Advanced Machine Learning

8. Time Series Analytics

Part II – Advanced Models

Prof. Dr. Volker Herbort

THU Technische Hochschule Ulm
University of Applied Sciences

Learning Objectives

- Understand the challenges of real world time series
- Get an overview of suitable ANN types for time series forecasts
  - Understand the advantages
  - Prepare data appropriately
- Understand the concept of Time Series classification
  - Classical Approach using Distance Based Methods
  - New approach using different types of ANNs

Contents

- **Time Series are not always simple….**
- Forcast using ANNs
- Time Series Classification

Downside of simple models

- Simple Models focus on…
  - **complete data:** missing or corrupt data is generally unsupported.
  - **linear relationships**: excludes more complex joint distributions.
  - **fixed temporal dependence**: the relationship between observations at different times, and in turn the number of lag observations provided as input, must be diagnosed and specified.

Downside ctd.

- Simple Models focus on…
  - **univariate data**: many real-world problems have multiple input variables.
  - **one-step forecasts**: many real-world problems require forecasts with a longtime horizon.

Time Series are not always simple

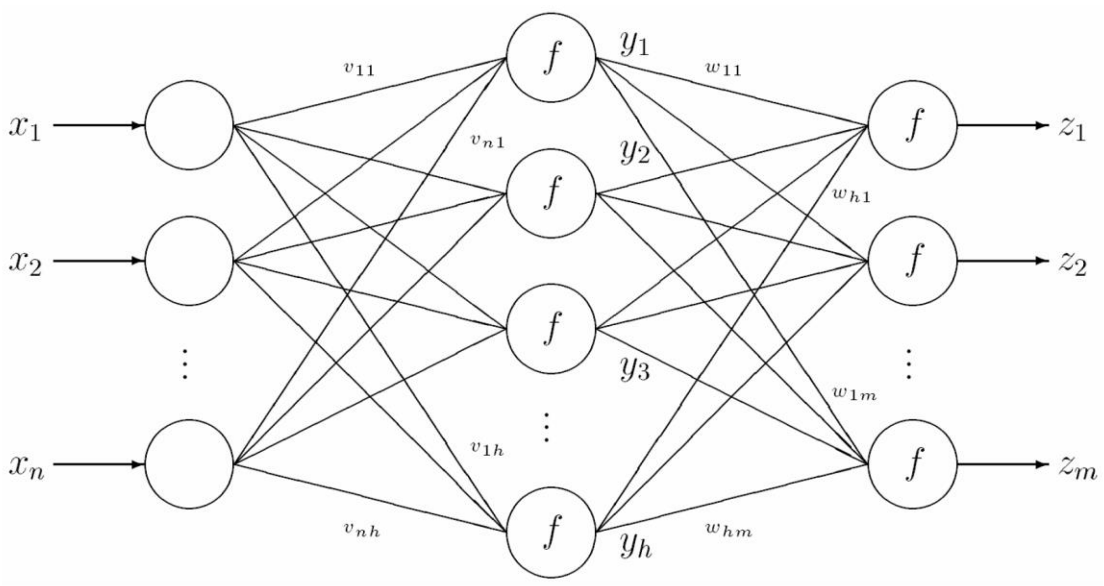
- Last week‘s focus…
  - Univariate time series with one-step forecast
    - 1 series as input, 1 forcast value as output
    - Simple models involving only 1 time dimension
- Many real world problems…
  - Multivariate time series with multi-step forecast
    - n series as input, m forecast values as output
    - Simple models do not work anymore…
  - Solution => Apply e.g. ANNs as regressors for forecast

Contents

- Time Series are not always simple….
- **Forcast using ANNs**
  - **Multi Layer Perceptron**
  - Recurrent Neural Networks
  - Long short-term Memory Networks
  - Gated Recurrent Units
- Time Series Classification

Multi Layer Perceptron (MLP)

- Feed-Forward Network with Back Propagation
  - Regressor and Classifier



  - feature vector X and output vector Z

MLP for Time Series

- MLPs can be used for forecasting time series
  - Feature extraction/modelling in data preparation
    - cleasing, interpolation, etc.
    - create samples using e.g. sliding windows
    - removing seasonality and trends
  - Modelling of non-linear relationships
  - Applicability
    - For simple univariate series gain over basic methods is questionable.
    - Advantages for multivariate time series

Data Preparation for MLP

- Recap: Sliding Window

i	1	2	3	4	5	6	7	8	9	10
t0	112	118	132	129	121	135	148	148	136	119
t1	112	118	132	129	121	135	148	148	136	119
t2	112	118	132	129	121	135	148	148	136	119
t..	...									

- Transformation into feature vector for supervised learning

	f1	f2	f3	t1	t2
s1	112	118	132	129	121
s2	118	132	129	121	135
s3	132	129	121	135	148
s4	129	121	135	148	148

=> also usable for e.g. Regression Trees

Challenges

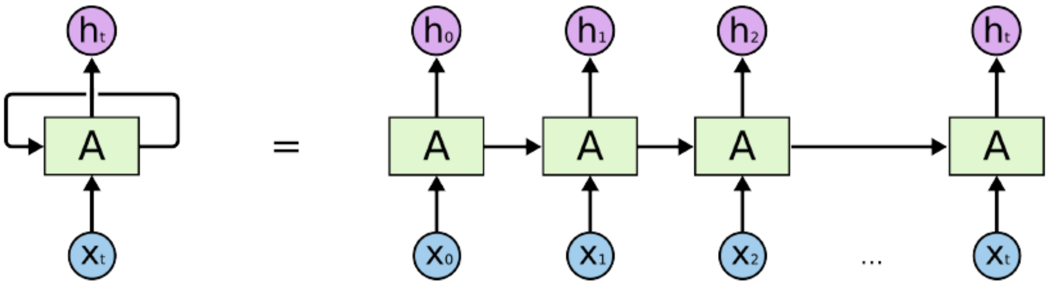
- Feature selection
  - choosing the rights lags for input vector
  - classic tools like (P)ACF might help
  - Transformation to stationary series esp. trend removal

Contents

- Time Series are not always simple….
- **Forcast using ANNs**
  - Multi Layer Perceptron
  - **Recurrent Neural Networks**
  - Long short-term Memory Networks
  - Gated Recurrent Units
- Time Series Classification

Recap: RNN

- MLP with Memory-Connection
  - Integrates the values of predecessors



  - natural sequential learning (like time series)

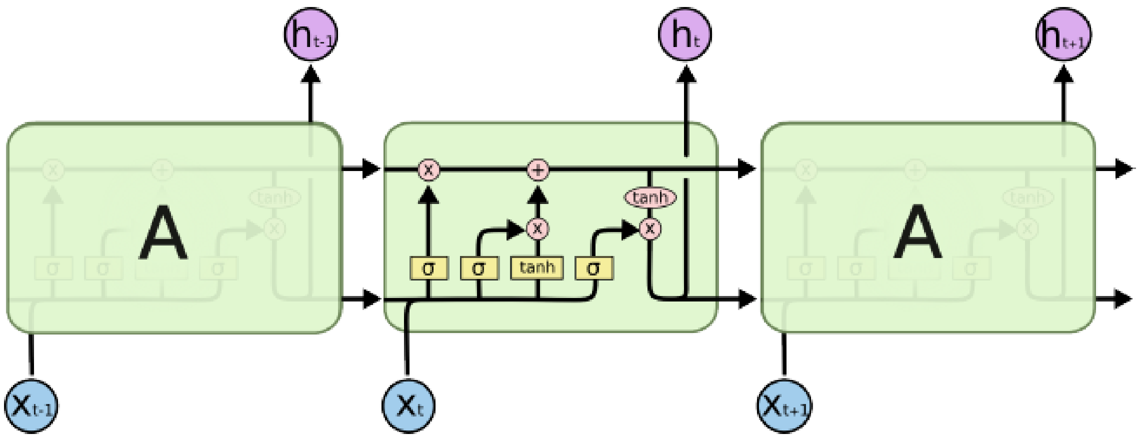
- Problems of RNNs
  - Only recent values are considered
  - Exploding/Vanishing Gradient

Contents

- Time Series are not always simple….
- **Forcast using ANNs**
  - Multi Layer Perceptron
  - Recurrent Neural Networks
  - **Long short-term Memory Networks**
  - Gated Recurrent Units
- Time Series Classification

Recap: LSTM

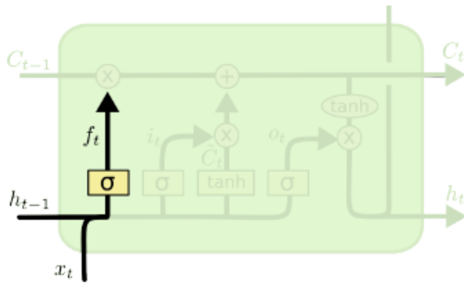
- Special implementation of RNN
  - Solves problem of vanishing gradient
  - Used for e.g. Speech Recognition



LSTM

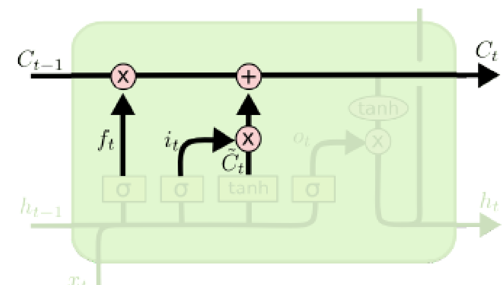
- ## Forget Gate

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$



- ## Actual Cell

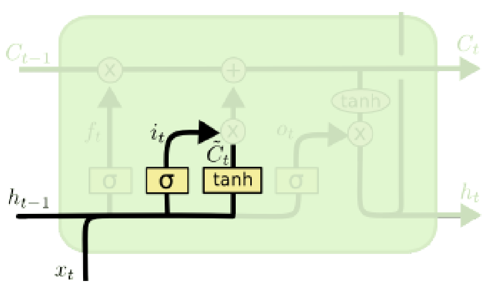
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$



- ## Input Gate

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

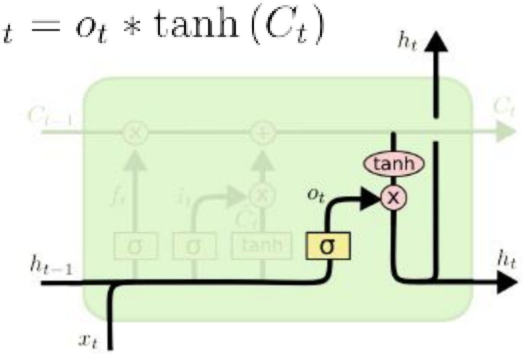
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$



- ## Output Gate

$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$



LSTM for Time Series[2]

- LSTMs
  - …can bridge very long time lags.
  - …can handle noise, distributed representations and continuous values.
  - …do not require an a priori choice of a finite number of states.
  - …work well over a broad range of parameters such as learning rate, input gate bias and output gate bias.

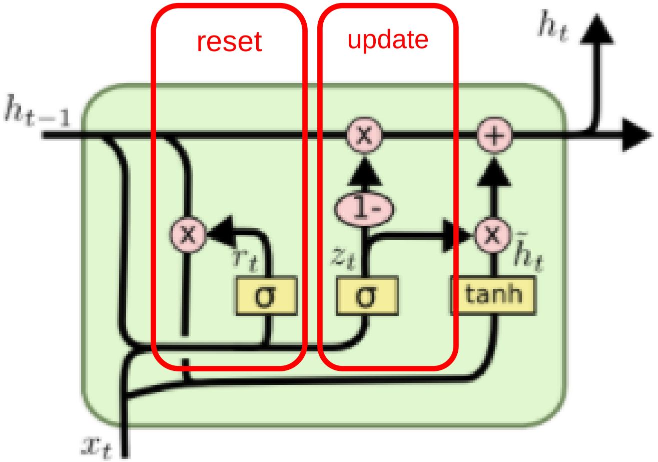
Contents

- Time Series are not always simple….
- **Forcast using ANNs**
  - Multi Layer Perceptron
  - Recurrent Neural Networks
  - Long short-term Memory Networks
  - **Gated Recurrent Units**
- Time Series Classification

Gated Recurrent Unit (GRU)

- Simplified LSTM
  - No Cell State => hidden state for memory
  - Update Gate replaces Input and Forget Gate
  - Reset Gate decides on long term memory

  => easier to compute



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

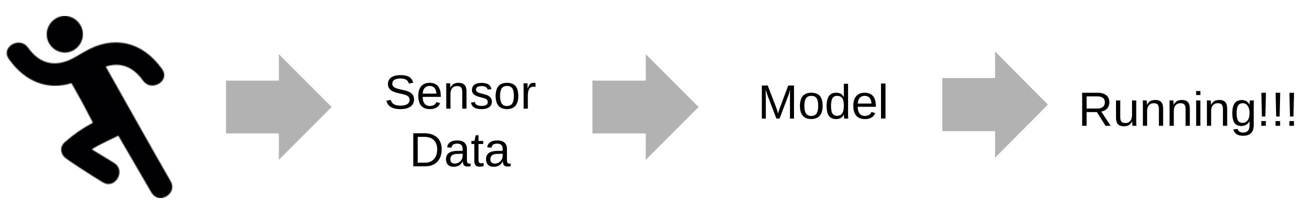
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Contents

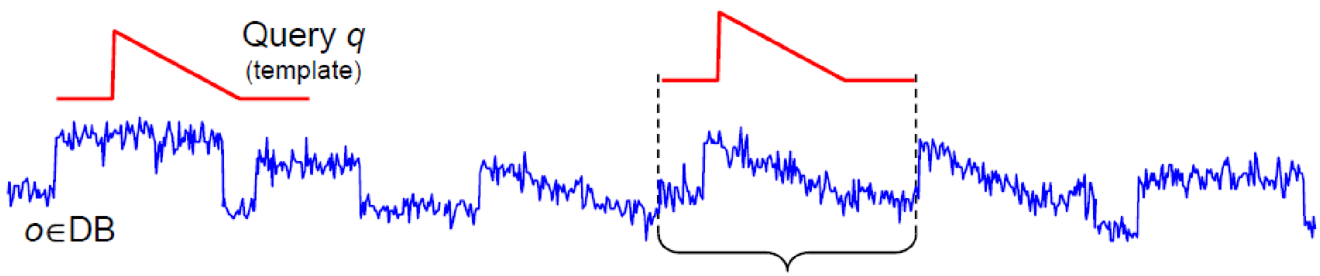
Time Series Classification (TSC)

- In addition to forecasts time series can be also be used for classification e.g.
  - Speech recognition (word from audio signal)
  - Appliance detection from electricity usage
  - Activity recognition from fitness watch

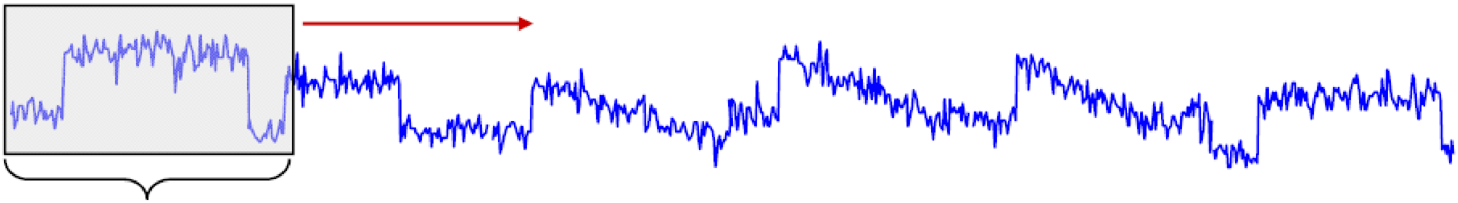


Classic TSC

- Subsequence Matching
  - Query (shapelet) q contains n values



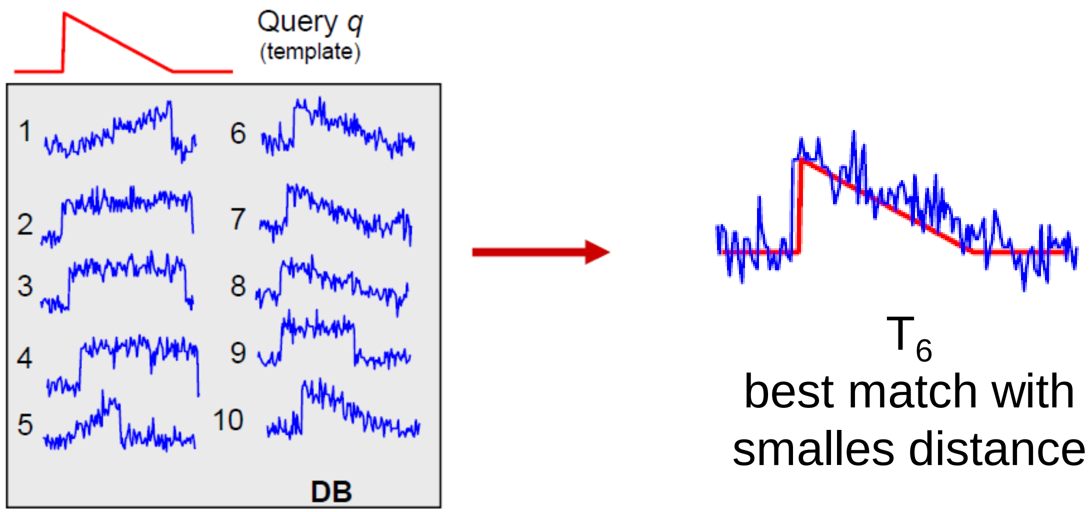
  - Approach: Sliding Window of size n



Source:Skript zur Vorlesung: Spatial, Temporal, and Multimedia Databases Sommersemester 2009, LMU München
Prof. Dr. Hans-Peter Kriegel, Dr. Peer Kröger, Dr. Peter Kunath, Dr. Matthias Renz, Arthur Zimek

Classic TSC ctd.

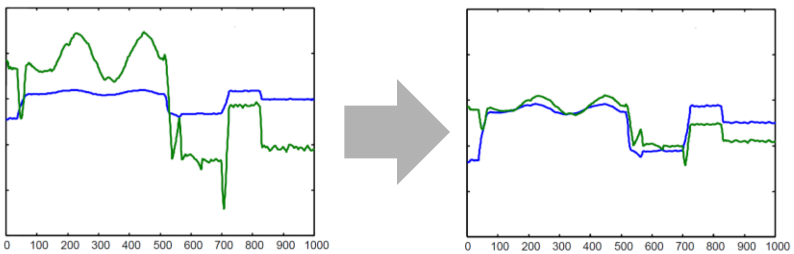
- Whole Sequence Matching
  - Compare whole series t with query q using distance function $dist(q,t) = \sqrt[2]{\sum_{i=1}^{n}(q_i - t_i\ )^2}$



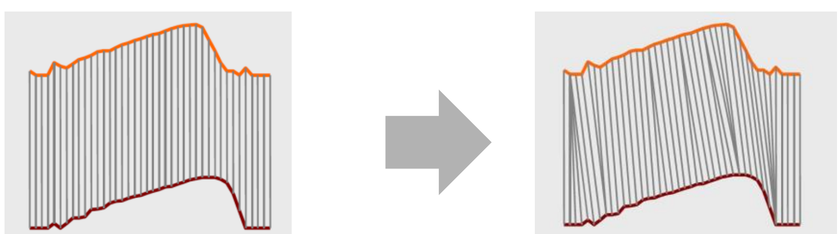
Source:Skript zur Vorlesung: Spatial, Temporal, and Multimedia Databases Sommersemester 2009, LMU München
Prof. Dr. Hans-Peter Kriegel, Dr. Peer Kröger, Dr. Peter Kunath, Dr. Matthias Renz, Arthur Zimek

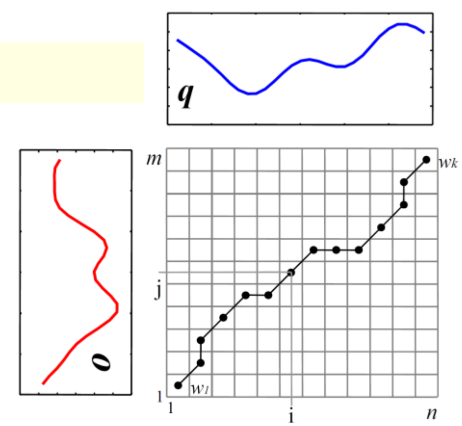
Challenges in Classical TSC

- Subject dependent sensor data
  - Amplitude (weight, volume,…)
    - normalisation of signals



  - Frequence (running speed, voice,..)
    - Dynamic Time Warping





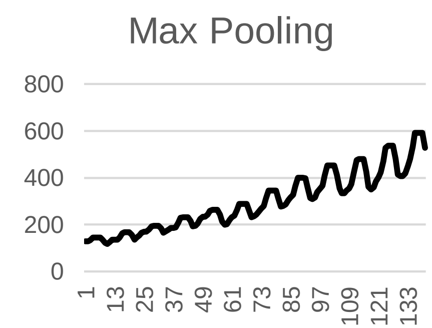
Source:Skript zur Vorlesung: Spatial, Temporal, and Multimedia Databases Sommersemester 2009, LMU München
Prof. Dr. Hans-Peter Kriegel, Dr. Peer Kröger, Dr. Peter Kunath, Dr. Matthias Renz, Arthur Zimek

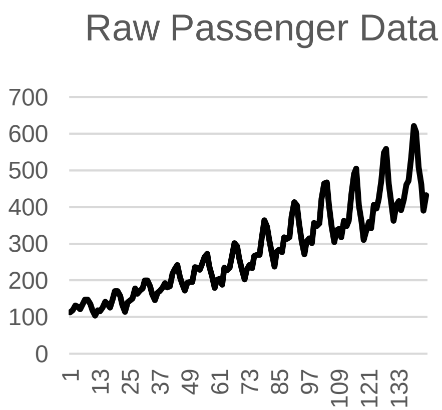
New Approach for TSC[3]

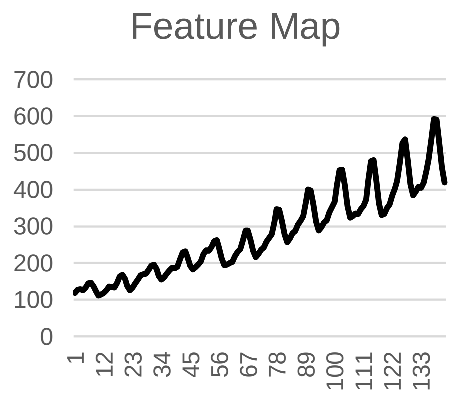
- Use CNNs who...
  - fuse feature extraction and feature classification into a single learning body. They can learn to optimize the features directly from the raw input.
  - can process large inputs with a great computational efficiency compared to the conventional fully-connected Multi-Layer Perceptrons (MLP) networks.
  - are immune to small transformations in the input data including translation, scaling, skewing, and distortion.
  - can adapt to different input sizes.

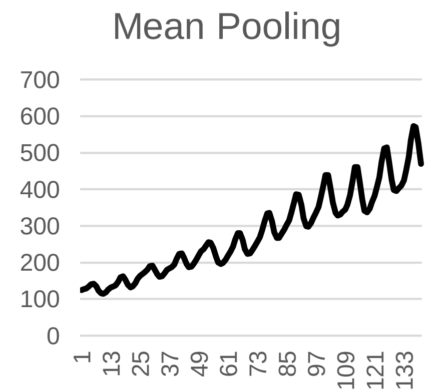
CNNs for TSC

- The trick is the use of a 1D convolution
  - filters are used to prepare data and learn features e.g.
    - up and down sequences
    - peaks and plateaus



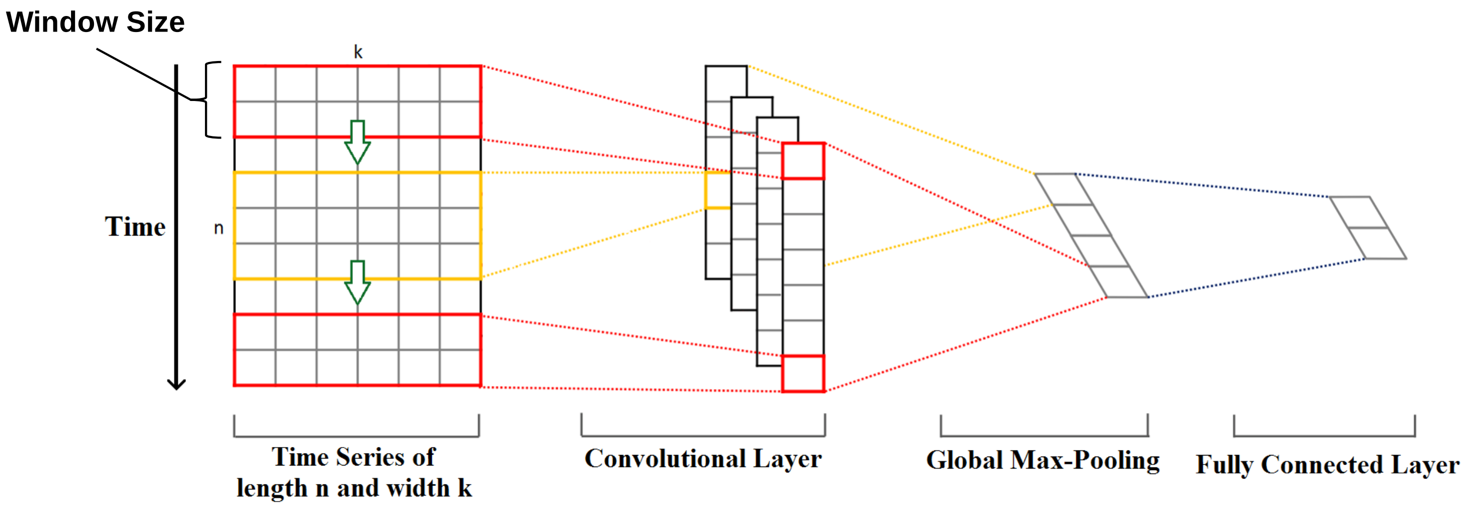






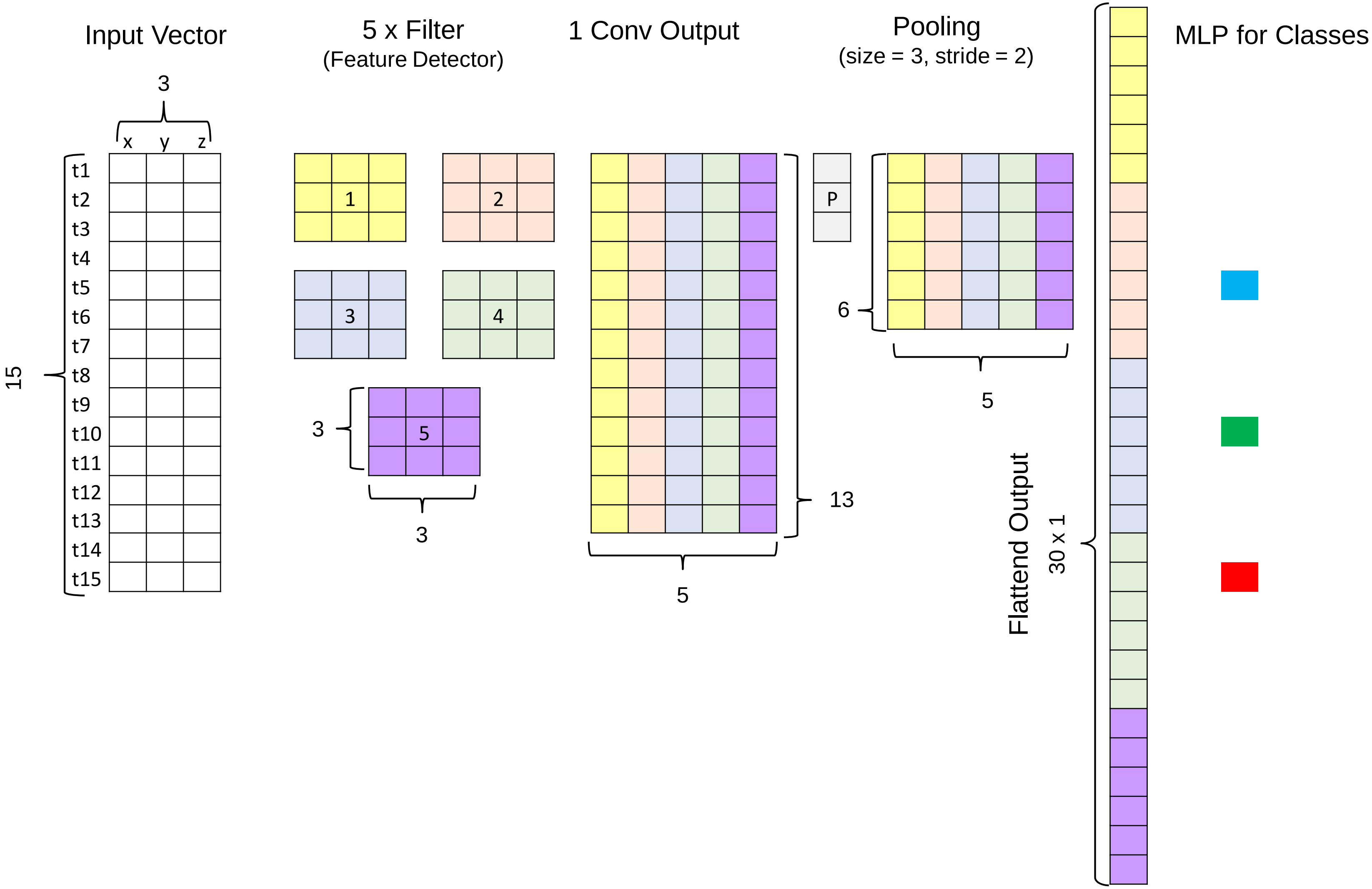
CNNs for TSC

- Multivariate time series ***t*** of length ***n***
  - filter width is similar to width of ***t*** = ***k*** (1d Convolution)
  - filter length depends on purpose of filter (window)
  - Number of filters reflects features to be detected
  - Filter(s) applied from t_1 until t_n depending on stride size



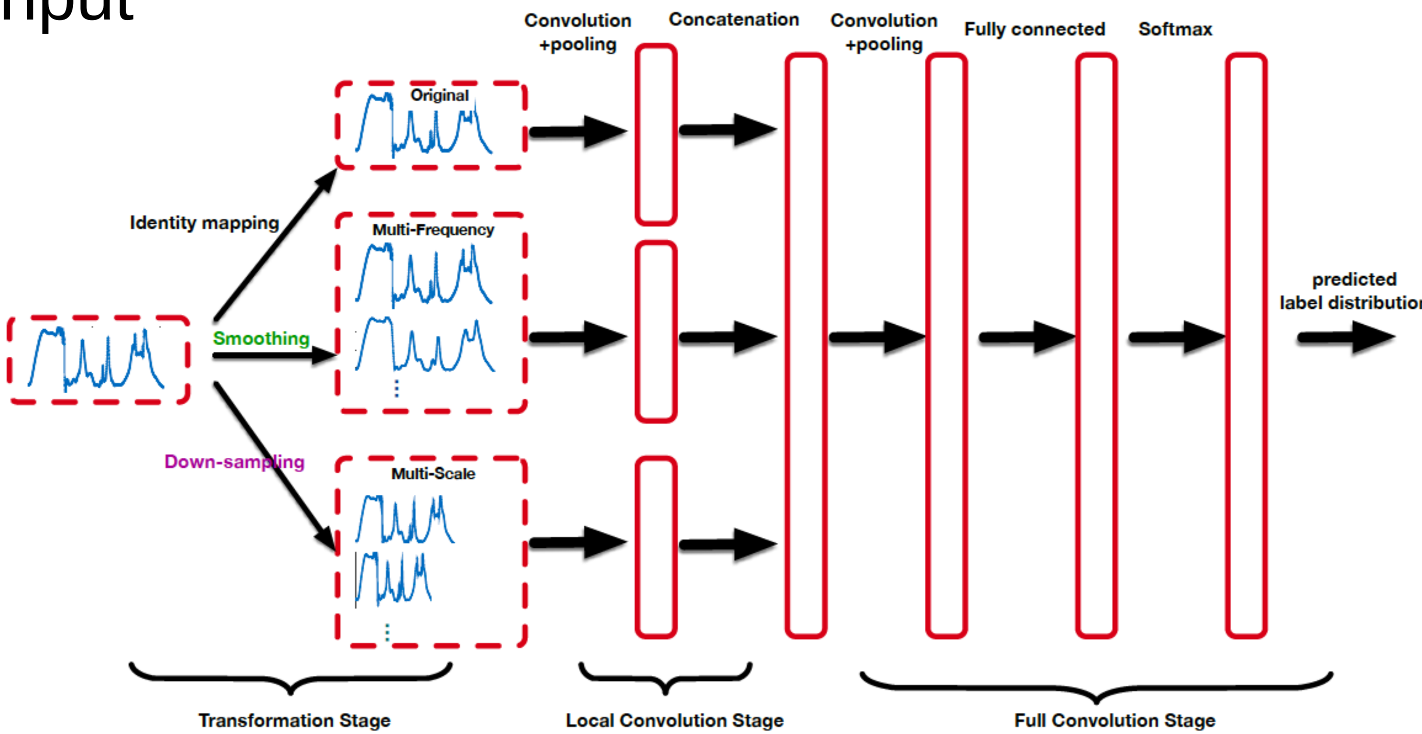
Source: https://towardsdatascience.com/how-to-use-convolutional-neural-networks-for-time-series-classification-56b1b0a07a57

Example 1D CNN



Multi-Scale CNN[4]

- Idea
  - Use already transformed versions of time series as input



References

- [2] Sepp Hochreiter & Jürgen Schmidhuber, „Long Short-Term Memory“, Neural Computation 9 (8) 1735-1780,1997
- [3] Kiranyaz et. al, 1D Convolutional Neural Networks and Applications – A Survey
- [4] Zhicheng Cui, Wenlin Chen, Yixin Chen, Multi-Scale Convolutional Neural Networks for Time Series Classification