Communicated by Fernando Pineda

Rotor Neurons: Basic Formalism and Dynamics

Lars Gislén
Carsten Peterson
Bo Söderberg
Department of Theoretical Physics, University of Lund,
Sölvegatan 14A, S-22362 Lund, Sweden

Rotor neurons are introduced to encode states living on the surface of a sphere in D dimensions. Such rotors can be regarded as continuous generalizations of binary (Ising) neurons. The corresponding mean field equations are derived, and phase transition properties based on linearized dynamics are given. The power of this approach is illustrated with an optimization problem—placing N identical charges on a sphere such that the overall repulsive energy is minimized. The rotor approach appears superior to other methods for this problem both with respect to solution quality and computational effort needed.

1 Background

Standard McCulloch–Pitts neurons are characterized by sigmoidal updating equations

$$v_i = g(u_i) = \tanh u_i \tag{1.1}$$

where the local field u_i is given by

$$u_i = \sum_j \omega_{ij} v_j / T \tag{1.2}$$

and the inverse temperature $1/T$ sets the gain. The neurons are binary in the high gain ($T \rightarrow 0$) limit. In feed-back networks (Hopfield and Tank 1985) with a quadratic energy function in terms of binary neurons s_i,

$$E = -\frac{1}{2} \sum_{i,j} \omega_{ij} s_i s_j \tag{1.3}$$

the iterative solutions of the mean field equations (equations 1.1 and 1.2) represent approximate minima to E for appropriate values of T, where $v_i =< s_i >_T$. In the more general case one has

$$u_i = -\frac{1}{T} \frac{\partial E}{\partial v_i} \tag{1.4}$$

Neural Computation 4, 737–745 (1992) © 1992 Massachusetts Institute of Technology

In a series of papers, we have investigated the generalization of this approach to multistate (Potts) neurons, which are superior in situations where one wants only one of the s_i (v_i) to be "on" and the others "off." In effect, equation 1.1 is replaced by[1]

$$v_i = \frac{e^{u_i}}{\sum_j e^{u_j}} \tag{1.5}$$

Such a constrained encoding turns out to be crucial for many optimization applications (Peterson and Söderberg 1989; Peterson 1990a). Potts neurons also play a crucial role in deformable templates or so-called elastic net methods (Durbin and Willshaw 1987). In feedforward networks with exclusive classifications Potts neuron encoding of the output layer could be profitable (Lönnblad *et al.* 1991).

In the present paper we investigate the generalization from binary neurons to the case of a continuum of states on a D-dimensional sphere, and apply the method to the problem of optimal configuration of charges on a sphere.

2 Rotor Neurons

Consider the general problem of minimizing an energy function $E(\mathbf{s}_1, \ldots, \mathbf{s}_N)$ with respect to a set of N D-dimensional unit vectors $\mathbf{s}_i$ (hereafter denoted *rotors*)

$$|\mathbf{s}_i| = 1 \tag{2.1}$$

A locally minimal configuration must satisfy

$$\mathbf{s}_i = -\nabla_i E/|\nabla_i E| \tag{2.2}$$

Local optimization consists of iterating these equations until convergence. This is in general not a very good method for finding the global minimum: the configurations easily get stuck in local minima. A more careful method is *simulated annealing* (Kirkpatrick *et al.* 1983), where a thermal distribution $\propto \exp(-E/T)$ is simulated. The temperature T is very slowly lowered, until a stable state results. This method is very time-consuming, if a good result is desired.

For this kind of problem, we suggest a *mean field theory* (MFT) rotor approach analogous to what is used for combinatorial optimization problems (Hopfield and Tank 1985; Peterson and Söderberg 1989).

2.1 Mean Field Equations. Consider a thermal distribution of configurations, characterized by the *partition function* Z,

$$Z = \int e^{-E[\mathbf{s}]/T} ds_1 \ldots ds_N \tag{2.3}$$

[1] Using [0,1] representation rather than the $[-1,1]$ of equation 1.1.

where the simplified notation ds_i means that the integration is to be performed only over the direction of $\mathbf{s}_i$, and normalized such that $\int ds_i = 1$.

For simplicity, consider first a single integral $I = \int H(\mathbf{s})\, ds$, over the directions of a D-dimensional unit vector $\mathbf{s}$, with H an arbitrary function. It can be rewritten as

$$\int H(\mathbf{s})\, ds = \int H(\mathbf{v})\delta(\mathbf{s}-\mathbf{v})\, ds\, d\mathbf{v} \propto \int H(\mathbf{v}) e^{\mathbf{u}\cdot(\mathbf{s}-\mathbf{v})}\, ds\, d\mathbf{v}\, d\mathbf{u} \tag{2.4}$$

Performing the $\mathbf{s}$ integral, one is left with

$$I \propto \int H(\mathbf{v}) e^{-\mathbf{v}\cdot\mathbf{u}+F(u)}\, d\mathbf{v}\, d\mathbf{u} \tag{2.5}$$

where $u = |\mathbf{u}|$ and $F(u)$ is defined by

$$F(u) = \log \int e^{\mathbf{u}\cdot\mathbf{s}}\, ds \tag{2.6}$$

For a D-dimensional integral, this evaluates to

$$F(u) = \log I_{(D-2)/2}(u) - \frac{D-2}{2}\log u + \text{const.} \tag{2.7}$$

where I_n are modified Bessel functions.

Repeating this trick N times with the multiple integral Z, we obtain

$$Z \propto \int \exp\left[-E[\mathbf{v}]/T - \sum_i \mathbf{v}_i\cdot\mathbf{u}_i + \sum_i F(u_i)\right]\, d\mathbf{v}_1\, d\mathbf{u}_1 \ldots d\mathbf{v}_N\, d\mathbf{u}_N \tag{2.8}$$

Next, we seek a saddlepoint of the effective potential, appearing in the argument of the exponent in the integrand in equation 2.8, by demanding its derivatives with respect to $\mathbf{u}_i$ and $\mathbf{v}_i$ to vanish. This results in the *mean field* equations

$$\begin{aligned} \mathbf{u}_i &= -\frac{1}{T}\nabla_i E[\mathbf{v}] && (2.9)\\ \mathbf{v}_i &= \mathbf{g}(\mathbf{u}_i) \equiv \hat{u}_i g(u_i) \equiv \hat{u}_i F'(u_i) && (2.10) \end{aligned}$$

where $\hat{u}_i = \mathbf{u}_i/u_i$. They give $\mathbf{v}_i$ as the average of $\mathbf{s}_i$ in the *local field* $\nabla_i E(\mathbf{v})$. In Table 1, $g(u)$ is listed for different values of D. When $D = 1$, equation 1.1 is recovered. The corresponding curves are shown in Figure 1. In the large D limit, the shape of g is given by

$$\lim_{D\to\infty} g(Du) = \frac{\sqrt{4u^2+1}-1}{2u} \tag{2.11}$$

We regard this system as a neural network, with $\mathbf{v}_i$ as a generalized neuron, $\mathbf{u}_i$ as its input (local field), and $\mathbf{g}$ as a generalized sigmoid function.

The obvious dynamics consists in iterating equations 2.9 and 2.10. The performance of the neural network thus defined will depend on T; this is to be discussed next.

Table 1: Proper $g(u)$ for Different Dimensions D.

D	$g(u)$	$g(u),\ u \to 0$	$g(u), u \to \infty$
1	$\tanh u$	u	$1 - 2e^{-2u}$
2	$I_1(u)/I_0(u)$	$u/2$	$1 - 1/2u$
3	$\coth u - 1/u$	$u/3$	$1 - 1/u$
4	$I_2(u)/I_1(u)$	$u/4$	$1 - 3/2u$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
D	$I_{D/2}(u)/I_{(D-2)/2}(u)$	u/D	$1 - (D-1)/2u$

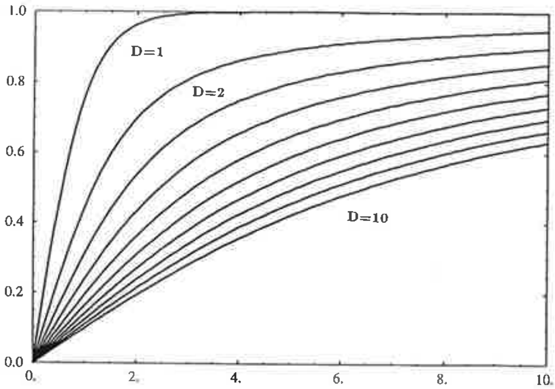

Figure 1: Graphs of $g(u)$ for different dimensions D.

2.2 Critical Temperature Estimation. From equation 2.10 we infer that for small u_i, $\mathbf{v}_i \approx \mathbf{u}_i/D$. Making the simplifying assumption that E is rotationally invariant, we can approximate E for small $\mathbf{v}$ with a quadratic form

$$E = E_0 - \frac{1}{2}\sum_{i,j} \omega_{ij}\mathbf{v}_i \cdot \mathbf{v}_j + O(v^2) \tag{2.12}$$

But then $\mathbf{v}_1 = \cdots = \mathbf{v}_N = 0$ is a fixpoint of the updating equations 2.9 and 2.10. Linearizing these for small v, we obtain

$$\mathbf{v}_i = \frac{1}{DT}\sum_{j} \omega_{ij}\mathbf{v}_j \tag{2.13}$$

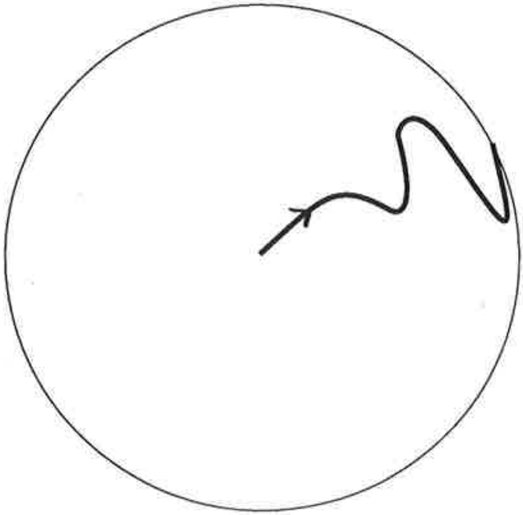

Figure 2: Schematic evolution of a $D = 2$ rotor initialized close to the center, for $T < T_c$.

If the temperature is higher than the *critical temperature*

$$T_c = \frac{1}{D}\max(\lambda_{\max}, -\lambda_{\min}) \tag{2.14}$$

where $\lambda_{\min/\max}$ are the extreme eigenvalues of ω, this trivial fixpoint is stable under *synchronous* updating, and the system is in a symmetric phase. For a lower T, it becomes unstable, and the mean fields $\mathbf{v}_i$ will be repelled by the origin. For low enough temperature they will stabilize close to the sphere $\mathbf{v}_i^2 = 1$ (cf. Fig. 2). The dynamics is thus very different from that of local optimization and simulated annealing, where moves take place on the surface.

For *serial* updating, things are similar, although T_c is different. In the special case of a constant self-coupling $\omega_{ii} = \mu$, we have instead

$$T_c = \frac{1}{D}\max(\lambda_{\max}, -\mu) \tag{2.15}$$

Thus, for a large set of energy functions, we can estimate T_c in advance. A good strategy is then to initialize the system close to the symmetry point, with T close to T_c, and slowly anneal while the system settles. When finally a stable state is reached, a possible solution to the minimization problem is extracted.

For both types of updating, it turns out to be advantageous to be able to adjust the self-coupling to achieve maximum stability. This is done by adding a term $-(\beta/2)\sum_i v_i^2$ to the energy. For a more detailed discussion of β and T_c, and of serial versus synchronous updating, the reader is referred to Peterson and Söderberg (1989).

3 Placing Charges on a Sphere

We now turn to the specific problem of the equilibrium configuration of N equal charges on a sphere ($D = 3$). With Coulomb interaction between the charges the energy function is given by

$$E = \frac{1}{2\sqrt{2}} \sum_{i \neq j} \frac{1}{(1 - \mathbf{s}_i \cdot \mathbf{s}_j)^{\frac{1}{2}}} - \frac{\beta}{2} \sum_i \mathbf{s}_i^2 \tag{3.1}$$

where we for dynamic reasons have added a β term as discussed above. With this energy function the local field $\mathbf{u}_i$ (cf. equation 2.9) is given by

$$\mathbf{u}_i = \frac{1}{T}\left[-\frac{1}{2\sqrt{2}} \sum_{i \neq j} \frac{\mathbf{v}_j}{(1 - \mathbf{v}_i \cdot \mathbf{v}_j)^{\frac{3}{2}}} + \beta \sum_i \mathbf{v}_i\right] \tag{3.2}$$

with the corresponding updating equation (see Table 1)

$$\mathbf{v}_i = \hat{u}_i(\coth u_i - 1/u_i) \tag{3.3}$$

The critical temperature T_c for this system in serial updating mode is, for reasonable β values (cf. equation 2.14)

$$T_c = (\beta + \tfrac{1}{2\sqrt{2}})/3 \tag{3.4}$$

The β-term also affects the low temperature behavior, controlling the tendency for the system to remain in a local minimum. A good final configuration, with the charges uniformly distributed over the sphere, should be stable. This means that the updated $\mathbf{u}_i$s should satisfy

$$\mathbf{u}_i(t) \cdot \mathbf{v}_i(t-1) > 0 \tag{3.5}$$

A necessary condition for accomplishing this can be derived for large N. The result is

$$\beta > \beta_0 \approx \frac{N^{3/2}}{4} \tag{3.6}$$

The role of β is thus twofold: it controls the phase transition *and* the dynamic behavior at low temperatures. Equipped with prior estimates of T_c and β_0, the algorithm for a given problem size can take the following "black box" form:

1. Compute T_c and β_0 according to equations 3.4 and 3.6.

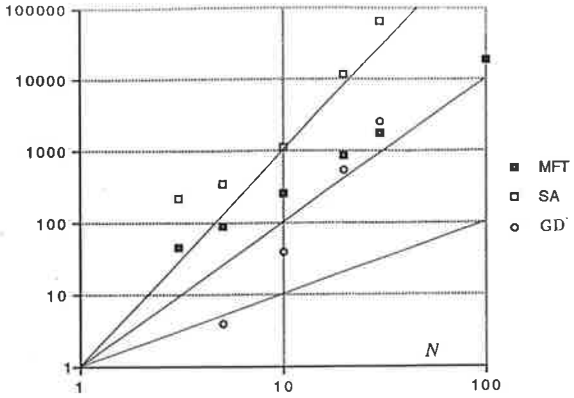

Figure 3: Time consumption as a function of problem size (N) for the MFT rotor (MFT), gradient descent (GD), and simulated annealing (SA) algorithms. Both axes are logarithmic. The three lines correspond to N, N^2, and N^3, respectively.

2. Initialize with $v_i^\alpha = 0.01 \cdot \text{rand}[-1, 1]$.
3. Update all $\mathbf{v}_i$s in turn according to equations 3.2 and 3.3.
4. Decrease the temperature, $T \to 0.95 \cdot T$.
5. Repeat from step 3, until the saturation (Peterson and Söderberg 1989) $\sum_i \mathbf{v}_i^2/N > 0.99$.
6. Extract configuration by setting $\mathbf{s}_i = \hat{v}_i$.

Using this prescription we have computed configurations of 3, 5, 10, 20, 30, and 100 charges, respectively. In Figure 3 the time consumption as a function of problem size is shown. As in the case of other MFT applications (Peterson and Söderberg 1989; Gislén *et al.* 1989, 1992), the number of iterations needed for convergence empirically shows no dependence on problem size. Hence, the time consumption scales as N^2 for the MFT rotor algorithm.

As for the quality of the solution, the MFT rotor model gives the correct solutions where these are explicitly known ($N = 2$, 3, 4, and 6). For larger problems we have compared the solutions with those from a gradient descent (GD) and a simulated annealing (SA) algorithm. In GD one charge at a time is moved on the sphere a step $\propto -\text{gradient}/N$. In SA, a maximum step size $\propto 1/N$ was used, with the annealing schedule $T \to 0.95 \cdot T$. These algorithms were clocked when the energy was within 1% of the MFT rotor result. The time consumption to achieve this is

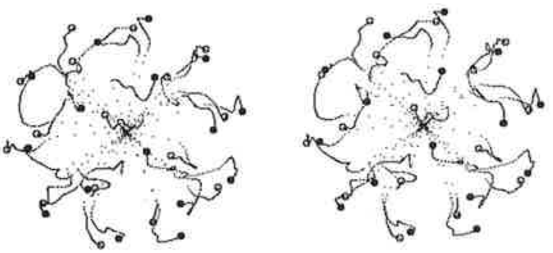

Figure 4: Evolutions of rotors for an $N = 32$ problem as T decreases. Open and filled dots represent charges placed in front and the back of the sphere, respectively. The two graphs are generated to provide a stereo effect.

shown in Figure 3. In Figure 4 the evolution of the rotors for an $N = 32$ problem is shown.

Comparing the MFT rotor approach with the conventional ones we find that for the MFT rotor algorithm the number of sweeps needed to reach a satisfactory solution is practically independent of problem size, while for the other methods it is (with optional step size) roughly proportional to the problem size. As for quality, the final energies obtained by the MFT rotor approach were always equal to or lower than the ones obtained with the other approaches.

We have also run the MFT rotor algorithm for a $D = 3$ system, where we substituted the appropriate sigmoid in equation 3.3 with a properly scaled tanh function

$$v_i = \hat{u}_i \tanh(u_i/3) \tag{3.7}$$

We find that the algorithm performs as well (if not better) with respect to the number of sweeps needed to obtain a good solution. The reason for this investigation is that this sigmoid is more natural in a VLSI implementation.

4 Summary

The formalism and dynamics for D-dimensional feedback rotor neurons have been developed. For $D = 1$ one recovers the conventional sigmoidal updating equations. As a first test bed for this approach in higher dimensions we applied the method to the problem of finding optimal charge configuration on a $D = 3$ sphere. The performance of the rotor method appears to be superior to that of gradient descent and simulated annealing for this problem.

Other potential problem areas of more practical use are, e.g., curve detection in the early vision system (Zucker *et al.* 1990), or the reconstruction of tracks from signals (Peterson 1990b). The $D > 1$ updating equations can of course also be used in feedforward multilayered networks.

References

Durbin, R., and Willshaw, G. 1987. An analogue approach to the travelling salesman problem using an elastic net method. *Nature (London)* **326**, 689.

Gislén, L., Peterson, C., and Söderberg, B. 1989. Teachers and classes with neural networks. *Int. J. Neural Syst.* **1**, 3.

Gislén, L., Peterson, C., and Söderberg, B. 1991. Scheduling high schools with neural networks. Lund university preprint LU TP 91-9 (to appear in *Neural Comp.*).

Hopfield, J. J., and Tank, D. W. 1985. Neural computation of decisions in optimization problems. *Biol. Cybern.* **52**, 141.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. 1983. Optimization by simulated annealing. *Science* **220**, 671.

Lönnblad, L., Peterson, C., and Rögnvaldsson, T. 1991. Using neural networks to identify jets. *Nuclear Phys. B* **349**, 675.

Peterson, C. 1990a. Parallel distributed approaches to combinatorial optimization. *Neural Comp.* **2**, 261.

Peterson, C. 1990b. Neural networks and high energy physics. In *Proceedings of International Workshop on Software Engineering, Artificial Intelligence and Expert Systems for High Energy and Nuclear Physics, Lyon Villeurbanne, France, March , 1990*, D. Perret-Gallix and W. Woijcik, eds. Editions du CRNS, Paris.

Peterson, C., and Söderberg, B. 1989. A new method for mapping optimization problems onto neural networks. *Int. J. Neural Syst.* **1**, 3.

Zucker, S. W., Dobbins, A., and Iverson, L. 1990. Two stages of curve detection suggest two styles of visual computation. *Neural Comp.* **1**, 68.

Received 17 September 1991; accepted 27 January 1992.