Lecturer: Ron Rivest *Scribe: Andrew Kirmse*

Outline

- The VC-dimension of neural networks.
- Asymptotic error rates of neural networks.

18.1 The VC-dimension of a neural network

If we could compute the VC-dimension of a neural network, then we could determine an upper bound on the number of examples we would need to see before the neural net is trained to classify according to a target concept. Below we will find the VC-dimension of a somewhat restricted set of neural nets. Instead of using the PAC model of learning, we will derive error and confidence bounds that depend on a single parameter ϵ. The results in this section are from [2]; the theorem numbers match those in the paper. Similar proofs appear in [4] and [1].

First we will consider neural networks that consist of an acyclic set of edges and a set of *computation nodes*. Each node computes some arbitrary function from its real-valued inputs to $\{\pm 1\}$, but there is only one output node for the entire network. This may seem to be much more restrictive than our earlier definition of neural nets, but the computation nodes in this model are quite general, in that the nodes aren't limited to using the weighted sum of their inputs.

Let F_i be the class of functions computable at node i, and let F be the class of functions computed by the entire net as the F_i vary. Then given a sample S of size m, we define

$$\begin{aligned} \Delta_F(S) &= \text{the number of dichotomies induced by } F \text{ on } S. \\ \Delta_F(m) &= \max_{S,\ |S|=m} \Delta_F(S) \end{aligned}$$

($\Delta_F(S)$ is what we called $\Pi_{\mathcal{C}}(S)$ in our earlier discussion of VC-dimension, and $\Delta_F(m)$ is what we called $\Pi_{\mathcal{C}}(m)$.) If our neural network has N nodes, then we have the following first result:

Theorem 1 *Define* $d = \sum_{i=1}^{N}$ *VC-dim*(F_i). *Then*

$$\Delta_F(m) \leq \prod_{i=1}^{N} \Delta_{F_i}(m) \leq \left(\frac{Nem}{d}\right)^d$$

Proof: For a given sample S, the maximum number of labelings as the F_i vary is the number of ways of choosing the net's output based on its inputs. Certainly we can't have more labelings than the number of ways of choosing each node's outputs independently; we may have fewer if some choices induce the same labelings. Each node has $\Delta_{F_i}(m)$ ways of labeling S. So the whole network has at most $\Delta_{F_1}(m)\Delta_{F_2}(m)\ldots$ ways of labeling S. This gives the first inequality in the theorem.

To establish the second inequality, define $d_i = \text{VC-dim}(F_i)$. Then $d = \sum_{i=1}^{N} d_i$ by definition. Now recall a result that we derived in our first lecture on VC-dimension. If F has (finite) VC-dimension k, then $\Delta_F(m) \leq \left(\frac{em}{k}\right)^k$. Applying this to each F_i individually, we have that

$$\prod_{i=1}^{N} \Delta_{F_i}(m) \leq \prod_{i=1}^{N} \left(\frac{em}{d_i}\right)^{d_i}$$

This product is maximized when all the d_i are equal; *i.e.* when $d_i = d/N$ for all i. So

$$\begin{aligned} \prod_{i=1}^{N} \Delta_{F_i}(m) &\leq \prod_{i=1}^{N} \left(\frac{em}{d/N}\right)^{d/N} \\ &= \left(\frac{em}{d/N}\right)^d \\ &= \left(\frac{Nem}{d}\right)^d \end{aligned}$$

■

For the remainder of this section, we will restrict ourselves to computation nodes that compute linear threshold functions. That is, a node outputs $+1$ if the weighted sum of its inputs is greater than some threshold, and -1 otherwise. The following corollary gives an upper bound on the VC-dimension of such nets:

Corollary 3 *Let E be the number of edges in a neural net, and $W = E + N$ the total number of weights (one per edge plus the thresholds). If $N \geq 2$, then*

- $\Delta_F(m) \leq \left(\frac{Nem}{W}\right)^W$ *for $m \geq W$.*
- *VC-dim*$(F) < 2W \log(eN)$

Proof:

- Recall that the VC-dimension of the class of half-spaces in $\mathcal{R}^n$ is $n+1$. Since each node computes a linear threshold in $\mathcal{R}^k$, where k is the number of inputs to the node, the class of functions computed by a node has VC-dimension at most $k+1$. Thus $d = \sum d_i = \sum$ VC-dim$(F_i) \leq E + N = W$. Applying Theorem 1 gives the inequality.
- If we take $m \geq 2W \log(eN)$, then we can show that

$$\left(\frac{Nem}{W}\right)^W < 2^m$$

  as long as $N \geq 2$. Using the first part of the corollary, this shows that $\Delta_F(m) < 2^m$. Thus no set of m points is shattered, so F has VC-dimension less than m.

■

Here are two other results from [2]:

Corollary 4 *If $m \geq \frac{32W}{\epsilon} \log \frac{32W}{\epsilon}$ and we can find a choice of weights that correctly labels at least a fraction $1 - \epsilon/2$ of the m training examples, then with probability at least $1 - 8^{-1.5W}$, the net will correctly classify at least a fraction $1 - \epsilon$ of examples drawn from the same distribution as the training examples.*

Theorem 4 *The class of one hidden layer linear threshold nets with k hidden nodes and n inputs has VC-dimension at least $2\lfloor\frac{k}{2}\rfloor n$.*

Note that this bound is approximately equal to the total number of weights if the net is highly connected (n edges from the inputs to each of the k hidden nodes, plus one edge from each hidden node to the output node, for a total of $k(n+1)$ in a fully-connected net).

We never mentioned back propagation in the discussion above. The results hold for an arbitrary training algorithm; in particular, back propagation performs better in practice than these results would suggest.

18.2 Asymptotic error rates of neural networks

In this section we are interested in determining the error rate of a neural network on a sample as the size of the sample and the number of hidden nodes in the net increase. We would like to compare our actual (or *empirical*) error rate with the optimal error rate (also known as the *Bayes risk*). The results below are from [3].

In the discussion below we will consider neural nets with k hidden nodes in a single hidden layer and d *fixed* inputs. There is a single output node which outputs either 0 or 1. Each node computes a step function σ of the weighted sum of its inputs:

$$\sigma(x) = \begin{cases} 1 & \text{if } x \geq b_i \\ -1 & \text{if } x < b_i \end{cases}$$

We label each edge from an input to a hidden node by a_i, and each edge from a hidden node to the output node by c_i. Considering a, b, and c as column vectors, we can write the output of the net as a function f of its inputs and weights:

$$f(x, a, b, c) = \sum_{i=1}^{k} c_i \sigma(a_i^T x + b_i) + c_0$$

Here x is a vector of the inputs, and c_0 is a constant input to the output node.

We can use these neural nets in situations where the inputs don't completely determine a classification. For example, suppose that the inputs are characteristics of a lung X-ray, and the output should predict whether or not a tumor is present. Under such circumstances, there is only a probability (not a certainty) of an output being correct given a set of inputs. It's easy to see that the best possible classification rule is the one that agrees with the true outcome most of the time. Let (X, Y) be an input/output pair. Then we can write the optimal classification rule as

$$g^*(x) = \begin{cases} 0 & \text{if } \Pr\{Y = 0 | X = x\} \geq 1/2 \\ 1 & \text{if } \Pr\{Y = 1 | X = x\} > 1/2 \end{cases}$$

Any given neural net corresponds to some classification rule, given by the mapping of its inputs to its outputs. Let g_n be the classification rule of a neural net after it has seen n examples. This rule will have a certain error rate on the set of training examples, denoted by $L(g_n)$, and another (possibly different) error rate on the set of all examples in the sample space, written $r(g_n)$. These can be written formally as

$$L(g_n) = \Pr\{g_n(X) \neq Y | \text{ some particular training sequence}\}$$

$$r(g_n) = E(L(g_n)) = \Pr\{g_n(X) \neq Y\}$$

The optimal classification rule g^* has the minimum possible error rate on the whole sample space. We denote this rate by L^*, which is also known as the *Bayes risk* of the sample space.

Definition 1 *A sequence g_n of classification rules is universally consistent if*

$$\lim_{n\to\infty} r(g_n) = L^*$$

regardless of the distribution on the sample space X.

A sequence g_n of classification rules is strongly universally consistent if

$$\lim_{n\to\infty} L(g_n) = L^*$$

with probability one, regardless of the distribution on the sample space X.

Strong universal consistency does indeed imply universal consistency: if a sequence of values chosen from some distribution always converges to a certain value, then the sequence of expected values chosen from the same distribution converges to the same value.

Let $g^*_{k,n}$ denote the optimal neural net with k hidden nodes after the first n examples.

Theorem 1 *If the number of hidden nodes k satisfies $k \to \infty$ and $\frac{k \log n}{n} \to 0$ as $n \to \infty$, then with probability one,*

$$\lim_{n\to\infty} L(g^*_{k,n}) = L^*$$

This theorem is somewhat useful in that it provides a guarantee of strong universal convergence, even when no neural net is a perfect classifier. However, Theorem 1 says nothing about the rate of convergence; the following theorem is more useful in practice:

Theorem 2 *If* $k = \sqrt{\frac{n}{d \log n}}$, *then*

$$r(g^*_{k,n}) - L^* = O\left(\left(\frac{d \log n}{n}\right)^{\frac{1}{4}}\right)$$

This theorem tells you roughly how fast the optimal neural net with k hidden nodes after n examples converges to the optimal error rate. Notice that the rate of convergence has a constant exponent; in particular, it doesn't depend on the number of inputs d.

References

[1] Anthony and Briggs. *Computational Learning Theory.* Cambridge University Press, 1992.

[2] Eric B. Baum and David Haussler. What size net gives valid generalization? In *Advances in Neural Information Processing Systems I*, pages 81-90. Morgan Kaufmann, 1989.

[3] András Faragó and Gábor Lugosi. Strong universal consistency of neural network classifiers. *IEEE Transactions on Information Theory*, IT-39(4):1146-1151, July 1993.

[4] Michael J. Kearns and Umesh V. Vazirani. *An Introduction to Computational Learning Theory*, pages 64-67. MIT Press, 1992.