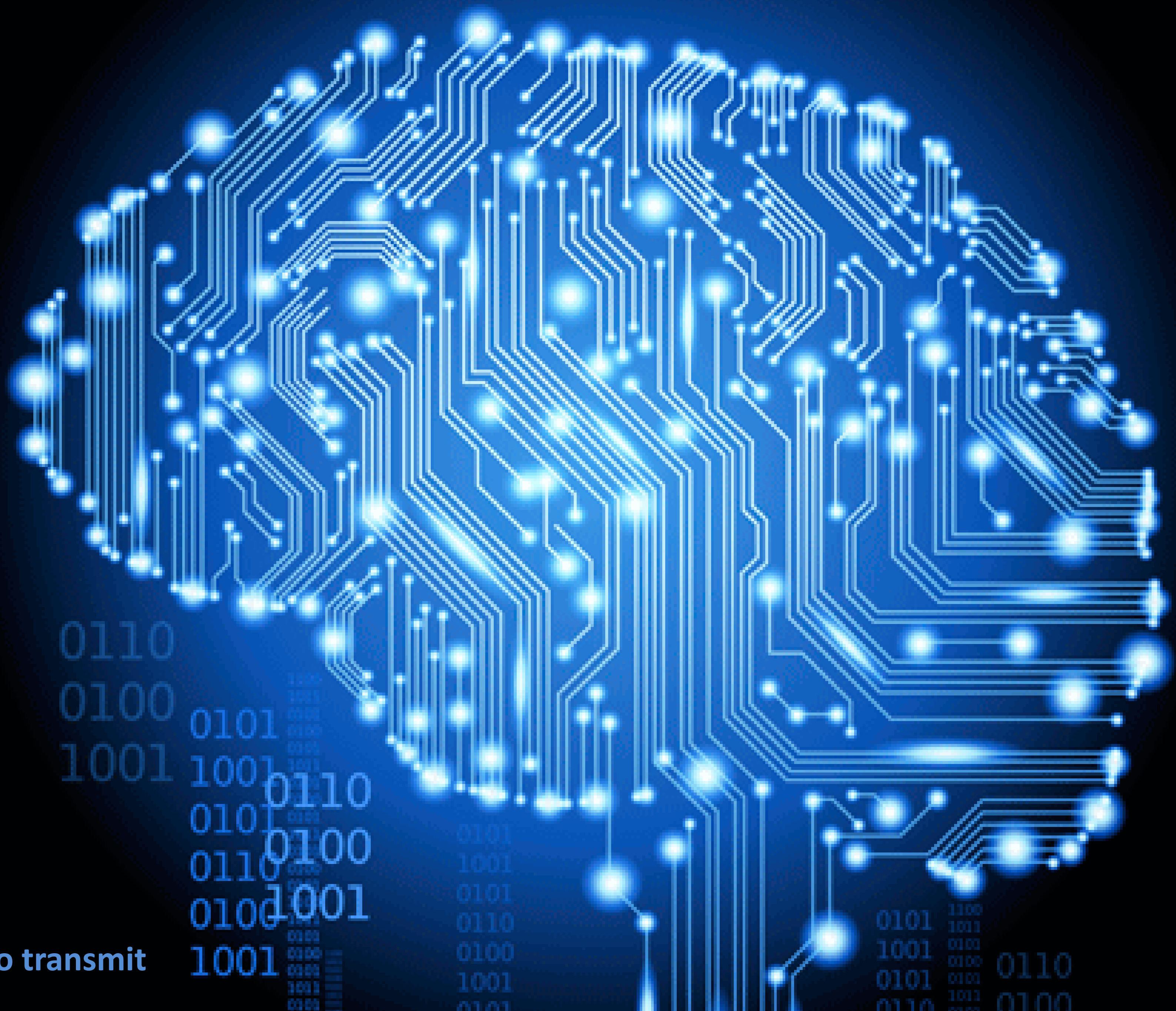


>ready to transmit

A neurális hálózatok tanításának alapjai II.: Módszerek a túltanulás elkerülésére

Szoldán Péter

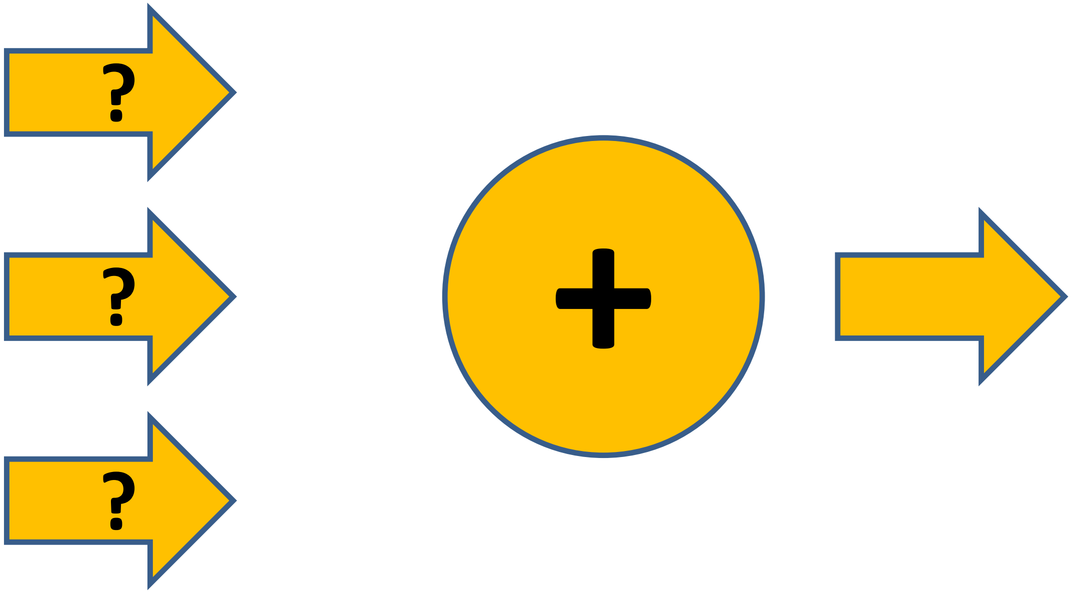
A hálózatnak nincs kontextusa

- Képzeljék el, hogy amióta megszülettek, semmit mást nem csináltak, csak egy székben ültek, és kutya vagy macska képeket néztek
- Ez az egész világuk, semmi mást nem tudnak semmiről
- Így kell elképzelni egy neurális hálózatot: csak arról van fogalma, amit mutattunk neki

Mit tud a gép?

- Csak azt, amit megtanítottak neki, amit látott
- A tudása a súlyokban tárolódik, amivel az előző neuronok kimenetét súlyozza, amit a tanulás során határoztunk meg
- A súlyokat kizárólag az befolyásolja, hogy a kitűzött feladatot mennyire jól tudja megoldani
- Ha olyasmit lát, amit még sose, nem tud rá mit mondani

Súlyok keresése: egyenletrendszer



Be

10

20

-30

Súlyok

Cél

50

$$10\boldsymbol{x} + 20\boldsymbol{y} - 30\boldsymbol{z} = 50$$

Ahány ismeretlen, annyi egyenlet kell, különben végtelen megoldás van

$10\boldsymbol{x} + 20\boldsymbol{y} - 30\boldsymbol{z} = 50$

$\boldsymbol{x}=5, \boldsymbol{y}=0, \boldsymbol{z}=0$

$\boldsymbol{x}=0, \boldsymbol{y}=2.5, \boldsymbol{z}=0$

$\boldsymbol{x}=0, \boldsymbol{y}=0, \boldsymbol{z}=-1.67$

$\boldsymbol{x}=2, \boldsymbol{y}=2, \boldsymbol{z}=0.33$

Egy ismeretlen, egy egyenlet kell

$$5\boldsymbol{x} = 10$$

$$\Downarrow$$

$$\boldsymbol{x} = 2$$

Két ismeretlen, két egyenlet kell

$$\begin{aligned} 5\boldsymbol{x} + 1\boldsymbol{y} &= 14 \\ \boldsymbol{x} + 2\boldsymbol{y} &= 10 \end{aligned}$$

$$\Downarrow$$

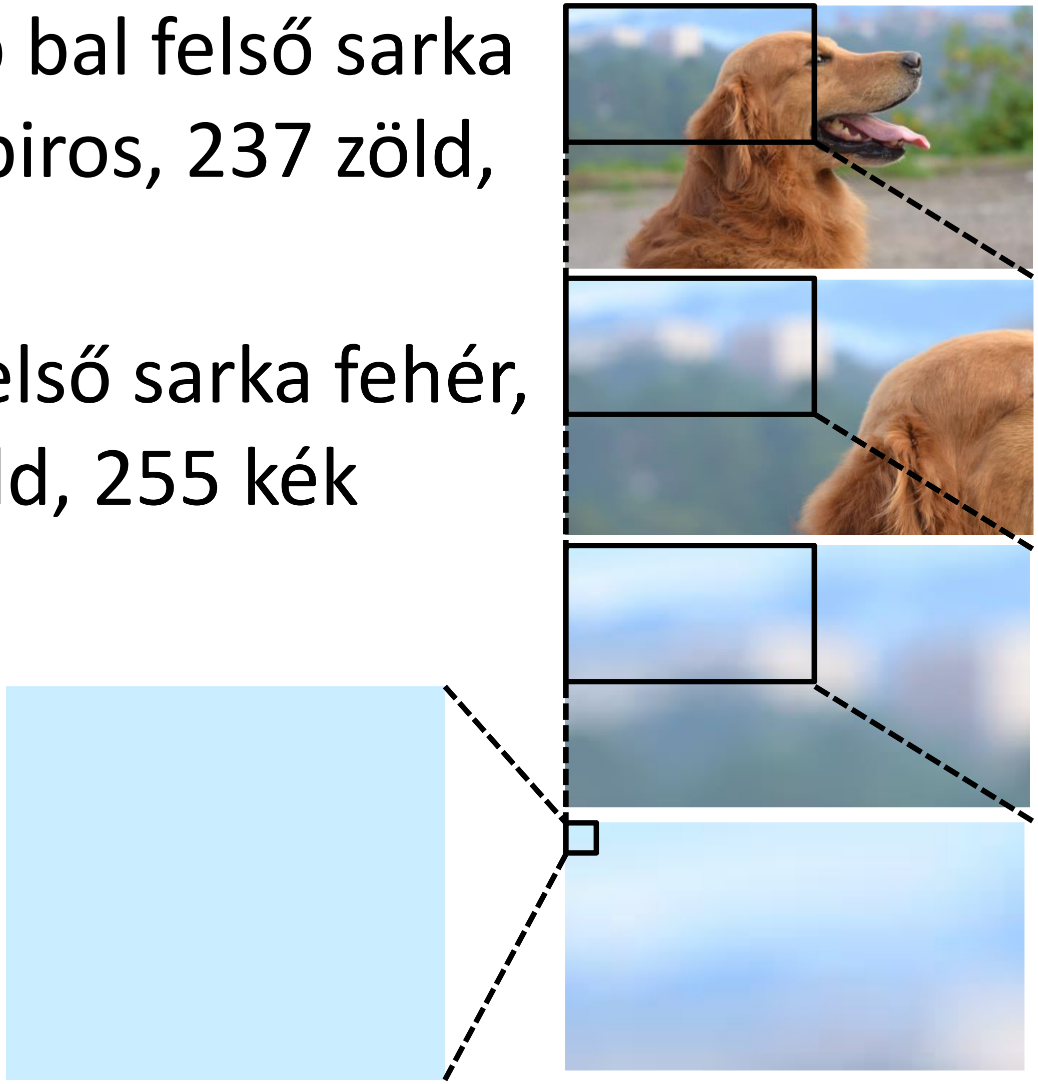
$$\boldsymbol{x} = 2,\ \boldsymbol{y} = 4$$

n ismeretlen, ***n*** egyenlet kell

- Különben az egyenletrendszer ***alulhatározott***, végtelen megoldása van
- A hálózatnak ez lehetőséget ad arra, hogy „lusta” módon a matematikailag legegyszerűbb megoldást keresse meg
- Ha kevés az adat (az egyenlet), ez oda vezethet, hogy valami számunkra nyilvánvalóan értelmetlen dolgot csinál
- Csak akkor csinálja meg, amit akarunk, ha muszáj neki, mert az adatok kényszerítik

Első pixel alapján lehet dönteni?

- Például kutya kép bal felső sarka halványkék, 202 piros, 237 zöld, 255 kék
- Macska kép bal felső sarka fehér, 255 piros, 255 zöld, 255 kék

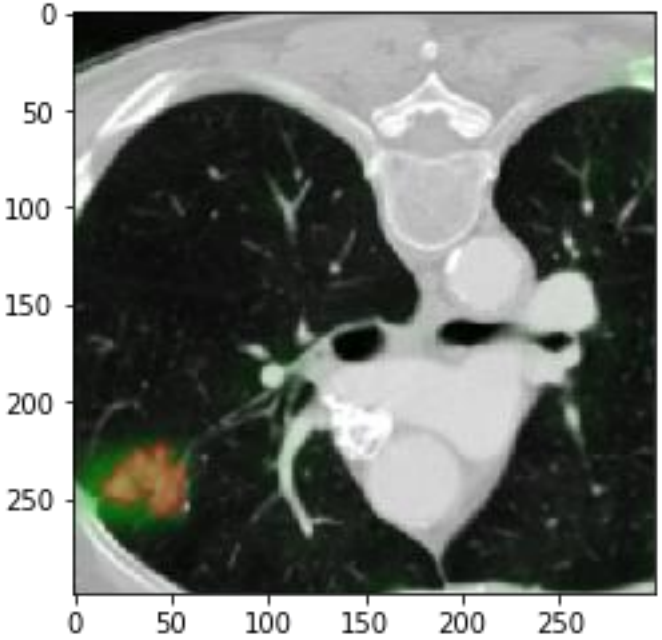


A lusta hálózat, azaz „túltanulás”

- Ha csak ezt a két képet mutatjuk, akkor:
- Az első pixelhez tartozó neuron hamar „rájön”, hogy ha kék, akkor mondja azt, hogy kutya, ha fehér, macska
- A hálózat többi része erre fog hallgatni, hiszen ez a neuron tökéletes választ ad, ő az „Andi”
- Így a hálózat megtanulja: ha halványkék a bal felső sarok, akkor kutya, ha fehér, akkor macska. Kész!

Radiológiai példa túltanulásra

37 éves, nő, nem rákos



75 éves, férfi, rákos

- Nő -> nem rákos, férfi -> rákos?
- 50 évnél fiatalabb -> nem rákos, 50 évnél öregebb -> rákos?

Túltanulás lehet bonyolult is

- A neuronok súlyaikkal komplexebb szabályokat is tudnak alkotni
- Lehet például olyasmi, hogy az első pixel kék, a tizenhetedik piros, a kétszázadik az ötödik sorban meg zöld
- A lényeg, hogy megfognak valamit, ami csak konkrétan az adott tanuló halmazra igaz, általában nem

A túltanulás (overfitting)

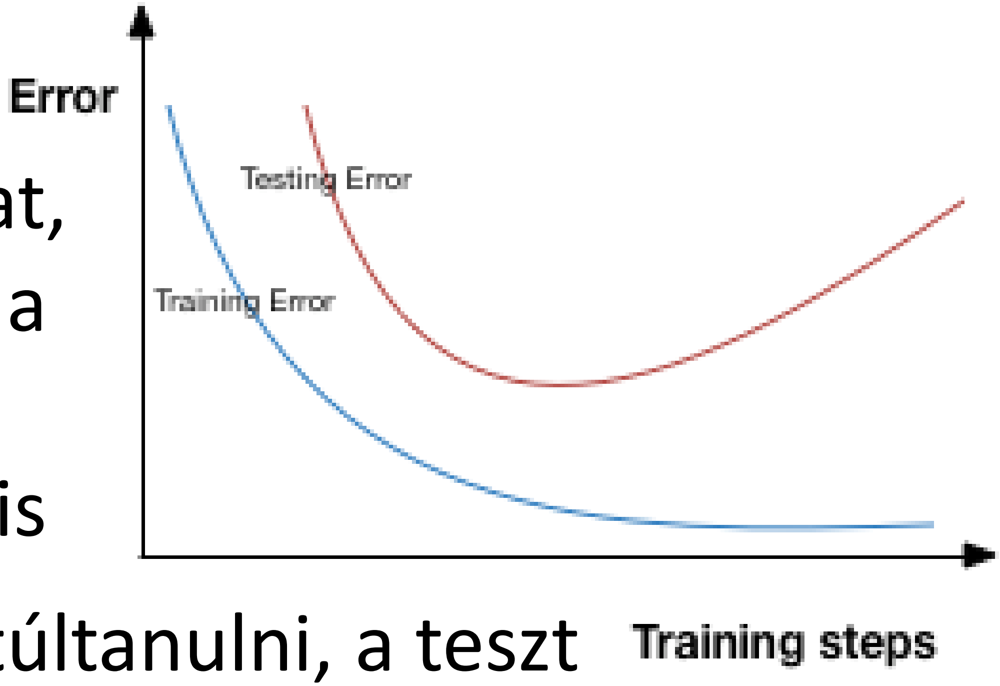
- Túltanulásnak nevezzük, amikor a hálózat az elvárt általánosítás helyett az adott adathalmaz valami egyébként jelentéktelen sajátosságát kihasználva tanulja meg megjósolni az elvárt értékeket, „bemagolja” a tanulóhalmazt
- Emiatt a túltanult hálózat általánosító képessége csökken, újabb tesztadatokra nem fog jó eredményeket produkálni

Teszthalmaz

- A hálózatokat mindig teszthalmazon értékeljük
- A teszthalmaz nem tartalmazhat a tanulóhalmazban levő képeket
- A teszthalmazon mindig rosszabbul teljesít a hálózat, de ha jelentősen, akkor túltanulásról beszélünk
- A gyakorlatban a rendelkezésre álló összes adat véletlenszerűen kiválasztott kb. 10%-át szokás félretenni teszthalmaznak

Túltanulás grafikonon

- Ahogy elkezd tanulni a hálózat, egyre jobb lesz a tanuló- és teszthalmazon is



- Amikor elkezd túltanulni, a teszt hiba növekszik, pedig a tanulóhalmazon a hiba csökken: kék vagy fehér pixel szerint kutya vagy macska nem általánosítható

További adatok lerontják a tévesen felismert összefüggéseket

- Bal felső pixel fehér egy újabb kutyánál, kék egy macskánál: fehér-kék szabály nem működik már!
- Valami mást kell kitalálnia
- Javítottuk a hálózat teljesítményét!

Legjobb: nagy tanulóhalmaz kell

- Minden egyes mintakép, vagy mintaadat, amit mutatunk a hálózatnak, egy-egy újabb egyenletnek felel meg
- Így ha például egy hálózatban 5 millió paraméter van, akkor hozzávetőlegesen 5 millió képet kellene neki mutatni, hogy jó legyen
- Ezt a google könnyen megteszi kutyás képekkel, de általában nagyon nehéz feladat

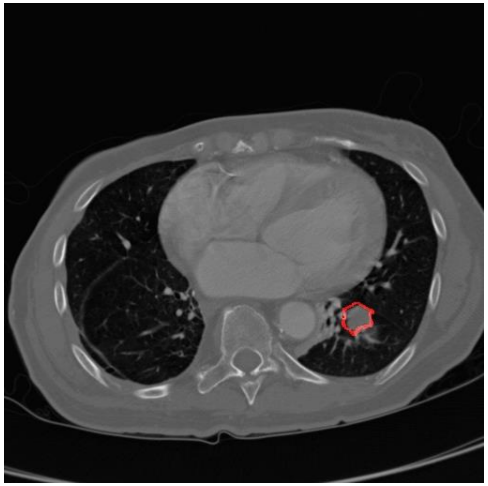
Adatjavítás (data augmentation) I.

- Véletlen eljárással csináljunk a meglevő adatokból újabbakat
- Forgatás

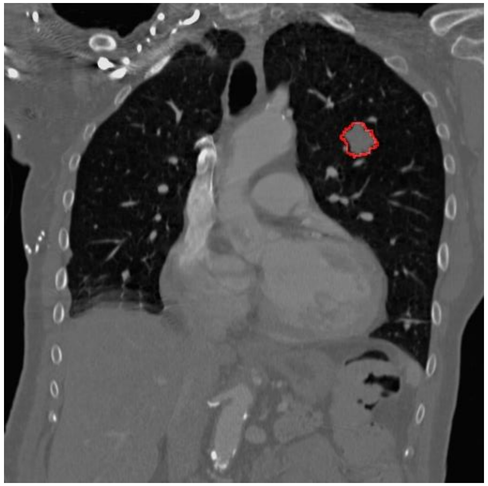
- Tükrözés

Adatjavítás (data augmentation) II.

- Háromdimenziós adatoknál a forgatás a dimenziók felcserélése is lehet, ha a tengely nem anatómiailag fix, mint például egy daganatnál



axialis

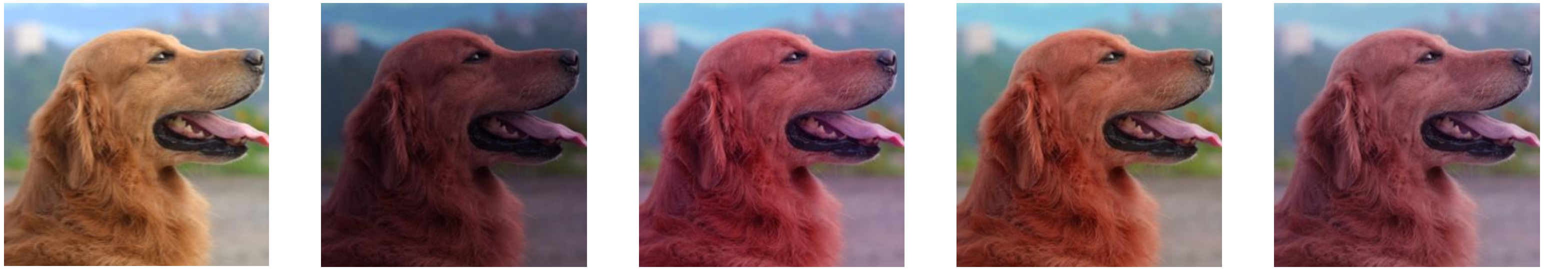


coronalis

sagittalis

Adatjavítás (data augmentation) III.

- Szín, telítettség, kontraszt, árnyalat



- Szélek levágása, átméretezés (mértékkel!)

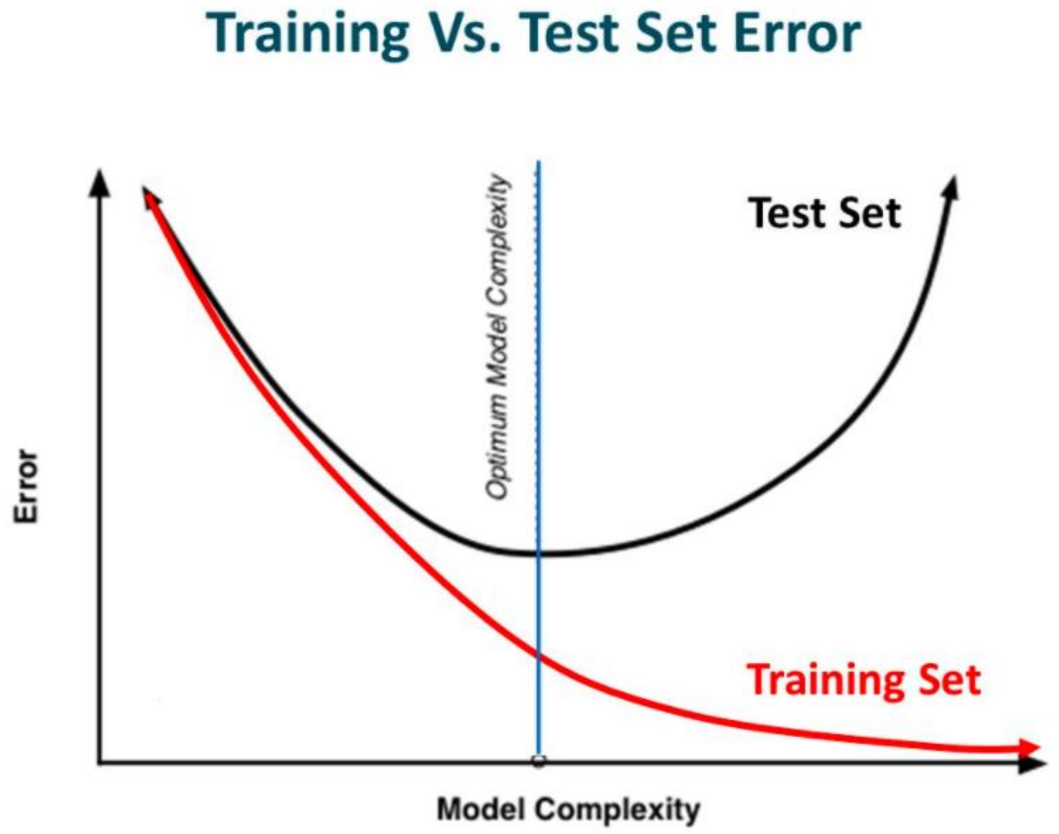
Adatjavítás (data augmentation) IV.

- Ha mindegyiket egyszerre alkalmazzuk, akkor sokszoros adatmennyiséget kapunk
- Például ***8*** féle tükrözés, ***32*** féle szín, ***32*** féle levágás ***8*** x ***32*** x ***32*** = ***8,192***-szeres adatmennyiséget jelent
- Ez csak elméleti, a javított adatok messze nem annyira jók, mint valódi új adatok, de azért sokat segítenek

Korai megállás (early stopping) I.

- Általában a túltanulás előtt a hálózat elkezd általánosabb összefüggéseket megtanulni, a véletlen inicializálás miatt
- A túltanulás, az apró jellegzetességekre figyelés csak később jön
- Ezért ha *jó pillanatban* megállítjuk a tanulást, az tud segíteni a túltanulás ellen

Korai megállás (early stopping) II.



- Ott kell megállítani, ahol a legjobb az eredménye a teszthalmazon

Korai megállás (early stopping) III.

- De mikor álljunk meg? Például minden 500 lépés után megnézzük, mi a teljesítmény a teszthalmazon, és megállunk, ha elkezd csökkenni
- Sajnos ez lassítja a tanulást, sok idő a teszthalmazon állandóan ellenőrizgetni
- Másik baj, hogy túltanulhatja a teszthalmazt is, mert pont ott állunk meg, ahol a teszthalmazra pont jó: kell egy harmadik halmaz is (tanuló, ellenőrző, teszt)

Súlycsökkentés (weight decay) I.

- Az ilyesfajta megoldások, ahol egy vagy néhány érték magas, a többi alacsony, általában túltanulást jeleznek: „semmi sem érdekel, csak hogy kék-e a bal felső pixel!”

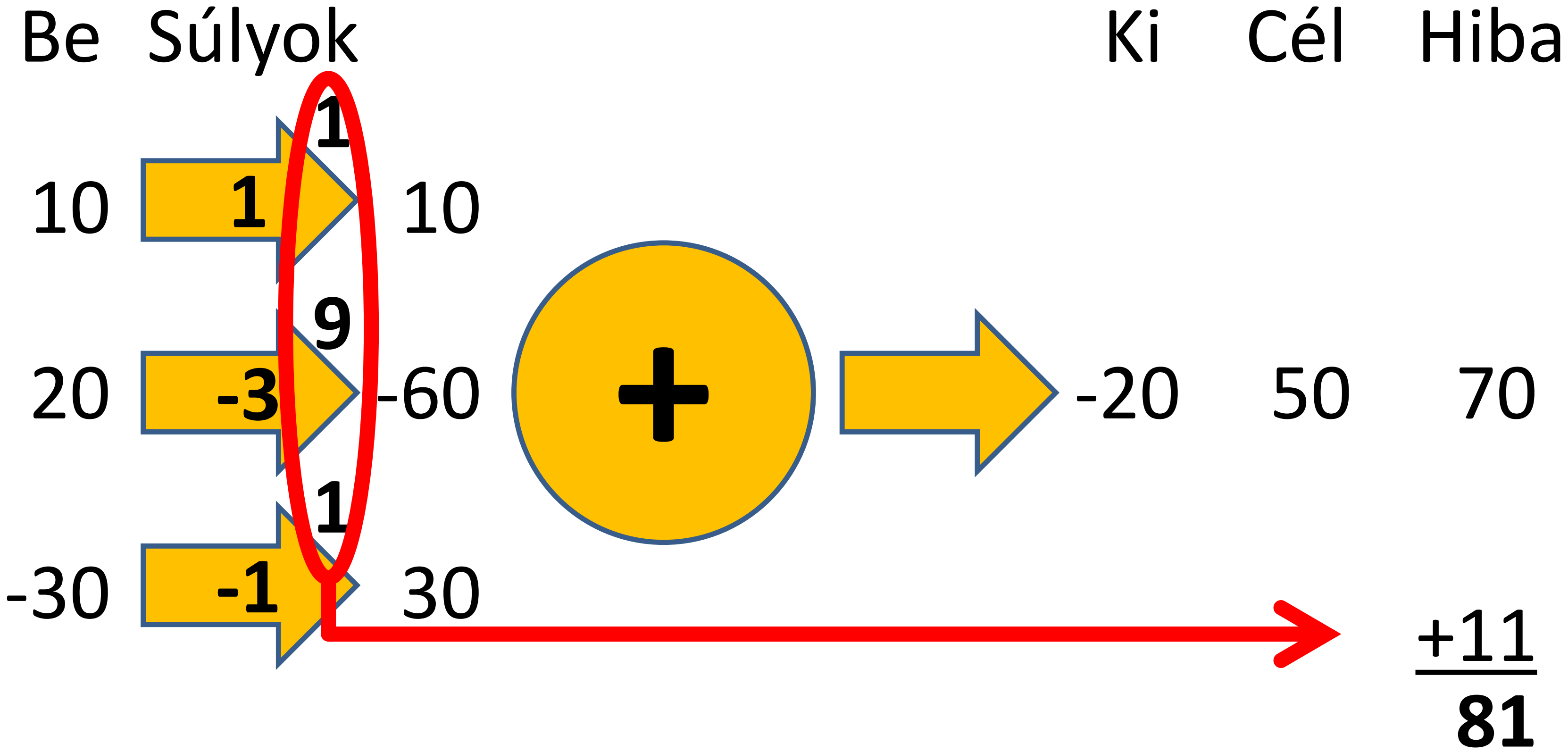
$$\boldsymbol{x}=5,\ \boldsymbol{y}=0,\ \boldsymbol{z}=0$$

$$\boldsymbol{x}=0,\ \boldsymbol{y}=2.5,\ \boldsymbol{z}=0$$

$$\boldsymbol{x}=0,\ \boldsymbol{y}=0,\ \boldsymbol{z}=-1.67$$

- Próbáljuk „egyenletesebbé” tenni!

Súlycsökkentés (weight decay) II.



Be Súlyok Ki Cél Hiba

10 1 1 10

20 -3 9 -60

-30 -1 1 30

+

-20 50 70

+11

81

Hibához adjuk hozzá a súlyok négyzetét!

Ez „bünteti” az egyenlőtlen súlyeloszlást.

Súlycsökkentés (weight decay) III.

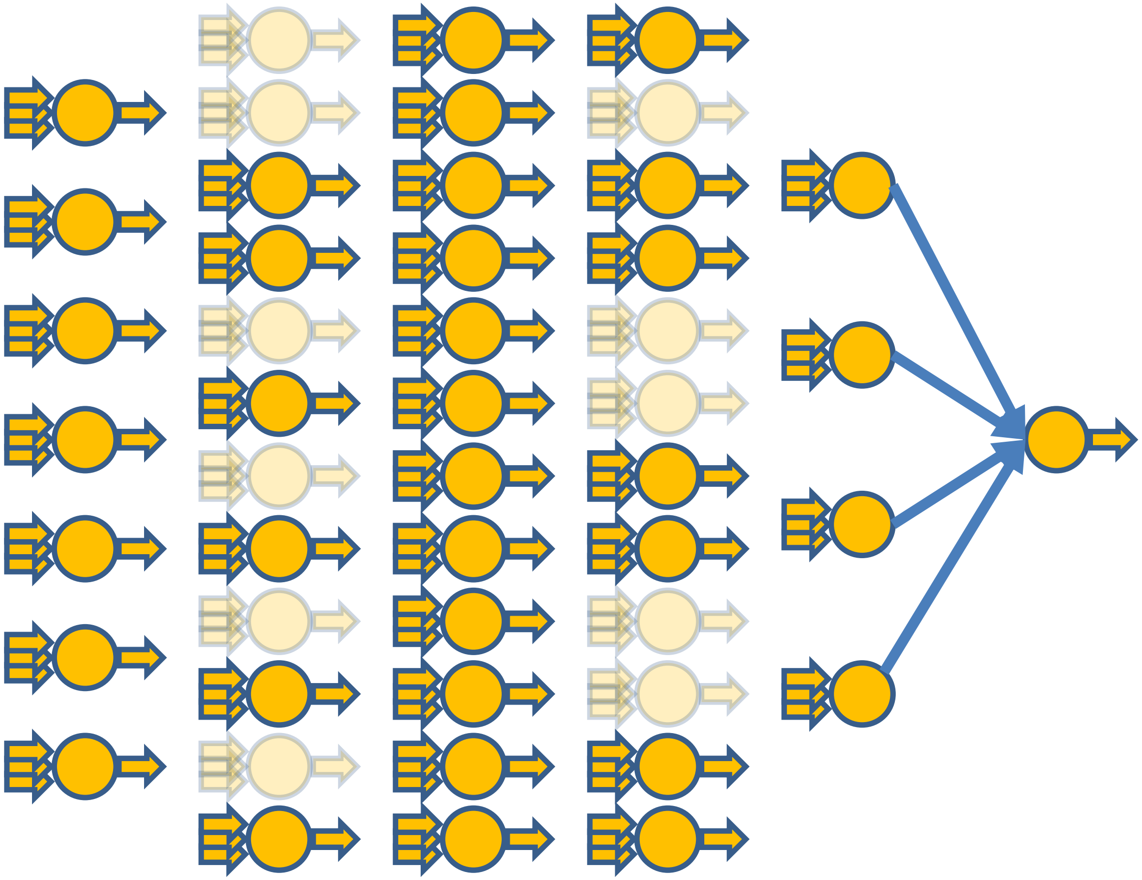
- Négyzetösszeg erősen regularizál, „***L2 regularizáció***”
- Például 3, 0, 0 négyzetösszege 9
- 1, 1, 1 négyzetösszege 3

Matematika iránt érdeklődőknek: a négyzetes közép mindig nagyobb a számtaninál, a különbség annál nagyobb, minél távolabb vannak egymástól a számok
Négyzetes közép: sqrt(($\boldsymbol{a}^2$ + $\boldsymbol{b}^2$ + $\boldsymbol{c}^2$) / 3)
Számtani közép („átlag”) : (***a*** + ***b*** + ***c***) / 3

Súlycsökkentés (weight decay) IV.

- Mivel a súlyok négyzete nagyon erős hatást gyakorol, ezért
- ezt az úgynevezett ***regularizációs tényezőt*** csak valamilyen kicsi együtthatóval (**λ**) szabad a hibához hozzáadni, például **λ =** 0.0001 egy gyakori érték
- Különben arra készteti a hálózatot, hogy minden súly nullává váljon („modell összeomlás”, „model collapse”)

Kihagyás (dropout) I.



- Bizonyos rétegeknél véletlenszerűen kihagyunk a számolásból neuronokat

Kihagyás (dropout) II.

- Az indoka ennek az, hogy általában a túltanulás jellemzője, hogy nagyon egyedi dolgokra figyel oda
- Ha véletlenszerűen kimarad egy-egy neuron, akkor a következő réteg kénytelen általánosabb összefüggéseket keresni, tanulni
- Minden tanulási lépésben más-más neuronokat hagyunk ki, mindig újra generáljuk a véletlen számokat

Kihagyás (dropout) III.

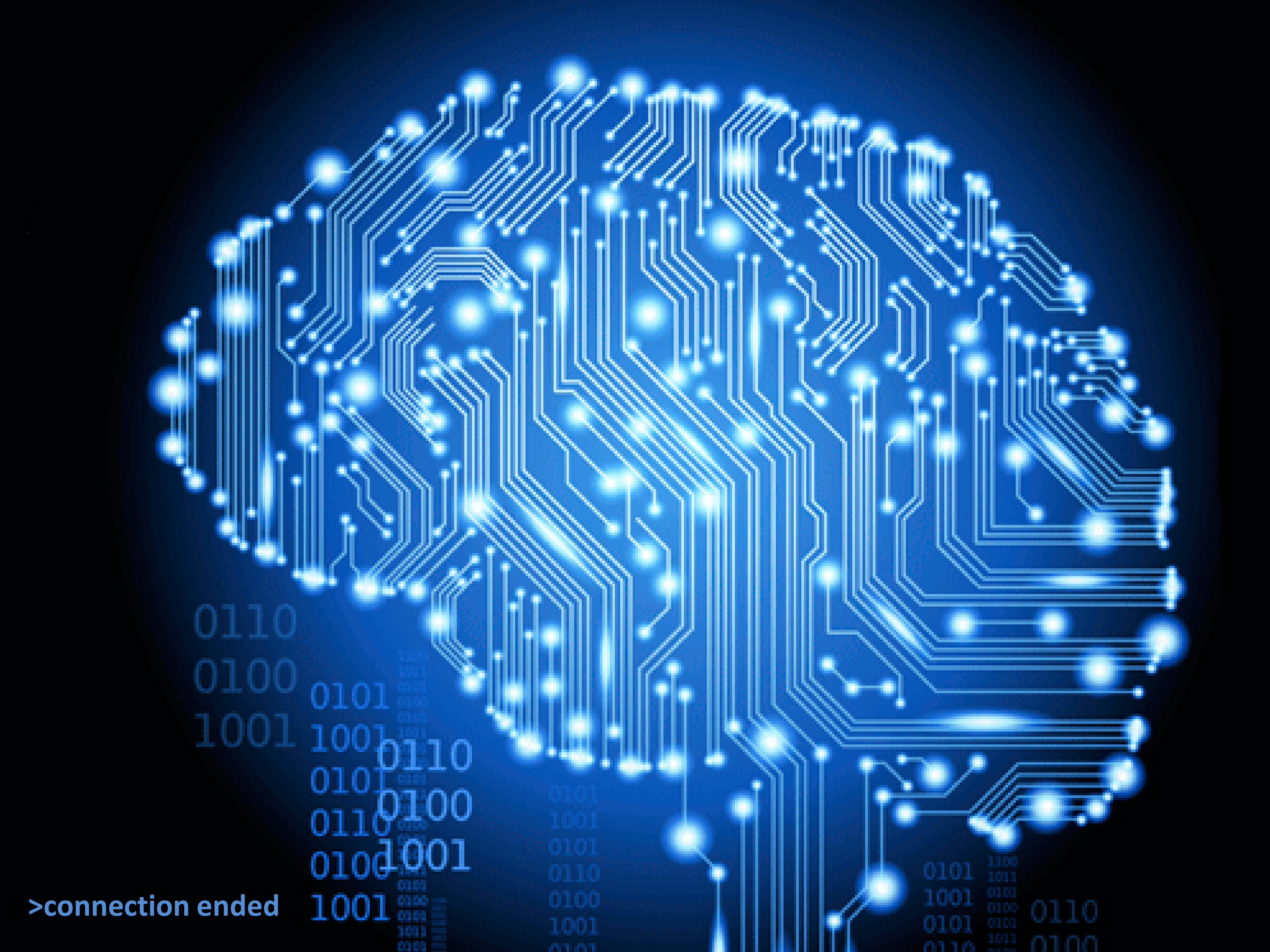
- Analóg azzal, amikor rossz a vétel a TV-n, de még látszik valami: az ember azért meg tudja mondani, hogy mi van ott nagyjából

Kihagyás (dropout) IV.

- Gyakorlati eredmények nagyon jók a kihagyással
- A változtatható paraméter az, hogy hány százalékát hagyjuk ki a neuronoknak? Például ***drop_rate = 0.2***, vagy ***keep_rate = 0.8***, akkor minden ötödiket
- A gyakorlatban a keep rate kb. ***0.5 – 0.8***

Várható eredmények

- Ebben a kontextusban értékeljék a teljesítményünket, hogy 187 pozitív (és 800 negatív) mellkas CT felvételből tudtunk kihozni 80.6% pontosságot tüdőrák diagnosztikára



>connection ended