Image Recognition with Neural Nets

In this assignment you are going to use neural networks to perform some basic image recognition tasks. The image dataset we have provided you contains images (in jpeg format) of dimensions 120x128, with the training set consisting of 315 images and the test set consisting of 90 images.

In each image, the subject has the following characteristics:

Name – name of the subject

Direction Faced – left, right, straight, up

Emotion – happy, sad, neutral, angry

Eyewear – open, sunglasses

Each image follows the naming convention "*name_directionFaced_emotion_eyewear.jpg"*

Even though the original dataset contains images of size 120x128, you will apply a basic image compression technique to compress the 120x128 pixel images into images of size 30x32, and for this whole assignment you will be working with the compressed version of the original images.

(a) **Image Compression**

For this section you will write a function which takes as input a matrix **A** of dimensions *m* x *n* and returns a matrix **B** of dimensions *m*/2 x *n*/2 where **B** is calculated as follows:

$\mathbf{B}_{11} = (\mathbf{A}_{11} + \mathbf{A}_{12} + \mathbf{A}_{21} + \mathbf{A}_{22})/4$

$\mathbf{B}_{12} = (\mathbf{A}_{13} + \mathbf{A}_{14} + \mathbf{A}_{23} + \mathbf{A}_{24})/4$

….

$\mathbf{B}_{21} = (\mathbf{A}_{31} + \mathbf{A}_{32} + \mathbf{A}_{41} + \mathbf{A}_{42})/4$

$\mathbf{B}_{22} = (\mathbf{A}_{33} + \mathbf{A}_{34} + \mathbf{A}_{43} + \mathbf{A}_{44})/4$

….

Where $\mathbf{X}_{jk}$ is the value of the matrix **X** at the j^{th} row and k^{th} column for any given matrix **X**.

You will write your code for finding the matrix **B** within the function *compress_image*. – **20 points**

(b) **Training and Model Evaluation**

You will now train a neural network for predicting the direction each subject is facing. Your neural network will have 960 units in the input layer, 30 units in the hidden layer and 4 units in the output layer, where each unit in the output layer denotes the direction the subject is facing. For this section you will have to compress each image to the dimensions 30x32 **using**

the function you defined in part (a) by applying this function twice on each image, and then flatten this matrix to a vector of dimensions 1x960. This means the input to your neural network will be a matrix of size *m*x960 where *m* is the number of instances in your training set. Additional hyperparameters you will use to train your network are as follows: l2=0.1, l1=0.0, epochs=1000, eta=0.001, alpha=0.001, decrease_const=0.00001, minibatches=50, shuffle=True, and random_state=1.

The function *model_evaluation* (in the script *NN_model.py*) takes as input a **test** image of dimensions 120x128 and using the network you trained above, it should return a string specifying the direction that the test subject is facing ("up", "straight", "left", "right"). This means you will also have to compress this test image to dimensions 30x32 before feeding it to the network. – **20 points**

(c) **Hyperparameter Optimization (Predicting direction faced)**

For this section you will optimize the hyperparameters in your neural net and find the ones that give you the highest accuracy in the task of predicting the direction that the subject is facing. The only constraints here are that your network should have 960 units in the input layer and only one hidden layer. In the script *custom_model1.py*, the function *model_evaluation* takes as input a **test** image of dimensions 120x128 and using the optimized network, it should return a string specifying the direction that the test subject is facing ("up", "straight", "left", "right"). – **30 points**

(d) **Hyperparameter Optimization (Predicting emotion felt)**

For this section you will optimize the hyperparameters in your neural net and find the ones that give you the highest accuracy in the task of predicting the emotion that the subject is feeling. The only constraints here are that your network should have 960 units in the input layer and only one hidden layer. In the script *custom_model2.py* the function *model_evaluation* takes as input a **test** image of dimensions 120x128 and using the optimized network, it should return a string specifying the emotion that the test subject is feeling ("happy", "sad", "neutral", "angry"). – **30 points**

Note:

Do **not** train your model in the *model_evaluation* functions. This means that after training your network you should save your weights to a file. The *model_evaluation* functions should just initialize your neural network, load the weights from the file and using those weights it should make its prediction. If you train your model in the *model_evaluation* functions, you will be penalized.