Due 12/05/2019 at 10:00 am

Creating a Self Organizing Feature Map

Bayley King
Dept EECE, University of Cincinnati
2600 Clifton Ave, Cincinnati, OH 45221
king2b3@mail.uc.edu

1.1. Problem Statement

This network is the first implementation that we have completed of an unsupervised learning architecture. The network will compute which neuron is the best performing for that test vector, then change the weights of the neighbors of that peak neuron. Equation 1 shows the equation to find the best neuron. The network then updates all of the weights based off the learning rate,

$$i^* = \arg\min \| w_i - x^q \| \quad (1)$$

the activation of that neuron, and the neighborhood function. Equation 2 below shows the equation used to calculate the change in weights. Equation 3 shows the equation for the neighborhood function.

$$\Delta w_{ij} = \eta \Lambda(i, i^*, t)(x_j^q - w_{ij}) \quad (2)$$

$$\Lambda(i, i^*, t) = \exp\left[\frac{-\|r_i - r_{r*}\|^2}{2\sigma^2(t)}\right] \quad (3)$$

1.2. System Description

The Self Organized Feature Map (SOFM) architecture that we were instructed to use has a 10x10 output map with 29 inputs. The network is trained on a set of 16 animals according to a set of 13 attributes. Each animal is assigned a specific classifier to be used during training which is 16 bits long. The 13 attributes are appended onto the end of the 16 bit classifier to make up the 29 bit input. I decayed my learning rate and neighborhood function every epoch by using the equations seen in Equation 4 below. The parameters that I used for the network can be seen below in

$$\eta(t) = \eta_o \exp\left(\frac{-t}{\tau_L}\right), \eta_o = 0.1 \qquad \sigma(t) = \sigma_o \exp\left(\frac{-t}{\tau_N}\right) \qquad \tau_n = \frac{1000}{(\ln \sigma_o)}, \sigma_o \approx 0.5\, r_{max} \quad (4)$$

Figure 1.1. I ran through lots of simulations using different parameters but found these sets to be the best. If I lowered my max epochs, the network seemed to not train long enough. One test ended up being the best performing test for each neuron, unless I let the network train for at least 4000 epochs. This also would happen if lowered the value of TauL, because then the network would stop learning significantly early during the training process. I set the base Sigma rate to 50, because there are 100 total outputs for the network (Rmax). I use a base learning rate of 0.1 because this was the suggestion in the notes. I then decided to set TauL to 2000 so that the network completed its self-organizing phase during the first half of training. After those first 2000 epochs, the learning rate would then be approximately 0.1 and then had the convergence phase could start. I did not see any major improvements to the SOFMs when letting the network train for any longer than 4000 epochs. All of the weights of the network were randomly initialized using the random library in python.

Base Learning Rate (η_o)	Base Sigma Rate (σ_o)	Rmax	Max Epochs	TauL (τ_L)	TauN
0.1	50	100	4000	2000	750

Figure 1.1: Parameters used for the Network

1.3. Results

The network was then tested on the same data-set, but this time the data was stripped of all its classifiers so that the only data the network was seeing were each test's attributes. Figure 1.2 shows which output had the highest activation for each test vector. Figure 1.3 shows which test vector had the highest activation for each output .

	0	1	2	3	4	5	6	7	8	9
0	-	-	-	-	-	-	-	-	-	-
1	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-
4	-	-	Cat	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	Duck Owl Hawk Fox Dog Wolf Lion Horse Zebra Cow	-	-
6	-	-	-	Dove Eagle	-	-	Goose Tiger	-	-	-
7	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-	Hen

Figure 1.2: Neurons with Maximal Response to each Animal

	0	1	2	3	4	5	6	7	8	9
0	Dog	Dog	Wolf	Wolf	Wolf	Lion	Lion	Lion	Horse	Horse
1	Fox	Fox	Wolf	Wolf	Wolf	Lion	Lion	Lion	Duck	Zebra
2	Duck	Duck	Duck	Wolf	Duck	Tiger	Lion	Duck	Zebra	Zebra
3	Cat	Cat	Cat	Tiger	Tiger	Tiger	Tiger	Duck	Cow	Cow
4	Cat	Cat	Cat	Tiger	Tiger	Tiger	Tiger	Duck	Cow	Cow
5	Eagle	Eagle	Eagle	Tiger	Tiger	Tiger	Tiger	Duck	Cow	Cow
6	Eagle	Eagle	Eagle	Duck	Tiger	Tiger	Duck	Duck	Duck	Duck
7	Eagle	Eagle	Dove	Dove	Goose	Goose	Duck	Duck	Duck	Duck
8	Owl	Hawk	Dove	Dove	Goose	Goose	Goose	Duck	Duck	Hen
9	Owl	Hawk	Hawk	Dove	Goose	Goose	Goose	Goose	Hen	Hen

Figure 1.3: Preferred Response for each Neuron

1.4. Discussion

Figure 1.2 shows that the network's results varied pretty greatly from the example results we saw in our textbook. My network had trouble assigning a unique neuron for each input, and that neuron 57 was the best performing neuron for most of the tests. There didn't seem to be any correlation between what was the best performing neuron for each input and the input. For example all 10 of the tests that identified neuron 57 were all different and had no similarities across them all.

Figure 1.3 showed better results, and for the most part showed that the network effectively clustered the data. Animals with wings were clustered to the bottom of the 10x10 output grid, with smaller animals tending to be clustered to the left. The only major contradiction to this was the Duck, which clustered itself in various places in the SOFM. The Duck was the only test point that ended up being clustered in multiple places on the map. Every other test was clustered in one place on the map. All of the predator animals tended to be in the middle of the map, while the larger non-predatory animals are on the right of the map.

There were slight similarities in figures 1.2 and 1.3 except for neuron 57 which was the highest performing neuron for 10 of the inputs. For the rest of the neurons in figure 1.2, they corresponded to the same outputs in figure 1.3. When I trained for around 2000 epochs, every single input had their maximum response all with the same neuron. I hoped that with more training, the neurons with maximum response to each animal would further diversify, but the output seen in figure 1.2 seemed to be the best results I could reach with my network.

1.5. Conclusions

Even though figure 1.2 showed poor results, the final clustering seen in figure 1.3 showed that the network clustered the data pretty well. With that being said, there wasn’t an even distribution of preferred responses for each neuron, and some data points caused more output responses than others. I think this was due to how I implemented the neighborhood function in my network, or due to how I varied my learning and sigma rates. I didn’t see any improvements when I varied these rates, so I believe that my parameters that set the neighborhood function were what mostly dictated the clustering.

2.1. System Description

There was no major change to the system architecture between problem 1 and problem 2, just a change in the testing set. The network parameters can be seen in figure 2.1 below. The testing set featured 7 new animals that the network was not trained on. Just like in the testing in problem one, all of the classifiers of the data were removed. I saved the final weights of the trained network, and then loaded these saved weights into the network to be used for this new testing set.

Base Learning Rate (η_o)	Base Sigma Rate (σ_o)	Rmax	Max Epochs	TauL (τ_L)	TauN
0.1	50	100	4000	2000	750

Figure 2.1: Parameters used for the Network

2.1. Results

The network was tested on this new testing set which featured 7 new animals that the network was not trained on. These animals were Deer, Bat, Wolverine, Ostrich, Blue Whale, Pig and Humans. Figure 2.2 shows the neurons with the maximum responses for each of these animals, and figure 2.3 shows the preferred response for each output neuron.

	0	1	2	3	4	5	6	7	8	9
0	-	Deer	-	-	-	-	-	-	-	-
1	-	-	-	-	-	-	-	-	-	-
2	-	-	-	Wolverine	-	-	-	-	-	-
3	-	-	-	-	-	-	Bat	-	-	-
4	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	Blue whale	Pig	-
6	-	-	-	-	-	-	Ostrich Human	-	-	-
7	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-	-

Figure 2.2: Neurons with Maximal Response to each Animal

	0	1	2	3	4	5	6	7	8	9
0	Deer	Deer	Deer	Deer	Deer	Wolverine	Pig	Pig	Pig	Pig
1	Wolverine	Wolverine	Deer	Deer	Deer	Wolverine	Pig	Pig	Pig	Pig
2	Wolverine	Wolverine	Wolverine	Deer	Wolverine	Wolverine	Pig	Pig	Pig	Pig
3	Wolverine	Wolverine	Wolverine	Wolverine	Wolverine	Wolverine	Pig	Pig	Pig	Pig
4	Wolverine	Wolverine	Wolverine	Wolverine	Wolverine	Pig	Pig	Pig	Pig	Pig
5	Bat	Bat	Bat	Wolverine	Wolverine	Wolverine	Wolverine	Pig	Pig	Pig
6	Bat	Bat	Bat	Bat	Wolverine	Wolverine	Blue whale	Blue whale	Blue whale	Blue whale
7	Bat	Bat	Bat	Bat	Bat	Bat	Ostrich	Ostrich	Ostrich	Ostrich
8	Bat	Bat	Bat	Bat	Bat	Bat	Bat	Ostrich	Ostrich	Ostrich
9	Bat	Bat	Bat	Bat	Bat	Bat	Bat	Bat	Ostrich	Ostrich

Figure 2.3: Preferred Response for each Neuron

2.3. Discussion

Figures 2.2 and 2.3 showed much better results then figures 1.2 and 1.3. This new testing data was better clustered, with no outlier clustering like the Duck seen in figure 1.3. The human was never a preferred response for any neuron, which can be seen in figure 2.3. I think this is because a human is quite different then all of the animals the network was trained on. Figure 2.3 showed a similar clustering that was observed in problem 1. The two winged animals (Bat and Ostrich) were at the bottom of the network, with larger animals (Pig and Blue Whale) on the right and the predatory animal (Wolverine) generally in the middle. The wolverine had quite a large clustering, but since only 6 animals were selected as preferred responses for the 100 outputs, its understandable that at least one of the animals dominated the clustering. The blue whale had the least amount of preferred neurons, besides the human, which I think is due to how different the blue whale is compared to each of training animals.

It was also interesting to note that only the Ostrich and the Human shared a neuron with their maximum responses, where in problem 1 we had one neuron containing the maximum response for 10 of the 16 training animals. This was quite interesting to see, and I expected this new testing set to have a similar response to the testing set used in problem 1. This leads me to believe that possibly I over trained my network, in problem 1 to the point that a few of the neurons dominated the rest of the network.

2.4. Conclusions

I think that the results of this new testing set were good, and that it proved that the network can cluster data in a un-supervised learning environment. I think that I might have over-trained my original network in problem 1, thats why the original testing set didn't cluster as well as I hoped it would. But this new testing set used in problem 2 showed that the architecture was correct enough to cluster un-trained data properly. If I was to continue working with this network I would want to further investigate the neighborhood function and try training my network with much lower parameters. Lowering these parameters would tell me if I was over-training the network in problem 1, or if the architecture just isn't strong enough to cluster the network on the data it was trained with.