

Gyires-Tóth Bálint

Deep Learning a gyakorlatban Python és LUA alapon Konvolúciós hálózatok III

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.

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

Milestone 2: November 22. 23:59

Alaptanítás, ahol a tanítást végző kód és a kiértékelés első változata már rendben lefut, de még nem kötelező, hogy használható pontosságot adjon.

Megajánlott jegyért a nyelv: ANGOL

Kis házi feladat

1. Eredmények online.
2. Javítás alatt.
3. Várjuk.

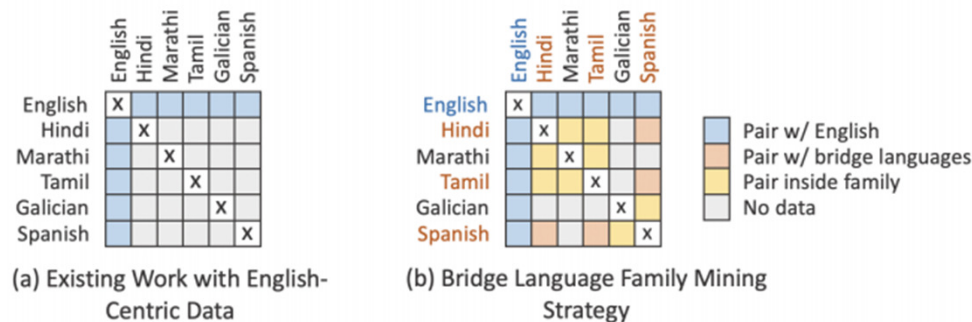


Deep Learning Híradó

Hírek az elmúlt 168 órából

M2M – Many-to-many Multilingual

- Nyelvcsoportok
- 7 különböző adatbázis
- Angoltól független fordítás
- 2200 nyelv irány



fr	French	Romance	Latin	si	Sinhala
gl	Galician	Romance	Latin	ur	Urdu
it	Italian	Romance	Latin	ta	Tamil
oc	Occitan	Romance	Latin	ceb	Cebuano
pt	Portuguese	Romance	Latin	ilo	Iloko
ro	Romanian	Romance	Latin	id	Indonesian
es	Spanish	Romance	Latin	jv	Javanese
be	Belarusian	Slavic	Cyrillic	mg	Malagasy
bs	Bosnian	Slavic	Latin	ms	Malay
bg	Bulgarian	Slavic	Cyrillic	ml	Malayalam
hr	Croatian	Slavic	Latin	su	Sundanese
cs	Czech	Slavic	Latin	tl	Tagalog
mk	Macedonian	Slavic	Cyrillic	my	Burmese
pl	Polish	Slavic	Latin	km	Central Khmer
ru	Russian	Slavic	Cyrillic	lo	Lao
sr	Serbian	Slavic	Cyrillic; Latin	th	Thai
sk	Slovak	Slavic	Latin	mn	Mongolian
sl	Slovenian	Slavic	Latin	ar	Arabic
uk	Ukrainian	Slavic	Cyrillic	he	Hebrew
et	Estonian	Uralic	Latin	ps	Pashto
fi	Finnish	Uralic	Latin	fa	Farsi
hu	Hungarian	Uralic	Latin	am	Amharic
lv	Latvian	Baltic	Latin	ff	Fulah
lt	Lithuanian	Baltic	Latin	ha	Hausa
sq	Albanian	Albanian	Latin	ig	Igbo
hy	Armenian	Armenian	Armenian	ln	Lingala
ka	Georgian	Kartvelian	Georgian	lg	Luganda
el	Greek	Hellenic	Greek	nso	Northern Sotho
br	Breton	Celtic	Latin	so	Somali
ga	Irish	Irish	Latin	sw	Swahili
gd	Scottish Gaelic	Celtic	Latin	ss	Swati

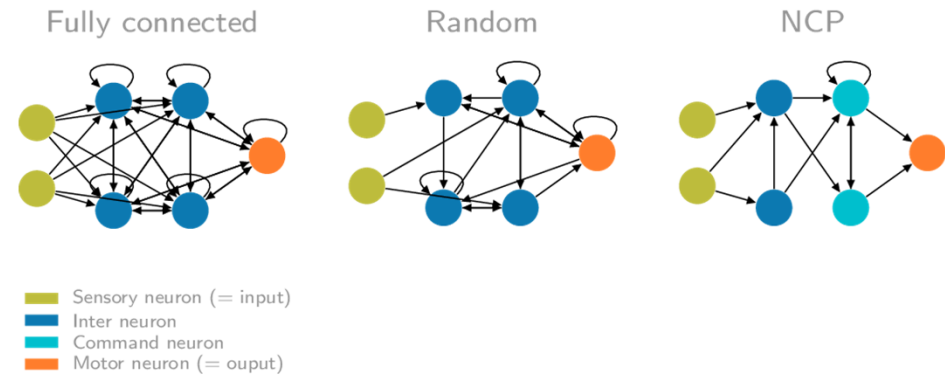
<https://about.fb.com/news/2020/10/first-multilingual-machine-translation-model/>

Brain-inspired neural network



Neural circuit policies enabling auditable autonomy

Mathias Lechner^{1,4}✉, Ramin Hasani^{2,3,4}✉, Alexander Amini³, Thomas A. Henzinger¹, Daniela Rus³ and Radu Grosu²



Source code: <https://github.com/mlech26l/keras-ncp>

Dátum: október 23.
Regisztráció:
http://bit.ly/nvidiadli_2020oct



DEEP
LEARNING
INSTITUTE

TEACHING YOU
TO SOLVE PROBLEMS
WITH DEEP LEARNING

Hiperparaméterek 2D konvolúció

INPUT: `{depth}@{input_x}x{input_y}`

FILTER / KERNEL: `{filter_x}x{filter_y}@{filter_depth}`

STRIDE: `{stride_x}x{stride_y}`

ZERO PADDING: `{zeropadding_x}x{zeropadding_y}`

DILATATION: `{dilatation_x}x{dilatation_y}`

OUTPUT SIZE:

```
{filter_depth}@  
{((input_x-filter_x+2*zeropadding_x)/stride_x)+1}x  
{((input_y-filter_y+2*zeropadding_y)/stride_y)+1}
```

Konvolúció

0	1	2	1	3	1	4
2	3	1	0	0	1	2
3	4	2	1	1	0	1
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.

Input

*

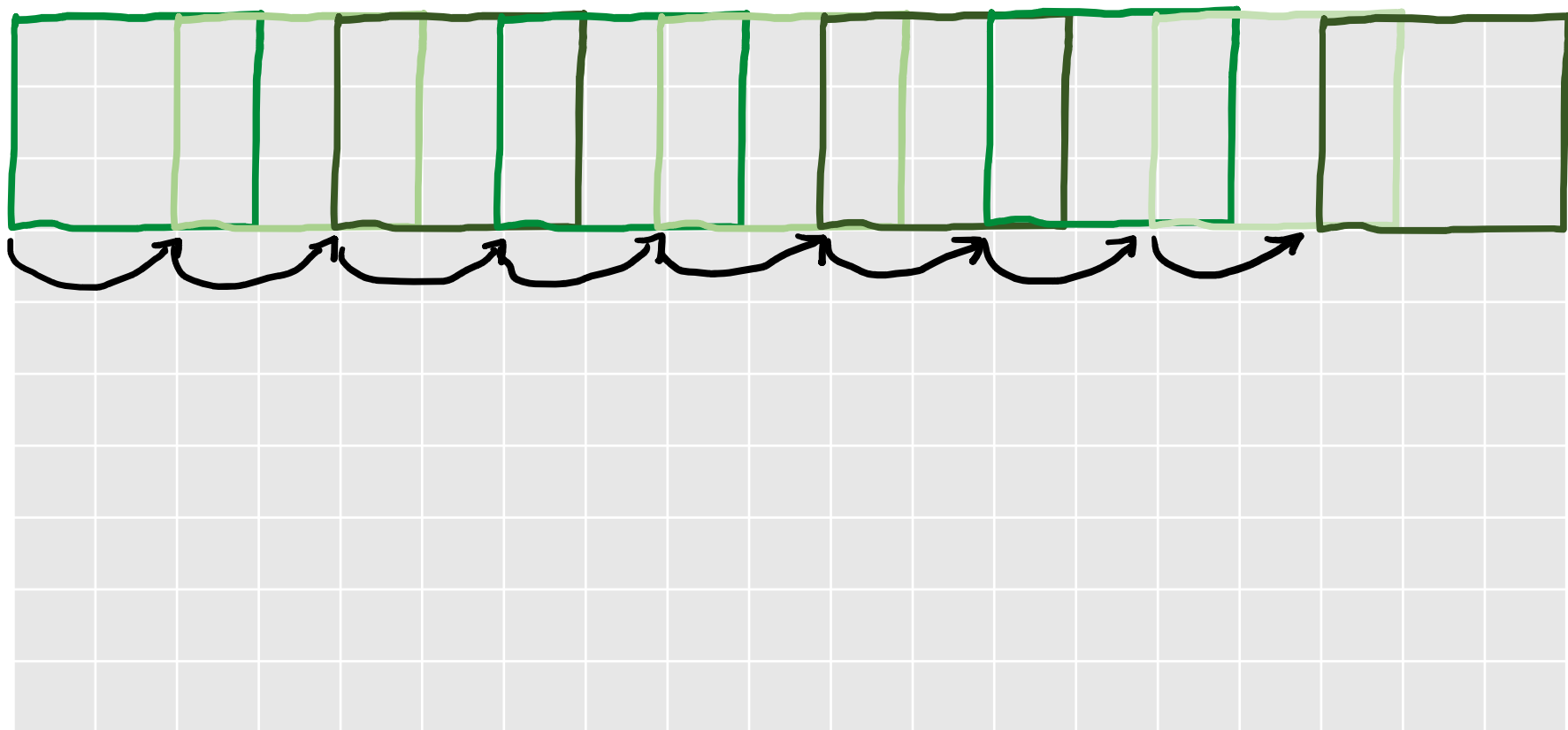
2	1
1	3

=

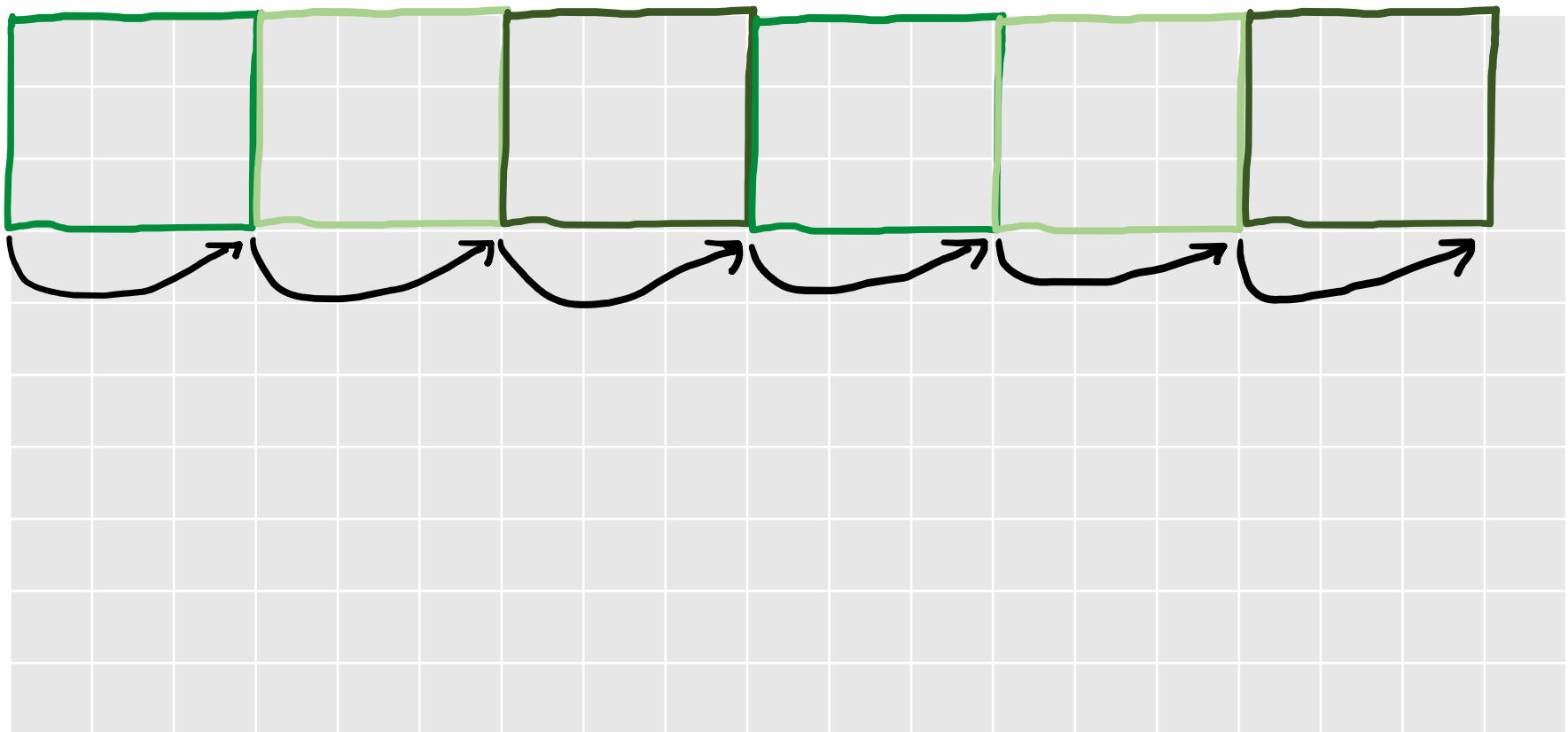
12	10	6	5	10	13
11	6	1	0	3	7
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

Output

Stride = 2



Stride = 3



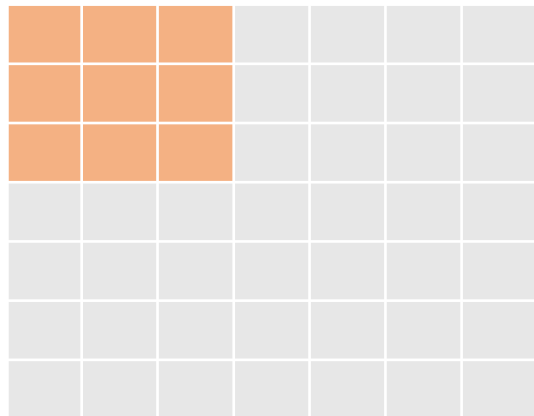
Zeropadding = 2x2

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0																0	0
0	0																0	0
0	0																0	0
0	0																0	0
0	0																0	0
0	0																0	0
0	0																0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

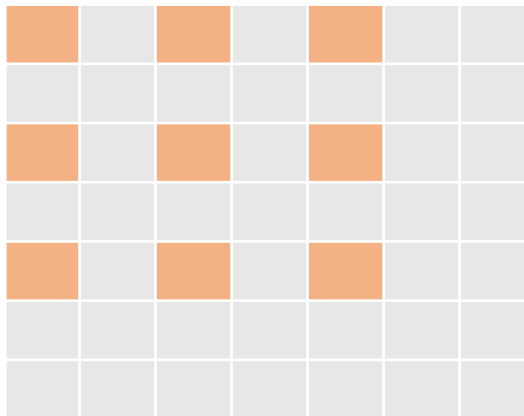
Zeropadding = 1x1

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0																		0
0																		0
0																		0
0																		0
0																		0
0																		0
0																		0
0																		0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

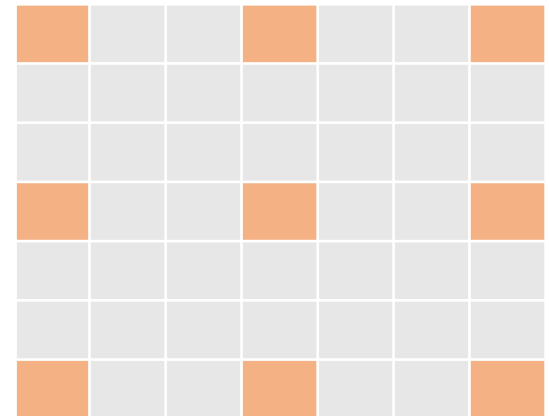
Dilation



Kernel: 3x3
Dilation: 1x1

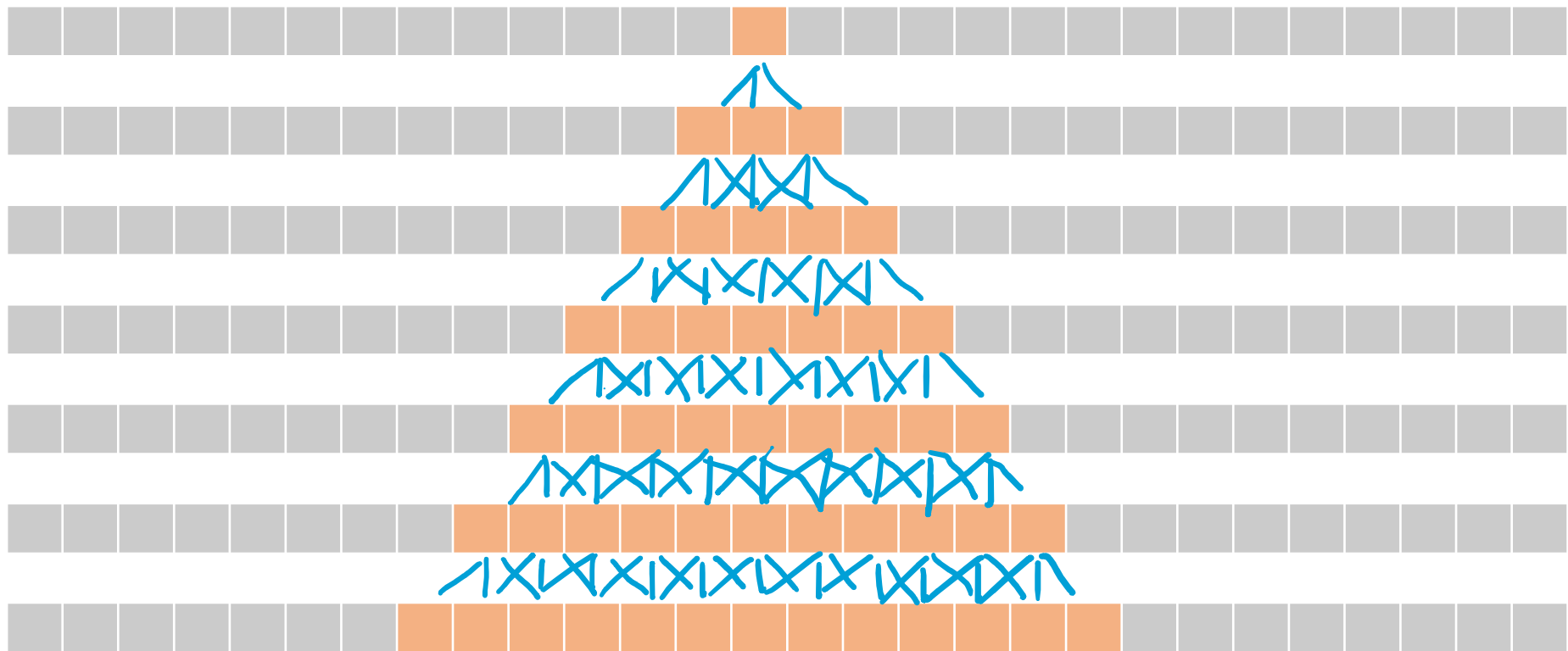


Kernel: 3x3
Dilation: 2x2

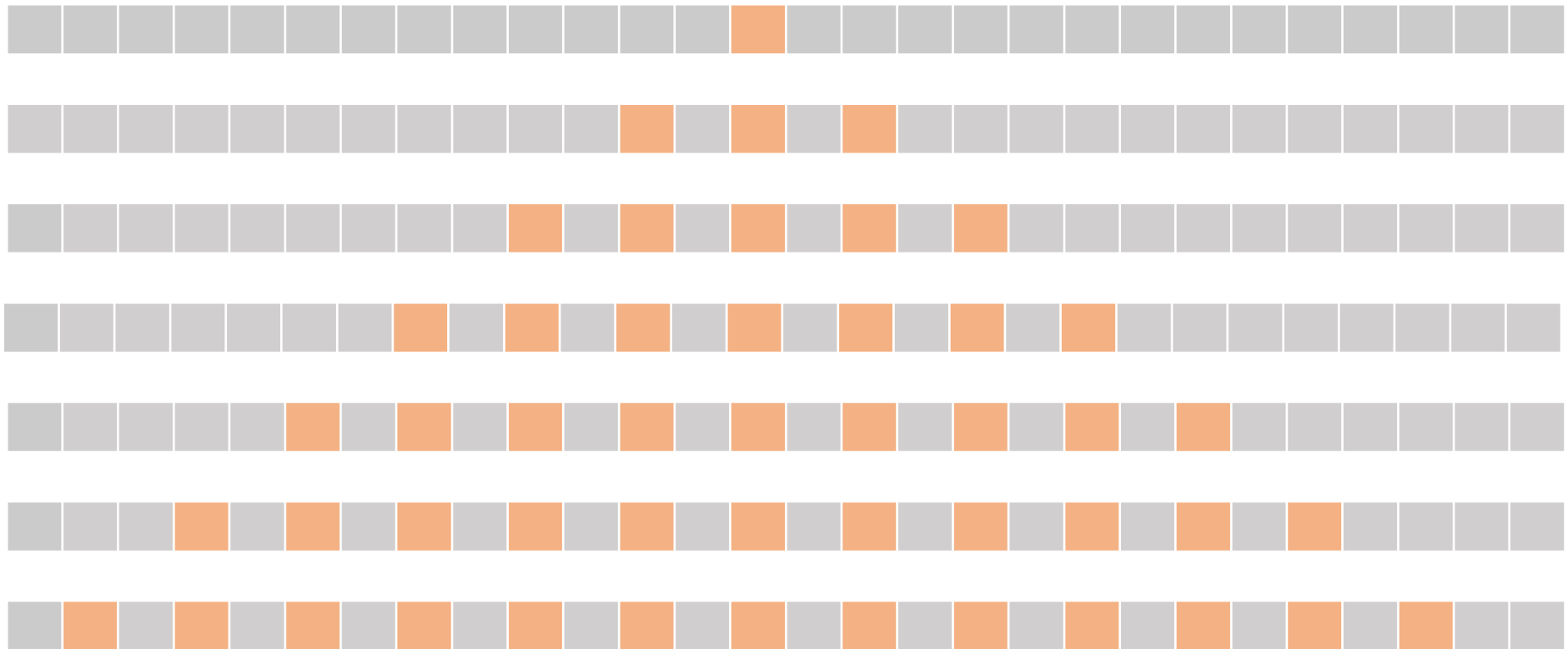


Kernel: 3x3
Dilation: 3x3

Több konvolúciós réteg

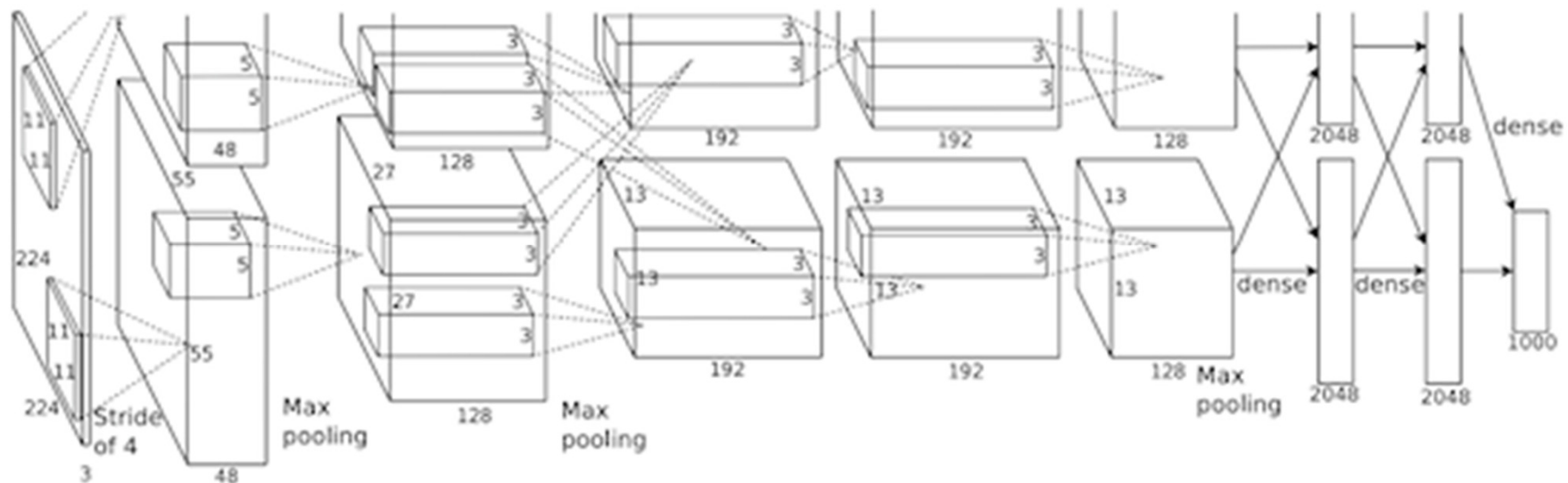


Több dilettáció konvolúciós réteg

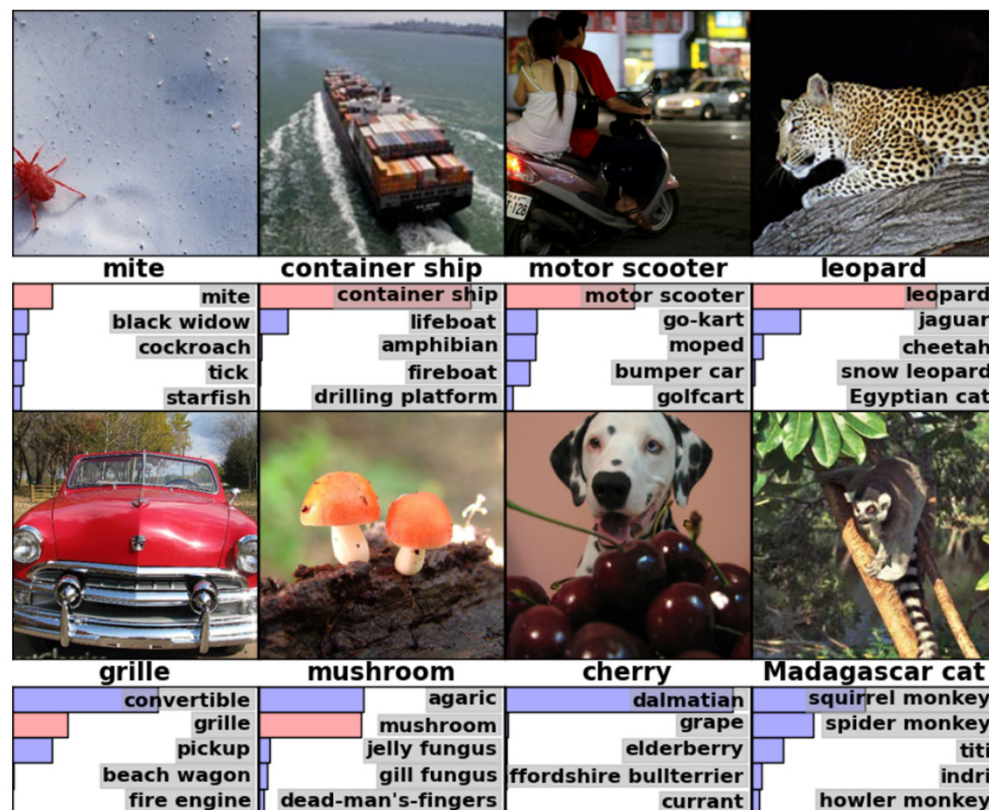


AlexNet

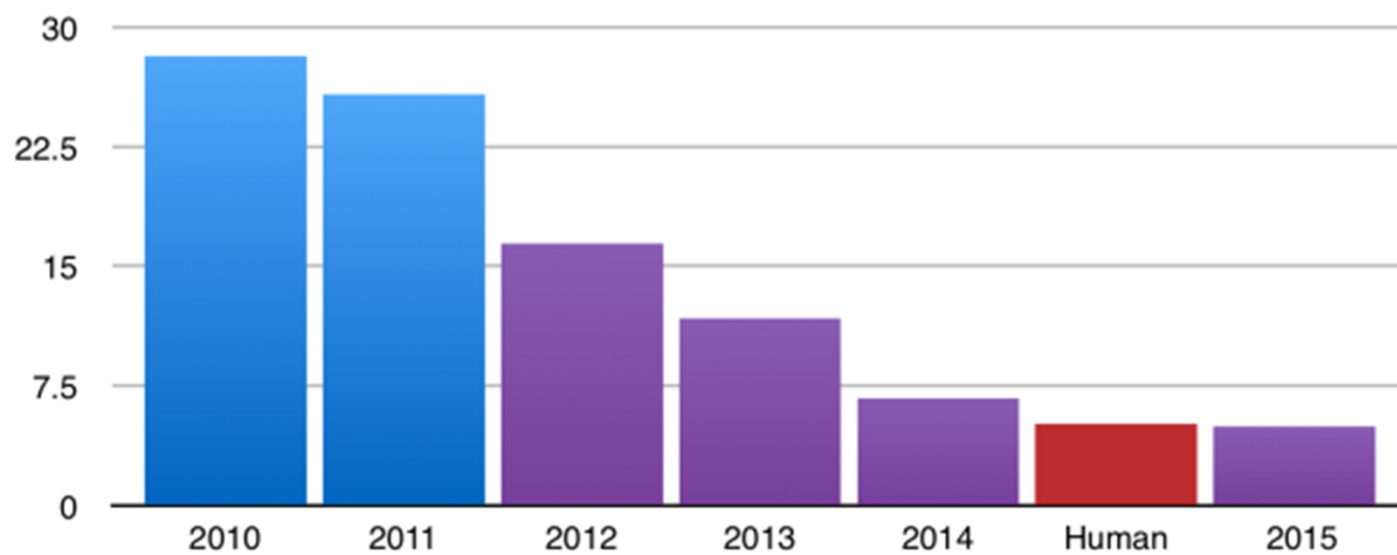
Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012).
Imagenet classification with deep convolutional neural
networks. In *Advances in neural information processing
systems* (pp. 1097-1105).



AlexNet 2012 - eredmények



ILSVRC top-5 error on ImageNet



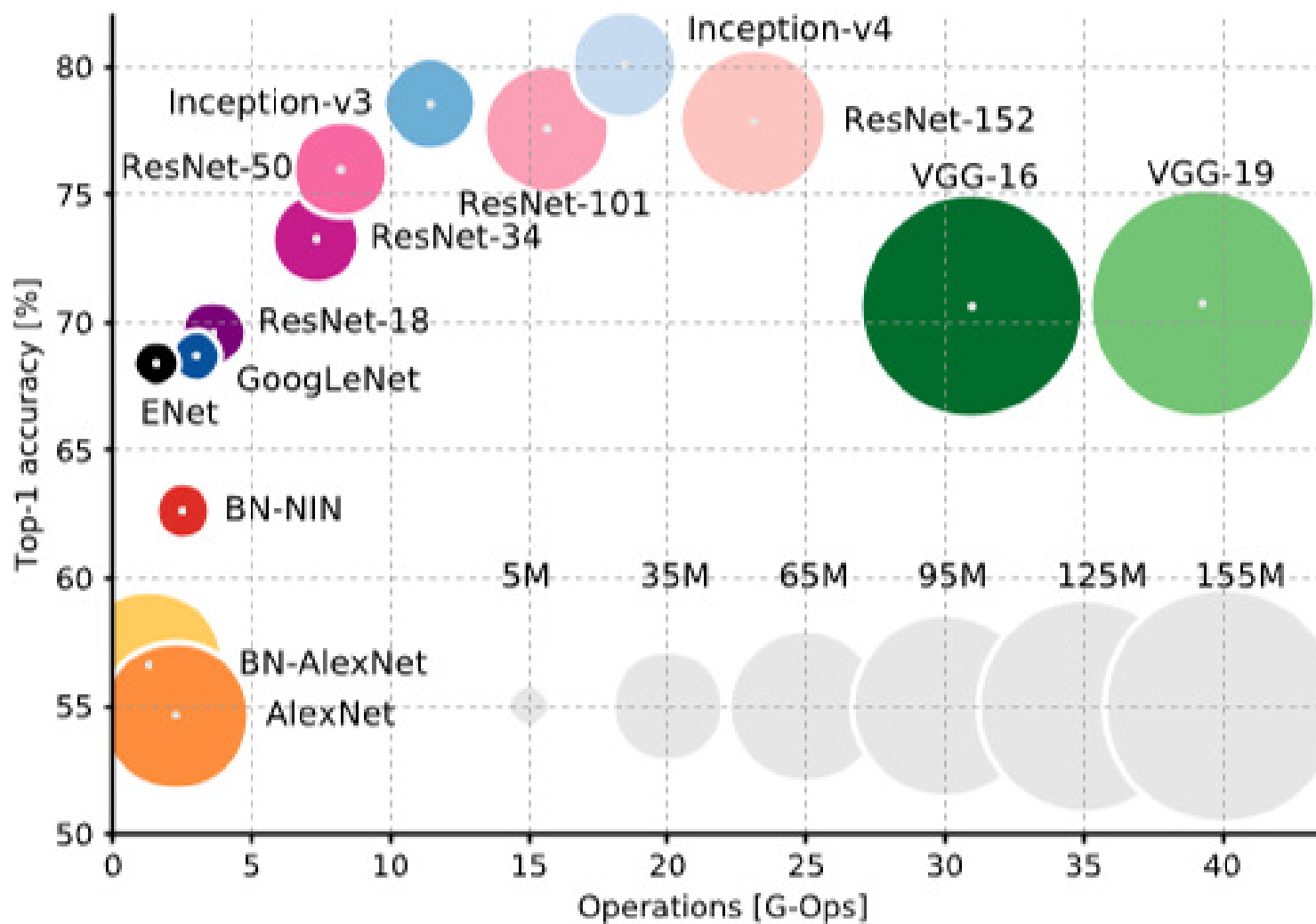
Népszerű hálók

- Overfeat
- VGG16, VGG19
- ResNet
- SqueezeNet
- Inception V3, V4
- Xception
- Stb.

Előtanított hálók:

- PyTorch: beépítve (`import torchvision.models as models`)
- Keras: beépítve (`from keras.applications....`)
- TensorFlow: beépítve (`import tensorflow.contrib.slim as slim`)

Népszerű hálók



Canziani, A., Paszke, A., & Culurciello, E. (2016). An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*.

Népszerű kép adatbázisok

- MNIST
- NIST
- CIFAR10/CIFAR100
- SLT-10
- SVHN
- IMAGENET (ILSVRC2012)
- PascalVOC
- Google Open Image Dataset
- Stb.

Alapfeladatok

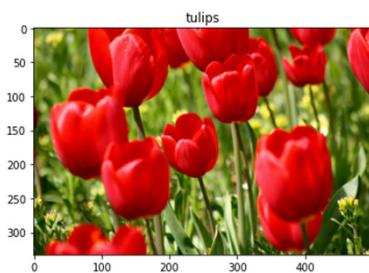
- Kép
 - Képfelismerés: egy vagy többosztályos
 - Objektum detekció és szegmentáció
 - Style transfer
 - Kép generálás, kép rekonstrukció
 - Kép feliratozás
 - Stb.
- Hang
 - Beszédfelismerés, beszélőfelismerés
 - Beszédszintézis
 - Spoofing felismerése
 - Stb.

Trükkök, módszerek

- Adatdúsítás (data augmentation)
- Transfer learning
- Rétegek elemzése

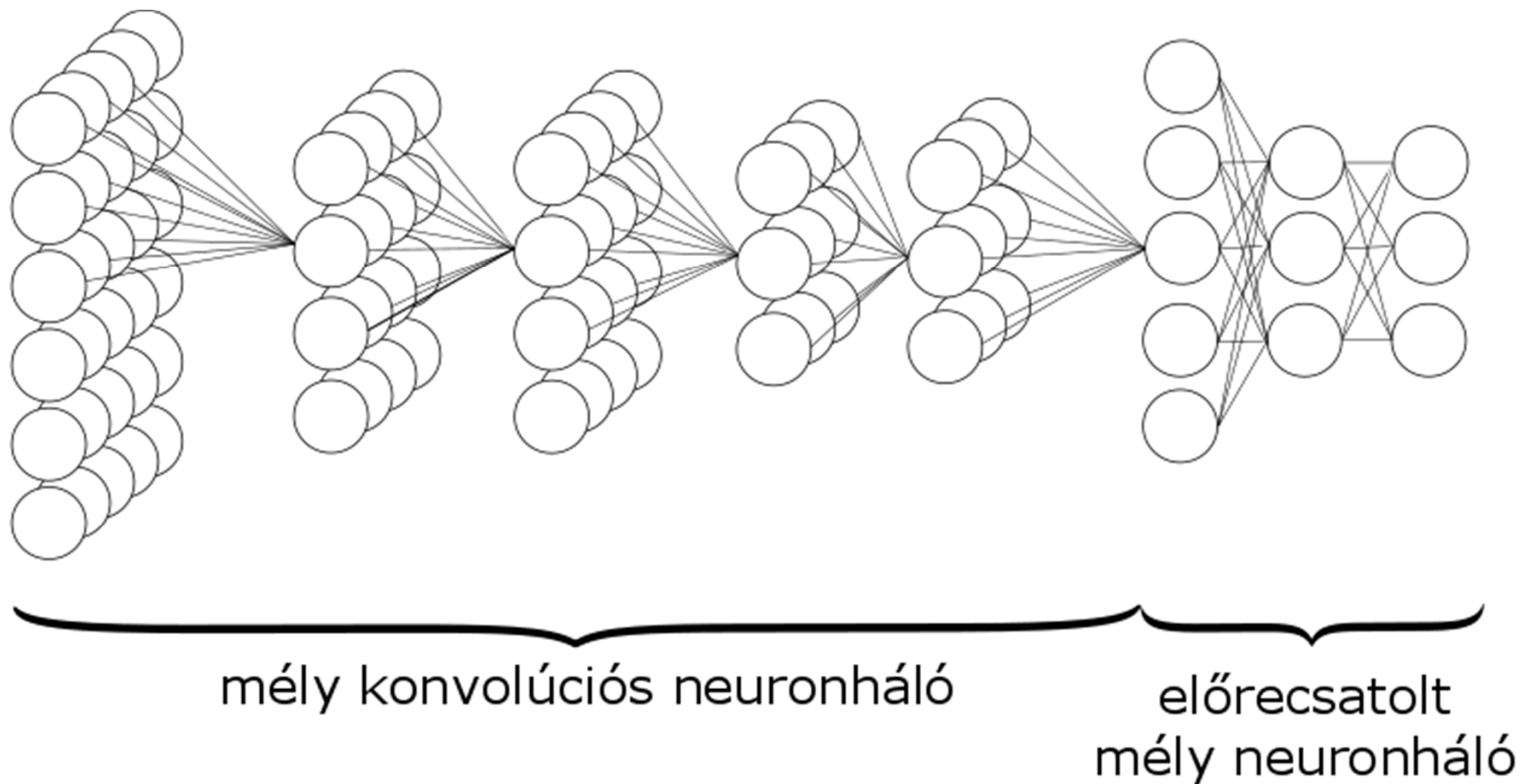
- Batch Normalization (gyorsabb konvergencia, nem szükséges DropOut)
- Dilated convolution (PixelCNN, WaveNet)
- Kb. 3x gyorsabb: Factorized Convolutional Network (~Separable Convolutions)

Adatdúsítás



Forrás: https://www.tensorflow.org/tutorials/images/data_augmentation

Transfer learning



Transfer learning

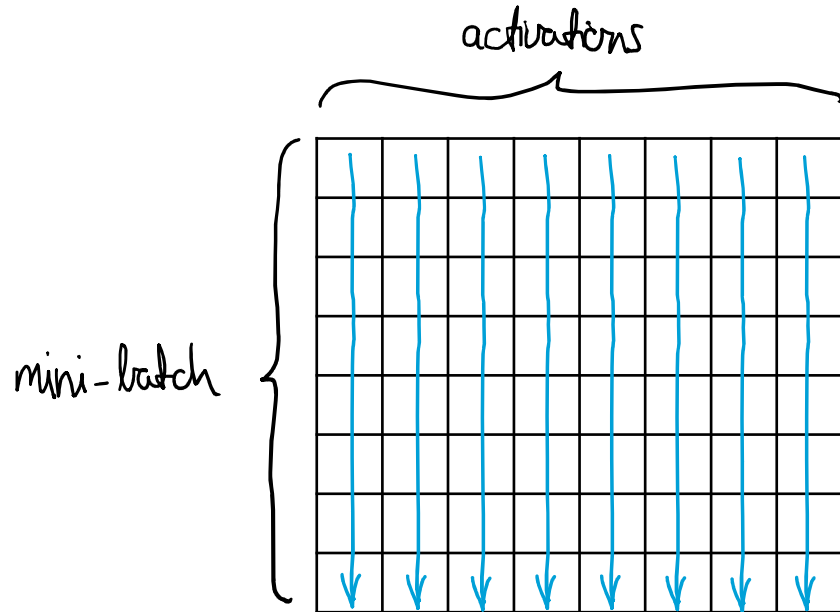
Egy komplett tanítás idő, hardver és adat igényes. Ezért gyakorlatban sokszor *előtanított* modelleket használunk.

Lehetőségek:

- Az előrecsatolt rétegeket kicseréljük és az előtanított hálót jellemző kinyerésre használjuk, fagyasztott súlyokkal.
- A CNN súlyait is újratanítjuk (elsősorban csak ha sok adat van).
 - Az összes súlyt, egységes vagy rétegenként megadott tanulási rátával (osztályozónak nagyobb, CNN-nek kisebb súlyokkal).
 - Csak a fentebbi rétegeket tanítjuk (alsó rétegek pl. éldetektorok).

Batch normalization

- Az aktivációk normalizálása mini-batch-enként
- Hatékony eljárás a dropout mellett.
- Sorrend
 - Batch Norm
 - ReLU/PReLU/LReLU
 - (DropOut)
- Nagyobb batch méretek szükségessé válnak



Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Konvolúciós rétegek elemzése

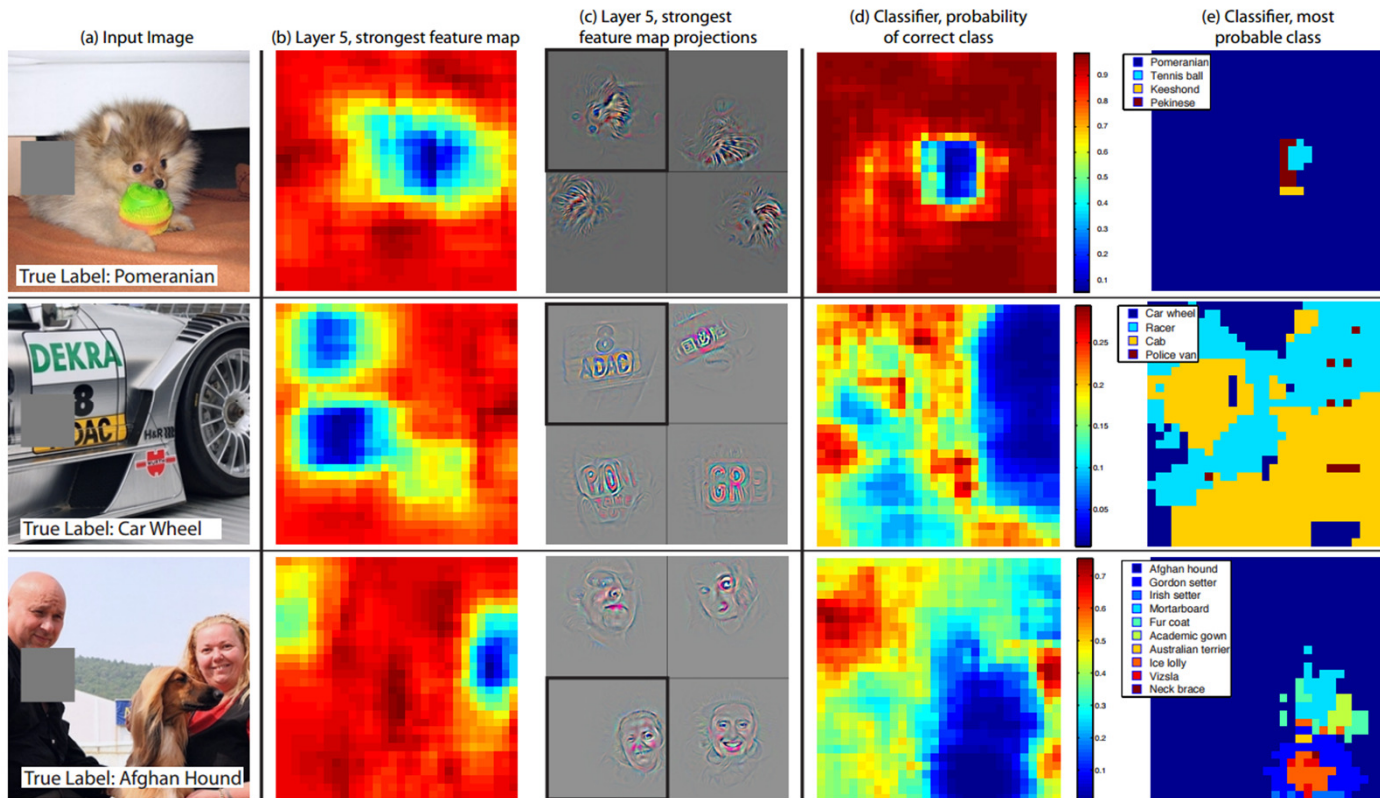
ZEILER, Matthew D.; FERGUS, Rob. Visualizing and understanding convolutional networks. In: *European Conference on Computer Vision*. Springer International Publishing, 2014. p. 818-833.

<https://arxiv.org/abs/1311.2901>

Példák:

- Rétegek aktivációs térképe
- Súlymátrixok
- Területek kitakarása
- Rétegaktiváció maximalizálása

Rétegaktiváció maximalizálása



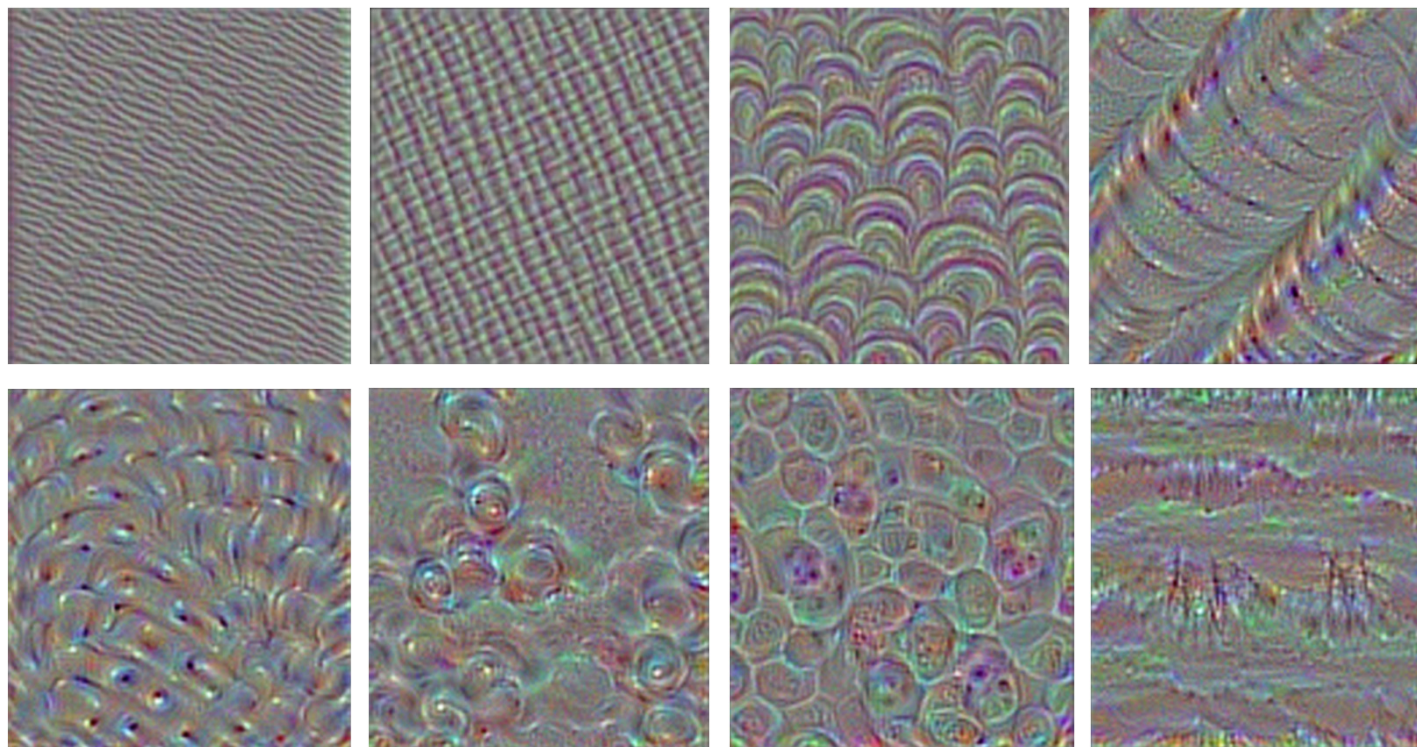
Forrás: <https://arxiv.org/pdf/1311.2901v3.pdf>

Rétegaktiváció maximalizálása

A bemenet „optimalizálása” az adott réteg maximalizálása céljából.

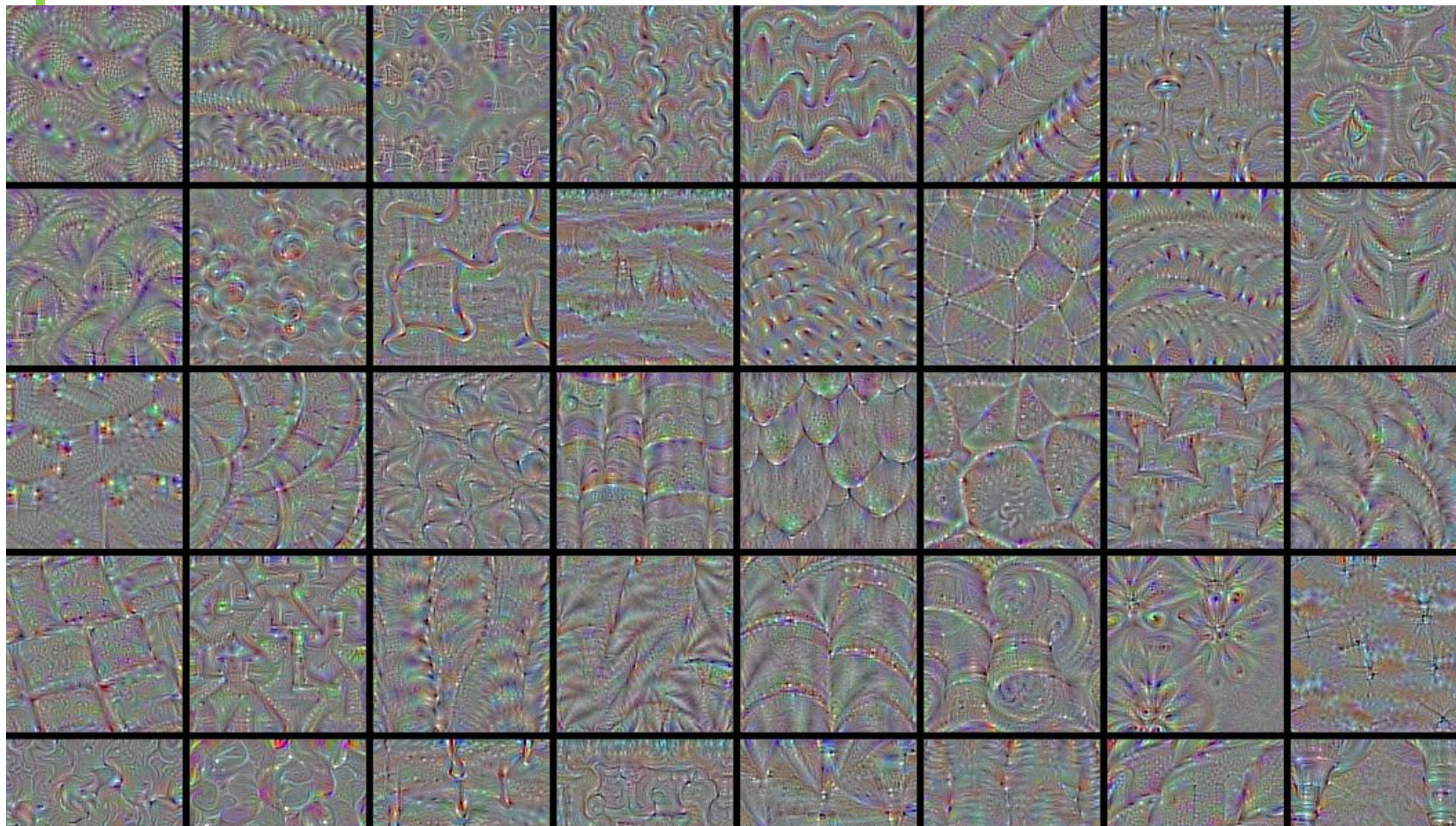
A rétegaktiváció összegének vagy normájának deriváltja a bemenet függvényében. A bemenethez a gradiensek hozzáadása.

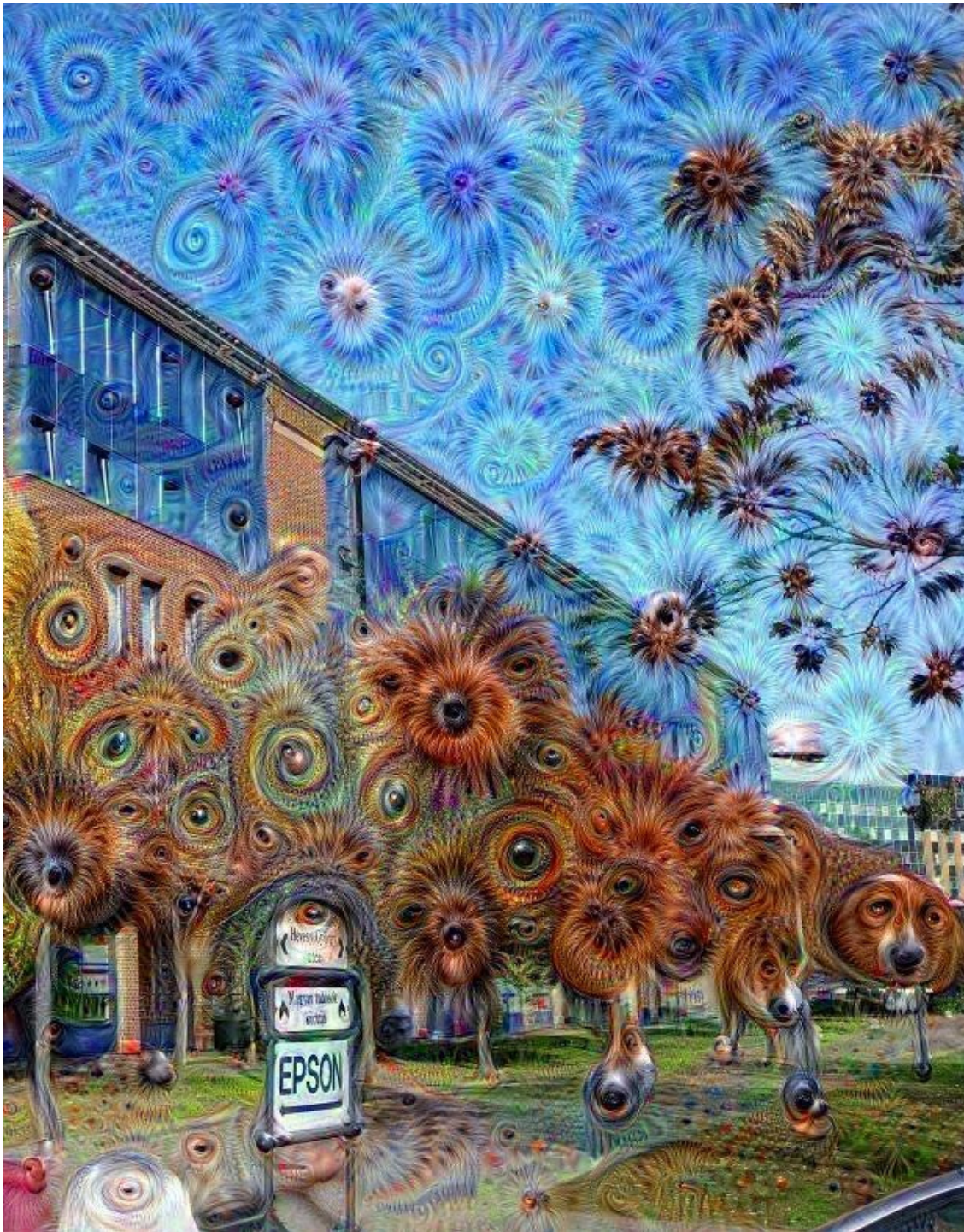
Rétegaktiváció maximalizálása



Forrás: <https://blog.keras.io/how-convolutional-neural-networks-see-the-world.html>

Rétegaktiváció maximalizálása





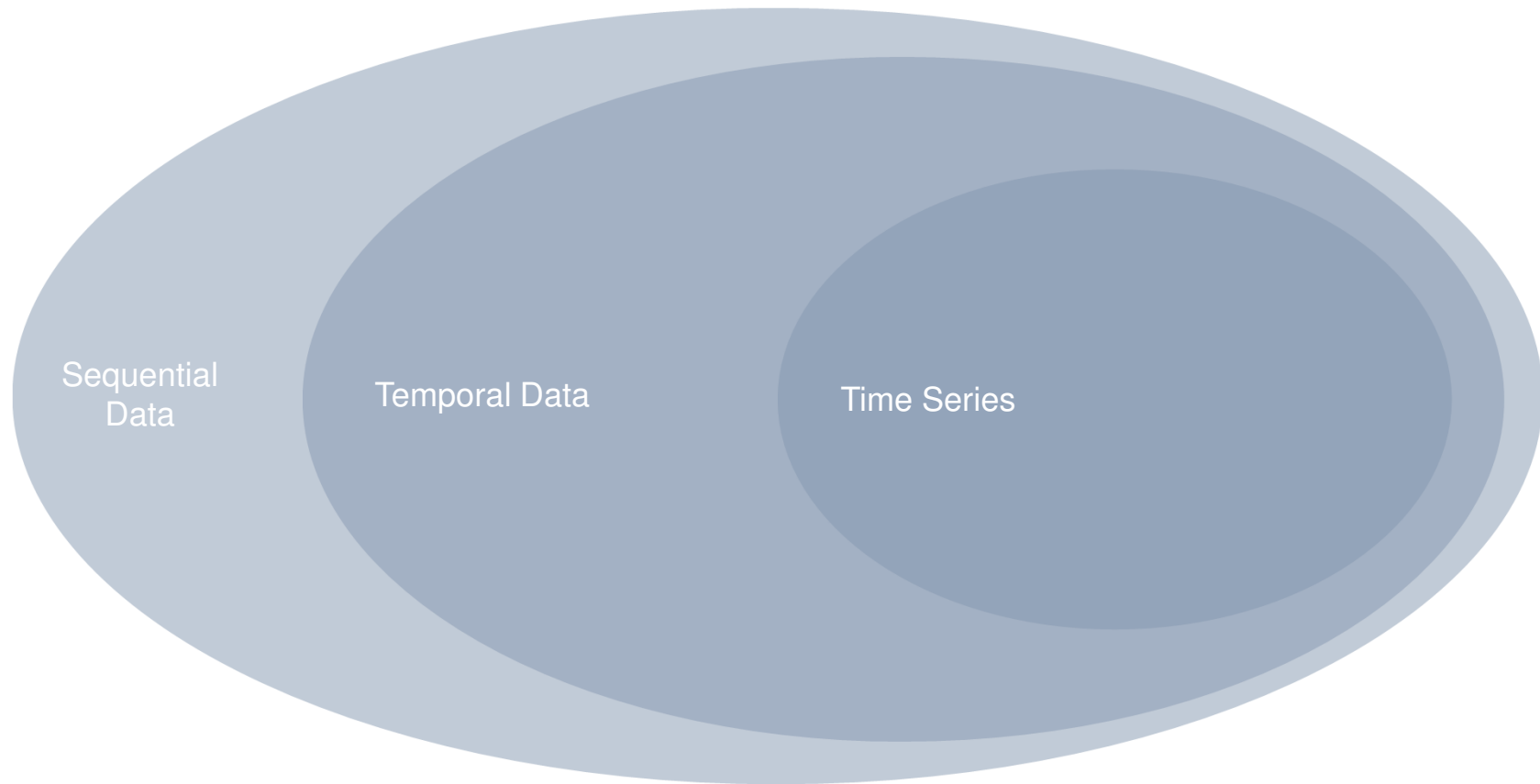
Deep Dream

Input is an image

Maximizing the
activations of several
layers.

Szekvenciális adatok modellezése

Szekvenciális adattípusok

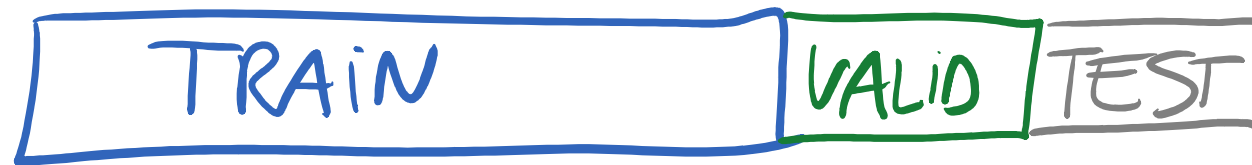


Elsődleges kihívások

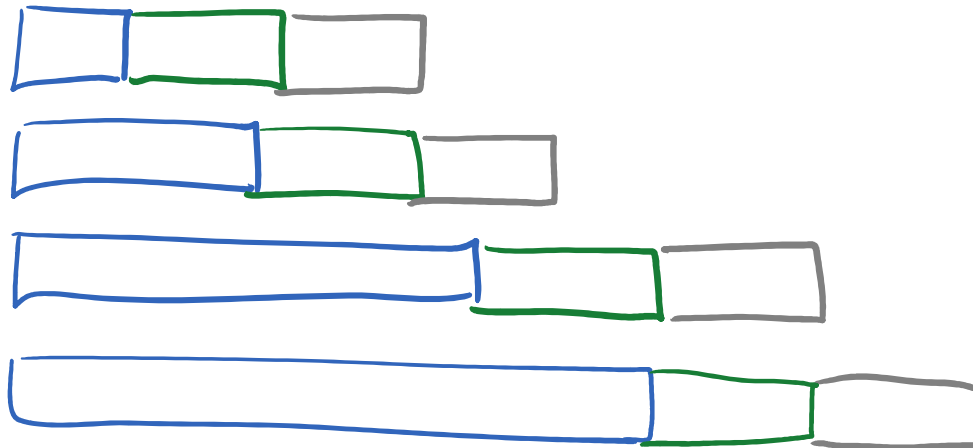
- A receptív mező növelése
- Sokdimenziós adat modellezése
- Több-skálás adatok modellezése

Adatok felosztása

Train – validation – test sets

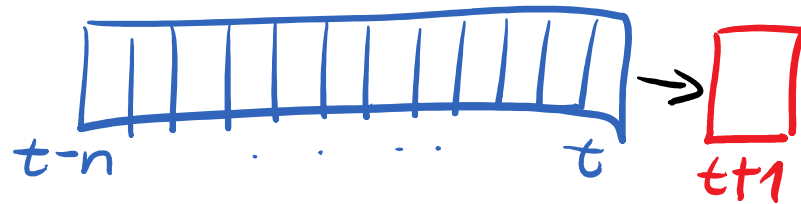


Keresztvalidáció

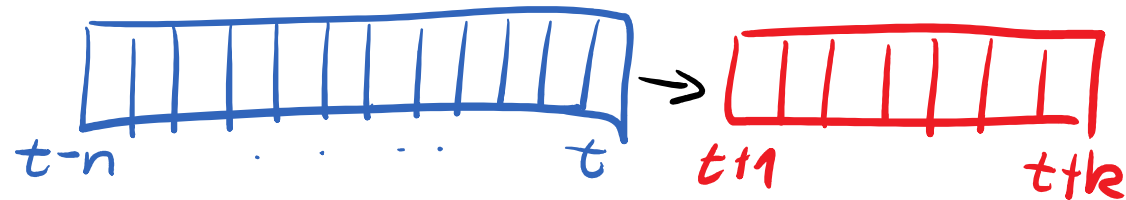


Célváltozó

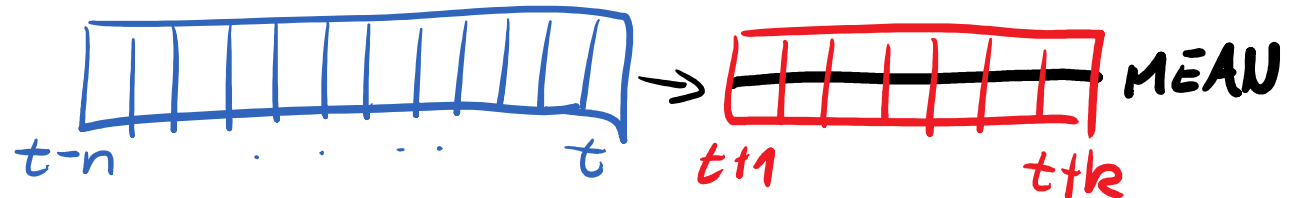
Következő érték



Következő értékek

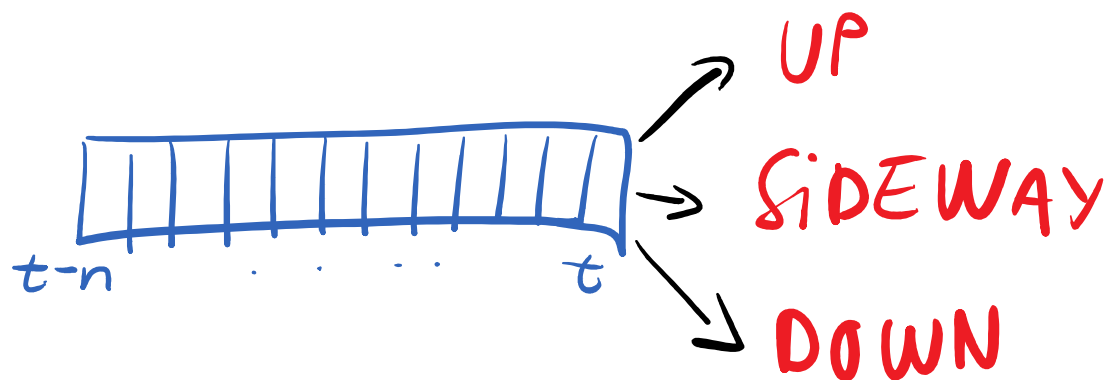


Következő értékek átlaga

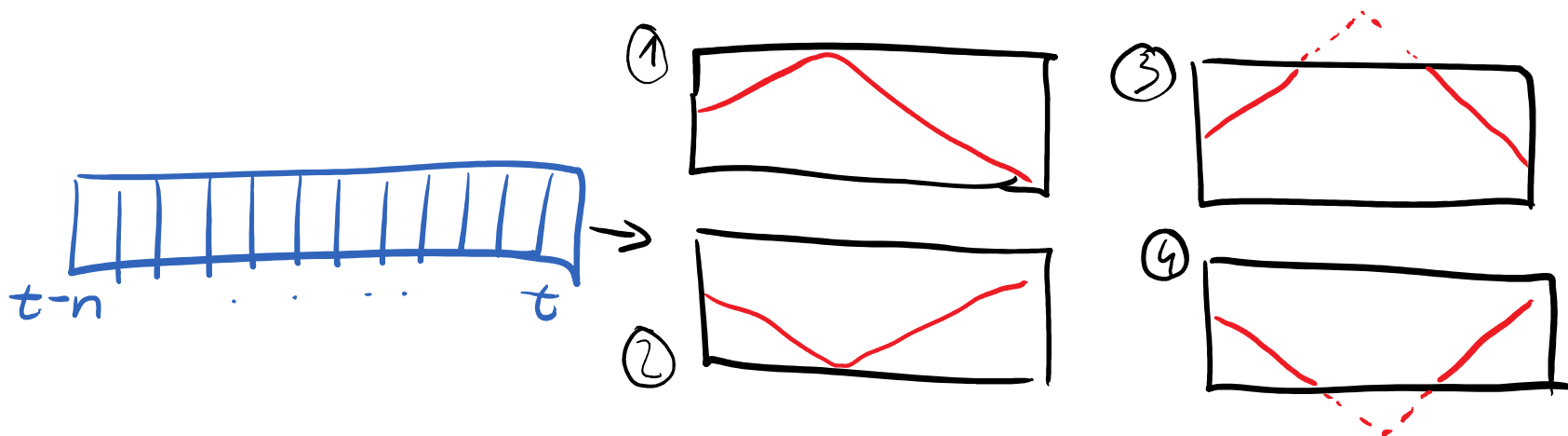


Célváltozó

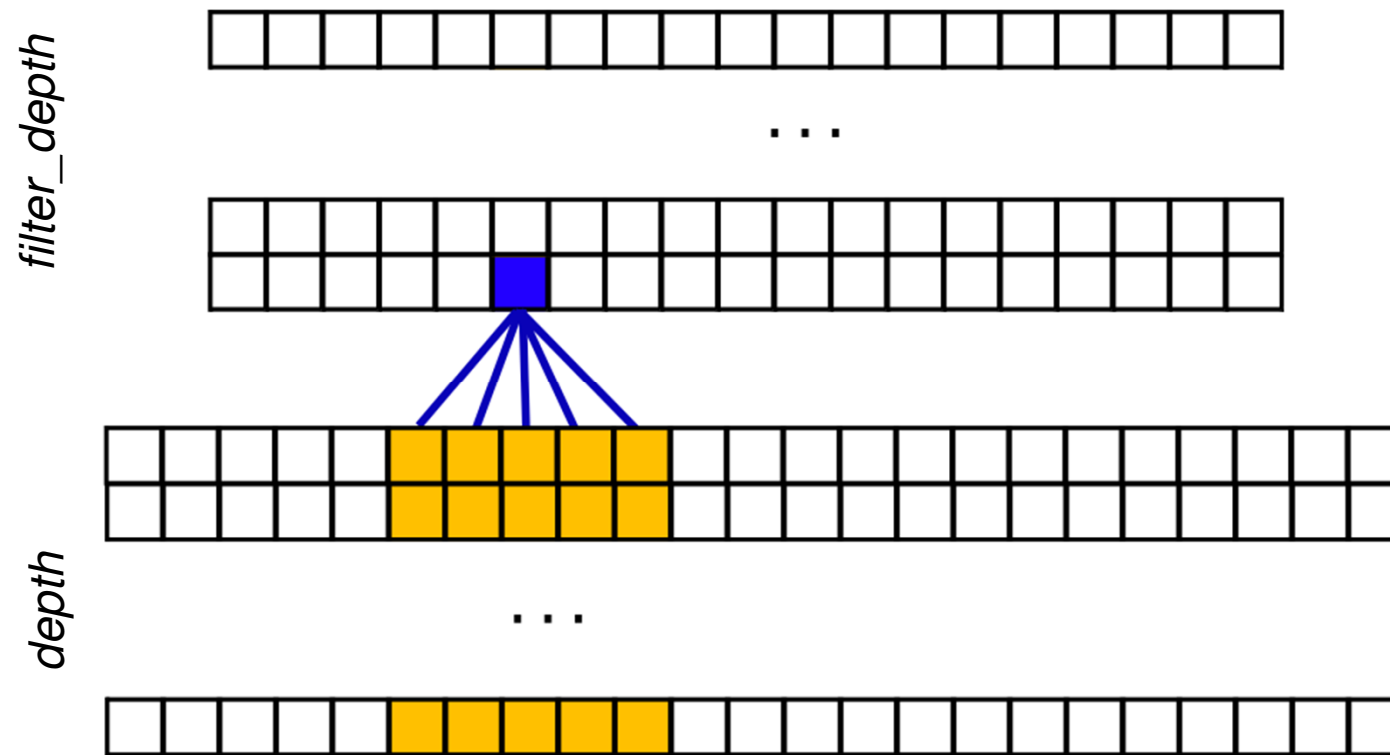
Trend



Mintázatok



1 D konvolúció



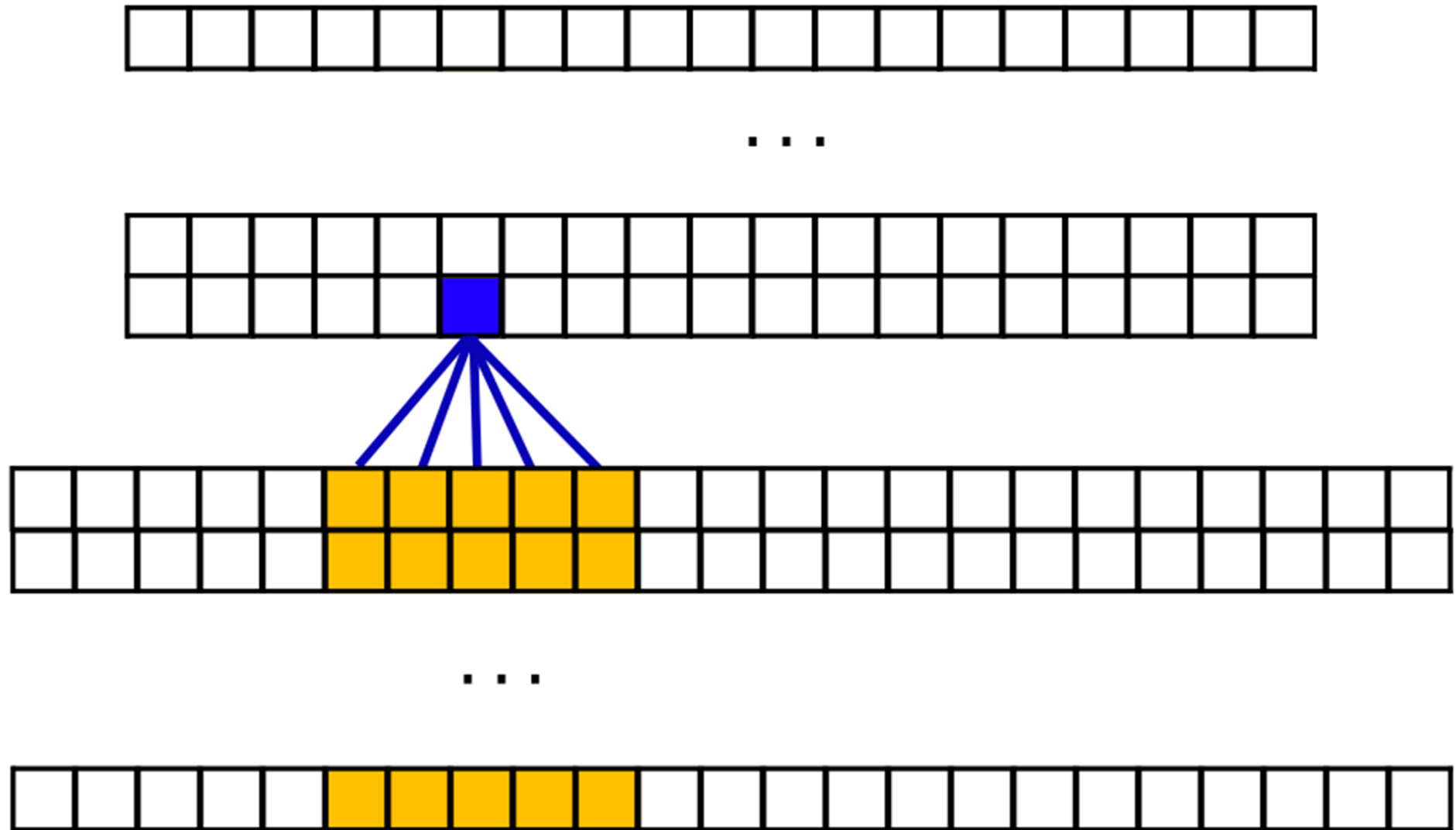
Hiperparaméterek:

depth@input, filter@filter_depth, stride, zeropadding

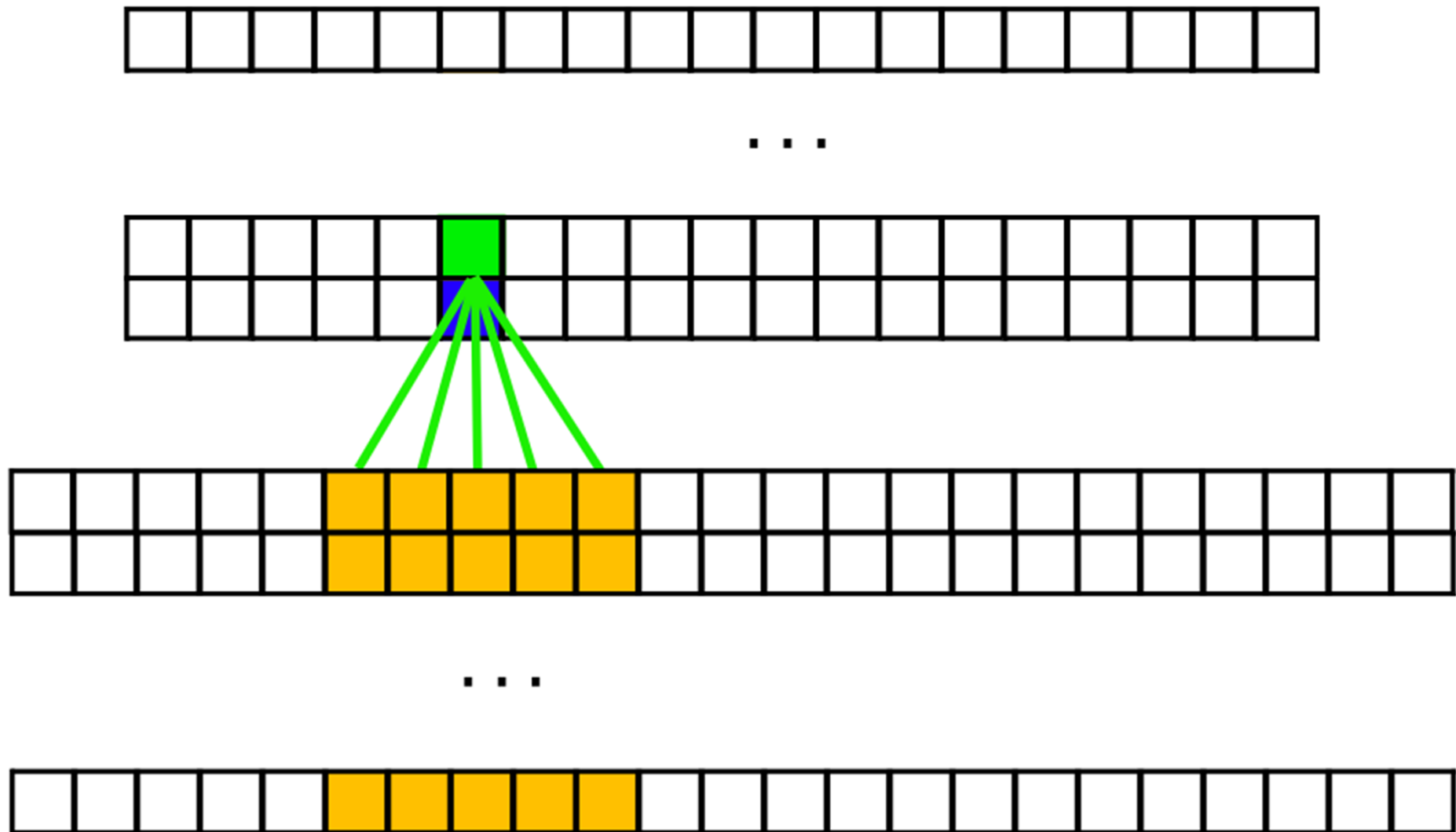
Pl. XYZ tengely gyorsulása bemenet → 12 conv1D:

3@200,5@12,s:1,z:0

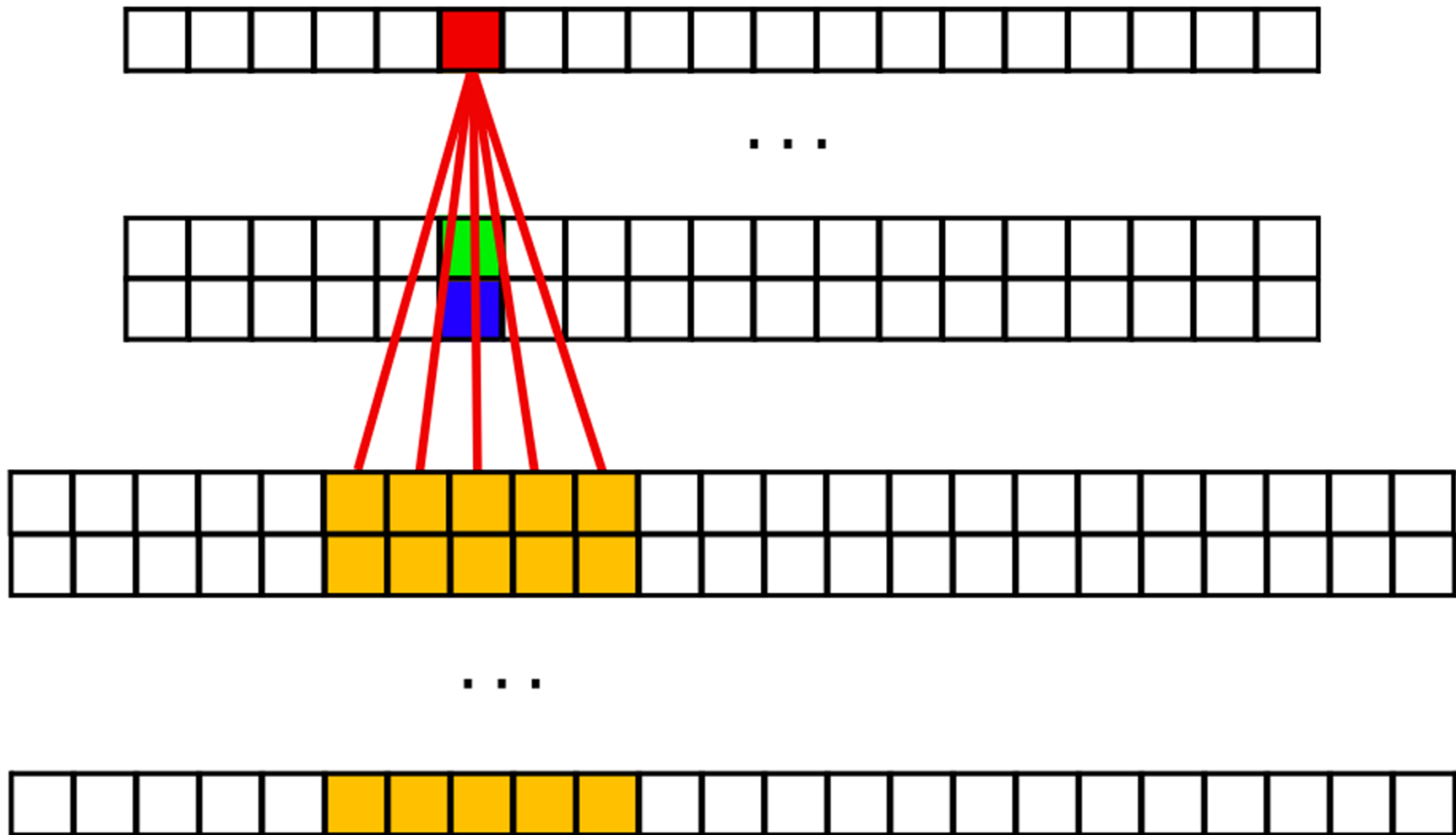
1D konvolúció szemléltetés



1D konvolúció szemléltetés



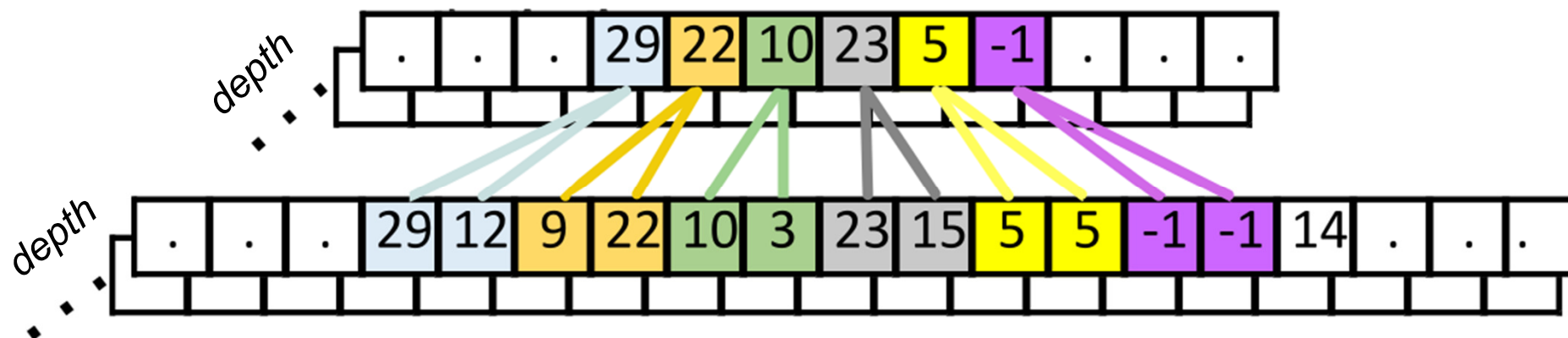
1D konvolúció szemléltetés



1D max-pooling

Hiperparaméterek: *filter*, *stride*

Például: f:2,s:2



1D konvolúció megvalósítás

TensorFlow

```
g = tf.Graph()

with g.as_default():

    # adat:[batch,x,y,depth]
    x = tf.Variable(np.array([0.0, 0.0, 0.0, 0.0, 1.0])
                    .reshape(1, 1, 5, 1), name="x")

    # szűrő:[filter_x,filter_y,depth,filter_depth]
    phi = tf.Variable(np.array([0.0, 0.5, 1.0])
                     .reshape(1, 3, 1, 1), name="phi")

    conv = tf.nn.conv2d(x, phi, strides=[1, 1, 1, 1],
                        padding="SAME", name="conv1d")
```

1D konvolúció megvalósítás

Keras

```
model.add(  
    Conv1D(filter_depth, filter, border_mode='same',  
    input_shape=(depth, x))  
)
```

Be-/kimenő adat: (*batch* x timesteps x features)

PyTorch

```
nn.Conv1d(depth, filter_depth, filter, stride)
```

Be-/kimenő adat: (*batch* x features x timesteps)

Adatok újraformázása 1.

Egy paraméterfolyam esetén mini-batch összeállítása:

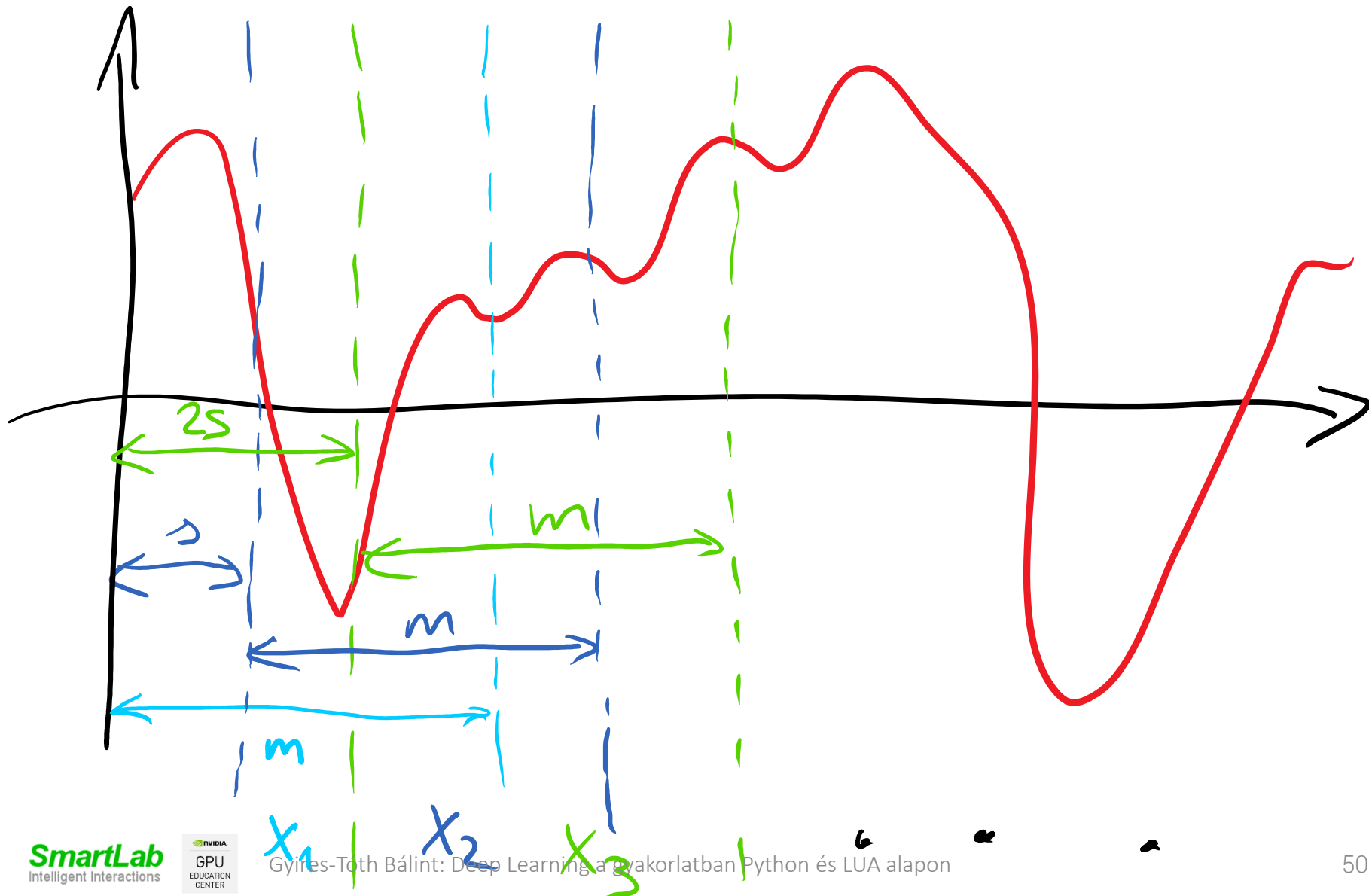
batch (b) x FrameSize (m) x 1

s: lépköz két minta között

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	.	.	.
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	---	---	---

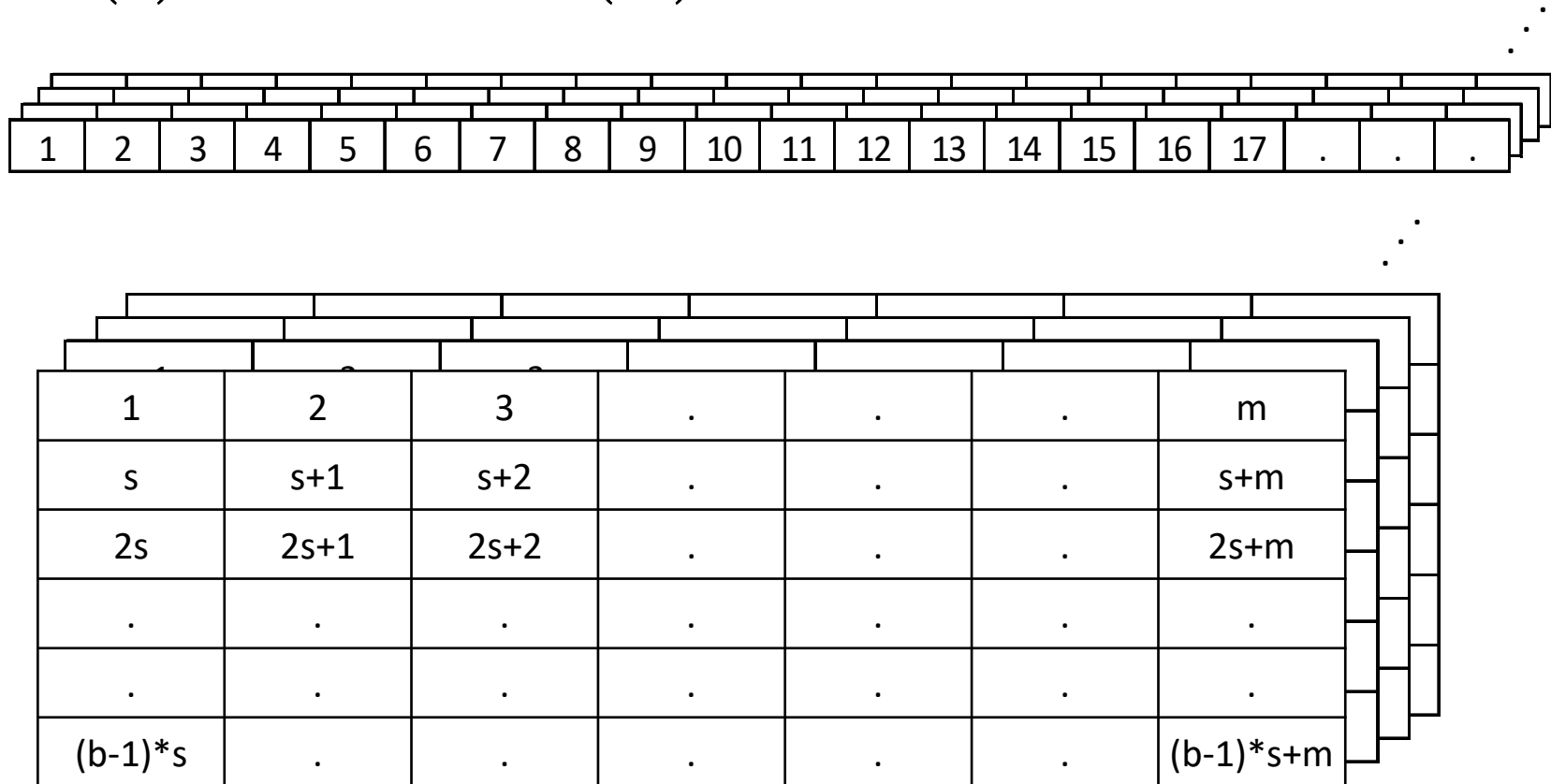
1	2	3	.	.	.	m
s	s+1	s+2	.	.	.	s+m
2s	2s+1	2s+2	.	.	.	2s+m
.	.	.	.	.	.	.
.	.	.	.	.	.	.
(b-1)*s	.	.	.	.	.	(b-1)*s+m

Ábrázolás



Adatok újraformázása 2.

Több paraméterfolyam esetén mini-batch összeállítása:
batch (b) x FrameSize (m) x nFrame



Példa 1D konvolúcióra

- Szövegek
 - Fordítás
 - Szentiment elemzés
 - Plágium detekció
 - Természetes nyelvfeldolgozás
 - Szövegszintézis
 - Beszédszintézis
- Idősorok
- Szenzoradatok
- Stb.

Kitekintés: 3D konvolúció.

Kahoot!

Névnek a NEPTUN
kódodat add meg!

<https://kahoot.it/>

Köszönöm a figyelmet!

`toth.b@tmit.bme.hu`



Jelen kutatás az Emberi Erőforrások Minisztériuma által az Új Nemzeti Kiválóság Program keretében meghírdetett Bolyai+ Felsőoktatási Fiatal Oktatói, Kutatói Ösztöndíj támogatásával készült (ÚNKP-20-5-BME-210).