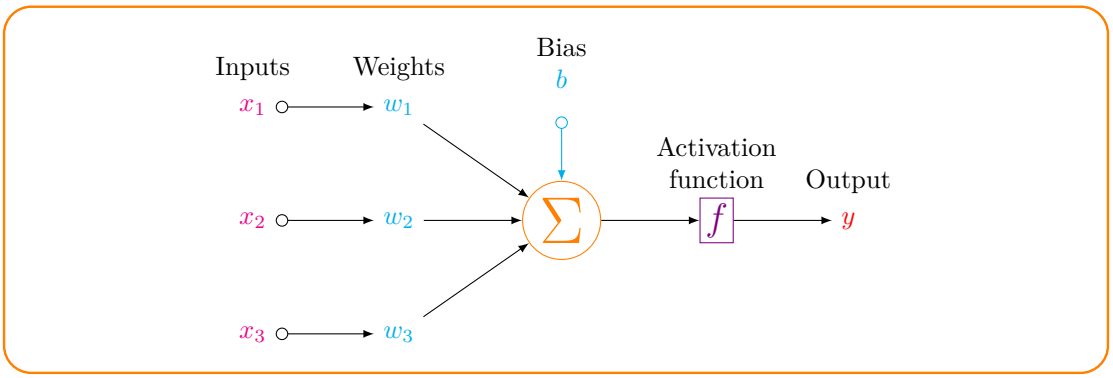
Submission until: **07.06.2015**
Discussion on: 09.06.2015
Submission as upload to your groups stud.IP folder as *groupNumber_sheet6.zip*

Assignment 1 *(PCA with whitening transformation (3p))*

We have already implemented PCA to reduce dimension of data for visualization. There is a preprocessing step called "whitening" to make the input data less redundant or to perform decorrelation transformation on the input. This leads to data with zero-mean and unit covariance. In PCA, we get the projection matrix P from eigenvectors with sorted descending eigenvalues V. For whitening PCA, the projection matrix can get from $P_w = \Lambda^{-1/2} V^T$ where Λ is a $N \times N$ diagonal matrix of eigenvalues λ_i, and V^T is an orthogonal matrix with columns given by v_i as in PCA.

Extend the PCA from ***assignment5.3a*** for iris dataset to compute the whitened data. Plot the data from PCA and PCA with whitening.

Assignment 2 *(Hebbian learning with Oja's rule (12p))*



(a) Load "dataLarge1.mat" and plot the data

(b) Choose a learning rate (it should be very small) and number of simulation steps

(c) Pick a step function as activation function with threshold $\theta = 0.5$

(d) Randomly initialize weights and plot the weight vector on the data (1p)

(e) For each iteration, the weights and output are calculated for each samples as explained in the lecture. We apply Oja's rule for the weight update using: $\triangle\vec{w} = \epsilon \cdot y \cdot (\vec{x} - y \cdot \vec{w})$ (5p)

(f) Update the plot of weight vector for each iteration. (2p)

(g) Insert the code to compute the correlation matrix C (1p)

Try running the code several times to see if starting with different initial weights will lead to convergence? Test the code again with another dataset, "dataLarge2.mat". Save the final plot with the weight vector of these two datasets to images (1p). Add Hebb's rule to the code to update the weight instead of Oja's rule. Discuss the result. (2p)