COMP90051 Statistical Machine Learning

Semester 2, 2015

Workshop Week 7: Artificial Neural Networks

Questions

1. Now, use the Perception learning rule to adapt a single perceptron with two inputs for the Boolean function $y = x_1\ AND\ x_2$. Generate a random training set. Use the same initial weights and learning rate from the lecture notes example.
HINT Use a perceptron with two input nodes $x_1$and x_2, and one dummy input node x_0 constantly set to 1. Connect these nodes with the output node using the step transfer function. Training procedure is then analogous to the example shown in lectures (deck 11). When generating the training set, remember that each input and the output should be in $\{0, 1\}$, but the weights can be arbitrary real numbers.

2. Let $s = \sum_{i=0}^{n} w_i x_i$ be an input function, and $f(s)$ an activation function. Consider three possible choices for the activation function: identity $f(s) = s$, step $f(s) = 1$ if $s \geq 0$, and $f(s) = 0$ otherwise, and $f(s) = \frac{1}{1+e^{-s}}$. Discuss advantages and disadvantages of using each of these functions.
HINT In the context of perceptron: Identity function yields a model equivalent to linear regression. Step function has an inconvenient derivative, but can be used to model Boolean functions. Sigmoid function yields logistic regression, and has a nice derivative.

3. Consider a fully connected feed-forward artificial neural network with an input layer with two nodes (x_1 and x_2), a middle layer with two nodes (y_1 and y_2), and an output layer with a single node z. Assume that a sigmoid activation function is used in each node (y_1, y_2, and z).
(a) How many parameters does this model have? What are these parameters? Draw this network, and clearly label all nodes and weights on your diagram.
(b) Initialize the weights with random values. Show the value of all weights in the two-layer network after one training iteration with the back propagation training algorithm. Assume the learning rate is $\eta = 0.2$ and that there is only one training data example $(x_1, x_2, t) = (1, 1, 0)$.
(c) Consider a general three-layer network. Derive the update rules for the weights from the input to hidden layer.
HINT a) Each node will have 3 inputs/weights: 2 coming from the previous layer, and one coming from the dummy input node to allow a constant offset, hence 9 parameters in total; b) Use equations from lecture notes (deck 12); c) These are equivalent to the equations shown in lecture notes, but the exercise is to re-derive these without looking it up. A derivative of a generic activation function $f(s)$ can be simply left as $f'(s)$.