Q1

Initializing all weights to zero creates a symmetry problem. In this instance, all neurons in a layer compute identical outputs and receive identical gradients during backpropagation. This prevents neurons from specializing, and the network fails to learn. Random initialization breaks this symmetry, allowing each neuron to learn different features. This diversity allows neurons to learn distinct, non-redundant features, enabling the network to capture more detailed representation of the data. This is essential for learning complex patterns and achieving better generalization.

Q2

Neural networks solve the XOR problem by using hidden layers with nonlinear activation functions. Unlike logistic regression, which can only create linear decision boundaries, neural networks learn complex, nonlinear relationships through multi-layer structures.

In the XOR problem, the classes are not linearly separable, so a single-layer network cannot solve it. However, a network with at least one hidden layer can. The hidden layer learns intermediate features that transform the input space, making the XOR patterns linearly separable in this new representation, and the output layer can then correctly classify. This ability to learn such representations through a hidden layer allows neural networks to act as universal function approximators, capable of learning complex decision boundaries.

Q3

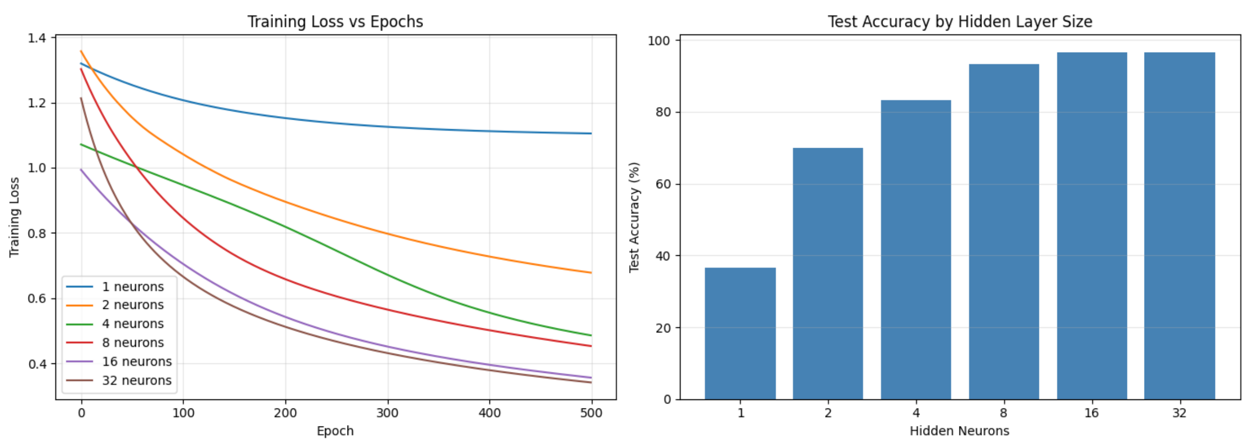

Figure 1: Training Loss vs Epochs and Test Accuracies for varying hidden layer sizes

The results show that model size strongly influences performance. The one-neuron network barely reduces the loss and plateaus around 40% accuracy, demonstrating severe underfitting. With only one hidden unit, the network can only carve out a single decision boundary (essentially the same capacity as plain logistic regression) so it fails to separate the three Iris classes effectively. Adding a second neuron improves accuracy significantly to around 70%, with a gently sloping loss curve. This proves that even a small increase in model size helps, though the network remains barely expressive enough. Gradients push weights in the right direction, but plateau before the classes separate cleanly.

At four neurons the loss curve bends further downward and finishes near 0.55, and the test accuracy jumps to around -80% As the number of hidden neurons is increased to 8 and beyond, the models show a significant decrease in training loss, as shown by the increasingly steeper curves in the graph. These models demonstrate a better capacity to learn the training data's underlying complexities.

When comparing to the previous classification lab, the one-vs-all logistic regression classifier with softmax achieved 100% test accuracy on the same Iris split. The smaller MLPs (1–2 neurons) fall far short of this baseline, while intermediate sizes (4–8 neurons) approach it. Larger networks (16–32 neurons) nearly match the logistic regression performance but don't exceed it, suggesting that for this relatively simple dataset, the added complexity of multiple hidden layers provides diminishing returns.

To conclude, model size drives performance. Too few hidden neurons (1–2) under-fit badly, and wider layers (8–32) push accuracy higher while converging faster. Beyond 16 neurons, model size increases show diminishing returns