5-2021

Neural Network With Nlp

Harshita Sharma

Tinkle Jain

NEURAL NETWORK WITH NLP

BY

HARSHITA SHARMA

TINKLE JAIN

MASTER OF SCIENCE

SCHOOL OF PROFESSIONAL STUDIES

CLARK UNIVERSITY

MAY 2021

SUBMITTED TO:

RICHARD AROIAN

ASSISTANT DIRECTOR FOR STEM PROGRAMS

ACKNOWLEDGEMENT

First and foremost, we would like to thank our adviser Richard Aroian for providing us with this priceless opportunity to work under his direction and for providing us with continual support, encouragement, ideas, and positivity. He not only assisted us with our thesis work, but also with honing and improving our approach to scientific challenges. We cannot thank him enough for taking time out of his busy schedule to assist us fulfill our research goals this year, despite the hurdles posed by the pandemic.

We would want to express our gratitude to the SPS department for providing us with all of the necessary resources.

Date: Harshita Sharma

5/24/2021 Tinkle Jain

TABLE OF CONTENT

ABSTRACT

This thesis is about neural networks and how their algorithmic systems work. Neural networks are well-suited to aiding people with complex challenges in real-world situations. Thesis topics include nonlinear and complicated interactions between inputs and outputs, as well as making inferences, discovering hidden links, patterns, and predictions, and modeling highly volatile data and variations to forecast uncommon events. Neural networks have the potential to help people make better decisions. NLP is a technique for analyzing, interpreting, and comprehending large amounts of text. We can no longer evaluate the text using traditional approaches due to the massive volumes of text data and the exceedingly unstructured data source, which is where NLP comes in. As a result, the research focuses on what a neural network is and how different types of neural networks are used in natural language processing. NLP (natural language processing) is a method for analyzing, interpreting, and comprehending vast amounts of text. Due to the huge volumes of text data and the extremely unstructured data source, we can no longer analyze the text using standard approaches, which is where NLP comes in. As a result, the study concentrates on what a neural network is and how various types of neural networks are used in natural language processing. Due to their exceptional success in numerous NLP tasks, BERT in particular has gotten a lot of attention. Google's Bidirectional Encoder Representations from Transformers (BERT) is a Transformer-based machine learning methodology for pre-training in natural language processing (NLP).

INTRODUCTION

1.a General Introduction of the Research Project

A neural network is a system of algorithms that attempts to identify underlying associations in a set of data using a method that mimics how the human brain works. Neural networks are also well-suited to assisting individuals in real-world environments with complicated problems. They can learn and model nonlinear and complex relationships between inputs and outputs; make inferences; discover hidden relationships, patterns, and predictions; and model highly volatile data and variances needed to predict unusual events. Therefore, neural networks have the potential to enhance decision-making processes. NLP is a tool for processing, interpreting, and comprehending vast quantities of text data. Due to the large volumes of text data as well as the extremely unstructured data source, we can no longer interpret the text using conventional methods, which is where NLP comes in. As a result, the study focusses on what a neural network is and how various forms of neural networks are used in NLP. BERT's key technical innovation is applying the bidirectional training of Transformer, a popular attention model, to language modelling. This is in contrast to previous efforts which looked at a text sequence either from left to right or combined left-to-right and right-to-left training. The paper's results show that a language model which is bidirectionally trained can have a deeper sense of language context and flow than single-direction language models. In the paper, there is a novel technique named Masked LM (MLM) which allows bidirectional training in models in which it was previously impossible.

In the field of computer vision, researchers have repeatedly shown the value of transfer learning — pre-training a neural network model on a known task, for instance ImageNet, and then performing fine-tuning — using the trained neural network as the basis of a new purpose-specific model. In recent years, researchers have been showing that a similar technique can be useful in many natural language tasks.

A different approach, which is also popular in NLP tasks and exemplified in the recent ELMo paper, is feature-based training. In this approach, a pre-trained neural network produces word embeddings which are then used as features in NLP models.

1.b Research Problem

BERT makes use of Transformer, an attention mechanism that learns contextual relations between words (or sub-words) in a text. In its vanilla form, Transformer includes two separate mechanisms — an encoder that reads the text input and a decoder that produces a prediction for the task. Since BERT's goal is to generate a language model, only the encoder mechanism is necessary. The detailed workings of Transformer are described in a paper by Google.

As opposed to directional models, which read the text input sequentially (left-to-right or right-to-left), the Transformer encoder reads the entire sequence of words at once. Therefore, it is considered bidirectional, though it would be more accurate to say that it's non-directional. This characteristic allows the model to learn the context of a word based on all of its surroundings (left and right of the word).

The chart below is a high-level description of the Transformer encoder. The input is a sequence of tokens, which are first embedded into vectors and then processed in the neural network. The output is a sequence of vectors of size H, in which each vector corresponds to an input token with the same index.

When training language models, there is a challenge of defining a prediction goal. Many models predict the next word in a sequence (e.g., "The child came home from ___"), a directional approach which inherently limits context learning.

1.c Rational for Research Project

Before feeding word sequences into BERT, 15% of the words in each sequence are replaced with a [MASK] token. The model then attempts to predict the original value of the masked words, based on the context provided by the other, non-masked, words in the sequence. In technical terms, the prediction of the output words requires:

1. Adding a classification layer on top of the encoder output.
2. Multiplying the output vectors by the embedding matrix, transforming them into the vocabulary dimension.
3. Calculating the probability of each word in the vocabulary with SoftMax.

The BERT loss function takes into consideration only the prediction of the masked values and ignores the prediction of the non-masked words. As a consequence, the model converges slower than directional models, a characteristic which is offset by its increased context awareness.

THEORY

2.a Brief Overview of Theoretical Foundations Utilized in the Research Study

BERT builds upon recent work in pre-training contextual representations — including Semi-supervised Sequence Learning, Generative Pre-Training, ELMo, and ULMFit. However, unlike these previous models, BERT is the first deeply bidirectional, unsupervised language representation, pre-trained using only a plain text corpus.

Why does this matter? Pre-trained representations can either be context-free or contextual, and contextual representations can further be unidirectional or bidirectional. Context-free models such as word2vec or GloVe generate a single word embedding representation for each word in the vocabulary. For example, the word "bank" would have the same context-free representation in "bank account" and "bank of the river." Contextual models instead generate a representation of each word that is based on the other words in the sentence. For example, in the sentence "I accessed the bank account," a unidirectional contextual model would represent "bank" based on "I accessed the" but not "account." However, BERT represents "bank" using both its previous and next context — "I accessed the ... account" — starting from the very bottom of a deep neural network, making it deeply bidirectional.

If bidirectionality is so powerful, why hasn't it been done before? To understand why, consider that unidirectional models are efficiently trained by predicting each word conditioned on the previous words in the sentence. However, it is not possible to train bidirectional models by simply conditioning each word on its previous and next words, since this would allow the word that's being predicted to indirectly "see itself" in a multi-layer model.

To solve this problem, we use the straightforward technique of masking out some of the words in the input and then condition each word bidirectionally to predict the masked words.

For example:

> **Input**: The man went to the [MASK]$_1$. He bought a [MASK]$_2$ of milk .
> **Labels:** [MASK]$_1$ = store; [MASK]$_2$ = gallon

While this idea has been around for a very long time, BERT is the first time it was successfully used to pre-train a deep neural network.

METHODS

3.a Study Method and Study Design

Using BERT for a specific task is relatively straightforward:

BERT can be used for a wide variety of language tasks, while only adding a small layer to the core model:

1. Classification tasks such as sentiment analysis are done similarly to Next Sentence classification, by adding a classification layer on top of the Transformer output for the [CLS] token.

2. In Question Answering tasks the software receives a question regarding a text sequence and is required to mark the answer in the sequence. Using BERT, a Q&A model can be trained by learning two extra vectors that mark the beginning and the end of the answer.

3. In Named Entity Recognition (NER), the software receives a text sequence and is required to mark the various types of entities (Person, Organization, Date, etc.) that appear in the text. Using BERT, a NER model can be trained by feeding the output vector of each token into a classification layer that predicts the NER label.

In the fine-tuning training, most hyper-parameters stay the same as in BERT training, and the paper gives specific guidance (Section 3.5) on the hyper-parameters that require tuning. The BERT team has used this technique to achieve state-of-the-art results on a wide variety of challenging natural language tasks. BERT (Bidirectional Encoder Representations from Transformers) is a big neural network architecture, with a huge number of parameters, that can range from 100 million to over 300 million. So, training a BERT model from scratch on a small dataset would result in overfitting.

So, it is better to use a pre-trained BERT model that was trained on a huge dataset, as a starting point. We can then further train the model on our relatively smaller dataset and this process is known as model fine-tuning.

Different Fine-Tuning Techniques

- Train the entire architecture – We can further train the entire pre-trained model on our dataset and feed the output to a SoftMax layer. In this case, the error is backpropagated through the entire architecture and the pre-trained weights of the model are updated based on the new dataset.
- Train some layers while freezing others – Another way to use a pre-trained model is to train it partially. What we can do is keep the weights of initial layers of the model frozen while we retrain only the higher layers. We can try and test as to how many layers to be frozen and how many to be trained.
- Freeze the entire architecture – We can even freeze all the layers of the model and attach a few neural network layers of our own and train this new model. Note that the weights of only the attached layers will be updated during model training.

3.b Explanation of Sample to Be Used in the Study

BERT is an acronym of *Bidirectional Encoder Representations from Transformers*. The term *bidirectional* means that the context of a word is given by both the words that follow it and by the words preceding it. This technique makes this algorithm **hard to train but very effective**. Exploring the surrounding text around words is computationally expensive but allows a deeper understanding of words and sentences.

Unidirectional context-oriented algorithm already exists. A neural network can be trained to predict which word will follow a sequence of given words, once trained on a huge dataset of sentences. However, predicting that word from both the previous and following words is not an easy task. The only way to do so effectively is to mask some words in a sentence and predict them too, e.g., the sentence "*the quick brown fox jumps over the lazy dog*" might be masked as "*the X brown fox jumps over the Y dog*" with label (*X* = *quick*, *Y* = *lazy*) to become a labelled record in a training set of sentences. One can easily derive a training set from a bundle of unsupervised texts by simply masking 15% of words (as BERT does) and training the neural network to deduce the missing words from the remaining ones.

Notice that **BERT is truly a deep learning algorithm**, while context-free algorithms such as *word2vec*, based on shallow recurrent networks, may not be. However, as such, BERT's training is very expensive, due to its transformer aspect. Training on a huge body of text – for example, all English-language Wikipedia pages – is a Herculean effort that requires decidedly nontrivial computational power.

As a result, BERT's creators **disentangled the training phase from the tuning phase** required to properly apply the algorithm to a specific task. The algorithm has to be trained once overall, and then fine-tuned specifically for each context.

FINDINGS

The GLUE benchmark includes the following datasets, the descriptions of which were originally summarized in:

MNLI Multi-Genre Natural Language Inference is a large-scale, crowdsourced entailment classification task. Given a pair of sentences, the goal is to predict whether the second sentence is an entailment, contradiction, or neutral with respect to the first one.

QQP Quora Question Pairs is a binary classification task where the goal is to determine if two questions asked on Quora are semantically equivalent.

QNLI Question Natural Language Inference is a version of the Stanford Question Answering Dataset which has been converted to a binary classification task. The positive examples are (question, sentence) pairs which do contain the correct answer, and the negative examples are (question, sentence) from the same paragraph which do not contain the answer.

SST-2 The Stanford Sentiment Treebank is a binary single-sentence classification task consisting of sentences extracted from movie reviews with human annotations of their sentiment.

The Corpus of Linguistic Acceptability is a binary single-sentence classification task, where the goal is to predict whether an English sentence is linguistically “acceptable” or not.

STS-B The Semantic Textual Similarity Bench- mark is a collection of sentence pairs drawn from news headlines and other sources. They were annotated with a score from 1 to 5 denoting how similar the two sentences are in terms of semantic meaning.

MRPC Microsoft Research Paraphrase Corpus consists of sentence pairs automatically extracted from online news sources, with human annotations for whether the sentences in the pair are semantically equivalent.

WNLI Winograd NLI is a small natural language inference dataset. The GLUE webpage notes that there are issues with the construction of this dataset, and every trained system that's been submitted to GLUE has performed worse than the 65.1 baseline accuracy of predicting the majority class. We therefore exclude this set to be fair to OpenAI GPT.

DISCUSSION

5.a Brief Overview of Material

All of the BERT results presented so far have used the fine-tuning approach, where a simple classification layer is added to the pre-trained model, and all parameters are jointly fine-tuned on a down- stream task. However, the feature-based approach, where fixed features are extracted from the pre- trained model, has certain advantages. First, not all tasks can be easily represented by a Trans- former encoder architecture, and therefore require a task-specific model architecture to be added. Second, there are major computational benefits to pre-compute an expensive representation of the training data once and then run many experiments with cheaper models on top of this representation.

In this section, we compare the two approaches by applying BERT to the CoNLL-2003 Named Entity Recognition (NER). In the input to BERT, we use a case-preserving Word Piece model, and we include the maximal document context provided by the data. Following standard practice, we formulate this as a tagging task but do not use a CRF layer in the output. We use the representation of the first sub-token as the input to the token-level classifier over the NER label set.

To ablate the fine-tuning approach, we apply the feature-based approach by extracting the activations from one or more layers without fine-tuning any parameters of BERT. These contextual embeddings are used as input to a randomly initialized two-layer 768-dimensional BiLSTM before the classification layer. $\text{BERT}_{\text{LARGE}}$ performs competitively with state-of-the-art methods. The best performing method concatenates the token representations from the top four hidden layers of the pre-trained Transformer, which is only 0.3 F1 behind fine-tuning the entire model. This demonstrates that BERT is effective for both fine- tuning and feature-based approaches.

5.b Post Analysis and Implications of Hypothesis and of Findings

Many important downstream tasks such as Question Answering (QA) and Natural Language Inference (NLI) are based on understanding the relationship between two sentences, which is not directly captured by language modeling. In order to train a model that understands sentence relationships, we pre-train for a binarized next sentence prediction task that can be trivially generated from any monolingual corpus. Pre-training data. The pre-training procedure largely follows the existing literature on language model pre-training. For the pre-training corpus we use the Books Corpus (800M words) and English Wikipedia (2,500M words). For Wikipedia we extract only the text passages and ignore lists, tables, and headers. It is critical to use a document-level corpus rather than a shuffled sentence-level corpus such as the Billion Word Benchmark in order to extract long contiguous sequences.

CONCLUSION

6.a Summary of Academic Study

BERT is undoubtedly a breakthrough in the use of Machine Learning for Natural Language Processing. The fact that it's approachable and allows fast fine-tuning will likely allow a wide range of practical applications in the future. In this summary, we attempted to describe the main ideas of the paper while not drowning in excessive technical details. For those wishing for a deeper dive, we highly recommend reading the full article and ancillary articles referenced in it.

6.b Reference to Literature Review

Training the language model in BERT is done by predicting 15% of the tokens in the input, that were randomly picked. These tokens are pre-processed as follows — 80% are replaced with a "[MASK]" token, 10% with a random word, and 10% use the original word. The intuition that led the authors to pick this approach is as follows (Thanks to Jacob Devlin from Google for the insight):

- If we used [MASK] 100% of the time the model wouldn't necessarily produce good token representations for non-masked words. The non-masked tokens were still used for context, but the model was optimized for predicting masked words.
- If we used [MASK] 90% of the time and random words 10% of the time, this would teach the model that the observed word is *never* correct.
- If we used [MASK] 90% of the time and kept the same word 10% of the time, then the model could just trivially copy the non-contextual embedding.

No ablation was done on the ratios of this approach, and it may have worked better with different ratios. In addition, the model performance wasn't tested with simply masking 100% of the selected tokens.

BIBLIOGRAPHY

7.a References

Liat Ein-Dor.(2020).Active Learning for BERT. https://www.aclweb.org/anthology/2020.emnlp-main.638.pdf

Christopher Thomas.(2019). Recurrent Neural Network.https://towardsdatascience.com/recurrent-neural-networks-and-natural-language-processing-73af640c2aa1#:~:text=Recurrent%20Neural%20Networks%20(RNNs)%20are,%2C%20audio%2C%20video%20among%20others.&text=Natural%20Language%20Processing%20(NLP)%20text%20generation

Michael J.(2018).Introduction to Natural Language Processing.https://becominghuman.ai/a-simple-introduction-to-natural-language-processing-ea66a1747b32#:~:text=Natural%20Language%20Processing%2C%20usually%20shortened,a%20manner%20that%20is%20valuable.

Rani Horev.(2018).State of the Art Language Model for NLP. https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

Jason Brownlee.(2017).What is Natural Language Processing.https://machinelearningmastery.com/natural-language-processing/