Problem Set 6

Ran Tao

(Dated: April 23, 2021)

Abstract

I was asked to predict temperatures based on historical records. I first used a simple neural network with only two dense layer and a RNN model to predict the temperature of next hour using only one feature(temperature). Then I used this RNN model to predict the temperature of future 24 hours and 72 hours. In the end, I tried to include more features(pressure and humidity) in my prediction model and compared the feature importance of these two features.

I. INTRODUCTION

I was asked to predict temperatures based on historical records. I built a simple neural network and a RNN to predict the temperature of next hour as well as future 24 hours and 72 hours using only temperatures. Then I tried to include more features(pressure and humidity) in the prediction model and compared the feature importance of these two features.

II. DATA EXPLORATION

The dataset has 45013 data and 7 features, namely datetime, temperature, humidity, pressure, weather, wind direction, and wind speed. The value of temperature through time is shown in Fig.1.

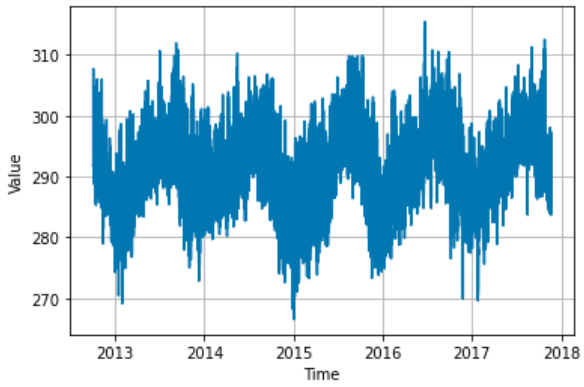

FIG. 1. Temperature values

III. DATA PREPROCESSING

After deleting all rows with NaN data, I tried to prepare the data for our model. The data preprocessing part is very complex, and different models need different data preprocessing.

My first model is a simple Neural Network with two dense layer, and I used historical temperature data to predict temperature of the next hour. I wrote a function that can divide a sequence into multiple input/output patterns, where several time steps are used as input and several time step is used as output.

For example, if I want to use the past 3 time steps to predict the future 1 time step, and the sequence is [10, 20, 30, 40, 50, 60, 70, 80, 90]. Then I can prepare this dataset as shown

in Fig.2. In this report, I will call the left part as X, and the right part as y.

```
[10 20 30] 40
[20 30 40] 50
[30 40 50] 60
[40 50 60] 70
[50 60 70] 80
[60 70 80] 90
```

FIG. 2. Split result 1

Then I used this function to split our dataset, and I used 120 time steps to predict the future 1 time step. Then I split the data and used 70 percent of data as training set and 30 percent of data as validation set. The shape of training X is (31185, 120), validation X is (13366, 120), training y is (31185, 1), validation y is (13366, 1). Then I used this data to training the model.

My second model is a RNN model, and I used historical temperature data to predict the temperature of the next hour. I also used the same function as following to prepare our dataset. However, there are some parts that's different.

First of all, I need to reshape X data from [samples, timesteps] into [samples, timesteps, features]. In this case, the shape of training X is (31185, 120, 1), and validation X is (13366, 120, 1).

Second of all, I normalized both X and y by subtracting their mean and divided by their standard deviation. Then I used this data to training the model.

My third model is also a RNN model, however, in this case, I tried to predict the temperature of future 24 hours and 72 hours. In this case, my function to divide a sequence into multiple input/output patterns need to change a little bit, because in this situation, I tried to predict multiple timesteps in the future.

For example, if I want to use the past 3 time steps to predict the future 2 time step, and the sequence is [10, 20, 30, 40, 50, 60, 70, 80, 90]. Then I can prepare this dataset as shown in Fig.3.

```
[10 20 30] [40 50]
[30 40 50] [60 70]
[50 60 70] [80 90]
```

FIG. 3. Split result 2

When I tried to predict the future 24 hours, the shape of training X is (1299, 120, 1),

validation X is (557, 120, 1), training y is (1299, 24), validation y is (557, 24).

When I tried to predict the future 72 hours, the shape of training X is (432, 120, 1), validation X is (186, 120, 1), training y is (432, 72), validation y is (186, 72).

My last two models is include more features(humidity and pressure) to predict the temperature of the next hour. In this case, my function to divide a sequence into multiple input/output patterns need to change a little bit.

For example, if I want to use the past 3 time steps to predict the future 1 time step, and the sequence is shown in Fig.4.

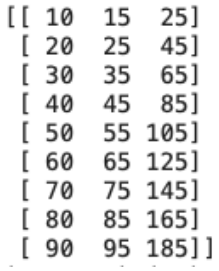

FIG. 4. The example sequence

If I want to use these three features to predict the first feature(first column), then I can prepare this dataset as shown in Fig.5.

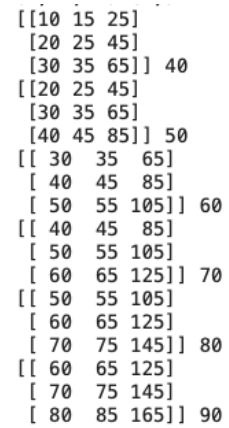

FIG. 5. Split result 3

When I tried to use 2 features, the shape of training X is (31185, 120, 2), validation X is (13366, 120, 2), training y is (31185, 1), validation y is (13366, 1).

IV. MODEL SELECTION AND MODEL EVALUATION

I used mean squared error to measure the performance of my models, and I used the last 13366 data in my dataset as my validation set.

My first model is a simple Neural Network with two dense layer, and I used historical temperature data to predict temperature of the next hour. See Fig.6

```
Model: "sequential_15"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_20 (Dense)             (None, 8)                 968
_________________________________________________________________
dense_21 (Dense)             (None, 1)                 9
=================================================================
Total params: 977
Trainable params: 977
Non-trainable params: 0
_________________________________________________________________
None
```

FIG. 6. Linear Model(Forecasting for a single timestep using a single feature)

The mean squared error of this model is 1.73.

My second model is a RNN model, and I used historical temperature data to predict the temperature of the next hour. See Fig.7

```
Model: "sequential_16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_4 (LSTM)                (None, 10)                480
_________________________________________________________________
dense_22 (Dense)             (None, 1)                 11
=================================================================
Total params: 491
Trainable params: 491
Non-trainable params: 0
_________________________________________________________________
None
```

FIG. 7. RNN Model(Forecasting for a single timestep using a single feature)

The mean squared error of this model is 1.03.

My third model is a RNN model, and I used historical temperature data to predict the temperature of the next 24 hours. See Fig.8

The mean squared error of this model is 34.81.

My fourth model is a RNN model, and I used historical temperature data to predict the temperature of the next 72 hours. See Fig.9

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 120, 50)           10400
_________________________________________________________________
lstm_1 (LSTM)                (None, 50)                20200
_________________________________________________________________
dense (Dense)                (None, 24)                1224
=================================================================
Total params: 31,824
Trainable params: 31,824
Non-trainable params: 0
_________________________________________________________________
None
```

FIG. 8. RNN Model(Forecasting for next 24 timesteps using a single feature)

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_4 (LSTM)                (None, 120, 50)           10400
_________________________________________________________________
lstm_5 (LSTM)                (None, 50)                20200
_________________________________________________________________
dense_2 (Dense)              (None, 72)                3672
=================================================================
Total params: 34,272
Trainable params: 34,272
Non-trainable params: 0
_________________________________________________________________
None
```

FIG. 9. RNN Model(Forecasting for next 72 timesteps using a single feature)

The mean squared error of this model is 39.39.

My fifth model is a RNN model, and the first time I used historical temperature and humidity data to predict the temperature of the next hour, the second time I used historical temperature and pressure data to predict the temperature of the next hour. See Fig.10

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 10)                520
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 11
=================================================================
Total params: 531
Trainable params: 531
Non-trainable params: 0
_________________________________________________________________
None
```

FIG. 10. RNN Model(Forecasting for next timestep using two features)

When I used historical temperature and humidity data to predict the temperature of the

next hour, the mean squared error is 1.85. When I used historical temperature and pressure data to predict the temperature of the next hour, the mean squared error is 2.45.

V. FEATURE IMPORTANCE AND INTERPRETATION

When using a single feature(temperature) to predict the temperature of next hour, the mean squared error of RNN model is smaller than the mean squared of the simple neural network model, this means RNN model is more suitable for this task.

When I tried to predict 24 hours using RNN model, the mean squared error exploded from 1.03 to 34.81. And when I tried to predict 72 hours using RNN model, the mean squared error become 39.39. One possible reason is, the model accuracy of predicting one hour is higher than predicting more hours. Another possible reason is, when predicting more hours, the number of training data decreases very much. When predicting the next hour, the number of training data is 31185, however, when predicting the next 24 hours, the number of training data decrease to 1299. And this lack of data maybe another reason for why the mean squared error of the model predicting the future 24 hours explodes. To get a reasonable accuracy when predicting more hours, it's better to use a more complex RNN model instead of this simple one.

When I tried to include two features to predict the temperature of the next hour, the mean squared error of the model using temperature and humidity is lower than the model using temperature and pressure. This shows the feature 'humidity' is more important than the feature 'pressure'.

However, I found the mean squared error of the models using two features is higher than the model using only one features. One possible reason is when including more irrelevant information, the accuracy of the model decreases. Another possible reason is I should find-tune my model, like finding the best learning rate.

VI. CONCLUSIONS

When using only one feature(temperature) to predict the temperature of the next hour, it's better to use RNN model instead of a simple neural network model. And when trying to use predict more hours, it's better to use a more complex model if you want a model with

reasonable accuracy. When it comes to feature importance, I test two features. And the feature 'humidity' is more important than the feature 'pressure'.

DATA AVAILABILITY

Data is available at https://www.kaggle.com/c/usc-dsci552-section-32415d-spring-2021-ps6/data

CODE AVAILABILITY

Code is available at https://github.com/usc-dsci552-32415D-spring2021/problem-set-06-rantao-usc