10-701/15-781 Machine Learning, Fall 2003

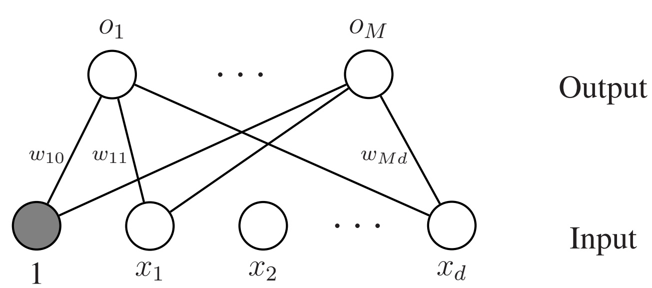
Homework 2
Out: October 2, 2003 **Due: start of class October 14, 2003**

If you have questions, please contact Jiayong Zhang <zhangjy@cs.cmu.edu>.

---

1. (Error Function) The sum-of-squares error is the most common training criterion for neural nets primarily because of its analytical simplicity. Nevertheless, there are many other possible choices of error function which can be considered depending on the particular application. In this excise you will explore one of such alternatives, which is designed to better approximate the sample error.

   Consider the following simple two-layer neural network for M-category classification.

   The network has M output units. Each has a sigmoid activation function. There are no hidden units. Given a set of training samples $\{(\mathbf{x}_n, y_n)\}_{n=1}^N, y_n \in \{1, \ldots, M\}$, the sum-of-squares criterion E_{MSE} can be written as

   $$E_{MSE} = \sum_{n=1}^{N} E_n = \sum_{n=1}^{N} \sum_{m=1}^{M} \left(t_m^{(n)} - o_m^{(n)}\right)^2$$

   where $\mathbf{t}^{(n)} = (t_1^{(n)}, \ldots, t_M^{(n)})$ is the binary valued target vector associated with the n-th sample

   $$t_m^{(n)} = \begin{cases} 1, & m = y_n \\ 0, & \text{otherwise.} \end{cases}$$

   The sample error is defined as $E_{MCE} = \sum_n \ell_n$, where

   $$\ell_n = \begin{cases} 0, & o_{y_n}^{(n)} = \max_j o_j^{(n)} \\ 1, & \text{otherwise.} \end{cases}$$

   (a) Briefly state possible strengths and weaknesses of E_{MCE} compared to E_{MSE}.

(b) Define

$$L_n = \frac{1}{1+e^{-\xi(d_n+\alpha)}},$$
$$d_n = -o_{y_n}^{(n)} + \left[\frac{1}{M-1}\sum_{m,m\neq y_n} o_m^{(n)\eta}\right]^{1/\eta}.$$

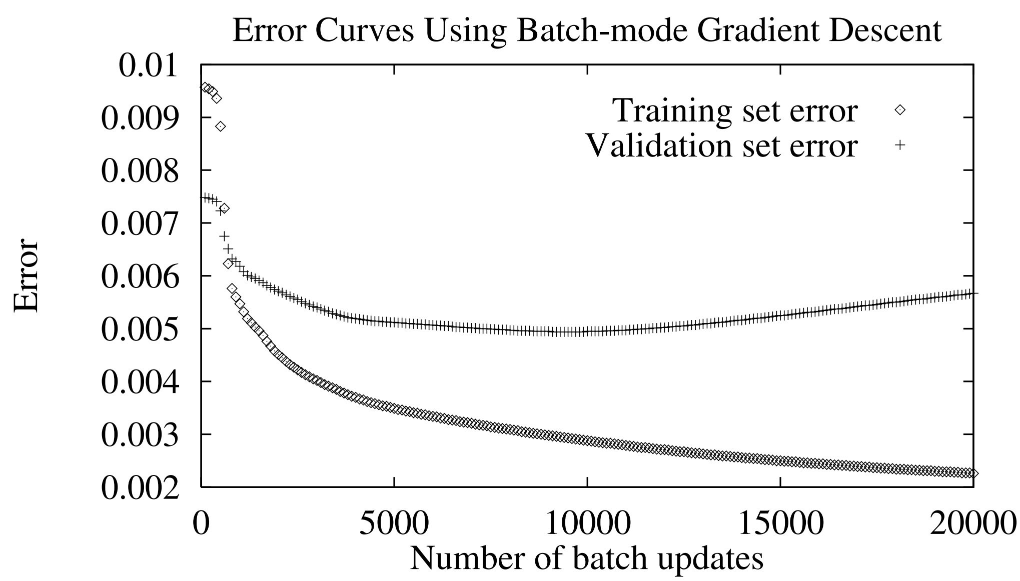
Briefly explain why L_n can be used to approximate ℓ_n by choosing appropriate parameters ξ, α and η.

(c) Find the network update rule when using the criterion function $E = \sum_n L_n$.

(d) What problem would you expect to arise in network training if we choose $\xi \gg 0$?

(e) Can you think of any continuous function other than L_n that is also able to approximate ℓ_n closely?

2. (Cross-validation) In a medical diagnosis problem, we want to train a neural network using the Back-propagation algorithm. In order to determine the amount of training, a simple validation technique is employed. Specifically, we randomly split the available samples into two equal sized parts, one for training and the other for validation. The error curves on these two sets are shown in the following plot. Note that the training error decreases monotonically against the number of batch updates, whereas the validation error does not.



Suppose now that we were to retrain the same neural network using exactly the same algorithm, but using ten times as much available data (50% for training and 50% for test).

(a) Would you expect the training curve to be different? If so, draw what you would expect. You only need to give a qualitative sketch. In either case, explain your reasoning.

(b) Would you expect the validation curve to be different? If so, draw what you would expect. You only need to give a qualitative sketch. In either case, explain your reasoning.

Suppose now that we replace the simple validation with m-fold cross-validation.

(c) True or false, because each of the test sets are independent, the validation error curves from each fold are also independent. Explain your reasoning.

(d) (**Optional, 5 pts**) True or false, there is an *a priori* good choice of m for us to decide the amount of training. Explain your reasoning.

3. (PAC Learning) Consider the hypothesis class H_{rd2} of "regular, depth-2 decision trees" over n Boolean variables. A "regular, depth-2 decision tree" is a depth-2 decision tree (a tree with four leaves, all distance 2 from the root) in which the left and right child of the root are *required to contain the same variable*. For instance, the following tree is in H_{rd2}.

```
          x3
        /    \
     x1        x1
    /  \      /  \
   +    -    -    +
```

(a) As a function of n, how many syntactically distinct trees are there in H_{rd2}? By "syntactically distinct", we mean trees that look different but may still represent the same function.

(b) Give an upper bound for the number of examples needed in the PAC model to learn any target concept in H_{rd2} with error ϵ and confidence δ.

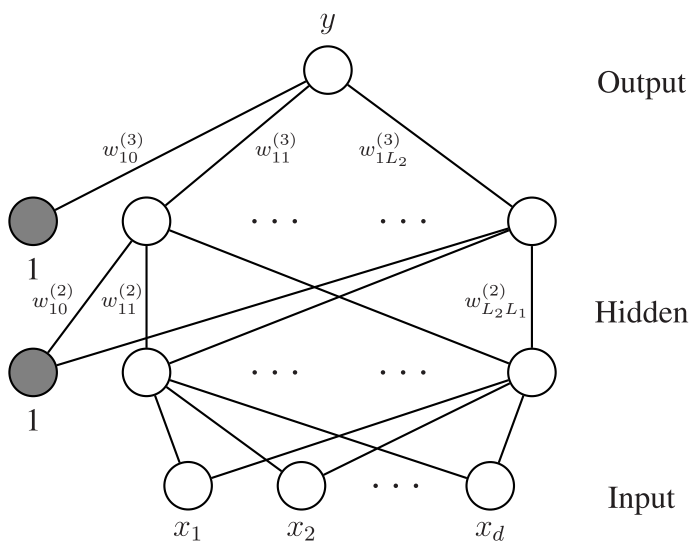
(c) Consider the following WEIGHTED-MAJORITY algorithm for the class H_{rd2}. You begin with all hypotheses in H_{rd2} assigned an initial weight equal to 1. Every time you see a new example, you predict based on a weighted majority vote over all hypotheses in H_{rd2}. Then, instead of eliminating the inconsistent trees, you cut down their weight by a factor of 2. How many mistakes will this procedure make at most, as a function of n and the number of mistakes of the best tree in H_{rd2}?

(d) (**Optional, 5 pts**) Derive the number of semantically distinct trees in H_{rd2}, which leads to a tighter bound in (b).

4. (**Optional, 10 pts**, Radial Basis Function) Consider the following four-layer neural network.

This network consists of an input layer, two hidden layers, and an output layer. The first hidden layer has L_1 units, each of which computes the Gaussian radial basis function

$$o_j^{(1)} = \phi_j(\mathbf{x}) = \exp\left\{-\frac{\|\mathbf{x}-\boldsymbol{\mu}_j\|^2}{2\sigma_j^2}\right\} \qquad (j = 1, \ldots, L_1),$$



where $\mathbf{x}$ is the d-dimensional input vector with elements x_i, and $\boldsymbol{\mu}_j$ is the vector determining the center of basis function ϕ_j. The second hidden layer and the output layer consist of units with activation functions $f^{(2)}$ and $f^{(3)}$ respectively.

$$o_j^{(l)} = f^{(l)}(net_j) = f^{(l)}\left(\sum_{i=1}^{L_{l-1}} o_i^{(l-1)} w_{ji}^{(l)} + w_{j0}^{(l)}\right) \qquad (l = 2, 3).$$

Note that the forms of $f^{(l)}$ are left unspecified. You are given a set of training data $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$.

(a) Suppose $y_i \in (0, +\infty)$, specify some network structure and parameter settings† that allow the entire network to exactly implement the kernel regression estimate:

$$y(\mathbf{x}) = \frac{\sum_n y_n \exp\{-\|\mathbf{x} - \mathbf{x}_n\|^2/2h^2\}}{\sum_n \exp\{-\|\mathbf{x} - \mathbf{x}_n\|^2/2h^2\}}.$$

(b) Suppose $y_i \in \{-1, +1\}$, and we further assume that both $p(\mathbf{x}|y = +1)$ and $p(\mathbf{x}|y = -1)$ follow the unimodal multivariate normal distribution with isotropic covariance

$$p(\mathbf{x}|y = \pm 1) = \frac{1}{(2\pi)^{d/2}\lambda_{\pm 1}^d} \exp\left\{-\frac{\|\mathbf{x} - \boldsymbol{\gamma}_{\pm 1}\|^2}{2\lambda_{\pm 1}^2}\right\}.$$

Specify some network structure and parameter settings† that allow the entire network to output exactly the posterior probability $P(y = +1|\mathbf{x})$ when the number of training samples $N \to \infty$.

† Including the number of hidden units $\{L_1, L_2\}$, the form of activation functions $\{f^{(2)}, f^{(3)}\}$, and the settings of $\{\boldsymbol{\mu}_j, \sigma_j^2, w_{ji}^{(l)}\}$.