Deep Learning - Assignment 1

Diez Pérez María i6268086 - Vico Gianluca i6183186 - Gianzina Alexandra i629435

1 Introduction

The goal of this assignment is to produce a neural network model that is best suited for training a given dataset. This dataset consists of a 1000-by-1 vector of 8-bit unsigned integers. The values of the vector are shown in Figure 1.

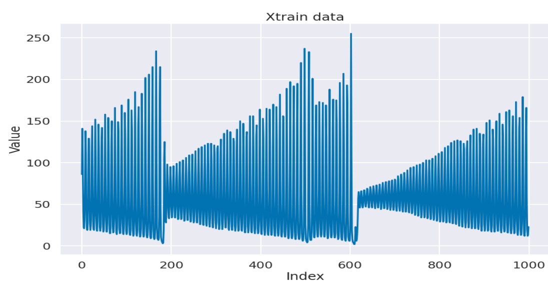

Figure 1: Real-life dataset of a laser measurement

Furthermore, we must train our model to predict a data point one step ahead for a fracture of the data. To solve this issue, we implemented two different architectures to train our data. In the following sections, we will explain why we chose these two architectures and present the results.

2 Implementation Details

For this assignment we use two separate architectures to train our data and predict a data point a step further. We take into account that we are handling sequences, so we first created an LSTM recurrent neural network (RNN) architecture [1]. For this model, we have a varying input size, which depends on the window size of the data, a layer with 10 LSTM units and a linear layer to generate the output. The output of the model is simply the next value in the time series. In addition, we noticed that the size of the offered training dataset is minimal.

As a result of the limited data, we chose to develop a smaller architecture. In this case, we have a neural network with 2 hidden layers with 50 and 10 hidden units. The activation function is ReLU [2]. The inputs and output are the same of the LSTM model.

Since we have little data, we are using a large value for the dropout [3] (0.5). Then we examine the results of each model derived from these two architectures and choose the best model to move forward with the predictions. For each architecture we scale the data in the range [0,1]. The predictions of the models are scaled back to the original range. Also, we subtract a small portion of our training data (the last 50 samples and the window size selected for the model) in order to have a validation set.

The loss function is MSE for all the experiments and the optimizer is Adam.

3 Training LSTM architecture analysis

We train each model for 1000 epochs and see that the minimum loss is reached after 400 epochs (see Figure 2). In Figure 3 we plot the predicted value versus the validation data value. We also observe that in Figure 3, the LSTM architecture appears to predict with a similar error in each model. However, when we examine the Mean Squared Error (MSE) of the models, we find that input sequences of size 20 have the lowest MSE.

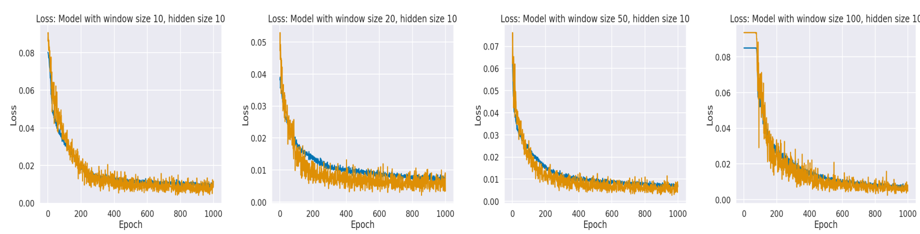

Figure 2: LSTM architecture loss functions with varying sizes for the input sequences

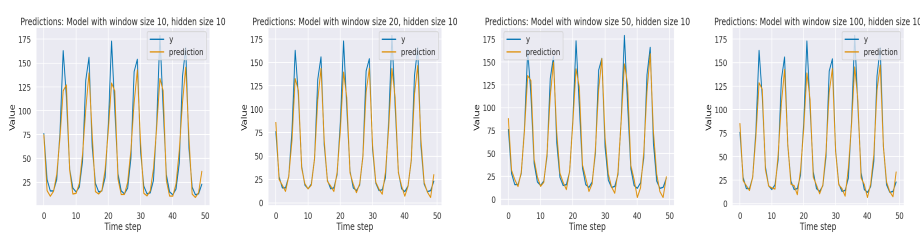

Figure 3: LSTM architecture prediction values for input sequences of size 10, 20, 50 and 100

4 Training ReLU architecture analysis

We decide to create a model using ReLU since the amount of training data is small. The model is trained for 1000 epochs and on different input sequences (10, 20, 50 and 100 length). Figure 4 shows, in most cases, the loss is smaller in the second and forth graph. The predicted values are similar to the output values (Figure 5), and if we compare them with the results using LSTM, as expected, we find more similarities with ReLU architecture.

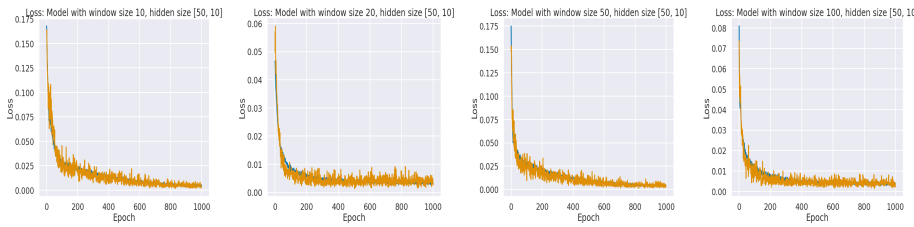

Figure 4: Smaller NN architecture loss functions with varying sizes for the input sequences

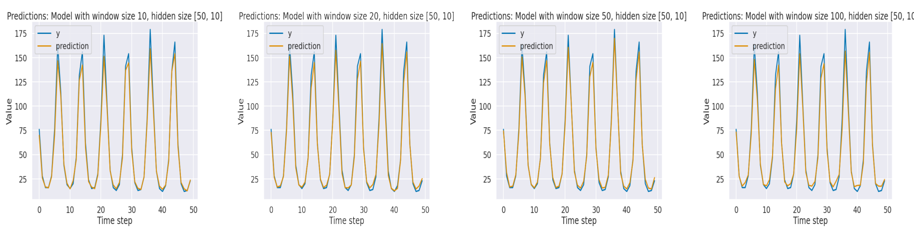

Figure 5: Smaller NN architecture prediction values for input sequences of size 10, 20, 50 and 100

5 Prediction of the next 200 points

Once we were through the previous models, we decide to use the ReLU model to predict the following 200 points of data. In this case, the window size used is 20 and we train the model for 380 epochs, because during the experiments we can observe that the minimum loss is reached before 400 epochs.
The obtained prediction is shown in the Figure 6. If we analyse the given dataset, Figure 1, we can see how the points are increasing until a severe decrease after which the points start again to increase. However, the predicted points are not increasing as we expected, this is because the data tends to reach the mean value, which in this set of data is 59.894.

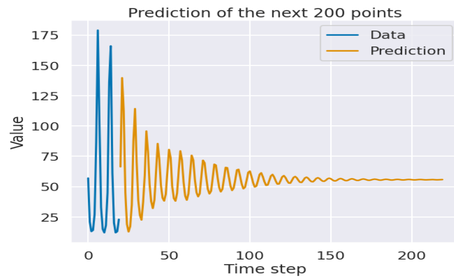

Figure 6: Prediction values of the Smaller NN architecture with the lowest MSE

6 Test with the best model

Finally, we evaluate the ReLU model of window size 20 on the test data and find a lot of similarities between the predicted and test data. In addition, we measure the MAE (33.86), MSE (2330.83) and RMSE (48.27) for the model.

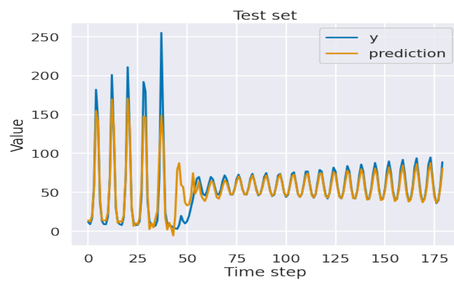

Figure 7: Prediction values on the test set

7 Conclusion

We analysed two different NN architectures that we used to train our data. Due to the fact that the dataset is small, the architecture that had the better performance was the ReLU with window size 20. This ReLu model was used to test our data. The predicted values and the test values were very similar, so we conclude that we chose the correct model for training, validating and testing our data.

References

[1] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[2] Abien Fred Agarap. “Deep learning using rectified linear units (relu)”. In: *arXiv preprint arXiv:1803.08375* (2018).

[3] Nitish Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: `http://jmlr.org/papers/v15/srivastava14a.html`.