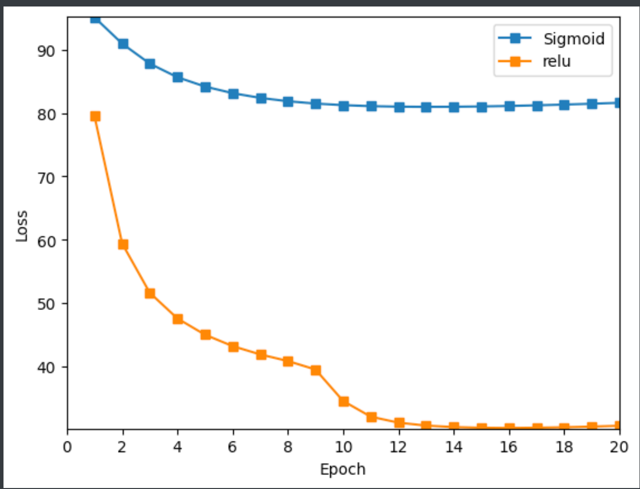
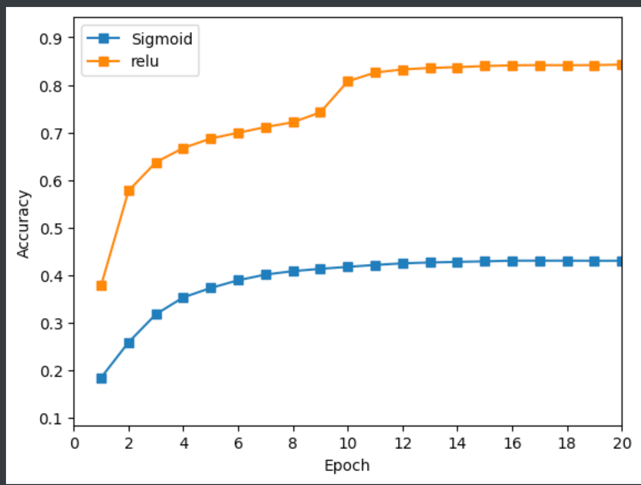
Experimental Report

Graphs

MLP with Euclidean Loss

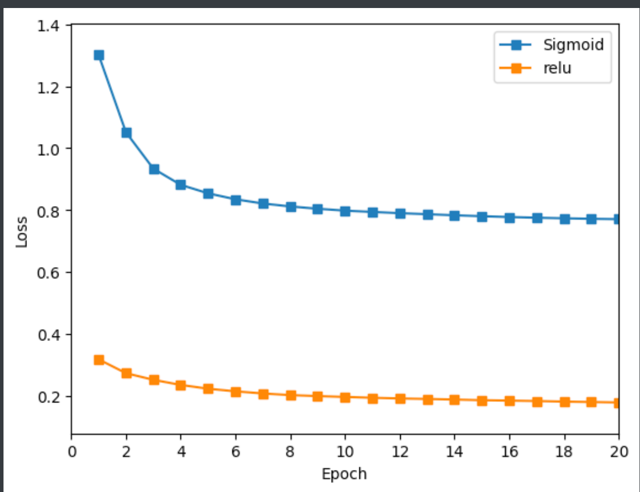


Comparison of Loss in each Epoch

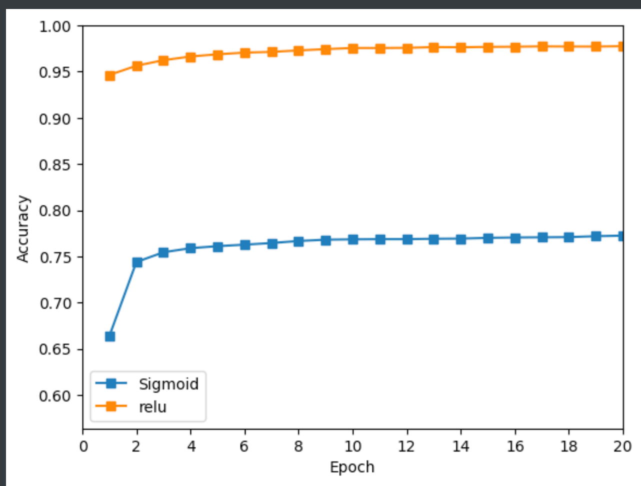


Comparison of Accuracy in each Epoch

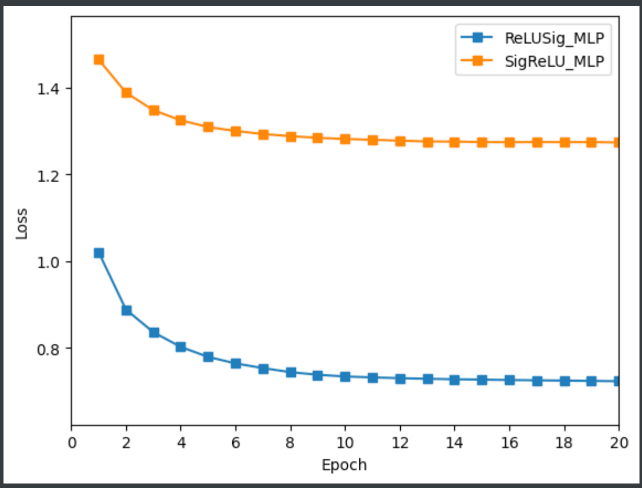
MLP with Softmax Cross-Entropy Loss



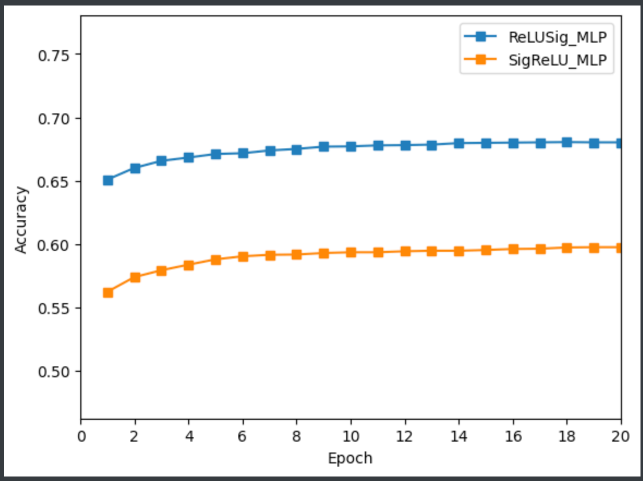
Comparison of Loss in each Epoch



Comparison of Accuracy in each Epoch



Comparison of Loss in each Epoch



Comparison of Accuracy in each Epoch

Accuracies Among Different Models

Loss Function	Activation Function	Aver. Training Acc	Aver. Validation Acc	Final Test Acc
Euclidean	Sigmoid	0.4150	0.4300	0.4321
Euclidean	ReLU	0.8155	0.8426	0.8270
Softmax	Sigmoid	0.7601	0.7724	0.7644
Softmax	ReLU	0.9813	0.9776	0.9735
Softmax	ReLU + Sigmoid	0.6847	0.6802	0.6787
Softmax	Sigmoid + ReLU	0.5932	0.5974	0.5962

Comparison between Different Learning Rates (Euclidean + ReLU)

Learning Rate	Aver. Training Acc	Aver. Validation Acc	Final Test Acc
0.001	0.8155	0.8426	0.8270
0.01	0.9817	0.9782	0.9729
0.05	0.5685	0.5856	0.5784

Comparison between Different Batch Sizes (Euclidean + ReLU)

Batch Size	Aver. Training Acc	Aver. Validation Acc	Final Test Acc
50	0.7240	0.7448	0.7315
100	0.8155	0.8426	0.8270
200	0.7222	0.7488	0.7280

Conclusion

From the data and graphs listed above, we can see that

- ReLU converges significantly faster than Sigmoid. Also, ReLU+Softmax performs best in this problem. (Convergence + Accuracy)
- In this problem, softmax performs better than euclidean in all cases; ReLU is better than Sigmoid in all cases.
- Sometimes the depth of a neural network doesn't necessarily lead to higher accuracy rates.
- The level of Learning Rate does not necessarily determine the level of accuracy, but more attempts may unexpectedly and dramatically improve the accuracy.
- It is possible to improve performance regardless of whether the Batch Size is small or large.