AC-BLSTM: Asymmetric Convolutional Bidirectional LSTM Networks for Text Classification

Abstract

Recently deeplearning models have been shown to be capable of making remarkable performance in sentences and documents classification tasks. In this work, we propose a novel framework called AC-BLSTM for modeling sentences and documents, which combines the asymmetric convolution neural network (ACNN) with the Bidirectional Long Short-Term Memory network (BLSTM). Experiment results demonstrate that our model achieves state-of-the-art results on five tasks, including sentiment analysis, question type classification, and subjectivity classification. In order to further improve the performance of AC-BLSTM, we propose a semi-supervised learning framework called G-AC-BLSTM for text classification by combining the generative model with AC-BLSTM.

Introduction

Deep neural models recently have achieved remarkable results in computer vision BIBREF0 , BIBREF1 , BIBREF2 , BIBREF3 , and a range of NLP tasks such as sentiment classification BIBREF4 , BIBREF5 , BIBREF6 , and question-answering BIBREF7 . Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) especially Long Short-term Memory Network (LSTM), are used wildly in natural language processing tasks. With increasing datas, these two methods can reach considerable performance by requiring only limited domain knowledge and easy to be finetuned to specific applications at the same time.

CNNs, which have the ability of capturing local correlations of spatial or temporal structures, have achieved excellent performance in computer vision and NLP tasks. And recently the emerge of some new techniques, such as Inception module BIBREF8 , Batchnorm BIBREF9 and Residual Network BIBREF3 have also made the performance even better. For sentence modeling, CNNs perform excellently in extracting n-gram features at different positions of a sentence through convolutional filters.

RNNs, with the ability of handling sequences of any length and capturing long-term dependencies, , have also achieved remarkable results in sentence or document modeling tasks. LSTMs BIBREF10 were designed for better remembering and memory accesses, which can also avoid the problem of gradient exploding or vanishing in the standard RNN. Be capable of incorporating context on both sides of every position in the input sequence, BLSTMs introduced in BIBREF11 , BIBREF12 have reported to achieve great performance in Handwriting Recognition BIBREF13 , and Machine Translation BIBREF14 tasks.

Generative adversarial networks (GANs) BIBREF15 are a class of generative models for learning how to produce images. Basically, GANs consist of a generator G and a discriminator D, which are trained based on game theory. G maps a input noise vector to an output image, while D takes in an image then outputs a prediction whether the input image is a sample generated by G. Recently, applications of GANs have shown that they can generate promising results BIBREF16 , BIBREF17 . Several recent papers have also extended GANs to the semi-supervised context BIBREF18 , BIBREF19 by simply increasing the dimension of the classifier output from INLINEFORM0 to INLINEFORM1 , which the samples of the extra class are generated by G.

In this paper, We proposed an end-to-end architecture named AC-BLSTM by combining the ACNN with the BLSTM for sentences and documents modeling. In order to make the model deeper, instead of using the normal convolution, we apply the technique proposed in BIBREF8 which employs a INLINEFORM0 convolution followed by a INLINEFORM1 convolution by spatial factorizing the INLINEFORM2 convolution. And we use the pretrained word2vec vectors BIBREF20 as the ACNN input, which were trained on 100 billion words of Google News to learn the higher-level representations of n-grams. The outputs of the ACNN are organized as the sequence window feature to feed into the multi-layer BLSTM.

So our model does not rely on any other extra domain specific knowledge and complex preprocess, e.g. word segmentation, part of speech tagging and so on. We evaluate AC-BLSTM on sentence-level and document-level tasks including sentiment analysis, question type classification, and subjectivity classification. Experimental results demonstrate the effectiveness of our approach compared with other state-of-the-art methods. Further more, inspired by the ideas of extending GANs to the semi-supervised learning context by BIBREF18 , BIBREF19 , we propose a semi-supervised learning framework for text classification which further improve the performance of AC-BLSTM.

The rest of the paper is organized as follows. Section 2 presents a brief review of related work. Section 3 discusses the architecture of our AC-BLSTM and our semi-supervised framework. Section 4 presents the experiments result with comparison analysis. Section 5 concludes the paper.

Related Work

Deep learning models have made remarkable progress in various NLP tasks recently. For example, word embeddings BIBREF20 , BIBREF21 , question answearing BIBREF7 , sentiment analysis BIBREF22 , BIBREF23 , BIBREF24 , machine translation BIBREF25 and so on. CNNs and RNNs are two wildly used architectures among these models. The success of deep learning models for NLP mostly relates to the progress in learning distributed word representations BIBREF20 , BIBREF21 . In these mothods, instead of using one-hot vectors by indexing words into a vocabulary, each word is modeled as a low dimensional and dense vector which encodes both semantic and syntactic information of words.

Our model mostly relates to BIBREF4 which combines CNNs of different filter lengths and either static or fine-tuned word vectors, and BIBREF5 which stacks CNN and LSTM in a unified architecture with static word vectors. It is known that in computer vision, the deeper network architecture usually possess the better performance. We consider NLP also has this property. In order to make our model deeper, we apply the idea of asymmetric convolution introduced in BIBREF8 , which can reduce the number of the parameters, and increase the representation ability of the model by adding more nonlinearity. Then we stack the multi-layer BLSTM, which is cable of analysing the future as well as the past of every position in the sequence, on top of the ACNN. The experiment results also demonstrate the effectiveness of our model.

AC-BLSTM Model

In this section, we will introduce our AC-BLSTM architecture in detail. We first describe the ACNN which takes the word vector represented matrix of the sentence as input and produces higher-level presentation of word features. Then we introduce the BLSTM which can incorporate context on both sides of every position in the input sequence. Finally, we introduce the techniques to avoid overfitting in our model. An overall illustration of our architecture is shown in Figure FIGREF1 .

Asymmetric Convolution

Let x INLINEFORM0 be the INLINEFORM1 -dimensional word vector corresponding to the INLINEFORM2 -th word in the sentence and INLINEFORM3 be the maximum length of the sentence in the dataset. Then the sentence with length INLINEFORM4 is represented as DISPLAYFORM0

For those sentences that are shorter than INLINEFORM0 , we simply pad them with space.

In general, let INLINEFORM0 in which INLINEFORM1 be the length of convolution filter. Then instead of employing the INLINEFORM2 convolution operation described in BIBREF4 , BIBREF5 , we apply the asymmetric convolution operation inspired by BIBREF8 to the input matrix which factorize the INLINEFORM3 convolution into INLINEFORM4 convolution followed by a INLINEFORM5 convolution.

And in experiments, we found that employ this technique can imporve the performance. The following part of this subsection describe how we define the asymmetric convolution layer.

First, the convolution operation corresponding to the INLINEFORM0 convolution with filter w INLINEFORM1 is applied to each word x INLINEFORM2 in the sentence and generates corresponding feature m INLINEFORM3 DISPLAYFORM0

where INLINEFORM0 is element-wise multiplication, INLINEFORM1 is a bias term and INLINEFORM2 is a non-linear function such as the sigmoid, hyperbolic tangent, etc. In our case, we choose ReLU BIBREF26 as the nonlinear function. Then we get the feature map m INLINEFORM3 DISPLAYFORM0

After that, the second convolution operation of the asymmetric convolution layer corresponding to the INLINEFORM0 convolution with filter w INLINEFORM1 is applied to a window of INLINEFORM2 features in the feature map m INLINEFORM3 to produce the new feature c INLINEFORM4 and the feature map c INLINEFORM5 DISPLAYFORM0 DISPLAYFORM1

with c INLINEFORM0 . Where INLINEFORM1 , INLINEFORM2 and INLINEFORM3 are the same as described above.

As shown in Figure FIGREF1 , we simultaneously apply three asymmetric convolution layers to the input matrix, which all have the same number of filters denoted as INLINEFORM0 . Thus the output of the asymmetric convolution layer has INLINEFORM1 feature maps. To generate the input sequence of the BLSTM, for each output sequence of the second convolution operation in the aysmmetric convolution layer, we slice the feature maps by channel then obtained sequence of INLINEFORM2 new features c INLINEFORM3 where INLINEFORM4 . Then we concatanate c INLINEFORM5 , c INLINEFORM6 and c INLINEFORM7 to get the input feature for each time step DISPLAYFORM0 where INLINEFORM0 for INLINEFORM1 and INLINEFORM2 . In general, those c INLINEFORM3 where INLINEFORM4 and INLINEFORM5 must be dropped in order to maintain the same sequence length, which will cause the loss of some information. In our model, instead of simply cutting the sequence, we use a simple trick to obtain the same sequence length without losing the useful information as shown in Figure FIGREF2 . For each output sequence INLINEFORM6 obtained from the second convolution operation with filter length INLINEFORM7 , we take those c INLINEFORM8 where INLINEFORM9 then apply a fullyconnected layer to get a new feature, which has the same dimension of c INLINEFORM10 , to replace the (INLINEFORM11 +1)-th feature in the origin sequence.

Bidirectional Long Short-Term Memory Network

First introduced in BIBREF10 and shown as a successful model recently, LSTM is a RNN architecture specifically designed to bridge long time delays between relevant input and target events, making it suitable for problems where long range context is required, such as handwriting recognition, machine translation and so on.

For many sequence processing tasks, it is useful to analyze the future as well as the past of a given point in the series. Whereas standard RNNs make use of previous context only, BLSTM BIBREF11 is explicitly designed for learning long-term dependencies of a given point on both side, which has also been shown to outperform other neural network architectures in framewise phoneme recognition BIBREF12 .

Therefore we choose BLSTM on top of the ACNN to learn such dependencies given the sequence of higher-level features. And single layer BLSTM can extend to multi-layer BLSTM easily. Finally, we concatenate all hidden state of all the time step of BLSTM, or concatenate the last layer of all the time step hidden state of multi-layer BLSTM, to obtain final representation of the text and we add a softmax layer on top of the model for classification.

Semi-supervised Framework

Our semi-supervised text classification framewrok is inspired by works BIBREF18 , BIBREF19 . We assume the original classifier classify a sample into one of INLINEFORM0 possible classes. So we can do semi-supervised learning by simply adding samples from a generative network G to our dataset and labeling them to an extra class INLINEFORM1 . And correspondingly the dimension of our classifier output increases from INLINEFORM2 to INLINEFORM3 . The configuration of our generator network G is inspired by the architecture proposed in BIBREF16 . And we modify the architecture to make it suitable to the text classification tasks. Table TABREF13 shows the configuration of each layer in the generator G. Lets assume the training batch size is INLINEFORM4 and the percentage of the generated samples among a batch training samples is INLINEFORM5 . At each iteration of the training process, we first generate INLINEFORM6 samples from the generator G then we draw INLINEFORM7 samples from the real dataset. We then perform gradient descent on the AC-BLSTM and generative net G and finally update the parameters of both nets.

Regularization

For model regularization, we employ two commonly used techniques to prevent overfitting during training: dropout BIBREF27 and batch normalization BIBREF9 . In our model, we apply dropout to the input feature of the BLSTM, and the output of BLSTM before the softmax layer. And we apply batch normalization to outputs of each convolution operation just before the relu activation. During training, after we get the gradients of the AC-BLSTM network, we first calculate the INLINEFORM0 INLINEFORM1 of all gradients and sum together to get INLINEFORM2 . Then we compare the INLINEFORM3 to 0.5. If the INLINEFORM4 is greater than 0.5, we let all the gradients multiply with INLINEFORM5 , else just use the original gradients to update the weights.

Datasets

We evaluate our model on various benchmarks. Stanford Sentiment Treebank (SST) is a popular sentiment classification dataset introduced by BIBREF33 . The sentences are labeled in a fine-grained way (SST-1): very negative, negative, neutral, positive, very positive. The dataset has been split into 8,544 training, 1,101 validation, and 2,210 testing sentences. By removing the neutral sentences, SST can also be used for binary classification (SST-2), which has been split into 6,920 training, 872 validation, and 1,821 testing. Since the data is provided in the format of sub-sentences, we train the model on both phrases and sentences but only test on the sentences as in several previous works BIBREF33 , BIBREF6 .

Movie Review Data (MR) proposed by BIBREF34 is another dataset for sentiment analysis of movie reviews. The dataset consists of 5,331 positive and 5,331 negative reviews, mostly in one sentence. We follow the practice of using 10-fold cross validation to report the result.

Furthermore, we apply AC-BLSTM on the subjectivity classification dataset (SUBJ) released by BIBREF35 . The dataset contains 5,000 subjective sentences and 5,000 objective sentences. We also follow the practice of using 10-fold cross validation to report the result.

We also benchmark our system on question type classification task (TREC) BIBREF36 , where sentences are questions in the following 6 classes: abbreviation, human, entity, description, location, numeric. The entire dataset consists of 5,452 training examples and 500 testing examples.

For document-level dataset, we use the sentiment classification dataset Yelp 2013 (YELP13) with user and product information, which is built by BIBREF22 . The dataset has been split into 62,522 training, 7,773 validation, and 8,671 testing documents. But in the experiment, we neglect the user and product information to make it consistent with the above experiment settings.

Training and Implementation Details

We implement our model based on Mxnet BIBREF37 - a C++ library, which is a deep learning framework designed for both efficiency and flexibility. In order to benefit from the efficiency of parallel computation of the tensors, we train our model on a Nvidia GTX 1070 GPU. Training is done through stochastic gradient descent over shuffled mini-batches with the optimizer RMSprop BIBREF38 . For all experiments, we simultaneously apply three asymmetric convolution operation with the second filter length INLINEFORM0 of 2, 3, 4 to the input, set the dropout rate to 0.5 before feeding the feature into BLSTM, and set the initial learning rate to 0.0001. But there are some hyper-parameters that are not the same for all datasets, which are listed in table TABREF14 . We conduct experiments on 3 datasets (MR, SST and SUBJ) to verify the effectiveness our semi-supervised framework. And the setting of INLINEFORM1 and INLINEFORM2 for different datasets are listed in table TABREF15 .

Word Vector Initialization

We use the publicly available word2vec vectors that were trained on 100 billion words from Google News. The vectors have dimensionality of 300 and were trained using the continuous bag-of-words architecture BIBREF20 . Words not present in the set of pre-trained words are initialized from the uniform distribution [-0.25, 0.25]. We fix the word vectors and learn only the other parameters of the model during training.

Results and Discussion

We used standard train/test splits for those datasets that had them. Otherwise, we performed 10-fold cross validation. We repeated each experiment 10 times and report the mean accuracy. Results of our models against other methods are listed in table TABREF16 . To the best of our knowledge, AC-BLSTM achieves the best results on five tasks.

Compared to methods BIBREF4 and BIBREF5 , which inspired our model mostly, AC-BLSTM can achieve better performance which show that deeper model actually has better performance. By just employing the word2vec vectors, our model can achieve better results than BIBREF30 which combines multiple word embedding methods such as word2vec BIBREF20 , glove BIBREF21 and Syntactic embedding. And the AC-BLSTM performs better when trained with the semi-supervised framework, which proves the success of combining the generative net with AC-BLSTM.

The experiment results show that the number of the convolution filter and the lstm memory dimension should keep the same for our model. Also the configuration of hyper-parameters: number of the convolution filter, the lstm memory dimension and the lstm layer are quiet stable across datasets. If the task is simple, e.g. TREC, we just set number of convolution filter to 100, lstm memory dimension to 100 and lstm layer to 1. And as the task becomes complicated, we simply increase the lstm layer from 1 to 4. The SST-2 is a special case, we find that if we set the number of convolution filter and lstm memory dimension to 300 can get better result. And the dropout rate before softmax need to be tuned.

Conclusions

In this paper we have proposed AC-BLSTM: a novel framework that combines asymmetric convolutional neural network with bidirectional long short-term memory network. The asymmetric convolutional layers are able to learn phrase-level features. Then output sequences of such higher level representations are fed into the BLSTM to learn long-term dependencies of a given point on both side. To the best of our knowledge, the AC-BLSTM model achieves top performance on standard sentiment classification, question classification and document categorization tasks. And then we proposed a semi-supervised framework for text classification which further improve the performance of AC-BLSTM. In future work, we plan to explore the combination of multiple word embeddings which are described in BIBREF30 .

2pt