Creating a Performance Predictor

Singh Dilpreet: 4563471

Project goals

- Predict final validation accuracy given network logs for the first n epochs.
- Experiment with different types of data scaling and transformations.
- Explore and exploit different architectures and compare respective performances.
- Use hyper-parameter optimization technique BOHB to tune the network and find optimal λ* for the task

Dataset : Task A

- Available : 2000 run configurations on the open ml dataset fashion_mnist, each entry comprises a set of hyperparameters and learning curve logs along with gradient statistics for 51 epochs.
- Extracted : 7 hyperparameters together with validation accuracy log for the first 10 epochs were used to target the final validation accuracy at epoch 51.
- Alternatively, Min max scaling along with several other gradient statistics and variable length train, validation/test sets were used but these did not yield improvements with the selected methodology.

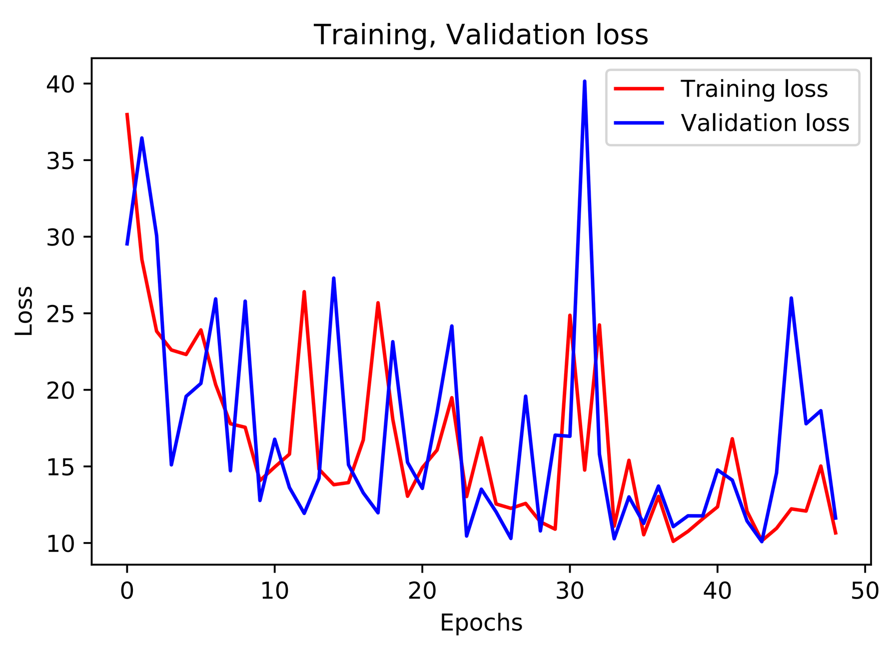
Methodology

- **Task A : Regression problem.**

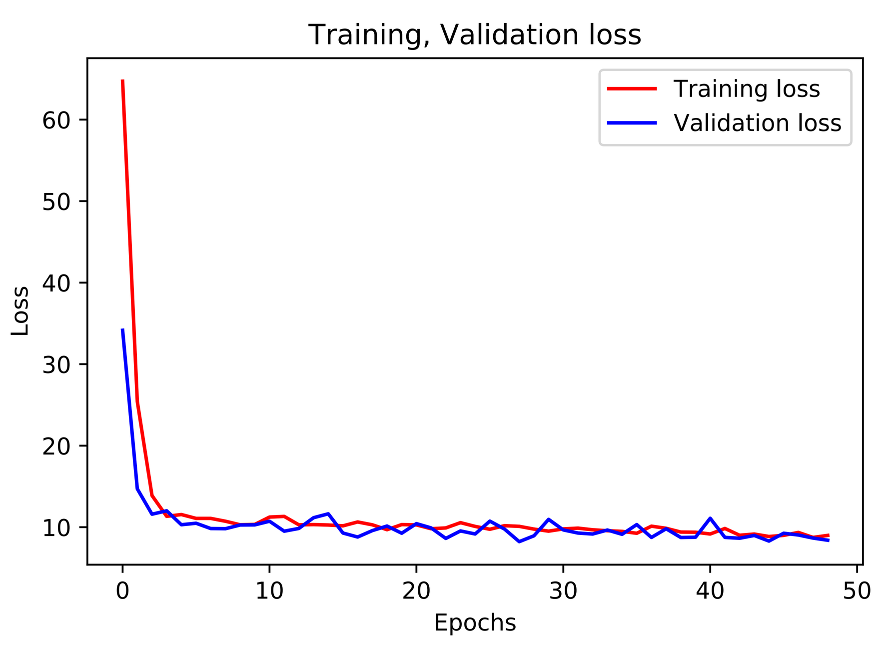
standard_CNN((pool):
MaxPool1d(kernel_size=2, stride=1, padding=0, dilation=1, ceil_mode=False)
(relu): ReLU()
(conv1): Conv1d(1, 64, kernel_size=(2,), stride=(1,))
(fc1): Linear(in_features=960, out_features=1, bias=True))
Final Validation loss: 11.6302
Final test loss: 9.1977

standard_LSTM(
(lstm): LSTM(1, 100)
(fc1): Linear(in_features=1700, out_features=1, bias=True))
Final Validation loss: 8.0053
Final test loss: 6.2994

CNN



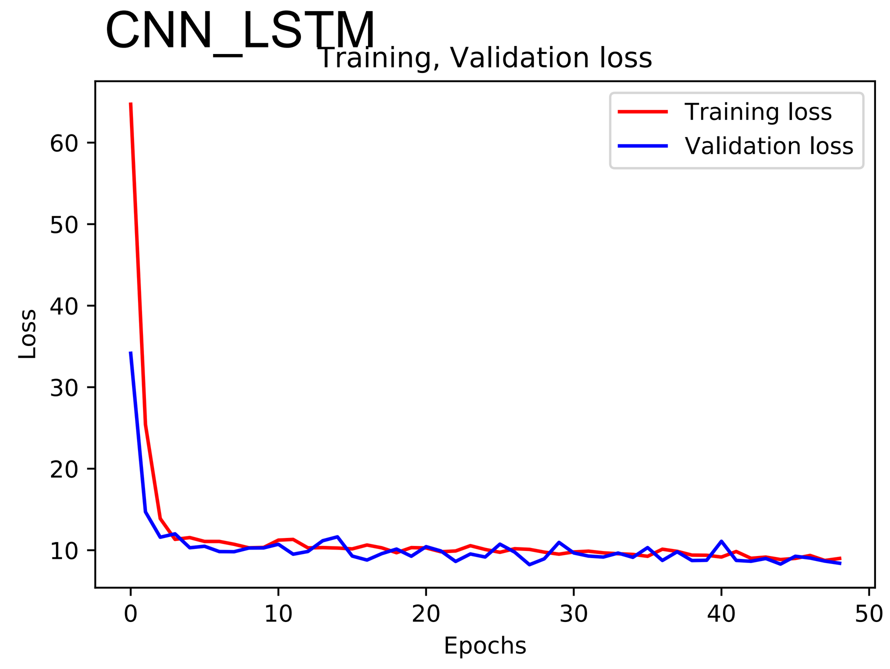
LSTM



- Another network wherein CNN layers were provided with the 7 valued vector for hyperparameters and the result consequently used to update the hidden state of 1 layer LSTM which in turn cycles the time series was employed.

CNN_LSTM((pool): MaxPool1d(kernel_size=2, stride=1, padding=0, dilation=1, ceil_mode=False)
(relu): ReLU()
(conv1): Conv1d(1, 16, kernel_size=(1,), stride=(1,), padding=(1,))
(lstm): LSTM(1, 128)
(fc1): Linear(in_features=1280, out_features=1, bias=True))
Final Validation loss: 8.3996
Final test loss: 6.3966

CNN_LSTM

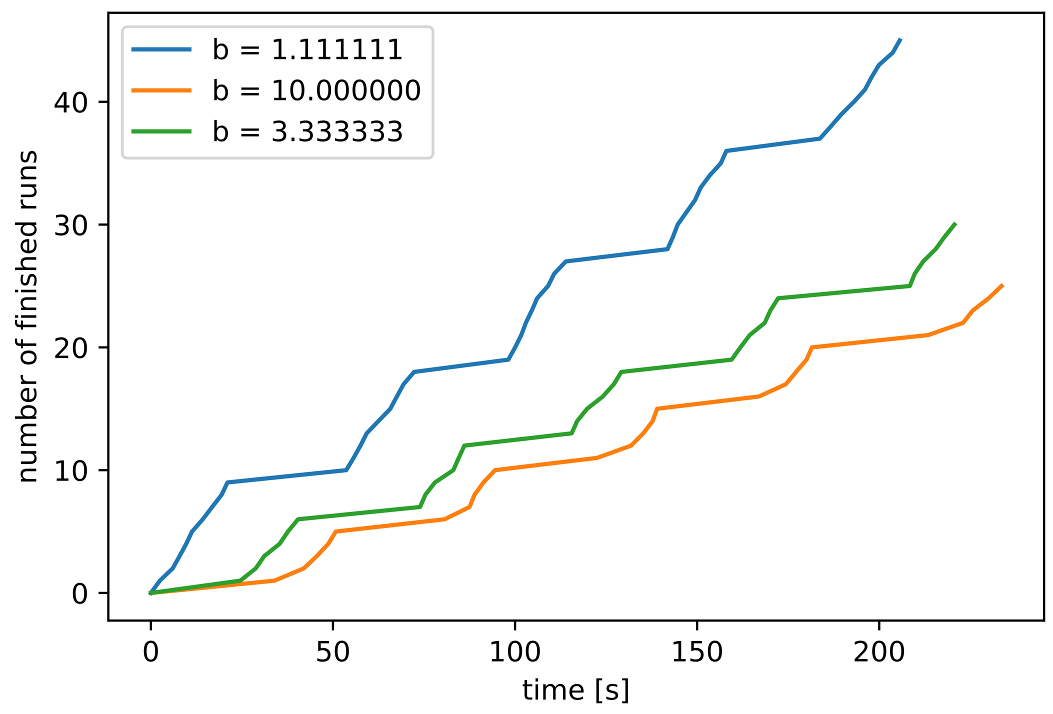


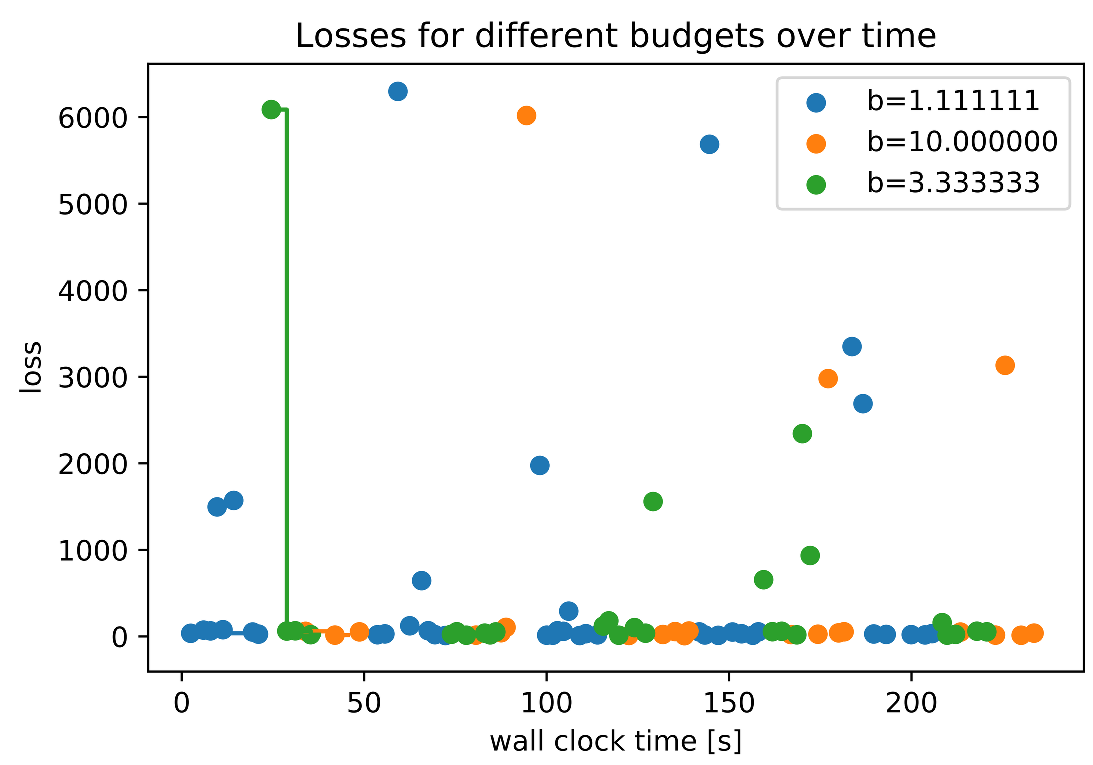
BOHB Statistics

A total of 75 unique configurations where sampled.
A total of 100 runs where executed.

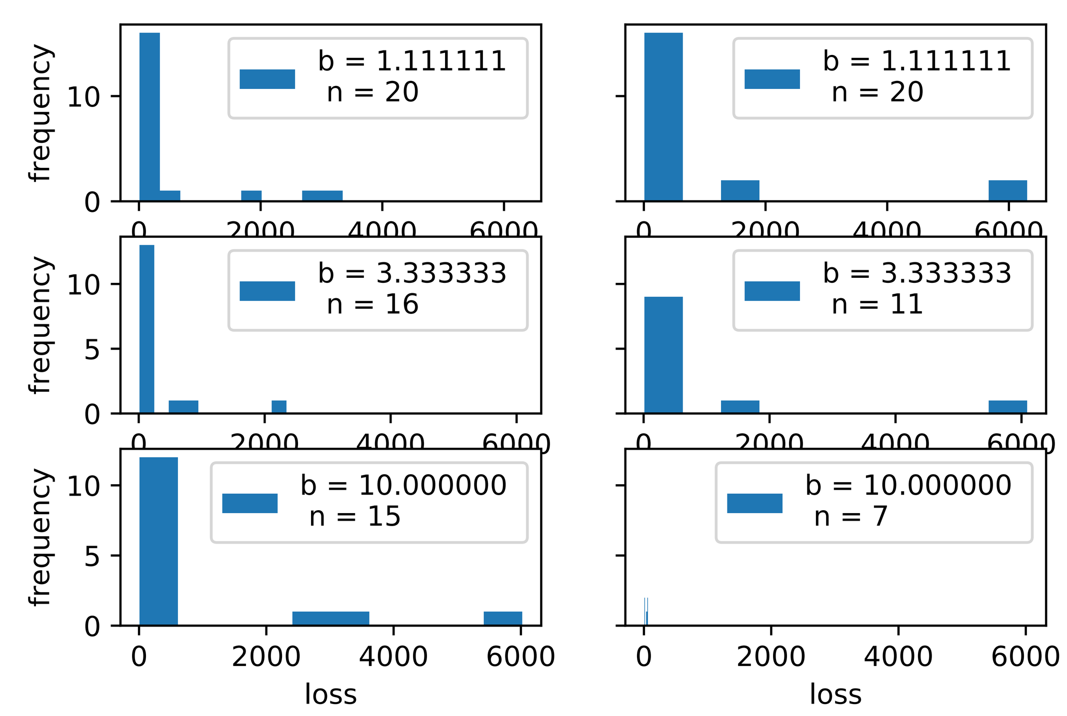
Best Config : {'lr': 0.04075527413023171, 'num_conv_layers': 2, 'num_filters_1': 40, 'optimizer': 'Adam', 'num_filters_2': 38}
The best config has the lowest final error with 8.6917 and test error 6.9943

Conv1d(1, 40, kernel_size=(2,), stride=(1,))(relu1): ReLU()
(pool1): MaxPool1d(kernel_size=2, stride=1, padding=0, dilation=1, ceil_mode=False)
(conv2): Conv1d(40, 38, kernel_size=(2,), stride=(1,))
(relu2): ReLU()
(pool2): MaxPool1d(kernel_size=2, stride=1, padding=0, dilation=1, ceil_mode=False)
(flatten): Flatten()
(linear): Linear(in_features=494, out_features=1, bias=True))





Loss of model based configurations (left) vs. random configuration (right)



For LSTM - Best Config : {'lr': 0.002299783496406268, 'num_hidden_neurons': 94, 'num_lstm_layers': 1, 'optimizer': 'Adam'}

The best config has the lowest final error with 10.7887 and test error 9.2822

standard_LSTM(
(lstm): LSTM(1, 94)
(fc1): Linear(in_features=1598, out_features=1, bias=True)).