Comparative Study of Model Optimization Techniques in Fine-Tuned CNN Models

Ramaprasad Poojary
School of Engineering & IT
Manipal Academy of Higher Education,
Dubai Campus
Dubai, UAE
rama.poojary@manipaldubai.com

Akul Pai
School of Engineering & IT
Manipal Academy of Higher Education,
Dubai Campus
Dubai, UAE
akulpai98@gmail.com

***Abstract*— Deep learning & especially Convolutional Neural Networks (CNNs) are taking a different shape in the area of Image Recognition & Classification. Number of different CNN architectures are found in the literature. Performance of any CNN model depends on various parameters such as size of the dataset, number of classes, weights of the model, hypermeters and optimizers so on. Transfer learning or Fine tuning a pretrained model has become very common during present time because of its advantages. Several model optimization techniques have been discussed in the literature. Stochastic Gradient Descent (SGD), Adam & RMS Prop model optimizers are more commonly used in CNN model optimization.**

This paper focusses on the effect of above three optimizers on well-known CNN models namely, ResNet50 and InceptionV3. Above optimizers are used to train the Fine-tuned CNN models for 15 Epochs on cat vs dog dataset created by hand picking hundreds of images of cats & dogs from Kaggle cat vs dog dataset. For this experimentation, learning rate of 0.001 is used, Categorical Cross Entropy is used to calculate the training & validation loss. Comparative analysis is made between optimizers by plotting training loss vs epochs & training accuracy vs epochs. Results showed that, SGD optimizer outperforms other two for ResNet50. Training accuracy of approximately 99% is observed for ResNet50 with 500 training & 100 validation images.

Keywords*— *Deep learning, Convolutional Neural Network, Fine-Tuning, Model-Optimizer.

I. INTRODUCTION

Deep learning has become one of the most important tools for image analysis. They yield very high accuracy in Image classification when compared to traditional computer vision algorithms when size of the dataset is large [1]. According to researchers, deep learning technique gives high accuracy when size of the dataset is large. [2]. This is evident, as, deep learning model must be trained on dataset before classifying an image. Fig. 1 shows that, when size of the dataset is smaller, traditional computer vision algorithms may outperform deep learning algorithms [3].

Deep learning models are built using neural network architectures. The neural network utilizes parameterized, sparsely connected kernels which preserve spatial characteristics of images [3]. Convolutional Neural Networks (CNN) are type of Artificial Neural Networks (ANN) especially suitable for Image analysis [4]. CNN models developed over last few years showing excellent results when used for various machine learning tasks.

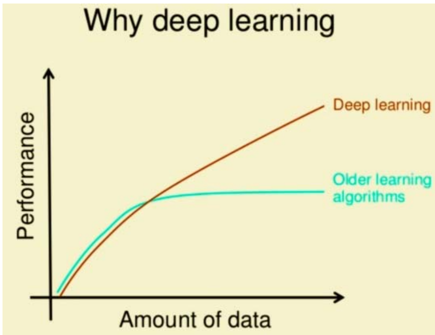

Fig. 1: Deep learning vs Traditional computer vision algorithms

Presently it is the most powerful and universally accepted deep learning model[5]. Convolutional Neural Network, a machine learning algorithm is well suited for the image classification. It is a supervised deep learning approach which requires large labelled data for training on the network [6]. CNNs are made of large number of convolutional layers with different dimensions. They also have other layers such as max pooling and fully connected layers. Advantage of CNN is that it needs very little preprocessing with original images [7]. CNNs can be trained on large datasets with millions of parameters, in form of 2D images as inputs and convolve it with filters to produce outputs [8].

In this paper, two CNN models are fine tuned for custom dataset created using Kaggle cat vs dog database. Performance analysis is carried out using Cross-Entropy and Accuracy as loss and metrics respectively. Model is optimized using Stochastic Gradient Descent (SGD) method. Experiment is carried out for 50 epochs with a batch size of 10. Proposed models have been fine-tuned using Keras deep learning framework. Results prove that overfitting can be prevented by Data Augmentation. Proposed models use Batch normalization and drop out layers in the classification section of the model.

The paper is organized as follows: Section 2 of the paper reviews several papers on fine-tuned CNN models. Section 3 of the paper provides the architectures of four different CNN models namely AlexNet, VGG16, ResNet50 & InceptionV3. Above CNN models were originally pre-trained on ImageNet dataset. ImageNet is an image database consisting of 1000 different classes. Section 4 focusses on the methodology followed in this work. Section 5 shows the results along with discussions. Section 5 concludes the work.

978-1-7281-5532-6/19/$31.00 ©2019 IEEE

II. CNN Models & Architectures

A. Alexnet

This was first CNN model which has been pre-trained for ImageNet data set. This architecture comprises of 5 convolutional layers and 3 fully connected layers. This architecture uses ReLU for the non-linear part and hence trains at faster rate. It has advantages such as less training parameters and strong robustness [9]. Top-5 accuracy of Alexnet when pretrained on ImageNet is 80.3%.

B. VGG 16

It is an improved version of Alexnet model, in which in place of large kernel-sized filters, multiple filters with smaller size of 3 x 3 are placed one after another. Architecture of VGG16 pretrained CNN model has been depicted in Fig 2. The network has 41 layers. There are 16 layers with learnable weights: 13 convolutional layers, and 3 fully connected layer and has been originally trained on 1000 image classes. The network has an input image size of 224 by 224. The VGG convolutional layers are followed by 3 fully connected layers. It achieves the top-5 accuracy of 92.3 % on ImageNet [10].

C. ResNet50

ResNet short for Residual Network is a classic neural network used as a backbone for many computer visions tasks [11]. This model was the winner of ImageNet challenge in 2015. This network is 50 layer deep. Main innovation in ResNet50 is the skip connection in which original input is added to the output of the Residual block as shown in Fig. 2. This eliminates the vanishing gradient which occurs during backpropagation when model is being trained.

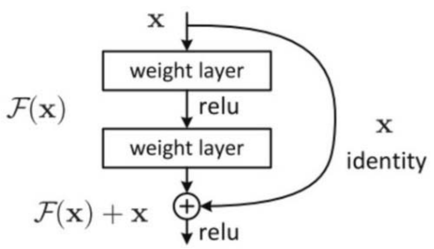

Fig. 2: Residual block

D. InceptionV3

Inception v3 has very complex CNN architecture when compared to other CNN models. Originally this model was trained on ImageNet dataset by using Tensorflow machine learnin framework [2]. Inception module consists of parallel combination of convolution layers with different sizes. This network is 48 layers deep with approximately 23 million parameters. This network is a winner of ILSRVC 2014.

III. Model Optimization Techniques

This section reviews some of the popular optimization algorithms used for training deep learning models:

A. Stochastic Gradient Descent (SGD)

SGD is one of the widely used algorithms for machine learning and for deep learning. Learning rate plays a critical role in SGD algorithm [13]. SGD updates model parameters in the negative direction of gradient by taking a subset of mini-batch of data size (m) as shown in Equations (1) & (2):

$$g = \frac{1}{m} \nabla_\theta \sum_i L(f(x^{(i)};\, \theta),\ y^{(i)}) \tag{1}$$

$$\theta = \theta - \epsilon_k \,.\, g \tag{2}$$

Where, θ is the model parameter, g is the gradient, ϵ_k is the learning rate & L is the loss-function. Computation-time per update doesn't increase with size of the training dataset. This is an important property of SGD [14].

B. Adaptive Moment Estimation (Adam)

This optimization technique is derived from "Adaptive moments". The ADAM algorithm combines the advantages of both the AdaGrad and RMSProp algorithms [15]. In Adam, momentum is incorporated directly as an estimate of the first-order moment of the gradient. Adam includes bias corrections to the estimates of both the first-order moments and the second-order moments to account for the initialization at the origin. Adam is computationally efficient and has very little memory requirement [16].

The first-order and second-order moments of the gradients can be calculated by:

$$\begin{cases} \beta_t \leftarrow \rho_1 \beta_{t-1} + (1-\rho_1)\, g_t \\ \gamma_t \leftarrow \rho_2 \gamma_{t-1} + (1-\rho_2)\, g_t^2 \end{cases} \tag{3}$$

C. RMSProp

This optimizer makes use of exponentially decaying averaging technique to optimize the CNN model and to discard the history from extreme past. Learning Rate (LR) in the case of RMSProp gets adjusted automatically and chooses different parameters accordingly. In the case of RMSProp, it divides the LR by the average of the exponential decay of squared gradients. This optimizer can be considered as very effective and practical algorithm for deep learning.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{(1-\gamma)\, g_{t-1}^2 + \gamma\, g_t + \epsilon}} \,.\, g_t \tag{4}$$

IV. Methodology

In this paper, two well-known CNN models are trained using following optimizers, SGD, Adam & RMSProp. Section 3 gives brief explanation on each of them. Inception v3 & ResNet 50 CNN models which are used for analyzing the effect of Model Optimizers. Fine tuning is achieved using Keras deep learning framework.

Advantage of fine-tuning is, it takes less time for training a model on custom dataset as number of trainable parameters is very less when compared to Total number of parameters. CNN architecture has two sections namely, Feature Extraction & Classification sections. In the case Transfer Learning, last few layers corresponding to classification part only will be trained and weights of bottom most layers corresponding to feature extraction are freezed and will not be trained. Where as in the case of Fine-Tuning, after above process some or all of the layers pertaining to feature extraction are unfrozen and fine-tuned. Fine tuning takes more time than transfer learning but yields better result.

V. RESULTS

Following section shows the Accuracy & loss curves in-terms of number of Epochs. Inception v3 & ResNet50 models have been compiled and trained using following specifications:

- Learning Rate: 0.001
- Metric: Accuracy
- Loss function: Cross-Entropy
- No. of Epochs: 50
- Training & Validation Batch size: 10
- Total no. of images in Train dataset: 250 for each class
- Total no. of images in Validation Dataset: 50 for each class

Fig. 3 & Fig. 4 show the training accuracy & training loss curves for Inception v3 with respect to number of epochs. It can be observed from the figure that, SGD optimizer is slower in reaching the optimal value when compared to other two optimizers but stabilizes after 15 epochs. For the given dataset, Fig. 3 shows that RMS Prop has slightly better accuracy compared to other two. In terms of training loss, SGD outperforms other two. SGD might have shown better accuracy if number of epochs would have been more than 50.

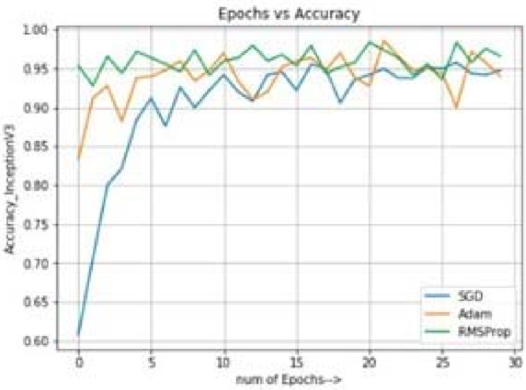

Fig. 3: Training Accuracy for Inception v3

Fig. 5 & Fig. 6 show training accuracy & training loss for ResNet 50 model. As obvious from the graph that, SGD has better response in terms of accuracy and loss.

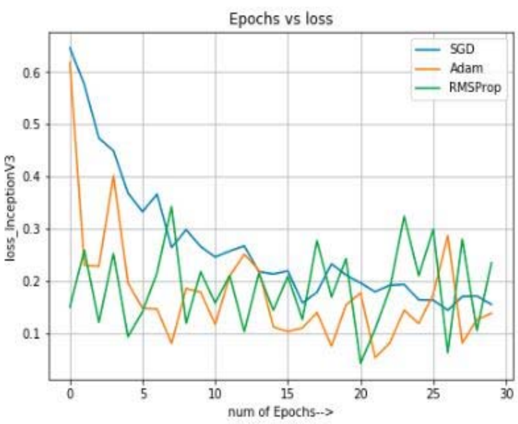

Fig. 4: Training Loss for Inception v3

This work uses a learning rate of 0.001. Reduction in the learning rate can achieve more accurate results but takes more time to attain the optimal value. By having additional layers such as batch normalization, stability in accuracy will be observed.

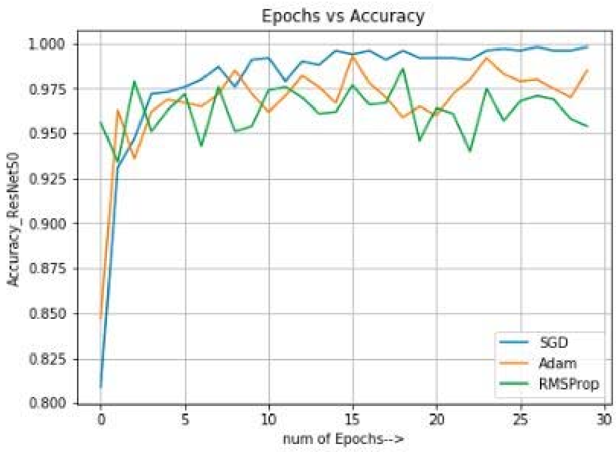

Fig. 5: Training Accuracy for ResNet 50

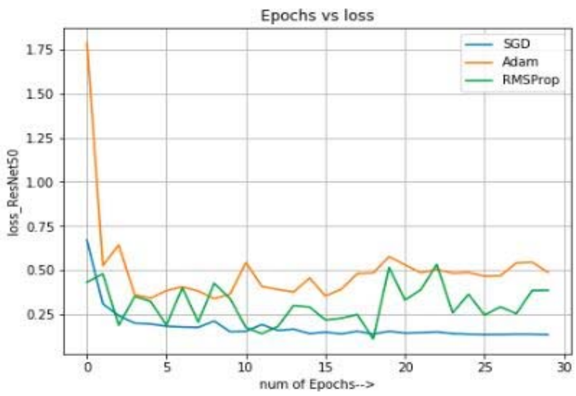

Fig. 6: Training Loss for ResNet50

VI. CONCLUSIONS

Transfer Learning & Fine-tuning CNN models are the easiest way to build models if the given task is similar to the original task. One of the parameters which determines the performance of fine-tuned model is model-optimization technique. This paper discusses the performance analysis of two fine-tuned CNN models which are trained using SGD, Adam & RMSProp optimizers. Inception v3 & ResNet50 CNN models are used for the analysis. Training accuracy & cross entropy curves are used as statistical parameters for comparison between optimizers. SGD outperforms other two optimizers in the case of ResNet50 where as RMSProp has slightly better accuracy when compared to other in the case of Inception v3 when number of epochs was set as 50.

REFERENCES

[1] Arkadiusz Kwasigroch, Agnieszka Miko?ajczyk , Micha? Grochowski, "Deep neural networks approach to skin lesions classification - A comparative analysis", 22nd International Conference on Methods and Models in Automation and Robotics (MMAR), 2017, Poland.

[2] Manisha Saini, Seba Susan, "Comparison of Deep Learning, Data Augmentation and Bag of-Visual-Words for Classification of

Imbalanced Image Datasets", Communications in Computer and Information Science book series (CCIS, volume 1035), pp 561-571, 2019.

[3] Sambit Mahapatra, "Why Deep Learning over Traditional Machine Learning?", Accessed on: Mar 21, 2018. [Online]. Available: https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063.

[4] Muthukrishnan Ramprasath, "Image Classification using Convolutional Neural Networks", International Journal of Pure and Applied Mathematics, Volume 119 No. 17 2018, 1307-1319.

[5] Guojian Cheng and Wenhui Guo, "Rock images classification by using deep convolution neural network", IOP Conf. Series: Journal of Physics, 887 (2017) 012089.

[6] Rahul Chauhan, Kamal Kumar Ghanshala,R C Joshi, "Convolutional Neural Network (CNN) for Image Detection and Recognition", First International Conference on Secure Cyber Computing and Communication (ICSCCC), 15-17 Dec 2018.

[7] Yu-Dong Zhang, Zhengchao Dong, Xianqing Chen etal, "Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation", An International Journal of Multimedia Tools and Applications, Volume 78, Issue 3, pp 3613-3632, February 2019.

[8] M. Shamim Hossain, Muneer Al-Hammadi, Ghulam Muhammad, "Automatic Fruits Classification Using Deep Learning for Industrial Applications", IEEE Transactions on Industrial Informatics, Vol.15, Issue 2, Feb 2019.

[9] Poonam Dhankhar, "ResNet-50 and VGG-16 for recognizing Facial Emotions", International Journal of Innovations in Engineering and Technology (IJIET), pp. 126-130, Volume 13, Issue 4, July 2019.

[10] Sergey Ioffe, Christian Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariance Shift", 32nd International Conference on Machine Learning, Volume 35, Lille, France, 2015.

[11] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Rusian Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Over?tting", Journal of Machine Learning Research 15 (2014) pp.1929-195.

[12] Milton-Barker, Adam, "Inception V3 Deep Convolutional Architecture For Classifying Acute Myeloid/Lymphoblastic Leukemia", Accessed on: February 17, 2019. [Online]. Available: https://software.intel.com/en-us/articles/inception-v3-deep-convolutional-architecture-for-classifying-acute-myeloidlymphoblastic.

[13] Vadim Smolyakov, "Neural Network Optimization Algorithms", Accessed on : Oct 21, 2019. [Online]. Available: https://towardsdatascience.com/neural-network-optimization-algorithms-1a44c282f61d.

[14] Ian Goodfellow and Yoshua Bengio and Aaron Courville, "Deep Learning", Chapter 3, MIT Press, 2016.

[15] Yingyi Sun, Hao Gu, Wenhua Xu, And Guan Gui, "Convolutional Neural Network Based Models for

[16] Improving Super-Resolution Imaging", IEEE Access, Vol.7, April 2019, pp. 43042-43051.

[17] Renu Khandelwal, "Overview of different Optimizers for neural networks", Accessed on: Oct 7, 2019. [Online].Available: https://medium.com/datadriveninvestor/overview-of-different-optimizers-for-neural-networks-e0ed119440c3.