

Szoftver mély neuronhálók alkalmazásához

1. előadás

Kovács Bálint, Varga Viktor
ELTE IK Mesterséges Intelligencia Tanszék

Elérhetőségek

Tárgyfelelős:

Dr. habil. Lőrincz András

Oktatók:

Varga Viktor - vv@inf.elte.hu

Kovács Bálint - vale@inf.elte.hu

E-learning felület

Canvas

<https://canvas.elte.hu/>

- **Kommunikáció:** Fórum, privát üzenet (Canvas, Teams)
- Előadás diáorok, videó linkek, Colab notebookok
- Házi feladat kiírás, beadás, kvízek
- Értékelés

Követelmények

Jegyszerzéshez szükséges:

- 3 házi feladat sikeres megírása
- Szóbeli vizsga

Opcionális:

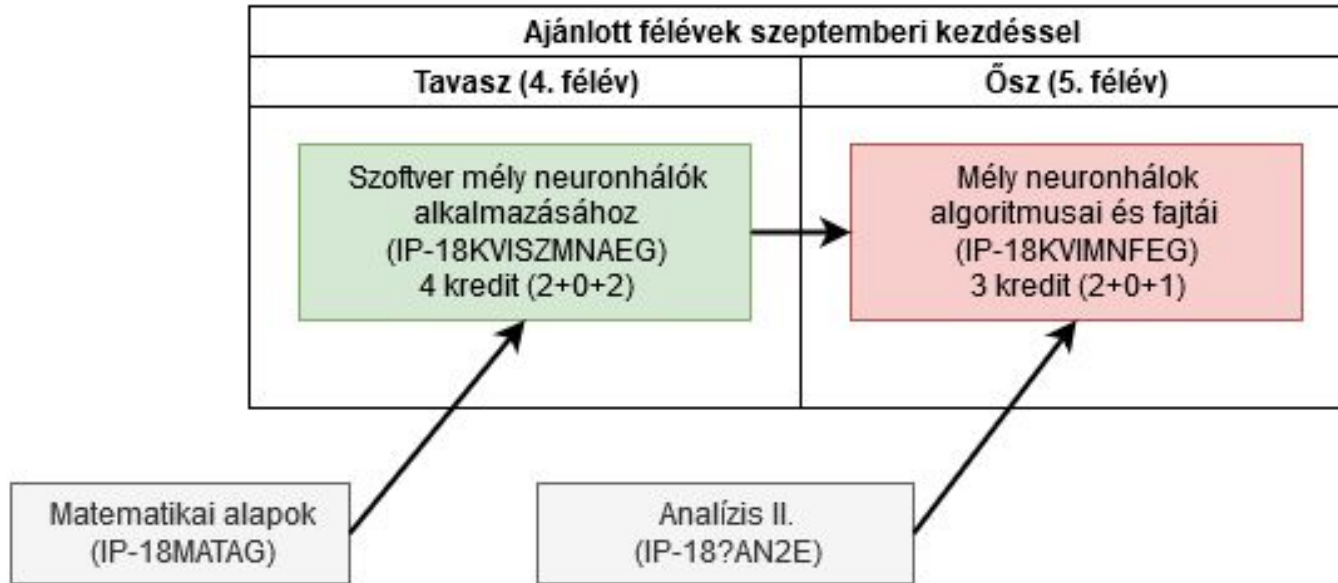
- Canvas kvízek
- 3 szorgalmi házi feladat

Követelmények

A kvízek, szorgalmi házi feladatok megírása és a kötelező házik időben történő beadása javítja a vizsga eredményét:

- Amennyiben a vizsga eredménye legalább elégséges
- 10 pont ér egy teljes jegynyi javítást
- Házi feladatok beadása az első határidőig: 3 x 2 pont
- Szorgalmi házi feladatok: 15-20 pont
- Kvízek: kb. 8 x 1 pont

BSc - MI tanszék, gépi tanulás tárgyak



Mi a gépi tanulás lényege?

Úgy oldunk meg problémákat, hogy nem a megoldás algoritmusát programozzuk le.

Ehelyett, egy **paraméteres modellhez keresünk** olyan **paramétereket** melyekkel a modell jól oldja meg a feladatot.

→ A feladat konkrét megoldásának ismerete nélkül tanuljuk meg megoldani a feladatot.

Milyen feladatokra alkalmazható jól a gépi tanulás?

- Ha közelítően optimális megoldás is megfelelő.
- Ha nem ismert konkrét algoritmus, ami megoldja a feladatot.
- Ha nem gazdaságos egyedi algoritmust fejleszteni a problémára.

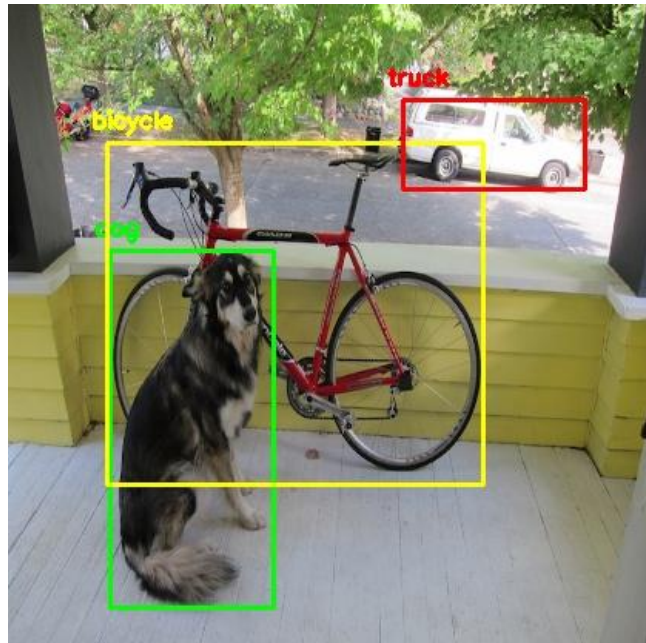


Milyen feladatokra alkalmazható jól a gépi tanulás?

- Ha a feladatot csak mintapéldák segítségével tudjuk formálisan definiálni.

Például: Különböző objektumok pozíciójának detektálása fényképeken.

Hogyan definiáljuk azt, hogy “kutya”, vagy “kerékpár” mintapéldák nélkül?



Milyen feladatokra nem ideális a gépi tanulás?

- Ha a probléma hatékony megoldása ismert és könnyen algoritmizálható.
- Ha a közelítő eredmény nem megfelelő.

Példák:

- Veszteségmentes tömörítés
- Rendezés (sorting)
- A legtöbb, pontosan definiált szabályrendszer szerint működő szoftver

```
def get_base_server_url(self):  
    if config.JENKINS_API in self.baseurl:  
        return self.baseurl[:-(len(config.JENKINS_API))]  
    else:  
        return self.baseurl  
  
def validate_fingerprint(self, id):  
    obj_fingerprint = Fingerprint(self.baseurl, id, jenkins_ob  
    obj_fingerprint.validate()  
    log.info("Jenkins says %s is valid" % id)  
  
def reload(self):  
    """Try and reload the
```

Gépi tanulás

Módszerek három főbb csoportja:

- **Felügyelt tanulás** (supervised learning)
- **Felügyeletlen tanulás** (unsupervised learning)
- **Megerősítő tanulás** (reinforcement learning)

Felügyelt tanulás

Adott: A tanítóminta (training set), input-címke párok halmaza

$$\{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$$

$$x \in X \subset \mathbb{R}^n, y \in Y \subset \mathbb{R}^k$$

Feladat: A címke (az elvárt output) minél jobb becslése az inputból.

Azaz, keresünk olyan h_θ függvényt (hipotézisfüggvényt), melyre:

$$h_\theta(x) = \hat{y} \approx y$$

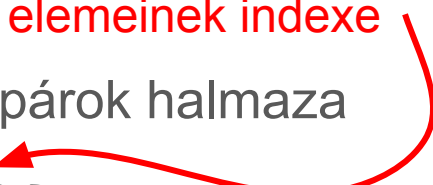
Felügyelt tanulás

Adott: A tanítóminta (training set), input-címke párok halmaza

$$\{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$$

$$x \in X \subset \mathbb{R}^n, y \in Y \subset \mathbb{R}^k$$

a zárójeles felső index nem
hatvány, hanem a minta
elemeinek indexe




Feladat: A címke (az elvárt output) minél jobb becslése az inputból.

Azaz, keresünk olyan h_θ függvényt (hipotézisfüggvényt), melyre:

$$h_\theta(x) = \hat{y} \approx y$$

a hipotézisfüggvény x inputból becsli az y
címket. A becslését “ y kalap”-pal jelöljük.

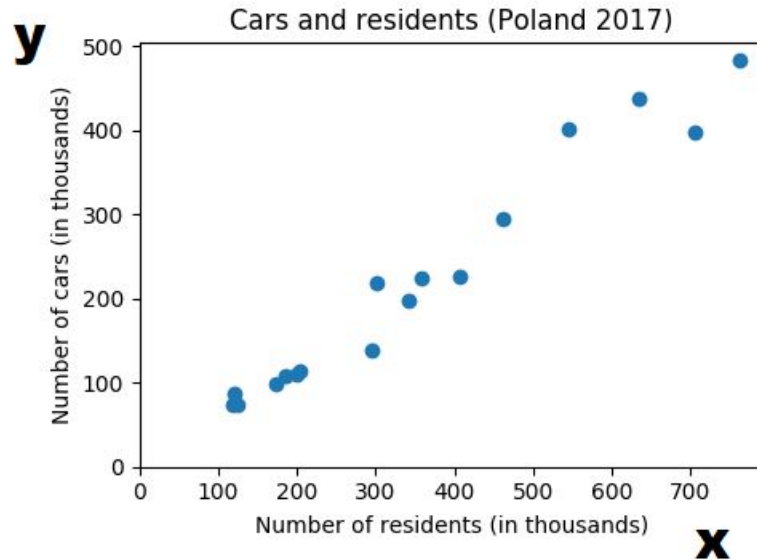


Felügyelt tanulás

Példa: Becsüljük meg az autók számát egy adott városban, ha ismerjük a város lakosságának számát.

x: egy adott város lakosságának száma

y: egy adott városban megtalálható autók száma



$$x^{(j)}, y^{(j)} \in \mathbb{R}$$

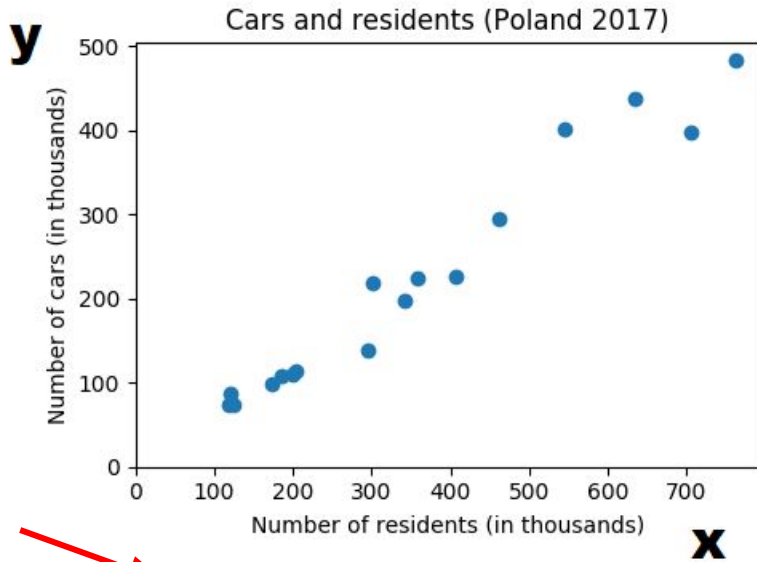
Felügyelt tanulás

Példa: Becsüljük meg az autók számát egy adott városban, ha ismerjük a város lakosságának számát.

x: egy adott város lakosságának száma

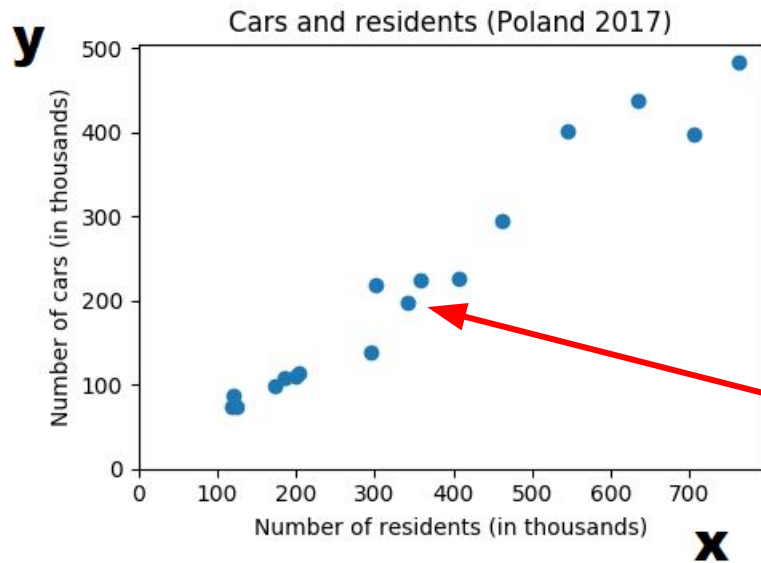
y: egy adott városban megtalálható autók száma

Egy input változó és egy címke: a mintaelemek a címkével, kétdimenziós vektortérben (síkon) elhelyezkedő pontok



$$x^{(j)}, y^{(j)} \in \mathbb{R}$$

A minta ábrázolása



$$x^{(j)}, y^{(j)} \in \mathbb{R}$$

	feature	címke
	x (egy adott város lakossága, ×1000)	y (autók száma a városban, ×1000)
Varsó (j = 1)	1760	910
Krakkó (j = 2)	770	465
Lublin (j = 3)	340	198
...

Felügyelt tanulás

Milyen hipotézisfüggvényt keressünk?

Az autók száma közel arányos a város lakosságának számával...

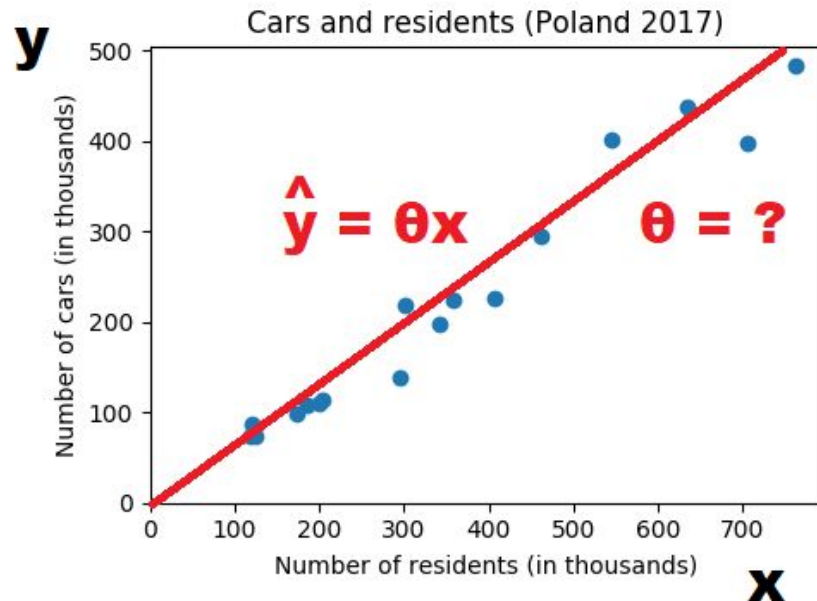
Felügyelt tanulás

Milyen hipotézisfüggvényt keressünk?

Nagyon egyszerű (lineáris)
hipotézisfüggvény:

$$y \approx \hat{y} = h_{\theta}(x) = \theta x$$

θ a hipotézisfüggvény paramétere,
ebben az esetben az egyenes
meredeksége lesz...



Felügyelt tanulás

Milyen hipotézisfüggvényt keressünk?

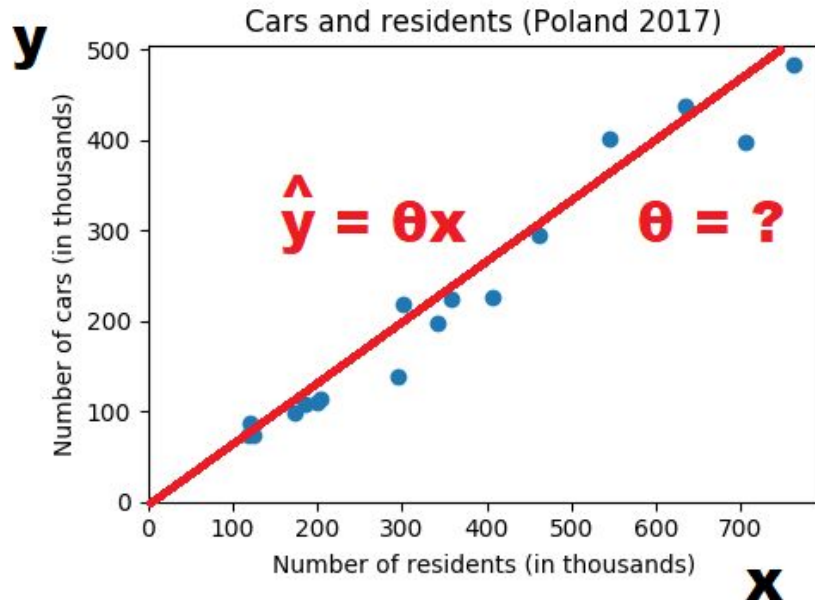
Nagyon egyszerű (lineáris)
hipotézisfüggvény:

$$y \approx \hat{y} = h_{\theta}(x) = \theta x$$

θ a hipotézisfüggvény paramétere,
ebben az esetben az egyenes
meredeksége lesz...

Keresünk egy olyan θ paramétert,
mellyel $h(x)$ jól közelíti az igazi y
címkéket!

Például a $h(x) = 0.65 \cdot x$ hipotézis-
függvény erre a konkrét mintára jól
illeszkedik, így tehát egy jó
paraméter a $\theta = 0.65$.



Felügyelt tanulás

A mintaelemek nem feltétlenül egy input változóból és egy címkéből állnak...

Felügyelt tanulás

Példa: Becsüljük meg a képeken látható emberek életkorát!

x: egy portré
(színes, rögzített felbontással)

y: a fényképen található személy életkora



$$x^{(j)} \in [0, 255]^{196608}$$

$$y^{(j)} \in \mathbb{R}$$

Hány dimenziós lenne a mintát ábrázoló grafikon ebben az esetben?

Felügyelt tanulás

Példa: Becsüljük meg a képeken látható emberek életkorát!

x: egy portré
(színes, rögzített felbontással)

y: a fényképen található személy életkora



Minden pixel minden színcsatornájának fényereje egy-egy input változó:
a mintaelemek 196608 (+1 címke) dimenziós vektortérben elhelyezkedő pontok.

$x^{(j)} \in [0, 255]^{196608}$
 $y^{(j)} \in \mathbb{R}$

A minta ábrázolása



$$x^{(j)} \in [0, 255]^{196608}$$

$$y^{(j)} \in \mathbb{R}$$

feature#1 feature#2 feature#196608 címke

	x_1	x_2	...	x_{196608}	y
	(1. pixel "red" fényerő)	(1. pixel "green" fényerő)	...	(65536. pixel "blue" fényerő)	(életkor)
1. kép (j = 1)	255	17	...	175	29
2. kép (j = 2)	125	0	...	0	9
3. kép (j = 3)	36	240	...	215	35
...

Felügyelt tanulás

Milyen hipotézisfüggvényt keressünk?

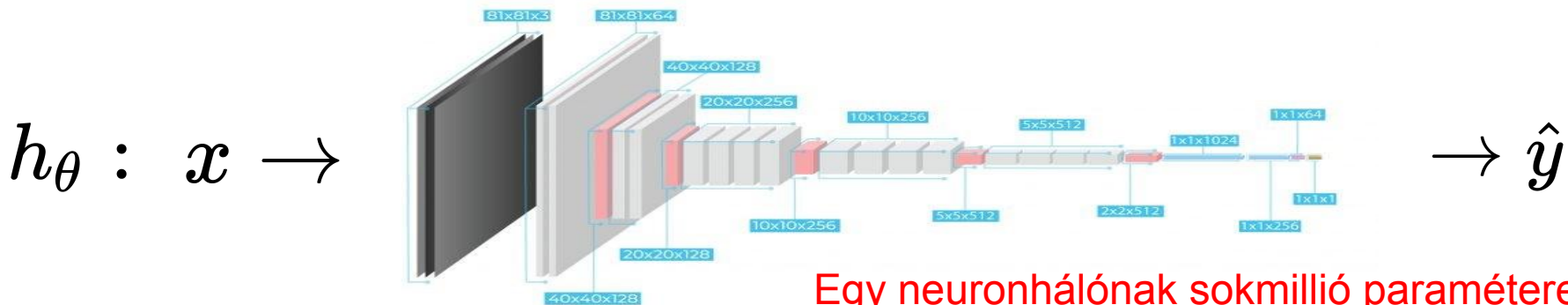
A lineáris hipotézisfüggvény nem biztos, hogy a legjobb választás...

Felügyelt tanulás

Milyen hipotézisfüggvényt keressünk?

A lineáris hipotézisfüggvény nem biztos, hogy a legjobb választás...

Helyette, használjunk egy igazán bonyolult függvényt, egy konvolúciós mély neuronhálót! **A neuronhálók is csak függvények...**



Egy neuronhálónak sokmillió paramétere lehet, így θ itt egy sokmillió elemű vektor

A felügyelt tanulás két fő feladata

Regresszió: A címkehalmaz összefüggő és végtelen

$$|Y| = \infty$$

Példa: Autók számának, vagy életkor becslése

Klasszifikáció: A címkehalmaz diszkrét és számossága véges

$$|Y| < \infty$$

Példa: Mintaelemek kategorizálása

- A lakosság számából eldönteni, hogy város-e, vagy falu egy adott település
- Mi a foglalkozása a képeken szereplő személyeknek?

Felügyelt tanulás

Hogyan állapítjuk meg mennyire jó a becslés?

$$h_{\theta}(x) = \hat{y} \overset{?}{\approx} y$$

Felügyelt tanulás

Hogyan állapítjuk meg mennyire jó a becslés?

J költségfüggvény segítségével. $J : \theta \rightarrow \mathbb{R}_{\geq 0}$

A költségfüggvény megadja, hogy mennyire tér el a valódi címke és a becslésünk adott paraméter értékek esetén.

Egy egyszerű példa költségre: $J(\theta) = | \underbrace{h_{\theta}(x)}_{\hat{y}} - y |$

Felügyelt tanulás

Hogyan állapítjuk meg mennyire jó a becslés?

J költségfüggvény segítségével.

A költség mindenképpen egy nemnegatív szám (skalár) kell, hogy legyen. Ha az y címke több változóból áll, az abszolútérték helyett pl. norma fog kelleni...

A költségfüggvény megadja, hogy mennyire tér el a valódi címke és a becslésünk adott paraméter értékek esetén.

Egy egyszerű példa költségre: $J(\theta) = |h_{\theta}(x) - y|$

Minél nagyobb az adott θ paraméterrel a becslésünk hibája, annál nagyobb a költsége θ -nak. Tökéletes becslés esetén a költség 0.

$\underbrace{h_{\theta}(x)}_{\hat{y}}$

Felügyelt tanulás

Felügyelt tanulás feladata:

Keressük azt/azokat a θ^* paramétereket, melyekkel a $h_{\theta}(x)$ hipotézisfüggvényünk jól közelíti a valódi címkét (y), azaz a J költségünk minimális.

$$\theta^* = \arg \min_{\theta} J(\theta)$$

Megkeressük (tanuljuk) az optimális paramétereket a tanítóhalmazon, majd új, címkézetlen adathoz is tudunk becsülni címkét a betanított modellel.

Felügyelt tanulás - alkalmazások

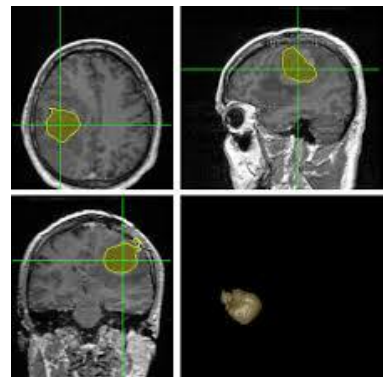
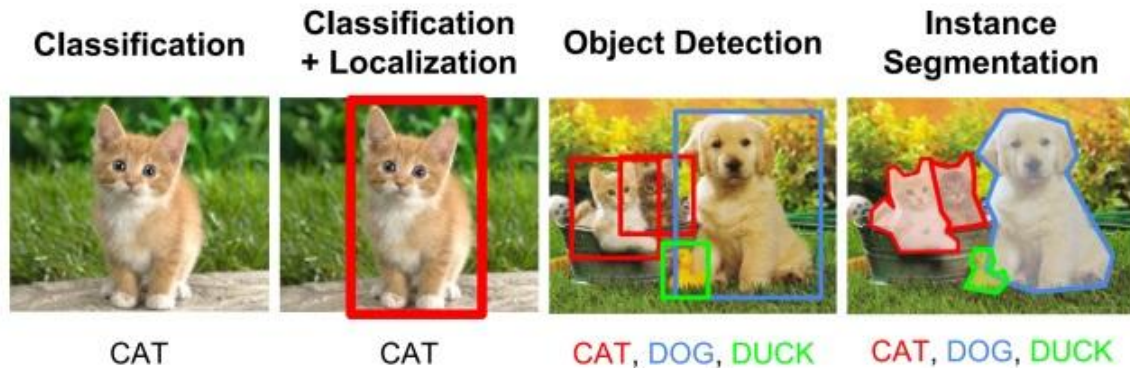
Összetettebb feladatokra is alkalmas a neuronhálókkal megvalósított felügyelt tanulás...

Felügyelt tanulás - alkalmazások

Objektumdetektálás, képszegmentálás

x: Egy fénykép

y: A képen található objektumok köré írt téglalap (bounding box) koordinátái, az objektum kategóriája, vagy pixel szintű kategóriatérkép

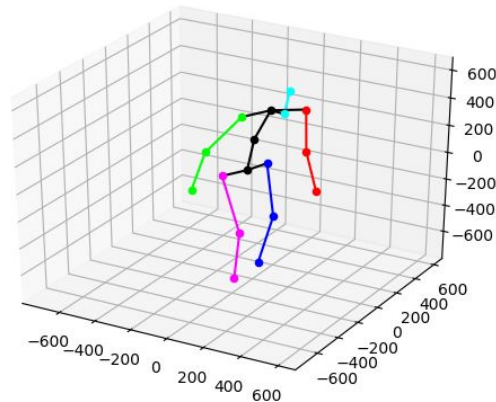


Felügyelt tanulás - alkalmazások

2D / 3D humán pózbecslés fényképről

x: Egy fénykép

y: A képen található személyek ízületeinek 2D vagy 3D koordinátái

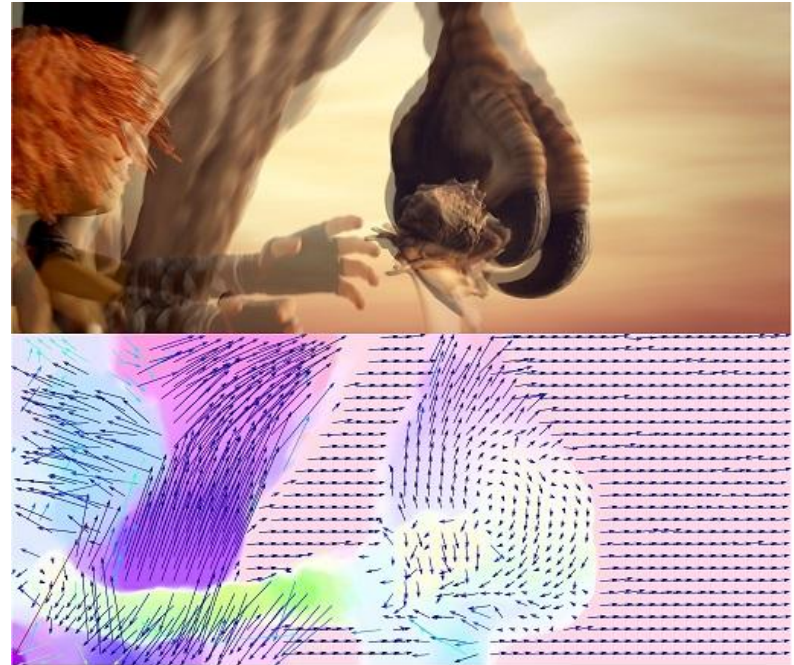


Felügyelt tanulás - alkalmazások

Optical flow becslés - két képkocka pixeleinek megfeleltetése egymással

x: Egy videó két egymást követő képkockája

y: Az első képkocka minden pixelére egy-egy elmozdulásvektor, amely a második képkocka valamelyik pixelére mutat



Felügyelt tanulás - alkalmazások

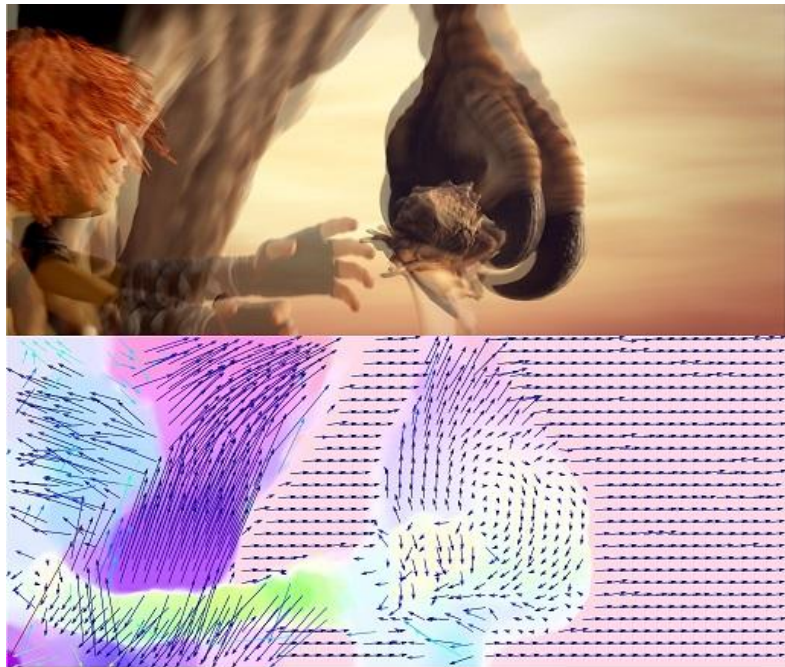
Nyilvánvaló, hogy nem minden feladat megoldása egyértelmű...

Optical flow becslés - két képkocka pixeleinek megfeleltetése egymással

x: Egy videó két egymást követő képkockája

y: Az első képkocka minden pixelére egy-egy elmozdulásvektor, amely a második képkocka valamelyik pixelére mutat

Optical flow felügyeletlen módon is tanulható.



Felügyelt tanulás - alkalmazások

Természeti nyelv-feldolgozás (NLP), érzelmi analízis (text sentiment analysis)

x: Karakterek / szavak sorozata

y: Érzelmi címke



Donald J. Trump
@realDonaldTrump

Follow

As I have stated strongly before, and just to reiterate, if Turkey does anything that I, in my **great** and unmatched **wisdom**, consider to be off limits, I will totally **destroy** and **obliterate** the Economy of Turkey (I've done before!). They must, with Europe and others, watch over...

8:38 AM - 7 Oct 2019



?



Felügyelt tanulás - alkalmazások

Természetesnyelv-feldolgozás (NLP), képek feliratozása (image captioning)

x: Egy kép

y: A kép tartalmát leíró karakterek / szavak sorozata



"man in black shirt is playing guitar."



"construction worker in orange safety vest is working on road."



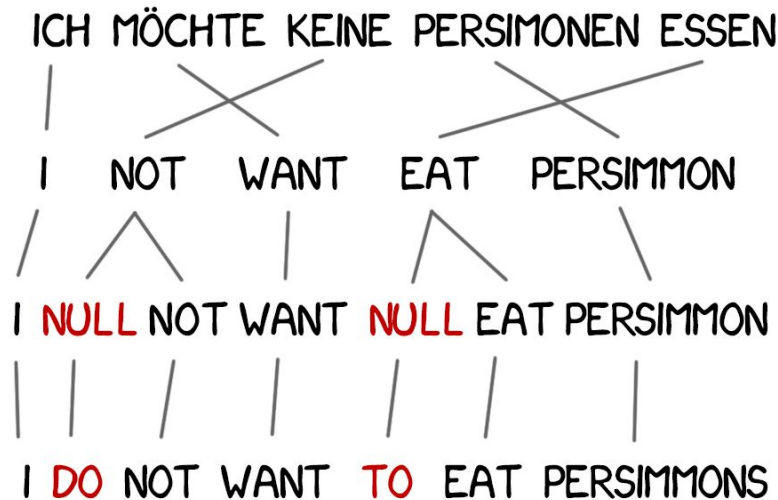
"two young girls are playing with lego toy."

Felügyelt tanulás - alkalmazások

Természeti nyelv-feldolgozás (NLP), gépi fordítás (machine translation)

x: Karakterek / szavak sorozata a forrásnyelven

y: Karakterek / szavak sorozata a célnyelven



Felügyelt tanulás - alkalmazások

A természetesnyelv-feldolgozás (NLP) problémája



"man in black shirt is playing guitar."



"construction worker in orange safety vest is working on road."



"two young girls are playing with lego toy."

ICH MÖCHTE KEINE PERSIMONEN ESSEN

I NOT WANT EAT PERSIMMON

I NULL NOT WANT NULL EAT PERSIMMON

I DO NOT WANT TO EAT PERSIMMONS

Felügyelt tanulás - alkalmazások

A természetesnyelv-feldolgozás (NLP) problémája



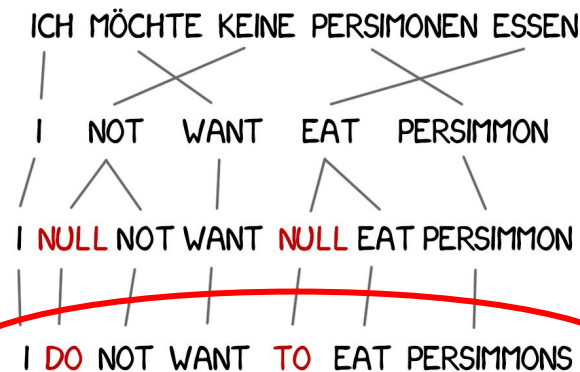
"man in black shirt is playing guitar."



"construction worker in orange safety vest is working on road."



"two young girls are playing with
lego toy."



A valódi címke (ground truth) nem egyértelmű. Többféleképpen is megfogalmazható ugyanaz a jelentés. Honnan tudjuk, hogy mennyire jó a becslésünk? **Hogy számoljuk ki, mekkorát hibáztunk** (mekkora a költség)?

Felügyelt tanulás

A felügyelt tanulás problémája:

- Az állatok / emberek általában nem szorulnak rá konkrét $\{(x, y)\}$ input-output párok jelenlétére, ahhoz, hogy tanuljanak
- A címkézett adat előállítása **nagy mértékű humán munkát igényelhet**, címkézetlen adat viszont rengeteg van



Felügyeletlen tanulás

Címke hiányában a feladat nem egyértelmű...

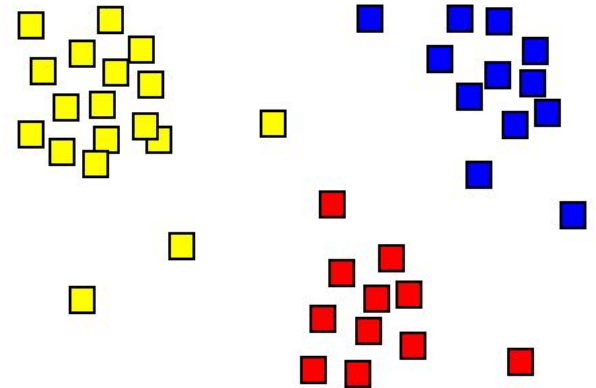
Többféle konkrét feladat lehetséges:

- **Klaszterezés:** Valamilyen hasonlósági reláció alapján a mintaelemek csoportosítása
- **Tömörítés:** Az adat jellegének ismeretében sokkal jobb tömörítési ráták érhetők el, mint a hagyományos algoritmusokkal
- **Példagenerálás:** Új, az adatbázishoz hasonló mintaelemek generálása (*a minta eloszlásának tanulása*)
- ...

Felügyeletlen tanulás

Klaszterezés

- **Marketing, célzott reklám, ajánlórendszerek:** hasonló ügyfelek csoportosítása célzott ajánlatokhoz
- **Klasszikus computer vision,** pl. képszegmentálás
- **Anomália (outlier) detektálás**



Felügyeletlen tanulás

Tömörítés / dimenziócsökkentés

- Bizonyos fajta adat jellegéhez illeszkedő tömörítés



Felügyeletlen tanulás

Tömörítés / dimenziócsökkentés

- Bizonyos fajta adat jellegéhez illeszkedő tömörítés

Emberi arcokon betanult tömörítés (PCA).
Rekonstrukció (x kalap) a tömörített reprezentációból, különböző tömörítési arányokkal.

0.1%

Eredeti input (x)



Felügyeletlen tanulás

Tömörítés / dimenziócsökkentés

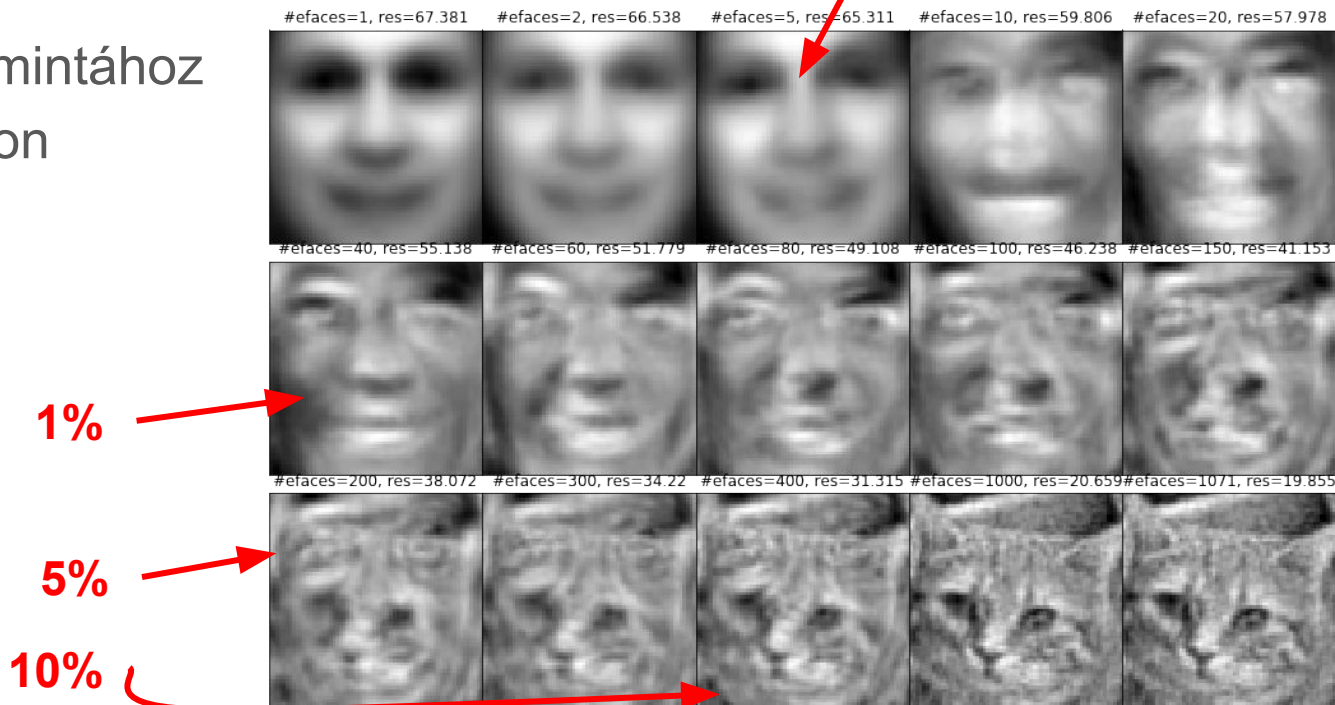
- Csak a tanítómintához hasonló inputon működik jól

Eredeti input (x)



Emberi arcokon betanult tömörítés (PCA).
Rekonstrukció (x kalap) a tömörített reprezentációból, különböző tömörítési arányokkal.

0.1%



Felügyeletlen tanulás

Tömörítés / dimenziócsökkentés

- Kevesebb tárhely, sávszélesség használat.
- Tömörebb (kevesebb változóból álló) reprezentációval sokszor könnyebb dolgozni és könnyebb lehet megtalálni a hasonló elemeket.
- 2-3 dimenziós reprezentációban a mintaelemek elhelyezkedése még vizualizálható is (pontok a síkon/térben).

Felügyeletlen tanulás

Tömörítés / dimenziócsökkentés

- Kevesebb tárhely, sávszélesség használat.
- Tömörebb (kevesebb változóból álló) reprezentációval sokszor könnyebb dolgozni és könnyebb lehet megtalálni a hasonló elemeket.
- 2-3 dimenziós reprezentációban a mintaelemek elhelyezkedése még vizualizálható is (pontok a síkon/térben).

A tömörítés / dimenzió redukció feladata jól definiált, azonban a hatékony tömörítés kihasználja, hogy a tömörített adatok hasonló jellegűek. Sok különböző fajta adat tömörítéséhez nehéz lenne egyedi algoritmusokat írni.
→ **gépi tanuláshoz ideális feladat**

Felügyeletlen tanulás

Tömörítés / dimenziócsökkentés

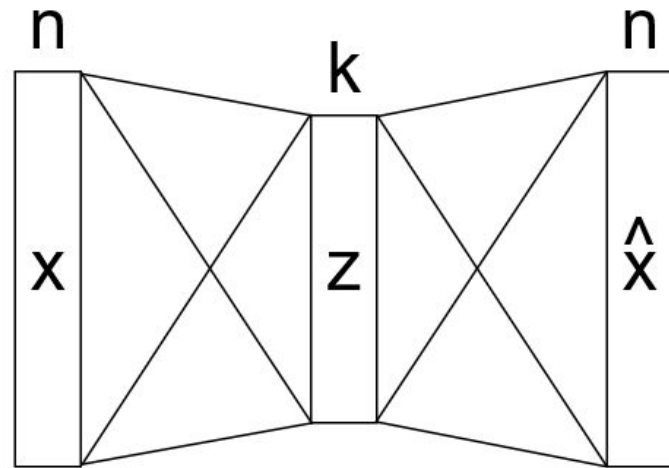
A feladat megfogalmazható formálisabban is:

x: Az input (tömörítendő mintaelem)

h: A tömörítést végző hipotézisfüggvény,
pl. egy (autencoder) neuronháló

$$h_{\theta}(x) = \hat{x} \approx x$$

J költség: \hat{x} közelítse meg minél
jobban x -et (jó rekonstrukció).



Felügyeletlen tanulás

Tömörítés / dimenziócsökkentés

A feladat megfogalmazható formálisabban is:

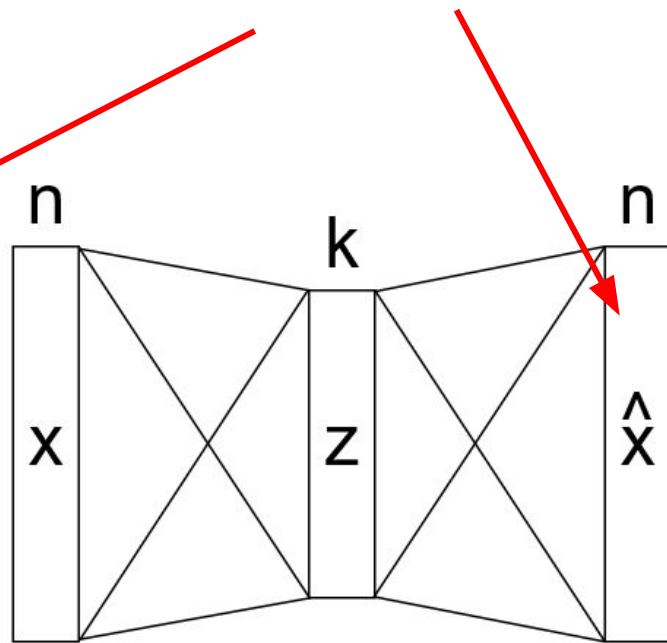
x: Az input (tömörítendő mintaelem)

h: A tömörítést végző hipotézisfüggvény,
pl. egy (autencoder) neuronháló

$$h_{\theta}(x) = \hat{x} \approx x$$

J költség: \hat{x} közelítse meg minél
jobban x -et (jó rekonstrukció).

Címke (y) hiányában az
inputot (x) tanuljuk
helyreállítani a tömörített
reprezentációból (z).



Felügyeletlen tanulás - alkalmazások

Zajszűrés (denoising)



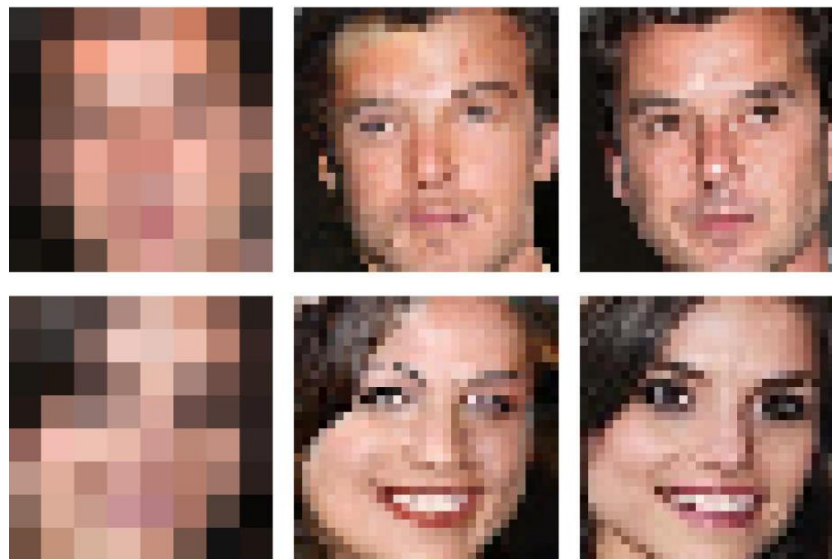
Denoising: Fénykép minőségének javítása



Inpainting: hiányzó rész kitöltése

Felügyeletlen tanulás - alkalmazások

Szuperrezolúció - “intelligens” felskálázás



(leskálázott) input (x')

eredeti input becslése (x kalap)

eredeti input (x)

Felügyeletlen tanulás

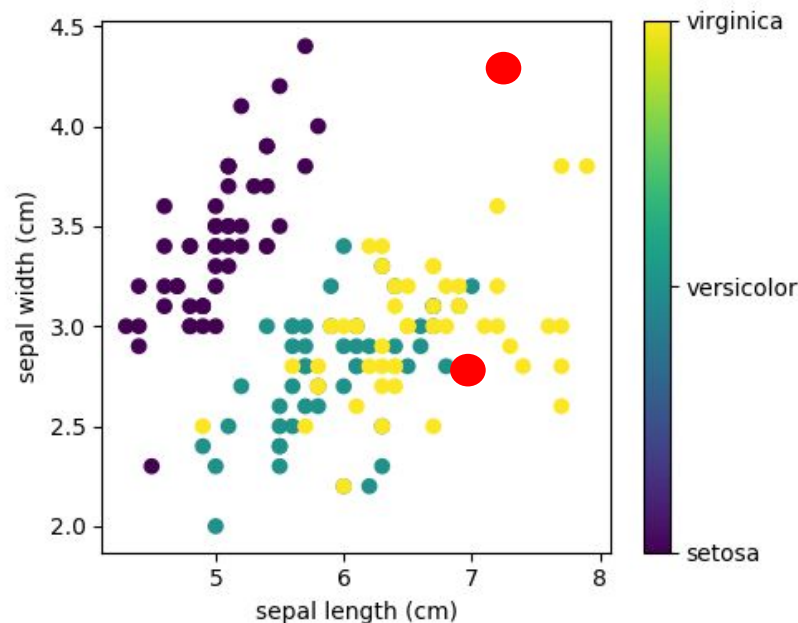
Példagenerálás (minta eloszlásának tanulása)

A tanítóhalmazban megtalálható mintaelemekhez hasonló, új elemek generálása.

Felügyeletlen tanulás

Példagenerálás (minta eloszlásának tanulása)

A tanítóhalmazban megtalálható
mintaelemekhez hasonló,
új elemek generálása.



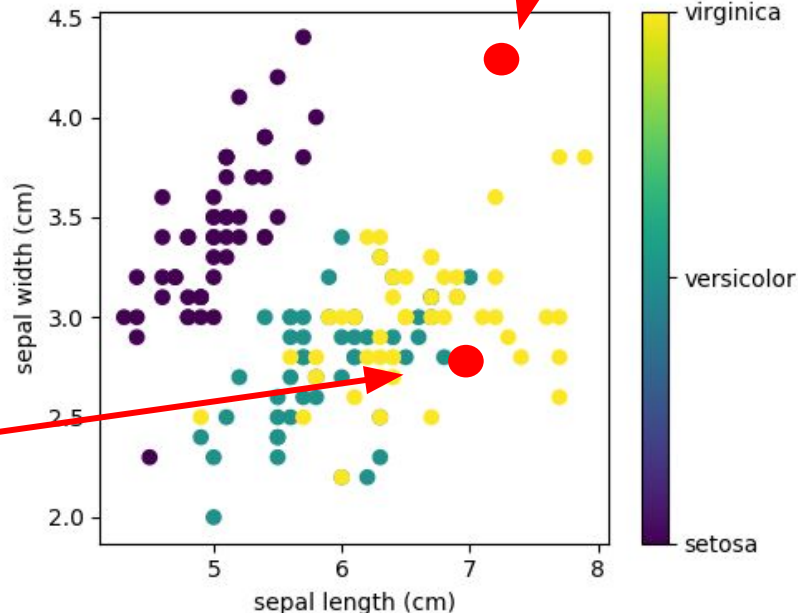
Felügyeletlen tanulás

Példagenerálás (minta eloszlásának tanulása)

A tanítóhalmazban megtalálható mintaelemekhez hasonló, új elemek generálása.

Ez az új mintaelem jó: a szirmok méretek hasonlóak a valódi virágokhoz

Ez az új mintaelem valószínűtlen: túl nagyok a szirmai



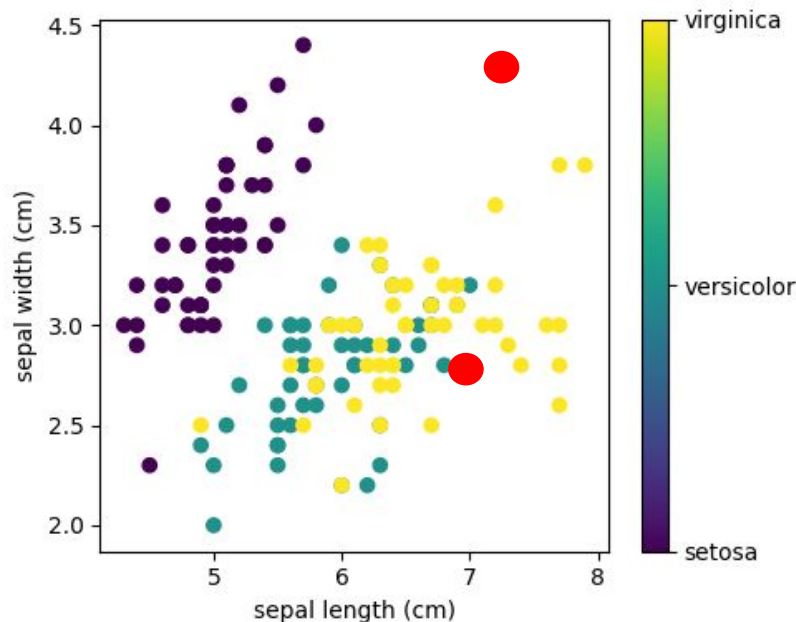
IRIS adatbázis: 3 fajta virág példányainak szirmainak hossza és szélessége (2 változó)

Felügyeletlen tanulás

Példagenerálás (minta eloszlásának tanulása)

2 változó esetén szabad szemmel is látjuk, hogy egy-egy generált minta-példa mennyire valószínű.

Sok változó esetén ez nem ilyen egyszerű: a minta *ritka* lesz a térben.



Felügyeletlen tanulás

Példagenerálás (minta eloszlásának tanulása)

A deep learning eszközeivel sok dimenziós adathoz is tudunk új példákat generálni.

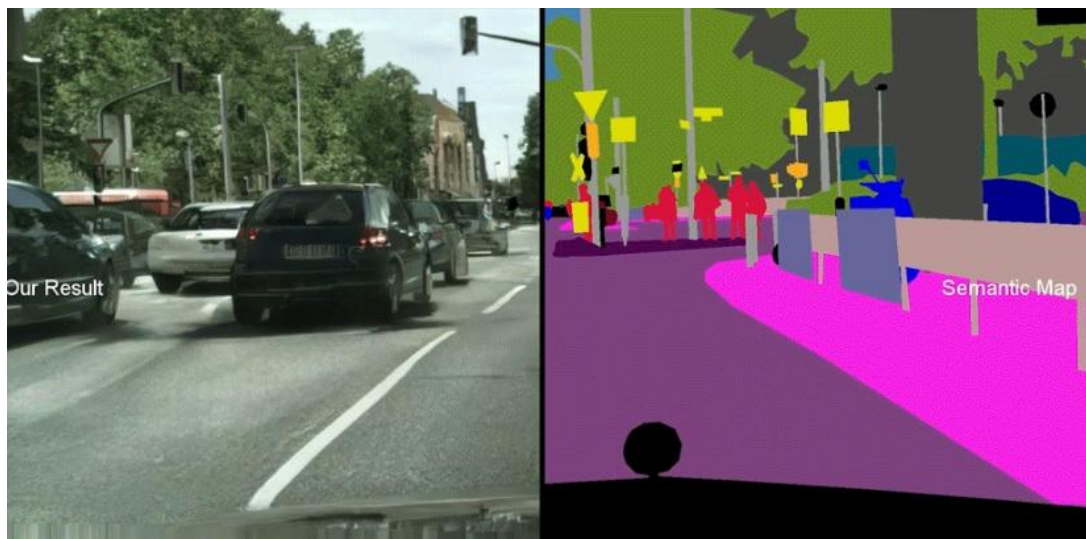
<https://www.youtube.com/watch?v=36IE9tV9vm0> (Karras et al., 2018)



Felügyeletlen tanulás - alkalmazások

Példagenerálás - nagyfelbontású textúra szintézise

https://www.youtube.com/watch?v=3AlpPlzM_gs (Wang et al., 2018)



Felügyeletlen tanulás - alkalmazások

Példagenerálás - DeepFake ("Face swap")

<https://www.youtube.com/watch?v=bVy2xwW3MHc>



Felügyeletlen tanulás - alkalmazások

Példagenerálás - Shakespeare drámák generálása (2015)

at first:

tyntd-iafhatawiaoihrdemot lytdws e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrqd t o idoe ns,smtt h ne etie h,hregtrs nigtike,aoaenns lng

↓
train more

"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwv fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."

↓
train more

Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and offer.

↓
train more

"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.

Felügyeletlen tanulás - alkalmazások

Példagenerálás - C forráskód generálása (2015)

Szintaktikailag (talán) helyes...

```
static void do_command(struct seq_file *m, void *v)
{
    int column = 32 << (cmd[2] & 0x80);
    if (state)
        cmd = (int)(int_state ^ (in_8(&ch->ch_flags) & Cmd) ? 2 : 1);
    else
        seq = 1;
    for (i = 0; i < 16; i++) {
        if (k & (1 << 1))
            pipe = (in_use & UMXTHREAD_UNCCA) +
                ((count & 0x00000000ffffffff) & 0x0000000f) << 8;
        if (count == 0)
            sub(pid, ppc_md.kexec_handle, 0x20000000);
        pipe_set_bytes(i, 0);
    }
    /* Free our user pages pointer to place camera if all dash */
    subsystem_info = &of_changes[PAGE_SIZE];
    rek_controls(offset, idx, &soffset);
    /* Now we want to deliberately put it to device */
    control_check_polarity(&context, val, 0);
    for (i = 0; i < COUNTER; i++)
        seq_puts(s, "policy ");
}
```

Felügyeletlen tanulás - alkalmazások

OpenAI GPT v3 (2020) - cikkek generálása pusztán a cím/alcím megadásával

Title: United Methodists Agree to Historic Split
Subtitle: Those who oppose gay marriage will form their own denomination
Article: After two days of intense debate, the United Methodist Church has agreed to a historic split - one that is expected to end in the creation of a new denomination, one that will be "theologically and socially conservative," according to The Washington Post. The majority of delegates attending the church's annual General Conference in May voted to strengthen a ban on the ordination of LGBTQ clergy and to write new rules that will "discipline" clergy who officiate at same-sex weddings. But those who opposed these measures have a new plan: They say they will form a separate denomination by 2020, calling their church the Christian Methodist denomination.

The Post notes that the denomination, which claims 12.5 million members, was in the early 20th century the "largest Protestant denomination in the U.S.," but that it has been shrinking in recent decades. The new split will be the second in the church's history. The first occurred in 1968, when roughly 10 percent of the denomination left to form the Evangelical United Brethren Church. The Post notes that the proposed split "comes at a critical time for the church, which has been losing members for years," which has been "pushed toward the brink of a schism over the role of LGBTQ people in the church." Gay marriage is not the only issue that has divided the church. In 2016, the denomination was split over ordination of transgender clergy, with the North Pacific regional conference voting to ban them from serving as clergy, and the South Pacific regional conference voting to allow them.

Legjobb cikk: az olvasók 88%-a emberi író tippelt.

Felügyeletlen tanulás - alkalmazások

OpenAI GPT v3 (2020) - cikkek generálása pusztán a cím/alcím megadásával

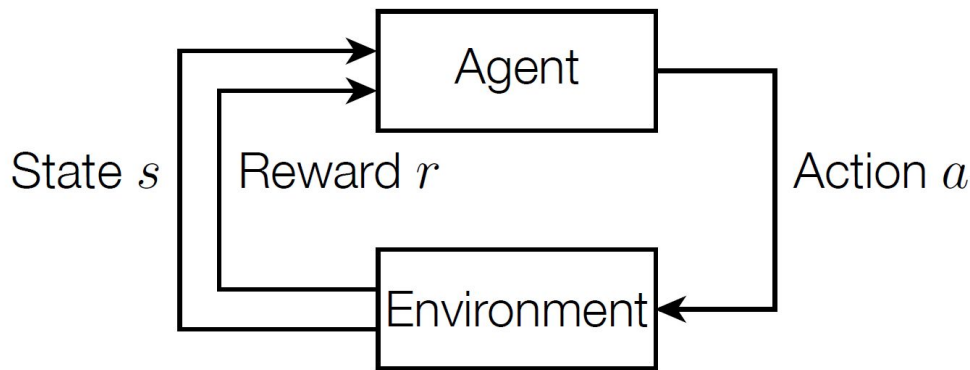
Title: Star's Tux Promise Draws Megyn Kelly's Sarcasm
Subtitle: Joaquin Phoenix pledged to not change for each awards event
Article: A year ago, Joaquin Phoenix made headlines when he appeared on the red carpet at the Golden Globes wearing a tuxedo with a paper bag over his head that read, "I am a shape-shifter. I can't change the world. I can only change myself." It was a promise to not change to fit into the Hollywood mold: "I think that's a really special thing, to not change yourself. I think it's a really special thing to say, 'This is what's inside of me, I'm proud of it, and I'm not going to be ashamed because of the way that someone else thinks I should be.'" Now, it's the Oscars, and Phoenix is at it again. But this time, his publicist is saying he'll be wearing a tux no matter what.
Megyn Kelly was not impressed, and she let him have it on The Tonight Show. "You know, I feel like, I feel like you could have worn the tux," she says. "But you're saying you're a shape-shifter. I don't know if you can change your tux, but you can change your mind. You can change your mind. You can change your mind." Phoenix says he did, but it didn't stick. "I was like, 'Okay, I'm going to wear a tuxedo to this thing.' And then I thought, 'I don't want to wear a tuxedo to this thing.'" Kelly goes on to encourage him to change his mind again, but Phoenix says it's too late: "I'm committed to wearing this."

Legrosszabb cikk: az olvasók 39%-a tippelt emberi író.

Megerősítéses tanulás \rightarrow csak MSc-n :(

Markov Döntési Folyamat (MDP):

- (\mathbf{x}, \mathbf{y}) input-címke párok helyett (\mathbf{s}, \mathbf{a}) állapot-akció párok
- Az akciónk nagyban befolyásolja a következő állapotunkat
- \mathbf{J} költség helyett \mathbf{r} jutalom (nem feltétlenül azonnal kapjuk)
- **Cél:** hosszútávon maximalizáljuk a jutalmak összegét



Megerősítéses tanulás → csak MSc-n :(

Példa: Automata autó

- **s - állapot:** Az autó sebessége, az autó környezete, stb.
- **a - akció:** Gázpedál, fékpedál, kormány forgatása, stb.
- **r - jutalom:**
 - Eljutottunk a célhoz → nagy jutalom
 - Elütöttünk valakit → nagy büntetés
 - Nem tartottuk be a sebességhatárt → kisebb büntetés
 - ...

Megerősítő tanulás → csak MSc-n :(

Eredményei:

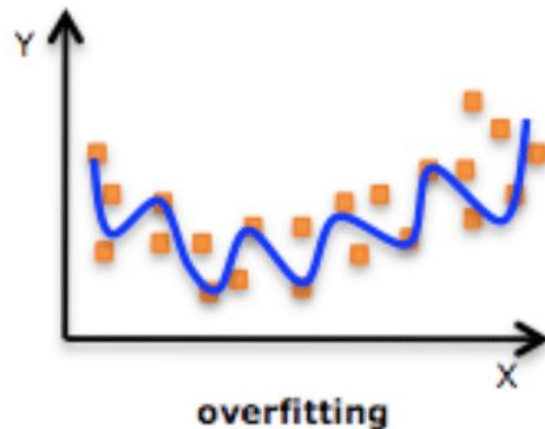
- Emberfeletti Atari 2600 játéktudás
- **AlphaGo** (emberfeletti Go játéktudás)
- **AlphaStar** (top liga StarCraft 2 játéktudás)

A bonyolult, való életbeli alkalmazásokban még nem működik igazán jól...

A mesterséges intelligencia főbb kérdései

Általánosítóképesség

A betanult modellünk memorizálja a tanítópéldákat egyenként, vagy valóban “hasznos” dolgokat tanul?

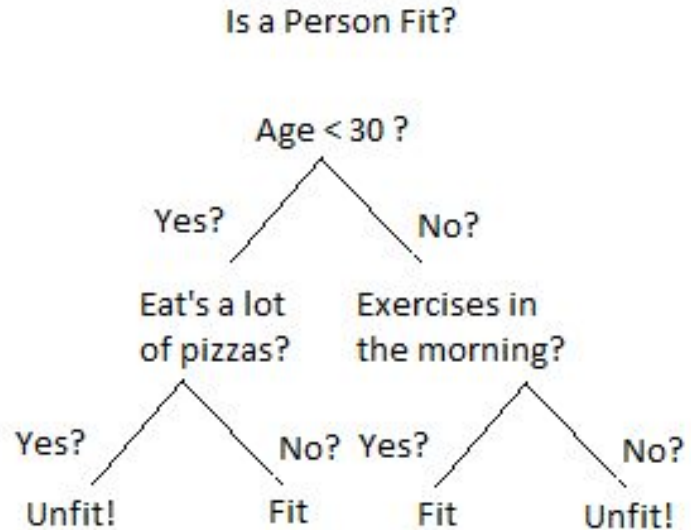


A mesterséges intelligencia főbb kérdései

Magyarázhatóság

Némelyik klasszikus gépi tanulási módszer könnyen magyarázható modellt tanul.

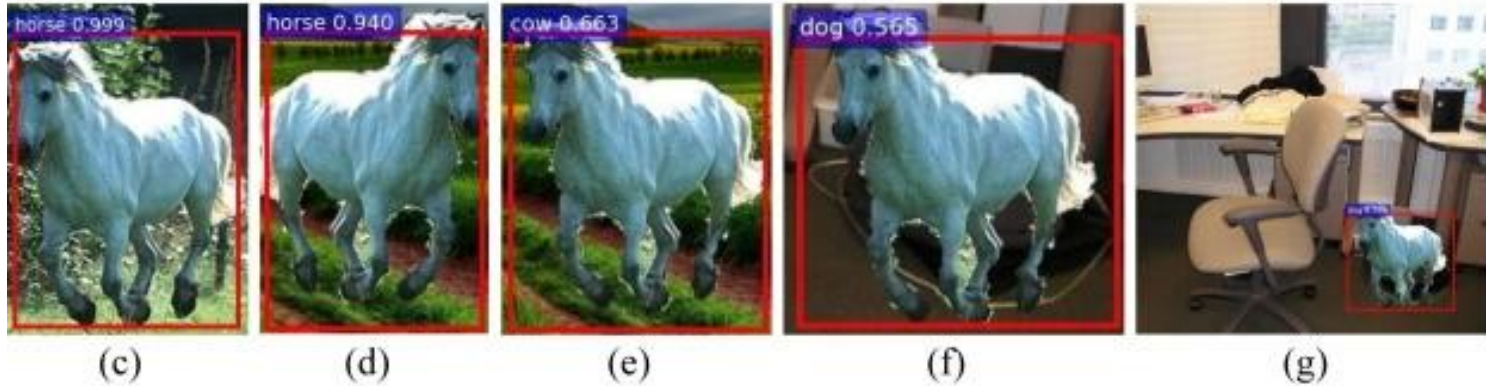
Például a döntési fák ilyenek:



A mesterséges intelligencia főbb kérdései

Magyarázhatóság

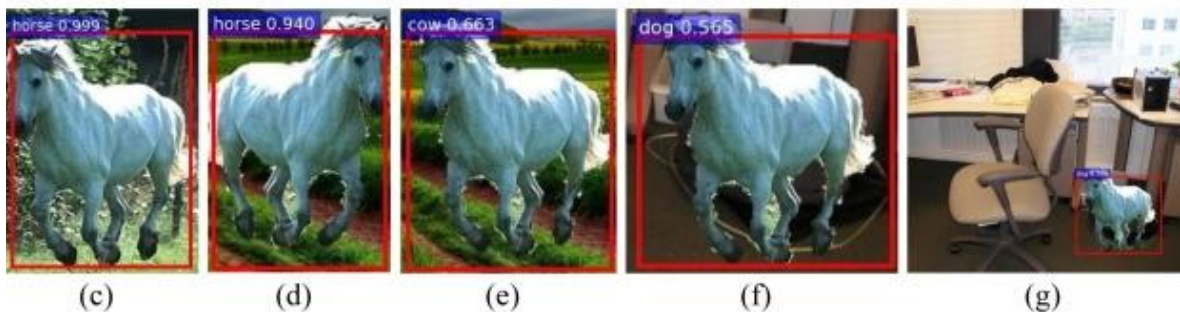
A neuronhálók által tanult tudás nehezen magyarázható.



A mesterséges intelligencia főbb kérdései

Magyarázhatóság

A neuronhálók által tanult tudás nehezen magyarázható.



Objektumokat (állatokat) kategorizáló neuronháló becslései: egy ló képére a háló a ló környezetétől függően “ló”, “tehén” és “kutya” címkéket becsül. A neuronháló tehát, jobban megtanulta a környezet kinézetéből becsülni az állatot, mint magából az állat kinézetéből...

A mesterséges intelligencia főbb kérdései

Magyarázhatóság

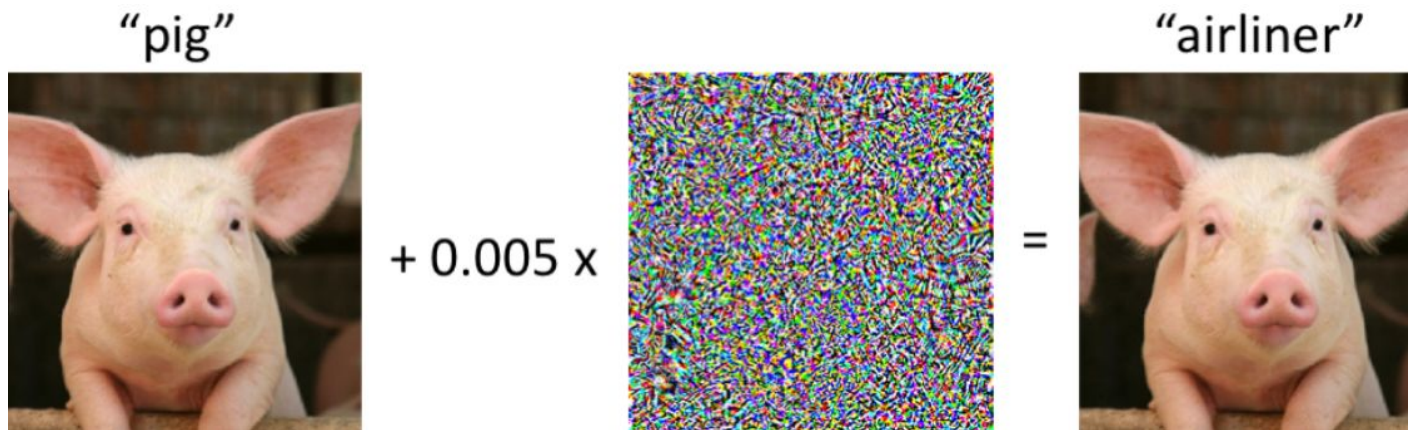
A kritikus alkalmazásokban használt szoftvereket behatóan tesztelik, formálisan verifikálják sokszor.

A deep learning módszerei csak tapasztalati úton tesztelhetők, **nem tudjuk róluk megmondani, hogy “logikus módon” hozták-e a döntéseiket...**



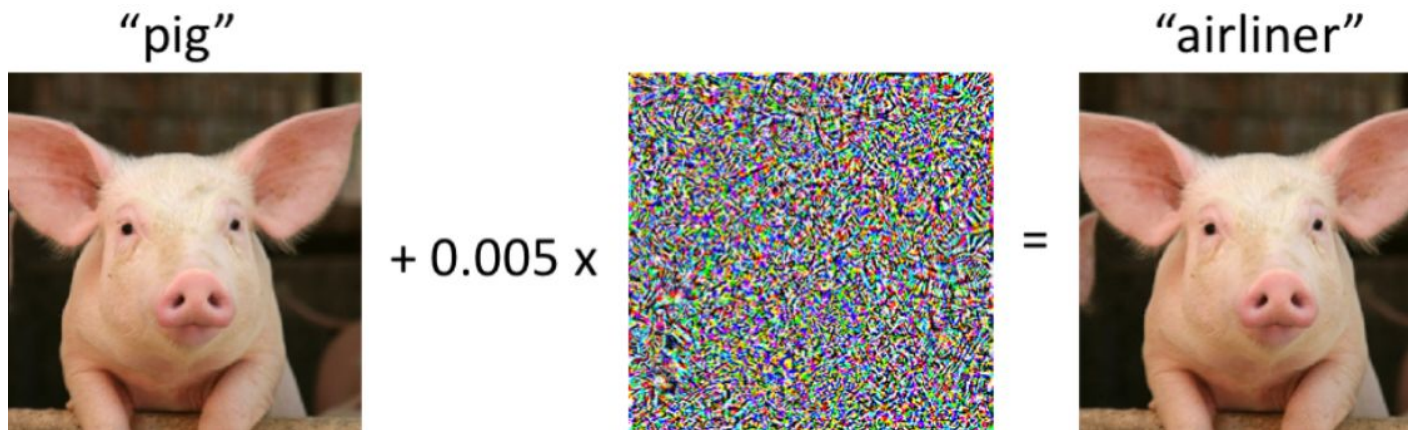
A mesterséges intelligencia főbb kérdései

Biztonság - adversarial attack



A mesterséges intelligencia főbb kérdései

Biztonság - adversarial attack



Adversarial attack: Akár a neuronháló architektúrája és paramétereinek ismerete nélkül kereshető kismértékű speciális zaj, mellyel módosítva az inputot, a becslés drasztikusan változik.

A mesterséges intelligencia főbb kérdései

Biztonság - adversarial attack

Sajnos a táblákra helyezett fóliával is működik a megtévesztés...



A mesterséges intelligencia főbb kérdései

Magyarázhatóság

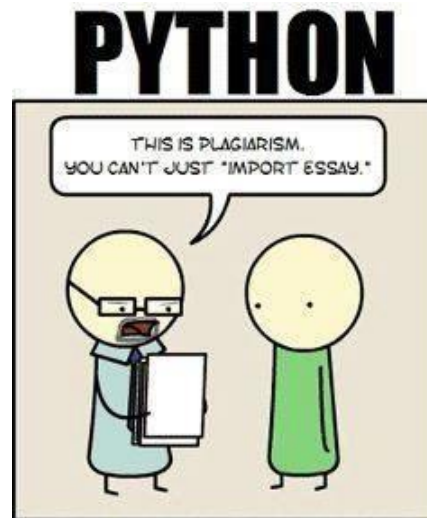
A mesterséges neuronhálók által tanult tudás nem magyarázható jól.

De vajon az emberek meg tudják magyarázni miről ismernek fel például egy ismerős arcot?

Szoftver

Python

- Magas szintű
- Automatikus memóriakezelés
- Futásidejű típushozzárendelés és ellenőrzés
- Értelmezett (interpretált) nyelv
- Egyszerű szintaxis, limitált nyelvi eszközök
- Hordozható
- Rengeteg eszköz az elérhető könyvtárakban

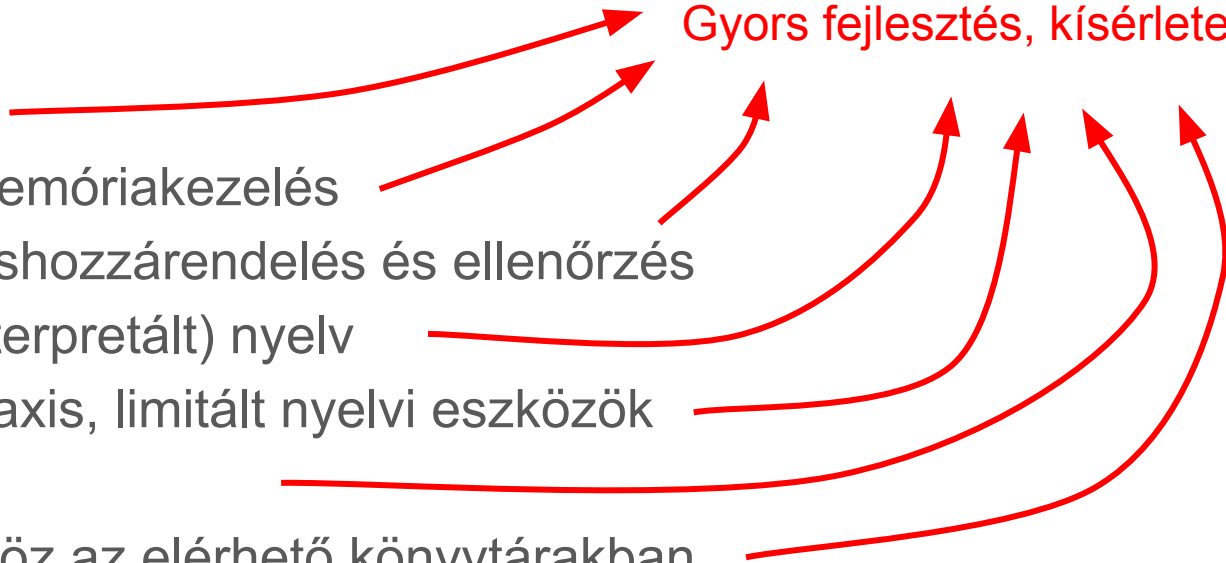


Szoftver

Python

- Magas szintű
- Automatikus memóriakezelés
- Futásidejű típushozzárendelés és ellenőrzés
- Értelmezett (interpretált) nyelv
- Egyszerű szintaxis, limitált nyelvi eszközök
- Hordozható
- Rengeteg eszköz az elérhető könyvtárakban

Kényelmes.
Gyors fejlesztés, kísérletezés.



Szoftver

Python

- Magas szintű
- Automatikus memóriakezelés
- Futásidejű típushozzárendelés és ellenőrzés
- Értelmezett (interpretált) nyelv
- Egyszerű szintaxis, limitált nyelvi eszközök
- Hordozható
- Rengeteg eszköz az elérhető könyvtárakban

Lassú végrehajtás.

A hibák többsége futásidőben derül ki.

Szoftver

Python

“Lassú végrehajtás”.

Ha nem hatékony a Python, miért lett népszerű a számításigényes gépi tanulásban?

Szoftver

Python

“Lassú végrehajtás”.

Ha nem hatékony a Python, miért lett népszerű a számításigényes gépi tanulásban?

**A számításigényes műveleteinket hatékony,
más nyelven írt könyvtárak segítségével valósítjuk meg.
A Python-t csak a szkriptelésükre használjuk!**

Szoftver

NumPy - Két véletlen mátrix generálása és összeszorzása (400 x 400)

Naiv Python: 14.79 sec

```
matrix1 = [[random.random() for col_idx in range(matrix_dim)] for row_idx in range(matrix_dim)]
matrix2 = [[random.random() for col_idx in range(matrix_dim)] for row_idx in range(matrix_dim)]
matrix_out = [[0. for col_idx in range(matrix_dim)] for row_idx in range(matrix_dim)]

for row1_idx in range(matrix_dim):
    for col2_idx in range(matrix_dim):
        for item_idx in range(matrix_dim):
            matrix_out[row1_idx][col2_idx] += matrix1[row1_idx][item_idx] * matrix2[item_idx][col2_idx]
```

NumPy: 0.019 sec

```
matrix1 = np.random.rand(matrix_dim, matrix_dim)
matrix2 = np.random.rand(matrix_dim, matrix_dim)

return np.matmul(matrix1, matrix2)
```

Szoftver

NumPy - Két véletlen mátrix generálása és összeszorzása (400 x 400)

Naiv Python: 14.79 sec

```
matrix1 = [[random.random() for col_idx in range(matrix_dim)] for row_idx in range(matrix_dim)]
matrix2 = [[random.random() for col_idx in range(matrix_dim)] for row_idx in range(matrix_dim)]
matrix_out = [[0. for col_idx in range(matrix_dim)] for row_idx in range(matrix_dim)]

for row1_idx in range(matrix_dim):
    for col2_idx in range(matrix_dim):
        for item_idx in range(matrix_dim):
            matrix_out[row1_idx][col2_idx] += matrix1[row1_idx][item_idx] * matrix2[item_idx][col2_idx]
```

NumPy: 0.019 sec

```
matrix1 = np.random.rand(matrix_dim, matrix_dim)
matrix2 = np.random.rand(matrix_dim, matrix_dim)

return np.matmul(matrix1, matrix2)
```

 Vektorizált programozás!

Szoftver

NumPy - Vektorizált programozás, vektorműveletek

```
a = np.arange(5, dtype=np.int32)
b = np.arange(5, dtype=np.int32)

print(a)          # -> [0 1 2 3 4]
print(b)          # -> [0 1 2 3 4]
print(a+b)        # -> [0 2 4 6 8]
print(a*b)        # -> [0 1 4 9 16]
```

Szoftver

NumPy - Egy tömb egyszerre több elemének hivatkozása

```
a = np.array([1,0,7,-2,5,-1], dtype=np.int32)

even_mask = a % 2 == 0
small_mask = a < 3

print(even_mask)           # [False  True False  True False False]
print(small_mask)         # [ True  True False  True False  True]
small_and_even_mask = small_mask & even_mask
print(small_and_even_mask) # [False  True False  True False False]

small_and_even_idxs, = np.where(small_and_even_mask)
print(small_and_even_idxs) # [1 3]

print(a[small_and_even_mask]) # [ 0 -2]
print(a[small_and_even_idxs]) # [ 0 -2]
```

Szoftver

Vektorizált programozás

- A grafikus kártyák a vektorizált műveletekre optimalizáltak.
- A gépi tanulásban használt műveletek szinte mindig vektorizálhatóak.
- A vektorizált programozással írt kód általában sokkal tömörebb.

Szoftver

NumPy - N dimenziós tömbök

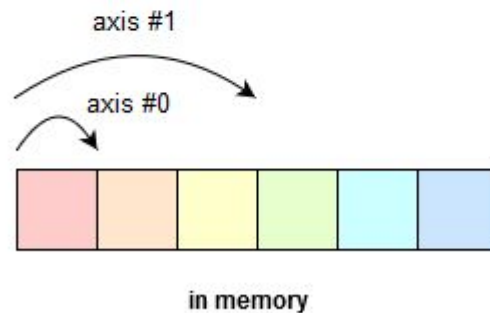
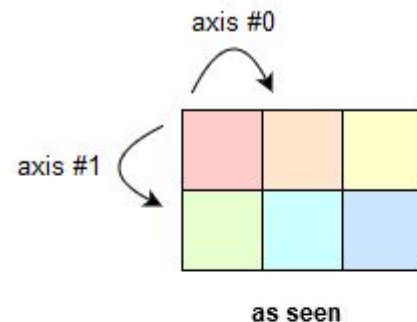
```
a = np.arange(6, dtype=np.int32).reshape((3,2))

print(a)          # [[0 1]
                  #  [2 3]
                  #  [4 5]]

print(a[1, 0])    # 2
print(a[1, :])    # [2,3]
print(a[:, 1])    # [1,3,5]

print(a[1:, 1:])  # [[3]
                  #  [5]]

print(a[0, ::-1]) # [1,0]
print(a[[1,1,2], [False, True]]) # [3,3,5]
```



Szoftver

NumPy - Broadcasting

```
a = np.arange(6, dtype=np.int32).reshape((2,3))

print(a)          # [[0 1 2]
                  #   [3 4 5]

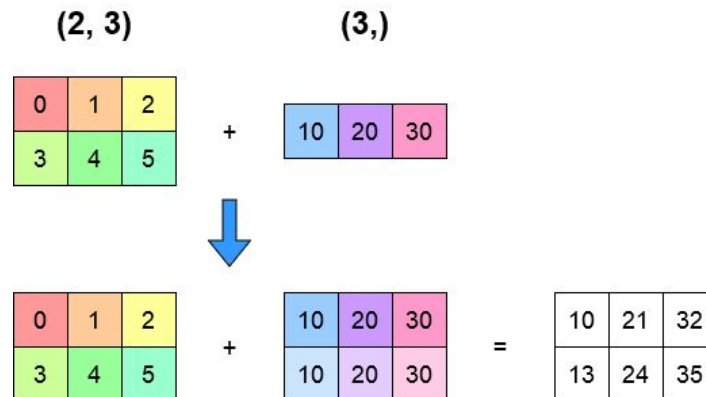
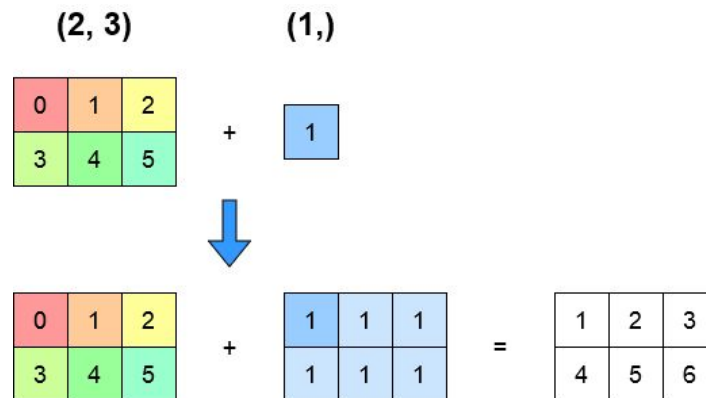
b = a + 1
print(b)          # [[1 2 3]
                  #   [4 5 6]

c = a + [10,20,30]
print(c)          # [[10 21 32]
                  #   [13 24 35]
```

Szoftver

NumPy - Broadcasting

```
a = np.arange(6, dtype=np.int32).reshape((2,3))  
  
print(a)           # [[0 1 2]  
                   #  [3 4 5]  
  
b = a + 1  
print(b)           # [[1 2 3]  
                   #  [4 5 6]  
  
c = a + [10,20,30]  
print(c)           # [[10 21 32]  
                   #  [13 24 35]
```



Szoftver

Keras - (Tensorflow-ra épülő) magas szintű könyvtár neuronhálókhöz

```
model = Sequential()  
model.add(Dense(20, activation='relu', input_dim=50))  
model.add(Dense(1, activation='sigmoid'))  
  
model.compile(loss='binary_crossentropy', optimizer='sgd')  
  
model.fit(X, y, epochs=100)
```

Szoftver

Google Colab

<https://colab.research.google.com>

Irodalom, egyéb anyagok

- Youtube playlist “Machine Learning — Andrew Ng, Stanford University [FULL COURSE]”
- deeplearning.ai kurzusok
- Stanford CS231n (konvolúciós hálók)
- Francois Chollet: Deep Learning with Python
- Google...

Haladó szint:

- I. Goodfellow, Y. Bengio, A. Courville: Deep Learning

Angol nyelvű megnevezések Google kereséshez...

- Minta, (adat) → sample
- Mintaelem, adatpont, példa → data sample, sample point, example, observation
- Tanítóhalmaz → training set
- (Input) változó → (input) variable, feature
- Becslés → prediction, estimation
- Címke → label, output variable, target variable, ground truth
- Hipotézisfüggvény → hypothesis function
- Költségfüggvény → cost function, loss function
- Paraméter, súly → parameter, weight