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

Lecture 21: Jump Ahead to Current Research

February 28, 2020

Lecturer: Matthew Hirn

9 Modern theory for ANNs

We now jump ahead ~ 15 years and consider more recent results on the theory of artificial neural networks.

9.1 More differences between one-layer and two-layer ANNs

We begin with a result from 2016 [17] that reinforces the analysis of Maiorov and Pinkus contained in Theorems 8.12 and 8.13. Interestingly, [17] does not cite [14], but we will see the results are of a similar flavor.

Namely, [17] obtains a result that says neural networks with two hidden layers are fundamentally different than those with one hidden layer. In particular, Eldan and Shamir prove there is a simple function on $\mathbb{R}^d$ that can be well approximated by a "small" two-hidden-layer neural network that cannot be approximated by any one-hidden layer neural network, to more than a certain constant accuracy, unless the width of this one-layer network is exponential in the dimension. This function, essentially, gives a concrete example for Theorem 8.12. Let us now state the results in more detail.

By a one-layer network with W neurons (previously we used $m = W$ as the number of neurons, but [17] uses W to indicate the "width" of the network which will not mean the number of neurons, at least as we define them, when we get to two layer networks), the authors mean the same functions as we have discuss previously, namely:

$$f(x;\theta) = \sum_{k=1}^{W} \alpha(k)\sigma(\langle x, w_k\rangle + b(k)) \quad \text{(one hidden layer of width } W\text{)}. \tag{36}$$

By a two-hidden layer network of width W, Eldan and Shamir in fact mean a function like the one used by Maiorov and Pinkus, which is a type of two layer network with a special structure:

$$f(x;\theta) = \sum_{\ell=1}^{W} \alpha(\ell)\sigma\left(\sum_{k=1}^{W} w_{2,\ell}(k)\sigma(\langle x, w_{1,k,\ell}\rangle + b_1(k,\ell)) + b_2(\ell)\right) \tag{37}$$
$$\text{(two hidden layers of width } W\text{)}.$$

Note, this network has a very similar structure as the one depicted in Figure 27, and by our naming convention we would say the first layer has W^2 neurons and the second layer

has W neurons. Eldan and Shamir say it has "width W." There are two assumptions on the activation function σ.

Assumption 9.1 (one-dimensional universality)**.** There exists a constant $c_\sigma \geq 1$ (depending only on σ) such that the following holds: For any Lipschitz function $G : \mathbb{R} \to \mathbb{R}$ with Lipschitz constant L, i.e., $|G(z) - G(z')| \leq L|z - z'|$, which is constant outside the interval $[-R, R]$, and for any $\epsilon > 0$, there exists $w \in \mathbb{R}^m$, $b \in \mathbb{R}^m$, $\alpha \in \mathbb{R}^m$ and $\beta \in \mathbb{R}$ with

$$W \leq c_\sigma R L \epsilon^{-1} ,$$

such that the function

$$g(z) = \beta + \sum_{k=1}^{W} \alpha(k)\sigma(w(k)z - b(k)) ,$$

satisfies

$$\sup_{z \in \mathbb{R}} |G(z) - g(z)| \leq \delta .$$

Remark 9.2. Note this remark is somewhat similar to Proposition 8.8 in that it reduces the main requirements on σ to the one-dimensional setting.

Assumption 9.3. The activation function σ satisfies

$$|\sigma(z)| \leq C(1 + |z|^\alpha) ,$$

for all $z \in \mathbb{R}$ and for some constants $C, \alpha > 0$.

Eldan and Shamir show these assumptions are satisfied by many standard activation functions, including ReLU and sigmoidal activation functions. Using these assumptions, they prove the following theorem.

Theorem 9.4 (Eldan & Shamir 2016, [17])**.** *Suppose σ satisfies Assumptions 9.1 and 9.3. Then there exist universal constants $c, C > 0$ such that the following holds: For every dimension $d > C$, there is a probability measure μ on $\mathbb{R}^d$ and a function $F_o : \mathbb{R}^d \to \mathbb{R}$ with the following properties:*

- *$\|F_o\|_\infty \leq 2$, $\operatorname{supp}(F_o) \subset \{x \in \mathbb{R}^d : \|x\| \leq C\sqrt{d}\}$, and $F_o(x)$ can be written as a two-hidden layer neural network of the form (37) of width*

$$W_{2\,\text{layer}} = C c_\sigma d^{19/4} .$$

- *Every one hidden layer network $f(x;\theta)$ of the form (36) of width at most*

$$W_{1\,\text{layer}} \leq c e^{cd} ,$$

satisifies

$$\mathbb{E}_X[(F_o(X) - f(X;\theta))^2] = \int_{\mathbb{R}^d} (F_o(x) - f(x;\theta))^2 \, d\mu(x) \geq c . \tag{38}$$

Note that if the probability measure μ admits a probability density function $p_X(x)$, then $d\mu(x) = p_X(x)dx$ and (38) is the same squared loss as we have seen previously.

Theorem 9.4 says there are label functions F that can be perefectly represented by neural networks with two hidden layers and with a width that is only polynomial in the dimension d, but cannot even be well approximated by a single hidden layer neural network unless the width of that single layer network is exponential in the dimension d. Thus by increasing the depth of the network by one (i.e., from one hidden layer to two hidden layers), the neural network becomes significantly more efficient in its ability to express certain label functions.

Furthermore, the function $F_o(x)$ is not that complicated. Indeed, it is, roughly speaking, a radial function, meaning that

$$F_o(x) \approx \widetilde{F}(\|x\|_2^2)\,,$$

where $\widetilde{F} : \mathbb{R} \to \mathbb{R}$. Within the two hidden layer network, the first layer approximates the map $x \mapsto \|x\|_2^2$ and the second layer approximates the function $\widetilde{F}$, which is relatively speaking straightforward and not unlike how the proof of Theorem 8.13 is carried out using the Kolmogorov Superposition Theorem. On the other hand, doing all of this in one layer is rather complicated, which results in the exponential growth of the width of the one-layer neural network.

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] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of Go without human knowledge. *Nature*, 550:354—359, 2017.

[4] 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.

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

[6] 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.

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

[8] 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.

[9] 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.

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

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

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

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

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

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

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

[17] 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.

[18] Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review. *International Journal of Automation and Computing*, 14(5):503–519, 2017.

[19] Hrushikesh Mhaskar. Neural networks for optimal approximation of smooth and analytic functions. *Neural Computation*, 8:164–177, 1996.