Name: Theodore Margoles
Student ID: 106083192

1.7-Summary

The first thing I tried was playing with the mating method. First I tried splicing the word together half and half. After playing around for a few hours I settled on generating a random variable from [-10, 10] and if it's greater than 0 I choose the character from the first parent to go into the child and if it's less than 0 then I choose the character from the other string. Satisfied with the mating method, I moved on to looking at my method for selecting a mating pool. I tried selecting essentially the 75th percentile in terms of fitness relative to other members of the population. I did this by taking an average between the mean and the best fit member and making this the new survival level. The idea was to eliminate about 50 - 75% of the least fit members from the mating pool. I also tested several different levels of probability for the rate of mutation of the child. I take a random chance of 1 in 100 and if it comes up as a mutation turn on a child then it will select a random char from the character set and change one of the chars in the child string. I increased the population size to 500, and the best performance I got was 82% accuracy with the test phrase “to be or not to be”. I also tried just using lower characters to make training easier. I was also having a really strange bug where my words would loose characters over time and at the end my strings would be shorter than the target. I spent hours trying to debug it and ultimately fixed it by appending new random DNA in the edge cases where a child string loses a character. This was a strange error and im glad its gone! I found that if I added more mutation the training would achieve a higher result sooner but then the evolution would have an odd effect of devolving after a certain point. So I realized I had to keep the mutation rate under control. I tried making mutation rate dynamic with the round number. As rounds went on it became less likely to mutate. Unfortunately I did not see a performance boost from this.

2.5-Summary

The first modifications I made once I got the image algorithm to work with color was to make most of the hyperparameters get passed into my main script so I could test different setups more efficiently. I increased the population size to compensate for the addition of a 3rd dimension to the image matrices. (3D arrays?). I spent a significant amount of time on the mating method. First I tried doing

an average between pixels. I found this created a smoother image however I paid for it in a lack of definition of edges or contrast in the image; as I let it train on and on the image kept loosing color depth and the pixels would all drift towards the 127; halfway from 0 to 255. So I reversed course to just choose a pixel at random from each parent as the mating method. Finally I introduced a hybrid of the two methods: there is a 45% chance the pixel comes from parent 1 there is a 45% chance it comes from parent 2 and a 10% chance that the pixel gets averaged together. I found this was the perfect mix because I got slightly smoother images with more edge definition. Mutation I set up with a random mutation range, such that on a small probability test it would select a pixel at random and increase or decrease its value by a pixel number in the [-10 10]. I also played with these numbers to see what worked. I found 10 worked well. I found over time that about 50-100 rounds of training was reasonable for most of the images. Even after trying to increase the fitness function in the earlier task I still found my best results came from just taking the loss to be the absolute difference between pixels and the same (i, j, k) coordinates, then optimize for the most fit images having the lowest loss by selecting lowest loss members for mating pool.

What I learned:

I learned how to implement a simplified model of DNA evolving over time. I learned how to manage and manipulate color and black and white images in Matlab. I also got a feel for building an artificial project from scratch and then testing all the hyperparameters by hand. I also learned how powerful Matlab is for certain tasks! I also believe I increased by coding ability as well.