### **MNIST (Neural Network)**

#### **What I Learned:**

* **Processing Data Before Model Training:** Images need to be converted into vectors (flattened) to be used as input in a neural network model. For example, a 28x28 image becomes a 784-dimensional vector.
* **Overfitting Prevention:** In case of overfitting, techniques like BatchNormalization and Dropout can help to reduce it by normalizing activations and randomly turning off units during training, respectively.
* **Regularization Techniques:**
* **L1 Regularization (Lasso):** Can be used to prevent overfitting by adding a penalty for large weights, promoting sparsity in the model.
* **L2 Regularization (Ridge):** Adds a penalty for large weights but doesn’t enforce sparsity. It encourages smaller weight values.
* **Optimizer:** The Adam optimizer is commonly used, as it adapts the learning rate based on the gradient and is efficient for most models.
* **Loss Function:**loss=sparse\_categorical\_crossentropy: This loss function is used when the target labels are integers representing classes (instead of one-hot encoded labels).
* **Metrics:** Metrics like accuracy, precision, and recall are used to evaluate model performance. The model will calculate these for each epoch.
* **Confusion Matrix:** After training, use a confusion matrix to analyze the model’s predictions versus true labels.
* **Hyperparameters to Tune:** Experimenting with the number of epochs, layers, hidden units, and batch size can have a significant effect on model accuracy.