

### **Objective:**

The primary goal of the AQNN project is to develop and evaluate neural network models for the classification of wine origin. This involves distinguishing between three classes of wine based on specific features.

### **Project Structure:**

#### **1. Data Preprocessing:**

Dataset is Randomly divided into 80% training set and the rest as test set.

- Mini batch size = 4
- Number of features = 13
- Number of classes = 3

#### **2. Building of Artificial Neural Network model:**

i) Number of hidden layers = 0

Number of hidden layers = 1

- Number of units = 32
- Number of units = 64

ii) Sigmoid Function for input and hidden layers

Relu function for output layer

iii) Forward and backward operation are defined in Train function  
and Prediction function is also defined

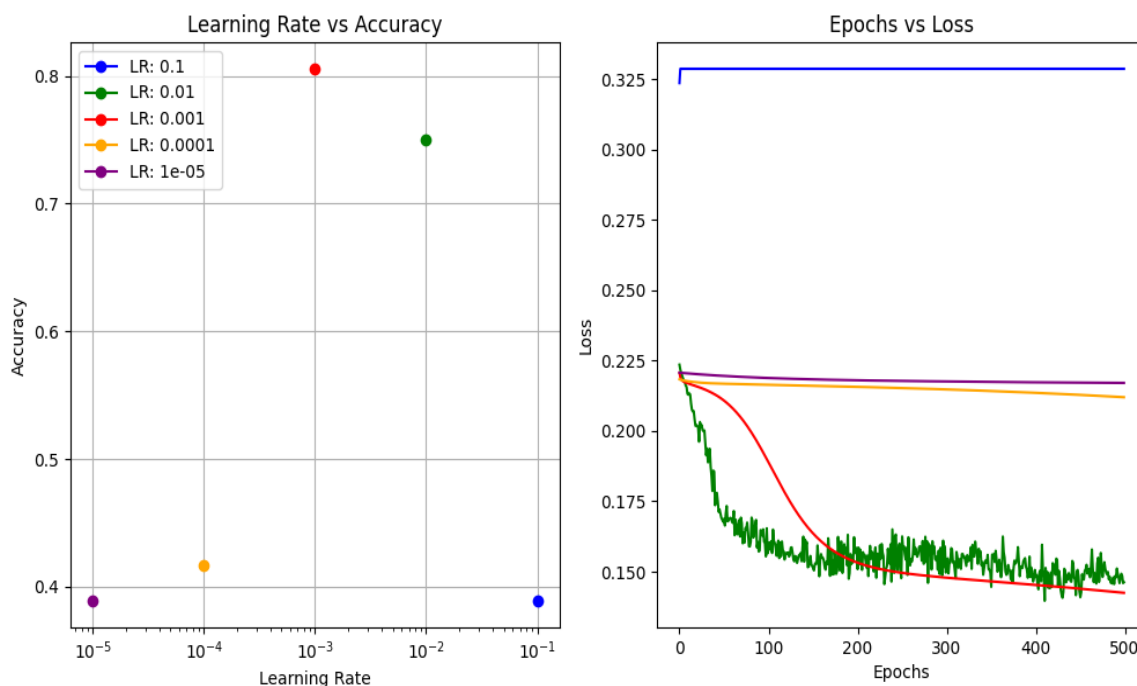
iv) Delta rule with Stochastic gradient descent is used for Error calculation

#### **3. Hyper-parameter tuning.**

1. For each of the architectures, vary the learning rates in the order of 0.1, 0.01, 0.001, 0.0001, 0.00001. Plot graph for the results with respect to accuracy and loss. (Learning rate vs accuracy/loss for each model).
2. Report test set accuracy for all the learning rates in a tabular form and identify the best model.

## Results:

Number of units = 64, Batch size = 4, Number of Epochs = 500



## Classification Report:

For learning rate = 0.001 {best among them}

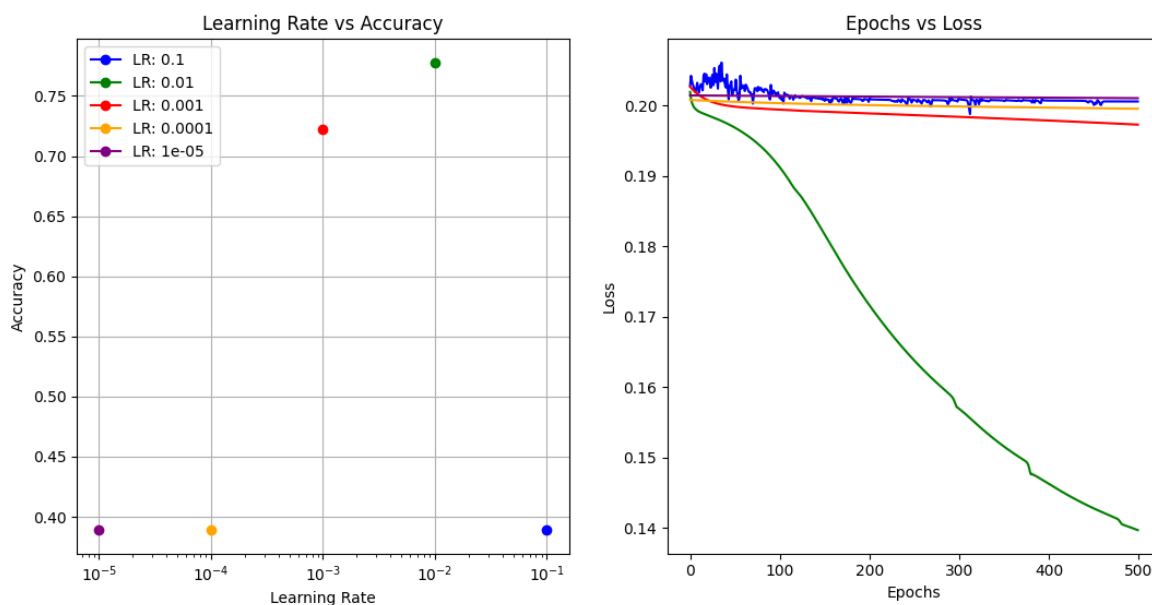
```
Training with learning rate: 0.001
```

```
Accuracy on test set: 0.8056
```

```
Classification Report:
```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 1            | 1.00      | 0.93   | 0.96     | 14      |
| 2            | 0.67      | 1.00   | 0.80     | 14      |
| 3            | 1.00      | 0.25   | 0.40     | 8       |
| accuracy     |           |        | 0.81     | 36      |
| macro avg    | 0.89      | 0.73   | 0.72     | 36      |
| weighted avg | 0.87      | 0.81   | 0.77     | 36      |

Number of units = 32, Batch size = 4, Number of Epochs = 500



## Classification Report:

Training with learning rate: 0.01

Accuracy on test set: 0.7778

Classification Report:

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 1            | 0.93      | 0.93   | 0.93     | 14      |
| 2            | 0.67      | 1.00   | 0.80     | 14      |
| 3            | 1.00      | 0.12   | 0.22     | 8       |
| accuracy     |           |        | 0.78     | 36      |
| macro avg    | 0.87      | 0.68   | 0.65     | 36      |
| weighted avg | 0.84      | 0.78   | 0.72     | 36      |

## Comparison with Pytorch Model:

Pytorch Model

```
Training with learning rate: 0.001
Accuracy on test set: 0.8333
Classification Report:
              precision    recall  f1-score   support

     1         1.00        0.93        0.96        14
     2         0.74        1.00        0.85        14
     3         0.75        0.38        0.50         8

 accuracy          0.83
 macro avg         0.83
weighted avg         0.84
```

My Model:

For Lr = 0.001, Epoch=500

```
Training with learning rate: 0.001
Accuracy on test set: 0.8056
Classification Report:
              precision    recall  f1-score   support

     1         1.00        0.93        0.96        14
     2         0.67        1.00        0.80        14
     3         1.00        0.25        0.40         8

 accuracy          0.81
 macro avg         0.89
weighted avg         0.87
```

My model tuned with Learning rate 0.001, Epoch 500 has an accuracy of 0.81