Prepare two python notebooks (recommended- use Google Colab) to build, train and evaluate a deep neural network on the Vision and NLP datasets (tensorflow or tensorflow.keras library recommended). Read the instructions carefully.

Question No.1. Vision Dataset: imagenette/160px-v2- Imagenette is a subset of 10 easily classified classes from the Imagenet dataset. Please find your dataset from the link - https://www.tensorflow.org/datasets/catalog/imagenette **(6 marks)**

1. **Import Libraries/Dataset (0 mark)**
    a. Import the required libraries and the dataset (use Google Drive if required).
    b. Check the GPU available (recommended- use free GPU provided by Google Colab).
2. **Data Visualization and augmentation (1 mark)**
    a. Plot at least two samples from each class of the dataset (use matplotlib/seaborn/any other library).
    b. Apply horizontal flip and width shift augmentation (horizontal_flip, width_shift_range) to the dataset separately. Print the augmented image and the original image for each class and each augmentation.
    c. Bring the train and test data in the required format.
    d. Print the shapes of train and test data.
3. **Model Building (0.2*5 = 1 mark)**
    a. Sequential Model layers- Use AT LEAST 3 hidden 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.
4. **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 a metric.
5. **Model Training (0.5 + 0.25 = 0.75 mark)**
    a. Train the model for an appropriate number of epochs. Print the train and validation accuracy and loss for each epoch. Use the appropriate batch size.
    b. Plot the loss and accuracy history graphs for both train and validation set. Print the total time taken for training.
6. **Model Evaluation (0.5 + 0.5 = 1 mark)**
    a. Print the final train and validation loss and accuracy. Print confusion matrix and classification report for the validation dataset. Analyse and report the best and worst performing class.

b. Print the two most incorrectly classified images for each class in the test dataset.

Hyperparameter Tuning- Build two more additional models by changing the following hyperparameters ONE at a time. Write the code for Model Building, Model Compilation, Model Training and Model Evaluation as given in the instructions above for each additional model. **(1 + 1 = 2 marks)**

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

Write a comparison between each model and give reasons for the difference in results.

Question No.2. NLP Dataset: Sentiment Analysis dataset - 1.6 Million tweets. Please find your dataset from the link - https://www.kaggle.com/kazanova/sentiment140

The column 'text' has the tweet and 'target' gives the sentiment of the text. **(6 marks)**

1. **Import Libraries/Dataset (0 mark)**
    a. Import the required libraries and the dataset (use Google Drive if required).
    b. Check the GPU available (recommended- use free GPU provided by Google Colab).
2. **Data Visualization (0.75 mark)**
    a. Print at least two movie reviews from each class of the dataset, for a sanity check that labels match the text.
    b. Plot a bar graph of class distribution in a dataset. Each bar depicts the number of tweets belonging to a particular sentiment. (recommended - matplotlib/seaborn libraries)
    c. Any other visualizations that seem appropriate for this problem are encouraged but not necessary, for the points.
    d. Print the shapes of train and test data.
3. **Data Pre-processing (0.25 mark)**
    a. **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.
    b. 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.
    c. 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 hidden 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 a metric.

6. **Model Training (0.5 + 0.25 = 0.75 mark)**

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

7. **Model Evaluation (0.5 + 0.5 = 1 mark)**

    a. Print the final train and validation loss and accuracy. Print confusion matrix and classification report for the validation dataset. Analyse and report the best and worst performing class.
    b. Print the two most incorrectly classified texts for each class in the test dataset.

Hyperparameter Tuning- Build two more models by changing the following hyperparameters one at a time. Write the code for Model Building, Model Compilation, Model Training and Model Evaluation as given in the instructions above for each additional model. **(1 + 1 = 2 marks)**

1. *Dropout:* Change the position and value of dropout layer
2. *Regularization:* Train a model without regularization

Write a comparison between each model and give reasons for the difference in results.

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).
2. There are a lot of subparts, so answer each completely and correctly, as no partial marks will be awarded for partially correct subparts.
3. Implementation- The model layers, parameters, hyperparameters, evaluation metrics etc. should be properly implemented.
4. Only fully connected or dense layers are allowed. CNNs are strictly not allowed.
5. Notebooks without output will not be considered for evaluation.

Additional Tips -

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.