Assignment #HW4 Artificial Neural Network

CSSE490: Bio-Inspired Artificial Intelligence

Prepared and submitted by:

Lyra (Yujung) Lee

Collaboration and resources:

I worked alone

Resources I used to complete this assignment (websites, textbook, friends, etc.):

Stackoverflow

Nature of Code

To pass this assignment you must provide the requested data as listed in this document. You do need to provide at least some minimal evidence of reflection on the results of the plots. If you want to use a different plotting tool that is completely OK, but you should be sure to watch what happens to the accuracy in particularly, although observing the weights can be insightful as well.

Checkpoint #1: Provide a screenshot of your code running a perceptron and outputting the correct output values for the four possible inputs values (00, 01, 10, 11). Also provide the two weights for you perceptron and the weight for bias for in each case. *(This was done by hand in class)*

logical OR

OR values	
w1	1.2
w2	1.5
b	-1

```
Created a perceptron that solves logical OR
Input: (0,0) / Output: -1.0
Input: (0,1) / Output: 1.0
Input: (1,0) / Output: 1.0
Input: (1,1) / Output: 1.0
```

---

and logical AND

AND values	
w1	0.6
w2	0.6
b	-1

```
Created a perceptron that solves logical AND
Input: (0,0) / Output: -1.0
Input: (0,1) / Output: -1.0
Input: (1,0) / Output: -1.0
Input: (1,1) / Output: 1.0
```

Checkpoint #2: Confirm training works on a well-defined example.

A. Provide a screenshot (or list them) of the weights of your "OR perceptron" ***before*** and ***after*** being trained with a single incorrect output. You can do this by providing a training set that has x=0, y=0 and answer is 1. (This is an incorrect answer according to logical OR). This should INCREASE your bias weight ONLY, since the input for x and y are 0.

```
Created a perceptron that solves logical OR
Checkpoint 2: Part A
Perceptron with w1:1.2 w2:1.5 bias:-1.0
Trained with bad data: x=0 / y=0 / answer=1
Perceptron with w1:1.2 w2:1.5 bias:-0.98
----------------------------------------
```

B. Try to train with a CORRECT example (x=0, y=0 and answer is -1), there should be NO change in the weights.

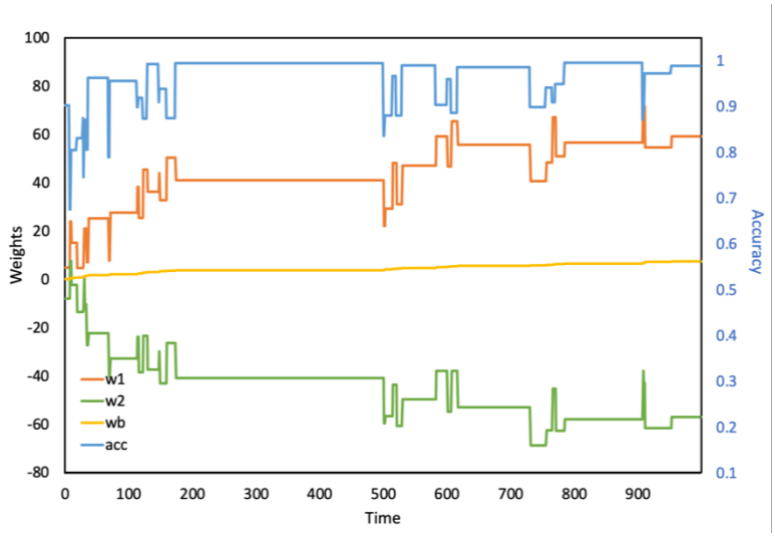
```
Checkpoint 2: Part B
Perceptron with w1:1.2 w2:1.5 bias:-0.98
Trained with data: x=0 / y=0 / answer=-1
Perceptron with w1:1.2 w2:1.5 bias:-0.98
----------------------------------------
```

C. Train using another bad example: x=1, y=1 and answer is -1. (This is an incorrect answer according to logical OR). This should change ALL your weights since the input for x and y are 1.

```
Checkpoint 2: Part C
Perceptron with w1:1.2 w2:1.5 bias:-0.98
Trained with data: x=1 / y=1 / answer=-1
Perceptron with w1:1.18 w2:1.48 bias:-0.96
----------------------------------------
```

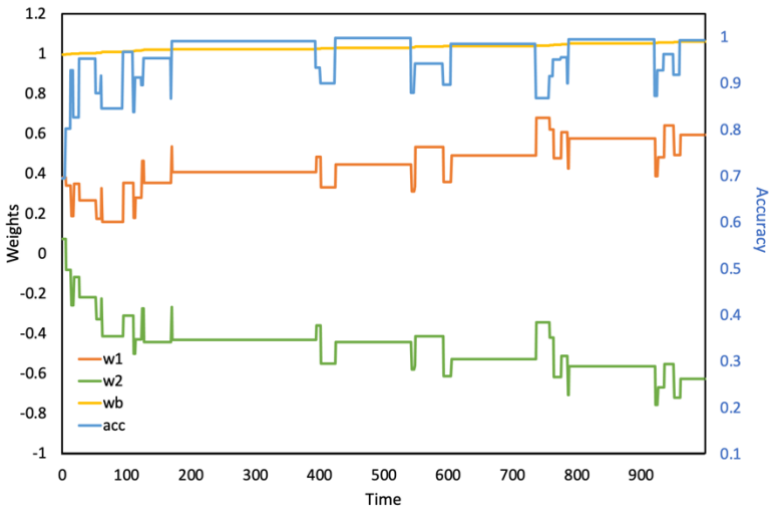
Checkpoint #3: Provide plots and note observations for each of the experiments as listed below:

Experiment 1: high learning rate - Repeat experiment #1, but with a learning rate of 0.1



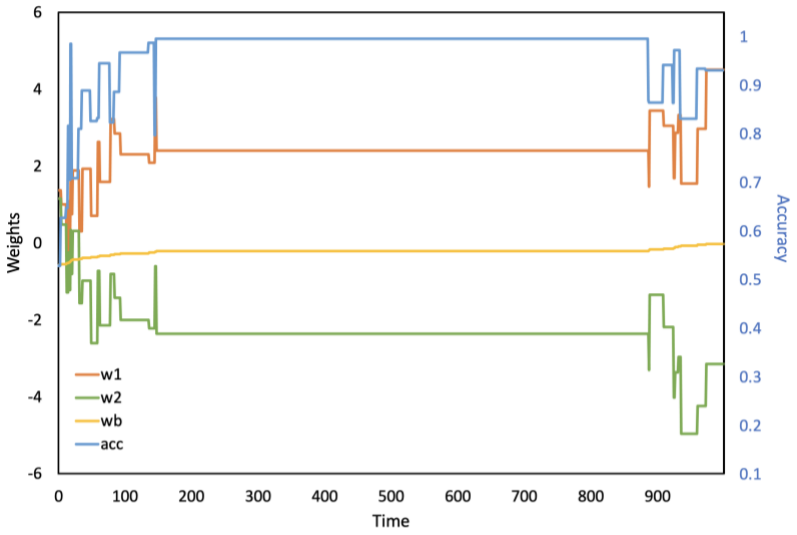
Weight1, weight2, accuracy becomes constant at the same in certain time ranges. Also, weight1 and weight2 is showing like a symmetric shape, which the two graphs start at a close position and end up diverging. However, the bias value does not fluctuate very much, it is almost constant compared to other variables.

Experiment 2: low learning rate - repeat experiment #1, but with a learning rate of 0.001



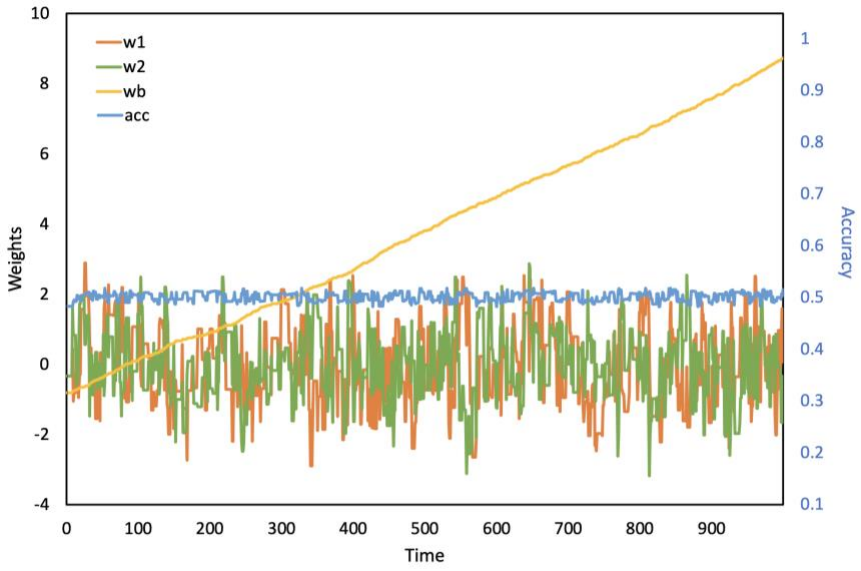
Compared to the first experiment, lower learning rates have much lower values of weights including the bias weights. However, the accuracy seems to be similar when the learning rate is higher. The bias values has very minimal change that the starting value and the ending value has small difference.

Experiment 3: different activation function - Try to use the tanh function instead of step function



As we differed the activation function to tanh function, the perceptron reaches accuracy 100% very quickly and maintains the state when it drops almost at the end whereas the bias value slightly increases from negative to positive.

Experiment 4: XOR repeat experiment #1, but let's give our poor perceptron an impossible task. Let's ask it to solve XOR and see what happens.



The bias value almost linearly increases while the accuracy of the perceptron fluctuates between a certain range that it seems to be staying on approximately 0.5. The most interesting point from this experiment is that weight1 and weight2 value fluctuates from positive to negative values.