VC-Dimension Bounds for Neural Circuits

Cameron Musco

February 26, 2017

Cameron: Note that these are my personal notes and everything is very rough and unpolished. If things are confusing or wrong, please ask me.

1 Motivating Example: The Selection Problem

We want to prove circuit size lower bounds. Eventually for neural circuits, but for today, just consider *feedforward* networks with gates of the form $\phi(\sum w_i x_i - \theta)$, where $\phi(\cdot) : \mathbb{R} \to \mathbb{R}$ is any function. Think of $\phi(\cdot)$ as being a sigmoid function or threshold function.

Very simple problem: $Selection(x, y) = x(y)$ – the bit of x at the y^{th} position, where y encodes the index in binary. To get y to be binary if the gates output real numbers, allow it to have an arbitrary threshold.

1.1 AND/OR Circuit Lower Bound

Theorem 1. *Any AND/OR circuit requires $\Omega(n)$ gates to compute Selection.*

Proof. Given any Selection circuit, after fixing the x inputs, you have a circuit that maps $y \in \{0,1\}^{\log n}$ to a bit. Any gate connected to any x input can only compute 2 functions of the y inputs depending on if it is connected to no positive inputs or at least 1 positive input. But the circuit needs to compute all possible binary functions on an input of size $\log n$, depending on the x input. There are $2^{2^{\log n}} = 2^n$ such functions. So if m is the number of gates in the circuit, we must have $2^m > 2^n$ and hence $m \geq n$. □

Remark 2. *Theorem 1 also holds for recurrent networks.*

Remark 3. *A similar argument can be made, for example, for sigmoid circuits with edge weights and thresholds of bounded precision – say representable using b bits.*

Proof. We can think of fixing the x inputs to the Selection circuit as just modifying the threshold values for the sigmoid gates. If the edge weights and thresholds can be expressed using b bits, then any summation $< n$ of these weights can be expressed using $logn + b$ bits. So there are only $2^{\log n + b}$ possible gates. So, a circuit with m gates can only compute $2^{(b \log n) \cdot m}$ possible functions after the x input is fixed. So we must have $(b + \log n) \cdot m > n$ and so $m = \Omega(n/(\log n + b))$. □

Unfortunately for more complex gates, the simple reasoning above doesn't extend. It fails even for linear threshold gates with real values weights and biases. The key step we must make is to figure out how to handle gates with *infinite* possible behaviors after fixing the x inputs. This is going to be where VC dimension becomes useful.

2 Intro to VC Dimension

VC (Vapnik-Chervonenkis) dimension is a concept used primarily in learning theory, but also in many areas to measure the complexity of classes of functions.

Let $\mathcal{F}$ be a set (possibly infinite) of functions $f : X \to \{0, 1\}$.

Definition 4 (Shattering)**.** *$\mathcal{F}$ shatters $A \subseteq X$ if for all possible functions $g : A \to \{0, 1\}$, there is some $f \in F$ such that $f(a) = g(a)$ for every $a \in A$.*

Definition 5 (VC Dimension)**.** *The VC Dimension of $\mathcal{F}$, $VC(\mathcal{F})$, is the maximum $|A|$ such that $\mathcal{F}$ shatters set A.*

Example 6 (Half Spaces)**.** *Let $X = \mathbb{R}^2$ and $Y = \{0, 1\}$. If $\mathcal{F}$ is the set of all half spaces in the plane, $VC(\mathcal{F}) = 3$.*

Proof. Any set of 3 noncolinear points in the plane is shattered by $\mathcal{F}$. Further *every set* of 4 points in the plane is not shattered by F. This requires considering 2 cases: 4 points whose convex hull is a quadrilateral, and 4 points whose convex hull is a triangle with one point inside. □

Remark 7. *For $X = \mathbb{R}^d$, VC dimension of half space is $d + 1$.*

3 Using VC Dimension to Bound Circuit Complexity

Here we are following the paper *VC Dimension in Circuit Complexity*, Koiran [Koi96].

3.1 A Fixed Selection Circuit as a Function Class

It is easy to see that a family of circuits defines a family of functions, and the VC dimension of these function classes has been studied extensively. This is generally to try to show that certain classes of circuits can't compute certain functions, or more commonly, to give learning bounds for neural networks. If the VC dimension of a classifier (e.g. a neural net) is bounded then you can prove that after training on a certain number of example input points, the behavior of the net will *generalize well* to future input points.

However, to give circuit size lower bounds for *Selection*, we are going to define a function class based on a *specific fixed circuit.*

Definition 8. *For any circuit $\mathcal{N}$ with n inputs talking values in A and 1 output taking values in B, and any $p < n$, let $\mathcal{F}^p_{\mathcal{N}}$ be defined as any function $f : A^p \to B$ given by fixing the values of the last $(n - p)$ inputs and considering the output of $\mathcal{N}$ as a function of the first p inputs.*

Note that we already used this sort of 'restriction' method in our lower bound proofs in Section 1.1. One way you can think of $\mathcal{F}^p_{\mathcal{N}}$ is as the class of functions computable using a fixed network architecture with $n-p$ programmable parameters – the $n-p$ fixed inputs which change the behavior of the circuit in some way.

Claim 9. *If $\mathcal{N}$ solves the selection problem and $p = n$, then $\mathcal{F}^n_{\mathcal{N}}$ is the set of* all *functions $f : \{0, 1\}^{\log n} \to \{0, 1\}$. Trivially, $VC(\mathcal{F}_{\mathcal{N}}) = \left|\{0, 1\}^{\log n}\right| = n$.*

3.2 Threshold Circuit Lower Bound for Selection

Theorem 10. *Any linear threshold circuit requires $\Omega(n/\log n)$ gates to compute Selection.*

Proof. This is because the VC dimension of a fixed circuit architecture with m threshold gates (and programable thresholds) is $\leq m \log m$ by [BH88]. □

We will now prove this fact.

We first give a sort of generalization of VC dimension, which measures the number of dichotomies that a class of functions can induce over a set.

Definition 11. *For a class of functions $\mathcal{F} : X \to \{0,1\}$, let $\Delta_{\mathcal{F}}(z)$ is the maximum over sets A with $|A| = z$ of $\Delta_{\mathcal{F}}(A)$, which is the number of different ways that A can be partitioned (the number of* dichotomies *induced) using the function.*

Note that $VC(\mathcal{F})$ is the maximum z with $\Delta_{\mathcal{F}}(z) = 2^z$.

Theorem 12 (Theorem 1 of [BH88])**.** *Let $\mathcal{F}$ be the class of functions computed by a fixed feedforward architecture with m nodes, where each node v_i can be chosen to compute any function in the class $\mathcal{F}_i$. Then $\Delta_{\mathcal{F}}(x) \leq \prod_{i=1}^{m} \Delta_{\mathcal{F}_i}(x)$.*

Applying Theorem 12 to a fixed threshold circuit architecture with programmable thresholds gives $\Delta_{\mathcal{F}_i}(z) = z$, since for z inputs, there are z possible functions that can computed by varying the threshold of the circuit (note that as edge weights are fixed, any input is just mapped to a single real number which is thresholded by v_i).

So $\Delta_{\mathcal{F}}(z) \leq z^m$. Thus, $VC(\mathcal{F})$ is the largest z with $z^m \geq 2^z$ so $m \log z \geq z$ so $z/\log z \leq m$. Setting $z = m \log m$ gives $z/\log z = m \log m/(\log m + \log\log m) \approx_2 m$. So this gives the VC bound used in Theorem 10.

Proof of Theorem 12. This is pretty straightforward. Order the nodes $v_1, v_2, ...v_m$ such that v_i only receives input from v_j with $j \leq i$. Then fixing a function on v_1 fixes the inputs seen by v_2. Fixing the function on v_2 then fixes the inputs seen by v_3 etc. So since there are $\prod_{i=1}^{M} \Delta_{\mathcal{F}_i}(z)$ ways to fix all the functions successively like this, we have $\Delta_{\mathcal{F}}(z) \leq \prod_{i=1}^{M} \Delta_{\mathcal{F}_i}(z)$. □

Theorem 13. *Any* recurrent *linear threshold circuit requires $t \cdot m = \Omega(n/\log n)$ where t is the runtime and m is the number of gates.*

Proof. By unrolling the circuit to be feedforward. A lot of these unrolling bounds, while seemingly very naive are actually close to tight. See [KS98]. □

Remark 14. *We can match Theorem 13 up to $\log n$ factors using a circuit with $\tilde{O}(\sqrt{n})$ gates and running in time $\tilde{O}(\sqrt{n})$.*

Remark 15. *We haven't really used the full power of VC dimension yet. You probably could have thought of the linear threshold bound pretty easily without the VC formalism. But the point is that this bound is for gates with infinite possible behaviors (or really exponential if you consider all possible thresholds induced by fixing some subset of the inputs). So it is doing something more powerful than our bounds for gate sets with very limited numbers of possible behaviors.*

3.3 Sigmoidal VC Dimension Bound for Selection

Theorem 16 ([KM97])**.** *Let $\mathcal{F}$ be the set of all functions computed by a* fixed architecture *of m sigmoidal gates with p programable parameters. $VC(\mathcal{F}) = O(m^2p^2)$.*

Corollary 17. *Any sigmoidal circuit requires $\Omega(n^{1/4})$ gates to compute Selection.*

Proof. If there is a circuit $\mathcal{N}$ that computes selection, with m gates, then it has at most m gates connected to the n bits corresponding to x. The connections effectively change the thresholds of these gates, and then apply a function to just x. So we have a circuit with m gates and m programable parameters (the thresholds which are modified by the input connections. We thus have $n = VC(\mathcal{F}_{\mathcal{N}}) \leq m^2p^2 = m^4$ and hence $m \geq n^{1/4}$. □

3.4 Piecewise Rational Lower Bound

Theorem 18. *Any circuit consisting of piecewise rational function gates requires $\Omega(\sqrt{n})$ gates to compute Selection. This is matched by a simple circuit using the linear softstep.*

Proof. This is by the VC dimension bound of m^2 given in [GJ95]. It is open if this VC dimension upper bound can be matched for sigmoidal and other exponential networks. □

Theorem 19. *The VC dimension of the concept class computed by a fixed gate with polynomial activation functions and m free parameters is $\leq m^2$.*

This proof follows from a bound of Warren '68:

Theorem 20. *If $\mathcal{P} = \{p_1, ...p_m\}$ is a set of polynomials of degree $\leq d$ in n variables, with $m \geq n$ the number of consistent sign assignments to $\mathcal{P}$ is $\leq (cdm/n)^n$ for some constant c.*

Let v be the VC dimension of the concept class over a fixed polynomial network with m programable parameters. Consider the set of shattered inputs $a_1, ..., a_v$. By fixing each of those inputs, we get v different polynomial functions over m variables. To shatter A, this set of v different polynomials must take 2^v different sign assignments. So we have by the theorem above:

$$(cdv/m)^m \geq 2^v$$

Now d, if each gate has constant degree is $O(1)^m$ since there are m gates so their composition has degree at most $O(1)^m$. So logging both sizes we have: $v \leq m\log(d) + m\log(v/m)$. Which basically gives $v \leq m^2$.

4 Open Questions

- Bounds on sigmoidal or polynomial gates for recurrent networks? Matching circuit constructions?
- Bounds for stochastic or spiking networks, which don't fit as cleanly into VC dimension model. How do we deal with stochastic functions?
- How can these techniques be used in continuous time, spiking networks? It is said that changing the time delay on synapses has much more 'power' than changing synapse weights. Can this be seen in VC bounds?
- Rule out simulation results via VC dimension bounds, or else use simulation to extend bounds to other network types.

References

[BH88] Eric B Baum and David Haussler. *What size net gives valid generalization?* Computer Research Laboratory, University of California, Santa Cruz, 1988.

[GJ95] Paul W Goldberg and Mark R Jerrum. Bounding the vapnik-chervonenkis dimension of concept classes parameterized by real numbers. *Machine Learning*, 18(2-3):131–148, 1995.

[KM97] Marek Karpinski and Angus Macintyre. Polynomial bounds for vc dimension of sigmoidal and general pfaffian neural networks. *Journal of Computer and System Sciences*, 54(1):169–176, 1997.

[Koi96] Pascal Koiran. Vc dimension in circuit complexity. In *Computational Complexity, 1996. Proceedings., Eleventh Annual IEEE Conference on*, pages 81–85. IEEE, 1996.

[KS98] Pascal Koiran and Eduardo D Sontag. Vapnik-chervonenkis dimension of recurrent neural networks. *Discrete Applied Mathematics*, 86(1):63–79, 1998.