

Question No.1. Vision Dataset: CIFAR-10- It dataset consists of 60000 32x32 colour images in 10 classes.

Please find your dataset from the link- <https://www.tensorflow.org/datasets/catalog/cifar10>. **(5 marks)**

Prepare a python notebook (recommended- use Google Colab) to build, train and evaluate a deep neural network on the CIFAR-10 dataset. Read the instructions carefully.

Import Libraries/Dataset (0.25 mark)

- a. Import required libraries (recommended- use tensorflow/keras library).
- b. Import the dataset (use Google Drive if required).
- c. Check the GPU available (recommended- use free GPU provided by Google Colab).

2. Data Visualization (0.25 mark)

- a. Plot at least one sample from each class of the dataset (use matplotlib/seaborn/any other library).
- b. Print the shapes of train and test data.

3. Data Pre-processing (0.25 mark)

- a. Bring the train and test data in the required format.

4. Model Building (0.2*5 = 1 mark)

- a. Sequential Model layers- Use AT LEAST 3 dense layers with appropriate input for each. Choose the best number for hidden units and give reasons.
- b. Add L2 regularization to all the layers.
- c. Add one layer of dropout at the appropriate position and give reasons.
- d. Choose the appropriate activation function for all the layers.
- e. Print the model summary.

5. Model Compilation (0.25 mark)

- a. Compile the model with the appropriate loss function.
- b. Use an appropriate optimizer. Give reasons for the choice of learning rate and its value.
- c. Use accuracy as metric.

6. Model Training (0.5 + 0.5 = 1 mark)

- a. Train the model for an appropriate number of epochs (print the train and validation accuracy/loss for each epoch). Use the appropriate batch size.
- b. Plot the loss and accuracy history graphs. Print the total time taken for training.

7. Model Evaluation (0.25 + 0.75 = 1 mark)

- a. Print the final test/validation loss and accuracy.
- b. Print confusion matrix and classification report for the validation dataset. Write a summary for the best and worst performing class and the overall trend.

Hyperparameter Tuning- Build two more models by changing the following hyperparameters one at a time **(0.5 + 0.5 = 1 mark)**

Write the code for Model Building, Model Compilation, Model Training and Model Evaluation as given in the instructions above for each additional model.

1. *Dropout*: Change the position and value of dropout layer
2. *Batch Size*: Change the value of batch size in model training

Write a comparison between each model and give reasons for the difference in results. Also, make a comparison with the state-of-the-art accuracy for this dataset.

Question No.2. NLP Dataset: Sentiment Analysis dataset - 1.6 Million tweets. Please find your dataset <https://www.kaggle.com/kazanova/sentiment140>. The column 'text' has the tweet and 'target' gives the sentiment of the text. (5 marks)

Prepare a python notebook (recommended- use Google Colab) to build, train and evaluate a deep neural network on the given dataset. Read the instructions carefully.

1. Import Libraries/Dataset (0.25 mark)

- Import required libraries (recommended- use tensorflow/keras library).
- Import the dataset (use Google Drive if required).
- Check the GPU available (recommended- use free GPU provided by Google Colab).

2. Data Visualization (0.25 mark)

- Print at least two tweets from each class of the dataset, for a sanity check that labels match the text.
- Plot a bar graph of class distribution in dataset. Each bar depicts the number of tweets belonging to a particular sentiment. (recommended - matplotlib/seaborn libraries)
- Any other visualizations that seem appropriate for this problem are encouraged but not necessary, for the points.
- Print the shapes of train and test data.

3. Data Pre-processing (0.25 mark)

- Need for this Step** - Since the models we use cannot accept string inputs or cannot be of the string format. We have to come up with a way of handling this step. The discussion of different ways of handling this step is out of the scope of this assignment.
- Please use [this pre-trained embedding layer](#) from TensorFlow hub for this assignment. This link also has a code snippet on how to convert a sentence to a vector. Refer to that for further clarity on this subject.
- Bring the train and test data in the required format.

4. Model Building (0.2*5 = 1 mark)

- Sequential Model layers- Use AT LEAST 3 dense layers with appropriate input for each. Choose the best number for hidden units and give reasons.
- Add L2 regularization to all the layers.
- Add one layer of dropout at the appropriate position and give reasons.
- Choose the appropriate activation function for all the layers.
- Print the model summary.

5. Model Compilation (0.25 mark)

- Compile the model with the appropriate loss function.
- Use an appropriate optimizer. Give reasons for the choice of learning rate and its value.
- Use accuracy as metric.

6. Model Training (0.5 + 0.5 = 1 mark)

- Train the model for an appropriate number of epochs (print the train and validation accuracy/loss for each epoch). Use the appropriate batch size.
- Plot the loss and accuracy history graphs. Print the total time taken for training.

7. Model Evaluation (0.25 + 0.75 = 1 mark)

- Print the final test/validation loss and accuracy.
- Print confusion matrix and classification report for the validation dataset. Write a summary for the best and worst performing class and the overall trend.

Hyperparameter Tuning- Build two more models by changing the following hyperparameters one at a time (0.5 + 0.5 = 1 mark)

Write the code for Model Building, Model Compilation, Model Training and Model Evaluation as given in the instructions above for each additional model.

1. *Network Depth*: Change the number of hidden layers and hidden units for each layer
2. *Optimiser*: Use a different optimizer with the appropriate LR value

Write a comparison between each model and give reasons for the difference in results. Also, make a comparison with the state-of-the-art accuracy for this dataset.

Evaluation process-

1. Task Response and Task Completion- All the models should be logically sound and have decent accuracy (models with random guessing, frozen and incorrect accuracy, exploding gradients etc. will lead to deduction of marks. Please do a sanity check of your model and results before submission). There are a lot of subparts, so answer each completely and correctly, as no partial marks will be awarded for partially correct subparts.
2. Implementation- The model layers, parameters, hyperparameters, evaluation metrics etc. should be properly implemented.

Additional Tips (No marks)-

1. Code organization- Please organize your code with correct line spacing and indentation, and add comments to make your code more readable.
2. Try to give explanations or cite references wherever required.
3. Use other combinations of hyperparameters to improve model accuracy.