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

Lecture 23: Deep Approximation of Compositional Functions I

March 11, 2020

Lecturer: Matthew Hirn

As a parallel to the space of compositional functions $\mathbf{C}_2^s[-1,1]^d$ defined above, we now define a corresponding class of deep networks $\mathcal{D}_{N,2}(\sigma)$. The space $\mathcal{D}_{N,2}(\sigma)$ will consist of all deep networks that use the activation function σ and themselves have a binary tree architecture. However, the network having a binary tree architecture does not mean the architecture applies at the level of an artificial neuron; let us explain. Recall that an artificial neuron is the function:

$$\eta(x) = \sigma(\langle x, w \rangle + b) \, .$$

Define a node $\overline{\eta}(x)$ as, essentially, a one hidden layer neural network that has been embedded in a larger network:

$$\overline{\eta}(x) = \sum_{k=1}^{N} \eta_k(x) = \sum_{k=1}^{N} \alpha(k)\sigma(\langle x, w_k \rangle + b(k)) \, .$$

A deep network $f \in \mathcal{D}_{N,2}$ has the binary tree architecture with respect to its *nodes*, meaning that it consists of many nodes composed together according to some binary tree, and each node $\overline{\eta}(z)$ takes as input a two-dimensional vector $z \in \mathbb{R}^2$. The sub-index N means that each node is in $\mathcal{M}_{m,2}(\sigma)$ with

$$m = N/|V| \, , \quad V = \text{ non-leaf vertices of the binary tree.}$$

That is, $\overline{\eta} \in \mathcal{M}_{m,2}(\sigma)$ and each such node has $m = N/|V|$ neurons; figure 29 illustrates the idea. The following proposition shows the number of trainable parameters of a network $f \in \mathcal{D}_{N,2}$ is $4N$.

Proposition 9.5. *Let $d = 2^J$ for some $J \geq 0$. Then the number of trainable parameters of a network $f \in \mathcal{D}_{N,2}(\sigma)$ is $4N$.*

Proof. If $d = 2^J$ then the number of leaves in the binary tree is $d = 2^J$ and the number of non-leaf nodes is (using a formula for geometric series):

$$\sum_{j=0}^{J-1} 2^j = \frac{1-2^J}{1-2} = 2^J - 1 = d - 1 \, .$$

Thus each of these $d-1$ nodes has m neurons with

$$m = N/|V| = N/(d-1) \Longrightarrow d - 1 = N/m \, .$$

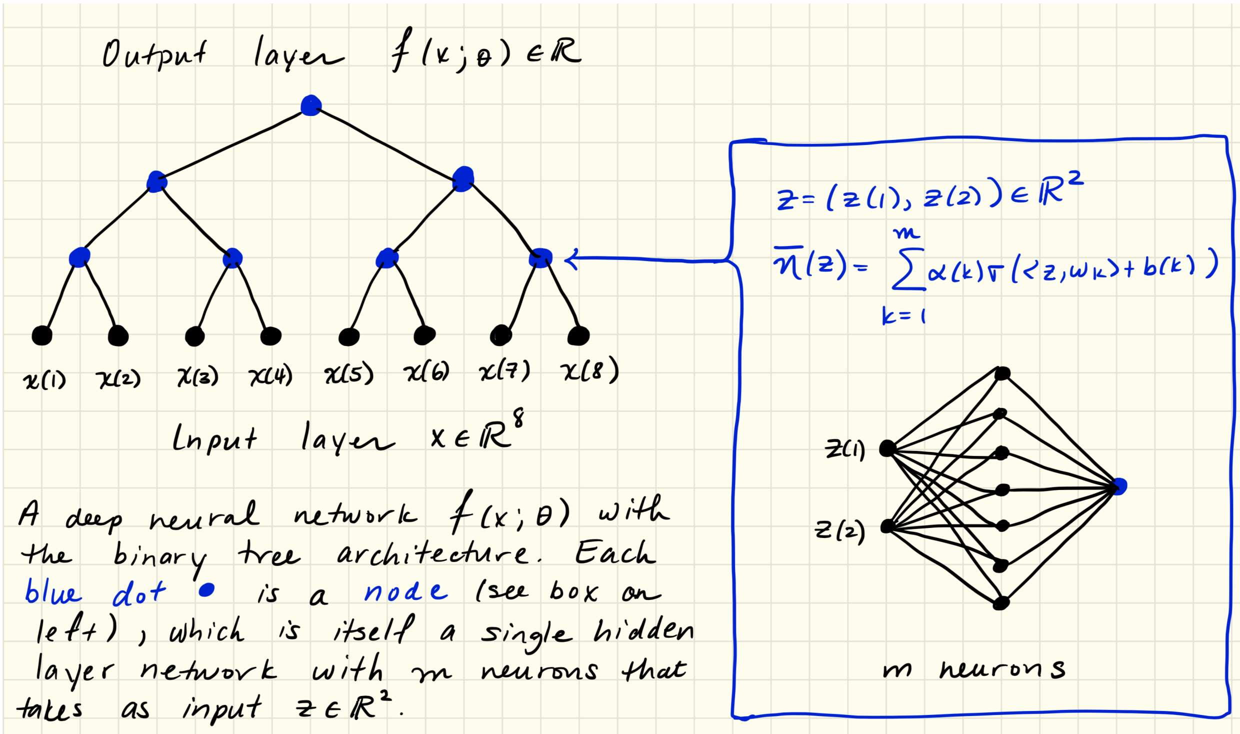

Figure 29: *A deep neural network $f \in \mathcal{D}_{N,2}$ with the binary tree structure on its nodes $\overline{\eta}(z)$, each of which takes an input $z \in \mathbb{R}^2$ and outputs a scalar, after passing z through m neurons.*

Since each node $\overline{\eta} \in \mathcal{M}_{m,2}(\sigma)$ it has $(2+2)m = 4m$ trainiable parameters. With $d-1$ such nodes in f, the number of trainable parameters is:

$$(d-1)4m = 4\frac{N}{m}m = 4N\,.$$

$\square$

Therefore networks in $\mathcal{M}_{N,d}(\sigma)$ have $(d+2)N$ trainable parameters and networks in $\mathcal{D}_{N,2}(\sigma)$ have $4N$ trainable parameters. If the dimension d is fixed, then both styles of networks have $O(N)$ trainable parameters.

Now let us compare the performance of shallow networks from $\mathcal{M}_{N,d}(\sigma)$ to deep networks from $\mathcal{D}_{N,2}(\sigma)$. The main point that these results will emphasize is the following. The space $\mathbf{C}_2^s[-1,1]^d \subseteq \mathbf{C}^s[-1,1]^d$, which means one layer neural networks can approximate any function $F \in \mathbf{C}_2^s[-1,1]^d$. However, they will not take advantage of the compositional structure of F. On the other hand, a network $f \in \mathcal{D}_{N,2}(\sigma)$ with a binary tree structure on its nodes that matches (or contains as a subgraph) the compositional binary tree structure of F can take advantage of the structure of F and circumvent the curse of dimensionality. Let us now describe the results in more detail.

In [19] the following one-layer theorem is proved, which is simliar to Theorem 8.11, but gives the rate of convergence for a large class of activation functions.

Theorem 9.6 (Mhaskar 1996, [19]). *Let $\sigma \in \mathbf{C}^\infty(\mathbb{R})$ not be a polynomial. Then for any $F \in \mathbf{C}^s[-1,1]^d$ with $\|F\|_{\mathbf{C}^s[-1,1]^d} \leq 1$,*

$$\inf_{f \in \mathcal{M}_N(\sigma)} \|F - f\|_{\mathbf{L}^\infty[-1,1]^d} \leq C N^{-s/d}\,.$$

Stated another way, in order guarantee

$$\inf_{f \in \mathcal{M}_N(\sigma)} \|F - f\|_{\mathbf{L}^\infty[-1,1]^d} \leq \epsilon$$

for an arbitrary $F \in \mathbf{C}^s[-1,1]^d$ with $\|F\|_{\mathbf{C}^s[-1,1]^d} \leq 1$, one must take

$$N = O(\epsilon^{-d/s})$$

neurons in the one hidden layer of f.

Note that since Theorem 9.6 applies to $\mathbf{C}^s[-1,1]^d$ it also applies to $\mathbf{C}_2^s[-1,1]^d$, and as examples such as the one described in Section 9.1 show, the result cannot be improved. Now let us restrict attention to $\mathbf{C}_2^s[-1,1]^d$ and consider the class $\mathcal{D}_{N,2}(\sigma)$ of deep networks with binary tree nodal structure.

Theorem 9.7 (Poggio, et al., [18]). *Let $\sigma \in \mathbf{C}^\infty(\mathbb{R})$ not be a polynomial. Let $F \in \mathbf{C}_2^s[-1,1]^d$ and let $\{H_\lambda \in \mathbf{C}^s[-1,1]^2\}_\lambda$ be the constituent functions of F, each satisfying $\|H_\lambda\|_{\mathbf{C}^s[-1,1]^2} \leq 1$. Then,*

$$\inf_{f \in \mathcal{D}_{N,2}(\sigma)} \|F - f\|_{\mathbf{L}^\infty[-1,1]^d} \leq C(d,s) N^{-s/2}\,.$$

Stated another way, in order to guarantee

$$\inf_{f \in \mathcal{D}_{N,2}(\sigma)} \|F - f\|_{\mathbf{L}^\infty[-1,1]^d} \leq \epsilon$$

for an arbitrary $F \in \mathbf{C}_2^s[-1,1]^d$ with $\|H_\lambda\|_{\mathbf{C}^s[-1,1]^2} \leq 1$, one must take

$$N = C'(d,s)\epsilon^{-2/s}\,.$$

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.