International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-9 Issue-3 September 2020

Deep Transform Learning Vision Accuracy Analysis on GPU using Tensor Flow

T. Tritva Jyothi Kiran

***Abstract:** Transfer learning is one of the most amazing concepts in machine learning and A.I. Transfer learning is completely unsupervised model. Transfer learning is a machine learning technique in which a network that has been trained to perform a specific task is being reused or repurposed as a starting point to perform another similar task. For this work I used ImageNet Dataset and MobileNet model to analyse Accuracy performance of my Deep Transform learning model on GPU of Intel® Core™ i3-7100U CPU using TensorFlow 2.0 Hub and Keras. ImageNet is an open source Large-Scale dataset of images consisting of 1000 classes and over 1.5 million images. And my overall idea is to analyse accuracy of Vision performance on the very poor network configuration. This work reached an Accuracy almost near to 100% on GPU of Intel® Core™ i3-7100U CPU which is great result with datasets used in this work are not easy to deal and having a lot of classes. That's why it's impacting the performance of the network. To classify and predict from tons of images from more classes on low configured network is really challenging one, it's a great thing the computer vision accuracy showed an excellent vision nearly 100% on GPU in my work.*

***Keywords:** Accuracy, Vision, TensorFlow, Transform Learning, Deep Learning, GPU, Dense layer, ImageNet Database.*

I. INTRODUCTION

Transfer learning is a machine learning technique in which intelligence or weights from a base or deficient network is being transferred to a new network as a starting point to perform a specific task. In standard transform learning, a dense basis is learned that analyses the image to generate the representation from the image [1]. Here, we learn a set of independent convolutional filters that operate on the images to produce representations each other [1]. A very important concept in machine learning and deep learning is Transfer learning. The idea of Transfer learning is a machine learning technique in which a network that has been trained to perform a specific task is being reused or repurposed as a starting point to perform another similar task, we wanted to transfer knowledge from a certain network that has been previously trained to perform a specific task and you take it and you transfer that knowledge somewhere else. By doing this you don't have to start from scratch. You don't have to go through all the learning through all the collection of the training data of the testing data. [1] In previous work they used Dictionary method for the testing. [1] [2]

Revised Manuscript Received on August 25, 2020.
* Correspondence Author
T. Tritva Jyothi Kiran, Assistant Professor, Department of Computer Science, AKNU, Rajahmundry, (A.P.), India. E-mail: tritvajkiran@gmail.com

The work is primarily focused on extracting domain invariant representations or learning a mapping from one domain to the other, in the previous work [2] they used Deep Transformer which leaded the work with good results.

In the work [4] they used MobileNet mobile architecture to improve the state-of-the-art performance of mobile models,

this work is based on an inverted residual structure convolution of non-linearity and showed good accuracy results. Based on previous work [3] this work concentrated on ImageNet Large Scale Visual Recognition [5] Challenge in object category classification and prediction on tons of object categories and millions of images. [4] Deep Transfer learning is the advance of learning in a new task through the transfer of information from a related task that has already been learned. Then, this trained network weights are then repurposed in a second ANN to be trained on a new dataset and function. Transfer learning works great if the features are general, such that trained weights can effectively repurposed. [5] [2] Intelligence is being transferred from the base network to the newly target network. The first CNN layers are used to extract high level general features. The last couple of layers are used to perform classification (on a specific task). So, we copy the first trained layers (base model) and then we add a new custom layer in the output to perform classification on a specific new task.

When to use transfer learning?

When there is a small training dataset available for your new task but there exists a large dataset in a different domain (such as ImageNet) and

When you have limited computational resources (GPU, TPU). In the next section I presented the concept of Transform learning and how it works.

II. CONCEPT OF TRANSFER LEARNING

Deep Transfer learning is extensively used since early from a pre trained models can vividly reduce the computational time required If training is performed from scratch. If you want to teach your network to classify images of cats and dogs you have two options either to start from a network that has been trained beforehand to classify let's say animals or let's say classify for example certain features that are pretty much similar to cats and dogs. That's one strategy. The other strategy is you know what I'm just go ahead and start from scratch. Just get a network and randomly initialize the weights. The problem with the second strategy is if you start from random weights it will take you forever to teach these networks because you're starting from scratch. It's like a baby and trying to teach him/her how to walk or how to say right. It will take him years of experience and accumulated knowledge to be able to do this.

However, what you could do which with the first strategy is that you can simply just get a network that has been trained beforehand to perform similar task and then you take it you tweak it a little bit by tweaking it and that will save you a massive amount of computational time and resources. And that's the power of what we call it transfer learning. In transfer learning we get base but what we call it a reference artificial, the network has been trained and we don't

get the entire network and we do all these trained network weights and then we try to repurpose them. The overall idea is the entire network has not been initialized from scratch. We actually got a base or reference artificial neural network beforehand so we have been able to basically instead of starting from scratch we got a head and started from something and that will save us a lot of computational time and sense for learning again works great if the features are general such that trend weights can effectively be repurposed again. Like a human that have been learned how to skate for example and now he's been able to know how to ski again. Use the same muscles in a way do the same kind of you know like balance in a way. So that's where you know the transfer of knowledge is relevant. In the next section I have Presented the procedure which we follow in the Transform learning.

III. TRANSFER LEARNING PROCESS

To transfer weights, we want to train convolution neural network on what we call it ImageNet. ImageNet dataset that contains millions and millions of images of almost a thousand classes just tons of classes.

Assume that we have a network convolution a neural network and this convolution network has been trained on ImageNet then upload convolutions form deep CNN, do Max pooling which compressing feature maps then apply network flatten on all featured maps and we needed to fully connected artificial neural network. Here fully connected dense layer convolution and network have been trained on ImageNet.

When it comes to transfer learning process, freeze these weights and take this network. This part of the network copies and going to transfer all these layers with the weights for example say 10 million weights copied the exact same weights with the exact same numbers which means it's the kind of transferring like knowledge from someone that has been trained already and you're transferring years and years of experience you're just putting it directly in here. And here uses optimizer which is gradient descent or back propagation to basically to find the optimal values of weights. This part is newly trained and take the entire network which is fully connected dense layers. These are mainly used to perform classification on a specific task. ImageNet for example has a thousand classes the output you have thousand neurons here.

All these first layers or initial layers can be used to extract general features and then the last layers can be used to perform classification on a specific task.

IV. STRATEGIES AND ADVANCES

Two strategies we use to perform transfer learning and they are commonly used throughout the industry.

The first strategy which is freeze the trend see and network weights from the first layers and only train the newly added dense layers which is basically created by adding randomly initialize weights. The second strategy is also copying the exact same network and initialize the CNN network with the pre train weights. But the point is we're going to set the learning rate to be very small. These are kind of the two strategies that we could use when we perform transfer learning and they are powerful and interesting. Transfer learning advantages are as follows, first it can provide fast training progress and another advantage, you can use small training data set to achieve incredible results. The second point is we need limited computational resources.

V. IMAGE-NET

ImageNet is an open source repository of images that consist of thousand classes and over one point five million images. It's just a huge dataset that is basically open source and readily available. And a lot of competitions. ImageNet Server is under maintenance. And you can go to the tree map visualization it can show you all the classes. We can simply capture all that data all these images in one place and train like a massive network with millions of weights to claim to perform all these classifications.

VI. IMPLEMENTATION

This project used trained artificial neural network known as MobileNet that can be readily trained ready in TensorFlow 2.0 hub and there are tons of networks that are readily available there is resonate or residual net there is Inception net and there is MobileNet which is simply one of the networks that are extremely efficient and can be able to run and much faster compared to other networks. To this research I used tons of flowers data set here. We call the flowers data set. Import all libraries panda's, Matlock and seaboard. Then Import here sensors. First, install TensorFlow hub. Think of it as a repository of all the weights of all the train networks and models and can able to download them. Fetch it from this model the input and run that. Download whatever image you want and test you the classification accuracy of MobileNet architecture afterwards normalize our data. Now ready to apply predict method on MobileNet which is trained MobileNet. I have to convert my image into a batch format. And that will give me my predictive class. Now we are running the model. We have various images of flowers data set and they belong to five classes. The overall idea is that because the network performance here is very poor freeze these layers. Then we apply a chance for learning and we know the network that has already been trained and freeze these layers and concatenate or add an additional classification head at the end and train these layers. Now add dense, and that's dense layer here. And now use an activation function of soft Max in the output. Then here going take my model pass along batch that would give me predictions. And again, that's the prediction shape. Now compile the model. Use my Adam optimizer and use loss which is going to be my categorical cross entropy. And the metrics is going to be accuracy. Note that here we used categorical or category categorical cross entropy because we have more than two classes. Here created loop. It will show me that I have two layers the first layer which is the pre trained one and that's my dense one which is the newly added layer that I have been able to train.

Afterwards added 50 convolutions dense layers and finally formed fully connected CNN, then run on GPU. In the next section I have Presented the results for this model.

VII. RESULTS AND COMPARISIONS

In this model, I have used the dataset ImageNet Flowers dataset and applied over 6 million parameters using sequential method on them using MobileNet model. In the result I got output which showing the number of classes with number of dense layers including the categorical parameters which is shown in below Output Figure (3).

In the simulation starting epoch the model shown 71% accuracy but on go, in the end of last epoch model showed great accuracy 99.89% which is almost 100% accuracy the model got on ImageNet data set, which we can clearly see in the Figure (1) given above indicates the Loss and Accuracy comparison result.

And all the classes are classified and Predicted by the model correctly without error, which we can see in the above Figure (2), result of the model indicated correct predictions of classes in green label name and if any wrong prediction for class classification happened then that will indicate in red label name. But this Deep Transform learning model predicted and classified with 99.89% accuracy, which is nearly 100% accuracy without errors and with thousands of dense layers and with millions of parameters with almost above 50 epochs the result of accuracy for the predictions and classification for transform learning is very pretty good result. This is indicating the computer vision accuracy is pretty good than human vision accuracy over a low configured network. In the next section I am concluding this work with future scope of this work.

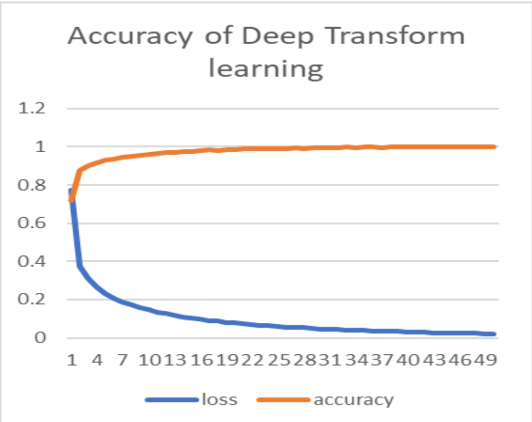

Figure (1): Accuracy vs Loss

Figure (2): Classification & Prediction Output.

OUTPUT

"sequential_1"

Layer (type)	Output Shape	Param #
keras_layer_1 (KerasLayer)	(None, 1280)	2257984
dense (Dense)	(None, 5)	6405

Total params: 2,264,389
Trainable params: 6,405
Non-trainable params: 2,257,984

Figure (3): Dense layers & Parameters Count

VIII. CONCLUSION

Deep Transfer learning is a machine learning method in which a network that has been trained to perform a specific task is being reused or repurposed as a starting point for another similar task. Deep Transfer learning is extensively used since preliminary from a pre-trained model can dramatically reduce the computational time required if training is performed from scratch. The Deep Transform model used in this work reached almost 100% Accuracy in Vision of Classification and Prediction with million number of Parameters, vast number of Epochs, thousand number of Dense layers of convolutional architecture on the Large-Scale dataset ImageNet with low configured network. The results are very Encouraging on Intel® Core™ i3-7100U CPU and GPU.

FUTURE SCOPE OF WORK

This work is tested for performance of computer vision accuracy for Classification and Predictions on Intel® Core™ i3-7100U CPU and GPU and got improvement 99.89%. I also have a plan to extend this work to test on TPUs and other different HPC hardware accelerated machines. And also, I have a plan to extend the model simulation by increasing the classes of images with various datasets again with vast number of Dense layers and vast number of Parameters.

REFERENCES

1. Jyothi Maggu, Emilie Chouzenoux, Giovanni Chiercha and Angshul Majumdar, Convolutional Transform learning, ICONIP, pp.162-174, 2018.
2. Shruti Nagpal, Maneet Singh, Richa Singh, Mayank Vatsa, Face Sketch Matching via Coupled Deep Transform Learning, ICCV, 2017.
3. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, ImageNet Large Scale Visual Recognition Challenge, cs.CV, 2015.
4. Mark Sandler, Andrew Howard, Menglong Zhu, Andey Zhmoginov, MobileNetV2: Inverted Residuals and Linear Bottlenecks, arXiv, 2019.
5. T. Chugh, M. Singh, S. Nagpal, Transfer Learning based evolutionary algorithm for composite face sketch recognition. In IEEE Conference, 2017.
6. Emilio Soria Olivas, Jose David Martin Guerreo, Marcelino Martinez Sober, Hand Book of Research on Machine Learning Applications and Trends.
7. Mobilenetv2 source code. Available from
8. https://github.com/tensorflow/models/tree/master/research/slim/
9. nets/mobilenet.
10. ImageNet dataset open source available from: http://www.image-net.org/

AUTHORS PROFILE

I am Mrs. Tritva Jyothi Kiran with 8 years of work as Assistant Professor in Computer Science Department. Previously I have completed AICTE funding Project on IEEE802.11e in JNTUH. 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 VDGOOD Professional Association. Present I am working in the research domain Deep Learning using TensorFlow. You can find my Lectures during COVID in my Blog is tritvajyothikiran.blogspot.com.