QuantumNet | SC Quantathon V1

QUANTUM IMAGE CLASSIFICATION

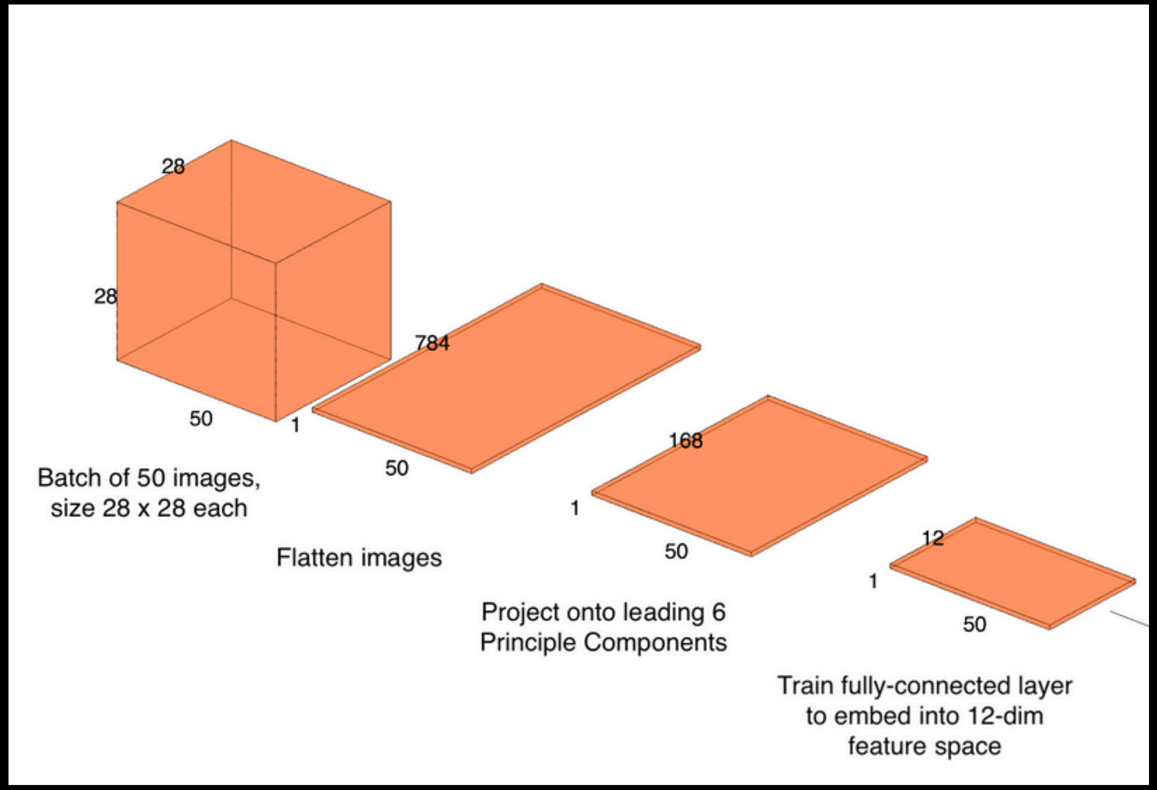
SIMPLE BLOCK - 0 OR 1 DIGIT CLASSIFICATION

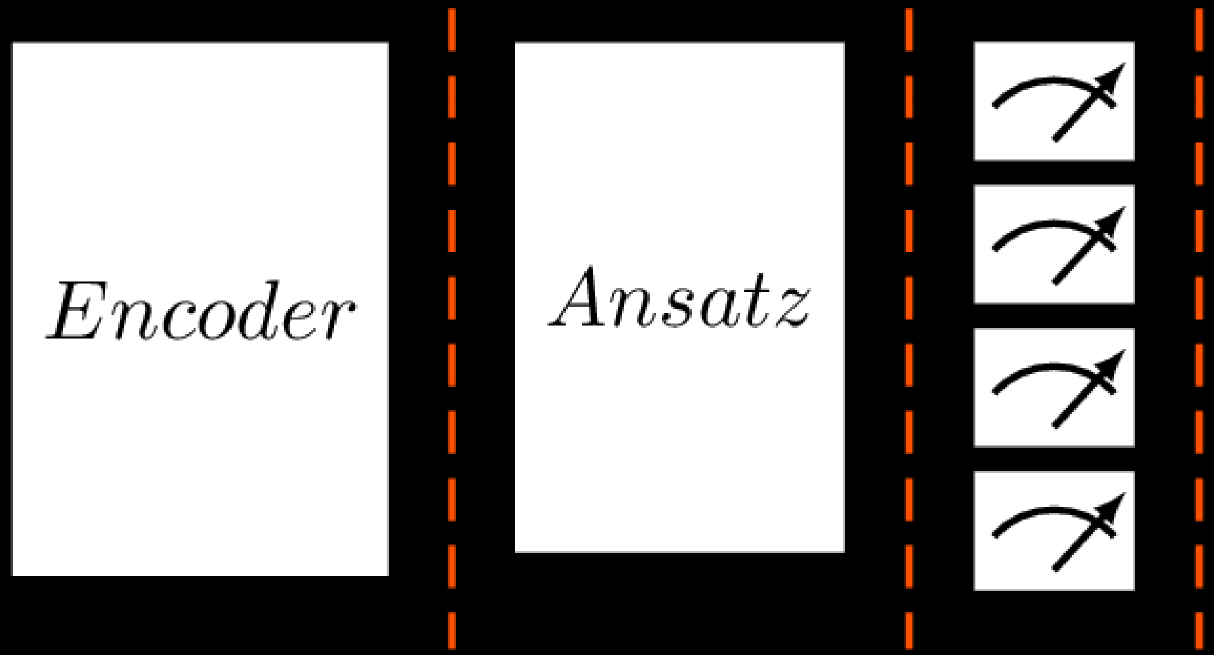
Mubarak Ganiyu & Rachel Koh | Vanderbilt University

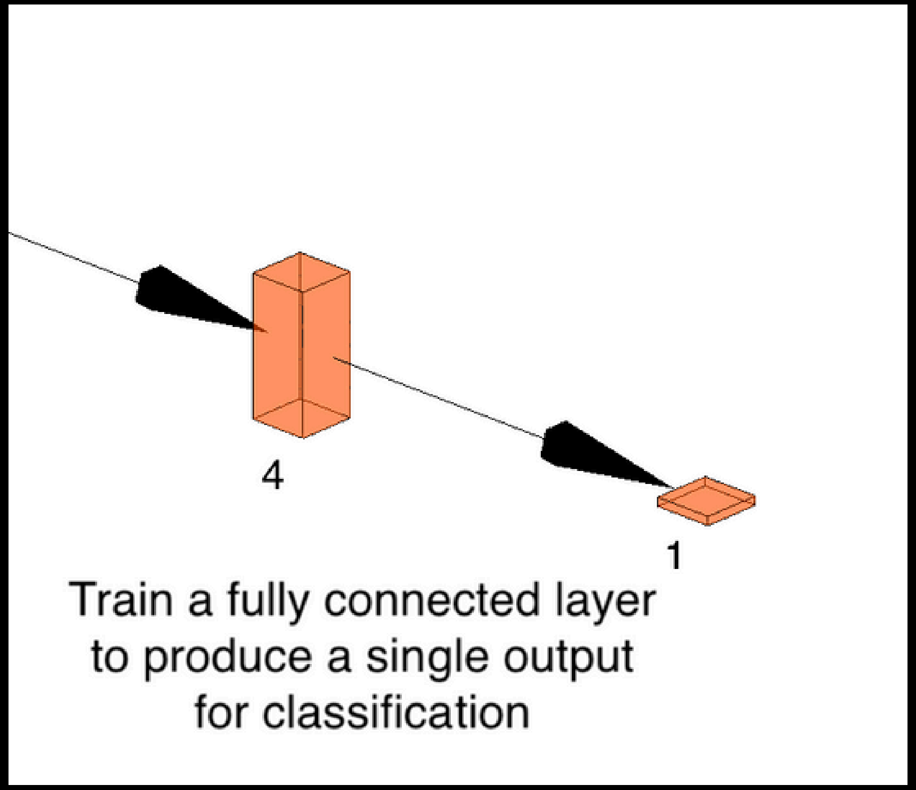
QuantumNet

THE CHALLENGE

Utilize quantum computing to design an algorithm that will act as a training layer within a hybrid quantum-classical machine learning architecture for differentiating between 0 and 1



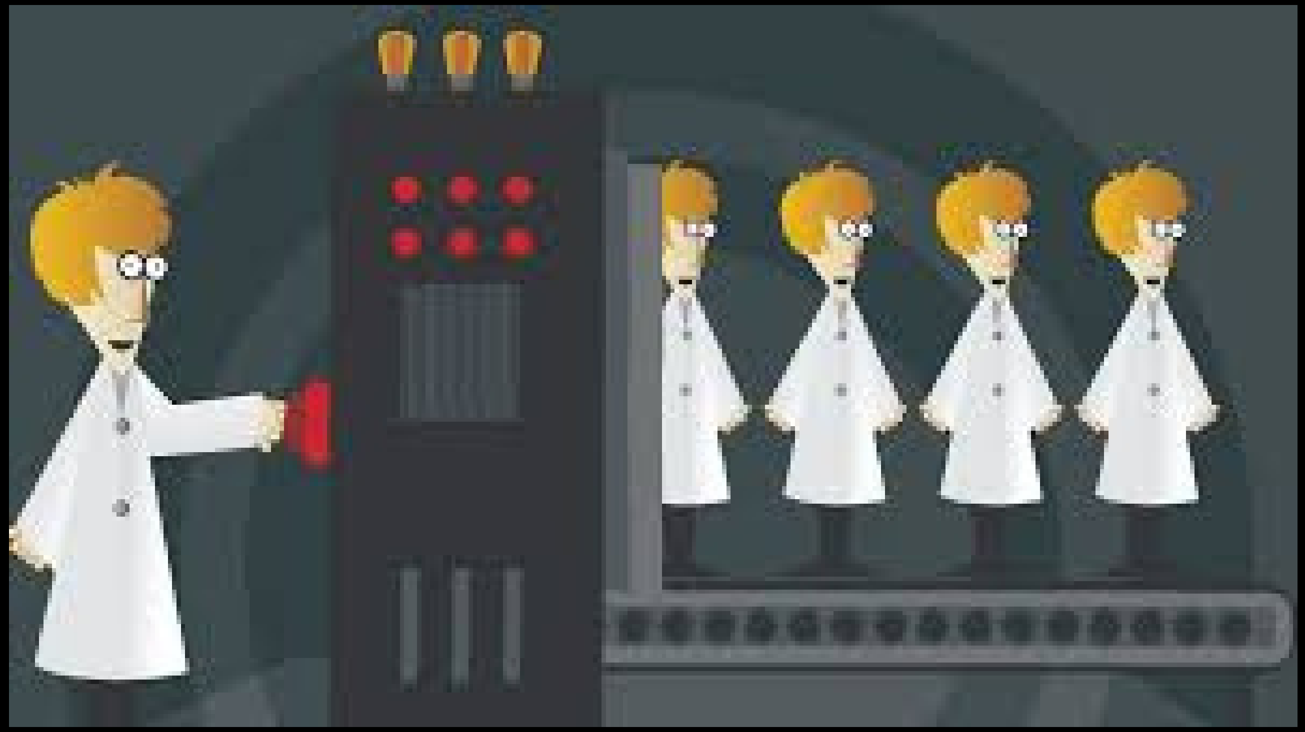




QuantumNet

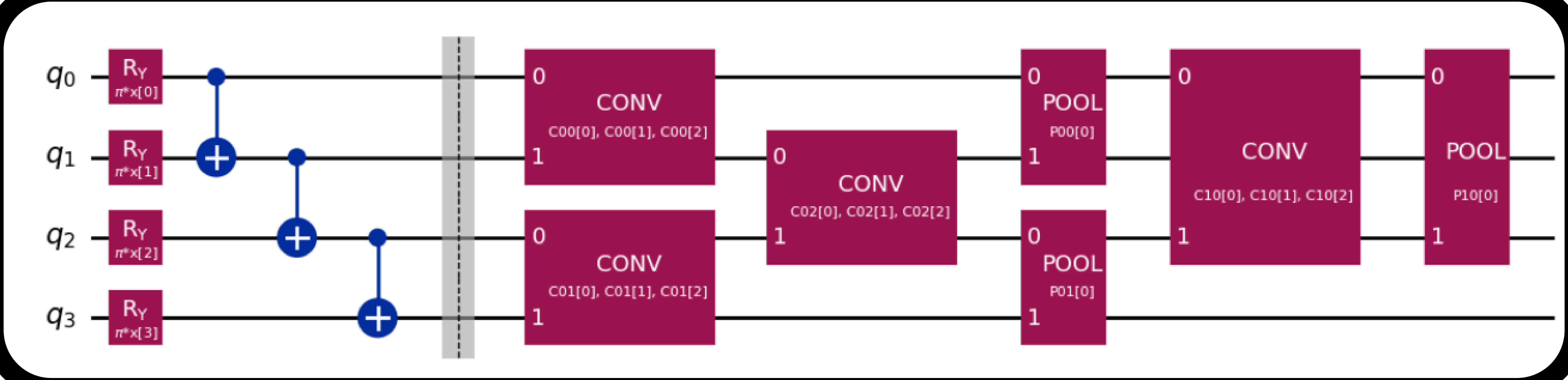
OUR OBJECTIVE

To develop the simplest quantum layer that can be embedded within a machine learning model to expedite the algorithmic runtime while being capable of providing an equivalent result when retrained on a similar data set



QuantumNet

THE INITIAL CIRCUIT



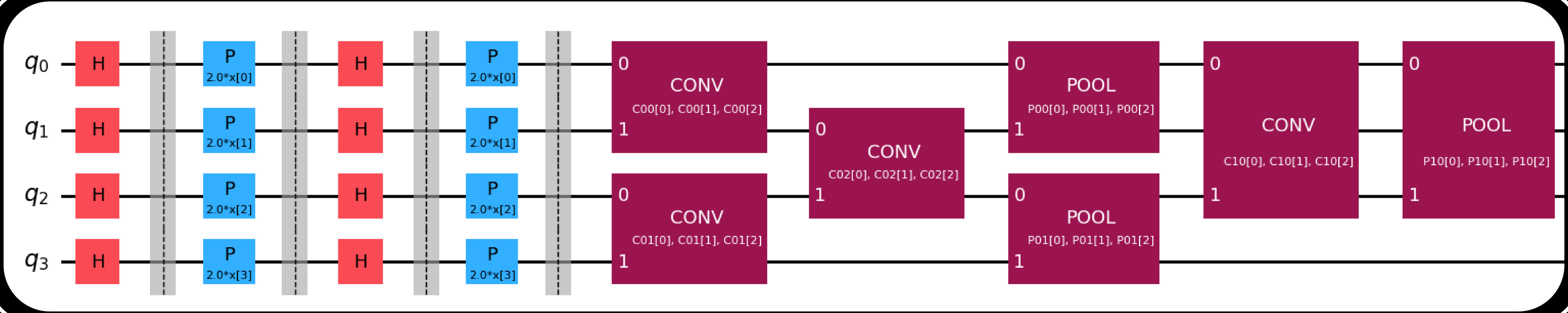
Quantum Features: IIIX + IIIY + IIIZ

Accuracy: 61%
Runtime: about 300 - 400 seconds per epoch

QuantumNet

FIRST ATTEMPT - DESIGN

- Updated the convolution and pooling subroutines using inspiration from Qiskit's Tutorial on QCNN
- Replaced AngleEncoder with ZFeatureMap as a means to simplify the algorithm

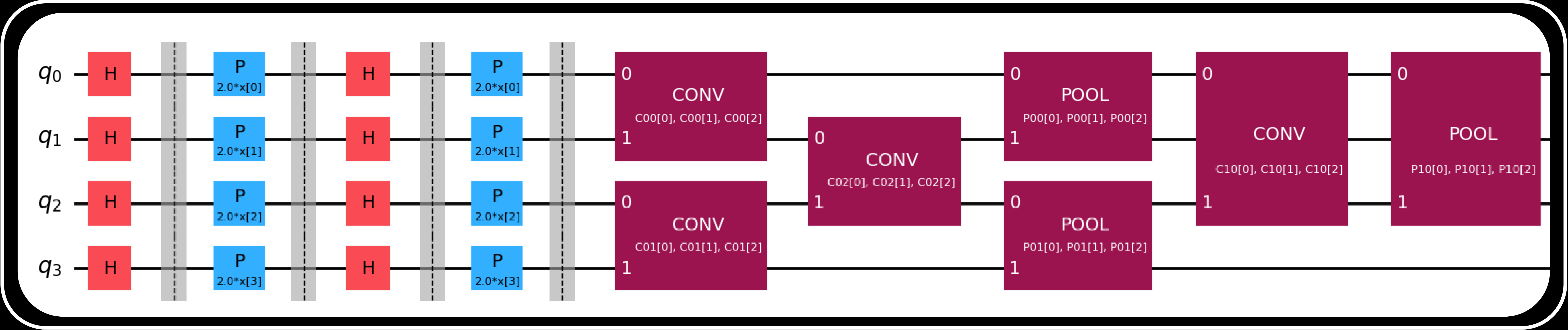


Quantum Features: ZZZZ

QuantumNet

FIRST ATTEMPT - OUTCOME

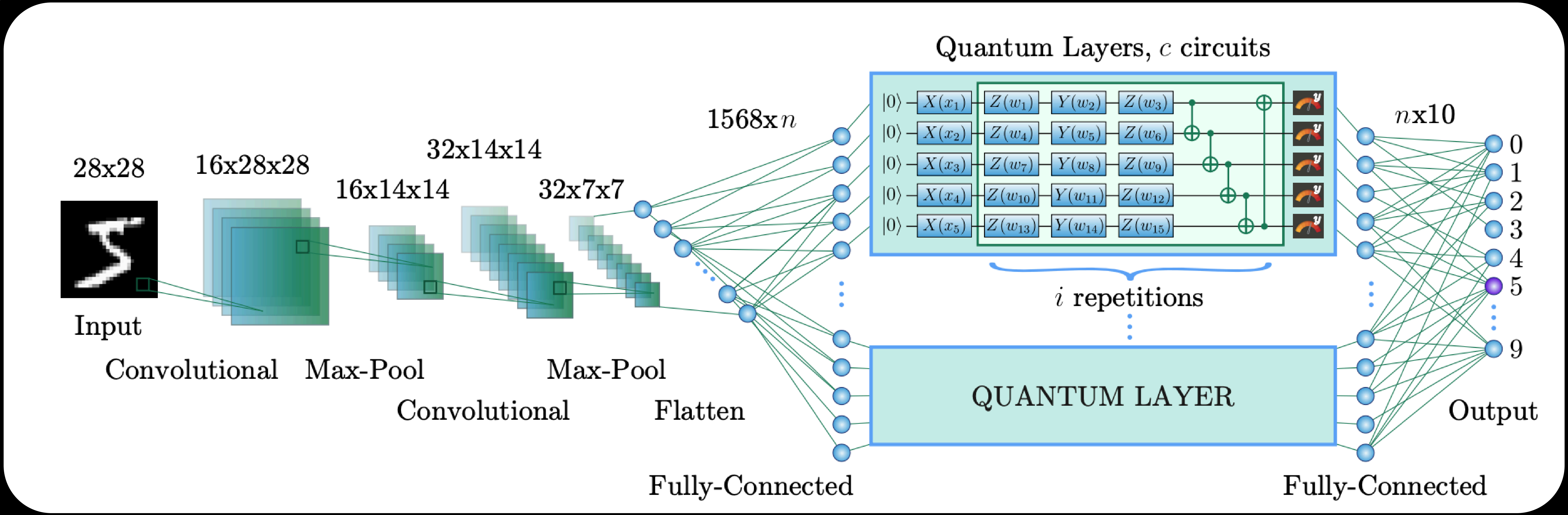
- Training time was quite long (roughly 200 seconds per epoch)
- Algorithm’s accuracy varied a lot (45% to 64%)
- Due to complexity of the model, it was hard to make minor incremental adjustments
    - After experimenting with multiple tweaks, this was apparent.



Quantum Features: ZZZZ

QuantumNet

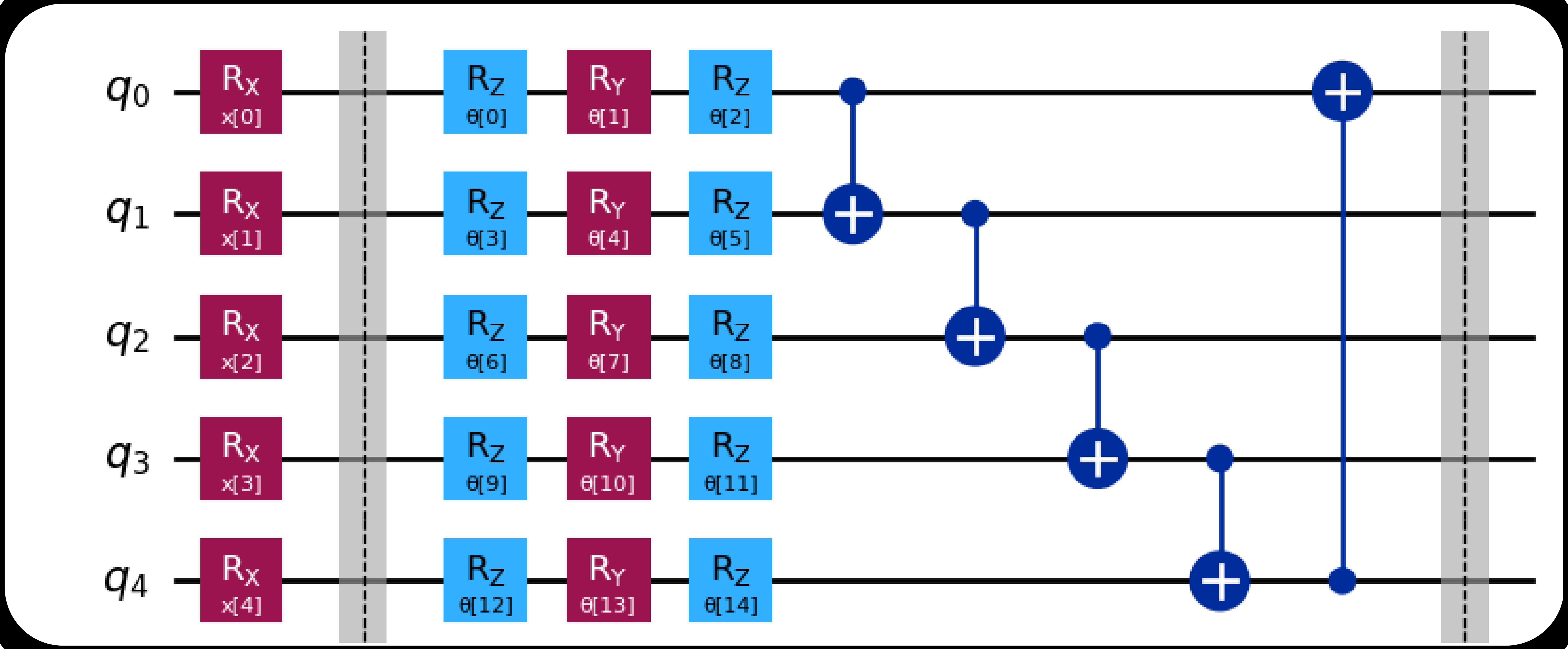
POTENTIAL SOLUTION I - HQNN



[A Senokosov et. al. (2024)]

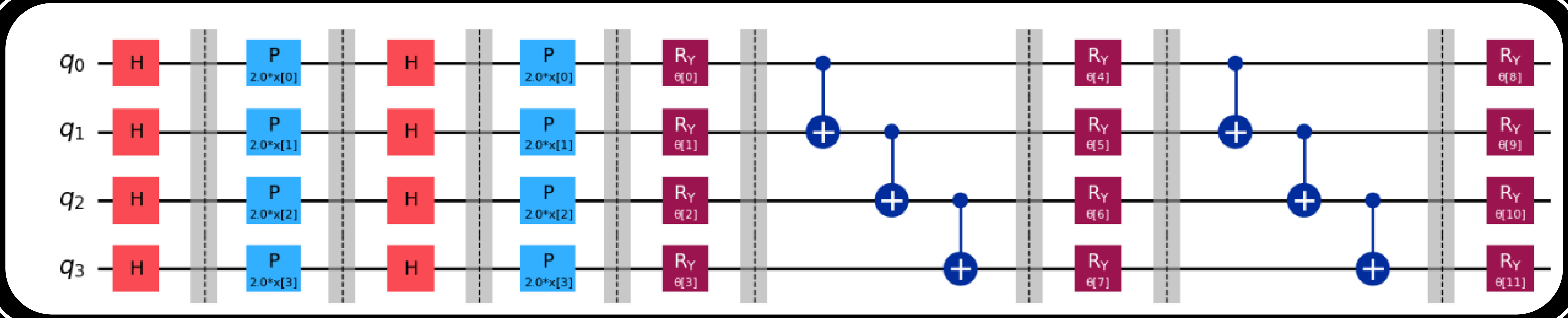
QuantumNet

POTENTIAL SOLUTION I - HQNN



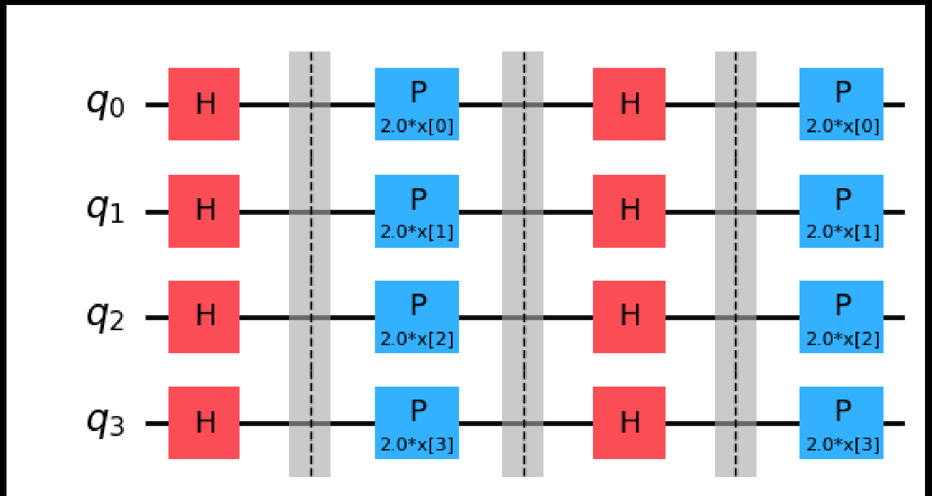
QuantumNet

POTENTIAL SOLUTION II - ZFEATUREMAP + TWO LOCAL

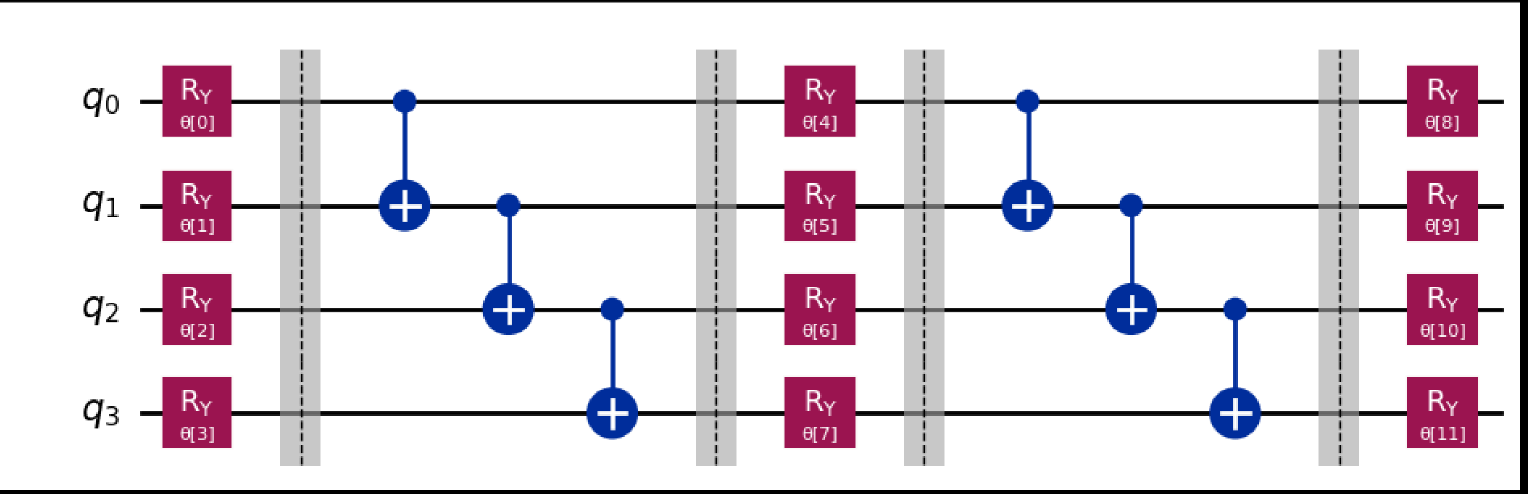


Quantum Features: ZZZZ

ZFeatureMap



Two Local



HYPERPARAMETER TUNING

Here are some paramaters that we constantly tweaked during the development of our algorithms

- **Learning Rate**
    - Varied the learning rate between 0.01 and 0.1
    - Preserving the learning rate at 0.1 as changes did not seem to have an impact

- **Batch Size**
    - Tried out different batch size values
    - Bigger batch sizes were not compatible with IonQ's backend system (i.e. anything > 50)

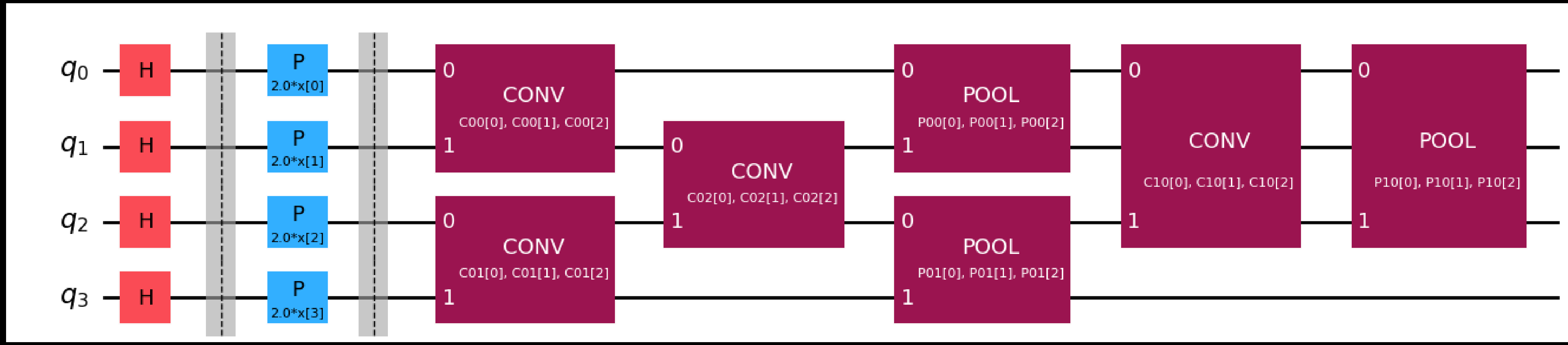
- **Epochs**
    - Used a range of epochs between 1 and 10.
    - Higher epochs led to the model's accuracy drastically reducing over time.
    - An epoch of 3 - 5 was sufficient

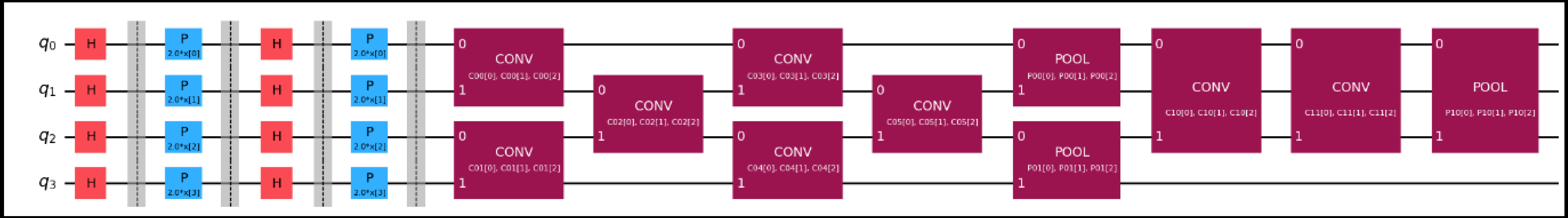
```
# Configure model training hyp
config = {
    "epochs": 10,
    "lr": 0.1,
    "batch_size": 50,
    "betas": (0.9, 0.99),
    "weight_decay": 1e-3,
    "clip_grad": True,
    "log_interval": 6,
}
```

QuantumNet

MAJOR TAKEAWAYS - SIMPLICITY VS COMPLEXITY

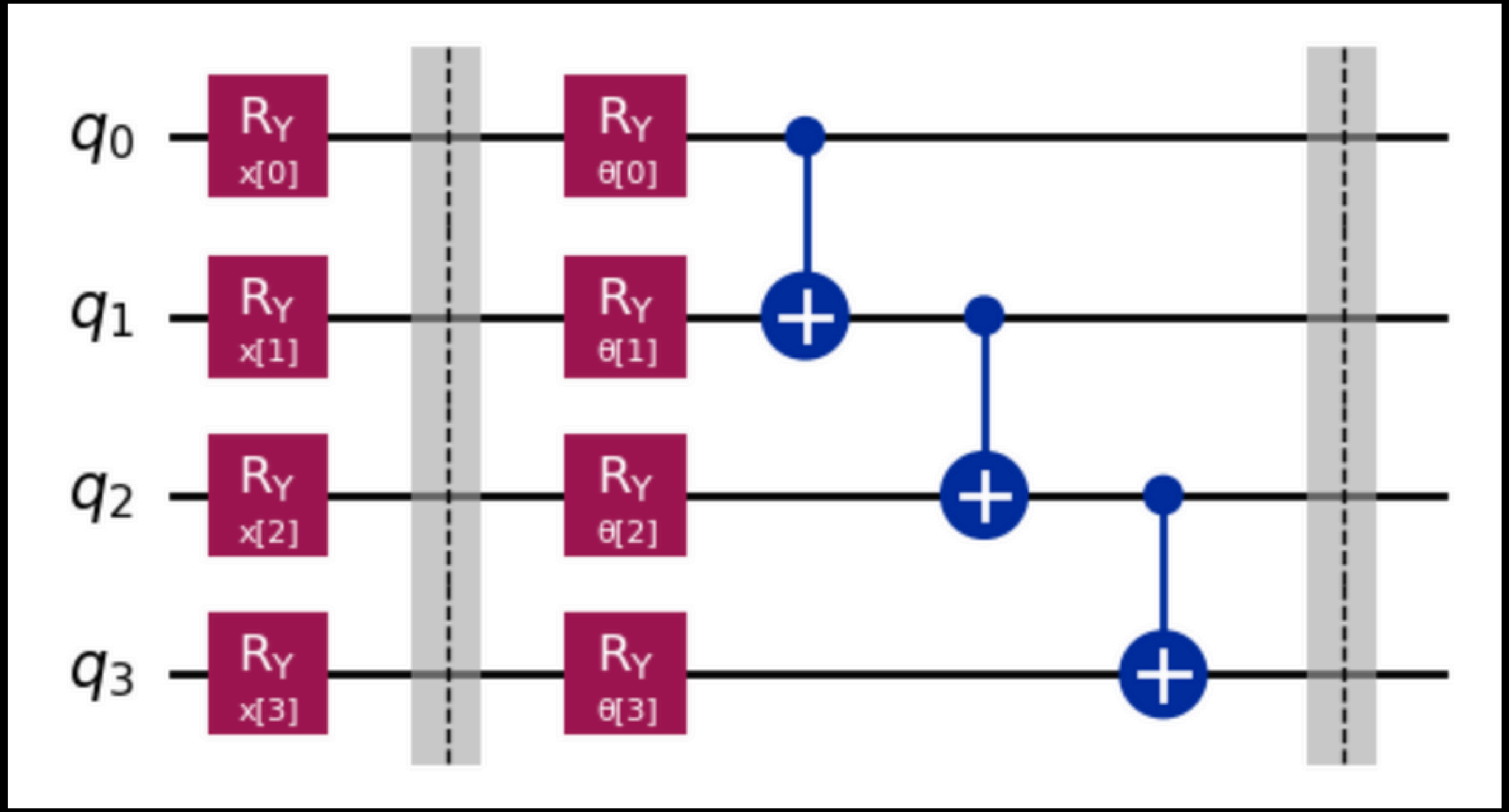
Too much depth increases training time and might not yield a very accurate result



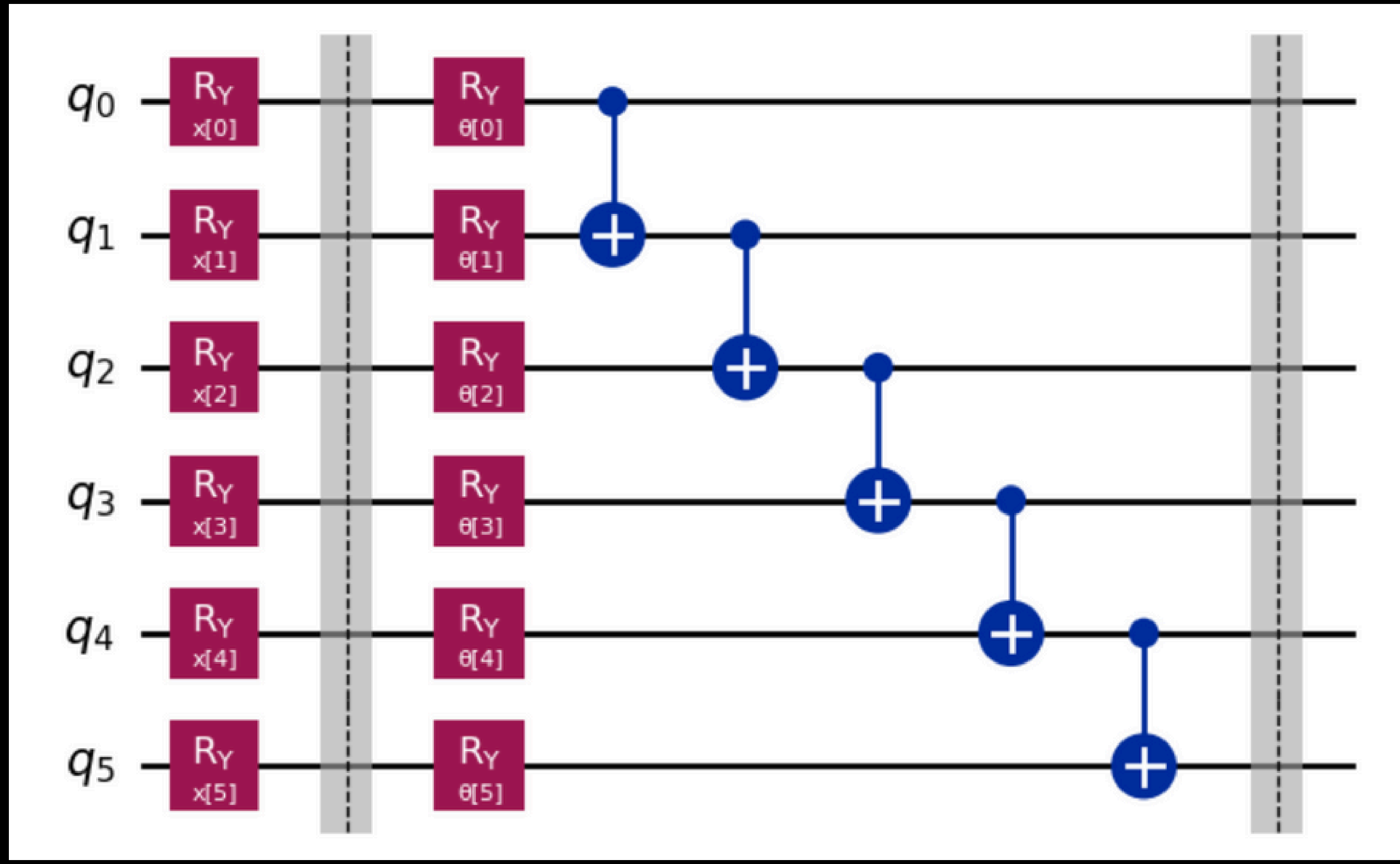


MAJOR TAKEAWAYS - NUMBER OF QUBITS

Adding more qubits does not necessarily yield better results



52-55% accuracy

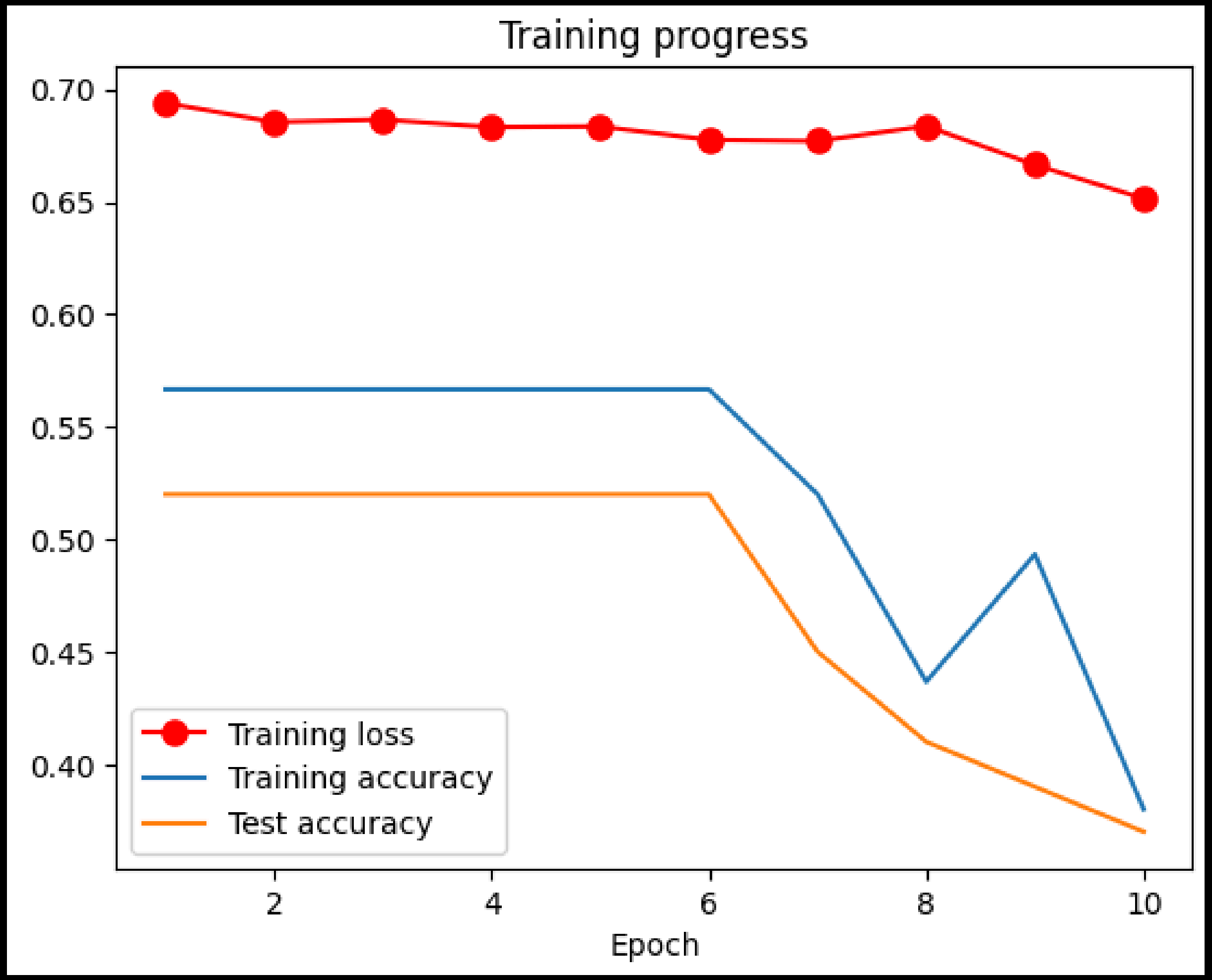


55% accuracy

QuantumNet

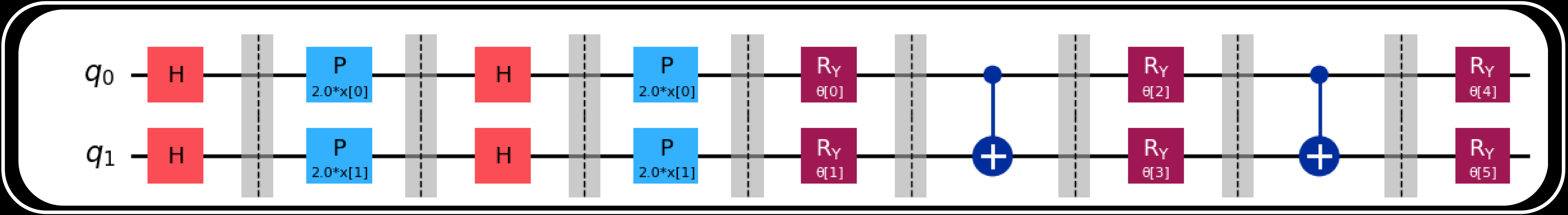
MAJOR TAKEAWAYS - NUMBER OF EPOCHS

Training for too long might lead to drop in accuracy (i.e. using a high number of epochs)



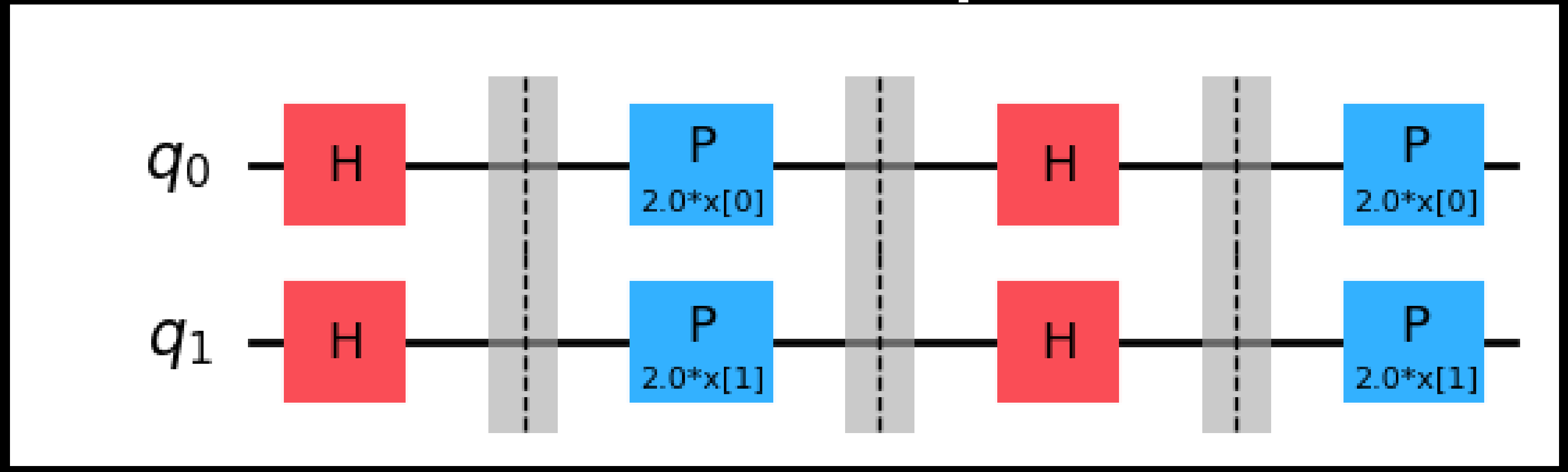
QuantumNet

FINAL SOLUTION - ZFEATUREMAP + TWO LOCAL (TWO QUBITS)

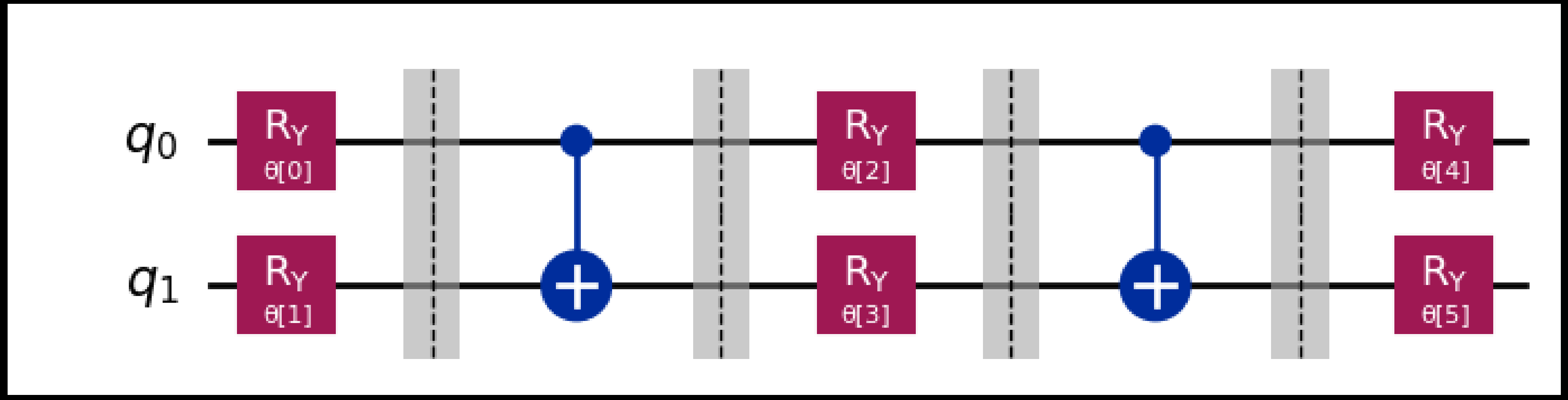


Quantum Features: ZI

ZFeatureMap



Two Local



QuantumNet

RESULTS

	HQNN	HQNN - Parallel	ZFeature Map + TwoLocal Circuit
Training Time	84.9 seconds per epoch	30 seconds per epoch	60 seconds per epoch
Accuracy	40 - 58 %	40 - 55%	55 - 65 %

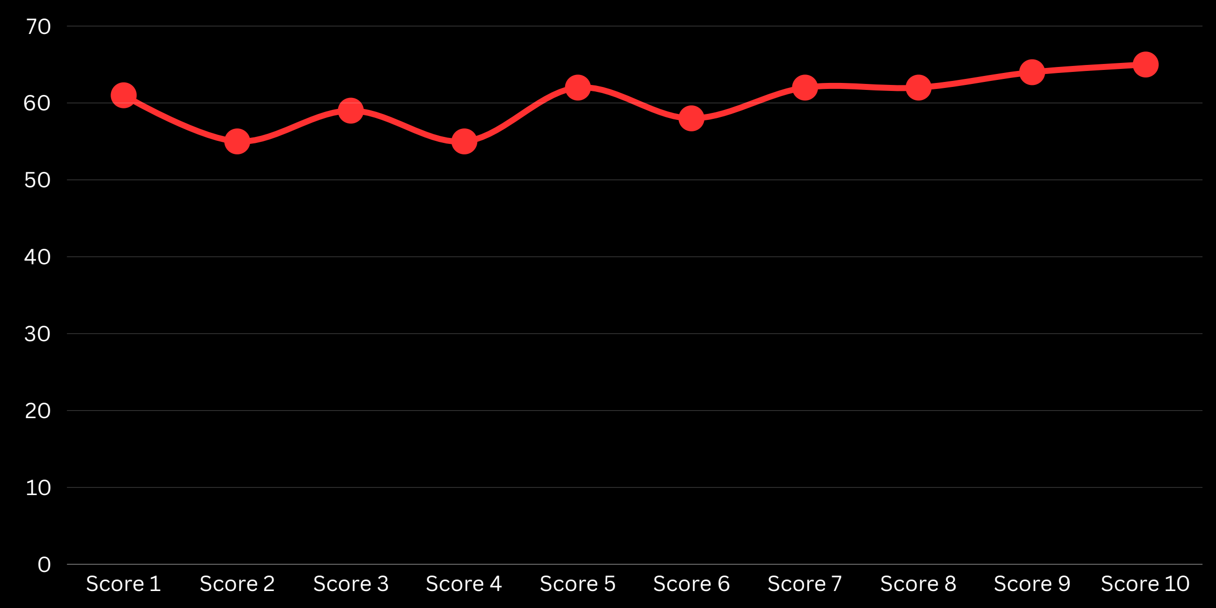
Configuration	epochs = 10	epochs = 10	epochs = 3;
			*batch_size = 25;
			*log_interval = 12

* indicates that these changes were made to upgrade the algorithm

LEADERBOARD PERFORMANCE

Below are the different scores that the algorithms achieved on the test set that was used on the leaderboard

- Initial Circuit from IonQ - 61%
- ZFeatureMap with transformed quantum convolution and pooling layers - 58%
- AngleEncoder with transformed quantum convolution and pooling layers - 59% and 64%
- ZFeatureMap with TwoLocal (4 qubits) - 62%
- ZFeatureMap with TwoLocal (2 qubits) - 62% and **65%**



QuantumNet

LIMITATIONS

Here are some obstacles that affected our performance

- No direct access to the source data
- Inability to experiment with the classical pre-processing mechanism
- 24 hours were insufficient to tackle the challenge rigorously
- Limited data size
- IonQ's backend system being unable to deal with bigger batch sizes

NEXT STEPS

Our game plan to improve our current work is:

- Conduct further research on the development of simpler quantum layers
- Develop a Hybrid Quantum-Classical Neural Network from scratch
    - More flexibility with the pre-processing step
- Gain access to the source data and better compute resources
- Re-run our experiment using newly developed methodologies

REFERENCES

- Liu, Bo, et al. "Quantum Convolutional Neural Networks: A Survey." arXiv, 2023, https://arxiv.org/pdf/2304.09224.
- Qiskit Community. "Quantum Convolutional Neural Networks." Qiskit Machine Learning Tutorials, https://qiskit-community.github.io/qiskit-machine-learning/tutorials/11_quantum_convolutional_neural_networks.html
- IBM Quantum. (n.d.). ZFeatureMap. IBM Quantum Documentation. Retrieved October 12, 2024, from https://docs.quantum.ibm.com/api/qiskit/qiskit.circuit.library.ZFeatureMap
- IBM Quantum. (n.d.). TwoLocal. IBM Quantum Documentation. Retrieved October 12, 2024, from https://docs.quantum.ibm.com/api/qiskit/qiskit.circuit.library.TwoLocal

LINK TO REPO

https://github.com/MUbarak123-56/image-digit-scquantathon

THANK YOU