Assignment 1

Statistical Learning – Fall 2018

Assignment Date: 1397/08/01

Due Date: Section 3) 1397/08/19

Section 3

Use of Stochastic gradient descent to estimate the parameters of logistic regression.

Data set

notMNIST data set is a data set with 10 classes, (letters A-J taken from different fonts). Here are some examples of letter "A"

For this assignment we want to build a model differentiating between letters "C" and "J". Use the following code to get the related data. At the end you will have 3500 training images, 100 validation images and 145 test images (taken from a UoT stat-learn course).

```python
with np.load("notMNIST.npz") as data :
    Data, Target = data ["images"], data["labels"]
    posClass = 2
    negClass = 9
    dataIndx = (Target==posClass) + (Target==negClass)
    Data = Data[dataIndx]/255.
    Target = Target[dataIndx].reshape(-1, 1)
    Target[Target==posClass] = 1
    Target[Target==negClass] = 0
    np.random.seed(521)
    randIndx = np.arange(len(Data))
    np.random.shuffle(randIndx)
    Data, Target = Data[randIndx], Target[randIndx]
    trainData, trainTarget = Data[:3500], Target[:3500]
    validData, validTarget = Data[3500:3600], Target[3500:3600]
    testData, testTarget = Data[3600:], Target[3600:]
```

Model

The aim is to construct a logistic regression model where the input is **x** (the vector consists of all pixels of an image of size 784 (28*28)) and the output is zero or one (it is class “C” or “J”). The logistic regression can be written as:

$$y = \frac{1}{e^{\mathbf{w}^t\mathbf{x}+\mathbf{b}}}$$

The goal is to find **w** (of size 784) such that the loss function gets minimum.

Loss function

1- Use the maximum likelihood loss function that we discussed in class
2- Use the binary cross-entropy loss function,

$$\sum_{n=1}^{N} \frac{1}{N}\Big[-y^{(n)}\log\hat{y}(\mathbf{x}^{(n)}) - (1-y^{(n)})\log(1-\hat{y}(\mathbf{x}^{(n)}))\Big]$$

3- Use the regularized binary cross-entropy loss function

$$\sum_{n=1}^{N} \frac{1}{N}\Big[-y^{(n)}\log\hat{y}(\mathbf{x}^{(n)}) - (1-y^{(n)})\log(1-\hat{y}(\mathbf{x}^{(n)}))\Big] + \frac{\lambda}{2}\|W\|_2^2$$

Learning:

Implement a program (preferably using TensorFlow) that finds the best "W" for each of the loss function above.

Apply SGD with weight-decay of $\lambda = 0.01$ for optimization and use 5000 iterations and mini-batch size of 500.

For each loss function, adjust the learning rate.

What to report

1- Plot the training curves for both cross-entropy loss and classification accuracy vs. the number of epochs
2- Plot the validation curves for both cross-entropy loss and classification accuracy vs. the number of epochs
3- Find the test classification accuracy obtained from the logistic regression model.