Image Classification Task

The following are the steps that are needed to be strictly followed for you to successfully train and validate a model to be used in classifying an image whether it is an apple or an orange. The model architecture has now been provided for you. Your main task is to configure two parameters of this model namely - **epochs** and **optimizers.**

1. Go to your Colab Notebooks folder of your Google Drive and create a folder named **images.** After it is created, you must now have a folder structure like this:

   content/drive/My Drive/Colab Notebooks/images

2. Inside **images** folder, create two folders and name them - **train** and **validation**. After they are created, you must have the following folder structures:

   content/drive/My Drive/Colab Notebooks/images/train
   content/drive/My Drive/Colab Notebooks/images/validation

3. Inside on both **train** and **validation** folders, create two folders and name them **apples** and **oranges.** After they are created, you must have the following folder structures

   content/drive/My Drive/Colab Notebooks/images/train/apples
   content/drive/My Drive/Colab Notebooks/images/train/oranges

   content/drive/My Drive/Colab Notebooks/images/validation/apple
   content/drive/My Drive/Colab Notebooks/images/validation/oranges

4. Download the dataset, unzip it and upload the images to their respective folders. The following number of images must be on each folder

train/apples	459
train/oranges	376
validation/apples	50
validation/oranges	50

5. Download the **ApplesOrangesClassification** notebook and upload it to your Colab Notebooks folder. After it was uploaded, it must be on this location:

   content/drive/My Drive/Colab Notebooks/**ApplesOrangeClassifcation.ipynb**

6. Open the upload notebook and run the code cells from A to H sequentially. A model architecture for image classification has now been created for you. Running the code below E will allow you to compile it

7. Code cell below F is for training and validation of the model. It is your goal to optimise the model by minimising its training loss and improving accuracy. This can be achieved by tweaking some parameters. An examples of these are listed below:

   - **epochs**: 1 epoch means the whole dataset is passed both forward and backward through the neural network. Possible values are whole numbers, the higher the number, the more time required in training the model.

   - **optimizer**: finds the optimised value of the weights to minimize the cost function. Possible values:

     - rmsprop
     - sgd
     - adagrad
     - adadelta
     - adamax
     - nadam

     NOTE: Please be careful on the spelling of these optimisers whenever you want to use one of them.

8. After configuring the parameters of your model, plot the training loss and validation loss using cell G. You should use this plot to determine whether or not your model has **overfit** to the data set (see course notes on neural network classifiers).

9. Save the model by running the code cell H. Use the following filename convention: **model_xxxx.h5,** where xxxx is your student number. For example, if your student number is 1234, then the filename will be **model_1234.h5.** The saved model is located inside "images" folder unless you changed the path to where will it be saved.

10. Submit to Canvas your Python notebook, training and validation loss plot, and the saved model.

The accuracy of your model will be marked based on 30 unseen images of apples and oranges. These unseen images are not provided for the students and will be used by your teacher for marking purposes.