Machine Learning Laboratory 3. Homework Neural Networks Report

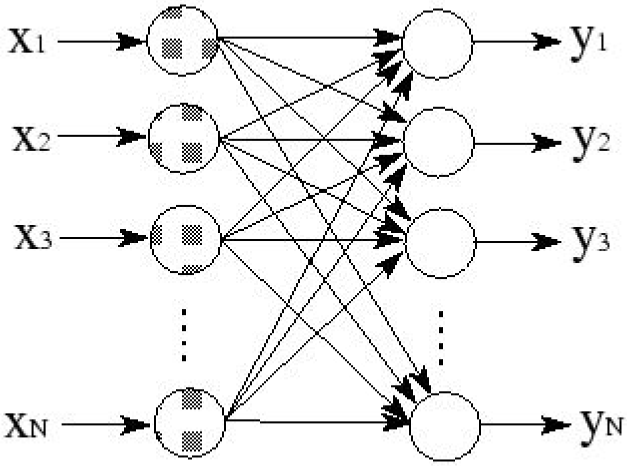
Name : Alper Fırat Kaya
Student No : 21228467

Index

1. **Single Layer Neural Network**

   a. **Introduction**

By connecting multiple neurons, the true computing power of the neural networks comes, though even a single neuron can perform substantial level of computation. The most common structure of connecting neurons into a network is by layers. The simplest form of layered network is shown in figure.



The shaded nodes on the left are in the so-called *input layer*. The input layer neurons are to only pass and distribute the inputs and perform no computation. Thus, the only true layer of neurons is the one on the right. Each of the inputs $x_1x_2x_3.....x_n$ is connected to every artificial neuron in the output layer through the connection weight. Since every value of outputs $y_1y_2y_3...y_n$ is calculated from the same set of input values, each output is varied based on the connection weights. Although the presented network is *fully connected*, the true biological neural network may not have all possible connections - the weight value of zero can be represented as ``no connection". [1]

   b. **How to run the program**

First you need to put the data files to the same folder with the program or you can specify the locations of the data files.
Then if you want you can set the learning rate ,batchsize and more
You can see more with “python main.py help”

c. How does it work

We have 28 x 28 = 784 inputs which have normalized values between 0-1 and we initialized our weights and bias between 0-0.01

$\sum_{i=0}^{i=784} x_i \ * w_i$ with this formula we calculate each neurons value

Then we give these values to the softmax function which gives us probabilities for each neuron(feedforward) . Then we calculate loss with negative log likelihood function. We need to minimize the loss for doing that (Calculating Loss). We get the derivative of loss respect to the w . We update our weights and biases according to loss for each image (back propagation)

d.Results

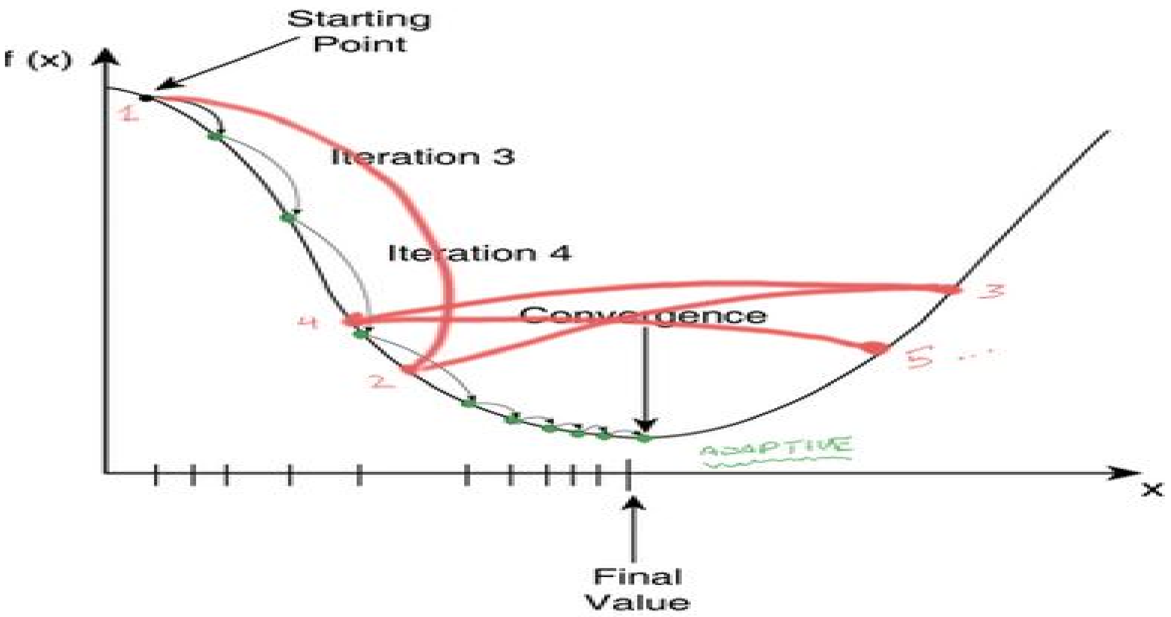
Tuning Learning Rate

Batch Size	Learning Rate	Loss	Accuracy
16	0.001	0.533	0.810
16	0.005	0.444	0.838
16	0.01	0.417	0.835
16	0.15	0.454	0.826
16	0.2	0.507	0.801

- Bigger Learning Rate
  - Advantages : Faster computation
  - Disadvantages : leads to bigger jumps hard to find global minimum of the loss function
- Smaller Learning Rate
  - Advantages : We can find global minimum more precise

  - Disadvantages : It can be slow

Bigger Learning Rate : Red ,Smaller Learning Rate : Green



Tuning Batch Size

Batch Size	Learning Rate	Loss	Accuracy
50	**0.01**	**0.443**	**0.839**
75	**0.01**	**0.419**	**0.843**
100	**0.01**	**0.418**	**0.843**
125	**0.01**	**0.417**	**0.844**
1280	**0.01**	**0.401**	**0.848**

Advantages :

- It requires less memory. Since you train network using less number of samples the overall training procedure requires less memory. It's especially important in case if you are not able to fit dataset in memory.
- Typically networks trains faster with mini-batches. That's because we update weights after each propagation.

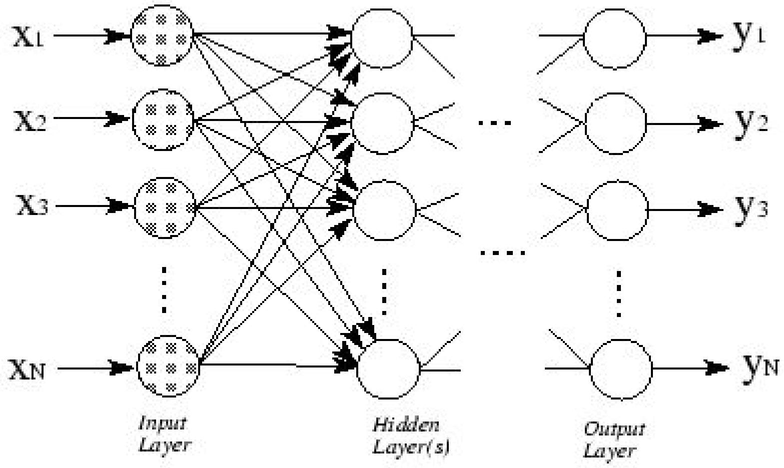
Disadvantages:

- The smaller the batch the less accurate estimate of the gradient .As seen above

4. Multi Layer Neural Network

a. Introduction

To achieve higher level of computational capabilities, a more complex structure of neural network is required. Figure shows the *multilayer neural network* which distinguishes itself from the single-layer network by having one or more *hidden layers*. In this multilayer structure, the input nodes pass the information to the units in the first hidden layer, then the outputs from the first hidden layer are passed to the next layer, and so on.



Multilayer network can be also viewed as cascading of groups of single-layer networks. The level of complexity in computing can be seen by the fact that many single-layer networks are combined into this multilayer network. The designer of an artificial neural network should consider how many hidden layers are required, depending on complexity in desired computation. [2]

b. How to run the program

Same as Single Layer but you need to specify -hl argument for program to run multiple layer neural network.For example -hl 1 has 1 hidden layer -hl 2 has 2 hidden layers and so on. I calculated the neuron count on hidden layers (0,9 * previous layer) +10

c. How does it work

Same as single layer but on hidden layers.I used relu activation function. On last layer network uses softmax function.

d. Results

Hidden Layer Count	Batch Size	Learning Rate	Loss	Accuracy
1	1280	0.01	0.754	0.780

4.References

[1]http://www.ece.utep.edu/research/webfuzzy/docs/kk-thesis/kk-thesis-html/node15.html

[2]http://www.ece.utep.edu/research/webfuzzy/docs/kk-thesis/kk-thesis-html/node16.html