ISBN 978-981-14-1684-2
Proceedings of 2019 the 9th International Workshop on Computer Science and Engineering
(WCSE 2019 SUMMER)
Hong Kong, 15-17 June, 2019, pp. 280-287
doi: 10.18178/wcse.2019.06.042

Sentiment Analysis of Movie Review Based on LSTM

Yuyao Cheng [1], Qiyan Kang [2] , Changying Wang [3],Yongfen Liu [2] and Li Cheng [2 +]

[1] Faculty of Education, The University of Hong Kong, Hongkong, China

[2] Jinshan College, Fujian Agriculture and Forestry University, Fuzhou, China

[3] College of Computer and Information Sciences, Fujian Agriculture and Forestry University, Fuzhou, China

Abstract. This paper introduces a sentiment analysis model based on LSTM (Long Short-Term Memory), which can be used for emotional polarity classification of movie review data. The model consists of five levels, namely input layer, embedding layer, LSTM layer, softmax layer, and output layer. IMDb (Internet Movie Database) dataset is used for training, and SGD (stochastic gradient descent), Adadelta, and RMSprop for optimization. The experiment result shows that Adadelta optimization algorithm can get the best training model and the lowest prediction error rate.

Keywords: LSTM, Deep Learning, Sentiment Analysis, Polarity Classification.

1. Introduction

Movie reviews are the expression of people's feelings after watching the movie. The movie review reflects the emotional to some extent. In psychology, the mood is classified into four kinds of, such as happiness, anger, fear and sadness. Among them, happiness is positive, and others are negative emotions. A movie can get its emotional tag appropriately by analysing movie review. Analysis methods include sentiment dictionary [1], machine learning [2], data-driven vs. dictionary [3], weakly annotation [4], and deep learning [5].

The dictionary method uses not only emotional dictionary but also syntactic structure of the commentary data to judge emotional polarity. This method, to a great extent, depends on artificial design and prior knowledge, and it is difficult to popularize. Classical classification models, such as support vector machine, naive Bayes, maximum entropy model and so on, are based on machine learning, and often used in emotion classification. Their performance depends on quality of tagged data, while obtaining high quality annotated data requires a lot of labour costs. Emotional classification method based on dictionary and machine learning combines dictionary information with syntactic features, POS features, etc. Weakly annotation method can be used to mine useful information (such as comment score, expression icon) from user's data. But natural language is unstructured, difficult to extract features and store. The structure of deep learning is flexible. It uses word embedding technology on different length text, and automatically mines complex semantics from non-characteristic data [6]. So deep learning is excellent in emotional classification [7], emotion extraction [8], and emotional vocabulary learning [9] of sentence or document.

In this paper, LSTM network is used to train a model for emotional classification of movie reviews. Emotional tendencies or polarity can be marked by the model. Then movie reviews are classified into two categories: positive and negative.

2. LSTM Model

+ Corresponding author. Tel.: + 86 591 83741723; fax: +86 591 83725639.
E-mail address: li.cheng@fafu.edu.cn

LSTM [10] is a variant of recurrent neural network. The critical problem of (simple) recurrent neural network is that gradient explosion or gradient dispersion is easy to occur with the increase of time interval (long term dependencies). In order to effectively solve this problem, the threshold is usually introduced to control the accumulation rate of information, and the cumulative information before forgetting can be selected.

LSTM consists of three gates, that is, input gate, forgetting gate and output gate. The mathematical analysis and network structure of input, threshold and output are shown in figure 1.

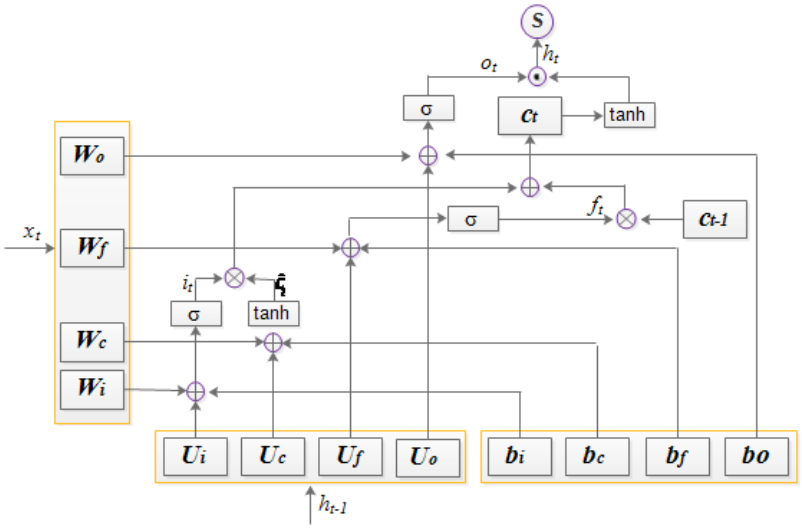

Fig. 1: LSTM network.

The symbol "$\otimes$" represents the multiplication of the corresponding elements in the corresponding vector.

2.1. Input Gate

The main purpose of the gate is to determine how many components in the input x_t are retained in c_t.

$$\begin{cases} i_t = \sigma(W_i \cdot x_t + U_i \cdot h_{t-1} + b_i) \\ \tilde{c}_t = \tanh(W_c \cdot x_t + U_c \cdot h_{t-1} + b_c) \end{cases} \tag{1}$$

i_t represents the input gate at the current time, σ represents the activation function sigmoid, x_t represents the input at the current time, W_i represents the weight matrix of the input at the current time in the input gate, h_{t-1} represents the output at the last time, U_i represents the weight matrix of the output at the last time in the input gate, and b_i represents the bias of the input gate. $\tilde{c}_t$ represents the new memory at the current time, W_c represents the weight matrix of the input at the current time in the new memory, and U_c represents the weight matrix of the output at the last time in the new memory, and b_c represents the bias of the new memory.

The $\tilde{c}_t$ times of the input i_t is retained after passing the input gate: $c_t^i = i_t * \tilde{c}_t$.

2.2. Forgetting Gate

The purpose of the gate is to determine how many components of c_{t-1} in the input are retained in c_t at time t.

$$f_t = \sigma(W_f \cdot x_t + U_f \cdot h_{t-1} + b_f) \tag{2}$$

f represents "forget", W_f represents the weight matrix of the input at the current time in the forgetting gate, U_f represents the weight matrix of the output at the last time in the forgetting gate, b_f represents the bias of the forgetting gate. The threshold of the forgetting gate can be calculated by formula (2). Just like the threshold of the input gate $\tilde{c}_t$, after passing the forgetting gate, c_t^f is retained in c_t: $c_t^f = f_t * c_{t-1}$.

2.3. Output Gate

The purpose of the gate is to use control unit c_t to determine how many components in output o_t are outputted to hidden layer s_t. First, after passing the input gate and the forgetting gate, the state is C.

$$c_t = c_t^i + c_t^f \tag{3}$$

c_t^i is the composition retained in c_t after passing the input gate, and c_t^f is the composition retained in c_t after passing the forgetting gate. Second, in order to determine how many components of c_t are retained in s_t, the formula for output is given as follows:

$$o_t = \sigma(W_o \cdot x_t + U_o \cdot h_{t-1} + b_o) \tag{4}$$

o_t represents the output gate of the current time, W_o represents the weight matrix of input at the current time in the output gate, U_o represents the weight matrix of output at the last time in the output gate, and b_o represents the bias of the output gate. Finally, passing the output gate, the components retained in hidden layer are:

$$h_t = o_t \odot \tanh(c_t) \tag{5}$$

3. Optimization Strategy

Gradient descent is a first-order iterative optimization algorithm for finding the minimum of a function. It is also the most commonly used method of optimizing neural network. The purpose of gradient descent is to minimize the objective function $J(\theta)$, of which θ is the model parameter, $\theta \in R$. The way is to update the corresponding parameter values for each variable according to the objective function in the opposite direction of the gradient in each iteration. The learning rate η determines the number of iterations that the function reaches (local) minimum.

3.1. Stochastic gradient descent

For m samples with a single feature $(x^{(i)}, y^{(i)})(i = 1,2,...m)$, x_i is the feature of i-th sample, y_i is the output of i-th samples. Each update of stochastic gradient descent is to solve penalty function for sample (x, y) of dataset, and then the partial derivative is obtained：

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^{(i)}; y^{(i)}) \tag{6}$$

The result of the gradient of the objective function for parameter θ_i is $g_{t,i}$ when the number of iterations is t, then the update rule of the stochastic gradient descent is:

$$\theta_{t+1,i} = \theta_{t,i} - \eta \cdot g_{t,i} \tag{7}$$

The stochastic gradient descent algorithm [11] randomly selects only one sample to update the parameters each time. Its advantage is fast and online update. But it may not be carried out in the right direction every time, so it perhaps to make the optimal fluctuation (disturbance), which increases the number of iterations (learning times), that is, the speed of convergence is slow.

3.2. Adadelta

Adadelta [12] simulates second order Newton method by first order method approximately. It adaptively adjusts learning rate and does not rely on global learning rate. Adadelta uses the window w, and the mean for sequence (excluding the current) of history windows of the parameter gradient. The final mean is the weighted average of the time attenuation of the historical window sequence mean and the current gradient. Adadelta update rule is:

$$\theta_{t+1} = \theta_t + \Delta\theta_t \tag{8}$$

Among them,

$$\Delta\theta_t = -\frac{\mathrm{RMS}[\Delta\theta]_{t-1}}{\mathrm{RMS}[g]_t} g_t \tag{9}$$

RMS represents the root mean square error of the gradient.

3.3. RMSprop

The full name of the RMSProp (Poljak, 1964) is called Root Mean Square Prop. By using the differential square weighted average to modify the swing amplitude, the problem of excessive swing amplitude in the update of the loss function is solved, and the convergence speed of the function is further accelerated.

$$\vartheta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\mathrm{E}[g^2]_t + \varepsilon}} g_t \tag{10}$$

$\gamma = 0.9$， $\eta = 0.001$

4. Model Design

In this paper, we design a model with LSTM, which is divided into five parts, including input layer, Embedding layer, LSTM layer, Softmax layer and output layer.

4.1. Input layer

Before training model, the dataset will be partitioned into data blocks, that is, minibatch. A data block consists of more than one sentence, n_samples is the number of sentences. In the input layer, the sentences of a data blocks have the same length n_timesteps, and the size of the blocks minibatch is n_timesteps * n_sample.

The sentences in the minibatch have the same length by using mask matrix, and then passed to the Embedding layer.

In this paper, n_timesteps is the length of the longest sentence in minibatch. When training, a minibatch includes 64 sentences, and a minibatch includes 256 sentences during verification and testing.

In order to keep the same length of each sentence in minibatch, a mask matrix is needed. The value of each element in the mask matrix is 1 for the corresponding position of the word in minibatch, otherwise is 0.

4.2. Embedding layer

The Embedding layer transforms each word in minibatch into a vector, and then let them as the input x_t of the LSTM. The processing is shown in figure 2.

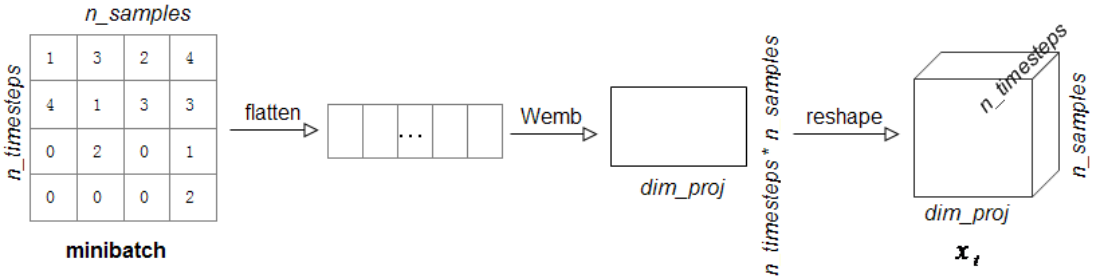

Fig. 2: The processing of the Embedding layer.

Firstly, making minibatch into one-dimensional array by using flatten function, and then getting the vector of each word from Wemb. Finally, transforming them into three-dimensional array x_t of n_timesteps * n_samples * dim_proj by reshape function, and let x_t as the input of LSTM.

In this paper, Wemb is 100000 * 128, that is, the size of the dictionary is 100000, and the dimension of the word vector is 128. n_timesteps is the length of the longest sentence. n_samples is the number of sentences, 64 for training and 256 for verification and testing. dim_proj is the dimension of the word vector, it equals to 128.

4.3. LSTM layer

Formula $W \cdot x_t + b$ can be used in input gate, new memory, forget gate and output gate, so we can combine them with:

$$W \cdot x_t + b = x_t \cdot lstm\_W + lstm\_b \tag{11}$$

x_t is the input of LSTM, $lstm\_W$ and $lstm\_b$ are the weight matrix and bias matrix of x_t for input gate, new memory, forget gate and output gate respectively. The dimension of $W \cdot x_t + b$ is n_timesteps * n_samples * (dim_proj * 4). dim_proj is 128 of the word dimension.

4.4. Softmax layer

In Softmax layer, Softmax regression model is used to learn the output information of LSTM layer. It will calculate the probability $pred$ of the commendatory and derogatory of each sentence, then transmit them to the output layer.

The average of each output of LSTM layer is taken as the final output $proj$, and the Softmax is processed.

$$pred = \text{softmax}(proj \cdot U + b) \tag{12}$$

U and b are weight matrix and bias of input in Softmax layer respectively.

4.5. Output Layer

The emotional label corresponding to the higher value of $pred$ is the result of the model, that is, the commendatory and derogatory of the input movie review are predicated by the movie review classification model.

5. Experimental Results

The dataset used in the experiment is a crawler dataset from Stanford, which comes from the Internet Movie Database IMDb contained with a total of 25000 data. First, the data in the dataset are scrambled, and then divided them into training set, validation set and test set according to the ratio of 8:1:1. If word ID is larger than 100000 (the size of dictionary), set it to 1, which is, set to unknown. Finally, the sentences contained in training set, validation set and test set are sorted according to the length.

In this paper, SGD, SGD & momentum, Adadelta and RMSprop optimization algorithm are applied to classify IMDb data. When using SGD, it calculates the gradient of minibatch for each iteration, and then updates the parameters of the model. But the convergence rate of SGD is slow, the suitable model cannot be trained at last. So, the stochastic gradient descent acceleration is carried out. The calculation formula is as follows:

$$\Delta\theta_t = momentum * \Delta\theta_{t-1} - \eta * g_t \tag{13}$$

$\Delta\theta_t$ is the update parameter of current time, $momentum$ is momentum, which is used for accelerated learning. In the experiment, $momentum$ sets to 0.9, and $\Delta\theta_{t-1}$ is the update parameter of last time.

In the beginning of the training, the learning rate was set to 0.5, and the stochastic gradient descent acceleration formula was used for acceleration training. When the training loss was less than 0.3, the stochastic gradient descent parameter updating formula was used for training, and the learning rate was attenuated by 50% per iteration for 15 times. The error rate, iteration number and time consuming of training set, validation set and test set are shown in table 1.

Table 1 error rate, iteration number, and time of training set, validation set and test set

	error rate			iteration number	time（s）
	Train	Valid	Test		
SGD & momentum	0.08174999	0.12239999	0.12919999	100	11187.7
Adadelta	0.02929999	0.1088	0.1156	62	7040.8
RMSprop	0.01834999	0.1216	0.12719999	100	11353.3

In the training process of stochastic gradient descent, the error rate and training loss of the validation set and test set are shown in figure 3.

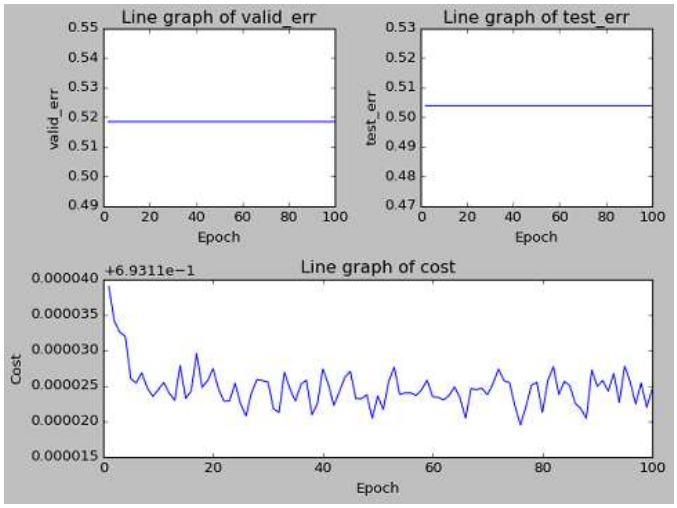

Fig. 3: error rate and training loss of validation set and test set in SGD training.

In the training process of stochastic gradient descent & momentum, the error rate and training loss of the validation set and test set are shown in figure 4.

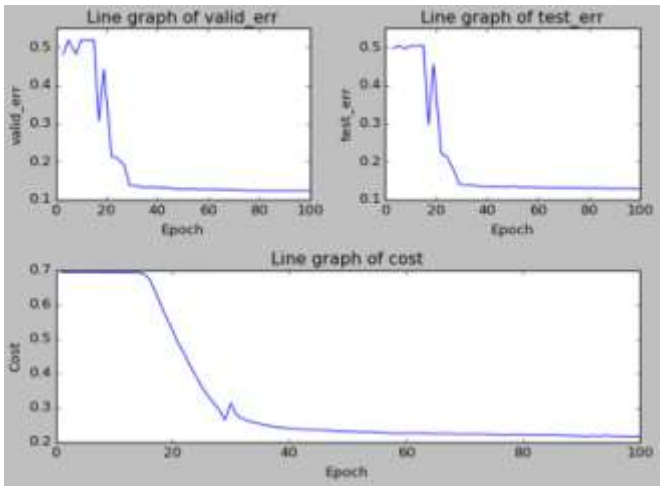

Fig. 4: error rate and training loss of validation set and test set in SGD accelerated training.

In the training process of Adadelta, the error rate and training loss of the validation set and test set are shown in figure 5.

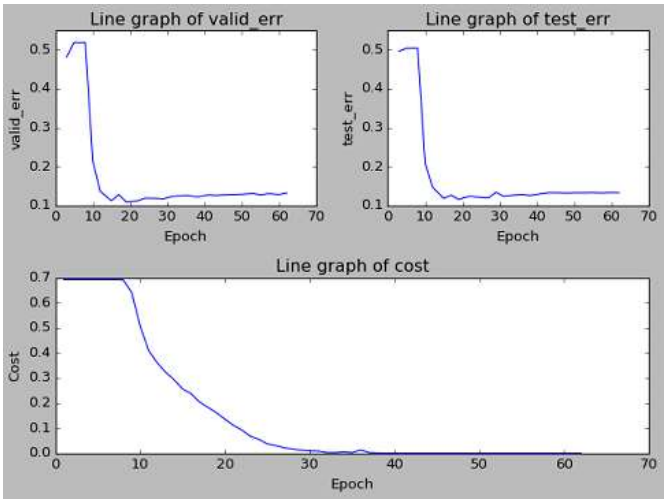

Fig. 5: error rate and training loss of validation set and test set in Adadelta training.

In the training process of RMSprop, the error rate and training loss of the validation set and test set are shown in figure 6.

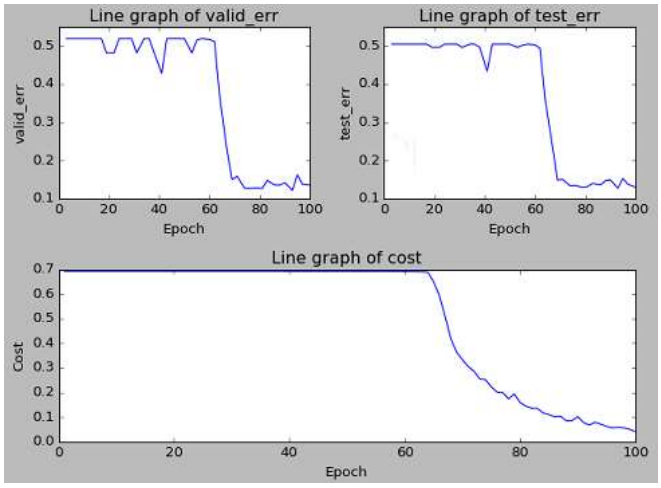

Fig. 6: error rate and training loss of validation set and test set in RMSprop training.

In the figures, the abscissa is Epoch, that is, the number of iterations of the dataset. The ordinates are valid_err, test_err, and Cost. They are the prediction error rate of the validation set, the prediction error rate of the test set, and the training loss respectively. The Cost is the average of the cost of all minibatch in Epoch.

5.1. Comparison

It can be seen from the training process of SGD that it is difficult to choose a suitable learning rate, it is inconvenient to adjust the learning rate in the course of training, SGD is easy to converge to the local optimal, and in some cases, it may be trapped in the saddle point. In this paper, the momentum algorithm is used to simulate the motion inertia of the object by using the momentum concept of physics. When the current gradient has the same direction with the historical gradient, the gradient of the direction will be enhanced and the learning speed is accelerated. Otherwise, the gradient will be attenuation and oscillation will be controlled.

Compared with SGD optimization algorithm, the training process of Adadelta optimization algorithm is relatively easy. Adadelta can adjust learning rate adaptively and does not rely on global learning rate. It can be seen from the figure that the speed of using Adadelta is very fast in the early and middle stages of training, and it often wavering near the local minimum value in the late training stage. This model has lower prediction error rate and is also more stable.

RMSprop optimization algorithm is a variant of Adadelta optimization algorithm, and it needs to rely on global learning rate. RMSprop is very slow at the early stage of training and drops suddenly in the later stage. The prediction error rate of the model is not stable.

So, the best model is trained by Adadelta, it does not depend on the global learning rate, and the training process is relatively easy. It is more suitable for training deeper and more complex networks to use the optimization method of adaptive learning rate.

6. Summary

This paper introduces the structure of LSTM, and polarity classification model based on LSTM, which can be used for emotional classification of the movie review data. The model contains five parts: input layer, Embedding layer, LSTM layer, Softmax layer and output layer. The input layer is responsible for making the comment sentences have the same length, and passing them to the Embedding layer. The Embedding layer converts the words of the sentence into vectors. The LSTM hidden layer learns the preprocessed sequence data, and passes the learning results to the next step for classification and prediction. The Softmax layer classifies the learning results into two categories. The output layer gives the classification results.

The experiment uses four optimization methods such as SGD, SGD & momentum, Adadelta and RMSprop to train the model. Comparing error rate and training loss of validation set and test set, the model trained by Adadelta is the best.

7. Acknowledgements

Project supported by Science and Technology Innovation Funding of Fujian Agriculture and Forestry University (Grant No. KFA 17032A), and the Research Fund of Education Department of Fujian (Grant No. JAT170894).

8. Reference

[1] S. H. Lee, J. Cui, and J. W. Kim. Sentiment analysis on movie review through building modified sentiment dictionary by movie genre. Journal of Intelligence & Information Systems. 2016, 22(2):97-113.

[2] E. Ahmed, M. A. U. Sazzad, M. T. Islam, and M. Azad. Challenges, comparative analysis and a proposed methodology to predict sentiment from movie reviews using machine learning. International Conference on Big Data Analytics & Computational Intelligence. IEEE, 2017, pp. 86-91.

[3] R. Remus and S. Rill. Data-Driven vs. Dictionary-Based Word n-Gram Feature Induction for Sentiment Analysis. Language Processing and Knowledge in the Web, 25th International Conference. Springer Berlin Heidelberg. 2013, pp. 176-183.

[4] L. Qu, R. Gemulla, and G. Weikum. A weakly supervised model for sentence-level semantic orientation analysis with multiple experts. Joint Conference on Empirical Methods in Natural Language Processing & Computational Natural Language Learning. Association for Computational Linguistics. 2012, pp. 149-159.

[5] Z. Hu, J. Hu, W. Ding, and X. Zheng. Review Sentiment Analysis Based on Deep Learning. International Conference on E-Business Engineering. IEEE. 2015, pp. 87-94.

[6] D. Tang, B. Qin, and T. Liu. Deep learning for sentiment analysis: successful approaches and future challenges. Wiley Interdisciplinary Reviews Data Mining & Knowledge Discovery. 2015,5(6): 292-303.

[7] S. Zhou, Q. Chen, and X. Wang. Active deep learning method for semi-supervised sentiment classification. Neurocomputing. 2013,120(10), pp. 536-546.

[8] N. B. M. Sharef and M. Y. Shafazand. An improved deep learning-based approach for sentiment mining. Information & Communication Technologies, 2015, pp. 344-348.

[9] M. D. Capua and A. Petrosino. A Deep Learning Approach to Deal with Data Uncertainty in Sentiment Analysis. International Workshop on Fuzzy Logic & Applications, 2016, pp. 172-184.

[10] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. Neural Computation. 1997, 9(8), pp. 1735-1780.

[11] L. Bottou. Large-Scale Machine Learning with Stochastic Gradient Descent. 19th International Conference on Computational StatisticsParis (COMPSTAT) 2010.

[12] M. D. Zeiler. ADADELTA: An Adaptive Learning Rate Method. Computer Science, 2012.