Emmanuel Velazquez
Exercise #4

4.3

After running the simulation, we can see that the left output unit has stronger weights coming from the two leftmost inputs which indicates that these have a significant influence on the activation of the output unit. When we take a look at the right output unit, we are able to notice that the stronger weights are coming from the two rightmost inputs. This implies that these input units have a great amount of significance on the activation of the output. The difference between the weight of the two units allows the network to distinguish the difference between the output units which allows it to learn and categorize the input layers based on their activation.

4.4

The Hebbian-Learning mechanism refers to the saying that neurons that fire together, wire together. This was observed in the Easy task as we noticed that the two leftmost input layers were consistently active together which affected the left output unit and the same held true for the right side. Because of what we know about Hebbian-Learning, we know that the weights for the layers that are co-activated are going to increase and begin to have stronger connections. This style of learning reflects that of the task where the goal was to associate certain patterns of input activation with output layers. The specific activity of the input and the desired outputs have a direct reflection on weight patterns. This allows the network to learn through what looks exactly like the Hebbian-Learning mechanism.

4.5

No, the network never does solve the task. Because the hard task contains layers that overlap without any clear, distinct signals with each output, the Hebbian-Learning mechanism is going to struggle here. The consistent ambiguity and overlap in the input enforces the Hebbian-Learning style to struggle to solve the task, which reflects the limitations that it has with complex patterns. In the beginning we see that MaxEpcs is 40, then it is changed to 50, where it is still unable to solve the task, and we see the same for when it is changed to 100. The final SSE was around 0.54.

4.6

Because the task at hand includes ambiguity, it becomes hard to predict the uniqueness of the input layer to the output layer. Due to all of the limitations of this learning model, it is never going to be able to solve the problem. This is illustrated through the results of changing the MaxEpcs where the goal is to get the SSE value to 0 or very close to it. Changing the MaxEpcs value to 200, my SSE value was 0.4, when I changed MaxEpcs to 300 my SSE was 0.3 and finally for an extreme value where MaxEpcs was 500, the SSE was still far from 0 where it was at about 0.5. With these results, it is evident that the network is never going to be able to solve the problem.