

A gépi tanulás céljai

Jelenleg az AI a gépi tanuláson alapul, a gépi tanulás pedig alapvetően különbözik a statisztikától. A gépi tanulás valóban statisztikai alapokon nyugszik, de a statisztikától eltérő feltételezésekkel él, mivel a céljai is mások. Az 1.1. táblázatban felsorolunk néhány szempontot, amelyet figyelembe kell venni, amikor összevetjük az AI-t és a gépi tanulást a statisztikával.

Technika	Gépi tanulás	Statisztika
Adatkezelés	Nagy mennyiségű adatot (big data) használ, hálózatok és gráfok formájában; az érzékelőkből vagy webes szövegből származó nyers adatokat tanító és tesztadatokra bontja.	A modelleket arra használják, hogy kis mintákon biztosítsanak előrejelző képességet.
Adatbevitel	Az adatokat mintavételezik, véletlenszerűvé teszik és átalakítják, hogy maximalizálják a mintán kívüli (vagy teljesen új) példák előrejelzésének pontossági pontszámát.	A paraméterek valós jelenségeket értelmeznek, a nagyságrendre helyezve a hangsúlyt.
Eredmény	A valószínűséget figyelembe véve vetik össze, hogy mi lehet a legjobb tipp vagy döntés.	A kimenet a paraméterek változékonyságát és bizonytalanságát ragadja meg.
Feltételezések	Az elemző tanul az adatokból.	Az elemző feltételez egy bizonyos kimenetet, és megpróbálja bizonyítani azt.
Eloszlás	Az eloszlás ismeretlen, vagy figyelmen kívül hagyják az adatokból való tanulás előtt.	Az elemző egy jól meghatározott eloszlást feltételez.
Illesztés	Az elemző a lehető legjobban illeszkedő, de általánosítható modellt hoz létre.	Az eredményt a jelenlegi adateloszláshoz illesztik.

1.1. táblázat.
A gépi tanulás és a statisztika összehasonlítása

A gépi tanulás határai a hardver alapján

A hatalmas adathalmazok hatalmas mennyiségű memóriát igényelnek. Sajnos ezzel még nincs vége a követelményeknek. Ha hatalmas mennyiségű adat és memória áll rendelkezésre, akkor többmagos és nagy sebességű processzorokra is szükség van. A szakemberek többek között annak a problémáját igyekeznek megoldani, hogy hogyan lehetne a meglévő hardvereket hatékonyabban hasznosítani. Bizonyos esetekben egyszerűen nem lehet napokat várni arra, hogy megkapjuk egy gépi tanulási probléma eredményét. Az elemzőknek, akik tudni akarják a vá-

laszt, gyorsan van rá szükségük, még akkor is, ha az eredmény nem a legmegfelelőbb. Ezt szem előtt tartva a jobb hardverbe való befektetés mellett a jobb elemzésbe is be kell fektetni. Ebben a könyvben a következő kérdések közül foglalkozunk néhányval, hogy jobb élményben lehessen részed a gépi tanulással:

- » **Használható eredmény elérése:** Ahogy végighaladsz a könyvön, felismerheted, hogy először használható eredményt kell kapnod, hogy aztán azt tudd finomítani. Emellett néha túlzásba viszik az algoritmus hangolását, és az eredmény nagyon érzékenyvé (és akár egy adott adathalmazon kívül használhatatlanná) válik.
- » **A helyes kérdés feltevése:** Sokan csalódnak, amikor megpróbálnak választ kapni a gépi tanulásból, mert úgy finomhangolják egyre jobban az algoritmusukat, hogy nem változtatnak a kérdésen. A hardver hatékony használatához néha vissza kell lépni egyet, és felül kell vizsgálni a kérdést, amit feltettél. Előfordulhat, hogy rossz a kérdés, ami azt jelenti, hogy még a legjobb hardver sem fogja megtalálni a választ.
- » **Túlzott támaszkodás az intuícóra:** Kezdetben minden gépi tanulási kérdés egy hipotézis. A szakemberek az intuíciójuk alapján választanak kiindulópontot ahhoz, hogy megkeressék a választ a kérdésre. A kudarc gyakoribb, mint a siker, amikor gépi tanulással kapcsolatos tapasztalatokat szerzel. Az intuíciód visz művészetet a gépi tanulási gyakorlatba, de néha a megérzések tévesek, és felül kell vizsgálnod a feltételezéseidet.



ELMÉLET

Amikor elkezded felismerni a gépi tanulás környezetének fontosságát, azt is el fogod kezdeni átlátni, hogy a kívánt eredmény eléréséhez megfelelő hardverre van szükség, méghozzá a megfelelő egyensúlyt kialakítva. A jelenlegi legmodernebb rendszerek valójában GPU-kra (grafikus processzorokra) támaszkodva végeznek gépi tanulási feladatokat. Az, hogy GPU-kra támaszkodnak, valóban jelentősen felgyorsítja a gépi tanulási folyamatot.¹⁷

Emelkedjünk felül az AI-vágyálmokon

Mint sok más technológiánál, a mesterséges intelligencia és a gépi tanulás felhasználásai között is megvannak a maguk ábrándjai vagy divathóbortjai. Néhányan például arra használják a gépi tanulást, hogy fotókból Picasso-jellegű műalkotásokat készítsenek, olyan termékekkel, mint például a NightCafé¹⁸, amely azokat támogatja, akik valóban élvezik ezt a művészeti formát.¹⁹ Természetesen az ilyen felhasználási módok sok problémával járnak. Először is, a legtöbben nem igazán vágnának egy így készült Picassóra, legfeljebb a divathóbort jelképeként (mert még senki nem csinált ilyet). A művészet lényege nem az, hogy létrehozzuk

egy adott valóságos ábrázolás érdekes értelmezését, hanem hogy lássuk, hogyan értelmezte azt a művész. Az imént ajánlott cikk végén rámutatnak, hogy a számítógép a jelenlegi szintjén csak egy meglévő stílust tud másolni – teljesen új, saját stílust nem tud kialakítani. A következő részekben különböző AI- és gépi tanulási vágyálmokkal foglalkozunk.

Az AI és a gépi tanulás felhasználásainak divathóbortjai

Az AI olyan innovációs korszakba lép, amelyről korábban csak a sci-fi regényekben olvashattál. Nehéz lehet eldönteni, hogy az AI egy adott felhasználási módja valós-e, vagy csupán egy elszánt szakember dédelgetett álma. A *The Six Million Dollar Man*²⁰ (A hatmillió dolláros férfi) például egy televíziós sorozat, amely egykor fantáziadúsnak tűnt. Amikor bemutatták, valójában senki sem gondolta, hogy egyszer a valóságban is megjelenik a bionika. Hugh Herr²¹ és mások²² azonban másképp gondolkodnak – most már valóban lehetséges a bionikus lábak és karok használata. Persze még nem mindenki számára elérhető, a technológia csak most válik használhatóvá. Tovább bonyolítja a helyzetet a *The Six Billion Dollar Man* (A hatmilliárd dolláros férfi) című film, amely részben a *The Six Million Dollar Man* sorozaton alapul²³, és különböző okok miatt késik a megjelenése²⁴. Tény, hogy az AI és a gépi tanulás is lehetőséget nyújt majd néhány elképesztő technológia megalkotására, és hogy már most eljutottunk ezen technológiák megalkotásáig, de még mindig nagy fenntartásokkal kell kezelni, amit hallasz.

A gépi tanulás egyik legérdekesebb felhasználási módja a szórakoztatóiparban a *B* című filmben²⁵, szerepel, melynek főszereplője egy Erica nevű android. Erica feltalálói, Hiroshi Ishiguro és Kohei Ogawa sok időt töltöttek azzal, hogy élethűvé tegyék őt azáltal, hogy megpróbálták megvalósítani a szándék és a vágy emberi tulajdonságait.²⁶ Az eredmény valami olyasmi lett, ami új módon lép be a „borzongások völgyébe” (eredeti nevén „uncanny valley”²⁷)²⁸. A film cselekménye ugyanabba a sorba fog illeszkedni, mint az *Ex Machina* című filmé.²⁹



FONTOS

Ahhoz, hogy az AI és a gépi tanulás jövőbeli felhasználási módjai illeszkedjenek a sci-fikben az évek során megjelenített elképzeléseknek, a valós világban dolgozó programozóknak, adatelemzőknek és más szereplőknek eszközöket kell készíteniük. Semmi sem történik varázslattal, még ha varázslatnak is tűnik, ha nem tudod, mi történik a színpad mögött. Annak érdekében, hogy az AI és a gépi tanulás használati módjai között megjelenő divathóbortokból valós felhasználási módok váljanak, a fejlesztőknek, az adatelemzőknek és másoknak továbbra is olyan valós eszközöket kell készíteniük, melyeket most még talán nehéz elképzelni.

Ismerjük meg az algoritmusok szerepét

A gépi tanulásban minden az algoritmusok körül forog. Az *algoritmus* egy probléma megoldására használt eljárás vagy képlet. A probléma szakterületétől függ, hogy milyen algoritmusra van szükség, de az alapfeltevés mindig ugyanaz – meg kell oldani valamilyen problémát, például vezetni kell egy autót vagy dominót kell játszani. Az első esetben a problémák összetettek és sokrétűek, de a végső probléma az, hogy el kell juttatni egy utast egyik helyről a másikra anélkül, hogy az autó balesetet szenvedne. Hasonlóképpen, a dominójáték célja a győzelem. A következő szakaszokban az algoritmusokról olvashatsz részletesebben.

Mit csinálnak az algoritmusok?

Az algoritmus egyfajta tároló. Biztosít egy dobozt, amelyben tárolhatjuk a módszert egy adott típusú probléma megoldására. Az algoritmusok jól meghatározott állapotok sorozatán keresztül dolgozzák fel az adatokat. Az állapotoknak nem kell determinisztikusnak lenniük, de az állapotok ettől függetlenül meg vannak határozva. A cél egy olyan kimenet létrehozása, amely megoldás egy problémára. Bizonyos esetekben az algoritmus olyan bemeneteket kap, amelyek segítenek a kimenet meghatározásában, de a hangsúly mindig a kimeneten van.

Az algoritmusokban az állapotok közötti átmeneteket egy jól meghatározott és formális nyelven kell kifejezni, amely a számítógép számára érthető. Az adatok feldolgozása és a probléma megoldása során az algoritmus meghatároz, finomít és végrehajt egy függvényt. A függvény mindig az algoritmus által kezelt problémához illeszkedik.

Tekintsük át az öt fő technikát

Ahogy az előző szakaszban leírtuk, mind az öt törzsnek más-más technikája és stratégiája van a problémák megoldására, amelyek egyedi algoritmusokat eredményeznek. Ezen algoritmusok kombinálásával kelleme végül eljutnunk ahhoz a mesteralgoritmushoz, amely képes bármely adott problémát megoldani. A következő szakaszokban ezt az öt fő algoritmikus technikát tekintjük át.

Szimbolikus következtetés

A *fordított következtetés* kifejezést általában indukcióként használják. A szimbolikus következtetésben a dedukció az emberi tudás körét bővíti, míg az indukció az emberi tudás szintjét emeli. Az indukció általában új kutatási területeket nyit meg, míg a dedukció ezeket a területeket tárja fel. A legfontosabb szempont azonban az, hogy az indukció a tudományos, míg a dedukció a mérnöki rész az ilyen típusú következtetésben.

A két stratégia szoros együttműködésével oldhatók meg problémák úgy, hogy először megnyitnak egy lehetséges kutatási területet a probléma megoldása érdekében, majd feltárják ezt a területet annak megálapításához, hogy valóban megoldja-e a problémát.

Lássunk egy példát erre a stratégiára: a dedukciót használva azt mondhatnánk, hogy ha egy fa zöld, és a zöld fák élnek, akkor a fának élőnek kell lennie. Indukciót használva azt mondhatnánk, hogy a fa zöld, és a fa élő is, tehát a zöld fák élnek. Az indukció arra ad választ, hogy milyen tudás hiányzik egy ismert bemenet és kimenet mellett.

Az agy neuronjairól modellezett kapcsolatok

A konnekcionista talán a leghíresebbek az öt törzs közül. Ez a törzs arra törekszik, hogy neuronok helyett szilíciummal reprodukálja az agyi funkciókat. Lényegében minden egyes neuron (amelyet a való világbeli megfelelőjét modellező algoritmusként hoztak létre) a probléma egy kis darabját oldja meg, és sok neuron párhuzamos használatával oldják meg a teljes problémát.

A *visszaterjesztés*, vagyis a hibák visszafelé történő terjesztésének alkalmazásával azokat a feltételeket próbálják meghatározni, amelyek mellett a hibák eltávolításra kerülnek az emberi neuronokhoz hasonló módon felépített hálózatokból, a hálózatban használt *súlyok* (vagyis hogy mennyire számít bele egy adott bemenet az eredménybe) és *torzítások* (vagyis hogy mely jellemzők lesznek kiválasztva) megváltoztatásával. A cél a súlyok és a torzítások folyamatos módosítása mindaddig, amíg a tényleges kimenet meg nem egyezik a célkimenettel. Ezen a ponton a mesterséges neuron kibocsát egy megoldást, és továbbadja azt a soron következő neuronnak. Az egyetlen neuron által létrehozott megoldás csupán a teljes megoldásnak egy része. Minden neuron továbbítja az információt a sorban következő neuronnak, amíg a neuronok csoportja el nem készíti a végső kimenetet.

A változást tesztelő evolúciós algoritmusok

Az evolucionisták az evolúció elveire támaszkodnak a problémák megoldása során. Más szóval, ez a stratégia a legerősebbek túlélésére épül (eltávolítva minden olyan megoldást, amely nem felel meg a kívánt kimenetnek). Egy alkalmassági függvény határozza meg az egyes függvények használhatóságát a probléma megoldására.

A megoldási módszer egy fastruktúra segítségével keresi a legjobb megoldást a függvények kimenete alapján. Mindig az adott evolúciós szint győztese kap lehetőséget arra, hogy felépítse a következő szint függvényeit. Az elképzelés az, hogy a következő szinten közelebb kerülünk a probléma megoldásához, bár lehet, hogy nem oldja meg teljesen, ami azt jelenti, hogy újabb szintre van szükség. Ez a különleges törzs nagymértékben támaszkodik a rekurzióra és a rekurziót erősen támogató

nyelvekre a problémák megoldásához. Ennek a stratégiának érdekes eredményei a fejlődő algoritmusok: az algoritmusok egyik generációja tulajdonképpen felépíti a következő generációt.

Bayesi következtetés

A bayesiánusok különböző statisztikai módszereket használnak a problémák megoldására. Mivel a statisztikai módszerek egynél több, látszólag helyes megoldást adhatnak, a függvény kiválasztásakor azt kell meghatározni, hogy melyik függvénynek van a legnagyobb valószínűsége a sikerre. Amikor ezeket a technikákat alkalmazod, bemenetként elfogadhatsz például egy tünetegyüttest, és kimenetként eldöntheted, hogy a tünetek alapján milyen valószínűséggel áll fenn egy adott betegség. Tekintettel arra, hogy több betegségnek is lehetnek ugyanazok a tünetei, a valószínűség azért fontos, mert a felhasználó látni fog olyan eseteket, amelyekben valójában az alacsonyabb valószínűségű kimenet a helyes az adott körülmények között.

Végső soron ez a törzs azt az elképzelést támogatja, hogy soha ne bízz meg teljesen semmilyen hipotézisben (egy eredményben, amelyet valakitől kaptál) anélkül, hogy láttad volna a felállításához használt bizonyítékokat (a bemenetet, amelyet a hipotézis felállításához használtak). A bizonyítékok elemzése igazolja vagy cáfolja az általuk alátámasztott hipotézist. Következésképpen, nem lehet megállapítani, hogy milyen betegségben szenved valaki, amíg nem vizsgáltad meg az összes tünetet. Ennek a törzsnek az egyik legjellegzetesebb eredménye a spamszűrő.

Analógia alapján tanuló rendszerek

Az analogizáló kernelgépeket használnak az adatokban lévő mintázatok felismerésére. Ha felismered a mintázatot a bemenetek egy készletében, és összeveted egy ismert kimenet mintázatával, azzal megoldást adhatsz egy problémára. A cél, hogy a hasonlóság alapján határozd meg a legjobb megoldást egy problémára. Ez az a fajta következtetés, amely megállapítja, hogy egy adott megoldás adott körülmények között, egy korábbi időpontban bevált; így ennek a megoldásnak hasonló körülmények között ismét működni kell. Ennek a törzsnek az egyik legjellegzetesebb eredményei az ajánlórendszerek. Amikor megnyitod például az Amazont vagy az eMAG-ot, és megveszel egy terméket, az ajánlórendszer felajánl más, kapcsolódó termékeket, amelyeket lehet, hogy szintén szeretnél megvenni.

Mit jelent a tanítás?

Sokan már többé-kevésbé megszokták, hogy az alkalmazások egy függvényrel indulnak, bemenetként adatokat fogadnak, majd adnak egy

eredményt. Egy programozó létrehozhat például egy `Összeadás()` nevű függvényt, amely két értéket fogad bemenetként, például az 1 és a 2 értéket. Az `Összeadás()` eredménye 3. A folyamat kimenete egy érték. Régebben egy program megírásához meg kellett érteni az adatok kezeléséhez használt függvényt, amely bizonyos bemenetekkel egy adott eredményt ad vissza.

A gépi tanulás megfordítja ezt a folyamatot. Ebben az esetben tudjuk, hogy vannak bemenetek, például 1 és 2. Azt is tudjuk, hogy a kívánt eredmény 3. Azt azonban nem tudjuk, milyen függvényt alkalmazzunk ahhoz, hogy a kívánt eredményt kapjuk. A tanítás mindenféle példát ad a tanuló algoritmusnak a kívánt bemenetekre és az ezekből a bemenetekből várt eredményekre. A tanuló algoritmus ezután ezen bemenetek alapján hoz létre egy függvényt. Más szóval, a tanítás az a folyamat, amely során a tanuló algoritmus egy rugalmas függvényt rendel hozzá az adatokhoz. A kimenet jellemzően egy adott osztály valószínűsége vagy egy numerikus érték.



FONTOS

Egyetlen tanuló algoritmus sokféle dolgot képes megtanulni, de nem minden algoritmus alkalmas bizonyos feladatokra. Egyes algoritmusok elég általánosak ahhoz, hogy tudjanak sakkozni, arcokat felismerni a Facebookon és diagnosztizálni a rákot betegeknél. Egy algoritmus minden esetben egy függvényre redukálja az adatbemeneteket és azok várható eredményeit, de a függvény az adott feladattípushoz kapcsolódik, amelyet az algoritmusnak végre kell hajtania.

A gépi tanulás titka az általánosítás. A cél az, hogy általánosítsuk a kimeneti függvényt, hogy az a tanító halmazon túli adatokon is működjön. Gondolj például egy spamszűrőre. A szótárad 100 000 szót tartalmaz (ez igazából egy kicsi szótár). Egy korlátozott, 4000 vagy 5000 szókombinációból álló tanító adathalmazból olyan általánosított függvényt kell tudni létrehozni, amely képes megtalálni a spameket abban a $2^{100\,000}$ kombinációban, amely a függvény számára látható lesz, amikor tényleges adatokhoz használod.

Ebből a szemszögből a betanítás lehetetlennek, a tanulás pedig még annál is rosszabbnak tűnhet. A tanuló algoritmus azonban mindössze három összetevőre támaszkodik ennek az általánosított függvénynek a létrehozásához:

- » **Ábrázolás:** A tanuló algoritmus egy *modell*t hoz létre, azaz egy olyan függvényt, amely adott bemenetekre adott eredményt fog adni. Az ábrázolás modellek egy halmaza, amelyeket a tanuló algoritmus meg tud tanulni. Más szóval a tanuló algoritmusnak olyan modellt kell létrehoznia, amely a kívánt eredményt fogja adni a bemeneti adatok alapján. Ha a tanuló algoritmus nem képes elvégezni ezt a feladatot, akkor nem tud tanulni az adatokból, és az adatok kívül esnek a tanuló algoritmus hipotézissterén. Az ábrázolás része annak feltárása, hogy mely *jellemzőket* (az adatforrás mely adatait) kell felhasználni a tanulási folyamathoz.

- » **Értékelés:** A tanuló algoritmus több modellt is létrehozhat. Nem látja viszont a különbséget a jó és a rossz modellek között. Egy értékelési függvény határozza meg, hogy a modellek közül melyik adja vissza a legjobban a kívánt eredményt bemenetek egy halmaza alapján. Az értékelési függvény pontozza a modelleket, mivel egy-nél több modell is visszaadhatja a kívánt eredményt.
- » **Optimalizálás:** A tanítási folyamat egyszer eljut arra a pontra, hogy előállítja modellek egy készletét, amelyek a bemenetek egy adott halmazára általában a megfelelő eredményt tudják adni. Ezen a ponton a tanítási folyamat áttekinti ezeket a modelleket, és meghatározza, hogy melyik modell működik a legjobban. Ezután a legjobb modellt adja meg a tanítási folyamat eredményeként.

A könyv nagy részében az ábrázolásra összpontosítunk. A 10. fejezetben például arról olvashatsz, hogy hogyan készíthetsz egy működő levélszemét- (spam-) felismerőt a naiv Bayes-algoritmus segítségével a probléma valószínűségi ábrázolása alapján. A tanítási folyamat azonban többről szól, mint egyszerűen az ábrázolás kiválasztásáról. Mindhárom lépés szerepet játszik a tanítási folyamatban. Szerencsére kezdetben összpontosíthatunk az ábrázolásra, és hagyhatjuk, hogy a könyvben tárgyalt különböző könyvtárak végezzék el helyettünk a munka többi részét.