Appunti sul paper *Fien-grained Sentiment Classification using BERT*

A.A. 2023/2024

Rosso Carlo

Contents

1 Introduction

Sentiment classification is a form of text classification in which a piece of text has to be classified into one of the predefined sentiment classes. It is a supervised machine learning problem. In fine-grained sentiment classification, there are five classes: very negative, negative, neutral, positive, very positive (there is a nice pic to show this). While transfer learning (pretraining and finetuning) has become the de-facto standard in computer vision, NLP is yet to utilize this concept fully.

Recently Google reserchers published BERT (Bidirectional Encoder Representations from Transformers), a deep bidirectional language model based on the Transformer architecture.

In this paper, we use the pretrained BERT model and finetuen it for the fine-grained sentiment classification task on the Stanford Sentiment Treebank (SST-5) dataset.

2 Related Work

Sentiment classification is one of the most popular tasks in NLP.

The first step in sentiment classification of a text is the embedding, where a text is converted into a fixed-size vector. Since the number of words in the vocabulary after tokenization and stemming is limited, researchers first tackled the problem of learning word embeddings.

The next step is to combine a variable number of word vectors into a single fixed-size document vector. The trivial way is to take the sum or the average, but they don't lose the ordering information fo words and thus don't give good results.

All the approaches seen this far are context-free: they generate single word embedding for each word in the vocabulary. Recent language model research has been trying to train contextual embeddings.

Someone proposed BERT (Bidirectional Encoder Representations from Transformers), an attention-based Transformer architecture, to train deep bidirectional representations from unlabeled texts. Their architecture not only obrains state-of-the-art results on many NLP tasks, but allows a high degree of paralleism.

3 Model

Sentiment classification takes a natural language text as input and outputs a sentiment score $\in \{0, 1, 2, 3, 4\}$. Here we give a brief description of the pretrained BERT model and then we describe our model architecture.

3.1 BERT

BERT is an embedding layer designed to train deep bidirectional representation from unlabeled text by jointly conditioning on both left and right context in all layers. It is pretrained from a large unsupervised text corpus using the following objectives:

- *Masked word prediction*: 15% of the words in the input sequence are masked out, the entire sequence is fed to a deep bidirectional Transformer encoder, and then the model learns to predict the masked words.

- *Next sentence prediction*: to learn the relationship between sentences, BERT takes two sentences A and B as inputs and learns to classify whether B actually follows A or is it just a random sentence.

Unlike traditional sequential or recurrent models, the attention architecture processes the whole input sequence at once, enabling all input tokens to be processed in parallel. Pretrained BERT model can be fine-tuned with just one additional layer to obtain state-of-the-art results in a wide range of NLP tasks.

3.2 Preprocessing

BERT requires its input token sequence to have a certain format. So we perform the follwoing preprocessing steps on the review text before we feed them into our model:

1. *Canonicalization*: first, we remove all the digits, punctuation sumbols and acent marks, and convert everything to lowercase;
2. *Tokenization*: we then tokenize the text using the word-piece tokenizer. It breaks the words down to their prefix, root and suffix to handle unseen words better;
3. *Sepcial toekn addition*: finally we add the `[CLS]` and `[SEP]` tokens at the appropriate positions.

3.3 Architecture

We build a simple architecture with just a dropout regularization and a softmax classified layers on top of pretrained BERT layer to demonstrate that BERT can produce gread resulsta even without any sophisticated task-specific architecture. There are four main stages:

1. preprocessing: described above;
2. sequence embedding: from BERT;
3. regularization: dropout layer with a dropout rate of 0.1 to prevent overfitting;
4. softmax classification: it will output the probabilities of the input text belonging to each of the class labels such that the sum of the probabilities is 1. The softmax layer is just a fully connected neural network with the softmax activation function. The softmax function $\sigma : \mathbb{R}^K \to \mathbb{R}^K$ is given by:

   $$\sigma(x)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} for i = 1, \ldots, K \tag{3.1}$$

   where $z = (z_1, \ldots, z_K) \in \mathbb{R}^K$ is the intemediate output of the softmax layer (also called logits). The output node wiht the highest probability is then chosen as the predicted label for the input.