Final Exam Instructions

~~Date: *Nov. 29th, Wednesday*~~

Data Description: CIFAR-10 Dataset, as it suggests, has ten different categories of images in it. There are 60000 images of 10 different classes: Airplane, Automobile, Bird, Cat, Deer, Dog, Frog, Horse, Ship, and Truck. All the images are of size 32×32. There are, in total, 50000 train images and 10000 test images.

Your Task is to build an image classification model using the TensorFlow library using CNN and classify the data into ten categories, including pre-processing, splitting the data, and making the CNN Model.

Following are the instructions for building the model:

1) Data Preprocess

a) Import the libraries, please use TensorFlow 2 (default version on Google Colab is 2.14.0) to build your model.

b) Load the data using the code:
*(x_train, y_train), (x_test, y_test) = tf.keras.datasets.**cifar10**.load_data()*

b) Reduce the pixel value:
x_train, x_test = x_train / 255.0, x_test / 255.0

c) Flatten the label values
y_train, y_test = y_train.flatten(), y_test.flatten()

2) Sequential Model

a) Build a Sequential model using Keras.

b) Add an Input Layer, set the shape parameter of it equals the shape of the images.

c) Add a Conv2D Layer. Suggested parameters: filter = 32, kernel_size = [3,3], strides = [1, 1], activation function = ReLU, padding='same'.

d) Add a Batch Normalization Layer.

e) Add another Conv2D Layer. Suggested parameters: filter = 32, kernel_size = [3,3], strides = [1, 1], activation function = ReLU, padding='same'.

f) Add another Batch Normalization Layer.

g) Add a MaxPool2D Layer. Suggested parameters: pool_size = 2, strides = 2, padding='valid'.

h) Repeat step c) ~ step g) in total 1~3 times.

i) Add a Flatten Layer.

j) Add a Dropout Layer, with 0.2 as its parameter.

k) Add 1~2 Dense Layers. Suggested parameters: unit_number = 128, activation function = 'ReLU'

l) Add a Dense Layer as output layer, with unit = 10 and activation function = softmax.

Note: You can reduce those parameters, like filter or unit_number to make the model run faster. If you are using Google Colab, you can also add a GPU in the session management, to make it run faster.

3) Training

a) Compile the model with adam optimizer, *keras.losses.SparseCategoricalCrossentropy(from_logits=True)* as loss function, *metrics=[keras.metrics.SparseCategoricalAccuracy()].*
b) Fit the model on the train set with 32 batch_size and more than 20 epochs, *validation_split=0.2* (Optional: set an early stop)
c) Obtaining the training and evaluate history using:
train_history = model.fit(x_train, y_train)
evaluate_history = model.evaluate(x_test, y_test)

4) Evaluating

a) Plot the *SparseCategoricalAccuracy* in training history.
b) Plot the Loss in training history.
c) Print the test accuracy.
d) Calculate the the predicting value of test set using:
y_pred = np.argmax(model.predict(x_test), axis = 1)
e) Print the confusion matrix and classification report

5) Summary

a) Briefly describe your findings regarding model building, insights, and evaluation.
b) Where can you use this algorithm in the industry? - Give one practical application and describe it. (Include this in your report)