*LETTER* **Communicated by Jean-Pierre Nadal**

# Neural Relax

**Elisa Benedetti**
*elibene@gmail.com*
**Marco Budinich**
*mbh@ts.infn.it*
*Physics Department and INFN, Trieste 34127, Italy*

**We propose a new self-organizing algorithm for a feedforward network inspired to an electrostatic problem that turns out to have intimate relations with information maximization.**

## 1 Introduction

In this letter, we present a new self-organizing algorithm for a layer of $h$ continuous perceptrons derived from the electrostatic problem of free electrical charges in a conductor. The algorithm is general and maximizes information.

The idea is simple: we use a layer of continuous perceptrons to map the inputs to point-like electrical charges that we imagine free to move within a hypercube in multidimensional space, and we let them evolve or, better, relax under Coulomb repulsion until they set in the minimal energy configuration. For this reason, we named this algorithm neural relax (NR).

We show that this is sufficient to obtain binary and statistically independent data as a natural consequence of the algorithm itself; in addition, by fixing the dimensions of the hypercube, one can freely adjust the rate of dimensional reduction. From a theoretical point of view, we show that in the simple one-dimensional case, this algorithm provides the maximum-information solution to the problem, and thus the learning rules result equal to those obtained by Bell and Sejnowski (1995) from their independent component analysis (ICA), exhibiting a completely different interpretation of the ICA algorithm. In the general multidimensional case, we show that NR gives a pure Hebbian rule and is also well suited to inject some redundancy that subsequently can be used to perform error correction on the processed patterns.

The letter is structured as follows. In section 2, we briefly describe our network. In section 3, we present the real physical problem we refer to, a system of point-like charges confined in a cube, and link it to our problem

E. B. is now at Physics Department T35, Technische Universität München, 85747 Garching bei München, Germany.

 

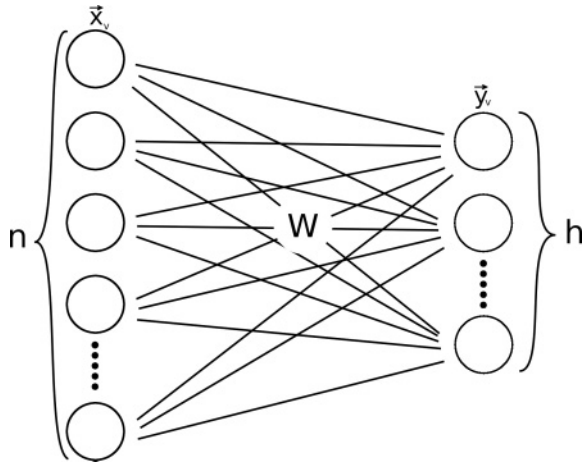


Figure 1: Schematic illustration of the network. An input $\vec{x}_\nu$ is fed to an input layer of $n$ neurons, connected to $h$ neurons that produce the output $\vec{y}_\nu$. The weight matrix $W$ also contains the thresholds that appear as weights of a fictitious 0th input clamped at 1.

and previous work using Coulomb-like forces in neural networks. Then we present a theoretical analysis for the one-dimensional case in section 4 and the general multidimensional case in section 5. We conclude with some preliminary computational results: to test our algorithm, we tackle the problem of preprocessing real-world binary images to make them unbiased, uncorrelated, and binary.

## 2 A Layer of Perceptrons

We consider a layer of $h$ perceptrons with $n$ inputs and tanh() transfer function. Given an input $\vec{x} \in \mathbb{R}^n$, each perceptron gives the output

$$y_i = \tanh\left(\vec{w}_i \cdot \vec{x}\right) = \tanh\left(\sum_{j=0}^{n} w_{ij} x_j\right) \qquad i = 1, \ldots, h, \tag{2.1}$$

and Figure 1 schematically illustrates the architecture of this network. We stretch the notation a bit, indicating the $h$ equations, equation 2.1, with the weight matrix $W$,

$$\vec{y} = \tanh\left(W\vec{x}\right). \tag{2.2}$$

This is a common, well-studied network that, among other things, can be used to approximate any continuous function since the transfer

function, $\tanh(x)$, is bounded in $(-1, 1)$, nonconstant, smooth, and monotone (Hornik, Stinchcombe, & White, 1989). We assume that the inputs follow a distribution $p(\vec{x})$ and that there is no noise; usually we consider binary inputs $\vec{x} \in \{\pm 1\}^n$. We focus on the case of binary outputs $\vec{y} \in \{\pm 1\}^h$, that is, the limit of the continuous case, equation 2.2, when the argument is large.[1]

Nadal and Parga (1993) studied this network when $\vec{y} = \text{sgn}(W\vec{x})$ in the frame of information theory. They showed that the information capacity $C$ that can be conveyed by $h$ binary neurons is bounded by $h$,

$$C := \max_{p(\vec{x})} I(\vec{x}; \vec{y}) \leq h,$$

where $I(\vec{x}; \vec{y})$ is the mutual information between the input $\vec{x}$, of distribution $p(\vec{x})$, and the output $\vec{y}$. The limitation comes essentially from the architecture since $h$ binary neurons can possibly implement only $C_{h,n} \leq 2^h$ of the theoretically possible $2^h$ output states, and they show that

$$C = \log_2 C_{h,n} = \begin{cases} h & \text{for } h \leq n \\ < h & \text{for } h > n \end{cases}.$$

So assuming $h \leq n$, we see that the architecture does not impose any limitation[2] and, for these binary neurons without noise, the upper bound $C$ can be reached if, and only if the distribution of the outputs $q(\vec{y})$ results fully factorized (Nadal & Parga, 1993):

$$q(\vec{y}) = \prod_{i=1}^{h} q\left(y_i\right) \quad \text{with} \quad q\left(y_i = \pm 1\right) = \frac{1}{2} \quad \forall i. \tag{2.3}$$

With the help of this analysis we can set up a list of desirable characteristics for the function $f : \mathbb{R}^n \to \mathbb{R}^h$, equation 2.2, implemented by our layer of $h$ perceptrons:

- The output patterns should be (essentially) binary, that is, $1 - |y_i| < \epsilon$.
- The map $f : \mathbb{R}^n \to \mathbb{R}^h$ should be injective and such that equation 2.3 holds.
- As a consequence, the produced data will be statistically independent:

  $$E[y_{i_1} y_{i_2} \dots y_{i_r}] = 0 \quad \forall\, i_1 \neq i_2 \neq \dots \neq i_r, \quad \forall\, 1 \leq r \leq h$$

  (and thus uncorrelated $E[y_i y_j] = 0 \quad \forall\, i \neq j$).

[1] Given that $\lim_{\beta \to \infty} \tanh(\beta x) = \text{sgn}(x)$.

[2] This is different from the request that there is no information loss that depends on the source entropy $\mathcal{S}(\vec{x})$ and would require that $h \geq S(\vec{x})$.

- It should accomplish dimensionality reduction whenever possible, that is, $h \ll n$.
- It should be learnable, that is, it should be possible to find it by gradient descent along an appropriate function of the weights.

The most demanding goal is satisfying equation 2.3, but it is not easy to find an algorithm that does it directly. Several authors followed the equivalent path of maximizing the mutual information $I(\vec{x}; \vec{y})$, for example, the ICA algorithm (Bell & Sejnowski, 1995; see also Pham, 2001). Our algorithm starts from a physical problem that leads naturally toward fulfilling these requests.

## 3 The Physical Problem

We consider the problem of finding the stable equilibrium position of $m$, equal, point-like, electric charges $Q_\nu$ within a cube of conductor. This is a problem very similar to the Thomson (1904) problem, where the charges are in a sphere. The problem is remarkably difficult to solve, and exact solutions are known only for a few values of $m$ (see Schwartz, 2010). From now on, we always consider our cube centered at the origin and with a side of length 2. The physical space available to the charges is the three-dimensional cube defined by

$$H_3 = \{\vec{y} \in \mathbb{R}^3 : |y_i| < 1 \quad i = 1, 2, 3\},$$

the extension to the $h$-dimensional hypercube $H_h$ being obvious. In an ideal conductor, the $m$ charges are free to move, and their stable rest positions $\vec{y}_\nu$ minimize the Coulomb potential (Jackson, 1999):[3]

$$U\left(\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_m\right) = \sum_{\mu<\nu} \frac{Q_\mu Q_\nu}{|\vec{y}_\mu - \vec{y}_\nu|} \qquad \mu, \nu = 1, \ldots, m.$$

$U\left(\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_m\right)$ is a harmonic function (Axler, Bourdon, & Ramey, 2001) and thus does not have minima in an open, convex set like $H_3$. Thus, the rest positions of the charges are on the border, namely, on the surface of the cube. Moreover we conjecture that if the charges are equal and their number is $m \leq 2^3 = 8$, the only stable positions of the charges are on cube vertices, shown in Figure 2, which contains the minimum energy arrangements for two, three, four, and five charges.[4]

[3] In gaussian units: $\frac{1}{4\pi\epsilon_0} = 1$.

[4] Despite several attempts, we have not been able to prove this formally, but numerical simulations support the conjecture.

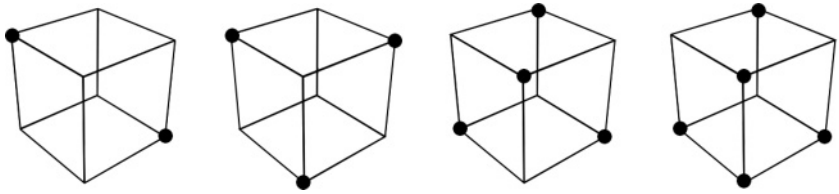

Figure 2: Stable equilibrium configurations of point-like charges in a cubic box. Particles are arranged in such a way as to maximize their reciprocal distances while minimizing the Coulomb potential energy. Since they occupy the vertices, they have (almost) binary coordinates in the defined set $H_3$.

This problem easily generalizes from $\mathbb{R}^3$ to $\mathbb{R}^h$ provided that $U(\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_m)$ remains harmonic, and this happens iff the distance between charges generalizes to

$$|\vec{y}_\mu - \vec{y}_\nu| := [(\vec{y}_\mu - \vec{y}_\nu) \cdot (\vec{y}_\mu - \vec{y}_\nu)]^{\frac{h-2}{2}}. \tag{3.1}$$

Also in this case, the rest positions of the charges must be on the border of $H_h$, and we generalize our conjecture that charges have stable rest positions on the vertices of $H_h$ and, consequently, (almost) binary coordinates.

We take inspiration from this physical problem to propose a self-organizing algorithm for a layer of continuous perceptrons. We map our set of $m$ inputs in $\mathbb{R}^n$ to point-like charges in $\mathbb{R}^h$, and these charges are bound to remain in the $h$-dimensional hypercube. Subsequently we let this system evolve under Coulomb repulsion in $\mathbb{R}^h$, minimizing its energy until it reaches equilibrium. Provided that our conjecture is true and if $m \leq 2^h$, the charges at rest will occupy the vertices of $H_h$ and thus have binary coordinates, which means that this approach allows us to get a binary representation of the input data as a natural consequence and without any further constraint. We will also show that this process maximizes information.

More in detail, given a set of $m$ inputs $\vec{x}_\nu \in \mathbb{R}^n$, $\nu = 1, 2, \ldots, m$ of distribution $p(\vec{x})$, by applying equation 2.2, we get $m$ outputs $\vec{y}_\nu \in \mathbb{R}^h$ that the hyperbolic tangent constrains within the $h$-dimensional hypercube $H_h$. To treat inputs of different probability $p(\vec{x}_\nu)$, we postulate that the probability of an output $\vec{y}_\nu$ is proportional to the energy of a charge $Q_\nu$ in the electric field,

$$q(\vec{y}_\nu) \propto E(Q_\nu) = Q_\nu \sum_{\substack{\mu=1 \\ \mu \neq \nu}}^{m} \frac{Q_\mu}{|\vec{y}_\mu - \vec{y}_\nu|}, \tag{3.2}$$

and the total energy of the system is

$$U(\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_m) = \sum_{\nu=1}^{m} E(Q_\nu) = \sum_{\mu<\nu} \frac{Q_\mu Q_\nu}{|\vec{y}_\mu - \vec{y}_\nu|}. \tag{3.3}$$

For simplicity, most of the time we assume that all inputs are equiprobable $p(\vec{x}_\nu) = \frac{1}{m}$ and thus will feel free to put $Q_\nu = 1$ for all $m$ charges and the function to minimize is the simplified Coulomb potential:

$$U\left(\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_m\right) = \sum\nolimits_{\mu<\nu} \frac{1}{|\vec{y}_\mu - \vec{y}_\nu|} \qquad \mu, \nu = 1, \ldots, m. \tag{3.4}$$

This "energy" is the function that the NR learning algorithm minimizes, modifying the elements of the weight matrix $W$ by gradient descent,

$$w'_{ij} = w_{ij} - \epsilon \frac{\partial U\left(\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_m\right)}{\partial w_{ij}}, \tag{3.5}$$

$\epsilon$ being a small positive constant.

Let us suppose that NR has been successfully applied and that the harmonic function $U$ has been minimized (more on this later). All of the $m$ charges have relaxed in the minimum energy configuration and necessarily lie on the $H_h$ surface. If $m \leq 2^h$ and our conjecture is true, they sit precisely on the vertices of the hypercube $H_h$. It follows that all coordinates of their positions $\vec{y}_\nu$ are binary and satisfactorily represent the outputs of $h$ binary neurons.

With the distance definition in equation 3.1, we know that $U$ is harmonic and the gaussian theorem holds. We use these properties to show that the positions of our charges satisfy equation 2.3 in the limit $n, m, h \to \infty$ when we can neglect the granularity of the charges and assume that the charge distribution becomes continuous. A similar approach is usually taken for idealized physical conductors where one forgets the quantization of electron charges since the single electron charge is considered negligible with respect to the total charge on the conductor.

When the charges have relaxed in the minimum energy configuration, we know that there is no electric field within the conductors and that all charges lie on the (hyper)-surface; moreover, the spatial density of the charges must be constant in the limit $n, m, h \to \infty$. It follows, given the $H_h$ structure, that every hyperplane through the origin of $\mathbb{R}^h$ and that does not hit any vertex of $H_h$ (to avoid complications) cuts $H_h$ into two parts that contain the same number of vertices, since if vertex $\vec{v}$ belongs to one of the semispaces, vertex $-\vec{v}$ must belong to the other one.[5] From the constancy of the spatial density of the charges, it follows that the two semispaces must also contain exactly the same charge, half of the total charge on $H_h$. Since this result is valid for any hyperplane through the origin of $\mathbb{R}^h$, it is true also for the $h$ hyperplanes

[5] One can observe that if the charges sit on hypercube vertices, they also lie on the hypersphere of radius $h^{\frac{h-2}{2}}$ and continue the following proofs for the hypersphere.

$y_i = 0$. This means that there are exactly $\frac{m}{2}$ charges with $y_i = 1$ (remember that all coordinates are binary) and the same number with $y_i = -1$. In the language of our layer of perceptrons and since $m \to \infty$, this means that the output distribution is such that

$$q\left(y_i = \pm 1\right) = \frac{1}{2} \quad \forall i.$$

It is also easy to prove by induction that $q(\vec{y}) = \prod_{i=1}^{h} q\left(y_i\right)$. We begin showing that $q(y_i, y_j) = q(y_i)q(y_j)$ for any couple of different coordinates $y_i$ and $y_j$. We suppose we have cut our charge distribution into two equal parts by the hyperplane $y_i = 0$ and consider the orthogonal hyperplane $y_j = 0$. It is easy to use the previous argument to show that in all four subspaces so defined, the charges must be equal to $\frac{m}{4}$, and thus for any choices of the values of $y_i$ and $y_j$, one gets $q(y_i, y_j) = \frac{1}{4}$ and thus $q(y_i, y_j) = q(y_i)q(y_j)$. Now suppose that $q(y_{i_1}, y_{i_2}, \ldots, y_{i_k}) = \prod_{j=1}^{k} q(y_{i_j}) = \frac{1}{2^k}$ for any choice of $k$ variables $y_{i_1}, y_{i_2}, \ldots, y_{i_k}$. It is easy to exploit the structure of $H_h$ to show that if one adds a $(k+1)$th coordinate, the hyperplane of equation $y_{i_{k+1}} = 0$ will cut all the previous charges into two halves and thus that $q(y_{i_1}, y_{i_2}, \ldots, y_{i_k}, y_{i_{k+1}}) = \prod_{j=1}^{k+1} q(y_{i_j}) = \frac{1}{2^{k+1}}$, completing the proof by induction. A technical point: only for $m = 2^h$ can one continue the induction chain up to step $k = h$, giving $q(\vec{y}) = 2^{-h}$ for any $\vec{y}$ and complete factorization of the distribution $q(\vec{y})$. If $m < 2^h$, one can prove only that all the moments of order $k$ of $q(\vec{y})$ are zero up to $k = \lfloor \log_2 m \rfloor$.

We have thus proved that if the $m$ charges relax in the configuration of minimal energy (that, by the way, is far from being unique, given the many symmetries of the system), the final positions of the charges satisfy all the requests set for a layer of perceptrons at the end of the previous section and in particular, that the final distribution is fully factorized (see equation 2.3) that implies that the information produced at the output is maximal.

One point we left behind deserves attention: we saw that $U(\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_m)$ is a function of the $mh$ charges coordinates and is provably harmonic, but in our case, with equation 2.1, we can change $\vec{y}_i$ coordinates only through the $(n+1)h$ weights $w_{ij}$. It is simple to verify that $U(w_{ij})$ is no more harmonic:

$$\frac{\partial U}{\partial w_{ij}} = \frac{\partial U}{\partial y_i}\frac{\partial y_i}{\partial w_{ij}},$$

$$\frac{\partial^2 U}{\partial w_{ij}^2} = \frac{\partial^2 U}{\partial y_i^2}\left(\frac{\partial y_i}{\partial w_{ij}}\right)^2 + \frac{\partial U}{\partial y_i}\frac{\partial^2 y_i}{\partial w_{ij}^2},$$

and in general, $\nabla^2 U(w_{ij}) = \sum_{i,j} \frac{\partial^2 U}{\partial w_{ij}^2} \neq 0$. This means that the restrictions imposed on the positions of the charges $\vec{y}_\nu$ by the fact that they are defined by $\vec{y} = \tanh(W\vec{x})$—which also enforces the constraints $\vec{y}_\nu \in H_h$—renders the energy no more harmonic in the "free" coordinates $w_{ij}$. This implies that we cannot formally prove that the function $U(w_{ij})$ is without local minima and that gradient descent 3.5 will always bring the system to one of the solutions we just described because we can move the charge positions $\vec{y}_\nu$ only through the variation of the weights $w_{ij}$, not freely.

One could argue that it is reasonable to expect that the characteristics of the solution will not change dramatically, especially if $m \ll 2^h$, and the charges are very far from each other on $H_h$. But the strength of a formal proof is lost. This argument surely deserves further investigation and will be the subject of future work.

We conclude this section with a brief review of other appearances of Coulomb-like forces in the context of neural networks. The series started in 1987 when Bachmann, Cooper, Dembo, and Zeitouni (1987) proposed an associative memory that attached negative electrical charges to the stored patterns and the memory played the role of a positive charge attracted by the patterns. In this fashion, they could store unlimited patterns and the memory had no spurious state. This idea resurged seven year later (Perrone & Cooper, 1995).

After some years, Marques and Almeida (1999) proposed a feedforward network dedicated to the separation of nonlinear mixtures that minimized a function of three terms. The first term, $W$, was inspired by the idea of repulsion of equal charges and produced a repulsive force. This force was nonphysical since the repulsion had a finite range and acted only in proximity to the patterns. The minimization of this term tended to keep the patterns far apart, producing an approximately uniform distribution of the patterns. To this term, they had to add a term $B$, enforcing the constraints of the outputs in $[-1, 1]$ not to have the patterns fly to infinity and a regularizing term $R$. This work has subsequently been analyzed in a mathematical setting (Theis, Bauer, Puntonet, & Lang, 2001), where it has been shown that within certain approximations, a repulsive force decreasing faster than the Coulomb force tends to produce uniform probability density of the outputs, which in turn maximizes output entropy and in turn minimizes mutual information and is thus amenable to ICA.

None of these works has a real, physical Coulomb energy that is central to our approach since it will allow us to define a positive-definite probability density (see equation 4.2) and will provide an energy that, at least in the ideal case, is harmonic and thus gives important properties to the function to be minimized. This kind of potential matches perfectly with the hypercube structure since charges tend to put themselves on the hypercube vertices, thus automatically satisfying the other request of having binary coordinates. This produces a distribution of the patterns that microscopically is highly

nonuniform, being the discrete sum of point-like charges. From a larger distance, this distribution appears uniform thanks to the gauss theorem (as in real conductors).

## 4 Analysis of the One-Dimensional Case

We start analyzing NR properties in a toy problem: a layer made of just one neuron with one input (i.e., a purely one-dimensional problem). This is a well-studied case (Atick, 1992; Nadal & Parga, 1994; Bell & Sejnowski, 1995), where theoretical analysis is simpler. Here equation 2.1 becomes

$$y = \tanh(wx + w_0). \tag{4.1}$$

To analyze this case, we relax the condition of digital inputs since this would restrict us to the too simple case $x = \pm 1$. So here we suppose continuous inputs $x$ with probability distribution $p(x)$. Correspondingly, we have continuos $y$ with an electrical charge density $\rho(y)$, and the energy of the system, equation 3.3, becomes

$$U = \iint \frac{\rho\left(y\right)\rho\left(y'\right)}{|y - y'|}\,\mathrm{d}y\mathrm{d}y'.$$

Calling $\phi\left(y\right) := \int \frac{\rho(y')}{|y-y'|}\,\mathrm{d}y'$ the total potential of point $y$, we have

$$U = \int \rho\left(y\right)\phi\left(y\right)\,\mathrm{d}y := \int q(y)\,\mathrm{d}y, \tag{4.2}$$

where $q(y)$ is the linear energy density that is by definition positive since it is proportional to the squared electric field (Jackson, 1999). It is thus possible to extend equation 3.2 and interpret $q(y)$ (suitably normalized) as the probability density distribution of $y$. Our problem, given $x$ and $p(x)$, is to determine the parameters $w$, $w_0$ that minimize $U$.

We can gain insight into the solution of this problem by first examining the corresponding physical problem. Since our charges in $y$ are to be imagined as free charges in a conductor, this is the physical problem of the charge distribution on a finite (remember $-1 < y < 1$), infinitely thin conductive wire.

It is a typical electrostatic problem: one has to find the charge distribution $\rho\left(y\right)$ that minimizes $U$. In this case, we are in a conductor, and when the energy is minimized, the potential is constant $\phi\left(y\right) = \phi_0$. Mathematically, the problem is to find the charge distribution $\rho\left(y\right)$ that realizes this condition. This is not an easy problem (it was the subject of James Clerk Maxwell's last scientific paper; see Jackson, 2002), but it is known (Jackson, 2000) that as the ratio of the physical dimensions of the wire goes to zero, the distribution

of the charges on the wire $\rho(y)$ tends to a uniform distribution: $\rho(y) \to \rho_0$. So we can conclude that the physical solution that minimizes equation 4.2 gives $q(y) = \rho_0\phi_0$.

This is true for the physical problem where, since the charges in the wire are free to move, the distribution of charges $\rho(y)$ can take any shape. It is also clear that in our case, where we can play only with the parameters $w, w_0$ to modify $\rho(y)$, in general it will be impossible to find values of $w, w_0$ that realize the condition $q(y) = \rho_0\phi_0$.

But let us suppose that we are in this lucky situation. To understand the meaning for our problem, we use the well-known relation for the transformation of a distribution $p(x)$ when the variable $x$ is transformed to $y = f_w(x)$, where $w$ represent the parameters of the function $f()$ that has to be invertible. In this case, the distribution $q(y)$ of $y$ is given by

$$q(y) = \frac{p(x)}{\left|\frac{\partial f_w(x)}{\partial x}\right|},$$

and this relation tells us that to get a constant $q(y)$ necessarily $|\frac{\partial f_w(x)}{\partial x}| \propto p(x)$ and thus the function $y = f_w(x)$ needs to be proportional to the primitive of the probability distribution of $x$, namely

$$f_w(x) \propto \int p(x)\,\mathrm{d}x \tag{4.3}$$

and it is well known that this represents the maximum entropy solution for our one-neuron net (Atick, 1992). So, if by adjusting $w$ and $w_0$ we can obtain that equation 4.3 holds, and our system minimizes energy (see equation 4.2), and this solution also gives the maximum information. In our case (see equation 4.1), one obtains

$$\tanh'(wx + w_0)|w| \propto p(x),$$

where we used the fact that $\tanh'(x) > 0$. This relation can also be interpreted to give the only possible $p(x)$ for which we get the optimal solution. As one of the referees pointed out, this can be a severe limitation that one could remedy by adapting not just the weights, but, as Nadal and Parga (1994) did, the transfer function itself $f_w(x)$. This would produce a more powerful neuron, but following Bell and Sejnowski's ICA, we decided purposely not to open this Pandora's box at this stage.

Now we analyze what happens in the general case when equation 4.3 cannot be satisfied exactly and the best one can do is to find the values of

the parameters $w$ that minimize $U$. We therefore study

$$\frac{\partial U}{\partial w} = \frac{\partial}{\partial w}\int q(y)\,\mathrm{d}y = \int \frac{\partial}{\partial w}\frac{p(x)}{\left|\frac{\partial f_w(x)}{\partial x}\right|}\,\mathrm{d}x, \tag{4.4}$$

where we applied Leibnitz's rule for differentiation under the integral since we are dealing with continuous functions. We observe that the only term that depends on $w$, and is thus affected by the derivative, is $|\frac{\partial f_w(x)}{\partial x}|$.

We conclude this section showing that the learning rules for our network, equation 4.1, obtained by equation 4.4, are equivalent to Bell and Sejnowski's ICA (1995). We start performing the derivation with respect to $w$ and $w_0$,

$$\begin{aligned}
-\frac{\partial U}{\partial w} &= \int \frac{p(x)}{\left[{f_w}'\left(wx+w_0\right)|w|\right]^2}\left[{f_w}''\left(wx+w_0\right)|w|x\right.\\
&\qquad \left.+\frac{|w|}{w}{f_w}'\left(wx+w_0\right)\right]\mathrm{d}x,\\
-\frac{\partial U}{\partial w_0} &= \int \frac{p(x)}{\left[{f_w}'\left(wx+w_0\right)|w|\right]^2}\left[{f_w}''\left(wx+w_0\right)|w|\right]\mathrm{d}x
\end{aligned}$$

with our choice $y = f_w\left(wx+w_0\right) = \tanh\left(wx+w_0\right)$. Then

$$\begin{cases} {f_w}'\left(wx+w_0\right) = 1-y^2 > 0 \\ {f_w}''\left(wx+w_0\right) = -2y\left(1-y^2\right) \end{cases}, \tag{4.5}$$

which, substituted in previous equations, gives

$$\begin{aligned}
-\frac{\partial U}{\partial w} &= \int \frac{p(x)}{\left[\left(1-y^2\right)|w|\right]^2}\left[-2y\left(1-y^2\right)|w|x+\frac{|w|}{w}\left(1-y^2\right)\right]\mathrm{d}x =\\
&= \int \frac{p(x)}{\left(1-y^2\right)|w|^2}\left[-2y|w|x+\frac{|w|}{w}\right]\mathrm{d}x =\\
&= \int \frac{p(x)}{\left(1-y^2\right)|w|}\left[\frac{1}{w}-2yx\right]\mathrm{d}x,\\
-\frac{\partial U}{\partial w_0} &= \int \frac{p(x)}{\left[\left(1-y^2\right)|w|\right]^2}\left[-2y\left(1-y^2\right)|w|\right]\mathrm{d}x =\\
&= \int \frac{p(x)}{\left(1-y^2\right)|w|^2}\left[-2y|w|\right]\mathrm{d}x = \int \frac{p(x)}{\left(1-y^2\right)|w|}\left[-2y\right]\mathrm{d}x.
\end{aligned}$$

Comparing these equations with equation 4.2, we note that the term $\int \frac{p(x)}{(1-y^2)|w|}\,dx$ is nothing but the Coulomb energy $U$ integrated over $x$. Hence, as anticipated, it is possible to interpret it as a distribution over which the terms in square brackets can be considered averaged, so we can also write them as expectation values:

$$\begin{cases} -\dfrac{\partial U}{\partial w} = E_U\left[\dfrac{1}{w} - 2yx\right] \\ -\dfrac{\partial U}{\partial w_0} = E_U\left[-2y\right] \end{cases}.$$

Comparing these relations with ICA's learning rules (Bell & Sejnowski, 1995) (remembering that we use slightly different transfer functions), we see that they are equal. This shows that NR and ICA are intimately related and that even if they start from completely different starting points, essentially both end up maximizing information.

## 5 The Multidimensional Case

We now proceed to examine the general multidimensional case. We start with $m$ binary inputs of $n$ bits each (that in our numerical simulations will be binary images),

$$\vec{x}_\nu \in \{\pm 1\}^n \quad \nu = 1, 2, \ldots, m,$$

fed to a layer of $h$ neurons thus producing, for each input,

$$\vec{y}_\nu = \tanh(W\vec{x}_\nu) \in (-1, 1)^h \quad \nu = 1, 2, \ldots, m,$$

where the dimensionality of the output layer $h$ is a quite arbitrary choice. It somehow represents the compression rate of the system.[6] To each output $\vec{y}_\nu$ produced, we attach an arbitrary unitary electric charge. Then we calculate the Coulomb potential (see equation 3.4) and apply gradient descent to it to obtain the learning rules. With the standard distance definition in $h-$dimensional space (see equation 3.1), we get

$$|\vec{y}_\mu - \vec{y}_\nu| = \left[\sum_{i=1}^{h} \left(y_{\mu i} - y_{\nu i}\right)^2\right]^{\frac{h-2}{2}},$$

[6] As proposed in Nadal and Parga (1993), one can distinguish three cases: $h < \mathcal{S}(\vec{x})$, where the net must "compress" the data with some information loss; $h = \mathcal{S}(\vec{x})$, where the net is perfectly matched to the incoming information; and $h > \mathcal{S}(\vec{x})$, where the net is redundant but, as explained later, with NR, this redundancy can be used for error correction.

which gives the learning rule for $h > 2$,

$$\Delta w_{ij} = -\frac{\partial U}{\partial w_{ij}} = -\frac{\partial}{\partial w_{ij}} \sum_{\mu<\nu} \frac{1}{|\vec{y}_\mu - \vec{y}_\nu|} =$$

$$= \sum_{\mu<\nu} \frac{2-h}{|\vec{y}_\mu - \vec{y}_\nu|^{\frac{h}{h-2}}} \left(y_{\nu i} - y_{\mu i}\right) \left[x_{\mu j}\left(1 - y_{\mu i}^2\right) - x_{\nu j}\left(1 - y_{\nu i}^2\right)\right], \quad (5.1)$$

where we used the properties 4.5 of the hyperbolic tangent.

We used the only possible definition of the distance $|\vec{y}_\mu - \vec{y}_\nu|$ that renders the energy $U$ harmonic in the $mh$ variables $y_{\nu i}$ but this is of little use for us, since in general, $U$ is not harmonic with respect to our "free" variables $w_{ij}$.

We have thus felt free to try another definition for the distance with the objective of obtaining a faster learning algorithm. For these reasons, we considered the expression

$$|\vec{y}_\mu - \vec{y}_\nu|_H := \left[2\left(h - \vec{y}_\mu \cdot \vec{y}_\nu\right)\right]^{\frac{h-2}{2}} = \left[2h\left(1 - \frac{\sum_{i=1}^h y_{\mu i} y_{\nu i}}{h}\right)\right]^{\frac{h-2}{2}},$$

that is, a distance in the mathematical sense. Indeed, it is a slightly modified version of the so-called Hamming distance, which is a measure of the difference between two strings of equal length.[7] With this new distance plugged in equation 3.4 we define a slightly different energy function $U_H$ that still diverges when any two charges get too near each other. When minimizing $U_H$, the learning rule becomes

$$\Delta w_{ij} = -\frac{\partial U_H}{\partial w_{ij}} = -\frac{\partial}{\partial w_{ij}} \sum_{\mu<\nu} \frac{1}{|\vec{y}_\mu - \vec{y}_\nu|_H} =$$

$$= \sum_{\mu<\nu} \frac{2-h}{|\vec{y}_\mu - \vec{y}_\nu|_H^{\frac{h}{h-2}}} \left[x_{\mu j} y_{\nu i}(1 - y_{\mu i}^2) + x_{\nu j} y_{\mu i}(1 - y_{\nu i}^2)\right], \quad (5.2)$$

which is similar to the previous rule in equation 5.1, with the only difference that it contains only the "crossed" Hebbian terms $x_{\mu j} y_{\nu i}$ and $x_{\nu j} y_{\mu i}$ without the subtraction of the "straight" terms $x_{\mu j} y_{\mu i}$ and $x_{\nu j} y_{\nu i}$ and that in numerical simulation appears to be faster.

This modified Hamming distance can be easily related to the Euclidean distance, equation 3.1, observing that since the output of the hyperbolic

[7] In our notation, the Hamming distance between binary vectors $\vec{y}_\mu, \vec{y}_\nu \in \{\pm 1\}^h$ is $\frac{1}{2}(h - \vec{y}_\mu \cdot \vec{y}_\nu)$.

tangent is in $(-1, 1)$, it follows that $0 \leq \vec{y}^2 \leq h$, and so

$$|\vec{y}_\mu - \vec{y}_\nu| = \left[(\vec{y}_\mu - \vec{y}_\nu) \cdot (\vec{y}_\mu - \vec{y}_\nu)\right]^{\frac{h-2}{2}} = \left[\vec{y}_\mu^2 + \vec{y}_\nu^2 - 2\vec{y}_\mu \cdot \vec{y}_\nu\right]^{\frac{h-2}{2}}$$
$$\leq \left[2(h - \vec{y}_\mu \cdot \vec{y}_\nu)\right]^{\frac{h-2}{2}} = |\vec{y}_\mu - \vec{y}_\nu|_H \qquad \forall \vec{y} \in H_h.$$

The Euclidean and the Hamming distances coincide if, and only if, each component of each output vector is binary, which is basically what we hope to get at equilibrium. In terms of the energy, we can thus write

$$U\left(\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_m\right) \geq U_H\left(\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_m\right) \qquad \forall \vec{y} \in H_h. \tag{5.3}$$

We see then that the energy defined with the Hamming distance is a lower bound for the energy defined making use of the Euclidean one. In principle, at equilibrium, we can expect the two energies to be equal.

Learning rules 5.1 and 5.2 share two characteristics. The first is that they are Hebbian since they are perfectly local in the sense that the synapse $w_{ij}$ connecting neuron $y_i$ to input $x_j$ is updated only with the values taken by these neurons. At the same time, the value of the synapse is updated by the product $x_j y_i$ referring only to different patterns. In other words, to update a synapse, one needs the "history" of the two neurons (one could say that the rule is local in space but nonlocal in time). The second interesting characteristic is that in both rules appear the terms $(1 - y_{\nu i}^2)$ that tend to kill the learning when $|y_{\nu i}| \simeq 1$, that is, when the coordinates are substantially binary. This inhibits the weights from growing indefinitely.

We conclude this section observing that the outputs produced by this network are suited to implement error detection and correction. In other words, the injective map $f : \mathbb{R}^n \to \mathbb{R}^h$, equation 2.2, implemented by our network de facto acts as an encoder that realizes a block $(m, h)$ code (see, e.g., Cover & Thomas, 2006). Suppose that $m < 2^h$; there are fewer patterns $\vec{y}_\nu$ and then hypercube vertices to park them and $U$ has been minimized. Given the form of the energy minimized by learning (see equation 3.4), we know that each charge $\vec{y}_\nu$ will be on a hypercube vertex and as far as possible from all other charges. Let us suppose that the minimum Hamming distance between different charges $\vec{y}_\nu$ is $d$. It is well known that in this case, one can detect up to $d - 1$ errors on the patterns $\vec{y}$ and correct up to $\lfloor \frac{d-1}{2} \rfloor$ errors. For example, in the numerical simulations of the next section, for $m = 7$ and $h = 64$, the minimum Hamming distance between different patterns is larger than $d = 36$. This means that if one is given a noisy version $\vec{y}'_\nu$ of the pattern $\vec{y}_\nu$ (e.g., as returned by an associative memory), one can try to restore the original pattern. By the way, the restoration could be done by minimizing again the potential energy $U$ that it is no more minimal when the correct pattern $\vec{y}_\nu$ is replaced by its noisy version $\vec{y}'_\nu$ that results "out of place."

## 6 Preliminary Numerical Results

We start introducing the problem we tackled to test NR: preprocessing real-world data to build a binary, uncorrelated representation. We had in mind preprocessing binary images for an associative memory, but this task is by no means limited to this particular problem.

Associative memories were one of the first applications of the neural networks paradigm: introduced in 1969 by Willshaw, Buneman, and Longuet-Higgins (1969), they have produced many offspring (see the classic book by Hertz, Krogh, & Palmer, 1991, or, for a more recent review, see Knoblauch, 2011, which, embeds all flavors of associative memories in a unique Bayesian frame). We focus on the (classic) family of associative memories made of a network of $n$ McCulloch and Pitts neurons, each of them updating its state $S_i \to S_i'$ with the standard rule

$$S_i' = t\left(\sum_{j=1}^{n} w_{ij} S_j\right), \tag{6.1}$$

where the transfer function $t(x)$ can be smooth (e.g., $t(x) = \tanh(x)$), or binary (e.g., $t(x) = \mathrm{sgn}(x)$). Different kinds of associative memories have different connection schemes and different rules for the synapses $w_{ij}$, but all models agree that the information is stored in synapses. An associative memory storing $m$ patterns $\vec{\xi_\nu}$, $\nu = 1, \ldots, m$ should be able to find any of the stored patterns starting from a partial or noisy cue. More precisely, if the network is initially in state $\vec{S_0}$, the (repeated) application of equation 6.1 should bring the network in one of the stored states, that is, $\vec{S_0} \to \vec{S} = \vec{\xi_\nu}$.

A common simplification easing analytical calculations is that of assuming the distribution of the stored patterns to be fully factorized and unbiased,

$$P(\vec{\xi}) = \prod_{i=1}^{n} p\left(\xi_i\right) \quad \text{with} \quad p\left(\xi_i = \pm 1\right) = \frac{1}{2} \quad \forall i, \tag{6.2}$$

which implies that the patterns are statistically independent and binary. This request is exacting, and if it is strictly respected, it rules out immediately all real-world data, for example, binary images or sparse coded data.

So to deal with these data, one needs to transform them first in data that fulfill these requirements. The simplest transformations are the linear ones, and if one is contented with uncorrelated data (and not independent), then the linear transformation of principal component analysis can do the job. Unfortunately, the transformed patterns are no longer binary, and it is an open problem to find a linear transformation that produces uncorrelated and binary data (see, e.g., Tang & Tao, 2006, or Schein, Saul, & Ungar, 2003),

an exact solution being in general impossible).[8] So to end up with binary data, one must yield to one of the constraints: uncorrelation or linearity of the transformation.

Here we abandon the request of a linear transformation, which allows us to focus on our other goal: we will produce data that are not just uncorrelated but independent, while at the same time remaining binary. More precisely, given $m$, $n$-dimensional, binary images $\vec{\xi}_\nu$, we look for

$$f : \mathbb{R}^n \to \mathbb{R}^h \quad \vec{y}_\nu = f(\vec{\xi}_\nu) \quad \text{with} \quad h \leq n.$$

The outputs $\vec{y}_\nu$ represent the preprocessed patterns that should be statistically independent and thus ready to be stored in an associative memory of $h$ neurons. At this point, it is clear that equation 2.2, obtained by NR, which satisfies equation 2.3, is tailored for the job.

Before presenting numerical results, we note an additional complication due to the fact that associative memories usually do not exactly recall the stored patterns $\vec{y}_\nu$ but return the pattern $\vec{S} = \vec{y}'_\nu$ with $\vec{y}'_\nu \simeq \vec{y}_\nu$, the difference being typically a few percent of the bits. If one wants to be able to get back the original image $\vec{\xi}_\nu$ from $\vec{y}'_\nu$, this imposes further requirements to the characteristics of the preprocessing and at the same time rules out standard algorithms for binary compression that produce statistically fragile data. As explained previously, NR, providing data that are as much further apart as possible in $R^h$, can fulfill this request.

We run a preliminary numerical test on a set of $m = 7$ binary images of $33 \times 33$ pixels. We had a network of $h = 64$ neurons with $n = 33 \times 33 + 1 = 1{,}090$ inputs totaling 69,760 weights. We run two different learning runs with the two gradient descent rules equations 5.1 and 5.2. The program stopped when $\max_{i,j}\{\Delta w_{ij}\} \leq 10^{-5}$ that required on the order of $10^7$ steps. Each simulation took several days of an Intel Core Duo 2.93 GHz processor, indicating that there is ample space for improvement (e.g., by taking advantage of standard electrostatics-relaxing algorithms).

Figure 3 shows the decrease in energy during learning for both the Euclidean $U$ and the Hamming distance $U_H$. The first impression is that, as one would expect, the decrease is compatible with a typical electrostatic potential and also, as foreseen, $U \geq U_H$. In this first run the, expected convergence of $U \to U_H$ was not observed, but there are indications that $U$ minimization was not terminated.

Our aim was to obtain both statistically independent and binary data. To check this last property is easier since we only have to check whether the patterns $\vec{y}_\nu$ rest on hypercube vertices. This can be seen from Figure 4, which shows a histogram of the values of coordinates $y_{\nu i}$ (obtained minimizing $U_H$), showing that this is true as expected.

[8] The covariance matrix has integer elements, but this is not true for its eigenvectors.

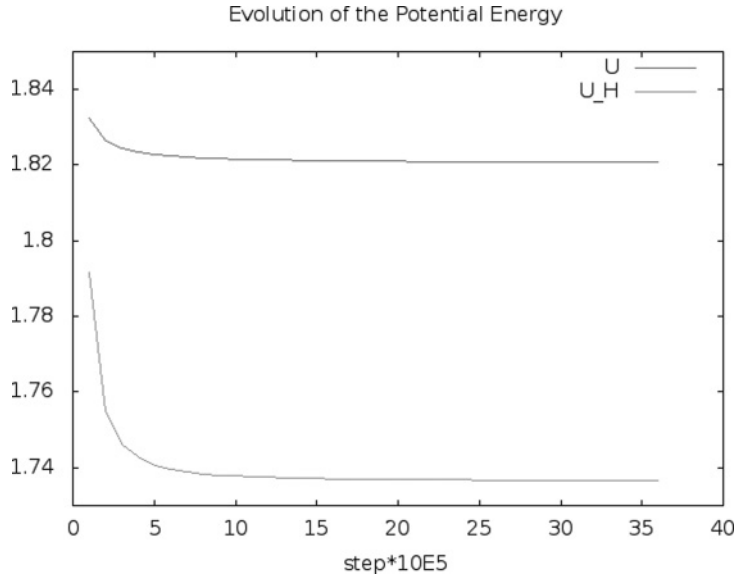


Figure 3: Behavior of the energy during the run for both the Euclidean $U$ and the Hamming distance $U_H$. As can be seen, the latter is smaller than the first, as predicted by equation 5.3. The $x$-axis represents the running step in units of $10^5$ elementary steps.

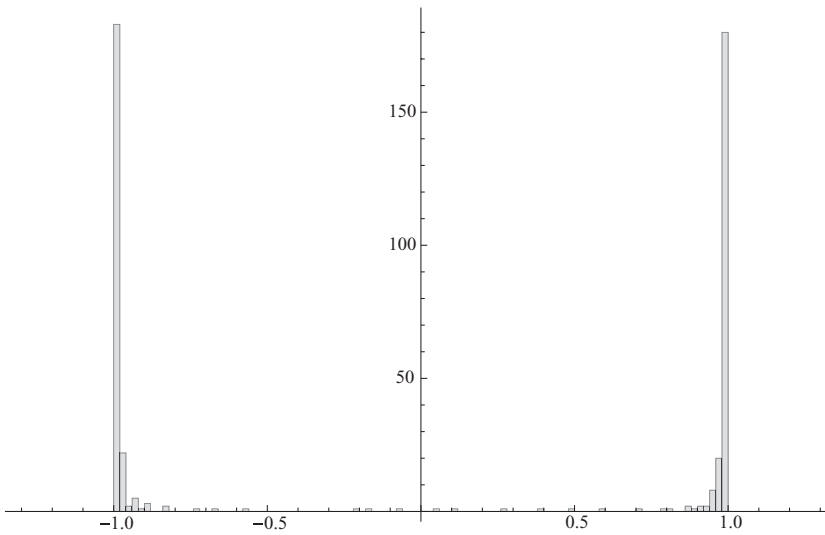


Figure 4: Histogram of the values of $y_{vi}$ coordinates showing that most of them are on hypercube vertices.

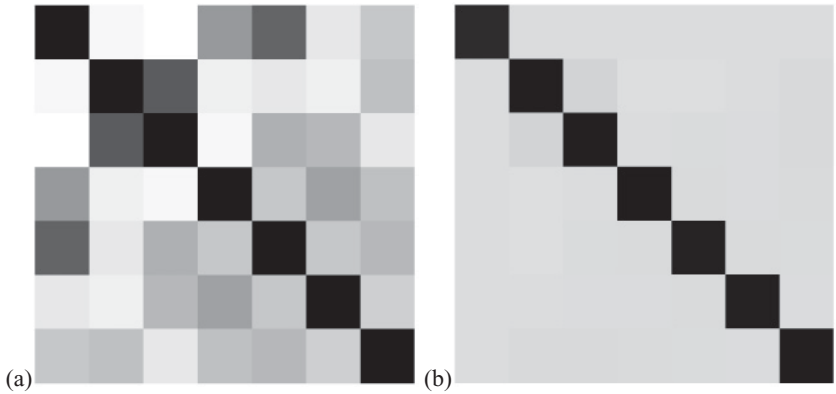


Figure 5: Matrices of scalar products $\vec{y}_\nu \cdot \vec{y}_\mu$ (converted to grayscale) for the systems defined by the Euclidean (a) and the Hamming (b) distance. The outputs are substantially equally spaced in both cases. From a computational point of view, it turned out that the NR version that made use of the Hamming distance converged faster than the other. This may suggest, as expected, that it succeeds in providing a greater gradient.

To verify the independence of data (see equation 2.3) with the reduced statistics of this simulation is a challenging task. A necessary condition is that the marginal distributions $p\left(y_i = \pm 1\right) = \frac{1}{2}$ (i.e., that each neuron cuts the input data set $\{\vec{x}_\nu\}$ in exactly two parts). In our simulation, this is perfectly achieved, since we got $\frac{m \times h}{2} = \frac{7 \times 64}{2} = 224$ positive coordinates and 224 negative ones. Moreover, each of the $h = 64$ output neurons has for the $m = 7$ inputs exactly three positive and four negative coordinates (or vice versa), suggesting that if we had a larger (and even) number of initial examples, we would get that each neuron would have $m/2$ positive and negative coordinates.

To investigate the quality of the solution, we analyzed the relative distances of the output data $\vec{y}_\nu$ since one can expect, once equation 3.4, has been minimized, that all relative distances should be equal, indicating a roughly constant hypersurface charge distribution. We did this calculating the $m \times m$ matrix of elements $\vec{y}_\nu \cdot \vec{y}_\mu$, that, when $\vec{y}_\nu$ sit on hypercube vertices, substantially represents the distance. In order to make it easier to understand, we converted these values to grayscale ($-h \rightarrow$ white, $h \rightarrow$ black). The result is shown in Figure 5. We can conclude that the $m$ outputs are substantially equally spaced, particularly in the second case.

## 7 Conclusion

We presented a new approach to the problem of data preprocessing by a layer of perceptrons. We treat each data vector as a point-like electric

charge confined in an $h$-dimensional hypercube, subject to simple Coulomb repulsive forces. We then let the system evolve as if it were a real physical system, that is, until it reaches the minimum of the electrostatic energy. At this point, we expect that the charges will occupy the hypercube's vertices and will be as far as possible from each other.

The potential energy function to minimize is continuous (since such is the transfer function $\tanh(x)$), well shaped, and, as far as we know, without the relative minima that plague so many cases in neural networks. For these reasons, in this case it is sensible to implement a simple gradient descent that produces a strictly local learning rule that is very similar to a Hebb rule, with the difference that to update a synapse, one needs all the data and not just the last seen one.

In our tests, this learning algorithm does not shine for its speed, but one can speculate that for actual calculations, one could use more refined minimization of the potential $U$, exploiting the relaxation techniques used routinely for similar electrostatic problems.

Even with a continuous transfer function at the end, one obtains binary and statistically independent data that guarantee that the entropy of the output is maximized.

Another characteristic of this network is that one can freely choose the number $h$ of output neurons without any adjustment of the learning algorithm. For small values of $h$, the network implements compression of the incoming data, and for larger $h$, just a dimensional reduction without any information loss. For even larger values of $h$, one introduces redundancy in the data, which are useful for subsequent error correction.

Despite some encouraging results, we feel that there still is ample space for further theoretical and computational developments.

## References


Atick, J. (1992). Could information theory provide an ecological theory of sensory processing? *Network: Computation in Neural Systems*, *3*(2), 213–251.

Axler, S., Bourdon, P., & Ramey, W. (2001). *Harmonic function theory* (2nd ed.). New York: Springer.

Bachmann, C. M., Cooper, L. N., Dembo, A., & Zeitouni, O. (1987). A relaxation model for memory with high storage density. *Proceedings of the National Academy of Sciences USA*, *84*(21), 7529–7531. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC299332/

Bell, A. J., & Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, *7*(6), 1129–1159.

Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory* (2nd ed.). New York: Wiley.

Hertz, J., Krogh, A., & Palmer, R. G. (1991). *Introduction to the theory of neural computation*. Reading, MA: Addison-Wesley.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366.

Jackson, J. D. (1999). *Classical electrodynamics* (3rd ed.). New York: Wiley.

Jackson, J. D. (2000). Charge density on thin straight wire, revisited. *American Journal of Physics*, *68*, 789–799.

Jackson, J. D. (2002). Charge density on a thin straight wire: The first visit. *American Journal of Physics*, *70*, 409–410.

Knoblauch, A. (2011). Neural associative memory with optimal Bayesian learning. *Neural Computation*, *23*, 1393–1451.

Marques, G. C., & Almeida, L. B. (1999). Separation of nonlinear mixtures using pattern repulsion. In *Proceedings of the First International Workshop on Independent Component Analysis and Signal Separation, ICA'99* (pp. 277–282). Norwell, MA: Kluwer.

Nadal, J., & Parga, N. (1993). Information processing by a perceptron in an unsupervised learning task. *Network: Computation in Neural Systems*, *4*, 295–312.

Nadal, J. P., & Parga, N. (1994). Non-linear neurons in the low noise limit: A factorial code maximises information transfer. *Network: Computation in Neural Systems*, *5*(4), 565–581.

Perrone, M. P., & Cooper, L. N. (1995). Coulomb potential learning. In M. A. Arbib (Ed.), *The handbook of brain theory and neural networks* (pp. 272–275). Cambridge, MA: MIT Press.

Pham, D. T. (2001). Contrast functions for blind separation and deconvolution of sources. In *Proceeding of ICA 2001 Conference*. Norwell, MA: Kluwer.

Schein, A. I., Saul, L. K., & Ungar, L. H. (2003). A generalized linear model for principal component analysis of binary data. In *Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics*.

Schwartz, R. E. (2010). *The five electron case of Thomson's problem.* http://www.math.brown.edu/∼res/papers/electron.pdf

Tang, F., & Tao, H. (2006). Binary principal component analysis. In *Proceedings of British Machine Vision Conference* (Vol 1, pp. 377–386). Malvern, UK: Author.

Theis, F. J., Bauer, C., Puntonet, C. G., & Lang, E. W. (2001). Pattern repulsion revisited. In J. Mira & A. Prieto (Eds.), *Bio-inspired applications of connectionism* (pp. 778–785). New York: Springer.

Thomson, J. J. (1904). On the structure of the atom: An investigation of the stability of the periods of oscillation of a number of corpuscles arranged at equal intervals around the circumference of a circle with application of the results to the theory of atomic structure. *Philosophical Magazine*, *7*(39), 237–265.

Willshaw, D. J., Buneman, O., & Longuet-Higgins, H. (1969). Non-holographic associative memory. *Nature*, *222*, 960–962.

Received February 10, 2012; accepted June 14, 2012.