**Programming Assignment-4: A Two-Layer ANN[[1]](#footnote-1) Part I**

**First Name1: \_Richu\_\_ Last Name1: \_\_Mathew\_\_\_**

**First Name2:\_\_Andrew\_\_\_\_ Last Name 2: \_\_King\_\_\_\_\_**

**Introduction**

In this programming assignment, you will build an artificial neural network with a hidden layer. You will implement the backward propagation algorithm and apply it to the given data set.

**1: Plot the training data**

Please load and plot the training data. By visualizing the data, you can see that it is not linearly separable. PLEASE copy the data plot to this document.

Chart, scatter chart

Description automatically generated

**2: Logistic Regression**

Use the logistic regression model you have built in your programming assignment-3 to classify the given data. What is the classification accuracy?

train accuracy: 66.25 %

test accuracy: 66.25 %

Chart, scatter chart

Description automatically generated

**3: Two layer artificial neural network**

In this part of the assignment, you will implement an artificial neural network with a hidden layer.

**Attention: You should use “self.method\_name” if you want to call a method inside a class.**

**Steps:**

1. Define the artificial neural network structure (# of input units and # of hidden units, as we only have one hidden layer).

2. Initialize the network’s parameters

3. Iterations:

- Implement forward propagation

- Compute loss function

- Implement backward propagation to get the gradients

- Update parameters (gradient descent)

**Question 1:** Can we initialize all weights including the b’s to 0? Why?

No we cannot initialize all weights to be zeros because it will make all the nodes identical and won’t learn (which means the weights won’t be updated).

**Question 2:** By using the provided alpha, # of epochs, and # of hidden units, what is the classification accuracy?

Accuracy: 90%

**Question 3:** For the given code, is this batch gradient descent or stochastic gradient descent? Why?

Gradient descent because the weights are updated once per iteration

**Question 4:** Please play with alpha, # of hidden units, and # of epochs, what’s the highest classification accuracy?

|  |  |
| --- | --- |
| **Hyperparameter & values** | **Loss & accuracy** |
| alpha = 0.5  # of hidden units = 4  # of epochs = 10,000 | Loss after iteration 0: 277.257411  Loss after iteration 1000: 120.808039  Loss after iteration 2000: 114.017234  Loss after iteration 3000: 110.906721  Loss after iteration 4000: 108.872125  Loss after iteration 5000: 107.303991  Loss after iteration 6000: 106.034939  Loss after iteration 7000: 104.985819  Loss after iteration 8000: 104.107507  Loss after iteration 9000: 103.364691  Accuracy: 90% |
| alpha = 0.5  # of hidden units = 5  # of epochs = 15,000 | Loss after iteration 0: 277.264633  Loss after iteration 1000: 121.054744  Loss after iteration 2000: 113.329987  Loss after iteration 3000: 97.539967  Loss after iteration 4000: 85.861050  Loss after iteration 5000: 81.244426  Loss after iteration 6000: 78.192675  Loss after iteration 7000: 76.225876  Loss after iteration 8000: 74.817466  Loss after iteration 9000: 73.720184  Loss after iteration 10000: 72.820682  Loss after iteration 11000: 72.061555  Loss after iteration 12000: 71.409970  Loss after iteration 13000: 70.844719  Loss after iteration 14000: 70.350675  Accuracy: 91% |
| alpha = 0.6  # of hidden units = 9  # of epochs = 15,000 | Loss after iteration 0: 277.264791  Loss after iteration 1000: 119.745547  Loss after iteration 2000: 112.731541  Loss after iteration 3000: 108.424626  Loss after iteration 4000: 105.382811  Loss after iteration 5000: 102.979125  Loss after iteration 6000: 101.142865  Loss after iteration 7000: 99.760614  Loss after iteration 8000: 98.672327  Loss after iteration 9000: 97.758287  Loss after iteration 10000: 96.928755  Loss after iteration 11000: 96.142098  Loss after iteration 12000: 95.417155  Loss after iteration 13000: 94.761515  Loss after iteration 14000: 94.170483  Accuracy: 90% |
| alpha = 0.6  # of hidden units = 7  # of epochs = 15,000 | Loss after iteration 0: 277.252307  Loss after iteration 1000: 118.005207  Loss after iteration 2000: 111.451018  Loss after iteration 3000: 107.984533  Loss after iteration 4000: 105.508258  Loss after iteration 5000: 103.497760  Loss after iteration 6000: 101.619006  Loss after iteration 7000: 90.643775  Loss after iteration 8000: 84.056822  Loss after iteration 9000: 80.857011  Loss after iteration 10000: 72.677795  Loss after iteration 11000: 69.993419  Loss after iteration 12000: 68.514937  Loss after iteration 13000: 67.582701  Loss after iteration 14000: 66.901099  Accuracy: 91% |
| alpha = 0.6  # of hidden units = 8  # of epochs = 20,000 | Loss after iteration 0: 277.262042  Loss after iteration 1000: 118.516640  Loss after iteration 2000: 112.015754  Loss after iteration 3000: 107.698233  Loss after iteration 4000: 104.320856  Loss after iteration 5000: 101.877099  Loss after iteration 6000: 100.053904  Loss after iteration 7000: 98.658855  Loss after iteration 8000: 97.561951  Loss after iteration 9000: 96.677426  Loss after iteration 10000: 95.951287  Loss after iteration 11000: 95.347512  Loss after iteration 12000: 94.839003  Loss after iteration 13000: 94.404676  Loss after iteration 14000: 94.028441  Loss after iteration 15000: 93.698114  Loss after iteration 16000: 93.404361  Loss after iteration 17000: 93.139876  Loss after iteration 18000: 92.898822  Loss after iteration 19000: 92.676452  Accuracy: 91% |

The best result from the table above

alpha = 0.6

# of hidden units = 7

# of epochs = 15,000

**Question 5:** Please paste the decision boundary diagram in this document (the one corresponding to the highest accuracy).

Chart

Description automatically generated

**Submission:**

* **Rule1:**
  + If you work with a partner, please name your zipped file as follows:

PA4\_LNAME1\_LNAME2.Zip for folder and PA4\_LNAME1\_LNAME2.docx for a word document, i.e., the file names should include both LAST NAMEs.

* + If you work on your own, the format should be

PA4\_LNAME.Zip for folder and PA4\_LNAME.docx for a word document.

* **Rule2:**
  + Put your FULL names whether working in a group or individual in the word document that answers all the questions.
* **Rule3:**
  + **EVERYONE** in the class should submit this Assignment, which should provide all files (like test excel files etc.. ) that are necessary for the execution of code in the submission folder.
* **Rule4:**
  + Please submit two Jupyter files, each of which is corresponding to Part I and Part II, respectively.

1. Data is obtained from Dr. Andrew Ng’s Machine Learning course. [↑](#footnote-ref-1)