Task A: Design Image classification (Time: 15 mins)

1. The deep learning model design should have following layer sequence:
    1. Input →Conv2D → Pool2D → TanH→Conv2D →Pool2D → TanH → Conv2D→ Pool2D→ Relu→ Flatten→DENSE→Softmax

       (For Individual layer details looking into the following steps.)
2. Define input image data (ImageNet data) with the following details:
    1. Image size: 256, 256
    2. Number of channels: 3
    3. Tensor dimensionality: channel last (256x256x3)
    4. Number of classes: 10
    5. Data samples and corresponding labels are stored in python pickle.
        1. Train data pickle: train_A.pkl
        2. Test data pickle: test_A.pkl
3. Perform convolution2D with following parameters:
    1. Number of filters: 64
    2. Filter kernel size: [3,3]
    3. Stride sampling size: [1,1]
    4. Border Mode is valid
4. Perform Pooling2D layer with following parameters:
    1. Operation: Max
    2. Filter kernel size: [3,3]
    3. Stride sampling size: [1,1]
    4. Border Mode is valid
5. Perform TanH as activation function
6. Perform convolution2D with following parameters:
    1. Number of filter: 64
    2. Filter kernel size: [3,3]
    3. Stride sampling size: [1,1]
    4. Border Mode is valid
7. Perform Pooling2D layer with following parameters:

    1. Operation: Max
    2. Filter kernel size: [3,3]
    3. Stride sampling size: [1,1]
    4. Border Mode is valid
8. Perform TanH as activation function
9. Perform convolution2D with following parameters:
    1. Number of filter: 64
    2. Filter kernel size: [3,3]
    3. Stride sampling size: [1,1]
    4. Border Mode is valid
10. Perform Pooling2D layer with following parameters:
    1. Operation: Max
    2. Filter kernel size: [3,3]
    3. Stride sampling size: [1,1]
    4. Border Mode is valid
11. Perform ReLU as activation function
12. Flatten the Tensor output
13. Add a dense layer with 10 nodes.
14. Use Softmax as activation
15. Use Hinge as a Loss function
16. Use Accuracy as metric to predict the results on test data
17. Using SGD as an optimizer with the following parameters:
    1. Learning rate: 0.1
    2. Decay: 0.05
    3. Momentum: 0
18. The code should train a model with batch size =16, number of epochs= 10

Task B: Design Image classification (Time: 15mins)

1. The deep learning model design should have following layer sequence:

   Input →Conv2D → Relu → Conv2D → Relu→Pool2D →Conv2D → Relu →Conv2D → Relu →Conv2D → Relu →Pool2D → Flatten→DENSE→Softmax

   (For Individual layer details follow below steps.)

2. Define input image data (CIFAR10 data) with the following details:
    a. Image size: 32,32
    b. Number of channels: 3
    c. Tensor dimensionality: channel last (32x32x3)
    d. Number of classes: 10
    e. Data samples and corresponding labels are stored in python pickle.
        i. Train data pickle: train_B.pkl
        ii. Test data pickle: test_B.pkl
3. Perform convolution2D with following parameters:
    a. Number of filters: 96
    b. Filter kernel size: [5,5]
    c. Stride sampling size: [1,1]
    d. Border Mode is valid
4. Perform ReLU as activation function
5. Perform convolution2D with following parameters:
    a. Number of filters: 96
    b. Filter kernel size: [1,1]
    c. Stride sampling size: [1,1]
    d. Border Mode is valid
6. Perform ReLU as activation function
7. Perform Pooling2D layer with following parameters:
    a. Operation: Max
    b. Filter kernel size: [3,3]
    c. Stride sampling size: [2,2]
    d. Border Mode is valid

8. Perform convolution2D with following parameters:
    a. Number of filters: 64
    b. Filter kernel size: [3,3]
    c. Stride sampling size: [1,1]
    d. Border Mode is valid
9. Perform ReLU as activation function
10. Perform convolution2D with following parameters:
    a. Number of filters: 64
    b. Filter kernel size: [1,1]
    c. Stride sampling size: [1,1]
    d. Border Mode is valid
11. Perform ReLU as activation function
12. Perform convolution2D with following parameters:
    a. Number of filters: 64
    b. Filter kernel size: [1,1]
    c. Stride sampling size: [1,1]
    d. Border Mode is valid
13. Perform ReLU as activation function
14. Perform Pooling2D layer with following parameters:
    a. Operation: Average
    b. Filter kernel size: [6,6]
    c. Stride sampling size: [1,1]
    d. Border Mode is valid
15. Flatten the Tensor output
16. Add a dense layer with 10 nodes.
17. Use Softmax as activation
18. Use Euclidean as a Loss function
19. Using RMSprop as an optimizer with the following parameters:
    a. Learning rate: 0.1, Decay: 0.05
20. The code should train a model with batch size =32, number of epochs= 20

Task C: Text Classification (Time: 15mins)

1. The deep learning model design should have following layer sequence:
    a. Input →Embedding →LSTM → LSTM → DENSE→SIGMOID

    (For Individual layer details follow below steps.)
2. Define input text data (Classification data) with the following details:

    Number of classes: 4

    Data samples and corresponding labels are stored in txt file.

    Task: Sentence Classification

    Filename: ClassificationData.txt

    First Column: Text Sentence

    Second Column: Label
3. Add an Embedding layer
    a. Embedding size:100
4. Add a LSTM layer with following properties:
    a. Number of Nodes: 64
    b. Return sequences: True
5. Add a LSTM layer with following properties:
    a. Number of Nodes: 32
    b. Return sequences: False
6. Add a dense layer with 10 nodes.
7. Add a dense layer with 4 nodes.
8. Use Softmax as activation
9. Use Euclidean as a Loss function
10. Using RMSprop as an optimizer with the following parameters:
    a. Learning rate: 0.1
    b. Decay: 0.05