

Exponential expresivity with depth.



Mathematical
Institute

THEORIES OF DEEP LEARNING: C6.5,
LECTURE / VIDEO 3
Prof. Jared Tanner
Mathematical Institute
University of Oxford

Oxford
Mathematics



DNNs as function approximators

Functions act as classifiers and other machine learning tasks

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

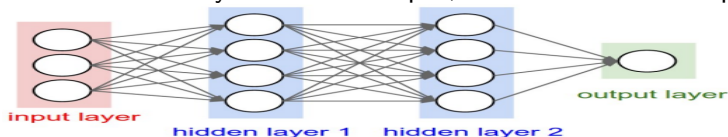
Repeated affine transformation followed by a nonlinear action:

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

where $W^{(i)} \in \mathbb{R}^{n_{i+1} \times n_i}$ and $b^{(i)} \in \mathbb{R}^{n_{i+1}}$ and $\sigma(\cdot)$ is a nonlinear activation such as ReLU, $\sigma(z) := \max(0, z) = z_+$.

The input is h_1 , the output is h_N , and h_i for intermediate $i = 2, \dots, N - 1$ 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 $C_n([0, 1])$

Consider the feedforward network with one hidden layer:

input $h_1 = x \in \mathbb{R}^n$

hidden layer $h_2 = \sigma(W^{(1)}h_1 + b^{(1)}) \in \mathbb{R}^m$

output $H(x, \theta) = \alpha^T h_2 = \sum_{i=1}^m \alpha_i \sigma(w_i^T x + b_i)$

with $\sigma(t) \in [0, 1]$, say $\sigma(t) = 1/(1 + e^{-t})$.

Theorem (Cybenko 89')

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

Consider the feedforward network with one hidden layer:

input $h_1 = x \in \mathbb{R}^n$

hidden layer $h_2 = \sigma(W^{(1)}h_1 + b^{(1)}) \in \mathbb{R}^m$

output $H(x, \theta) = \alpha^T h_2 = \sum_{i=1}^m \alpha_i \sigma(w_i^T x + b_i)$

with $\sigma(t) \in [0, 1]$ non-constant.

Theorem (Hornik 90')

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

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

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

Representational benefits of depth (Telgarsky 15')

Two layer ReLU network: sawtooth basis function

Telegarsky (2015) considered a specific construction of a function from a deep network which requires an shallow network to have exponential width.

Let $\sigma(x) = \text{ReLU}(x) = \max(x, 0)$ and consider the two layer net:

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

and $h_3(x) = \sigma(h_2(x))$ 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

For $\sigma(x) = \max(x, 0)$ let $f(x) = h_3(x) = \sigma(2\sigma(x) - 4\sigma(x - 1/2))$ 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 $i = 0, 1, \dots, 2^k$ and moreover takes on the values $f^k(x_i) = 0$ for i even and $f^k(x_i) = 1$ for i odd.

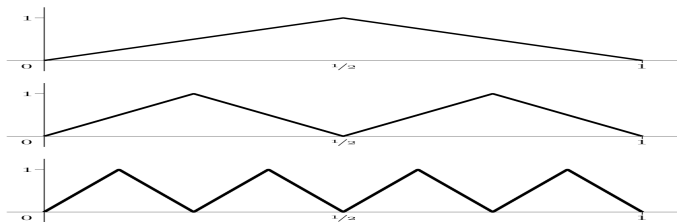


Figure 2: f_m , f_m^2 , and f_m^3 .

Representational benefits of depth (Telgarsky 15')

Composition gives exponential growth in complexity: width vs. depth

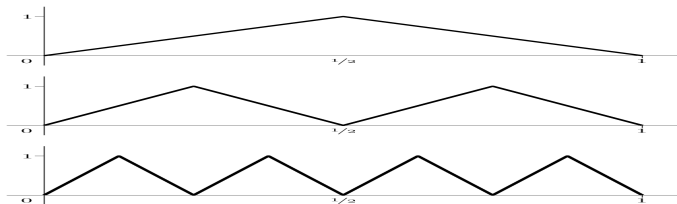


Figure 2: f_m , f_m^2 , and f_m^3 .

In contrast, a two-layer network with the same $\sigma(x)$ of the form $\sigma\left(\sum_{j=1}^m \alpha_j \sigma(w_j x - b_j)\right)$ 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

Define the function class $F(\sigma; m, \ell)$ be the space of functions composed of ℓ layer fully connected m width feed forward nets with nonlinear activation function σ . 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

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

<https://arxiv.org/abs/1509.08101>

Representational benefits of depth (Yarotsky 16')

ReLU nets can approximate x^2 exponentially well

Returning to the saw-tooth function composed of

$\sigma(x) = \max(x, 0)$ let $f(x) = h_3(x) = \sigma(2\sigma(x) - 4\sigma(x - 1/2))$ and iterate this 2-layer network k times to obtain a $2k$ -layer network $f^k(x) = f(f(\dots(f(x)\dots)))$.

Let $h_m(x)$ denote the piecewise linear interpolation of $h(x) = x^2$ at 2^{m+1} equispaced points, then

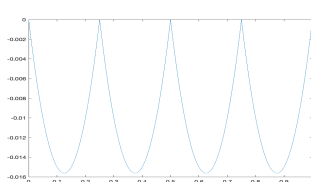
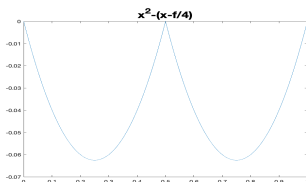
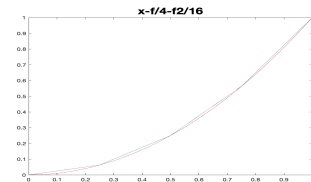
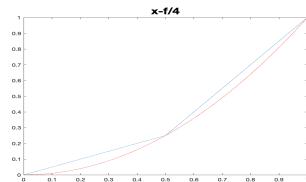
$$h_m(x) = x - \sum_{s=1}^m 2^{-2s} f^s(x).$$

Consequently, $h(x) = x^2$ can be approximated on $[0, 1]$ to uniform accuracy ϵ by a ReLU network having 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 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

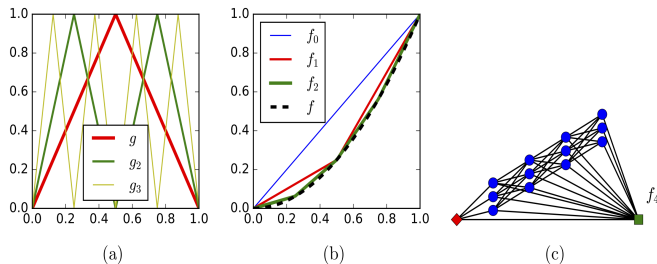


Figure 2: Fast approximation of the function $f(x) = x^2$ from Proposition 2: (a) the "tooth" function g and the iterated "sawtooth" functions g_2, g_3 ; (b) the approximating functions f_m ; (c) the network architecture for f_4 .

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

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

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

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

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.

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

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 .

Representational benefits of depth (Yarotsky 16')

Sketch of the proof 1 of 2

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; $\sum_m \phi_m(x) = 1$. E.g. $\phi_x(x) = \prod_{k=1}^d \psi(3N(x - \frac{k-1}{N}))$

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

Then approximate each $\phi_m(x)f(x)$ locally with Taylor series around the mid point, leading to an error bounded by $2^d d^n / N^n n!$; giving an ϵ bound if $N \geq (n! \epsilon / 2^d d^n)^{-1/n}$.

Representational benefits of depth (Yarotsky 16')

Sketch of the proof 2 of 2

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

Optimal function approximation ability of deep networks

DNNs can achieve optimal rates for function classes

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, α -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.