A RELATION BETWEEN HEBBIAN AND MSE LEARNING

Chuan Wang, Jyh-Ming Kuo, Jose C. Principe

Computational NeuroEngineering Lab. CSE 447
Electrical Engineering Department
University of Florida, Gainesville, FL 32611

ABSTRACT

Traditionally, adaptive learning systems are classified into two distinct paradigms---supervised and unsupervised learning. Although a lot of results have been published in these two learning paradigms, the relations between them have been seldom investigated. In this paper we focus on the relationship between the two kinds of learning and show that in a linear network the supervised learning with mean square error(MSE) criterion is equivalent to the basic anti-Hebbian learning rule when the desired signal is a zero mean random noise independent of the input. At least for this case there is a simple relationship between the two apparent different learning paradigms.

1. INTRODUCTION

During the past three decades there has been a considerable increase of interest in adaptive learning system. Many different approaches have been proposed for the design of engineering systems which exhibit adaptation and learning capabilities.

Generally speaking, most learning systems can be divided into one of two learning paradigms---supervised and unsupervised. It is accepted that the distinction between these two kinds of learning system resides on whether a teacher signal is used in learning. In learning with supervision, it is traditionally assumed that at each time instant we know in advance the desired response for the learning system, and we use the difference between the desired signal and the actual response to correct its behavior [1]. In the unsupervised learning framework, an internal adaptation constraint must be specified and the system does self-learning based on this underlying rule. It is generally accepted that the supervised and the unsupervised learning are totally different learning methods. But, we believe that the way constrains are placed in the optimization is really the fundamental difference between the two learning methods. When unsupervised learning is used the output of the net is not directly constrained, but in fact an implicit input output relationship is being specified. Nadal and Parga showed that the maximum information that can be stored in the weights adapted with supervised learning is equal to the maximum information that can be transmitted by a dual network learning with the unsupervised model [2]. In our paper, we study the relation between Hebbian learning and MSE learning, and we show that MSE learning defaults to anti-Hebbian when the desired signal is a zero mean random noise.

2. A RELATION BETWEEN SUPERVISED AND UNSUPERVISED LEARING

1.1. Unsupervised learning.

Unsupervised learning is depicted in Figure 1. The learning goal is not specified as an output response and learning is done based on some underlying rule.

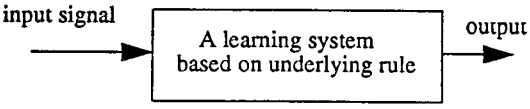

Figure 1 The unsupervised system

The most famous unsupervised learning rule is the so-called Hebbian rule [3], which adapts the learning system based on its input and output data vectors

$$\Delta w_{ij}(t) = \eta F(y_i(t) x_j(t)) \qquad (1)$$

where η is the step size, F(.,.) is a function of both postsynaptic and presynaptic activities, $x_i(t)$ is the input vector (presynaptic activities), $y_i(t)$ is the output vector (postsynaptic activities), $\Delta w_{ij}(t)$ is the learning network's weight increment vector, and t is the discrete-time variable which is assumed finite. In all the analysis in this paper, we assume that the input and the desired signal are finite sequences for simplicity.

As a special case of Eq. (1), we may write

$$\Delta w_{ij}(t) = \eta y_i(t) x_j(t) \qquad (2)$$

which is sometimes referred to as the activity product rule [3].

If a negative sign is put in the right side of (2), we get the anti-Hebbian rule which has the form

$$\Delta w_{ij}(t) = -\eta y_i(t) x_j(t) \qquad (3)$$

1.2. Supervised learning

In the supervised learning framework, learning is done on the basis of direct comparison of the system output with a known correct answer (desired signal), which can be represented as in Figure 2.

0-7803-2431-5/95 $4.00

Usually, the mean square error (MSE) is selected as the criterion because of its analytical simplicity and good properties.

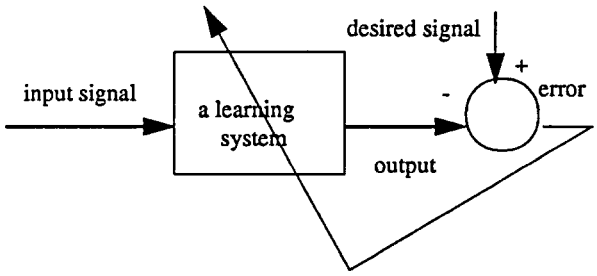

Figure 2 The supervised learning system

The MSE has the form:

$$\varepsilon = \frac{1}{2}\sum_i (d_i(t) - y_i(t))^2 \qquad (4)$$

where $y_i(t)$ is the output of the learning system, and $d_i(t)$ is the corresponding desired output.

Let's consider supervised learning with MSE in a linear network. The output of the network is

$$y_i(t) = w_{ij}(t)\, x_j(t) \qquad (5)$$

where w_{ij} is the weight connecting the input x_j to the output y_i.

The adaptation rule is

$$\Delta w_{ij}(t) = -\eta \nabla_{w_{ij}} \varepsilon = \eta (d_i(t) - y_i(t))\, x_j(t)$$

$$= -\eta y_i(t)\, x_j(t) + \eta d_i(t)\, x_j(t) \qquad (6)$$

where ∇ represents the gradient operator.

Assuming $x_i(t)$ and $d_i(t)$ are random sequences, expectation is needed in both sides of (6), hence, we get

$$E(\Delta w_{ij}(t)) = -\eta E(y_i(t)\, x_j(t) - d_i(t)\, x_j(t))$$
$$= -\eta E(y_i(t)\, x_j(t)) + \eta E(d_i(t)\, x_j(t)) \qquad (7)$$

Notice that the first term in the right hand side(RHS) of (7) is the anti-Hebbian rule given in (3) and the second term in the RHS of (7) is the forced Hebbian. In forced Hebbian, the output variable in Hebbian learning is substituted by the desired response. This term implements nothing but correlation learning. Therefore, in a linear network, learning with the MSE is equivalent to learning with the combination of the forced Hebbian and the anti-Hebbian rule.

1.3. Relationship between MSE and Hebbian learning

The traditional view is that in unsupervised learning the network output is decided by the underlying adaptation rule and the input data, and then cannot be specified in advance. But, Eq. (7) shows that there is an explicit relation between weight adaptation with MSE and a linear combination of two Hebbian like terms, involving both the input, output and desired response. So for this case we can prove the following:

Proposition 1: In a linear network the MSE learning defaults to anti-Hebbian learning when the desired signal is a zero mean random sequence independent of $x_i(t)$, or $d_i(t)$ and $x_i(t)$ are orthogonal.

In fact, the forced Hebbian in the right hand side of (7) becomes zero, and then

$$E(\Delta w_{ij}(t)) = -\eta E(y_i(t)\, x_j(t)) \qquad (8)$$

The proof is straightforward and is omitted. Comparing (8) with (3) which is the anti-Hebbian rule in the deterministic form, it is clear that they have the same form in a stochastic environment. Therefore, we conclude that when training with anti-Hebbian learning, the implicit desired signal in a MSE framework can be viewed as a zero mean random noise. Intuitively random noise should lead to some form of unsupervised learning since it does not provide any knowledge for learning.

Proposition 2:The anti-Hebbian rule is equivalent to the MSE when the desired signal is a zero mean random noise. In order to prove this proposition, we will use the minimization of the system output energy.

Proof: Eq. (4) can be written into

$$\varepsilon = \frac{1}{2}E\left(\sum_i (d_i(t) - y_i(t))^2\right) \qquad (9)$$

It is not difficult to show [4] that Eq. (9) is equal to

$$\varepsilon = \frac{1}{2}E\{\sum_i (y_i(t) - E\{(d_i(t))/(x(t))\})^2\}$$
$$+ E\{\sum_i \mathrm{var}((d_i(t))/(x(t)))\} \qquad (10)$$

When the desired signal d_i is a random noise with zero mean, $E\{d_i/x\} = 0$, the second term in Eq. (10) is independent of the weights, so minimizing Eq. (10) is equivalent to minimizing the output energy function

$$\varepsilon = \frac{1}{2}E\{\sum_i (y_i)^2\} \qquad (11)$$

Minimizing Eq. (11) in a linear network can be accomplished by a stochastic gradient-descent search with anti-

Hebbian rule[5].

$$w(t+1) = w(t) - \eta x(t) y(t) \qquad (12)$$

Eq. (12) is an implementation of the anti-Hebbian adaptation rule.

3. SIMULATION RESULTS

In order to verify the theoretic analysis given above, we provide some computer simulation results in this section. First of all, we analyze the anti-Hebbian learning (we do the experiments with anti-Hebbian rule instead of Hebbian rule due to the instability of Hebbian rule) for understanding better the equivalence between the supervised and the unsupervised learning.

3.1. Anti-Hebbian learning and its representation in signal space

The basic linear network for unsupervised learning is shown in Figure 3.

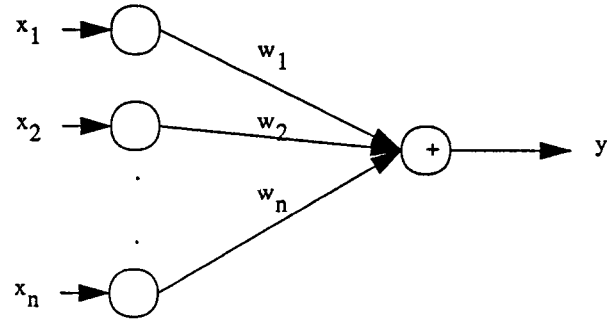

Figure3 A linear network with one output unit

Let $x = [x_1, x_2, \ldots, x_n]$ be the input vector, $w = [w_1, w_2, \ldots, w_n]$ be the weight vector, and the scalar y be the output.

The vector representation of anti-Hebbian learning of Eq. 3 has the form of

$$\Delta w(n) = -\eta y(n) x(n) \qquad (13)$$

And in a linear network, we know the output

$$y(n) = w(n) x(n)^T \qquad (14)$$

where T denotes the transpose operator.

Eq. (13) tells us that $\Delta w(n)$ is the outer product of the output and the input vector. When the output y is a scalar, each $\Delta w(n)$ is parallel to the input space. Hence, the cumulative weight vector increment $w_{increment}$ which is defined as the difference between the final weight and the initial weight vectors is also parallel to the input space. We can describe the relationship among vectors in the anti-Hebbian learning in a signal space as given in Figure 4,

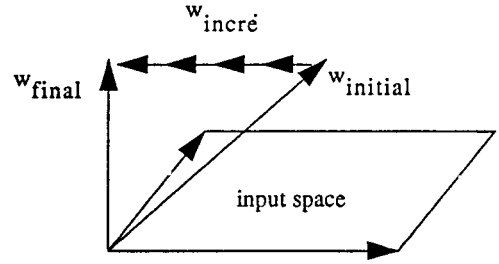

Figure 4 The signal space description of the anti-Hebbian learning

where $w_{initial}$ represents the initial weight vector, w_{final} is the weight vector after training, and $w_{increment}$ is the cumulative increment of the weight vector during adaptation.

The cumulative weight vector is a very important variable since it represents the new information the network learns from the training data set with a given learning rule.

It is very useful to notice in the anti-Hebbian learning that the $w_{increment}$ is parallel to the input space in each learning step and the final weight vector w_{final} is just the orthogonal projection of the initial weight vector onto orthogonal complement of the input space. Hence, the anti-Hebbian learning in a linear network is a process to find the projection of the initial weight vector onto the orthogonal complement of the input space.

3.2. Computer experimental results

The network used is depicted in Figure 3. The input signal is a sinusoid with frequency 1/50 Hz which is shown in Figure 5. The length of the segment data is 50. We do the experiment 20 times in order to see the statistical behavior with the same input signal and different initial weight vector which is set randomly. The cumulative of weight vector is shown in Figure 6 for the unsupervised anti-Hebbian learning and in Figure 7 for the supervised learning using zero mean random noise as the desired signal.

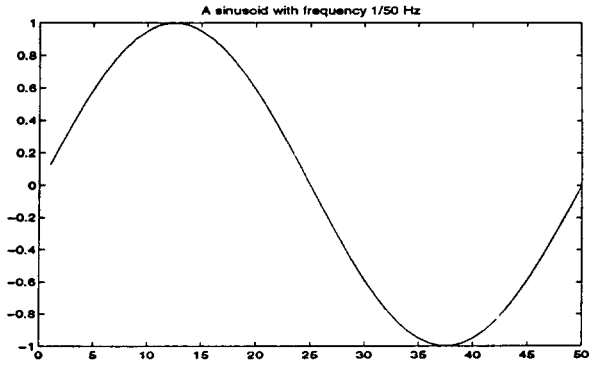

Figure 5 The input signal

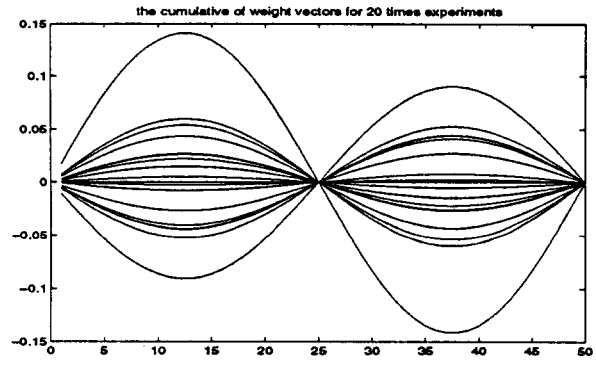

Figure 6 Results for unsupervised learning

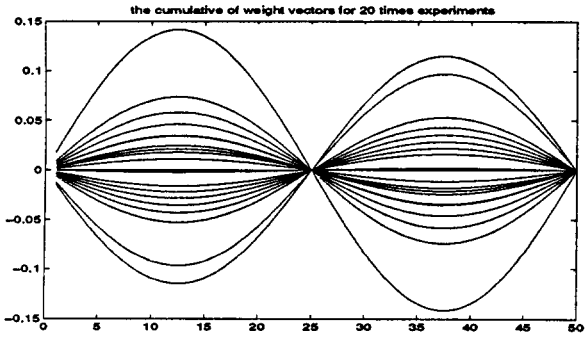

Figure 7 Results for supervised learning

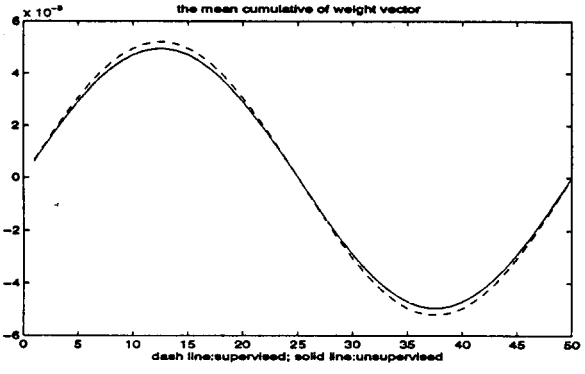

Figure 8 The mean of cumulative of weight vectors

The results given in Figure 6 and Figure 7 are two sinusoid-like waves and are parallel to the input signal in Figure 5 since they have the same phase.

The mean values of the cumulative of weight vectors for the 20 times experiments are given in Figure 8. It is obvious that from the statistical point of view the cumulative weight vectors are the same in both learning paradigms. Therefore, we can conclude from the simulation results that the supervised learning changes qualitatively into the unsupervised learning when the desired signal is a zero mean random noise.

4. CONCLUSION

We revealed a relationship between the supervised MSE learning paradigm and the anti-Hebbian unsupervised learning for a linear network. We showed that supervised MSE learning with a zero mean random noise desired signal defaults to anti-Hebbian learning. Hence, it is immaterial to train the network with either method.

There are some interesting implications of this work. First, there may be also equivalent supervised formulations for the other unsupervised paradigms (Sanger's, competitive, etc.). (We have already extended the results of this paper to nonlinear neural networks which will be published elsewhere). Second, this shows the fine relationship that exists between supervised and unsupervised learning schemes. Instead of being a dividing factor, more effort should be spent in trying to unify supervised and unsupervised learning paradigms. Third, the use of random noise as a desired signal was shown to be an efficient way to train neural networks for transient detection [6]. The random noise was used as the desired signal during the background. Now we can say why this is a reasonable choice. Effectively we were training the transient detector with a mixture of supervised (during the occurrence of the transient), and anti-Hebbian (during the background) schemes. Since anti-Hebbian minimizes the output norm, we were confining the net output to small values during background without explicitly using a lot of processing elements to impose this constraint. Moreover, this combination of supervised/unsupervised learning was attained using the same algorithm (MSE), which was very easy to implement.

REFERENCES

[1] Tsypkin, IA. Z, Adaptation and learning in automatic systems, New York, Academic Press 1971.

[2] Nadal J.-P., and Parga N., "Duality between learning machines: A bridge between supervised and unsupervised learning," Neural Computation, 6. 491-508(1994).

[3] Haykin, S, Neural Networks---A Comprehensive Foundation, Macmillan College Publishing Company, New York, 1994.

[4] Richard M., Lippmann R. P., "Neural network classifiers estimate Bayesian a posteriori probability," Neural Computation, 3, 461-483, 1991.

[5] Palmeri F., Zhu J., Chang C.,"Anti-Hebbian learning in topologically constrained linear nets", IEEE Trans. Neural Nets, vol 4, #5, 1993.

[6] Principe J., Zahalka A., "Transient detection using neural networks: the search for the desired signal", in Neural Information Processing Systems 5, (Hanson, Cowan, Giles),1993.