LP-IV Sample Problem Statement

1. Implementing Feedforward neural networks with Keras and TensorFlow for classification of hand-written MNIST dataset using below steps:
2. Import the necessary packages
3. Load the training and testing data
4. Define the network architecture using Keras
5. Train the model using SGD with 11 epochs
6. Evaluate the network
7. Plot the training loss and accuracy

(A2)

1. Implement the Image classification CNN model for classifying hand-written MNIST dataset by dividing the model into following 4 stages:
   1. Loading and preprocessing the image data
   2. Defining the model's architecture
   3. Training the model
   4. Estimating the model's performance

(A3 )

1. Build Feedforward neural networks with Keras and TensorFlow for classification of CIFAR10 image dataset using the following steps:
2. Import the necessary packages
3. Load the training and testing data
4. Define the network architecture using Keras
5. Train the model using SGD/Adam optimizer
6. Evaluate the network
7. Plot the training loss and accuracy
8. Implement the CNN model for classifying CIFAR10 image dataset by dividing the model into following 4 stages:
9. Loading and preprocessing the image data
10. Defining the model's architecture
11. Training the model
12. Estimating the model's performance
13. Implement anomaly detection for given credit card dataset using Autoencoder and build the model by using the following steps:
    1. Import required libraries
    2. Upload / access the dataset
    3. Encoder converts it into latent representation
    4. Decoder networks convert it back to the original input
    5. Compile the models with Optimizer, Loss, and Evaluation Metrics
14. Implement the Continuous Bag of Words (CBOW) Model for the given (textual document 1) using the below steps:
15. Data preparation
16. Generate training data
17. Train model
18. Output
19. Implement the Continuous Bag of Words (CBOW) Model for the given (textual document 2) using the below steps:
20. Data preparation
21. Generate training data
22. Train model
23. Output
24. Implement the Continuous Bag of Words (CBOW) Model for the given (textual document 3) using the below steps:
25. Data preparation
26. Generate training data
27. Train model
28. Output
29. Object detection using Transfer Learning of CNN architectures for the given (image dataset 1) using the below steps:
30. Load in a pre-trained CNN model trained on a large dataset
31. Freeze parameters (weights) in model's lower convolutional layers
32. Add custom classifier with several layers of trainable parameters to model
33. Train classifier layers on training data available for task
34. Fine-tune hyper parameters and unfreeze more layers as needed
35. Object detection using Transfer Learning of CNN architectures for the given (image dataset 2) using the below steps:
36. Load in a pre-trained CNN model trained on a large dataset
37. Freeze parameters (weights) in model's lower convolutional layers
38. Add custom classifier with several layers of trainable parameters to model
39. Train classifier layers on training data available for task
40. Fine-tune hyper parameters and unfreeze more layers as needed
41. Object detection using Transfer Learning of CNN architectures for the given (image dataset 3) using the below steps:
42. Load in a pre-trained CNN model trained on a large dataset
43. Freeze parameters (weights) in model's lower convolutional layers
44. Add custom classifier with several layers of trainable parameters to model
45. Train classifier layers on training data available for task
46. Fine-tune hyper parameters and unfreeze more layers as needed
47. Implementing Feedforward neural networks with Keras and TensorFlow
48. Import the necessary packages
49. Load the training and testing data (MNIST/CIFAR10)
50. Define the network architecture using Keras
51. Train the model using SGD
52. Evaluate the network
53. Plot the training loss and accuracy
54. Build the Image classification model by dividing the model into following 4 stages:
55. Loading and preprocessing the image data
56. Defining the model’s architecture
57. Training the model
58. Estimating the model’s performance
59. Use Autoencoder to implement anomaly detection on ecg dataset. Build the model by using:
60. Import required libraries
61. Upload / access the dataset
62. Encoder converts it into latent representation
63. Decoder networks convert it back to the original input
64. Compile the models with Optimizer, Loss, and Evaluation Metrics
65. Implement the Continuous Bag of Words (CBOW) Model. Stages can be:
66. Data preparation
67. Generate training data
68. Train model
69. Output
70. Object detection using Transfer Learning of CNN architectures
71. Load in a pre-trained CNN model trained on a large dataset
72. Freeze parameters (weights) in model’s lower convolutional layers
73. Add custom classifier with several layers of trainable parameters to model
74. Train classifier layers on training data available for task
75. Fine-tune hyper parameters and unfreeze more layers as needed