Group 1

M5 Forecasting - Uncertainty Report

Group Members:

Chen Tian	20795950	tchenaz@connect.ust.hk
Li Muxiao	20787989	mlidm@connect.ust.hk
Qiu Qingqing	20799190	qqiuae@connect.ust.hk

1 Introduction

1.1 Background

A sales forecast is an estimate of the quantity and value of products to be sold at a specified time in the future. Its essence is in the full consideration of the future on the basis of various factors combined with the actual sales performance of the enterprise, through a certain analysis method to put forward feasible sales goals. Sales forecasting based on historical data plays an important role in helping enterprises make better business strategies. It can be used to calculate the sales target, determine the total cost, so as to predict the profit and loss. In anticipation of the loss, measures can be taken to control and reduce costs, which ensure that the enterprise will not die quickly because of high costs and high inventory.

In this project, our goal is to predict retail giant Walmart's sales over the next 28 days. The difference between this project aim and our last one (accuracy), is that we need to do probabilistic forecasting for the corresponding median and four prediction intervals (50%, 67%, 95%, and 99%).

1.2 Dataset

The data set used in this project is from Kaggle's competition M5 Forecasting which covers Walmart from three US states(California, Texas, and Wisconsin) , involving 10 stores, 3 categories of products, 3049 products and a total of 42840 time series. In addition, it has explanatory variables such as price, day of the week, and special events.

Time series data from 2011-01-29 to 2016-06-19 are day granularity data with a time span of about five and a half years, which can be subdivided into 12 levels according to different object categories and levels. There are four data files we mainly used in our project, which are as follows.

train-evaluation.csv: Contains the historical daily unit sales data per product and store; [d1,d1941].

calendar.csv: Contains the dates on which the products are sold along with associated functions (such as day of the week, month, year) and 3 binary markers that indicate whether stores in each state are allowed to purchase SNAP food stamps on that date.

sell-prices.csv: Contains information about store, item ID, and average weekly prices of the products sold per store.

For these tables, the hierarchy relationship is shown as follows. The bottom layer is about the item number in detail. The top is given by the unit sales of all products, aggregated for all stores and states. Then, unit sales of all products, aggregated for each state. Unit sales of all products, aggregated for each store, etc.

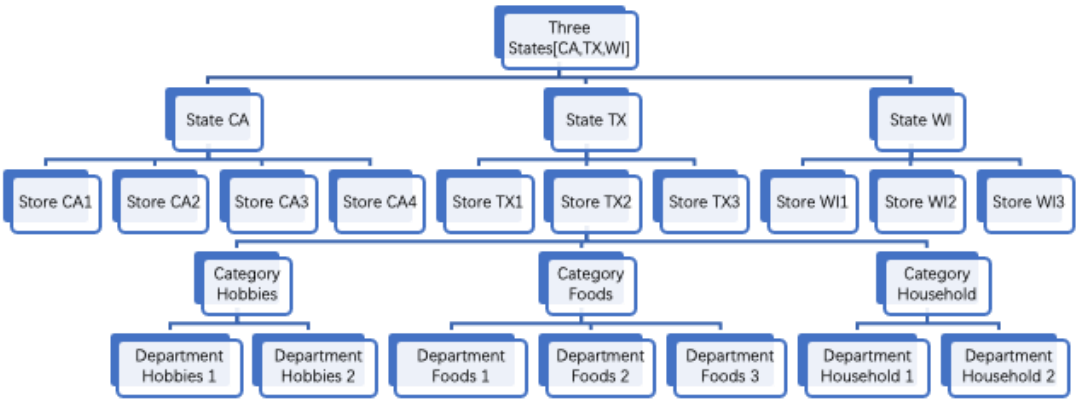

Figure 1: Dimension hierarchy relation in the dataset

2 Theory

2.1 Artificial Neural Network (ANN)

An Artificial Neural Network (ANN) is a computational model that is inspired by the way biological neural networks in the human brain process information. It is an extensive and interconnected network of adaptive simple neurons whose organization mimics the interactive responses of the biological nervous system to real-world objects.

ANN models are based on the input of multiple nonlinear models and the weighted interconnection of different models to obtain the final output model. The weighting process is completed in the hidden layer, which contains a nonlinear function. In the multivariate input layer, the input independent variables will be combined to the middle layer by weighting, which is called the hidden layer. The hidden layer is the so-called black box, which contains nonlinear kernel functions. Few people can read how the nonlinear kernel functions in the hidden layer combine the independent variables.

ANNs has good self-group learning ability, strong ability to classify untrained data patterns, can find the inherent nonlinear law effectively, and has a relatively high tolerance to outliers and noise data. However, this model also has some defects, such as requiring a long training time, being sensitive to missing values, and having the tendency of overfitting data. Most importantly, models are extremely complex and poorly interpretable.

2.2 Long Short Term Memory Network (LSTM)

LSTM is one of the Recurrent Neural Networks (RNN) that consists of repeated modules of neural networks. In each module, there are three control gates that are the forget gate, the input gate, and the output gate respectively, which allows the network to retain long-term dependencies between data at a given time from many timesteps before.

The input of the LSTM is a sequence of vectors $x = \{x_1, x_2, ..., x_t, ...\}$, where $x_t \in R^m$ represents an m-dimensional vector of readings for m variables at time t. The basic unit of an LSTM network is the memory block containing one or more memory cells and three adaptive,multiplicative gating units shared by all cells in the block. Each memory cell has a

recurrently self-connected linear unit which provides short-term memory storage for extended time periods by recirculating activation and error signals indefinitely. The input, forget, and output gate can be trained to learn what information to store in the memory, how long to store it, and when to read it out. And combining memory cells into blocks allows them to share the same gates, which is helpful for reducing the number of adaptive parameters.

3 Methodology

3.1 Data Analysis

As we have declared the hierarchy structure of the dataset, we went further into the sales changing over time at different aggregated levels.

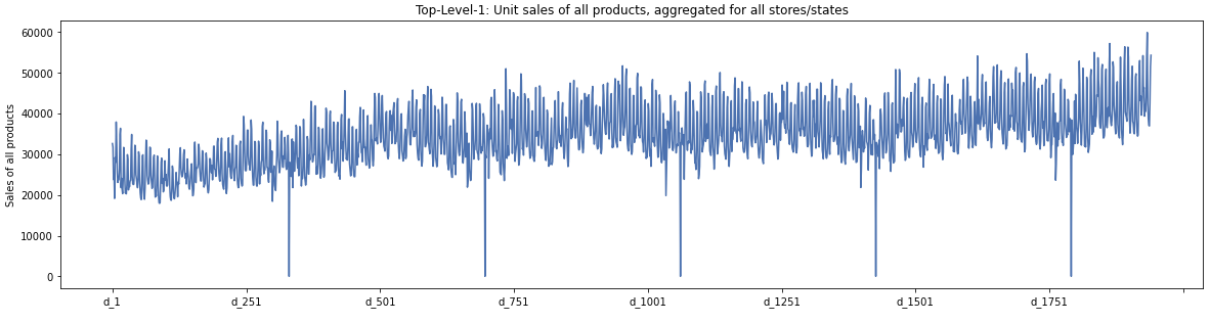

Figure 2: Top-level aggregated sales

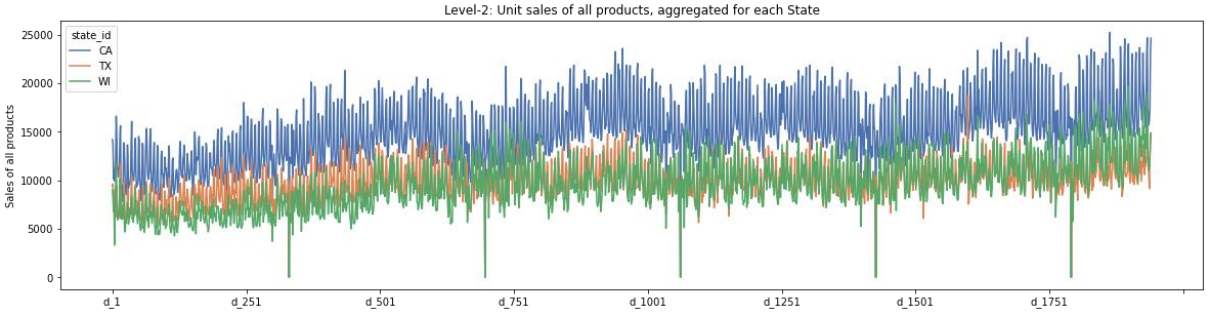

Figure 3: Second-level aggregated sales

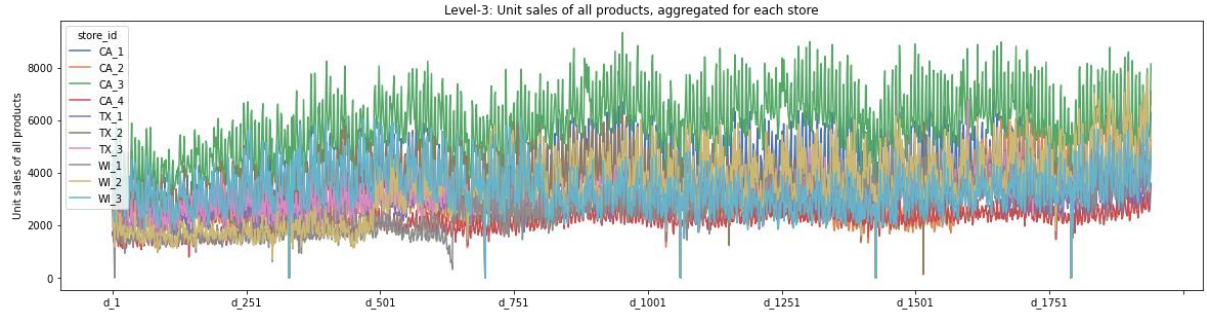

Figure 4: Third-level aggregated sales

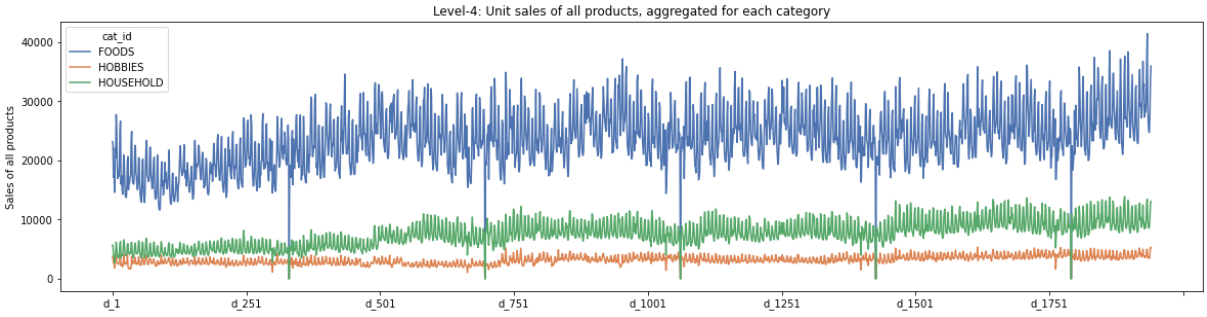

Figure 5: Fourth-level aggregated sales

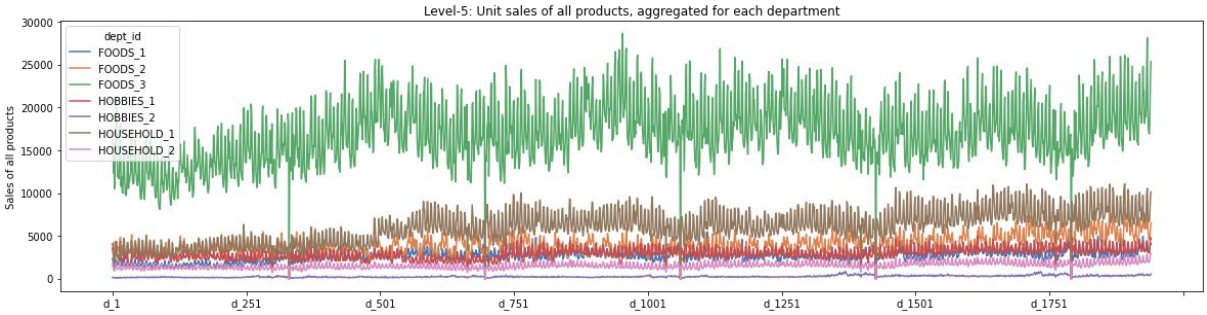

Figure 6: Fifth-level aggregated sales

Figure 2 shows the unit sales of all products, aggregated for all states and sales. Figure 3, 4, 5, 6 show the unit sales of all products, aggregated for each state/store/category/ department. From these figures, we can find that there exist remarkable periodic patterns. Within each level, there exist differences of groups. For example, in figure 4, it is obvious that the category ‘food’ sales is much higher than the other two. Meanwhile, the fluctuating range of food is larger. So, we are asked to generate forecasts for grouped time series.

By observing the hierarchy relationships, the time series in training data belong to the bottom-level: unit sales of product x, aggregated for each store. A simple method to generate forecasts for all levels is to focus only on the bottom level. All of its predictions are then summed up to create the forecasts of all levels up to the top. This is called the bottom-up approach. Thus, in this project, the necessary step is trying to solve the bottom-level aggregated time series sales prediction.

3.2 Model Evaluation

This competition uses a Weighted Scaled Pinball Loss (WSPL). We defined the pinball loss function and applied it as our model loss function. The math of the loss function enforces this by penalizing cases when the truth is on the more unusual side of your prediction more heavily than when it is on the expected side, that is, if quantiles less than 0.5, predicting lower is better than higher.

$$\mathbf{SPL(u)} = \frac{1}{h}\frac{\sum_{t=n+1}^{n+h}(Y_t - Q_t(u))u\mathbf{1}\{Q_t(u) \le Y_t\} + (Q_t(u) - Y_t)(1-u)\mathbf{1}\{Q_t(u) > Y_t\}}{\frac{1}{n-1}\sum_{t=2}^{n}|Y_t - Y_{t-1}|}$$

where $Y_t$is the actual future value of the examined time series at point t, $Q_t(u)$ the generated forecast for quantile u, h the forecasting horizon, n the length of the training sample (number

of historical observations), and 1 the indicator function (being 1 if Y is within the postulated interval and 0 otherwise).

Forecasting using this loss function corrects the odds of being above and below the forecast by purposefully introducing biases. Many firms reflect particular asymmetries in trading activity by purposefully introducing a bias in resource allocation, and this loss function is more suitable for tasks that deal with the risk asymmetry between overestimation and underestimation of demand.

3.3 Model Building

3.3.1 Data Preprocessing and Feature Engineering

(1) Data preprocess

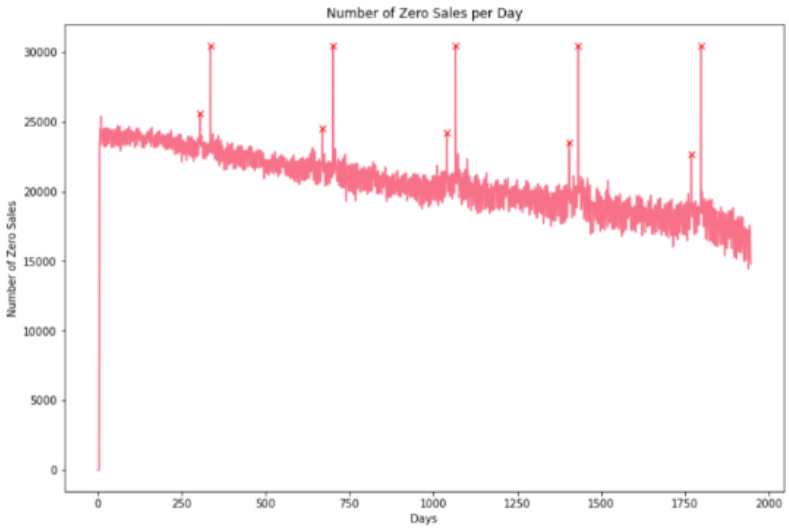

Figure 7: Number of zero sales per day

The figure shows that the period found in the above content may be caused by special events. So, the project further explores these events specifically. The result shows that the peak days are around 25 November in each year, which is about Thanksgiving Day and Christmas Day. To get rid of the influence caused by dates, we replace them using the mean value. After checking the outliers, the project moves to the necessary and important part, the aggregation part. According to the table containing 12 levels, here comes the aggregation result with the dimension (42840,1941). Additionally, the result should be tested for correctness by comparing the row names: whether the combination of the levels is right, and the rows contain the same input. The result turns out that the aggregation is well designed and in the right order.

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9
Total	32631.0	31749.0	23783.0	25412.0	19146.0	29211.0	28010.0	37932.0	32736.0
CA	14195.0	13805.0	10108.0	11047.0	9925.0	11322.0	12251.0	16610.0	14696.0
TX	9438.0	9630.0	6778.0	7381.0	5912.0	9006.0	6226.0	9440.0	9376.0
WI	8998.0	8314.0	6897.0	6984.0	3309.0	8883.0	9533.0	11882.0	8664.0
CA_1	4337.0	4155.0	2816.0	3051.0	2630.0	3276.0	3450.0	5437.0	4340.0
...	...	...	...	...	...	...	...	...	...
FOODS_3_823_WI_3_evaluation	0.0	0.0	2.0	2.0	0.0	3.0	1.0	4.0	1.0
FOODS_3_824_WI_3_evaluation	0.0	0.0	0.0	0.0	0.0	5.0	0.0	1.0	1.0
FOODS_3_825_WI_3_evaluation	0.0	6.0	0.0	2.0	2.0	4.0	1.0	8.0	5.0
FOODS_3_826_WI_3_evaluation	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
FOODS_3_827_WI_3_evaluation	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

42840 rows × 1941 columns

Figure 8: Aggregation result

(2) Feature engineering

After finding outliers and aggregation, the project tries to find some additional features. Of course, it is required to finish encoding the categorical variables event name and event type into numbers firstly. The additional features include whether the day is the event day, whether the day is before the event day and the unit sale from the file of sale training. Moreover, take prices per product per week into account. It is noteworthy that if simply considering the weekly prices, there are 3049 products * 10 stores, leading to 30490 additional columns. If we add price feature, the model efficiency would be too low, so we did not use it.

Lag feature is a classical method to transform time series prediction problems into supervised learning problems. A lag is a fixed amount of passing time. If we make a 1 lag change and train a model with this new feature, the model can predict a forward step and observe the current state of the sequence. Increasing the lag, for example, to 28, will allow the model to predict 28 steps in advance. We set lag range in [28, 35, 42, 49, 56, 63], and the 6 columns were created.

3.3.2 LSTM Model

In this project, the type of the LSTM can be regarded as multivariate and multiple input series.The model is trained using past sales values for each 30490 item and a feature which represents whether there exists an event at the following day. Since LSTM is time-consuming and requires quite large memory, to make the prediction available, the features about sell prices are not included in the LSTM model. Here, the main idea is to predict the sales value on the 15th day using the values within 14 days before. Thus, the training time series x and the training time series of y can be created. Meanwhile, instead of establishing a simple model, a better way is to apply embedding to reduce the data used by obtaining the parameters in the hidden layers. The settings of parameters are shown in the table, which are with pinball loss as the loss and with the Adam optimizer. It is noteworthy that there may be better settings without memory storage and time limitation. The epoch is about 30 with batch size 44.

Layer	First Layer	Second Layer	Third Layer	Fourth Layer
Units	40	400	400	Dense(42840)
Dropout	0.2	0.2	0.2	

For different quantiles, run the prediction model.

3.2.3 ANN Model

Scale Pinball losses are used to evaluate the neural network model. Compared with the traditional loss function, this loss function can be predicted without any prior data processing, and the range of output can be obtained naturally by using this output. In order to achieve a better prediction result evaluated by the Pinball losses, we constructed an extra dataset sales_add.csv based on sales_train_evaluation.csv and sample_submission.csv. In this new dataset, we generated the 12 other series with their respective aggregation level and created two new attributes "start" and "scale". "Start" is the starting selling date of the product. And "scale", the mean of demand absolute variation, refers to the normalization constant. It's actually the denominator of the SPL.

The neural network model we constructed consists of input, flatten, dense, dropout, and output layers. Embedding operation is performed on the input layer to transform features into vector form. For discrete features, one-hot mode can be used for transformation. However, for features with very high dimension but very sparse after one-hot transformation, it is generally done to transform them to embedding. Just like in our case.

The main body of the neural network model is a three-layer fully connected network with dropout. The number of units in the Dense Layer is 500 and the dropout parameter is 0.3. Because the full connection layer can only receive one-dimensional data, a layer of flatten needs to be added to convert the data of the input layer into one-dimensional data and then pass it into the full connection layer.

Models only learn to classify on the training set, but having poor performance on the test set is a common problem in neural network models. In order to reduce the impact of overfitting on model performance, we add dropout layers, which allow the neural network to temporarily remove some units from the network according to a certain probability during learning and training. Due to the randomness of discarding, the network model can be regarded as a new model with a different structure after each dropout, while the number of parameters to be trained remains the same. Therefore, the time consumption problem of training multiple independent and different neural networks is avoided. Dropout can achieve average performance across multiple models.

4 Model Performance

4.1 LSTM Model

The predicted total sales of the top level are shown in the figure 9. The specific changes of predicted values are not obvious visually.

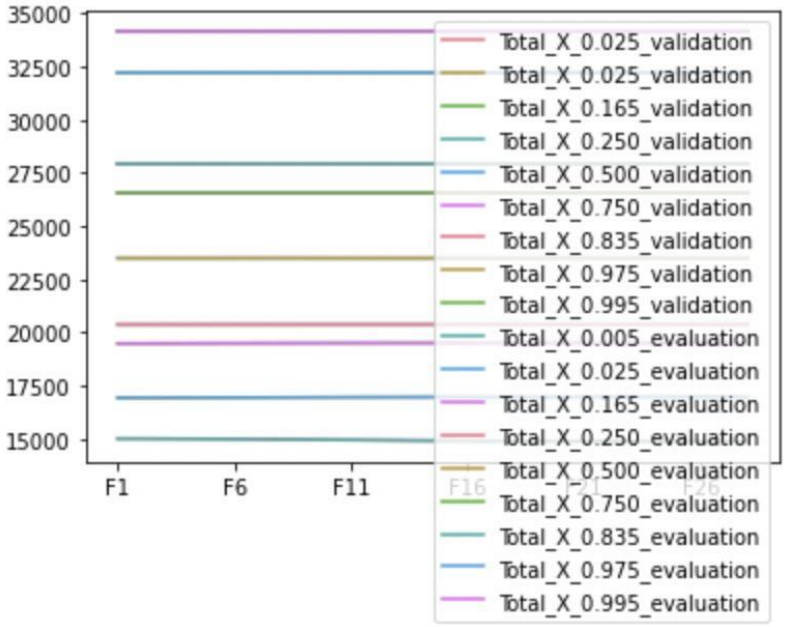

Figure 9: Predicted total sales of the top level

Before showing the results, it is admitted that the model needs improvement. Since only sales of past days and the feature representing whether the date is the bay before events are used for prediction, more features can be added to improve the model. For example, the lag features as well as sell prices. Moreover, multi-step LSTM can be used for prediction of more than one day (for example 28 days at once or 14 days, 7 days, 2 days etc.)

4.2 ANN Model

Since the performance of the LSTM model is not as good as expected, we decided to create some new attributes that can be helpful to improve the prediction effect. After constructing the new dataset which has been mentioned before, we firstly tried to fit a basic ANN model to validate the effect of the auxiliary dataset. By visualizing the prediction results (Figure), we can see the curves of true value and quantile prediction results of the three stalls obtained by model learning. The two curves with quantiles 0.25 and 0.75 can be used as upper and lower bounds of the predicted results and can contain about 80% of the predicted results. Selecting a smaller lower bound quantile and a larger upper bound quantile can obtain higher interval confidence.

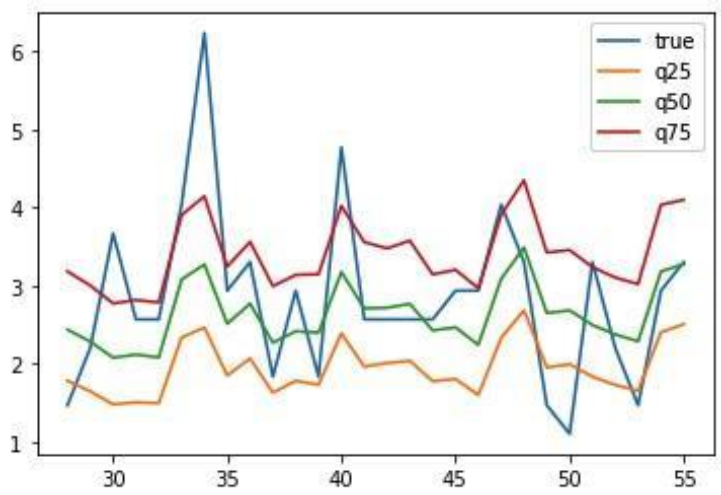

Figure 10 : Quantile Prediction Result

This ANN model obtains a much better score compared to the LSTM model. On the basis of this model, we also tried to add other structures such as GRU to enhance the model's performance. However, due to the large amount of data and the much more complex network structure, the training time and consuming resources have increased dramatically, which led to the failure of training a GRU model.

4.3 Comparison

	Public Score	Private Score
LSTM	0.54083	0.56559
NN	0.16162	0.17087

As is shown in the table, the ANN model obviously performs better than LSTM. We think the main reason is that for the LSTM model, we used the original data for training while an extra newly constructed dataset “sales_add” was used during the process of training the ANN model. In the competition, Weighted Scaled Pinball Loss is used for evaluation, so attributes ‘start’ and ‘scale’ in the new dataset helps to improve the score.

Reference

1. https://www.kaggle.com/code/bountyhunters/baseline-lstm-with-keras-0-7?scriptVersionId=37855876#Future-Improvements

Each member’s contribution to this project is as follows.

Member	Contribution
Chen Tian	LSTM model(code and report)
Li Muxiao	ANN model(code and report); video
Qiu Qingqing	ANN model(code and report); video