

Dr. Zainkó Csaba

Deep Learning a gyakorlatban Python és LUA alapon Komplex architektúrák

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.

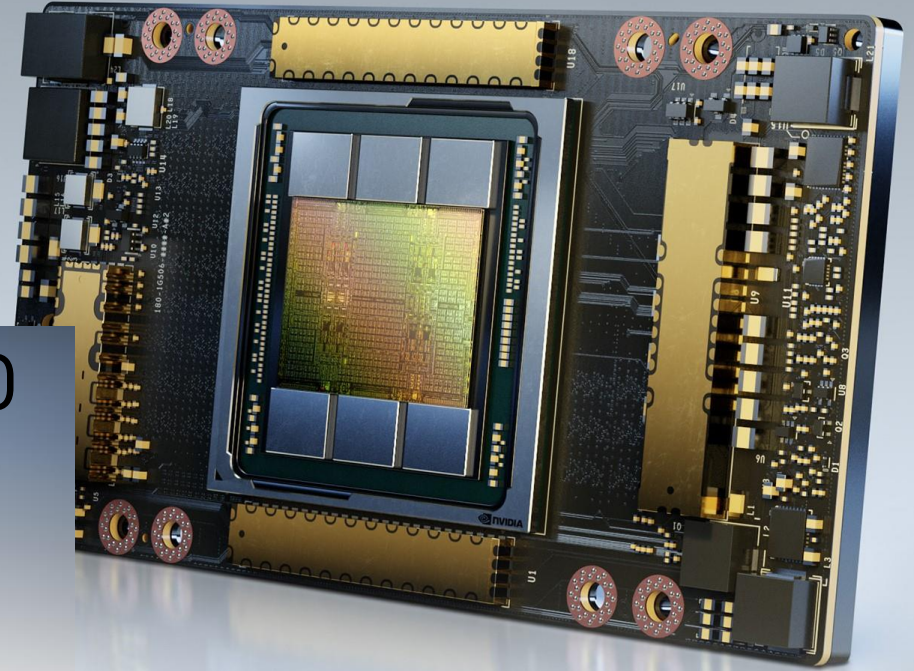
Deep Learning Híradó

Blizzard Challenge and VCC videó

- https://www.youtube.com/watch?v=shpDTs_qHk0&list=PLHty9pS2rtelQRXLjmFUUug41DZAZmYVK

Supercomputing 2020

NVIDIA® A100 80GB GPU



NVIDIA DGX Station A100



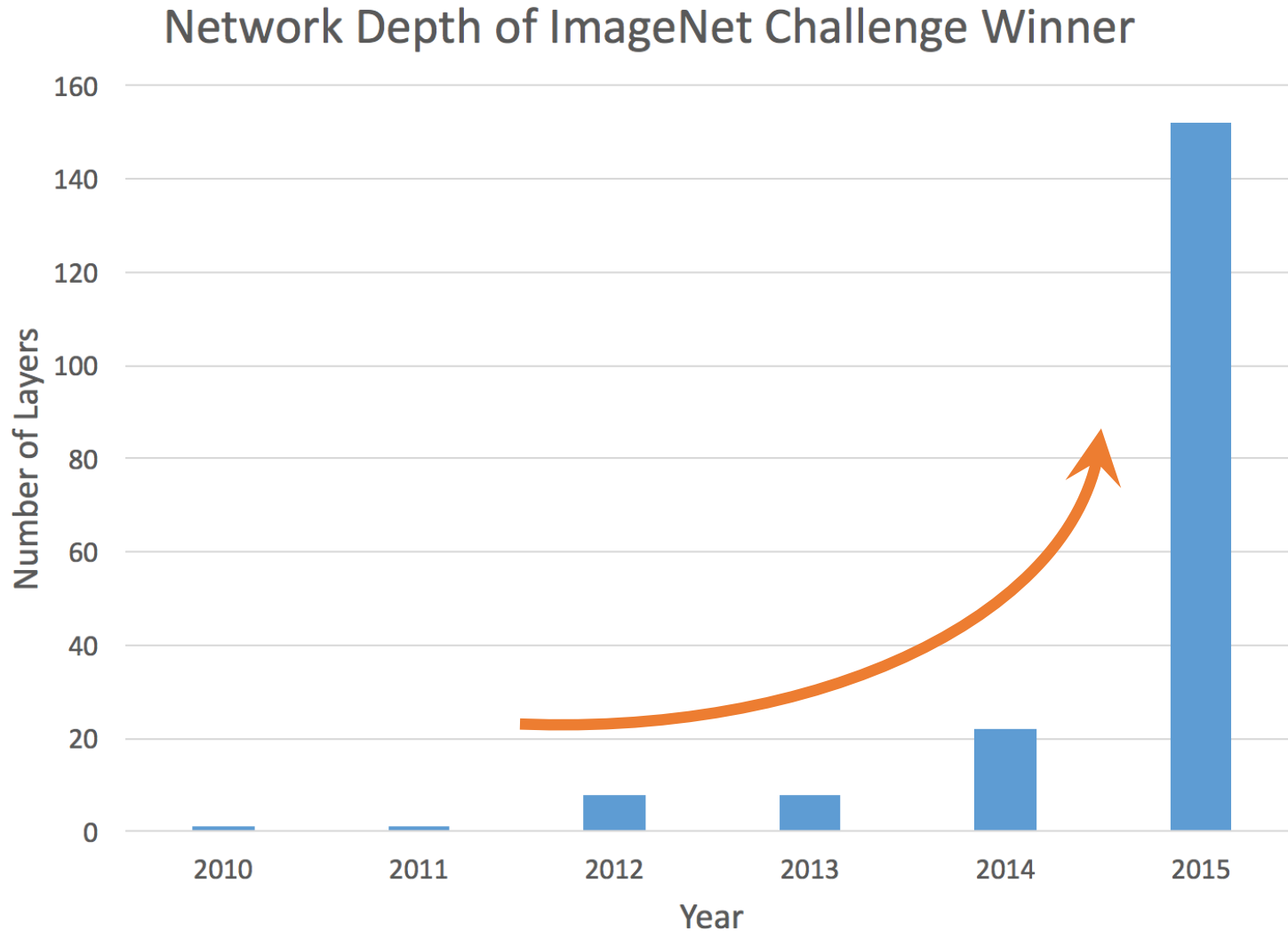
320GB GPU memory

<https://www.nvidia.com/en-us/events/supercomputing/>

Hirdetések

- 3. Kisházi kijavítva

ImageNet nyertes hálózatok

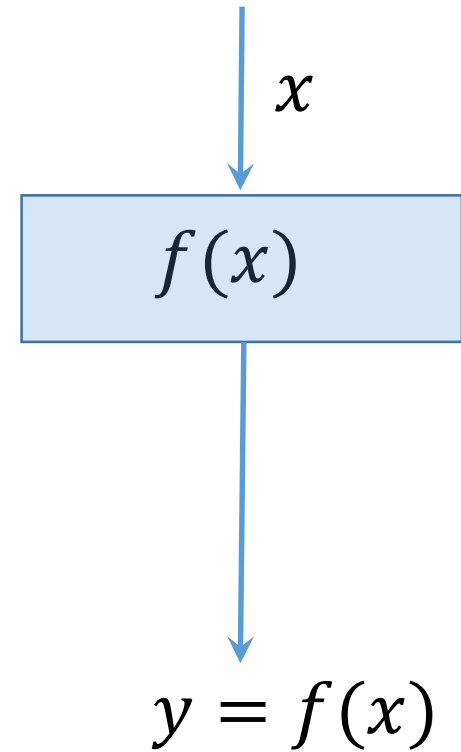


Forrás: <https://chatbotslife.com/resnets-highwaynets-and-densenets-oh-my-9bb15918ee32>

Residual Network

- Elenyésző gradiensek problémája (Vanishing gradient)

$$y = f(x)$$



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.

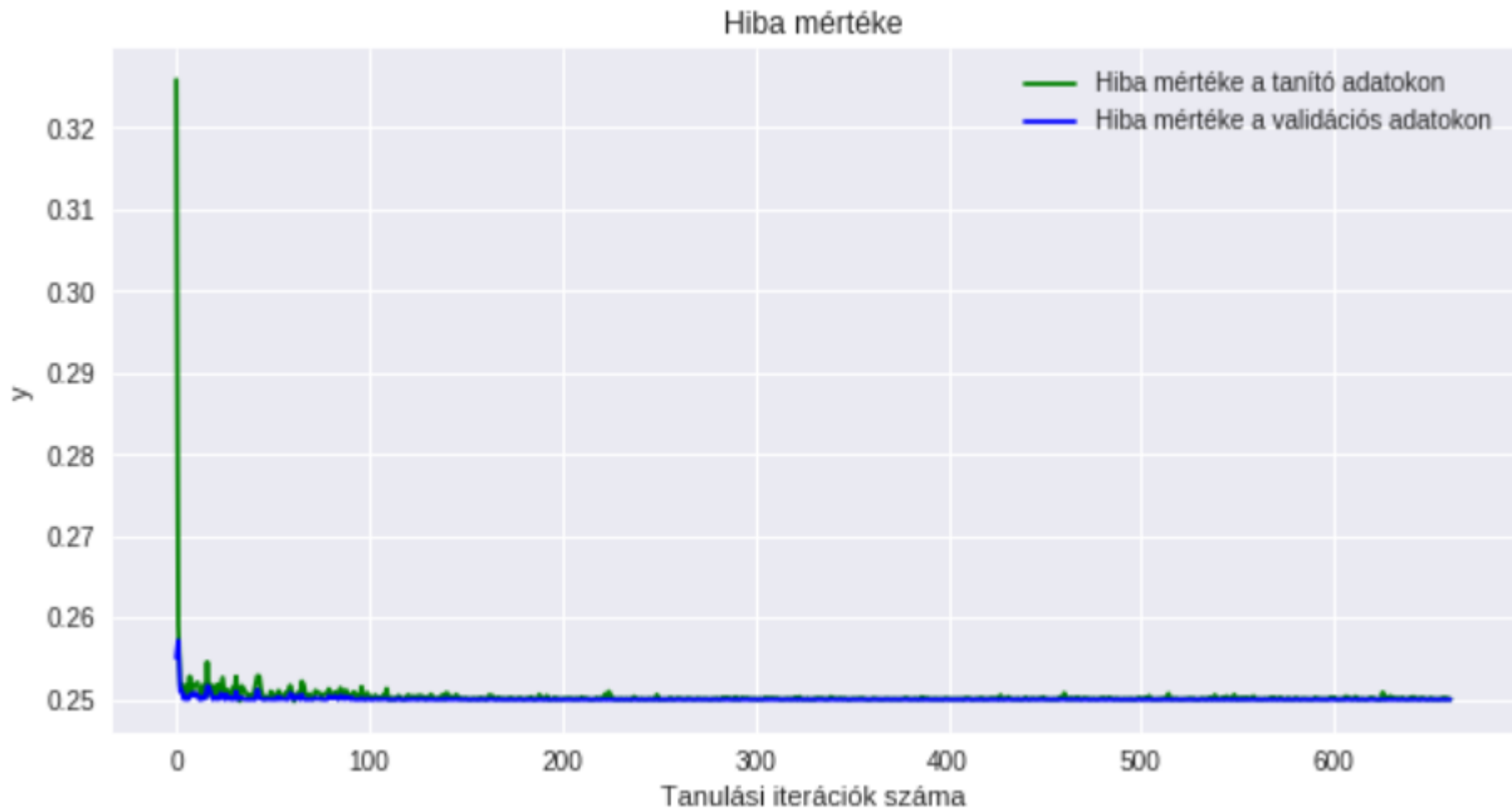
Oka: sigmoid / tanh az egész értéktartományt 0..1 (vagy -1..1) tartományba „nyomja össze”. A réteg bemenetén lévő nagy változások a kimeneten kis változást okoznak.

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

Keras XOR példa

```
inputs = Input(shape=(2,))
x =Dense(10)(inputs)
x =Activation("tanh")(x)
x1=x
for i in range(100):
    x=Dense(10)(x1)
    x=Activation("tanh")(x)
    x1=x
x=Dense(1)(x1)
pred=Activation("tanh")(x)
model=Model(inputs=inputs,outputs=pred)
```

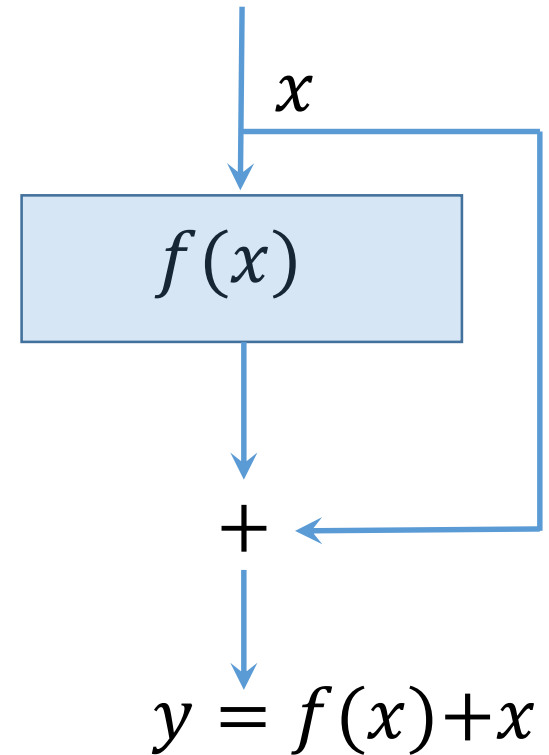
Keras XOR példa



Residual Network

- Ötlet: shortcut

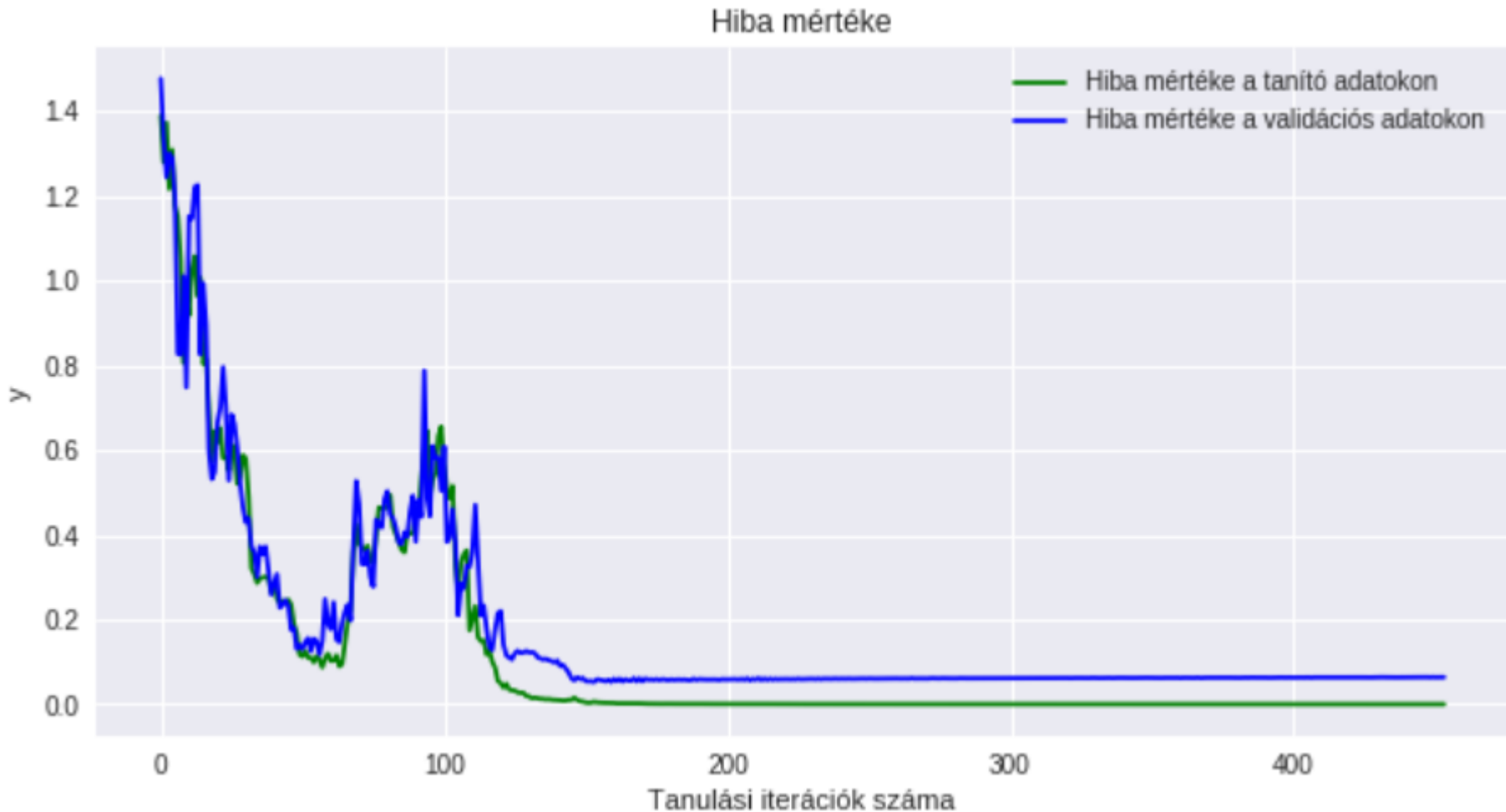
$$y = f(x) + x$$



Keras XOR példa - residual

```
inputs = Input(shape=(2,))
x =Dense(10)(inputs)
x =Activation("tanh")(x)
x1=x
for i in range(100):
    x=Dense(10)(x1)
    x=Activation("tanh")(x)
    x1 = Add()([x1, x])
x=Dense(1)(x1)
pred=Activation("tanh")(x)
model=Model(inputs=inputs,outputs=pred)
```

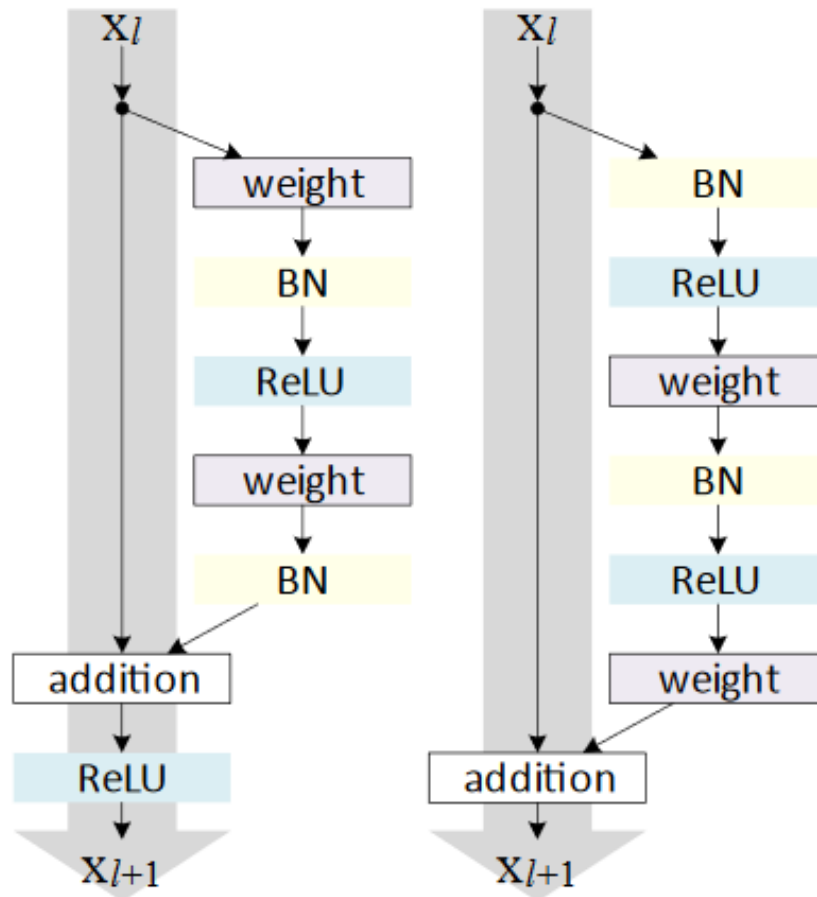
Keras XOR példa -residual



Test MSE: 0.060276

Residual Network

- ResNets
- 1001 layers
- CIFAR-10 4,62%
- Original:
 - 110 layers (6,61)
 - 1202 layers (7,93)



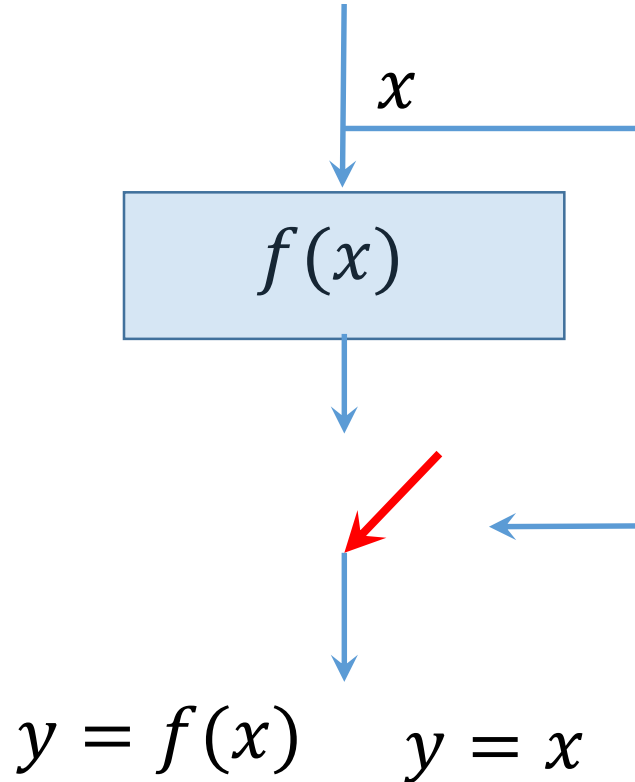
(a) original

(b) proposed

Forrás: Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun: Identity Mappings in Deep Residual Networks <https://arxiv.org/pdf/1603.05027.pdf>

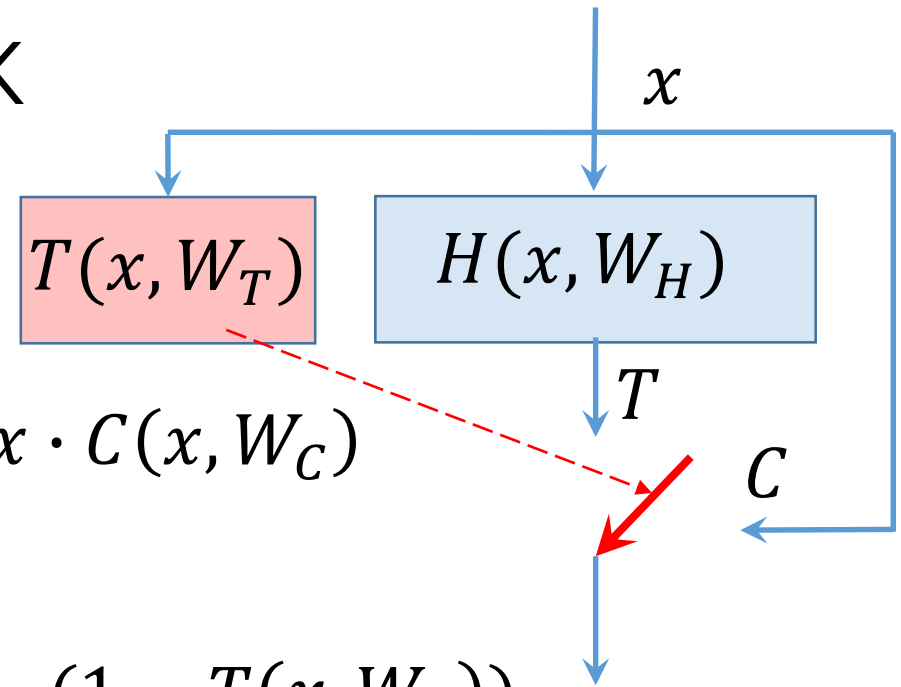
Highway network

- Hasonló a ResNet-hez, de tanulható paraméter, hogy kell-e shortcut.



Irodalom: Rupesh Kumar Srivastava, Klaus Greff, Jürgen Schmidhuber: Highway Networks <https://arxiv.org/abs/1505.00387>

Highway network



$$y = H(x, W_H) \cdot T(x, W_T) + x \cdot C(x, W_C)$$

$$C = 1 - T$$

$$y = H(x, W_H) \cdot T(x, W_T) + x \cdot (1 - T(x, W_T))$$

$$T(x) = \sigma(W_T^T x + b_T)$$

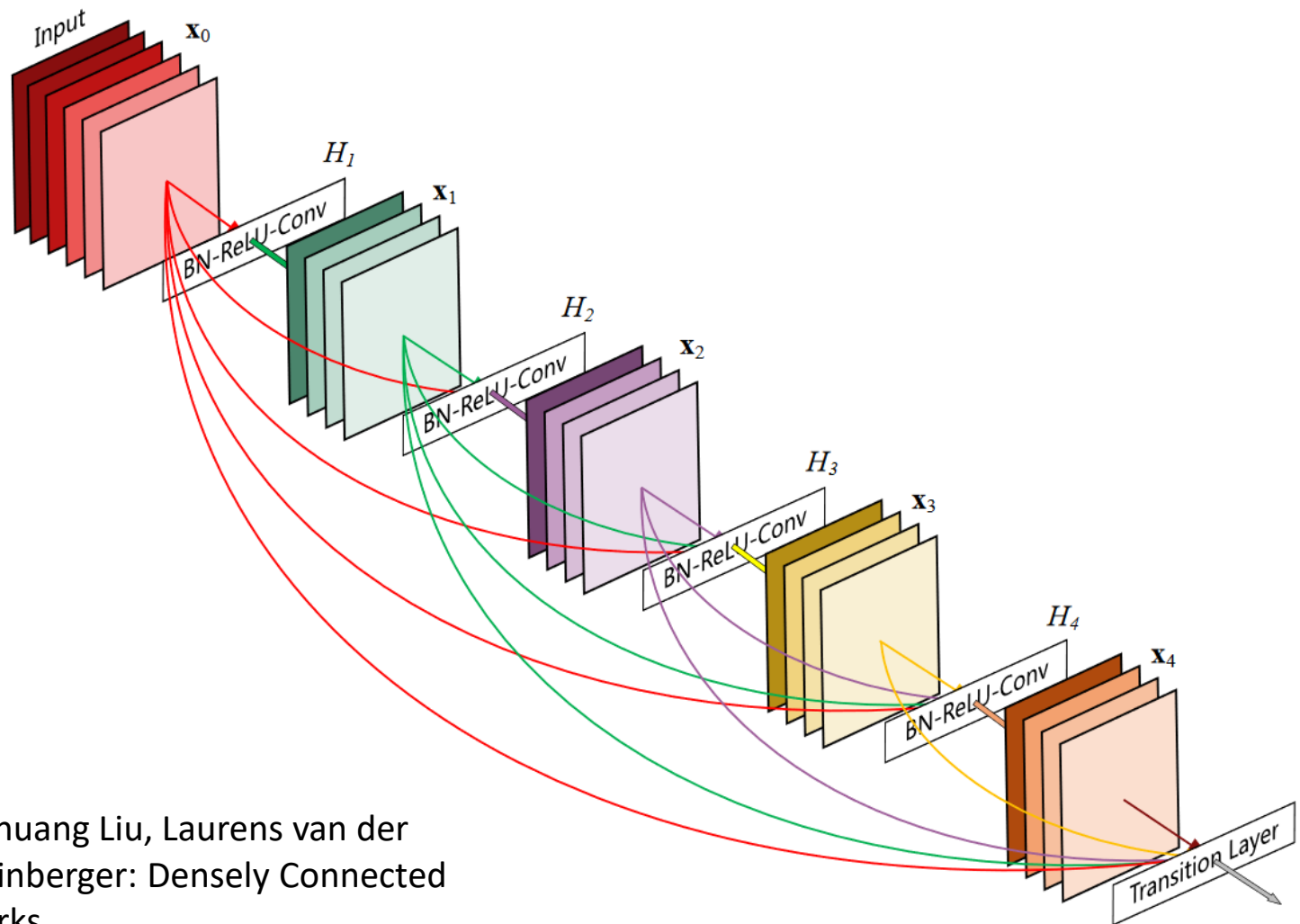
- b_T Bias, -1..-3 közé inicializáljuk,

Dense Network

- Alapesetben az adott réteg, az előző kimenetét használja
- $x_l = H(x_{l-1})$ vagy
- $x_l = H(x_{l-1}) + x_{l-1}$ ResNet esetén

- $x_l = H_l([x_0, x_1, \dots, x_{l-1}])$

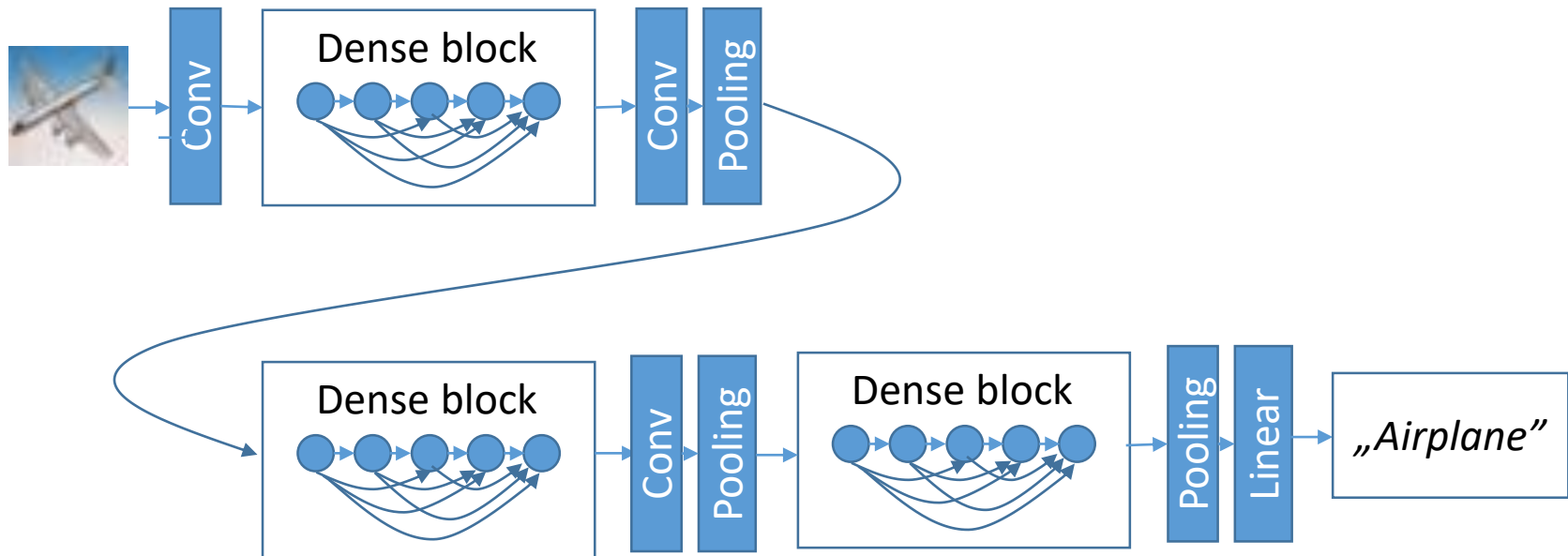
Dense Network



Forrás: Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger: Densely Connected Convolutional Networks

<https://arxiv.org/abs/1608.06993>

DenseNet



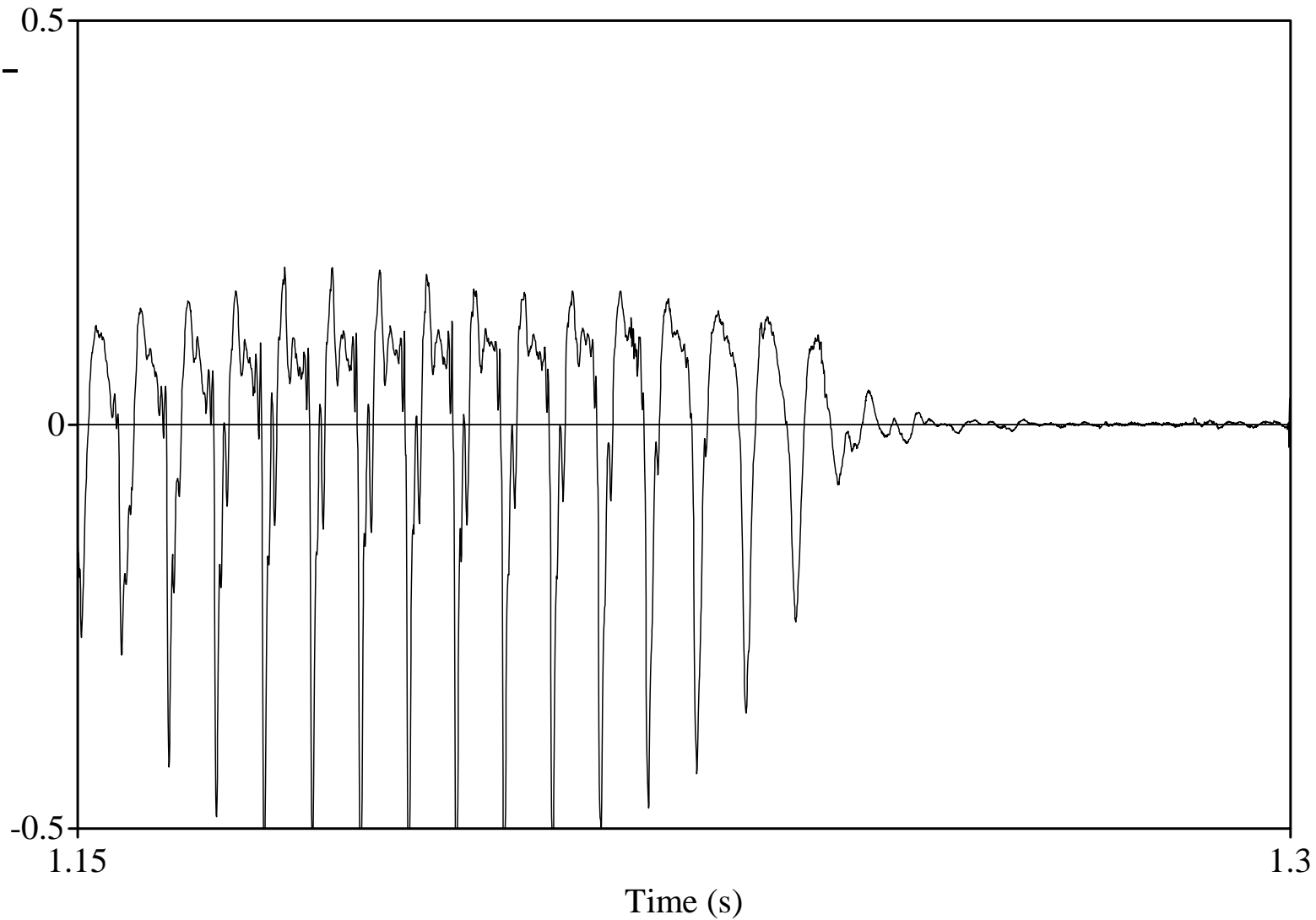
Forrás: Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger: Densely Connected Convolutional Networks
<https://arxiv.org/abs/1608.06993>

WaveNet

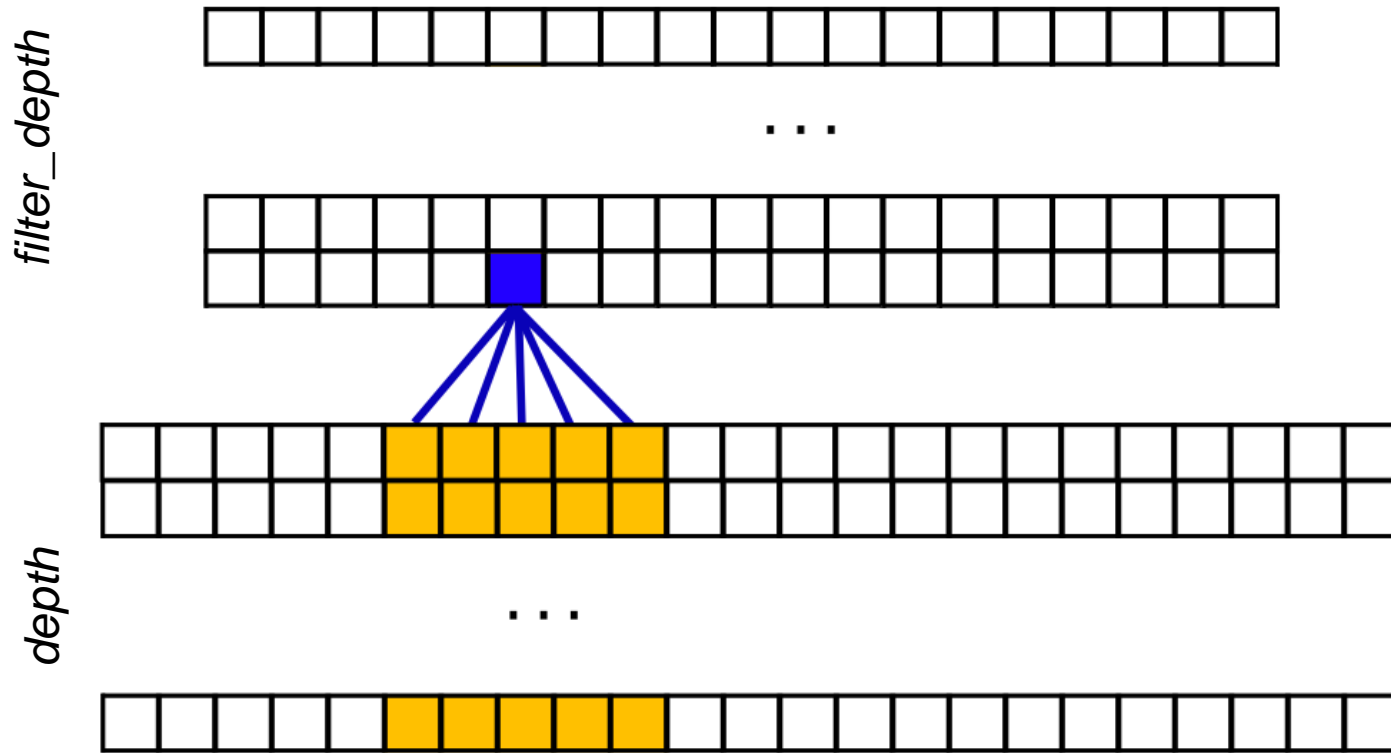
- Google DeepMind 2016 szeptember
 - Hullámforma előállítás közvetlenül
 - Google Assistant hang
-
- 16 kHz v. 24 kHz

WaveNet

- Hullámforma



1D 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



...



...



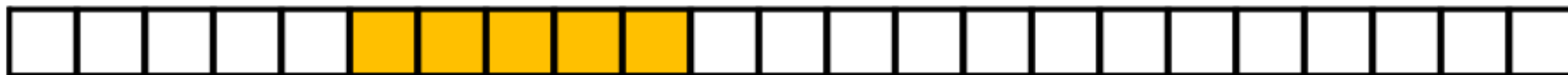
1D konvolúció szemléltetés



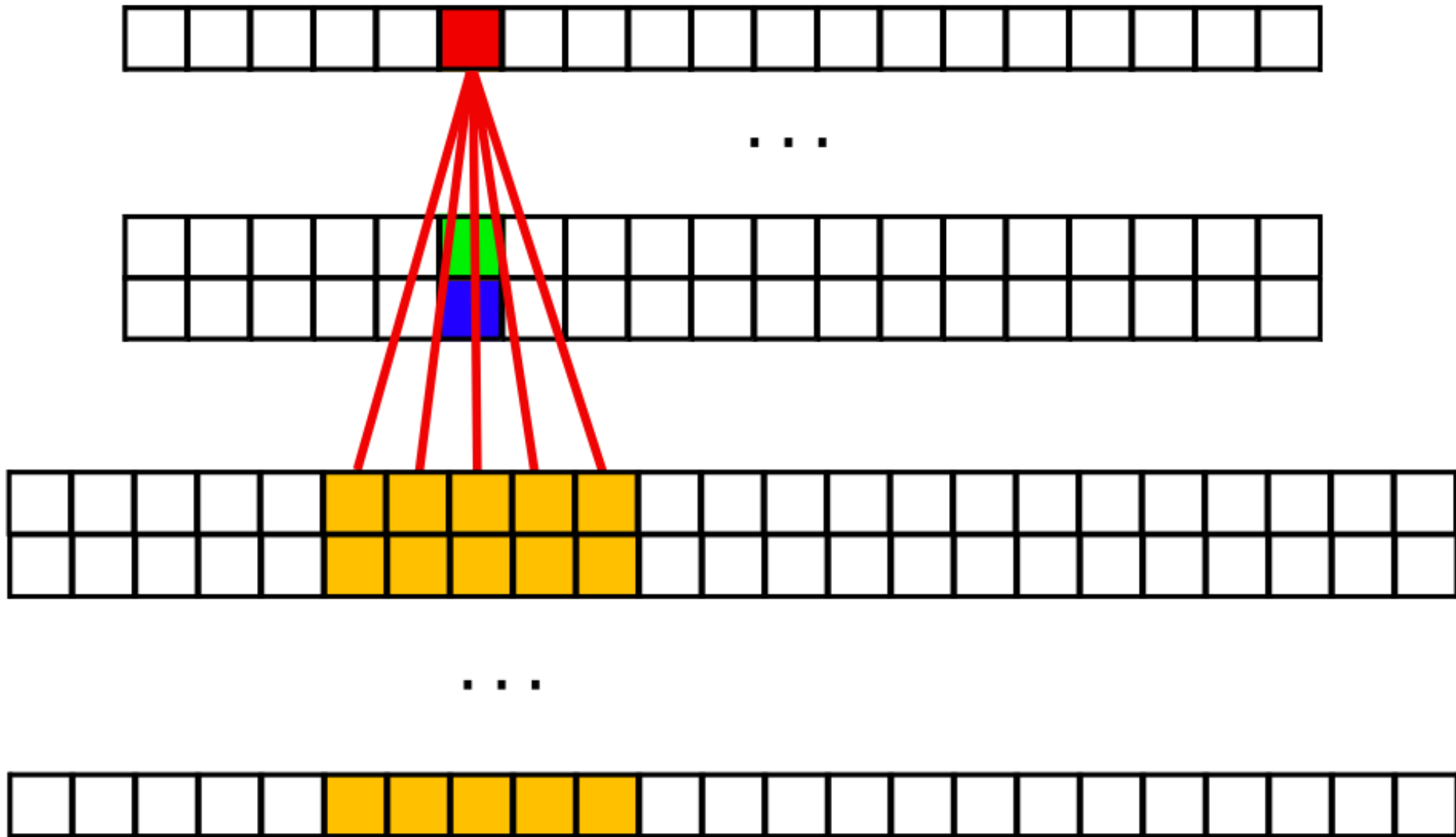
...



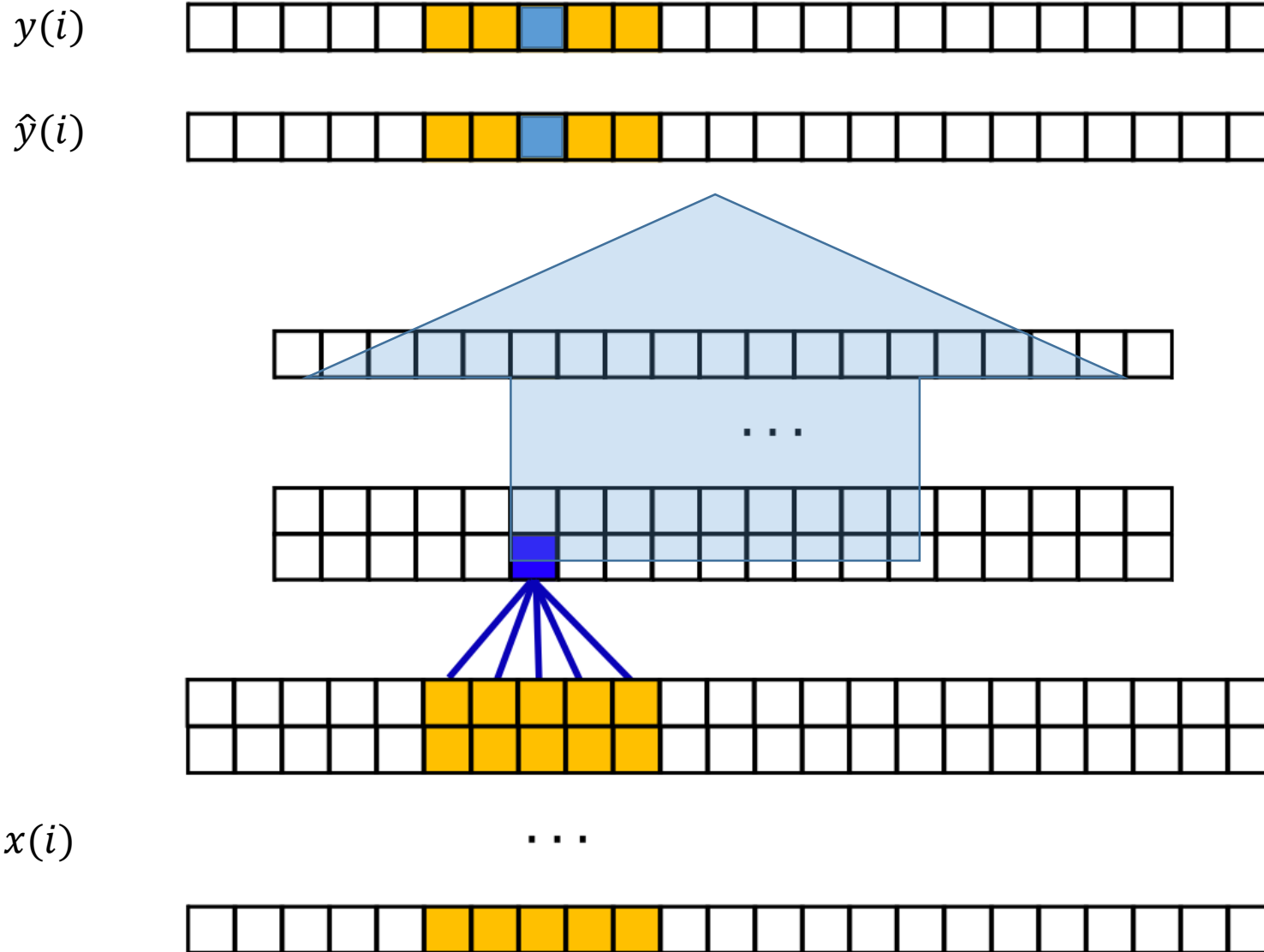
...



1D konvolúció szemléltetés



1D konvolúció

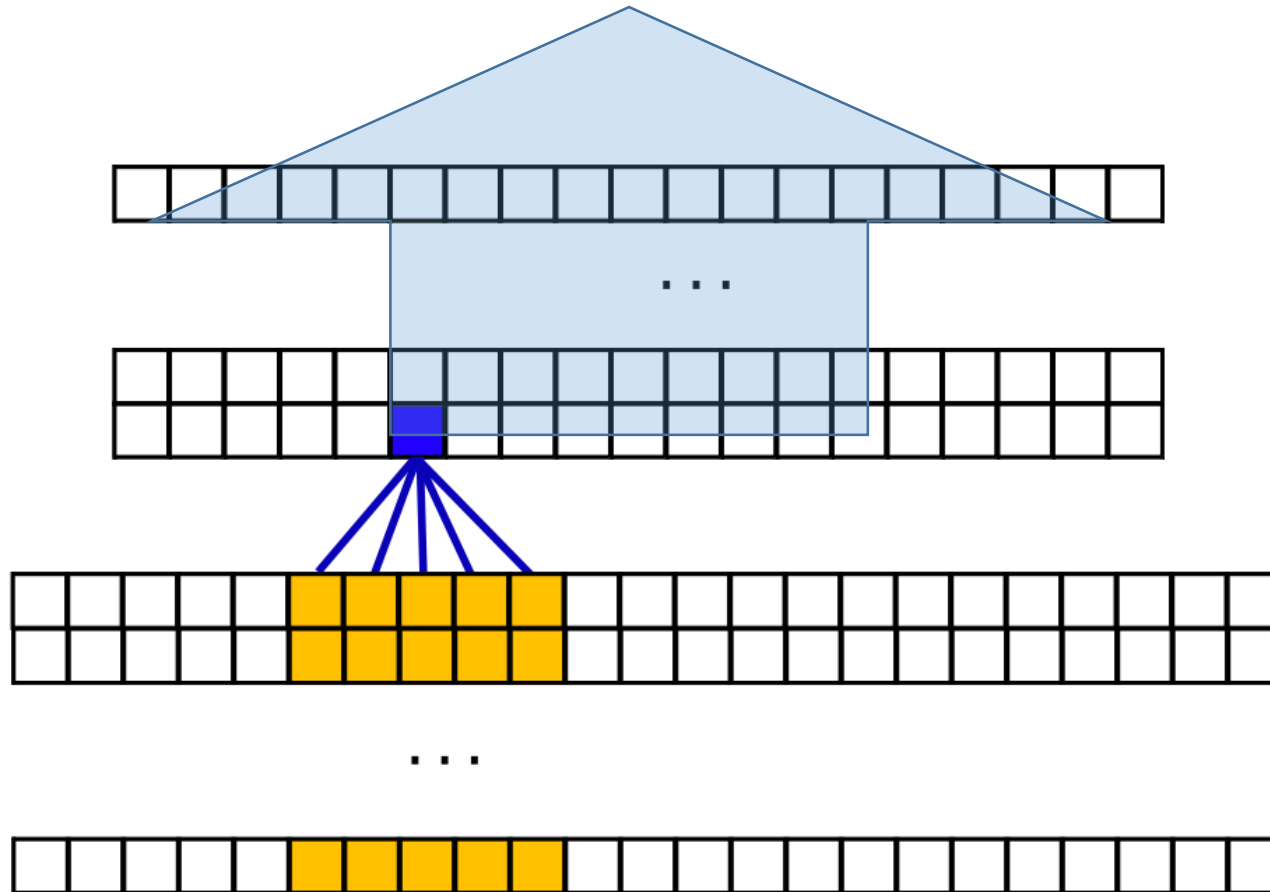


Kauzális konvolúció

$y(i)$

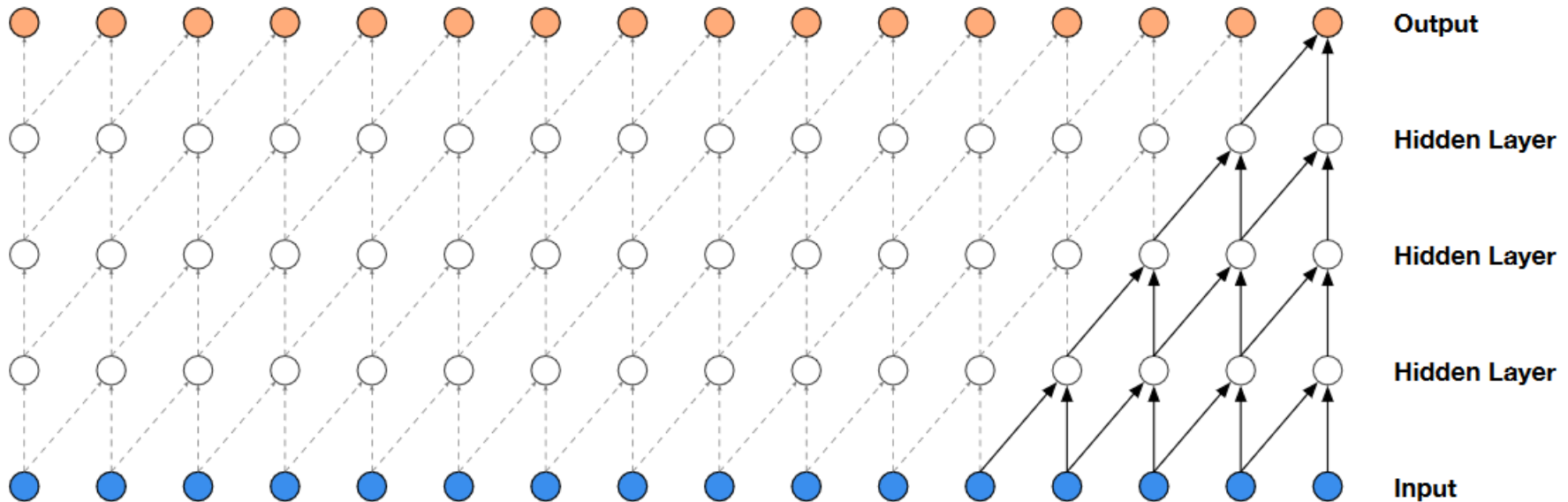


$\hat{y}(i)$



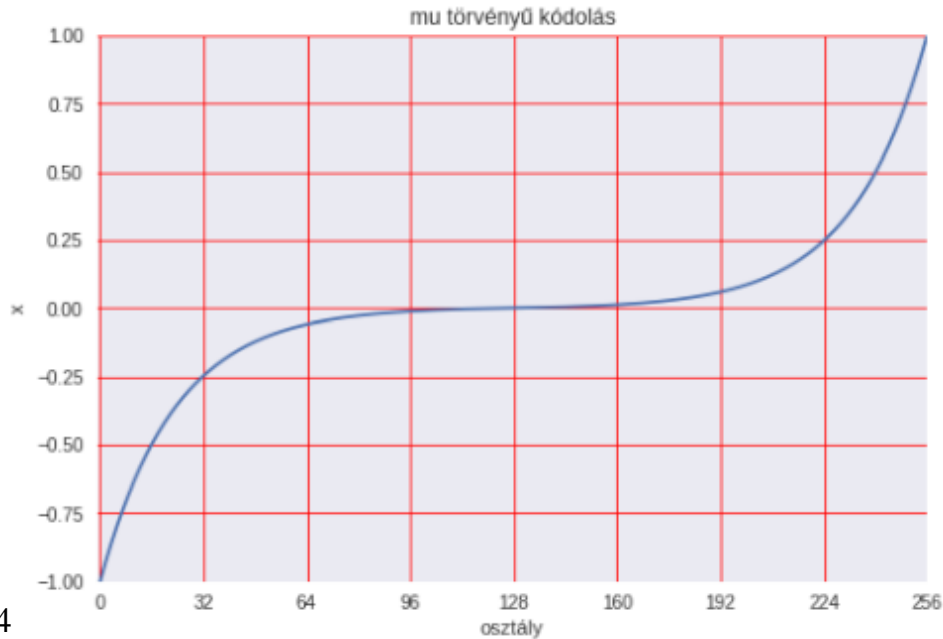
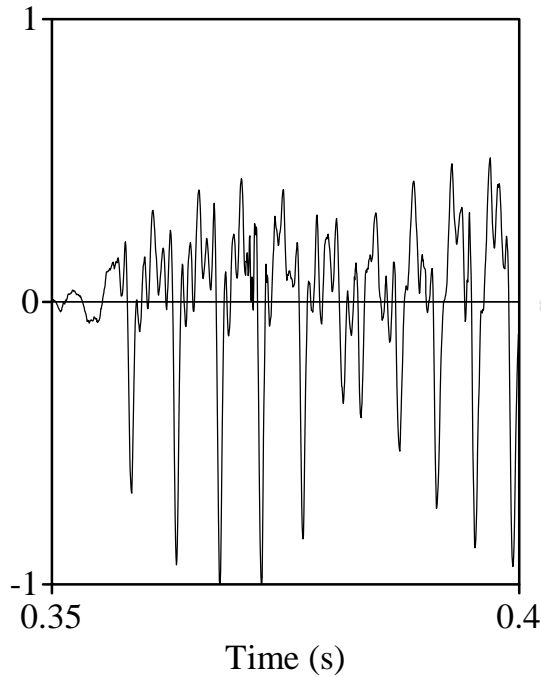
WaveNet

- Causal convolution

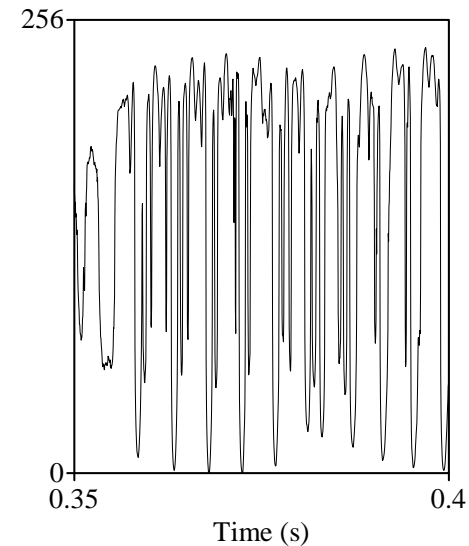
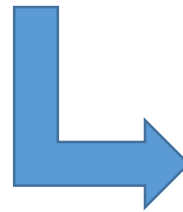


Forrás: Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu:
WaveNet: A Generative Model for Raw Audio <https://arxiv.org/pdf/1609.03499.pdf>

WaveNet - input



$$f(x_t) = \text{sign}(x_t) \frac{\ln(1 + \mu|x_t|)}{\ln(1 + \mu)}$$



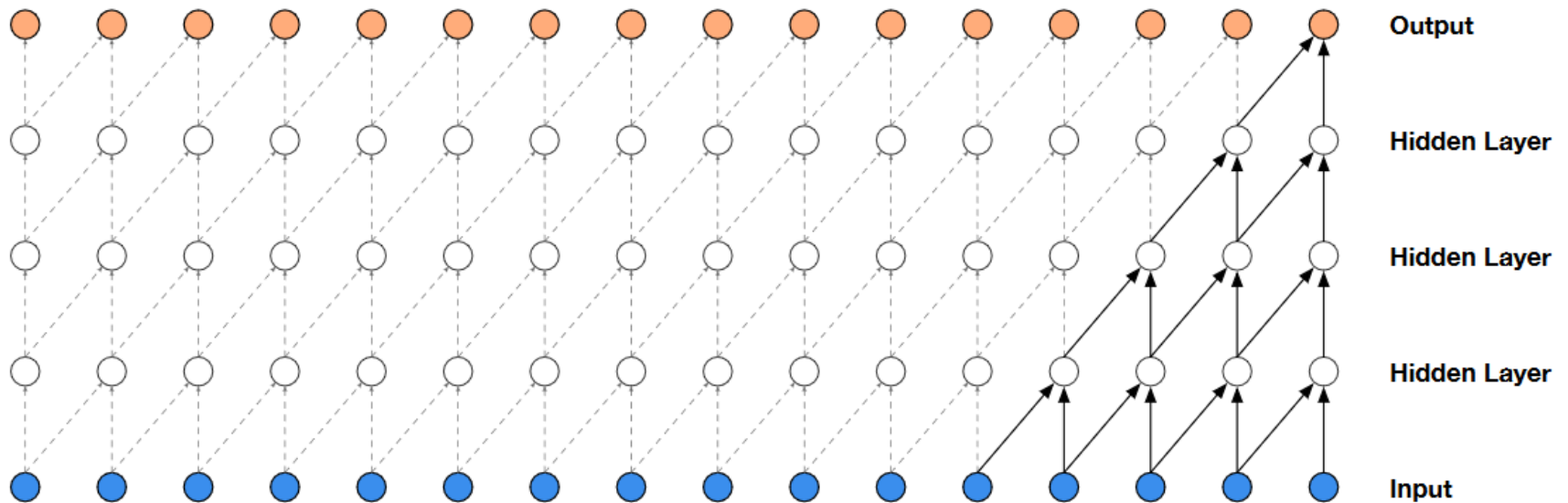
WaveNet

- One-hot kódolás
 - valóságban 256 osztály
 - példa: 4 osztály

Bemeneti csatornák	1	0	0	0	0	1	0	1
	0	0	1	1	0	0	0	0	
	0	1	0	0	0	0	1	0	
	0	0	0	0	1	0	0	0	
Idő									

WaveNet

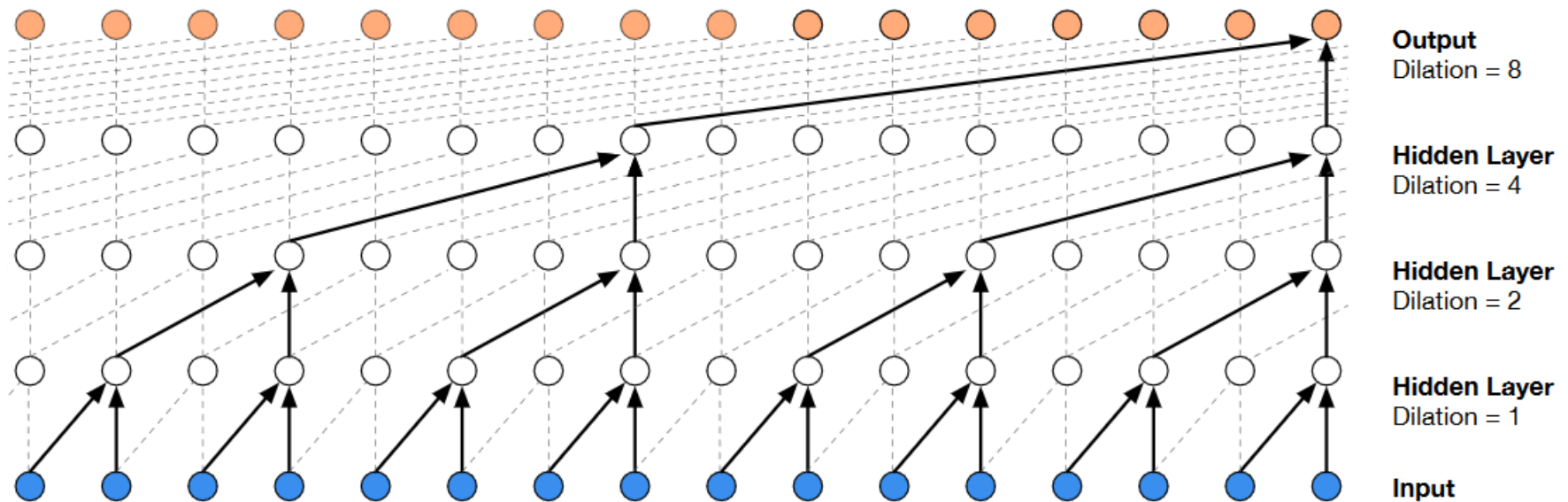
- Receptiv field



- 3 rejtett réteg -> 5 minta
- Pl: 80 réteg -> 83 minta ~5 ms

WaveNet

- Dilated convolution



- 3 rejtett réteg -> 16 minta

WaveNet

- Dilated convolution
- 10 „rejtett réteg”
 - 1,2,4,8,16,32,64,128,256,512 -> 1024 minta
- 20-80 „rejtett réteg” $\sim 2-8 * 1024 \rightarrow \sim 125 \text{ ms} - 500 \text{ ms}$

WaveNet

- Dilated konvolúció példa

Be:

1	3	2	0	4	3	1	0
---	---	---	---	---	---	---	---

Ki:

7	7	2	8	10	5	1
---	---	---	---	----	---	---

Szűrő, dilation=1

1	2
---	---

Szűrő, dilation=2

1	2
---	---

1	3	2	0	4	3	1	0
---	---	---	---	---	---	---	---

5	3	10	6	6	3
---	---	----	---	---	---

Ekvivalens szűrő:

1	0	2
---	---	---

Szűrő, dilation=4

1	2
---	---

1	3	2	0	4	3	1	0
---	---	---	---	---	---	---	---

9	9	4	0
---	---	---	---

Ekvivalens szűrő:

1	0	0	0	2
---	---	---	---	---

WaveNet

Szűrők width=2, input=4, dilation=2, output=3

Be:

1	2	0	1	4	1	3	1
4	2	6	1	0	0	0	3
1	5	6	4	3	0	1	0
2	0	1	0	1	4	0	1

.....

1	0	4	1	3	5
3	6	2	0	1	2
2	2	1	4	0	2
4	1	3	1	2	4

1	2	0	1	4	1
4	2	6	1	0	0
1	5	6	4	3	0
2	0	1	0	1	4

.....

0	1	4	1	3	1
6	1	0	0	0	3
6	4	3	0	1	0
1	0	1	4	0	1

.....

+

0	6	2	1
1	0	4	1
5	2	2	4

=

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

WaveNet

Szűrők width=2, input=4, dilation=2, output=3

Be:

1	2	0	1	4	1	3	1
4	2	6	1	0	0	0	3
1	5	6	4	3	0	1	0
2	0	1	0	1	4	0	1

.....

1	0	4	1	3	5
3	6	2	0	1	2
2	2	1	4	0	2
4	1	3	1	2	4

1	2	0	1	4	1
4	2	6	1	0	0
1	5	6	4	3	0
2	0	1	0	1	4

.....

0	1	4	1	3	1
6	1	0	0	0	3
6	4	3	0	1	0
1	0	1	4	0	1

.....

+

0	6	2	1
1	0	4	1
5	2	2	4

=

49	14	7	4	2	19
25	17	17	5	7	2
28	15	30	21	17	15

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

23	18	34	12	14	17
19	17	21	10	22	16
11	8	8	4	14	11

WaveNet

Szűrők width=2, input=4, dilation=2, output=3

Be:

1	2	0	1	4	1	3	1
4	2	6	1	0	0	0	3
1	5	6	4	3	0	1	0
2	0	1	0	1	4	0	1

.....

1	0	4	1	3	5
3	6	2	0	1	2
2	2	1	4	0	2
4	1	3	1	2	4

23	18	34	12	14	17
19	17	21	10	22	16
11	8	8	4	14	11

+

49	14	7	4	2	19
25	17	17	5	7	2
28	15	30	21	17	15

=

WaveNet

Szűrők width=2, input=4, dilation=2, output=3

Be:

1	2	0	1	4	1	3	1
4	2	6	1	0	0	0	3
1	5	6	4	3	0	1	0
2	0	1	0	1	4	0	1

.....

1	0	4	1	3	5
3	6	2	0	1	2
2	2	1	4	0	2
4	1	3	1	2	4

23	18	34	12	14	17
19	17	21	10	22	16
11	8	8	4	14	11

+

49	14	7	4	2	19
25	17	17	5	7	2
28	15	30	21	17	15

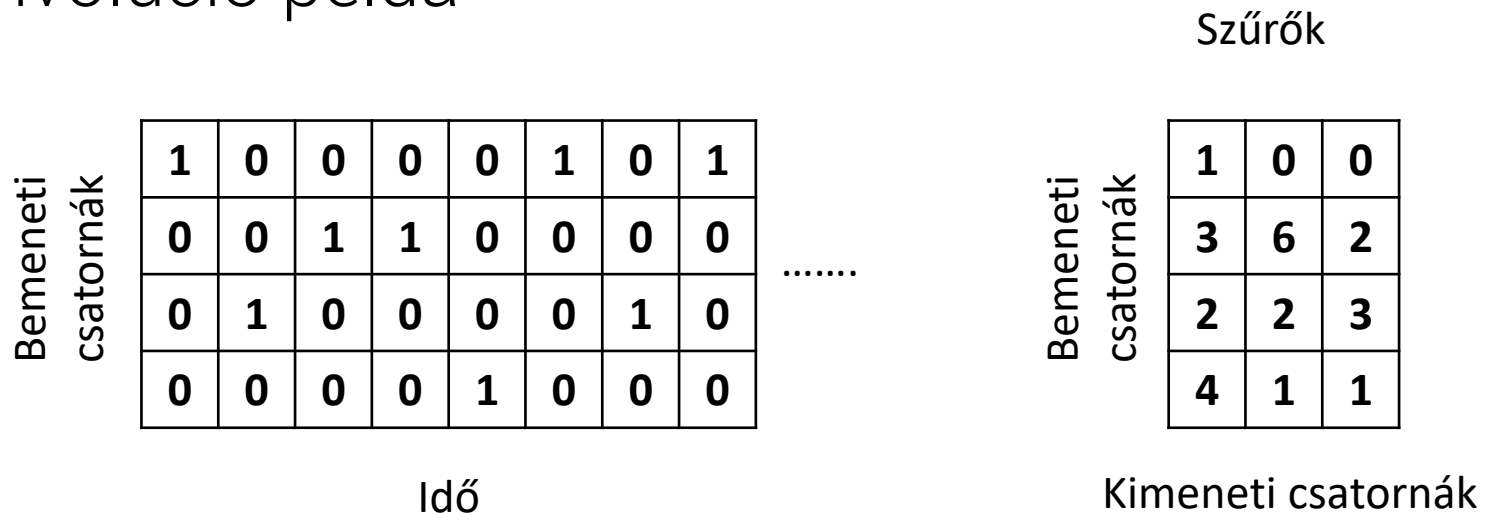
=

72	32	41	16	16	36
44	34	38	15	29	18
39	23	38	25	31	26

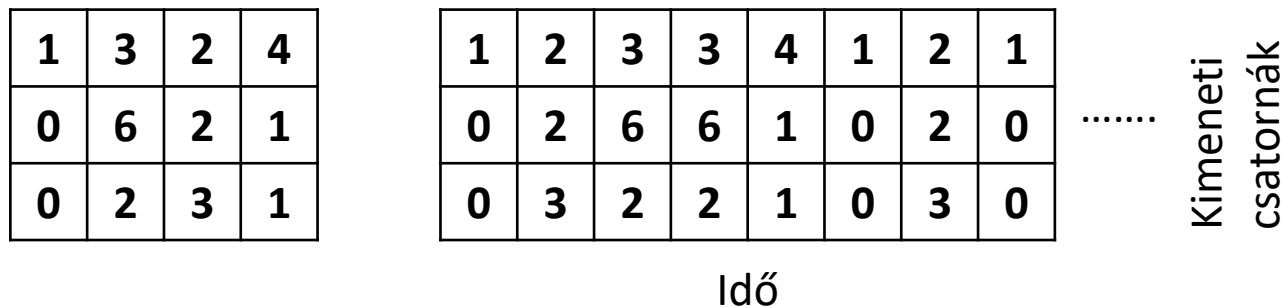
.....

WaveNet

- 1x1 konvolúció példa



Transponált szűrők



WaveNet

- Loss function
 - Softmax

$$y_i(x) = \frac{e^{x_i}}{\sum_{j=1}^{256} e^{x_j}}$$

- Áltagos kereszt-entrópia (idő)

$$C = - \sum_{n=1}^N \sum_{i=1}^{256} y_i^{(n)} \ln \left(\hat{y}_i^{(n)} \right) , \quad N = 100000$$

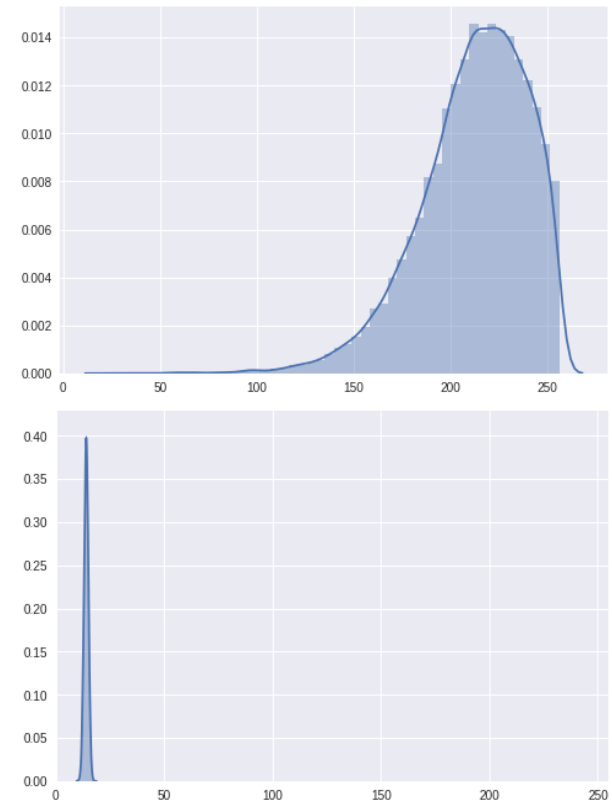
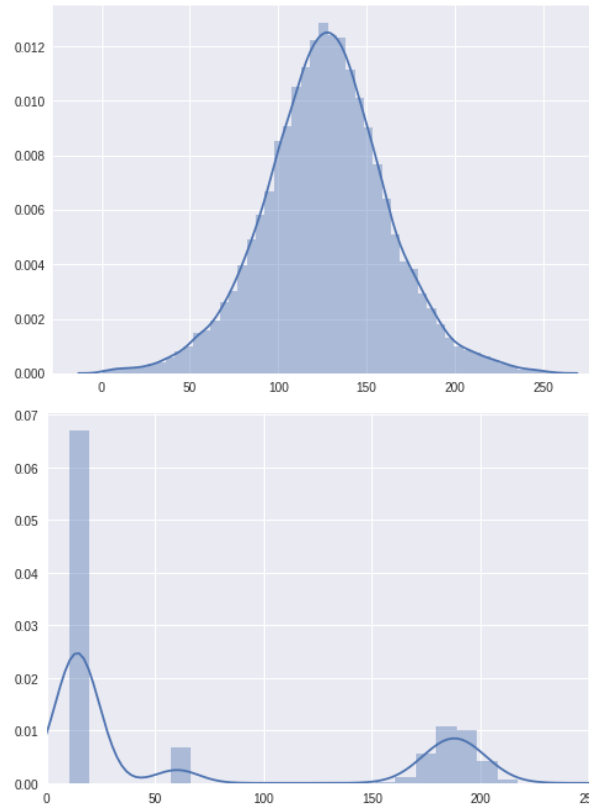
Kahoot!

Névnek a NEPTUN
kódodat add meg!

<https://kahoot.it/>

WaveNet

- Audió generálás
 - Eloszlásnak megfelelően
 - Legvalószínűbb
 - Temperáltan
 - Várható érték
 - Top k

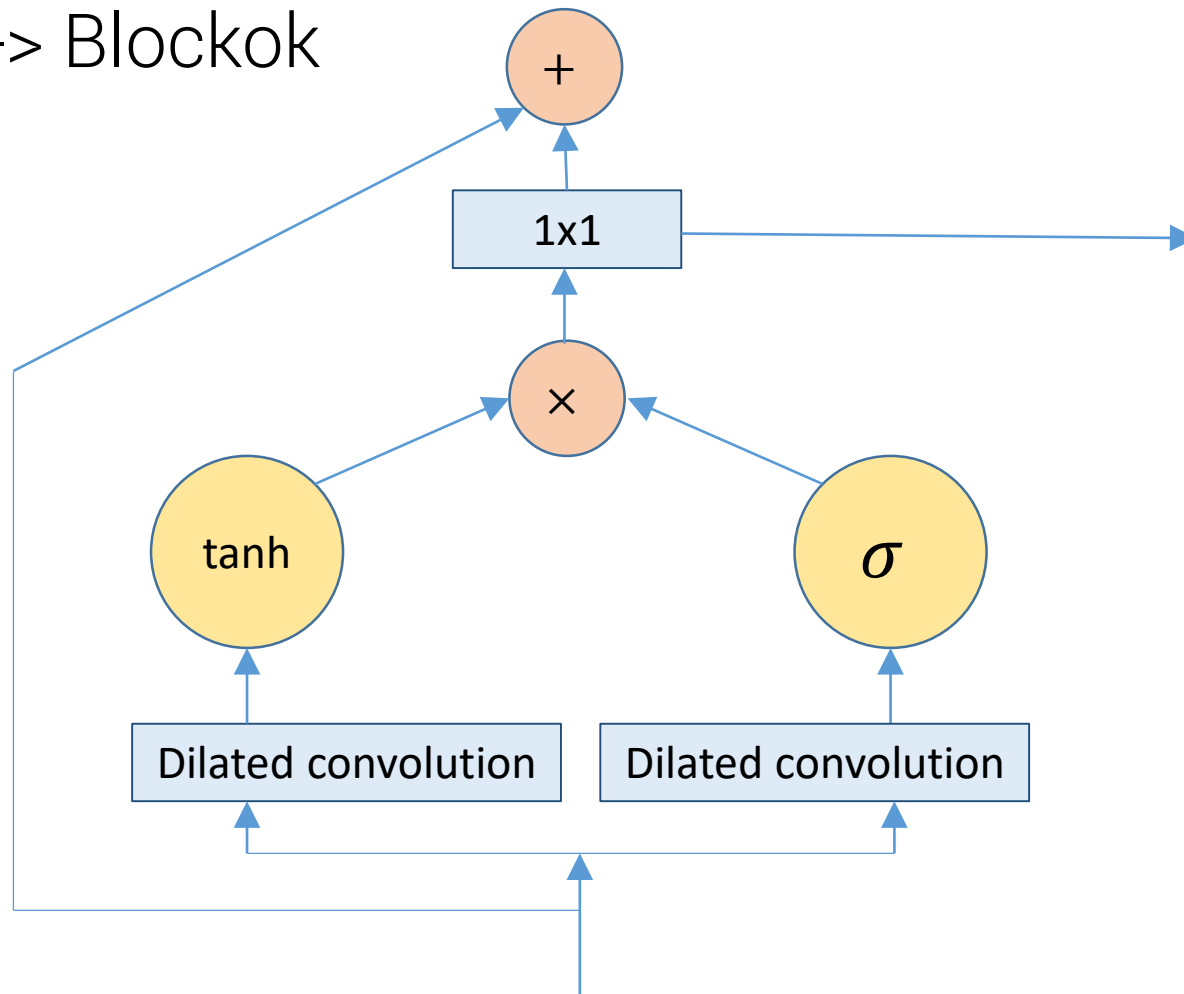


- Invert u-law
átalakítás:

$$f^{-1}(y_t) = \frac{\text{sign}(y_t)}{\mu} \left((1 + \mu)^{|y|} - 1 \right), -1 \leq y \leq 1$$

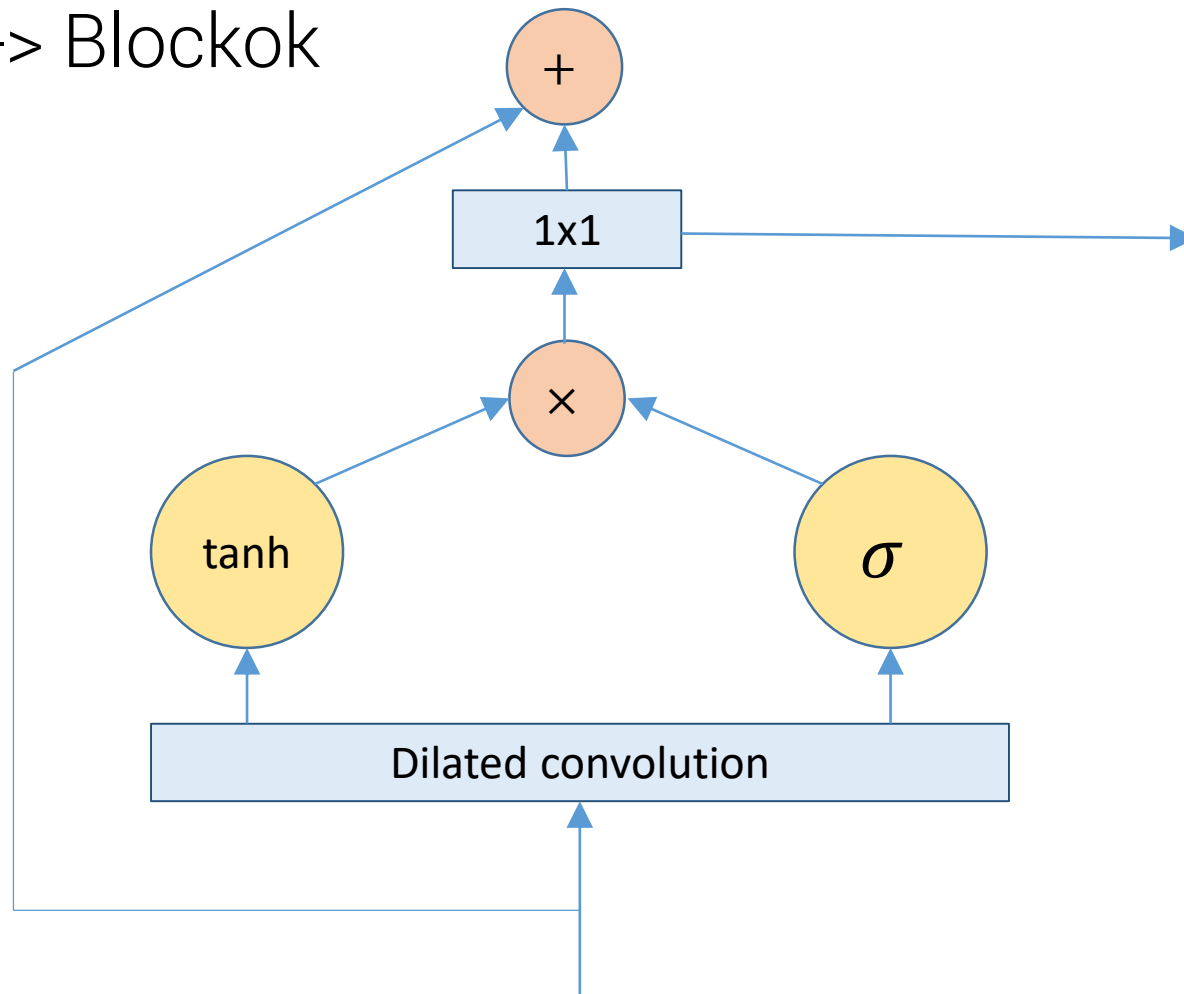
WaveNet

- Layerrek -> Blockok



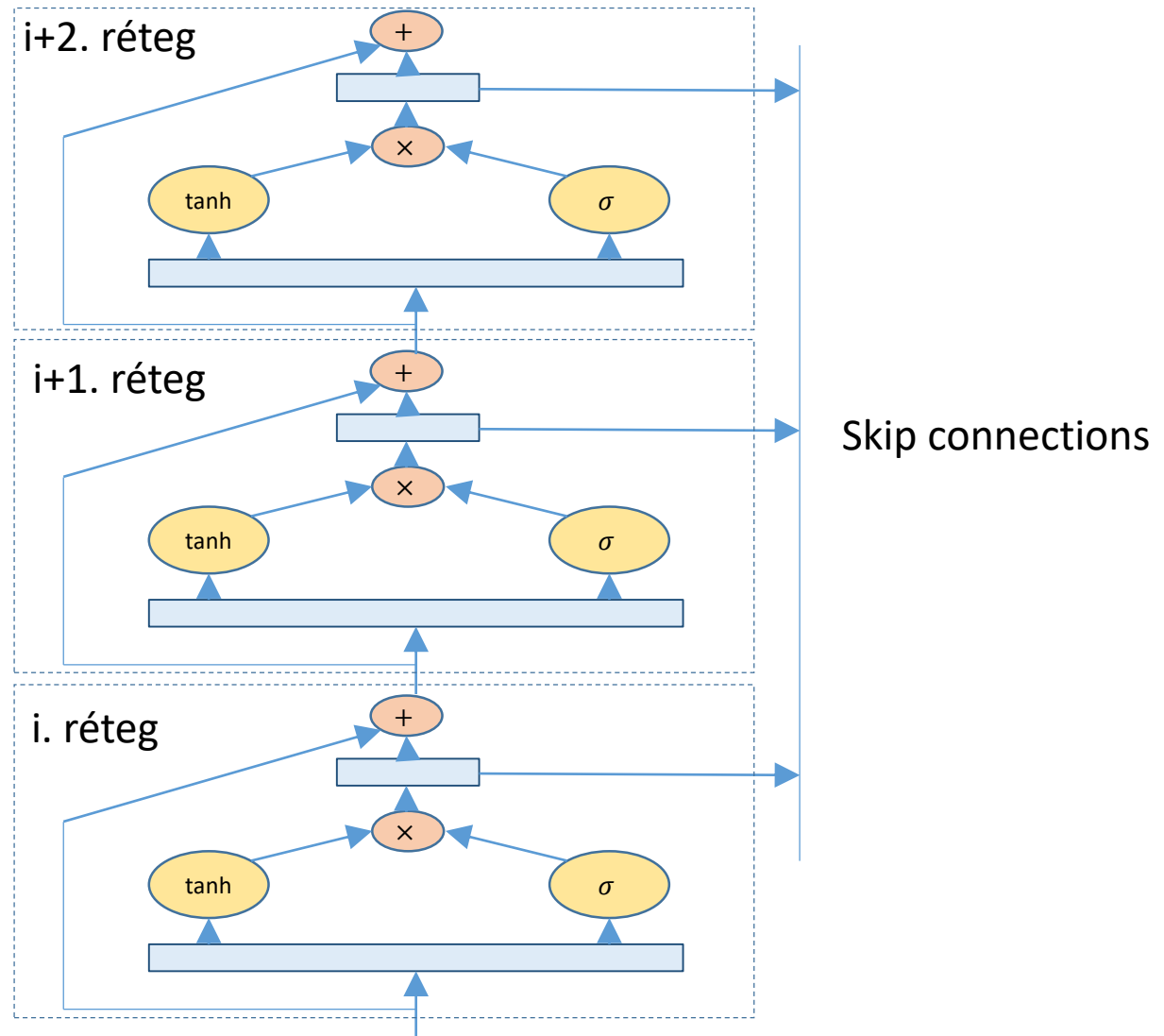
WaveNet

- Layerrek -> Blockok



WaveNet

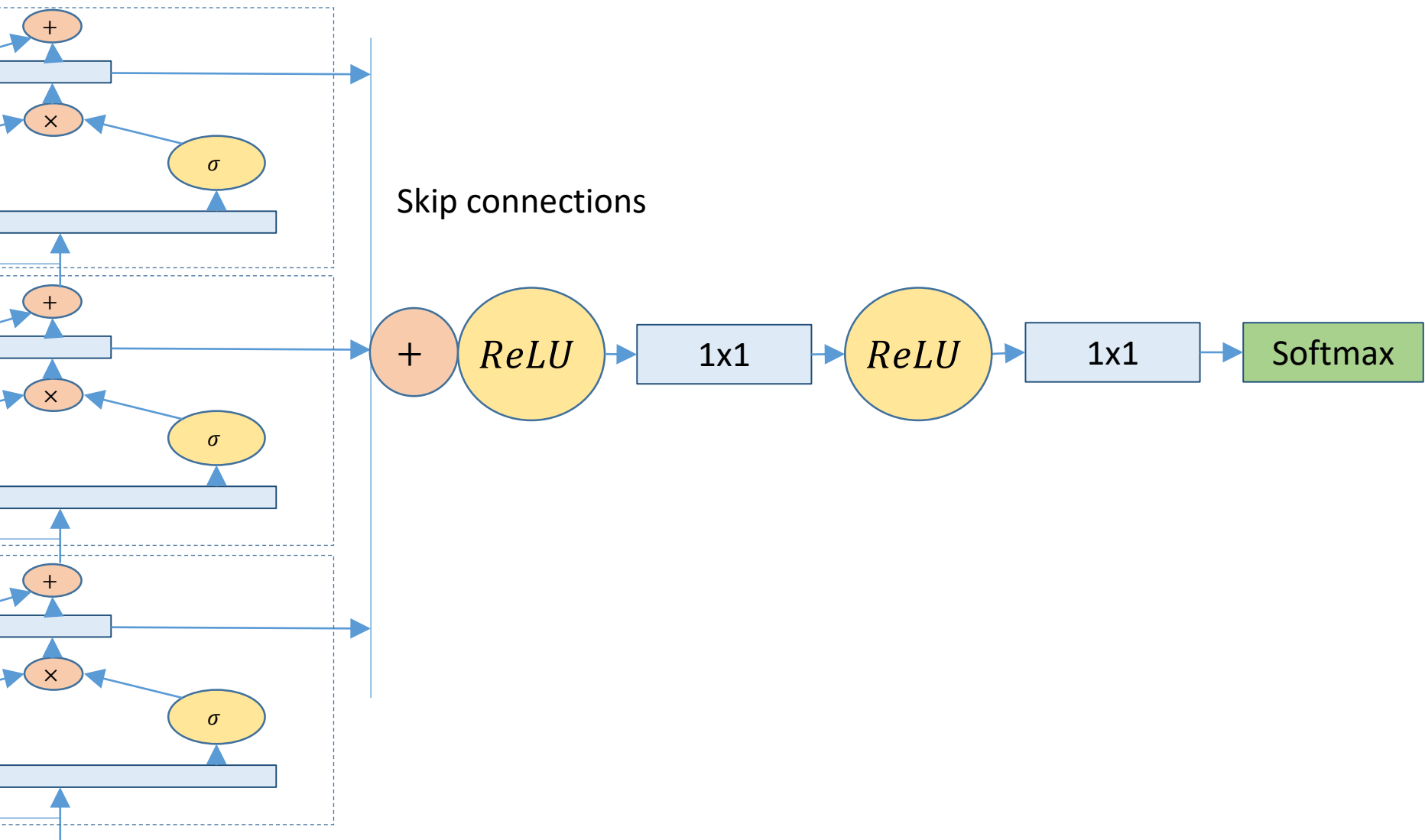
- Layerrek



Dilation:

1,2,4,8,16,32,64,128,256,512,1,2,4,8,16,32,64,128,256,512,1,2,4,8,16,32,64,128,256,512

WaveNet



WaveNet

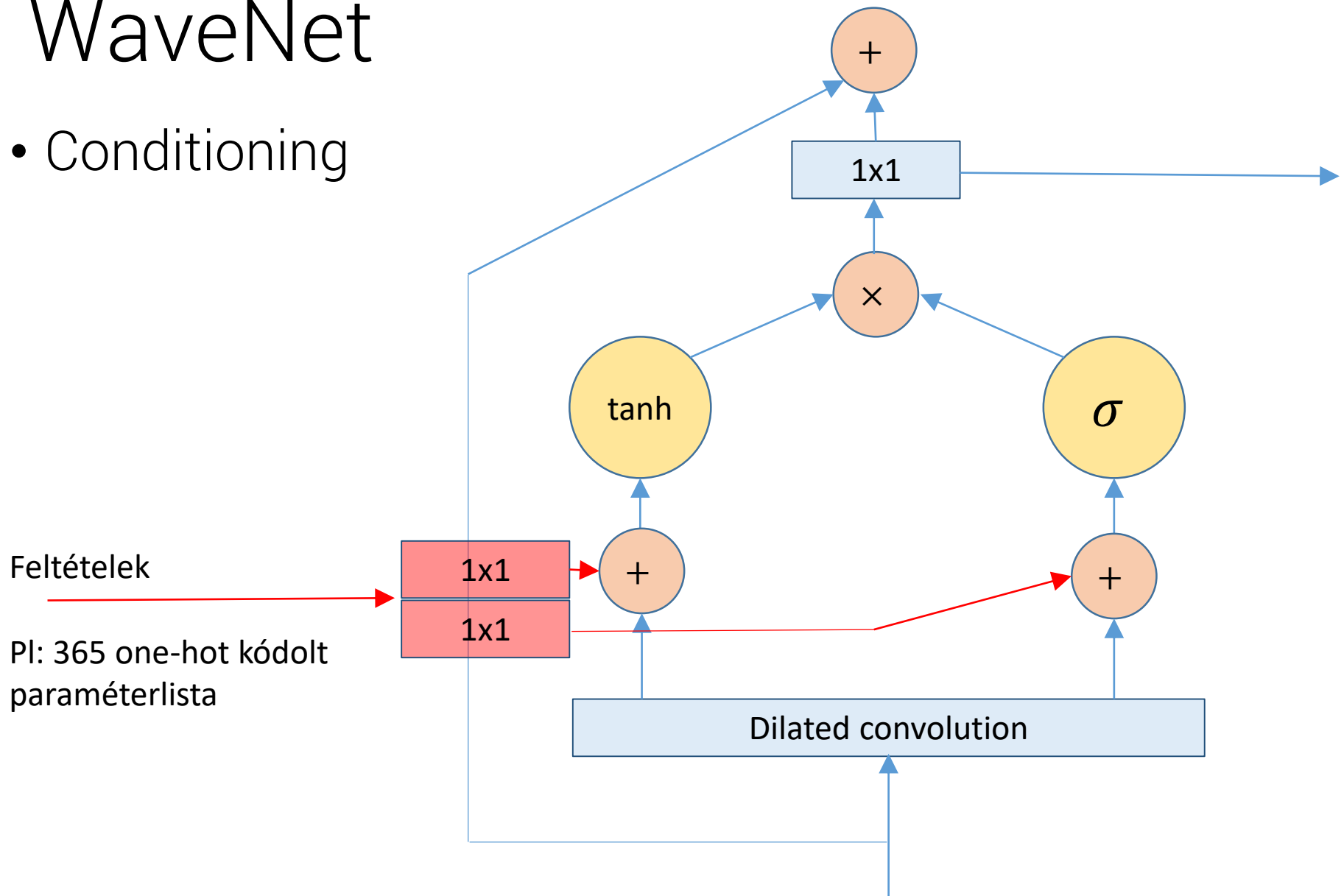
- Példa generálásra:
 - Hangminták: <https://deepmind.com/blog/wavenet-generative-model-raw-audio/>

WaveNet

- Text-to-speech
 - Nyelvi jellemzők
 - Beszédhang
 - F0
 - Hangsúly
 - Mondat, szó, szótag pozíció
 - Beszélő
 - ID
 - Nyelv
 - Stílus
 - Spektrum

WaveNet

- Conditioning



WaveNet

- Generálás:
 - Hangminták: <https://deepmind.com/blog/wavenet-generative-model-raw-audio/>

WaveNet

- Idő
- Tanítási idő:
 - 2-5 mp/iteráció
 - Kb. 100 000 minta (12 Gbyte GPU)
 - 600 000 – 1 000 000 iteráció
 - ~5000 iteráció/epoch -> 120-200 epochs
 - ~7 óra /epoch -> több hét 1 GPU-n
- Generálás:
 - Valós idő: 16000 minta/sec
 - GPU: ~200-300 minta/sec
 - CPU: 2000-3000 minta/sec (Fast WaveNet)
 - (Paralell WaveNet: valós időben vagy gyorsabban)

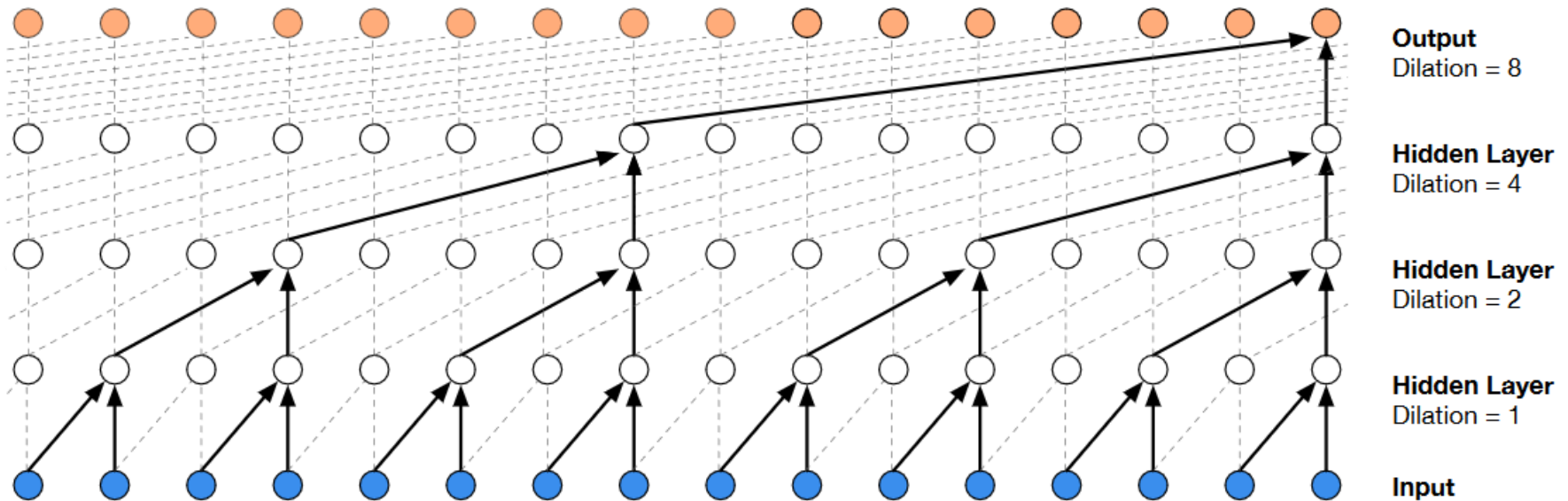
Iteráció / Epoch

- Iteráció \neq Epoch
- Függ a tanító adatbázis méretétől
- Kapcsolat csak a feldolgozás teljes ismeretében lehetséges

- Pl: nem teljes hang feldolgozása

WaveNet

- Fast WaveNet



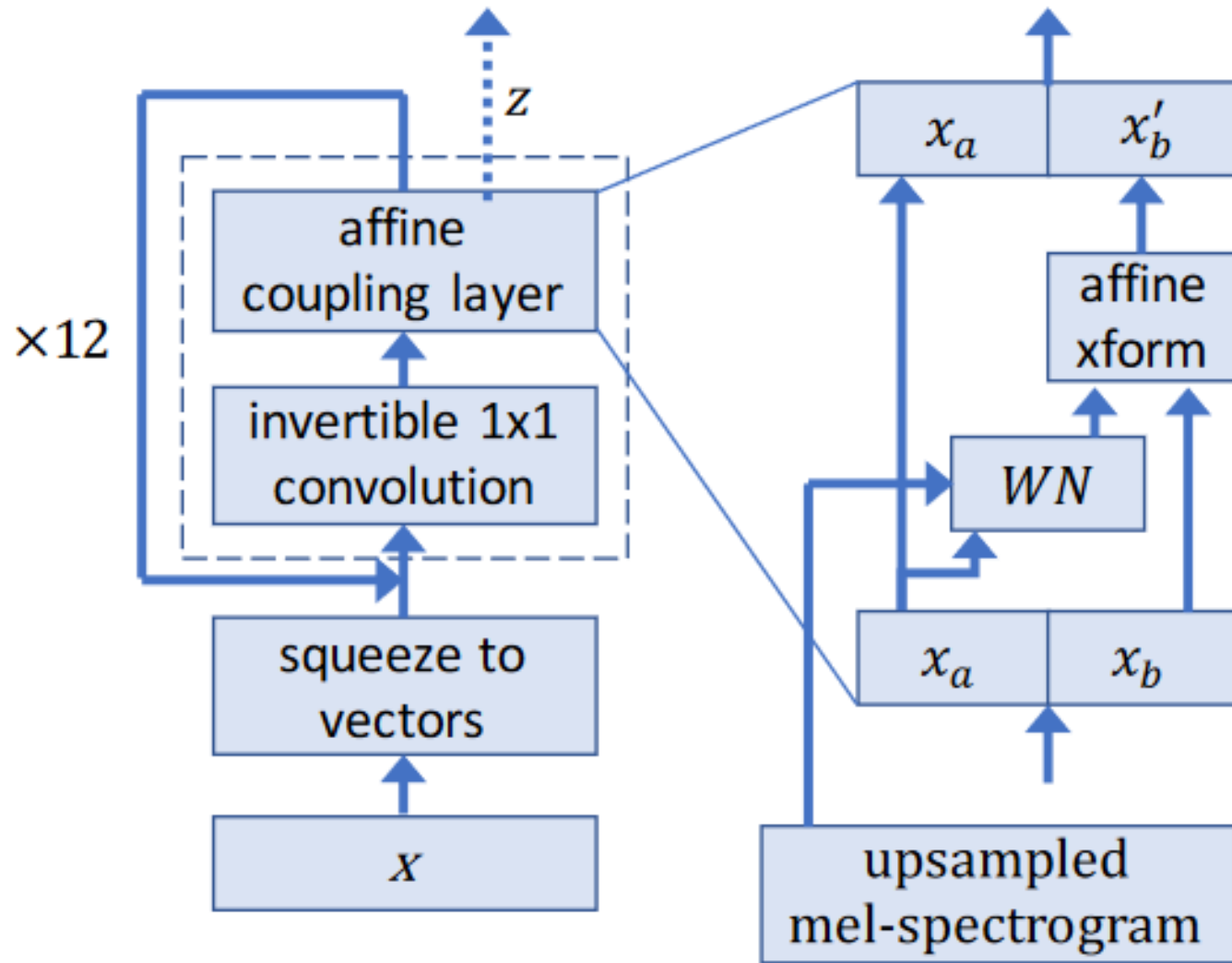
WaveNet

- Párhuzamos WaveNet
 - Teacher-Student

	Módszer	MOS (1-5)
16 kHz, 8-bit, μ -law, 25 óra	LSTM-RNN parametric	3,67
	HMM-driven concatenative	3,86
	WaveNet	4,21
24 kHz, 16 bit, linear PCM, 65 óra	HMM-driven concatenative	4,19
	Autoregressive WaveNet	4,41
	Distilled WaveNet	4,41

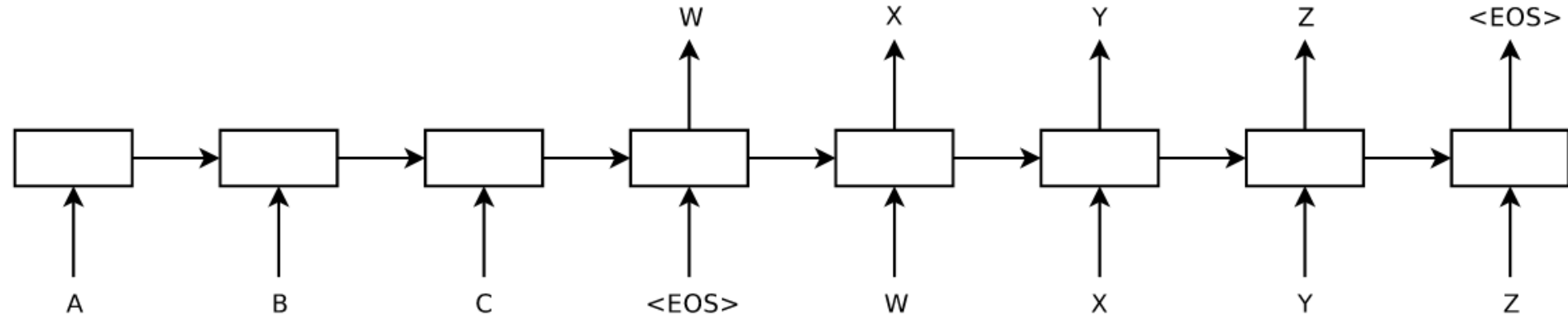
- NVIDIA P100 GPU:
 - Teacher: 172 minta/mp
 - Student: 500 000 minta/mp

WaveGlow



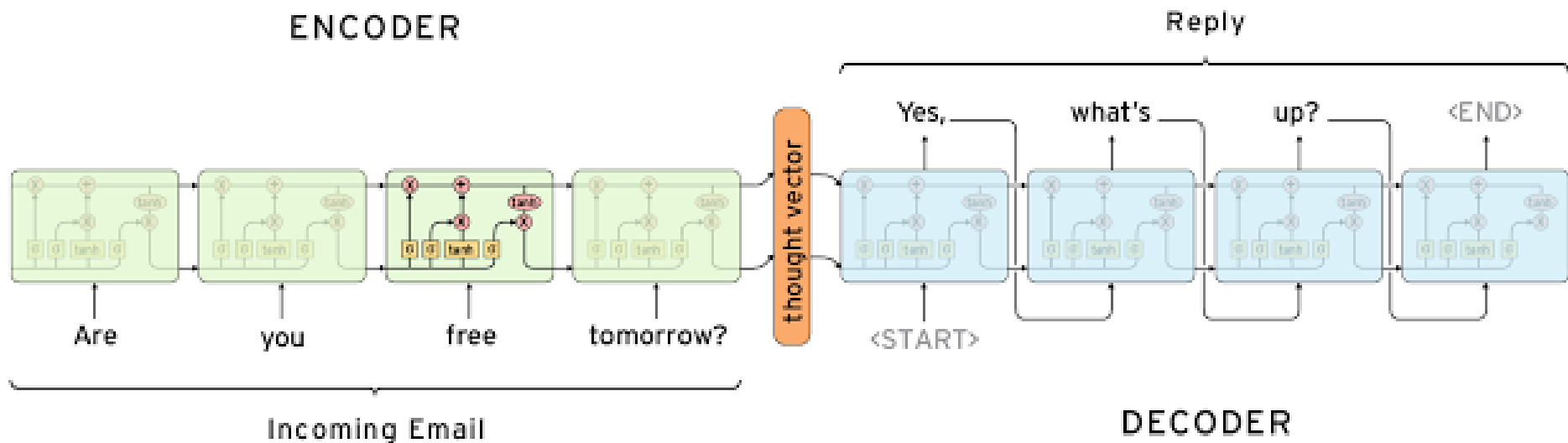
Seq-to-seq

- Encoder-decoder



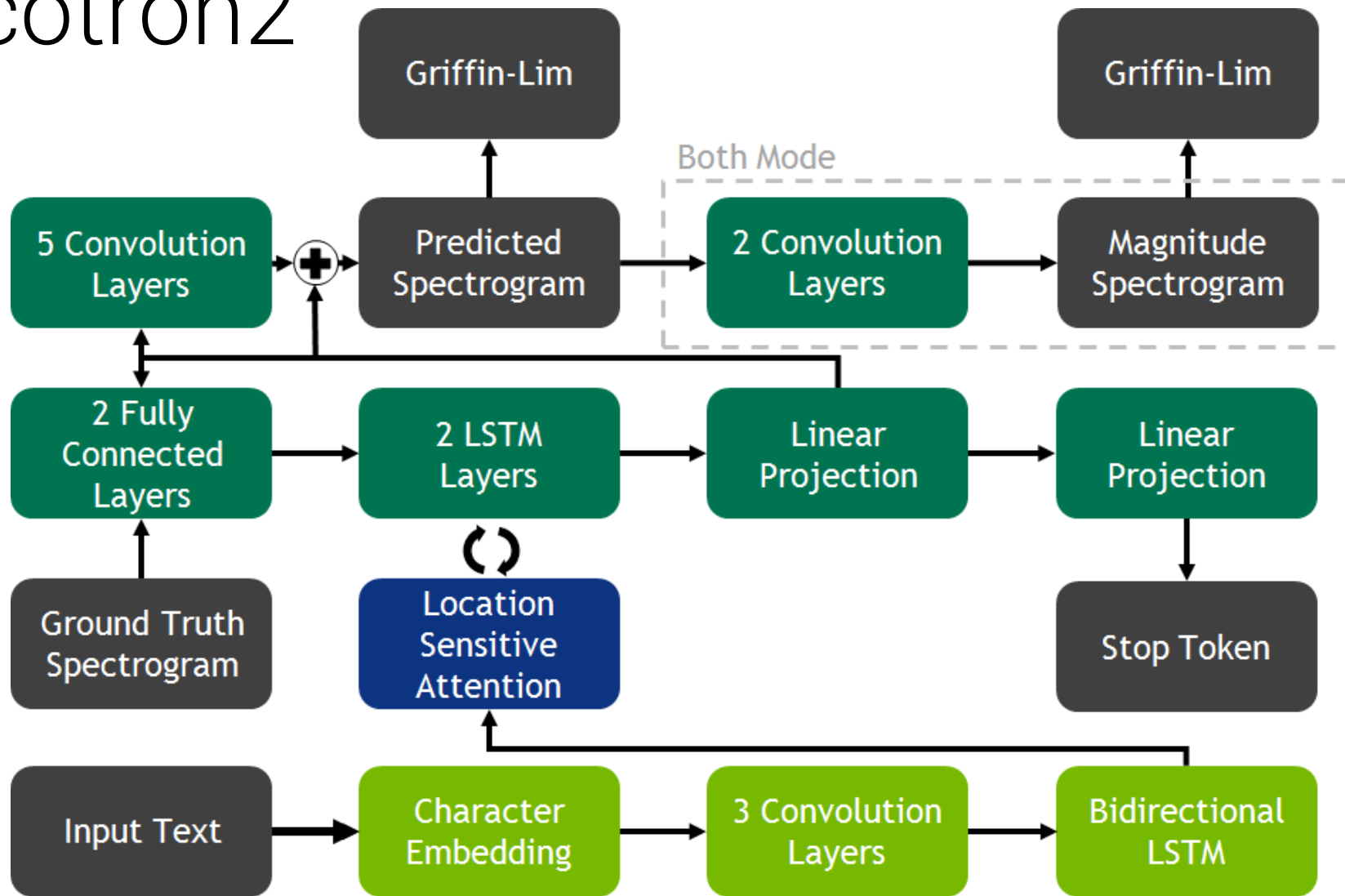
Forrás: Ilya Sutskever et al.: <https://arxiv.org/abs/1409.3215>

Seq-to-seq



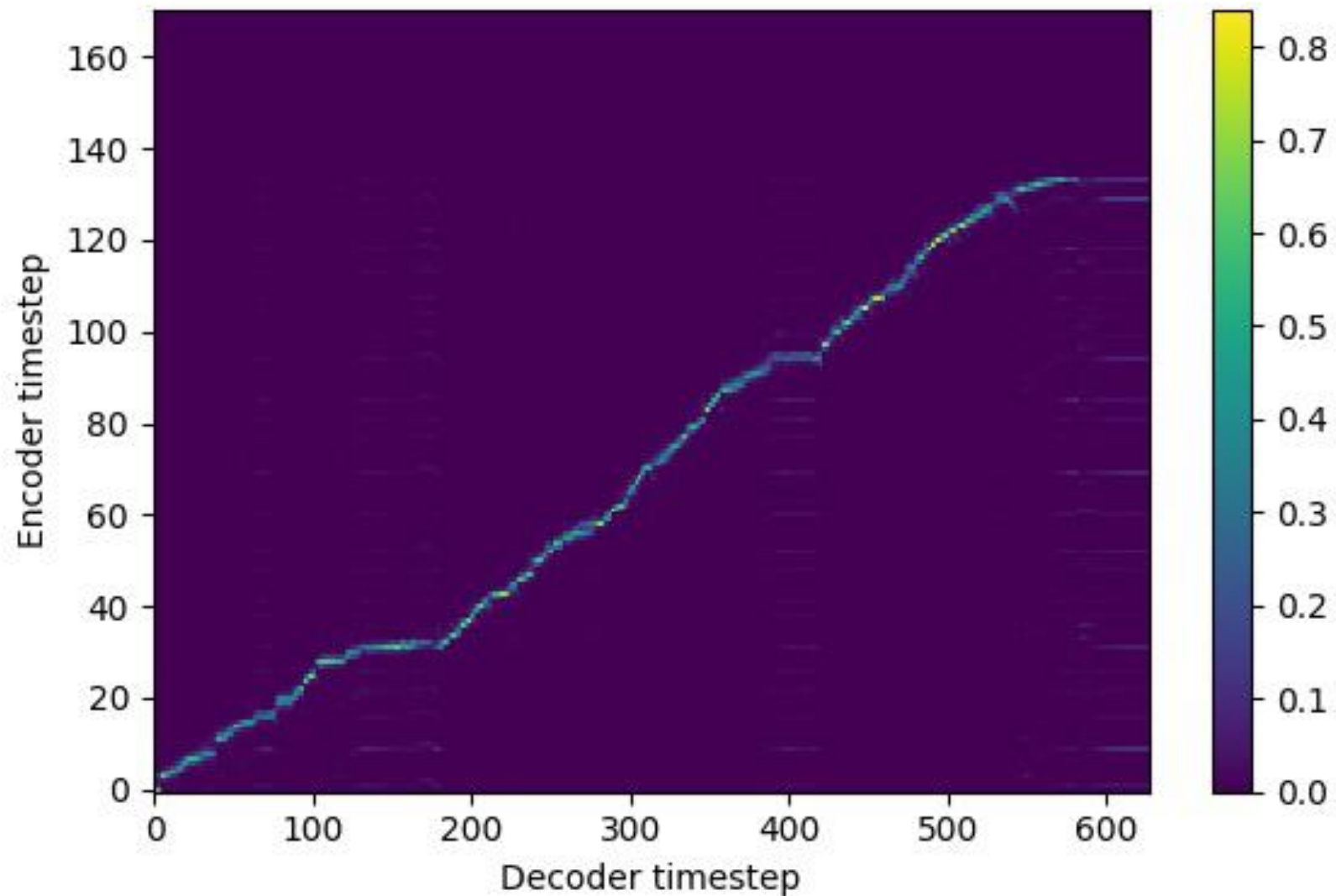
Forrás: <http://www.wildml.com/2016/04/deep-learning-for-chatbots-part-1-introduction/>

Tacotron2



Forrás: <https://nvidia.github.io/OpenSeq2Seq/html/speech-synthesis/tacotron-2.html>

Attention

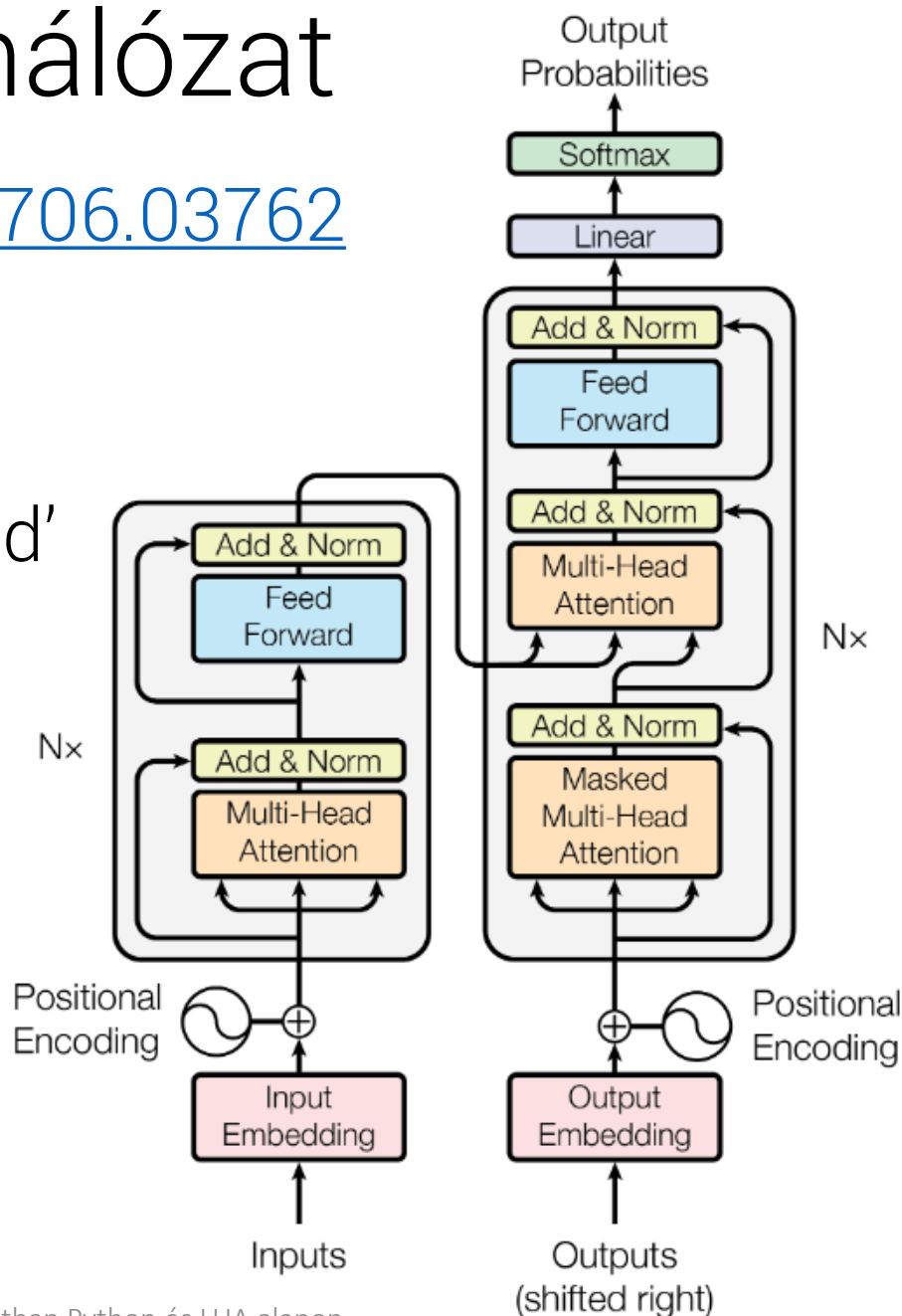


Transzformer hálózat

- <https://arxiv.org/abs/1706.03762>

Vaswani et al.:

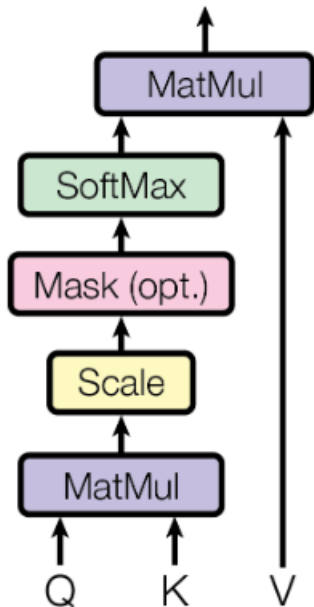
‘Attention Is All You Need’



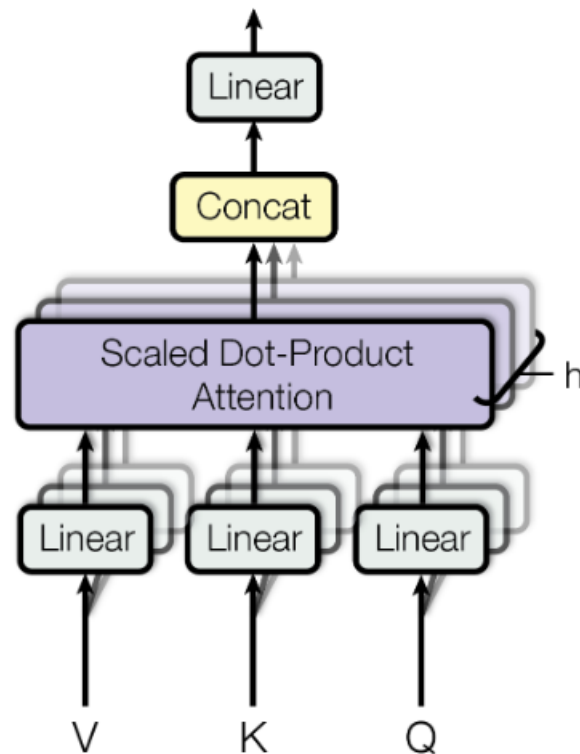
Transzformer hálózatok

- <https://arxiv.org/abs/1706.03762>

Scaled Dot-Product Attention



Multi-Head Attention



Köszönöm a figyelmet!

`zainko@tmit.bme.hu`