

Mathematical
Institute

Approximation Theory for Deep Learning: Exponential expresivity with depth.

Theories of Deep Learning: C6.5,
Lecture / Video 3

Prof. Jared Tanner
Mathematical Institute
University of Oxford

Oxford
Mathematics

Classification of inputs $x \in \mathbb{R}^n$ to c classes denoted by $\{e_i\}_{i=1}^c$, is modelled by a function $H(x)$ for which $H(x) = e_i$ for all x in class i where $e_i(\ell) = 1$ for $i = \ell$ and 0 otherwise.
Approximation Theory concerns the ability to approximate functions from a given representation; see Approximation of Function (C6.3).

Some of the most well studied examples include approximation of a function $f(x)$ over $x \in [-1, 1]$ with some smoothness, say three times differentiable, by polynomials of degree at most k or trigonometric exponentials.

Here our focus is on the ability to approximate functions $H(x; \theta)$ given by a deep network architecture; for $x \in \mathbb{R}^n$.

What functions can a DNN approximate arbitrarily well? What is the advantage of depth?

- Network architectures are able to approximate any function (Cybenko (89') and Hornik (90')).
- There are functions which DNNs are able to construct with polynomially many parameters, that require exponentially many parameters for a shallow network to represent. (Telgarsky 15').
- Deep networks can approximate nonlinear functions on compact sets to ϵ uniform accuracy with depth and width scaling like $\log(1/\epsilon)$. (Yarotsky 16')

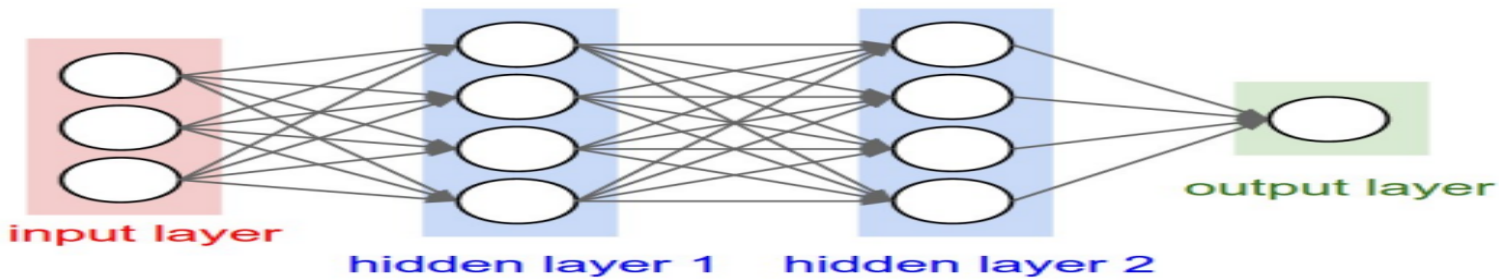
Example of a fully connected DNN:

Two layer fully connected neural net

Repeated affine transformation followed by a nonlinear action:

$$h_{i+1} = \phi_i \left(W^{(i)} h_i + b^{(i)} \right) \qquad \text{for} \quad i = 1, \ldots, N-1$$

where $W^{(i)} \in \mathbb{R}^{n_{i+1} \times n_i}$ and $b^{(i)} \in \mathbb{R}^{n_{i+1}}$ and $\phi(\cdot)$ is a nonlinear activation such as ReLU, $\phi(z) := max(0, z) = z_+$.
The input is h_1, the output is h_N, and h_i for intermediate $i = 2, \cdot, N-1$ are referred to as "hidden" layers.
The number of layers N is the depth, $N \gg 1$ is called "deep."



https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Architecture/feedforward.html

Superposition of sigmoidal functions (Cybenko 89')

DNNs with sigmoidal activations are dense in $C_n([0,1])$

University of Oxford
Mathematical Institute

Consider the feedforward network with one hidden layer:

input $h_1 = x \in \mathbb{R}^n$
hidden layer $h_2 = \phi\left(W^{(1)} h_1 + b^{(1)}\right) \in \mathbb{R}^m$
output $H(x,\theta) = \alpha^T h_2 = \sum_{i=1}^m \alpha_i \phi(w_i^T x + b_i)$
with $\phi(t) \in [0,1]$, say $\phi(t) = 1/(1+e^{-t})$.

Theorem (Cybenbko 89')

Let $\phi(t)$ be a continuous monotone function with $\lim_{t\to-\infty} \phi(t) = 0$ and $\lim_{t\to\infty} \phi(t) = 1$, then the set of functions of the form $H(x;\theta) = \sum_{i=1}^m \alpha_i \phi(w_i^T x + b_i)$ is dense in $C_n([0,1])$.

That is, one (or more) layer fully connected nets are sufficient to approximate any continuous function, provided m is large enough.

`https://link.springer.com/article/10.1007/BF02551274`

Approximation of multilayer feedforward nets (Hornik 90')

DNNs with continuous bounded activations are dense in $C_n([0,1])$

University of Oxford
Mathematical Institute

Consider the feedforward network with one hidden layer:
input $h_1 = x \in \mathbb{R}^n$
hidden layer $h_2 = \phi\left(W^{(1)} h_1 + b^{(1)}\right) \in \mathbb{R}^m$
output $H(x,\theta) = \alpha^T h_2 = \sum_{i=1}^m \alpha_i \phi(w_i^T x + b_i)$
with $\phi(t) \in [0,1]$ non-constant.

Theorem (Hornik 90')

Let $\phi(t)$ be unbounded then $H(x;\theta) = \sum_{i=1}^m \alpha_i \phi(w_i^T x + b_i)$ is dense in $L^p(\mu)$ for all finite measures μ and $1 \leq p < \infty$. Moreover, if $\phi(t)$ is continuous and bounded, then $H(x;\theta) = \sum_{i=1}^m \alpha_i \phi(w_i^T x + b_i)$ is dense in $C_n([0,1])$.

Much of the result includes showing $L(\phi) = \int_{I_n} \phi(x) d\mu(x) = 0$ for $\phi(x)$ in the specified class implies $\mu(x) = 0$.

`https://www.sciencedirect.com/science/article/pii/089360809190009T`

Telegarsky (2015) considered a specific construction of a function from a deep network which requires an shallow network to have exponential width.
Let $\phi(x) = ReLU(x) = max(x, 0)$ and consider the two layer net:

$$h_2(x) = 2\phi(x) - 4\phi(x - 1/2) = \begin{cases} 0 & x < 0 \\ 2x & x \in [0, 1/2] \\ 2 - 2x & x > 1/2 \end{cases}$$

and $h_3(x) = \phi(h_2(x))$ set to zero the negative portion for $x > 1$.
`https://arxiv.org/abs/1509.08101`

Representational benefits of depth (Telgarsky 15')

Composition gives exponential growth in complexity

University of Oxford
Mathematical Institute

For $\phi(x) = \max(x, 0)$ let $f(x) = h_3(x) = \phi(2\phi(x) - 4\phi(x - 1/2))$ and iterate this 2-layer network k times to obtain a $2k$-layer network $f^k(x) = f(f(\cdots(f(x)\cdots))$ with the property that it is piecewise linear with change in slope at $x_i = i2^{-k}$ for $i = 0, 1, \ldots, 2^k$ and moreover takes on the values $f^k(x_i) = 0$ for i even and $f^k(x_i) = 1$ for i odd.

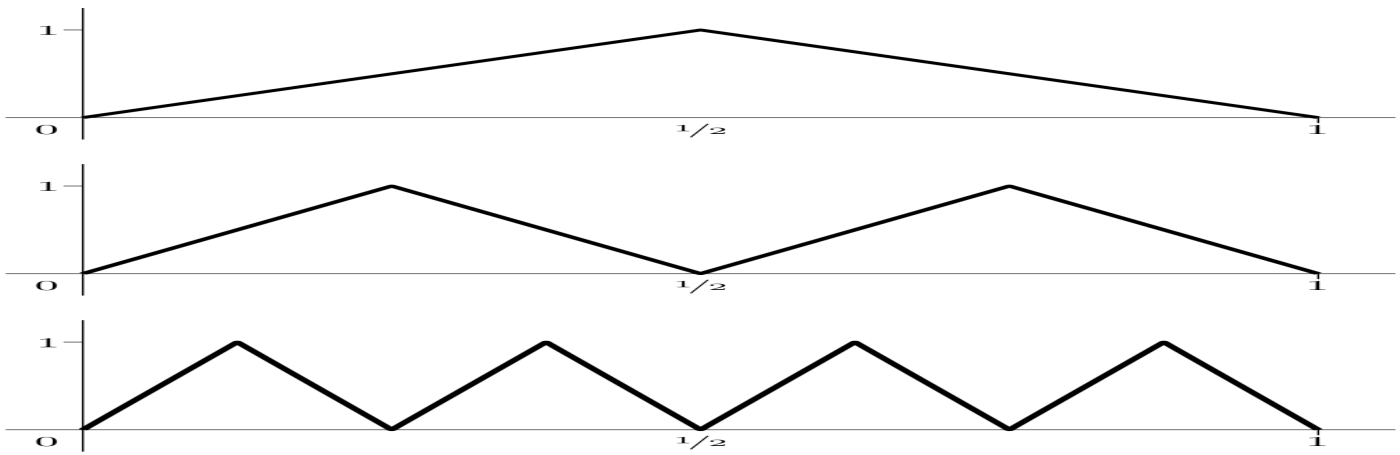


Figure 2: f_m, f_m^2, and f_m^3.

Representational benefits of depth (Telgarsky 15')

Composition gives exponential growth in complexity: width vs. depth

Mathematical Institute

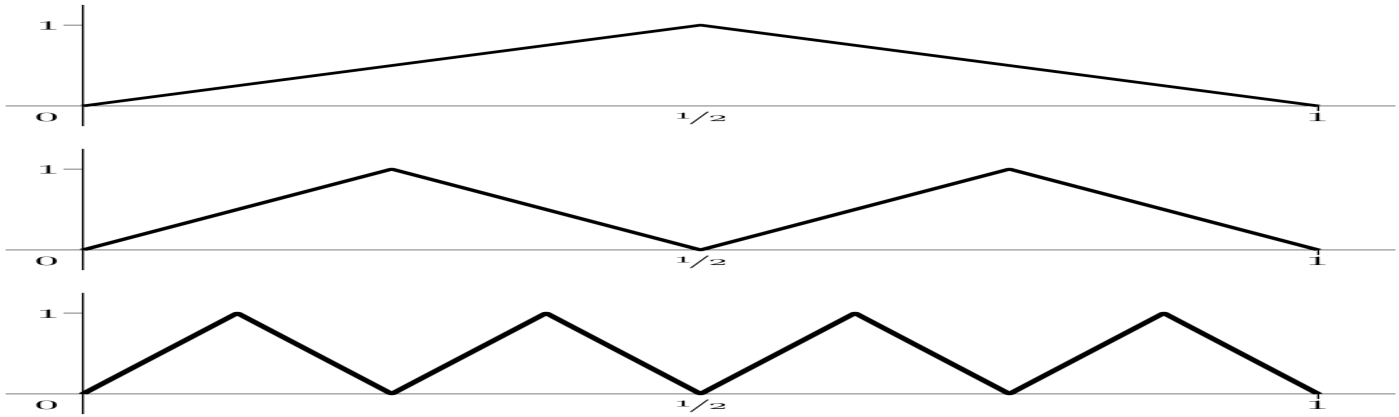


Figure 2: f_{m}, f_{m}^2, and f_{m}^3.

In contrast, a two-layer network with the same $\phi(x)$ of the form $\phi\left(\sum_{j=1}^{m} \alpha_j \phi(w_j x - b_j)\right)$ requires $m = 2^k$ to exactly express $f^k(x)$.

The deep network can be thought of as having $6k$ parameters, whereas the two-layer network requires $3 \cdot 2^k + 1$ parameters; exponentially more. `https://arxiv.org/abs/1509.08101`

Representational benefits of depth (Telgarsky 15')

Classification error rates

UNIVERSITY OF OXFORD
Mathematical Institute

Define the function class $F(\phi; m, \ell)$ be the space of functions composed of ℓ layer fully connected m width feed forward nets with nonlinear activation function ϕ. Let $\mathcal{R}(f) := n^{-1} \sum_{i=1}^{n} \chi[f(x_i) \neq y_i]$ count the number of incorrect labels of the data set $\{(x_i, y_i)\}_{i=1}^{n}$.

Theorem (Telgarsky 15')

Consider positive integers k, ℓ, m with $m \leq 2^{(k-3)/\ell - 1}$, then there exists a collection of $n = 2^k$ points $\{(x_i, y_i)\}_{i=1}^{n}$ with $x_i \in [0, 1]$ and $y_i \in \{0, 1\}$ such that

$$\min_{f \in F(\phi; 2, 2k)} \mathcal{R}(f) = 0 \quad \text{and} \quad \min_{g \in F(\phi; m, \ell)} \mathcal{R}(g) \geq \frac{1}{6}.$$

`https://arxiv.org/abs/1509.08101`

Representational benefits of depth (Yarotsky 16')

ReLU nets can approximate x^2 exponentially well

UNIVERSITY OF OXFORD
Mathematical Institute

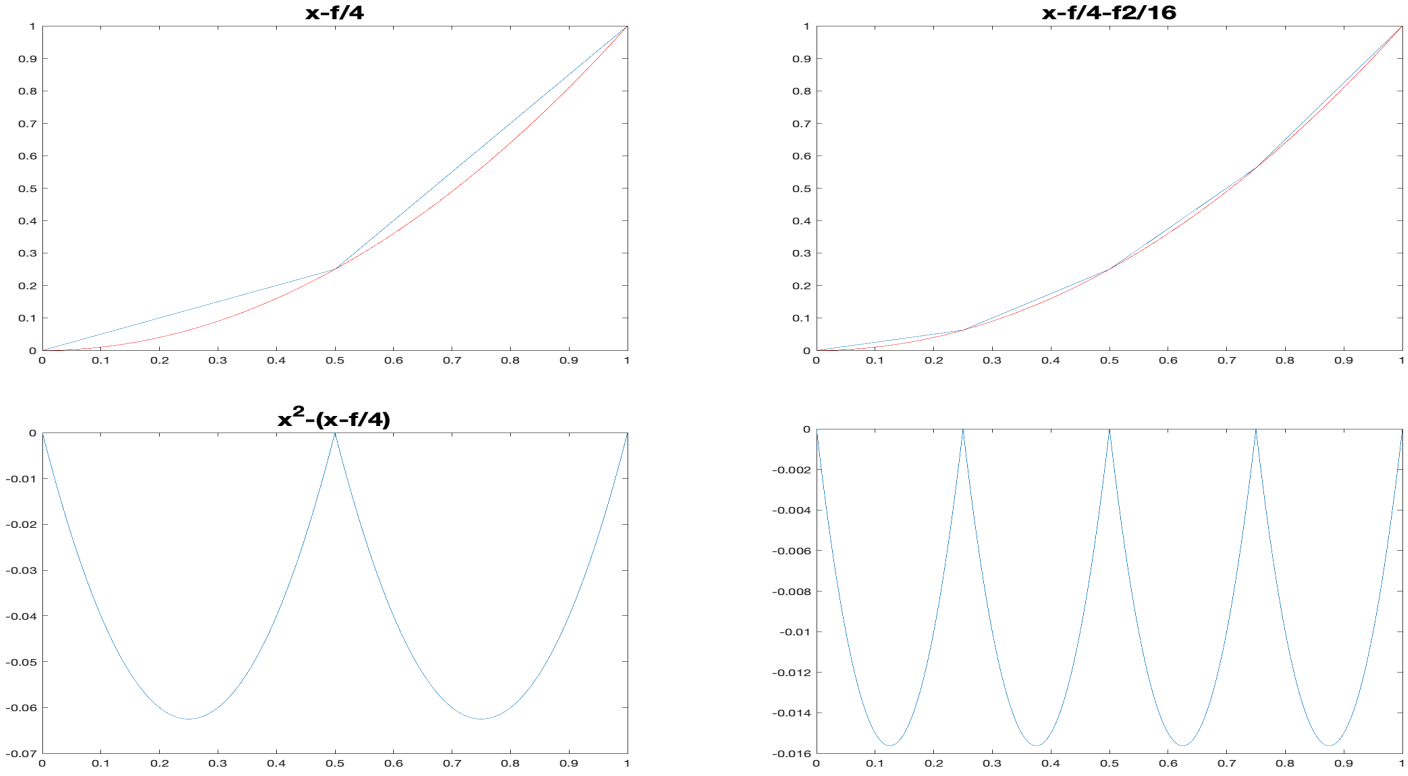
Returning to the saw-tooth function composted of $\phi(x) = \max(x, 0)$ let $f(x) = h_3(x) = \phi(2\phi(x) - 4\phi(x - 1/2))$ and iterate this 2-layer network m times to obtain a $2m$-layer network $f^m(x) = f(f(\cdots(f(x)\cdots))$ with $6m$ weights.
Let $h_m(x)$ denote the piecewise linear interpolation of $h(x) = x^2$ at 2^{m+1} equispaced points, then

$$h_m(x) = x - \sum_{s=1}^{m} 2^{-2s} f^s(x)$$

and $\max_{x\in[0,1]}|x^2 - h_m(x)| = 2^{-2(m+1)}$. Consequently, x^2 can be approximated on $[0, 1]$ to uniform accuracy ϵ by a ReLU network having depth $\frac{1}{2}\log_2(1/\epsilon)$ and 6 weights per pair of layers.
`https://arxiv.org/pdf/1610.01145.pdf`

Representational benefits of depth (Yarotsky 16')

ReLU nets can approximate x^2 exponentially well: plots 1



Yarotsky (16') approximation of x^2 with ReLU DNN.
`https://arxiv.org/pdf/1610.01145.pdf`

Representational benefits of depth (Yarotsky 16')

ReLU nets can approximate x^2 exponentially well: plots 2

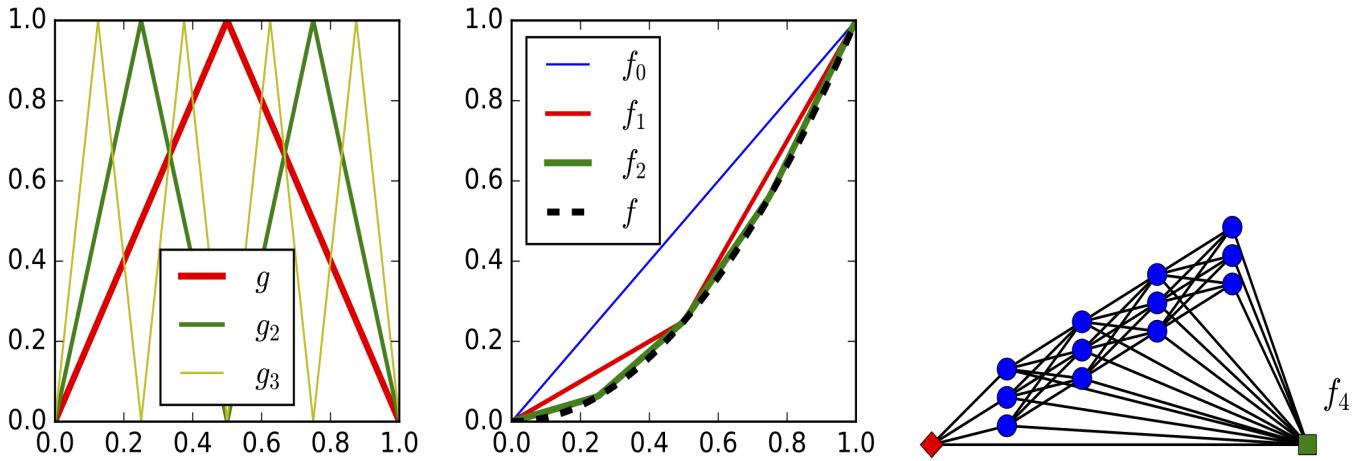


Figure 2: Fast approximation of the function $f(x) = x^2$ from Proposition 2: (a) the "tooth" function g and the iterated "sawtooth" functions g_2, g_3; (b) the approximating functions f_m; (c) the network architecture for f_4.

Telgarsky (15') and Yarotsky (16') follow from exponential nature of composition of the same function, self similarity.

`https://arxiv.org/pdf/1610.01145.pdf`

Representational benefits of depth (Yarotsky 16')

ReLU nets can approximate x^2 exponentially well: plots

University of Oxford
Mathematical Institute

High order approximation can be shown by extending that a DNN with depth and number of weights proportional to $\ln(1/\epsilon)$ can approximate any quadratic function within ϵ to polynomials of arbitrary degree. This follows by noting the relationship

$$xy = \frac{1}{2}\left((x+y)^2 - x^2 - y^2\right)$$

which demonstrates that the ability to square a number allows general multiplication. For example, letting $H(x;\theta)$ denote a network approximating x^2, then the above relation can be applied to compute $x^3 = xH(x)$ by letting $y = H(x)$. Similarly polynomials of arbitrary degree can be approximated within ϵ by a DNN with depth and number of weights proportional to $\ln(1/\epsilon)$.
`https://arxiv.org/pdf/1610.01145.pdf`

Representational benefits of depth (Yarotsky 16')

ReLU nets can approximate Sobolev spaces

University of Oxford
Mathematical Institute

The Sobolev norm is similar to that of functions with $n-1$ derivatives that are Lipschitz continuous $C^{n-1}([0,1]^d)$ excluding sets of measure zero.

$$\|f\|_{W^{n,\infty}}([0,1]^d) = \max_{|s|\leq n} \text{esssupp}_{x\in[0,1]^d} |D^s f(x)|.$$

Define the unit ball of functions in $W^{n,\infty}([0,1]^d)$ as

$$F_{n,d} = \left\{ f \in W^{n,\infty}([0,1]^d) : \|f\|_{W^{n,\infty}}([0,1]^d) \leq 1 \right\}.$$

Theorem (Yarotsky 16')

For any d, n and $\epsilon \in (0,1)$, there is a ReLU network with depth at most $c(1+\ln(1/\epsilon))$ and at most $c\epsilon^{-d/n}(1+\log(1/\epsilon))$ weights (width $\mathcal{O}(\epsilon^{-d/n})$), for c a function of d, n, that can approximate any function from $F_{d,n}$ within absolute error ϵ.

UNIVERSITY OF OXFORD
Mathematical Institute

Localize an arbitrary function in $\mathbb{R}^d$ into $(N+1)^d$ local continuous regions using local (compactly supported) functions $\phi_m(x)$ which sum to 1. E.g. let

$$\text{with } \psi(x) = \begin{cases} 1 & |x| < 1 \\ 2 - |x| & 1 \leq |x| \leq 2 \\ 0 & |x| > 2 \end{cases}$$

and note that $\sum_{m=0}^{N} \psi\left(3N(x_k - m/N)\right) = 1$ for $x_k \in [0,1]$. Multiplying $f(\cdot)$ by each shift $\psi\left(3N(x_k - m/N)\right)$ for $m = 0, \cdots, N$ localizes the x_k variable over and can be done via a one-dimensional convolutional layer with one filter that doesn't require trainable parameters. This can then be repeated over d times to localize each of the d variables into $(N+1)^d$ partitions.

UNIVERSITY OF OXFORD
Mathematical Institute

Taylor series of $f(\cdot)$ **about** $\{(x_k - m/N)\}_{m=0}^{N}$ **to degree** n **in each dimension** x_k $k = 1, \ldots, d$ **is**

$$P_{k,n}(f)(x) := \sum_{s=0}^{n} \frac{\partial^s f(x)}{s! \partial x_k} (x_k - m/N)^s$$

and the composite over all dimensions is $P_n(f)(x) := \Pi_{k=1}^{d} P_{k,n}(f)(x)$.

The resulting error approximating $f(x)$ **about** $\{(x_k - m/N)\}_{m=0}^{N}$ **is bounded by at most** 2^d **local terms (as any location** x **interacts with at most 2 local dilated** $\phi(3N(x_k - m/N)$ **with each term bounded using the standard Taylor series truncation bound**

$$\frac{d^n}{n! N^n} \mathbf{max}_{|s| \leq n} \mathbf{esssupp}_{x \in [0,1]^d} |D^s f(x)|.$$

Representational benefits of depth (Yarotsky 16')

Sketch of the proof 3 of 3: combining terms

Mathematical Institute

Treating $\|f\|_{W^{n,\infty}([0,1]^d)} := \max_{|s|\leq n} \text{esssupp}_{x\in[0,1]^d} |D^s f(x)|$ as bounded independent of n (not really true) gives a total bound on the local error of $2^d d^n / N^n n!$ which is bounded by ϵ if $N \geq \left(n!\epsilon/2^d d^n\right)^{-1/n}$.

It then remain to construct a network that can approximate the local Taylor series with the claimed width and depth. The partition has $d^n(N+1)^d$ terms of the form $\phi_m(x)(x-m/N)^n$, each of which can be approximated efficiently using the aforementioned ReLU networks using order $\log(2^d d^n/\epsilon)$ depth for a total of $d^n(N+1)^d \log(2^d d^n/\epsilon)$ weights.

Recalling the number of partitions $N \geq \left(n!\epsilon/2^d d^n\right)^{-1/n}$ and Stirling's Inequality that $n! \sim (n/e)^n\sqrt{2\pi n}$, gives the claimed depth and width.

Representational benefits of depth (Yarotsky 16')

Near optimality

UNIVERSITY OF OXFORD
Mathematical Institute

- Yarotsky's result shows a neural network with ReLU activation can approximate any n-smooth function in $d-$dimensions using at most order $\epsilon^{-d/n}(1+\log(1/\epsilon))$ trainable parameters.
  `https://arxiv.org/pdf/1610.01145.pdf`
- DeVore et al. proved the minimal number of trainable parameters for any method is of order $\epsilon^{-d/n}$,
  `https://link.springer.com/article/10.1007/BF01171759`
- Boulle et al. consider nonlinear activations that are rational functions of the form of a cubic over a quadratic, giving $\epsilon^{-d/n}(1+\log\log((1/\epsilon)))$ parameters.
  `https://arxiv.org/abs/2004.01902`
- Goverse et al. used two nonlinear activations (e.g. ReLU and x^2) to achieve optimal order $\epsilon^{-d/n}$
  `https://arxiv.org/abs/2301.13091`

Optimal function approximation ability of deep networks

DNNs can achieve optimal rates for function classes

UNIVERSITY OF OXFORD
Mathematical Institute

There is a growing literature on the ability to express high dimensional data using deep networks, to name a few:

- Approximation space for univariate functions; Daubechies, DeVore, Foucart, Hanin, and Petrova (19')
  `https://arxiv.org/pdf/1905.02199.pdf`
- That neural networks achieve the same approximation rate as methods such as wavelets, ridgelets, curvelets, shearlets, $\alpha-$molecules; Bölcskei, Grohs, Kutyniok, and Petersen (18')

  `https://www.mins.ee.ethz.ch/pubs/files/deep-approx-18.pdf`

The exponential complexity generated by depth allows these remarkable approximation rates. Note however, one needs to be able to train the network parameters to achieve these rates.

Overcoming the curse of dimensionality with additional structure.

Curse of dimensionality (Yarotsky 16')

Exponential in d/m growth in number of weights.

UNIVERSITY OF OXFORD
Mathematical Institute

Yarotsky 16' results show exponential approximation in depth, but the overall number of weights is $\mathcal{O}(\epsilon^{-d/m})$. Recall

$$\|f\|_{W_m^\infty([0,1]^d)} = \max_{|s|\leq m} \text{esssupp}_{x\in[0,1]^d} |D^s f(x)|.$$

Theorem (Yarotsky 16')

For any d, m and $\epsilon \in (0,1)$, there is a ReLU network with depth at most $c(1+\ln(1/\epsilon))$ and at most $c\epsilon^{-d/m}(1+\log(1/\epsilon))$ weights (width $\mathcal{O}(\epsilon^{-d/m})$), for c a function of d, m, that can approximate any function from $F_{d,m}$ within absolute error ϵ.

`https://arxiv.org/pdf/1610.01145.pdf`

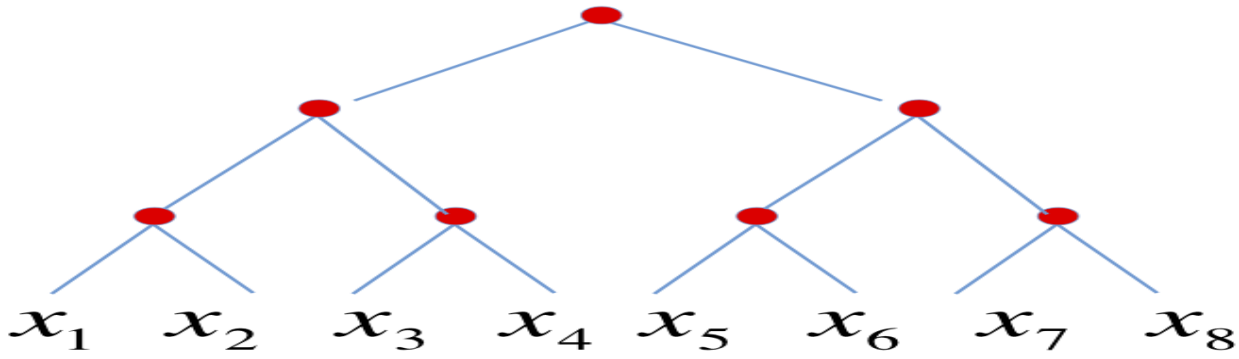
To avoid curse of dimensionality need $m \sim d$ or more structure in the function F to be approximated; e.g. compositional structure.

Compositional structured functions (Poggio et al. 17')

Extending the compositional nature of Yarotsky dimensionally

UNIVERSITY OF OXFORD
Mathematical Institute

Consider functions with a binary tree hierarchical structure:



where $x \in \mathbb{R}^8$ and
$f(x) = h_3(h_{21}(h_{11}(x_1, x_2), h_{12}(x_3, x_4)), h_{22}(h_{13}(x_5, x_6), h_{14}(x_7, x_8)))$
Let $W_m^{n,2}$ be the class of all compositional functions $f(\cdot)$ of n variables with binary tree structure and constituent functions $h(\cdot)$ of 2 variables with m bounded derivatives.
`https://arxiv.org/pdf/1611.00740.pdf`

Compositional structured functions (Poggio et al. 17')

Each constituent function is a map from $\mathbb{R}^2 \to \mathbb{R}$

UNIVERSITY OF OXFORD
Mathematical Institute

The set $W_m^{n,2}$ of of all compositional functions $f(\cdot)$ of n variables with binary tree structure and constituent functions $h(\cdot)$ of 2 variables with m bounded derivatives can be effectively approximated using a DNN with a rate dictated by the ability to approximate functions $\mathbb{R}^2 \to \mathbb{R}$; e.g. effectively locally $d = 2$.

Theorem (Poggio 17')

Let $f(\cdot) \in W_m^{n,2}$ and consider a DNN with the same binary compositional tree structure and an activation $\sigma(\cdot)$ which is infinitely differentiable, and not a polynomial. The function $f(\cdot)$, can be approximated by ϵ with a number of weights that is $\mathcal{O}\left((n-1)\epsilon^{-2/m}\right)$.

`https://arxiv.org/pdf/1611.00740.pdf`

Compositional structured functions (Poggio et al. 17')

Compositional functions $W_m^{n,2}$ compared to shallow NNs and Yarotsky(16')

UNIVERSITY OF OXFORD
Mathematical Institute

The set $W_m^{n,2}$ of of all compositional functions $f(\cdot)$ of n variables with binary tree structure are effectively $d = 2$ in the DNN approximation requirements, but are much richer than $d = 2$.

Functions can be approximated within ϵ with a DNN from $\mathcal{O}(In(1/\epsilon))$ layers with a number of weights:

- $\mathcal{O}(\epsilon^{-d/m})$ for general locally smooth functions (Yarotsky 16'),
- $\mathcal{O}\left((n-1)\epsilon^{-2/m}\right)$ for $f(\cdot) \in W_m^{n,2}$, binary tree structure and constituent functions in $C_m[0,1]^2$.
- $\mathcal{O}(\epsilon^{-d/m})$ for shallow NNs is best possible for $f(\cdot) \in W_m^n$ which have non-binary hierarchical tree structures.

`https://arxiv.org/pdf/1611.00740.pdf`

Definition of local effective dimensionality (Poggio et al. 17')

Local dimensionality determined by approximation rate ϵ^{-d}.

UNIVERSITY OF OXFORD
Mathematical Institute

Definition (Poggio 17')

The effective dimensionality of a function class W is said to be d if for every $\epsilon > 0$, any function within W can be approximated within an accuracy ϵ by a DNN at rate ϵ^{-d}.

In the prior slide we had examples of complex compositional functions with effective dimensionality 2. These could be extended naturally to local *effective dimensionality* d_{eff} and *local smoothness* m_{eff} for rate $\epsilon^{-d_{eff}/m_{eff}}$.

Restriction to a data class decreases d_{eff} and localisation can increase the smoothness m_{eff} substantially.
`https://arxiv.org/pdf/1611.00740.pdf`

Intrinsic dimensionality of sub-manifolds (Hein et al. 05')

MNIST exemplar dataset classes are approximately under 15 dimensional

Mathematical Institute

Estimates of dimensionality within MNIST digit classes using three approaches: the reference below, and two others building on local linear embedding.

Table 7. Number of samples and estimated intrinsic dimensionality of the digits in MNIST.

1	2	3	4	5
7877	6990	7141	6824	6903
8/7/7	13/12/13	14/13/13	13/12/12	12/12/12
6	7	8	9	0
6876	7293	6825	6958	6903
11/11/11	10/10/10	14/13/13	12/11/11	12/11/11

https://icml.cc/Conferences/2005/proceedings/papers/037_Intrinsic_HeinAudibert.pdf

The hidden manifold model (Goldt et al. 19')

An alternative manifold model: one layer GAN

UNIVERSITY OF OXFORD
Mathematical Institute

A manifold model can explicitly represent the data through:

$$X = f(CF/\sqrt{d}) \in \mathbb{R}^{p,n}$$

where:

- $F \in \mathbb{R}^{d,n}$ are the d features used to represent the data
- $C \in \mathbb{R}^{p,d}$ combines the $d < n < p$ features
- $f(\cdot)$ is an entrywise locally smooth nonlinear function.

This data model is the same as a generative adversarial network (GAN) and is similar to dictionary learning and subspace clustering models where C is typically sparse.

`https://hal-cea.archives-ouvertes.fr/cea-02529246/document`

Additional approximation theory resources

Review articles and courses elsewhere

Mathematical Institute

Further references for the approximation theory perspective of deep learning include:

- Telgarsky's "Deep Learning Theory" course, lectures 1-11:
  `http://mjt.cs.illinois.edu/courses/dlt-f20/`
- Matthew Hirn's "Mathematics of Deep Learning" course: lectures 20-24.
  `https://matthewhirn.com/teaching/spring-2020-cmse-890-002/`
- DNN Approximation Theory by Elbrachter et al. (19')
  `https://www.mins.ee.ethz.ch/pubs/files/deep-it-2019.pdf`

Expressivity of deep nets prior to 2016, for reference.

Prior to the approximation rate results from Telgarsky 15' and Yarotsky 16', there were qualitative geometric results showing showing potential for exponential expressivity:

- On the number of response regions of deep feedforward networks with piecewise linear activations (Pascanu et al. 14')
  `https://arxiv.org/pdf/1312.6098.pdf`
- On the expressive power of deep neural networks (Raghu et al. 16')
  `https://arxiv.org/abs/1606.05336`
- Trajectory growth lower bounds for random sparse deep ReLU networks (Price et al. 19')
  `https://arxiv.org/abs/1911.10651`

ReLU hyperplane arrangement

Partition of the input domain $\mathbb{R}^{n_0}$: one layer

UNIVERSITY OF OXFORD
Mathematical Institute

The action of ReLU to an affine transform is a linearly increasing function orthogonal to hyperplanes; let $W \in \mathbb{R}^{n_1 \times n_0}$ then:

$$H_i := \{x \in \mathbb{R}^{n_0} : \ W_i x + b_i = 0\} \quad \forall i \in [n_1]$$

where W_i is the i^{th} row of W.
The normals to these hyperplanes partition the input dimension n_0, and if W is in general position (all subsets of rows are maximal rank), then the number of partitions is:

$$\sum_{j=0}^{n_0} \binom{n_1}{j}$$

`https://arxiv.org/pdf/1312.6098.pdf`

The number of partitions in one layer is lower bounded by

$$\sum_{j=0}^{n_0} \binom{n_1}{j} \geq n_1^{\min\{n_0, n_1/2\}}$$

and each hidden layers can further subdivide these regions:

Theorem (Pascanu et al. 14')

An L layer DNN with ReLU activation, input $\mathbb{R}^{n_0}$, and hidden layers of width $n_1, n_2, \ldots, n_L$ partitions the input space into at least

$$\Pi_{\ell=0}^{L} n_\ell^{\min\{n_0, n_\ell/2\}}$$

This shows an exponential dependence on depth L.

`https://arxiv.org/pdf/1312.6098.pdf`

ReLU hyperplane arrangement

Partition of the input domain $\mathbb{R}^{n_0}$: plot Pascanu et al. 14'

Mathematical Institute

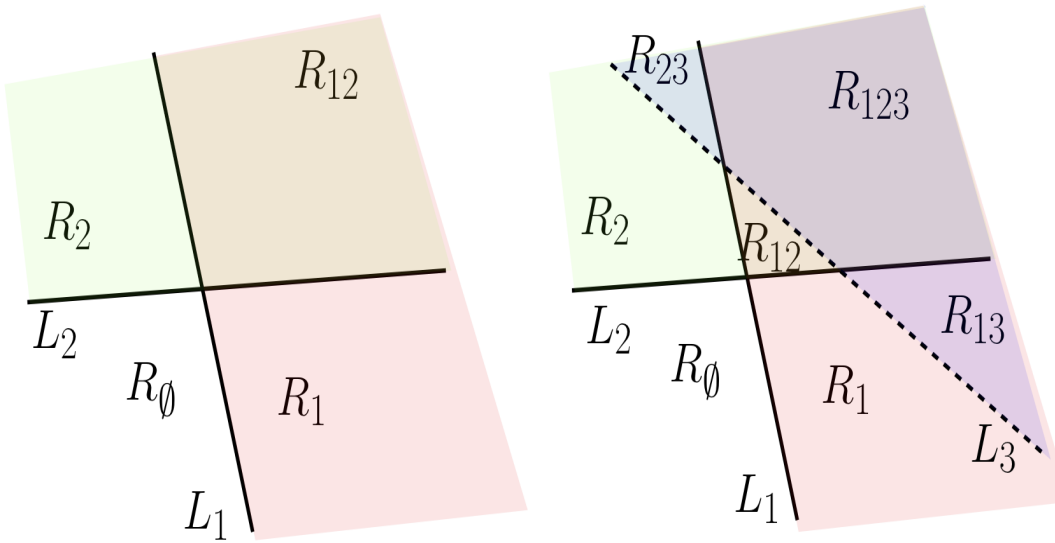


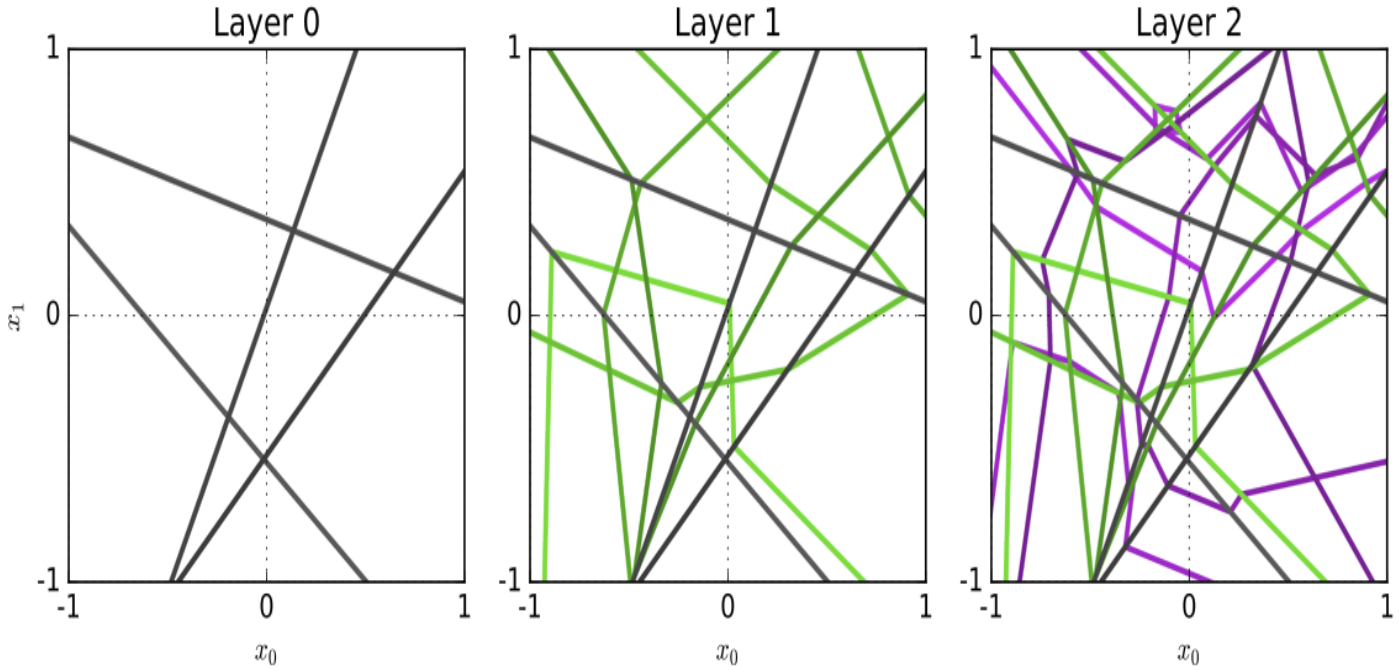
Figure 2: Induction step of the hyperplane sweep method for counting the regions of line arrangements in the plane.

`https://arxiv.org/pdf/1312.6098.pdf`

ReLU hyperplane arrangement

Partition of the input domain $\mathbb{R}^{n_0}$: plot Raghu et al. 16'

Mathematical Institute



`https://arxiv.org/abs/1606.05336`

This "activation region" perspective is a useful intuition for ReLU, but lacks the quantitative convergence rates we observed in more recent approximation theory results of Yarotsky 16'.

A **random network** $f_{NN}(x; \mathcal{P}, \mathcal{Q})$ denotes a deep neural network:

$$h^{(d)} = W^{(d)} z^{(d)} + b^{(d)}, \qquad z^{(d+1)} = \phi(h^{d}), \qquad d = 0, \ldots, L-1,$$

which takes as input the vector x, and is parameterised by random weight matrices $W^{(d)}$ with entries sampled iid from the distribution $\mathcal{P}$, and bias vectors $b^{(d)}$ with entries drawn iid from distribution $\mathcal{Q}$.

While our goal is always to train a network, DNNs typically start as random networks which influence their ability to be trained.

Popular choices are Gaussian, $\mathcal{P} = \mathcal{N}(0, \sigma_w^2)$, or uniform, $\mathcal{P} = \mathcal{U}(-C_w, C_w)$ initialisations.
(*Note, for random networks we use $\phi(\cdot)$ as the nonlinear activation and σ to denote variance.)

Raghu et al. 16' introduced the notion of trajectory length

$$l(x(t)) = \int_t \left|\left| \frac{dx(t)}{dt} \right|\right| dt.$$

as a measure of expressivity of a DNN. In particular, they considered passing a simple geometric object $x(t)$, such as a line $x(t) = tx_0 + (1-t)x_1$ for $x_0, x_1 \in \mathbb{R}^k$ and measure the expected length of the output of the random DNN at layer d:

$$\frac{\mathcal{E}\left[\ell(z^{(d)})\right]}{\ell(x(t))}$$

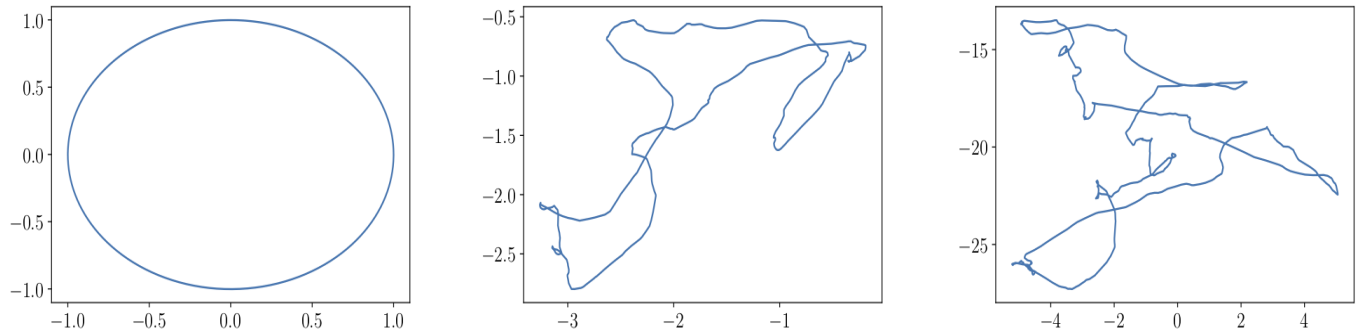
`https://arxiv.org/abs/1606.05336`

Example of circle passed through a random DNN

Complexity of output increasing with depth

UNIVERSITY OF OXFORD
Mathematical Institute

A circle passed through a random DNN and the pre-activation output $h^{(d)}$ at layers $d = 6$ and 12.



(a) Input (b) Layer 6 (c) Layer 12

Figure 1: A circular trajectory, passed through a ReLU network with $\sigma_w = 2$. The plots show the pre-activation trajectory at different layers projected down onto 2 dimensions.

DNNs can be used to *generative* data, GANs, and there one might consider the complexity of the manifold the GAN can generate as a measure of expressivity.

Random DNN: expected path length lower bound

Path length bound dependence on σ_w.

Mathematical Institute

Consider random DNNs of width n and depth L with weights and bias are drawn i.i.d. $W^{(\ell)}(i,j) \sim \mathcal{N}(0, \sigma_w^2/n)$, $b^{(\ell)}(j) \sim \mathcal{N}(0, \sigma_b^2)$

Theorem (Raghu et al. 16')

Consider as input a one dimensional trajectory $x(t)$ with arc-length $\ell(x(t)) = \int_t \left\| \frac{dx(t)}{dt} \right\| dt$ and let $z^{(L)}(t)$ be the output of the Gaussian random feedforward network with ReLu activations, then

$$\frac{\mathcal{E}\left[\ell(z^{(L)})\right]}{\ell(x(t))} \geq \mathcal{O}\left(\left(\frac{\sigma_w}{(\sigma_w^2+\sigma_b^2)^{1/4}} \cdot \frac{n^{1/2}}{(n+(\sigma_w^2+\sigma_b^2)^{1/2})^{1/2}}\right)^L\right).$$

https://arxiv.org/abs/1606.05336

Exponential growth of path length with depth.

Empirical experiments for htanh activation

UNIVERSITY OF OXFORD
Mathematical Institute

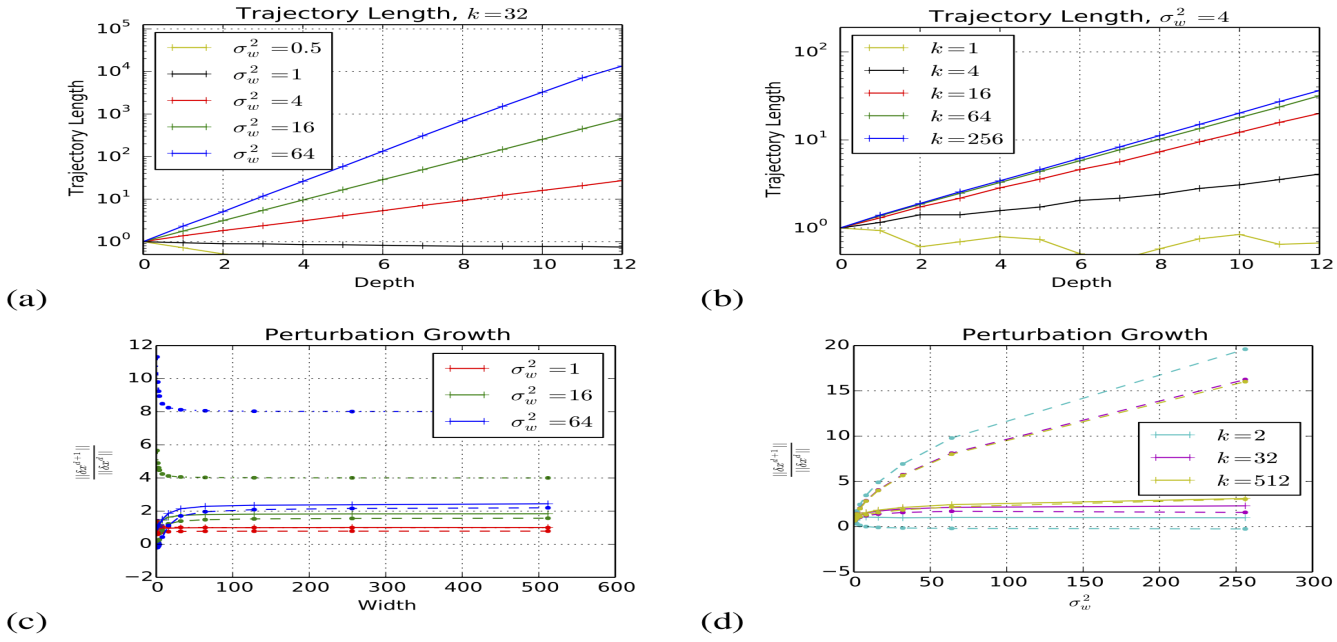


Figure 2: The exponential growth of trajectory length with depth, in a random deep network with hard-tanh nonlinearities. A circular trajectory is chosen between two random vectors. The image of that trajectory is taken at each layer of the network, and its length measured. *(a,b)* The trajectory length vs. layer, in terms of the network width k and weight variance σ_w^2, both of which determine its growth rate. *(c,d)* The average ratio of a trajectory's length in layer $d+1$ relative to its length in layer d. The solid line shows simulated data, while the dashed lines show upper and lower bounds (Theorem 1). Growth rate is a function of layer width k, and weight variance σ_w^2.

https://arxiv.org/pdf/1611.08083.pdf

Random DNN: expected path length lower bound

Generalised and simplified lower bound

Mathematical Institute

Theorem (Price et al. 19')

Let $f_{NN}(x; \alpha, \mathcal{P}, \mathcal{Q})$ be a random sparse net with layers of width n. Then, if $\mathrm{E}[|u^T w_i|] \geq M\|u\|$, where w_i is the i^{th} row of $W \in \mathcal{P}$, and u and M are constants, then

$$\mathrm{E}[l(z^{(L)}(t))] \geq \left(\frac{M}{2}\right)^L \cdot l(x(t))$$

for $x(t)$ a 1-dimensional trajectory in input space.

Exponential growth with depth for random initialisations such as Gaussian, uniform, and discrete; e.g. for Gaussian $M = \sigma_w\sqrt{2/\pi}$.
`https://arxiv.org/abs/1911.10651`

Observed growth rate (solid) and bounds (dashed)

Empirical experiments showing dependence on σ_w and sparsity α

University of Oxford
Mathematical Institute

Price et al. 19' also extended the results to have all but α fraction of the entries in W equal to 0.

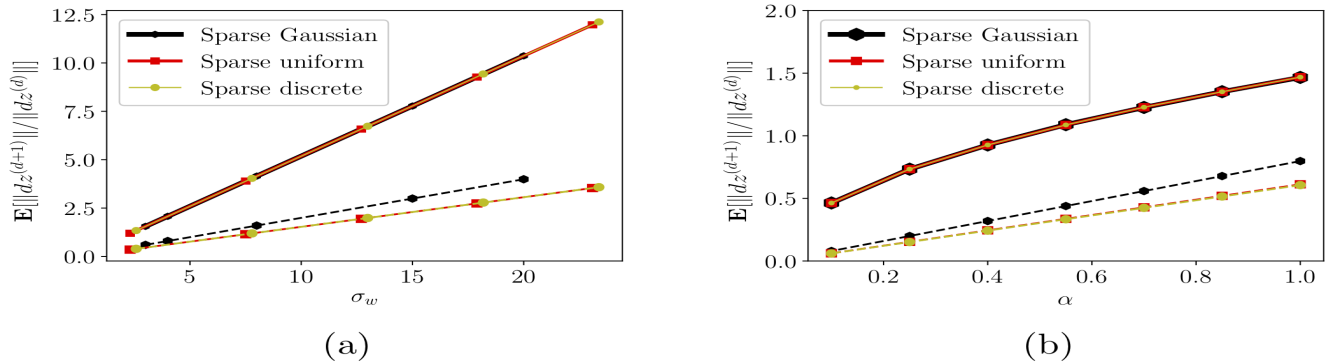


Figure 3: Expected growth factor, that is, the expected ratio of the length of any very small line segment in layer $d+1$ to its length in layer d. Figure 3a shows the dependence on the variance of the weights' distribution, and Figure 3b shows the dependence on sparsity.

Unless σ_w or α small enough at initialisation the pre-activiation output is exponentially complex.

`https://arxiv.org/abs/1911.10651`