# Effects of Approximation in Computation on the Accuracy and Performance of Deep Neural Network Inference

**Hui Nee Ow[1], Usman Ullah Sheikh[2] and Musa Mohd Mokji[3]**
[1, 2] Intel (M) Sdn Bhd, Pulau Pinang, MALAYSIA.
(E-mail: hannahow984@gmail.com)

[2, 3] School of Electrical Engineering, Faculty of Engineering, Universiti Teknologi Malaysia, MALAYSIA.
(E-mail: usman@utm.my, musa@fke.utm.my)

**Abstract.** Recently, deep learning is at the forefront of the state-of-the-art machine learning algorithms and has shown excellent results in a variety of applications such as medical field, consumer as well as autonomous vehicles. Convolutional Neural Network (CNN) – is the leading deep learning architecture that is mostly applied. However, huge dataset is needed to train with complex architecture to achieve precise learning. Inference can be performed when given a ready CNN model and its weight file to another user. Inference takes time with precise weights and huge dataset. To overcome this problem, and enhance the inference system, approximation computation will be applying in terms of weight for changed of decimal place. The smaller size of the dataset is used in the inference process to reduce the inference time. MobileNetV2 architecture is used with the new weight for inference. Also, open source libraries such as TensorFlow, Keras and python is used. GPU (NVIDIA GeForce GTX 1060 6GB 64 Bit) is used as training and inference platform. Inference time is shortened, and the accuracy of performance for new weights compare with the precise weight only has a small gap which still has a great performance for classification. This work has proved that with 4 decimal places is able to obtain the same accuracy for inference when compared to benchmark with 9 decimal places. Inference time for 4 decimal places is also less than benchmark time.

*Keywords*: Approximation computing, deep neural network, performance

## 1. Introduction

One of the pillars of Industry 4.0 is the ability of cyber-physical systems to perform decentralized decisions. This ability to make decisions autonomously requires complex machine learning algorithms and the ability to produce instant output. Recently, deep learning is at the forefront of state-of-the-art machine learning algorithms and has shown excellent results in a variety of applications. For example, deep learning is used in plagiarism checkers in education to trace students who have copied works from other authors. Deep learning is also used in autonomous vehicles for auto-driving and auto-parking. In social media, Facebook applied deep learning for face detection on images uploaded by users for easy recognition and tagging of persons inside images.

CNN is the leading deep learning architecture that is mostly applied. Deep learning is used because it has great power and flexibility and is trained with a huge amount of data [1]. A high-end machine such as GPU is used in deep learning for training and testing. Machine learning normally has feature extraction and classification separately while deep learning combines feature extraction and classification in one process. Deep learning is able to solve problems from end to end. So, it is great to use for complex applications such as natural language processing, image classification, speech recognition and other applications. Typical CNN consists of multiple layers of convolutional, pooling and connected layers. For example, the AlexNet CNN was made up of 5 convolutional layers, 3 connected layers, dropout layers and max pooling layers. In this thesis, the effects of the precision of weights on a CNN deep learning architecture is explored. Improvements in terms of inference accuracy and inference time are analysed to obtain optimum performance.

Image classification is highly used in social media, industry for data collection and distribution purposes. For example, an autonomous car needs image classification to determine the object when it is on the road and to make the correct decision. However, most implementation of deep learning faces uncertainty when selecting the precision of weights. Long training and inference time is needed due to complex architecture with precise values or data. High precision gives high accuracy [2], but it also increases the inference time. Inference time is important in real time systems as the data is streaming at a fast rate. By lowering the precision, the inference time can be reduced but the accuracy will also be decreased [3]. So it is important to have a methodology to decide the balanced point or optimum trade off between inference accuracy and inference time. The objective for this paper is to evaluate the effects of different levels of precision on weights in terms of inference accuracy and inference time and to determine the best precision of weights in terms of accuracy and inference time for MobileNetV2.

## 2. Related Works

In the year 2012, Alex Krizhevsky, Llya Sutskever and Geoffrey E.Hinton have proposed an image classification using CNN model [4] to classify 1.2 million high-resolution image in the ImageNet LSVRC-2010 which contain 1000 classes. It has achieved top-1 and top-5 error rates of 37.5% and 17% which are better error rates than previous state-of-art methods. The first five layers of the architecture are convolution layers while three layers of fully connected layers thus having a total of eight layers with weights. Another work by Xiaoling Xia, Cui Xu and Bing Nan in 2017 proposed Inception-v3 model [5] with transfer learning technology to retrain the flower category dataset to improve the accuracy of flower classification. Transfer learning technique achieve accuracy of 95% using Oxford-17 input dataset and 94% by using Oxford-102 input dataset. Jonathan Bhaskar and Ankit Patel have used pre-trained model of AlexNet [6] with Caffe as a platform to classify Instagram photos into two categories, bad or good. Using another CNN model such as VGG-16 achieve more than10% accuracy [6].

Fixed point quantization of deep convolutional network has been introduced by Darryl D.Lin, Sachin S Talathin and V.S reekanth Annapureddy [7] in 2015. They proposed a quantizer design for fixed point implementation of DCN and it optimized more than 20% reduction in the model size without a loss in accuracy, and achieved fixed point performance of 6.78% error-rate on CIFAR-10 benchmark. In the year 2017, Incremental Network Quantization: Toward Lossless CNN with Low Precision Weight [8] has been proposed by Aojun Zhou. Different types of CNN models have been used to measure the accuracy of low precision bit width. Accuracy has been proven to increase by lowering the bit width precisio. Another Training Inference proposed by Shuang Wu developed a method called as WAGE which discretizes both training and inference parameters such as weight, activation, gradient and errors [9]. The approximation is by lowering the bit-width with an integer in the DNN model. In the year 2018, Yian Seo has proposed image classification of fine-grained Fashion Image based in Style using Pre-trained CNN [10]. The last

fully- connected later of the pretrained GooLeNet is remove and trained with ImageNet dataset. Last fully-connected layer is fine-tuned with own fine-grained fashion dataset. In 2018, Intel has introduced the lower precision of deep learning for training and inference [2]. The concept has been proven that improvement with minimal or no reduction for computational performance in term of accuracy. Inference with 8-bit precision has been used in the Intel Xeon Scalable processor as well [2].

## 3. Methodology

Fig. 1 shows the core process flow. The first step is to prepare and categorized the dataset. Next, a pre-trained model is obtained from Keras for training, validating and testing to obtain JSON and weight file in H5 format. The weight file is opened with H5 and read into each hierarchy to reach the kernel for a specific layer. Changing of weights with different precision (decimal place) is processed by python and then saved to be used for inference. Different combination of patterns and arrangements of layer precision are make to experiment the effects of precision on the inference accuracy and time. After the inference process, accuracy, inference time and confusion matrix are collected and saved in a folder.

There are many types of CNN's models introduced recently. Some CNN pre-trained models are available in the open source library, Keras [11]. MobileNetV2 [12] is used in this project because it is small in size which is about 14MB and the top-1 accuracy and top-5 accuracy is the same as VGG16 which has a size of 528MB. The depth of MobileNetV2 is 88. Depth is the topological depth of the network that including activation layers, batch normalization layers etc. MobileNetV2 is composed of a fully connected layer followed by 16 inverted layers of residual block with a fixed expansion ratio. It is using a depthwise separable convolution building block. The structure of the inverted layers of residual block consists of an expansion layer followed by depthwise layer and projection layer. Each type of layer consists of a sub-layer which is the convolutional layer, ReLU, and batch normalization layer. The expansion layer acts as decompressor or an unzip model that will restore the data in its full form. The depthwise layer performs filtering which is an important stage in the MobileNetV2 network. The projection layer is to compress the data to make it small again.

Different neural network models have been trained with different input datasets. It is important to pre-process an image before using it so it is compatible with the CNN model. There are common image data parameters such as the width and height of the image, the number of levels per pixel and the number of channels [15]. Pixel level is from 0 to 255 while three channels are corresponding to Red, Green and, Blue (RGB). The image is converted to the described format before load to the training model. Every image must have a uniform aspect ratio and size. Most of the neural network models will assume the image is in square shape. So, cropping is needed and the center part of the image will be taken as the focus. Next, the image is scaled into appropriate pixel resolution. For example, if an image is has a resolution of 250 x 250 and requires rescaling to 100 x 100, then the scale factor will be 0.4.

In this project, the dataset used is Cifar-10 which consists of 10 classes. In this project, Cifar-10 is distributed into 3 sets (40,000 images for training, 10,000 images for validating and 10,000 images for testing). The 10 classes in Cifar-10 are airplane, automobile, bird, cat, deer, dog, frog, horse, ship and, truck. Training set is used to adjust the weight, validating set is used to minimize the overfitting while testing set is used for the inference process.

The varying of weights precision in decimal places is applied to weights in MobileNetV2. Fig. 2 shows the targeted layers of its kernel to have weight precision changed. The kernel is the target for weight precision changes because it is important for image feature extraction and mapping process. There are 16 bottleneck residual blocks or inverted layer residual blocks affected and also the first 2 layers of convolutional layer kernels affected. Dataset dataspace and datatype is viewed by using the HD5 viewer. There are different type of weights precision to

apply on kernel of layers. Precision changed in terms of decimal places from one decimal place to nine decimal places of kernel of the targeted layers. Weights precision changed has categories into four experiments. First experiment, weights for all targeted layers changed from 1 decimal place to 9 decimal places. Second experiment, precision changed only for specific layers with range 1 to 9 decimal places. Third experiment, one or more layers weights precision are changed at one time. Last experiment is only changed specific layer at one time with odd numbers.

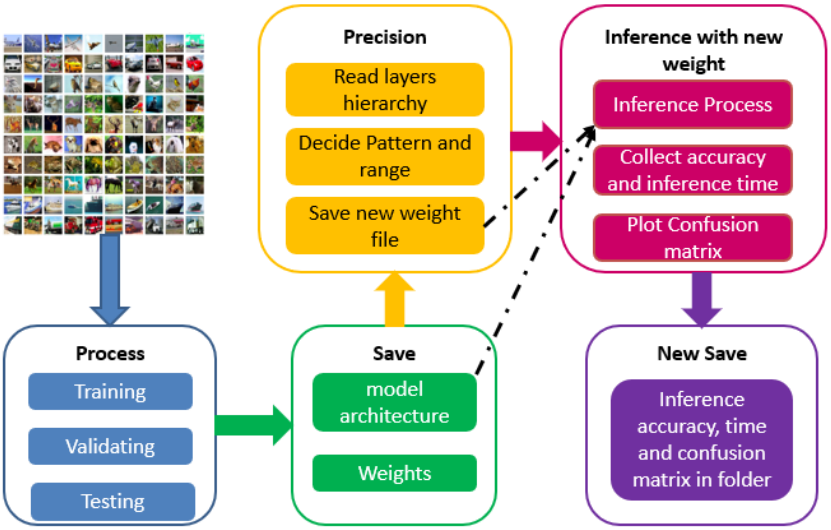


**Figure 1.** Core process flow.

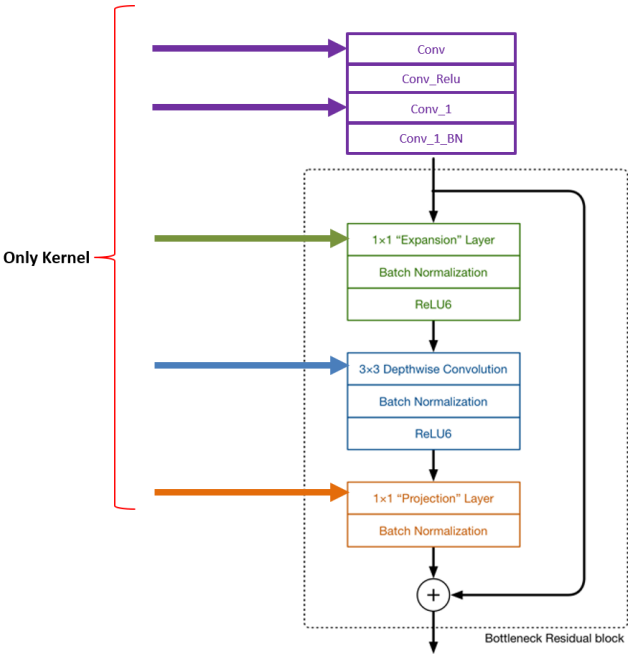


**Figure 2.** Targeted layers

## 4. Results and Discussion

The benchmark of this project is done through the training and testing process by using the MobileNetV2 model and Cifar-10 dataset. There are several parameters that have to be set before the start of the training and testing process. The purpose of the parameters setup is not to obtain optimum accuracy. The purpose of this project is to evaluate the inference accuracy and time. After the training and testing process, model architecture and weight file are obtained and ready to be used for inference. The inference accuracy for the benchmark is 86.91% with inference time 87.72 seconds. The kernel size in terms of element is 2,202,560. In this experiment, the weights for kernels in target layers is changed from decimal places 1 to 9 as shown in Table I. Overall kernel data size has been affected is 2,196,736 which is 100% for the targeted kernel. Fig. 7 shows the inference accuracy and time for all decimal place combinations. 4 decimal places has an accuracy of 86.95% compared to base accuracy of 86.91% which is higher by 0.04% with less inference time of 69.04 seconds.

**Table 1.** Combination of Different Precision Weights Used for Each Layer in Experiment 1

| | Experiment | Layers | | | | | Kernel Size |
|---|---|---|---|---|---|---|---|
| | | *Conv* | *Conv_1* | *EXP* | *DW* | *PRJ* | |
| **Precision** | a1 | 1 | 1 | 1 | 1 | 1 | 2,188,960 |
| | a2 | 2 | 2 | 2 | 2 | 2 | 2,188,960 |
| | a3 | 3 | 3 | 3 | 3 | 3 | 2,188,960 |
| | a4 | 4 | 4 | 4 | 4 | 4 | 2,188,960 |
| | a5 | 5 | 5 | 5 | 5 | 5 | 2,188,960 |
| | a6 | 6 | 6 | 6 | 6 | 6 | 2,188,960 |
| | a7 | 7 | 7 | 7 | 7 | 7 | 2,188,960 |
| | a8 | 8 | 8 | 8 | 8 | 8 | 2,188,960 |
| | a9 | 9 | 9 | 9 | 9 | 9 | 2,188,960 |


**Figure 3.** Results of Experiment 1

## 5. Conclusion

In this work, MobileNetV2 is used with transfer learning technique for image classification. Precision computation is implemented on the weights of MobileNetV2 in terms of decimal places to represent the weights. There are many layers in MobileNetV2, only the kernel on each layer are targeted for precision computation in this work. The result shows that the decrease in number of decimal places in the weights decreased the inference time, while the inference accuracy affected by types of layer selected for precision and the decimal places used in the kernel.

## Acknowledgment

The authors would like to thank Universiti Teknologi Malaysia and Intel Microelectronic (M) Sdn Bhd for the support.

## References

[1] Sambit Mahapatra, "Why Deep Learning over Traditional Machine Learning?," Towards Data Science, 2018. [Online]. Available: https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063. [Accessed: 12-May-2019].
[2] A. Rodriguez, E. Segal, E. Meiri, E. Fomenko, Y. J. Kim, and H. Shen, "Lower Numerical Precision Deep Learning Inference and Training," pp. 1–19, 2018.
[3] Z. Carmichael, H. F. Langroudi, C. Khazanov, J. Lillie, J. L. Gustafson, and D. Kudithipudi, "Performance-Efficiency Trade-off of Low-Precision Numerical Formats in Deep Neural Networks," 2019.
[4] Alex Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Am. J. PharmacoGenomics, pp. 1–9, 2012.
[5] X. Xia and B. Nan, "Inception-v3 for Flower Classification," pp. 783–787, 2017.
[6] J. Bhaskar and P. Ankit, "Image Classification using Convolutional Neural Network."
[7] D. D. Lin, S. S. Talathi, and V. S. Annapureddy, "Fixed Point Quantization of Deep Convolutional Networks," vol. 48, 2015.
[8] A. Zhou, A. Yao, Y. Guo, L. Xu, and Y. Chen, "Incremental Network Quantization: Towards Lossless CNNs with Low-Precision Weights," pp. 1–14, 2017.
[9] S. Wu, G. Li, F. Chen, and L. Shi, "Training and Iference with Integers in Deep Neural Networks," pp. 1–11, 2018.
[10] Y. Seo and K. S. Shin, "Image classification of fine-grained fashion image based on style using pre-trained convolutional neural network," 2018 IEEE 3rd Int. Conf. Big Data Anal. ICBDA 2018, pp. 387–390, 2018.
[11] "Applications - Keras Documentation." [Online]. Available: https://keras.io/applications/. [Accessed: 09-Jun-2019].
[12] M. Sandler, M. Zhu, A. Zhmoginov, and C. V Apr, "MobileNetV2: Inverted Residuals and Linear Bottlenecks."
[13] "Google AI Blog: The NeurIPS 2018 Test of Time Award: The Trade-Offs of Large Scale Learning." [Online]. Available: https://ai.googleblog.com/2018/12/the-neurips-2018-test-of-time-award.html?m=1. [Accessed: 29-Apr-2019].
[14] "MobileNet version 2." [Online]. Available: https://machinethink.net/blog/mobilenet-v2/. [Accessed: 08-Apr-2019].
[15] Nikhil Balaji, "Image Data Pre-Processing for Neural Networks – Becoming Human: Artificial Intelligence Magazine," 2017. [Online]. Available: https://becominghuman.ai/image-data-pre-processing-for-neural-networks-498289068258. [Accessed: 11-Dec-2018].