Mat Davidson

CS4990 Practical A.I.

Project 5 – Unsupervised Learning

For this project, I initially designed a complicated scenario in which my dog would learn to predict what I would give him based on where we were in the house. That idea also depended on back propagation, so it was not suitable for this unsupervised learning project. After a while, I finally got a good enough understanding of Hebbian learning in neural networks and came up with a new scenario. Now, I have guy who enjoys dancing, but he is very shy. So, when he goes to the club, he will only dance when no one is looking. Then he discovers that when he has a refreshing beverage, he feels more confident and will dance whether someone is looking or not.

This project was written in Greenfoot, an application used to build visual applications with Java. This program includes two major components and several minor ones. The first major component is the ShyGuy class itself. Upon creation, it initializes a neural network, the second major component. The pair of them work together to determine what ShyGuy is doing. The minor components are mostly visual representations for ShyGuy’s confidence level and the affects of whatever beverage he drinks. There is also a “Bro” who is or is not watching ShyGuy. The binary option of the bro watching and the beverage level are taken together as inputs for the neural network. The network then gives two outputs. When the average of these outputs is greater than 0.5, ShyGuy begins dancing.

The neural network consists of two nodes. The first node handles whether the bro is looking and has low weights. The second accounts for ShyGuy’s beverage level. Since the beverage level is a value between 0 and 1, it is rounded before being entered into the network. The beverage node has high weights. Both nodes use the sigmoid function as their activation function. As a result, when both inputs are 0(the bro is looking and ShyGuy has no beverage), the outputs will be numbers less than 0.5. When the bro is not looking, ShyGuy’s confidence goes up slowly and eventually he will begin dancing. If ShyGuy has a beverage, because of the high beginning weight on that node his confidence will rise much faster.

In order to run the program, start by clicking run. You can give ShyGuy a beverage by clicking the button on the top right. You can Change whether the bro is watching by clicking him. ShyGuy’s beverage level is constantly decreasing, so he will have to have multiple beverages. Overall, the program is simple, but I believe it demonstrates Hebbian learning. The weights update properly, however I’m not sure that it works exactly right. ShyGuy reacts to the beverage and to the bro, but when both inputs are zero, the output is low and constant after so many epochs. My assumption was that ShyGuy would continue to dance with the bro still watching when the beverage’s effect wears off, but that is not the case in the final product. I suspect it is because I am using discrete inputs (1 and 0). I think if I were to parameterize the bro watching input and include the raw beverage value, it may work more like I imagined it. Sadly, I have run out of time to work on this project.