Review of Pre-Training Methods of Language Models

Wusi Chen (NetID: wusic2)

1. Introduction

Language model is a probability distribution over sequences of words. It is used to determine the probability of generating a text with the length of m. If the text is long, the estimation of $P(w_i \mid w_1, w_2, \ldots, w_{i-1})$ will be very complicated. In this case, researchers proposed a simplified model: n-gram model. To improve the data spareness when estimating the probability using n-gram model, Bengio et. al. [1] proposed neural language model. This kind of models use a 3-layer forward neural network for modeling. The first layer is found to be a good candidate for word representation, which is a foundation of using word vectors (like word2vec). The algorithm for training neural network is mainly based on the backpropagation algorithm, and the model parameters are initialized at random. The general idea of pre-training is to initialize the model parameters using a training task, instead of initializing them randomly.

2. ELMo

2.1 Intro

The paper of *Contextualized Word Representations* [2] was published by a team in Washington University in 2018's NAACL, and selected as the best conference paper. The authors believe a pre-trained word presentation shall not only model the complex characteristics of syntax and semantic information, but also be able to model the various meanings of a word across different contexts. However, the traditional word vectors (such as word2vec) are independent from the context, and therefore, it is unable to model the multiple meanings of words. The author leveraging language modeling to get pre-trained contextualized representation, called Embeddings from Language Models (ELMo). This model makes improved performance in 6 NLP tasks.

2.2 Method

In ELMo, the authors constructed a bidirectional LSTM language model, which consists of a forward language model and a backward language model. The objective function is to jointly maximize the log likelihood of the two models. After pre-training, the ELMo becomes the sum of the intermediate layers in the bidirectional model. In the simplest case, ELMo can be represented by the top layer.

In the supervised NLP tasks, Elmo can be used as a feature and concatenated to the word vector input of a specific task, or to the top-layer representation of model.

Unlike the traditional word vectors, in which each word is represented by a single vector, ELMo leverages the pre-trained bidirectional language models, and gets the contextual word representation from the language model based on a specific input, and finally adds the representation to an NLP supervised model as a feature.

3. Open AI GPT

3.1 Intro

This method was published in the paper of Improving Language Understanding by Generative Pre-Training by OpenAI team [3]. The authors set a goal to learn a universal representation that transfers with little adaptation to a wide range of tasks. The highlight of this paper is to use transformer networks as the language model to better capture long-term linguistic structure, and includes language models as auxiliary training objectives when fine tuning for specific tasks. The model significantly improves upon the state of art in 9 out of the 12 studied task.

3.2 Method

In unsupervised training, the objective function of a standard language model is used, which is to use the previous k words to predict the current word. However, in the language model network, the authors use the multi-layer transformer decoder as the language model, which is published in the paper of Attention is all your need by the Google team. In supervised fine-tuning, different from ELMo which is used as a feature, OpenAI GPT does not need to need to re-build the model for a specific task, but it concatenates a softmax layer as a task output layer to the last layer of the language model, and fine tunes the concatenated model. The authors additionally found that, including language modeling as an auxiliary objective can improve the generalization performance of the supervised model, and can accelerate the convergence.

Since different NLP tasks require different types of input, the input for the transformer model shall be different for different tasks. For text classification tasks, the text data can be directly used as the input. For text entailment tasks, the premised and hypothesis shall be concatenated by a Delim vector. For text similarity tasks, the 2 text vectors shall be concatenated by a Delim vector in the both directions. For multiple choice tasks, each context shall be concatenated to the corresponding answer.

4. BERT

4.1 Intro

In the paper of BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [4], the authors categorized the pre-trained language representations as the feature-based approaches (ELMo) and the fine-tuning-based approaches (OpenAI GPT), and both approaches use unidirectional language models to learn general language representations. The authors demonstrate the importance of bidirectional pre-training for language representations.

The overall framework of the method in this paper is similar to GPT. The difference is that BERT uses transformer encode as the language model, and it uses 2 new objectives: One is the masked language model (MLM), and the other is a “next sentence prediction” task that jointly pre-trains text-pair representations. BERT advances the state-of-the-art for 11 NLP tasks.

4.2 Method

In language model, BERT’s architecture is a multi-layer bidirectional transformer, which includes a smaller base architecture and a larger network architecture.

For the comparison of the architecture of the language models: BERT is a transformer encoder. Due to the self-attention mechanism. The upper and lower layers of the model are completely inter-connected. OpenAI GPT uses a transformer decoder which is a limited left-to right transformer. Although ELMo uses bidirectional LSTM, the 2 unidirectional LSTM are simply concatenated at their top layer.

For input representation, BERT uses WordPiece embedding as the token vectors, and adds position vectors and segment vectors. Besides, a CLS vector is added to the beginning of every text sequence.

In the pre-training of language models, BERT uses 2 new tasks, instead of using the objective of predicting the next word in the left-to-right sequence. The first task is called MLM. In this task, 15% of the input tokens is masked at random, and the task is to predict those mask tokens. Unlike the objective function in the traditional language models, MLM allows the prediction from the both left and the right context. However, it creates a mismatch between pre-training and fine-tuning since the masked tokens are never seen during fine tuning, and the convergence is slower since the prediction is for the 15% masked tokens instead of the whole sentence. To mitigate the mismatch, rather than always replacing the chosen words with mask, the data generator will replace the word with a random word in 10% of the time, and keep the word unchanged in another 10% of the time. The second task is next sentence prediction. This task is actually a 2-factor categorization problem. When choosing the sentence A and B for each pre-training example, B is the next sentence that follows A in 50% of the time, and it is a concatenation of a sentence and a random sentence from the corpus. The objective function of the whole pre-training is the maximum likelihood of these two mentioned tasks.

5. Conclusion

The language model pre-training is a contextual word representation at the sentence level. It can model polysemy by utilizing a large-scale monolingual corpus. After the pre-training on large-scale corpus, the models demonstrated in [3] and [4] can be used as a stand-alone model or added to other simple models, and they achieved a good performance on various NLP tasks. It also greatly reduces the dependency of actual tasks on model architecture. However, the discussed models have a high space complexity and a time complexity. Besides, the discussed methods involve a lot of details. The performance of these methods will be discounted if these details are not well handled.

Reference:

[1] Bengio, Y. et al. A Neural Probabilistic Language Model. (2003).
[2] Peters, M. E. et al. Deep contextualized word representations. NACCL (2018).
[3] Radford, A. & Salimans, T. Improving Language Understanding by Generative Pre-Training. (2018).
[4] Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. (2018).