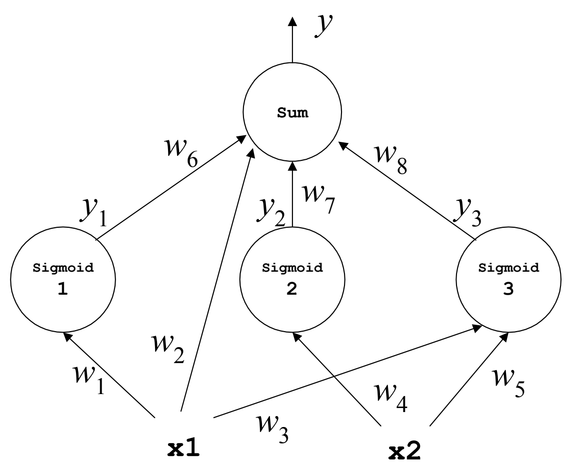
MSc Project Assignment:

Implementing Second Order Training Algorithms for MLP Neural Networks

Design and implement the second-order Newton's training algorithm for MLP neural networks. Present the algorithmic structure of the Newton's algorithm and the formulae for updating the network weights. Implement your own version of the algorithm in Matlab using the approximate Hessian matrix with second-order network derivatives with respect to the weights.

Test initially the Newton's algorithm with the simple neural network given below, and after that create another realistic fully connected network model with 10 inputs and 5 hidden nodes suitable for training with the Sunspots series. Compare the performance (in terms of normalized mean squared error) of the Netwon's algorithm using the approximate Hessian on the Sunspots series with: 1) the classical backpropagation algorithm (studied during the lectures); and 2) the Netwon's algorithm using the exact Hessian computed with the R-propagation algorithm (given in the books below). Give the approximation of the Sunspots series with these three algorithms on a common plot to illustrate the differences.



Assume that the initial weights and thresholds are:

$w_1 = -0.25$ $w_2 = 0.33$ $w_3 = 0.14$ $w_4 = -0.17$ $w_5 = 0.16$
$w_6 = 0.43$ $w_7 = 0.21$ $w_8 = -0.25$

References:

Bishop, C.M. (1995). *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, UK.
pp.150-160 Approximations of the Hessian matrix (R-propagation algorithm)
pp.253-290 Second-order training algorithms (in Chapter 7: Parameter Optimisation)

Nabney, I. (2002). Netlab: Algorithms for Pattern Recognition, Springer series Advances in Pattern Recognition.
pp.160-163 Fast Multiplication by the Hessian (R-propagation algorithm)

Haykin, Simon (1999). *Neural Networks. A Comprehensive Foundation.*,
Second Edition, Prentice-Hall, Inc., New Jersey.
pp.234-235 Newton's algorithm

Deadline: 31 March 2017.