3.7 An Application to Neural Networks

We conclude this chapter by giving a useful general lemma that bounds $VCD(\mathcal{C})$ when each concept in the class $\mathcal{C}$ is actually a **composition** of simpler concepts. Such classes arise frequently — for instance, a DNF formulae is simply a (very constrained) composition of boolean conjunctions (the constraint being that we can only compute disjunctions of conjunctions). After giving this lemma, we then apply it to obtain upper bounds on the sample size required for PAC learning neural networks.

To formalize a general notion of concept composition, let G be a **layered** directed acyclic graph. By this we mean that the nodes of G can be partitioned into layers, and the directed edges of G go only from a node at layer ℓ to a node at layer $\ell+1$. We let n be the number of nodes at layer 0, and we assume that all of these have indegree 0. We think of these n layer 0 nodes as being the inputs to the graph. We also assume that there is only a single node of outdegree 0 at the highest level of the graph, and we think of this node as being the output node of the graph. All internal (that is, non-input) nodes have the same indegree r, and we let s denote the number of internal nodes. Figure 3.5 shows an example of such a layered graph with $n = 8$, $s = 8$ and $r = 3$.

Now let $\mathcal{C}$ be a concept class over r-dimensional Euclidean space $\Re^r$. Suppose we take such a layered graph G, and we label each internal (that is, non-input) node N_i with a concept $c_i \in \mathcal{C}$. Then such a labeled

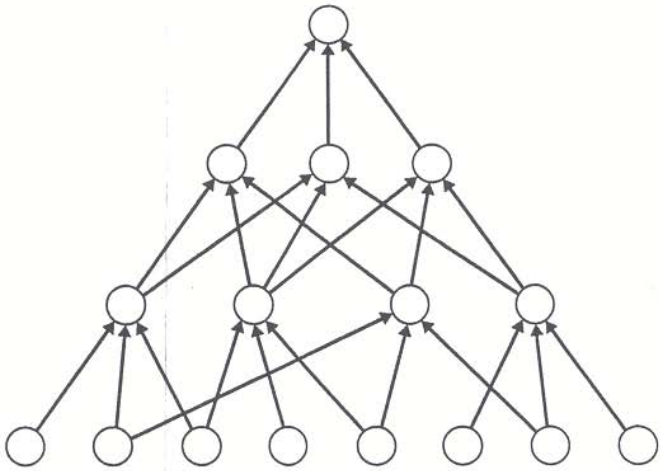

Figure 3.5: *A layered directed acyclic graph.*

graph represents a concept over n-dimensional Euclidean space $\Re^n$ in the obvious way: if we label each of the n input nodes at layer 0 with a real number, then starting with layer 1 we can compute the value at each node N_i by applying the concept c_i labeling node N_i to the values computed at the nodes feeding N_i. (Note that although concepts in $\mathcal{C}$ are defined over $\Re^r$, the input values feeding nodes at level 2 and higher will actually only be from $\{0,1\}^r$.) The output of the entire labeled graph is the binary value computed at the output node. We will call the class of all concepts over $\Re^n$ that can be obtained by labeling G with concepts from $\mathcal{C}$ the **G-composition of $\mathcal{C}$**, which we denote $\mathcal{C}_G$.

Theorem 3.6 *Let G be a layered directed acyclic graph with n input nodes and $s \geq 2$ internal nodes, each of indegree r. Let $\mathcal{C}$ be a concept class over $\Re^r$ of VC dimension d, and let $\mathcal{C}_G$ be the G-composition of $\mathcal{C}$. Then $VCD(\mathcal{C}_G) \leq 2ds\log(es)$.*

Proof: The idea is to first bound the function $\Pi_{\mathcal{C}_G}(m)$. Let us fix any

set S of m input vectors $\vec{x}_1, \ldots, \vec{x}_m \in \Re^n$ to the graph G (thus, each $\vec{x}_i$ determines a complete setting of the n input nodes of G). For this fixed input set S, if we now also label each node in G with a concept from $\mathcal{C}$, then for each $\vec{x}_i$ we have completely determined the binary values that will be computed at every node of G when the input is $\vec{x}_i$. Let us call the collection of all the values computed at each node, for each $\vec{x}_i \in S$, a *computation* of G on S. Thus, a computation can be represented by labeling each internal node with the vector in $\{0,1\}^m$ of the values computed at that node on the m vectors in S. Then the set of all possible computations of G on S is obtained by ranging over all possible choices of labels from $\mathcal{C}$ for the nodes of G. Note that two computations of G on S differ if and only if the value computed at some node on some input from S differs in the two computations. Clearly, $|\Pi_{\mathcal{C}_G}(S)|$ is bounded by the total number of possible computations of G on S, which we shall denote $T_{\mathcal{C}_G}(S)$.

To bound $T_{\mathcal{C}_G}(S)$, let G' be the subgraph obtained by removing the output node N_o from G. Let $T_{\mathcal{C}_{G'}}(S)$ denote the total number of computations of G' on S. Each fixed computation of G' can be extended to at most $\Pi_{\mathcal{C}}(m)$ computations of G, because fixing the computation of G' determines for each $1 \leq i \leq m$ the input $\vec{y}_i \in \{0,1\}^r$ that is fed to N_o when $\vec{x}_i$ is fed to G, and at most $\Pi_{\mathcal{C}}(m)$ labelings of $\vec{y}_1, \ldots, \vec{y}_m$ can be obtained at N_o by varying the choice of concept from $\mathcal{C}$ placed at N_o. Thus we obtain that for any S, $T_{\mathcal{C}_G}(S) \leq T_{\mathcal{C}_{G'}}(S) \times \Pi_{\mathcal{C}}(m)$, and a simple inductive argument establishes

$$|\Pi_{\mathcal{C}_G}(S)| \leq T_{\mathcal{C}_G}(S) \leq (\Pi_{\mathcal{C}}(m))^s \leq \left(\frac{em}{d}\right)^{ds}$$

where the second inequality comes from the polynomial bound on the $\Pi_{\mathcal{C}}(m)$ given in Section 3.4. Since S was arbitrary, this bound in fact holds for $\Pi_{\mathcal{C}_G}(m)$.

Thus in order for $\mathcal{C}_G$ to shatter m points, the inequality $(em/d)^{ds} \geq 2^m$ must hold. Conversely, if $(em/d)^{ds} < 2^m$ for some m, then m is an upper bound on $VCD(\mathcal{C}_G)$. It is easy to verify that this latter inequality holds for $m = 2ds\log(es)$ provided $s \geq 2$. □(Theorem 3.6)

To apply Theorem 3.6 to the problem of PAC learning neural networks, we simply let the function at each node in the graph G be a **linear threshold function**. If the indegree is r, such a function is defined by real **weights** $w_1, \ldots, w_r \in \Re$ and a **threshold** $\Theta \in \Re$. On inputs $x_1, \ldots, x_r \in \Re$ the function outputs 1 if $\sum_{i=1}^{r} w_i x_i \geq \Theta$, and outputs 0 otherwise. We call G the underlying **architecture** of the neural network.

Now as we mentioned in Section 3.3, it is known that the VC dimension of the class of linear threshold on r inputs is $r+1$. By Theorem 3.6 we find that the Vapnik-Chervonenkis of the class of neural networks with architecture G is at most $2(rs+s)\log(es)$, and combined with Theorem 3.3, we obtain:

Theorem 3.7 *Let G be any directed acyclic graph, and let C_G be the class of neural networks on an architecture G with indegree r and s internal nodes. Then the number of examples required to learn C_G is*

$$O\left(\frac{1}{\epsilon}\log\frac{1}{\delta} + \frac{(rs+s)\log s}{\epsilon}\log\frac{1}{\epsilon}\right).$$

3.8 Exercises

3.1. Compute the VC dimension of the class of boolean conjunctions of literals over $\{0,1\}^n$.

3.2. Consider the concept class over the Euclidean plane $\Re^2$ consisting of the interior regions of circles; thus, the positive examples of each concept form a disk in the plane. Compute the VC dimension of this class. Compute the VC dimension of the class of interiors of triangles in the plane.

3.3. Show that there is no 1-decision list over $\{0,1\}^n$ computing the exclusive-or function $x_1 \oplus x_2$. Then show that the VC dimension of