Research Article

Hierarchical LSTM network for text classification

Keivan Borna[1] · Reza Ghanbari[1]

Abstract
Text classification has always been an important and practical issue so that we need to use the computer to classify and discover the information in the text. If we want to recognize the offending words in a text without human intervention, we should use this. In this article we will compare recurrent neural networks, convolutional neural networks and hierarchical attention networks with detailed information about each of which. We will represent a HAN model using Theano framework, which indicates more accurate validation for large datasets. For text classification problem in large datasets, we will use hierarchical attention networks to get a better result.

Keywords Computer science · Machine learning · Text classification · Hierarchical attention network

1 Introduction

Text classification is an important and practical issue that can be used in many cases, like spam detection, smart automatic customer reply, sentiment analysis. These are commonly known as the most important topics in natural language processing (NLP) and natural language generation (NLG). The main goal in text classification is to assign text to one or more categories. Suppose in a profanity check problem we have to find the offensive words in document. Nowadays, machine learning is the outstanding way to create such classifiers. These classifiers are upon classification rules. So with the help of labeled documents we can create classifiers. There are a lot of traditional methods for text classification, such as n-grams with a linear model. Recent researches are using supervised and unsupervised machine learning methods, such as convolutional neural network (CNN) [1], recurrent neural network (RNN) or hierarchical neural network (HAN). In this article we benchmark these three methods with creating a general text classifier using these three methods on GloVe d-300 dataset. Our primary contribution is benchmark these methods and building a Hierarchical LSTM network, which the input tensor is 3D rather than 2D to demonstrate documents as a hierarchical model and retrieve categories. The key difference to previous works is that our algorithm uses tokens that are taken from context (not just filtering sequences of tokens). In order to check the performance of our model, we looked at three datasets, to compare CNN, RNN and HAN. Our model uses hierarchical LSTM network.

2 Convolutional neural networks

Convolutional neural networks are group of neurons with weights and biases that we can learn them. With the score function, for example for a classification problem, from raw text to categories, it receives inputs calculate a differentiable score. For a common 3-layer neural network, a convolutional neural network put its neurons in 3 dimensions (x, y, z), in a Euclidean space. The duty of every layer in a CNN is converting a 3-dimension input to a 3-dimension output set of neurons. Actually the input layer is according to the problem, which means the input layer value is 2D document (rows, columns) and the other layers will hold characteristic values for input properties. We can find that every CNN is a

✉ Keivan Borna, borna@khu.ac.ir; Reza Ghanbari, reza91@aut.ac.ir | [1]Department of Computer Science, Faculty of Mathematics and Computer Science, Kharazmi University, Tehran, Iran.

SN Applied Sciences (2019) 1:1124 | https://doi.org/10.1007/s42452-019-1165-1

Received: 21 May 2019 / Accepted: 26 August 2019 / Published online: 30 August 2019

sequence of layers and every layer converts a vector value to another one with the help of a score function. In a simple way, suppose how a child learns how to speak and how to classify the words in a sentence, in order to create an effective model, we need to show algorithm millions of categorized texts. In a CNN the neurons only connect to small region of next layer not all of them. The building block of CNN is convolution which refers to combining two functions to create a third one. As we know from neural networks, we can use activation function to make a non-linear out and the output of convolution will be passed through this function. After the convolution part, the classification part consists of connected layers, which these layers can only accept 1D input vector. Note that in this part we should consider flatting function to generate a 1D from 3D [2].

3 Recurrent neural networks

Suppose if you want to think about a problem, you won't think from scratch. It means you don't have to rebuild and not use all your previous intellectual foundations. We understand everything based on the previous training [3]. In a traditional neural network, you have to rebuild all structures from scratch, which means it does not have persistent thinking capability. With recurrent neural networks, we can use the previous events in a input layer, such as context of a document, and use it in later events. Every unrolled recurrent neural network, contains multiple networks, that are same in structure and functionality. Recurrent neural networks, like convolutional, accept and produce fixed-sized input vectors and amount of computational steps is the number of layers. The main feature of RNNs is they process over input and output tensors and the output vector is influenced by input and all past inputs during the completion period. Usually recurrent neural networks have STM. We combine a long term model of historical thinking with STM, and it will produce a LSTM based RNN. In case of using the general RNN, we will face to two issues. Both of them are with gradients [4]. With bad weight choice in algorithm we will have an exploding gradient. Contrary to this case, when gradient values are too small we will have vanishing gradients which means model stops to learn or takes too long to learn. These issues are solved with the help of LSTM. Actually they are an extension for RNN, the results of past thoughts as a memory [5].

4 Hierarchical neural networks

The HAN network consists of several parts. According to the name of this model, as expected, the structure of this model is in the form of interconnected and network layers so that these layers are independent. In a text classification problem, usually we divide the whole architecture to these four parts. Word encoder, specific time distributed HAN model for words, bidirectional LSTM encoder for sentence and specific model for sentences.

5 About natural language processing (NLP)

Actually natural language processing is a branch of artificial intelligence. Text classification is one of use cases of NLP. With the help of natural language processing we can learn computers to learn about human language. With growth of classification algorithms, the ability to more progress in this field was discovered. Suppose you want to classify news of a Chinese broadcast website, with the help of NLP you can convert and classify text to any valuable data, like category management, monitoring the custom news, retrieving breaking news for a keyword or for example you can classify the user comments in an online market website like Amazon, to get the positive comments percentage for a new brand. These are approachable with NLP.

6 Classifier pipeline

Our classifier consists of several steps. We should prepare training data which our method will predict the requested queries. After that we should make vector from input data, also providing labels. We use HAN algorithm for text classification in our model and the result of these processes is a predictive model [6].

7 Analyzing our data

We are using 3 datasets with a different type. Each of which has various classes. Below is a detailed table of datasets (Table 1).

Table 1 Our datasets

Dataset	Data size	Classes	Train/validation samples
Dataset 1	3155	29	14658/3664
Dataset 2	18,322	362	2524/631
Dataset 3	191	17	153/38

8 Text classification algorithms

8.1 Text classification using CNN

CNN is a feed-forward and deep method and there isn't cycle among nodes. In this method, variation is designed to require minimal preprocessing. This method will return a pattern, that represents a result of convolution. If we put together the results of these processes, we can detect multiple size patterns. CNN identify the patterns in the sentence without knowing the word index in sentence. This works like n-grams.

The architecture of this methods will start with input layer, contains input vector and output vector. We will continue with embedding layer, and a triple cycle that each of which contains convolution layer, pooling, and at the last creating a one dimensional vector, we name it flattening layer and also we need a dense layer to compact the results.

8.2 Text classification using RNN

RNN is connected node network which uses a directed graph structure. We can refer to historical memory process using the directions among graph nodes. In a RNN we will convert text to vector using a tokenizer. The numbers of vector, represent the index of word in a sentence or document. It depends on out band of tokenization process. We have used a LSTM tokenizer in our comparison, which means we have split the words using the historical knowledge [7]. We use a feed-forward network for classification. In order to get a better result in the tokenization step, we have use both feed-forward and bidirectional LSTM. The architecture of this model is a simple four simple steps. Input layer that contains input and output vector. Embedding layer, bidirectional LSTM layer and at the end a dense layer to compact the results.

8.3 Text classification using HAN

The architecture of a HAN model is like RNN with a key change. At the second step we have a time distributed model instead of embedding layer. We also use a bidirectional LSTM in third step. We use Keras python library to create time distribution model. Keras is a high-level neural network library, written in Python. We can use it on top of many neural network frameworks like Tensorflow, Microsoft CNTK, Cafe2 and Theano. Keras magic function TimeDistributed, constructs a HAN according to above architecture. This is a character level deep learning model which builds a sentiment model and process the sentence encoder. We can also use this model in CNN.

9 Results

According to the following table we have benchmarked three different methods with three different datasets. CNN method has good validation accuracy, but the RNN and HAN are not consistent for all datasets. Which means CNN method is a good general method, we can refer to this in many problems can we will get a good validation accuracy result. There is a lot of hard work on implementation of RNN. In case of using a RNN model in production, we should have a huge dataset with a good training hardware. In huge datasets, HAN outperforms other methods in final result. In dataset 1 and 2 with HAN has the best validation accuracy. HAN is not a good option in small datasets but CNN method is really better for small datasets (Table 2).

To achieve the best results, we should fine-tune the learning rate, batch size and epoch numbers. There are a lot of methods for fine tuning like manual searching, random search, grid search and etc. The preprocessing stage is really important. Eliminating irrelevant and illegible data helps us to get a better result. With the help of regularization layers, we can avoid overfitting. Dropout is a regularization technique which increases the validation

Table 2 Outputs of algorithms on datasets

Dataset	Algorithm	Time/epoch (s)	Training accuracy	Validation accuracy
Dataset 1	CNN	51	96.138	94.12
Dataset 1	RNN	171	95.699	95.21401
Dataset 1	HAN	253	95.603	95.21465
Dataset 2	CNN	361	87.901208	83.45
Dataset 2	RNN	1812	82.598	80.98
Dataset 2	HAN	1338	87.9111	85.931
Dataset 3	CNN	5.1111	92.12	92.897
Dataset 3	RNN	157	89.487	92.135678
Dataset 3	HAN	31.12	93.12	87.14

accuracy. Actually there's no doubt that we'll need the right hardware to get the best results.

10 Conclusions

Voice classification in non-classical music or pedestrian checker using modern algorithms would be interesting to study and we have to define an intelligent model for classification. There are many models that fit to this problem but we can improve performance, as if we want to classify the voice over wireless Walkie Talkie in a very busy telecommunications network with thousands of users. In a Police Telecommunication Network, we are faced with a difficult sound recognition problem. On the other hand, other applications of this model can be mentioned in the industry and daily work. Of course, much work has been done in this field and still more can be done. Another problem is detecting items in an image. On the other hand, we have used many metrics on these models. Finding better metrics instead of Minkowski metric is another interesting problem to study.

Acknowledgements The authors would like to thank Prof. Bahram Sadeghi Bigham for kind hospitality and invitation to the CIDAS 2019, Zanjan, Iran.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Kim Y (2018) Convolutional neural networks for sentence classification. ACL, Stroudsburg
2. Zhou Ch, Sun Ch, Lau F (2015) A C-LSTM neural network for text classification. https://arxiv.org/abs/1511.08630
3. Lee JY, Dernoncourt F (2016) Sequential short-text classification with recurrent and convolutional neural networks. ACL, Stroudsburg
4. Liu P, Qiu X, Huang X (2016) Recurrent neural network for text classification with multi-task learning. In: IJCAI-16, pp 2873–2879
5. Lyyer M, Manjunatha V, Boyd-Garber J, Daume H (2015) Deep unordered composition rivals syntactic methods for text classification. Association for Computational Linguistics, Stroudsburg, pp 1681–1691
6. Li J, Luong M, Jurafsky D (2015) A hierarchical neural autoencoder for paragraphs and documents. Association for Computational Linguistics, Stroudsburg, pp 1106–1115
7. Yang Z, Yang D, Dyer Ch, He X, Smola A, Hovy E (2016) Hierarchical attention networks for document classification. In: HLT-NAACL