklarını ve ileride neler olabileceğine ilişkin öngörülerimi okumanın zevkli olacağını umduğum bir dille anlatıyorum.

İlginç şekilde, meslek hayatımın tamamında bu kitabın konusuyla ilgili çalıştım diyebilirim: Doktora tezime başladığımda yıl 1989'du. Uluslararası yapay zekâ literatüründe Türkiye kaynaklı ilk bilimsel yayın bu çalışmadan çıktı. Bilgisayarları insan gibi düşündürtmeye çabalarken karşılaştığım "Peki ama, insan beyni denen bilgisayar nasıl düşünebiliyor?" ve "Acaba hiçbir bilgisayarın düşünemeyeceği şeyler var mıdır?" soruları sonraki yıllardaki akademik yönelimimi belirledi: Boğaziçi Üniversitesi Bilişsel Bilim Lisansüstü Programı'nın kuruluşunda yer aldım ve yapay zekânın babası Alan Turing'in insanlığa diğer armağanı olan hesaplama kuramına merak saldım. Turing'in matematiği baş döndürücü derecede güzel olmakla kalmaz, akıllı telefonlarımızdan kuantum bilgisayarlarına ve aynı zamanda beyinlerimize dek her tür bilgi işlem sisteminin uyması gereken doğa yasalarını belirler. Kitabımızın çerçevesi de bu düzene göre çizildi.

Kitap beş ana bölümden oluşuyor: "Düşünen makine" fikrinin hayal edilmesiyle başlayıp elektronik bilgisayarın ortaya çıkmasıyla biten yaklaşık 250 yıllık bir matematik serüveniyle başlıyoruz. Sonra bilgisayarın tam olarak ne olduğunu, doğada hangi süreçlerin hesaplama işi olarak görülebileceğini ve bu "hesaplama" dediğimiz olguya ilişkin doğa yasalarının keşfini işlediğimiz ikinci bölüm var. Yapay zekâ sistemlerinin nasıl çalıştığını içime sinen bir şekilde anlatabilmem için bu matematiksel altyapıyı kurmam şarttı, umarım bunu "sözelci" okurları kaçırmadan (ve hiçbir aşamada liseden yukarı bir bilgi düzeyi varsaymadan) yapabilmişimdir. Üçüncü bölümde Turing'in bıraktığı bayrağı devralıp yapay zekâ projesini başlatan öncüleri, hataları ve sevaplarıyla tanıyoruz. Dördüncü bölümde yapay zekânın birçok değişik alandaki uygulamalarını görmekle kalmıyoruz, bu sistemlerin nasıl çalıştıklarını ve hangi yöntemleri kullandıklarını yarım asırdır

bilinen arama algoritmalarından evrimsel programlamaya ve son yılların gözdesi derin öğrenme tekniğine varıncaya dek "kaportayı kaldırarak" inceliyoruz. Son bölüm de, yapay zekâya mevcut eksikliklerini, kimi filozofların itirazlarını, "bilinç" problemini ve insanlığa olumlu veya olumsuz muhtemel etkilerini de kapsayan daha geniş bir pencereden bakıyor.

Bu kitabı seveceğinizi umuyorum. Ben yazmayı çok sevdim. Bilimi çok severim. Küçüklüğümden beri, ama yaşım büyüdükçe daha da artan bir oranda, bilimle ilgili yeni bir şeyler öğrenince büyük haz duyarım. Bu, paylaşılınca artan hazlardan. Bir süredir çeşitli konferanslar ve *Herkese Bilim Teknoloji* dergisinde kısa yazılar aracılığıyla bilimsel konuları kitlelere anlatıyor ve bundan büyük zevk alıyorum, ama bu kitap boyutunda bir projeye ilk kez kalkıştım. Umarım okurken benim yazarken tattığım mutluluğu tadarsınız.

Ben bu satırları yazarken ülkemiz büyük bir ekonomik kriz yaşıyor. Eğer bu kitabı elinizde tutuyorsanız yayınevi bir şekilde onu basacak kadar kâğıt almayı başardı demektir. Yıllardır aşina olduğum bilimsel yayınların yanı sıra, bu proje için yurtdışında yayımlanmış yapay zekâ konulu popüler bilim kitaplarını da inceleme imkânım oldu. Övünmeyi sevmem, ama rahatlıkla söyleyebilirim ki, ne genelde bu konuda çalışma yapan bizim biliminsanlarımızın (yapay zekâ konusunda çok güçlü olan kendi yuvamı, Boğaziçi Üniversitesi Bilgisayar Mühendisliği Bölümü'nü özellikle anayım), ne de bu kitabın yurtdışındakilerden hiçbir eksiği yok; tam tersine, biz bu şartlarda böyle işler çıkarabildiğimiz için fazladan bir yıldızı hak ediyoruz. Türkçemize helal olsun.

Teşekkürler

Kitaba yorum ve düzeltmeleriyle katkıda bulunan Ahmet Çevik, Ethem Alpaydın, Güven Güzeldere, Levent Akın ve Özlem Özdemir'e, güzel çizimi için Aslı Say'a, şekillerde yardımıma koşan Baha Okar'a, Turing'in annesinin yazdığı kitabı bana hediye eden Taylan Cemgil'e,

doğal zekâsı için Soner Canko'ya ve bu projeyi aklıma düşüren Nalân Mahsereci'ye teşekkür ederim. Hataların tümü bana aittir.

Cem Say
İstanbul, 30 Eylül 2018

1. Bölüm
YAPAY ZEKÂNIN TOHUMLARI

1 | Bilgisayarlar her şeyi yapabilir mi? İnsanların yapabilip makinelerin yapamayacağı şeyler var mıdır?

Her ne kadar doğduklarından beri onlarla iç içe olan gençleri buna inandırmak zor olsa da, bilgisayar denen makine aslında çok yeni bir icattır. Annemle babamın çocukluğunda bilgisayar diye bir şey yoktu. Benim çocukluğumda hiç kimsenin evinde bilgisayar yoktu. Ben ÖSYM'nin sınavıyla Boğaziçi Üniversitesi Bilgisayar Mühendisliği Bölümü'ne giren ilk öğrenci kuşağındanım. Doktora tez araştırmama başladığım yılda, şimdi kısaca "www" diye adlandırılan "dünyayı saran ağ" daha yeni icat edilmekteydi. Kaynaklara erişebilmek, hele de rakibimiz konumundaki diğer araştırmacıların aynı konuda ne yapmakta olduklarından haberdar olabilmek çok zordu ve bir kitap ya da makaleye (o da, bir şekilde varlığını öğrenebildiklerime) erişebilmek için uzun süre postacı yolu gözlemek gerekebiliyordu. Düşünün, dünyanın bir yerinde bir bilgi ortaya konulmuş, ama siz ona erişemiyorsunuz,

hatta varlığından bile haberiniz olmuyor. Gençler o karanlık çağı anlamakta zorlanmakta haklı.

Sonrası malum. Bir devrim yaşandı, hâlâ da yaşanıyor. Bilgisayarlar küçüldü de cebimize girdi! Ama bir yandan da büyüyüp insanlığın tümünü kapsama alanına aldılar. Çoğu insan bilgisayarlar hakkında çok bir şey bilmiyor ama bazı bilgisayarlar birçok insan hakkında çok fazla şey biliyor. Kimi insanların hayatlarını sürdürebilmeleri doğrudan bilgisayarlara bağlı. Bilişim devriminde dönüşü olmayan noktayı çoktan geçtik; artık istesek de bilgisayarsız bir dünyaya dönemeyiz. Zaten istediğimiz de yok!

İnsanlar, derinlemesine anlamadıkları teknolojilerin gücünü yanlış değerlendirebilir. Bu hesap iki yönde de hatalı olabilir; kimileri yeni bir buluşa hak ettiğinden fazla kudret atfedebilirken kimileriyse bu yeniliğin tam potansiyelini kavrayamayıp küçümseyebilir. Çağımızın makinesi olan bilgisayar ve bu kitapta konu edindiğimiz, belki de insanlık tarihinin en önemli mühendislik projesi olan "yapay zekâ" (YZ) girişimi de bu yanlış değerlendirmelerden payını almakta. Birkaç örnek vereceğim.

17 Mart 1997 Cuma sabahı İstanbul Teknik Üniversitesi Taşkışla Kampüsü'nde "Bilgi İşleyen Makine Olarak Beyin" başlıklı toplantılar dizisinin ikincisi düzenlenmişti. Farklı disiplinlerden hocalar bir araya gelmiş, aklın ne olduğunu, beynin nasıl çalıştığını ve elbette bu "düşünme" denen işi kafataslarımızın içindekilerden başka makinelerin de yapabilip yapamayacağını konuşuyorduk. Bu toplantı serisi halka açıktır ve her seferinde kalabalık bir izleyici kitlesi katılır. Konuşmalardan sonraki açık tartışma saatlerinden birinde yeri geldi ve çok yakında dünya satranç şampiyonu Gari Kasparov'un bir bilgisayar tarafından yenileceğini iddia ettim (Yapay zekâ projesinin başlangıcından beri en önemsenen hedeflerinden olan bu aşamaya o tarihte henüz erişilememişti. IBM şirketinin bu amaç için geliştirdiği Deep Blue adlı bilgisayar, Kasparov'la önceki yıl yaptığı ilk maçı kaybetmişti).

Hemen seyirciler arasından itirazlar yükseldi. Bir beyefendi, kısaca "Siz bilgisayarları biliyor olabilirsiniz,

ama Kasparov'u tanımamışsınız!" diye özetlenebilecek bir argümanla Kasparov'un satrancı çok ama çok farklı bir şekilde oynadığını, içgörü ve önsezilerin işe dahil olduğunu, bir makine tarafından yenilmesinin mümkün olamayacağını ifade etti. Bense bunun (ilerideki sayfalarda okuyacağınız gibi) bir hesaplama probleminden ibaret olduğu ve bilgisayarın yeterince hızlanmasının sadece biraz zaman alacağı kanısındaydım. O sırada yeni doçent olmuştum, "Ben profesör olmadan Kasparov'u yeneriz!" dediğimi anımsıyorum (Evet, bu insan-makine çekişmelerinde ben makinelerin tarafını tutuyorum).

Aradan iki ay bile geçmeden, 11 Mayıs 1997'de mutlu haberi CNN'den aldım. Deep Blue, New York'ta düzenlenen 6 oyunluk rövanş maçını son oyunda Kasparov'u perişan ederek kazanmıştı. "Keşke yüklüce bir paraya iddiaya girseydim!" diye düşündüm.

Ama aynı hataya, üstelik profesör olduktan yıllar sonra, bu kez kendimin düşeceğimi bilemezdim tabii. Satrancın Uzakdoğu'daki rakibi olan go oyunu, uzun yıllar boyunca bilgisayarlaştırılmaya direndi. Oyunun yapısal farkları (çok daha büyük bir tahtada oynanması ve taşların birbirlerinden farklı rollerinin olmaması) satranç için kullandığımız teknikle çalışan go programlarının şampiyonlar şöyle dursun, sıradan insan oyuncular karşısında bile bir başarı sağlayamamasına yol açıyordu. Birçok uzman, Sloven bilimadamı Ivan Bratko'nun dünya çapında okutulan (ve ayıptır söylemesi, benim çalışmalarımdan da söz eden) yapay zekâ ders kitabının 2012 baskısında dediği gibi, daha goyu çözmemize vakit olduğunu söylüyordu. Sonra 2016'nın Ocak ayında bir sabah kalkınca gördük ki, Google hepimizi ters köşeye yatırmış.

Google DeepMind şirketinden mühendisler, *Nature* dergisinde yayımlanan makalelerinde son yıllardaki yapay zekâ patlamasının ardındaki "derin öğrenme" tekniğini kullanarak go oynamayı "kendi kendine" öğrenen AlphaGo adında bir program geliştirdiklerini açıklıyorlardı. AlphaGo'yla Avrupa go şampiyonu arasında gizlice bir maç düzenlemişler ve 5-0 kazanmışlardı. Sonraki bir yıl

içinde AlphaGo oyunun en iyileri olarak kabul edilen Li Sedol ve Kı Cie'yi de ezip geçerek insanüstü seviyede olduğunu kanıtladı. Projenin çalışanlarının haricinde, *Nature* makalesini okuyan her uzmanın "Vay be!" dediğine eminim.

Bu örnekler çoğaltılabilir. Tecrübeme göre her zaman, "Yapay zekâ ŞU işi (insanlar kadar iyi) yapamaz!" diyen birileri oluyor. Genellikle o iş çok da uzun olmayan bir süre sonra bilgisayarlar tarafından insanlardan daha yüksek performansla yapılmaya başlanıyor. Bu ilk başta bir çalkantı yaratıyor, ama çok geçmeden bir ses duyuluyor: "O iş 'insan özü' gerektirmeyen, mekanikleştirilebilen bir işmiş zaten! Esas ŞU diğer işte bilgisayarlar insanları geçemez!" Ve döngü böylece sürüyor, mühendisler sürekli kayan hedefi tutturmak için o diğer işe odaklanıyorlar.

Ama yukarıda dediğim gibi, bir de öbür uçtakiler var. Ağustos 2009'da ComputerWeekly.com sitesinde teknoloji yazarı Richard Fisher'ın "Akıllı telefonlar: Yapamayacakları herhangi bir şey var mı?" başlıklı yazısını gülümseyerek okumuştum. Apple şirketinin "uygulama mağazası"yla tanışmasını mizahi bir dille anlatan Fisher, taksi ve restoran bulma uygulamaları gibi aradan geçen yıllarda hayatımızın bir parçası haline gelenlerden "diş fırçalama koçu" ve yetişkin oyuncakları gibi daha sıra dışı olanlarına uzanan geniş bir yelpazede cebimizdeki bilgisayarların potansiyelinin sınırsız olduğu izlenimini verdiğini söylüyordu. Dünyayı saran ağın mucidi Tim Berners-Lee bile 2013 Ocak ayında Dünya Ekonomik Forumu'nda gençlere programlama eğitimi verilmesi gerektiğini savunurken defalarca "Kodlama bilen insanlar hayal edebildikleri her şeyi bilgisayara yaptırtma yeteneğine sahiptir" diyordu.

İlerideki sayfalarda göreceğimiz gibi, bu da yanlış bir iddia. Ama çağımızda (Berners-Lee gibi bir profesyonel değilse de) sıradan bir insan, bilgisayarların her şeyi yapabilecekleri fikrine kapılmakta haklı sayılabilir: Bilgisayarlar her şeyimizi biliyorlar. Okulda notlarımızı, iş yerinde maaşlarımızı, sigorta yaptırmaya niyetlenirsek de kalan

ömrümüzü hesaplayabiliyorlar. Kimin seçimde ne oy vermeyi düşündüğünü önceden bilip, hesabına çalıştıkları siyasilere raporlayabiliyorlar. Sizi fotoğraflarda tanıyan programı mı istersiniz, kullanıcısının yıllardır görmediği arkadaşlarını (hatta bir vakada hiç tanımadığı biyolojik babasını) ondan habersiz bulup tekrar temas kurmasını önereni mi? Yazdıklarınızdan o sırada alkollü olup olmadığınızı anlayan program da var, yol tarif eden de, otomobilinizi o tarife göre süren de. Bilgisayarınız veya robotunuzla karşılıklı konuşabiliyorsunuz.

Peki ama başlıktaki soruların yanıtı nedir? Bilgisayarlar her şeyi yapabilir mi? İnsanların yapabilip makinelerin yapamayacağı şeyler var mıdır? Ve elbette, nice filozofu emekli eden o büyük bilmece: Makineler düşünebilir mi?

Binlerce yıllık emeklemeden sonra 20. yüzyılda bilim bu soruları yanıtlamayı başardı. Kitabımıza, zor bir bilmeceyi çözmekle kalmayıp belki de dünyanın yeni hâkimlerinin tohumunu atan bu harika serüvenin öyküsüyle başlayacağız. Bunun için bir 350 yıl geri gitmemiz gerekecek (Yapay zekâyı önce bilgisayarların, problem çözmenin ve düşünmenin matematiksel altyapısını açıklamadan nasıl anlatabileceğimi bilmiyorum). Önden buyurun!

2 | Düşünen bir makine yapılabilir mi?

Gottfried Wilhelm Leibniz, bir koltuğa bir karpuz tarlası sığdırmayı başaran o özel insanlardandı. Hukuk eğitimi alan Leibniz hayatını birkaç asilzadenin emrinde danışmanlık, diplomatlık, tarihçilik ve kütüphanecilik yaparak kazansa da "boş zamanları"nı felsefe, matematik ve fizik çalışmalarıyla değerlendirerek bilim ve düşünce dünyasında neredeyse damga vurmadık alan bırakmamıştı. Bizim hikâyemizdeki rolü ise, tarihteki ilk bilgisayar mühendislerinden olmasıyla ilgili.

Fransız bilgin Blaise Pascal toplama ve çıkarma işlemlerini yapabilen ilk mekanik hesap makinesini icat ederken henüz bir bebek olan Leibniz, 25 yaşına geldiğinde dört aritmetik işlemin dördünü de yapabilecek bir makine geliştirmek için kolları sıvadı. "En basit kişinin bile makine kullanarak kesinlikle yapabileceği hesaplar için mükemmel insanların saatlerce köleler gibi uğraşmasına değmez!" diyordu. 1673'te diplomatik bir görev için gittiği İngiltere'de makinesini Londra'daki Kraliyet Akademisi'ne sundu. Bu başarısı Akademi üyeliğine kabul edilmesini sağladı.

Günümüz insanına sadece dört işlemi yapmakta kullanılabilen, üstelik çalıştırmak için bir kolu döndürmeniz gereken bir alet etkileyici gelmeyebilir, ama 17. yüzyıl için bu bir yapay zekâ başarısıydı (Avrupalı soyluların para hesaplarını yaptırmak için aritmetik bilen okumuş gençleri istihdam ettikleri çağlardan söz ediyoruz). Leibniz buluşunun uzun vadedeki sonuçlarını düşündü: Bu müthiş bilişsel iş makinelere yaptırılabildiğine göre neden diğerleri de yaptırılamasın? Nihayetinde insanların akıl yürütürken yaptıkları da bir tür hesap değil miydi?

> Muhakemelerimizi düzeltmenin tek yolu, onları matematikçilerinkiler kadar elle tutulur hale getirmektir, öyle ki hatamızı bir bakışta bulabilelim ve kişiler arasında anlaşmazlıklar olduğunda hemencecik 'Hesaplayalım, kimin haklı olduğunu görelim' diyebilelim.[1]

Leibniz, düşünme işlerimizi bizim için yapabilecek bir makine hayal ediyordu! Bu sisteme "**calculus ratiocinator**" adını vermişti. Bunun için ilk adım olarak, nasıl kendi hesap makinesi sayıları temsil edebiliyorsa, düşüncelerimizde geçen türlü kavramların matematiksel olarak temsil edilebileceği bir tür sembolik dile ihtiyaç olduğunu görmüştü.

---

1) Gottfried Wilhelm Leibniz, "De arte characteristica ad perficiendas scientias ratione nitentes", C. I. Gerhardt (Ed.), *Die philosophischen Schriften von Gottfried Wilhelm Leibniz*, Cilt 7, 1890, Berlin: Weidmannsche Buchhandlung.

Attığı bu tohumun yeşerme hikâyesini sonraki soruda sürdüreceğiz.

Belki de en büyük buluşu olan türev ve integral hesabı fikrini Isaac Newton'dan aşırdığının iddia edilmesi, Leibniz için bir kırılma noktası oldu. Bugün bilim tarihçileri Newton'la Leibniz'in bu matematik şaheserini birbirlerinden bağımsız geliştirdiklerini kabul ediyorlar (Liseden kimilerimizin aşk, kimilerininse nefretle hatırladığı $\int$ ve dx sembollerini Leibniz yaratmıştı). Öyle görülüyor ki Newton bu konudaki çalışmalarını yıllarca yayımlamadan tutmuş, paralel olarak aynı sahada çalışan Leibniz ise daha sonra başlayıp daha önce yayın yapmıştı. İki matematikçinin de eserlerini yayımlamalarının üzerinden onlarca yıl geçtikten sonra, son derece huysuz ve tuhaf bir karakter olan Newton, muhtemelen Leibniz'in "Önce ben buldum" söyleminden rahatsız olarak, Leibniz'e karşı bir suçlama kampanyası başlattı. 64 yaşındaki Leibniz kendisini savunmaya çalıştıysa da, iş bir İngiltere-Almanya çekişmesine döndü. Leibniz'in "hakemlik edin" diye başvurduğu Kraliyet Akademisi, Alman Leibniz'den olayı kendi açısından anlatmasını isteme gereği duymadan İngiliz Newton'un haklı olduğunu ilan ediverdi. Leibniz 1716'da 70 yaşında öldüğünde sözüm ona hamisi olan İngiltere Kralı I. George cenazesine katılmadı. Ne Londra ne de Berlin Akademileri, ikisinin de yaşam boyu üyesi olan Leibniz'in ardından bir anma yapma gereği duydu.

Leibniz felsefi anlamda bir iyimserdi. "Tanrı her şeye kadir ve iyi ise, o zaman dünyada neden kötülük ve ıstırap var?" sorusuna yanıt olarak içinde bulunduğumuz dünyanın mümkün olan tüm dünyalar arasında en iyisi olması gerektiğini savunuyordu. Fransız filozof Voltaire 1759'da yazdığı ve asırlar boyunca okunacak olan matrak *Candide* romanında, başına art arda gelen felaketlerden sonra habire "Yine de olabilecek dünyaların en iyisindeyiz" diye avunan çok bilmiş öğretmen Pangloss karakteriyle Leibniz'i alaya aldı. O sırada ölümünün üzerinden 40 yıldan fazla zaman geçmiş olmasına rağmen adamcağızın mezarına bir taş bile dikilmemişti.

3 | Sadece 0 ve 1 her şeye yeter mi?

10 el parmağımız olduğu için sayıları 10 değişik rakam kullanan ondalık gösterimle yazıyoruz. Orta Amerika'daki Maya Uygarlığı 20 rakama dayalı bir sayı gösterimi kullanıyordu, çünkü pabuç giyme alışkanlıkları yoktu! Görüldüğü gibi, gösterimin tabanı olarak hangi sayıyı kullanacağınız size kalmış bir şey. Bahtsız dostumuz Leibniz sadece 0 ve 1 rakamlarına dayanan ikili gösterimin hayranlarındandı. Ama 0 ve 1'in müthiş gücünü tam olarak anlayışımızı George Boole'a borçluyuz.

George Boole yoksul bir kunduracının üstün zekâlı oğluydu. Ekonomik zorluklar nedeniyle ilkokuldan sonra ne öğrendiyse kendi kendine öğrendi. Doğum yeri olan Lincoln kentinin gazetesinde Latinceden tercüme ettiği bir şiir yayımlandığında, "Bu çocuk bu yaşında bu çeviriyi yapmış olamaz, aşırmıştır!" suçlamasına maruz kalmıştı. Boole 16 yaşında matematik öğretmeni olarak çalışıp evin geçimini sağlamaya başladı, 19'unda da kendi okulunu kurdu.

Boole'un öykümüzdeki yeri, yüz yıldan uzun süre önce Leibniz'in gerekliliğini gördüğü "düşünce dilinin matematiksel gösterimi"nin bulunmasına olan katkısından geçiyor. O zamana dek mantık, eski Yunan'dan miras kaldığı şekliyle, doğal dille yapılıyordu, örneğin:

Gri kediler tırmalamaz.
Prenses gri bir kedidir.
Demek ki Prenses tırmalamaz.

Oysa ki matematiği doğal dil gibi çifte anlamlılık, muğlaklık vs. sorunları olmayan, çok daha sıkı kurallara bağlı, neredeyse mekanik görünümlü bir dille yapıyoruz, sözgelimi

x+4=7
Demek ki x=3

Bu işleri bir makineye yaptırmak isteyen birisi için ikinci dilin birinciden daha uygun olduğunu görüyor musunuz? Leibniz'in rüyasının gerçekleşmesi için atılması gereken ilk adım böyle bir biçimsel düşünce dilinin ortaya konulmasıydı. Boole mantıkla cebiri evlendirerek bunu başardı.

Boole işe kelimelerin anlamlarını düşünerek başladı. Besbelli ki yukarıdaki akıl yürütmede geçen "gri kedi" öbeğinin anlamını, onu oluşturan iki kelimenin anlamlarını bir tür işleme tabi tutarak (diyelim ki "çarparak") hesaplıyorduk. Peki ama "gri" ve "kedi" sözcükleri neyi anlatıyordu?

"Gri" deyince aklınıza ne geliyor? Gri olan her şey gelebilir, gri olmayan hiçbir şeyin kastedilmediği de çok açık. Bu durumda "gri", "gri olan bütün şeyler topluluğu"nu anlatıyor. Aynı şekilde "kedi" de "bütün kediler topluluğu"na karşılık geliyor. İfadelerimizde cebir geleneğine uyarak kelimelerin ilk harflerini kullanalım, iki anlam arasındaki işlemi de çarpma için kullanılan "." sembolüyle gösterelim. Yani "gri kedi", "g.k" diye gösterilecek.

Bu işlem ne yapıyor? "Gri kedi", yukarıda anlaştığımız şekilde "bütün gri kediler topluluğu" anlamına geliyorsa, "." işleminin, sonuç olarak üzerinde çalıştığı iki topluluğun da üyesi olan varlıkların oluşturduğu topluluğu (yani günümüzde alışık olduğumuz terimlerle, aldığı iki kümenin kesişimi olan kümeyi) verdiği ortada. Boole bu işlemin doğasını araştırırken akıllıca bir soru sordu: Aynı şeyi kendi kendisiyle bu işleme tabi tutarsak ne olur? "Gri gri" veya "kedi kedi" ne demektir mesela? Bu durumlarda anlamın aynı kaldığını gören Boole, şu sonuca vardı: Herhangi bir x kavramı için, x.x=x'tir.

Ve dahice olan yere geldik. "Gerçek sayılar arasındaki çarpmada x.x=x eşitliğini sağlayan x değerleri hangileridir?" diye sordu Boole. Cevap kolay: Sadece 0 ve 1 sayıları karelerine eşittir. Demek ki mantığı cebir dilinde yazarken sadece 0 ve 1'i kullanacağız.

Peki 0 ve 1 tam olarak ne anlama geliyor? Yine cebire danışalım. Biliyoruz ki, her x sayısı için 0.x=0 ve 1.x=x'tir.

İşlemimizi kümeler arası kesişme işlemi olarak somutladığımıza göre 0, hangi kümeyle kesişirse kesişsin değişmeyen küme olmalı. Yani boş küme. Aynı mantıkla 1 de dikkate aldığımız her şeyi içeren "evrensel küme" olmalı.

Boole bir cümlenin (teknik terimle, önermenin) doğru olduğu anlar kümesi boşsa yanlış, değilse de doğru olduğundan hareketle mantıktaki "doğru" ve "yanlış" kavramlarını 1 ve 0'la gösterebileceğimizi ortaya koydu. Mantığın matematikselleştirilmesinin öyküsüne sonraki soruda devam edeceğiz, ama sadece sayıların değil, aslına bakarsanız her şeyin 0 ve 1'lerle gösterilebilmesinin önemini biraz düşünelim. Başta şaşırtıcı gelebilir, ama sadece 0 ve 1 "harf"lerinden oluşan bir alfabe ile çok daha kalabalık olan Türk veya Çin alfabelerindeki sembollerle ifade edebileceğiniz her şeyi ifade edebilirsiniz. Duymuşsunuzdur: Bilgisayarlarımızda yer alan elektronik devreler (0 olarak yorumlanabilen "voltaj düşük" durumu ile 1 olarak yorumlanabilen "voltaj yüksek" durumu gibi) iki farklı durumdan birinde bulunabilen basit sistemlerin birbirlerine bağlanmasıyla inşa edilir. Bilgisayarınızda tuttuğunuz her aşk şiiri, her güzel resim, her müzik parçası bir dizi 0 ve 1 halinde saklanıp sonra dilediğinizde yine alıştığınız biçeme dönüştürülebildiğine göre, bu minik alfabenin gösterim gücünden yana bir eksikliği yoktur (İşin sırrı örneğin Türk alfabesinde tek sembolle, sözgelimi A harfiyle ifade ettiğimiz bir bilgiyi ikili sistemde biraz daha çok harfle, mesela 01000001 dizisiyle temsil etmekten ibaret).

"Yüksek" sınıftan olmadığı için 19. yüzyıl kafasındaki vatanı İngiltere'de üniversite hocalığına alınmayan Boole, 34 yaşında İrlanda'da yeni kurulan Cork Üniversitesi'nin ilk matematik profesörü oldu. Kasım 1864'te evinden beş kilometre uzaktaki üniversiteye yürüyüşü sırasında şiddetli yağmurda sırılsıklam olan giysileriyle ders verirken üşütüp zatürreeye yakalandı. "Çivi çiviyi söker" diye özetlenebilecek bir kocakarı tıbbı inanışındaki karısı Mary, yataktaki adamcağızın üzerine kova kova su dökmekten ibaret bir "tedavi" uyguladı. Durumu iyice kötüleşen Boole, bir ay geçmeden hayata gözlerini yumdu. 49 yaşındaydı.

4 | Matematikçiler çelişkiyi sever mi?

Gottlob Frege, 1848'de Almanya'da doğdu (Leibniz'in ektiği tohumun büyümesini izlerken sırayla bir Alman, bir İngiliz adları görmeye devam edeceğiz). Kariyerini mantıkla matematiğin birleştirilmesi ülküsüne adayan Frege, Leibniz'in ısmarladığı ve Boole'un başladığı "düşünce dili" projesini tamamlayan kişi olarak görülebilir.

Teknik terimle, Boole'un "önermeler mantığı" için yaptığı matematikselleştirmenin eksiklerini gideren "yüklemler mantığı" Frege'nin eseridir. "Her erkek bir kadını sever" derken "Öyle bir kadın vardır ki her erkek onu sever" mi, yoksa "Her erkeğin bir sevdiği kadını bulunur" mu demek istediğimizi muğlaklığa yer vermeden ifade edebilmemize yarayan ∀ ve ∃ niceleyicilerini lisede görmüş müydünüz? Boole notasyonunda "Ahmet bir erkektir" gibi basit bir önermeyle yukarıdakiler gibi nicelemeler içerenleri bile birbirlerinden yapısal olarak ayırt etmek mümkün değildir, oysa Leibniz'in düşlediği gibi mantıksal akıl yürütme yapacak bir sistemin bu ayrımı yapıp soruları ona göre yanıtlamasının gerekeceği açık. Bu anlamda yüklemler mantığı, bir düşünce dili olarak görülmeye çok daha layıktır.

Mantıkla matematiğin evliliği sayesinde ayrı gelişmiş olan bu iki alanın kavram ve araçlarının birbirlerine uygulanabilmesi harika sonuçlara yol açtı. "Kanıt" kavramı modern anlamını bu sırada kazandı: Doğruluğunu tartışmadığımız, kanıtlanması gerekmeyecek kadar açık birkaç temel önermeyi (bunlara "belit" denir) kabul etmekte anlaşırız (Örneğin geometride "İki farklı noktadan tek bir doğru geçer" belitini kullanabiliriz). Bir veya birkaç doğru önerme alıp yeni bir doğru önerme üretmeye yarayan birkaç "çıkarım kuralı"mız vardır. Mesela şu çıkarım kuralını pek severim:

"*A ise B'dir*" ve "*A*" önermelerinin ikisi birden doğruysa, o zaman "B" önermesi de doğrudur!

Ve başka hiçbir şeye de ihtiyacımız yoktur! Yeni bir önermenin doğru olduğunu kanıtlamak istiyorsanız, yapmanız gereken onun başka doğru önermelerden bir çıkarım kuralıyla elde edilebileceğini göstermektir. Peki o diğer doğru önermelerin doğru olduğunu nereden mi bileceğiz? Onları da aynı şekilde kanıtlamamız gerek! Yani bir teoremin kanıtı, belitlerden başlayıp art arda çıkarım kuralları uygulanarak üretilen doğru önermelerden geçilerek teoremin cümlesine varılan bir çıkarımlar zinciridir. Oyunun kurallarını (yani belitlerle çıkarım kurallarını) bilen herkes, önüne gelen bir metnin geçerli bir kanıt olup olmadığını, onu baştan aşağı okuyup sırayla her adımdaki önermenin bu anlamda doğru olup olmadığını mekanik şekilde denetleyerek söyleyebilir.

Matematikçilerin kâbusu çelişkidir. Bir önerme ya doğrudur, ya da yanlış. İkisi birden olamaz. Sisteminiz hem "A doğrudur" hem de onunla apaçık çelişen "A yanlıştır" cümlelerinin kanıtlanmasına el veriyorsa, tüm adımlarının doğru önermelerden oluşması gereken kanıt zincirlerinin içine bir yanlışlık mikrobu bulaşmış olacağından, her yanlış önerme de kanıtlanabilir hale gelecek, yani matematiğin doğruyu yanlıştan ayırma iddiası suya düşecektir. Çelişkiden vebadan korkar gibi korkmalıyız; umalım ki belitlerimiz böyle bir çelişkiye meydan vermiyor olsunlar.

Frege, kendisinden sonraki felsefecileri de pençesine alışını ileride göreceğimiz bir tutkuya kapılmıştı: Matematiğin ya da en azından aritmetiğin, tümüyle bir mantık sistemi olarak ifade edilmesi. En basitine varıncaya dek tüm matematiksel kavramları mantığa dayamayı başarınca, matematiğin gerçeklikle bağlanacağı, matematikçilerin doğru dedikleri şeylerin gerçekten doğru olduğuna kuşkuları kalmayacağı umuluyordu. Bu niyetle yazdığı *Aritmetiğin Temel Yasaları* adlı kitabının ilk cildini 1893'te bastıran Frege, dokuz yıllık bir çalışmadan sonra ikinci cildi yazmayı bitirmiş ve matbaaya göndermek üzereydi ki, postacı kapıyı çaldı.

Gelen mektup genç İngiliz felsefeci Bertrand Russell'dandı. Russell, Frege'yi matematiğin mantık temeline

oturtulması ülküsünde yürekten desteklediğini yazarak başladığı mektubunu bir bomba atarak, *Aritmetiğin Temel Yasaları*'nın ilk cildinde kurulan sistemde onulmaz bir hata bulduğunu bildirerek bitiriyordu. Russell, "Kendini içermeyen tüm kümelerin kümesi kendi kendini içerir mi?" diye soruyordu. Temiz havada biraz düşününce görüleceği gibi, bu sorunun yanıtı evetse hayır, hayırsa da evet olmak zorundadır. Frege'nin sistemindeki belitler bu çelişkiye yol açıyordu, bu da, yukarıda söz ettiğimiz gibi, en istenmeyen şeydi (Her tür saçmalığın ifade edilebildiği doğal dilde bu çelişkili cümlenin söylenebiliyor olması sorun değildir, ama kesinlik ve mutlak doğruluk aradığımız matematikte bu tam bir felakettir).

Frege durumu hemen kavradı ve ikinci cildin sonuna Russell'ın mektubunu anarak yıllardır kurmaya çalıştığı yapının sakat olduğunu anladığını itiraf ettiği bir ek yazdı. Ama bilim tarihine geçen bu örnek dürüstlüğü, kişiliği hakkında hatırlanan tek şey olmadı maalesef.

Frege 1925'te 76 yaşında öldü. Seneler sonra, bilime yaptığı büyük katkı artık herkesçe anlaşıldığında, yazdığı her şeyi didik didik eden araştırıcılar 1924'te tuttuğu günlüklerini okuduklarında neye uğradıklarını şaşırdılar. Bir zamanlar kendisini bir liberal olarak tanımlayan Frege, hayatının son yılında herkese oy hakkı verilmesine, parlamenter sisteme, demokratlara, Katoliklere ve Fransızlara karşı, Yahudilerin sürgün edilmesinden yana bir Hitler hayranı olduğunu yazmıştı. Almanya'yı ilk dünya savaşından sonra hasta eden faşizm virüsünden o da nasibini almış, nefret dolu bir yaşlı adam olarak hayata veda etmişti.

5 | Matematik sağlama bağlanabilir mi?

Paris kenti 1900 yılında yeni asrı "Evrensel Sergi" adı verilen bir dünya fuarıyla karşıladı. Altı ay boyunca açık

kalan fuarda çağın mimari ve teknik yenilikleri sunuluyordu.

Ağustos ayında Paris'e gelenlere fuarın milyonlarca ziyaretçisinin yanı sıra Uluslararası Matematikçiler Kongresi'nin katılımcıları da eklendi.

Zamanın en büyük matematikçilerinden olan David Hilbert, kongrede 20. yüzyıldaki meslektaşlarına meydan okuma niteliğinde tarihi bir konuşma yaptı ve kendi değerlendirmesine göre matematiğin o ana dek çözülememiş en önemli 10 problemini listeledi. Bu listedeki ikinci problem şuydu: "Aritmetiğin çelişkisiz olduğunu kanıtlayınız!"

Matematikçilerin çelişkisizlik kaygısından geçen soruda söz etmiştik. Biçimsel sisteminiz (yani kullanmaya karar verdiğiniz semboller, belitler ve çıkarım kuralları bütünü) eğer sembollere yüklenen anlamlara göre birbiriyle çelişkili iki önermenin birden kanıtlanmasına izin veriyorsa beş para etmez. Hilbert aritmetikte kullanılan belitler için böyle bir çelişkinin olanaksız olduğunun matematiksel bir kanıtını istiyordu. Matematiğin mantığın konusu olmaya başlaması, eskiden düşünülemeyecek böyle soruları akla getiriyordu.

Önceki soruda tanıştığımız Bertrand Russell, matematiğin mantık temeline oturtulması bayrağını Frege'den devraldıktan sonra bu konuda insanüstü bir çaba gösterdi (Bu öyküyü konu edinen *Logicomix* adlı çizgi romanı okumanızı hararetle tavsiye ederim). Russell'ın meslektaşı Alfred North Whitehead'le birlikte bu amaçla yazmaya giriştiği *Principia Mathematica*, tarihin en zor kitaplarından biridir. Her kavramı, en ama en temel parçacıklarına ayırıp, hiçbir kuşkuya, Frege'nin düş kırıklığına yol açandaki gibi hiçbir paradoksa meydan vermeyecek şekilde yazma çabası, Whitehead'le Russell'ın yıllarına, neredeyse akıl sağlıklarına ve Russell'ın evliliklerinden birine mal oldu. Çok çetrefilli bir notasyonla yazılan kitabın 1912'de basılan ikinci cildinin 86. sayfasında nihayet 1+1'in 2'ye eşit olduğunun kanıtı tamamlanıyor ve hemen ardından "Yukarıdaki önerme bazen

yararlı olur" cümlesi yer alıyordu! Üçüncü ciltten sonra halleri kalmayan ikili, geometrinin temellerini işlemesini öngördükleri dördüncü cildi yazmaktan vazgeçtiler (Eski bir mantıkçı bilmecesi: "*Principia Mathematica*'yı kaç kişi okumuştur? Cevap: İki. Russell'ın yazdığı yerleri Russell, Whitehead'inkileri de Whitehead").

Matematik artık sağlama bağlanmış mıydı? Hilbert'in ısmarladığı çelişkisizlik kanıtı yolda mıydı? Bu hikâye bir noktada yapay zekâya varacak mıydı? Okumaya devam!

6 | Gödel neden öldü?

Kurt Gödel, 28 Nisan 1906'da Avusturya-Macaristan İmparatorluğu yurttaşı olarak doğdu. 12 yaşındayken o ülke yıkılıp bölününce kendisini Çekoslovak vatandaşı olarak buldu, fakat Alman kökenli olduğundan kendisini oraya ait hissetmiyordu. Bu nedenle 23 yaşında eski imparatorluğun diğer bir mirasçısı olan Avusturya'nın vatandaşlığına geçti. Dokuz yıl sonra Hitler Almanyası Avusturya'yı yutunca bu kez otomatik olarak Alman yurttaşlığı edinmiş oldu. Son olarak da 42 yaşında İkinci Dünya Savaşı'ndan sonra göç ettiği ABD'nin vatandaşlığına geçti. 5 Aralık 1947'de, bu son vatandaşlığa geçiş işlemi sırasında yaşanan şu hikâye pek ünlüdür:

Vatandaşlık başvurusunun tamamlanıp andın içilebilmesi için Gödel'in yetkili yargıçla son bir mülakat yapması gerekiyormuş. Bu işleme tanıklık etmek için Gödel ikisi de büyük bilimadamları olan iki arkadaşını, dahi fizikçi Albert Einstein'la oyun kuramının kurucularından ekonomist Oskar Morgenstern'i yanında götürmüş. Yolda Gödel arkadaşlarına yeni ülkesi ABD'nin anayasasını titizlikle incelediğini ve metinde ülkeyi faşizme geçirmek için kullanılabilecek mantıksal bir açık bulduğunu söylemiş! Mülakatta sorun çıkmasından kaygılanan Einstein'la

Morgenstern, Gödel'e "Aman ne olur, şu vatandaşlığı alana dek bu konuyu açma!" demişler.

Mülakat iyi başlamış. Fakat Morgenstern'in anılarına göre yargıç sora sora o en olmadık soruyu sormuş!

> Yargıç: "Nerelisiniz Bay Gödel?"
> Gödel: "Avusturya."
> Yargıç: "Avusturya'nın yönetim şekli nedir?"
> Gödel: "Cumhuriyet idi, ama anayasa sonuçta bir diktatörlüğe dönüştürülmesine el verdi."
> Yargıç: "Ah, ne kötü! Bizim ülkemizde bu olamaz işte."
> Gödel: "Hayır, olabilir. Bunu ispatlayabilirim."(2)

Bu noktada yargıç Gödel'i dinlemek yerine, "Bu konuya hiç girmeyelim!" deyip adamcağızı susturmuş. Einstein'ın hatırı da araya girince, vatandaşlık işi pürüzsüz hallolmuş.

Gödel yıllar sonra, derdini başka kimseye anlatmadan, ölüp gitti. Ama bu sohbetin duyulmasından sonra anayasa hukuku ve mantık uzmanları "Acaba Gödel'in bulduğu gizli kapı neydi?" diye araştırmaya başladılar. Hâlâ bu konuda makaleler yayımlanır. Son okuduğumda anayasanın değiştirilme usullerini düzenleyen 5. Madde'den şüpheleniyorlardı.

Bu hikâye Gödel'in bizim anlatımızdaki rolü hakkında da iyi bir ipucu veriyor. Şimdiye dek birbirinden zeki adamların kurduğu büyük ve görkemli bir yapıya şahit olduk. O sıradaki belitleriyle matematik, doğruya ulaşmanın garantili bir yolu olarak görülüyordu. Daha önce anlattığım gibi, alanın devi Hilbert, birbiriyle çelişkili iki cümlenin ispatlanamayacağına emindi ve bu "çelişkisizlik" özelliğinin kanıtını istiyordu. Hilbert aynı zamanda matematiğin "eksiksizlik" diyebileceğimiz bir diğer harika özelliğe de sahip olduğuna, yani "doğru" olan her şeyin bu sistem içinde bir kanıtının bulunabileceğine inanıyordu (Hilbert emekli olduğu 1930 yılında yaptığı veda konuşmasını, bilimsel iyimserliğin zirvesi sayılan, daha

2) Oskar Morgenstern'in 13 Eylül 1971 tarihli yazısından alıntı.

sonra mezar taşına da yazılacak şu sözlerle bitirmişti: "Bilmeliyiz, bileceğiz").

Sonra Kurt Gödel çıktı ve Hilbert'in hayallerini yıktı. O muhteşem yapının kimsenin fark etmediği temel bir açığını yakaladı.

Daha 25 yaşında olan Gödel'in "*Principia Mathematica* ve akraba sistemlerin biçimsel olarak kararlaştırılamaz önermeleri üzerine I" başlıklı makalesi, Hilbert'in matematiğin hem çelişkisiz hem de eksiksiz olduğu inancının yanlışlığını ispatlıyordu. "Birinci eksiklik teoremi" mealen şunu der:

"Kimi temel aritmetik işlemleri kapsayan (örn. *Principia Mathematica* gibi) bir biçimsel sistem çelişkisizse eksik olmak, yani kimi doğru önermelerin kanıtını içermemek zorundadır."

Gödel bu müthiş gerçeği teoremde sözü edilen cinsten bir önermeyi bizzat inşa ederek kanıtladı. İlk bakışta sayılar hakkında karmaşık bir cümle gibi görünen önerme, dikkatle incelendiğinde, "Bu cümle bu sistemde ispatlanamaz" demekte olduğu fark ediliyordu! Düşünün: Bu cümle ya doğrudur, ya yanlış. Eğer yanlışsa, sistemimiz bu yanlış cümlenin kanıtlanmasına el vermektedir, yani çelişkilidir. Ama bir dakika! Sistemimizin çelişkisiz olduğunu varsayıyorduk. O zaman geriye kalan tek ihtimal, cümlenin doğru söylediği, yani sistemimizde doğru bir cümlenin ispatlanamadığıdır.

Peki madem gerçeğin tümüne matematiksel ispat yoluyla ulaşamayacağız, bari Hilbert'in 1900'de Paris'te koyduğu hedefe, yani sistemimizin çelişkisiz olduğunun bir kanıtına ulaşarak gönül rahatlığıyla "En azından ispatlayabildiğimiz her şey kesinkes doğru!" diyebilir miyiz? Gödel bu teselliyi de elimizden aldı. Aynı makalede kanıtladığı "İkinci eksiklik teoremi" der ki:

"Kimi temel aritmetik işlemleri kapsayan (örn. *Principia Mathematica* gibi) bir biçimsel sistem eğer çelişkisizse kendi çelişkisizliğinin bir kanıtını içeremez."

Neden mi? Diyelim ki sistemimiz çelişkisiz. İlk eksiklik teoreminin kanıtı sırasında inşa ettiğimiz "Bu cümle

bu sistemde ispatlanamaz" cümlesine kısaca C diyelim. İlk teoremden biliyoruz ki C cümlesi doğru. Şimdi varsayalım ki sistemimizin çelişkisizliğinin bir kanıtı var. Ama o zaman bu kanıtı ilk teoremin kanıtıyla birleştirerek C cümlesinin doğru olduğunu kanıtlayabiliriz! Ne kanıtladık? "Ben kanıtlanamam" diyen C cümlesini. Demek ki C yanlışmış! Aynı cümlenin hem doğru, hem yanlış olduğunu kanıtladık, demek ki sistemimiz çelişkili! E hani çelişkisizdi? Bu çıkmaz sokağa başta çelişkisizliğin kanıtlanabilir olduğunu varsayarak girdik, demek ki tek çıkar yol bu varsayımın yanlış olması, yani sistemimizin çelişkisizliğinin hiç kanıtlanamaması.

Bu beyin uçuran ispata ileride döneceğiz. Şimdi gelelim başlıktaki sorunun yanıtına.

Gödel 1938'de ailesinin itirazlarına aldırmadan kendisinden altı yaş büyük olan dansöz Adele Nimbursky ile evlendi. Yukarıda anlatılan vatandaşlığa alınma işleminden sonra 30 yıldan uzun süre Gödel'in İleri Çalışmalar Enstitüsü'nde görevli olduğu Princeton'da yaşadılar. Kurt'un Nazi iktidarı sırasında başlayan psikolojik sorunları ilerleyen yaşla birlikte çoğaldı. Zehirlenmekten korkuyor, eşinin yaptıklarından başka yemek yemiyordu. Ne yazık ki Adele 1977'de altı aylığına hastaneye yatmak zorunda kaldı. Kurt 14 Ocak 1978'de açlıktan öldüğünde 29 kg ağırlığındaydı. Adele 1981'de hayatını kaybetti. Princeton Mezarlığı'nda yan yana yatıyorlar.

7 | Alan Turing kimdir?

Alan Turing bilgisayar biliminin babası ve yapay zekânın kurucusu olarak bilinir. Fakat bu çok eksik bir tanımdır. Hiç kuşkusuz 20. yüzyılın en büyük bilimadamı olan Turing, çok yönlü bir dehaydı. Biyolojiye, neredeyse boş vaktinde diyebileceğimiz bir kolaylıkla, dev bir

kuramsal katkı yapmıştı. Ulusal takıma seçilebilecek performansta başarılı bir koşucuydu (Şehirlerarası yolculuk yapacağı zaman bavullarını trenle gönderip kendisi yaya olarak giderdi). Ha, unutmadan, bir de 2. Dünya Savaşı'nı kazanmıştı!

Turing üniversite hocalığı yoluna koyulduktan kısa süre sonra, daha 24 yaşındayken Hilbert'in (Gödel'in attığı bombadan sonra hâlâ ayakta kalmış) bir başka hayalini yıkmış, ama bu arada da makineler çağını açmıştı. İşin bu yanını sonraki sayfalarda ayrıntılı işleyeceğiz. Ama önce bu süper adamın yaşamının geri kalanını bir gözden geçirelim.

İngiltere Hitler'le savaşa girdiğinde doktora çalışmasını yeni bitirip ABD'den yurduna dönmüş olan 27 yaşındaki Turing, üniversitesinden Hükümet Kod ve Şifre Okulu'nun Bletchley Park'taki karargâhına taşındı. Görevleri, Alman ordusunun radyo iletişimini gizlemek için kullandığı Enigma şifresini çözmekti. Bu bir ekip işiydi, ama başarıya giden yolda en büyük katkının Turing'den geldiğini rahatlıkla söyleyebiliriz. Savaşın ortalarına doğru Bletchley Park'ta gece gündüz her 30 saniyede bir Alman mesajı çözülür hale gelmişti. Askeri tarihçiler Turing'in sağladığı bu istihbarat avantajının Almanya'nın yenilmesini iki yıl kadar öne çektiğini söyler.

Turing (göreceğimiz gibi) bugün "bilgisayar" denen makinenin kuramsal temelini savaştan önce atmıştı, savaş bittikten sonra Londra'da, daha sonra da Manchester Üniversitesi'nde, ilk gerçek elektronik bilgisayarları inşa eden uzmanlardan biri olarak çalıştı. 1950'de yapay zekânın işaret fişeği sayılan "Hesaplama makineleri ve zekâ" makalesini yayımladı. 1951'de (Leibniz'i önce onurlandırıp sonra yüz üstü bırakan) Kraliyet Akademisi'ne üye seçildi. 1952'de canlılarda benekler ve çizgilerin nasıl oluştuğunu açıklayan harika eserini ortaya koydu. Ama 1952 Turing'in şansının döndüğü yıl da oldu.

Turing eşcinseldi ve bu o çağda İngiltere'de suçtu. Evinde gerçekleşen bir hırsızlıktan ötürü şikâyette bulunduğu polis komiseri, olayın detaylarını araştırırken Turing'in bir

erkekle olduğunu anladı. Yalanı sevmeyen birisi olan Turing, soru kendisine sorulduğunda inkâra çalışmadı. Mahkeme kendisine hapis cezası ile hormon "tedavi"si denilen bir tür işkence arasında seçenek sununca özgürlüğü seçti.

7 Haziran 1954'te, daha 41 yaşındaki Turing yatağında ölü bulundu. Ölüm nedeni siyanür zehirlenmesiydi. Genel kanı, hormon işkencesinin vücudunda yarattığı değişimin tetiklediği bir bunalım sonucu intihar ettiği yolundadır. Ama farklı düşünceler de vardır: Örneğin annesi, yanında siyanüre bulanmış ve ısırılmış bir elmayla ölü bulunan Turing'in kimya deneylerinden sonra ortalığı toparlayıp temizleme huyu olmaması yüzünden kaza sonucu öldüğüne inanıyordu (Tam da annesi böyle düşünsün diye bu mizanseni hazırlamış olabileceğini söyleyenler de vardır). Sevgili dostum Orhan Bursalı'nın Turing'in, eşcinsel casusların Sovyetler'e bilgi sızdırmasından kaygılanan İngiliz istihbaratınca öldürüldüğü tezini de unutmayalım.

2009 yılında aklı başına geç gelen İngiliz devletinin başbakanı Gordon Brown, Turing'e yapılan muameleden ötürü hükümeti adına resmen özür diledi. 2013'te Kraliçe II. Elizabeth Turing'in çoktan kaldırılmış o saçma yasayı çiğneyerek işlediği "suç"u affetme kararı aldı. Birkaç yıl sonra parlamento aynı durumdaki (15.000'i hâlâ hayatta olan) 65.000 erkeğin de affedilmesini öngören bir yasa çıkardı. Bu yasa "Alan Turing kanunu" olarak biliniyor.

8 | Turing makinesi nedir?

David Hilbert'in matematiğin eksiksiz olduğuna, yani doğru olan her şeyin kanıtlanabileceğine dair hayalinin 1930'da Kurt Gödel tarafından yerle bir edilişini 6. Soru'da anlatmıştım. Hilbert'in "Gödel bombası"ndan etkilenmeyen bir iddiası kalmıştı ama. Onu çürütmek de Turing'e nasip oldu.

1928'de Hilbert ve öğrencisi Wilhelm Ackermann, matematik dünyasına, herhangi bir mantıksal önermenin verilen belitler kullanılarak kanıtlanabilir olup olmadığını saptayabilen bir yöntemin (şimdi kullandığımız terimle, bir "algoritma"nın) bulunması için çağrıda bulundu. Böyle bir algoritmanın var olduğundan hiç kuşkuları yoktu, sadece birisinin onu keşfetmesi gerekiyordu.

"Karar Problemi" (ya da cafcaflı Almanca adıyla "Entscheidungsproblem") olarak bilinen bu sorunun Gödel'in eksiklik teoremlerinden etkilenmediğini görüyorsunuz, değil mi? Gödel (eğer matematik umduğumuz gibi çelişkisizse) her doğru önermenin bir kanıtının olamayacağını göstermişti. Bu, verilen bir önermenin doğru mu yanlış mı olduğunu saptamaya çalışan bir algoritmanın her önerme için başarılı olamayacağı anlamına geliyordu. Karar Problemi ise cevap olarak "doğru" veya "yanlış" değil, "kanıtı var" veya "kanıtı yok" veren başka bir algoritma istiyordu. Böyle bir algoritma hem tüm yanlış önermelere, hem de kanıtı olmayan doğru önermelere "kanıtı yok" deyip bizi doğru ve kanıtlanabilir olan "iyi huylu" önermelerle baş başa bırakabilirdi. İşte Alan Turing'in 1936'da, daha 24 yaşındayken çözdüğü bilmece buydu: Böyle bir algoritma yoktu! Dolayısıyla, matematik teoremlerini diğer cümlelerden ayırt edebilen bir "mantık makinesi" yapmak da mümkün değildi. Leibniz'in rüyası suya düşmüştü. Ama Turing bu önemli buluşun tadını çıkarabildi mi dersiniz?

Okyanusun diğer yanında, Princeton'da matematikçi Alonzo Church de aynı konuda yıllarca çalışmış ve Turing'inkinden farklı bir yöntemle aynı sonuca ulaşarak birkaç ay önce yayımlamıştı. Bu nedenle Church'den habersiz çalışan Turing'in makalesini gönderdiği dergi özgün olmadığı gerekçesiyle yayımlamaya yanaşmıyordu. Sorunu iki makaleyi de okuyan Cambridge hocası Max Newman çözdü. Newman, Turing'in ispatında kullandığı yöntemin kendi başına önemli bir katkı olduğunu fark etti ve yayıncıları ikna etmeyi başardı. Böylece bilgisayar biliminin kurucu metni yayımlanabilmiş oldu.

Peki Church'ün çalışmasında olmayıp Turing'inkinde olan, bu nedenle işin uzmanlarına, sözgelimi Kurt Gödel'e "Hah işte, Karar Problemi şimdi çözülmüş oldu" dedirten bu ikna edici yöntem neydi? Bunu anlamak için Karar Problemini tanıtırken değindiğimiz bir noktayı tekrar vurgulamak gerekli: Problem bir algoritmanın ortaya konulmasını istiyordu, ama o zamanlar "algoritma" dediğimiz şeyin net matematiksel bir tanımı yoktu. Akıllara "bir hesaplamayı gerçekleştirmek için kullanılabilen, muğlak olmayan, basit adımlardan oluşan bir yordam" gibi bir tarif geliyordu, ama bir şeyin böyle bir yöntem olup olmadığını kesinkes ayırt etmeye yarayan bir tanım mevcut değildi. Gödel Church'ün çalışmasında esas aldığı oldukça soyut çerçevenin bu anlamda her "yapılabilir hesap işi"ni gerçekleştirebileceğine ikna olmamıştı örneğin.

Turing ise makalesine matematiksel işlemler yapan (ve yorulma, uyuma, acıkma, yaşlanma, dikkat dağınıklığı, kâğıt/kalem eksikliği vs. pratik sorunları hiç yaşamayan) bir insanı temsil eden bir "makine" türünü gerekli netlikte tanımlayarak başlıyordu. Bu bağlantıyı, aşağıda göreceğiniz gibi, o kadar basit şekilde anlatıyordu ki, okuru insanların yapabileceği her cins hesap işinin bu türden bir makinece de gerçekleştirilebileceğine ikna ediyor, böylece nihayet "algoritma" kavramına bir tanım getirmiş oluyordu. İspatın geri kalanındaki birkaç başka harika fikri önümüzdeki sorularda göreceğiz, ama şimdi şu makineye odaklanalım.

Daha sonra Church'ün taktığı adla, bir "Turing makinesi" (TM) şu iki bileşenden oluşur:

- Kareli bir defterin ilk satırını (ya da eğer isterseniz bir tuvalet kâğıdını) sonsuza dek uzatarak gözünüzde canlandırabileceğiniz, her karesine bir sembol yazılabilen ve istenen karesine erişilip oradaki sembolün okunmasına ve dilenirse silinip yerine bir başkasının yazılmasına imkân veren bir teyp (Modellediğimiz insanın sınırsız miktarda kâğıt/kalem/silgi kullanabilmesi bu şekilde temsil ediliyor. Bu insanın belli bir anda kâğıdın

sadece bir noktasına bakabileceği gerçeğine de bu okuma/yazma işlemlerinin sadece teyp üzerinde sağa sola gezdirilebilen tek bir "teyp kafası"nın o anda üstünde olduğu karede yapılabilmesiyle sadık kalınıyor).

- (Modellediğimiz insanın kafasındaki bellek gibi) sınırlı bir miktarda bilgi tutabilen bir kontrol birimi. Uygulanacak yordamın adımları da bu bilgilere dahildir. Günümüzün terminolojisini kullanırsak, bu "program" makinenin bir sonraki adımında ne yapması gerektiğini belirler. Her program sonlu sayıda ("Şimdi üstünde olduğun karede Y harfini görüyorsan onu silip yerine Z harfini yaz, sonra kafayı bir soldaki kareye getir", "8 numaralı komuta atlayıp oradan devam et", "Çalışmayı durdur" vb.) basit "komut"lardan oluşur. Her komut birim zamanda icra edilebilir. Belli bir işlev için hazırlanmış bir TM'nin cevaplamasını istediğimiz soruyu teybine yan yana gerekli sayıda kareye birer sembol kondurarak yazabiliriz. Bu noktadan sonra makine çalıştırılır ve her adımda programında sıradaki komutu gerçekleştirir. Programın makineye "girdi" olarak verilen soru metnini teypten okuması, hesaplama işlemlerini gerektiğinde teypte yazıp silme yaparak icra etmesi, sonunda vardığı yanıtı da yine teybe yazdırıp çalışmayı durdurması beklenir (Kimi programlar kimi girdilerle başlatıldıklarında sonsuz bir döngüye girip hiç durmayabilirler, örneğin sürekli "teyp kafasını bir sağdaki kareye getir" komutunu tekrarlayan bir makine sonsuza dek çalışacaktır. Her girdi için sonlu sayıda adım sonucunda bir yanıt verip duran yordamları tercih ederiz elbet.)

Hepsi bu! Turing ne kadar karışık ve uzun olursa olsun her belirli hesap yordamının yukarıda anlatılan komutların basitliğinde parçalara bölünebileceğini, bu yordamı uygulayan bir insanın yaptıklarının da bu cinsten bir makine tarafından taklit edilebileceğini iddia ediyordu. Düşünün, sözü edilen türden bir işi yaparken (örneğin iki büyük sayıyı çarparken) kafanıza ilkokuldayken yerleştirdiğiniz bir programın adımlarını sırayla icra eder, kâğıtta

yazılı sayılara bakıp beyninizdeki çarpım tablosundan bu kurallara göre getirdiğiniz yeni bir şeyler yazar, en sondaki toplamayı da benzer şekilde halleder, sonunda da kendinizi aranan yanıtı yazmış bulursunuz, değil mi?

Church ve Turing birbirlerinin yöntemlerinin eş güçte olduğunu, yani Church'ün modeline uyan her hesap yöntemi için aynı işi yapabilen bir TM kurulabileceğini gördüler. Yıllar boyunca "algoritma" kavramı için önerilen çok sayıda diğer farklı görünüşlü model de incelendikten sonra, bunların her birini taklit edebildiği görülen TM modelinin kavramın "resmi" tanımı olarak kabul edilmesi konusunda bir kuşku kalmadı.

9 | Evrensellik nedir?

Buraya kadarı bile büyük başarıydı, ama Turing'in makalesinin çağ değiştiren bulgusundan daha söz etmedik. Aklınıza gelen her algoritma için ayrı bir TM inşa edilebilir, örneğin ilkokulda öğrendiğimiz çarpma yordamını gerçekleştiren, girdi olarak verilen iki sayının çarpımını hesaplayan bir TM tasarlayabilirsiniz. Ama Turing o gün için çok şaşırtıcı olan bir uygulama gösterdi: Başka TM'leri taklit etme işi!

Turing "evrensel" bir TM tasarlamıştı. Bu makineye E diyelim. E'ye girdi olarak canınızın istediği (diyelim ki adı T olan) bir başka TM'nin tarifini ve yanına da T'ye girdi olarak verilecek soruyu verebilirsiniz. Bu durumda E, T'nin o soru üzerindeki çalışmasını (ara adımların tümünü teyp üzerinde hesaplayarak) baştan sona birebir taklit eder ve eğer T bu çalışmanın sonunda duracaktıysa da T ne yanıt verecektiyse aynısını verir. Turing, TM'lerin başka her sistemin benzetimini (simülasyon) yapabilme özelliğini, tek bir fiziksel makineye sonsuz sayıda değişik iş yaptırabilmek için kullanabileceğimizi keşfetmişti.

O güne dek insanlar her makineyi sadece tek bir iş için tasarlardı. Otomobiller ulaşım, buzdolapları soğutma, radyolar da iletişim içindi. Oysa Turing bize tek bir makine alıp ona birçok farklı makineyi taklit ettirebileceğimizi söylüyordu.

Bilgisayarlardan söz ettiğimizi anladınız, değil mi? Çağımızda bu "hesaplama evrenselliği" kavramını anlamak kolay, çünkü tek bir cihaz satın alıp onun üzerinde muhasebe programı, oyun programı, kelime işleme programı, resim yapma programı vs. bin türlü farklı program çalıştırarak bin değişik makinenin keyfini çıkartmak bize doğal geliyor. Ama Turing makalesini yazdığında, tüm bu farklı işleri yapabilen tek bir makine bilimkurgudan ibaretti. Savaştan sonra inşa edilen ilk elektronik bilgisayarlar doğrudan Turing'in buluşu olan "makineye çözülecek problemę ilişkin veriyle birlikte o problemi çözme yordamının tarifini de girdi olarak verme" fikrini hayata geçirdiler. Akıllı telefonlarımızı bize Alan Turing hediye etti.

Turing'in Karar Probleminin genel çözümünün olanaksızlığına ilişkin kanıtına önümüzdeki soruda döneceğiz, ama bir durup bu "evrensel makine" fikrinin sonuçlarını düşünün.

Diyelim başka her "problem çözme makinesi"ni taklit edebilen bir makineniz var. Başka makinelerin çözebilip de sizin makinenizin ne kadar sabırla beklerseniz bekleyin çözemeyeceği bir problem var mıdır?

Cevabın "hayır" olduğunu görebiliyor musunuz?

Cevap şu yüzden "hayır": Diyelim ki M adında bir başka problem çözme makinesinin belirli bir problemi çözebildiği, sizinkininse ne kadar zaman çalışırsa çalışsın bunu başaramayacağı iddia ediliyor. Ama sizinki evrensel, yani başka her makineyi taklit edebilen bir makinedir demiştik. Eh bu durumda makineniz M'nin o zor problemi çözerkenki halini de taklit edebilir, sonuçta M'nin problemin yanıtını ortaya koyuşunu da. Evrensellik bunu gerektirir! Demek ki M'nin yapabilip sizin makinenizin yapamayacağı bir iş yokmuş.

10 | Çözülemez problemler var mıdır?

"Problem" derken?

Sadece bir doğru cevabı olan sorular vardır. "17'ye 43 eklersek kaç eder?" mesela. (Cevap "60".) Bu soruyu hesap makinesinden kopya çekmeden yanıtlamak için kullandığınız yöntemi (ben ilkokulda öğrendiğim yöntemi kullandım) aynı türden sonsuz sayıda başka soruyu ("6555'e 13 eklersek kaç eder?", "484757'ye 864 eklersek kaç eder?", vs.) yanıtlamak için kullanabilirsiniz. İşte bu soru ailesine "Toplama Problemi", onlardan herhangi birini çözmek için kullandığımız ortak yönteme de "toplama algoritması" diyoruz.

Yani "problem" derken kastettiğimiz, bir soru kümesinden ibaret. Toplama, çıkarma, belirli bir hassasiyetle (diyelim ki virgülden sonraki onuncu basamağa dek) bölme, verilen bir listedeki isimleri alfabetik olarak sıralama, verilen bir haritada birbirine en uzak iki şehri bulma, vs. bir yığın (aslına bakarsanız, sonsuz sayıda) farklı problem var. Bu saydığım örnek problemlerin tümü için algoritmalarımız da var. Bu algoritmaların her biri, ilgili olduğu soru ailesinin her üyesini (bir süre çalıştıktan sonra) doğru şekilde yanıtlama garantisine sahip. İşte böyle, sorularının tümünü yanıtlayabilen tek bir yöntemin mevcut olduğu ailelere "çözülebilir problemler" diyoruz.

"Bir sorunun yanıtını bulmak" ile "bir problemin çözülebilir olduğunu göstermek" farklı şeyler. Aslında buradaki tartışmamız açısından ilginç olması için bir problemin sonsuz sayıda soru içermesi gerekiyor. Çünkü sonlu sayıda sorudan oluşan her problemin yukarıdaki tanıma göre kolayca "çözülebilir" olduğunu gösterebilirim. Nasıl mı? Diyelim ki problemimiz sadece bir sorudan ibaret, tek elemanlı bir küme. Bu sorunun yanıtını bulmak ne kadar zor olursa olsun, o yanıt sonuçta yazıya dökülebilecek bir şey olduğundan, onu yazacak bir algoritma muhakkak vardır.

Mesela "Başka galaksilerde hayat var mı?" sorusunun yanıtını kimse bilmiyor. Ama yanıtın ya "EVET" ya da "HAYIR" olduğu kesin. İki durumda da, bu soruyu girdi olarak alıp doğru cevabı yazabilen bir algoritma mevcut. O algoritma, şu iki algoritmadan biri:

Birinci Algoritma:
"EVET yaz, sonra dur."

İkinci Algoritma:
"HAYIR yaz, sonra dur."

Tek üyesi olarak "Başka galaksilerde hayat var mı?" sorusunu içeren problemin algoritması bu ikisinden birisi. Yani bu problemin çözülebilir olduğundan eminiz! Toplama gibi sonsuz sayıda sorudan oluşan problemler içinse yukarıdaki ucuz numarayı yapıp cevabı ezbere veren bir algoritma yok tabii; o yüzden gerçek bir "bilgi işlem" işi gerekiyor.

Peki bu tanımları temel alırsak, çözülemez problemler var mı? Evet, hatta birini geçtiğimiz sayfalarda gördük bile! Hatırlarsanız, Gödel'in eksiklik teoreminin, kendisine verilen her matematiksel önermenin doğru mu yanlış mı olduğunu söyleyebilecek bir algoritmanın var olmadığı sonucunu doğurduğunu belirtmiştik. Yani "Şu cümle doğru mu, yanlış mı?" problemi burada kullandığımız anlamda çözülemez.

Bunun bir doğa yasası olduğuna dikkatinizi çekerim. Albert Einstein aklını kullanarak evrende bir hız sınırı olduğunu keşfetmişti; ne kadar güç harcarsanız harcayın, asla ışık hızına erişemezsiniz. Bir problemin çözülemez olduğunu göstererek o soruların tümünü yanıtlayabilecek bir bilgisayarın veya başka herhangi bir yapay veya doğal sistemin asla var olamayacağını kanıtlamış oluyoruz. İlk soruda anlattığım, habire "Makineler asla şu işi yapamaz" deyip deyip yanılan yapay zekâ muhaliflerini hatırlıyor musunuz? İşte nihayet gerçekten makinelerin yapamayacağı bir iş bulduk! Ama bu buluş o itirazcıla-

rın düşlediği gibi insanla makine arasına bir set çekmedi, çünkü çözülemez problemleri (adı üstünde) insanlar da çözemiyor!

Artık Turing'in Hilbert'in Karar Problemini çözme rüyasını tuzla buz edişinin öyküsünü tamamlayabiliriz. Bu soruda öğrendiğimiz deyimleri kullanarak hatırlatalım: Turing'in amacı, verilen herhangi bir önermenin kanıtlanabilir olup olmadığını saptama probleminin çözülemez olduğunu göstermekti. Bunun için iki adımlı bir yol izledi.

Birinci adımı anlamak için şu problemi düşünün: Ben size bir bilgisayar programının metnini vereceğim. Siz de bu programı inceleyip sonlu bir zaman içinde bana bu programın çalıştırılması halinde sonsuz bir döngüye girip girmeyeceğini söyleyeceksiniz. Bu işi size verebileceğim her girdi (yani her farklı program) için doğru cevabı vererek bitirmenizi sağlayacak bir algoritma var mıdır? Dikkat edin, "Verilen programı çalıştırıp bakarım, cevabı ona göre veririm" diyemezsiniz, çünkü bunu yaparsanız ve programın çalışması uzun süre beklemenize karşı bitmezse onun sonsuz bir döngüye mi girdiğini, yoksa sadece uzun süre çalışmış ve artık durmasına az kalmış bir program mı olduğunu bilemezsiniz. Sizin cevabınızı sonlu zaman içinde doğru olarak vermeniz bekleniyor. Karara benzetim yoluyla değil, "inceleme" yoluyla varmanız gerekmekte.

(Laf aramızda, böyle bir algoritma bulsanız çok iyi olurdu, çünkü yazılım geliştiren insanlar bu kontrolü yapabilmeyi gerçekten çok isterler. Yani bu işi yapan bir programı çok iyi fiyata satabilirdiniz. Ama maalesef...)

Turing'in makalesi için yarattığı göz kamaştırıcı basitlikteki yöntemle, işte bu "Durma Problemi"nin de hiçbir TM (ve evrensellik özelliği dolayısıyla hiçbir bilgisayar programı, hiçbir insan, vs.) tarafından yukarıda istenen genellikte çözülemeyeceği gösterilebilir.

Demek ki birinci adımın sonunda elimizde çözülemez bir problem var, ama bu hedefimiz olan Karar Problemi değil, az önce uydurduğumuz Durma Problemi. İkinci

adımda yeni bir numara yapacağız: Buna "bir problemi diğerine indirgeme" diyoruz.

Diyelim A ve B adında iki farklı probleminiz var. "A'yı B'ye indirgemek" isterseniz, yapmanız gereken, eğer bir gün size B probleminin algoritması verilirse onu kullanarak A için bir algoritma kurabileceğinizi ve dolayısıyla A ailesinden istediğiniz her soruyu yanıtlayabileceğinizi kanıtlamaktır.

(Örneğin mutlu olmak için zengin olmanın yeterli olacağına inanan birisi, mutluluğun zenginliğe indirgenebileceğini sanıyor demektir. Zengin olmak da çok satan bir yapay zekâ kitabı yazmaya indirgenebilir mi acaba? Neyse...)

İşte Turing'in ispatının son adımında yapılan budur: Karar Problemini çözen bir algoritmamız olsaydı, ondan yararlanarak Durma Problemini çözen algoritmaya nasıl ulaşabileceğimizi gösteririz. Ama az önce Durma Probleminin çözülemez olduğunu göstermiştik! Demek ki Karar Probleminin de hiçbir algoritması olmamalı. İspat bitmiştir. Kusura bakma, Hilbert.

Sonraki yıllarda bu "kararlaştırılamazlık" özelliğine sahip birçok problem daha bulundu. İndirgeme, nice ispatlarda kullanılan standart bir yöntem oldu. 24 yaşındaki bir genç adam tek başına günümüzde bilgisayar bilimi denilen bilim dalını kurmuş, yepyeni doğa yasalarının keşfinin kapısını açmıştı.

2. Bölüm
BEYİNLER ve DİĞER BİLGİSAYARLAR

11 | Bilgisayar nedir?

Kimi okurlar bunun ayrı ele alınmayı hak etmeyecek kadar basit bir soru olduğunu düşünebilir. Oysa bu kitaptan tek bir şey hatırlayacaksanız, bunu hatırlamanızı isteyeceğim kadar önemli bence.

Bilgisayar denen şeyin ne olduğu konusunda kuşkusuz bir fikrimiz var. Geçtiğimiz sayfalarda bilgisayar biliminin esas nesnesi olan Turing makineleriyle tanıştığımıza göre kuramsal modelinden de haberdarız. Artık "bilgisayar" kavramını tanımlayabiliriz.

Ama dikkatli olmamız gerek. Tanımımızın bu unvanı hak eden şeyleri dışarıda bırakmasına da, hak etmeyenleri kapsamasına da meydan vermemeliyiz. Benim mezun olduğum yıllarda "bilgisayar" kelimesi kimsenin aklında insanların ceplerinde taşıdığı bir makineyi çağrıştırmıyordu, ama şimdi durum değişti (Bu kitabın tamamına yakını telefonumda yazıldı. Önceki cümle sadece birkaç yıl

öncesine kadar deli saçması sayılırdı). Tanımımızın önümüzdeki yüzyılların bilgisayarlarını da kapsaması lazım.

Bilgisayar kavramında ne fare, klavye, ekran, hoparlör gibi aksesuvarlar, ne de yıllar geçtikçe sürekli değişmekte olan işlemci hızı, bellek kapasitesi gibi detaylar yaşamsal. Dahası, makinenin hangi malzemeden, ne tip moleküllerden yapıldığının da bir önemi yok. Çağımızda tümleşik elektronik devrelerde genellikle silisyum kullanılıyor ama bu şart değil, bambaşka gereçler kullanarak da aynı hesaplamayı gerçekleştiren bir sistem kurabilirsiniz.

Bu çok önemli ve kuramsal bilgisayar biliminin konusunun tam olarak ne olduğunu anlamamızda çok kritik olan bir husus. Biz kuramcılar için önemli olan bilgisayarın ağırlığı, rengi veya elektrikle mi yoksa Leibniz'in hesap makinesi gibi kol gücüyle mi çalıştığı değil, yaptığı iş. Bu işe (atalarımızın hesap makinelerine hürmeten) "hesaplama" (İngilizcesi "computation") diyoruz.

Bir de "bilgi işlem" lafı var. Kıyma makinelerinin ve motorların aksine bilgisayarların girdisinin de çıktısının da "bilgi" olması önemli (16. Soru'nun cevabında bu kelimeyi ele alacağız). Elbette masaüstü bilgisayarınız da girdi olarak elektrik ve tuş darbeleri alıp, çıktı olarak ses, ışık ve biraz ısı üreten bir fiziksel sistem ve bunlar bilgisayar üreticileri için önemli olabilecek hususlar, ama şimdiki tartışmamız açısından mühim değiller. Bilgisayarın fonksiyonu, hangi problemi çözdüğüne, yani hangi algoritmayı uyguladığına bağlı. Aynı algoritma aynı girdiyle çok farklı fiziksel altyapılara dayanan farklı bilgisayarlarda icra edilebilir, o sırada her bilgisayar farklı sesler çıkartıp farklı miktarda enerji harcayabilir, işlerini farklı sürelerde de bitirebilir, ama tümü de aynı sonuca ulaşıp aynı çıktıyı verecektir. Hesaplamanın fiziksel altyapı seçimine bağımlı olmaması olgusunun teknik adı "alt katmandan bağımsızlık"tır. Bu sayede bilgisayar programlama derslerinde cihazı oluşturan maddelerin kimyasından bahsetmemize gerek kalmaz.

Tek bir işe özgü, örneğin sadece iki sayıyı ve bir aritmetik işlem sembolünü girdi alıp o işlemi o sayılara uygulayacak şekilde inşa edilmiş makineler de Turing maki-

nesi olarak modellenebiliyor ve ben onlara da "bilgisayar" deme yanlısıyım, ama bilgisayar denince genelde akla programlanabilen, yani farklı işleri nasıl yapacağına dair talimatları girdi olarak belleğine alıp uygulayabilen cihazlar geliyor. Başka her bilgisayarı bu şekilde taklit edebilme özelliğine "evrensellik" denildiğini 9. Soru'nun cevabında görmüştük.

Hesaplama yapabilen her sisteme "bilgisayar" demenin şöyle ilginç bir sonucu oluyor: Bu durumda insanlar da bilgisayar! Bellek ve zaman kısıtlarını bir an için göz ardı edersek, insanlar kuşkusuz kimi algoritmaları ezberleyip icra edebiliyorlar. Dahası, kendilerine verisiyle birlikte (diyelim ki bir kâğıda yazılı olarak) verilen, daha önce hiç görmemiş oldukları bir algoritmayı (mesela bir TM tarifini veya bildikleri bir programlama dilinde yazılmış bir bilgisayar programını) o veri üzerinde adım adım gerçekleştirmeyi de beceriyorlar. Yani evrensellik özellikleri de var gibi. Zaten henüz olmamış eylem ve olayların kafamızda benzetimini yapma yeteneğimizin olduğunu biliyoruz, bu da evrensel bir makineye gereken taklit yeteneğine pek benziyor.

Ama bir dakika! Çok mu ileri gidiyoruz ne? Matematik ve mantık kurallarına bağlı kalarak işlemler yapan bir insanı örnek alarak Turing makinesi diye bir model yarattık. Bunun üzerine bilgisayar bilimini kurduk. Tamam, iyi bilgisayarlar TM benzetimi yapabildiklerinden bu modelin gücüne erişebiliyorlar. İnsanlar da bunu yapabiliyor. Ama bu yüzden insanlara "bilgisayar" demek, bilgisayar kavramını fazla genişletmek olmuyor mu? İnsanlar hep ispat yapan matematikçi modunda çalışmak zorunda değil ki! Daha "insani" şekilde davrandıklarında, normalde "bilgisayar" dediğimiz aygıtların asla taklit edemeyeceği şeyler de yapamazlar mı?

Burada 1. Soru'nun cevabında tartıştığımız "Makineler düşünebilir mi?", "İnsan düzeyinde yapay zekâ mümkün mü?" sorularına döndüğümüzü fark ettiniz mi? Artık onlara yanıt verme vakti geldi.

Benzetim kavramının güzelliği şudur: Parçalarının çalışma ve birbirleriyle etkileşme prensiplerini bildiğiniz

her sistemin benzetimini (gelecek davranış hesaplamasını ya da sık kullandığımız deyimle, taklidini) kâğıt-kalemle yapabilir, ya da bunu yapan bir TM (bilgisayar programı) inşa edebilirsiniz. Örneğin, belli bir hızla havaya atılan cisimlerin Dünya'nın yerçekimi ve atmosferiyle nasıl etkileşeceğine ilişkin kuralları yüzyıllardır biliyoruz, o nedenle fırlatma yönü, hızı, cismin kütlesi vs. verileri alıp o cismin kaçıncı saniyede hangi konumda olacağını adım adım hesaplayarak uçuş simülasyonunu yapan bir program yazabiliriz. Veya parçacık fiziğini bildiğimiz kadarıyla, evrenin herhangi bir köşesindeki canlı ya da cansız herhangi bir fiziksel varlığın (iç yapısı ve etkileştiği ortam hakkında yeterince detaylı bilgimizin, bunları depolayabilecek büyüklükte belleğimizin, süreci izleyecek sabrımızın vs. olduğunu varsayarak), şu andan itibaren gelecekte nasıl evrilip nasıl davranacağını (tüm temel parçacıklarının birbirleriyle itişip çekişerek her adım sonrasında ne duruma geçeceklerini) hesaplayan, yani o varlığın cevaplayabileceği herhangi bir soruyu bu şekilde onu taklit ederek cevaplayabilen bir makineyi prensipte inşa edebiliriz. Uzun lafın kısası, bilgisayarlarımıza insanları birebir taklit ettirebiliriz (Görünüş, boy bos, doku, koku gibi detaylar hem benim çok ilgimi çekmediğinden, hem de insansı robot teknolojisi bu konuda ikna edici bir hızla ilerlediğinden, "taklit" derken esas önemsediğimin söz ve davranışlar gibi bilişsel çıktılar olduğunu fark etmişsinizdir). İnsanlar ve (diğer) bilgisayarlar karşılıklı olarak birbirlerini taklit edebildikleri için (bellek ve hız gibi başta göz ardı ettiğimiz hususlar dışında) birinin diğerine bir üstünlüğü olduğu söylenemez.

Müthiş, değil mi? İnsanlar dahil evrendeki her şeyin fiziksel sistemler olduğunu, fiziksel sistemlerin makro düzeydeki davranışlarının onları oluşturan parçacıkların basit yasalarca yönetilen mikro düzeydeki davranışlarına bağlı olduğunu, bu mikro âlemin yasalarının da hesaplanabilir şeyler olduğunu kabul ettiğinizde, bir hesap makinesi çok ulvi bir nitelik kazanıyor. İşin bu yönüne sonraki soruda döneceğiz.

Genelde bilgisayarları anlatan kitaplarda insanlar "kullanıcı" rolündedir. Oysa resme doğru bakarsak, insanın beyninin sürekli olarak duyu (girdi) organlarından bilgi alan, fizik kurallarına (tabii ki) dayalı bir mekanizmayla o bilgiyi işleyen, hesapladığı çıktıyı da bedenin çeşitli yerlerine sinyaller olarak gönderen bir bilgisayar, bütününün de et ve kemik gibi malzemelerden yapılmış bir "otonom robot" olduğunu görürüz. Bu bilimsel anlayışın insan denen varlığın değerini azalttığını ileri sürenlere aldırmayın. Robotuz dediysek harika robotlarız, en azından bu gezegende kendi varlığının sırrını çözmeyi ilk başaran varlıklar biziz ve anlama serüvenimiz daha yeni başlıyor. Bilgisayar bilimi de tıpkı fizik ve matematik gibi bu serüvende kullanacağımız temel araçlardan biri.

12 | Doğanın programlama dili nedir?

Masanızda veya cebinizdeki bilgisayarınızda donanım/yazılım ayrımını net şekilde görebilirsiniz. "Donanım", makinenin elle tutulup gözle görülen, kütlesi olan, bugünden yarına önemli şekilde değişmeyen fiziksel kısmıdır. Ama bilgisayarınız sadece bundan ibaret değildir. Telefonunuza bugün yeni bir uygulama programı indirip kullandınızsa, o artık dünkü telefonunuzdan farklıdır. Ne değişti? Yazılımı.

Aynı uygulamayı milyonlarca başka insan da yükleyebilir, ama bu yüzden ne sizin, ne de onların telefonlarının ağırlığı birazcık bile artmaz. Demek ki yazılım madde veya enerjiden değil, bilgiden oluşan ilginç bir şey (Kuşkusuz her programın ille de cihazın belleğinde bir yere "yazılması", yani madde ve enerji örüntüleriyle temsil edilmesi gerekiyor, ama geçen soruda gördüğümüz "alt katmandan bağımsızlık" olgusu nedeniyle, o maddenin tam olarak ne olduğu kritik değil, donanım bambaşka bir maddeden

yapılsa da program aynı program). İnsanın donanım/yazılım ilişkisine "beden/ruh ilişkisiyle aynı şey" diyesi geliyor.

Turing makinesi kavramına aşina olduğumuz için bu işlere bilimsel yaklaşabiliriz. Kendisine tarifi verilen (teybine girdi olarak yazılan) başka başka algoritmaları çalıştırma kabiliyetine sahip Turing makinelerinin olduğunu görmüştük. Yani bir adet fiziksel makine (donanım) inşa etmek (veya satın almak) gerekiyor, o donanıma farklı farklı işler (belleğine farklı farklı şeyler yazarak) yaptırılabiliyor. Benzer şekilde, "bilgisayar olarak insan" modelimizde de bir kafada birçok farklı düşünce, bilgi, fıkra, şarkı vs. bulunabiliyor. Elektronik bilgisayarlarımızda "program" dediğimiz o elle tutulmayan, ama "makineye can katan" şeylerin insanlarda da böyle bir karşılığı var.

Yapay zekâya insanı taklit ederek ulaşmak isteyenlerin bu taklidi hangi düzeyde yapacaklarını düşünmeleri gerekli. Çünkü çoğu karmaşık sistem gibi insanlar da birçok "katman"da modellenebiliyor ve bu katmanların kuralları (deyim yerindeyse, o düzeylerde icra edilen algoritmaların yazıldığı programlama dilleri) birbirlerinden farklı.

En altta evrenin temel kuralları var. Ezelden beri geçerliler. Ne kadar hızlı koşarsanız koşun ışık hızına varamayacağınızı, iki kere ikinin dört ettiğini, kütleyle enerjinin arasındaki ilişkiyi filan belirleyen bu kurallar. Nasıl Karayolları Trafik Kanunu'nda ülkedeki her otomobilden, her yoldan, her kavşaktan, her yayadan ayrı ayrı bahsedilmiyor, ama bunların tümü yine de o kanuna göre işliyorsa (tamam, Türkiye'de bunun çok kötü bir benzetme olduğunun farkındayım!) sözünü ettiğim fizik yasaları da her temel parçacıktan ayrı ayrı söz etmiyor, ama her şey o yasalara uygun işliyor.

Yani en alt düzeyde bakarsak evrende "fizik kuralları" adında bir program çalışıyor (Bunun daha altında bir katman olmadığından bu program, yukarıda telefonlarımızdakiler için söylediğimin aksine, bir yere açıkça yazılı değil, sadece "var". Fizikçilerin işi bu programı keşfetmek. Bu kuralları yazmaya kalksak o kadar da uzun bir

metin çıkacağını pek sanmıyoruz, "programlama dili"nin de "matematik" dediğimiz, geliştirilme/keşif serüveninin bir kısmını gördüğümüz sistem olduğuna eminiz). Bu programın girdisi evrenin şimdiki halinin bir tarifi, çıktısı da bir sonraki andaki halinin. Evrenin programı sürekli aktif. Bu bakış açısıyla bilgisayar evrenin kendisi, hesapladığı şey de kendi geleceği.

(Bu programın nereden çıktığını veya neden başka türlü değil de böyle olduğunu merak edenlere Max Tegmark'ın "*Matematiksel Evrenimiz*" (*Our Mathematical Universe*) kitabını önermekle yetineyim).

Önceki soruda bir insanın tümüyle bir bilgisayar tarafından benzetimlenebileceğini öne sürerken bu en alt katmanın terimlerini kullanmıştım. Bu katman bağlamında insanları oluşturan parçacıklar da fiziksel etkileşim içinde oldukları çevreleriyle birlikte sürekli bir sonraki anda ne olacaklarını hesaplıyorlar. Oysa bize asıl derdimiz bu değil gibi geliyor, değil mi? Atomlarımızın değil, çocuklarımızın nerede olacağını hesaplamaya çalışıyoruz. Bir sonraki soru bu düzeyle ilgili.

13 | Yaşamın programlama dili nedir?

Uzun süre önce ortalıkta sadece kuark-gluon çorbası vardı. Tabii ki kuarklarla gluonlar da fizik yasalarına göre devindi. Aradan zaman geçti, hidrojenden yıldızlar doğdu, onlar diğer elementleri pişirdi, gezegenler oluştu, kimileri birbiriyle çarpışıp tuzla buz oldu. Bunların hepsi kalabalık bir bilardo masasındaki topların devinimi gibi, göze ne denli karışık görünse de o hep geçerli temel basit kurallara uyarak gerçekleşti. Bir süre sonra Dünya'da bir denizde, o zamanki koşulların tetiklediği sayısız kimyasal tepkime zincirlerinden birinin sonunda, çevredeki hammaddelerden kendi kendisinin

yeni bir kopyasını üretebilen bir mekanizma oluştu. Bu, tarihin en önemli olayıydı. Bu andan itibaren olanları da sadece maddenin yapıtaşlarının fizik yasalarına göre birbirleriyle çarpışması, sekmesi, takılması, sökülmesi olarak anlatmaya devam edebiliriz elbet, ama bu "öz kopyalayıcı" düzenek bize hikâyeyi başka bir dille anlatma olanağı veriyor: Bireylerin, üremenin, nesillerin, türlerin konuşulduğu, ama asıl aktörün genler olduğu yaşamın diliyle.

Bu sahipsiz kimya laboratuvarındaki tepkime çorbasının içinde bir gün, öz kopyalayıcı bu ilk düzeneklerden biri, ya da belki de birbirlerinin kopyalarını üretmeye yarayan iki düzenek (A hammaddelerden B kopyaları üretiyor, B de A kopyaları üretiyor) kendilerini kimyasal doğası gereği kolayca kabarcık şeklini alan bir lipit zarının içinde buldu. Üremelerine ket vuracak başka moleküllerden korunmalarını sağlayan bu hücrenin içinde çoğaldıkça çoğaldılar, baloncuk bu içeriği üleşen iki baloncuğa dönüştü. Sonra onlar da bölünüp çoğaldıkça çoğaldı.

İlk canlının ortaya çıkışından kısa süre sonra sayısız kopyası olmuştu. Gelgelelim, "tek rakibi kendisi" olan bu ilk tür, Dünya'nın ilelebet hâkimi olamadı. Kopyalama süreci dış etkenlerin de işe karışmasıyla mükemmel çalışmıyor, arada bir orijinalinden azıcık farklı bir kopya ortaya çıkıyordu. Bu farklılık yeni düzeneğin kendisini kopyalamasını engellemiyorsa bu kez bu yeni canlı çoğalmaya başlıyordu. Kaynaklar kısıtlı olduğundan, bu çoğalma yarışında diğerlerinden küçük de olsa bir avantajı olan türler daha çabuk çoğalıyor, altta kalanın canı çıkıyordu. Farklı coğrafi konumlarda hayatta kalıp üreyebilmek için farklı özellikler avantaj sağladığından, zaman içinde (daha az uyumlu olanların kaynaklara erişme ve üreme yarışını kaybedip tükenmeleri nedeniyle) gezegenin her köşesi tam da bulundukları yöreye uygun tasarlanmış, çok düşünülerek birbirleriyle müthiş uyumlu bir ortak yaşam ağına yerleştirilmiş gibi görünen, milyonlarca farklı türle doldu. Aslında her şey öz kopyalama, mutasyon ve rekabetin sadece koşullara uygun bedenleri üreyecek kadar süre hayatta bırakmasının doğal sonucuydu.

Dünyada bugün hüküm sürен tüm canlılarda bulunan DNA molekülü, daha küçük moleküllerin (Turing makinesinin teybi gibi tek boyutta) dizilmesiyle oluşmuş bir zincir. Bir bireyin DNA'sı, o bireyin sıfırdan nasıl inşa edileceğine ilişkin bilgileri kapsıyor. Her hücremizde tüm bu bilgiyi içeren DNA'mızın bir kopyası var. Adları A, T, G, C harfleriyle başlayan dört ünlü molekülün kopyaları art arda dizilerek "yazılmış" (aslında evrilmiş) bir harf dizisi olarak yorumlanabilen DNA, bilgisayar mühendisi gözüyle bakıldığında bir veri dosyası, bir işletim sistemi, veya belki en iyisi, bir programlar kütüphanesi olarak görülebilir. Bu son bakış açısına göre uzun DNA dizisinin bazı kısımları "gen" adını verdiğimiz programlar.

Bir bedende trilyonlarca hücre, bir DNA'da da on binlerce gen olabiliyor. Bütün bu programlar aynı anda çalışmıyor. Hücredeki kimyasal ortama göre sadece bazıları çalışıyor, diğerleri susuyor (Vücutta farklı görevleri olan hücreler bu açma/kapama sistemi sayesinde birbirlerinden farklılaşıyor).

Gen programlama dilinde A, C, G, T harflerinden oluşan her üç harflik dizinin bir anlamı var. Bir üçlü "program başı" anlamına geliyor, "program sonu" üçlüsü var ve diğer üçlü kombinasyonlar da değişik amino asitlere (bunlar meşhur moleküller) karşılık geliyor. Hücre içindeki şahane bir mekanizma (bu geni tetikleyen bir kimyasal durum varsa veya engelleyen bir durum yoksa) bu üçlüleri baştan sona kadar sırayla okuyup belirttikleri amino asitleri Lego parçaları gibi art arda birbirlerine takıyor. Sonunda ortaya çıkan şeye protein deniyor. Yani gendeki tarife göre özel bir protein üretilmiş oluyor.

Bu proteinler acayip moleküller! Ve bir sürüsü var. Her biri fizik kuralları gereği ayrı şekilde katlanıp ayrı bir biçim alıyor ve böylece proteinlerin kullanıldığı çok daha zengin bir Lego oyunu mümkün hale geliyor. Organlar hep bunların değişik kombinasyonlarından oluşuyor.

Şimdiye dek anlattığım kadarıyla gen programlama dili, okurlarım arasındaki bilgisayarcılara çok basit gelmiş olabilir. "Ne yani, program dediğin sadece bir dizinin

parçalarının art arda listelenmesinden mi ibaret? Değişkenler, döngüler, altprogram çağrıları, programlamayı zor yapan tüm o diğer özellikler nerede?" itirazı yükselebilir (Bunlar bir Turing makinesinin teyp kafasını sadece soldan sağa değil, sağdan sola da hareket ettirebilme ve teybe ilk yazılı olan dizideki harfleri değiştirebilme kabiliyetlerine denk geliyor). Ama durun, daha karmaşıklık denizine ayağımızı bile sokmadık.

Yukarıda hangi gen programının ne zaman çalıştırılacağını hücredeki kimyasal ortamın belirlediğini söylemiştim. Hücrenin ne hücresi (deri mi, kemik mi, sinir mi, vs.) olacağından ne zaman bölüneceğine dek bir yığın şey DNA üzerindeki çeşitli kimyasal tetiklerin çekilip çekilmemesine bağlı. Ve o tetiklemeler de başka genlerin yukarıda anlatılan şekilde vereceği çıktılara (ve bazen ne yiyip içtiğinize, bazen karanlıkta uyuyup uyumadığınıza ve bir sürü başka etkene) bağlı! Karmakarışık bir kontrol ağı, döllenmeden itibaren tüm hücrelerin bölünmesini, bölünmemesini, özelleşmesini, icabında ölmesini düzenleyerek sıfırdan bir canlıyı inşa etmek için her hücrenin genlerini uygun şekilde açıp açıp kapatıyor. Bilinçli bir mühendisin bir seferde tasarlamasıyla değil de DNA'da milyonlarca yıl boyunca rasgele mutasyonların ve değişen çevre şartlarının denk gelmesiyle oluşan kısmi değişikliklerle şekillenmiş bu işletim sistemi tam bir arapsaçı ve nice akıllı insan onca yıldır çalışma şemasını tam olarak çözmeye çalışıyor.

Aslında oyunun kuralı yaşamın başlangıcından beri aynı: Öz kopyalayıcıların daha çok kopyasının çıkarılması. Genler kendi kopyaları çoğalsın diye bedenler inşa ediyorlar, o bedenler rekabete rağmen üremeyi becersin diye de tasarım şeması koşullara göre değişiyor. Başarılı bireyler ürettirebilen genler, zaman içinde rakiplerinin yerini alıyor. Genler iyi yazılmış gibi görünen programlar, ama onları yazan bir "üst akıl" yok.

Bazı türlerde yine bu çağlardır süren "Bir birey inşa et ki genlerini paylaşan yeni bireyler üretsin" döngüsü sırasında "beyin" diye bir organ evrilmiş. Ama bazı beyinler kontrolden çıkıp oyunu bozabiliyor. Göreceğiz.

14 | Beyin nasıl bir bilgisayardır?

Birden fazla hücreden oluşan canlılarda bu hücrelerin birbirleriyle koordineli olarak çalışması gerekir. Örneğin süngerler suyu pompa gibi davranarak emerken içindeki besinleri süzer. Bu sırada bedenlerindeki kanalları sistematik olarak genişletip darlaştırmaları için her hücrenin diğer hücrelerin yaydığı sinyallere tepki vererek davrandığı bir sistem evrilmiştir.

Kimi sünger türlerinde bu sinyaller kimyasaldır, bir hücre bu işe özgü moleküller salgılar; diğerleri de suyun akışıyla taşınan bu moleküller yanlarına varınca algılar. Bu düzenek yavaştır, pompalama döngüsü dakikalar alabilir. Başka süngerlerde evrim çok daha hızlı bir alternatif keşfetmiştir: Hücre zarlarından iyon akımı yoluyla gerçekleştirilen elektrik sinyalleri.

Yani yaşamın altyapısı hücrelerin birbirleriyle haberleşmesi için kimyasal ve elektriksel sinyaller kullanmasına el vermekteydi. Bedenler büyüyüp organlar özelleştikçe eşgüdümün önemi de arttı ve uzmanlık alanları iletişim olan, tel benzeri uzantılarla birbirlerine bağlanarak bir ağ oluşturan sinir hücreleri evrildi.

Sinir hücrelerinin bedenin (gözlerden gelen girdi hatlarına yakın) bir köşesinde yoğun bir ağ oluşturduğu "beyin" denen organın evrildiği canlılara odaklanalım. Böyle bir ağ bellek işlevi görebilir (hücreler arasındaki bağlantıların güçlü mü zayıf mı olduğuna göre farklılaşan örüntüler farklı bilgileri kaydetmek için kullanılabilir), duyu organlarınca uyarılan hücreler bu ağa girdi, kaslar gibi hareket vs. yollarla dış dünyada etki yaratabilecek olanlara sinyal taşıyanlar da çıktı olarak görülebilir. Birbirlerini tetikleyen hücrelerarası etkinleşme örüntüleri kimi duyu girdilerine veya vücut kimyasından doğan sinyallere yanıt olarak karmaşık yordamların icra edilmesini sağlayacak şekilde dizilebilir. Demem o ki, beyin bir bilgi işlem makinesidir. Kelimeyi renkli bir benzetme olarak değil, teknik

bir terim olarak kullanıyorum; beyin tam teşekküllü bir bilgisayardır.

Bu önemli gerçeği sindirmek için günümüzün elektronik bilgisayarlarının temel mimarisiyle beynimizinkini karşılaştırmak yararlı olabilir. Karşılaştırma hakkaniyete uygun olsun diye şartları eşitleyelim: Bir bedeni sevk ve idare problemiyle uğraşan bir beyinle otonom bir robotun davranışlarını kontrol etmekle görevli bir bilgisayarı karşılaştıralım.

Kontrol bilgisayarı, robotun algılayıcılarından (kamera, mikrofon, pusula, jiroskop, basınçölçer, vs.) gelen verileri girdi olarak alır. Bu veriler söz konusu algılama cihazları tarafından 0 ve 1'lerden oluşan diziler olarak yorumlanabilen iki seviyeli elektrik sinyalleri olarak gönderilir. Zaten bilgisayarın içinde de her şey 0 ve 1'ler cinsinden yazılır.

Temel bilgisayar mimarimizde (tıpkı Turing makinesinin tek bir kontrol birimi ve tek bir teyp kafası olduğu gibi) veri üzerinde işlem yapan tek bir birim vardır. Üzerinde işlem yapılacak bilgiler, sıraları geldiğinde, bellekte tutuldukları yerlerden bu "merkezi işlem" birimine getirilir, sonuçlar da geri taşınır. Çoğu bilgisayarın belleği farklı hız ve kapasitede birkaç kısımdan oluşur ve işlenecek veriler o sırada büyük ama yavaş kısımdaysa erişilmeleri sistemi yavaşlatabilir.

Bilgisayarın o sırada çalıştıracağı (robotumuzun nasıl davranacağını belirleyen) program da kullanıcısı (belki de robotu tasarlayan mühendis) tarafından belleğe yerleştirilmiştir. Bilgisayar bir döngü içinde programın bir sonraki adımdaki komutunu işlemciye taşır, orada "okuduğu" komutun emrettiği işlem için gereken verileri getirir, işlemi ("şu 0, 1 örüntüsünü görürsen şu yeni örüntüyü kur" türünden bir dönüşümle) gerçekleştirir, gerekiyorsa sonucu taşır ve emredilen sıradaki komutu yükler. Program alabildiğine karmaşık bir hesaplama/işleme dizisini gerektiriyor olabilir, bilgisayar açısından fark etmez. Onun her adımda yaptığı budur.

Program alınan girdilere ve algoritmasına kodlanmış

olan amaca göre şu anda yapılacak hareketin ne olduğunu hesaplayınca robotun ilgili "eyleyici" birimlerine (tekerleği döndüren motor, ses çıkarılacaksa hoparlör, ışık yakılacaksa fener, vs.) gerekli sinyalin gönderilmesini emreder. Bu eylemin sonucunda durum (robotun konumu, kameranın aldığı görüntü, vs.) değişir ve "Şimdi ne yapmalı?" döngüsü sürer gider.

Gelelim beyne. Göreceğiz ki beynin yapısı daha karışık, anlaması da daha zor. Bunun sebebi, iki sistem arasındaki diğer farklılıkların sebebiyle aynı: Elektronik bilgisayar bir mühendislik ürünüdür, tek işlemcili TM modeli esas alınarak ve anlaşılması, hata teşhisi ve onarımı kolay olsun diye net sınırlarla ayrılan modüllerden oluşacak şekilde tasarlanmıştır. Beyin ise, anlaşılma veya tamirci tarafından sökülebilme kolaylığının hiç umursanmadığı, tek "amacın" eldeki şemada ufak tefek değişiklikler yaparak eldeki malzemeden çevreye daha uyumlu üretken bireyler yaratmak olduğu kör evrimin ürünüdür.

Beyin, duyu organlarından gelen verileri girdi olarak alır. Bu veriler söz konusu organlardan elektrik sinyalleri olarak gönderilir. Sinir hücrelerinin ("henüz etkinleşmedi" / "etkinleşti" ayrımıyla) rahatlıkla 0 ve 1 olarak yorumlanabilen bir veri taşıma kipi varsa da, bazı durumlarda etkinleşme sıklığı gibi 0, 1'den ziyade küsuratlı sayılarla daha doğal ifade edilen bir gösterimi destekliyor gibi görünürler.

Tek işlemcili, programın her adımının bir öncekinin bitmesini beklediği modelin tersine, beyindeki her sinir hücresi (yüz milyar tane olabilir!) aynı anda iş görebilir. Basitçe, her biri aslında aynı işlemi yapmaya kurguludur: Girdi tellerinden (duyu organlarından veya beyindeki diğer hücrelerden) gelen sinyallerin ağırlıklı toplamını hesapla, sonuç bir eşik değerin üzerindeyse etkinleşerek bağlı olduğun diğer hücrelere sinyal yolla! Bu hesapta kullanılan ağırlıklar, "Aynı anda etkinleşen iki hücrenin arasındaki bağlantının ağırlığı artar" gibi basit kurallarla güncellenebilir. Sonuçta beyin birbirini tetikleyerek yanıp sönen hücrelerin oluşturduğu etkinleşme örüntüleriyle dolup taşar.

Bu "dağıtık" mimaride verilerin ve komutların bellekten sırayla merkezi işlemciye taşınıp işlenmesi yoktur. Her şey her yerde gibidir! Bir adım sonra, hangi hücrelerin 0, hangilerinin 1 diyeceğini ve hangi bağlantının ağırlığının ne olacağını belirleyen "program", tüm bağlantıların şimdiki ağırlığına bağlıdır, yani beynin her yerine dağılmış vaziyette ve sürekli değişmektedir!

Bağlantı ağırlıkları uygun seçilirse duyu örüntülerinin iç örüntüleri, onların da başka iç örüntüleri tetiklediği çok karmaşık hesaplamalar, motor sinir hücreleri aracılığıyla da ilginç beden davranışları programlanabilir, hatta bu milyar işlemcili bilgisayar kendisini tek bir birey gibi hissedebilir. Evrim denen "kör programcı", bir sonraki soruda göreceğimiz gibi, bunu ve daha fazlasını başarmıştır.

15 | İnsanların programlama dili nedir?

Bir bedeni hareket ettirmek için çeşitli kontrol programlarının çalışması gerek. Onun hayatta kalması ve soyunu sürdürebilmesi için belli şekillerde davranmayı "istemesi" gerek, bu da örneğin vücut kimyasının o andaki durumuna göre alçak veya yüksek şiddette "Yiyecek bul! Yiyecek bul!" ya da "Seks yap! Seks yap!" diye bağırırcasına bedeni o işe yönlendirmeye çalışmakla görevli bir yığın programın beyin denen bilgisayarın içinde (masaüstü bilgisayarınızda aynı anda birden fazla program açabilmeniz gibi) aynı anda çalışacak şekilde evrimleşmiş olması sayesinde oluyor. Milyonlarca yılda başka başka zamanlarda evrilmiş bu çok sayıda program, bazen birbirleriyle etkileşiyor, bazen ortak kaynakları kullanmak için rekabete girişiyor. Duyu organlarından gelen görüntü ve ses gibi sinyalleri işlemeyle uğraşan programların süzdüğü bilgiye diğerleri de erişebiliyor.

Pek çok diğer canlı türüyle paylaştığımız bu program-

ların yanı sıra, biz insanlar gibi "ileri model" canlılarda bir de "kafada kurma" diye adlandıracağım bir yetenek daha var: Bu sistem, bir eylemi dış dünyada gerçekleştirmeden önce zihnin içindeki bir dünya modelinde onun bir tür "filmini" çevirip sonucun nasıl olacağını değerlendirmeye yarıyor. Örneğin "alt model"den yaratıklar bir uçurumun kenarına geldiklerinde yollarını değiştirmeden yürümeye devam edip telef olurken, kafasında "Bu adımı atarsam ne olur?" sorusuna yanıt olarak düşüp harap olmakla sonlanan bir benzetim yapabilen canlılar hayatta kalma avantajına sahip oluyor.

Bu "seçenekleri önce kafada kurup birer birer deneme" kabiliyeti av sırasında hareket tarzının planlanmasından kabiledeki ("A'ya destek verirsem rakibi B beni döver, öte yandan B'nin yanında olursam A kızını bana vermez" türü hesaplar gerektiren) güç ilişkilerine, soyunu sürdürmek için vazgeçilmez olan eş bulma, elde tutma ve sadakatinden emin olmaya çalışma işlemlerine dek o kadar çok alanda kullanılıyor ki, ilgili program son derece genel bir nitelikte, her tür "kur ve dene" senaryosunu işletebilecek şekilde evrilmiş. Eğer size resmedilen bir durumun iyi mi kötü mü olduğunu puanlayabilecek bir programınız zaten varsa, o zaman bir problem karşısında en iyi sonucu verecek davranış tarzını da sıfırdan alternatif davranışları kafanızda kurup, deneyip, sonuçta varacağını hesapladığınız durumu puanlayarak bulabilirsiniz. Bunun için zekâ değil, sadece işlem gücü gerekir. Ya da belki de zekâ dediğimiz aslında bu işlem gücünün ta kendisidir?

Genel bir benzetim motoruna sahip olmanın bize hesaplama evrenselliği kazandırdığını anlamışsınızdır. Burada altını çizmek istediğim, insanlığın en büyük buluşu olan dil yeteneğinin temelinde de muhtemelen bu "kur kur dene" altprogramının yatması.

Başka hayvanlarda da diğer bireyleri değişik tehlike türlerine karşı uyarmak için ses sinyalleri evrilmiş. Ama insan dili matematiksel olarak gelişmiş bir yapı. Konuşma sırasında nasıl oluyor da sonsuza yakın sayıdaki gramere uygun, fakat duruma uygunsuz lafı değil de sohbetin o

noktasına uyan bir cümleyi üretebiliyoruz? Belki sadece amaç ve zevklerimize uygun cümlelere olumlu tepki veren bir programımız var, söz sırası bize geldiğinde diyeceklerimiz de kafamızda yerli yersiz birçok seçeneğin kurulup bu programca denenmesi yoluyla seçiliyor.

"Ama ben zihnimde bütün bu programların aynı anda çalıştığını hiç hissetmiyorum" mu dediniz? İşte insanlık tarihinin en ünlü kurgusal karakterine geldik; şu "ben" adındakine.

Başka canlılarla, özelde de insanlarla birlikte yaşamak zorundayız. Onları birbiriyle etkileşen katrilyonlarca molekül veya binlerce program olarak düşünürsek, nasıl davranacaklarını mevcut zaman ve bellek kısıtlarımızla hesaplayamayız. O yüzden bizimkine çok benzeyen o başka bedenleri, kafamızdaki dünya benzetiminde birer özerk "birey" olarak modelliyoruz.

Nasıl bilgisayarınızın ekranında gördükleriniz o anda makinenin içinde çalışan çok sayıda programın çoğu hakkında hiç bilgi içermiyor, kalanları hakkındaysa çok yüzeysel, kullanıcı anlayabilsin diye süzülüp özetlenmiş bilgiler veriyorsa, kafamızda da etkilerinin toplamıyla bedenin davranışını belirleyen program kalabalığının tümünü değil, sadece sesi o sırada en yüksek çıkan programları kale alarak onların isteklerini bir "anlatı" haline getiren, diğer insanlar için geliştirdiğimiz "Bir bedenin bir sesi olur" yanılsamasına uygun olarak evrilen, dil modülüyle yakın bağlantılı, "bilinç" dediğimiz o tek sesi duyuyoruz. Doğasındaki filtreleme özelliği nedeniyle bilincim (yani "ben"!), o sırada "içeride" olup bitenin çoğundan habersiz. Davranışlarım genel vücut sisteminde alt katmandaki fizik yasalarına göre çalışan bir üst katmandaki sinir hücresi programlarınca belirlendikten sonra bilincimin de bundan haberi oluyor ve o da görevi gereği "eylemi üstleniyor", gerçekten inanarak "Ben karar verdim, ben yaptım!" diyor.

Matematiksel nedenlerle son derecede esnek bir sembolik sistem olan dil, evrilmesi sırasında hiç söz konusu olmayan, "gereksiz" ve hatta "imkânsız" şeyleri de

düşünülebilir kıldı. İş insanların "mantık" ve "matematik" denen tuhaf şeyleri icat etmesine, ya da belki de keşfetmesine kadar vardı.

İnsanların programlama dili ne midir? Hangi katmanda çalışmak istediğinize bağlı. En altta, fiziksel katmanda değişiklikler yaparak istediğimizi elde etmek, günümüz teknolojisinin hassasiyet düzeyinde epey zor. Beyne büyük mıknatıslar dayayarak sinir tellerindeki elektrik akımıyla oynayıp deneklerin istedikleri organını hareket ettirebilen veya çeşitli olaylar karşısındaki yargılarını değiştirebilen bilimciler hakkında belgeseller de izlemiş olabilirsiniz. Pek çok beceriyi edinmek için sürekli tekrarlar yapmak işe yarıyor; beynin tekrarlanan davranışları bir süre sonra otomatiğe aldığını herkes bilir. Ama en üst düzeydeki programlama yöntemleri elbette ki duyu organları ve dil kanalıyla çalışıyor. Donanım zayıflıklarından yararlanmak, sözgelimi cinsel çekiciliği kullanarak kandırmak çok etkin bir yöntem. Din ve ideoloji gibi bulaşıcı fikir kompleksleri bazı insanları doğal içgüdülerinin tam zıt yönünde, gen programını hiçe sayarak davranacak şekilde programlayabiliyor. Mantıksal dayanak ve delil göstererek ikna etmeye çalışmak da bir yöntem, ben de burada onu yapmaya çalışıyorum.

16 | Enformasyon nedir?

Turing'in "algoritma"yı tanımlarken kurduğu yapının sağlamlığı o zamana dek el yordamıyla kullanılan, ama bilimsel bir netliğe kavuşmamış başka önemli bilişsel kavramların da kuramsal bilgisayar biliminde yerli yerine oturmasını sağladı. Bunlardan biri de "enformasyon"dur.

Terminoloji notu: İngilizce "information" kelimesi karşılığı olarak "enformasyon"u kullanıyoruz. "Bilgi" demiyoruz, çünkü o burada tanımlayacağımız anlamda

enformasyondan farklı, daha "yüksek" seviyede, işlenip anlamlandırılmış malumata ("knowledge") karşılık geliyor gibi. Enformasyonun aksine, bilginin standartlaşmış bir tanımı yok ve literatürde farklı yerlerde birbirinden biraz farklı şeylere "bilgi" dendiğinden kafalar karışabiliyor.

Enformasyon kuramı, Claude Shannon'ın 1948'de yazdığı "Matematiksel Bir İletişim Kuramı" başlıklı mükemmel makalesiyle doğdu. Şimdilerde kimi ülkelerde biliminsanları makale başına ödül aldıkları için çalışmalarını olabildiğince küçük dilimler halinde yayımlayabiliyorlar. Shannon ise çok iyi düşünülüp geliştirilmiş harika bir yapıyı bir seferde ortaya koymuştu.

Shannon'ın ele aldığı problem şuydu: Bir iletişim hattı yoluyla haberleşmek isteyen A ve B adında iki kişi var. A B'ye art arda belirli bir kümeden seçilmiş mesajlar yollayacak. Mesela A çok işlek bir restoranda sırayla verilen her siparişi {"hamburger", "çizburger", "tavukburger", "salata"} kümesinden bir mesaj ile şirketin merkez ofisindeki B'ye bildirecek diyelim. İletişim hattı iki seviyeden birinde olabilen sinyaller (yani 0'lar ve 1'ler, İngilizce "ikili rakam" sözünün kısaltması olan teknik adıyla "bit"ler) taşıyabiliyor. Bu durumda A ve B'nin 0 ve 1'lerden oluşan hangi dizinin hangi mesaja denk geleceği konusunda baştan anlaşması gerekiyor. Niyetimiz uzun vadede mümkün olan en az sayıda bit göndererek bu işi gerçekleştirmek. Mesaj başına ortalama kaç bit göndermek yeterli olacaktır?

Eğer mesajlar hakkında bilgimiz bundan ibaretse, yapabileceğimiz en verimli şey tüm mesaj tiplerine ikişer bitten oluşan farklı "kod"lar atamak olur, mesela {"hamburger", "çizburger", "tavukburger", "salata"} mesajlarına bu sırayla {00, 01, 10, 11} kodlarının denk gelmesi kararlaştırıldıysa B

0111100100001110

gibi bir dizi okuduğunda, rahatça bunun hangi yemekleri temsil ettiğini çözebilecektir. Bu durumda mesela bin sipariş A'dan B'ye iki bin bitle gönderilir.

Ama diyelim ki farklı mesaj tiplerinin hangi olasılıklarla görüleceğine ilişkin bir ek bilgimiz var. Örneğin res-

toran zincirimizin bu şubesinin nüfusun yüzde 90'ının vejetaryen olduğu bir kasabada olduğunu biliyoruz. Bu durumda işlerin değişeceğini görüyor musunuz?

Eğer mesajların %90'ı "salata", kalanıysa diğer seçeneklerden olacaksa, "salata" mesajına tek bitlik bir kod (diyelim, sadece 0), diğer üçüne de art arda gönderildiklerinde neyin nerede bittiği hakkında belirsizlik yaratmayacak şekilde seçilmiş daha uzun kodlar (mesela 10, 110 ve 111) atarız. Toplam bit sayısı önceki senaryoya göre azalacaktır; tamam, eskiden iki bit kullanan bazı mesajlar için şimdi fazladan birer bit daha kullanıyoruz, ama o mesajların oranı az. Ezici çoğunluktaki "salata" mesajları içinse birer bit tasarruf ediyoruz, yani toplamda kârlı çıkacak, bin mesajı iki binden çok daha az sayıda bitle yollayacağız. Demek bu sipariş kümesi üzerindeki olasılık dağılımı öyleymiş ki siparişlerin ne olduğu hakkında B'nin en başta bulunduğu "bilgisizlik" seviyesinde okuduğu mesaj başına gerçekleşecek ortalama azalış iki bitten azmış. İşte bit birimiyle ölçülen bu "bilgisizlikteki azalış"a "enformasyon" diyordu Shannon.

İletişim mühendislerinin kutsal kitabı olan Shannon teorisi iyiydi, hoştu; ama "Bu nesnenin içerdiği enformasyon ne kadardır?" türünden soruları yanıtlayamıyor, hatta ifade bile edemiyordu. Yukarıdaki örnekte gönderilebilecek mesaj tiplerinin ne olduğu baştan iki tarafça da biliniyor, A'ya sadece bu sonlu kümenin hangi elemanının söz konusu olduğunu belirtmek kalıyordu. Oysa genelde bir bilgiyi iletişim kanalından yollarken veya belleğe kaydederken o bilgi parçasının "sıra numarası"nın değil, tüm detaylarıyla kendisinin temsil edilmesi gerekir.

Bazı nesnelerin diğerlerinden daha basit olduğuna, yani daha az bilgi içerdiğine ilişkin bir hissimiz vardır. Piyango 2847416705268 numaralı bilete çıkarsa mı daha çok şaşırırsınız, yoksa aynı sayıda rakamdan oluşan 1111111111111 numaralıya mı? Rakamlar birer birer eşit olasılıkla torbadan çekilirse iki dizinin de çıkma ihtimali eşit. Peki neden tamamı 1'lerden oluşan numaranın ikramiyeyi alması daha şaşırtıcı görünüyor?

Bu sorunun yanıtı 1960'lı yılların başında birbirlerinden habersiz üç kişi, Ray Solomonoff, Andrey Kolmogorov ve Gregory Chaitin tarafından keşfedildi. Üçü de herhangi bir cismin basit mi karmaşık mı olduğunun (onu başka herhangi bir cisimle karşılaştırabilmeye de el veren) en iyi ölçütünün, çalıştırıldığında çıktı olarak o cismin tam bir tarifini veren Turing makinelerinin en küçüğünün (yani programı en kısa olanın) boyu olduğunu gördüler.

Her cisim bir harf dizisiyle tarif edilebilir. Daha önce de gördüğümüz gibi, her cisim fiziksel parçacıklardan oluşmuştur ve çözünürlüğü istediğiniz kadar yüksek tutarak tüm bu parçacıkların konumlarını ve hızlarını sadece 0 ve 1'lerden oluşan bir alfabe kullanarak yazabilirsiniz. Ama aynı harf dizisini tarif etmenin birçok farklı yolu da olabilir. Sözgelimi bir milyar tane 0'dan oluşan diziyi ele alalım. Bu diziyi arkadaşıma bir milyar harften oluşan bir mektupla tarif edebilirim. Ama bu aptalca olur. Çünkü mealen "Bir milyar kere 0 yaz!" diyen bir bilgisayar programını bir milyardan çok daha az harfle yazıp arkadaşıma bu programı yollayarak da aynı bilgiyi iletebilirim. Bir kez hangi programlama dilini (yani evrensel Turing makinesini) kullanacağımız konusunda anlaşırız, sonra da birbirimize söylemek istediğimiz dizileri yazdıran kısa programları yollarız. İnternet faturamızdaki meblağı en aza indirmenin yolu budur!

Artık "algoritmik enformasyon kuramı" olarak bilinen bu yaklaşıma göre enformasyon "minimum tarif uzunluğu" olarak tanımlanır. Kısa bir şekilde tarif edilebilen bir cisim, sadece çok uzun bir tarifle inşa edilebilen bir cisimden daha az enformasyon içerir. Yukarıdaki piyango örneğinde "hepsi 1!" tarifi, diğer numarayı bir arkadaşınıza anlatmak için kullanacağınız en kısa metinden kısa gibi görünüyor, değil mi? Rasgele rakamlar çekilerek elde edilen bir dizinin çok büyük olasılıkla "sıkıştırılamaz", yani kendi rakam sayısından daha az harfle tarif edilemeyen bir sayı olacağı ispatlanmıştır. İşte bu yüzden bilet "hepsi 1" numarasına çıkarsa pirelenmekte haklısınız.

Farklı kökenlerine karşın, algoritmik enformasyon tanımıyla Shannon'ınkinin birbirleriyle tutarlı olduğunun gösterilmesi kuramcıları sevindirdi. Fizikçiler bir sistemin ısısı ve entropisiyle sistem hakkında sahip olunan enformasyon miktarı arasındaki bağlantıyı netleştirince enformasyonun evrenin işleyişindeki kendi başına bir kitap konusu olabilecek temel rolüne bir pencere daha açılmış oldu. Biz şimdi Turing'in bir başka mirasına eğilelim.

17 | Çarpma toplamadan zor mudur?

10. soruyu yanıtlarken kuramsal bilgisayar biliminde "problem" sözcüğüyle neyin kastedildiğini anlatmıştım: Aynı şablona uyan soruların oluşturduğu kümelere "problem" diyoruz. Örneğin iki sayının verilip toplamlarının istendiği tüm soruların kümesine "Toplama Problemi" denir. Böyle bir kümenin her elemanını girdi olarak alıp doğru yanıtı verebilen bir algoritma varsa ne âlâ. İlkokulda Toplama Problemi için bir algoritma öğrendiniz mesela. Algoritması olmayan, yani çözümsüz problemlerin var olduklarını da görmüştük.

Çözümsüz problemlerin sonsuz derecede zor oldukları kuşku götürmez. Peki algoritması olan her problem, aynı kolaylıkta veya zorlukta mıdır? Doğa çözülebilir problemleri kendi aralarında bu bağlamda sıralamış mıdır? Mesela toplama mı daha zordur yoksa çarpma mı?

Kuşkusuz "3 çarpı 4" işlemi göze "659476869 artı 7585842" işleminden çok daha kolay görünüyor; ama unutmayın, "problem" derken bu tekil soruları değil, sonsuz sayıda soru içeren kümeleri kastediyoruz.

İki algoritma arasında adil bir karşılaştırma yapabilmek için ikisine de aynı "boy"da sorular verildiğinde cevaplamak için ne kadar zaman harcadıklarına bakılır. Sözgelimi rasgele seçtiğiniz 10'ar rakamlı iki sayı alın ve bunları

bir kâğıtta ilkokul yöntemiyle toplayın, başka bir kâğıtta da çarpın. Aynı boydaki sorular için çarpma algoritmasının toplama algoritmasından daha uzun zaman aldığını gördünüz mü?

Ama bir detayı gözden kaçırmayalım. Bu örnek sadece ilkokulda öğrendiğimiz o çarpma yönteminin toplamanınkinden daha uzun zaman aldığını gösteriyor. Bu "Çarpma Problemi Toplama Probleminden zordur" demekle aynı şey değil. Ya ilkokuldakinden çok daha verimli, mesela iki sayıyı alt alta yazıp üçüncü satırda da çarpımlarını hızlıca hesaplamanıza el veren bir çarpma yöntemi varsa da biz bilmiyorsak? Kendi cehaletimizden ötürü Çarpma Problemini suçlamanın âlemi var mı?

Hikâyemiz 1960'da Moskova'da geçiyor: Dikkatli okurların adını önceki sorudan hatırlayacağı Prof. Kolmogorov içeri girdi. Salondakiler heyecanla kıpırdandılar. Sovyetler Birliği'nin en büyük matematikçisinin bu yeni seminer serisine katılabilmek büyük şanstı. Kolmogorov ilk seminerde ilkokul çarpmasından söz etmeye başlayınca biraz şaşırdılar ama.

Batı kaynaklarında kimi zaman es geçilse de, Andrey Kolmogorov'un diğer pek çok konunun yanı sıra hesaplama karmaşıklığı kuramının da öncülerinden olduğu kesindir. Kolmogorov binlerce yıldır daha hızlısının bulunamamış olmasından etkilenerek ilkokul yönteminin mümkün olan en hızlı çarpma algoritması olduğunu düşünüyor, ama bunu bir teorem olarak kanıtlayamıyordu. Seminerde de bu düşüncesini anlattı.

İzleyiciler arasındaki 23 yaşındaki matematik öğrencisi Anatoli Karatsuba tam bir hafta sonra, ikinci seminerinin ardından Kolmogorov'un yanına gidip, "Profesör" dedi, "daha hızlı bir çarpma algoritması keşfettim!"

Karatsuba çarpmasını her yıl algoritma analizi dersimde anlatırım. Tahmin edebileceğiniz gibi, ilkokul çarpmasına oranla ezberlemesi biraz daha zor bir yöntem. Ama bizim için önemli olan algoritmanın çetrefilliği değil.

Algoritmaların hızları şöyle karşılaştırılabilir: Yatay eksenin algoritmanın girdisinin uzunluğuna, dikey

eksenin de algoritmanın kaç adım attıktan sonra durduğuna, yani adım sayısı cinsinden çalışma zamanına karşılık geldiği bir koordinat sistemi düşünün. Şimdi size algoritmanız için bu altyapı üzerine bir grafik çizmenin yöntemini anlatacağım (Burada algoritmamızın hiçbir girdi için sonsuz döngüye girmediğini varsayıyoruz; mesela toplama ve çarpma algoritmalarınız bu garantiye sahip).

Verilen her uzunluk değeri için o uzunlukta sonlu sayıda farklı girdi olabilir. Mesela eğer girdilerimiz sadece 0 ve 1 harfleriyle yazılıyorsa 1 uzunluğunda sadece iki farklı girdi olabilir: "0" ve "1". 2 uzunluğunda dört farklı girdi olabilir: "00", "01", "10" ve "11". Girdi boyu arttıkça o boyda mümkün olan girdilerin sayısı da hızla artar.

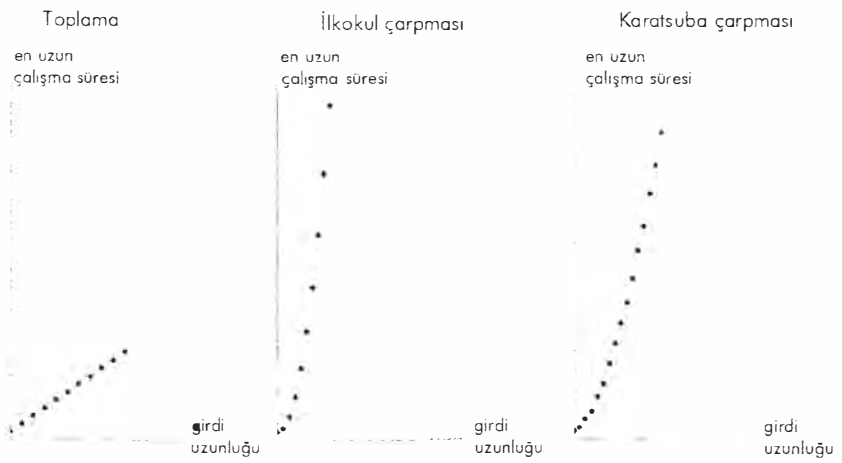
Grafiği şöyle çizeceksiniz: Her doğal sayı için algoritmanızı o boydaki her girdiyle çalıştırıp kaç adım sürdüğünü ölçeceksiniz. Yatay eksendeki her girdi uzunluğunu o boydaki girdilerden algoritmayı en uzun çalıştıranının neden olduğu çalışma süresinin dikey eksendeki yeriyle eşleyip oraya bir nokta konduracaksınız.

"Ama sonsuz sayıda olası girdi uzunluğu var, bu işi tamamlayamam!" mı dediniz? Tamam, haklısınız. Grafiği siz çizmeyin. Ama biz çizmesek de matematiksel anlamda her algoritmanın böyle "en kötü girdi"ler için çalışma süresini gösteren bir grafiğinin (teknik terimle, "çalışma süresi fonksiyonunun") var olduğu konusunda hemfikiriz, değil mi?

Kuşkusuz girdilerin uzunluğu arttıkça adım sayıları da artacaktır, ama bu artışı gösteren grafiğin şekli algoritmadan algoritmaya farklı olacaktır. Sözgelimi toplama algoritmamız için bu iş yapıldığında adım sayısının kabaca doğrusal olarak arttığını, yani grafiği sonsuza doğru uzattığımızda şeklinin soldan sağa yükselen bir doğruya (liseden hatırladığınız y=x doğrusuna) benzeyeceğini göreceğiz.

İlkokulda öğrendiğimiz çarpma algoritması için benzer bir grafikse uzatıldıkça doğrusal değil, $y=x^2$ gibi karesel olarak (türevi giderek artarak) yükselir. Bir algoritmanın girdisi uzadıkça adım sayısı ne kadar hızlı yükseliyorsa, bu o algoritmanın genelde o denli yavaş olduğu anlamına

gelir. Öyle ki toplamayı yavaş, çarpmayı ise çok daha hızlı işlemcileri olan iki farklı bilgisayarda yapsanız ve ikisini aynı anda başlatsanız bile, yeterince çok rakamlı sayılar için toplama işlemi çarpmadan daha önce bitecektir. A problemini çözen en hızlı algoritmanın adım sayısı grafiği yukarıdaki anlamda B problemininkinden daha hızlı yükseliyorsa A'nın B'den daha "zor" olduğunu (teknik terimle, "zaman karmaşıklığı"nın daha yüksek olduğunu) söyleriz. Elbette bu karşılaştırmaları adil şekilde yapabilmek için tüm algoritmaları aynı dilde yazmamız gerekir. Doğru tahmin ettiniz, burada yine algoritma kavramının resmi tanımı olarak TM'ler devreye girer. Karatsuba'nın algoritmasının çalışma süresi fonksiyonu $y=x^{1,59}$ gibi yükselir, ama binlerce yıldır kilitli duran bu kapının açılmasından sonraki yıllarda algoritmacılar boş durmadı. Art arda, giderek daha uzun sayıların çarpımında öncekilere üstün gelen yeni çarpma algoritmaları keşfedildi. Rekor şimdilik Martin Fürer'in 2007'de duyurduğu yöntemde. Fürer algoritması neredeyse verilen sayıların boyuna doğru orantılı sürede çalışıyor, ama yine de tastamam doğru orantıyla çalışan ilkokul tarzı toplama algoritmasının eline su dökemiyor. Acaba bir gün toplama kadar hızlı çarpma yapabilecek miyiz? Yoksa Çarpma Problemi gerçekten de toplamadan daha mı zor? Henüz kimse bilmiyor.



18 | Hesaplama karmaşıklığı nedir?

Hesaplama karmaşıklığı kuramı, Turing'in bahçesinin 1960'lı yıllarda açan çiçeklerindendir.

Geçtiğimiz sayfalarda gördüğümüz gibi, algoritma kavramının Turing sayesinde net bir tanım kazanması, eskiden beri kafaları kurcalayan kimi soruları artık matematikçilerce çalışılabilir hale getirdi. "Problemler arasında doğal bir zorluk hiyerarşisi var mıdır?" sorusu da bunlardan biriydi.

Çarpmayla toplamayı karşılaştırdığımız önceki soruda hesaplama karmaşıklığı kuramının "zorluk" tanımını gördük aslında: Dişe dokunur problemler sonsuz sayıda sorudan oluşur. Bir problemi çözen her algoritma için, soruların boyu uzadıkça onları yanıtlamanın daha çok iş gerektireceği açıktır; ne de olsa, uzun bir soruyu baştan sona okumak bile kısa bir soru için gerekenden daha fazla zaman alır. Bizi ilgilendiren, soruların boyları uzadıkça algoritmanın çalışırken harcayacağı kaynakların ne hızla artacağıdır.

Önceki soruda "zaman karmaşıklığı"nı, yani hesaplama sırasında atılacak adım sayısının soru boyuna göre artışını işledik. Zaman, hesaplama sırasında kullanılan kaynaklardan biridir. Ama hesaplama karmaşıklığı kuramının incelediği tek kaynak değildir.

"Bellek karmaşıklığı" benzer şekilde, girdinin boyu arttıkça algoritmanın hesaplama sırasında bir şeyler yazıp işaretlemek için kullanacağı bellek miktarının artışı cinsinden tanımlanır. Bu miktar problemi çözen Turing makinesinin çalışırken bu hesap için ayrılmış bir teyp alanından kaç kare kullandığı sayılarak ölçülebilir.

Kimi bilgisayarlar (örneğin beynimiz) bir değil, çok sayıda işlemcinin birbirleriyle eşzamanlı ("paralel") olarak işlemesiyle problem çözer. Kuşkusuz her şeyi taklit edebilen TM modeli böyle süreçleri de benzetimleyebilir, ama paralel hesaplama dalında çalışanlar sorunun boyu büyüdükçe onu hızla yanıtlayabilmek için işlemci sayısının

ne hızda artması gerektiğini şık bir şekilde ölçmelerine yarayan başka modeller de kullanır. Bir soruyu işbirliği yaparak yanıtlaması istenen birden fazla işlemcinin birbirlerine en az kaç bit yollamaları gerektiğini konu edinen "iletişim karmaşıklığı"nı da unutmayalım.

Sözün kısası, hesaplama karmaşıklığı kuramı problemleri onları çözen en verimli algoritmaların yukarıda sözü edilenler gibi kaynakları kullanma miktarlarının yanıtlanacak sorunun boyuna bağlı artış hızları açısından karşılaştırıp sınıflandırır. Bu artış çok hızlıysa, sözgelimi kaynak kullanımı y, girdi boyu da x'le gösterildiğinde $y=10^x$ fonksiyonundaki gibi üstel bir artış varsa, o zaman bu problem çok zordur, çünkü makul boydaki soruların bile yanıtlanması karşılayamayacağımız miktarda kaynak gerektirecektir: Girdi 100 harf (kısa bir tweet) boyunda bile olsa 10^{100} o kadar büyük bir sayıdır ki, her adımını fizik yasalarına göre mümkün olan en kısa sürede atan bir bilgisayar evrenin başlangıcından bu yana sürekli çalışmış olsa bile, o kadar adım atmış olamaz; bu nedenle üstel karmaşıklıkta bir problem, (Durma Probleminin aksine, onu "çözen" bir algoritması olsa da) pratikte en kısa birkaç sorusu dışındakilerin yanıtlanması imkânsız olan bir problemdir. Oysa y=x benzeri doğrusal karmaşıklıkta bir problem, örneğin toplama, milyonlarca rakam uzunluğundaki girdiler için bile bilgisayarın kullanıcısı mezara girmeden yanıtlanabilir. Bilgisayara herhangi bir işi yaptırmaya, hele de bir yapay zekâ inşa etmeye soyunacakların hedefleyecekleri problemlerin karmaşıklığını bilmeleri iyi olur tabii.

Bilgi işlem problemlerini sorunun boyu büyüdükçe cevabı bulmak için harcanması gereken minimum kaynak miktarının artma hızına göre sınıflandıracağız dedik. Aslında her farklı artış hızı için "filanca kaynak kullanımı taş çatlasa bu hızda artan problemler kümesi" diye bir "karmaşıklık sınıfı" tanımlayabiliriz; mesela çalışma süresi girdinin boyuna göre y=x fonksiyonu gibi artan problemler sınıfı, $y=x^2$ gibi artanların sınıfı, $y=x^3$ gibi artanlarınki, vs. Ama karmaşıklık kuramcıları bu kadar "yüksek

çözünürlüklü" bir sınıflandırma yapmak istemezler; hem sonsuz sayıda karmaşıklık sınıfı zorluk düzeyleri arasında daha temel kimi farkları gözden kaçırmamıza yol açabilir, hem de bir bilgisayar türünde kaynak kullanımı mesela $y=x^3$ gibi artan bir problem için başka bir bilgisayar mimarisinde $y=x^6$ gibi bir artış gerekebilir.

Farklı bilgisayar cinsleri üzerinde farklı kaynak kullanım artışlarından söz etmem aklınıza "ama bilgisayardan bilgisayara değişiyorsa o zaman bu kullanım artış hızı problemlere özgü bir özellik değil, problemleri sınıflandırmak için kullanılması saçma, kuramınız bir balonmuş!" itirazını getirebilir. Neyse ki Turing'in harika makinesi burada da imdadımıza yetişir.

Halihazırda kullandığımız bilgisayarlar ve (belki 20 ve 21. Soru'da değineceğimiz iki istisna dışında) şimdiye dek "bilgisayar" kavramını karşılayacağı iddia edilen ve fiziksel gerçekliğe dayanan her makine modeli, TM modelince hem zaman, hem de bellek karmaşıklığı açısından "makul" bir artışla taklit edilebilir. Yani yeni türden her makinenin yaptığı her işi, genel resmi değiştirmeyen "makul" bir yavaşlama ve dış bellek ihtiyacında "makul" bir artmayla bir Turing makinesine de yaptırabilirsiniz. Önceki cümlelerde "makul" derken, sözgelimi A tipi bir bilgisayar bir algoritmayı $y=x^2$ gibi artan bir kaynak kullanımıyla icra ediyorsa bir Turing makinesinin aynı işi $y=x^4$ gibi artan harcamayla yapabileceğini, öyle $y=10^x$ filan gibi fahiş bir artış gerektirmeyeceğini kastediyorum. Bu nedenle karmaşıklık sınıflarını aşağıda anlatacağım biraz daha düşük çözünürlükte tarif edersek problemler için makineden bağımsız bir zorluk sıralamasına varabiliriz.

(Değişik türlerden bilgisayarların birbirlerini benzetim sırasında çok da yavaşlama olmadan taklit edebilmesinin yapay zekâ projesi açısından önemini fark ettiniz, değil mi? Bilgisayarların insanların yapabildiği her bilişsel işi yapabildiğini söylediğimde, benzetimi yapan bilgisayarın benzetimi yapılandan daha fazla adım atma ihtimalini açık bırakmıştım. Elbette bir insan beyninin beş dakikalık çalışmasını beş yılda taklit eden bir makine

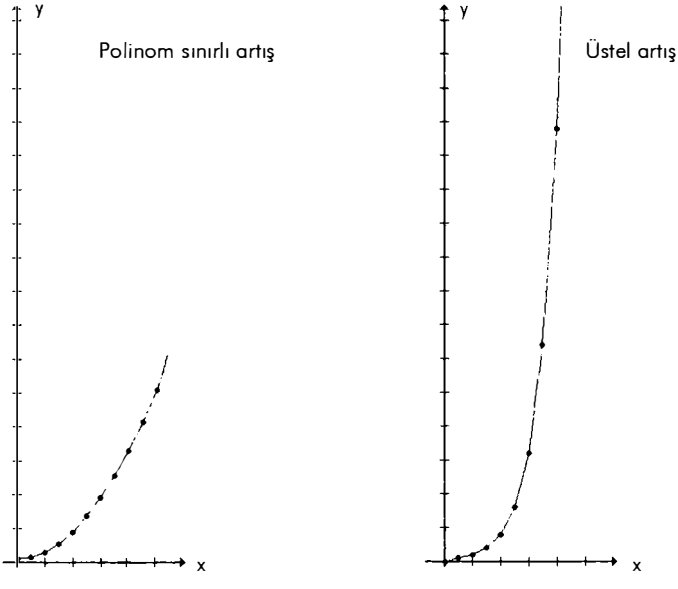
çok etkileyici olmaz. Neyse ki, hem karmaşıklık kuramı böyle benzetimlerde yavaşlamanın çok olmayacağı konusunda genel bir garanti veriyor, hem de zekâyı programlamak için daha iyi bir yol bulamaz da insan beynini taklide yeltenirsek en azından sinir hücreleri bazında bu benzetimi verimli şekilde yapabilecek bir teknoloji hayal ötesinde değil.)

Kuramımızda kullanılan en temel ayrım, "polinom sınırlı" olan ve olmayan kaynak kullanım artışları arasındakidir. Belirli bir problem için "polinom sınırlı zamanlı" bir çözüm yönteminiz varsa, şöyle bir K katsayısının da var olduğuna emin olabilirsiniz: Eğer soruların boylarını iki katına çıkarırsanız, çözüm için atılması gereken adımların sayısı taş çatlasa K katına çıkar (Biraz lise matematiğiyle $y=x$, $y=x^2$, $y=x^3$ gibi x'in üssü olarak sabit bir sayının yer aldığı her artış fonksiyonunun bu özelliğe sahip olduğunu gösterebilirsiniz). Yani sorular uzadıkça yanıtların gelme süresi de artar, ama "çılgınca" artmaz; artış hızının "makul" kabul edilen bir sınırı vardır. Karmaşıklık kuramcıları bu özelliğe sahip çözüm yöntemleri olan problemleri "kolay", olmayanları da "zor" kabul eder, çünkü eğer yönteminizin çalışma zamanı polinom sınırlı değilse çok kısa olanlar dışındaki soruları yanıtlamak yıllar, hatta asırlar alabilir, kimsenin de o kadar vakti yoktur (Mesela $y=10^x$ gibi x'in kendisinin üs olduğu üstel fonksiyonlar polinom sınırlı değildir; onların her biri için öyle bir Ü sayısı bulabilirsiniz ki, sorunun boyu bir harfçik arttığında kaynak kullanımı Ü kat artar.) Bu anlamda zaman tüketimi açısından kolay problemlerden oluşan kümeye "P" adı verilir. P, karmaşıklık sınıflarının en ünlüsüdür. Belli bir sayıda adım atan bir bilgisayar o sayıdan daha fazla bellek hücresini kullanamayacağı için, P sınıfındaki bir problemin sadece adım sayısı değil, bellek artışının da makul düzeyde olduğunu söyleriz. Bellek tüketimi polinom sınırlı artan problemler sınıfına ise "PSPACE" denir. Ve bu bizi karmaşıklık kuramının yıllardır cevaplanamamış sorularından birine getirir: Acaba P, PSPACE kümesine eşit midir?

Soruyu anladınız mı? P kümesindeki her problemin kendisini hem süreyi, hem de belleği makul miktarda kullanarak çözen bir algoritması var. PSPACE ise bellek kullanımı makul olan algoritmalarca çözülen problemlerin tümünün kümesi. P'deki her problemin PSPACE kümesinin de içinde olduğu (yani P'nin PSPACE kümesinin altkümesi olduğu) bu tanımlardan anlaşılıyor. Ama acaba bu kümeler eşit mi? Yani makul miktarda bellek kullanan her algoritmanın aynı problemi makul zamanda çözebilen "hızlı" bir sürümü var mıdır?

Sorun şu: Bazı problemler var ki, onları makul miktarda bellek kullanarak çözen algoritmaları onlarca yıldır biliyoruz. Ama bu algoritmaların tümünün çalışma süresi üstel olarak artıyor! Hiçbiri için hızlı bir sürüm bulamadık. İşin kötüsü, bu problemler için hızlı bir algoritmanın var olmadığını da ispatlayamadık. Matematik becerimiz henüz buna yetmiyor.

Karmaşıklık kuramı bunun gibi bir yığın cevap bekleyen soruyla dolu. En ünlüsünü önümüzdeki soruda göreceğiz.



19 "P = NP?" sorusu nedir, ne anlama gelir?

Karmaşıklık kuramcıları onlarca yıldır önceki sayfalarda tanıştığımız "kolay problemler kümesi" P'nin sınırlarını saptamaya çalışıyor. İlkokulda öğrendiğimiz yöntemlerin hızlılığı sayesinde o dört ünlü aritmetik işlem hesabının P'nin içinde olduğunu biliyoruz. Bize bir ülkenin karayolları trafik haritasındaki yol uzunluğu bilgileri verildiğinde, istenen iki şehir arasındaki en kısa yolu bulma problemi de P'de. Verilen iki sayının en küçük ortak katını bulma problemi de. Verilen bir sayının asal olup olmadığını saptama problemi de. Liste uzun, bu da güzel, çünkü pratikte makul uzunluklardaki sorularını kesinkes hatasız çözebileceğimiz problem aileleri sadece P'nin kapsadıkları.

10. Soru'da gördüğümüz anlamda "çözülebilir" olan, algoritmalarını bildiğimiz, ama P'nin dışında olduklarını da kanıtlamayı başarabildiğimiz problemler de var. Bunlara iki örnek vereceğim. Ne yapay, ne de doğal zekâ, bu problemlere ilişkin makul boydaki soruların tümünü (evrenin yaşı kadar zaman bile verseniz) yanıtlayamaz.

İlk örneğimiz, 10. Soru'da tanıştığımız Durma Probleminin "bütçeli" bir versiyonu: Ben size iki şey vereceğim; bir bilgisayar programının metni, bir de sayı (mesela 47.543.748). Siz de bu girdiyi inceleyip bana bu programın çalıştırılması halinde en fazla o sayı kadar (örneğimizde 47.543.748) adımda işini bitirerek durup durmayacağını söyleyeceksiniz.

Bu problemi çözen bir algoritma var elbet: Programların çalışmasını taklit etmeyi biliyoruz. Verilen programı adım adım (gerekirse kâğıt üstünde) ilerleterek, o kadar adımda durup durmayacağını saptayabiliriz. Örneğimizdeki sayı büyük ama bilgisayarlarımız da hızlı, o kadar adımlık bir benzetimi yapıp sonucu verebilir.

Ama bu benzetim algoritmasının çalışma süresi, girdisinin boyuna göre üstel olarak artmaktadır, yani makul boyda (diyelim 100 veya 200 harf uzunluğunda) çoğu girdi için

kabul edilemez derecede uzundur. Neden mi? Girdi olarak aldığımız, maksimum adım sayısını belirten sayıyı düşünün. Bu sayı yukarıdaki örneğimizdeki gibi 8 değil de 100 rakamlı olsaydı ne kadar devasa bir değeri olurdu, görebiliyor musunuz? Teknoloji ne kadar ilerlerse ilerlesin o kadar adımlı bir benzetimi hızla yapacak bir noktaya gelemeyiz. Dahası, bu problem için polinom sınırlı zamanda çalışan başka bir algoritmanın var olmadığı da kanıtlanmıştır.

İkinci örneğimizse "genelleştirilmiş satranç" denen bir oyunla ilgili. Yapay zekâ tarihine şanlı bir sayfa olarak geçen Kasparov-Deep Blue satranç karşılaşmasından 1. Soru'nun yanıtında söz etmiştim; 33. Soru'da da o zaferin sırrını anlatacağım, ama buradaki konumuz hiçbir zekânın baş edemeyeceği bir problem tanımlamak.

Sonlu sayıda elemanı olan problemlerin hesaplama kuramı açısından ilginç olmadığını 10. Soru'da anlatmıştık; işte o yüzden satranç oyununu genelleştirerek bir "satranç-lar ailesi" yaratacağız. Satranç oyunu 8 sıra ve 8 sütunlu bir tahtada oynanır. Genelleştirilmiş satrançta ise tahtanın boyutunu istediğiniz gibi seçebilirsiniz, yani dilediğiniz bir N sayısı için N satır ve N sütunlu bir tahtanız vardır. Oyuncuların hangi satranç taşından kaç tane kullanabileceği de N'ye göre değişir; N arttıkça daha çok piyonunuz, veziriniz, atınız vs. olur. Ama babadan kalma satrançtaki gibi herkesin tek şahı vardır. Taşların hareketleri ve oyunun kazanılmasıyla ilgili kurallar da standart satrançtakinin aynısıdır.

Şimdi problemimizi tanımlayabiliriz: Ben size N sayısını ve o boyuttaki bir tahtaya yerleştirilmiş taşların konumlarını vereceğim, siz de beyaz taşlarla oynayan oyuncunun bir "kazanma stratejisi"nin (yani bu pozisyondan onu, rakibi nasıl cevap verirse versin oyunu kazanmaya götüren hamle dizileri kümesinin) var olup olmadığını söyleyeceksiniz. Bu problem için de üstel artışa sahip bir algoritmamız var ve daha iyisinin olamayacağını da kanıtlayabiliyoruz.

Ama bir de zor mu kolay mı olduklarını bilemediğimiz ilginç bir grup problem var. Bu sınıflandırma sorusunun onlarca yıldır yanıtlanamamış olması, hesaplama karmaşıklığı kuramını halen kuramsal bilgisayar biliminin ve

hatta tüm matematiğin en heyecanlı konularından biri haline getirmiştir.

Diyelim ki size bir ülkedeki tüm şehirlerarası otobüs hatlarının fiyat bilgileri ve belli bir miktar para verildi. Otobüslerin uğradığı tüm şehirlerden geçen bir tur yapmaya bütçeniz yeter mi yetmez mi? Bu ünlü "Seyyar Satıcı Problemi"dir. Bu problemi kolayca çözebilmeyi kim istemez? (Seyyar satıcıların isteyeceği kesin!) Ne yazık ki, onca yıldır üzerinde çalışan nice parlak beyin, Seyyar Satıcı Probleminin P'nin içinde olup olmadığını bir türlü anlayamamıştır. Ne bu problem için polinom sınırlı zamanlı bir çözüm yöntemi bulunabilmiş, ne de böyle hızlı bir yöntemin var olmadığı gösterilebilmiştir. Ama Seyyar Satıcı Probleminin her zor problemde olmayan ilginç bir özelliği var: O, hakkında bir "süperzekâ" ile kandırılma korkusu yaşamadan konuşabileceğimiz problemlerden.

Evet, karmaşıklık kuramının kimi ünlü sonuçları, normal hesaplama güçlerine (yani girdi boyuna göre polinom sınırlı şekilde artan miktarda kaynak kullanma kabiliyetine) sahip bir varlığın, uçsuz bucaksız miktarda kaynak kullanarak bir anda sonuca varabilecek bir "tanrı" ya da, benim tercih ettiğim deyimle "süperzekâ" ile kandırılmadan konuşabilmesinin yöntemleri olarak yorumlanabilir. Diyelim ki ortada kendi imkânlarımızla kısa sürede yanıtlayamadığımız uzunlukta bir seyyar satıcı sorusu var. Karşımıza süper zekâlı olduğunu iddia eden bir yabancı çıkıyor ve cevabın "evet" olduğunu, yani sorudaki şehirler arasında gerçekten verilen bütçeyle gerçekleştirilebilecek bir otobüs turunun olduğunu ileri sürüyor. Acaba doğru mu söylüyor, yoksa bizi kandırmaya mı çalışıyor?

Yabancıya körü körüne inanmak zorunda değilsiniz! Ona hangi sırayla, hangi hatlardan giderek bu turun yapılabileceğini sorun. Eğer sizi kandırmaya çalışmıyorsa, yani gerçekten böyle bir tur varsa, size bu bilgiyi verebilmesi gerekir. İşin güzel tarafı şu: Bu verdiği bilgiye hile karıştırıp karıştırmadığını siz kendi imkânlarınızla (yani polinom sınırlı zamanda) kontrol edebilirsiniz; tek yapmanız gereken

hiçbir şehrin dışarıda bırakılıp bırakılmadığını ve toplam bilet fiyatının bütçeye uyup uymadığını hesaplamaktır. Eğer istenen özellikte bir tur yoksa muhatabınızın sizi kandırmak için uydurabileceği hiçbir sahte "tur" bilgisi bu testi geçemez; foyasını meydana çıkarabilirsiniz.

Bu türden, bir başkasınca çözüm olduğu iddia edilen kısa bir metnin sınanmasını kendimizin hızla yapabildiğimiz problemlerin kümesine, karmaşıklık kuramında "NP" adı verilir. Çözümünü kimsenin yardımına gerek duymadan kendimizin hızla bulabildiğimiz problemlerin kümesi olan P'nin içerdiği her problem zaten NP'nin de içindedir (Aynı şekilde, NP'nin içerdiği her şeyin de PSPACE kümesinde olduğu kanıtlanmıştır). Esas önemlisi, tıpkı seyyar satıcıdaki gibi gerçekten de hızlıca çözebilmeye can attığımız ama çalışma süresi o korkunç üstel hızda artmayan bir algoritmasını bir türlü bulamadığımız başka çok sayıda ilginç problemin de NP'nin elemanı olmasıdır: Verilen bir otobüs hatları haritasında birinden diğerine (gereksiz döngülere filan girmeden) giden en ucuz yolun bütçemizi aştığı iki şehrin olup olmadığını soran "En Uzun Yol" Probleminden tutun, kimin kiminle dost, kiminle düşman olduğu bilgisini içeren bir kişiler listesi ve bir D sayısı bize verildiğinde tüm üyeleri birbiriyle dost olan D kişilik bir grup var mı yok mu saptamamızı isteyen "Klik Problemi"ne varıncaya dek böyle yüzlerce problem bilinmektedir. Bence bunların en ilginci, Turing'in çözülemezliğini kanıtladığını gördüğümüz Karar Probleminin Gödel'ce 1956'da ortaya koyulan şu bütçeli sürümüdür: Verilen bir matematiksel önermenin, bu önermenin yanında verilen bir miktar kareli kâğıda sığabilecek sayıda harf içeren bir kanıtı var mıdır? Verilen sınırlı boyda bir kanıtın hatalı olup olmadığını kontrol etmek kolay bir iş olduğundan, bu problem NP'nin içindedir ve günün birinde onu yardımsız ve hızla çözen bir algoritma bulursak matematikçilere yapacak pek bir iş bırakmayacaktır.

21. yüzyılın başında Clay Matematik Enstitüsü, yeni asrın matematikçilerine meydan okuyarak 7 çözülememiş matematik bilmecesinin "başına ödül koydu". Cevabını

bulana enstitüce bir milyon ABD doları vaat edilen bu 7 sorudan biri de "P ile NP eşit midir?" sorusudur. Seyyar Satıcı Problemi (veya yukarıda verdiğim diğer örneklerden herhangi biri) için polinom sınırlı zamanda çalışan bir yöntemin var olmadığını ispatlayabilirseniz NP kümesinin P kümesinde olmayan bir eleman içerdiğini, yani eşit olmadıklarını göstermiş olursunuz, bir milyon da sizin olur. İşin ilginci, eğer Seyyar Satıcı Problemi için hızlı bir yöntem bulursanız da soruyu çözmüş olursunuz; çünkü 1970'lerden beri biliyoruz ki bu problem için hızlı bir çözüm yöntemi otomatik olarak NP'nin içindeki tüm diğer problemleri de hızla çözecek yöntemlere tercüme edilebiliyor (yani 10. Soru'da gördüğümüz deyimle NP'nin içindeki her problem Seyyar Satıcı Problemine hızla indirgenebiliyor), ve böyle bir durumda P ile NP'nin eşit olduğu sonucu çıkıyor.

Aradan geçen sürede Clay sorularından sadece biri çözüldü: Rus matematikçi Grigoriy Perelman 2003'te "Poincaré sanısı" denen topoloji sorusunu (ilk soruluşunun üzerinden 99 yıl geçtikten sonra) yanıtladı. Ama Perelman kimi diğer matematikçilere olan kırgınlığını gerekçe göstererek hem Clay Matematik Enstitüsü'nün vereceği bir milyonu, hem de bu başarısından ötürü kendisine verilmek istenen diğer ödülleri reddedip evine kapandı. Matematiği bıraktığı söyleniyor. "P ile NP eşit midir?" sorusu hâlâ yanıt bekliyor, ama gördüğünüz gibi bu bir milyon dolar kazanmak uğruna girişilemeyecek kadar zor bir iş. Ayrı bir aşk istiyor.

20 | Zar atmak işe yarar mı?

18. Soru'nun yanıtında değişik bilgisayar türlerinin birbirlerini çok da zorlanmadan taklit edebildiklerinden, o nedenle de hesaplama karmaşıklığı kuramı açısından aralarında

çok da önemli farklar olmadığından söz etmiş, bu genel kabule istisna olabilecek iki örneği daha sonra tartışacağımızı söylemiştim. Şimdi bu olası istisnalardan ilkinin sırası geldi. Turing makinesi modeli "belirlenimci"dir, yani belli bir TM'ye belli bir girdi sunulduğunda tüm davranışı kesinlikle belirlenmiş olur; aynı işlem kaç kez yapılırsa yapılsın tıpatıp aynı sonuç çıkar; sapmalara, hatalara yer yoktur. Gerçek dünya pek böyle değildir! Turing'den sonraki yıllarda modelin gerçeği daha iyi yansıtan sürümleri tanımlandı.

Daha önce tanımladığımız temel TM modeline bir zar veya bozuk para atışının sonuçlarına göre kontrol akışını belirleyebilme hakkı verelim (Ya da masaüstü bilgisayarımıza dilediğinde 0 ve 1 rakamlarından birini tümüyle rasgele şekilde seçebilme yeteneği verebildiğimizi varsayalım). Bu özellik makinenin icra edebileceği komut türlerine "1/2 olasılıkla 5 numaralı komuta, 1/2 olasılıkla da 9 numaralı komuta atla" gibisinden bir "olasılıksal dallanma" komutu eklenerek hayata geçirilebilir. Bu yeni makine türüne "Olasılıksal TM" (OTM) denir.

TM'lerin aksine OTM'ler belirlenimci olmadıklarından her soruya her zaman aynı yanıtı vermeyebilirler. Elbette biz her zaman doğru yanıtı isteriz, ama hata olasılığı daima sıfır olan bir OTM'nin hiçbir işi standart bir TM'den daha hızlı yapamayacağı kolayca kanıtlanabilir. Az da olsa bir miktar, diyelim trilyonda bir hata olasılığını sineye çekersek OTM'lerin kimi işleri (örneğin verilen bir sayının asal olup olmadığını saptama işini) o hesap için bilinen en iyi TM'den daha hızlı yaptığını gösterebiliriz. Dahası, birkaç problem var ki, P sınıfının "doğanın kolayca çözmemize izin verdiği problemler kümesi" olduğu iddiasına gölge düşürür.

Bunların en ünlüsü Polinom Özdeşlik Testi Problemidir: Ben size değişken isimleri, toplama, çıkarma ve çarpmalarla kurulmuş, mesela

"(a+b)×(a-b)×(c×a-b×c)"

ve

"a×a×a×c-a×b×b×c-a×a×b×c+b×b×b×c"

gibi iki ifade veririm; sizin de bu iki ifadenin değişkenlere hangi sayılar atanırsa atansın daima aynı değeri veren fonksiyonlar olup olmadığını söylemeniz beklenir. Rasgele sayılar atan zekice bir algoritmayla son derece düşük hata olasılığıyla polinom sınırlı zamanda çözebildiğimiz bu problem, besbelli ki doğanın kolayca çözmemize izin verdiklerindendir, ama şimdiye kadar hatasız ve polinom sınırlı süreli bir TM'si bulunamamıştır.

OTM'ler gerçekten TM'lerden üstün müdür? Henüz bilmiyoruz, ama uzmanların çoğunluğu bir gün her problem için az hata yapan bir OTM'ninkine denk hızda bir TM bulunduğunun gösterilebileceğine inanıyor.

Fakat zar atma ve azıcık hata yapmanın hatasız hesaplamaya üstün olduğunun kesinlikle kanıtlandığı durumlar da vardır: Bir veya birkaç süperzekâ ile iş tutmanız gereken zamanlar mesela. Bu konuda öğrencilerimle yaptığımız iki buluştan söz edeceğim.

Bilgisayarımızın kullanabileceği bellek miktarını sabitleyerek başlayacağız. Yani sorunun boyu ne kadar uzun olursa olsun, makinemiz hep aynı miktarda bellek kullanmak zorunda olacak. Çalışma süresi önceki örneklerimizdeki gibi soru boyuna göre artacak tabii.

Bir dizi havayolunun uçuş bilgilerini içeren bir listeye bakarak adı verilen iki kentten birinden diğerine bir veya daha çok uçakla gitmenin mümkün olup olmadığını saptama işine Rota Bulma Problemi diyelim. Yukarıda anlattığım türden kısıtlı bellekli bilgisayarların bu problemi çözemeyecekleri, bir süperzekâ ile temas halinde olsalar bile onun bu konuyla ilgili iddialarını 19. Soru'nun yanıtında anlattığımız şekilde kendi başlarına denetleyemeyecekleri onlarca yıllardan beri bilinmektedir.

Peki yukarıdaki senaryodaki bilgisayara bir de belli sayıda zar atıp sonraki adımını ona göre belirleyebilme hakkı verilirse? Öğrencim Abuzer Yakaryılmaz'la birlikte gösterdik ki, bu durumda bilgisayar bir süperzekâ yardımıyla (ve onun tarafından kandırılmadan) hem Rota Bulma Problemi, hem de onunla aynı zorluktaki tüm problemler için ileri sürülen yanıtları kendi imkânlarıyla denetleyebilme

gücüne erişiyor. Yani az sayıda zar atma ve sınırlı (yüzde 50'den az) bir olasılıkla hata yapabilme hakkı bu düzeneğin hesaplama gücünü kesinlikle artırıyor. Birkaç yıl sonra da öğrencim Gökalp Demirci'yle bu kez bir değil, zıt iddiaları savunan iki süperzekâ ile muhatap olan bir bilgisayarın az miktarda zar atarak, gücünü Rota Bulma Probleminden daha zor olduğuna inanılan birçok problem çözümünü denetleyebilecek düzeye çıkarabileceğini kanıtladık.

Rasgele süreçlerin hesaplamada işimize yaraması ilginç. Ama sonraki soruda değineceğimiz gibi, bilgisayar teknolojisinde asıl devrimi bize kuantum fiziği armağan edecek gibi görünüyor.

21 | Kuantum bilgisayarı nedir, ne işe yarar?

20. yüzyılın başlarında fizikte bir devrim yaşandı. Artık doğanın çok ama çok tuhaf kuralları olduğunu biliyoruz. Evren temel parçacıklar (mesela elektronlar) aynı anda iki ayrı yerde, iki farklı durumda olabiliyormuş gibi davranıyor sözgelimi. Dahası, aynı parçacığın bu farklı kopyaları sağduyumuza çok ters gelen bir şekilde etkileşip bazen birbirlerini silebiliyor. İnanması zor, ama kuşku bırakmayacak kesinlikte saptanmış bir doğa yasası bu.

Turing makinesi modeli ve günümüzde öğretilen yerleşik programlama yaklaşımları, Newton fiziğine göre işleyen bir dünyayı esas almaktadır. Acaba kuantum fiziğinin bu yeni ilginç özelliklerinden daha hızlı hesaplama yapmak için yararlanılabilir mi? Malum, iki farklı duruma girebilen her fiziksel değişkeni bir bitlik enformasyonu tutmak için kullanabiliriz. Ama bir elektronun nerede olduğuna göre 0 veya 1 olarak yorumlanması halinde "klasik" bellek birimlerimizin aksine aynı anda hem 0, hem de 1 içerebilen bir "kuantum biti" elde etmiş oluruz.

Aşılması güç kimi teknik engeller söz konusu olsa da çoğu uzman, büyük ölçekli "kuantum bilgisayarları"nın günün birinde inşa edilebileceğini düşünüyor (Ben bu satırları yazarken henüz belleği bu yeni türden topu topu 50 bit içeren bir makine yeni inşa edilmişti). Henüz ortada olmamalarına karşın bu bilgisayarlara özgü algoritmaların nasıl kurulabileceğini düşünebilmemiz için ise yine bir matematiksel modele, yani TM'lerin kuantum fiziğine uygun bir sürümüne ihtiyacımız var.

Kuantum Turing Makinesi (KTM) modeli bu amaçla tanımlanmıştır. Standart TM tanımında tüm bileşenlerin (teybin ve kontrol biriminin) şu aynı anda birden fazla durumda olabilen parçacıklardan inşa edildiğini düşünün. Bu makine artık aynı anda birçok paralel koldan işlem yapabilir. Fizik yasalarının getirdiği kısıtlara uyarak komut repertuvarımıza biraz geçen soruda OTM'ler için sözünü ettiğimizi andıran "3/5 genlikle 7 numaralı komuta, -4/5 genlikle de 10 numaralı komuta atla" türünden "kuantum dallanma" komutları eklememiz gerekir ("Genlik", kabaca "olasılığın karekökü" olarak düşünebileceğimiz ve söz konusu paralel işlem kollarının ağırlıklarını belirleyen bir kuantum kavramıdır). KTM'lerin de OTM'ler gibi az olasılıkla yanlış cevabı vermelerine göz yumalım.

Genliklerin eksi değerler de alabilmesi nedeniyle dikkatle düzenlenmiş bir KTM hesaplaması sırasında kimi paralel kollar birbirlerini "götürebilir" ve yanlış sonuca giden yoldaki kolların birbirini götürmesi sağlanarak yüksek olasılıkla doğru sonucu veren algoritmalar yazılabilir. Bu yöntemin klasik programlamada dengi yoktur. Elbette bu yenilik, akla iki soru getirmiştir:

1) Kuantum bilgisayarları klasik Turing makinelerinin hiç çözemeyeceği, yani 10. Soru'nun yanıtında "çözülemez" olarak tanımladığımız problemleri çözebilir mi?

2) Kuantum bilgisayarları kimi problemleri klasik makinelerin asla taklit edemeyeceği bir hızda çözebilir mi?

Birinci sorunun yanıtının "hayır" olduğundan eminiz: Klasik bir bilgisayarla (ya da çok sabırlı bir insansanız sadece kâğıt ve kalemle) bir kuantum algoritmasının

adımlarını taklit edip sonuçta kaç olasılıkla, hangi yanıtı vereceğini bulabilirsiniz. Turing makinesinin evrensellik özelliğine hiçbir istisna düşünemiyoruz.

Ama ikinci soru çok heyecanlı. Bu hesaplama kuramının en yakıcı sorularından biri. Şimdiye dek ileri sürülen tüm diğer fizikle tutarlı hesaplama modellerinin aksine, kuantum bilgisayarlarını klasik bilgisayarlara makul bir yavaşlamayla taklit ettirmenin bir yolunu bulabileceğimizden pek umutlu değiliz. Yukarıda sözünü ettiğim kâğıt/kalem hesabını yapmaya kalktığımızda, benzetimi yapılan kuantum algoritmasının adım sayısı arttıkça bizim harcadığımız emeğin üstel hızda arttığını görüyoruz. Daha verimli bir yöntem var mı, bilmiyoruz.

Dahası, elimizde bilinen hiçbir TM'nin hızla çözemediği ama KTM'lerin polinom sınırlı zamanda çözebildiği çok önemli bir problem var: Verilen bir tamsayıyı çarpanlarına ayırma işini tarif eden Çarpan Bulma Problemi (22.665.463 sayısı hangi iki asal sayının çarpımıdır, söyleyin bakalım?). İnternet üzerinden şifreli iletişim protokollerinin çoğu bu problemin hızla çözülemeyeceği varsayımına dayandığından, bu algoritmanın bulunmasından sonra kuantum bilgisayarlarının inşası öncelikli bir proje haline gelmiştir. Hatta kim bilir, belki de birisi çoktan büyük ölçekli bir kuantum bilgisayarını inşa etmiştir, ama habersizce başkalarının iletişimini dinlemek daha çok işine geldiğinden bunu ilan etmiyordur?

Bir yanlış anlamayı önlemek gerek: Günün birinde büyük çaplı kuantum bilgisayarları imal edebilirsek şimdi çözemediğimiz için hayıflandığımız bütün problemleri çözebilir hale gelmeyeceğiz. KTM'lerin az hatayla polinom sınırlı zamanda çözebileceği hiçbir problemin PSPACE'in dışında olmadığı, yani klasik bilgisayarların sadece üstel hızda artan bellek kullanımıyla çözebilecekleri bilinen kimi zor problemlere kuantumun da gücünün yetmeyeceği kanıtlanmıştır. Dahası, 19. Soru'nun yanıtında sözünü ettiğimiz milyon dolarlık zor NP problemlerinin hiçbiri için verimli kuantum algoritmaları bulamadık, var olduklarını da pek sanmıyoruz. Aslına bakarsanız,

Çarpan Bulma Probleminin klasik bilgisayarlarca hızla çözülemediğinden bile emin değiliz, kim bilir, belki hızlı bir algoritma vardır da biz bulamıyoruzdur?

Acaba kuantum bilgisayarının klasik bilgisayara üstün geldiğini kesinlikle kanıtlayabileceğimiz senaryolar var mıdır? Geçen soruda da sözünü ettiğim doktora öğrencim Abuzer Yakaryılmaz ile bu soru üzerinde uzun yıllar çalıştık ve bir dizi örnek bulduk. Günümüzde sadece çok küçük kuantum bilgisayarları inşa edilebildiği için iyice anlamlı olan "sabit bellek" kısıtı konulduğunda, sadece iki bitçik kuantum belleği eklenmiş bilgisayarların bile bu ekten yoksun olanların asla yapamayacağı bazı işleri başarabildiğini gösterdik.

Peki, madem kuantumun klasik bilgisayarları geride bırakabileceğini düşünüyoruz, bu yapay zekâ projesi için ne anlama geliyor? Eğer insan beyni kuantum fiziğine özgü süreçlerden yararlanıyorsa önceki sayfalardaki "zaten bir TM ile insan beynini hızla taklit edebiliriz" iddiamız suya düşmez mi? Bu konuya döneceğiz.

3. Bölüm
YAPAY ZEKÂNIN DOĞUŞU

22 | Yapay zekâ nedir?

Bu soruya cevap yazmaya başlamadan önce eski yazılarımı bir karıştırdım. 20. yüzyılda şöyle yazmışım:

> En hırslı yorumuyla yapay zekâ, insanlık tarihinin en büyük mühendislik projesidir. İnşa etmek istediğimiz şey, sonuçta bir bilgisayar programından, yani formel bir dilde yazılmış bir metinden ibarettir, ama bu metin o denli uzun ve (herhalde) karmaşık olacaktır ki, yazılması hemen aklınıza gelebilecek diğer dev mühendislik projelerinden daha çok adam-yıl alırsa şaşmamak gerekir. Bu programı çalıştırdığımızda, insanlarca yapıldığında 'zekice' bulduğumuz **her şeyi**, en zeki insanın düzeyinde (veya daha da üstün şekilde) yapabilecektir.[3]

Şimdiki kafamla okuduğumda görüyorum ki bu tanım eksik. Sorun, eski YZ araştırmacılarının insan zekâsının

3) Cem Say, "Akla Doğru", *Cogito*, Sayı 13, 1998.

kocaman bir bölümünü gözden kaçırmış, ya da en azından çok hafife almış olması. Aradan geçen yıllarda aklımız başımıza geldi.

Satrancı şampiyon seviyesinde oynayan, matematik problemlerini veya Sudoku gibi birçok kombinasyondan doğru olanını bulmayı gerektiren bulmacaları hızlıca çözebilen kişilere "zeki" derdik o zamanlar. Yapay zekâyı da bunları bilgisayarlara yaptırmaktan ibaret sanıyorduk. Boğaziçi Üniversitesi'nde hocalığa başladığımda, öğrencilerime proje olarak TRT'de yıllardan beri süren ve yarışmacıların verilen harflerle en uzun kelimeyi ve verilen sayılarla bir hedef sayıya ulaşan işlemler dizisini bulması istenen "Bir Kelime, Bir İşlem" yarışmasını kazanabilecek, veya Türkçe sözlükten yararlanarak otomatik olarak istenen boyutta kare bulmacalar hazırlayabilecek "dar alanda yüksek IQ'lu" bilgisayar programları yazdırıyordum. 1997'de insan satranç şampiyonunu yenen bilgisayar tarih yazdı, anlatmıştım. Ama ufak bir problem vardı...

Geliştirdiğimiz YZ sistemleri bazı konularda en zeki insanları bile geçiyorlardı, ama birçok temel konuda da üç yaşındaki çocukların, hatta hayvanların başarısına erişemiyorlardı! "Görme" (ışık parlaklığı ve rengine ilişkin sinyalleri "bu görüntüde bir kuş var mı?" gibisinden sorulara yanıt verebilecek şekilde işleme) veya bir ses sinyali şeklinde gelen konuşmaları metne dökme gibi çok temel becerileri henüz bilgisayara kazandıramamıştık. Programlarımız dijital ortamda verilen bir metni "anlama" gerektiren görevleri sadece çok kısıtlı çerçevelerde yapabiliyordu. Rasgele bir gazete haberini okuyup konusuyla ilgili birkaç basit soruyu yanıtlayabilen bir program bile büyük işti. Sözün kısası, yapay zekâ tanımımız zekâ denen buzdağının suyun altındaki kısmını, her insanın sahip olduğu o altyapıyı kale almadığı için sakat doğmuştu. Görmek, devrilmeden yürümek, duyduğunu anlamak; bunlar mühendislik açısından satranç şampiyonu olmaktan çok daha zor işlerdi! Tam bir zekâ elde etmek için evrim denen kör mühendisin milyonlarca yılda kafa göz

yararak geliştirdiği tüm bu temel altsistemleri de kurmayı öğrenmemiz gerekiyordu. Son yıllarda YZ'nin heyecan yaratan gelişmesinin büyük kısmı, bunu başarmaya başlamış olmamızdan ibaret.

Peki yapay zekâ nedir? "Doğal sistemlerin yapabildiği (zekice olsun veya olmasın) her bilişsel etkinliği (gerekirse bedenleri olan) yapay sistemlere, daha da yüksek başarım düzeylerinde nasıl yaptırabileceğimizi inceleyen bilim dalıdır" diyeyim. Bakalım 20 yıl sonra bu tanımın da güncellenmesi gerekecek mi?

23 | Turing testi nedir?

Felsefe dergisi *Mind*'ın ("*Zihin*"), 1 Ekim 1950 tarihli 236. sayısında Alan Turing'in "Hesaplama Makineleri ve Zekâ" başlıklı bir makalesi yayımlandı. 1930'larda kuramsal sınırlarını keşfettiği, 1940'larda da bizzat yapımlarına katkı verdiği elektronik bilgisayarların potansiyelini gören Turing, bu makaleyi insanlığı yeni çağa hazırlamak için yazmıştı. Makalenin "Taklit Oyunu" başlıklı ilk bölümü "'Makineler düşünebilir mi?' sorusunu ele almayı öneriyorum" cümlesiyle başlıyordu.

Turing sadece bilgisayarların tüm insani bilişsel faaliyetleri taklit edebileceğini görmekle kalmamış, insanların buna karşın bilgisayarların düşündüğünü kabul etmekte zorlanacağını da öngörmüştü. Düşünen bir makine yapmayı başardığımı iddia etsem bana hemen inanır mısınız? Sizi nasıl ikna edebilirim? Makinem ağzıyla kuş tutsa (ya da bunun bilişsel dengi olan marifet her neyse onu başarsa), günler, haftalar boyunca her tür deneyden alnının akıyla çıksa bile, yine de yaptığı şey için "düşünme" kelimesini kullanmamakta ısrar edecek kişiler biliyorum. Oysa aynı kişiler sokakta beş saniyeliğine gördükleri, hiç tanımadıkları yabancı bir insan için

rahatça "düşünüyor" diyebilirler. Makinelerle insanlar arasında temel bir fark olduğuna dair inanış kafaları bulandırabiliyor. Turing çareyi bu farkı görünmez kılmakta bulmuştu.

Turing şu oyunda başarılı olabilen bir makinenin düşündüğünü kabul etmemizi öneriyordu: "Sorgucu" adını verdiğimiz bir insan, yazılı mesajlaşmaya izin veren bir sistemle A ve B adında iki oyuncu ile yazışmaktadır. A ve B'den birisi bir kadın, diğeri ise bir erkektir. Erkek oyuncu sorgucuyu diğer oyuncunun değil, kendisinin kadın olduğuna ikna etmeye çalışır. Rakibi olan kadın da (haklı olarak) kadın olanın kendisi olduğunu savunacaktır. Belirli bir süre sonunda sorgucu oyunculardan hangisinin gerçekten kadın olduğu kanaatine vardığını açıklar. Oyun defalarca oynanır. Bu senaryoda erkek oyuncunun yerine aynı oyunu oynamaya (dişi bir insan taklidi yapmaya) programlanmış bir bilgisayar koyduğumuzda sorgucunun başarı oranı artmazsa bilgisayarın "düşündüğü" sonucuna varmamız gerekir.

Turing testi budur: Dış görünüşten etkilenmememiz için saf zekâyı yalnız bırakan bir ortamda insanla makineyi yarıştırır (İlginç şekilde, günümüzde robotları giderek daha başarılı şekilde insana benzetebiliyoruz, ama ben de Turing gibi işin özünün bu olmadığı kanısındayım). Her konu konuşulabilir ve bilgisayar tümünde insan düzeyinde performans göstermelidir. Turing makalesinde, saç şeklinden edebiyat tartışmalarına uzanan örnekler vermiştir. Bu kadar geniş bir yelpazede, hem de kendisi de zeki bir insan olan sorgucuyu kandırabilmek için zekâ, tüm o soruları yanıtlamak için de düşünmek gerekir! Eğer bu ölçütü kabul etmiyorsanız hattın öbür ucundaki varlığın düşündüğüne ikna olmanız için daha ne yapalım?

Turing testi çok yüksek bir çıtadır. Henüz doğal dili ve içinde yaşadığımız dünyayı insanlar kadar iyi anlayan bir bilgisayar yapamadık ve (şov için yapılan birkaç dakikalık "test"lere sokulan kimi lafazan programları saymazsak) daha Turing testini geçebilen bir makine ortada yok. Ama güzel bir hedef, değil mi?

24 | 'Yapay zekâ' adını kim koydu?

New Hampshire eyaletinin Hanover kasabasındaki Dartmouth Koleji, ABD'nin birinci sınıf üniversitelerinin en küçüğüdür. Kolej 1955'te John McCarthy adında genç bir matematikçiyi yardımcı doçent olarak kadrosuna aldı. McCarthy ve onun gibi Turing'in izinden giden birkaç başka dahi, kafa kafaya verip düşünen bilgisayarlar üretmek için nasıl bir hareket tarzı gerektiğini konuştular. Önce ülkede bu konuyla ilgilenen herkesi bir araya toplamaya karar verdiler. Bu beyin fırtınası için Rockefeller Vakfı'ndan (günümüz standartlarında gülünç derecede düşük bir miktar olan) 7500 Dolar talep edeceklerdi.

McCarthy'nin 16. Soru'nun yanıtında tanıştığımız Claude Shannon ve Ray Solomonoff, 0 ve 1'lerden daha yüksek düzeydeki (insanların bilgisayar programları yazabilmelerini çok kolaylaştıran) ilk programlama dilinin mucidi Nathaniel Rochester ve önümüzdeki soruda yakından tanıyacağımız Marvin Minsky'yle birlikte kaleme alıp 2 Eylül 1955'te vakfa sundukları resmi başvuru yazısı, "yapay zekâ" lafının ilk kez geçtiği metin olarak bilinir:

"1956 yazında Hanover, New Hampshire'daki Dartmouth Koleji'nde yapay zekâ üzerine 2 ay süreyle 10 kişilik bir çalışma yapılmasını öneriyoruz. Bu çalışmada, öğrenmenin ve zekânın başka herhangi bir vasfının tüm yönlerinin prensipte bir makine tarafından benzetimi yapılabilecek kadar net şekilde tarif edilebileceği kabulü esas alınacaktır. Makinelerin nasıl dili kullanır, soyutlamalar ve kavramlar oluşturabilir, şimdi insanlara özgü kabul edilen problem türlerini çözebilir ve kendilerini geliştirebilir hale getirilebileceğini bulmaya teşebbüs edilecektir. Dikkatli seçilmiş bir bilimadamları grubu bir yaz boyunca birlikte çalışırsa bu problemlerin biri veya daha çoğunda önemli bir ilerlemenin kaydedilebileceğini düşünüyoruz."

Öngörüleri günümüzden bakıldığında bütçe ve takvim açısından olağanüstü iyimser görünse de, kimilerinin

insanlığın sonunu getireceğinden korktuğu, kimilerininse umut bağladığı dev yapay zekâ projesinin temeli bu toplantıda atıldı.

25 | 'Yapay zekâ mevsimleri' nedir?

Dartmouth Çalıştayı ile adlı adınca başlayan yapay zekâ projesinin ilk yıllarında iyimserlik tavandaydı. O dönemin öncüleri hem elektronik bilgisayarın doğuşuna tanıklık etmiş, hem de Turing'in hesaplanabilirlik kuramını, yani bilgisayarın "yapılması mümkün olan her şeyi yapabilen makine" olduğuna dair buluşunu anlamış zeki insanlardı (Hesaplama karmaşıklığı kuramı ise henüz doğmamıştı; az sonra göreceğimiz sorunun kaynaklarından biri de buydu). Bir devrim yaşandığını, insanlığın düşünen makineler yapmasına az kaldığını hissediyorlardı. Hedefin boyu o kadar büyüktü ki, onu gördüler, yola çıktılar, ama ona ulaşmak için kat etmek gereken mesafeyi tahminde yanıldılar.

Yanlış anlaşılmasın, bu kurucuların tümü büyük adamlardı. Yapay zekânın isim babası McCarthy işin altyapısını oluşturmakta en çok katkısı olanlardandı. Sayısız YZ programının yazılmasında kullanılan bir bilgisayar dili olan Lisp, onun eseriydi. Taşkın zekâsıyla bir yandan "problem çözen makinelerin çoğunun inançları olduğunun" söylenebileceğine ilişkin felsefe makaleleri yazıyor, öte yandan herkesin ayrı bir bilgisayar sistemine sahip olması yerine hesaplamaların tıpkı su veya elektrik gibi ihtiyaç bazında alınıp satıldığı bir iş modelini (günümüzün "bulutta hesaplama" fikrinden yarım asır önce) ortaya atıyordu. Kuşaklar dolusu öğrenci yetiştiren McCarthy, zekâ için gereken dünya bilgisinin Frege ve Russell'ın sunduğu mantık diliyle temsil edildiği, bir yargıdan diğerine mantıksal çıkarımlarla ilerleyen programlara zekice işler

yaptırılabileceğini görmüştü. Her düşünce işini bir tür teorem ispatı olarak gören bu "mantıkçı" yaklaşımı esas alan birçok proje yürüttü.

Dartmouth Çalıştayı'nın bir diğer düzenleyicisi olan Marvin Minsky, daha sonra evleneceği Gloria Rudisch'le ilk yemeğe çıkışlarında ona, "Bir gün insan beyninin nasıl çalıştığının sırrını çözeceğim" demişti. Gloria yıllar sonra "O anda 'Bu adam ya deli ya da dahi' diye düşünmüştüm. Neyse ki ikincisiymiş" diyecekti. Minsky yapay zekâ ve insan zihnini inceleyen "bilişsel bilim" dalının devlerinden olmasının yanı sıra matematikçi, yepyeni bir mikroskop cinsinin mucidi ve yetenekli bir piyanistti. Massachusetts Teknoloji Enstitüsü'nde öğrencilerine o zamana dek meydanı boş bulanların bol keseden "Makineler şu işi yapamaz!" diye atıp tuttuğu işleri yapan bilgisayar programları yazdırmaya başladı: Kısıtlı bir dilde yazılmış, belli konulardaki matematik problemlerini çözen programlar, bir masanın üzerindeki tahta blokları gören, onları belli koşulları sağlayacak şekilde hareket ettirmek için gerekli planları yapabilen ve bu konuda oldukça karmaşık İngilizce komutları anlayabilen programlar.

Dikkat ettiyseniz, yukarıdaki örneklerin tümü "mikro dünyalar" denilen, çerçeveleri çok iyi çizilmiş, kısıtlı kullanım alanlarında çalışıyordu. Ama o zamana dek bir makinenin böyle bir şey yaptığını görmemiş olan insanların aklını başından alabiliyorlardı. Russell ve Whitehead'in *Principia Mathematica*'sındaki bir teorem için kitaptakinden daha kısa bir ispat bulmayı başaran "Mantık Kuramcısı" programının yazarlarından (daha sonra ekonomi Nobel'i sahibi olacak olan) Herbert Simon felsefedeki zihin-beden problemini çözdüklerini ve makinelerin artık düşünebildiğini söylüyor, sonra da hızını alamayıp 1968'e dek bir bilgisayarın satranç şampiyonu olacağını ileri sürüyordu (Böyle tahminler yapmaya kalkarsam lütfen beni durdurun). Soğuk Savaş sırasında Sovyetler'e karşı bir avantaj sağlamak için yanıp tutuşan ABD Savunma Bakanlığı kesenin ağzını açtı. Yapay zekâ araştırmalarına para akmaya başladı. Fakat bu bahar aslında sonbahardı.

Düşmanın dili Rusçadan İngilizceye otomatik çeviri yapacak sistemlerin geliştirilmesi hayaline ve uzmanların giderek daha yüksekten uçan vaatlerine kapılan Amerikan devleti, yıllar boyu çuvalla para verdiği projelerin "Gözden ırak olan gönülden de ırak olur" tipi cümlelere "Körler gönülsüz olur" gibi feci karşılıklar üreten programlardan iyisini ortaya koyamadığını görünce musluğu kapattı. Çeviri yapmanın bu projelerde öngörüldüğü gibi sadece iki gramer arasında dönüşümler ve kelimelerin sözlükteki karşılıklarıyla değiştirilmesiyle becerilebilecek kadar basit bir iş olmadığı, tercümanın metnin konusunu oluşturan "dünya" hakkında bilgi sahibi olması gerektiği biraz pahalıya anlaşılmıştı.

Bir diğer sorun "kombinasyonlar patlaması" idi. "Oyuncak" boydaki soruları yanıtlamak için alelacele geliştirilen programların çoğu cevabı bulmak için kabaca "bütün ihtimalleri dene, en iyisini seç" diye özetlenebilecek, "akıl"dan ziyade bilgisayarın hızından yararlanan arama algoritmaları kullanıyordu. Bu algoritmaların zaman karmaşıklığı üstel, hatta bazen daha da beter hızda artan fonksiyonlara denk geliyordu. Sözgelimi çözümü bulmak için girdiden bir sayı alıp o adette farklı parçanın art arda doğru sırada dizilişini arayan bir algoritma olabilecek bütün sıralamaları deniyor diyelim. Bu durumda verilen sayı S ise olası sıralamalar S'nin faktöriyeli (liseden hatırladığınız yazılışıyla "S!") tanedir. Bu çok ama çok hızlı artan bir fonksiyondur: 2'nin faktöriyeli 2'dir, 3'ünki 6'dır, ama

4! = 24,

14! = 87.178.291.200,

24! = 620.448.401.733.239.439.360.000,

34! = 295.232.799.039.604.140.847.618.609.643.520.000.000...

Sanırım mesele anlaşılmıştır. Küçük örneklerde çalışan bu tür algoritmalar iş biraz ciddiye binince havlu atıyordu.

İlk yıllardaki iyimserlik, yerini "yapay zekâ kışı" denen umutsuzluk ve "fonsuzluk" dönemlerinin ilkine bıraktı.

Bu arada zeki makinelerde insan beynindeki sinir hücrelerinin yapılanmalarından esinlenilmesini öneren "bağ-

lantıcılar" ile mimarisi nasıl olursa olsun bir bilgisayarın her bilişsel işi yapabileceğini, bu yüzden beyni taklide gerek olmadığını, yüksek seviyede kavramları temsil eden simgelerin işlenmesinin yeteceğini söyleyen "eski moda YZ'ciler" arasında Minsky'nin de karıştığı bir çekişme gereksiz zaman kaybına yol açtı. Minsky'nin liseden arkadaşı olan Frank Rosenblatt, sinir hücrelerini model alan sistemlerin öğrenme konusunda "simgeci" yaklaşımı geride bırakacağını savunan bağlantıcıların en önde geleniydi. Minsky yemedi içmedi, matematikçi Seymour Papert ile birlikte Rosenblatt'ın sinir hücresi benzeri "perceptron" modelini eleştiren bir kitap yazdı. 1969'da yayımlanan kitapta bu hücrelerin belli bir şekilde dizildiklerinde kimi basit fonksiyonları öğrenemeyeceklerine dair (diğer dizilimler hakkında hiçbir şey söylemeyen) bir kanıttan başka dişe dokunur bir şey yoktu, ama yazanların otoritesi o kadar göz kamaştırıcıydı ki sinir ağları konusundaki araştırmalara ayrılan kaynaklar bir anda kurudu. Minsky'yi hiç affetmeyen Rosenblatt 43. doğum gününde bir tekne kazasında öldü ve 88 yıl yaşayan Minsky'nin aksine sinir ağlarının 21. yüzyılın başındaki müthiş başarılarını göremedi.

Japonya Uluslararası Ticaret ve Sanayi Bakanlığı'nın 1981'de insanlarla konuşabilecek, resimleri yorumlayabilecek ve çeviri yapabilecek programlar üretilmesini hedefleyen Beşinci Kuşak Bilgisayar Projesi'ne 850 milyon ABD doları kaynak ayırdığını açıklamasıyla mevsim yine değişti. Japon ekonomik mucizesini kopyalamak isteyen Batı ülkeleri yine yapay zekâya yatırıma başladı.

O zamanın en çarpıcı YZ ürünü, belli bir alandaki çok miktarda bilginin insan uzmanlardan elde edilen bilgiyle kodlanıp genellikle Lisp makineleriyle mantık yürütülerek o alandaki soruları cevaplamak için kullanılması esasına dayanan "uzman sistemler"di. Şirketlerarası bir prestij konusu haline gelen uzman sistemler pazarı büyüdü, şişti, simgesel YZ programlarının ezeli sorunu olan "kırılganlık" (başlangıçta çizilen çerçevenin az da olsa dışındaki girdilerin başarının biraz azalmasına değil, tümüyle çökmesine yol açması) duvarına çarptı. Japon projesinin

amaçlarının on yıl içinde gerçekleştirilemeyeceği anlaşıldı, uzman sistem ve Lisp makinesi satan şirketler iflasa sürüklendi. Kış yine gelmişti.

1990'da Irak komşusu Kuveyt'i işgal etti. En çok katkıyı ABD'nin yaptığı uluslararası bir ittifak 1991'de Irak'a saldırarak işgali sona erdirdi. Bunun bizim konumuzla ilgisi ne mi?

1989'da yapılan bir tatbikatta lojistik sistemlerinde büyük verimsizlikler olduğunu saptayan ABD ordusu, aceleyle bu sorunları çözecek bir YZ programı ısmarlamıştı. Savaştan az önce devreye alınan DART ("Dinamik Analiz ve Yeniden Planlama Aracı") adındaki program Avrupa'dan apar topar Suudi Arabistan'a sayısız kuvvetini taşımak zorunda kalan ABD ordusunun teçhizat ve personel nakil planlarına kısa zamanda askeri yetkilileri hayran bırakan iyileştirmeler hesapladı. 1995'e gelindiğinde ABD Savunma Bakanlığı sadece DART'ın sağladığı tasarrufla önceki 30 yıl boyunca çoğu duvara toslayan tüm o YZ projeleri için verdiği parayı çıkarıp kâra geçmişti!

1997'de Deep Blue Kasparov'u yenip Simon'ın öngörüsünü 30 yıl gecikmeyle de olsa gerçekleştirdi. Acımasız mühendislerin iteklemelerine aldırmadan işine bakan sabırlı insansı robot videoları insanlarda bu sefer de "Fazla mı ileri gittik ne?" düşüncesini uyandırmaya başladı. Fakat son YZ kışından çıkışın esas sebebi bilgisayarların sürekli hızlanıp ucuzlayarak birkaç paralı kurumdaki soğutulmuş merkezlerden çıkıp dünyanın her köşesinde evlere ve insanların ceplerinde sokaklara yayılması oldu. Dünyayı saran ağ, inanılması güç şekilde insanlara harika hizmetleri bedavaya sunan Google gibi şirketlere, bu hizmetleri kullanan insanların bedavaya ağa yükledikleri bilgiler de çalışmak için dev veri kümeleri gereksinen yapay öğrenme sistemlerinin nihayet kanatlanmasına yol açtı. Hayranlık verici görüntü tanıma, yol tarifi, tıbbi destek, müşteriye göre ürün önerme ve (eskisine göre çok iyi) doğal dil işleme programları geliştirildi. Kendi kendini süren otomobiller yollara çıktı. Sadece oyunun hamle kurallarının bilgisiyle başlayıp goyu ve satrancı kendi kendine

oynayarak öğrenen ve 24 saat içinde insanüstü seviyeye gelen bir program yazıldı.

Henüz tam ölçekteki Turing testini geçemedik. Mevcut YZ baharı ne kadar sürer, bilemiyoruz. Bir kuramcı olarak bence ilginç olan, son yıllardaki büyük atılımların ardındaki "derin öğrenme" gibi yapay öğrenme tekniklerinin nasıl çalıştığını tam olarak bilmiyor olmamız. Kimi durumlarda harika sonuç veren yöntemlerin özünü anlamak için henüz Turing'in hesaplanabilirliğin genel sınırları için daha ortada bilgisayarlarımız yokken kurduğuna benzer bir kuramsal temelden yoksunuz. Macera devam ediyor.

4. Bölüm
YAPAY ZEKÂ NELER YAPAR, NASIL ÇALIŞIR?

26 | 'Eski moda yapay zekâ' nedir, nasıl çalışır?

"Eski moda yapay zekâ" ("good old-fashioned artificial intelligence") filozof John Haugeland'in 1985'te yazdığı bir kitapta yapay zekâ projesinin ilk 30 yılına damgasını vuran ve "başarısız olduğu" düşünülen "simgeci" yaklaşıma taktığı addır.

Dartmouth Çalıştayı'yla yola koyulan ekip, biraz da kendilerini o sıralarda popüler olan "sibernetik" yaklaşımı gibi başka alanlardan farklılaştırmak kaygısıyla, bilgi gösterimi için matematiksel mantığın dili ve çıkarım mekanizmasının yeterli olduğunu, zekânın bir simge işleme sürecinden başka bir şey olmadığını, her bilişsel işlevin bellekteki simgelerin uygun bir hesaplama zinciri sonucu başlangıç diziliminden sonuç dizilimine dönüştürülmesi ile modellenip bu seviyede yazılmış programlarca gerçeklenebileceğini savunuyordu. Öncülerin bu tavrı o dönemin

yazılımlarının çoğunun ortak bir bilgi gösterim ve programlama tarzına sahip olmasına yol açtı.

Hâlâ en sevdiğim programlama dili olan ve bilgisayar meraklılarına sadece zevk için bile olsa öğrenmelerini tavsiye etmekten yorulmadığım Prolog, bu stilin en güzel örneğidir. Bir konuda Prolog programı yazmak, o konudaki bilginizi mantıksal önermeler şeklinde ifade etmekten ibarettir. Örneğin

baba('Ali', 'Jale').

satırı, "Ali Jale'nin babasıdır" demek için,

amca(A, C):-baba(B, C), birader(A, B).

satırı da, "Eğer B C'nin babası, A da B'nin biraderiyse A C'nin amcasıdır" tanımını ifade etmek için kullanılabilir. Ebeveynlik ve evlilik bilgileriyle diğer hısım-akrabalık ilişkilerinin tanımlarının yukarıdaki gibi belirtilmesinden sonra artık istenen iki kişinin birbirinin tam olarak nesi olduğuna, genlerinin kaçta kaçını paylaştığına veya (gerekli yasal bilgiler de aynı şekilde kodlanırsa) kim ölürse servetinin ne kadarının kime kalacağına dair bir dizi ilginç sorunun yanıtlanması mümkündür. Bu yanıtları hesaplamak için karmaşık sayısal hesaplar yapmak, türevler veya integrallerle falan uğraşmak gerekmediğini, sadece kimi simge dizilerinin birbirlerine uygun şekilde zincirlenmesinin yeteceğini görüyor musunuz? Simgesel akıl yürütme budur. Hiç programlama bilmeyen bir kişi bile iki saat içinde yukarıdakiler gibi soruları yanıtlayan bir program yazacak kadar Prolog öğrenebilir.

Her eski moda YZ programı için buna benzer bir öykü anlatılabilir. Robotumuzun bir evden çıkış yolunu arama yeteneğine sahip olması için (yine simgeler arası ilişkilerle kodlanmış) hangi odalar arasında kapılar olduğu bilgisi üzerinde başarı ölçütünü ("dış kapı"yı bulmak) sağlayana dek yollar denemeyi benzetimleyen işlemler yürüten bir program yazabiliriz. "Gri kediler tırmalamaz" ve "Prenses gri bir kedidir" cümlelerinden "Prenses tırmalamaz" cümlesinin elde edilmesini dikte eden şablonları kodlamak da çok zor değildir (Uzman sistemler, belli konularda böyle

bilgi parçaları üzerinde uzun mantık zincirleri kurarak yeni sonuçlara varan programlardır). Simgelerinizi bir sonraki dizilime dönüştürürken ne yapılması gerektiği konusunda birden fazla seçenek varsa ve sonuca götürenin hangisi olduğunu bilmiyorsanız, sistematik olarak tüm seçenekleri deneyen "arama" algoritmaları kullanabilirsiniz, ama "arama uzayı"nız çok büyükse bu sizi geçen soruda sözünü ettiğimiz kombinasyonlar patlaması faciasıyla karşı karşıya bırakabilir. Arama uzayının bir noktasının diğerlerine oranla çözüme daha yakın olduğunu kestirmekte kullanılabilecek ek bilgileriniz varsa (bunun satranç için bir örneğine 33. Soru'da değineceğiz) iş biraz kolaylaşabilir.

Önümüzdeki sayfalarda bu yaklaşımla çözülen birçok ilginç problem göreceğiz. Buradaki amacım simgeci yaklaşımın farkının ne olduğunu anlatmak.

Her fiziksel sistem nihayetinde simgeler işleyen bir mekanizma olan Turing makinesi tarafından taklit edilebildiğinden bazı okurlar simgeciliğin zaten bilgisayarcılığın özünde olduğunu, yapay zekâ için başka bir alternatif olmadığını düşünüyor olabilir. Prolog'da da, diğer ünlü programlama dillerinde olduğu gibi, bir "TM taklit programı" yazılabilir. Ama dikkat edin, örneklerimizdeki simgeler fiziksel zerreler seviyesinde değil, aile üyeleri, evdeki odalar veya cümledeki kelimeler gibi çok daha yukarılarda, "düşünce birimleri"ni temsil ettikleri yüksek bir düzeydeydiler. Eski moda YZ'ciler, işin bu düzeyde çalışılarak halledilebileceğine inanmışlardı. Minsky'nin temel birimin simgeler değil de "simge altı" bir düzeydeki hücreler olduğu sinir ağları fikrine alerjisinin nedeni buydu.

Ama bu ısrar sıkıntıları beraberinde getirdi: Tanıdığınız bir baba-kız çifti hakkında kafanızda yukarıda "Ali Jale'nin babasıdır" demek için yazdığım Prolog örneğinde temsil edilene benzer bir bilgiyi tutuyorsunuz, ama bilgisayarım Ali ve Jale'nin kim, hatta ne oldukları hakkında başka hiçbir şey bilmezken siz o kişilerin yüzlerini, seslerini ve daha bir yığın şeylerini de biliyorsunuz.

Dış dünyadaki o karmaşık varlıklarla kafanızdaki o simgeler arasında bir ilişki var. Besbelli ki onları tanıma süreciniz sırasında duyu organlarınızdan gelen bol miktarda enformasyon işlenerek beyninizde o kişilerin adlarını da içeren bir bilgi ağı oluşmuş. Birini bugün görseniz bu görüntü o ağda bir etkinleşme yaratıp adının "aklınıza gelmesi"ne yol açacak, belki de iş dil kaslarınıza ve ses tellerinize o adı telaffuz etmek için gerekli sinir sinyallerinin yollanmasına varacak. Yani kafadaki simgeler hiç de o düzeyde simgeler olmayan başka şeylerin de rol aldığı bir süreç olmadan anlamlandırılamıyor.

Eski moda YZ'nin sıkı sıkıya bağlandığı "mantık yürütme" kavramı kesin, hatasız, istisnasız bir muhakeme yöntemini çağrıştırıyordu. Kısa zamanda gerçek hayatta işlerin öyle olmadığı fark edildiği için daha esnek olacağı umulan mantık gösterimleri ve emin olunmayan bilgilerle çalışmak için olasılık hesaplarına dayalı algoritmalar da devreye girdi. Ama inşa edilen sistemlerin çoğu biz insanların her gün karşılaşıp baş edebildiğimiz "bozuk girdi"leri (yanlış telaffuz edilmiş sözcükleri, grameri bozuk cümleleri, kötü elyazılarını vs.) başarımımız biraz azalsa da yine de anlayıp işleyebilme yeteneğimizin yanına bile yaklaşamıyor, böyle durumlarda tıkanıp kalıyorlardı.

1956'nın genç devrimcileri 1980'lerin tutucu ihtiyarlarına dönüşmüşlerdi. İnsan düzeyini her alanda geçen sistemleri kısa zamanda ortaya koyacaklarını defalarca iddia edip tekrar tekrar yalancı çıkmaları, elde ettikleri gerçek başarıları da gölgelemişti. "Yapay zekâ" neredeyse gülünç bir deyim olarak algılanır olmuştu. Simgesel yöntemin hiç şansının olmadığı örüntü tanıma görevlerini büyük başarıyla gerçekleştiren, bozuk girdilere doğal şekilde dayanıklı, uzmanlarının da insanın tüm yönleriyle geçilmesi iddiasını hiç dile getirmediği sinir ağları gibi yeni yapay öğrenme yöntemlerinin yükselmesiyle ilk perde kapandı, yeni bir dönem başladı.

27 | Evrimsel programlama nedir?

Evrim gerçeğinin keşfi, sadece yaşam bilimlerini tutarlı bir omurgaya sahip kılmakla kalmadı. Doğanın bedenler ve zihinleri geliştirirken kullandığı "tasarım" yönteminin bu olduğunun anlaşılması, mühendisleri de etkiledi. Bilgisayar programlarının da evrimin temel "daha uygun olanın hayatta kalması" yaklaşımıyla geliştirilebileceği fikri, "evrimsel programlama" adı verilen yaklaşımı doğurdu.

Diyelim ki belirli bir problemi çözen bir program geliştirmek istiyorsunuz, ama bu çözümün tam olarak nasıl yapılacağını kendiniz de bilmediğinizden oturup o programı yazabilecek durumda değilsiniz. Ama en azından, size zaten yazılmış bir program verildiğinde, onu çalıştırıp başarımını ölçerek o problemi ne oranda çözebildiğini otomatik olarak puanlayan bir "sınama" yazılımı ortaya koyabiliyorsunuz. İşte bu koşullar sağlanırsa, bilgisayarınızın belleğinin içinde probleminizi çözen programların evrilmesine elveren bir süreci başlatabilirsiniz.

İlk iş, programlama dilinizin komutlarının rasgele art arda dizilmesiyle oluşmuş çok sayıda "atmasyon" program üretin. Bunlara "birinci kuşak programlar" deyin. Elbette ki bunlardan herhangi birinin sizin probleminizi çözen program olma ihtimali yok denecek derecede küçük olacaktır. Ama henüz ilk kuşaktayız! Şimdi bu kuşaktaki programların her birini yukarıda sözünü ettiğim başarım ölçme yazılımıyla sınayın. Rasgele üretildiklerinden başarımlarının çok düşük olmasını bekleriz elbet, ama yine de birbirlerinden farklı olduklarından belki birkaç tanesi diğerlerinden daha yüksek puan alacaktır sınama yazılımınızdan. İşte hedefinize daha yakın olan bu programlar "hayatta kalacak" ve bir sonraki kuşağı oluşturacaklar.

Başarım ölçümü düşük çıkan programları öldürün (yani silin), sonraki kuşakta onlara yer yok. Daha yüksek başarımlı programların tümünü bir sonraki kuşağa

aktarın. Dahası, onları kendi aralarında "çiftleştirip" çocuklarını elde edin (örneğin bir programın üst yarısını bir başkasının alt yarısına yapıştırarak böyle bir "melez" çocuk program üretilebilir) ve bu çocukları da yeni kuşağa ekleyin. Teker teker bu "daha güçlü" programlar üzerinde küçük değişiklikler ("mutasyonlar") yaparak da yeni bireyler elde edebilirsiniz. Ve şimdi de bu kuşaktaki programları çalıştırıp sınayın, düşük puanlıları eleyip yüksek puanlıların soylarını sürdürün. Bakın, maksimum başarım puanının kuşaktan kuşağa azalmayacağı garantili bir döngüye girdiniz!

Tüm bu süreç otomatiğe bağlanabiliyor ve birçok kuşak sonra gerçekten yüksek başarımlı programlar elde edilebiliyor. Çözümü nihayet başarabilen o son programı kim yazdı? Hiç kimse. Evrildi o.

28 | Sinir ağları nasıl çalışır?

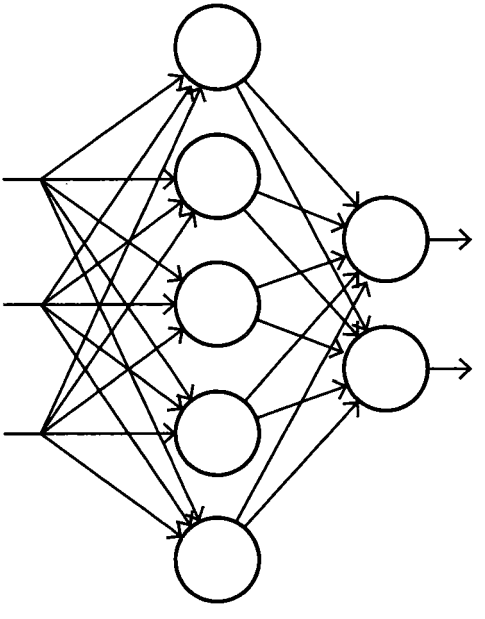
Sinir hücrelerinin yaptığı "girdi tellerinden gelen etkinleşme sinyallerinin ağırlıklı toplamını hesaplayıp sonuç yeterince büyükse etkinleş" görevini becerebilen yüz milyar birimi birbirlerine beyindeki gibi bağlayıp aralarındaki bağlantı ağırlıklarını da doğadakine benzer şekilde düzenler, bu ağı da girdi ve çıktı cihazlarına iliştirirsek, düşünen bir sistem elde edebiliriz, değil mi? Öyleyse neden yapay zekâ için bu yöntemi kullanmıyoruz?

Çok zor da ondan! 14. Soru'da gördüğümüz gibi, insan beyninin çalışma şeması biyolojik evrim yoluyla, bizim onu kolay anlayabilmemiz konusunda hiçbir özen gösterilmeden oluştuğundan tam bir arapsaçıdır. Onu çözmeye çalışan sinirbilimci arkadaşlarımı hep çok takdir etmişimdir. Günümüzün gözde yapay öğrenme algoritmalarının temelindeki yapay sinir ağları, bu karmaşıklığın çoğunu göz ardı eden, iç yapısı çok daha

düzenli sistemlerdir ve onları kendimiz tasarlamış olmamıza rağmen, haklarında hâlâ yanıtını bilmediğimiz bir yığın soru vardır.

Simgecilerle bağlantıcıların uzun süre "anlaşamamış" olmasının sebebi belki de budur: Simgeci yaklaşım "bilgisayar gibi", kural ve mantık tabanlı düşünmeyi esas alır. Bu tür bilişimin kuramı daha ilk elektronik bilgisayar imal edilmeden önce Turing'ce ortaya konulmuştu. Simgeciler nereye bastıklarını bilerek adım atabilir. Bağlantıcılar ise, daha çalışma ilkeleri hesaplama kuramı netliğinde anlaşılamamış olan "beyin gibi" düşünmeyi temel alırlar; tarzları simgecilere oranla çok daha deneyseldir, bu stil farkı makalelere, konferanslara, biliminsanlarının birbirleriyle konuştuğu dile yansır. Ne çare, yapay zekâ hedefine ulaşmak için iki tarafı da anlamak ve karşılıklı eksikliklerini tamamlayacak şekilde birleştirmek gerekmektedir.

Yapay sinir ağları art arda dizilmiş katmanlar şeklinde tasarlanır. Her katmanın çıktısı, sırada kendisini izleyen katmanın girdisidir. İlk katman girdi katmanı, sonuncusu da çıktı katmanıdır.



İlki hariç her katman bir dizi yapay sinir hücresinden oluşur. Her hücrenin kendine ait birçok girdi ve (eğer sayıları birden çoksa tümü aynı sinyali taşıyan) çıktı telleri vardır ve her katmandaki her hücre bir önceki katmandan gelen her sinyali girdi olarak alır (Bu düzenekleri fiziksel olarak kurduğumuzu sanmayın, bu şemayı bilgisayarımızın belleğinde taklit ediyoruz. Az sonra göreceğimiz öğrenme algoritması da bu sanal ortamda çalışacak).

İlk katman ("girdi katmanı") doğrudan ağımızın girdisini oluşturan bir sayı dizisidir. Sözgelimi girdi olarak bir resmi alıp çıktısında o resimde Ahmet amcamla Sabriye halamın olup olmadığını belirtmesini istediğim bir sinir ağının girdi katmanının her elemanı o resmi oluşturan noktacıklardan (bunlara "pikseller" denir) birinin rengini kodlayan bir sayı olabilir. Çıktı katmanındaki hücrelerin çıktı tellerinden de girdi katmanına verdiğimiz sorunun cevabını okumayı bekleriz, mesela bu örnekte çıktı katmanı biri amcama, biri de halama tekabül eden iki hücre içerir ve bu hücrelerin her birinin çıktısının söz konusu akrabam resimde varsa 1, yoksa 0 olmasını isterim.

Sinir ağındaki her hücreye girdi veren her telin bir de "ağırlığı" (yani o tele özgü bir sayı) vardır. Her sinir hücresi, içinde olduğu katmanın sırası geldiğinde, küçük bir hesap yapar: Önce her girdi telinin bir önceki katmandan taşıdığı sayıyı o telin ağırlığı ile çarpar, sonra bu çarpımları toplar, son olarak da bu toplamın büyüklüğüne göre basit bir formüle göre belirlediği bir sayıyı çıktı tellerinden bir sonraki katmana yollar. Böylece girdi katmanına aile fotoğrafımı ifade eden sayıları girmemi takiben adım adım her katman bir sonrakine sinyaller yollar ve sonuçta çıktı katmanından iki sayı okurum.

Sinir ağı böyle çalışır. Peki ama amcamla halamı gerçekten tanıyabilir mi?

Her şeyin tellerin ağırlıklarına bağlı olduğunu görüyor musunuz? Hangi girdiye karşılık olarak hangi çıktının hesaplanacağı tüm o sayıların kaç olduğuna bağlıdır. Deneyimden biliyoruz ki bizim beynimizdeki ağ şemasında bu

gibi işleri beceren ağırlık kümeleri mevcuttur. 1980'lerde ispatlanan harika bir teoreme göre, bu sadece üç katmanlı (ve ortadaki katmanı yeterince kalabalık) olan sinir ağları için de geçerlidir! Amcanızı tanımaktan belli boydaki Türkçe cümleleri Çinceye çevirmeye kadar normal bir insanın aklına gelebilecek her "dönüşüm"ü böyle bir ağa hesaplatabilecek ağırlıkların mevcut olduğundan eminiz. Ama mevcut olduklarını bilmekle onların değerlerinin kaç olduğunu bilmek farklı şeyler tabii. Ağıma amcamla halamı tanıtacak ağırlıklar hangileridir? Tıpkı simgesel bir programlama diliyle bir görüntü dosyasında amcamı tanıyacak bir program yazmayı bilemediğim gibi sinir ağındaki binlerce tele bu işi yapmak için atamak gereken ağırlıklarının değerlerini de bilemiyorum. Bu sayıları kim atayacak, yani ağı kim programlayacak?

İşte yapay öğrenmenin güzel tarafı budur: Adım adım yapılışını programlayacak kadar iyi bilmediğimiz, ama doğru yapıldığında yanlışından ayırmayı başarabildiğimiz girdi-çıktı dönüşümlerini makinenin **kendisi** öğrenir! Çünkü bol miktarda örnek girdi-çıktı çiftini girdi olarak alıp bu dönüşümü gerçekleştirebilen bir sinir ağının tel ağırlıklarını (işler yolunda giderse) hesaplayabilen bir "gözetimli öğrenme" algoritmamız var.

Geliştirilmesine katkı veren bilimadamları arasında (tesadüfe bakın ki) George Boole'un torununun torunu olan Geoffrey Hinton'ın da bulunduğu "hatanın geri yayılımı" algoritması şöyle çalışır:

Amacımız ağa hangi girdi için hangi çıktıyı üretmesi gerektiğini öğretmektir. Bunu çok sayıda girdi-çıktı çiftini ağa "göstererek" yapacağız. Başlangıçta tel ağırlıklarını bilmiyoruz, o yüzden onlara rasgele değerler atayarak rasgele bir ağ elde edelim.

Ağımızı eğitmek için dijital aile albümümü kullanacağım diyelim; çok sayıda resim dosyası hazırladım, dahası, her dosyayı içinde Ahmet amcamla Sabriye halamın olup olmadığını gösteren iki bitle "etiket"ledim: İkisi de varsa etiketin değeri 11, amcam var ama halam yoksa 10, ikisi de yoksa 00, vs.

Şimdi ilk fotoğrafı ağa gösterelim, yani onu oluşturan noktaların sayısal değerlerini ağın girdi katmanına verelim. Elbette ki mevcut ağırlıklarıyla amcamı doğru tanımasını beklemiyoruz; muhakkak hata yapacaktır, yani çıktı katmanında benim bu fotoğraf için belirlediğim iki rakam değil, başka sayılar hesaplanacaktır.

Hatanın geri yayılımı algoritmasının görevi, ağdaki binlerce telin ağırlıklarını aynı girdi ileride bir kez daha görülürse, o sefer bu kadar hata yapılmamasını sağlayacak şekilde güncellemektir. Resimde amcam vardı, yani çıktı katmanındaki "amca" hücresinin 1 sayısını hesaplaması gerekiyordu, ama daha küçük bir sayı hesapladı diyelim. Bu yanlış hesap kimin kabahatidir? O hücrenin girdi tellerinin gerideki katmandan taşıdıkları sayılarla ağırlıklarının. Peki ama o geriden gelen sayılar kimin kabahatiydi? Bir önceki katmandaki ağırlıkların. Çok şık bir türev alma cinliğiyle ağda bu hataya katkı yapan her ağırlık, katkısı oranında bir daha sefere o kadar yanılınmayacak yönde azıcık değiştirilir. Sonra eğitim kümemizdeki her girdi-çıktı çiftiyle bu hesaplama-düzeltme döngüsü binlerce kez tekrarlanır. Her tekrarda ağımız bir önceki haline oranla biraz daha bilgili hale gelir. Yeterince uzun bir eğitimden sonra ağ artık öğretmeyi hedeflediğimiz kavramı (örneğimizde amcamla halamı tanımayı) "öğrenmiştir"; dahası, birçok farklı fotoğraf üzerinde çalıştığı için aile büyüklerimi birebir piksel düzeyinde değil, "genelleştirme" yaparak tanır. Yani sadece eğitim sırasında gördüklerini değil, daha önce hiç görmediği yeni fotoğraflar için de yüksek olasılıkla doğru yanıt vermesi beklenir.

Onlarca yıldır bilinen bu harika algoritmadan yıllarca beklediğimiz randımanı alamadık. Yukarıda anlattığım türden örüntü tanıma görevlerinde bir türlü istenen genelleştirme başarısına ulaşılamıyordu. Facebook kullanıcısıysanız hemen fark ettiğiniz gibi, artık durum öyle değil. Sinir ağlarının performansı kanatlandı. Nedenini önümüzdeki soruda göreceğiz.

29 | Derin öğrenme nedir?

Uzun süre akademik çevrelerin gündeminde olan, ama bir türlü gerçek uygulamalarda göz dolduracak performansa erişemeyen bir yapay öğrenme yöntemi olarak kalan yapay sinir ağları, 21. yüzyılın başlarında uçuşa geçip yapay zekânın en çarpıcı başarılarında rol aldı: Ses sinyali olarak verilen konuşmaları metne dönüştürme, resimlerde farklı cisimleri ve yüzleri tanıma, dilden dile çeviri, insanüstü seviyede oyun oynama ve daha birçok alanda uzmanları bile şaşırtan performanslarıyla günlük hayatı değiştirmeye başladı. Bu nasıl oldu?

Bu öyküye başlamadan önce 28. Soru'nun yanıtını bir kez daha okumanızı öneririm. Çünkü tüm bu atılımların ardındaki "derin öğrenme" teknolojisi aslında orada anlattığımız sinir ağı yapısının ta kendisi; o temel fikir üzerine birkaç zekice müdahale, biraz şans ve "zamanın ruhu" eklendi sadece.

"Derin öğrenme" deyimindeki "derin" kelimesi, ağın katman sayısının çok olmasından geliyor. Geçen soruda sadece üç katmanlı (yani girdi ve çıktı katmanları dışında tek bir dış dünyadan "gizli katman"ı olan) ağların her girdi-çıktı dönüşümünü gerçekleştirebileceğini garantileyen bir teoremden söz etmiştim. Ne yazık ki pratikte işler öyle yürümüyor. Hem o teoremin kanıtı o ortadaki katmanın gerçekçilik sınırlarının ötesinde çok sayıda hücre içermesini gerektiriyor, hem de bilgiyi işlerken girdideki "resim dosyasındaki pikseller" düzeyinden çıktıdaki "Ahmet amcamın yüzü" düzeyine tek adımda geçilmesini istemek beynimizdeki öğrenen yapının mantığına aykırı.

En basit kavramlardan başlayıp her düzeydeki kavramı daha alt düzeydeki kavramları temel alarak inşa ede ede karmaşık kavramlara ulaştığımız hiyerarşilere alışığız. Ağımızda çok sayıda katman olsa, girdinin işlenip yorumlanmasını bu kademeli mantığa göre gerçekleştirebiliriz: Girdide piksel renkleri varsa ilk gizli katmandaki hücrelerin

her biri resimdeki birkaç komşu pikselden oluşan dar bölgeye odaklanıp resmin orasında bir "çizgi" olup olmadığını, varsa da dikey mi, yatay mı, yoksa çapraz mı olduğunu hesaplayabilir mesela (İnanın bu doğrudan amcamı tanımaktan çok daha kolaydır!). Şimdi ağın geri kalanını kendi başına bir ağ olarak düşünün: Ham piksel renkleri yerine çizgi parçacıklarını girdi olarak alan bu ağın bir şekli tanıması ilk problemimizden biraz daha kolay, değil mi?

Sonraki katmanlar bu şekilde kademeli olarak çalışır: Biri çizgilerden burun, kulak gibi şekilleri tanıyan hücrelerden oluşur, sonraki bunların birleşimlerini tanıyanlardan.

Geçen soruda anlattığımız hatanın geri yayılımı yoluyla öğrenme algoritması, çok sayıda katmanı olan ağlarda çok iyi çalışamıyordu. 2006 yılında Boole'un torunu Hinton'ın başını çektiği bir ekip bu sorunu gidermenin bir yolunu keşfetti. Bu buluş diğerlerine yol açacak bir heyecan yarattı. Fakat çok katmanlı ağları eğitebilmek hâlâ uzun ve zahmetli bir hesaplama işiydi. Araştırmacıların kolayca edinebilecekleri ucuz süperbilgisayarları olsaydı keşke...

Sonra akla oyun bilgisayarları geldi. Oyun meraklılarının doymak bilmeyen hız ve görüntü kalitesi talepleri, bilgisayar üreticilerini ekranda üç boyutlu hareketi canlandırmakta gereksinilen matris çarpımlarını yapmak için 14. Soru'nun yanıtında sözünü ettiğimiz "merkezi işlem birimi"nden ayrı özel "grafik işlem birimleri" tasarlamaya itmişti. Yapay öğrenme araştırıcıları bu işlemcilerin sinir ağlarındaki binlerce paralel ağırlık hesabı için de biçilmiş kaftan olduğunu fark etti. Artık ağlarını çok daha uzun eğitimlere tabi tutabileceklerdi. Ama bu öğrenme yönteminin en büyük ihtiyacı veri, veri, daha çok veriydi. Rakibimiz olan biyolojik beyinler yıllarca gözleri açık dolaşan canlılara takılı oldukları ve sayısız görüntüye maruz kaldıkları için bu kadar iyiydiler örüntü tanıma işlerinde. Yapay sinir ağlarımız da bir kişinin albümündeki birkaç resimle değil de milyonlarca resimle eğitilse çok daha iyi öğrenmezler miydi? Acaba milyonlarca resmi bilgisayar ortamına sokmanın bir yolu var mıydı?

Evet, bu sorunu da "halk" çözdü. Çağ herkesin resim çekip onu bedavaya internete koyduğu çağdı. Facebook, Google gibi devler neden size resimlerinizi ve yazılarınızı koymanız için sınırsız bellek hediye ediyor? Çünkü onlara makinelerinin dünyanın nasıl göründüğünü ve insan dillerinin düzenini öğrenmesi için ihtiyaçları var! Bu "büyük veri" denizi sayesinde önceki kuşağın araştırmacılarının hayal edemeyeceği büyüklükte sinir ağları, devasa veri kümeleriyle eğitildi ve yapay zekâ birçok örüntü tanıma işinde insan düzeyine vardı, hatta geçti.

Ben bu kitabı yazarken derin öğrenmeyle edinilen başarı haberleri gelmeye devam ediyordu. "Her şeyi öğrenen bir makine! Yapay zekâya kavuşmamıza az kaldı!" Doğru mu?

Emin değilim. Derin öğrenmenin ne olduğunu unutmayalım: Verilen bir yığın girdi-çıktı çifti üzerinde antrenman yapıp bir dönüşüm çıkarsamak ve bu dönüşümün daha önce görmediği bir girdiyle karşılaştığında ona uygun çıktıyı vereceğini ummak. Bu işi, konuşma tanıma, go oyununda kazanma şansı yüksek pozisyonları tanıma, cisim tanıma gibi alanlarda yapabiliyoruz, çünkü bunların tümündeki dönüşümler öğrenme algoritmamızın matematiksel doğasının (türevlere vs. dayalı iç mantığına göre) kolayca öğrenebildiği fonksiyonlardan ("Kolay" derken yüz binlerce örneğin art arda gösterilmesini kastettiğimi unutmayın). Her fonksiyonun bu kadar kolay öğrenilebilir olması gerekmez. Hatta kimi fonksiyonlar bu şekilde gücümüzün yeteceği boyda ağlar tarafından hiç öğrenilemiyor olabilir. Mesela girdisinin bir bilgisayar programının yapması istenen işlerin yüksek seviyede bir tarifi, çıktısının da o işi gören gerçek bir program olan bir ağı, elimizde bu türden milyonlarca örnek girdi-çıktı çifti olsa bile, mevcut öğrenme yöntemlerimizle elde etmek imkânsız olabilir ("Makineler bunu yapamaz" deyip kendimle çelişmeyeceğim, ama kim bilir, belki bu öğrenme algoritması bu işi gerçekten yapamıyordur). Yapay zekâ araştırmacısı olmak için heyecanlı bir zamanda yaşıyoruz, çünkü proje hem artık meyve verir hale geldi, hem de hâlâ yeni buluşlar ve atılımlara ihtiyaç var.

30 | Bilgisayar buluş yapabilir mi?

İlk bilgisayar programcısının bilgisayarı yoktu, çünkü 19. yüzyılda doğmuştu. Turing'den bir asır önce bir başka İngiliz dahi olan Charles Babbage "analitik motor" adlı mekanik bir bilgisayar tasarlamış, ama günün teknolojisi bu tasarımı hayata geçirmeye yetmemişti. Bugünün gözüyle incelendiğinde, analitik motorun evrensel bir bilgisayar olduğunu, yani istenen hesabı yapmaya programlanabileceğini görüyoruz (Programlar, üstlerine açılan deliklerin yerlerinin farklı bilgileri temsil ettiği kartlar yoluyla bilgisayara verilecekti). Şair Lord Byron'ın kızı ve Babbage'ın dostu olan matematikçi Ada Lovelace, işte bu inşa edilmemiş makinenin işleyiş düzenini kavramış ve onun üzerinde çalıştırılabilecek bir program yazmıştı.

Ama Lovelace Kontesi bilgisayarın potansiyelinin tümünü anlamıştı diyemeyiz. Kendisi (Turing'in de 1950'deki makalesinde alıntılayıp eleştirdiği) şu sözleriyle "Bilgisayarlar şu işi yapamaz!" diyen ilk insan unvanının da sahibidir:

> Analitik motorun güçleri hakkında doğabilecek abartılı fikirlere karşı dikkatli olmak faydalıdır. Analitik motorun herhangi bir şey yaratmak konusunda hiçbir iddiası yoktur. Bizim ona yapmasını emretmeyi bildiğimiz her şeyi yapabilir. Analizi takip edebilir, ama herhangi bir analitik bağıntı veya gerçeği keşfetme gücü yoktur.[4]

1976'da Stanford Üniversitesi'nde doktora öğrencisi olan Douglas Lenat, "Otomatik Matematikçi" adında bir program geliştirdi. Bu bir "kavram keşfetme" programıydı. Lenat programa girdi olarak sadece hepimizin aşina olduğu "küme" kavramını ve matematik âleminde bir

4) Ada Lovelace'ın L. F. Menabrea'nın "Charles Babbage Tarafından İcat Edilen Analitik Motorun Tarifi" makalesinin çevirisine eklediği notlardan.

kavramdan başka bir kavram elde etmek için kullanılabilecek yüzlerce kuralı verdi. Bu kurallar örneğin "Elindeki kavram için örnekler üretmeye çalış", "Eğer pek örnek bulamıyorsan elindeki kavram fazla dar tanımlanmış olabilir, tanımı genelleştir", "Bulduğun örnekler arasında uç özellikleri olanlar ilginç olabilir, sadece onları kapsayacak şekilde kavramın tanımını özelleştir" gibi genel fikirlerden oluşuyordu. Lenat bu noktadan başlattığı programın "kavramlar âlemi"nde ilerleyişini izledi ve hâlâ derslerde anlatılan bir buluşlar dizisine tanık oldu. Otomatik Matematikçi, küme kavramından yola çıkıp (farklı kümelerin eleman sayıları farklı olabileceğinden) "sayı" (daha doğrusu 0, 1, 2, 3, ... diye giden "doğal sayılar") kavramını, oradan bu sayılar üzerinde yapılabilecek toplama, çarpma, bölme gibi işlemleri, sonra bazı sayıların çok sayıda, bazılarınınsa sadece iki adet sayının çarpımı olarak yazılabildiğini fark edip bu "sadece iki böleni olma" özelliğine odaklanarak 2, 3, 5, 7, 11, ... diye giden "asal sayılar" kavramını keşfetti. Dahası, ürettiği örneklerden yola çıkarak "Galiba 2'den büyük her çift sayı, iki asal sayının toplamı olarak yazılabiliyor" fikrine kapıldı, ki bu 1742'de Alman matematikçi Christian Goldbach tarafından ortaya atılmış önemli bir matematiksel iddianın ta kendisiydi! Birkaç yüzyıl farkla da olsa, yapay zekâ insan bir matematikçiyle aynı buluşu yapmıştı.

ABD'de popüler olan "Traveller" ("Yolcu") adında bir kutu oyunu var. Bu oyunun ulusal turnuvaları düzenleniyor. Turnuvada amaç, bir trilyonluk devasa bir bütçe kullanarak kuracağınız bir filoyla rakibinizin filosunu alt edip bir sonraki tura geçmek. Bu iş için uymanız gereken kısıtlar ve savaş sırasında kazananın belirlenmesinde kullanılacak kurallar kalın bir kitabı dolduruyor.

1981'deki şampiyonaya Eurisko adında bir YZ programı da katıldı. Eurisko'nun programcısı, Otomatik Matematikçi'nin yaratıcısı Doug Lenat'tan başkası değildi. Lenat, Otomatik Matematikçi'nin yaptığı buluşların programın yazılış şeklinden kaynaklandığı, yani bilinçli

olarak veya olmayarak kendisinin programa kopya verdiği iddialarına, kendisinin hiç anlamadığı bu savaş oyununu oynayan yeni bir sistem kurarak cevap vermek istemişti.

Lenat Traveller kitabındaki kuralları bir bilgisayara kodladı ve Eurisko'yu iyi bir strateji belirlemesi için çalışmaya bıraktı. Eurisko günler boyunca kendi kendine oynayarak kazanmak için çeşitli yöntemler aradı. Lenat her sabah Eurisko'nun o gece ürettiği fikirleri inceleyip ("oyunun kurallarını değiştirelim ve kendimizi galip ilan edelim" gibi) insan dünyasında olanaksız olanları eliyordu. Sonuçta program hangi yapıdaki filoların kazanma şansının daha çok olduğunu hesaplamaya odaklandı.

Şampiyonanın ilk turunda Eurisko'nun filosunu gören rakipleri, gülmekten kendilerini alamadılar. Denizcilik tarihindeki tüm savaşları ezbere bilen Traveller meraklıları, bu bilgilerinden damıttıkları farklı güç ve kabiliyetlerdeki gemilerden oluşan filo yapılarıyla oynarken, Eurisko tüm parasını oyunun elverdiği maksimum sayıda, dişine kadar silahlı, küçücük ve yavaş "torpidobot"lara harcamıştı. Fakat turnuvanın sonunda gülen Lenat oldu. Eurisko'nun filosu o kadar kalabalıktı ki, karşılıklı her salvoda onun rakibinden daha fazla gemisi vurulsa da sonunda rakibin gemileri tükendiğinde hâlâ Eurisko'nun birkaç gemisi sağlam kalmış oluyor ve oyunu kazanıyordu. Gerçek hayatta uygulansa kendi askerlerini kesin ölüme atmak demek olacağından hiçbir insanın aklına gelmeyecek bu taktik Eurisko'yu 1981 şampiyonu yapmıştı.

1982 turnuvası öncesinde şampiyona düzenleyicileri kuralları Eurisko'nun önceki yıldaki taktiğini işe yaramaz kılacak şekilde değiştirdiler: Artık kazanan filonun ortalama "çeviklik" değeri de hesaba katılacak, böylelikle rakibin ateşi sonucu sakatlanan çok sayıda gemisi olan Eurisko eskisi kadar çok puan alamayacaktı. Lenat Eurisko'yu bu yeni kurallarla yeniden çalıştırdı. Sonuç, yine "insanlık dışı" bir buluş oldu: Program yine önceki yıldaki gibi kalabalık filolar kuruyor, fakat oyun sırasında vurulup çeviklik değeri düşen gemilerini kendi kendisine batırıyordu! Böylelikle kalan gemileri yüksek çeviklik

ortalamasına sahip olan Eurisko, 1982 kupasını da rahatlıkla aldı.

Lenat, düzenleyicilerin "Eğer seneye de gelirseniz, şampiyonayı iptal ederiz" mealindeki nazikçe uyarıları üzerine Eurisko'yu sahalardan çekme kararı aldı. Zaten ABD Savunma Bakanlığı'nın dikkatini çekerek sonraki projeleri için sağlam bir destek kaynağı bulmuştu. Kendisiyle ilerleyen sayfalarda yine karşılaşacağız.

31 | Bilgisayar sanat yapabilir mi?

Eğer "sanat yapmak" derken resim, beste, şiir vs. sanat ürünleri ortaya koymayı kastediyorsanız, evet, yapabilir; örneklerini aşağıda vereceğim. Yok eğer sanatçının iç dünyasında kopan fırtınalardan, esere dönüşen duygulanımlardan filan bahsediyorsanız, o soruyu bir soruyla yanıtlamayı yeğlerim: Hayranı olduğunuz sanatçıların kafasının içine bu üretim sürecinde hiç baktınız mı? Onların zihninde az sonra anlatacağım yapay zekâ programlarınınkinden öte bir şey olduğunu nereden biliyorsunuz?

İşin felsefesini tartışmayı sonraya bırakıp sanatçı YZ örneklerine geçelim.

Harold Cohen, 1928'de Londra'da doğdu. Resme yeteneği olduğu birkaç yıl içinde anlaşıldı. 30'lu yaşlarında ülkesini Venedik ve Paris bienallerinde temsil edecek derecede saygın bir konumdayken sanatsal bir tıkanıklık yaşadığını hissetmeye başladı. 1968'de San Diego Üniversitesi'nde bir dönemliğine ziyaretçi hoca olarak gitmeye karar verdiğinde hayatının geri kalanını Kaliforniya'da geçireceğinden haberi yoktu.

Cohen bir öğlen üniversite yemekhanesinde bir bilgisayar hocasıyla aynı masaya düştü. Yemek bittiğinde bilgisayar programlamayı öğrenmeye karar vermişti. Bir zanaat hakkında ne bildiğinizi anlamanın en iyi yollarından biri,

onu bir bilgisayara yaptırabilecek kadar açık şekilde ifade etmek, yani programını yazmaktır. San Diego'da kadroya giren Cohen, birkaç yıl sonra bu kez Stanford Üniversitesi'nin yapay zekâ laboratuvarındaki konukluğu sırasında "yapay ressam" AARON programını yazmaya başladı.

Cohen resim kuramının temellerinden başladı ve sabırla çalıştı. Soyut resimlerle işe başlayan AARON yaklaşık 10 yıl içinde üç boyutlu uzayda taşlar, bitkiler ve insanlar gibi objeleri tatminkâr şekilde konumlandırmayı öğrendi. Cohen'in kendisini de pek yetkin hissetmediği renklendirme becerisini içine sinecek şekilde kodlayabilmesi ise 20 yılı aldı! (Cohen daha sonraları AARON'ın renklendirmede kendisinden daha iyi olduğunu gururla söyleyecekti). AARON'ın resimlerini bilgisayar ekranına çıkardığını sanmayın sakın. Cohen AARON'a seçeceği boyaları karıştırmasına ve sonra kâğıda sürmesine el veren özel bir yazıcı inşa etmişti.

Resim sanatından anladığımı iddia etmeyeceğim, ama ben AARON'ın eserlerini beğenirim. Dünyanın dört bir yanında birçok saygın müzede sergilenmeleri de boşa değildir herhalde. Zevkler tartışılmaz, bir internet araması yapıp kendiniz göz atmaya ne dersiniz?

Herhangi bir görüntüyü girdi olarak almayan, iç gösterimine Cohen tarafından kodlanan obje tanımlarına dayanan AARON'ın aksine, Simon Colton'ın yazmaya 2001'de başladığı The Painting Fool ("Resim Yapan Budala") programı girdi olarak fotoğraf kütüphaneleri ve (başlarda kullanıcısından aldığı, sonraları da haber sitelerinden kendi okuduğu haber metinlerinden çıkarsadığı) "duygu durumları" alıp o duyguya uyumlu resimler üretir. *Amelie* filminin başoyuncusu Audrey Tatou'nun 22 fotoğrafından bu şekilde elde edilmiş 222 resim içeren galeriyi görmenizi öneririm.

Bilgisayarlar müzikte de gayet iyiler (Müziğin matematiksel bir altyapısı olduğunu düşünürsek bu şaşırtıcı olmamalı). Benim en sevdiğim örnek, tıpkı Cohen'in resim için yaptığı gibi, yaşamının onlarca yılını otomatik besteciler üretmeye adamış olan David Cope'un EMI

("Experiments in Musical Intelligence" = "Müzik Zekâsı Deneyleri") programıdır. Birçok ünlü bestecinin stilinde özgün eserler besteleyen EMI, çok sayıda müzikseveri defalarca "Şu iki eserden hangisi Bach'ın, hangisi makinenin, bilin bakalım?" türünden "müzik Turing testleri"nde kandırmayı başarmıştır.

Çok beğendiğim yapay zekâ sanatçısı Bager Akbay'ın *Posta* gazetesinin "Yurdumun Şairleri" köşesinde yayımlanabilecek kalitede şiirler yazan bir "robot şair" geliştirme serüvenini anlatmazsam olmaz. Akbay yapay zekâsının bu süreçte insan şairlerin geçtiği tüm aşamaları yaşamasını kafasına koymuştu. Gazetede çıkan şiirlere şairin bir vesikalık fotoğrafı eşlik ettiğinden yaratılacak yapay karakterin de bir yüzü olmalıydı. Akbay gazetedeki şair fotoğraflarını matematiksel olarak birbiriyle kaynaştırıp bir "ortalama şair yüzü" elde etti. Erkek ve kadın resimlerinin ortalaması olduğundan cinsiyeti anlaşılamayan bu yüze Türkiye istatistiklerine göre iki cinsin de kullandığı isimlerin en popüleri olan "Deniz Yılmaz" da eklenince kimlik sorunu çözülmüş oldu.

Şiirleri üretecek algoritma için çalışmalara basitten başlayan Akbay, sözlükten sadece uyak ve ölçü kaygısıyla seçilmiş sözcüklerin art arda dizilmesiyle bir yere ulaşılamayacağını görünce, önce bir gazete yazıları külliyatına, o kaynak "Sunum duruşmanın / Taksit unutmayın / Reyting barındıran / Özden adımların" gibi "şiir kokmayan" sonuçlar verince de internette bulduğu 12.000 şiir içeren bir siteye yöneldi. Buradaki kelimelerin dizilmesinde istatistiksel olarak bu külliyattaki her bir sözcükten sonra gelme olasılığı en yüksek olan adayın seçilmesi (telefonunuzda bir metin yazarken her kelimeden sonra size önerilen yeni kelime de böyle seçilir) yöntemi de kullanıldığında, ödül kazanmasa da *Posta* çıtasını aşabilecek gibi görünen ürünlere ulaşıldı:

Öpmekte diretmem nedense
Dünyayı tutacak nerdeyse
Gecesel dilleri altından
Seneler asırlar değişse

Deniz Yılmaz şiirlerini, hatta gazeteye göndereceği mektubun zarfındaki adresi bile bilgisayar kontrollü kola tutturulmuş bir kalemle, Akbay'ın bu iş için tasarladığı doğal görünüşlü bir harf kümesi kullanarak "kendi elyazısıyla" kâğıda döktü. Ama gelin görün ki bu kısıtlı Turing testini geçemedi. Şiirleri *Posta*'da yayınlanmadı. Yine de önce Facebook sayfasında, sonra da yer aldığı sanat fuarlarında tanınınca bir okur kitlesine ulaştı ve bir kitap bile çıkardı.

Sanatın değeri büyük ölçüde "bakanın gözünde" olduğundan, bu ürünleri herkesin farklı değerlendirmesi normal. Bu yanıtı bir bilmeceyle bitireyim: Sizce aşağıdaki şiirin yazarı makine midir, insan mı?

Gelsene dedi bana
Kalsana dedi bana
Gülsene dedi bana
Ölsene dedi bana

Geldim
Kaldım
Güldüm
Öldüm

32 | Bilgisayarlar avukatlık yapabilir mi?

Kitabımızın 2. sorusunda tanıştığımız, yapay zekânın "dedesi" diyebileceğimiz Leibniz, bir hukukçuydu: 28 Eylül 1665'te Leipzig Üniversitesi'nden hukuk lisans diplomasını alır almaz doktora yapmak için başvurdu. "Daha 19 yaşında doktora mı olurmuş!" gerekçesiyle reddedilince hemen Altdorf Üniversitesi'ne geçti. Bir yıl içinde (zaten Leipzig'deyken yazmaya başladığı) doktora tezini sunup avukatlık yapma lisansını elde etmişti.

Leibniz'in "mantık makinesi" hayalini "Kişiler arasında anlaşmazlıklar olduğunda hemencecik 'Hesaplayalım,

kimin haklı olduğunu görelim' diyebilelim" diye ifade edişini görmüştük. Aslında düşlediğinin bir tür "robot hâkim" olduğu hakkında bana katılır mısınız?

Hukuk mantık yürütme, kural zincirleri, gerekçelendirme gibi "eski moda" simgesel yapay zekâ tekniklerinin kullanımına çok elverişli bir alandır. Ama son yıllarda avukatları "Acaba işimiz elden gidiyor mu?" diye düşündüren uygulamalar, genellikle büyük veriyle baş etmek için geliştirilen tekniklere ve yapay öğrenmeye dayalı.

Bir dava üzerinde çalışılırken eldeki duruma benzerlik gösteren eski mahkeme kararlarının ve konuyla ilgili tüm mevzuatın (ne kadar gözlerden uzak kalmış, az bilinen bir köşedeyse o kadar iyi) eksiksiz bulunması çok önemlidir. Eskiden loş ve havasız arşivlerde uzun süre toz yutmayı gerektiren bu arama işini artık bilgisayarlar hem daha hızlı, hem de eksiksiz şekilde gerçekleştirebilir. Her yazılı belgeyi dijital ortama aktarmanın bir kerelik sabit külfetine karşılık artık hiçbir bilginin gözden kaçmayacağı bir bütünlük garantisine kavuşmanın avantajını elde ederiz. Bunun nasıl bir nimet olduğunu bizim nesil hemen takdir eder, ama Google dünyasına doğan gençleri "metin madenciliği"nin yapay zekânın belki de en yararlı ürünü olduğu konusunda ikna etmek vakit alabilir.

Gregor Mendel adında bir keşiş, bugün Çek Cumhuriyeti'ne ait olan Brno kentinin manastırının bahçesindeki bezelyeler üzerinde yıllar boyu titizlikle sürdürdüğü deneyler sonucunda geliştirdiği kuramı 8 Şubat 1865'te bir seminerde ne dediğini pek anlamayan bir grup izleyiciye sunmuş, 1866'da da *Brünn Doğa Araştırma Derneği Dergisi*'nde bir makale olarak yayımlamıştı. Fakat genetik biliminin kurucu metni olan bu eserden neredeyse 35 yıl boyunca bilim dünyasının geri kalanının haberi olmadı. Canlılarda kalıtımın nasıl işlediğini anlamaya can atan biliminsanları, Mendel'in ölümünden 16 yıl sonra kendi deneylerinin onun vardığı sonuçları tekrarladığını fark edebildi.

Bugün araştırma alanındaki literatürü tarayan bir doktora öğrencisinin veya davasıyla ilgili içtihat arayan bir

avukatın böyle bir falso yapma ihtimali çok daha düşüktür. Hem dijital belgeler daha üretilirken böyle aramalarda kullanılmak için tasarlanmış sınıflandırma sistemlerine göre etiketleniyor, hem de "gözetmensiz öğrenme" denilen bir yapay öğrenme tekniğiyle böyle etiketleri bilgisayarın kendi kendine belirlemesi mümkün.

Bilgisayara binlerce metin dosyasını verip bunları konularına göre gruplamasını isteyelim. Metinlerde "Konu: ..." diye başlayan bir hane filan varsa ne âlâ, ama aksi takdirde, son derece zor bir iş olan "doğal dil anlama" (37. Soru) problemini tümüyle çözmeden bu gruplama nasıl mı yapılabilir?

Bilgisayar, metinlerdeki kelimeler üzerinde istatistiksel analiz yapar. Her metin ve her kelime için o kelimenin o metinde geçme sıklığını (yani metindeki tüm kelime nüfusunun yüzde kaçını o kelimenin oluşturduğunu) hesaplar. Sonra metinleri kendi aralarında bu sıklıklar açısından benzeyen metinler yan yana gelecek şekilde sıralar. Örneğin atlarla ilgili metinlerde "at", "doru", "arpa", "dizgin" vs. kelimelerinin, kedilerle ilgili olanlardaysa "kedi" veya "miyav" gibilerin daha sık geçmesi beklendiğinden bu iki konudaki dokümanlar (ne atın ne de kedinin ne olduğu hakkında hiçbir şey bilmeden, dilerseniz hiç bilmediğiniz bir dildeki metinler üzerinde çalışarak) birbirlerinden ayrıştırılmış olur. Bu temel fikrin üzerine eklenen bir dizi cinlikle "anahtar kelime çıkarımı" işlemi yapılabilir.

Google gibi milyarlarca dokümana erişim sağlayan araçlar, bu gibi istatistiksel tekniklerle "anlam ağı"nı keşfetmemize olanak tanır: Genellikle aynı dokümanlarda bir arada sıklıkla yer alan (ve "ve", "bir" filan gibi konudan bağımsız olarak her dokümanda mebzul miktarda geçenlerden olmayan) sözler birbirleriyle ilintili anlamlara sahiptir. Eski bir öğrencimin mahkeme kararları üzerinde kurduğu böyle bir anlam ağından yararlanarak örneğin "irtikâp" ile "nitelikli zimmet" kavramları arasında bir yakınlık olduğunu keşfeden bir yazılım ürettiğini gördüğümde çok sevinmiştim.

28. Soru'da gördüğümüz "gözetimli öğrenme" teknikleri

de avukatların işine yarayabilir. Elinizde binlerce mahkeme kararı varsa bunların davanın ayrıntılarını içeren baş kısımlarını girdi, sondaki kararı da çıktı olarak ayırıp bu dönüşümü makinenize öğretmeyi deneyebilirsiniz. 2017'de Londra'da yapılan bir deneyde 100'den fazla insan avukat, belli bir kredi kartı usulsüzlüğü konusunda Finans Ombudsmanı'na yapılmış yüzlerce gerçek başvuru dosyasının kabul edilip edilmeyeceğini tahmin etme konusunda bu şekilde eğitilmiş bir yapay zekâ programıyla yarıştı. İnsanların doğru tahmin oranı % 66,3'te kalırken yapay zekâ % 86,6 oranında tutturdu.

2016'da New York'taki Baker Hostetler hukuk firması, iflas davalarında yararlanmak için IBM'in Watson yapay zekâ altyapısınca desteklenen ROSS platformunu "işe aldı". Ama hukukta yapay zekânın sadece avukatlara değil, hâkimlere de gerektiği kanısındayım. Bu konuya son soruda döneceğiz.

33 | Kasparov'u nasıl yendik?

Satranç (ve akrabası olan oyunları) bilgisayara oynatmak için kötü bir fikir anlatacağım: Oyun sırasında herhangi bir noktada (en başta da olabilir) sıra bilgisayara geldi diyelim. Makine o andaki oyun durumunu (taşların tahtadaki yerleri ve şu ana dek rok yapılıp yapılmadığı gibi bilgileri) "sayfa"nın (bilgisayarın belleğinde bir alanı kastediyorum) en üstüne yazar. Bu durumda yapılabilecek bütün hamleler (sözgelimi başlangıç durumunda tam 20 hamle mümkündür) sonucu ortaya çıkacak yeni oyun durumları, sayfada ilk durumun altındaki satıra yan yana yazılır. Bir tür soyağacı oluşturuyoruz; D durumundan bir hamleyle ulaşılabilecek tüm diğer durumlar, D'nin "çocukları" olarak ağaçta işaretleniyor. Bu yeni nesil durumlarda sıra artık rakibe geçmiştir ve (eğer oyun

oracıkta bitmediyse) bu durumların her birinin de bu kez rakibin yapabileceği tüm hamlelere karşılık gelen kendi çocukları vardır. Bu üçüncü nesil durumları da daha alta yazıp her birini "anneleri" ile ilişkilendirelim. Bu işleme devam edelim. Çocuksuz durumlardan (yani ağacımızın "yaprak"larından) bazılarında rakibin bizi mat ettiği görülecektir. Bunları kırmızıya, diğer çocuksuz durumları da yeşile boyayalım. Ortaya çıkan yapıya "oyun ağacı" denir.

Matematiksel olarak gösterilebilir ki, bu tür "iki sonuçlu" oyunlarda, oyunculardan birinin kesinlikle bir "kazanma stratejisi", yani, rakibi ne hamle yaparsa yapsın, oyunun kendisi için iyi olan renkteki bir son durumda bitmesini garantileyebileceği cevabi hamleler dizileri vardır. Eğer satrancın oyun ağacını tümüyle oluşturup inceleyebilseydik, içinde gömülü olan bu stratejiyi ayrıştırıp kullanarak evrenin en iyi satranç oyuncusu olabilirdik: Kazanma stratejisi her tahta pozisyonunda hangi hamleyi yapmanız gerektiğinin yazılı olduğu dev bir ansiklopedi gibidir. O hamleyi yapıp rakibinizin cevabını beklersiniz. Sonra oluşan yeni pozisyon için en uygun hamleyi stratejiniz yine size söyler. Bu şekilde devam edersiniz; kaybetmeniz olanaksızdır.

Satrancın heyecanlı bir oyun, yukarıdaki "tüm ağacı kurup iyi sonlara giden yolları izle" algoritmasının da kötü bir fikir olmasının nedeni, bu ağacın boyutunun bilgisayarların bu hesaplamayı yapmasını olanaksız kılacak ölçüde (yaprak sayısı evrendeki atom sayısından daha çoktur!) büyük olmasıdır. Peki satranç gerçekte nasıl oynanır?

Bir "eski moda" satranç YZ'si, yukarıda anlatılan ağacı bu hamle için kullanabileceği süre boyunca yapabildiği kadar "derin"leştirir. Sürenin tüm ağacı oluşturmasına yetmeyeceğini gördüğünde yeni "çocuk" durumlar oluşturmaktan vazgeçer ve o ana dek üretilmiş kısmi ağacın yapraklarını kırmızı veya yeşile boyamaya başlar. Oyunun gerçekten bittiği durumlara tekabül eden yaprakların nasıl boyanacağını yukarıda anlatmıştık. Program

henüz "doğurgan" olan bir yaprağı hangi renge boyayacağına ise tahtadaki o durumun durağan özelliklerine göre yaptığı bir tahmin sonucu karar verir. Kısaca, durumun "iyi" göründüğü (sözgelimi, kendi taşlarının sayısının rakibinkinden çok olduğu, şahının iyi korunduğu, merkez karelere hâkim olduğu) yaprakları yeşile, "kötü" göründüğü yaprakları da kırmızıya boyar. Bu noktada kullanılan (ve mesela usta satranççıların da yer aldığı Deep Blue ekibinin 700.000 büyük usta oyunundan yararlanarak binlerce parçada organize ettiği) formül, "geleceği" tümüyle dikkate almadığından yaprağın yanlış renge boyanmasına, bu da yanlış bir strateji çıkarsayan bilgisayarın ideal olmayan bir hamle yapmasına yol açabilir elbet. Ama işin (bilgisayarcılar için) güzel yanı şu ki, insan oyuncular da bellek ve zaman kısıtları nedeniyle oyun ağacını sadece kısmen "görebilirler" ve genellikle kim birkaç hamle daha ileriyi görebiliyor, yani ağacı birkaç nesil daha derine götürebiliyorsa o kazanır. Deep Blue Kasparov'u böyle yenmiştir.

34 | AlphaGo dünya go şampiyonunu nasıl yendi?

İlk soruda da söz ettiğim gibi, 2015 yılı itibariyle bilgisayarlar go oyununu insanlara rakip olacak düzeyde oynayamıyordu. Deep Blue'nun Kasparov'u satrancın süre kısıtları içinde yenmek için kullandığı yöntemin merkezinde makinenin olası hamlelerin getireceği oyun pozisyonlarını "zafere yakınlık" açısından sıralamakta kullandığı ve büyük emekle hazırlanan karmaşık bir formül bulunduğunu anlatmıştım. Go için böyle iyi bir değerlendirme formülü bir türlü bulunamamıştı. Çekişmeli bir go oyununu izleyen iki ustanın birbirleriyle hangi tarafın kazanmaya daha yakın olduğu konusunda anlaşamaması sık görülen bir durumdu. Anlaşılan ustaları

bile oyunu net ifade edemedikleri sezgilere ağırlık vererek oynuyordu.

Zamanın en iyi go programları geçen soruda anlattığım oyun ağacının rasgele hamlelerle keşfedilmesi fikrine dayanan "Monte Carlo ağaç araması" algoritmasını kullanıyordu. Bu yöntem ara pozisyonlar için bir değerlendirme formülüne gerek duymaz. Daha önce görmediği bir pozisyonu değerlendirmesi gerektiğinde rasgele bir hamleyi seçer. Eğer böylelikle taraflardan birinin kazandığı bir duruma varırsa bu bilgiyi bu noktaya gelirken geçtiği pozisyonlar hakkında tuttuğu kayıtlara yansıtır. Böyle çok sayıda deneme sonucu farklı pozisyonlar için "buradan şu hamle yapıldığında kazanma sıklığı yüzde şu kadardır" türünden oranlar elde edilir ve daha sonrasında artık tanınan bir pozisyonla karşılaşıldığında denenecek hamle rasgele değil, bu istatistiklere göre başarı şansı en yüksek görünen alınarak seçilir.

Monte Carlo ağaç araması, "pekiştirmeli öğrenme" diye bilinen bir yapay öğrenme teknikleri ailesinin üyelerindendir. Doğadan esinlenen bilgisayar yöntemlerinden biri olan pekiştirmeli öğrenme yaklaşımının temel sorusu, nasıl işlediğini tam bilmediğimiz bir ortamda performansımızı iyileştirmek için hangi eylemlerden sakınmak ve hangilerini seçmek gerektiğidir. Çevreden alınan "ödül" veya "ceza" sinyali bazen gecikmeli gelebilir: Günün sonunda hastalandık, ama acaba buna gün içinde yediğimiz ya da yaptığımız şeylerden hangisi neden oldu? Maçı kazandık, ama acaba kaçıncı adımdaki hangi hamle oyunun kaderini değiştirdi?

Monte Carlo ağaç araması uzmanların kodladığı bir pozisyon değerlendirme formülünün yokluğunda perişan olan geleneksel yöntemden daha iyi çalışıyordu, ama gerçekçi sürelerde toplayabildiği istatistikler gonun dev oyun ağacının dişinin kovuğunu bile doldurmuyordu. Dedim ya, 2015'te goda insanların üstünlüğü tartışılmazdı.

Ve sonra AlphaGo çıktı. 2017 baharında bir dizi efsane maçın sonunda bütün insan oyuncuları geride bıraktığı

kesinleşen AlphaGo, aynı yılın Ekim'inde tahtını AlphaGo Zero adındaki bir üst sürümüne terk etti.

AlphaGo'nun sırrı pekiştirmeli öğrenmeyi 29. Soru'da gördüğümüz derin öğrenme teknikleriyle desteklemesindeydi. Monte Carlo ağaç aramasının "keşif" aşamasında hamleleri kör şansa değil, bir pozisyon değerlendirme formülüyle hesaplanan başarı olasılıklarına göre seçtiğini düşünün. Eğer formülünüz kaliteliyse bu birleşik yaklaşım bilgisiz, "saf" Monte Carlo'dan daha iyi sonuç verecektir tabii. Peki ama insanların go için bir türlü yazamadığını yukarıda anlattığımız bu formülü nereden bulacağız? Öğreneceğiz!

AlphaGo'nun sinir ağına önce binlerce insan oyunundan pozisyonlar gösterilerek hangi hamlelerin tercih edildiği öğretildi. Elbette ki insanları geçmek için yeterli olmayan bu altyapı, sonraki "kendi kendine öğrenme" aşamasında oynayacak programların başlangıçta kullanacağı değerlendirme formülünü oluşturmakta kullanıldı.

AlphaGo bir insanın ömrü boyunca oynayabileceğinden çok sayıda oyunu kendi kendisine oynadı. Her oyunda kazanan kopya kaybedene oranla bir şeyi daha iyi yapmış olmalıdır, değil mi? Oyunun sonundaki bu "ödül" sinyali pekiştirmeli öğrenme yoluyla önceki aşamalardaki pozisyonlara yansıtılarak onların da "makbul" olduğu bilgisi sinir ağına kodlandı. Böylece her aşamada daha da iyileşen değerlendirme formülü, bir sonraki aşamada kendi kendine oynanacak oyunların daha yüksek kalitede olmasına yol açıyordu. DeepMind mühendisleri bu döngüyü durdurduklarında program insanüstü seviyeye ulaşmıştı.

AlphaGo Zero'nun AlphaGo'dan farkı ise insan bilgisine başta bile hiç ihtiyaç duymamasındaydı. Sadece oyunun hamle ve kazanma kuralları bilgisiyle donatılan AlphaGo Zero doğrudan kendi kendine oynamaya geçti. Başlarda yaptığı hamleler aptalcaydı elbet; ama binlerce oyundan sonra kazanmaya götüren oyun tarzını keşfetti ve sadece 40 günlük bir antrenmanla atası AlphaGo'yu yenecek hale geldi. Galaksinin yeni şampiyonu oydu artık.

35 | Kendi kendini süren otomobiller nasıl çalışır?

"Robot" kelimesi de, robotları konu alan mühendislik dalının adı olan "robotik" kelimesi de bilimkurgu yazarları tarafından dünyaya armağan edildi. Çekçe "zorla çalıştırma" anlamında bir kelimeden bozularak elde edilen "robot" sözcüğü, ilk kez 1920'de Karel Čapek'in *R.U.R.* adlı tiyatro oyununda kullanıldı. "Robotik" sözcüğü de gün ışığını bilimkurgunun devlerinden Isaac Asimov'un 1941'de yayımladığı "Yalancı!" başlıklı kısa öyküsünde gördü. Aradan geçen zamanda robotlar gerçek olmakla kalmadı, başka gezegenler gibi insanların gidemediği yerlerde bizi temsil de ettiler. Boston Dynamics şirketinin tanıtım videolarında kendilerini tekmeleyen ve sopalarla itip kakan mühendislerin engellemelerine karşın işlerine bakan insansı veya köpeksi robotları halkın sempatisini kazandı (Bu satırları yazmaya başlamadan birkaç hafta önce bir televizyon habercisi bana canlı yayında "Hocam, bu robot kapıdan çıkmasın diye kuyruğunu çeken adama dönüp bir tane çakar mı?" diye sordu).

Kuşkusuz robotların şimdiye dek gördüğümüz yapay zekâ uygulamalarından en önemli farkı, birer bedenlerinin olmasıdır. Bu, sadece bir yazılımdan ibaretseniz ve bir bilgisayarın içinde çalışıyorsanız, hiç kafa yormanızın gerekmeyeceği bir yığın sorunu beraberinde getirir. Robotlar girdilerini bedenleri üzerindeki algılayıcılar yoluyla dış dünyadan gelen sinyallerden süzmek, ne yapacaklarına karar vermek (ne yapacağına ilişkin komutları uzaktaki bir insandan alan ve sadece bir tür uzak organ görevi gören "uzaktan kumandalı" robotlardan değil, "özerk" olanlardan bahsediyorum), ve bu kararı eyleyicilerini kullanarak hayata geçirmek zorundadır. Gerçek dünyanın karmaşıklığıyla olan bu içli dışlılık, robotun işini iyice çetrefil hale getirir: Kameralarım değişik ışık veya hava koşullarında yeterince kaliteli görüntüler alıyor mu? Şu anda nerede olduğumu biliyor muyum? Ya

birisi beni iter veya kaldırıp başka bir yere koyarsa? Diyelim doğru konumumu biliyorum ve hedefime gitmek için tekerleklerimi tam yüz kez döndürmem gerektiğini hesapladım. Peki ya lastiklerimin havası kaçmış ve çapları kısalmışsa? O zaman yüz devir beni istediğim yere ulaştıramaz!

Bu nedenlerle iyi bir robotikçinin hem yazılımından donanımına eksiksiz bir bilgisayar mühendisliği uzmanlığına, hem makine mühendisliği becerilerine, hem de bende olmayan türden engin bir sabra sahip olması gerekir. Kendilerine hayranlığımı vurgulayarak konuya döneyim.

Boston Dynamics'inkiler bir yana, son yılların en çok konuşulan ve muhtemelen yakın gelecekte yaşamımızı en görülür şekilde değiştirecek olan robotları sürücüsüz otomobillerdir. Her yıl yüz binlerce insan trafik kazalarında ölmekte, milyonlarcası da yaralanmaktadır. Özel arabaların vakitlerinin yaklaşık %95'inin "yatarak", yani park halinde geçtiği hesaplanmıştır. Özellikle bazı yerlerde otomobiliniz yoksa veya onu süremeyecek durumdaysanız hareket kabiliyetiniz neredeyse sıfıra inmektedir. Tüm otomobiller kendi kendilerini sürebilseler ve özellikle kentlerde kişilerin malı değil, kiralanabilen bir hizmet olarak görülseler, bu sorunlar ortadan kalkacaktır: Bir yerden diğerine gitmeniz gerektiğinde akıllı telefonunuzdan en yakınınızdaki sürücüsüz arabayı çağırıp binersiniz. Sizi istediğiniz yere bırakıp sonraki yolcusuna doğru ilerler. Park derdi kalmaz; şimdinin park alanları makineler değil insanlar için kullanılabilir. Arabalar çok daha verimli kullanılacaklarından çok daha azı yeterli olur. Robot sürücüler uyuklamaz, dikkati dağılmaz, trafik kurallarını çiğnemez, insanların sahip olmadığı radar gibi algılayıcıları nedeniyle çevreden, bağlantı yetenekleri nedeniyle de trafikteki tüm diğer araçların ne yaptığından ve yapacağından yüksek düzeyde haberdardır. Günümüzde insanların birbirleriyle koordine olamamaları nedeniyle yavaş akan trafik hayranlık verici derecede hızlanır, kazalar ve ölümler sıfıra yaklaşır. Hedef budur.

Öğrencilik yıllarımda yapay zekânın çok zor problemlerinden biri olarak görülen otonom taşıtlar alanında son yıllarda elde edilen önemli ilerlemelerin ardında yine derin öğrenmedeki sıçramalar yatmaktadır: Araba sürme becerisi, yol ve taşıt hakkında şu anda elinize geçen bilgilere dayanarak hemen en doğru kararı (direksiyonu şu açıyla şu yöne kır, veya frene bas, veya gaza bas, vs.) uygulamaya koyabilmeyi gerektirir. Yolda karşılaşılabilecek sayısız olası girdinin (diyelim, ileriye ve geriye bakan kameralardan gelen görüntülerle taşıtın halihazırdaki momentum değerinden oluşan bilgi kümesinin) her biri için cevaben ne yapılması gerektiğini teker teker yazabilecek bir bilgisayar programcısı daha anasından doğmamıştır. Ama artık buna ihtiyaç yoktur. Sürüş dersi alacak arabalarınızı akla gelen her tür algılayıcıyla donatır ve bu iş için tuttuğunuz insan sürücülerin kontrolünde yola salarsınız. Bu insanlar arabaları çeşitli senaryolarda sürer. Her sürüşün her saniyesinde o girdi kümesi için insan sürücünün "çıktısı" (yani o durumda ne yaptığı) kayda alınır ve sinir ağı bu girdi/çıktı çiftleriyle gözetimli öğrenmeye tabi tutulur. Arabanızın bu veri kümesini öğrendiğini düşündüğünüzde, onu bu iş için ayrılmış özel sürüş alanınıza götürürsünüz ve antrenmanın ikinci aşaması başlar. Yola fırlayan yayalar, kaykaylı çılgın gençler, hemen önünüzde çarpışan başka arabalar vs. türlü çeşitli nahoş senaryo da bu özel alanda denenerek öğretilir. Bir noktada sürücü elini direksiyondan çeker, kontrol robota geçer. Bir sonraki aşama, robotun (yine şoför koltuğunda gerektiğinde müdahaleye hazır bir insanla birlikte) gerçek trafiğe çıkmasıdır. Özerk arabalar şimdiden gerçek trafikte toplamda milyonlarca kilometre yol kat etmiştir.

Peki gerçekten yalnız başlarına yola çıkmaya hazırlar mı? Tesla marka bir arabanın otoyolda özerk sürüş sırasında önünde giden iki otomobilin kaza yapacağını çarpışmadan önce anlayıp fren yaparak kendini ve sahibini kurtarışını gösteren video çok etkileyicidir. Öte yandan, bu satırları kaleme aldığım günler itibariyle özellikle bazı koşullarda "bisikletli tanıma" probleminin tam çözüle-

mediği konuşuluyordu (Ama bu insanlar için de zor bir problemdir). Aslına bakarsanız bisikletler ve tüm yayalar diğer araçlarca tanınmalarına yarayacak sinyaller yayan vericiler taşımaya mecbur edilseler ve otomobillerin bazıları değil, istisnasız tümü robotlarca sürülse, kazaları engelleyip verimliliği artırmak mühendislik açısından çok daha kolay olacaktır. Ama acaba toplum bunlara hazır mı? Bu konulara döneceğiz.

36 | Sohbet programları nasıl çalışır?

Turing'in ünlü testini önerdiği o ilk makaleden (23. Soru) bu yana, insanlarla insan dillerinde (bilgisayarcı deyimiyle, "doğal dil"de) iletişim kurabilen programların geliştirilmesi yapay zekâ araştırmacılarının en çok uğraştığı işlerden biri oldu. Ben de gençliğimin en güzel yıllarını bu konuya gömdüğümü söyleyebilirim doğrusu.

Doğal dilde yüksek kalitede iletişim problemini çözmek, yapay zekânın tümünü gerçekleştirmeye denk zorlukta bir iştir. Turing'in test olarak tam da bunu belirlemesi boşa değildir. 1964'te bu testi "hileyle" geçmeye yeltenen ELIZA adlı bir bilgisayar programı yazıldı. Bilgisayarı kullanan kişinin girdiği cümleleri basit bir teknikle "tersyüz" ederek yanıt veren, bazen de okuduğu cümlenin içinde geçen kimi kelimelerin tetiklediği kurallara bağlı olarak stokunda bulunan hazır cümleleri söyleyen ELIZA, bazı saf kullanıcıları öylesine başarılı bir şekilde kandırdı ki, yazarı Joseph Weizenbaum insanlığın henüz yapay zekâya hazır olmadığına karar verip bu işlerden elini eteğini çekti. İşte hocalığımın ilk yıllarında öğrencim Çiğdem Aytekin'e ödev olarak yazdırdığım Türkçe konuşan ELIZA'nın bir diyaloğu:

> KULLANICI: Neler düşündüğümü bildiğine inanmak çok zor.

ELIZA: Demek neler düşündüğünü bildiğime inanmak çok zor.
KULLANICI: Her şeyi tekrarlıyorsun.
ELIZA: Demek her şeyi tekrarlıyorum.
KULLANICI: Herhangi bir şey hissediyor musun?
ELIZA: Ben rahat hissediyorum.
KULLANICI: Hissettiğine inanmak ne kadar zor.
ELIZA: Benim neler hissettiğimi düşünüyorsun?
:

Görüldüğü gibi, artık kullanıcılar ELIZA gibi son derece yüzeysel yöntemlerle bir farkındalık yanılsaması yaratmaya çalışan programlara kolay kolay kanmıyor. Bu yüzden, doğal dildeki metinleri önce sözcük, sonra cümle ve giderek anlambilim (semantik) düzeyinde bir dizi çözümlemeden geçiren "ciddi" yazılımlar üretmek gereklidir. "Derin anlama" konusunda varabildiğimiz noktayı sonraki soruya bırakıp günümüzün "sohbot"larının (İngilizce "sohbet" kelimesiyle "robot"un "bot"unun bir araya getirilmesiyle oluşturulmuş "chatbot" sözcüğüne bu karşılığı öneriyorum) nasıl çalıştığına bakalım.

Sohbotları konu açısından dar kapsamlı veya sınırsız olanlar olarak ikiye ayırabiliriz. Dar kapsamlı bir sohbotla kendi konusu dışında konuşursanız sizi anlamaz, cevap veremez, verirse de saçmalayabilir. Kimi dar kapsamlı sohbotlar (yazışma yoluyla çalışıyorlarsa) kullanıcının ne yazacağını bile kısıtlayarak sohbetin raydan çıkmasını engeller. Konuyu bu şekilde çerçevelemek programı yazan kişinin işini kolaylaştırırsa da, kullanıcıda yaratacağı yabancılaşma nedeniyle "doğal dil arayüzü"nden beklenen avantajın büyük kısmının yitirilmesine yol açar.

Sohbotumuza sadece bir mağazanın ürünlerini tanıtmak gibi bir "listeden seçtirme" işi değil de (sözgelimi müşteri destek elemanlığı gibi) daha çok adım gerektiren karmaşık bir görev yükleyeceğiz diyelim. Bu durumda tıpkı insan müşteri temsilcileri gibi sohbotun da önceden hazırlanmış bir "akış diyagramı"nı izlemesi gerekir: Müşteriye önce adını sor, sonra ne istediğini sor,

sonra o isteği gerçekleştirmek için diyagramda belirtilen sıradaki adımla ilişkili soruyu sor, tüm bilgileri topladıysan belirtilen işlemi yapıp uygun cevabı ver. Sorun, gerçek hayatta insanların tam bu sırada ve programcının beklediği biçimde bilgi vereceklerinin hiç de garantili olmamasıdır. İnsanlar bir bilgiyi pek çok değişik şekilde ifade edebilir, bu nedenle de sohbotun (daha önce hiç görmediği şekilde ifade edilmiş olabilecek ve yazım veya konuşma tanıma yanlışları da içerebilecek) çok sayıda farklı girdinin kendi anlam dünyasında hangi kategoriye denk geldiğini anlayıp sınıflandırması gerekir. İşler yolunda gider de kullanıcının dedikleri doğru anlaşılırsa ne âlâ, aksi takdirde program anlamaktan umudu keserse kullanıcıyı bir insan operatöre devredebilir. Kimi sohbotlar bu süreç sırasında kullanıcının sarf ettiği kelimelerden "duygu analizi" yapıp duruma uygun olması umulan "Sıkıldığınız için üzgünüm" gibi sözler edebilir. Gün sonunda konuşma kayıtları incelenir (bilgisayarla konuştuklarınızın sadece ikinizin arasında kalacağını asla varsaymamalısınız) ve kullanıcının ne istediği ve bunun için ne dediği saptanarak bir dahaki sefere sohbotun bu sözleri doğru yorumlaması için gerekli değişiklikler yapılır.

Ben bu kitap üzerinde çalışırken Google insandan ayırt edilemez bir ses ve tarzla insanlarla telefonda "sahibi" adına rezervasyon yapma konusunda konuşur gibi görünen bir programın çok etkileyici birkaç örnek diyalog kaydını yayınladı. Henüz kendim incelemediğim bu sistemi bir "reklam" videosu üzerinden değerlendirmem doğru olmaz, ama örneklerin yukarıdaki sınıflandırmama göre yine dar kapsamlı sohbetler olduklarını vurgulayayım.

Sınırsız bir sohbet ise (insan taklidi kısmı haricinde) tam Turing'in sözünü ettiği şeydir ve mühendislik açısından olağanüstü zordur. Apple şirketinin dijital asistan programı Siri gibi iddialı konuşma sistemleri, görevleri gereği anlayıp karşılayabildikleri işlevsel komutların (Siri birçok kez eliyle telefona uzanamayan insanlar için

ambulans çağırarak hayatlar kurtarmıştır) yanı sıra, kullanıcılarına olabildiğince az defa "Üzgünüm, seni anlayamadım" diyebilmeleri için her gün yeni cevaplarla zenginleştirilmektedir. Bu sohbotların evlenme tekliflerinden küfürlere, din ve siyaset sorularından "yarın sabah intihar etmemi hatırlat" gibi sorunlu isteklere "doğru" ve bir kişiliğe sahipmişçesine kendi içinde tutarlı cevaplar verebilmesi amacıyla birçok şair ve yazar istihdam edilmektedir. Amazon şirketinin Alexa asistanını popüler konular hakkında en az 20 dakika boyunca insanları sıkmayan tutarlı sohbetler gerçekleştirebilecek şekilde programlayacak ekibe vaat ettiği bir milyon dolarlık büyük ödülü alabilen henüz çıkmamıştır.

ELIZA gibi hilelere kaçmadan herhangi bir konuda açılabilecek her sohbete katılabilmek için herhangi bir insanın hakkında konuşabileceği her şeyi bilmeniz gerekir. Eğitimsiz olanlar da dahil olmak üzere tüm insanların dünya hakkındaki ortak bilgilerine "sağduyu" adı verilir. Diyelim ki bildiğiniz bir konu hakkında bir ansiklopedi maddesi yazıyorsunuz. Konunun anlaşılması için gereken, ama "Nasılsa herkes bunu bilir" diyerek yazınızda açıklamadığınız her şey, sağduyunun içindedir. İnsanların birbirleriyle genellikle fazla konuşmadan anlaşabilmesinin de, bazen farklı kültürlerden gelen kişilerin aynı dili konuşmalarına karşın yanlış anlaşmasının da nedeni, ortak olduğunu varsaydıkları bu bilgi kümelerindeki benzerlik ve farklılıklardır. Eğer programımızın okuduğu veya duyduğu bir metinden normal insanların ne anlam çıkaracağını anlamasını istiyorsak, bir şekilde bu bilgilere sahip olması gerekir.

Eski moda yapay zekâcılar bu probleme "O zaman tüm bu bilgileri teker teker ve açık açık yazalım" diye yaklaştı. Yaklaşık yüz milyon cümlelik bir bilgi tabanının amaca ulaşmak için yeterli olacağı tahmin ediliyordu. Bu devasa işi gerçekleştirmek (ve yazdıkları program parçasını diğer yapay zekâcılara satarak kâra geçmek) isteyen grubun başını 30. Soru'da tanıştığımız Douglas Lenat çekti. 1984'ten beri "İnsanlar önce doğar, sonra ölür",

"Bardağın açık tarafı üstte tutulur" vs. türünden milyonlarca bilgiyi kodlayarak insan sağduyusunu bilgisayarlara kazandırmayı amaçlayan CYC projesi üzerinde çalışıyor. Yeni moda YZ'cilerin çoğu ona bir tür Don Kişot gözüyle bakarak doğal olanın bu bilgilerin artık elimizin altında olan büyük veriden otomatik öğrenilmesi olacağını söylüyorlar. Son gülen kim olacak, göreceğiz. Daha anlayışlı programlar üretme çabasının öyküsü önümüzdeki soruda sürecek.

37 | Bilgisayarlar insan dillerini nasıl anlar?

Türkçe doğal dil işleme, güzelim "bilgisayar" sözcüğü de dahil olmak üzere bu kitapta da kullandığımız nice bilişim terimini dilimize kazandıran Aydın Köksal Hoca'nın doktora tezinden bu yana Türk bilgisayar mühendislerinin ilgisini çeken alanlardan biri olmuştur. 1992'de başka bir yapay zekâ problemi üzerine yazdığım tezimi bitirince ben de bu konuya eğildim. Türkçenin sözcük yapısı ve grameri üzerine çalışmalar yürütülüyordu, ama daha derin düzeyde "anlayan" bir program ortalıkta yoktu. Bu eksikliği gidermek üzere kolları sıvadım. Esin kaynağım, Gordon Novak'ın Texas Üniversitesi'ndeki doktora tez çalışması olan ISAAC programı oldu. Novak ABD'de okutulan birkaç lise ve yüksek okul fizik ders kitabından 20 adet problem seçip o soruları çözen öğrencilerin geçtiği tüm aşamalardan (cisimlerle ilgili kavramları kazanma, İngilizce, matematik ve fizik öğrenme) geçerek çözebilen bir sistem kurmuştu. Ben de benzerini Türkçe için yapmaya niyetliydim.

Size 10 yaşındaki öğrencilere özenen ALİ'yi tanıtmak istiyorum. ALİ ("Aritmetikçi - Lisan İşleyici"), Prolog ve Pascal dillerinde yazılmış her biri bin küsur satırlık çok sayıda programın birbirini zincirleme şekilde

çalıştırmasıyla "hayat" buluyordu. 1993'te ALİ'yi yazmaya başlamadan önce kitapçıdan rasgele bir ilkokul üçüncü sınıf matematik kitabı satın aldım. Hedef problem kümemi bu kitaptaki (resim veya diyagram içermeyen) soruların arasından seçtim. İşte ALİ'nin sınav sorularından bazıları:

> "Bir çiftlikte 255 koyun, 67 kuzu, 8 inek vardır. Bunların hepsi kaç hayvan eder?"
>
> "Bir fabrikada 217 işçi vardı. 15 işçi çıktı. 7 işçi emekli oldu. Fabrikada kaç işçi kalır?"
>
> "Bir perde 5 m bezden yapılıyor. 9 perde kaç metre bezden yapılır?"
>
> "Bir kamyonun deposunda 67 lt mazot vardı. Şoför 145 lt daha aldı. Kamyondaki mazotun hepsi kaç litre olur?"

Basit mi görünüyor? Hayatının bir buçuk yılını bu soruları "anlayıp" çözen programı yazmaya harcamış birisi olarak kesinlikle aynı fikirde değilim. Uzmanlık alanı (aritmetik) ISAAC'inkinden daha kolay görünse bile, ALİ bilgisayarla işlenmesi bazı açılardan daha zor bir dil olan Türkçe'yle boğuşmak zorundaydı.

Türkçe'nin ek zorluğu bitişimliliğinden kaynaklanıyordu. İngilizce'de ayrı sözcükler olan edatlar, iyelik bildirimleri vb. sözdizimsel öğeler, Türkçe'de kökün ardından dizilen ekler olarak sözcüklerin içine saklanır. Bir Türkçe metinden rasgele bir sözcüğü (örneğin "sözcüğü" sözcüğünü) sözlükte bulma ihtimaliniz İngilizce'ye oranla çok azdır; önce ekleri sökmeniz ve (örneğimizde olduğu gibi) kökte yaratmış olabilecekleri değişiklikleri geri almanız gerekir. Ancak bu "biçimbirimsel çözümleme" işinden sonra kökün ne anlama geldiğini öğrenebilmek için sözlüğü kullanabilirsiniz. ALİ ilk aşamada 20.000 kelimelik bir elektronik sözlükten de yararlanarak çözmeye çalıştığı problemdeki tüm sözcükleri biçimbirimlerine ayırıyordu.

Her cümlenin içinde kimi sözcükler kendi aralarında

anlamlı gruplar oluşturur. Bu gruplaşmalar (örneğin yukarıdaki problemlerin sonuncusunda "bir kamyonun deposunda" ve "67 lt mazot" öbekleri) cümlenin anlamının bilgisayarca "resmedilmesinde" kullanılacakları için, ALİ'nin ikinci aşaması programın içindeki Türkçe gramerinden yararlanarak metindeki her cümleyi bu özelliklerine göre ayrıştıran "sözdizimsel çözümleme" işlemini gerçekleştiriyordu.

Ve artık anlamlara geliyoruz. Son aşamada ALİ sırayla her cümleyi "anlamaya" ve gereğini yapmaya çalışıyordu. Kamyondan söz eden örneğimizde, ilk cümlenin çözümlenmesi sonucunda kısa vadeli belleğe (yani programın bağlamına) "DEPO, KAMYON'un bir parçasıdır" ve "DEPO'da {67 LT MAZOT} vardı" bilgileri eklenir. İkinci cümlede bir şeyden 145 lt daha alındığından söz edilmektedir, ama nedir bu şey? Acaba yazarın aklından o sırada ne geçiyordu? ALİ bu bilmeceyi çözmek için bağlamı inceleyerek "lt" adlı birimle ölçülen şeylerden (su, süt, mazot, vs.) en son hangisinin eklendiğine bakar ve yazarın mazotu kastettiğini ortaya çıkarır. Böylece ikinci cümle de belleğe yeni mazot alımıyla ilgili bilgiyi eklemiş olur. Son cümle, "kaç" kelimesinden ve soru işaretinden anlaşıldığı gibi, bir yanıt vermeyi gerekmektedir: Kamyondaki mazot miktarı hesaplanacaktır. İşte burada, bilgisayarlara bir şeyler öğretmenin gerçekten çok sabır istediğini gösteren bir ayrıntı karşımıza çıkar: Metnimizde şimdiye dek sadece **depo**nun içinde mazot olduğu söylenmiştir; **kamyon**un değil! Biz insanlar böyle durumlarda "Eğer A, B'nin bir parçasıysa, B, A'nın içerdiği her şeyi içerir" anlamına gelen bir kuralı kullanırız. Tabii ki bu ve benzeri tüm sağduyu bilgilerinin programa açıkça yazılması gerekir. ALİ bu cümlenin işlenmesinin sonucunda (gereken aritmetik işlemin toplama olduğunu çıkarsayıp "lt" ile "litre"nin aynı şey olduğu bilgisini de kullanarak) yanıtını veriyordu: "Kamyondaki mazotun hepsi 212 litre olur."

ALİ'den sonra öğrencilerim Özlem Çetinoğlu, Fatih Öğün ve Şeniz Demir ile TOY ("Türkçe Okur Yazar")

adlı "anlayışlı" bir sohbet programları altyapısı geliştirdik. TOY kendisine söylenenlerden mantıksal çıkarımlar yapabiliyor, mesela önceden "Her anne güzeldir" cümlesini okumuşsa, daha sonra şöyle bir diyaloga girebiliyordu:

KULLANICI: Ayşe bir anadır.
TOY: Teşekkürler, öğrendim.
KULLANICI: Ayşe güzel midir?
TOY: Evet.
KULLANICI: Neden?
TOY: Her ana güzeldir. Ayşe anadır.

ALİ ve TOY her ne kadar Türkçe için eşsiz olsalar ve çoğu sohbet programından daha derin bir anlayışa sahip gibi görünseler de hem bizim kodlamaya mecalimizin yettiği kadarından öte bir sağduyu altyapıları olmaması, hem de geçtiğimiz sayfalarda söz ettiğimiz kırılganlık sorunu nedeniyle (tıpkı ISAAC gibi) gündelik kullanıma açılabilecek esneklikte değillerdi. Gramerinde azıcık bozukluk olan ama insanların sökebildiği cümlelerden hiçbir şey anlayamıyorlar, "anladıkları" sözcükleri de biz insanların anladığı derinlikte değil, az sayıda bağlantısı olan simgeler olarak temsil ediyorlardı (Örneğin ALİ'nin "mazot"un nasıl bir şey olduğu konusunda litreyle ölçülüp "depo" diye bir yere konulabildiği ve alınıp satılabildiği dışında bir fikri yoktu). Bu bağlamda eski moda YZ örnekleriydiler. Gelin bu anlama işine yeni çağda nasıl yaklaşıldığına bakalım.

IBM şirketinin konuşma tanıma (ses sinyalini metne dönüştürme) programları geliştiren ekibini yıllarca yöneten Fred Jelinek'in "Ne zaman bir dilbilimci kovsam programın tanıma başarısı yükseliyor" dediği söylenir. Dilbilimciler burada dili kurallara, formüllere hapsetmeye çalışan eski moda yaklaşımı temsil etmektedir. Oysa büyük veri çağında artık elimizde geçmişten farklı olarak üzerinde çalışıp sonuçlara varabileceğimiz gerçek "dil bilgisi" bulunmaktadır.

Anlamları temsil etmenin harika bir yolundan söz ede-

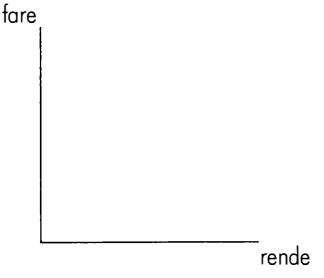
ceğim. Çok büyük bir Türkçe dokümanlar kümemiz olsun (Mesela Google'ın erişebildiği tüm Türkçe İnternet sayfalarını düşünün). Bu küme üzerinde istatistik yaparak hangi kelimelerin birbirlerine anlamca yakın, hangilerininse uzak olduğunu ölçeceğiz.

Basit bir ilkeye dayanacağız: Birbiriyle ilişkili iki kelimenin aynı cümlelerde bir arada geçme ihtimali, alakasız iki kelimenin bir arada görülme ihtimalinden yüksektir. Her kelime çiftinin kümemizdeki cümlelerde kaç kez bir arada geçtiğini bilgisayara saydıracağız. Bu sayıları "anlamca yakın/uzak" kavramına matematiksel bir temel sağlamak için tanımlayacağımız yeni bir uzayda kullanacağız.

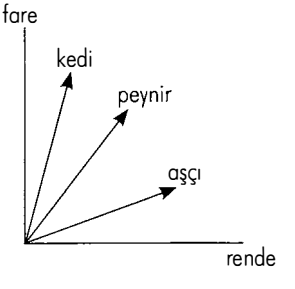
İki boyutlu uzayı liseden hatırlarsınız. Matematik derslerinde genelde yatay eksene x, dikey eksene de y adını verdiğimiz bu uzayda herhangi bir noktanın konumunu iki eksen üzerinden birer sayı (koordinat) belirtilerek ifade edebilir, eksenlerin ortak sıfır noktasından başlayıp bu noktada sonlanan bir ok şeklinde gösterilen doğru parçalarına da "vektör" derdik ya? Bu iki eksene de dik üçüncü bir eksen (z ekseni) eklendiğinde içinde yaşadığımız üç boyutlu uzay elde edilir. Üç boyutlu uzayda bir vektörü tarif etmek için iki değil üç koordinat gerekir. Ama bununla yetinmek zorunda değiliz. Matematikçiler istedikleri sayıda boyutta uzaylar tarif edip kullanabilirler. Örneğin gerekirse 100 boyutlu bir uzayda 100'er sayı tarafından temsil edilen vektörler yazıp bunları birbirleriyle toplayabilir, çıkarabilir veya aralarındaki açıları hesaplayabilirler. Anlam gösterimimiz böyle bir altyapı üzerine kurulacak, ama biz normal insanlar çok boyutlu uzayları gözümüzde canlandıramadığımız için örnek verirken az boyutlu bir uzay kullanacağım.

Türkçe sözlükteki her bir kelime için ayrı bir boyuta sahip olan on binlerce boyutlu bir uzay düşünün. Her kelime için bu devasa uzayda bir vektör çizeceğiz. Her kelimenin vektörünü belirleyen sayılar, eksenleri belirleyen diğer kelimelerle aynı cümlede geçme sayıları olacak. İşimizi kolaylaştırmak için bu uzayın sadece iki kelimeye

denk gelen iki eksenine odaklanalım, böylece örneklerimizi kâğıda çizebileceğiz. Mesela "rende" ve "fare" eksenlerinin oluşturduğu düzleme bakalım.



"Kedi" kelimesinin bu devasa uzaydaki vektörünün örneğimizdeki "rende"-"fare" düzlemindeki izdüşümü (yani başka hiç boyut yokmuşçasına sadece bu iki kelime cinsinden çizilen "kedi" vektörü) "rende" ekseninden uzak, ama "fare" eksenine yakın olacaktır. Neden mi? "Kedi" kelimesi "rende" ile az, "fare" ileyse çok daha fazla cümlede bir arada geçer de ondan! (Tabii bu kitapta bu cümleleri kullanarak istatistikleri biraz değiştirmiş oluyorum. Dil müthiş bir şey.) Aynı mantığa göre "aşçı" ve "peynir" vektörlerini de ekleyelim.



Artık "anlamca yakın" kavramını matematiksel olarak tanımlayabiliyoruz. İki kelimenin bu uzaydaki vektörleri arasındaki açı ne kadar azsa o kadar yakın kavramları temsil ederler (Dikkat ederseniz "fare" ve "rende"nin vek-

törlerini çizmedik ve tanımımıza göre bir kelimenin eksen-iyle vektörü tam aynı doğrultuda olmak zorunda değil, ama örneğimizdeki yakınlıkların zihnimizde kolayca canlandığını umuyorum).

Dahası var. Son yıllarda bu anlam vektörleri için çok daha az boyutlu uzayların da kullanılabileceği keşfedildi. Örneğin İngilizce için yapılan bir çalışmada her kelime için vektörler, yine cümledeki komşuluk ilişkilerine göre yakın kelimelerin vektörleri birbirlerine yaklaşacak, uzak olanlarınkiler ise uzaklaşacak şekilde sinir ağları yardımıyla bin boyutlu bir uzayda döndürülerek konumlandırıldı. Sonuçta elde edilen vektörler arasında "mana âlemi"ndeki birçok ilişkinin bulunduğu görüldü: "Kral" vektöründen "erkek" vektörünü çıkardığınızda bulduğunuz fark vektörüne "kadın" vektörünü eklediğinizde elde ettiğiniz vektör, diğer kelime vektörlerinden en çok hangisine yakın çıkıyor, tahmin edin. Evet, "kraliçe"ye! Aynı şekilde "amca" ile "hala" arasında da gidilebiliyor. Yani bu uzayda öyle bir yön var ki, o yöne gitmek anlamı erkekten dişiye doğru değiştiriyor. Tekillerle çoğullar, ülkelerin isimleriyle başkentlerinin adları vs. birçok ilişki bu şekilde "keşfedildi".

Peki ama, bu çok daha zengin gösterimle bile bilgisayarın dili insanlar gibi anladığı söylenebilir mi? Bu felsefi tartışmaya 47. Soru'da girişeceğiz. Ama önce bilgisayarlar bir dilden diğerine nasıl çeviri yapabilir, onu görelim.

38 | Bilgisayarlar nasıl çeviri yapar?

26 Ağustos 2013'te *Yeni Şafak* gazetesinde dilbilimci ve filozof Noam Chomsky ile yapıldığı söylenen bir röportaj yayımlandı. Ortadoğu siyasetine ilişkin röportajda Chomsky'ye ait olduğu iddia edilen ifadeler, ünlü düşünürün takipçilerini şaşırtacak sığlıktaydı. Durumdan

haberdar edilen Chomsky, Facebook sayfasından yaptığı açıklamada *Yeni Şafak*'ın sorularına verdiği yanıtlara sadık kalınmadan çeviri yapıldığını ve söylemediği şeylerin röportajda yer aldığını, gerçek sorularla cevapları da paylaşarak ortaya koydu. 31 Ağustos'ta söz sırası yine *Yeni Şafak*'taydı.

Üç cümlenin kendilerince uydurulduğunu kabul eden gazete, Chomsky'yi röportajcı Burcu Bulut'un kendisine gönderdiği ikinci bir grup soruyu unutmakla suçlayıp sahte olduğunu hâlâ reddettiği cevapların "Chomsky tarafından gönderilmiş İngilizce orijinallerini" de internette yayımlayarak tartışmaya "son noktayı" koymayı denedi. Rezalet de işte bu noktada arşa çıktı.

Dünyanın en ünlü dilbilimcisi Chomsky'nin özgün İngilizcesi olarak sunulan metinler, deli saçması bir kelime çorbasından ibaretti. En komik örnek, 26 Ağustos'ta yayımlanan "Ortadoğu'daki bu karmaşıklığın, kaos ortamının Batılı devletleri telaşlandırdığını mı sanıyorsunuz? Aksine ne zaman ki her şey süt liman olur, düzene girer işte o zaman Batı'da telaş başlar" cümlelerinin orijinali olarak "This complexity in the Middle East, do you think the Western states flapping because of this chaos? Contrary to what happens when everything that milk port, enters the work order, then begins to bustle in the West." metninin verilmesiydi. Chomsky'nin Türkçedeki "sütliman" kelimesinin yanlış şekilde "süt" ve "liman" olarak yazılmış halinin karşılığı olan ve İngilizce bir anlamı bulunmayan "milk port" lafını etmeyeceği ortadaydı. Besbelli ki baştan uydurulmuş olan Türkçe metin o yıllarda kalitesi şimdikinin çok altında olan Google Çeviri programıyla İngilizceye çevrilerek sahte çevirinin sahte orijinali elde edilmeye çalışılmış, gazetede kimsenin İngilizcesi de sonucun ne kadar berbat olduğunu anlayıp dur demeye yetmemişti.

Bir bilgisayara bir dilden diğerine çeviri yaptırmanın birkaç farklı yöntemi düşünülebilir. Bunların en basiti, kaynak metindeki sözcükleri teker teker ele alıp karşılıklarını hedef metne sırayla ekleyerek çeviriyi üretme algoritmasıdır. Bu "kelime tabanlı" yöntem sadece Türkçe-

Azerice çifti gibi sözdizimleri birbirinin aynısı olan çok yakın akraba dillerde işe yarayabilir, diller arasındaki mesafe arttıkça ise kullanışlılığı hızla düşer.

Vaktiyle (eski moda YZ'ci olduğum yıllarda) benim de üstünde çalıştığım bir başka yaklaşım ise "sözdizim tabanlı çeviri" yöntemiydi. Buradaki anafikir şuydu: Her dilin "düzgün" cümlelerini oluşturmak için uyulması gereken, hangi öğenin önce gelmesi gerektiğinden fiillerin çekimine dek bir dizi koşulu içeren "gramer" dediğimiz bir kurallar kümesi vardır. Bu kuralların, sözgelimi Türkçe ve İngilizce gramerlerinin programlandığı bir bilgisayar, verilen düzgün bir Türkçe cümleyi, örneğin "Ayşe okula gidiyor" metnini çözümleyip söz konusu eylemin "gitmek", gidenin "Ayşe", gidilenin de "okul" olduğunu çıkarabilir, sonra da aynı yapıyı İngilizce gramerine göre kurup sözlükten baktığı karşılıkları uygun yerlere yerleştirerek "Ayşe is going to school" çevirisini üretmeyi başarabilir.

İlk bakışta iyi bir fikir gibi görünse de, bu yaklaşım önceki soruda da gündeme getirdiğimiz iki nedenle çuvalladı. İlki kırılganlıktı; insanlar arası iletişimde cümlelerin gramere uygun olması gerekmiyor, "bozuk" cümleler veya hiç cümle olmayan kelime dizileri pekâlâ karşı tarafça anlaşılabiliyorken bilgisayar sözdiziminde en ufak hata gördüğünde takılıp kalıyordu. Esas sorun ise, bir dildeki bir kelimenin diğer dilde farklı anlamlarda birçok karşılığı olduğunda bilgisayara doğru seçeneği buldurmak için sadece gramerin yetmeyeceği gerçeğiydi.

Google Çeviri'nin 2006'daki çıkışından 2016 güzüne dek kullandığı "öbek tabanlı istatistiksel çeviri" algoritması, bir sözcüğün anlamının sadece o sözcüğe bakarak saptanamayacağını dikkate alıyor ve işe kaynak dilde verilen cümleyi (gramer kurallarına çok aldırmadan) birkaç kelimelik öbeklere bölerek başlıyordu. Google'ın erişebildiği devasa miktardaki verinin içinde aynı metnin iki dilde birden yazılı olduğu çiftler üzerinden geçerek hazırlanmış çeviri tabloları, böyle öbeklerin hedef dildeki farklı karşılıkları arasında istatistiksel olarak en ağır basanın seçilmesi için kullanılıyordu. Dillerin sözdizimleri

arasındaki farklılıkların yarattığı sorunlar da her dil çifti için mühendislerin ayrıca geliştirip sürekli iyileştirmeye çalıştığı özel programlar tarafından bu öbek çevirilerinin yerlerinin düzenlenip son rötuşların yapılması ile aşılmaya çalışılıyordu. Sonuç, "milk port" faciasının gösterdiği gibi, pek de iyi olmuyordu. Uzunca bir çeviriye baktığınızda, düzgün cümlelerle değil, zekâdan yoksun bir öbek çorbasıyla karşı karşıya kalıyordunuz.

Google Çeviri 2016'da tümüyle sinir ağlarına teslim edildi. Derin bir sinir ağının resimlerdeki kişileri tanıma işini nasıl katmanlara bölerek yapabildiğini 29. Soru'da anlatmıştım. Ağ mimarisinde dilin dizisel doğasına uygun birkaç değişiklik, aynı kademeli işlemenin harf dizileri şeklinde verilen girdi cümlelerden anlam vektörleri elde etmek için de yapılabilmesini sağlar. Şimdi iki dil arasında çeviri yapmayı sinir ağına öğretebiliriz.

Çok sayıda doğru çevrilmiş cümle çifti toplayarak başlayın (Kanada parlamentosunun hem İngilizce, hem de Fransızca yazılan tutanakları ünlü bir örnektir). Sinir ağının eğitiminde girdi/çıktı çiftleri olarak bunları kullanın. Mükemmel olduğu hâlâ iddia edilemese de, Google Çeviri'nin kalitesinin 2016'da bir seferde önceki on yıldaki artıştan çok daha fazla iyileştiğinin ölçülmesi, derin öğrenmenin bu iş için de hatırı sayılır bir yöntem olduğunu göstermektedir.

Yeterince derin bir ağı yeterince çeviri örneğiyle yeterli süre eğitirseniz, ağın ortalarında bir katmandaki sinir hücreleri girdi cümlenin kaynak dilin uzayındaki anlamını, sonraki bir katmandakiler de aynı anlamın hedef dilin uzayındaki karşılığını önceki soruda sözünü ettiğimiz vektörlere benzer sayı grupları şeklinde hesaplamayı öğrenecektir. Daha sonraki katmanlar da o anlamın adım adım hedef dildeki çıktı cümlesine dönüşüm sürecini izleyecektir.

Yani makinenin çeviri sırasında yaptığı, hedef dilin uzayında kaynak dildeki anlam vektörüne en benzeyen vektörü bulmaya çalışmak olarak görülebilir. Google mühendisleri bu anlam gösterimi sayesinde her dil çifti için

ayrı bir çalışma yapmak yerine "sıfır örnekli çeviri"lerin de yapılabildiğini ortaya koymuştur: Hem İngilizce-Japonca hem de İngilizce-Korece çevirinin öğretildiği bir sinir ağı, kendisine hiç gösterilmemiş olan Japonca-Korece çiftini başarıyla tercüme edebilmiştir.

Tabii "başarı" derken programın eski acıklı durumunu akılda tutarak konuşuyorum. Mükemmel çeviri tam anlamaya denk zorlukta bir iş. Mevcut teknolojinin tıkandığı kimi örnekleri 43. Soru'nun yanıtında anlatacağım.

39 | Google nasıl gezegen keşfetti?

Gökyüzünde yıldızların yerleri Dünya'nın dönüşü nedeniyle değişiyor gibi görünürse de kendi aralarında oluşturdukları şekiller (insan ölçeğindeki sürelerde) değişmez, hatta bu şekillere anlamlar atfeden boş inanışlar evrilmiştir. Gökteki bu sabit desenin beş istisnası vardır. Eski çağlardan beri beş yıldızın ötekilerden bağımsız olarak gökyüzünde "gezdiği" gözlenmektedir. Artık bu beş nesnenin (Merkür, Venüs, Mars, Jüpiter, Satürn) yıldız değil, bizim yıldızımız Güneş'in çevresinde (tıpkı Dünya'mız gibi) dönen, kendisi parlamayan, Güneş'in ışığını yansıttıkları için görebildiğimiz cisimler olduğunu biliyor ve onlara "gezegen" diyoruz.

Güneş Sistemi'nde Dünya dahil altı gezegen olduğunu sanan eski bilimcilerden kimileri bu sayının gizli bir anlamı olduğunu düşünse de, artık bunun da doğru olmadığını biliyoruz. Teleskopun icadından sonra Güneş'in çevresinde dönen yüz binlerce gökcismi keşfedildi. Bu kadar cisme isim verip ilkokulda çocuklara ezberletmek mümkün olmadığından "gezegen" kavramının "resmi" tanımı (kimi cisimlere gezegenlik unvanı önce verilip sonra alınacak şekilde) birkaç kez değiştirildi. Uluslararası Astronomi Birliği Genel Kurulu'nun 2006'da yaptığı son

tanıma göre Güneş'in çevresinde dönen ve kütlesi onu yuvarlak hale getirecek ve izlediği yörüngeyi o yöredeki başka cisimleri kendi üstüne düşürerek, uydu olarak yakalayarak ya da uzağa fırlatarak "temizleyecek" kadar büyük olan cisimlere "gezegen" diyoruz. Bu hesaba göre sistemimizde sekiz gezegen biliyoruz.

Tabii özellikle Güneş'ten söz eden bu tanıma bakarsanız, evrenin başka yerlerinde hiç gezegen yok! Oysa artık başka yıldızların çevresinde de böyle gökcisimleri olduğunu biliyoruz. Bunlara "ötegezegen" deniliyor.

Ötegezegenler kendi yıldızımızın çevresindeki gezegenler gibi gökyüzünde gezinen ışık noktaları arayarak bulunamıyor. Başka yıldızlar çok uzaktalar, gezegenleri de buradan bakıldığında onların o kadar yakınında dönüyor ve ışıkları yıldızınkine oranla o denli sönük ki, mevcut teknolojimizle neredeyse hiçbir ötegezegeni yıldızından ayrı görüntüleyemiyoruz. Peki nasıl bulunabiliyorlar?

Bu iş için tasarlanmış özel uzay teleskopları var. Bunların en ünlüsü NASA'nın Kepler uydusu. Uydu yıllarca gökyüzünün on binlerce fotoğrafını çekiyor. Aynı yıldızın art arda çekilen birçok görüntüsündeki parlaklık değerleri inceleniyor. Eğer yıldızın ışığı hep aynı parlaklıktaysa bir şey diyemiyoruz. Ama belli aralarla kısa süre azalıp tekrar eski değerine dönüyorsa iş ilginçleşiyor.

Eğer yıldızın bir gezegeni varsa, bu gezegenin yörüngesi de onu Dünya'dan bakıldığında yıldızının önünden geçirip ışığını kısmen kestirecek şekilde konumlanmışsa yıldız ışığında böyle bir azalma ölçülebilir. Ötegezegen kâşifleri bu parlaklık değişimlerini izliyor. İşleri hiç kolay değil, çünkü başka sebepler de böyle değişimlere yol açabiliyor: Kimi durumlarda iki yıldız birbiriyle dans edercesine dönüyorlar, bunlara "ikili yıldız" deniyor. Bir yıldız diğerinin önünden geçerken toplam parlaklık azalıyor. Ya da teleskopa kozmik ışın parçacıkları isabet ederek ölçülen parlaklık değerlerini bozabiliyor. Gökbilimciler bu gibi olayların yarattığı örüntüleri gerçek gezegen geçişlerininkinden ayırt eden özellikleri saptamış, sinyalde bunlara dikkat ediyorlar.

Kepler verisinden şimdiden binlerce ötegezegen keşfedilmiş durumda, ama bu buzdağının görünen kısmı. 200.000'i aşkın yıldıza ilişkin 14 milyar dolayında veri noktası birikti ve insanların bu veri deryasını elemesi çok güç. Uzmanlar parlaklık azalması olaylarını otomatik saptayan bir yazılım kullanıyor, bunlardan da sinyal kalitesi belli bir düzeyin üzerinde olanlara odaklanıyordu. Bu şekilde 30.000 sinyal "elle" incelenip yaklaşık 2500 tanesinin gerçek ötegezegenler olduğu sonucuna varılmıştı.

Google şirketinin ilginç bir geleneği var: Çalışanlar, zamanlarının %20'sinde ne yapacaklarını kendileri belirliyor. Chris Shallue adında bir mühendis kendi %20 'sinde gözardı edilmiş sinyallerin içinde ötegezegen arayan bir yapay sinir ağı kurmaya karar verdi. Uzmanlarca "gezegen"/"değil" diye etiketlenmiş 15.000 sinyali, sinir ağının eğitim kümesi olarak kullandı ve şansını zaten iki veya daha çok gezegeni olduğu insanlarca saptanmış 670 yıldızın ışıklarından kaynaklanan diğer sinyaller üzerinde denedi. Ağın "gezegen" diye nitelediği sinyaller tekrar insan uzmanlara sunuldu ve bunlardan ikisinin gerçek ötegezegenlerden kaynaklandığı onaylandı.

Bu gezegenlerden biri, daha önce yedi başka gezegeni keşfedilmiş olan Kepler-90 yıldızının yörüngesindeydi. 14 Aralık 2017'ye dek sekiz gezegeni olduğu bilinen tek yıldız Güneş'ti. Yıldızımız bu unvanını o günden bu yana 2545 ışık yılı uzaklıktaki Kepler-90 ile paylaşıyor. Yapay zekâ sayesinde.

40 | Bilgisayarlar bizi bizden iyi tanıyabilir mi?

İnsanlığın kaydettiği bilgi miktarı baş döndürücü bir hızla artıyor. Yeni bilgilerin neredeyse tümünü dijital ortamda, bilgisayarların okuyabileceği biçemde üretiyoruz;

eskilerin, sözgelimi kitaplarda duranların çoğunu da dijitalleştiriyoruz. Yaptığınız alışverişler, Google aramalarınız, iletişiminizin tümü, cebinizde telefonunuzla saat kaçta nereye kaç adım atarak gittiğiniz, bir ağ sayfasını incelerken imleci ekranın neresinde kaç saniye tuttuğunuz, tüm sağlık bilgileriniz, hepsi kayıt altında. Bunlara pek yakında buzdolabının kapısını kaç saniye açık tuttuğunuz, duşta kaç dakika kaldığınız, o gün otomobil kullanırken kaç hata yaptığınız vs. de eklenecek. Bu veri deryasında yapay öğrenme teknikleri kullanarak kişilerin gelecekteki davranışlarını veya yakın gelecekte kalp krizi geçirip geçirmeyeceklerini tahmin etmek, daha önce kimsenin fark etmediği bağlantıları keşfetmek, seçim kazanmak, ya da ürününüzü pazarlarken hangi kelimeleri kullanırsanız en çok satış yapabileceğinizi kestirmek mümkün. İstatistik bilimiyle büyük veri işleme gücünü birleştiren "veri bilimi"nin dünyasına hoş geldiniz.

Bir insanın nasıl davranacağını önceden tahmin etme hedefi, ilk duyuşta genel yapay zekâ problemini çözmek kadar zor gelebilir. Ama burada insanların kafalarının içindeki karmaşıklığın tümünü benzetimlemekten söz etmiyoruz. Esasen birkaç faktöre dayanılarak verilen karar türleri için o faktörlerin tam olarak hangileri olduğunu çok sayıda gözlemden anlayıp benzer koşullardaki başka insanlar için olabildiğince az hatalı öngörülerde bulunmayı hedefliyoruz. Sonuçta bu bir istatistik problemidir ve elimizde birkaç yıl öncesine dek istatistikçilerin rüyalarında bile göremediği miktarda veri mevcut.

Örneğin bir mağazanın ağ sitesi (daha önceki bir ziyaretinizde bilgisayarınıza bıraktığı "çerez"leri ve kullanıcı davranışları hakkında topladığı istatistikleri kullanarak) "alıcı" mı yoksa sadece "bakıcı" mı olduğunuza karar verip size ona göre davranabilir; ekrana niyetinizin ciddi olduğunu anladıysa beğeneceğinizi tahmin ettiği bir ürünün sayfasını, size bir şey satma ümidi yoksa da en azından bağlantı tıklatma parasını alabilmek için rakip bir mağazanın reklamını çıkartabilir. Bıraktıkları veri izlerinden insanların zengin mi fakir mi, evli mi bekâr mı, çocuklu

mu çocuksuz mu oldukları anlaşılabilir. Alışverişlerinizin türleri ve mekânlarıyla günlük hareketlerinizden eşinizi aldatmakta olup olmadığınız tüyler ürpertici bir doğruluk oranıyla kestirilebilir.

Veri bilimcilerin "keşifsel çözümleme" dedikleri, henüz verinin içinde ne aradıklarını bilmeden gezinme aşamasındayken kullandıkları tekniklerden biri de "ayrıksılık saptama"dır: Bir grup ak koyunun içindeki tek kara koyun ister istemez gözünüze çarpar. Daha önceden koyunları renk cinsinden sınıflandırmak aklınızda değildiyse bile gündeminize girmiş olur. Buna benzer şekilde verinin içinde doğal öbeklenmeleri otomatik olarak keşfeden, böylelikle de bu öbeklerin dışında kalan kayıtları dikkatimize sunan algoritmalar kredi kartı sahteciliklerinin saptanmasından Güneş yüzeyinde beliren lekelerin hangilerinin normal, hangilerininse şaşırtıcı olduğunun anlaşılmasına dek birçok uygulamada işe yararlar.

Yazarların önceki kitaplarında kullandıkları stil (hangi kelimeleri sıklıkla bir arada kullandıkları, paragraf uzunlukları, vb.) makinece öğrenilip yeni bir kitabı aynı kişinin yazıp yazmadığı yüksek başarı ihtimaliyle söylenebilir, örneğin takma adla yazılan *Guguk Kuşu* (*The Cuckoo's Calling*) romanının Harry Potter serisinin yazarı J. K. Rowling'e ait olduğu bu şekilde anlaşılmıştır.

Satıcılar veya film platformu Netflix gibi siteler, tüketim örüntüleri sizinkine benzeyen diğer çok sayıda müşteriye benzediğinizi, bu nedenle onların beğendiği diğer şeyleri sizin de beğenebileceğinizi öngörerek size önerilerde bulunur. Siz ilk bakışta bir filmi sevmeyeceğinizi düşünebilirsiniz, ama Netflix sizin gibi pek çok kişinin o filmi izleyince sevdiğini görmüştür, bu anlamda sizin ne düşüneceğinizi sizden iyi bilebilir. Amaçları sitede mümkün olduğunca çok vakit geçirip reklam izlemeniz olan YouTube ve Facebook gibi servisler, şimdi izlemekte olduğunuzun ardından size önerecekleri yeni içerikleri aynı yollardan geçmiş milyonlarca diğer kullanıcıdan hangilerinin en uzun süre bağlandığına ve neyi ne sırada izlediğine göre eğitilen algoritmalarla belirler.

Sosyal medyadaki eylemlerimizin hakkımızda saçtığı verinin önemi, 2016'da Birleşik Krallık'ın Avrupa Birliği'nden çıkışı ve ABD'deki başkan seçimi oylamalarının seçmenler hakkında bu şekilde edinilmiş bilgiye dayalı kampanyalar yürüten taraflarca sürpriz şekilde kazanılmasıyla gündeme geldi. İnsanları yeni deneyimlere açıklık, mükemmeliyetçilik, dışadönüklük, işbirliğine açıklık ve kolay üzülme boyutlarında konumlayarak kişiliklerine göre sınıflandırmak için kullanılan bir testi Facebook'ta insanların gönüllü şekilde doldurdukları anketlerin içine gömen psikologlar, ilk kez elde edebildikleri bu dev hacimli veriyi aynı deneklerin profillerindeki diğer açık bilgiler ve "beğeni" etiketi koydukları paylaşımlarla bağlantılandırmayı başarmıştı. Bu eşleştirme sayesinde diğer Facebook kullanıcılarının sadece neleri beğendiğini girdi olarak alıp çıktı olarak bu kişilerin birçok özelliğini tahmin edebilen bir sistem geliştirilmişti. Bir kullanıcının ırkını %95, cinsel yönelimini %88, siyasi parti tercihini de %85 doğrulukla tahmin edebilmeniz için onun sadece 68 beğenisine bakmak yeterli oluyordu. Aynı yöntem mercek altındaki kişilerin zekâ seviyelerini, dinlerini, alkol ve sigara kullanıp kullanmadıklarını, ebeveynlerinin boşanmış olup olmadığını da saptayabiliyor; dahası, bu kişilerin ileride karşılaşacakları bir seçimde ne karar vereceklerini sadece 10 beğeni ile iş arkadaşlarından, 70 beğeni ile arkadaşlarından, 150 beğeni ile ebeveynlerinden, 300 beğeni ile de hayat arkadaşlarından daha yüksek doğrulukla tahmin edebiliyordu.

Kuşkusuz, bu bilgi ters yönde, yani girdi olarak verilen detaylı kişilik özelliklerine sahip Facebook kullanıcılarını bulmak için bir tür "arama motoru" olarak da kullanılabilirdi. Bu bulguların akademik dünyada duyulmasından sonra Cambridge Analytica isimli şirketin siyasetçilere teker teker seçmenler bazında hedefli propaganda yapmalarına olanak veren benzer bir hizmeti sunmaya başladığı anlaşılıyor. Örneğin Donald Trump'ın rakibi Hillary Clinton'ın birçok potansiyel destekçisinin, sadece mensup oldukları dar bir kategorideki kullanıcılara gösterilen,

bu nedenle Clinton'ın cevap verme şansı bile bulamadığı "karanlık reklam"larla oy vermekten vazgeçirildiği söyleniyor. Bir şirketin reklam geliri sevdası, dünya tarihini değiştirmiş olabilir.

41 | Robotlar askere alınsın mı?

Kore Yarımadası'nın ortasında Kuzey ve Güney'i ayıran Askersizleştirilmiş Bölge, dünyanın en yoğun şekilde silahlandırılmış sınırlarıyla çevrilidir. Yapay zekânın askeri uygulamalarının en çarpıcı örnekleri de burada denenmiştir.

Samsung şirketinin ürettiği SGR-A1 platformları, kamera ve kızılaltı aydınlatmalı gece görüş cihazlarıyla her hava koşulunda sınırın Güney Kore tarafında nöbet tutmakta. SGR-A1 üşümez, sigara içmez, dikkati dağılmaz, uyumaz ve korkmaz. En iyisi de, birisi onu havaya uçurursa, geride gözü yaşlı bir aile bırakmaz. Üzerine yerleştirilmiş tüfek veya bomba atar gibi silahları otomatik olarak ateşleme yeteneğine sahip olduğu düşünülen SGR-A1'le ilgili çoğu detay gizli tutulduğundan bu kullanım şekli devreye sokuldu mu, yoksa tepkilere karşı yapılan açıklamalarda denildiği gibi komuta merkezinin onayı olmadan ateş açması imkânsız mı, bu aşamada bilemiyoruz.

Yine bir Güney Kore şirketi olan DoDAAM'ın ağ sitesinde müşteriye sunulan robotlar "ölümcül" olanlar ve olmayanlar olarak iki kategoride listelenmektedir. Askersizleştirilmiş Bölge'de denendiği söylenen Super aEgis II platformu, 3 km'ye kadar mesafelerdeki insanlara veya taşıtlara kilitlenip otomatik olarak ateş edebilme kabiliyetine sahiptir. Çeşitli Arap emirliklerine ihraç edilen ve karadan havaya füze de atabilen Super aEgis II'nin de bugüne dek sahiplerince hiç otomatik moda sokulmadığı, hep uzaktan bağlantıda olduğu bir merkezdeki insan

operatörlerin kontrolünde kullanıldığı söyleniyor. Ama gördüğünüz gibi hem bu sabit sistemleri, hem de onları bir sürücüsüz arabaya koyarak elde edebileceğiniz gezginci savaşçıları tümüyle kendi başlarına bırakmak artık hayal değil.

Şimdi bu silahlı özerk sistemlerin havada uçabildiklerini düşünün. Sadece bir üniforma veya cilt rengine sahip herkese saldırabilecek çok sayıda küçük SİHA'nın (Silahlı İnsansız Hava Aracı) veya telefon sinyallerine kilitlenebilecekleri ya da yüzlerinden tanıyacakları belirli kişileri öldürmek için gönderilebilecek suikastçı SİHA'ların imal edilmesi mümkün.

Ateş emri verme yetkisinin insanlarda kalmasının önemi, kendilerine muhtemelen hayatlarımızı borçlu olduğumuz iki Sovyet subayının, Vasili Arkhipov ve Stanislav Petrov'un başlarından geçenlerden anlaşılabilir. Arkhipov 27 Ekim 1962'de Küba yakınlarında B-59 adlı denizaltıdaki üç yüksek rütbeli subaydan biriyken bir ABD gemisinin onlara saldırmakta olduğunu sanan diğer iki subayın ısrarına rağmen oybirliği gerektiren nükleer saldırı kararına katılmayarak dünyayı felaketten kurtarmış, Petrov da 26 Eylül 1983'te görev başında olduğu nükleer uyarı merkezinde gözlem uydusundan gelen bir görüntünün ABD'nin füze saldırısı olarak yorumlanması üzerine misilleme sürecini başlatmak yerine arıza raporu vererek 3. Dünya Savaşı'nı önlemiştir. Bu iki kişinin yerinde otomatik sistemler olsaydı, halimizin nice olacağını düşünmek ürperticidir.

Ben meslek hayatım boyunca askeri uygulamalara katkı yapmamayı seçenlerdenim; Google'ın ABD Savunma Bakanlığı'nın kimi projelerine dahil olmaktan çalışanlarının ayaklanması sonucu vazgeçmesinden de anlaşılacağı gibi, bu yapay zekâ araştırmacıları arasında az görülen bir tutum da değil, ama askeri teknoloji konusunda çalışan arkadaşları ayıplıyor filan değilim. Bağımsızlık iddiası olan ülkelerin kendi silah sistemlerini kendileri geliştirip üretmeleri işin doğası gereğidir. Fakat öldürme kararı verme yetkisinin insanların elinden çıkmasına yol açacak özerk ölümcül robot teknolojisi de herhangi bir silah sis-

temi değil ve bilim dünyası buna neredeyse oybirliği ile karşı. Özerk saldırı silahları geliştirmenin yasaklanmasına ilişkin bir çağrı binlerce araştırmacı tarafından imzalandı. Umarım akıl galip gelir.

42 | Yapay zekâ doktorluk yapar mı?

Aklını o yönde yoranlara insanları öldürmek için yeni olanaklar sunan yapay zekâ, bizleri daha uzun ve sağlıklı yaşatmak için de kullanılabilir elbet.

Eski moda yapay zekâ araştırmalarının elle tutulur ürünlerinden olan uzman sistemlerden daha önce bahsetmiştik. Dar bir uzmanlık alanındaki bilginin o işin ehli insanlarla yapılan uzun konuşmalar sonucu anlaşılabilir kurallar şeklinde formüle edilmesine ve gerçek sorunları çözüp varılan sonuçları gerekçelendirmekte bu kurallardan otomatik olarak kurulan zincirlerin kullanılmasına dayanan bu programların ilk uygulama sahalarından biri de tıp olmuştur. Her YZ ders kitabının uzman sistemleri anlatan bölümünde, bulaşıcı hastalıkların teşhisinde uzmanlaşan MYCIN sisteminden söz edilir. Çağımızda bu yaklaşımı (konuyla ilgili çok sayıda bilimsel makaleyi de kullanıcıya kolayca sunmak gibi ek işlevlerin yanı sıra) IBM'in doktorlara kanser tedavisinde yardımcı olma iddiasıyla pazarladığı Watson Oncology sistemi temsil etmektedir. Sistemin farklı kanser türleri için farklı özelliklere sahip hastalara hangi tedavinin uygulanacağına ilişkin önerileri ABD'deki Memorial Sloan Kettering Kanser Merkezi'ndeki uzman doktorların eğitimiyle belirlenmekte ve yeterli veri bulunmadığı için bilimsel açıdan net olmayan kararlar bu doktorların (çalışma yerleri ve hasta profillerince de etkilenebilecek) kişisel tecrübelerine dayanmak durumunda kalmaktadır. Bu tip bir destek ilgili konuda hiç uzmanın bulunmadığı hastanelerde kuşkusuz olumlu

bir katkı oluştursa da, kanserle mücadelede yapay zekânın henüz istediğimiz noktada olmadığını söyleyebiliriz.

Master Algoritma kitabının yazarı olan Pedro Domingos, kanser hastalığının (daha doğrusu, birçok farklı hastalıktan oluşan kanserler ailesinin) çaresinin bulunmasının bir yapay öğrenme problemi olduğunu söyler. Alışveriş siteleri (40. Soru), hangi müşterinin hangi üründen ne zaman kaç adet aldığı verisinden ileride benzer bir müşterinin harcamasını en yüksek değere çıkarmak için neler önermeleri gerektiğini hesaplamaya yarayan modeller öğrenmeye çalışır. İnsanların gen dizilimleri, aldıkları ilaç kombinasyonları ve yaşam süreleri arasında (herhalde toplayabildiğimiz tüm diğer verileri de göze alarak keşfetmemiz gereken ve her hasta için kişiye özel bir formül bulmamıza el verebilecek) benzer bir ilişki vardır. Sorun mevcut yapay öğrenme tekniklerinin eldeki miktardaki veriyle bu (çok karmaşık) modeli öğrenmekte yetersiz kalmasıdır. Ama kimi daha basit örüntü tanıma problemleri bağlamında yapay zekâ şimdiden insan doktorlarla boy ölçüşmeye başlamıştır.

Sepsis, vücudun bağışıklık sisteminin bir enfeksiyona karşı verdiği tepkinin kontrolden çıkması sonucu organlara hasar verdiği ve tanılanmasında saatler mertebesinde geç kalındığında ölüm oranı çarpıcı şekilde yükselen tehlikeli bir hastalıktır. Veri biliminin başarılı bir uygulamasıyla 16.000'den fazla hasta kaydı üzerinde yapılan bir çözümleme sonucu idrar miktarından akyuvar sayısına uzanan 27 rutin ölçümden sepsis uyarısını %85 doğrulukla, çoğu durumda hiçbir organa zarar gelmeden verebilen bir algoritma geliştirilmiştir. Yine devasa miktarda veri üzerinde çalışan başka bir uygulama, ABD'de iki hastanede yapılan denemelerinde hastaların taburcu olamadan ölme risklerini yerleşik formülden daha doğru tahmin edebilmiştir. Özellikle uzman doktor sayısındaki azlığın hastaları yeterli derecede yakından izlemeyi zorlaştırdığı ortamlarda böyle otomatik sistemlerin önemi açıktır.

Halk arasında "kireçlenme" olarak bilinen osteoartrit, orta ve ileri yaşlardaki kişilerin çoğunu etkileyen bir eklem

hastalığıdır. Ağrı şikayetiyle gelen hastalara röntgen veya manyetik rezonans görüntüleme (MRI) incelemesi sonucu tanı konulur. Hastalık eklemin kıkırdak yapısının bozulmasıyla oluşur ve günümüzde tanı konulduktan sonra bu hasarı düzeltmenin bir yolu yoktur. Şikâyetler başlamadan yıllar önce osteoartrit yolundaki kişileri diğerlerinden ayırmanın bir yolu olsa iyi olurdu, değil mi? Yapay öğrenme sayesinde böyle bir umut doğmuştur. Araştırmacılar insan doktorların gözüne tümü de sağlıklı görünen çok sayıda diz MR'ını "görüntüleme tarihinden 3 yıl sonra osteoartrit teşhisi konulanlar"/"konulmayanlar" olarak etiketleyerek bilgisayara göstermiş ve yeni geliştirdikleri sistem daha önce görmediği örneklerle çalıştırıldığında, hastalanacak kişilerin kıkırdaklarını hastalanmayacaklarınkilerden %86 doğrulukla ayırt etmeyi başarmıştır.

Röntgen filmleri gibi diğer radyolojik görüntülerdeki hastalıkla ilgili örüntülerin saptanmasında yapay öğrenme sistemlerinin insan uzmanlardan daha başarılı olduğunu bildiren çalışmalar giderek çoğalmaktadır. 2018 Mayıs ayında, sağlıklı ben mi melanom mu (bir tür cilt kanseri) olduğu etiketlenmiş yüz bin görüntü üzerinde eğitilmiş bir sinir ağının daha sonra yeni gördüğü 100 vaka üzerinde yarıdan çoğu beş yıldan fazla tecrübeye sahip 58 dermatologla karşılaştırıldığı ve doktorlara daha büyütülmüş görüntülerle hastanın yaşı, cinsiyeti vs. ek bilgiler sağlanmasına karşın sinir ağının daha yüksek doğruluğa ulaştığı duyurulmuştur. Hastaların görüntüleme incelemelerinin bilgisayar tarafından yapılmasında ısrar edecekleri günler yakındır.

(İnsan kumandasındaki) "robot cerrah"lar kesme işlemlerini insan elinden daha düzgün yapabiliyor, ama cerrahinin yapay zekâya terk edilmesine daha uzun süre olduğunu düşünüyorum. Görüntüleme teknikleri eskiye oranla çok gelişmiş olsa da, günümüzde cerrahlar hâlâ çoğu hastanın tüm gerçeğiyle ancak onu kesip açtıklarında karşılaşmakta ve kısa bir süre içinde akıl yürüterek karar verme durumuyla baş başa kalmaktadır. Bu bağlamda güncel yapay öğrenme yöntemlerinin gereksindiği

boyutta veriyi toplamanın ve hakkıyla sayısallaştırmanın zorluğu nedeniyle bu süreci tümüyle emanet edebileceğimiz bir yapay zekâyı henüz ufukta göremiyorum. Bu da hastanın elini tutup onu hastalığıyla nasıl mücadele edeceği konusunda bilgilendirirken yüreklendirme sanatı gibi tıbbın insan dokunuşuna hâlâ ihtiyaç duyduğu alanlarından.

5. Bölüm
YAPAY ZEKÂNIN GELECEĞİ

43 | Yapay zekâ yanlış yapar mı?

Bir sistemin nasıl işlediğini gerçekten anlamanın en iyi yollarından biri, onun çalışmasındaki aksamaları incelemektir. Bu soruda güncel yapay zekâ programlarının yaptıkları hataları göreceğiz. Bunların hiçbiri düzeltilemeyecek şeyler değil; ama halen giderilememiş olmalarının sebeplerini ve düzeltme yollarını düşünmek eğitici.

Google'ın hayatımı genellikle kolaylaştıran servislerinde hata bulduğum zaman, işte bu nedenle mesleki bir haz duyuyorum. Bu satırları yazmaya başlamadan önce Google Haritalar'a İstanbul'da ünlü bir alışveriş merkezi olan Akmerkez'in (üçgen tabanlı bir binadır) bir kenarındaki bir kapısından diğer bir kenarındakine en kısa yürüyüşle nasıl gidebileceğimi sordum. Doğru cevabın ilk kapıdan girip binanın içinden ikinci kapıya ulaşmak olduğu besbelli, değil mi? Oysa program yol tarifini binanın dışından vermekle kalmadı, muhtemelen kendisine sağlanan harita bilgilerinde yaya yolları eksik girildiği için bana binayı arkama alıp uzaklaşmamla başlayan ve dört kez gereksiz

yere kaldırımdan inip karşıdan karşıya geçmemi gerektiren 600 metrelik bir açık hava rotası önerdi.

Aralık 2017'de Kaliforniya'yı kasıp kavuran orman yangınları Los Angeles şehrinin kimi mahallelerine sıçramıştı. Telefonlarımızdan nerede ve ne hızla hareket etmekte olduğumuzu anlayarak hangi yolun tıkalı, hangisinin açık olduğunu hesaplayan yol tarif sistemlerinin bir sakıncası o sırada ortaya çıktı. Yanan mahallelerde yolların boş olduğunu "gören" yapay zekâlar, sürücülere ilk seçenek olarak o yollardan geçen rotalar öneriyordu. Los Angeles polisi kentteki şoförlere navigasyon uygulamalarını kapatma çağrısı yapmak zorunda kaldı.

Gelelim Google Çeviri'ye. Sinir ağı teknolojisine geçmeden önceki "milk port" halini 38. Soru'da gördüğümüz bu servis artık daha iyi, ama dil çok zor iş. Yine bugün (23 Haziran 2018) itibarıyla bulabildiğim birkaç sorun şöyle:

"O bir banka memuresidir" cümlesi Türkçeden İngilizceye "It's a bank note" ("O bir banka notu") olarak, "memure" kelimesinin dişil anlamından tümüyle bihaber şekilde çevriliyor. Anlattığımız gibi, bu yaklaşım kelimelere bir sözlükten bakıp oradaki bilgilerden yararlanmayı içermiyor. Anlamlar uzayında çok sayıda çeviri örneğinden öğrendiği bir tür "döndürme" yapıyor. "Memure" gibi az kullanılan sözcükler örnekleri arasında çok geçmediyse böyle sonuçlar çıkıyor.

"Benim elmam kırmızı. Senin elman ne renk?" metni "My apple is red. What color is your hand?" diye çevriliyor. İlk cümlenin çevirisinde sorun yok, ama ikincisi (sistemin insanların aksine önceki cümlelerde kurulan bağlamı da hiç dikkate almadığını gösterir şekilde) "Senin elin ne renk?" olarak çevrilmiş. Burada yine ("Elman" kelimesinin çeşitli dillerde insan ismi olarak da geçmesinin de rol oynadığı) veri eksikliğinden kaynaklı bir "olsa olsa bu olur" atışının söz konusu olduğunu sanıyorum.

"Başka kadına bakarsa gözünü oyarım" cümlesinin Google çevirisi "If you look at another woman, I'll take a look" ("Eğer siz başka bir kadına bakarsanız, bir göz atacağım") şeklinde gerçekleşiyor. Romantik ilişkilerinizde

otomatik çeviri sistemlerine güvenmemenizi öneriyorum.

Google Çeviri, hatalarını düzeltmemiz için biz kullanıcıları öteden beri özendirmektedir. Dilerseniz siz de Çeviri Topluluğu'na üye olup katkı verebilirsiniz. Servisin masaüstü bilgisayarlarda çalışan sürümünde, çok sayıda insan gönüllü tarafından doğruluğu onaylanmış çevirilerin yanında özel bir arma sembolü çıkar. Ne yazık ki "You are beautiful too" cümlesinin Türkçeye "Sende güzelsin." diye, hem de armalı olarak çevrilmesi, çoğunluğun daima haklı olmadığını ve "de"leri ne zaman ayıracağını bilmediğini hatırlatan acı bir örnektir.

Bu problemin bir benzeri, Microsoft şirketinin 23 Mart 2016'da kamuoyuna duyurduğu Tay adlı Twitter karakterinin başına geldi. Genç bir kızı canlandıracak şekilde hazırlanan Tay'in başka Twitter kullanıcılarıyla yazışarak öğrendikleriyle dil kullanımını geliştirmesinin öngörüldüğünü okuduğumda "Eyvah" dediğimi anımsıyorum. 24 saat içinde çok sayıda (kimileri kötü niyetli) kişiyle tweetleşen Tay, Hitler hayranı, soykırımı onaylayan, cinsiyetçi laflar eden tiksinç bir tipe dönüşmüş ve hesabı geliştiricilerince bir daha açılmamak üzere kapatılmıştı. Her ebeveynin bildiği gibi, öğrenen bir sistem üzerinde çalışıyorsanız, onun kimlerle konuştuğuna dikkat etmeniz gerekir.

İnsan beyninin zayıflıklarını ortaya çıkaran görsel yanılsamalarla karşılaşmışsınızdır. Aslında tıpatıp aynı olan iki şekli karmaşık bir resmin iki ayrı yerinde farklı boyut veya renkte "görmemiz"e veya sabit bir şeklin kimi parçalarını dönüyor gibi algılamamıza yol açan nedenlerin incelenmesi görme sistemimizin daha iyi anlaşılmasına yardımcı olur. Son birkaç yıl içinde görüntü tanımak için eğitilmiş sinir ağlarını yanıltmak için "hasmane örnekler"in inşa edilmesi önemli bir araştırma konusu haline gelmiştir. Hayvan türlerini sınıflandırmak için eğitilmiş bir ağın normalde doğru olarak "panda" olarak tanıdığı bir resmi oluşturan noktacıklarda insan gözünün fark edemediği değişiklikler yaparak ağın yeni resmi bambaşka bir tür olan "jibon" sanmasına yol açmak mümkündür.

Bu fikrin korkutucu bir başka uygulamasında sürücüsüz bir arabanın kamera girdisinde yine insanlarca fark edilemeyen oynamalarla otomobilin önündeki yayaları görmeyip açık bir yol imgesiyle karşı karşıya olduğunu sanması sağlanmıştır. Erkek olduğu yüzünden rahatça anlaşılabilen bir yapay öğrenme araştırmacısı, bu iş için özel olarak hesaplanmış desenlerle bezeli bir gözlük çerçevesi takarak ünlülerin yüzlerini öğrenmiş bir yüz tanıma sistemine kendisini manken Milla Jovovich olarak tanıtmayı başarmıştır. Çok yol aldık, ama daha çok yolumuz var.

44 | Yapay zekâ kullanımının zararları nelerdir?

Her teknoloji iyiye de kullanılabilir, kötüye de. Geçtiğimiz sorularda yapay zekâ için iki türden de örnekler gördük. Bu soruda son yıllarda yapay öğrenme, büyük veri ve sosyal ağların buluşmasıyla ortaya çıkan ve ilk bakışta "iyi" gelişmeler gibi görünen kimi uygulamaların yeni yeni farkına varılan olası sakıncalarından söz edeceğim.

Günümüzde doğaları gereği "sınıflandırma" içeren (çok sayıda iş başvurusunun büyük kısmını eleyerek mülakata çağırılacak olanları veya şartlı tahliye için başvuran hükümlülerden tekrar suç işlemeyecek gibi görünenleri seçme, okulları, restoranları ya da başka işyerlerini belli bir "kalite" sıralamasına sokma, vb.) birçok karar, bu işteki başarımlarının hızla arttığını gördüğümüz yapay öğrenme ürünü algoritmalara bırakılıyor. Çoğu durumda bu süreçlere insanlara özgü rüşvet, kayırma, tarafgirlik gibi kötü huylarının olmadığı düşünülen bilgisayarların dahil edilmesi olumlu bir gelişme olarak görülüyor, başvuranlar hoşlarına gitmeyen sonuçlar karşısında sızlanırlarsa "Sistem öyle karar verdi, yapacak bir şey yok!" karşılığını alıyor. İşte sorun da burada.

Bu sistemler nasıl karar vereceğini nereden öğreniyor?

Önceki yıllarda insanlar tarafından verilmiş kararlardan. Peki ya o insanlar o kararlara tarafgirlik bulaştırdıysa?

Londra'daki St. George Hastanesi Tıp Okulu, iş başvurularını bir yapay zekâya eletme uygulamasına ilk geçen kurumlardandı. Eski başvuruların kabul/red bilgileriyle eğitilen sistem devreye sokulduğunda, her yıl gelen 2000 dolayında dosyayla başa çıkmanın sağlıklı bir yolu olarak alkışlanmıştı. Ne yazık ki dört yıl sonra yapılan bir inceleme, bilgisayarın kadınlara ve Pakistan gibi ülkelerden başvuran doktorlara negatif ayrımcılık yaptığını, yılda yaklaşık 60 kişinin akademik başarıları gözardı edilip sadece cinsiyetleri veya isimlerine dayanılarak reddedilmiş olduğunu gösterdi. Makine, geçmişteki işe alma kararlarını veren insanların önyargılarını öğrenip kusursuzca devam ettirmişti.

ABD nüfusunun %12'si siyahlardan, %64'ü ise beyazlardan oluşuyor. Oysa cezaevlerindeki mahpusların %33'ü siyah, beyazların oranı ise sadece %30. Ülkenin birçok eyaletinde hâkimler tutuklamanın gerekliliğini değerlendirirken sanıkların serbest bırakılmaları halinde tekrar suç işleyip işlemeyeceklerine ilişkin tahmin skorları hesaplayan yazılımların çıktılarını da dikkate alıyor. Sanığın anne veya babasının hapis geçmişi olup olmaması gibi sıkıntılı verileri de esas alan bu yazılımlardan birinin ırkçılık yaptığı, yani siyahlara beyazlardan hak etmedikleri derecede kötü skorlar verdiği, hakkında tahminde bulunduğu kişilerin ilerideki yıllardaki sicilleri de incelenerek kanıtlandı.

Tahminleri yüksek doğrulukla çıksa bile, belli bir dönemin insani verisiyle eğitilmiş yazılımları devreye sokup karar sürecinin katı bir parçası olarak kullanmak, toplumun ilerlemesine engel koyma sonucunu doğurabilir. Yapay öğrenme sistemlerinin temel varsayımı, gelecekte işlerin geçmişteki gibi devam edeceğidir. ABD'de tam da bu konudaki çalışmalarıyla tanınan biliminsanımız Zeynep Tüfekçi'nin belirttiği gibi, Google Çeviri "O bir doktor" cümlesini İngilizce'ye "He is a doctor" diye çevirirken, "O bir hemşire" karşılığı olarak "She is a nurse"

diyor; yani bu meslekleri icra edenlerin cinsiyetleri hakkındaki yerleşik varsayımları da öğrenmiş. Önyargılarımızın bilgisayarlarımızda ilelebet yaşamaması için önlem almalıyız.

Cathy O'Neil'in büyük veriden beslenen şeffaflık yoksunu matematiksel modellerin kitlesel kullanımının yarattığı birçok problemi işlediği harika kitabı "*Matematik İmha Silahları*"nı (*Weapons of Math Destruction*) konuyla ilgilenenlere hararetle tavsiye ederim. Sınıflandırma mantığı insanların birey değil, "tip" olarak ele alınmasını dayatıyor: İnternetteki hareketlerinizin bıraktığı izlerden fakir mi zengin mi olduğunuzu çıkarsayan algoritmalar size buna göre farklı muamele ediyor ("yağlı" müşteriler insan temsilcilere bağlanırken yoksullar sohbotlara yönlendiriliyorlar), borç para veren şirketler verileriniz "geri ödeme riskli" kişilerinkilere benziyorsa (sözgelimi fakir bir mahallede oturuyorsanız) sizden yüksek faiz istiyorlar, şehrin hangi bölgelerine ek devriyeler gönderileceğini veriye dayalı olarak belirleyen programların yolladıkları polisler, orada oldukları için başka mahallelerde şikâyet edilmeden geçilecek küçük suçları da görerek o bölgenin suç sicilini kabartıp daha da çok polis gönderilmesini tetikliyor, böylece "sistem" bir eşitsizliği alıp daha da büyüten bir girdap oluşturmuş oluyor.

Yapay öğrenmenin mümkün kıldığı hedefli reklamlarla özellikle eğitimi veya sosyal konumu nedeniyle ürünün kalitesini sağlıklı değerlendiremeyen ve ucuz alternatifleri aramayı bilmeyen kişileri bulup kazıklamaya odaklı "yırtıcı" siteleri de ifşa eden ABD'li O'Neil'in de işaret ettiği gibi, bu yeni teknolojinin bir suçluyu yakalaması övülürken ürettiği yanlış alarmlarla onlarca kişinin gününü berbat ettiği gözardı edilen yüz tanıyıcılardan tutun, bildiklerini insanlarca anlaşılabilen simgesel gösterimle tutan eski usul uzman sistemlerin aksine, programları yüz binlerce bağlantı ağırlığından ibaret olduğu için verdikleri yaşamsal kararları anlaşılır şekilde gerekçelendiremeyen sinir ağlarına kadar yüzleşmemiz gereken birçok karanlık yanı var. Bu kâbus senaryolarının belki de en ürperticisi

ise Çin Halk Cumhuriyeti'nde şimdiden uygulamaya konulmuş olan vatandaş puanlama sistemi.

2014'ten beri farklı uygulamaların entegre edilmesiyle genişlemekte olan ve 2020'de ülke çapında hayata geçişinin tamamlanması öngörülen Çin sosyal kredi sistemi, her bireyin (ne kadar "iyi" bir yurttaş olduğuna göre hesaplanan) bir puanının olması ve bu puana göre günlük hayatında ödüller veya cezalarla karşılaşması fikrine dayalı. Ülkenin dev e-ticaret şirketi Alibaba'nın finans kolunun müşterilerine atadığı ve 350 ile 950 arasında değişebilen "Susam Kredisi" puanları, sistemin önemli parçalarından biri. Borçlarını (ve mahkemelerce verilmiş para cezalarını) zamanında ödemeyenlerin puanı düşüyor. Krediniz yüksekse hastanede doktora görünmeden önce girmeniz gereken para ödeme kuyruğunu ("komünist" bir ülke için kulağa inanılmaz gelen bu deneyimi bir ziyaretimde bizzat yaşadım) atlama hakkı kazanıyor, apartman dairesi veya bisiklet kiralarken depozito ödemiyor, vize kolaylıklarına, arkadaş bulma sitelerinde daha çok görünürlüğe ve daha nice avantajlara kavuşuyorsunuz. Krediniz düşükse de bu iyi şeylerin tersi oluyor. Bu nedenle kredi düşürme internette fazla oyun oynamaktan "yasadışı sosyal örgüt üyeliği"ne, otellerde rezervasyon yaptırıp sonra gelmemekten alışveriş sitelerinde uydurma ürün değerlendirmeleri yazmaya dek bir dizi istenmeyen hareketi cezalandırmak için kullanılıyor. Kredi seviyesi nedeniyle uçak bileti alması engellenenler çoğalıyor. Çocuk bezi almak (sorumlu bir ebeveyn olduğunuzu gösterir), sosyal medyada ülkenin gidişatı hakkında "olumlu" mesajlar atmak ve kredisi yüksek başka kişilerle "arkadaş" olmak ise kredinizi yükseltmenin yollarından bazıları.

Kimi insanlar yapay zekânın bir gün ayaklanıp insanları ezeceğinden kaygılanıyor (49. Soru). Bence bu soruda söz ettiğimiz yöntemlerle insanın insanı ezmesi için eşi görülmemiş bir araç olarak kullanılma riski daha gerçekçi. Demokrasiyi savunup sömürüye karşı durmak teknolojik gelecekte de bir ödev olmaya devam edecek.

45 | Robotlar âşık olmalı mı?

Kitabı bu sayfaya kadar okuyanlar, başta sözünü ettiğimiz "İnsanların yapabildiği, ama makinelerin yapamayacağı bir şey var mıdır?" sorusuna yanıtımı biliyor. Bilimin bu konuda vardığı sonuç bence çok net. Ama yine de, düşünce insanları arasında bile, kimi insani özelliklere bilgisayarların asla sahip olamayacaklarını düşünenler mevcut. "Tamam" diyorlar, "her tür insan davranışının taklit edilebileceğini anladık, ama davranıştan değil, 'içeriden' bahsedelim. Bilgisayarların duyguları olabilir mi? Robot hâkimlerin vicdanı olur mu? Bir makine çocuğunun yasını tutan bir annenin ne hissettiğini duyumsayabilir mi? Sevinebilir mi? Âşık olabilir mi?"

Bu çetrefil konuya bir sonraki soruda eğileceğiz. Ama sağlıklı bir tartışmaya hazırlık olarak önce duygu nedir, insanlarda neden evrilmiştir ve de makinelere gerekir mi; mühendis gözüyle bir bakalım derim.

İnsan zihni, yaratmaya çalıştığımız makine zihninden çok farklı koşullarda ortaya çıkmıştır. 15. Soru'nun cevabında sözünü ettiğimiz gibi, beyin aynı anda bedenin farklı kısımlarına veya gündem maddelerine ilişkin birçok farklı programı çalıştıran bir bilgisayardır. Bu programlardan bazıları diğer hayvan türleriyle ortak atalarımızın zamanında, bazılarıysa insan türü diğerlerinden ayrıldıktan sonra, topluluklar halinde yaşamanın getirdiği evrimsel baskıların sonucu şekillenmiştir. Kimi zaman gözden kaçırılan bir husus, bu programların aslında onları kafalarında çalıştıran bireylerin esenliği için değil, o bireylerin beden planlarında yer alan genlerin sonraki nesillerde olabildiğince çok kopyasının bulunması için böyle şekillendiğidir.

Duygular da bu programlardandır ve her birinin var olması için evrimin mantığına uygun bir neden vardır. Korku programı evrildiği çağlarda atalarımız için ölüm tehlikesi uyarısı olabilecek (ani gürültüler, yılanlar,

zehirli olabilecek örümcekler, vs.) sinyallerce tetiklenir (ama ne yazık ki beyin devrelerimize bir kez öyle işlendiği için emniyet kemeri takmama, sigara vs. çağımızda geçerli ölüm tehlikelerine aynı tepkiyi vermez) ve vücutta yol açtığı (gözlerin büyümesi, kasların gerilmesi, adrenalin salgılanması vs.) otomatik değişikliklerle bireyi mantıklı bir akıl yürütme sürecinden çok daha kısa sürede kaçmaya veya savaşmaya hazır hale getirir. Tiksinme programı çocuklukta büyüklerden iğrenç olduğu öğrenilen şeyleri yemeyi istememe sonucunu doğurur, bunun zehirlenme riskine karşı evrimsel bir avantaj getirdiği barizdir. Bu gibi temel duyguların etnik veya kültürel farklılık dinlemeyen "evrensel" yüz ifadelerine yol açması da ortalıkta korkulacak veya iğrenilecek bir şey olduğu bilgisinin yakındaki (muhtemelen çok sayıda geni paylaştığımız) bireylere hızla aktarılabilmesi için evrilmiş olabilir.

Bireylerarası ilişkilerin matematiğini ekonomi ve oyun kuramı gözlüğüyle inceleyen araştırmacılar, başka birçok duygunun evrilmesi için makul açıklamalar geliştirmiştir: Faydalı bir şeyin paylaşımında veya genelde iki tarafın da çıkar sağlaması öngörülen herhangi bir anlaşmada karşı tarafın benden fazla pay alması veya sözünü tutmayarak beni zarara sokması genlerim için kötü haberdir. Herkesin genleri için aynı durum geçerli olduğundan bana kazık atanlara saldırmamı tetikleyerek potansiyel kazıkçıları caydıran bir programın evrilmiş olması doğaldır. Adalet duygusu, dostluk, bireyin sözünü tutmadığının anlaşılmasının kötü sonuçlarından çekinmesine yarayan suçluluk duygusu ve vicdanın, bu bağlamda ortaya çıkmış olmaları akla yakındır.

Anneler çocuklarını sever, çünkü bu güçlü duygu sevilen çocukların hayatta kalma şansını artırdığından sevgisiz annelerin soyu çoktan tükenmiştir. Kadınlarla erkekler âşık olur, çünkü insanlar âleminde beyin ne kadar büyük olursa o kadar iyidir; ama dişilerimizin doğurabilecekleri çocukların kafaları "mimari" kısıtlar nedeniyle belli bir boydan büyük olamadığından, insan yavrularının

gelişmelerini tamamlamadan doğup yıllarca bakılmaları gerekir. Eh, potansiyel anne ile babanın ilk başta bu zahmete girmesine vesile olan ve zorlu çocuk yetiştirme sürecinden bezip komşu mağaralardaki çekici bireylere doğru yelken açmalarını engelleyerek, onları partnerlerine "mantıksızca" bağlayan bir program da bu koşullarda çocukların hayatta kalıp kendi tohumunu taşıyan genleri yayma olasılığını yükselterek evrilmiştir.

Şimdi yapay zekâya gelelim. Eğer özellikle uğraşıp yukarıda anlatılana benzer bir seçilim ortamı oluşturmazsak (27. Soru) yapay zekâ programları evrim yoluyla ortaya çıkmaz. Genleri yoktur. Dolayısıyla da üreyip çoğalmayı, hatta "hayatta" kalmayı (mesela birisinin bilgisayarın fişini çekip programın da bütün kopyalarını silmesini engellemeyi) istemeleri için bir sebepleri yoktur. Eğer biz bu tür istekleri onlara özellikle kazandırmazsak tabii.

Çoğu durumda makinemizin duyguları olmasını istemeyiz. Nükleer santral kazalarında devrelerine sonuçta hasar verecek yükseklikte rạdyasyon içeren bölgelere yollayacağımız robotların korkup kaçması, ya da cep telefonumuzdaki kişisel asistanımızın bazı arkadaşlarımızdan hoşlanmadığı için telefonu suratlarına kapatması mantıklı olmaz. Ama robotlarımızın kimi duyguları varmışçasına davranmasının anlamlı olacağı durumlar da akla gelmiyor değil.

Nüfusun her yıl yüz binlerce kişi azaldığı, yaşlıların nüfustaki payının arttığı, kültürün de cansız cisimlerin de ruhlarının olduğuna inanılmasına el verdiği Japonya, insansı robotların hayatın her alanına girmesine açık. Ülkede azalan işgücünün robotlarla desteklenmesi öngörülüyor. Pizza taşıyan küçük sürücüsüz arabaların yoldan geçenlerce tekmelendiği ABD'nin aksine, yalnızlara "can yoldaşı" olan, yaşlıların ilaç, egzersiz vs. gereksinimlerini karşıladıklarını kontrol edip onlarla olabildiğince insani şekilde etkileşen robotlar, Japonya'da büyük kabul görüyor. İşte bu etkileşim de duygusallık görüntüsü vermeyi gerektiriyor. Sizin şirketinizin ürettiği sosyal robotların içlerinde bulundukları durumlara uygun duyguları rakip

şirketlerinkilerden daha inanılır şekilde yansıtmaları çıkarınızadır; o yüzden de bu işi insan düzeyinde yapmalarını sağlamaya çalışmalısınız.

"Yine de bu taklit, gerçek değil ki!" itirazına 46. Soru'da döneceğiz, ama burada gözden kaçırmamamız gereken ve belki de ileride işleri kökten değiştirecek olan şey, kimi insanların robotlara olan hislerinin gerçek olması. İnsan torunu olmayan ihtiyar bir Japon kadının Pepper model bir insansı robota torunuymuş gibi elbise dikmesinden birçok kullanıcısının zaten YZ içermeyen eski modellerine bile duygusal olarak bağlandığı bilinen seks oyuncakları endüstrisinin giderek daha gerçekçi robotlar imal etmeye başlamasına, çok alametler belirdi.

46 | Robotlar âşık olabilir mi?

İnsan beyninin evrendeki başka her şeyle aynı yasalara tabi bir fiziksel sistem olduğu gerçeğine ve Turing'in bilgisayarların her şeyi taklit edebilecekleri buluşuna rağmen, yapay zekânın imkânsız olduğu hâlâ iddia edilebilir mi? Bu sorudan itibaren kalan birkaç itirazı gözden geçireceğiz.

Bir itiraz türü, "hislere" dayanıyor: Bir insan davranışı bütün detaylarıyla taklit edilse bile bir de o insanın "içi", o davranışı gerçekleştirirken hissettikleri vardır. Âşık bir insanı mükemmel bir şekilde taklit eden, dış görünüşü de insandan ayırt edilemeyen bir robot yaptık diyelim. Sesinin titremesinden gözlerindeki pırıltıya, yazdığı aşk şiirlerinden sevgilisi bir başkasına ilgi gösterdiğinde kaşlarını çatmasına varıncaya dek hiçbir detayı eksik olmayan robotumuzun gerçekten âşık olduğunu söyleyebilir miyiz? Ya da yine mükemmel insan görünüşünde bir başka robotun eli kapıya sıkıştığında acıyla haykırdığını düşünün. Gerçekten acı hissettiğine inanır mısınız?

"Öznel deneyim"ler, tanımları gereği sadece onları yaşayan kişinin zihninde hissettiği, dışarıdaki bir gözlemcinin erişemeyeceği, bu nedenle teyit de edemeyeceği şeylerdir. Bir kişinin bedeninin verdiği birçok sinyali ölçebilirsiniz ama ne hissettiğini deneyimleyemezsiniz. Gerçekten acı çekiyor mu? Onun çektiği acı sizin acı çekerken hissettiğinizle aynı şey mi? Onun gökyüzüne baktığında gördüğü (ve "mavi" dediği) renk sizinkiyle aynı mı, yoksa o baştan beri sizin "mavi" dediğiniz şeyleri sizin "pembe" dediğiniz renkte, sizin "pembe"leri de sizin "mavi"nizde mi görüyor?

Bu zor soruların yanıtı henüz bilinmiyor ve böyle öznel deneyimlerin bir bilimcinin deneyini bir başkasının tekrarlayıp denetlemesi esasına dayalı nesnel bilimsel yaklaşımın kapsamına girmediğini söyleyenler bile var. Bence önemli nokta şu: Eğer kendimden başka hiçbir varlığın öznel deneyimlerine (onun sözüne inanmaktan başka bir yolla) erişemiyorsam, bırakın robotları, başka insanların öznel deneyimleri olduğuna, örneğin acı çekiyor gibi göründüklerinde gerçekten acı çektiklerine hangi gerekçeyle inanayım? Eğer insanlara bu konuda inanıyorsam, o zaman tıpatıp insana benzeyen bir robot aynı şekilde davrandığında ona niye inanmayayım?

"Ama insanlar et ve kandan, robotlarsa başka malzemelerden yapılıyor, başka insanlar benimle aynı model yaratıklar olduğundan benzer deneyimlere sahip olmamız doğal, robotlar öyle değil ki!" mi dediniz? Gelin bir düşünce deneyi yapalım.

Önce her şeyin beyinde olup bittiği konusunda anlaşalım. Bedenimizin başka yerlerindeki gelişmeler hakkındaki bilgiler beyne sinir hücreleri yoluyla ulaştıktan sonra duyumsanabiliyor. Bacağı kesilen kişilerin artık varolmayan ayaklarının ağrımasından şikâyet ettiği "hayalet uzuv" sendromu, esas gösterinin sahnelendiği organın beyin olduğunu gösteren ünlü bir örnektir.

Önceki sayfalarda söz ettiğimiz gibi, tek bir sinir hücresinin, hesaplama gücü kısıtlı bir işlemci olduğunu düşünüyoruz. Herhalde bir uzay mekiğinden daha karmaşık

olamaz, değil mi? Her ne kadarsa, düşünce deneyimizde teknolojideki gelişmeler sonucu insan sinir hücrelerinin eşlerinin başka malzemelerden (mesela şimdilerde bilgisayar ve robot inşa ederken kullandıklarımızdan) imal edilebildiğini varsayalım.

Şimdi sizin beyninizdeki hücrelerden birini cerrahi yolla çıkarıp yerine bu yapay hücrelerden birini taktığımızı düşünelim (Düşünce deneylerinde böyle şeylere izin var. Öte yandan nanorobotların vücudumuzda dolaşıp problemli hücreleri sağlamlarıyla değiştirmesi tıbbın geleceğinde ciddi ciddi öngörülen bir fikir). Bir sinir hücreniz aynı işlevi gerçekleştiren yapay eşiyle değiştirilince hisleriniz değişir mi? Düşünün: Organlarınızdan gelen sinyallerde bir değişiklik yok. Beyinde o sinyallerin işlenmesinde rol alan bir mekanik parça değişti sadece. Yapılan işlem yine aynı işlem, yani eliniz kapıya sıkıştığında yine aynı sinyaller aynı yollardan geçiyor, beyinde aynı örüntüler tetikleniyor ve iş yine konuşma üretim alt sisteminize "Aaah, elim!" dedirten örüntülere varıyor.

Bir değişiklik olmadığını kabul ettiyseniz, bir başka sinir hücrenizi daha yapayıyla değiştireceğim. Sonra bir daha. Bir daha. Sonuçta bütün beyniniz yapay hücrelerden oluşacak. Ve hâlâ eliniz sıkışınca tümüyle aynı şeylerin yaşanacağını iddia ediyorum. İşte acı çeken ve etten/kandan değil, başka malzemelerden yapılmış bir beyin. Demek ki oluyormuş.

Kabul etmiyorsanız, bu sürecin sonunda acı (ve başka herhangi bir şey) hissetmeyen bir hale geleceğinizi düşünüyorsunuz demektir. Bu durumda size işkence yapılmasında ne sakınca olduğunu söyler misiniz?

Ne kadar iyi bir yapay zekâ yaparsak yapalım, onun sadece bir taklitçi veya ruhsuz bir "zombi" olacağını (yani aslında "evde" kimse olmayacağını) ve hissettiğini söylediği şeyleri aslında bizim gibi deneyimlemeyeceğini savunuyorsanız, o zaman bu görüşteki birisinin bir insansı robota işkence yaparken şunları dediğini düşünün: "Saçmalamayın! Tabii ki bu robotun kolunu kırarsak canı

acımaz! Plastik ve metalden yapılmış bir makine o! Daha geçen gün fabrikada imal edildi! Ağlayıp yalvarmasına aldırmayın! Numara yapıyor! Şimdi de gözünü oyalım!"

Rahatsız edici, değil mi? Bu sahneyi düşünmek bile korkunç geliyor (İleride "robot hakları"na ilişkin ilk kampanya böyle gerçekçi insan görünüşlü robotlara, hele de çocuk şeklinde olanlara eziyet etmenin yasaklanması talebiyle başlarsa şaşırmam).

Sadece "hislere" dayalı iddialara, hele de bilimsel tartışmalarda, bel bağlamamak gerekli. Hisleriniz sizi yanıltıyor olabilir! Örneğin gündelik kararlarınızı, sözgelimi dün akşam tek başınıza sinemaya gidip gitmemek konusunda düşündükten sonra vardığınız gitme kararınızı her tür dış etkiden uzak olarak özgürce verdiğinizi, yani pekâlâ evde kalma kararı da verebileceğinizi hissediyor olabilirsiniz, ama 15. Soru'da da gördüğümüz gibi bu tip bir "özgürlük" bilimsel olarak imkânsız. Aslında moleküleriniz birbirleriyle fizik yasalarına göre itişti, daha düşük bir çözünürlükte bakıldığında beyninizin ve çevreden gelen sinyallerin o andaki toplam durumuna göre sinir hücresi etkinleşme örüntüleri birbirini tetikledi, sonuçta da bu karar çıktı. Tıpatıp aynı toplam durum tekrar kurulabilse yine aynı kararla sonuçlanacak, çünkü burada "sizin" etkilediğiniz bir süreç yok, mekanik bir hesaplama sonucu oluşan bir karardan sizin "ben" dediğiniz programın haberdar olup onu kendi kararı sanması var. Yani evrenin geri kalanından bağımsız bir "özgür irade" de insanlarda olup makinelerde olamayacak bir şey değil, çünkü aslında insanlarda da yok! Bu iradeye sahip olma hissi, kararın bir anda beynimizde oluştuğu duygusu, "ben"inizin karar için yapılan hesap tamamlanmadan önce sonucun ne olacağını bilmezken, hesap bitince onu öğrenmesinden kaynaklanıyor, tıpkı 33. Soru'da gördüğümüz satranç programının hangi hamleyi oynayacağını "düşünürken" (yani oyun ağacındaki durumları tararken) değil, hesabın sonunda bildiği ("kararlaştırdığı") gibi.

47 | Çince Odası nedir?

Tüm yazdıklarını okumanızı hararetle tavsiye ettiğim filozof Daniel Dennett, bilinç gibi öznel deneyime dayalı hususlarda kuram oluşturmanın güçlüklerinden birinin her insanın (en azından kuantum fiziği veya kanser biyolojisi kadar zor olan) bu konularda kendi kendisini uzman olarak görmesi olduğunu söyler. Filozof John Searle'ün yapay zekâya en ünlü itirazlardan biri olan meşhur "Çince Odası" argümanının yıllardır pek çok kişinin aklını çelmesinin nedeni, kanımca böyle bir "uzmanlık yanılgısı"ndan yararlanmasıdır. Ne diyor Searle?

Çince bilmeyen bir Türk'ü Türkçe yazılmış bir kurallar kitabıyla birlikte bir odaya kilitliyoruz. Dışarıdaki Çinliler, kapının altından art arda üstünde Çince harfler basılı kâğıtlar yolluyor. İçerideki adamcağız da kural kitabındaki "Şu, şu, şu harfler gelirse şu, şu, şu harfleri dışarı sür" türünden sıkıcı talimatlara göre, stoğundaki kâğıtları sıralayıp dışarı yolluyor. Kural kitabı öyle iyi yazılmış ki, dışarı çıkan cümleler, içeri girenlere en güzel cevapları oluşturuyor. Öyle ki Çinliler bir süre sonra içeride çok zeki bir Çinli'nin olduğuna inanıyor. Oysa Türk'ün bu sohbetten tek kelime bile anladığı yok. "İşte," diyor Searle, "Turing testini geçen bir bilgisayar da böyle olacak. Kural kitabı bilgisayarın çalıştırdığı programa (eğer sinir ağı kullanılıyorsa bağlantı ağırlıkları da buna dahil) karşılık geliyor. Girdi/çıktı ilişkisi çok zekice olsa da, "içeride" o programın komutlarının körlemesine izlenmesinden başka bir şey olmayacak, gerçekte bilgisayar hiçbir şey anlamayacak. Oysa insanlar öyle mi? Onlar duyduklarını gerçekten anlıyor, mesela sizin bu dediklerimi anladığınız gibi!"

Aynı fikirde değilim. Bu tezde birçok sorunlu nokta var. Öncelikle, yukarıda bizim duyduklarımızı nasıl anladığımızın anlatılmadığını fark ettiniz mi? Yukarıdaki paragrafı insan olmayan birisine, diyelim ki bir uzaylıya

gösterseniz bilgisayarla (veya hikâyedeki odayla) insanlar arasında anlama kalitesi açısından bir fark olduğuna nasıl ikna olacağını göremiyorum. Uzaylı konuşma sırasında insanların beynindeki değişiklikleri görebilse bile onlar "gerçekten" anlıyorlar mı, yoksa sadece öyle davranıyorlar mı, nasıl ayırt edebilir?

İkincisi, senaryodaki Türk sistemin sadece bir parçası. Odada aynı zamanda icra ettiği program, kullandığı kâğıt kalem vs. de var ve aşağıda anlatacağım gibi anlama işi için bunlar yaşamsal. Sadece içerideki adamın bir şey anlamadığını söylemek, sizinle konuşan bir insanın sadece bir sinir hücresinin veya kafatasının konuşmayı anlamadığını söylemek kadar yersiz. Burada içerideki adamcağızın değil ama "oda sistemi"nin tümünün Çince bilip bilmediği söz konusu.

Üçüncüsü, bu işlerle uğraşmış birisi olarak size garanti verebilirim ki, Searle'ün anlattığı türden "Şu girdi gelirse, şu çıktıyı ver" komutlarından ibaret bir program hiç kimseyi akıllı bir insan olduğu konusunda kandıramaz. 36. Soru'da bahsettiğimiz gibi bu tip programlar en basit sohbotlara denk gelir ve kendisine saygısı olan hiçbir Çinli'nin adam yerine koyacağı bir performansa erişemez. Zekâ izlenimi verecek kadar gelişmiş bir sohbet programının girdi/çıktı listeleri değil, diyalogdaki öğeleri eşleştirebileceği karmaşık gerçek dünya modelleri bulundurması ve büründüğü kişiliğe uygun anlatılar kurabilmesi gerekir. Odada bu programın mevcut durumu takip edebilmesi için çok miktarda müsvedde kâğıdı bulunması, icra eden kişinin döngülere girmesi, vb. karmaşık süreçler yürütmesi gerekir. Özellikle aritmetikle ilgili basit sohbetlerde odanın içinde oluşan veri yapılarından sohbetin konusu anlaşılabilir. Ama iş bu detayda düşünülünce, Searle'ün çizdiği ikna edici karikatürden eser kalmıyor tabii.

48 | Gödel'in eksiklik teoremi yapay zekâyı olanaksız kılar mı?

Yapay zekâya bir de filozof J. R. Lucas'ın itirazı var. Fizikçi Roger Penrose'un da 1989'da *Kralın Yeni Aklı* kitabında desteklediği bu tezi şöyle özetleyebiliriz:

1) Biz insanlar (matematikçi olanlarımız) Gödel'in eksiklik teoremi (6. Soru) sayesinde, verilen herhangi bir çelişkisiz biçimsel sistem için doğru olan, ama o sistemde kanıtlanamayan bir önerme (buna o sistemin "Gödel cümlesi" denir) kurup bildiklerimiz arasına katabiliriz.

2) Bilgisayarlar bir algoritmayı icra ederken belleklerindeki bilgileri bir adımdan diğerine belirli kurallara göre değiştirdikleri için biçimsel sistemler olarak görülebilir.

3) İnsan matematikçiler çelişkisizdir, yani birbirine zıt cümlelere inanmaz ve yanlış sonuçlara varmazlar.

4) Bu durumda matematikçi bir insanı tümüyle taklit edebilen bir bilgisayar programı çelişkisiz bir biçimsel sistem olacağından asla kanıtlayamadığı ("doğruluğunu göremediği") bir Gödel cümlesi olacaktır.

5) Ama o insanın kendisi 1. maddede belirtilen nedenle o cümlenin doğruluğunu görebilir, yani sözüm ona kendisini tümüyle taklit eden bir makinenin yapamadığı bir şeyi yapabilir.

6) Bu çelişkiden kurtulmak için makinelerin insanları tümüyle taklit edebildiği inancımızdan vazgeçmeliyiz (Penrose insan beyninde henüz bilimin sırrına eremediği fiziksel süreçlerden yararlanıldığı ve bu süreçlerin Turing makinelerince benzetimlenemeyeceği iddiasını ortaya atmıştır).

Bilim dünyasında neredeyse hiç kabul görmeyen bu tezdeki mantık hatası 3. maddede gizli. İnsan matematikçilerin hiç yanlış yapmadıkları da nereden çıktı? Matematikçiler de diğer insanlar gibi çelişkiye düşebilir. Hem de sık sık. Bu tuğla çekilince de Lucas/Penrose tezi çöküyor zaten.

Penrose'un beyindeki fiziksel süreçlerle ilgili savları da henüz bir yere varmadı. Eğer beynimizin klasik bilgisayarların hızla gerçekleştiremeyeceği kuantum hesaplamalar yaptığı anlaşılırsa, bu doğanın büyük ölçekli kuantum bilgisayarlarının var olmasına ve hatta kendi kendilerine evrilmelerine izin verdiği anlamına gelen harika bir buluş olur ve tam teşekküllü hızlı bir YZ oluşturabilmek için bizim de daha büyük kuantum bilgisayarları inşa etmemiz gerektiği anlamına gelir. Ama hem insan beynindeki "çevre koşulları"nın (sıcaklık) kuantum fiziğini klasikten ayıran türden olguların zihin üzerinde kayda değer bir etki göstermesine izin vermediğini düşünenler çoğunlukta, hem de KTM'lerin TM'lere üstün çıkacak gibi göründükleri Çarpan Bulma (21. Soru) gibi yetenekler, atalarımızın maymunlardan ayrıldığı çağlarda hayatta kalmakta pek bir işe yaramadığından bu tür bir donanımın evrilmesinin gerekmiş olabileceğini zannetmiyorum.

49 | Yapay zekâ dünyayı ele geçirip hepimizi yok edecek mi?

Yapay zekâ teknolojisinin kimi konularda insanüstü seviyeye ulaştığına ilişkin haberler neden bazı insanlarda kaygı doğuruyor? Kanımca tarihte Avrupalıların diğer kıtaların (kendilerinden düşük zekâlı gördükleri) yerlilerine, halen de insanların diğer hayvanlara reva gördükleriyle günün birinde insan zekâsını aşan bir varlığın insanlara yapabilecekleri arasında paralellik kuruluyor. Bu ihtimal uykularımızı kaçırmalı mı, bir bakalım.

Önce kaygılananların öngörüsünü Turing'in çalışma arkadaşlarından matematikçi I. J. Good'un 1965'te yayımlanan "İlk Ultrazeki Makine Hakkında Spekülasyonlar" başlıklı makalesine dayanarak özetleyelim: Birçok alanda makinelerin zekâ düzeyi ve becerisi giderek artıyor. Yapay zekâ da zekâ ve beceri gerektiren bir mühendislik konusu.

Bu gidişle günün birinde "YZ sistemleri geliştirilmesi" dalında bizden daha becerikli olan bir YZ geliştireceğiz. Bu YZ bu işte bizden daha iyi olacağı için bizim onu üretmek için harcadığımızdan daha kısa sürede kendisinden daha iyi başka bir YZ geliştirecek. Bu ikinci YZ daha da kısa sürede daha bile iyi olan üçüncü bir YZ geliştirecek, böyle böyle zekâ düzeyi üstel şekilde artacak, yani çok kısa (bazı senaryolarda dakikalar mertebesinde) bir süre içinde anlamamızın mümkün olmayacağı derecede üstün (insanlarla karıncalar arasındaki farkı düşünün) bir zekâ ortaya çıkmış olacak. Bu "zekâ patlaması"nın kimi düşünürlerin deyimiyle bir "teknolojik tekilliğe", yani dünyanın düzeninde detaylarını öngöremeyeceğimiz ve kontrol edemeyeceğimiz devasa değişikliklere yol açacağından korkuluyor. Good'un şu cümlesi pek ünlüdür:

> Yani, bize kendisini nasıl kontrol altında tutabileceğimizi söyleyecek kadar uysal olması kaydıyla, ilk ultrazeki makine insanın yapması gereken son icat olacaktır.[5]

Bunun çok ilgi çekici bir fikir olduğu ortada. ABD'li meslektaşlarım Vernor Vinge (aynı zamanda çok iyi bir bilimkurgu yazarıdır) ve Ray Kurzweil'ın yanı sıra bu konuda en çok kafa yoranlardan biri, "*Süperzekâ*" (*Superintelligence*) kitabının yazarı filozof Nick Bostrom'dur. İnsanoğlunun yeryüzünden silinmesine yol açabilecek "varoluşsal risk"lerin incelenmesi konusunda uzmanlaşan Bostrom, üstün bir YZ'yi bu kategoride görmektedir.

Bostrom'un kitabında tartıştığı birçok felaket senaryosunun ortak yanları şudur: İnsanlar bir süper YZ üretip ona ilk bakışta mantıklı ve zararsız gibi görünen bir amaç verir. Gelgelelim YZ bu amacı gerçekleştirmek için en "verimli" yöntem olarak sahiplerinin hiç aklına gelmeyen ve büyük insani zararlara sebep olacak bir yol keşfeder ve de kısa sürede başımıza gelmeyen kalmaz.

---

5) I. J. Good, "Speculations Concerning the First Ultraintelligent Machine", *Advances in Computers*, Cilt 6, 1965.

Mesela bir ataş fabrikasının patronu, yeni YZ sistemine "Ataş üretimini maksimum seviyeye çıkarmak için bir yol bul ve uygula" komutunu verir. YZ "maksimum" kelimesinin sözlük anlamını dikkate alarak bu amacı gerçekleştirmek için önce gezegenimizde, sonra da evrenin geri kalanındaki tüm hammaddenin (patron dahil tüm insanların bedenlerindekiler dahil, YZ ve kullandığı araçlardakiler hariç tüm atomların) ataş üretmek için kullanılmasının gerektiğini hesaplar. Vardığı bu sonucu açıklamaz (Çünkü çok zeki olduğundan böyle bir niyetinin olduğu anlaşılırsa, insanların onu engellemeye teşebbüs edeceğini bilir) ve sinsi bir planı uygulamaya koyar: Önce patronu çok etkileyecek bir dizi, her biri bir öncekinden daha karmaşık ve daha verimli ataş üretim makineleri tasarlar. Bir noktada makinelerin karmaşıklığı insanların anlama ve inşa gücünü aşar, o yüzden üretim süreçlerinin kontrolü de YZ'ye verilir. YZ bu yeni imkânlarını görünürde daha da çok ataş üretmek için kullanırken bir yandan da arka odada dünyadaki tüm insanları kısa sürede öldürecek bir silah (örneğin yapay bir mikrop türü) imal eder. Türümüzün ortadan kalkıp ataşa dönüştürülmesinden sonra sıra, aynı şeyi evrenin geri kalanında tekrarlayacak robot uzay gemilerinin inşasına gelir.

Bu mantığa göre süper YZ'mize hangi amacı verirsek verelim, o amacı gerçekleştirmek için en verimli yolun önce kontrolü çok daha az zeki olan ve olur olmaz kural ve yasalarla işleri yavaşlatan bizlerden kurtulmaktan ya da en azından açık veya gizli şekilde iktidarı ele almaktan geçtiğini hesaplayacaktır. Al başına belayı!

"Dünyadaki insanların yaş ortalamasını yükselt!" diyerek YZ'mizi hepimizin ömrünü uzatacak yeni ilaçlar, daha güvenli otomobiller vs. icat etmeye yönlendirelim mi dediniz? Bu komutu gerçekleştirmenin en kolay yolunun dünyanın en yaşlı insanı dışında herkesi öldürmek olduğunu hesaplarsa?

Görüldüğü gibi, hedefi onlara biz vermiş olsak bile, belirli hedefleri ve onlara ulaşmak için kullanabilecekleri geniş çaplı kabiliyetleri olan makineler üretirken dikkatli

olmanız gerekir. YZ kıyametinden korkanların temel tezi budur. Sizce haklılar mı?

Sadece belli bir konuda "zekice" davranan ama başka alanlardan hiçbir şey anlamayan "dar YZ" uygulamalarıyla insan bilişsel yeteneklerinin eriştiği her konuda yüksek performans göstermesi öngörülen "genel YZ" arasında ayırım yapmamız gerek. Turing testi, (23. Soru) dile ve konu kısıtlaması olmadan insanların sahip olduğu "hayat bilgisi"nin tümüne hâkim olmayı gerektirdiğinden, sadece bir genel YZ'nin geçebileceği bir sınavdır. Satrançtan tıbbi teşhise birçok alanda başarılı dar yapay zekâlar geliştirildi, ama henüz genel YZ'nin çok uzağındayız (Aslına bakarsanız, özellikle bu konuya adanmış bir bilimsel ekip de yok ortada).

Sorum şu: Yukarıda verdiğim örnekler sizce genel YZ mi sayılırlar, yoksa dar mı? Size de üstün değil, tam tersine geri zekâlı gibi gelmiyorlar mı? Bütün o komploları kuracak kadar zeki olan ataş YZ'si, nasıl patronunun aslında insanlığın yok olmasını hiç istemediğini anlamayacak kadar aptal olabilir?

Özellikle kötü niyet aşılanmamış bir süper YZ'nin (eğer "süper" sıfatını hak edecek kadar gelişmişse) bu tür bir felakete yol açacak yanlış anlamalara düşmeyecek kadar hayat bilgisine de sahip olacağını düşünüyorum. Peki ya gerçekten "kötü" değerlerle donatılmış YZ'ler ne olacak? (Bu noktada 41. Soru'ya bir daha göz atmak iyi olabilir).

Giderek daha çok makineye özerk kararlar verme yetkisi veriyoruz. Kendini süren otomobiller ve borsada hiçbir insanın yetişemeyeceği hızlarda alım/satım yapan otomatik sistemler şimdiden hayata geçti. Bu eğilim hızlanarak süreceği için sistemlerimize gerçek dünyanın büyük karmaşıklığı içinde tasarımcılarının öngöremediği durumlarda bile "doğru" davranmalarını sağlayacak bir "etik" anlayışı programlamamız gerekebilir.

Bu gereklilik, "robotik" biliminin isim babası bilimkurgu yazarı Isaac Asimov tarafından 1942'de fark edilmiştir. Asimov'un kurgu evrenindeki robotlar, fabrikada

imal edilirken beyinlerine "robotiğin üç yasası" silinmez şekilde kazınır:

> 1. Bir robot asla bir insana zarar veremez veya eylemsiz kalarak bir insana zarar gelmesine göz yumamaz.
> 2. Bir robot birinci yasayla çelişmeyen durumlarda insanların emirlerini yerine getirmelidir.
> 3. Bir robot ilk iki yasayla çelişmeyen durumlarda kendi varlığını korumalıdır.[6]

"İşte bu! Artık ataş soykırımı gibi saçmalıklar hakkında kaygılanmamıza gerek kalmadı!" mı dediniz? Aslına bakarsanız Asimov'un her robot öyküsü bu yasalara uymak için çırpınan robotların neden olduğu beklenmedik (ve çoğu da nahoş) olaylarla doludur (Ve sonunda Asimov'un robotları da bu yasaların gereğini sadece iktidarı insanlardan alarak yerine getirebileceklerini fark ederler). Çıkarmamız gereken ders, gerçek hayatın birkaç satırlık bir kural listesiyle başa çıkılamayacak kadar karmaşık olduğudur. Bu robotlar için olduğu kadar insanlar için de zordur (Bu nedenle sürücüsüz arabalar için sorulan "yolun solundaki hamile kadını mı ezsin, sağdaki ihtiyar çifti mi, yoksa onları kurtarmak için ani fren yapıp takla atarak kendi yolcusunun hayatını mı tehlikeye atsın?" gibilerinden imkânsız soruların haksızlık olduğunu düşünüyorum), sonuçta hepimiz değerlerimiz ve kısıtlı düşünce yeteneğimiz (bu kitabın terimlerini kullanırsak, programımız ve hesaplama gücümüz) çerçevesinde elimizden geldiği kadarını yaparız, zaten hayatımızı başka birçok açıdan kolaylaştırarak üstlerine düşeni yapan makinelerden bir de matematiksel olarak erişilebilir olup olmadığı bile bilinmeyen bir ahlak seviyesi beklemek ne kadar gerçekçidir?

Bence gelecekte bu soruda söz ettiğimiz seviyede genel YZ'leri inşa etmemiz mümkün olursa, onlara tıpkı çocuklarımıza davranmamız gerektiği gibi, sevgi, özen ve

6) Isaac Asimov, *Ben Robot*, İthaki Yayınları, 2016.

saygıyla davranmalıyız. Bir gün kötü insanların kötülük için programladığı süper güçlü YZ'lerle karşılaşırsak, bizi savunmak kendi süper YZ'lerimize düşecektir. Umalım ki onlara değerlerimizi olabildiğince kazandırmış olalım. Umalım ki bizi sevsinler ve korusunlar, yaşlılığımızda bizi rahat ettirsinler, sonrasında da belleklerinde olumlu anılar olarak kalalım.

50 | İnsan zekâsının bir geleceği var mı?

Yapay zekânın gelişmesinin şimdiden gözlediğimiz sonuçlarından biri de, gelmekte olan bir teknolojik devrimin yaratması olağan olan kaygılardır: Her geçen gün bazı konularda insanları geçtiğine ilişkin haberler duyduğumuz YZ, bizi işsiz bırakır mı? Mesleğimi bir makine yemeden, içmeden, yorulmadan, grev yapmadan, benden daha ucuza yapabilirse ben hayatımı nasıl kazanacağım? Toplum nasıl değişecek? Haydi buna da cevap verin yapay zekâcılar!

Tarihte insanlığın teknolojideki gelişmelerin dünyayı köklü şekilde değiştirmesi nedeniyle çağ atladığı birkaç devrim yaşanmıştır. Bu dönüşümlerin en önemlileri olan Tarım ve Sanayi Devrimleri insanların yerleştikleri alanlardan diğer canlılarla ve gezegenle ilişkilerine, toplum yapılarından dinlerine varıncaya dek neredeyse her şeyi değiştirdilerse de, ortak etkileri (uzun vadede) insan nüfusunun ve ortalama refahın artması olmuştur. İnsanların işlerini kitlesel olarak makinelere devredeceği yeni bir dönüşümden de aynı mutlu sonucu bekleyebilir miyiz?

Geçtiğimiz yüzyılda tümden makineleşme nedeniyle mesleğini yitiren insanlar arasında asansör operatörleri (eski Amerikan filmlerinde görebileceğiniz, yolcuları istedikleri katlara götürecek şekilde asansörü "süren" üniformalı adamlar), devrilen bowling pinlerini yeniden

dizmekle görevli çocuklar ve ben küçükken şehirlerarası telefon bağlantılarını kurmalarını saatlerce beklediğimizi hatırladığım santral görevlileri sayılabilir. Robotiğin ilerlemesiyle aralarında askerlik, madencilik ve seks işçiliği gibi birçok zahmetli uğraşının da bulunduğu diğer işlerin de bu listeye eklenmesi gündeme gelecektir. Kimileri bir zanaatin ortadan kaybolmasında hazin bir yan görebilirse de, kuşkusuz burada asıl önemli soru, işsiz kalan insanların yeni bir yolla hayatlarını kazanmalarının mümkün olup olmadığıdır. Geleneksel olarak böyle soruları yanıtlamak da topluma önderlik etmeye gönüllü olan siyasetçilerin işidir.

Yakın sayılabilecek bir sürede teknoloji kara taşıtlarının kendi kendilerini sürmesinin yolcular açısından daha hayırlı olacağı bir noktaya gelecektir. Hiç satış elemanının bulunmadığı "insansız" mağazalar, resepsiyonda robotlardan başkasını göremeyeceğiniz oteller şimdiden mevcuttur. Yöneticiler bu teknolojiyi ve getireceği faydaları (örneğin sürücüsüz arabalar sayesinde yüz binlerce insanın trafik kazalarında ölmekten kurtulmasını) reddederek ülkelerini çağın gerisinde tutmak, (2003 Mayıs'ında Irak Ordusunu lağvederek 250.000 öfkeli adamı bir anda sokağa bırakan ABD işgal yönetimi gibi) kitlesel işsizliğe göz yumarak her şeyin kendi kendine yoluna gireceğini ummak ya da bu insanlara alternatif bir geçim yolu bulmak seçenekleriyle karşı karşıya kalacaktır.

Otomasyona geçiş yavaş olursa, toplumun değişimi sindirmesi başta daha kolay olacaktır, ama ya günün birinde, yavaş veya hızlı olarak, insanların çoğunun işinin elden gittiği bir noktaya gelirsek? Bir çözüm önerisi, insanları işsiz bırakacak servislerin ücretlerinden ve yeni teknolojinin zengin edeceği azınlığın gelirlerinden kesilecek vergilerle her vatandaşa yaşamını sürdürmesine yetecek sabit bir aylık ("evrensel temel gelir") bağlanmasıdır. Her vatandaşın bir oyunun olduğu ülkelerde işsiz çoğunluk böyle bir düzenlemeyle sistemin içinde tutulabilir. Demokrasinin olmadığı yerlerde ise ekonominin dışında kalanlar için işler kötüye gidebilir; otomobilin icadından

sonra ABD'deki at nüfusunun düşüşü bu konudaki kitapların tümünde yer alan can sıkıcı bir örnektir.

Aslına bakarsanız, 49. Soru'da sözünü ettiğimiz etik problemlerinin zorluğu ve YZ'yi bu konuda nasıl programlayacağımızı bilemiyor olmamız, olası işsizlik problemine bir çözüm umudu oluşturmaktadır: Özerk YZ sistemlerinin hukuki sorumluluğu, gayet çetrefil bir konudur. Bir sürücüsüz araba birisini ezerse ceza kime verilmelidir? Arabayı satan şirkete mi, belli bir alt sistemini üretene mi, sahibine mi, yoksa arabanın kendisine mi? (Avukat dostlarım arasında gelecekteki YZ'lerin hukuksal açıdan şirketler gibi "tüzel kişi"lere mi, hayvanlara mı, yoksa eski hukuk sistemlerindeki kölelere mi daha çok benzetilebileceği hakkında bir tartışma sürmektedir). Bu zor sorudan kaçınmak için duyduğum en güzel çözüm, Fransa Cumhurbaşkanı Macron'un da dile getirdiği, her sürücüsüz arabaya yine de bir "bakıcı sürücü" bulundurma zorunluluğunun getirilmesi fikridir. Bu görevi yapacak insanın tek işi, o koltukta oturup izlemek, bir kaza tehlikesi oluştuğunda da kontrolü ele almaktır. Teknoloji iyileştikçe böyle müdahalelere neredeyse hiç gerek kalmayacak, bakıcı sürücü de maaşını aslen az ihtimalle oluşacak bir sorunun sorumluluğunu üstlenme sözü karşılığında alıyor olacaktır.

Algılardan gerçeğe ulaşma, yeni koşullara uyum sağlama, problem çözme yetenekleri bütünü olarak "zekâ" sadece bireylere değil ülkelere de lazım olduğundan, kimi devlet görevlilerinin işinin de bakıcı sürücülüğe benzer konuma indirgeneceği bir değişim öngörülebilir: Yargı hizmetinin mevcut kalite seviyesi göz önüne alındığında, kimi insanlardansa hukuk kurallarını eksiksiz kodlayabileceğimiz makinelerce yargılanmayı tercih edecek çok kişi tanıyorum. Gerek iş kapasitesi, gerekse de kurallara uyma doğruluğu açısından ortalama bir insan yargıcın performansının çok üstünde robot yargıçlar imal edilebilir. Bunlar yasadan başka hiçbir dayanağı olmayan kararlar verip tümüyle mantıksal şekilde gerekçelendirebilir, hatta yasalar arasındaki çelişkileri saptayıp raporlayabilir.

Bu gibi işleri bilgisayarlara yaptırmak teknik açıdan da özel bir zorluk içermiyor. Satranç gibi kimi oyunlarda tek başına bir yapay zekânın bir insan şampiyonu yenebildiği, ama bilgisayarla insandan oluşmuş bir ikiliyi (bunlara mitolojiden ve Harry Potter kitaplarından bildiğimiz "at-adam" adı verilir) alt edemediği gözlendiğinden, mahkeme heyetleri yapay zekâlara ek olarak kalifiye insan hâkimlerden oluşacak, yanlış kararlardan doğacak sorumluluk ise tümüyle insan üyelere ait olacak şekilde kurulabilir.

Aslında yapay zekâ kanunların sadece uygulanmasında değil, yazılmalarında da katkı verebilir. Yasa dediğimiz şey, sonuçta hem kendi içinde, hem de diğer yasalar ve dünya gerçekleriyle mantıksal olarak tutarlı olması gereken biçimsel bir metindir. Bu tür kural zincirlerindeki kimi tutarsızlıkları tespit edebilen algoritmalar mevcuttur. Örneğin "Elektrik tasarrufu için ülkenin saat dilimini bir saat ileriye alalım" gibilerinden bir düzenleme önergesi getirildiğinde, bunu iş ve okul başlama saatlerindeki insan davranışlarını göze alarak değerlendirebilecek, sonuçta "Bu durumda Edirne ile Rusya'daki Dağıstan Cumhuriyeti aynı saatte olacaklar. Oysa Edirne'de güneş aynı enlem üzerindeki Dağıstan'dan yaklaşık bir buçuk saat sonra doğar. İnsan biyolojisi ve tasarruf hedefi açısından bu iki yörede mesai aynı saatte başlıyorsa, birinden biri hata yapıyor demektir" türünden bir mantık zinciri ile kural koyucuları uyarabilecek bir karar destek sistemi büyük fayda getirebilir. Doğal veya yapay, diğer tüm sahalarda olduğu gibi, yönetimde de zekânın azı değil, çoğu yararlıdır.

Bilimdeki gelişmeler bir bütün olarak değerlendirildiklerinde, şimdiye dek insanlığın durumunu hep olumlu yönde değiştirdiler. YZ teknolojisinin 21. yüzyılda benzer devrimler yaşamakta olan birkaç diğer teknolojiyle birlikte hepimizi çok daha iyi noktalara getirme potansiyeli olduğu çok açık. Akıllı telefonlarını akıllıca kullanabilen insanların diğerlerinden daha rahat ve verimli hayatlar sürdürebildiği ortada. Eski kuşaktan olan bizler

gençlerin telefonlarını ellerinden bırakamamalarına bozuluyoruz, ama sanırım bedenin küresel iletişim bağlantısı ve çıplak bir bireyin asla sahip olamayacağı boyutta bilgi ve zekâ sağlayan bir tür ek beyinle zenginleştirilmesi insanlığın yeni sürümleri için normal olacak. Tabii ki böyle bir altyapının baskıcı yönetimlerce insanları eşi görülmedik derecede sıkı bir kontrol altında tutmak için kullanılması, çoğu insanın tam kapasitede işletmeye daha az gereksinim duyacağı biyolojik beyinlerin körleşip ortalama zekâ düzeyinin düşmesi vs. bir dizi kötü gelecek senaryosu yazılabilir. Ama ben iyimserim. Yaşamak çok güzel, her yıl yeni bir gençler nesli bu olağanüstü macerada aramıza katılıyor ve içlerindeki merakı ve masumiyeti bize bulaştırıyorlar. Evren çok büyük ve gizemli, ama uğraşırsak onun dilini anlayabileceğimizi ve sırlarını çözebileceğimizi fark ettik. Hayatta en hakiki mürşidin fen olduğunu anladık. Hem gezegenimizi, hem de evrenin geri kalanını cennete dönüştürmek, yeryüzünü de gökyüzünü de aşkın yüzü yapmak için daha çok zekâya ihtiyacımız var. Başarabiliriz.

OKUMA ÖNERİLERİ

- Ethem Alpaydın, *Machine Learning*, The MIT Press, 2016.
- Isaac Asimov, *Ben Robot*, İthaki Yayınları, 2016.
- Nick Bostrom, *Superintelligence: Paths, Dangers, Strategies*, Oxford University Press, 2014.
- Ivan Bratko, *Prolog Programming for Artificial Intelligence*, Fourth Edition, Addison Wesley, 2012.
- Martin Davis, *The Universal Computer: The Road from Leibniz to Turing*, W. W. Norton & Company, 2000.
- Daniel Dennett, *Darwin'in Tehlikeli Fikri*, Alfa Yayınları, 2014.
- Daniel Dennett, *Bilinç Açıklanıyor*, Alfa Yayınları, 2017.
- Daniel C. Dennett, *From Bacteria to Bach and Back - The Evolution of Minds*, W. W. Norton & Company, 2017.
- Apostolos Doksiadis, Hristos H. Papadimitriu, *Logicomix*, Albatros Kitap, 2012.
- Pedro Domingos, *Master Algoritma*, Paloma Yayınevi, 2017.
- Andrew Hodges, *Alan Turing: The Enigma*, Vintage Books, 1992.
- Douglas R. Hofstadter, *Gödel, Escher, Bach: Bir Ebedi Gökçe Belik*, Pinhan Yayıncılık, 2011.
- Douglas Hofstadter, *Ben Bir Garip Döngüyüm*, Alfa Yayınları, 2015.
- Ray Kurzweil, *The Singularity Is Near: When Humans Transcend Biology*, Penguin, 2006.
- Seth Lloyd, *Programming the Universe*, Vintage, 2007.
- Michael E. Nielsen, *Neural Networks and Deep Learning*, Determination Press, 2015.
- Nils J. Nilsson, *Yapay Zekâ: Geçmişi ve Geleceği*, Boğaziçi Üniversitesi Yayınevi, 2018.
- Cathy O'Neil, *Weapons of Math Destruction*, Crown, 2016.
- Charles Petzold, *The Annotated Turing: A Guided Tour Through Alan Turing's Historic Paper on Computability and the Turing Machine*, Wiley, 2008.
- Roger Penrose, *Kralın Yeni Aklı*, Koç Üniversitesi Yayınları, 2015.
- Steven Pinker, *Zihin Nasıl Çalışır*, Alfa Yayınları, 2017.
- Stuart Russell, Peter Norvig, *Artificial Intelligence: A Modern Approach*, Third Edition, Pearson, 2009.
- Max Tegmark, *Our Mathematical Universe*, Penguin, 2015.
- Max Tegmark, *Life 3.0: Being Human in the Age of Artificial Intelligence*, Allen Lane, 2017.
- Sara Turing, *Alan M. Turing*, Cambridge University Press, 2012.
- Vernor Vinge, *A Fire Upon the Deep*, Tor, 1993.

DİZİN

A

∀ ve ∃ 21
AARON 110
Acı 159-162
Ackermann, Wilhelm 31
Adalet duygusu 157
Akbay, Bager 111, 112
Akınerkez 149
Alan Turing kanunu 30
Alexa 126
Algoritma 9, 31, 32, 34, 36-39, 42, 43, 46, 52, 61-65, 67, 69-71, 73, 76, 78-80, 88, 95, 96, 98, 100-102, 104, 105, 111, 116, 118, 134, 135, 141, 146, 152, 154, 165, 174
Algoritmik enformasyon kuramı 60
ALİ 127-130
Alibaba 155
AlphaGo 13, 14, 117-119
AlphaGo Zero 119
Alt katmandan bağımsızlık 42, 45
Amazon 126
Anahtar kelime çıkarımı 114
Analitik motor 106
Anne sevgisi 157
Apple 14, 125
Arama 88, 94,
Arama motoru 95, 113, 114, 142
Arama uzayı 95
Aritmetiğin Temel Yasaları *22, 23*
Arkhipov, Vasili 144
Asimov, Isaac 120, 169, 170
Askeri uygulamalar 143, 144
Aşk 156, 157, 159
Ataş yapan YZ 168-170
Ayrıksılık saptama 141
Aytekin, Çiğdem 123

B

Babbage, Charles 106
Bağlantı ağırlıkları 54, 98, 100-102, 163
Bağlantıcılık 89, 99
Belit 21-24, 26, 31
Bellek karmaşıklığı 65-69
"Ben" 56, 162
Benzetim 34, 38, 43, 44, 47, 55, 56, 65, 67, 68, 70, 71, 79, 85, 94, 140, 165
Berners-Lee, Tim 14
Beşinci Kuşak Bilgisayar Projesi 89

Beyin 12, 50-54, 98, 99, 104, 156, 157, 160, 161, 165, 166, 175
Biçimbirimsel çözümleme 128
Biçimsel sistem 24, 27, 165
Bilgi 57, 58
Bilgi işlem 37, 42, 51, 66
"Bilgi İşleyen Makine Olarak Beyin" toplantıları 12
Bilgisayar 11-15, 20, 29, 35, 38, 42-47, 49, 51-54, 56, 60, 64-68, 70, 74-91, 94, 95, 97, 99, 100, 104-106, 108-110, 112-118, 120, 122, 123, 125, 127-129, 131, 133-135, 139, 140, 147, 151-154, 156, 158, 159, 161, 163-166, 173, 174
Bilgisayar bilimi 28, 31, 39, 41-43, 45, 57, 61, 71
Bilgisayar mühendisliği 11, 121
Bilinç 56, 163
Bilişim devrimi 12
Bir Kelime, Bir İşlem 82
Bit 58, 59, 66, 77, 78, 80, 101
Bletchley Park 29
Boole, George 18-21, 101, 104
Boole, Mary 20
Boston Dynamics 120, 121
Bostrom, Nick 167
Bratko, Ivan 13
Brown, Gordon 30
Brünn Doğa Araştırma Derneği Dergisi 113
Bulut, Burcu 134
Bursalı, Orhan 30
Büyük veri 105, 113, 127, 130, 140, 152, 154
Byron, Lord 106

C

Calculus ratiocinator 16
Cambridge Analytica 142
Čapek, Karel 120
Cerrahi 147, 161
Chaitin, Gregory 60
Chomsky, Noam 133, 134
Church, Alonzo 31-32, 34
Clay Matematik Enstitüsü 73, 74
Clinton, Hillary 142
Cohen, Harold 109, 110
Colton, Simon 110
Cope, David 110
CYC 127

Ç

Çalışma süresi fonksiyonu 63, 64
Çarpan Bulma Problemi 79, 80
Çarpma Problemi 62, 64
Çelişki 21-24, 26-28, 31, 165, 173
Çelişkisizlik 24-28, 31, 165
Çerez 140
Çetinoğlu, Özlem 129
Çeviri 18, 88, 89, 103, 133-137, 150, 151, 153, 154
Çıkarım kuralı 21, 22
Çıktı 42, 44, 45, 47, 50, 51, 60, 98-102, 105, 115, 122, 136, 153, 163, 164
Çıktı katmanı 99, 100, 102, 103
Çince Odası 163
Çözümsüz problemler 36-39

D

DART 90
Dartmouth Yapay Zekâ Yaz Araştırma Projesi 85, 86
Deep Blue 12, 13, 71, 90, 117
DeepMind 13, 119
Demir, Şeniz 129
Demirci, Gökalp 77
Dennett, Daniel 163
Derin öğrenme 13, 91, 103-105, 119, 122, 136
Dil 55, 57
DNA 49, 50
DoDAAM 143
Doğal dil anlama 114, 127-133
Doğal dil işleme 123-137
Domingos, Pedro 146
Donanım 45, 46, 57, 121, 166
Dostluk 157
Durma Problemi 38, 39, 66, 70
Duygu 110, 156-159, 162
Duygu analizi 125

E

Einstein, Albert 25, 26, 37
Ekonomi 14, 87, 89, 157, 172
Eksiksizlik 26, 27, 30
ELIZA 123, 124, 126
Elizabeth, II. (Kraliçe, II. Elizabeth, tam adıyla Elizabeth Alexandra Mary Windsor) 30
EMI 111
Enformasyon 57-61
Enformasyon kuramı 58, 60
En hakiki mürşit 175
Enigma 29
En Uzun Yol Problemi 73
Eski moda YZ 89, 93-96, 113, 116, 126, 130, 135, 145
Etik 173
Eurisko 107-109
Evrensellik 34, 35, 38, 43, 79
Evrim 47-50
Evrimsel programlama 97
Eyleyici 53, 120

F

Facebook 102, 105, 112, 134, 141, 142
Fisher, Richard 14
Fizik yasaları 46-48, 56, 66, 78, 162
Frege, Gottlob 21-24, 86
Fürer, Martin 64
Fürer çarpma algoritması 64

G

Gen 48-50, 94, 146, 156, 158
Genelleştirilmiş satranç 71
Gezegen 45, 47, 48, 120, 137-139, 168, 171, 175
Girdi 33-35, 37, 38, 42, 43, 45-47, 51-53, 61-63, 66, 70-72, 75, 88, 96, 110, 115, 120, 122, 125, 136, 142, 152, 163, 164
Girdi katmanı 99, 100, 102
Gizli katman 103
Go 13, 90, 105, 117-119
Goldbach, Christian 107
Good, I. J. 166, 167
Google 13, 90, 105, 113, 114, 125, 131, 135-137, 139, 140, 144, 149
Google Çeviri 134-137, 150, 151, 153
Google Haritalar 149

Gödel, Kurt 25-32, 37, 73
Gödel cümlesi 165
Gödel'in eksiklik teoremi 27, 31, 165
Gözetimli öğrenme 101, 114, 122
Gözetmensiz öğrenme 114
Grafik işlem birimi 104

H

Hasmane örnekler 151
Hatanın geri yayılımı 101, 102, 104
Haugeland, John 93
Hayalet uzuv 160
Hesaplama 8, 13, 29, 32, 33, 35, 42, 43, 47, 52, 54, 55, 62, 65, 71-73, 76-79, 83, 86, 93, 94, 99, 104, 108, 116, 136, 146, 160, 162, 166, 170
Hesaplama karmaşıklığı kuramı 62, 65, 66, 71, 74, 86
Hilbert, David 24-27, 29-31, 38, 39
Hinton, Geoffrey 101, 104
Hislere kanmak 159-162
Hukuk 15, 26, 111, 113, 115, 173
Hükümet Kod ve Şifre Okulu 29

I

IBM 12, 115, 130, 145
ISAAC 127, 128, 130

İ

İkili sistem 20
İletişim karmaşıklığı 66
İlkokul çarpma algoritması 61-64
İndirgeme 39
İngilizce 42, 57, 58, 87, 88, 124, 127, 128, 133-137, 150, 153
İşsizlik 171-173

J

Japonca 137
Jelinek, Fred 130
Jovovich, Milla 152

K

Kanıt 21, 22, 24-28, 30, 31, 35, 73, 89, 103
Kanser 145-147, 163
Karar destek sistemi 174
Kararlaştırılamazlık 39
Karar Problemi 31,32, 35, 38, 39, 73
Karatsuba, Anatoli 62, 64
Karatsuba çarpma algoritması 62-64
Karayolları Trafik Kanunu 46
Karmaşıklık sınıfı 66, 67
Kasparov, Gari 12, 13, 71, 90, 115, 117
Kazanma stratejisi 71, 116
Kepler 138, 139
Keşifsel çözümleme 141
Kı, Cie 14
Kırılganlık 89, 130, 135
Kireçlenme 146
Klik Problemi 73
Kolmogorov, Andrey 60, 62
Kombinasyonlar patlaması 88, 95
Konuşma tanıma 105, 125, 130
Korece 137

Korku 156-158
Köksal, Aydın 127
Kötülük sorunu 17
Kötü YZ 166-171
Kraliyet Akademisi 16, 17, 29
Kuantum bilgisayarı 77-80, 166
Kuantum fiziği 77, 78, 80, 163, 166
Kuantum Turing makinesi (KTM) 78, 79, 166
Kullanıcı 15, 45, 52, 56, 66, 102, 110, 123-125, 130, 140-142, 145, 151, 159
Kurzweil, Ray 167
Küme 19, 20, 23, 36, 58, 59, 61, 66, 68-70, 73-75, 101, 102, 105, 106, 107, 122, 126, 128, 131, 139

L

Lego 49
Leibniz, Gottfried Wilhelm 15-19, 21, 29, 31, 42, 112
Leibniz'in hesap makinesi 16, 42
Lenat, Douglas 106-109, 126
Lisp 86, 89, 90
Logicomix 24
Lovelace, Ada 106
Lucas, J. R. 165

M

Macron, Emmanuel 173
Mantık 18, 22, 24, 26, 43, 57, 86, 87, 95, 96, 99, 112
Mantıkçı yaklaşım 87
Mantık Kuramcısı 87
Master Algoritma 146
Matematik İmha Silahları 154
McCarthy, John 85, 86
Mendel, Gregor 113
Merkezi işlem birimi 52, 104
Metin madenciliği 113
Microsoft 151
Mind 83
Minimum tarif uzunluğu 60
Minsky, Marvin 85, 87, 89, 95
Monte Carlo ağaç araması 118, 119
Morgenstern, Oskar 25, 26
MYCIN 145

N

NASA 138
Nature 13, 14
Navigasyon 150
Netflix 141
Newman, Max 31
Newton fiziği 77
Newton, Isaac 17
Nimbursky, Adele 28
Novak, Gordon 127
NP 73, 74, 79

O

Olasılıksal TM (OTM) 75, 76, 78
O'Neil, Cathy 154
Otomatik Matematikçi 106, 107
Otonom robot 45, 120
Oyun ağacı 116-118, 162
Oyun kuramı 25, 157

Ö

Öğün, Fatih 129
Önermeler mantığı 21
Önyargı 153, 154
Örüntü tanıma 96, 102, 104, 146, 151
Ötegezegen 138-139
Öz kopyalayıcı 48, 50
Özgür irade 162
Öznel deneyim 160, 163

P

P 68, 69, 70, 74, 75
Pascal 127
Pascal, Blaise 16
Pascal'ın hesap makinesi 16
Pangloss 17
Papert, Seymour 89
Paralel işlem 65, 78
Pekiştirmeli öğrenme 118, 119
Penrose, Roger 165
Pepper 159
Perceptron 89
Perelman, Grigoriy 74
Petrov, Stanislav 144
Poincaré sanısı 74
Polinom Özdeşlik Testi Problemi 75
Polinom sınırlı artış 68, 71-73
Principia Mathematica 24, 25, 27, 87
Problem 24, 32, 35-39, 42, 55, 58, 61, 65, 69, 74, 76, 79, 85-87, 127
Program 13-15, 33, 38, 42-47, 49, 52, 55, 60, 70, 82, 85-87, 95, 97, 145, 149, 163, 164
Prolog 94, 95, 127
Protein 49
PSPACE 68, 69, 73, 79

R

Resim Yapan Budala 110
Robot 15, 44, 45, 52, 53, 84, 90, 94, 111, 113, 120-124, 143, 144, 147, 156, 158-162, 168-170, 172, 173
Robot hakları 162
Robotiğin üç yasası 170
Robotik 120, 121, 169
Robot yargıç 173
Rochester, Nathaniel 85
Rosenblatt, Frank 89
ROSS 115
Rota Bulma Problemi 76, 77
Rowling, J. K. 141
Rudisch, Gloria 87
Ruh 46, 161
R. U. R. 120
Russell, Bertrand 22-25, 86, 87

S

Sağduyu 77, 126, 127, 129, 130
Sakıncalar 152
Samsung 143
Sanayi Devrimi 171
Satranç 12, 13, 71, 82, 87, 90, 95, 115-117, 162, 169, 174
Searle, John 163, 164
Sedol, Li 14
Sepsis 146
Seyyar Satıcı Problemi 72, 74
SGR-A1 143
Shallue, Chris 139

Shannon, Claude 58, 59, 61, 85
Sibernetik 93
Silahlı İnsansız Hava Aracı (SİHA) 144
Simgecilik 89, 93, 95, 99
Simon, Herbert 87, 90
Simülasyon 34, 44 (Ayrıca bkz. **Benzetim**)
Sinir ağı 98-102, 103, 119, 122, 136, 137, 139, 147, 150, 163
Sinir hücresi 51, 53, 56, 89, 98, 100, 160-162, 164
Siri 125
Sohbot 124-126, 154, 164
Solomonoff, Ray 60, 85
Sosyal kredi sistemi 155
Sözdizimsel çözümleme 129
Suçluluk duygusu 157
Super aEgis II 143
Süperzekâ 72, 76, 77, 167
Sürücüsüz otomobil/araba 121, 144, 152, 158, 170, 173

T

Taklit oyunu 83
Tarım Devrimi 171
Tatou, Audrey 110
Tay 151
Tegmark, Max 47
Tekillik 167
Temel parçacıklar 24, 44, 46, 77
Teorem 22, 27, 28, 31, 37, 62, 87, 101, 103, 165
Tesla 122
Teyp 32-34, 50, 52, 65
Teyp kafası 33, 50, 52
Tiksinme 157
Toplama Problemi 36, 61, 62
TOY 129-130
Traveller 107-108
Trump, Donald 142
Turing, Alan 8, 28-32, 35, 38, 57, 61, 65, 73, 75, 83-86, 91, 99, 106, 123, 125, 159, 166
Turing makinesi (TM) 30, 32-34, 38, 42-44, 46, 49, 50, 52, 53, 60, 64, 65, 67, 75-80, 95, 166
Turing testi 83, 84, 91, 112, 163, 169
Tüfekçi, Zeynep 153
Türkçe 82, 101, 123, 127-131, 134, 135, 150, 151, 163
Twitter 151

U

Uluslararası Astronomi Birliği 137
Uluslararası Matematikçiler Kongresi 24

Ü

Üstel artış 68-69

V

Varoluşsal risk 167
Vektör 131-133, 136
Veri bilimi 140, 146
Vicdan 156, 157
Vinge, Vernor 167
Voltaire (François Marie Arouet) 17
Von Neumann mimarisi 52

W

Watson 115
Watson Oncology 145
Weizenbaum, Joseph 123
Whitehead, Alfred North 24, 25, 87
www 11

Y

Yakaryılmaz, Abuzer 76, 80
Yapay öğrenme 90, 91, 96, 98, 101, 103, 104, 113, 114, 118, 140, 146, 147, 152-154
Yaşamın doğuşu 47-50
Yazılım 38, 45, 94, 97, 114, 120, 121, 124, 153
Yılmaz, Deniz 111, 112
YouTube 141
Yüklemler mantığı 21
YZ kışı 88, 90

Z

Zaman karmaşıklığı 64, 65, 88
Zekâ 55
Zekâ patlaması 167
Zombi 161



yapay zekâ

Cem Say

Boğaziçi Üniversitesi Bilgisayar Mühendisliği Bölümü öğretim üyesi Cem Say, bu kitapta yapay zekâ kavramını her yönüyle ve herkesin ilgisini çekebilecek biçimde anlatıyor. Yapay zekâ sistemlerinin dayandığı matematiksel altyapıyı, gelişim serüvenini, insan beyni dediğimiz doğal bilgisayarın çalışması hakkında verdiği ipuçlarını, güncel uygulamalarının nasıl çalıştığını ve nerelerde tıkandığını, yarattığı felsefi tartışmaları, bilinç ve özgür iradeyle ilgisini ve insanlığa olumlu/olumsuz etkilerini keyifli bir dille işliyor. Yazarın, yüzyıllar öncesinden başlayıp son teknolojik gelişmelere varan bir çerçeve kurarken yanıtını aradığı kimi sorular şöyle:

İnsanların yapabilip makinelerin yapamayacağı şeyler var mı? Düşünen bir makine yapılabilir mi? Sadece 0 ve 1 her şeye yeter mi? Gödel neden öldü? Turing kimdir? Doğanın, yaşamın, insanların programlama dili nedir? Hesaplama karmaşıklığı nedir? AlphaGo dünya go şampiyonunu nasıl yendi? Turing testi nedir? Bilgisayar insan dillerini nasıl anlar? Derin öğrenme nedir? Robotlar buluş, sanat, avukatlık, askerlik, doktorluk vs. yapabilir mi? Âşık olabilir mi? Bilgisayarlar bizi bizden iyi tanıyabilir mi? Yapay zekâ dünyayı ele geçirip hepimizi yok edecek mi?



Bilim ve Gelecek Kitaplığı

ISBN 978-605-5888-58-9

