Michigan Technological University
Digital Commons @ Michigan Tech

Dissertations, Master's Theses and Master's Reports

2017

Performance Comparison of Binarized Neural Network with Convolutional Neural Network

Lopamudra Baruah
Michigan Technological University, lbaruah@mtu.edu

Copyright 2017 Lopamudra Baruah

Recommended Citation

Baruah, Lopamudra, "Performance Comparison of Binarized Neural Network with Convolutional Neural Network", Open Access Master's Report, Michigan Technological University, 2017.
https://doi.org/10.37099/mtu.dc.etdr/487

Follow this and additional works at: https://digitalcommons.mtu.edu/etdr

Part of the Other Computer Engineering Commons, and the Other Electrical and Computer Engineering Commons

PERFORMANCE COMPARISON OF BINARIZED NEURAL NETWORK WITH CONVOLUTIONAL NEURAL NETWORK

By

Lopamudra Baruah

A REPORT

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

In Electrical Engineering

MICHIGAN TECHNOLOGICAL UNIVERSITY

2017

© 2017 Lopamudra Baruah

This report has been approved in partial fulfillment of the requirements for the Degree of MASTER OF SCIENCE in Electrical Engineering.

Department of Electrical and Computer Engineering

Report Advisor: *Zhuo Feng*

Committee Member: *Zhaohui Wang*

Committee Member: *Timothy Havens*

Department Chair: *Daniel R. Fuhrmann*

TABLE OF CONTENTS

1. Introduction 5

2. Methods 11

    2.1.Environment Setup 11

    2.2.Software Packages 12

    2.3.Neural Network Architecture 13

3. Results and Discussion 15

4. Conclusion 31

5. Acknowledgement 33

6. References 34

ABSTRACT

Deep learning is a trending topic widely studied by researchers due to increase in the abundance of data and getting meaningful results with them. Convolutional Neural Networks (CNN) is one of the most popular architectures used in deep learning. Binarized Neural Network (BNN) is also a neural network which consists of binary weights and activations. Neural Networks has large number of parameters and overfitting is a common problem to these networks. To overcome the overfitting problem, dropout is a solution. Randomly dropping some neurons along with its connections helps to prevent co-adaptations which finally help in reducing overfitting. Many researchers have analyzed the performance of CNN and studied about the effect of dropout on CNN using datasets like MNIST and CIFAR10. The factors like Dropout rate, Dataset size, Batch Normalization layer, Filter size, and Dropout layer addition has been studied on CNN. But there is a lack of literature in the study of dropout and the various factors in Binarized Neural Network. This report will provide a brief introduction about BNN, the advantage of using dropout and the performance comparison between BNN and CNN. A detailed description of the software packages, coding environment, algorithm flow, and deep learning framework is provided. A comprehensive analysis on the performance of BNN and CNN is performed, and BNN shows near state-of-the-art results like CNN. The research demonstrates the adding of dropout layer to a BNN for MNIST and CIFAR10 datasets, and shows that it might provide improvement to the baseline BNN's classification accuracy. Finally, the report investigates the different factors such as Dropout rate, Dataset size, Batch Normalization layer, Filter size, and Dropout layer addition on BNN.

1. Introduction

In recent times, there has been a tremendous rise of data, and with the increase in the amount of data the focus has shifted to processing, managing, and analyzing the data. Thus, the focus on analyzing data and making accurate predictions based on that has become a major area of interest. Machine Learning algorithms have been studied by many researchers for data classification and recognition tasks. Deep Learning is a subset of machine learning which can learn on large amount of data that are unstructured and unlabeled. It is vastly used in autonomous driving, computer vision, natural language processing, speech recognition and many others. Deep learning uses many layers which perform feature extraction and transformation. There are many deep learning architectures, and Convolutional Neural Networks (CNNs) are one of the most widely used architecture.

Neural Networks mimic the structure of human brain which consist of neurons and connections between the neurons, i.e. synapses. The neuron connections store the interrelation between the neurons. Neural Networks are biologically inspired, and it serves as a powerful tool when combined with deep learning. When a child is initially taught about alphabets, it is hard for them to learn. But the more training is given to the child's brain, the brain infers its own rules and gives a good output. We as humans are training our brain every day, and our brain uses this information to make the decisions better. Just like a human brain, the neural network gives better output and learns better when provided with more training examples. While training, the neural network infers its own set of rules to make better predictions. Thus, if we train the neural network for a longer time, we will get a better prediction accuracy [5].

Neural Networks consist of input layer, hidden layers and output layer. The input layer consists of the input neurons, and the output layer has the output neurons. There can multiple hidden layers between the input and output layer. The connections or links between the neurons store the information of the relation between different nodes or neurons. There are different types of Neural Networks like feedforward network, recurrent neural network, and recursive neural network. But the feedforward neural network is the most widely used. [7]

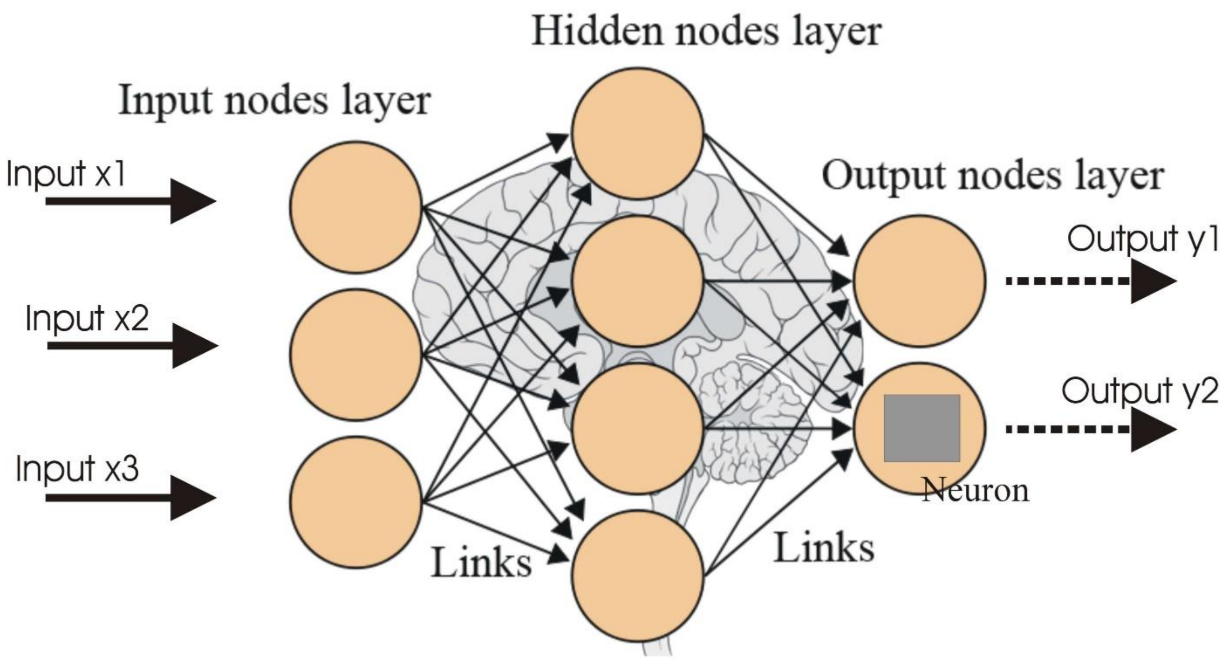


Figure 1: Architecture of Neural Network showing input nodes, hidden nodes and output nodes. Connections between the nodes have been shown in a graphical manner. [6]

Convolutional Neural Networks (CNNs) are more popular deep learning architectures, because they show high representation power and superior performance in both image and speech recognition tasks. They are also comparatively easier to train. All the neurons in one layer are not connected to all the neurons in the second layer. A CNN uses a filter which scans the input image and creates a feature map. [2] CNN consists of various prominent features which make them so effective.

In convolutional layer, the input feature map is convolved with the filter or kernel, and the convolution results are summed up and a bias is added to result in the output feature map. The output volume dimension can be determined using the following formula. It's a function of input volume image, and other hyper parameters [13].

$$W_{out} = 1 + \frac{W_{in} - F + 2P}{S}$$

A feature map is the output obtained when one filter is applied to a previous layer. When a 32*32 image is taken as an input, and a 3*3 filter size is applied across the input image with a stride 1 will result in a feature map of (32-3+1)*(32-3+1) which is 30*30 distinct activations per image.

Pooling layer controls overfitting, and it makes the network invariant or resistant to small distortions, scale invariant image representation of input image. Most commonly used dimensions for pooling layer is F=3, S=2, or F=2, S=2. Receptive field sizes>3 might result in worse performance [5]. For a 224*224*64 image, the pooling layer will generate 112*112*64 with F=2, S=2.

In fully connected layers, all the neurons are connected to all the neurons of the previous layer. They are used in the later part of the neural network to connect to the output layer and also to get the required number of outputs. Rectified Linear Unit (ReLU) is commonly used as an activation layer in a CNN [5].

Binarized Neural Network is similar to CNN, but consists of binary weights and activations. Matthieu C, et al, mentions that the conversion of floating-point weights to binary weights, eliminates the matrix accumulate operations and replaces them with simple additions and subtractions [3]. Binarized Neural Network contains multiple layers like Convolutional layer, Non-linear layer, Pooling layer, dropout layer etc just like Convolutional Neural Network.

The most computation intensive layer in the CNN is the convolutional layer. It takes the maximum computation time to complete, because of the matrix vector or matrix-matrix multiplication [4]. The convolutional neural networks use floating point weights in the convolutional layer to represent the connections. One of the alternatives to reduce the computation complexity is to use binary weights instead of floating point weights. Thus, Binarized Neural Network were introduced where the weights in the convolutional layer can be just two values: 0 or 1. Their performance is surprisingly on-par with the traditional CNN. [3]

Matthieu C, et al, worked on introducing Binarized Neural Network (BNN) which binarizes both the weights and activations simultaneously, and provides comparable accuracy to CNN. Although they have worked on BNN, they did not focus on the several factors which might affect the classification accuracy. The effect of dataset size, filter size, adding a dropout layer etc. were not explored. Related work has been done [8], [9], [10], [11] on Binarized Neural Network.

Since BNN or CNN deals with substantial number of parameters, regularization is a key step to reduce overfitting. Dropout is a common technique to address regularization [1]. In the dropout layer, randomly chosen units along with all its incoming and outgoing connections are dropped or removed from the network. The parameter p in the dropout layer is the probability of a unit of getting dropped out. It works as a regularization because when random neurons are dropped, the network learns several independent representations of the patterns with the same input and output. Thus, any dominating influence of a single node is decreased [1].

Many work has been done to regularize neural networks. Srivastava et al. [1] has demonstrated their work on dropout layer and proved that Convolutional Neural Networks using dropout achieved state-of-the-art results on different datasets like CIFAR-10, MNIST, SVHN, ImageNet. They also studied about the effect of data set size and dropout rate after adding dropout in convolutional layer of CNN. S.Park et al. [2] worked on introducing two variants of dropout in Convolutional Neural Network which gave competitive results to the conventional dropout. The results were implemented on various datasets, and the accuracies were calculated. Although Dropout has been studied by researchers, and the improvement of dropout on Convolutional Neural Network (CNN), there is lack of literature on the implementation of dropout layer on Binarized Neural Network (BNN), and thus it is an interesting area to focus on.

The focus of my research will be to compare the performance of CNN and BNN on different datasets like MNIST and CIFAR-10. The effect of implementation of the dropout layer on both a CNN and BNN will be studied. Factors like filter size, number of filters, size of data set, batch normalization, dropout rate will be studied on a BNN. According to S. Park et al. dropout layer is usually applied to fully connected layer in CNN. It is believed that convolutional layer suffers from less overfitting due to its smaller number of parameters compared to feature maps. This research will focus on adding dropout layer in convolutional layer of BNN, and analyze if it is effective for regularization. S. Park et al. mentions that the regularization effect due to adding dropout in convolutional layer is due to the additional robustness which is achieved by adding noise to the inputs of convolutional layer. Unlike convolutional layer, in fully connected layer dropout addition leads to generalization due to the averaging of models [2]. There is a lack of literature in the study of using dropout layer in both fully connected layer and convolutional layer in BNN. In this research, there will be analysis of how adding dropout in fully connected layer and convolutional layer affect the classification accuracy of a BNN. A detailed report on the experiments and its results are provided in the later sections.

2. Methods

2.1. Environment Setup

The cloud service platform Amazon Web Services (AWS) has been used to utilize the GPU computing power available online. Amazon EC2 is a part of Amazon Web Services cloud computing platform which provides people to use the cloud for running applications in a virtual environment. Amazon Web Services allows its users to rent virtual computing environments, thus giving flexibility to run high computation expensive tasks without investing on hardware. The virtual computing environments in Amazon Web Services are known as instances. Amazon Machine Image (AMI) is a preconfigured setting including operating system and additional software, and an AMI can be used with the instances [12]. An Ubuntu Server 16.04 LTS AMI is used to run the tests. Ubuntu operating system is used because of its compatibility with various deep learning software packages and libraries.

In AWS, there is an option to choose from different processor configurations which are called Instance Type. A GPU is needed to run the programs for deep learning applications. A g2.2x large instance type is utilized to run the tests. This instance comes with the following specifications:

g2.2x large: 26 ECUs, 8 vCPUs, 2.6 GHz, Intel Xeon E5-2670, 15 GiB memory, 1 x 60 GiB Storage Capacity

g2.2x large has two NVIDIA GRID K520 GPUs. This instance type requires approximately 2 hours to run a single iteration/epoch in my CNN, and the cost of running a g2.2x large EC2 instance is \$0.65/hour. Amazon Web Services allows secure login to the instances using key pairs. Using key pair, the user can access and run the GPU instance using secure shell (ssh).

2.2. Software Packages

The environment will need the following software packages for the application or code to run:

- Python

  The code is written in Python, and it is run in a Linux environment.

- Theano

  Theano is a Python library whose computations are done to run efficiently on a CPU or GPU. Theano framework is used to run the Binarized Neural network and Convolutional Neural Network.

- Lasagne

  Lasagne which is tightly integrated with Theano is a useful Python module whose blocks make building neural networks more convenient and simple. It is a library which has been created to build and train neural networks in Theano.

- Pylearn2

  Pylearn2 is a machine learning library which is used for accessing the different datasets like MNIST and CIFAR-10.

2.3. Neural Network Architecture

The Binarized Neural Network consists of 4 convolutional layers, 2 pooling layers, and 3 dense or fully connected layers. The input is first fed into the convolutional layer, where the filter is slided through the image to get feature maps. Figure 2 shows the architecture of the Binarized Neural Network.

CIFAR10 is an image classification dataset which consists of 50,000 images which includes 45,000 training images and 5,000 test images. The images are 32*32 size. For CIFAR10, the code is run for 25 epochs with a batch size of 50. The initial learning rate is set to be 0.001. The classification error reported for CIFAR10 is the test error of the model.

MNIST is a handwritten digits' dataset which consists of 60,000 images, including 50,000 training images and 10,000 test images. The input images are of size 28*28. The test error reported for MNIST is the average of the error reported after running the model 3 times. The Binarized Neural Network code for MNIST is run for 20 epochs with a batch size of 128. The initial learning rate is 0.01.

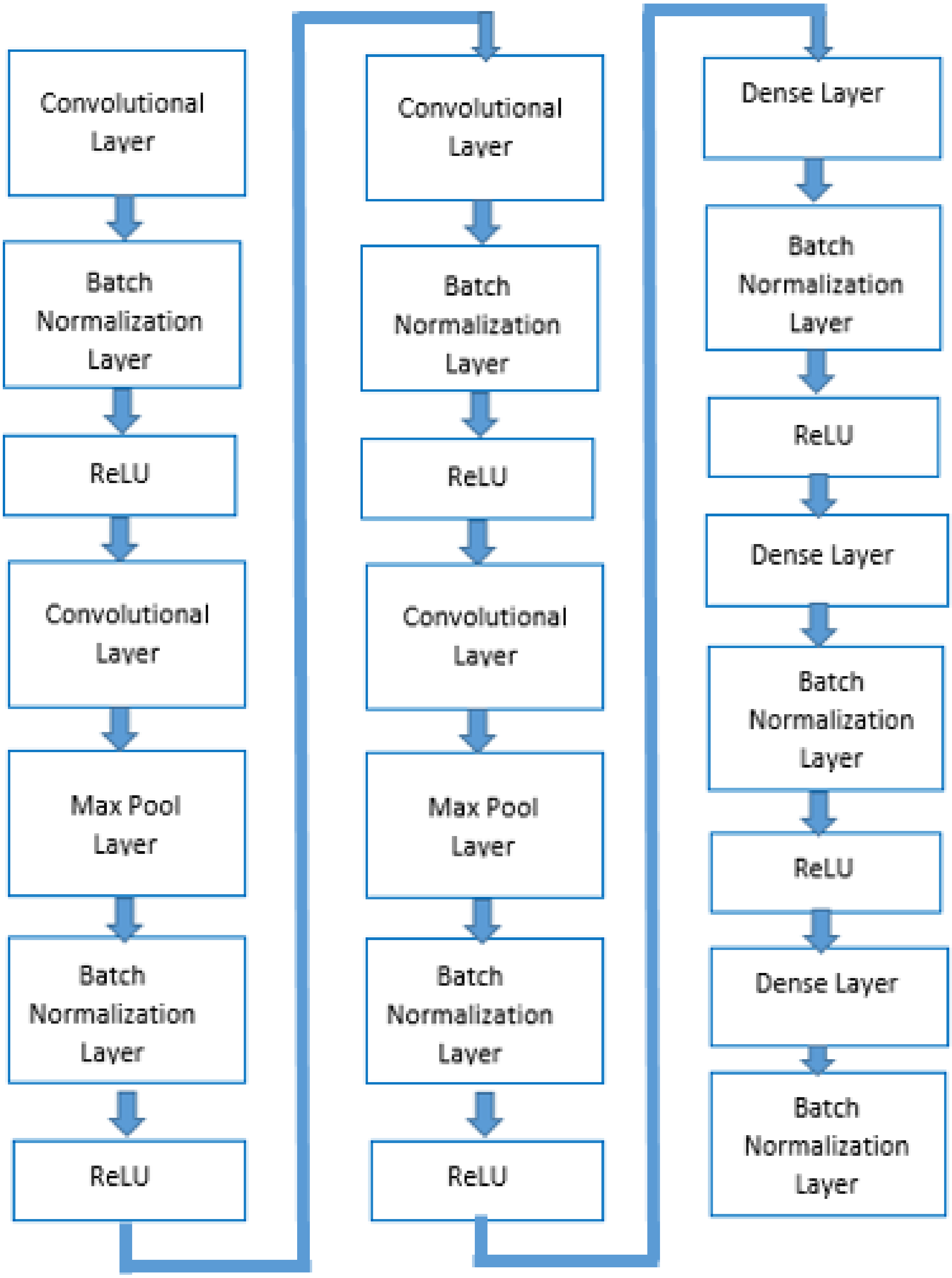


Figure 2: Binarized Neural Network Architecture

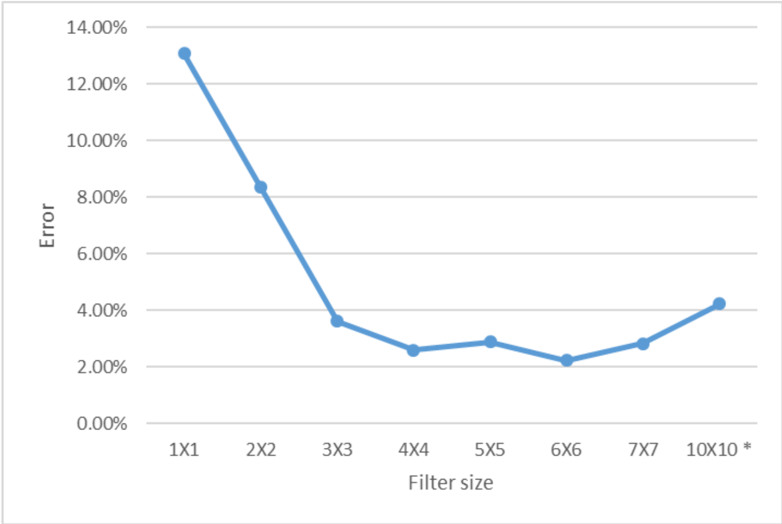
3. Results and Discussion

Srivastava et al. studied about the improvement of generalization performance by adding dropout layers on Convolutional Neural Network. The authors have not provided the exact values of the hyperparameters. On MNIST dataset, the CNN achieved an improvement in classification error from 1.60 % in baseline to 1.35% when dropout layer is used. When the CNN was trained on CIFAR10 dataset, classification error of 14.32% was achieved when dropout was added to the fully connected layers. After adding dropout to all the layers, the CNN model achieved a classification error of 12.61% [1].

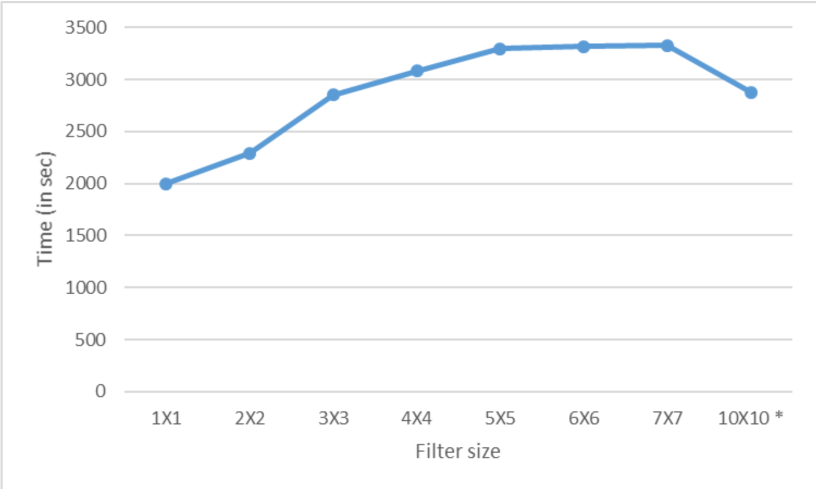
The work done by S.Park et al. shows the accuracy comparison for Convolutional Neural Network, and also after adding dropout layer to the CNN. They have also introduced two variants of dropout. For the training of CNN on CIFAR10 dataset, the model trained with 250 epochs, batch size of 128, and initial learning rate of 0.02. The classification accuracy was around 83.16% for baseline, and 87.78% with a dropout rate of 0.1. To train on MNIST, the CNN was trained for 60 epochs, batch size of 128 and initial learning rate of 0.01. It achieved a classification error of 0.604% for baseline CNN and 0.430% for the CNN having dropout layer with p=0.2 [2].

BinaryConnect [4] is a method to train neural networks with only binary weights. The binarization scheme showed near state-of-the-art results on MNIST, CIFAR10 and SVHN datasets. Courbariaux et al. [3] then later introduced Binarized Neural Networks, which showed a method to train neural networks with binary weights and activations. To the best of my knowledge, BNN is the first work which binarized both weights and activations, and achieves comparable state-of-the-art accuracy on MNIST and CIFAR10 datasets.

Filter size determines the local receptive area that slides through the image in a convolutional layer. Figure 3 shows the effect of filter size on the classification error of BNN for MNIST dataset.



(a)

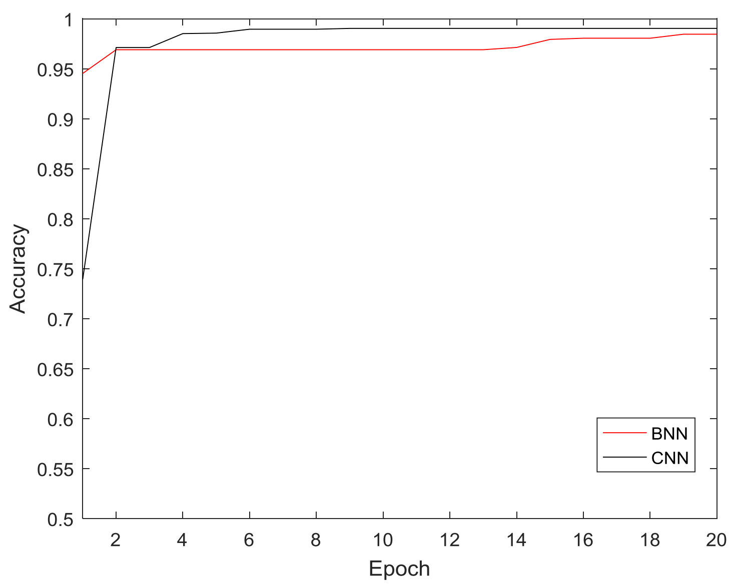


(b)

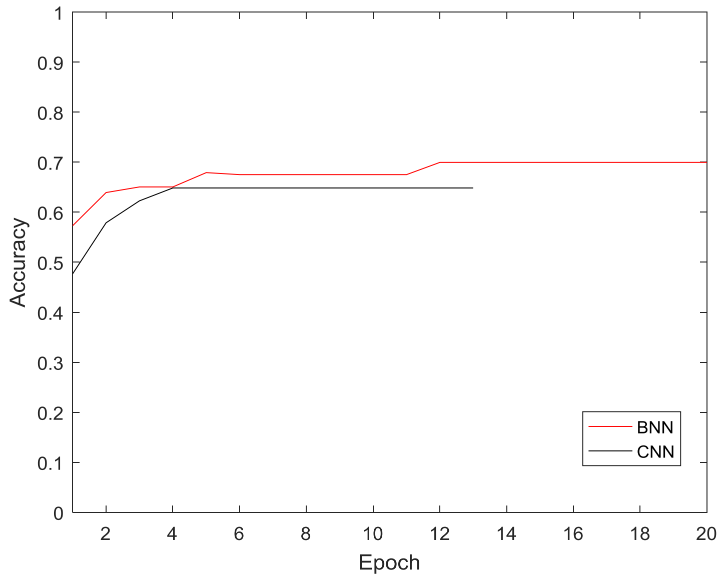
Figure 3: (a) Effect of filter size on test error using MNIST dataset (b) Effect of filter size on execution time using MNIST dataset

Figure 3(a) shows that for smaller filter sizes like 1*1 and 2*2 the test error is high as above 8%. The graph goes flat for filter sizes between 3*3 to 7*7. The filter size implemented for my experiments is 3*3 because the execution time required for filter size 3*3 is the least amongst filter sizes 4*4, 5*5, 6*6, and 7*7. This observation can be seen in Figure 3(b).

The Binarized Neural Network used in this research is built on Theano framework. Different datasets like MNIST and CIFAR10 are used to conduct experiments on the BNN. For the baseline BNN, the model is run for 25 epochs on CIFAR10 with batch size of 50, initial learning rate 0.001, and 20 epochs for MNIST with a batch size of 128, initial learning rate 0.01. BNN gives comparable accuracy like CNN and Figure 4 shows the comparison between BNN and CNN for CIFAR10 and MNIST:



(a)



(b)

Figure 4: Comparison of classification accuracy between BNN and CNN on (a) MNIST dataset, and (b) CIFAR10 dataset

The results in Figure 4 shows that the Binarized Neural Network gives comparable performance when compared to Convolutional Neural Network. The binarization of weights and activations in BNN doesn't affect the performance of the neural network, although it leads to a reduction in the computational complexity. The first epoch shows a stark difference between BNN and CNN, and BNN shows better results in the first epoch. Gradually the accuracy of both BNN and CNN becomes similar for both datasets. For both BNN and CNN, the classification accuracy for MNIST is above 95%, and for CIFAR10 it is around 70% at 20 epochs.

The effect after adding dropout layer after the convolutional layers in the baseline BNN model is studied for both the datasets. Figure 5 shows the BNN architecture for baseline+dropout where the dropout is added. Figure 6 show the classification accuracy of a BNN for baseline and baseline+dropout model for MNIST with p=0.2 and CIFAR10 with p=0.1.

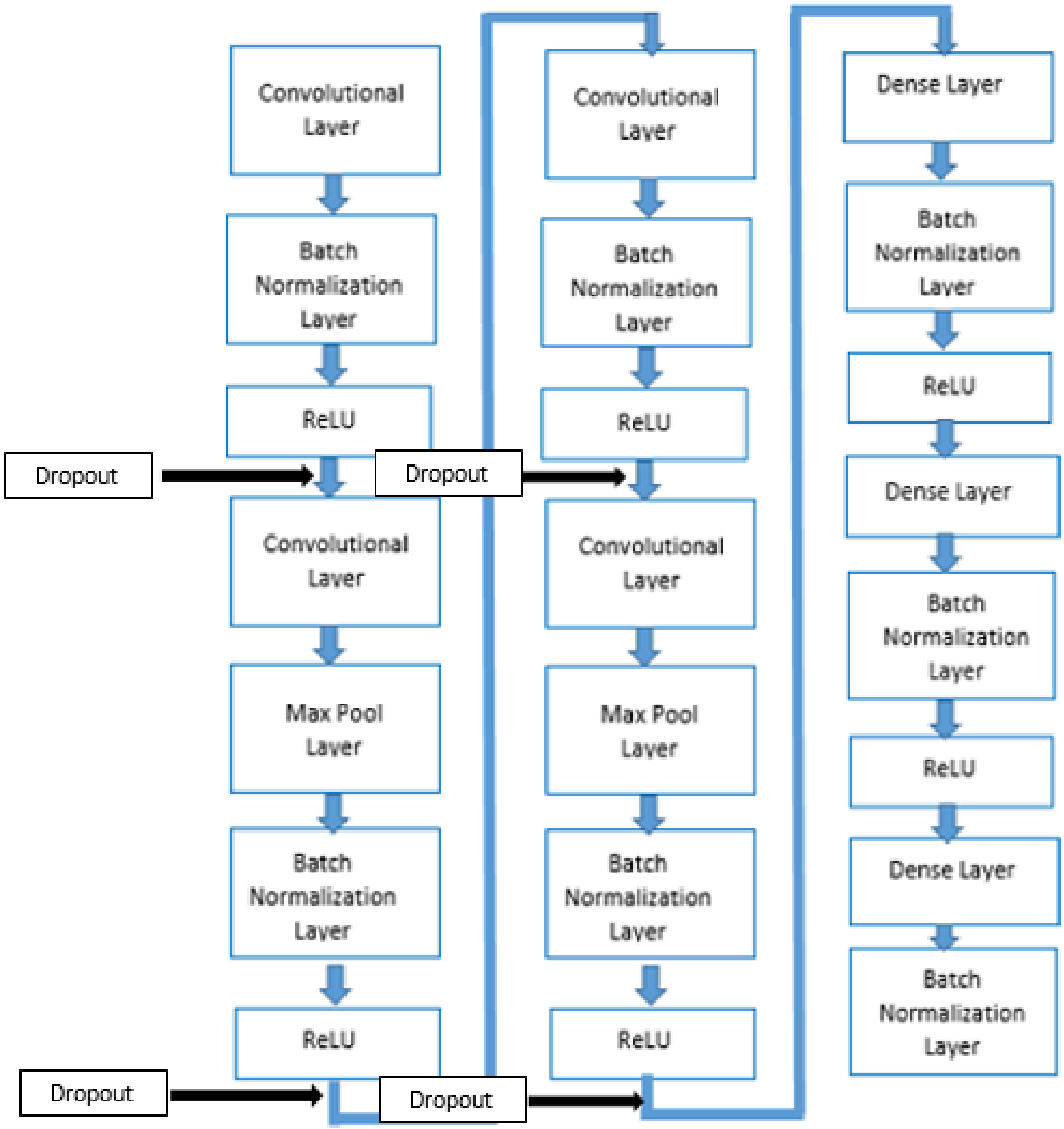
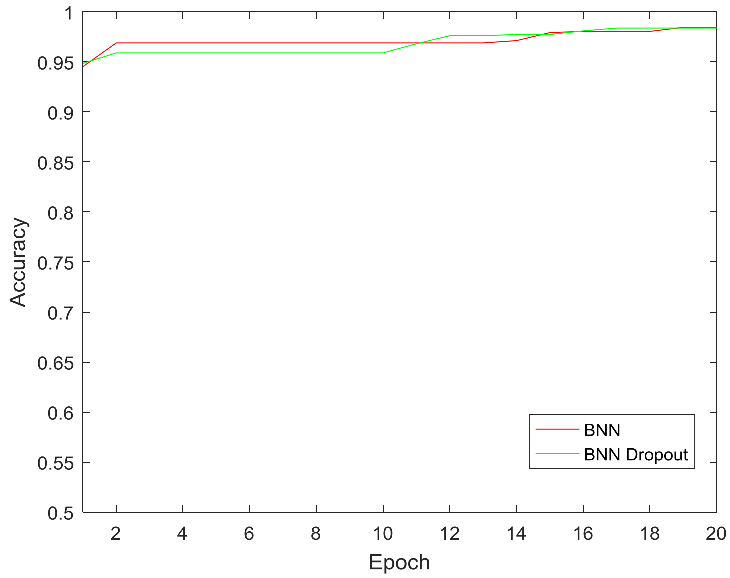
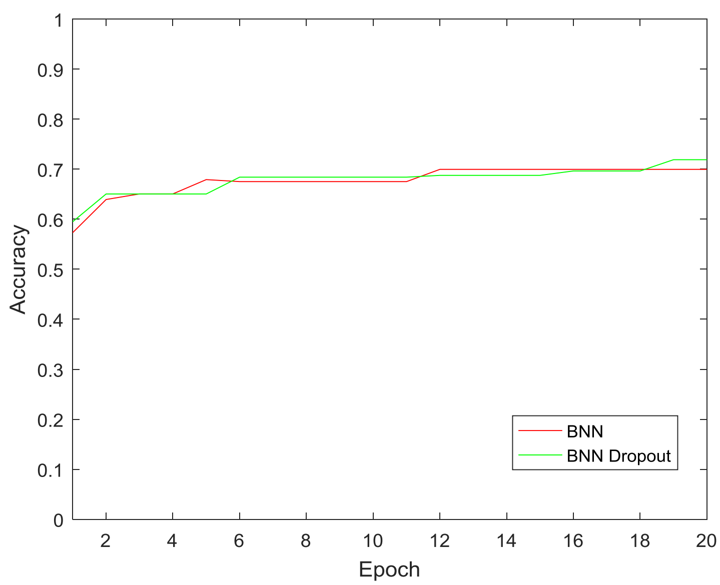


Figure 5: Network architecture of baseline+dropout showing the places where the dropout was added after the convolutional layers in BNN



(a)

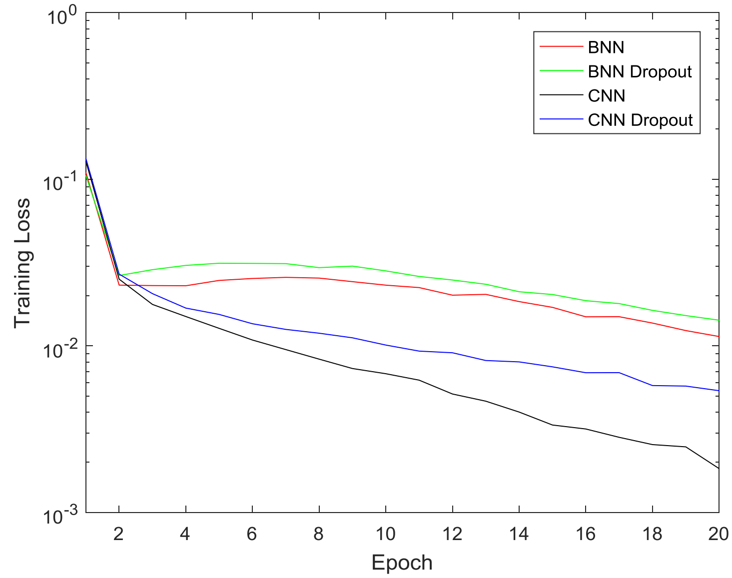


(b)

Figure 6: Shows classification accuracy of BNN after adding dropout layer to the baseline model using (a) MNIST dataset with p=0.2 (b) CIFAR10 dataset with p=0.1

Figure 7 and Figure 8 shows classification accuracy, test loss and training loss for CNN and BNN on MNIST and CIFAR10 datasets respectively:

(a)



(b)

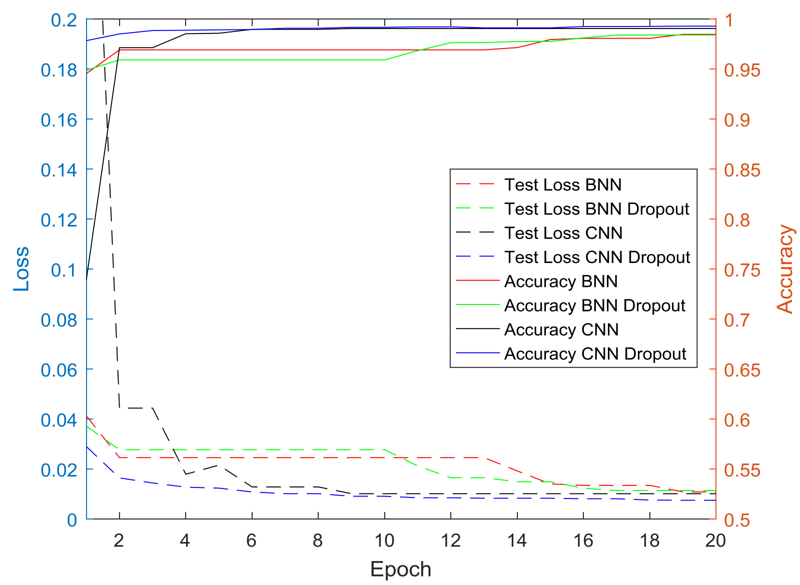
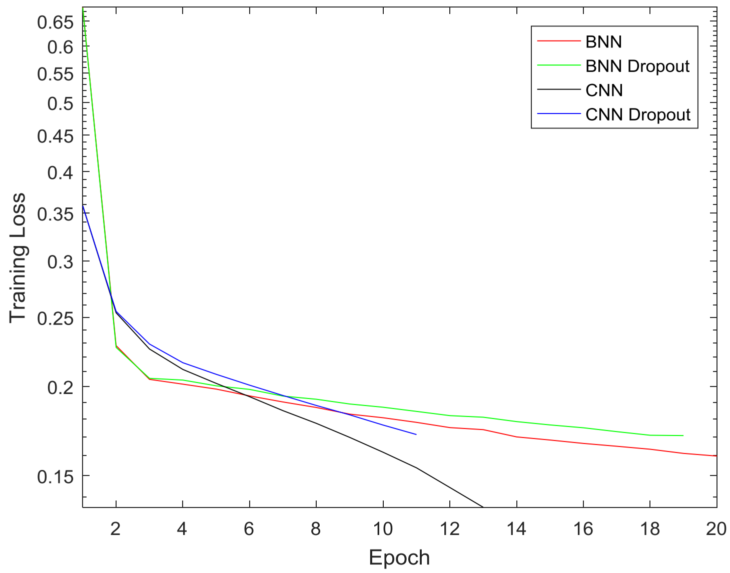
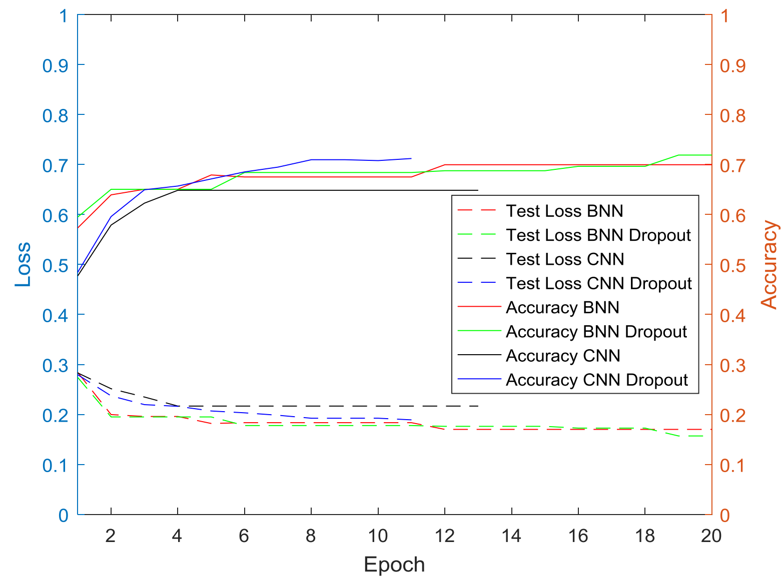


Figure 7: Shows (a) training loss (b) accuracy and test loss of BNN and CNN on MNIST dataset for baseline and baseline+dropout layer model. In Figure (b) the dotted lines represent test loss (left axis) and the solid lines represent accuracy (right axis)



(a)



(b)

Figure 8: Shows (a) training loss (b) accuracy and test loss of BNN and CNN on CIFAR10 dataset for baseline and baseline+dropout layer model. In Figure (b) the dotted lines represent test loss (left axis) and the solid lines represent accuracy (right axis)

It can be seen from Figure 6(a) that at 20 epochs the baseline+dropout model shows comparable accuracy to baseline model for MNIST dataset. There is not a notable improvement seen for MNIST dataset if dropout is added after the convolutional layers in BNN. Figure 6(b) shows that for CIFAR10 dataset, the baseline+dropout model shows better accuracy than baseline BNN model after 19 epochs.

In Figure 7(a), the graph shows that the training loss for baseline BNN and CNN shows similar training loss. The test losses and classification accuracies on a BNN and CNN for baseline and baseline+dropout models are illustrated in Figure 7(b) for MNIST dataset. It can be seen from the graph that the test loss for CNN dropout and BNN dropout is better than the baseline models for both the networks. The classification accuracies are also very close for both BNN and CNN on MNIST dataset.

Similarly, figure 8(a) shows the training loss for baseline models for BNN and CNN, and baseline+dropout models in CIFAR10 dataset. The losses are very similar for both the models. Figure 8(b) illustrates the test losses and accuracies for baseline BNN and CNN, and baseline+dropout models for BNN and CNN for CIFAR10. The classification accuracy of CNN dropout is better than baseline CNN, and accuracy for BNN dropout is better than baseline BNN after 20 epochs. The accuracy for BNN and BNN Dropout lies between the accuracies of CNN and CNN Dropout, thus showing comparable performance of BNN and CNN.

Dropout is a useful feature in BNN, and the following discussion and graphs will closely examine how dropout affects the neural network. The size of a training set will determine how the dropout affects the BNN. The training dataset sizes are varied and given the following values of 50k, 10k, 5k, 1k, 500, and 100. These images are randomly chosen from the MNIST dataset. Figure

9 shows the effect of dataset size on classification error when dropout is added to BNN with batch size of 128, dropout rate 0.2 and initial learning rate of 0.01.

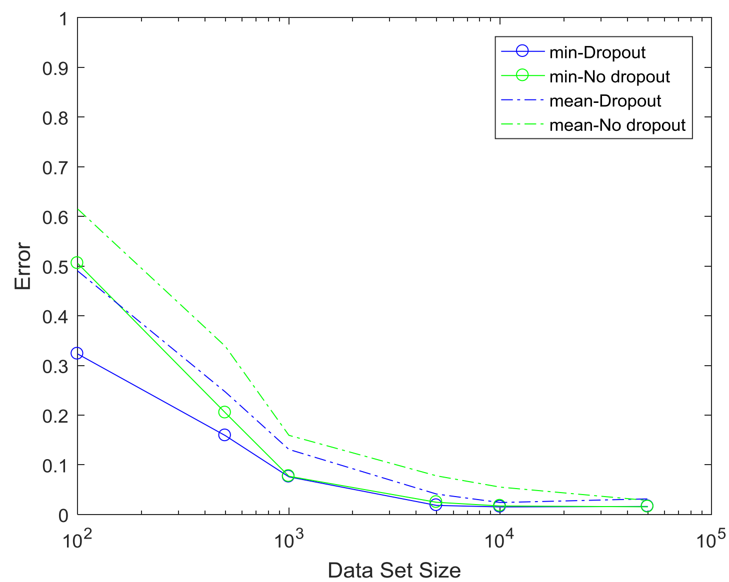


Figure 9: Shows the variation of classification error with different data set size given as an input to the BNN using MNIST dataset, p=0.2. Solid lines represent minimum of classification error for 20 epochs and dotted lines represent mean of classification error for those 20 epochs.

A good regularizer is defined as one which gives good generalization error from models having large parameters while being trained on small data sets. In Figure 9, it can be seen that the dropout model works better for small data sets 100, 500, 1k, 10k. It can be observed from Figure 8 that the effect of adding dropout in BNN after convolutional layers is observed to be strong when the data set size is small. For small datasets, the results for BNN is unlike CNN, where the classification error in CNN is higher if dropout is added to CNN using small dataset size (upto 500) [1]. In BNN, upto 1000 dataset size, the performance is better with dropout layer. Above 1000 dataset size, CNN gives better performance when dropout is added, but in BNN the performance for dropout and no dropout are very similar. The addition of dropout for small datasets adds noise which helps in reducing overfitting on the datasets.

The probability of retaining units in a neural network is called the dropout rate, and varying the dropout rate will affect the performance of the BNN. The dropout layer is added after the convolutional layers in BNN as shown in Figure 5, and the p is varied with the values 0.1, 0.2, 0.4, 0.6 and 0.8. Figure 10 shows the classification accuracy for the different dropout rates on MNIST with batch size 128 and initial learning rate 0.01. The number of layers in the BNN is held constant for this experiment.

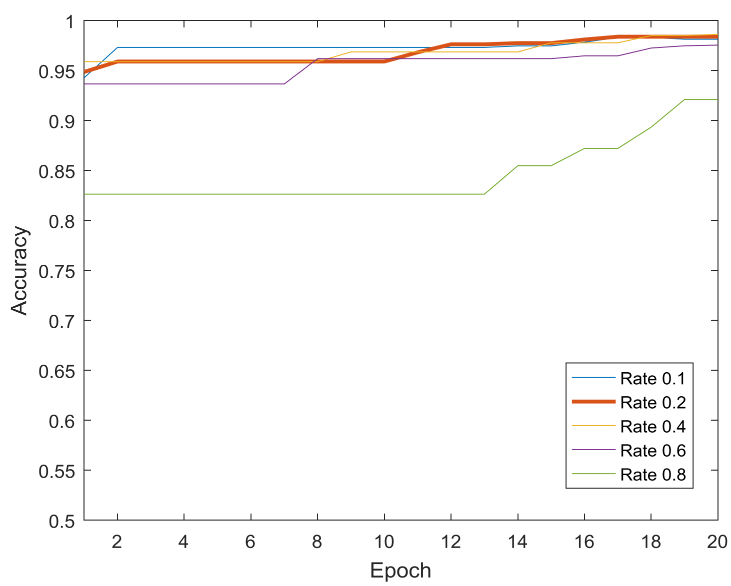


Figure 10: Shows the trend of classification accuracy for different dropout rates on MNIST. Epochs=20

Figure 10 shows that the accuracy for dropout rate 0.8 is poor than the rest of dropout rate values. For $0.1 \leq p < 0.6$ the performance is very similar when the BNN is trained for 20 epochs. For dropout rate 0.6 the accuracy is comparable with $p<0.6$, although it shows a little bit lesser classification accuracy.

Figure 11 shows the test error as a function of the dropout rate p. It shows the effect of dropout rate on the classification error. The experiment is run on MNIST dataset for 20 epochs, with dropout after convolutional layers with p=0.2. The same network architecture is used for the experiment with different amount of dropout rate.

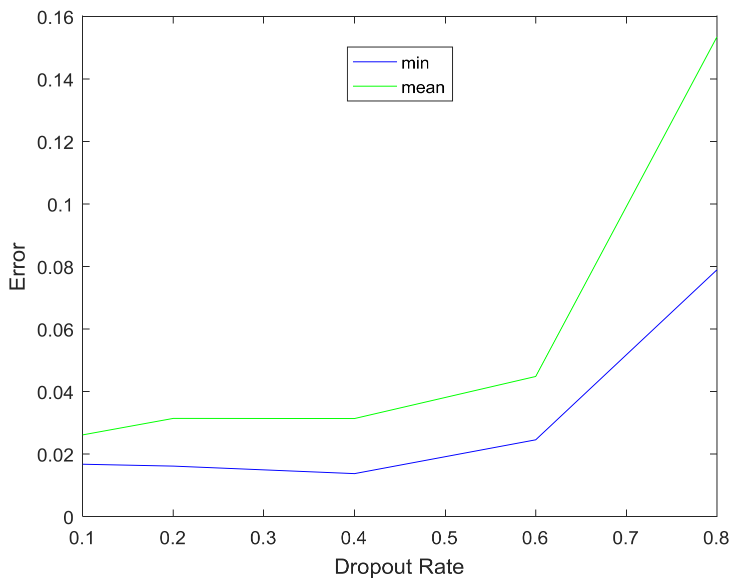


Figure 11: Effect of changing dropout rates on MNIST when number of layers is fixed. The curve in green represents the mean of the test errors and the blue curve represents the minimum of the test errors for different dropout rates.

It can be observed from Figure 11 that when the dropout rate is low as 0.1, the classification error is not poor. This result is unlike the one performed using CNN on MNIST [1], as in CNN the test error was higher for small dropout rates like 0.1 and 0.2. For BNN, as the p increases the error goes down until 0.4. The error is comparable and becomes flat when $0.1 \leq p \leq 0.6$, and then error increases as p becomes close to 1.

Regularization can be added also after the fully connected layers, and Figure 12 shows the architecture when the dropout layer is added. White boxes represent the places where dropout was added after convolutional layer along with its p values. Green boxes represent the architecture when dropout is added after dense layer and that also shows its p values. Orange boxes represent the network architecture when different p values are implemented for lower and higher layers in BNN. For adding dropout after all layers, white and green boxes together represent the architecture

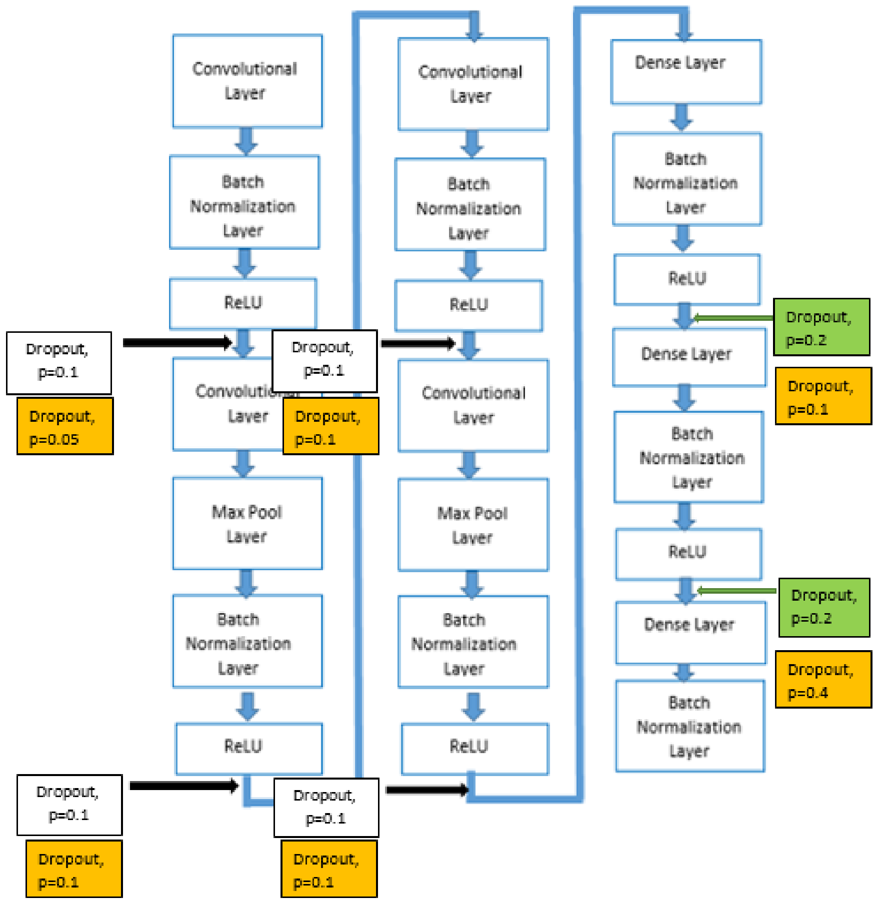


Figure 12: Network architecture after adding dropout layers. White boxes represent the places where dropout was added after convolutional layer along with its p value. Green boxes represent the dropout after dense layer architecture that also shows p value. Orange boxes represent the network architecture when different p value is implemented for lower and higher layers in BNN. For adding dropout after all layers, white and green boxes together represent the architecture

Figure 13 shows the effect of adding dropout to convolutional layer, full connected layers, and after both convolutional and fully connected layers. The experiment is also run with different dropout rate for lower (p=0.05) and higher dropout layers (p=0.4) which is represented as Different Regularization in the graph.

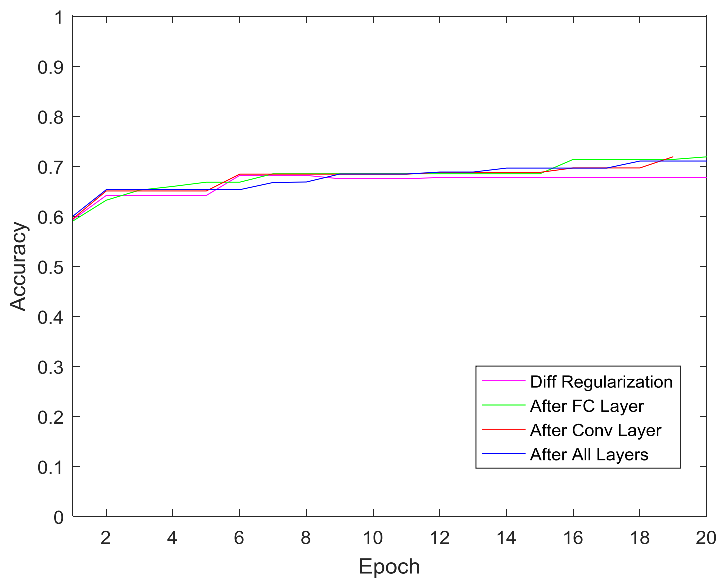
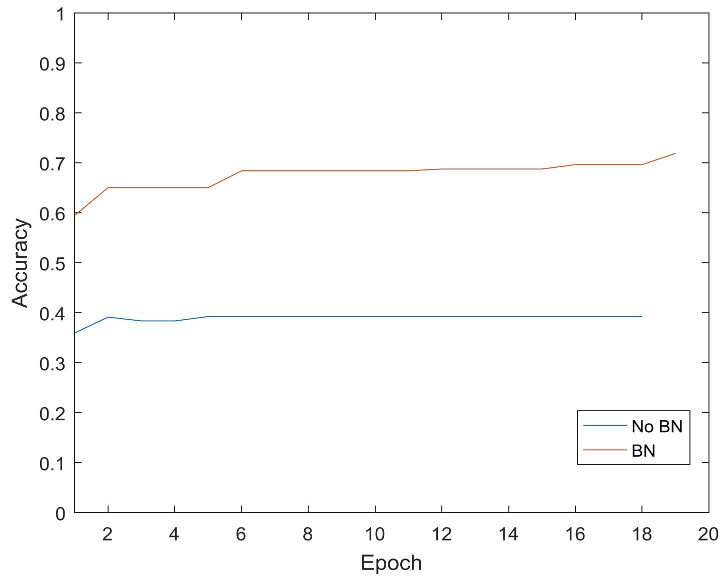


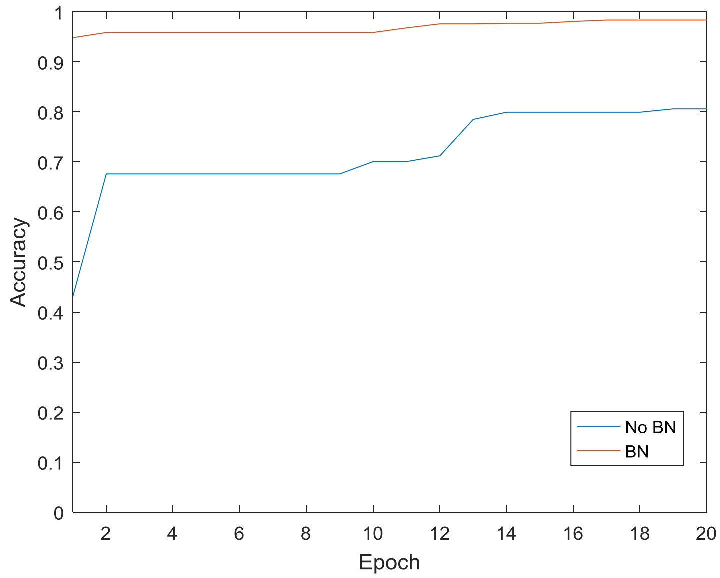
Figure 13: Effect on classification accuracy of adding dropout after Convolutional Layers, Dense Layers, Convolutional+Dense layers, and varying dropout rates to the dropout layers added after Convolutional+Dense layers. Different regularization represents the experiment where different dropout rates are implemented in lower and higher layers in BNN.

Figure 13 shows that the addition of dropout after convolutional layers, dense layers, and convolutional and dense layers are comparable. It can be observed that addition of dropout layer after convolutional layers and fully connected layers give comparable results. After doing different regularization experiment, the accuracy is a little bit less after 20 epochs when compared to other results. When dropout is added after all the layers, no significant improvement was observed in the classification accuracy of the BNN.

The batch normalization layer in a BNN improves the performance of a BNN by normalizing the activations of a layer in the network. The dropout layer was added to the BNN after the Batch Normalization is performed in the network. This combination gave better results as both improves generalization performance. Figure 14 shows the effect of adding batch normalization to the BNN's performance.



(a)



(b)

Figure 14: Shows the performance of the BNN with baseline (with batch normalization+dropout) and when Batch normalization is removed from the baseline (with batch normalization+dropout) on (a) CIFAR10 dataset (b) MNIST dataset

Figure 14(a) shows the observation that when batch normalization is removed from the BNN using CIFAR10 dataset, the classification accuracy drops from 70% to 40% after 20 epochs. The results are best when batch normalization is added with dropout layer. Similarly, from Figure 14(b) it can be observed that using no Batch Normalization layer with dropout layer decreases the classification accuracy from 98% to 82%.

4. Conclusion

The experiments conducted on Binarized Neural Network showed comparable performance of BNN and CNN. Datasets like CIFAR10 which is a dataset containing natural images, and MNIST which consists of handwritten digits were used to perform the experiments. The comparison showed that it is possible to train Binarized Neural Network on MNIST, CIFAR10 and achieve near state-of-the-art results.

The effect on the performance in BNN by adding dropout layer is thoroughly studied. The research demonstrated that addition of dropout layer to a BNN might improve the BNN's performance. One of the future scope can be to conduct random repetitive training to get definitive results. The effect of adding dropout in BNN after convolutional layers is observed to be stronger when the data set size is small. This shows similar results as observed by S. Park et al. on Convolutional Neural Network. The dropout rate controls the intensity of dropout, and it was observed that comparable BNN performance is found for dropout rate $0.1 \leq p \leq 0.6$. It was also observed that the best performance was achieved when Batch Normalization was implemented with dropout. This is because Batch Normalization further assists dropout in providing better generalization.

On adding dropout after dense layers and convolutional layers, it was observed that the classification accuracies were comparable. It was observed that adding dropout in convolutional and fully connected layers increased training time, but didn't improve the BNN performance notably. Hence, like CNN the effect of dropout on convolutional layer is observed to be at par with adding dropout layer after full-connected layers in BNN.

It was observed from the experiments that the training time increased when dropout was added in BNN. A reason for this might be that because of the noise addition, each training is done in a new random architecture. The gradients for the new architecture are not related to the gradients of the previous architecture. Hence it takes longer time to train, but it does reduce overfitting. In the experiments, it was observed that the BNN dropout architecture takes around 1.5 times the training time needed for the architecture without dropout. This value is less than the value reported by Srivastava et al. for CNN, and the reason for this might be because binary weights take lesser time to train. Another observation noted in the experiments is that the Binarized Neural Networks are slower to train, but they are nearly as accurate as floating-point CNNs.

In this research, an attempt was made to study the change of number of filters on execution time. I observed that the number of filters have an impact in a network with m layers. The n filters per layer will take $O(mn)^2$ runtime. Each of the n filters in layer i has to be computed by looking at each of the n filters in layer i - 1, so there's an n^2 dependency.

In the future, research can be extended to working with other different datasets like SVHN, ImageNet etc. Efforts can be made to reduce the training time after adding dropout layer to the BNN. Data preprocessing is also a future scope of this research as to how will it affect the accuracy of BNN with dropout layer for different datasets. Studying and implementation of adding different dropout rate in all the layers can be done in detail in the future. The effect on computational complexity due to the dropout layer on BNN can be studied, since reduced computational complexity will conclude that addition of dropout layer to BNN is advantageous.

5. Acknowledgement

The author would like to acknowledge Dr. Zhuo Feng (Associate Professor) from Michigan Technological University and Dr. Ian Goodfellow (Staff Research Scientist) from Google Brain for their support and guidance during the entire research.

6. References

[1] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1929–1958 (2014)

[2] Park, Sungheon & Kwak, Nojun. (2017). Analysis on the Dropout Effect in Convolutional Neural Networks. 189-204. 10.1007/978-3-319-54184-6_12.

[3] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Binarized Neural Networks. In Advances in Neural Information Processing Systems 29. 4107–4115.

[4] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. 2015. BinaryConnect: Training Deep Neural Networks with binary weights during propagations. In Advances in Neural Information Processing Systems 28. 3123–3131.

[5] Goodfellow, I., Bengio, Y. and Courville, A. (2017). Deep learning. Cambridge, Mass: The MIT Press.

[6] Blog, Guest, et al. "The Evolution and Core Concepts of Deep Learning & Neural Networks." Analytics Vidhya, 6 Oct. 2016, www.analyticsvidhya.com/blog/2016/08/evolution-core-concepts-deep-learning-neural-networks/. Accessed 10 Sept. 2017.

[7] Michael A. Nielsen, "Neural Networks and Deep Learning", Determination Press, 2015

[8] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. 2016. XNOR-Net: ImageNet Classi€ cation Using Binary Convolutional Neural Networks. In Computer Vision - ECCV 2016. 525–542

[9] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. 2016. DoReFa-Net: Training Low Bitwidth Convolutional Neural Networks with Low Bitwidth Gradients. arXiv:1606.06160 [cs] (2016).

[10] Yang, Haojin & Fritzsche, Martin & Bartz, Christian & Meinel, Christoph. (2017). BMXNet: An Open-Source Binary Neural Network Implementation Based on MXNet. .

[11] Tang W., Hua G., and Wang L. 2017. How to Train a Compact Binary Neural Network with High Accuracy? In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2625–2631.

[12] Docs.aws.amazon.com. (2017). What Is Amazon EC2? - Amazon Elastic Compute Cloud. [online] Available at: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html [Accessed 10 Sep. 2017].

[13] Cs231n.github.io. (2017). CS231n Convolutional Neural Networks for Visual Recognition. [online] Available at: http://cs231n.github.io/convolutional-networks/ [Accessed 16 Sep. 2017].