A Recurrent Neural Network Model for Gender Classification

Abstract

In this paper, we present a simple recurrent neural network approach for the gender classification task. The strength of the model is that it can be trained end-to-end and thus does not require any feature extractions. Empirical evaluation using a real-life blog data set shows that our model is able to achieve good accuracy.

1. Introduction

The problem of automatically classifying the gender of a person has many potential commercial applications. With the rapid growth of the amount of information people published on the web, including blogs, tweets, and posts on public forums, recognizing people's gender by using textual information is receiving increasing attention.

In the past few years, several researchers have studied the problem of gender classifications. Most existing work require feature extractions, and majority of them only deal with formal writings. Our model is able to be trained end-to-end on unstructured, informal sentences which contain grammar errors, slang words, abbreviations, and spelling mistakes. We will discuss the model in detail in Section 2.

We trained our model on real-life blogs collected from the Internet. We also experimented with reddit comments from two subreddits(femalefashionadvice and Battlefield), the result shows the model successfully recognized the male-oriented subreddit(Battlefield), and the famale-oriented subreddit(femalefashionadvice).

2. Model

A standard recurrent neural network (RNN)(D. Rumelhart & Williams, 1986; Werbos, 1990) is a nonlinear dynamical system that maps sequences to sequences. Given a sequence of input$(x_1, x_2, ..., x_T)$, a RNN computes a sequence of output $(o_1, o_2, ..., o_T)$ by iterating the following equation:

$$h_t = f(W_{hx}x_t + W_{hh}h_{t-1} + b_h)$$
$$o_t = g(W_{oh}h_t + b_o)$$

f(·) and g(·) are nonlinear functions. W_{hx}, W_{hh}, W_{oh}, b_h, b_o denote weight matrices and bias vectors. The derivatives of the RNNs can be computed with the backpropagation through time algorithm(BPTT)(Werbos, 1990; Rumelhart & Williams, 1986). It is known that RNNs are difficult to train on problems with long-term dependencies(Bengio & Frasconi, 1994). However, the Long Short-Term Memory (LSTM)(Hochreiter & Schmidhuber, 1997) is able to solve problems with pathological long-range temporal dependencies.

We propose a 4-layer recurrent neural network model with LSTM units. Each layer has an input dimension of 128 and an output dimension of 128, except for the last layer which has an output dimension of 1. Dropout units(Srivastava et al., 2014) with probability of 0.5 of dropping out values are inserted between two consecutive recurrent layers. The model takes a sequence of word vectors with a size of 128, and produces a binary value indicating author's gender. The goal of the model is to estimate the conditional probability $p(y|x_1, ..., x_T)$ where $(x_1, ..., x_T)$ is an input sequence of word vectors, and y is its corresponding output value, or the gender indication value.

3. Experiments

3.1. Dataset

we selected blogs with a length between 50 to 200 words from the blog dataset. 47% of blogs are written by female authors, 52% of blogs are written by male authors. Our model was trained on 98,729 blogs and tested on 10,970 blogs. We used 50,000 of the most frequent words from the blog dataset. Punctuations, tabs, and newlines were filtered out and out-of-

---

CS 486 Bonus Project

vocabulary words were replaced with a special "UNK" token.

3.2. Training details

We experimented with several optimization methods, including standard stochastic gradient descent, RMSprop(Tieleman & Hinton, 2012) and Adam(Kingma & Ba, 2014). Adam achieved the highest accuracy in the testing dataset. Weights were initialized with random orthogonal conditions(Saxe et al., 2013). We used batches of 512 randomly chosen sentences, and padded sentences with zeros if their lengths are less than 512. The whole training process toke about 5 hours, and reached the highest cross-validation accuracy in the 7th epoch.

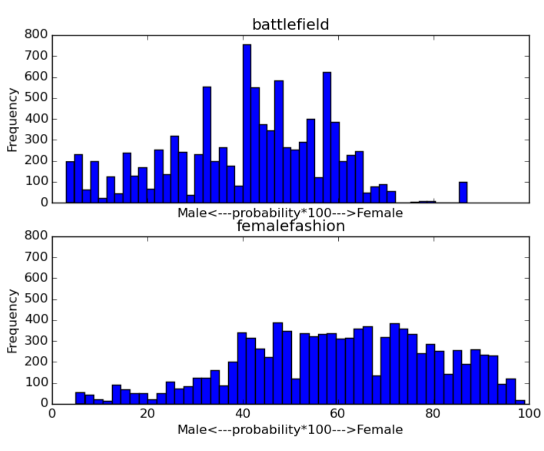
3.3. Experimental Results

In order to compare our approach, we considered six other models mentioned in (Arjun Mukherjee, 2010). We need to be careful when we compare results since the other six models were tested on a different blog dataset. The results are presented in table 1.

Table 1

System	Accuracy
Gender Genie	61.69
Gender Guesser	63.78
(Yan and Yan, 2006)	68.75
Our method	**72.2**
(Argamon et al., 2007)	77.86
(Schler et al., 2006)	79.63
(Arjun Mukherjee, 2010)	88.56

We also tested our model with 10000 comments collected from two subreddits(femalefashionadvice and Battlefield). The battlefield subreddit is male-oriented and the femalefashionadvice subreddit is female-oriented. The predicted distribution of male and female authors is presented below.



3.4. Implementation

We used variety of tools to implement our method. For the recurrent neural network, we created a python implementation with Theano and Keras. A reddit api called PRAW was used to extract comments from reddit. We used Panda and Keras' text prepossessing scripts for data preparation.

4. Conclusion

In this paper we considered the problem of gender classification in informal, unstructured blog dataset. Our modest results show that a simple recurrent neural network model is a viable option for gender classification in a noisy open-domain dataset.

References

Arjun Mukherjee, Bing Liu. Improving gender classification of blog authors. In *In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Cambridge, MA, October. Association for Computational Linguistics*, 2010.

Bengio, Y., Simard P. and Frasconi. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions*, 5(2):157–166, 1994.

D. Rumelhart, G. E. Hinton and Williams, R. J. Learning representations by back-propagating errors. In *Nature*, pp. 323(6088):533–536, 1986.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. In *Neural Computation*, 1997.

Kingma, D. and Ba, J. Adam: A Method for Stochastic Optimization. *ArXiv e-prints*, December 2014.

Langley, P. Crafting papers on machine learning. In Langley, Pat (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Rumelhart, D., Hinton G. and Williams, R. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *ArXiv e-prints*, December 2013.

Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan.

Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

Tieleman, Tijmen and Hinton. Lecture 6.5- rmsprop: Divide the gradient by a running average of its recent magnitude. In *COURSERA: Neural Networks for Machine Learning, 4*, 2012.

Werbos, P. Backpropagation through time: what it does and how to do it. In *Proceedings of IEEE, 1990*, 1990.