Project plan

Warm up

- Change from 0&1 → 3 & 6
- Change from 9x1 input to 3x3

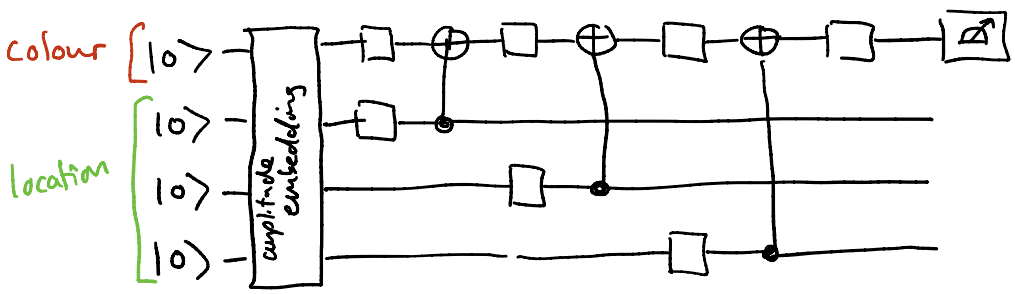
1 Image encoding

Using circuit structure in lecture notes with fixed layers (=2)
Adjoint diff method + ADAM optimiser
Binary classification 0 & 1 and 3 & 6

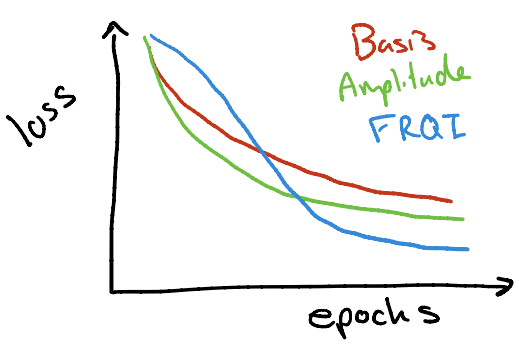
- Basis encoding (9 qubits - 9x1 image)
- Amplitude encoding (10 qubits - full image using amplitude embedding)
- FRQI (11 qubits - full image using amplitude encoding).

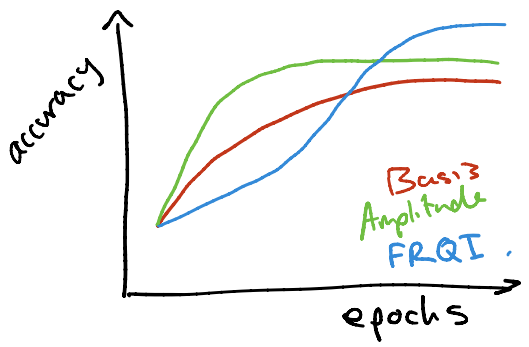
For FRQI, there is a choice for the measured qubit.
Could be colour qubit, or could be some other.

e.g.



Compare performance (final value + optimisation speed)





2 Different ansatz

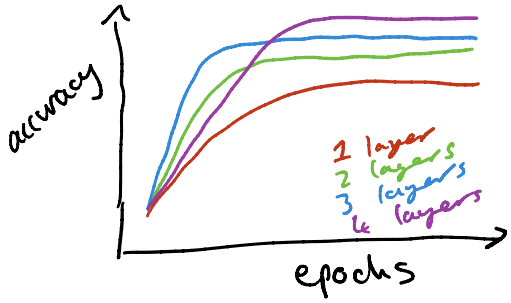
Adjoint diff method + ADAM optimiser

Binary classification 0 & 1

Start by fixing encoding (to best from previous)

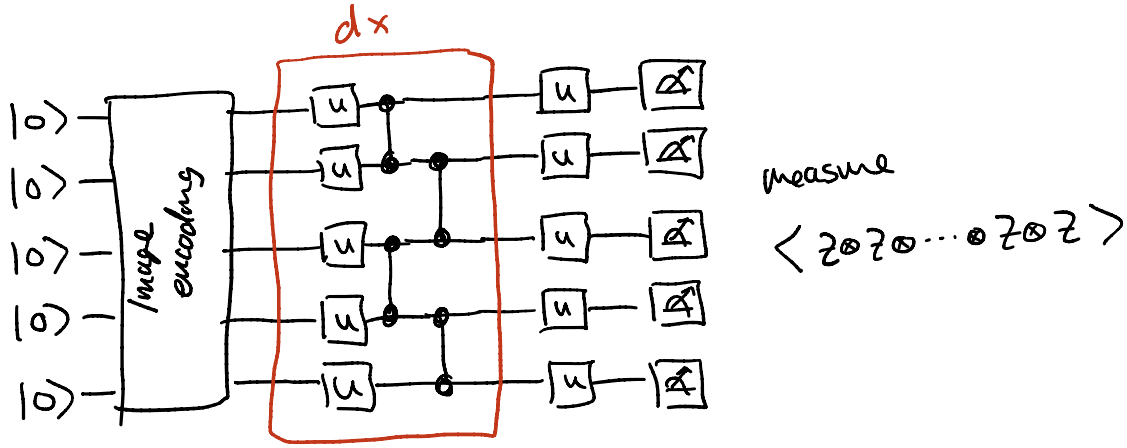
Bonus: Can also compare different encodings!

- Ansatz from lecture notes.
  - Vary number of layers ($d = 1, 2, 3, 4$)



- New ansatz / measurement.

e.g.



for varying layers $d = 1, 2, 3, 4$. (plot accuracy / loss)

3 Change optimization

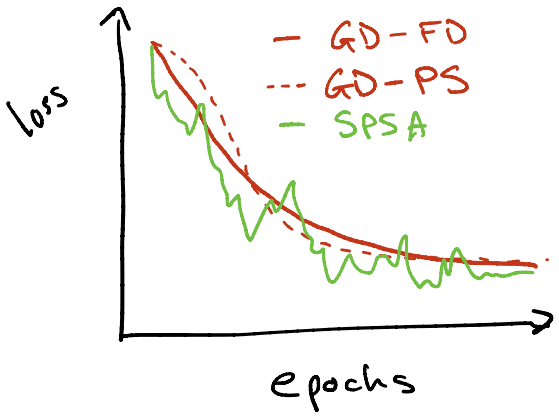
Binary classification.

Choose a circuit ansatz (Bonus: could also compare)

Choose an image encoding (Bonus: could also compare)

- Gradient descent. (pennylane GradientDescentOptimizer)
  - diff-method = "finite-diff"
  - diff-method = "parameter-shift"
  - Start with default step size. You may want to increase.

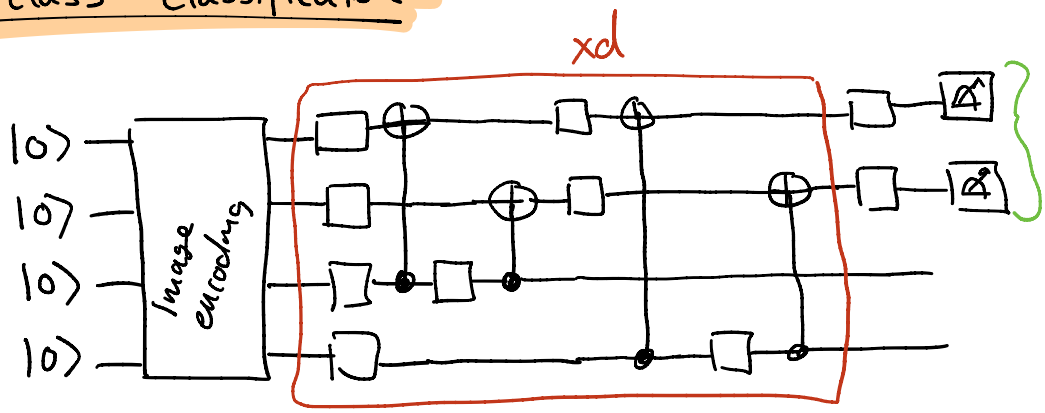
- SPSA (pennylane SPSAOptimizer)
  - doesn't use gradients.
  - Start with default parameters.



Similarly for accuracy.

4 Multi-class classification

E.g.



5 Fashion MNIST

Repeat steps using FashionMNIST