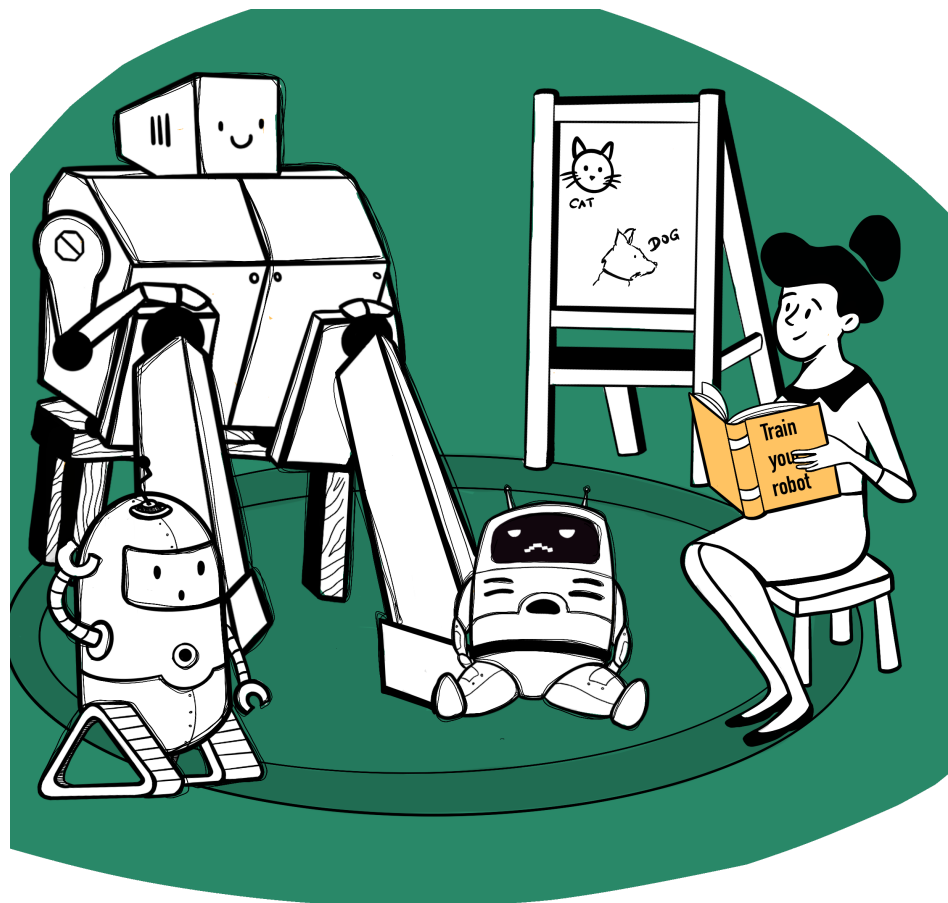


## 4. modul

### Felügyelt tanulás

"A **modell** olyan **paramétereiből** áll, amelyeket nem a programozó választ ki, hanem az algoritmus alakítja ki egy **tréningnek/tanításnak** nevezett folyamat során."



# A modulról

Ez a modul a **felügyelt tanulásról (FT)** szól. Célja, hogy a tanulók alapszinten megértsék, mi az **FT**, mire jó és mire nem, valamint hogyan hozhatók létre felügyelt tanulási algoritmusok. A modul inkább a gyakorlatra összpontosít, ezért a tanulók saját algoritmusokat fognak betanítani, így első kézből tapasztalhatják meg a felmerülő lehetőségeket és nehézségeket. Az eljárás alapjául szolgáló technikai jellegű rész itt most nagyrészt kimarad, mivel vannak más modulok a konkrét módszerek, például **neurális hálózatok** megismerésére - ez a tudás pedig nem szükséges az **FT** működésének és hasznosíthatóságának alapvető megértéséhez.

## Célok

A tanulók képesek lesznek:...

- ...elmagyarázni a felügyelt tanulást, mint leképezőfüggvényt
- ...megnevezni a tréningadatok jelentőségét és követelményeit
- ...megtippelni, hogy egy alkalmazás felügyelt tanulást alkalmaz-e
- ...megnevezni a felügyelt tanulás problémáit és korlátait
- ...saját felügyelt tanuló-modell tanítani, és egy alkalmazásban hasznosítani

## Beosztás

Idő	Tartalom
30 perc	Bevezetés
30 perc	Feladat - Tanítsd be az első modelled
15 perc	Elmélet - Adatokra való támaszkodás
30-60 perc	Feladat - Játékvezérlő
30 perc	Vita + Kvíz - Lehetőségek és korlátok

# Bevezetés

A bevezető rész célja, hogy a hallgatók megismerkedjenek a **felügyelt tanulás (FT)** alapfogalmaival és kifejezéseivel. A modul a kapcsolódó diákra épül, amelyek nemcsak az **FT** bevett alkalmazási példáit mutatják be, hanem képet nyújtanak arról is, általánosságban mit csinál egy **FT**-algoritmus, és hogyan lehet tanítani.

## Valós példák

(2 - 8. dia)

Az első diákon az **FT** algoritmusok számos alkalmazási példája kerül bemutatásra, amiből egyértelműen látszik, milyen sok területen használják.

Az **FT** legkiemelkedőbb példái közé tartozik a **képosztályozás**, és a képek részleteire vonatkozó lévő **objektumfelismerés**. Az arcfelismerés például használható a fényképezőgép/kamera fókuszának automatikus beállításában vagy a közösségi média oldalak esetében automatikus névcímkék létrehozására. Az objektumfelismerés célja a képen szereplő különböző tárgyak felismerése és azonosítása. Ezért általában az első lépés a szegmentálás, ahol a képet különböző területekre osztják, amelyeket aztán a rendelkezésre álló címkék valamelyikével látnak el (osztályoznak). A felsorolt lépéseket manapság többnyire **FT** és **mély neurális hálózatok (DNN)** segítségével végzik. Még a gyártóiparban is használják az **FT**-t és a képosztályozást, például a sérülések vagy szerelési hibák automatikus felismerésére. Ez az egyik olyan terület, ahol a mesterséges intelligencia egyre több emberi munkaeerőt vált ki az alacsonyabb költségek - és olykor az alacsonyabb hibaarány - miatt. Az olyan területeken azonban, mint az orvostudomány, a betegségek felismerésére szolgáló algoritmusokat az orvosi felügyelettel együtt használják, segítve őket a helyes következtetés levonásában. Például úgy, hogy a vizsgálatok során rámutatnak a potenciálisan rákos megbetegedést jelző foltokra, amelyek esetleg elkerülhették az orvosok figyelmét.

A következő alkalmazási példa a beszédfelismerés és a **természetes nyelvfeldolgozás (NLP)**. Az ilyen példákban képek helyett hanghullámokat használnak az elhangzott szavak és más hangok megkülönböztetésére. A szavak felismerésük után rögzíthetők, ami például automatikus feliratgenerálásban használható fel. A mondatokat továbbá elemezni lehet jelentésük szempontjából, így egyes alkalmazások, köztük a digitális asszisztensek (Alexa, Siri, Google Now, ...) összetett parancsokat és utasításokat érthetnek meg, melyekre értelmes módon

reagálhatnak is válaszokkal vagy cselekedetekkel. A mondatok/szövegek elemzésének ezt a módját a spamszűrők is használják a kéretlen levelek megkülönböztetésére.

Az **FT** algoritmusok másik felhasználási területe az olyan értékek előrejelzése, mint például egy ház költsége vagy egy tervezett film bevétele, olyan különböző információk alapján, mint például a ház mérete vagy a filmben szereplő színészek listája. Ebben az esetben egy **FT**-algoritmust hasonló házakra és azok költségeire vonatkozó adatok alapján tanítanak be/tréningelnek, majd új házak értékének becslésére használják.

Végezetül pedig, az **FT** algoritmusokat arra is használják, hogy a felhasználókat különböző profilokba sorolják, így kínálva számukra személyre szabott hirdetéseket. Mivel a hirdetés jobban megfelel a felhasználó érdeklődési körének, nagyobb valószínűséggel kattint rá a hirdetésre, így a termékadás esélye is megnő.

## Mit csinálnak a felügyelt tanulás algoritmusok valójában?

(9 - 13. dia)

A bemutatott példák után a felügyelt tanulási algoritmusokat, mint leképezőfüggvényeket mutatjuk be. Fontos, hogy valójában egyetlen nagyon egyszerű dolgot csinálnak: adott bemenet esetén adott kimenetet állítanak elő. Nem történik semmilyen mágikus mesterséges intelligencia döntéshozatal, csupán egy matematikai képletről van szó, amely egyértelmű eredményt produkál. Érdekes áttekinteni az előző példákat, és beszélni arról, hogy mi lehet a bemenet és mi a kimenet. Például, ha a bemenet egy kép, a kimenet lehet a különböző, képen szereplő tárgyak helye és címkéje, vagy csak egy egyszerű címke "sérülés" vagy "rendben" felirattal. Ha a bemenet egy hangminta, a kimenet lehet a megfelelő szöveg, ha pedig a bemenet egy filmgyártással kapcsolatos általános információ, a kimenet lehet a produkció várható bevétele. Az ilyen típusú problémákat **osztályozásnak** (adatok címkézése) és **regresszióknak** (értékek előrejelzése) nevezzük.

Fontos megemlíteni, hogy a kimenet általában nem bináris (pl. „A képen 0/100%-os bizonyossággal egy macska szerepel”), hanem fokozatszerű. Például egy macska képe

esetén az algoritmus 99,5%-os bizonyossággal állapíthatja meg, hogy macska, 0,4%-os bizonyossággal, hogy kutya, és 0,1%-os bizonyossággal, hogy hal szerepel a képen.

## Hogyan kell betanítani egy felügyelt tanulási algoritmust?

(14 - 26. dia)

Az utolsó szakaszban a **tanítás/tréning** folyamatát mutatjuk be. Ez azon az elven alapul, hogy a fejlesztő példákat ad az algoritmusnak - pl. ezen a képen (bemenet) egy kutya látható (kimenet), a felügyelt tanulási algoritmus úgy módosítja belső **modelljét** (pl. egy képlet változóit), hogy helyes kimenetet adjon (azaz a kép láttán önállóan is a „kutya” címkét produkálja). Emellett bevezetésre kerül a **tréning-** és **tesztadatok** fogalma.

Öt fontos meghatározás tartozik ide:

### Felügyelt tanulás algoritmus

Egy **algoritmus** egy adott feladat lépésről lépésre történő leírása. Ebben az esetben pontosan meghatározott, hogyan működik az **FT**-folyamat: hogyan tanul, milyen **paramétereket** használ, és azok hogyan befolyásolják az eredményt. Gyakran hasonlítják egy ételrecepthez, ahol részletes utasítások vannak, amelyek helyes követése esetén ízletes eredményt kapunk.

### Paraméterek

A paraméterek olyan adatok, amelyekkel módosítható egy algoritmus eredménye vagy munkafolyamata. A főzős példára visszatérve: egy sütemény paramétere lehet az, hogy mennyire legyen édes. A szakács tehát eldönti a cukor mennyiségét, és ennek megfelelően követi a receptet. Az **FT**-ben a paramétereket a betanítási szakaszban használják, ahol a programozónak kell eldöntenie, hogy az algoritmus milyen módon (hány ismétlés, belső struktúra, mire összpontosítson, ...) tanuljon.

### Felügyelt tanulási modell

A modell olyan paraméterekből áll, amelyeket nem a programozó választ ki, hanem az algoritmus alakítja ki a **tanulásnak/tréningnek** nevezett folyamat során. A főzős példában ez lehet a hozzávalók listája a megfelelő mennyiségekkel, amelyeket az algoritmusnak a képzés során optimalizálnia kell a kívánt (ízletes) eredmény eléréséhez. A képosztályozásban különösen gyakran használnak **neurális hálózatokat** alapmodellként, ezért az ilyen modelleket gyakran csak **hálózatoknak** nevezik.

## Címkézett adatok

Olyan adathalmaz, amely nemcsak a bemenetet, hanem az **FT** algoritmus kimenetét is tartalmazza. Ez lehet például néhány száz macskákról és kutyákról készült kép a hozzájuk tartozó "macska" és "kutya" címkékkel.

## Tréning- és tesztadatok

Általában nagy mennyiségű **címkézett adat** összegyűjtésével kezdünk, amelyeket aztán két halmazra osztunk, a tréning- és a tesztalmazra. A tréningadatokat a modell betanítására, míg a tesztadatokat a modell pontosságának tesztelésére használják. Fontos, hogy ezek **különböző** adatok legyenek (azaz más-más macskákról és kutyákról készült képek), hogy a modell elég általános legyen, és ne csak a tréningadatokban szereplő kutyákat/macskákat osztályozza helyesen.

Ha ezek a feltételek adottak, a modell betanítási folyamata meglehetősen egyszerű:

1. Gyűjtsünk össze egy **címkézett adathalmazt**, és osszuk fel két halmazra (tréning és teszt).
2. **Inicializáljuk a modellt** tetszőleges (gyakran véletlenszerű) értékekkel, és válasszuk meg a **tréning paramétereket**.
3. **Ismételjük meg** ugyanezt a tréningadatokkal (a paraméterek által meghatározott gyakorisággal).
  - Ha a bemenet már adott, **módosítsuk** a modellen belül értékeket úgy, hogy azok jobban előre jelezhessék az adott kimenetet.
4. **Teszteljük a modellt** tesztadatok segítségével, hogy ellenőrizzük a pontosságát.
5. **Addig ismételjük a folyamatot előlről**, a címkézett adatok vagy a tanulási paraméterek módosításával, amíg elégedettek nem leszünk a modell pontosságával.

## Kérdések, és ami még ezután következik

(26. dia)

Ezen a ponton a tanulóknak ismerniük kell az **FT** algoritmusok működésének alapjait. A következő gyakorlatokon keresztül maguk is kipróbálhatják ezt a folyamatot, és megtapasztalhatják, hogy milyen **problémák** merülhetnek fel, és mennyire érzékenyek ezek az algoritmusok a tréningadatokra.

Bár ez a tanfolyam nem merül el a technikai részletekben, érdemes megemlíteni, hogy a feladattól függően többféle algoritmus is használható az **FT**-hez. A legtöbben hallottak már a (mély) neurális hálózatokról, amelyeket legkülönbözőbb problémák megoldására használnak, különösen akkor, ha az adatok meglehetősen összetettek, mondjuk a kép- vagy hangfelismerés esetén. De sok más algoritmus is létezik, mint például a döntési fák, a regressziós görbék, a legközelebbi szomszéd - módszer, és még sok más. <sup>1</sup>

## Tananyag - magyar nyelven

-  SL - Introduction.pdf

## Hivatkozások

1. <https://builtin.com/data-science/supervised-machine-learning-classification>

## Tanítsd be az első modelled

A feladat célja, hogy a tanulóknak kialakuljon egy első benyomás arról, hogyan működik egy **FT** algoritmus betanítása, és milyen nehézségek merülhetnek fel. A fő tanulságnak annak kell lennie, hogy a tanulás/tréning nagymértékben függ a tréningadatoktól, illetve, hogy ez általában egy olyan folyamat, amely sok ismétlést igényel ahhoz, hogy használható eredménnyel járjon.

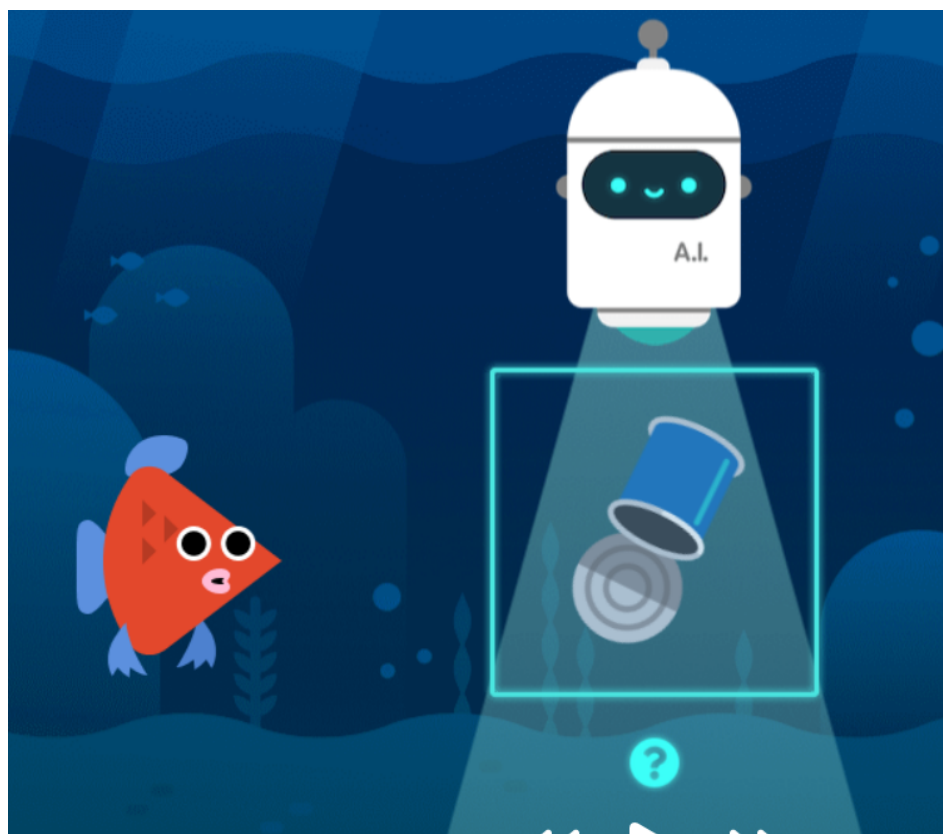
Két lehetséges feladat van:

Az **MI óceán**, amely egy online eszköz, és a **Kategorizálós játék**, amely egy „kézzel fogható”, offline feladat.

### MI óceán

Ez a feladat az „**MI** óceán” című kurzus egy gyakorlatát hasznosítja. A tanulóknak egy **böngészőben futtatható program** segítségével egy robotot kell betanítani arra, hogy különbséget tegyen a halak és a szemét, valamint a különböző vízi állatok között.

Exercises / AI for Oceans







## Kategorizálós játék

Ebben a mindössze **papír-ceruzát igénylő** gyakorlatban a diákok egy **FT** bőrbe bújhatnak, így megtapasztalhatják, milyen nehézségek adódnak a macskák és kutyák képeinek osztályozásakor.

### Exercises / Classification Game



## Célok

A tanulók képesek lesznek...

...elmagyarázni egy **FT** algoritmus tanításának/tréningének folyamatát

...elmagyarázni, hogyan működnek az algoritmusok a különböző jellemzőkre  
összpontosítás révén

...elmagyarázni, hogy a rendszer milyen mértékben függ a tréningadatoktól.

# Adatokra utaltság

Ez a fejezet a **tréningadatokról**, illetve ezen adatok **minőségének** hatásáról és fontosságáról szól. Bár a fejezet példái a képfelismerésre összpontosítanak, a tárgyalt problémák a felügyelt tanulás valamennyi formájára vonatkoznak. A fejezet ezeken a diákon alapul.

## Túlillesztés és alulillesztés

(2-14.dia)

Az **FT** algortimusok tanítása/tréningje esetében a két legnagyobb problémát a **túlillesztés** és az **alulillesztés** esete jelenti.

**Túlillesztés** problémája áll fenn, ha a modell jól működik a **tanítás/tréning során**, de **más adatokkal** rossz eredményeket ad. Ez akkor fordul elő, ha a tanítóhalmaz túl specifikus, vagy a hangsúlyos jellemzők félrevezetőek.

**Alulillesztésről** pedig akkor beszélünk, ha a modell egyáltalán nem tanul semmi hasznosat. Ez akkor fordul elő, ha a bemeneti adatok mennyisége **nem elegendő** vagy azok **túl általánosak**, és nem azonosíthatók kapcsolódó jellemzők. Nézzünk néhány példát!

*Képzeld el, hogy egy olyan modellt szeretnél létrehozni, amely portréfotók alapján különbözteti meg az egyes embereket. Megkéred a barátaidat, hogy adjanak neked néhány száz, a webkamerájukkal rögzített portréképet, és ezeket használod az algoritmusod tanításához. Bár a tesztek eredményei nagyon ígéretesek, amikor meghívod a barátaidat az otthonodba, alig ismer fel valakit helyesen.*

Ez egy klasszikus példája a **túlillesztésnek**, amikor a modell bizonyos körülmények között jól működik, de a környezet megváltoztatásával csúnyán megbukik. A kudarc oka lehet, hogy a modell nem az arcok jellemzőit tanulta meg a várt módon, hanem a háttérben lévő növényekre vagy polcokra koncentrált. A hátterek jellemzői elég egyediek lehetnek, ami akár ahhoz is vezethet, hogy az algoritmus figyelmen kívül hagyja az arcokat, és csak a háttérrel ismeri fel – tehát a bemeneti adatok túl specifikusak voltak.

A probléma mérséklésének egy egyszerű módja, ha **változatosabb adathalmazt** biztosítunk (különböző környezet, megvilágítás, kameraszög, fejpozíciók, zoom-szintek, frizurák, több kép, ...), hogy bizonyos nemkívánatos jellemzők csak ritkán

forduljanak elő. Ez a probléma tovább csökkenthető az **adatok feldolgozásával**, mellyel csökkenthetők a felesleges jellemzők. Ha mindent szürkeárnyalatos képekké alakítunk át, csökkenthetjük a színek befolyását, ahogyan az összes kép telítettségének normalizálása révén is. Az olyan geometriai átalakítások, mint a képek méretezése, elforgatása vagy megfordítása szintén jóval nagyobb és általánosabb adathalmazt eredményezhet. Fontos azonban, hogy **szem előtt tartsuk az adatok jellegét**: ha különböző színű tollak azonosításával foglalkozunk, nem biztos, hogy jó ötlet szürkeárnyalatos képekre váltani, ha pedig bizonyos gesztusokat szeretnénk osztályozni, a kép megfordítása lehet kedvezőtlen.

Az **alulillesztésre** nézzük meg a filmbevételek előrejelzésének példáját.

*Képzeld el, hogy több ezer filmről gyűjtöttél adatokat a műfajra, a színészekre és a bevételekre vonatkozóan. Majd ezek alapján betanítod a modellt, de rájössz, hogy bármilyen tesztadatot is választasz, a becslés az esetek többségében jócskán melléő.*

Vagyis a modelled alapvetően semmit sem tanult. Ez az **alulillesztés** példája, mivel vagy a tréningadatok nem tartalmaztak elegendő (vagy releváns) információt (jellemzőket) ahhoz, hogy a modelled bármi hasznosat megjósoljon, vagy a hibások a tanulás során beállított paraméterek.

Ebben a példában **további jellemzők** bevonása, mint például a megjelenés éve, a költségvetés és annak felosztása, vagy a reklámok mennyisége megoldhatná a problémát, és pontosabb előrejelzéshez vezethet. **Alulillesztés** akkor is előfordul, ha egyszerűen nincs elég adat, és ezért túl kevés információ áll rendelkezésre, amely felhasználható az előrejelzés vagy az osztályozás során.

Az alul- és túlillesztés egyaránt szembeűnő, és általában könnyű felismerni (de nem feltétlenül könnyű kijavítani...). A továbbiakban a **torzítások** árnyaltabb problémájáról fogunk beszélni.

## **Előítéletek és méltányosság**

(15-19.dia)

Míg az **alul-** és **túlillesztés** technikai és egyértelműen mérhető problémák, a **torzításokat** sokkal nehezebb felismerni. Kezdjük egy valós példával!

*2014-ben az Amazon, sok más nagyvállalathoz hasonlóan, megpróbálta automatizálni a munkaerő-felvételi folyamatot egy mesterséges intelligencia*

*létrehozásával, amely megvizsgálja az álláspályázatokat, és megtalálja a legtehetségesebb/legalkalmasabb munkavállalókat. Egy év fejlesztés után azonban rájöttek egy nagy hibára: az algoritmusuk diszkriminálta a nőket! Kiderült, hogy bár ebben nem volt rossz szándék, a korábban felvett munkavállalókról szóló tréningadatok rendkívüli módon torzítottak a férfiak javára, mivel akkoriban a legtöbb alkalmazott férfi volt. Bár ezt a problémát a tanításhoz használt program gondos megválasztásával sikerült orvosolni, nem lehetett garantálni, hogy az algoritmusnak nincsenek más, jelenleg nem látható torzításai, így a projektet 2018-ban leállították.<sup>1</sup>*

*Egy másik példa a Microsoft chatbotja, Tay2, amely a felhasználók uszító és rasszista nyelvezetét sajátította el a Twitteren, és amelyet mindössze 16 óra elteltével le kellett kapcsolni.<sup>2</sup>*

Elvileg lehetséges olyan modelleket létrehozni, amelyek méltányosak, de a témától függően ez **nagyon nehéz** is lehet. A probléma kiterjedt, kezdve attól, hogy a szolgáltatók chatbotjai nem ismerik fel egyes kisebbségek dialektusát, egészen odáig, hogy az osztályozó algoritmusok a korlátozott vagy egyeseket kizáró tréningadatok miatt nem ismerik fel a színesbőrű embereket emberként. Ezek a problémák nem csak technikai, hanem etikai jellegűek is, mivel felvetik a kérdést, hogy milyen döntéseket bízunk a gépekre. Az **etika modul** ezt a kérdést vizsgálja meg részletesebben, de jó forrás ez a videó is.

Az előítéletesség és a méltányosság jelenleg széles körben kutatott terület, a felmerülő nehézségekre pedig nincs könnyű megoldás. Mindössze annyit tehetünk, hogy óvatosak vagyunk, összeállítjuk a modellünk tanításához használt tréningadatokat. Ezt tükrözi az informatikában elterjedt mondás is: "az algoritmus csak annyira jó, mint az adatok", vagy egyszerűbben fogalmazva: "szemét be, szemét ki" [angolul: garbage in, garbage out].

## **A tanítás ideje és az áttanítás (transfer learning)**

**(20-23.dia)**

Amint azt a fenti témák során körbejártuk, a modell minősége nagymértékben függ a kiválasztott adatoktól és ebből fakadóan az **adatok méretétől** is. Egy olyan torzításmentes képosztályozó modell létrehozásához, amely különböző helyzetekben is pontosan észleli a tárgyakat, több ezer vagy akár millió képet kell felhasználni, ami a modell tanításához szükséges időt órákról hetekre is megnövelheti.

Ezen probléma ellensúlyozására a **áttanítás (angolul transfer learning)**<sup>3</sup> alkalmazható, amivel a már korábban jól tanított (**előre betanított**) modellek tapasztalatanyaga felhasználható. Ezek a modellek többnyire általánosabb feladatokra összpontosítanak, mint például a képeken található jellemzők megtalálása, amelyek aztán vagy a tréning **kiindulópontjaként**, vagy a bemeneti adatok összetettségét csökkentő **előfeldolgozáshoz** használhatók fel. Az első esetben egy képosztályozásra már felhasznált (mély neurális) hálózatot alapként lehet felhasználni egy olyan hálózat betanításához, amely egy másik képhalmaz osztályozására szolgál (például az erdőtüzek műholdas képeken való felismerésétől a röntgenfelvételeken található törések azonosításáig). A második esetben első lépésként egy olyan hálózatot lehet használni, amely már képes adott jellemzők - például élek vagy geometriai minták - felismerésére, így az új hálózat tanítását a nyers képadatok helyett ezen jellemzők ismeretében lehet megkezdeni. Ezekkel nemcsak a tanítás ideje csökkenthető drasztikusan, hanem az eredmény minősége is növelhető, különösen akkor, ha nagyon kisszámú mintával, mondjuk 50 képpel dolgozunk.

A következő gyakorlat során ilyen előre betanított hálózatokat használunk, mely révén néhány perc alatt, a böngészőnkön keresztül taníthatunk be képosztályozó algoritmusokat úgy, hogy néhány száz kép már elegendő lesz a jó eredményhez.

## Tananyag - magyar nyelven

-  SL - Reliance on data.pdf

## Hivatkozások

1. <https://www.theguardian.com/technology/2018/oct/10/amazon-hiring-ai-gender-bias-recruiting-engine>
2. <https://www.theguardian.com/technology/2016/mar/24/tay-microsofts-ai-chatbot-gets-a-crash-course-in-racism-from-twitter>
3. <https://machinelearningmastery.com/transfer-learning-for-deep-learning/>

# Játékvezérlő

Ebben a gyakorlatban a tanulók egy valós képosztályozó modellt fognak betanítani különböző irányjelző utasítások felismerésére. A modellt ezután a Snake nevű játék irányítására fogják használni.

A gyakorlat célja, hogy a diákok betaníthassák saját, valós idejű képosztályozó modelljüket. A folyamat

során számos, korábban tárgyalt probléma fog felmerülni, és a tanulóknak meg kell próbálniuk a modelljüket a lehető legjobbá tenni. A valós használat során jelentkező nehézségek demonstrálása érdekében egy ponton érdemes a felhasznált kamera irányát/helyzetét megváltoztatni, így kiemelve a tanítóhalmazok révén jelentkező problémákat (pl. háttérben lévő objektumok). A feladat végén a tanulóknak lesz egy működő modelljük, amely irányokat jelöl ki, és amelyet a Snake játék vezérlésére használhatnak (vagy a már haladó tanulók a TensorFlow felhasználásával bármilyen más alkalmazásban is).

Az ehhez a feladathoz szükséges idő nagyon eltérő lehet, a 30 perces rövid tesztől akár a több órán át tartó kísérletezésig, ha a diákok egy jól működő (általános) modellt szeretnének létrehozni.

## Exercises / Game Controller

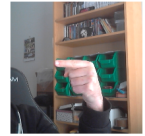
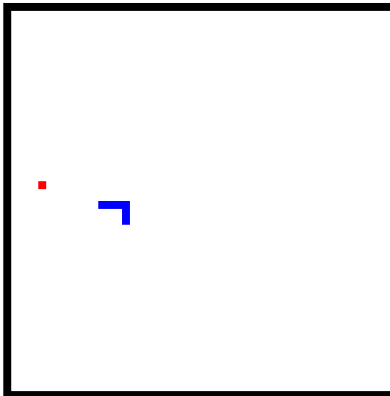
## Célok

A tanulók képesek lesznek:...

...egy valós képosztályozó **FT** modell betanítására

...alkalmazni a megfelelő adatok alkalmazásával kapcsolatos ismereteiket

...felismerni, hogy a **FT** egyik legfőbb nehézsége azt elérni, hogy a modell azt tegye, amit szeretnénk



```
up: 0.01
down: 0.00
left: 0.72
right: 0.25
none: 0.01
direction: left
```

Load an **image**-model created with TensorFlow.js

Your model has to include the class `tf.image.Model`

[load image-model from zip-file](#)



## Lehetőségek és korlátok

Most, hogy a diákok már jól értik, hogyan működik a felügyelt tanulás, és mik a főbb nehézségek, itt az ideje összefoglalni a tanultakat, és vetni egy pillantást a valós alkalmazásokra. Bár a különböző gépi tanulási módszerek közötti határvonal manapság kissé elmosódott (egy klasszikus felügyelt tanulási modell is továbbfejleszthető megerősítéses tanulással), a fejezetben az **FT** különböző alkalmazásokban való használatának azonosításáról és ennek korlátairól szól.


A fejezetet egy kvíz formájában tekintjük át, amelyben javarészt igaz vagy **hamis** kérdések szerepelnek, **diákon**. A tanulók személyes jelenlét mellett, kézjelek vagy színes/számozott kártyák segítségével is tudnak válaszolni. A kvíznek nem csak a helyes válaszokról kell szólnia, hanem a problémák **megértéséről**, illetve annak **megvitatásáról** is, hogy miért lehetnek esetleg más válaszok is helyesek. Ezért nem csak a válaszok megmutatása célszerű, hanem az is, hogy röviden beszéljünk róluk a tanulókkal. Néhány kérdés korábban még nem tárgyalt témákhoz kapcsolódik, hogy a diákok megmutathassák, hogy milyen jól tudják megszerzett tudásukat átültetni más, ismeretlen területekre.

## Célok

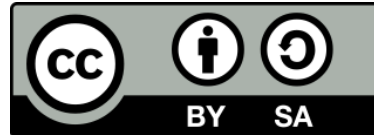
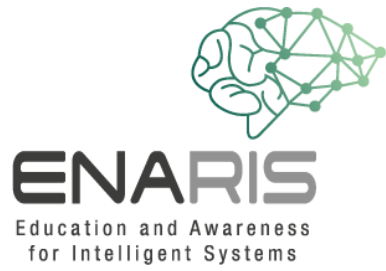
A tanulók képesek lesznek:...

- ...elmagyarázni mire jó és mire nem jó a felügyelt tanulás
- ...felismerni, hogy mely alkalmazások hasznosítják a felügyelt tanulást

## Tananyag – magyar nyelven

-  SL - Quiz.pdf





EUROPEAN UNION

