Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Mesterséges Intelligencia és Rendszertervezés Tanszék

VIMIAC16 2025/26/I.

Neurális hálók –regularizáció

Előadó: Dr. Hullám Gábor

A mesterséges neurális háló tanításának kérdései

- Mekkora (hány réteg, rétegenként hány processzáló elem) hálózatot válasszunk?
- Hogyan válasszuk meg a tanulási tényező, α értékét?
- Milyen kezdeti súlyértékeket állítsunk be?
- Hogyan válasszuk meg a tanító és a tesztelő minta készletet?
- Hogyan használjuk fel a tanító pontokat, milyen gyakorisággal módosítsuk a hálózat súlyait?

- http://mialmanach.mit.bme.hu/neuralis/index

Mekkora hálózatot válasszunk?

- Mekkora hálózatot válasszunk?(hány réteg, rétegenként hány processzáló elem)

- Nincs egzakt számítás ennek meghatározásához.

1.) Kiindulás egy ‘nagyobb’ hálózatból, majd a hálózatban megmutatkozó redundancia csökkentése minimális értékre.

- pruning technikák (http://mialmanach.mit.bme.hu/neuralis/ch04s03)
- regularizáció

$$C_r(\mathbf{w}) = C(\mathbf{w}) + \lambda \sum_{i,j} |w_{ij}|$$

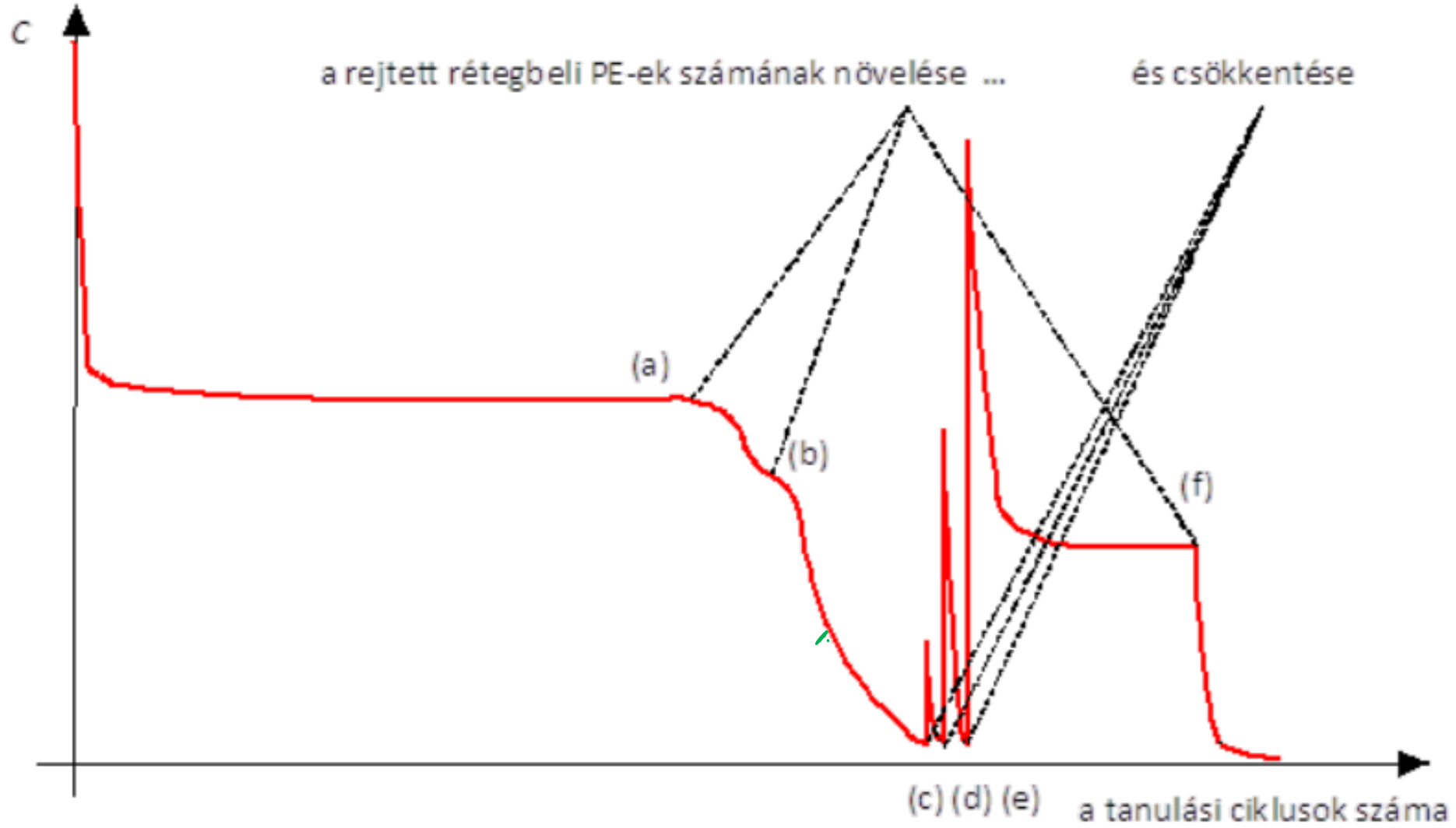
Mekkora hálózatot válasszunk?

- Mekkora hálózatot válasszunk?(hány réteg, rétegenként hány processzáló elem)

2.) Kiindulás egy ‘kisebb’ méretű hálózatból, majd fokozatos bővítéssel (processzáló elemekkel, ill. rétegekkel) vizsgáljuk meg tudja-e oldani a feladatot.

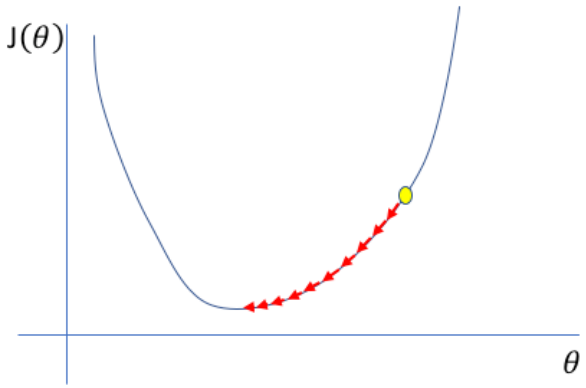
A két megközelítés nem feltétlenül vezet azonos eredményre!

Veszteség alakulása a tanítás folyamán

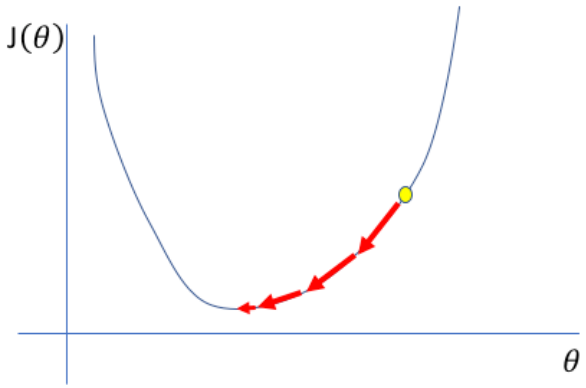


http://mialmanach.mit.bme.hu/neuralis/ch04s03

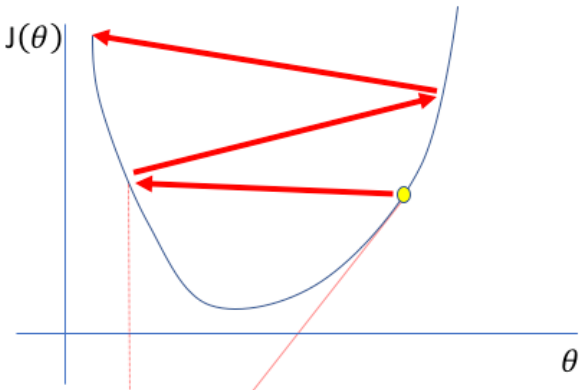
Hogyan válasszuk meg a tanulási tényező értékét?



A small learning rate requires many updates before reaching the minimum point

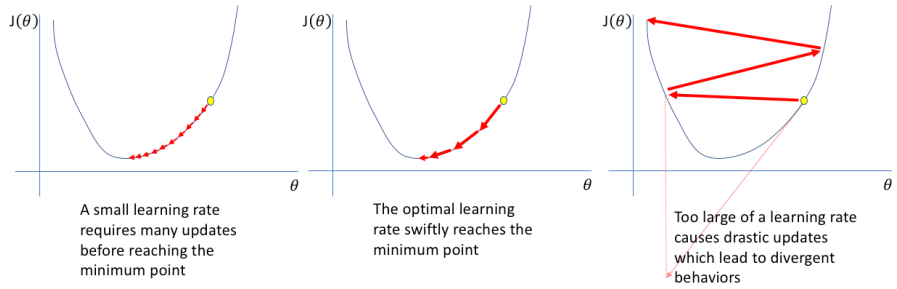


The optimal learning rate swiftly reaches the minimum point



Too large of a learning rate causes drastic updates which lead to divergent behaviors

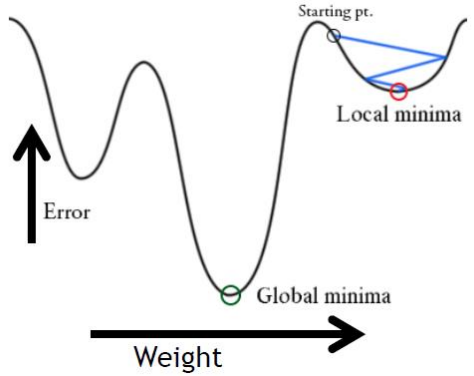
Tanítási tényező (learning rate) megválasztása



- nincs egyértelmű módszer α (=μ) meghatározására
- változó, lépésfüggő α alkalmazása
  - kezdeti nagy értékből kiindulva m lépésenként csökkentjük α -t.
- adaptív α alkalmazása
  - α változtatását attól tesszük függővé, hogy a súlymódosítások csökkentik-e a kritériumfüggvény értékét

Milyen kezdeti súlyértékeket állítsunk be?

- Nincs egyértelmű módszer
- A priori ismeret hiányában a véletlenszerű súlybeállítás is egy lehetőség
  - az egyes súlyokat egy egyenletes eloszlású valószínűségi változó különböző értékeire választhatjuk



- Szigmoid aktiváció függvénynél kerüljük el, hogy már a tanítás elején telítéses szakaszra kerüljön a rejtett rétegek kimenete
  - Minél nagyobb a hálózat, annál kisebb véletlen értékek választása célszerű

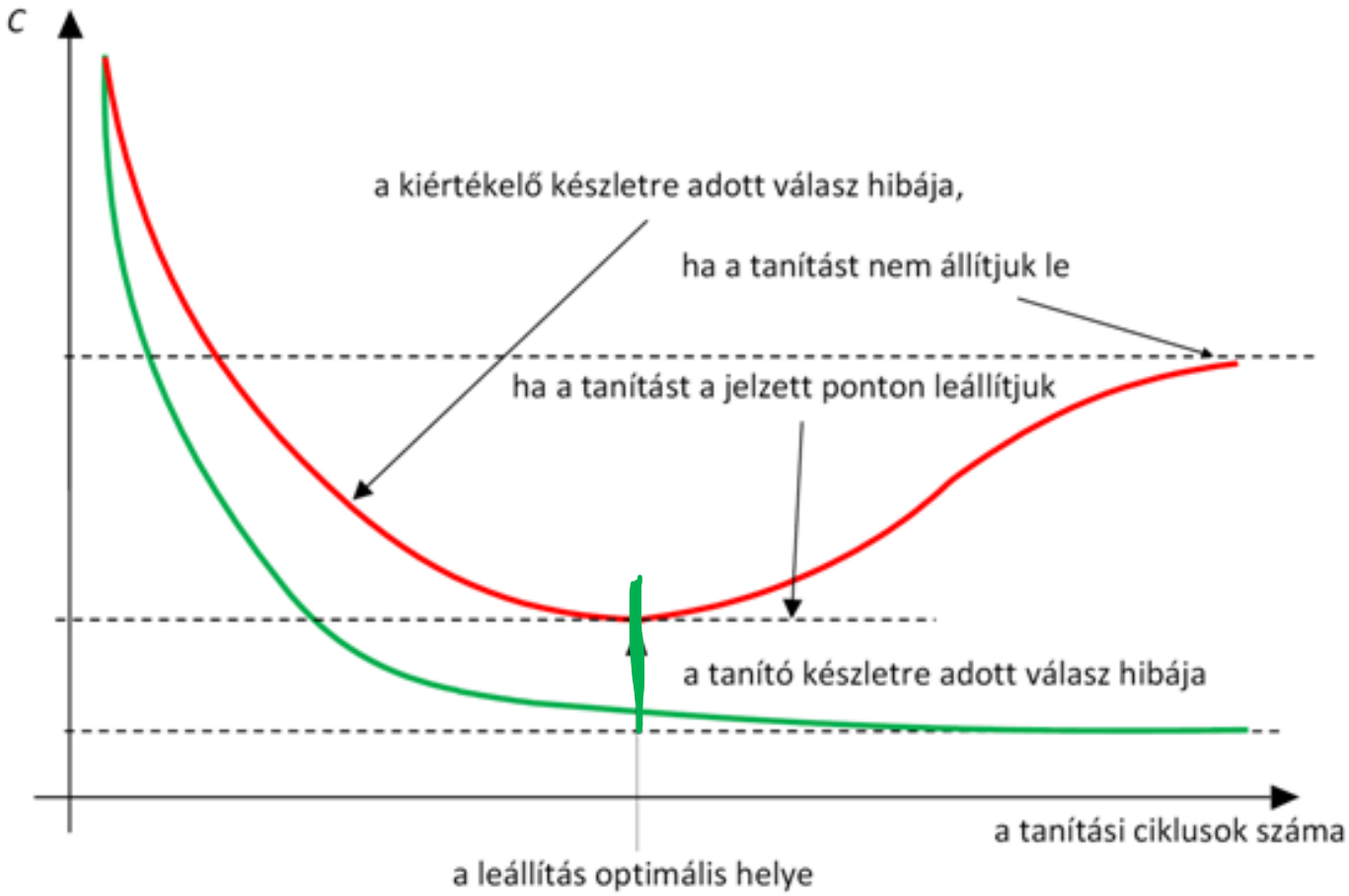
Hogyan válasszuk meg a tanító, kiértékelő és tesztelő minta készletet?

Neurális hálóknál:

- Tanító minta : felhasználjuk a tanításhoz
- Kiértékelő (validációs) minta: nem használjuk súlymódosításhoz, de ahhoz igen, hogy mikor álljon le a tanítás
- Tesztelő (minősítő) minta: nem használjuk a tanításhoz csak a minősítéshez

- http://mialmanach.mit.bme.hu/neuralis/ch04s03

Meddig tanítsuk a hálózatot?



Early stopping

Hogyan válasszuk meg a tanító, kiértékelő és tesztelő minta készletet?

- Nagy számú mintánál nem jelent problémát akár három egyenlő részre osztani az adathalmazt
- A legtöbb valós probléma esetén azonban „sosem elég” a minta
  - Léteznek elméleti és gyakorlati korlátok a szükséges tanítópontok számára a háló mérete és teszt készleten elvárt hiba alapján
    - Elméleti minimum mintaszám: Valószínűleg közelítően helyes tanulás (probably approximately correct – PAC learning)
      - Gyakorlatban kevésbé alkalmazható
    - Gyakorlati minimum mintaszám: korábbi tapasztalatok alapján
  - Kereszt-kiértékelési (cross validation) eljárások alkalmazása

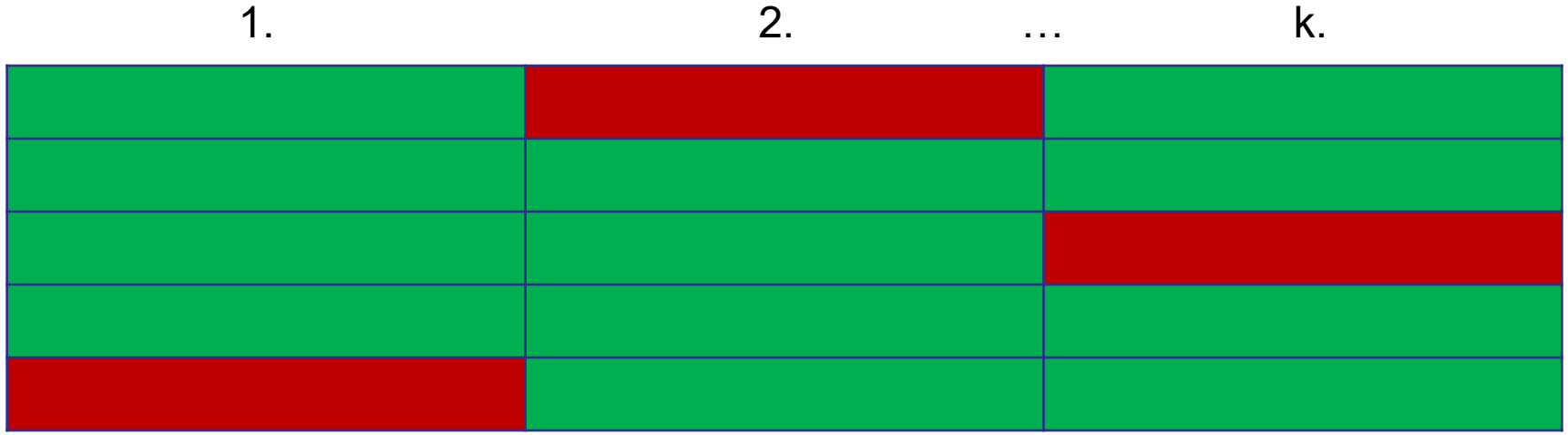
http://mialmanach.mit.bme.hu/neuralis/ch04s03

Adathalmaz particionálás

- Ökölszabály: Tanító – Validációs – Teszt : 80%-10%-10%
- Az adathalmazok particionálása többször is elvégezhető

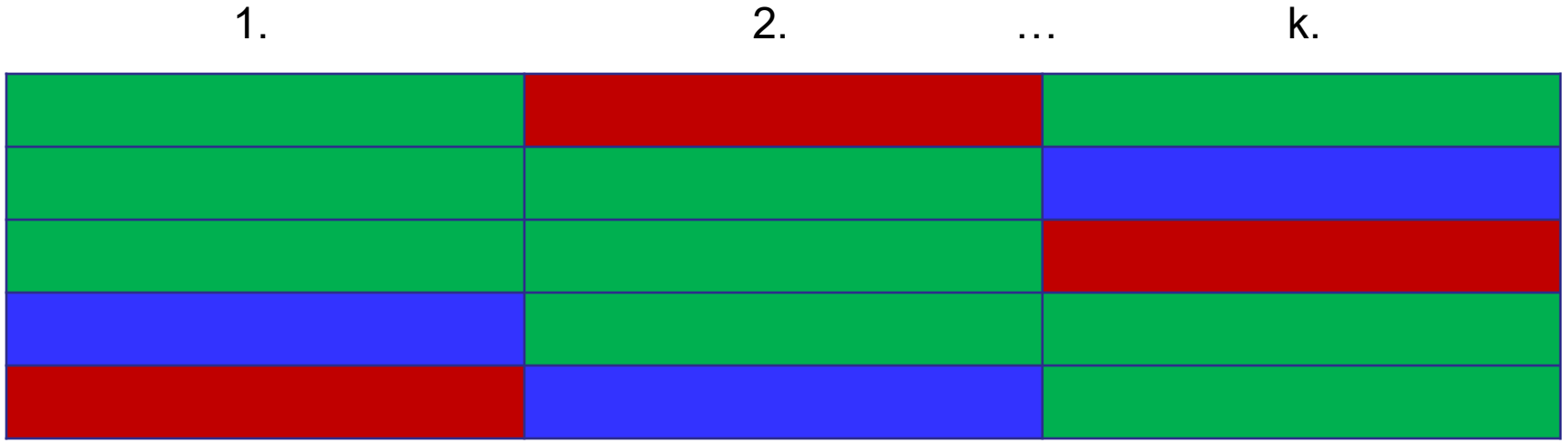
Keresztvalidációs módszerek (cross validation)

- K-keresztvalidáció : adathalmaz particionálása k darab részre
  - A tanításnál k-1 darab partíció lesz a tanító, a k-adik a teszt adathalmaz
  - K-szor ismételt tanítás



Adathalmaz particionálás

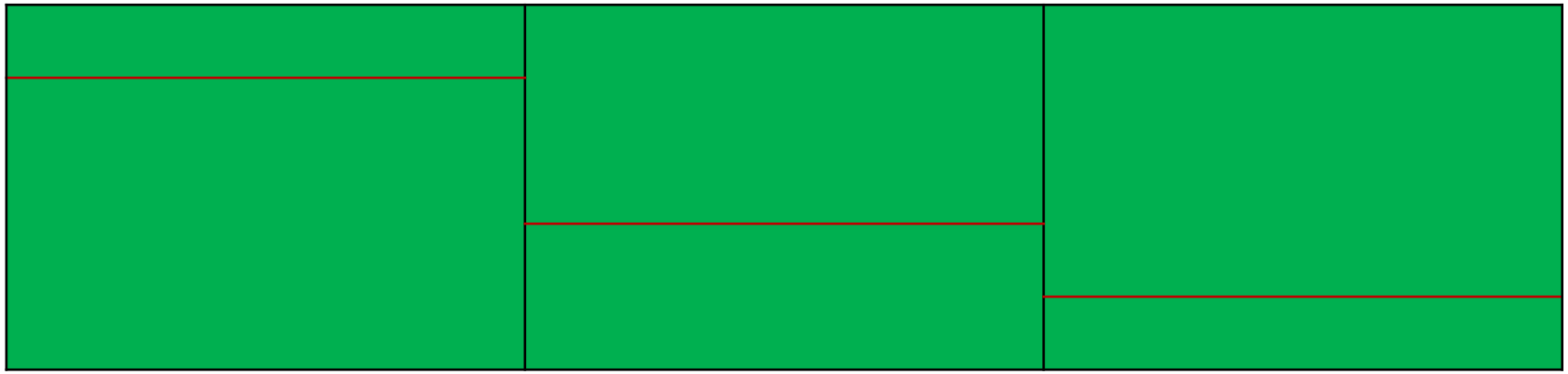
- K-keresztvalidáció : ha Tanító – Validációs – Teszt adathalmazra is szükség van
  - A tanításnál k-2 darab partíció lesz a tanító, az egyik megmaradó a validációs, a másik pedig a teszt adathalmaz
  - K-szor ismételt tanítás



Adathalmaz particionálás

- Leave-one-out keresztvalidáció
  - Mindig egyetlen minta a teszt, az összes többi tanító

1. 2. … n.



Keresztvalidált teljesítménymetrikák

- A teljesítménymetrikák lehetséges tartományát lehet vele vizsgálni
  - tartomány
  - átlag, medián, szórás
  - konfidencia intervallum (confidence interval: CI, pl. 95%)
- robusztusságot mér

k=3 esetén

- TPR Átlag: 0.84
- Szórás: 0.0374
- 95%-os konfidencia intervallum (Student-t): 0.84±0.0929
- [0.7471, 0.9329]

k=10 esetén

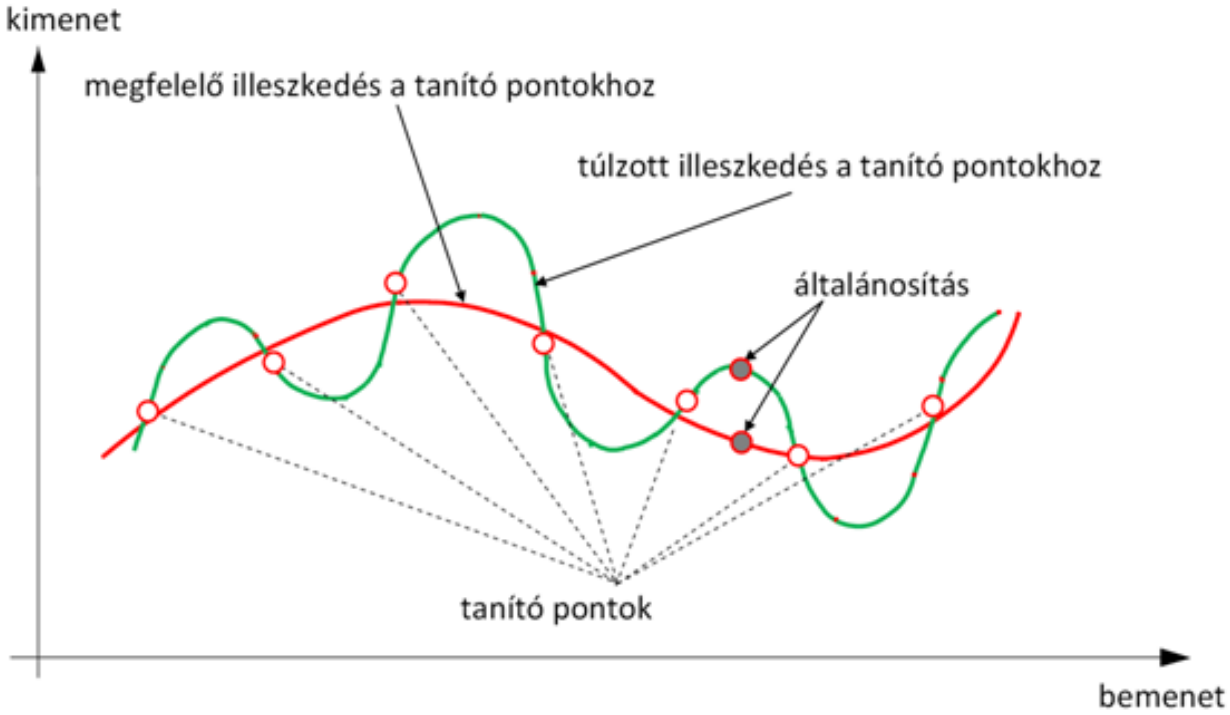
- 0.84±0.02677
- [0.8132, 0.8668]

Hogyan használjuk fel a tanító pontokat, milyen gyakorisággal módosítsuk a hálózat súlyait?

- Pontonként
  - Súlymódosítás tanítópontonként
- Kötegelt (batch) eljárás szerint
  - Súlymódosítás k db tanítópontonként vagy
  - Csak a teljes tanító készlet (epoch) felhasználása után

Hogyan védekezzünk a túltanulással szemben?

- Akkor lép fel (overfitting), ha a tanító adathalmaz mintáira már nagyon kis hibájú válaszokat kapunk, miközben a validációs adathalmazra egyre nagyobb hibával válaszol a hálózat.



Hogyan védekezzünk a túltanulással szemben?

- A hálózat válaszai túlzottan illeszkednek a véges számú tanító pont által megadott leképezéshez
- Ok: a hálózat mérete (szabadságfoka) a tanító pontok számához viszonyítva túl nagy

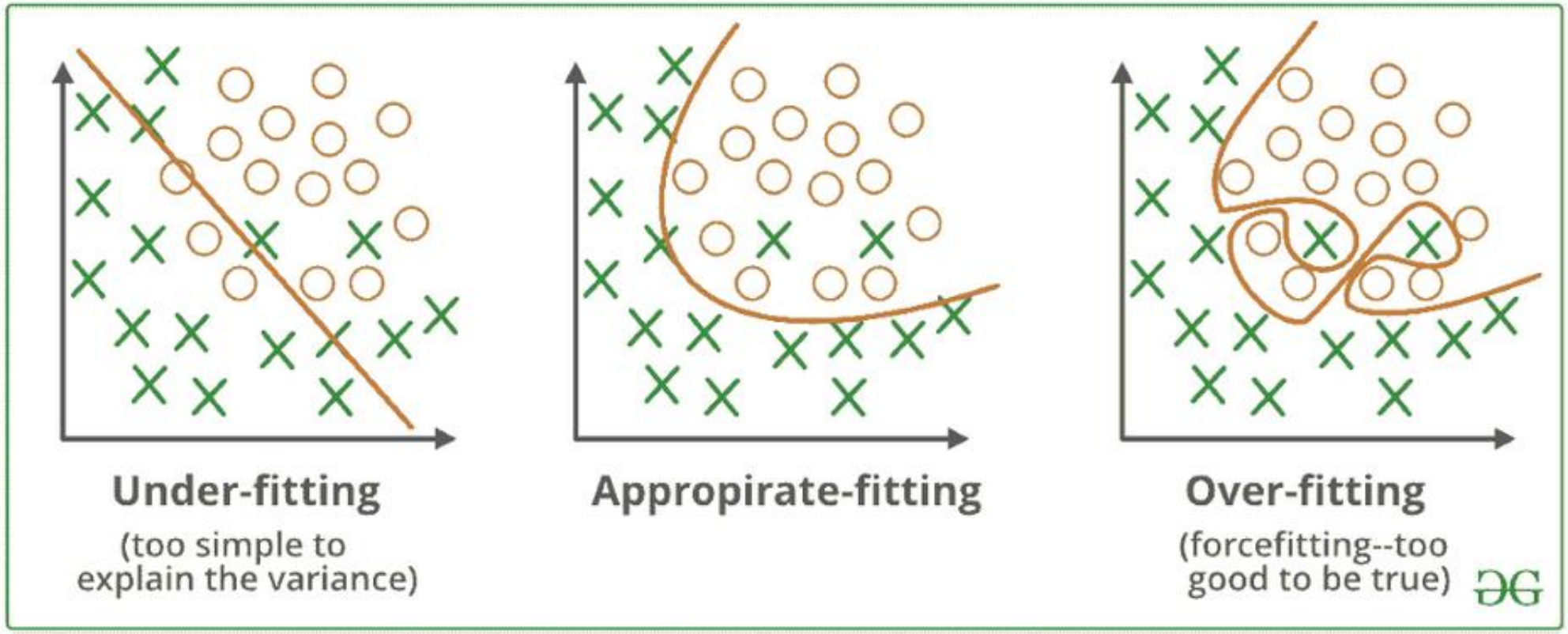
Miért kell védekezni ellene?

- Túltanulással romlik az általánosító képesség.

Védekezés:

- Regularizációs technikák
- Validációs adathalmaz – early stopping

Túlilleszkedés - alulilleszkedés



https://www.geeksforgeeks.org/regularization-in-machine-learning/

Bias – variancia kompromisszum

- A bias (torzítás/elfogultság) és a variancia kiegyensúlyozása
- A regularizálás segíthet egyensúlyozni a
  - modell torzítása (alulilleszkedés) és a
  - modell varianciája (túlilleszkedés) között

- Fordított kapcsolat áll fenn a torzítás és a variancia között. Amikor az egyik csökken, a másik hajlamos növekedni, és fordítva.

Modell torzítás

A bias (torzítás/elfogultság) azokra a hibákra utal, amelyek akkor fordulnak elő, amikor megpróbálunk egy statisztikai modellt valós adatokra illeszteni

- A modell nem képes megtanulni a mintázatokat a rendelkezésre álló adatokban, ezért rosszul teljesít.
- Ha túlságosan leegyszerűsített modellt használunk az adatokhoz való illesztéshez, akkor valószínűbb, hogy a magas torzítás helyzetével szembesülünk

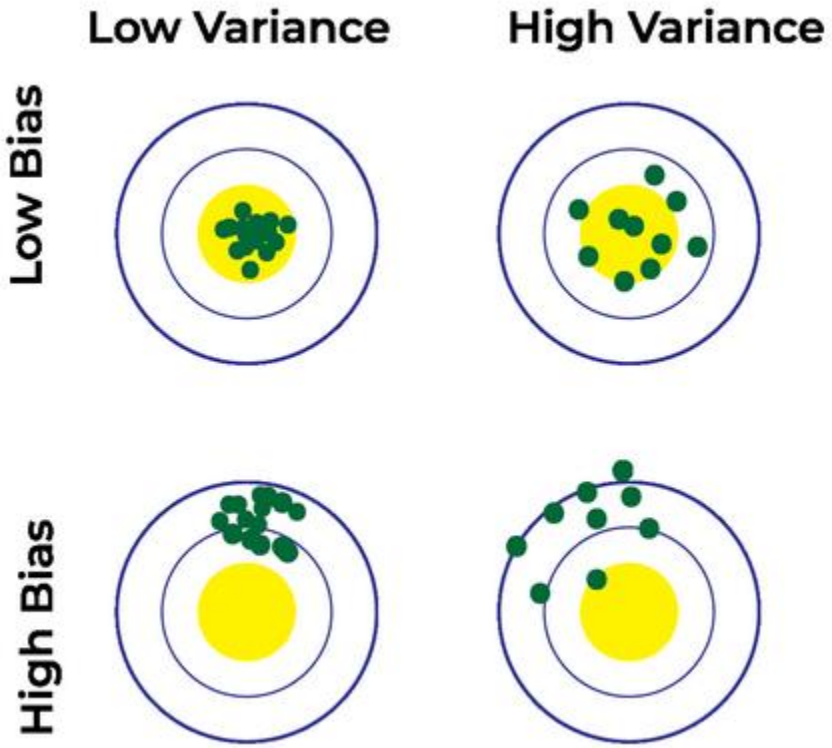
Modell variancia

- A variancia azt a hibát jelenti, amely akkor fordul elő, amikor a modell által korábban nem látott adatok felhasználásával próbálunk előrejelzéseket készíteni.
- Magas variancia akkor fordul elő, amikor a modell megtanulja az adatokban jelen lévő zajt.

Torzítás – variancia kombinációk

Low Bias, Low Variance:
- Az alacsony torzítású és alacsony varianciájú modell, ez a tökéletes forgatókönyv.
- Jól általánosít, és konzisztens, pontos előrejelzéseket készíthet.
- A valóságban azonban ez jellemzően nem megvalósítható.

High Bias, Low Variance:
- A nagy torzítású és alacsony varianciájú modell alulilleszkedőnek minősül.



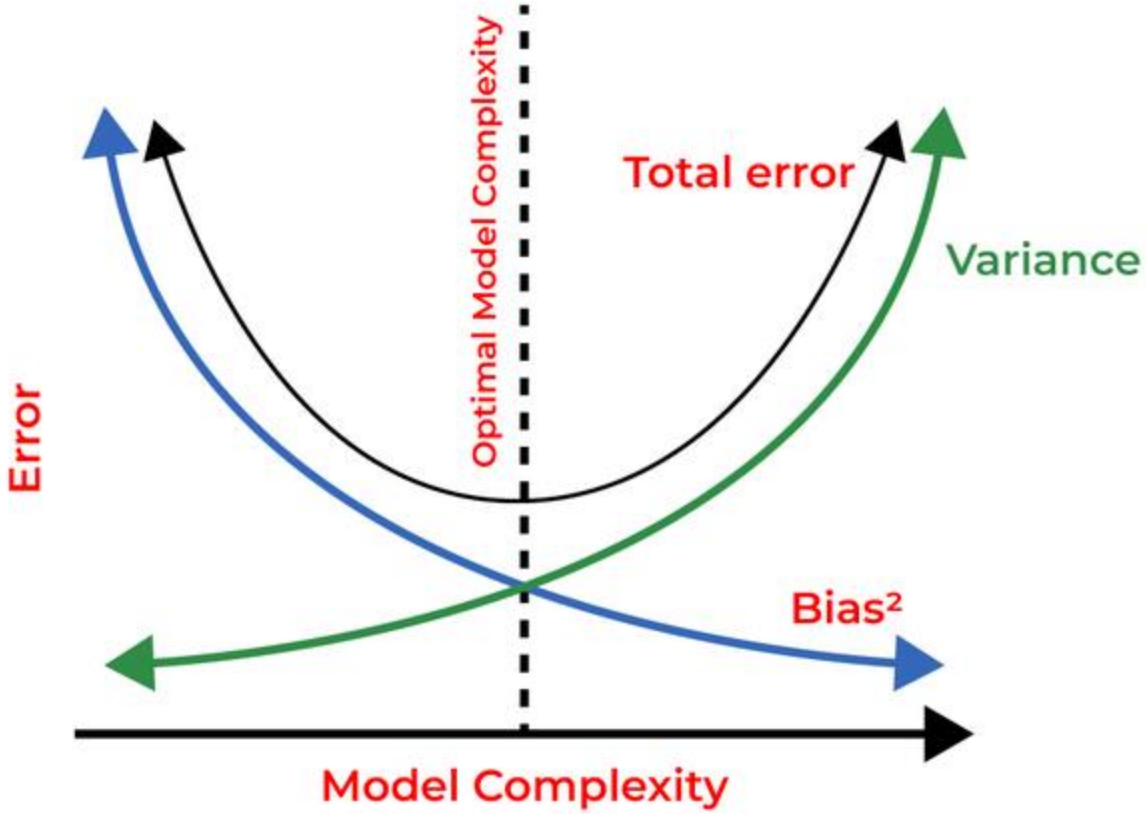
High Variance, Low Bias:
- A nagy varianciájú és alacsony torzítású modell túlilleszkedőnek minősül.

High-Bias, High-Variance:
- A nagy torzítású és nagy varianciájú modellek nem tudják megtanulni a mögöttes mintázatokat, és túl érzékenyek a tanító adatok változásaira.
- A modell általánosan megbízhatatlan és inkonzisztens előrejelzéseket generál.

https://www.geeksforgeeks.org/regularization-in-machine-learning/

Torzítás – variancia kompromisszum

- Fordított kapcsolat áll fenn a torzítás és a variancia között. Amikor az egyik csökken, a másik hajlamos növekedni, és fordítva.
- A túl egyszerű, nagy torzítású modellek nem rögzítik a mögöttes mintázatokat, míg a túl összetett, nagy varianciájú modellek túlilleszkednek az adatok zajához.



https://www.geeksforgeeks.org/regularization-in-machine-learning/

Regularizáció

- Early stopping
- Adat augmentáció
- L1 and L2 regularizáció
- Zaj adása bemenethez/címkékhez/gradienshez
- Dropout

Bala Priya: Regularization in Neural Networks, 2023 összefoglalója alapján

Regularizáció

Early stopping

- Ha a hiba/veszteség (loss) túl alacsony lesz, és tetszőlegesen közelíti a nullát, akkor a hálózat biztosan túlilleszkedik a tanító adathalmazon.
- Ezért, ha meg tudjuk akadályozni heurisztikusan, hogy a hiba /veszteség tetszőlegesen alacsony legyen, a modell kevésbé valószínű, hogy túlilleszkedik a tanító adathalmazon, és jobban fog általánosítani.

Early stopping 2.

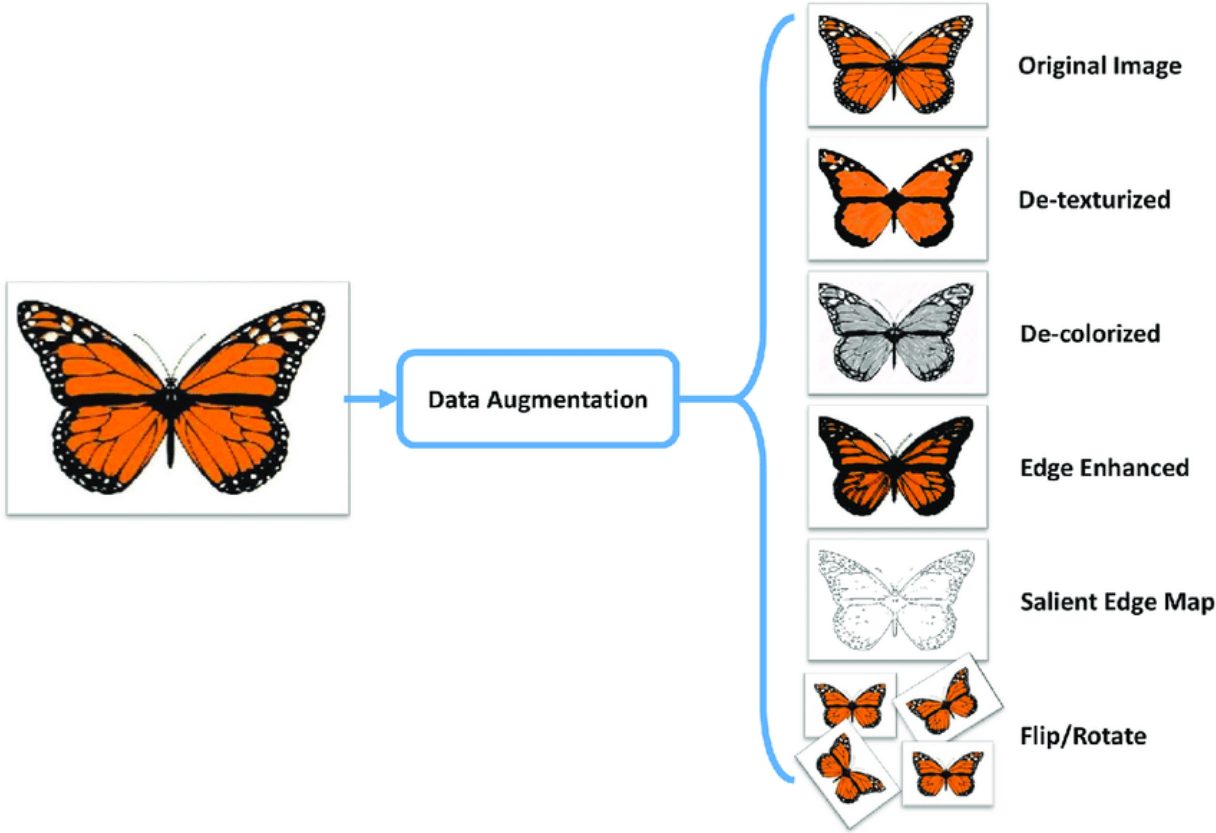
Egy másik módja annak, hogy tudjuk, mikor kell leállítani:

- a hálózat súlyai változásának figyelése
- legyen w_t és w_{t-k} a *t* és *t-k* epochok súlyvektorai
- kiszámítjuk a különbségvektor L2 normáját
- ha ez a mennyiség kisebb, mint ϵ, akkor leállíthatjuk a tanítást.

$$||w_t - w_{t-k}||_2 < \epsilon$$

Adataugmentáció

- Az adataugmentáció egy olyan regularizációs technika, amely segít a neurális hálózatnak jobban általánosítani azáltal, hogy változatosabb tanítómintáknak teszi ki azt.



- Mivel a mély neurális hálózatoknak nagy tanító adathalmazra van szükségük, az adataugmentáció akkor is hasznos, ha nem áll rendelkezésünkre elegendő adat egy neurális hálózat képzéséhez.

L1 és L2 regularizáció

- Általában az Lp normák (p>=1 esetén) a nagyobb súlyokat büntetik. Arra kényszerítik a súlyvektor normáját, hogy kellően kicsi maradjon. Az n-dimenziós térben az xx vektor Lp normája a következő:

$$Lp(\mathbf{x}) = ||\mathbf{x}||_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

- L1 norma: Ha p=1, akkor L1 normát kapunk, ami a vektor összetevői abszolút értékeinek összege:

$$L_1(\mathbf{x}) = ||\mathbf{x}||_1 = \sum_{i=1}^{n} |x_i|$$

L1 és L2 regularizáció

- L2 norma: Ha p=2, akkor az L2 normát kapjuk, ami a pontnak az origótól való euklideszi távolsága az n-dimenziós vektortérben:

$$L_2(\mathbf{x}) = ||\mathbf{x}||_2 = (\sum_{i=1}^{n} |x_i|^2)^{1/2}$$

L1 regularizáció

- Legyen L és L~ a veszteségfüggvények regularizáció nélkül, illetve regularizációval. Az L1 regularizált veszteségfüggvény a következő:

$$\tilde{\mathcal{L}}(\mathbf{w}) = \mathcal{L}(\mathbf{w}) + \alpha||\mathbf{w}||_1$$

- ahol α a regularizációs konstans.
- Ha a súlyok túl nagyok lesznek, a második kifejezés növekszik.
- Mivel azonban az optimalizálás célja a veszteségfüggvény minimalizálása, a súlyoknak csökkennie kell.

L1 regularizáció

$$\tilde{\mathcal{L}}(\mathbf{w}) = \mathcal{L}(\mathbf{w}) + \alpha||\mathbf{w}||_1$$

- ahol α a regularizációs konstans.
- Ha a súlyok túl nagyok lesznek, a második kifejezés növekszik.
- Mivel azonban az optimalizálás célja a veszteségfüggvény minimalizálása, a súlyoknak csökkennie kell.
- Különösen az L1 regularizáció segíti elő a súlymátrix ritkaságát azáltal, hogy a súlyokat nullára kényszeríti.
- Ezáltal jegykiválaszlást is végez.

L2 regularizáció

- Hasonlóképpen, a regularizált veszteségfüggvény L2 regularizációval (más néven L2 súlycsökkenés) a következő:

$$\tilde{\mathcal{L}}(\mathbf{w}) = \mathcal{L}(\mathbf{w}) + \frac{\alpha}{2}||\mathbf{w}||_2{}^2$$

- Ha a súlyok túl nagyok lesznek, ennek következtében a fenti egyenlet második tagja megnő.
- Mivel azonban a cél a veszteségfüggvény minimalizálása, az algoritmus a kisebb súlyokat részesíti előnyben.

L2 regularizáció

$$\tilde{\mathcal{L}}(\mathbf{w}) = \mathcal{L}(\mathbf{w}) + \frac{\alpha}{2}||\mathbf{w}||_2{}^2$$

- L2 regularizáció csökkenti a súlyokat, miközben biztosítja, hogy a súlyvektor fontos összetevői nagyobbak legyenek a többinél.

Zaj hozzáadás bemenethez/címkékhez/gradienshez

Zaj hozzáadása…

- Bemenetekhez
- Címkékhez
- Gradienshez

Zaj hozzáadás bemenethez/címkékhez/gradienshez

Zaj hozzáadása bemenetekhez

$$\hat{x}_i = x_i + \epsilon_i$$

$$\hat{y}_i = \sum_{i=1}^{n} w_i \hat{x}_i$$

- Négyzetösszeg veszteségfüggvény esetén, normális eloszlású zaj hozzáadása a bemenetekhez egyenértékű az L2 regularizációval

Zaj hozzáadás bemenethez/címkékhez/gradienshez

Zaj hozzáadása címkékhez

- Ha „zajt adunk” a kimeneti címkékhez, a hálózat nem tudja megjegyezni a tanító adathalmazt
- Címkék perturbálása (megkeverése)
  - Minden tanítómintát p valószínűséggel „zavarunk meg”.
  - Minden megzavart minta esetében a címkét egyenletes eloszlásból kell sorsolni {1,2,3,…,N} függetlenül a valódi címkétől.
- Label smoothing

Class Label	1	2	3	…	k	…	N
With Label Smoothing	$\frac{1-\epsilon}{N-1}$	$\frac{1-\epsilon}{N-1}$	$\frac{1-\epsilon}{N-1}$	…	ϵ	…	$\frac{1-\epsilon}{N-1}$

Zaj hozzáadás bemenethez/címkékhez/gradienshez

Zaj hozzáadása gradienshez

$$G_{w_i}^t = G_{w_i}^t + \mathcal{N}(0, \sigma_t^2)$$

Ahol $G_{w_i}^t$ a t-dik lépésben kiszámolt gradiensvektor (wi súlyokra), és $\mathcal{N}(0, \sigma_t^2)$ normális eloszlású zaj (Gauss-zaj) 0 várható értékkel és σ_t^2 varianciával.

A variancia idővel (lépésszám függvényében) változhat (csökkenhet)

$$\sigma_t^2 = \frac{\eta}{(1+t)^\gamma}$$

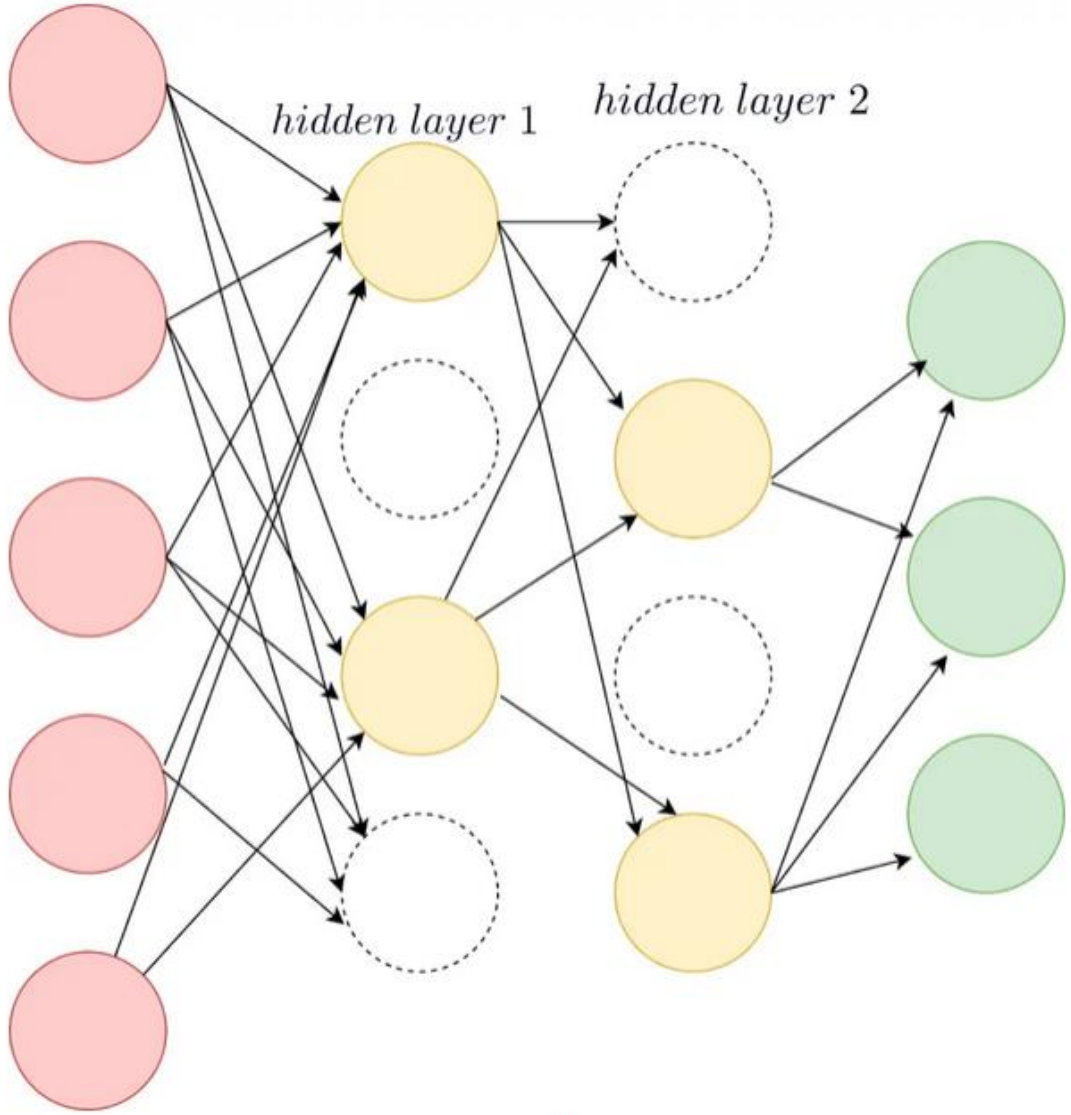
Dropout

- A dropout a rejtett rétegek és (opcionálisan) a bemeneti réteg neuronjainak elhagyását jelenti.
- A tanítás során minden neuronhoz hozzárendelünk egy "dropout" valószínűséget, például 0,5.

- A 0,5-ös dropoutvalószínűség esetén 50% az esélye annak, hogy egy neuron részt vesz a tanításban minden egyes tanítási szakaszon (batch-en) belül.

Dropout

- Ez azt eredményezi, hogy a hálózat architektúrája minden egyes batch-ben kissé eltérő.
- Ez egyenértékű a különböző neurális hálózatoknak a tanító adatok különböző részhalmazain történő tanításával.



Dropout

https://www.pinecone.io/learn/regularization-in-neural-networks/

Hiperparaméterek

A hiperparaméterek a modelltervezéssel kapcsolatos kérdésekre adhatnak választ.

- Milyen fokú polinomiális jellemzőket használjunk egy lineáris modellhez?
- Mekkora legyen a döntési fa maximális mélysége?
- Mennyi legyen a minimálisan szükséges minták száma a döntési fa egy levélcsomópontjában?
- Hány fát kell felvenni a véletlen erdőbe?
- Hány neuron legyen a neurális hálózat egy rétegében?
- Hány réteggel kell rendelkeznie a neurális hálózatnak?
- Mekkora tanulási tényezőt állítsunk be a gradiens ereszkedéshez?

Hiperparaméterek hangolása

- A modell meghatározása
- Az összes lehetséges érték tartományának meghatározása minden hiperparaméterhez
- Mintavételi módszer meghatározása a hiperparaméter értékek kiválasztásához
- A modell jósága megítéléséhez kritériumok definiálása és számítása
- Keresztvalidációs módszer meghatározása