Phi Duong
4/9/2025
CSE 5693

Machine Learning Written Assignment 4

(a) Consider applying GAS to the task of finding an appropriate set of weights for an artificial neural network (in particular, a feedforward network identical to those trained by BACKPROPAGATION (Chapter 4)). Consider a 3 x 2 x 1 layered, feedforward network. Describe an encoding of network weights as a bit string and describe an appropriate set of crossover operators. Hint: Do not allow all possible crossover operations on bit strings. State one advantage and one disadvantage of using GAS in contrast to BACKPROPAGATION to train network weights.

The encoding network to apply GAS on an ANN we would represent a bit string with a 1 dimensional the list of weights from each node in the current layer to each downstream node, then move to the next layer.

Example: with a 1 layered network (1 input, 1 hidden, 1 output)

$[W[h_1 i_1], W[h_1, i_2], \ldots, W[h_1, i_n], W[h_2, i_1], \ldots, \ldots, \ldots, W[h_k, i_n], W[o_1, h_1], \ldots, \ldots, \ldots, W[o_k, h_n]]$

Where, w is the weights at the input layer 'i ' to the hidden layer 'h', or the hidden layer 'h' to the output layer 'o' using n as the current layer size and k as the number of weights of a node in the nest layer. For more layers we can represent leach layer in groups of weights moving from layer H(n) to layer H(n+1).

To encode crossover operations we should only allow the bit string to crossover groups of weights going into one downstream node, but not the weights in between going into the nodes. This is because these weights correlate with each other. They are part of the same net weight algorithm. Any small changes will disrupt the meaningful association with each layer upstream. Randomly changing these weights will ruin and logical association that the neural network created. This creates a limitation of how crossover can work.

One advantage of using GA for ANN vs backpropagation is the hypothesis space search. Will a population, we can represent different hypothesis in a single generation which allows the algorithm to search multiple areas of the space at once, while the backpropagation algorithm only looks at one hypothesis space at a time. Gas looking al multiple hypothesis reduces the local minimum problem. A disadvantage is the computation time required to search down a local minimum. The problem here is that backpropagation will increment slowly down increment towards a better hypothesis, while the GA does not try to learn from the training set and will randomly set its weights until it finds a better hypothesis. Therefore, the GA algorithm does not guarantee it approaches any minimum. In addition, the hypothesis space for ANN are very large compared to other algorithms meaning it will be harder for a GA algorithm to stumble upon a minimum. This means that the algorithm could potentially waste a lot of computational time compared to an ANN.

(b) Consider a sequential covering algorithm such as CN2 and a simultaneous covering algorithm such as ID3. Both algorithms are to be used to learn a target concept defined over instances represented by conjunctions of n boolean attributes. If ID3 learns a balanced decision tree of depth d, it will contain 2^d - 1 distinct decision nodes, and therefore will have made 2^d - 1 distinct choices while constructing its output hypothesis. How many rules will be formed if this tree is re-expressed as t a disjunctive set of rules? How many preconditions will each rule possess? How many distinct choices would a sequential covering algorithm have to make to learn this same set of rules? Which system do you suspect would be more prone to overfitting if both were given the same training data?

Given a balanced binary decision tree using ID3 with depth of d, we can say that the number of leaf nodes of the tree will be 2^{d-1}. Each leaf node will consist of a unique path to get to, which can be represented as a rule, Therefore the rule number will be at the number of leaf nodes (size(t) = 2^{d-1}). Since each level is consider a path choice, then the number of preconditions will be equivalent to the depth size d. a CN2 algorithm using the same data would take longer for it to learn the same rules sets because of the beam search. Based on the k number of rules to keep for the next layer using beam search, we will need to make k times more decisions for each depth searched, while the ID3 makes a single choice, it takes 2^d-1 choices for the ID3 while CN2 takes k(2^d-1) for 1 single rule to learn. In addition, ID3 uses the same parents for each path, while CN2 restarts its search 1 rule from the beginning. Therefore, the total number of decision for CN2 is $[k(2^d-1)]^d$. since CN2 searches through a lot more rules and rule combinations than ID3, ID3 should be more prone to overfitting. CN2 is able to generalize more from the training set which can reduce overfitting, while ID3 is very greedy since it only looks at one attribute once, which reduces the capability of it generalizing the training data.

(c) Refine the LEARN-ONE-RULE algorithm of Table 10.2 so that it can learn rules whose preconditions include constraints such as nationality E {Canadian, Brazilian}, where a discrete-valued attribute is allowed to take on any value in some specified set. Your modified program should explore the hypothesis space containing all such subsets. Specify your new algorithm as a set of editing changes to the algorithm of Table 10.2

Lets generalize the constraints section to included combinations for values for the attributes and include disjunctions between them, therefore we rewrite the all constraints algorithm section to

- All_constraints <- the set of all constraints and combination of constraints in the form (a = ($v_{i,}$...,v_n) , forall i in s, where s E C(a)), where a is a member of Attributes, and C(a) is a set s of all combinations of value v for attribute a where v E (values(a)), that occurs in the current set of Examples.

(d) Apply inverse resolution to the clauses C = R(B, x) v P(x, A) and CI = S(B, y) v R(z, x). Give at least four possible results for C2. Here A and B are constants, x and y are variables.

Using inverse resolution we can write it in 4 different ways

- C2 = -S(z, y) V R(z,x) V P(x, A)
- C2 = -S(B, y) V R(B,x) V P(x, A)
- C2 = -S(B, y) V -R(z,x) V R(B,x) V P(x, A)
- C2 = -S(z, y) V R(z,x) V P(x, u)

(e) From testIrisSelection in the programming assignment, compare the three selection strategies. Plot training and test set accuracy against number of generations and discuss your observations.

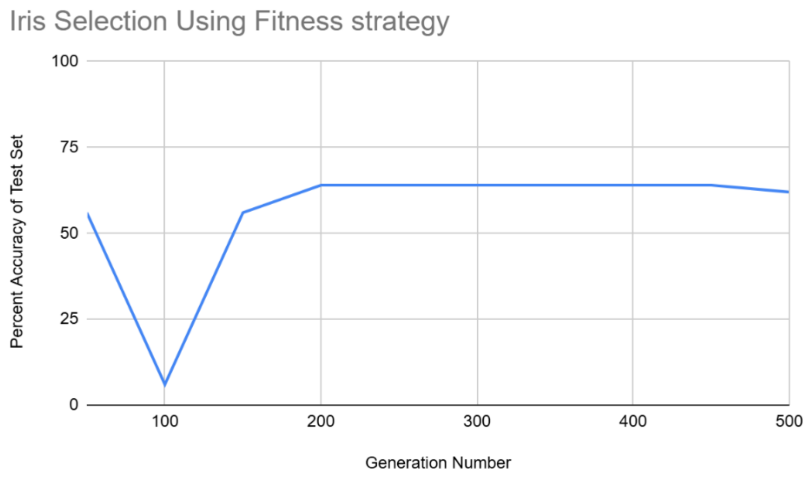

Table 1: iris Selection using fitness selection

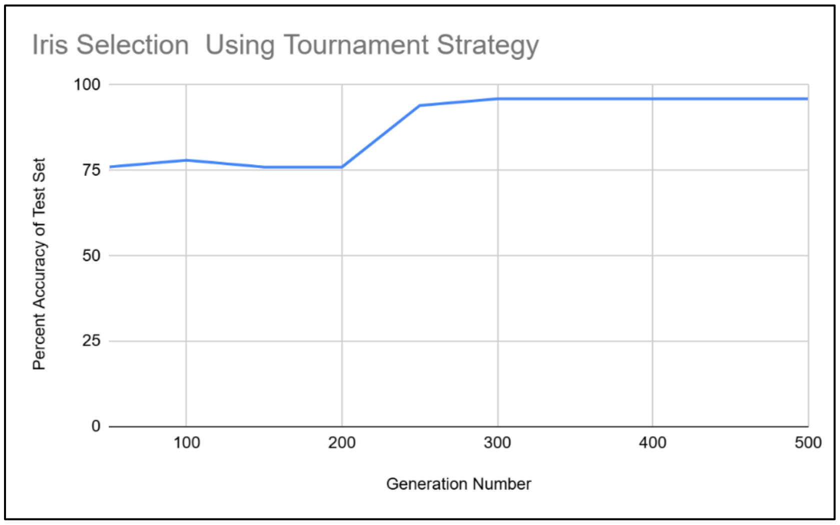

Table 2: iris Selection using tournament selection

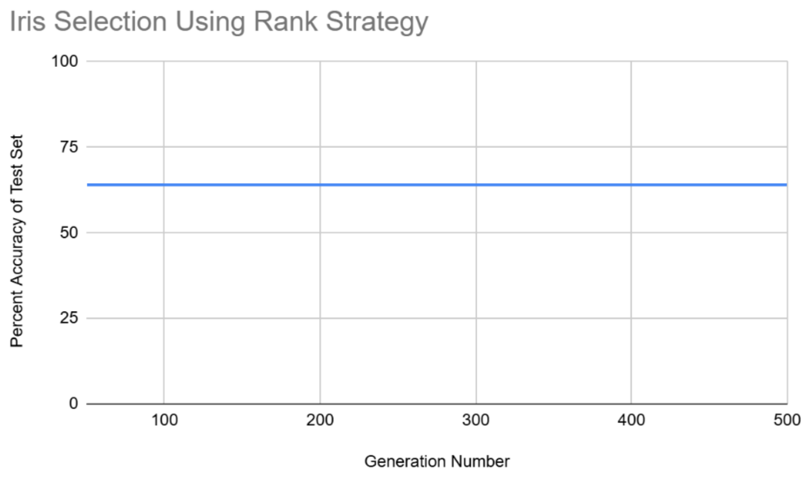

Table 3: iris Selction using rank selection

Overall it looks like tournament selection strategy work best with the training set. This is probably because ot was able to pass the local minimum problem by having the selection of the population staying more diverse than Rank selectiona and fitness selection. Rank selection did a slight better job at finiding the local minimum faster than fitness selection. This is probably due to Rank having a slight more diverse population than Fitness, but not enough to bypass the local minimum problem

(f) From testIrisReplacement in the programming assignment, plot test set accuracy against replacement rate (r) and discuss/explain your observations.

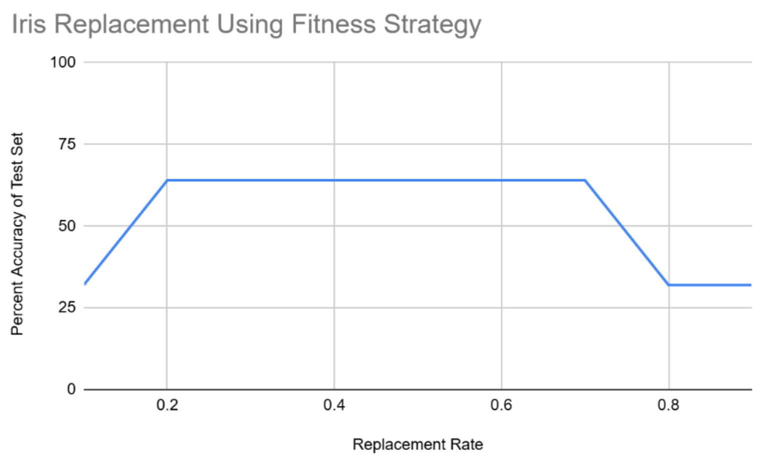

Table 4: iris Replacement using fitness selection

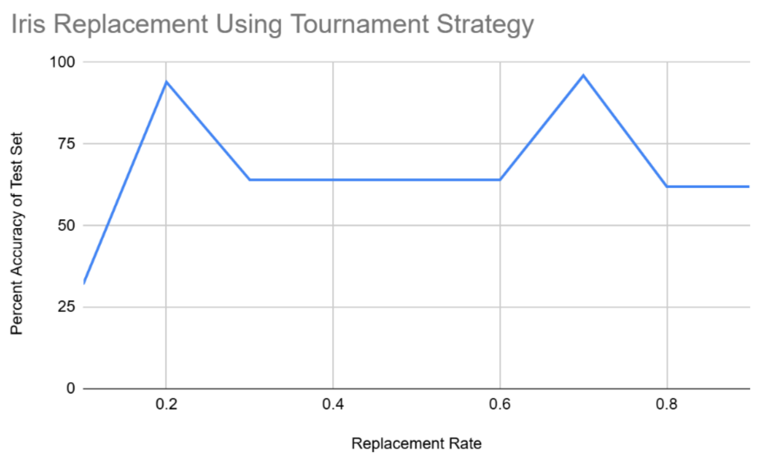

Table 5: iris Replacement using tournament selection

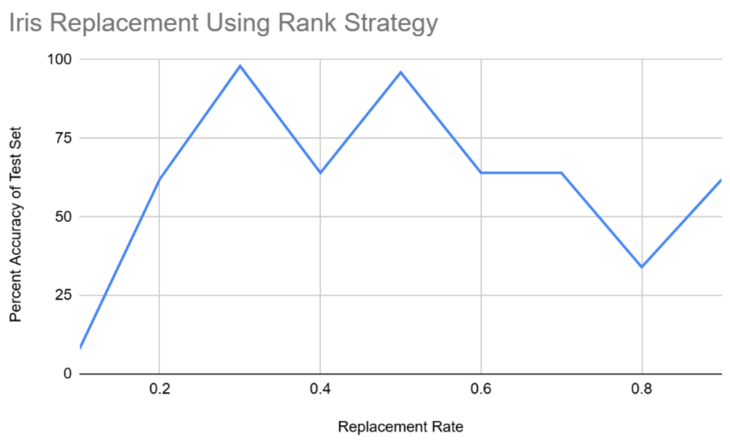

Table 6: iris Replacement using Rank selection

Overall, both tournament selection and rank selection straties were able to reach percentages of 90% accuracy or greater. This is probably due to these strategies doing better at reducing the local minimum problem. The fitness function unfortunately was unable to pass the local minimum problem and stayed at best 64%. It looks like all strategies do wrose at the extremes (0.1 and >0.80). This probably due to the low replacement rate not encouraging enough change or diversity, while the higher replacements rates increases chances of loosing the best individuals. Therefore, for this data set, it seems that a middle range replacement rate (around 0.5) does best for the training set.