Statistical Learning Theory

and

Applications

2025

Class 13

Deep Learning:

approximation theory

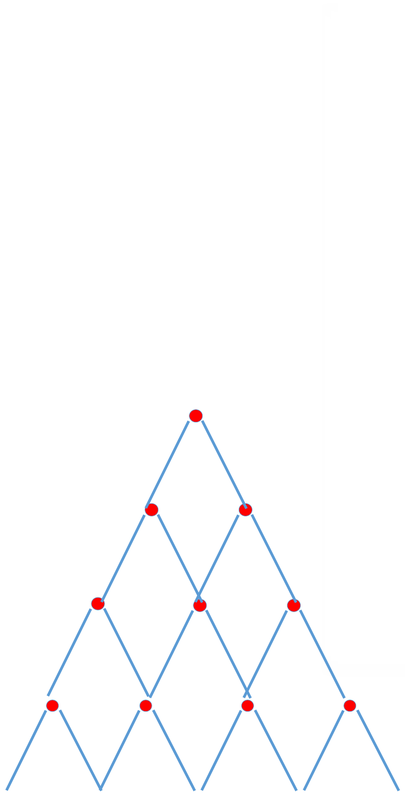
TOMASO POGGIO, MIT, CBMM

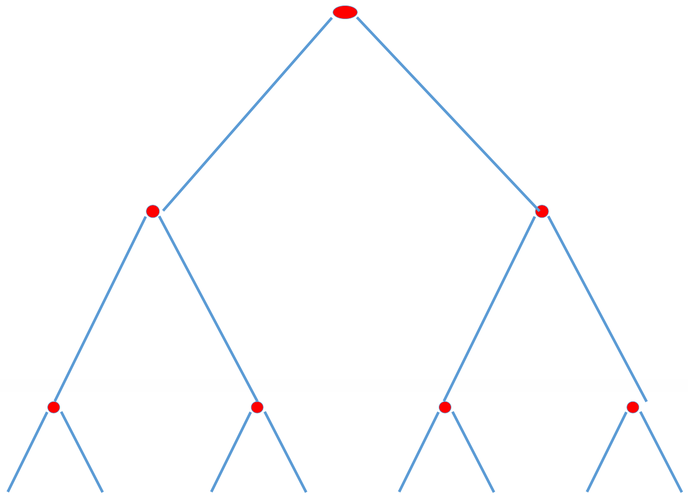
Preamble

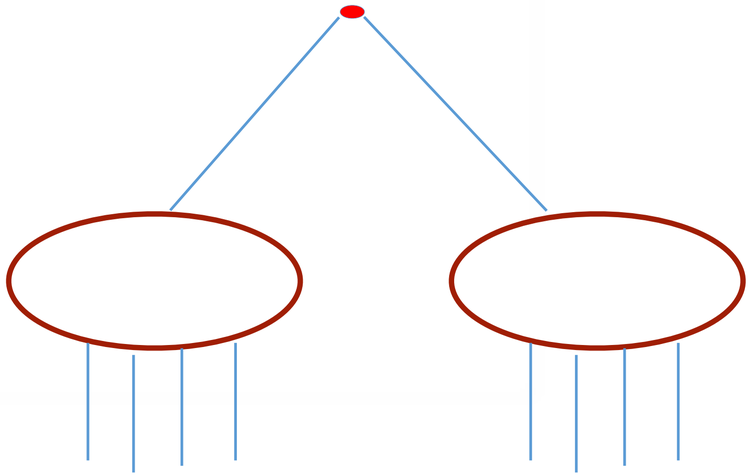
Key principle in ML architectures: (compositional) sparsity

The thesis of this course — its conceptual backbone — is that (sparse) compositionality is key in ML. The idea that sparsity is fundamental seems to be new — it is a conjecture which may be wrong. We will see it first in approximation theory, then in generalization and then in optimization.

Graphical Notattion







Conceptual Framework of Machine Learning

1. choose $\mathbb{H}_\lambda$, a space $\mathbb{H}_\lambda$ of functions with parameters λ - in which to search for f. The functions in $\mathbb{H}_\lambda$ must have a *non-exponential* number of parameters (in the effective dimensionality) AND be powerful enough to approximate a broad range of target functions (realm of approximation theory);

2. optimize the values of the parametric approximator by minimizing a loss function on a training set (realm of optimization theory);

3. study generalization properties of the trained representation (domain of learning theory).

Conceptual Framework of Machine Learning

In fact we minimize $E_S = \frac{1}{n}\sum_{i=1}^{n}\left(f\left(x_i\right) - y_i\right)^2 + \lambda\rho^2$ over $f \in \mathbb{F}$ where $\mathbb{F}$ is the hypothesis space of deep sparse networks to obtain a unique hypothesis $f_z : X \to Y$, called the *empirical minimizer*. Thus we focus on the problem of estimating $\int_X \left(f_z - f_\mu\right)^2 d\mu$.

It is useful to reformulate the problem by defining a "true optimum" $f_{\mathbb{F}}$ by taking the minimum over $\int_X \left(f - f_\mu\right)^2$. Thus we have

$$\int_X \left(f_z - f_\mu\right)^2 = S(z, \mathbb{F}) + \int_X \left(f_{\mathbb{F}} - f_\mu\right)^2$$
$$= S(z, \mathbb{F}) + A(\mathbb{F})$$

where the *sample error* (variance) is $S(z, \mathbb{F}) = \int_X \left(f_z - f_\mu\right)^2 - \int_X \left(f_{\mathbb{F}} - f_\mu\right)^2$ and $A(\mathbb{F})$ is the *approximation error* (bias).

Main Message of the approximation classes

Deep sparse NNs are universal, efficient approximators

Corollary Every function which is efficiently Turing computable can be approximated, without curse of dimensionality, by a *compositionally sparse, deep* network.

Poggio, Fraser , Bulletin of the AMS, 2024

Approximation classes

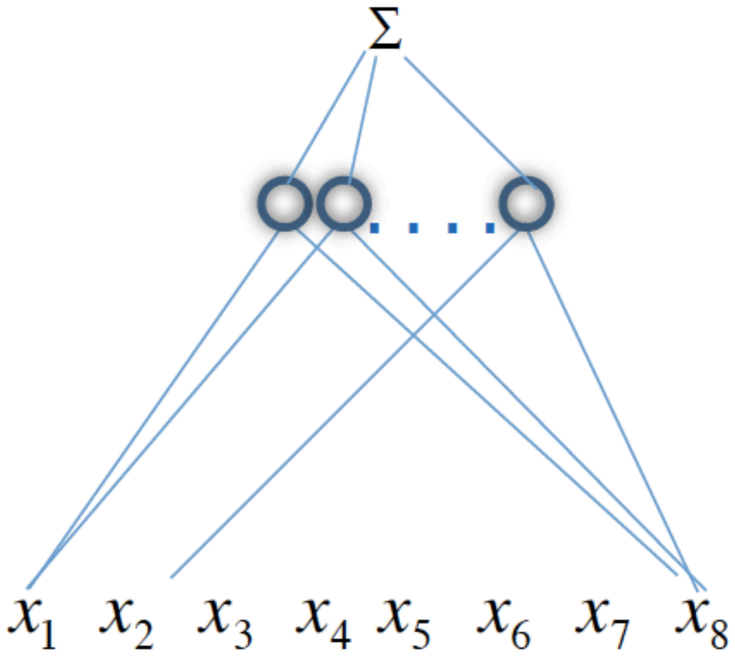
- Shallow and Deep Networks
- Universality, degree of approximation, curse of dimensionality
- Compositionally sparse functions
- Equivalence between compositional sparsity and efficient Turing computability
- Implications: deep sparse networks are universal

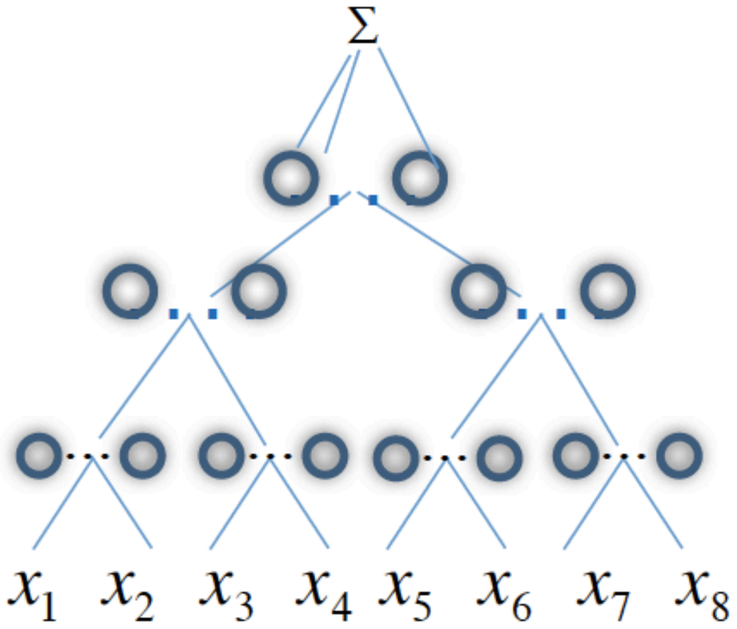
Approximation theory

Let us start with shallow networks

Deep and shallow networks: approximation universality

Theorem *Assume as target continuous functions on a compact domain. Shallow, one-hidden layer networks with a a nonlinear $\sigma(x)$ (which is not a polynomial) can approximate such targets arbitrarily well. Deep networks with a nonlinear $\sigma(x)$ (including polynomials) can also approximate such target functions arbitrarily well.*





$$\phi(x) = \sum_{i=1}^{r} c_i \left| < w_i , x > + b_i \right|_+$$

Cybenko, Girosi, ….

Universality of shallow (and deep) networks

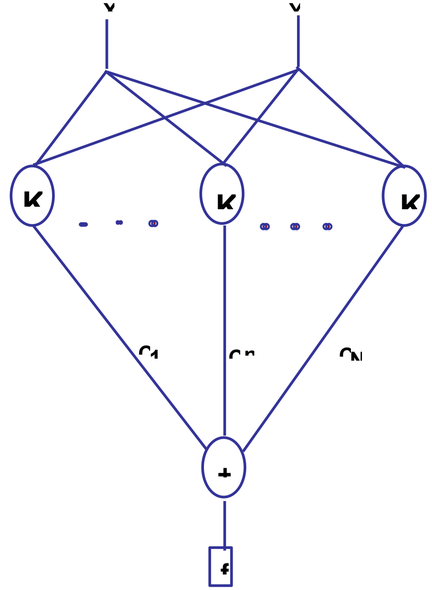
This is a density result: what is a famous one, very similar?

Typical shallow networks: Kernel Machines

$$\min_{f \in H} \left[\frac{1}{\ell} \sum_{i=1}^{\ell} V(f(x_i) - y_i) + \lambda \left\| f \right\|_K^2 \right]$$

implies

$$f(\mathbf{x}) = \sum_i^l \alpha_i K(\mathbf{x}, \mathbf{x}_i)$$



Equation includes splines, Radial Basis Functions and Support Vector Machines (depending on choice of V).

RKHS were explicitly introduced in learning theory by Girosi (1997), Vapnik (1998).
Moody and Darken (1989), and Broomhead and Lowe (1988) introduced RBF to learning theory. Poggio and Girosi (1989) introduced Tikhonov regularization in learning theory and worked (implicitly) with RKHS. RKHS were used earlier in approximation theory (eg Parzen, 1952-1970, Wahba, 1990).

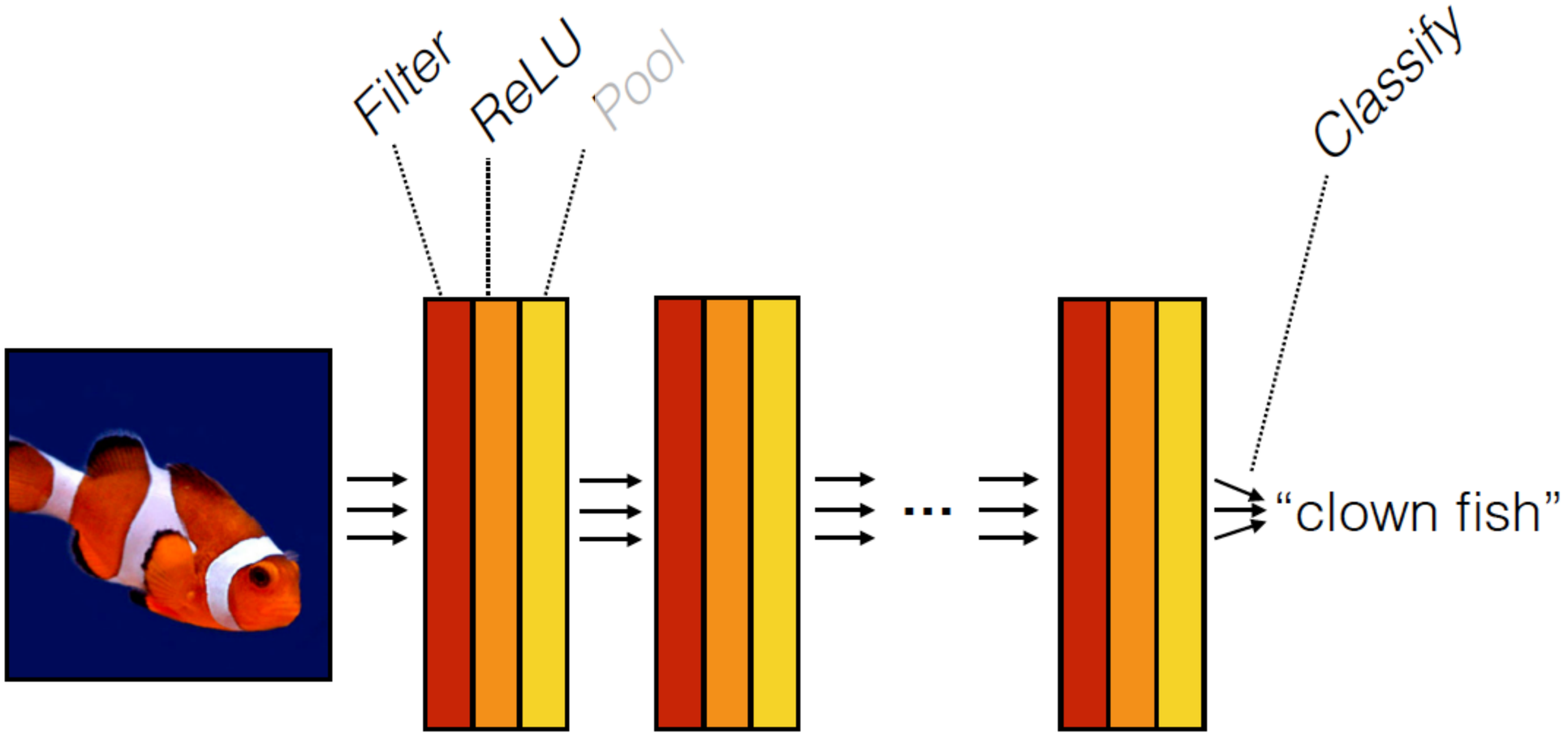
CENTER FOR
Brains
Minds+
Machines

Mhaskar, Poggio, Liao, 2016

Approximation theory

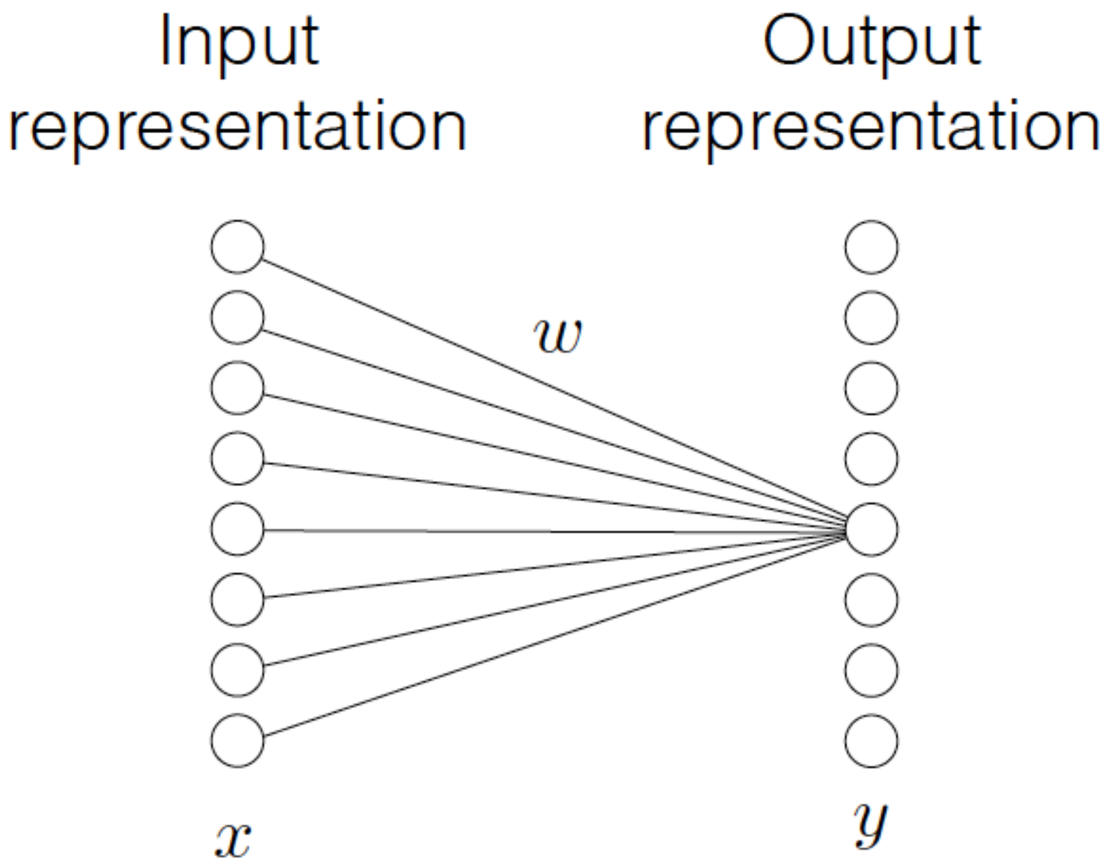
Now…deep networks…

Computation in a neural net



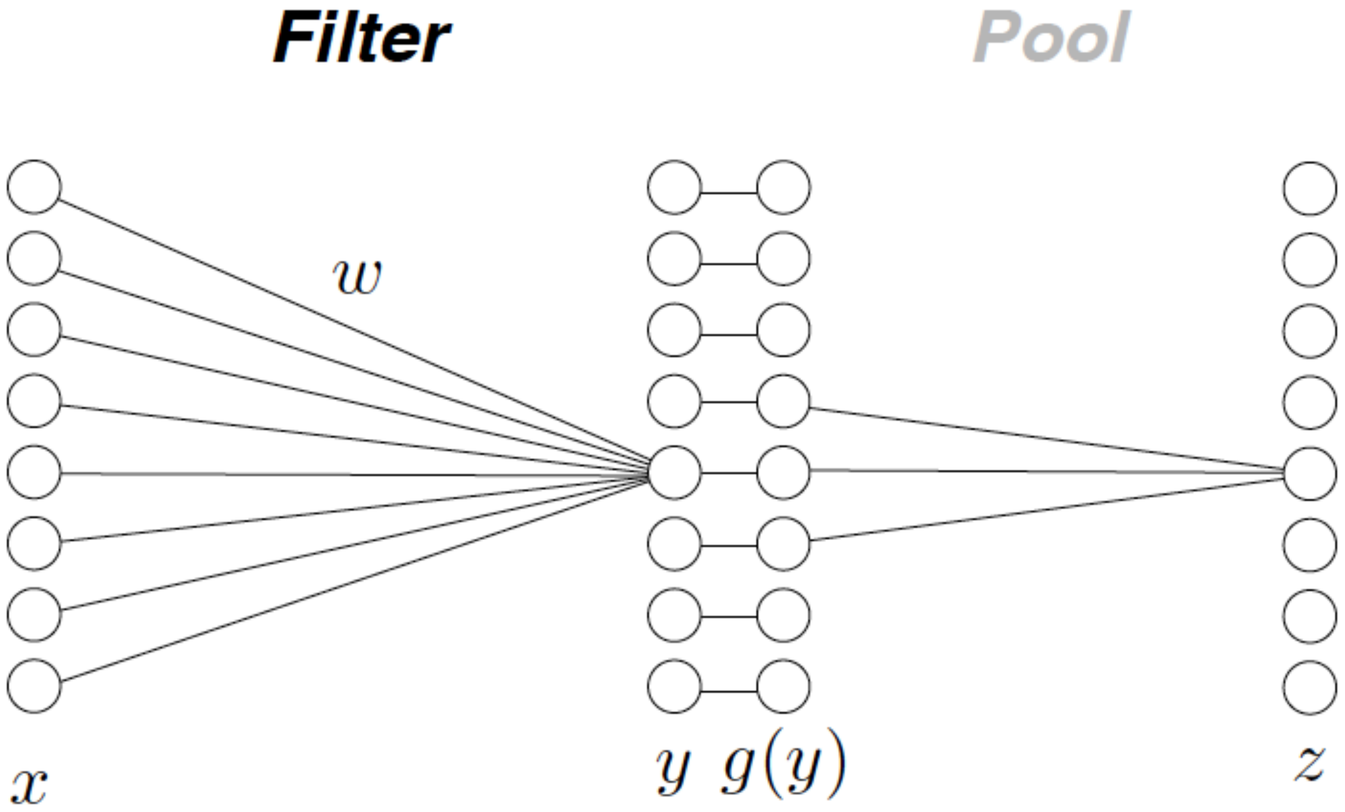
$$f(\mathbf{x}) = f_L(\ldots f_2(f_1(\mathbf{x})))$$

Computation in a neural net



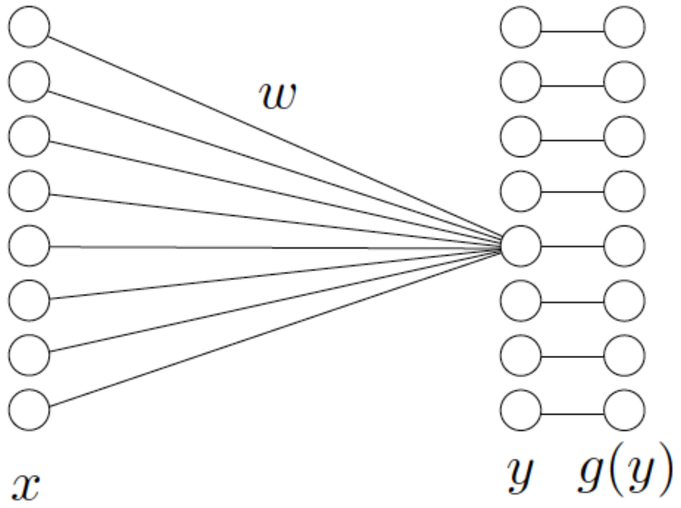
$$y_j = \sum_i w_{ij} x_i$$

Computation in a neural net

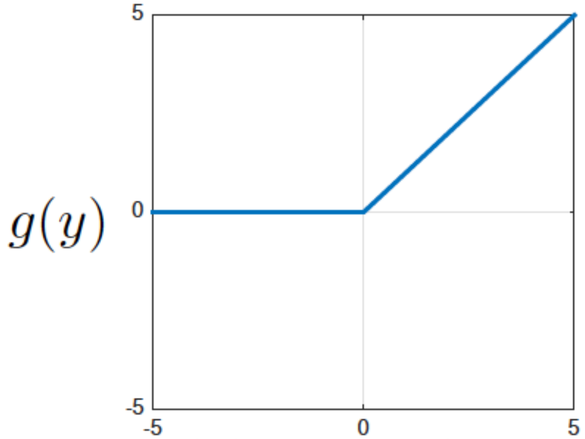


$$y_j = \sum_i w_{ij} x_i \qquad z_k = \max_{j \in \mathcal{N}(j)} g(y_j)$$

Deep nets architecture with RELU activations



Rectified linear unit (ReLU)



$$g(y) = \max(0, y)$$

GELU also used

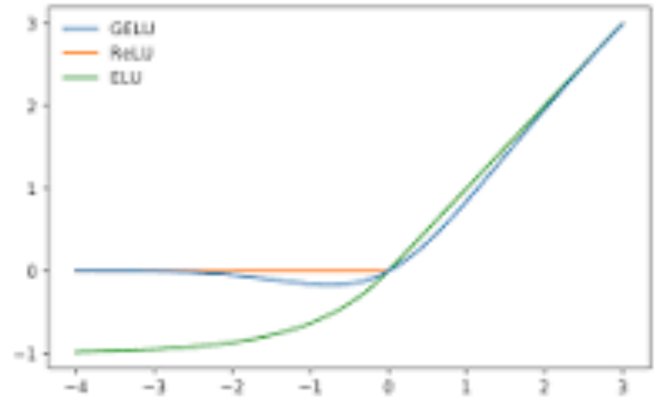


Figure 1: The GELU ($\mu = 0, \sigma = 1$), ReLU, and ELU ($\alpha = 1$).

CENTER FOR
Brains
Minds+
Machines

Old questions (in 2022)…now answered in these classes!

Can deep nets represent/approximate every function of a finite number of variables?

Can deep nets escape the curse of dimensionality ?

Under which conditions are functions sparse?

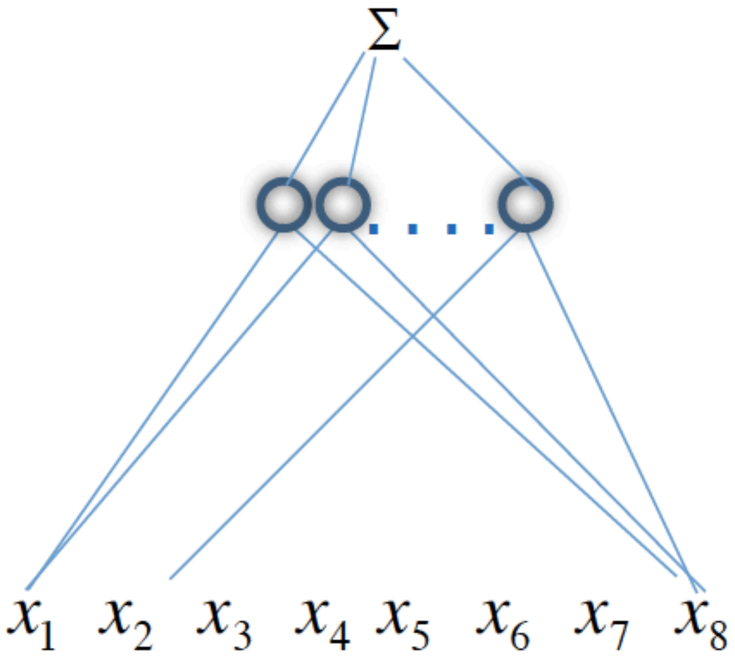
When and why are deep networks better than shallow networks?

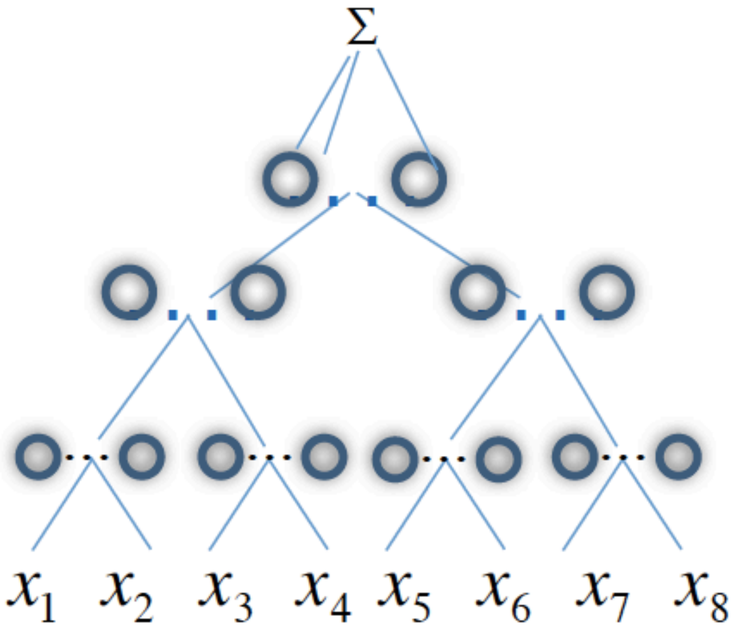
Approximation class

- Shallow and Deep Networks ✓
- Universality, degree of approximation, curse of dimensionality
- Compositionally sparse functions
- Equivalence between compositional sparsity and efficient Turing computability
- Implications: deep sparse networks

Deep and shallow networks: representational universality

Theorem *Shallow, one-hidden layer networks with a nonlinear $\phi(x)$ which is not a polynomial are universal. Arbitrarily deep networks with a nonlinear $\phi(x)$ (including polynomials) are universal.*

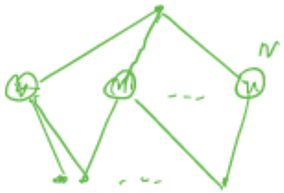




$$\phi(x) = \sum_{i=1}^{r} c_i \left| < w_i , x > + b_i \right|_+$$

Cybenko, Girosi, ….

Functions and approximators



$f \in C(\mathbb{R}^d)$ $\quad |f(x+\delta) - f(x)| < \varepsilon$

$g \in V_N$ $\quad$ span $\sigma(w^{iT} x_T)$

Density

$$\forall f \in C\left(\mathbb{R}^d_{[0,1]}\right)$$

$$\exists g \in V_n \qquad \sup \left| f(x) - g(x) \right| < \varepsilon$$

Degree of approximation

$\forall f \in C(\mathbb{R}^d)$

$\inf_{g \in V_N} |f - g_N| < \varepsilon \Rightarrow$

how large ?

$$N \approx T(\epsilon)$$

Degree of approximation

How *complex* a network ought to be to theoretically guarantee approximation of an unknown target function f up to a given accuracy $\epsilon > 0$? To measure the accuracy, we need a norm $\|\cdot\|$ on some normed linear space $\mathbf{X}$. We use in this class the sup norm.

Let V_N be the be set of all networks of a given kind with complexity N (for instance the total number of units in the network. The degree of approximation is defined by

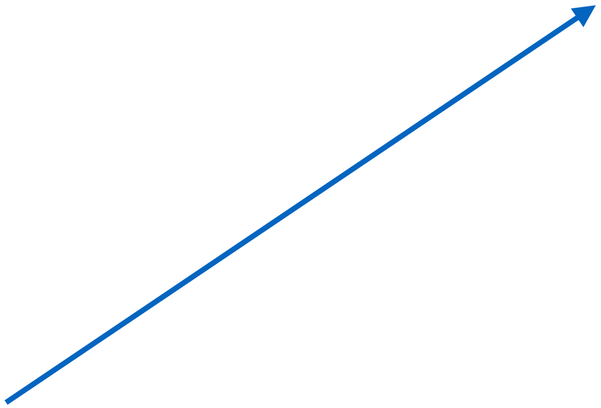
$$\mathrm{dist}\left(f, V_N\right) = \inf_{P \in V_N} \|f - P\|.$$

For example, if $\mathrm{dist}\left(f, V_N\right) = \mathcal{O}\left(N^{-\gamma}\right)$ for some $\gamma > 0$, then a network with complexity $N = \mathcal{O}\left(\epsilon^{-\frac{1}{\gamma}}\right)$ will be sufficient to guarantee an approximation with accuracy at least ϵ. Since f is unknown, we need to make some assumptions on the class of functions from which the unknown target function is chosen. This a priori information is codified by the statement that $f \in W$ for some subspace $W \subseteq \mathbb{X}$. This subspace is usually a smoothness class characterized by a smoothness parameter s. Here it will be generalized to a smoothness and compositional class, characterized by the parameters s and d.

Degree (or rate) of approximation (theorem)

$$f(x_1, x_2, ..., x_8)$$

Both shallow and deep network can approximate a function of d variables equally well (density). The number of parameters in both cases depends exponentially on d as $N = O(\epsilon^{-\frac{d}{s}})$ or $\epsilon = O(N^{-\frac{s}{d}})$



Curse of dimensionality

Mhaskar, Poggio, Liao, 2016

Networks and Sobolev spaces

We assume that the activation nonlinearity is a smoothed version of the so called ReLU. Each node h_i is a ridge function, comprising one or more neurons.

Let $I^d = [-1,1]^d, \mathbb{X} = C\left(I^d\right)$ be the space of all continuous functions on I^d, with $\|f\| = \max_{x \in I^d} |f(x)|$. Let $\mathcal{S}_{N,d}$ denote the class of all shallow networks with N units of the ridge function form

$$x \mapsto \sum_{k=1}^{N} a_k \sigma\left(\left\langle w_k, x \right\rangle\right),$$

where w_k in $R^d, a_k \in R$. The number of trainable parameters here is N. Let $m \geq 1$ be an integer, and W_m^d be the set of all functions of d variables with continuous partial derivatives of orders up to $m < \infty$ such that $\|f\| + \sum_{1 \leq |\mathbf{k}|_1 \leq m} \left\| D^{\mathbf{k}} f \right\| \leq 1$, where $D^{\mathbf{k}}$ denotes the partial derivative indicated by the multi-integer $\mathbf{k} \geq 1$, and $|\mathbf{k}|_1$ is the sum of the components of $\mathbf{k}$.

Theorem

$g(x) = \sum^{N} c_i \sigma(\langle \underline{w}^i, \underline{x} \rangle)$

$\forall f \in W_m^d, \exists g \in V_N$ s.t.

$|g(x) - f(x)| < \varepsilon$ with

$N = O(\varepsilon^{-d/m})$

Summary proof

Look at Pinkus, *Acta Numerica*, 1999

$$\sum_{k=1}^{r} a_k \sigma\left(\langle w_k, x \rangle\right),$$

① $\sum_i^k c_i \sigma(a_i x + b_i) \approx p(x) \in \Pi_{k-1}$

② $\sum_i^r p_i^k(\langle w_i, x \rangle) \in P_r^d$ (pol up degree k in d variables)

$r \sim k^d \Rightarrow k \sim r^{1/d}$

③ Sobolev $E(W_d^m, R_r, L_p) \leq C k^{-m}$ — space of ridge functions with r elements

② + ③ $\Rightarrow$ $E \leq C r^{-m/d}$

$\sum_{i=1}^{r} g_i(\langle a_i \cdot, x \rangle)$: $a_i \in \mathbb{R}^n$, $g_i \in C(\mathbb{R})$, $i = 1, \ldots, r$

(1)

Theorem

If σ is not a polynomial, $\sigma \in C^\infty$, the closure of V_R (span of R σ) contains the linear space of $\mathbb{T}^{R-1}$

Proof

$$\lim_{h \to 0} \frac{\sigma((a+h)x+b) - \sigma(ax+b)}{h} =$$

$$= \frac{d}{da} \sigma(ax+b)\Big|_{a=0} = x\sigma'(ax+b)$$

σ is a nonlinear smooth function such as a smooth RELU; consider two units with

$$\sigma((w+\delta)x+b) - \sigma(wx+b)\big|_{w=0} \approx x\sigma'(b)$$

$$x\sigma'((w+\delta)x+b) - x\sigma'(wx+b)\big|_{w=0} \approx x^2\sigma''(+b)$$

Thus V_R is dense in $C(K)$ because of Weierstrass theorem.

Parenthetical remark: How to get squares from a shallow network

σ is a nonlinear smooth function such as a smooth RELU; consider two units with

$$\sigma((w+\delta)x+b)-\sigma(wx+b)\approx x\sigma'(wx+b)$$

$$x\sigma'((w+\delta)x+b)-x\sigma'(wx+b)\approx x^2\sigma''(wx+b)$$

$$\frac{\partial^k \sigma(wx+b)}{\partial w^k}\Big|_{w=0} = x^k\sigma^{(k)}(b)$$

From squares to polynomials….

$$(x+y)^2-x^2-y^2=2xy$$

From 1D to dD

$$P(x) = \sum_{i}^{r} P_i(\langle w^i, x \rangle)$$

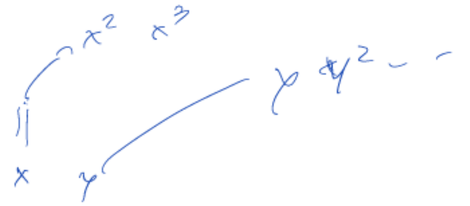
P_i: univariate, $\langle w^i, x \rangle \in \mathbb{R}$

$P \in H_k^d$ (homogeneous pol in d variables of degree k)

$$r = \dim H_k = \binom{d+k-1}{k} = \frac{(d+k-1)!}{(d-1)!\,k!} =$$

$$\sim k^d$$

- Thus H_k^d pol. can be represented by a network with $r \sim k^d$ units



① + ② $\Rightarrow$ a shallow net

represents P_k^d with

$\geq$ units $\geq n \, k^d$

③ Define $E(B, X, L_p) =$

$$= \inf_{P \in X} \|P - f\|_{L_p} \qquad \forall f \in B$$

Theorem

$$E(W_d^m, N_n, L_p) \le C n^{-m/d-1}$$

$$N_n(\sigma)(\langle w^i, x \rangle) = \sum_{i=1}^{n} \sigma(\langle w^i, x \rangle)$$

proof

$$E(W^m, P_n, L_p) \le C n^{-m}$$

since $n \sim k^{d-1} \Rightarrow k \sim n^{1/d-1}$

$$\Rightarrow E(W^m, N_n, L_p) \sim E(W_1^m, P_n L_p) \sim C n^{-m/d-1}$$

Summary proof

① $\sum_i^k c_i \sigma(a_i x + b_i) \sim p(x) \in \Pi_{k-1}$

② $\sum_i^r p_i^k(\langle w_i, x \rangle) \in P_k^d$ (pol up degree k in d variables)

$r \sim k^d \Rightarrow k \sim r^{1/d}$

③ Sobolev $E(W_d^m, R_r, L_p) \leq C k^{-m}$ (space of ridge functions with r elements)

② + ③ $\Rightarrow$ $E \leq C r^{-m/d}$

$\sum_i^r g_i(\langle a^i, x \rangle)$: $a_i \in \mathbb{R}^n$ $g_i \in C(\mathbb{R})$ $i = 1, \dots, r$

Let us repeat: I will state theorem and proof again

Theorem 1. *Let* $\sigma : R \to R$ *be infinitely differentiable, and not a polynomial. For* $f \in W_s^d$ *the complexity of shallow networks that provide accuracy at least* ϵ *is*

$N = \mathcal{O}\left(\epsilon^{-d/s}\right)$ *and is the best possible.*

A parenthesis on the curse of dimensionality $N = \mathcal{O}\left(\epsilon^{-d/s}\right)$. One of the most influential principle in discovered and cultivated by mathematicians: the curse (and the blessings) of dimensionality.

Proof of Theorem (sketch)

The standard approach to prove degree of approximations uses polynomials. It can be summarized in three steps:

1. Let us denote with $\mathscr{H}_k$ the linear space of homogeneous polynomials of degree k in R^d and with $P_k = \bigcup_{s=0}^{k} \mathscr{H}_s$ the linear space of polynomials of degree at most k in d variables. Set

   $r = \binom{d-1+k}{k} = \dim \mathscr{H}_k$ and denote by π_k the space of univariate polynomials of degree at most k. We recall that the number of monomials in a polynomial in d variables with total degree $\leq N$ is $\binom{d+N}{d}$ and can be written as a linear combination of the same number of terms of the form $(\langle w, x \rangle + b)^N$.

We first prove that

$$P_k(x) = \operatorname{span}\left(\left(\langle w^i, x \rangle \right)^s : i = 1, \cdots, r, s = 1, \cdots, k \right.$$

and thus, with $p_i \in \pi_k$

$$P_k(x) = \sum_{i=1}^{r} p_i \left(\langle w_i, x \rangle \right).$$

Proof of Theorem (sketch)

2. Second, we prove that each univariate polynomial can be approximated on any finite interval from $\mathcal{N}(\sigma) = \text{span}\{\sigma(\lambda t - \theta)\}, \lambda, \theta \in R$ in an appropriate norm.

3. The last step is to use classical results about approximation by polynomials of functions in a Sobolev space: $E\left(\mathcal{B}_p^m; P_k; L_p\right) \leq Ck^{-m}$ where $\mathcal{B}_p^m$ is the Sobolev space of functions supported on the unit ball in R^n.

The key step from the point of view of possible implementations by a deep neural network with ReLU units is step number 2.

Theorem. *If* $\sigma \in \mathscr{C}(R)$ *is not a polynomial and* $\sigma \in C^{\infty}$ *, the closure of* $\mathscr{N}$ *contains the linear space of algebraic polynomial of degree at most* $r-1$*.*
Since $r \approx k^{d}$ *and thus* $k \approx r^{1/d}$ *we have*

$$E\left(\mathscr{B}_{p}^{m}; P_{k}; L_{p}\right) \leq Cr^{-\frac{m}{d}} .$$

End repeat

Some Remarks

Another way to obtain squares: RELU networks

- First I use a layer with 2 units to compute

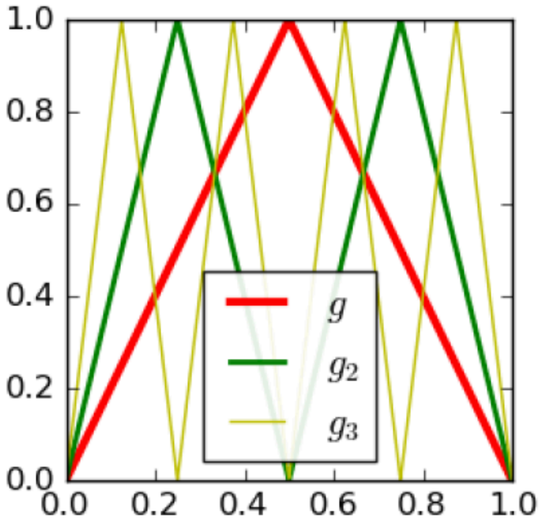
$$g(x) = 2\sigma(x) - 4\sigma(x - \frac{1}{2})$$

that is $g(x) = 2x$ if $0 < x < \frac{1}{2}$

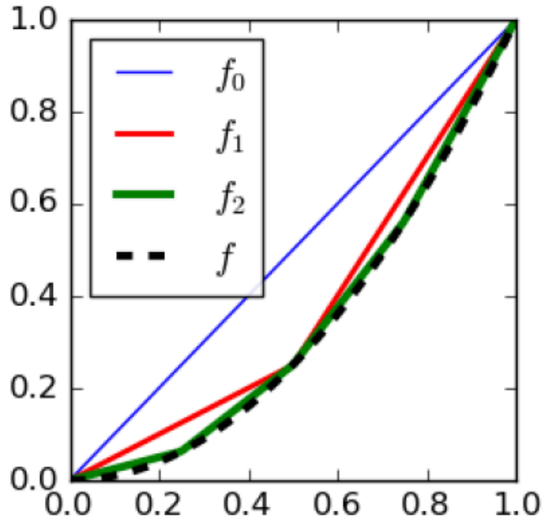
and $g(x) = 2(1 - x)$ if $x \geq \frac{1}{2}$

- Then I compute $g_2(x) = g \circ g(x)$ by composing 2 layers etc for $g_m(x)$
- Then I combine linearly the g_m and compute

$$f_m(x) = x - \sum_{k=1}^{m} \frac{g_k(x)}{2^{2k}}$$ which is a piece-wise linear interpolation of x^2 with $2^m + 1$ uniformly distributed nodes



(a)



(b)

An example of the Tagaki class of (mostly fractals) functions is

$$x(1 - x) = \sum_{k \geq 1} 4^{-k} g_k(x)$$

Yarotsky, 2016

RELUs

Yarotsky proved a theorem showing that the benefit of depth allows the (non-smooth) ReLU network to quickly remove the handicap of crudeness of the ReLU activation function and catch up with shallow networks with smooth activation functions.
A natural question raised by these results is how optimal is the complexity bound $O(e^{-\frac{d}{n}} \log \frac{1}{\epsilon})$ is. It can be shown that for generic functions of fixed smoothness we cannot substantially improve the network's expressiveness ensured by Yarotsky theorem, regardless of the depth we use.

Curse of dimensionality

DeVore result shows that is the best one can do…but how can one fight the curse of dimensionality?

Blessing of smoothness

$$O\left(\varepsilon^{-d/m}\right)$$

A property of continuous functions

Theorem 6. (Kolmogorov, 1957). There exist fixed increasing continuous functions $h_{pq}(x)$, on $I = [0,1]$ so that each continuous function f on I^d can be written in the form

$$f(x_1, \ldots, x_d) = \sum_{q=1}^{2d+1} g_q \left(\sum_{p=1}^{d} h_{pq}(x_p) \right)$$

where g_q are properly chosen continuous functions of one variable.

A property of continuous functions

This means that the $(2d+1)(d+1)$ univariate functions g_q and $\psi_{p,q}$ are enough for an exact representation of a-variate function. Kolmogorov published the result in 1957 disproving the statement of Hilbert's 13th problem that is concerned with the solution of algebraic equations.

A ridge function is a function of the form $f(\mathbf{x}) = \sum_{p=1}^{m} g_p\left(\mathbf{w}_p^{\mathsf{T}}\mathbf{x}\right)$, with vectors $\mathbf{w}_p \in \mathbb{R}^d$ and univariate functions g_p. The structure of the KA representation can therefore be viewed as the composition of two ridge functions. There exists no exact representation of continuous functions by ridge functions. The composition structure is thus essential for the KA representation. A two-hidden-layer feedforward neural network with activation function σ , hidden layers of width m_1 and m_2, and one output unit can be written in the form

$$f(\mathbf{x}) = \sum_{q=1}^{m_1} d_q \sigma\left(\sum_{p=1}^{m_2} b_{pq}\sigma\left(\mathbf{w}_p^{\mathsf{T}}\mathbf{x} + a_p\right) + c_q\right), \text{ with parameters } \mathbf{w}_p \in \mathbb{R}^d, a_p, b_{pq}, c_q, d_q \in \mathbb{R}\,.$$

A property of continuous functions

SchmiedHieber (2021) derived modifications of the Kolmogorov-Arnold representation that transfer smoothness properties of the represented function to the outer function and can be well approximated by ReLU networks. It appears that instead of two hidden layers, a more natural interpretation of the Kolmogorov-Arnold representation is that of a deep neural network where most of the layers are required to approximate the interior function. Any efficient neural network construction mimicking the KA representation and approximating β-smooth functions up to error $\epsilon = m^{-\beta}$ should have at most of the order of $\epsilon^{-d} = m^{d}$ many network parameters. Thus this representation cannot avoid the curse of dimensionality

The following KA representation is much more explicit and practical.

Theorem 1 (Theorem 2.14 in [4]). Fix $d \geq 2$. There are real numbers a, b_p, c_q and a continuous and monotone function$\psi : \mathbb{R} \to \mathbb{R}$, such that for any continuous function $f : [0,1]^d \to \mathbb{R}$, there exists a continuous function $g : \mathbb{R} \to \mathbb{R}$ with

$$f\left(x_1, \ldots, x_d\right) = \sum_{q=0}^{2d} g\left(\sum_{p=1}^{d} b_p \psi\left(x_p + qa\right) + c_q\right).$$

This representation is based on translations of one inner function ψ and one outer function g. The inner function ψ is independent of f. The dependence on q in the first layer comes through the shifts. The right hand side can be realized by a neural network with two hidden layers. The first hidden layer has d units and activation function ψ and the second hidden layer consists of $2d + 1$ units with activation function g.

End Remarks

Approximation class

- Shallow and Deep Networks ✓
- Universality, degree of approximation, curse of dimensionality ✓
- Compositionally sparse functions
- Equivalence between compositional sparsity and efficient Turing computability
- Implications: deep sparse networks