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

Lecture 22: Compositional Functions

March 9, 2020

Lecturer: Matthew Hirn

9.2 Compositional functions

The proof of Theorem 8.13 showing that two-layer networks require far fewer neurons than one-layer networks, and the example in Section 9.1, both leverage compositional structure to obtain their results. In fact, recent work contained in [18] studies the approximation theoretic capabilities of shallow networks (one-layer networks) versus deep networks for the class of smooth, compositional functions. A prototypical example is the following:

$$
\begin{aligned}
x \in \mathbb{R}^8, \quad F(x) &= F(x(1), \ldots, x(8)) \\
&= H_3(H_{21}(H_{11}(x(1), x(2)), H_{12}(x(3), x(4))), \\
&\qquad H_{22}(H_{13}(x(5), x(6)), H_{14}(x(7), x(8)))),
\end{aligned} \tag{39}
$$

where $F : \mathbb{R}^8 \to \mathbb{R}$, but each function $H_\lambda : \mathbb{R}^2 \to \mathbb{R}$ for $\lambda \in \{11, 12, 13, 14, 21, 22, 3\}$. This function $F(x)$ can be represented by a binary tree graph, as in Figure 28.

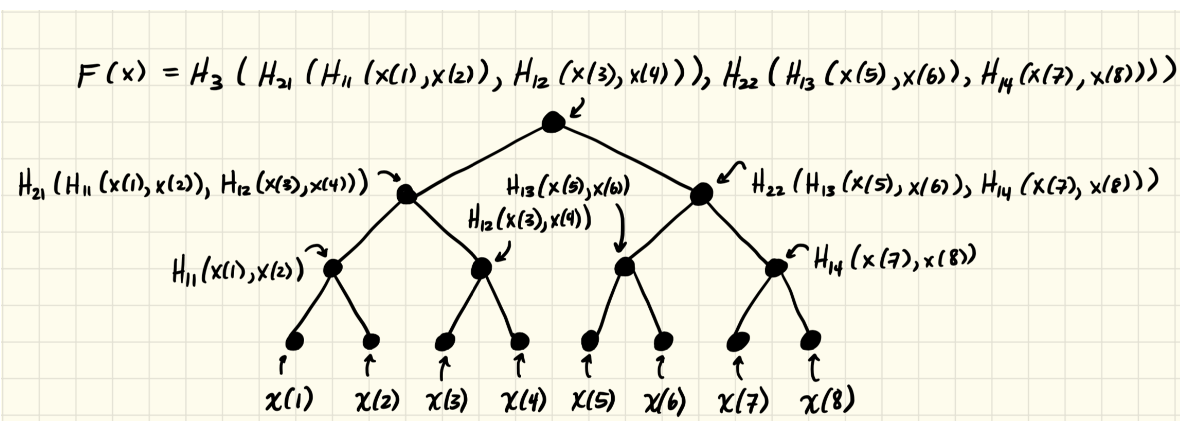

Figure 28: *Illustration of the compositional function in (39) as a binary tree graph.*

Compositional functions such as (39) are "local," meaning that $F(x)$ consists of a compositional heirarchy of functions that only depend on at most q variables; in the case of the example (39), $q = 2$. While both one-layer networks and deep networks are universal function approximators, only deep networks can take advantage of the compositional structure of functions (if the function has this structure). We saw this phenomenon to some extent in the result of Maiorov and Pinkus, Theorem 8.13, which used the Kolmogorov Superposition

Theorem to write any $F \in \mathbf{C}[0,1]^d$ as a compositional function. Here we will explicitly assume the function has such a structure and quantify what we mean by the statement that deep networks can take advantage of this structure.

To that end let us again consider the space of functions $\mathbf{C}^s[0,1]^d$. To be as precise as possible, we note that [18] uses the a different norm on $\mathbf{C}^s[0,1]^d$, which is different than the norm we introduced earlier, but which is equivalent to (33). Anyway, here is the new norm:

$$\|F\|_{\mathbf{C}^s[0,1]^d} = \sum_{\|\beta\|_1 \leq s} \|\partial^\beta F\|_{\mathbf{L}^\infty[0,1]^d} \,,$$

where we remind the reader that

$$\|F\|_{\mathbf{L}^\infty[0,1]^d} = \sup_{x \in [0,1]^d} |F(x)| \,.$$

Now let us define

$$\begin{aligned}\mathbf{C}_2^s[0,1]^d = \{ &\text{ all compositional functions } F(x) \text{ defined on } [0,1]^d \\ &\text{that have a binary tree architecture as in Figure 28,} \\ &\text{and for which the constituent functions } H_\lambda \in \mathbf{C}^s[0,1]^2 \} \,.\end{aligned}$$

We recall the space $\mathcal{M}_N(\sigma) = \mathcal{M}_{N,d}(\sigma)$ of one layer networks with N neurons and that take as input vectors $x \in \mathbb{R}^d$, which consists of functions $f(x;\theta)$ of the form:

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

We remark that the number of trainable parameters in this network is:

$$\begin{aligned}\#\text{ of trainable parameters } &= \text{ weight vectors } w_k + \text{ biases } b(k) + \text{ final weights } \alpha(k) \\ &= dN + N + N = (d+2)N \,.\end{aligned}$$

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.