

Gyires-Tóth Bálint

Deep Learning a gyakorlatban Python és LUA alapon Backpropagation

Jogi nyilatkozat

Jelen előadás diái a „*Deep Learning a gyakorlatban Python és LUA alapon*” című tantárgyhoz készültek és letölthetők a <http://smartlab.tmit.bme.hu> honlapról.

A diák nem helyettesítik az előadáson való részvételt, csupán emlékeztetőül szolgálnak.

Az előadás diái a szerzői jog védelme alatt állnak. Az előadás diáinak vagy bármilyen részének újra felhasználása, terjesztése, megjelenítése csak a szerző írásbeli beleegyezése esetén megengedett. Ez alól kivétel, mely diákon külső forrás külön fel van tüntetve.

Kis házi feladat

- GitHubon hirdetve

- Beadni:

<http://smartlab.tmit.bme.hu/oktatas-deep-learning>

Nagy házi feladat

- Csapatok toborzása Microsoft Teams-en
- 4. hét végéig csapat minden tagjának regisztrációja a <http://smartlab.tmit.bme.hu/oktatas-deep-learning> oldalon.
- Témalabor, BSc, MSc diploma
- PhD téma
- 3 fős csapatok
- 1, 2, 4 fős csapatok
- Téma javaslatok

Tanítási feladat, adatok



- Bemenet: [láz (°C), gyógyszer (mg)] = x
- Kimenet
 - Regresszió: [láz 2 óra múlva] = y
 - Osztályozás: [láz/hőemelkedés/normális 2 óra múlva]

Tanító és teszt adatok

- Példa

$$X = \begin{bmatrix} 38.6 & 25 \\ 37.8 & 25 \\ 37.9 & 50 \\ 38.2 & 50 \end{bmatrix}$$

$$y = \begin{bmatrix} 37.2 \\ 37.3 \\ 36.6 \\ 36.9 \end{bmatrix}$$

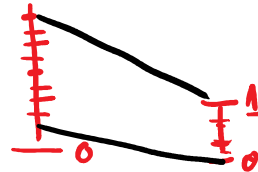
$$X_{\text{test}} = \begin{bmatrix} 38.3 & 35 \end{bmatrix}$$

$\hat{y} = ?$

$$x = (x - \text{mean} X) / \text{std} X$$

→ szórási
várható érték

$$y = \text{minmax}(y, 0, 1)$$

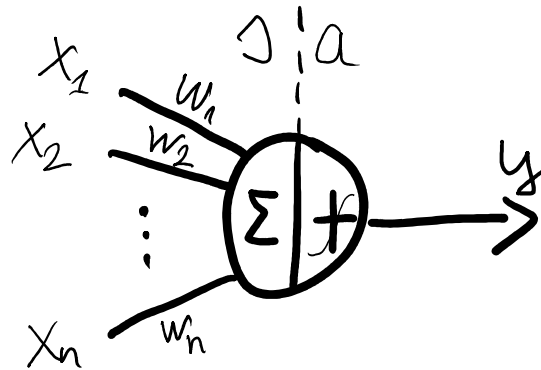


Min max scaler

Kimenetek átskálázása

$$\begin{aligned} y_{std} &= (y - \min y) / (\max y - \min y) \\ y_{scaled} &= y_{std} \cdot (\max - \min) + \min \end{aligned} \quad \left| \begin{array}{l} \max = 1 \\ \min = 0 \end{array} \right.$$

Elemi neuron



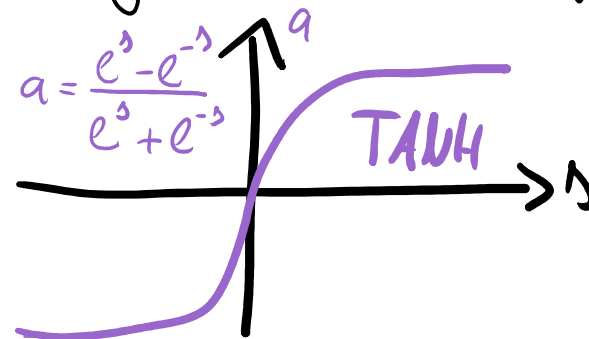
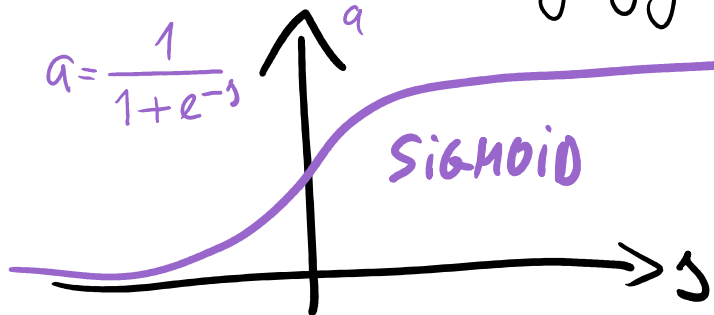
Summa

$$\Delta = \sum_{i=1}^n w_i x_i$$

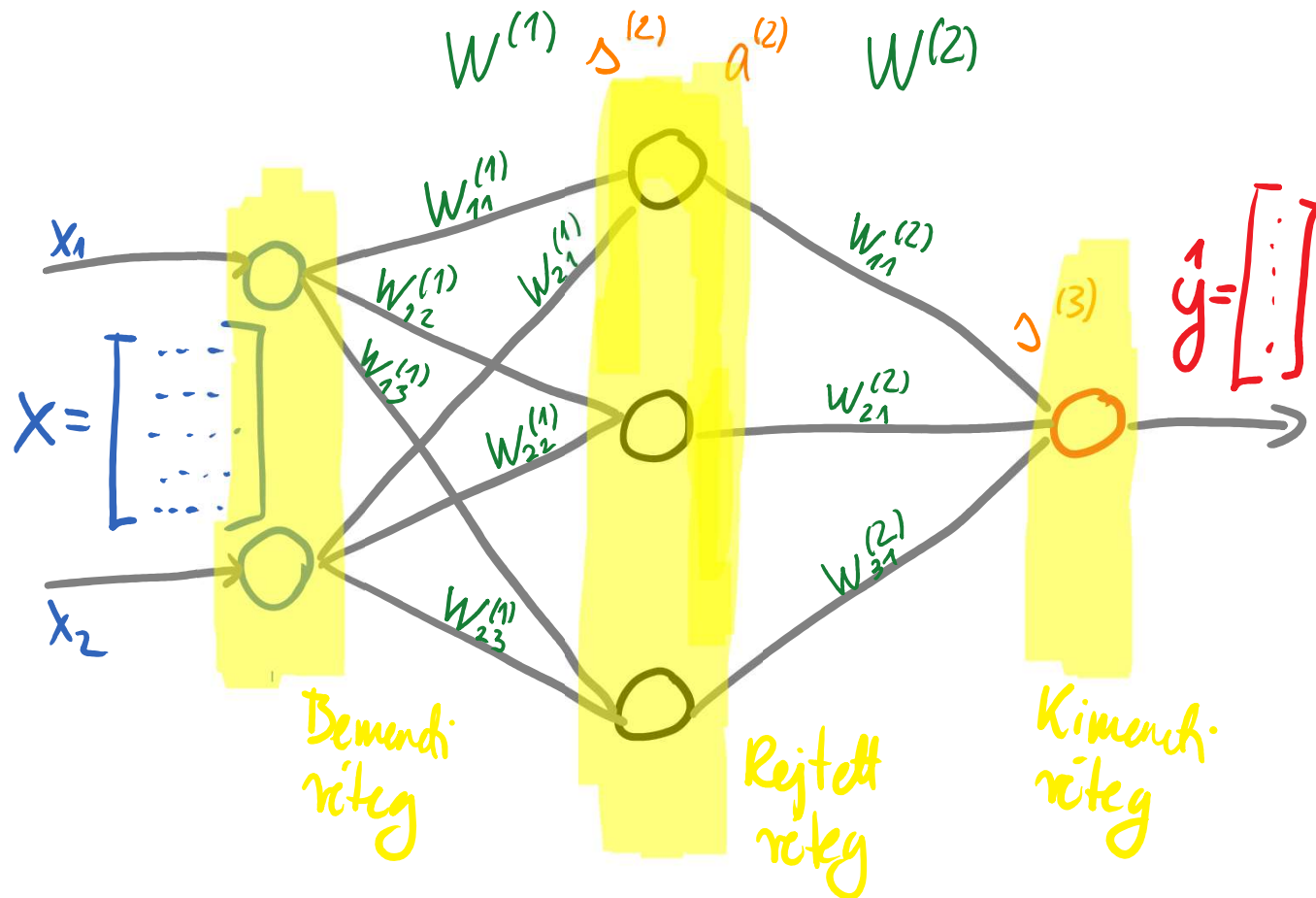
$y = a(\Delta)$

aktivitás

a : aktivációs függvény (sgm, tanh, ReLU...)



Egyszerű neurális hálózat



Mátrixalgebra

- Mátrix szorzás
- Transzponálás
- Parciális derivált



theano



Forward lépés I.

$$\textcircled{1} \quad \overset{(4 \times 2)}{X} \overset{(2 \times 3)}{W^{(1)}} = \overset{(4 \times 3)}{\Delta^{(2)}}$$

$$W^{(1)} = \begin{bmatrix} W_{11}^{(1)} & W_{12}^{(1)} & W_{13}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} & W_{23}^{(1)} \end{bmatrix}$$

$$\underbrace{\overset{\text{minták sora}}{X} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} \\ x_1^{(2)} & x_2^{(2)} \\ x_1^{(3)} & x_2^{(3)} \\ x_1^{(4)} & x_2^{(4)} \end{bmatrix}}_{\text{tulajdonságok (features)}} \begin{bmatrix} x_1^{(1)} W_{11}^{(1)} + x_2^{(1)} W_{21}^{(1)} & x_1^{(1)} W_{12}^{(1)} + x_2^{(1)} W_{22}^{(1)} & x_1^{(1)} W_{13}^{(1)} + x_2^{(1)} W_{23}^{(1)} \\ x_1^{(2)} W_{11}^{(1)} + x_2^{(2)} W_{21}^{(1)} & x_1^{(2)} W_{12}^{(1)} + x_2^{(2)} W_{22}^{(1)} & x_1^{(2)} W_{13}^{(1)} + x_2^{(2)} W_{23}^{(1)} \\ x_1^{(3)} W_{11}^{(1)} + x_2^{(3)} W_{21}^{(1)} & x_1^{(3)} W_{12}^{(1)} + x_2^{(3)} W_{22}^{(1)} & x_1^{(3)} W_{13}^{(1)} + x_2^{(3)} W_{23}^{(1)} \\ x_1^{(4)} W_{11}^{(1)} + x_2^{(4)} W_{21}^{(1)} & x_1^{(4)} W_{12}^{(1)} + x_2^{(4)} W_{22}^{(1)} & x_1^{(4)} W_{13}^{(1)} + x_2^{(4)} W_{23}^{(1)} \end{bmatrix} = \Delta^{(2)}$$

Forward lépés II.

$$\textcircled{2} \quad a^{(2)} = f(\Delta^{(2)}) = \text{Sigmoid}\{\Delta^{(2)}\} \quad (\text{tanh/ReLU/ReLU} \dots)$$

sigmoid $\equiv$ sgm

$$a^{(2)} = \begin{bmatrix} \text{sgm}(\Delta_{11}^{(2)}) & \text{sgm}(\Delta_{12}^{(2)}) & \text{sgm}(\Delta_{13}^{(2)}) \\ \vdots & \vdots & \vdots \\ \text{sgm}(\Delta_{33}^{(2)}) \end{bmatrix}$$

(4x3)

$$\textcircled{3} \quad \Delta^{(3)} = a^{(2)} W^{(2)}$$

(4x1) (4x3) (3x1)

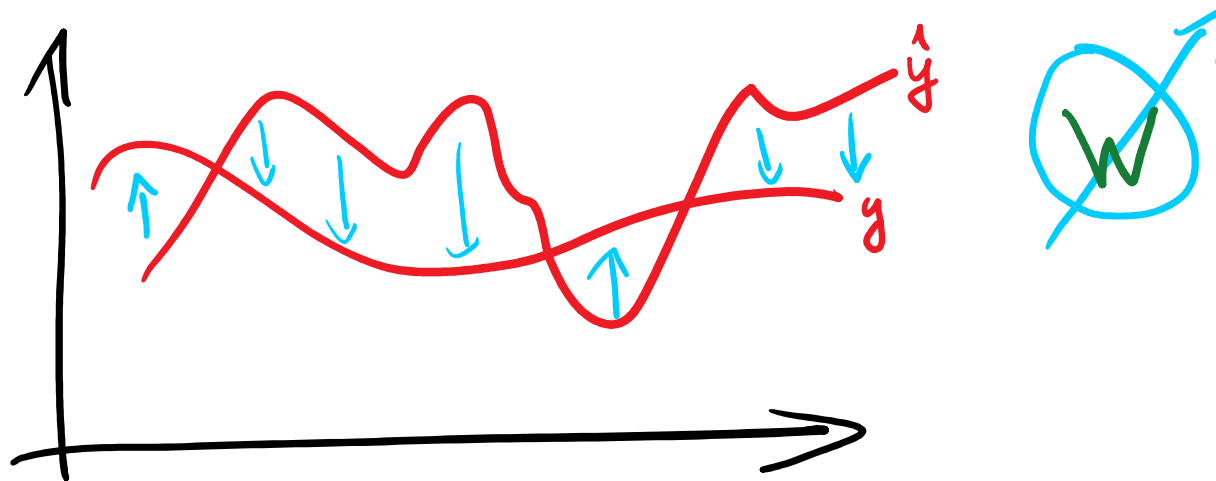
$$\textcircled{4} \quad \hat{y} = f(\Delta^{(3)}) = \text{sigmoid}\{\Delta^{(3)}\} \quad (\text{tanh/ReLU/ReLU} \dots)$$

(4x1)

Forward lépés III.: Cost function

- Magyarul: költségfüggvény
- Négyzetes hiba (regresszió) vagy ^⑤ keresztentrópia (osztályozás)
- Kezdetben súlyok értéke véletlenszám

$$C = \sum \frac{1}{2} (y - \hat{y})^2$$



Súlyok finomhangolása

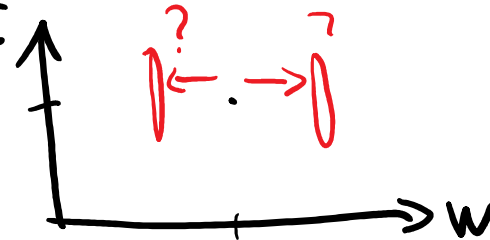
- Brute force
1 súly $\rightarrow [-100, 100]$ 0.1-es felbontással 2000 eset
2 súly $\rightarrow 2000^2 = 4 * 10^6$

...

9 súly $\rightarrow 2000^9 = 512 * 10^{27}$

közepes méretű háló, 6 rejtett réteg, 1000 neuron
rétegenként: $6 * 1000^2$ súly

- Numerikus gradiens keresés



Kahoot!

Névnek a NEPTUN
kódodat add meg!

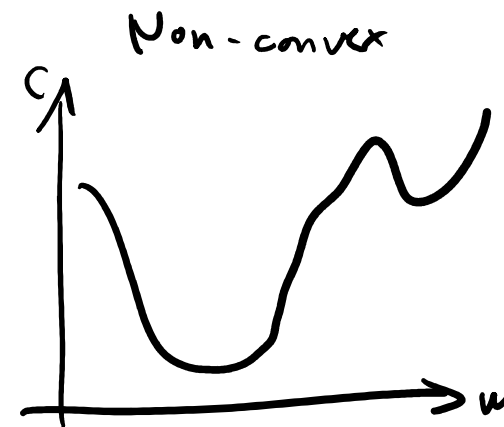
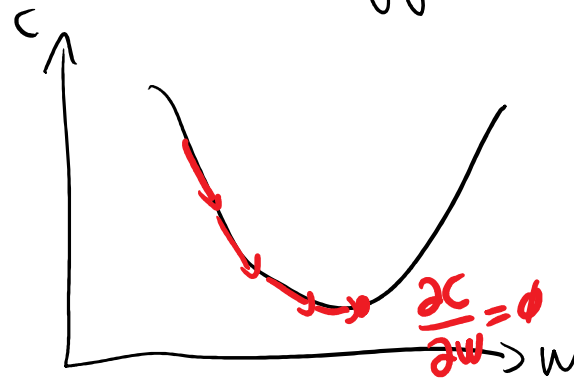
<https://kahoot.it/>

Gradient descent

$$\textcircled{6} C = \sum \left\{ \frac{1}{2} (y - f(f(XW^{(1)}|W^{(2)})))^2 \right\}$$

$$\frac{\partial C}{\partial W} > 0 \nearrow \quad < 0 \searrow$$

Cél $\Rightarrow$ elérni hogy $= 0$



Backprop: kimeneti – rejtett réteg

$$\frac{\partial C}{\partial W^{(2)}} = \frac{\partial \sum \frac{1}{2} (y - \hat{y})^2}{\partial W^{(2)}} \stackrel{(a)}{=} \sum \frac{\partial \frac{1}{2} (y - \hat{y})^2}{\partial W^{(2)}}$$

$$\stackrel{(b)}{\Rightarrow} \frac{1}{2} (y - \hat{y})^2 \cdot - \frac{\partial \hat{y}}{\partial W^{(2)}} \stackrel{(c)}{=} -$$

$$= -(y - \hat{y}) \cdot \frac{\partial \hat{y}}{\partial s^{(3)}} \cdot \frac{\partial s^{(3)}}{\partial W^{(2)}} =$$

$$= \underbrace{-(y - \hat{y}) \cdot f'(s^{(3)})}_{\sigma^{(3)} \quad (1 \times 1)} \cdot \underbrace{\frac{\partial a^{(2)}}{\partial W^{(2)}}}_{a^{(2)} \quad (1 \times 3)}$$

$$\textcircled{a} \frac{d}{dx} (u+v) = \frac{du}{dx} + \frac{dv}{dx}$$

$$\textcircled{b} \text{lánc szabály:} \\ (f \cdot g)' = (f' \cdot g) + f \cdot g'$$

$$\frac{ds}{dx} = \frac{ds}{dy} \cdot \frac{dy}{dx}$$

$$\textcircled{d} f(s) = \frac{1}{1 + e^{-s}} = v$$

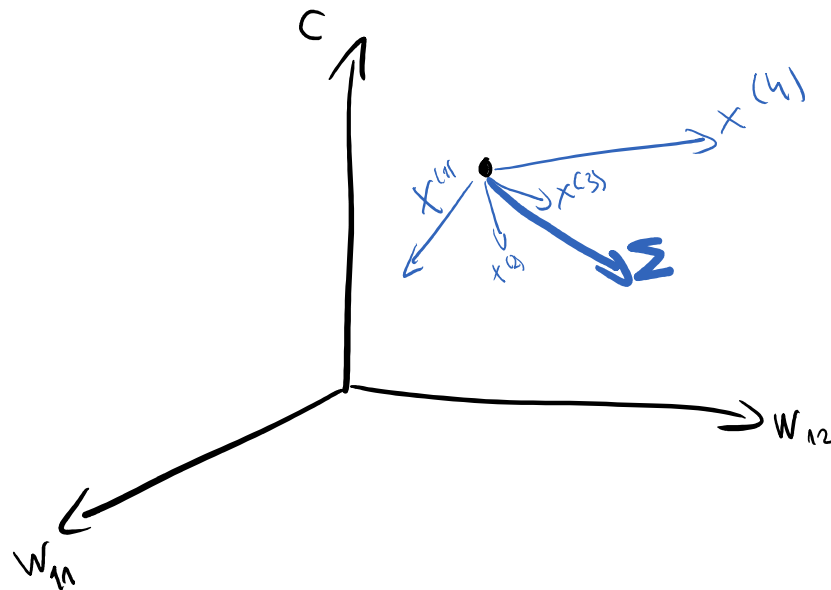
$$f'(s) = \frac{u'v - uv'}{v^2} = \frac{0 \cdot v - 1 \cdot (-e^{-s})}{(1 + e^{-s})^2} = \frac{e^{-s}}{(1 + e^{-s})^2}$$

Backprop: Batch gradient descent

$$\begin{aligned}
 \textcircled{7} \quad \frac{\partial C}{\partial W^{(2)}} &= \sum \frac{\partial \frac{1}{2} (y - \hat{y})^2}{\partial W^{(2)}} = (a^{(2)})^T \delta^{(3)} = \\
 & \quad \begin{bmatrix} \delta_1^{(3)} \\ \delta_2^{(3)} \\ \delta_3^{(3)} \\ \delta_4^{(3)} \end{bmatrix} \\
 & \quad \text{tanító minták száma} \\
 &= \begin{bmatrix} a_{11}^{(2)} & a_{21}^{(2)} & a_{31}^{(2)} & a_{41}^{(2)} \\ a_{12}^{(2)} & a_{22}^{(2)} & a_{32}^{(2)} & a_{42}^{(2)} \\ a_{13}^{(2)} & a_{23}^{(2)} & a_{33}^{(2)} & a_{43}^{(2)} \end{bmatrix} \begin{bmatrix} a_{11}^{(2)} \delta_1^{(3)} + a_{21}^{(2)} \delta_2^{(3)} + a_{31}^{(2)} \delta_3^{(3)} + a_{41}^{(2)} \delta_4^{(3)} \\ a_{12}^{(2)} \delta_1^{(3)} + a_{22}^{(2)} \delta_2^{(3)} + a_{32}^{(2)} \delta_3^{(3)} + a_{42}^{(2)} \delta_4^{(3)} \\ a_{13}^{(2)} \delta_1^{(3)} + a_{23}^{(2)} \delta_2^{(3)} + a_{33}^{(2)} \delta_3^{(3)} + a_{43}^{(2)} \delta_4^{(3)} \end{bmatrix}
 \end{aligned}$$

Backprop: Batch gradient descent

Az összes tanítómintára kiszámoljuk a gradienst és ezeket összegezzük.



Backprop: rejtett – bemeneti réteg

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial w^{(1)}} &= \frac{\partial \frac{1}{2}(y - \hat{y})^2}{\partial w^{(1)}} = -(y - \hat{y}) \cdot \frac{\partial \hat{y}}{\partial w^{(1)}} = -(y - \hat{y}) \cdot \frac{\partial \hat{y}}{\partial s^{(3)}} \cdot \frac{ds^{(3)}}{dw^{(1)}} = -(y - \hat{y}) \cdot f'(s^{(3)}) \frac{ds^{(3)}}{dw^{(1)}} \\ &= \delta^{(3)} \cdot \frac{ds^{(3)}}{dw^{(1)}} = \delta^{(3)} \cdot \frac{ds^{(3)}}{da^{(2)}} \cdot \frac{da^{(2)}}{dw^{(1)}} = \delta^{(3)} \cdot (W^{(2)})^T \cdot \frac{da^{(2)}}{dw^{(1)}} = \\ &= \delta^{(3)} \cdot (W^{(2)})^T \frac{da^{(2)}}{ds^{(2)}} \cdot \frac{ds^{(2)}}{dw^{(1)}} = \underbrace{\delta^{(3)} \cdot (W^{(2)})^T \cdot f'(s^{(2)})}_{\delta^{(2)}} \underbrace{\frac{ds^{(2)}}{dw^{(1)}}}_X = X^T \cdot \delta^{(2)}\end{aligned}$$

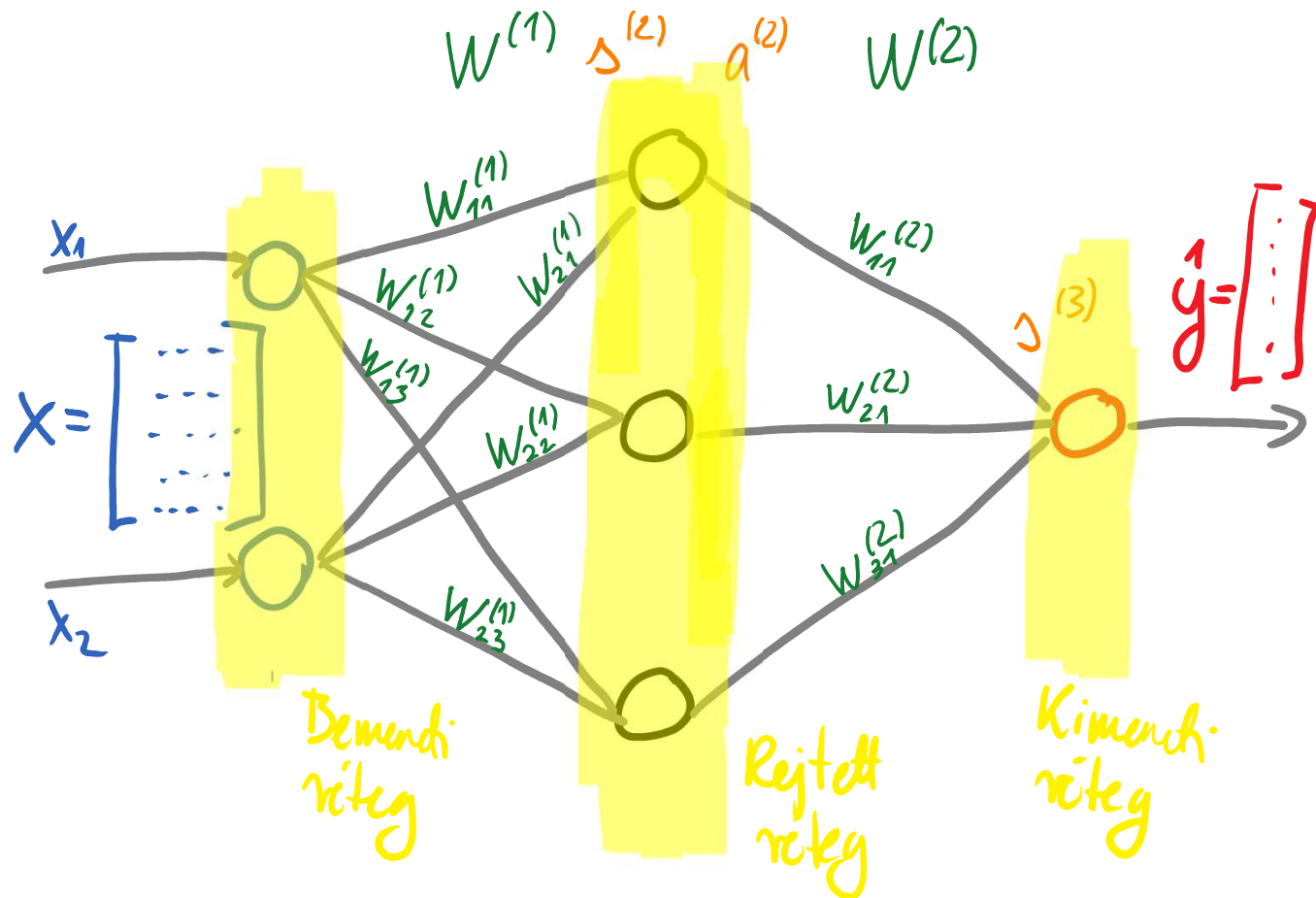
TANULÁS

$$W^{(1)} = W^{(1)} - \mu \frac{\partial \mathcal{L}}{\partial w^{(1)}}$$

$$W^{(2)} = W^{(2)} - \mu \frac{\partial \mathcal{L}}{\partial w^{(2)}}$$

μ : tanulási ráta
(learning rate)

Egyszerű neurális hálózat



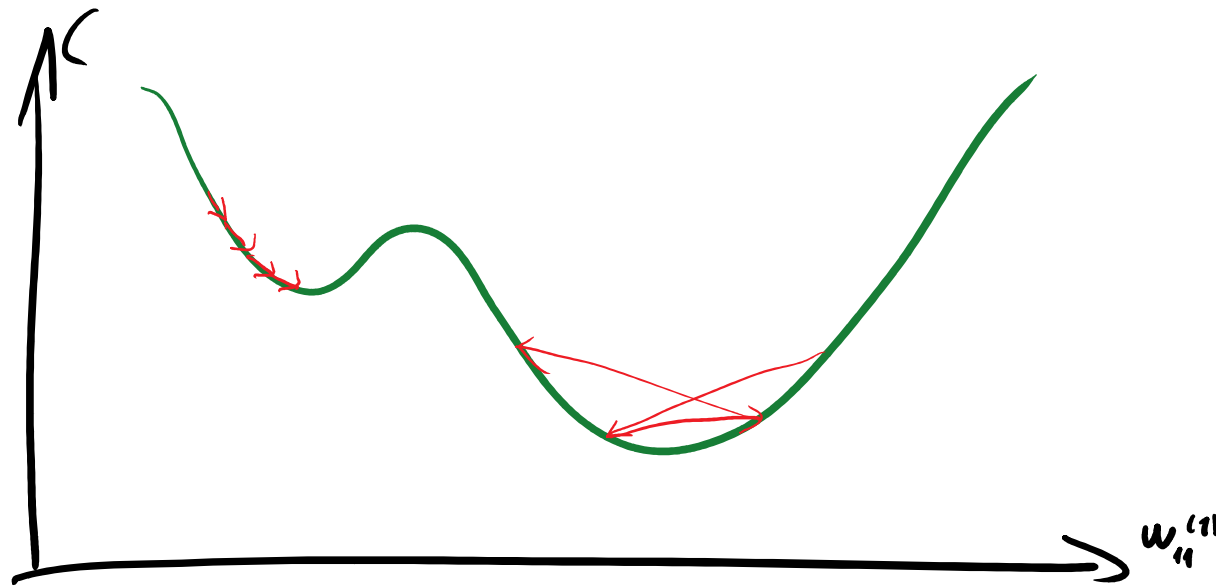
Gradient descent

- Stochastic Gradient Descent (SGD)
- Batch GD
- Mini-Batch GD

Gradient descent

Lokális vs. Globális minimum

Batch learning, Mini-batch learning



Intuitív magyarázat

- Deep learning: több rétegen keresztül ugyanez
- Yann LeCun: Efficient Backprop (1998)
<http://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf>
- További magyarázat és online demo:
<https://www.deeplearning.ai/ai-notes/optimization/>

Acknowledgement

The picture of Slide 6 was designed by Freepik.com.



Köszönöm a figyelmet!

`toth.b@tmit.bme.hu`