Deep Learning - Assignment 2

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

1 Introduction

The goal of this assignment is to produce a neural network model in a recursive fashion to predict the temperature of a city given previous meteorological measurements. In addition, to find a strategy to determine which features are the more relevant for the results, and to use visualisation techniques to show them. This dataset consists of a 70128 by 4 by 5 vector that contains the information of the time, the location and the meteorological features respectively. The temperature of the four locations are shown in Figure 1.

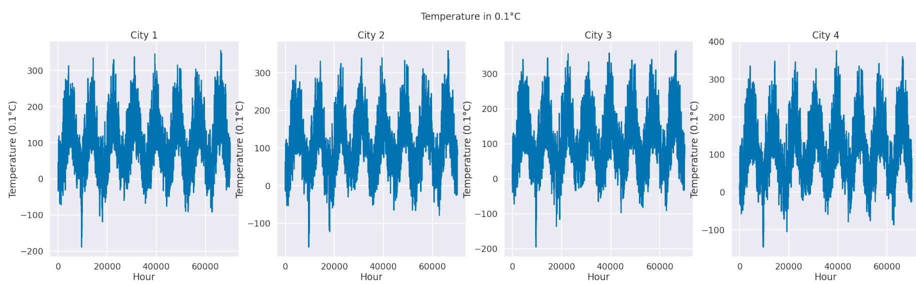

Figure 1: Real-life weather data-set in 4 different locations

2 Implementation Details

The dataset available contains the measurements of 5 meteorological indicators for 4 cities. We treat each city independently, so that each element in the dataset is a matrix 5 by the size of the window size. The different models are trained to predict the next time step, which is a 5x1 vector. We need to be able to predict all 5 features to recursively multiple time steps. We split the dataset in three sets: train, validation (second to last week of measurements) and test (last week). Also, each feature is scaled in the range [0, 1] and the prediction of the model are scale back to the original range.

For the model, we first train an LSTM model (1 LSTM layer with 10 cells each and a fully connect layer with 5 outputs) on three different window size (10, 20, 50), similar to the previous assignment. For the convolutional network, we test a residual network. We used three residual blocks made up of convolution + batch normalization + ReLU + convolution + batch normalization. Each convolutional layer contains 10 3x3 filters. For the output, we use a fully connected layer similarly to the LSTM model.

We decided to keep the model small to prevent the risk of not having enough data.

3 Training the models

We train the LSTM model and CNN model with three window sizes of 10, 20, and 50. LSTM was trained for 250 epochs, however, as the top row in Figure 2 shows, the minimum loss is reached after about 25 or 35 epochs. In the row below of Figure 2, it is plotted the predicted value in yellow among the validation data in blue. Looking at the Mean Squared Error (MSE), we find that the lowest is with an input sequence of 20.

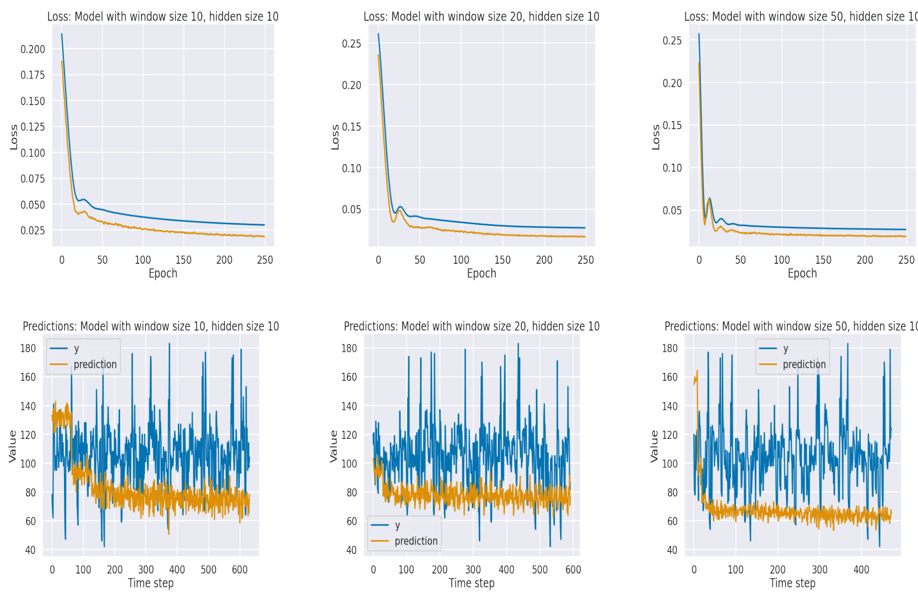

Figure 2: LSTM architecture loss functions and predictions with varying sizes for the input sequences

On the CNN model, we train it during 1000 epochs finding that the minimal loss is reached before 100 epochs. The loss is plotted in the top row of Figure 3. Comparing the predicted values shown in the bellow row in Figure 3, with the predicted values obtained with the LSTM model, we can see that the performance is more precise. In addition, if we examine the Mean Squared Error (MSE) of this model, we find that input sequences of size 10 have the lowest MSE, and also, it is much lower than the MSE found with LSTM.

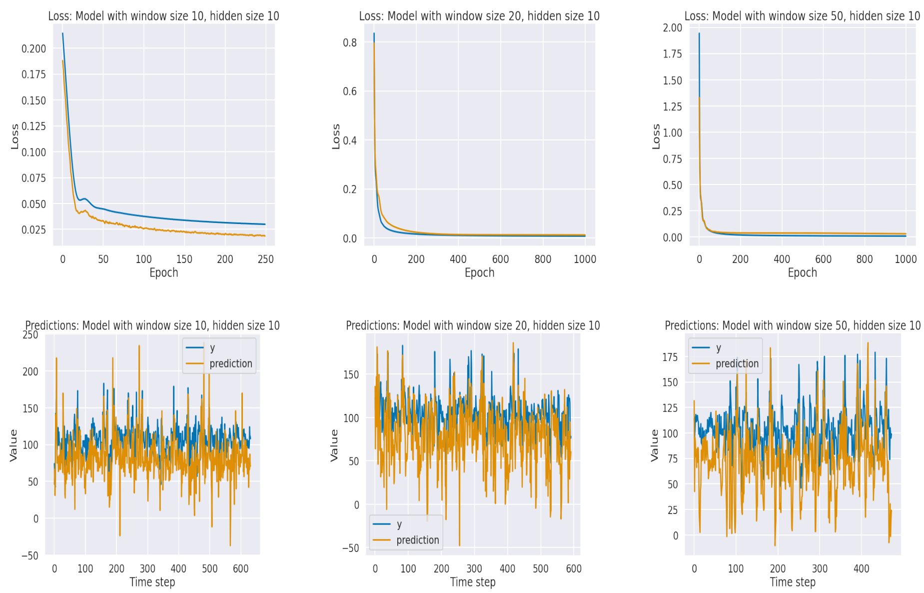

Figure 3: CNN architecture loss functions and predictions with varying sizes for the input sequences

The errors are the following: MAE = 38.06, MSE = 3293.61 and RMSE 57.39.

4 Temperature prediction of the next week

To predict the next 168 data points, we utilize a CNN model where we measured low errors. We input the features from the last city. The window size of the model is set to 20, as a result, the input size is equal to 20x5, and the hidden size is 10. We train the model for 200 epochs. To represent all features, each prediction is stored in a 1x5 vector. Figure 4 illustrates the resulting predictions for the temperature feature.

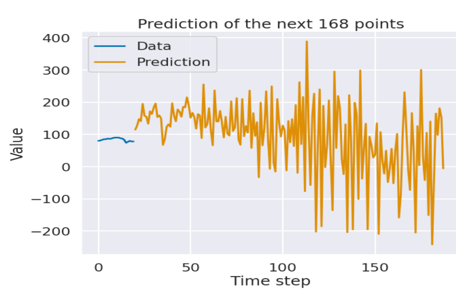

Figure 4: Prediction values for the next week's temperature

In Figure 4 we observe that at the beginning of the week the temperature deviation from one time step to another is not large. Later the predictions become very volatile and the deviations become larger. Normally, in real life, the temperature from one step to the next will not drop or increase by 40 degrees.

5 Experiments between different features

Figure 5 displays the model MSE for all possible feature removals and Table 1 shows for each index (0-29) which features have been removed. According to the following Figure we notice that if we delete either the temperature (index 1), the dew point (index 3) or the air pressure (index 4) the MSE is almost the same. It seems that if we delete only the wind speed (index 0) the output of the model will affected less than the deletion of the other features. We also observe that if we remove a specific combination (index 22 and index 29) of features, as seen in Figure 5, then the MSE will most likely increase in general. Also, we notice that if we have to keep one feature to predict the outcome, then the index 28 that keeps the wind direction has error that is lower than any other feature we could keep. Thus, as shown in Figure 5, no feature alone plays an important role in the output of the model, but the combination of different features affects the outcome.

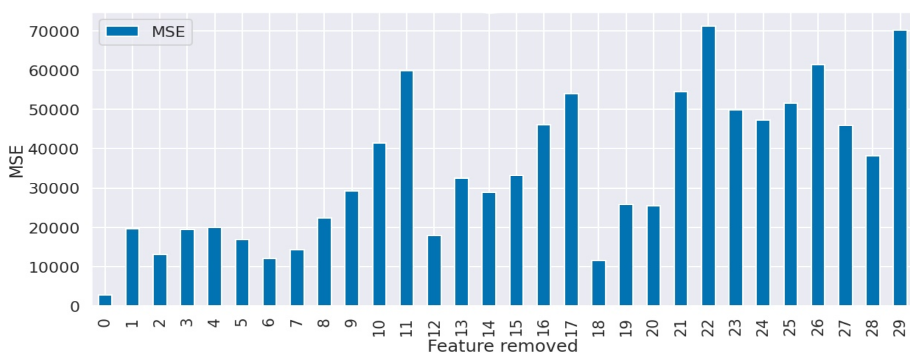

Figure 5: MSE for removed features. Table 1 shows for each index (0-29) which features have been removed.

6 Conclusion

We tested 2 different architectures and selected a low loss model to perform the prediction. We realized that predicting the temperature of an approaching day is more accurate than a day after 1 week. Also, we observed that almost all input features are important to predict the outcome with higher accuracy.

7 Appendix

Index	Removed features	MSE
0	(WIND_SPEED,)	2856.533203
1	(WIND_DIR,)	19648.166016
2	(TEMPERATURE,)	13172.108398
3	(DEW_POINT,)	19484.216797
4	(AIR_PRESSURE,)	19954.589844
5	(WIND_SPEED, WIND_DIR)	16858.748047
6	(WIND_SPEED, TEMPERATURE)	12142.556641
7	(WIND_SPEED, DEW_POINT)	14334.765625
8	(WIND_SPEED, AIR_PRESSURE)	22368.171875
9	(WIND_DIR, TEMPERATURE)	29306.632812
10	(WIND_DIR, DEW_POINT)	41468.800781
11	(WIND_DIR, AIR_PRESSURE)	59890.421875
12	(TEMPERATURE, DEW_POINT)	17903.380859
13	(TEMPERATURE, AIR_PRESSURE)	32520.636719
14	(DEW_POINT, AIR_PRESSURE)	28816.005859
15	(WIND_SPEED, WIND_DIR, TEMPERATURE)	33247.234375
16	(WIND_SPEED, WIND_DIR, DEW_POINT)	46044.097656
17	(WIND_SPEED, WIND_DIR, AIR_PRESSURE)	53925.789062
18	(WIND_SPEED, TEMPERATURE, DEW_POINT)	11465.565430
19	(WIND_SPEED, TEMPERATURE, AIR_PRESSURE)	25790.181641
20	(WIND_SPEED, DEW_POINT, AIR_PRESSURE)	25534.623047
21	(WIND_DIR, TEMPERATURE, DEW_POINT)	54516.281250
22	(WIND_DIR, TEMPERATURE, AIR_PRESSURE)	71162.171875
23	(WIND_DIR, DEW_POINT, AIR_PRESSURE)	49784.859375
24	(TEMPERATURE, DEW_POINT, AIR_PRESSURE)	47263.429688
25	(WIND_SPEED, WIND_DIR, TEMPERATURE, DEW_POINT)	51614.421875
26	(WIND_SPEED, WIND_DIR, TEMPERATURE, AIR_PRESSURE)	61347.871094
27	(WIND_SPEED, WIND_DIR, DEW_POINT, AIR_PRESSURE)	45908.570312
28	(WIND_SPEED, TEMPERATURE, DEW_POINT, AIR_PRESS...	38163.867188
29	(WIND_DIR, TEMPERATURE, DEW_POINT, AIR_PRESSURE)	70179.554688

Table 1: Table of removed features