Compressing Neural Nets

Dasari Rishikesh[#], Matangi Sravan[*], Sompalli Ajay Kumar, MSVSS Hanuma[#]

[#]*Computer Science and Engineering, Indian Institute of Information Technology Dharwad*
Address

[1]19bcs035@iiitdwd.ac.in

[2]19bcs068@iiitdwd.ac.in

[3]19bcs101@iiitdwd.ac.in

[4]19bec023@iiitdwd.ac.in

***Abstract*— Deep Learning is advanced for usage in wide range of applications including Image classification, speech recognition, and text recognition. In order to obtain accuracies neural networks became larger. Its hard to embed them on the low computation devices like mobiles, drones and ARs. They are computationally expensive and memory intensive. To embed these Networks on the low computation systems Network computation is came into existence. Several models were developed to the size of the models. In this project we focussed on Convolution Neural Network and used knowledge distillation on the Residual Network models and the compressed the same models of their families with different datasets.**

***Keywords*— Deep Neural Networks, Low computation devices, Knowledge Distillation, Network compression, Residual network**

I. INTRODUCTION

In recent years the Deep Neural Networks has grown with huge popularity in computer vision tasks as they are used in different applications and the result accuracies has improved the tasks. Trying to chase the accuracies made them large. Many are them are over the order of millions too. Using them in real time applications made difficult as they require high computation and more memory. They use high end GPUs and powerful processors for processing. Low computational devices like mobiles, drones, and UAV (unmanned ariel vehicles) are must needed to embed the neural nets for better privacy, less network bandwidth and better privacy. Compressing the neural nets makes them smaller and able to embed them on the low computation devices. There are different compression models. Some of them are categorized as parameter pruning and quantisation, low rank approximation, knowledge distillation. In this project we look forward into knowledge distillation of Convolution Neural networks and Residual network because of complex architecture. [4] ResNet needs costly hardware such as high-powered GPUs to reach real time performances and architecture with dimensionality dependencies. Therefore, importance of compression came into factor. Further sections describe the progress of the project.

II. BACKGROUND

[1] The Deep compression model has obtained great results with three compression models with reduction of 36x to 49x of AlexNet and VGG CNN architectures. But growing with complex architectures the models may get less accurate. The concept of knowledge distillation came up by Caruana [3] for deep network compression in which the driven model learned from large model. [4] deals with selective layer pruning and knowledge distillation of other layer for avoiding the network structure break. The paper by Lucas Bayer [2] focussed on the issue of frequent used models and most download counts from the TensorFlow Hub i.e., ResNet-50 model, as a result many improvements in the computer vision has not applied in real world. It went through the ResNet upon knowledge distillation as model pruning arise reduces them by stripping away their parts.

III. METHODOLOGY

In this project, we used parameter pruning and quantisation at our initial stages on smaller neural networks and opted the results with the aim of achieving towards the Deep Compression [1]. Being usage of pruning restricted the experimentation with architecture challenges and unchanged in model family. Encountering such issues our next progress turned to usage of the recent introduced and mostly explored model i.e., knowledge distillation. Primarily we experimented with CNN model on MINST and CIFAR-10 datasets.

Further progress went on with the usage of knowledge distillation using function matching hypothesis.

A. Datasets

The dataset that implemented with the compression models are MNIST containing handwritten examples with 60,000 training datasets and 10,000 test sets, CIFAR 10 dataset has 80 million images and consists 60,000 images with 32×32 color images with 10 labels and 6000 images per class. In Addition, Flower 102 dataset has 102 categories containing the images of flowers, each class between the 40 to 258 images and this is experimented as because of vary in classes and make the distilled model for practical usage and assurance of validity. The dataset is further on the with mixup augmentation in inception-style for the training under function matching hypothesis.

B. Knowledge Distillation

Knowledge distillation is the idea of distilling the knowledge the larger model (i.e., teacher model) into the smaller models (i.e., student model). In this technique the knowledge is transferred from teacher model to the student model. It means making the student to match the outputs prediction values of the teacher. The knowledge distillation reduces the distillation loss (loss of soft targets) and matches the logits. The logits are applied with the temperature T in Softmax function for effective smoothing out the probability distribution.

$$q_i = \exp(z_i/T) / \sum_j \exp(z_j/T)$$

The q_i is the outvalues from the softmax function. As growth of T values more knowledge will be extracted from the teacher model. With high T values the model will be overfitted. The hard predictions are the are the labels where the highest student prediction taken as 1 and rest remain as 0s. Cross entropy calculated for soft targets and hard targets.

Loss of soft targets = Cross entropy (softmax outputs of teacher, softmax outputs of student)

Loss of hard targets = Cross entropy (student hard prediction, hard labels)

Where, Cross entropy = $-1/N \sum_j y_j * \log(\hat{y}_j)$

$y_j * \log(\hat{y}_j)$ = true label*log(predicted)
It's the sum over all sequences in each batch to the number of samples.

Loss=α*(Loss of soft targets) + (1- α) (Loss of hard targets)

α is the weight. The final loss is the weighted average of the loses of both the soft targets and hard targets. We need to reduce them as minimum as possible to match the student model prediction that of the teachers.

Function Matching:

The function matching hypothesis deals with the following key takeaways:
During distillation no ground truth tables are considered. Teacher and student models should go through the same augmentation or same crop of input images and the images are mixed up with an inception style augmentation. The images are trained to match the intersection points between the images and label them. The long training schedules for distillation are most recommended from the authors for getting more key accuracies.

[2] $KL(p_t||p_s) = \sum_{i \in C} [-p_{t,i} \log p_{s,i} + p_{t,i} \log p_{t,i}]$

Where C is the class sets and the distillation loss is considered as the KL divergence of predicted class probabilities vectors of teacher's p_t and student's p_s.

C. Experimentation

The initial stage of project aimed with the pruning and quantisation models. We constructed a small neural network that consists of six layers and pruned the network with 5x sparsity. The models give accuracy with minimal reduction. But the pruning doesn't work for large models as the complexity increases with pruning of more networks which loses the information. Then progressed with the knowledge distillation of teacher CNN (layers 256, 512) and student model CNN (16, 32) with datasets MNIST and CIFAR -10 gained the nearest accuracies to the teacher models. On further references of latest studies the function matching hypothesis is introduced and experimented with same dataset Flower-102 but change in the teacher model i.e., ResNet101 architecture. The dataset is converted to the mixed-up augmented data labels, then trained on the same augmented data for the knowledge transfer from the teacher to student and minimizes the distillation loss between the teacher and student.

D. Tables and Results

The Table-1 shows the results CNN with knowledge distillation and Table-2 ResNet with function matching hypothesis.

TABLE I

Datasets	Models	Architecture	Accuracy (sparse categorical accuracy)	Epochs
MNIST	CNN(Teacher)	Layers (256,512)	97.78%	25
MNIST	CNN(Student)	Layers (16,32)	97.50%	25
CIFAR-10	CNN(Teacher)	Layers (256,512)	69.72%	10
CIFAR-10	CNN(Student)	Layers (16,32)	58.55%	10

TABLE II

Dataset	Models	Accuracy (Top-1 Accuracy)
Flowers -102	BiT ResNet 101*3(Teacher)	98.18%
Flowers -102	BiT ResNet 50*1 (Student) for 1000 epochs	81.02%

IV. Conclusions

The project went through usage of the different models and different network architectures and aimed to compress them. Primarily the existed pruning has shown the good results but on observed that knowledge distillation on the large networks architectures gave more better results than the other models as they don't give good results for the large architectures. The analysis with application of the knowledge distillation makes the results accurate nearest to the teacher. On results we believe the model is the better compression in practical view as well as has a wide range of applications for the future research in the area of the compression.

References

[1] Song Han, Huizi Mao, William J. Dally, "*Deep Compression: Compression Deep Neural Networks with pruning, Trained quantization and Huffman coding",* Conference at ICLR 2016

[2] Lucas Beyer, Xiaohua Zhai, Amelie Royer, Larisa Markeeva*, "Knowledge distillation: A good teacher is patient and consistent",* CVPR 2022.

[3] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 2014, pp. 2654-2662.

[4] Nima Aghli, Eraldo Ribeiro*, "Combining Weight Pruning and knowledge Distillation for CNN compression",* CVPR 2021.

[5] Yu Cheng, Duo Wang, Pan Zhou, *"A survey of Model Compression and Acceleration for Deep Neural Networks",* IEEE signal processing Magazine.