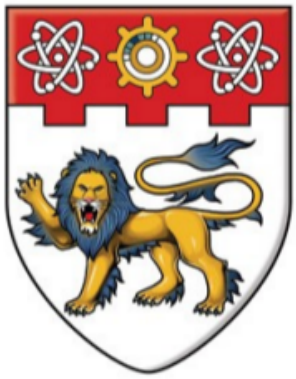

NANYANG TECHNOLOGICAL UNIVERSITY

CZ4041 – MACHINE LEARNING

PETER

SCHOOL OF COMPUTER ENGINEERING

2016/17

Contents

List of Tables

List of Figures

1. Introduction

Lorem ipsum dolor sit amet.

1.1 Team Members

Lorem ipsum dolor sit amet.

1.2 Problem Statement

Lorem ipsum dolor sit amet.

1.2.1 Challenge

Lorem ipsum dolor sit amet.

2. Preprocessing

The problem contains three files, *train.csv*, *test.csv*, and *store.csv*. The file *train.csv* contains historical data including sales for 1115 stores everyday from 1 Jan 2013 to 31 July 2015. While *store.csv* contains supplemental information about each store. Then, *test.csv* contains historical data excluding sales and number of customers everyday from 1 Aug 2015 to 17 Sept 2015.

Each row in *train.csv* contains store ID, day of week, date, number of customers, sales, whether the store is open, whether the store is doing a promo, state holiday, and whether that day is a school holiday. Columns in *test.csv* is almost identical to those of *train.csv*, except that sales and number of customers are unknown in *test.csv*. Each row in *test.csv* contains a submission ID for the purpose of evaluation on prediction result. On the other hand, each row in *store.csv* contains the details of a store, such as store type, assortment type, whether the store has competition, since when the competition exists, competition distance, whether the store is doing *promo2*, and *promo2* period.

Initially, our preprocessing method merges *train.csv* and *store.csv* by store ID. The columns *DayOfWeek* and *StateHoliday* in *train.csv* are transformed into one-hot vector respectively. The columns *StoreType* and *Assortment* in *store.csv* are transformed into one-hot vector respectively as well. The columns *CompetitionOpenSinceMonth* and *CompetitionOpenSinceYear* are substituted into a single column *HasCompetition* which depends on its respective *Date* column. The same thing also applies to the columns *Promo2SinceWeek*, *Promo2SinceYear*, and *PromoInterval*, they are substituted into a single column *IsDoingPromo2* which depends in its respective *Date* column. *CompetitionDistance* is set to the maximum value in the training set if a store does not have a competition in any given date. All store data are retained even when a store is closed on a particular date.

After *train.csv* and *store.csv* has been merged, we proceed to merge *test.csv* and

store.csv by store ID as well. The preprocessing method in this step is identical in the previous paragraph. The difference is that sales and the number of customers are unknown in the test dataset.

3. Experiments

Lorem ipsum dolor sit amet.

3.1 LSTM

LSTM (abbreviation for Long-short Term Memory) is a variant of recurrent neural network. LSTM seeks to solve vanishing gradient and exploding gradient problem in vanilla recurrent neural network. These problems happen when a training sequence is too long. LSTM introduces gating concept in recurrent neural network. An LSTM contains three gates, an input gate, an output gate, and a forget gate. A forget gate determines when to reset the content of the cell memory. An input and output gate control the flows of input and output of the LSTM respectively.

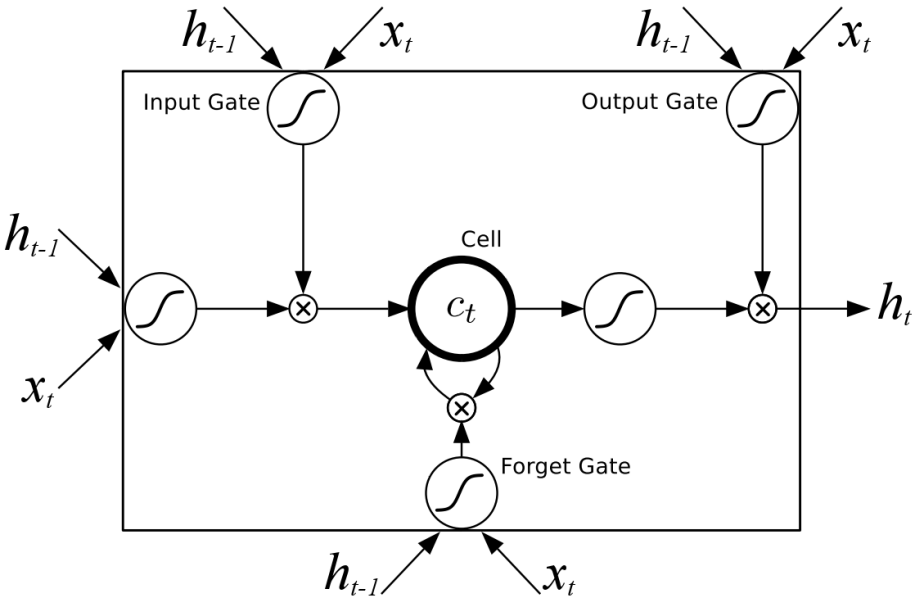

Figure 3.1: An LSTM unit

We use LSTM unit provided by Caffe[1] framework. It requires CUDA 8 and

[1]Caffe is developed by Berkeley Vision and Learning Center (BVLC)

cuDNN v5.1 from Nvidia, which enable Caffe framework to utilize Nvidia GPU to train a neural network. We used Nvidia GTX 850m to train our LSTM network. GTX 850m is pretty decent to train a medium-sized network.

3.2 Neural Network

Lorem ipsum dolor sit amet.

4. Results

Lorem ipsum dolor sit amet.

5. Conclusion

Lorem ipsum dolor sit amet.