CS559 - Spring 2020 Homework:
Estimation of Facial Attractiveness Level using TensorFlow

This homework aims that you will get familiar with TensorFlow and gain some hands-on experience about several deep learning concepts covered in class. You are allowed to work as a group of two students or --if required-- individually. If you cannot form a group and you will work individually, then you need to inform me via email (dibeklioglu@cs.bilkent.edu.tr) as soon as possible (subject line should be cs559_2020s_hw_group).

Due date: April 13th, 2020, 23:59 (Turkish time).
Please send an email (subject line should be cs559_2020s_hw) that indicates the weblink (e.g. wetransfer) of your submission, to dibeklioglu@cs.bilkent.edu.tr

Your submission should be a single zip archive named "YourBilkentID.zip" that includes a report (must be a pdf file), README file and source code files. If you work as a pair, only one file should be submitted for the pair. Each submission should include a README file that provides the name(s) of the authors and a brief summary of contents of other files in the zip archive. Late submissions will not be accepted.

Step 1: Preparation

- Install TensorFlow following the instructions at https://www.tensorflow.org/install. You do not need to install with GPU support.
- Use a Python based installation.
- (Optional) You can use ipython and jupyter notebook for an interactive python shell and web-based development environment. If you would prefer, you are allowed to submit your source code in the form of jupyter notebook files.
- Download the downsampled subset of SCUT-FBP5500 dataset from: http://www.cs.bilkent.edu.tr/~dibeklioglu/teaching/cs559/docs/SCUT_FBP5500_downsampled.zip
  - Note that the dataset is downloadable only within Bilkent network. Use VPN to access from home.
  - You are provided the aligned and cropped color face images. You can directly use them. Label of attractiveness level for each image is given in the beginning of the filename as follows: < attractiveness_level >_<acquisition_id>.jpg
    Train and test your models using the labels of attractiveness level.

Step 2: Learn TensorFlow basics

- Through the following tutorials, study TensorFlow basics and get used to them.
  - https://adventuresinmachinelearning.com/python-tensorflow-tutorial/
  - https://www.tensorflow.org/tutorials

- Make sure you understand the computation graph, session, placeholder, variable and other fundamental TF concepts very well.
- Get familiar with TensorFlow documentation. To complete the assignment, you may need to learn additional details of TensorFlow yourself using TensorFlow API documentation and/or its source code.

Step 3: Build your facial attractiveness estimation network

Your task is to design and train a deep network for facial attractiveness estimation that will accurately predict the level of attractiveness of people from their (80×80 pixel) facial images. Note that you need to approach this problem as a regression task, not classification.

Your network should consist of a series of convolutional layers with ReLU activations, pooling and fc layers, and a regression layer. You are required to explore the following details and techniques to improve your neural network:

- design a deep architecture and tune the number and configuration of layers (importance: 30%)
- tune your loss function (importance: 10%)

- investigate the effect of Xavier vs Gaussian (random) initialization (importance: 10%)
- try using batch normalization layers (importance: 10%)
- try using l2-regularization and dropout-regularization, tune the l2-regularization weight and dropout keep probability hyper-parameter (importance: 15%)
- use Adam optimizer and tune its hyperparamaters (number of iterations, batch size, learning rate, etc). Note that early-stopping based on validation set error can sometimes be used as a regularization technique. (importance: 25%)

Investigate these techniques in the order that you prefer / need, such that you progressively build a better model by using combinations of these techniques. Note that you may need to re-visit some of these techniques after you update your architecture, as the behaviour of deep learning techniques and their hyper-parameters depends on overall network architecture.

Train your models on SCUT-FBP5500 training set, and evaluate them on the validation set in terms of Mean Absolute Error (MAE):

MAE = Mean_for_all_Samples(Absolute_Value(Attractiveness_Label – Predicted_ Attractiveness))

Since the provided attractiveness labels are integers, you should also round your predictions (obtained from regressor) for MAE computation.

In your report, explain all the techniques/architectures/hyper-parameters/etc that you try (together with your motivations), provide and discuss the evaluation results that you obtain for them. The source file(s) you submit should also provide the main code for all these experiments that you perform (no need to submit separate source files just for hyper-parameter variations).

Step 4: Evaluate your network on the test set

Once you are done with the design and development of your network architecture, report your MAE on the test set for the model with the smallest validation set error.

Additionally, provide a couple of informative success and failure examples (ie. image and the corresponding –estimated-- attractiveness level) in your report. Discuss your results.

Grading

The goal, clearly, is not to randomly try a very large number of different architectures and hyper-parameters. Instead, you are expected to run a series of meaningful, purposeful experiments towards building a successful age estimation model, analyze and understand the results in a detailed way, and thoroughly discuss what works well/poorly (and why). Provide plots (eg, MAE over training iterations), tables and qualitative examples to support your discussion, if needed.

The final accuracy on the test set will be taken into account in the following way: if your test set error is much higher than what most people achieves, you may loose points up to 15 points. If you obtain exceptionally good results (without cheating) on the test set, you may get extra points up to 15 points. The total grade of the homework may not exceed 100 points.

The quality of the report is important. You may loose points up to 40 points, due to poor presentation and/or insufficient discussion of the results in the report. The reports have to be 5 pages at most. The number of pages is not a factor in grading. Reports must be prepared using the IEEE double-column transactions article template.

Rules and FAQ

Homework can be done **in pairs or –if required-- individually**. Larger groups are not allowed.

The source code that you submit should cover all results and experiments that you provide in your report in an accessible manner. That is, do not throw away the code for the models/approaches that you try, report but do not use at the end (exploration of various deep learning architectures and techniques is part of the homework). Also the README file should be informative enough so that I can locate the code sections corresponding to your results.

You are **not allowed to use keras**. You are allowed to use TensorFlow standard implementations of dropout and batch-normalization.

Can I re-use existing code? You may re-use the codes (with citation) from the tutorial pages whose links are explicitly provided in the homework specification. However, in most cases, you cannot reuse existing code from other sources, even if it is your own code, unless it is a small utility code that is not directly related to the homework contents (eg, zip file I/O). If you decide to use/adapt an existing code, beware of the following: (i) you should cite and explain anything that you are reusing/adapting clearly in your source code and your report, (ii) you may get partial/zero grade for those parts (even if you cite it properly), if the re-used code covers a part of the homework that needs to be implemented by you, (iii) you should not violate the license of the code that you are reusing. If you are in doubt, please contact me.

Do I need a computer with a GPU to complete the assignment? You do not need to use a GPU to complete the homework. Experiments can be completed in a bearable amount of time on a modern CPU. However, if you have access to a high-end GPU or online computing resources, you definitely can use them (please mention this in your report), which may significantly speed up your experiments. Also note that, as long as you use the standard operators / layers in tensorflow, your code is ready to run on a GPU, in most cases.

Will you teach TensorFlow in class? There will not be a separate in-class tutorial on TensorFlow due to lack of time. The homework itself aims to be the tutorial. If you encounter an issue, you may look for a solution through online resources like tensorflow documentation, tensorflow discussion groups, and stackoverflow. You may also discuss your problems with your classmates.