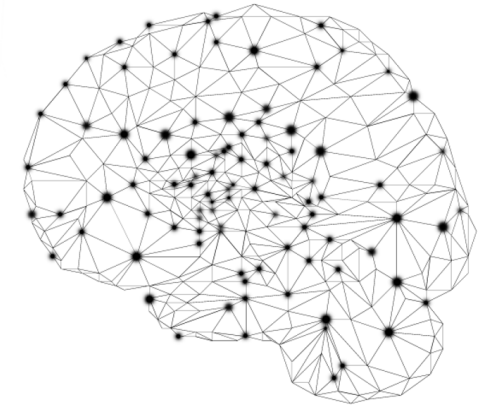

Neural Networks and Data Science

Lab #12

17.01.2024
Deadline: 24.01.2024, 12:10

Nick Krüger
Marcel Völschow

HAW HAMBURG

Problem 1

In this lab, we will forecast power consumption (SMARD) data provided by the German Netzagentur. A chunk of this dataset in provided in the CSV file `SMARD.csv`. However, before we can begin to train our networks, we have to implement the pre-processing functions for this task.

a) **Splitting the dataset:** Implement a function which splits the data set into a training and validation data set. The input of the function should be a dataframe containing the SMARD data, as well as the first and last timesteps of both the training and validation data sets. These four values should be pandas `datetime` objects. After the data is split, the columns containing `datetime` objects should be removed. The output of the function should be two numpy arrays, one containing the training data set and one containing the validation data set.

b) **Standardization:** Implement a function which normalizes the training and validation data sets, using the mean $\bar{x}$ and standard deviation σ of only the training data set. The input should be both datasets, and the output the standardized datasets. (Hint: $x_{\text{norm}} = \frac{x-\bar{x}}{\sigma}$)

c) **Creating data windows:** Finally, we need to create labelled input data from our training and validation datasets to train our networks. For this purpose, implement a function which creates data windows from a given data set. The input of the function should be the data set as well as the number of input and output timesteps. Starting at each timestep, the function shall create a new data window, first selecting the next timesteps as an input series, based on the number of input timesteps, followed by the output series. An example of this method is provided in Fig. 1. These two time series should then be saved in two numpy arrays, one for input data and one for labelled data. The shape of the input array should be (number of data windows, input timesteps, input features). Conversely, input array should be (number of data windows, input timesteps, input features). Note that the first index in both arrays have to be the same for each pair of input and label. The output of the function should be the input and label arrays.

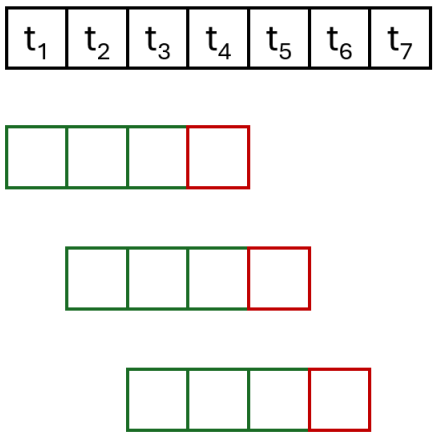

Figure 1: Example for windowed time-series data. The green timesteps indicate the input series, the red the label series. In this example, in number of input timesteps is 3 and the number of output timesteps is 1

Problem 2

Now that we have created the required functions, we can begin to train our time-series forecasting networks.

a) Load the .csv using the provided function `read_SMARD_data()`. For this exercise, set `remove_bad_columns` as True.

b) Apply your pre-processing functions to the dataset. The first timestep of the training dataset should be `01.01.2020 00:00`, ending at `30.06.2022 23:45`. The validation dataset shall start at `01.06.2022 00:00` and end at `31.12.2022 23:45`. Set the number of input timesteps to 96 and the number of output timesteps to 8.

c) Build a recurrent neural network to forecast the created datasets. Below, you can see a couple of example architectures for such a network, however feel free to test different hyperparameters as you wish.

   - A GRU layer with 128 units
   - A LSTM layer with 64 units
   - A LSTM layer with 128 layers, followed by a LSTM layer with 32 units

   Note that every network would have a dense output layer after the recurrent layers with a linear activation function and the number of units matching the number of output timesteps from b).

c) Compile and train your network for up to 30 epochs.

d) Using the `model.predict()` function, explicitly predict individual input series. Plot and compare the model output to the actual labels. How would you rate your network performance?

e) **Optional:** Implement various different models with different hyperparameters and plot their predictions in the same plot. How does their performance vary? Can you spot a trend?