Instructor: Aykut Erdem
TAs: Levent Karacan
Tugba Erdogan

Assignment 3

Due on November 24, 2017 (23:59:59)

Instructions. There are two parts in this assignment. The first part involves a series of theory questions and the second part involves coding. The goal of this problem set is to make you understand and familiarize with Neural Network algorithm.

PART I: Theory Questions

1. What are differences between logistic regression and linear regression?
2. What are differences between logistic regression and naive bayes methods?
3. Assume you have a network as given below and $tanh(x)$ function is used as activation function. Show forward and backward propagations with notations.
4. Suppose you are using a 2 layer neural network that has 2 input units (+1 bias), a hidden layer with 2 input units (+1 bias) and one output unit. Suppose you have learned

$$\theta^{(1)} = \begin{bmatrix} 1 & -1.5 & 3.7 \\ 1 & 5.1 & 2.3 \end{bmatrix}$$

which is used to compute second layer's activations, and $\theta^{(2)} = [1 \quad 0.6 \quad -0.8]$ which is used to compute output layer's activation ($h_\theta(x)$). Suppose we swap the weights as

$$\theta^{(1)} = \begin{bmatrix} 1 & 5.1 & 2.3 \\ 1 & -1.5 & 3.7 \end{bmatrix}$$

and $\theta^{(2)} = [1 \quad -0.8 \quad 0.6]$. How will this change the $h_\theta(x)$? Explain.

PART II: Classification of Fashion Products using Neural Network

For this assignment, you will implement a single layer and multilayer neural network architecture to classify mnist-fashion images to 10 classses.

1 Single Layer Neural Network

In the first step, you will implement the network given in Figure 1 and train the network feeding by given training set as 784 dimensional gray-level image values. It is important to normalize image values $(0 - 255)$ to between 0 and 1. We can express this network mathematically as:

$$o_i = w_{ij}x_j + b_i \tag{1}$$

As loss function, you will use sum of negative log-likelihood of the correct labels. Write a python function to compute loss function. Then you will update network parameters w and b to minimize the loss function using gradient descent algorithm. You will implement a function which computes the derivative of the loss function with respect to the parameters. To make sure your function is correct, you must also implement numerical approximation of gradients.

Write a function to minimize your cost function using mini-batch gradient descent. You should try different learning rate$(0.005 - 0.02)$ and batch sizes$(16 - 128)$. Make a table to show learning performance for each setting you tried.

Finally, you will visualize the learned parameters as if they were images. Visualize of the each set of parameters that connect to $O_0, O_1, \ldots, O_9$. Please discuss the visualization of parameters that your model learned. You have sample code for how to visualize images.

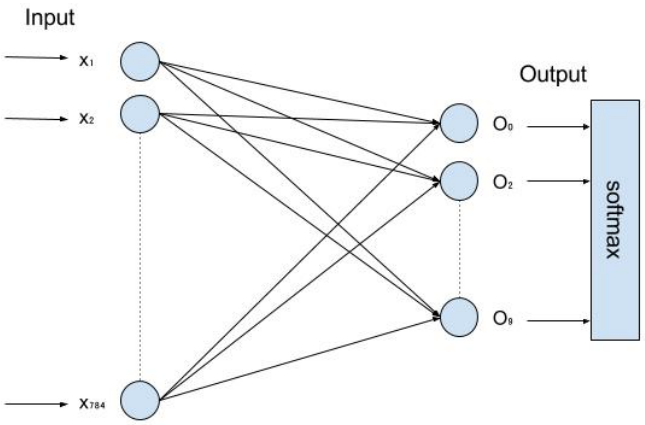

Figure 1: Single layer neural network.

You will use fashion-mnist `https://github.com/zalandoresearch/fashion-mnist` dataset which is explained in Dataset section 3, for both classification methods.

2 Multi Layer Neural Network

In this part of the assignment, you have to implement multi layer neural network for classification. In other words your network consists of one input layer, n hidden layer(s) and one output layer. You will implement forward and backward propagations with the loss function and learning setting as explained in the previous section. Actually, you will implement a back-propagation algorithm to train a neural network.

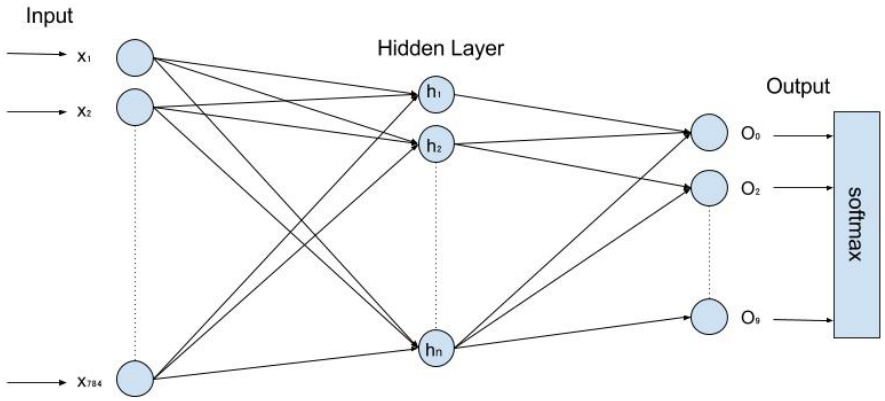

Figure 2: Multi layer neural network.

Training a network

- You should determine the number of units in your hidden layer.
- You should determine batch size as you learned in the class.
- You should determine a learning rate for your gradient descent method.
- Remember, learning rate parameter may be a problem (too big - may not converge, too small - very slow convergence). For this reason you can define a learning rate decay parameter. You will start with a learning rate value and after each epoch you will reduce the learning rate by multiplying it by a decay rate. This operation can deal with mentioned problem.
- You can use different activations functions: Sigmoid, tanh, ReLU etc.
- You can use different objective functions: Cross-entropy error, sum of squared error (SSE) etc.
- You can control your implementation by plotting loss. You can see if it converges or if it needs a different parameter setting.
- You should discuss about your each experiment in the report. Comment about their effects.
- Save your trained models to use later in test time.

Notes:

- You will implement single layer neural network and run experiments on fashion-mnist dataset. You'll change parameters (activation func., objective func. etc.) and report results with a table format in your reports.
- You will implement a neural network which contains one hidden layer. You'll change the mentioned parameters (unit number in the hidden layer, activations function etc.) and report the results.
- Then you'll change your architecture and use a network that contains two hidden layers. Repeat same experiments and comment about the results.
- You have to comment about results and parameters' effects.
- Your implementation should be reproducible, in other words, do not write separate code for each architecture. If you use n hidden layers, your method should create a n-layer network and learn the classifier.)
- Comment your code with corresponding mathematical functions and explain what is going on your code in the code scripts.

Submit

- report.pdf (PDF file containing your report)
- theory_questions.pdf (PDF file containing answers for theory questions)
- code/ (directory containing all your codes as Python file .py)
- model/ containing your learned parameters to use in test time (you can use numpy.save and numpy.load)

Running

Someone manage to run your code as shown :

python train.py –data_path /path/to/train/data

python test.py –data_path /path/to/test/data –model_path /path/to/tranined/model

You are free to add arguments for loss function, hidden layer size, activation function, etc.(see "argparser" library in Python). Please give information about your arguments and how to run your code. Make sure you saved the trained parameters to use in test time.

The ZIP file will be submitted via Github Classroom. The answers of theory questions (part 1) will be summited as a pdf file prepared with Latex to same github repository

NOTE: To enter the competition, you have to register Kaggle in Class with your department email account. The webpage of the competition will be announced later. Top 5 assignment will earn extra points.

3 Dataset

- fashion-mnist dataset `https://github.com/zalandoresearch/fashion-mnist` contains handwritten digits and corresponding labels. Dataset contains 60000 training and 10000 test images. For Kaggle competition, we do not use same splits with original dataset. Please use shared data with you.
- Dataset and the code to load it are provided to you. You can donwload them from `https://drive.google.com/open?id=1JRvVhLBxlm57GsB0RMAVE5EBX1c09mTz`. The dataset is split into three sets: Training (60000 images), test (10000 images)and validation (10000 images). You can use validation set for training (60000+10000 images).
- When you load the dataset, you'll see that every image is represented with [784x1] vector and has a label. You'll use the given representation (do not have to extract new features).

Grading

- Code (58) : 18 points for single-layer neural network, 40 points for multi layer neural network.
- Report(42): Theory part: 12 points, Analysis of the results for classification: 30 points. You have to write your report with LaTeX.

  Notes for the report: You should analyse the method you employed. How did you improve your results? Explain every step you choose. Comment about the results. Compare single layer and multi layer neural network. Comment about the activation functions, loss functions etc. that you used for your experiments. Your reports have to include your classification accuracy.

Late Policy

You may use up to five extension days (in total) over the course of the semester for the three problem sets you will take. Any additional unapproved late submission will be weighted by 0.5.

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specic problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else. [1]

[1] This assignment is adapted from http://www.cs.toronto.edu/ guerzhoy/321/proj2/