QOSF Mentorship – Screening Task Research Document

Ashwin Kaliyaperumal

**First Ansatz**

This is my first attempt at “Task 2 – Encoding and Classifier” from the Cohort 5 Screening Tasks.

Diagram

Description automatically generated with medium confidence

1: my first ansatz for Task #2; the barrier separates the data encoding bit from the variational circuit part

My variational circuit is based off of the [quantum variational classifier](https://medium.com/qiskit/building-a-quantum-variational-classifier-using-real-world-data-809c59eb17c2) by Rodney Osodo for a heart attack dataset. I used an angle encoding scheme after normalizing all of the data from columns 0 to 3. It was the easiest to implement as it would map the four features to four qubits. For the variational model, I followed the prompts suggestion and used a layer of CNOT’s sandwiched between an Rx and Rz gate. I am using the SPSA optimizer and able to consistently get 100% accuracy on the test data.

**Observations and Places to Improve**

I want to improve on a couple of aspects. I want to 1) experiment with different encoding schemes and variational circuits to see which would be best for this dataset and 2) somehow make the cost curve smoother and steeper.

I mostly chose the angle encoding scheme because it was the easiest to implement using quantum gates. As I said before, I based the variational circuit on the task prompt. I want to experiment with different variational algorithms, especially the prebuilt ones provided by Qiskit. One question I want to answer is how I can best encode the data.

The cost value seems to jump everywhere while the circuit is optimizing its parameters. Though it eventually finds the optimal parameters and value, I don’t really understand why the cost keeps jumping all over the place instead of smoothly decreasing.

Chart

Description automatically generated

2: the cost value shows a downward trend, but is very chaotic

**Testing the Data Encoding Scheme and Variational Ansatz**

I am going to test different feature map encoding schemes with the Real Amplitudes ansatz. For some reason, I couldn’t get any other ansatz, like EfficientSU2, to work. I will test the PauliFeatureMap and the ZZFeatureMap with my original angle encoding scheme.

|  |  |  |
| --- | --- | --- |
| Feature Map + Variational Circuit | Cost | Accuracy |
| Original Circuit | 0.13612224259323233 | 100% |
| PauliFeatureMap | 0.17419994312864498 | 83.33% |
| ZZFeatureMap | 0.15614310722036695 | 99.17% |

My original circuit performed the best. I am not fully sure why, but this could be because RealAmplitudes uses RY gates, which match with my original circuit. The Pauli Feature Map and ZZ Feature Map use unitary, X, and Hadamard gates. To test if the gate that maps the function decreases the cost, I am going to use a TwoLocal ansatz with RX and CX gates, along with full entanglement.

The TwoLocal ansatz is very similar to mine, except mine includes RZ gates before and after the CNOT entangling layer. I haven’t tested my original ansatz with different feature maps, but I predict that it will have similar results to this TwoLocal circuit.

|  |  |  |
| --- | --- | --- |
| Feature Map + Variational Circuit | Cost | Accuracy |
| Original Circuit | 0.17225285660724407 | 95% |
| PauliFeatureMap | 0.15986905659960599 | 96.67% |
| ZZFeatureMap | 0.15765375841371423 | 95% |

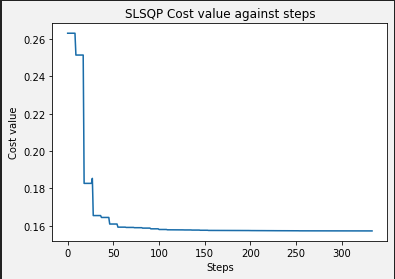
Chart

Description automatically generated

3: my data encoding scheme w/ the RX TwoLocal ansatz

I am going to use the ZZFeatureMap with the RealAmplitudes ansatz, like shown in both the Qiskit and Rodney’s implementation. I will test some optimizers. I am going to be checking COBYLA, ADAM, SLSQP, and POWELL, and comparing the results with the SPSA optimizer. I believe the reason for the uneven descent of my original variational circuit was due to the randomness inherit in the SPSA optimizer. This will be proven right if the other optimizers show a cleaner descent.

|  |  |  |
| --- | --- | --- |
| Optimizer | Cost | Accuracy |
| SPSA | 0.15765375841371423 | 95% |
| COBYLA | 0.16658115543874177 | 82.5% |
| ADAM | 0.23147974067601282 | 59.17% |
| SLSQP | 0.15728091581461578 | 94.17% |
| POWELL | 0.15725962284687664 | 93.33% |



4: SLSQP has a clean descent, converges on parameters faster than SPSA but isn't as accurate

**Conclusion**

I was really surprised with the difference in accuracy between optimizers. I was right regarding SPSA and the uneven curve; most of the other optimizers showed a smooth convergence. I will be using the ZZFeatureMap, the RealAmplitudes ansatz, and the SPSA optimizer for my second variational circuit.