ICP – 5

**Bhavana Parasa**

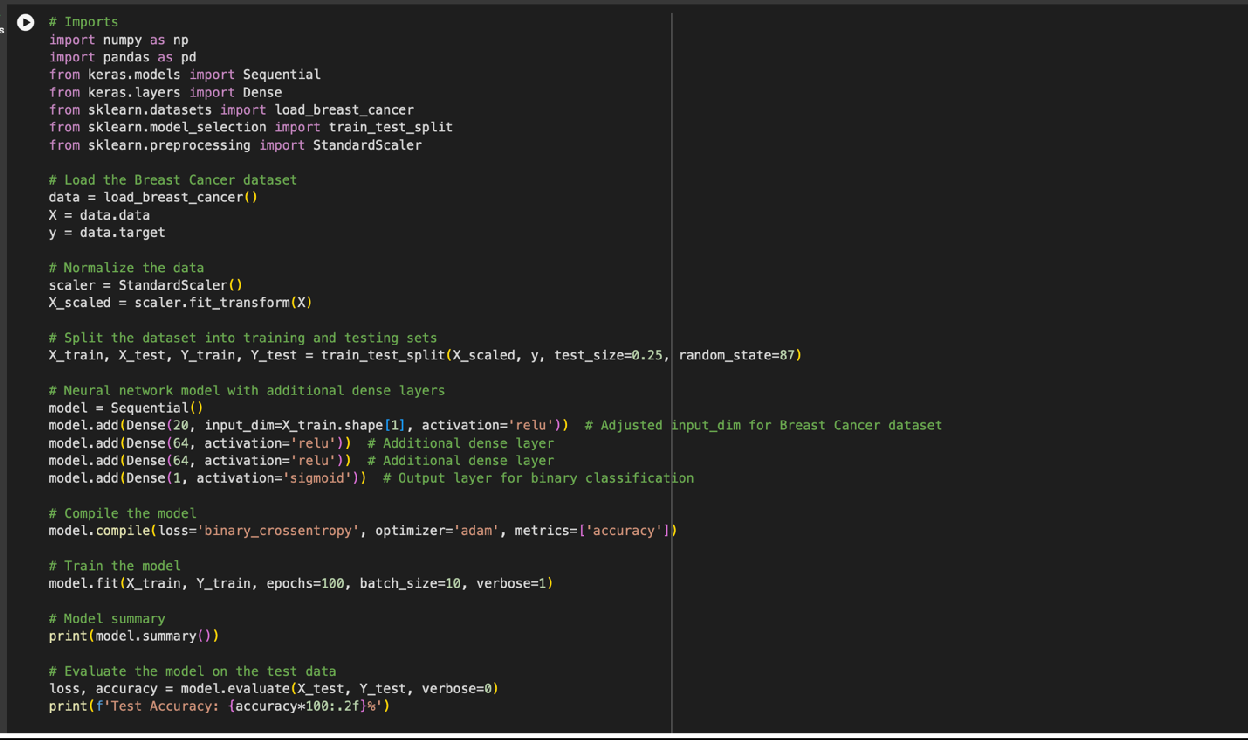
**700761185**

**GitHub Link:** [**https://github.com/BXP11850/Neural-Networks-Deep-Learning-Assignments-BXP11850/tree/main/Assignment%20-4**](https://github.com/BXP11850/Neural-Networks-Deep-Learning-Assignments-BXP11850/tree/main/Assignment%20-4)

**Video link :** [**https://github.com/BXP11850/Neural-Networks-Deep-Learning-Assignments-BXP11850/blob/main/Assignment%20-4/700761185.mp4**](https://github.com/BXP11850/Neural-Networks-Deep-Learning-Assignments-BXP11850/blob/main/Assignment%20-4/700761185.mp4)

**Q-1:**

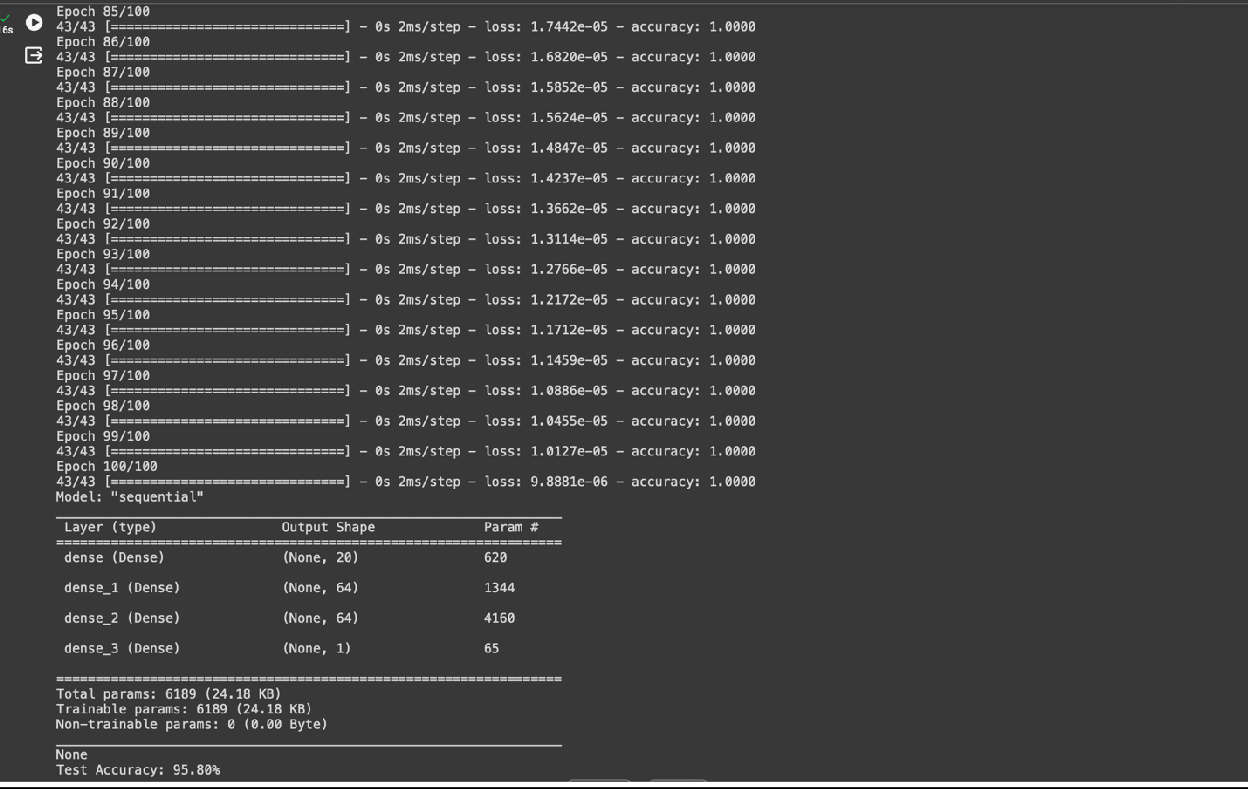
**CODE:**

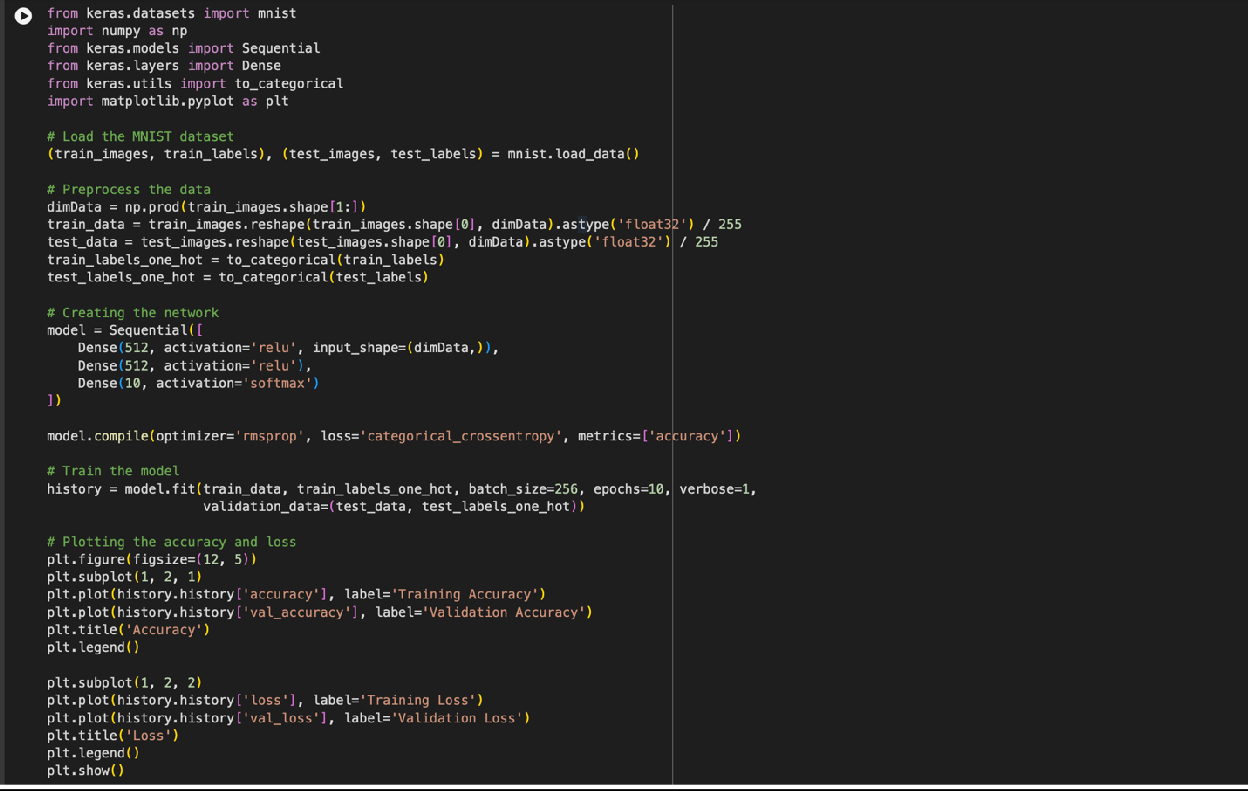


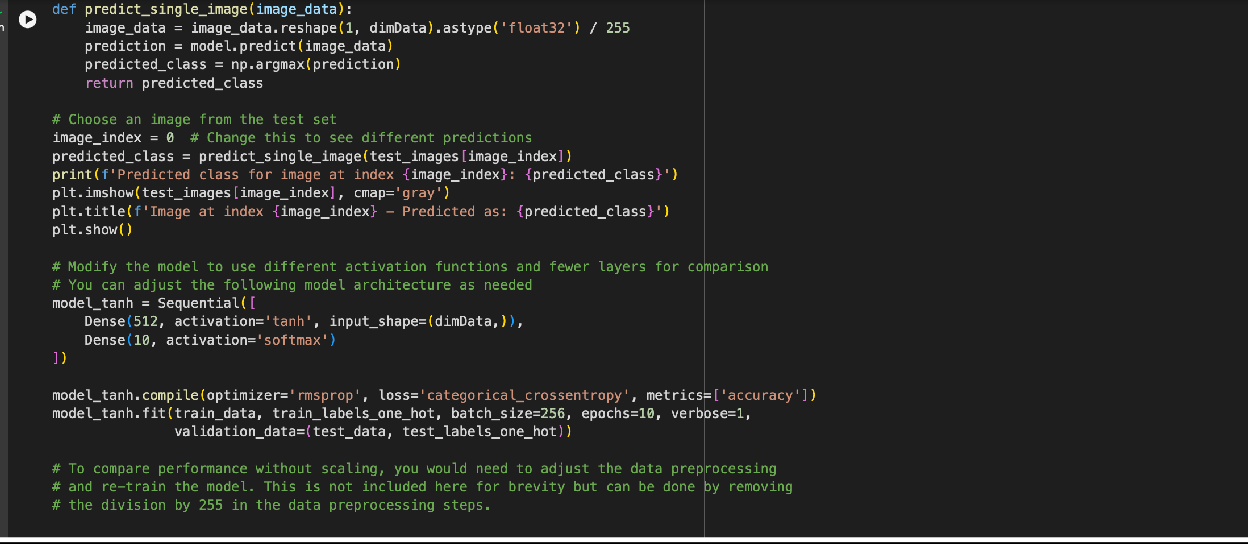
**EXPLANATION:**

1. Dataset Loading: It loads the Breast Cancer dataset from scikit-learn, which contains features for breast cancer diagnostic data and binary labels indicating malignancy.
2. Data Preprocessing: The features are normalized using `StandardScaler` to scale the data, ensuring that the neural network model receives input data within a manageable range, improving training stability and performance.
3. Dataset Splitting: The normalized data is split into training and testing sets, with 75% used for training and 25% for testing, facilitated by the `train\_test\_split` method from scikit-learn.
4. Model Construction: A `Sequential` model is defined with four layers:
   * An input dense layer with 20 neurons and ReLU activation, adjusted to match the number of features in the Breast Cancer dataset.
   * Two additional dense layers, each with 64 neurons and ReLU activation, to increase the model's capacity to learn complex patterns.
   * An output layer with a single neuron and sigmoid activation to predict the binary outcome (malignant or benign).
5. Model Compilation: The model is compiled with the Adam optimizer, binary crossentropy as the loss function, and accuracy as the metric to evaluate performance.
6. Model Training: The model is trained for 100 epochs with a batch size of 10 on the normalized and split training data, with verbosity set to 1 to display training progress.
7. Evaluation: After training, the model's performance is evaluated on the test set, and the test accuracy is printed, indicating how well the model can predict breast cancer malignancy.
8. Summary Output: Finally, a summary of the model's architecture is printed, detailing the configuration of each layer, including the number of parameters in the model.

**OUTPUT:**



**Q-2: CODE:**



**EXPLANATION:**

The provided script is a comprehensive Python program for training a neural network model on the MNIST dataset for handwritten digit recognition. It highlights key steps in data preprocessing, model training, evaluation, and experimentation with model architecture. Here's a summary in a few points:

1. MNIST Dataset: The script loads the MNIST dataset, which consists of 28x28 pixel grayscale images of handwritten digits (0 through 9) and their associated labels.
2. Data Preprocessing: The images are flattened from a 2D 28x28 format to a 1D 784 vector and normalized so that pixel values are in the range [0, 1]. Labels are converted to one-hot encoded vectors for classification.
3. Model Architecture: A `Sequential` model with two dense layers of 512 neurons each (using ReLU activation), followed by a softmax output layer for classifying the digits into 10 categories, is defined and compiled with RMSprop optimizer and categorical crossentropy loss.
4. Training and Validation: The model is trained for 10 epochs with a batch size of 256, using both training and validation data to monitor performance and avoid overfitting.
5. Performance Visualization: Training and validation accuracy and loss are plotted using matplotlib to visually assess the model's learning progress over epochs.
6. Prediction on Test Data: The script includes a function to predict the digit class for a single image from the test set, demonstrating the model's inferencing capability.
7. Model Experimentation: A variant of the model using tanh activation and a simplified architecture (one hidden layer) is trained to explore the impact of different network configurations on performance.
8. Note on Scaling: A comment mentions the significance of scaling image pixel values for model performance, suggesting an experiment to compare results with and without this preprocessing step.

This script serves as a complete workflow for neural network training on image data, encompassing data handling, model building, training, evaluation, and experimentation with architecture modifications.

**OUTPUT:**

