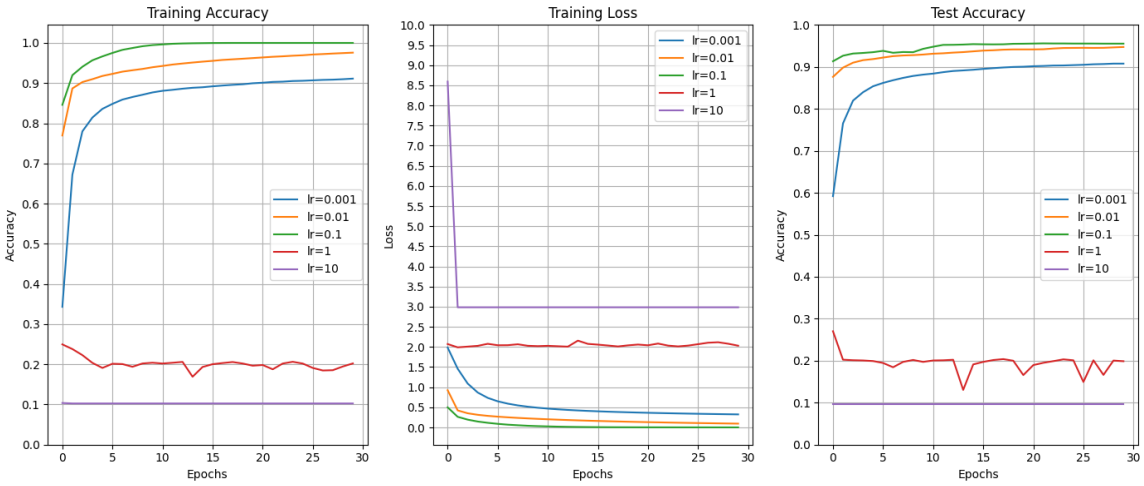
Introduction To ML
Neural Networks

a. Coding section.

b. What happens when the learning rate is too small or too large? Explain the phenomenon.

Here are the plots:



training accuracy: When the learning rate is too small, training is slow and can get stuck in suboptimal solutions and takes a lot of epochs to converge. On the other hand, a rate that's too large causes erratic training and poor accuracy as we can see in the red curve at the left plot.

Test accuracy: The graphs show that with 0.1, the network steadily improves and exceeds 93% test accuracy. Lower rates (0.001, 0.01) result in slower convergence, and higher rates (1, 10) cause unstable training, demonstrating that an optimal rate is crucial for effective training.

c. Now train the network on the whole training set and test on the whole test set. What is the test accuracy on the test set in the final epoch?

We obtained the best performance in the previous section at a learning rate of 0.1 so we will continue with that. Here is the code:

```python
x_train, y_train, x_test, y_test = load_as_matrix_with_labels(60000, 60000)

net = Network([784, 40, 10])
learning_rate = 0.1
epochs = 30
batch_size = 10

res = net.train(x_train, y_train, epochs=epochs, batch_size=batch_size, learning_rate=learning_rate, x_test=x_test, y_test=y_test)
parameters, train_losses, test_losses, train_accuracies, test_accuracies = res

print(f"Final test accuracy: {test_accuracies[-1]:.4f}")
```

And here is the output:

```
Final test accuracy: 0.9659
```