[DRAFT] Depth Separation

Alon Netser, Eran Malach

June 2, 2021

While deep neural-networks can **express** any function that can be run efficiently on a computer, in the general case, **learning** neural-networks is computationally hard. However, in practice, deep neural networks are successfully trained on real world datasets. In recent years there are many positive results on learning neural-networks, but these works usually show learnability of models that can be learned using linear methods. This is quite disappointing, as neural-networks seem much stronger than linear models.

A work by Daniely and Malach [2] shows learnability of k-Parities using neural-networks. Formally, they show that this family of distributions can be learned efficiently using a 1 hidden layer neural-network, and it cannot be approximated by a linear classifier on top of a fixed embedding of the input space in $\mathbb{R}^N$ (unless N or the norm of the linear separator is exponentially large). Note that classical results from statistical queries literature implies that no gradient-descent algorithm can learn parities over the uniform distribution, so the distribution they take is different - a mixture between a uniform distribution and a distribution where all the bits in the parity are equal.

We plan to do something similar (in spirit) - show a distribution family which can not be approximated by a shallow neural-network (i.e. 1 hidden layer), but can be learned efficiently using a deeper neural-network (i.e. 2 hidden layers). The motivation is that deeper networks work better than shallow ones on real world datasets, but currently there is no theoretical result showing that using deeper architecture is preferable.

A work by Malach and Shalev-Shwartz [4] shows a distribution family of tree-structured boolean circuits, that is learnable using a neural-network of depth $\log(n)$ (as long as the distribution satisfies the "local correlation assumption"). In addition, they show that this distribution family can not be expressed by a 1 hidden layer neural network, unless the number of hidden neurons is exponentially large. This could be thought of as a depth separation result, but we plan to show something stronger - while [4] presents a learnable distribution family that can not be **expressed** by a shallow network, we want to show a learnable distribution that can not even be **approximated** by a shallow network. Indeed, the goal of a learner is to approximate the target function and not to compute it exactly, so showing hardness of exact expressivity seems irrelevant from a machine learning perspective.

The function we want to investigate is the following. Consider the input space $\mathcal{X} \times \mathcal{Z} := \{\pm 1\}^n \times \{\pm 1\}^n$, so each input is of the form $(\mathbf{x}, \mathbf{z}) = (x_1, \ldots, x_n, z_1, \ldots, z_n)$. Consider the function class $\mathcal{H}$ where each function is of the form $h_g(\mathbf{x}, \mathbf{z}) = \prod_{i=1}^n (g_i(x_i, z_i))$ for some choice of boolean operators $g \in \{\vee, \wedge\}^n$. A work by Malach and Shalev-Shwartz [5] shows that for $g = (\vee, \vee, \ldots, \vee)$ the function $h_g(\mathbf{x}, \mathbf{z}) = \prod_{i=1}^n (x_i \vee z_i)$ is hard to approximate using 1

hidden layer neural-network, under the uniform distribution.

Note that any function in $\mathcal{H}$ can be expressed by a 2 hidden layers neural-network - the first layer is locally connected and performs element-wise OR/AND between each adjacent x_i, z_i, and the rest of the network will implement the parity function (it is known that 1 hidden layer neural-network can implement any parity function). What's left is to come up with some distribution family over the inputs which enables efficient learning. For simpler analysis, as was done in [3] and [4], we want to examine the performance of layer-wise gradient descent algorithm, a method which was recently shown to have competitive results with regular gradient descent, scaling up to the ImageNet dataset [1].

Note that the result in [5] shows that this function class can not be approximated using 1 hidden layer networks under the uniform distribution, since it contains the function $\prod_{i=1}^{n}(x_i \vee z_i)$ for which they show the hardness result. However, in order to learn $\mathcal{H}$ using a deeper network we must change the distribution, since classical results from statistical queries implies that learning parities is impossible using a gradient based algorithm under the uniform distribution (even for deeper networks that can express the target function). In order to benefit from the hardness result under the uniform distribution from [5], we want to define a distribution which is a mixture of the uniform distribution and another one. Formally, we will define $\mathcal{D} = \frac{1}{2} \cdot \mathcal{D}_{easy} + \frac{1}{2} \cdot \mathcal{D}_{hard}$ where $\mathcal{D}_{hard}$ is the uniform distribution (under which $\mathcal{H}$ is hard to approximate using a shallow network) and $\mathcal{D}_{easy}$ will be some other distribution which enables learning using a deeper network. Since the approximation error of $\mathcal{H}$ under $\mathcal{D}_{hard}$ is $\Omega(1)$, its approximation error under $\mathcal{D}$ will be at least its error on $\mathcal{D}_{hard}$ i.e. $\frac{1}{2} \cdot \Omega(1) = \Omega(1)$. Note that this is the same idea as was done in [2], where $\mathcal{D}_{easy}$ is a distribution that is uniform on all the bits outside the parity, but all the bits inside the parity are equal. In essence, this distribution "reveals" the target function directly from the data points, without the need to observe the labels.

References

[1] Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon. Greedy layerwise learning can scale to imagenet. In *International conference on machine learning*, pages 583–593. PMLR, 2019.

[2] Amit Daniely and Eran Malach. Learning parities with neural networks. *Advances in Neural Information Processing Systems*, 33, 2020.

[3] Eran Malach and Shai Shalev-Shwartz. A provably correct algorithm for deep learning that actually works. *arXiv preprint arXiv:1803.09522*, 2018.

[4] Eran Malach and Shai Shalev-Shwartz. Learning boolean circuits with neural networks. *arXiv preprint arXiv:1910.11923*, 2019.

[5] Eran Malach and Shai Shalev-Shwartz. When hardness of approximation meets hardness of learning. *arXiv preprint arXiv:2008.08059*, 2020.