

HOMEWORK-4

PROBLEM 8.6:

The weight update according to the Hebb's postulate of learning is given by

$$w_i(n+1) = w_i(n) + \eta y(n) x_i(n) \quad i=1, 2, \dots, m \quad - \textcircled{1}$$

where, $n \rightarrow$ discrete time index,

$\eta \rightarrow$ learning rate parameter.

Using the above update rule, the weights w_i have unlimited growth which is not desirable.

To overcome this, we include normalization into Eq. 1 to get

$$w_i(n+1) = \frac{w_i(n) + \eta y_i(n) x_i(n)}{\left(\sum_{i=1}^m (w_i(n) + \eta y_i(n) x_i(n))^2 \right)^{1/2}} \quad - \textcircled{2}$$

Normalization as in Eq. 2 introduces competition among the synaptic weights over limited resources which in turn leads to stabilization.

Expanding the denominator of Eq. 2 for small η , we get

$$\begin{aligned} \left[\sum_{i=1}^m (w_i(n) + \eta y(n) x_i(n))^2 \right]^{1/2} &= \left[\sum_{i=1}^m (w_i^2(n) + 2\eta w_i(n) y(n) x_i(n)) \right]^{1/2} + O(\eta^2) \\ &= \left[\sum_{i=1}^m (w_i^2(n)) + \cancel{2\eta} 2\eta y(n) \sum_{i=1}^m w_i(n) x_i(n) \right]^{1/2} + O(\eta^2) \\ &= (1 + 2\eta y^2(n))^{1/2} + O(\eta^2) \\ &\approx 1 + \eta y^2(n) + O(\eta^2) \quad - \textcircled{3} \end{aligned}$$

Using Eq. 3 in Eq. 2, we get

$$\begin{aligned}w_i(n+1) &= \frac{w_i(n) + \eta y(n) x_i(n)}{1 + \eta y^2(n) + O(\eta^2)} \\&= [w_i(n) + \eta y(n) x_i(n)] [1 + \eta y^2(n) + O(\eta^2)]^{-1} \\&= [w_i(n) + \eta y(n) x_i(n)] [1 - \eta y^2(n)] + O(\eta^2) \\&= w_i(n) + \eta y(n) x_i(n) - \eta y^2(n) w_i(n) + O(\eta^2)\end{aligned}$$

Ignoring the second order terms, we get

$$w_i(n+1) = w_i(n) + \eta y(n) [x_i(n) - y(n) w_i(n)] \quad - \textcircled{4}$$

Comparing Eq. 1 and Eq. 4, we can see that the input is modified as

$$x_i'(n) = x_i(n) - y(n) w_i(n). \quad - \textcircled{5}$$

Effects of the normalization term:

- From Eq. 4, the term $y(n) x_i(n)$ represents Hebbian update to synaptic weights and hence accounts for self-amplification.
- In Eq. 5, the second term gives negative feedback which in turn results in stabilization.
- Using the positive and negative feedback as discussed above, the synaptic weights stabilize over time which avoids the unlimited growth of the synaptic weights.

The learning rule for minor-components analysis (MCA) is given by

$$w_i(n+1) = w_i(n) - \eta y(n) [x_i(n) - y(n) w_i(n)] \quad - \textcircled{6}$$

Writing the above equation in vectorized form we get

$$W[n+1] = W[n] - \eta X[n]^T [W[n] [X[n] - W^T[n] X[n] W[n]]] \quad - \textcircled{7}$$

Expanding $\textcircled{7}$ and considering $X(n)^T X(n) = R$ as the correlation matrix, we get

$$W(n+1) = W(n) - \eta [R W(n) - (W^T(n) R W(n)) W(n)]$$

Considering the incremental weight change $\Delta w(n)$ in discrete time to be proportional to the rate of change of weight $w(t)$ in the continuous time and absorbing the learning rate parameter η into proportionality factor, we get

$$\frac{dw(t)}{dt} = (W^T(t) R W^T(t)) W(t) - R W(t) \quad - \textcircled{8}$$

Expanding $w(t)$ using the orthonormal eigenvectors of the correlation matrix R , we get

$$W(t) = \sum_{k=1}^m \Theta_k(t) \underline{q}_k \quad - \textcircled{9}$$

where \underline{q}_k is the k^{th} normalized eigen vector and $\Theta_k(t)$ is the projection of $w(t)$ onto \underline{q}_k .

Substituting (9) in (8) and simplifying, we get

$$\frac{d\theta_k(t)}{dt} = \theta_k(t) \sum_{l=1}^m \lambda_l \theta_l^2(t) - \lambda_k \theta_k(t) \quad k=1, 2, \dots, m. \\ - (10)$$

Considering the two cases:

Case I: $1 \leq k < m$

Let $\alpha_k(t) = \frac{\theta_k(t)}{\theta_m(t)}$ for fixed m

$$\frac{d\alpha_k(t)}{dt} = \frac{1}{\theta_m(t)} \cdot \frac{d\theta_k(t)}{dt} - \frac{\theta_k(t)}{\theta_m^2(t)} \cdot \frac{d\theta_m(t)}{dt} \quad - (11)$$

Using (10) in (11) and simplifying, we get

$$\frac{d\alpha_k(t)}{dt} = \alpha_k(t) \left[-(\lambda_k - \lambda_m) \right] \quad \left(\lambda_k > \lambda_m \right) \\ - (12)$$

Therefore, $\alpha_k(t) \rightarrow 0$ as $t \rightarrow \infty$

Case II: $k=m$

$$\frac{d\theta_m(t)}{dt} = \theta_m(t) \sum_{l=1}^m \lambda_l \theta_l^2(t) - \lambda_m \theta_m(t)$$

Simplifying the above we get

$$\frac{d\theta_m(t)}{dt} = \lambda_m \theta_m(t) \left[\theta_m^2(t) - 1 \right] \quad - (13)$$

For the system to be stable, $\theta_m(t) = \pm 1$ as $t \rightarrow \infty$

Using the results of our analysis, we have

$$\Theta_m(t) = \pm 1 \quad \text{as } t \rightarrow \infty$$

$$\Theta_k(t) = 0 \quad \text{as } t \rightarrow \infty \quad \text{for } 1 \leq k < m.$$

• $W(t) \xrightarrow[t \rightarrow \infty]{} \underline{q}_m$ where \underline{q}_m is the normalized eigen vector corresponding to the smallest eigen value λ_m .

Therefore the sign change in the learning rule

leads to

$$\lim_{n \rightarrow \infty} W(n) = \eta \underline{q}_m$$

where \underline{q}_m is the eigenvector associated with λ_m .

PROBLEM 8.11

(a) Similarities between the symmetric algorithm and the generalized Hebbian-learning algorithm (GHA):

- Both the algorithms are based on Hebb's postulate of learning.
- Both the algorithms are generalizations of Oja's rule, i.e., the synaptic weights converge asymptotically to eigenvectors of the correlation matrix of the data.
- Both the algorithms have incorporated normalization within the learning rule to obtain stabilization of the growth of synaptic weights.

Differences between the symmetric algorithm and the GHA:

- In GHA, the modification of the input is in a sequential order. However the symmetric algorithm perform in parallel.

In turn, the symmetric algorithm may converge faster than the GHA.

(b) Oja's rule assumes a single neuronal model. Therefore the synaptic weight learnt based on the Hebbian rule converges to the eigen vector corresponding to the maximum eigen value.

- The symmetric algorithm has 'l' neuronal units such that the corresponding weights $\{w_j\}_{j=1}^l$ converge to the 'j' eigen vectors corresponding to the j^{th} largest eigen value.
- For $l_f = 1$, the symmetric algorithm degenerates to Oja's rule

$$w_j(n+1) = w_j(n) + \eta y_j [x(n) - w_j(n) y_j(n)]$$

Therefore, the principal subspace algorithm using the symmetric algorithm can be viewed as the generalization of the Oja's rule.