

Convolutional Neural Networks for Time Series Classification

2020.12.07 NTU Artificial Neural Network course project

En-Chi Su

1-D Convolution for Time Series Classification

- Step 1:

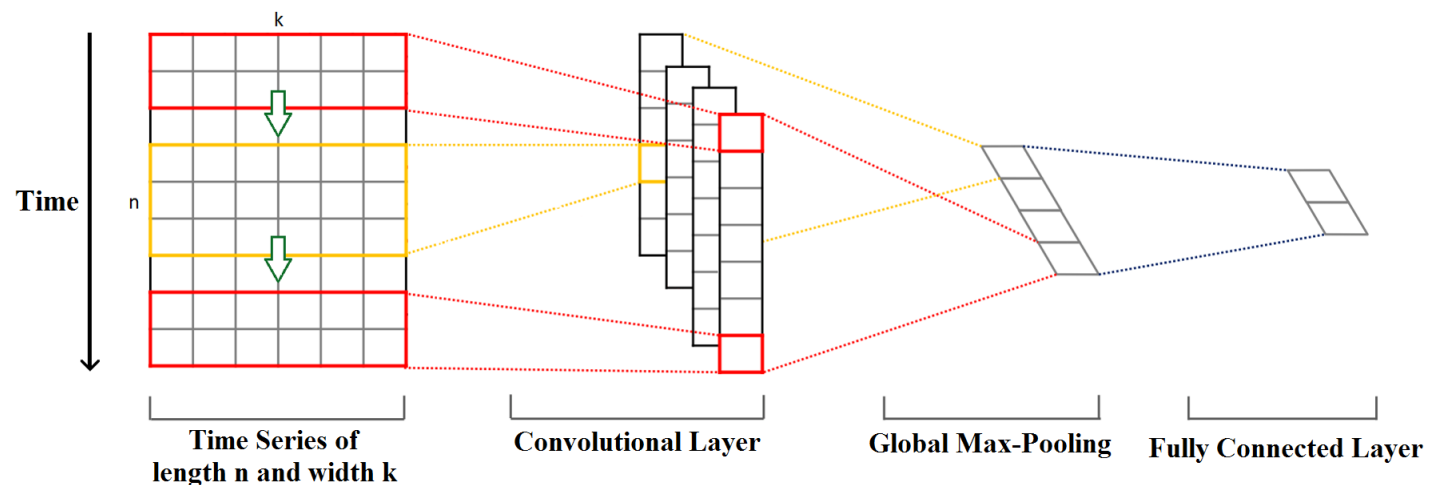
The kernel moves in one direction from the beginning of a time series towards its end, performing convolution.

- Step 2:

Apply global max-pooling to each of the filtered time series vectors: the largest value is taken from each vector.

- Step 3:

The vector of maximums is the final feature vector that can be used as an input to a regular fully connected layer.



Multi-scale convolutional neural network

- Step 1: Transformation Stage

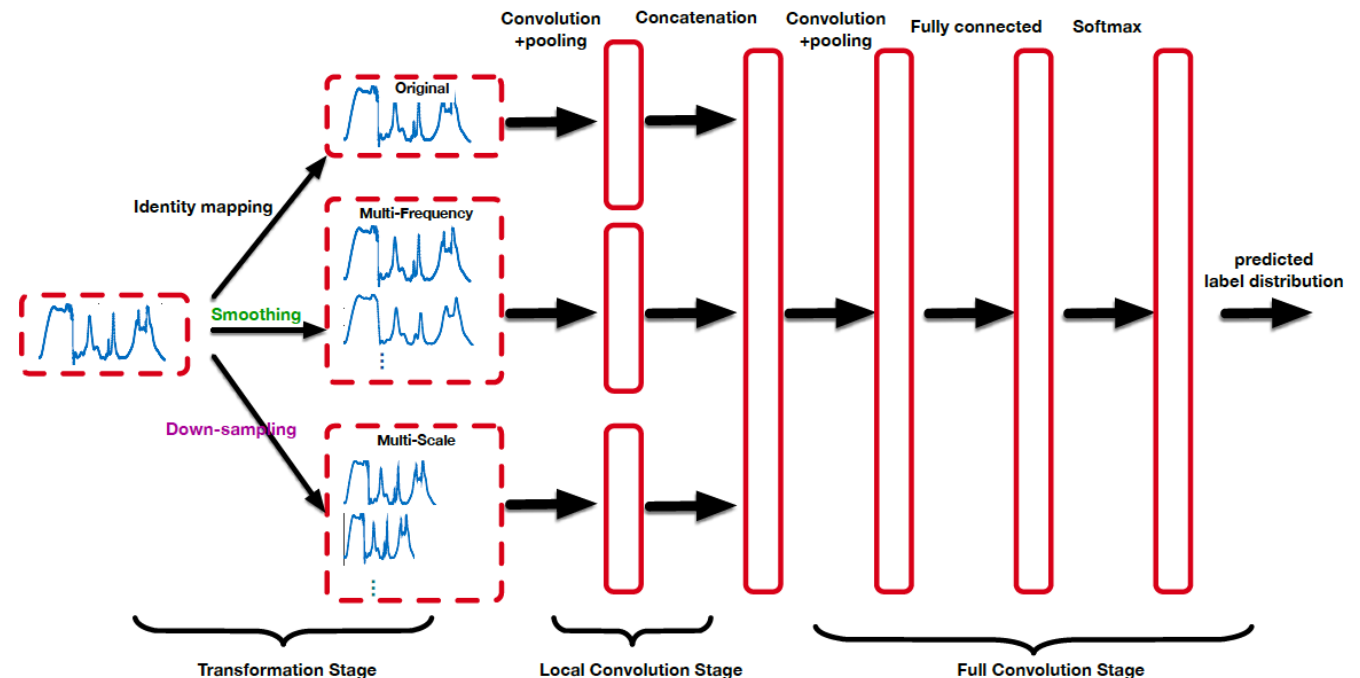
Different transformations are applied to the original time series

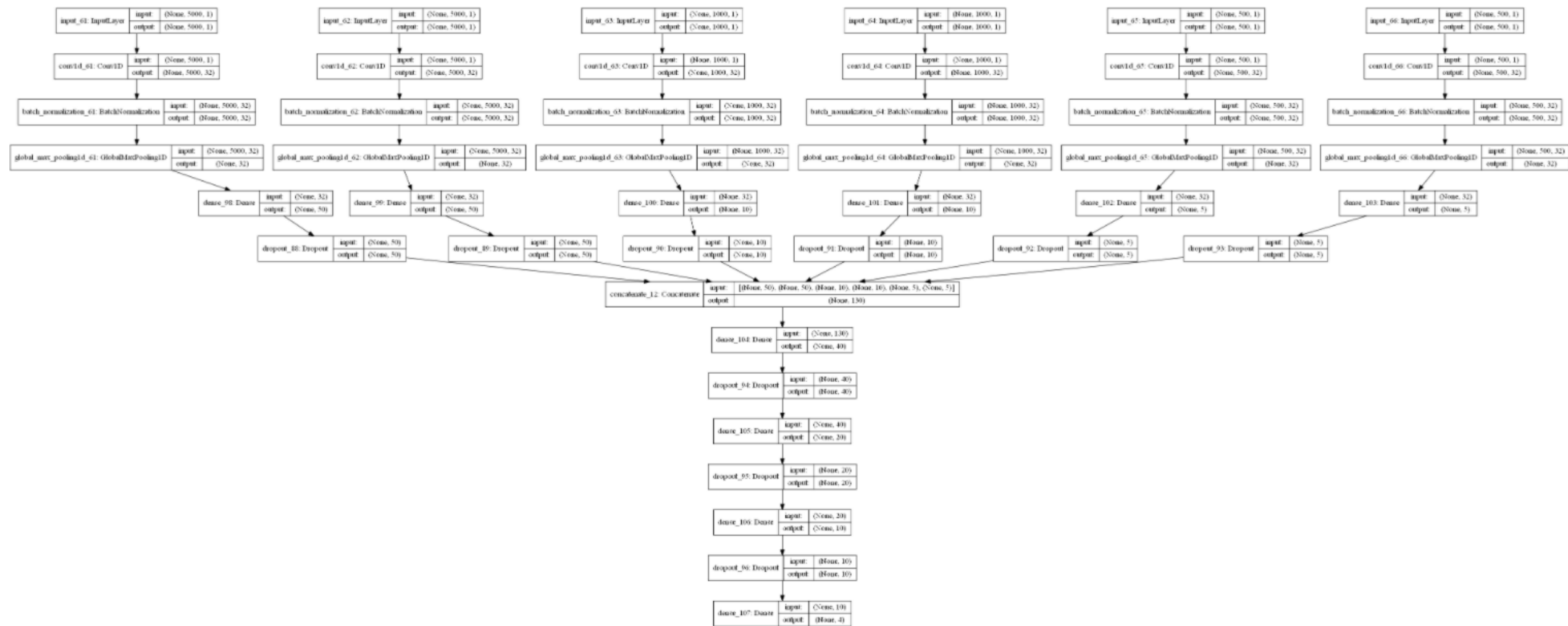
- Step 2: Local Convolution Stage

Perform 1-D convolution with different filter on each of the time series

- Step 3: Full Convolution Stage

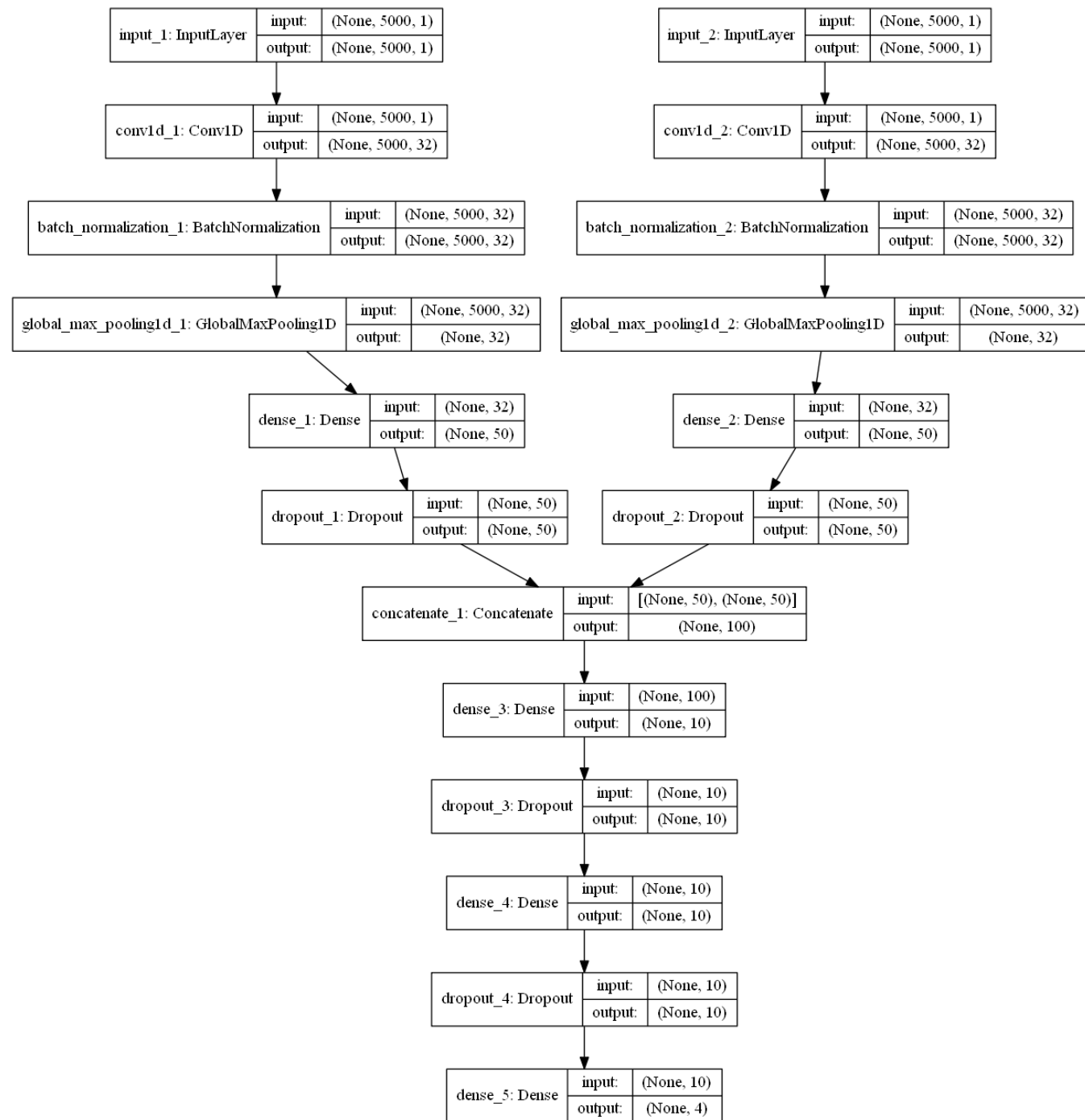
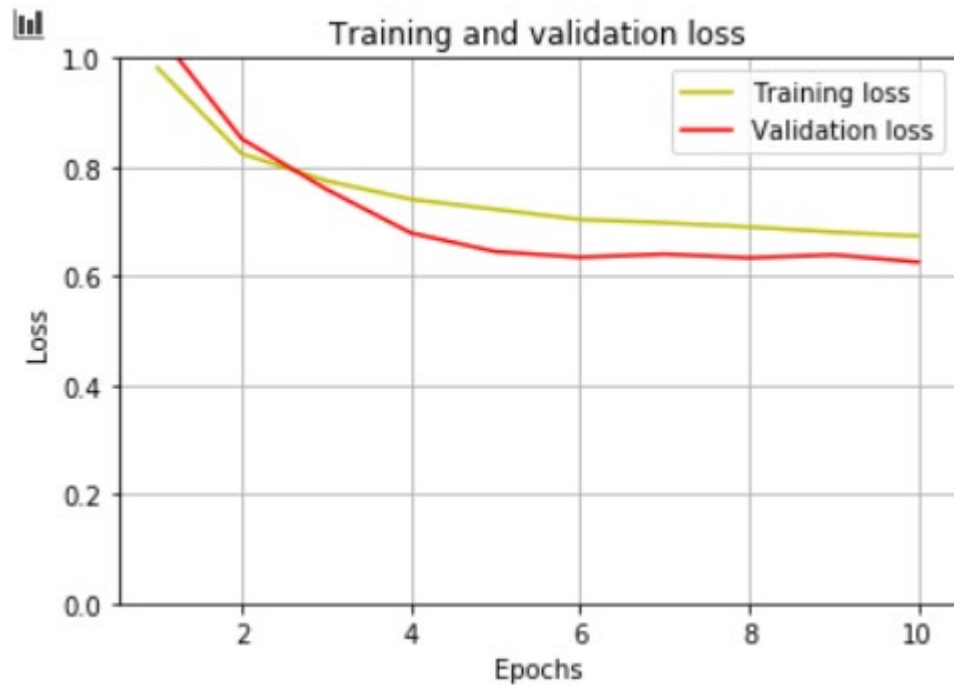
Concatenate outputs of local convolution stage from all branches. Then several more convolutional and max-pooling layers are added.





Results

Measurement	Performance
Cross Entropy	0.606
MSE	6.659
Accuracy	0.829



Reference

- <https://towardsdatascience.com/how-to-use-convolutional-neural-networks-for-time-series-classification-56b1b0a07a57>
- https://medium.com/@Rehan_Sayyad/how-to-use-convolutional-neural-networks-for-time-series-classification-80575131a474
- <https://medium.com/@hanify/sequential-api-vs-functional-api-model-in-keras-266823d7cd5e>
- <https://stackoverflow.com/questions/46503816/keras-conv1d-layer-parameters-filters-and-kernel-size>
- <https://stackoverflow.com/questions/43235531/convolutional-neural-network-conv1d-input-shape>