Proof of Concept Experiment

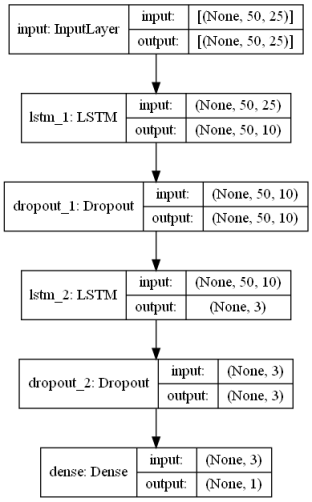
Goal:

Evaluation whether the concept of processing data through multiple small LSTM splits speeds up computation time while maintaining a similar prediction accuracy.
The idea being that LSTM splits can use information locality while processing data to achieve comparable prediction accuracy with overall smaller models, thus resulting in decreased computation (and training) time.

Setup:

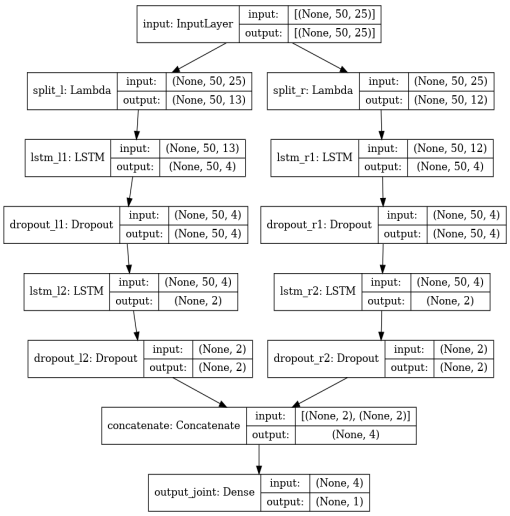
We evaluated two LSTM Models, one standard LSTM Layers (normal LSTM) and one splitting input data in half and processing both parts through smaller LSTM Layers (“split LSTM”).
We measured and compared training time and validation accuracy, for three runs per model.
One run consisted of 50 training epochs.

Normal Model Layout:



The normal LSTM model consists of an obligatory input layer with shape (50, 25) representing (sequence length, number of features), followed by an LSTM-Layer with hidden size 10, a Dropout Layer with dropout rate 0.2, another LSTM-Layer with hidden size 3, followed by another Dropout Layer with dropout rate 0.2. Finally a Dense output layer with 3 nodes, each representing classification class.

Split model Layout:



The split LSTM model consists of an obligatory input layer with shape (50, 25) representing (sequence length, number of features), followed by two Split Layers (Keras implementation of splitting the input in two halves) with shape (50, 12) and (50,13) respectively. Then for each half of the input we have an LSTM-Layer with hidden size 4, a Dropout Layer with dropout rate 0.2, another LSTM-Layer with hidden size 2, followed by another Dropout Layer with dropout rate 0.2. Finally a Dense output layer with 3 nodes, each representing classification class.

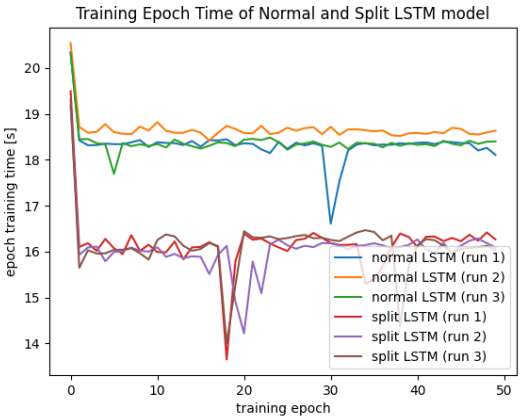
Task:

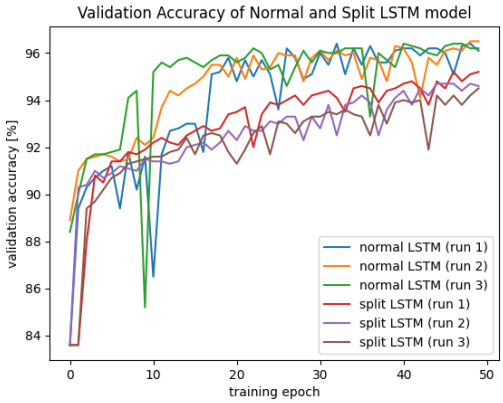
Both models were tasked with classification of aircraft engine “Remaining Useful Life” (RUL) based on the simulated data of aircraft turbines into two classes:
RUL <= 30 days or RUL > 30 days.
For more information on the task, see:
[https://gallery.azure.ai/Collection/Predictive-Maintenance-Template-3]

For more information on the dataset, see: [hier Referenz zu dem Paper]

We used the data processing pipeline introduced here:
[https://github.com/Azure-Samples/MachineLearningSamples-DeepLearningforPredictiveMaintenance/blob/master/Code/1_data_ingestion_and_preparation.ipynb]

Results:





As seen in the figures above, training time per epoch is for the split LSTM is significantly lower (avg: 16s/epoch) than the normal LSTMs (avg: 18.3s/Epoch). However we see a slight drop in accuracy within the first 50 training epochs. Noteworthy is that the convergence of the normal LSTM model appears to happened earlier, within ~20 Epochs, while the split LSTM sees a slight, but steady increase in validation accuracy during the entire training. This could implicate that split LSTM models will take more iterations to fully train, which can offset the increase gained by reduced epoch training time. However, given accuracy the prediction accuracy stays comparable, it is likely we still see lower computation time, when processing input “on the fly” during production, where time can be of significant importance.

Constraints :

Note that simply splitting the data and passing it in seperate LSTM cells is not exactly the research idea, but was done to model the approximate behavior of such models, while still being able to quickly implement them using the Keras framework (with Tensorflow backend).
Also note that the model architecture was chosen rather randomly. Especially the architecture of the split LSTM is assumed to benefit from taking information locality into account. This was not done in this experiment, since the researcher is no expert of aircraft turbines and thus unable to provide valid information about which sensor data can be aggregated in a meaningful way.