Is Deeper Better only when Shallow is Good?

Eran Malach and Shai Shalev-Shwartz

Mobileye and The Hebrew University of Jerusalem

Simons Institute, Berkeley, 2019

https://joelgrus.com/2016/05/23/fizz-buzz-in-tensorflow/

interviewer: OK, so I need you to print the numbers from 1 to 100, except that if the number is divisible by 3 print "fizz", if it's divisible by 5 print "buzz", and if it's divisible by 15 print "fizzbuzz".
Do you need help getting started?

me: No, no, I'm good. So let's start with some standard imports:

```
import numpy as np
import tensorflow as tf
```

https://joelgrus.com/2016/05/23/fizz-buzz-in-tensorflow/

interviewer: OK, so I need you to print the numbers from 1 to 100, except that if the number is divisible by 3 print "fizz", if it's divisible by 5 print "buzz", and if it's divisible by 15 print "fizzbuzz".
Do you need help getting started?

me: No, no, I'm good. So let's start with some standard imports:

```
import numpy as np
import tensorflow as tf
```

Postscript: I didn't get the job. So I tried actually running this, and it turned out **it got some of the outputs wrong! Thanks a lot, machine learning!**

https://joelgrus.com/2016/05/23/fizz-buzz-in-tensorflow/

interviewer: OK, so I need you to print the numbers from 1 to 100, except that if the number is divisible by 3 print "fizz", if it's divisible by 5 print "buzz", and if it's divisible by 15 print "fizzbuzz".
Do you need help getting started?

me: No, no, I'm good. So let's start with some standard imports:

```
import numpy as np
import tensorflow as tf
```

Postscript: I didn't get the job. So I tried actually running this, and it turned out **it got some of the outputs wrong! Thanks a lot, machine learning!**

I guess maybe I should have used a deeper network ...

Depth Efficiency

- Basic question: on which distributions deeper networks are much better than shallow ones?

Depth Efficiency

- Basic question: on which distributions deeper networks are much better than shallow ones?
- Several recent results show

Depth Separation

There exist functions which can be expressed by a small deep network but must have an exponential width in order to be expressed by a shallow network

E.g. Telgarsky 2015, Safran and Shamir 2016, Cohen et al 2016, Daniely 2017, Poggio et al 2017

Outline

Main Claim

Strong depth separation $\Rightarrow$ Gradient based Algorithms fail

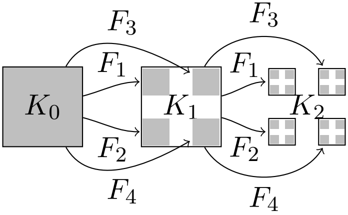
1. Case study: Fractal Distributions
2. Depth Separation
3. Approximation Curve and Strong Depth Separation
4. Success of SGD depends on the Approximation Curve

Fractals

- Iterated Function System:

$$K_0 = [-1,1]^d$$
$$K_n = F_1(K_{n-1}) \cup \ldots \cup F_r(K_{n-1})$$

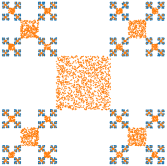
- We assume F_i are affine, invertible, contractive, and for $i \neq j$, the images of F_i and F_j are disjoint.
- The "depth" of the fractal is n
- Example: $F_i(x) = c_i + \frac{1}{4}(x - c_i)$ for $c_i \in \{\pm 1\}^2$



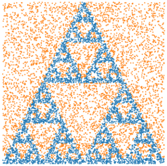
Fractal Distributions

- A "fractal distribution" is a distribution in which positive examples are sampled from the set K_n and negative examples are sampled from its complement
- Examples:

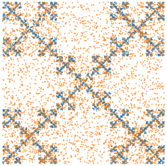
Cantor



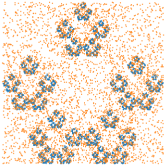
Sierpinsky



Vicsek



Pentaflake



Depth Separation

Theorem

Consider an IFS over $[-1,1]^d$ with r generating functions and depth n. For any fractal distribution D_n there exists a ReLU feed forward network of depth $2n+1$ and width $5dr$ which realizes D_n.

Depth Separation

Theorem

Consider an IFS over $[-1,1]^d$ with r generating functions and depth n. For any fractal distribution D_n there exists a ReLU feed forward network of depth $2n+1$ and width $5dr$ which realizes D_n.

Proof by induction:

- Basis: a shallow ReLU network can approximate $I_0(x) = 1_{x \in K_0}$
- Suppose we have a deep network expressing: $I_{n-1}(x) = 1_{x \in K_{n-1}}$
- Recall: $K_n = F_1(K_{n-1}) \cup \ldots \cup F_r(K_{n-1})$ and F_i are affine, invertible, and have disjoint images
- Take $x \in K_n$, then there's $z \in K_{n-1}$ and i s.t. $x = F_i(z)$, or equivalently, $z = F_i^{-1}(x)$
- Therefore, $\left[\sum_i I_{n-1}(F_i^{-1}(x))\right]_+ - \left[\sum_i I_{n-1}(F_i^{-1}(x)) - 1\right]_+ = 1_{x \in K_n}$

Depth Separation

Theorem

If D_n has non-zero probability in any area of K_n, then a network of depth t must have a width of at least $\frac{d}{e} r^{\frac{n}{td}}$ to realize D_n.

Depth Separation

Theorem

If D_n has non-zero probability in any area of K_n, then a network of depth t must have a width of at least $\frac{d}{e} r^{\frac{n}{td}}$ to realize D_n.

Proof idea:

- A network of width k and depth t has at most $(ek/d)^{td}$ linear regions
- To realize the fractal distribution, we need r^n linear regions

Outline

Main Claim

Strong depth separation $\Rightarrow$ Gradient based Algorithms fail

1. Case study: Fractal Distributions
2. Depth Separation
3. Approximation Curve and Strong Depth Separation
4. Success of SGD depends on the Approximation Curve

Approximation Curve

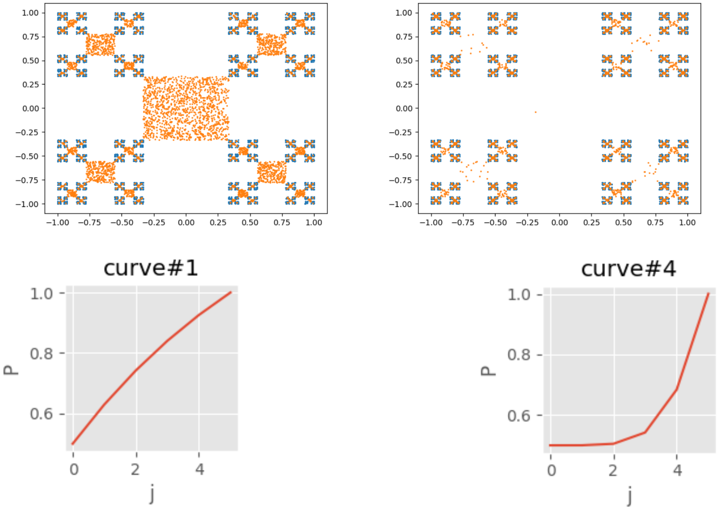
- We saw: a network of depth $O(n)$ can express a depth n fractal, but a shallower network requires exponential width to fully realizes the distribution
- Approximation curve: How much of the negative examples are on the fine details of the fractal:

$$P(j) := 1 - L_{D_n}(1_{x \in K_j}) := 1 - \mathbb{P}_{(x,y) \sim D_n}\left[x \in K_j \ \wedge \ y = -1\right]$$

- Note: $P(0) = 1/2$, $P(n) = 1$, and P is monotonically increasing

Approximation Curve: coarse vs. fine

$$P(j) = 1 - L_{D_n}(1_{x \in K_j})$$



Approximation Curve and Strong Depth Separation

The following theorem shows that with reasonable width, the error of a depth $\Theta(j)$ network is roughly $1 - P(j)$

Theorem

Fix a depth n distribution with approximation curve P. Then, for every j

1. *For a depth $t = 2j + 2$ and width $k = 5dr$ network we have*

$$L_{D_n}(H_{t,k}) \ \leq \ (1 - P(j))$$

2. *For every s, if $k < r^s$ and $t < j/s$ then*

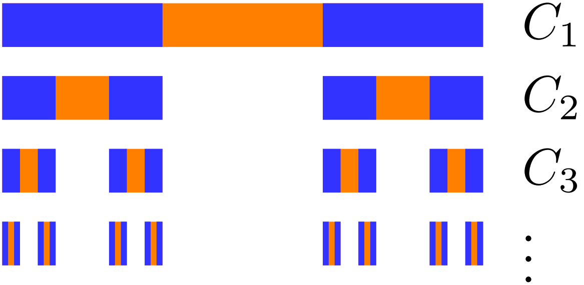
$$(1 - r^{st-j})(1 - P(j)) \ \leq \ L_{D_n}(H_{t,k})$$

Outline

Main Claim

Strong depth separation $\Rightarrow$ Gradient based Algorithms fail

One dimensional Cantor Fractal with "Fine" Distribution



- $C_0 = [0, 1]$ and $C_n = F_1(C_{n-1}) \cup F_2(C_{n-1})$, where $F_1(x) = \frac{1}{3} - \frac{1}{3}x$ and $F_2(x) = \frac{2}{3} + \frac{1}{3}x$
- "Fine" cantor distributions of growing depth. Negative areas in orange, positive in blue.

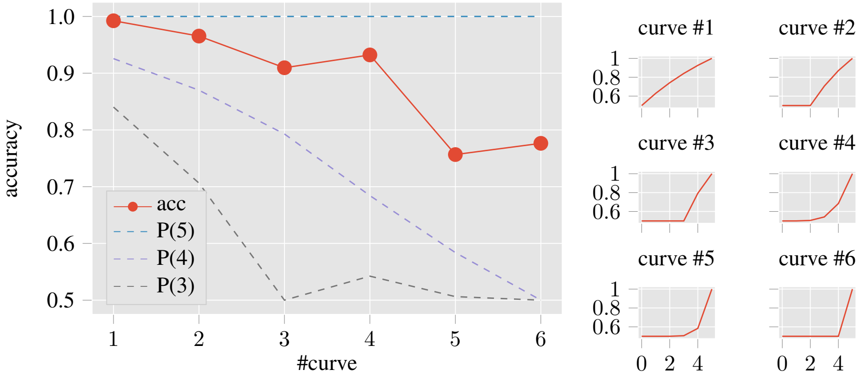
Gradient at Initialization and the Approximation Curve

Theorem

Consider a depth t, width k, network, and suppose the weights, W, are initialized randomly in the "normal" way. Consider a depth n, one-dimensional Cantor fractal, and let $j = \lceil \log(tk^2/\delta) \rceil$. Then, with probability $> 1 - \delta$, all elements of the gradient at W are of magnitude $< 5(P(j) - \frac{1}{2})$.

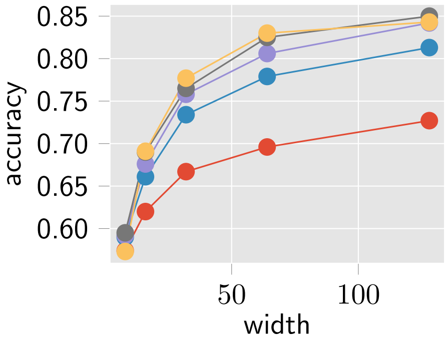
- Corollary: gradient descent is likely to fail on every cantor distribution with strong depth separation, even though the deep network is expressive enough

Success of SGD depends on the Approximation Curve



Learning depth 5 network on 2D cantor set of depth 5, with different approximation curves.

Is Deep Good only When Shallow is Also Good ?



- The effect of depth on learning CIFAR-10.
- We train CNNs with Adam for 60K steps. All layers are 5x5 Convolutions with ReLU activation, except the readout layer
- Line colors correspond to different network depth

Summary

- Fractal distributions are natural for studying depth efficiency of deep learning
- The "approximation curve" is correlated with how much going deeper really helps
- Strong depth separation: shallow networks perform like random guess while deeper networks realize the distribution
- Conjecture: gradient based algorithms fail when there is strong depth separation. In other words,
  deep is better only when shallow is also good

A more concrete formalism of the conjecture

Conjecture:

- Let $\mathcal{H}$ be all functions which cannot be approximated by a shallow network. Then:
  1. For each $f \in \mathcal{H}$ there exists a distribution D_f on $\mathcal{X} \times \{\pm 1\}$ for which f achieves zero loss while the best shallow network achieves a loss $> 1/2 - \epsilon$.
  2. For every such D_f, gradient-descent fails to learn a deep network.