Hamming Distance Machine Learning
Luke Wood

Project Description

This is a simple python script that displays the power of the hamming distance fitness evaluation function. The program uses a adds random characters to a string and then evaluates the fitness of this string and adapts it further if needed. At the end of the program, a matlab script is created which will graphically display the success of the progression of the program.

Implementation

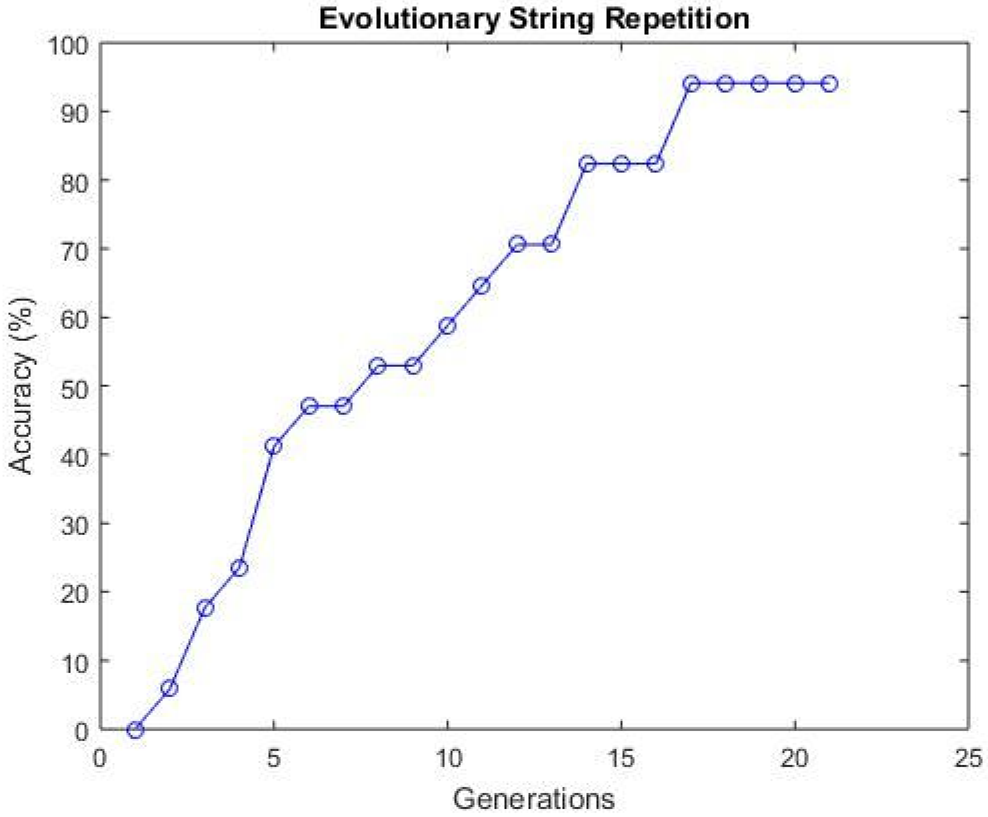
The implementation first part of this implementation was a hamming distance function. This function loops through the string passed to the function and checks each the character in the same position in the target. If the character does not match, the fitness value is changed as needed. Below is the implementation of the hamming distance function in python.

```
def hamming_distance(a,b):
        if(len(a)!=len(b)):
                return -1
        return sum(a0 != b0 for a0,b0 in zip(a,b))
```

Following this I defined a few other functions including random string, random character, and then prompt the user to input a string that the program will then learn to repeat.

Graph

The program generates a matlab script that creates a graph. Below is an example graph that was produced based on a test I ran.



Results and Discussion

This program shows the power of a simple fitness function. The hamming distance function is incredibly easy to implement in almost any programming language and proves very effective for tasks such as this.

Concluding Remarks

While this program does not do anything groundbreaking, it is an interesting visualization of the progression of some basic machine learning.