Kevin Lopez

CIS400

4/1/24

HW2

Implementation

Neural Network Model:

The neural network model is implemented using PyTorch.

It consists of a single hidden layer with ReLU activation and an output layer with sigmoid activation.

Genetic Algorithm:

The genetic algorithm is employed to evolve the population of bit vectors representing the masked weights.

The algorithm involves selection, crossover, mutation, and fitness evaluation steps.

Elitism is weak, with only the best parent retained in each generation.

Fitness Function:

The fitness of each individual (bit vector) is evaluated based on the negative loss (minimizing loss, maximizing fitness) obtained by applying the corresponding masked weights to the neural network.

Data Preparation:

The code uses the digits dataset from sklearn, with two classes for binary classification (0 and 1).

100 data points from each class are kept aside for testing, and the rest are used for training.

The training data is unbalanced and contains randomly relabeled samples.

Hyperparameters

Input Size: Determined by the number of features in the dataset.

Hidden Size: Set to 20 for the 10-20-1 neural network architecture.

Population Size: 420 individuals.

Number of Generations: 256.

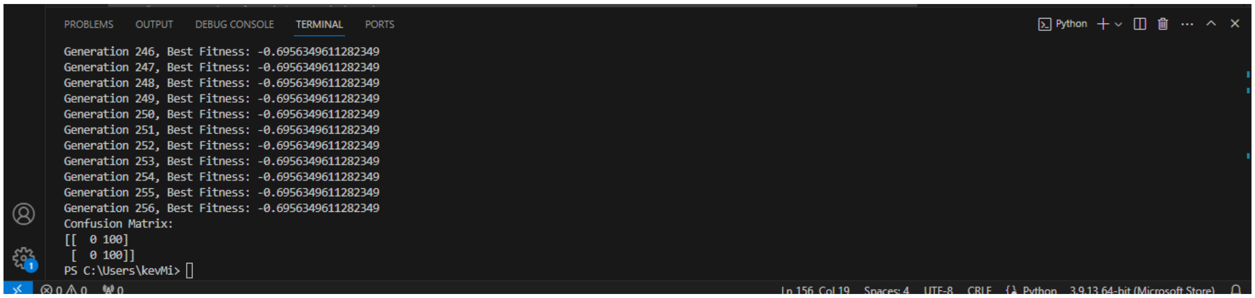
Crossover Probability: 0.9.

Mutation Rate: 2 / Input Size.

Results

The algorithm outputs the best fitness achieved in each generation.

After optimization, the confusion matrix is computed for the best individual on the test data to evaluate the model's performance.



```
Generation 246, Best Fitness: -0.6956349611282349
Generation 247, Best Fitness: -0.6956349611282349
Generation 248, Best Fitness: -0.6956349611282349
Generation 249, Best Fitness: -0.6956349611282349
Generation 250, Best Fitness: -0.6956349611282349
Generation 251, Best Fitness: -0.6956349611282349
Generation 252, Best Fitness: -0.6956349611282349
Generation 253, Best Fitness: -0.6956349611282349
Generation 254, Best Fitness: -0.6956349611282349
Generation 255, Best Fitness: -0.6956349611282349
Generation 256, Best Fitness: -0.6956349611282349
Confusion Matrix:
[[  0 100]
 [  0 100]]
PS C:\Users\kevMi> 
```

After generation 10 the results began to plateau. The confusion matrix also shows that it began to plateau to an optimal solution. It was unnecessary for 256 generations but it could be the data inputted.

Conclusion

The genetic algorithm successfully optimizes the weights of the neural network to achieve weight masking while maximizing fitness.

The resulting confusion matrix provides insights into the model's performance on the test data.

Further experimentation and tuning of hyperparameters may be required to improve performance or adapt the approach to different datasets and architectures.