International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org
Volume 11, Issue 6, June 2022 ISSN 2319 – 4847

TPU Vision Accuracy Performance analysis of Optimized Deep learning models using TensorFlow

[1]T. Tritva J Kiran, [2]Dr. Pramod Pandurang Jadhav

[1]*Assistant Professor, CSE Dept, Dr. A.P.J. Abdul Kalam university, Indore, M.P., INDI.*

[2] Associate Professor, Computer Science & Engineering, Dr. A.P.J. Abdul Kalam University, Indore, INDIA

ABSTRACT

Computer vision is the state-of-art of understanding and manipulating images and videos. Disclosure of this work relates to achieve and compare more accurate optimization by measuring the performance analysis of accuracy in vision for classification and for the predictions on TPU using TensorFlow2.0 Keras with GPU. The ability to process large number of features makes Deep Learning models very powerful when dealing with unstructured data. Previous work presented the work extension of testing the optimization growth on vision accuracy of the deep learning models on TPU of the TensorFlow. This work presenting the comparative growth results of using GPU and TPU of TensorFlow. All the results clearly showing great difference between previous works tested on GPU and with TPU the optimized growth performance of vision accuracy of deep learning models using various difficult datasets with the effect of QoS on the TPU of the TensorFlow.

Keyword: TPU (Tensor Processing Unit), TensorFlow, Deep learning models, Optimization, Loss, Accuracy.

I. INTRODUCTION

Deep learning models vision is the knowledge of understanding or modeling images and videos. Deep learning models vision has a lot of usability in various fields like autonomous driving, industrial inspection, and greater than before veracity. With Deep learning vision we can get and apply for various types of classification, detection, segmentation, and generation problem solutions for both in images and videos performance QoS calculation results. This work train deep learning models for computer vision accuracy applications and deploy them on TensorFlow TPU platform [2][3][4][5][6][7]. We will use TensorFlow, a popular HPC GPU and TPU supportive platform developed on the top of Keras and python libraries for deep learning model architectures. The capacity to process huge quantity of features makes Deep Learning very dominant when dealing with unstructured data potentially. Graphical Processing Units corresponding with a well optimized implementation of 2D convolution, and are potent enough to make easy the training of interestingly large CNNs, and up to date datasets such as ImageNet be full of enough labeled samples to train such models without rigorous over fitting [1][2][3].

For the optimization deep learning NN widely uses Stochastic gradient-based algorithms. However, various problems are emerging when employing stochastic gradient-based algorithms [1]. Advancement in optimization algorithms is dependent on the improvement on convergence rate along with adding the SGD characteristics of its variant. Especially, switching an adaptive algorithm to the stochastic gradient descent method can improve the accuracy and convergence speed of the algorithm [1]. Reinforcement learning (RL) is a branch of ML, for which an agent interacts with the environment by trial-and-error mechanism and learns an optimal policy by maximizing cumulative rewards [1]. The recent research showing interest in reinforcement learning problems optimization with deep learning. Stochastic optimization algorithms are commonly used in RL and deep RL models [1]. Meta learning has recently become very popular in the field of machine learning. The goal of meta learning is to design a model that can efficiently adapt to the new environment with as few samples as possible [1]. Meta learning methods are the following 3 types metric- based methods, model-based methods and optimization-based methods [1].

The advancement of optimization contributing the improvement of ML. [1] However, there are still many challenges and open problems for optimization problems in machine learning, 1) How to improve optimization performance with insufficient data in deep neural networks is a tricky problem. A reason behind the problem of high variance and over fitting in the training of deep neural networks is not enough samples. Non-convex optimizations are the other complications in deep neural networks, leading the optimization towards a locally optimal solution than the global optimal solution. 2) And the sequential models, the samples are condensed by batches if the sequence is much long, leads to deviation and to examine the deviation we use stochastic optimization methods. 3) The stochastic variational inference is best to developing the methods of applying high-order gradient details. 4) Stochastic technique to the conjugate gradient model to get an well-designed and potent optimization algorithm. The detailed techniques to make improvements in the stochastic conjugate gradient is an interesting and challenging problem [1][2][3][4][5][6][7].

II. LITERATURE SURVEY

A neural network-based representation is frequently measured to be like a black box because it's tricky for humans to rationale out the operation of a deep learning model. The trainee and test set data images over layers by deep learning models are nonlinear due to activation functions, so visualization is difficult. In relaxed, the nearby confession provides systems and methods that power particular data increase schemes and a learnable nonlinear transformation among the image and the contrastive loss to give improved image representations intended to model frameworks which include a learnable nonlinear transformation between the version and the contrastive loss. Learnable nonlinear transformation between the image and the contrastive loss insignificantly improves the QoS of the learned representational deep learning model to avoid the effect on original input image with the loss calculation function and improve the accuracy in the representation output with the modeled network. This work presents a simple framework and its instantiation for contrastive visual representation learning for accuracy optimization within deep learning models network architectures. by means of combining these result, the projected systems and methods progress significantly over prior methods of AI Machine learning deep learning models.
So many researchers analyzed performance of Deep Learning models implemented using Python and in previous researchers work Linearity resulted less accuracy. To defeat those issues of accuracy results with datasets, here we used are difficult to deal and maintained with a lot of classes. The prediction and classification labeling from tons of images from more classes on low configured network architecture is really difficult. Trainee set image contains all features that the kinds of trippy effects with low resolutions gives you a class of sense of how close an eye is to the actual biological neurons that our human level intelligence. We can see unusual features in the concrete image. In This work I would like to achieve Optimization in terms of Vision Accuracy of the Deep Learning models using TensorFlow testing on TPU and would like to present all my simulations towards Optimized results of performance analysis for Vision Accuracy of Deep Learning Models with comparing to the results of GPU tested in the previous work [2][3][4][5][6][7].

III. DESIGN

[2][3][4][5][6][7] State-of-art of the system/method for accurate optimization in vision for classification and predictions for vision accuracy through the Deep Learning Models performance by using TensorFlow and keras on TPU comprises following steps are;

- Get the optimization in Non-Linearity of the Deep Learning model.
- To show the performance analysis of CNN (convolutional neural network) Deep Learning model.
- To achieve the Optimization in Computer Vision for Classification and Predictions of the Deep Learning model.
- For further Optimization used Deep Transform Learning and tested Vision Accuracy Performance.
- By using further optimization technique called Deep Representation Learning to achieve more vision accuracy QoS performance, also popularly known as Deep Auto Encoders.
- By using the Deep Inceptionism learning, popularly known as Deep Dream Algorithm to achieve the dream like effects on computer vision.
- To achieve great optimization in text predictions through long short-term memory (LSTM) recurrent neural network (RNN) Performance, popularly known as Deep Adaptive Learning.

Convolutional neural networks (CNNs) have weights, biases, and outputs through a nonlinear activation and they take test set or trainee set image inputs and the neurons fully connected to the next layers. Usual neural networks are very large in size due to a huge number of neurons, which is resulting over fitting. It is not suitable to use this for images, as images are large in size. Augment the model because it required handling a huge number of neurons. Each image can be measured with a volume of dimensions of height, width, and depth. Depth is the strait of an image, which is red, blue, and green. The neurons of a CNN are set in volumetric dimensions to take advantage of the capacity. Among each layer transforms the input volume to an output volume as shown in figure (1).

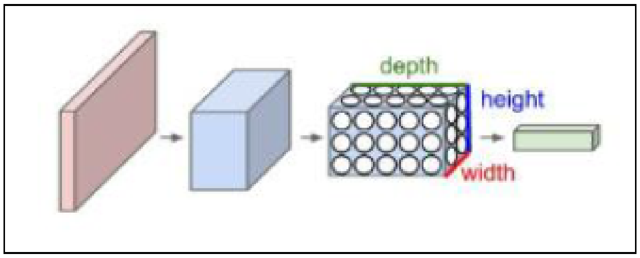

Figure (1)

The method of transfer learning is the way to learning from a pre-trained model that was trained on a larger dataset with random initialization often takes time and energy to get the result. Initializing the model with a pretrained model which will gives you faster divergence, save time and power. In this work deep learning models are pre-trained with carefully chosen hyperparameters. Either the several layers of the pre-trained model can be used without any modification, or can be bit trained to adapt to the changes and how to fine-tune or transfer learning for a model that was trained on the ImageNet dataset with millions of classes are well designed in the model. In the Deep representation learning, the features in the model are learned during training process. The resultant output features extracted during the training process in the hidden layers can be used to generate a distance metric. All these models learn how to detect or extract the mappings with edges, patterns, and so on at various layers, based on the classification and prediction process [2][3][4][5][6][7].

[5] Inceptionism learning was introduced the concept of inception that gives a better way of generalization. Also called deep dream learning, this architecture was won the ImageNet competition in 2014, which we used for this work. It is optimized greatly towards efficiency of speed and size. Inceptionism learning is the micro-architecture on which a macro-architecture is modeled. Every hidden layer has a deep higher-level representation of the input-output image. On every layer we apply pooling operation. Instead of using one type of kernel, Inceptionism learning uses numerous kernels. Regular pooling is followed by a variety of size convolutions and then they are concatenated. The neuron activations can be augmented at some layer in the network rather than synthesizing the image. This model of amplifying the novel image to see the effect of features is called Deep Dream learning. The steps for creating the Inceptionism learning algorithm are:
1. Feed the input to trained fully connected CNN.
2. Apply activation operation.
3. Adapt the gradient so that the gradient and activations should equal.
4. Calculate the gradients of the image and backpropagate.
5. The jittered image has to be normalized using regularization.
6. Abrupt the pixel values.
7. Use multi-scale operations on the image for the effect of fractal.

[6] Recurrent neural networks (RNN) are modeled to process sequential information with feedback architecture. They perform the same task from the output of the previous data of a series of sequence data as feedback propagations resembling the memory facility. RNN cannot remember from longer sequences or time because it is unfolded for the period of training the model. for the period of back propagation, the gradients can vanish under time. To beat this difficulty, long short-term memory can be used to memorize over a longer time epochs. Long short-term memory (LSTM) can store the sequence for longer periods of time, and for this reason, it is proficient in capturing long-term efficiencies. LSTM has numerous gates, they are forget, input, and output. Forget gate will do the task to maintain the information of previous state. By using the input, input gate updates the current state information. The output gate decides the sequence to be passed to the next state layer. The ability to forget and retain only the important things enables LSTM to memorize above longer time duration.

IV. IMPLEMENTATION

This work is designed to implement in Six (6) STEPS to achieve more Optimization within Vision Accuracy of Deep Learning Models using various types of complex, vast in classes of Datasets built and testing on TPU of the TensorFlow for the comparisons with previous work test results done using GPU of the TensorFlow. Below are the steps followed for this work,

STEP- 1: Optimization of Non-Linearity implementation
STEP- 2: Computer Vision Classification Accuracy with Deep Learning model

STEP- 3: Deep Transform Learning Vision Accuracy
STEP- 4: Deep Representation Learning QoS – Deep Auto Encoders
STEP- 5: Deep Inceptionism learning performance – Deep Dream Algorithm
STEP- 6: Text Predictions of LSTM RNN Performance– Deep Adaptive Learning

Tools and Libraries used in this work are, TensorFlow**,** Keras, matplotlib, Numpy, Pandas. First, mount the drive and installed TensorFlow 2.0, then loaded the corresponding step wise chosen dataset. Perform data visualization then use Matlock lib to perform Imus show and then can show the actual image along with its original label so it can show extreme along with white train of the index. And create grid and within that grid visualize the actual label of the image. First built our first layer convolutions fitted two filters each with corresponding step required size of the kernel. Perform activation function. Add additional convolution layers which can compatible with the kernel size and sufficient count for forming the correct architecture of the CNN network required to fulfil the step of built deep learning model. And apply activation value and then added the max pooling layer of two by two. And then add a drop out. And then flatten the network up and then added a dense network dense network of 1000 almost a thousand neurons. Apply soft max function on output. By repeating it we will get millions of parameters with respect to our built model. Run it on a TPU. Compile the model. Specify the optimizer as proposed as the optimization strategy presented in the previous work [1][2][3][4][5][6][7]. Specify the loss will be categorical cross entropy. Create a matrix consists of seven rows and seven columns and use subplots. Flatten it up. Then import confusion matrix. And then import seaboard as well because to use a heat map out of seaboard and then call confusion matrix pass along true classes along with predictive classes as described in previous work [2][3][4][5][6][7].

Now import libraries and also import data and here load step respective dataset then visualize again couple of data samples and covered it or we discover that the actual resolution of the images is very low. And then created grids or this matrix with all the images along with the labels and then afterwards normalize the data and build the model. It's the complete model and tested by running it on TensorFlow with TPU. The projection head neural network be a multi-layer Perceptron with one hidden layer to obtain $z_i=g(h_i)=W(2)\sigma(W(1)h)$ where σ is a ReLU non-linearity. it is beneficial to define the contrastive loss on final representations z_i's rather than intermediate representations h_i's. For unsupervised pretraining learning encoder network without labels are done using the ImageNet dataset. To evaluate the learned representations followed where a linear classifier is trained on top of the frozen base network, and test accuracy is used as a proxy for representation quality. Beyond linear evaluation, comparisons are also made against state-of-art on semi-supervised and transfer learning. Image datasets in both linear evaluation has fixed feature extractor, fine-tuning settings and Hyper parameter tuning was performed for each model-dataset combination and the best hyper parameters on a validation set were selected, Then apply a chance for learning and we know the network that has already been trained and freeze these layers and concatenate an additional classification head at the end and train these layers. Add dense, apply activation function of soft Max in the output. Pass the model along to batch it gives predictions. Compile. Use Adam optimizer and use loss which is categorical cross entropy. because we have more than two classes. Create loop. It will show two layers the first layer is the pre trained one and that's dense one which is the newly added layer that have been able to train. Afterwards add number of convolutions dense layers and finally formed fully connected CNN, then run on TPU, its setting the TPU hyper parameter on TensorFlow board.

V.RESULTS

In this session we presenting the simulation of deep learning model results achieved from step -1 to step- 6 as mentioned previous sessions to get more Optimization within Vision Accuracy of Deep Learning Models using various types of Datasets with the comparison of the previous work done on GPU to the comparison with TPU here using TensorFlow. The result of the last epoch value are showing below the loss value and corresponding accuracy value, as of observations and comparison with the previous work done on GPU and here with TPU the accuracy we achieved a great percent of improvement for the deep learning models. The loss ratio resulted here is very small leading us as symbol of optimization achievement on Tensor Processing unit with more accuracy on vision of the inputs and outputs.

```
- 41s 26ms/step - loss: 0.1828 - accuracy: 0.9387 - val_loss: 0.4057 - val_accuracy: 0.8804

/step - loss: 0.2168 - accuracy: 0.9837 - val_loss: 0.4550 - val_accuracy: 0.9680
```

Below figure (2) clearly depicting the results of confusion matrix, showing that the clear prediction results of TRUE and FALSE labelling counts of the dataset used around 10000 different classes of various types of images with low inaccurate resolution values.

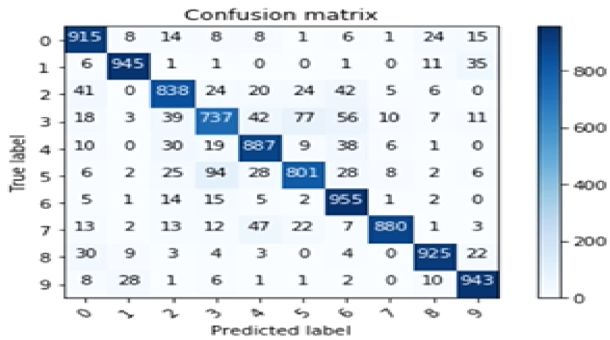

Figure (2)

Figure (3) presenting the growth from starting epoch to last 50th epoch Training set accuracy in comparison with loss performance. From the result, we can see clearly the Tensor Processing unit showed loss fluctuation values decreasing gradually though fluctuated high some times.

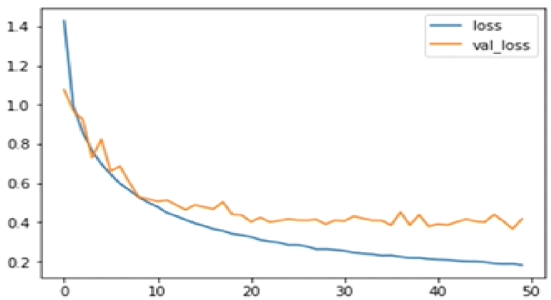

Figure (3)

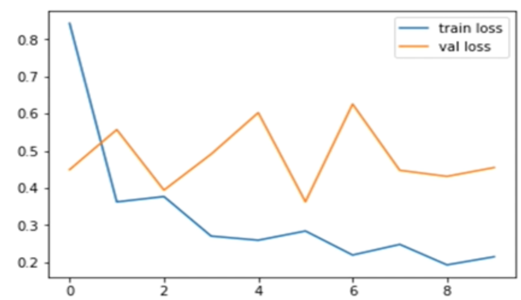

Figure (4)

Figure (4) presenting the growth from starting epoch to last 50th epoch Training set validations accuracy in comparison with loss performance. From the result, we can see clearly the Tensor Processing unit showed loss fluctuation values decreasing gradually though fluctuated high some times. Figure (5) presenting the growth from starting epoch to last 50th epoch Test set validations accuracy in comparison with loss performance. From the result, we can see clearly the Tensor Processing unit showed loss fluctuation values decreasing gradually though fluctuated high some times.

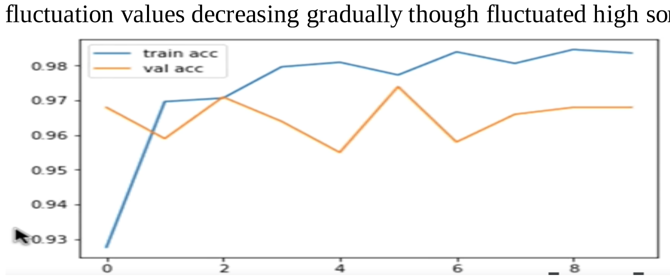

Figure (5)

The final validations results of accuracy growth rates ratios presented in Figure (6). The growth from starting epoch to last 50th epoch Test set validations accuracy in comparison with loss performance. From the result, we can see clearly the Tensor Processing unit showed loss fluctuation values decreasing gradually though fluctuated high some times.

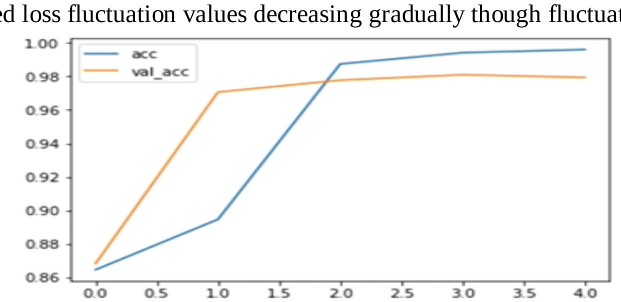

Figure (6)

VI. CONCLUSION

Disclosure relates to achieve more accurate optimization by measuring the performance analysis of accuracy for the vision on classification and predictions with GPU and TPU using TensorFlow. Though Deep Learning to achieve more vision accuracy QoS performance with the deep convolution neural network (DCNN) configured various neural network components of deep representation learning, deep auto encoder, deep transform learning, deep adaptive learning and deep Inceptionism learning used to achieve more vision accuracy QoS performance & configured for each layer of the DCNN during the classification of an image. it's a great thing the computer vision accuracy performed an excellent vision nearly 97% on TPU in this work.

Acknowledgement: This work is the extension proposed by corresponding author T. Tritva Jyothi Kiran with references [2][3][4][5][6][7][8] done on GPU, each step work for how the deep learning models growth of optimization in vision accuracy are published as separated papers and also as a journal [8].

REFERENCES

[1] Shiliang Sun, Zehui Cao, Han Zhu, and Jing Zhao, A Survey of Optimization Methods from a Machine Learning Perspective, orXiv, 2019.

[2] T. Tritva Jyothi Kiran, Analysis of Non-Linearity Accuracy for a Deep Learning model using GPU on TensorFlow, **Elsevier & Scopus** indexed IJAIEM, VOL 10, Issue 4, April 2021, ISSN 2319 – 4847

[3] T. Tritva Jyothi Kiran, Computer Vision Accuracy Analysis with Deep Learning using TensorFlow, IJIRCST, VOL. 8, Issue 4, July 2020, Page 319-325, ISSN 2347-5552, **DOI:** https://doi.org/10.21276/ijircst.2020.8.4.13

[4] T. Tritva Jyothi Kiran, Deep Transform Learning Vision Accuracy Analysis on GPU using TensorFlow, **Elsevier & Scopus** indexed IJRTE, VOL 9, Issue 3, September 2020, 224-227, ISSN 2277-3878, **DOI:**10.35940/ijrte.C4402.099320

[5] T. Tritva Jyothi Kiran, Deep Inceptionism learning performance analysis using TensorFlow with GPU – Deep Dream Algorithm, JETIR May 2021, Volume 8, Issue 5, ISSN-2349-5162, **DOI**: **http://doi.one/10.1729/Journal.26762**

[6] T. Tritva Jyothi Kiran, Text Predictions of LSTM RNN Performance Analysis using TensorFlow on GPU – Deep Adaptive Learning, JETIR International Journal, Volume 8, Issue 5, 06-05-2021, ISSN: 2349-5162, **DOI: http://doi.one/10.1729/Journal.26778**.

[7] T. Tritva Jyothi Kiran, Deep Representation Learning QoS on GPU using TensorFlow – Deep Auto Encoders, **ICCSEA 2021Conference –** Unique Paper ID: KU/ ICCSEA /August 2021/602.

[8] T. Tritva J Kiran, Computer Science – Deep learning Journal on Optimization in Vision Accuracy of Deep learning models with Tensorflow, IARDO RACE-2022/ June/P714 Scopus indexing conference proceeding.

[1]Author:

As of now, time being I am working as online coding trainer, I am Mrs. T. Tritva Jyothi Kiran with 10 years of work as Assistant Professor in Computer Science Department. Previously I have completed AICTE funding Project on IEEE802.11e in JNTUH and published in IEEE. I have been awarded Two times for the National Award for Excellence "Adarsh Vidya Saraswathi Rastriya Puraskar" from Glacier Global Management in 2020. And Received "Women Researcher" Award from 9th international conference organized by VDGOOD Professional Association. Extant I am working in the research domain Deep Learning using TensorFlow and Swarm Intelligence. You can find my Lectures during COVID in my Blog is tritvajyothikiran.blogspot.com and in Tritva Jyothi YouTube Channel. I have a roll position as REVIWER in IARDO and ASDF along with Silver membership. FSIESRP (Fellowship) & Editorial Board life membership and Life Member- IINF EBM PiscoMed Publishing Insight.

[2]Author:

I am Dr. Pramod Pandurang Jadhav, Professor in Dr. A.P.J. Abdul Kalam University, Indore, India. My research area of interest on software engineering Networking, IOT, Distributed system, security, Data science, Artificial Intelligence, Machine learning. I am a life time member of professional Body called ISTE (Indian Society for Technical Education). And awarded for Best Professional Award from ESN International Publication Chennai. I published various paper in Standard International Journal and Conference like Scopus, SCI etc, and guided various Under graduate, Post graduate Engineering student and PhD Candidate as a Research Guide. I have more than fourteen years of teaching experience in Pune University.