Predict Future Sales using LSTM

Zhao Jin, Jiaxu Shao, Kangjing Fan, Tong Xiang
Instructor: Keegan Hines

Background

The amount of products sold every month is one of the most important information for businessmen because it is highly related to the tendency of markets. Thus people will gain advantages in investment if they can predict future sales.

Problem statement

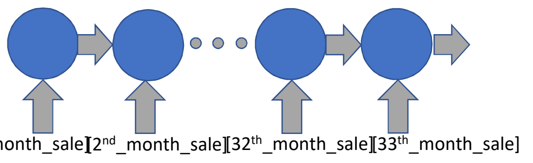
People are provided with daily historical sales data. The task is to forecast the total amount of products sold in every shop for the test set. The final goal is creating a robust model that can predict the total sales of products even in some situations where the list of shops and products changes every month.[1]

Dataset

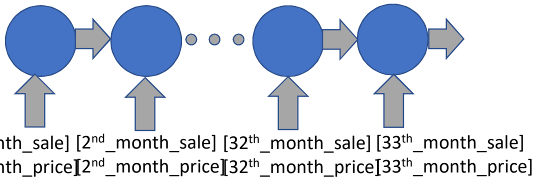
1. Training set contains 1048576 entries of data, each entry of data contains information about price of that product, the id of that product etc.
2. Test set contains 214201 entries of data, each entry only contains the corresponding shop id and item id.
3. Overall there are 22171 unique products; they belong to 84 categories and sold by 60 corresponding shops.
4. The training data is unbalanced; roughly half of the products belong to 2 categories.

Models description

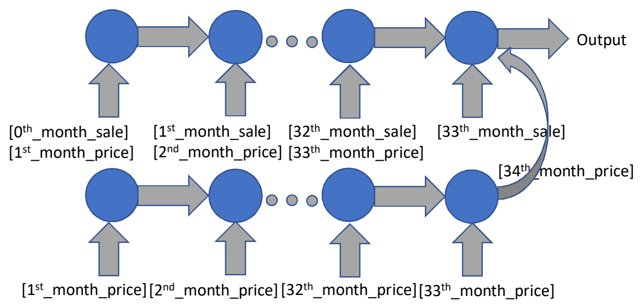
1. Basic_lstm



2. Lstm_price


[1st_month_sale] [2nd_month_sale] [32th_month_sale] [33th_month_sale]
[1st_month_price][2nd_month_price][32th_month_price][33th_month_price]

3. two_layer_lstm



Evaluation&Analysis

For both tables, we evaluate root mean square error(RMSE) for all the models mentioned. The lower the score is, the better.

Table1: Our LSTM models with different design

Model	Description	Performance
Basic LSTM	Use monthly sales as time series data	1.022
LSTM1	Add monthly prices together with monthly sales as series data	1.044
LSTM2	Use data division	1.018
LSTM(GRU)	Use GRU, a variation of LSTM, on original data	1.026

From the table we notice that adding monthly price feature to our model will hurt the performance. The major reason is that the time series data is too short, with only 33 months of data, And most of the goods are missing lots of monthly sale data. Even use the 33 monthly prices to predict 34th monthly sale in a lstm model cannot get a convincing result.

Our model can do even better with data division. Firstly, this is because that this operation brings data augmentation to our model, thus make it more predictive. Even though the length of each sample is much smaller, it doesn't matter because the result suggests that the patterns of the data in previous years may not have much relationship to the pattern we are looking for.

Secondly, by doing this, we are able to focus on the most recent data and prevent the model to learn something from previous years which may not apply to the current year.

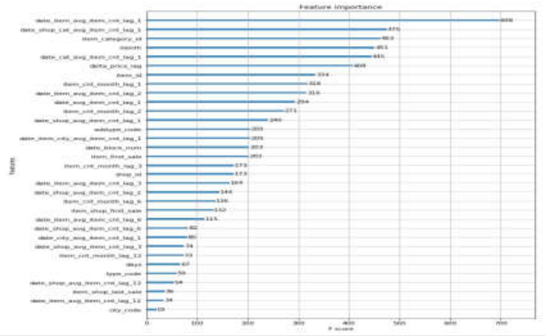
Table2: Our LSTM models compared with other models

Model	Description	Performance
LSTM	LSTM2 in the last table	1.018
XGboost	An implementation of gradient boosted decision trees	0.912
MLP	Don't treat input as sequenced data	1.099
CNN	One convolutional hidden layer followed by a max pooling layer	1.123
CNN-LSTM	A hybrid CNN-LSTM model	1.087

First of all, it is still necessary to say that because the time series is too short and data is missing, the data does not reflect the integrity of the time series and it is difficult to find a fixed pattern. But tree model does not rely much on the integrity of the time series. Secondly, tree model is not only a vertical comparison on a single product, but can learn correlation between peer item where lstm is weak in that aspect.

Exploration about features

Based on the analysis above, we went deeper into the importance of different features. This graph comes from Denis Larionov's notebook "Feature Engineering ,xgboost"[2] posted on Kaggle.



[1]: https://www.kaggle.com/c/competitive-data-science-predict-future-sales/overview
[2]: https://www.kaggle.com/dlarionov/feature-engineering-xgboost