CSE492 SCIENTIFIC PROGRAMMING FINAL PROJECT REPORT

Prepared for: Asst.Prof.Dr. Hüseyin Gökhan Akçay
Prepared by: Mustafa Arif Şişman
23 June 2020

TABLE OF CONTENTS

1. PROBLEM

The competition I choose is ‘Predict Future Sales’ in Kaggle competitions. The problem description is as follows.

‘In this competition you will work with a challenging time-series dataset consisting of daily sales data, kindly provided by one of the largest Russian software firms - 1C Company.
We are asking you to predict total sales for every product and store in the next month. By solving this competition you will be able to apply and enhance your data science skills.

You 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. Note that the list of shops and products slightly changes every month. Creating a robust model that can handle such situations is part of the challenge.’

1.1. File descriptions

- sales_train.csv - the training set. Daily historical data from January 2013 to October 2015.
- test.csv - the test set. You need to forecast the sales for these shops and products for November 2015.
- sample_submission.csv - a sample submission file in the correct format.
- items.csv - supplemental information about the items/products.
- item_categories.csv - supplemental information about the items categories.
- shops.csv- supplemental information about the shops.

1.2. Data Fields

- ID - an Id that represents a (Shop, Item) tuple within the test set
- shop_id - unique identifier of a shop
- item_id - unique identifier of a product
- item_category_id - unique identifier of item category
- item_cnt_day - number of products sold. You are predicting a monthly amount of this measure
- item_price - current price of an item
- date - date in format dd/mm/yyyy
- date_block_num - a consecutive month number, used for convenience. January 2013 is 0, February 2013 is 1,..., October 2015 is 33
- item_name - name of item
- shop_name - name of shop
- item_category_name - name of item category

I chose this competition because I wanted to work on time series and MATLAB. It is also interesting to create a model for a company that sells goods online and predict future sales for this company.

2. RELATED WORKS

2.1. Long Short-Term Memory

Authors: Sepp Hochreiter, Jürgen Schmidhuber
This article contains the introduction and innovations of Long Short Term Memory. The current backpropagation causes decaying error back flow and mostly insufficient for extended time intervals, LSTM provides a more efficient gradient-based solution for time series. LSTM will be explained in detail later in the report.

2.2. Learning to forget: continual prediction with LSTM

Authors: Felix A. Gers, Jürgen Schmidhuber, Fred Cummins
Long Short-Term Memory can solve many tasks not solvable by previous learning algorithms for recurrent neural networks (RNNs). They identify a weakness of LSTM networks processing continual input streams without explicitly marked sequence ends. Without resets, the internal state values may grow indefinitely and eventually cause the network to break down. They added forget gates to LSTM networks which provides releasing internal resources at the appropriate time. They compared the standard LSTM and LSTM with forget improvement.

2.3. Classification and Regression Tree Methods

Authors: Wei-Yin Loh
This article compares different classification and regression tree methods by their algorithms, performance, features, and properties. C4.5, CART, CRUISE, GUIDE and QUEST methods are compared in this article's context and different methods examined.

2.4. An Introduction to Classification and Regression Tree (CART) Analysis

Authors: Roger J. Lewis
This article introduces and analysis Classification and Regression Tree (CART) especially. In the context of this article, CART is used to create reliable clinical decision rules. Traditional statistical methods are not sufficient for these kinds of problems, first CART introduced then CART is used for classification for triage data like HIV.

2.5. Comparing performances of logistic regression, classification and regression tree, and neural networks for predicting coronary artery disease

Authors: Imran Kurt, Mevlut Ture, A. Turhan Kurum
In this study, performances of classification techniques were compared in order to predict the presence of coronary artery disease(CAD). They compared performances of logistic regression (LR), classification and regression tree (CART), multi-layer perceptron (MLP), radial basis function(RBF), and self-organizing feature maps (SOFM).

3. METHODOLOGY

3.1. Decision Tree Learning

Decision Tree Learning is a type of predictive modeling approach used in statistics, machine learning, and data mining. It is a supervised learning algorithm. It is a tree structure where an internal node represents features, the branch represents a decision rule, and each leaf node represents an outcome. The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the feature value. It partitions the tree in a recursive manner called recursive partitioning. This tree structure helps decision making. It's visualization like a flowchart diagram that easily mimics human-level thinking. That is why decision trees are easy to understand and interpret. Decision trees are among the most popular machine learning algorithms given their intelligibility and simplicity.

3.1.1. Classification Trees

Classification trees are tree models that input and output variables are discrete values. In this structure, leaves represent class labels and branches represent conjunctions of features that lead to those class labels.

3.1.2. Regression Trees

Regression trees are tree models that input and output variables are continuous (typically real numbers) values. Leaves represent predicted outcomes according to input variables.

3.2. Time Series

Time series refers to the frequency of data points in statistics, signal processing, econometrics, and mathematical finance, and is typically measured at regular time intervals, consecutive time zones. Examples of time series are weather, sales, stock exchange, forex, etc.

3.3. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is an artificial Recurrent Neural Network (RNN) architecture used in deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

LSTM networks are well-suited to classifying, processing, and making predictions based on time series data since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs, hidden Markov models, and other sequence learning methods in numerous applications.

3.4. Recurrent Neural Network (RNN)

A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable-length sequences of inputs.

4. IMPLEMENTATION

In the implementation of this problem, 2 different approaches were used for comparing them. These are decision trees (both classification and regression) and long short-term memory. Decision trees are easier to implement than LSTM architecture.

4.1.1. Classification Tree Training

In the classification decision tree learning part, first the data grouped by months, shop_ids, and item_ids, then we sum the monthly sales according to these groups. The groups are used for trains and the sale counts are used for labels. fitctree method used to train the classification decision tree.

34th month is added to the test data as month data. Classification decision tree's prediction gives the number of sales. Because, while the train is being used, sales data were used as labels.

4.1.2. Regression Tree Training

The steps applied when training the Classification tree were used exactly when training the regression tree. Unlike classification tree training, fitrtree method was used instead of fitctree method. While prediction with regression tree, it predicts as continuous data.

4.1.3. Long Short-Term Memory Network Training

In the LSTM part, the data are grouped by shop_ids and item_ids because we need another type of grouping with months data. We need to create time-series lags for monthly sale items. To achieve this, a function called count_sales was written. This function takes the month vector and sales vector for each vector in the group and returns monthly sales as a 1x34 vector. The columns of this vector represent the months and the internal data represents monthly sales.

Before the training, data was standardized using the ‘Studentized Residuals’ method. For each item, their average value subtracted and it multiplied with their standard deviation. This ensures that the data is between 1 and -1 and the train operation is more accurate.

Then the standardized train data were separated into 2 different parts as x_train and y_train. x_train contains the data first to end minus 1. y_train contains the data second to end. For each step LSTM tries to predict the weight of x_train and y_train, then forgets their weights. This process repeats 200 epochs for all data.

Adam optimizer was used in the training and the model was trained on the GPU. In every 50 epoch, the initial learning rate drops half. Mini batch size is 20 for training. The model inputs monthly sold items data and predict for next month’s sales.

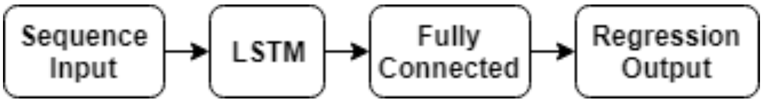


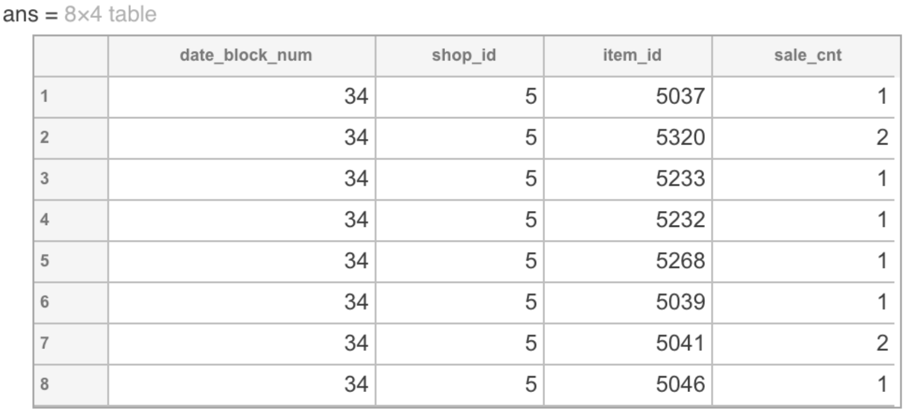
Figure 1 LSTM Architecture

This diagram illustrates the architecture of a simple LSTM network for regression. The network starts with a sequence input layer followed by an LSTM layer. The network ends with a fully connected layer and a regression output layer.

4 layers were used in LSTM Architecture. Sequence input layer takes the inputs, LSTM layer learns long-term dependencies between time steps in time series, fully connected layer multiplies the input by a weight matrix, and adds a bias vector and regression layer creates a regression output.

5. EXPERIMENTAL EVALUATION

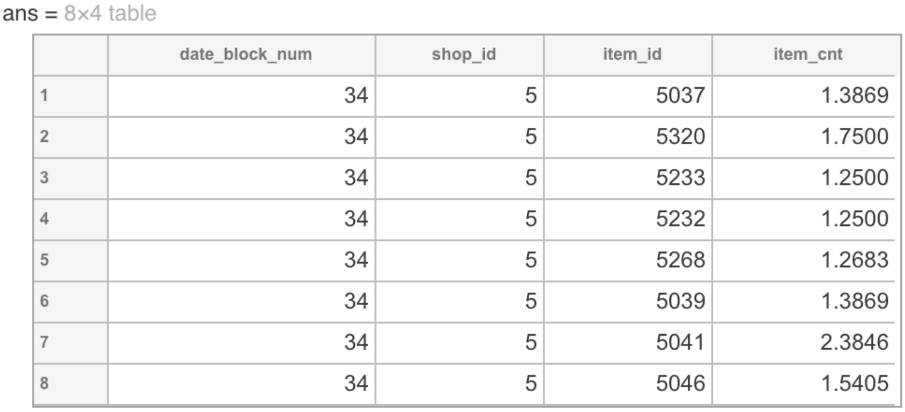
5.1.1. Decision Tree Predict Results



ans = 8×4 table

	date_block_num	shop_id	item_id	sale_cnt
1	34	5	5037	1
2	34	5	5320	2
3	34	5	5233	1
4	34	5	5232	1
5	34	5	5268	1
6	34	5	5039	1
7	34	5	5041	2
8	34	5	5046	1

Figure 2 Classification Tree Prediction Result



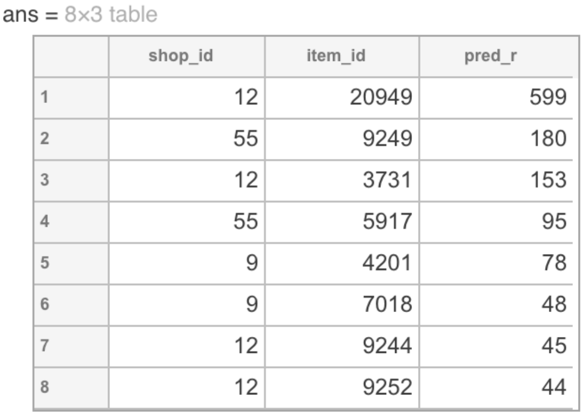
ans = 8×4 table

	date_block_num	shop_id	item_id	item_cnt
1	34	5	5037	1.3869
2	34	5	5320	1.7500
3	34	5	5233	1.2500
4	34	5	5232	1.2500
5	34	5	5268	1.2683
6	34	5	5039	1.3869
7	34	5	5041	2.3846
8	34	5	5046	1.5405

Figure 3 Regression Tree Prediction Result

In decision tree prediction, the labels predicted with 0.855701 accuracy. The result table contains 214.200 predictions (for each test data). The above results are given as an example.

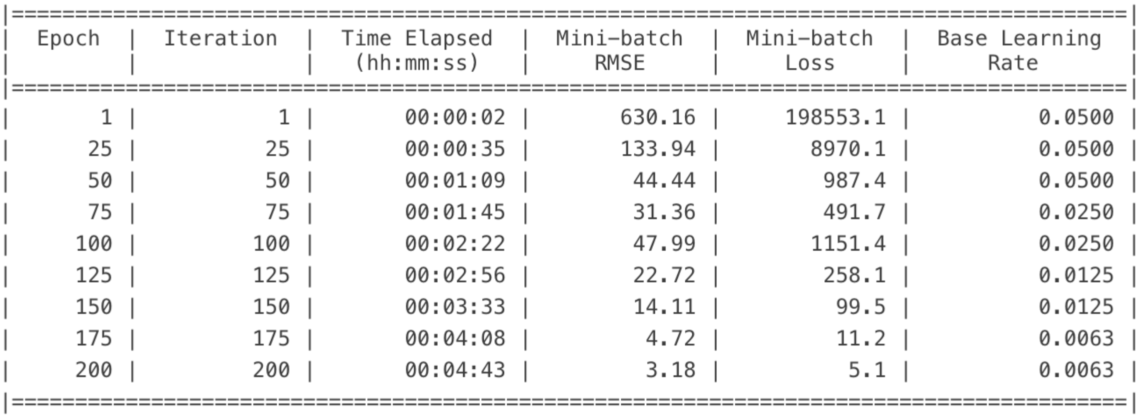
5.1.2. Long Short-Term Memory Results



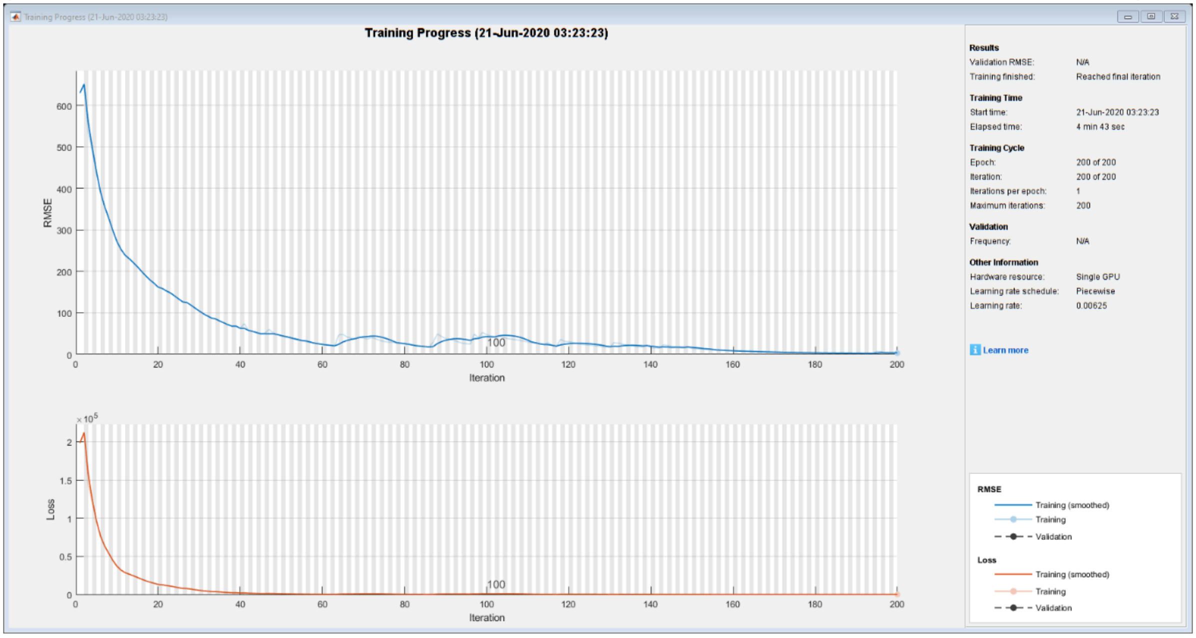
ans = 8×3 table

	shop_id	item_id	pred_r
1	12	20949	599
2	55	9249	180
3	12	3731	153
4	55	5917	95
5	9	4201	78
6	9	7018	48
7	12	9244	45
8	12	9252	44

Figure 4 LSTM Prediction Result



```
|========================================================================================|
|  Epoch  |  Iteration  |  Time Elapsed  |  Mini-batch  |  Mini-batch  |  Base Learning  |
|         |             |   (hh:mm:ss)   |     RMSE     |     Loss     |      Rate       |
|========================================================================================|
|       1 |           1 |       00:00:02 |       630.16 |     198553.1 |          0.0500 |
|      25 |          25 |       00:00:35 |       133.94 |       8970.1 |          0.0500 |
|      50 |          50 |       00:01:09 |        44.44 |        987.4 |          0.0500 |
|      75 |          75 |       00:01:45 |        31.36 |        491.7 |          0.0250 |
|     100 |         100 |       00:02:22 |        47.99 |       1151.4 |          0.0250 |
|     125 |         125 |       00:02:56 |        22.72 |        258.1 |          0.0125 |
|     150 |         150 |       00:03:33 |        14.11 |         99.5 |          0.0125 |
|     175 |         175 |       00:04:08 |         4.72 |         11.2 |          0.0063 |
|     200 |         200 |       00:04:43 |         3.18 |          5.1 |          0.0063 |
|========================================================================================|
```



Results
Validation RMSE: N/A
Training finished: Reached final iteration

Training Time
Start time: 21-Jun-2020 03:23:23
Elapsed time: 4 min 43 sec

Training Cycle
Epoch: 200 of 200
Iteration: 200 of 200
Iterations per epoch: 1
Maximum iterations: 200

Validation
Frequency: N/A

Other Information
Hardware resource: Single GPU
Learning rate schedule: Piecewise
Learning rate: 0.00625

Figure 5 LSTM Training Progress

LSTM can predict next month's sales for every item shop tuple in the train data with their sale counts. It is observed that, while some predictions match with the decision tree, some predictions do not match.

It is observed that the normalization of the train data and using mini-batches accelerate the training and contribute to the fact that the model is more accurate. With different parameters, LSTM has been trained many times.

6. CONCLUSION

Three models were trained to predict sales data next month and tests were carried out with the test data given in the problem.

MATLAB is extremely useful for scientific programming, machine learning and deep learning purposes. In addition, a very important acceleration was observed when the training process done with GPU (CUDA).

7. REFERENCES

7.1.1. Related Works

- https://www.bioinf.jku.at/publications/older/2604.pdf
- https://pdfs.semanticscholar.org/e10f/98b86797ebf6c8caea6f54cacbc5a50e8b34.pdf
- http://pages.stat.wisc.edu/~loh/treeprogs/guide/eqr.pdf
- https://pdfs.semanticscholar.org/6d4a/347b99d056b7b1f28218728f1b73e64cbbac.pdf
- https://pdfs.semanticscholar.org/72d2/e19a374a0353689dc6af54e904b9300f5fed.pdf

7.1.2. MATLAB

- https://www.mathworks.com/help/deeplearning/ug/long-short-term-memory-networks.html
- https://www.mathworks.com/help/deeplearning/ug/time-series-forecasting-using-deep-learning.html
- https://www.mathworks.com/help/deeplearning/ref/trainnetwork.html
- https://www.mathworks.com/help/deeplearning/ref/trainingoptions.html
- https://www.mathworks.com/help/stats/fitctree.html
- https://www.mathworks.com/help/stats/fitrtree.html
- https://www.mathworks.com/help/matlab/ref/findgroups.html
- https://www.mathworks.com/help/matlab/ref/splitapply.html
- https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.fullyconnectedlayer.html
- https://www.mathworks.com/help/deeplearning/ref/regressionlayer.html
- https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.lstmlayer.html
- https://www.mathworks.com/help/stats/compactclassificationtree.predict.html
- https://www.mathworks.com/help/deeplearning/ref/predictandupdatestate.html

7.1.3. Other

- https://www.kaggle.com/c/competitive-data-science-predict-future-sales
- https://www.wikiwand.com/en/Studentized_residual
- https://www.wikiwand.com/tr/Long_short-term_memory
- https://www.wikiwand.com/en/Recurrent_neural_network
- https://www.wikiwand.com/en/Decision_tree_learning