[22AML161] DEEP LEARNING

LAB. TEST – I

# PREPARATION GUIDELINES FOR EXPERIMENTS

1. **MODEL PERFORMANCE IMPROVEMENT WITH BATCH NORMALIZATION LAYERS**

* Loading and Preprocessing Data
  + Checking the shape of both training and test set
  + Splitting train set further into train and validation set using stratification
* Modeling
  + Adding Flatten and Dense layers and addiing them into Sequential layer
  + Adding Dense layer as output layer
  + Compiling model
  + Callbacks
    - Setting parameter "save\_best\_only" to save only the best model checkpoint
    - Setting parameters "patience" and "restore\_best\_weights" in early stopping callback to stop model training
    - Setting up these callback while calling model's fit method
* Modeling with Batch Normalization (BN) Layer
  + Same modeling but with the addition of a BN layer before each dense hidden layer
  + Addition of BN layer just before output layer
  + Compiling model
  + Setting up model checkpoint and early stopping callback
  + Setting up validation data and callbacks to call BN model.fit method

1. **TRANSFER LEARNING FOR IMPROVED MODEL WITH LESS DATA**

* Loading and Preparing Data
  + Checking shape of the dataset
  + Splitting datasets
* Modeling
  + Training Model to be Considered as Pretrained
    - Considering the length of this experiment, code-snippets for this section to build the pretrained model will be provided.
  + Training Target Model from Scratch
    - Building a Keras sequential model with layers such one Flatten, one or more Dense as hidden layers and one Dense output layer
    - Setting epochs while calling fit method of the model
  + Transfer Learning
    - Removing last layer from an existing model
    - Adding a top/outout layer with specific output(s) and activation function and optionally a name for the layer
    - Setting layer's "trainable" property to True or False
    - Compiling model with loss and optimizer

1. **BUILDING A CONVOLUTIONAL NEURAL NETWORK FROM SCRATCH**

* Loading & Analyzing Data
  + Checking shape of data sets (e.g. train and test set)
  + Checking data type of data sets
  + Consider the data type, scaling the data so that its values range between 0 and 1
* Preprocessing Data
  + Scaling dataset to have range of the transformed dataset is between 0 and 1
  + Splitting dataset further with stratification to seperate validation set from full train set
* Modeling
  + Initializing tf.keras.layers.Conv2D layer
  + Initializing tf.keras.layers.Dense layer

1. **BUILDING IMAGE CLASSIFIER USING PRETRAINED XCEPTION MODEL**

* Loading & Analyzing Data
  + Observe the information about the downloaded dataset

* Data Preparation
  + Resizing layers and its parameters
  + Lambda layer as a wrapper to application specific function (such as tf.keras.applications.xception.preprocess\_input)
  + Calling preprocessing layer by passing all the raw input (images)

* Modeling
  + Parameters to application/mode class tf.keras.applications.xception.Xception
  + Using GlobalAveragePooling2D() layer as function that accepts output of base model
  + Setting activation function to the Dense layer that accepts above pooling layer
  + Setting layer's "trainable" property to True or False

# INSTRUCTIONS FOR CODING

**Editor:**

1. Click Windows Start --> Anaconda3 (64-bit) --> Anaconda Prompt to open it. If both "Anaconda Prompt" and "Anaconda Prompt (Anaconda3)" prompts are available, open the later one.
2. Type "conda activate MLLab01" to activate lab. specific environment.
3. Then type "jupyter lab" to open Jupyter Lab as the recommended code editor for this lab. Test.
4. Open the template notebook file .ipynb by uploading it.

**NOTE:**

Just in case you prefer Microsoft Visual Studio Code (VS Code) over Jupyter Lab., then make sure to connect to right kernel by select "MLLab01" from the available once.

**Cleaning Up:**

1. After evaluation, make sure to delete the Jupyter notebook file from computer

2. Shutdown the computer before leaving.

# GENERAL INSTRUCTIONS

1. No printed question paper is required as the all the experiment titles are available at the end of this document.

1. Jupyter notebook templates along with datasets are available in the test folder.

1. Only section mentioned as "[CODE HERE]" are required to be coded.

1. DO NOT making any changes in any of the files the test folder.

1. Document all relevant information including important code blocks and observations in the green/blue book.