R. Brause
The Minimum Entropy Neuron- a new building block for clustering transformations
Int. Conf. on Artif. Neural Networks ICANN-92, Brighton, 1992

The Minimum Entropy Neuron - a new building block for clustering transformations

Rüdiger W. Brause

Computer Science Department, University of Frankfurt, Postbox 1119 23, 6000 Frankfurt 11, Germany

Abstract
One of the most interesting domains of feedforward networks is the processing of sensor signals. There do exist some networks which extract most of the information by implementing the maximum entropy principle for Gaussian sources. This is done by transforming input patterns to the base of eigenvectors of the input autocorrelation matrix with the biggest eigenvalues. The basic building block of such a transformation is the linear neuron, learning with the Oja learning rule.

Nevertheless, some researchers in pattern recognition theory claim that for pattern recognition and classification clustering transformations are needed which reduce the intra-class entropy. This leads to stable, reliable features and is implemented for Gaussian sources by a linear transformation using the eigenvectors with the *smallest* eigenvalues.

This paper states the problem and shows that the basic building block for this transformation can be implemented by a linear neuron using an Anti-Hebb rule and restricted weights. The fixpoints of the transformation are computed and the stability of the desired solution is shown. Thus, the networks containing this building block will first select the stable features for object recognition, in contrast to the traditional ones.

1. INTRODUCTION

For many purposes the necessary processing of sensor input signals is realized by using a system which implements the maximization of the transinformation from the input to the output of the system. For deterministic systems, this corresponds to the maximization of the output entropy (*maximum entropy* principle). In pattern recognition theory, it is well known that for Gaussian distributed sources this corresponds to the minimization of the mean square error of the output. For linear systems, this is done by a linear transformation to base of the eigenvectors of the autocorrelation matrix [FUK72]. Furthermore, we can compress (encode) the input information by using only those base vectors (eigenvectors) with the biggest eigenvalues. Neglecting the ones with the *smallest* eigenvalues results in the smallest reconstruction error of the encoded input [FUK72]. Generally, this approach can be used for sensor signal coding such as picture encoding, see e.g. [JAY84].

The neural network implementations of this approach use linear neurons, where each neural weight vector corresponds to one eigenvector. Examples of those architectures are the Oja subspace network [OJA89], the Sanger decomposition network [SAN89] and the lateral inhibition network of Rubner and Tavan [RUB89]. The last two mentioned networks decompose sequentially the input vector **x** in the learning process, see figure 1. They use as a basic building block the linear correlation neuron which learns the input

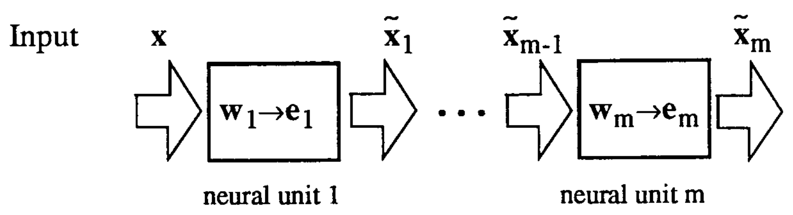

Figure1. The sequential construction of eigenvectors by the neural units

weights by an Hebb-rule, additionally restricting the weights $w_1,..,w_n$. As Oja showed [OJA82], this learning rule let the weight vector of the neuron converge to the eigenvector of the expected autocorrelation matrix C of the input patterns x with the biggest eigenvalue λ_{max}:

$$\mathbf{w} \rightarrow \mathbf{e}^k \quad \text{with} \quad \lambda_k = \max_i \lambda_i \qquad \text{and} \quad \mathbf{C}\mathbf{e}^i = \lambda_i \mathbf{e}^i, \qquad \tilde{\mathbf{x}}_i := \tilde{\mathbf{x}}_{i-1} - y_i \mathbf{w}_i \qquad \mathbf{C} := \langle \mathbf{x}\mathbf{x}^T \rangle$$

2. THE MINIMUM ENTROPY PRINCIPLE

The maximum entropy principle maximizes the entropy, i.e. for Gaussian sources it minimizes the quadratic error of the output coding. This aimes to minimize the reconstruction error for the input data from the encoded output.

In many applications, this is not the appropriate goal. If we want just to identify an object, we are not interested in the noisy representation of the object but in the code for the pure prototype of the object neglecting all variances. In the language of pattern recognition, all noisy instances of the object form a data point cloud (a cluster) around the prototype in the n-dimensional feature space. Here the goal of the transformation consists of projecting the cloud of data points onto the prototype. This is done by removing some uncertainty from the data points: the entropy of the cluster is reduced. It was shown by Tou and Gonzales [TOU74], that for Gaussian distributed clusters with uniform variance the cluster entropy is maximally reduced by the linear transformation on the basis of the eigenvectors of the input covariance matrix. Here the most reliable feature is given by the projection of the input to the eigenvector with the *smallest* eigenvalue; the eigenvectors with the biggest eigenvalues can be neglected. This necessity for clustering transformations motivates the question: Can we implement such a transformation also by a neural network ?

3. THE MINIMUM ENTROPY NEURON

The base of all three cited eigenvector decomposition networks consists of a neural unit learning the eigenvector with the biggest eigenvalue. In analyzing this approach, we can derive the proper learning rule for the eigenvector with the *smallest* eigenvalue and prove the stability of the solution.

Let us assume an input $\mathbf{x}=(x_1,..,x_n)$ for one neuron. Traditionally, the input is weighted by the weights $\mathbf{w}=(w_1,...,w_n)$ and summed up to the activation z of the neuron

$$z(t) = \Sigma_i\, w_i x_i = \mathbf{w}^T\mathbf{x}, \qquad \text{(and for the whole network } \mathbf{z} = \mathbf{W}\mathbf{x}) \qquad (3.1)$$

Since we assume linear neurons, with the linear output function S(z)=z the output y(t)=S(z) becomes z(t). The mean output variance $\langle (y-\bar{y})^2 \rangle$ is for centralized input $\bar{\mathbf{x}}:=\langle \mathbf{x} \rangle=\mathbf{0}$ (and therefore $\bar{y}:=\langle y \rangle=0$) equal to the output intensity

$$f(\mathbf{w}) = \langle (y-\bar{y})^2 \rangle = \langle y^2 \rangle = \langle \mathbf{w}^T\mathbf{x}\mathbf{x}^T\mathbf{w} \rangle = \mathbf{w}^T\mathbf{C}\mathbf{w} \qquad (3.2)$$

Since we are not interested in uniformly squeezing or expanding the pattern space, the volume should be conserved by the linear base transformation of (3.1). Thus, we assume det(**W**)=1 which is confirmed by the demand $|\mathbf{w}|=1$. This restriction of the weights is often used in learning systems to prevent the Hebbian learning rule from "blowing up" the weights.

Let us now investigate the necessary conditions for the local extrema of the objective function (3.2) with respect to the constrain $|\mathbf{w}|=1$. It is well known that the necessary conditions for the local extrema of the function with the Lagrange multiplier μ

$$L(w_1,\ldots,w_n,\mu) := f(\mathbf{w}) + \mu(|\mathbf{w}|^2 -1) \ = \mathbf{w}^T\mathbf{C}\mathbf{w} + \mu(\mathbf{w}^T\mathbf{w} -1) \qquad (3.3)$$

represent the desired conditions for the corresponding constrained function f(**w**).
It is easily shown (see [BR92]) that the necessary extremum conditions provide as *k* solutions with $\mu=-\lambda_k$ the eigenvectors $\mathbf{e}^k$ of the autocorrelation matrix **C**

$$\mathbf{w}^* = \mathbf{e}^k \qquad \text{with} \qquad \mathbf{C}\mathbf{e}^k = \lambda_k\mathbf{e}^k \qquad (3.4)$$

with the corresponding eigenvalues $\lambda_k=\langle y^2\rangle$. Unfortunately, the approach with Lagrangian multipliers does not determine what kind of extrema we do have. In [BR92], it is also shown by a different, more detailed approach that the fixpoint of the eigenvector with the maximal eigenvalue λ_{max} is an unique maximum, the eigenvector with the minimal eigenvalue λ_{min} an unique minimum. Beside these two fixpoints all other fixpoints are unstable saddle points. Thus, to reach the minimum we can use a simple gradient descend algorithm

$$\begin{aligned} &\bar{\mathbf{w}}(t+1) = \mathbf{w}(t) - \gamma \operatorname{grad} f(\mathbf{w}) = \mathbf{w}(t) - \gamma\mathbf{C}\mathbf{w}(t) && && (3.5) \\ \text{and}\quad &\mathbf{w}(t+1) = \bar{\mathbf{w}}(t+1) \,/\, |\bar{\mathbf{w}}(t+1)| && \textit{Normalization} && \end{aligned}$$

The stochastic version of this algorithm is with $\mathbf{C}\mathbf{w}=\langle\mathbf{x}\mathbf{x}^T\mathbf{w}\rangle=\langle\mathbf{x}y\rangle$

$$\begin{aligned} &\bar{\mathbf{w}}(t+1) = \mathbf{w}(t) - \gamma(t)\mathbf{x}(t)\, y(t) && \textit{Anti-Hebb-Rule} && (3.6) \\ \text{and}\quad &\mathbf{w}(t+1) = \bar{\mathbf{w}}(t+1) \,/\, |\bar{\mathbf{w}}(t+1)| && \textit{Normalization} && \end{aligned}$$

If the learning rate $\gamma(t)$ satisfies all the convenient conditions for the stochastic approximation process (e.g. $\gamma(t):=1/t$), the convergence of the approximation process is confirmed, see e.g. [OJA82]. If we replace the negative sign by the positive sign at (3.5) and (3.6), the gradient uphill climbing will provide us with the familiar Hebb-Rule for the maximal eigenvalue.

4. EXAMPLE

For the visualization of the convergence process we chose an example which is not too low-dimensional (and therefore trivial) and can also be shown satisfactory on a 2-dim sheet of paper. The 2-dim input pattern $\mathbf{x}^1=(1,1)$ and $\mathbf{x}^2=(0,1)$ have an autocorrelation matrix **C**. Analytically, we can compute the eigenvectors $\mathbf{e}^1=(0.851, 0.526)$, $\mathbf{e}^2=(-0.526, 0.851)$ and the eigenvalues $\lambda_1=1.309$, $\lambda_2=0.191$. In polar coordinates **w** is $\mathbf{w}=(w_1,w_2)^T= |\mathbf{w}|(\cos\alpha, \sin\alpha)^T$ with the constrain $|\mathbf{w}|^2=1$. Thus, $f(\mathbf{w})=\mathbf{w}^T\mathbf{C}\mathbf{w}= (\cos\alpha, \sin\alpha)\ \mathbf{C}\ (\cos\alpha, \sin\alpha)^T = 0.5 + 0.5\cos^2\alpha + \cos\alpha \sin\alpha$ with the maximum of $f(\mathbf{w}^*)=\lambda_1$ taken at $\alpha_1^*=0.553$ and $\alpha_1^*=3.69$, the minimum of $f(\mathbf{w}^*)=\lambda_2$ at $\alpha_2^*=2.12=\alpha_1^*+\pi/2$ and at $\alpha_2^*=5.26$.

The convergence process can be visualized by a needle-field picture. For the field of 20x20=400 possible values of **w** (small dots in figures 2 and 3) we plot the change in **w** by a small needle which is proportional to the length $|\Delta\mathbf{w}|=|\mathbf{w}(t+1)-\mathbf{w}(t)|$ and points in the direction of $\Delta\mathbf{w}$ of the deterministic algorithm (3.6).

The two stable fixpoints on the unit circle are the eigenvectors $\mathbf{e}^2$ and $-\mathbf{e}^2$ with the smallest eigenvalue λ_2. The two eigenvectors with the biggest eigenvalue λ_1 are unstable. If we use instead the maximum gradient search algorithm, the two stable fixpoints become unstable and the unstable ones with the biggest eigenvalue become stable (figure 3).

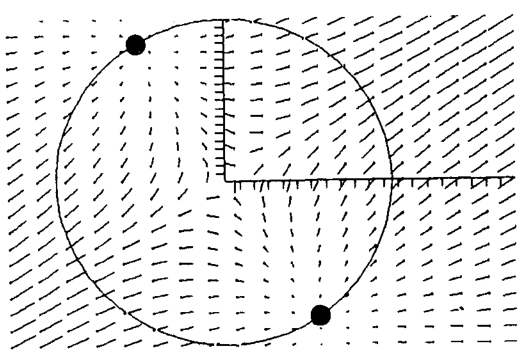
Figure 2. The minimum entropy fixpoints

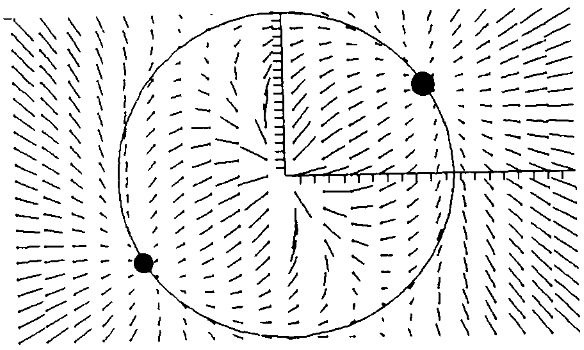
Figure 3. The maximum entropy fixpoints

5. DISCUSSION AND CONCLUSION

The paper showed how cluster transformation can be implemented by the base unit of a linear neuron where the weight vector converges to the eigenvector of the input pattern autocorrelation matrix with the smallest eigenvalue.

The base unit can be used in several ways, see [BR92]. The direct approach replaces the Oja-unit of the cited eigenvector decomposition networks. Thus, the sequential learning networks [SAN89], [RUB89] will first find the eigenvector with the *minimal* eigenvalue and subtract all its components from the input space, cf. figure 1. In the remaining space the second neuron will find the eigenvector with the *smallest* eigenvalue again which is the next one of the eigenvectors in ascending order of their eigenvalues.

It should be noted that the proposed mechanism involves only linear neurons. Additional non-linearities in the neural output function S(z) (squashing function) will lead to further reduction of the cluster entropy, but do not provide directly the eigenvector decomposition [OJA91]. In the binary version it becomes the vector quantization which can directly be used for symbolic postprocessing of an object recognition system.

6. REFERENCES

[BR92] R. Brause: The minimum entropy network; Fachbereich Informatik, University of Frankfurt, Internal Report 1/92

[FUK72] K.Fukunaga: Introduction to Statistical Pattern Recognition; Academic Press, New York 1972.

[JAY84] N.S. Jayant, Peter Noll: Digital Coding of waveforms, Prentice Hall 1984.

[OJA82] Erkki Oja: A Simplified Neuron Model as a Principal Component Analyzer J. Math. Biol. 13: 267-273 (1982)

[OJA89] Erkki Oja: Neural Networks, Principal Components, and subspaces Int. J. Neural Systems, Vol 1/1 pp. 61-68 (1989)

[OJA91] E. Oja: Learning in non-linear Constrained Hebbian Networks; Proc. ICANN91, T.Kohonen et al. (Eds.), Artif. Neural Netw., Elsevier Sc. Publ. 1991, pp. 385-390

[RUB89] J. Rubner, P. Tavan: A Self-Organizing Network for Principal-Component Analysis Europhys.Lett., 10(7), pp. 693-698 (1989).

[SAN89] Sanger: Optimal unsupervised Learning in a Single-Layer Linear Feedforward Neural Network; Neural Networks Vol 2, pp.459-473 (1989)

[TOU74] J.T. Tou, R.C. Gonzales: Pattern Recognition Principles; Addison-Wesley Publ. Comp., 1974