Mandatory Assignment 2 - Brage Hamre Skjørestad, August Berge Andersen, Linus Krystad Raaen

**Data Exploration**

We started with loading the data and doing some exploration. It had 150 datapoints, with three classes with their values being in the range [0.1, 7.9]. Furthermore, the data was perfectly balanced. See image of class distribution below (Figure 1):Et bilde som inneholder tekst, skjermbilde, diagram, programvare

Automatisk generert beskrivelse

*Figure 1*

**Pre-processing / Encoding**

Not much pre-processing was needed because the data was so balanced, but we did normalize it from 0 to pi because we planned on using angle encoding. We chose this type of encoding because it’s efficient, scalable, and aligns well with standard quantum circuit structures. After that we split the data into train/validation/test sets at 70/15/15, respectively.

**Loss Function**

We started off with trying out Mean Square Error (MSE) and Root Mean Square Error (RMSE), but those didn’t really work because we were solving a classification problem and not a regression problem. Therefore, we ended up going for Log Loss.

**Two QNNs**

We first implemented IBM’s “real amplitudes” QNN from the lecture notes as this was a good way to just get the ball rolling and start understanding the structure of QNNs. Here’s the drawing (Figure 2):Et bilde som inneholder diagram, plan, skjematisk, line

Automatisk generert beskrivelse

*Figure 2*

After we got that to work, we implemented our own version of a Quantum Convolutional Neural Network (QCNN). Here’s the drawing (Figure 3):

Et bilde som inneholder tekst, diagram, plan, skjematisk

Automatisk generert beskrivelse

*Figure 3*

**Gradient Descent**

We first started by using SPSA as our optimizer just so we could quickly see if our real amplitudes implementation worked and actually gave us an answer. After that we implemented our own gradient descent using the methods described in the lecture notes.

It took some back and forth to find the perfect learning rate and other hyperparameters for the gradient descent, but we settled on learning\_rate=0.7, maxiter=100, shots=100 and that seemed to work well. KANSKJE SKRIVE LITT MER OM HVORDAN DU KOM FRAM TIL RIKTIG PARAMTERE HER. Få med negative results.

**Final Results Real Amplitudes**

We ended up getting 100% accuracy after about 80 minutes of runtime when using Real Amplitudes with 5 layers.

Here is the confusion matrix (Figure 4):  
Et bilde som inneholder tekst, skjermbilde, Rektangel, diagram

Automatisk generert beskrivelse

*Figure 4*

Here are the learning curves (Figure 5):

Et bilde som inneholder Plottdiagram, skjermbilde, line, diagram

Automatisk generert beskrivelse  
*Figure 5*

**Final Results Own Implementation OF QCNN**

We ended up getting \_\_\_% accuracy after about \_\_ minutes of runtime when using our QCNN with \_ layers.

Here is the confusion matrix (Figure 6):  
ADD THE CONFUSION MATRIX HERE

*Figure 6*

Here are the learning curves (Figure 7):

ADD THE LEARNING CURVES HERE  
*Figure 7*

**What We Learned**

Simpler is often the best solution.

**Easy, difficult, surprises**

a

**If We Had More Qubits**

We didn’t really need more qubits in this instance since we managed to get 100% accuracy.

**Joke**

Why do elephants have trunks?

Because they would look funny with a suitcase.