PCA, ICA, and Nonlinear Hebbian Learning

Erkki Oja

Helsinki University of Technology,
Rakentajanaukio 2C, 02150 Espoo, Finland
Email: `Erkki.Oja@hut.fi`

Abstract

Nonlinear extensions of one-unit and multi-unit Principal Component Analysis (PCA) neural networks, introduced earlier by the author, are reviewed. The networks and their nonlinear Hebbian learning rules are related to other signal expansions like the Projection Pursuit (PP) and the Independent Component Analysis (ICA).

1 Introduction

A Principal Component Analysis (PCA) network is a one-layer feedforward neural network which is able to extract the principal components of the stream of input vectors. Typically Hebbian type learning rules are used based on the one-unit learning algorithm originally proposed by the author in (Oja, 1982). Many different versions and extensions of this basic algorithm have been proposed during the recent years; for reviews and introductions, see e.g. (Oja, 1992; Haykin, 1994). PCA networks are useful in signal characterization, optimal feature extraction, and data compression. PCA is also the basis of subspace classifiers that have been used e.g. in speech and texture classification.

In the field of neural networks, there has been growing interest in nonlinear extensions of the PCA. The present author generalized the one-unit and multi-unit PCA learning rules to nonlinear versions in (Oja et al, 1991). Recently, some authors have discussed the relation of these rules to Projection Pursuit (Fyfe et al, 1994, sterberg, 1994). Another interesting development is the relation of neural networks to Independent Component Analysis (ICA) (Jutten and Herault, 1991); Karhunen et al (1995) proposed a multilayer network based on linear and nonlinear PCA layers as a neural realization of ICA.

These developments will be reviewed here. The starting point are the nonlinear constrained Hebbian rules introduced by the author in Oja et al (1991). These will be reviewed in Section 2 and related to optimization and Exploratory Projection Pursuit. In Section 3, the ICA expansion is reviewed and a two-layer structure is analyzed in which the first layer normalizes the input data and the second layer is the nonlinear PCA layer. This structure is closely related to both multidimensional PP and ICA. Section 4 is a review of the full ICA network.

2 Nonlinear Hebbian learning

2.1 Learning rule for one neuron

Consider first a single artificial neuron receiving an n-dimensional input vector x (Oja et al, 1991). The neuron is trying to adapt its weight vector w so that a function $E\{e(w^T x)\}$ is maximized, where E is the expectation with respect to the (unknown) density of x and $e(.)$ is a continuous objective function. In order to prevent the objective function from getting arbitrarily large values, the problem must be constrained by setting the Euclidean norm of w constant.

The only available information on x is the sequence of training samples $x_1, x_2, \ldots$ during a learning period. A *constrained gradient ascent learning rule* based on sample functions is then

$$\tilde{w}_{k+1} = w_k + \alpha_k \frac{d}{dw_k} e(w_k^T x_k), w_{k+1} = \frac{\tilde{w}_{k+1}}{\|\tilde{w}_{k+1}\|} \tag{1}$$

where α_k is the learning rate. The first equation is a pure gradient ascent step, and the other is a normalizing step. Denoting the derivative of $e(y)$ by $g(y)$, the gradient with respect to w becomes

$$\frac{d}{dw} e(w^T x) = g(w^T x)x. \tag{2}$$

For small values of α_k, eq. (1) can be approximated by

$$w_{k+1} = w_k + \alpha_k (I - w_k w_k^T) x_k g(w_k^T x_k) \tag{3}$$

in which terms proportional to α_k^2 have been dropped (Oja et al, 1991).

Depending on the function $e(w^T x)$, several cases are covered by this formalism. For instance, if the objective function is

$$e(y) = lncosh(y), \tag{4}$$

then $g(y) = (d/dy)e(y) = tanh(y)$, the activation function commonly used in Perceptron networks.

2.2 Connection to Exploratory Projection Pursuit

The Explatory Projection Pursuit (PP) technique was introduced by Friedman and Tukey (1974) for visualization purposes, especially clustering. The purpose is to find directions w' in the input space that are somehow interesting, for instance, such that projections $y = w'^T x$ of the data on the direction w' is multimodal. This is achieved by maximizing some suitable quality function. An especially useful quality function is the *fourth cumulant*, also called *kurtosis*, of the projection, defined as $E\{y^4\} - 3(E\{y^2\})^2$. In general high kurtosis signifies a peaked unimodal density, while low kurtosis (high negative value) indicates multimodality. Therefore a direction that has low kurtosis might be interesting for visualization and further processing (sterberg, 1994).

The constrained Hebbian learning rule of Section 2.1 was used by Fyfe et al (1994) in exploring neural learning for exploratory projection pursuit. In case $e(y)$ is a polynomial, the neuron weight will find the direction in the input space that will maximize a higher moment: for $g(y) = y^2$ and $e(y) = 1/3y^3$, the skewness $E\{y^3\}$ will be maximized[1]. For $g(y) = y^3$, the kurtosis $E\{y^4\}$ will be maximized.

As noted above, multimodality is revealed by a low kurtosis, and hence the objective function should rather be $e(y) = -y^4$. There is an interesting relationship between this and the criterion (4): expanding in Taylor series yields

$$lncosh(y) = 1/2y^2 - 1/12y^4 + 1/45y^6 + O(y^8), \tag{5}$$

hence the function that the neuron is maximizing is $E\{lncosh(y)\} = 1/2E\{y^2\} - 1/12E\{y^4\} + 1/45E\{y^6\} + E\{O(y^8)\} = 1/2E\{(w^Tx)^2\} - 1/12E\{(w^Tx)^4\} + 1/45E\{(w^Tx)^6\} + E\{O[(w^Tx)^8]\}$. The term $1/2E\{(w^Tx)^2\}$ is dominating in the expression especially for small values of $y = w^Tx$, and so the neuron with this activation function will have a tendency to learn the direction with maximal variance – i.e., the principal component of the input data. However, let us explicitly prevent this by *sphering* the data first (see Fyfe et al, 1994): by linear preprocessing, the covariance $C = E\{xx^T\}$ can be made equal to the unit matrix. Let us assume that the input vector has been sphered: $C = I$. Then the first term in the above expression becomes $1/2E\{(w^Tx)^2\} = 1/2w^TCw = 1/2w^Tw = 1/2$, since the weight vector is normalized to unit length. Now this term has no influence in the minimization and it is the second term, $-1/12E\{(w^Tx)^4\}$ that is dominating. Hence, kurtosis is minimized at least approximately.

Sudjianto and Hassoun (1994) made the interesting observation that the $tanh(\beta y)$ function, with a suitable parameter β, follows very closely the function $2\phi(y) - 1$ where $\phi(y)$ is the cumulative distribution function of the Gaussian density. Thus $z = tanh(w^Tx)$ will have an almost *uniform density* if x is Gaussian. In Projection Pursuit, such a direction should be avoided. Friedman's index in PP is based precisely on the deviation of z from the uniform density.

In the nonlinear Hebbian learning rule this same effect can now be obtained by using the objective function $e(w^Tx) = tanh^2(w^Tx)$, which would force the output of the neuron z close to its saturation values ± 1 - hence, make the density of z very non-uniform. This was one of the criteria proposed in Oja, 1991, and the derivative to be used in the learning rule (3) is then $g(y) = tanh(y)(1 - tanh^2(y))$.

The function $tanh^2(y)$ also has the effect of minimizing kurtosis because $tanh^2(y) = y^2 - 2/3y^4 + 34/90y^6 - ...$, and in the expectation the fourth order

[1] We will assume throughout that the inputs x are zero mean.

term will again dominate assuming the second order term is constant. However, the domination is not as strong as in the $lncosh(y)$ function.

2.3 Learning rules for several parallel neurons

A straightforward generalization of eq. (3) to the case of many identical parallel units is possible. Consider a one-layer neural network of k parallel units. Each unit i has the same n-element input vextor x and its own n-dimensional weight vector $w^{(i)}$, which together comprise an $n \times k$ *weight matrix* $W = (w^{(1)}...w^{(k)})$. Following Oja et al (1991), the following learning rules are generalizations of the linear PCA learning rule:

$$W_{k+1} - W_k = \alpha_k(I - W_k W_k^T)x_k g(x_k^T W_k), \tag{6}$$
$$W_{k+1} - W_k = \alpha_k[x_k g(x_k^T W_k) - W_k g(W_k^T x_k) g(x_k^T W_k)]. \tag{7}$$

The vector function $g(x_k^T W_k)$ must be understood element by element. It was shown by Karhunen and Joutsensalo (1994) that eq. (6) is a stochastic approximation algorithm for the minimization problems

$$J_1(w^{(i)}) = E\{e(x^T w^{(i)})\} + \Sigma_{j=1}^k \lambda_{ij}(w^{(i)T} w^{(j)} - \delta_{ij}),\ i = 1, ..., k, \tag{8}$$

where λ_{ij} are Lagrangian coefficients. Each neuron is maximizing its objective function under an orthonormality constraint. Likewise, eq. (7) approximates a stochastic approximation algorithm for the minimization problem

$$J_2(W) = E\{\|x - Wg(W^T x)\|^2\}. \tag{9}$$

This is a least squares criterion in which input vector x is represented by a nonlinear expansion of the weight vectors; for details, see Karhunen and Joutsensalo (1994).

3 Relation to Independent Component Analysis

A linear transformation of data such that the elements become statistically independent is called *Independent Component Analysis* (ICA) and has been proposed by Jutten and Herault (see their 1991 paper). For an overview, see Comon (1994). In the latter reference, a number of contrast functions are given whose maximization gives the independent components either exactly or approximately. Especially, the sum of squares of marginal cumulants of order 4 is suggested as a contrast function of particular interest. The fourth cumulant of a zero-mean random variable y is equivalent to the kurtosis widely used in Projection Pursuit (see Section 2.2). If all variances are equal and the cumulant is negative, then maximizing the absolute value of the fourth cumulant is again equivalent to minimizing the kurtosis, which brings an interesting connection between PP and ICA.

This suggests the following scheme: first, make a *sphering transformation* to the data vectors x such that all the elements will have unit variance and the covariances are zero. Then, solve the directions on which the *kurtosis is minimized.* This can be done approximately by using nonlinear Hebbian learning rules (6) or (7) with a $g(y) = tanh(y)$ function.

These algorithms could be further analyzed by writing down the averaged differential equations corresponding to eqs. (6) and (7). The limit of convergence is among the asymptotically stable solutions. Due to the nonlinearities, the analysis is not easy. Let us look at the fixed points only, concentrating on the learning rule (7): taking averages on the right hand side and equating it to zero yields

$$E\{xg(x^T W)\} = WE\{g(W^T x)g(x^T W)\} \tag{10}$$

where W is now a stationary solution of the averaged equation, thus a possible limit for the learning rule (7).

The elements of $W^T x = (w^{(1)T} x ... w^{(k)T} x)^T$ correspond to the outputs $y^{(i)} = w^{(i)T} x$ of the neurons. Substituting these in (10) yields for the p-th column of the matrix equation:

$$E\{xg(y^{(p)})\} = E\{\Sigma_{i=1}^k w^{(i)} g(y^{(i)}) g(y^{(p)})\}. \tag{11}$$

One possible solution is given by the condition

$$E\{g(y^{(i)})g(y^{(p)})\} = 0,\ i \neq p \tag{12}$$

which is satisfied for an odd nonlinear function $g(y)$ if $y^{(i)}$ and $y^{(p)}$ are *statistically independent*; note that both are zero mean if $g(y)$ is odd, i.e. $g(y) = -g(-y)$ for all y, and if the density of x is symmetrical and centered at zero mean.

I this case the matrix W that makes the outputs $y^{(i)} = w^{(i)T} x$ independent is also an asymptotically stable limit of the averaged differential equation under certain conditions depending on the function $g(y)$ and the density of x. Numerical simulations indicate that the stability is strong in the sense that this limit is attained from a wide range of initial values. A practical example taken from Karhunen et al (1995) is shown in the next Section.

4 Two-layer networks

Consider a two-layer network in which the first layer is linear and spheres the data. If the original n-element input vector is x and the output of the first layer is $v = Vx$, then sphering means that $E\{vv^T\} = I$. The elements of v have variances equal to 1 and are uncorrelated - but in general not independent. The sphering transformation V is implemented by the weights of the linear layer and can be learned in a neural learning algorithm (Plumbley, 1993)

$$V_{k+1} = V_k + \mu_k (V_k x_k x_k^T V_k^T - I) V_k. \tag{13}$$

In the second layer, one of the nonlinear Hebbian learning rules (6) or (7) is used, with the sphered v_k as input instead of the original x_k. Algorithm (7) seems to work somewhat better.

An example is given by Karhunen et al (1995), in which three time domain signals were used - a uniformly distributed noise signal, a ramp signal, and a sinusoid. A mixture of the signals was made artificially using three fixed vectors. The mixture looks quite noisy and the structure of the underlying three signals is lost. The purpose of ICA is to retrieve the independent components, knowing only samples x_k of their mixture.

The network described above was used with 3 input lines and 3 outputs. Vectors x_k were sphered in the first layer and then fed to the nonlinear Hebbian layer. The weight matrix W_k of the second layer learned with algorithm (7). The three outputs $y_k^{(1)}, y_k^{(2)}, y_k^{(3)}$ are fairly good replicas of the original signals. For details, see Karhunen et al (1995).

5 Conclusions

The nonlinear Hebbian learning rules proposed by the author in Oja et al (1991) were reviewed in the light of Projection Pursuit and Independent Component Analysis, following recent ideas of several authors. Previously, it has been shown that these rules are related to Principal Component Analysis in the sense that they tend to find that direction in the input space that has *maximal variance*. However, provided that all directions in the input space have equal variance, the one-unit case with a $tanh(y)$ nonlinearity in the learning is approximately *minimizing the kurtosis* of the neuron output. This means that the weight vector of the unit will find a direction in the input space on which the projection of the input data is heavily clustered and deviates strongly from the Gaussian distribution, like is the case in the Projection Pursuit technique. This is useful in data clustering and visualization.

The multi-unit case has the extra advantage that while minimizing the kurtosis in several directions, it tends to find those directions along which the input data projections are *statistically independent*. Therefore these algorithms can be used for Independent Component Analysis. In both applications, it is essential to remove the influence of first and second order statistics, which can be done by first making the inputs zero mean and then removing different variances by data sphering. This can be done in a simple two-layer network.

References

Comon, P. (1989), in *Proc. of Workshop on Higher-Order Spectral Analysis*, Vail, Co., June 1989, pp. 174 - 179.
Comon, P. (1994), *Signal Processing* 36, pp. 287 - 314.
Friedman, J. and Tukey, J. (1974), *IEEE Trans. Comput.* C-23, pp. 881 - 889.
Fyfe, C. et al (1994), Res. Report 94/160, U. of Strathclyde, Dept. of Computer Science.

Haykin, S. (1994): Neural networks: a comprehensive foundation. Macmillan College Publ. Co., New York, 1994.

Jutten, C. and Herault, J. (1991), *Signal Processing* 24, pp. 1 - 10.

Karhunen and Joutsensalo (1994), *Neural Networks* 7, pp. 113 - 127.

Karhunen, Wang and Joutsensalo (1995), submitted to *ICANN'95*, Paris, Oct. 1995.

Oja, E. (1982), *J. Math. Biol.* 15, pp. 267 - 273.

Oja, E. et al (1991), in *Proc. ICANN'91*, Espoo, June 1991, pp. 385 - 390.

Oja, E. (1992), *Neural Networks* 5, pp. 927 - 935.

Plumbley, M. (1993), *Proc. IEE Conf. ANN'93*, Brighton, UK, May 1993, pp. 86 - 90.

Sudjianto, A. and Hassoun, M. (1994), in *Proc. IEEE ICNN*, Orlando, Fla., July 1994, pp. 1247 - 1252.

sterberg, M. (1994), Dr. Tech. Thesis, U. of Linkping, Dept. of Electrical Engineering, Sept. 1994.