

Results

Dereck Piché Guillermo Martinez Jonas Gabirot

April 2, 2023

1 Results

1.1 Baseline: XGB

We trained and tested an Extreme Gradient Boosting (XGB) regressor using DNA sequences as features and their respective free-energy levels as labels. 900 000 samples were used for training and 100 000 samples for testing. The model's train and test performance were assessed by calculating the Mean Square Error (MSE) between the predicted and actual free energy levels given specific DNA sequences. 0.993166 and 1.016363 MSE were achieved on train and test evaluation respectively. These results will be used as baselines to evaluate further models' predictive performance.

1.2 Baseline: MLP

We built a standard MLP model as a deep learning baseline for our regression task. It consists of 21 fully connected layers. We trained the model on 900,000 DNA sequences of length 30 and tested it's MSE loss on the remaining 100,000 training examples. The DNA sequences were preprocessed into one hot encoded vectors of length 120. The accuracy throughout the epochs can be seen in figure

1.3 LSTM

We built a model using the regressive LSTM architecture for our regression task. Our regression LSTM is made up of a 2 LSTM layers and one fully connected layer. We trained the model on 900,000 DNA sequences of length 30 and tested it's MSE loss on the remaining 100,000 training examples. The DNA sequences were preprocessed into one hot encoded vectors of length 120. The accuracy throughout the epochs can be seen in figure

1.4 Regression Transformer

We built a model using the main components of the Transformer architecture proposed in [?] for our regression task. Our regression transformer is made up

Figure 1: MLP loss

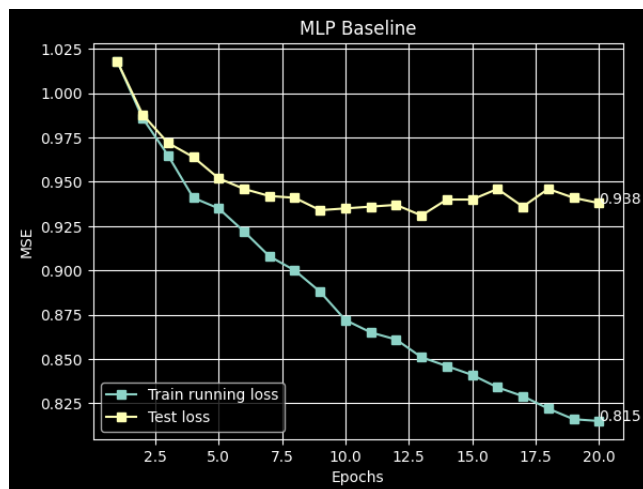


Figure 2: LSTM loss

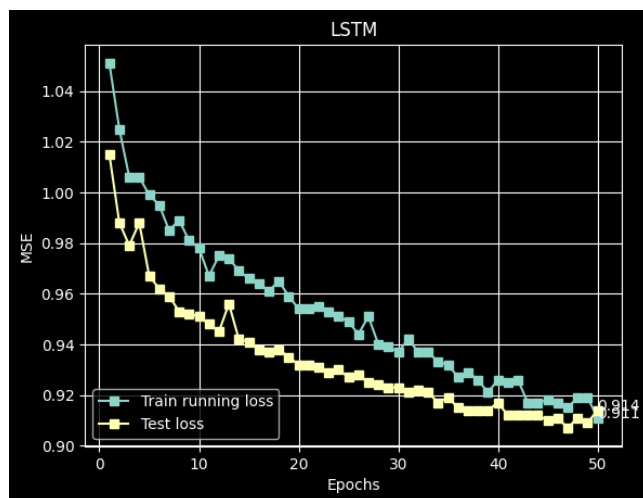
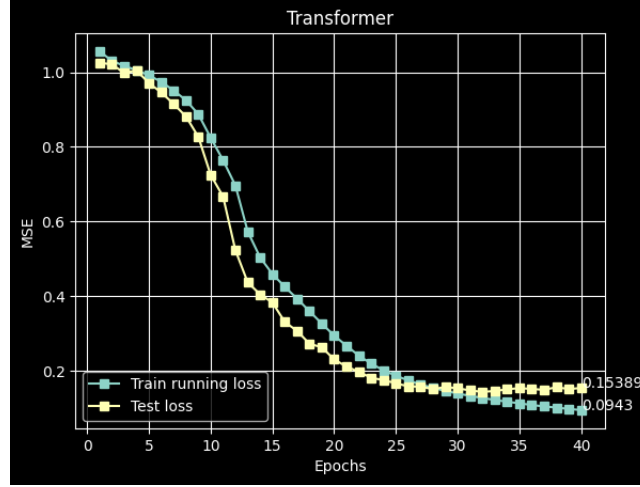


Figure 3: Transformer loss



of a sequence of encoder layers. Each of the encoder layers contains 8 attention heads with a ReLU activated linear layer. We trained the transformer on 900,000 DNA sequences of length 30 and tested it's MSE loss on the remaining 100,000 training examples. The transformer used the same positionnal encoding as the one proposed in [?]. The DNA sequences came in string form. Each of the 4 tokens were transformed into an integer and fed into a learned embedding layer that transformed them into vectors of 16 dimensions. The number of dimensions of the embeddings is a bit arbitrary, but since it works really well we left it there. The accuracy throughout the epochs can be seen in figure 1.4.

2 Analysis

2.1 Comparisons

3 Conclusion

Figure 4: Loss comparison

