Assignment 4: Perceptrons and Neural Networks

CS486/686 – Spring 2015

Out: July 9, 2015
Due: July 24 (11:59 pm), 2015. Submit an electronic copy of your assignment via LEARN. Late assignments may be submitted within 24 hrs for 50% credit.

Be sure to include your name and student number with your assignment.

1. **[15 pts]** Prove the following properties of the logistic sigmoid function σ:
   - $\sigma(-a) = 1 - \sigma(a)$
   - $\sigma^{-1}(a) = ln(a/(1-a))$
   - $\frac{\partial \sigma}{\partial a} = \sigma(a)(1-\sigma(a))$
2. **[15 pts]** Consider a two-layer neural network of the form

   $$y_i(x, W) = \sigma(\sum_j W_{ji}^{(2)} g(\sum_k W_{kj}^{(1)} x_k + W_{0j}^{(1)}) + W_{0i}^{(2)})$$

   in which the hidden unit nonlinear activation functions $g(\cdot)$ are given by logistic sigmoid functions of the form

   $$\sigma(a) = \frac{1}{1+e^{-a}}.$$

   Show that there exists an equivalent network, which computes exactly the same function, but with hidden unit activation functions given by

   $$tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}.$$

   Hint: first find the relation between $\sigma(a)$ and $tanh(a)$ and then show that the parameters of the two neural networks differ by linear transformations.

[See the reverse for Questions 3 and 4.]

3. **[30 pts]** Implement the threshold perceptron learning algorithm. Initialize the weights to 0. Train and test the perceptron with the dataset posted on the course website.

   What to hand in:

   - Is the dataset linearly separable? Explain briefly.
   - Train and test accuracy of the threshold perceptron
   - A printout of the final weights of the threshold perceptron
   - A printout of your code

4. **[40 pts]** Implement a feed forward neural network with one layer of hidden nodes and a single output node. Use the sigmoid function as the activation function for all the hidden nodes and the output node. Train and test the neural network with the dataset posted on the course website. Since the output node uses a sigmoid activation function that returns a number between 0 and 1, associate any output greater than 0.5 to one class and any output less than or equal to 0.5 to the other class. Use the following parameters:

   - learning rate: 0.001
   - number of hidden nodes: 5 to 15
   - number of iterations in back propagation: 1000 (where each iteration consists of updating the weights based on all the instances in the training set)
   - weight initialization: random numbers in $[-0.5, 0.5]$

   What to hand in:

   - Graph of the train and test accuracy for your neural network as a function of the number of hidden nodes (from 5 to 15 hidden nodes)
   - A brief discussion of the results in the graph
   - Which algorithm performs best (the threshold perceptron or the neural network)? Why?
   - A printout of your code