ASSIGNMENT

Part 1 : Use an artificial neural network (ANN) to classify whether a customer will accept a personal loan offer. You will use the Python programming language and the Keras library to build and train your ANN model. You will also evaluate the performance of your model using various metrics.

Instructions:

1. Use the dataset which is provided
2. Split the dataset into a training set and a test set. The training set should be 80% of the dataset, and the test set should be 20% of the dataset.
3. Build an ANN model with the following architecture:
    - Input layer: 10 features
    - Hidden layer: 10 neurons
    - Output layer: 1 neuron
4. Choose a loss function and an optimizer. The most common loss function for classification is the cross-entropy loss. The most common optimizer for classification is the stochastic gradient descent (SGD) optimizer.
5. Train the ANN model for 100 epochs.
6. Evaluate the performance of the ANN model using the test set. The Keras library will calculate the accuracy, precision, and recall of the ANN model. The accuracy is the percentage of data points that were correctly classified. The precision is the percentage of data points that were classified as positive that were actually positive. The recall is the percentage of data points that were actually positive that were classified as positive.
7. Report the accuracy, precision, and recall of the ANN model.

Bonus:

1. Try different model architectures and hyperparameters to improve the performance of the ANN model.
2. Use the ANN model to classify whether a customer will accept a personal loan offer with different characteristics.

Part 2 : Use a convolutional neural network (CNN) to classify images of objects in the Tiny ImageNet Dataset. You will use the Python programming language and the Keras library to build and train your CNN model. You will also evaluate the performance of your model using various metrics.

Instructions:

1. Download the Tiny ImageNet Dataset from Kaggle.
2. Split the dataset into a training set and a test set. The training set should be 80% of the dataset, and the test set should be 20% of the dataset.
3. Build a CNN model with the following architecture:
    - Input layer: 64x64x3 (RGB images)
    - Convolutional layer 1: 32 filters, kernel size 3x3, stride 1, padding 'same'
    - Max pooling layer 1: pool size 2x2, stride 2
    - Convolutional layer 2: 64 filters, kernel size 3x3, stride 1, padding 'same'
    - Max pooling layer 2: pool size 2x2, stride 2
    - Flatten layer
    - Dense layer 1: 128 neurons
    - Dropout layer: 0.2
    - Dense layer 2: 200 neurons
    - Output layer: 1 neuron
4. Choose a loss function and an optimizer. The most common loss function for classification is the cross-entropy loss. The most common optimizer for classification is the stochastic gradient descent (SGD) optimizer.
5. Train the CNN model for 100 epochs.

Evaluate the performance of the CNN model using the test set. The Keras library will calculate the accuracy, precision, and recall of the CNN model. The accuracy is the percentage of data points that were correctly classified. The precision is the percentage of data points that were classified as positive that were actually positive. The recall is the percentage of data points that were actually positive that were classified as positive.

Report the accuracy, precision, and recall of the CNN model.

Bonus:

Try different model architectures and hyperparameters to improve the performance of the CNN model.

Use the CNN model to classify images of other objects, such as cars, or people.

Note : Instruction for downloading the dataset

1. Go to the Tiny ImageNet Dataset page on Kaggle: https://www.kaggle.com/c/tiny-imagenet
2. Click the "Download" button.
3. In the pop-up window, select the "Download all files" option.
4. Click the "Save" button.

The Tiny ImageNet Dataset will be downloaded to your computer in a zip file. Once the download is complete, extract the zip file to a directory of your choice.

The Tiny ImageNet Dataset is organized into two directories:

- train: This directory contains the training images.
- test: This directory contains the test images.

Each directory contains a subdirectory for each class. The subdirectory name is the name of the class. Each subdirectory contains the images for that class.

The images in the Tiny ImageNet Dataset are 64x64 pixels in size. They are stored in the JPEG format.

The Tiny ImageNet Dataset is a valuable resource for anyone interested in image classification. It is a small dataset, so it is easy to work with and it can be used to train models on a budget. The Tiny ImageNet Dataset is also a good dataset for benchmarking models.