Mawaba Pascal Dao

ML HW3

CSE5693

Dr. Chan

Apr 06 , 2022

1.a

A 3x2x1 Neural Network as 11 weights including the bias units' weights. We can represent the value of each weight as a binary string using the IEEE 754 format. Then a hypothesis would be represented as bit string containing 352 bits. That is, 11 concatenated 32 bit strings. As the size of any hypothesis of this type will have a fixed number of 352 bits. We can use the regular single-point, double point and uniform crossover techniques as well as single point mutation. However, single-point and double-point crossover are likely to cover the exponent part of IEEE 754 representation, thus producing offsprings that are vastly different from their parents. To ensure that the hypothesis search converges, it is desirable to avoid performing cross-over the exponent part of the 32-bit binary representation. For this reason, uniform cross over modified to ommit bits 30 through 23 of each 32 bit segment is chosen as the crossover operator.

Because GA keeps a several high-performing hypotheses at each iteration, it is less likely than BackPropagation to get stuck at a local minimum. However, performing operations on several hypotheses instead of one comes at the cost of running time, making GA much slower than Backpropagation.

1.b

10.1

- $2^d - 1$ rules will be formed
- Each rule will posess $d - 1$ preconditions
- It is challenging to determine how many choices seq-covering will make, because seq-convering generates rules until a performance threshold is met.

Seq-covering is more prone to overfitting because decisions about rules are made on smaller subset of rules compared to ID3. seq-covering will output set of rules that have high accuracy on the data they cover but have low coverage (They do not cover many examples)

1.c

10.3

- Add User_constraints input parameter to Learn-One-Rule function `(LEARN-ONE-RULE(Target_attribute, Attributes, Examples, k, User_constraints))`
- Under the following line:

```
Allcronstraints <- the set of all constraints of the form (a = v), where a is a member of Attributes, and v is a value of a that occurs in the current set of Examples
```

add

```
for c in Allconstraints
    for u_c in User_constraints
        if c.attribute == u_c.attribute
            c = u_c
```

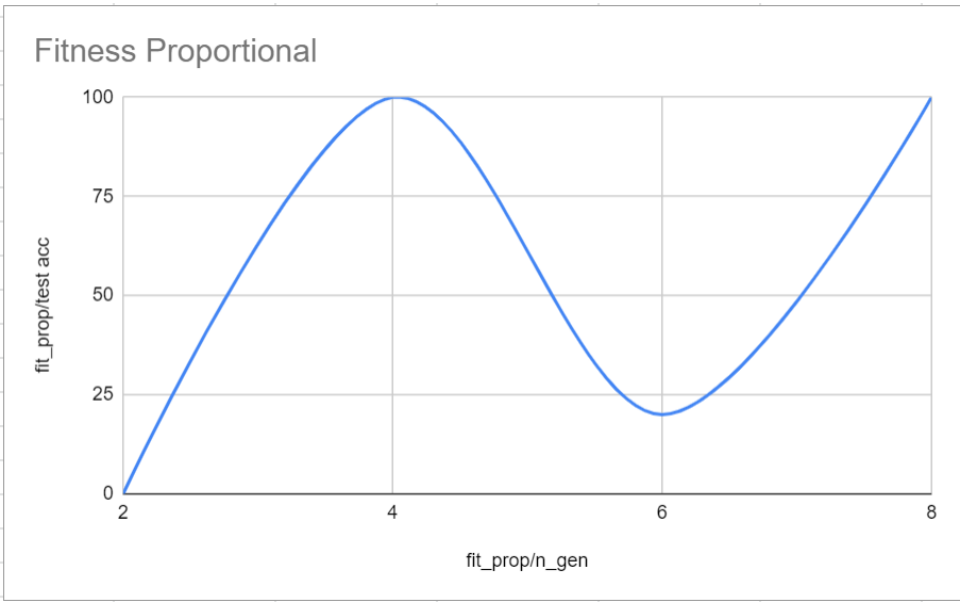
1.d

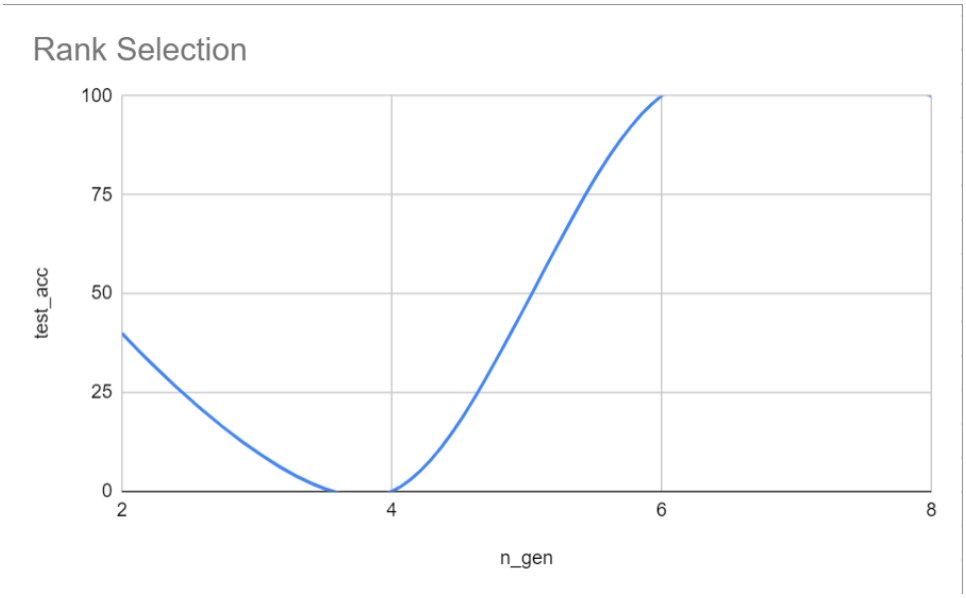
10.6

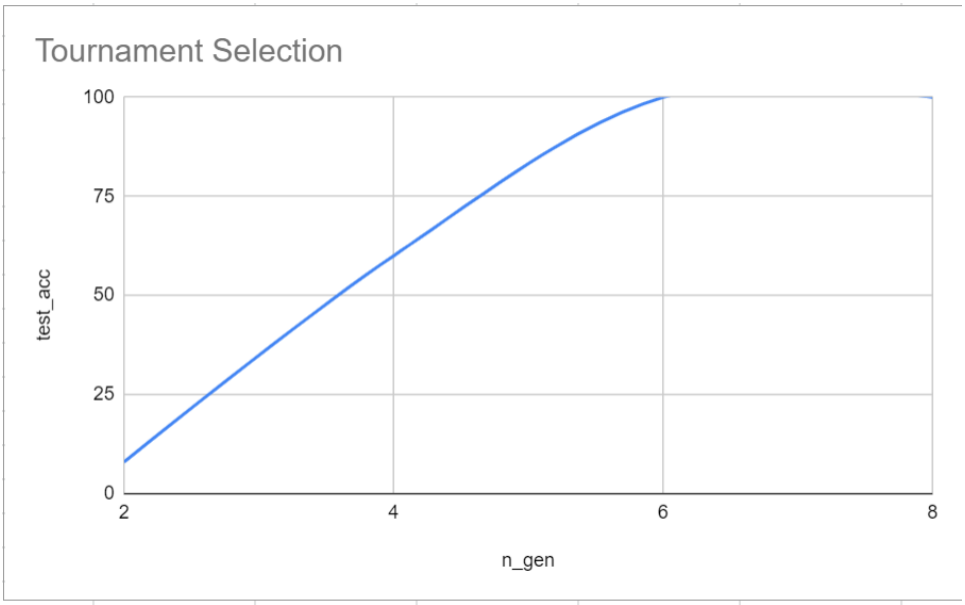
$C2 = P(x, A) V \neg S(B, y)$

subject to $A/x, B/x, A/y, A/x$

1.e

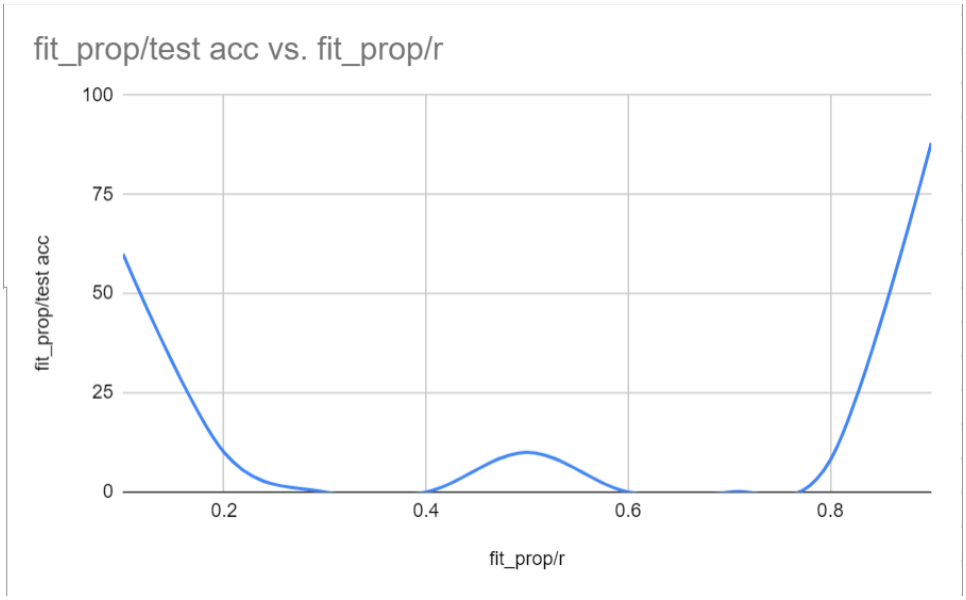


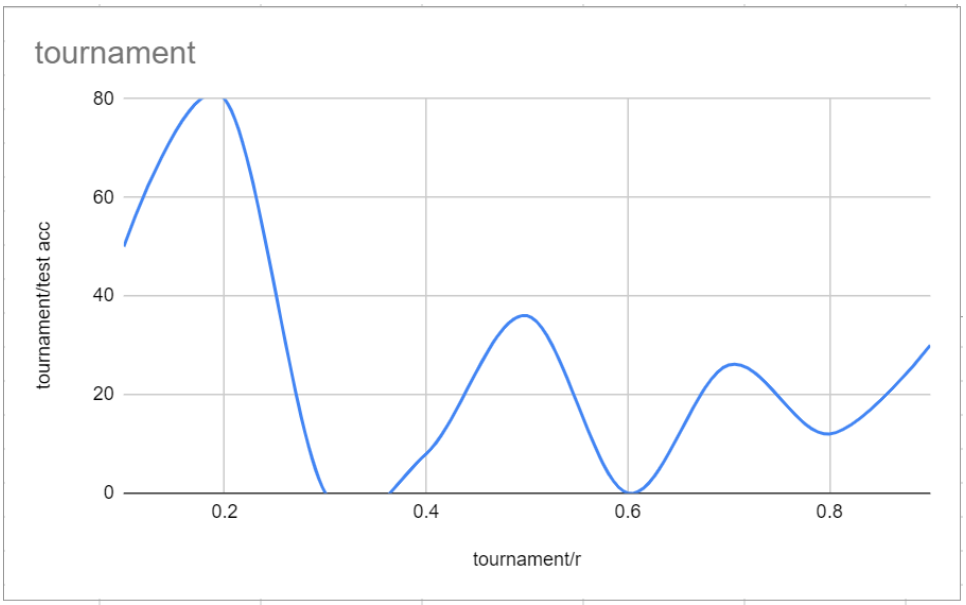


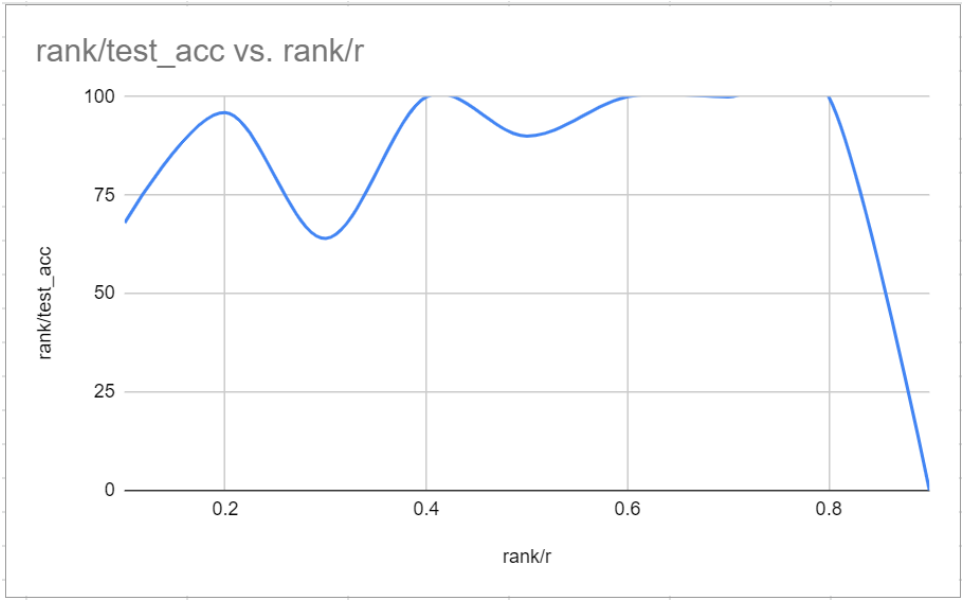


As shown in the plots above, increasing the number of generations increases the accuracy on the test set. This is expected as we are giving more time to the “evolutionary process” to find a progressively fitter individual. Of the 3 selection methods, tournament selection showcases the best improvement over a number of generations. As tournament selection selects the fittest of 2 individuals with some pre-determined probability it provides a better guarantee of progressively improving population quality over time. At each selection round, the selected individual must be as good or better as its “opponent”. As the average fitness of the population increases over time, selected individuals must progressively be better than the average. In contrast, in rank selection, although individuals with a higher rank (based on fitness) are preferred there is no direct elimination process of the individuals with low rank. Such hypotheses may still be selected with low probability. Furthermore, rank selection does not incentivize progressively keeping the best relative to the current population, as such the best hypotheses in a generation may remain at the top for several successive generations. Finally, fitness proportional has an even weaker incentive to improve population quality over time. Hypotheses are only eliminated given their fitness and the replacement rate, a mediocre individual may remain in the population for several successive generations and allowed to spread its genes. As seen the in the figures above, fitness proportional requires more generations to improve performance.

1.f


fit_prop/test acc vs. fit_prop/r
100
75
50
25
0
fit_prop/test acc
0.2
0.4
0.6
0.8
fit_prop/r


tournament
80
60
40
20
0
tournament/test acc
0.2
0.4
0.6
0.8
tournament/r


rank/test_acc vs. rank/r
100
75
50
25
0
rank/test_acc
0.2
0.4
0.6
0.8
rank/r

The expected behavior is a decrease in test accuracy as the replacement rate increases. A higher replacement rate “kills” a high percentage of the population potentially eliminating high fitness individuals or individuals with the potential to have a high fitness. Fitness proportional and tournament follow this expected trend, although fitness proportional unexpectedly got a really good performance on the last run. Rank selection overcame this challenge better likely because it gives a disproportionately high chance of selection to high-rank individuals, thus better shielding such individuals from elimination. However, rank selection also suffers from the high replacement rate on the last run with a 90% replacement rate. The image above shows the program output. Note that these results were obtained over only 2 generations. This is due to the running time limitations of the algorithm. More accurate results can be obtained with a higher number of generations.

```
-----------------------------------------------------
REPLACEMENT RATE = 0.1
                          | train acc      | test acc
-----------------------------------------------------
FITNESS_PROPORTIONAL      | 62.0 %          | 60.0 %
TOURNAMENT_SELECTION      | 85.0 %          | 50.0 %
FITNESS_PROPORTIONAL      | 52.0 %          | 68.0 %
-----------------------------------------------------
REPLACEMENT RATE = 0.2
                          | train acc      | test acc
-----------------------------------------------------
FITNESS_PROPORTIONAL      | 15.0 %          | 10.0 %
TOURNAMENT_SELECTION      | 72.0 %          | 80.0 %
FITNESS_PROPORTIONAL      | 96.0 %          | 96.0 %
-----------------------------------------------------
REPLACEMENT RATE = 0.30000000000000004
                          | train acc      | test acc
-----------------------------------------------------
FITNESS_PROPORTIONAL      | 5.0 %          | 0.0 %
TOURNAMENT_SELECTION      | 6.0 %          | 0.0 %
FITNESS_PROPORTIONAL      | 69.0 %          | 64.0 %
-----------------------------------------------------
REPLACEMENT RATE = 0.4
                          | train acc      | test acc
-----------------------------------------------------
FITNESS_PROPORTIONAL      | 1.0 %          | 0.0 %
TOURNAMENT_SELECTION      | 8.0 %          | 8.0 %
FITNESS_PROPORTIONAL      | 93.0 %          | 100.0 %
-----------------------------------------------------
REPLACEMENT RATE = 0.5
                          | train acc      | test acc
-----------------------------------------------------
FITNESS_PROPORTIONAL      | 6.0 %          | 10.0 %
TOURNAMENT_SELECTION      | 33.0 %          | 36.0 %
FITNESS_PROPORTIONAL      | 100.0 %          | 90.0 %
-----------------------------------------------------
REPLACEMENT RATE = 0.6
                          | train acc      | test acc
-----------------------------------------------------
FITNESS_PROPORTIONAL      | 2.0 %          | 0.0 %
TOURNAMENT_SELECTION      | 4.0 %          | 0.0 %
FITNESS_PROPORTIONAL      | 100.0 %          | 100.0 %
-----------------------------------------------------
REPLACEMENT RATE = 0.7
                          | train acc      | test acc
-----------------------------------------------------
FITNESS_PROPORTIONAL      | 4.0 %          | 0.0 %
TOURNAMENT_SELECTION      | 22.0 %          | 26.0 %
FITNESS_PROPORTIONAL      | 100.0 %          | 100.0 %
-----------------------------------------------------
REPLACEMENT RATE = 0.7999999999999999
-----------------------------------------------------
REPLACEMENT RATE = 0.7999999999999999
                          | train acc      | test acc
-----------------------------------------------------
FITNESS_PROPORTIONAL      | 14.000000000000002 %          | 8.0 %
TOURNAMENT_SELECTION      | 12.0 %          | 12.0 %
FITNESS_PROPORTIONAL      | 100.0 %          | 100.0 %
-----------------------------------------------------
REPLACEMENT RATE = 0.8999999999999999
                          | train acc      | test acc
-----------------------------------------------------
FITNESS_PROPORTIONAL      | 100.0 %          | 88.0 %
TOURNAMENT_SELECTION      | 51.0 %          | 30.0 %
FITNESS_PROPORTIONAL      | 16.0 %          | 0.0 %
```