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