1. Initial weights : Random float between 0 and 0.1:

- An extremely small value seemed like a good start for my initial weights, the reason for this is because when a weight changed its value, it was by an extremely small amount. By reducing the size of the weights, the network would learn faster as it was not on either extreme of the gradient descent

2. Learning rate: 1.0

- For my learning rate, I chose an extremely large number at 1.0. The reason I chose this number was because it was able to get above 90% accuracy in less epochs than smaller alpha values.
- I think this was because the backpropagation algorithm changed the weights by such a small value already (which helped avoid overshooting), so having a smaller alpha value would just slow down the process in finding more optimal weights

3. Number of epochs ranged between 100-150 (117 epochs for my best test of 82.6% accuracy)

4. Test accuracy: between 70% and 82.6% from what I tested

If we look at the 82.6% accuracy one,

1s classified= 16 /16 = 100% accuracy

8s classified = 10/15 = 66.6% accuracy

9s classified = 12/15 = 80% accuracy