Transfer Learning for Natural Language Processing

Chua Yeow Long, ylchua2@illinois.edu

With the recent advances in machine learning and birth of deep learning, neural network architectures have shown performance improvements in several Natural Language Processing (NLP) tasks such as text classification and machine translation and many others. Transfer learning has been instrumental in the recent success of deep learning in computer vision tasks. However, the performance of transfer learning for NLP pales in comparison with its computer vision counterparts mainly due to the lack of large labeled text datasets.

Transfer learning is a technique in machine learning where a model is first trained on a large dataset before being used to perform similar tasks on another dataset. This model is called a pre-trained model and the most renowned pre-trained models for computer vision is the ImageNet dataset. It is almost always recommended to use a pre-trained model as a starting point rather than building a model from scratch by oneself. Most of the NLP tasks such as text classification and language modelling are sequence modelling tasks and as such traditional machine learning techniques are unable to capture this sequential information present in text. Hence, neural network architectures such as RNN and LSTMs are used to model sequential information present in texts. However, these techniques pose a fair amount of its own problems. One issue for RNNs is that they can only take on input at a time and cannot be parallelised and as a result, training a huge dataset will take a long time. In 2018, one major milestone for transfer learning in NLP is the introduction of the transformer model by Google in the paper “Attention is All You Need”. There are two benefits of using transformer-based models. The first is these models take the entire sequence as input in one go which is a significant improvement over RNN based models because models now can be accelerated by GPUs. The second is we do not need labeled data to pre-train these models. What this means is that we just use the models to make predictions on unlabelled text data to train a transformer-based model. We can also re-use this model for other NLP tasks. BERT and GPT-2 are the two most popular transformer-based models.

Recently, in modern NLP, zero-shot learning has been used to get a model to do something that it was not explicitly trained to work on. A popular example would be the GPT-2 paper where authors evaluate a language model on downstream NLP tasks such as machine translation without fine-tuning or re-training the entire or part of the neural network architecture. This technique is pretty flexible and is easily adapted to scenarios where a limited amount of label data becomes available and this is known as few-shot learning where we have data for the classification tasks we are interested in.

To achieve zero-shot learning, we model the classification problem as a Natural Language Inference (NLI) problem. NLI considers two sentences: a “premise” and a “hypothesis”. The task at hand is to determine whether the hypothesis is true which means it entails the premise or false where there is a contradiction given the premise. When using transfer architectures like BERT, NLI datasets are modelled as sequence-pair classification and we feed both the premise and hypothesis through the model together as distinct segments and predict whether it is a contradiction or not. It is pretty easy to use a pre-trained MNLI sequence-pair classifier out-of-the-box which works pretty well. The idea is to take sentences which we are interested in and an example would be a sentence with positive sentiment as the “premise” and in turn each target label as a “hypothesis”. If the NLI model predicts that the premise “entails” the hypothesis, we take the label to be true. The follow code snippet below illustrates how to easily do this using Huggingface transformers.

```
from transformers import pipeline

classifier = pipeline("zero-shot-classification")
# classifier = pipeline("zero-shot-classification",
device=0) # to utilize GPU

sequence = "I hated this movie. The acting sucked."
candidate_labels = ["positive", "negative"]

classifier(sequence, candidate_labels)
{'labels': ['negative', 'positive'],
 'scores': [0.9916268587112427, 0.008373176679015 16],
 'sequence': 'I hated this movie. The acting sucked.’}
```

In this code snippet, we made use of the pre-trained MNLI zero shot classifier to perform sentiment analysis.

Natural language processing is a very exciting field right now and in recent years, the research community have figured out pretty effective methods of learning from the enormous amounts of unlabelled data in this field. The success of transfer learning or one-shot learning has allowed us to surpass pretty much most of all the existing benchmarks on downstream supervised NLP tasks. As we continue to explore new model architectures and different learning objectives, the state-of-the-art shifts and continues to be a moving area in NLP research where large amount of annotated data are readily available.

References

Language Models are Unsupervised Multitask Learners
Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever

Attention Is All You Need
Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin