In *The Handbook of Brain Theory and Neural Networks*, M. Arbib (ed), La Jolla, CA: MIT Press

Unsupervised Learning With Global Objective Functions

Suzanna Becker
Department of Psychology
McMaster University
1280 Main Street West
Hamilton, Ontario, L8S 4K1
Canada

RUNNING HEAD: Unsupervised learning

Correspondence:
Suzanna Becker
Department of Psychology, McMaster University
1280 Main Street West, Hamilton, Ontario, Canada L8S 4K1
Phone: (905) 525-9140 ext. 23020
Fax: (905) 529-6225
email: becker@mcmaster.ca

1 INTRODUCTION

In this article, we review three types of neural network learning procedures which can be considered *unsupervised*: information-preserving algorithms, density estimation techniques, and invariance-based learning procedures. This decomposition does not necessarily imply three strictly non-overlapping classes, but rather it is meant to emphasize the different underlying principles that motivated each algorithm's development. We will use the term *unsupervised* to refer to those algorithms for which there is no externally derived teaching signal informing the network as to whether or not it has produced the correct response for each input pattern. Invariably, though, for each unsupervised learning procedure there is an implicit *internally-derived* training signal; this training signal may be based on the network's ability to predict its own input, or on some more general measure of the quality of its internal representation.

1.1 Global objective functions or synaptic learning rules?

Since our concern is with unsupervised learning in *networks* and their global behaviour, we will focus on algorithms based upon globally-defined objective functions, rather than synaptic learning rules. By performing gradient descent in a global objective function we can reduce a global algorithm into synaptic-level steps (weight changes), but the converse is not necessarily true; i.e., a given synaptic learning rule may not correspond to the derivative of any global objective function. There are many advantages afforded by the "global approach". It allows us to understand the operation of the network in an information-processing sense, i.e., in terms of what sort of transformation the network applies to the input; such an understanding can be elusive if we begin with a synaptic learning rule and then try to predict its global behavior. The global approach also adheres to the principles of good algorithm design well-known to the computer scientist: we start with a conceptual specification of what the learning is meant to accomplish; this is translated into a computational specification - the objective function, which is then refined into detailed computational steps - the synaptic learning rules. This top-down approach allows us to explore different implementations of the same learning algorithm, such as batch versus online versions. Finally, the global objective function provides a quantitative measure of the success of the learning procedure, and we can (usually) detect its convergence.

In contrast to this top-down approach, the earliest computational models of learning were based on Hebb's synaptic learning principle; Hebb postulated that synaptic efficacy should increase whenever two pre- and post-synaptic neurons are co-active. Many computational models have built upon this principle (see HEBBIAN RULES AND TENSOR PRODUCTS). It has also gained popularity among neurobiologists as a plausible candidate for a cortical synaptic learning mechanism. It is therefore of interest to computational modellers to try to translate their global learning procedures into local, biologically plausible learning rules such as Hebbian learning.

1.2 Self-organization in perceptual systems

One of the major motivations for studying unsupervised learning is to discover the general computational principles underlying brain self-organization. Evidence of experience-dependent plasticity has been reported in a wide variety of brain areas. Perhaps the most startling evidence comes from a series of studies by Sur and colleagues (reviewed in Sur, 1989), who found that by artificially rerouting primary visual cortical input pathways to the auditory cortex in ferrets, the "auditory" cortical cells develop responses to visual stimuli, and exhibit typical visual cortical receptive fields. According to Asanuma (1991, pp. 217), "... the long-held belief that the cortical representation of the sensory periphery is hard wired in adults has become less and less tenable." It seems that the brain has a dynamic restructuring capacity which is not only restricted to primary sensory areas, and may be a ubiquitous property of the adult neocortex (Asanuma, 1991). This possibility raises a number of questions: Are there any general, unsupervised organizing principles which predict cortical reorganization, and can they be expressed computationally, as global objective functions for learning? Is more than one such principle required? What architectural constraints are necessary for successful learning, and how do they interact with the choice of objective function? It is these sorts of questions that unsupervised learning research is concerned with.

2 INFORMATION-PRESERVING ALGORITHMS

Since there is no external teaching signal for unsupervised learning, the goal of the learning must be stated solely in terms of some transformation on the input which will preserve the interesting structure. The first task then is to define what constitutes interesting structure. The most general possible goal is to try to preserve *all* of the information by simply memorizing the input patterns. Pattern-associators (see ASSOCIATIVE MEMORY) can be used as such by operating in auto-associative mode, i.e., by storing each input pattern associated with itself. All of these models suffer capacity limitations: only a limited number of patterns can be stored and perfectly recalled by a network of fixed size.

2.1 Minimizing reconstruction error

Given the limited ability of networks to store a set of patterns exactly, a better strategy might be to try to find a *compressed* representation of the patterns. This may be helpful for preprocessing noisy data, and for modelling early stages of perceptual processing. A standard data compression technique is principal components analysis (PCA) (see PRINCIPAL

COMPONENTS ANALYSIS). Several learning procedures (reviewed in Becker and Plumbley, 1994) have been developed which converge to the first N principal directions of the input distribution. These methods are optimal with respect to minimizing the mean squared reconstruction error for linear networks. However, there is no guarantee that a linear method like PCA will capture the interesting structure in arbitrary input distributions.

A more general method for finding a compressed representation that minimizes reconstruction error is to use a nonlinear back-propagation network as an auto-encoder (Hinton, 1989), by making the desired states of the N output units identical to the states of the N input units on each case. Data compression can be achieved by making the number of hidden units $M < N$. Further, the features discovered by the hidden units may be useful for subsequent stages of processing such as classification. However, with complicated input patterns containing multiple features, it may not be possible to relate the activities of individual hidden units to specific features. One way to constrain the hidden unit representation is to add extra penalty terms to the objective function. For example, Saund (1989) added a constraint that caused hidden units to represent high-dimensional data as single points on a lower-dimensional constraint surface, by penalizing activation patterns that deviated from unimodal distributions. This encourages units to represent a single scalar dimension that best characterizes the input. Zemel and Hinton (see MINIMUM DESCRIPTION LENGTH APPLICATIONS OF NEURAL NETWORKS) generalized this idea by imposing an MDL-based penalty term on hidden unit activities.

2.2 Direct minimization of information loss

Another approach to ensuring that the important information in the input is preserved in the output is to use concepts from information theory. Many learning procedures have been proposed which minimize the information loss in a network, subject to processing constraints (reviewed in Becker and Plumbley, 1994). The common feature of these methods is the preservation of mutual information (Shannon, 1948) between the input vector $\mathbf{x}$ and output vector $\mathbf{y}$:

$$I_{x;y} \quad = \quad H(\mathbf{x}) - H(\mathbf{x} \mid \mathbf{y}) \tag{1}$$

where $H(\mathbf{x}) = -\int_{\mathbf{x}} p(\mathbf{x}) \log p(\mathbf{x}) dx$ is the entropy of random variable x with probability distribution $p(x)$, and $H(\mathbf{x} \mid \mathbf{y}) = -\int_{\mathbf{x},\mathbf{y}} p(\mathbf{x},\mathbf{y}) \log p(\mathbf{x} \mid \mathbf{y}) d\mathbf{x}\ d\mathbf{y}$ is the entropy of the conditional distribution of $\mathbf{x}$ given $\mathbf{y}$. This measure tells us the amount of information (uncertainty) in $\mathbf{x}$ less the uncertainty remaining in $\mathbf{x}$ when $\mathbf{y}$ is known. Thus, $I_{x;y}$ is high when $\mathbf{x}$ is difficult to predict *a priori*, and becomes much easier to predict after being told $\mathbf{y}$.

If the network is free of processing noise and has enough units, its output layer can convey all the information contained in the input simply by copying the input. Linsker (1988) proposed applying the “Infomax principle” in the presence of Gaussian processing noise at the output layer for linear networks; when the input distribution is Gaussian, the information is:

$$I = 0.5 \log \left(\frac{|\mathbf{Q}^{\mathbf{y}}|}{V(n)} \right)$$

where $|\mathbf{Q}^{\mathbf{y}}|$ is the determinant of the covariance matrix of the output vector $\mathbf{y}$ (the signal plus noise) and $V(n)$ is the noise variance. Maximizing this quantity results in a tradeoff between maximizing the variances of the outputs, and decorrelating them, depending on the noise level. For a single output unit, this leads to a simple Hebb-like learning rule.

An alternative optimality criterion proposed by Barlow (1989) is to find a minimally *redundant* encoding of the sensory input vector into an n-element feature vector, which should facilitate subsequent learning. If the n features are statistically independent, then the formation of new associations with some event V (assuming the features are also approximately independent conditioned on V) only requires knowledge of the conditional probabilities of V given each feature y_i, rather than complete knowledge of the probabilities of events given each of the 2^m possible sensory inputs. Barlow proposes that one could achieve featural independence by finding a *minimum entropy encoding*: an invertible code which minimizes the sum of the feature entropies.

Several approximate solutions to Barlow's model in the linear case are reviewed by Becker (1991). The nonlinear case is of course much more difficult to learn, requiring a much stronger result of statistically independent, rather than just decorrelated, outputs. In general this is an intractable problem; that is, to verify the statistical independence of n items requires the enumeration of on the order of n^n statistics. Thus, tractable approximations to this objective function are needed.

3 DENSITY ESTIMATION TECHNIQUES

Rather than trying to retain all the information in the input, we could try to characterize its underlying probability distribution by developing a more abstract representation. Many standard statistical methods fall under the category of density estimation techniques (for a good introduction, see Silverman, 1986), and several unsupervised learning procedures can be viewed in this way. The general approach is to assume *a priori* a class of models which constrains the general form of the probability density function; then search for the particular model parameters defining the density function most likely to have generated the observed data. This can be cast as an unsupervised learning problem by treating the network weights as the model parameters, and the overall function computed by the network as being directly related to the density function.

3.1 Mixture models and competitive learning

One possible choice of prior model is a mixture of Gaussians. The prior assumption in this case is that each data point was actually generated by one of n Gaussians having different means μ_i, variances ${\sigma_i}^2$, and prior probabilities π_i. Fixing the model parameters μ_i, σ_i, and π_i, we can compute the probability of a given data point $\mathbf{x}$ under a mixture-of-Gaussians model as follows:

$$p(\mathbf{x} \mid \{\mu_i\}, \{\sigma_i\}, \{\pi_i\}) \quad = \quad \sum_{i=1}^{n} \pi_i P_i(\mathbf{x}, \mu_i, \sigma_i) \tag{2}$$

where $P_i(\mathbf{x}, \mu_i, \sigma_i)$ is the probability of $\mathbf{x}$ under the ith Gaussian. Applying Bayes' rule, we can also compute the probability that any one of the Gaussians generated the data point $\mathbf{x}$:

$$p(i \mid \mathbf{x}, \{\mu_j\}, \{\sigma_j\} \{\pi_j\}) = \frac{\pi_i \; P_i(\mathbf{x}, \mu_i, \sigma_i)}{\sum_{j=1}^{n} \pi_j \; P_j(\mathbf{x}, \mu_j, \sigma_j)} \tag{3}$$

Given these probabilities, we can now use as a cost function the log likelihood of the data given the model: $\log(L) = \sum_x \log(p(\mathbf{x} \mid \{\mu_i\}, \{\sigma_i\}, \{\pi_i\}))$ By maximizing this function, we can approximate the true probability distribution of the data, given our prior model assumptions. Note that by taking the log of L, we obtain a cost function which is a sum (rather than a product) of probabilities for each input pattern. The model parameters can then be adapted by performing gradient ascent in $\log(L)$. The EM algorithm (Dempster et al., 1977) alternatingly applies equation 2 (the Expectation step) and adapts the model parameters (the Maximization step) to converge on the maximum likelihood mixture model of the data.

Competitive learning procedures (see FEATURE DISCOVERY BY COMPETITIVE LEARNING) perform a discrete approximation to density estimation. The general idea is that units compete to respond (e.g. by a winner-take-all activation function or lateral inhibition), so that only the winning unit in each competitive cluster is active. Only this unit learns on each case, by moving its weight vector closer to the current input pattern. Hence, each unit minimizes the squared distance between its weight vector and the patterns nearest to it, as in standard k-means clustering. This version of competitive learning is closely related to fitting a mixture of Gaussians model with equal priors π_i and equal fixed variances ${\sigma_i}^2$. Using the EM algorithm, every unit (not just the winner) moves its mean closer to the current input vector, in proportion to the probability that it's Gaussian model accounts for the current input (equation 3). Competitive learning approximates this step by making a binary decision as to which unit accounts for the input. Thus, the same learning rule applies, except that the proportional weighting is replaced by an all-or-none decision.

Nowlan (1991) proposed a "soft competitive learning" model for neural networks. Rather than only allowing the winner to adapt, each unit adapts its weights for every input case, in proportion to how strongly it responds on a given case. This is an online version of the

EM algorithm for Gaussian densities with equal priors, and adaptive means and variances. Nowlan found this method to be superior to the traditional "hard competitive learning models" on several classification tasks.

3.2 Combinatorial representations

A major limitation of mixture models and competitive learning is that they employ a 1-of-n encoding, in which a single unit or model is assumed to have generated the data. A *multiple causes* model is more appropriate when the most compact data description consists of several independent parameters (e.g. color, shape, size). Several examples of this approach are reviewed in (Becker and Plumbley, 1994). For example, Neal's (1992) multilayer "connectionist belief networks" resemble stochastic Boltzmann machines (see BOLTZMANN MACHINES), but they are strictly feedforward. Output states are clamped to patterns selected from the environment, while the hidden unit state space is randomly explored. The weights are adjusted so as to increase the probability of the hidden units generating the clamped output patterns. The network thereby learns to represent features in the hidden layer which explain correlations in the pattern set.

4 INVARIANCE-BASED LEARNING

The methods discussed so far try to extract useful structure from raw data, assuming minimal prior knowledge. How can unsupervised learning be applied beyond these preprocessing stages, to extract higher order features and build more abstract representations? One approach is to restrict our search to particular kinds of structure. We can make constraining assumptions about the structure we are looking for, and build these constraints into the network's architecture and/or objective function to develop more efficient, specialized learning procedures.

4.1 Spatially and temporally coherent features

Becker and Hinton's (1992) Imax learning procedure discovers properties of the input that are coherent across space and time, by maximizing the mutual information between the *outputs*, y_a and y_b, of network modules that receive input from different parts of the sensory input (e.g. different modalities, or different spatial or temporal samples). Note how this objective function differs from the Infomax principle; the latter tries to retain *all* of the information in the input by maximizing the mutual information between inputs and outputs, whereas Imax tries to extract only those features common to two or more distinct parts of the input.

Under Gaussian assumptions about the signal and noise, Becker and Hinton derived the following objective function for the learning:

$$I = 0.5 \log \frac{V(y_a + y_b)}{V(y_a - y_b)}$$

This measure tells how much information the average of y_a and y_b conveys about the common underlying signal, i.e., the feature which is coherent across the two input samples. When applied to networks composed of multi-layer modules that receive input from adjacent, non-overlapping regions of the input, Imax discovered higher order image features (i.e., features not learnable by single-layer or linear networks) such as stereo disparity in random dot stereograms. One way to apply Imax to more than two modules is to have each module make a prediction about a linear combination of several neighboring modules' outputs. Becker and Hinton showed that a layer of linear units can thereby interpolate surface depth by learning to optimally combine local depth measurements. Note that Imax requires back-propagation of derivatives to train the weights to the hidden units, and the storage of several statistics on each link to compute the mutual information derivatives. Thus, a more biologically plausible approximation is needed.

5 DISCUSSION

We have argued in favor of the "global objective function" approach to modelling unsupervised learning processes, and explored several powerful learning procedures based on this approach. These methods have had success in modelling early perceptual processing. With the incorporation of highly constraining prior models, unsupervised learning procedures can form even more abstract representations of data, and extract higher-order features. A major direction for future research is to find tractable instantiations of these learning procedures, and to apply them in multiple learning stages to form a diversity of representational levels. Additionally, in order to remain within the realm of biological plausibility, many of these learning models must be extended to yield simple, local synaptic learning rules.

Acknowledgments

The author acknowledges support from the McDonnell-Pew Program in Cognitive Neuroscience, research grant #92-40, and the Natural Sciences and Engineering Research Council of Canada. Many thanks also to Michael Arbib, Chris Williams and Rich Zemel for helpful comments.

References

Asanuma, C. (1991). Mapping movements within a moving motor map. Trends in Neursciences, 14:217-218.

Barlow, H. B. (1989). Unsupervised learning. Neural Computation, 1:295-311.

Becker, S. (1991). Unsupervised learning procedures for neural networks. International Journal Of Neural Systems, 2:17-33.

Becker, S. and Hinton, G. E. (1992). A self-organizing neural network that discovers surfaces in random-dot stereograms. Nature, 355:161-163.

Becker, S. and Plumbley, M. (1994). Unsupervised neural network learning procedures for feature extraction and classification. *To appear in the* International Journal of Applied Intelligence, Special Issue on Applications of Neural Networks, (F. Pineda, Ed.).

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. Proceedings of the Royal Statistical Society, B-39:1-38.

Hinton, G. E. (1989). Connectionist learning procedures. Artificial Intelligence, 40:185-234.

Linsker, R. (1988). Self-organization in a perceptual network. IEEE Computer, 21:105-117.

Neal, R. M. (1992). Connectionist learning of belief networks. Artificial Intelligence, 56:71-113.

Nowlan, S. J. (1990). Maximum likelihood competitive learning. Neural Information Processing Systems, Vol. 2, (D.S. Touretzky, Ed.), San Mateo, CA: Morgan Kaufmann, pp. 574-582.

Saund, E. (1989). Dimensionality-reduction using connectionist networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11:304-314.

Shannon, C. E. (1948). A mathematical theory of communication. Bell System Technical Journal, 27:379-423,623-656.

*Silverman, B. (1986). Density Estimation for Statistics and Data Analysis, London: Chapman and Hall.

Sur, M. (1989). Visual plasticity in the auditory pathway: Visual inputs induced into auditory thalamus and cortex illustrate principles of adaptive organization in sensory systems. in Dynamic Interactions in Neural Networks; Models and Data, (M.A. Arbib and S.I. Amari, Eds.), Springer-Verlag, pp. 35-51.