AML Assignment-2

Name	ID
Ananya Sinha	MDS202307
Divyanshi Kumari	MDS202322
Rohit Roy	MDS202340

Question 1

SMS Spam Classifier - RNN & LSTM

The whole SMS messages corpus was tokenized keeping the top 5000 words as vocabulary. The tokenized messages was then converted to sequences, and a maximum length of 150 words was taken for each SMS message, padding the messages of shorter length. An 80-20 train-test split was taken.

The **RNN Model** was defined sequentially with an embedding layer with dimension 50, a RNN layer with 32 units, a fully connected layer with 256 units and ReLU activation, a dropout layer with rate 0.25, and finally an output layer with sigmoid activation to predict if the message was spam or not. It was trained using Adam optimizer with binary cross-entropy loss for 10 epochs. The total parameters for this is 261,361. The accuracy of this model for the test set came out to be 98.39% with 98% precision.

The **LSTM Model** was also defined similarly with an embedding layer with dimension 50, a LSTM layer with 32 units, and finally an output layer with sigmoid activation for prediction. The training steps were same as that of the RNN model. The total parameters came out to be 260,657. The accuracy was 98.48% with 99% precision.

Question 2

SMS Text Generation - RNN & LSTM

The SMS dataframe was converted to a text file with an end character after every message. The vocabulary is all the unique characters that can be found in the text. The training dataset was then created by sliding a window of 100 characters and the label being the next character of the sequence over the whole dataset.

The **RNN Model** is defined with a RNN layer of 278 units keeping a dropout layer with 0.2 rate. Then finally an output layer was created with units same as length of the vocabulary, with softmax activation. The model was trained with RMSProp optimizer with categorical cross-entropy loss, with an inital learning rate of 0.01, while defining a callback to the best model which will reduce the learning rate by factor of 0.2 if the loss doesn't improve. It was trained for 30 epochs. The number of parameters is 142,731.

The **LSTM Model** is defined with a LSTM layer with 128 units without dropout, keeping everything else the exact same as the RNN model. It was also trained for 30 epochs, and the number of parameters was 141,045.

On comparing the results, it was clear that LSTM performed better than RNN in every case.

Question 3

FashionMNIST Image Generation - Conditional GAN

The dataset is loaded from CSV files and is processed in batches of 64. The dataset class FashionMNIST handles loading and preprocessing images, applying transformations including normalization to input to the model.

The **generator**, initialized with a random latent vector of size 100, incorporates a class embedding for the label, allowing the model to create images specific to each category. The generator then passes this concatenated input through several dense layers of increasing size (256, 512, and 1024 neurons), each equipped with LeakyReLU activation, ultimately outputting a 28x28 image.

The **discriminator**, responsible for distinguishing real images from generated ones, uses a similar class embedding to incorporate the label information. The combined image and label information is passed through three layers (1024, 512, and 256 neurons), also with LeakyReLU activation to classify the image as real or fake, with dropout layers helping prevent overfitting.

Training is conducted over 30 epochs with a learning rate of 0.0001. The generator training function optimizes the generator to produce images that the discriminator classifies as "real", while discriminator training function trains the discriminator to distinguish between real and generated images accurately. Both losses are optimized using Adam optimizers.

After the training is complete, we see that pretty decent images are generated, and are classifiable into the given class.