Gyires-Tóth Bálint

Deep Learning a gyakorlatban Python és LUA alapon
Súly inicializálás, Konvolúciós hálózatok

SmartLab
Intelligent Interactions

http://smartlab.tmit.bme.hu

NVIDIA
GPU
EDUCATION
CENTER

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

1. Javítás alatt.
2. Leadás jövő hét kedd.

Nagy házi feladat további teendők

http://smartlab.tmit.bme.hu/oktatas-deep-learning-nagy-hazi

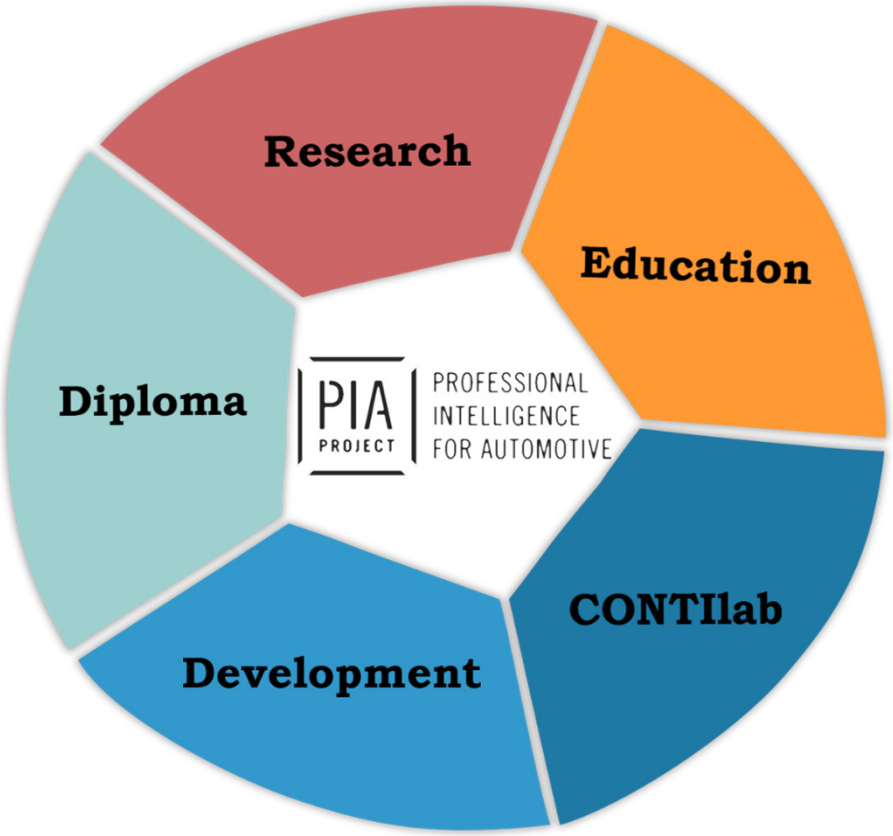
Milestone 1: Október 25., 23:59

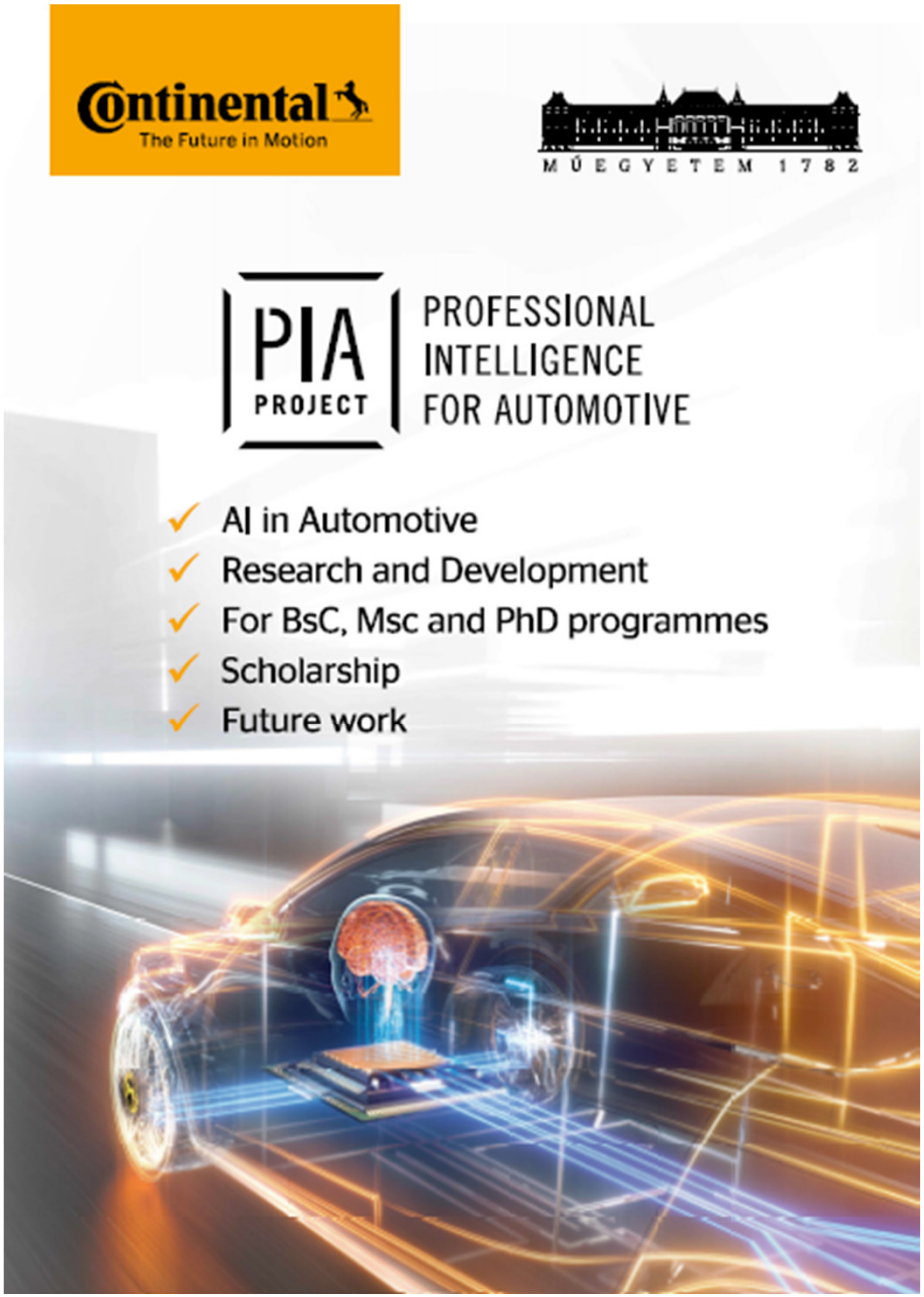
Adatok beszerzése, adatfeltárás, vizualizáció (ha szükséges) és előkészítés tanításhoz.
Eredmény: tanító, validációs és teszt adatbázisok

Megajánlott jegyért a nyelv: ANGOL

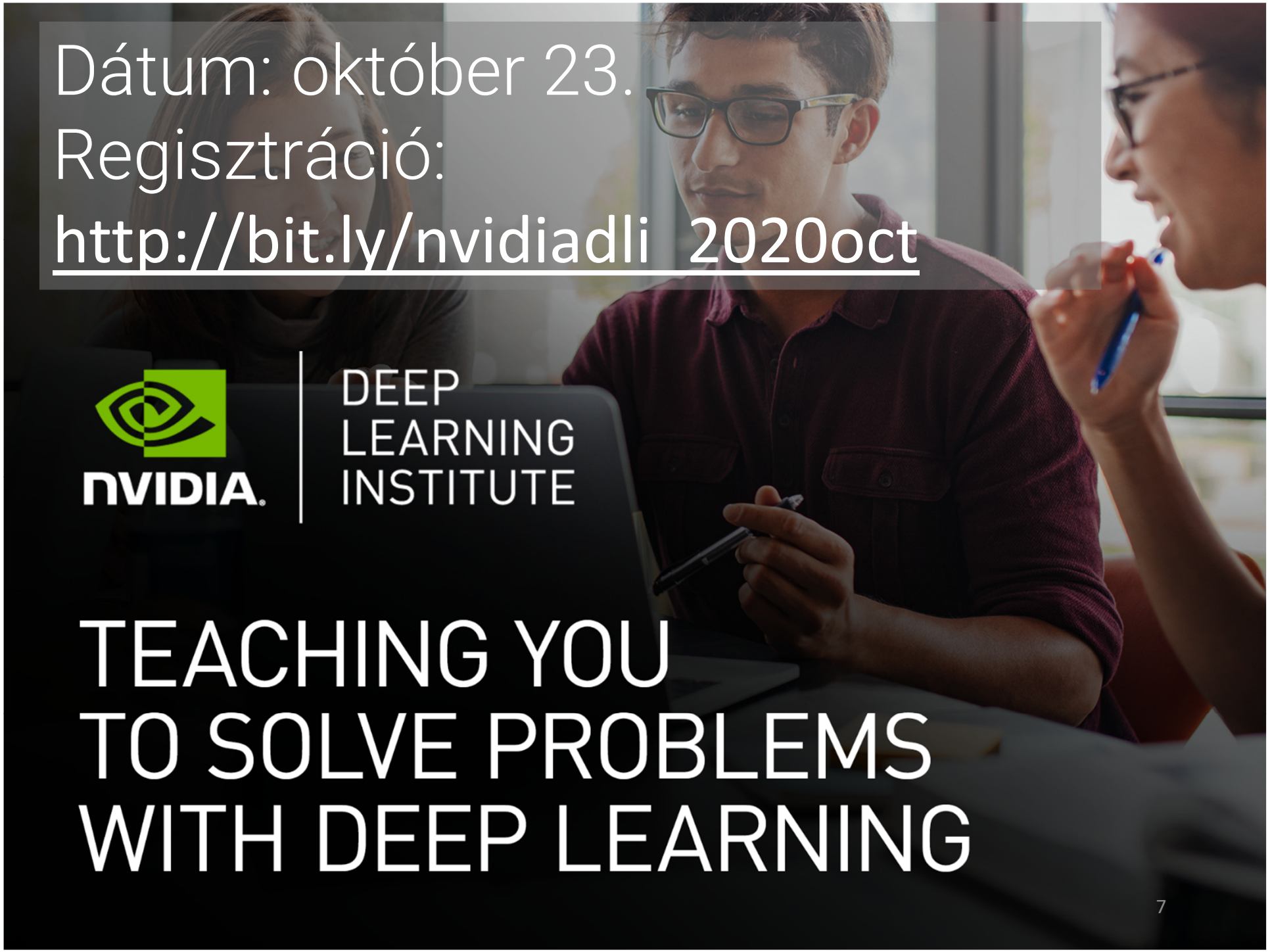
Deep Learning Híradó

Hírek az elmúlt 168 órából

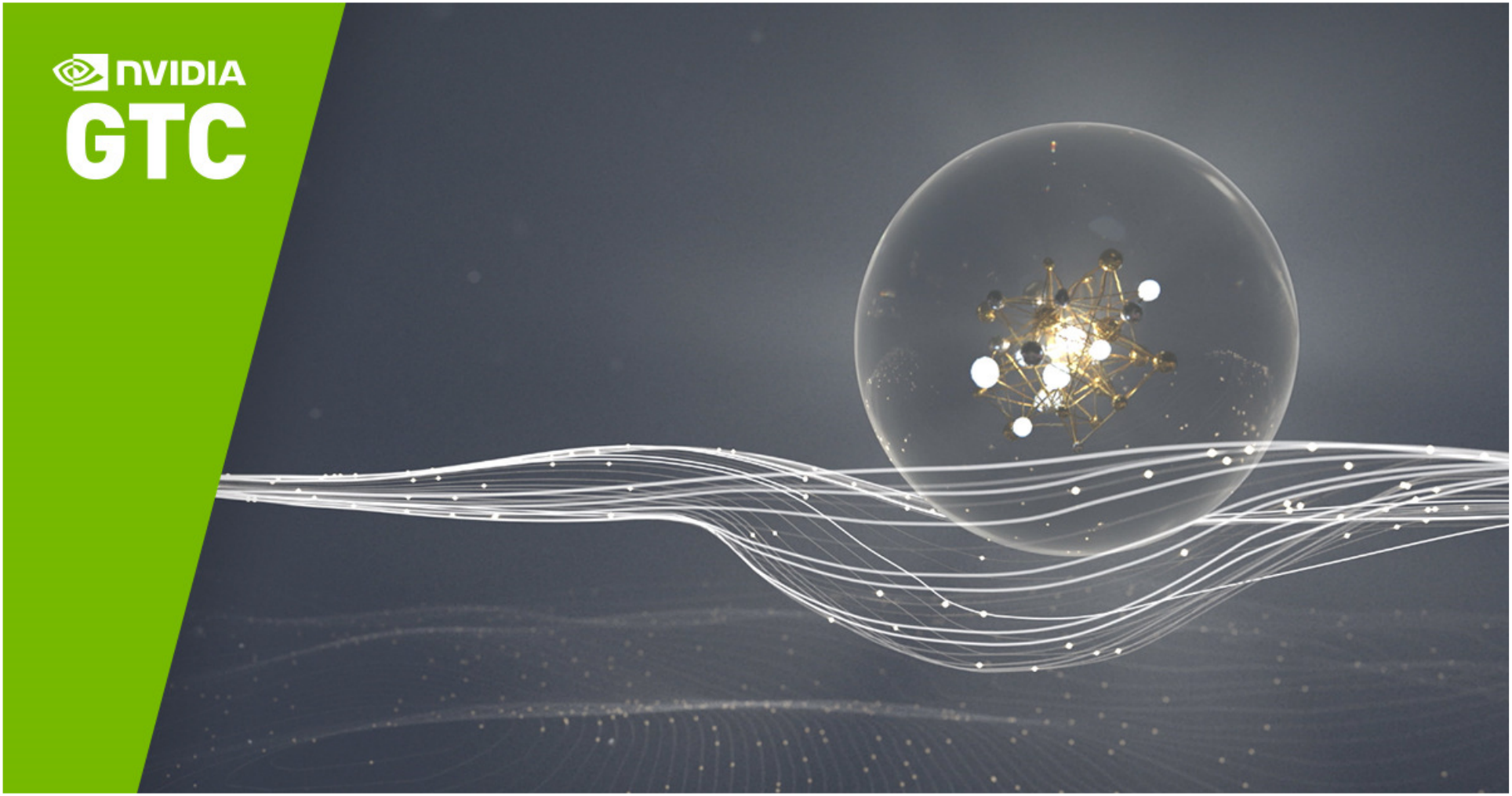




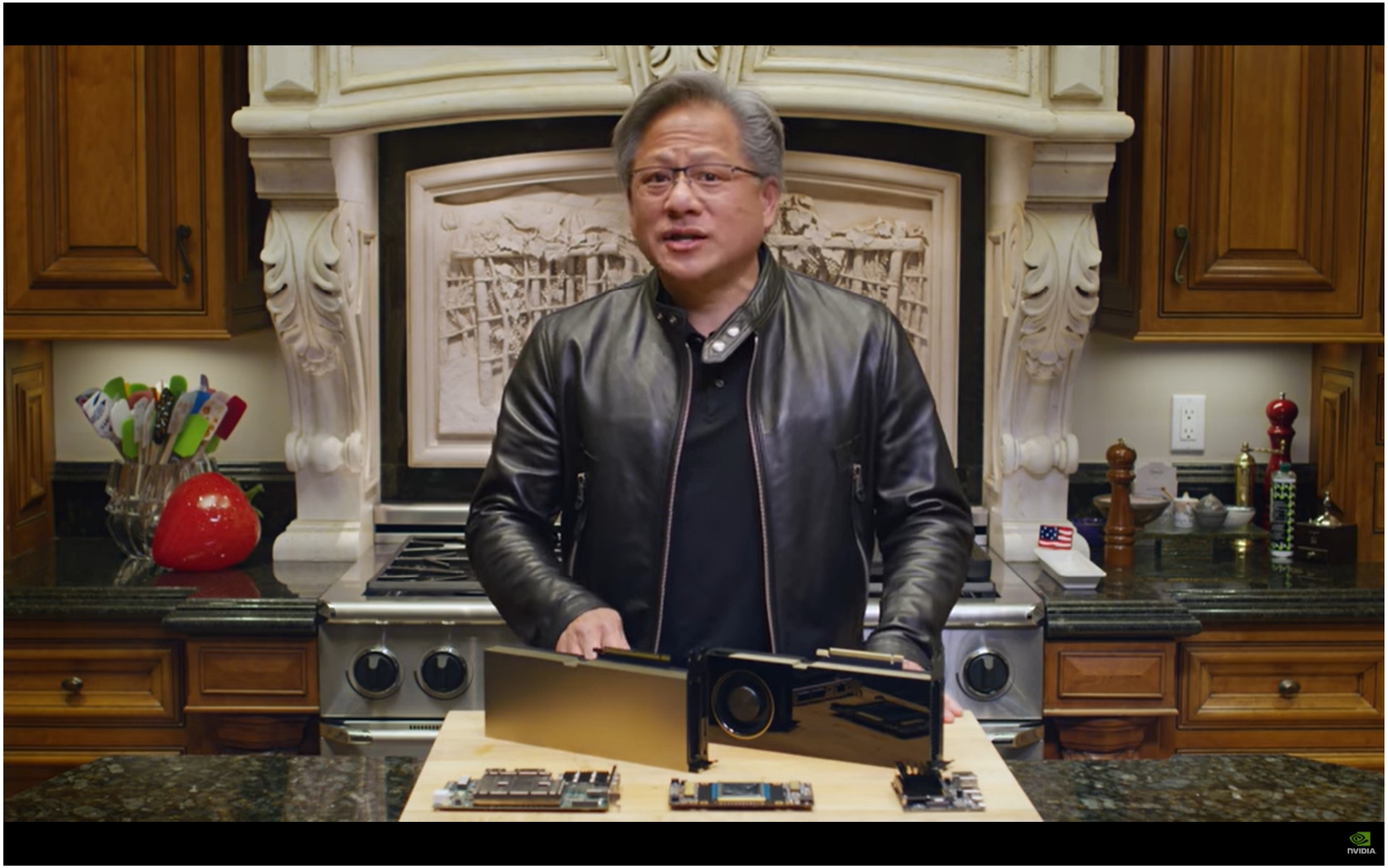
http://bit.ly/piaday2020


Dátum: október 23.
Regisztráció:
http://bit.ly/nvidiadli_2020oct
NVIDIA
DEEP LEARNING INSTITUTE
TEACHING YOU TO SOLVE PROBLEMS WITH DEEP LEARNING

Regisztráció:
www.nvidia.com/gtc
Referencia kód: NVMDIERINGER



https://www.nvidia.com/en-us/gtc/keynote/



Nvidia MAXIM

Egy kis elmélet...

Segédlet: Vanishing gradient probléma

Gradiens: célja, hogy megadja, hogy egy-egy súly aktuális értéke milyen mértékben járul hozzá a hibához.

Vanishing gradient: A mély rétegek súlyai nagyon kis mértékben járulnak hozzá a hibához.

Oka: sigmoid / tanh az egész értéktartományt 0..1 (vagy -1..1) tartományba „nyomja össze”. A réteg bemenetén lévő nagy változások a kimeneten kis változást okoznak.

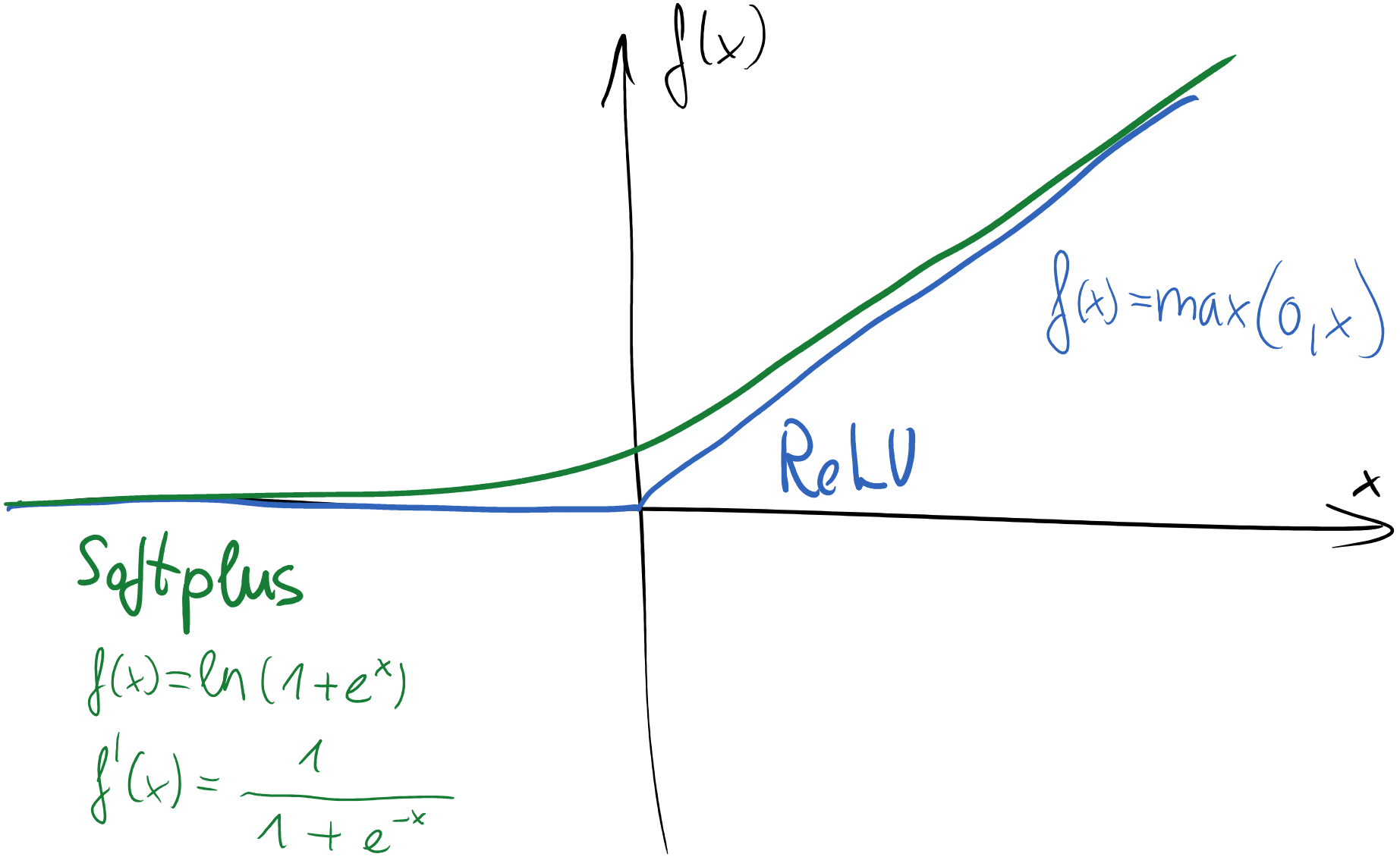
Ez sok rétegen át, kis súlyok mellet a gradiens nagyon kicsire csökkenéséhez vezet → mély rétegek nem tanulnak.

ReLU, LReLU, PReLU

Rectified Linear Unit, Leaky ReLU, Parametric ReLU

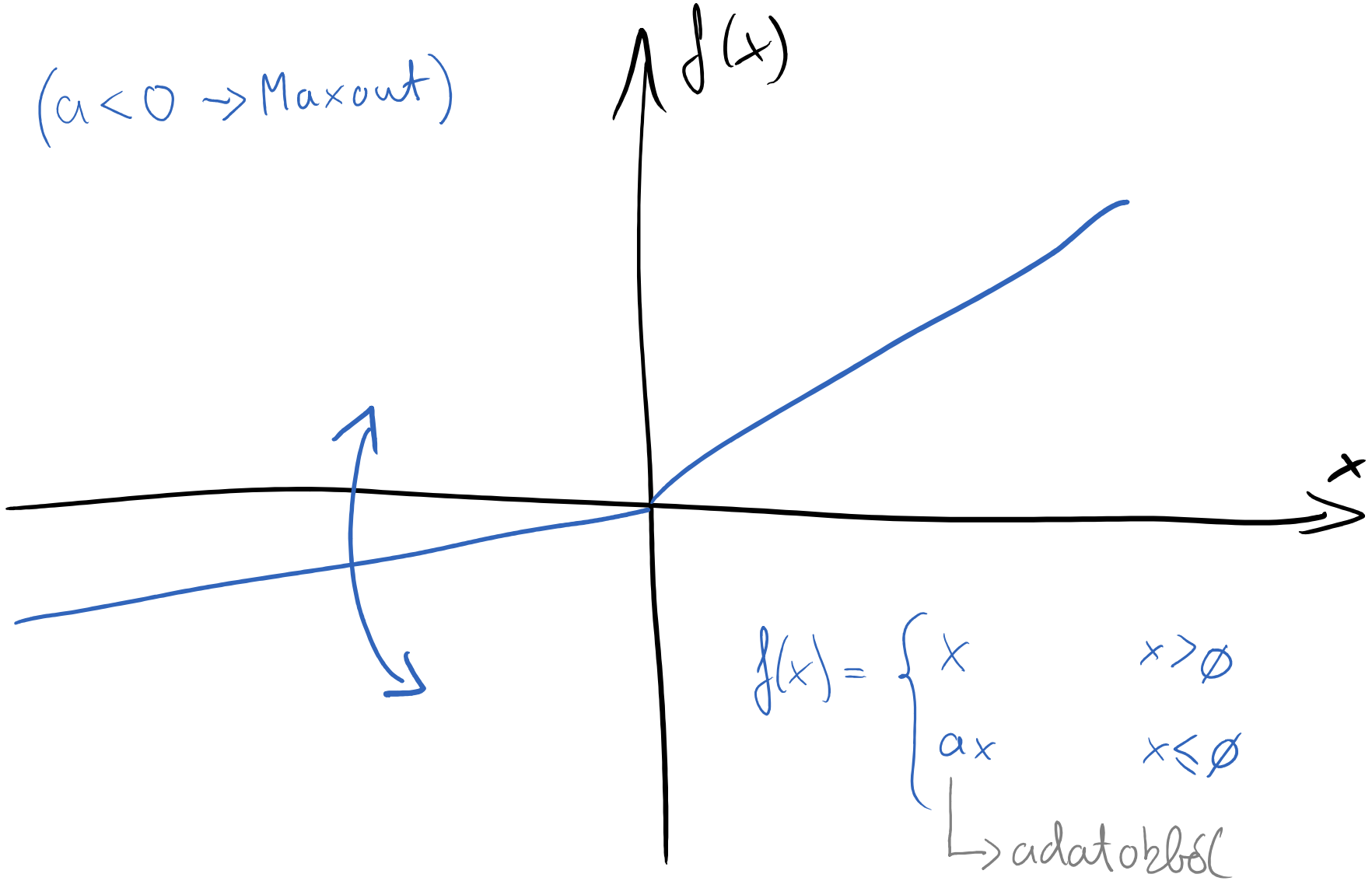
- Gyors
- Ritka aktiváció (sparse activation)
- Nincs szükség többé előtanításra
- Nincs „elenyésző” gradiens (*vanishing gradient*)
- Közelítés: softplus
- Kimeneti rétegben itt is figyelni kell!!!
- G. E. Hinton professzor nevéhez kötődik

Rectified Linear Unit



Leaky ReLU

Parametric ReLU



Súly inicializálás

Cél:

- Az aktivációk várható értéke legyen 0 körül
- Az aktivációk szórásnégyzete legyen kb. azonos

Miért kell:

- Ha túl kicsik a súlyok, nagyon kicsi lesz végül a jel
- Ha túl nagyok a súlyok, túl nagy lesz végül a jel

Súly inicializálás - alapgondolat

$$Y = WX$$

$$Var(WX) = E(X)^2 Var(W) + E(W)^2 Var(X) + Var(W) Var(X)$$

$$E(X)^2 = 0 \quad E(W)^2 = 0$$

$$Var(WX) = Var(W) Var(X)$$

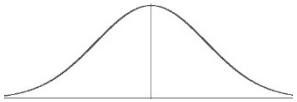
If W and X are i.i.d

$$Var(Y) = Var(WX) = n Var(W) Var(X) \Rightarrow Var(W) = \frac{1}{n}$$ ~ number of neurons (in or out)

Glorot-Xavier súly inicializálás

Glorot/Xavier normal/uniform inicializálás mély hálókhoz nem lineáris szimmetrikus aktivációs függvények esetén

$$W^{(i+1)} \sim U\left\{-\sqrt{\frac{6}{n^{(i)}+n^{(i+1)}}}\ ;\ +\sqrt{\frac{6}{n^{(i)}+n^{(i+1)}}}\right\}$$



$$W^{(i+1)} \sim N\left\{0\ ,\ \sqrt{\frac{2}{n^{(i)}+n^{(i+1)}}}\right\}$$

Glorot, X., & Bengio, Y. (2010, March). Understanding the difficulty of training deep feedforward neural networks. In Proceedings of 13th International Conference on Artificial Intelligence and Statistics (pp. 249-256).

Demo: http://www.deeplearning.ai/ai-notes/initialization/

Kaiming súly inicializálás

Aszimmetrikus aktivációs függvényekhez (pl. ReLU).

- n bemenet 0 várható értékkel és 1 szórással, súlyok 0 várható érték, 1 szórás, aszimmetrikus aktiváció
- A szórása a kimeneteknek kb. sqrt(n/2)

$$W^{(i)} \sim U\left\{-\frac{\sqrt{2}}{\sqrt{n^{(i)}}}; \frac{\sqrt{2}}{\sqrt{n^{(i)}}}\right\} \qquad \sim N\left\{0; \frac{\sqrt{2}}{\sqrt{n^{(i)}}}\right\}$$

He, Kaiming et. al. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1026-1034).

Osztályozás

- Osztályozás
  - SoftMax + kereszt-entrópia
  - Bináris keresztentrópia

- Kiértékelési módszerek, metrikák
  - Konfúziós (tévesztési) mátrix
  - Precision, Recall, Accuracy, F1, MAP, stb.

Osztályozás vs regresszió

- Regresszió: sigmoid / tanh / ReLU + MSE
- Osztályozás: softmax + kereszt-entrópia
- Költségfüggvény: kereszt-entrópia

$$C = -\sum_{n=1}^{M}\sum_{i=1}^{K} y_i^{(n)} \cdot \ln\left(\hat{y}_i^{(n)}\right)$$

M: minták száma
K: osztályok száma

- SoftMax: számokat „*valószínűségekké*” alakít, amik összege 1

$$y_i(x) = \frac{e^{x_i}}{\sum_{k=1}^{K} e^{x_k}} \qquad i = 1 \dots K$$

Kereszt-entrópia osztályozásra 1

Y_target (OneHot)			Y_predicted (valószínűség)			Osztály	Helyes?
1	0	0	0.1	0.1	0.8	Autó	nem
0	1	0	0.3	0.4	0.3	Rakéta	Igen
0	0	1	0.3	0.3	0.4	Repülő	Igen

Osztályozási pontosság: $\frac{2}{3} = 0.6$

Keresztentrópia: $-\{(\ln(0.1)\cdot 1)+(\ln(0.1)\cdot 0)+(\ln(0.8)\cdot 0)\} = -\ln(0.1)$

$-\{(\ln(0.3)\cdot 0)+(\ln(0.4)\cdot 1)+(\ln(0.3)\cdot 0)\} = -\ln(0.4)$

$-\{(\ln(0.3)\cdot 0)+(\ln(0.3)\cdot 0)+(\ln(0.4)\cdot 1)\} = -\ln(0.4)$

Átlagos keresztentrópia: $\frac{-(\ln(0.1)+\ln(0.4)+\ln(0.4))}{3} \approx 1.38$

MSE: $(0.1-1)^2+(0.1-0)^2+(0.8-0)^2 = 1.46$

$(0.3-0)^2+(0.4-1)^2+(0.3-0)^2 = 0.54$

$(0.3-0)^2+(0.3-0)^2+(0.4-1)^2 = 0.54$

$\frac{1.46+0.54+0.54}{3} \approx 0.847$

Kereszt-entrópia osztályozásra 1

Y_target (OneHot)			Y_predicted (valószínűség)			Osztály	Helyes?
1	0	0	0.3	0.4	0.3	Autó	nem
0	1	0	0.1	0.8	0.1	Rakéta	Igen
0	0	1	0.1	0.1	0.8	Repülő	Igen

Osztályozási pontosság: $\frac{2}{3} = 0.6$

Kereszt-entrópia: $-\{(\ln(0.3)\cdot 1)+(\ln(0.4)\cdot 0)+(\ln(0.3)\cdot 0)\} = -\ln(0.3)$

$-\{(\ln(0.1)\cdot 0)+(\ln(0.8)\cdot 1)+(\ln(0.1)\cdot 0)\} = -\ln(0.8)$

$-\{(\ln(0.1)\cdot 0)+(\ln(0.1)\cdot 0)+(\ln(0.8)\cdot 1)\} = -\ln(0.8)$

Átlagos kereszt-entrópia: $\frac{-(\ln(0.1)+\ln(0.8)+\ln(0.8))}{3} \approx 0,91$

MSE:
$(0.3-1)^2+(0.4-0)^2+(0.3-0)^2 = 0.74$
$(0.1-0)^2+(0.8-1)^2+(0.1-0)^2 = 0.06$
$(0.1-0)^2+(0.1-0)^2+(0.8-1)^2 = 0.06$

$\frac{0.74+0.06+0.06}{3} = 0,286$

Kiértékelési módszerek

- Konfúziós (tévesztési) mátrix (TP, TN, FP, FN)
- Osztályozás metrikák (Precision, Recall)

- ROC
- AUC
- AUROC
- RECALL@k
- Precision@k
- F1

Konfúziós-mátrix

- Könnyen olvasható, sok értékes eredmény van rajta.
- Például 3 osztály:

		Y_predicted		
	Osztály	Rakéta	Autó	Repülő
Y_target	Rakéta	**15**	**3**	**8**
	Autó	**4**	**23**	**1**
	Repülő	**9**	**2**	**19**

Konfúziós-mátrix: osztályonként

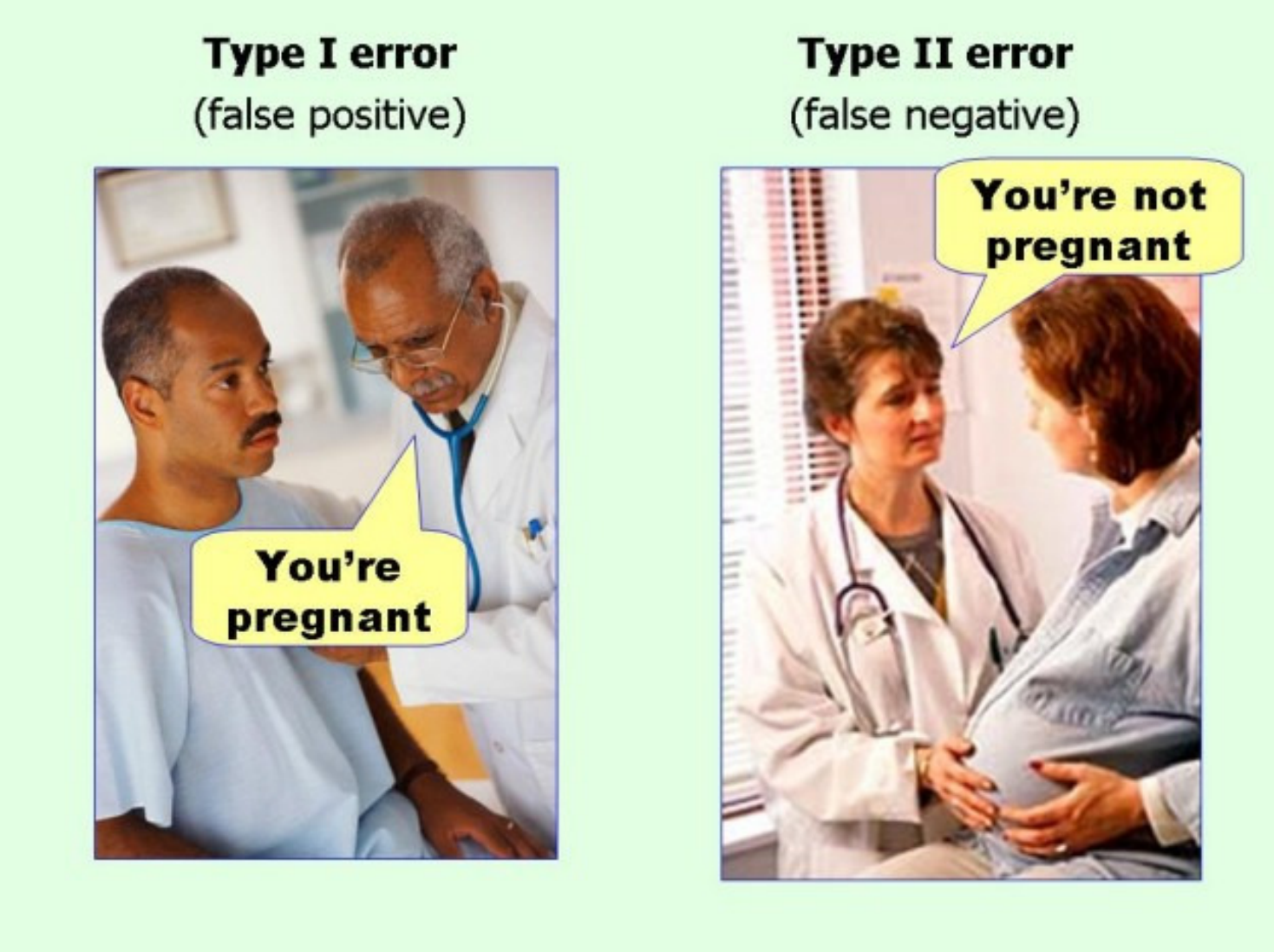
Rakéta

		Y_predicted	
	Kondíció	Pozitív	Negatív
Y_target	Pozitív	15 True Positive (TP)	3+8=11 False Negative (FN)
	Negatív	4+9=13 False Positive (FP)	23+19+1+2=45 True Negative (TN)

Pl. lelőjük, ha rakéta! (védjük a várost)

- TP: rakétának hittük és az volt
- TN: úgy gondoltuk nem rakéta és tényleg nem volt az
- FP: úgy gondoltuk, hogy rakéta és nem az volt ☹
- FN: úgy gondoltuk nem rakéta és az volt ☠

Első és másodfajú hiba



Kiértékelési módszerek

- Precision $= \frac{TP}{TP + FP}$ (=precizitás)
- Recall $= \frac{TP}{TP + FN}$ (=felidézés)
- Accuracy $= \frac{TP + TN}{TP + TN + FP + FN}$ (=pontosság)
- F1 $= \frac{2TP}{2TP + FP + FN} = 2 \cdot \frac{\text{PRECISION} \cdot \text{RECALL}}{\text{PRECISION} + \text{RECALL}}$

- Továbbiak: MAP, ROC, AUROC, stb.

Konvolúciós rétegek

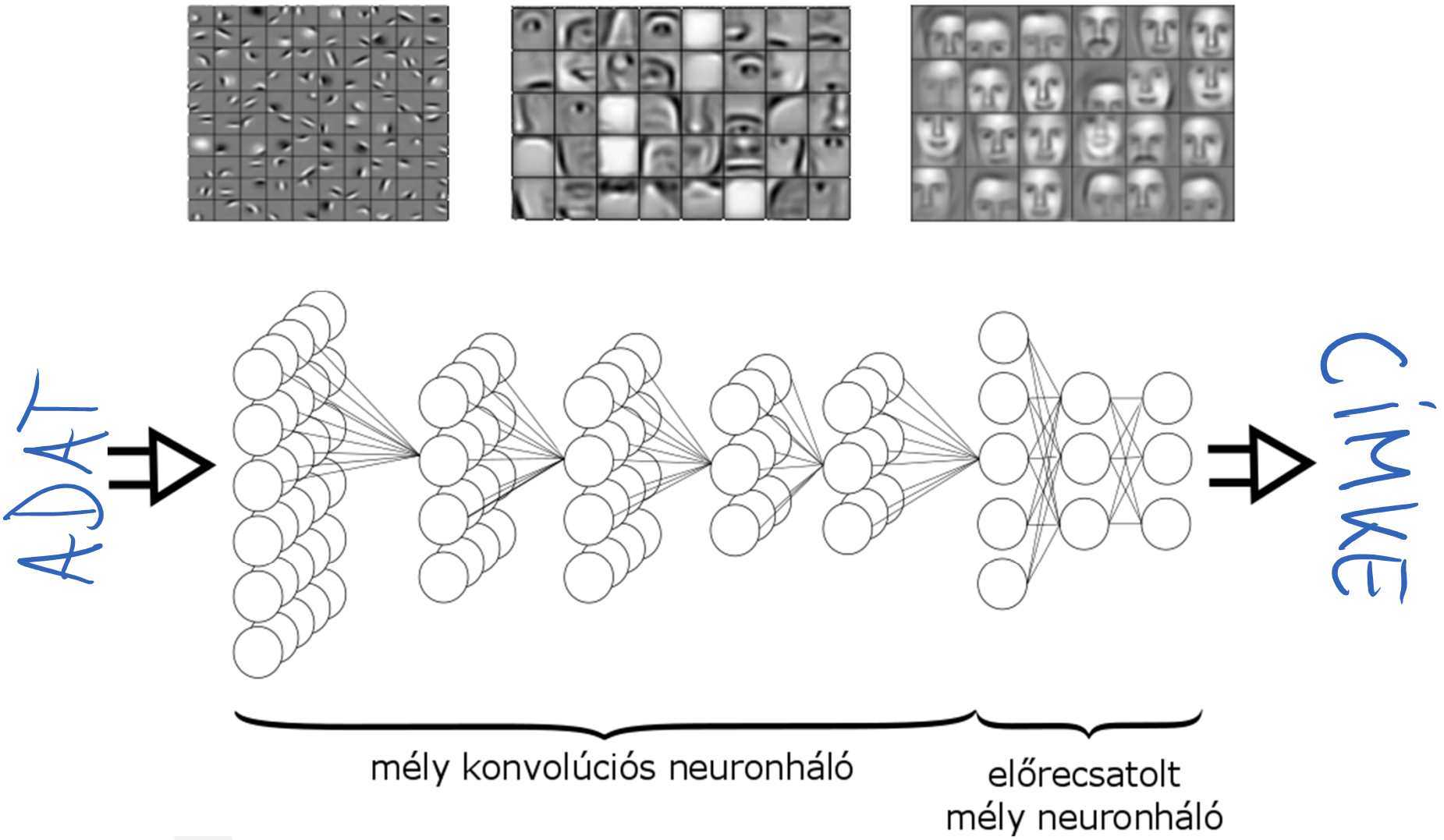
Főbb mérföldkövek

- LeNet: LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, *3361*(10), 1995.
- LeNet5: LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.
- AlexNet: Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).
- … és elindult a CNN lavina!

De mi is az a CNN?

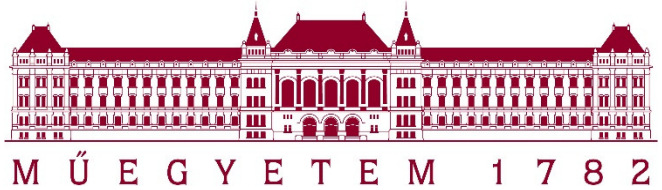
- Convolutional Neural Network (CNN)
  - Konvolúciós és Pooling rétegek és aktivációs függvények struktúrált összessége.
  - Közöttük számos további réteg lehet, pl. DropOut, BatchNorm, stb.
  - A hálózat utolsó rétegei jellemzően előrecsatolt rétegek, melyek a megtanult jellemzők osztályokba sorolását végzik.
- Gyors és hatékony megoldás *jellemző tanulásra* (*feature learning*). Térbeli függőség a minták között.
- Jellemző tanulás és modellezés egy lépésben.
- Hang, kép, szöveg, videó és általános felhasználási terület (pl. AlphaGo, szenzoradatok).

CNN háló általános felépítése



Névnek a NEPTUN kódodat add meg!

https://kahoot.it/



Köszönöm a figyelmet!

toth.b@tmit.bme.hu

SmartLab
Intelligent Interactions

http://smartlab.tmit.bme.hu

NVIDIA
GPU
EDUCATION
CENTER