BOLTZMANN MACHINES : HIGH-ORDER INTERACTIONS AND SYNCHRONOUS LEARNING

Robert AZENCOTT(*)

École Normale Supérieure (DIAM)
and
Université Paris-Sud

1. INTRODUCTION

A now classical innovative paper [H.S.A.] by Hinton-Sejnowski-Ackley introduced a class of formal neural networks, the Boltzmann machines, governed by *asynchronous* stochastic dynamics, *quadratic* energy functions, and *pairwise* interactions defined by synaptic weights. One of the exciting aspects of [H.S.A.] was the derivation of a locally implementable learning rule linked to a scheme of decreasing (artificial) temperatures, in the spirit of simulated annealing.

However actual simulations of these machines for pattern classification problems have run into practical difficulties, the main one being the heavy load of computing time involved. Thus, the "neural network community" has often had a tendancy to consider the Boltzmann machines as useless slow learners.

We feel that these speed problems are enhanced by two facts, inherent to the

(*) CNRS labs : [LMENS] and [Stat. App. Univ. Paris-Sud]
Address : R. Azencott, ENS, 45 rue d'Ulm - F-75230 Paris Cedex 05
Tel. 33-1-47702465 - FAX : 33-1-48010754.

original [H.S.A.] presentation : at low temperature T, the stabilization times of these stochastic networks are *extremely long* (they grow exponentially with $1/T$), and moreover the *sequential update* of neurons, where only one neuron fires at a time is an obvious crucial cause of lengthy computations.

Hence we have been suggesting the use of learning at *constant temperatures* (suitably selected) and the implementation of highly *synchronous* neural updates, which in view of the availability of parallel hardware is quite natural in this context.

This raised the mathematical problem of devising suitable *synchronous learning rules*, which we have described in a recent paper [A], for the pairwise interactions situation. An interesting new feature was the natural emergence of *one-step delayed correlations* between the activities of pairs of neurons, as crucial indicators for weights updates.

However, in many low-level vision applications, the use of Markov field approaches instigated by D. and S. Geman [G.G], can be *reinterpreted in terms of sequential* Boltzmann machines for which *the energies are far more general than quadratic functions and for which cliques of interacting neurons have cardinals higher than three.* We have sketched the reinterpretation of Markov field approaches in the forthcoming paper [A]. Of course this requires the derivation of *new learning rules for sequential dynamics, general energies and high-order interactions.* The proper formal setup is described here and involves *a pair of dual networks* : the neural network S and the clique indicators network K. The synapses link *only* individual neurons to individual cliques, and synaptic weights w_c are indexed by cliques $C \in K$.

Our *sequential learning rules* for these generalized Boltzmann machines are still *locally implementable* ; they involve the *correlation between current clique activity* $J_C(X^n)$ *and the current score* $\lambda(X_R^n)$ of the machine response. We compute these scores through the introduction of fairly arbitrary *loss functions* which compare desired and current responses of the machine.

For *very specific and simple choices of loss function*, these multiple interaction, general energies, sequential learning rules include in particular a weight update Δw_c proportional to the difference in average clique activity $E(J_c)$ between two regimes

(clamped output and unclamped output) which generalizes by a quite different route the sequential learning rule derived in [H.S.A.] for quadratic energy functions and pairwise interactions.

Of course, it is quite tempting and natural to study these general Boltzmann machines in the synchronous case too, in order to gain important speed factors. It turns out that *synchronous dynamics with high-order interactions* involve a serious mathematical difficulty : *the equilibrium probability distribution M on the set of global neurons configurations cannot be computed explicitly* in general. Hence in the present paper, we use more sophisticated probabilistic tools to compute the gradient $\frac{\partial M}{\partial w_c}$, and to interpret the results at the empirical level.

We have thus obtained here *quite new learning rules for synchronous dynamics in presence of high-order interactions and general energies*. These rules are still *implementable locally*. For each clique C they involve *the sum of strings delayed correlations* between the *past clique unexpected transition activity* $u_c(X^{n-k-1}, X^{n-k})$ and *the current score* $\lambda(X_R^n)$ of the machine response. Here again scores involve general loss functions, computed at the level of response units and fed back to all clique indicators. For a clique C, the *transition activity* between successive configurations x and y is computed by $[\sum_{s \in C} J_c(x_{c-s}, y_s)]$, a quantity which appears naturally in our gradient computations. Finally we sketch a few results on the *choice of optimal temperatures* T, and obtain interesting physical and probabilistic interpretations of the gradient, with respect to T, of the expected score.

In [A], we have deduced from these synchronous learning rules a set of *algorithms for parameter estimation in synchronous Markov field model fitting*, an interesting new twist in Markov field approaches for low-level vision models.

The learning rules derived here are currently being tested experimentally in our research group DIAM at ENS (Paris), using simulations on computers offering high degrees of parallelization (Cray, connection machine). The possibilities of implementing generalized Boltzmann machines on specialized hardware are being currently evaluated in collaboration with P. Garda and other researchers at ENS Paris and I.E.F. Orsay. The vision application of synchronous Markov field models

are currently studied in collaboration with other researchers at ENS Paris (*cf.* recent work in collaboration with A. Doutrieux, L. Younès, J. Lacaille).

2. NEURAL NETWORKS WITH MULTIPLE INTERACTIONS

Consider an arbitrary finite set S of "formal neurons". The *state* x_s of neuron s will be a variable with values in an arbitrary finite set A. The *configuration* $x = (x_s)_{s \in S}$ of the network S is an arbitrary element of $\Omega = A^S$.

Fix an arbitrary family K of subsets of S, which will be called the set of *cliques* in the network S. The *activity* of any clique $C = \{s_1 \cdots s_p\} \in K$ will be measured by an *interaction potential*

$$J_C(x) = J_C(x_{s_1}, \cdots, x_{s_p}).$$

These interaction potentials are *arbitrary numerical functions of the clique configuration*

$$x_C = (x_{s_1}, \cdots, x_{s_p}).$$

For each clique $C \in K$, introduce a numerical parameter $w_c \in \mathbf{R}$ which will be called the *clique weight*. The *weight vector* $w = (w_c)_{c \in K}$ in $\mathbf{R}^K$ will parametrize the network architecture defined by $\{A, S, K, (J_C)_{C \in K}$.

Recall that widely used standard models of format neurons tend to consider only pairwise interactions, that is to say situations where all cliques have cardinal ≤ 2. On the other hand, Markov field theory and particularly its application to image analysis often involve cliques of cardinal > 3.

3. SEQUENTIAL STOCHASTIC DYNAMICS

Introduce now the *sequential energy function*

$$U(x) = \sum_{C \in K} w_C \, J_C(x) \tag{3.1}$$

which measures the *weighted activity of the cliques.* A classical Glauber type of sequential stochastic dynamics can now be defined to ensure that the sequential network has at equilibrium (in the long run) a Gibbs probability distribution on the set Ω of configurations, given by

$$G(x) = \frac{1}{Z} \exp\left[-\frac{U(x)}{T}\right] \tag{3.2}$$

$$Z = \sum_{y \in \Omega} \exp\left[-\frac{U(y)}{T}\right], \tag{3.3}$$

where Z is the *partition function*,and T is a positive fixed parameter called the *temperature.*

In the sequential dynamics, at each instant $n \in N$, only one of the neurons attempts to modify its state. Call s_n its index, which is generally preassigned by an arbitrary deterministic sequence $(s_1 \cdots s_n)$ visiting *periodically* all neurons $s \in S$. Such a sequence can also be random, provided *it is ergodic and uniformly distributed on S.* In either case, whenever the current random configuration $X^n = x$, and the neuron $s_n = s \in S$ is preassigned as seen above for a possible change of state at time n, then the new configuration X^{n+1} coincides with X^n at all neurons in $[S - s_n]$, and the conditional distribution of $X_{s_n}^{n+1}$ is given by :

$$P\left(X_{s_n}^{n+1} = a \,|\, X^n = x, X^{n-1}, \cdots, X^0\right) = G_s(a \,|\, x_{S-s}) \tag{3.4}$$

where $G_s(a \,|\, z)$ is the *conditional probability* $G(X_s = a \,|\, X_{S-s} = z)$ under the Gibbs distribution G defined in (3.2).

Define the set Ns of *neighbours of neurons* s as the set of all $t \in S$ such as $t \neq s$ and there is some clique $C \in K$ containing both s and t. Then the *local action potential* of configuration x at site s is defined by :

$$V_s(x) = V_s(x_s, x_{Ns}) = -\sum_{\{\text{cliques } C \text{ containing } s\}} w_C \, J_C(x) \tag{3.5}$$

and (3.1) classically yields :

$$G_s(a \,|\, x_{S-s}) = \frac{e^{V_s(a, x_{Ns})/T}}{\sum_{\alpha \in A} e^{V_s(\alpha, x_{Ns})/T}} \tag{3.6}$$

which shows that the random updating of neuron s is purely "*local*" since it involves only the configuration x restricted to s and its set of neighbours Ns.

As is well known, whatever the initial configuration X^0 of the network, the limit distribution is given by :

$$\lim_{n \to +\infty} P(X^n = x) = G(x) \tag{3.7}$$

where $G(x)$ is the Gibbs distribution (3.1). We point out that *this sequential dynamics takes place at fixed temperature.*

4. SYNCHRONOUS STOCHASTIC DYNAMICS

In view of the purely local computations involved in asynchronous random updates, it is quite tempting to *parallelize completely* the random updating, and hence to define *synchronous stochastic dynamics*

$$P_{\text{syn}}\left(X^{x+1} = y \,|\, X^n, X^{n-1}, \cdots, X^0\right) = \prod_{s \in S} G_s(y_s \,|\, X^n_{S-s})$$

where G is the Gibbs distribution (3.1) above. This simply means that all neurons $s \in S$ update their states with *simultaneous independent random choices*, each one of the individual random choices being governed by a conditional law computed exactly as in the sequential case, by (3.5).

Since X^n is obviously again an ergodic Markov chain on the state space Ω, *the limit distribution*

$$M(x) = \lim_{n \to +\infty} P_{\text{syn}}(X^n = x)) \tag{4.2}$$

exists and does not depend on the initial configurations X^0. However, *in general* M *does not coincide at all with the Gibbs distribution* G.

Of course M is the invariant distribution for P_{syn}, and hence calling $Q(x, y)$, $x \in \Omega$, $y \in \Omega$ the one step transition of P_{syn}, we have :

$$\sum_{x \in \Omega} M(x)\, Q(x, y) = M(y) \tag{4.3}$$

which will be as usual noted in matrix form

$$M\,Q = M. \tag{4.4}$$

Note also that formula (4.1) implies

$$Q(x,y) = \prod_{s\in S} G_s(y_s \,|\, x_{S-s}) \tag{4.5}$$

with $G_s(.\,|\,.)$ given by (3.6).

The main difficulty below will be the fact that in the generic case M is only known *implicitly* through equations (4.4) (4.5). In fact the only cases where M has been computed explicitly are those where all cliques have cardinal ≤ 2. We refer to our paper [A] for a detailed treatment of this important particular case, which of course includes the case of Boltzmann machines with quadratic energy introduced by Hinton-Sejnowski-Ackley. Let us recall one of our results from [A] :

4.6. THEOREM.— *Assume that all cliques $C \in K$ are of cardinal ≤ 2. Then the synchronous stationary measure M is given by*

$$M(x) = m \prod_{s\in S} \Big[\sum_{a\in A} \exp \frac{1}{T} V_s(a, x_{Ns})\Big] \tag{4.7}$$

where the constant m is determined by $\sum_{x\in\Omega} M(x) = 1$. Moreover M verifies the microbalance equations

$$M(x)\,Q(x,y) = M(y)\,Q(y,x). \tag{4.8}$$

In particular the doubly infinite stationary synchronous Markov chain $(Y^n)_{n\in\mathbb{Z}}$ is then *reversible* in the sense that

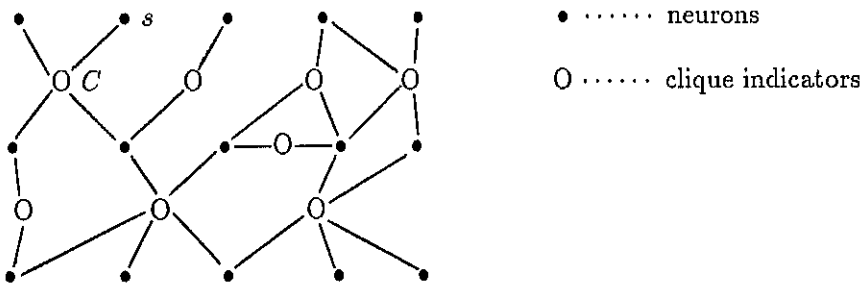
$$P_{\text{syn}}(Y^{n+1} = y \,|\, Y^n = x) = P_{\text{syn}}(Y^n = y \,|\, Y^{n+1} = x) = Q(x,y). \tag{4.9}$$

5. LOCAL IMPLEMENTATION OF THE STOCHASTIC DYNAMICS

5.1. A pair of dual networks

A convenient architecture is based on a pair (S, K) of dual networks, where S is the network of neurons, and K the network of *clique indicators*. To each clique $C \in K$ is associated *a cell called the indicator of clique* C, *whose state* y_C belongs to a finite subset of R, and is given by

$$y_C = J_C(x). \tag{5.2}$$



Example of a small network with cliques of various cardinals

5.2. Dynamics of the dual networks

The only *connections* in the pair (S, K) of networks are *links between one neuron* s *and one clique indicator* C. Namely *such a link exists if and only if* s *belongs to* C. At this stage such a connection is simply a transmissions channel. From s to C, this channel transmits the state x_s to the clique indicator C, which once it has gotten hold of all the $(x_s)_{s \in S}$ computes deterministically its state $y_C = J_C(x_C)$.

Conversely from C to s, the connection transmits the state y_C to the neuron s, which can then compute its action potential

$$U_s(x) = - \sum_{\text{C containing s}} w_C \, y_C$$

by a simple sum over all cliques connected to s. Then the neuron s can use $U_s(x)$ for its own random updating. Detailed setups for the effective computation of such an update will be proposed elsewhere in a more realistic context, and are easily imagined (see §9.12).

Hence on the pair (S, K) of dual networks, *the sequential and the synchronous stochastic dynamics are purely local.*

6. GENERALIZED BOLTZMANN MACHINES AS PATTERN CLASSIFIERS

Consider a stochastic network S with the above structure, defined by

$$\{A, S, K, (J_C)_{C\in K}\,,\, w = (w_C)_{C\in K}\}.$$

Select and fix *two arbitrary disjoint subsets D and R in S, the data units D* and *the response units R*. Their union $D \cup R$ will be the set of *visible units*, and the other neurons $H = S - (D \cup R)$ constitute the set of *hidden units*, to come back to a terminology introduced by Hinton-Sejnowski-Ackley [H.S.A.].

The environment provides on the data set D a family of inputs which are random configurations $d \in A^D$. To each input configuration $d \in A^D$, we want to associate a preassigned output configuration $r = F(d) \in A^R$ of the response units. The map $F : A^D \to A^R$ is for the moment assumed known to the supervisor of the learning process, at least on a " *training set*" also called a *set of examples*, which is simply a subset Γ of A^D.

The stochastic network will now be used to emulate F, assuming the weight vector w properly adjusted during a previous training period which will be studied further down.

To compute the response of the network to the input $d \in A^D$, the data units are *clamped* on the initial stimulus $X_D^0 = d$ while the rest of the network starts with an arbitrary initial configuration, and evolves freely according to one of the two

stochastic dynamics defined above (sequential or synchronous), *at fixed temperature* T. When stochastic equilibrium is (approximately) reached, the response units still have a *random configuration* $X_R^n \in A^R$, but the asymptotic distribution of X_R^n is well defined. When the data units D are clamped on the input $d \in A^D$, the network configuration x remains in the set

$$\Omega^d = \{x \in \Omega = A^S \,|\, x_D = d\} \tag{6.1}$$

which is of course isomorphic to the set A^{S-D} of configurations for the *reduced network* $(S - D)$. Then the *sequential equilibrium distribution for the (sequential) Markov chain* (X^n) *with clamped inputs* $\{X_D^n \equiv d \text{ for all } n \geq 0\}$ is the Gibbs measure on Ω^d

$$\begin{aligned} G^d(x) &= \frac{1}{Z^d} \exp\left[-\frac{1}{T} U(x)\right] \qquad \text{for all } x \in \Omega^d \\ Z^d &= \sum_{x \in \Omega^d} G^d(x). \end{aligned} \tag{6.2}$$

Clearly, $G^d(x)$ coincides with the conditional probability $G[X = x \,|\, X_D = d]$ for all $x \in \Omega^d$.

The *synchronous Markov chain* (X^n) *with clamped inputs* $X_D^n \equiv d$ for all $n \geq 0$, has *a stationary distribution* M^d *on* Ω^d, which is the unique solution of the matrix equation

$$M^d Q_d = M^d.$$

Here $Q_d(x, y)$ denotes, for $x, y \in \Omega^d$, the transition matrix of the synchronous chain (X^n) with clamped inputs $X_D^n \equiv d$. Hence we obviously have

$$Q_d(x, y) = \prod_{s \in S-D} G_s(y_s \,|\, x_{S-s}) \qquad \text{for all } x, y \in \Omega^d \tag{6.4}$$

which in view of (3.6) can be written explicitly as

$$Q_d(x, y) = \frac{1}{Z(x)} \exp \frac{1}{T} \Big[\sum_{s \in S-D} V_s(y_s, x_{Ns}] \qquad \text{for } x, y \in \Omega^d$$

with

$$Z(x) = \prod_{s \in S-D} \Big[\sum_{a \in A} \exp \frac{1}{T} V_s(a, x_{Ns}] .$$

The asymptotic marginal distribution of the output X_R^n for clamped inputs $X_D^n \equiv D \in A^d$ is then obtained by summing $M^d(x)$ (or $G^d(x)$ in the asymptotic case) over hidden configurations $x_H \in A^H$.

The response $\widehat{F}_n(d)$ of the machine to the input $D \in A^d$ will be the random configuration X_R^n of response units *with n large enough to ensure that stochastic equilibrium has practically been reached.* The pattern classifier $\widehat{F}_n : A^D \to A^R$ thus emulated by the machine is a random mapping. Note that again these classifiers $\widehat{F}_n$ are emulated at fixed temperature T. The choice of T will be evoked further below in §10.

The main problem in practical applications is to select the weight vector $w \in \mathbf{R}^K$ so that $\widehat{F}_n$ is *as close as possible to a preassigned classifier* $F : A^D \to A^R$.

To evaluate the performance of the machine as an emulator of F, we introduce *a loss function* $L(x_R, x'_R)$ for pairs of output configurations. Namely
$L : A^R \times A^R \to \mathbf{R}^+$ *is an arbitrary positive function, equal to zero whenever* $x_R = x'_R$.

Now for each configuration $x \in \Omega$, we can compute *the score* $\lambda(x)$ *of* x by

$$\lambda(x) = L[F(x_D), x_R)]. \tag{6.5}$$

A natural Bayesian point of view is to assume that *the environment provides inputs with a fixed a priori probability distribution p on A^D.* Les us point out that the actual knowledge of p will not be necessary below.

We introduce a random configuration Y with values in Ω having the distribution

$$P(Y = y) = p(y_D)\, G^{y_D}(y) \qquad \text{in the sequential case} \tag{6.6}$$

$$P_{\text{syn}}(Y = y) = p(y_D)\, M^{y_D}(y) \qquad \text{in the synchronous case} \tag{6.7}$$

To simulate Y, we use the *Markov chain X^n with clamped inputs* defined by the constraints

$$\begin{cases} X_D^n \equiv X_D^0 \\ \text{distribution } (X_D^0) \equiv p \end{cases} \tag{6.8}$$

and free stochastic dynamics on $S - D$; if the dynamics on $S - D$ remains sequential (resp. synchronous), the limit distribution of X^n is the corresponding distribution (6.6) [resp. (6.7)] of Y.

In this context, *the expected score*

$$\ell = E[(y)] = \lim_{n\to\infty} E[\lambda(X^n)] \tag{6.9}$$

is a natural quantity to *minimize* with respect to the weight vector $w \in \mathbf{R}^R$ which parametrizes the machine. If F_n is the classifier obtained by reading the response X_R^n of the machine at time n, one has obviously

$$\ell = \lim_{n\to\infty} E\, L[F(X_D^0), \widehat{F}_n(X_D^0)]. \tag{6.10}$$

With our previous notations, we have in *the synchronous case*

$$\ell = \sum_{x\in\Omega} p(x_D)\, M^{x_D}(x)\, \lambda(x) \tag{6.11}$$

while *in the sequential case* M^{x_D} is replaced by G^{x_D}.

The *learning rule* will simply be a gradient descent on the score function, of the following type : the k^{th} update of the weights will be given by

$$w^{k+1} - w^k = -\eta_k \frac{\partial \ell}{\partial w}(w^k) \tag{6.12}$$

where $\eta_k > 0$ is a *slowly decreasing* gain coefficient. We shall suggest here the choice $\eta_k = \frac{\alpha}{\beta+k}$ where *the coefficient* $\alpha > 0$, $\beta > 0$ *remain fixed during learning at fixed given temperature* T, and of course should be T-dependent if several temperatures are used. This slow decrease at speed (ℓ/k) for the gain is classical for gradient descent algorithms, and has the advantage of at least forcing convergence of the sequence (w^k) $k = 1, 2, \cdots$ whenever the sequence comes back infinitely often in the neighborhood of an isolated local minimum of the expected score ℓ.

Thus *for all practical purposes*, either the sequence w^k is unbounded, or the sequence $\ell(w^k)$ converges almost surely to a *local* minimum of ℓ.

The main problem is of course to compute $\frac{\partial \ell}{\partial w}$ through a *fast and parallel algorithm implementable on a dual neuronal architecture* of the type described in §5 above. In particular $\frac{\partial \ell}{\partial w_C}$ *should only involve computations confined to the neighborhood* NC *of clique* C.

Let us tackle first the much easier case of sequential dynamics.

7. LEARNING RULES FOR THE SEQUENTIAL CASE

The explicit formulas (3.1) (3.2) (3.3) for the energy U, the Gibbs distribution G, and the partition function Z immediately give

$$\frac{1}{Z}\frac{\partial Z}{\partial w_C} = -\frac{1}{T}E[J_C(x)] \qquad \text{for all cliques } C \in K \tag{7.1}$$

where X is a random configuration on S having the Gibbs distribution G, and then

$$-\frac{\partial G}{\partial w_C}(x) = \frac{1}{T}G(x)\,[J_C(x) - \overline{J}_C] \qquad \text{for all } x \in \Omega \tag{7.2}$$

where

$$\overline{J}_C = E[J_C(x)] = \sum_{x\in\Omega} J_C(x)\,G(x). \tag{7.3}$$

Let $d \in A^D$ be an arbitrary input, and call Ω^d the set of configurations x with clamped input $x_D = d$. Let G^d be the sequential Gibbs measure on Ω^d [*cf.* (6.1)), (6.2)]. We may now apply formulas (7.2) (7.3) to G^d. Hence we set for all $d \in A^D$

$$\overline{J}_C(d) = \sum_{\{x\in\Omega\,|\,x_D\equiv d\}} J_C(x)\,G^d(x). \tag{7.4}$$

and we define *the centered activity of clique* C by

$$j_C(x) = J_C(x) - \overline{J}_C(x_D) \qquad \text{for all } x \in \Omega. \tag{7.5}$$

Then formula (7.3) yields for all $d \in A^D$

$$-\frac{\partial G^d}{\partial w_C}(x) = \frac{1}{T}G^d(x)\,j_C(x) \text{ for } x \in \Omega \text{ and } x_D \equiv d. \tag{7.6}$$

From (6.11) we immediately get in the sequential case

$$\frac{\partial \ell}{\partial w_C} = \sum_{x\in\Omega} p(x_D)\,\lambda(x)\,\frac{\partial}{\partial w_C}\left[G^{x_D}(x)\right]. \tag{7.7}$$

and hence in view of (7.6)

$$-\frac{\partial \ell}{\partial w_C} = \frac{1}{T}\sum_{x\in\Omega} p(x_D)\,G^{x_D}(x)\,j_C(x)\,\lambda(x) \tag{7.8}$$

which is immediately interpretable as

$$-\frac{\partial \ell}{\partial w_C} = \frac{1}{T} E[j_C(Y)\,\lambda(Y)] \tag{7.9}$$

where the random configuration Y is defined by (6.6).

Since by construction

$$E[J_C(Y)\,|\,Y_D] = \overline{J}(Y_D)$$

we see that (7.9) has an interpretation in terms of *correlations*, by

$$-\frac{\partial \ell}{\partial w_C} = \frac{1}{T}\,\mathrm{cor}\,[j_C(Y), \lambda(Y)] = \frac{1}{T} E\{\mathrm{cor}_{Y_D}\,[J_C(Y), \lambda(Y)]\} \tag{7.10}$$

where for *all inputs* $d \in A^D$ *one defines* cor_d *as the correlation with respect to the conditional distribution of* Y *given* $\{Y_D = d\}$.

Using the sequential Markov chain X^n with clamped inputs introduced in (6.8), we have obviously

$$\mathrm{cor}_d\,[J_C(Y), \lambda(Y)] = \lim_{n\to\infty} \mathrm{cor}\,\{[J_C(X^n), \lambda(X^n)]\,|\,X_D^0 = d\} \tag{7.11}$$

and hence using the ergodicity of the chain X^n conditioned by $X_D^0 = d \in A^D$, we get

$$\mathrm{cor}_d\,[J_C(Y), \lambda(Y)] = \lim_{n\to\infty} \left[\frac{1}{n}\sum_{k=1}^{n} J_C(Z^n)\,\lambda(Z^n) - \overline{J}_{C,n}\,\overline{\lambda}_n\right] \tag{7.12}$$

where

$$\overline{J}_{C,n} = \frac{1}{n}\sum_{k=1}^{n} J_C(Z^n)$$

$$\overline{\lambda}_n = \overline{\lambda}_n = \frac{1}{n}\sum_{k=1}^{n} \lambda(Z^n)$$

(Z^n) is the sequential Markov chain on $(S - D)$ with clamped input

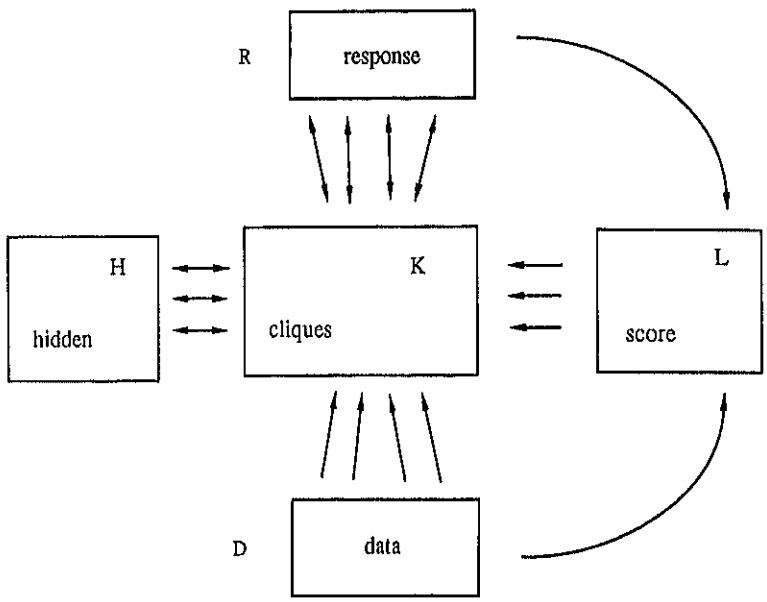
$Z_D^n \equiv d$ for all $n \geq 0$.

Call $\mathrm{cor}_d^n\,[J_C, \lambda)$ *the empirical correlation at time* n *between clique activity* J_C *and score* λ, *conditioned by* $Z_D^k \equiv d$, $k = 0, 1, \cdots, n$. Fix a random sample $\Gamma \subset A^D$

of inputs, *selected with the a priori distribution* p, which is equivalent to saying that *the training set* Γ *of inputs is generated by the environment*, and let $N = \mathrm{card}(\Gamma)$. Then (7.10) (7.12) yield the practical approximation

$$-\frac{\partial \ell}{\partial w_C} = \frac{1}{T} \lim_{N,n\to\infty} \frac{1}{N} \sum_{d\in\Gamma} \mathrm{cor}_d^n (J_C, \lambda). \tag{7.13}$$

This approximation is of course easy to implement locally through a *feedback from the response units to the clique indicator* C, which provides the locally computed score $\lambda(X^k) = L[F(X_D^k), X_R^k]$. This is summarized by the sketch below.



7.14. Communications between various computing blocks in the dual network with score feedback

In sketch 7.14, the block L computes the instantaneous score between the desired response $F(X_D)$ to the input X_D and the current machine response X_R, and then feeds back the score $\lambda(X) = L[F(X_D), X_R]$ to all cliques $C \in K$.

7.15. An important example of score function

Let us now apply the preceding result to a particular score function L_{lik} defined by

$$L_{lik}(r,r') = \begin{cases} 0 \text{ if } r = r' \\ 1 \text{ if } r \neq r' \end{cases} \qquad r,r' \in A^R. \tag{7.16}$$

Then the associated expected score ℓ_{lik} is given by

$$\ell_{lik} = P[Y_R \neq F(Y_D)] \tag{7.17}$$

where Y is the random configuration defined previously in §6, and hence minimizing ℓ_{lik} is equivalent to solving in $w \in \mathbf{R}^K$

$$\max_{w \in \mathbf{R}^K} P[Y_R = F(Y_D)].$$

In this case (7.9) becomes

$$-\frac{\partial \ell_{lik}}{\partial w_C} = \frac{1}{T} E\left[j_C(Y) 1_{Y_R} = F(y_D)\right]. \tag{7.19}$$

But *minimizing ℓ_{lik} is equivalent to maximizing*

$$\log(1 - \ell_{lik}) = \log P[Y_R = R(Y_D)] \tag{7.20}$$

and (7.19) implies

$$\frac{\partial}{\partial w_C}\left[\log(1-\ell_{lik})\right] = \frac{1}{T} E\left[j_C(Y) \,|\, Y_R = F(Y_D)\right]. \tag{7.21}$$

On the other hand, elementary manipulations and the definition $j_C(Y) = J_C(Y) - J_C(Y_D)$ yield

$$\begin{aligned} E\left[j_C(Y) \,|\, Y_R = F(Y_D)\right] &= E\left\{E[j_C(Y) \,|\, Y_D\,;\, Y_R = F(Y_D)]\right\} \\ &= E\left\{E[j_C(Y) \,|\, Y_D\,;\, Y_R = F(Y_D)] - \overline{J}_C(Y_D)\right] \end{aligned} \tag{7.22}$$

Recall that $\overline{J}_C(Y_D) = E[J_C(Y) \,|\, Y_D]$ to conclude that

$$\frac{\partial}{\partial w_C} \log P[Y_R = F(Y_D)] = \frac{1}{T} E\left\{J_C^1(Y_D) - J_C^2(Y_D)\right\}. \tag{7.23}$$

Here for each input $d \in A^D$,

(7.24) $J^1_C(d)$ is *the expected activity of clique C at stochastic equilibrium when the input Y_D remains clamped on d* and *the output Y_R remains clamped on $F(d)$,while the hidden neurons $s \in S-D-R$ evolve according to the sequential dynamics.* **(7.25)** $J^2_C(d)$ is *the expected activity of clique C at stochastic equilibrium when the input Y_D remains clamped on d while all the remaining neurons $s \in S - D - R$ evolve freely according to the sequential stochastic dynamics.*

Since the weight update Δw_C is proportional to $\frac{\partial}{\partial w_C} \log(1 - \ell_{lik})$, we see that **(7.26)** *Δw_C is propotional to the difference in average activity (for clique C) between two regimes : clamped output and free output.* In both regimes the data units remain clamped on the initial input $d \in A^D$, which should run through *a random training set $\Gamma \subset A^D$ "generated" by the environment*, in the sense specified above.

In this setup the expectations in (7.23) can correctly be estimated by the ordinary average on $d \in \Gamma$ and for each d and each regime by empirical averages of clique activity over time.

Of course the learning rule defined by (7.23) (7.24) (7.25) (7.26) generalizes directly the learning rule proposed by Hinton-Sejnovski-Ackley [H.S.A.] for sequential Boltzmann machines with quadratic energies, in which all cliques C contain only two neurons and $J_C(x) = -x_s x_t$ for $C = \{s,t\}$. However we see that the learning rule (7.26) is linked to a particularly rigid choice of loss function, and is only one example within the much wider family of learning rules which we have just derived here.

8. THE GRADIENT OF THE STATIONARY MEASURE IN THE SYNCHRONOUS CASE

In the particular case where all cliques in K have cardinal ≤ 2 a direct computation of the gradient M' of the synchronous stationary measure M, using the explicit expression (4.7), is possible and we have carried it out in our paper [A],

deriving from it several natural learning rules.

However we want to handle here the case of general stochastic networks with multiple interactions, for which M cannot be computed, and is known only implicitly as the unique solution of $MQ = M$. From this equation we get

$$M' = MQ' + M'Q. \tag{8.1}$$

An iterative use of (8.1) immediately yields

$$M' - M'Q^{n+1} = MQ'[I + Q + Q^2 + \cdots + Q^n]. \tag{8.2}$$

Now we have for all $x, y \in A^S = \Omega$

$$\lim_{n \to +\infty} Q^n(x, y) = M(y)$$

and hence

$$\lim_{n \to +\infty} M'Q^n(y) = \left[\sum_{x \in \Omega} M'(x) \right] M(y).$$

Since $\sum_{x \in \Omega} M(x) \equiv 1$, we get

$$\lim_{n \to +\infty} M'Q^n = 0 \tag{8.3}$$

and the following result

8.4 THEOREM.— *If the ergodic transition matrix Q depends smoothly on a parameter $w \in \mathbf{R}^K$, then the invariant probability measure M of Q also depends smoothly on w and we have*

$$M' = \lim_{n \to +\infty} MQ'(I + Q + \cdots + Q^n). \tag{8.5}$$

Call K_s *the set of cliques in* K *containing a given neuron* $S \in S$ and let

$$g(x_{Ns}) = \sum_{a \in A} \exp \frac{1}{T} \left[- \sum_{C \in K_s} w_C J_C(x_{C-s}, a) \right]. \tag{8.6}$$

From (3.5) (3.6), we get

$$G_s(y_s \mid x_{S-s}) = \frac{1}{g(x_{Ns})} \exp \frac{1}{T} \left[- \sum_{C \in K_s} w_C J_C(x_{C-s}, y_s) \right]. \tag{8.7}$$

And hence if C is an arbitrary fixed clique, *we have for* $s \in C$

$$(8.8) \qquad T \frac{\partial \log G_s}{\partial w_C}(y_s \,|\, x_{S-s}) = -J_C(x_{C-s}, y_s) + \sum_{a \in A} J_C(x_{C-s}, a)\, G_s(a \,|\, x_{S-s})$$

where the notation $J_C(x_{C-s}, a)$ stands short for $J_C(z)$ with $z_{C-s} = x_{C-s}$ and $z_s = a$. Of course we also have

$$(8.9) \qquad \frac{\partial \log G_s}{\partial w_C}(y_s \,|\, x_{S-s}) = 0 \qquad \text{for } s \notin C.$$

On the other hand, the relation (4.5) says that, in the synchronous dynamics,

$$(8.10) \qquad Q(x, y) = \prod_{s \in S} G_s(y_s \,|\, x_{S-s})$$

and hence we get in the synchronous case

$$(8.11) \qquad \frac{\partial \log Q}{\partial w_C}(x, y) = \sum_{s \in S} \frac{\partial \log G_s}{\partial w_C}(y_s \,|\, x_{S-s}).$$

In view of (8.8) the right-hand side of (8.11) depends only on x_{NC} and y_C where

$$(8.12) \qquad NC = \cup_{s \in C} Ns$$

is the neighborhood of clique C.

To give a probabilistic interpretation of the right-hand sides of (8.8) (8.11), call X^n, $n = 1, 2, \cdots$ the successive random configurations of the network S in the synchronous dynamics. Then clearly

$$(8.13) \qquad G_s(a \,|\, x_{S-s}) = P\left(X_s^{n+1} = a \,|\, X_{S-s}^n\right)$$

and hence

$$(8.14) \qquad \sum_{a \in A} J_C(x_{C-s}, a)\, G_s(a \,|\, x_{S-s}) = E\left[J_C\left(X_{C-s}^n, X_s^{n+1}\right) \,|\, X_{S-s}^n\right].$$

Introduce then the notion of *transition activity* $a_C(x, y)$ *for the clique* C, defined by

$$(8.15) \qquad a_C(x, y) = \sum_{s \in C} J_C(x_{C-s}, y_s).$$

Note that $a_C(x,y)$ depends only on x_C, y_C. The *expected transition activity at time n, for clique C, given the configuration* X^n, is then

$$\overline{a}_C(X^n) = E\left[a_C(X^n, X^{n+1}) \,|\, X^n\right] \tag{8.16}$$

where $\overline{a}_C(x)$ depends only on x_{NC} and is given by

$$\overline{a}_C(x) = \sum_{s \in C} \sum_{\beta \in A} J_{(x_{C-s}, \beta)}\, G_s(\beta \,|\, x_{S-s}). \tag{8.17}$$

Introduce now *centered transition activity* $[a_C(x,y) - \overline{a}_C(x)]$, which we prefer to call the *unexpected transition activity of clique C.*

$$u_C(x,y) = a_C(x,y) - \overline{a}_C(x). \tag{8.18}$$

Clearly $u_C(x,y)$ depends only on (x_{NC}) and hence is still a *local* notion.

We may now interpret (8.8) (8.11) to get

$$\frac{\partial Q(x,y)}{\partial w_C} = -\frac{1}{T}\, Q(x,y)\, u_C(x,y). \tag{8.19}$$

Using (8.5) (8.19), we now obtain for all $z \in \Omega$

$$-T\,\frac{\partial M}{\partial w_C}(z) = \sum_{k=0}^{+\infty} \sum_{x \in \Omega, y \in \Omega} M(x)\, u_C(x,y)\, Q(x,y)\, Q^k(y,z). \tag{8.20}$$

The following result gives a crucial probabilistic interpretation of (8.20).

8.21. THEOREM.— *Call* $(Y^n)_{n \in \mathbf{Z}}$ *the* doubly infinite stationary synchronous Markov chain *of network configurations, having the synchronous transition matrix Q, and such that every* Y^n *has the stationary synchronous distribution M. For every function* $f : \Omega \to \mathbf{R}$, *every clique* $C \in K$, *every* $n \in \mathbf{Z}$, *we have* :

$$-\sum_{z \in \Omega} f(x)\,\frac{\partial M}{\partial w_C}(z) = \frac{1}{T} \sum_{k=1}^{+\infty} \operatorname{cor}\left[f(Y^n), u_C(Y^{n-k}, Y^{n-k+1})\right] \tag{8.22}$$

where $u_C(x,y)$ *is the* unexpected transition activity *of clique C defined by* (8.18) *above.*

Proof.— Using (8.20), one gets immediately

$$-T \sum_{z \in \Omega} f(z) \frac{\partial M}{\partial w_C(z)} = \sum_{k=0}^{+\infty} E\left[u_C(Y^0, Y^1)\, f(Y^{k+1})\right]. \tag{8.23}$$

By construction, we clearly have

$$E\left[u_C(Y^0, Y^1)\right] = 0 \tag{8.24}$$

and hence

$$E\left[u_C(Y^0, Y^1)\, f(Y^{k+1})\right] = \mathrm{cor}\left[f(Y^{k+1}), u_C(Y^0, Y^1)\right]$$

a correlation which, in view of the stationary of (Y^n), coincides with

$$\mathrm{cor}\left[f(Y^n), u_C(Y^{n-k-1}, Y^{n-k})\right]$$

for arbitrary integers $n \in \mathbf{Z}$. This proves the equivalence between (8.23) and the announced formula (8.22).

From a practical point of view, $(Y^n)_{n \in \mathbf{Z}}$ cannot be simulated in general since M is unknown. However, *large segments of that chain can easily be approximately simulated to an arbitrary degree of accuracy.* Indeed, let $(X^n)_{n=0,1,2,\cdots}$ be the (easily simulated) sequence of random configurations obtained by synchronous stochastic dynamics with arbitrary initial configuration X^0. For N large, the sequence $(X^{N+n})_{n=0,1,2,\cdots}$ has finite joint distributions arbitrarily close to those of $(Y^{j+n})_{n=0,1,2,\cdots}$ for arbitrary fixed $j \in \mathbf{Z}$. This suggests several practical approximations of (8.22) based on (X^n).

Define first the *cumulative transition activity of clique C, between instants i and j*, with $i \leq j+1$, by

$$\mathrm{cum}_C(i,j) = \sum_{k=i}^{j-1} u_C(X^k, X^{k+1}). \tag{8.25}$$

Several practical approximations are summarized in the following theorem :

8.26. THEOREM.— *Let (X^n) $n = 0, 1, 2, \cdots$ be the chain of configurations obtained by synchronous dynamics with arbitrary initial configuration X^0. Let C be*

an arbitrary clique in K, w_C its weight, and cum_C *its cumulative transition activity* [*cf.* (8.25)]. *For every function $f : \Omega \to \mathbf{R}$, the following three limits exist*

$$\lim_{n,k\to+\infty} \mathrm{cor}\left[f(X^{n+k}), \mathrm{cum}_C(n, n+k)\right] \tag{8.27}$$

$$\lim_{n\to+\infty} \mathrm{cor}\left[f(X^{2n}), \mathrm{cum}_C(n, 2n)\right] \tag{8.28}$$

$$\lim_{n\to+\infty} \mathrm{cor}\left[f(X^{n}), \mathrm{cum}_C(0, n)\right] \tag{8.29}$$

and they coincide with $[-T \sum_{z\in\Omega} f(z) \frac{\partial M}{\partial w_C}(z)]$, *where M is the synchronous stationary distribution on Ω.*

Proof.— The proof of Th. 8.2 is an easy technical variation on the crucial formula (8.22) and details will be given elsewhere. In fact, it can also be shown that all limits (8.27) (8.28) (8.29) are *uniform* with respect to the clique $C \in K$, the function f, and the weight vector w provide $\|f\|_\infty$ and $\|w\|$ remain *bounded*. Speeds of convergence for those three limits can also be computed explicitly.

Of course in view of (8.24), correlations in (8.27) (8.28) (8.29) may be replaced by expectations. Also, using the ergodicity of X^n, *these correlations can be correctly estimated by long time averages* such as, for (8.27),

$$\lim_{q\to\infty} \frac{1}{q} \sum_{j=1}^{q} f(X^{n+j+k})\, \mathrm{cum}_C(n+j, n+j+k) \tag{8.30}$$

with a similar expression for (8.28).

9. LEARNING RULES IN THE SYNCHRONOUS GENERAL BOLTZMANN MACHINE

We place ourselves in the general case of an energy function involving interactions of arbitrary orders and with synchronous dynamics. For the simpler case of synchronous machines with only pairwise interaction, we refer to [A].

Formula (6.11) for the expected score implies

$$\frac{\partial \ell}{\partial w_C} = \sum_{x \in \Omega} p(x_D) \left[\frac{\partial}{\partial w_C} M^{x_D}(x) \right] \lambda(x). \tag{9.1}$$

To compute $\left[\frac{\partial}{\partial w_C} M^{x_D}(x)\right]$ for $x_D = d \in A^D$, we can apply the results of §8 to the reduced network $S - D$, with energy function $U^d(x_{S-D}) \equiv U(x)$ for all x such that $x_D = d$, and synchronous stochastic dynamics on $S - D$.

Call X^n the *synchronous Markov chain with clamped inputs* defined in (6.8) by

$$X_D^n \equiv X_D^0 \qquad \text{and} \qquad \{\text{distribution } X_D^0 \equiv p\}. \tag{9.2}$$

To the notion of transition activity a_C of clique C, we have associated its expected, unexpected, and cumulative versions $\overline{a}_C$, u_C, cum_C. *Since the basic network is here reduced to $S - D$, we introduce the reduced versions of a_C, $\overline{a}_C$, u_C, cum_C, defined by*

$$a_C(x, y) = \sum_{s \in C \cap (S-D)} J_C(x_{C-s}, y_s) \tag{9.3}$$

$$\overline{a}_C(x) = \sum_{s \in C \cap (S-D)} \sum_{\alpha \in A} J_C(x_{C-s}, \alpha)\, G_s(\alpha \,|\, x_{S-s}) \tag{9.4}$$

$$u_C(x, y) = a_C(x, y) - \overline{a}_C(x) \tag{9.5}$$

$$\text{cum}_C(i, j) = \sum_{k=i}^{j-1} u_C(X^k, X^{k+1}). \tag{9.6}$$

The direct application of (8.27) yields for each input $d \in A^D$

$$-\sum_{\left[x_{S-D} \in A^{S-D}, x_D \equiv d\right]} \lambda(x) \frac{\partial M^d}{\partial w_C}(x) = \frac{1}{T} \lim_{n,k \to \infty} \text{cor}_d \left[\lambda(X^{n+k}), \text{cum}_C(n, n+k)\right] \tag{9.7}$$

where as in (7.10), $\text{cor}_d(V, W)$ denotes *the correlation of V and W with respect to the conditional distribution given $X_D^0 = d$.*

Using (9.1) we then obtain

$$(9.8) \qquad -\frac{\partial \ell}{\partial w_C} = \frac{1}{T} \lim_{n,k\to\infty} E\left\{ \mathrm{cor}_{X_D^0}\left[\lambda(X^{n+k}); \mathrm{cum}_C(n, n+k)\right] \right\}$$

and as in §8, *the correlations given* X_D^0 *may be replaced in* (9.8) *by expectations given* X_D^0 *as well as empirical correlations given* X_D^0. From (9.8), we now deduce a family of synchronous learning rules.

9.9. Approximate learning rules for the synchronous case

Consider a general Boltzmann machine with multiple interactions and synchronous dynamics. Fix a loss function L on pairs of outputs and a desired input-output mapping $F : A^D \to A^R$. Let $\lambda(x) = L[F(x_D), x_R]$ be the associated score of configuration $x \in \Omega$.

Fix a *training set* Γ *of inputs, generated by the environment*, so that Γ is a finite random sample of the real life *a priori* distribution p of inputs. *Fix the temperature* T. Choose then two "large" integers n, k and two positive parameters α, β. To n, k, Γ, α, β, we now associate *an approximate learning rule* LR for which the q^{th} update of weight w_C, $C \in K$, is defined by

$$(9.10) \qquad \Delta w_C = \frac{1}{\mathrm{Card}\,\Gamma} \sum_{d\in\Gamma} \Delta w_C(d)$$

$$(9.11) \qquad \Delta w_C(d) = \frac{\alpha}{\beta + q}\left[\frac{1}{n} \sum_{j=1+k}^{n+k} \lambda(X^j)\,\mathrm{cum}_C(j-k, j)\right]$$

where for all j in (9.11) the Markov chain X^j has clamped inputs $X_D \equiv d$ and synchronous dynamics on $(S - D)$. As in 9.6, cum_C is the (reduced) cumulative transitional activity of clique C over a past of length k.

Then for large n, k, and card(Γ), the learning rule LR will tend to achieve local minimization of the expected score ℓ.

We now give another interpretation of the synchronous learning rules LR. Indeed the empirical correlation between score and past cumulative transition activity

can be rewritten as

$$\sum_{i=1}^{k}\left[\frac{1}{n}\sum_{j=1+k}^{n+k}\lambda(X^j)\,u_C(X^{j-i},X^{j-i+1})\right]$$

which is *the sum of k empirical correlations between score and delayed unexpected transition activity.* This points out the importance of *delays* in automatic learning processes for synchronous networks. As easily seen, *for i large, the correlation between $\lambda(X^j)$ and $u_C(X^{j-i},X^{j-i+1})$ tends to zero at exponential speed*, and hence in many situations, only a fairly moderate number k of delays are significant.

9.12. Implementation of synchronous learning rules

Just as in the asynchronous case, the learning rule (9.10) (9.11) requires a feedback of the score $\lambda(X^n)$ to all clique indicators $C \in K$. Thus the communication scheme is similar to the sketch 7.15 described above.

Consider the pair (S,K) of dual networks introduced in Section 5, *fix an input $d \in A^D$*, and call X^n the configuration of neurons at time n. The flow of parallel computations — communications between configurations n and $n+1$ can be roughly described as follows, *the data units D remaining clamped on the input $d \in A^D$.*

(9.13) At the end of period n, the memory of each clique C has stored precisely

- the local configuration X_C^n
- the past unexpected transitional activities

$$u_C^j = u_C(X^{j-1},X^j) \qquad j = n, n-1, \cdots, n-k+1$$

- the cumulative transitional activity

$$\mathrm{cum}_C^n = \sum_{j=n-k+1}^{n} u_C^j$$

- the empirical correlation

$$n\,\mathrm{cor}_C^n = \sum_{j=1}^{n}\lambda(X^j)\,\mathrm{cum}_C^j.$$

(9.14) For each possible individual neural state $a \in A$, execute the following loop of *parallel* computations

9.14.1. each clique C computes for each $s \in C$ the lateral activity

$$\gamma_C(s,a) = J_C(X^n_{C-s}, a)$$

and transmits $\gamma_C(s,a)$ to the neuron $s \in C$.

9.14.2. every neuron $s \in S - D$ computes

$$\pi_s(a) = \exp -\frac{1}{T} \sum_{C \ni s} w_C \, \gamma_C(s,a)$$

and transmits $\pi_s(a)$ to all cliques C containing s.

9.14.3. - if A is not exhausted, go back to the beginning of loop (9.14)

- if A is exhausted, go to (9.15).

(9.15) Every neuron $s \in S - D$ computes a (random) state X^{n+1}_s with distribution $P(X^{n+1}_s = a) = \frac{\pi_s(a)}{\pi_s}$, where $\pi_s = \sum_{a \in A} \pi_s(a)$, and transmits X^{n+1}_s to all cliques C containing s.

(9.16) The score $\lambda^{n+1} = \lambda(X^{n+1})$ is computed in the L-block (*cf.* 7.15) and fed back to all cliques C.

(9.17) Every clique C computes successively

- $\pi_s = \sum_{a \in A} \pi_s(a)$ for all $s \in C \cap (S - D)$
- $\overline{a}_C(X^n) = \sum_{s \in C \cap (S-D)} \sum_{a \in A} \gamma_C(s,a) \frac{\pi_s(a)}{\pi_s}$
- $a_C(X^n, X^{n+1}) = \sum_{s \in C \cap (S-D)} J_C\big(X^n_{C-s}, X^{n+1}_s\big)$
- $u^{n+1}_C = a_C(X^n_{C-s}, X^{n+1}) - \overline{a}_C(X^n)$
- $\mathrm{cum}^{n+1}_C = \mathrm{cum}^n_C + u^{n+1}_C - u^{n+1-k}_C$
- $(n+1)\,\mathrm{cor}^{n+1}_C = n\,\mathrm{cor}^n_C + \lambda^{n+1}\,\mathrm{cum}^{n+1}_C$

During the learning phase, for each input d in the training set Γ, the algorithm 9.13) $\cdots$ (9.17) is iterated with clamped input,until the correlations cor^n_C stabilize with an upper bound on n of course). Call $\mathrm{cor}_{d,C}$ this limit correlation.

The weight vector w will be updated after each complete pass of the training set Γ. If the current pass corresponds to the q^{th}-update of w, this update Δw_C is given by

$$\Delta w_C = \frac{\alpha}{\beta + q} \frac{1}{\text{Card}(\Gamma)} \sum_{d \in \Gamma} \text{cor}_{d,C}. \tag{9.18}$$

Recall that *we assume the training set Γ of inputs to be a finite random sample "generated" by the environment.* This last point is crucial for the validity of (9.18) and has often been overlooked in the neural network literature. As pointed out by Bourlard [BO] in another context, *the empirical a priori distribution exhibited by the training set should be close to the a priori distribution of real-life inputs.*

10. OPTIMAL CHOICE OF THE TEMPERATURE

As was pointed out several times, the general Boltzmann machines considered here are meant to operate at *fixed temperature T.* However T is a natural and important parameter, and hence we may try to optimize the choice of T in the learning phase. Note however *a crucial point*: since the machine is actually parametrized by $\left[\frac{w}{T}\right]$, *no generality is lost in principle* if $T \equiv 1$.

Introduce as before the desired input-output mapping $F : A^D \to A^R$ and a loss function $L : A^R \times A^R \to \mathbf{R}^+$; call $\ell = \lim_{n \to \infty} E[\lambda(X^n)]$ the expected score. To choose an optimal T, it is natural to use gradient descent to *minimize* ℓ in $T \in \mathbf{R}^+$. Thus after each complete pass of the training set $\{d^1 \cdots d^n\}$ of random inputs (provided by the environment), we may introduce a temperature update

$$\Delta T \quad \text{proportional to} \quad -\frac{\partial \ell}{\partial T} \tag{10.1}$$

with small decreasing gain as for Δw.

The computation of $\frac{\partial \ell}{\partial T}$ is an easy consequence of the computation of $\frac{\partial \ell}{\partial w}$.

10.2 PROPOSITION.— *Consider a general Boltzmann machine with either synchronous or sequential dynamics. Select a learning rule as in §7 and 9 above for the*

veight vector w, and call Δw the corresponding weight update after one complete ass of the training set. The gradient descent in temperature is then given by

10.3) $$\Delta T = -\frac{1}{T}\left[\sum_{C\in K} w_C\,\Delta w_C\right] = -\frac{1}{2T}\,\Delta(\|w\|^2).$$

ndeed one has for the expected score ℓ

10.4) $$\frac{\partial \ell}{\partial T} = -\frac{1}{T}\left[\sum_{C\in K} w_C\,\frac{\partial \ell}{\partial w_C}\right].$$

roof.— For any function $g : \mathbf{R}^K \to \mathbf{R}$, one has trivially, if $f(w,T) = g\left(\frac{w}{T}\right)$

$$\frac{\partial v}{\partial w} = \frac{1}{T}\,g'\left(\frac{w}{T}\right) \quad \text{et} \quad \frac{\partial f}{\partial T} = -\frac{1}{T^2}\left\langle w\,.\,g'\left(\frac{w}{T}\right)\right\rangle$$

hich proves immediately (10.4) and hence (10.3).

0.5 PROPOSITION.— *Consider a general* sequential *Boltzmann machine with ıergy function U. Let (X^n) $n = 0,1,\cdots$ be the sequential Markov chain for which $^n_D \equiv X^0_D$ for all n, and X^0_D has the a priori distribution of inputs generated by the ıvironment. Let $\ell = \lim_{n\to\infty} E[\lambda(X^n)]$ be the expected score. We then have*

0.6) $$\frac{\partial \ell}{\partial T} = \frac{1}{T^2}\lim_{n\to\infty} E\left\{\mathrm{cor}_{X^0_D}[U(X^n),\lambda(X^n)]\right\}$$

here as in 7.10, $\mathrm{cor}_{X^0_D}$ *denotes the conditional correlation given $X^n_D \equiv \cdots \equiv X^0_D$.*

roof.— Call Y the random configuration defined by (6.6). From formula (7.10) and 0.4), we get immediately

0.7) $$\frac{\partial \ell}{\partial T} = \frac{1}{T^2}\,E\left\{\mathrm{cor}_{Y_D}[U(Y),\lambda(Y)]\right\}$$

ıce $U(y) \equiv \sum_{C\in K} w_C\,J_C(y)$. Then (10.6) is a direct consequence of (10.7).

0.8. Another probabilistic interpretation

Let G be the Gibbs distribution on Ω associated to energy U at temperature Since $\log G(y) = -\frac{U(y)}{T}$

$$\log G(y) = -\frac{1}{T}\,U(y) - \log Z$$

where $\log Z$ is a constant, we get the interesting interpretation

$$\frac{\partial \ell}{\partial T} = -\frac{1}{T} E\left\{ \mathrm{cor}_{Y_D} [\log G(Y), \lambda(Y)] \right\} \tag{10.9}$$

so that the sequential temperature update ΔT should be proportional to the average correlation between the log likelihood and a random configuration and its score.

10.10. Global transition energy for synchronous networks

Consider now a synchronous Boltzmann machine with energy $U = \sum_{C \in K} w_C J_C$ and data units D. In (9.3) (9.4) (9.5), we have defined the notions (reduced to $S-D$) of clique transition activities $a_C(x,y)$ as well as their expected and unexpected versions $a_C(x)$, $u_C(x,y)$.

We now introduce three global quantities, the *transition energies* $a(x,y)$, the *expected transition energy* $a(x)$, and the *unexpected transition energy* $u(x,y)$ defined for pairs $x, y \in \Omega$ of global configurations, by

$$a(x,y) = \sum_{C \in K} w_C \, a_C(x,y) \tag{10.11}$$

$$\overline{a}(x) = \sum_{C \in K} w_C \, \overline{a}_C(x) \tag{10.12}$$

$$u(x,y) = \sum_{C \in K} w_C \, u_C(x,y) \tag{10.13}$$

with a_C, $\overline{a}_C$, u_C given by (9.3) (9.4) (9.5).

It turns out that a, $\overline{a}$, u have interesting global interpretations in terms of the *synchronous Markov chain* (X^n) *on* S *with clamped inputs on* D which verifies $X_D^n \equiv X_D^0$ for all $n \geq 0$. For each input $d \in A^D$, let

$$\Omega^d = \left\{ x \in \Omega = A^X \,|\, x_D = d \right\} \tag{10.14}$$

and for $x, y \in \Omega$, call $Q_d(x,y)$ the transition matrix of X^n given $X_D^0 \equiv d$.

From definitions (9.3) (9.10) (3.5), we get for $x, y \in \Omega$

$$a(x,y) = -\sum_{s \in S-D} V_s(y_s, x_{Ns}) \tag{10.15}$$

where the V_s are the local action potentials (*cf.* (3.5)). In view of the explicit expression (6.4) of $Q_d(x,y)$, from (10.15) we deduce *for* $x, y \in \Omega_d$

$$-\frac{1}{T}\, a(x,y) = \log\, Q_d(x,y) + \log\, Z_d(x) \tag{10.16}$$

where $Z_d(x)$ has been defined in (6.2).

On the other hand, (9.4) (10.12) and (6.4) give the interpretation of $a(x)$ as a conditional expectation

$$\overline{a}(x) = \sum_{y \in \Omega_d} Q_d(x,y)\, a(x,y) \qquad \text{for } x \in \Omega_d \tag{10.17}$$

Now a comparison of (10.13) (10.16) (10.17) provides trivially a new interpretation of $u(x,y)$ for $x, y \in \Omega_d$

$$-\frac{1}{T}\, u(x,y) = \log\, Q_d(x,y) - \sum_{z \in \Omega_d} Q_d(x,z)\, \log\, Q_d(x,z). \tag{10.18}$$

Thus we now introduce the *entropy* $\mathrm{ent}_d(x)$ *of the transition distribution* $Q_d(x,.)$ classically defined by

$$\mathrm{ent}_d(x) = -\sum_{z \in \Omega_d} Q_d(x,y)\, \log\, Q_d(x,y) \tag{10.19}$$

and interpret (10.18) in the following statement

10.20 PROPOSITION.— *For an arbitrary input* $d \in A^D$, *let* $Q_d(x,y)$, $x, y \in \Omega_d$ *be the transition matrix of the synchronous chain* (X^n) *with clamped inputs* $X_D^n \equiv d$. *Let* $\mathrm{ent}_d(x)$ *be the entropy of* $Q_d(x,.)$. *Then the global unexpected transition energy* $u = \sum_{C \in K} w_C\, u_C$ *of the network is given by*

$$\frac{1}{T}\, u(x,y) = -\log\, Q_d(x,y) - \mathrm{ent}_d(x) \qquad \text{for all } x, y \in \Omega_d. \tag{10.21}$$

We now introduce the *cumulative transition in energy* of the network along a sequence $(X^i X^{i+1} \dots X^j)$ random configurations, defined by

$$\text{cum}(i,j) = \sum_{k=i}^{j-1} u((X^k X^{k+1}) = \sum_{C \in K} w_C \, \text{cum}_C(i,j). \tag{10.22}$$

In view of (10.21), *we see that* $\left[\frac{1}{T} \text{cum}(i,j)\right]$ *can roughly be interpreted as a centered version of the joint log likelihood of the sequence* $(X^i X^{i+1} \dots X^j)$ *given* $X_D^0 \equiv d$. Of course, from a practical point of view, the *actual computation* of $\text{cum}(i,j)$ is far easier to obtain by formula $\sum_C w_C \, \text{cum}_c(i,j)$ since the $\text{cum}_c(i,j)$ are locally computed by each clique C during the learning process. We may now interpret $\frac{\partial \ell}{\partial T}$ in the synchronous case.

10.23 PROPOSITION.— *Consider a general synchronous Boltzmann machine, with data units D. Fix a desired mapping F, a loss function L and let $\lambda(x)$ be the associated score function. Let (X^n) be the synchronous Markov chain with clamped inputs $X_D^n \equiv X_D^0$ generated by the environment with a fixed a priori distribution of inputs. Then the expected score $\ell = \lim_{n \to \infty} E[\lambda(X^n)]$ verifies*

$$\frac{\partial \ell}{\partial T} = \frac{1}{T} \lim_{n,k \to \infty} E\left\{ \text{cor}_{X_D^0} \left[\lambda(X^{n+k}), \text{cum}(n, n+k) \right] \right\} \tag{10.24}$$

where $\text{cum}(n, n+k)$ *is the cumulative transition energy along the sequence* $(X^n X^{n+1} \dots X^{n+k})$ *defined by* (10.20) (10.22). *As before,* cor_d *denotes conditional correlation given* $X_D^0 \equiv d$.

Proof.— Formula (10.24) is a direct consequence of (10.4) and (9.8).

10.25. Interpretation

Note that in view of (10.21) (10.24) (10.1), we see that *the temperature update* $\Delta T = -\eta \frac{\partial \ell}{\partial T}$ *is proportional to the average correlation, at stochastic equilibrium, between the current score* $\lambda(X^n)$ *and the centered log likelihood of the infinite past* $(X^n X^{n-1} \dots)$.

0.26. Practical temperature adjustment

We point out that the updates ΔT of the temperature computed here deal only with one aspect of the performance, namely the expected score. However another aspect of the performance is crucial in applications, *namely the speed of stabilization of the Boltzmann machine.* Indeed, *for very low temperatures, stochastic equilibrium is reached only after a very long time*, and this second criterion should be taken into consideration when the temperature is adjusted. We shall come back to this problem in a forthcoming paper.

REFERENCES

[A] R. AZENCOTT - *-Markov fields and low-level vision tasks*, Proc. Int. Cong. App. Math. ICIAM, Paris (1987).

- *Gibbs fields, simulated annealing, and low-level vision tasks*, Proc. Cong. Pattern Recognition, AFCET-INRIA, Antibes (1987).

- *Synchronous Boltzmann machines : learning rules*, Proc. Congress "Neural networks", Les Arcs (1989), Springer-Verlag, NATO series (1990), vol. 68, Editors : Fogelman-Herault.

- *Parameter estimation for synchronous Markov fields* (to appear).

Bo] P. BOURLARD - *Multilayer perceptions and learning*, Proc. Congress "Neural networks", Les Arcs (1989), to appear in Springer-Verlag, NATO series (1990).

[G.] D. and S. GEMAN - *Gibbs fields, simulated annealing, and Bayesian reconstruction of images*, IEEE, PAMI (1984).

[H. A.] G. HINTON, T. SEJNOWSKI, D.H. ACKLEY - *Boltzmann machines : constraint satisfaction networks that learn*, Technical Report, Carnegie Mellon University (1984).