

Learning sin wave between $[0, 2\pi]$

I will explain the steps for learning the sin wave sinewave.ipynb

1. Generate sin data between 0, 2π

I generated sin data between 0, 2π by using below function

```
def generate_sine_data1():  
    for theta in range(0, 145):  
        print(f"{theta*10*np.pi/180}")
```

2. Imports

Import pennylane, matplotlib for plots

3. Create qml node

Create a qml device and initialize the device on which we can run othe quantum circuit.

4. Build the NN circuit

Build the network and create layers. Each layer consists of rotation gates which are initialized with random weights and bias.

5. Build the cost function

Create a loss function that calculates square loss and update weights for every iteration.

6. Initialize the optimizer

Create AdamOptimizer and we run step and cost in optimizer for every iteration with the cost function we created. We initialize the circuit with 4 layers and iterations to 1000.