ARTIFICIAL NEURAL NETWORK

Chinmayananda A.
Department of Electronics and Communication Engineering

HEBBIAN-BASED MAXIMUM EIGENFILTER

Chinmayananda A.
Department of Electronics and Communication Engineering

OUTLINE

- ***Hebbian-based Maximum Eigenfilter***
  - Correspondence between Self-Organization and Principal Component Analysis.
  - Hebb's rule with Oja's Normalization.
  - Signal Flow Graph of Max. Eigenfilter.
  - Matrix Formulation.
  - Asymptotic Stability Theorem.

Chinmayananda A.
Department of Electronics and Communication Engineering

HEBBIAN-BASED MAXIMUM EIGENFILTER

Correspondence between Self-organization and PCA

- There is a close correspondence between the behavior of self-organized neural networks and statistical method of PCA.
- Key Result :
  - ``A single linear neuron with a Hebbian-type adaptation rule can evolve as a filter for the first principal component of the input data.''
- Linear neuron
  - Inputs : $x(n) = [x_1(n),\ x_2(n),\ \dots\ , x_m(n)]$
  - Weights : $\mathrm{w(n)} = [w_1(n),\ w_2(n),\ \dots\ , w_m(n)]$
  - Output : $y(n) = \sum_{i=1}^{m} w_i(n)x_i(n).$
  - Note that $y(n)$ is a scalar.

HEBBIAN-BASED MAXIMUM EIGENFILTER

Meaning of the Correspondence

- Key Result :
  - ``A single linear neuron with a Hebbian-type adaptation rule can evolve as a filter for the first principal component of the input data.''
- Hebb's postulate : $w_i(n+1) = w_i(n) + \eta y(n) x_i(n).\ i = 1,2,3, \ldots m.$
  - η – learning rate parameter.
- Meaning of the Key Result
  - View $x(n)$ and $y(n)$ as the input and output of a filter.
  - Let q_1 - eigenvector with unit length, corresponding to the maximum eigenvalue of the correlation matrix R.
  - Key Result $\Rightarrow y(n) = q_1^T x(n) = a_1 \rightarrow$ First Principal Component of $x(n)$.

HEBBIAN-BASED MAXIMUM EIGENFILTER

Problems with Hebb's Rule and Need for Normalization

- Hebb's postulate : $w_i(n+1) = w_i(n) + \eta y(n) x_i(n)$.
- Problems with Hebb's rule :
  - Above rule leads to unlimited growth of $w_i(n)$.
  - ⇒ Synapse is driven into saturation.
  - ⇒ Selectivity of the neuron is lost.
- Remedy :
  - Employ some form of normalization in Hebb's Rule.
  - ⇒ Competition among the synapses over limited resources.
  - Same as Principle 2 of Self-Organization and is required for stability.

HEBBIAN-BASED MAXIMUM EIGENFILTER

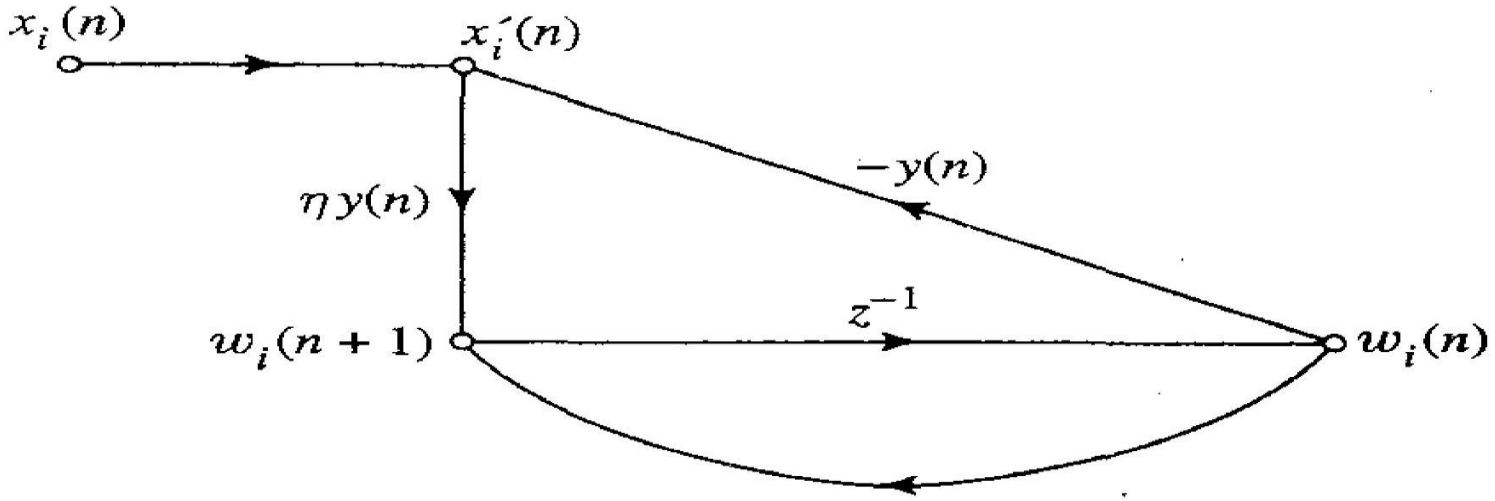
Oja's Normalization

- Normalization : $w_i(n+1) = \frac{w_i(n)+\eta y(n)x_i(n)}{\sqrt{\sum_{i=1}^{m}[w_i(n)+\eta y(n)x_i(n)]^2}}$.
- Hebb's original rule : $w_i(n+1) = w_i(n) + \eta y(n)x_i(n)$.
- Note that $\sum_{i=1}^{m} w_i(n+1)^2 = 1$, for any n.
- $\Rightarrow$ Competition among the synapses.
- Assuming η to be very small, and using power series
  - $w_i(n+1) = w_i(n) + \eta y(n)[x_i(n) - y(n)w_i(n)] + O(\eta^2)$.
  - $\Rightarrow w_i(n+1) = w_i(n) + \eta y(n)[x_i(n) - y(n)w_i(n)]$.
- $y(n)x_i(n)$ – self-amplification (Principle 1 of Self-organization).
- $-y(n)w_i(n)$ – stabilization (Principle 2 of Self-organization).

HEBBIAN-BASED MAXIMUM EIGENFILTER

Signal-flow Graph of Maximum Eigenfilter

- $w_i(n+1) = w_i(n) + \eta y(n)[x_i(n) - y(n)w_i(n)]$
- Effective input to i^{th} synapse : $x_i'(n) = x_i(n) - y(n)w_i(n)$.
- $\Rightarrow$ Learning rule : $w_i(n+1) = w_i(n) + \eta y(n)x_i'(n)$.



HEBBIAN-BASED MAXIMUM EIGENFILTER

Some Prerequisites and Related Jargon

- Difference Equation - an equation that shows the relationship between consecutive values of a sequence and the differences among them.
- Linear difference equation
  - $y(n) = a_1 y(n-1) + a_2 y(n-2) + \ \ldots + a_k y(n-k) + b.$
  - $a_i, i \in \{1,2, \ldots k\}$, and b are constants.
- Stochastic Approximation Algorithms - a class of iterative algorithms used for root-finding or optimization problems. Recursive update rules are used for solving linear systems when the collected data is corrupted by noise, or for approximating extreme values of functions which cannot be computed directly, but only estimated via noisy observations.

HEBBIAN-BASED MAXIMUM EIGENFILTER

Matrix Formulation of the Algorithm

- Input vector (random) : $x(n) = [x_1(n), x_2(n), \dots x_m(n)]^T$
- Synaptic weight vector (random) : $w(n) = [w_1(n), w_2(n), \dots w_m(n)]^T$
- Output : $y(n) = x^T(n)w(n) = w^T(n)x(n)$
- Weight update rule (Hebbian rule) for i^{th} synapse :
  - $w_i(n+1) = w_i(n) + \eta y(n)[x_i(n) - y(n)w_i(n)]$.
- Noting that η and $y(n)$ are scalars, and stacking the weight update rule for all synapses one above the other, we get the following :
  - $w(n+1) = w(n) + \eta y(n)[x(n) - y(n)w(n)]$.
  - $\Rightarrow w(n+1) = w(n) + \eta[x^T(n)w(n)x(n) - w^T(n)x(n)x^T(n)w(n)w(n)]$.
  - This is a Nonlinear Stochastic Difference Equation.

HEBBIAN-BASED MAXIMUM EIGENFILTER

Nonlinear Stochastic Difference Equation

- General nonlinear stochastic difference equation :
  - $w(n+1) = w(n) + \eta(n)h\big(w(n), x(n)\big),\ n = 0,1,2, \ldots \ldots$
  - h is a deterministic function with some regularity conditions.
  - Nonlinear - if $\eta(n)$ is not a constant or h is nonlinear in $w(n)$.
  - Stochastic - as $w(n)$ is a random vector.
  - Difference Equation - as it defines relation between $w(n)$ and $w(n+1)$.
- Weight update rule for all synapses (Nonlinear stochastic diff. equation) :
  - $w(n+1) = w(n) + \eta[x^T(n)w(n)x(n) - w^T(n)x(n)x^T(n)w(n)w(n)]$.
- Note : $h\big(w(n), x(n)\big) = [x^T(n)w(n)x(n) - w^T(n)x(n)x^T(n)w(n)w(n)]$.

HEBBIAN-BASED MAXIMUM EIGENFILTER

Ordinary Differential Equation and Weight Updates

- Weight update rule for all synapses :
  - $w(n+1) = w(n) + \eta[x^T(n)w(n)x(n) - w^T(n)x(n)x^T(n)w(n)w(n)]$.
- Note that $w^T(n)x(n)x^T(n)w(n)w(n)$ is not linear in $w(n)$.
- Nonlinear stochastic diff. equations - difficult to analyse convergence.
- Weight update equation defines a stochastic approximation algorithm.
- We describe a procedure to associate a deterministic ordinary differential equation (ODE) with a stochastic nonlinear difference equation .
  - Stability properties of the differential equation can be related to the convergence properties of the stochastic approx. algorithm.
- The procedure makes some assumptions on the weight update equation.

HEBBIAN-BASED MAXIMUM EIGENFILTER

Assumptions on the Stochastic Approximation Algorithm

1. $\eta(n)$-decreasing sequence of positive real numbers satisfying
   a. $\sum_{n=1}^{\infty} \eta(n) = \infty$ - necessary to move the estimate to a desired limit, regardless of initial conditions.
   b. $\sum_{n=1}^{\infty} \eta^p(n) < \infty$ for $p > 1$ – condition on how fast $\eta(n)$ must move towards 0.
   c. $\eta(n) \to 0$ as $n \to \infty$.
2. Sequence of vectors $w(n)$ is bounded with probability 1.
3. Update function $h(w, x)$ is continuously differentiable with respect to w and x, and its derivatives are bounded in time.

HEBBIAN-BASED MAXIMUM EIGENFILTER

Assumptions on the Stochastic Approximation Algorithm

4. Existence of the limit $\bar{h}(w) = \lim_{n\to\infty} E[h(w, X)]$ for each w.
5. Existence of a locally asymptotically stable solution (in Lyapunov sense) to the ODE $\frac{dw(t)}{dt} = \bar{h}(w(t))$.
6. Let q_1 be the solution to the above ODE with a basin of attraction $\mathcal{B}(q)$. Then $w(n)$ enters a compact subset $\mathcal{A}$ of the basin of attraction infinitely often, with probability 1.

HEBBIAN-BASED MAXIMUM EIGENFILTER

Asymptotic Stability Theorem

- ``For the given class of stochastic approximation algorithm (given by weight update equation), $\log_{n\to\infty} w(n) = q_1$ infinitely often with probability 1.'' (Assuming assumptions 1-6 hold)
- The procedure that we describe does not tell us how large 'n' (number of iterations) is required for the results of the analysis to be applicable.
- In tracking problems with time-varying parameter vector to be tracked, it is not feasible to have $\eta(n) \to 0$ as $n \to \infty$ (Assumption 1.c).
  - Assign a small value to η.

THANK YOU

Chinmayananda A.
Department of Electronics and Communication Engineering

chinmay@pes.edu

+91 8197254535