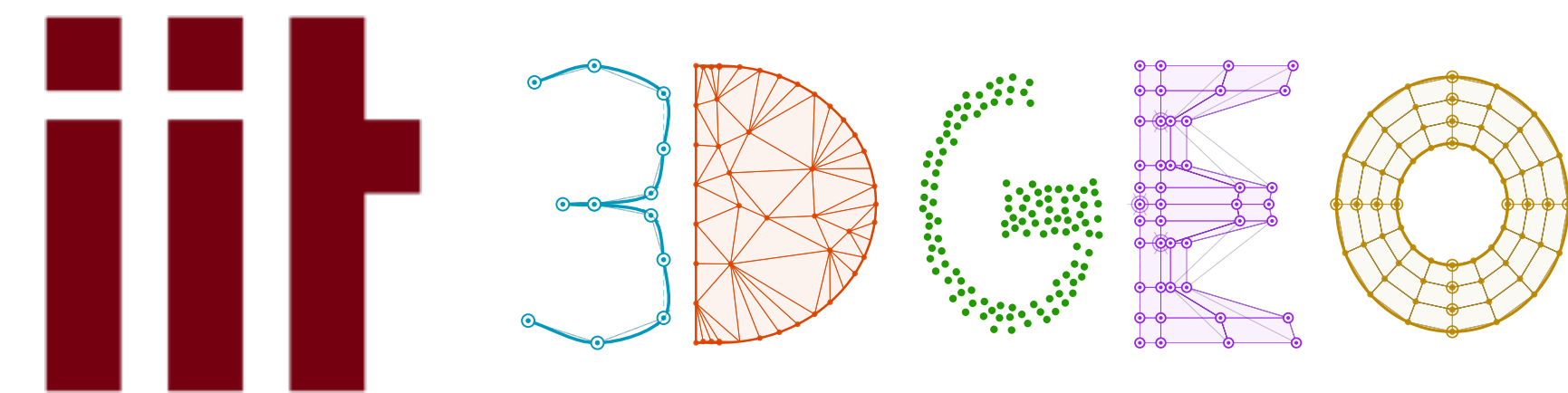
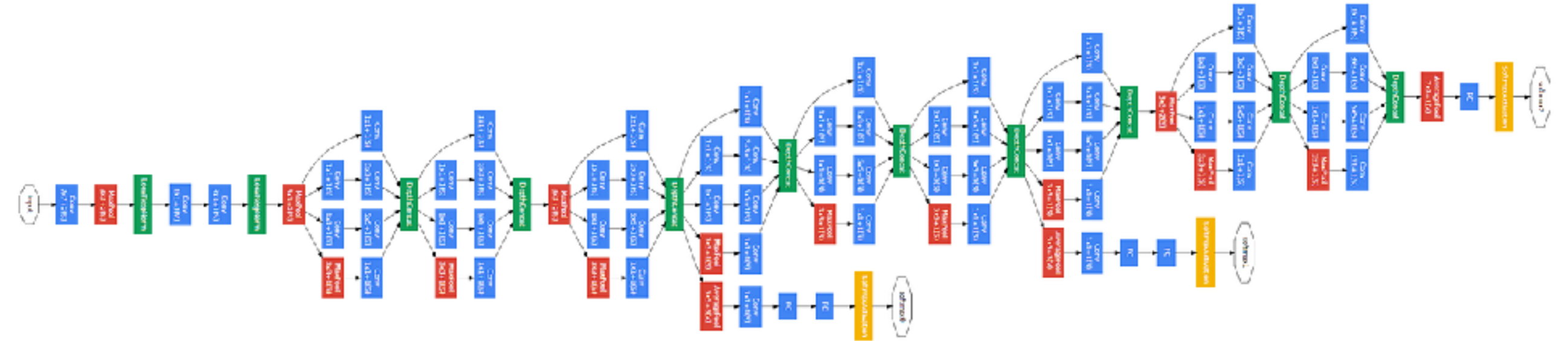


3. Előadás: Neurális Hálózatok

Generatív AI és Inverz Módszerek a Képszintézisben
BME-VIK IIT, 2026

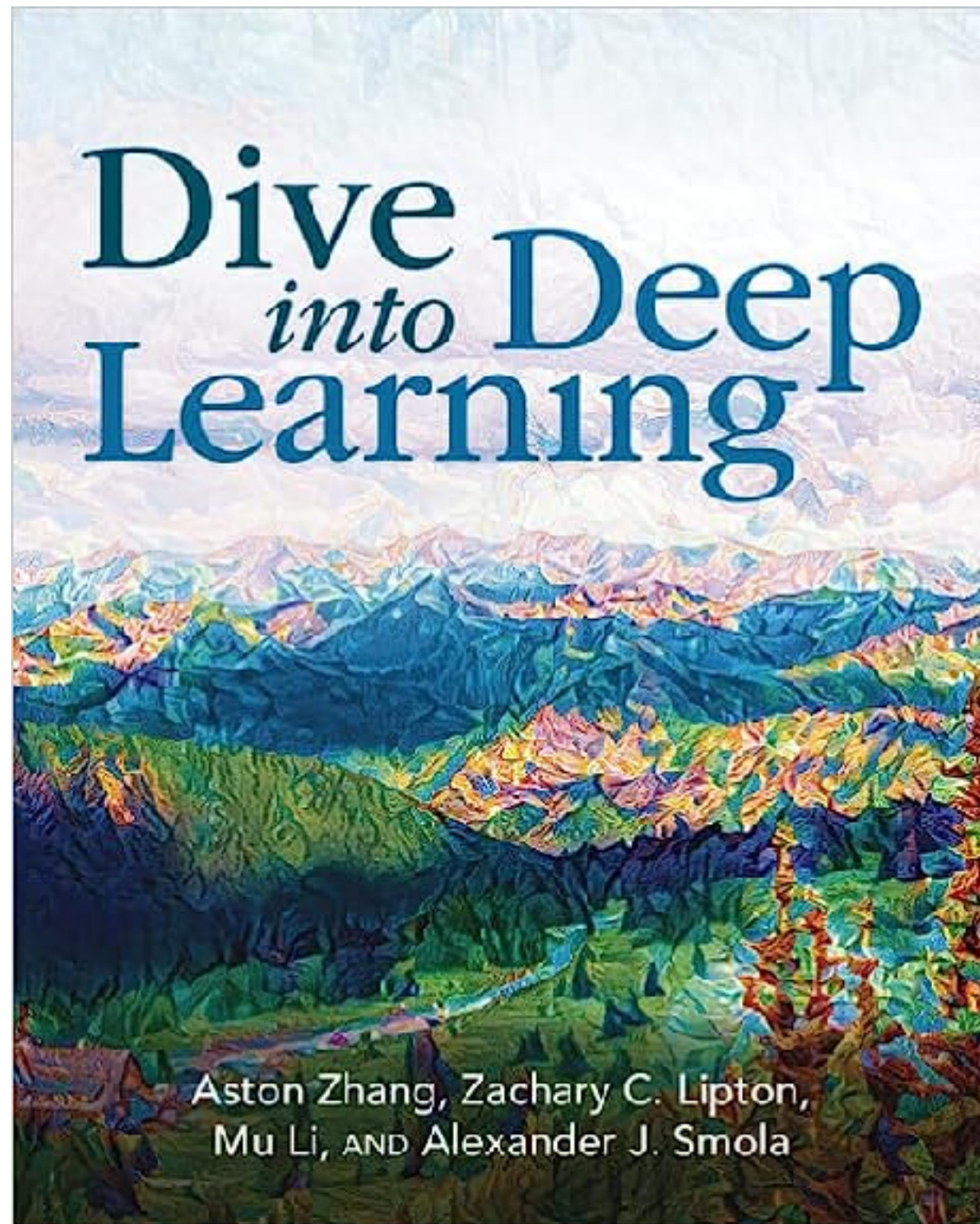


Dr. Vaitkus Márton

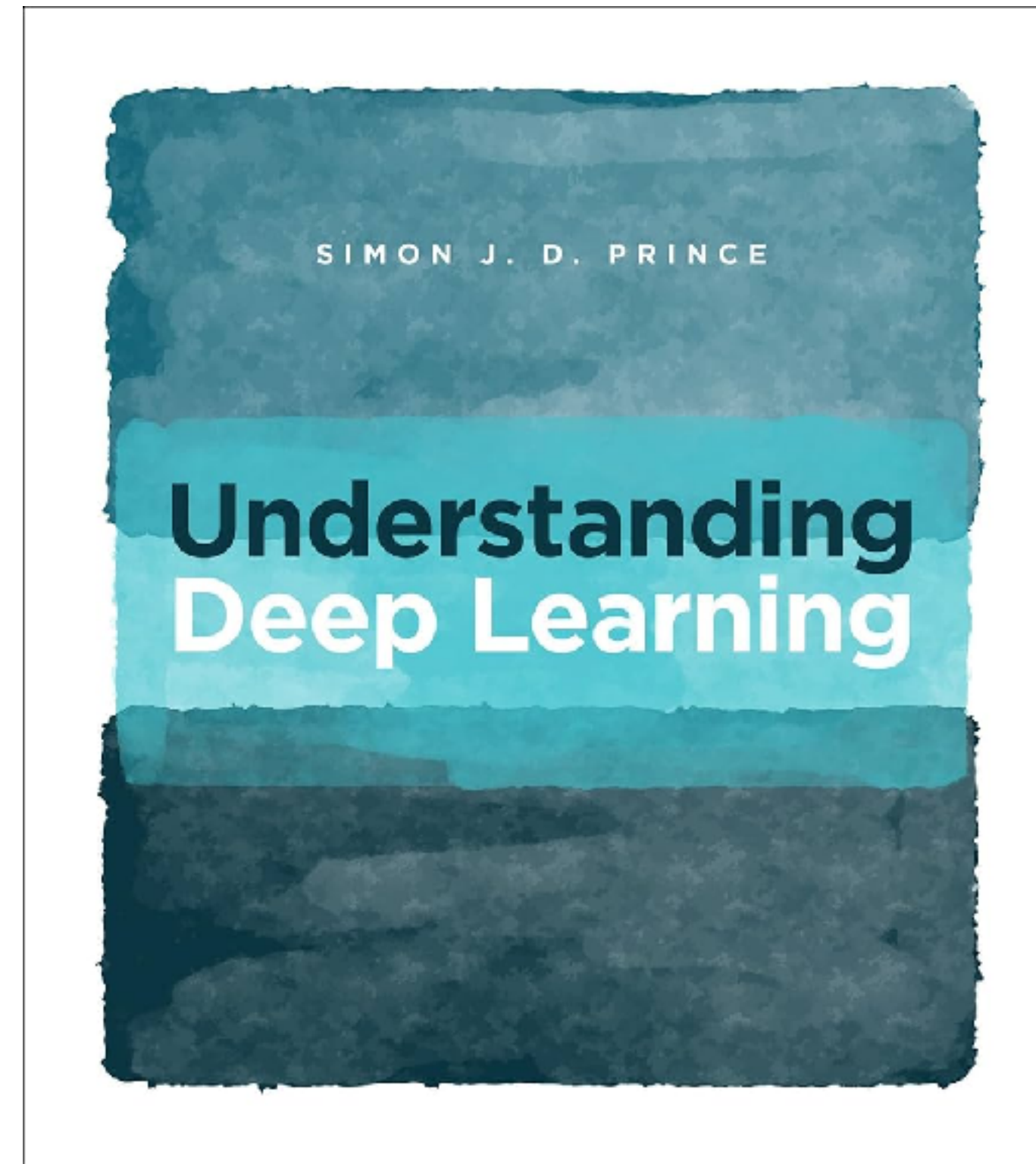


Neurális Hálózatok

Könyvajánló

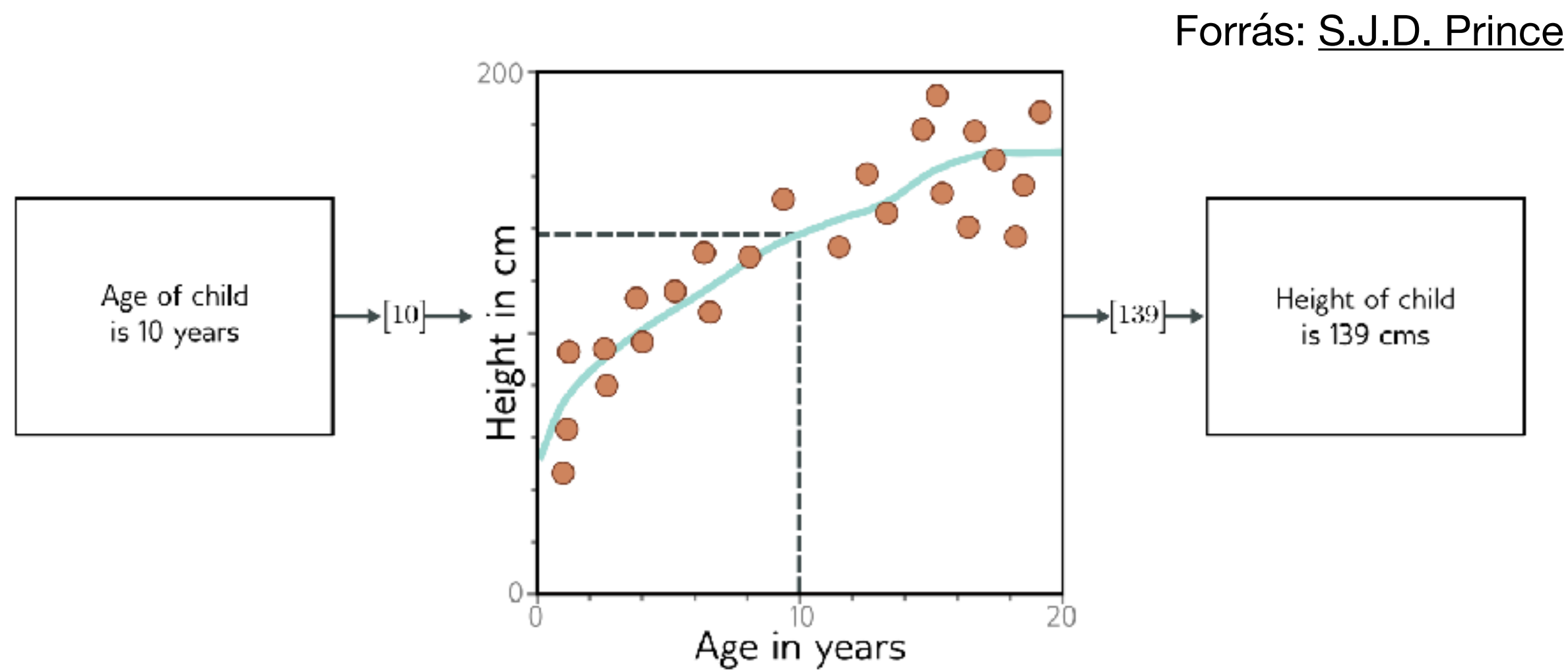


<https://d2l.ai/>

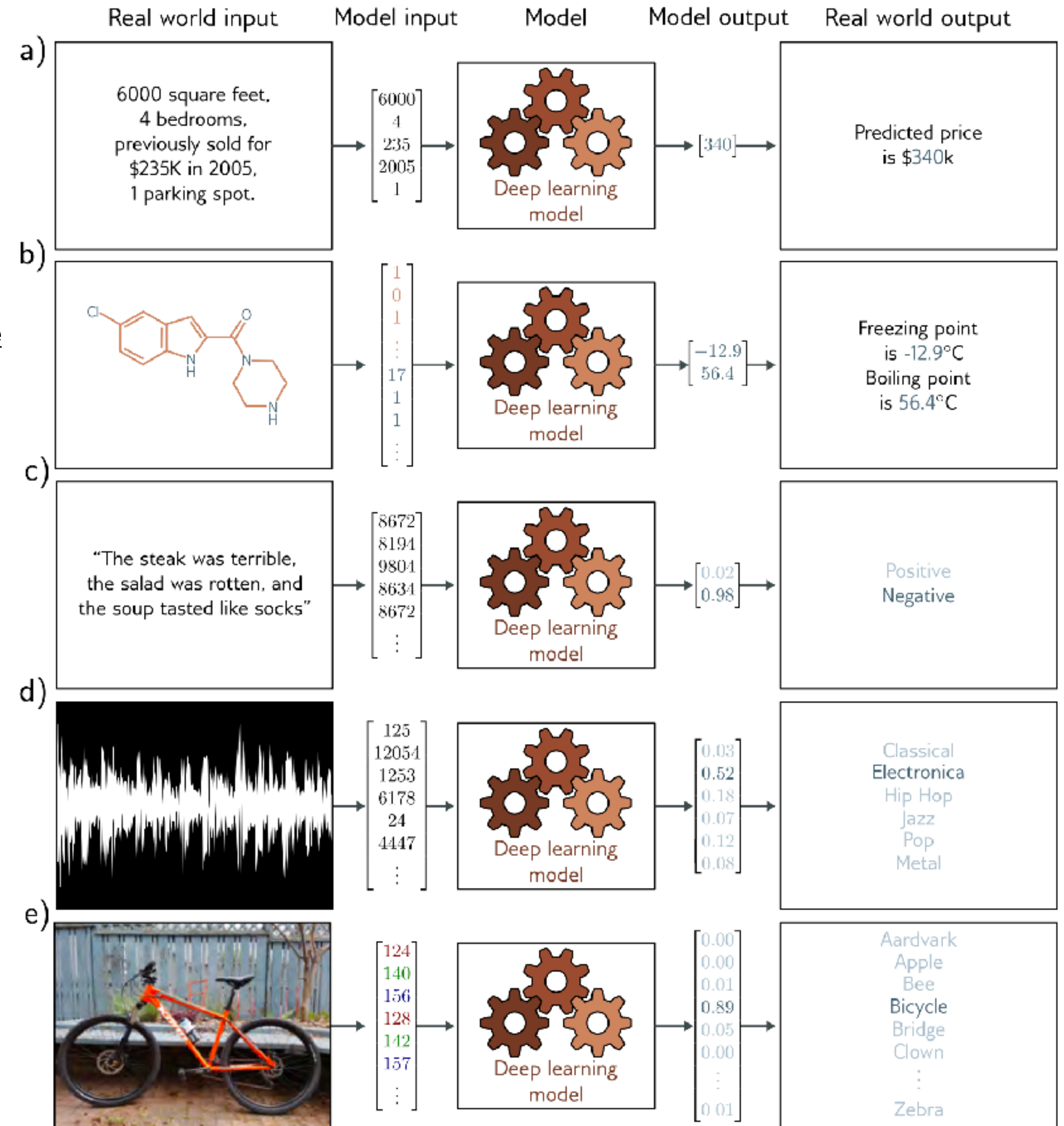


<https://udlbook.github.io/udlbook/>

Felügyelt Tanítás

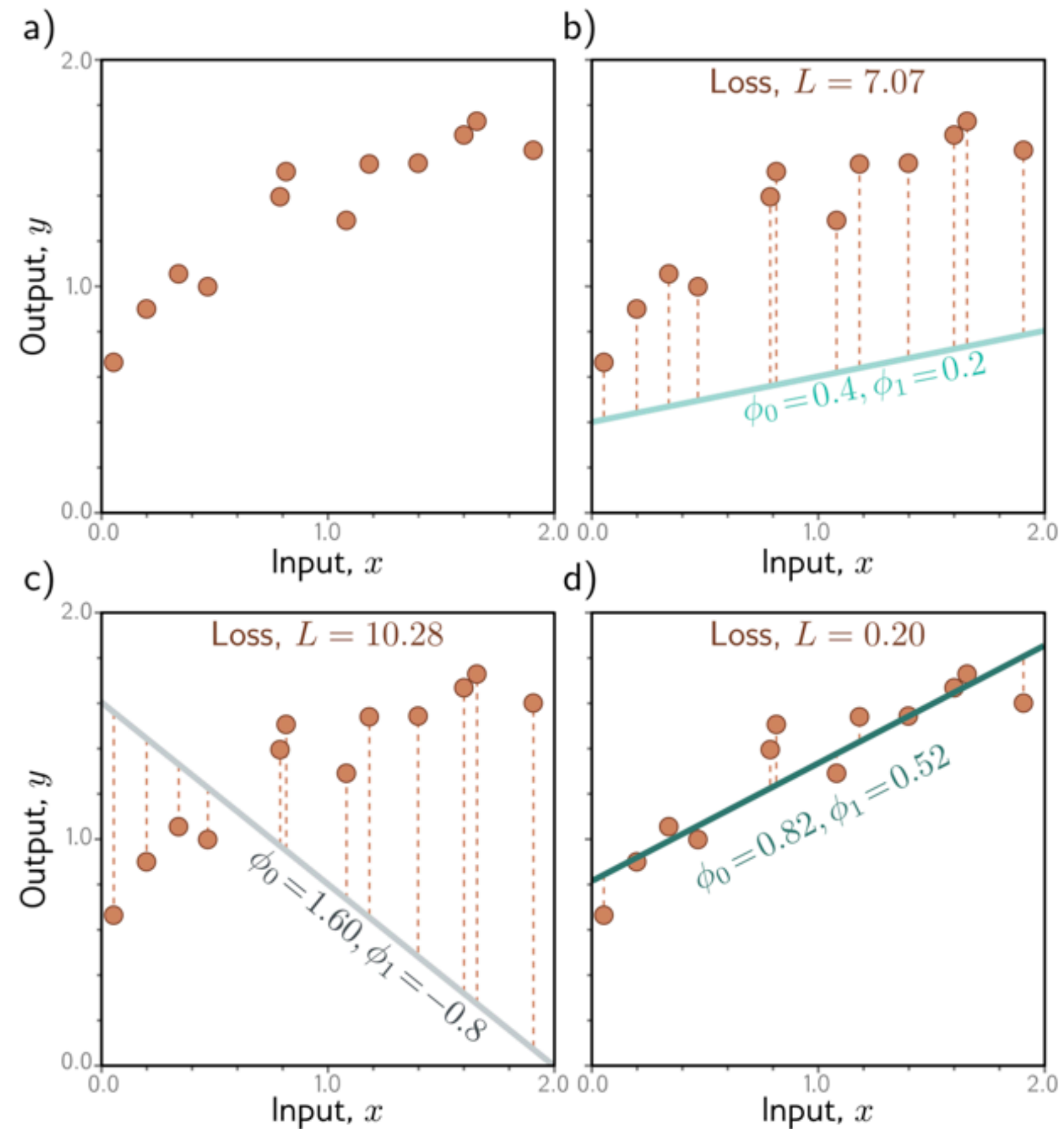


Bemenet-kimenet közötti összefüggés
modellje: $y = f[x_i, \phi]$



Felügyelt Tanítás

Lineáris regresszió

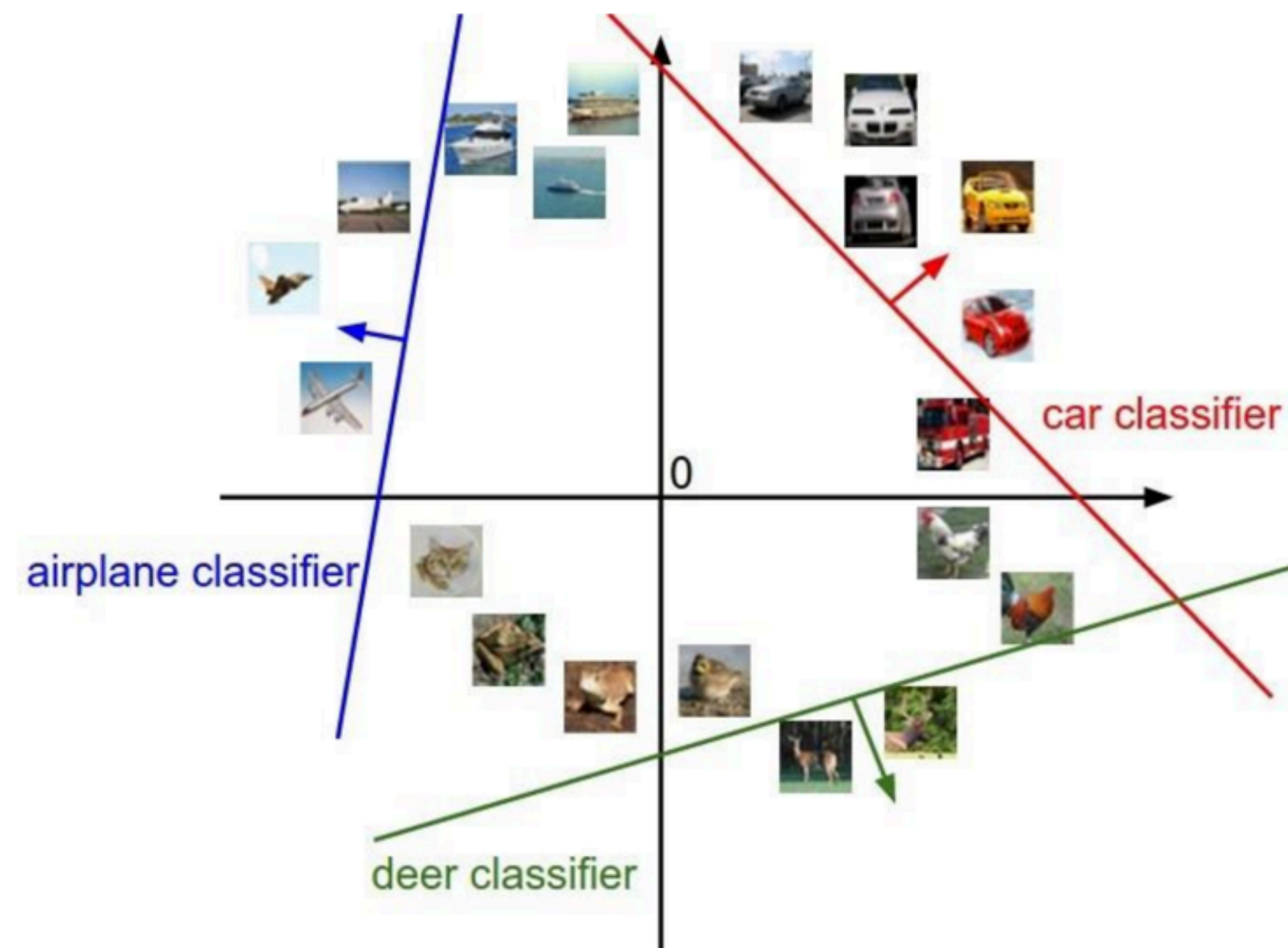


Forrás: [S.J.D. Prince](#)

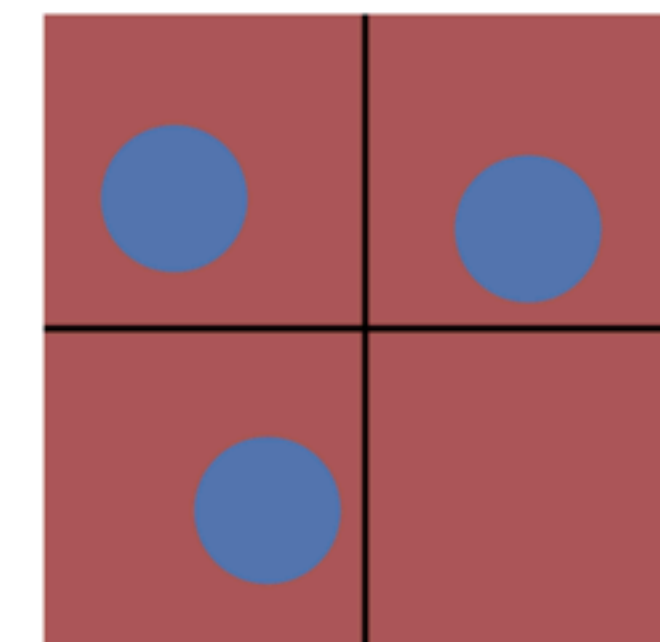
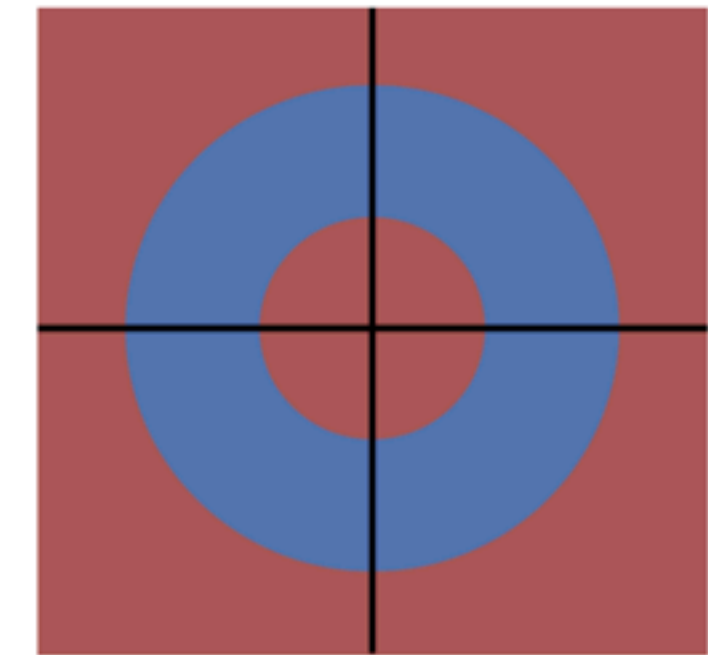
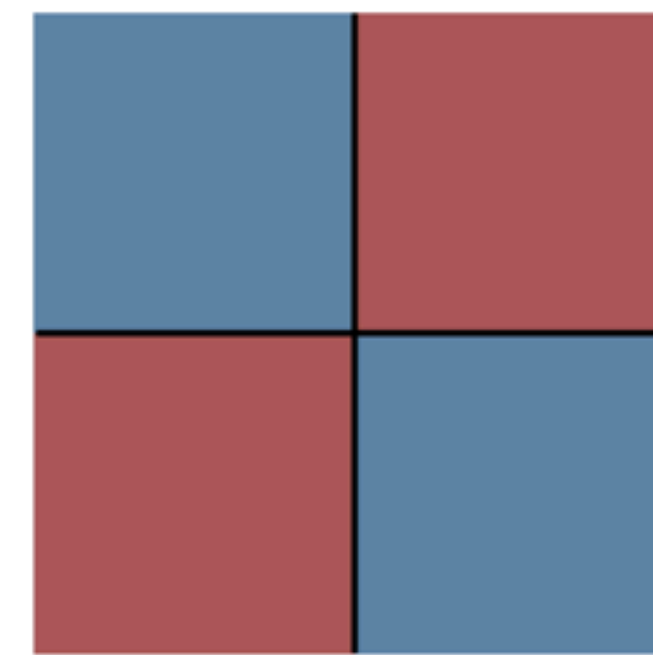
$$\hat{\phi} = \underset{\phi}{\operatorname{argmin}} [L[\phi]]$$
$$= \underset{\phi}{\operatorname{argmin}} \left[\sum_{i=1}^I (f[x_i, \phi] - y_i)^2 \right]$$

Felügyelt Tanítás

Osztályozási feladatok



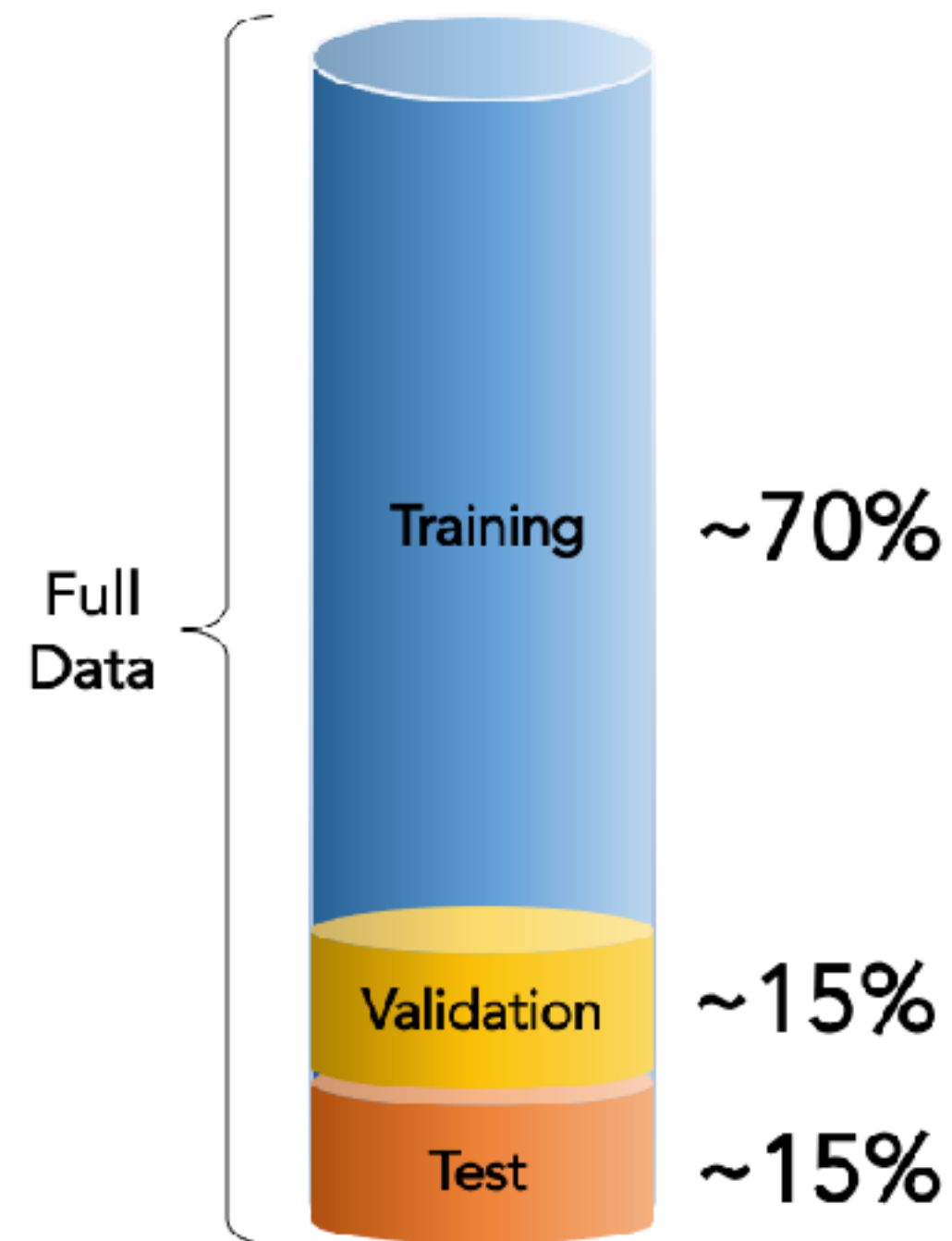
Lineáris osztályozás



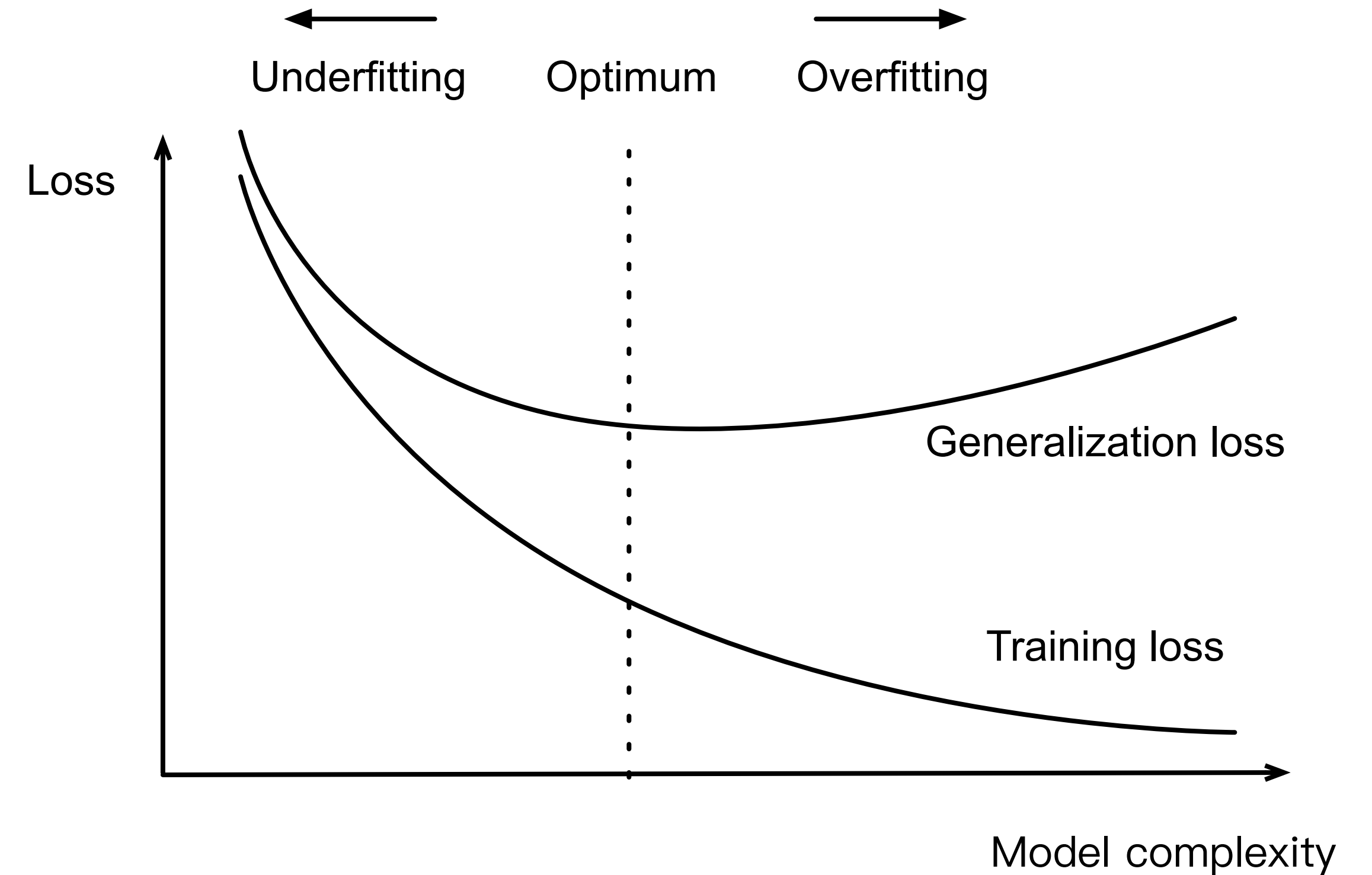
Ezeket hogyan osztályozzuk?

Felügyelt Tanítás

Tanítás vs. Tesztelés



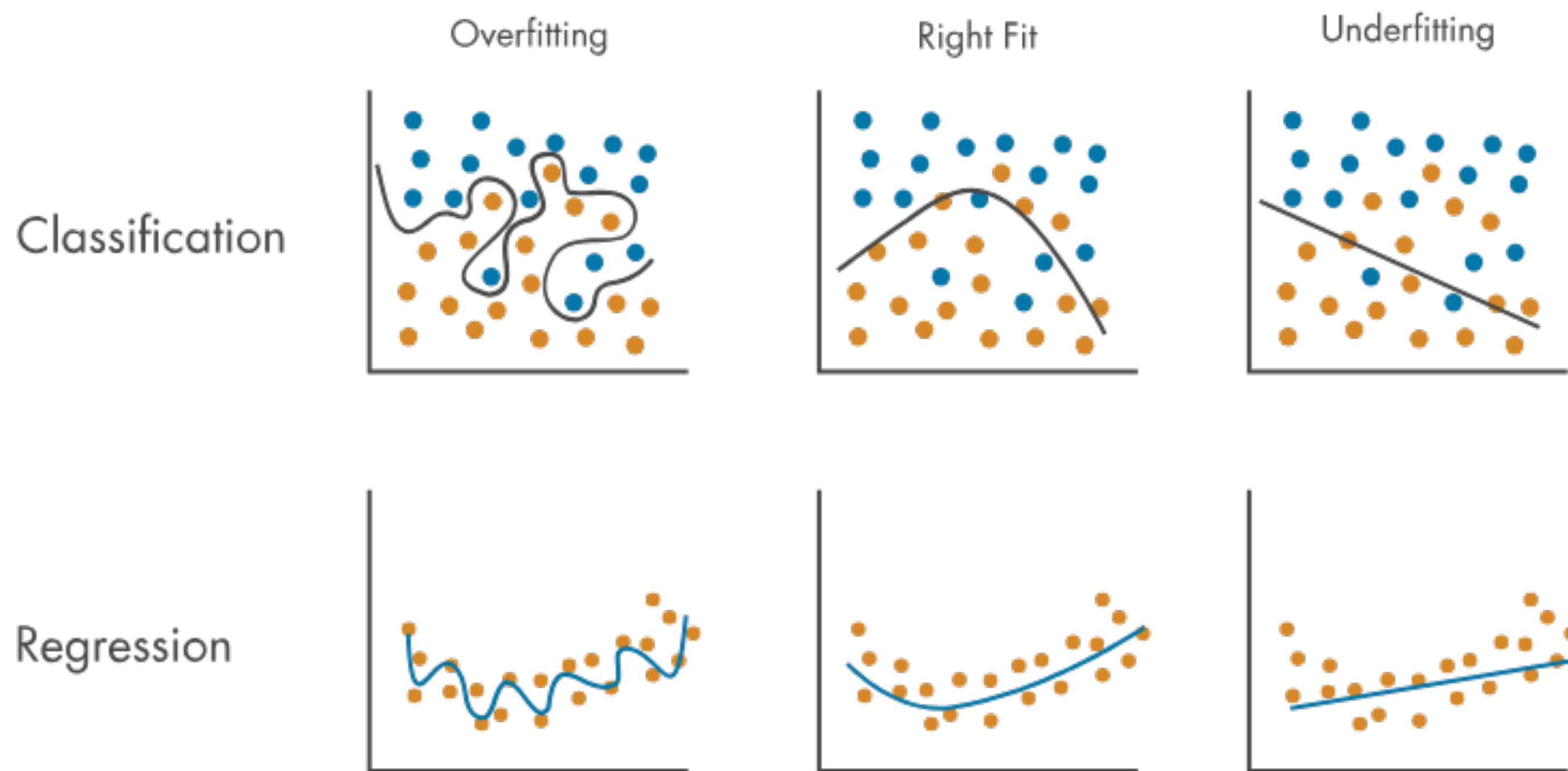
Cél: a tanító adat alapján **általánosítani** teszt adatokra



Elméletileg: bonyolultabb modell – rosszabb általánosítás!

Felügyelt Tanítás

Underfitting vs. Overfitting



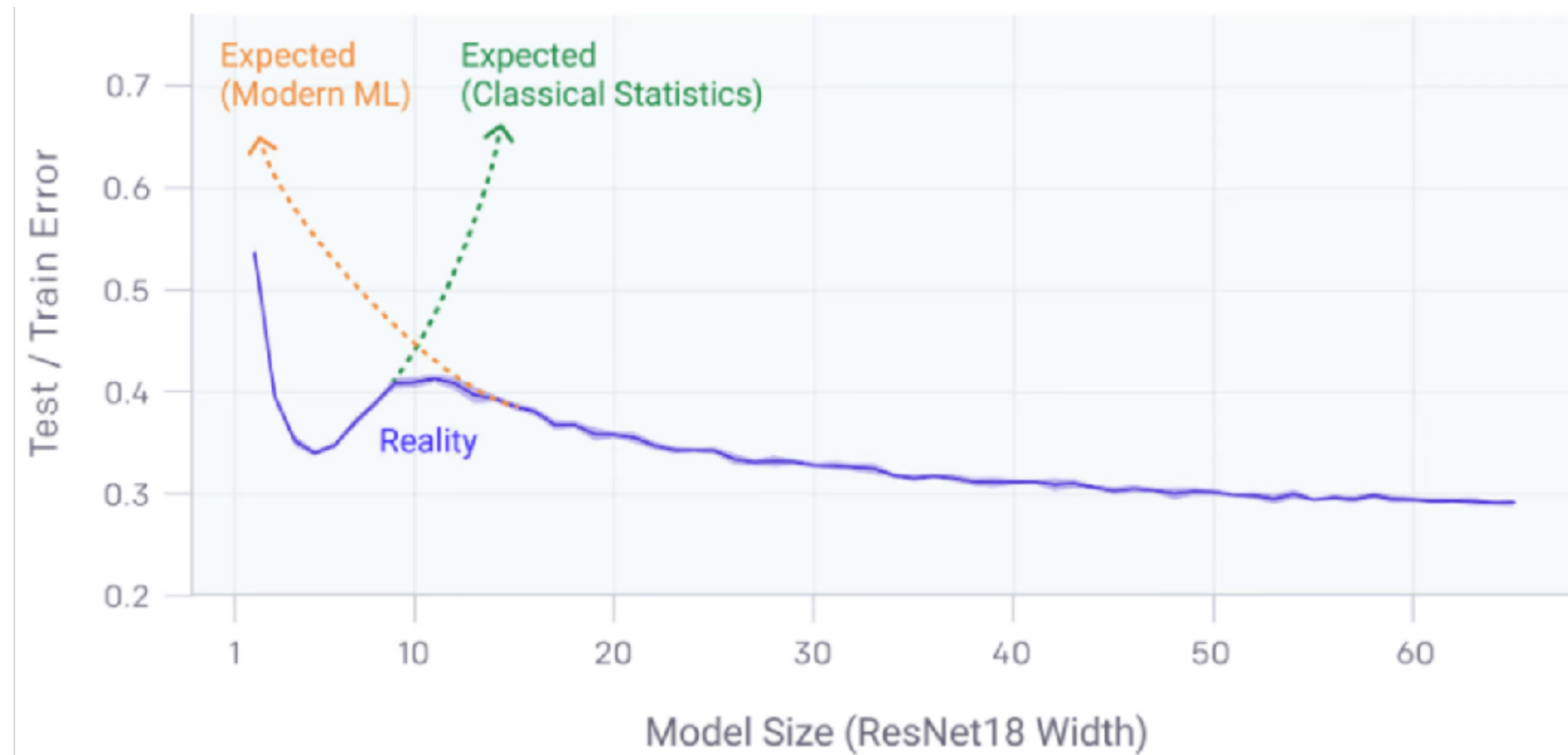
Túl kevés paraméter: csak nagy hibával tudunk illeszteni a tanító adatokra is (**underfitting**)

Túl sok paraméterrel: tanító adatokra akár *tökéletes*, de általában már rosszabb illeszkedés (**overfitting**)

Egyszerű megoldás jobban általánosítható — a problémát általában **regularizálni** szükséges!

Felügyelt Tanítás

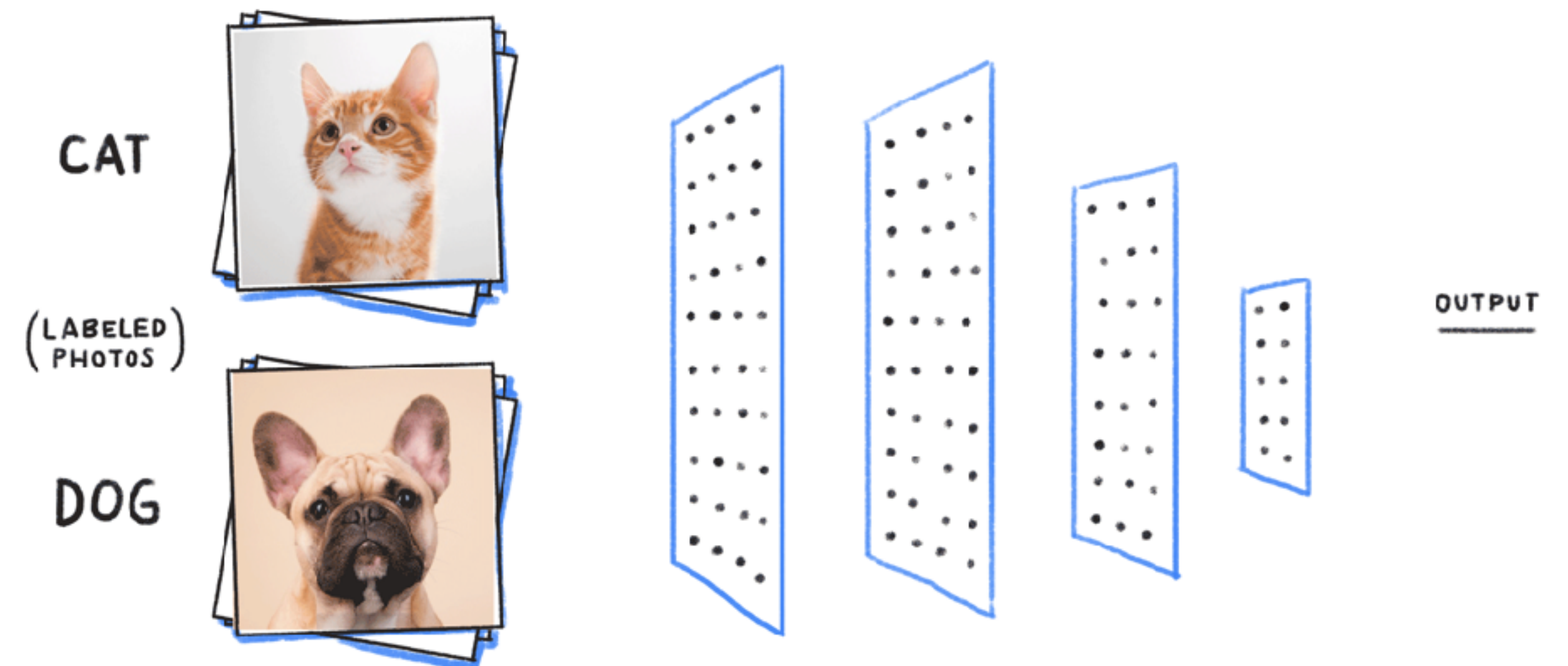
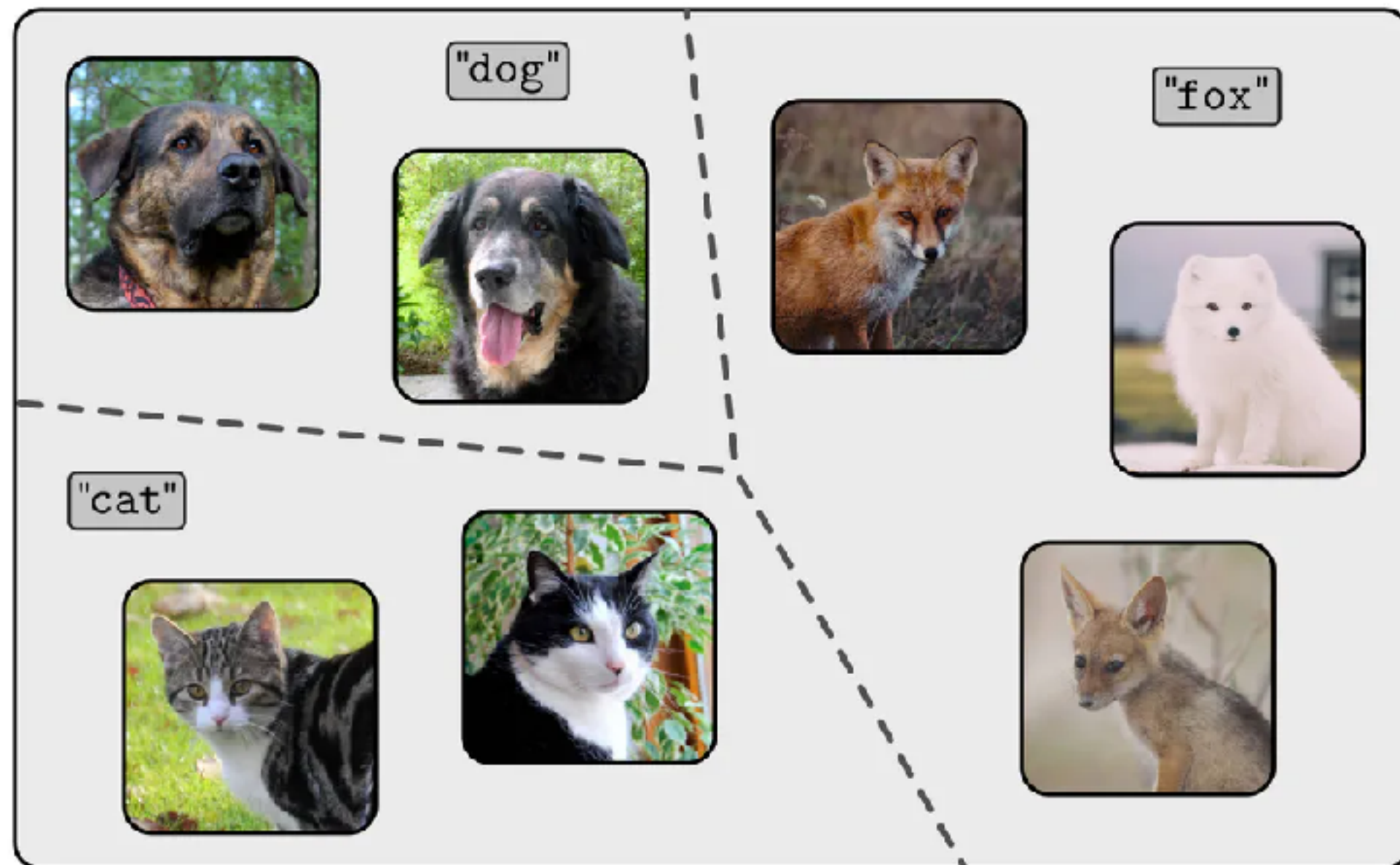
“Double Descent” jelenség



A deep learning forradalom felismerése: minél több a paraméter, annál kisebb lehet a hiba!
(Később visszatérünk rá...)

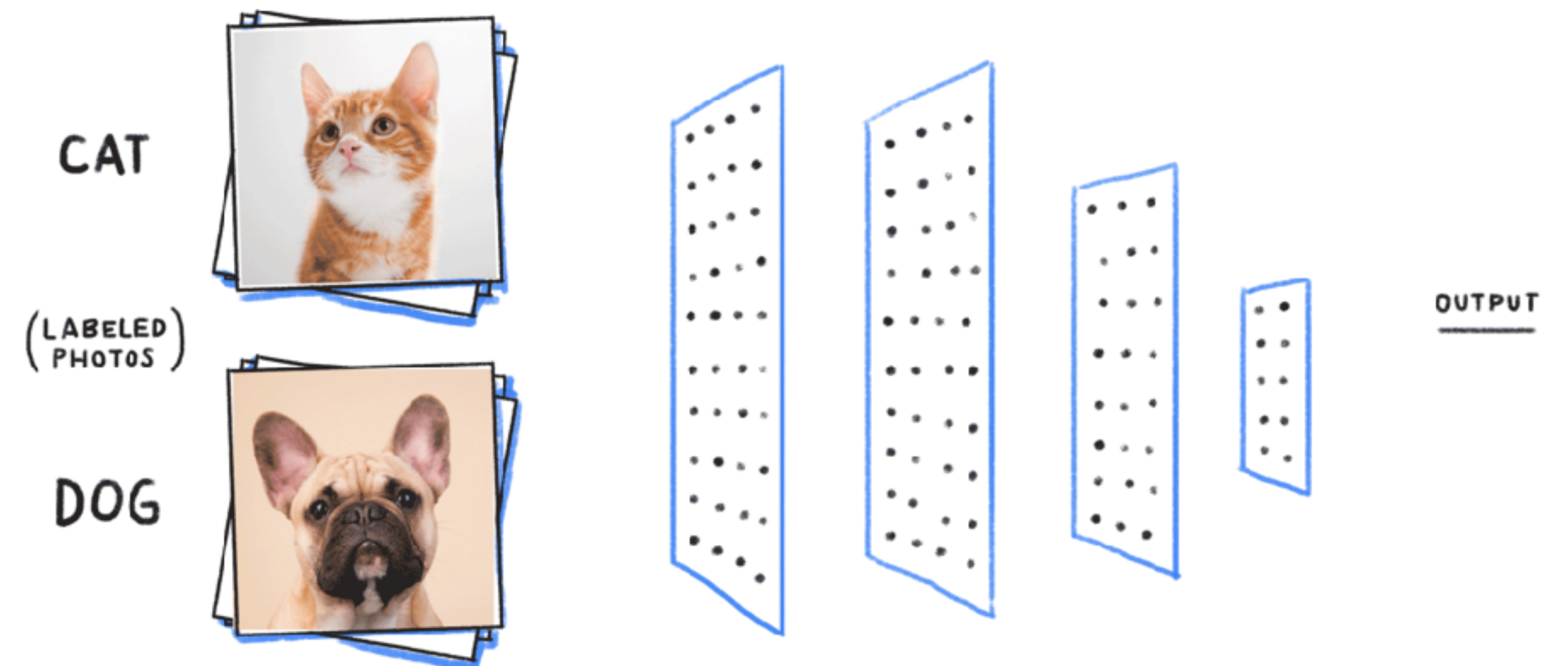
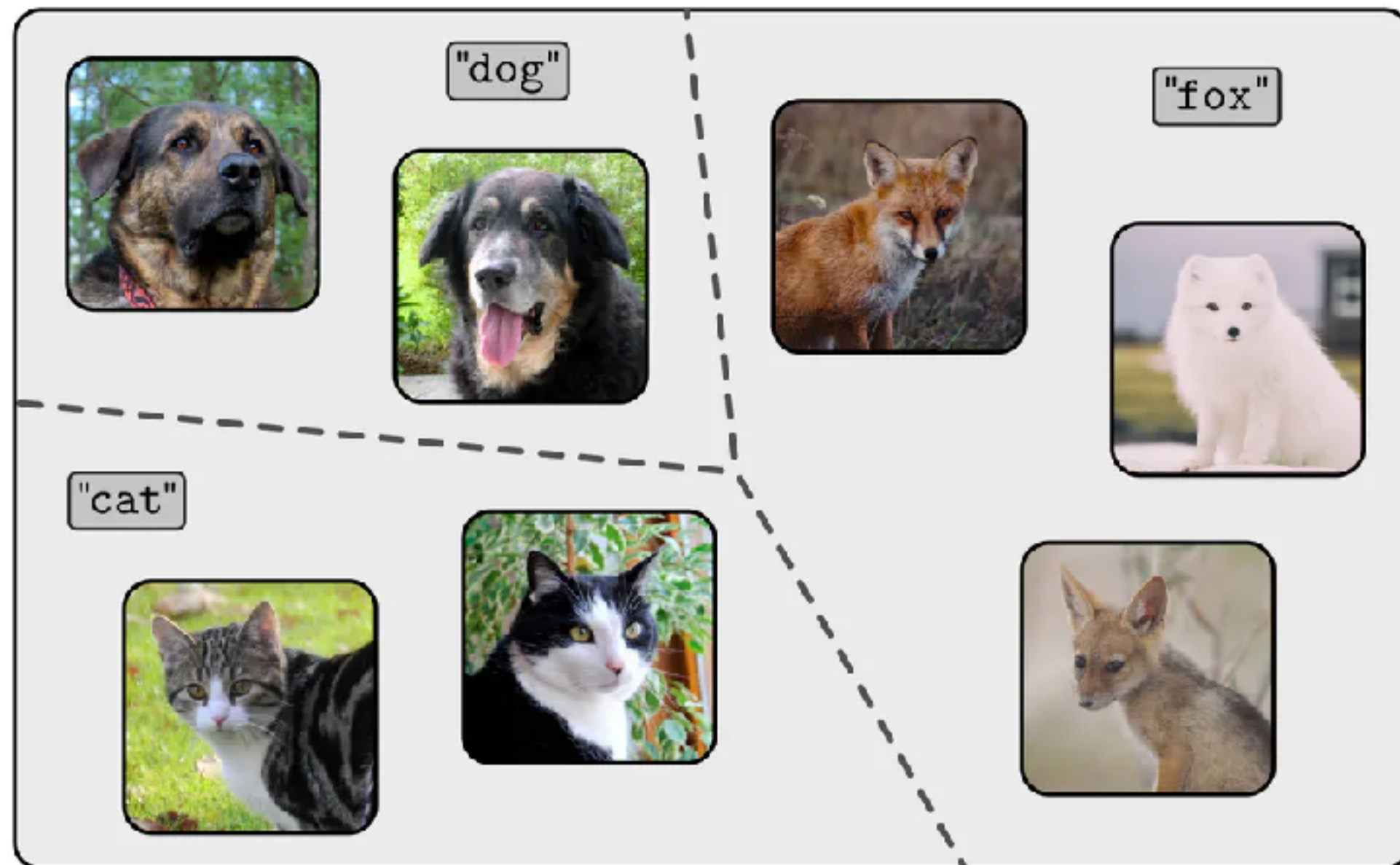
Felügyelt Tanítás

Képosztályozás



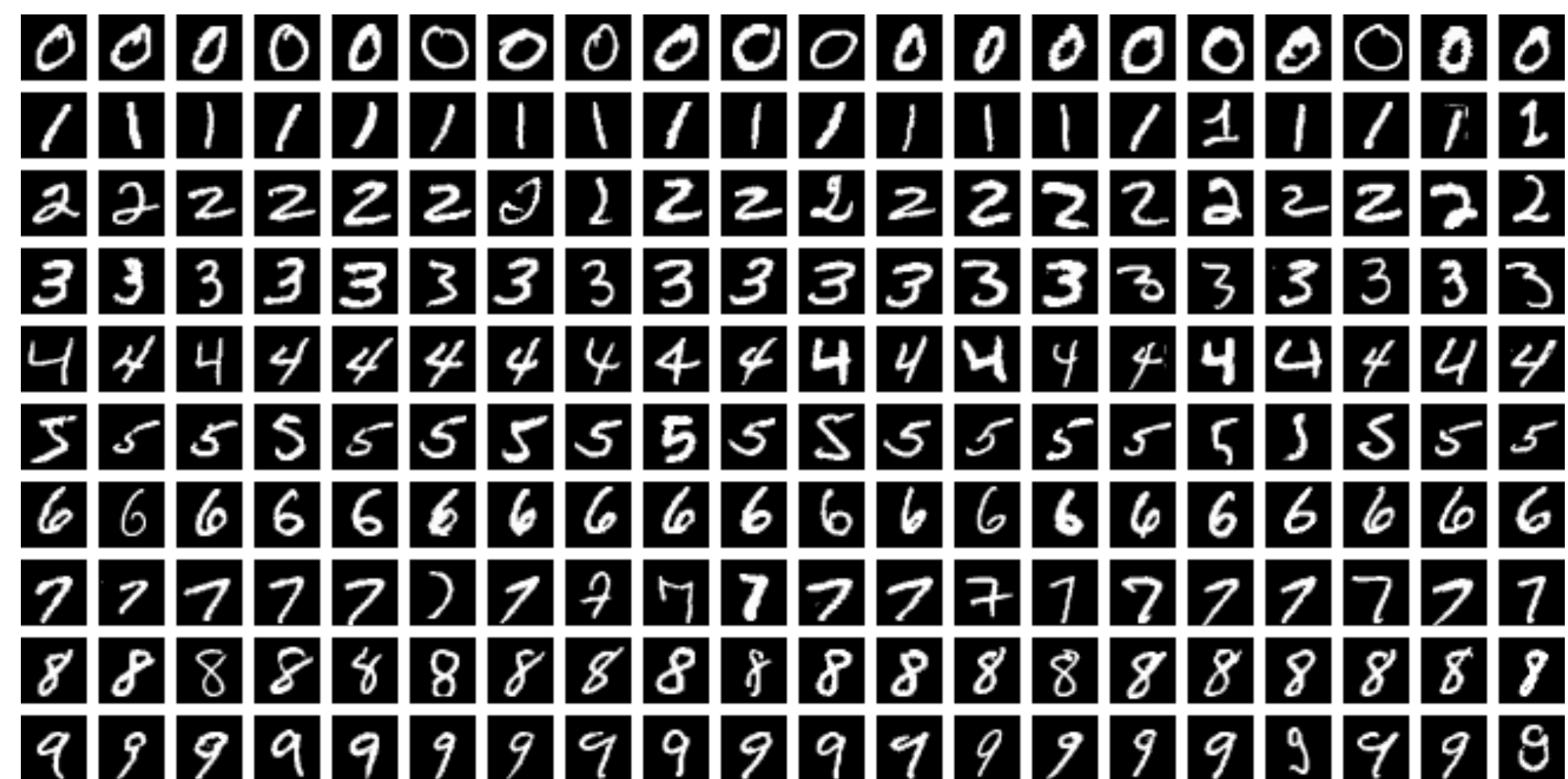
Felügyelt Tanítás

Képosztályozás



Felügyelt Tanítás

Képosztályozás – Adathalmazok



28 x 28, szürkeárnyaltos
60000 (train) + 10000 (test)
10 osztály

MNIST

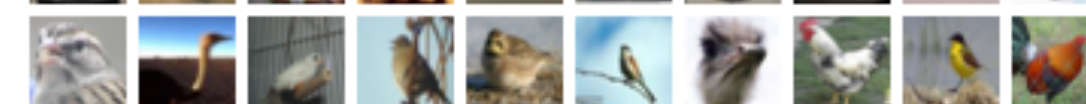
airplane



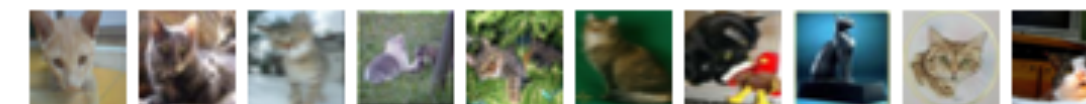
automobile



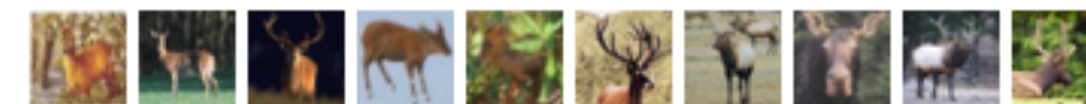
bird



cat



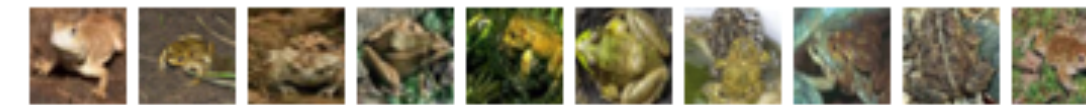
deer



dog



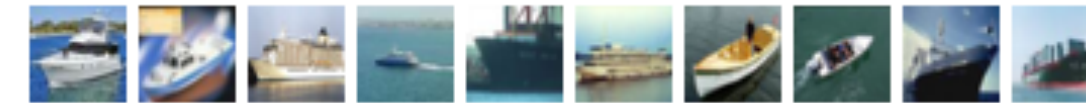
frog



horse



ship



truck



32 x 32, színes
50000 (train) + 10000 (test)
10 osztály

CIFAR-10

Felügyelt Tanítás

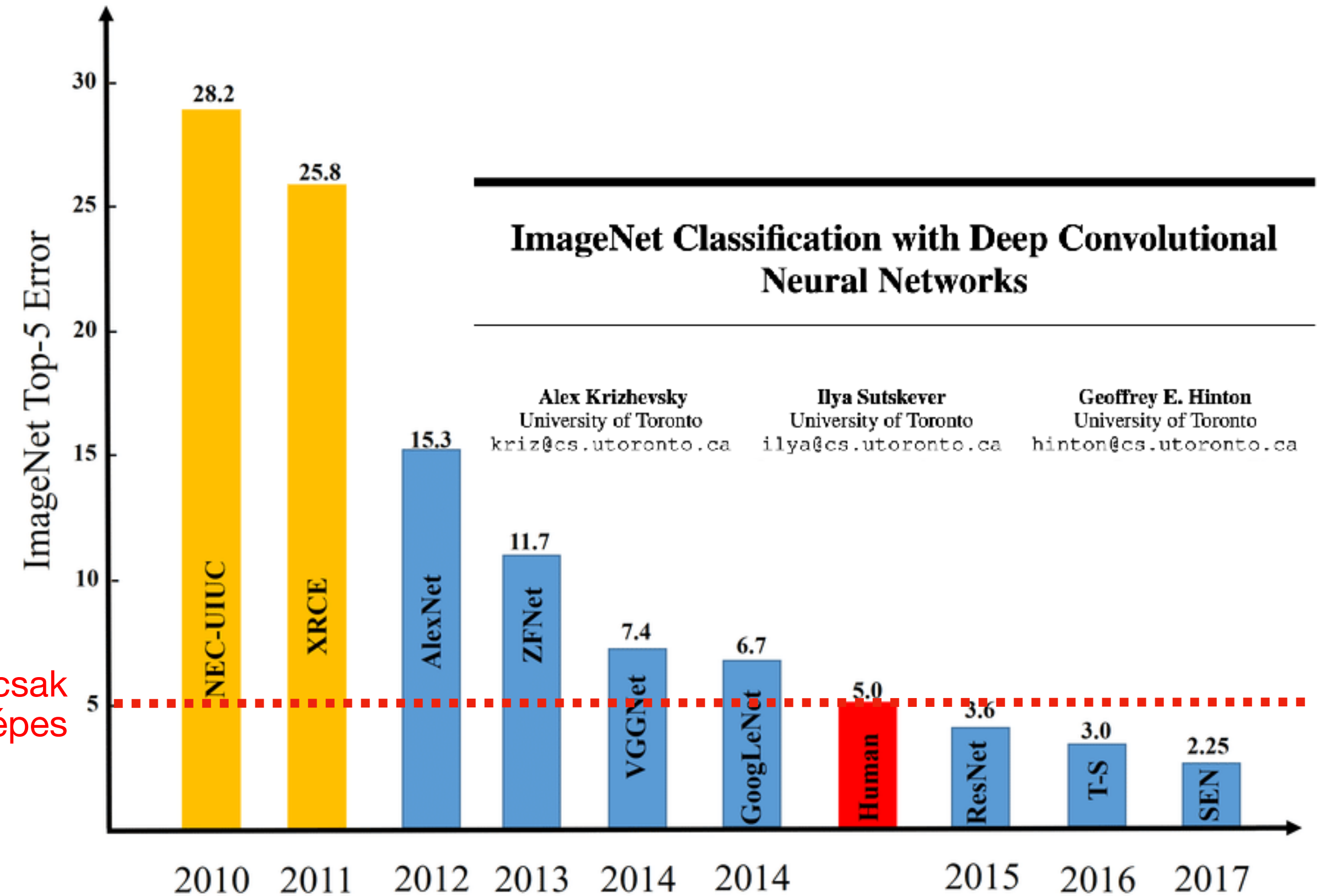
Képosztályozás – Adathalmazok



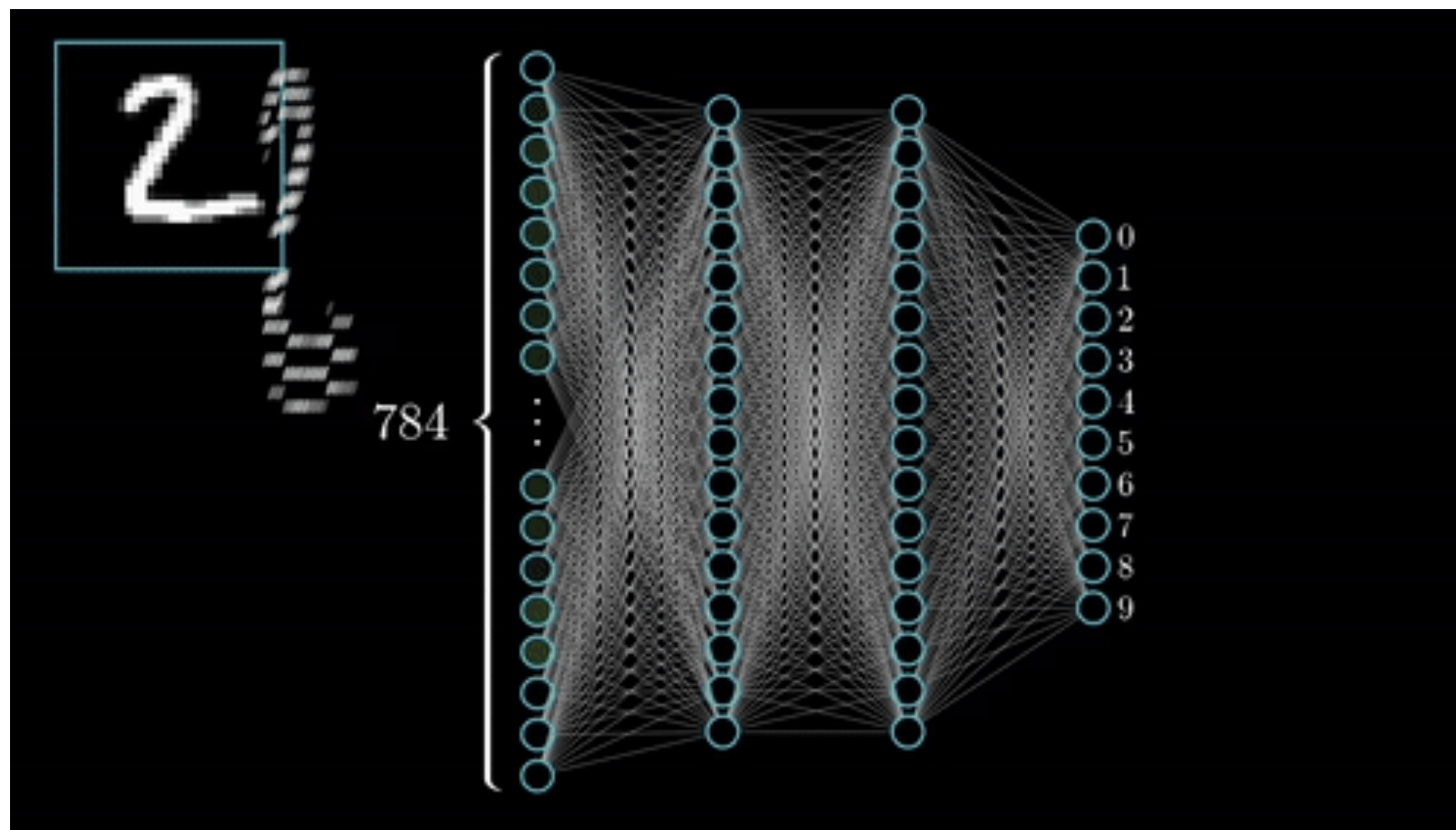
ImageNet-1K:

1281167 (train) + 50000 (validation) + 100000 (test)
 1000 osztály
 155 GB!

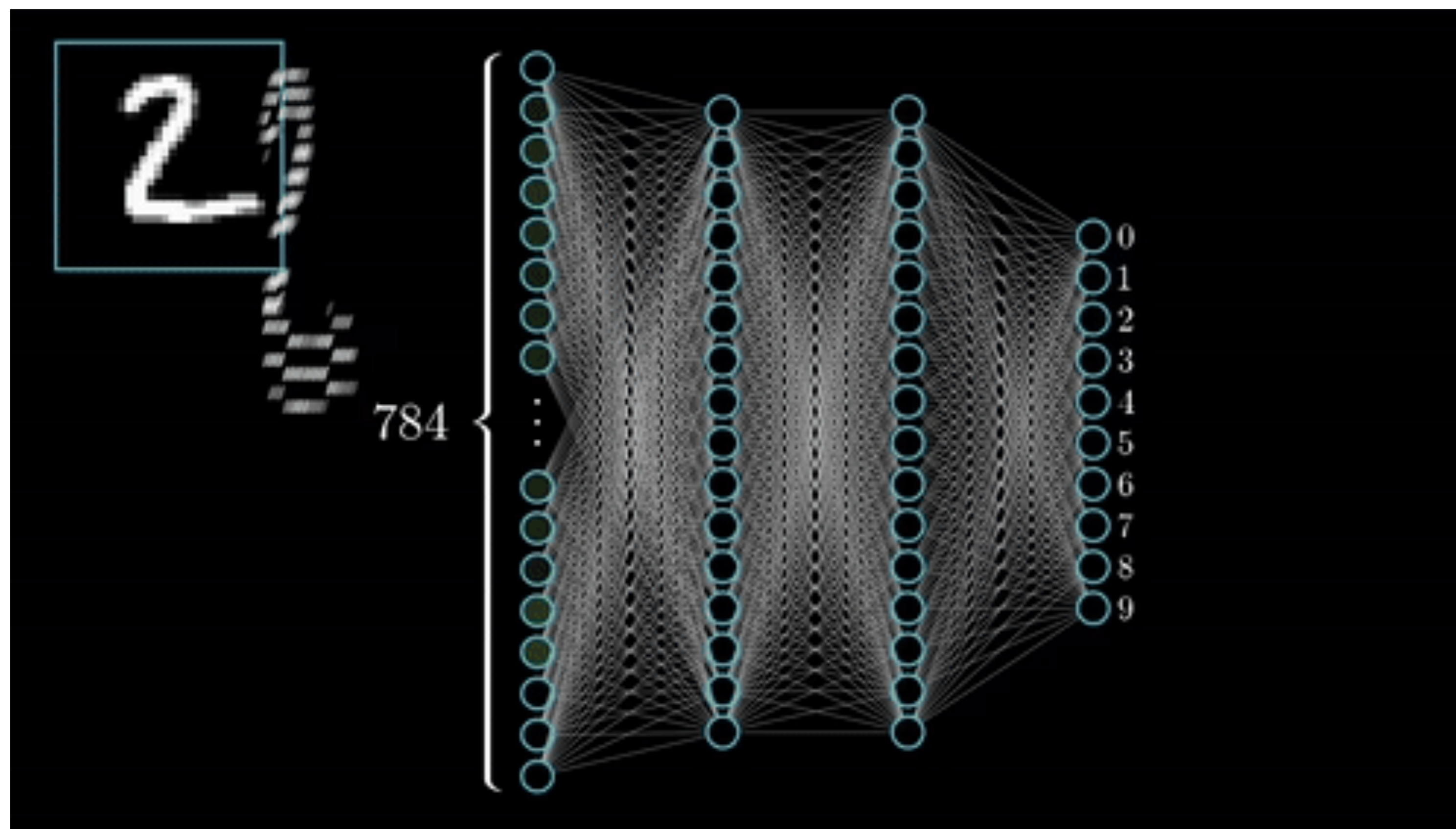
Az ember is csak ennyire képes



Neurális Hálók

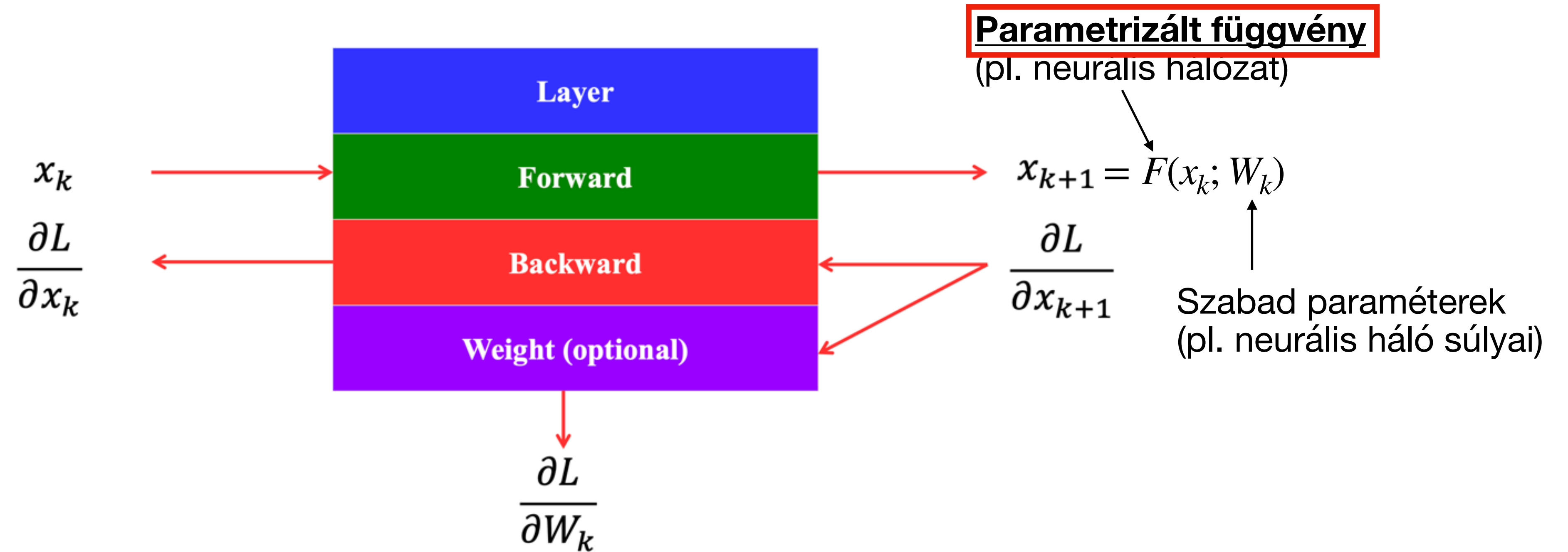


Neurális Hálók



Differenciális Programozás

Általános Interfész



Ilyen “rétegekből” bármilyen számítási procedúra felépíthető és differenciálható!
(Pl. minden neurális hálózat is)

Forrás: Szemenyei M.

Neurális Háló

Lineáris réteg

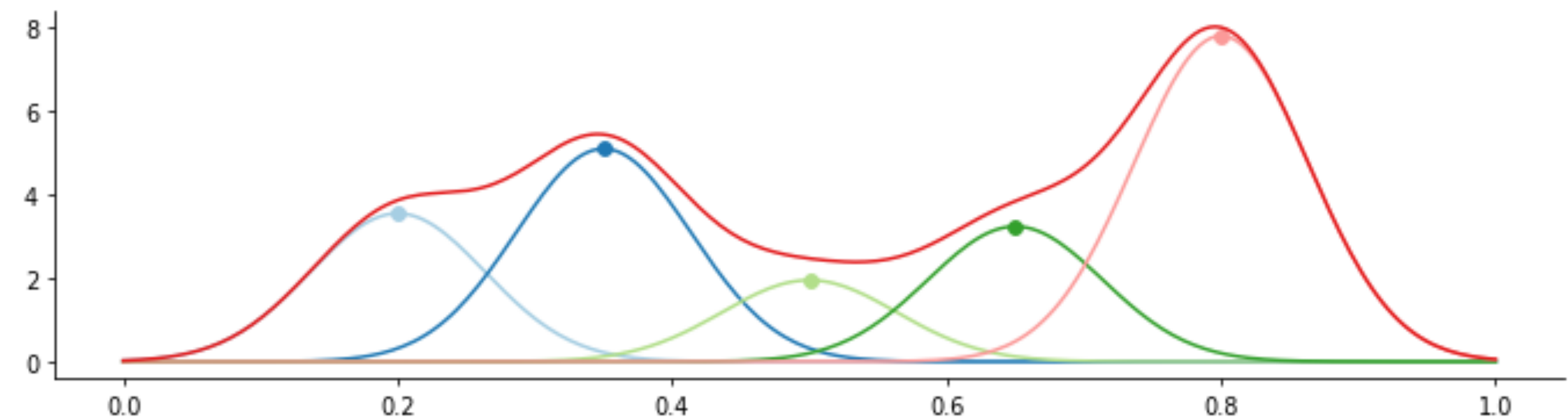
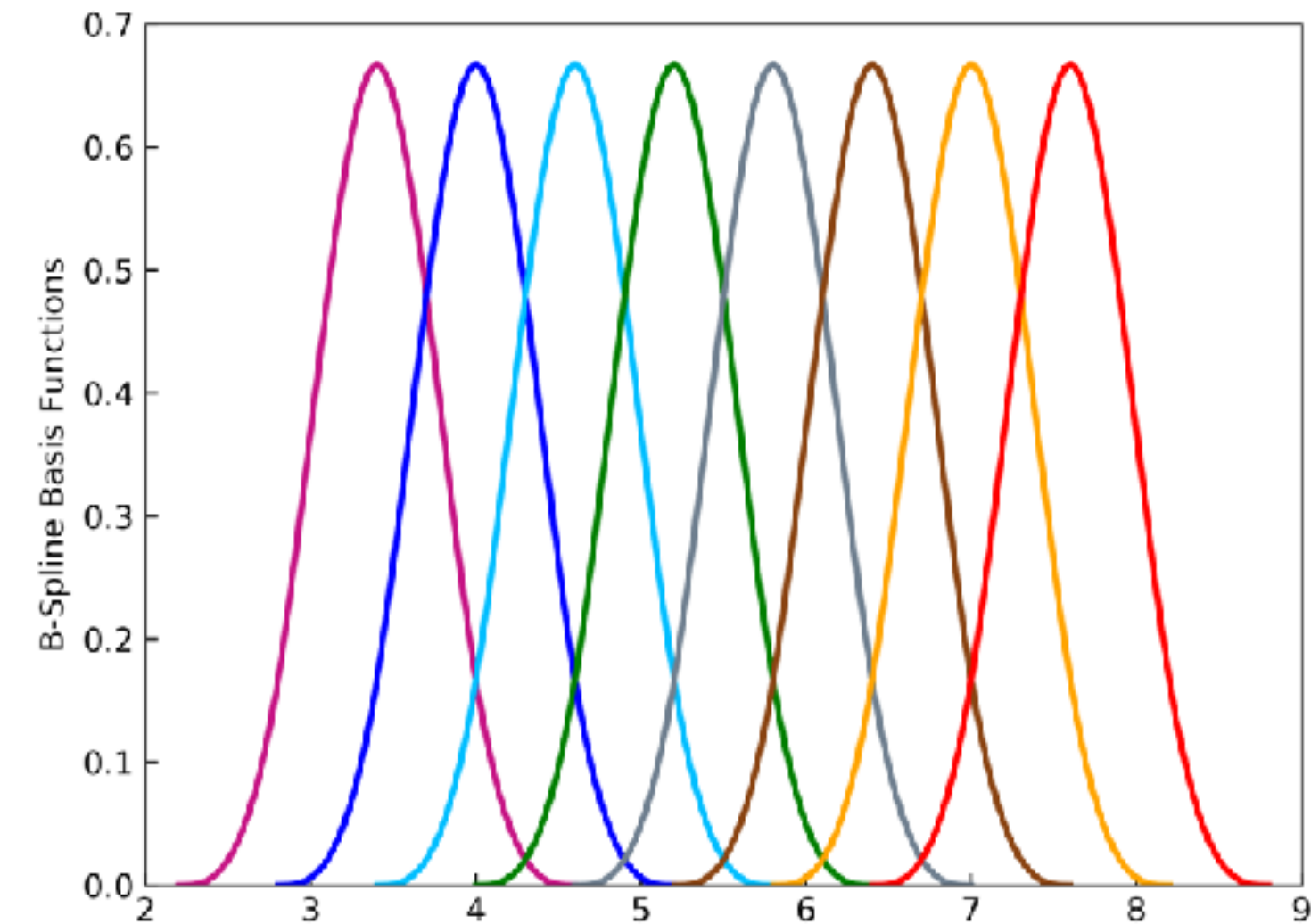
$$y = \sum_i w_i x_i = Wx$$

Több lineáris réteg:
Ekvivalens egy (speciálisan paraméterezett) lineáris réteggel!

Neurális Háló

Általánosított lineáris réteg

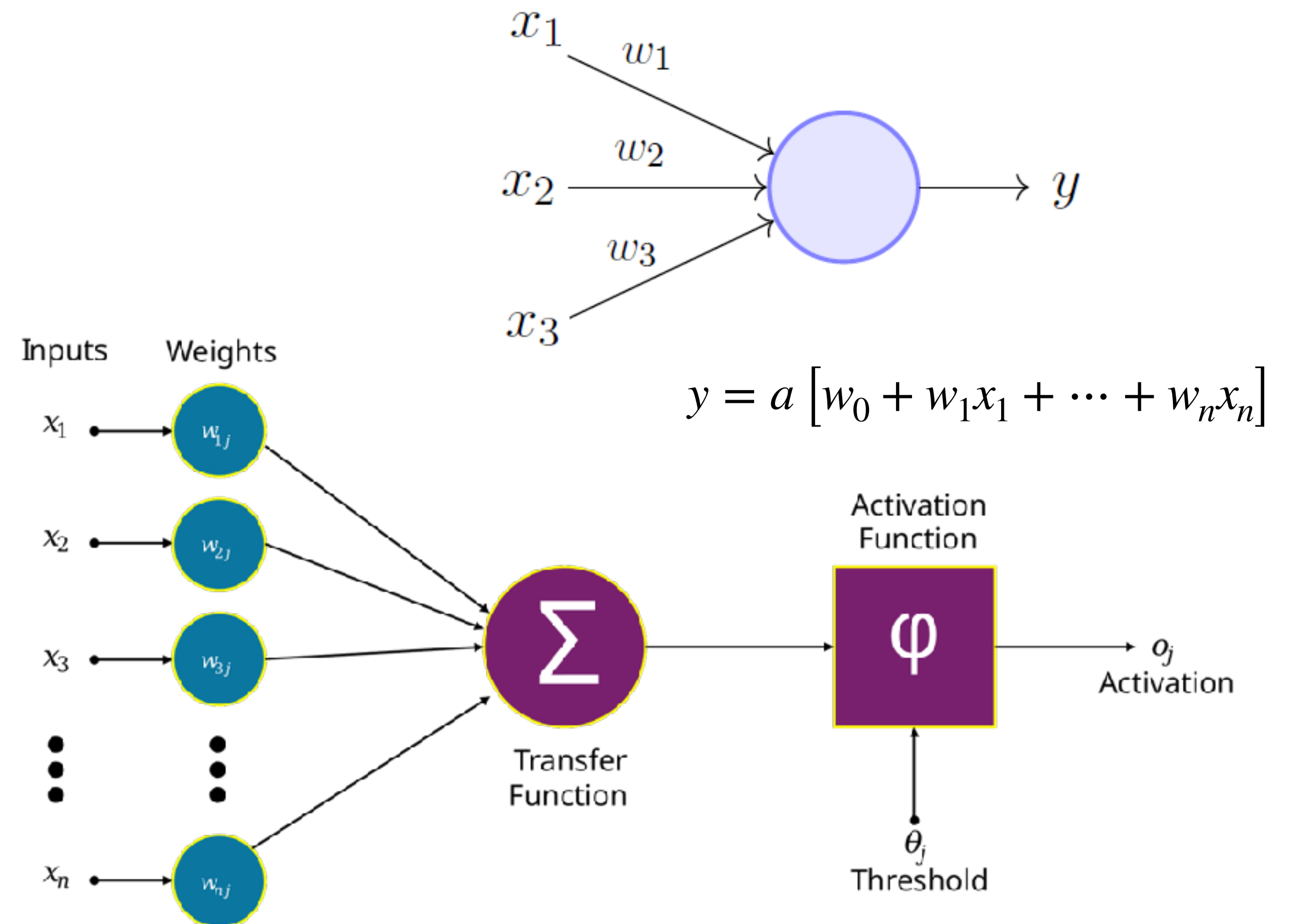
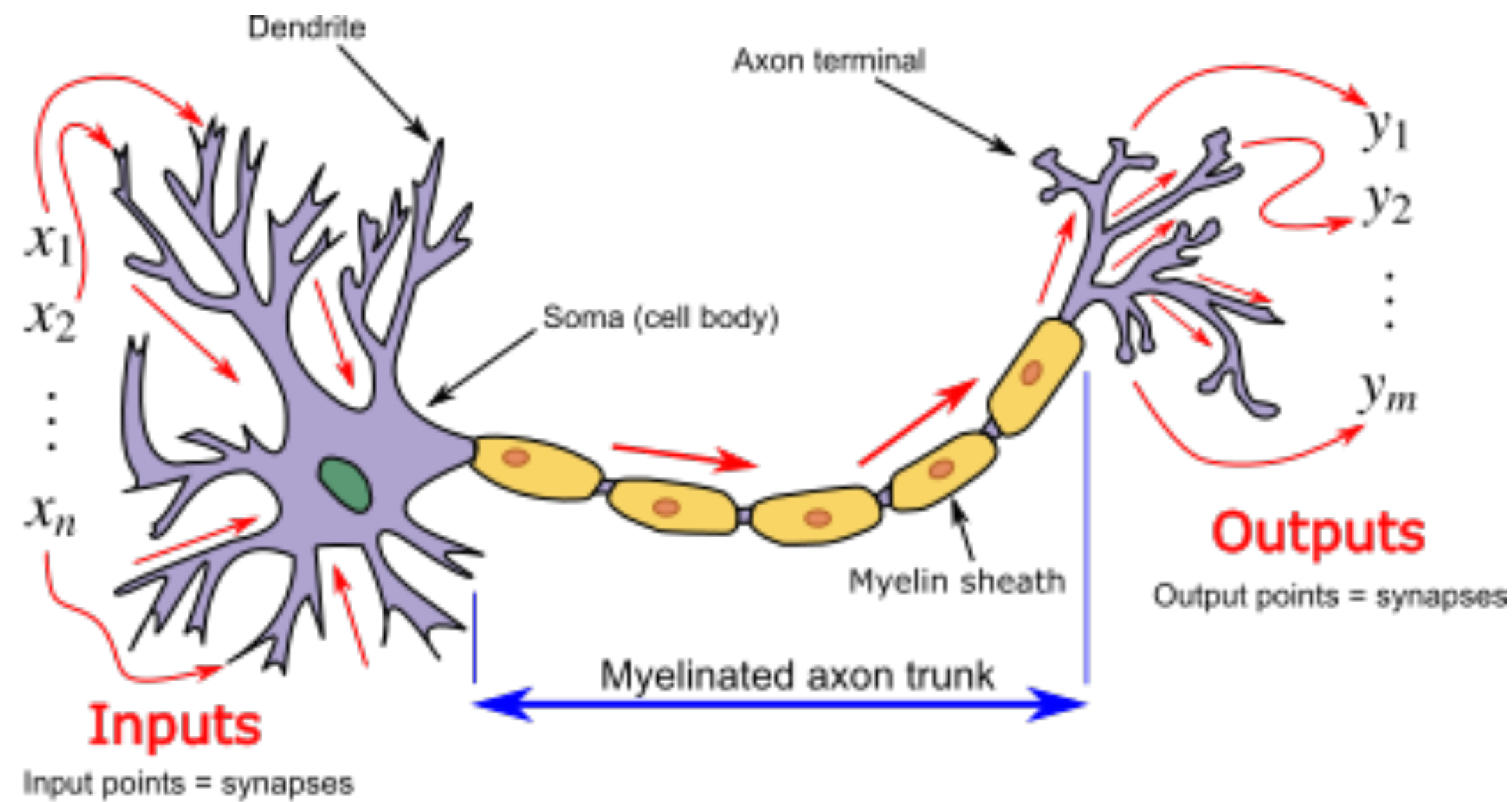
- Parametrizált függvény: $y = f[x_i, \phi]$
- Klasszikusan — lineáris függvénycsalád
 - Polinom $y = \sum_i w_i \beta_i(x)$
 - Trigonometrikus (Fourier)
 - Spline (szakaszos polinom)
 - Gauss (RBF)
- Nagyon rosszul skálázódik magasabb dimenziókra...



Neurális Háló

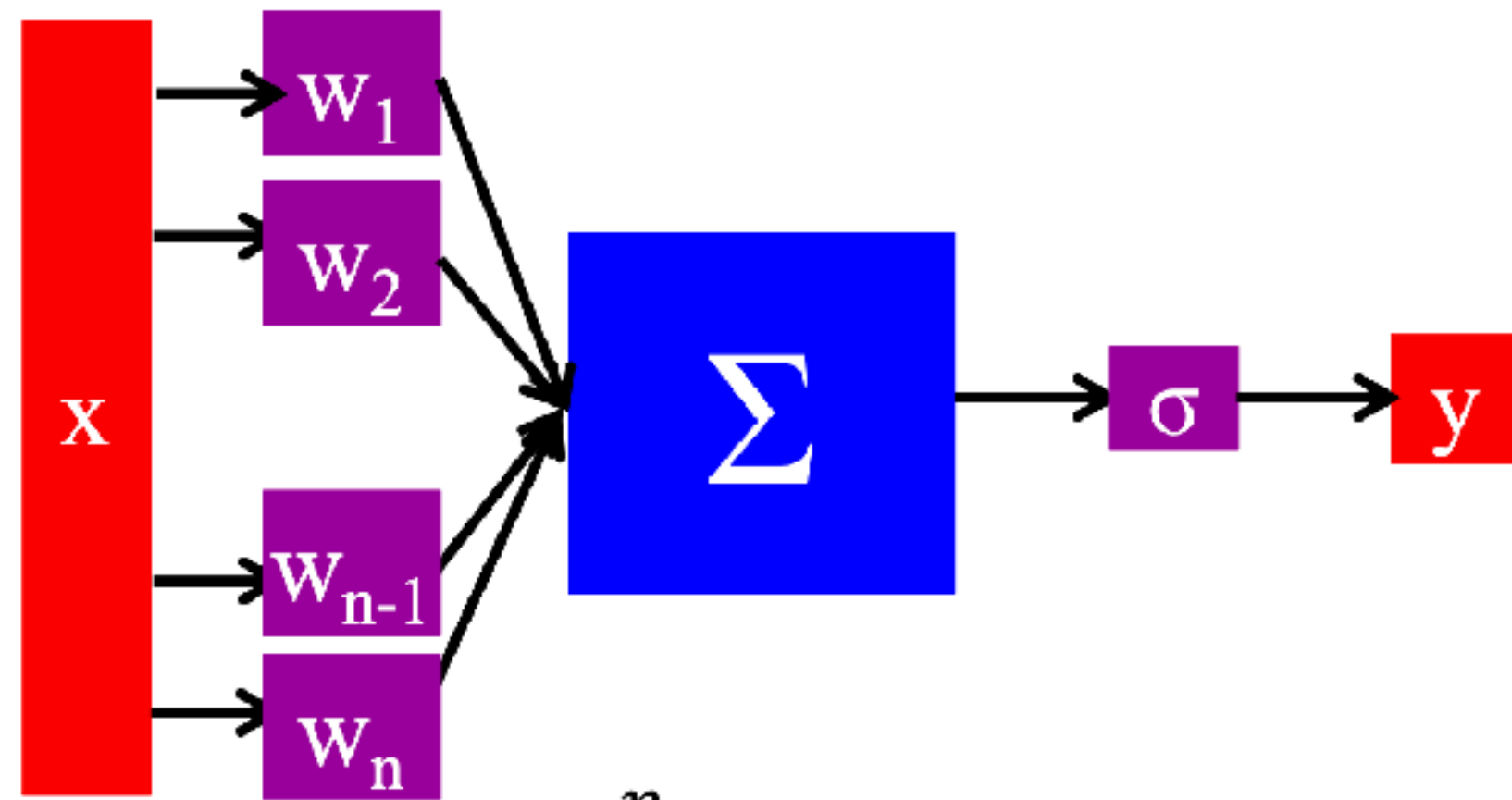
Neuronok, aktivációs függvények

A LOGICAL CALCULUS OF THE
IDEAS IMMANENT IN NERVOUS ACTIVITY
WARREN S. MCCULLOCH AND WALTER PITTS
(1943)



Neurális Háló

Neuron / Perceptron

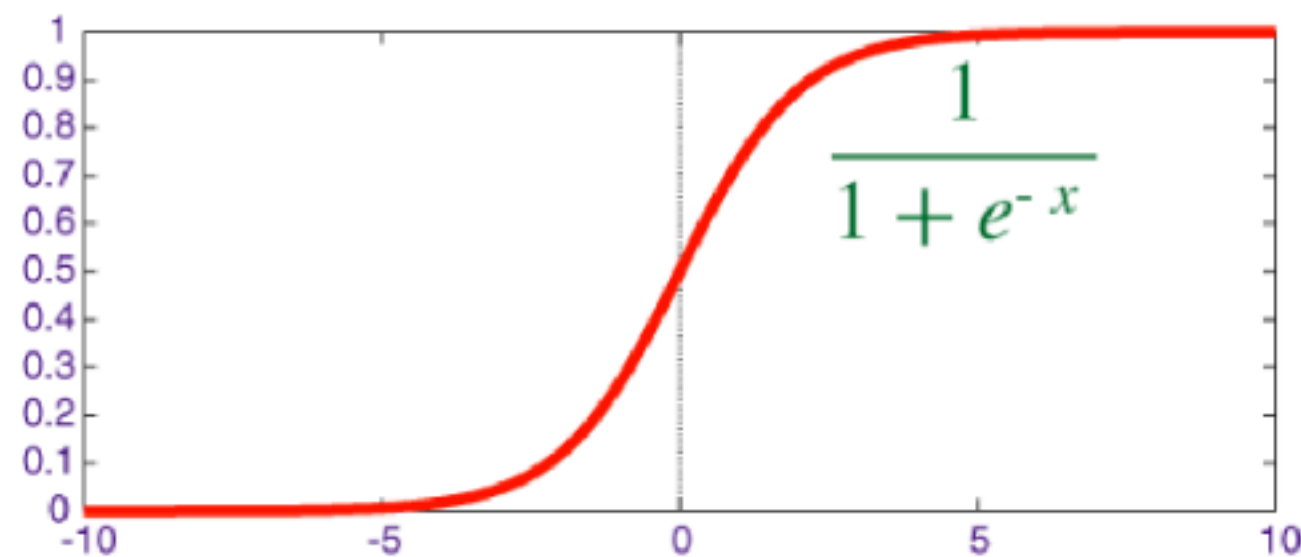


$$y = \sum_i^n w_i x_i = w^T x$$

Lineáris kombináció
+
Nemlineáris aktivációs függvény
=
(Mesterséges) **Neuron** / **Perceptron**

σ

=



Forrás: Szemenyei M.

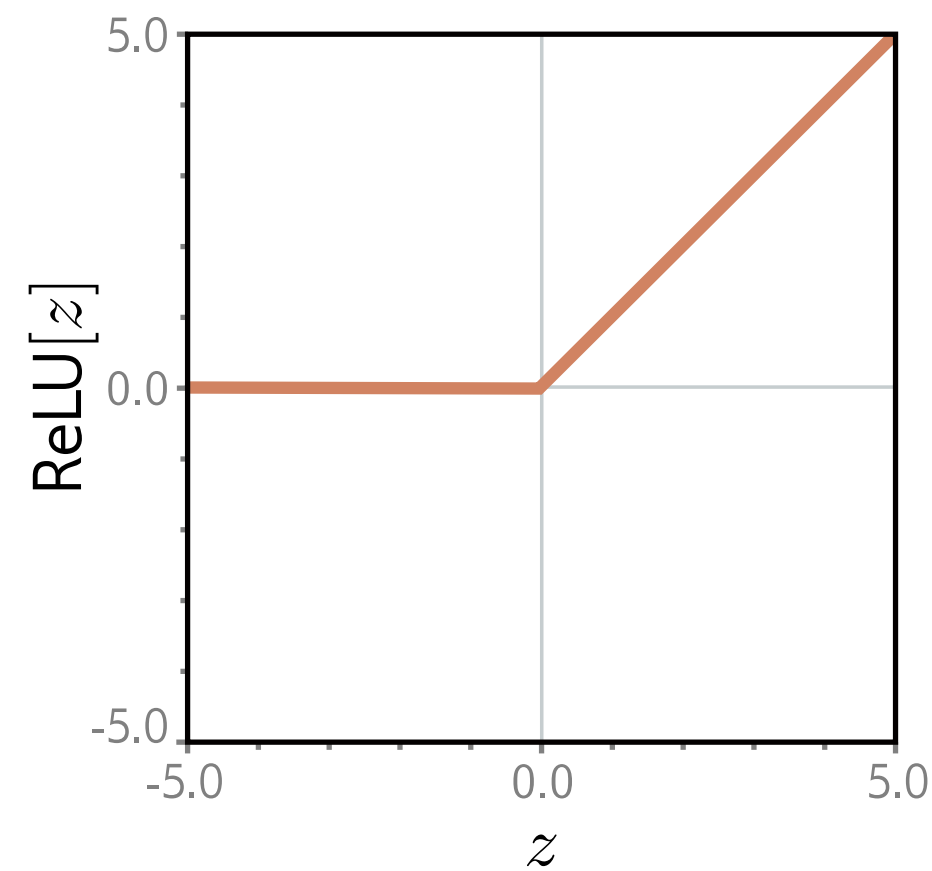
szigmoid aktiváció

(bár természetes, ma már csak elvétve használjuk!)

Neurális Háló

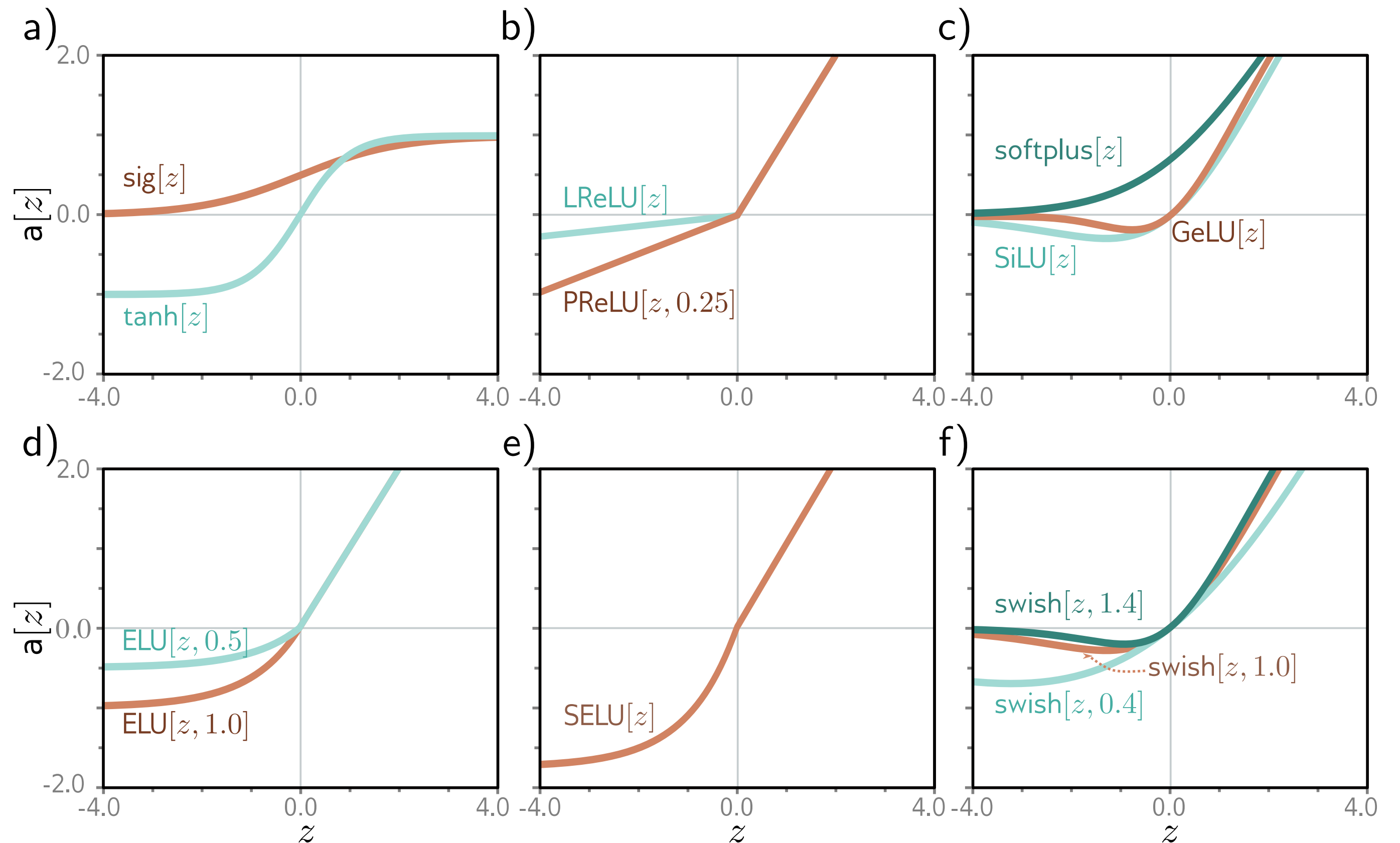
Aktivációs függvények

$$a[z] = \text{ReLU}[z] = \begin{cases} 0 & z < 0 \\ z & z \geq 0 \end{cases}$$



Korszerű aktivációs függvény választás:
Rectified Linear Unit (ReLU)

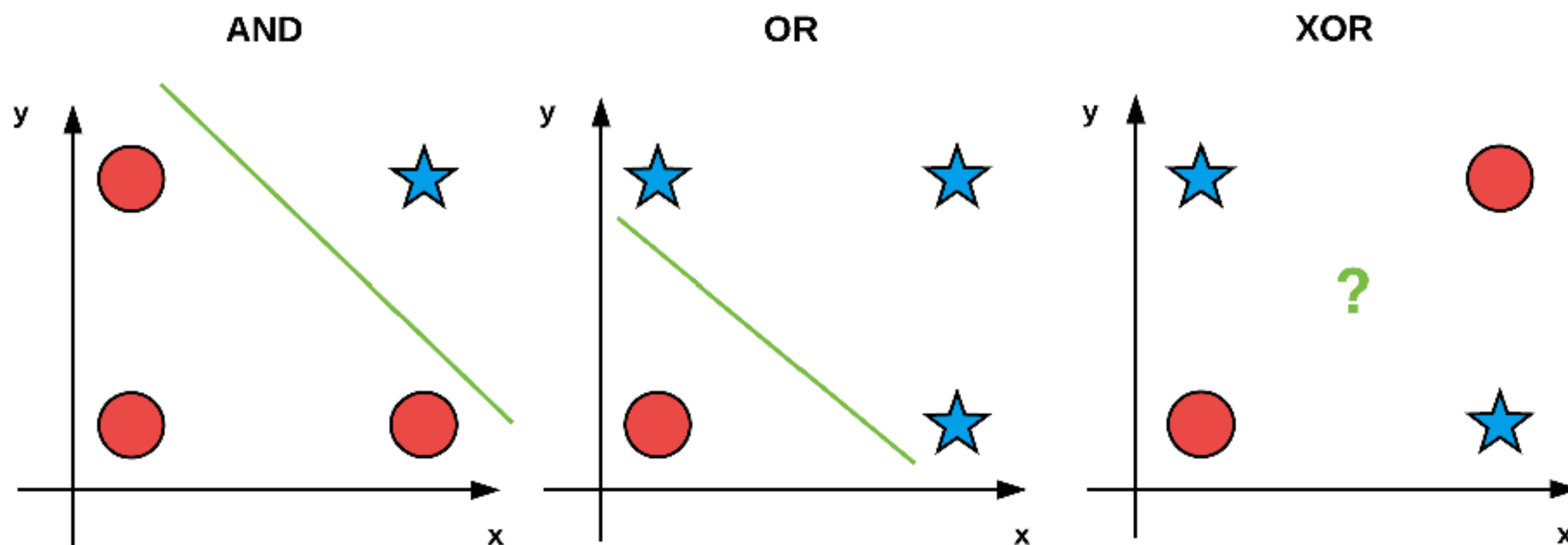
Forrás: [S.J.D. Prince](#)



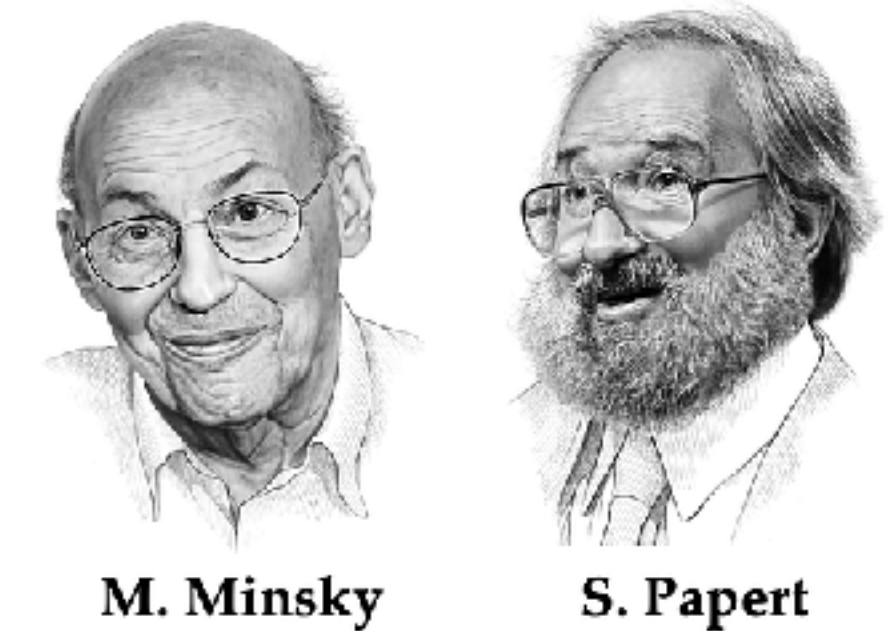
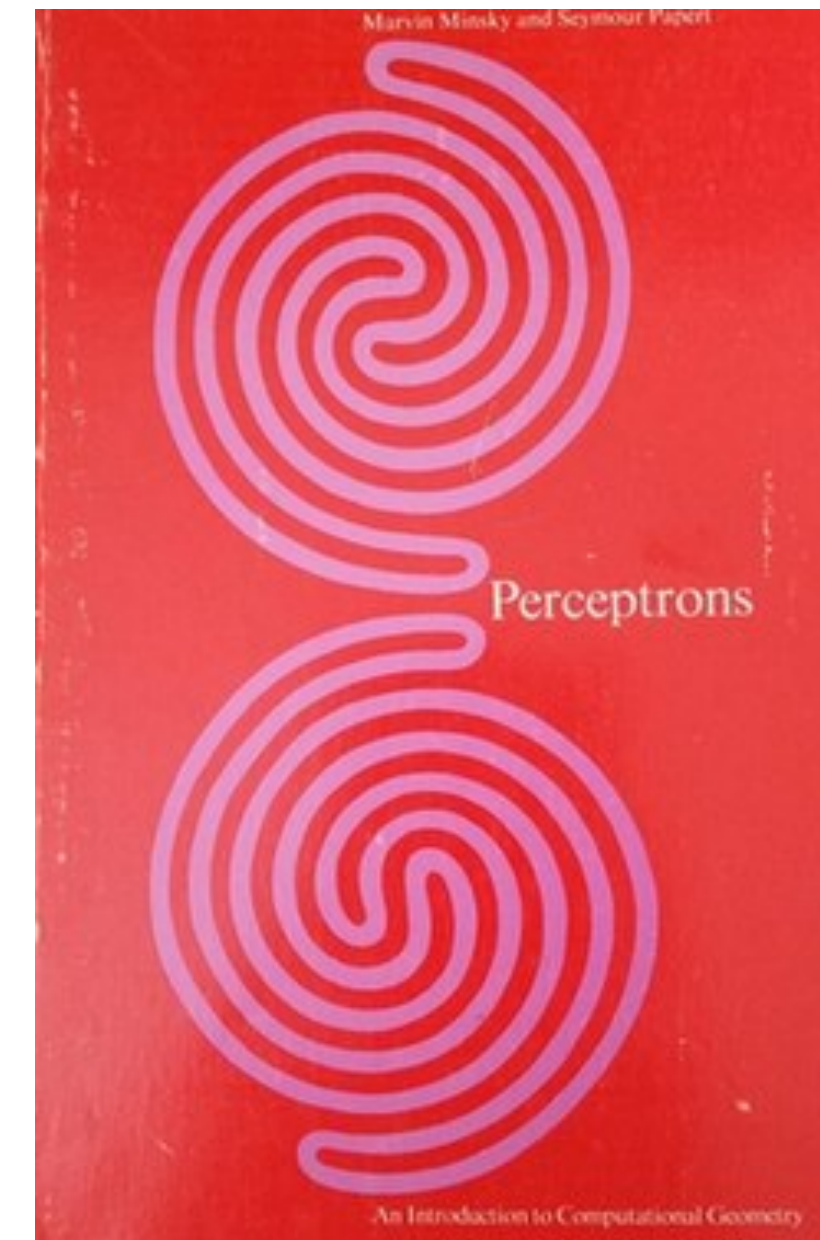
Neurális Háló

Neuronok képességei

- Mit képes egy mesterséges neuron (perceptron) megtanulni?
- XOR-t pl. nem...



Több perceptron rétegre van szükség!



[1969]

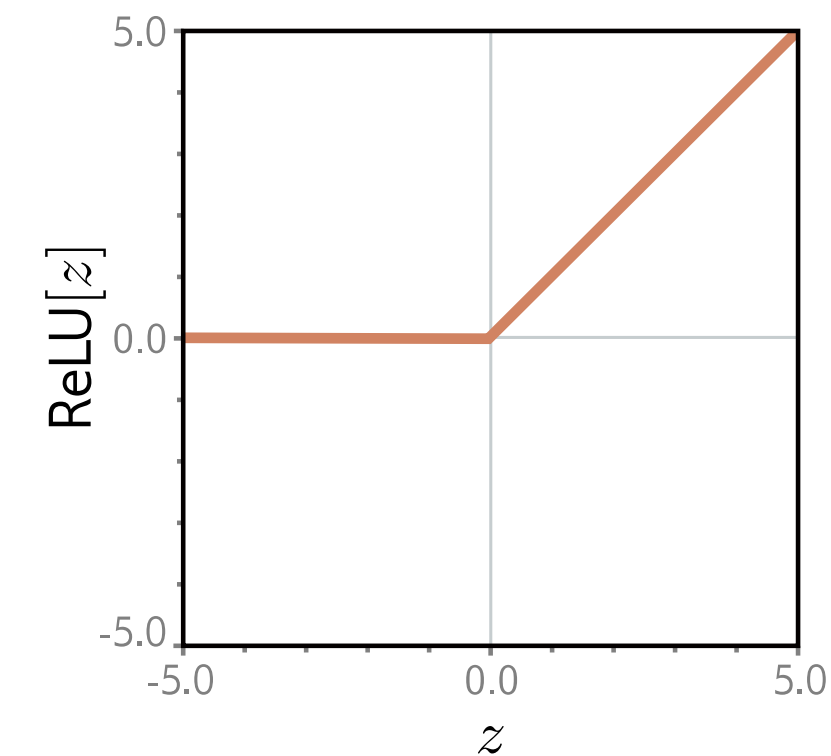
“Neural networks trained by gradient descent would fail to scale up, due to local minima, extremely large weights, and slow convergence.” (1989-es kiadás)

Sokáig ez volt az általános vélemény...

Neurális Háló

Kétrétegű háló – 1 bemenet

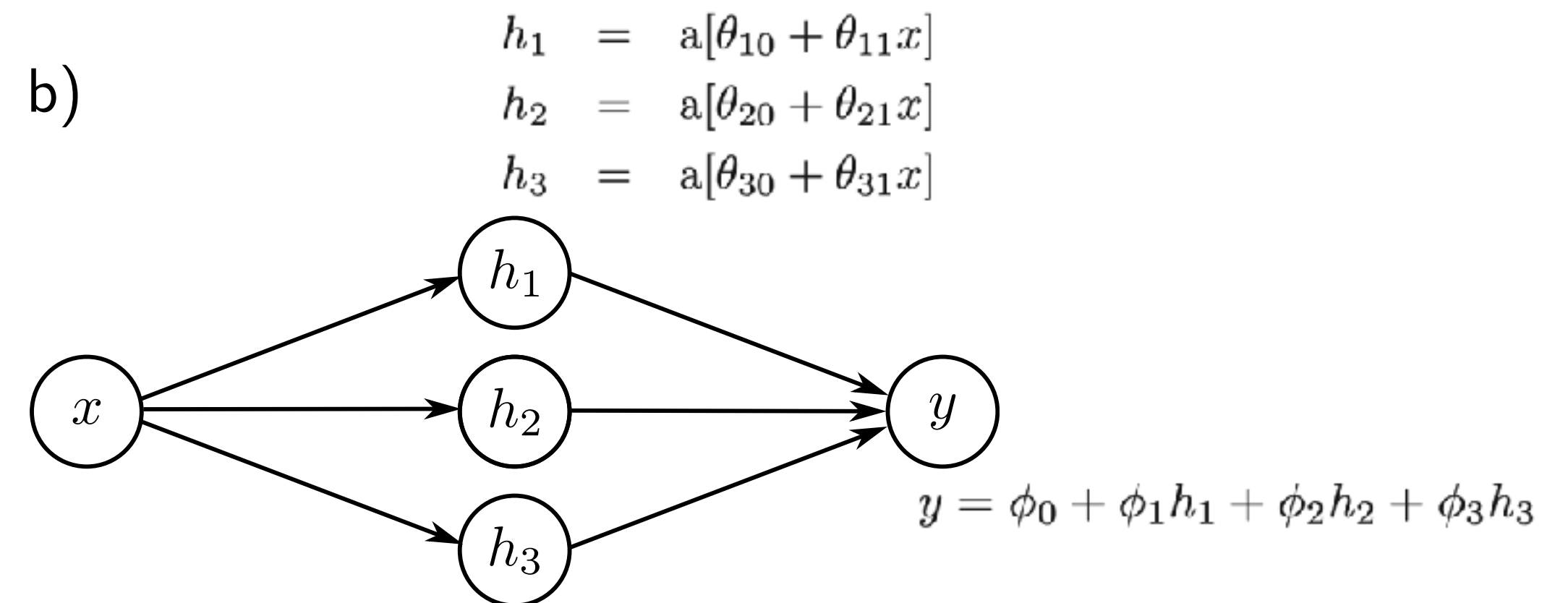
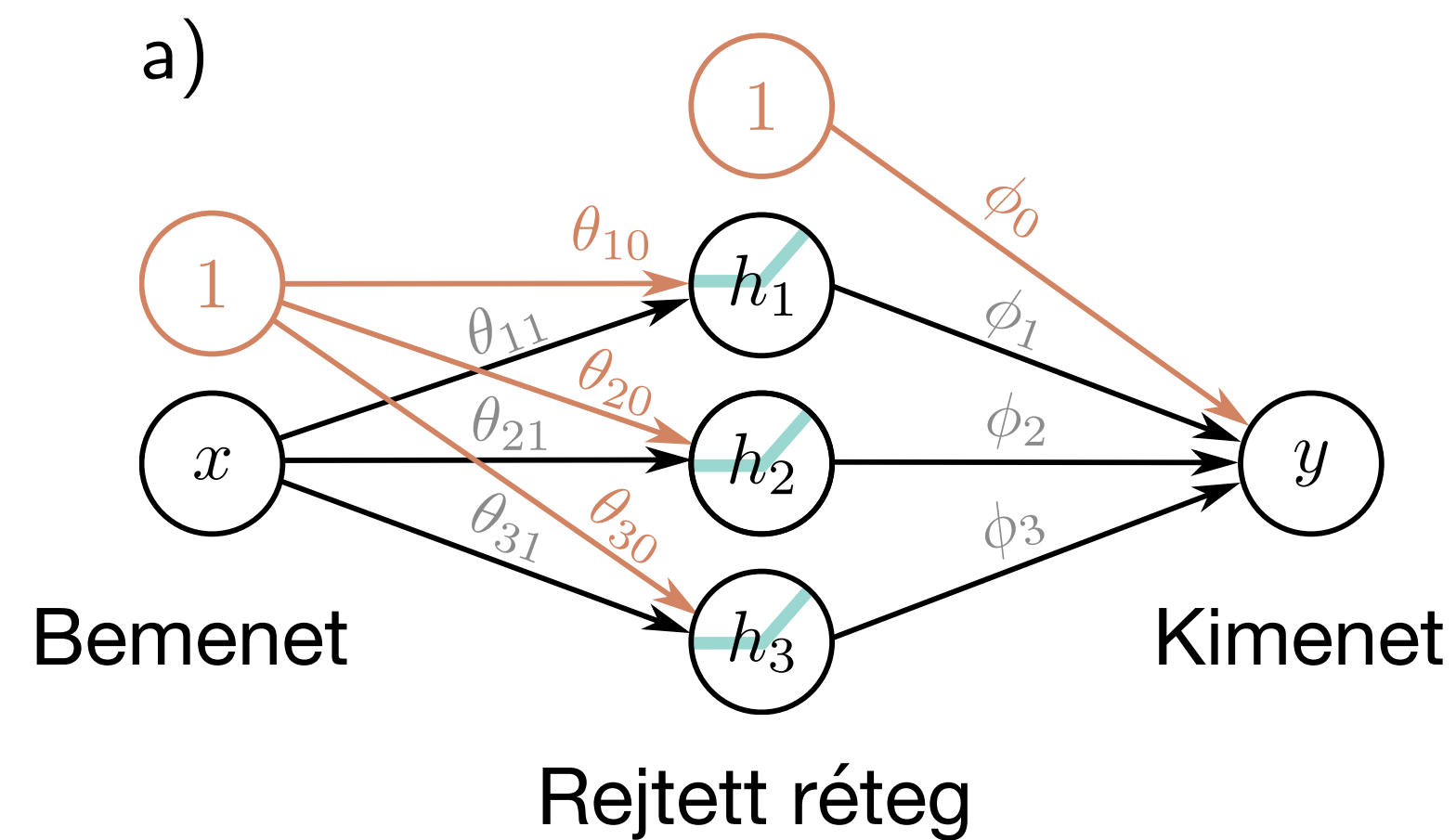
$$a[z] = \text{ReLU}[z] = \begin{cases} 0 & z < 0 \\ z & z \geq 0 \end{cases}$$



Forrás: [S.J.D. Prince](#)

$$y = f[x, \phi]$$

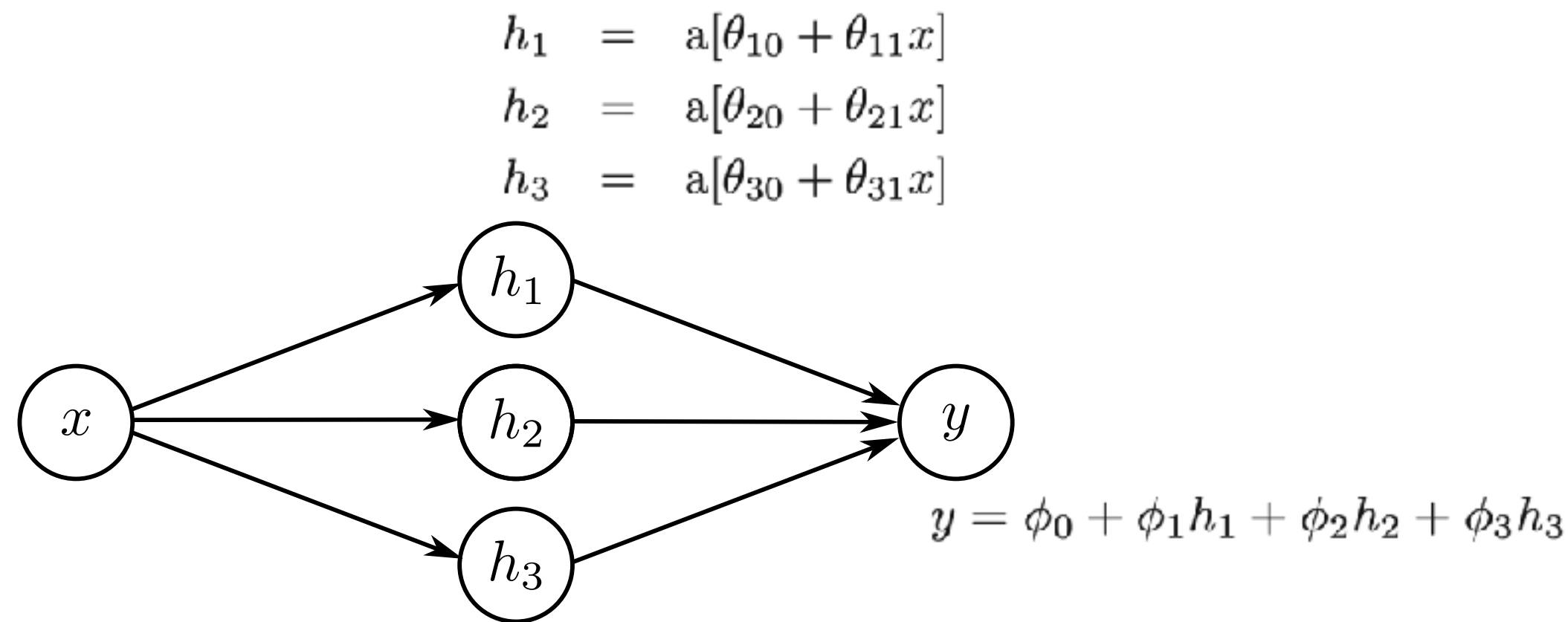
$$= \phi_0 + \phi_1 a[\theta_{10} + \theta_{11}x] + \phi_2 a[\theta_{20} + \theta_{21}x] + \phi_3 a[\theta_{30} + \theta_{31}x]$$



(Egyszerűsített diagram)

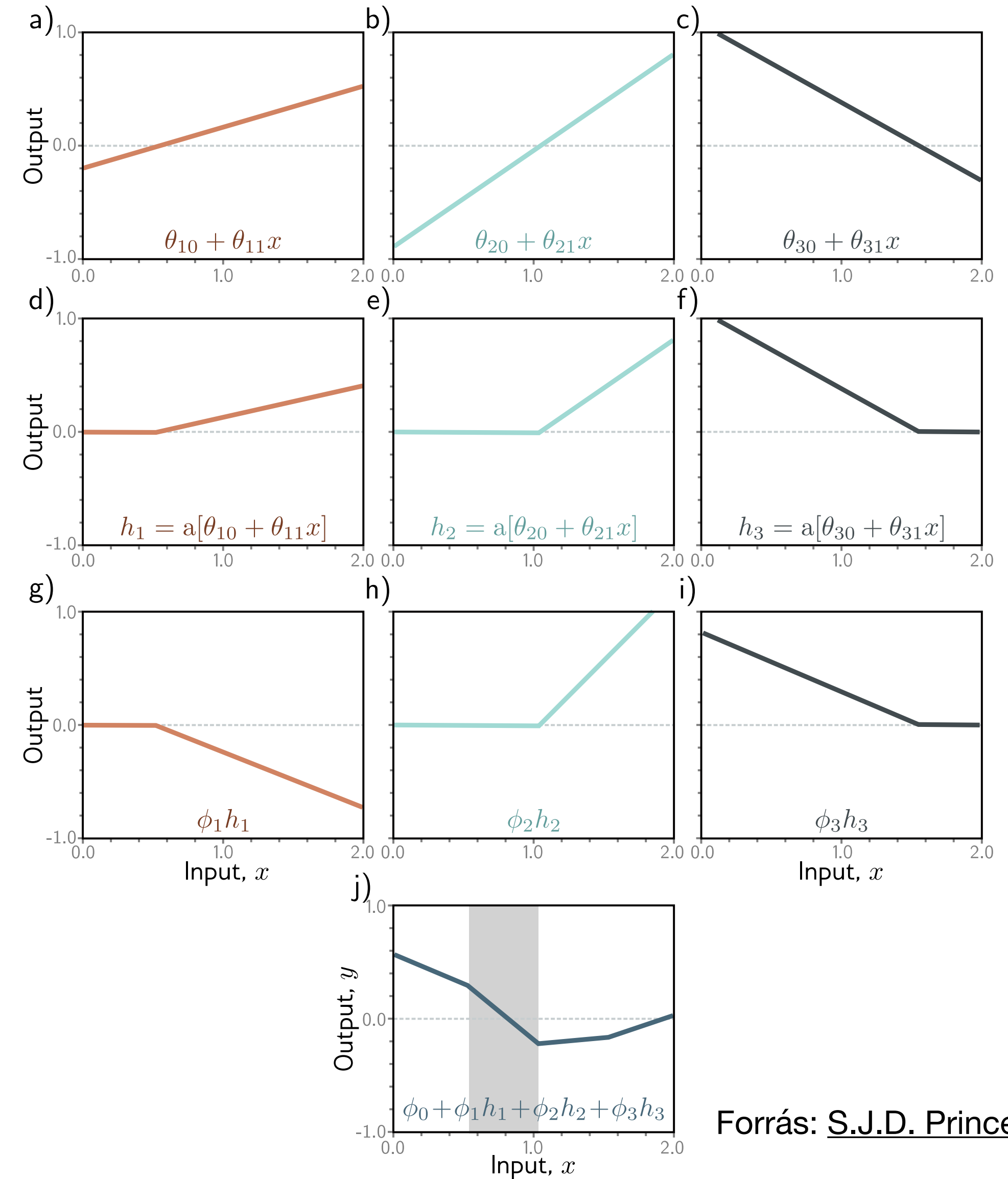
Neurális Háló

Kétrétegű háló – 1 bemenet



Applet: [LINK](#)

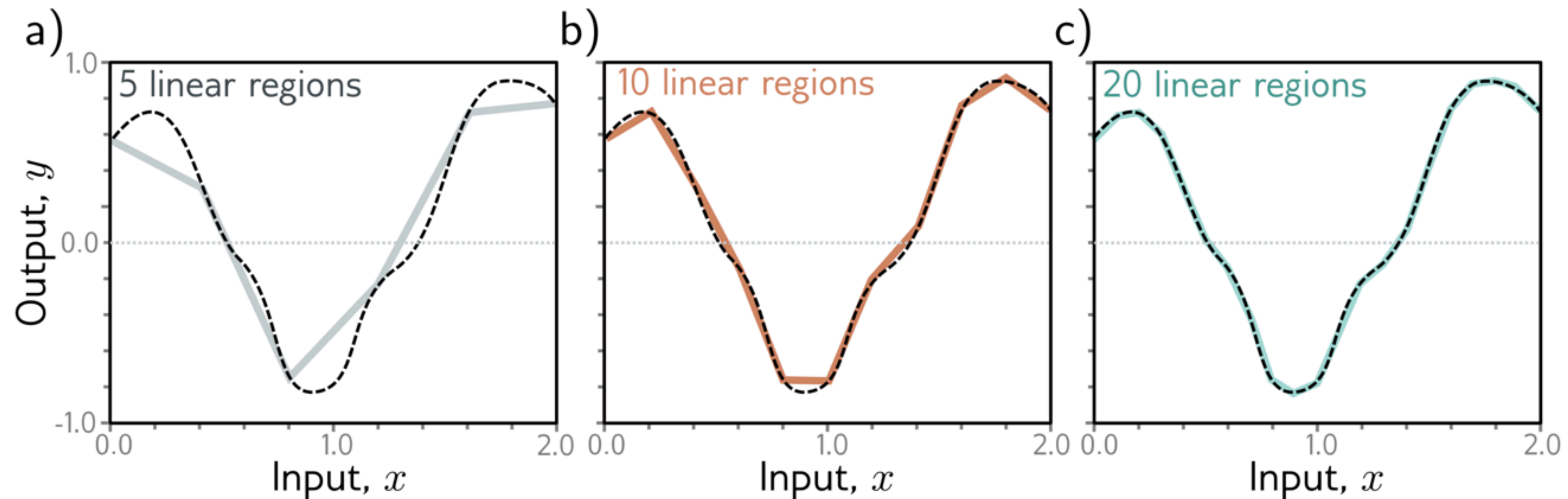
Eredmény: szakaszosan lineáris függvény!



Forrás: [S.J.D. Prince](#)

Neurális Háló

Kétrétegű háló – 1 bemenet



Forrás: [S.J.D. Prince](#)

Univerzális Approximátor:

Kellően sok neuronnal bármilyen függvényt képes megközelíteni a kívánt pontossággal!

Neurális Háló

Kétrétegű háló – 2 bemenet

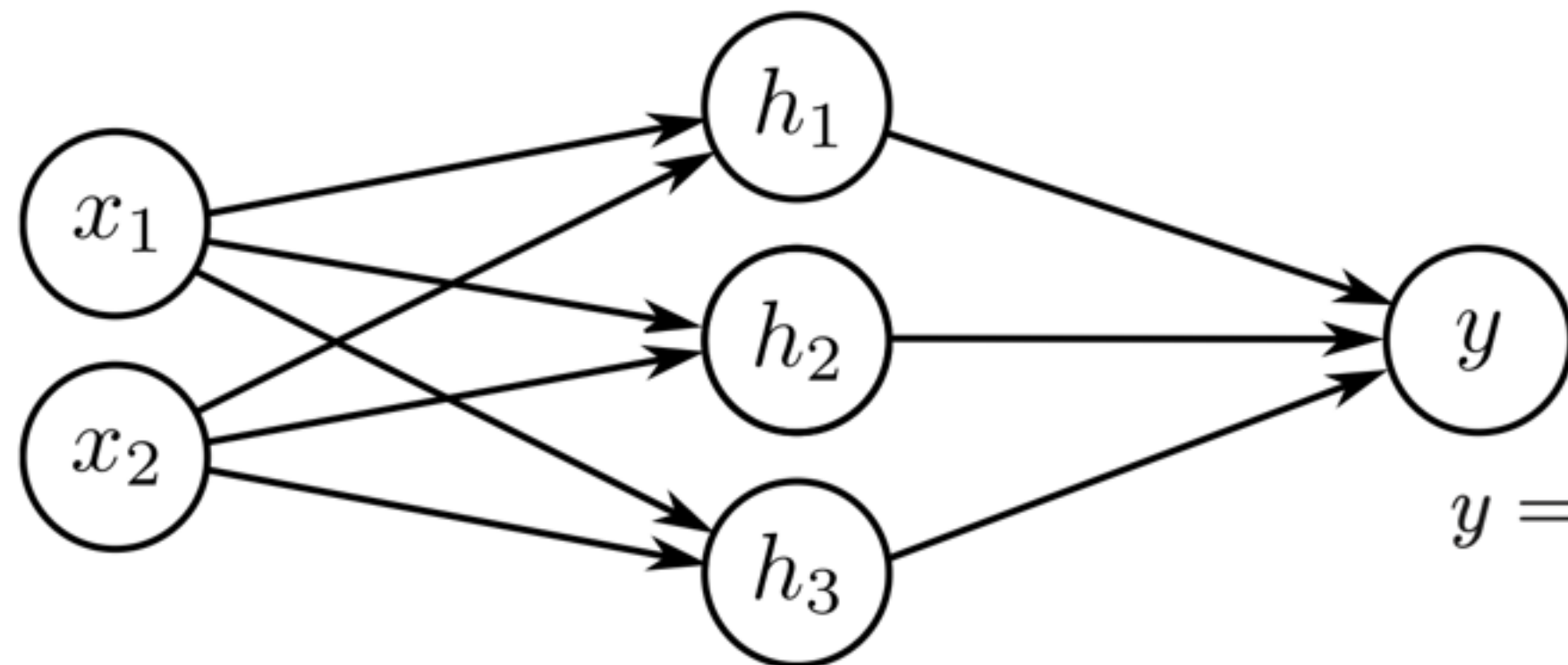
Forrás: S.J.D. Prince

Applet: [LINK](#)

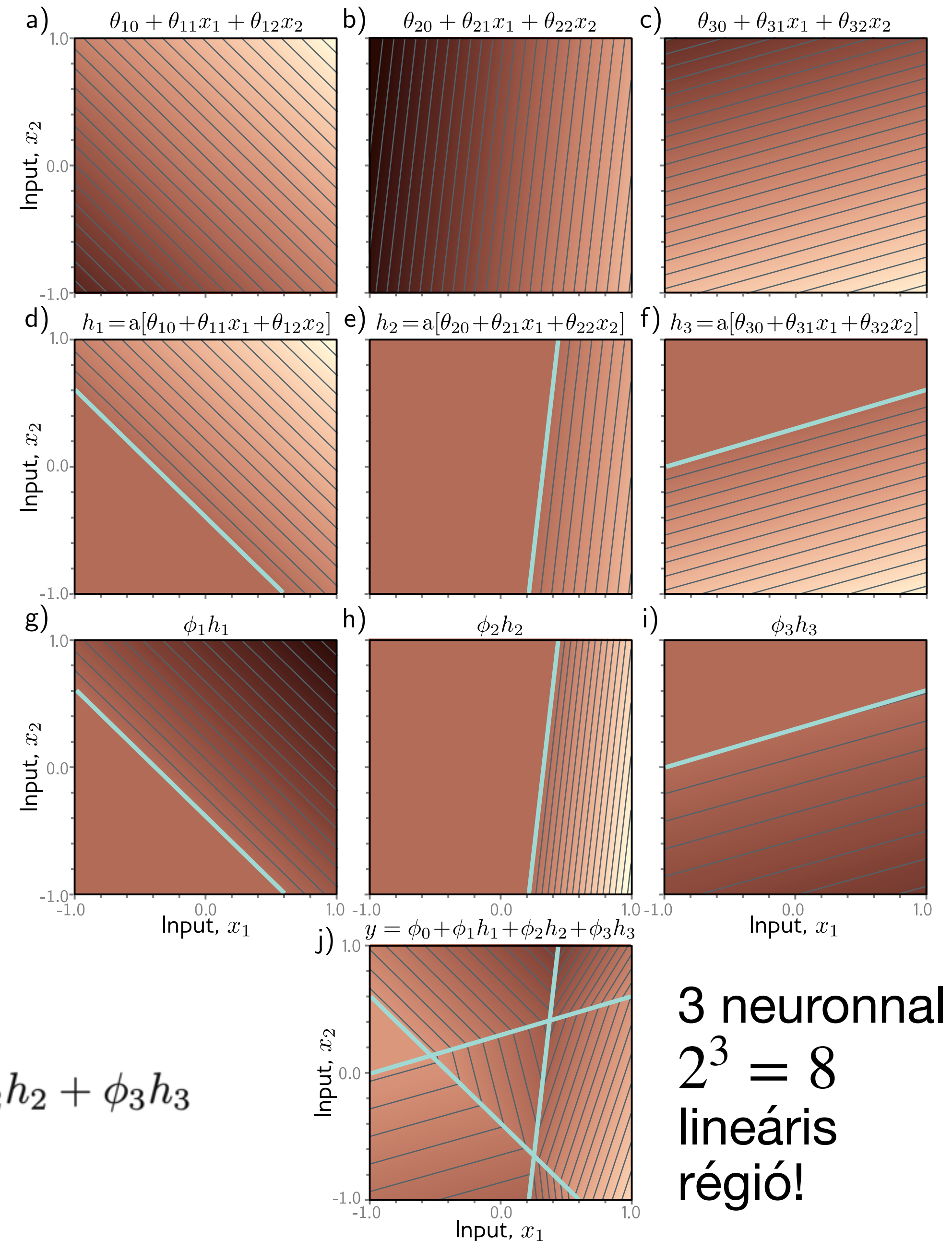
$$h_1 = a[\theta_{10} + \theta_{11}x_1 + \theta_{12}x_2]$$

$$h_2 = a[\theta_{20} + \theta_{21}x_1 + \theta_{22}x_2]$$

$$h_3 = a[\theta_{30} + \theta_{31}x_1 + \theta_{32}x_2]$$



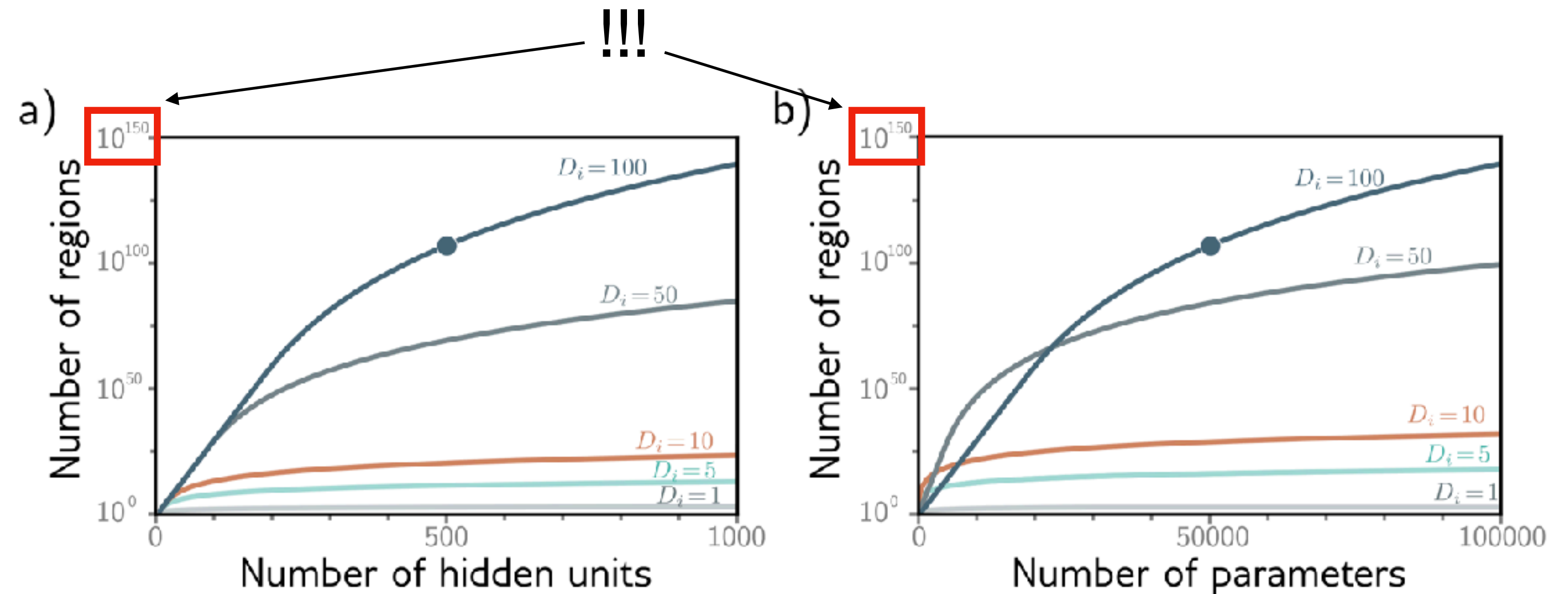
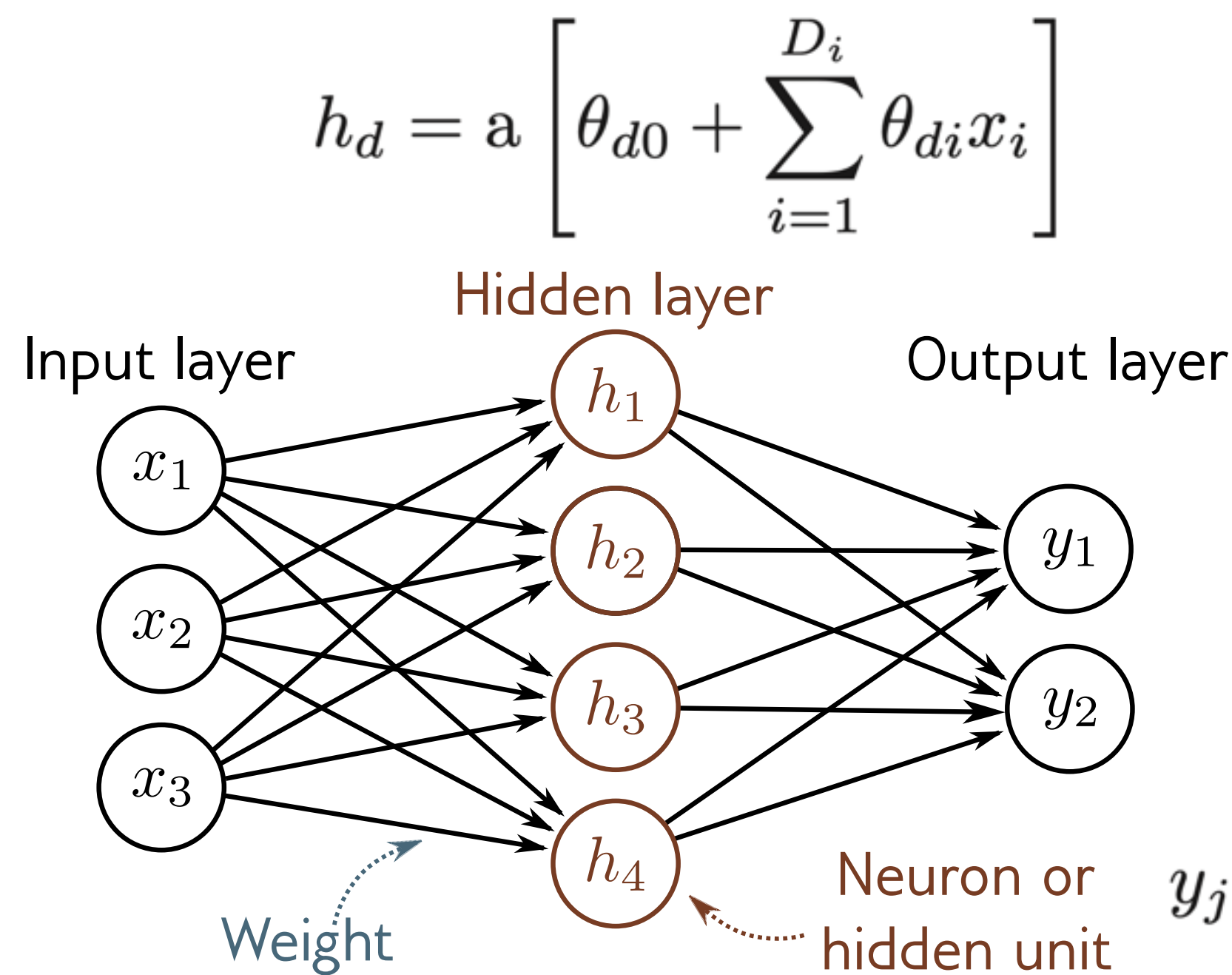
$$y = \phi_0 + \phi_1 h_1 + \phi_2 h_2 + \phi_3 h_3$$



3 neuronnal
 $2^3 = 8$
 lineáris
 régió!

Neurális Hálók

Kétrétegű hálók – Általános eset



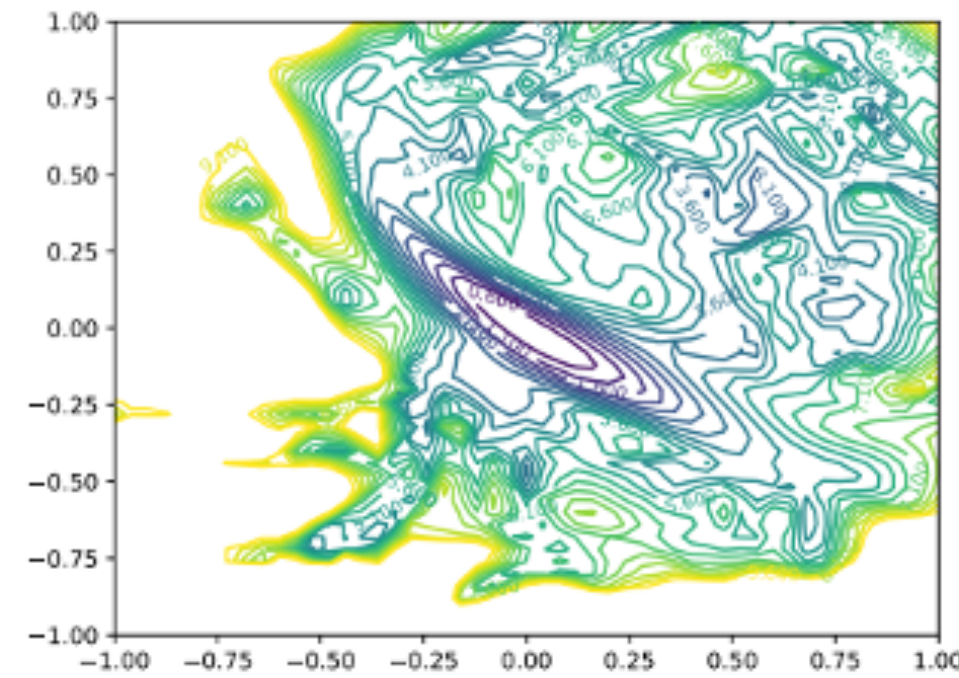
D_i neuron $\sum_{j=0}^{D_i} \binom{D}{j}$ lineáris régióra osztja \mathbb{R}^D -t!

$$y_j = \phi_{j0} + \sum_{d=1}^D \phi_{jd} h_d$$

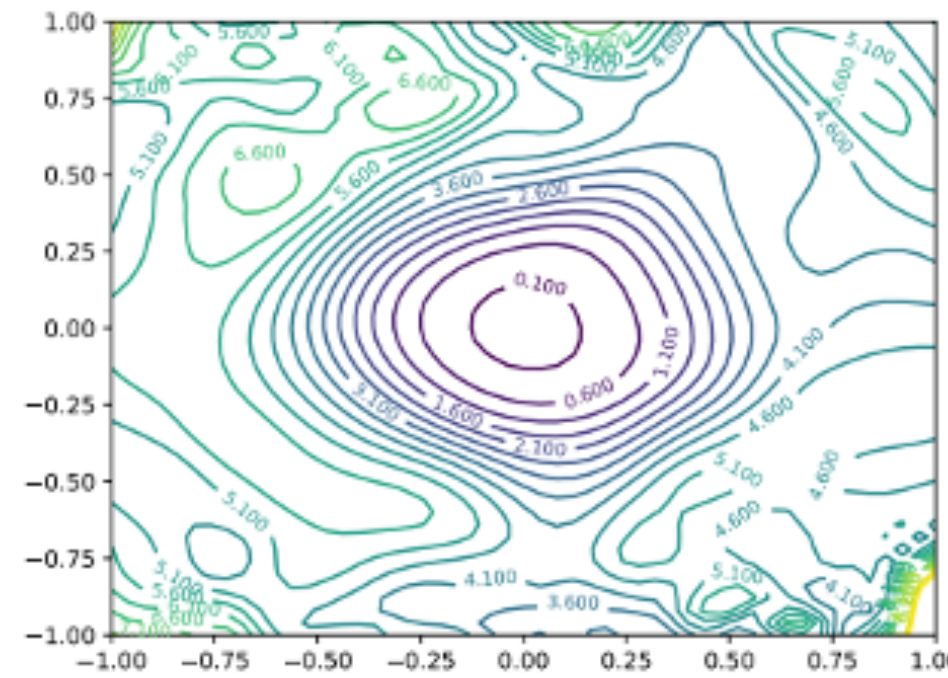
Forrás: [S.J.D. Prince](#)

Neurális Háló

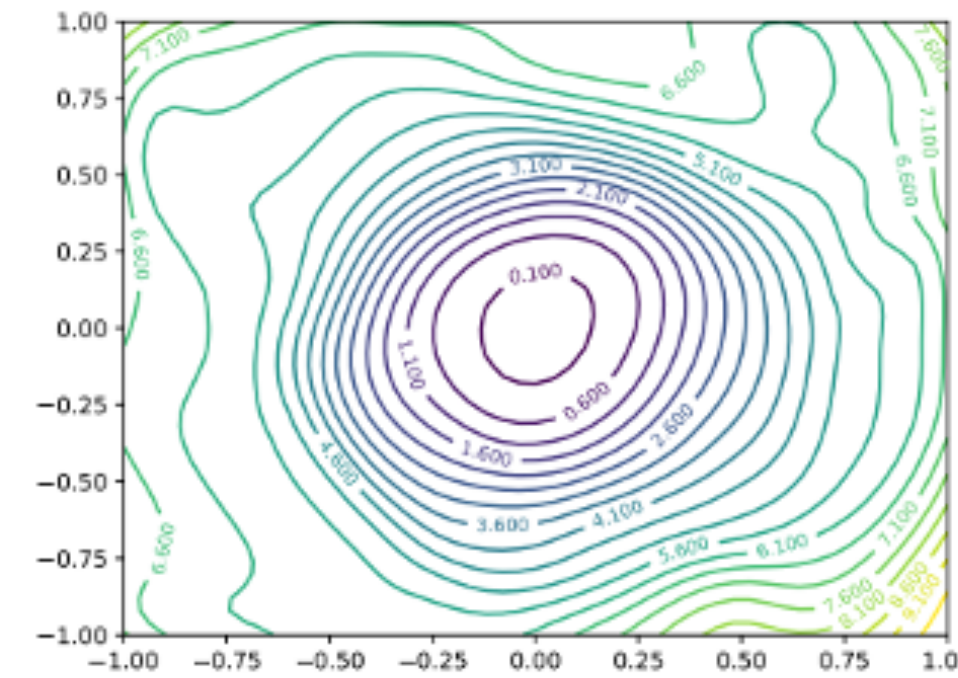
Kétrétegű hálók – Neuronszám hatása



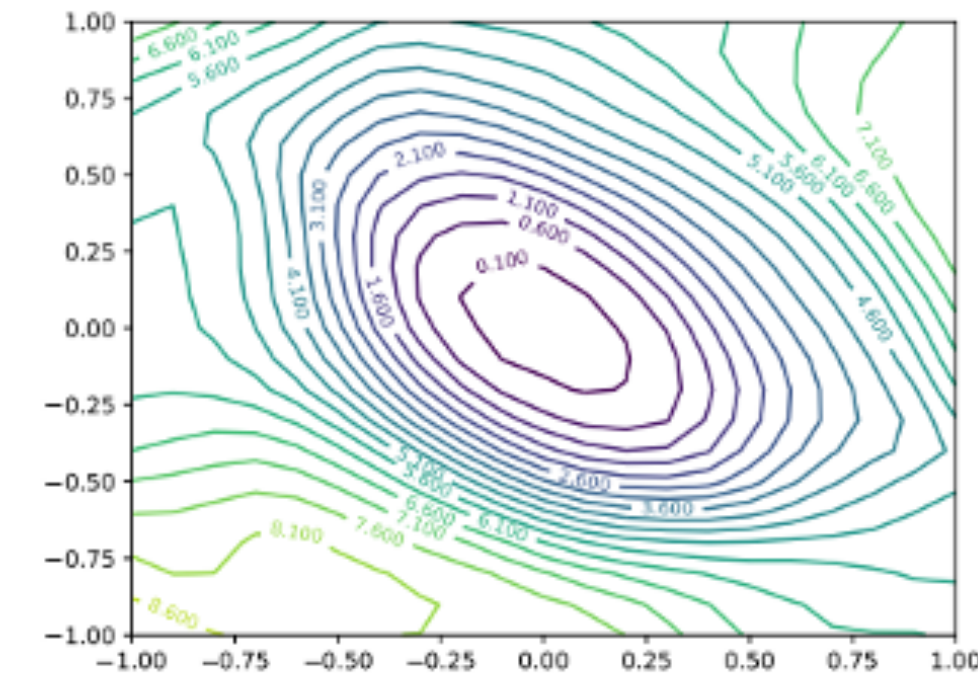
(e) $k = 1$, 13.31%



(f) $k = 2$, 10.26%



(g) $k = 4$, 9.69%



(h) $k = 8$, 8.70%

Ahogy növeljük a neuronok számát a loss egyre “konvexebb” függvény lesz!

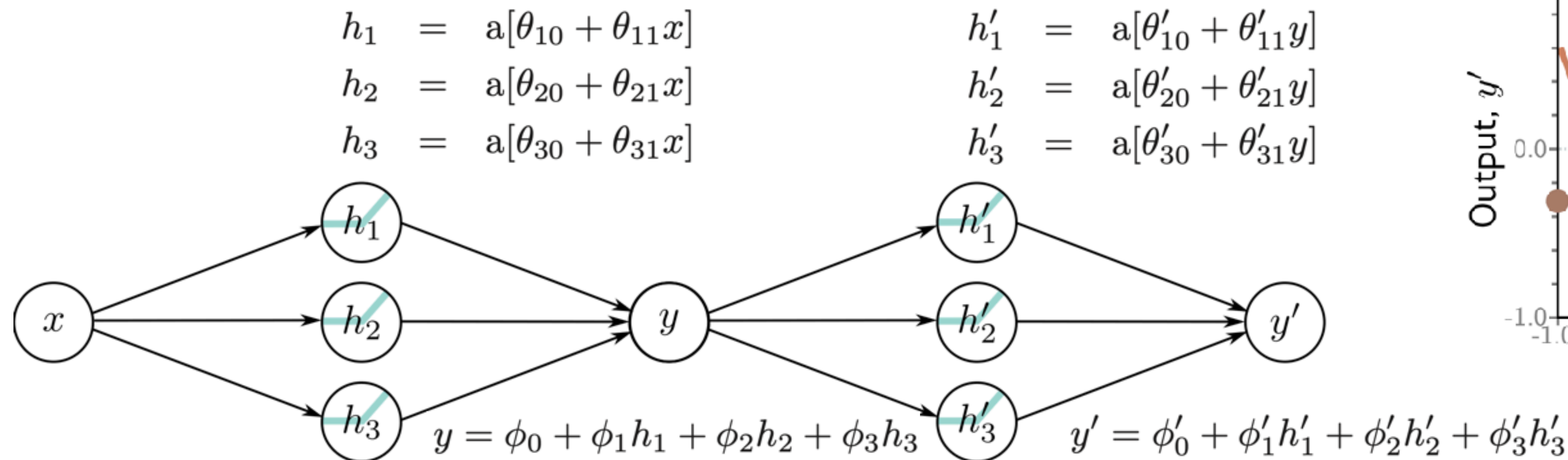
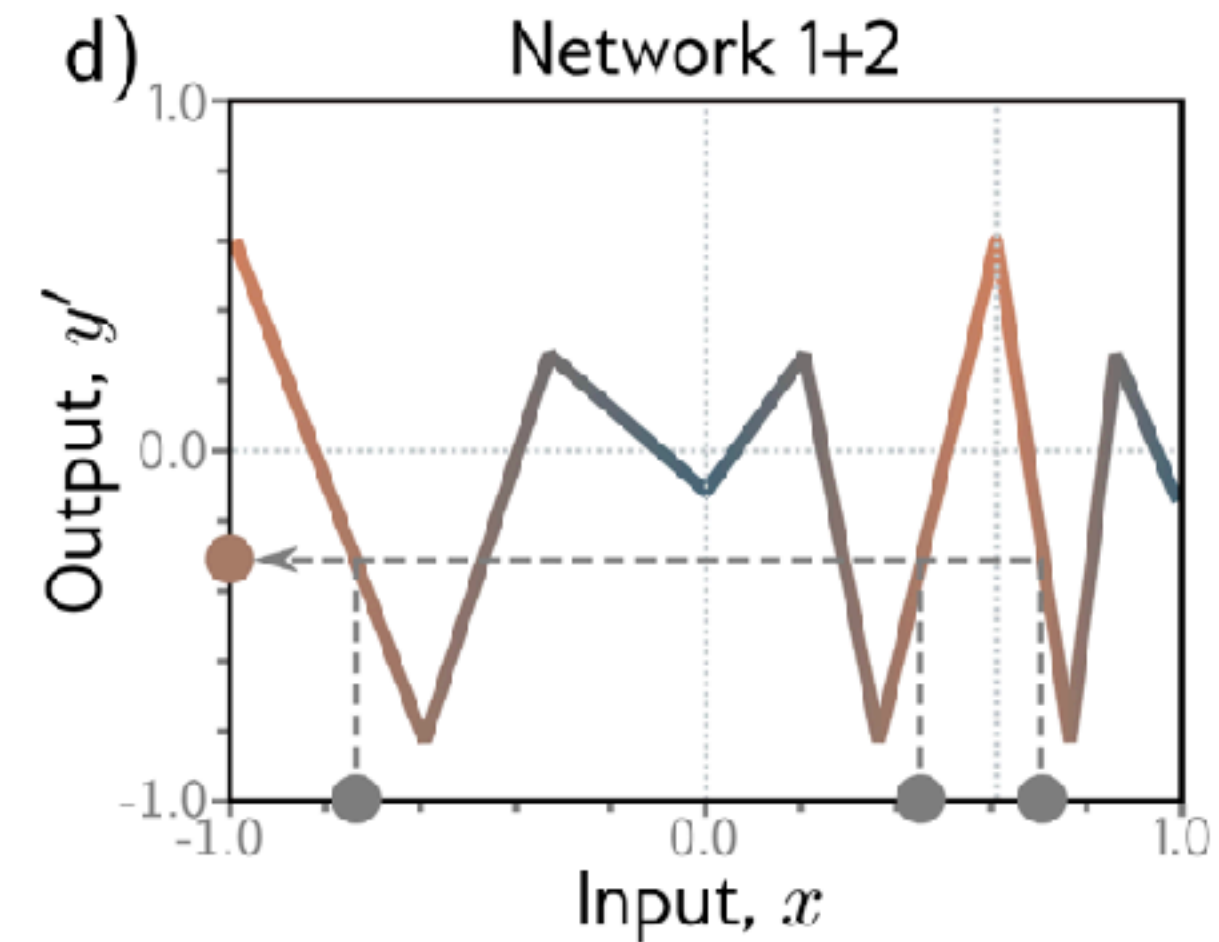
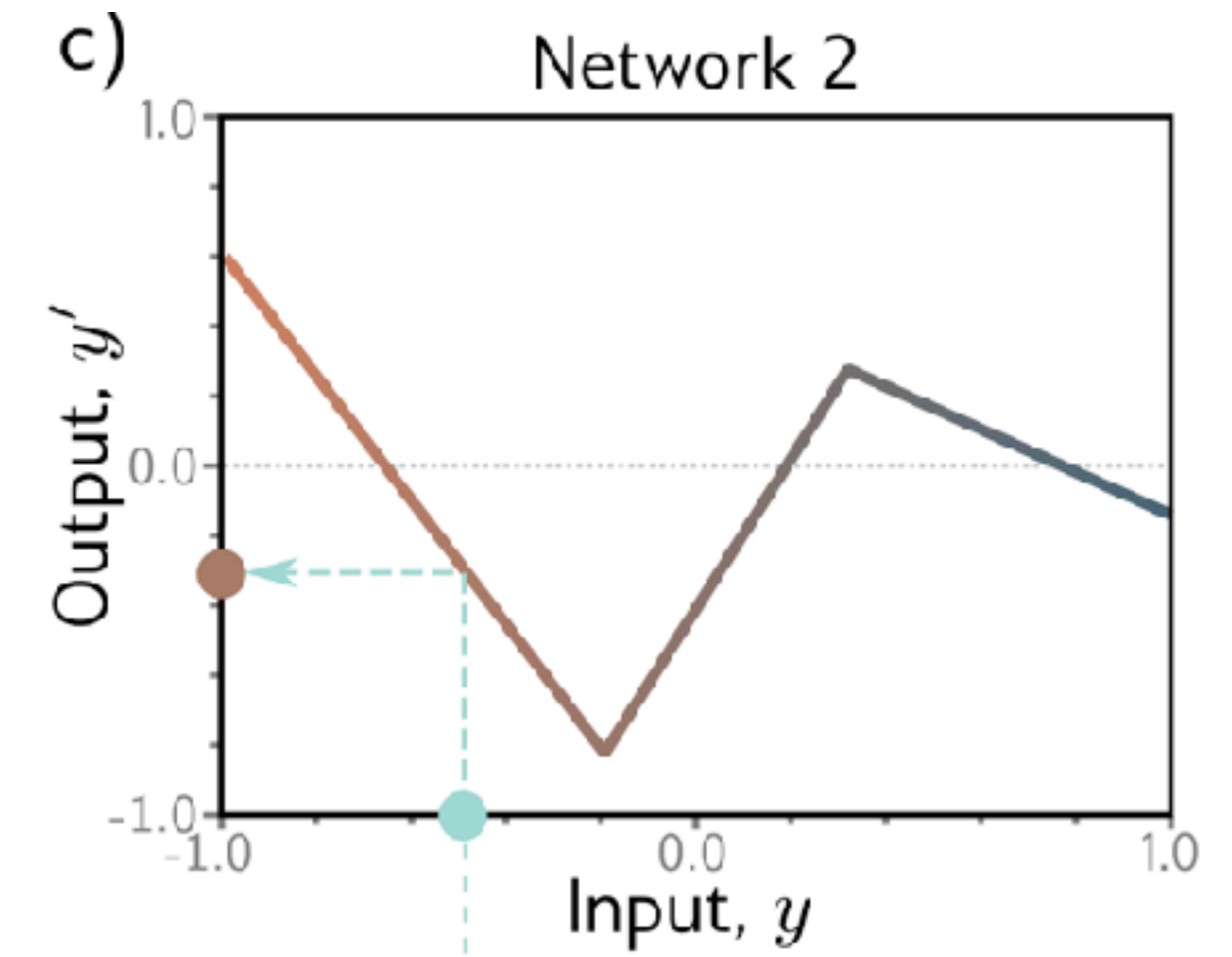
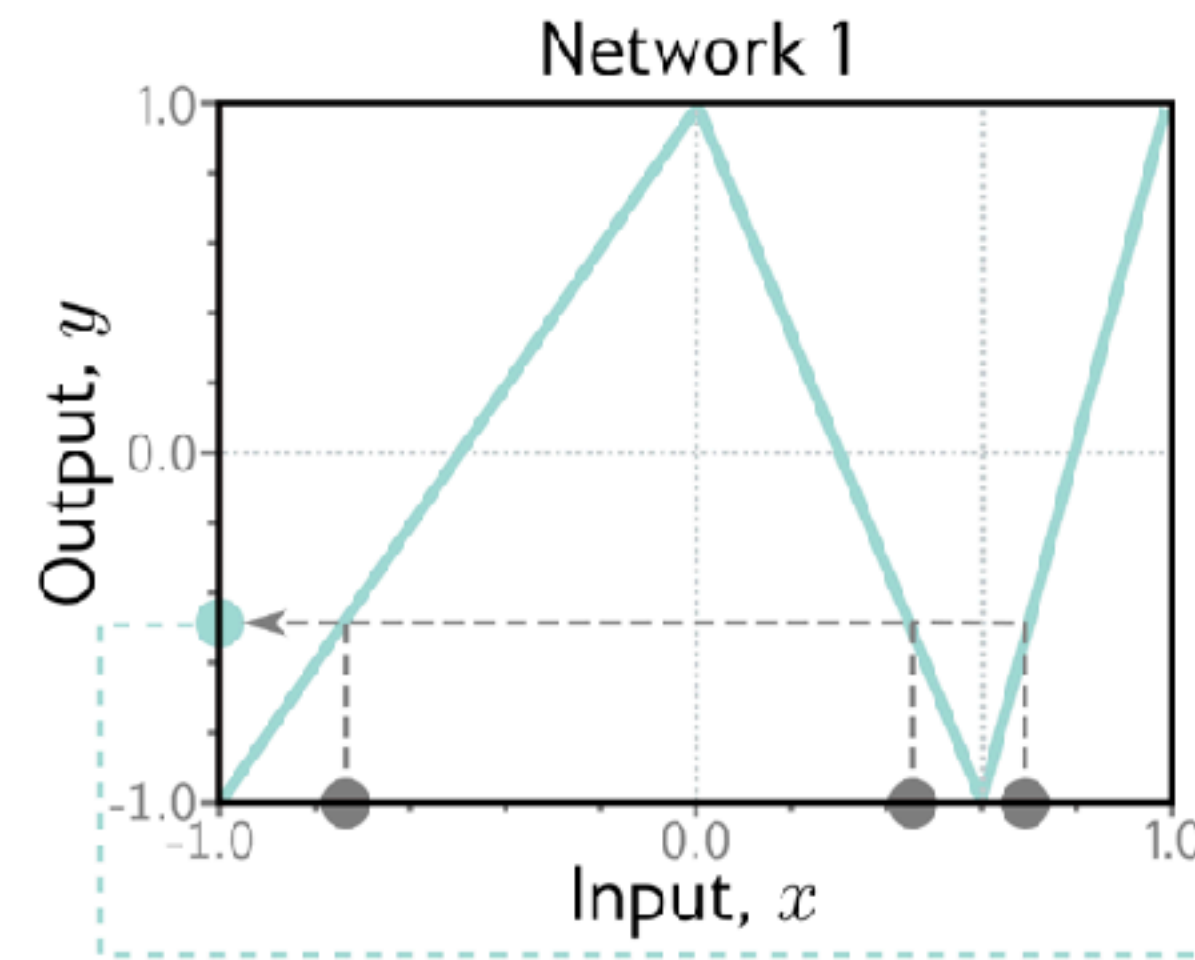
(∞ neuronra akár konvex kvadratikusan loss-hoz és lineáris regresszióhoz is konvergálhat — **Neural Tangent Kernel (NTK)**)

Érdeklődőknek: [LINK](#)

Neurális Háló

Többrétegű (“mély”) háló

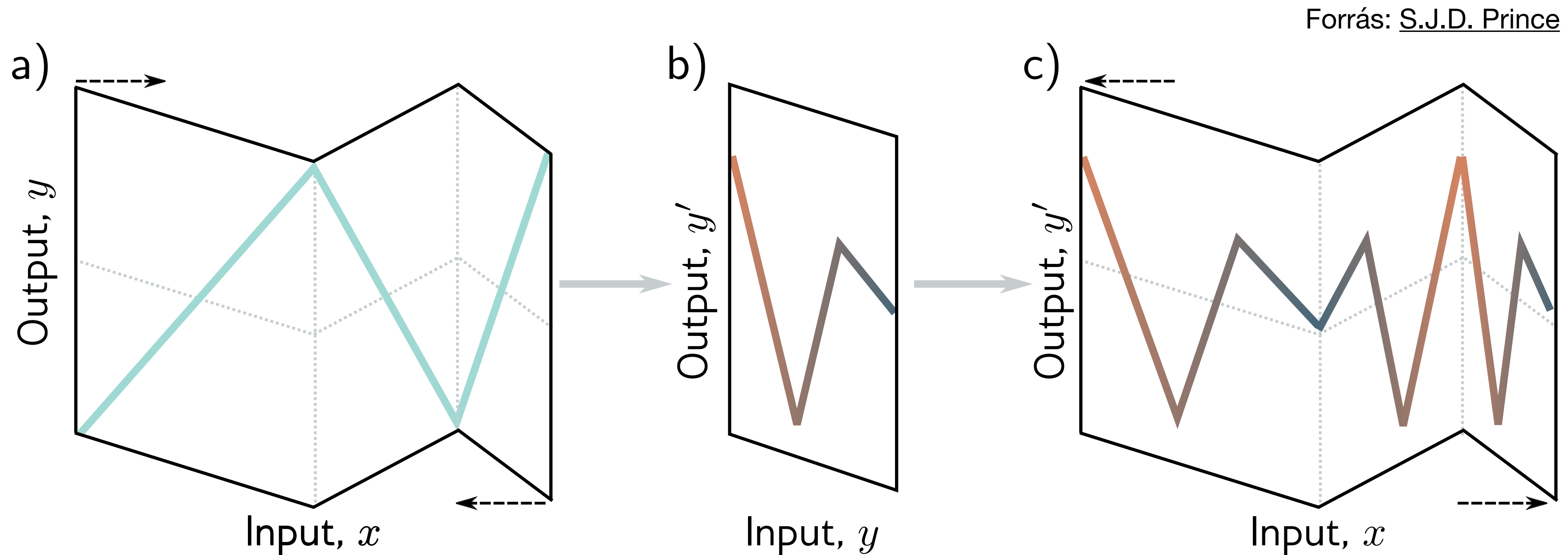
Forrás: S.J.D. Prince



Applet: [LINK](#)

Neurális Háló

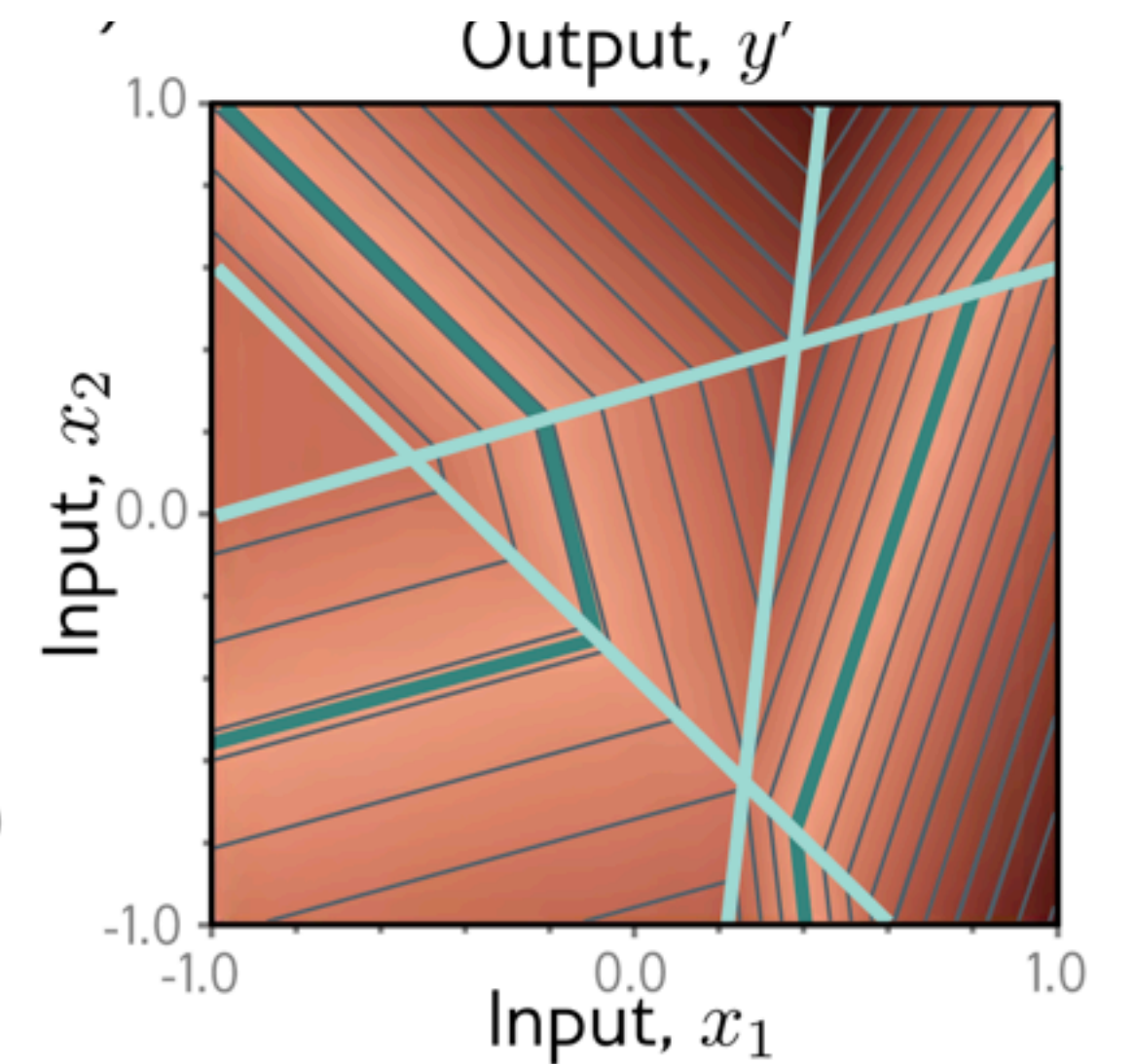
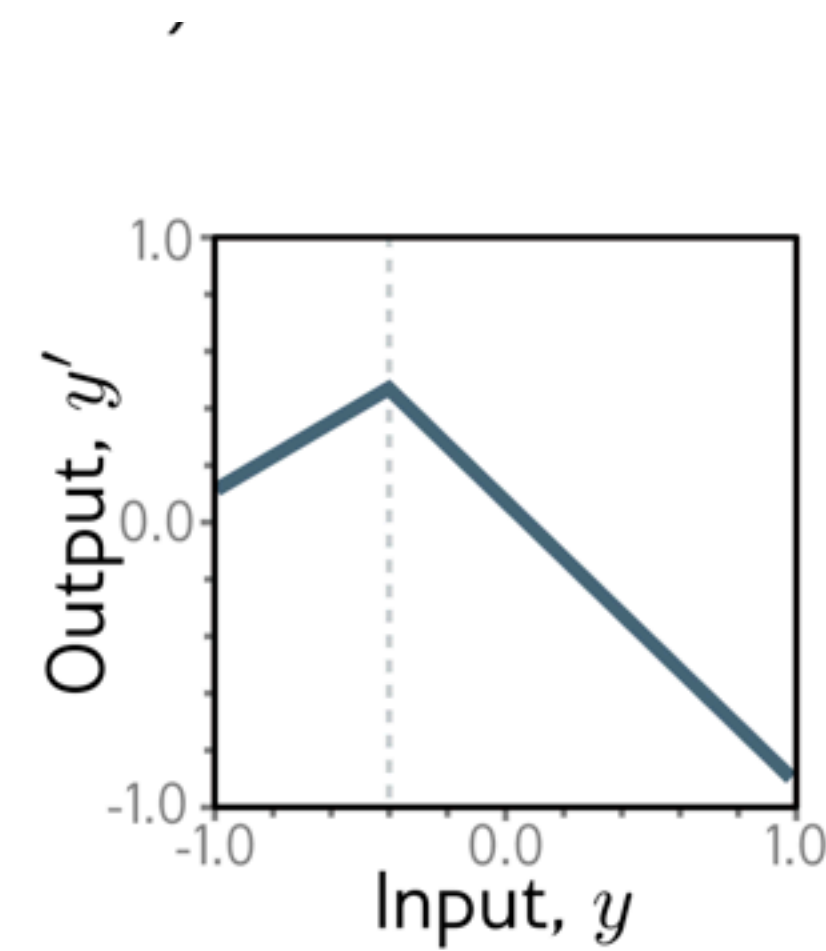
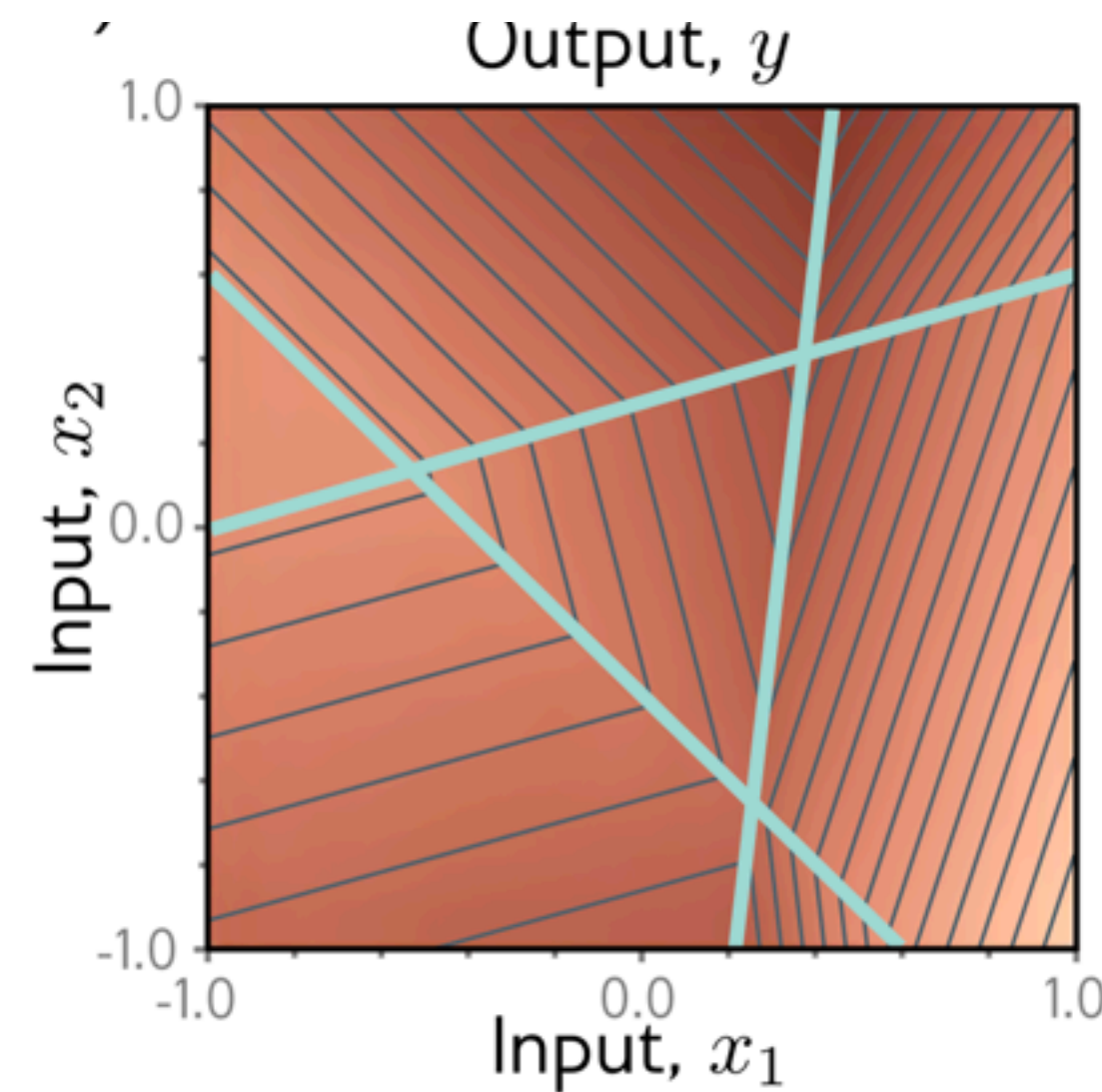
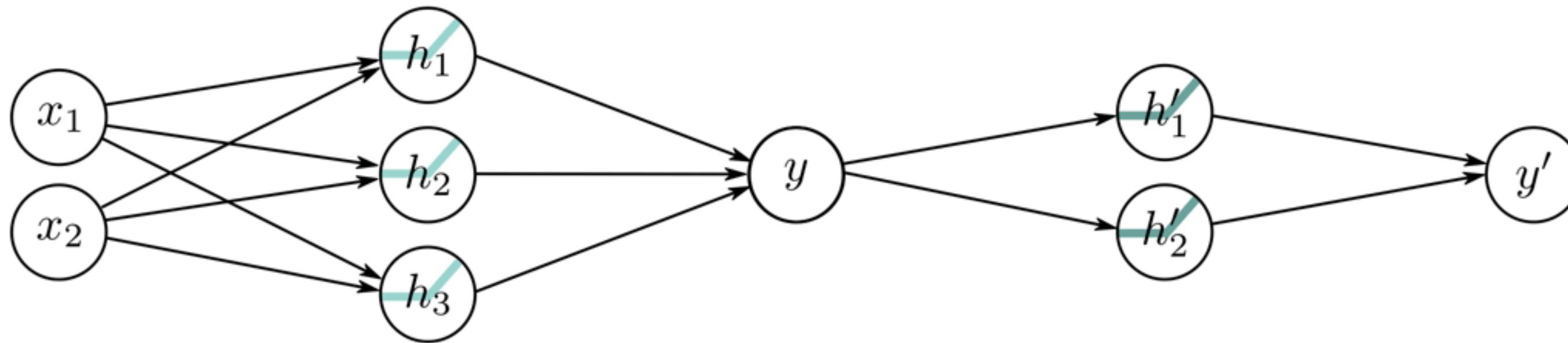
Többrétegű (“mély”) háló



Az egyes rétegek “összehajtogatják” a bemeneti teret

Neurális Háló

Többrétegű (“mély”) háló



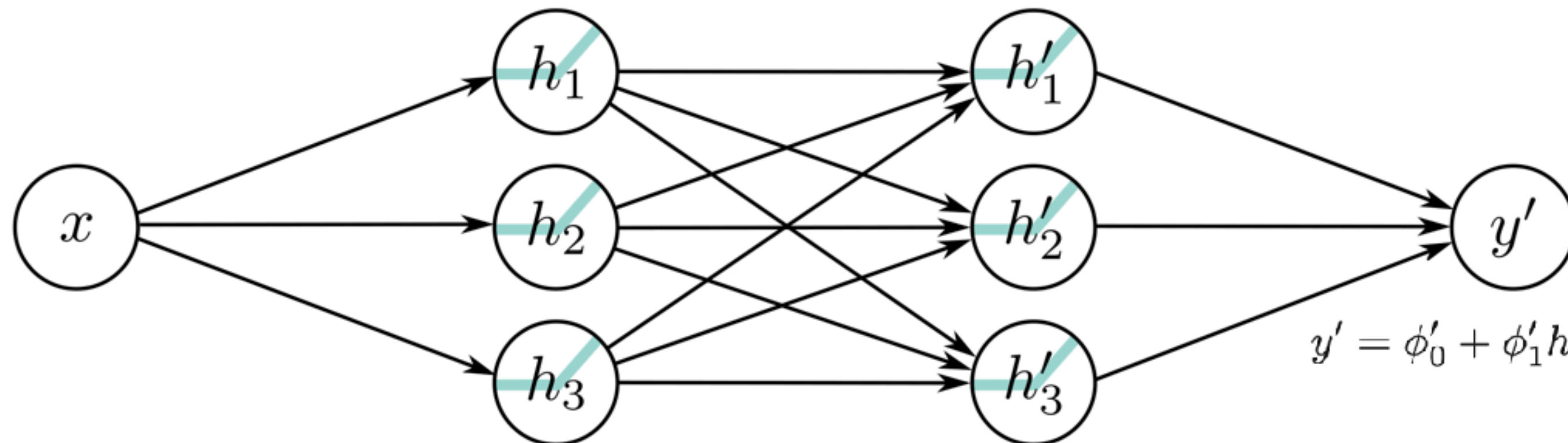
Forrás: [S.J.D. Prince](#)

Neurális Háló

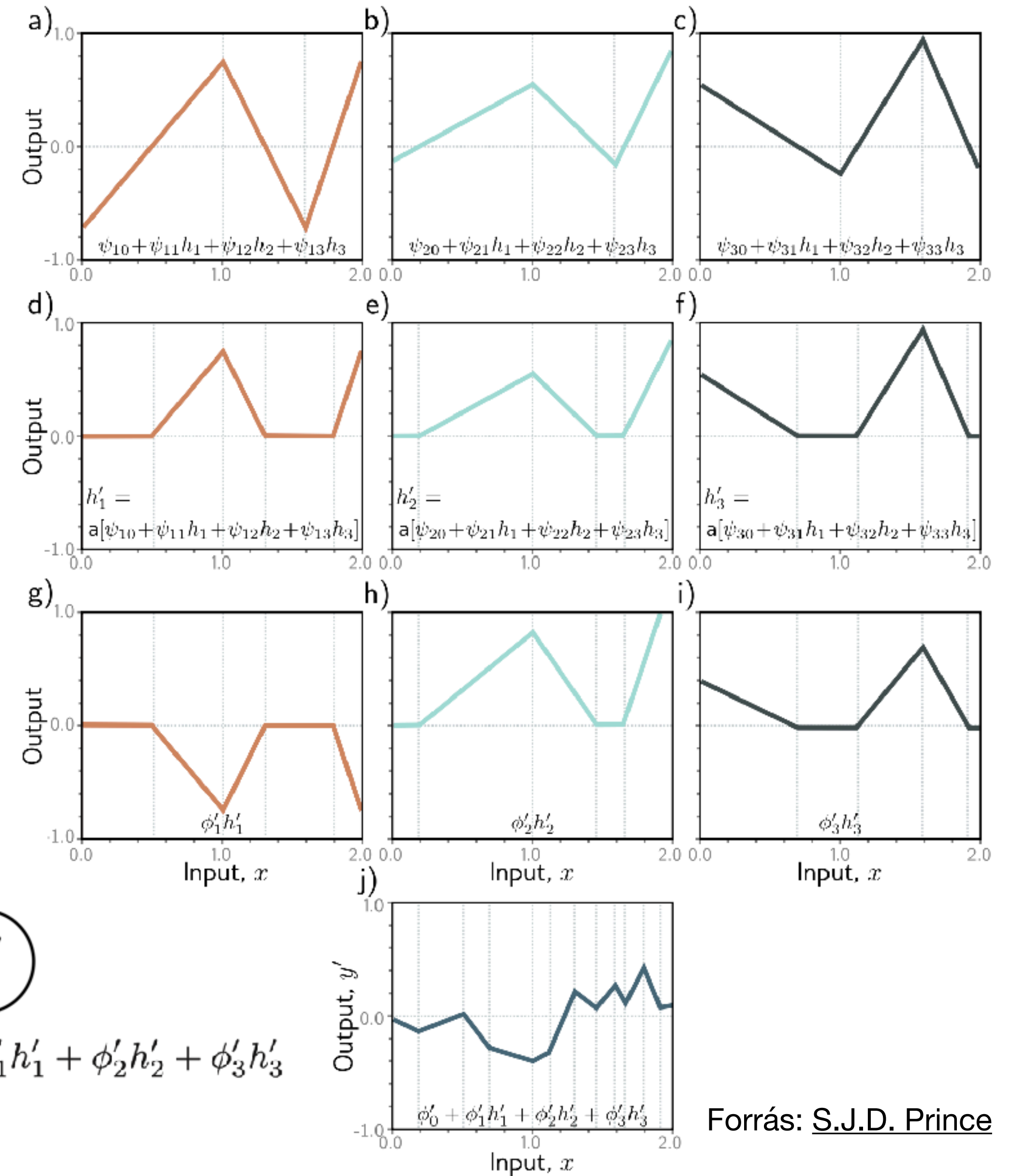
Többrétegű (“mély”) háló

Applet: [LINK](#)

$$\begin{aligned}
 h_1 &= a[\theta_{10} + \theta_{11}x] & h'_1 &= a[\psi_{10} + \psi_{11}h_1 + \psi_{12}h_2 + \psi_{13}h_3] \\
 h_2 &= a[\theta_{20} + \theta_{21}x] & h'_2 &= a[\psi_{20} + \psi_{21}h_1 + \psi_{22}h_2 + \psi_{23}h_3] \\
 h_3 &= a[\theta_{30} + \theta_{31}x] & h'_3 &= a[\psi_{30} + \psi_{31}h_1 + \psi_{32}h_2 + \psi_{33}h_3]
 \end{aligned}$$



$$y' = \phi'_0 + \phi'_1 h'_1 + \phi'_2 h'_2 + \phi'_3 h'_3$$



Forrás: S.J.D. Prince

Neurális Háló

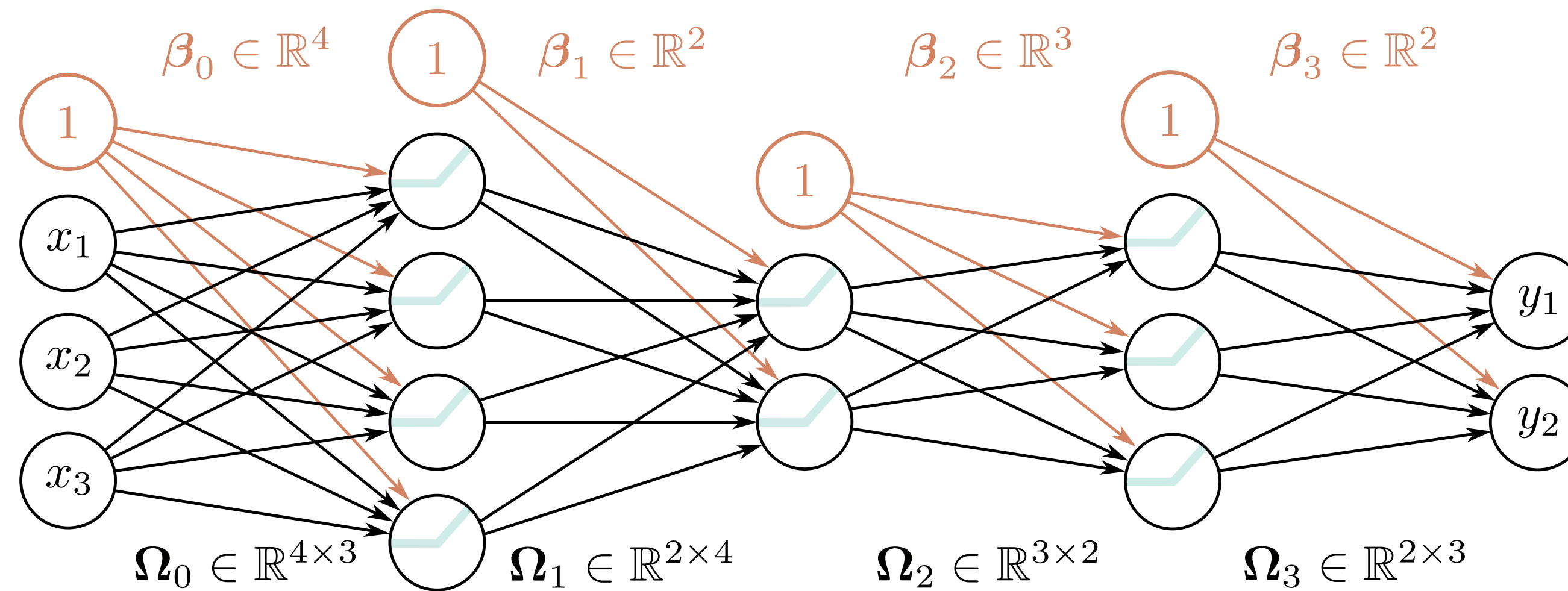
Többrétegű (“mély”) háló

Multi-Layer Perceptron (MLP)

Aka “Fully Connected Network”

$$y = \beta_K + \Omega_K a [\beta_{K-1} + \Omega_{K-1} a [\dots \beta_2 + \Omega_2 a [\beta_1 + \Omega_1 a [\beta_0 + \Omega_0 x] \dots]]]$$

$$h_1 = a[\beta_0 + \Omega_0 x] \quad h_2 = a[\beta_1 + \Omega_1 h_1] \quad h_3 = a[\beta_2 + \Omega_2 h_2]$$



Depth \longleftrightarrow
 Width \updownarrow
 “Hiperparaméterek”

$$y = \beta_K + \Omega_K h_K$$

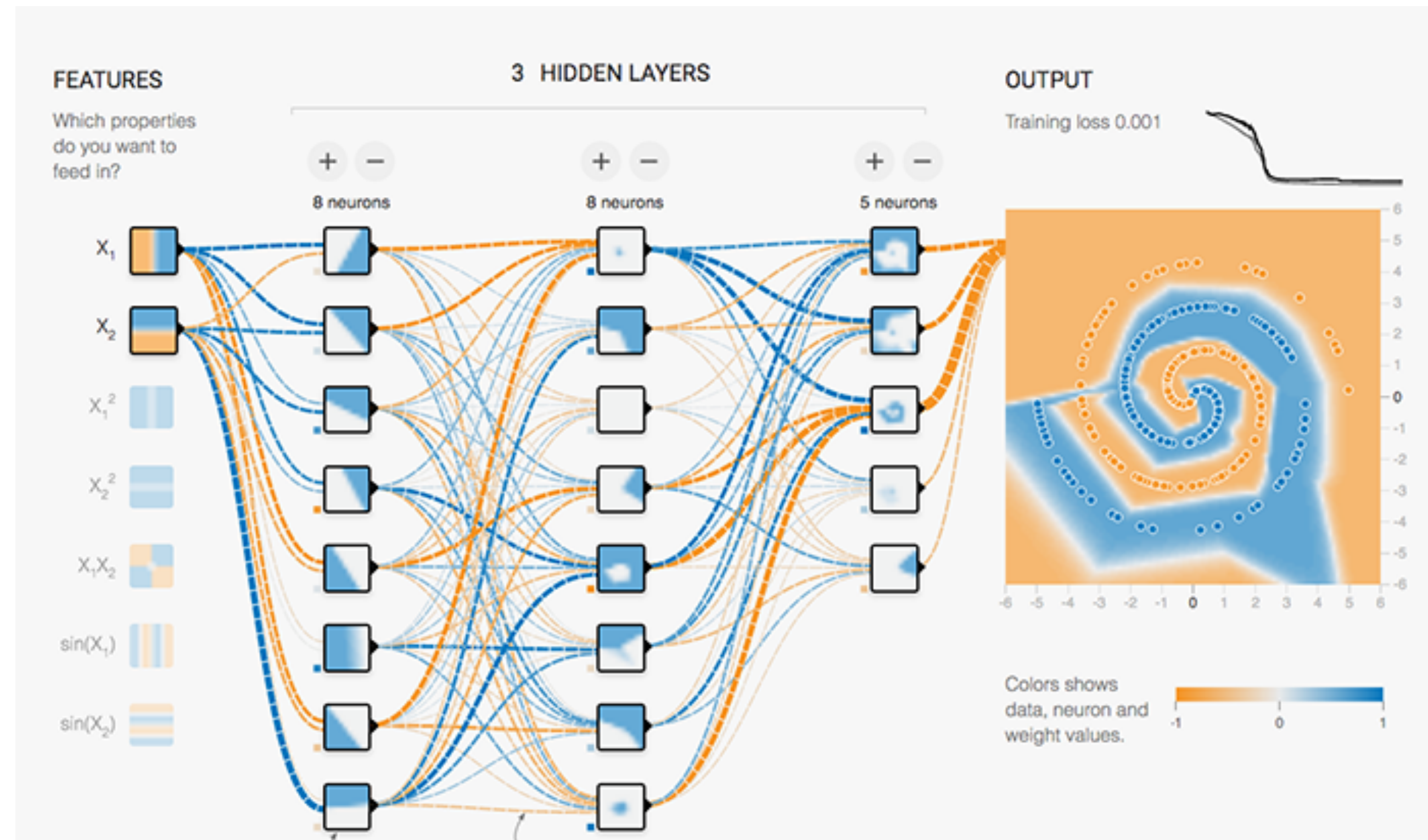
Súly mátrixok

Input, \mathbf{x}	Hidden layer, \mathbf{h}_1	Hidden layer, \mathbf{h}_2	Hidden layer, \mathbf{h}_3	Output, \mathbf{y}
$D_i = 3$	$D_1 = 4$	$D_2 = 2$	$D_3 = 3$	$D_o = 2$

Forrás: S.J.D. Prince

Neurális Háló

Multi-Layer Perceptron (MLP) – Applet

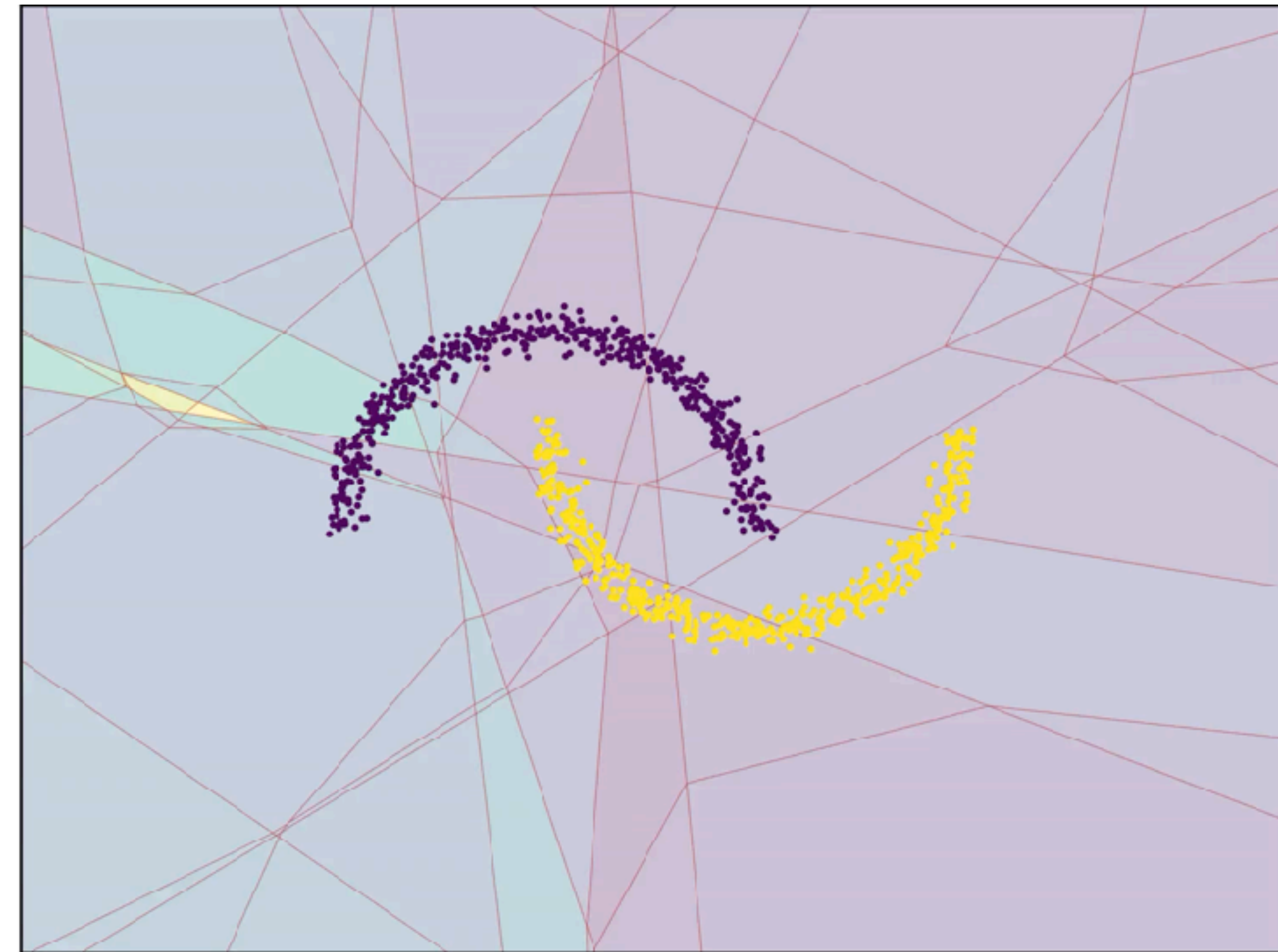
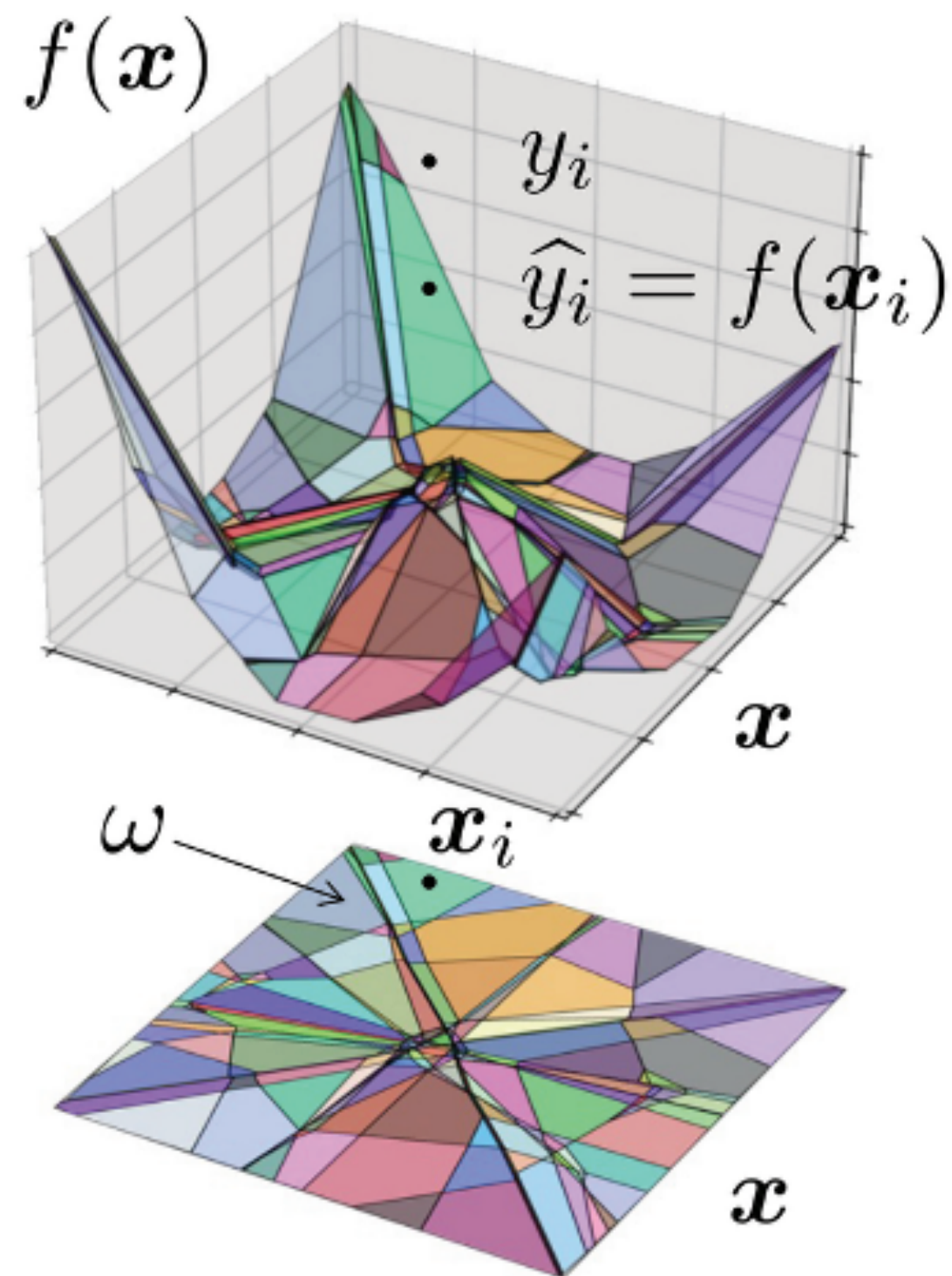


<https://playground.tensorflow.org/>

Neurális Háló

Multi-Layer Perceptron (MLP) – Geometria

Forrás: [R. Baraniuk](#)



ReLU MLP = szakaszosan lineáris függvény (spline)!

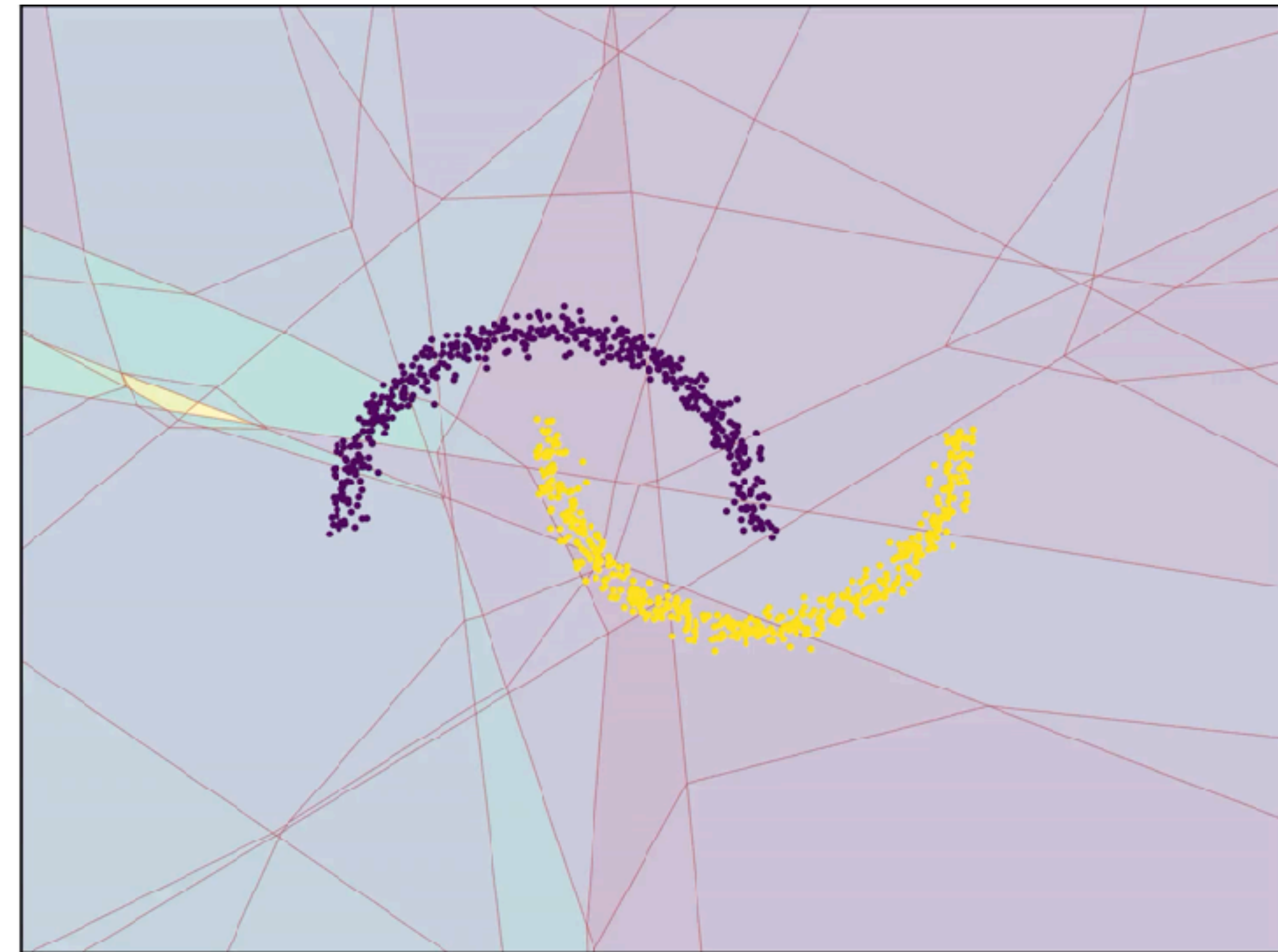
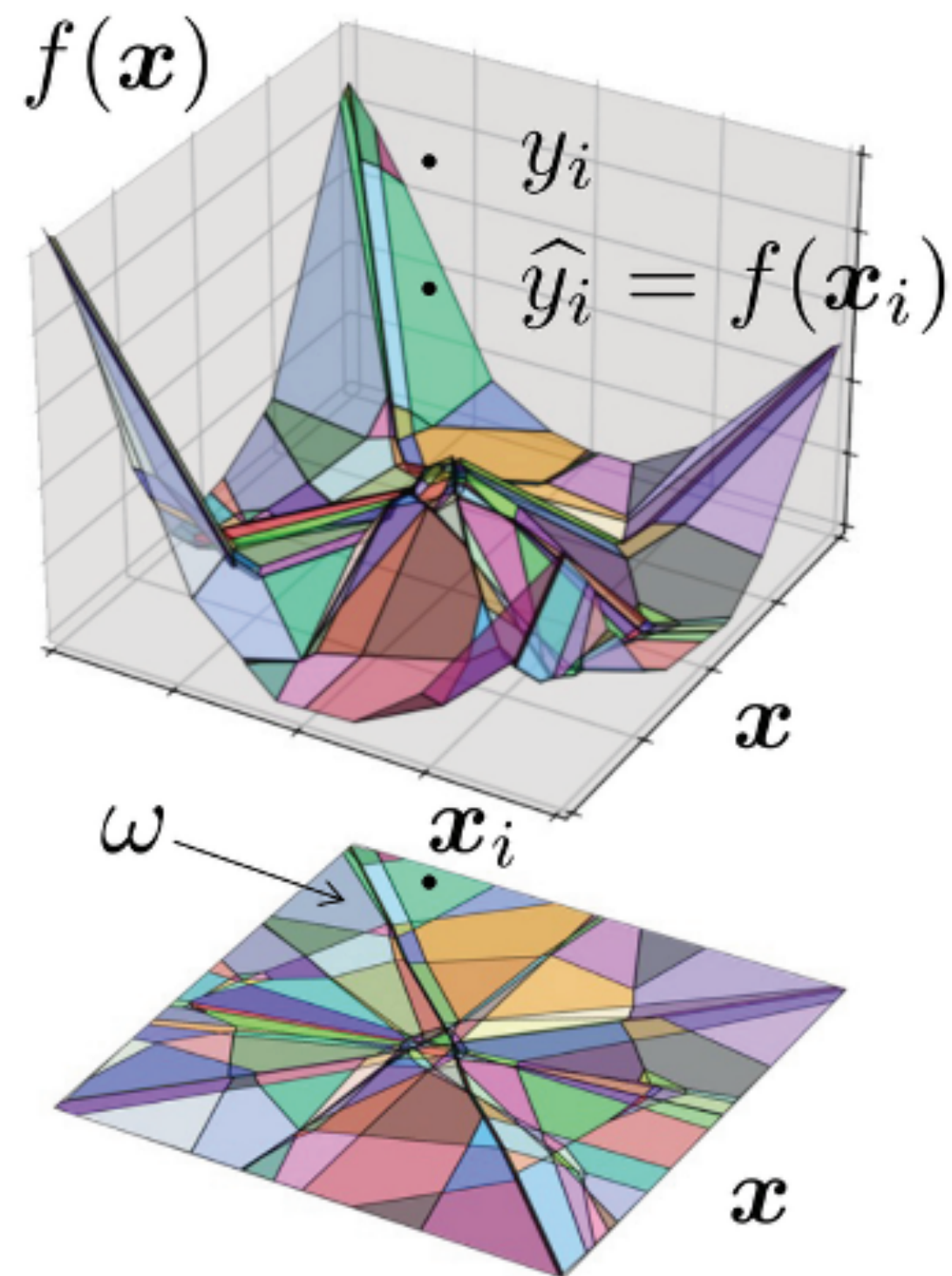
5 rétegű MLP

Érdeklődőknek: [LINK](#)

Neurális Háló

Multi-Layer Perceptron (MLP) – Geometria

Forrás: [R. Baraniuk](#)



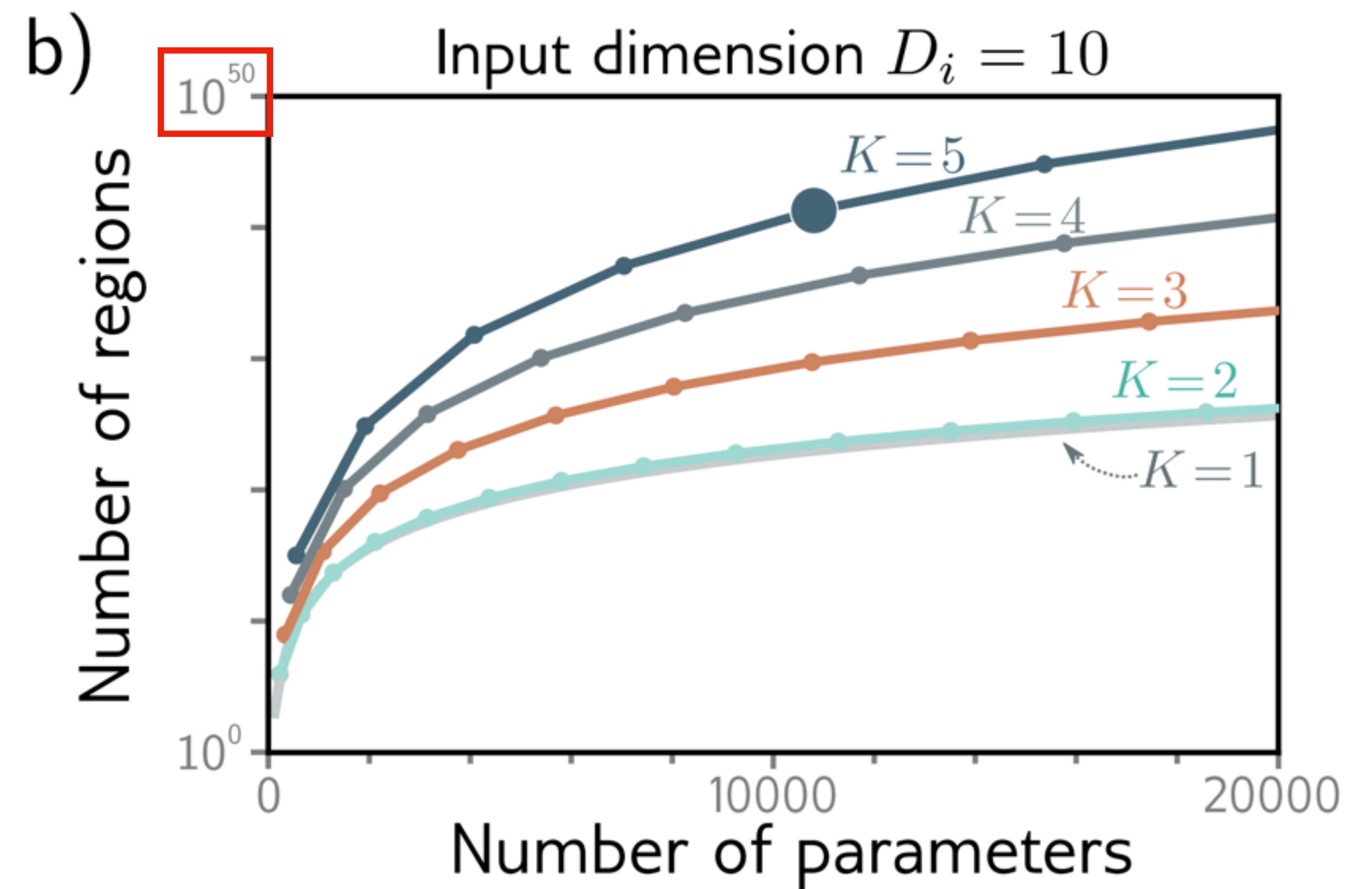
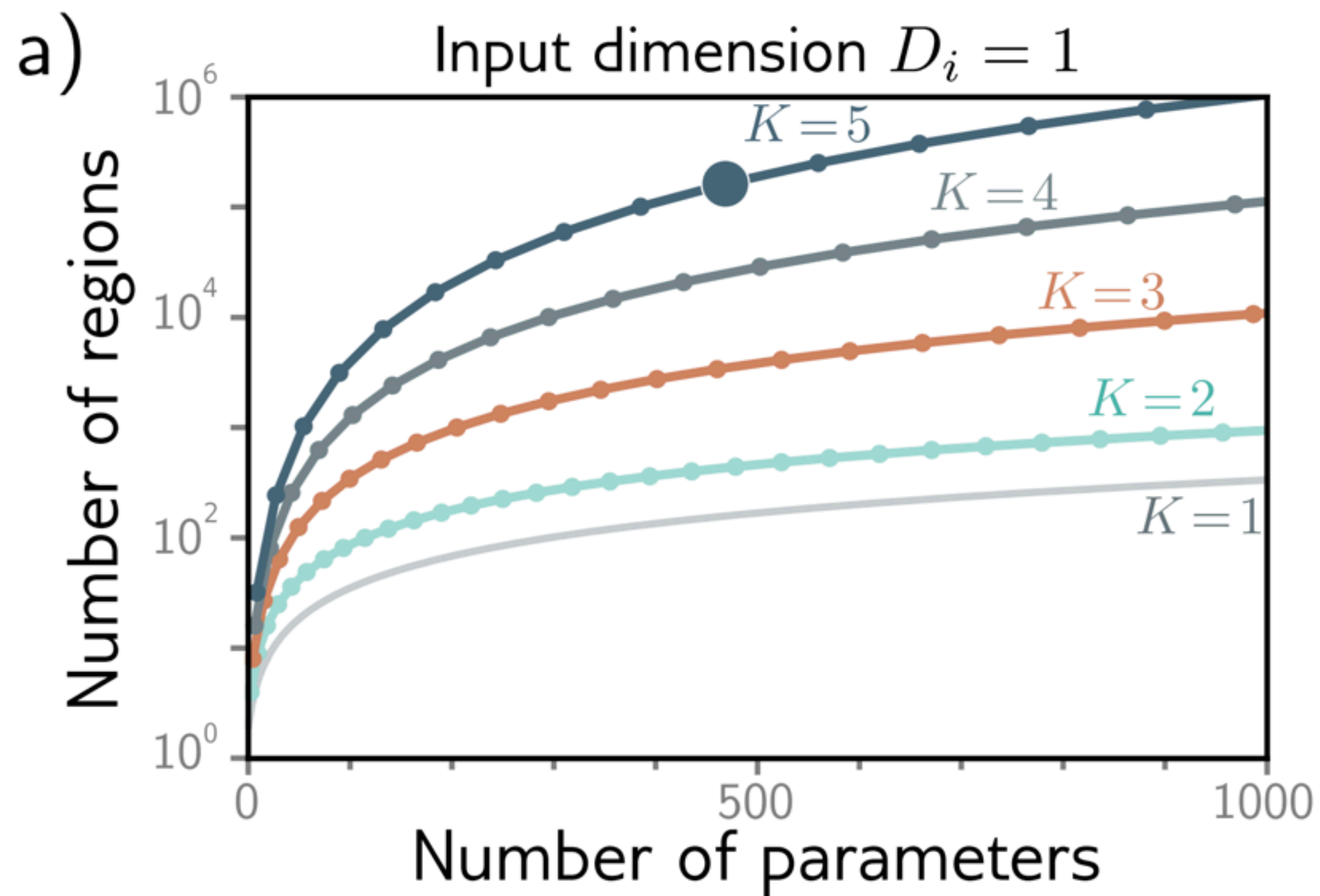
ReLU MLP = szakaszosan lineáris függvény (spline)!

5 rétegű MLP

Érdeklődőknek: [LINK](#)

Neurális Háló

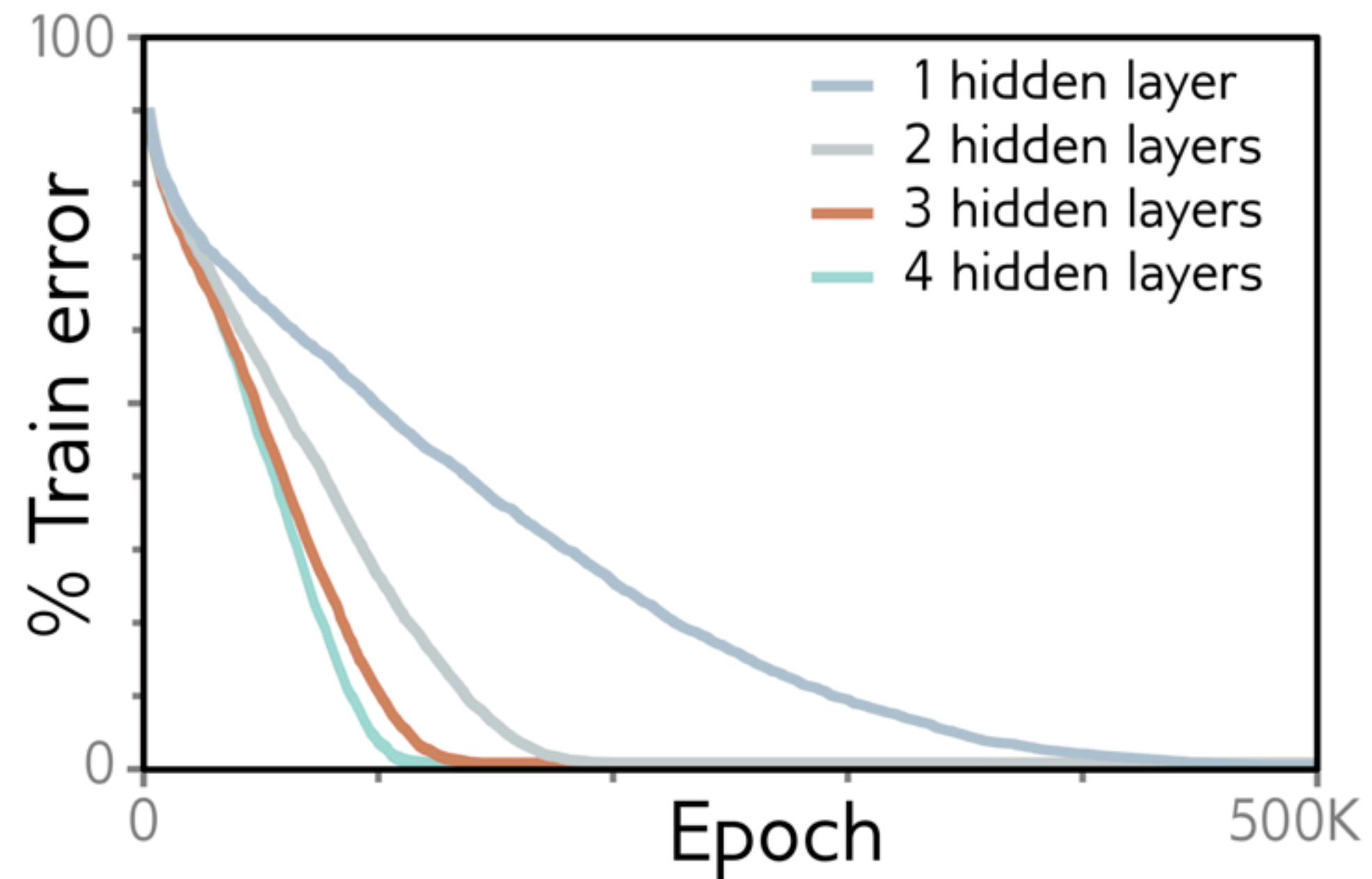
Mélység hatása



Forrás: S.J.D. Prince

Neurális Hálók

Mélység hatása

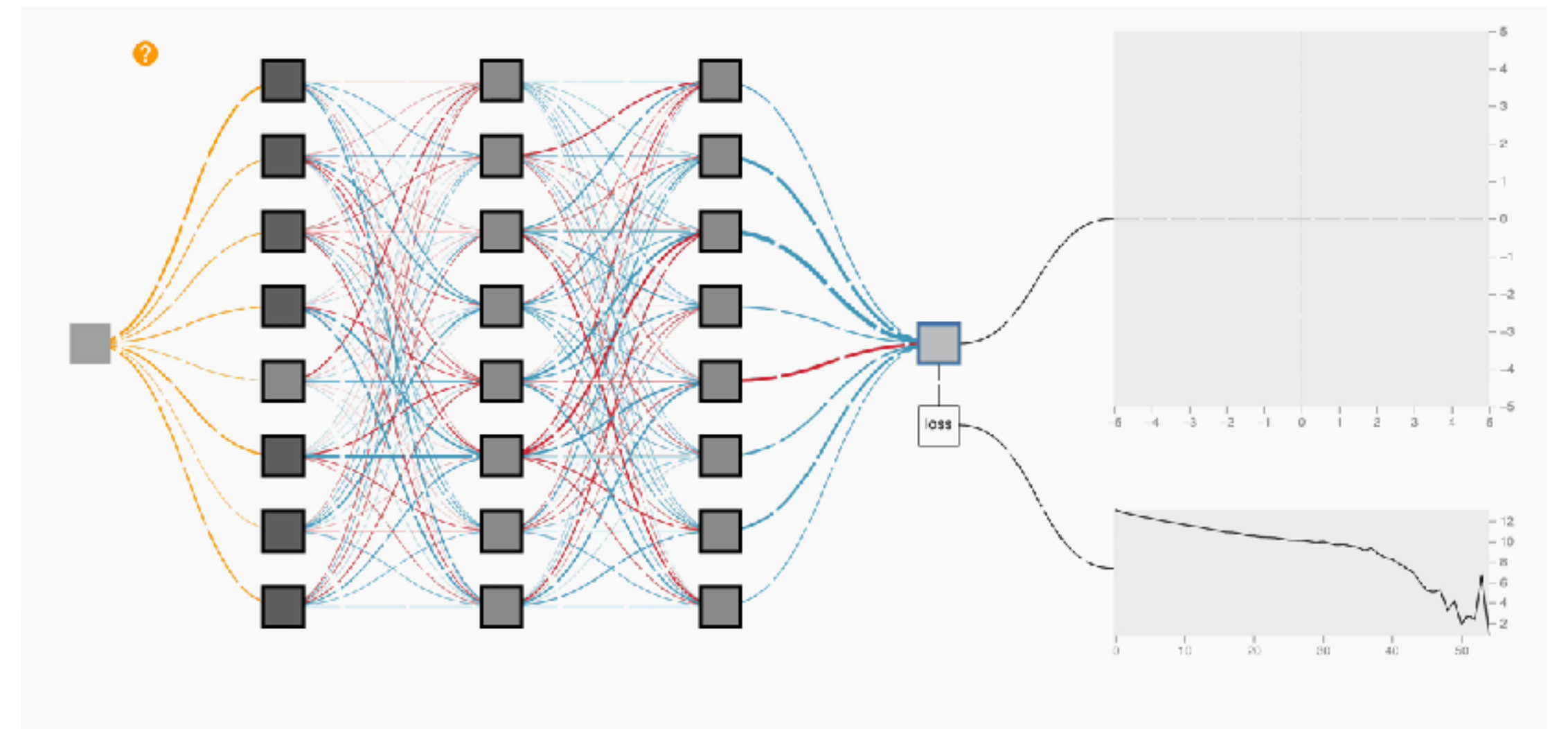


Megfelelő architektúrával (reziduális kapcsolatokkal) a mélyebb hálók gyorsabban tanulnak!

Neurális Háló

Tanítás – Backpropagation

- Neurális háló tanítása gradiens-alapú módszerekkel történik:
 - **Mini-batch SGD / ADAM**
 - Kisebb hálókra esetleg (kvázi-)Newton (pl. L-BFGS)
- Gradiensek számítása:
backpropagation – reverse módú automatikus differenciálás speciális esete!

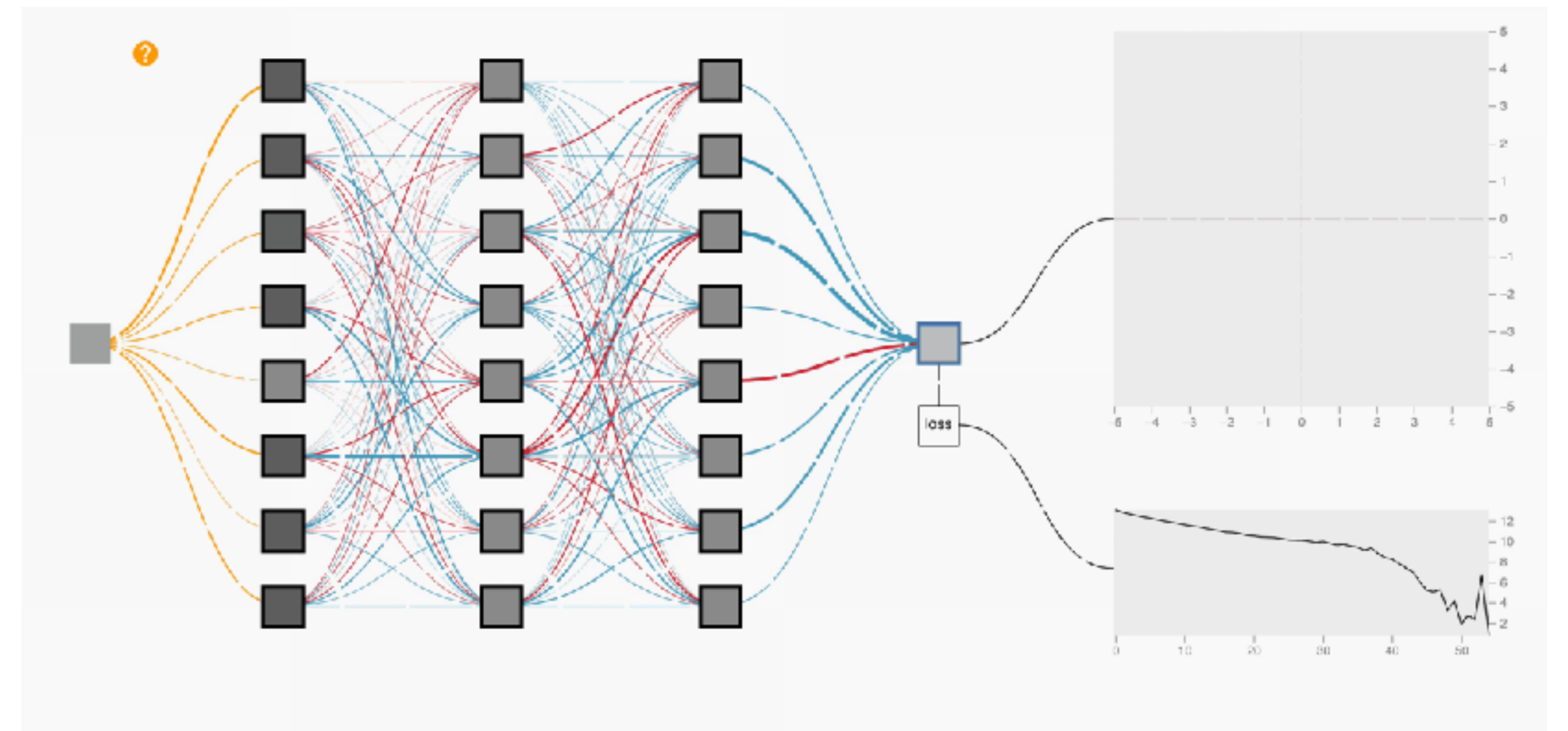


<https://xnought.github.io/backprop-explainer/>

Neurális Háló

Tanítás – Backpropagation

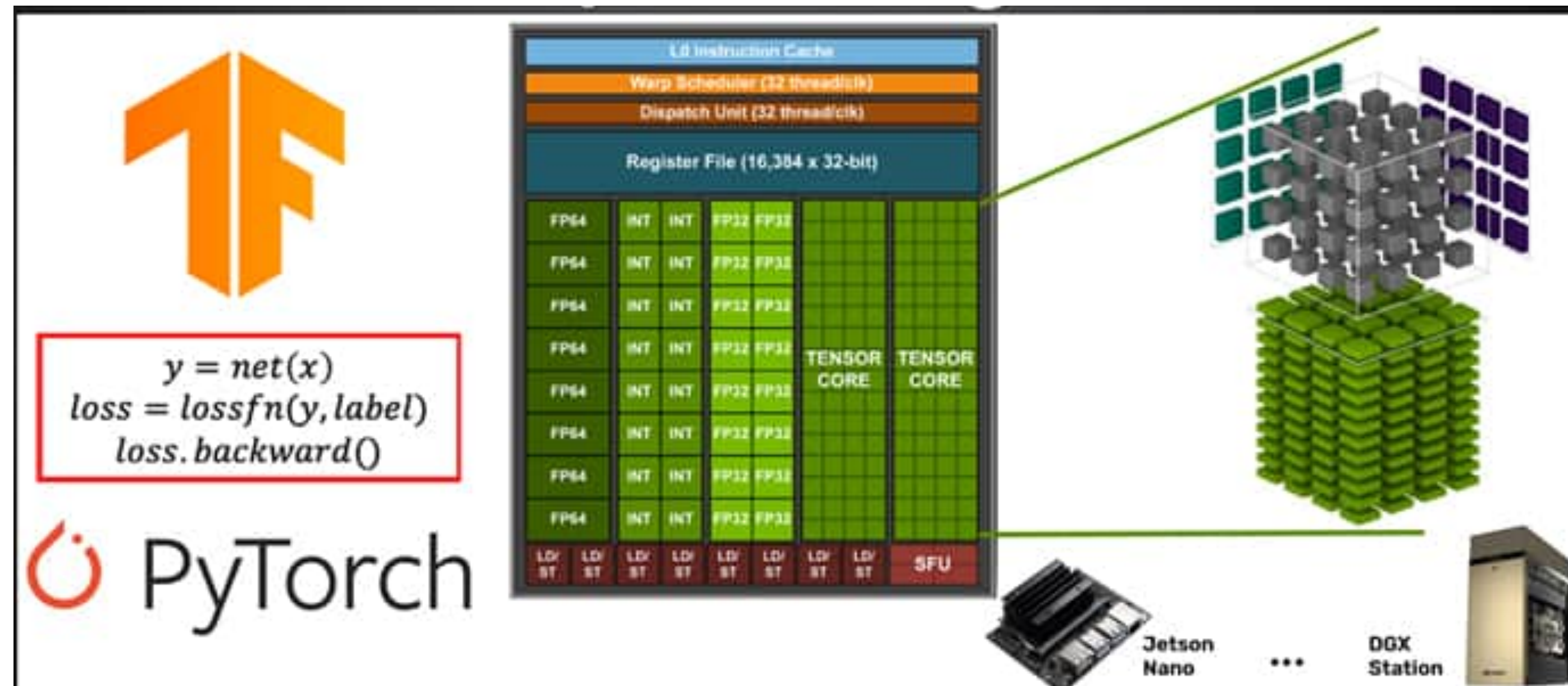
- Neurális háló tanítása gradiens-alapú módszerekkel történik:
 - **Mini-batch SGD / ADAM**
 - Kisebb hálókra esetleg (kvázi-)Newton (pl. L-BFGS)
- Gradiensek számítása:
backpropagation – reverse módú automatikus differenciálás speciális esete!



<https://xnought.github.io/backprop-explainer/>

Neurális Hálók

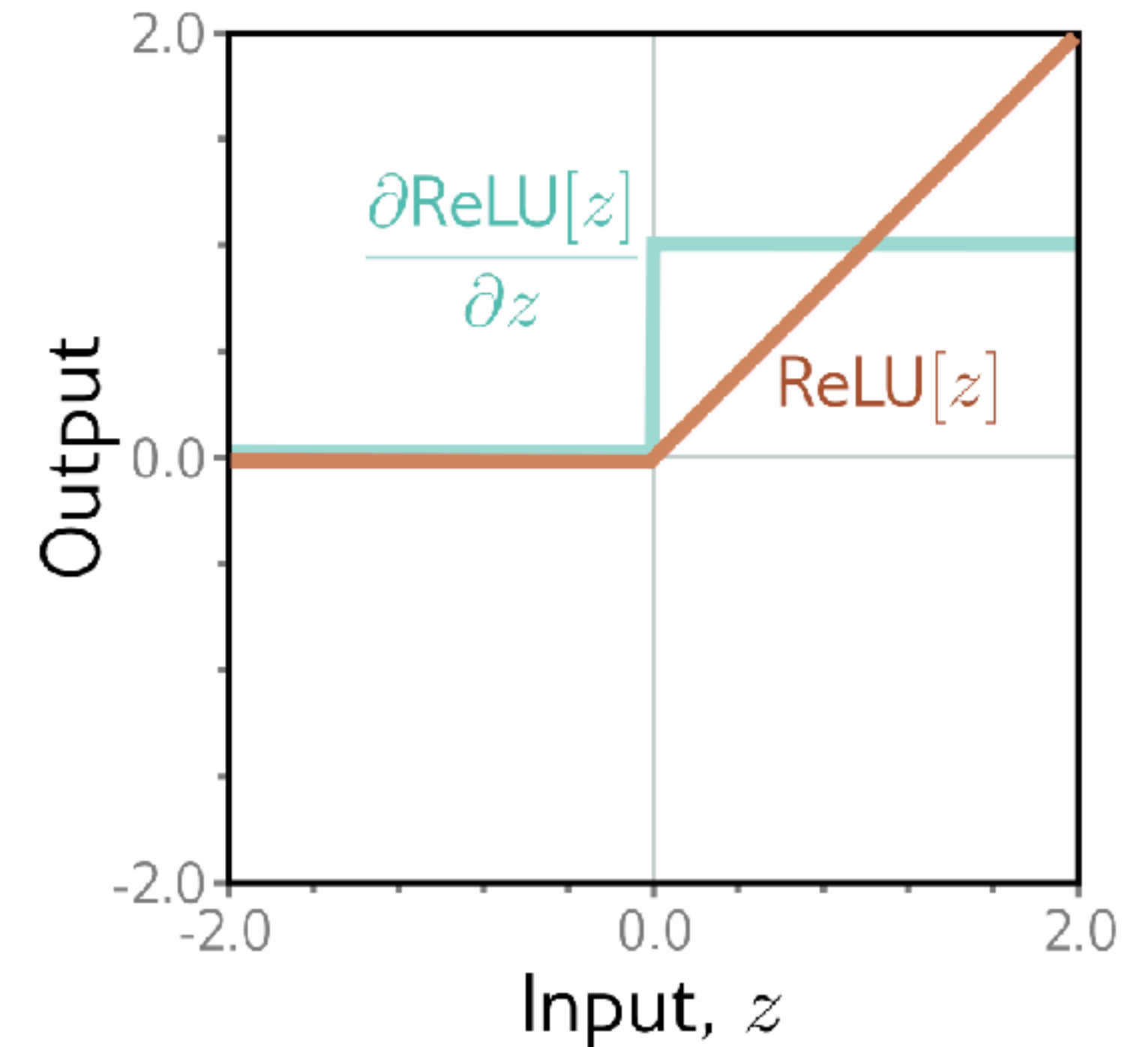
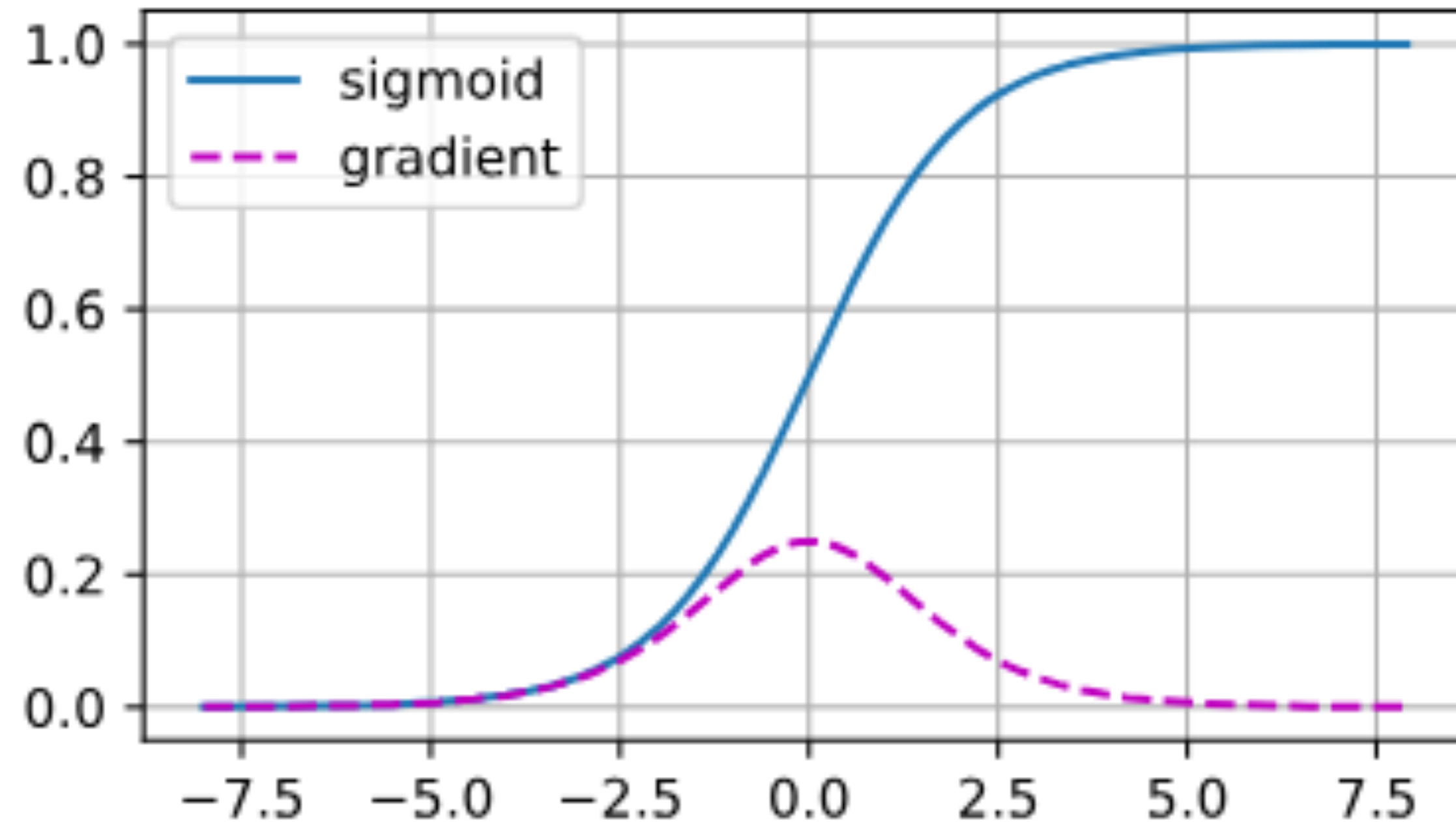
Tanítás – Párhuzamos hardver szerepe



Neurális hálók kiértékelése / tanítása:
Ideális feladat a SIMD párhuzamos hardver (pl. GPU) számára!

Neurális Hálók

Tanítás – Aktivációs függvények szerepe

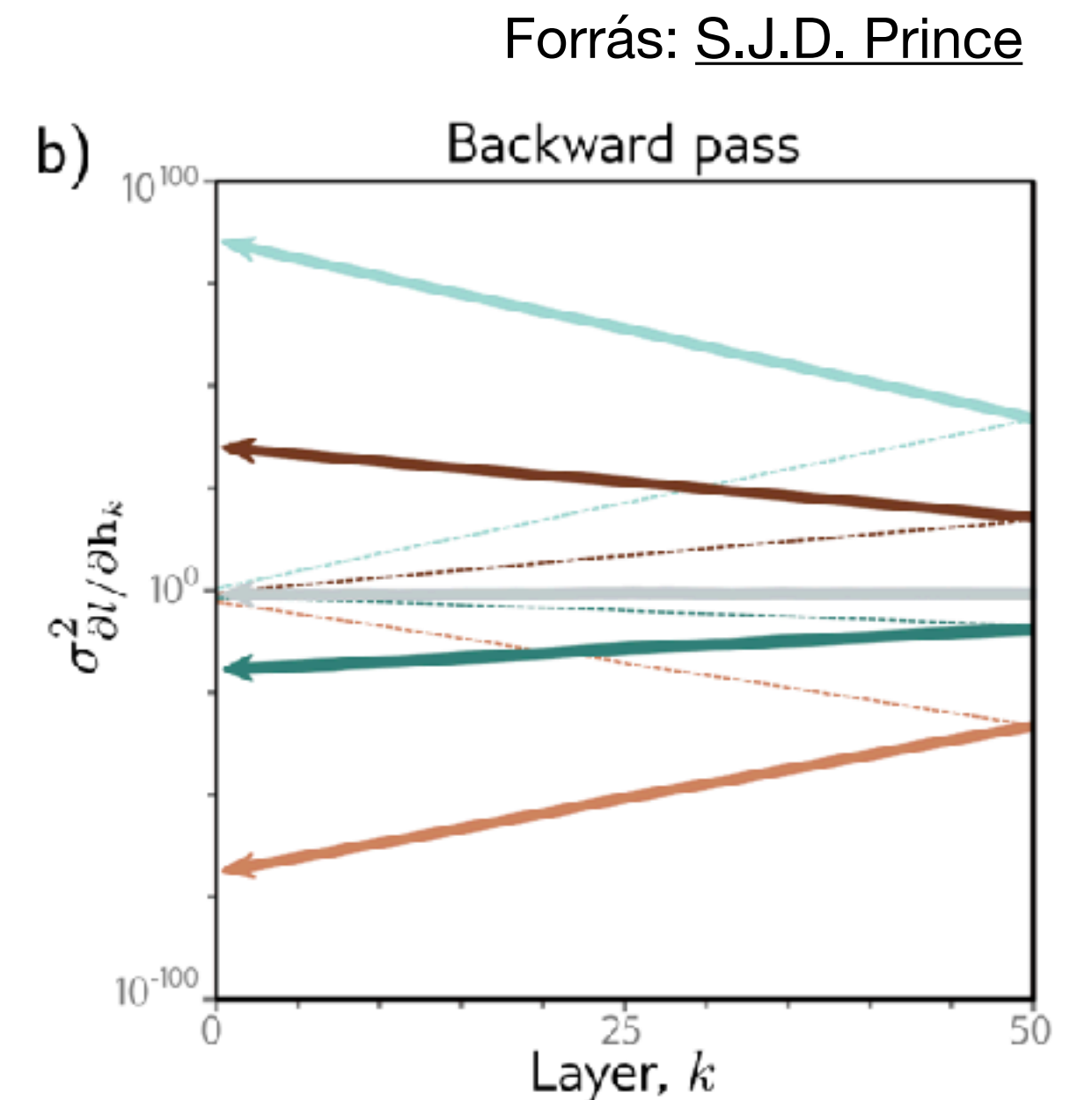
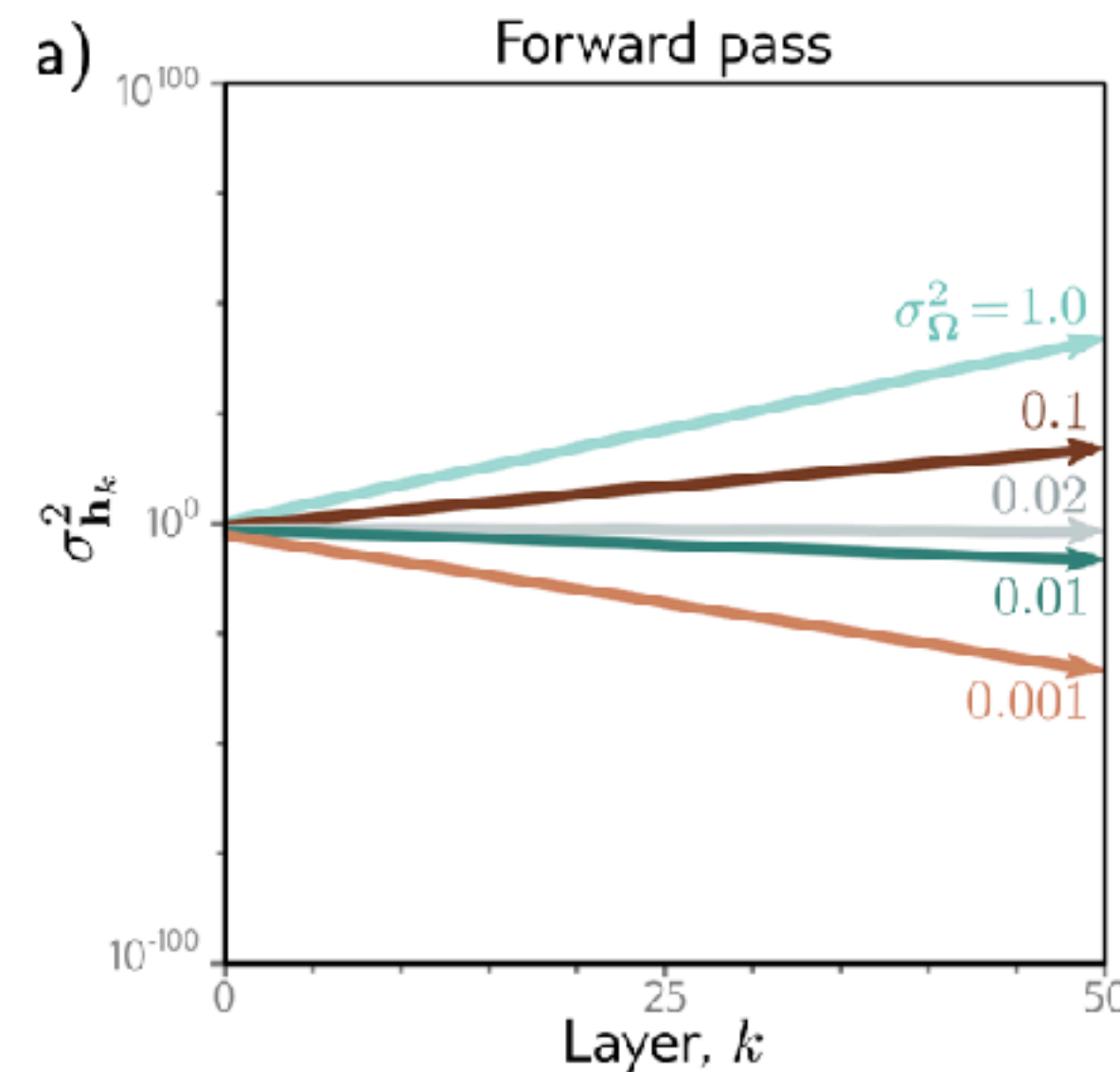


Klasszikus aktivációs függvények (sigmoid/tanh/stb.): **eltűnő gradiens** probléma!
Modern aktivációs függvényekkel (ReLU, stb.) ez a probléma megszűnt!

Neurális Háló

Inicializációs módszerek

- Hogyan inicializáljuk a háló súlyait?
 - Csupa 0? a neuronok megkülönböztethetlenné válnak egymástól, nincs ami megtörje a szimmetriát...
 - Random súlyok: az aktivációk és a gradiensek nagysága folyamatosan csökkenhet/nőhet!
 - Xavier/Glorot/He inicializáció: $w \sim \mathcal{N}(0, \sigma_{\Omega}^2)$ (normalizált variancia)



$$\sigma_{\Omega}^2 = \frac{4}{D_h + D_{h'}}$$

Neurális Háló

Regularizáció

$$L = \frac{1}{N} \sum_i L_i + \lambda R(w) \rightarrow \min$$

Regularizáció

L2

L1

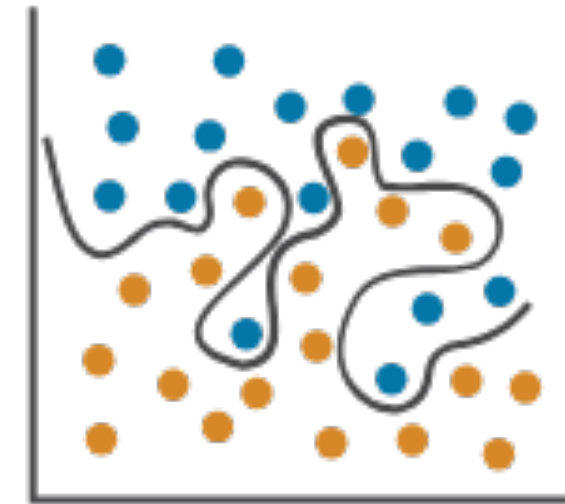
Elastic (L1+L2)

$$R(W) = \sum_k \sum_l W_{k,l}^2$$

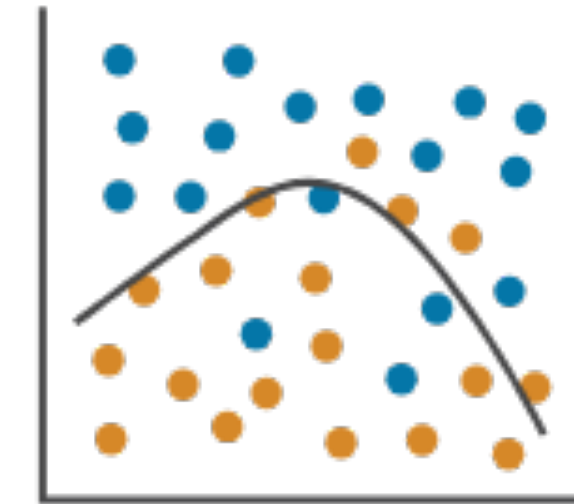
$$R(W) = \sum_k \sum_l |W_{k,l}|$$

$$R(W) = \sum_k \sum_l \beta W_{k,l}^2 + |W_{k,l}|$$

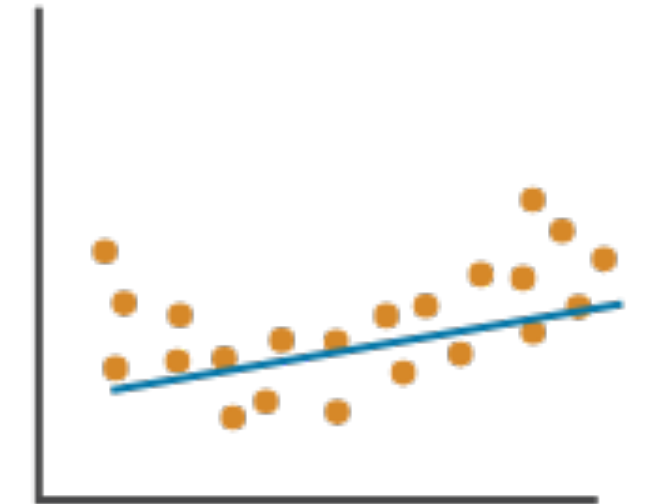
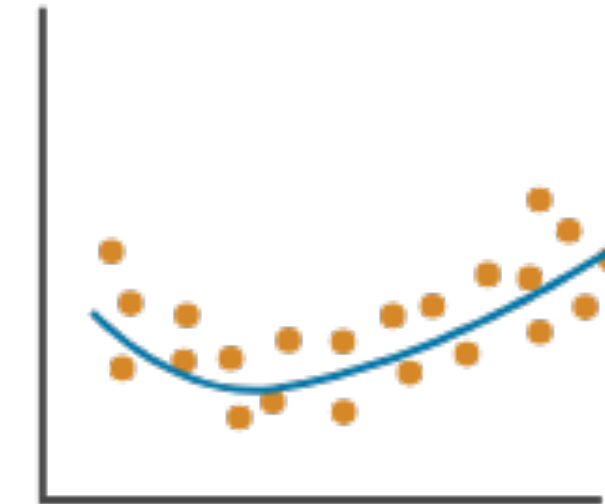
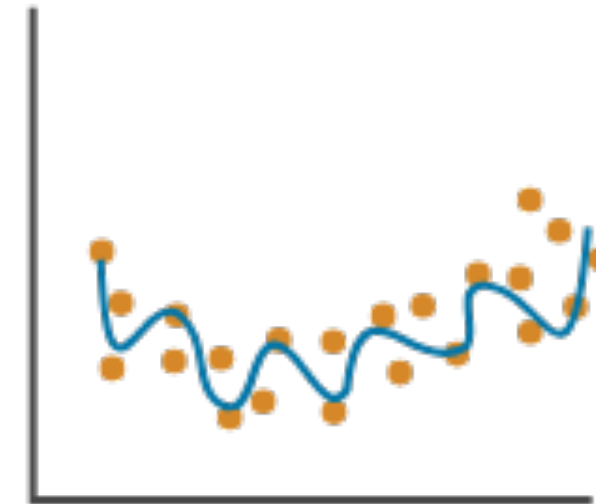
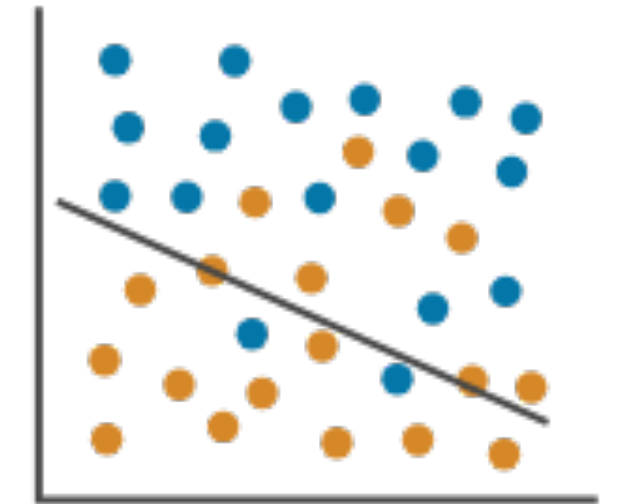
Overfitting



Right Fit



Underfitting




Neurális Háló

Regularizáció – Weight decay

$$L = \frac{1}{N} \sum_i L_i + \lambda \sum_k \sum_l W_{k,l}^2$$

L2 regularizáció
gradiense zárt alakban
kifejezhető!

$$W_{k+1} = W_k - \frac{\alpha}{N_{MB}} \sum_i (\nabla L_i - \lambda W_k)$$


Weight decay

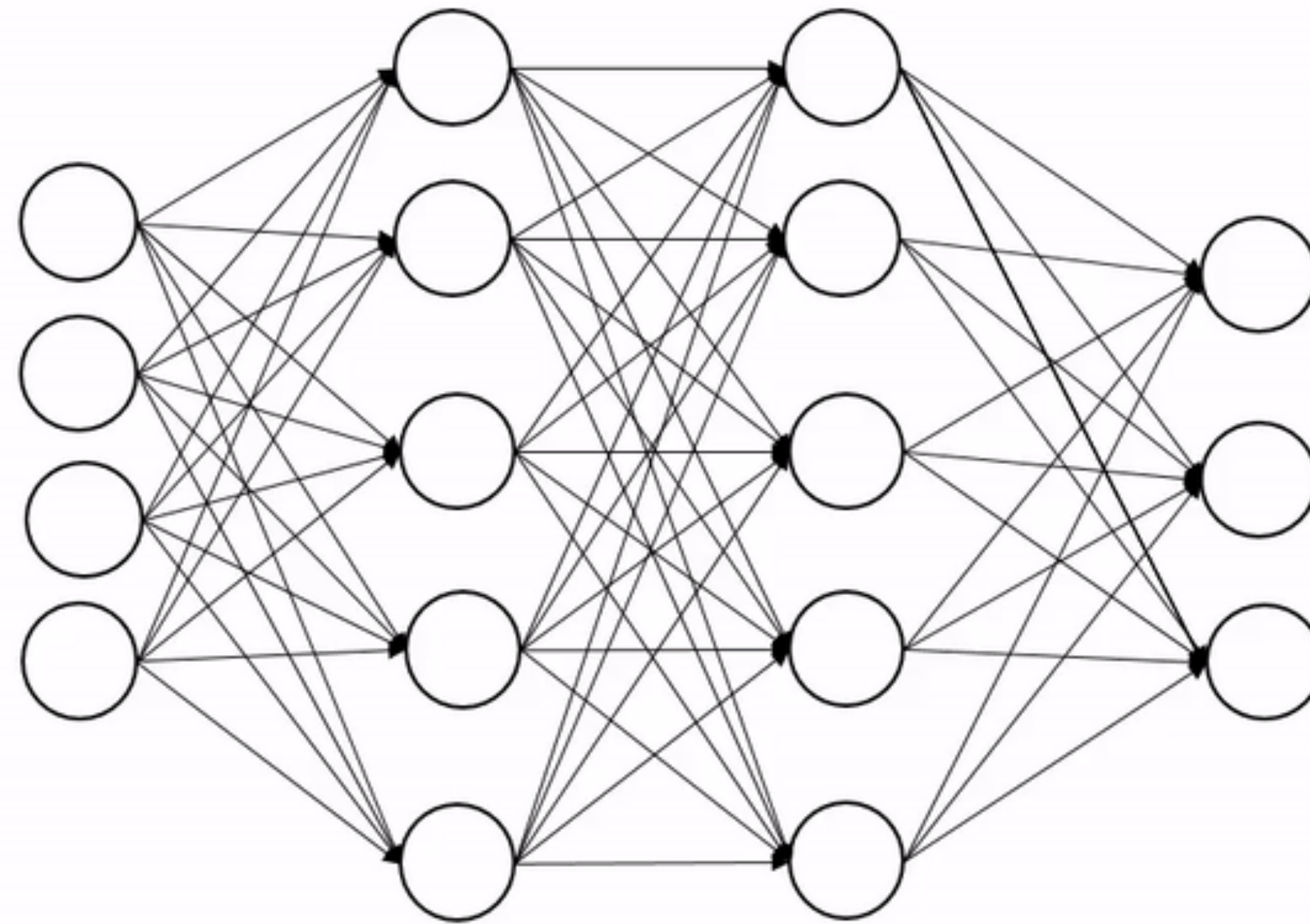
“Weight decay”

$$W_{k+1} = W_k - \eta \left(\frac{\hat{M}_k}{\sqrt{\hat{G}_k + \varepsilon}} + \lambda W_k \right)$$

AdamW

Neurális Háló

Regularizáció – Dropout

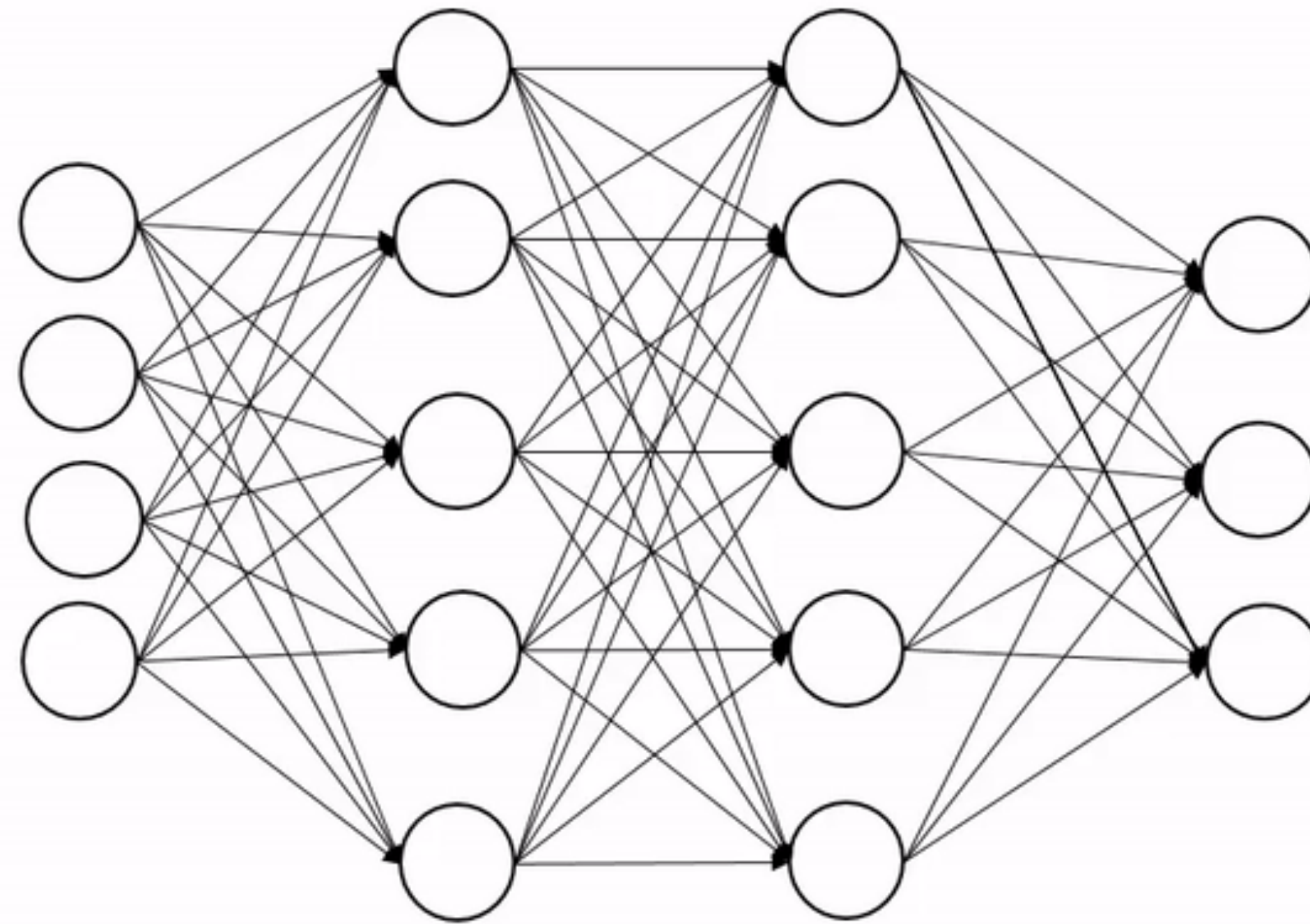


Dropout:

minden iterációban véletlenszerűen kiválasztott neuronokat optimalizálunk

Neurális Háló

Regularizáció – Dropout



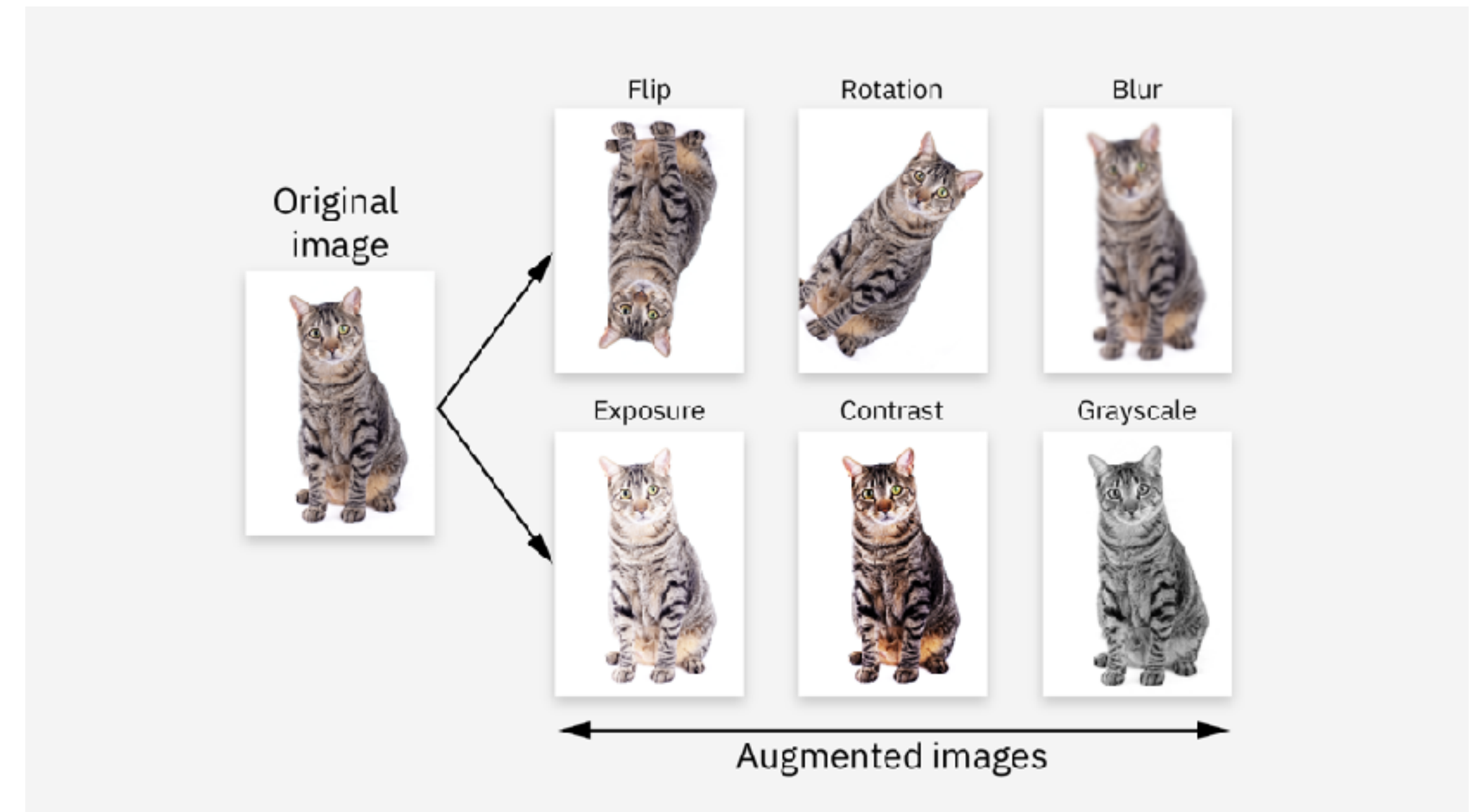
Dropout:

minden iterációban véletlenszerűen kiválasztott neuronokat optimalizálunk

Neurális Háló

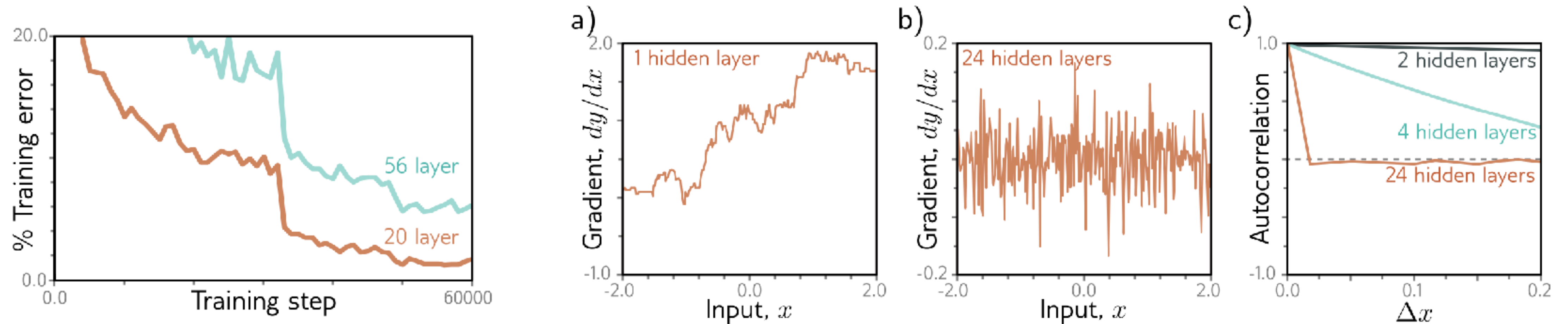
Regularizáció – Adat augmentáció

- Van, hogy a “nyers” adatok nem elégségesek az általánosításhoz...
- **Adat augmentáció** – az adathalmazt mesterségesen kibővítjük
 - Pl. képekre különböző transzformációkat alkalmazunk
- Alkalmazás függő, hogy milyen módon érdemes csinálni
- (Az önfelügyelt tanítás prototípusa)



Neurális Háló

Reziduális kapcsolatok – Motiváció

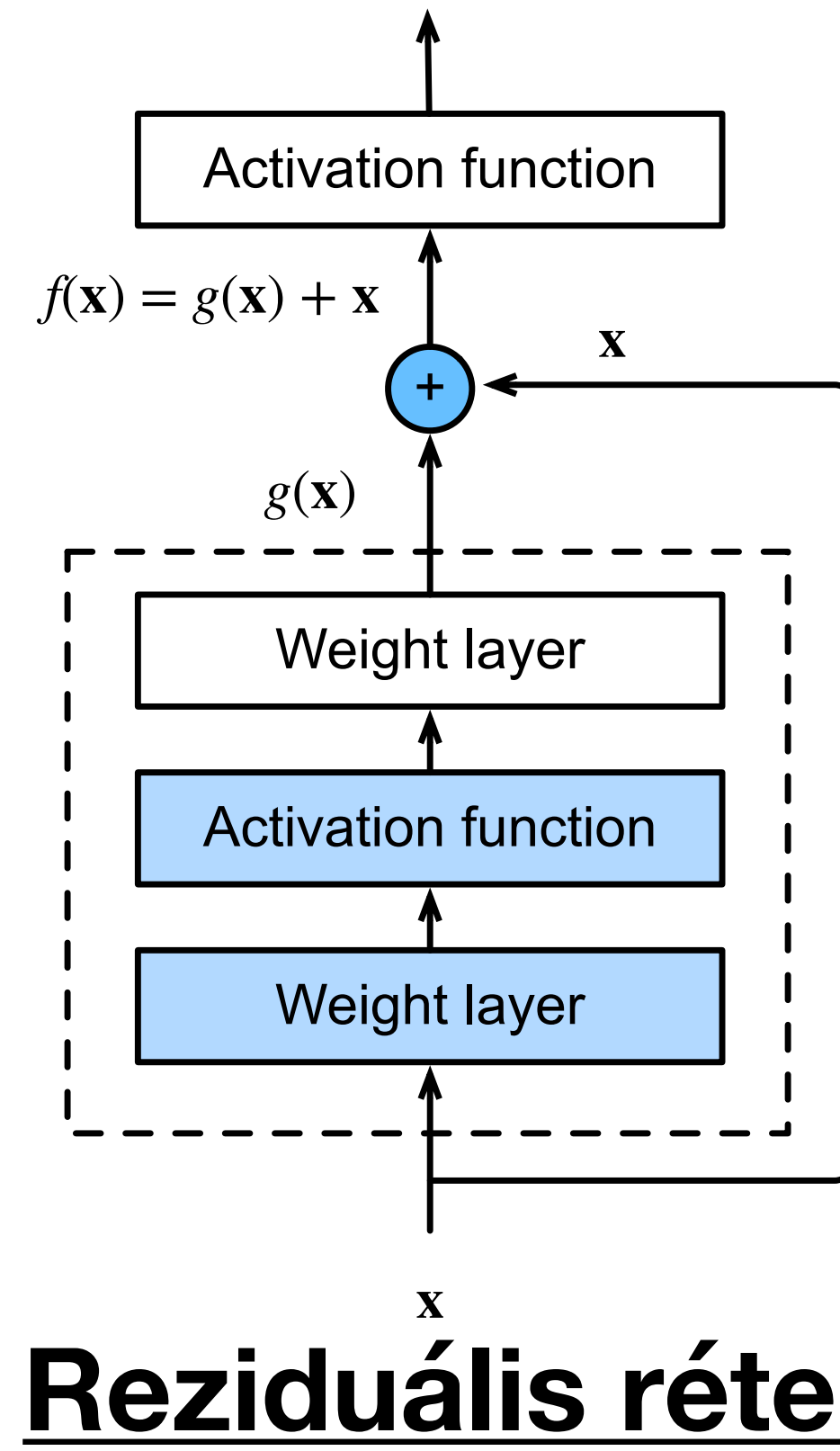
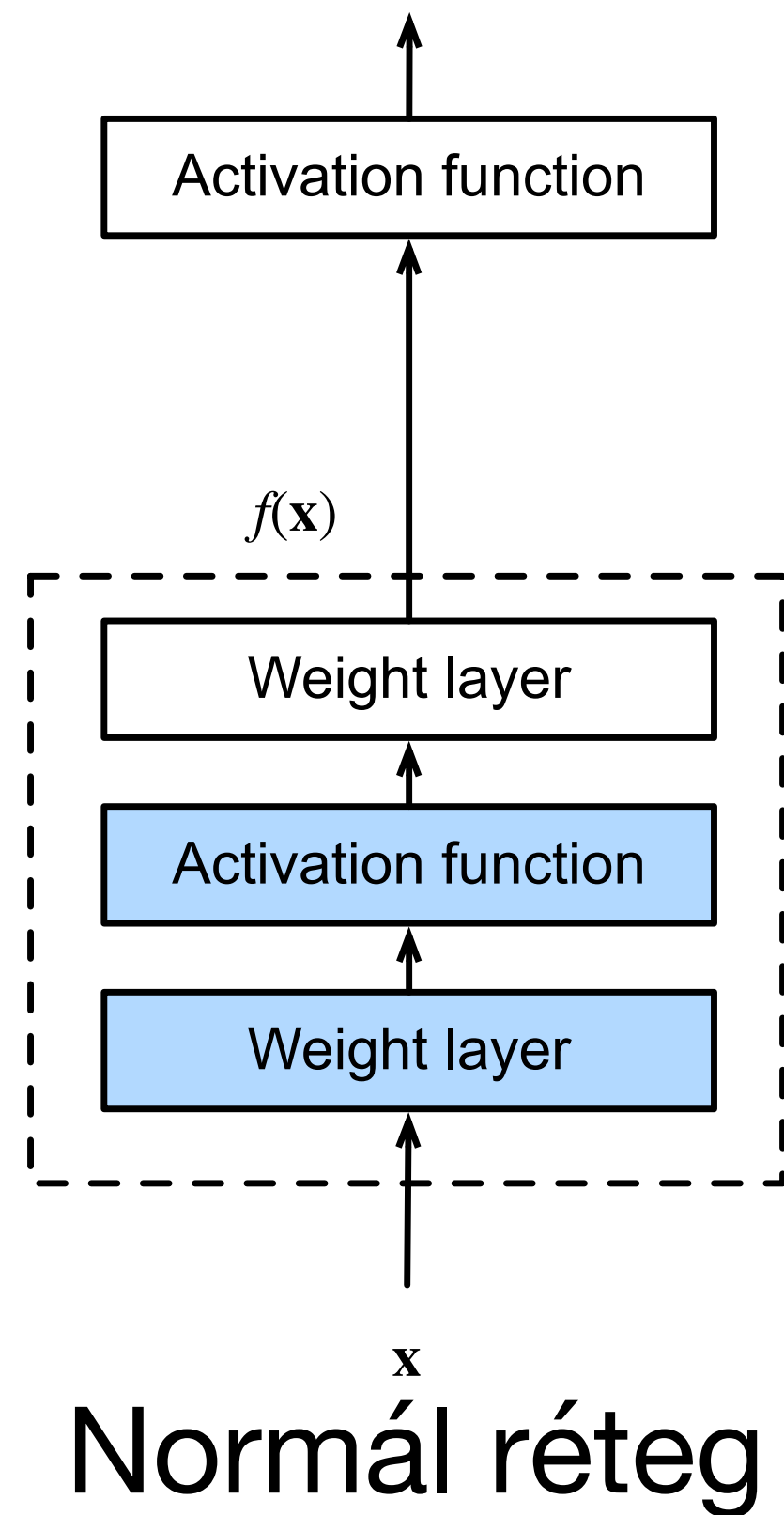
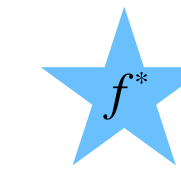


Az igazán mély (> 20 rétegű) neurális háló tanítása nem látszik reálisnak...
Megoldás: reziduális rétegek

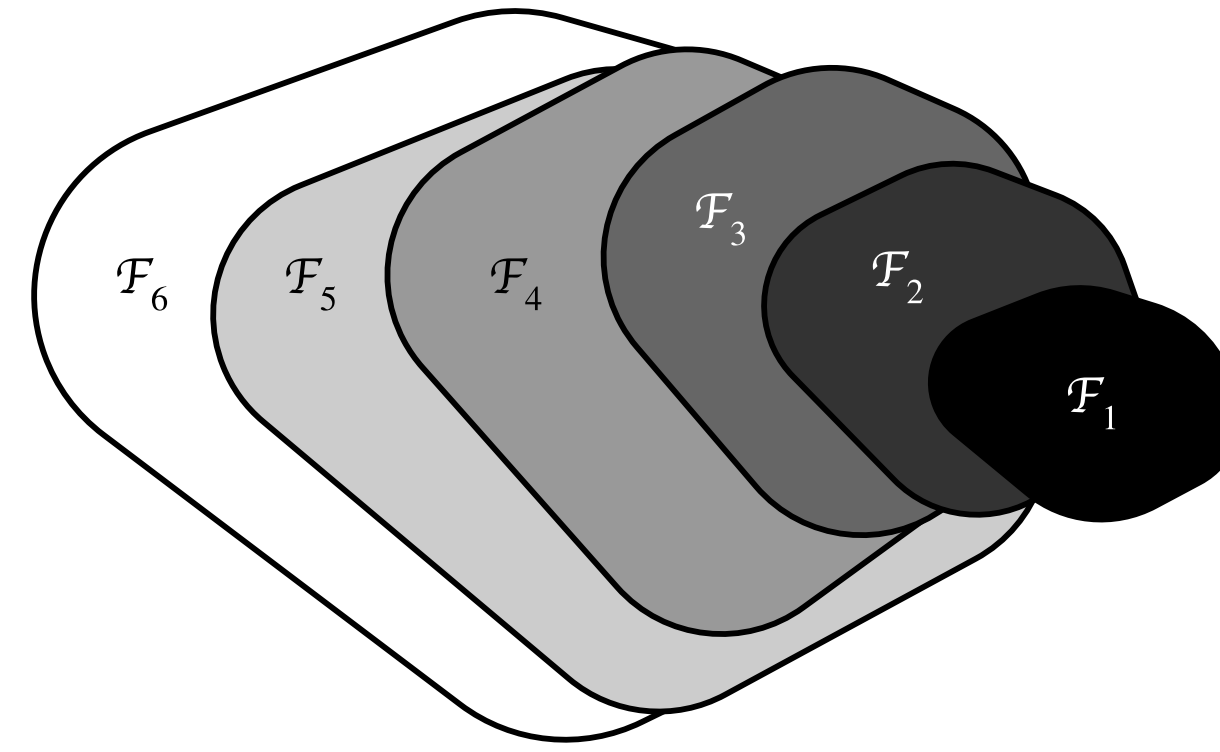
Neurális Háló

Reziduális kapcsolatok

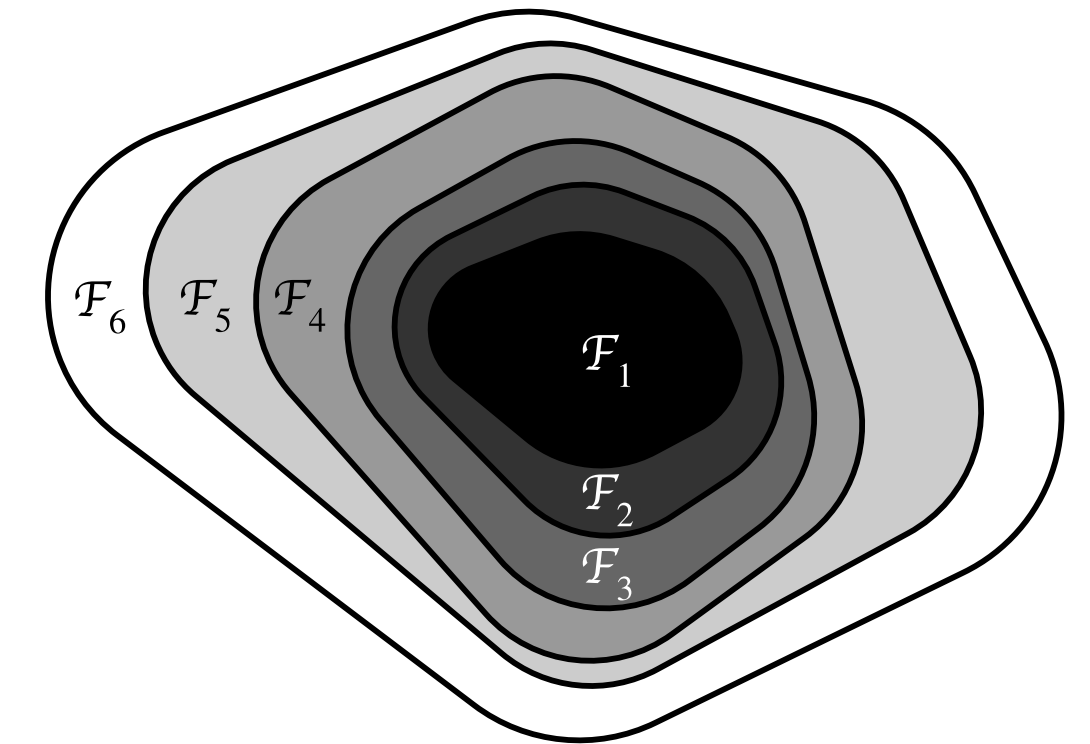
Forrás: <https://d2l.ai/>



skip connection



Nem egymásba ágyazott függvények



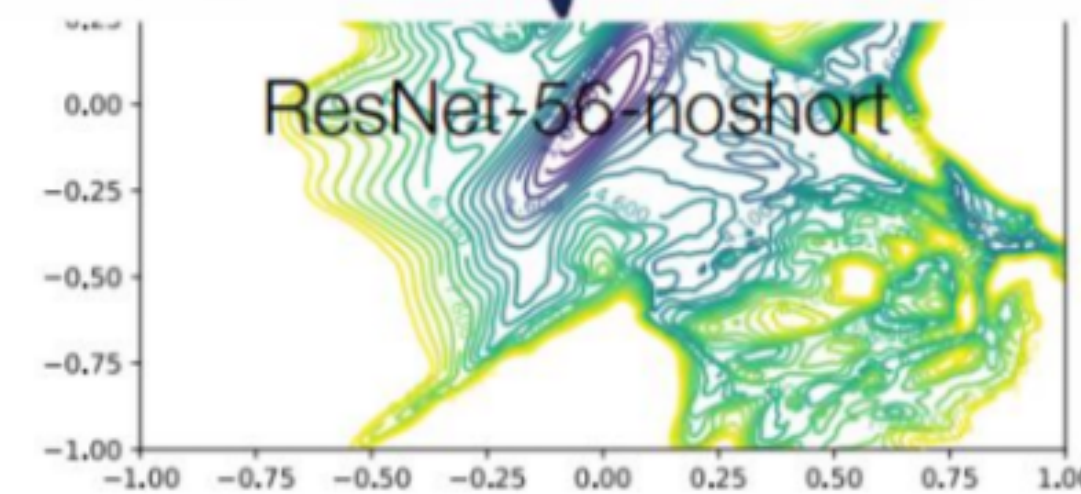
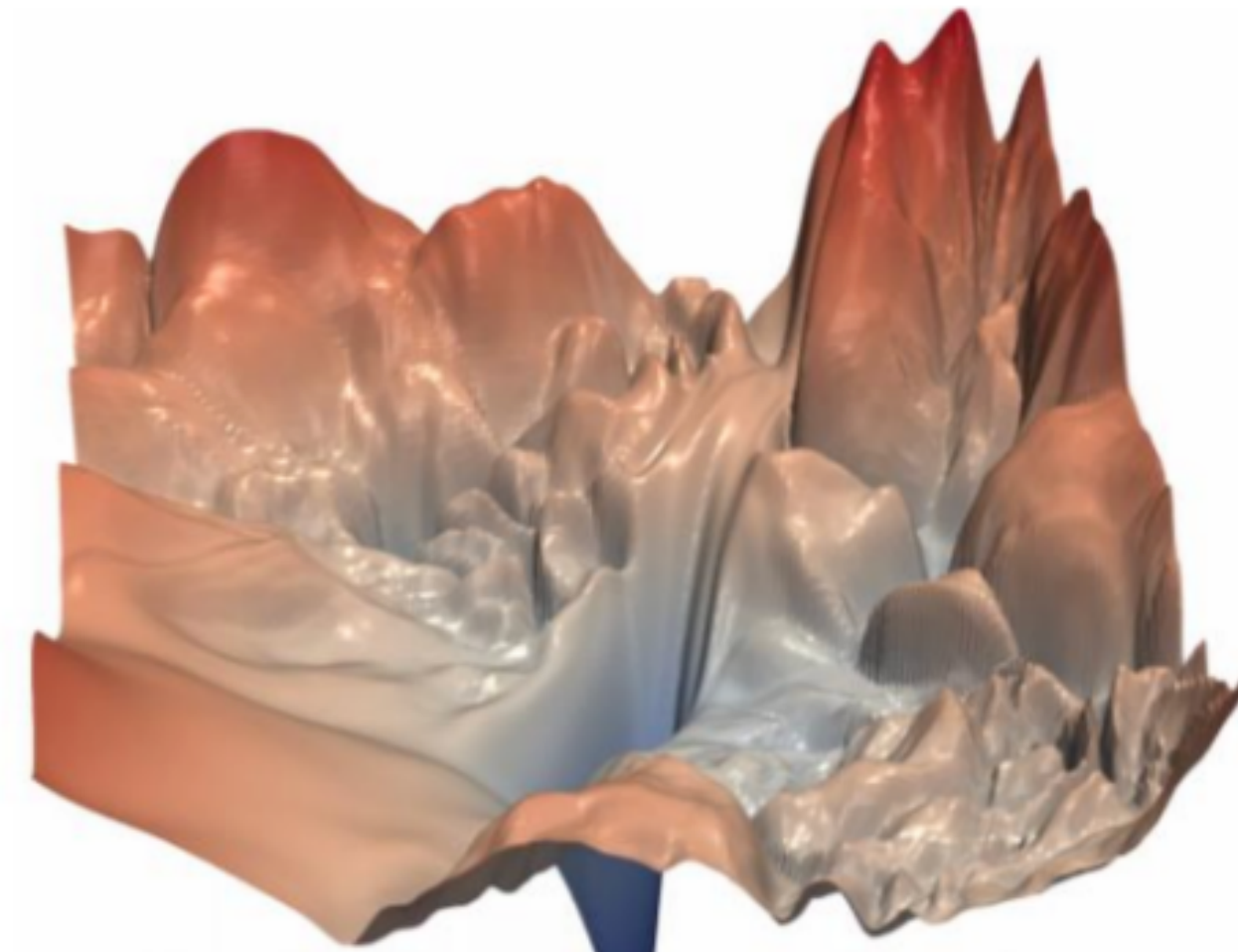
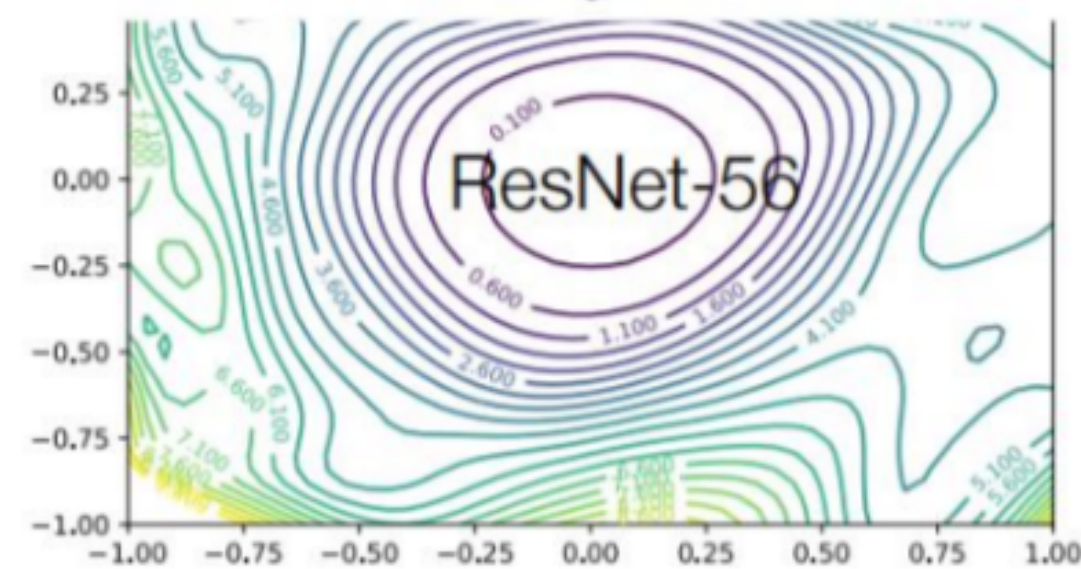
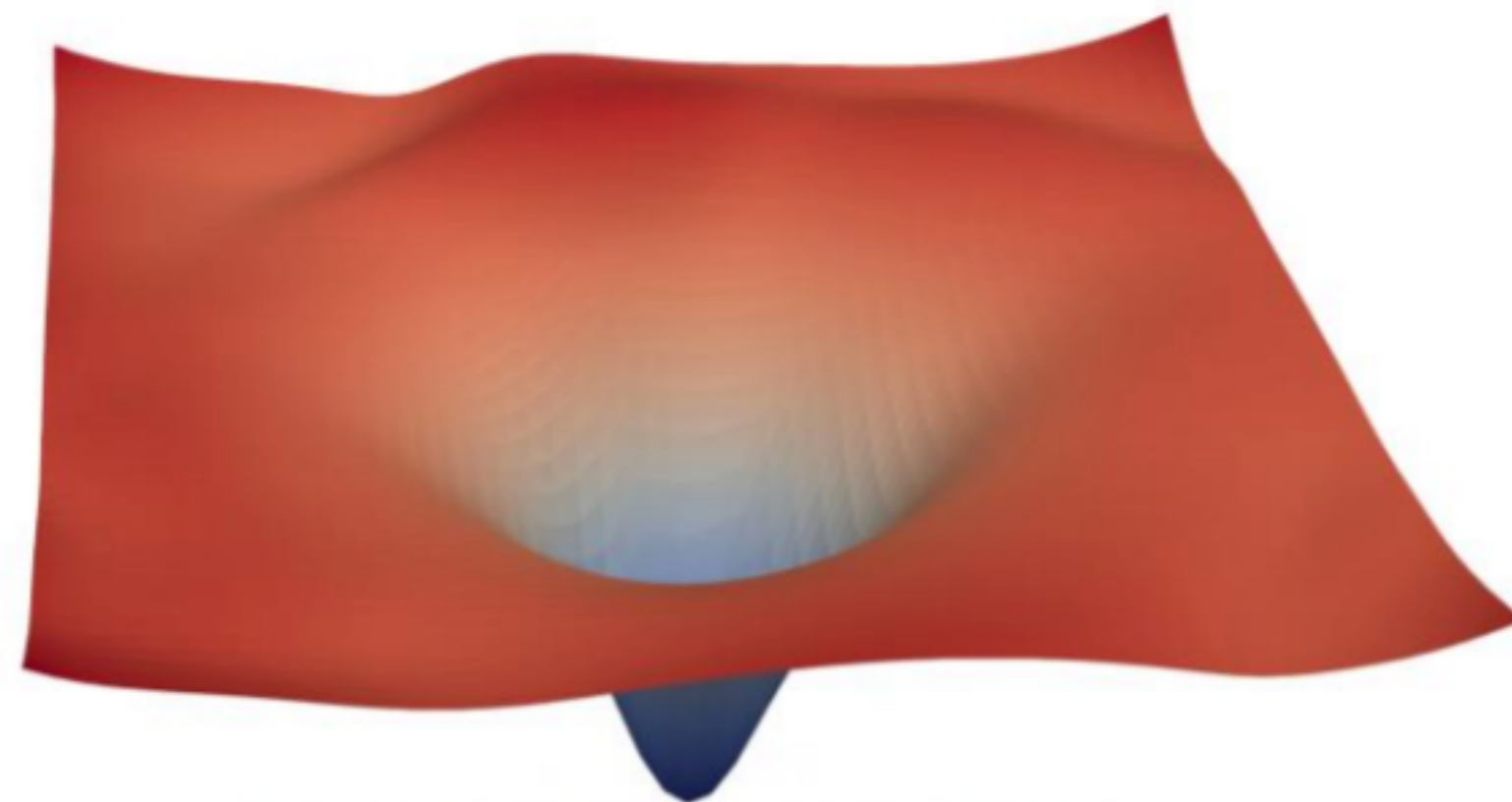
Egymásba ágyazott függvények

Identitás függvényre inicializálható!

Akár több 100 rétegű háló is taníthatóak!

Neurális Hálók

Reziduális kapcsolatok



A reziduális hálók veszteségfüggvénye *sokkal* simább — könnyebb optimalizálni!

Neurális Háló Normalizáció

Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift

Sergey Ioffe
Google Inc., sioffe@google.com

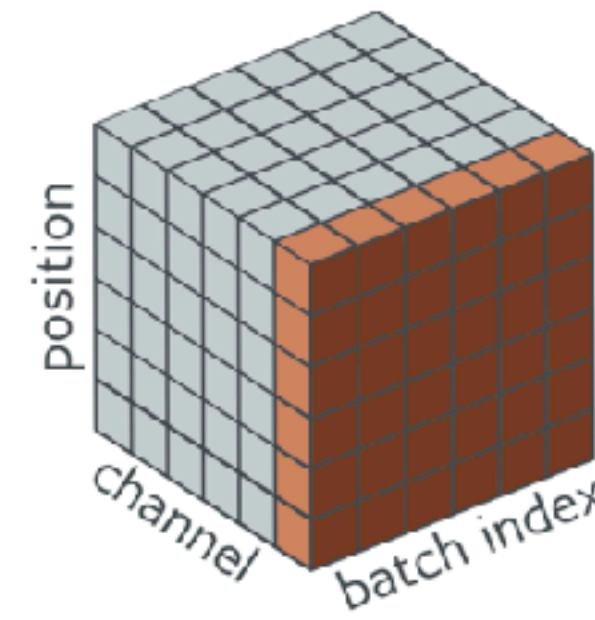
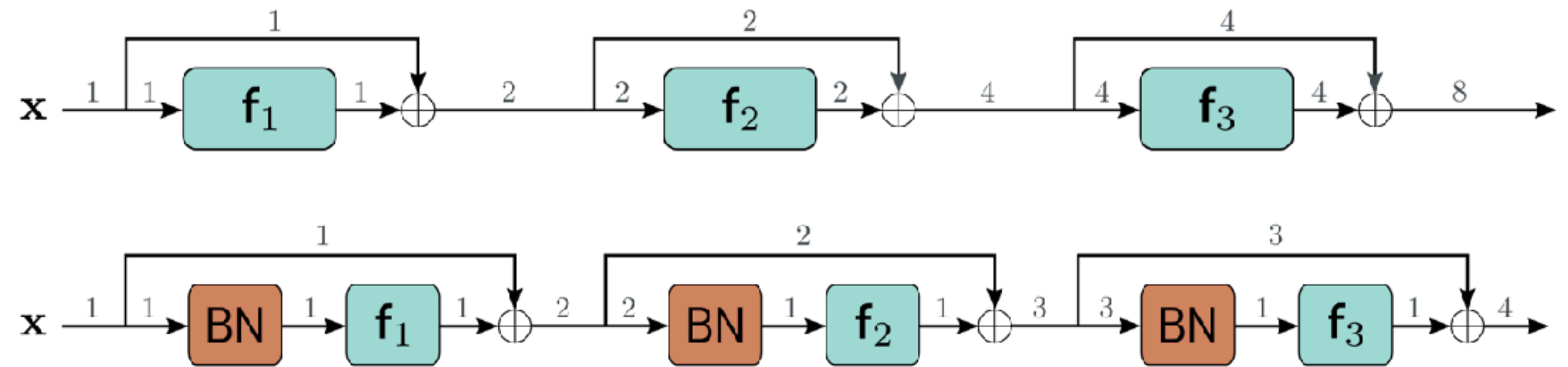
Christian Szegedy
Google Inc., szegedy@google.com

$$m_h = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} h_i$$

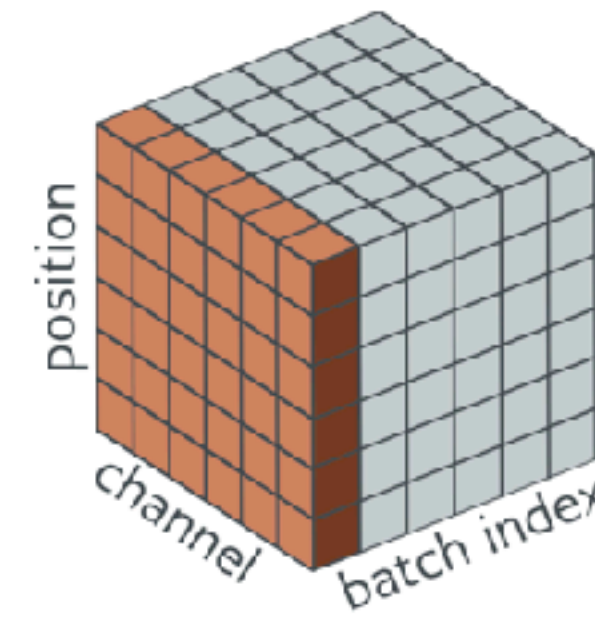
$$s_h = \sqrt{\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} (h_i - m_h)^2}$$

$$h_i \leftarrow \frac{h_i - m_h}{s_h + \epsilon}$$

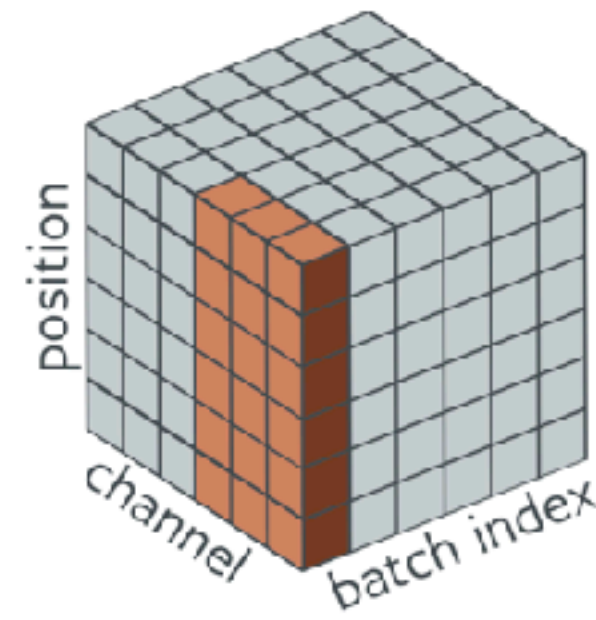
$$h_i \leftarrow \gamma h_i + \delta$$



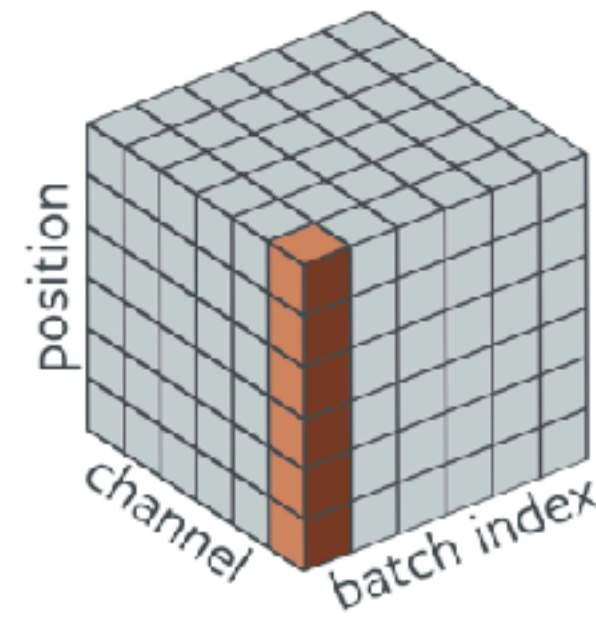
BatchNorm



LayerNorm



GroupNorm

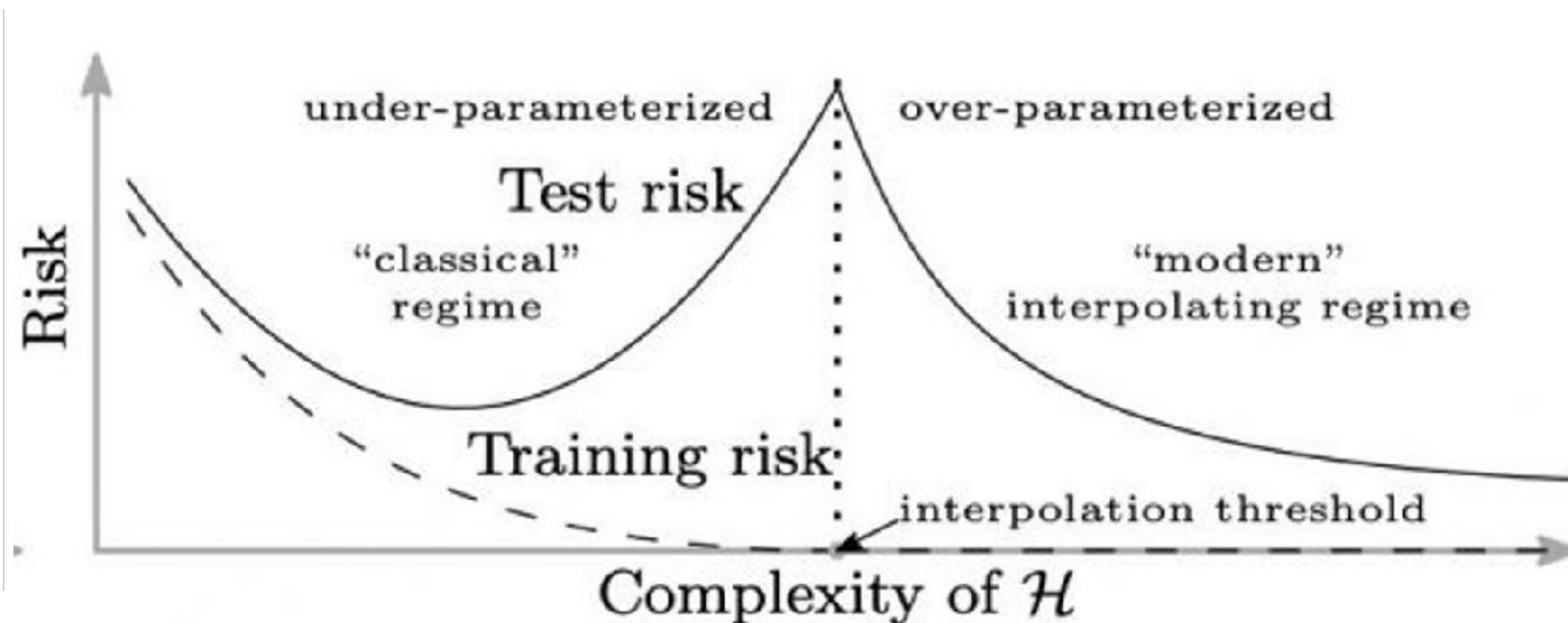
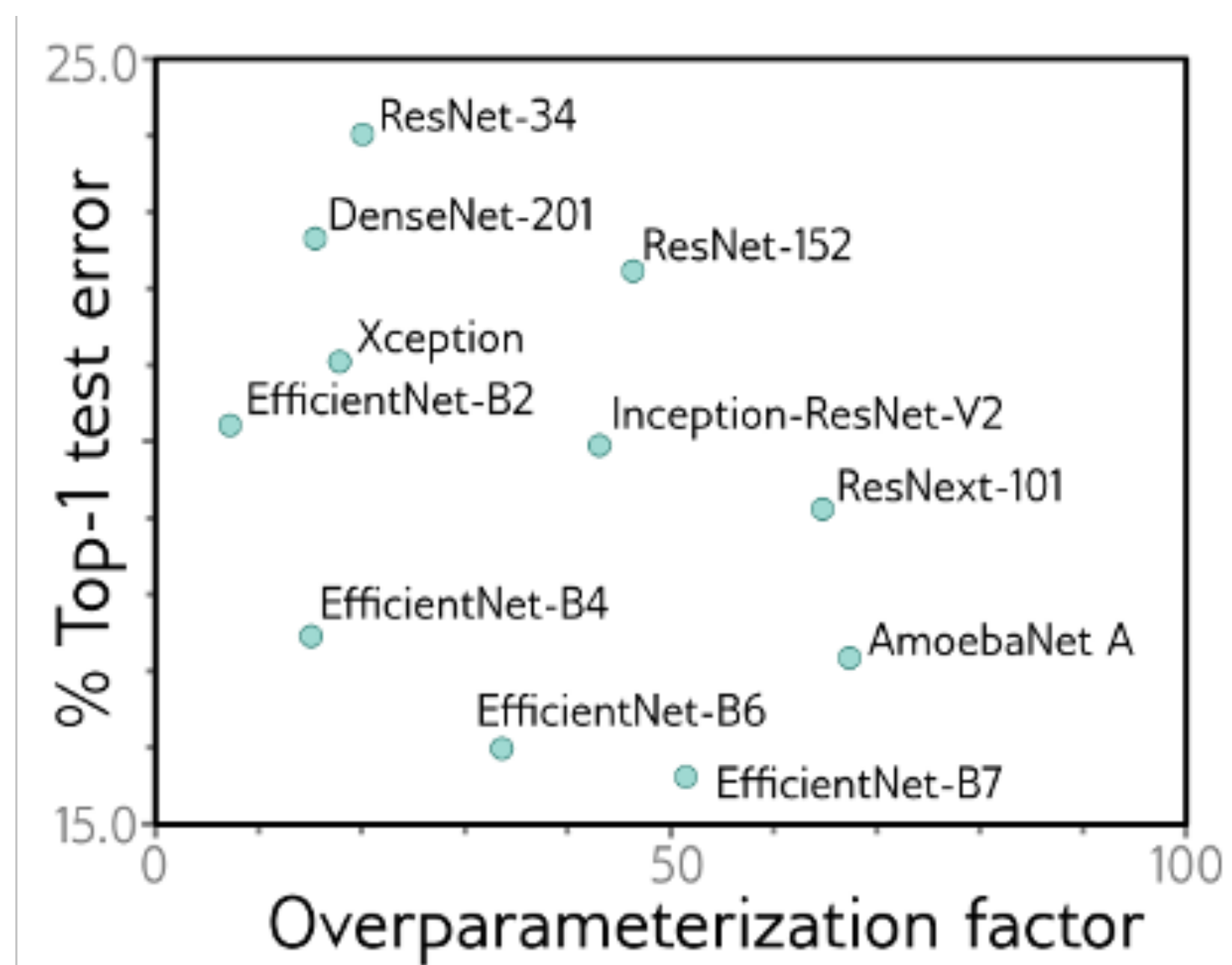


InstanceNorm

Batch Normalizáció (BatchNorm)

Neurális Háló

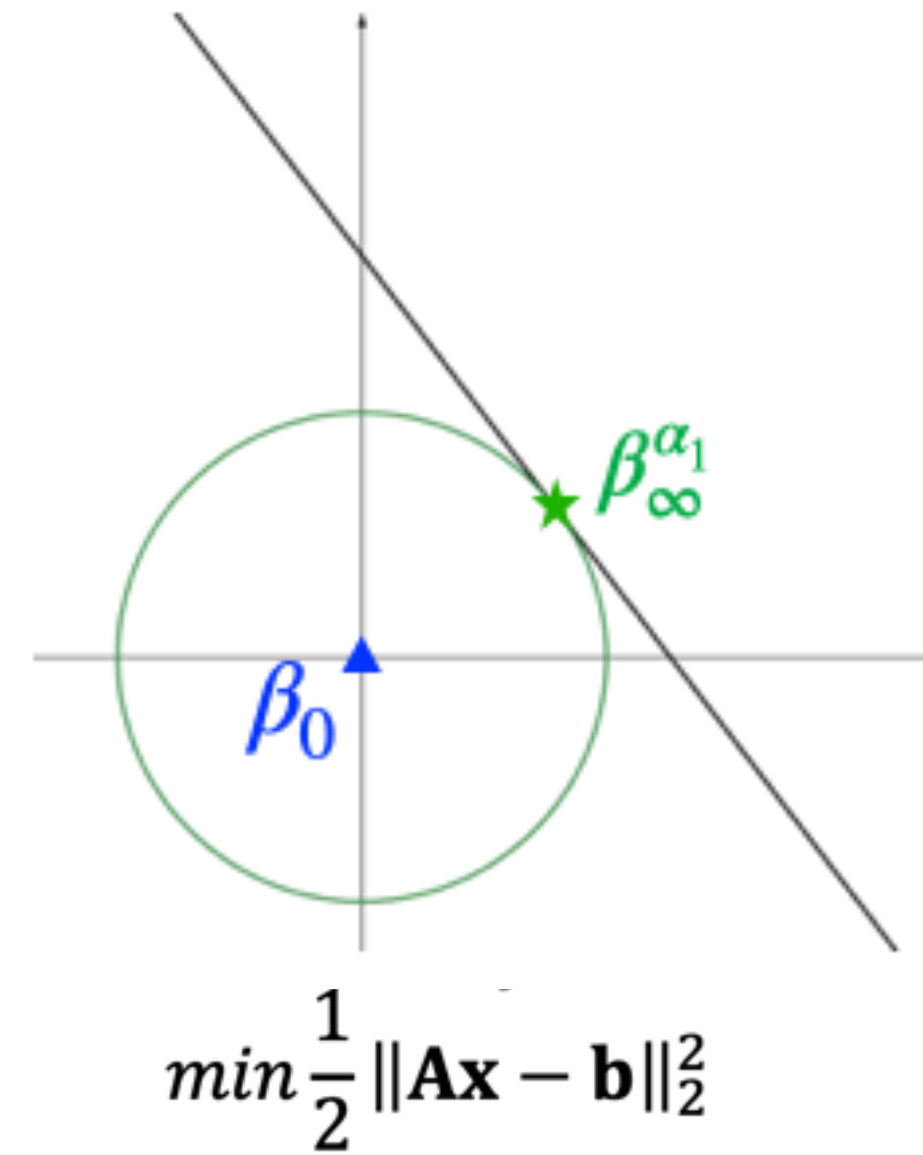
Implicit Regularizáció



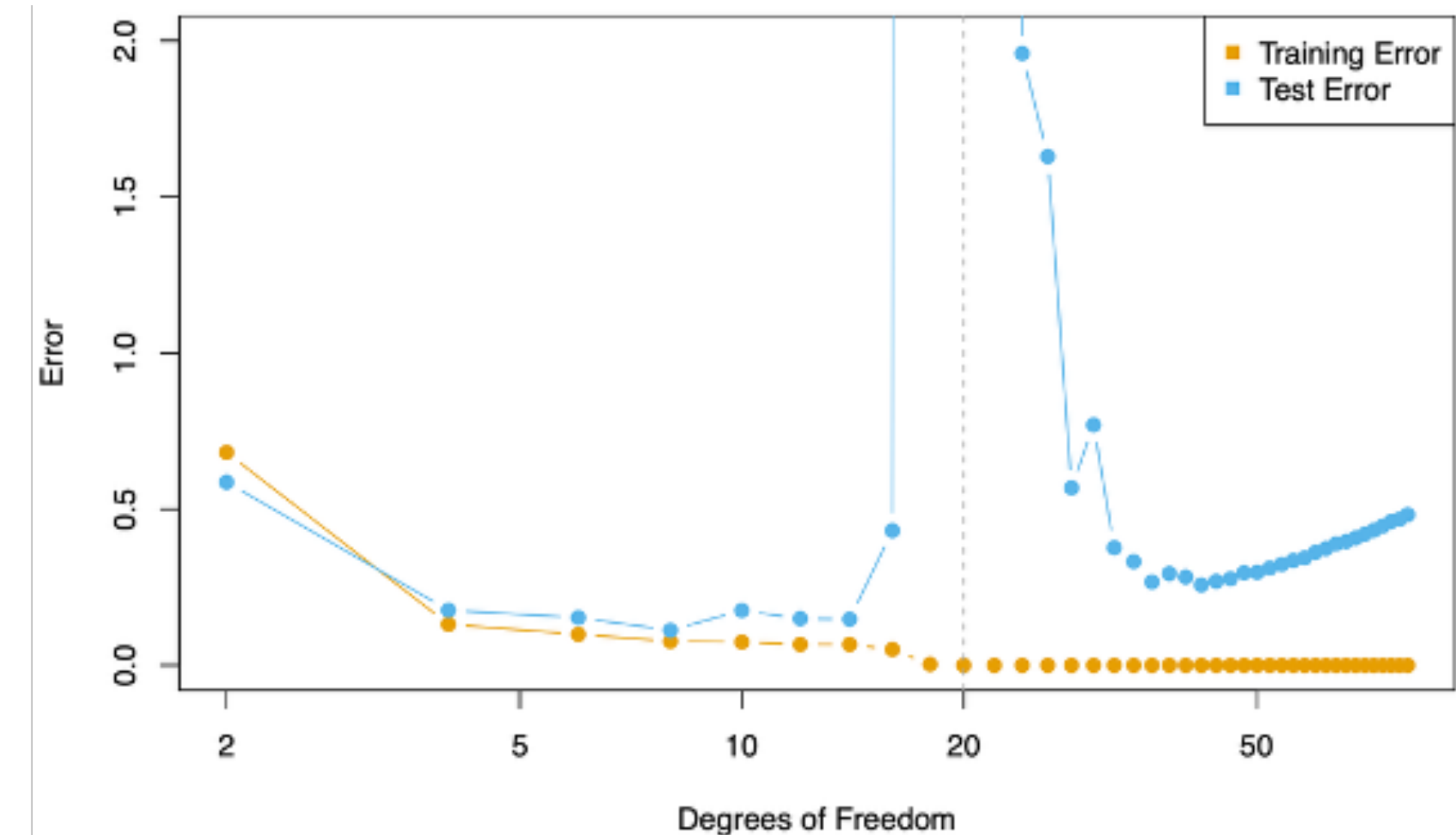
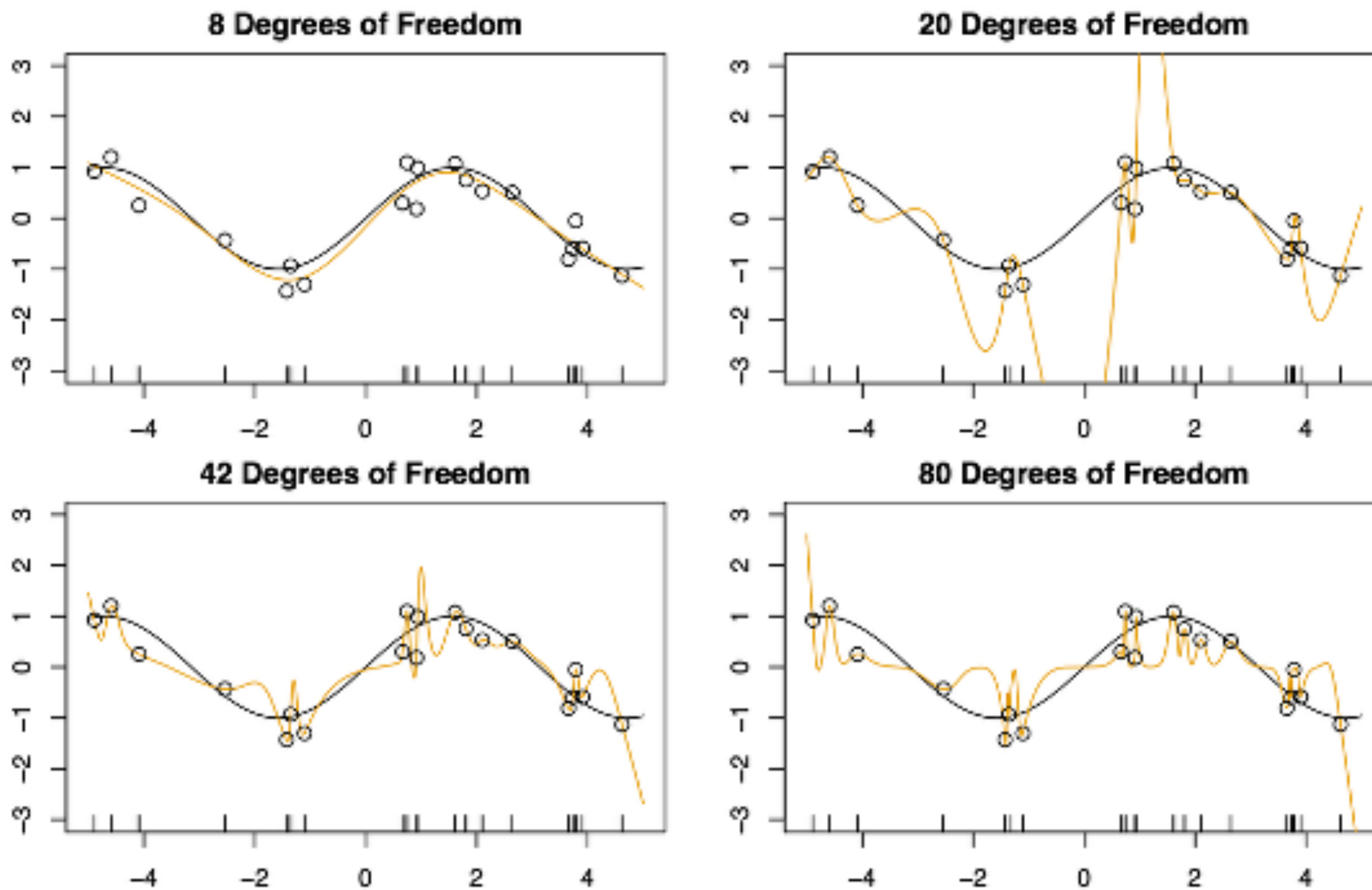
A modern neurális háló erősen túlparaméterezettek, mégis jól általánosítanak...
“Double descent”: minél több paraméter, annál jobb általánosítási képesség!

Neurális Hálók

Implicit Regularizáció



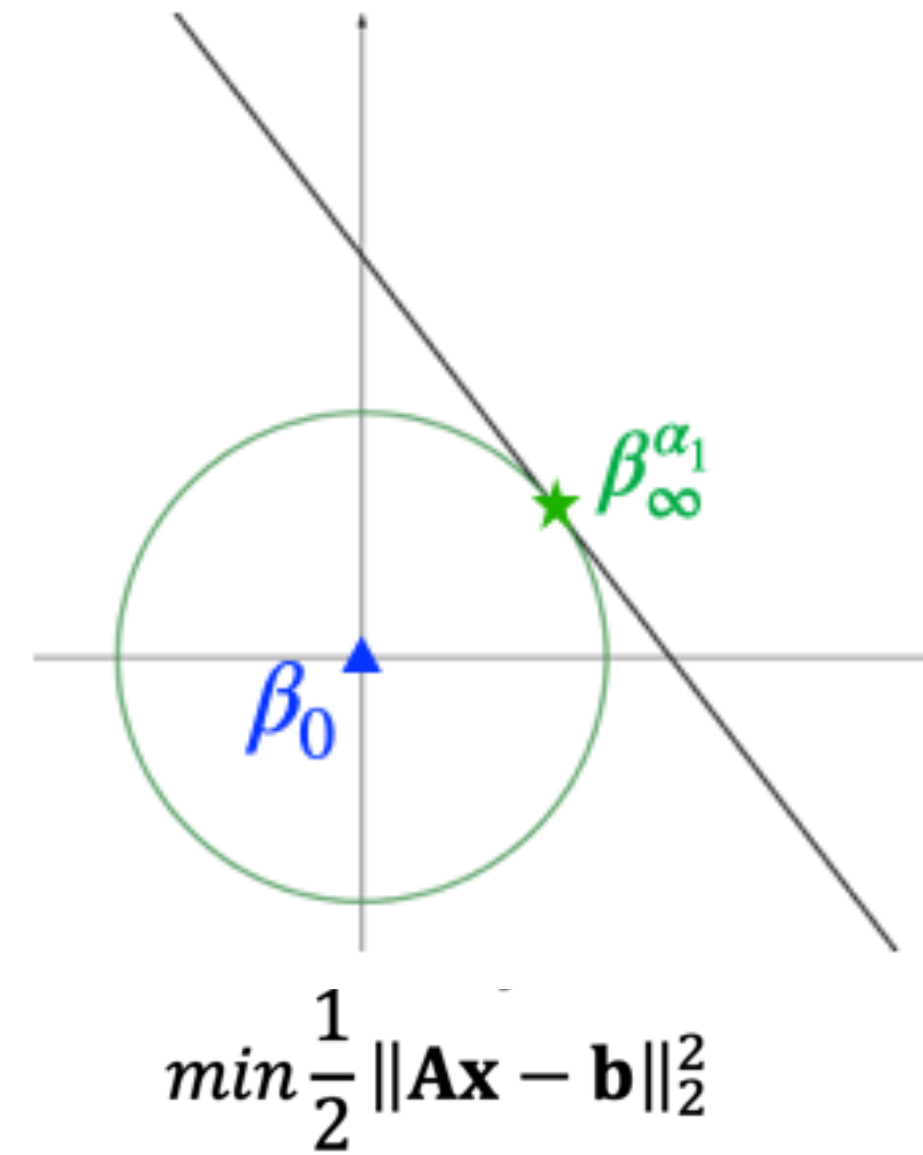
Gradient descent a legkisebb L2 normájú megoldáshoz konvergál!



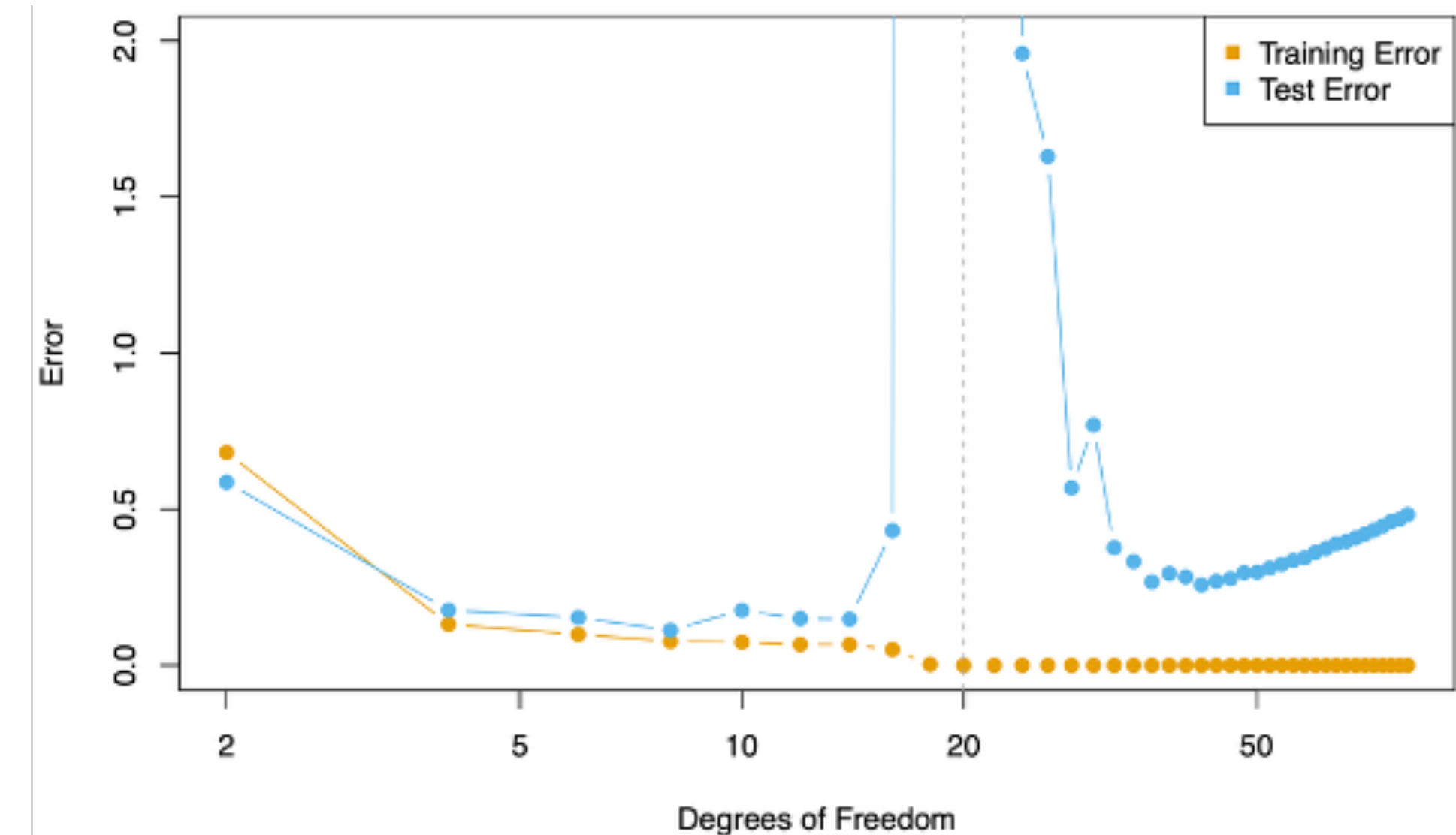
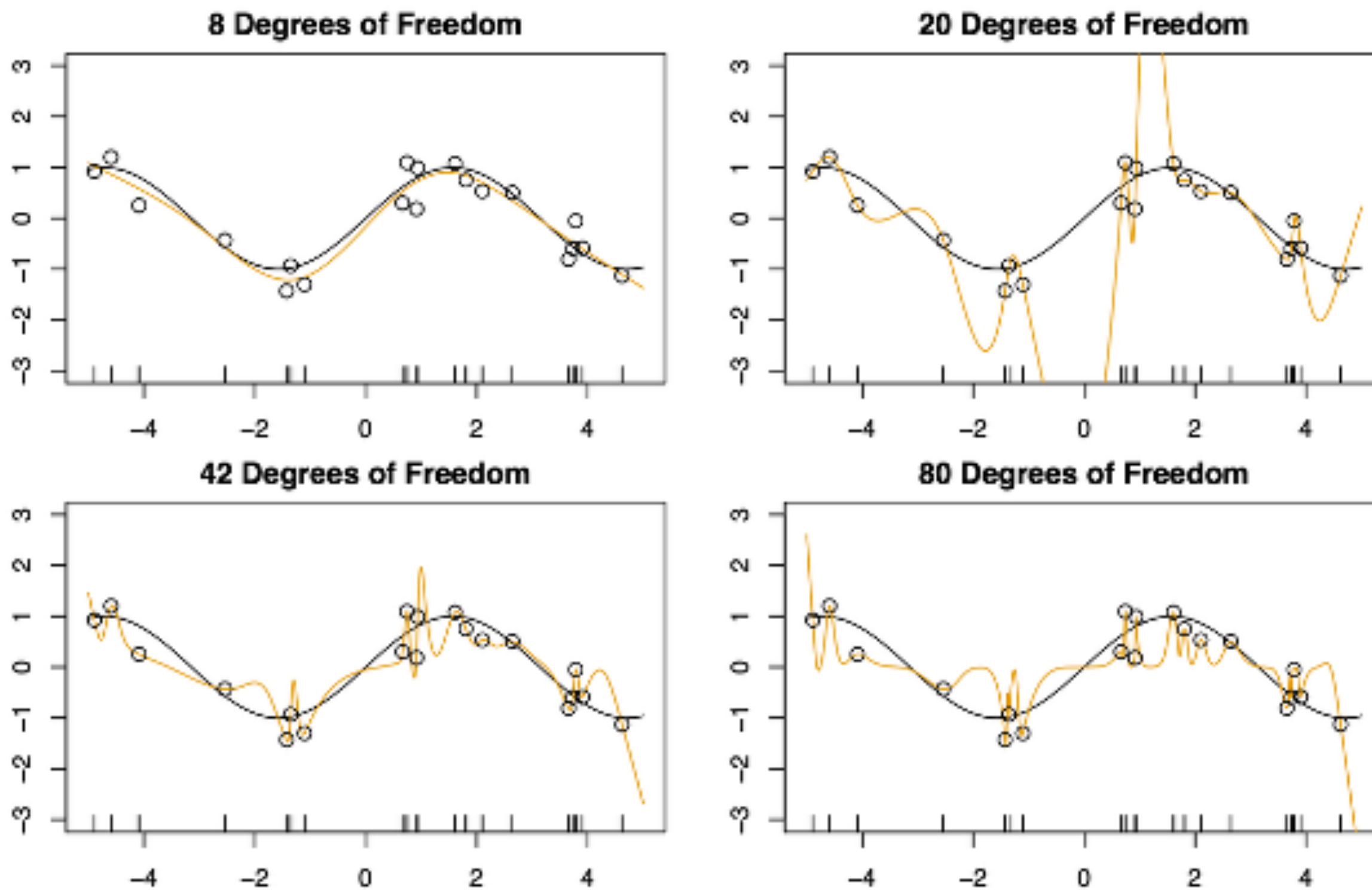
Double descent — még lineáris regresszióra is megfigyelhető!

Neurális Hálók

Implicit Regularizáció



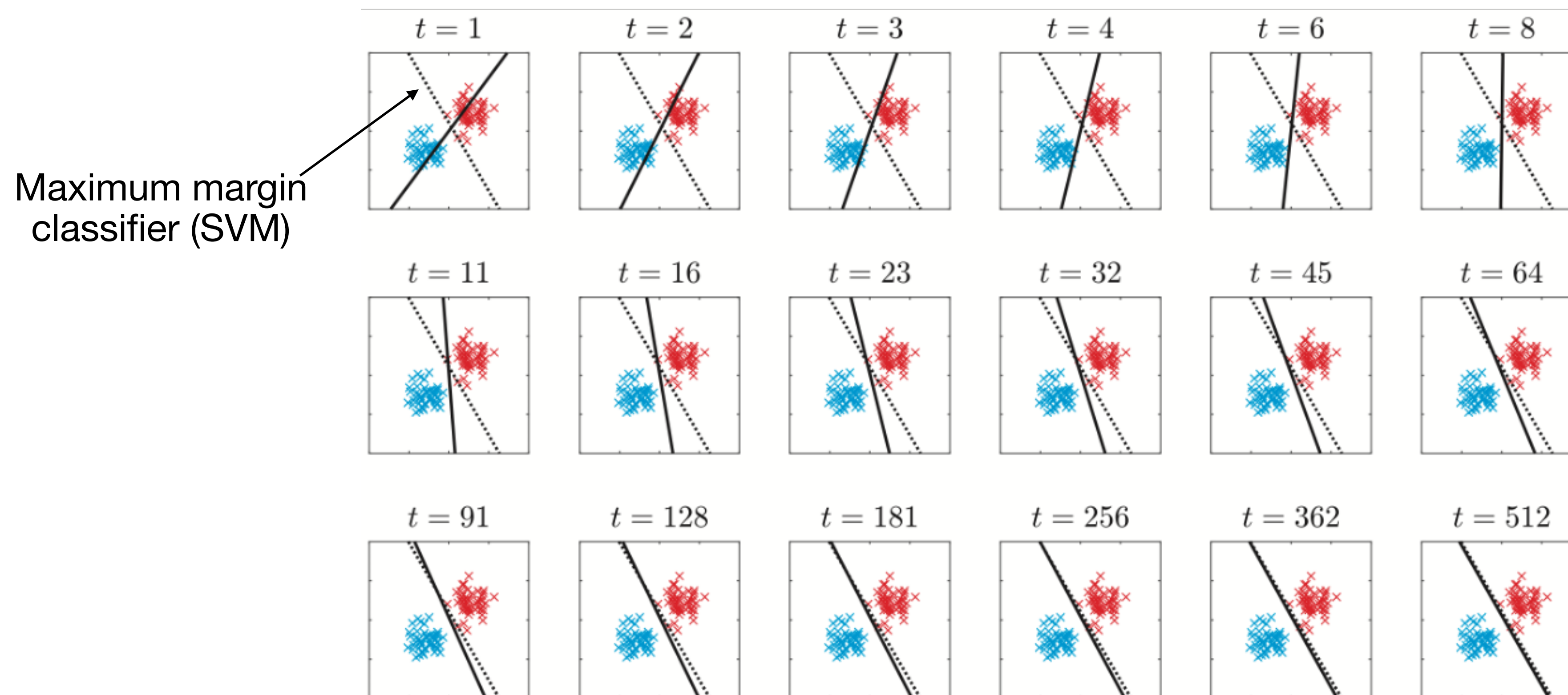
Gradient descent a legkisebb L2 normájú megoldáshoz konvergál!



Double descent — még lineáris regresszióra is megfigyelhető!

Neurális Háló

Implicit Regularizáció



Double descent — még lineáris klasszifikációra is megfigyelhető!

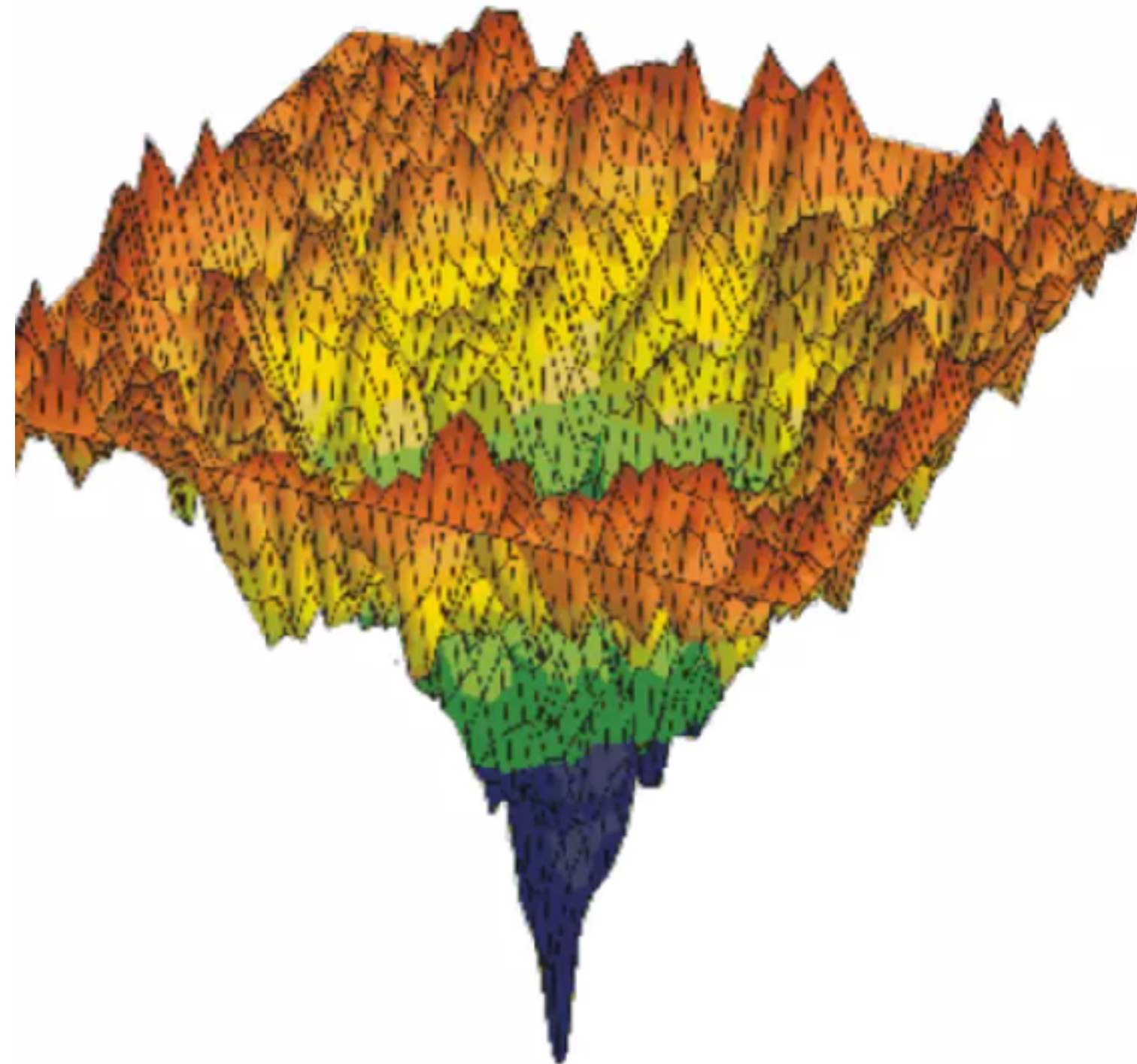
Neurális Hálók

Implicit Regularizáció

- **Implicit regularizáció**: a gradient descent “egyszerű” megoldáshoz konvergál, még túlpaméterezett modell esetén is!
- Implicit regularizáció forrásai:
 - Architektúra (skip connection, normalizáció, stb.)
 - Batch-méret
 - Sztochasztikus optimalizáció
 - Inicializáció
 - Korai leállítás (early stopping)

Neurális Háló

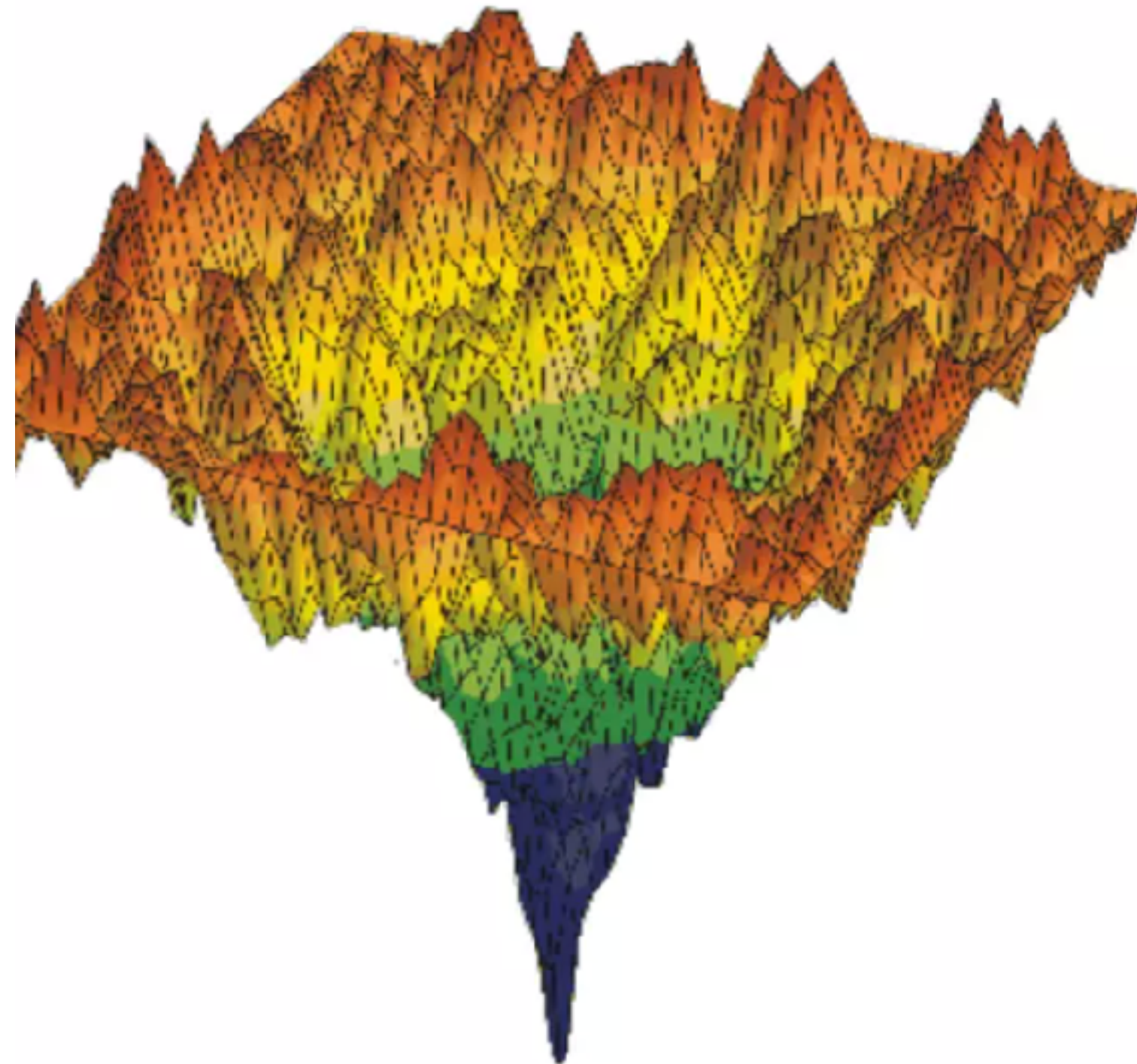
Lokális minimumok?



Neurális háló: nemlineáris, nem-konvex függvények
A gradient descent miért nem ragad lokális optimumba?

Neurális Háló

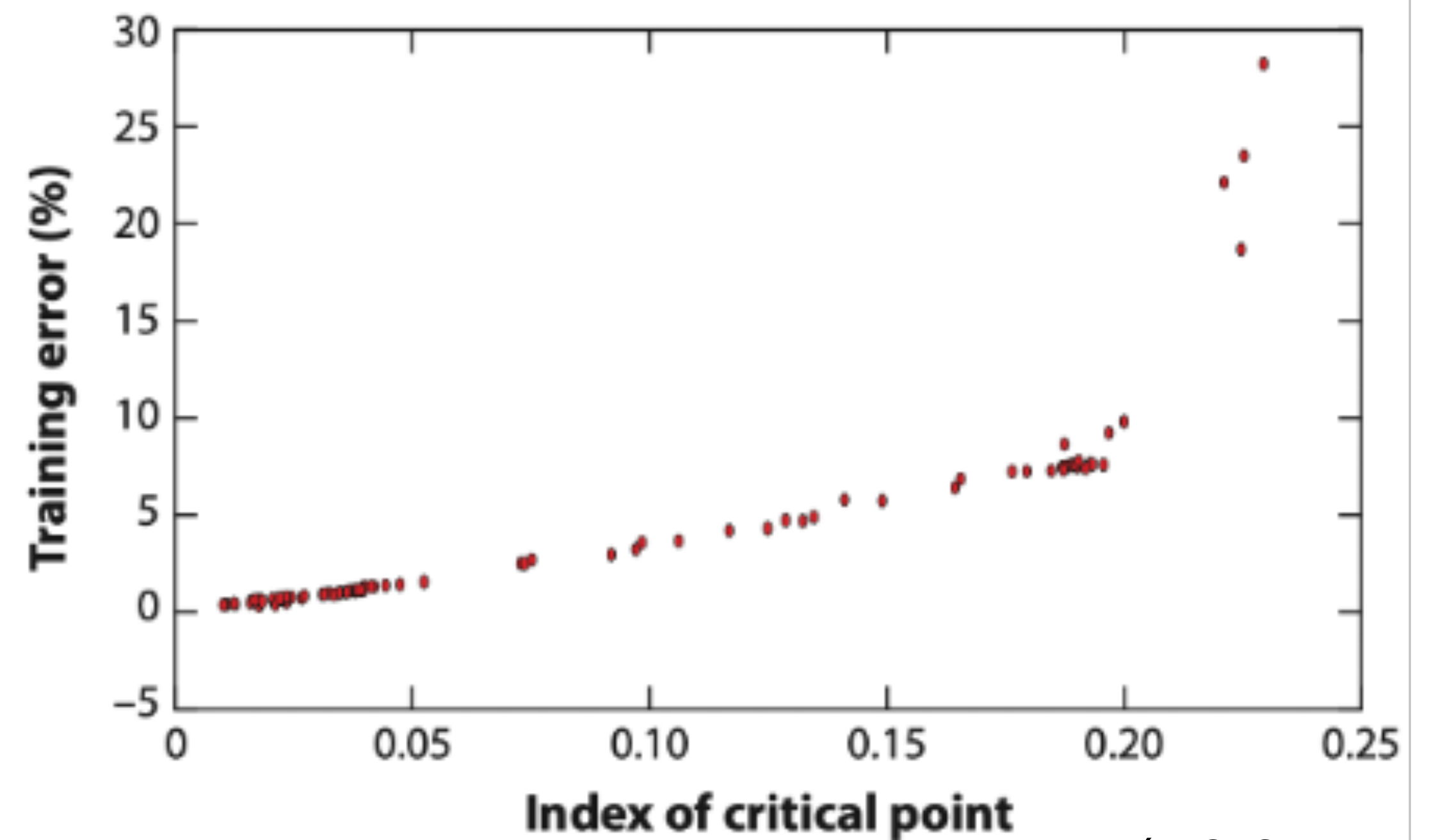
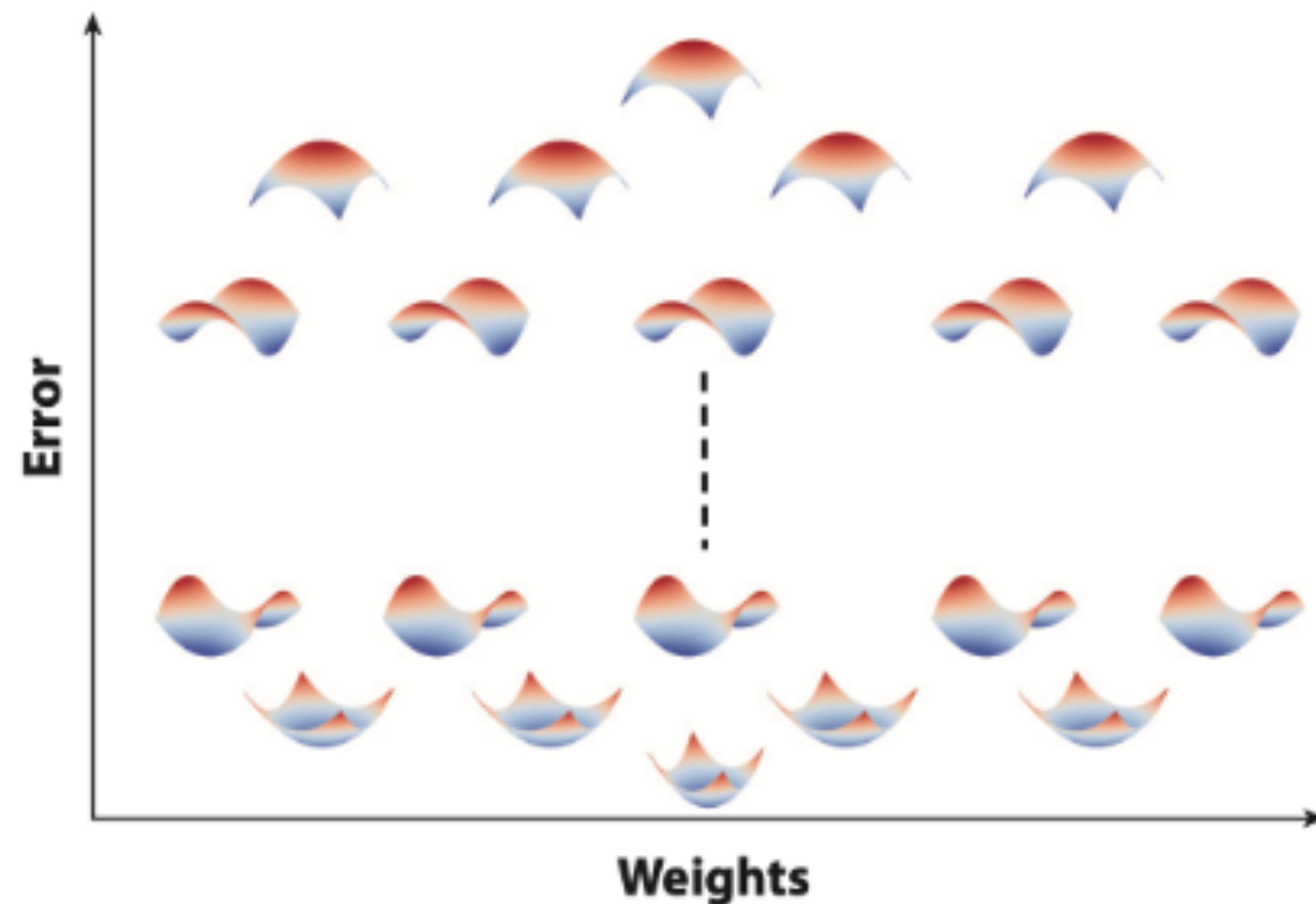
Lokális minimumok?



Neurális háló: nemlineáris, nem-konvex függvények
A gradient descent miért nem ragad lokális optimumba?

Neurális Háló

Lokális minimumok?



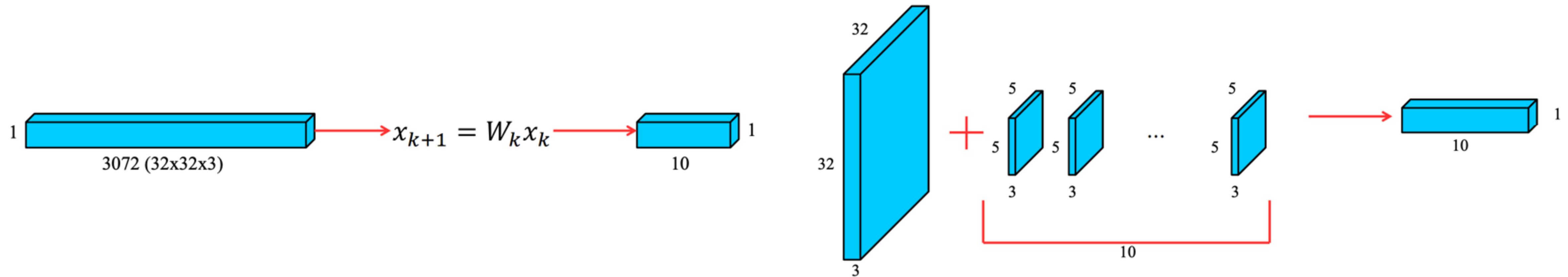
Forrás: S. Ganguli

A tipikus kritikus pont nyeregpont — a lokális minimumok nagyon ritkák!

Konvolúciós Neurális Háló

Motiváció

Forrás: Szemenyei M.



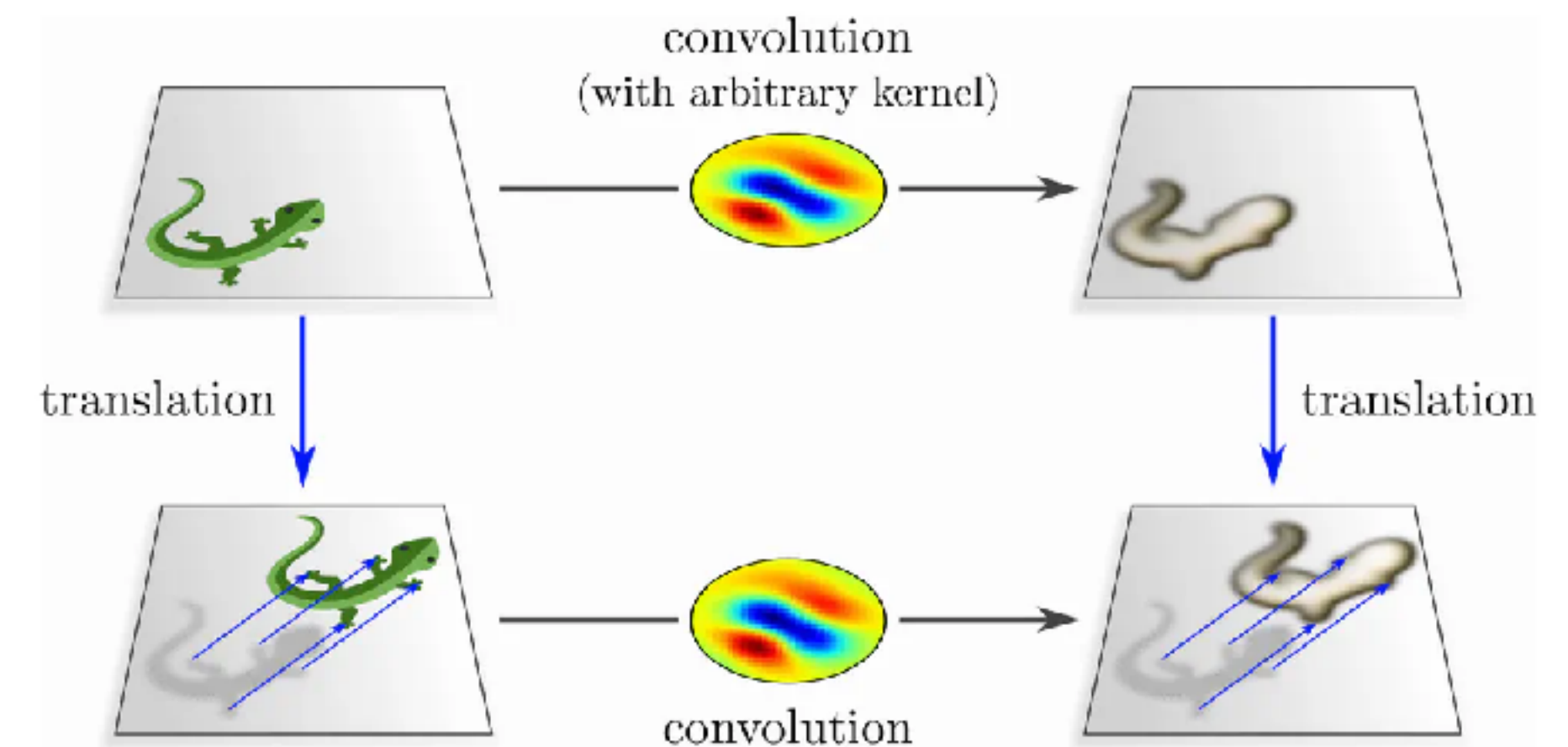
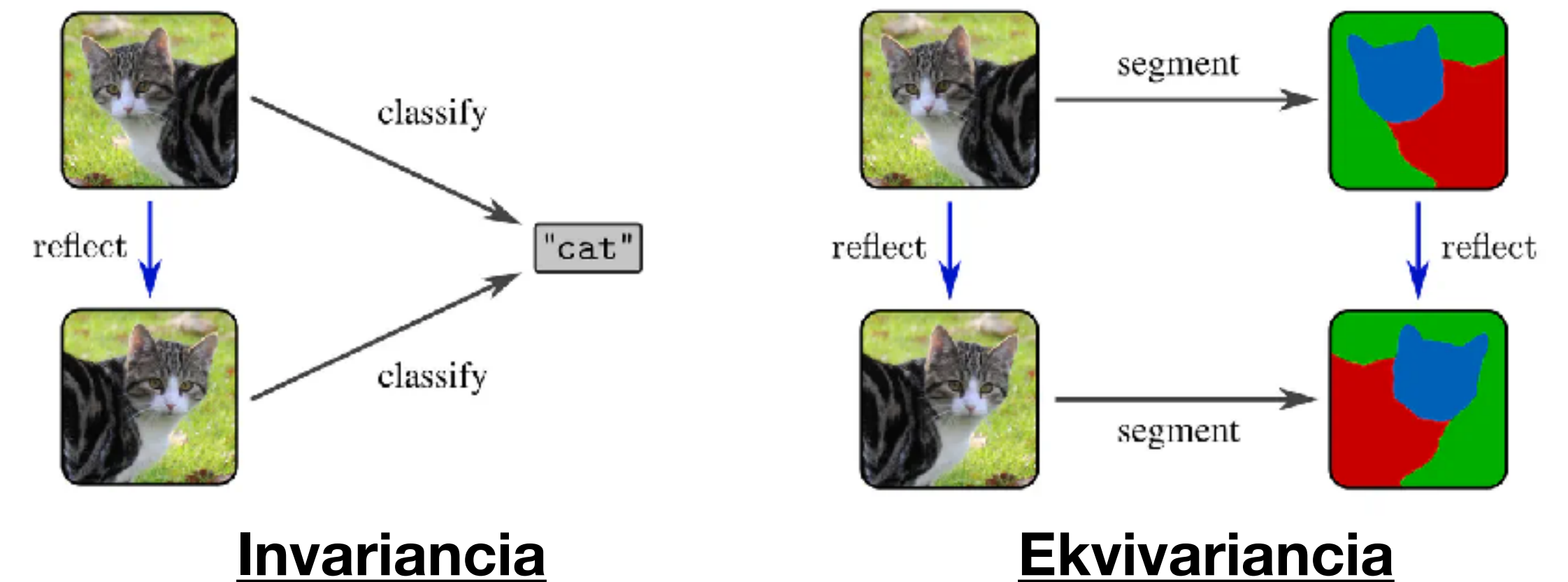
MLP (“fully connected”) réteg:
10 kimenetre: kb. 30000 paraméter
(512 kimenetre: kb. 1,5 Millió paraméter!)

Konvolúciós (CNN) réteg:
10 kimenetre: $10 \times 3 \times 5 \times 5 = 750$ paraméter
(512 kimenetre: kb. 38000 paraméter!)

Konvolúciós Neurális Háló

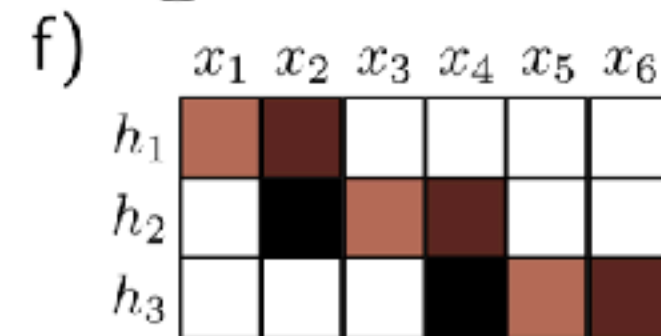
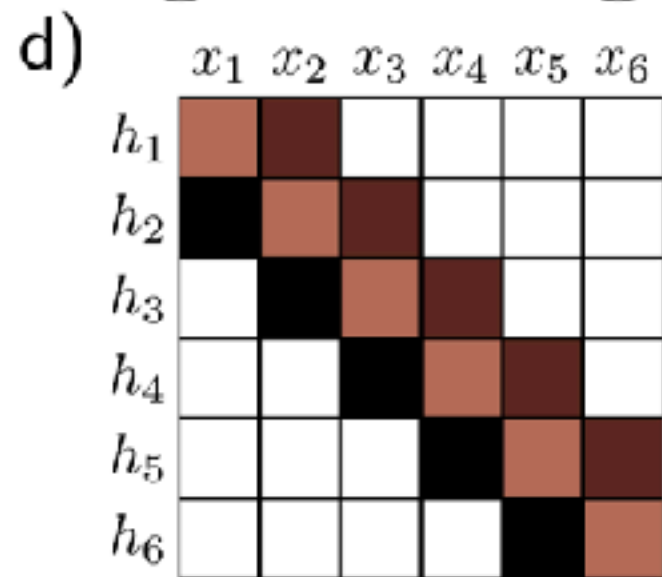
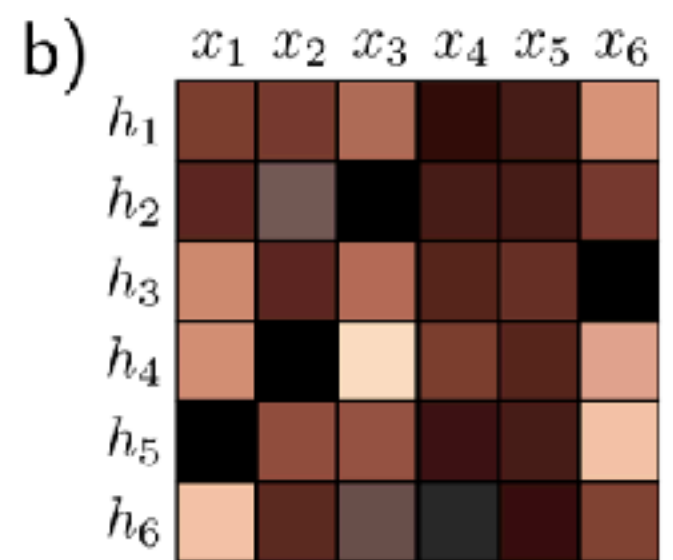
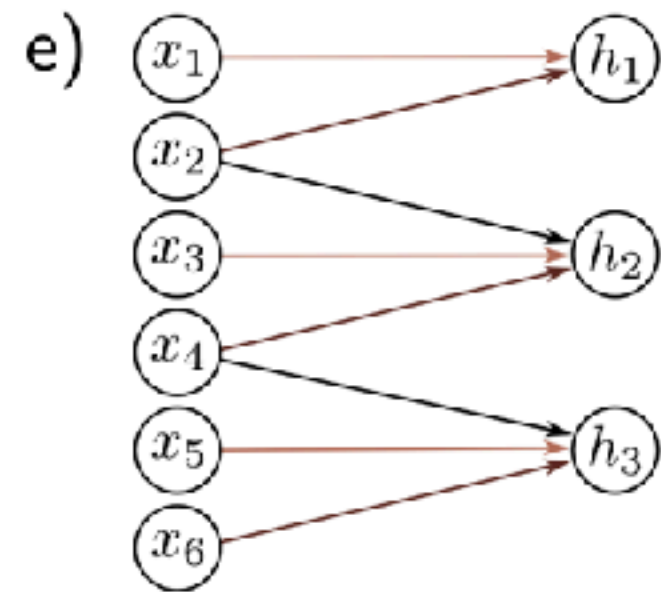
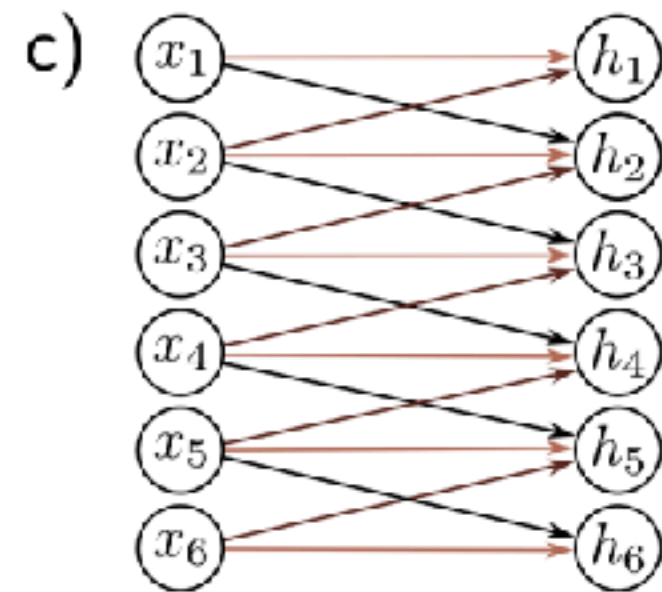
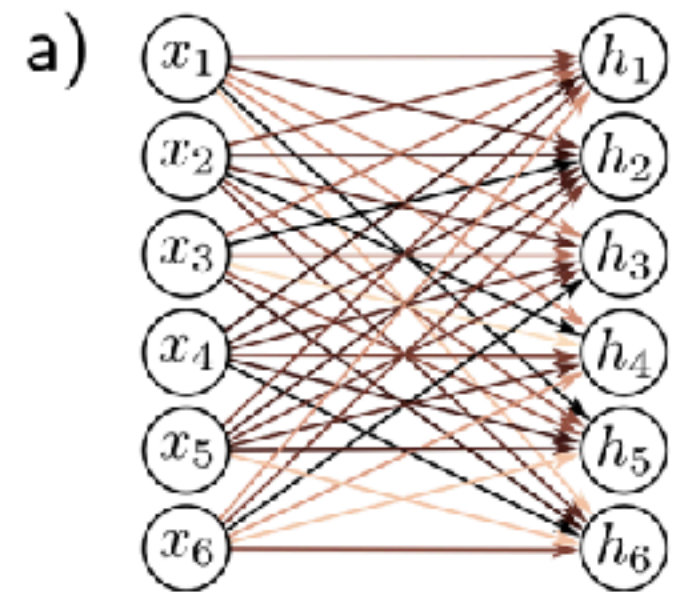
Motiváció – Invariancia és ekvivariancia

- Hogyan változik a kimenet, ha módosítjuk a bemenetet? – pl. eltolás, forgatás, tükrözés (szimmetria transzformáció)
 - Invariancia:** nem változik
 - Ekvivariancia:** a transzformációnak megfelelően változik
- Az invariancia/ekvivalencia tulajdonságot meg lehet tanulni, vagy be is lehet építeni a háló struktúrájába!
 - Pl. eltolási ekvivariancia – **konvolúciós neurális hálózat (CNN)**

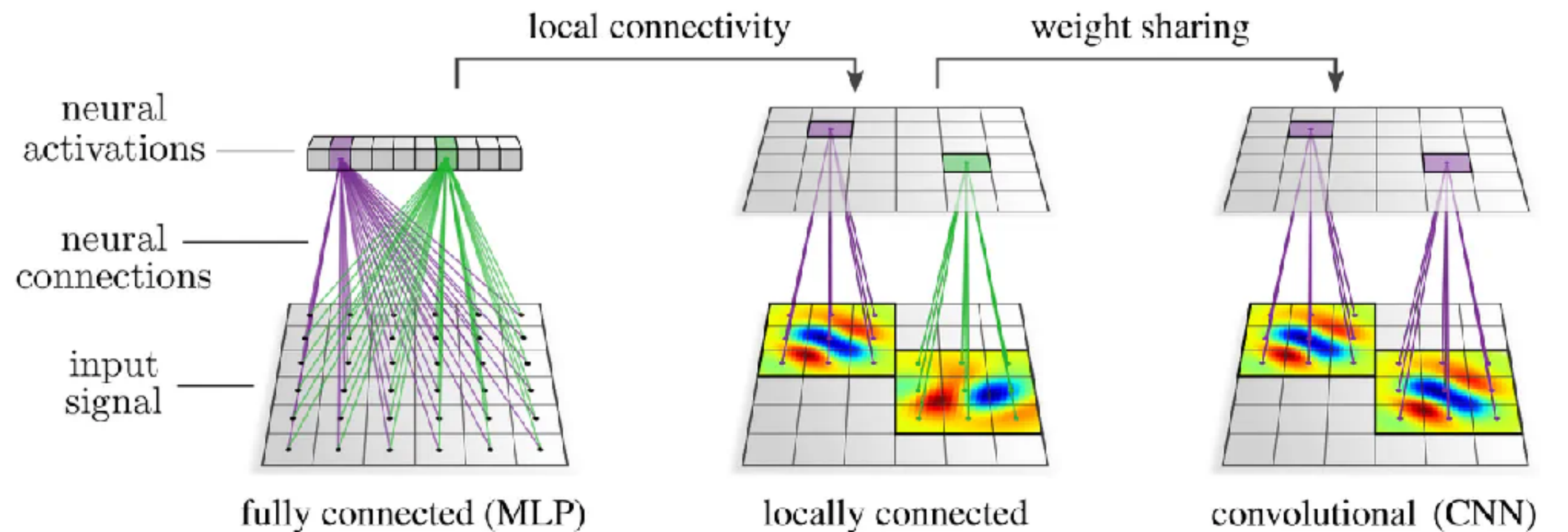


Konvolúciós Neurális Háló

MLP-ből CNN



Lokális konnektivitás
+
megosztott súlyok
=
Konvolúciós neurális háló (CNN)



Konvolúciós Neurális Háló

Konvolúciós kernel

- **Konvolúciós réteg:** minden bemeneti érték (pl. pixel) lokális környezetében alkalmaz konvolúciós szűrést (**kernel**), tanult súlyokkal
- (Szigorúan véve ez *korreláció*, de konvolúciónak szokás nevezni...)

Input Kernel Output

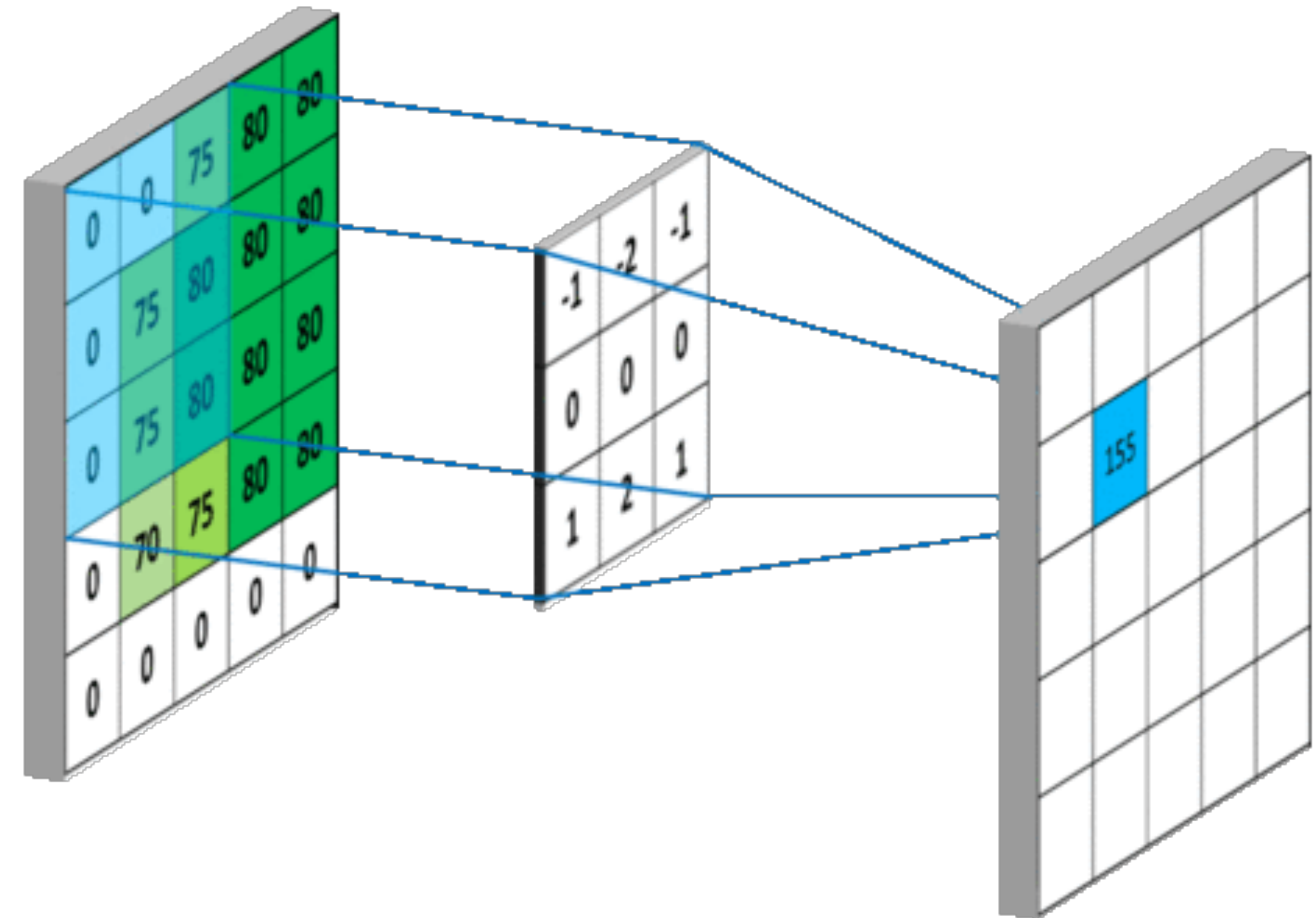
0	1	2
3	4	5
6	7	8

 *

0	1
2	3

 =

19	25
37	43



Konvolúciós Neurális Háló

Konvolúciós kernel

- **Konvolúciós réteg:** minden bemeneti érték (pl. pixel) lokális környezetében alkalmaz konvolúciós szűrést (**kernel**), tanult súlyokkal
- (Szigorúan véve ez *korreláció*, de konvolúciónak szokás nevezni...)

Input Kernel Output

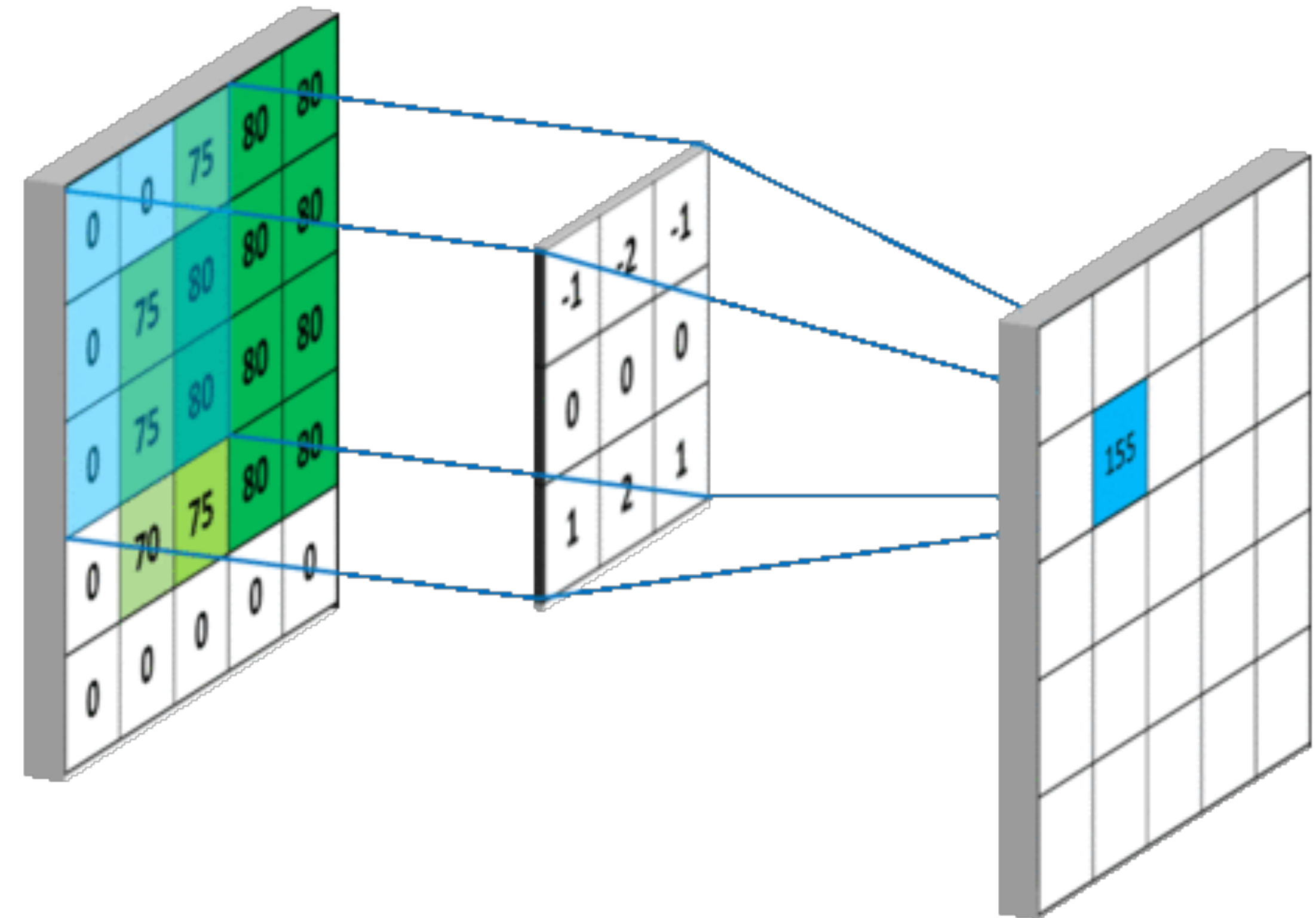
0	1	2
3	4	5
6	7	8

 *

0	1
2	3

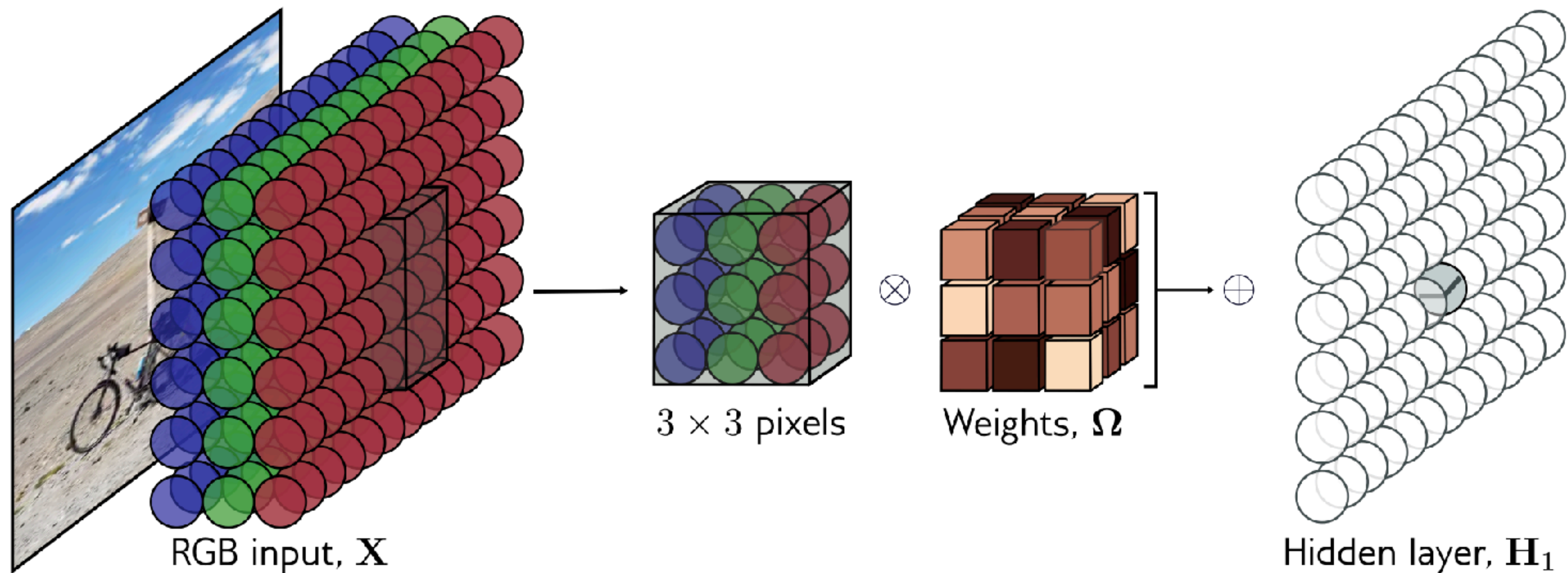
 =

19	25
37	43



Konvolúciós Neurális Háló

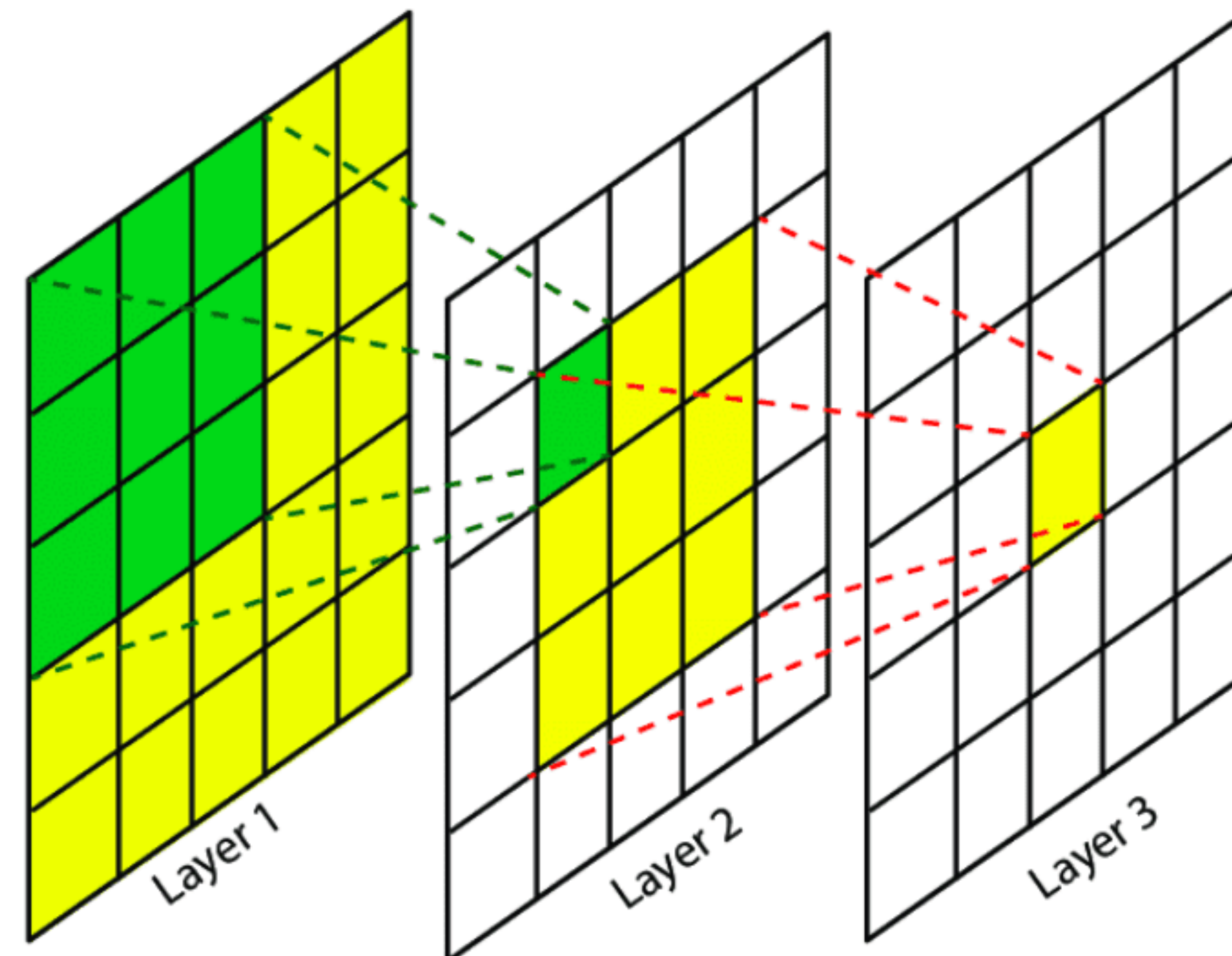
Többcsatornás konvolúció



Több kimenetű rejtett rétegek esetén hasonlóan járunk el...

Konvolúciós Neurális Háló

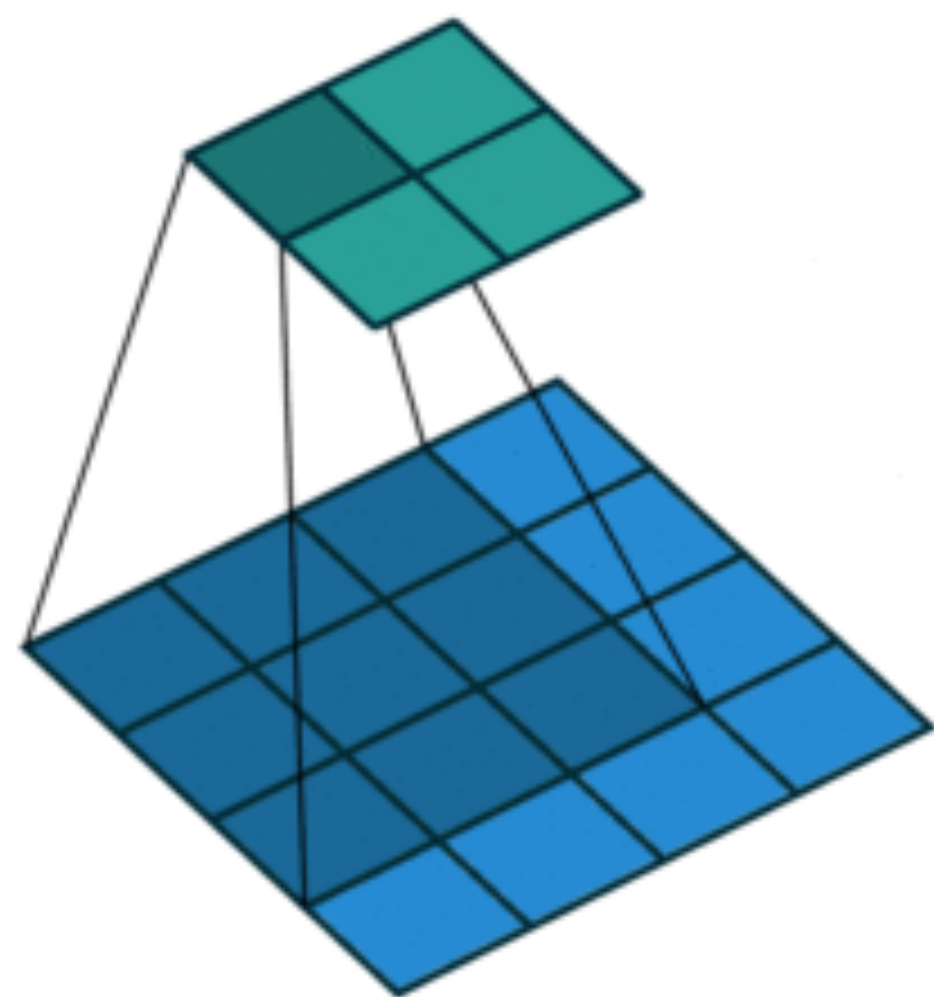
Receptív Mező



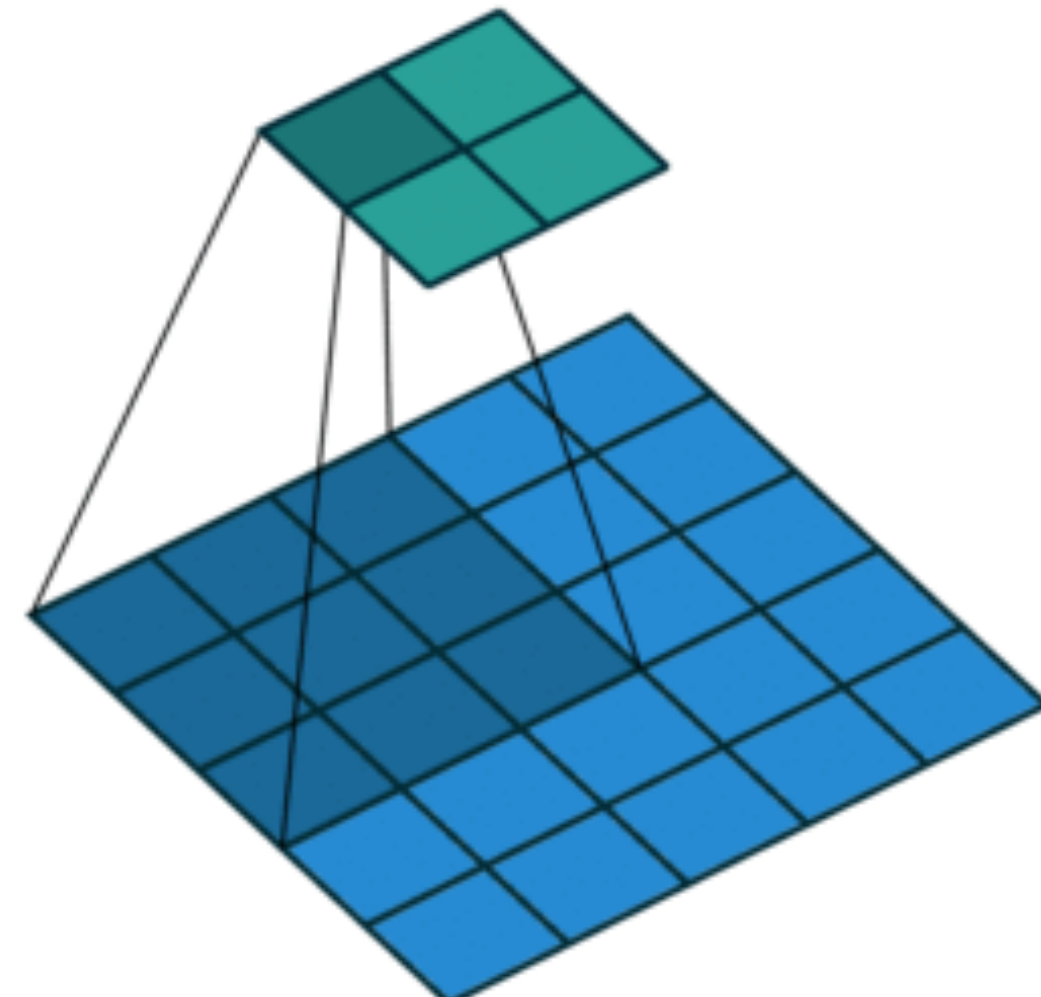
A későbbi rétegek nagyobb területet “látnak”

Konvolúciós Neurális Háló

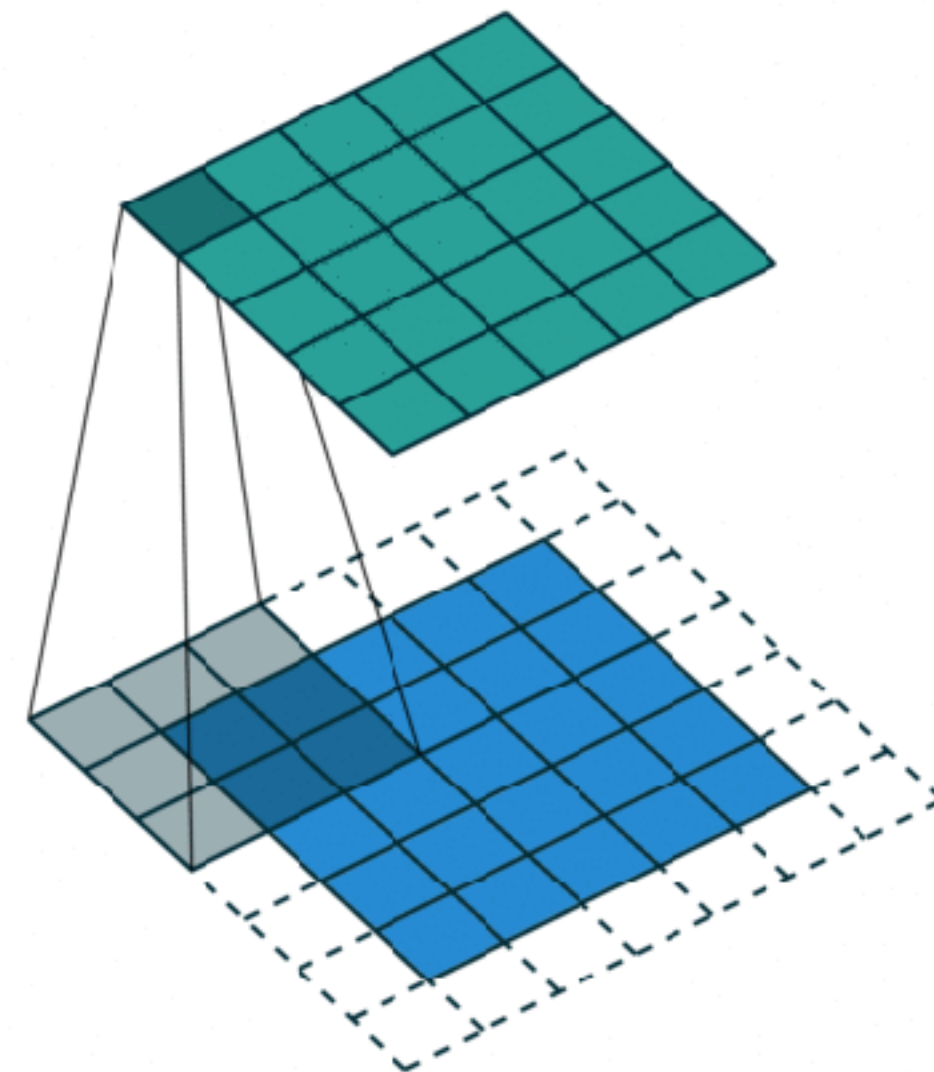
Stride, Padding, Dilation



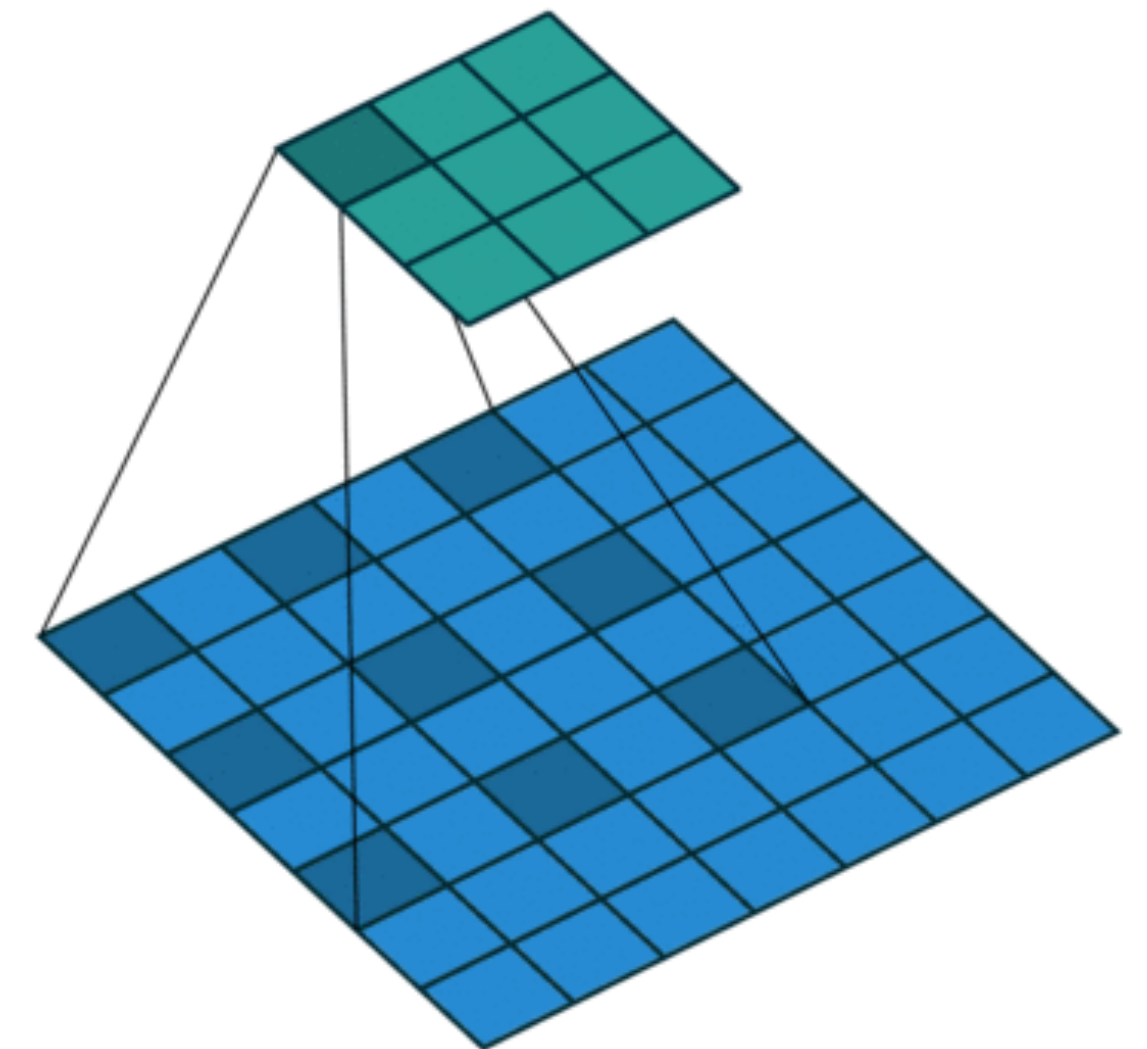
Stride = 0
Padding = 0
Dilation = 0



Stride = 1
Padding = 0
Dilation = 0



Stride = 0
Padding = 1
Dilation = 0

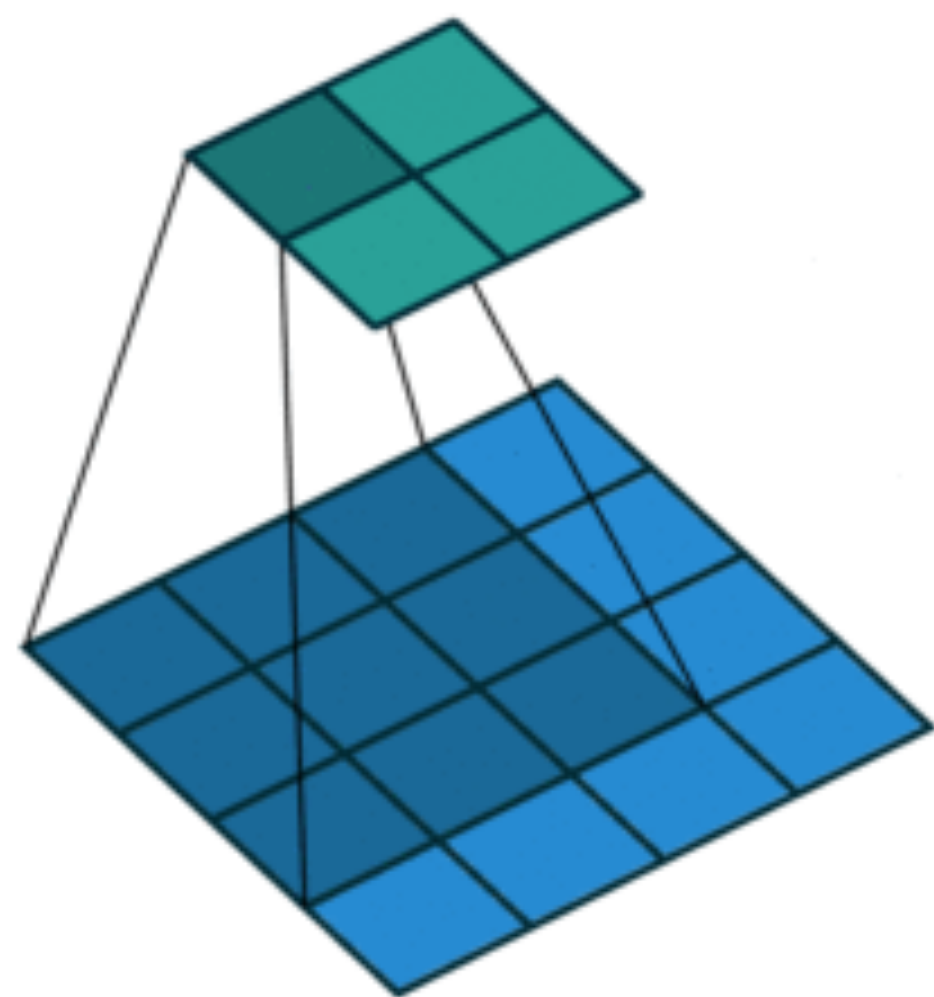


Stride = 0
Padding = 0
Dilation = 1

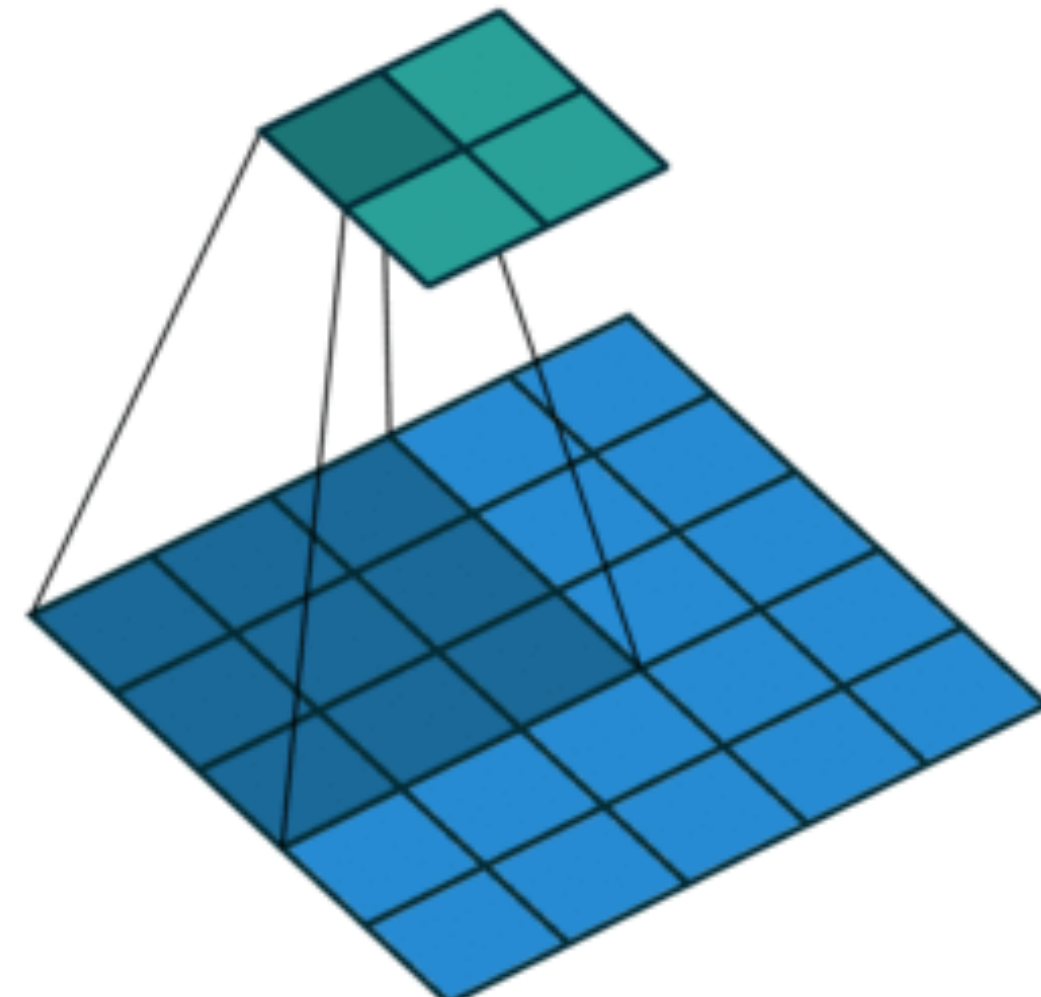
Forrás: https://github.com/vdumoulin/conv_arithmetic/

Konvolúciós Neurális Háló

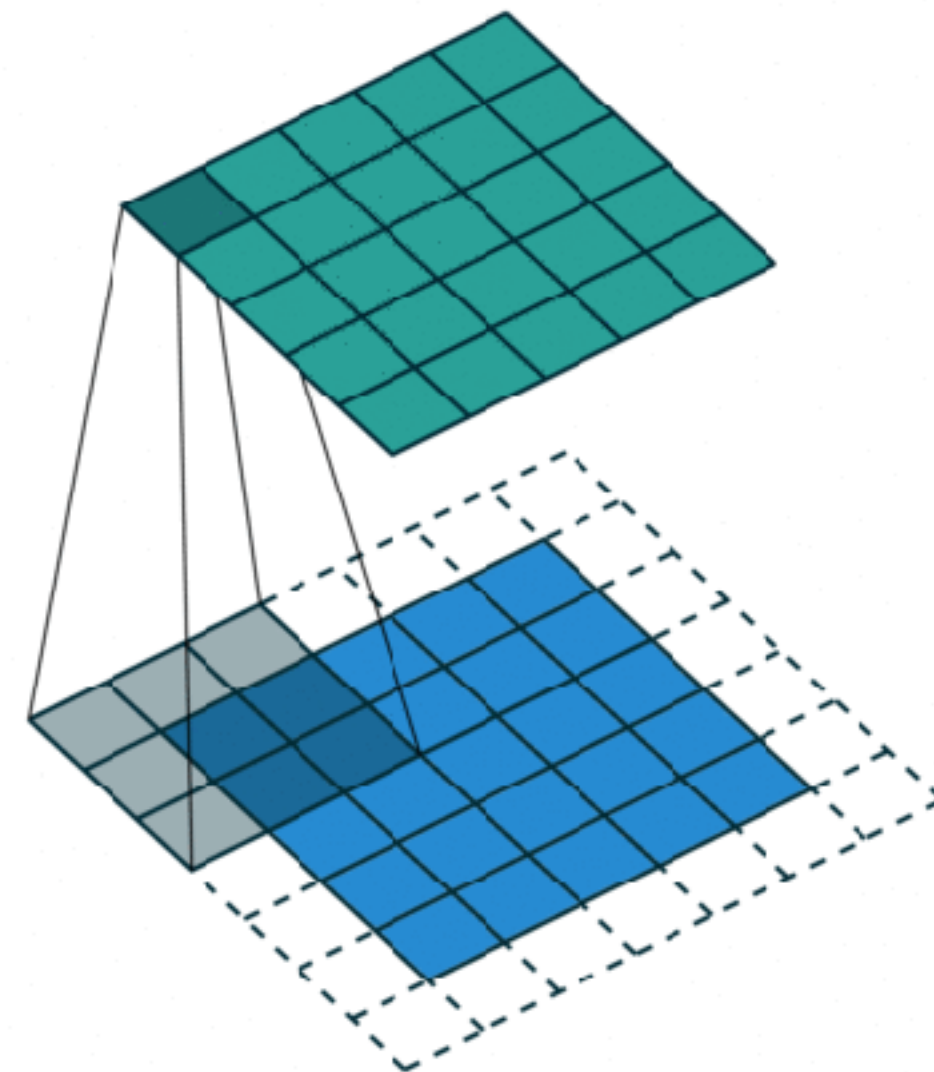
Stride, Padding, Dilation



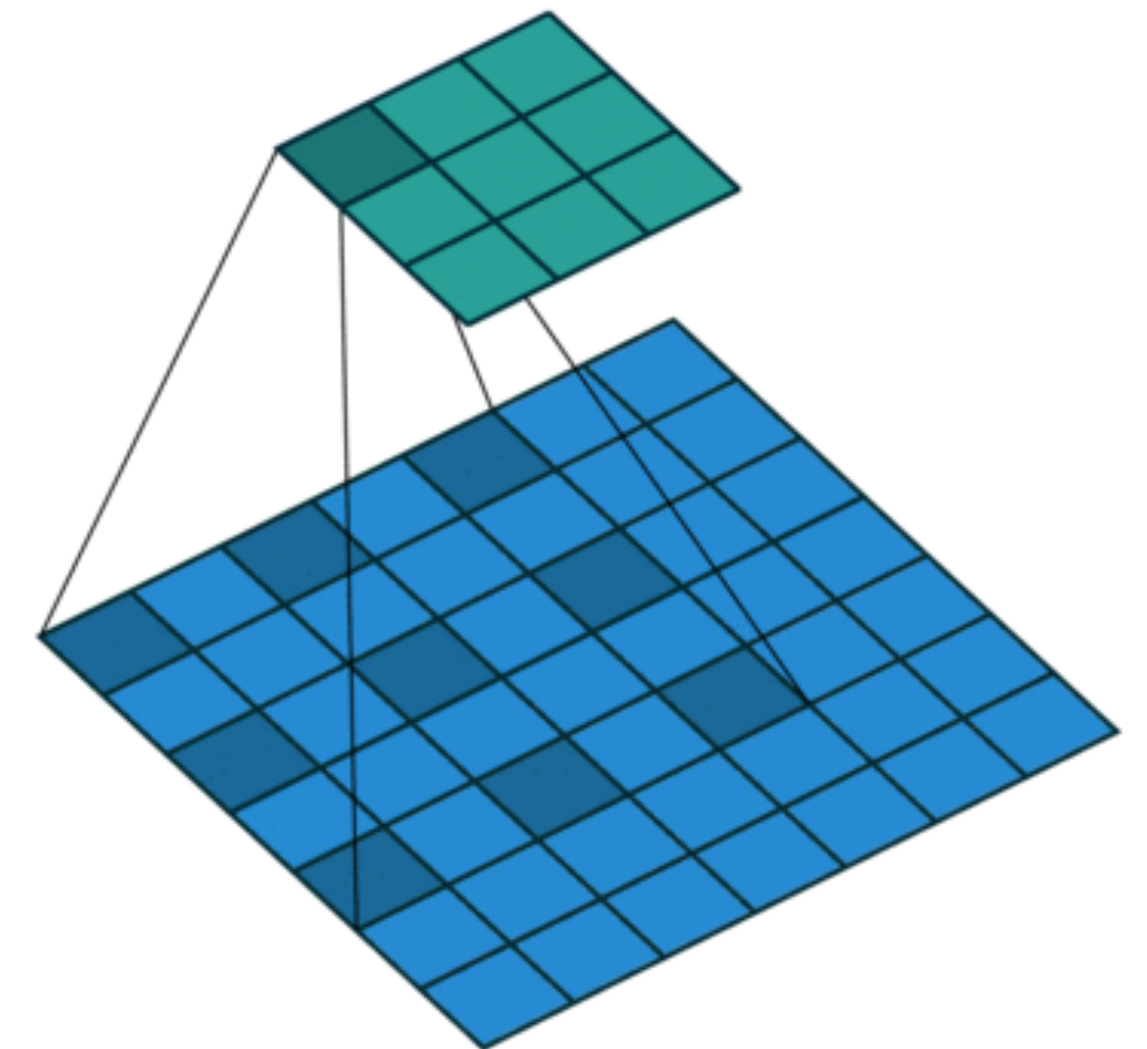
Stride = 0
Padding = 0
Dilation = 0



Stride = 1
Padding = 0
Dilation = 0



Stride = 0
Padding = 1
Dilation = 0

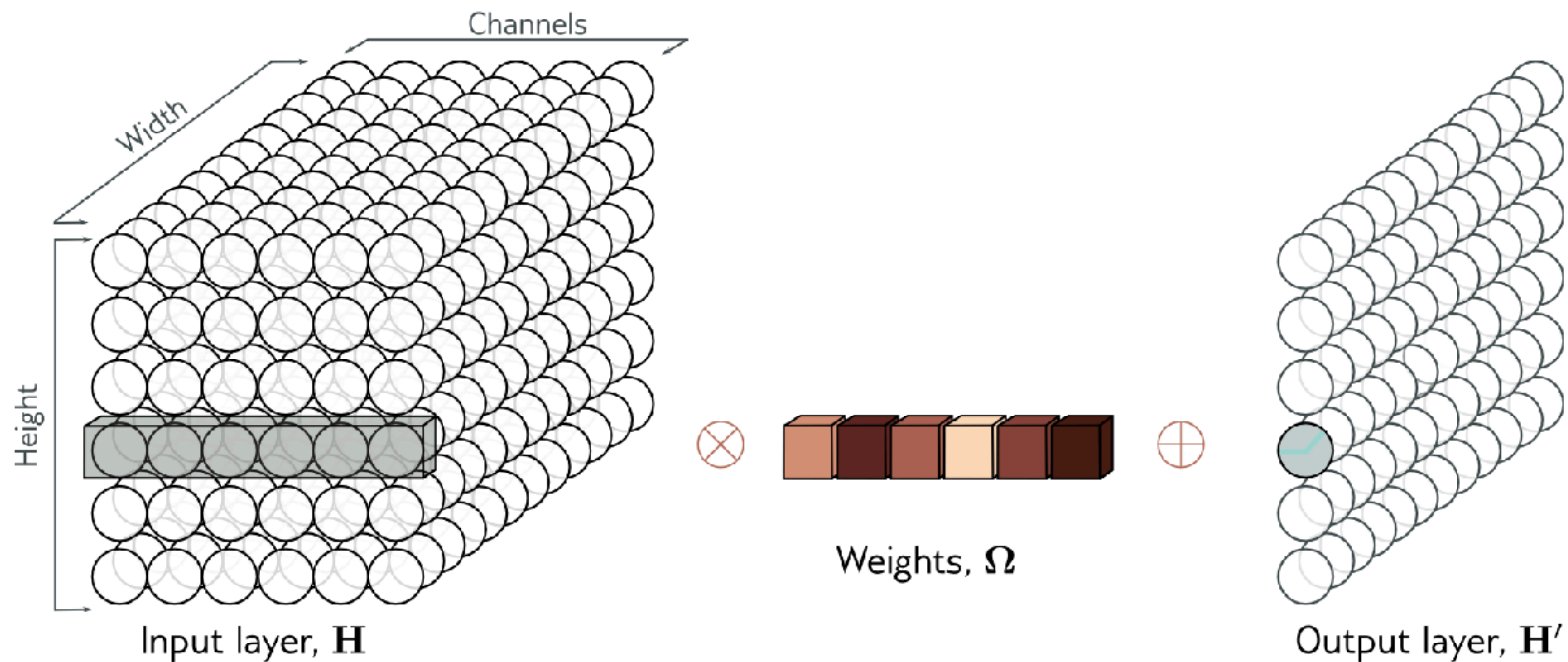


Stride = 0
Padding = 0
Dilation = 1

Forrás: https://github.com/vdumoulin/conv_arithmetic/

Konvolúció Neurális Háló

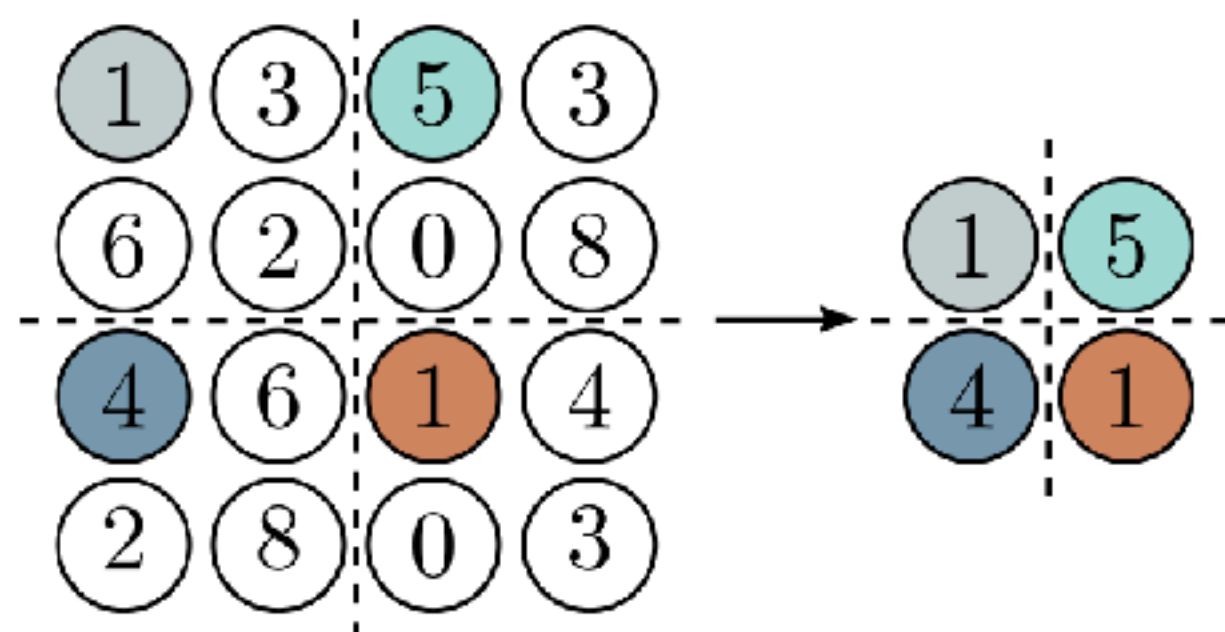
1x1 Konvolúció



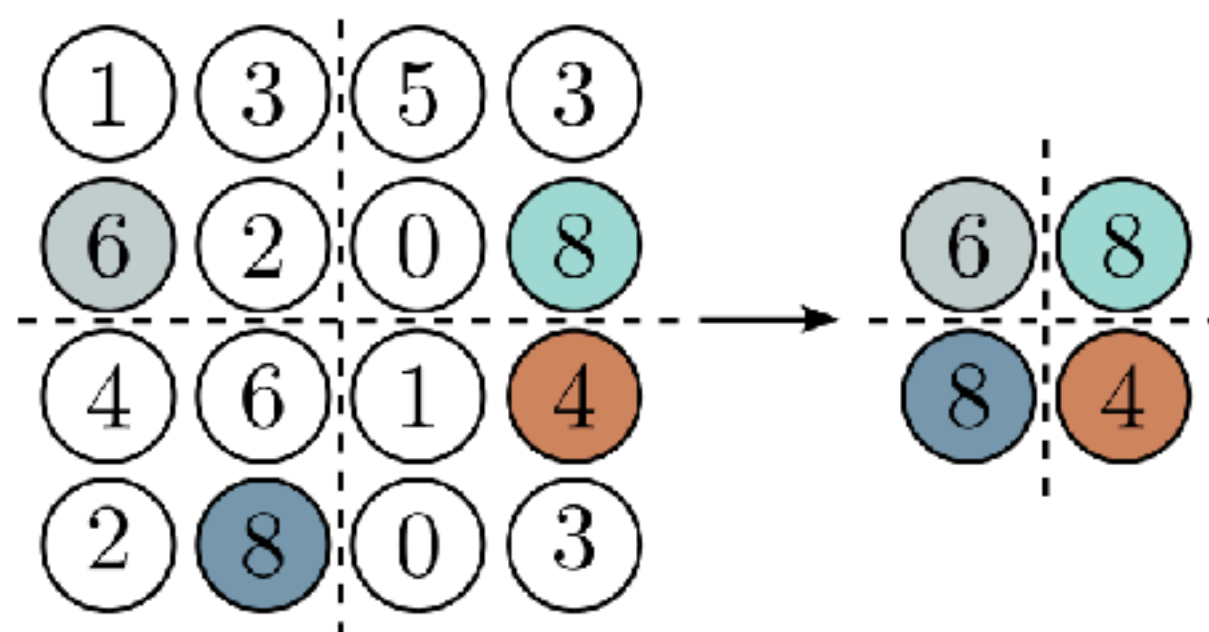
1x1 konvolúció réteg = csatorna-menti MLP

Konvolúciós Neurális Háló

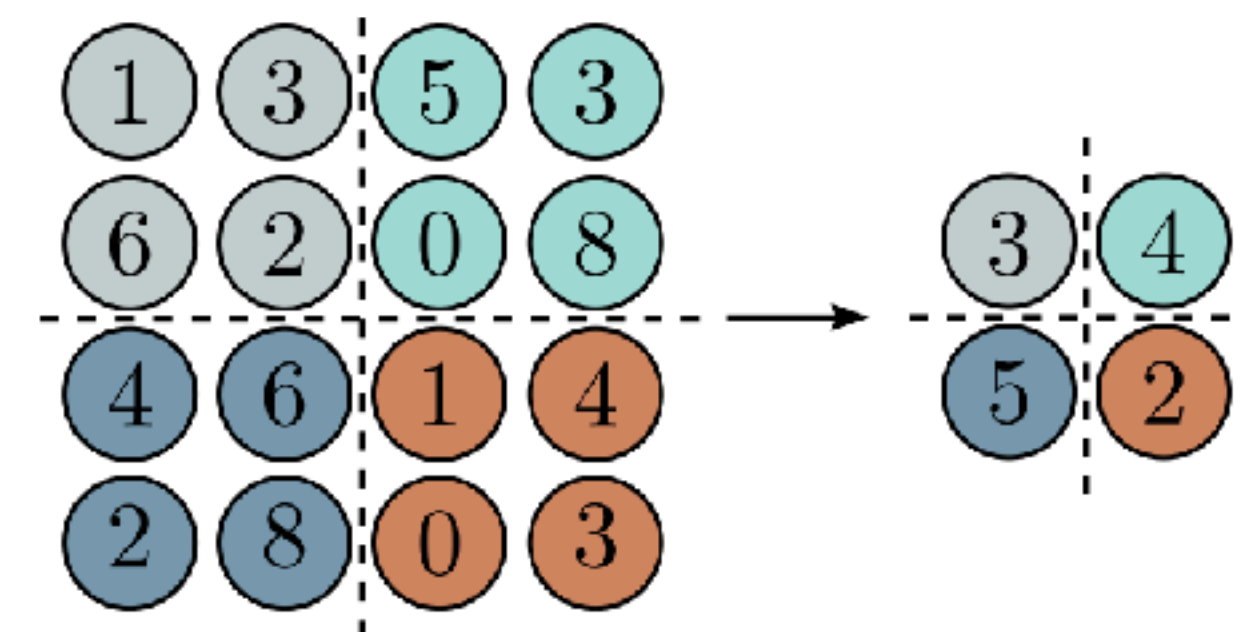
Pooling, le/felskálázás



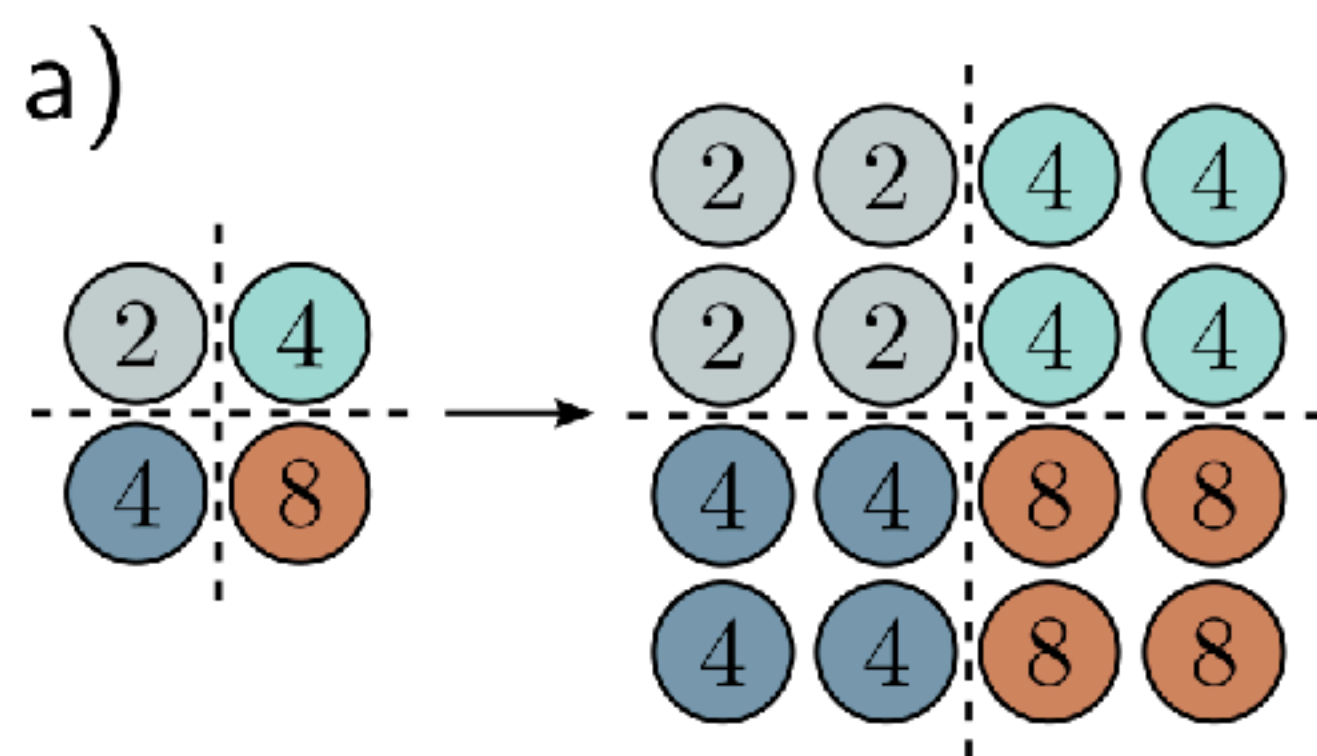
Leskálázás



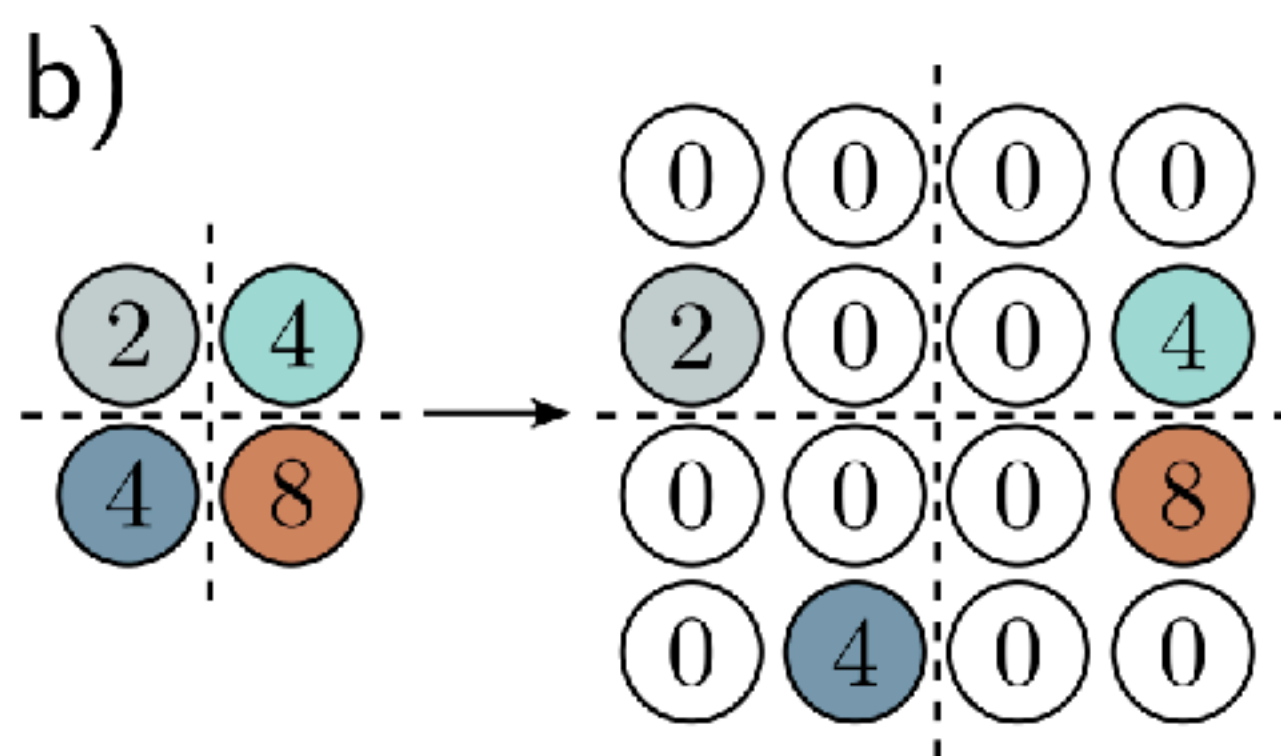
Max pooling



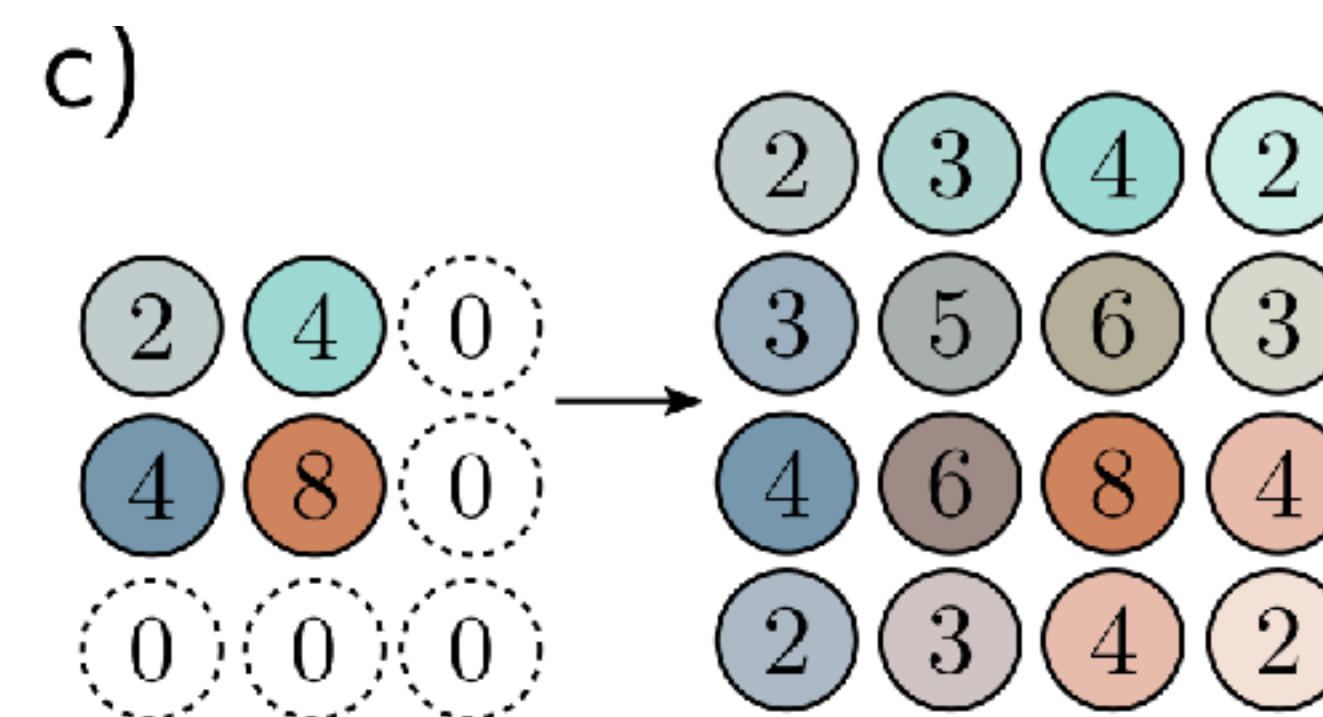
Avg pooling



Felskálázás



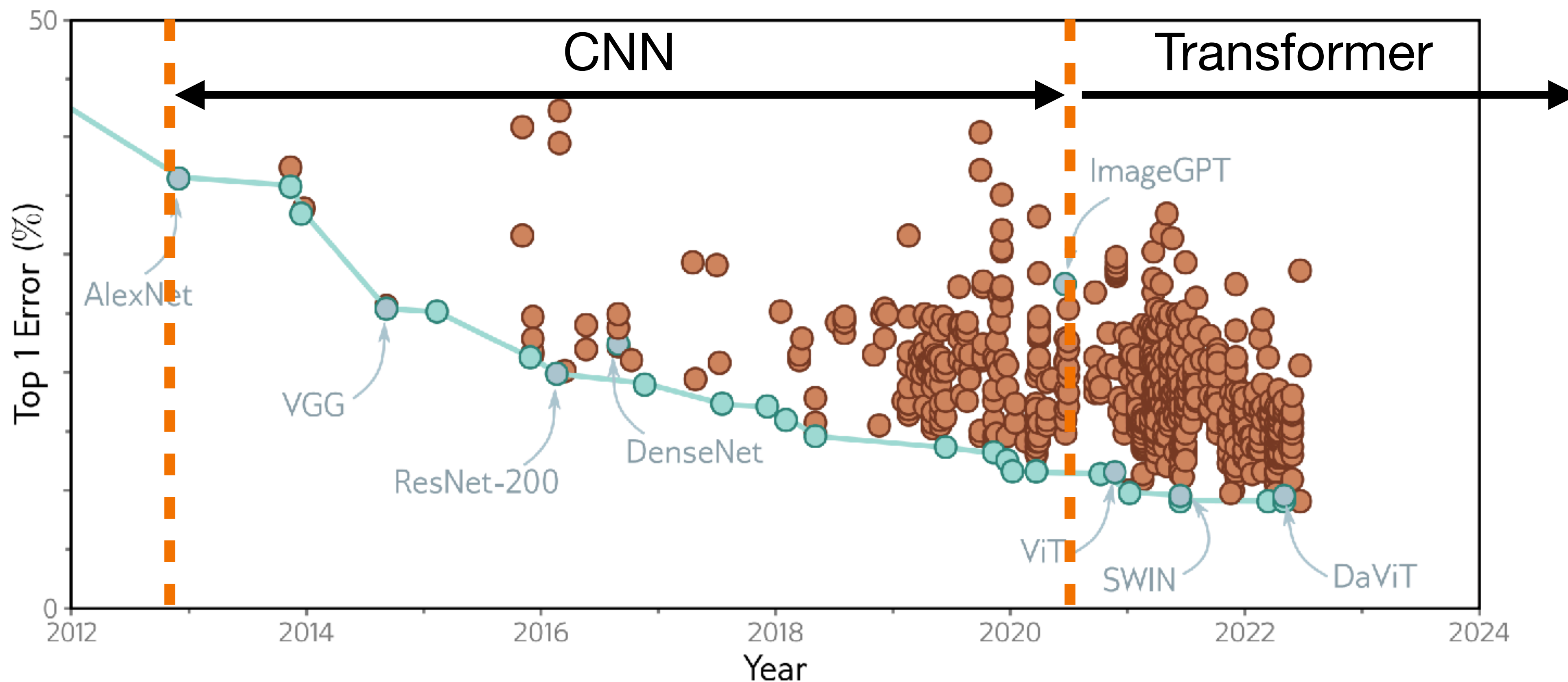
Unpooling



Bilineáris interpoláció

Konvolúciós Neurális Háló

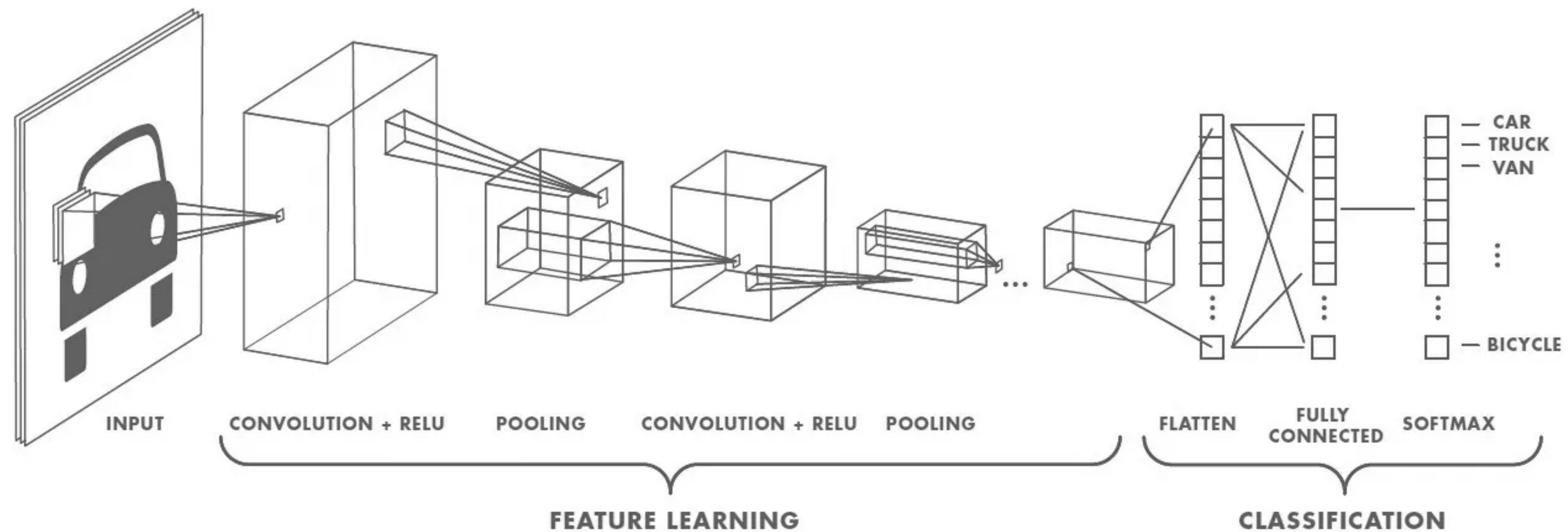
Architektúrák fejlődése



ImageNet max. hiba fokozatosan csökken!

Konvolúciós Neurális Háló

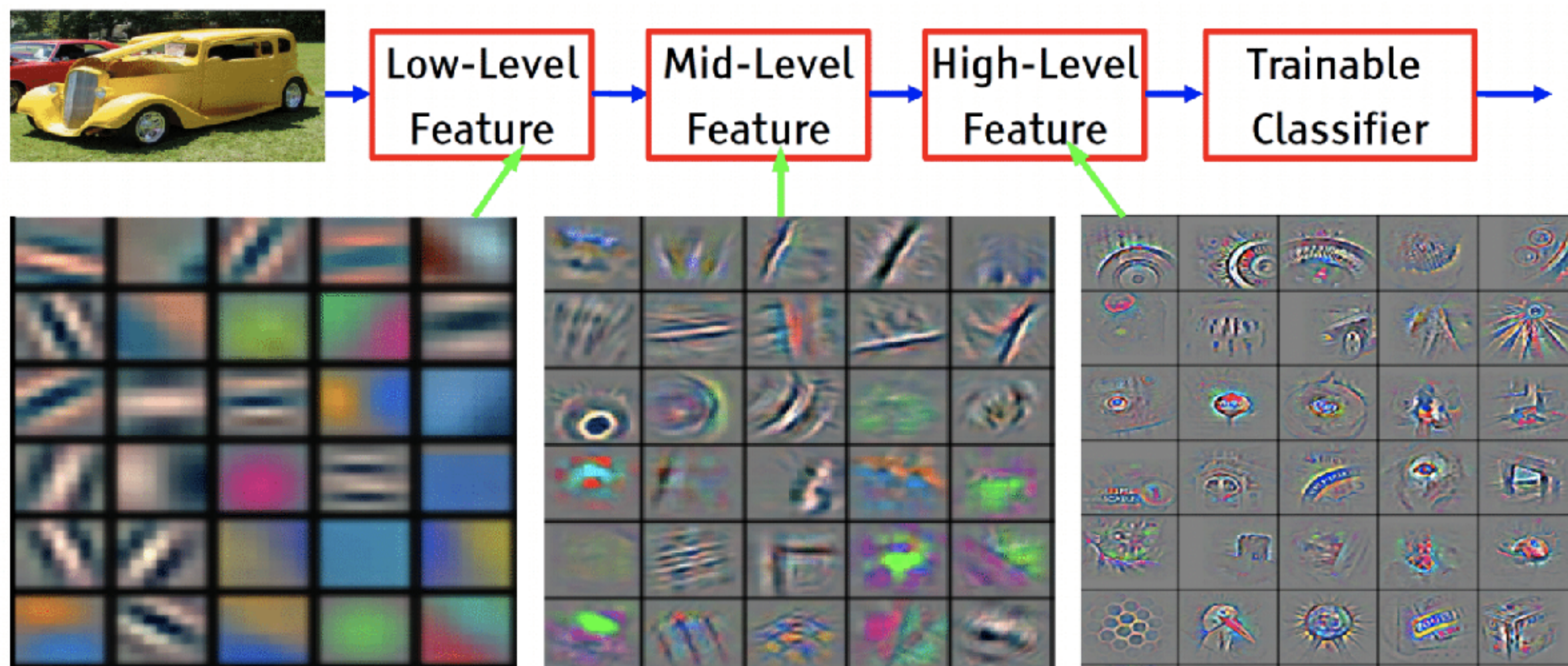
Architektúrák – Felépítés



CNN architektúrák tipikus felépítése

Konvolúciós Neurális Háló

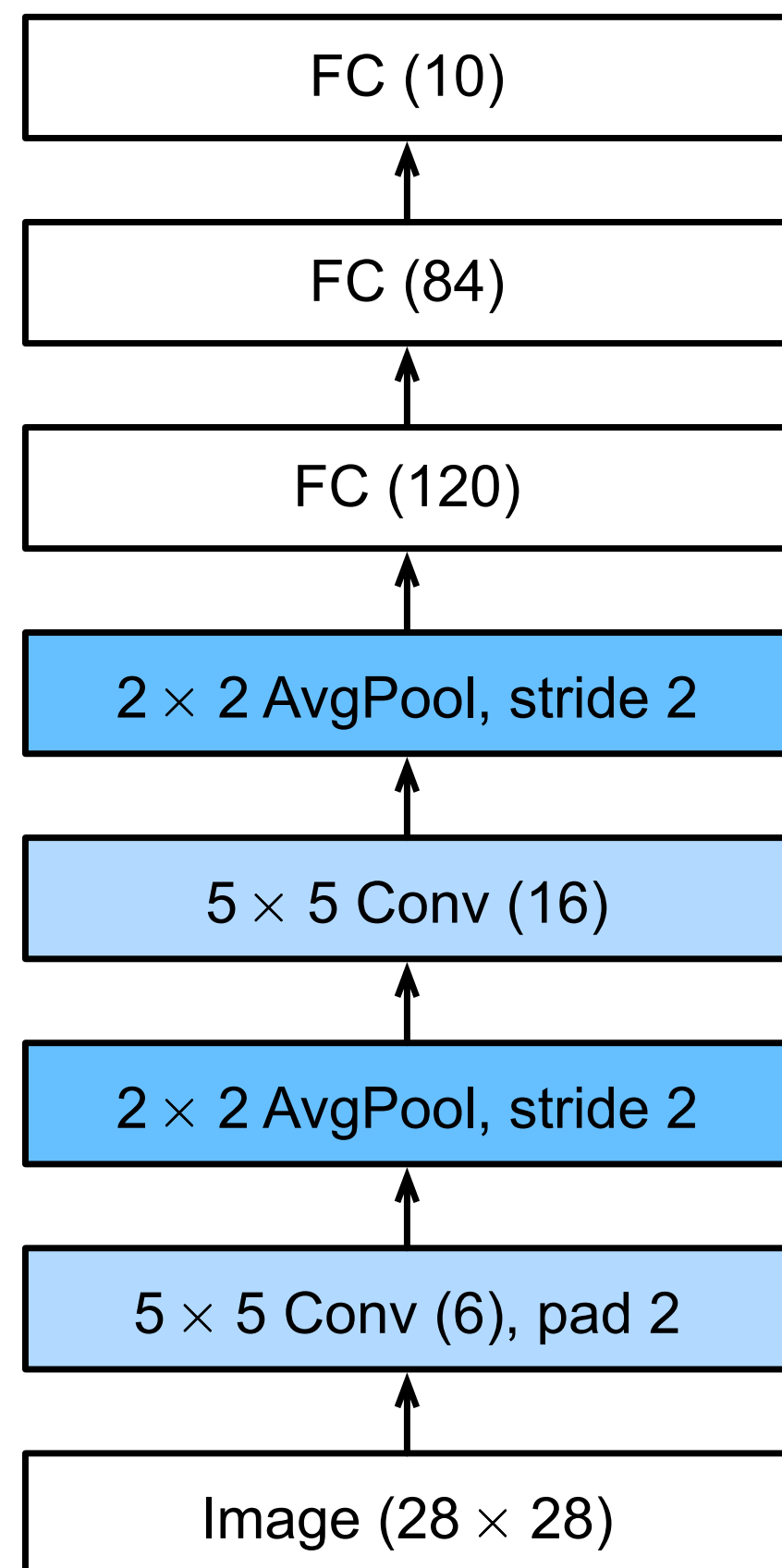
Architektúrák – Feature-ök



Tipikus CNN feature hierarchia
(Érdeklődők számára érdekes olvasmány: [Circuits](#))

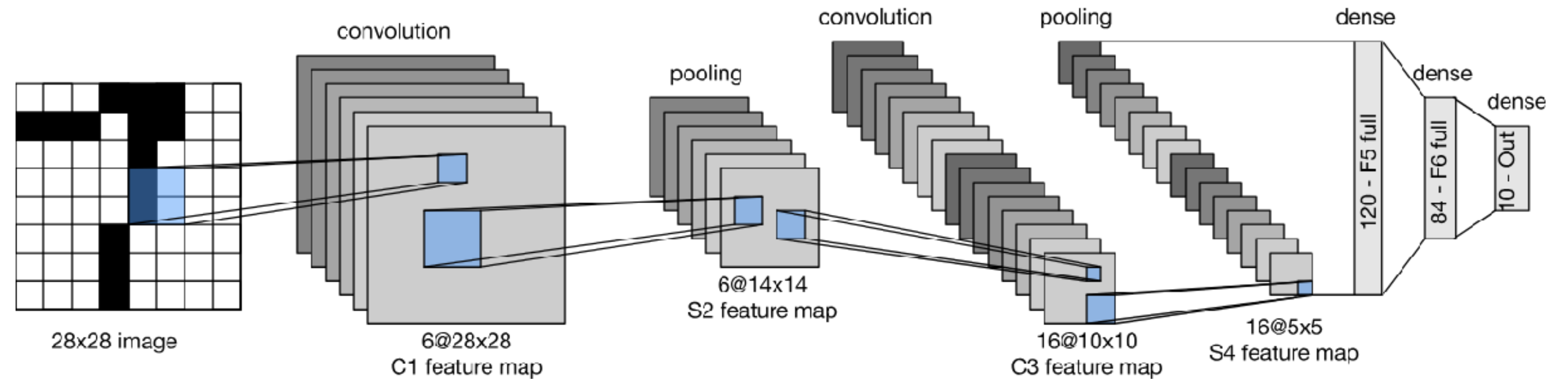
Konvolúciós Neurális Háló

Architektúrák – LeNet [1998]



Gradient-Based Learning Applied to Document Recognition

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner



Konvolúciós Neurális Háló

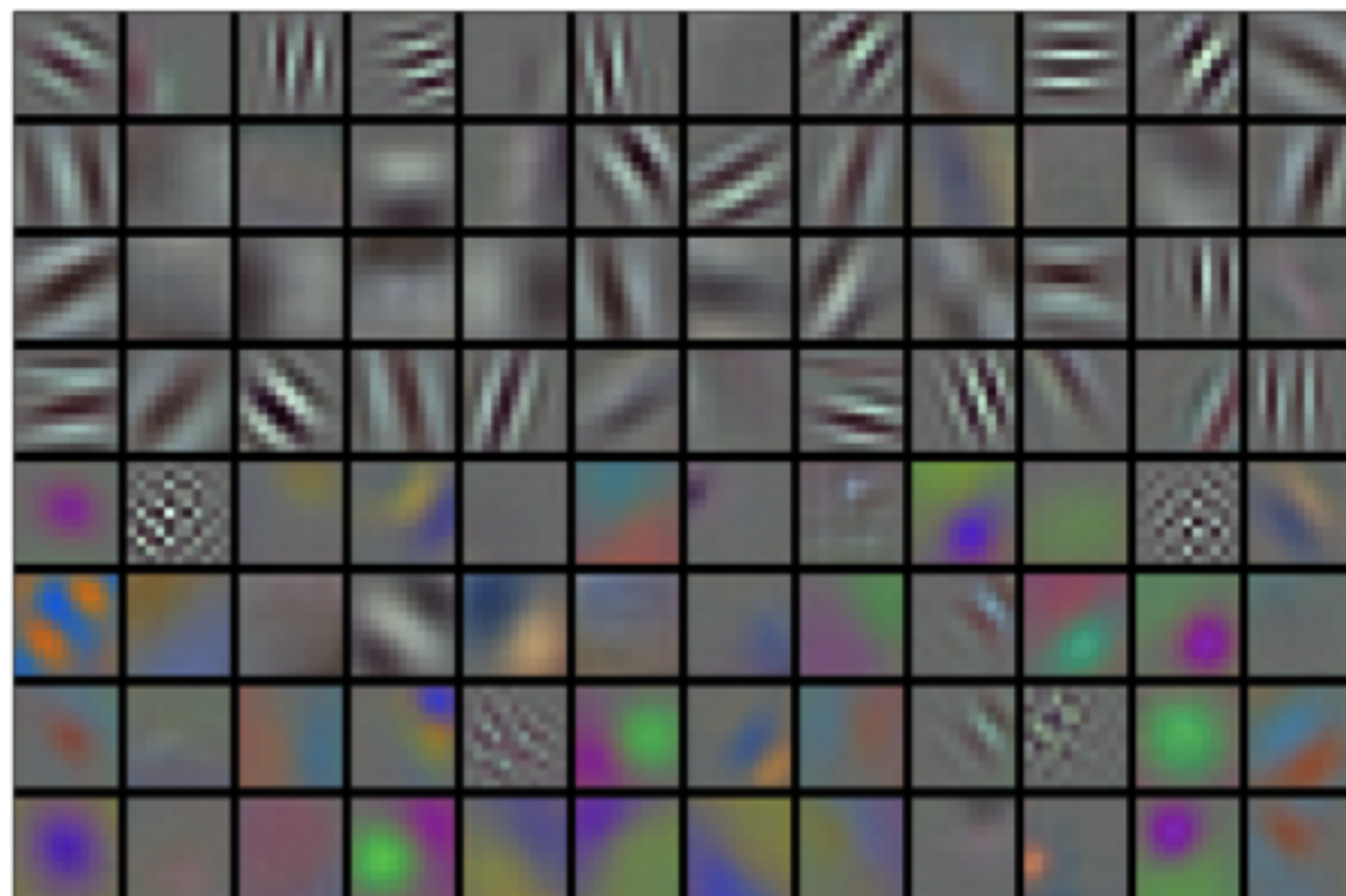
Architektúrák – AlexNet [2012]

ImageNet Classification with Deep Convolutional
Neural Networks

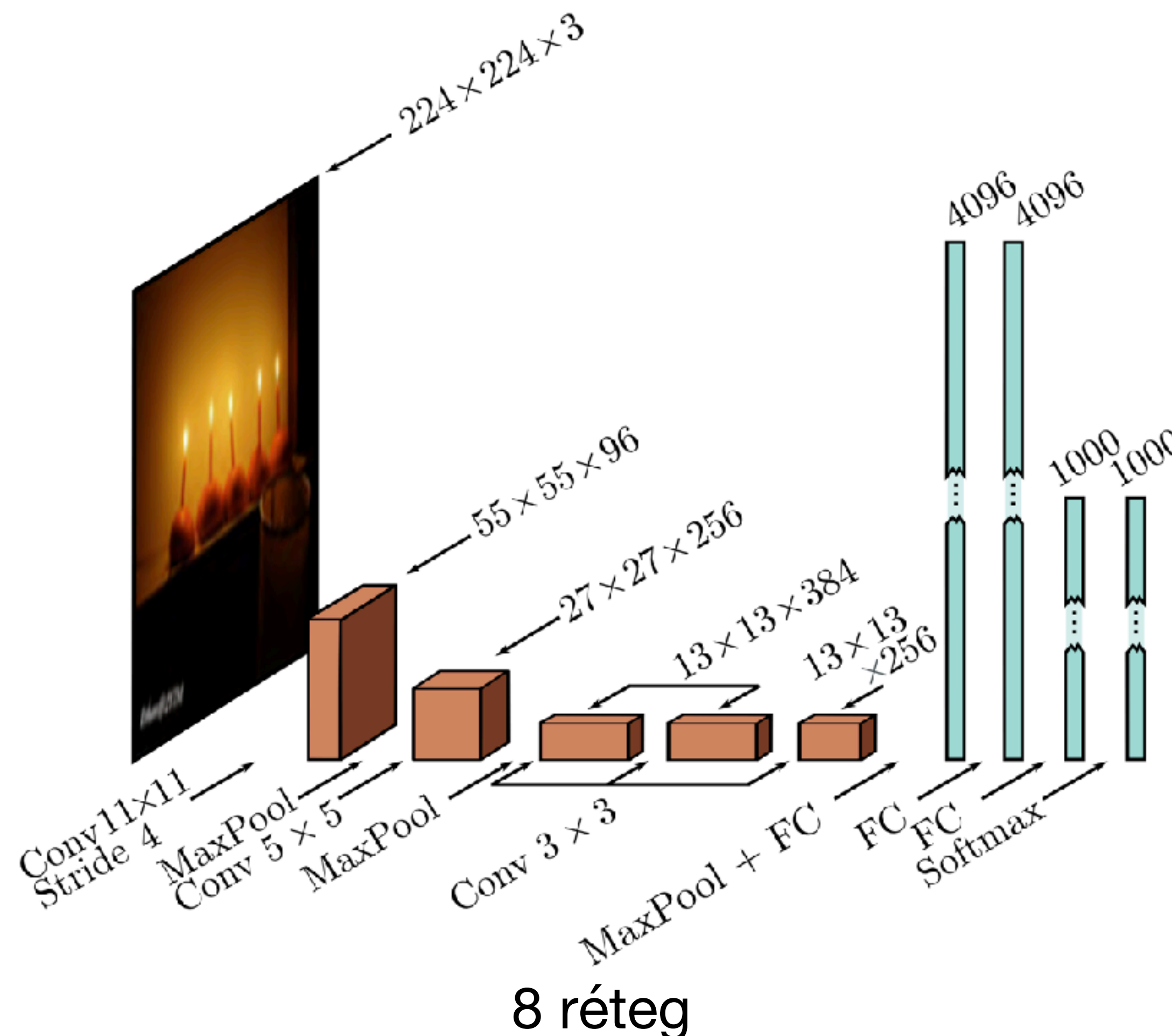
Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca



1. réteg által megtanult szűrők

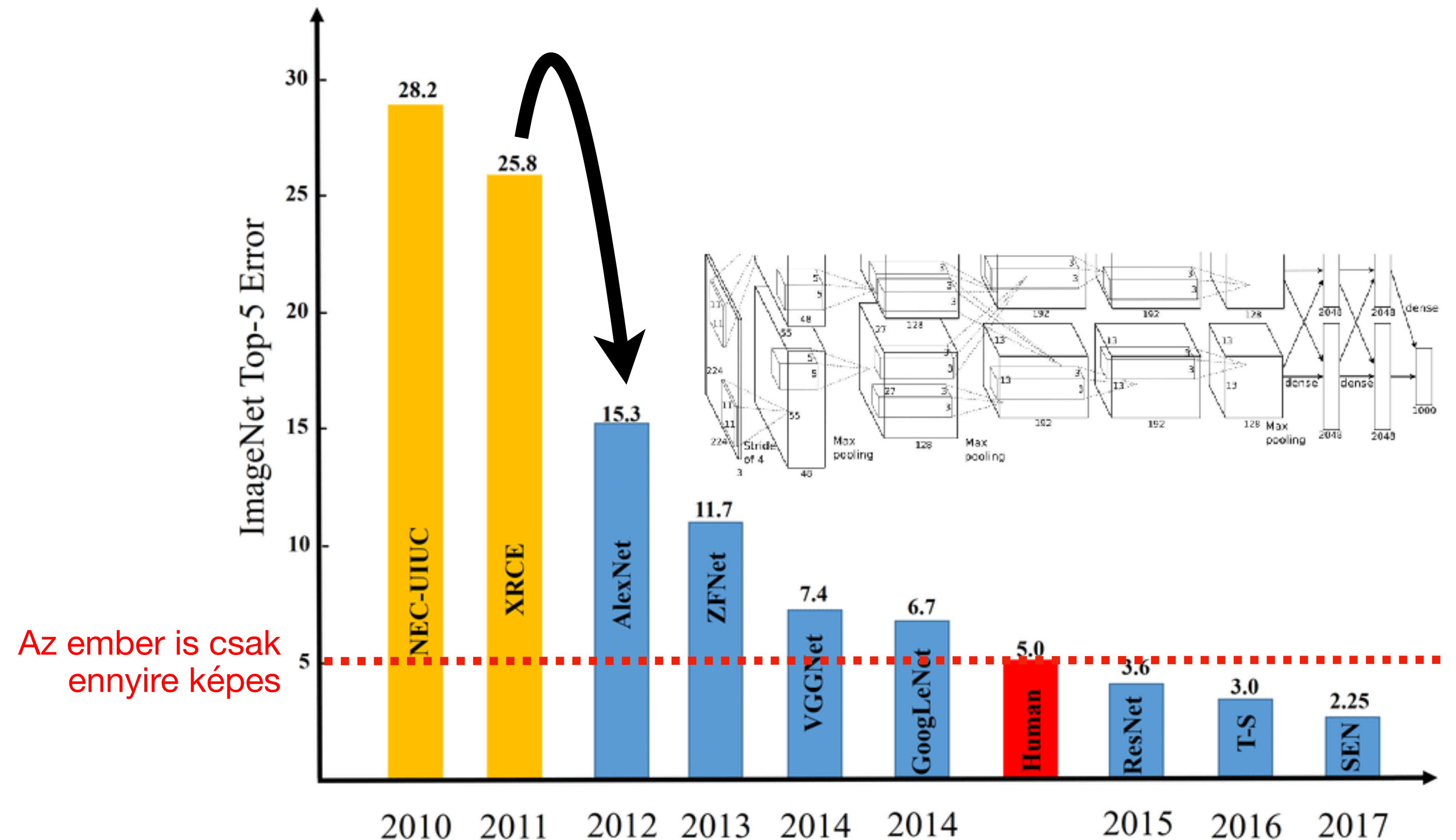


Konvolúciós Neurális Háló

Architektúrák – AlexNet [2012]

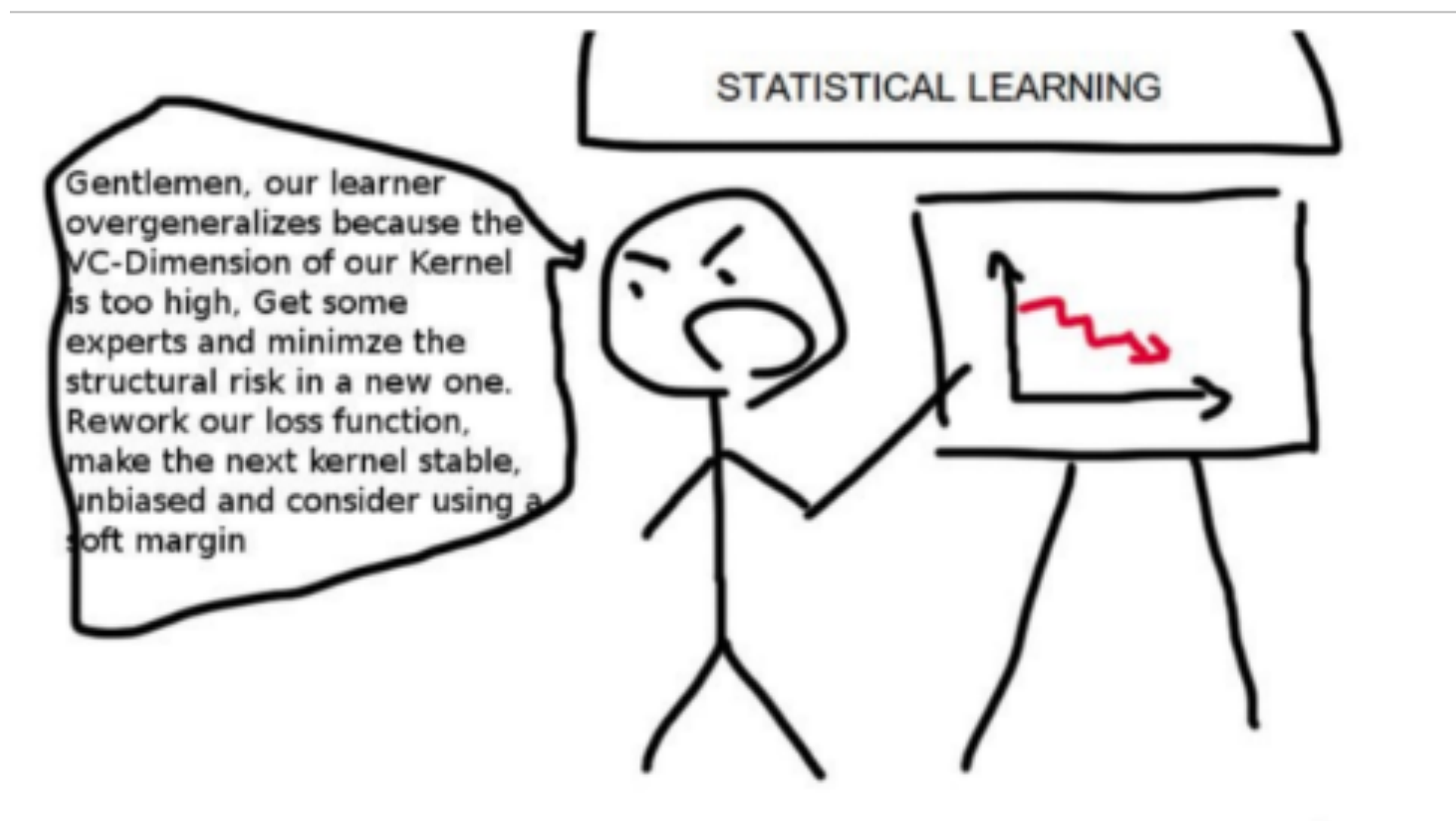
Mi kellett végül a mély neurális háló átöréséhez?

- Nagy mennyiségű és jó minőségű tanító adat (ImageNet)
- Megfelelő aktivációs függvények (ReLU)
- Megfelelő inicializáció
- Megfelelő regularizáció (dropout, normalizációs rétegek)
- Párhuzamos hardver (GPU)

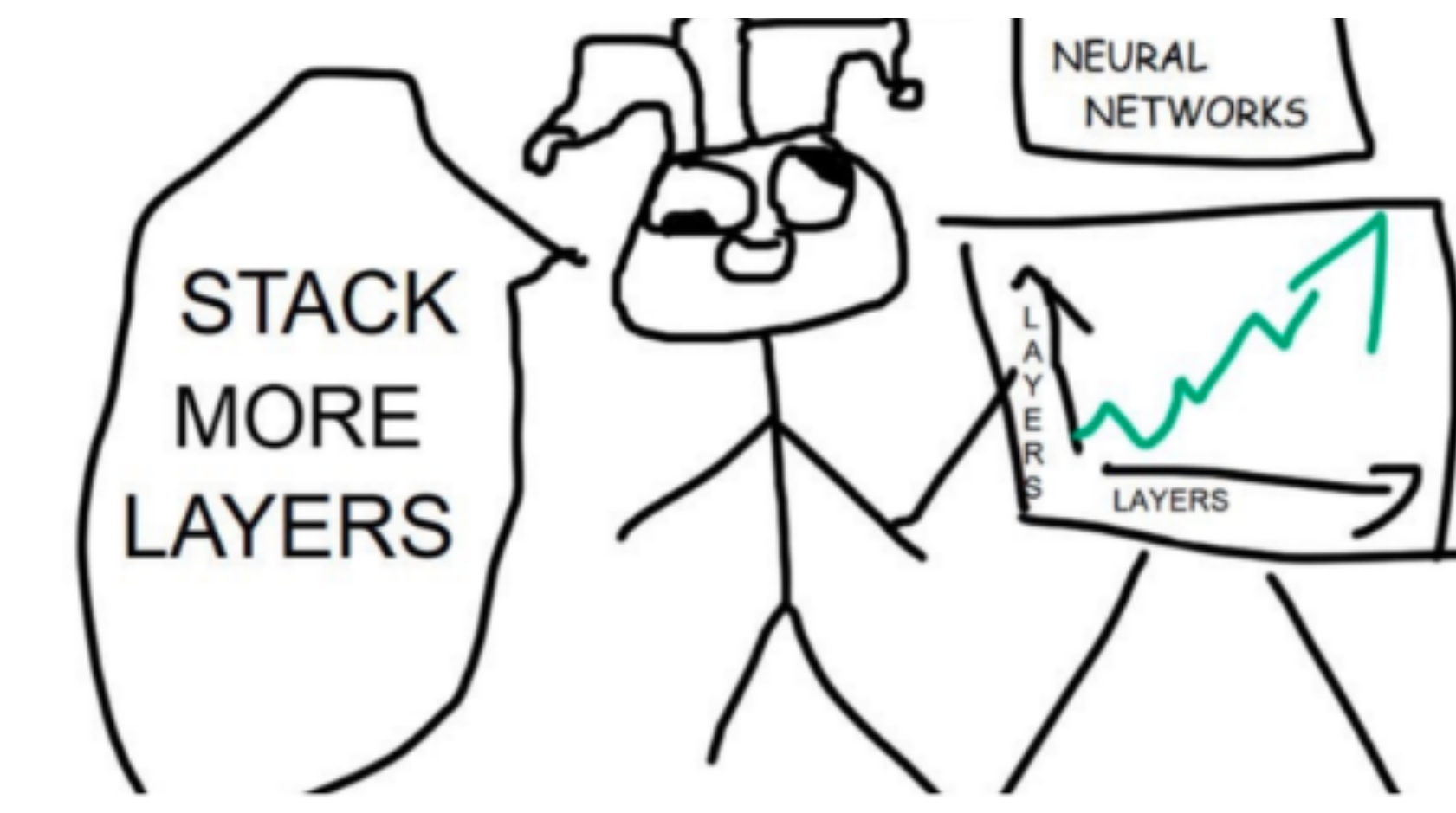


Konvolúciós Neurális Háló

Architektúrák – AlexNet [2012]



AlexNet előtt



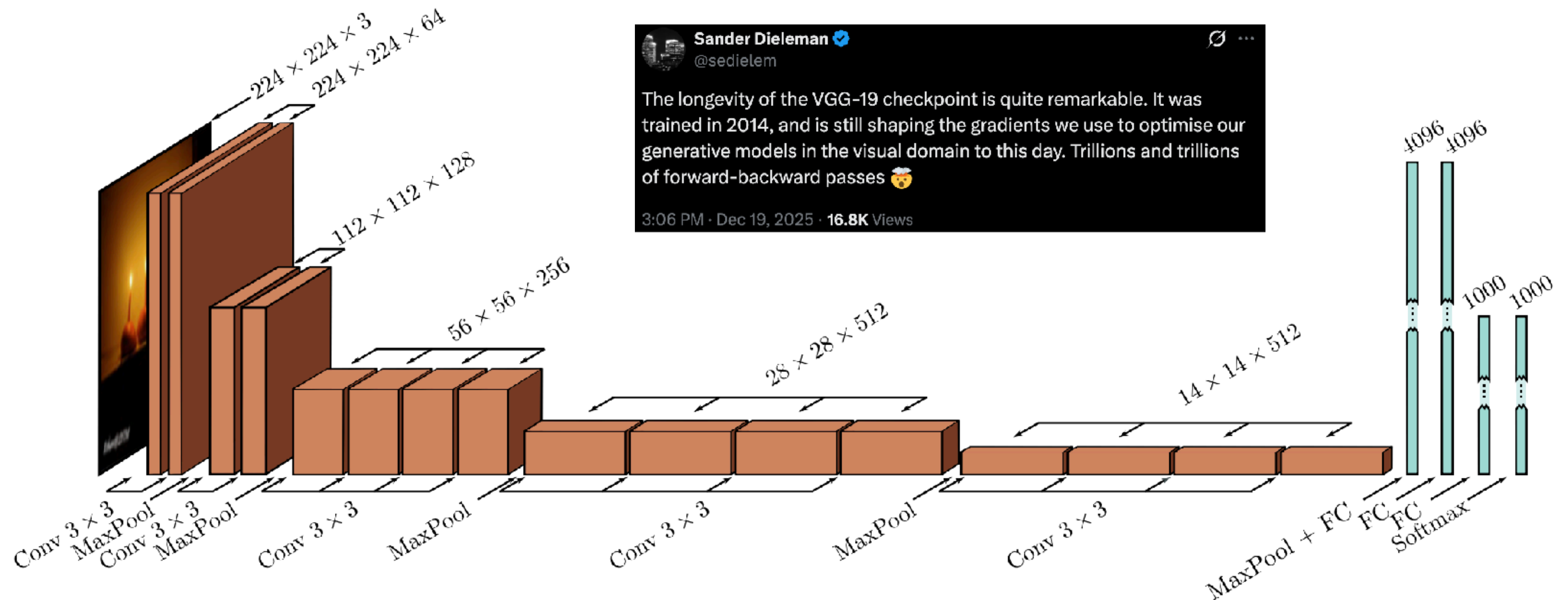
AlexNet után

Konvolúciós Neurális Háló

Architektúrák – VGG [2014]

VERY DEEP CONVOLUTIONAL NETWORKS
FOR LARGE-SCALE IMAGE RECOGNITION

Karen Simonyan* & Andrew Zisserman+
Visual Geometry Group, Department of Engineering Science, University of Oxford



Sander Dieleman @sedielem

The longevity of the VGG-19 checkpoint is quite remarkable. It was trained in 2014, and is still shaping the gradients we use to optimise our generative models in the visual domain to this day. Trillions and trillions of forward-backward passes 🤖

3:06 PM · Dec 19, 2025 · 16.8K Views

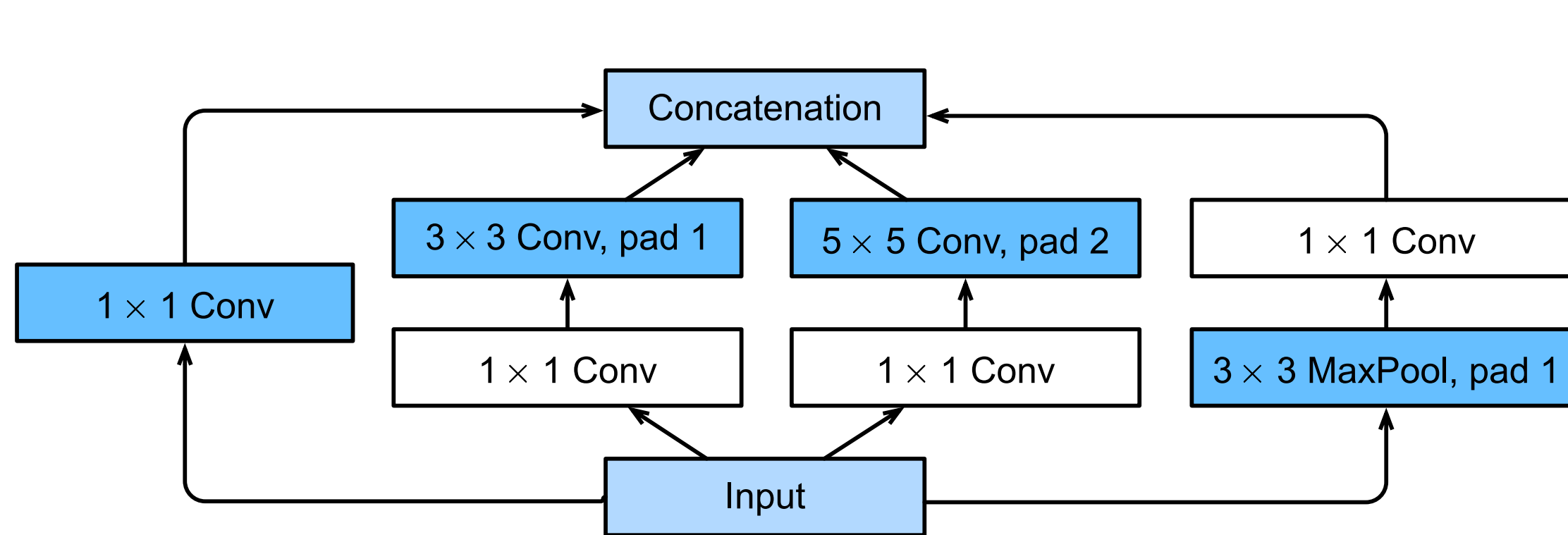
16-19 réteg – generatív hálóknban ma is használatos!

Konvolúciós Neurális Háló

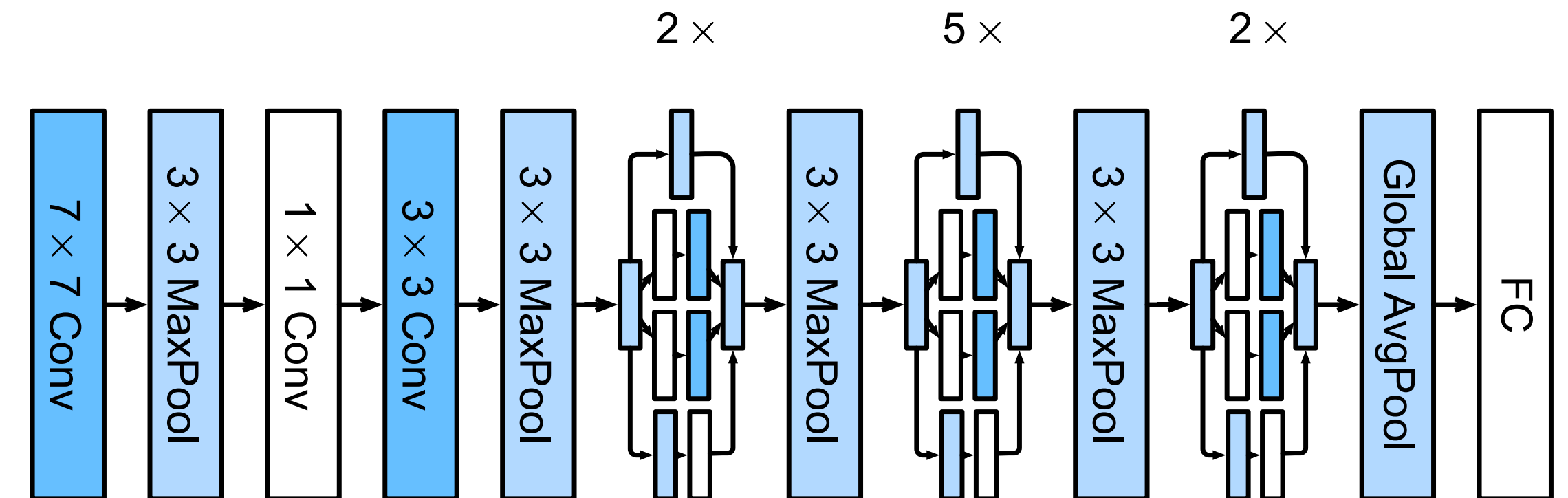
Architektúrák – Inception / GoogLeNet [2014]

Going deeper with convolutions

Christian Szegedy Google Inc.	Wei Liu University of North Carolina, Chapel Hill	Yangqing Jia Google Inc.	
Pierre Sermanet Google Inc.	Scott Reed University of Michigan	Dragomir Anguelov Google Inc.	Dimitru Erhan Google Inc.
Vincent Vanhoucke Google Inc.	Andrew Rabinovich Google Inc.		



Inception blokk



GoogLeNet

22 réteg!

Konvolúciós Neurális Háló

Architektúrák – ResNet [2015]

Deep Residual Learning for Image Recognition

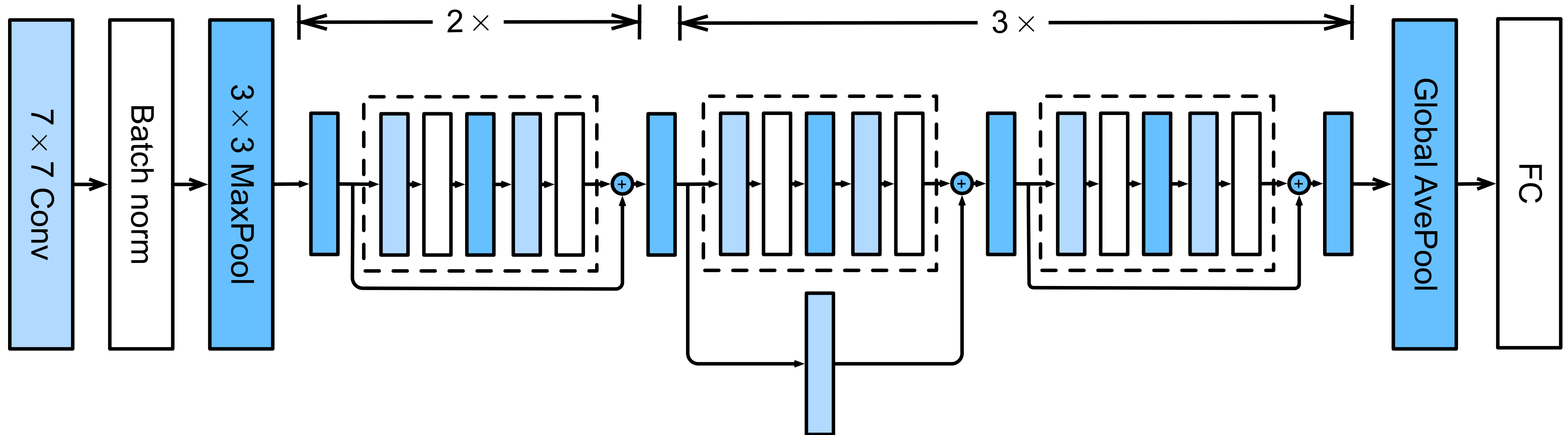
Kaiming He

Xiangyu Zhang

Shaoqing Ren

Jian Sun

Microsoft Research



ResNet-18

Konvolúciós Neurális Háló

Architektúrák – ResNet [2015]

Deep Residual Learning for Image Recognition

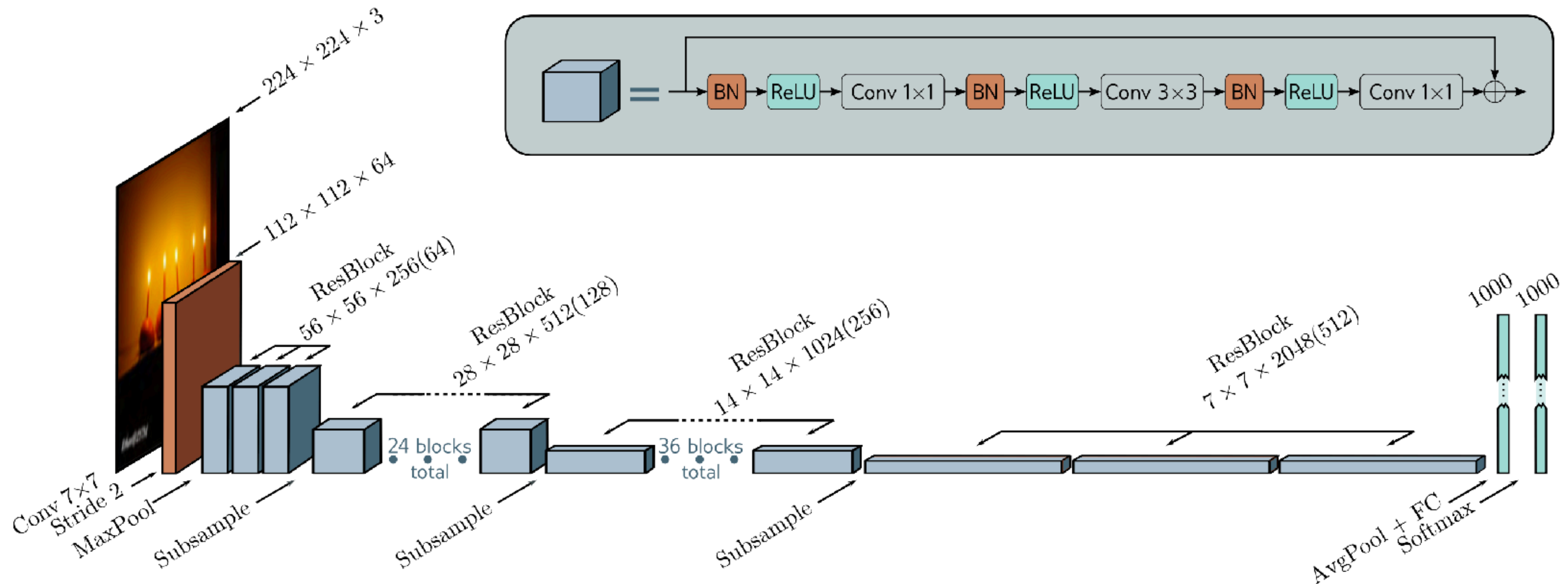
Kaiming He

Xiangyu Zhang

Shaoqing Ren

Jian Sun

Microsoft Research



Reziduális blokkokkal akár 200+ rétegű háló is tanítható!

Konvolúciós Neurális Háló

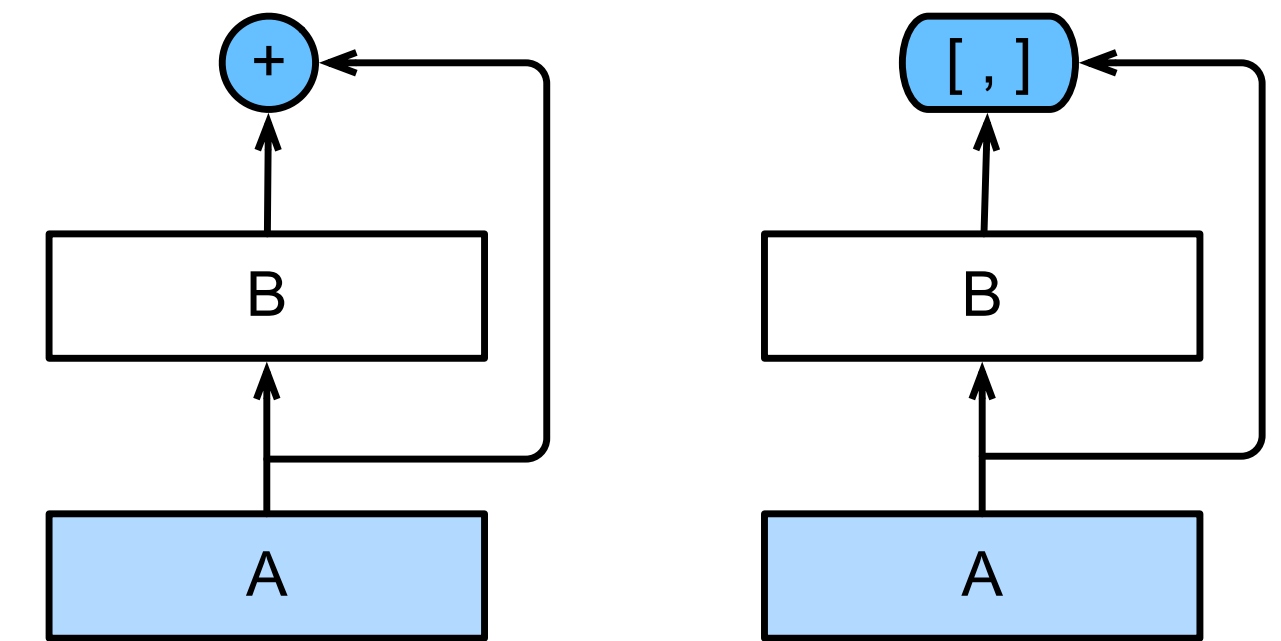
Architektúrák – DenseNet [2017]

Densely Connected Convolutional Networks

Gao Huang*
Cornell University

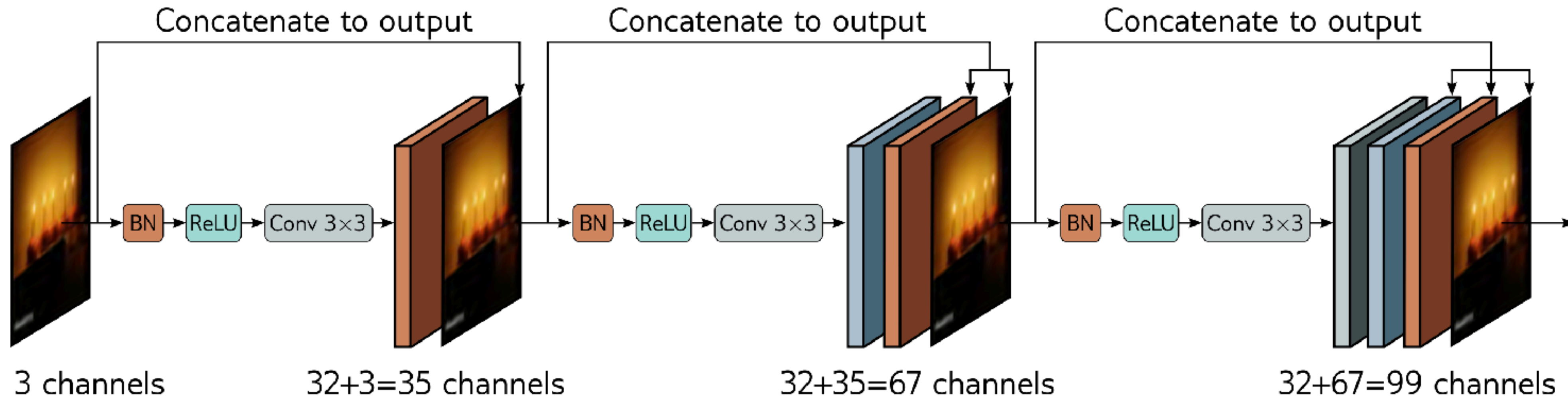
Zhuang Liu*
Tsinghua University

Laurens van der Maaten
Facebook AI Research



Residuális blokk:
összeadás

Dense blokk:
konkatenálás

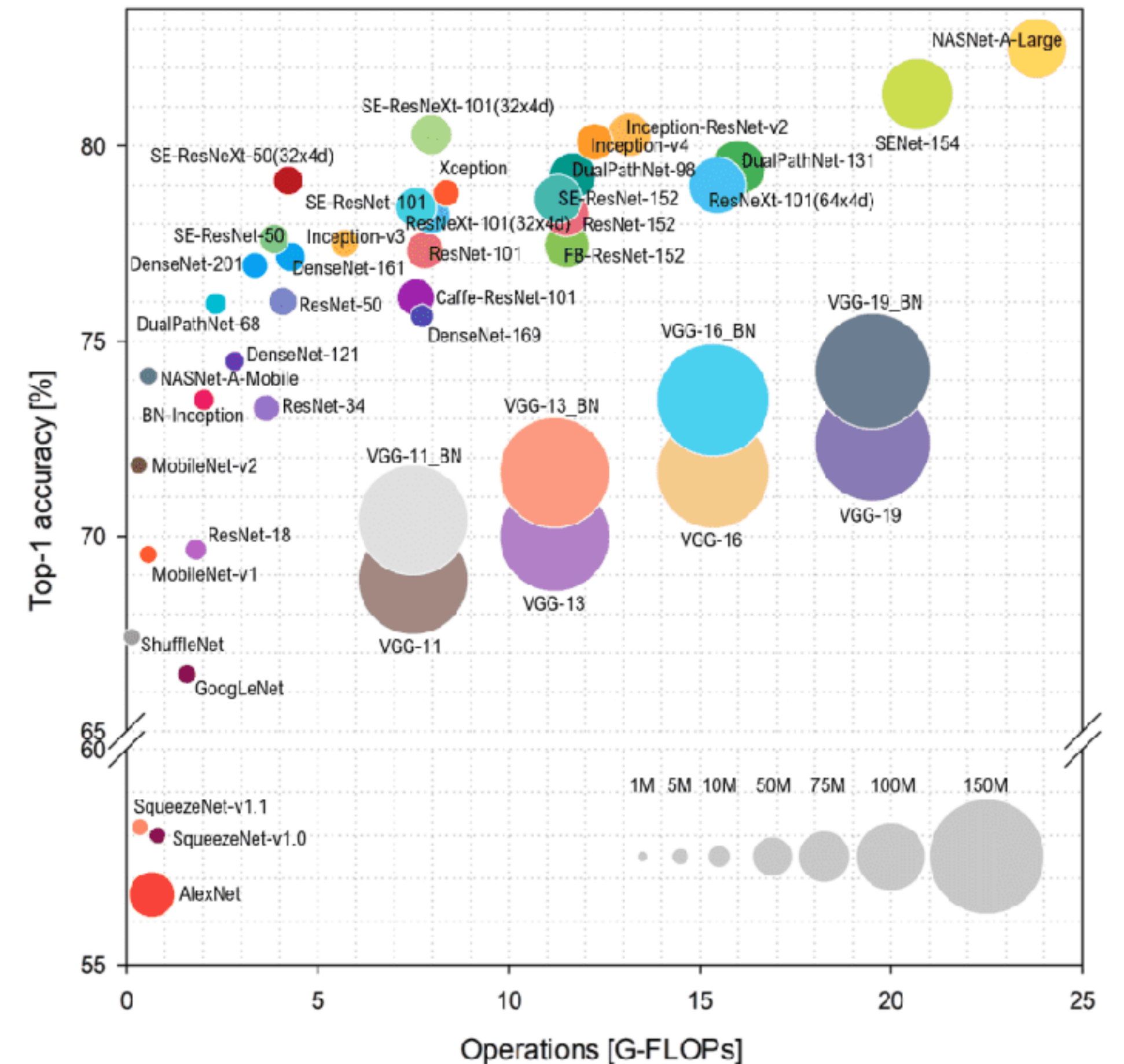


Akár 200+ rétegű háló is tanítható!

Konvolúciós Neurális Háló

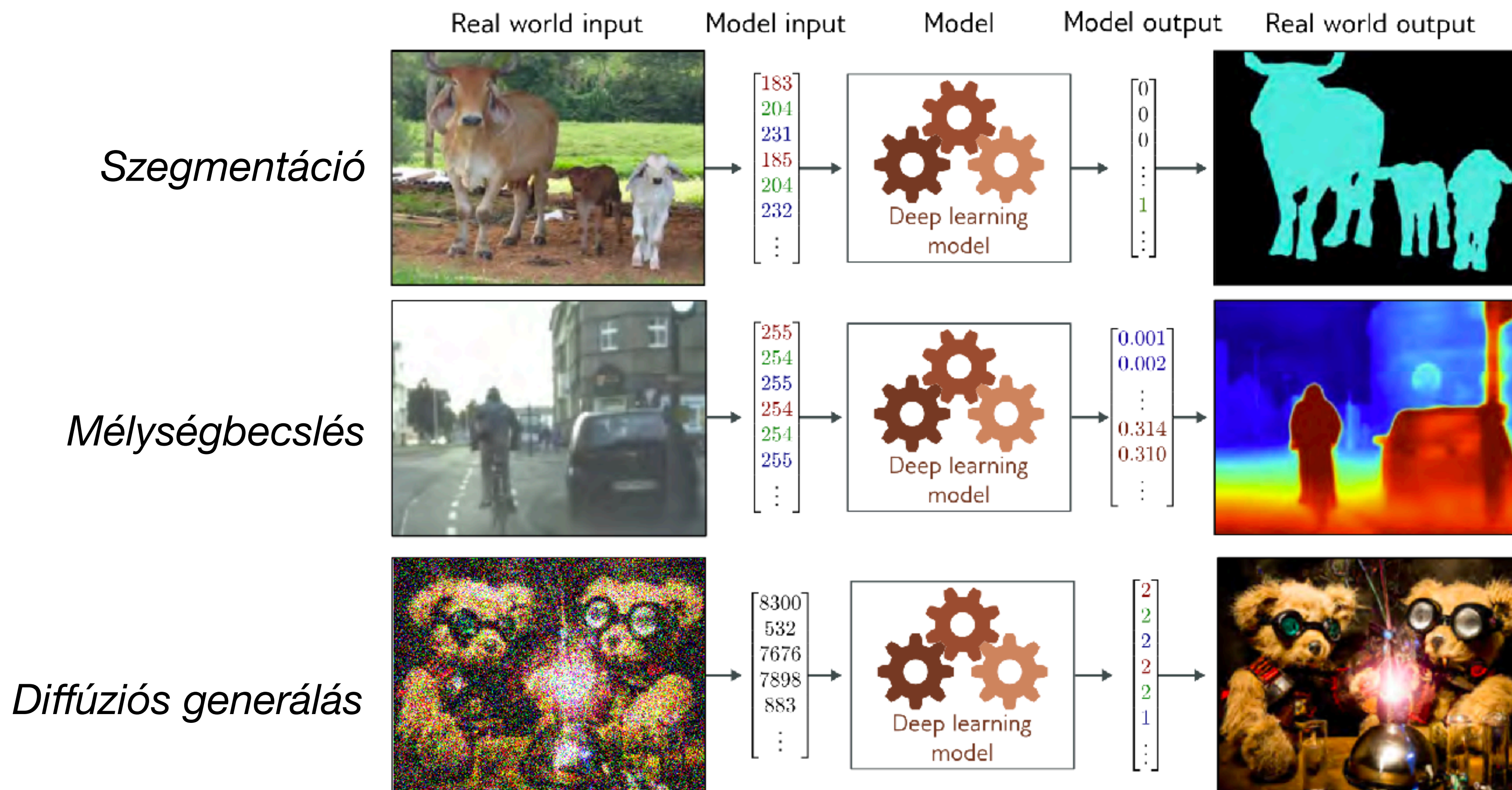
Architektúrák fejlődése

Model	No. of Layer	Parameters (Million)	Size
AlexNet	8	60	-
VGGNet-16	23	138	528 MB
VGGNet-19	26	143	549 MB
Inception-V1	27	7	-
Inception-V3	42	27	93 MB
ResNet-152	152	50	132 MB
ResNet-101	101	44	171 MB
InceptionResNetV2	572	55	215 MB
MobileNet-V1	28	4.2	16 MB
MobileNet-V2	28	3.37	14 MB
EfficientNet B0	-	5	-



Konvolúciós Neurális Háló

Architektúrák – Képből kép?

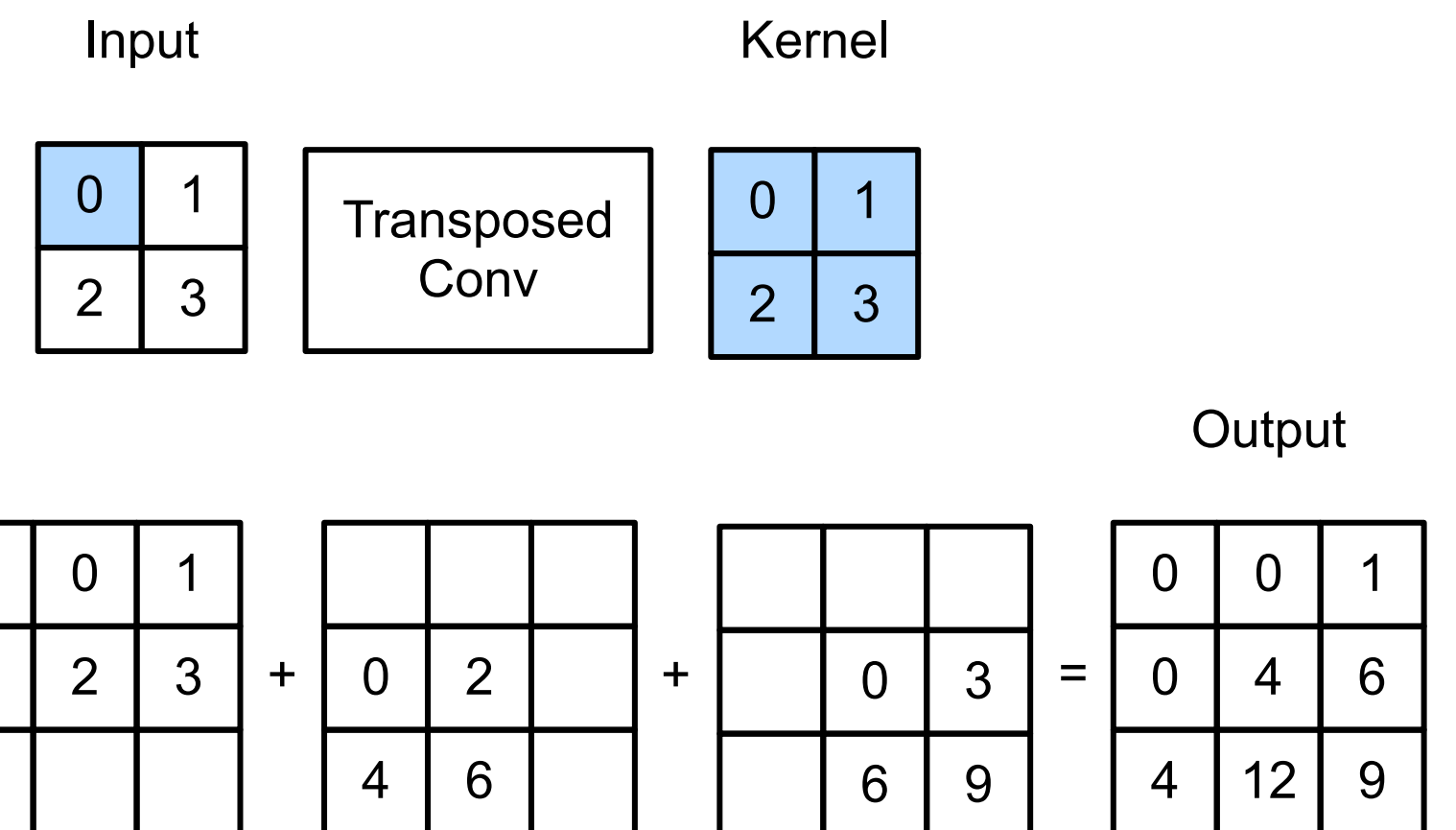
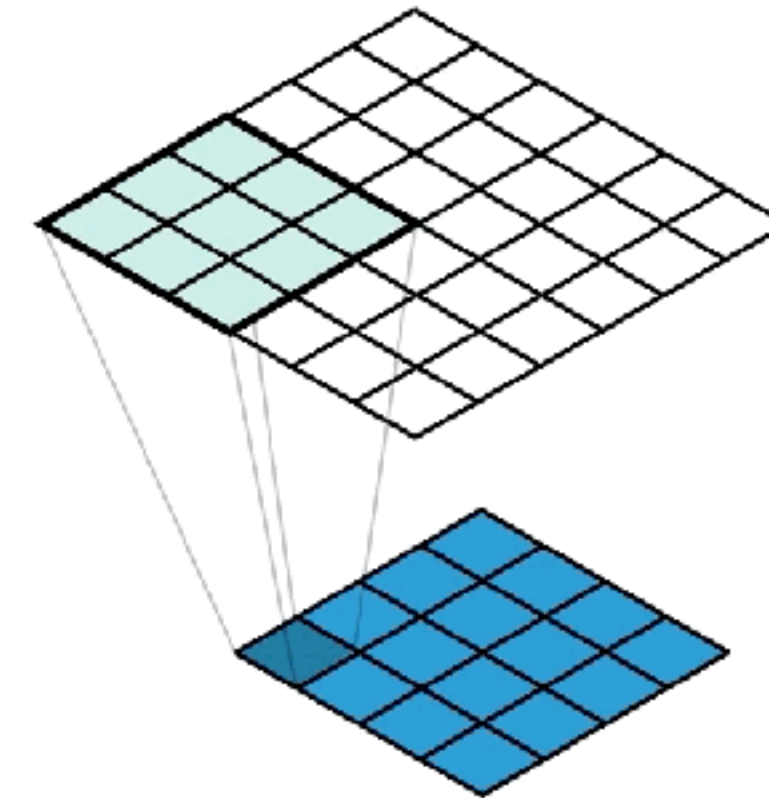


Konvolúciós Neurális Háló

Transzponált konvolúció

- A konvolúciós rétegek általában leskálázzák a bemenetet...
- Mi a felskálázás *tanítható*, *konvolúciós* megfelelője?
- Transzponált konvolúció!
 - Konvolúció mint mátrixszorzás:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \iff \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 4 & 5 \\ 2 & 3 & 5 & 6 \\ 4 & 5 & 7 & 8 \\ 5 & 6 & 8 & 9 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix}$$



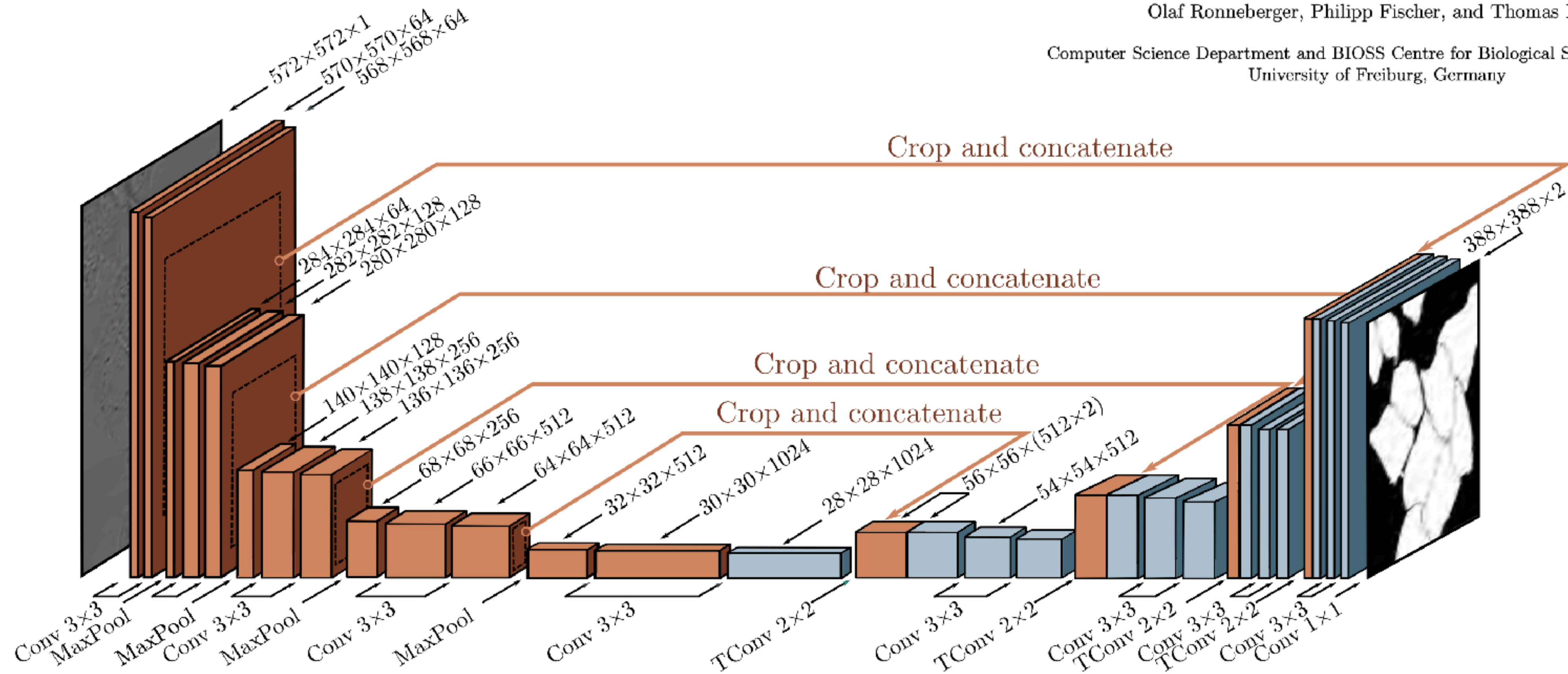
Konvolúciós Neurális Háló

Architektúrák – U-Net

U-Net: Convolutional Networks for Biomedical Image Segmentation

Olaf Ronneberger, Philipp Fischer, and Thomas Brox

Computer Science Department and BIOS Centre for Biological Signalling Studies,
University of Freiburg, Germany



Következő előadás:

Automatikus differenciálás és mélytanulás a gyakorlatban

- PyTorch alapjai
- Jupyter notebookok
- Neurális hálózatok definíciója és tanítása

