G-Maximization: an Unsupervised Learning Procedure for Discovering Regularities

Barak A. Pearlmutter
Geoffrey E. Hinton

Department of Computer Science
Carnegie-Mellon University
Pittsburgh, PA 15213

May, 1986

Abstract

Hill climbing is used to maximize an information theoretic measure of the difference between the actual behavior of a unit and the behavior that would be predicted by a statistician who knew the first order statistics of the inputs but believed them to be independent. This causes the unit to detect higher order correlations among its inputs. Initial simulations are presented, and seem encouraging. We describe an extension of the basic idea which makes it resemble competitive learning and which causes members of a population of these units to differentiate, each extracting different structure from the input.

Introduction

There are two distinct classes of theory about how to modify weights in networks of neuron-like units. Supervised learning theories like perceptrons and their more recent generalizations[1] assume that there is a special input to a unit which indicates how it ought to behave or how it ought to modify its behavior. Unsupervised learning theories assume that weight modification is based solely on the inputs to the unit and its actual responses. Typically, these theories have first suggested an intuitively plausible weight modification rule[2, 3, 4, 5] and then investigated the consequences of this rule. This paper presents an alternative approach in which the learning procedure is derived from a principle which specifies how the unit should behave. The principle proposed[6] is that the unit should respond to patterns in its inputs that occur more often than would be expected if the activities of the individual input lines were assumed to be independent. This is equivalent to saying that the unit should respond to higher order statistical regularities in its ensemble of input vectors.

At first sight, it seems that a unit would have to keep a record of its history of input vectors in order to discover higher order regularities. As we shall see, however, it is only necessary to keep track of two variables for each weight and a few more at the level of the whole unit.

Consider a unit with to 8 input lines, each of which alone is active 1/2 of the time, 4 of which are completely unrelated to any of the others, and the other 4 of which are highly correlated, either all being on at the same time or all being off. Our unit would ignore the 4 inputs with no higher order structure and latch onto the regularity present in the other 4. It

0094-243X/86/1510333-6$3.00 Copyright 1986 American Institute of Physics

could do this by developing strong positive weights to the 4 correlated inputs and a very high positive threshhold which could only be overcome if all those 4 input lines were on. Because these lines are correlated, the unit would come on with probability 1/2, but a statistician who assumed the lines were independent would predict that the unit would come on with probability 1/16.

The One Unit Case

Consider a unit which takes the weighted sum of its binary inputs s_i, runs that sum, x, through a logistic function σ to get y, and generates output 1 with probability y and output 0 with probability $1-y$.

$$x=\sum_{i=0}^{n} w_i s_i \quad \text{and} \quad y=\sigma(x)$$

where $\sigma(x)=1/(1+e^{-x})$, s_i is the state of the i^{th} input, and the w_i are the weights. Note that $d\sigma(x)/dx=\sigma'(x)=\sigma(x)(1-\sigma(x))$ resembles a gaussian. Let the unit be exposed to some stationary probability distribution over the 2^n possible input vectors. Given this input distribution P, the unit has expected output of

$$\langle y\rangle=\sum_{\alpha} P(\alpha)\sigma(x^{\alpha}).$$

Imagine someone who thought that the various inputs to the unit were statistically independent. Suppose this person recorded the first order statistics of the input lines, $p_i=\sum_{\alpha} P(\alpha)s_i^{\alpha}$ where s_i^{α} is the state of the i^{th} input line for the α^{th} input vector. Assuming independence, this person would expect the input vectors to follow the distribution P' and would predict the unit to have an expected output $\langle y\rangle'$:

$$P'(\alpha)=\prod_{i}\begin{cases} p_i & \text{when } s_i^{\alpha}=1 \\ 1-p_i & \text{when } s_i^{\alpha}=0 \end{cases} \qquad \langle y\rangle'=\sum_{\alpha} P'(\alpha)\sigma(x^{\alpha}).$$

The unit is detecting an interesting feature of the actual input distribution to the extent that the actual expected output differs from this predicted expected output. More formally, we can measure how many bits of information this person gains about the actual input distribution P when told that the actual expected output of the unit is $\langle y\rangle$.[7]

$$G=\langle y\rangle\log\frac{\langle y\rangle}{\langle y\rangle'}+(1-\langle y\rangle)\log\frac{1-\langle y\rangle}{1-\langle y\rangle'}$$

This measure tells us how good a feature detector the unit is, so in order to develop our unit into a good feature detector we can hill climb in G by modifying the w_i, the only parameters under our control.

$$\frac{\partial G}{\partial w_i}=\frac{\partial G}{\partial\langle y\rangle}\frac{\partial\langle y\rangle}{\partial w_i}+\frac{\partial G}{\partial\langle y\rangle'}\frac{\partial\langle y\rangle'}{\partial w_i}$$

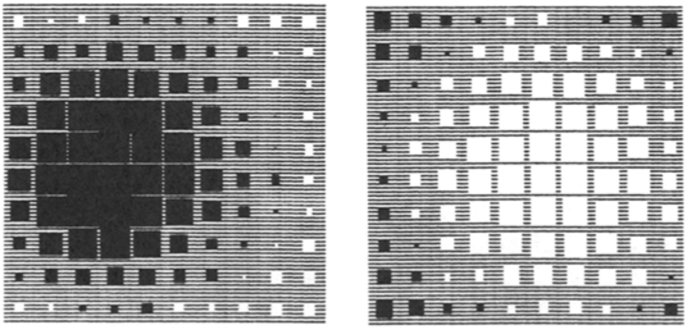
$$=[\log\frac{\langle y\rangle}{\langle y\rangle'}-\log\frac{1-\langle y\rangle}{1-\langle y\rangle'}]\sum_{\alpha} P(\alpha)\sigma'(x^{\alpha})s_i^{\alpha} + [\frac{1-\langle y\rangle}{1-\langle y\rangle'}-\frac{\langle y\rangle}{\langle y\rangle'}]\sum_{\alpha} P'(\alpha)\sigma'(x^{\alpha})s_i^{\alpha}$$

Note that the "prime" notation is used in different senses in P' and σ'. We proceed by accumulating the quantities on the right hand side of the above equation and using these to modify the weights with the simple rule $w_i^{\mathrm{new}} = w_i^{\mathrm{old}} + \epsilon\partial G/\partial w_i$.

To accumulate these right hand side quantities, we sample the distributions P and P'. We call the phase during which P is sampled the "structured phase" because the higher order structure is present in the ensemble of input vectors; to sample P' we introduce an "unstructured phase" in which the input lines are statistically independent. Thus, for each unit we accumulate $\sigma(x^\alpha)$ during the structured phase to give us $\langle y\rangle$ and the same quantity during the unstructured phase to give us $\langle y\rangle'$. In addition, during the structured phase we accumulate $\sigma'(x^\alpha)s_i^\alpha$ for each weight to give us $\partial\langle y\rangle/\partial w_i$, and the same quantity during the unstructured phase to give us $\partial\langle y\rangle'/\partial w_i$. Initially, we set the weights to small random values to help break symmetry.

A Simulation

If the input vector is a 10 by 10 array and the distribution is composed of single, randomly oriented, randomly positioned, black-white edges on this "retina," a unit typically develops into on-center off-surround detector (or vice versa) as in figure 1. To understand why this is so, consider the regularity captured in the input. If the input was random uncorrelated noise, like static on a TV screen, this unit would almost always come on, so $\langle y\rangle'$ is almost 1. However, the center of the receptive field is frequently not on the bright side of the edge, so $\langle y\rangle$ is (in this case) about 0.7. The unit is capturing the fact that nearby pixels tend to have the same value. It is interesting to note that, given this input distribution, changing the signs of all the weights would leave G unchanged.



Some typical detectors developed in response to randomly oriented black-white edges on a 2D field.

Figure 1:

Multiple Units

The above treatment deals only with a single unit. Were we to have a number of such units, they could all develop to detect the same feature. We need some force that will cause them to differentiate. One obvious method is to require each pair of units to be pretty much uncorrelated. If r_{ab} is the correlation between units a and b, rather than maximizing G we can maximize a new measure,

$$G^* = (1-k)\sum_a G_a - k\frac{1}{2}\sum_{a\neq b} r_{ab}^2, \qquad \frac{\partial G^*}{\partial w_{ai}} = (1-k)\frac{\partial G_a}{\partial w_{ai}} - k\sum_{b\neq a} r_{ab}\frac{\partial r_{ab}}{\partial w_{ai}}$$

where k is a constant controlling the relative importance of making good feature detectors and making the feature detectors uncorrelated, and r and its derivatives are computed as follows:

$$r_{ab} = q^{11}_{ab} q^{00}_{ab} - q^{10}_{ab} q^{01}_{ab}$$

$$q^{11}_{ab} = \sum_{\alpha} P(\alpha)\sigma(x^{\alpha}_{a})\sigma(x^{\alpha}_{b}), \qquad q^{10}_{ab} = \sum_{\alpha} P(\alpha)\sigma(x^{\alpha}_{a})(1-\sigma(x^{\alpha}_{b}))$$

$$q^{01}_{ab} = \sum_{\alpha} P(\alpha)(1-\sigma(x^{\alpha}_{a}))\sigma(x^{\alpha}_{b}), \qquad q^{00}_{ab} = \sum_{\alpha} P(\alpha)(1-\sigma(x^{\alpha}_{a}))(1-\sigma(x^{\alpha}_{b}))$$

Once again we do gradient descent on a measure by sampling the distribution and accumulating quantities as we sample, notably these q^{xx}_{ab} and their dervivatives with respect to each weight,

$$\begin{aligned}
w^{\text{new}}_{ai} &= w^{\text{old}}_{ai} + \varepsilon \frac{\partial G^{*}}{\partial w_{ai}} \\
&= w^{\text{old}}_{ai} + \varepsilon[(1-k)\frac{\partial G_a}{\partial w_{ai}} - k \sum_{b \neq a} r_{ab}\frac{\partial r_{ab}}{\partial w_{ai}}] \\
&= w^{\text{old}}_{ai} + \varepsilon[k\frac{\partial G_a}{\partial \langle y_a \rangle}\frac{\partial \langle y_a \rangle}{\partial w_{ai}} + k\frac{\partial G_a}{\partial \langle y_a \rangle'}\frac{\partial \langle y_a \rangle'}{\partial w_{ai}} \\
&\quad -(1-k)\sum_{b \neq a} r_{ab}[q^{11}_{ab}\frac{\partial q^{00}_{ab}}{\partial w_{ai}} + q^{00}_{ab}\frac{\partial q^{11}_{ab}}{\partial w_{ai}} - q^{10}_{ab}\frac{\partial q^{01}_{ab}}{\partial w_{ai}} - q^{01}_{ab}\frac{\partial q^{10}_{ab}}{\partial w_{ai}}]]
\end{aligned}$$

so we accumulate:

for each pair of units:		for each w_{ai} and other unit b:	
q^{11}_{ab}	$\sigma(x^{\alpha}_{a})\sigma(x^{\alpha}_{b})$	$\partial q^{11}_{ab}/\partial w_{ai}$	$\sigma'(x^{\alpha}_{a})\sigma(x^{\alpha}_{b})s^{\alpha}_{i}$
q^{10}_{ab}	$\sigma(x^{\alpha}_{a})(1-\sigma(x^{\alpha}_{b}))$	$\partial q^{10}_{ab}/\partial w_{ai}$	$\sigma'(x^{\alpha}_{a})(1-\sigma(x^{\alpha}_{b}))s^{\alpha}_{i}$
q^{01}_{ab}	$(1-\sigma(x^{\alpha}_{a}))\sigma(x^{\alpha}_{b})$	$\partial q^{01}_{ab}/\partial w_{ai}$	$(1-\sigma'(x^{\alpha}_{a}))\sigma(x^{\alpha}_{b})s^{\alpha}_{i}$
q^{00}_{ab}	$(1-\sigma(x^{\alpha}_{a}))(1-\sigma(x^{\alpha}_{b}))$	$\partial q^{00}_{ab}/\partial w_{ai}$	$(1-\sigma'(x^{\alpha}_{a}))(1-\sigma(x^{\alpha}_{b}))s^{\alpha}_{i}$

If we have n input bits and m units, our original scheme (without decorrelation) takes 2 units of storage for each unit, to hold $\langle y \rangle$ and $\langle y \rangle'$, and 2 for each weight, to hold $\partial\langle y \rangle/\partial w_i$ and $\partial\langle y \rangle'/\partial w_i$. Assuming we wish to decorrelate each pair of units, our new decorrelation scheme requires an additional 4 values for each of the $m(m-1)/2$ pairs of units, and an additional $4(m-1)$ for each weight. Although simulations show that this decorrelation method is effective, we find it heavyhanded and implausible.

With a simple approximation we can greatly simplify the decorrelation. Given that the units come on rarely, the decorrelation scheme described above can be approximated by mutual inhibition. For instance, if units are on only one time in a thousand then two decorrelated units will come on together only one time in a million, which is negligible.

Mutual inhibition between rarely active units also eliminates higher order correlations (which are not precluded by explicit pairwise decorrelation.) We have not yet simulated this mutual inhibition technique.

It is interesting to note that Boltzmann machines[8] handle higher order correlations in a way that is both principled and space-efficient (but slow.) At thermal equilibrium, a Boltzmann machine communicates information about higher order correlations via local pairwise interactions. This allows it to develop weights which ensure that the higher order correlations between its units are the same in two different phases. Notice that a Boltzmann machine learns by making its spontaneously generated output be as similar as possible to the required structured output, whereas G-Maximization learns by making its response to structured input be as different as possible from its response to unstructured input.

Further Elaborations

If different units are connected to different subsets of the total set of input lines, they will tend to detect different things. This means that decorrelation or mutual inhibition is only needed for nearby units.

If we know what we want a unit to detect, we can supervise it by adding an extra input to the unit and initializing the weight on that input to a high value. We then turn this input bit on when the feature we are interested in is present and off when its not. The unit's other weights will develop to detect this feature, unless such a feature isn't really present in which case the weight to our extra input will decrease until the unit can ignore it and pick up some real feature.

If one desires the feature detectors to be rarely active,[9] one can add another term to the G measure to impose this additional constraint. We let

$$G^{**} = (1-k_1-k_2)G - k_1\frac{1}{2}\sum_{a \neq b} r_{ab}^{\,2} - k_2\frac{1}{2}\sum_{a}(\langle y_a\rangle - d)^2$$

where d is the desired activity level. The corresponding modification to the learning procedure is simple, requiring no additional state.

$$\frac{\partial G^{**}}{\partial w_{ai}} = \cdots - k_2(\langle y_a\rangle - d)\frac{\partial\langle y_a\rangle}{\partial w_{ai}}$$

A unit which is forced to be rarely active will tend to maximize G by responding to very high order regularities.

If we want a unit to be helpful for deciding which of two distributions gave rise to the input vector, we can replace the structured and unstructured phases by these two distributions.

Relation to Hebbian Learning

A careful examination of the single unit case reveals that the learning rule resembles Hebbian learning. An intuitive way of looking at the process is as follows. We define a marginal case to be one in which the total input, x, to the unit is on the steep part of the logistic function, where $\sigma(x)$ is higher than usual (if we assume that units are rarely active.) We assume that $\langle y \rangle$ is higher than $\langle y \rangle'$. If an input line, i, is involved in more marginal cases during the structured phase than during the unstructured phase, raising w_i will raise $\langle y \rangle$ more than it raises $\langle y \rangle'$ so it will normally raise G. If we identify the unstructured phase with sleep[10] we expect Hebbian learning during wake and reverse Hebbian learning during sleep.

Acknowledgements

This research was supported by contract N00014-86-K-00167 from the Office of Naval Research. Barak Pearlmutter is a Hertz Fellow. We thank Richard Szeliski for useful discussions.

References

1. Rumelhart, D. E., Hinton, G. E., & Williams, R. J., "Learning internal representations by error propagation", in *Parallel distributed processing: Explorations in the microstructure of cognition,* D. E. Rumelhart, J. L. McClelland, & the PDP research group, eds., Bradford Books, Cambridge, MA, Vol. I, 1986.

2. Hebb, D. O., *The Organization of Behavior,* Wiley, New York, 1949.

3. Marr, D., "A theory of cerebellar cortex", *Journal of Physiology (London)*, Vol. 202, 1969, pp. 437-470.

4. Von der Malsburg, C., "Self-organizing of orientation sensitive cells in striate cortex", *Kybernetik*, Vol. 14, 1973, pp. 85-100.

5. Rumelhart, D. E. and Zipser, D., "Competitive Learning", *Cognitive Science*, Vol. 9, 1985, pp. 75-112.

6. Hinton, G. E., "Implementing semantic networks in parallel hardware", in *Parallel Models of Associative Memory,* G. E. Hinton & J. A. Anderson, eds., Erlbaum, Hillsdale, NJ, 1981.

7. Kullback, S., *Information Theory and Statistics,* Wiley, New York, 1959.

8. Ackley, D. H., Hinton, G. E., Sejnowski, T. J., "A learning algorithm for Boltzmann machines", *Cognitive Science*, Vol. 9, 1985, pp. 147-169.

9. Barlow, H. B., "Single units and sensation: A neuron doctrine for perceptual psychology?", *Perception*, Vol. 1, 1972, pp. 371-394.

10. Crick, F. & Mitchison, G., "The function of dream sleep", *Nature*, Vol. 304, 1983, pp. 111-114.