CMSE 890-002: Mathematics of Deep Learning, MSU, Spring 2020

Lecture 18: Fundamental Limits of One-Layer Networks

February 21, 2020

Lecturer: Matthew Hirn

Proof sketch of Theorem 8.7. The first part of Pinkus proof is to reduce the problem over $[0,1]^d$ to a problem over compact subsets of $\mathbb{R}$. First define $\mathcal{N}(\sigma)$ as the one-dimensional analogue of $\mathcal{M}(\sigma)$ (recall $z \in \mathbb{R}$):

$$\mathcal{N}(\sigma) = \text{span}\{\sigma(wz+b) : w \in \mathbb{R}\,,\, b \in \mathbb{R}\}\,.$$

The following proposition proves that solving the problem in one dimension using $\mathcal{N}(\sigma)$ is enough to solve the problem in d-dimensions with $\mathcal{M}(\sigma)$:

Proposition 8.8 (Pinkus 1999, [13])**.** *Let $K \subset \mathbb{R}$ be a compact set, which means K is closed and bounded. Suppose for every such K, for each $G \in \mathbf{C}(K)$, and for each $\epsilon > 0$, there exists a $g \in \mathcal{N}(\sigma)$ such that*

$$\sup_{z \in K} |G(z) - g(z;\gamma)| < \epsilon \qquad (\gamma \text{ are the parameters of } g)\,.$$

Then for each each $F \in \mathbf{C}[0,1]^d$ and each $\epsilon > 0$, there exists an $f \in \mathcal{M}(\sigma)$ such that

$$\sup_{x \in [0,1]^d} |F(x) - f(x;\theta)| < \epsilon\,.$$

We will not prove this proposition, but we will use it, since it allows us to restrict our attention to functions $G \in \mathbf{C}(K)$, with $K \subset \mathbb{R}$, as opposed to $F \in \mathbf{C}[0,1]^d$. The following proposition, combined with Proposition 8.8, does most of the work in proving Theorem 8.7.

Proposition 8.9 (Pinkus 1999, [13])**.** *Let $\sigma \in \mathbf{C}^\infty(\mathbb{R})$ (i.e., σ can be differentiated an infinite number of times) and assume σ is not a polynomial. Let $K \subset \mathbb{R}$ be compact. Then for each $G \in \mathbf{C}(K)$ and each $\epsilon > 0$ there exists a $g \in \mathcal{N}(\sigma)$ such that*

$$\sup_{z \in K} |G(z) - g(z;\gamma)| < \epsilon\,.$$

Proof of Proposition 8.9. It is a known fact that if $\sigma \in \mathbf{C}^\infty(\mathbb{R})$ and σ is not a polynomial, then there exists a point $z_0 \in \mathbb{R}$ for which

$$\sigma^{(k)}(z_0) \neq 0\,, \quad \forall\, k \geq 0\,.$$

The proof is tricky and requires the use of tools from graduate functional analysis, so we omit it. But let us use this fact. Note that the function

$$\frac{\sigma((w+h)z + z_0) - \sigma(wz + z_0)}{h}$$

is in $\mathcal{N}(\sigma)$ for every $h \neq 0$. Letting h tend to zero we obtain:

$$\lim_{h \to 0} \frac{\sigma((w+h)z + z_0) - \sigma(wz + z_0)}{h} = \frac{d}{dw}\sigma(wz + z_0) = z\sigma'(wz + z_0)\,.$$

Furthermore, if we set $w = 0$, we obtain:

$$\frac{d}{dw}\sigma(wz + z_0)\bigg|_{w=0} = z\sigma'(z_0)\,.$$

Therefore the function $\widetilde{p}_1(z) = z\sigma'(z_0)$, and thus the function $p_1(z) = z$ (by rescaling by $1/\sigma'(z_0)$), can be approximated to arbitrary accuracy on K by functions in $\mathcal{N}(\sigma)$.

By similar arguments, we can conclude that the functions

$$\frac{d^k}{dw^k}\sigma(wz + z_0)\bigg|_{w=0} = z^k\sigma^{(k)}(z_0)\,,$$

can be approximated to arbitrary accuracy on K by functions in $\mathcal{N}(\sigma)$, which means that $p_k(z) = z^k$ can be approximated on K to arbitrary accuracy by functions in $\mathcal{N}(\sigma)$. But that is every monomial, which means that every polynomial can be approximated on K to arbitrary accuracy by functions in $\mathcal{N}(\sigma)$.

Now remember the Stone-Weierstrass Theorem from Example 8.4. In fact it holds for any compact set $K \subset \mathbb{R}$, meaning there exists a polynomial $p(z)$ such that

$$\sup_{z \in K} |G(z) - p(z)| < \epsilon/2\,.$$

Let $g \in \mathcal{N}(\sigma)$ approximate $p(z)$ uniformly, which we know we can do by the analysis carried out already in this proof:

$$\sup_{z \in K} |p(z) - g(z;\gamma)| < \epsilon/2\,.$$

It follows that

$$\sup_{z \in K} |G(z) - g(z;\gamma)| \leq \sup_{z \in K} |G(z) - p(z)| + \sup_{z \in K} |p(z) - g(z;\gamma)| < \epsilon\,.$$

□

The proof of Theorem 8.7 is completed by using the fact that compactly supported $\mathbf{C}^\infty(\mathbb{R})$ functions are dense in $\mathbf{C}(\mathbb{R})$, which allows one to remove the restriction that $\sigma \in \mathbf{C}^\infty(\mathbb{R})$ in Proposition 8.9 and replace with the less restrictive assumption $\sigma \in \mathbf{C}(\mathbb{R})$. Theorem 8.7 then follows by applying this modified version of Proposition 8.9 combined with Proposition 8.8. □

8.3.2 Back to the rate of approximation

Recall in Section 8.2 we constructed ReLU networks that interpolated the training data exactly, but needed a number of neurons that was only one less than the number of training points. Any one layer nerual network with a continuous activation function σ that is not a polynomial can also accomplish this feat.

Theorem 8.10 (Pinkus 1999, [13]). *Let $\sigma(z)$ be a continous function that is not a polynomial. Given training data $T = \{(x_i, y_i)\}_{i=1}^N$ with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, there exists a one layer neural network $f \in M(\sigma)$ with exactly N neurons such that $f(x_i; \theta) = y_i$ for all $1 \leq i \leq N$. That is, there exists $w_1, \ldots, w_N \in \mathbb{R}^d$, $b \in \mathbb{R}^N$, and $\alpha \in \mathbb{R}^N$, such that*

$$\sum_{k=1}^{N} \alpha(k)\sigma(\langle x, w_k \rangle + b(k)) = y_i\,, \quad \forall\,, 1 \leq i \leq N\,.$$

Notice that similar to the ReLU activation function we require the number of neurons to be the same as the number of training points.

In the ReLU case, we also were able to obtain the rate of approximation. We can do the same for certain continuous activation functions $\sigma(z)$ that are not polynomials, but not all of them. These results are for $\mathcal{X} = B^d$, the unit ball in $\mathbb{R}^d$, i.e.,

$$B^d = \{x \in \mathbb{R}^d : \|x\|_2 \leq 1\}\,.$$

Like in the extension of the ReLU results, we will consider smooth label functions $F \in \mathbf{C}^s(B^d)$, where $s \geq 1$ indicates the smoothness of $F(x)$ ($s = 0$ just means $F(x)$ is continuous). Let $\mathcal{M}_N(\sigma) \subset \mathcal{M}(\sigma)$ denote one-layer neural networks with at most N neurons, i.e.,

$$f(x; \theta) = \sum_{k=1}^{N} \alpha(k)\sigma(\langle x, w_k \rangle + b(k))\,.$$

In light of Theorem 8.10, here we use N to denote the number of neurons since this is the number of neurons needed just to fit the training data. We also define the $\mathbf{L}^\infty(B^d)$ norm as:

$$\|F\|_{\mathbf{L}^\infty(B^d)} = \sup_{x \in B^d} |F(x)|\,,$$

which we note allows us to write:

$$\sup_{x \in B^d} |F(x) - f(x; \theta)| = \|F - f\|_{\mathbf{L}^\infty(B^d)}\,.$$

We also recall the $\mathbf{L}^p(B^d)$ norm:

$$\|F\|_{\mathbf{L}^p(B^d)} = \left(\int_{B^d} |F(x)|^p \, dx \right)^{1/p}\,.$$

Note that $\|F - f\|^2_{\mathbf{L}^2(B^d)}$ is the squared loss. Finally we recall the $\mathbf{C}^s(B^d)$ norm is:

$$\|F\|_{\mathbf{C}^s(B^d)} = \max_{\|\beta\|_1 \leq s} \sup_{x \in B^d} |\partial^\beta F(x)| .$$

Our first result is the following.

Theorem 8.11 (Pinkus 1999, [13]). *There exist $\sigma \in \mathbf{C}^\infty(\mathbb{R})$ which are sigmoidal and strictly increasing such that for every $F \in \mathbf{C}^s(B^d)$ with $\|F\|_{\mathbf{C}^s(B^d)} \leq 1$,*

$$\inf_{f \in \mathcal{M}_N(\sigma)} \|F - f\|_{\mathbf{L}^p(B^d)} \leq CN^{-s/(d-1)} ,$$

for each $1 \leq p \leq \infty$, $s \geq 1$, and $d \geq 2$. The constant C is independent of F and N.

Theorem 8.11 has nearly the same rate of approximation as "local polynomial neuron" analysis in (30). Like with that result, it shows that increased smoothness in $F(x)$ increases the rate of approximation, but it is still dampened by the dimension d which can be very large in modern machine learning and deep learning applications. Also note, the theorem says there exists a $\sigma \in \mathbf{C}^\infty(\mathbb{R})$ that is sigmoidal and strictly increasing for which the result holds, but it does not say the result holds for all such $\sigma(z)$. In particular, one cannot conclude the result holds for the regular sigmoid function, and in fact Pinkus remarks that the activation functions $\sigma(z)$ for which Theorem 8.11 holds are pathalogical and overly complex, despite them having the nice properties of being infinitely differentiable, sigmoidal, and strictly increasing. Nevertheless, the result gives a worst case bound on the rate of approximation for certain one layer neural networks with smooth sigmoidal activation functions. Notice as well that it holds for all $\mathbf{L}^p$ loss functions, for $1 \leq p \leq \infty$, which includes the squared loss and uniform approximtion.

One may ask, though, are these worst case results on the rate of approximation an artifact of proof technique, and thus not indicative of the true rate of approximation? For the squared loss, it turns out the answer is "no." Here is the remarkable result.

Theorem 8.12 (Maiorov 1999, [14]). *Let $\sigma(z)$ be a continuous activation function that is not a polynomial. Then for each $s \geq 1$, $d \geq 2$, and N, there exists an $F \in \mathbf{C}^s(B^d)$ with $\|F\|_{\mathbf{C}^s(B^d)} \leq 1$, such that*

$$\inf_{f \in \mathcal{M}_N(\sigma)} \|F - f\|_{\mathbf{L}^2(B^d)} \geq cN^{-s/(d-1)} ,$$

for each $s \geq 1$ and $d \geq 2$. The constant c is independent of F and N.

Combining Theorem 8.11 and Theorem 8.12 we see there exists sigmoidal activation functions such that the resulting one layer neural network achieves a rate of approximation of $O(N^{-s/(d-1)})$ for $F \in \mathbf{C}^s(B^d)$, but without additional information on F, one cannot expect to do better than $O(N^{-s/(d-1)})$.

As a final remark, we recall Example 8.4 which dealt with polynomial approximation. By the Stone-Weierstrass theorem we observed the space of all polynomials also have the

universal approximation property. What we did not quantify was the rate of approximation. In fact it turns out that it is the same as the rate in Theorem 8.11. It thus follows, rather definitively, from this and the previous considerations, that if we want to understand the power of neural networks we must move beyond the one layer model.

References

[1] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations*, 2015.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition*, 2009.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105. 2012.

[4] Larry Greenemeier. AI versus AI: Self-taught AlphaGo Zero vanquishes its predecessor. *Scientific American*, October 18, 2017.

[5] Pankaj Mehta, Marin Bukov, Ching-Hao Wang, Alexandre G.R. Day, Clint Richardson, Charles K. Fisher, and David J. Schwab. A high-bias, low-variance introduction to Machine Learning for physicists. arXiv:1803.08823, 2018.

[6] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer-Verlag New York, 2nd edition, 2009.

[7] J. Hartlap, P. Simon, and P. Schneider. Why your model parameter confidences might be too optimistic - unbiased estimation of the inverse covariance matrix. *Astronomy and Astrophysics*, 464(1):399–404, 2007.

[8] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Adaptive Computation and Machine Learning. The MIT Press, 2002.

[9] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[10] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference for Learning Representations*, San Diego, CA, USA, 2015.

[11] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2:303–314, 1989.

[12] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4:251–257, 1991.

[13] Allan Pinkus. Approximation theory of the mlp model in neural networks. *Acta Numerica*, 8:143–195, 1999.

[14] Vitaly E. Maiorov. On best approximation by ridge functions. *Journal of Approximation Theory*, 99:68–94, 1999.

[15] Vitaly E. Maiorov and Allan Pinkus. Lower bounds for approximation by mlp neural networks. *Neurocomputing*, 25:81–91, 1999.

[16] Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. In *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 907–940. PMLR, 2016.