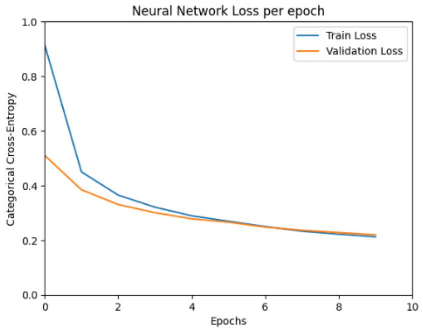
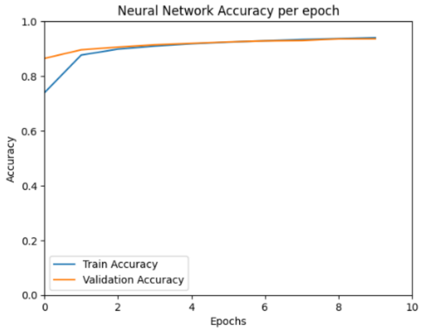
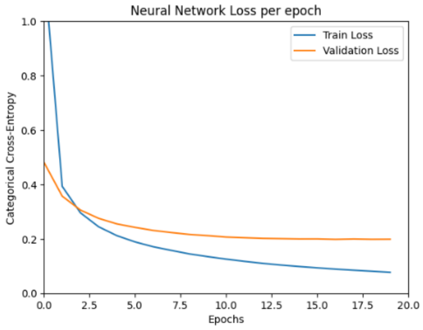
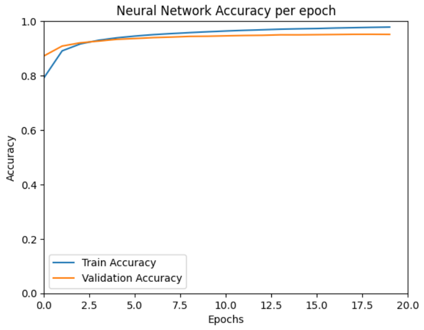
Miltiades Lefkopoulos

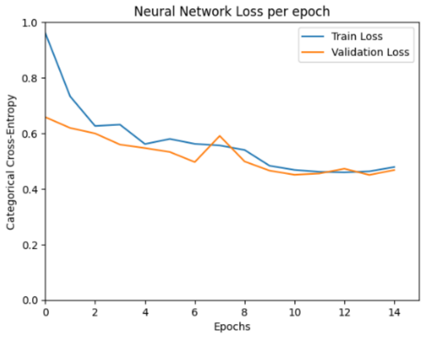
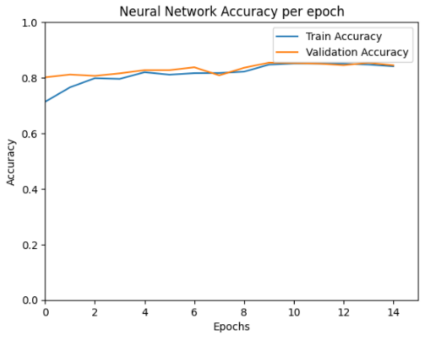
ARTIFICIAL INTELLIGENCE

1) Hyper-parameter Tuning:

Examples	**Neural Network Loss per epoch**	**Neural Network Accuracy per epoch**

activation = tanh
learning_rate = 0.001
epochs = 10
Shape → 784
hidden-1 → 256 units
hidden-2 → 256 units
	
	
		

activation = relu
learning_rate = 0.001
epochs = 20
Shape → 784
hidden-1 → 128 units
hidden-2 → 64 units
	
	
		

activation = than learning_rate = 0.01 epochs = 15 Shape → 784 hidden-1 → 64 units hidden-2 → 64 units hidden-3 → 32 units	 	

2) Improvements

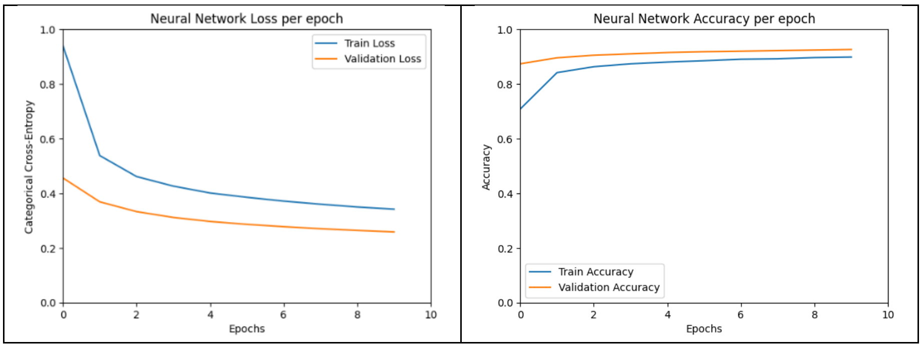
To improve neural network performance beyond ***Hyper-parameter Tuning***, I implemented some improvements to the training process.

First, I normalized the input images so that the pixel values range in the range [0, 1] so that the neural network learns more efficiently.

I then added Batch Normalization after each dense level to stabilize and speed up the training through the normalization of intermediate outputs.

In addition, I put Dropout at a rate of 0.2 to reduce overtraining by randomly disabling neurons during training.

Neural Network Loss per epoch	**Neural Network Accuracy per epoch**



3) ***Comprehension Questions**:*

a) MNIST's data has been shown to be effective in training this model, but due to the limitation to 28x28 images the application cannot proceed to use for larger, more complex real-world images.

b) Not all pixels are necessary for prediction as a subset contains the information (the white pixels) that make up the number.

c) Deep neural networks should be used when the data is large, complex and unstructured. They also help when there are patterns in the data that are not visible.

d) Yes, deep learning can be used in all 3 branches of Machine Learning (Supervised Learning, Unsupervised Learning and Reinforcement Learning), but with the particularity that it is more complex and requires more resources.

Github Repository

Google Colab Notebook