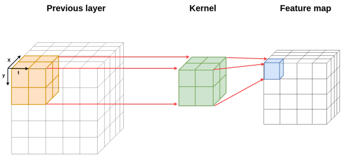
Deep Learning

1D / 2D / 3D Convolutions



Learning goals

- 1D Convolutions
- 2D Convolutions
- 3D Convolutions

1D Convolutions

1D CONVOLUTIONS

Data situation: Sequential, 1-dimensional tensor data.

- Data consists of tensors with shape [depth, xdim]
- Depth 1 (single-channel):
  - Univariate time series, e.g. development of a single stock price over time
  - Functional / curve data
- Depth $>$ 1 (mutli-channel):
  - Multivariate time series, e.g.
    - Movement data measured with multiple sensors for human activity recognition
    - Temperature and humidity in weather forecasting
  - Text encoded as character-level one-hot-vectors

$\rightarrow$ Convolve the data with a 1D-kernel

1D CONVOLUTIONS – OPERATION

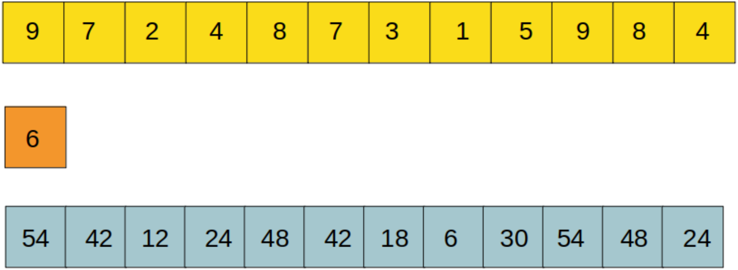


Figure: Illustration of 1D movement data with depth 1 and filter size 1.

1D CONVOLUTIONS – OPERATION

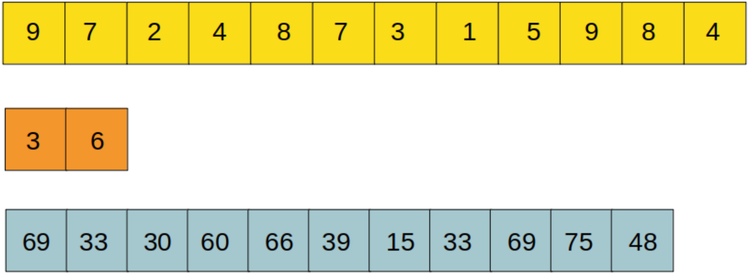


Figure: Illustration of 1D movement data with depth 1 and filter size 2.

1D CONVOLUTIONS – SENSOR DATA

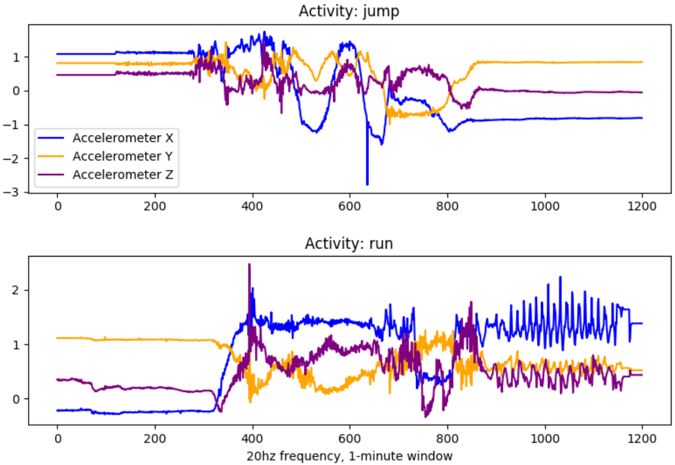


Figure: Illustration of 1D movement data with depth 3 measured with an accelerometer sensor belonging to a human activity recognition task.

1D CONVOLUTIONS – SENSOR DATA

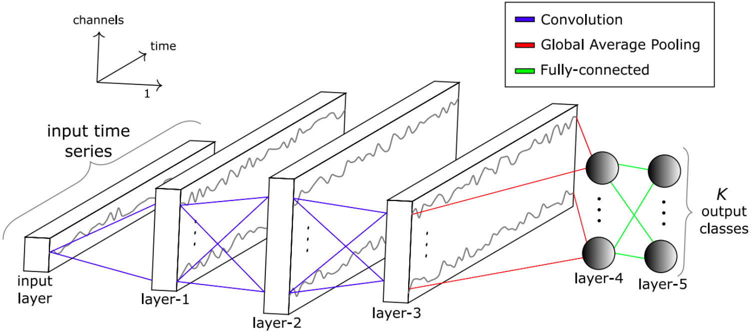


Figure: Time series classification with 1D CNNs and global average pooling (explained later). An input time series is convolved with 3 CNN layers, pooled and fed into a fully connected layer before the final softmax layer. This is one of the classic time series classification architectures.

1D CONVOLUTIONS – TEXT MINING

- 1D convolutions also have an interesting application in text mining.
- For example, they can be used to classify the sentiment of text snippets such as yelp reviews.

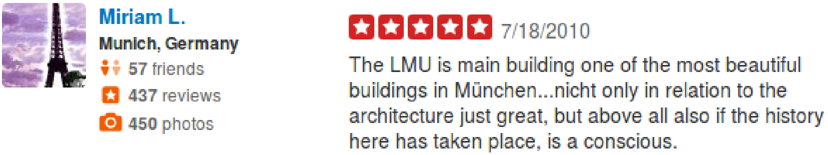
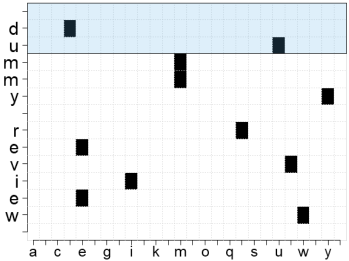


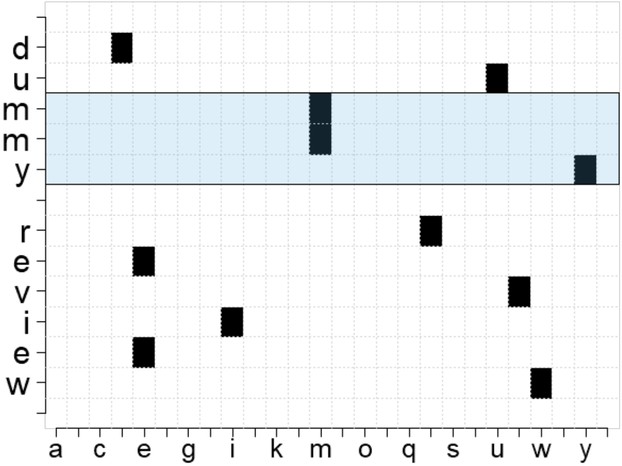
Figure: Sentiment classification: can we teach the net that this a positive review?

1D CONVOLUTIONS – TEXT MINING



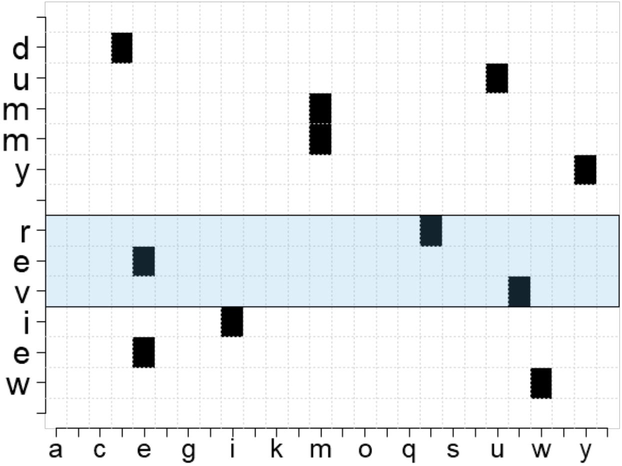
- We use a given alphabet to encode the text reviews (here: *"dummy review"*).
- Each character is transformed into a one-hot vector. The vector for character *d* contains only 0's at all positions except for the 4th position.
- The maximum length of each review is set to 1014: shorter texts are padded with spaces (zero-vectors), longer texts are simply cut.

1D CONVOLUTIONS – TEXT MINING



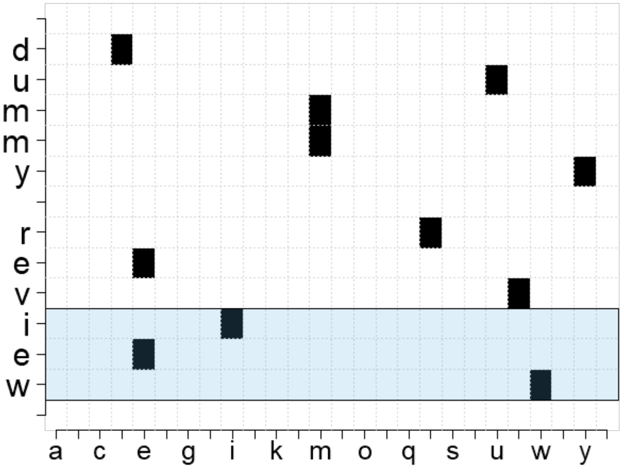
- The data is represented as 1D signal with *depth = size of the alphabet* .

1D CONVOLUTIONS – TEXT MINING



- The temporal dimension is shown as the y dimension for illustrative purposes.

1D CONVOLUTIONS – TEXT MINING



- The 1D-kernel (blue) convolves the input in the temporal y-dimension yielding a 1D feature vector.

ADVANTAGES OF 1D CONVOLUTIONS

For certain applications 1D CNNs are advantageous and thus preferable to their 2D counterparts:

- Computational complexity: Forward propagation and backward propagation in 1D CNNs require simple array operations.
- Training is easier: Recent studies show that 1D CNNs with relatively shallow architectures are able to learn challenging tasks involving 1D signals.
- Hardware: Usually, training deep 2D CNNs requires special hardware setup (e.g. Cloud computing). However, any CPU implementation over a standard computer is feasible and relatively fast for training compact 1D CNNs.
- Application: Due to their low computational requirements, compact 1D CNNs are well-suited for real-time and low-cost applications especially on mobile or hand-held devices.

2D Convolutions

2D CONVOLUTIONS

The basic idea behind a 2D convolution is sliding a small window (called a "kernel/filter") over a larger 2D array, and performing a dot product between the filter elements and the corresponding input array elements at every position.

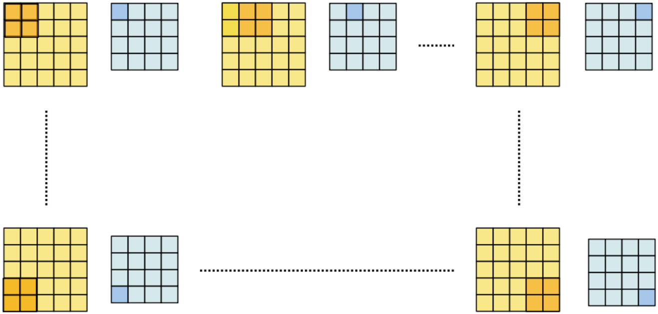
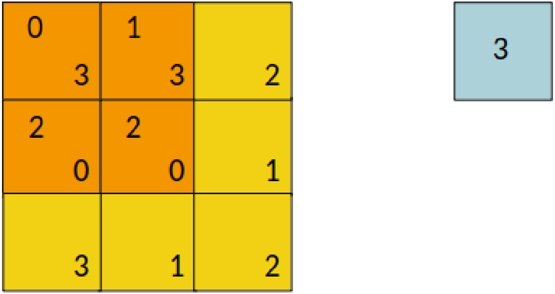


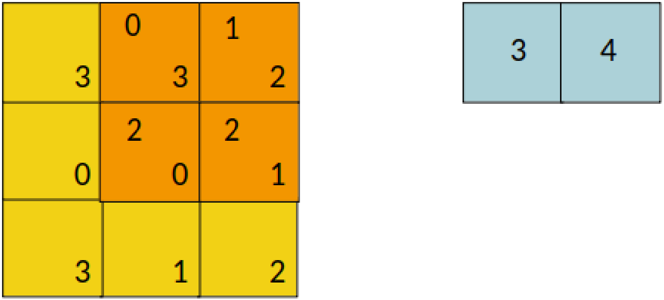
Figure: Here's a diagram demonstrating the application of a 2×2 convolution filter to a 5×5 array, in 16 different positions.

2D CONVOLUTIONS – EXAMPLE



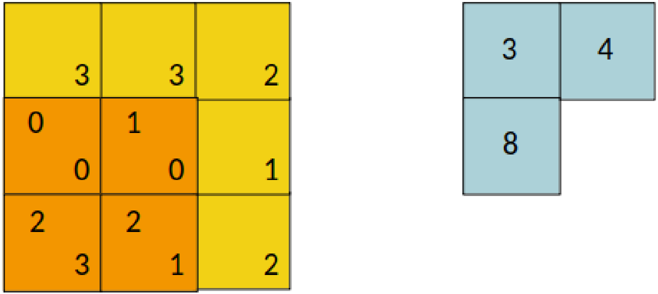
- In Deep Learning, convolution is the element-wise multiplication and addition.
- For an image with 1 channel, the convolution is demonstrated in the figure below. Here the filter is a 2×2 matrix with element [[0, 1], [2, 2]].

2D CONVOLUTIONS – EXAMPLE



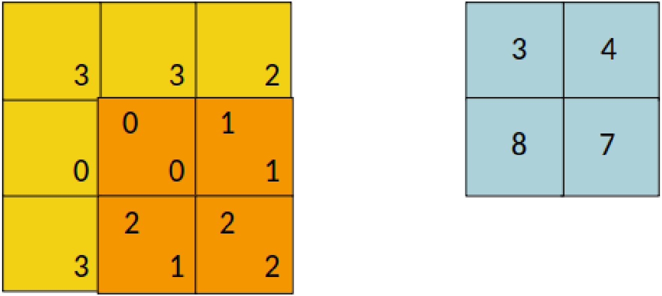
- The filter is sliding through the input.
- We move/convolve filter on input neurons to create a feature map.

2D CONVOLUTIONS – EXAMPLE



- Notice that stride is 1 and padding is 0 in this example.

2D CONVOLUTIONS – EXAMPLE



- Each sliding position ends up with one number. The final output is then a 2×2 matrix.

3D Convolutions

3D CONVOLUTIONS

Data situation: 3-dimensional tensor data.

- Data consists of tensors with shape [depth, xdim, ydim, zdim].
- Dimensions can be both temporal (e.g. video frames) or spatial (e.g. MRI)
- Examples:
  - Human activity recognition in video data
  - Disease classification or tumor segmentation on MRI scans

Solution: Move a 3D-kernel in x, y and z direction to capture all important information.

3D CONVOLUTIONS – DATA

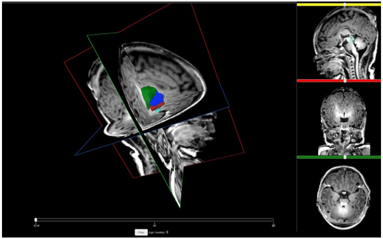


Figure: Illustration of depth 1 volumetric data: MRI scan. Each slice of the stack has depth 1, as the frames are black-white.

3D CONVOLUTIONS – DATA

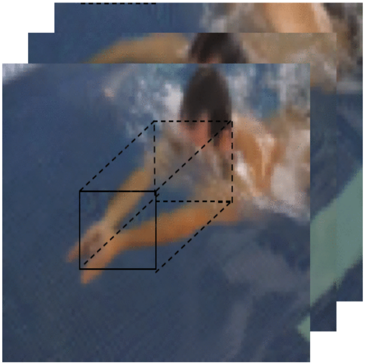
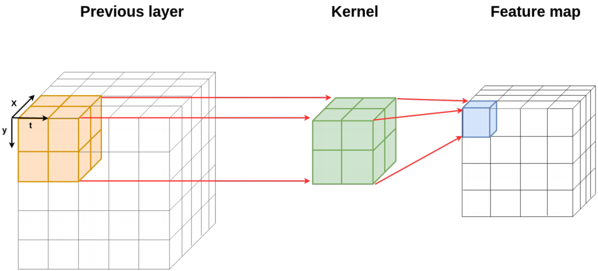


Figure: Illustration of volumetric data with depth > 1: video snippet of an action detection task. The video consists of several slices, stacked in temporal order. Frames have depth 3, as they are RGB.

3D CONVOLUTIONS



- Note: 3D convolutions yield a 3D output.

3D CONVOLUTIONS

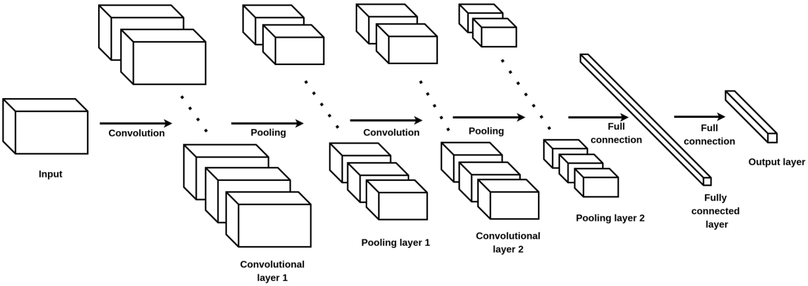


Figure: Basic 3D-CNN architecture.

- Basic architecture of the CNN stays the same.
- 3D convolutions output 3D feature maps which are element-wise activated and then (eventually) pooled in 3 dimensions.

REFERENCES

Dumoulin, Vincent and Visin, Francesco (2016)
A guide to convolution arithmetic for deep learning
https://arxiv.org/abs/1603.07285v1

Van den Oord, Aaron, Sander Dielman, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, and Koray Kavukocuoglu (2016)
WaveNet: A Generative Model for Raw Audio
https://arxiv.org/abs/1609.03499

Benoit A., Gennart, Bernard Krummenacher, Roger D. Hersch, Bernard Saugy, J.C. Hadorn and D. Mueller (1996)
The Giga View Multiprocessor Multidisk Image Server
https://www.researchgate.net/publication/220060811_The_Giga_View_Multiprocessor_Multidisk_Image_Server

Tran, Du, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani and Paluri Manohar (2015)
Learning Spatiotemporal Features with 3D Convolutional Networks
https://arxiv.org/pdf/1412.0767.pdf

REFERENCES

Milletari, Fausto, Nassir Navab and Seyed-Ahmad Ahmadi (2016)
V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation
https://arxiv.org/pdf/1606.04797.pdf

Zhang, Xiang, Junbo Zhao and Yann LeCun (2015)
Character-level Convolutional Networks for Text Classification
http://arxiv.org/abs/1509.01626

Wang, Zhiguang, Weizhong Yan and Tim Oates (2017)
Time Series Classification from Scratch with Deep Neural Networks: A Strong Baseline
http://arxiv.org/abs/1509.01626

Fisher Yu and Vladlen Koltun (2015)
Multi-Scale Context Aggregation by Dilated Convolutions
https://arxiv.org/abs/1511.07122

REFERENCES

Bai, Shaojie, Zico J. Kolter and Vladlen Koltun (2018)
An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling
http://arxiv.org/abs/1509.01626

Augustus Odena, Vincent Dumoulin and Chris Olah (2016)
Deconvolution and Checkerboard Artifacts
https://distill.pub/2016/deconv-checkerboard/*https://distill.pub/2016/deconv-checkerboard/*

Andre Araujo, Wade Norris and Jack Sim (2019)
Computing Receptive Fields of Convolutional Neural Networks
https://distill.pub/2019/computing-receptive-fields/

Zhiguang Wang, Yan, Weizhong and Tim Oates (2017)
Time series classification from scratch with deep neural networks: A strong baseline
https://arxiv.org/1611.06455

REFERENCES

Lin, Haoning and Shi, Zhenwei and Zou, Zhengxia (2017)
Maritime Semantic Labeling of Optical Remote Sensing Images with Multi-Scale Fully Convolutional Network