CSC578 Section 901
Class project: Part A Kaggle competition
Amy Aumpansub

Display name: Amy_A
Ranking: 4 (Public)
Score: 1.56475
Date: Jun 5th , 2020

Introduction

The weather dataset was recorded by the Max-Planck-Institute for Biogeochemistry. The dataset contains a date-time and 14 features recorded in each hour. The dataset is used and transformed to the form of 48-hour roll out to develop the deep learning model to solve a Multivariate Single-step Time-series problem to forecast the weather which is an aim of this project. The details on dataset and models can be found on Amy_csc578_weather.ipynb.

Model Development

Several parameters were tuned to find the optimal model in which each parameter was adjusted one at a time in each experiment, fixing other parameters constant. The best hyperparameter value was carried over to the next experiment which is chosen in regard to its performance on loss convergence, overfitting, total time taken to fit the model . The three models from experiments were further discussed in this section. All data were normalized with min-max values. Thus, the MAE loss reflects the error on normalized data. Predicted values were later reinverted to the original scale.

The Baseline Model

Baseline Model
MAE on Z-Score Data

Epoch	MAE Loss	
	Train	Validate
1	0.270	0.206
10	0.167	0.165
50	0.079	0.097
100	0.052	0.074

The base model has 2 LSTM layers in which each layer contains 100 recurrent units and takes a normalized input of shape (48,4) where 48 is a timestep and 4 is the data dimension representing 4 features: air pressure, temperature, air density, and wind direction. The activation function for all layers except an output layer is ReLU. The based model was trained in 10 epochs. The optimizer is "adam" and the loss is "MAE". The mean absolute layer is chosen because it is appropriate to measure averaged error (the difference between predicted values and real target values) in this regression problem. The LSTM was chosen instead of RNN as LSTM performs better when dealing with the long-distance dependence in which LSTM has a good memory and remembers longer sequences which is suitable for our time-series dataset in which each record contains 48-rolling window. The MAE loss shows that the model loss decreasing faster in the beginning and still has some deviation between the loss of training and test data. After 20 epochs, the overfitting begins to occur, so other hyperparameters were tuned in other experiments to improve the models as follows:

Hyperparameter Tuning

1. Number of recurrent units: As the number of recurrent units, we expect to see the lower loss, but it will make the network more complex, increase the training time, and probably lead to the overfitting. Beginning with the number of 100 nodes, the MAE loss gradually decreases from 0.167 to 0.156 in 256 nodes at epoch 10 as expected. The loss also gradually decreases as the number of nodes increases. Compared to other numbers, the 256-node model has the lowest loss rate of validation set which is closed to those of training set. Thus, the number of 256 nodes was carried over to the next experiment.

2. Number of batch size: The base model was fit with 32 batch size so the batch number was increased to 64. The increases in number of batches are expected to improve the model. However, the experiment shows that the loss is smaller with 64 batch size which is more situatable in prediction the long sequence of our time series data.

Model 3

Model #3

Epoch	MAE Loss	
	Train	Validate
1	0.2289	0.2051
5	0.1800	0.1846
10	0.1604	0.1632

3. The bidirectional LSTM was used to develop the model #3. This model contains 2 bidirectional LSTM layers of 256 units. The optimizer, activation functions, and loss are same as the base model. The Bidirectional LSTM is expected be more effective than regular RNNs as it can preserve the information and predict outputs in both in the past and future context, so it is situatable for our time-series forecasting. The model shows an improvement in which the loss for both training and validation set are lower than those of LSTM model. The loss also decreases as the process goes. The model was fit with only 10 epochs, so the model needs to be trained longer to improve the loss, but the overfitting issue may occur, so we need to be further examined.

Model 4

Model #4

Epoch	MAE Loss	
	Train	Validate
1	0.2392	0.1981
5	0.1867	0.1851
10	0.1692	0.1658

4. To solve the overfitting problem, the dropout was added to the second layer on the previous model. The model #4 has 2 bidirectional LSTM layers of 256 units and fitted with the dropout of 0.15. The model with dropout is expected to have less overfitting problem and its loss would converge faster. As shown on the right table, the model #4 shows less overfitting problem when adding dropout to the model.

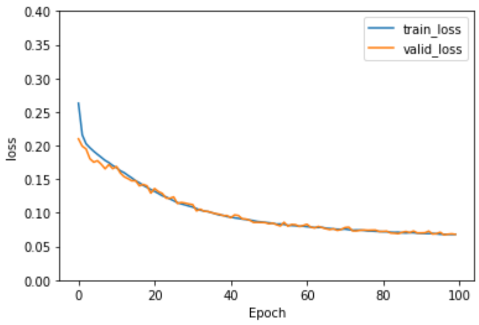
5. To find the optimal model, the model architecture was adjusted by adding one more bidirectional layer and a dropout after each layer to avoid overfitting problem. The best model was selected because it has the lowest MAE for both training and validation set and has less overfitting problem. The best model is discussed below

The Best Model

The best model has two bidirectional LSTM layers in which each layer contains 150 recurrent units and takes a normalized input of shape (48,4). The dropout of 0.15 was added after each layer. The activation function for all layers except an output layer is ReLU. The based model was trained in 100 epochs, so we use 150 units which consumes a more appropriate time to build model than 256 units. The optimizer is "adam" and the loss is "MAE". The model has the lowest loss for both training and validation set. The loss rate decreases smoothly through the whole process. The plot below shows a comparable MAE loss between training and validating data which confirms that the model is not overfitting. However, the addition of dropout causes a little fluctuation of loss for the validating data. The LSTM is appropriate for our time-series forecasting with timesteps. This is because the LSTM comprises both update gate and forget gate and remembers longer sequences, so it is more effective than simple RNNs. Additionally, the bidirectional LSTM is also more effective as it has access to the past as well as the future information due to the bidirectional architecture which has both input sequence and a reverse copy of it, so it trained two input sequences instead of one sequence like regular LSTM. Overall, this model performed the best which has the low loss rate, no overfitting, and has a moderate time taken to build the model.

The Best Model
MAE on Standardized Data Z-Score

Epoch	MAE Loss	
	Train	Validate
1	0.2629	0.2100
10	0.1703	0.1656
30	0.1101	0.1132
50	0.0858	0.0855
100	0.0682	0.0680



Personal Reaction and Reflection

I found this project is very helpful and challenging as it requires the applications of solid knowledge of deep learning. The time-series data with a rolling window are more complex than other datasets. The data preprocessing for creating a rolling window is simpler when following the professor's drawing diagram. I found the Google CoLab to be very useful in fitting the models especially for this project as it takes a long time to fit one model. The results of my experiments are as I expected and aligned with the concepts I learned in this course. For the competition, I found it is very fun and gave me an opportunity to apply my knowledge of deep learning and see what we can achieve based on the same dataset. I found all professor's notes to be very useful to help me understand the concepts more clearly. The assignments fairly reflect what I learned in class. Overall, I like this course and it developed my passion for deep learning. I learned a lot about deep learning, use alternative tools to run the models, and see the real-world applications of deep learning. I very enjoyed this class. Thank you for a great class.