Middle East Technical University | Department of Computer Engineering

CENG 483

Introduction to Computer Vision

Spring 2017-2018

Take Home Exam 2

Age Estimation from Facial Image

Due date: **14.05.2018 - 23:55**

1 Objectives

The purpose of this assignment is to familiarize yourselves with the fundamental deep learning solutions to computer vision problems and PyTorch framework on age estimation (AE) problem. The assignment aims to give insights about the deep learning based computer vision research and their evaluation methods.

Keywords: *Age estimation, Deep Learning, PyTorch*

2 Specifications

In this assignment you are required to implement an age estimation system based on deep learning methods, and to evaluate it with the provided dataset. This evaluation should be reported in a **3-4 pages** long paper prepared in IEEE Conference Proceedings Template (LaTeX is recommended) provided in the following link.

`https://www.ieee.org/conferences_events/conferences/publishing/templates.html`

The text continues with a detailed explanations of the methods and requirements.

2.1 Age Estimation

The main purpose of the AE systems is to determine age of the person in a query image. The estimation is done by evaluating semantic contents of the query image. However, there is a difficulty in revealing the semantics of images due to the semantic gap. In order to overcome this difficulty, images are described as feature vectors which are higher level representations than collection of numbers. With these feature vectors, age estimation can be formulated as a learning problem to match an image representation with the age of person in the image.

In this assignment you are required to construct a fully-connected network with rectified linear unit (ReLU) as nonlinearity function between layers and train it with RMSprop optimizer using the provided

feature vectors which have been extracted using Resnet18. This is basically learning a function that matches deep image features to age of the person in images. While training the network you are required to use mean squared error loss function (1) to minimize the difference between actual age and the estimated one.

$$L = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{1}$$

2.2 Programming and Interpretation Tasks

You are required to implement the aforementioned AE system using fully connected neural networks with four different number of hidden layers (e.g., l=0,1,2,3). Your system should output a file named *"estimations_validation.npy"* for validation set or *"estimations_test.npy"* for test set that contains a serialized NumPy array holding age estimations for each image (see numpy.save and PyTorch tutorial for the NumPy bridge). After implementation, you should evaluate your solution with different configurations as mentioned before using the provided validation set. For evaluation you should use 0-1 loss (2) with threshold **10**. Finally, you will decide on the most successful configuration based on your experiments. One day before the deadline, you will be provided with test set. Test images are named as **"i.jpg"**. The expected output is of the following form: estimations_test[i]=estimation_for_i.jpg. For detailed explanation see Section 2.3.

$$L = \frac{1}{n}\sum_{i=1}^{n} l_i, \qquad l_i = \begin{cases} 1 & |\hat{y}_i - y_i| > \theta \\ 0 & |\hat{y}_i - y_i| \leqslant \theta \end{cases}. \tag{2}$$

An important **hint** about the implementation is saving model at intermediate epochs. While training the network, dataset is usually divided into mini-batches. After computing the loss for each batch, parameters of the network are updated. One pass of whole training set is called an epoch. In order to get a good fit to data, the number of epochs that the network will be trained should be determined. This can be done with the help of loss history plots that shows the loss computed using training and validation sets for each epoch. After examining the plot, one can decide on the number of epochs. In order not to retrain the network, you can save model and optimizer parameters at some epochs (i.e. at each 5 epochs).

Another important **hint** is setting a seed for random number generators. This allows the experiments to be repeatable since whenever it is run, it starts from the same random parameters.

Along with the implementation of an AE system, you are required to prepare a report that explains your work, rationale behind your choices and results of the experiments. It should include at least the following items;

- Discussion on the effects of the number of layers.
- Rationale behind your choices of hyper-parameters like number of layers, number of epochs, layer sizes etc.
- Discussion on best, moderate and worst estimates for validation set.
- Analysis with your self portrait. For this part you are required to test your network with a normal self portrait and one with a hat or sunglasses, etc. In order to extract features from your images, a script is provided in homework files.

2.3 Dataset

The dataset consists of 5000 training, 2000 validation and 2000 test images of different sizes. In order not to bother you with feature extraction process, the deep features extracted with Resnet18 pretrained network is provided. The training images are named from "t0.jpg" to "t4999.jpg" and validation images are named from "v0.jpg" to "v1999.jpg". As explained before, test images are named from "0.jpg" to

"1999.jpg". For features and ground truth information, you will be provided with files named train.npy, valid.npy, test.npy, train_gt.npy, valid_gt.npy. Each file contains a NumPy array holding the data indexed with image identification number given in the image name. For example, features for the image "t24.jpg" can be accessed with train[24].

3 Restrictions and Tips

- Your implementation should be in Python 3.
- You are required to use PyTorch library (v0.4.0) to implement neural networks and train them.
- Do not use any available Python repository files without referring to them in your report.
- Don't forget that the code you are going to submit will also be subject to manual inspection.

4 Submission

- **Late Submission:** As in the syllabus.
- Implement the task in a directory named **the2**. The implementation together with a 3-to-4 pages long report focusing on theoretical and practical aspects you observed regarding this task and the ***"estimations_test.npy"*** file for the test set should be uploaded on ODTÜClass before the specified deadline as a compressed archive file whose name is <student_id>_the2.tar.gz, e.g., 1234567_the2.tar.gz. While creating the archive, do not compress with another format and change it with renaming to "*.tar.gz". Since it is not a tar-ball, it cannot be decompressed with "tar xvf *.tar.gz" command.
- The archive must contain **no directories** on top of implementation directory, report and the results.
- Do not include the database and unmentioned files in the archive.
- If needed, you can provide a README.txt file.

5 Regulations

1. **Cheating: We have zero tolerance policy for cheating**. People involved in cheating will be punished according to the university regulations.
2. **Newsgroup:** You must follow the course web page and ODTÜClass (odtuclass.metu.edu.tr) for discussions and possible updates on a daily basis.