# **Homework 2**

# **COSC6342: Machine Learning**

# **University of Houston**

# **Department of Computer Science**

# **Sent on: Sept. 23, 2019**

**Due: Oct. 7, 2019**

|  |  |
| --- | --- |
| Name(s) | PeopleSoft Id. |
|  |  |

**General instructions:**

Teams can be made of up to 3 members. Each team member should submit the same final one document. The document should have their code as an appendix. The names and peopleSoftIDs of all the members should be contained in the document (1st page).

**Motivation:**

The main goal of this assignment is to design a tool to train and monitor a neural network. Using this tool, you would be able to analyze the behavior of the network and eventually answer the questions at the end of this document.

**Implementation:**

You can use any programming language such as Python, Matlab, etc. You can also rely on any existing toolbox (Matlab, Tensorflow, Pytorch, etc.) to run a neural network. It is highly recommended to design your code in a modular manner so you would be able to reuse it and change it. You should link your code to an existing toolbox that already implements a neural network. The core of the assignment is the backpropagation learning algorithm. Assume an input layer, a hidden layer and an output layer in your neural network.

**Requirements:**

Define the following items as input of the tool:

* + Activation unit (sigmoid, hyperbolic tangent or rectifier (ReLU))
  + learning rate, momentum
  + the number of iterations
  + the batch size
  + number of hidden layers, number of nodes in each layer separately (including input, hidden and output)
  + The input dataset and the corresponding class labels. You can choose any dataset you prefer. Some potential datasets include the MNIST or a dataset from UCI repository [[link](https://archive.ics.uci.edu/ml/datasets.html)].
* The user should be able to see plots after each iteration and at the end of the training. The plots include a histogram of activation of one selected hidden unit, the training and test error along iterations and the weight change for a selected hidden unit along iterations. In order to do so you can either use the debugger to pause the training after each iteration and plot the figures you want (e.g. in Matlab) or design your code in an interactive way.

**Reports:**

In your final report, include the following items:

* The dataset that you have chosen and their feature types and ranges of values and number of classes
* The parameter settings for the network e.g. nonlinearity, learning rate, momentum, etc. (refer to requirements section)
* The training and test error chart along iteration (see figure below)
* The histogram of activation for 2 hidden units from each layer (see figure below)
* The weight change along iterations for two hidden units in each layer (see figure below)
* Explanation of the experiments you designed to answer the questions below.

**Questions you should answer in your report:**

You should design appropriate experiments using your tool and setting different input parameters to answer the following questions. It is better to fix all parameters and only change one parameter to observe the effect of that parameter and answer the question.

A) **Weight changes** - By plotting the weight changes for hidden units along iterations, answer the following questions:

1. Do the weights of units in the same layer become stable almost at the same time? Does it depend on the layer?
2. Do the weights in different layers become stable around the same time? Explain.
3. In which iteration do you think the model overfits the data? Do you see any relation between the weight change and overfitting?

B) **Activation changes** – By plotting the histogram of activations for hidden units along iterations, answer the following questions:

1. Do different neurons have similar distribution of activations in the same layer? Explain.
2. Do different neurons have similar distribution of activations in different layers? Explain.

C) **Parameter changes** – By changing the input parameters and training the network, answer the following questions:

1. How does nonlinearity (sigmoid, tanh or ReLU) affect overfitting?
2. How does the number of hidden layers and number of units in hidden layers affect the training?

**Definitions:**

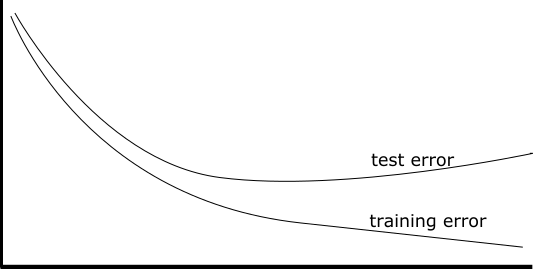
* + **Non-linear function** – The non-linear functions used in this assignment include logistic sigmoid [[link](https://en.wikipedia.org/wiki/Sigmoid_function)], hyperbolic tangent [[link](https://en.wikipedia.org/wiki/Hyperbolic_function)] and rectified linear unit [[link](https://en.wikipedia.org/wiki/Rectifier_(neural_networks))]. You would probably need to implement these functions and their derivatives.
  + **Activation unit –** The activation of a hidden unit is defined as the result of applying non-linear function on the projected input using the input weights. The activation is defined as follows:

where f is the non-linear function (sigmoid, etc.) W is the input weight vector to the hidden unit, x is the input and b is the bias.

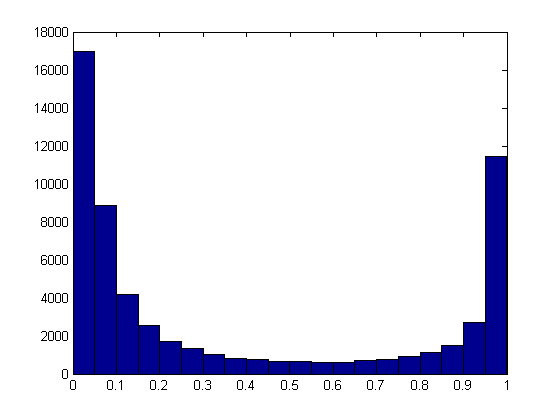
* + **Weight change –** the weight change for hidden unit h in iteration t is defined as the angle between the input weight vector to the hidden unit at time I and time t-1. You can use the dot product to compute the angle between two vectors.
  + **Over-fitting** *- A model starts to over-fit the data if it keeps improving on training data while the performance starts to degrade in over test data. In order to observe over-fitting, you can train your model on 2/3 of the data and compute the accuracy on both the training data and the rest 1/3 test data in each iteration.*
  + **Stable weight vector** *- A weight vector is considered as stable if it does not change for a while along several iterations (refer to weight change).*

**Sample outputs:**

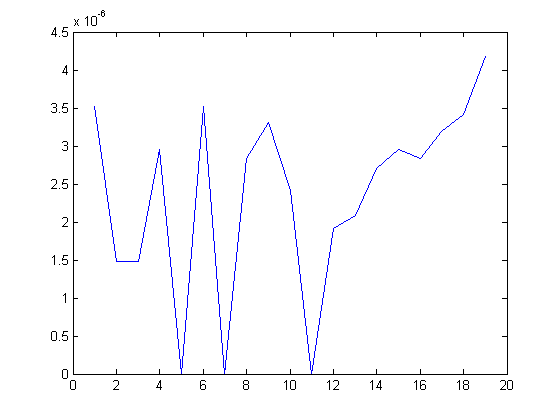
* The figure below shows an example of training versus test error along iterations. The X axis corresponds to the iteration and the y axis corresponds to the error.



* The figure below illustrates a sample output for histogram of activation of a hidden unit. The X axis corresponds to the activation value and the y axis corresponds to the frequency of samples with that activation value.



* The figure below corresponds to the weight changes for a hidden unit. The X axis corresponds to the iteration. The Y axis corresponds to the angle between the weight vectors of a sample hidden unit in two consecutive iterations.



**Additional Guidelines:**

* The assignment is ideal for teamwork. We recommend teams of 3 people. However smaller teams are also allowed.
* Make sure your report contains enough charts to support your conclusions. Also, do not forget to add your code as an appendix.
* Remember you are not allowed to copy code or ask someone else to provide code to you and your team; in case of occurrence, this will be severely penalized.