Stats 231C - Homework 2

Peter Racioppo (103953689)

April 27, 2020

Theorem. Let $s : R \mapsto R$ satisfy:
(i) $\lim_{\alpha\to-\infty} s(\alpha) = 0$ and $\lim_{\alpha\to\infty} s(\alpha) = 1$, and
(ii) s is differentiable with $s(\alpha_0) \neq 0$ at some α_0.
Then, for any $L \geq 1$ and $W \geq 10L - 14$, there is a feedforward network with L layers, W parameters, computation units with activation s (except the output unit, which is a linear threshold unit), for which $VCdim(H) \geq \lfloor L/2 \rfloor \lfloor W/2 \rfloor$.

Proof. (I will follow the proof in [1] and [2], and try to add some exposition.) We need to show that there exists a set of $M \times N$ points that can be shattered by a neural network with $O(N)$ weights and $O(M)$ layers. Following the proof of Theorem 22, we define a set of parameters $\{a_i\}, i \in [1, N]$ s.t. $a_i = \sum_{j=1}^{M} a_{i,j} 2^{-j}$, with $a_{i,1}, ..., a_{i,M} \in \{0, 1\}$. We consider input points $x \in B_N \times B_M$, where $B_N = \{e_i : 1 \leq i \leq N\}$. Given an input $x = (e_l, e_m)$, the network outputs a matrix of $a_{i,j}$ values. There are NM inputs of the form (e_l, e_m). The power set of the NM points has cardinality 2^{NM}, so each element of the power set maps to an $n \times M$ matrix of $a_{i,j}$ values. This shows that the network can shatter a set of NM points. The proof proceeds in steps: we must first demonstrate how to calculate the $\{a_i\}$ from the inputs, and then how to extract the M bits from the N parameters, that is how to extract each $a_{i,j}$. In what follows, we suppose that the network input is $x = ((u_1, ..., u_N), (v_1, ..., v_M)) = (e_l, e_m)$.

Let $c_k = \sum_{j=k}^{M} 2^{k-1-j} a_{l,j}$. Observe that $c_{k-1} = \sum_{j=k-1}^{M} 2^{k-1-1-j} a_{l,j}$, so $2c_{k-1} = \sum_{j=k-1}^{M} 2^{k-1-j} a_{l,j} = \sum_{j=k}^{M} 2^{k-1-j} a_{l,j} + a_{l,k-1} = c_k + a_{l,k-1}$. Thus, $c_k = 2c_{k-1} - a_{l,k-1}$.

Recall that $a_{i,j} \in \{0, 1\}$, so $a_i \geq 0, \forall i$. Now, $c_k = a_{l,k}/2 + \sum_{j=k+1}^{M} 2^{k-1-j} a_{l,j}$. Thus, $a_{l,k} = 1 \Rightarrow c_k \geq 1/2$. On the other hand, the term $\sum_{j=k+1}^{M} 2^{k-1-j} a_{l,j}$ can be no more than $\sum_{j=k+1}^{M} 2^{k-1-j} < \sum_{j=2}^{\infty} 2^{-j} = \sum_{j=1}^{\infty} 2^{-j} - 1/2 = 1/2$. Thus, $a_{l,k} = 0 \Rightarrow c_k < 1/2$. In summary, $a_{l,k} = 1 \iff c_k \geq 1/2$. Thus, $a_{l,k} = 1 \iff \text{sgn}(c_k - 1/2) = 1$ and $a_{l,k} = 0 \iff \text{sgn}(c_k - 1/2) = 0$, so we can write $a_{l,k} = \text{sgn}(c_k - 1/2)$. Note that here we are defining the range of sgn as $\{0, 1\}$ rather than the usual $\{-1, 1\}$.

Thus, we have the recursion relation:
$c_k = 2c_{k-1} - a_{l,k-1}$
$a_{l,k} = \text{sgn}(c_k - 1/2)$

The initial conditions are given by $c_1 = \sum_{j=1}^{M} 2^{-j} a_{l,j} = a_1$ and $a_{l,1} = \text{sgn}(c_1 - 1/2) = \text{sgn}(a_1 - 1/2)$. Now, a_1 can be computed with a single computational unit, i.e. $\sum_{i=1}^{N} u_i a_i = a_l$. Each additional term from 2 to M can be computed by a single computational unit in a single layer, in total $2(M-1)+1$. However, we can avoid computing c_M, and thus perform the computation in fewer layers, by defining $b = \text{sgn}(2c_{M-1} - a_{1,M-1} - \sum_{i=1}^{M-1} v_i) = \text{sgn}(c_M - \sum_{i=1}^{M-1} v_i)$. If $m \neq M$, meaning the last entry of e_m is 0, then $\sum_{i=1}^{M-1} v_i = 1$. Note that $c_k \leq 1, \forall k$, so $\sum_{i=1}^{M-1} v_i = 1 \Rightarrow b = \text{sgn}(c_M - 1) \leq \text{sgn}(0) = 0$. (To show that $c_k \leq 1$ we perform a strong induction on k. Note that $a_i = \sum_{j=1}^{M} a_{i,j} 2^{-j} < \sum_{j=1}^{\infty} 2^{-j} = 1$. In particular, $c_1 = a_1 < 1$. Furthermore, $a_{l,k} \in \{0, 1\}$. We have two base cases: if $c_1 < 1/2, a_{1,k} = 0$ and $c_2 = 2c_1 < 1$. If $1/2 < c_1 < 1$, $a_{1,k} = 1$ and $c_2 < 2 - 1 = 1$. Now, suppose by way of strong induction, that $c_p < 1, \forall 1 \leq p \leq k$. Then, if $c_k < 1/2, a_{1,k} = 0$ and $c_{k+1} = 2c_k < 1$, whereas if $1/2 < c_k < 1$, $a_{1,k} = 1$, and $c_{k+1} < 2 - 1 = 1$, which completes the induction.) On the other hand, if $m = M$, meaning the last entry of e_m is 1, then $\sum_{i=1}^{M-1} v_i = 0$, so $b = \text{sgn}(c_M)$. But $\text{sgn}(c_M) = \text{sgn}(a_{l,M})$ by the foregoing inductive argument, so $b = \text{sgn}(a_M) = a_M$. Thus we can write $b = a_{1,M}\mathbf{1}(m = M)$.

It remains to show how to recover each of the $a_{i,j}$ from the inputs. We can write, for every row, $a_{l,m} = b \vee \bigvee_{i=1}^{M-1} (a_{l,i} \wedge v_i)$. Each v has exactly one nonzero entry, so this set of disjunctions must pick out exactly one $a_{l,i}$.

We use 1 layer to compute a_1, $2(M2) + 1$ layers to compute the recursion relation, and 2 layers to compute the disjunctions, for a total of $2M$ layers. We use $N + 1$ parameters to compute a_1, $5(M - 2) + 2$ parameters for the terms in the recursion relation, and 5M parameters to compute the disjunctions, for a total of $10M + N - 7$ parameters. (The factors of five here comes from a complex network architecture used in Bartlett, Maiorov, and Meir (1998). Simply applying the recursion relation for c_k and $a_{l,k}$ up to $a_{l,M}$ would only increase this quantity by a constant factor.) Thus, for a feedforward network with L layers and W parameters, we can set $M = \lfloor L/2 \rfloor$ and $N = W + 7 - 10M$, so $N \geq \lfloor W/2 \rfloor$ provided $W - W/2 \geq 10L/2 - 7$, that is $W \geq 10L - 14$. Now since the network shatters NM points and $M = \lfloor L/2 \rfloor$, $N \geq \lfloor W/2 \rfloor$, we have that VC-dim $\geq \lfloor L/2 \rfloor \lfloor W/2 \rfloor$.

This proof uses linear threshold units and linear units. Lemma 8.10 in Anthony and Bartlett shows that any function S computed on a network of only linear threshold units and linear units can be computed on a network with activation functions s satisfying the conditions in the theorem (namely that $\lim_{\alpha \to -\infty} s(\alpha) = 0$ and $\lim_{\alpha \to \infty} s(\alpha) = 1$, and s is differentiable with $s(\alpha_0) \neq 0$ at some α_0).

References

P. Bartlett, V. Maiorov, R. Meir, *Almost Linear VC-Dimension Bounds for Piecewise Polynomial Networks*. Neural Computation, 1999.

M. Anthony and P. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.