Convolutions in the world of sequences

Alexey Zaytsev

Skoltech

Main types of models for sequential data

1. Recurrent Neural Networks

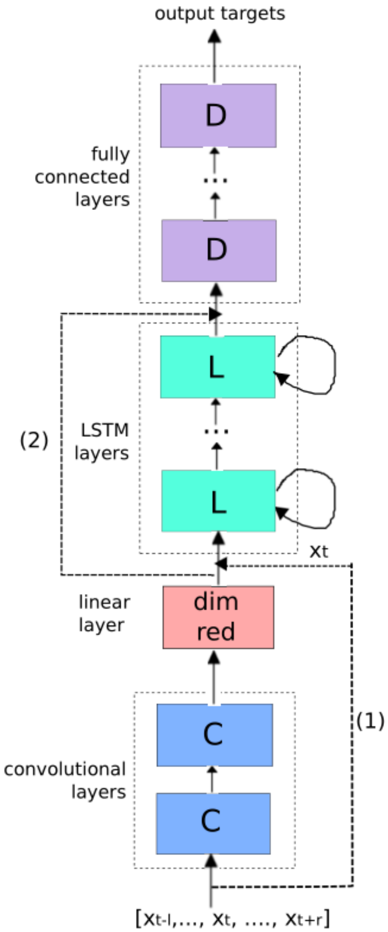
2. **One-dimensional convolutional neural networks**

3. Transformers

Skoltech

Sequential data models for acoustic dat processing

Model	WER ↓
FCNN	18.4
CNN	18.0
LSTM	18.0
CNN+LSTM	17.6
LSTM+FCNN	17.6
CNN+LSTM+FCNN	**17.3**



CNN+LSTM+FCNN architecture

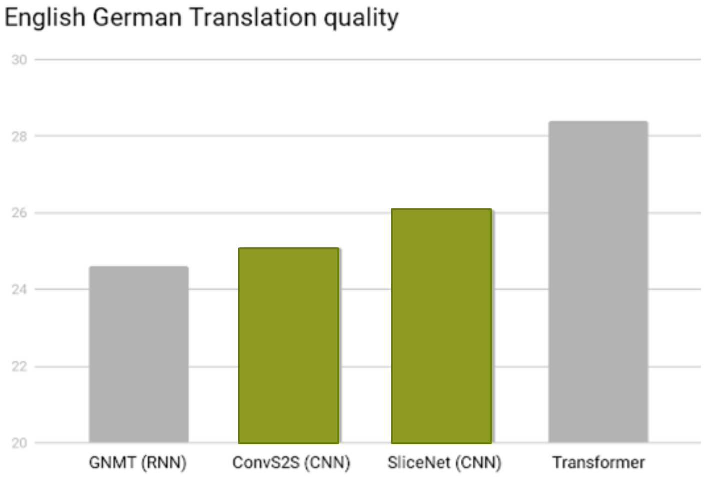
Sainath, Tara N., et al. "Convolutional, long short-term memory, fully connected deep neural networks." *IEEE ICASSP*. 2015.

Skoltech

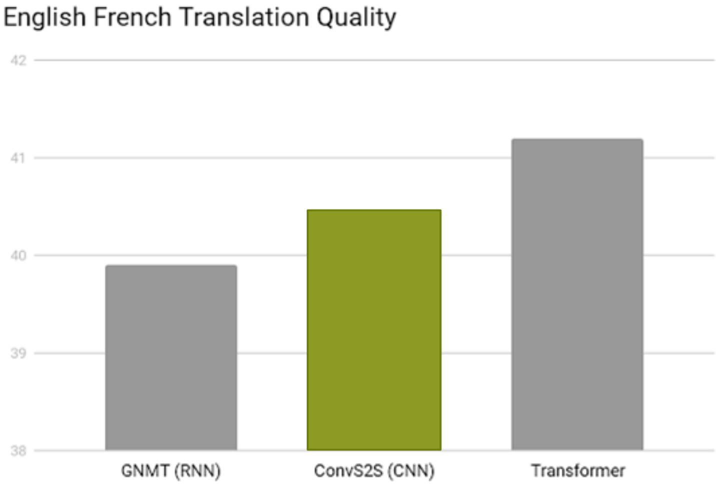
Motivation for CNNs

CNNs are better than RNNs for Machine Translation

BLEU



BLEU



We maximize BLEU

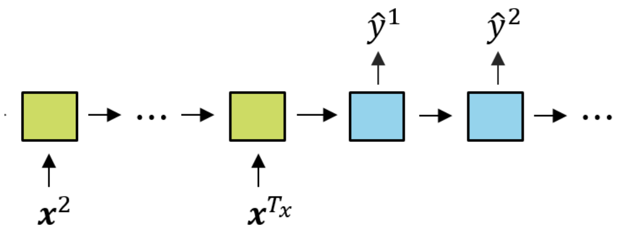
Image source: https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

What is wrong with RNNs?

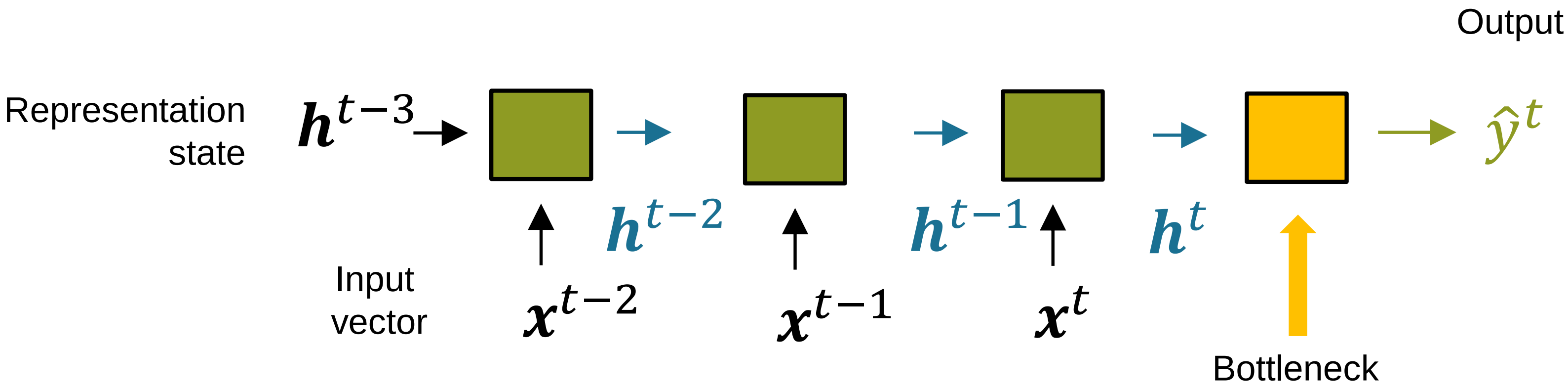
A. The bottleneck for the last token

B. Goes from past to future. *Bidirectional RNNs can help there*

C. Long time to train with GPU



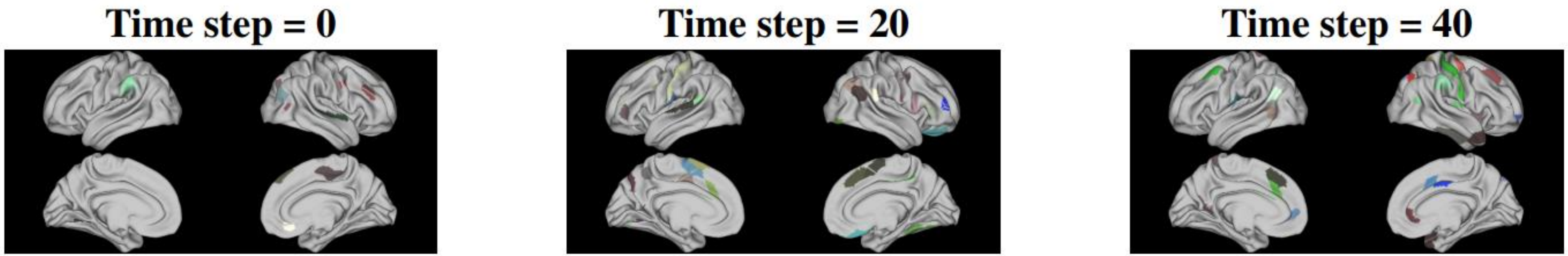
The bottleneck is more evident for seq2seq problems



Skoltech

What is wrong with RNNs?

A. Most important part is the end



Ismail, Aya Abdelsalam, et al. "Input-cell attention reduces vanishing saliency of recurrent neural networks." *NeurIPS. 2019*.

CNN idea

1. Compute embeddings for all subsequences of a certain length
2. Combine them in a meaningful way

Example: “Now I need a place to hide away” has vectors related to:

“now I need”, “need a place”, “a place to”, “place to hide”, “to hide away”

Issues:

- No grammatical sense
- Approach not backed by linguistics

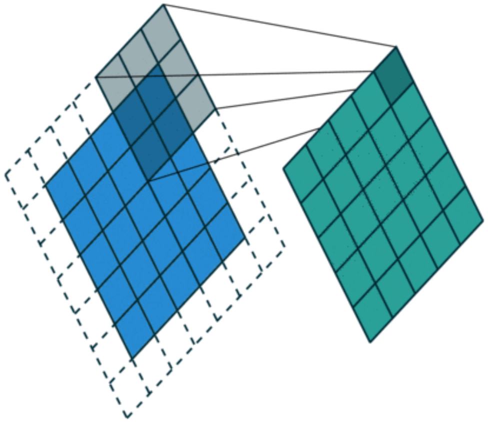
Skoltech

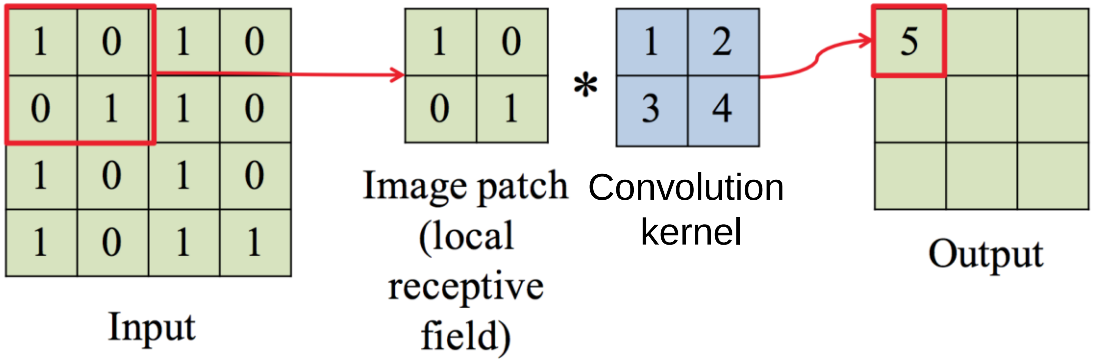
1D CNN architecture

Skoltech

“Classic” 2D convolution for images

Input Output

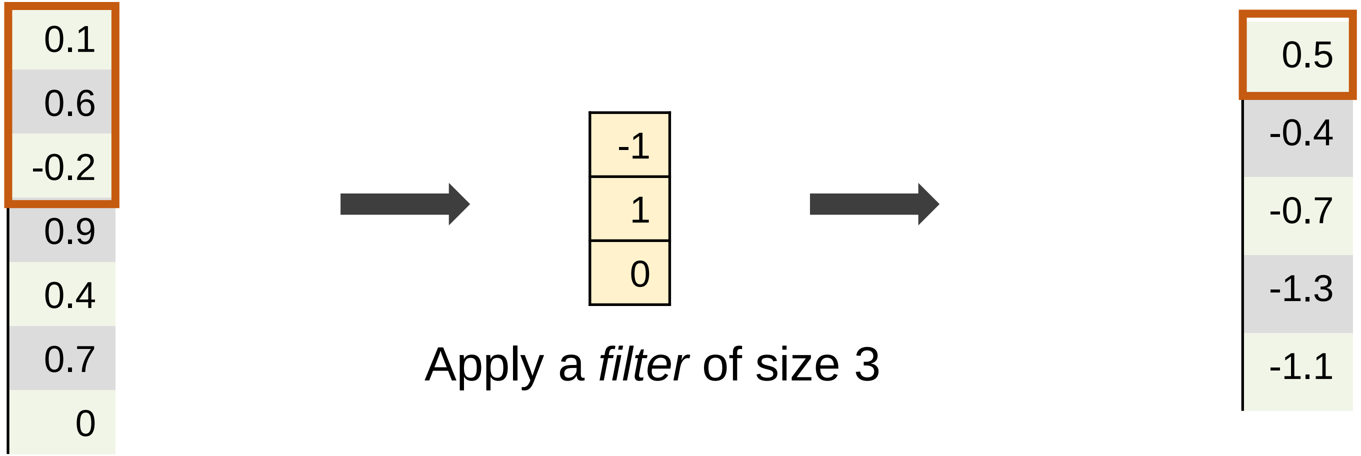




Definition of 1D convolution

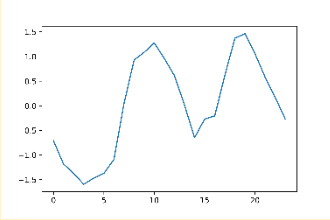
$$\tilde{x}_t = \sum_{i=-h}^{h} x_{t+i} w_i$$

convolution of size $(2h+1)$, t is from 1 to T
$\mathbf{w}$ is the convolution kernel, $\mathbf{x}$ is input data, $\tilde{\mathbf{x}}$ is input data



1D CNN at home: ROCKET

Input: 1D time series



$O(knTh)$

Convolved time series for 10K random kernels

$O(knTh)$

Max and percentage of positive values

$O(n(2k)^2)$ or $O(n^2 2k)$

Ridge regression (or Logistic regression) as a classifier

k – number of kernels
n – number of examples
T – length of the time series
h – kernel width

Skoltech

Dempster, Angus et al. "ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels." *Data Mining and Knowledge Discovery. 2020*.

ROCKET and Mini ROCKET hyperparameters

	Rocket	MiniRocket
length	$\{7, 9, 11\}$	9
weights	$\mathcal{N}(0, 1)$	$\{-1, 2\}$
bias	$\mathcal{U}(-1, 1)$	from convolution output
dilation	random	fixed (rel. to input length)
padding	random	fixed
features	PPV + max	PPV
num. features	20K	10K

Skoltech

Dempster, Angus, Daniel F. Schmidt, and Geoffrey I. Webb. "MiniROCKET: A very fast (almost) deterministic transform for time series classification." ACM SIGKDD. 2021.

1D convolution for tokens

$$\tilde{x}_t = \sum_{i=-h}^{h} \sum_{j=1}^{k} x_{t-i,j} w_{i,j}$$

convolution of size $(2h+1) \times k$, j is from 1 to T
$\mathbf{w}$ is the convolution kernel, $\mathbf{x}$ is input data, $\tilde{\boldsymbol{x}}$ is input data

now	0.1	-0.2	0.4
I	0.6	0.2	0.1
need	-0.2	0.1	0.0
a	0.9	0.1	0.2
place	0.4	0.2	0.6
to	0.7	-1.2	1
hide	0	0.5	0.5

→

3	2	-1
1	1	1
2	-1	0

→

n,I,n	-0.1
i,n,a	3.3
n,a,p	1.4
a,p,t	3.7
p,t,h	1.1

Apply a *filter* of size 3

Skoltech

1D convolution for tokens with padding

pad	0	0	0
now	0.1	-0.2	0.4
I	0.6	0.2	0.1
need	-0.2	0.1	0.0
a	0.9	0.1	0.2
place	0.4	0.2	0.6
to	0.7	-1.2	1
hide	0	0.5	0.5
pad	0	0	0

3	2	-1
1	1	1
2	-1	0

Apply a *filter* or *kernel* of size 3

P,n,I	1.3
n,I,n	-0.1
i,n,a	3.3
n,a,p	1.4
a,p,t	3.7
p,t,h	1.1
t,h,P	-0.3

Skoltech

1D convolution for tokens with max pooling

now	0.1	-0.2	0.4
I	0.6	0.2	0.1
need	-0.2	0.1	0.0
a	0.9	0.1	0.2
place	0.4	0.2	0.6
to	0.7	-1.2	1
hide	0	0.5	0.5

3	2	-1
1	1	1
2	-1	0

Apply a *filter* or *kernel* of size 3

n,I,n	-0.1
i,n,a	3.3
n,a,p	1.4
a,p,t	3.7
p,t,h	1.1

Max pool	3.7

1D convolution for tokens with mean pooling

now	0.1	-0.2	0.4
I	0.6	0.2	0.1
need	-0.2	0.1	0.0
a	0.9	0.1	0.2
place	0.4	0.2	0.6
to	0.7	-1.2	1
hide	0	0.5	0.5

3	2	-1
1	1	1
2	-1	0

Apply a *filter* or *kernel* of size 3

n,I,n	-0.1
i,n,a	3.3
n,a,p	1.4
a,p,t	3.7
p,t,h	1.1

Mean pool	1.88

1D convolution for tokens with many kernels and mean pooling

now	0.1	-0.2	0.4
I	0.6	0.2	0.1
need	-0.2	0.1	0.0
a	0.9	0.1	0.2
place	0.4	0.2	0.6
to	0.7	-1.2	1
hide	0	0.5	0.5

3	2	-1
1	1	1
2	-1	0

2	2	-1
1	1	1
2	-1	0

Apply many *filters* or *kernel* of size 3

n,I,n	-0.1	-0.2
i,n,a	3.3	2.7
n,a,p	1.4	1.6
a,p,t	3.7	2.8
p,t,h	1.1	0.7

Mean pool	1.88	1.52

Skoltech

PyTorch 1D convolution summary

```
batch_size = 16
word_embed_size = 4
seq_len = 7
input = torch.randn(batch_size, word_embed_size, seq_len)
conv1 = Conv1d(in_channels=word_embed_size, out_channels=3,
               kernel_size=3)  # can add: padding=1
hidden1 = conv1(input)
hidden2 = torch.max(hidden1, dim=2)  # max pool
```

Skoltech

Other things to do: stride, pooling, dilation, multiple layers

- Dilation: increase receptive field
- Multiple convolutional layers: more layers
- Stride: stride=2 – skip odd triplets
- Max pooling of stride
- Top k-max pooling

now	0.1	-0.2	0.4
I	0.6	0.2	0.1
need	-0.2	0.1	0.0
a	0.9	0.1	0.2
place	0.4	0.2	0.6
to	0.7	-1.2	1
hide	0	0.5	0.5

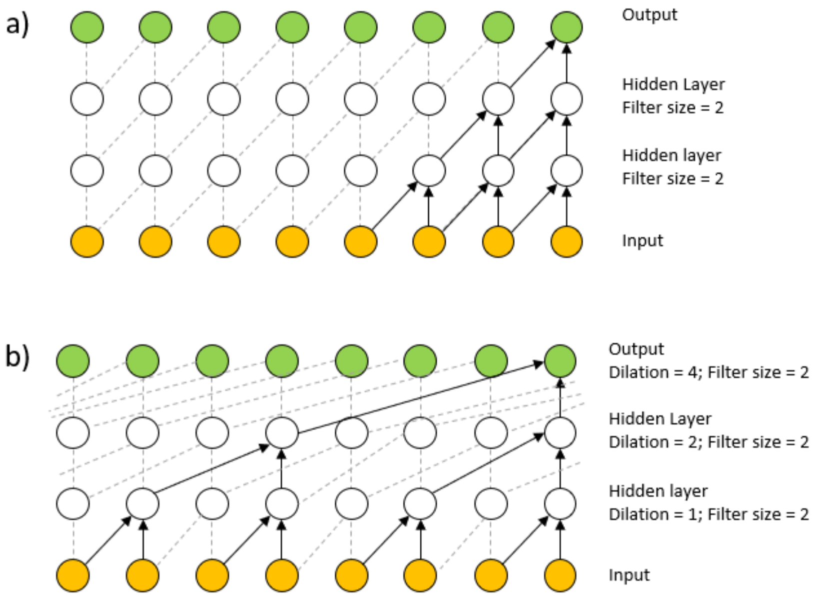


Figure source:
Benson, B., et al. "Forecasting solar cycle 25 using deep neural networks." *Solar Physics* 295 (2020): 1-15.

Skoltech

Single Layer CNN for sentence classification

- Simple: one convolutional layer + pooling
- Word vectors $\mathbf{x}_\mathrm{i} \in R^k$
- Sentence $\mathbf{x}_{0:n} = [\mathbf{x}_0, \mathbf{x}_2, \dots, \mathbf{x}_{n-1}]$
- Concatenation of words $\mathbf{x}_{i:i+h}$
- Convolutional filter $\mathbf{w} \in R^{hk}$
- Filters can be of different size

Skoltech

Kim, Yoon. "Convolutional neural networks for sentence classification." *arXiv preprint arXiv:1408.5882* (2014).

Single Feature Inference

- Convolutional filter $\mathbf{w} \in R^{hk}$ applied to all possible windows
- Feature computation

$$c_{\mathrm{i}} = \mathrm{f}(\mathbf{w}^{\mathbf{T}}\mathbf{x}_{i:i+h} + b)$$

- Result: a feature map

$$\mathbf{c} = [c_o, c_2, \dots, c_{n-h}] \in R^{n-h+1}$$

Skoltech

Pooling and channels

- Idea: capture the most important information: maximum over time
- Feature map $\mathbf{c} = [c_o, c_2, \dots, c_{n-h}]$
- Pooled single number $\tilde{c} \; = \max(\boldsymbol{c})$
- Use multiple filters with different weights, different windows sizes
- For max pooling length is irrelevant

Skoltech

Full network architecture

- One convolution followed by max-pooling
- Feature map $\mathbf{z} = [\widetilde{c_1}, \dots, \widetilde{c_m}]$
- Final softmax layer

$$y = \mathrm{softmax}(\mathrm{W}\mathbf{z} + \mathrm{b})$$

Skoltech

Multiple-channel input idea

- Initialize with pre-trained word vectors
- Start with two similar instances
- Change one instance, freeze other
- Both channels are added before max-pooling

Regularization

Dropout!

- During *training*: Bernoulli random variable to drop with probability p features
- During *inference*: Multiply parameters matrix W by p

Constrain L2 norms for weights vector for each class:

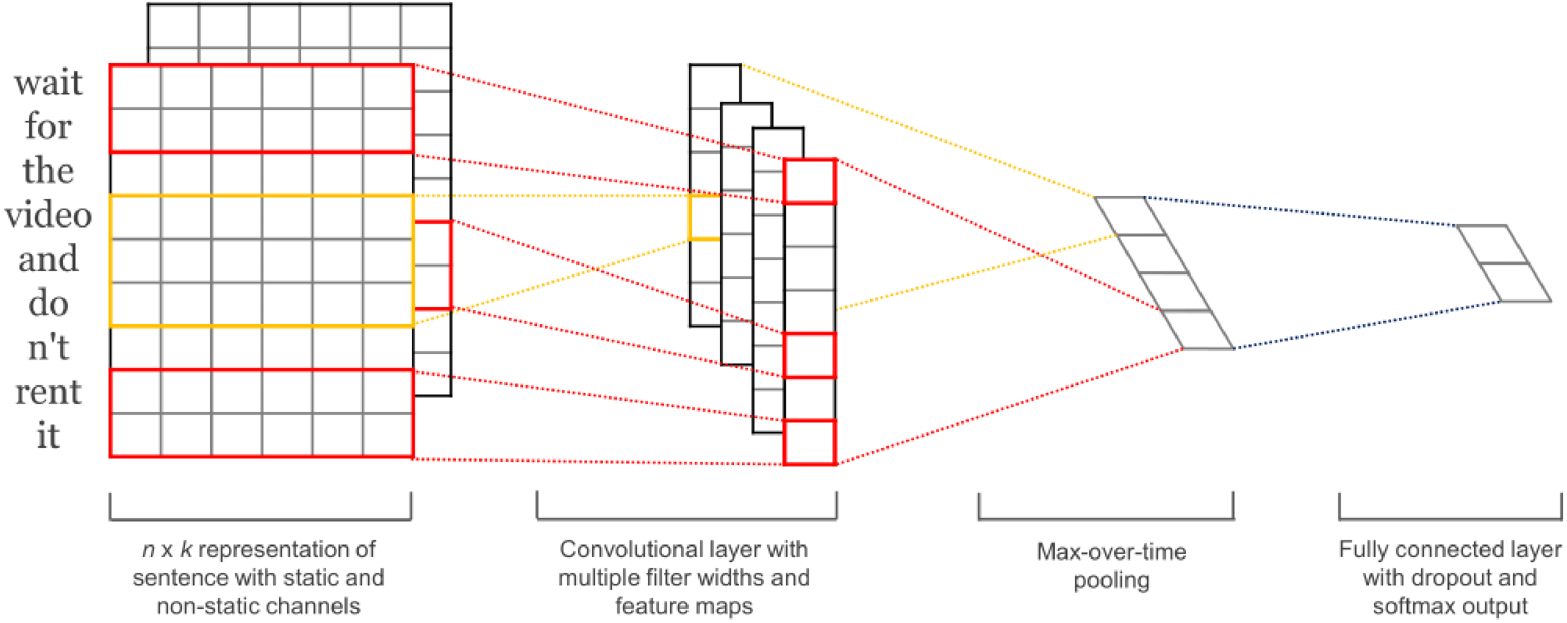
- Row norm for each row is smaller than s
- If bigger – rescale to meet the constraint
- Not very common (may be unnecessary)

Hyperparameters

- ReLU activation
- Window filter sizes h = 3, 4, 5
- Each filter size: 100 feature maps
- Dropout 2-4% accuracy improvement via dropout
- L2 constraint *s* for rows of softmax, *s=3*
- Mini-batch size for SGD training: 5
- Word2vec with k=300 features
- Use validation set error as a stopping criterion

Skoltech

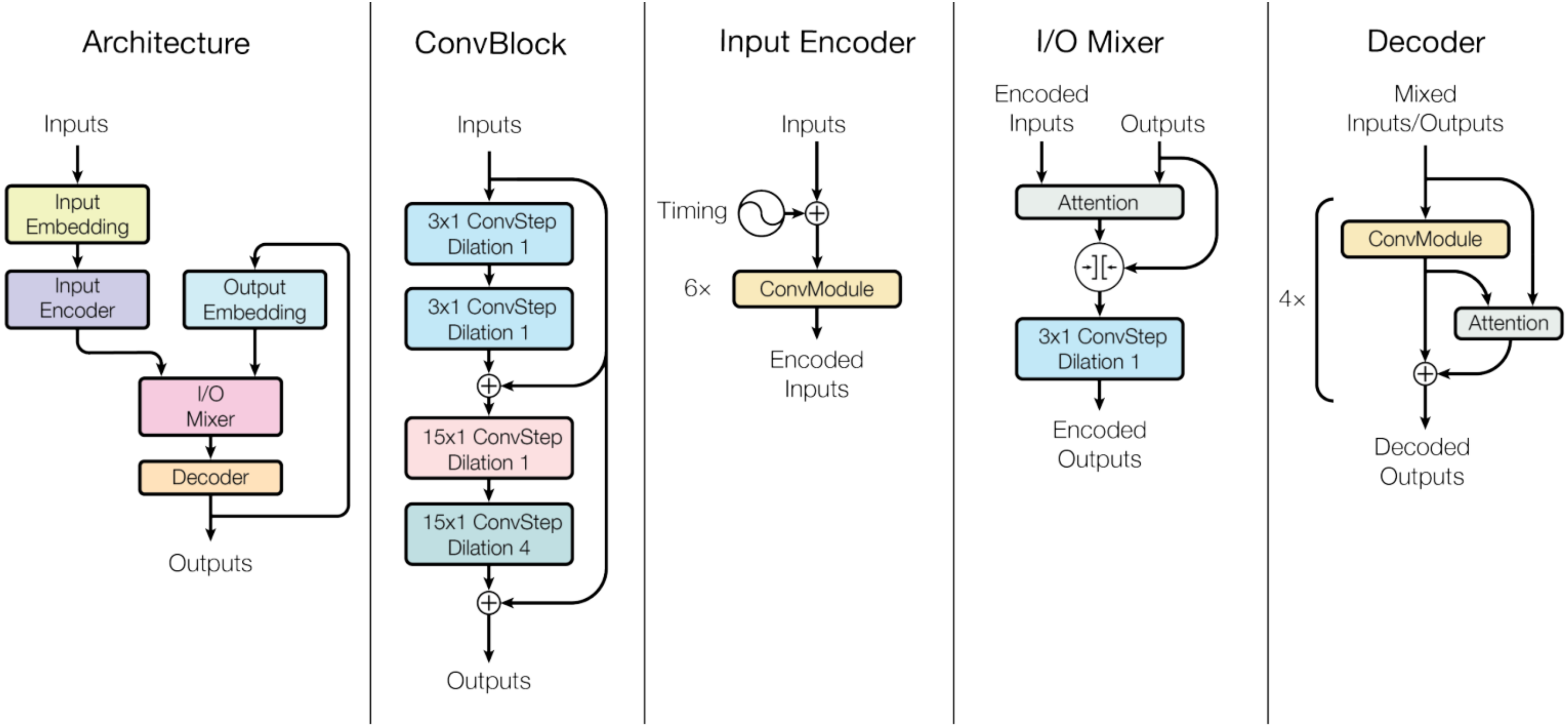
CNN overall architecture from Yoon Kim [2014]



Skoltech

Kim, Yoon. "Convolutional neural networks for sentence classification." *arXiv preprint arXiv:1408.5882* (2014).

CNNs with fewer parameters

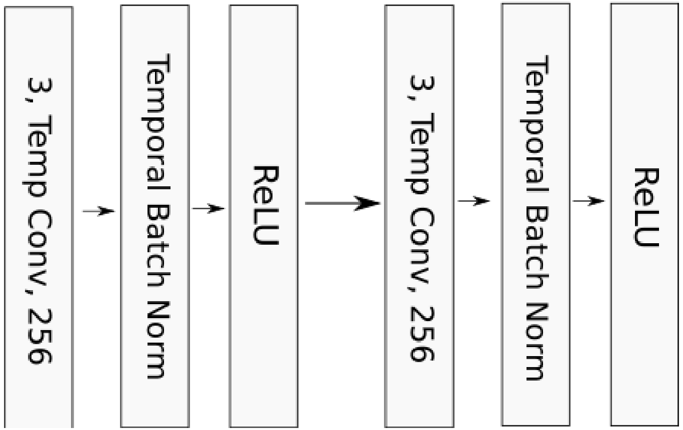


Kaiser, Lukasz, Aidan N. Gomez, and Francois Chollet. "Depthwise separable convolutions for neural machine translation." ICLR. 2018.

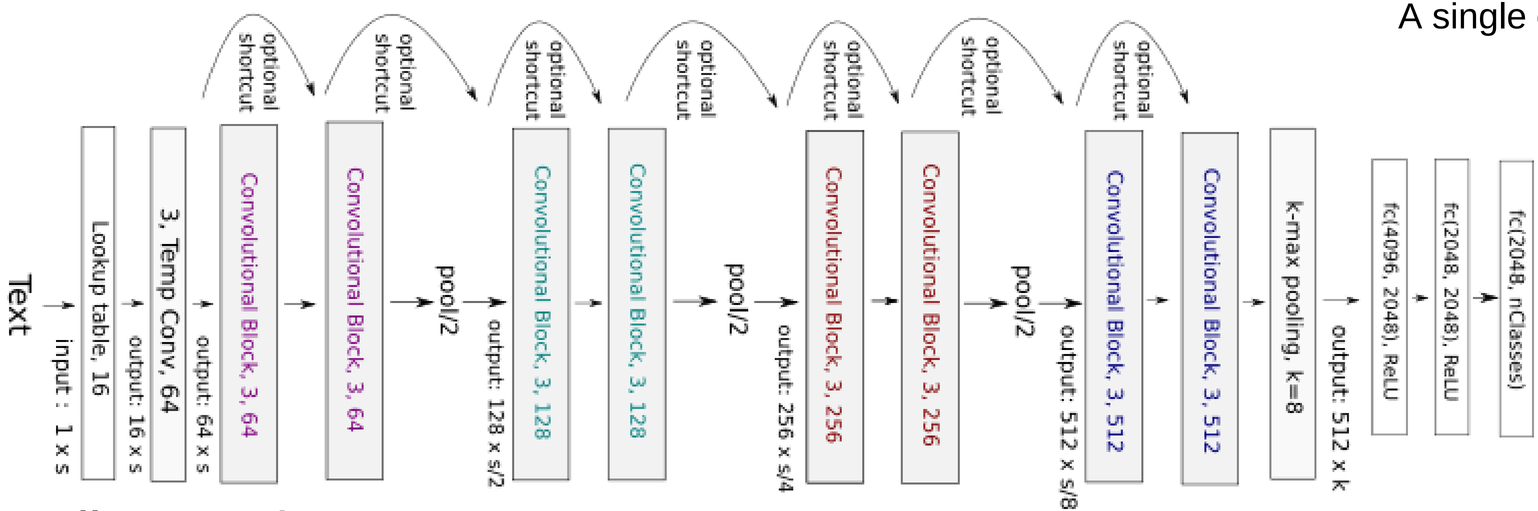
Skoltech

Very deep CNNs for text classification

Brings ideas from ResNet + character level



A single convolutional block



Full network

Conneau, A., Schwenk, H., Barrault, L., & Le Cun, Y. Very deep convolutional networks for text classification. EACL. 2017

Skoltech

WaveNet: a generative model for audio

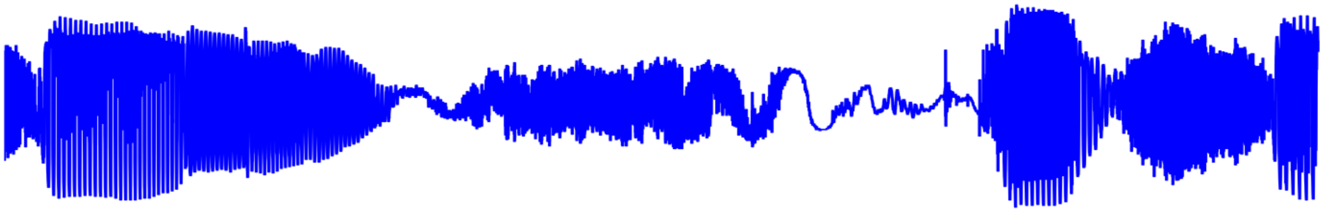
van den Oord, Aäron, et al. "WaveNet: A Generative Model for Raw Audio." *9th ISCA Speech Synthesis Workshop*. 2016

Skoltech

WaveNet key features

WaveNet, an audio generative model based on the PixelCNN architecture:

- SOTA in 2016 for the text-to-speech problem
- New architecture with dilated convolutions
- The main ingredient is causal convolutions



A second of generated speech

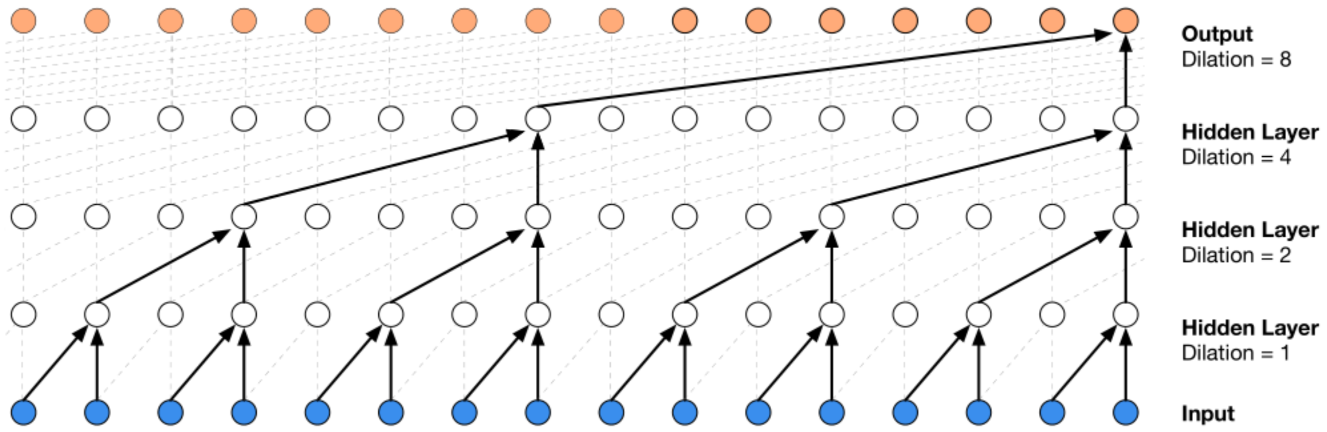
van den Oord, Aäron, et al. "WaveNet: A Generative Model for Raw Audio." *9th ISCA Speech Synthesis Workshop*. 2016

Skoltech

WaveNet model

Causal convolutions looks only in the past and model

$p(x^t|x^1, x^2, \dots, x^{t-1})$



Dilated convolutions allow to make the receptive field larger

Skoltech

van den Oord, Ääron, et al. "WaveNet: A Generative Model for Raw Audio." *9th ISCA Speech Synthesis Workshop*. 2016

WaveNet convolutions

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

Convolutions with sigmoid gate

$$\mathbf{z} = \tanh(W_{f,k} * \mathbf{x}) \odot \sigma(W_{g,k} * \mathbf{x})$$

Convolutions with global context ***h***

$$\mathbf{z} = \tanh\left(W_{f,k} * \mathbf{x} + V_{f,k}^T \mathbf{h}\right) \odot \sigma\left(W_{g,k} * \mathbf{x} + V_{g,k}^T \mathbf{h}\right)$$

Convolutions with local context ***y***

$$\mathbf{z} = \tanh(W_{f,k} * \mathbf{x} + V_{f,k} * \mathbf{y}) \odot \sigma(W_{g,k} * \mathbf{x} + V_{g,k} * \mathbf{y})$$

Prediction: softmax over quantized outputs

Skoltech

van den Oord, Aäron, et al. "WaveNet: A Generative Model for Raw Audio." *9th ISCA Speech Synthesis Workshop*. 2016

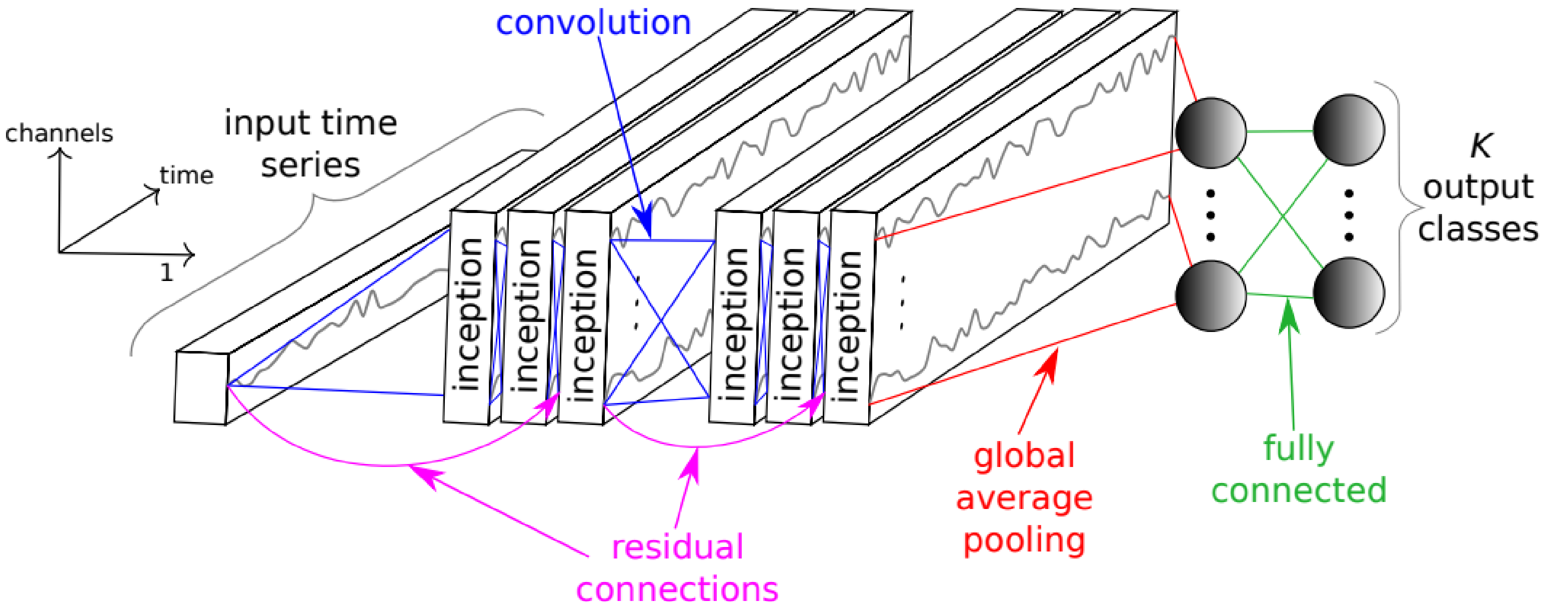
InceptionTime: 1D CNN for time series classification

Ismail Fawaz, Hassan, et al. "InceptionTime: Finding AlexNet for time series classification." *Data Mining and Knowledge Discovery.* 2020.

Skoltech

InceptionTime architecture

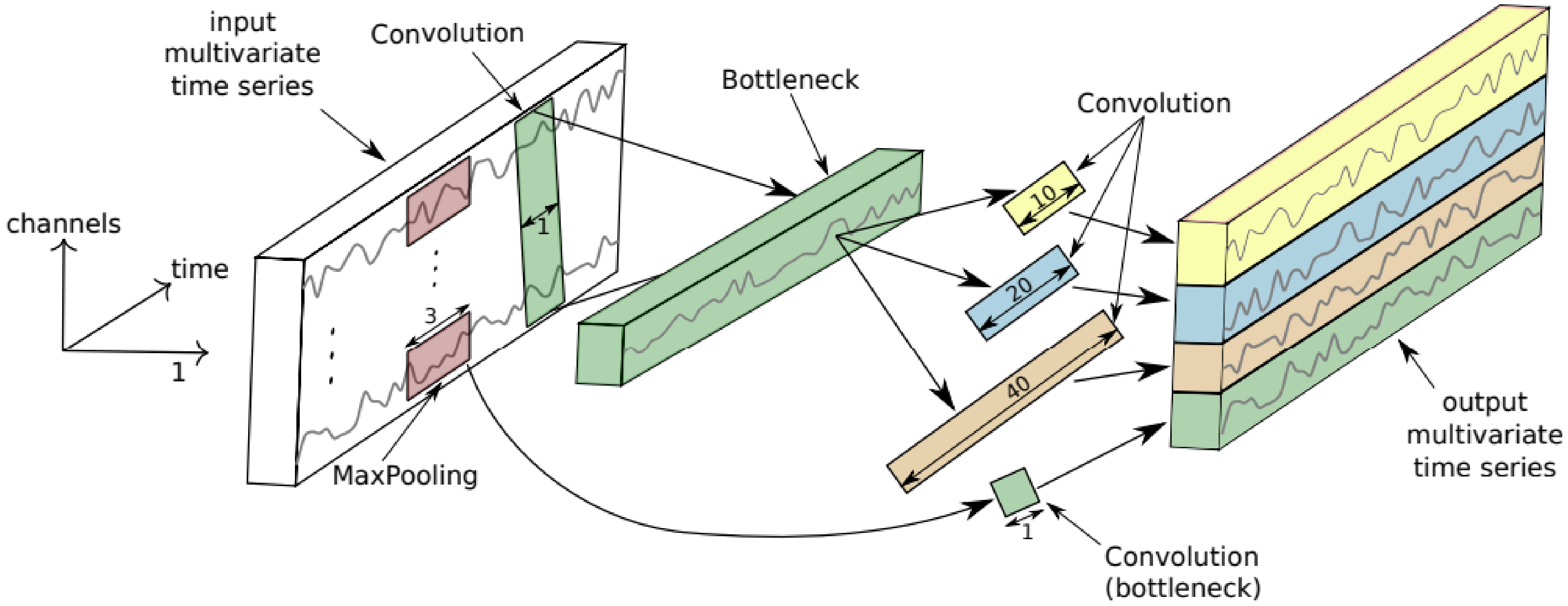
Goal: time series classification, one label for a series,
Model: ensemble of 5 independent neural networks



Ismail Fawaz, Hassan, et al. "InceptionTime: Finding AlexNet for time series classification." *Data Mining and Knowledge Discovery.* 2020.

InceptionTime block

…similar to ResNet



Ismail Fawaz, Hassan, et al. "InceptionTime: Finding AlexNet for time series classification." *Data Mining and Knowledge Discovery.* 2020.

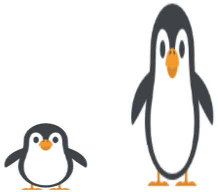
Conclusions

Skoltech

Conclusions

- CNN is a powerful tool for sequential data processing

- With pooling we can overcome varying length of a sequence



- Simple CNN is a good point to start

- Many tricks from Computer Vision CNNs work

- Deeper CNNs for sequential data processing can be better

References

1. van den Oord, Aäron, et al. "WaveNet: A Generative Model for Raw Audio." *9th ISCA Speech Synthesis Workshop*. 2016.

2. Ismail Fawaz, Hassan, et al. "InceptionTime: Finding AlexNet for time series classification." *Data Mining and Knowledge Discovery.* 2020.

3. Conneau, A., Schwenk, H., Barrault, L., & Le Cun, Y. Very deep convolutional networks for text classification. EACL. 2017.

4. Ismail, Aya Abdelsalam, et al. "Input-cell attention reduces vanishing saliency of recurrent neural networks." *NeurIPS. 2019*.