PARTIAL LIKELIHOOD FOR ESTIMATION OF MULTI-CLASS POSTERIOR PROBABILITIES

Tülay Adalı, Hongmei Ni, and Bo Wang

Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County, Baltimore, MD 21250
{adali,honi,bwang1}@engr.umbc.edu

ABSTRACT

Partial likelihood (PL) provides a unified statistical framework for developing and studying adaptive techniques for nonlinear signal processing [1]. In this paper, we present the general formulation for learning posterior probabilities on the PL cost for multi-class classifier design. We show that the fundamental information-theoretic relationship for learning on the PL cost, the equivalence of likelihood maximization and relative entropy minimization, is satisfied for the multi-class case for the perceptron probability model using softmax [2] normalization. We note the inefficiency of training a softmax network and propose an efficient multi-class equalizer structure based on binary coding of the output classes. We show that the *well-formed* property of the PL cost [1, 7] is satisfied for the softmax and the new multi-class classifier. We present simulation results to demonstrate this fact and note that though the traditional mean square error (MSE) cost uses the available information more efficiently than the PL cost for the multi-class case, the new multi-class equalizer based on binary coding is much more effective in tracking abrupt changes due to the well-formed property of the cost that it uses.

1. INTRODUCTION

The probabilistic view of a neural network classifier such that the network outputs are associated with posterior class probabilities is quite attractive for a number of reasons. Among others, this view offers advantages both in understanding the properties of learning in neural networks and in developing new approaches for learning. Partial likelihood, on the other hand, provides a general probabilistic framework for designing nonlinear classifiers, and is particularly suitable for developing adaptive techniques for nonlinear signal processing. In [1], we introduce the use of PL for nonlinear signal processing, demonstrate its successful application in binary classification, and show a fundamental information theoretic relationship, the equivalence of the relative entropy minimization and likelihood maximization provided that two regularity conditions are satisfied. These conditions are satisfied for the binary multi-layer perceptron (MLP) classifier [1] and the finite normal mixtures (FNM) probability model [6]. In this paper, we first present the PL formulation for the multi-class case, and then show that the regularity conditions are also satisfied for the perceptron network that uses softmax normalization [2]. Note that normalization is needed for the multi-class case to ensure that the network outputs are valid probabilities.

The performance advantages of using the relative entropic (or partial likelihood) cost are presented in [1] for a binary classification example. This is a direct consequence of the *well-formed* property [7] of the relative entropic cost which guarantees the recovery of steepest descent learning from convergence at the wrong extreme, a property not satisfied by the mean square error (MSE) cost function. The advantages of the probabilistic framework in learning, shown with a number of numerical studies for binary classification [1, 4], do not easily carry to the multi-class case. This is primarily because of the inefficiency of using only the true class label information during training, i.e., only one target is nonzero for each sample during training. Hence, typically the convergence of a multi-class classifier that uses a statistical cost function (such as relative entropy that uses relative errors) is considerably slower than that of the MSE cost that uses absolute errors [5]. In [4], a modified softmax normalization is used that eliminates the inherent redundancy in the standard softmax and the training is achieved by a Gauss-Newton scheme which is shown to increase convergence rate considerably compared to learning by the Robbins-Monro procedure. In this paper, we provide an alternative structure for estimating multi-class probabilities in a feedforward network that is trained using the Robbins-Monro scheme but has faster convergence characteristics. For a multi-class problem where the total number of classes is given by M, we consider binary coding of the classes and map the multi-class estimation problem to estimation of $\lceil \log_2 M \rceil$ binary probabilities. Such a formulation is particularly convenient for applications in digital communications where the transmitted data are already binary encoded. The scheme also provides possibilities for incorporation of coding into the scheme to further decrease the bit error rate. We present simulation studies for a channel equalization example that demonstrate the efficiency of the scheme for learning class probabilities in a 4-level pulse amplitude modulation example. While the convergence of the MSE based equalizer is still slightly faster than that of the new multi-class equalizer that uses binary coding, the new equalizer is much more effective in tracking parameter changes since it directly inherits the well-formed properties of the binary classifier introduced in [1].

This work was supported by the National Science Foundation Career Award, NSF NCR-9703161.

2. NEURAL NETWORKS AS ESTIMATOR OF MULTI-CLASS POSTERIOR PROBABILITIES

Assume that we have a training set $\mathcal{T}$ of N related input and output pairs $\mathcal{T} = \{x_n, \mathbf{y}_n\}_{n=1}^N$ and the problem is to train the classifier such that for a given observation $\mathbf{y}_k$, $\mathbf{x}_k$ will be assigned to one of m classes $C_1, C_2, \cdots, C_m$, such that $\mathbf{x}_k$ takes a value from a finite alphabet $\mathcal{S} = \{\mathbf{a}_0, \mathbf{a}_1, ..., \mathbf{a}_m\}$. The actual value that the random variable $\mathbf{x}_k$ takes, i.e., the value $\mathbf{a}_i$ is of consequence only in the case of training using relative errors, e.g. when using the MSE cost.

Given the posterior class probabilities $P(C_i|\mathbf{y})$ for $i = 1, 2, \cdots, m$, Bayes classifier will assign $\mathbf{y}$ to class C_i if $P(C_i|\mathbf{y}) > P(C_j|\mathbf{y}) \quad \forall j \neq i$, a choice which minimizes the classification error probability. Note that, the total distribution information, rather than that of the most likely class only, can be useful depending on the particular application. Since, the goal is the estimation of the probabilities $P(C_i|\mathbf{y}) \; \forall \mathbf{a}_i \in \mathcal{S}$, we can use a feedforward neural network probability model such that

$$P_\theta(C_i|\mathbf{y}) = f_i(\theta, \mathbf{y}) \tag{1}$$

where θ is the vector of network parameters which we can estimate/ learn using the appropriate criterion. It is important to remember that the probabilistic formulation brings the additional constraint that the network outputs lie in the range [0,1] and that $\sum_{i=1}^m f_i(\theta, \mathbf{y}) = 1$.

For the binary case, $\mathcal{S} = \{0, 1\}$, and we only need to estimate $P_\theta(C_1|\mathbf{y})$ as $P_\theta(C_2|\mathbf{y}) = 1 - P_\theta(C_1|\mathbf{y})$. For example, we can use the MLP probability model as shown in [1]. For the general case of multiple classes however, to ensure that the network outputs are valid probabilities (i.e., they sum upto one), there is usually a second normalization stage that is cascaded to the feedforward structure. The exponential normalization, the so-called *softmax* function [2] has been the most popular for multi-class learning with the likelihood cost. We can introduce softmax normalization for a single hidden layer feedforward neural classifier with logistic activation function $h(\cdot)$ as

$$f_i(\theta, \mathbf{y}_n) = \frac{\exp(\phi_i)}{\sum_{j=1}^m \exp(\phi_j)} \tag{2}$$

where $\phi_j = \sum_{i=1}^q h\left(\mathbf{y}_n{}^T \mathbf{w}^i\right) v^{ij}$ with $\mathbf{w}^i \in \mathbf{R}^{L\times 1}$ and $\mathbf{y}_n \in \mathbf{R}^{L\times 1}$ and v^{ij} is the weight between the hidden node i and the jth output node.

Note that, rather than modeling the probability mass function (pmf), we can also choose probability models with continuous outputs, i.e., can model the probability density function (pdf) as a direct consequence of the Bayesian formula as shown in [6].

3. PARTIAL LIKELIHOOD FORMULATION AND THE INFORMATION-THEORETIC VIEW

We use a recent extension of maximum likelihood, the *partial likelihood* (PL) theory [3] in [1] to develop a general probabilistic framework for neural classifiers which is particularly suitable for application to problems in which time-ordering is essential (e.g. time-series problems), or can be conveniently defined. To write the PL, consider the training set $\mathcal{T}$ and define $\mathcal{F}_k$ as the σ-field generated by the past x_i, $i \leq k-1$, and the outputs (past covariate information) $\mathbf{y}_i$, $i \leq k-1$. It can also include the current output value $\mathbf{y}_k$. Hence, $\mathcal{F}_k$ is a collection of all relevant events upto discrete (time) instant k, i.e., represents the history at k and $\mathcal{F}_{k-1} \subset \mathcal{F}_k$, i.e, $\mathcal{F}_k$ is an increasing sequence of σ-fields. In a time series problem where the observation vector is defined as $\mathbf{y}_n = [y_n, y_{n-1}, \cdots, y_{n-L+1}]$ and a new sample is shifted in at each new time instant, the condition $\mathcal{F}_{n-1} \subset \mathcal{F}_n$ is easily satisfied. Also, the filtration requirement on the sigma-fields allos us to easily handle missing data problems.

The PL is written as the product

$$\mathcal{L}_N^p(\theta) = \prod_{j=1}^N \prod_{i=1}^m f_i(\theta, \mathcal{F}_j)^{T_{i,j}}. \tag{3}$$

where the indicator index $T_{i,j}$ is defined as $T_{i,j} = 1$ if $\mathbf{x}_j \in C_i$ and 0 otherwise. Note that to write the PL, we defined a new probability $P_\theta(C_i|\mathcal{F}_n)$ which is conditioned on all the past information available at the current instant n rather than the current output $\mathbf{y}_n$. Hence PL provides a formulation suitable for use of recurrent network probability models $f_i(\theta, \mathcal{F}_j)$ and the PL theory can be used to study properties of recurrent networks as well.

The relative entropy (RE), or the Kullback-Leibler distance, [**?**], on the other hand, is a fundamental information-theoretic measure of how accurate the estimated probability distribution p_θ is an approximation to the true probability distribution p and is given by $D(p\|p_\theta) = E\left\{\log \frac{p}{p_\theta}\right\}$ where the expectation is with respect to the true distribution p. The RE is always nonnegative and is zero only when the two distributions match, $p = p_\theta$. We can define the *accumulated* relative entropy (ARE) as the total Kullback-Leibler discriminatory information contained in the training set $\mathcal{T}$ as

$$\mathcal{I}_N(\theta) = \sum_{j=1}^N E\{\log \frac{P(C_i|\mathcal{F}_j)}{f_i(\theta, \mathcal{F}_j)}|\mathcal{F}_j\} \tag{4}$$

We assume that for θ_0, $f(\cdot)$ defined in (1) achieves the true probability distribution and define $r_j(\theta) \equiv \log \frac{P_{\theta_0}(C_i|\mathcal{F}_j)}{f_i(\theta, \mathcal{F}_j)}$ which allows us to write the ARE as $\mathcal{I}_N(\theta) = \sum_{j=1}^N i_j(\theta)$ where $i_j(\theta) = E\{r_j(\theta)|\mathcal{F}_j\}$ and define $\mathcal{J}_n(\theta) = \sum_{k=1}^N j_k(\theta)$ where $j_k(\theta) \equiv Var\{r_k(\theta)|\mathcal{F}_k\}$. The expectations in the above definitions are with respect to the true distribution $P_{\theta_0}(C_i|\mathcal{F}_j) \; \forall \mathbf{a}_i \in \mathcal{S}$. Based on these definitions, we establish the relationship between PL maximization and ARE minimization for the general case of dependent observations by the following theorem [1]:

Theorem: Given continuous functions $f_i(\cdot) \; \forall \mathbf{a}_i \in \mathcal{S}$, if, for each $\theta \neq \theta_0$, there exists a constant $\delta > 0$ such that, as $N \to \infty$,

$$P(\mathcal{I}_N(\theta)/N > \delta) \longrightarrow 1 \tag{5}$$

and

$$\mathcal{J}_N(\theta)/N^2 \longrightarrow 0 \;\; \text{in probability} \tag{6}$$

then at least one $\arg\min_\theta \mathcal{I}_N(\theta)$ tends to one $\arg\max_\theta \bar{\mathcal{L}}_N^p(\theta)$ almost surely on $\Omega = \{\theta \mid \mathcal{I}_N(\theta) \uparrow \infty, \sum_{i=1}^N j_i(\theta)/\mathcal{I}_i^2(\theta) < \infty\}$ where $\bar{\mathcal{L}}_N^p(\theta) \equiv \ln \mathcal{L}_N^p(\theta)$.

Thus the optimal model parameters θ_0 have the fundamental information theoretic interpretation that they minimize the Kullback-Leibler information given a probability model. Thus viewing learning as related to Kullback-Leibler information minimization in this way implies that learning is a *maximum likelihood* statistical estimation procedure. The proof of the theorem is given in [1].

The theorem establishes the equivalence of PL maximization and ARE minimization under two regularity conditions. The first condition of the theorem, (5), represents the rate by which the Kullback-Leibler information accumulates with N, and guarantees that for each $\theta \neq \theta_0$, $\mathcal{I}_N(\theta) \rightarrow \infty$ as $N \rightarrow \infty$, i.e. the information continues to accumulate. The second condition, (6), on the other hand implies asymptotical stability of variance. The conditions can be shown to be satisfied for the perceptron probability model [1] and the FNM model [6] for the binary case.

In what follows, we show that the two conditions of the above theorem, (5) and (6), are satisfied for a multi-class perceptron classifier using the softmax normalization representation given in (2). We first write

$$\ln f_k(\theta, \mathbf{y}_n) = \phi_k - \ln[\sum_{j=1}^{m} \exp(\phi_j)] \tag{7}$$

Using the definition given for the theorem, we can write

$$\begin{aligned} i_n(\theta) &= E[r_n(\theta)|\mathcal{F}_n] = E[\ln \frac{p_{\theta_0}(C_k|\mathbf{y}_n)}{f_k(\theta, \mathbf{y}_n)}|\mathcal{F}_n] \\ &= E[\ln p_{\theta_0}(C_k|\mathbf{y}_n)] - E[\phi_k] + E[\ln(\sum_{j=1}^{m} \exp(\phi_j))] \\ &= \sum_{j=1}^{m} [\ln p_{\theta_0}(C_k|\mathbf{y}_n)] p_{\theta_0}(C_k|\mathbf{y}_n) - \sum_{j=1}^{m} \phi_k p_{\theta_0}(C_k|\mathbf{y}_n) \\ &\quad + \sum_{j=1}^{m} [\ln(\sum_{j=1}^{m} \exp(\phi_j))] p_{\theta_0}(C_k|\mathbf{y}_n) \end{aligned} \tag{8}$$

Since we assume $\theta \in \mathbf{\Theta}$, where $\mathbf{\Theta}$ is a compact parameter set, ϕ_k is finite. Thus the last two terms in (8) are finite. We can show that the first term is also finite (when $p_{\theta_0}(C_k|\mathbf{y}_n)$ is equal to 0, the term is 0 by definition.) Hence, $i_n(\theta)$ is finite. From the definition of $i_n(\theta)$, we know $i_n(\theta) > 0$, for $\theta \neq \theta_0$. So there exists a constant $\delta > 0$ such that, as $n \rightarrow \infty$, $P(\mathcal{I}_n(\theta)/n > \delta) \longrightarrow 1$. Similarly, we can obtain

$$\begin{aligned} j_n(\theta) &= Var(r_n(\theta)|\mathcal{F}_n) = E[(r_n(\theta) - i_n(\theta))^2|\mathcal{F}_n] \\ &= E[r_n(\theta)^2|\mathcal{F}_n] - i_n^2(\theta) \\ &= E[\ln^2(p_{\theta_0}(C_k|\mathbf{y}_n))]E[\ln^2(f_k(\theta, \mathbf{y}_n))] - \\ &\quad -2E[\ln(p_{\theta_0}(C_k|\mathbf{y}_n))\ln(f_k(\theta, \mathbf{y}_n))] - i_n^2(\theta) \end{aligned} \tag{9}$$

In the softmax model, $1 > f_k(\theta, \mathbf{y}_n) > 0$ because the observation vector is assumed to be finite and the parameter set compact, so the first three terms in (10) are finite. We have shown that $i_n(\theta)$ is also finite, thus $j_n(\theta)$ is finite. From its definition, we have $j_n(\theta) \geq 0$ which implies $\sum_{k=1}^{n} j_k(\theta)/n^2 \longrightarrow 0$. Since, the two conditions in the above Theorem are satisfied for the softmax model, we can estimate/learn the parameters of the softmax model directly by PL maximization, which minimizes the ARE distance between the true and estimated conditional probabilities.

4. A MODIFIED MULTI-CLASS PROBABILITY ESTIMATOR

The likelihood (or the relative entropy) cost that uses class membership information during training is shown to yield highly efficient estimators. However, when we use the class membership information to estimate M-class posterior probabilities [2], or the $M-1$ [4] probabilities by getting rid of the inherent redundancy, the convergence rate is typically very slow [5, 4]. To increase efficiency in learning, we map the problem to estimation of $\lceil \log_2 M \rceil$ binary probabilities. Though, the procedure implies definition of a possibly more difficult mapping to be learned and some loss of information, it can provide easy bit-by-bit coding options in certain applications and as we show by simulations yields highly satisfactory performance. For example, consider a 4-level pulse amplitude modulation (PAM) data transmission system transmitting $\{-3, -1, 1, 3\}$. In our scheme, qw first decide the sign of the transmitted symbol, then its amplitude. In other words, we use 2 bits: b_0 to represent the sign bit and b_1 the amplitude. When estimating the binary probabilities, we only need to estimate $P_\theta(C_1|y)$ as $P_\theta(C_2|y) = 1 - P_\theta(C_1|y)$, hence in implementation resulting in $\lceil \log_2 M \rceil$ network outputs instead of the $M-1$ required in modified softmax. This mapping of the problem implies that the network shown in Figure 1 also satisfies the conditions of the Theorem, (5) and (6) as this network is now a special case of the MLP classifier studied in [1]. We also show by simulations that, by this implementation the convergence rate is increased considerably compared to softmax, and the scheme is also very effective in tracking abrupt changes. To derive the least relative entropy (LRE)

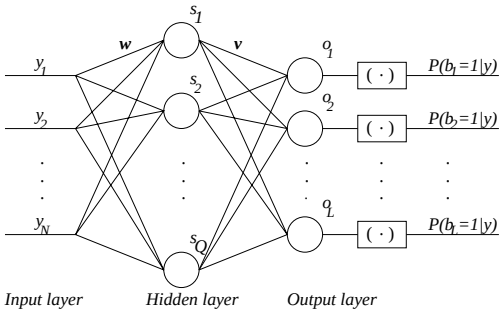

Figure 1: Modified Multi-class Probability Estimator

algorithm for the modified multi-class probability estimator, assume a MLP with N inputs, Q hidden units, and $L = \lceil \log_2 M \rceil$ outputs. For the activation functions of the hidden and output layer, we use the hyperbolical tangent activation function, and define $-\ln \mathcal{L}_n^p(\theta) = \sum_{i=1}^{n} l_i(\theta)$ to write the cost function as

$$l_n(\theta) = \sum\nolimits_{i=1}^{L} \left(\frac{1+b_i}{2} \ln p_\theta^n + \frac{1-b_i}{2} \ln(1-p_\theta^n) \right) \tag{10}$$

where $p_\theta^n \equiv P_\theta(b_i = 1|\mathbf{y}_n))$ and we assumed that the transformation $\frac{1}{2}[(\cdot)+1]$ is applied to transform network outputs to probability measures. Here, $\theta = \{w, v\}$ is the network parameter vector. Let w_{ij} denote the weight from the ith input to the jth hidden neuron, v_{jl} denote the weight from the jth hidden neuron to the lth output, s_j denote the output after the activation funtion at the jth hidden neuron, i.e., $s_j = \tanh(\sum_{i=1}^{N} w_{ij} y_i)$. Gradient descent minimization of the negative log PL cost function for this network results in the following weight update equations:

$$v_{jl}(n+1) = v_{jl}(n) + \mu_1 s_j(n) e_l(n) \quad (11)$$

$$w_{ij}(n+1) = w_{ij}(n) + \mu_2 y_i(n) g_j(n) \sum_{l=1}^{L} v_{jl}(n) e_l(n) \quad (12)$$

for $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, Q$, $l = 1, \ldots, L$, with $g_j(n) = 1 - s_j^2(n)$. Here, μ_1, μ_2 are the step sizes, $e_l(n)$ is the error signal at the lth network output, $e_l(n) = d_l(n) - o_l(n)$.

5. SIMULATION RESULTS

The performances of the LRE algorithm derived above is compared with that of the softmax and the MSE based MLP classifier for a 4 level equalization problem. The classification problem is posed as, given the channel observations $y(n)$ at the output of a nonlinear channel, determine the transmitted symbol $x(n)$ that takes values from $\{-3, -1, 1, 3\}$. The channel is selected as $y(n) = y_l(n) + \alpha y_l^2(n) + \eta(n)$ where $y_l(n)$ is generated by the response $H(z) = 1 + 0.5z^{-6}$, $\eta(n)$ is the zero mean white Gaussian noise, and the PAM communication system uses 8 bits per sample with Nyquist pulse shaping. The performances of the three algorithms are compared at 20 dB signal to noise ratio (SNR) (SNR is defined in terms of the *input* signal power to the noise variance). We use a 3-8-1 MLP neural network to implement the MSE-MLP equalizer, and a 3-8-3 MLP for softmax. For the LRE, the 4-level equalization problem is reduced to estimation of 2 binary probabilities by binary coding by using the mapping described in Section 4. Thus, a 3-8-2 MLP classifier is used for the task.

To compare the best performances of different algorithms, the step sizes μ_1, μ_2 are separately chosen for each classifier. The convergence curves (with $\alpha = -0.02$) shown in Figure 2 are averaged over 25 independent runs. In Figure 2, we can see that the LRE based equalizer has a much faster convergence rate than the softmax equalizer and a similar, slightly slower convergence rate than the MSE-MLP equalizer. To show the recovery property of our reduced complexity LRE equalizer, we introduce an abrupt change (an exact sign change) in the channel characteristics at the 5000th iteration. In Figure 3, we can observe that, after the abrupt change, LRE based equalizer recovers very rapidly, softmax also tracks quite effectively though it has high error due to its initial slow convergence. The MSE-MLP equalizer, however, recovers much slower. Hence the modified LRE provides a good tradeoff in providing considerably faster convergence than the multi-level classifier that uses a normalization stage such as softmax and its convergence rate approaches that of MSE based classifier which uses information much more effectively for the multi-class

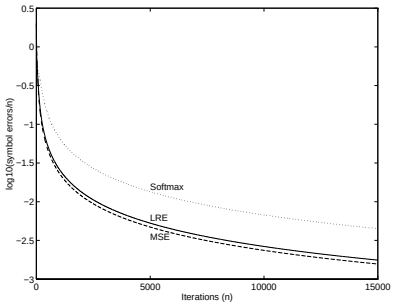

Figure 2: Convergence curves for (i) softmax (ii) MSE-MLP (iii) LRE-MLP equalizers

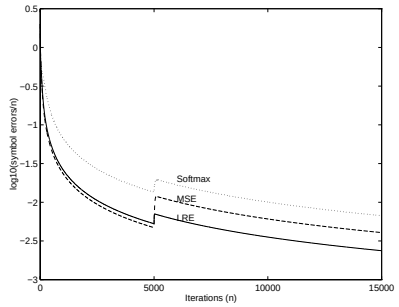

Figure 3: Tracking performance for (i) softmax (ii) MSE-MLP (iii) LRE-MLP equalizers

case. However, by inheriting the well-formed property [7] for learning on the PL (or ARE) cost with a MLP network as shown in [1], LRE provides considerable advantages in tracking performance over the MSE based MLP classifier.

6. REFERENCES

[1] T. Adalı, X. Liu, and M. K. Sönmez, "Conditional distribution learning with neural networks and its application to channel equalization," *IEEE Trans. Signal Processing*, vol. 45, no. 4, pp. 1051-1064, Apr. 1997.

[2] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," *NATO ASI Series, vol. F68, Neurocomputing*, pp. 227-236.

[3] D.R. Cox, "Partial likelihood," *Biometrika*, vol. 62, pp. 69-72, 1975.

[4] M. Hintz-Madsen, M.W. Pedersen, L.K. Hansen, and J. Larsen, "Design and Evaluation of Neural Network Classifiers," in *Proc. IEEE Workshop Neural Networks for Signal Proc., VI*, Kyoto, Japan, Sep. 1996, pp. 223-232.

[5] N. Li, T. Adalı, and X. Liu, "Finite alphabet least relative entropy algorithm for channel equalization," in *Proc. World Congress on Neural Networks*, Washington, DC, July 1995, vol. 3, pp. 76-79.

[6] B. Wang, T. Adali, X. Liu, and J. Xuan, "Partial likelihood for real-time signal processing with finite normal mixtures," in *Proc. IEEE Workshop on Neural Networks for Signal Proc. VIII*, Cambridge, England, Sep. 1998.

[7] B. S. Wittner and J. S. Denker, "Strategies for teaching layered networks classification tasks," *Neural Info. Proc. Systems* (Denver, CO), 1988, pp. 850-859.