2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS) | 979-8-3503-9737-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICACCS57279.2023.10112829

Robust Transfer Learning Based Modelling for Accelerating the Learning of Ai in the Field of NLP

S. Ravimaram
Department of Artificial Intelligence and Data Science
Saranathan college of Engineering
Tiruchirappalli, Tamilnadu
diprivi@gmail.com

Jerald Nirmal Kumar S
Department of Computer Science and Engineering,
Sharda school of Engineering and Technology,
Sharda University.
jeraldcse@gmail.com

A. Sathish,
Department of CSE,
Roever Engineering College,
Perambalur - 621212.
sathishroever05@gmail.com

S. Vatchala,
Department of School of Computer Science and Engineering (SCOPE),
VIT – Chennai Campus,
Vandalur – Kelambakkam Road,
Chennai – 600 127
vatchalacse@gmail.com

Ruchira Rawat
Department of Computer Science & Engineering
Graphic Era Deemed to be University
Dehradun, India
ruchira.rawat.cse@geu.ac.in

Michael Raj TF
School of Engineering,
SIMATS University, Chennai,
TamilNadu India
tfmichaelraj23@yahoo.com

***Abstract*—People have been able to communicate with one another via the languages that have developed naturally within their societies. It is estimated that somewhere over 6,500 distinct languages are currently being used in communication around the globe. Because language is the primary means by which humans communicate with one another, it is only natural that this fundamental concept has been investigated in detail in the scholarly literature for a considerable amount of time. This investigation has taken place over a considerable amount of time. In this paper, federated learning is used as an integral feature of the transfer learning model to accelerate the process of learning in NLP. The simulation shows a reduced mean square error in training, testing and validation stages**

Keywords—Federated learning, natural language processing, transfer learning, Artificial Intelligence.

I. INTRODUCTION

Deep learning is used to generate models for these vocations because they are so much more complex than anything that traditional machine learning algorithms are capable of handling. This is why deep learning is used to develop models for these occupations. In addition, deep learning is being utilized to assist these jobs. Due to the considerable costs involved with both the quantity of computer power and the size of the data, it is not always practical to train a deep learning model. This is because of the magnitude of the data. It is heartening to study alternate possible channels for the transfer of information between different deep learning models [1].

The goal of transfer learning is to improve the overall performance of a new model by making use of the parameters or information from a previously trained model that was used as a resource. This is accomplished through the process of using a model that has been utilized as a resource in the past. Depending on the labels that are associated with the datasets that are being utilised, transfer learning can either be classified as inductive or as transductive [2].

When it comes to recommendation systems, for instance, extra data is utilized in conjunction with transfer learning methodologies. Within the framework of the field of bioinformatics, the implementation of machine learning and data mining approaches provides a solution to the challenges posed by multitasking and transfer learning. In the field of natural language processing, transfer learning has emerged in recent years as a technology that has taken on an increasingly significant role [3].

The incorporation of transfer learning into natural language processing systems in a step-by-step fashion. They are primarily concerned with sequential fine tuning, which is the form of transfer learning that is employed most of the time. On the other hand, we are of the opinion that a more in-depth investigation of transfer learning in NLP is required, namely one that places all methods of transfer learning on a same footing and assigns an equal degree of significance to each technique [4].

Over the course of the past few years, developments in language models have made it possible for them to overtake their predecessors in several respects. After receiving training, these language models were able to produce useful information that was use in the process of resolving a range of problems relating to the processing of natural language.

II. NLP AND PRE-TRAINING

It should not come as a surprise that there has been a meteoric growth in the total number of model parameters in tandem with the meteoric surge in popularity of deep

learning. It is possible to prevent overfitting, and the model parameters can be trained with the substantially larger dataset. Despite this, most natural language processing applications are confronted with considerable difficulties when it comes to the production of large-scale labelled datasets due to the insurmountably high costs of annotation. This is especially important to keep in mind while applying for positions involving syntax and semantics [5].

Creating big datasets that are not labelled is a piece of cake when viewed in the proper context. Before we can make use of the enormous volumes of unlabeled text data that are at our disposal, we need to first produce an accurate representation of those data. Recent research has shown that extracting representations from PTMs on vast unannotated corpora can significantly increase performance on a wide variety of natural language processing tasks. This improvement in performance can be attributed to the PTMs.

A condensed description of the numerous advantages that come with pre-training is provided in the following paragraphs. In the first instance, doing some preliminary training on the vast text corpus can assist in the acquisition of universal language representations that will be helpful in the tasks that are to follow. Second, performing pre-training in advance makes it possible to initialize the model in the most effective way possible, which in turn improves the model generalization performance and speeds up its convergence on the target task. Third, performing post-training in the target environment improves the model convergence on the target task. (3) Pre-training, which may be thought of as a form of regularization, can be used to prevent overfitting on sparse data sets. This is one way that overfitting can be avoided [6].

In most cases, pre-training is the method of choice for learning the parameters of deep neural networks, which are then fine-tuned on subsequent tasks. This has been the case for quite some time. In deep learning came about in the form of greedy layer-wise unsupervised pre-training, which was subsequently followed by supervised fine-tuning. This was followed by the development of supervised deep learning [7].

Before moving on to undertake fine-tuning on more manageable data sets for specific tasks, it is common practice in CV to pre-train models on the huge ImageNet corpus. This is done before moving on to focus on specific tasks. This is preferable to starting the process at a random point because, as the model proceeds, it catches up on common aspects of images that may be utilized to treat a wide range of various kinds of vision disorders. Starting the process at a random point would be less efficient. It has also been discovered that PTMs carried out on big corpora are beneficial to downstream natural language processing applications. Word embeddings on the simpler end and more complex neural models on the other are both examples of these uses [8].

It is essential for the pre-training assignments to be difficult and to deliver a sizeable quantity of data for the training that will follow. This section provides a condensed overview of the pre-training activities, classifying them as supervised, unsupervised, or self-supervised activities respectively. (1) In supervised learning, also known as SL, input-output pairs are the training data that are utilized to discover a mapping function. These pairs are used to uncover the mapping function. (2) The goal of unsupervised learning, also known as UL, is to gain insight from data that has not been explicitly labelled by discovering patterns, structures, or correlations in the data. This is accomplished with computer algorithms. Several different approaches are possible for achieving this goal. (3) Secure Socket Layer (SSL) uses the best beneficial parts of supervised and unsupervised learning methodologies and combines them in a way that creates a more effective overall learning environment.

SSL uses the same learning paradigm as supervised learning, with the exception that it generates the labels for the training data on its own. In supervised learning, the labels are provided by an instructor. SSL generates the labels on its own. The primary purpose of Secure Sockets Layer (SSL) is to derive conclusions about one part of the input based on the information obtained concerning the various other parts of the input. The masked language model, often known as MLM, is an example of a self-supervised task that may be used to predict the masked words in a phrase given the other words [9].

III. Proposed Method

Even if there has been a sizeable increase in the overall number of images that have been trained, the size of the dataset that is now available is still not sufficient to train an entirely new deep model from the ground up. On top of the already-trained AlexNet architecture, we apply three separate implementations of transfer learning theory to correct this issue. The classification layer is changed to a SoftMax layer that either has two or three classes in the initial step of the process. After the weights have been fine-tuned in the step before, the next stage, which is called backpropagation, is carried out to train the new weights. The weights of the fully connected layers are seeded with random values at the beginning of the training process, and the learning rate is set low such that the weights of the convolutional layer do not change much over the course of the training process [10].

The network weights are kept up to date with the assistance of a federated learning (FL) algorithm that was trained using natural language processing (NLP) datasets. NLP datasets are used for natural language processing. At the very end, the datasets are enhanced by the addition of extra photos that are suitable for employing in the process of training deep neural networks. By utilizing this strategy, the ideal weights were located, and a sizeable increase in accuracy in classification was achieved by utilizing the SoftMax layer that had been adjusted. Both results may be attributed to the utilization of the modified SoftMax layer [11]

A. Federated Learning(FL)

The field of machine learning makes use of a method called FL, which is a distributed technique. To train a model with this approach, a huge number of users collaborate with one another rather than employing a

centralized storage facility. After that, the term FL will be used to refer to information that has been stored in a variety of organizations or geographical places all over the world. It possible that this information has a wide range of different qualities. During the evolution of FL models, it is conceivable to apply FL systems at any one of a number of different points along the process. The management of the distributed training process with dispersed assets is the major responsibility of a distributed FL system [12].

This is the work that the system is meant to do FL distinguishes out from other approaches to distributed machine learning due to the three key features that will be discussed in the following paragraphs. These differences are highlighted in the following paragraphs. To begin, in contrast to other approaches, FL does not provide provision for the possibility of the direct exchange of raw data. The principle of data minimization stipulate that the collection and storage of personally identifiable information must be restricted to only that which is required for the processing of the data and for which the consumer has provided their consent [13].

FL makes use of the distributed computing capabilities that are available in several locations or organizations. This contrasts with most techniques, which only make use of a single server or cluster that is located in a single location and is owned by a single enterprise. Because FL is facilitating the process, it is possible for several different parties to work together.

In contrast to the other techniques, which pay little attention to this security issue, FL routinely utilizes encryption in addition to other protection tactics to guarantee the data privacy and security. This contrasts with the other methods, which do not apply encryption. FL takes measures to protect the confidentiality and security of the raw data, the revelation of which could have substantial repercussions for the company both financially and in terms of its reputation.

The formula provides an illustration of the solution to an optimization problem that developed over the course of the training for FL. The D=D1,D2,...,Dn is used to indicate a training dataset, and the value of D might range anywhere from 1 to n.

Each iteration of FedAvg attempts to minimize the global model objective w, which is merely the weighted average of the local device losses. This is the goal of the iteration.

$$\min f(w)=\sum k=1 N p k F k(w) \quad (1)$$

where

Fk(w) - loss

Randomization is used to choose not only the customers but also the gadgets. Every client receives from the server an instance of the exact same global model. After performing stochastic gradient descent in parallel on their individual loss function, the clients will then submit the resulting model to the FL server, where it will be aggregated with the models from the other clients. This process will continue until all the models have been sent. The server computes the worldwide revision by first calculating the average of all these regional models and then utilizing that result as its starting point in the calculation of the global revision. Before arriving at a decision, this process is carried out an infinite number of times.

Both a forward and a backward propagation take place within each iteration of the pattern. Both processes are referred to as propagation. In the process of forward propagation, the model is used to decide the output based on the input data x; however, in the process of backward propagation, the model is updated by computing the gradients ∇Fk(x). This contrasts with the process of forward propagation, in which the model is used to decide the output based on the input data x. (x).

Whenever the calculation is distributed across many computers, an aggregation technique is carried out. Because of local adaptive optimizers such as Adam and cross-round learning rate schedulers, the learning rate has the capability of being updated in a dynamic manner.

Within each iteration, there are two phases, i.e., forward propagation and backward propagation. The forward propagation calculates the output based on the input data x using the model, while the backward propagation calculates the gradients

B. Fine Tuning

For the model to be successful in accomplishing this objective, it must first acquire the personal information of the patient. Because of this, the scenario represents a one-of-a-kind instance of the named entity identification problem. After merging token embeddings and character embeddings, the resulting data is then fed into an LSTM unit, which results in the creation of a transfer-learned model.

The output of this unit is transmitted into a fully linked layer to try at forecasting the positions of label elements. This is done to improve the accuracy of the forecast. After being formed by this fully connected layer, the label vectors for each token are then passed on to a sequence optimization layer, which is responsible for producing the label that is most likely to apply to each token. This occurs after the label vectors for each token have been sent from the fully connected layer that formed them.

Training a neural network often begins with the back-propagation method serving as the fundamental component of the process. An error rate from the previous epoch is often utilized as the input for the purpose of performing fine-tuning on the weights of a neural network (i.e., iteration). It may be possible to generate a forecast that is more accurate by first lowering the overall error rate and then, while doing so, broadening the scope of the model applicability by judiciously adjusting the weights of the variables.

To improve the overall functionality of the network, we make use of several different algorithms for optimization, such as stochastic gradient descent (SGD). Multiplying gamma by the partial derivative of the loss function L with respect to the weight W yields the most recent update to the weight, which may be found by clicking here. The outcome can be attributed to this. The speed at which you

may learn via gradient descent is directly proportional to the size of the iterative steps used in this method.

$$W=W-\gamma\partial L/\partial W$$
$$\partial L/\partial W\approx m^{-1}\sum_{i\in B}\partial l(i)/\partial W \quad (2)$$

It is vital to keep in mind that (2) does the SGD computation using a minibatch version of the GD formula. The gradient matrix is obtained by taking the average of the gradient matrices for all B batches, with m training samples included in each batch, as shown in Equation (3). This produces the gradient matrix. The calculation of the gradient matrix is the end outcome of this process.

The partial derivative is better to the full-batch generalized derivative, which uses all the training data. The partial derivative just uses a subset of the information. It is likely that GD will slow down training and the process of memorizing batches when the entire batch is being digested all at once. Back-propagation and a gradient error input, represented by the symbol e and expressed in the form e=LBy, are employed to generate the gradient matrix g. Back-propagation is used in conjunction with the gradient error input. To keep costs down, iterations of feed-forward and reverse training are carried out over a substantial number of epochs.

IV. Result And Discussions

The descriptions of the experiments that are contained in this part are provided in the following outline. It was determined how several factors, such as the number of initial convolution kernels, the size of the training set, and the size of the residual units, affected the accuracy of the classification. One of the criteria that is used to evaluate the experimental results is the degree of precision with which they can be classified in general.

The dot product of two vectors can be used to determine how closely they are related to one another in terms of their degree of similarity. If the calculation is finished, the total is then divided by a normalization factor, which is a value that is tied to the scale of the model.

We make use of the SoftMax function to perform the calculations necessary to determine the probability. It does this by appointing a probability score, which indicates the proportionate significance of each word in the keys to each word in the queries, to each sequence. This allows it to find matches more quickly. This score is calculated once each sequence is completed. Self-attention refers to the practise of paying a great deal of attention to the possibility that the set of queries and the set of keys are one and the same. This practice involves paying a great deal of attention to the possibility that the set of queries and the set of keys are one and the same.

Third, to apply weights to the values, we multiply the value matrix by the weights that we just determined. This completes the application of weights.

There are specific ones of these data sets that are always being worked on to make them better and enhance them even further. It is possible to say that the SQUAD problem has been solved because some trained models, for instance, are capable of outperforming people.

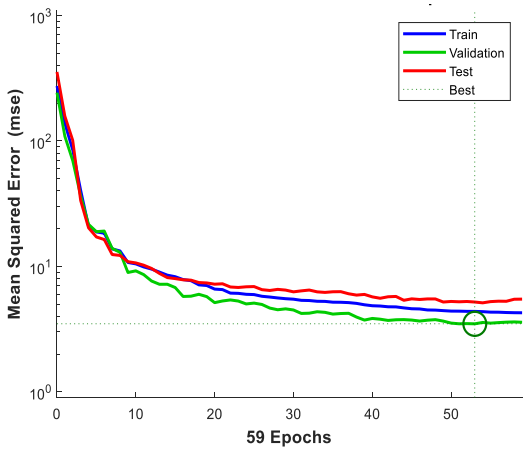

Fig. 1. MSE

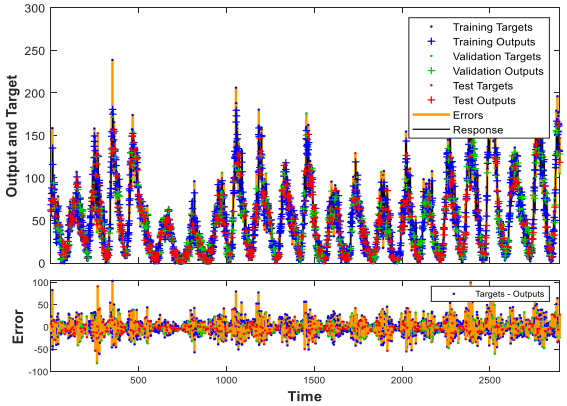

Fig. 2. Error Vs. Iteration Time

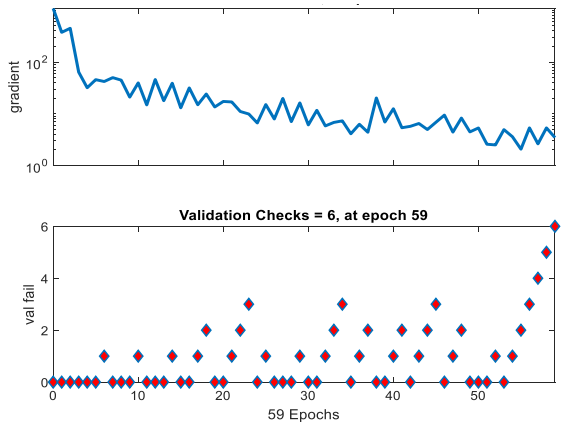

Fig. 3. Validation Performance

An example of a typical value for the mean squared error is presented in Fig. 1 and the Outcomes of the Validation is in Fig. 3. The correlation between the number of iterations and the total number of errors is illustrated in Fig. 2.

To estimate the amount of time necessary to train the model, which requires a significant amount of computational work, we consider the overall computational complexity of the training process. Memory consumption is a statistic that may be used to quantify the

amount of memory that is used up during the transmission and processing of data. Memory consumption measures how much memory is used up during these processes.

To assess how the framework impacted the recognition performance of the method, we combined numerous NLP data types and evaluated them depending on how well they performed. This allowed us to discover how the framework impacted the approach. A total of 45 features, including contextual data and other NLP data features, were gathered for the purpose of conducting the evaluation. Experiments were run with feature sets including anything from 4 to 45 parameters each to determine the effect that the aforementioned characteristics had on the identification performance of the techniques.

V. Conclusion

Since FL are dependent on their surroundings, these models are still necessary even after they have been used to construct vectors to train the models that are deployed to carry out the following tasks: Even though these models have been utilized before, this is still the case. For us to properly accomplish our categorization objective, FL will be updated to include an output layer soon. Even though it is possible to train any classifier model by using pre-trained versions of Word2vec to construct word embeddings for the text, it is still necessary to use FL after the synthesis of word embeddings to carry out classification tasks. This is because FL has a built-in classifier on the output layer, which means that even though it is possible to train any classifier model by using pre-trained versions of Word2vec, it is still possible to do so. FL is a more time-efficient alternative to other approaches, which require us to feed in the complete phrase as input. In contrast, other methods require us to feed in the entire phrase. On the other hand, FL only asks us to supply it with the local words to successfully extract the relevant context. This turns out to be an extremely useful piece of software whenever one is working with datasets that contain a great number of lengthy articles

References

[1] Azunre, P. (2021). Transfer learning for natural language processing. Simon and Schuster.

[2] Cai, C., Wang, S., Xu, Y., Zhang, W., Tang, K., Ouyang, Q., ... & Pei, J. (2020). Transfer learning for drug discovery. Journal of Medicinal Chemistry, 63(16), 8683-8694.

[3] Yao, L., Huang, H., Wang, K. W., Chen, S. H., & Xiong, Q. (2020). Fine-grained mechanical Chinese named entity recognition based on ALBERT-AttBiLSTM-CRF and transfer learning. Symmetry, 12(12), 1986.

[4] Lei, C., Dai, H., Yu, Z., & Li, R. (2020). A service recommendation algorithm with the transfer learning based matrix factorization to improve cloud security. Information Sciences, 513, 98-111.

[5] Liu, R., Shi, Y., Ji, C., & Jia, M. (2019). A survey of sentiment analysis based on transfer learning. IEEE Access, 7, 85401-85412.

[6] Chan, J. Y. L., Bea, K. T., Leow, S. M. H., Phoong, S. W., & Cheng, W. K. (2022). State of the art: a review of sentiment analysis based on sequential transfer learning. Artificial Intelligence Review, 1-32.

[7] Wiggins, W. F., & Tejani, A. S. (2022). On the opportunities and risks of foundation models for natural language processing in radiology. Radiology: Artificial Intelligence, 4(4), e220119.

[8] Alomari, A., Idris, N., Sabri, A. Q. M., & Alsmadi, I. (2022). Deep reinforcement and transfer learning for abstractive text summarization: A review. Computer Speech & Language, 71, 101276.

[9] Gruetzemacher, R., & Paradice, D. (2022). Deep Transfer Learning & Beyond: Transformer Language Models in Information Systems Research. ACM Computing Surveys (CSUR), 54(10s), 1-35.

[10] A. Saini, A. S. Kumar, S. J. N. Kumar and M. U, "Analysis And Implementation of a Novel AI-Based Hybrid Model for Detecting, Predicting and Identification Of COVID-19 Spread," 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2021, pp. 2059-2064, doi: 10.1109/ICAC3N53548.2021.9725778

[11] Naveen Kumar H N, A Suresh Kumar, Guru Prasad M S, Mohd Asif Shah, " Automatic Facial Expression Recognition Combining Texture and Shape Features from prominent facial regions", IET Image Processing, 2022, 1-15, https://doi.org/10.1049/ipr2.12700

[12] Abolfaz Mehbodiya, A. Suresh Kumar, Kantilal Pitambar Rane, Komal Kumar Bhatia, Bhupesh Kumar Singh, "Smartphone-Based mHealth and Internet of Things for Diabetes Control and Self-Management", Journal of Healthcare Engineering, vol.2021, Article ID 2116647, 10, https://doi.org/10.1155/2021/2116647.

[13] B. Sindhusaranya, V. Ellappan and A. Suresh kumar, "A survey on Separation of Blood Vessels for detecting Retinal Vascular Disorders," 2018 Conference on Emerging Devices and Smart Systems (ICEDSS), Tiruchengode, India, 2018, pp. 5-10, doi: 10.1109/ICEDSS.2018.8544328.