

MULTI CLASS TEXT CLASSIFICATION USING LSTM

^[1] Bhavya K, ^[2] Niyas Mohammed A, ^[3] Ajeesh Ramanujan

^{[1][2]} Dept of Computer Science and Engineering, Govt. Engineering College Sreekrishnapuram, Palakkad, Kerala, India

^[3] Asst. Prof. Dept. Computer Science and Engineering. Govt. Engineering College Sreekrishnapuram, Palakkad, Kerala, India
^[1] aamibhavya@gmail.com, ^[3] ajeeshramanujan@gmail.com

Abstract: Automatic multi class text classification is a machine learning task which categorizes document to one among a predefined set of classes. In recent years, deep learning technique such as Recurrent Neural Networks (RNNs) has become state-of-the-art model for a variety of machine learning problems. This paper introduces the scope of Long Short Term Memory (LSTM) - a type of RNN, for multi class text classification. LSTMs are capable of learning long-term dependencies while avoiding the vanishing gradient problem usually found in neural network algorithms. The proposed system is carried out in Reuters corpus, a dataset of 11,228 news wires from Reuters, labeled over 46 topics.

Keywords: LSTM; RNN; Deep learning; Embedding Layer; Dense Layer; Corpus

I. INTRODUCTION

Automatic text classification is an important topic because of the ever increasing volume of digital documents. Manual classification is a very tedious and time consuming process. Our goal is to minimize this effort by automation.

Text is a typical kind of sparse data. From a computation perspective, words that do not contribute much to the purpose can be discarded, whereas the word frequency distribution needs to be considered. So it is essential to design classification methods which effectively provisions for the characteristics of text data. Some classifiers which are commonly used for text classification are Rule Based Classifiers, SVM Classifiers, Decision Tree Classifiers, and Bayesian Classifiers [6].

In general, text classification can be divided into two classes. One is according to the topics discussed in the documents and second one according to the genre of the document. Topic-based text categorization classifies documents according to their topics. Genre based categorization classifies documents according to the genres like scientific articles, news reports, movie reviews, and advertisements. In this paper, we use LSTM neural network for classifying text documents to their respective topics.

The remainder of the paper is organized as follows. Section 2 describes, from a theoretical point of view, the basic structure of LSTM recurrent neural networks. Section 3 presents the proposed system and in Section 4, we describe the implementation details of the proposed system. Finally, conclusions and future research directions are discussed in Section 5.

II. LSTM RECURRENT NEURAL NETWORK

In mid-90s, a variation of recurrent net with Long Short-Term Memory units, or LSTMs, was proposed by the German researchers Sepp Hochreiter and Juergen Schmidhuber as a solution to the vanishing gradient problem which was an obstacle to recurrent net performance [3]. During neural network training with back propagation, the (local) minimum of the error function is found by iteratively taking small steps in the direction of the negative error derivative with respect to networks weights (i.e. gradients). With each subsequent layer the magnitude of the gradients gets exponentially smaller (vanishes) thus making the steps also very small which results in very slow learning of the weights in the lower layers of a deep network [2].

LSTM networks have the ability to remove or add information to the cell state, carefully regulated by gates [1]. Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer and a point wise multiplication operation [4].

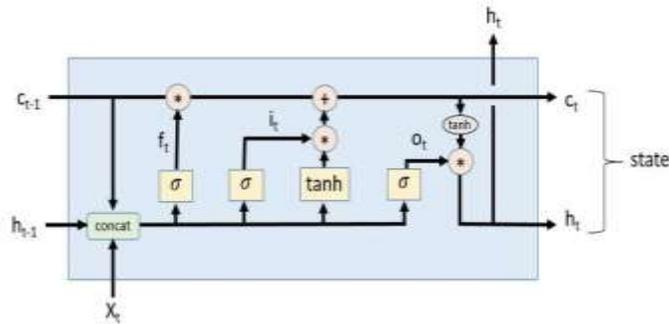


Fig. 1. Internals of an LSTM cell

Forget Gate:

$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

Input Gate:

$$i_t = \sigma (W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

Next a *Tanh* layer creates a vector of new candidate values, c_t , that could be added to the state:

$$c_t = \tanh (W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3)$$

Creates new cell state C_t by:

$$C_t = f_t * C_{t-1} + i_t * c_t \quad (4)$$

Output Gate:

$$o_t = \sigma (W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

In the above equations x_t is an element of the sequence at time t , h_{t-1} is the output of the memory cell at time $t-1$. w_f, w_i, w_c, w_o are the distinct weight matrices, and b_f, b_i, b_c, b_o are the bias terms respectively. σ and \tanh are nonlinear activation functions where σ denotes a sigmoid function. f_t, i_t, o_t are the output from forget gate, input gate, output gate respectively. c_{t-1} is the previous cell state and c_t is the new cell state.

In the above equations x_t is an element of the sequence at time t , h_{t-1} is the output of the memory cell at time $t-1$. w_f, w_i, w_c, w_o are the distinct weight matrices, and b_f, b_i, b_c, b_o are the bias terms respectively. σ and \tanh are nonlinear activation functions where σ denotes a sigmoid function. f_t, i_t, o_t are the output from forget gate, input gate, output gate respectively. C_{t-1} is the previous cell state and C_t is the new cell state.

III. PROPOSED SYSTEM

The Proposed method implements an LSTM neural network to classify a text to a topic. The overall system can be described in three stages- pre-processing, training and evaluation.

A. Pre-processing stage

We have used the Reuters corpus as our dataset. The documents underwent a pre-processing phase where the words were replaced with word indices. The corpus is divided into an 80:20 split between training and test sets.

B. Training stage

The structure of the neural network is defined and the training data is fed to this architecture as input.

C. Evaluation stage

The overall accuracy of the trained classifier over a test dataset consisting of 20% of the total documents is calculated. Some minor linear interactions. This helps for information to just flow along it unchanged. The functions of each gate can be summarized as follows [5].

IV. IMPLEMENTATION

In this proposed system we used the Reuters newswire corpus to classify an unseen news article into one of the existing categories. The corpus is a dataset of 11,228 news wires from Reuters, labeled over 46 topics. When we load the dataset, each wire is encoded as a sequence of word indexes. The data set split into training set and test set. Twenty percent of the total dataset is taken as the test set. The rest of the dataset used to train the model.

A. Model

We define a model for the classification problem. A neural network can be thought of as a network of “neurons” organized in layers. The inputs form the bottom layer, and the outputs form the top layer. There may be intermediate layers containing “hidden neurons”. In this implementation we add an embedding layer as the first layer and the output of this layer is given as the input for next LSTM layer. The last layer is a dense layer which accepts the output from LSTM layer and produce output as a probability over 46 categories.

B. Embedding Layer

Embedding layer generates word embedding by multiplying an index vector with a word embedding matrix. Word embedding is a key building block of deep learning models for NLP. Embedding layer takes words from corpus' vocabulary as input and embeds them as vectors into a lower dimensional space so that semantically similar word will be closer in the provisioned space, which it then fine tunes through backpropagation.

C. LSTM Layer

Long Short-Term Memory is a specific RNN architecture that designed to model the long-range dependencies more accurately than conventional RNNs [1]. The LSTM contains special units called memory blocks in the recurrent hidden layer. For each memory block there are three types of gates, input gate, output gate and forget gate. The input gate decides what new information it is going to store in the memory block whereas the output gate decides what part of a memory block it is going to output. The forget gate is for adaptive forgetting or resetting the cell's memory.

D. Dense Layer

A dense layer corresponds to a fully connected layer. After embedding layer and LSTM layer, the high-level reasoning in the neural network is done via fully connected layer. A fully connected layer takes all neurons in the previous layer and connects it to every single neuron it has. The number of neurons in dense layer is the total number of classes that we have in a classification problem. Softmax activation function is applied in this layer.

Fig. 2 shows a simplified architecture of the proposed system only for three word sequence. Depending on the classification domain we can vary the sequence length. We trained the deep neural network with fifteen epochs, where each epoch consists of one full training cycle on the training set. Once every sample in the set is seen, start again marking the beginning of the second epoch. Weights can be updated in two primary ways: batch training and on-line (also called sequential or pattern-based) training. Here we implemented batch training with a number of batches each with size of thirty two training samples taken from the training set.

This model is implemented in Python using Keras, a highly modular neural network library, capable of running on top of either Tensorflow or Theano.

3.1.2 L293D Motor Driver

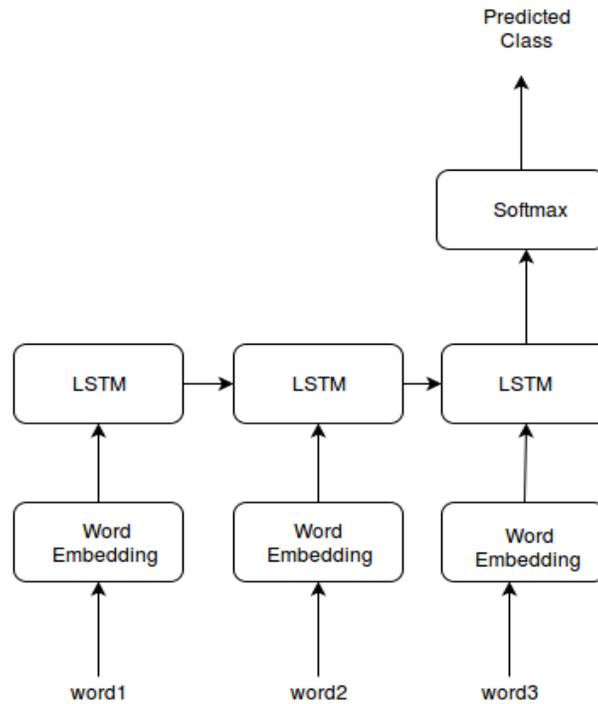


Fig 2. Simplified Architecture of the Proposed System

V. CONCLUSION AND FUTURE WORK

In this paper, we have presented a recurrent neural scheme for multi class document classification with long short term memory cells. The model built is not specific to a particular domain, but is rather a general implementation, and can be used to classify documents on a wide range of domains. More generally, we have demonstrated that deep learning can be effectively extended to classify text corpora. The method approach, though in infancy, is still promising.

In the future we plan to modify the internal connections of the LSTM cell and compare the performance of the resultant models. We also plan to work on a multi-layer implementation of the proposed method.

ACKNOWLEDGMENT

I would like to express my thanks and gratitude to the guides and advisors for their valuable support.

REFERENCES

- [1] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.

- [2] Baccouche, Moez, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt.. "Action classification in soccer videos with long short-term memory recurrent neural networks." *Artificial Neural Networks-ICANN 2010*. Springer Berlin Heidelberg, 2010. 154-159.
- [3] Liwicki, Marcus, and Horst Bunke. "Combining diverse on-line and off-line systems for handwritten text line recognition." *Pattern Recognition* 42.12 (2009): 3254-3263.
- [4] Wöllmer, Martin, Martin, Björn Schuller, Florian Eyben, and Gerhard Rigoll. "Combining long short-term memory and dynamic bayesian networks for incremental emotion-sensitive artificial listening." *Selected Topics in Signal Processing, IEEE Journal of* 4.5 (2010): 867-881.
- [5] Graves, Alex, Douglas Eck, Nicole Beringer, and Juergen Schmidhuber. "Biologically plausible speech recognition with LSTM neural nets." In *Biologically Inspired Approaches to Advanced Information Technology*, pp. 127-136. Springer Berlin Heidelberg, 2004.
- [6] Ren, Jimmy, Yongtao Hu, Yu-Wing Tai, Chuan Wang, Li Xu, Wenxiu Sun, and Qiong Yan. "Look, Listen and Learn-A Multimodal LSTM for Speaker Identification." *arXiv preprint arXiv:1602.04364* (2016).