Department of Mathematics and Computer Science
University of Southern Denmark, Odense

February 13, 2013
Marco Chiarandini

DM825 - Introduction to Machine Learning

Sheet 5, Spring 2013

Exercise 1
Suppose you had a neural network with linear action functions. That is, for each unit the output is some constant c times the weighted sum of the inputs.

a.) Assume that the network has one hidden layer. For a given assignment to the weights $\mathbf{w}$, write down equations for the value of the units in the output layer as a function of $\mathbf{w}$ and the input layer $\mathbf{x}$, without any explicit mention of the output of the hidden layer. Show that there is a network with no hidden units that computes the same function.

b.) Repeat the calculation in part (a), but this time do it for a network with any number of hidden layers.

c.) Suppose a network with one hidden layer and linear activation functions has n input and output nodes and h hidden nodes. What effect does the transformation in part (a) to a network with not hidden layers have on the total number of weights? Discuss in particular the case $h \ll n$.

Exercise 2
Using the data from the previous exercises, investigate the use of `nnet` from the homonymous package in R to learn a neural network. The function provides an implementation to fit single-hidden-layer neural networks, possibly with skip-layer connections (i.e., a link from the input node directly to the output nodes). In particular, use that function for both regression and binary classification and become acquainted with the changes that they the two cases require. For multinomial classification try the function `multinom` from the same function. Start by looking at the examples. Finally, compare the performance against the previous linear regression models via `glm` and `mlogit`.