**Assignment 4**

**Student id:700741326**

**Triveni Bala**

**1. Add one more hidden layer to autoencoder**

**2. Do the prediction on the test data and then visualize one of the reconstructed version of that test data. Also, visualize the same test data before reconstruction using Matplotlib**

**3. Repeat the question 2 on the denoisening autoencoder**

**4. plot loss and accuracy using the history object**

1. In this step, we modify the autoencoder architecture to include an additional hidden layer. This new layer will be placed after the original encoded layer and before the decoder part. The number of nodes in this new layer is chosen to be 64 with the ReLU activation function.

**A screenshot of a computer

Description automatically generated**

After training the autoencoder, we perform predictions on the test data and visualize one randomly selected reconstructed image along with its original image using Matplotlib.

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer screen

Description automatically generated**

We use the trained autoencoder to predict the reconstructed images from the test data. Then, we randomly choose one test image, display the original image using plt.imshow(), and show the corresponding reconstructed image using the same function.

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

We monitor the loss during the training process and plot the training and validation losses to visualize how the autoencoder's performance changes over epochs.

**A screenshot of a computer

Description automatically generated**

**A screenshot of a graph

Description automatically generated**

During training, we store the training and validation loss in the history object. Then, we access the loss values for each epoch using history. history['loss'] and history. history['val\_loss']. We plot these values using Matplotlib to observe how the autoencoder's loss changes over the training process, which can provide insights into the model's performance and overfitting tendencies.

Github Link : https://github.com/TriveniBala/Neural-Networks-Assignment-4