Statistical Learning Theory

and

Applications

2025

Class 14

Sparse Compositionality:

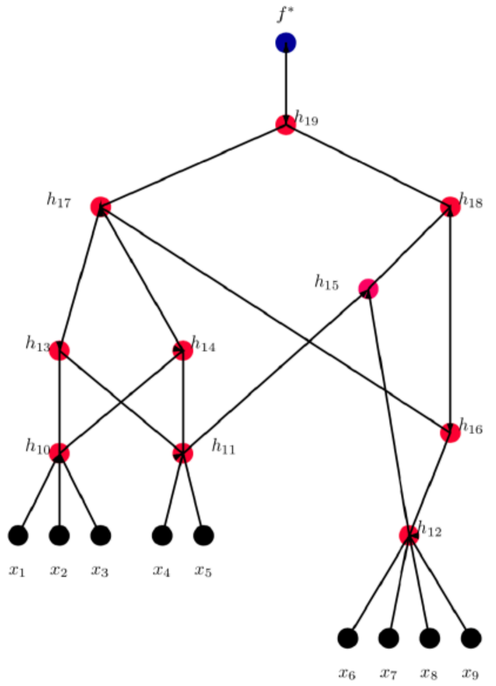
TOMASO POGGIO, MIT, CBMM

Plan

- Sparsity: motivations
- Compositionality
- Sparse compositionality implies 1) no curse of dimensionality 2) deep nets are a good H
- Compositionality implies good approximation + generalization+optimization
- Generalization: sparsity of weight matrices
- Optimization: sparse functions are easy to learn
- Which functions are compositional? Compositional structure of P
- Efficient Turing computability implies (sparse) compositionality
- Transformers and autoregressive
- Implications: lottery ticket and physics

Definition: Compositional Sparsity

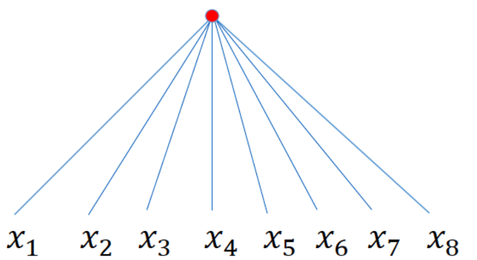
A function $f : \mathcal{X}^d \to \mathcal{X}$ is *compositionally sparse* if it can be represented as the composition of at most $O(poly(d))$ *constituent functions* each of which is *sparse*, i.e., depends on at most a constant (small) number of variables (so that it does not suffer the curse of dimensionality).

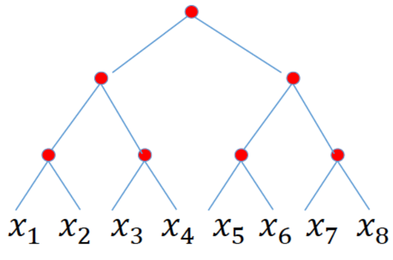


A compositionally sparse function can be represented in terms of a DAG (Directed Acyclic Graph), in which the leaves represent the inputs, the root denotes the output function, and the internal nodes represent the constituent functions.

What is the curse of dimensionality?

$$y = f(x_1, x_2, \cdots)$$





$$f(x_1, x_2, ..., x_8) = g_3(g_{21}(g_{11}(x_1, x_2), g_{12}(x_3, x_4))g_{22}(g_{11}(x_5, x_6), g_{12}(x_7, x_8)))$$

Theory says that both shallow and deep network can approximate a function of d variables equally well. It says that both should suffer from a curse of dimensionality: number of parameters in both cases depends exponentially as $N = O(\epsilon^{-\frac{d}{m}})$

As an example for a CIFAR image $N \approx (0.1^{-\frac{1000}{1}}) \approx 10^{1000}$

Mhaskar, Poggio, Liao, 2016

Some of the puzzles of Deep Learning

Why do kernel machines suffer the curse of dimensionality?

Can deep nets represent/approximate even high-dimensional functions?

How can deep nets escape the curse of dimensionality (in approximation)?

Why deep networks? Why is depth important?

Why are CNNs so much better than dense NN?

What controls generalization? Norm-like complexity, sparsity, rank?

Are all functions that can be computed by physical systems sparse?

Statistical Learning Theory

and

Applications

2025

Class 14

Sparse Compositionality:

TOMASO POGGIO, MIT, CBMM

Plan

- Sparsity: motivations
- Compositionality
- Sparse compositionality implies 1) no curse of dimensionality 2) deep nets are a good H
- Compositionality implies good approximation + generalization+optimization
- Generalization: sparsity of weight matrices
- Optimization: sparse functions are easy to learn
- Which functions are compositional? Compositional structure of P
- Efficient Turing computability implies (sparse) compositionality
- Transformers and autoregressive
- Implications: lottery ticket and physics

Compositionality in Computer Science and Language

Compositionality is a foundational principle in both computer science and natural language that enables scalability, generalization, and interpretability. At its core, compositionality means building *complex systems or meanings from simpler parts* in a structured way.

Compositionality in Language and CS

Semantics

- Principle of Compositionality (Frege's Principle): the meaning of a sentence is determined by the meanings of its parts and the rules used to combine them.
- Enables generalization to an infinite number of novel sentences, von Humboldt's *"infinite use of finite means"* (Humboldt, 1836; Chomsky, 1965)

Software Design

- Modularity: Programs are built from composable modules (functions, classes, objects)

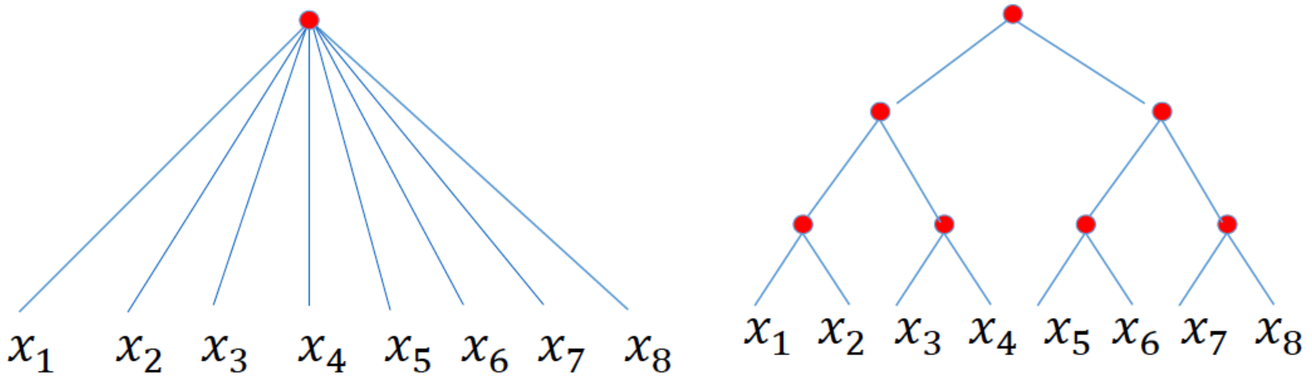
Plan

- Sparsity: motivations
- Compositionality
- Sparse compositionality implies 1) no curse of dimensionality 2) deep nets are a good H
- Compositionality implies good approximation + generalization+optimization
- Generalization: sparsity of weight matrices
- Optimization: sparse functions are easy to learn
- Which functions are compositional? Compositional structure of P
- Efficient Turing computability implies (sparse) compositionality
- Transformers and autoregressive
- Implications: lottery ticket and physics

Definitions

Definition

A sparse compositional function of d variables is a function that has a representation in terms of the composition of no more than $poly(d)$ sparse functions, each depending on a small number of variables $\leq d_0$ (such that each sparse function does not suffer from curse of dimensionality).

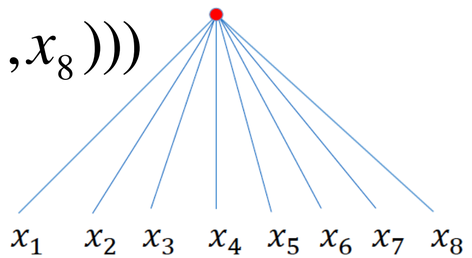


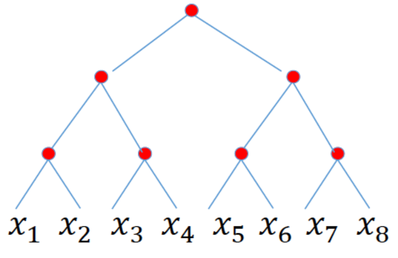
(Sparse) Compositionality

I use the term Sparse Compositionality to emphasize simplicity of constituent functions (compositionality is always trivially true because $f = I \circ f$). Notice that sparse compositionality is in general not unique: if $f = g_L \circ g_{L-1} \circ \ldots \circ g_1 = g_L \circ \overline{g}$ with $\overline{g} = g_{L-1} \circ \ldots \circ g_1$.

Deep NNs can avoid the curse for compositional functions

$$f(x_1,x_2,...,x_8) = g_3(g_{21}(g_{11}(x_1,x_2),g_{12}(x_3,x_4))g_{22}(g_{11}(x_5,x_6),g_{12}(x_7,x_8)))$$





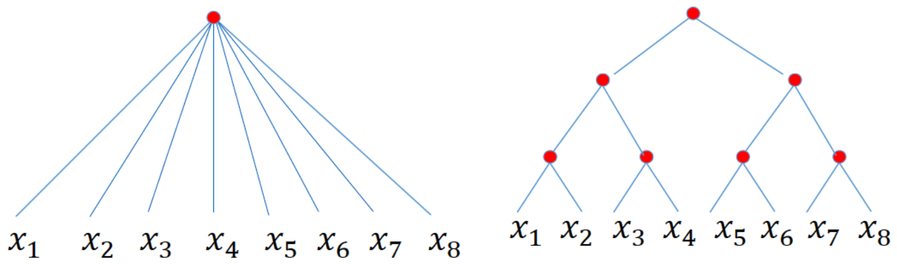
Theorem, Mhaskar, Poggio, Liao 2016 (informal statement)

Suppose that a target function of d variables is hierarchically locally compositional, that is has *compositional sparsity*. Both shallow and deep network can approximate f equally well. For a target function with the binary tree graph, the number of parameters of the shallow network depend exponentially on the dimension - that is $O(\epsilon^{-d})$ - whereas the number of parameters of the deep network (with binary tree architecture) is $O(d\epsilon^{-2})$.

H. N. Mhaskar and Poggio, T., "Deep vs. shallow networks: An approximation theory perspective", Analysis and Applications, vol. 14, no. 06, pp. 829 - 848, 2016.

Sobolev spaces and networks

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 $\|f\| + \sum_{1 \leq |\mathbf{k}|_1 \leq m} \|D^{\mathbf{k}} f\| \leq 1$, 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 $\mathbf{k}$.



We define $W_m^{d,2}$ to be the class of all compositional functions f of d variables with a binary tree architecture and constituent functions h in W_m^2. The corresponding class of deep networks $\mathscr{D}_{L,2}$ is the set of all networks of depth L with a binary tree architecture, where each of the constituent nodes is a shallow networks.

Proof outline

Proof outline We observe that each of the constituent functions being in W_m^2, implies that each of these functions can be approximated from $\mathcal{S}_{N,2}$ up to accuracy $\epsilon = cN^{-m/2}$. Our assumption that $f \in W_m^2$ implies that each of these constituent functions is Lipschitz continuous. Hence, it is easy to deduce that, for example, if P, P_1, P_2 are approximations to the constituent functions h, h_1, h_2, respectively within an accuracy of ϵ, then since $\|h - P\| \leq \epsilon$, $\left\| h_1 - P_1 \right\| \leq \epsilon$ and $\left\| h_2 - P_2 \right\| \leq \epsilon$, then

$$\left\| h\left(h_1, h_2\right) - P\left(P_1, P_2\right) \right\| = \|h\left(h_1, h_2\right) - h\left(P_1, P_2\right) +$$

$$\|h\left(P_1, P_2\right) - P\left(P_1, P_2\right)\| \leq \|h\left(h_1, h_2\right) - h\left(P_1, P_2\right)\| + \left\| h\left(P_1, P_2\right) - P\left(P_1, P_2\right) \right\| \leq \overline{c\epsilon}$$ by Minkowski inequality. Thus

$$\left\| h\left(h_1, h_2\right) - P\left(P_1, P_2\right) \right\| \leq c\epsilon,$$

for some constant $c > 0$ independent of the functions involved.

Lipshitz property of h

Remark

New Proof. Linear combinations of 6 units *provides an indicator function; k partitions for each coordinates require 6 k n units in one layer. The next layer computes the entries in the 2D table corresponding to ; they also correspond to tensor products. Two layers with 6kn + (6kn)^2 units represent one of the g functions. For convolutional nets total units is (l (6kn + (6kn)^2))*

Mhaskar, Poggio, Liao, 2016

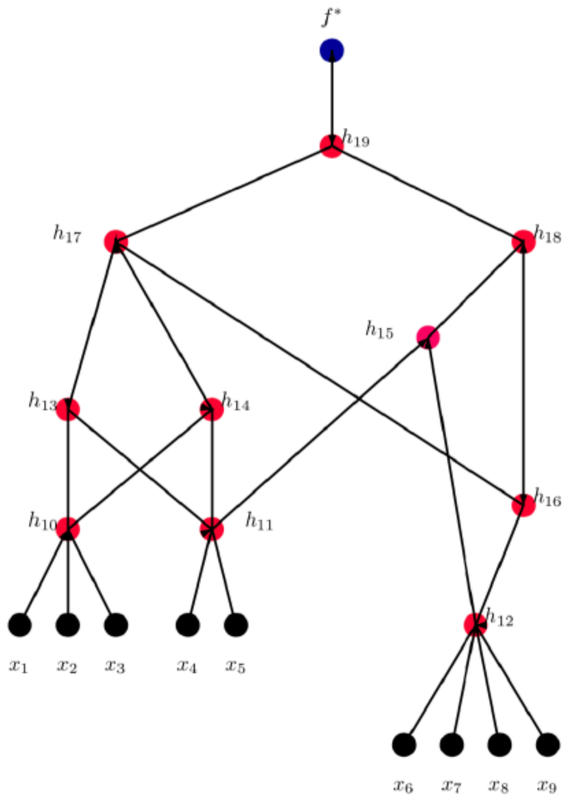
More general result

Theorem. Let $\mathcal{G}$ be a DAG, n be the number of source nodes, and for each $v \in V$, let d_v be the number of in-edges of v. Let $f : \mathbb{R}^d \mapsto \mathbb{R}$ be a compositional $\mathcal{G}$-function, where each of the constitutent function is in $W_{m_v}^{d_v}$. Consider shallow and deep networks with infinitely smooth activation function as in Theorem 1. Then deep networks - with an associated graph that corresponds to the graph of f — avoid the curse of dimensionality in approximating f for increasing d, whereas shallow networks cannot directly avoid the curse. In particular, the complexity of the best approximating shallow network is exponential in d

$$N_s = \mathcal{O}\left(\epsilon^{-\frac{d}{m}}\right),$$

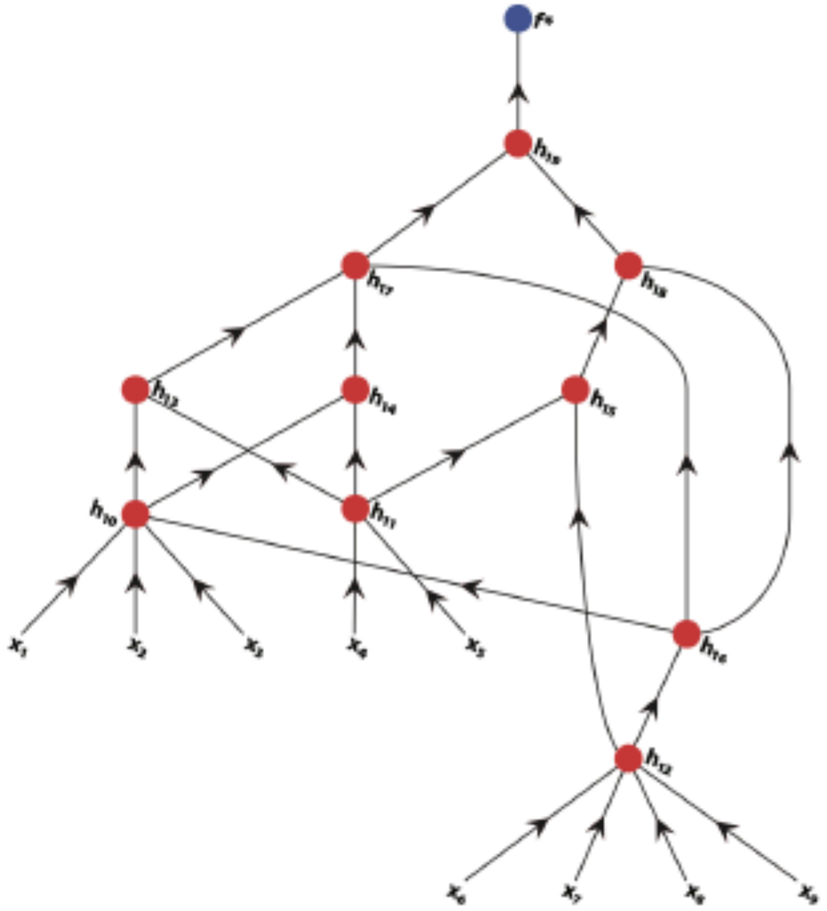
where $m = \min_{v \in V} m_v$, while the complexity of the deep network is

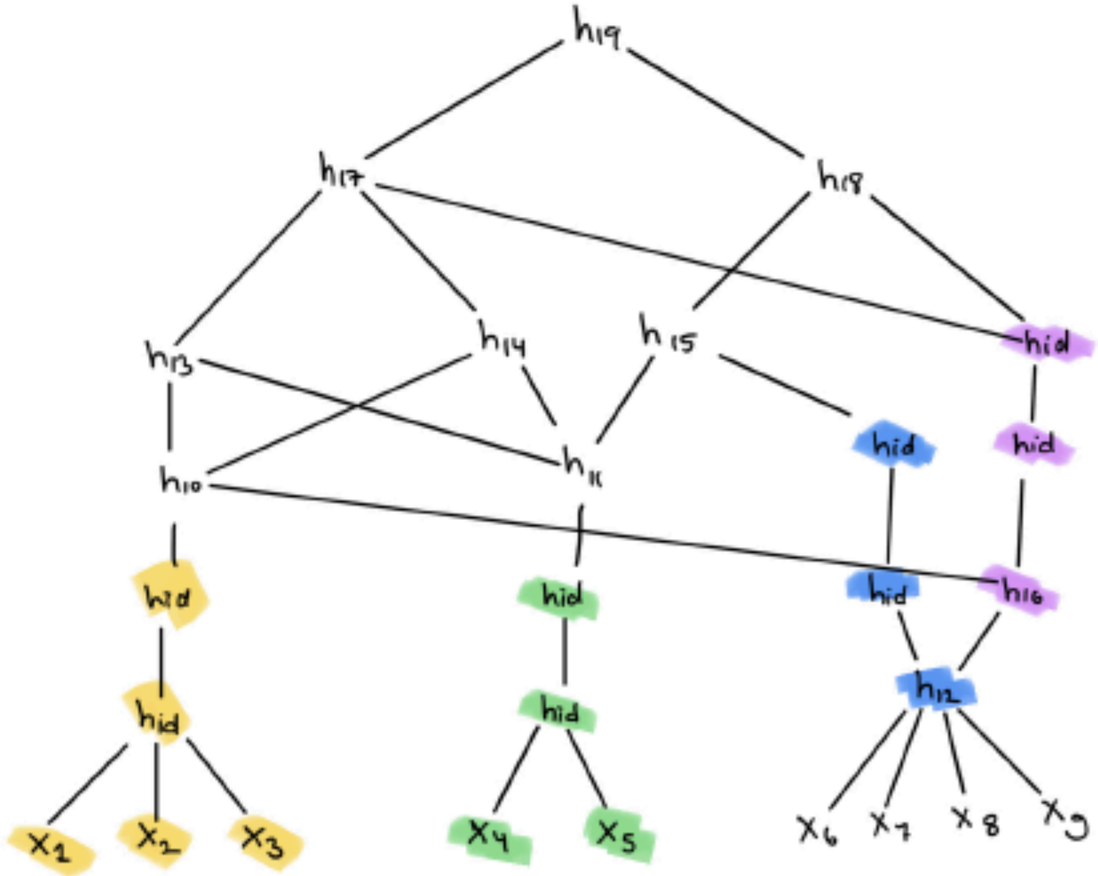
$$N_d = \mathcal{O}\left(\sum_{d \in V} \epsilon^{-d_v/m_v}\right).$$



Mhaskar, Poggio, Liao, International Journal of Automation and Computing 2016; PNAS, 2020

Remark: a graph can be rewritten as a graph with nodes at separate discrete levels





Compositional sparsity is becoming an assumption in recent approximation theory (implying belief that you need compositionality)

Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review

Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda & Qianli Liao

International Journal of Automation and Computing (2017)

Nonparametric regression using deep neural networks with ReLU activation function
Johannes Schmidt-Hieber,
The Annals of Statistics, (2020)

On the rate of convergence of fully connected deep neural network regression estimates
Kohler, M. und *Langer, S.*
Annals of Statistics (2021)

Compositional Sparsity, Approximation Classes, and Parametric Transport Equations*

Dedicated to Ronald DeVore on the occasion of his 80th birthday

Wolfgang Dahmen†

July 14, 2022

ArXiv 2022

SGD learning on neural networks: leap complexity and saddle-to-saddle dynamics

Emmanuel Abbe*, Enric Boix-Adserà†, Theodor Misiakiewicz‡

September 4, 2023

Approximation by tree tensor networks in high dimensions: Sobolev and compositional functions*

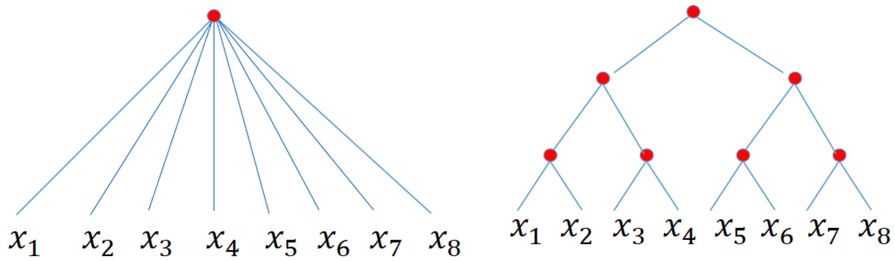
M. Bachmayr†, A. Nouy‡, R. Schneider§

Dedicated to Ronald DeVore on the occasion of his 80th birthday

ArXiv 2022

Remark: Kernel method *cannot* exploit compositional sparsity

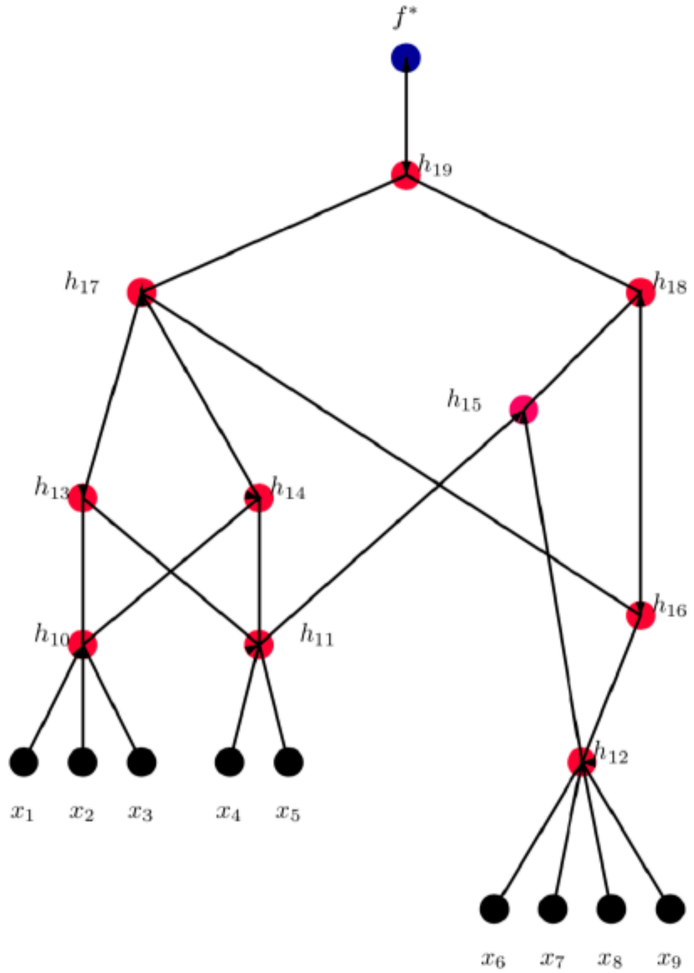
For kernel methods the ambient dimension D (8 in the figure) controls the sample complexity irrespectively of any compositionally sparse structure of f



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

Note: HyperBF can exploit sparsity $f(x) = \sum_i c_i K(\|x - x_i\|_M)$

Remarks



In general, it can be quite hard to decide for a given function whether the function is compositional with respect to a given DAG. Furthermore, one fixed compositional function can be compositional with respect to many different DAGs. Finally, even if the DAG is fixed, there can be many different choices of the constituent functions that give rise to the same function. *Sparse compositionality is very non-unique.*

Figure 4: An example of a $\mathcal{G}$-function (f^* given in (4.2)). The vertices of the DAG $\mathcal{G}$ are denoted by red dots. The black dots represent the input to the various nodes as indicated by the in–edges of the red nodes, and the blue dot indicates the output value of the $\mathcal{G}$-function, f^* in this example.

Some of the puzzles of ML answered by this result

Why do kernel machines suffer the curse of dimensionality?

Can deep nets represent/approximate every function even if high-dimensional?

How can deep nets escape the curse of dimensionality (in approximation)?

Why deep networks? Why is depth important?

Why are CNNs so much better than dense NN?

What controls generalization? Norm-like complexity, sparsity, rank?

Are all functions that can be computed by physical systems compositionally sparse? Can they be simulated by a Turing machine?

$f : \mathbb{R}^d \to \mathbb{R}^q$

Our results do not change

Intuition : monomials (units) are reused for different outputs

Lemma

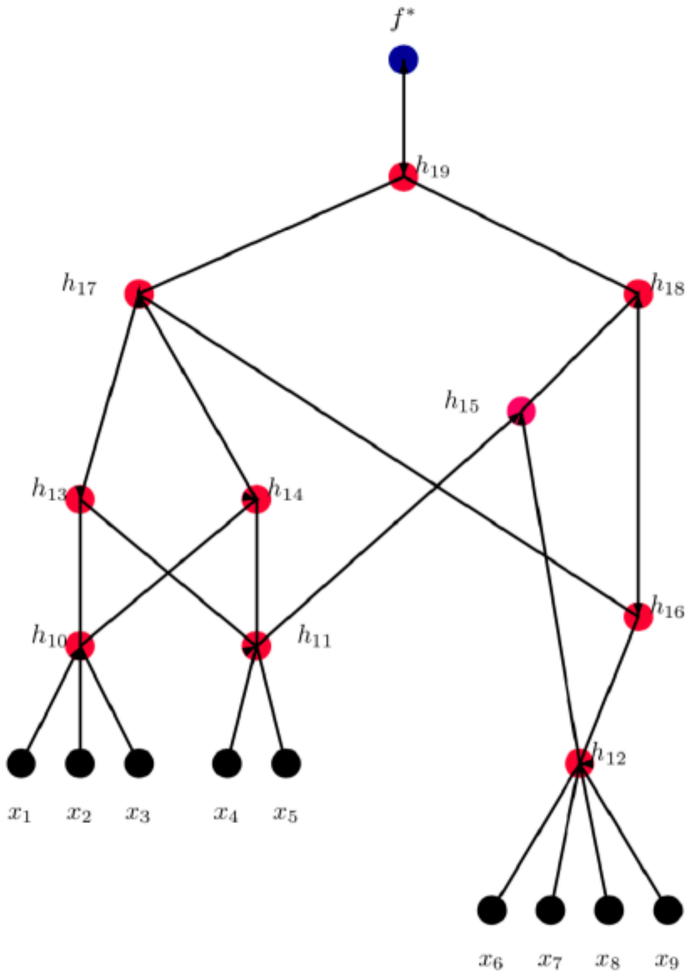
For $f : \mathbb{R}^d \to \mathbb{R}^q$, $f_i \in W_m^d$

$i = 1, \dots, q$ the # hidden units N for ε accuracy is $N \sim O\left(\varepsilon^{-d/m}\right)$

Plan

- Sparsity: motivations
- Compositionality
- Sparse compositionality implies 1) no curse of dimensionality 2) deep nets are a good H
- Compositionality implies good approximation + generalization+optimization
- Generalization: sparsity of weight matrices
- Optimization: sparse functions are easy to learn
- Which functions are compositional? Compositional structure of P
- Efficient Turing computability implies (sparse) compositionality
- Transformers and autoregressive
- Implications: lottery ticket and physics

Compositionality ==> approximation+generalization+optimization



The key for all of this is the “small” number of effective parameters that are required compared with exponential number of parameters required otherwise.

Approximation and Generalization tradeoff

For an increasing value of ρ, the approximation error decreases. On the other hand, the generalization error decreases with decreasing ρ (product of norms of weight matrices).

So, one tries to find an optimal tradeoff between these two errors, considering factors like dimensionality and smoothness of the target function. It's a delicate balancing act between fitting the model well and ensuring it generalizes effectively.

Why is sparsity key for both approximation and generalization?

Why sparsity appears as a key in both *approximation* and *generalization*? For one simple reason: compositional sparsity of the functions implies that sparse networks can approximate well with low number of bits (and small norm of them), which implies good generalization. In other words: efficient computability implies *structural sparsity* of deep networks which implies a good tradeoff between approximation and generalization.

The *curse of dimensionality and* 3 blessings of compositionality

- low dim of constituent functions h (blessing of low dim)
- high smoothness of some of the h (blessing of smoothness)
- sharing across tasks of K in F= H K for better generalization

This latter condition is required in some definitions of compositionality but not in ours!

Another caveat: there are different definitions of compositionality!

Plan

- Sparsity: motivations
- Compositionality
- Sparse compositionality implies 1) no curse of dimensionality 2) deep nets are a good H
- Compositionality implies good approximation + generalization+optimization
- Generalization: sparsity of weight matrices
- Optimization: sparse functions are easy to learn
- Which functions are compositional? Compositional structure of P
- Efficient Turing computability implies (sparse) compositionality
- Transformers and autoregressive
- Implications: lottery ticket and physics

Novel bounds for Sparse Networks

Theorem (for convolutional nets)

Let G be a neural network architecture of depth L and let $\rho > 0$. Let $X = \{x_i\}_{i=1}^m$ be a set of samples. Then,

$$\mathscr{R}_S(\mathscr{F}_{G,\rho}) \leq \frac{\rho}{m}\left(1+\sqrt{2L\log(2\deg(G))}\right)\cdot\sqrt{\prod_{l=1}^{L} k_l \cdot \max_{j\in[d_0]}\sum_{i=1}^{m}\|z_j^0(x_i)\|^2},$$

where k_l denotes the kernel size in the l 'th layer . A similar result holds for non-convolutional nets with similar sparsity per layer.

Galanti, Xu, Poggio, Neurolps 2023

Novel bounds for Sparse Networks (Galanti, Poggio, Xu, 2023)

Theorem (informal)

Consider a deep RELU network with a $n \times n$ *dense weight matrix* W *at a certain layer. Its contribution to the Rademacher complexity of the network is* $\|Wx\| = \sqrt{\|Wx\|^2} = \sqrt{(\sum_k^n \|w_k\|^2)\|x\|^2} \leq \|W\|\|x\|$

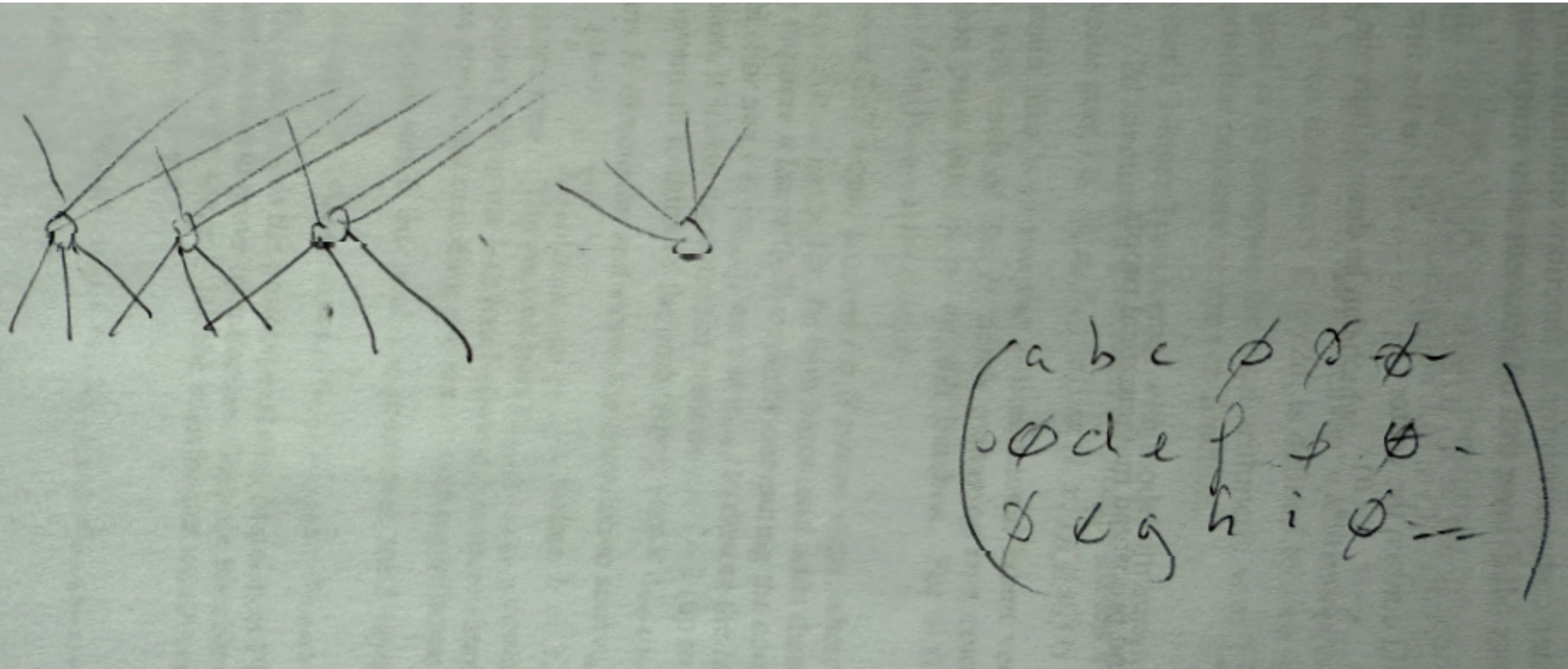
Suppose now that W *is sparse, that is row* k *contains only* q *non-zeros (*$q << n$*) components (assume of equal size w). Then each row contributes* $\|w\|^2(\frac{q}{n}\|x\|^2)$ *to the sum in* $\sqrt{(\sum_k^n \|w_k\|^2)\|x\|^2}$ *implying that the contribution of* Wx *to the Rademacher complexity of the network is*

$$\sqrt{(\sum_q^n \|w\|^2)\frac{q}{n}\|x\|^2}$$

The case of non-overlapping convolution gives $\sqrt{(\|w\|^2)\|x\|^2} = \|w\|\|x\|$.

Galanti, Xu, Poggio, Neurolps 2023

Sparse Networks



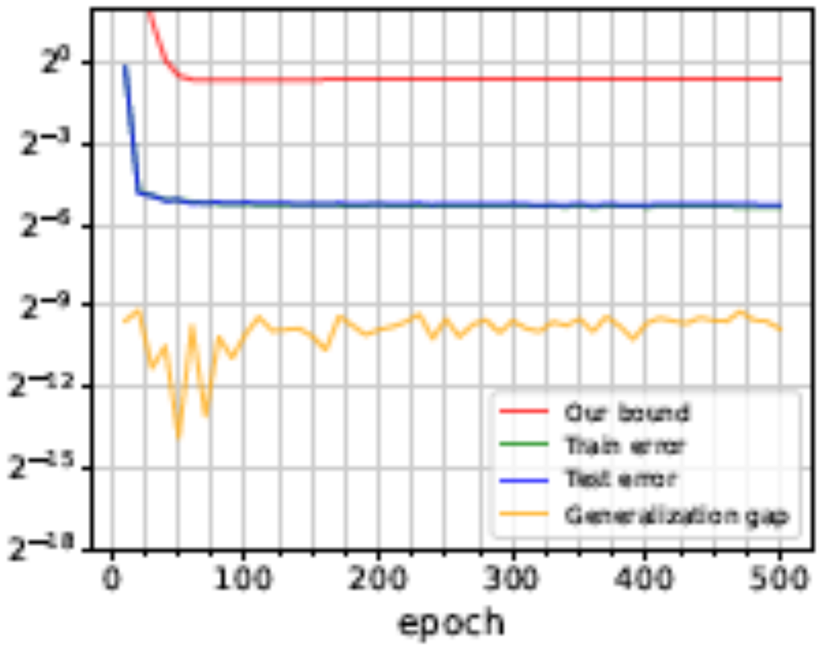
Sparsity in the weight matrices

- Sparsity of the DAG implies sparsity in the weight matrices which implies generalization bounds that are order of magnitude better than for deep networks with dense matrices (new theorem)

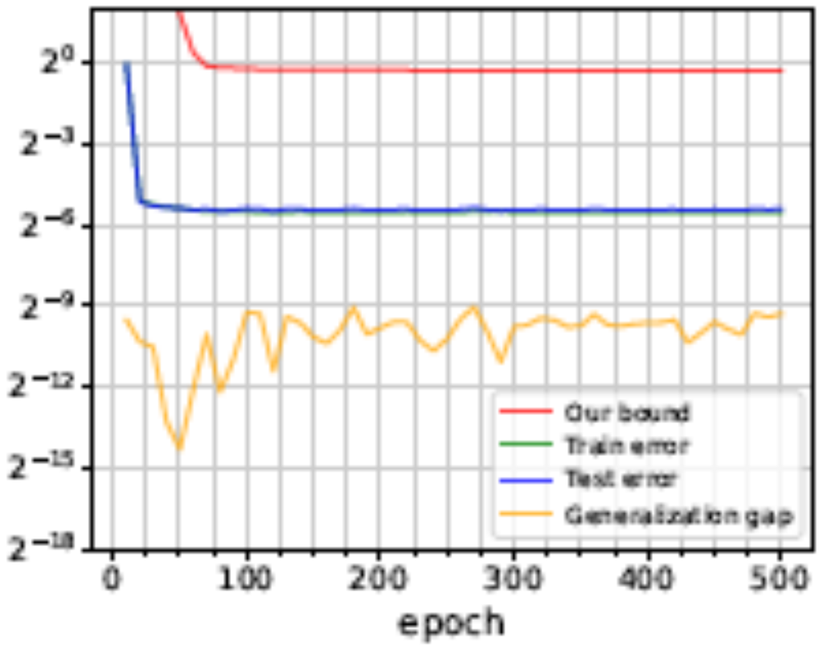
- Weight sharing does not improve bounds…so it is not convolution in CNNs that leads to better generalization but the fact that the weight matrices are sparse

Galanti, Xu, Poggio, Neurolps 2023

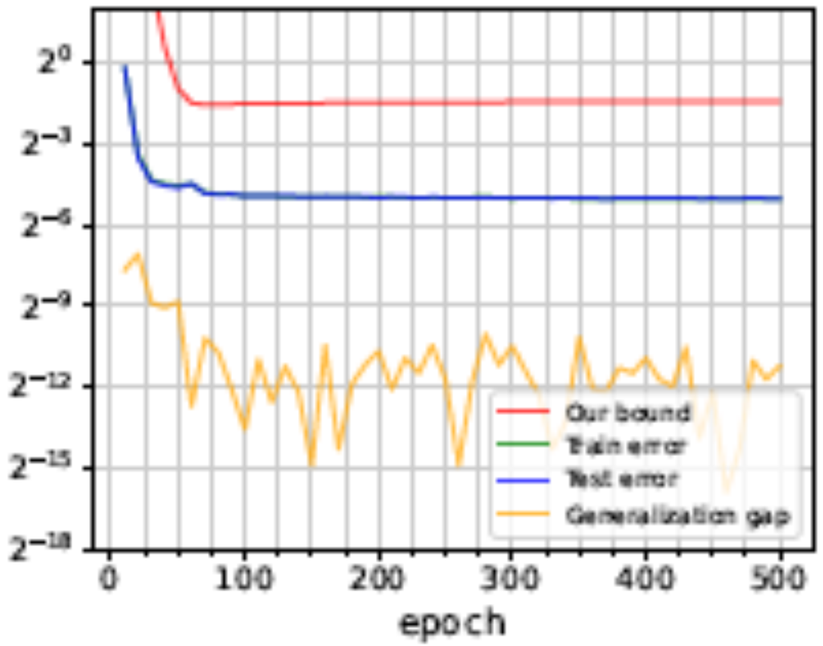
MNIST sparse network: non-vacuous bound



CONV-4-50



CONV-4-500



CONV-3-100

Galanti, Xu, Poggio, Neurolps 2023

Plan

- Sparsity: motivations
- Compositionality
- Sparse compositionality implies 1) no curse of dimensionality 2) deep nets are a good H
- Compositionality implies good approximation + generalization+optimization
- Generalization: sparsity of weight matrices
- Optimization: sparse functions are easy to learn
- Compositional structure of P
- Which functions are compositional? Efficient Turing computability implies (sparse) compositionality
- Transformers and autoregressive
- Implications: lottery ticket and physics

P

The (Boolean) class $\mathbf{P}$ consists of all Boolean functions *that are computable by a deterministic Turing machine in polynomial time.* Formally:

$$\mathbf{P} = \bigcup_{k \in \mathbb{N}} \mathrm{DTIME}\left(n^k\right)$$

The (Boolean) class $\mathbf{P}$ is closed under composition. $\mathbf{P}$ consists of Boolean functions with a Fourier representatpolynomials that are efficiently computable, that is contains a poly- number of monomials.

Plan

- Sparsity: motivations
- Compositionality
- Sparse compositionality implies 1) no curse of dimensionality 2) deep nets are a good H
- Compositionality implies good approximation + generalization+optimization
- Generalization: sparsity of weight matrices
- Optimization: sparse functions are easy to learn
- Compositional structure of P
- Which functions are compositional? Efficient Turing computability implies (sparse) compositionality
- Transformers and autoregressive
- Implications: lottery ticket and physics

Computable functions

Classical

f : $N^k \rightarrow N$ is **computable** if there is an effective procedure or algorithm that correctly calculates f.

Also

A function $f : N^d \rightarrow N$ is **efficiently computable** if it can be computed/stored in time/memory that is at most polynomial in effective dimensionality $(\frac{d}{s})$.

with Seth Lloyd

Definitions

Definition

A sparse compositional function of d variables is a function that has a representation in terms of the composition of no more than $poly(d)$ sparse functions, each depending on a small number of variables $\leq d_0$.

Definition

A real-valued continuous function $f : \mathbf{R}^d \rightarrow \mathbf{R}^k$ is Turing computable if there is a procedure or algorithm that correctly calculates a sequence of approximations f_n to f (n = number of parameters) with decreasing errors $\leq \epsilon_n$ in the sup norm; it is efficiently computable if it is computable in polynomial time/space in d.

Goal: show that *deep+sparse* NNs are universal, efficient approximators

Theorem For every function which is efficiently Turing computable there exist a *sparse, deep* network that can approximate it without curse of dimensionality; furthermore the function has — in general several —representations that are compositionally sparse.

Poggio, Fraser , Bulletin of the AMS, 2024

Definition: Efficient Turing computability

Let $f : [0,1]^d \to \mathbb{R}$. We say that f is Turing-computable to precision 2^{-k} if:

There exists a deterministic Turing machine M such that, given:

- "A point $x \in [0,1]^d$, where each coordinate is represented by a binary string of m bits,
- "A precision parameter $\varepsilon = 2^{-k}$, with $k \in \mathbb{N}$,

the machine M halts in time polynomial in m and k, and outputs a rational number y such that: $|f(x) - y| \leq \varepsilon$. Moreover:

- The machine M can be encoded by a Boolean function $F : \{0,1\}^{d \cdot m + \log k} \to \{0,1\}^t$ where the input encodes the bits of x and ε, and the output is a binary encoding of y.
- This Boolean function is $F \in \mathbf{P}$, i.e., it is computable in deterministic polynomial time.

Thus, f is approximable to arbitrary precision by a Boolean function in class $\mathbf{P}$ operating on bitstring representations of its inputs.

Problem Setting

- Domain: a real-valued (or vector-valued) function where d is the effective dimension

$$f : [0,1]^d \to \mathbb{R}^m$$

- Inputs are finite-precision encodings: an input x is given by an n -bit encoding.
- Approximation target: for any accuracy $\varepsilon > 0$, build a network Φ_ε with size/depth poly $(n + \log(1/\varepsilon))$ such that

$$\max_{x \in [0,1]^d} \left\| \Phi_\varepsilon(x) - f(x) \right\| \leq \varepsilon .$$

x has an n-bit encoding

(Norm can be ℓ_∞ on $\mathbb{R}^m$.)

Why Boolean functions

Every (efficiently computable) function is represented in a computer in terms of its Boolean approximation because a Turing machine (a program) can be represented as the composition of threshold functions.

Compositional Sparsity of $\mathbf{P}$ (theorem later)

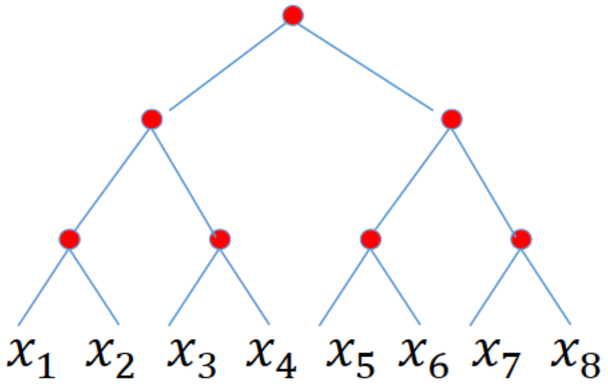
All functions in $\mathbf{P}$ are compositionally sparse, some are sparse. They can be represented as DAGs of small constituent functions:
- Low arity (e.g., 2-3 variables per function),
- Constant or logarithmic depth,
- Polynomial total size.

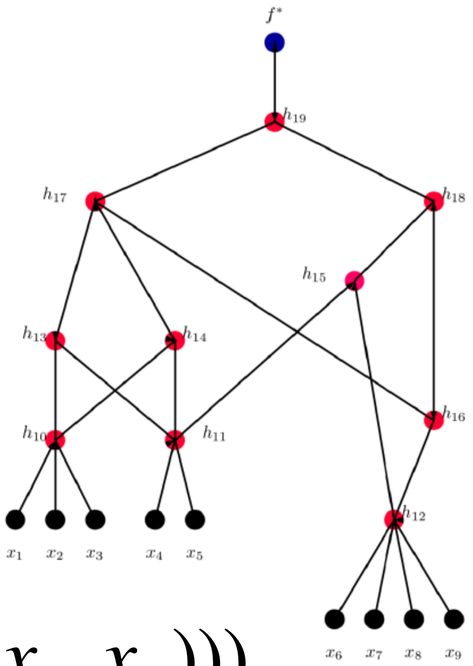
This is formalized in results such as:
- Poggio-Fraser Theorem (every P-time Boolean function admits a compositional sparse representation),
- Barrington's Theorem and Valiant's results on succinct representations are related, earlier results

Poggio Fraser, AMS, 2024

Theorem: efficient computability implies compositional sparsity and the latter implies existence of learnable representation

Functions that are *efficiently Turing computable* are 1) *compositionally sparse*, in the sense of a sparse DAG. Thus P = compositionally sparse Boolean functions. Furthermore, 2) for each sparse compositional function, *there exist a deep network that approximates it arbitrarily well* without curse of dimensionality (Poggio, Fraser, Bulletin of the AMS, 2024; Poggio, et al., Int. J. Computing, 2017))





$$f(x_1,x_2,...,x_8) = g_3(g_{21}(g_{11}(x_1,x_2),g_{12}(x_3,x_4))g_{22}(g_{11}(x_5,x_6),g_{12}(x_7,x_8)))$$

Theorem: efficient computability implies compositional sparsity and the latter implies existence of learnable representation

---

Functions that are *efficiently Turing computable* are

1. *compositionally sparse*, in the sense of a sparse DAG. In particular, P = compositionally sparse Boolean functions;
2. admit *a deep network that approximates it arbitrarily well* without curse of dimensionality

Poggio, Fraser, Bulletin of the AMS, 2024

Problem Setting

Target function:

$$f : [0,1]^d \to \mathbb{R}^m$$

Inputs given as n-bit encodings.

Approximation goal:

$$\max_{x \in [0,1]^d} \|\Phi_\varepsilon(x) - f(x)\| \leq \varepsilon$$

Efficient Computability

Definition. A function f is *efficiently computable* if there exists a Turing machine M that, given (x, ε) encoded with $n + \log(1/\varepsilon)$ bits, computes an ε-approximation of $f(x)$ in time $\mathrm{poly}(n, \log(1/\varepsilon))$.

Main Theorem

Theorem. If f is efficiently computable, then for every $\varepsilon > 0$ there exists a network Φ_ε with

$$\text{size, depth} = \text{poly}(n + \log(1/\varepsilon))$$

such that

$$\max_{x \in [0,1]^d} \|\Phi_\varepsilon(x) - f(x)\| \leq \varepsilon.$$

Proof Sketch

1. Efficient TM $\Rightarrow$ bounded-fan-in Boolean circuits of poly size.
2. Each gate realized by a constant-depth subnetwork.
3. Wiring gives a deep Φ_ε.
4. On all n-bit inputs:

$$\|\Phi_\varepsilon(x) - f(x)\| \leq \varepsilon.$$

5. Complexity: size, depth $= \mathrm{poly}(n + \log(1/\varepsilon))$.

Corollary (Compositional Sparsity)

Corollary. If f is efficiently computable, then for any $\varepsilon > 0$ there exists a compositionally sparse g_ε such that

$$\max_{x \in [0,1]^d} \|g_\varepsilon(x) - f(x)\| \leq \varepsilon,$$

with each constituent subfunction depending on at most c variables and realized by a shallow network.

Implications

- Approximation complexity depends on $n + \log(1/\varepsilon)$, not on d.
- Deep networks avoid the curse of dimensionality.
- Structure is compositionally sparse (bounded local arity).

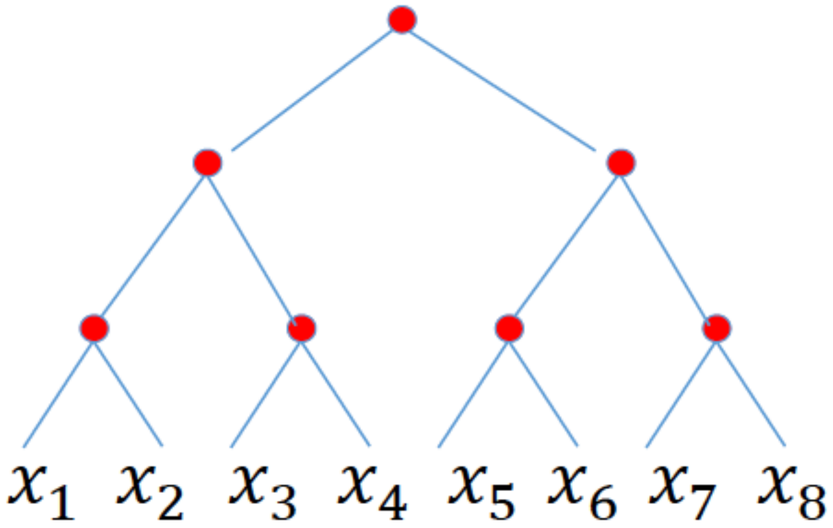
Subtleties

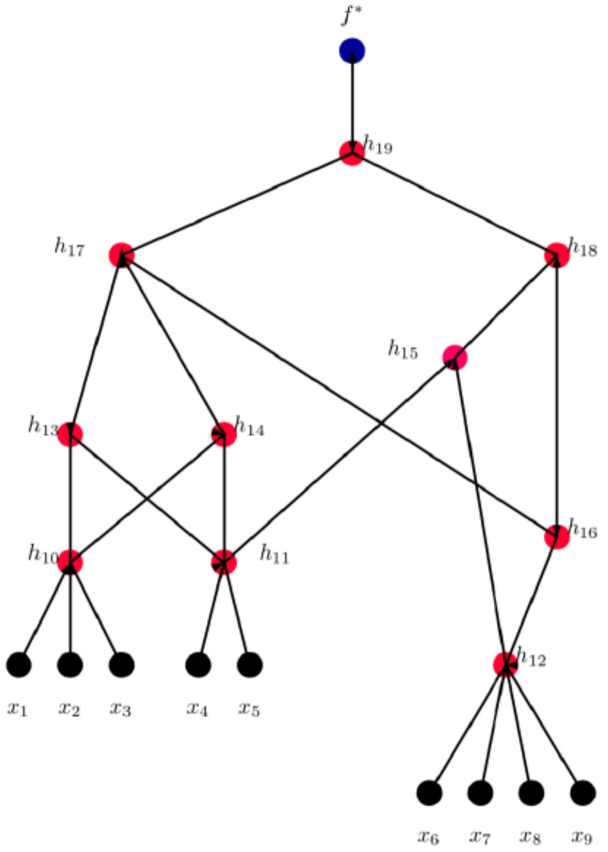
- Boolean compositional sparsity does not imply the original continuous f is sparse.
- For each ε, we approximate by a sparse g_ε.
- Discrete-to-continuous identification is approximate.

Take-Home Messages

- Efficiently computable $\Rightarrow$ deep approximants of polynomial complexity.
- These approximants can be compositionally sparse.
- Deep sparse networks explain approximation without curse of dimensionality.

Theorem: efficient computability implies compositional sparsity





$$f(x_1,x_2,...,x_8) = g_3(g_{21}(g_{11}(x_1,x_2),g_{12}(x_3,x_4))g_{22}(g_{11}(x_5,x_6),g_{12}(x_7,x_8)))$$

Optimization (learnability) of Boolean functions

A compositionally sparse Boolean function is *in general* difficult to learn because learnability of P would violate cryptographic assumptions; but…the constituent functions — since they are *sparse* — are *learnable.* In fact, very sparse Boolean functions are usually learnable, especially if they have a special structure — as staircase functions (Bregler et al.) or decision trees (Mansour) or low degree Boolean function (Mansour).

Compositional sparsity <==> efficient computability

- Foundation of learning: general form of networks for learning
- Relevant as foundation/justification for tensor representations (since they aim to approximate generic functions)

Plan

- Sparsity: motivations
- Compositionality
- Sparse compositionality implies 1) no curse of dimensionality 2) deep nets are a good H
- Compositionality implies good approximation + generalization+optimization
- Generalization: sparsity of weight matrices
- Optimization: sparse functions are easy to learn
- Which functions are compositional? Compositional structure of P
- Efficient Turing computability implies (sparse) compositionality
- Transformers and autoregressive
- Implications: lottery ticket and physics

The power of next word prediction (and transformers)

From our theorems it follows *that every Turing computable function is learnable if* training examples for each of the constituent functions are available. In other words *our results prove*, as a special case, the following

Theorem 1 (informal, Malach, 2023) *For any function f that can be efficiently computed using a Turing machine, there exists a dataset D such that training a (linear) auto-regressive next-token predictor on D results in a predictor that approximates f.*

This means that any computer program or intelligent agent that can be simulated by a computer, can be learned, given the right training set, by a simple next-token predictor.

Auto-Regressive Next-Token Predictors are Universal Learners, 2023 E. Malach

Several Implications

- Depth is key to avoid curse of dimensionality. Kernel machines cannot avoid it.

- The result also implies that autoregressive prediction and learning can be much easier than end-to-end prediction. This is the main "magic" of transformers.

Implications for optimization: end-to-end learning vs autoregressive framework

Suppose $f = g_1 \circ g_2 \cdots \circ g_d$

Two learning frameworks:

1.I have training sets for each g_i, that is x_i, y_i, where $g_i(x_i) = y_i$ (autoregressive framework)

2. I have a training set for f that consists of (x_i, y_i) where $f(x_i) = y_i$ (standard supervised framework/end-to-end)

<u>"Folk" Corollary</u> *Learnability in 2. cannot be guaranteed and is, in fact, impossible in general (crypto assumptions); 1. is usually learnable for sparse* g_i.

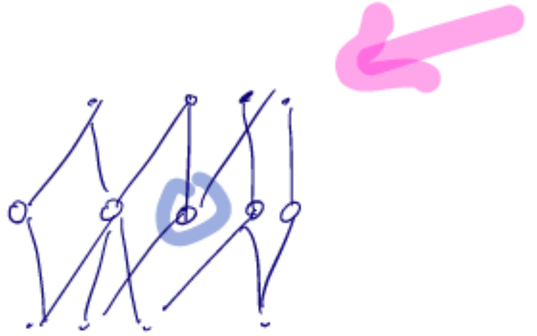
Multi outputs

Our basic result was

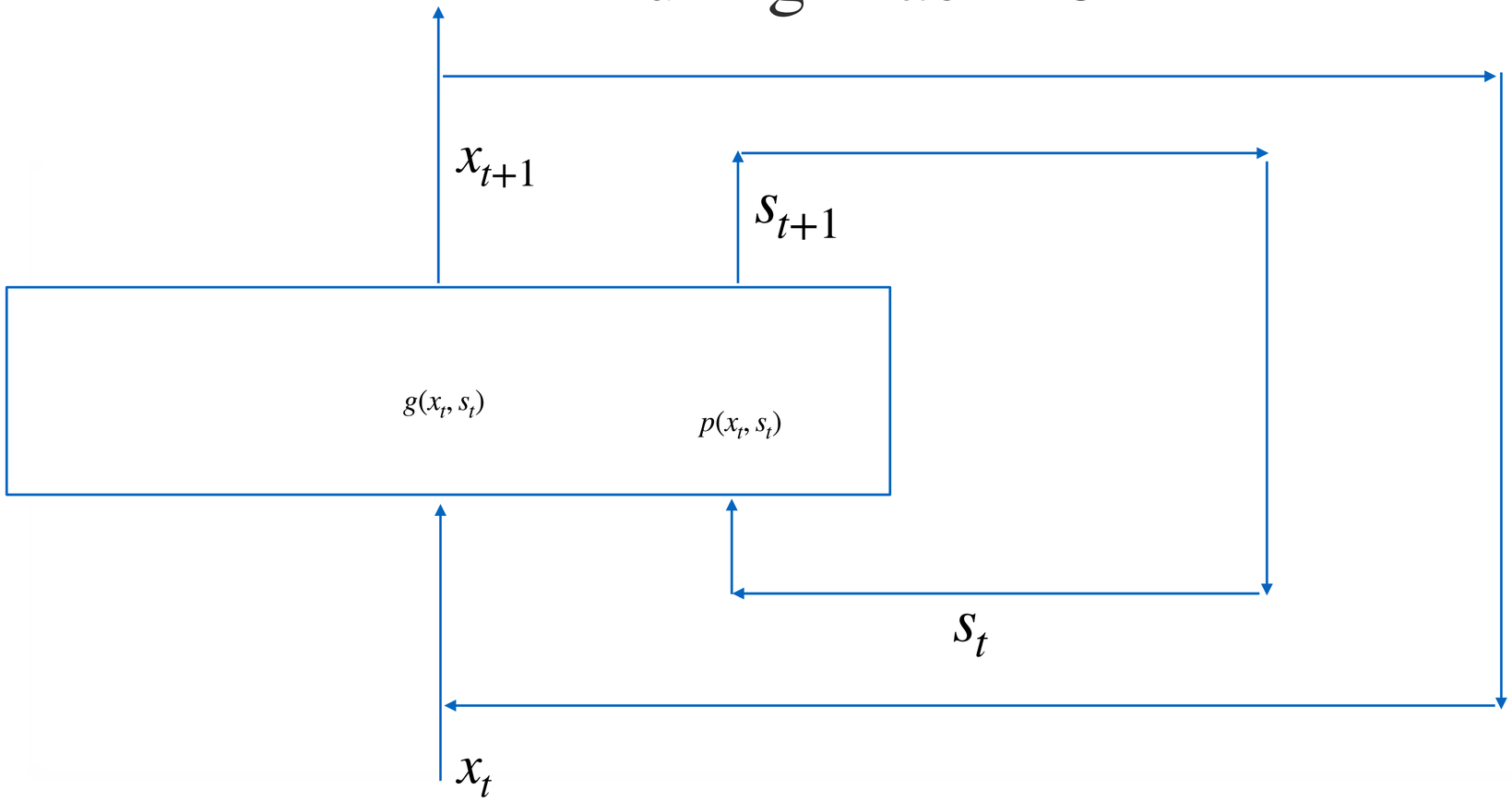
for $f : \mathbb{R}^d \longrightarrow \mathbb{R}$

using

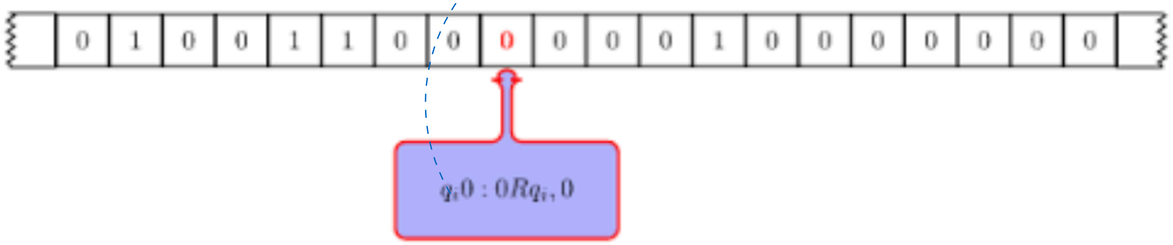
In reality many channels in each layer



Turing Machine



$$x_{t+1} = g(x_t, s_t) \quad s_{t+1} = p(s_t, x_t)$$



Turing Machine

A Turing machine as a deterministic dynamical system

At each discrete time step, the machine updates its configuration consisting of a finite control state and the contents of the tape (including the head position).

At time step t, the configuration of the machine is given by

$$C_t = (s_t, x_t),$$

where $s_t \in S$ is the finite internal state, $x_t$encodes the entire tape contents together with the current head position.

The transition function of the Turing machine induces a global deterministic update:

$$C_{t+1} = \Phi(C_t)\,.$$

Thus, the evolution of the machine can be described as an iterated map on the space of configurations.

Summary Table

Property	Without Compositionality	With Compositionality
Hypothesis complexity	Exponential	Polynomial (modular reuse)
Generalization bound	Loose	Tight if components are simple
Optimization	Often intractable	Tractable with known modules
Transfer	Poor	Strong module reuse

Curriculum Learning

- Sequentially train on subfunctions: D_1, ..., D_L
- Like language acquisition in humans
- Bottom-up learning: later stages depend on earlier ones

Conjecture on compositional learning

Pretrained LLMs learn compositional functions but usually without *reusing components*

Plan

- Sparsity: motivations
- Compositionality
- Sparse compositionality implies 1) no curse of dimensionality 2) deep nets are a good H
- Compositionality implies good approximation + generalization+optimization
- Generalization: sparsity of weight matrices
- Optimization: sparse functions are easy to learn
- Which functions are compositional? Compositional structure of P
- Efficient Turing computability implies (sparse) compositionality
- Transformers and autoregressive
- Implications: lottery ticket and physics

Compression, lottery ticket and sparsity

Theorem (Approximate Lottery Ticket Hypothesis for Compositionally Sparse Targets)

Let $f^\star \in \mathscr{C}_{\text{comp}}(s, k, L, r)$ be a *compositionally sparse* function, computable by a bounded-fan-in DAG of size s and depth L. Consider an overparameterized architecture $\mathscr{A}_{\text{dense}}$ with layer weights $\{W_\ell\}_{\ell=1}^{L}$, trained to global convergence on data drawn from $\mathscr{D}$ under a loss $L(\theta) = \mathbb{E}[\ell(F_\theta(X), Y)]$, with a regularizer favoring minimal $\rho(\theta)$. Assume that training converges to a minimum $\hat{\theta}$ with smallest value of $\mathscr{L}(\theta)$ among all global minimizers.

Compression, lottery ticket and sparsity

Then there exists a structured subnetwork $\Theta_{S,r} \subset \mathscr{A}_{\text{dense}}$ with $|S| = \tilde{O}(s)$ active parameters and per-layer ranks $r_\ell = \tilde{O}(1)$ such that:

- Approximation: there exists $\$\theta^\star \in \Theta_{S,r}\$$ satisfying $\|F_{\theta^\star} - f^\star\|_{L^2(\mathscr{D}_X)} \leq \varepsilon$..

- Proximity of minima: the dense minimum $\hat{\theta}$ and its projection $\theta_0 = \Pi_{S,r}(\hat{\theta})$ yield almost identical functions,

- $R(\theta_0) \underset{L}{\leq} R(\hat{\theta}) + O(\varepsilon + \delta)$, where δ is the aggregate spectral/pruning error

- $\delta \sim \sum_{\ell=1} \left(\sigma_{r_\ell+1}(W_\ell) + \|E_\ell^{\text{prune}}\|\right)$.

Moreover, fine-tuning makes θ_0 converge to a subnetwork $\tilde{\theta}$ satisfying $R(\tilde{\theta}) \leq R(\hat{\theta}) + O(\varepsilon + \delta)$. Hence, a *warm-start winning ticket* exists: a sparse, low-rank subnetwork realizing essentially the same minimum-risk solution as the dense model, and minimizing the product of Frobenius norms across layers.

Compression, lottery ticket and sparsity

Remark

Under minimal-norm convergence, dense optimization implicitly identifies a low-complexity region of parameter space containing a compositionally sparse subnetwork $\Theta_{S,r}$ of size $\tilde{O}(s)$.
Projecting onto this manifold and retraining within it recovers the dense model's performance up to $O(\varepsilon + \delta)$. This provides a succinct, regularized form of the *approximate lottery ticket hypothesis*: the effective solution is already sparse and compositional, even if the training representation is dense.
This shows that compositional sparsity and convergence imply large compression.

Sparsity: empirical evidence

Activation sparsity:

For a given token x_t the post-activation vector $h_t = \phi(Wx_t)$ in MLP and attention blocks satisfies $|\{i : h_{t,i} \neq 0\}| \ll \dim(h_t)$. This implies that at inference time the effective subnetwork $\Theta_{S_t,r}$ activated by x_t occupies a small fraction of the total architecture, corresponding to the local sparsity pattern S_t.

Weight compressibility:

Singular spectra of layer operators W_ℓ obey power-law decay $\sigma_i(W_\ell) \sim i^{-\alpha_\ell}$. with $\alpha_\ell > 1$. Truncation to rank r_ℓ retaining $90\% - 95\,\%$ of Frobenius norm yields small performance loss. Thus $\sigma_{r_\ell+1}(W_\ell)$ in the theorem is small, explaining the empirically low projection error.

Sparsity: empirical evidence

Redundant attention heads and neurons:

Ablation studies reveal that only a subset of attention heads and MLP neurons are significant. This corresponds to a small effective mask S, consistent with the theoretical subnetwork size $\tilde{O}(s)$.

Warm-start lottery tickets:

Empirically, pruning and low-rank compression of pretrained LLMs followed by brief fine-tuning (`rewinding'') recovers nearly full accuracy. This aligns with the theorem's statement that the projected initialization $\theta_0 = \Pi_{S,r}(\hat{\theta})$ serves as a *warm-start winning ticket* whose subsequent restricted optimization achieves risk $R^\star + O(\varepsilon + \delta)$.

Plan

- Sparsity: motivations
- Compositionality
- Sparse compositionality implies 1) no curse of dimensionality 2) deep nets are a good H
- Compositionality implies good approximation + generalization+optimization
- Generalization: sparsity of weight matrices
- Optimization: sparse functions are easy to learn
- Which functions are compositional? Compositional structure of P
- Efficient Turing computability implies (sparse) compositionality
- Transformers and autoregressive
- Implications: lottery ticket and physics

For functions (or densities) that are efficiently Turing computable…

… compositionality holds with all its implications (no curse of dimensionality, good generalization, easier optimization)

But …what about functions from nature, from physics?

***Sparse compositional function* property of the world**

An extreme version of the “physical Church-Turing” thesis is:

all *physical phenomena can be efficiently simulated by a Turing machine* (many are).

This thesis would be exactly what you expect *if our Universe were a giant computer simulation.*

Notice that this version of the thesis is likely false because of the existence of physical phenomena that are not efficiently Turing computable such as chaotic dynamical systems.

Compositional sparsity <==> efficient computability

Remark

Suppose a function is physically realizable but is not *compositionally sparse.*Then it may not be *efficiently Turing computable and may not be simulated. Does* this mean *t*hat *it cannot be really “understood”?*

ML vs Physics

ML systems such as ChatGPT are compositionally sparse precisely because they are efficiently computable. On the physics side, however, computability cannot be assumed *a priori.*

Yet hierarchical and compositional structures are empirically ubiquitous in physical phenomena. An explanation is that finite propagation speed of information construing action-at-a-distance to to a time-space cone induces layered dependency structures for any fixed time horizon. In this paper we prove two complementary theorems formalizing this connection. The *Physics Causal–Locality Theorem* shows that local, finite-speed information propagation generates exactly sparse DAGs as in the computability case

ML vs Physics

We now show that *finite propagation and locality* induce a bounded--fan-in, depth-T computation graph whose node maps inherit the C^r and Lipschitz properties, and hence fall under the ML Approximation Theorem. Let $F_T := U_T \circ \cdots \circ U_1$. We assume a *finite speed of propagation*, that is, there exists a nondecreasing $R : \mathbb{N} \to \mathbb{N}$ such that for all $i \in V$ and all$x, x' \in X^V$,

$$x_{B_{R(T)}(i)} = {x'}_{B_{R(T)}(i)} \to \big(F_T(x)\big)_i = \big(F_T(x')\big)_i.$$

ML vs Physics

Then

Theorem: *Causal locality ⇒ compositional sparsity*

Fix a finite region $\Lambda \subset V$ and a horizon $T \in \mathbb{N}$. Under, the map

$$\mathscr{F}_{\Lambda,T} : X^V \to X^\Lambda, \qquad \mathscr{F}_{\Lambda,T}(x) := \big(F_T(x)\big)_\Lambda$$

admits a factorization through a depth-T DAG $\mathscr{G}_{\Lambda,T}$ whose node fan-in is at most$\kappa(\Delta, r)$, and whose node maps are C^r and L-Lipschitz. Consequently, for any $\varepsilon \in (0,1/4)$ there exists a ReLU network $\mathscr{N}_\varepsilon$ such that $\|\mathscr{F}_{\Lambda,T} - \mathscr{N}_\varepsilon\|_\infty \le \varepsilon$, with

$$\mathrm{depth}(\mathscr{N}_\varepsilon) \le c\, T \log(1/\varepsilon) \text{ and } \mathrm{size}(\mathscr{N}_\varepsilon) \le C\, s\, \varepsilon^{-d_*/r} \log(1/\varepsilon),$$

where s is the node count of $\mathscr{G}_{\Lambda,T}$ and $d_* = \max_v d_v$ is the maximal local arity ($d_v := |B_r(i)|$ for site i-nodes). The constants c, C depend only on (r, Δ), on the C^r bounds of the $\phi_{t,i}$, and on L.

What about the brain?

In Poggio and Smale (2003) we wrote "*A comparison with real brains offers another, and probably related, challenge to learning theory. The ``learning algorithms'' we have described in this paper correspond to one-layer architectures. Are hierarchical architectures with more layers justifiable in terms of learning theory?* Twenty years later, a most interesting theoretical question, both for machine learning and neuroscience, is indeed *why hierarchies, why depth.*

Plan

- Sparsity: motivations
- Compositionality
- Sparse compositionality implies 1) no curse of dimensionality 2) deep nets are a good H
- Compositionality implies good approximation + generalization+optimization
- Generalization: sparsity of weight matrices
- Optimization: sparse functions are easy to learn
- Which functions are compositional? Compositional structure of P
- Efficient Turing computability implies (sparse) compositionality
- Transformers and autoregressive
- Implications: lottery ticket and physics

Final Remarks

The property of *Sparse Compositionality* enables

- *Avoiding curse of dimensionality*
- *Sparse representation*: Efficient approximations of functions
- *Transfer learning*: Modules trained on one task can be reused.
- *Robust generalization*

Remarkably, sparse compositionality is very common: all efficiently computable functions are compositionally sparse (but their decomposition is not unique).
Remarkably, the magic of transformers is based on compositionality (autoregressive, *curriculum learning, CoT, etc.)*