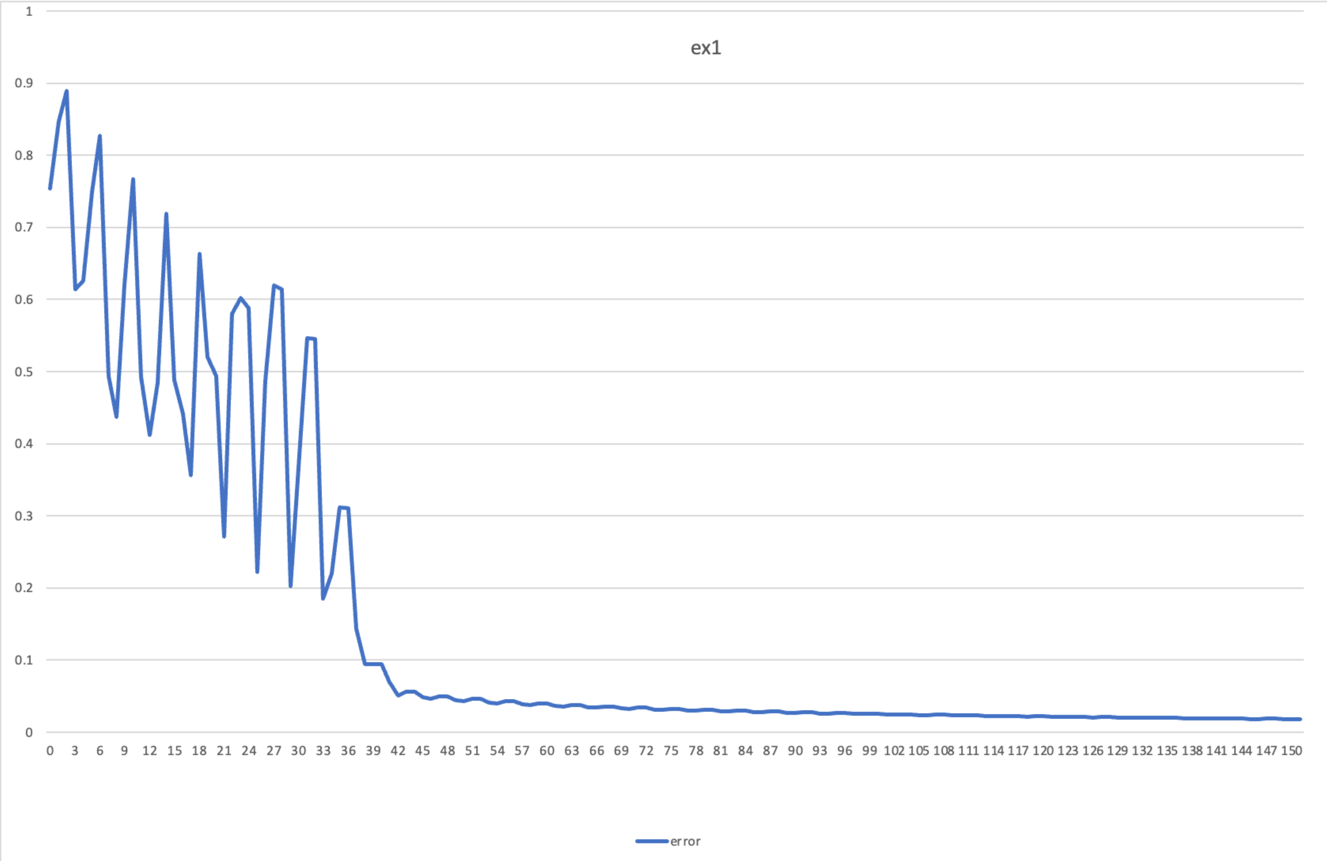
Pear/Thitapa Saelee 300622594

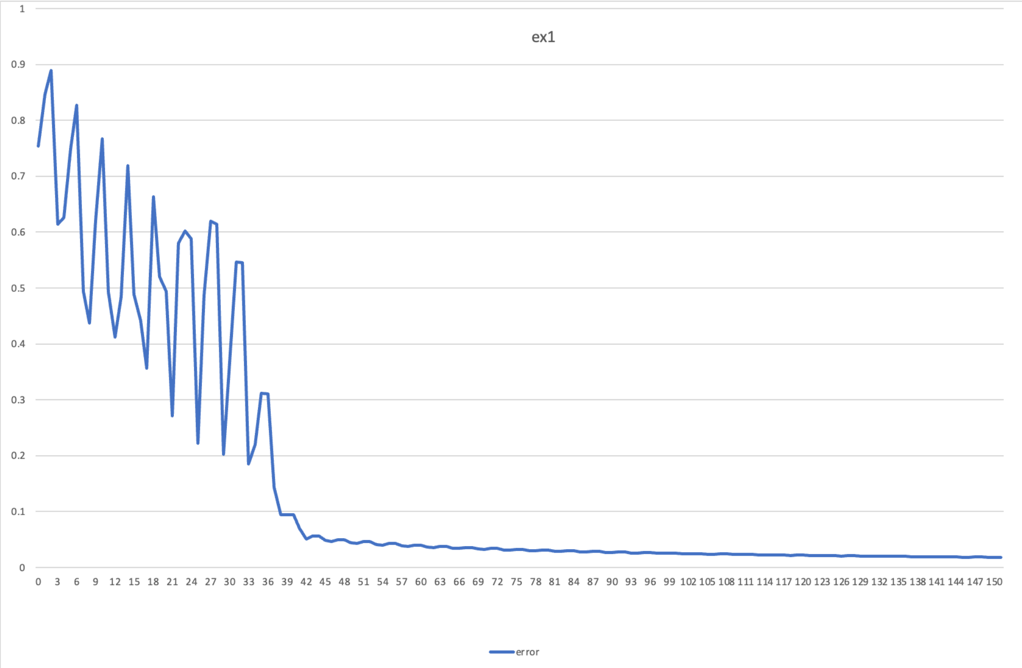
# ML assignment 2

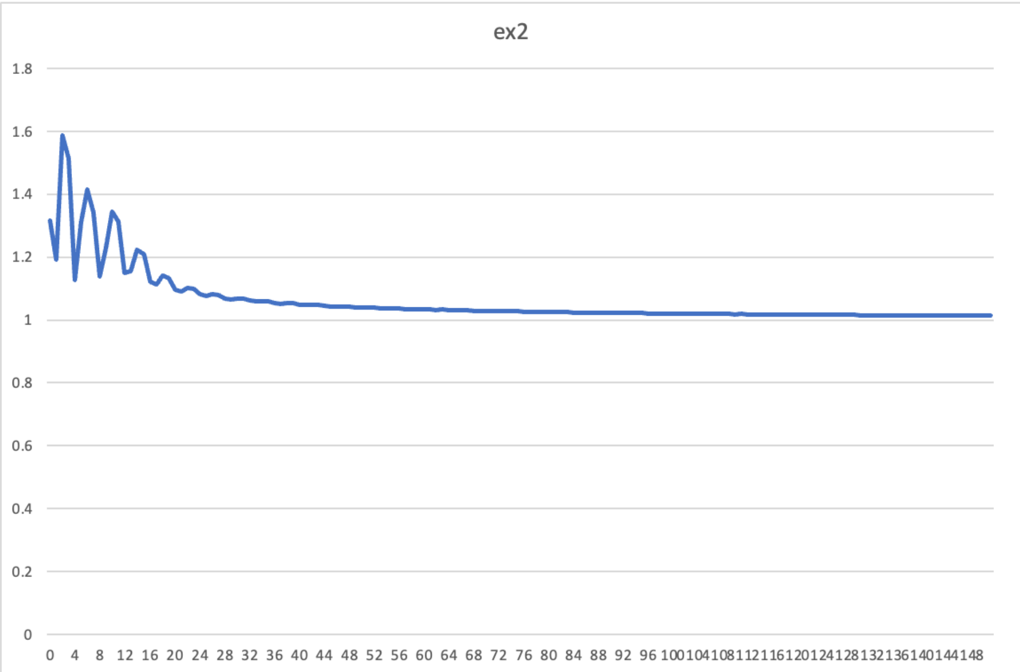
Exercise 1: modified code that runs and search converges




Exercise 2: Two search convergence graphs, for two different training sets.
Explanation of what is the difference (if any) and what is the reason

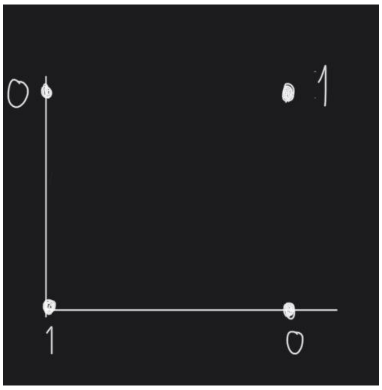
Pear/Thitapa Saelee 300622594




By changing the target value of (0,0) point from 0 to 1(ex2 graph). The search fails as it has reached the limitations, which is called a single-layer network. According to the change of target value from 0 to 1, on one side of the line neuron output y is 0 and on another side it is 1 as a diagram below. The search fails because it is impossible to separate the training sets. As it is not possible to draw a single decision boundary line on ex2 graph, which separates all 1's from all 0's on a graph below. (reading material from engr110)

# Pear/Thitapa Saelee 300622594




The main differences are 1) the ex1 graph converges, but the ex2 graph does not converge due to the change of value from 0 to 1. 2) it is possible to draw a decision boundary line in ex1 graph, but it is impossible for ex2 graph.

Exercise 3: Explanation of the search failure(?) for a two-layer network with the same weights(15%) Explain briefly what is happening.

Because they all started with a value of 0, the two neurons of the hidden layer are identical in weights and bias as they have the same inputs. Therefore, when we calculated how errors change for both of them, we got the same value for both neurons. These two neurons are equivalent because they have the identical input, so they calculate the same gradient and steps in the same way. Due to this, it leads them to produce only one decision boundary line that separates all 1's and 0's, which means that they only have one neuron and it does not work.

Exercise 4: Table of acceptable (search converges) deviation values. Explanation why range is limited (15%) Explain briefly what is happening.

| Derivation value | Converges (yes/no) | After training error |
|---|---|---|
| 0.0 | no | 0.167359 |
| 0.01 | no | 0.167358 |
| 0.1 | no | 0.16729 |
| 1.0 | yes | 0.000997538 |
| 2.0 | yes | 0.000994797 |
| 3.0 | no | 0.187919 |
| 4.0 | no | 0.187898 |
| 5.0 | no | 0.18784 |

| | | |
|---|---|---|
| 6.0 | no | 0.187783 |
| 7.0 | no | 0.187737 |
| 8.0 | no | 0.187677 |
| 9.0 | no | 0.187679 |
| 10.0 | no | 0.187701 |

Because if we weight big derivations, some of the initial numbers can also be large and it should not be too big. For example, if the derivation value is 100.0, the bias might be 100 (sigmoid function is getting input 100 or -100, meaning that the derivation is 0, so the output remains unchanged and the neuron is saturated). The derivation should not be 0, but keep it in the limited range. The reason is if the neuron is saturated, the neuron will not move anywhere and it does not work.