ARTIFICIAL NEURAL NETWORK

Chinmayananda A.
Department of Electronics and Communication Engineering

OUTLINE

- ***Hebbian-Based PCA***
  - Generalized Hebbian Algorithm and Heuristic Understanding
  - Convergence Considerations.
  - Optimality of the Generalized Hebbian Algorithm.

Chinmayananda A.
Department of Electronics and Communication Engineering

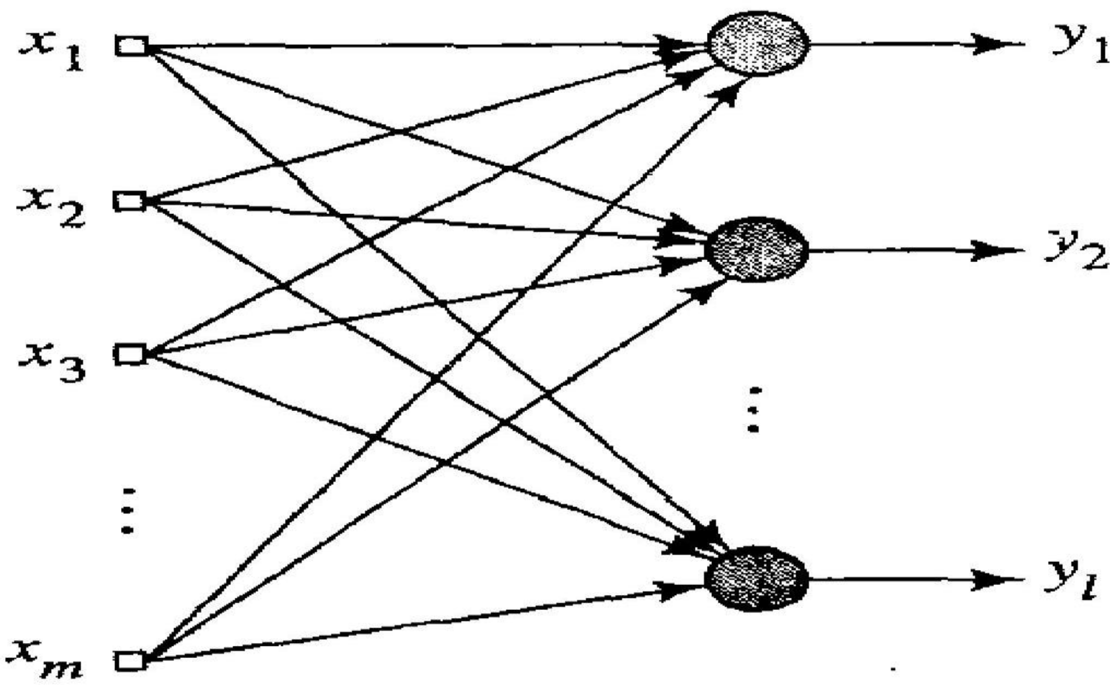
HEBBIAN-BASED PCA

Chinmayananda A.
Department of Electronics and Communication Engineering

HEBBIAN-BASED PCA

Introduction

- Single linear neuronal model can be expanded to a feedforward network with a single layer of linear neurons to perform PCA of inputs of arbitrary size. Assumption $l < m$.



HEBBIAN-BASED PCA

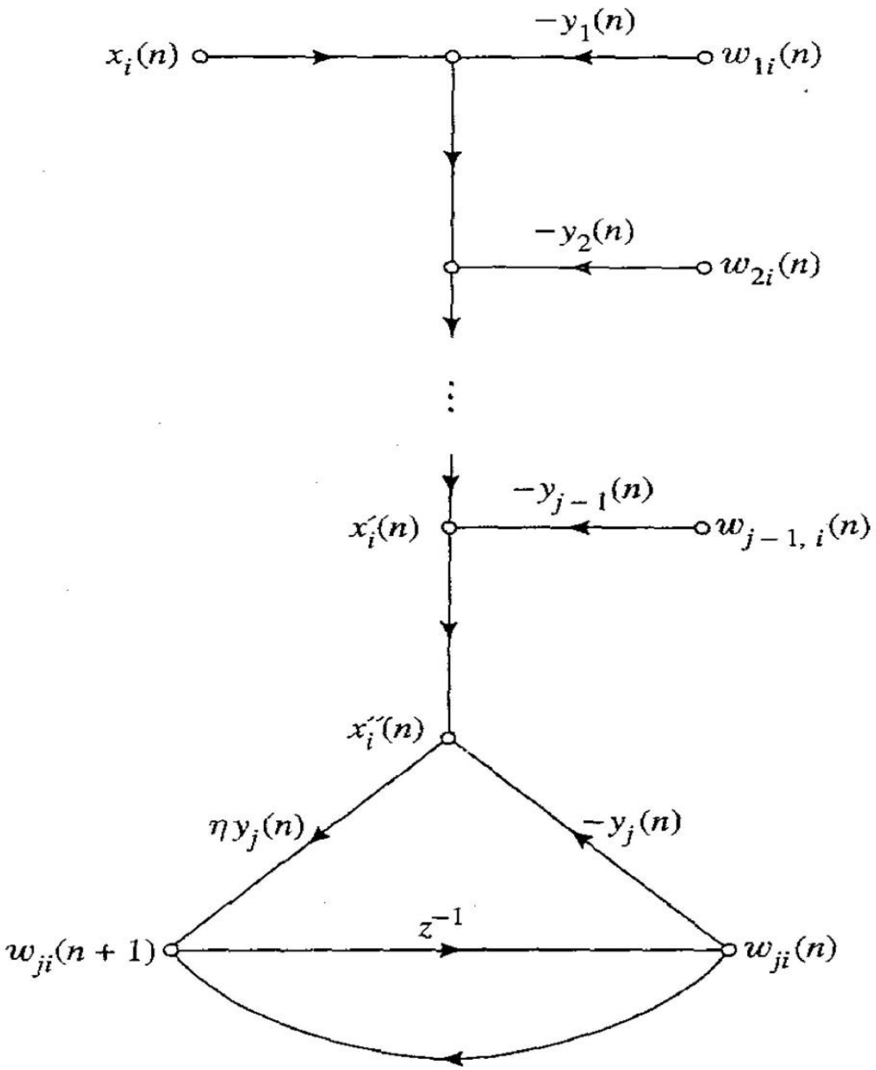
Generalized Hebbian Algorithm

- Output of jth neuron : $y_j(n) = \sum_{i=1}^{m} w_{ji}(n)x_i(n)\,, \quad j = 1,2,\ldots,l.$
- Generalized form of Hebbian learning rule :
  - $\Delta w_{ji}(n) = \eta\left[y_j(n)x_i(n) - y_j(n)\sum_{k=1}^{j} w_{ki}(n)y_k(n)\right], i = 1,2,\ldots,m.$
- Let $x_i'(n) = x_i(n) - \sum_{k=1}^{j-1} w_{ki}(n)y_k(n)$.
- $\Rightarrow \Delta w_{ji}(n) = \eta y_j(n)\left[x_i'(n) - w_{ji}(n)y_j(n)\right]$ resembles the weight update equation for Hebbian-based Maximum Eigenfilter.
- Let $x_i''(n) = \mathrm{x}_i'(\mathrm{n}) - \mathrm{w}_{ji}(\mathrm{n})\mathrm{y}_j(\mathrm{n}),\ \mathrm{w}_{ji}(\mathrm{n}+1) = \mathrm{w}_{ji}(\mathrm{n}) + \Delta\mathrm{w}_{ji}(\mathrm{n})$.
- Hebb's postulate : $\Delta w_{ji}(n) = \eta y_j(n)x_i''(n)$.

HEBBIAN-BASED PCA

Signal Flow Graph of Generalized Hebbian Algorithm

- $x_i'(n) = x_i(n) - \sum_{k=1}^{j-1} w_{ki}(n) y_k(n)$.
- $x_i''(n) = \mathrm{x}_\mathrm{i}'(\mathrm{n}) - \mathrm{w}_{\mathrm{ji}}(\mathrm{n})\mathrm{y}_\mathrm{j}(\mathrm{n})$.
- $\mathrm{w}_{\mathrm{ji}}(\mathrm{n}+1) = \mathrm{w}_{\mathrm{ji}}(\mathrm{n}) + \Delta \mathrm{w}_{\mathrm{ji}}(\mathrm{n})$.
- $\Delta w_{ji}(n) = \eta\left[y_j(n) x_i(n) - y_j(n) \sum_{k=1}^{j} w_{ki}(n) y_k(n)\right]$.



- Vectorization of Generalized Hebbian Algorithm
  - $\Delta w_{ji}(n) = \eta\left[y_j(n)x_i'(n) - y_j^2(n)w_{ji}(n)\right]$
  - $i = 1,2, \dots, m$ and $j = 1,2,3, \dots, l$.
  - Stacking all the above m equations for a given j, we get,
  - $\Delta w_j(n) = \eta y_j(n)x'(n) - \eta y_j^2(n)w_j(n), j = 1,2,3, \dots, l.$
  - $x'(n) = x(n) - \sum_{k=1}^{j-1} w_k(n)y_k(n)$.
- With $j = 1$, we see that $x'(n) = x(n)$, and weight update rule is same as given by Hebbian Maximum Eigenfilter.

HEBBIAN-BASED PCA
Heuristic Understanding

- With $j = 2, x'(n) = x(n) - w_1(n)y_1(n)$.
- If the first neuron has already converged to the first principal component, the second neuron sees an input vector $x'(n)$ from which the first eigenvector of the correlation matrix R has been removed.
- The second neuron extracts the first principal component of $x'(n)$, which is equivalent to the second principal component of the original input $x(n)$.
- Similarly, for $j = 3, x'(n) = x(n) - w_1(n)y_1(n) - w_2(n)y_2(n)$.

HEBBIAN-BASED PCA
Heuristic Understanding

- If the first two neurons have already converged to the first two principal components, the third neuron sees an input vector $x'(n)$ from which the first two eigenvectors of R are removed.
- The third neuron extracts the first principal component of $x'(n)$, which is equivalent to the third principal component of the original input $x(n)$.
- In general, we obtain l principal components of the input vector, at the output of the neural network in the decreasing order of eigenvalues.

HEBBIAN-BASED PCA

Convergence Considerations

- $W(n) = [w_1(n), w_2(n), \dots, w_l(n)]^T$ - $l \times m$ matrix .
- Let $\lim_{n \to \infty} \eta(n) = 0$ and $\sum_{n=0}^{\infty} \eta(n) = \infty$
- Vectorise the update rule $\Delta w_j(n) = \eta y_j(n) x'(n) - \eta y_j^2(n) w_j(n),\ j = 1,2,3, \dots, l$
- To write $\Delta W(n) = \eta(n)\{y(n)x^T(n) - LT[y(n)y^T(n)]W(n)\}$
- $LT[\,]$ – An operator which sets all the elements above the diagonal of its matrix argument to zero, such that the matrix becomes lower triangular.
- Under above assumptions, convergence of the GHA is proved by following a procedure similar to that presented for Maximum Eigenfilter.

HEBBIAN-BASED PCA

Theorem on Convergence (Sanger)

- Theorem : If the synaptic weight matrix W(n) is assigned random values at the time step $n = 0$, then with probability 1, the generalized Hebbian algorithm ($\Delta W(n) = \eta(n)\{y(n)x^T(n) - LT[y(n)y^T(n)]W(n)\}$) will converge to a fixed point $W^T(n)$ approaching a matrix whose columns are the first l eigenvectors of the $m \times m$ correlation matrix R of the $m \times 1$ input vector, ordered by decreased eigenvalue.
- This guarantees the algorithm to find the first l eigenvectors of the correlation matrix R.
- Most important : No need to calculate R. $\Rightarrow$ Computational savings if dimensionality m of the input is very large, and $l \ll m$.

HEBBIAN-BASED PCA

Optimality of the Generalized Hebbian Algorithm

- Let $\Delta w_j(n) \to 0, w_j(n) \to q_j$ as $n \to \infty$ for $j = 1, \ldots, l,$ and $\left|\left|w_j(n)\right|\right| = 1$.
- Then the limiting values $q_1, q_2, \ldots, q_l$ of the synaptic weight vectors of the neurons in the l-neuron feedforward network, represent normalized eigenvectors associated with $'l'$ dominant eigenvalues of the correlation matrix R, and ordered in descending eigenvalue.
- At equilibrium, we can write $Q^T R Q = \Lambda$.
- The output of j^{th} neuron has limiting value $\lim_{n\to\infty} y_j(n) = x^T(n) q_1$.
- Note that $\lim_{n\to\infty} E\left[Y_j(n) Y_k(n)\right] = E[q_j^T X(n) X^T(n) q_k] = q_j^T R q_k$
- $\Rightarrow$ At equilibrium, the generalized Hebbian algorithm is an eigen-analyser.

HEBBIAN-BASED PCA

Summary of the Algorithm

- Initialize the weights w_{ji} to small random values at $n = 1$, and a small positive value to the learning-rate parameter η.
- For $n = 1$, $j = 1,2, \dots, l$, and $i = 1,2, \dots, m$, compute
  - $y_j(n) = \sum_{i=1}^{m} w_{ji}(n) x_i(n)$
  - $\Delta w_{ji}(n) = \eta \left[y_j(n) x_i(n) - y_j(n) \sum_{k=1}^{j} w_{ki}(n) y_k(n) \right]$.
- Increment n and repeat the above step until the weights reach their steady-state value.

THANK YOU

Chinmayananda A.
Department of Electronics and Communication Engineering

chinmay@pes.edu

+91 8197254535