Due: November 6, 2018, 11:59 PM EST

Instructions

Your homework submission must cite any references used (including articles, books, code, websites, and personal communications). All solutions must be written in your own words, and you must program the algorithms yourself. If you do work with others, you must list the people you worked with. If you solve any problems by hand just digitize that page and submit it (make sure the problem is labeled).

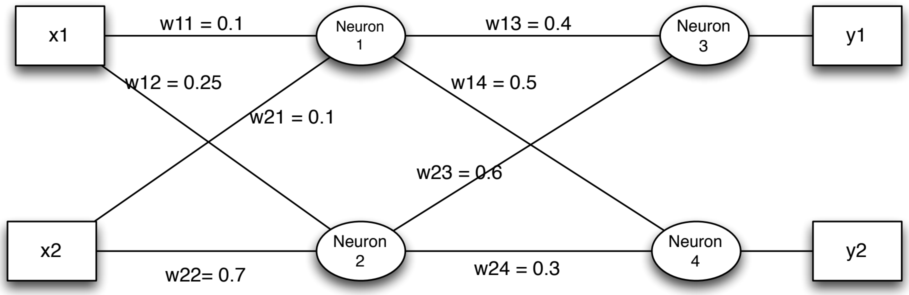
Your programs must be written in Python. All code must be able to compile and run for full credit. Comment all code following proper coding conventions. Remember, if we can't read it, we can't grade it! (For more information on python coding standards, refer to: *https://www.python.org/dev/peps/pep-0008/*)

You should submit your assignment via Github. Submit your solutions as a PDF named "hw(hw #).pdf". For example, homework 5 should be submitted as hw05.pdf. If the assignment requires coding, submit your working code as a .py file with the same name.

If you have any questions address them to:

- Xiaolei Guo (TA) – suninth@ufl.edu
- Connor McCurley (TA) – cmccurley@ufl.edu
- Daniel Wells (TA) – dwells@ufl.edu

1. Consider the Neural Network below.



All weights are initialized to the values shown (and there are no biases for simplicity). Consider the data point $\mathbf{x} = [1, 1]^T$ with desired output vector $\mathbf{d} = [1, 0]^T$ Complete one iteration of backpropagation by hand assuming a learning rate of $\eta = 0.1$. What would all the weight values be after the one backpropagation iteration? Show your work. Use the following activation function:

$$\sigma(v) = \frac{1}{1 + \exp\{-v\}} \tag{1}$$

2. Derive the update equation for output layer neurons if the activation function used is the hyperbolic tangent, $\phi(v) = \tanh(v)$ (instead of the activation function used in the notes). Show your work.

3. Derive the update equation for output layer neurons if the activation function used is the softmax function, $\phi(v) = \frac{e^{v_n}}{\sum_{i=1}^{O} e^{v_i}}$, where O is the number of output neurons and v_i is from the i^{th} neuron. Note: In this case the output is multi-dimensional (i.e., $\mathbf{d}_i \in \mathbb{R}^O$) and there will be a specific d_i for each output neuron. Show your work.