**Problem statement:** Analyzing changes in evolving data fromadvanced habitation systems.

The goal of this task is to create a framework that monitors and provides change detection in a multimodal system. **Our objective** is two-fold (a) To provide the mining of patterns from older data as changes in data could reflect long term/ previous trends. (b) Mining for patterns over near data – as recent changes in data could indicate the recurrences of the previously known pattern or an upcoming event.

**Reported outcomes:** An exhaustive survey of related datasets that capture multi-modal environments/systems.

**Specific Aims:**

1. Create a CNN architecture to make an estimation of Remaining Useful Life from high-level representation of the data sequence
2. Structuring pieces of the architecture:
   1. Saving and loading the weights of trained models
   2. Preparing the preprocess of the data, for when the model is finished and switching to a data that’s is close or same as the one on the paper matters
   3. Looking at what types of analysis could be done with the results from the CNN

**Key Accomplishments:**

1. Create a CNN architecture to make an estimation of Remaining Useful Life from high-level representation of the data sequence
2. Prepare to transition from a mock dataset to a dataset that is closer/same to that used in the paper

**Red Flags:**

1. The CNN architecture is complete but its two parts are disconnected and some details are still missing
2. All the results shown on the report comes from a mock dataset (fashion\_mnist), which comes with tensorflow
3. No optimizing of code was done so far

**Future Work:**

1. Get the CNN running on data that is closer to the one on
2. Finish building the CNN architecture

**Timeline (tentative timeline for the upcoming week)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Future Specific aims** | **10/16** | **10/17** | **10/18** | **10/21** | **10/22** | **10/23** |
| Connecting the CNN architecture |  |  |  |  |  |  |
| Using other dataset |  |  |  |  |  |  |
| Analysing results |  |  |  |  |  |  |
| Optmizing code, if necessary |  |  |  |  |  |  |

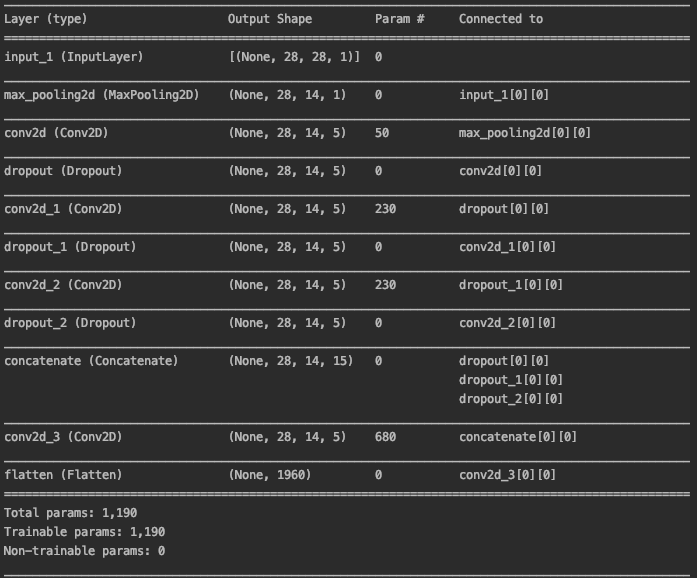
**References:**

[1] Xiang Li, Wei Zhang, Qian Ding (2018). Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction

[2] Keras Documentation. https://keras.io/

**Appendix A**

**Results**

****

**Fig 1 – First part of the CNN architecture**

The first part of the CNN is implemented and running as shown in Fig 1. It is doing the multi-scale feature extraction and its output is a high-level representation of the input data. The only two differences from the architecture of the paper is that is not being initialized with a *Xavier* normal initializer and the learning rate of the system is not manually defined.

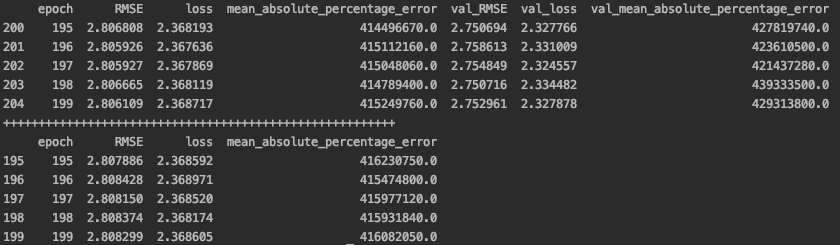
The weights of the system are being saved and are able to restore the system to a state that is as close as possible as when it had just finished training. Fig 2 and 3 shows the model being restored with two different weights, from two different training sessions. The two training sessions were done sequentially, in a scheduled fashion as it takes about three and a half hours to complete each training session.

****

**Fig 2 – Model restoration (with validation)**

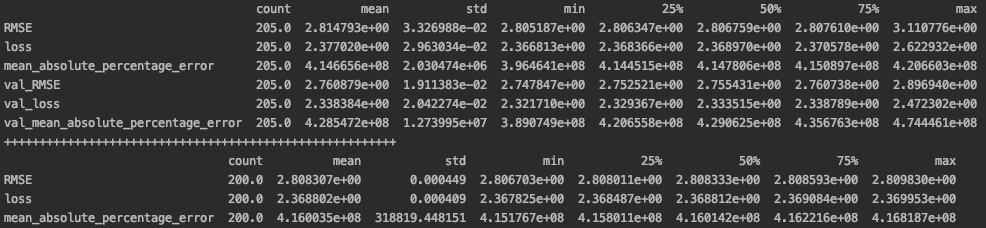
****

**Fig 3 - Model restoration (without validation)**

****

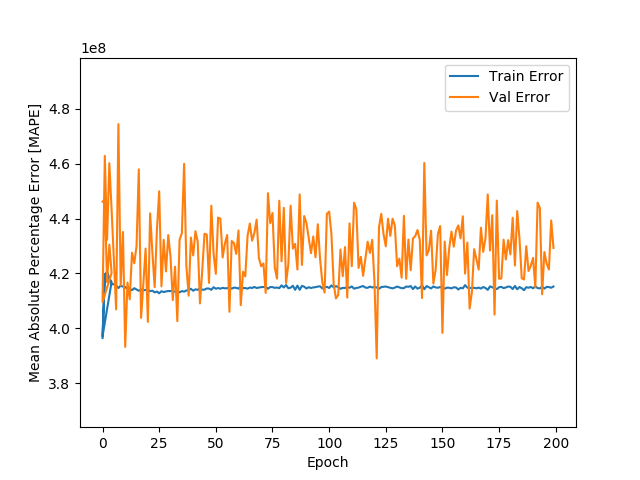
**Fig 4 – The tail of the results. Above the “+” signs is the first model and below the “+” signs is the second model, both at the end of training**

As much as the second model had slightly different weights, Fig 5 confirms (and Fig 8 and 9 shows) that the second model training was most likely just a continuation of the first model training. The standard deviation was very different from the first model. This issue will be addressed.

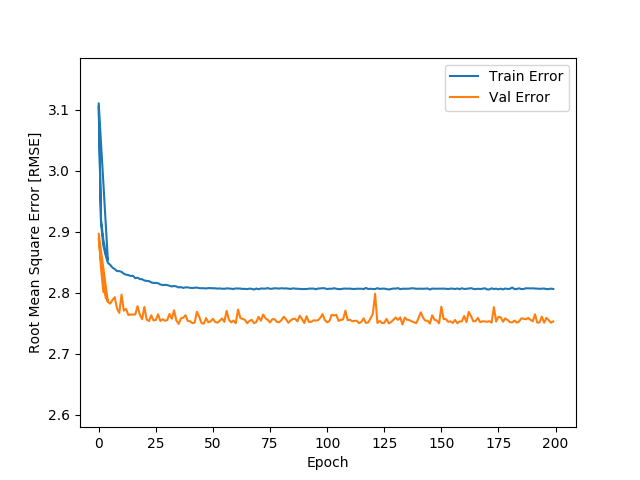
****

**Fig 5 – Statistics of the results**

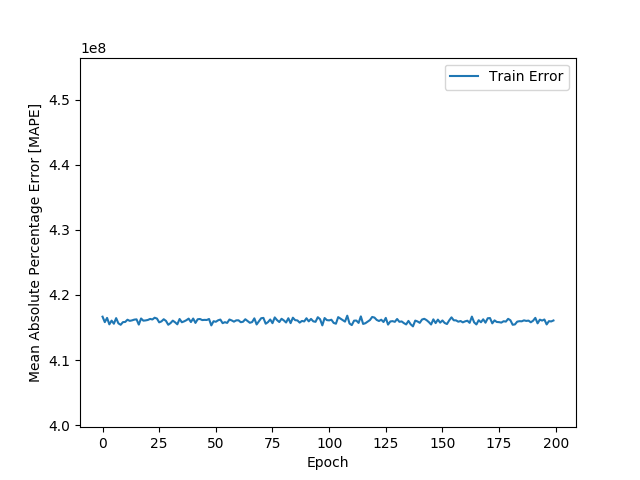
Next, it will be discussed future work.

****

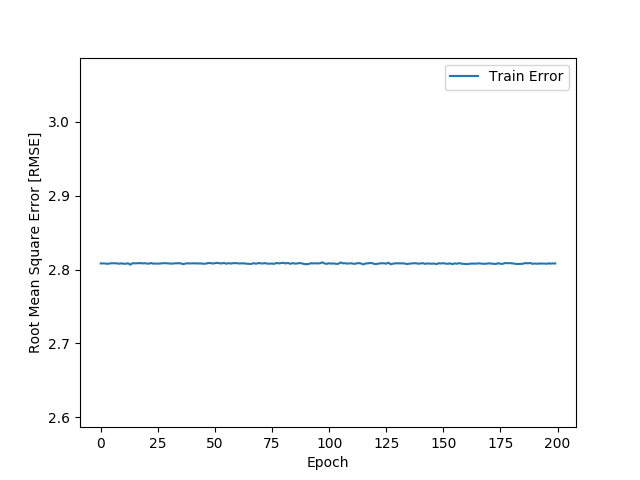
**Fig 6 – MAPE of first model**

****

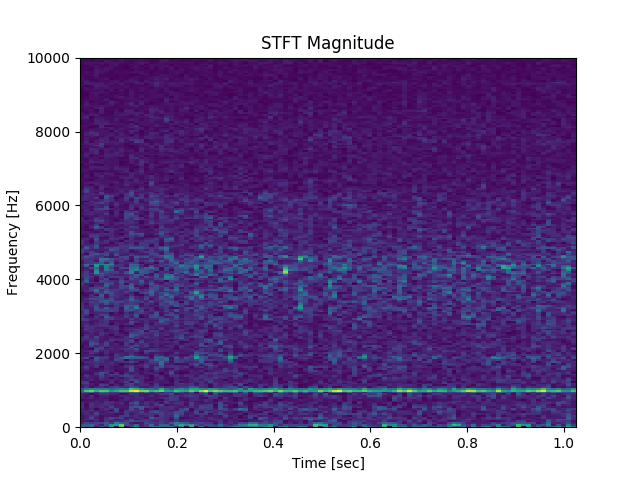
**Fig 7 – RMSE of first model**

****

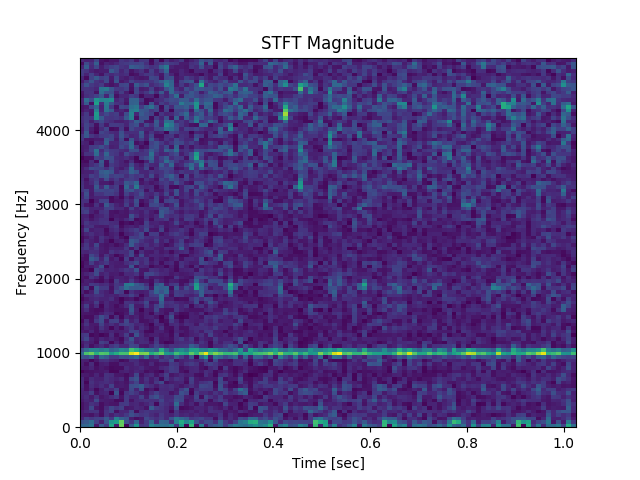
**Fig 8 – MAPE of second model**

****

**Fig 9 – RMSE of second model**

****

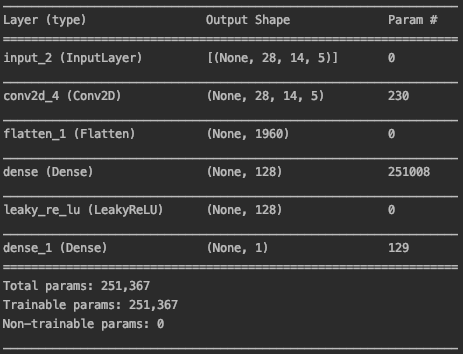
**Fig 10 – Bearing signal after STFT (before denoising)**

****

**Fig 11 – Bearing signal after STFT (after denoising)**

The datasets that are intended to be used to put the architecture to test are two bearings datasets in which the bearings were run from test to failure. One dataset is the same as the one presented on the paper, for result comparison, and the other is believed to be very close in nature of that of the paper and similar results are expected from it as well.

Fig 10 and 11 show the signal of a bearing in the beginning of a test, after applied Short-Time Fourier Transform (STFT), with a Hamming window of 20ms and shift step of 10ms. The signal is 1 second long, with a sampling rate of 20Hz and 20480 samples (i.e. rows in the file). The proposed denoising consists of ignoring the frequency range above half the Nyquist frequency. The denoised signal is going to feed the CNN architecture.

****

**Fig 12 – Second part of the CNN architecture**

The second part of the CNN architecture gets the high-level representations of the data, which have been added all together (element-wise addition) and outputs an estimation of the Remaining Useful Life (RUL) of the bearing.