v2

EECS 4404E/5327
Assignment 3: Convolutional Neural Networks (10 pts)

Due date: **Sunday, December 1st by 18:00**

Submission: Submit a .zip package of your work including a single pdf file of your assignment with your solutions, each question at a new page, plus a folder containing your TensorFlow code, each question as a separate .py file, on Moodle's respective assignment tab. Make sure you write your name, student ID, and assignment# on each of the file.

Objectives:

The purpose of this assignment is to investigate the classification performance of convolutional neural networks. In this assignment, you will gain some experience in training a neural network and will use an effective way to avoid overfitting. All the implementations need to be done using Python and TensorFlow. For consistency, use **TensorFlow 1.15 version** (either the CPU or GPU version). More info can be found https://www.tensorflow.org/install/gpu. You are encouraged to look up TensorFlow APIs for useful utility functions, at: `https://www.tensorflow.org/versions/r1.15/api_docs/python/`. Also, look for a quick installation and guide at Moodle and under Practical Materials > TensorFlow Materials.

Note. You must write **vectorized** TensorFlow function using the provided API by TensorFlow, i.e., define operations, matrices, etc. in tf format so it uses the optimized backend for both CPU and GPU. For instance, `tf.matmul(a, b)` for multiplying tensors of matrices a and b.

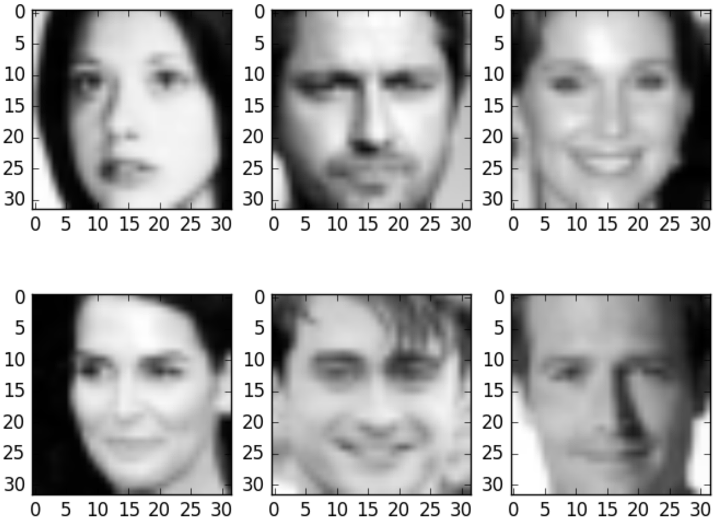
FaceScrub Dataset

This assignment will be done using the FaceScrub[1] dataset. We will be using a tiny version of this, with 6 celebrities and cropped images of 32-by-32. The target labels are the actor/actress name, encoded as integers, as well as the gender, encoded as '0' and '1'. You are provided with two .npy files which have 936 rows of images and labels, and you should divide the dataset into 80/10/10% for training, validation and test, respectively.

[1] http://vintage.winklerbros.net/facescrub.html

The name (ID) of the actors: ‘Lorraine Bracco’, ‘Gerard Butler’, ‘Peri Gilpin’, ‘Angie Harmon’, ‘Daniel Radcliffe’, and ‘Michael Vartan’ are encoded as ‘0’, ‘1’, ‘2’, ‘3’, ‘4’, and ‘5’, respectively.

The gender of the actors: ‘Male’ and ‘Female’ are encoded as ‘0’ and ‘1’, respectively.



You should use the following code to load the dataset.

```
def data_segmentation(data_path, target_path, task):
    # task = 0 >> select the name ID targets for face recognition task
    # task = 1 >> select the gender ID targets for gender recognition task
    data = np.load(data_path)/255
    data = np.reshape(data, [-1, 32*32])
target = np.load(target_path)

np.random.seed(45689)
rnd_idx = np.arange(np.shape(data)[0])
np.random.shuffle(rnd_idx)

trBatch = int(0.8*len(rnd_idx))
validBatch = int(0.1*len(rnd_idx))

trainData, validData, testData = data[rnd_idx[1:trBatch],:], \
                                 data[rnd_idx[trBatch+1:trBatch + validBatch],:],\
                                 data[rnd_idx[trBatch + validBatch+1:-1],:]

trainTarget, validTarget, testTarget = target[rnd_idx[1:trBatch], task], \
                            target[rnd_idx[trBatch+1:trBatch + validBatch], task],\
                            target[rnd_idx[trBatch + validBatch + 1:-1], task]

return trainData, validData, testData, trainTarget, validTarget, testTarget
```

1 Convolutional Neural Networks [10 pt.]

Implement a convolutional neural network with one convolutional layer, one max-pooling layer, and two layers of hidden units for classification of the FaceScrub dataset. Train the model over the dataset for all of the provided training data. You should use the same Xavier initialization of the weight matrices as before. A CNN is an efficient way to share weights in the model and reduce the amount of parameters in a deep pipeline that provides an intuitive interpretation of learning to recognize and compose image patches.

1. **Convolutional layer**: Write code that instantiates a 5-by-5 kernel of 32 filters for the image, and perform a 2d convolution of the image with stride 1 in each direction. Define a bias variable of shape [32] for the output of the kernel, and add the bias to the output of the 2d convolution. What will be the output tensor dimensions if we had used 64 filters of 5-by-5 and a stride of 1? What will it be if we used 32 filters of 7-by-7 and a stride of 2?

2. **Max pooling layer**: Write code that instantiates a max pooling layer of size 3-by-3 and a stride of 2 in each direction.

3. **Fully connected layer**: Now flatten the output of the previous layer and pass through two hidden layers with ReLU activations of size 384 and 192 with a dropout rate of 0.5. At the final layer, output the probabilities of predicting each celebrity ID.

4. **Learning**: Use cross entropy as the loss, as you've done before, and train the model. Plot the training and validation loss, as well as the training and validation accuracy over 50 epochs of training. Run on 3 different hyperparameter settings (i.e. change the learning rate, weight decay coefficient, dropout) and report your results.

5. **Visualization**: To get more insight into what a convolutional neural network achieves with its architecture, you will visualize the function that the convolutional layer provides. Visu-alize 8 of the 5-by-5 kernels trained in question 1, comment on what the network is trained to recognize with these kernels and how further layers of convolution may improve the per-formance.