SCE301 Bio-computation Week 10 Computer Exercises 2016

PCA Neural Learning Models

Ex 1. Hebbian Learning Rule

Apply Hebbian learning rule for a linear neuron. The supplied datasets have dimension 2. The linear neuron computes the output

$$y = \sum_{i=1}^{2} w_i x_i.$$

The given Matlab code shall be completed by you. Those parts of the code which should be changed are marked by %----------. Make the following changes in the code:

- choose a learning rate .
- choose the number of simulation steps.
- insert code to compute the correlation matrix C
- insert code to compute the weight updates (Hebbs rule).

Apart from this, the code is complete. The dataset and the normalized weight vector is plotted.

The eigenvectors and eigenvalues of the correlation matrix C are computed and printed on the screen.

The program loads the dataset ue3_dp1.mat (first dataset). To analyse the second dataset, exchange the line load ue3_dp1.mat with load ue3_dp2.mat.

Does the weight vector converge to the eigenvector with the largest eigenvalue for dataset 1, 2? If not, why?
Does the weight vector maximize (approximately) die variance of the output on datasets 1, 2? Make plots to show that. If not, why?

Ex 2

Download the Wisconsin Data from the UC Irvine repository, extract PCAs from the data, test scatter plots of original data and after projecting onto the principal components, plot eigen values

Ex 3.

Practise with Sanger's generalized Hebb Learning: Try different initial weight vectors with different initial norms and observe its convergence. Does the learning

rule normalize the weight vector? Does it converge to the eigenvector of the correlation matrix with the largest eigenvalue?

Ex 4.

Implement Sanger's GHA neuron with random 2D input (image patches) from an image,

(a) Show the synaptic weight evolution. (1000 patterns at least)

(b) Calculate the correlation matrix of the input data.

(c) Find the eigen-values, eigen-vectors of this matrix. Compare to a.

(d) Repeat a for an Oja neuron, compare to b.