ECBM E4040 Neural Networks and Deep Learning, 2016 Fall
HOMEWORK #1

INSTRUCTIONS: This homework contains two parts - Programming (I) and Theory (II). Please submit your homework via your bitbucket repository. Your submission should consist of

1. a file called hw1_writeup.pdf providing the solution to the theory questions;
2. completed code in hw1a.py and hw1b.py ;
3. an output folder containing png files created by executing hw1a.py and hw1b.py.

PART 1 : PROGRAMMING

For this homework, you will be using the FEI(frontal face) dataset (already included in the bitbucket repository).

Clone the repository created for you via bitbucket. The repository already contains the dataset.

A public AMI **'ECBM4040_HW1_AMI'** has been created with all necessary packages and patches pre-installed for solving this assignment. If you would like to use your own Ubuntu machine/AMI, the following instructions might be helpful

1. Install `libjpeg` by executing `sudo apt-get install libjpeg-dev`
2. Install `libx11` by executing `sudo apt-get install libx11-dev`
3. Install `PIL` in your theano environment by executing `pip install Pillow`
4. Apply changes from this PR to your theano installation.

An alternative to the last step above is downgrading numpy by running the following commands in your conda environment

```
1. conda uninstall numpy
2. conda uninstall scipy
3. conda uninstall matplotlib
4. conda install numpy==1.9.3
5. conda install scipy==0.15.1
6. conda install matplotlib==1.4.2
7. pip uninstall theano
8. pip install theano
```

If you are using a different OS, please search online for how to perform the above steps for your configuration.
Use single precision for both problems in this assignment.

ECBM E4040 Neural Networks and Deep Learning, 2016 Fall
HOMEWORK #1

PROBLEM a:
In this problem, you will be dividing the images into blocks of sizes (8, 8), (32; 32), (64, 64) and performing principal component analysis in each case. For each case you will visualize reconstructions using different number of principal components and also visualize the top components.

Skeleton code for this problem has been provided in hw1a.py in the repository.

Some useful links:-
- PIL documentation
- PIL convert documentation
- theano.tensor.nnet.neighbours documentation
- numpy.linalg.eigh documentation

PROBLEM b:
In this problem, you will be essentially performing the same tasks as in PROBLEM a, but on the whole image instead of blocks. Since the images are of size 256x256, the matrix $X^T X$ will be of size 65,536 x 65,536. You most likely won't be able to even load this matrix (unless you have enormous amount of RAM available), let alone perform eigenanalysis on it. Hence, you will be solving this problem by using gradient descent.

Recall from class, that the top principal component can be extracted by solving the following optimization problem

$$\underset{\mathbf{d}}{\text{argmin}} \quad -d^T X^T X_d$$
$$\text{subject to} \quad d^T d = 1$$

The above can be resolved by using gradient descent with the cost function

$$f(d) = -d^T X^T X_d$$

while normalizing **d** after each update (descent).

Other principal components can be found by ***"taking out the contribution of the already determined components"***. This can be done as in the following pseudocode

ECBM E4040 Neural Networks and Deep Learning, 2016 Fall
HOMEWORK #1

```
$for\ i = 0, ..., N\ do:$
    $A_i \leftarrow X^T X - \sum_{j=0}^{i-1} \lambda_j d_j d_j^T$
    $initialize\ d_i\ randomly\ and\ let\ t = 1$
    $while\ (t \le T\ \&\ Stopping\ condition\ is\ not\ True)\ do:$
        $y \leftarrow d_i - \eta \nabla_{di}(-d_i^T A_i d_i)$
        $d_i \leftarrow \frac{y}{\|y\|}$
        $t \leftarrow t + 1$
    $\lambda_i \leftarrow d_i^T X^T X d_i$
```

Gradient descent can be performed very easily in theano as it supports symbolic differentiation. Please note that there shouldn't be a need to compute large matrices of order 65,536 x 65,536 at any point. You should write your theano expressions and functions such that it avoids computing the large matrices. Choose an appropriate learning rate and an appropriate stopping criteria (for example, when the change in the cost is below some small or the change in kdik is below some small $\in$ or the change in $\|d_i\|$ is below some small $\in$)

PROBLEM c:
Adapt the above pseudo code by using Stochastic Gradient Descent (SGD) /Minibatch SGD instead of using the vanilla Gradient Descent method mentioned above. The algo/pseudo code may be recorded in a Markdown cell of the Jupyter Notebook.(You do not need to implement the SGD version however)

Some useful links:-
1. Theano Logistic Regression Example
2. Math Equations in Jupyter Notebook(useful for writing pseudo code)

PART 2 : THEORY

PROBLEM d:
(i) Let us consider a vectorized image $I \varepsilon [0, 255]^{M \cdot N}$ and assume that the distribution of values in the image is given by

$$p(x) = \frac{\cdot f(x)}{\int_0^{255} f(x)dx} \approx f(x),\ where\ f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},\ \sigma^2 = 1,\ \mu = 127.5.$$

In classical histogram equalization, the task is that of mapping the values in the original image so that

ECBM E4040 Neural Networks and Deep Learning, 2016 Fall
HOMEWORK #1

$$p(y) = \begin{cases} 1/255, & x \in [0, 255] \\ 0, & \text{otherwise.} \end{cases}$$

Find this mapping $g(x) = y$. Hints: The functions at hand have special names; use Google to find out!

(ii)

$$p(\mathrm{x}=x, \mathrm{y}=y, \mathrm{z}=z) = \begin{cases} 8xyz, & \text{for x,y,z} \in [0,1], \\ 0, & \text{otherwise.} \end{cases}$$

Find p(x=x), p(y=y), p(z=z), E(xyz).
Are x and y conditionally independent given z? Justify.

PROBLEM e:
Consider data $X_i \sim \{x^{(1)}, x^{(2)}, \ldots\ldots, x^{(m)}\}$, $x^{(i)} \in R^n$ to have been drawn independently from a multivariate gaussian distribution $N(\mu, \Sigma)$

$$N(\mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}}} \frac{1}{|\Sigma|^{\frac{1}{2}}} exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$

where μ is an n-dimensional vector and Σ is an n x n covariance matrix.

Also, consider a gaussian prior distribution on the mean vector μ of the form

$$p(\mu | u_0, \Sigma_0) = \frac{1}{(2\pi)^{\frac{n}{2}}} \frac{1}{|\Sigma_0|^{\frac{1}{2}}} exp\left(-\frac{1}{2}(\mu-\mu_0)^T \Sigma_0^{-1} (\mu-\mu_0)\right)$$

1. Derive Maximum A Posteriori estimates for μ and Σ.
2. Are the estimators biased.
3. Compare the Maximum Likelihood Estimate and Maximum A Posteriori Estimate of Σ. What do you observe?