v.0.1

Mesterséges Intelligencia

mindenkinek

rixel


BY NC ND

Tartalomjegyzék

Bevezetés

Ez a könyv áttekintést kínál a mesterséges intelligenciáról. Az írás során arra törekedtünk, hogy mindenféle előismeret nélkül is megérthető tudást adjunk át mindenkinek, aki érdeklődik a mesterséges intelligencia iránt vagy egyszerűen csak szeretne tisztábban látni egy felkapott, népszerű témában. Azok számára, akik egy vállalkozás tulajdonosaként vagy döntéshozóként érdeklődnek a mesterséges intelligencia iránt, az ugyanilyen címen a „vállalkozók, vállalatok" számára készített változatot ajánljuk. Fejlesztők vagy leendő fejlesztők számára pedig a „fejlesztőknek" készült változatot ajánljuk.

Mi a rixelnél úgy látjuk, nagyon sok kérdőjel van az emberekben a mesterséges intelligencia alkalmazását, alkalmazhatóságát illetően. Személyes, üzleti tapasztalataink inspiráltak bennünket arra, hogy megmutassuk, fejlesztőként hogyan látjuk a mesterséges intelligenciát és mit gondolunk annak felhasználhatóságáról. Szerintünk a megértéshez hozzátartozik az is, ha egy témáról képesek vagyunk bonyolult szakkifejezések mögé bújás nélkül, közérthetően beszélni. Erre teszünk kísérletet a könyvben.

A könyv szerkesztése során törekedtünk a logikus felépítésre és az olvasmányos végeredményre. Természetesen az előfordulhat, hogy egyes fejezetek nem mindenkit egyformán érdekelnek. Ajánljuk, hogy a könyvet folyamatosan, fejezetről fejezetre olvasd, de amennyiben úgy látod, téged nem érdekel az adott kérdés, nyugodtan ugord át. Ezt több helyen segítjük azzal, hogy utalunk korábbi vagy későbbi tartalmakra, így az esetlegesen kihagyott fejezetekre vissza tudsz térni, ha szükségét érzed.

Ez a könyvváltozat mindenkinek szól, így óhatatlanul előfordulhat, az olvasás közben felmerül benned az igény a tudományosságra, másfajta példákra, programkódokra, vagy éppen még egyszerűbb megfogalmazásokra. Mivel elsődleges célunk volt, hogy a megértéshez ne legyen szükség előismeretre, különleges tudásra, így a nyelvezetet is ehhez igazítottuk. Amennyiben úgy érzed, másfajta, mélyebb tudást szeretnél szerezni, úgy ajánljuk figyelmedbe e bevezető fejezet elején is említett egyéb változatokat, melyek a vállalkozóknak és vállalatoknak, vagy a fejlesztőknek szólnak.

A visszaérkező tapasztalatok és az ipari fejlődés miatt szükséges lehet a könyv frissítése, aktualizálása. A szoftveriparból megszokott módon ezért a könyvünket verziószámmal láttuk el, melyet a borító jobb felső sarkában látsz. Amikor egy újabb változatot készítünk, megemeljük ezt a számot. Kis változtatás esetén csak a második számot, nagy változás esetén pedig a bal oldali, úgynevezett főverziót növeljük.

Mesterséges intelligencia a – „a nagy kép”

A mesterséges intelligencia alatt olyan programot értünk, amely az emberi tudatot „másolja”. Két olyan képesség van az emberi gondolkodással kapcsolatosan, amiket a fejlesztők át akarnak ültetni a gépi világba. Az első a tanulás, a másik a problémamegoldás. Egyszerűen fogalmazva, ezek azon képességek, amelyek IQ-teszttel mérhetők, illetve amelyeket az iskolás évek alatt próbálnak fejleszteni a diákokban. Érdemes leszögezni, a gép alapvetően olyan problémák megoldásában jeleskedik, amelynél nagy mennyiségű tanulható adat áll rendelkezésre. Többféle tanulási módszer létezik, de mindegyikről általánosan elmondható, a számítógépet irányítani kell. Mindig rögzíteni kell azt, hogy az adott tanulásnak mi legyen az eredménye. E tekintetben a gépek sokkal korlátozottabbak, mint az emberek, hiszen nekünk rendelkezésre áll többek között érzelmi, vagy társas intelligencia is, melyek segítségével könnyebben meg tudjuk érteni a környezetünket és a körülöttünk lévő embereket. Korlátozott módon ezek a képességek tanulhatók, így akár mi emberek is jobbá tudunk válni. A gépek így e behatárolt területen képesek tanulni.

Ezen részterületeken az emberek képességeit jelentősen meghaladó eredményeket érhetnek el. A betanulás ideje jelentősen rövidebb, hiszen a gép esetén órákról, napokról, vagy esetleg hetekről beszélhetünk olyan területeken, amelyet az emberek évekig tanulnak és gyakorolnak. Az eredményesség tekintetében is a gépek oldalára billen a mérleg. Fontos hangsúlyozni, ezek részterületek, nem általános értelemben vett tudatról beszélünk. Attól, mert egy önvezető autó sokkal hamarabb észleli a veszélyhelyzetet, sokkal biztonságosabban végzi az adott manővert, önerőből nem fogja kikerülni a dugóban előtte lerobbant autót. A közlekedési dinamika addig tart, hogy követjük vagy előzzük az előttünk lévő járművet. A jelenlegi rendszerek nincsenek felkészítve arra, hogy a közlekedés többi résztvevője nem rendeltetésszerűen közlekedik. Igaz, az autó nem is lesz ideges, amikor az előtte haladó a zöld jelzés ellenére nem indul el.

A mesterséges intelligencia egy gyűjtőfogalom. Mindegyik alrendszere más módon közelíti meg a megoldandó problémát, vagy eltérő módon adja vissza a kívánt eredményt. Ezek alapján beszélhetünk olyan rendszerekről, amelyek statisztikai összefüggéseket keresnek. Léteznek azonban az emberi agy felépítését alapul vevő rendszerek. A túlmisztifikálás itt sem

helyes. Az emberi agyban az ingerek egy úgynevezett neuronokból álló hálózaton keresztül haladnak és dolgozódnak fel. Ezt úgy kell elképzelni, mint egy várost. Amennyiben el akarunk jutni a város másik pontjába, akkor mehetünk többféle úton, több különböző kereszteződést, csomópontot érintve. Annál könnyebb eljutni két pont között, minél több út épült ki. A város annál felkapottabb lesz, minél szervezettebb a közlekedési struktúrája. A neurális hálók is annál „okosabbak", minél több adatpontot és ezek összeköttetését tartalmazzák. A méretük azonban nagy hatást gyakorol a hatékonyságukra is. A kisebb hálózatok gyorsabbak, de a betáplált információk között jóval kevesebb kapcsolat felderítésére képesek. Érdemes azon elgondolkodni, hogy a gépek esetén néhány ezer ilyen adatpont és összekötés alkalmas arra, hogy jobban vezessenek, mint az ember. Nekünk több milliárd ilyen neurális kapcsolatunk van. Felmerül a kérdés, mire lehetne képes egy gép akkor, ha képesek lennénk az emberét legalább részben közelítő rendszert felépíteni. Jelenleg azonban ez a tudományos fantasztikum és a jövőkutatók területe és nem a mindennapok része.

A számítógépes agyak képesek túltanulni magukat, ami az embernél jelen tudásunk szerint nem lehetséges. A jelenség azt jelenti, hogy a gép a megtanulandó

adatokból olyan következtetéseket von le, olyan összefüggéseket tár fel, amelyek csak az adott mintázatra jellemzőek, általánosan nem használhatók. Tételezzük fel, kutatásokból tudjuk, hogy a moziban az extrém hajszínű emberek több kukoricát vásárolnak, mint a természetes hajszínű társaik. Ehhez készíteni akarunk egy neurális háló alapú képfeldolgozó rendszert, amely jelez a büfé munkatársainak, ideje megtölteni a nagyobb kosarakat kukoricával, esetleg automatikusan átkapcsolni a pult felett a menükről a különböző ízesítésű pattogatott kukoricákra. Ha a tanulás folyamán csak kék és lila színű hajakat mutatunk, akkor a gép ezt nagyon jól fel fogja ismerni, azonban nem jelez be sem a zöld, sem a pink festésre. Ez akkor is előfordulhat, ha mind a négy színből kap mintát, de pinkből és zöldből olyan keveset, hogy egyszerűen azt figyelmen kívül hagyja. A tanulás alapját képező adathalmaz összeállítása ezért külön tudománynak tekinthető. Habár e jelenségnek nincs humán változata, hogyha mégis emberi oldalról szeretnénk megközelíteni, akkor a „szakbarbár" kifejezés lenne a legközelebbi. Ezt olyan társainkra használjuk, akik tudása és érdeklődése kizárólag a szűken vett feladatukhoz, hivatásukhoz kapcsolódik és az egész világot ezen a szűrőn keresztül szemlélik. Abraham Maslow közgazdásznak tulajdonítható az a gondolat, hogy ha a kalapács az egyetlen szerszámod,

minden problémát szögnek nézel. A gép a túltanulás miatt elkezdi elveszíteni a szerszámait, míg végül csak a kalapács marad a kezében. Innentől a valóság minden problémáját szögként azonosítja és megpróbálja ráhúzni a tanultakat akkor is, ha azok nem is illeszkednek.

Az algoritmusok fejlesztői rájöttek erre a problémára, így több különböző módszer van, amellyel elkerülhető a túltanulás, vagy eredeti kifejezéssel élve az overfittelődés. A fejlesztők (programozók) szándékosan „eltakarják" a gép elől a tanulandó adatok egy részét, vagy a neuronokat kapcsolgatják ki és be. A neuronokról már volt szó korábban, ezek a neurális háló alapját képező elektronikus idegsejtek. Ezzel elérhető, hogy a tévesen megtanult ismeretek súlya csökkenthető, hiszen a gép rájön a saját hibájára.

A hibás tudást nem úgy kell elképzelni, hogy a gép gondolkodás útján rájön, hiszen ő nem tud gondolkodni. Amikor a hűtlen házastársat lebuktatja a viselkedése, akkor a megcsalt fél a jeleket köti össze: lezárja a telefonját, nem hagyja elől a kommunikációs készülékeket, furcsán viselkedik, jelszót változtat. Mi emberek képesek vagyunk ezekből következtetéseket levonni. Az csak a gondolkodásunkban gyökerező hiba, hogy általában ezeket biztosnak is fogadjuk el, így például előfordulhat, hogy a hatalmas születésnapi

meglepetésbulit előkészítő felet éri a megcsalás vádja. A gép velünk szemben másképpen jön rá a hibájára. A bevitt adatoknak és a közöttük lévő összefüggéseknek a neurális hálóban van egy súlya, amelyet a gép állítgat. Ezt nevezzük tanulásnak. A megcsalós példára visszatérve ez gépi nyelven jelentheti azt, hogy a jelszó változtatást 5%-ra értékeli, vagyis ez még önmagában nem jelenti a hűtlenséget. Elegendő csak az informatikai szakértők tanácsára gondolni, miszerint legritkábban három havonta érdemes a fontosabb jelszavainkat lecserélni. Az sem bizonyító értékű, ha hajszál van a másik ruháján. A rúzsfolt sem, mivel ez származhat a házastárstól is. Ha azonban olyan rúzstól származik, amilyet a házastárs sosem használ, akkor a gépi agy is „vészjeleket küld". Nagyon leegyszerűsítve a példát a gép itt a rúzsfolt tényéből és színéből, esetleg formájából döntött, mert ezt értékelte a betanulás folyamán elegendőnek. Persze lehet, hogy a rúzsfolt az édesanyjától került az arcára a családi ebédre való megérkezést követő köszönéskor. A neurális hálók lényege, hogy nagyon sokféle bemeneti adatból tudnak dolgozni, akár milliónyi apró jelet figyelve.

A mesterséges intelligenciákról általánosan elmondható, hogy konkrét feladatra, egy meghatározott probléma megoldására íródnak. Az a gép, amely jól tud sakkozni és megveri a

világbajnokokat is, nem lesz jó a dámában, autós vagy lövöldözős játékban. Sőt, még francia sakkban sem lesz jó, hiszen pont ellentétesen kellene játszania, amely újratanulás nélkül nem megvalósítható. Az ember ezzel szemben néhány perc alatt képes váltani egyik játékról a másikra, vagy profi gamerek esetén egyik játékstílusról a másikra.

A tanulás történhet egy gépen, vagy gépek tucatjain egyszerre. A nagyvállalatok hatalmas számítási kapacitású gépparkot, úgynevezett klasztereket üzemeltetnek, amelyen az alkalmazottak tudják tanítani a leprogramozott mesterséges intelligenciát. A tanulásnak van egy új, egyre jobban terjedő módja, az elosztott tanulás, vagyis a federated learning. A lényege, hogy adatvédelmi és adatbiztonsági okból más készíti a modellt és máshol tanítják. Olyan ez, mint az iskola, ahol nem a szülő tanítja a gyermekét, hanem elviszi egy erre a célra kialakított intézménybe. Az elosztott tanulás lényege, hogy az érzékeny, üzleti titkot képező betanulás alapjául szolgáló adathoz „viszik" a mesterséges intelligenciát, így a modellt készítőknek nem kell odaadni az adatot, ők csak kész és betanult rendszerrel dolgoznak.

A mesterséges intelligenciákra tehát összességében jellemző, hogy bemeneti adatokból tanulnak, képesek ezek közül válogatni és a tanulás során finomítani a

következtetéseket azért, hogy minél jobban illeszkedjenek az életben felmerült problémák megoldásaihoz. Nem képesek ugyanakkor önállóan gondolkodni és komoly, összetett feladatot elvégezni.

Sokan már azt is mesterséges intelligenciának tartják, amikor egy „ha" és „akkor" utasítás van a programban. Ezek annyira alaputasítások, hogy még a legegyszerűbb programok is tucatnyi ilyen parancsot tartalmaznak. Van egy informatikus vicc, amelyben a feleség mondja az informatikus férjnek, hogy nincs otthon margarin és tojás, ezért hozzon margarint és ha van tojás, akkor hozzon hatot. A programozó erre hazaállít hat margarinnal és mondja a feleségnek, hogy volt tojás. A vicc azt példázza, hogy „ha" van tojás, „akkor" hozzon hatot. A programozó számára ez csak egy feltétel, amely alapján el kell döntenie, egy vagy hat margarint visz haza. Könnyen belátható, az ilyen egyszerű felépítésű rendszerek azért elég messze állnak a mesterséges intelligenciától. A tudományos eszmefuttatásokat ezért mellőzzük és a könyv további részében mesterséges intelligencia alatt olyan rendszereket értünk, amelyek legalább megközelítik az emberi gondolkodás lemásolásán alapuló egyszerűbb struktúrákat, legyen szó akár megerősítéses tanulásról, akár neurális hálókról és mélytanulásról.

Mesterséges intelligencia – típusok és a tanulás módszerei

A mesterséges intelligencia meghatározása, elhatárolása és típusainak elkülönítése komoly tudományos viták tárgyát képezi, sőt, még az sem írható le pontosan, mit tekintenek a kutatók gépi tanulásnak (machine learning) vagy mélytanulásnak (deep learning). Mi a rixelnél úgy gondoljuk, jelen könyv keretei közt, illetve a problémák megoldása során úgy általában nincs is értelme ilyen szintű tudományos vitákba bonyolódni, mivel e kérdések tisztázása a gyakorlatias problémamegoldáshoz egyáltalán nem szükséges, sőt. Ugyanakkor áttekintés szintjén mindenkinek, aki érdeklődik a téma iránt, érdemes tisztában lenni azzal, milyen jellemző típusai vannak a mesterséges intelligenciának, illetve a tanulás, tanítás folyamatának.

A mesterséges intelligencia fogalmának határmezsgyéjén mozognak az egyszerűbb tanuló algoritmusok, vagy gépi tanulásos megoldások, melyek kevés összefüggést képesek átlátni, ugyanakkor kis számítási kapacitást is igényelnek. Ilyen szoftverekkel napi szinten találkozhatunk például az automatizált ajánlókban (film, zene, reklám, stb.) illetve

optimalizálási megoldásokban. Klasszikusan mesterséges intelligenciának tekinti mindenki a neurális hálókat, melyek működése és felépítése igencsak változatos. Ezeket az algoritmusokat mélytanuló (deep learning) modelleknek is szokták nevezni. A fogalom nagyon sokféle működési mechanizmust takar, melyek tetszőleges formában kombinálhatók egymással, ha rendelkezésre áll a szükséges erőforrás. A neurális hálók szerkezeti elemeit rétegeknek nevezik. Ezek nagyon sokfélék és többségük különféle módon konfigurálható is. Minden réteg, illetve beállítás bír előnyökkel és hátrányokkal egyaránt. A rétegek tipológiája annyira szerteágazó, hogy ismertetésük még az áttekintés szintjén is jelentősen meghaladná e könyv terjedelmét, ráadásul időről-időre újabb és újabb típusokat alkotnak a fejlesztők, így a naprakész ismeretek könyvbe foglalása gyakorlatilag lehetetlen. A neurális mechanizmusok alkalmazása távolról sem tekinthető lezárt egésznek, még számos lehetőséget kínál minden kalandvágyó fejlesztő számára.

Egy mesterséges intelligencia alapú rendszert használhatunk a megadott adatok értékelésére vagy előrejelzések készítésére egyaránt. Az adatok értékelése többféle céllal történhet, így kiválaszthatunk fontos tulajdonságokat, például hogy

milyen körülmények segítik a jobb termésátlagot, vagy pontokat köthetünk össze, amelyek egy halmazba tartoznak, mint egy tárgy egy képen, ezen kívül adatok alapján eldönthetjük, hogy egy terméket milyen minőségi kategóriába sorolunk, stb. Az előrejelzés, vagy más néven predikció, a bemeneti adatokból történő jövőbeli következtetések levonását jelenti, az ilyen célt szolgáló algoritmusokkal lehet előre jelezni egy kisebb térség légszennyezettségének alakulását, a tőzsdei árfolyamokat vagy épp azt, mikor hibásodik meg egy alkatrész a gyártósoron, így műszakok kiesése nélkül, időben lehet elvégezni egy esetleges karbantartást.

A kimeneti adatok tekintetében két nagy csoportja van a mesterséges intelligenciának, az egyikbe az érték alapú, vagy más néven regressziós értékek tartoznak, a másik esetében pedig osztályozást kapunk eredményül. Az érték alapü mesterséges intelligenciára jó példa az, amikor a rendszer képes megmondani egy adott részvény jövőbeni árfolyamát, egy adott terület termésátlagát vagy például azt, egy adott banki tranzakció mennyire tekinthető tipikusnak az ügyfél számlatörténete alapján. Az osztályozó modellek esetében előre meg kell határozni, az eredményt hány csoportba szeretnénk sorolni, mert ennyi adatponttal fog rendelkezni az utolsó, kimeneti

réteg. A modell itt a bemenő adatok összességét sorolja egy csoportba és a megfelelő csoport valószínűségét állítja a legmagasabb értékre. Az ilyen tanulásnak külön érdekessége, hogy a legritkább esetben ad egy csoportra 100 %-os valószínűséget, de ez nem is feltétlenül probléma. Ha például kézzel írott számok felismerésére tanítjuk be a neurális hálót, 10 kimeneti pontra van szükségünk 0-tól 9-ig. A hibátlanul teljesítő betanult modell esetében is előfordulhat, hogy egy csúnyán írt 4-es számra, a 4-est 75%-os, az 1-est 18%-os míg a 7-est 7%-os valószínűséggel adja meg eredményként. Mi emberek is vagyunk ezzel így. Ilyenkor a modellhez kapcsolódó kód feladata az, hogy a döntést 4-esként értékelje, hiszen annak az értéke volt a legmagasabb. A hétköznapokban osztályozó algoritmus az írásfelismerő mellett például a levélszemét-felismerő vagy olyan modell, amely betegségeket ismer fel röntgenfelvétel vagy betegadatok alapján. Az osztályozó rendszerek különleges problémája még, hogy milyen irányban hibatűrők. A levélszemét-szűrő esetében például elfogadhatóbb, ha egy tömeg-üzenetet nem minősít kéretlennek, mint az, ha üzletileg fontos levelet címkéz nem kívánatosnak. Ugyanakkor az egészségügyi alkalmazásokban jobb, ha valakit egészségesként is betegnek címkéz, mint ha nem ismeri fel a valóban beteg pácienseket, mivel a

betegként azonosított páciens esetében további vizsgálatok el tudják dönteni, valóban is milyen mértékben beteg az illető, ugyanakkor egy tévesen egészségesként diagnosztizált ember valódi betegsége kezeletlen maradt. A gyakorlati betanítási folyamatban ilyen kérdésekre megfelelő válaszok adása a legfontosabb.

A fentebb taglalt esetekben a modell úgy tudott tanulni, hogy az elvárt végeredmény minden egyes különálló esetben ismert volt számára. Ezt felügyelt tanulásnak is nevezik. A mesterséges intelligencia azonban ennél többre is képes. A megerősítéses tanulás esetében valamilyen módszer alapján jutalmat határoz meg a fejlesztő, amely nincs vagy nem szükségszerűen van közvetlen kapcsolatban minden egyes önálló lépéssel. Ilyen lehet egy kártyajáték, illetve a játékok úgy általában, illetve ilyen feladat kitalálni egy útvesztőből is. A jutalom ilyenkor a folyamat végén érkezik és a jutalom nagyságától függően a modell erősíti saját magát az adott döntési irányban vagy épp új utakat keres saját beállításainak újrahangolásával. A megerősítéses tanulás két fő jellemzője, hogy rendszerint csak részben felügyelt, hiszen egy folyamat végén érkezik a megerősítés és általában lehetséges cselekvések közül választja ki a leginkább megfelelőt. Létezik teljes mértékben

felügyelet nélküli tanulás is, ezek alkalmazása inkább adattudományi jelentőséggel bír. Az ilyen szoftverek külön érdekessége, hogy pusztán az algoritmus felépítéséből és az adatokból következik a tanulási folyamat anélkül, hogy konkrét, betanító célértékek kerülnének meghatározásra.

Minden mesterséges intelligencia alkalmazásra igaz, hogy a végeredmény minősége nagyban függ az adatok előkészítésének minőségétől. Ez külön és komoly szakértelmet kíván, olyat, amely túlmutat a klasszikus adattudományi ismereteken és mesterséges intelligencia fejlesztői tapasztalatot is igényel. Az adatok kapcsán fontos tudni, hogy azokat előzetesen is érdemes feldolgozni. Ez alatt olyan módszereket kell érteni, amelyek az ember számára is nyilvánvaló eredményekkel járnak, illetve a modell számára megkönnyítik az adatok elemzését. Nyilvánvaló eredmény alatt érthető például az, hogy a hibásan felvett adatokat törölni kell és a vélhetően nem fontos adatok nem kerülnek feldolgozásra, például fogröntgen felvételek esetén egyáltalán nem fontos, hogy melyik kórházban készült. A modell számára nélkülözhetetlen átalakítás, hogy minden számszerű és egységes formában kerüljön feldolgozásra. Ez különösen nagy kihívás nyelvi adatok esetében. Az adatok kapcsán meg kell még említeni, hogy léteznek

olyan szoftverek is, amelyek idősoros adatokból dolgoznak. Ez azt jelenti, hogy a modell nem egyetlen időállapot, hanem időállapotok sorozata alapján hoz döntést. Ilyen esetekben az időköz általában azonos. Az idősoros neurális hálók egyébként általában belső memóriával is rendelkeznek. Rendszerint ilyen megoldások becsülnek tőzsdei árfolyamokat vagy készítenek időjárási előrejelzéseket.

Az eddig felsorolt mesterséges intelligenciák közös jellemzője, hogy egyikük sem tekinthető a szó klasszikus értelmében véve kreatívnak. Azonban ma már az alkotás képessége is programozható, erre két jó példa a stílus átvitel (style transfer) és a versengő generatív hálózat (GAN, cGAN). A stílus átvitel esetében arról van szó, hogy egy képre vagy hangra egy másik kép vagy hang stílusát húzzuk rá, így működnek azok a népszerű alkalmazások, ahol a saját magunkról készült vagy általunk feltöltött bármilyen képet tudjuk neves festők stílusára alakítani, de ugyanezen elv alapján működnek hamis hang vagy videó előállító alkalmazások is. A versengő generatív szoftverekben általában két modell van, az egyik azzal foglalkozik, hogy eldöntse, az általa megkapott adat valós vagy hamis, a másik pedig hamis adatokat generál. A két modell folyamatosan versenyben van egymással, ha a hamisító lebukik, fejleszti magát, hogy

olyan adatot állítson elő, amit a másik hálózat valódinak gondol, ha viszont a bíráló modell igazinak ismeri fel a hamisítványt, kénytelen fejlődni. Ilyen alapon működnek azok a megoldások, amelyek a semmiből generálnak emberi arcokat, tájképet, nem létező térképet vagy épp zenét.

A mesterséges intelligencia lehetőségeinek és alkalmazási körének kimerülése még a távoli jövőbe vész, de már ma is sok helyen találkozhatunk vele. Az élet számos apró kellemetlenségén segít át vagy épp vigyáz a biztonságunkra, egészségünkre. Nap mint nap újabb területeket hódít meg és alakítja életünk.

Erőforrások

A mesterséges intelligenciáról sokan azt gondolják, túl drága dologról van szó, amelyet csak a nagy cégek engedhetnek meg maguknak. Ennél nagyobbat nem is lehet tévedni, hiszen nagyon sokféle eszközön van lehetőség lefuttatni egy betanult rendszert. A mesterséges intelligencia ára több tényezőtől függ: Milyen problémát kívánunk megoldani? „Mennyire számításigényes az adott megoldás? Mennyi erőforrás-tartalék szükséges a rendszerbe a jövőbeli fejlesztési igényekre? Mennyire optimális a kód?

A fenti felsorolásból látszik, a fejlesztő csapat nagyon nagy hatással lehet a felhasznált pénzmennyiségre. Habár a több általában jobb, a gazdasági életben a tőke szűkös erőforrás, így a fejlesztők sem költhetnek felesleges kapacitásokra. A mesterséges intelligenciáról elmondható, egyszerűbb változatai nem kizárólag irodai számítógépes konfigurációkon képesek elfutni, hanem akár mikroszámítógépeken is. Ahhoz, hogy a szükséges erőforrásokat pontosan meghatározzuk, először a probléma elemzése szükséges. A probléma körülhatárolásához tartozik a megfelelő adathalmaz összeállítása, amely a tanulás alapját fogja képezni. Az alkalmazott modell is

kulcsfontosságú. Nem érdemes például túl sok energiát pazarolni olyan adatokra, amelyek a végeredmény tekintetében 1% alatti befolyással bírnak. Ha túl sok ilyen adatot használunk, könnyen abba a hibába eshetünk, hogy sokkal nagyobb memóriára lesz szükség, vagy a tanulási folyamat jelentősen lassabb lesz. Amennyiben olyan dolgot akarunk megtanítani a számítógépnek, amely hosszú távon is változatlan, nem szükséges jelentős erőforrástöbblet beépítése a rendszerbe. A kód optimalizálásán is rengeteg múlik. Ezért érdemes olyan fejlesztőkkel dolgozni, akik rendelkeznek a szükséges tudással és tapasztalattal. Sok pénz spórolható úgy is, ha az állandó fejlesztők mellé ideiglenesen külsős szakértő is csatlakozik az optimalizációs folyamat végéig.

A mesterséges intelligenciára épülő rendszerek a tanulás és felhasználás szempontjából két nagy csoportba tartoznak. Először a tanulás történik, majd utána a betanított modell alkalmazza a tudást. Összességében elmondható, a tanulás sokkal lassabb és erőforrás-igényesebb, mint a mindennapi működés. Kisebb cégek esetén nem érdemes a betanításhoz méretezni az erőforrást, mert utána csak üresjáratban fog működni. Ilyenkor hasznos a betanításhoz szervert bérelni, vagy olyan fejlesztőket megbízni, akiknek rendelkezésükre áll a betanításhoz szükséges

kapacitás. A különbség az eszközök árában akár háromszázszoros is lehet.

Egyszerűbb feladatokat, mint például munkahelyen objektumvédelem, vagy arcfelismerős beléptető-rendszer már egy kis mikroszámítógép is el tud látni. Arról nem is beszélve, hogy egy ilyen rendszer tökéletesen integrálható egy robotizált környezetbe is, amely képes vezérelni a sorompót, statisztikákat és jelentéseket készíteni, kapcsolni a fényeket, szervezni a munkarendet. Komolyabb, de még mindig közepes költségigényű rendszerek már telefonok fogadására is képesek, így tehermentesíteni tudják az ügyfelekkel vagy partnerekkel foglalkozó munkatársakat. Szenzorok alkalmazásával a költségek növekednek ugyan, de még intelligensebb és a környezetre jobban reagáló mesterséges intelligencia alapú rendszer alkotható meg. A legtöbb, céges környezetben előforduló probléma nem indokolja a sok millió forintos eszközök beszerzését.

A mesterséges intelligenciához felhasználható eszközök nagyon széles skálát ölelnek fel. Az egyszerű irodai gépekben lévő videókártyáktól kezdve a nagy számítási kapacitású szuperszámítógépekig. Az eszközök tekintetében jelentős eltérés van aszerint, hogy éppen a világ melyik pontján járunk. Mi azt tapasztaltuk, hogy Magyarországon inkább az alacsony

vagy középkategóriás videókártyákra épülnek a gépi agyak, míg nyugaton sokkal elterjedtebbek a felhő alapú megoldások. Mindegyiknek van előnye és hátránya is, egyik sem nevezhető abszolút jobbnak a másiknál. Komplex rendszerek is képesek olyan videókártyákon elfutni, amelyeket a legtöben otthonra játék céljából vásárolnak. Természetesen ahhoz, hogy egy autonóm jármű képes legyen közúti forgalomban közlekedni, nem elégséges egy olyan kártya, amivel otthon autóversenyes játékot játszunk. Ugyanígy nem megoldás a felhő alapú számítás sem, mivel ahhoz folyamatos internetkapcsolatra lenne szükség. Nem engedhető meg az, hogy az autó megálljon vagy balesetet szenvedjen azért, mert nem tud kiértékelni egy adott közlekedési helyzetet.

Sok cég alkalmaz saját felhőt, úgynevezett klasztert. Ezek remek házon belüli megoldások akár közepes méretű vállalkozások számára is. Természetesen a rendelkezésre álló pénztől függ, mekkora géppark alakítható ki. Amennyiben egy ügyvédi iroda a beérkező telefonokat szeretné kiszolgálni olyan hangalapú intelligens rendszerrel, amely képes az ügyfél igényeit felmérni és a megfelelő ügyvédhez irányítani, akkor kisebb szerver is elegendő.

Az erőforrás-szűkösségre egy lehetséges megoldás a később bővebben is kifejtésre kerülő federated

learning. Lehetséges, hogy egy-egy számításban résztvevő eszköz nagyon lassan tud csak számolni, így tanulni, de sok ilyen eszköz összehangolt módon erősebb és olcsóbb megoldás lehet, mint hatalmas kapacitású központi számítógépeket fenntartani.

A rixelnél azt tapasztaltuk, sok esetben a hatékony erőforrás-menedzsmenttel rengeteg pénz spórolható, vagy jóval nagyobb, okosabb rendszer építhető ki. Elegendő arra gondolni, egy irodában a számítógépek akkor üzemelnek, amikor az alkalmazottak dolgoznak. Hivatali időn kívül a kapacitás üresen áll. Jó munkaszervezéssel és a számítógépek megfelelő üzemeltetésével új szintre emelhető a hatékonyság.

Egyre tipikusabb az, hogy a mesterséges intelligencia alapú rendszereket nem önmagukban használják fel, hanem valamilyen robotikai megoldással egybekötve. Ez lehet egyszerű okosiroda projekt, amelyben a hőmérsékleti- és fényviszonyok szabályozása mesterséges intelligenciára van bízva, de lehet komplett gyártósor-vezérlés is, amelyben a szükséges eszközök rendelését is a mesterséges intelligencia végzi. A robotika ebben az értelemben a számítógépes elme kiterjesztett karja, amivel változást tud előidézni a világban: hőmérsékletet állítani, autót összeszerelni, telefonhívást indítani, gombot megnyomni.

Mesterséges intelligencia és programkód

A mesterséges intelligencia is csak egy program, mely utasításokból áll. Mint minden kódra, erre is igaz, csak arra képes, amire előzetesen a fejlesztő felkészítette. Éppen ezért például az öntudatra ébredő mesterséges intelligencia képe puszta vízió, hacsak nem kifejezetten erre ír valaki egy programot. Ugyanakkor a mesterséges intelligencia bizonyos értelemben több egy egyszerű programnál, hiszen a program készítője nem írja meg előzetesen a döntéshozatali mechanizmust, pusztán megteremti annak körülményeit, hogy a szoftver „felfedezze" a döntéshez szükséges összefüggéseket. A felfedezés jelen esetben matematikai műveletek millióinak ismételgetését jelenti. A döntések aszerint változnak, hogy az egyes műveletek milyen súllyal számítanak bele a végeredménybe.

A mesterséges intelligenciát tartalmazó algoritmusok három jól elkülönülő részre oszthatók. Az első fontos elem a programnak az a szakasza, amelyben a beérkező adatok olyan módon kerülnek átalakításra, hogy az a tényleges mesterséges intelligencia számára a legoptimálisabban legyen feldolgozható, illetve

feldolgozható legyen egyáltalán. A második egység maga a gépi tanulás, amely megkapja az első rész által előkészített adatokat, azokon matematikai műveletek millióit végzi el és végül létrehoz egy döntést, melyet adatok formájában továbbít a harmadik egységnek, amely a fejlesztő által meghatározott módon tényleges cselekvést hajt végre. Abban az esetben, ha a szoftver még tanulási fázisban van, a harmadik egység nem vagy nem feltétlenül hajt végre valamit, hanem visszajelez, hogy a döntés megfelelő volt-e vagy sem. A betanítási folyamat során ugyanis a középső programrész az ilyen visszajelzésekből változtatja saját döntéshozatali mechanizmusát. A mesterséges intelligencia típusától és felhasználásától függően a tanulás és tényleges használat el is válhat egymástól élesen, de lehet folyamatos is. Ez utóbbi esetben a program harmadik része akár egyszerre cselekedhet és vissza is jelezhet. illetve az első és a harmadik rész funkcionalitása egybe is olvadhat így végső soron egy keretbe foglalva a tényleges mesterséges intelligenciát.

A mesterséges intelligencia legjellemzőbb formája a neurális háló. Az ezt tartalmazó szoftveren belül a tényleges döntéshozatali folyamatot jelentő kódot, azaz a hálót, modellnek is szokták nevezni. Minden modell egy bemeneti rétegből, néhány „rejtett” rétegből és egy – nagyon ritkán esetleg több –

kimeneti rétegből áll, a rétegek elnevezése nem tükröz lényegi különbséget, csak a rétegek egymáshoz képesti sorrendjére utal. A „rejtett” szó pusztán szakmai kifejezés, a fejlesztő valójában pontosan ismeri a rejtett rétegek felépítését is, hiszen ő írta a kódot, ami kialakítja azokat. Ugyanakkor vélhetően az elnevezésnek és a folyamat bonyolultságának köszönhető, hogy a mesterséges intelligenciát fekete doboznak is szokták nevezni, ahol a csoda történik. A valóságban egyáltalán nincs fekete doboz és csoda sem történik, legfeljebb az ember csodálkozik rá, milyen egyszerű programok milyen gyorsan képesek bizonyos összefüggéseket átlátni, adott esetben akár olyanokat is, amelyeket az ember még hosszabb idő alatt is csak nehezen vagy egyáltalán nem.

A program megírásának pillanatában egyetlen mesterséges intelligencia sem képes jó döntést hozni. Ehhez be kell őket tanítani. A betanított hálózat két fontos ismérve a modell felépítése és a súlyok, amelyekkel a modell saját magát beállítja. A felépítést a programkód maga tartalmazza, a súlyok pedig számok sokasága, melyek ugyanúgy elmenthetők és betölthetők, mint bármely más adat, fájl. Ez lehetővé teszi, hogy ugyanazon modellt akár többféleképpen vagy több célra is be lehessen tanítani, illetve egy már betanított modell is tovább tanítható, ha új, más célra

van szükség, vagy épp valamilyen feltétel megváltozása miatt az eredeti modell már nem kellőképpen pontos. A tárolhatóság miatt az egyes verziók összehasonlíthatók és bármikor újra használhatók vagy felhasználhatók.

A valós használatban egy szoftveres megoldás sokszor nem csak egy modellt tartalmaz, illetve döntéshozatali folyamatot több program is táplálhat adattal és több program működése is függhet a végeredménytől. A problémától és rendelkezésre álló erőforrásoktól (idő és hardver) függően változik, változhat egy teljes megoldás bonyolultsága is, sőt, olykor erőforrás-szűke miatt nem feltétlen a mesterséges intelligencia a legjobb, de legalábbis nem az egyetlen megoldás. Néhány egyszerűbb és komplexebb példán keresztül valamennyi fent tárgyalt jellegzetesség bemutatásra kerül.

Egy a mobiltelefont feloldó arcfelismerő szoftver, noha számunkra jelentős biztonságot nyújt, fejlesztői szemmel nézve egy viszonylag egyszerű szoftver, hiszen egyetlen bemeneti forrása van, a készülék kamerája, egyetlen modellt alkalmaz, mely eldönti, a felhasználó látható-e a képen és egyetlen döntés függ a modelltől, feloldja-e a telefont vagy sem. Ráadásul magának a programnak saját működésére sem kell figyelnie, hiszen egy adott esemény, pl. egy gomb

megnyomása vagy a telefon megrázása elindítja a kódot, amely a képnek megfelelő cselekvést követően le is áll, visszaadja az irányítást a rendszernek. A felépítés szintjén ennél nem sokkal bonyolultabb egy arcfelismerésen alapuló beléptető rendszer, legfeljebb a modell sokkal nagyobb, hiszen mobiltelefon helyett akár egy kifejezetten ilyen célra épített számítógépet is használhatunk. A felhasználó vagy megrendelő számára ennél nem tűnik bonyolultabbnak, ha az a cél, hogy az embereket ne csak felismerje a gép és eldöntse, beengedi-e vagy sem, hanem kövesse is a személyek mozgását az épületben. A fejlesztő számára ez azonban már lényegesebb nagyobb kihívás, hiszen az előzőeknél sokkal összetettebb szoftveres megoldást igényel. Az arcfelismerő mellett ugyanis rögtön beléptetéskor egy olyan modellt is le kell futtatni, amely több szempont, például ruházat, hajszín, testalkat, mozgás alapján is képes egy személyt azonosítani. Miért van erre szükség? A válasz nagyon egyszerű. Az ember testének egy része vagy mozgása egy jól bekamerázott épületben szinte mindig látszik, ugyanakkor az arca egy tökéletesen bekamerázott épületben sem látható folyamatosan. Tehát rögtön legalább két modellel kell dolgoznunk. Ezen felül azonban szükség lehet még egy harmadikra is, amely például a hiányos kameraképek alapján felderíti, mi lehet a személy tényleges mozgási útvonala. Az

útvonalnak fejlesztői szemmel is van jelentősége, hiszen egy ekkora rendszer már akár végtelen mennyiségű erőforrást is fel tud emészteni, ha nincs optimalizálva. Ezért a jellemző útvonalak alapján vélelmezzük a személy mozgását és a konkrét személyre csak akkor fut le a modell, ha jó eséllyel akár rajta is lehet a képen. Így gondolkodva jelentős mennyiségű erőforrás takarítható meg.

Erőforrást nem csak a futtatások számának optimalizálásával, de akár a mesterséges intelligencia kihagyásával is el lehet érni. Tegyük fel, hogy építeni akarunk egy olyan eszközt, amely képes a fű lenyírására. A betanítás során észleljük, a rendszer teljesen másképpen viselkedik sötétben, mint napfényben. Rájövünk, ez két mesterséges intelligencia alkalmazását igényli. A hatékony eredmény érdekében szükség lesz egy olyan programra, amely egy fényérzékelő eszköz segítségével eldönti, melyik alrendszer kapja meg a feldolgozandó adatot. Természetesen arra is lenne lehetőség, hogy a gép a kamerakép alapján saját maga döntse el, milyen napszak van. Miért nem érdemes ezt a megoldást választani? Jelen esetben egy fényérzékelő lap filléres tétel a mesterséges intelligenciát futtató eszközökhöz képest, felépítésében jóval egyszerűbb, a javítása olcsóbb és könnyebb, valamint nem szükséges hozzá

semmilyen betanítás és fejlesztés, a szenzortól kapott adat rögtön feldolgozható. A számítási kapacitás véges, érdemes mindig a célnak leginkább megfelelő megoldást választani. Ha azonban még a szenzort is meg szeretnénk spórolni, letölthetünk egy adatbázist, amelyből meghatározzuk, az adott területen az év melyik napján mikor kel és mikor nyugszik a nap, így egy jól beállított órával is mindig a megfelelő rendszert tudjuk használni, az óra aktualizálása pedig jellemzően rendszerszinten történik, így ezzel még foglalkoznunk sem kell. Ugyanakkor az optimalizálási kísérletek túlzásba vitelének jelensége is létezik. Ha például csak a naptárra hagyatkozunk a fenti példában egy napfogyatkozás könnyen visszafordíthatatlan károkat okozhat kerti növényeinkben a nem megfelelő modell használata miatt. Éppen ezért a fejlesztőnek és a felhasználónak mindig közösen kell eldöntenie, hol milyen szűk keresztmetszetek adódhatnak, hiszen végső soron nem az a fontos, hogyan, hanem az, mennyire hatékonyan és milyen jól oldunk meg egy problémát.

Adatvédelem és adatbiztonság

A mesterséges intelligencia alapú rendszereknél el kell különíteni a tanulás alapjául szolgáló adatot és a gépi intelligenciát. Ezeket általában különböző csapatok készítik, hiszen a mesterséges intelligencia fejlesztőknek a modell készítése a dolgok, az adatokat az adatbirtokosok szolgáltatják. Vannak ingyenesen is elérhető nagyon jól összeállított ilyen adatcsomagok, legyen szó akár egészségügyön belül mondjuk a fogröntgenek elemzéséről, a tüdőgyulladás analízisen át egészen a melanóma azonosításig, mezőgazdaságon belül a különböző növénybetegségekről, vagy pénzügyi szektoron belül bankkártyacsalások kiszűrésére használt adatokról a tőzsdei árfolyamokig. Közös tulajdonságuk, hogy mindegyik hozzáférhető, nem tartalmaz üzleti titkot, hiszen akkor nem kerültek volna megosztásra az adatok.

A mesterséges intelligencia alkalmazásának olykor gátja az a helyzet, amikor az adat birtokosa nem akarja kiadni az adatot, a modell birtokosa pedig a modellt a kezéből. Erre a patthelyzetre megoldás az elosztott tanulás, vagyis a federated learning. Minden olyan helyen alkalmazható, ahol érzékeny adatok kerülnek felhasználásra a betanítási folyamat során, vagyis a két

kulcsszó az adatbiztonság és az adatvédelem. Ezek egyaránt lehetnek érzékeny egészségügyi adatok, vagy éppen üzleti titkok, egyedi ipari megoldások. Nem is az ok számít, ami miatt az adat birtokosa nem akarja megosztani a mesterséges intelligencia fejlesztővel az adatait, hanem az az állapot, amelybe így kormányozzák magukat. Az elosztott tanulás megoldást kínál erre a problémára. A mesterséges intelligencia fejlesztők titkosított módon és úton küldik át a modell adatait az adatbirtokosnak, vagy egy harmadik megbízható félnek, akiben megállapodnak. Ezt követően vagy az adatbirtokos, vagy harmadik fél igénybevétele esetén ez a megbízott közvetítő elvégzi a betanítást. A betanítás eredményét kapják meg a fejlesztők, így ők sosem fogják kezelni vagy látni az eredeti adatokat, csak az eredményeket.

Az előbbi példa az adatvédelemnek az egyik ágát érintette. Az adatokat ugyanakkor nem csak úgy lehet és kell megvédeni, hogy nem adjuk ki a kezünkből. Egy betanított modell alkalmas lehet arra, hogy kellő ismeret és tudás birtokában értékes információt nyerjünk ki belőle. Akár olyan információt is, amelyet a tulajdonosa legszívesebben elrejtene. Képzeljük el, van egy adathalmazunk, amely egy konkrét kórháztól kapott egészségügyi adatokat tartalmaz. A betegek nevei anonimizáltak, a nevek helyett csupán egy

számsor van, így látszólag nem lehetséges az beazonosítani egy konkrét személyt. A betegségek mellett ott van, hogy milyen étrendet követ az illető, mikor volt utoljára nyaralni és hol, hiszen ez bizonyos betegségek esetén fontos információ. Tegyük fel, sikerül megszereznünk egy vagyonnyilatkozati adatbázist is, szintén anonimizáltan. A két adatbázist összevetve még mindig anonim módon összeköthetővé válhatnak a személyek. Ha a vagyonnyilatkozatban vannak ritka, egyedi tárgyak, akkor az egész már névhez is kapcsolható, amennyiben például az adott térségben nem tipikus gépjármű a Bugatti Veyron. Ez csak egy példa volt azok közül, ahol az adatbázisokból nem kívánt információk szivároghatnak ki. Jogos igényként merül fel tehát az, hogy maga az adatbázis és a betanított mesterséges intelligencia modell is védett legyen az ilyen jellegű kockázatokkal szemben. Létezik megoldás arra, hogy egyedi eljárásokkal a betanított eredményekbe olyan jellegű anonimizálást biztosító zajt helyezzenek el, amely segítségével még az érzékeny adatok sem lesznek hozzákapcsolhatók a betanítás alapjául szolgáló adathalmaz valamely eleméhez. Ezzel az eljárással megvalósul az adatok megfelelő anonimizációja, ezáltal a védelme is. Mi a rixelnél adatvédelmi és adatbiztonsági konzultációk keretében átvizsgálva egy-egy partner cég adatbázisait és

modelljeit úgy tapasztaltuk, egyre nagyobb hangsúlyt kap az üzleti szférában az adatok és az alkalmazott szoftveres megoldások, modellek védelme.

Elosztott tanulási megoldásokat használnak például a mobiltelefonokban, de alkalmazható iparban, egészségügyben, banki rendszerekben is. Mobiltelefon esetében például felhasználói igény van arra, hogy a telefon gépelés közben a legjobb szavakat ajánlja fel, vagy a leginkább passzoló emodzsit próbálja a sor végére illeszteni. A felhasználók természetesen nem szeretnék, ha a nagy cégek elvinnék a telefonról az üzeneteket elemzés céljából, így inkább a mobilokon történik meg a modellek tanítása és csak az elkészült és betanított adatcsomag kerül át a cégekhez.

Mit tud egy gép

A könyv elején már említésre került, a gépi és az emberi elme annak ellenére más, hogy az előbbi a humán felépítést vagy tanulási folyamatot próbálja másolni. A gépi intelligenciák legfőbb problémája a tudás nehézkes átfordítása más élethelyzetekre. A gépnek újratanulásra van szüksége ahhoz, hogy az előbbihez akár nagyon hasonló problémát oldjon meg, ahogyan láttuk ezt a korábbi sakkos és francia sakkos példán. Számítógépes játékokban is alkalmaznak egyszerűbb mesterséges intelligenciákat az ellenfelek szimulálására. Évekkel ezelőtt még az algoritmust kiismerve nagyon egyszerűen le lehetett győzni a gépeket. Manapság vannak olyan játékok, amelyek korlátozottan képesek alkalmazkodni a játékoshoz, így nehezítve és ezáltal izgalmasabbá téve a játékot. Vannak kísérletek, ahol a gépeket valós idejű stratégiai (RTS) játékokra tanítják. Attól azonban, mert egy gép nagyon ügyesen játszik a Red Alert nevű játékkal, még a Starcraftban kikap az embertől, vagy akár a butább gépi ellenfeleitől is. Amennyiben valaki nagyon szereti az ilyen játékokat, egy újat is könnyedén megtanul, amennyiben a logikája hasonló. A gép nem látja a hasonlóságot, mert másképpen szerveződik a tudása.

A tanulás sebességében is vannak különbségek. Az emberek sokkal ügyesebbek abban, hogy egy tanulási körből vonjanak le következtetést. Ez talán abból fakad, hogy életünk során sokszor vannak olyan élethelyzetek, amikor egy dolgot kevés alkalommal tudunk a valóságban tesztelni. Ha dartsozni akarunk, nagyon sokat tudunk próbálkozni, hogy megtanuljuk a helyes dobási technikát. Az áram alatt lévő alkatrészt azonban nem akarjuk tízezerszer megfogni, hogy biztosan tudjuk, megráz bennünket. Jobb esetben a forró sütőhöz sem kell hozzáérnünk, de rosszabb esetben is maximum egy alkalommal megszerezzük a szükséges tudást. A gép nagyon rossz abban, hogy egy tanulási ciklusból sok információt kinyerjen. Ugyanakkor sokkal jobb abban, hogy egységnyi idő alatt lényegesen több ilyen tanulási ciklust hajtson végre. A dartsos példánál maradva, ha egy robotkarra szerelünk egy tűadagolót, így elérve a korlátlan dobás lehetőségét, akkor a gép non-stop képes dobálni, nem lesz szomjas, fáradt. A lendítési idő is rövidebb lesz, mert könnyebben beállítja a kívánt paramétereket. Van egy sokkal izgalmasabb példa a rixel projektjei közül. Dolgoztunk egy olyan mesterséges intelligencia alapú szövegfeldolgozó rendszeren, amelynek feladata jogi szövegek megértése volt. Túl azon, hogy a magyar nyelv gépi megértése nyelvészeti és nyelvtani okok miatt nehézkes, az algoritmus remekül vizsgázott a

jogszabályok megértésében. Kidolgoztunk egy olyan informatikai és nyelvészeti módszert, amellyel a jogi összefüggések könnyedén megérthetők a gép számára. Innentől a kiválasztott jogszabálytól függően órákra vagy akár percekre volt szükség a belső hierarchia feldolgozásához. Miért érdekes ez? Gondoljunk csak arra, a jogászok képzése során évekig tanítják a jogszabályok értelmezését. Vannak olyan jogágak, mint a polgári jog vagy a büntetőjog, amelyet több évig tanulnak a hallgatók. Vajon ők a vonatkozó törvénykönyv hány százalékát tudnák pontosan, megfelelően rendszerezetten és szöveghűen visszaadni? A magyar polgári törvénykönyv vagy a büntető törvénykönyv összesen jóval több, mint 150.000 szóból áll. Ha elfelejtjük, hogy a jogi szöveg megértése néha még a gyakorló jogászoknak is nehéz, és kissé túlbecsülve 200 szó/percben határozzuk meg az emberi olvasási sebességet, akkor nekünk 750 percig tartana elolvasni mindkét joganyagot, vagyis körülbelül fél napunk menne el csak az olvasásra. Az algoritmusunk ezzel szemben 15 perc alatt elolvassa és 12 órán belül megtanulja az összefüggéseket. Ugyanannyi idő alatt, amíg az ember egyszer elolvassa, a gép teljesen feldolgozza és megtanulja a joganyagot. Arról nem is beszélve, hogy a gép nem korlátozódik egy-két jogterületre, bármelyikből hasonló vagy akár gyorsabb sebességgel képes tanulni. Mindehhez az

algoritmusnak nem volt szüksége az 5 éves jogi képzés elvégzésére.

Másik fő különbség az, hogy az ember agya struktúrája viszonylag fix, normál esetben lassan változik, míg egy gépnek bármikor írhatunk újat. Nincs mesterséges intelligencia alapú betegség, amely a tanulást vagy a megtanult adatok kinyerését gátolná. Az emberek esetében a Parkinson-kór, Alzheimer-kór, Sclerosis Multiplex, Asperger-szindróma mind nehezítik a tanulást, az adatok értékelését vagy az előhívást. A mesterséges intelligenciák tanítása során léteznek olyan problémák, mint a korábban is említett túltanulás, vagy az, amikor a gép úgy gondolja, megtalált egy optimális megoldást, de az valójában csak egy látszat. Ezekre már kialakult módszertanok vannak, így egy többlépcsős betanítási fázisból álló rendszerben ezek maradéktalanul kiszűrhetők.

A felejtésben is eltér a gép az embertől. Mindannyiunkkal előfordult már, hogy egy vizsgán nem jutott eszünkbe a megfelelő fogalom, elfelejtettünk egy fontos évfordulót, vagy összetalálkoztunk valakivel, akinek nem tudtuk felidézni a nevét. A gépi felejtés két eltérő módon közelíthető meg. Az egyik olvasatban a gép sosem felejt, hiszen a megtanult adatok valamilyen feldolgozott formában mindig rendelkezésre állnak

mondjuk a neuronokhoz rendelt súlyozási számokban. Tehát a gép nem arra fog emlékezni, miből tanult, hanem arra, hogy mit tanult meg belőle. Olyan ez, mintha mi nem emlékeznénk a könyvre és a konkrét szövegre, amit olvastunk, csak az általunk leszűrt lényegi információkra. Ezt emberként nagyon nehéz elképzelni, mert ami emlékezésünk és tudásunk nem így épül fel. Sokszor pont ez jelenti a problémát, mert emlékszünk arra, hogy a megtanult információ hol található a könyvben, milyen képek vannak mellette, de magára a szóra egyáltalán nem. Másik megközelítésben a gép nagyon gyorsan felejt, hiszen a betanulás alapjául szolgáló adatra csak addig emlékszik, amíg az adott tanulási kör véget nem ér. Olyan ez, mint amikor ránézünk egy könyvre, majd elpillantva már nem is emlékszünk arra, mi volt a lapon. A gépi tanulásban ezt úgy oldották meg, hogy léteznek olyan megoldások, amelyek során a gép képes lesz hosszabb távon is emlékezni a betanulási adatra. Ezzel úgymond a gépi rövid távú memória mellé létrehozásra kerül a hosszú távú memória is. Itt nem arra kell gondolni, hogy évek vagy órák múlva fog emlékezni az adott információra, hanem arra, hogy néhány tanulási körrel később még tudja, mi volt az előző. Ezt a módszert használják fel például a fordítási feladatokban, amikor a megfelelő szó kiválasztásához ismerni kell a szövegkörnyezetet, vagyis a gépnek

szükséges emlékezni a megelőz szavakra, mondatokra, vagy akár bekezdésekre is. A képi feldolgozás során is alkalmazzák ezt a módszert, amikor először egy algoritmus megnézi a teljes képet és eldönti milyen környezetről van szó. A következő mesterséges intelligencia már ezt a tudást is megkapja, hogy kiválassza és kategorizálja a konkrét tárgyakat. Ha például egy koncertfelvételt mutatunk, akkor a színpadon lévő emberek azonosítását célszerű az előadókkal kezdeni, mert ők jóval szűkebb kör, az összes emberhez képest. Ugyanígy a reptéri fotón lévő repülőgép igazi gép, míg a gyerekszobában lévő repülő csak játék akkor is, ha pontos, kicsinyített másolata az eredetinek és csupán a gépet fotózva még egy ember sem lenne képes eldönteni, hogy igazi gépről vagy játékról van-e szó.

A tanult adatok előhívásában is vannak különbségek. A gépek esetén egy egyszerű programkód segítségével minden különösebb megterhelés nélkül előhívhatók az adatok. Emberként tudjuk, nem csak a felejtéssel kell megküzdeni, hanem a bennünk lévő több-kevesebb pszichés gáttal is. Megtanulhatjuk a világ legjobb nyitómondatát, ha az állásinterjún lámpaláz miatt dadogunk vagy hadarunk. Egy nekünk tetsző személy jelenlétében zavart érezhetünk, ami megnehezíti az összeszedett gondolkodást. Hiába értünk a

pizzasütéshez, az atomfizikához, vagy tudunk a fürdőszobában szépen énekelni, ha a külső tényezők miatt ezen képességek előhívásában probléma merül fel és a kinyert információ hibás, zajos lesz.

Az emberek többsége egyszerre csak egy dologra képes koncentrálni. Természetesen a testünk ösztönösen is sok feladatot elvégez, így nem kell külön energiát szánni a légzésre, vagy képesek leszünk vezetés közben is összefüggően beszélni az utastársunkkal, miközben a forgalmat is szemmel tartjuk. A gondolkodás azonban egy szűkös csatornarendszeren történik. A gépek ezzel szemben képesek párhuzamosan nagyon sok információt feldolgozni. Az Nvidia nevű cég egyik legfejlettebb, mesterséges intelligencia számításhoz használt eszközén, amely a Tesla V100 nevet viseli, 640 mag dolgozik. Ezek mindegyikére igaz, hogy önmagukban képesek egyszerre több műveletet elvégezni. Egy másik, a K80 jelzetű eszköz 4992 magot tartalmaz, igaz ezek gyengébb számítási képességekkel rendelkeznek a korábban említett társaikhoz képest. Míg nekünk embereknek egy agyunk van, addig egyetlen kártyán olyan, mintha egy kisebb városnyi ember gondolkodna közösen.

A mesterséges intelligencia és az emberek összehasonlításának célja nagyon sok esetben

szenzációhajhászás vagy hangulatkeltés. A gépek és az emberek eltérőek, más-más feladatok megoldásában vagyunk jók mi emberek, és másban a gépek. A gépek nagyon hatékonyak egy szöveg feldolgozásában és olyan információk gyors visszaadásában, amelyek megtalálhatók a szövegben. Ezzel szemben az ember elvont gondolatokra is képes választ adni, sőt a fantáziájának segítségével teljesen új környezetet is tud teremteni. Nézzünk egy példát! Julcsi elmegy a boltba kiflit venni. Út közben lát egy macskát. Julcsi odamegy a macskához és megsimogatja, amit a macska hangos nyávogással köszön meg. Egy természetes nyelvfeldolgozással foglalkozó gép képes válaszolni olyan kérdésekre, hogy ki ment a boltba, miért, mit látott, mit csinált a macska. A szövegben nincs információ arról, Julcsi miért akart kiflit venni. Emberként erre legalább egy, de általában több öltetünk is lenne: elfogyott, nem volt elég otthon, az otthoni megszikkadt és frisset akart. A gép nem tudja, hogy a főszereplő szereti-e a macskákat. Nekünk könnyű azt mondanunk, hogy igen, vagy legalábbis az biztos, hogy nem ellenséges velük, mert odament megsimogatni egyet. Emberként el tudunk azon is gondolkodni, vajon mi lett volna, akkor, ha kutyával találkozik, vagy éppen mit akar majd a kifliből csinálni. Ha szeretjük a gubát, akkor eszünkbe jutnak a családi emlékek, amikor közösen ettünk ilyen édességet. A

gépek erre jelenleg nem alkalmasak, nem képesek elvont gondolatokra. Egy mesterséges intelligencia alapú rendszer egy eszköz. Bármilyen fejlett vagy bonyolult is legyen, eszközről van szó. Minden gépi intelligencia mögött fejlesztők állnak, így végső soron mi teremtjük meg magunknak azokat a megoldásokat, amelyek könnyebbé teszik az életünket. Az élet egy-egy területén lehetnek a gépek sokkal jobbak és hatékonyabbak, azonban működésük specifikus. Nem szabad a mesterséges intelligencia alapú megoldásokat túlmisztifikálni és önálló entitásként kezelni őket. Attól, mert egy autonóm jármű képes elvezetni önmagát és képes közlekedési balesetet előre jelezni, csupán az előtte lévő autók mozgásának megfigyelésével, még nem lesz személyisége. Mi a rixelnél, mint fejlesztők, úgy gondoljuk, egyik legfőbb célunk olyan technológiai megoldások megvalósítása, amely az életet könnyebbé és egyszerűbbé teszik, hogy az emberek kiteljesedhessenek azokban a feladatokban, amelyekben ténylegesen jobbak. Egy rövid mondattal megfogalmazva: Robotoljanak inkább a robotok.

Tévképzetek, tévutak

A könyvben számos oldalról közelítettük meg a mesterséges intelligencia kérdését, igyekeztünk közérthető formában minél teljesebb képet adni bonyolult fogalmak kifejtése nélkül. Mivel a téma nagyon felkapott és sokan beszélnek mesterséges intelligenciáról anélkül is, hogy gyakorlati hozzáértésük bármilyen formában igazolt volna, úgy gondoljuk a valóban átfogó leíráshoz szükséges megemlíteni néhány olyan jellemző gondolatot, melyeket mi tévképzetnek, tévútnak tartunk. Reményeink szerint az olvasó a megszerzett ismeretek birtokában maga is képes lesz eldönteni, mi a mesterséges intelligencia és mi biztosan nem az. Milyen kockázatokat és lehetőségeket rejt magában és miben nem kell reménykedni, illetve mitől nem kell tartani?

A mesterséges intelligencia egy szoftver, melyben a tényleges intelligencia egy olyan modell, amit konkrét értékekkel beállított paraméterek alapján hoz létre a számítógép. E modell minden egyes adatpontjában pontosan ismert matematikai műveletek mennek végbe és e funkciók paraméterei is megismerhetők, kinyerhetők. A neurális háló két szélső rétegét

kimeneti és bemeneti rétegnek, a közötte elhelyezkedő valamennyi réteget rejtett rétegnek hívják. A rejtett szó itt egy technikai elnevezés, a fent leírt megismerhetőség ezekre a rétegekre is igaz. Ezek alapján véleményünk szerint a mesterséges intelligencia semmiképp sem nevezhető fekete doboznak, végképp nem olyan fekete doboznak, amiben nem tudjuk, mi történik, illetve amiben csoda történik. A modellekben lezajló matematikai műveletek mindegyike visszafejthető, csak épp azok jelentős számossága miatt nem tesszük ezt meg folyamatosan. Bizonyos esetekben az ilyen visszafejtést is el kell végezni, vagyis fel kell nyitni a fekete dobozt.

A mesterséges intelligencia az emberhez képest sokkal gyorsabban el tud sajátítani bizonyos ismereteket, ugyanakkor egy modell tudása csak az adott ismeretre korlátozódik. Bár elméletben akár létezhetne is általános, a világ minden gondját és problémáját egészében átlátó és mindenre megoldást kínáló mesterséges intelligencia, a gyakorlatban ez egyetlen modell keretében soha sem fog megvalósulni, hiszen a bemeneti és kimeneti adatpontok számossága egyaránt végtelen kell legyen. Épp ezért tartjuk hiteltelennek a mindenre megoldást kínáló rendszereket, még az egyszerre sok eltérő problémára megoldást kínáló rendszerek esetében is valószínű,

elképesztő mennyiségű erőforrást emészt fel a használatuk nem is beszélve a betanítás idő- és adatigényéről. A neurális hálók előtanítása hasznos és erőforrás-kímélő lehet, de a modellt mindig célfeladatra kell kialakítani. Egy adott szoftver akkor tud több, hasonló célt egyszerre kiszolgálni, ha a benne lévő modellt minden egyes célra külön-külön betanították és az így keletkezett súlyokat céltól függően tölti fel a keretszoftver.

A mesterséges intelligencia és a hozzá kapcsolódó tudományterületek folyamatosan változnak, fejlődnek. A legritkább esetekben lehet helytálló olyan kijelentés, amely nagyon markáns állítást fogalmaz meg, esetleg élesen bírálja a többi, másik megközelítést. Ezek az állítások egyaránt igazak arra, amikor kutatók vagy fejlesztők beszélnek a mesterséges intelligencia csoportosításáról, felépítéséről, jogi szabályozásának, vagy ipari szabványosításának kérdéséről. Folyamatosan születnek olyan új megoldások, amelyek jelentős hatást fejtenek ki a gépi intelligenciára. Új modellek és új számítási módok képesek teljesen megváltoztatni egy addigi rendszer hatékonyságát vagy éppen hatékonytalanságát. Általános igazságok megfogalmazására ezért mindig csak az adott fejlettségi szinten van lehetőség, amely könnyen eredményezheti azt, hogy a kijelentés akár napok

múlva elavulttá válik. A kérdés az, mennyire van értelme ezeknek a kijelentéseknek az ilyen gyorsan változó környezetben. Természetesen el kell határolni a tudományos vizsgálatot a mindennapoktól és a gyakorlattól, hiszen mindkettőre egyformán szükség van. Érdemes egyfajta optimista kétkedéssel fogadni minden szenzációhajhászó hírt, minden olyan kijelentést, amely arról szól, most megalkotásra került a bölcsek köve. A mesterséges intelligencia alapja a tanulás. Ez nem csak a gépekre igaz, a fejlesztőkre is. Mi a rixelnél minden nap kísérletezünk és tanulunk, hogy újabb és újabb megoldásokkal álljunk elő. Éppen ezért nem hisszük azt, hogy léteznek mindent-tudó emberek. A mesterséges intelligencia a kíváncsi emberek terepe, akik együtt képesek fejlődni a modellekkel.

Egyre többet hallani azt a gondolatot, hogy a mesterséges intelligencia a válasz minden egyes felmerülő problémára. Korábban említésre került az a jelenség, miszerint ha valakinek csak egy kalapácsa van, minden problémát szögnek néz majd. Szerintünk akkor lesz egy fejlesztő hiteles, ha képes valódi és hatékony megoldást adni a felmerülő problémára. Ezek egy része valóban mesterséges intelligencia alkalmazását igényli, míg mások egyszerűbb struktúrákon is tökéletesen futnak. Előfordulhat az is,

amikor még egyszerűbb „ha” és „akkor” utasításokat tartalmazó programok jelentik a tökéletes megoldást. Korábban már említésre került, hogy ez utóbbiakat nem tartjuk valódi mesterséges intelligenciának. A jó fejlesztő többféle szerszámmal rendelkezik és mindegyiket tudja is használni. Egy adott megoldásra sok módon el lehet jutni. A kérdés az, mekkora idő, pénz és energia árán. Az elmélet és a gyakorlat itt válik el egymástól.

Záró gondolatok

Reméljük, tetszett a könyv és sikerült néhány kérdőjel helyére pontot tenni a fejedben. Amennyiben úgy érzed, még mindig maradtak benned kérdések, ajánljuk figyelmedbe többi kiadványunkat és a honlapunkat. Személyesen is elmondhatod véleményed, amikor találkozunk egy konferencián vagy megkeresel minket valamelyik kapcsolattartási csatornánkon. Szívesen vesszük véleményed, észrevételed, kritikád.

Dr. Ország-Krisz Axel

https://www.facebook.com/dr.orszag.krisz.axel

Dr. Vécsey Richárd Ádám

https://www.facebook.com/drvecseyrichardadam

https://www.hyperrixel.hu

https://hyperrixel.github.io

rixel