

**1. Implementing Feedforward neural networks with Keras and TensorFlow.(Use MNIST Dataset)**

- a. Import the necessary packages**
- b. Load the training and testing data**
- c. Define the network architecture using Keras**
- d. Train the model using SGD**
- e. Evaluate the network**
- f. Plot the training loss and accuracy**

**2. Implementing Feedforward neural networks with Keras and TensorFlow.(Use CIFAR10 Dataset)**

- a. Import the necessary packages**
- b. Load the training and testing data**
- c. Define the network architecture using Keras**
- d. Train the model using SGD**
- e. Evaluate the network**
- f. Plot the training loss and accuracy**

**3. Build the Image classification model (Use MNIST Dataset)**

**by dividing the model into following 4 stages:**

- a. Loading and preprocessing the image data**
- b. Defining the model's architecture**
- c. Training the model**
- d. Estimating the model's performance**

**4. Build the Image classification model (Use ECG Dataset)**

**by dividing the model into following 4 stages:**

- a. Loading and preprocessing the image data**
- b. Defining the model's architecture**
- c. Training the model**
- d. Estimating the model's performance**

**5. Use Autoencoder to implement anomaly detection. (Use Credit Card Dataset)**

**Build the model by using:**

- a. Import required libraries**
- b. Upload / access the dataset**
- c. Encoder converts it into latent representation**
- d. Decoder networks convert it back to the original input**
- e. Compile the models with Optimizer, Loss, and Evaluation Metrics**

**6. Implement the Continuous Bag of Words (CBOW) Model.**

**Stages can be:**

- a. Data preparation**
- b. Generate training data**
- c. Train model**
- d. Output**

**7. Object detection using Transfer Learning of CNN architectures (Use CIFAR10 Dataset)**

- a. Load in a pre-trained CNN model trained on a large dataset**
- b. Freeze parameters (weights) in model's lower convolutional layers**
- c. Add custom classifier with several layers of trainable parameters to model**
- d. Train classifier layers on training data available for task**
- e. Fine-tune hyper parameters and unfreeze more layers as needed**