MSSE 277B: Machine Learning Algorithms

Homework assignment #3: Meta-heuristic algorithms
Assigned Feb. 9 and Due Feb. 21

1. Genetic Algorithms. (11pt) Use GA to determine the maximum of the solution $f(x) = -x^2 + 8x + 15$ over the discrete range of x-values: [0,15], where $f(x)$ is the fitness function. Consider the following two possible encodings.

Encoding A					
Solution	**Fitness**	**Vector**	**Solution**	**Fitness**	**Vector**
0	15	1011	8	15	0100
1	22	0011	9	6	1100
2	27	1001	10	-5	0101
3	30	1000	11	-18	0110
4	31	0010	12	-33	0111
5	30	0001	13	-50	1101
6	27	0000	14	-69	1110
7	22	1010	15	-90	1111

Encoding B					
Solution	**Fitness**	**Vector**	**Solution**	**Fitness**	**Vector**
0	15	0001	8	15	1010
1	22	0010	9	6	0110
2	27	0100	10	-5	0111
3	30	1101	11	-18	0011
4	31	1011	12	-33	1000
5	30	1111	13	-50	1110
6	27	1001	14	-69	0000
7	22	1100	15	-90	0101

(a) (3pt) List all good solutions as those with $f(x) > 27$, and write out individual solutions for each encodings. Define schema by looking for perfect conservation along each column; if perfect conservation holds, give that value for that position, else represent that column with a *. What is length and order of the two schema? Which encoding will you choose?

(b) (2pt) Assume we draw candidate solutions $x = 10, x = 1, x = 15, x = 6, x = 0, x = 9$ as our initial population for GA optimization. Using your chosen encoding, list encoded solutions and their fitness. Pair the population so that the fittest members are paired with the least fit.

(c) (1pt) Use the cross-over operator by defining the cross-over point between the 1st and 2nd element. Exchange the last three elements between the pairs of strings. Do we have new solutions, and if so, what is their fitness? Have we increased fitness of population as a whole? What is the best solution?

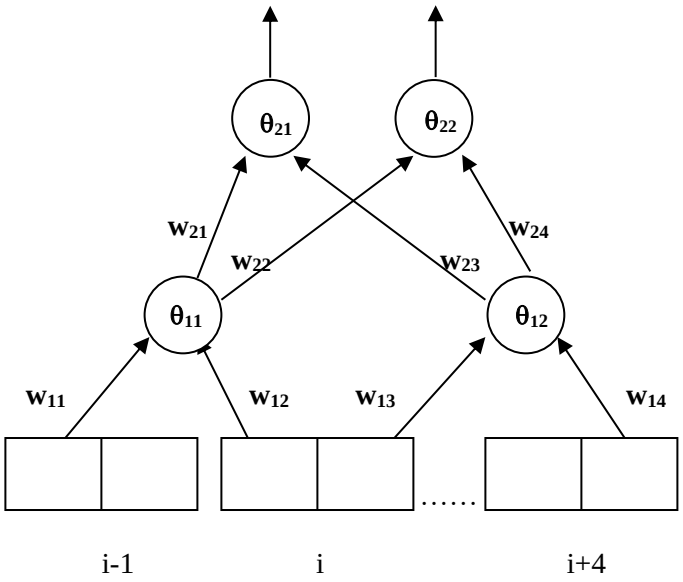
(d) (1pt) Using population generated in (c), mutate 3rd element. Do we have new solutions, and if so, what are their fitness? Does mutation increase fitness of population as a whole? Did we find a better solution?

(e) (1pt) Using population generated in (d), eliminate least fit member, and replace with cloned best member. Do 2-point cross-over by exchanging inner two elements between pairs. Do we have new solutions and what are their fitness? Have we increased fitness of population as a whole? Did we find a better solution?

(f) (1pt) Using population generated in (e), eliminate least fit member, and replace with cloned best member. Do cross-over between 3rd and 4th elements and exchange first three elements between pairs. Do we have new solutions, and what are their fitness? Have we increased the fitness of population as a whole? Did we find a better solution?

(g) (2pt) Do you think that the encoding of the solution space was adequate? Why?

2. Artificial Neural Networks. (9pt) In lecture we saw that ANN can be used for secondary structure prediction of alpha-helix, beta-sheet, or random coil given the amino acid sequence as input. Each amino acid is represented by two numbers, the first representing its propensity to be hydrophobic (+1) or hydrophilic (-1), while the second is its propensity to form helix (+1) or not helix (-1). A helix is predicted by the network if the output is (1,-1), β–sheet if output is (-1,1), and coil if (-1,-1). We could design two simple Boolean functions as part of a bigger network with the following connectivity:



Please code up your own little neural network using NumPy! If you choose to do it with code, to make life easier you can also treat this architecture as a fully connected network of shape (6,2,2).

(a) (1pt) Initialize the weights to random values between 0 and 1.0

(b) (3pt) Given the following input values for one pattern:

i-1=(-1,1) i=(-1,-1) i+4=(1,-1)

Feedforward through the above network and give the calculated output for this pattern and corresponding secondary structure definition (assume a hyperbolic tangent activation function, and its derivative is given as: $\frac{d \tanh x}{dx} = 1 - \tanh^2 x$).

(c) (2pt) The actual *observed* output for this one pattern is (-1,-1). Define the error and calculate it for all nodes that are not in the input layer.

(d) (3pt) Using back-propagation, give a formula for the weight update of all of the w_{ij}'s, and then calculate weight adjustments with a learning parameter.

Reference Reading: http://neuralnetworksanddeeplearning.com/chap2.html