LOCAL ADAPTIVE ALGORITHMS FOR INFORMATION MAXIMIZATION IN NEURAL NETWORKS, AND APPLICATION TO SOURCE SEPARATION

Jeroen Dehaene[*]

Dept. Electrical Engineering
K.U.Leuven ESAT-SISTA,
Kard. Mercierlaan 94,
B 3001 Leuven, Belgium.

Nanayaa Twum-Danso[†]

Div. of Engineering and Applied Sciences,
Harvard University,
29 Oxford Street,
Cambridge MA 02138, USA.

ABSTRACT

Information theoretic criteria for neural network adaptation laws have recently become an important focus of attention. We consider the problem of adaptively maximizing the entropy of the outputs of a deterministic feedforward neural network with real valued stochastic input signals, as considered by Bell and Sejnowski. We give a new explanation for the relevance of output information (entropy) maximization for source separation applications and reinterpret Bell and Sejnowski's approach in a more general context of probability density estimation. This insight is the basis for a generalization of the approach, and we consider a family of gradient based algorithms.

1. BLIND SOURCE SEPARATION, INFORMATION MAXIMIZATION AND PROBABILITY ESTIMATION

The problem of blind separation of independent sources can be formulated as follows. A vector of n stationary input signals $x(t) \in \mathbb{R}^n$ is known to result from mixing n stochastically independent sources $s(t) \in \mathbb{R}^n$:

$$x(t) = \Psi_A(s(t)),$$

where A parameterizes a family of invertible mixing maps Ψ_A, and t denotes continuous or discrete time. Below we will mostly work in continuous-time terms. We will omit t when no confusion is possible. The problem is to reconstruct A and the original sources s from the given input x (and the knowledge that the original sources were independent). The sources s can at best be reconstructed up to a permutation and scaling of the signals. We will assume all signals to be zero mean.

If we consider linear mixtures only, we have $x = As$, where A is a nonsingular $n \times n$ matrix. In general second order statistics $E\{xx^T\}$ are not sufficient to reconstruct s (except if A is further constrained to be orthogonal or triangular for instance). If more than one source is Gaussian, they can be separated from the others but cannot be reconstructed individually. If apart from small "noise sources" there are only $m < n$ sources, one can first apply subspace estimation techniques to reduce the problem to dimension m.

Bell and Sejnowski propose to separate linear mixtures with a one-layer invertible neural network, i.e. an invertible linear transformation, followed by bounded, monotonously increasing, nonlinear functions applied to all outputs separately, and maximize the output entropy[1]. They argue that this approximately minimizes the mutual information between different outputs and therefore achieves independence. Below we give a different justification. The following statements are equivalent:

1. The input signals result from a linear mixture of independent sources, with probability density functions (pdf) $p_{s_k}(s_k)$, $k = 1, \ldots, n$.

2. The input signals result from a "nonlinear mixture" of independent "pre-sources" u_k, uniform on $(0, 1)$, consisting of first a nonlinear transformation $c_{s_k}^{-1}(u_k)$ on all pre-sources separately (yielding s_k), followed by a linear mixing transformation, where $c_{s_k}(s) = \int_{-\infty}^{s} p_{s_k}(\sigma)d\sigma$ is the cumulative density of s_k.

3. The input signals can be transformed into a uni-

[*]Jeroen Dehaene is a postdoctoral researcher of the Flemish Fund for Scientific Research (F.W.O.). This research was also supported by a Concerted Action Project of the Flemish Community, entitled "Model-based Information Processing Systems". The scientific responsibility is assumed by its authors.

[†]This work was supported in part by the National Science Foundation under Engineering Research Center Program, NSF D CDR-8803012, by the US Army Research Office under grant DAAL03-86-K-0171(Center for Intelligent Control Systems), by the Office of Naval Research under Grant N00014–1887, and by the Joint Services Electronics Program Laboratory under grant N00014-89-J-1023

formly distributed signal on $(0,1)^n$ by a one-layer neural net with nonlinearities $c_{s_k}(.)$, playing the role of inverse filter in a reconstruction of the uniform pre-sources.

4. The input pdf belongs to a family of pdfs that can be parameterized as the determinant of the Jacobian (differential) of a one-layer neural net with invertible weight matrix and nonlinearities $c_{s_k}(.)$. The linear part of this net models the linear mixture.

Let $x(t) \in \mathbb{R}^n$ be a real-valued stochastic signal. Let $y(t)$ be the output of a one-layer neural network. For the sake of generalization we consider a family of invertible nonlinear maps, parameterized by a "weight vector" w: $y = \Phi_w(x)$. The third formulation above justifies adaptation of w as to maximize output entropy, since the uniform distribution on $(0,1)^n$ has maximal entropy. Bell and Sejnowski propose to maximize the output entropy

$$E\{-\ln p_y(y)\} = E\{-\ln p_x(x)\} + E\{\ln|\det \tfrac{d\Phi_w}{dx}|\},$$

where p_x and p_y denote the pdfs of x and y. The first term is independent of w and can be dropped, leading to an objective function

$$F(w) = E\{\ln|\det(\tfrac{d\Phi_w}{dx}(x))|\}.$$

The fourth formulation is explained as follows. If the map Φ_w transforms x into uniformly distributed signals on $(0,1)^n$, we have $p_y(y) = p_x(x)/|\det \frac{d\Phi_w}{dx}| = 1$ and $p_x(x) = |\det \frac{d\Phi_w}{dx}|$. Now, we can look at the problem as a probability density estimation problem and parameterize the pdf estimate as $q_w(x) = \det\|\frac{d\Phi_w}{dx}\|$. This way, w parameterizes the density of the signal that would be obtained by applying the inverse transformation Φ_w^{-1} to uniformly distributed signals. A classical estimation procedure consists in minimizing the Kullback-Leibler distance $E\{\ln(p_x(x)/q_w(x))\}$ between the real and the estimated distribution, or equivalently maximizing $E\{\ln q_w(x)\}$. This yields exactly the same objective function $F(w)$ as above.

2. APPLICATIONS

The above interpretation leads to several generalizations, both in the context of probability density estimation and in the context of source separation. The main difference between these two types of problems is that for source separation applications it is important to reach the global maximum as one is interested in the parameters that correspond to the solution of the optimization problem. On the other hand, if one is only interested in a good approximation of the probability density, a local maximum can be a good solution too.

In the next section we will derive gradient algorithms for the case where the map Φ_w is a composition of linear transformations with positive determinant and nonlinear, elementwise monotonously increasing transformations (with positive diagonal Jacobian). In this section we consider different applications.

The case of a linear transformation, followed by a fixed nonlinear transformation corresponds to the one-layer neural network as considered by Bell and Sejnowski. Algorithms for this case can be used for source separation. If the pdf of the sources is known, the fixed nonlinearities should be taken equal to the cumulative density functions of the sources. Optimality conditions give strong necessary conditions for independence of the reconstructed sources. Simulations show good results for nonlinearities that don't exactly match the inequalities. Cardoso and Laheld show for a related algorithm that the separating solution is attractive under an inequality condition involving nonlinear moments of the input signals, which is satisfied in many practical applications[2].

One can also work with adaptable nonlinearities parameterized by a small number of parameters. One way to achieve such modeling is by considering families of nonlinearities parameterized by extra parameters. Another way consists in replacing the nonlinear transformation, by a sequence of linear and nonlinear elementwise transformations, where the linear transformations are diagonal. The latter idea can also be generalized to the case of block diagonal layers instead of diagonal layers. We can think of this case as the separation of vector sources. The first linear transformation can be thought of as separating a mixture while the diagonal linear transformations and the nonlinear transformations model the probability distributions of the sources. The adaptation of the source density models could adapt on a slower time scale, as to prevent fast adaptation to output signals when the mixture is still far from the separating mixture. Simulations with this setting have not yet been carried out.

An alternation of full linear maps with positive determinant (implying an equal number of nodes in every layer) and elementwise monotonously increasing nonlinear maps, where only the last layer of nonlinear functions has to map the signals into $(0,1)^n$, results in a multilayer perceptron. In this case, the interpretation as a source separation algorithm is less appealing. The algorithm can be considered to separate nonlinear mixtures of sources, but only if the unknown nonlinear mixture is known to belong to the family of maps that result from inverting such perceptrons with exclu-

sion of the last layer of nonlinear functions (by analogy with argument of the previous section). However, this objection does not apply to applications in probability estimation and we believe that the presented way of parameterizing pdfs as the determinant of the Jacobian of a neural network type transformation, may be advantageous in different applications where pdfs are now often directly parameterized as a neural network.

3. GRADIENT ALGORITHMS

In this section we consider gradients of the objective function, which can form the basis for different optimization strategies. As the objective function is the expectation of a random variable, this is also the case for the gradient. Stochastic gradient algorithms are obtained by instantaneously estimating the gradient with the current sample of x, that is by omitting the expectation operator.

If the map $\Phi_w(x)$ is a composition of maps $\Phi = \Phi^{(n)}_{w^{(n)}} \circ \cdots \circ \Phi^{(1)}_{w^{(1)}}$, the objective function $F(w)$ falls apart into a sum of terms. Let $y^{(0)} = x$ and $y^{(i)} = \Phi^{(i)}(y^{(i-1)})$, then

$$\begin{aligned} F(w) &= \textstyle\sum_{i=1}^n F^{(i)}(w) \\ &= \textstyle\sum_{i=1}^n E\{\ln|\det(\frac{d\Phi^{(i)}_{w^{(i)}}}{dy^{(i-1)}}(y^{(i-1)}))|\}. \end{aligned}$$

We will consider maps $\Phi^{(i)}_{w^{(i)}}$ that are either linear, $\Phi^{(i)}_{W^{(i)}} = W^{(i)}y^{(i-1)}$ with $\det W^{(i)} > 0$, or elementwise nonlinear and monotonously increasing, $\Phi^{(i)}_k(y^{(i-1)}) = f_k(y^{(i-1)}_k)$, where f_k, $k = 1, \ldots, n$, are n monotonously increasing functions from $\mathbb{R}$ to $(0, 1)$. That is, $\Phi^{(i)}$ has a positive diagonal Jacobian matrix

$$\begin{aligned} J^{(i)}(y^{(i-1)}) &= d\Phi^{(i)}/dy^{(i-1)} \\ &= \operatorname{diag}(f_1'(y_1^{(i-1)}), \ldots, f_n'(y_n^{(i-1)})), \end{aligned}$$

where $'$ denotes the derivative, and the subscript $_k$ indicates the k-th component. With every linear map $\Phi^{(i)}$ corresponds a term $F^{(i)}(W) = \ln\det W^{(i)}$. With a nonlinear map corresponds a term $\ln\det d\Phi^{(i)}/dy^{(i-1)} = \sum \ln f_k'(y_k^{(i-1)})$.

First consider the case of a one-layer neural network, with fixed nonlinearities . Let $y^{(1)} = \Phi^{(1)}(x) = Wx$ with $\det W > 0$ and $y = y^{(2)} = \Phi^{(2)}(y^{(1)})$ where $\Phi^{(2)}_k(y^{(1)}) = f_k(y^{(1)}_k)$.

The objective function $F(W)$ is now

$$F(W) = \textstyle\sum_{k=1}^n E\{\ln f_k'(y_k^{(1)})\} + \ln\det W.$$

Bell and Sejnowski[1] find a gradient

$$\nabla F(W) = E\{h(y^{(1)})x^T + W^{-T}\},$$

where h is an elementwise nonlinear function, defined by $h_k(y_k^{(1)}) = f_k''(y^{(1)})/f_k'(y^{(1)})$. The vector $h(y^{(1)})$ is the gradient of $\sum_{k=1}^n \ln f'(y_k^{(1)})$ as a function of $y^{(1)}$. If $f_k(.) = \tanh(.)$ one finds $h_k(.) = -2\tanh(.)$. The second term in the objective function, $\ln\det W$, gives rise to the term W^{-T} in the gradient, which is responsible for much of the computational cost and complicates parallel realization.

However, a simpler gradient can be obtained by working with a different inner product. We will use the defining property that $\nabla F(w)$ is the gradient of F if any velocity $\frac{dw}{ds}$ at w (i.e. considering any path through w) results in a corresponding change $dF/ds = \langle \nabla F(w), dw/ds \rangle$. This definition is easily generalized for w belonging to a differentiable Riemannian manifold, by requiring ∇F to be a tangent vector at w.

The above result can then be written as

$$\tfrac{d}{ds}F(W) = \operatorname{Tr}((\tfrac{d}{ds}W)^T E\{h(y^{(1)})x^T + W^{-T}\}).$$

Now, it pays to parameterize velocity vectors (i.e. tangent vectors) at W as $\frac{d}{ds}W = KW$, and work with the standard inner product as applied to the parameters K instead of the vector KW: $\langle K_1W, K_2W \rangle = \operatorname{Tr}(K_1^T K_2)$. One finds

$$\begin{aligned} \tfrac{d}{ds}F(W) &= \operatorname{Tr}(W^T K^T E\{h(y^{(1)})x^T + W^{-T}\}) \\ &= \operatorname{Tr}(K^T E\{h(y^{(1)}){y^{(1)}}^T + I)\}, \end{aligned}$$

(using $\operatorname{Tr}(AB) = \operatorname{Tr}(BA)$). Therefore, with this inner product, one finds

$$\nabla F(W) = E\{(h(y^{(1)}){y^{(1)}}^T + I)W\}.$$

One can also work with $\langle WK_1, WK_2 \rangle = \operatorname{Tr}(K_1^T K_2)$, yielding

$$\nabla F(W) = E\{W(W^T h(y^{(1)})x^T + I)\}.$$

Parameterizing $\frac{d}{ds}W$ as $\frac{d}{ds}W = KW$ or $\frac{d}{ds}W = WK$ corresponds to considering first order perturbations $\tilde{W} = (I+\epsilon K)W + o(\epsilon^2)$ or $\tilde{W} = W(I+\epsilon K) + o(\epsilon^2)$, rather than additive perturbations. This can be intuitively interpreted as inserting an infinitesimal linear filter, before or after W. This is closely related to the idea of serial updating [2].

This approach can be generalized to the case where W is constrained to belong to a Lie group or when W is parameterized as a product of matrices that belong to Lie groups. For instance if one considers W to be constrained to the group of matrices with determinant 1 (to simplify the cost function), one should simply constrain K to belong to the corresponding Lie Algebra of traceless matrices. Similarly, if W belongs to the group of orthogonal matrices, K should be skew symmetric. If W is triangular with unit diagonal, K

should be strictly triangular. If W is block diagonal (and invertible), K should be block diagonal.

Imposing constraints on W is meaningful in linear blind source separation applications in view of the fact that the sources, if their densities are unknown, can only be reconstructed up to a scaling (and a permutation). Alternatively, one can reparameterize W to separate the scaling from the real separation. As an example, parameterization of W as DLQ with D diagonal, L lower triangular with unit diagonal and Q orthogonal (with determinant 1), is a way to separate the scaling of the sources by D (which can be thought of as source density modeling) and the separation of the sources, and replacing the term $\ln\det W$ in the cost function, by the simpler term $\ln\det D$, since both L and Q have determinant 1. Also, if the sources are known to be spatially white, W can be constrained to be an orthogonal matrix.

Next, we consider multilayer neural networks, with fixed nonlinearities. For the sake of clarity we focus on a two-layer network. Let $y^{(1)} = \Phi^{(1)}(x) = W^{(1)}x$, $y_k^{(2)} = \Phi_k^{(2)}(y^{(1)}) = f_k^{(2)}(y_k^{(1)})$, $y^{(3)} = \Phi^{(3)}(y^{(2)}) = W^{(3)}y^{(2)}$ and $y_k = y_k^{(4)} = \Phi_k^{(4)}(y^{(3)}) = f_k^{(4)}(y_k^{(3)})$ The objective function $F(W)$ is now

$$F(w) = \sum_{k=1}^n E\{\ln f_k^{(4)'}(y_k^{(3)})\} + \ln\det W^{(3)}.$$
$$+ \sum_{k=1}^n E\{\ln f_k^{(2)'}(y_k^{(1)})\} + \ln\det W^{(1)}.$$

(where $w = (W^{(1)}; W^{(3)})$). If we parameterize again $\frac{d}{ds}W^{(1)}$ as $K^{(1)}W^{(1)}$ and $\frac{d}{ds}W^{(3)}$ as $K^{(3)}W^{(3)}$, the derivation is similar to the one above. The main difference is the influence of the $\frac{d}{ds}W^{(1)} = KW^{(1)}$ on the term $\sum_{k=1}^n E\{\ln f_k^{(4)'}(y_k^{(3)})\}$ through $y_k^{(3)} = W^{(3)}\Phi^{(2)}(W^{(1)}x)$, yielding

$$\begin{aligned}
&\tfrac{d}{ds}\textstyle\sum_{k=1}^n E\{\ln f_k^{(4)'}(y_k^{(3)})\} \\
&= E\{(h^{(4)}(y^{(3)}))^T \tfrac{d}{ds} y_k^{(3)}\} \\
&= E\{(h^{(4)}(y^{(3)}))^T W^{(3)} J^{(2)}(y^{(1)}) \tfrac{dW^{(1)}}{ds} x\} \\
&= \mathrm{Tr}(K^{(1)T} E\{J^{(2)}(y^{(1)}) W^{(3)T} h^{(4)}(y^{(3)}) (W^{(1)}x)^T\}),
\end{aligned}$$

where $J^{(2)}(y^{(1)})$ denotes the Jacobian $\frac{d\Phi^{(2)}}{dy^{(1)}}(y^{(1)})$ as above.

This yields the following (partial) gradients of the total objective function as a function of $W^{(1)}$ and $W^{(2)}$ (together forming the gradient of $F(w)$).

$$\begin{aligned}
&\nabla_{W^{(1)}} F(w) = \\
&E\{([h^{(2)}(y^{(1)}) + J^{(2)}(y^{(1)}) W^{(3)T} h^{(4)}(y^{(3)})] y^{(1)^T} + I) \\
&\quad W^{(1)}\} \\
&\nabla_{W^{(3)}} F(w) = E\{(h^{(4)}(y^{(3)}) y^{(3)^T} + I) W^{(3)}\}.
\end{aligned}$$

Except for the choice of the inner product and extra terms due to costs of the type $\ln\det W^{(i)}$, this derivation is equivalent to backpropagation schemes. The given formulas can easily be extended to more than two layers.

Generalization to the case of adaptable nonlinearities is straightforward. The case where the nonlinear maps are replaced by an alternation of diagonal linear maps and nonlinear maps, is a special case of the above. If the fixed nonlinear functions are $f^{(i)}(.)$ are replaced $f_{w^{(i)}}^{(i)}(.)$ parameterized by $w^{(i)}$, the maps $h(.)$ and $J(.)$, in the formulas for the partial gradients for $W^{(1)}$ and $W^{(3)}$, become also dependent on the parameters, and the extra partial gradients for the extra parameters take the form $\nabla_{w^{(i)}} F(w) = \frac{\partial f_{w^{(i)}}^{(i)}}{\partial w^{(i)}}(y^{(i-1)})$. (Note that $f_{w^{(i)}}$ itself is not needed and need not be easily computable).

4. CONCLUSION

We have considered the problem of adaptively maximizing the entropy of the outputs of a deterministic feedforward neural network with real valued stochastic input signals, and derived different gradient algorithms that can be used for source separation and probability estimation.

5. REFERENCES

[1] A.J. Bell and T.J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

[2] J.F. Cardoso and B. Laheld. Equivariant adaptive source separation. *IEEE Transactions on signal processing*, December 1996.