Homework 6 – Deep Neural Networks (CS/DS 541, Whitehill, Spring 2021)

You may complete this homework either individually or in teams up to 3 people.

1 Facial Age Estimation using VGG16 [35 points]

In this project, you will train a VGG16 network to estimate how old each person looks in a set of face images (see `facesAndAges.zip` on Canvas, which contains 7500 grayscale images of size 48x48). You can use either TensorFlow (`https://www.tensorflow.org/api_docs/python/tf/keras/applications/VGG16`) or PyTorch (`https://pytorch.org/vision/stable/models.html`). You can either train a VGG16 from scratch (which gives you the advantage that the input size can be tailored exactly to the dimensions of your dataset), *or* you can use a pre-trained VGG16 (with fixed input size of 224x224; you'll need to transform your dataset accordingly) with the strong advantage of harnessing the information stored in ImageNet. In either case, you will need to account for the fact that the images in your dataset (see below) are grayscale (1 input channel), whereas VGG16 expects RGB (3 channels); the easiest strategy is to replicate the grayscale values 3x. You will also need to modify the network so that the last layer outputs a real number, not a 1000-dim vector.

In particular:

1. Randomly partition the data into 80% training+validation, and 20% testing.
2. Either initialize the weights randomly, or initialize them according to their pre-trained values using ImageNet.
3. Either train the network from scratch, or perform supervised pre-training (train last few layers and possibly fine-tune the remaining layers).
4. After optimizing hyperparameters on the validation set, report your root mean squared error (RMSE) loss on the test set. Report your training, validation, and testing loss (RMSE) in the PDF file you submit. For full credit, you should get < 13 years RMSE on the 20% test partition.

In addition to your Python code (`homework6_WPIUSERNAME1.py` or `homework6_WPIUSERNAME1_WPIUSERNAME2.py` for teams), create a PDF file (`homework6_WPIUSERNAME1.pdf` or `homework6_WPIUSERNAME1_WPIUSERNAME2.pdf` for teams) containing the screenshots described above. **Please submit both the PDF and Python files in a single Zip file.**

2 Competition for the Fabulous Prize [Optional]

In this part, you can use state-of-the-art neural networks (e.g., ResNet, ResNeXt, BoTNet, etc.) to tackle the facial age estimation challenge. In contrast to part 1 above, the model you choose and the training approach you take in the competition is up to you.

Rules:

- You can use the face images and associated ages in `facesAndAges.zip` to train a neural network and to optimize its hyperparameters. How you partition the data is up to you.
- You are permitted to use any other *publicly available* dataset of face images and labels as long as you disclose where they came from.
- You are encouraged to explore different neural network architectures, hyperparameter settings, data augmentation, pre-training, etc.
- You are strongly suggested to use an off-the-shelf neural networks training framework such as TensorFlow or PyTorch.

- Your network will be evaluated on a test set that will be released on Thursday, April 1 at 8am. These test images are randomly sampled from the same population of face images and labels as were the training data.
- Prior to receiving the test data, you will be required to *freeze* and submit your trained neural network and all associated weights that you will use to estimate the ages of the faces in the test set.
- You are *not* permitted to manually label any of the face images in the test set.
- After receiving the test data, you will have 5 hours to submit your network's estimates for the age of each test face. *In addition*, you must also submit your *estimate* for your guesses' mean-squared error relative to the ages according to ground-truth labels.
- The winner will be announced during class on Wednesday, March 4. The team with the lowest actual RMSE wins the Fabulous Prize.