**ML PROJECT 4 - REPORT**

**Group Members:** Yildirim Kocoglu, Vaishnavee Sharma, Ishtiaque Zaman, Denzel Smith

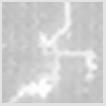
Requirement: Write a **brief** report.

**Introduction**

Softmax regression (multinomial logistic regression) with linear mapping was used both for Part 1 (MNIST) and Part 2 (C. Elegans) of the Project. Gradient Descent algorithm type is mini-batch gradient descent with momentum. Regularization was added as an option but, not used (lambda = 0). Main programming language used in this project was MATLAB.

**Image Processing for C. elegans**

“DetectionHelper” was converted into a function called “DetectionHelper\_function” and used inside “Image\_processing” script. In this script, the height and the width of the rectangles were changed to be the same size to acquire a square image of the same size for each C. elegan. Additionally, the object detected was placed into the center of the box to acquire images with C. elegans in the center of the image. This was believed to be successful to some degree due to some images having defects along with the C. elegans and also some boxes containing no C. elegans or defects. Further processing was done to cropped images of C. elegans such as z-score transformation and resizing (shrinking) images to reduce features for training purposes in “read\_and\_resize”. Additionally, “read\_and\_resize” creates training and test data (combined at this point) by flattening each image and appending it to a matrix and saving this matrix as “Original.mat” into the automatically created “TrainingData” folder. Examples of some of these processed images are given below to show the difference between the image quality. For some particular images, it is hard to distinguish whether it is a worm or not a worm even for the human eye. Images detected in the corners of the microchip were neglected at this point.



**Figure - 1:** Various examples of processed images

**Validation procedure for your implementation**

The validity of the softmax regression implementation was checked with the “**fisheriris**” data that can be loaded directly from the MATLAB. A “softmaxregression” function was written by the group and the results were compared to the results from the **built-in functions** “mnrfit” (training) and “mnrval” (prediction). Both methods gave a training accuracy of 98.67% and matched exactly. Due to lack of examples (150 examples), all of the data was used for training and only the training accuracy was compared. The reason for not checking the validity via the MNIST or C. elegans data was because mnrfit was not efficient enough to converge to an answer in a timely manner due to number of features and examples in these datasets. The code can be run via “Implementation\_check.m” script and the results are printed on the command window.

**Input image size**

**For the MNIST data**, each image is a 28X28 image for both training and test datasets. When each image is flattened, there are 784 features for each example. Adding a bias increases this size to 785 features.

**For the C. elegans data,** resized images were used for training with the size of 16X16 both for training and test datasets. When each image is flattened there are 256 features for each example. Adding bias increases this size to 257 features.

**Number of classes and total number of images in each class**

**For the MNIST data**, there are 10 classes that are handwritten digits between 0-9.

**For the C. elegans data,** there are 2 classes which are worm or no worm.



**Training set and test set sizes**

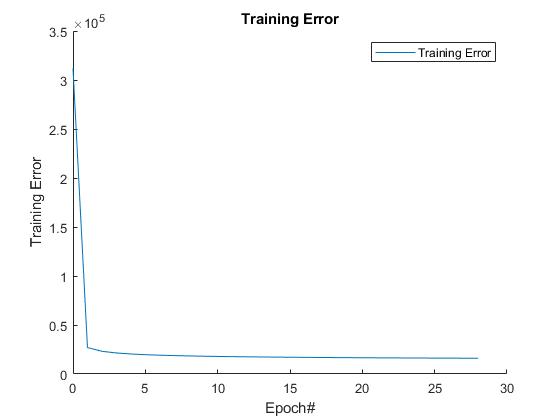
**For the MNIST data**, there are a total of 60,000 training images and 10,000 test images available.

**For the C. elegans data,** there are a total of 6,240 training images and 41,79 test images available. Separation between training and test is done within the “Part\_2” script and it is flexible: User can choose the percentage of training and test data. 60/40 was the personal choice of the group in this case.

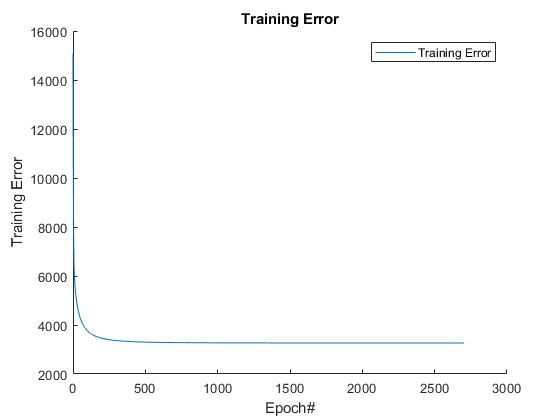
**Training set parameters and training times**

**Training Error & Training/Test Accuracy**



**Figure - 2 :** Training Error at each Epoch in the **MNIST data**



**Figure - 3:** Training Error at each Epoch in the **C. Elegans data**



**Conclusion & Suggestions**

Using mini-batch gradient descent with momentum speeds up the training process tremendously. Softmax regression with linear mapping works well with images that are processed to amplify the pattern to be recognized by the algorithm such as MNIST database. Low quality images with a lot of background noise that are hard to remove such as defects, etc. produces a lower accuracy when compared to high quality images of MNIST. For this kind of work, it is suggested to use high quality images with no defects (good quality microchips) with the same size images of microchips (a good camera with a standard image size) to capture the pattern more effectively. It is also suggested to use a lot more examples to increase the accuracy that might be affected by the position and shape of C. elegans.

**APPENDIX:** **LIST OF FUNCTIONS, SCRIPTS, FOLDERS, DATA**

|  |  |  |
| --- | --- | --- |
| **LIST OF ALL THE FUNCTIONS & SCRIPTS** | | |
| **LIST** | **Type** | **Description** |
| **DetectionHelper\_function** | function | Detects objects in images |
| **featureNormalize** | function | Z-score normalization |
| **Implementation\_check** | script | Compare softmaxregression function with built-in mnrfit and mnrval |
| **loadMNISTImages** | function | Load all MNIST images |
| **loadMNISTLabels** | function | Load all MNIST labels |
| **Part\_1** | script | Full script for Part-1 of Project 4 (MNIST) |
| **Image\_processing** | script | Script for pre-processing of C. elegands images |
| **read\_and\_resize** | script | Script for creating training data from the cropped C. elegans images |
| **softmaxregression** | function | Softmax regression function (linear only) |
| **Part\_2** | script | Full script for Part-2 of Project 4 (C. elegans) |

|  |  |  |
| --- | --- | --- |
| **LIST OF ALL THE DATA &FOLDERS** | | |
| **LIST** | **Type** | **Description** |
| **train-images.idx3-ubyte** | .idx3-ubyte | Training Images for MNIST |
| **train-labels.idx1-ubyte** | .idx1-ubyte | Training Labels for MNIST |
| **t10k-images.idx3-ubyte** | .idx3-ubyte | Test Images for MNIST |
| **t10k-labels.idx1-ubyte** | .idx1-ubyte | Test Labels for MNIST |
| **NemaLife Images\_Converted** | folder | Contains original images of C. elegans |
| **BoxedImages** | folder | Contains cropped images of C. elegans |
| **TrainingData** | folder | Contains saved .mat file (input for training and testing) for C. elegans |
| **Combined\_Labels** | .xlsx | All the labels for all the cropped C. elegands images |
| **ML\_PROJECT4\_REPORT** | .docx | Project 4 Report |

**Formula for Accuracy:**