Grenze International Journal of Engineering and Technology, June Issue

Performance of Bidirectional Encoder Representations from Transformers using various metrics

Dokka Surya Teja[1] Arulmurugaselvi N[2] and Dr.B.Surendiran[3]
[1]Master's degree in National Institute of Technology, Puducherry
Email:suryatejadokka@gmail.com
[2]HOD, GPT, Tamilnadu
Email: arulms4u@gmail.com
[3]Associate Professor, National Institute of Technology, Puducherry.
Email: surendiran@gmail.com

***Abstract*— NLP or Natural Language Processing is a dramatically de- veloping field. In the "new ordinary" forced by COVID-19, a huge extent of instructive material, news, and conversations occur through computer- ized media stages. This gives more text information accessible to work on!**
Initially, straightforward RNNS (Recurrent Neural Networks) were uti- lized for preparing text classification. Yet, as of late, there have been many new examination distributions that give better cutting-edge out- comes. One such is BERT. In this paper, I'll put down my presentation of the BERT transformer utilizing different measurements.

***Index Terms*— Natural Language Processing · Recurrent Neural Networks text classification · Text classification · Bidirectional Encoder Representations from Transformers · Loss Functions.**

I. INTRODUCTION

A. Text classification

Text classification is a task in which we assign texts to the appropriate class. Before Machine Learning became popular, this activity was mostly done manu- ally by a group of annotators. That will become a problem in the future as the data grows larger, and it will take so much time just to do it. As a result, we should automate the task while also improving its accuracy.
Many recent breakthroughs in the field of data science and machine learning have made practitioners' jobs easier. BERT is a cutting-edge development in this industry that has accomplished numerous complex tasks with minimal effort. BERTs are preferred for complicated text-based modeling due to their ease of use and out performance.

B. what is BERT

BERT, which stands for Bidirectional Encoder Representations from Transform- ers, is a technique for pre-training deep bidirectional representations from unla- beled text by taking both the left and right context into account. This allows for the creation of top-performing models for a range of natural language processing tasks. Moreover, the pre-trained BERT model can be customized by adding just one extra output layer.
That sounds excessively intricate as a beginning stage. In any case, it sums up what BERT does pretty well so how

Grenze ID: 01.GIJET.10.2.199

about we separate it?
First and foremost, it is obvious that BERT represents Bidirectional Encoder Representations from Transformers. Each word on this document has a specific meaning, which we will discover as we read. Until further notice, the critical focus of this line is - BERT is dependent on Transformer engineering.
Another reason why BERT is so successful is that it is trained on a massive corpus of unlabeled text, which includes the entirety of Wikipedia (2.5 billion words!) and the Book Corpus (800 million words). This pre-training process plays a significant role in BERT's performance because the model gains a deeper and more nuanced understanding of how language works when it is trained on a large text corpus. This knowledge is incredibly useful for a wide range of NLP tasks. Additionally, BERT is considered to be highly bidirectional because it utilizes data from both the left and right sides of a given context during the training process. This means that the model has a more comprehensive understanding of how words and phrases relate to one another in a sentence.

C. Need for BERT

For the most part, language models read the information arrangement in one course: either left to right or right to left. This sort of one-directional preparation functions admirably when the point is to anticipate/create the following word.
However, to have a more profound feeling of language setting, BERT uti- lizes bidirectional preparation. Some of the time, it's additionally alluded to as "non-directional". Thus, it considers both the past and next tokens at the same time. BERT applies the bidirectional preparation of the Transformer to language displaying and learns the message portrayals.

II. DESCRIPTION

Before we can feed our data to a model, it must be translated into a format that the model can understand. As a result, we preprocess the data without any information related to sample ordering to affect the relationship between texts and labels. The data was then divided into testing and training segments. Typically, 80 percent of the samples are used for training and 20% for validation. Then, using the Bert preprocess and encoder methods, we deploy our pre-trained model. To address the overfitting issue, add a dropout layer that will randomly remove a few neurons from the network. The model is then compiled using the loss function optimizers, but for all trials, Adam was used as the optimizer. The model is then evaluated, and the findings are printed.

Loss functions

To optimize the performance of the model, the error for the current state of the model must be calculated multiple times. This involves selecting an error function, also known as a loss function, to measure the model's error so that adjustments can be made to reduce the loss in subsequent iterations.
Since neural network models learn how to map inputs to outputs through examples, the choice of loss function must align with the structure of the specific computational modeling problem, such as classification or regression. Furthermore, the output layer's structure must be appropriate for the chosen loss function.

A. Mean Squared Error Loss

If the intended variable's distribution is Gaussian, then the mean squared error (MSE) is the preferred loss function under the probabilistic inference paradigm. It is important to examine the choice of loss function before proceeding and only change it if there is a valid reason to do so.
The MSE loss function calculates the mean of the squared differences between the actual and predicted values. Regardless of the sign of the predicted and actual numbers, the result is always positive, and a perfect score is 0.0. By taking the square of the errors, the model is penalized more for larger errors than for smaller errors. This means that the model is incentivized to minimize the overall error, rather than just focusing on a few specific data points.

B. Mean Squared Logarithmic Error Loss

MSLE (Mean Squared Logarithmic Error Loss) is an abbreviation for Mean Squared Logarithmic Error Loss. When forecasting a large value, you may not want to penalize a model as severely as with mean squared error.
To mitigate the impact of large discrepancies in expected values, you can first take the natural log of each expected value, and then calculate the mean squared error. This is known as MSLE loss.
MSLE loss is particularly useful when a model directly predicts unscaled quantities. It provides a less severe penalty for errors involving large values, which can be a better fit for some modeling situations.

C. Mean Absolute Error Loss

For some regression problems, the target variable's distribution may be generally Gaussian, but it may contain outliers, such as very large or very small values that are far from the mean.

In such cases, the Mean Absolute Error (MAE) loss function is preferred because it is more robust to outliers. It is calculated by taking the average of the absolute differences between the actual and predicted values.

To switch to the MAE loss function, the model can be modified while keeping the same output unit configuration. This can be advantageous when the target variable contains outliers that can significantly impact the performance of the model if not handled appropriately.

D. Binary Cross-Entropy

Binary Cross-Entropy is the standard loss function for binary classification prob- lems, where the target variable has binary values of 0 and 1.

Mathematically, it is the preferred loss function under the expectation-maximization inference paradigm. It is important to evaluate the choice of loss function first and only change it if necessary.

When constructing a model for binary classification, 'binary cross-entropy' should be used to define the cross-entropy as the loss function. This loss function measures the difference between the predicted probability distribution and the true probability distribution of the binary target variable.

E. Hinge Loss

For binary classification problems, an alternative to the cross-entropy loss func- tion is the hinge loss function, which was specifically designed to be used with Support Vector Machine (SVM) models.

The hinge loss function is intended for binary classification problems, with objective values ranging from -1 to 1. It allocates more error when the polarity of the actual and predicted class labels differ, which helps to push samples towards the correct sign.

The performance of the hinge loss function can vary depending on the prob- lem, with some studies suggesting that it outperforms the cross-entropy loss function on binary classification tasks.

To use the hinge loss function in the model, it can be specified as 'hinge' in the compile function.

F. Squared Hinge Loss

SVM models are commonly used to explore alternative versions of the hinge loss function. One such variation is the squared hinge loss function, which is obtained by simply squaring the score hinge loss. This modification smooths the surface of the error function, making it mathematically easier to work with.

The squared hinge loss is often used when the hinge loss function does not perform well on a particular binary classification problem. Like the hinge loss function, the target variable must have values between -1 and 1.

To utilize the squared hinge loss function in the model, it can be specified in the compile() method when defining the model.

III. Experiment And Results

In order to compare the BERT loss functions we have done two experiments that are described as follows.

A. Email spam classification experiment

As our first experiment, we chose a typical classification problem, email spam classification, in which we must determine whether the email is spam or not. The dataset has two columns: one for category, which is divided into two types: spam (irrelevant data) and ham (relevant data), and another for message, which contains the original text of the email. The dataset is made up of 6k emails, 3k for training, and 3k for testing. We compared our dataset using several loss functions, and the findings are as follows:

TABLE I. EMAIL CLASSIFICATION

Loss function	Accuracy
Mean Squared Logarithmic Error Loss	96.70
Mean Absolute Error Loss	97.85
Binary Cross-Entropy	96.12
Hinge Loss	98.35
Squared Hinge Loss	98.13

B. IMDB movie review classification experiment

For our second experiment, we chose another classification problem, movie review classification, in which we must determine whether the movie review is positive or negative. The dataset has many columns but we consider only two columns: one is category, which is divided into two types: positive and negative, and review, which contains the original review of the movie. The dataset is made up of 50k reviews, 25k for training, and 25k for testing. We compared our dataset using several loss functions, and the findings are as follows:

IV. CONCLUSION

BERT is undeniably a breakthrough point in the application of machine learn- ing to natural language processing. Because it's easy to use and allows for quick fine-tuning, it'll most likely find a wide range of useful applications in the future.

TABLE II. MOVIE REVIEW CLASSIFICATION

Loss function	Accuracy
Mean Squared Logarithmic Error Loss	96.15
Mean Absolute Error Loss	97.35
Binary Cross-Entropy	95.75
Hinge Loss	98.17
Squared Hinge Loss	97.83

In this paper, we propose a comparison of BERT loss functions using email spam classification and IMDB Movie review classification has been performed and several loss functions have been evaluated. Although selecting a loss function is sometimes disregarded, it is critical to recognize that there is no one-size-fits- all solution, and selecting a loss function is just as crucial as selecting the correct machine learning model for the task at hand.

REFERENCES

[1] Shanshan Yu, Su Jindian, Da Luo, "Improving BERT-Based Text Classification With Auxiliary Sentence and Domain Knowledge", November 2019, IEEE Access PP(99):1-1 https://doi.org/10.1109/ACCESS.2019.2953990

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova," BERT: Pre-training of Deep Bidirectional Transformers for Language Understand- ing",arXiv:1810.04805v2 [cs.CL] 24 May 2019.

[3] Jing Fan, Xin Zhang, Sheng Zhang, Yan Pan, Lixiang Guo,"Can depth- adaptive BERT perform better on binary classification tasks", arXiv:2111.10951 https://doi.org/https://doi.org/10.48550/arXiv.2111.10951

[4] Yanrong Ji, Zhihan Zhou, Han Liu, Ramana V Davuluri,"DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in the genome ", Bioinformatics, Volume 37, Issue 15, 1 August 2021, Pages 2112–2120,

[5] C. Jefferson, H. Liu, and M. Cocea. "Fuzzy approach for sentiment analysis," in Proceedings of the 2017 IEEE International Conference on Fuzzy Systems (FUZZ- IEEE), Naples, Italy, July 2017.

[6] R. U. Mustafa, P. M. Saqib Nawaz, and F. Viger.,"Early detection of controversial Urdu speeches from social media," Data Sci. Pattern Recognit, vol. 1, no. 2, pp. 26–42, 2017.

[7] K. Kumar, B. S. Harish, and H. K. Darshan, "Sentiment analysis on IMDb movie reviews using hybrid feature extraction method," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 5, no. 5, p. 109, 2019.

[8] H. Xia, Y. Yang, X. Pan, Z. Zhang, and W. An, "Sentiment analysis for online reviews using conditional random fields and support vector machines," Electronic Commerce Research, vol. 20, no. 2, pp. 343–360, 2020.

[9] A. Krishna, V. Akhilesh, A. Aich, and C. Hegde, Sentiment Analysis of Restaurant Reviews Using Machine Learning Techniques, vol. 545, Springer, Singapore, 2019