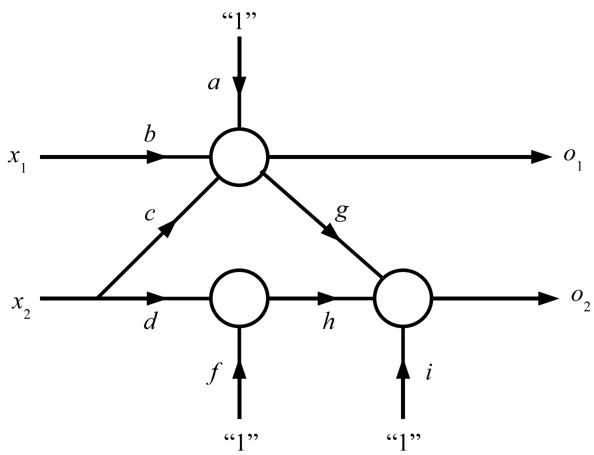
Multilayer perceptrons

Problem 1

Consider the following multilayer perceptron



The units of the first layer have as activation function (nonlinearity) the hyperbolic tangent. The unit of the second layer (which produces output y_2) is linear. The training set is

x_1	x_2	y_1	y_2
-1	-1	1	1
1	-1	2	-2

The initial values of all weights are 0.5. The cost function is the total squared error.

1. Draw the backpropagation network. Indicate, by means of appropriate symbols, the gains of all branches, and all the node values.

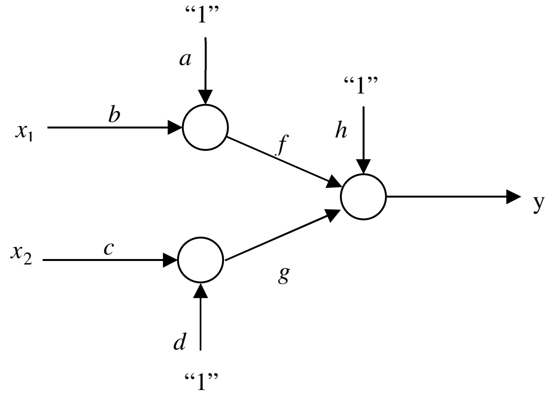
2. Compute the value of weight *c* after the first update, using backpropagation in **online mode**, assuming that the training set is repeatedly presented to the network with the training patterns in the order given above. The training is performed with a step size parameter $\eta = 0.1$ and with no momentum term.

3. Repeat item 2 using backpropagation in **batch mode**.

4. Repeat item 3 assuming that the cost function is the total squared error plus the regularization term known as weight decay, with $\lambda = 0.05$.

Problem 2

Consider a multilayer perceptron with the structure indicated in the figure.



All units have, as activation function, the logistic function:

$$g(s) = \frac{1}{1+e^{-s}}$$

The cost function is the total squared error. The training set has a single pattern:

x_1	x_2	y
1	1	1

Assume that all weights had an initial value of 1, and that, after the first iteration of optimization in **batch mode**, they all had a value of 0.5.

1. Draw the backpropagation network that corresponds to the network shown in the figure. Indicate, by means of appropriate symbols, the gains of all branches, and all the node values.
2. Find, through the backpropagation method, the value of weight *f* after one more iteration of optimization, with a step size coefficient $\eta = 0.5$ and with a momentum coefficient $\alpha = 0.7$.
3. Repeat item 2 above, but now using the total absolute error as cost function.