46040 - Introduction to energy analytics F24

Hands-on session of Lecture 9: Machine Learning II

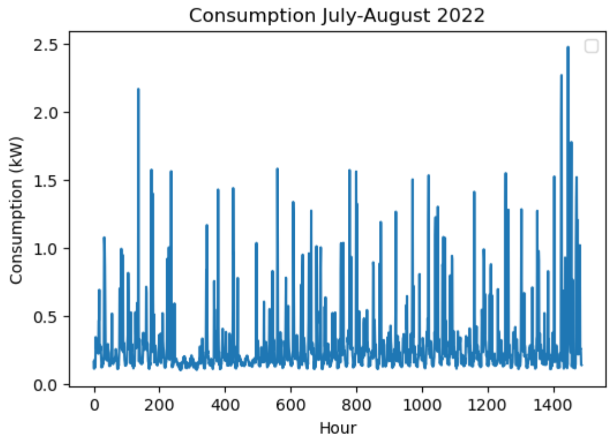
The aim of this hands-on session is to apply the theory from lectures 8 and 9 and learn how to train and evaluate LSTM models.

Task 1: Predicting the consumption 1 hour ahead

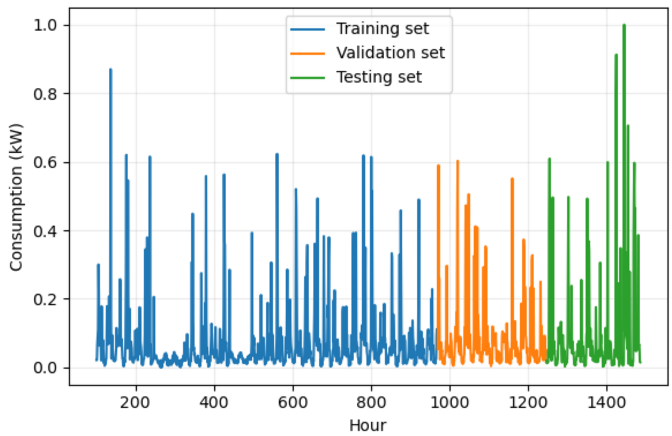
In this task you will learn to implement an LSTM model and predict the consumption in the next hour of a consumer profile.

You will use the dataset that was used before in Hands-On 6 and 7: ProsumerHourly_withUTC.csv and WeatherData.csv. You can find the Hands-on 9.ipynb script on DTU Learn.

1.1 **Loading and plotting the data**.
We start by plotting the target variable that we want to predict, in this case the consumption, for the time-period that we are interested in. You can choose the period on which you want to train and test your LSTM model. For example, see the plot below for the time-period of 1st of July - 31st of August 2022.



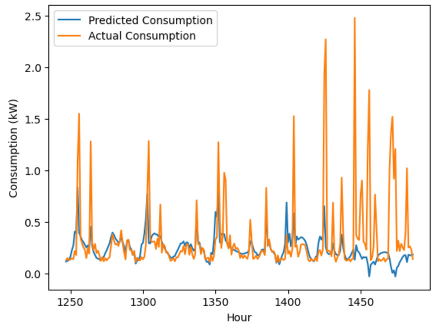
1.2 **Splitting the dataset**.
The goal for this task is to predict the consumption 1 hour ahead. Fill in the values for 'look_ahead' and 'look_back' variables. 'look_back' decides how many time steps the model uses as input features, while 'look_ahead' determines the number of steps that the model will look ahead. For our model, we need to split the dataset into a training set, a validation set and a test set. You can define these ratios yourself. Plot the different sets. Your plot could for example look like this:



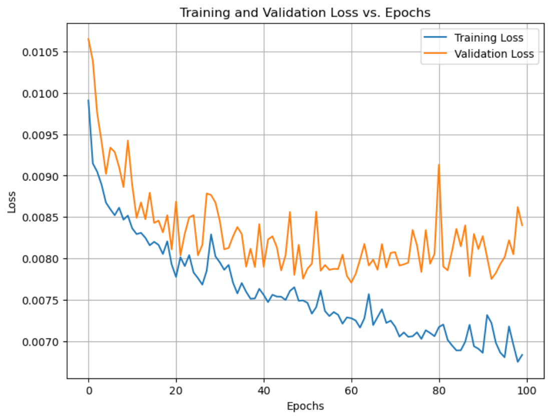
1.3 **Train the model.**

Once we have identified the training, validation and test set, we are able to train the model. Set the epoch and batch size. What changes if you vary the epochs or batch size?

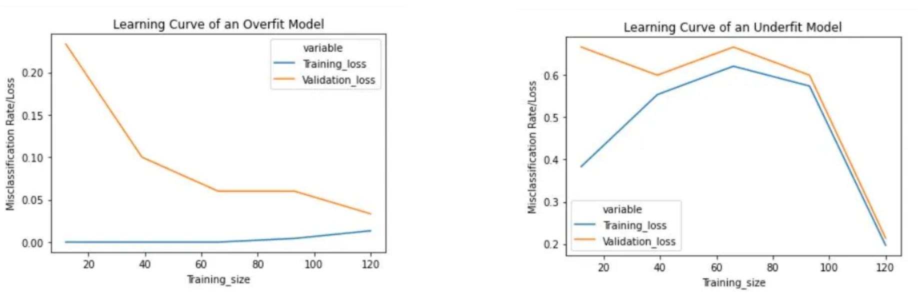
Plot the predicted consumption and the actual consumption. What do you observe?



Plot the training and validation loss. See below for an example:



You can see if your model is under- or overfitting by looking at the examples in Figure 1.



(a) **(b)**

Figure 1: Examples of models that are under- and overfitting

1.4 **Evaluate the model**.

First, plot your predicted consumption and the actual consumption.
Typically, you would evaluate the model using certain metrics.

- Mean Squared Error (MSE): This measures the average of the squares of the errors or deviations. It gives higher weight to larger errors. The lower the value, the better.
- Mean Absolute Error (MAE): This measures the average of the absolute errors. It gives equal weight to all errors. The lower the value, the better.
- Root Mean Squared Error (RMSE): This is the square root of the MSE. It is interpretable in the same units as the target variable.
- R-squared (R2) Score: This measures the proportion of the variance in the dependent variable that is predictable from the independent variables. It provides an indication of the goodness of fit. Higher R2 values (close to 1) generally indicate a better fit of the model to the data.

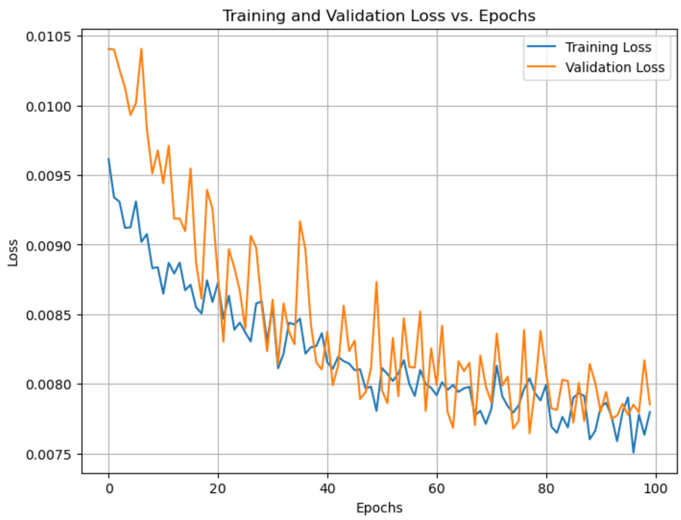
How does your model perform? Can you improve the model? If so, how?

1.5 **Techniques to improve the model**.

In the previous hands-on, the bi-directional LSTM was used with an activation function and a dropout. For this step, use an activation function and see if you can improve the performance of your model. ReLU introduces non-linearity to the model, allowing it to learn complex patterns and relationships in the data.

Next, add a dropout to your model. During training, dropout randomly sets a fraction of input units to zero with a specified probability (dropout rate) at each update, effectively "dropping out" some neurons. By randomly dropping out neurons, dropout prevents the model from relying too heavily on specific neurons and encourages the network to learn more robust features. Dropout helps prevent overfitting by randomly dropping out neurons during training, while ReLU introduces non-linearity and promotes efficient learning of complex patterns in the data.

Plot the training and validation loss of your model:



1.6 **Predicting 12 hours ahead.**

Now, repeat the previous steps and build a model that predicts the consumption 12 hours ahead.

Task 2: Predicting the PV production Now that you have established an LSTM for predicting the consumption, you can forecast the PV production.

Repeat the steps of task 1 and build your own model for predicting the PV prediction.

2.1 **Training a model for the PV Production**.

- Load & visualize the data
- Split your data in a training, testing and validation set
- Train the model
- Evaluate the model
- Improve the model

2.2 **Using exogenous variables**.
The last step involves using exogenous variables in your model. First, you need to explore how the other columns in *df_merged* relate to the target variable.

- Create a heatmap of a correlation matrix between your exogenuous variables and your target variable. Decide on which of the features you would like to use as additional input to your model.
- Add these additional features as input to your model. Does it change the performance of your model?