CS 156a Learning Systems - Homework #2

Ling Li, ling@cs.caltech.edu

October 24, 2000

2.1 Growth Function and VC Dimension

(i) For any N different x's, the two intervals can generate three categories of output patterns:

(a) Act as 0 interval. That is, no x is included in either interval. Thus all $g_\alpha(x)$ are -1 and the number of such pattern is 1.

(b) Act as 1 interval. That is, x's in two intervals can be covered by only 1 interval. This happens when two intervals overlap or adjoin or or no x is in one interval. However, to be different from previous category, some x is included in the intervals. The number of such patterns is $\binom{N+1}{2}$.

(c) Act as 2 intervals. Two intervals separate from each other and in each interval there's at least one x. The number of such patterns is $\binom{N+1}{4}$.[1]

Thus, the total number of patterns this learning model can produce on any N different x's is

$$m(N) = 1 + \binom{N+1}{2} + \binom{N+1}{4} = \frac{1}{24}(N^4 - 2N^3 + 11N^2 + 14N + 24).$$

Here we list some values of $m(N)$:

N	1	2	3	4	5	6
$m(N)$	2	4	8	16	31	57

Thus $D = 5$, and the VC bound is $N^{D-1} + 1 = N^4 + 1$. We can verify the bound by

$$N^{D-1} + 1 - m(N) = \frac{1}{24}N(N-1)(23N^2 + 25N + 14) \geq 0.$$

(ii) Let's consider seven different points on a circle.

- For each output pattern, we can group neighbor points on the circle with same output. For instance, in Figure 1 on page 2, the outputs of points 1, 2, 4, 6 are $+1$ and those of 3, 5, 7 are -1; And points 1 and 2 are neighbors. So there are 6 groups: $\{1, 2\}$, $\{3\}$, $\{4\}$, $\{5\}$, $\{6\}$ and $\{7\}$, half of them are related with output $+1$. The number of groups must be even.

[1] Number all x's in ascending order. x's with number i to j are included in one interval, and those with number k to l in another interval. Without loss of generality, $1 \leq i \leq j \leq k \leq l \leq N$. When two intervals separate from each other, we have $j < k - 1$. Thus $1 \leq i < j + 1 < k < l + 1 \leq N + 1$. So i, $j + 1$, k, and $l + 1$ are four different numbers within $1..(N + 1)$, which leads to $\binom{N+1}{4}$.

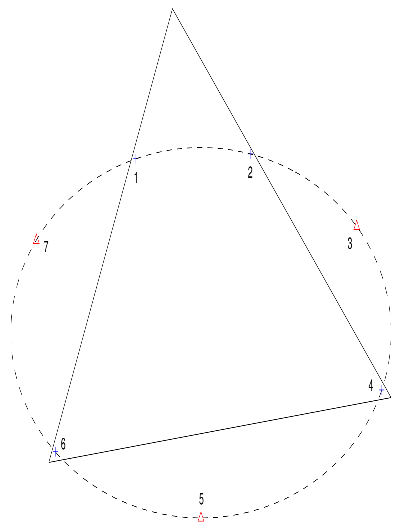


Figure 1: Small red triangles stand for points with output -1, and blue pluses stand for points with output $+1$.

- For 7 points, the maximum number of groups is 6. Thus there are at most 3 groups with output $+1$. With one triangle we can put those groups with output $+1$ in it and leave all the groups with output -1 outside it, just as demonstrated in Figure 1.
- Thus for some 7 points, the 'triangle' model can implement all output patterns. That is, the VC dimension for this model is bigger than 7, $D \geq 8$.

(iii) Consider a direct extending of two intervals: 'three intervals.' Analyzing similarly, we get

$$\begin{aligned} m(N) &= 1 + \binom{N+1}{2} + \binom{N+1}{4} + \binom{N+1}{6} \\ &= \frac{1}{720}(N^6 - 9N^5 + 55N^4 - 75N^3 + 304N^2 + 444N + 720). \end{aligned}$$

Here we list some values of $m(N)$:

N	1	2	3	4	5	6	7	8
$m(N)$	2	4	8	16	32	64	127	247

Thus $D = 7$, and the VC bound is $N^{D-1} + 1 = N^6 + 1$. And again, we find out

$$N^{D-1} + 1 - m(N) = \frac{1}{720} N(N-1)(719N^4 + 728N^3 + 673N^2 + 748N + 444) \geq 0.$$

2.2 VC Dimension of Perceptrons

The mathematical model of a perceptron with d inputs $x_1, x_2, \ldots, x_d$ is

$$y = \operatorname{sgn}(W\mathbf{x}),$$

where $\mathbf{x} = (-1, x_1, \ldots, x_d)^T$, and $W = (\theta, w_1, \ldots, w_d)$ is the weights matrix. $\mathrm{sgn}(\cdot)$ is the sign function. The sign of 0 can either be defined as 0 or 1, which doesn't affect following proof.

(i) $(d+1)$ points. Define $(d+1)$ points as below:

$$X = \left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(d+1)}\right) = \begin{pmatrix} -1 & -1 & -1 & \ldots & -1 \\ 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{pmatrix},$$

where X is a $(d+1) \times (d+1)$ matrix with each column a point. The determinant of X is -1 so X is invertible. Thus for any desired output pattern $Y = \left(y^{(1)}, y^{(2)}, \ldots, y^{(d+1)}\right)$, we can design a perceptron with matrix

$$W = YX^{-1},$$

which means $Y = WX$, or for $i = 1..(d+1)$,

$$y^{(i)} = W\mathbf{x}^{(i)} = \mathrm{sgn}\left(W\mathbf{x}^{(i)}\right).$$

So for these $(d+1)$ points, we can design perceptrons with different weights to fully dichotomize them.

(ii) $(d+2)$ points. Since each point is a vector of $(d+1)$ dimension, $(d+2)$ vectors have to be linearly dependent. Without loss of generality, we can say

$$\mathbf{x}^{(d+2)} = \sum_{i=1}^{d+1} \lambda_i \mathbf{x}^{(i)}.$$

Thus for any weights matrix W,

$$W\mathbf{x}^{(d+2)} = \sum_{i=1}^{d+1} \lambda_i W\mathbf{x}^{(i)}.$$

We can design an output pattern:

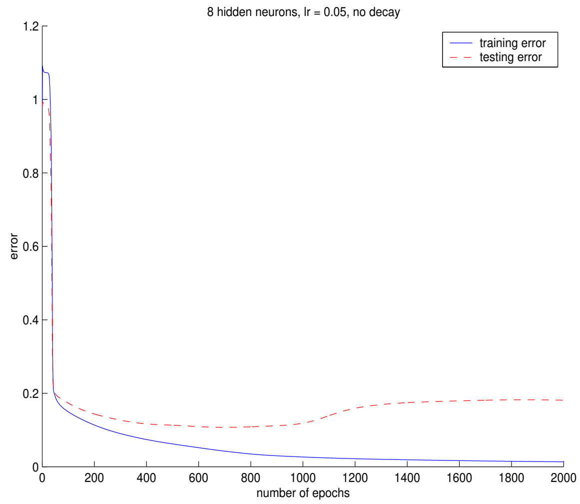
$$y^{(i)} = \begin{cases} \mathrm{sgn}(\lambda_i), & i = 1..(d+1); \\ -1, & i = d+2. \end{cases} \tag{1}$$

If some W can implement this pattern, $W\mathbf{x}^{(i)}$ must have the same sign with $y^{(i)}$, i.e., with the same sign with λ_i, for $i = 1..(d+1)$. Thus $\lambda_i W\mathbf{x}^{(i)} \geq 0$ for $i = 1..(d+1)$, and

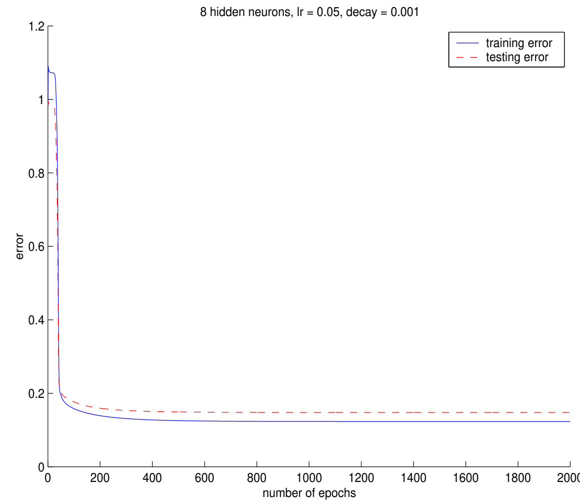
$$W\mathbf{x}^{(d+2)} = \sum_{i=1}^{d+1} \lambda_i W\mathbf{x}^{(i)} \geq 0.$$

So $y^{(d+2)}$ cannot be -1. Pattern (1) cannot be implemented by any W. This just says there are no $(d+2)$ points that can be fully dichotomized.

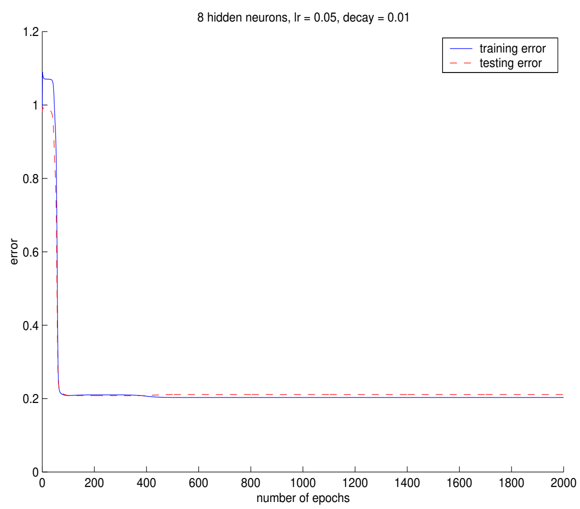
Summarizing above, we get the VC dimension of a perceptron with d inputs is $D = d+2$.



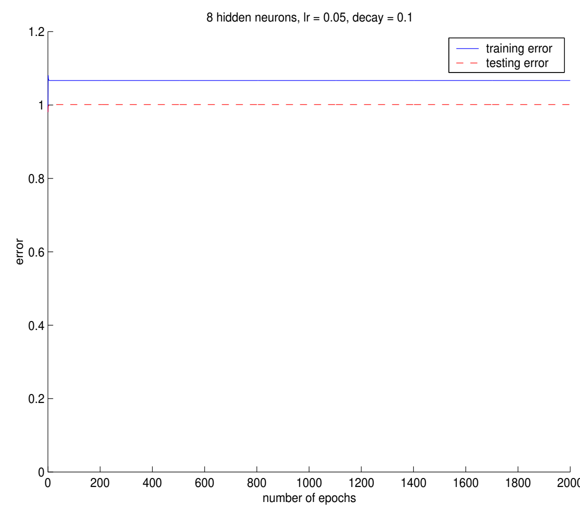
(a) without weight decay



(b) weight decay $\lambda = 0.001$



(c) weight decay $\lambda = 0.01$



(d) weight decay $\lambda = 0.1$

Figure 2: The training errors and testing errors of network with/without weight decay. There're 8 hidden neurons and the learning rate is fixed at 0.05.

2.3 Weight Decay

The training and testing errors with and without weight decay are in Figure 2. Comparing those 4 plots, we find out:

- Testing error in Figure 2(a) first decreases and then increases roughly after epoch 900. That means the learning model has been over-fitted to the training data and thus increases the testing error. We can also explain it in a way related to model complexity. The learning procedure continues exploring the weights (together with the threshold) space and increases the model complexity, thus decreases the probability of correctly predicting the out-of-sample error, or in other words, impairs the ability of generalization.

- With weight decay, we can confine the weights to relatively smaller ranges and thus improve the generalization. That explains why the training and testing errors are closer in Figure 2(b) and 2(c) than that in Figure 2(a).

- However, if λ is too big, which makes the weights decay too fast (and finally are all near zero), the learning model is no longer capable of learning the samples. The trial of Figure 2(d) shows this. In fact, at epoch 2000 all the weights of the first layer of the trained network is almost zero (in the degree of 10^{-71}) and the output of the network is always 0.1205.

2.4 Validation for Early Stopping

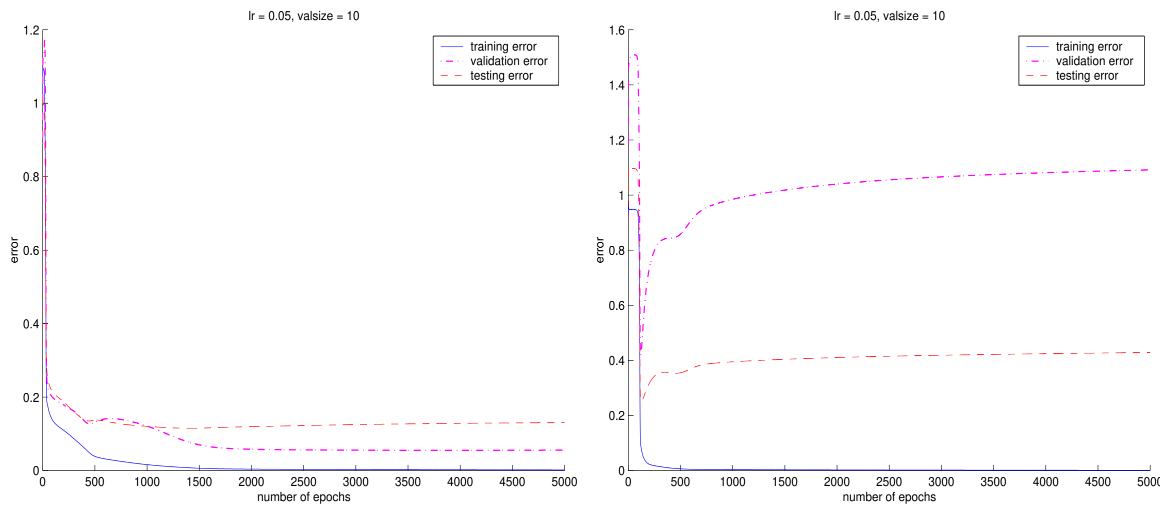


Figure 3: 25 examples in the training set and 10 examples in the validation set.

(i) Two plots of the training, testing and validation errors are in Figure 3. Since the in-sample data are *randomly* divided, the behavior of the trained network on the test set and validation set also varies, some with smaller errors and some with bigger errors. We may see that (in shape) the testing error is more like the validation error than the training error.

(ii) For 10 runs, the no-validation errors and early-stopping errors are

#	1	2	3	4	5	6	7	8	9	10
E_{NV}	0.1718	1.1523	0.1061	0.4286	0.7521	1.0178	1.0156	0.5208	1.2319	0.6665
E_{ES}	0.2157	0.9035	0.2355	0.2574	0.2766	0.9633	0.4884	0.5239	0.9722	0.2070

Thus the average values are $\overline{E_{NV}} = 0.7063$ and $\overline{E_{ES}} = 0.5044$. The latter is smaller, which means better generalization due to learning validation.

(iii) For 10 examples in the training set and 25 in the validation set, $\overline{E_{NV}} = 1.1996$, $\overline{E_{ES}} = 0.6319$. Plots are in Figure 4.

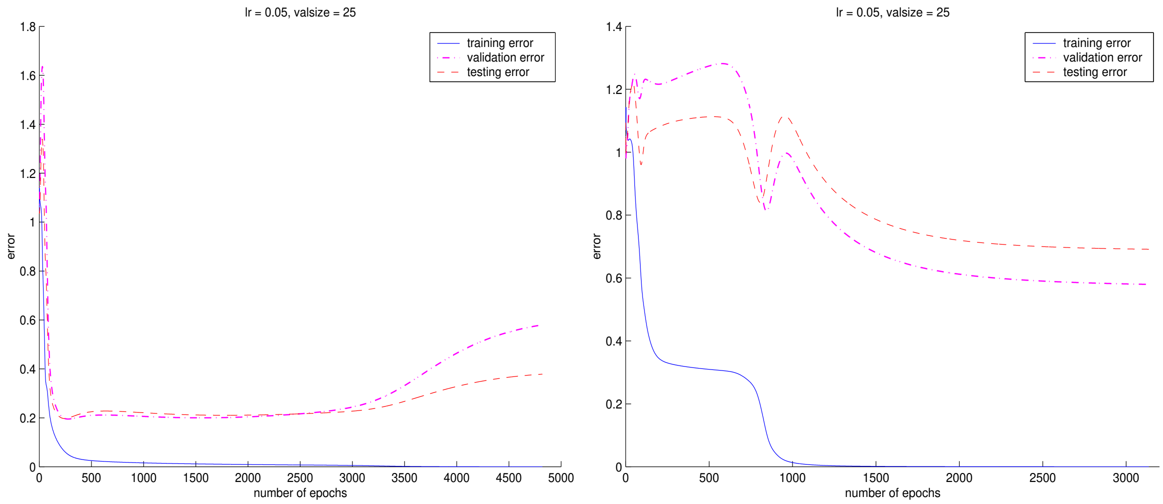


Figure 4: 10 examples in the training set and 25 examples in the validation set.

#	1	2	3	4	5	6	7	8	9	10
E_{NV}	1.3795	0.3620	1.0060	0.4911	1.4485	0.3786	0.3431	0.6915	1.3337	4.5624
E_{ES}	0.9948	0.2395	0.2501	0.3357	0.9870	0.1994	0.2867	0.6915	1.2072	1.1267