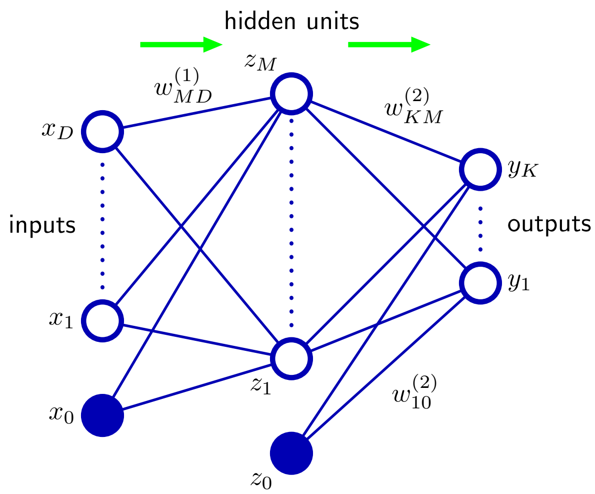
Task 1 - Forward propagation



Consider a two-layer network of the form illustrated above together with a sum-of-squares error, in which the output units have linear activation functions, so that $y_k = a_k$, while the hidden units have sigmoidal activation functions given by

$$h(a) = \tanh(a) \tag{1}$$

$$\text{with} \qquad \tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}} \tag{2}$$

$$\text{and} \qquad h'(a) = 1 - h(a)^2. \tag{3}$$

We denote the inputs to the hidden layer with a_j for $j = 1 \ldots M$.
We also consider a standard sum-of-squares error function, so that for datapoint n the error is given by

$$E_n = \frac{1}{2} \sum_{k=1}^{K} (y_k - t_k)^2 \tag{4}$$

where y_k is the activation of output unit k, and t_k is the corresponding target, for a particular input x_n.

Compute forward propagation, i.e. compute the activation of the neurons in the hidden layer and the activation of the output layer neurons.

a) $a_j =$

b) $y_k =$

Task 2 - Backpropagation

To perfom backpropoagation, we need to compute the derivative of the error function with respect to the corresponding weights. Compute the derivative wrt the first and second-layer weights. You therefore need to apply the chain rule.

(a) $\dfrac{\partial E_n}{\partial w_{kj}^{(2)}} =$

(b) $\dfrac{\partial E_n}{\partial w_{ji}^{(1)}} =$