Using Convolutional Neural Networks to perform classification on MNIST dataset images.

Rahul Joshi
ISME department
Wichita State University
Wichita, USA
rxjoshi1@shockers.wichita.edu

Edgar Chavez
Computer Science department
Wichita State University
Wichita, USA
erchavez1@shockers.wichita.edu

***Abstract*—In this paper, three different types of neural networks are presented and used based on classical computing, quantum computing and hybrid computing to perform classification task on MNIST dataset of hand-written images. Firstly, a classical CNN is used where image preprocessing is used to eliminate unrelated information in hand-written images, which could help to locate numbers and then feature extraction by using convolutional layer and finally a fully connected layer along with soft-max layer is used for classification. Secondly, a quantum convolutional neural network employs two qubit gates, with the readout qubit always acted upon. The keras model in python is built using quantum components and is fed with quantum data that encodes the classical data. Thirdly, hybrid CNN uses the quantum circuit as neurons for our network. It feeds outputs from the classical neural nodes fed as the input vector to the quantum circuit.**

Keywords—CNN, MNIST dataset, quantum computing, neural network.

I. Introduction

Now-a-days, most of the complex and large problems are solved using classic computation on powerful computers and supercomputers. But as the complexity of problems increase, the traditional algorithms to solve large and complex problems like travelling salesman problem takes exponentially large amount of time and computation power using classical computation. Quantum computation is new technology which can solve highly complex problems in much less time than classic computation. Quantum computation creates multidimensional spaces where patterns linking individual data points can be found. Image classification is a complex procedure and can be solved efficiently by large neural networks, especially CNNs. But, as the data collection keeps on increasing exponentially with advancement of machine learning technologies, we may need to look into alternatives of classical neural networks and using quantum computation with deep learning technology can be one of them. In this paper, we compare classical classification to quantum and hybrid classification algorithms for their computational strength.

Identify applicable funding agency here. If none, delete this text box.

II. Deep learning

A. Basic Understanding

Deep Learning is considered as a sub-field of machine learning along with the other sub-field of traditional machine learning. It is a machine learning technique which consists of an artificial neural network mimicking the structure of human brain. The neural network learns the data feature hierarchies where higher-level hierarchy features are formed using combination of multiple lower-level features. Using the learned feature hierarchies, the neural networks learn complex functions which maps the input data to the available output labels. The learning of the complex functions by the neural networks is an iterative process where the model (consisting of complex functions) gradually improves over iterations.

B. Architecture of Neural Networks

The architecture of a neural network generally consists of an input layer which is used to feed the predicting features, one or multiple hidden layers responsible for the bulk of computation in the network and an output layer which gives out final response. Every layer in the neural network contains neurons. The number of hidden layers and neurons in hidden layer is a factor of the size of the data. As the data size increases, additional hidden layer helps in explaining the complex variability in the data. The input layers contain the same number of neurons as the number of input features. The number of neurons in each hidden layer depends on the size of data as well as the complexity of the data. The number of neurons in the output depends on the desired output type and is equal to required number of outputs, for example, one neuron is required for regression while two neurons will be required for binary classification model and five neurons for a multi-class classification model with five classes. Each neuron in the hidden and output layers are associated with respective parameters (weight and bias) and an activation function. The parameters are used to make computations in hidden and output layers, and these are randomly initialized at the start of deep learning algorithm. The activation function determines whether the neuron should be activated or not and is also responsible for introducing non-linearity in the model to help explain the complexity of data. A simple neural network with two input

features (x1,x2), a hidden layer with two neurons and respective parameters and an output layer with one neuron with its own parameters is represented below. The weights can be considered as a vector of two elements and bias can be considered as a scalar entity. Along with the weights and biases, there is a activation function at both hidden network neurons and at the output neuron.

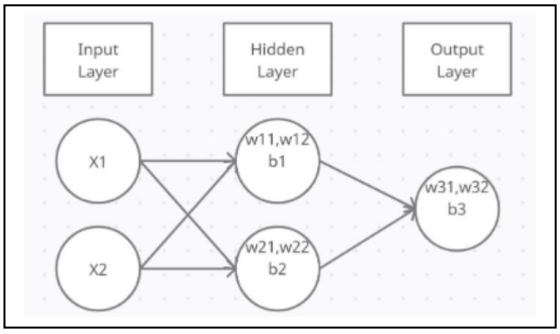

Figure 1. Neural network example.

C. Convolutional neural networks

Convolution neural networks are neural networks which are used to analyze visual imagery. CNN is generally used for image classification purposes. The input of these networks are images (matrices of pixel values) and the output neurons represents the probability of particular image class. The weights and biases help learning special features (for example, edges, points etc.) at each hidden layer neurons in input images which are used to classify the images. They generally have three type of layers- convolutional layer, pooling layer and fully connected layer. The convolutional layer is the core building block of a CNN, and it is where most of the computation occurs. It requires a few components, which are input data, a filter, and a feature map. CNN also has feature detector which is a two-dimensional array of weights, also known as a kernel or a filter, which will move across the receptive fields of the image, checking if the feature is present. This process is known as a convolution. When applied to an image, the filter calculates the dot product between input pixels and itself which is then fed into output array. Similarly, multiple dot products are calculated as the filter slides by a fixed stride until it is swept through entire image. The final output of the convolution step is called as feature map. Padding such as valid padding, same padding & full padding can be used when filters do not fit the input image. If the images have complex features, multiple convolution layer should be used. Next, the pooling layer conducts dimensionality reduction, reducing the number of parameters in the input. It operates similar to convolutional layer by sweeping through the entire image but unlike convolutional layer, it does not have any weights and instead applies an aggregation function to the values within the receptive field, populating the output array. Pooling can be of two types- max pooling & average pooling. Convolutional and pooling layers generally uses ReLu activation functions. Finally, the fully connected layer (each node in the output layer connects directly to a node in the previous layer) performs the task of classification based on the features extracted through the previous layers and their different filters Fully connected layer generally uses soft-max activation function to classify inputs.

III. QUANTUM COMPUTATION

Quantum computation is emerging technology which makes use of quantum mechanics to solve problems too large and complex for classical computers. Quantum algorithms performs calculations based on the probability of an object's state before it is measured. The information is coded as quantum bits or qubits which can exist in superposition. As quantum computer can contain multiple states simultaneously, it can potentially store an exponential time of data compared to today's classical computing of powerful supercomputers. Qubits can either be in state of superposition which is combination of all possible configurations, and a bunch of super positioned qubits can help create multi-dimensional computational space. Or they can be in entanglement state which is correlated to behavior of two separate things and any change to one qubit affects the other. The entanglement state can be used to solve complex problems.

Quantum CNN can be used to reduce the complexity of CNN to O(log(N)) variational parameters for input sizes of N qubits which can allow for efficient training and implementation on realistic, near-term quantum devices.

IV. CLASSIFICATION OF MNIST IMAGES

The MNIST dataset is an acronym that stands for the Modified National Institute of Standards and Technology dataset. It contains 60,000 small 28 x 28 pixel grayscale images of handwritten single digits between 0 & 9. It also contains 10,000 images as a testing set. MNIST dataset is generally used as a sources for researchers and learners for machine learning projects.

Figure 2. Sample images from MNIST dataset.

A. Classical convolutional neural network

Firstly, before any classification task can be employed on the images, we prepared the data by loading it through keras api on python. The loaded dataset contains training and testing set with its response labels. Once the images are loaded, we convert the categorical labels into vector format to assist model formation. For example, label for digit '1' should be converted to

[0,1,0,0,0,0,0,0,0,0].For the classification of MNIST images using classical CNN, we employed the architecture where there are two convolution layers with ReLu activation followed by pooling layer, a fully connected layer with soft-max activation function. The two convolutional layer are used so that one filter will help in detecting straight lines and the other filter will help with curved lines. Pooling is of max pooling type which calculates the maximum value for patches of feature map. Further, we added a dropout layer, so that we avoid any potential over-fitting and high variance problem.

The CNN model is compiled with categorical cross-entropy loss function because it is a problem with two or more output label class, and we use AdaDelta as the gradient descent for optimization of the model weights and bias. We selected the batch size as 128 and epoch was set at 50. After the training process is completed, we achieved pretty good accuracy of 87.1%. By increasing the number of epochs, we could have achieved higher training accuracy, but we decided against it to respect the time constraints. It took a total of 2,762 seconds to train on a 16 gb, 7th generation i7 processor laptop. The trained model was tested on the testing dataset and the classification accuracy was found to be 92.10% which is pretty high and does not indicate any over-fitting of the model.

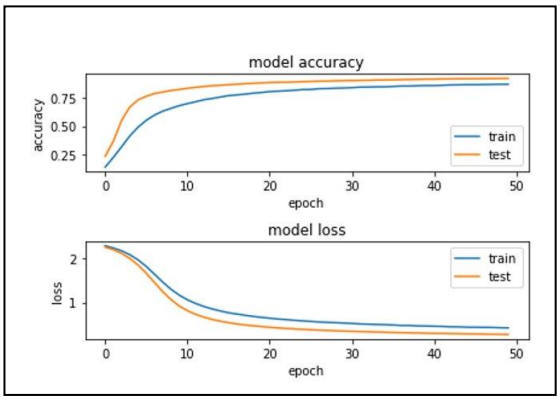

Figure 3. Training and testing accuracies.

B. *Quantum convolutional neural network*

Quantum Convolutional Neural Network (QCNN) is developed from CNN but it's based is on quantum computing. QCNN extends the main features of and structures of existing CNN to quantum systems. The model of QCNN applies the convolution and pooling layer, the main features of CNN, to the quantum systems.

C. *Quantum Convolutional Neural Network for Image Classification*

Image classification is widely used with neural networks, for our study we used the MNIST database for handwritten images. Once the images are loaded, we begin by rescaling the images from a range of 0 to 255 to a range of 0 to 1. The MNIST data we used only consisted of the numbers 0 and 1. A function was used to filter the data and keep only the numbers 0 and 1. Before we could begin to implement the QCNN with a large amount of data, we decided to resize our data to 2 x 2 pixels as well as remove any contradicting data. Next a threshold of 0.5 was created to approximate our pixel values to a zero or a one depending on if the original pixel value was less than or greater than 0.5. Once this is complete a cluster state is created as the data processing environment for or 2 x 2 pixels which outputs 4 qubits with H gates and control z gates. With this data we then can transform it into a data circuit where pixels with a value of 1 will have an x gate added while a pixel with a value of 0 will get no gated added. Once this data is converted to a circuit, we must also transform them into a tensor format.

Once we have our circuit, we begin the construction of the quantum circuit which is broken down into three components. The first component needed is called a one qubit unitary circuit which involves the x, y, and z gates. The second component is a two-qubit unitary circuit and lastly a two qubit pooling circuit is used to reduce the entanglement to one qubit. These components come to form a circuit to help build our quantum layer. In our quantum layer we have the quantum convolution layer, quantum pooling layer and quantum network layer. In our quantum convolution layer we apply the two qubit unitary to all pairs of qubits (hidden state). The quantum pooling layer then works to reduce the size of our circuit from two qubits down to one qubit. This process can be repeated until the quantum system is sufficiently small enough. Once this process is complete we have our quantum layer and can go on to arrange our Quantum Convolutional Neural Network that consists of quantum convolution layer with four qubit inputs, the quantum pooling layer with four qubit inputs and the quantum neural network layer with four qubit inputs.

D. *Quantum Convolutional Neural Network Training*

Our training for the QCNN model involved only using 5 epochs and 500 data samples for a relatively fast training. Our highest accuracy percentage was 78% which could potentially increase with a higher number of epochs and a large data sample. In the figure below is the training and validation for the quantum cnn model.

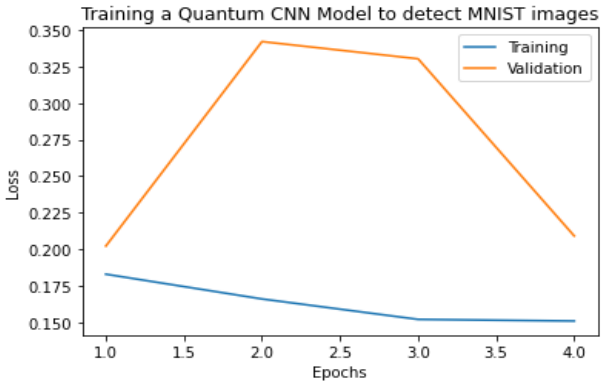

Figure 4. Quantum CNN training accuracy.

E. *Quantum-Classical Convolutiona Neural Network*

The quantum and classical convolution neural network is considered to be a hybrid model in the case that a quantum parameter is inserted into a classical neural network along with other classical parameters thus forming the hybrid model. For our study we used the previously created quantum layer but afterwards sent these results into a classical neural network.

F. *Quantum-Classical Convolutional Neural Network Training*

We trained the hybrid with the exact same conditions as the QCNN model for purpose of comparing the two models. We used 5 epochs and 500 data samples, and we were able to get an accuracy of 77% slightly lower than the QCNN model below is the figure for the training and validation of the hybrid model as well as the figure for the comparison between QCNN and hybrid models

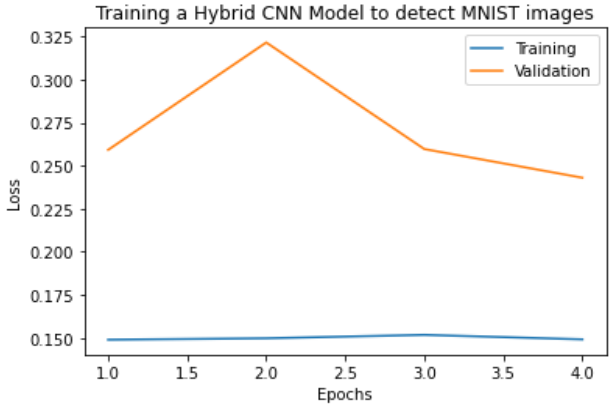

Figure 5. Hybrid CNN training accuracy.

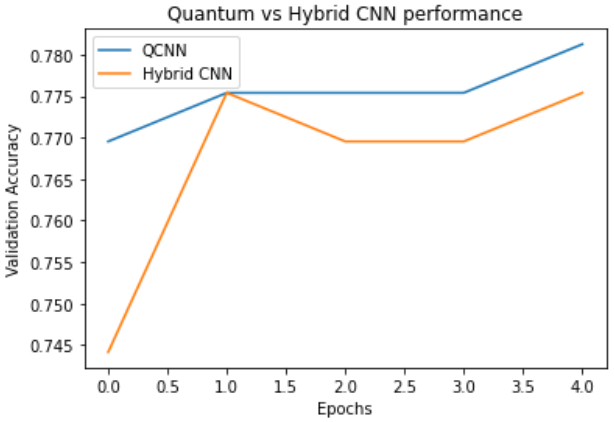

Figure 6. Quantum CNN vs Hybrid CNN performance.

V. Conclusion

The classification task for MNIST dataset images is not a very complex process and all three types of CNN- classical, quantum & hybrid are equally well adept at performing the classification. Although for this study we only trained quantum and hybrid CNN for 5 epochs when compared to 50 epochs of classical CNN due to processing limitations, we observed they achieved around 78% and 77% accuracy respectively which is better than classical at 5 epochs. So, the pure quantum CNN was the best at convergence rate while training followed by hybrid CNN and then classical CNN. However, accuracy wise, all three algorithms should be comparably similar. Although, the difference between the three algorithms are not very large, they would certainly be very different for image classification of higher resolution and RGB format images. We estimate that the quantum CNN will perform significantly better than classical CNN at image classification when assessed using computation time.

For future research, we should perform image classification for colored higher resolution images for same number of epochs on a same computer, so that much fairer comparison between classical, quantum and hybrid CNN can be made successfully.

References

[1] By: IBM Cloud Education. (n.d.). *What are convolutional neural networks?* IBM. Retrieved December 9, 2021, from https://www.ibm.com/cloud/learn/convolutional-neural-networks.

[2] Fisher, C., 2021. *IBM | What is Quantum Computing?*. [online] IBM Quantum. Available at: <https://www.ibm.com/quantum-computing/what-is-quantum-computing/> [Accessed 9 December 2021].

[3] Medium. 2021. *A simple 2D CNN for MNIST digit recognition*. [online] Available at: <https://towardsdatascience.com/a-simple-2d-cnn-for-mnist-digit-recognition-a998dbc1e79a> [Accessed 9 December 2021].

[4] Medium. 2021. *A simple 2D CNN for MNIST digit recognition*. [online] Available at: <https://towardsdatascience.com/a-simple-2d-cnn-for-mnist-digit-recognition-a998dbc1e79a> [Accessed 9 December 2021].

[5] Medium. 2021. *A simple 2D CNN for MNIST digit recognition*. [online] Available at: <https://towardsdatascience.com/a-simple-2d-cnn-for-mnist-digit-recognition-a998dbc1e79a> [Accessed 9 December 2021].

[6] Gavali, P. and Banu, J., 2019. Deep Convolutional Neural Network for Image Classification on CUDA Platform. *Deep Learning and Parallel Computing Environment for Bioengineering Systems*, pp.99-122.

[7] Poonia, R. and Kalra, M., 2016. Bridging approaches to reduce the gap between classical and quantum computing. *Journal of Information and Optimization Sciences*, 37(2), pp.279-283.

[8] Farhi, Edward & Neven, Hartmut. (2018). Classification with Quantum Neural Networks on Near Term Processors.