Assignment 2: Fundamentals (Part 1)

Start Assignment

Due Tuesday by 5:59pm **Points** 6 **Submitting** a text entry box or a file upload

Question 2 (1 point) ...

A dense layer consists of a weight matrix, W, and a bias vector, b. The number of rows in W is determined by the number of input features from the previous layer, while the number of columns in W and the length of b are determined by the number of units (neurons) in the dense layer. To produce output for a dense layer: the input is multiplied by the weight matrix; the bias vector is added to each row of the result; then an activation function is applied to the result.

On page 28 of Deep Learning with Python, a network to predict class labels for handwritten digits was defined as follows:

```
from keras import models
from keras import layers
network = models.Sequential()
network.add(layers.Dense(512, activation='relu', input_shape=(28 * 28,)))
network.add(layers.Dense(10, activation='softmax'))
```

This Multi-Layer Perceptron (MLP) model consists of two layers: a single hidden layer followed by the output layer.

The output layer uses the 512 features from the previous layer as input to predict probabilities for the 10 different classes; so the second Dense layer has (512 + 1) * 10 = 5130 parameters.

How many parameters are in the first Dense() layer, which uses 784 input features to produce 512 new features to be used as input to the next layer? Please note that this layer has both a weight matrix and a bias vector.

To check your answer, consider changing "keras" to "tensorflow.keras" in the above code example, then use network.summary() to print the parameter counts for the model.

```
from tensorflow.keras import models
from tensorflow.keras import layers
network = models.Sequential()
network.add(layers.Dense(512, activation='relu', input_shape=(28 * 28,)))
network.add(layers.Dense(10, activation='softmax'))
network.summary()
```

Model 2 (5 points) ...

This week's exercise involves another image classification problem involving 28x28 grayscale images. This time, however, we'll be classifying images into one of ten fashion categories (instead of 10 handwritten digits): **GitHub - zalandoresearch/fashion-mnist: A MNIST-like fashion product database. Benchmark**

0: t-shirt/top
1: trousers
2: pullover
3: dress
4: coat
5: sandal
6: shirt
7: sneaker
8: bag
9: ankle boot

There are three new wrinkles in our HyperBand search script:
1) we've added 3 new activation functions [all 3 resemble the rectified linear unit, but allow for negative values as well]:
a) leaky ReLU: the leaky rectified linear unit
b) ELU: the exponentual linear unit
c) SELU: the scaled exponential linear unit
2) we've added 2 new learning rate scheduling options [alternatives to ReduceLROnPlateau(); more on these next week]
a) ExponentialDecay()
b) CosineDecayRestarts()
3) we've added 2 new "black box" layers [which we'll talk about more in a couple of weeks]
a) Conv2D(): a layer where neurons only receive inputs from a subset of features from the previous layer [i.e. "local" features for the convolution layer, instead of "global" features for the dense layer]
b) MaxPooling2D(): a layer that changes the scale of the image

Please navigate to the following URL to accept the invitation for this Kaggle task:
https://www.kaggle.com/t/90efb52a00be40849b3ccde825e6a4c9

After accepting the invitation, you should connect to your VM and run the following commands:

```
conda activate py37_tensorflow
kaggle competitions download ml530-2021-sp-fashion
wget https://www.cross-entropy.net/ML530/fashion-search.py.txt
python fashion-search.py.txt
kaggle competitions submit ml530-2021-sp-fashion -f predictions.csv -m "fashion submission"
kaggle competitions leaderboard ml530-2021-sp-fashion -s
```

As before, you only need to upload your model's predictions to kaggle; but please include your kaggle display name in your submission for the homework question.