

---

## Computer Project 1

---

**Goal:** To implement two classification methods using engineered features and raw data. Further you will test these methods accuracy.

**Instructions:**

- For this project, you can use any programming language that you want. However, it is recommended to use Python for two reasons: it is considered to be in the first place in the list of all AI development languages; to get familiarized with it since it will be the programming language for the second project and some assignments (see the example on how to use Google Colab).
- The point of this project is to get a deep understanding of these algorithms by coding them "from scratch". For this reason, build-in functions that do back propagation, training etc. are not allowed. The only build-in functions allowed are the ones to extract features from the images. Therefore, only the following package definitions are allowed:

```
import numpy as np
import matplotlib.pyplot as plt
from keras.datasets import mnist
from skimage import measure # To collect features from images
```

- Submit a report in PDF format limited to no more than 10 pages (not including the computer code), and the code. Your report will be graded based on the following: correctness, clarity and conciseness of the work.

**Dataset:**

We are going to use the MNIST handwritten digit database. It can be downloaded using the mnist function from keras.datasets as

```
(x_train, y_train), (x_test, y_test) = mnist.load_data()
```

**PART A - Fisher Discriminant** (special case of generalized likelihood ratio test)– which is equivalent to generalized Bayesian decision rule given for the case when the classes obey a Gaussian PDF and have the same covariance matrix.

Tasks (use Feature\_extraction\_example.ipynb as a starting point):

1. Collect feature from the following 2 classes: handwritten 0's and handwritten 1's.
2. Write a computer program to implement a Fisher discriminant for these two classes using only the training data. Report the number of images of each class and the balanced classification accuracy of the training data. Does the Fisher discriminant separate perfectly the training data with the selected features? Comment the results.
3. Use your discriminant function from 2) to classify the images in the test set. Report results, the method's accuracy and examine when the method fails. Comment on your findings.
4. Repeat 2) and 3) but this time separating the following 2 classes: handwritten 5's from 6's. Are the same features that you used to separate 0's from 1's good for this new task? Report the balanced accuracies and comment the results.

**PART B - Logistic Regression with a Neural Network mindset**

In this part, we are going to build a logistic regression from a neural network point of view to separate 2 classes from the MNIST dataset. The neural network will have  $28 \times 28 = 784$  input

neurons, no hidden layers and 1 output with a sigmoid as activation function. The output ( $\hat{y}$ ) will take values from 0 to 1 and can be thresholded at 0.5 to predict the classes.

For an image  $x^{(i)}$ :

$$z^{(i)} = w^T x^{(i)} + b$$

$$\hat{y}^{(i)} = \text{sigmoid}(z^{(i)})$$

$$L(\hat{y}^{(i)}, y^{(i)}) = -y^{(i)} \log(\hat{y}^{(i)}) - (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})$$

Then, the cost function is computed as:

$$J = \frac{1}{m} \sum_{i=1}^m L(\hat{y}^{(i)}, y^{(i)})$$

where  $m$  is the number of images in the training set.

Tasks (use Project1\_NN\_empty.ipynb as starting point):

1. Program and test the following functions:

- sigmoid: computes and returns the sigmoid of a number.
  - Input:  $x$
  - Output: sigmoid of  $x$
- propagate: implement the cost function and its gradient.
  - Inputs:  $w$ ,  $b$ ,  $X$  (matrix where each column represents an image),  $y$  (vector of labels)
  - Outputs: cost,  $dw$ ,  $db$
- gradient\_descent: optimizes  $w$  and  $b$  by running a gradient descent (minimizes the cost function  $J$ ). Use the previously implemented propagate function. Print the cost after every 100 training samples.
  - Inputs:  $w$ ,  $b$ ,  $X$ ,  $y$ , num\_iterations, learning\_rate
  - Outputs:  $w$ ,  $b$ , costs
- predict: predicts the label by using the learned parameters and thresholding the predictions at 0.5.
  - Inputs:  $w$ ,  $b$ ,  $X$
  - Output:  $y_{\text{prediction}}$

2. Use the previous functions to build a model that classifies handwritten 0's from 1's using the MNIST database:

Steps:

- Load the MNIST dataset, extract the 2 classes and reshape the images to vectors.
- Standarize by dividing the image vectors by 255.
- Initialize to zero the parameters of the model (the weights  $w$  and the bias  $b$ )
- Training: Choose the number of iterations and the learning rate. Learn the parameters on the training set by using the gradient\_descent function that was previously implemented.
- Testing: Use the learned parameters to predict the labels for the test set  $x_{\text{test}}$ . Compute the accuracies of the logistic regression model in the test set.

3. Re-train the Neural Network but this time to separate handwritten 5's from 6's and compute the balanced accuracy for the test set.
4. Compare and comment on the advantages and disadvantages of using the Fisher discriminant versus the neural network logistic regression.