Neural Networks - Task (2) Report

MLP Back-Propagation

ID	NAME
2022170447	منة ياسر حامد مصطفى
2022170162	رؤى محمد لطفي محمد علي
2022170340	مايا سامح احمد سمير بيبرس
2022170629	ميسون حلمي عبدالرحمن علي شعبان
2022170441	منة الله محروس السيد عيد محروس
2022170611	منى خالد عثمان عواد عثمان

Introduction

The goal of this experiment is to implement a multi-layer perceptron (MLP) using back-propagation and evaluate its performance on a dataset with three classes. We investigate how the choice of activation function, learning rate, number of epochs, and network architecture affects classification accuracy.

Experimental Setup

Network Parameters:

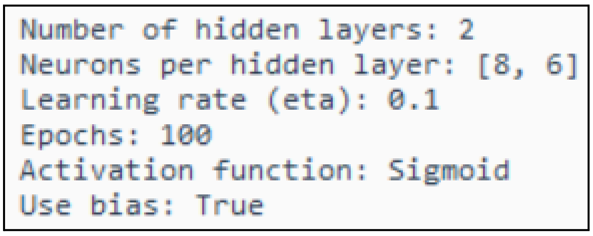
- Hidden layers
- Neurons per layer
- Activation function
- Learning rate
- Epochs

Experiments:

1. Sigmoid Activation

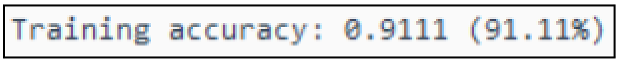
Experiment 1

Network Configuration:

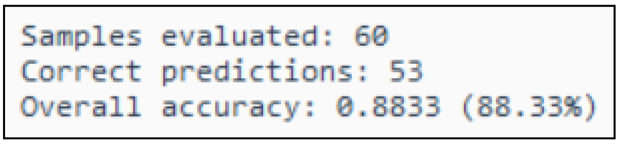


```
Number of hidden layers: 2
Neurons per hidden layer: [8, 6]
Learning rate (eta): 0.1
Epochs: 100
Activation function: Sigmoid
Use bias: True
```

Results:

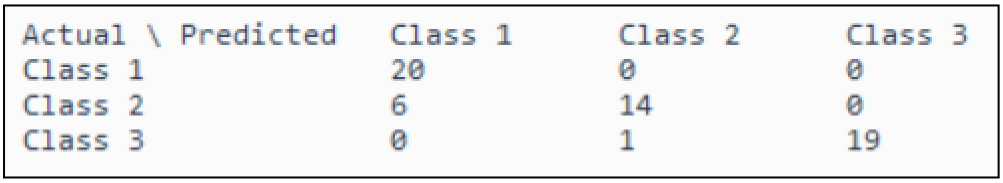


```
Training accuracy: 0.9111 (91.11%)
```



```
Samples evaluated: 60
Correct predictions: 53
Overall accuracy: 0.8833 (88.33%)
```

Confusion Matrix:



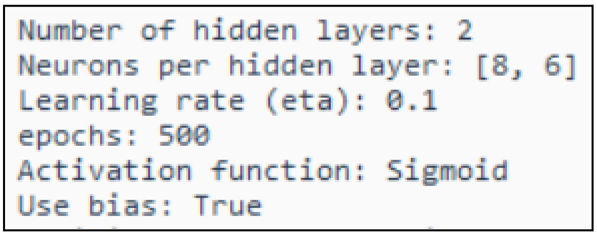
```
Actual \ Predicted    Class 1     Class 2     Class 3
Class 1               20          0           0
Class 2               6           14          0
Class 3               0           1           19
```

Observations:

- With only 100 epochs, the model showed moderate performance (88.33% test accuracy), highlighting that insufficient training time can limit the network’s ability to learn complex patterns effectively.

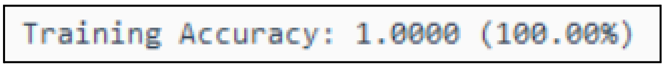
Experiment 2

Network Configuration:

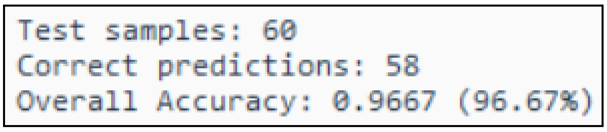


```
Number of hidden layers: 2
Neurons per hidden layer: [8, 6]
Learning rate (eta): 0.1
epochs: 500
Activation function: Sigmoid
Use bias: True
```

Results:

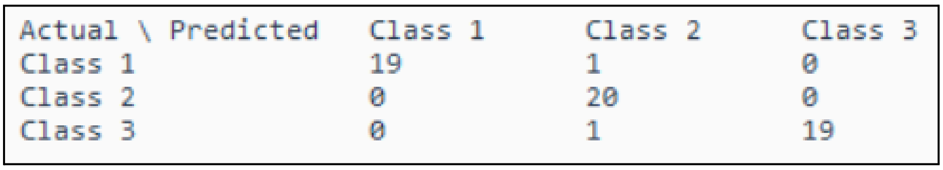


```
Training Accuracy: 1.0000 (100.00%)
```



```
Test samples: 60
Correct predictions: 58
Overall Accuracy: 0.9667 (96.67%)
```

Confusion Matrix:



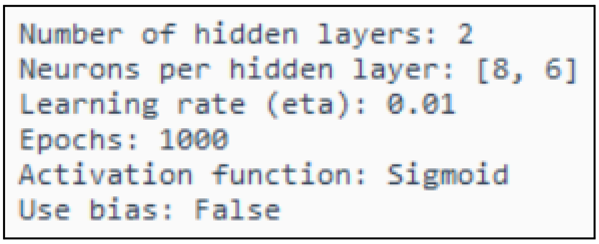
```
Actual \ Predicted   Class 1     Class 2     Class 3
Class 1              19          1           0
Class 2              0           20          0
Class 3              0           1           19
```

Observations:

- The model achieved perfect training accuracy and the highest test accuracy (96.67%), indicating strong generalization with this specific configuration of learning rate, epochs, and bias inclusion.

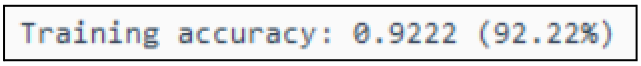
Experiment 3

Network Configuration:

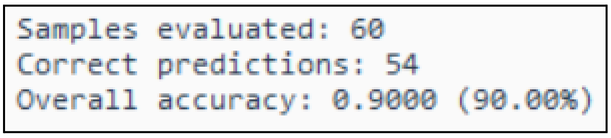


```
Number of hidden layers: 2
Neurons per hidden layer: [8, 6]
Learning rate (eta): 0.01
Epochs: 1000
Activation function: Sigmoid
Use bias: False
```

Results:

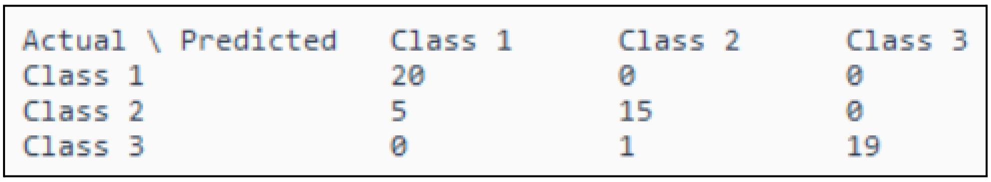


```
Training accuracy: 0.9222 (92.22%)
```



```
Samples evaluated: 60
Correct predictions: 54
Overall accuracy: 0.9000 (90.00%)
```

Confusion Matrix:



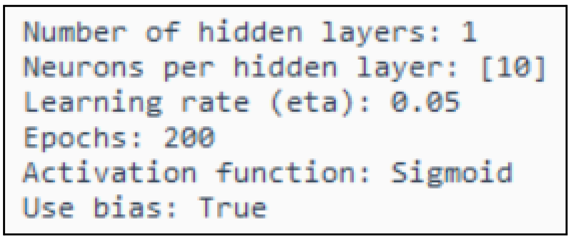
```
Actual \ Predicted    Class 1      Class 2      Class 3
Class 1               20           0            0
Class 2               5            15           0
Class 3               0            1            19
```

Observations:

- Despite a deeper network and more epochs, performance (90.00% test accuracy) was lower than simpler models, suggesting potential underfitting or the negative impact of omitting bias terms.

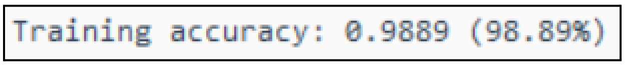
Experiment 4

Network Configuration:

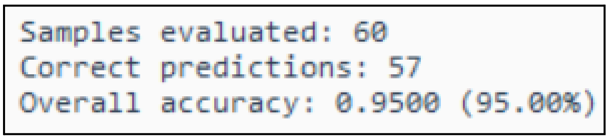


```
Number of hidden layers: 1
Neurons per hidden layer: [10]
Learning rate (eta): 0.05
Epochs: 200
Activation function: Sigmoid
Use bias: True
```

Results:

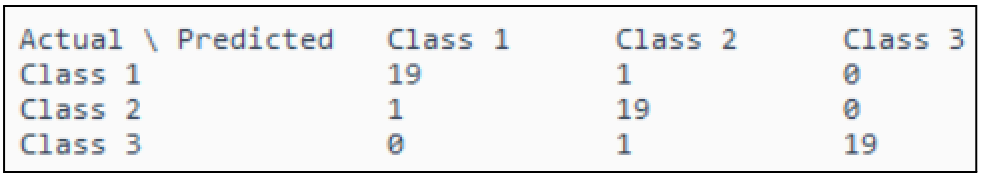


```
Training accuracy: 0.9889 (98.89%)
```



```
Samples evaluated: 60
Correct predictions: 57
Overall accuracy: 0.9500 (95.00%)
```

Confusion Matrix:



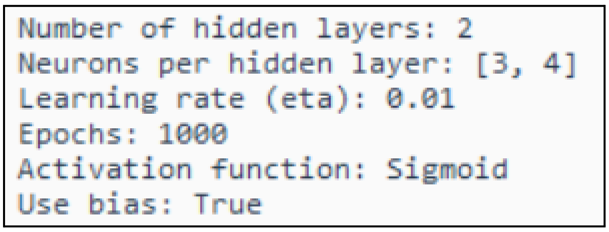
```
Actual \ Predicted    Class 1      Class 2      Class 3
Class 1               19           1            0
Class 2               1            19           0
Class 3               0            1            19
```

Observations:

- Achieved high training (98.89%) and test accuracy (95.00%) with a single hidden layer, demonstrating effective learning despite fewer parameters compared to deeper architectures.

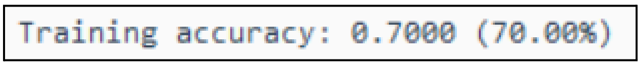
Experiment 5

Network Configuration:

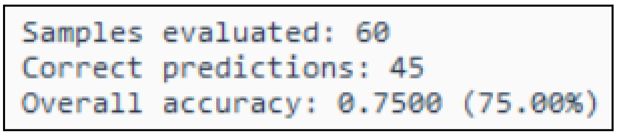


```
Number of hidden layers: 2
Neurons per hidden layer: [3, 4]
Learning rate (eta): 0.01
Epochs: 1000
Activation function: Sigmoid
Use bias: True
```

Results:

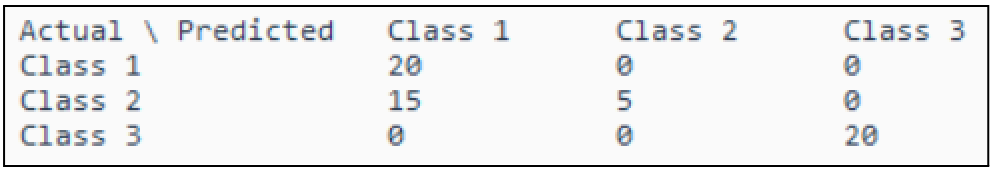


```
Training accuracy: 0.7000 (70.00%)
```



```
Samples evaluated: 60
Correct predictions: 45
Overall accuracy: 0.7500 (75.00%)
```

Confusion Matrix:



```
Actual \ Predicted   Class 1     Class 2     Class 3
Class 1              20          0           0
Class 2              15          5           0
Class 3              0           0           20
```

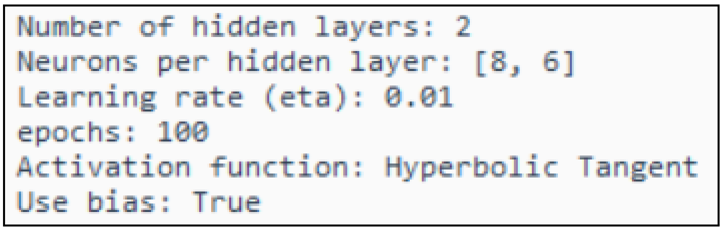
Observations:

- The network struggled (75% test accuracy) because the hidden layers were too small to capture the data's complexity, especially confusing Class 2 with Class 1.

2. Hyperbolic Tangent Activation

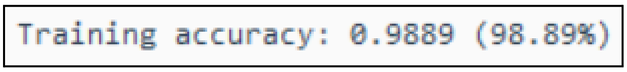
Experiment 1

Network Configuration:

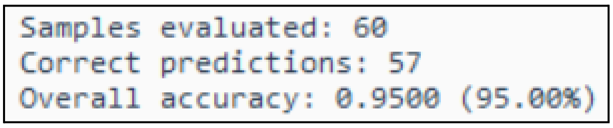


```
Number of hidden layers: 2
Neurons per hidden layer: [8, 6]
Learning rate (eta): 0.01
epochs: 100
Activation function: Hyperbolic Tangent
Use bias: True
```

Results:

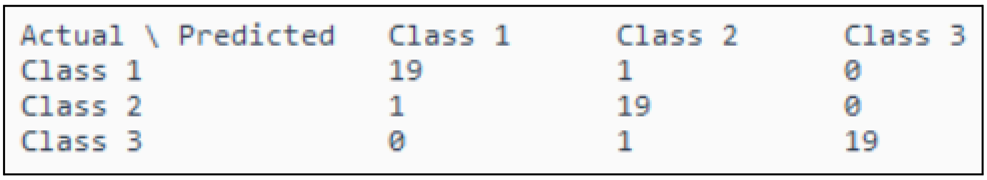


```
Training accuracy: 0.9889 (98.89%)
```



```
Samples evaluated: 60
Correct predictions: 57
Overall accuracy: 0.9500 (95.00%)
```

Confusion Matrix:



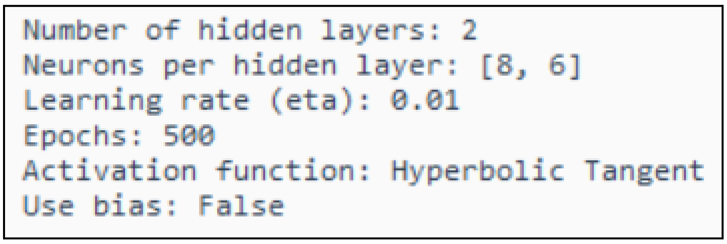
```
Actual \ Predicted    Class 1      Class 2      Class 3
Class 1               19           1            0
Class 2               1            19           0
Class 3               0            1            19
```

Observations:

- Achieved high test accuracy (95.00%) with a low learning rate and bias, showing Tanh can perform well even with limited training epochs.

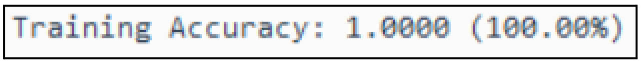
Experiment 2

Network Configuration:

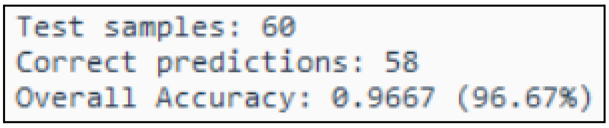


```
Number of hidden layers: 2
Neurons per hidden layer: [8, 6]
Learning rate (eta): 0.01
Epochs: 500
Activation function: Hyperbolic Tangent
Use bias: False
```

Results:

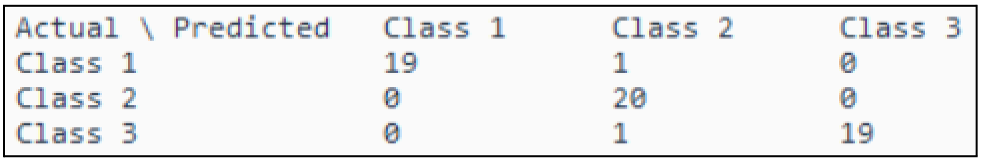


```
Training Accuracy: 1.0000 (100.00%)
```



```
Test samples: 60
Correct predictions: 58
Overall Accuracy: 0.9667 (96.67%)
```

Confusion Matrix:



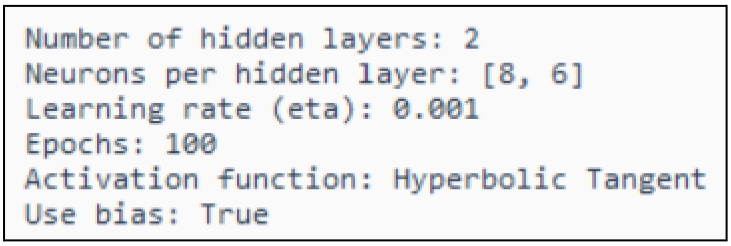
```
Actual \ Predicted   Class 1      Class 2      Class 3
Class 1              19           1            0
Class 2              0            20           0
Class 3              0            1            19
```

Observations:

- Despite omitting bias, the model achieved perfect training and near-perfect test accuracy (96.67%), suggesting that the symmetric nature of the Tanh activation may compensate for the lack of bias in this specific setup.

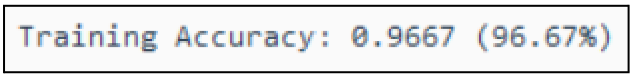
Experiment 3

Network Configuration:

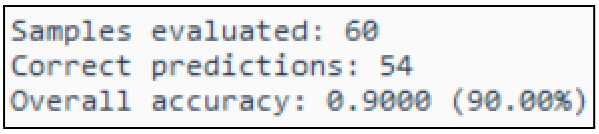


```
Number of hidden layers: 2
Neurons per hidden layer: [8, 6]
Learning rate (eta): 0.001
Epochs: 100
Activation function: Hyperbolic Tangent
Use bias: True
```

Results:

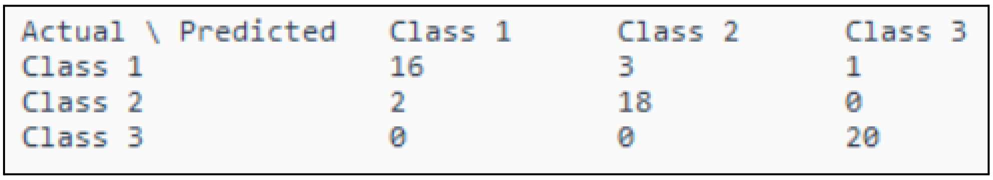


```
Training Accuracy: 0.9667 (96.67%)
```



```
Samples evaluated: 60
Correct predictions: 54
Overall accuracy: 0.9000 (90.00%)
```

Confusion Matrix:



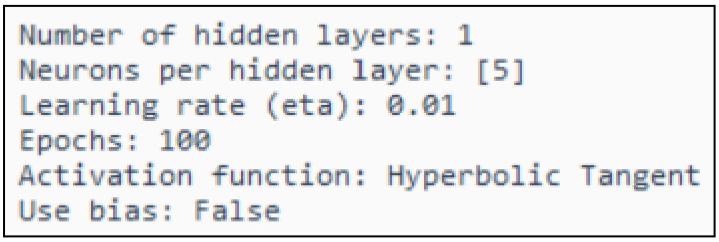
```
Actual \ Predicted   Class 1      Class 2      Class 3
Class 1              16           3            1
Class 2              2            18           0
Class 3              0            0            20
```

Observations:

- A very low learning rate (0.001) led to lower performance (90.00% test accuracy), indicating that Tanh networks may require careful tuning of the learning rate to avoid slow or suboptimal convergence.

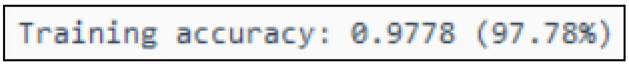
Experiment 4

Network Configuration:

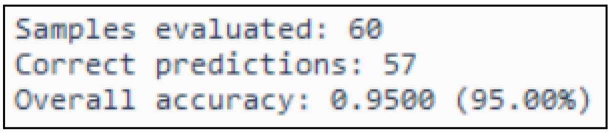


```
Number of hidden layers: 1
Neurons per hidden layer: [5]
Learning rate (eta): 0.01
Epochs: 100
Activation function: Hyperbolic Tangent
Use bias: False
```

Results:

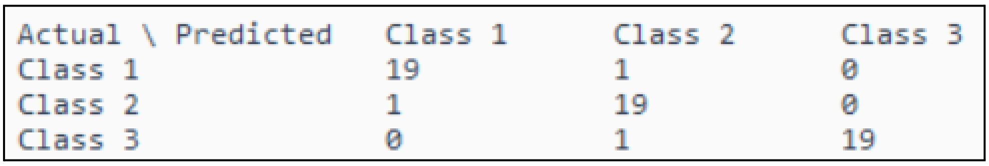


```
Training accuracy: 0.9778 (97.78%)
```



```
Samples evaluated: 60
Correct predictions: 57
Overall accuracy: 0.9500 (95.00%)
```

Confusion Matrix:



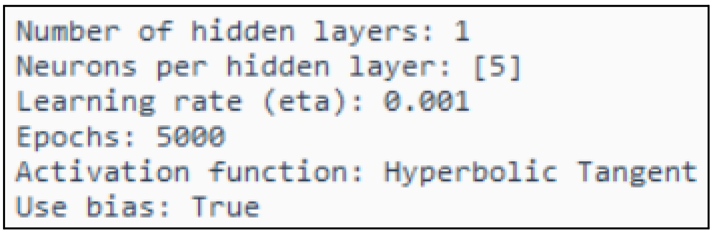
```
Actual \ Predicted   Class 1      Class 2      Class 3
Class 1              19           1            0
Class 2              1            19           0
Class 3              0            1            19
```

Observations:

- A simpler single-layer network (5 neurons) matched the performance of deeper models (95.00% test accuracy), demonstrating that for this task, a shallower Tanh architecture can be equally effective.

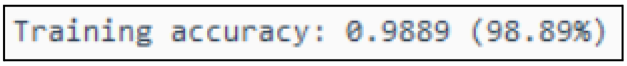
Experiment 5

Network Configuration:

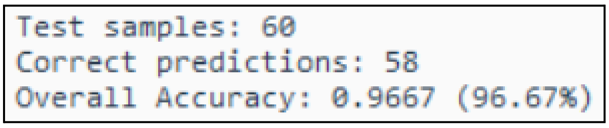


```
Number of hidden layers: 1
Neurons per hidden layer: [5]
Learning rate (eta): 0.001
Epochs: 5000
Activation function: Hyperbolic Tangent
Use bias: True
```

Results:

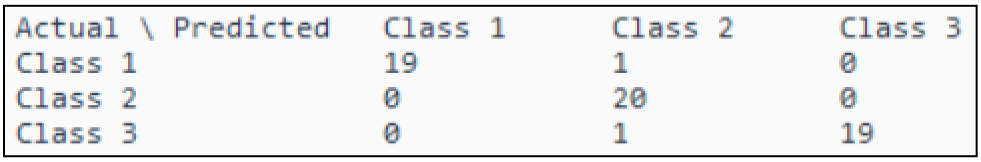


```
Training accuracy: 0.9889 (98.89%)
```



```
Test samples: 60
Correct predictions: 58
Overall Accuracy: 0.9667 (96.67%)
```

Confusion Matrix:



```
Actual \ Predicted    Class 1      Class 2      Class 3
Class 1               19           1            0
Class 2               0            20           0
Class 3               0            1            19
```

Observations:

- A single hidden layer with just 5 neurons achieved strong performance (96.67% test accuracy) using Tanh and a very low learning rate — showing that even simple networks can generalize well with sufficient training time.

Best Accuracies:

Activation Function	Train Accuracy (%)	Test Accuracy (%)	Learning Rate	Epochs	#Layers	Neurons per Layer	Using bias
Sigmoid	100	96.67	0.1	500	2	8,6	✔
Tanh	100	96.67	0.01	500	2	8,6	✖

Observation:

- The identical peak performance of Sigmoid and Tanh highlights that, for this task, network architecture and training parameters (like epochs and learning rate) are more critical than the choice of activation function — though Tanh's success without bias hints at its potential efficiency in simpler setups.

Conclusion

- The network’s performance is significantly influenced by hyperparameters such as learning rate, number of epochs, network architecture, and use of bias.

- Increasing the number of epochs generally improves accuracy, especially for networks with more complex architectures.

- Including bias in the network helps improve classification performance, particularly for underrepresented classes.

- A higher learning rate can speed up convergence but may require careful tuning to avoid instability.

- Comparing activation functions, both Sigmoid and Hyperbolic Tangent (Tanh) achieved similar best accuracies; however, Tanh reached high accuracy with fewer epochs and a simpler network.

- Overall, careful selection of activation function, learning rate, number of epochs, and network size is essential to achieve optimal performance.