Homework 4

October 24, 2019

Chapter 5

1) Exercises: #2
   Remarks:
   - Use $\eta = 0.1$.
2) Give it some thought: #1, #2
3) Computer Assignment: #1, #2
   Remarks.
   - In both problems, use two hidden layers and one output layer. The number of output neurons in the output layer is equal to the number of classes. The number of hidden neurons in each hidden layer is equal to the number of attributes.
   - Initial weights are randomly selected in the range $[-0.1, 0.1]$.
   - In problem 1, assume that the learning rate is 0.1. In problem 2, compare five learning rates $0.1, 0.2, 0.3, 0.4$, and 0.5.
   - Define convergence in problem 2 to be the condition that the absolute fraction of change in MSE is less than 10^{-4}, i.e
     $$\left| \frac{MSE(t+1) - MSE(t)}{MSE(t)} \right| \leq 10^{-4},$$
     for epoch t, where $MSE(t)$ is the mean-square-error defined in Eq. (5.3) for epoch t.
   - Use the data set in ILMS for problem 2.
   - Only C and Python are acceptable. No built-in functions on machine learning are allowed.

Due dates.

- Due date for the exercises and give-it-some-thought problems: 5 pm, Thursday, Oct. 31, 2019.
- Due date for the computer assignments: 5 pm, Thursday, Nov. 7, 2019.

Exercises

2. Hand-simulating backpropation of error as in the previous example, repeat the calculation for the following two cases:

   High output-layer weights: $w_{11}^{(1)} = 3.0, w_{12}^{(1)} = -3.0, w_{21}^{(1)} = 3.0, w_{22}^{(1)} = 3.0$
   Small output-layer weights: $w_{11}^{(1)} = 0.3, w_{12}^{(1)} = -0.3, w_{21}^{(1)} = 0.3, w_{22}^{(1)} = 0.3$

   Observe the relative changes in the weights in each case.

Give it Some Thought

1. How will you generalize the technique of backpropagation of error so that it can be used in a *multilayer perceptron* with more that one hidden layer?
2. Section 5.1 suggested that all attributes should be normalized here to the interval $[-1.0, 1.0]$. How will the network's classification and training be affected if the attributes are not so normalized? (hint: this has something to do with the sigmoid function)

Computer Assignments

1. Write a program that implements backpropagation of error for a predefined number of output and hidden neurons. Use a fixed learning rate, η.
2. Apply the program implemented in the previous task to some benchmark domains from the UCI repository.[3] Experiment with different values of η, and see how they affect the speed of convergence.