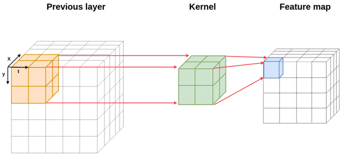
Deep Learning

1D / 2D / 3D Convolutions



Learning goals

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

1D Convolutions

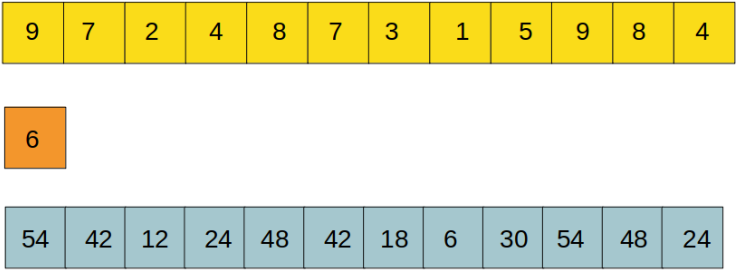
1D CONVOLUTIONS

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



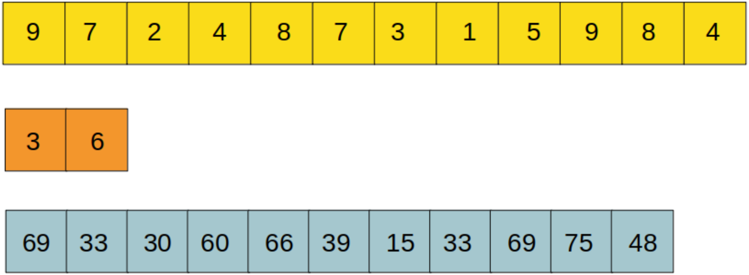
9	7	2	4	8	7	3	1	5	9	8	4

6

54	42	12	24	48	42	18	6	30	54	48	24

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

1D CONVOLUTIONS – OPERATION



9	7	2	4	8	7	3	1	5	9	8	4

3	6

69	33	30	60	66	39	15	33	69	75	48

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

1D CONVOLUTIONS – SENSOR DATA

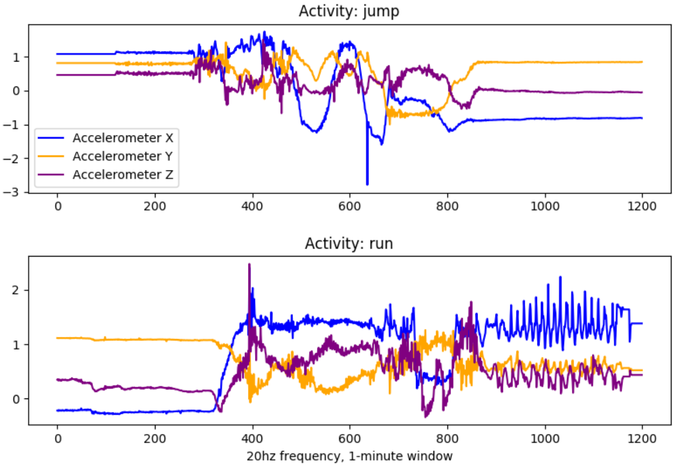


Figure: Illustration of 1D movement data with depth 3 measured with an accelerometer sensor belonging to a human activity recognition task.

1D CONVOLUTIONS – SENSOR DATA

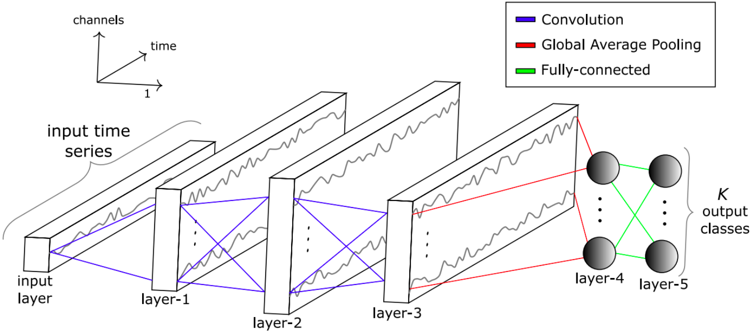
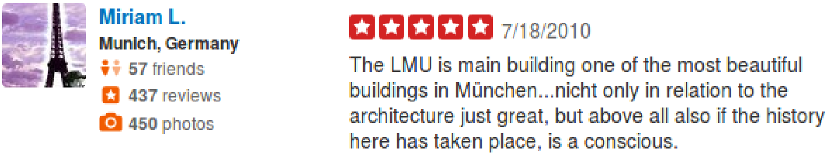


Figure: Time series classification with 1D CNNs and global average pooling (explained later). An input time series is convolved with 3 CNN layers, pooled and fed into a fully connected layer before the final softmax layer. This is one of the classic time series classification architectures (Fawaz et al., 2019).

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.



Miriam L.

Munich, Germany

57 friends

437 reviews

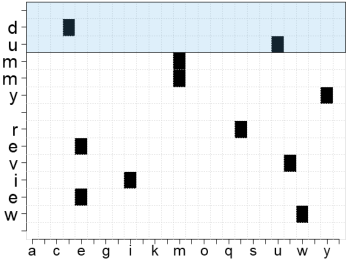
450 photos

★★★★★ 7/18/2010

The LMU is main building one of the most beautiful buildings in München...nicht only in relation to the architecture just great, but above all also if the history here has taken place, is a conscious.

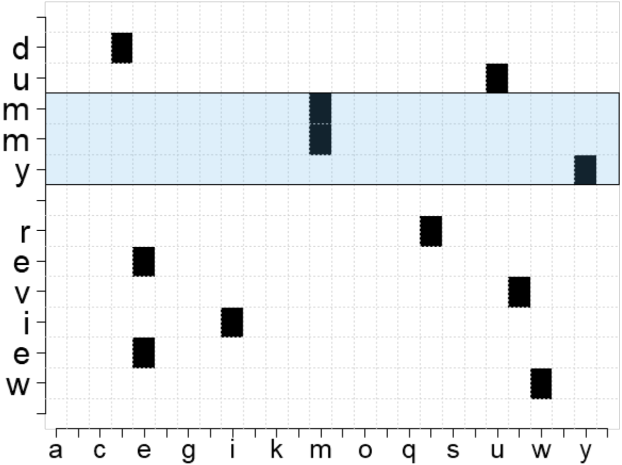
Figure: Sentiment classification: can we teach the net that this a positive review (Yelp, 2010)?

1D CONVOLUTIONS – TEXT MINING



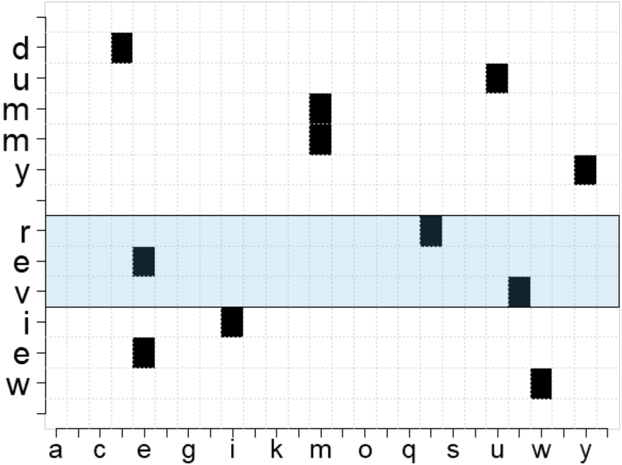
- We use a given alphabet to encode the text reviews (here: *“dummy review”*).
- Each character is transformed into a one-hot vector. The vector for character *d* contains only 0’s at all positions except for the 4th position.
- The maximum length of each review is set to 1014: shorter texts are padded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

1D CONVOLUTIONS – TEXT MINING



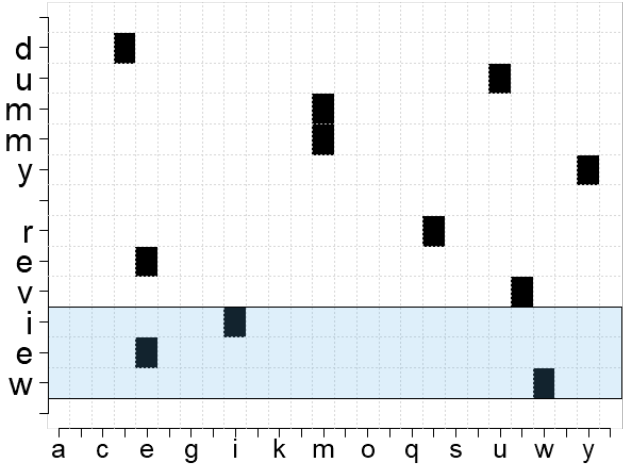
- The data is represented as 1D signal with *depth = size of the alphabet* .

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.

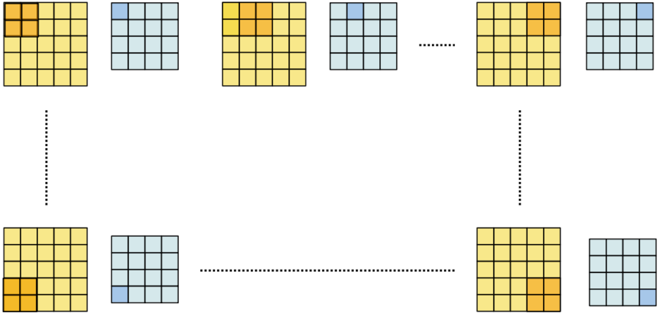
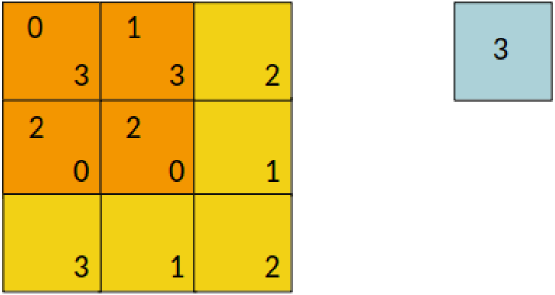


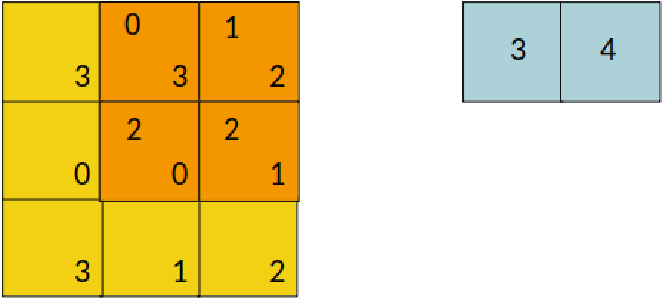
Figure: Here's a diagram demonstrating the application of a 2×2 convolution filter to a 5×5 array, in 16 different positions.

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

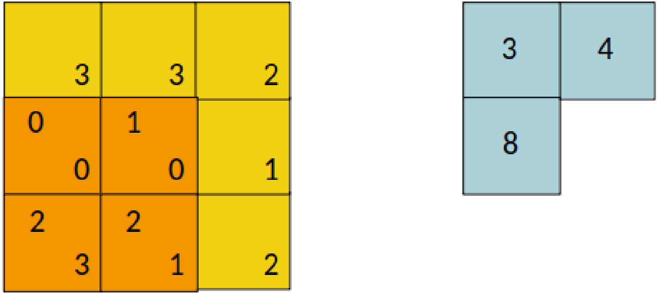


3	0 3	1 2
0	2 0	2 1
3	1	2

3	4

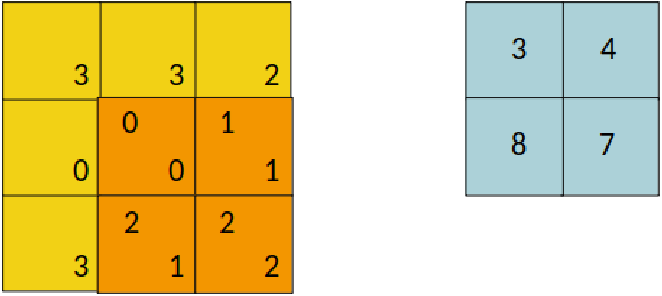
- The filter is sliding through the input.
- We move/convolve filter on input neurons to create a feature map.

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

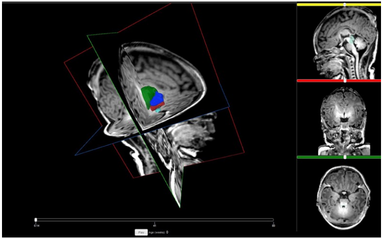


Figure: Illustration of depth 1 volumetric data: MRI scan. Each slice of the stack has depth 1, as the frames are black-white (Gutman et al., 2014).

3D CONVOLUTIONS – DATA

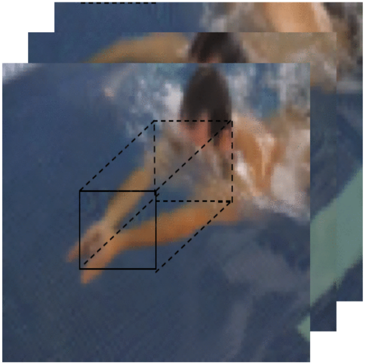
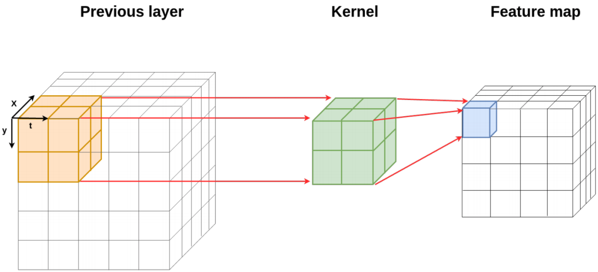


Figure: Illustration of volumetric data with depth > 1: video snippet of an action detection task. The video consists of several slices, stacked in temporal order. Frames have depth 3, as they are RGB (Ghosh, 2018).

3D CONVOLUTIONS



- Note: 3D convolutions yield a 3D output (Jin et al., 2023).

3D CONVOLUTIONS

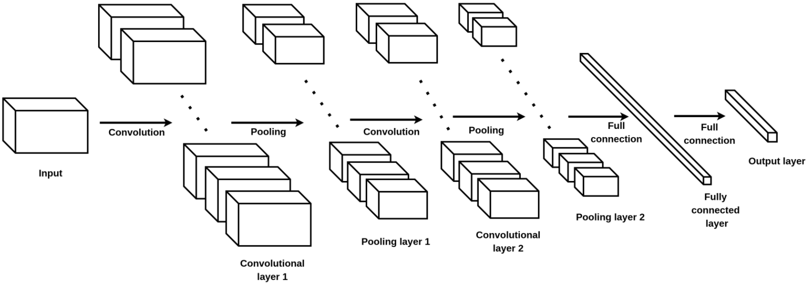


Figure: Basic 3D-CNN architecture.

- 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

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33. https://doi.org/10.1007/s10618-019-00619-1

Gutman, D., Dunn Jr, W., Cobb, J., Stoner, R., Kalpathy-Cramer, J., & Erickson, B. (2014). Web based tools for visualizing imaging data and development of XNATView, a zero footprint image viewer. *Frontiers in Neuroinformatics*, 8, 53. https://doi.org/10.3389/fninf.2014.00053

Ghosh, R. (2018, June 11). Deep learning for videos: A 2018 guide to action recognition. Deep Learning for Videos: A 2018 Guide to Action Recognition. https://blog.qure.ai/notes/deep-learning-for-videos-action-recognition-review

Jin, X., Yang, H., He, X., Liu, G., Yan, Z., & Wang, Q. (2023). Robust LiDAR-Based Vehicle Detection for On-Road Autonomous Driving. *Remote Sensing, 15*(12). https://doi.org/10.3390/rs15123160

Yelp. (2010). *Ludwig-Maximilians-Universität München - Munich*. https://www.yelp.com/biz/ludwig-maximilians-universit%C3%A4t-m%C3%BCnchen-m%C3%BCnchen-2