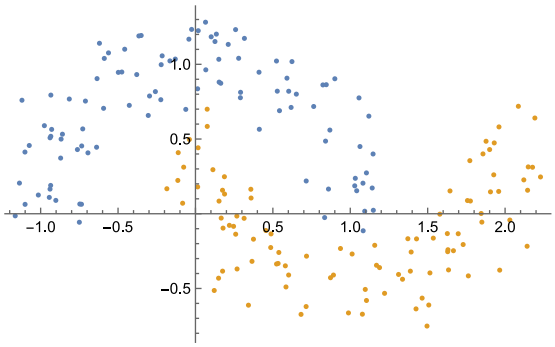
Project 2: QML class spring 2025. 60% of final grade

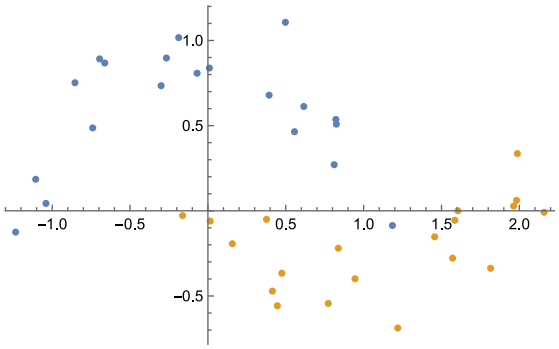
Due to 18 June 2025

Build a binary Variational Quantum Classifier

Use python/pytorch and libraries from qiskit or pennylane if you wish, to build a variational quantum classifier that is trained the trained and classifies the test data:



train data



test data

You may choose the method of encoding, #of qubits, #parameters, loss function, optimization method that gives you the best outcomes ,e.g., higher accuracy, less training steps, less parameters.

You also need to compare your results with a classical method e.g. deep NN, SVM.

You should provide:

1) A pdf file where you shortly describe your program, the design of quantum circuit in a pictorial way, the loss function that you selected, the optimization method that you have chosen, the evolution of loss functional with the epocs in a graph, the classification metrics for your best run (among all different initializations of weights): [Precision, F1-score, Recall, Accuracy]. Please also note the number of epochs for achieving this result, the number of epochs.
   Compare the results with the ones obtained via the classical classifier of your choice.
   Write down your conclusions.

2) An ipynb file with your program (executed) about the variational quantum classifier. For the classical classifier I do not need to see the program.

Extra credit if you depict the train data on the Bloch sphere before and after training.