Image classification using a quantum ResNet18 hybrid convolutional network

Vivek Patil
Buffalo, NY
vivekp@buffalo.edu

Abstract

Quantum Computers have recently been seen as a new platform for deep learning applications. If these current NISQ era quantum computers have any quantum advantage over the classical deep learning algorithms is a question yet to be answered. In this project I try to compare the classical ResNet18[2] image classification algorithm with classical-quantum hybrid algorithm on a pet classification dataset and analyze if there is any quantum advantage with a hybrid architecture over the former. The setup includes training the original ResNet18 versus training the same original model by removing the final fc layer and replacing it with a variation quantum ansatz, which is a subroutine consisting of sequence of gates applied to specific qubit wires, like set of layers of a neural network for classification of 8 different kinds of pets with the help of 6 qubits.

1 Introduction

Quantum Computing has been a point of interest for machine learning enthusiasts since early 2000s. With recent developments in the quantum computing space, researchers were able to develop variational quantum circuits, which could replicate the working of a neural network. Quantum variational circuits are used in the quantum neural network models, where these circuits are also called as ansatz. We train these ansatzes in the same way as we would train a neural network model. ResNet18 is known for effectively classify 1000 object categories on a pretrained network trained on ImageNet dataset. The network is already known to be very fast, but with quantum integration by replacing the final Fully Connected layer and freezing all the layers, with quantum variational circuit, the number of trainable parameters decreased by many folds thus making the circuit less complex to train and achieve similar results compared to a classical setup on the Pet dataset.

1.1 Quantum Computing

Quantum Computing takes advantage of quantum states such as superposition and entanglement, interference to perform calculations. Quantum computers operate on qubits, quantum equivalent of bits, with the only difference being, in superposition state, the qubit can be in 0 and 1 state at the same time. Quantum computers are known to have time complexity advantage over classical computers, thus giving exponential time speedups in various applications where classical computers would take years to solve. Since quantum computing is essentially linear algebra, and with new algorithm to calculate linear system of equations using quantum computer a.k.a. HHL[4] algorithm, has given hope that it may provide quantum advantage in the domain of machine learning too.

1.2 Transfer Learning

The transfer learning is a process where we use a pre-trained neural network model, which is trained on some dataset A, and modify the head of the model to make it work for a completely different dataset B. This method has been proven to accelerate the learning process and improve the accuracy on the new dataset. There are 4 different strategies to accomplish the task of transfer learning with quantum computing.

1. Classical-Classical (CC)
2. Classical-Quantum (CQ)
3. Quantum-Classical (QC)
4. Quantum-Quantum (QQ)

For my case study, I will be using CQ strategy and compare the results with CC strategy.

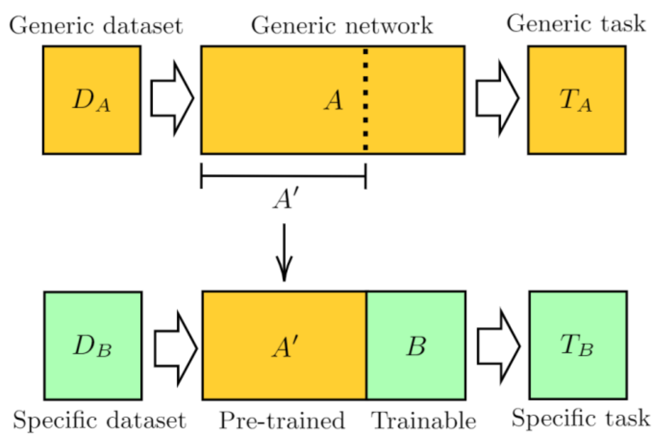

Fig 1. Transfer Learning implementation.

2 Dataset

I will be using Oxford-IIIT pets dataset called Cats and Dogs [3]. This dataset contains 37 pet categories with roughly 200 images for each class. Images have many variations in terms of scale, lighting and pose. Out of 37 pets, I will be working on only 8 pets to show working of proof of concept.

1.1 Data Transformation

The image dataset is first transformed by randomly flipping some images horizontally, then resized to 256x256, further center cropped to 224x224 and then normalized its mean and variance according to the ImageNet dataset. This transformed image is fed to ResNet18 network pretrained on ImageNet dataset.

3 Methodology

Here I have constructed two separate models, classical model, and classical-quantum hybrid model. Both having the same body of pre-trained ResNet18 model with its fully connected layer removed and replaced with different heads as shown below in Fig 2. I have frozen all the layers of ResNet18, so that they are not trained again on the new image dataset. This is the advantage of transfer learning, where we take the ImageNet data's classification training and used the weights and biases on our pet data with the aim of significantly reducing the training time and improving accuracy.

```
(1): Sequential(
  (0): AdaptiveConcatPool2d(
    (ap): AdaptiveAvgPool2d(output_size=1)
    (mp): AdaptiveMaxPool2d(output_size=1)
  )
  (1): Flatten(start_dim=1, end_dim=-1)
  (2): BatchNorm1d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (3): Dropout(p=0.5, inplace=False)
  (4): Linear(in_features=1024, out_features=8, bias=True)
  (5): ReLU(inplace=True)
)
```

Fig 2. Modified Fully Connected Layer on Classical implementation

The quantum hybrid-model has a Dressed Quantum network using PennyLane [5] by Xanadu with PyTorch[6] interface. The dressed quantum network is constructed as a layer of Hadamard gates, followed by a hidden layer-like layer of parameterized qubit rotations around Y-axis, with a final layer which would entangle all the qubits together using CNOT. This hidden layer is repeated total 6 times to simulate a hidden layer depth in a classical neural network as shown in the Fig 3.

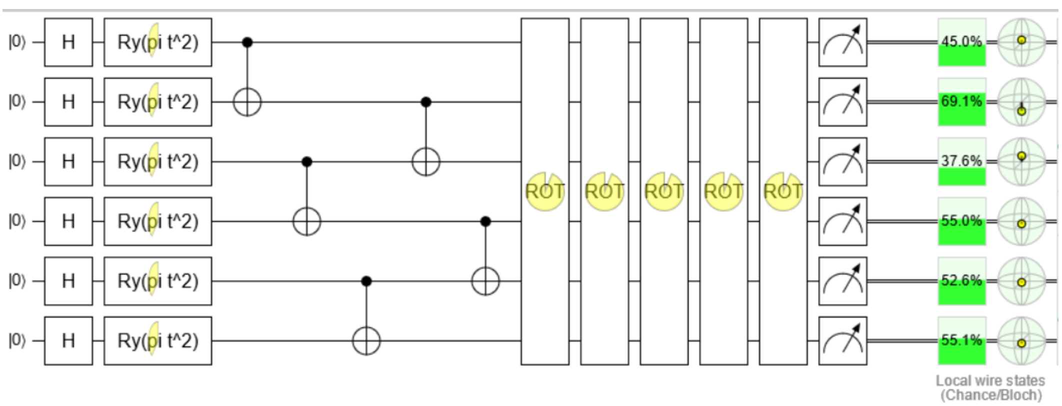

Fig 3. Quantum Circuit

This quantum circuit is dressed between classical Linear layers which takes classical input from the ResNet18 and passes to the quantum circuit and other layer, which takes the quantum output measurement in the Z-axis to make it binary, for classification.

```
  AdaptiveAvgPool2d-67          [-1, 512, 1, 1]              0
             Linear-68                  [-1, 6]          3,078
             Linear-69                  [-1, 8]             56
DressedQuantumNet-70                    [-1, 8]              0
```

Fig 4. Modified Fully Connected layer on Hybrid Quantum implementation

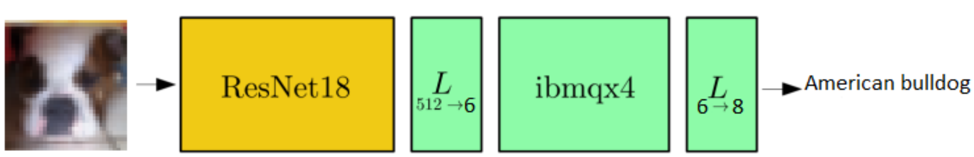

Fig 5. Dressed Quantum Transfer Learning Classification Architecture

One notable thing about the updated models is the number of trainable parameters in the network. For classical – Total params: 11,186,760, Trainable params: 10,248 & non-trainable params: 11,176,512. In the case of quantum network – Total params: 11,179,646 Trainable params: 3,134 non-trainable params: 11,176,512

4 Results

As expected, the classical approach achieves the accuracy of 98% on train data and 95% on test data in just 15 epochs, while taking significantly less time. The reason why the network was able to learn so well is because of transfer learning. The network was able to correctly classify the 8 different pets with an accuracy of 90%+ in just 4 epochs, and the loss also dropped to it lowest point in 3 epochs.

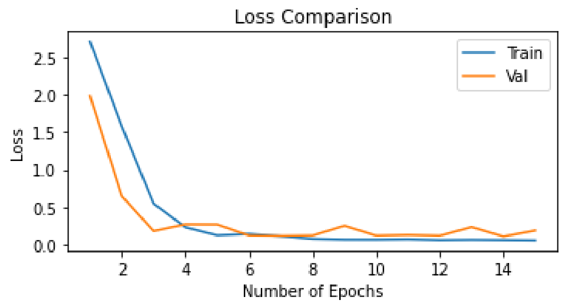

Fig 5. Loss comparison of Train and Validation Data on Classical implementation

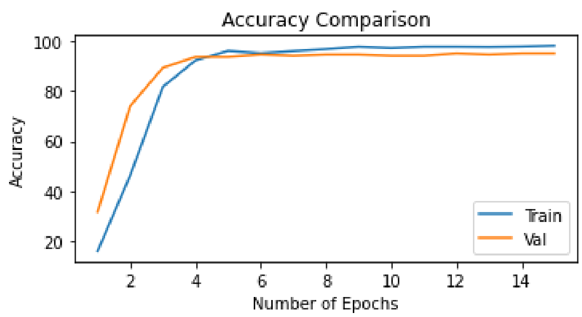

Fig 6. Accuracy comparison of Train and Validation Data on Classical implementation

In the case of Quantum approach, I observed that the loss took a lot of epochs to fall under an acceptable level as compared to classical implementation. With learning rate scheduler implementation, the learning rate is decreased by 90% after 11 epochs, after which we can see the stability in the model's accuracy and loss.

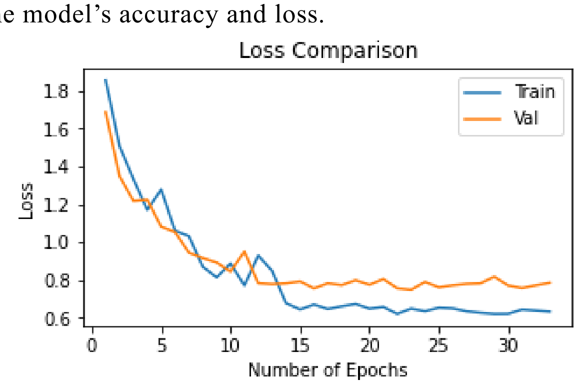

Fig 7. Loss comparison of Train and Validation Data on Quantum implementation

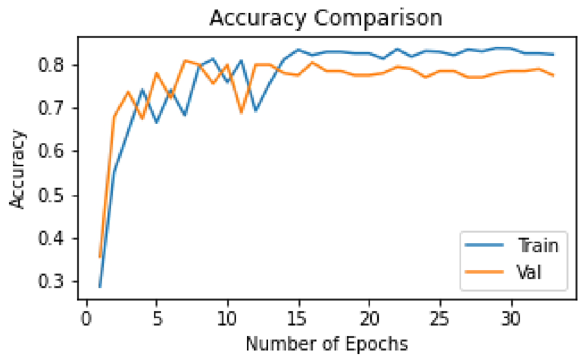

Fig 8. Accuracy comparison of Train and Validation Data on Quantum implementation

Result of running the classical and the hybrid-quantum model on a test sample of the validation set is shown below in Fig 9 and 10. A sample of 10 images were tested, and the number of correct predictions is similar to what we got above.

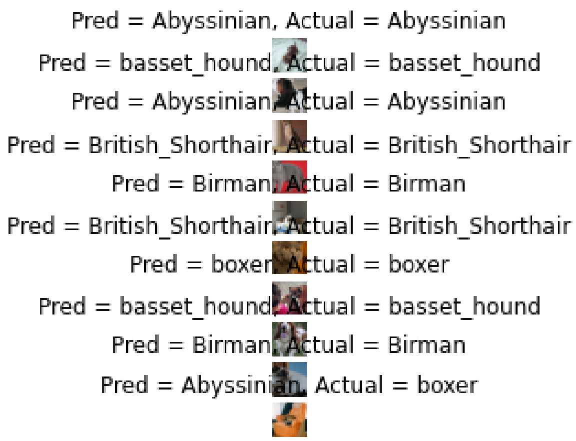

Fig 9. Predictions on Classical implementation

Fig 10. Predictions on Quantum implementation

5 Conclusion

With 1/3rd of the trainable parameters compared to classical implementation; I could achieve respectable accuracy levels with the hybrid-quantum implementation. The only downside was the training time being higher than the classical, and the simulation was run on a classical processor. If performed on an actual quantum computer on cloud, there is a good chance of improving the training time by a significant margin. So as of now, there is no quantum advantage to the classical problem of image classification. But there seems to be a promising future for Quantum Machine Learning.

6 Future Scope

Today's Quantum Computers are very noisy in nature and are very prone to disturbances and limits the number of rotations that can be performed on a qubit. These tests were thus run on a simulator device provided by the Pennylane package. In future, we can benchmark these results again when we have a noise-free quantum computer.
Furthermore, we can engineer more complex embedding and measuring layers[7], we can also add more ancillary subsystems and discarding and measure some of them in the middle of the circuit. We can also increase the number of qubits and increase the trainable parameters to improve the accuracy.

Code

https://github.com/ChaakuDaaku/CSE676-Quantum-Classification

References

[1] Andrea Mari, Thomas R. Bromley, Josh Izaac, Maria Schuld, and Nathan Killoran. *Transfer learning in hybrid classical-quantum neural networks*. arXiv:1912.08278 (2019).

[2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Classification.* arXiv:1505.04597 (2015)

[3] Omkar M Parkhi Andrea Vedaldi Andrew Zisserman1 C. V. Jawahar. *Cats and Dog.* robots.ox.ac.uk/~vgg/publications/2012/parkhi12a/parkhi12a.pdf (2012)

[4] Harrow, Aram W; Hassidim, Avinatan; Lloyd, Seth (2008). *"Quantum algorithm for solving linear systems of equations". Physical Review Letters: 150502*. arXiv:0811.3171

[5] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, M. Sohaib Alam, Shahnawaz Ahmed, Juan Miguel Arrazola, Carsten Blank, Alain Delgado, Soran Jahangiri, Keri McKiernan, Johannes Jakob Meyer, Zeyue Niu, Antal Száva, and Nathan Killoran. *PennyLane: Automatic differentiation of hybrid quantum-classical computations.* 2018. arXiv:1811.04968

[6] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. *Automatic differentiation in PyTorch. In NIPS Autodiff Workshop, 2017.*

[7] Andrea Mari, Thomas R. Bromley, Josh Izaac, Maria Schuld, and Nathan Killoran. *Transfer learning in hybrid classical-quantum neural networks.*