Foundations of ML:

the computable math of learning

Tomaso Poggio, MIT, CBMM+CSAIL+BCS

A theory of learning:

*approximation, **optimization, generalization***






Research
A SCIENCE PARTNER JOURNAL

RESEARCH ARTICLE

Dynamics in Deep Classifiers Trained with the Square Loss: Normalization, Low Rank, Neural Collapse, and Generalization Bounds

Mengjia Xu[1,2], Akshay Rangamani[1], Qianli Liao[1], Tomer Galanti[1], and Tomaso Poggio[1*]

[1]Center for Brains, Minds and Machines, Massachusetts Institute of Technology, Cambridge, MA, USA.
[2]Division of Applied Mathematics, Brown University, Providence, RI, USA.

*Address correspondence to: tp@ai.mit.edu

Citation: Xu M, Rangamani A, Liao Q, Galanti T, Poggio T. Dynamics in Deep Classifiers Trained with the Square Loss: Normalization, Low Rank, Neural Collapse, and Generalization Bounds. Research 2023;6:Article 0024. https://doi.org/10.34133/research.0024

Submitted 26 July 2022
Accepted 24 November 2022
Published 8 March 2023

We need a theory of deep learning

- because we need better systems
- because a theory provides *deep explainability*
- because it may help understand basic principles of intelligence, including our own intelligence

We need a theory of deep learning

As we have discussed, current attempts to understand, predict and control behaviors of machine learning models are not based on fundamental principles and are unlikely to be successful. On the other hand, for the reasons given above, there is a real hope that such fundamental principles may exist and can be discovered. This is evidenced by recent progress in the theory of deep learning… These considerations call for a major expansion in the effort to understand fundamental principles of deep learning. …. **Understanding the fundamental theory of deep learning and applying its analyses to real systems is a compelling and urgent need.**

Misha Belkin, Jul 30, 2023

Theory of Learning: conceptual framework

Conceptual Framework of Machine Learning

A supervised learning problem is defined by an unknown probability measure μ on the product space $X \times Y$ from which the training data $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{m}$, -- supposed to be random -- is drawn.

The measure μ defines a function

$f_\mu : X \to Y$

satisfying $f_\mu(x) = \int y d\mu_x$, where μ_x is the unknown conditional measure on $\{x\} \times Y$.

The goal of learning theory might be said to be: ``find'' f that minimizes the generalization error, that is $\int_X (f - f_\mu)^2 d\mu_X$; in other words find a parametric f that approximates the unknown f_μ as well as possible

Conceptual Framework of Machine Learning

To do this we need to choose a space $\mathbb{F}$ of parametric functions - in which to search for f. The functions in $\mathbb{F}$ must be powerful enough to approximate a broad range of target functions with have a non-exponential number of parameters.

The steps in this conceptual framework, are thus

1. choose $\mathbb{F}$, in order to enable the next two steps (this is approximation theory);

2. find optimal values of the parameters of the approximator by minimizing a loss function on a training set (this is optimization theory);

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

A theory of learning:

approximation, ***optimization, generalization***

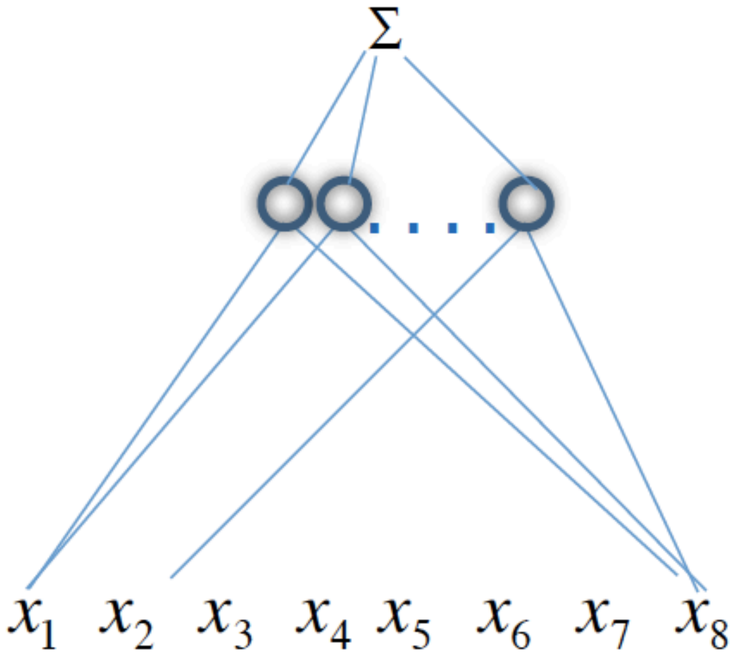
Approximation theory

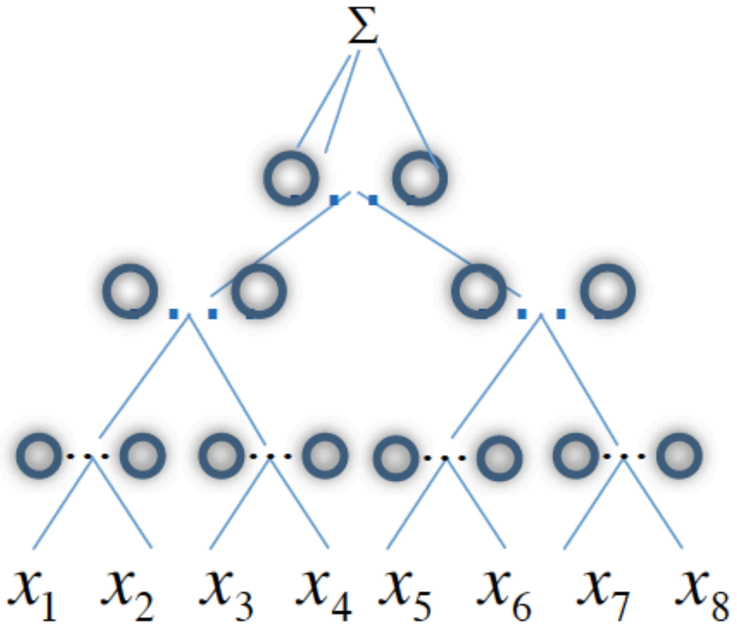
Example of an approximation puzzle in NN

- Why do deep neural nets apparently escape the curse of dimensionality?
- The puzzle has a long history in mathematics starting with Hilbert's thirteen problem
- It affects approximation and optimization

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, 1990

But…rate of approximation and curse of dimensionality

$$f(x_1, x_2, \ldots, x_8)$$

Both shallow and deep network can approximate a function of d variables equally well. However, they both suffer from a curse of dimensionality: number of parameters in both cases depends exponentially as

$$N = O(\epsilon^{-\frac{d}{m}})$$

Curse of dimensionality

Mhaskar, Poggio, Liao, 2016

Curse of dimensionality

A generic function associated to the CIFAr problem $f : \mathscr{R}^{1000} \rightarrow \mathscr{R}$ requires a very large memory (10 bins per coordinate corresponds to 10^{1000}, numbers which is $>> N_{Edd} \approx 10^{80}$ the number of all the protons in the universe…there are ~ 10^{15} synapses in the human brain).

with Seth Lloyd

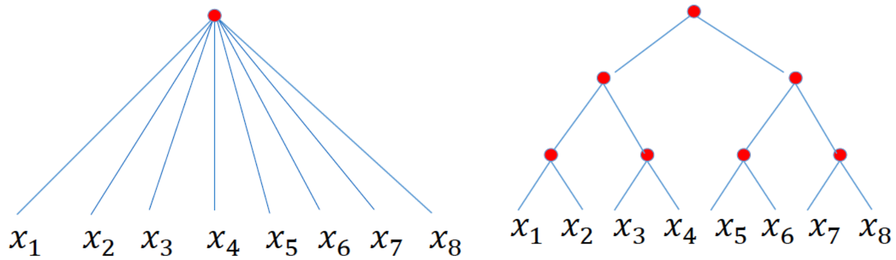
Definition

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 *constituent functions*, each depending on a small number of variables $\leq d_0$. Sparse constituent function means that there it does not suffer from the *"curse of dimensionality".*

Deep NNs can avoid the curse for compositionally sparse 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 compositionally sparse (example: its function graph is a binary tree). Both shallow and deep network can approximate f equally well. 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 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.

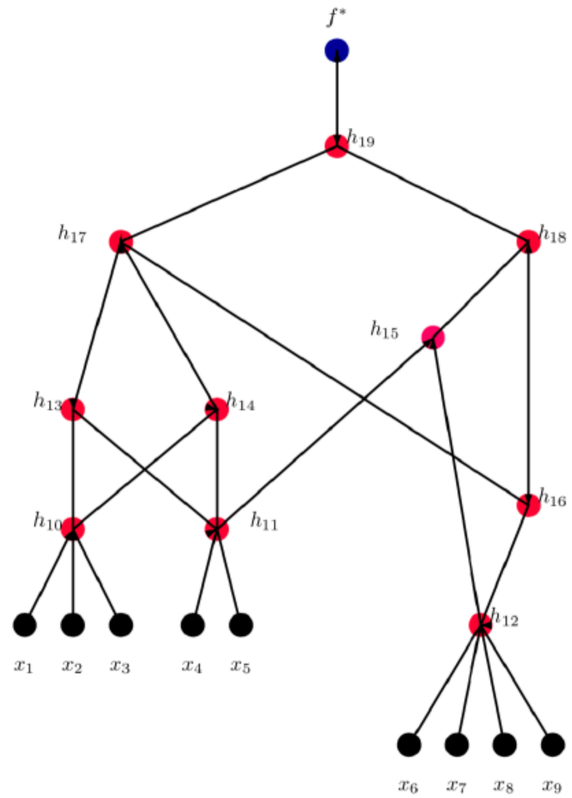
More general result

Theorem 3. *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 $f : \mathbb{R}^n \mapsto \mathbb{R}$ 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 n, whereas shallow networks cannot directly avoid the curse. In particular, the complexity of the best approximating shallow network is exponential in n*

$$N_s = \mathcal{O}(\epsilon^{-\frac{n}{m}}), \tag{9}$$

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

$$N_d = \mathcal{O}(\sum_{v \in V} \epsilon^{-d_v/m_v}). \tag{10}$$



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

For smooth functions that are compositionally sparse,
approximation is possible without *curse of dimensionality.*

But …is it possible to characterize compositionally sparse functions?

Perhaps — we conjectured — compositional functions
emerge for specific reasons,
such as Physics (interactions of physics are local and thus sparse…)

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 $\leq d_0$. Sparse means *"no curse of dimensionality"*.

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.

Compositional sparsity <==> efficient computability

- *Theorem (new)* For Lipschitz continuous functions on the reals, *compositional sparsity* is equivalent to *efficient computability*.

- *Corollary* For every function which is efficiently computable by a Turing machine there exist a sparse deep network that approximates it without curse of dimensionality.

Poggio, Fraser , Bulletin of the AMS, 2024
with thanks to Santosh Vempala and Eran Malach

Summary

- Theorems says that all efficiently Turing computable functions/tasks/programs have a sparse compositional representation, in terms of sparse functions/modules/tasks/subroutines. Such a compositional representation is not unique.

- For every (efficiently) Turing computable function there exist a *sparse deep* network that approximates it arbitrarily well, without curse of dimensionality — corresponding to a sparse graph of the function.

- In the limit of maximum depth, a Turing machine corresponds to a Boolean function which can be represented as the composition of two-inputs AND and OR operations.

Generalization

Rademacher Bounds

Let $\mathbb{H}$ be a set of real-valued functions $h : \mathcal{X} \to \mathbb{R}$ defined over a set $\mathcal{X}$. Given a fixed sample $S \in \mathcal{X}^m$ the empirical Rademacher complexity of $\mathbb{H}$ is defined as follows:

$$\mathcal{R}_S(\mathbb{H}) := \frac{2}{m}\mathbb{E}_\sigma \left[\sup_{h \in \mathbb{H}} \left| \sum_{i=1}^{m} \sigma_i h(x_i) \right| \right] .$$

The expectation is taken over $\sigma = (\sigma_1, \ldots, \sigma_m)$, where, $\sigma_i \in \{\pm 1\}$ are i.i.d. and uniformly distributed samples.

ρ in this bound controls expected error (well known)

A typical generalization bound that holds with probability at least$(1-\delta)$, $\forall f \in \mathbb{F}$ has the form:

$$|L(g) - \hat{L}(g)| \leq \frac{1}{\gamma}\mathbb{R}_N(\mathbb{G}) + c_2\sqrt{\frac{\ln(\frac{1}{\delta})}{2N}}$$

where $L(g) = \mathbf{E}[\ell_{gamma}(g(x), y)]$ is the expected loss and $\hat{L}(g) = \frac{1}{N}\sum_n \ell_{gamma}(g(x_n), y_n)$ is the empirical loss. If $\hat{L}(g) = 0$ and the margin is $= 1$ at convergence then

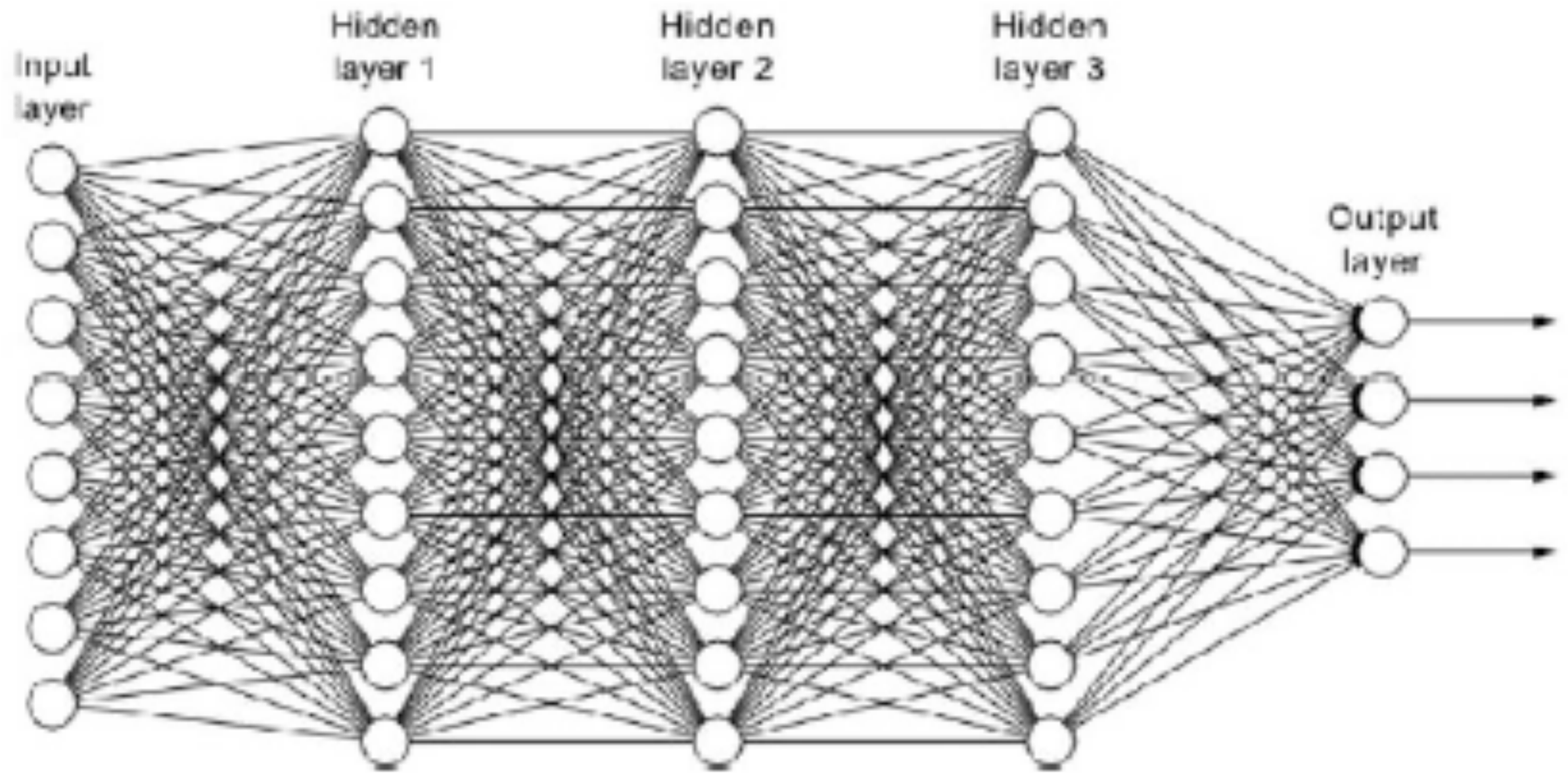
$$\mathbb{R}_N(\mathbb{G}) = 2\rho\mathbb{R}_N(\mathbb{F}) = 2\rho\frac{\mathbb{R}(\mathbb{F})}{\sqrt{(N)}}$$

implying that

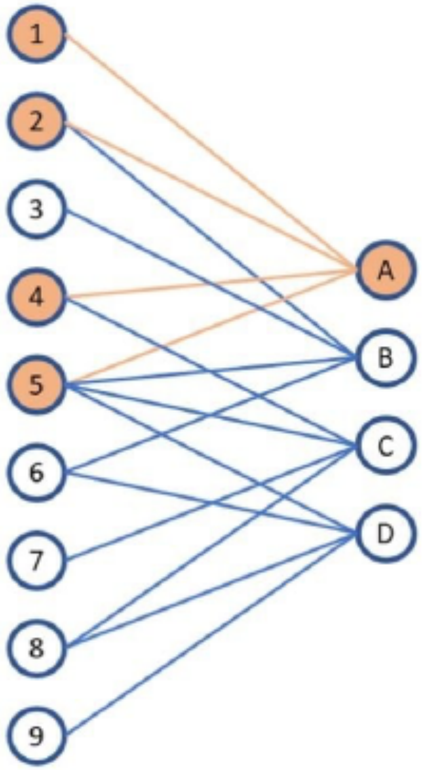
$$L(g) \leq \frac{2\rho}{\sqrt{(N)}}$$

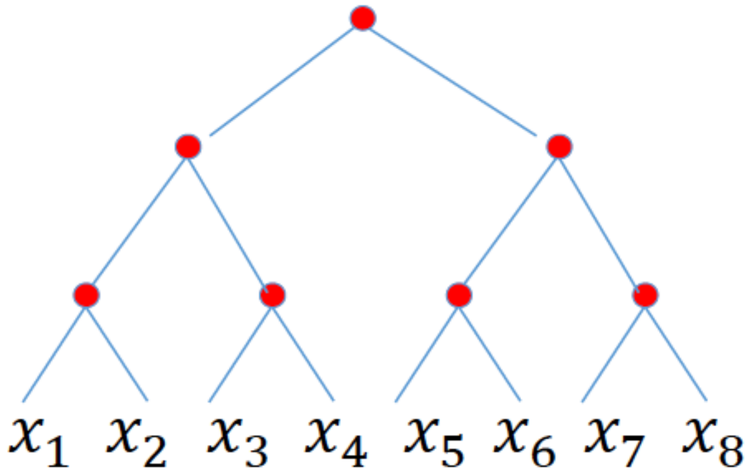
Menjia Xu et al. *Research 2023*

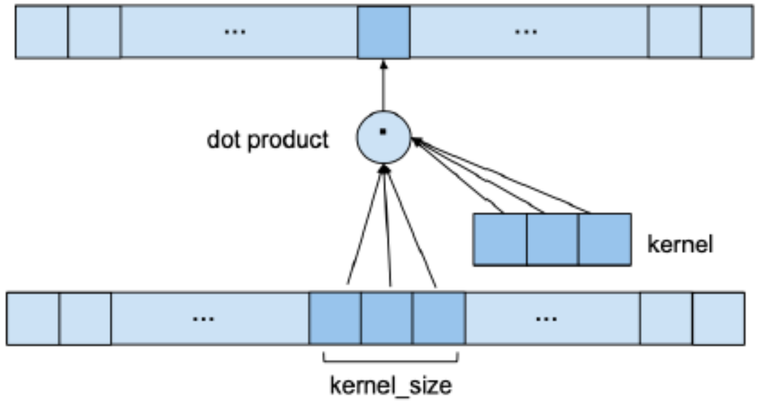
Dense architecture



Sparse (“convolutional” is standard term) architecture







Novel bounds for Sparse Networks

Theorem

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.

Galanti, Xu, Galanti, Poggio, Neurolps 2023

Novel bounds for Sparse Networks (Galanti, Xu, Galanti, Poggioi)

Theorem (informal)

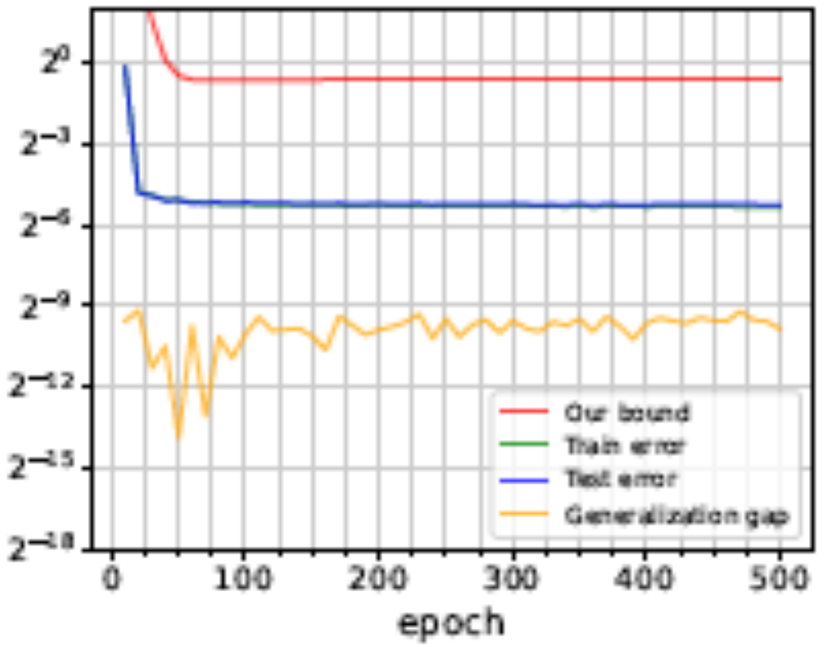
Consider a deep RELU network with a $n \times n$ *dense weight matrix* W *at a certain layer. Its contribution to the Rademacher complexity of the network is* $\|Wx\| = \sqrt{\|Wx\|^2} = \sqrt{(\sum_k^n \|w_k\|^2)\|x\|^2} \leq \|W\|\|x\|$

Suppose now that W *is sparse, that is row* i *contains only* k_i *non-zeros (*$k_i < n$*) components. Then each row contributes* $\|w_i\|^2(\frac{k_i}{n}\|x\|^2)$ *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*

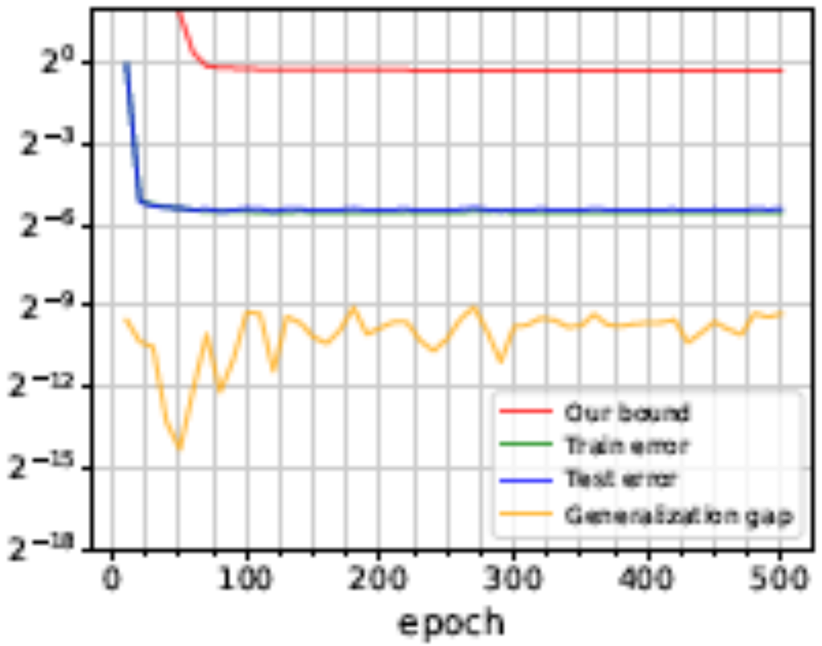
$$\sqrt{(\sum_i^n \|w_i\|^2)\frac{k_i}{n}\|x\|^2}$$

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

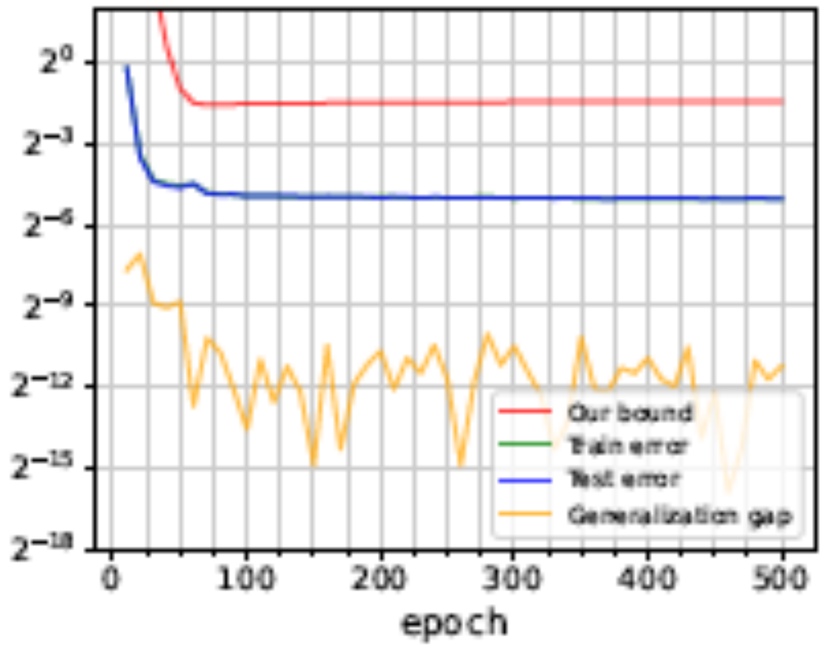
MNIST sparse network: non-vacuous bound



CONV-4-50



CONV-4-500



CONV-3-100

Galanti, Xu, Galanti, Poggio, NeurIps 2023

Sparsity in the weight matrices

- Sparsity in the weight matrices implies generalization bounds that are order of magnitude better than for deep networks with dense matrices

- Weight sharing plays a minor role…so it is not convolution that leads to better generalization but the fact that most kernels are small inducing sparse (Toeplitz) weight matrices

- *Sparsity is key for approximation and for generalization*

Low rank of the weight matrices

- Lot of empirical evidence for low-rank bias in NN training

Tomer will speak about low rank bias

- So…does low rank implies better generalization?

- What about rank instead of sparsity in the weight matrices? Good Rademacher complexity?

- Low rank constraint improves generalization (theorem)

Low rank of the weight matrices

Theorem 4.1 (Gaussian complexity of deep Lipschitz neural network). *Let $\mathcal{F}_L$ be the class of deep neural networks of depth L as defined in Preliminaries 1. The Gaussian complexity of this class of function over the sample of data $\mathbf{X} \in \mathbb{R}^{m \times d}$ is upper bounded as follows;*

$$\hat{\mathcal{G}}_m(\mathcal{F}_L) \lesssim \frac{||\mathbf{X}||_F}{m} \left\{ \left(||\mathbf{W}_1||_F \sqrt{h_1} \right) \prod_{i=2}^{L} C_1 \|\mathbf{W}_i\|_2 + \sum_{i=2}^{L} C_1^{L-i} C_2 \sqrt{2r_i} \kappa_i \|\mathbf{W}_i\|_F \sqrt{h_i} \right\} \quad (4)$$

where $\kappa_i \propto \left(\prod_{j=1, j \neq i}^{d} \|\mathbf{W}_j\|_2 \right)$ *and* $r = \min_{j \leq i}(\text{rank}\,(\mathbf{W}_j))$.

Akshay will speak about generalization bounds based on rank

Pinto, Ranghamani, Poggio,2024

Remarks

- Sparsity based on number of dimensions is a guideline but obviously it is too coarse a criterion…so better bounds are possible as shown by our sparsity-based Rademacher bounds.

- Notice that these bounds contains the product of norms. This is suboptimal as one can see in the case of linear deep networks where clearly $\|W_L\| \cdots \|W_1\| \geq \|W_L \cdots W_1\|$

- Better bounds eliminating the product of norms may be achieved by using path-norm based bounds. Path-norm summates, for instance, the L_p norms of all possible paths (equivalent to the product of $L_{p,\infty}$ norms of weight matrices).

Summary on sparsity

Why hierarchies (in deep nets, in primate visual cortex)?

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.*

More puzzles, possibly yielding fundamental results for intelligence (natural and artificial)

Can deep nets represent/approximate every function even with a large number of variables? How could deep nets escape the curse of dimensionality (in approximation)?

Why depth?

Why is generalization possible despite overfitting?

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

More puzzles, possibly yielding fundamental results for intelligence (natural and artificial)

Why're CNNs so much better than dense NN?

Why is generalization possible despite overfitting?

Why does the RELU work so well?

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

Why is optimization of deep networks so "easy" despite the very high dimensionality?

Optimization

Optimization: fitting parameters of the approximating neural network using data

Motivation: exploit sparse compositionally for optimization

I mostly described existence results: there exist sparse networks that can approximate any function without curse of dimensionality.

*The next obvious question: under which conditions can optimization **learn** these sparse networks?*

It is clearly impossible always to learn the underlying function (think about a training sets of random names and random telephone numbers).

The guiding idea is that sparse compositionally is important not only for approximation but also for optimization (remember the non-exponential requirement on the parameters)

Learning: end-to-end vs autoregressive framework

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

Two learning frameworks:

- I have training sets for each g_i, that is x_i, y_i, where $g_i(x_i) = y_i$ (autoregressive framework)
- I have a training set for f that is x, y where $f(x) = y$ (standard framework/end-to-end)

Remarks

- The conjecture in guiding future work is that useful learning algorithms (that learn well) exploit sparsity. For instance, decision trees (which are an old, effective approach in ML) can be approximated by sparse Boolean functions, where the number of Fourier coefficients is polynomial in the size of the tree. Notice that every Boolean function of n variables can be represented by a tree of $\leq 2^n$ nodes and of depth $\leq n$.

Claim 5.3 (Mansour, 1994) Let T be a decision tree with m nodes. There exists a sparse function h such that all the Fourier coefficients of h on sets larger than t are 0 , where $t = \log m/\varepsilon$, and $E\left[(T-h)^2\right] \leq \varepsilon$.

Remarks

- Some sparsity property is probably true for transformer architectures since sparse approximations of transformers via pruning, low-rank approximations etc. seem to work well.

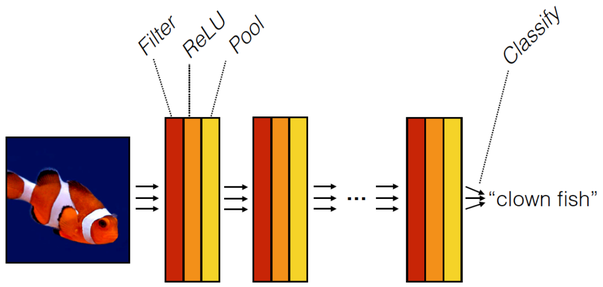
- Interestingly, transformers also are very good at learning sparse Boolean functions with low Fourier terms and at generalizing well (they are much better than LSTMs).

- Notice that *learning the support* solves a big part of the problem of finding the important variables in a sparse Boolean function (the number of examples needed to learn a Boolean formula is exponential in the size of the vocabulary in the general case).

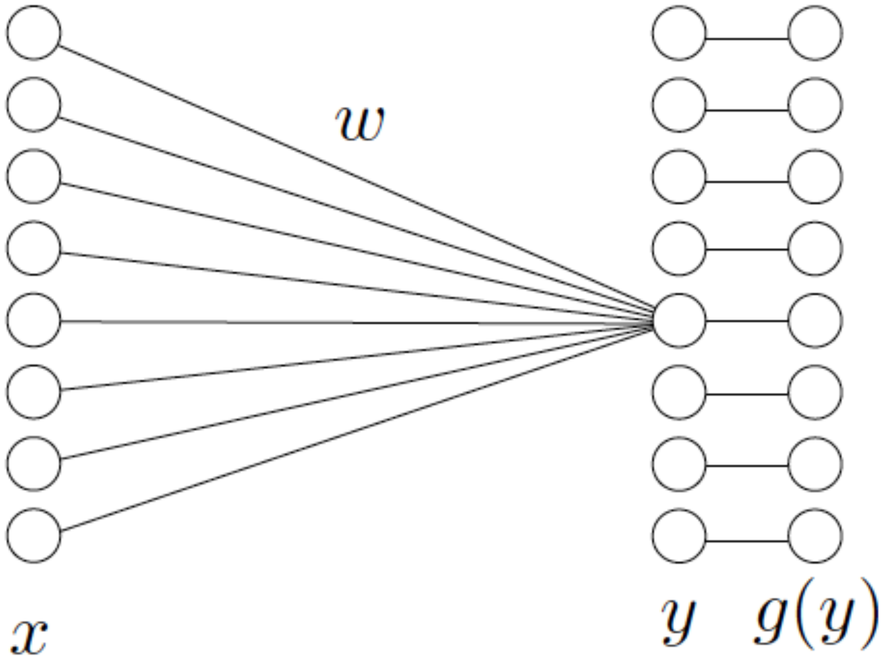
Pierfrancesco will speak about learning support

Computation in a deep neural net

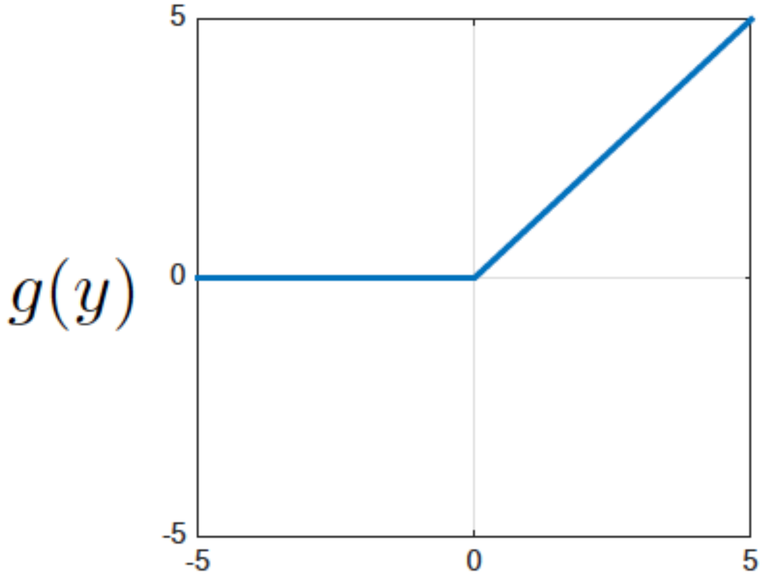
Computation in a neural net



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



Rectified linear unit (ReLU)



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

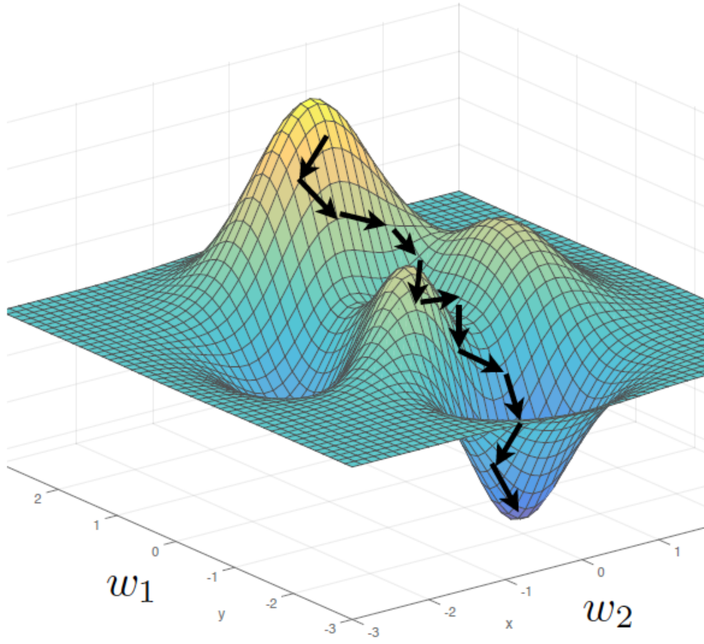
Gradient descent and SGD

$$\underset{\mathbf{w}}{\operatorname{argmin}} \sum_i \ell(\mathbf{z}_i, f(\mathbf{x}_i; \mathbf{w})) = L(\mathbf{w})$$

One iteration of gradient descent:

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta_t \frac{\partial L(\mathbf{w}^t)}{\partial \mathbf{w}}$$

learning rate



Open problems

We want to understand how weights and activations evolve

- **in time**

and

- **across layers**

during optimization/training.

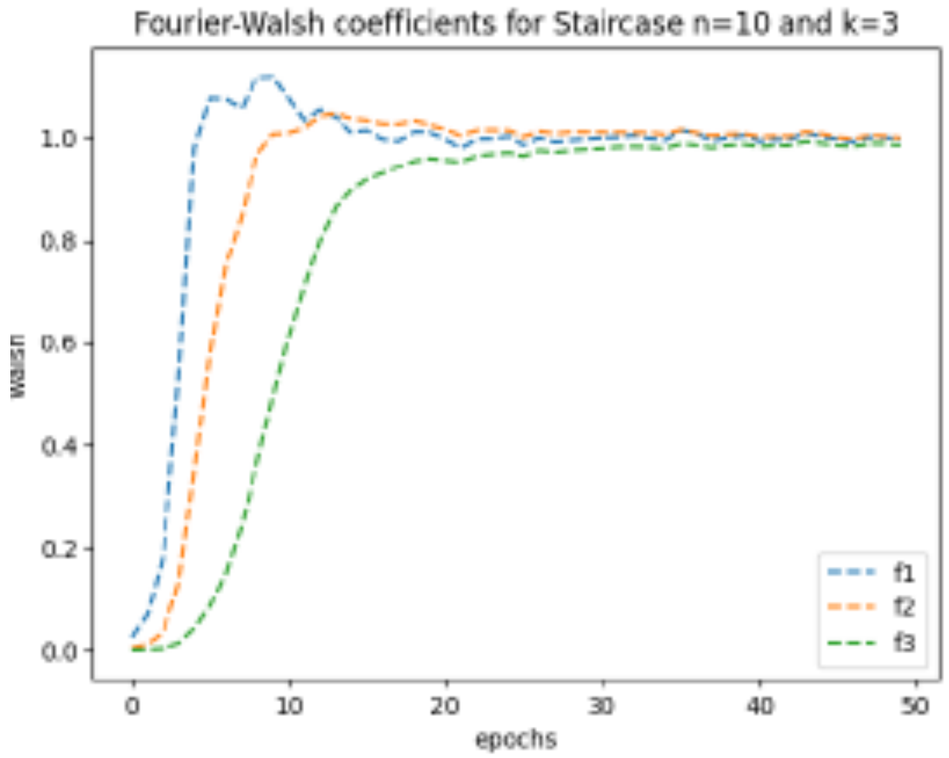
The phenomenon of NC that happens at convergence of overparametrized networks (with square loss regularization is required) characterizes in part the *evolution of weights*.

It is conjectured that simple *features* are learned first and more complex ones later.

Conjecture

- Assume $\sigma(z) = z^2$
- Assume input $x = x_1, \cdots, x_{d-1}, 1$
- In the first layer the asymptotic result should be a W_1 weight matrix that set to zero the variables outside of the unknown function support because regularization induces a minimum norm solution (linear terms gets large weights , weights associated with nonlinear terms in target have small but nonzero weight).
- In the second layer, after the σ, the new inputs to W_2 are mixtures of constant, linear and quadratic monomials; under the assumption of a sufficiently large number of units in the second layer, a minimum norm solution for W_2 eliminates monomials that are not included in the support of the function (unless they are part of monomials in the target function in which case they need to small weights). The same process takes place at higher layers.

Remarks



$$S_k(x) = \sum_k \chi_{1:k} = x_1 + x_1x_2 + \ldots + \chi_{1:k}$$

The previous conjecture describes how features are learned across layers, but not across time.

Wing architecture

For a class of monotonic activation functions including GeLU, neural networks having a pyramidal structure, that is no layer has more hidden units than the input dimension, produce necessarily connected decision regions. This implies that a sufficiently wide hidden layer (at least $> d + 1$) is necessary to guarantee that the network can produce disconnected decision regions.

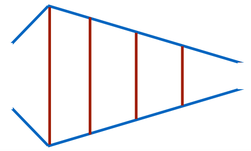
Theorem (Nguyen et al., 2018)

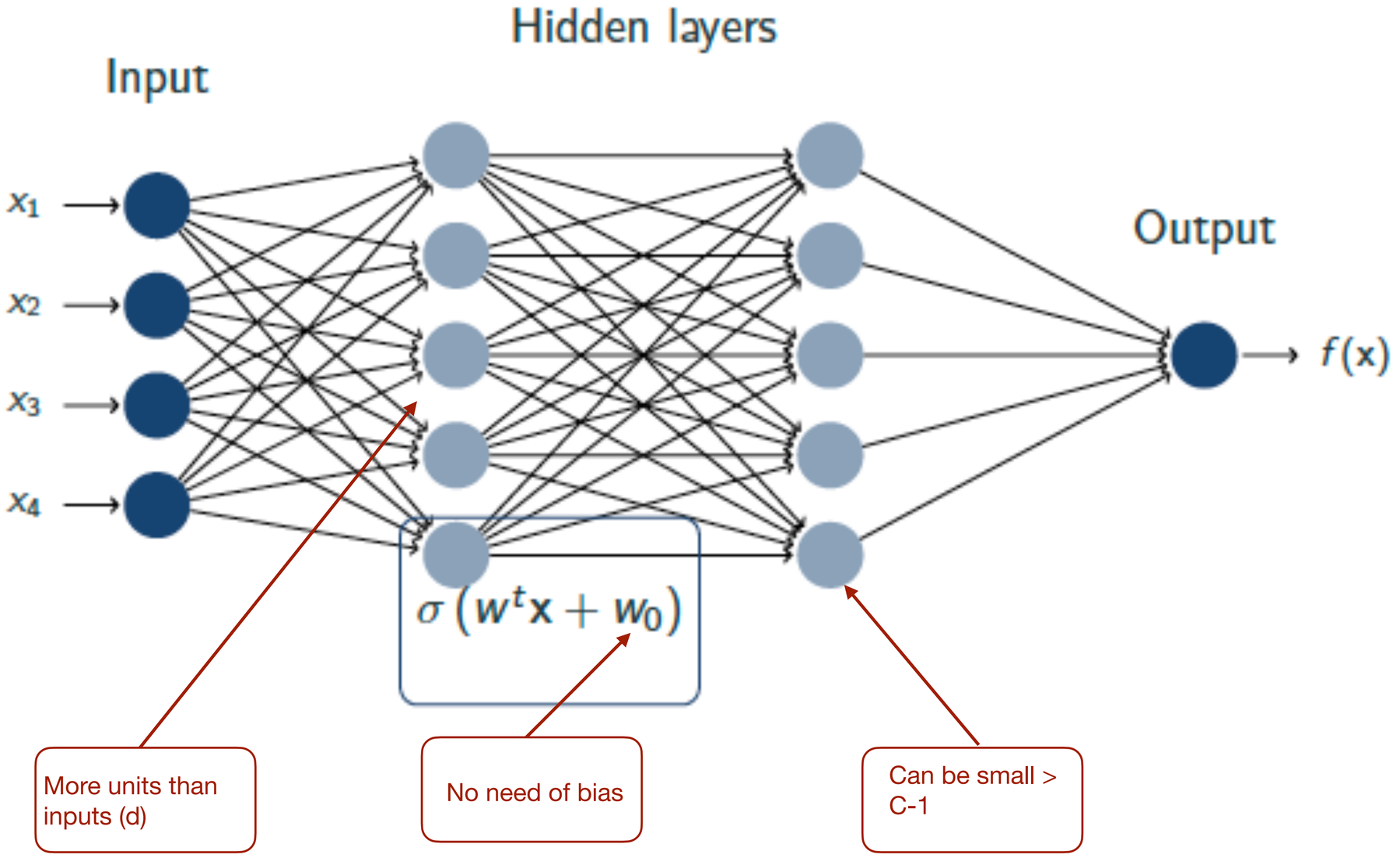
Let the width of the layers of the feedforward network network satisfy $d = n_0 \geq n_1 \geq \ldots \geq n_{L-1}$ *and let* $\sigma_l : \mathbb{R} \to \mathbb{R}$ *be a continuous, strictly monotonically increasing function with* $\sigma_l(\mathbb{R}) = \mathbb{R}$ *for every layer* $1 \leq l \leq L - 1$ *and all the weight matrices* $\left(W_l\right)_{l=1}^{L-1}$ *have full rank. Then every decision region* C_j *is an open connected subset of* $\mathbb{R}^d$ *for every* $1 \leq j \leq m$.

Remark:

Full rank of the weight matrices is necessary to get connected decision regions when the other conditions are satisfied.

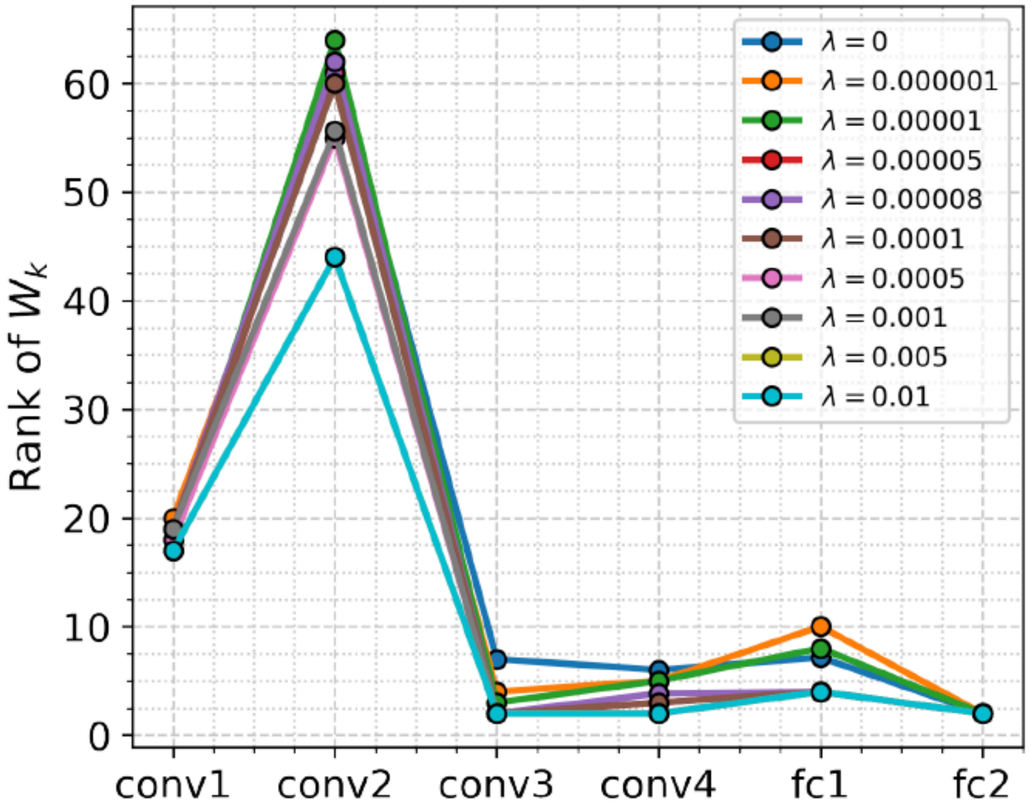
Wing architecture

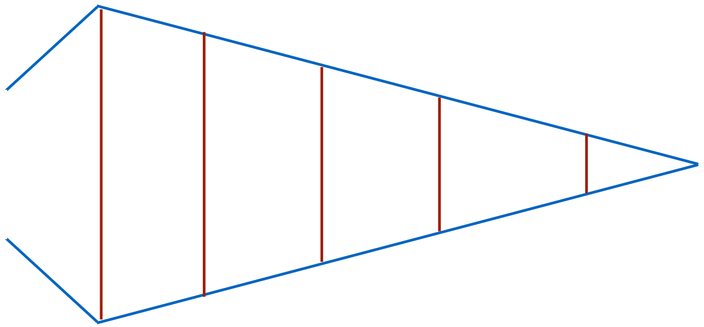




CENTER FOR
Brains
Minds+
Machines

Rank increase and then decreases towards higher layers





Weight matrices across layers: Neural Collapse

Akshay will speak about NC

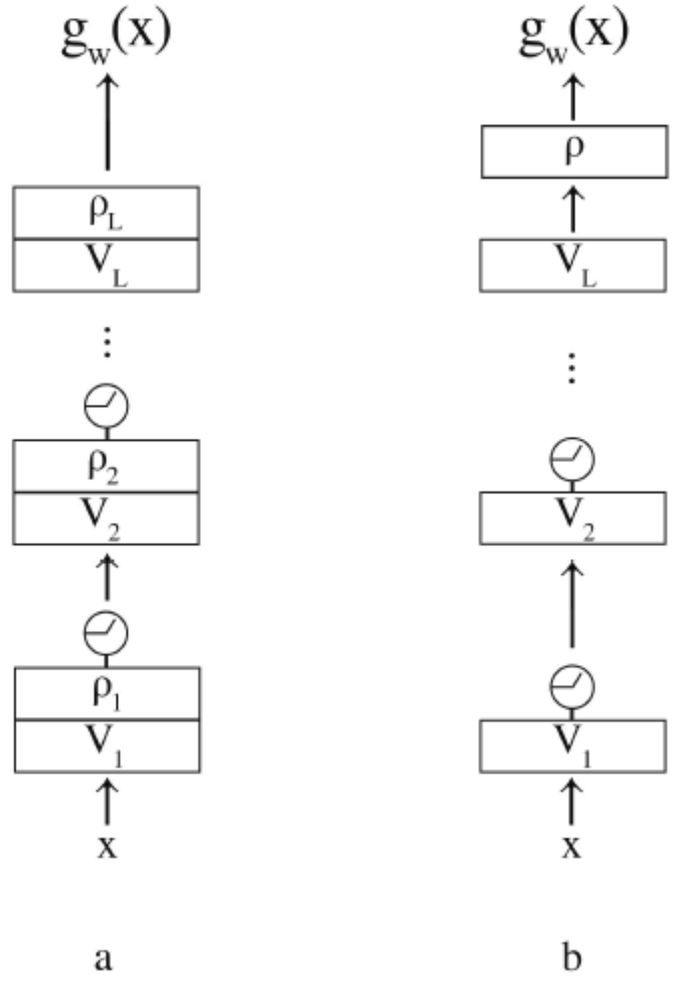
Why bias towards rank minimization across layers

Tomer will speak about rank

Gradient flow dynamics

$$\dot{\rho} = -2[\sum_{S} \rho(f_n)^2 - \sum_{S} f_n y_n] - 2\lambda\rho$$

$$\dot{V}_k = 2\rho \sum_{S} [(\rho f_n - y_n)(V_k f_n - \frac{\partial f_n}{\partial V_k})]$$



$$f(x) = (V_L\sigma(V_{L-1}\cdots\sigma(V_1 x))) = V_L D_{L-1}(x)V_{L-1}\cdots D_1(x)V_1 x$$

$g(x) = \rho f(x)$ ρ is the product of the Frobenius norms of the weight matrices

Landscape theorems (informal)

Theorem (Cooper)

We assume an overparametrized deep network g with smooth RELU activation functions and square loss. Then the minimizers W^* achieve zero loss and are highly degenerate with dimension $d - N$

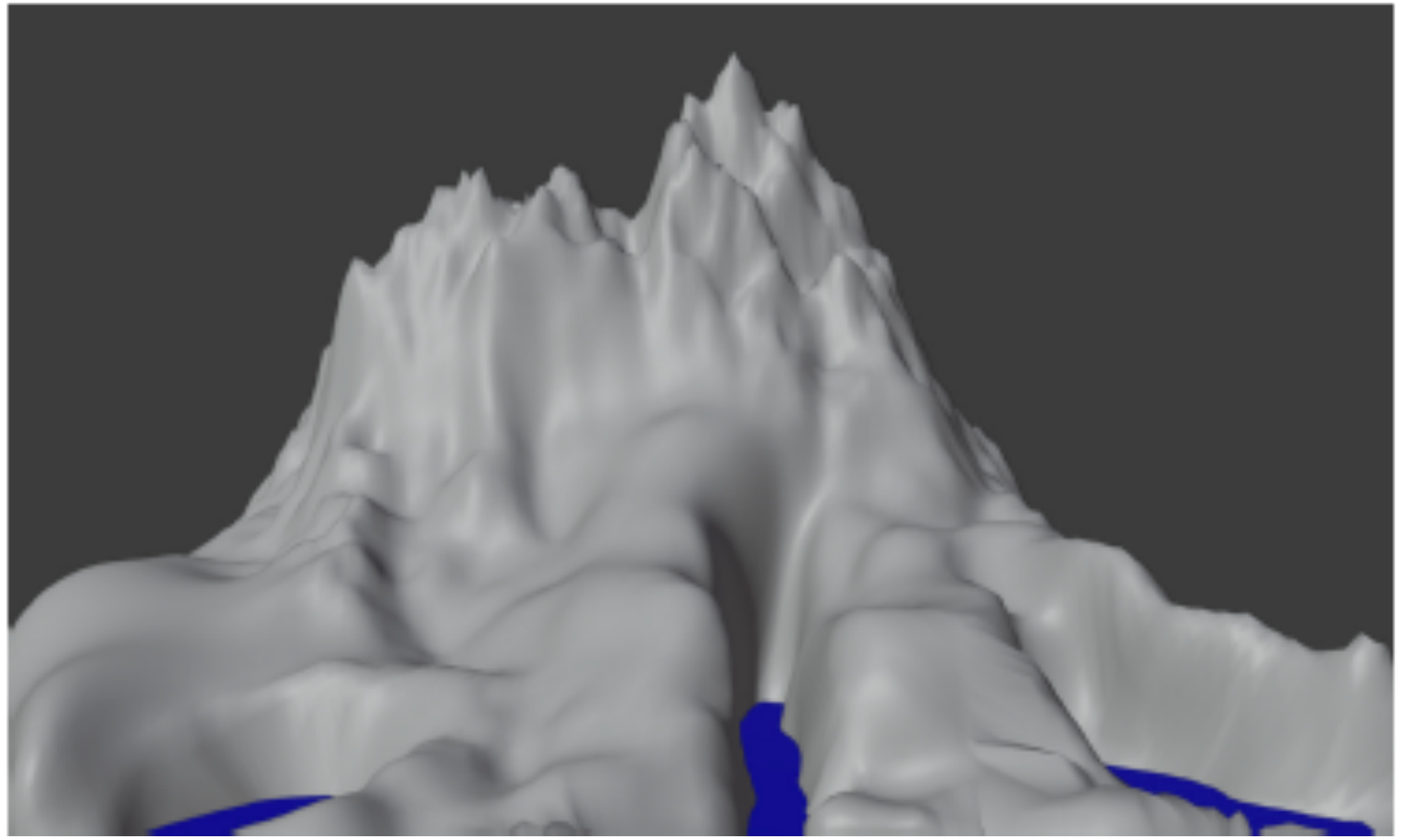
Theorem (Nguyễn)

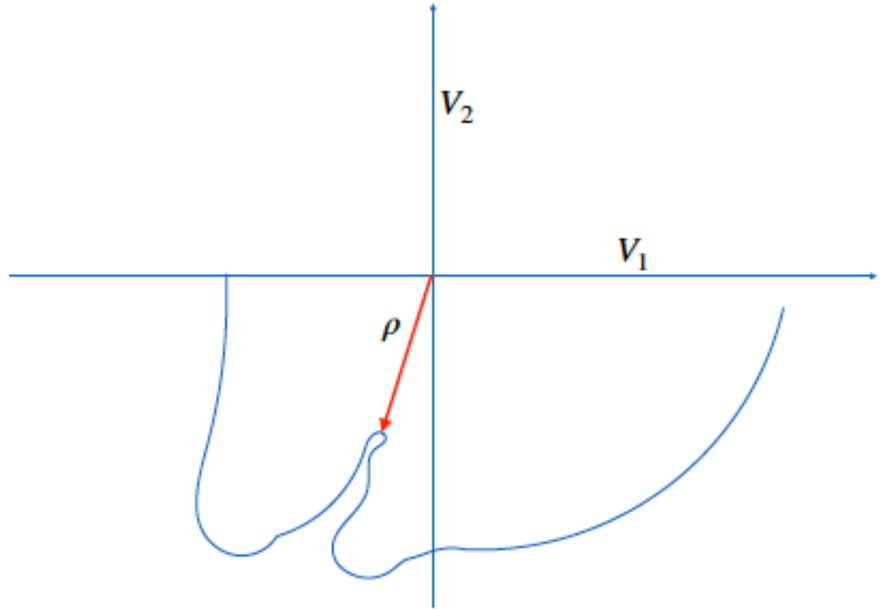
If the first layer of the network has at least $2N$ neurons, where N is the number of training data and if the number of neurons in each subsequent layer decreases, then every sublevel set of the loss is connected.

Theorem (N. Bottman, Y. Cooper, A. Lerario)

Let L denote the square loss function for a feedforward neural network. For a multiplicative regularizer, if the depth ℓ of the neural network is greater than or equal to 4, not only does $L_\epsilon = L + \lambda R_m$ fail to be Morse, but it has a positive-dimensional locus of degenerate critical points.

Qualitative dynamics: observations on landscape





Assume regularization: which one?

We know that

$$\mathscr{L} = \sum_{n}^{N} (g(x_n) - y_n)^2 + \sum_{k=1}^{L} \lambda_k ||\rho_k||^2$$

implies isolated minima but we know that

$$\mathscr{L} = \sum_{n}^{N} (g(x_n) - y_n)^2 + \lambda \sum_{k=1}^{L} ||\rho_k||^2$$

may not (it does not in linear case , may in the nonlinear case). Is this also true for

$$\mathscr{L} = \sum_{n}^{N} (g(x_n) - y_n)^2 + \lambda \Pi_{k=1}^{L} ||\rho_k||^2$$

?

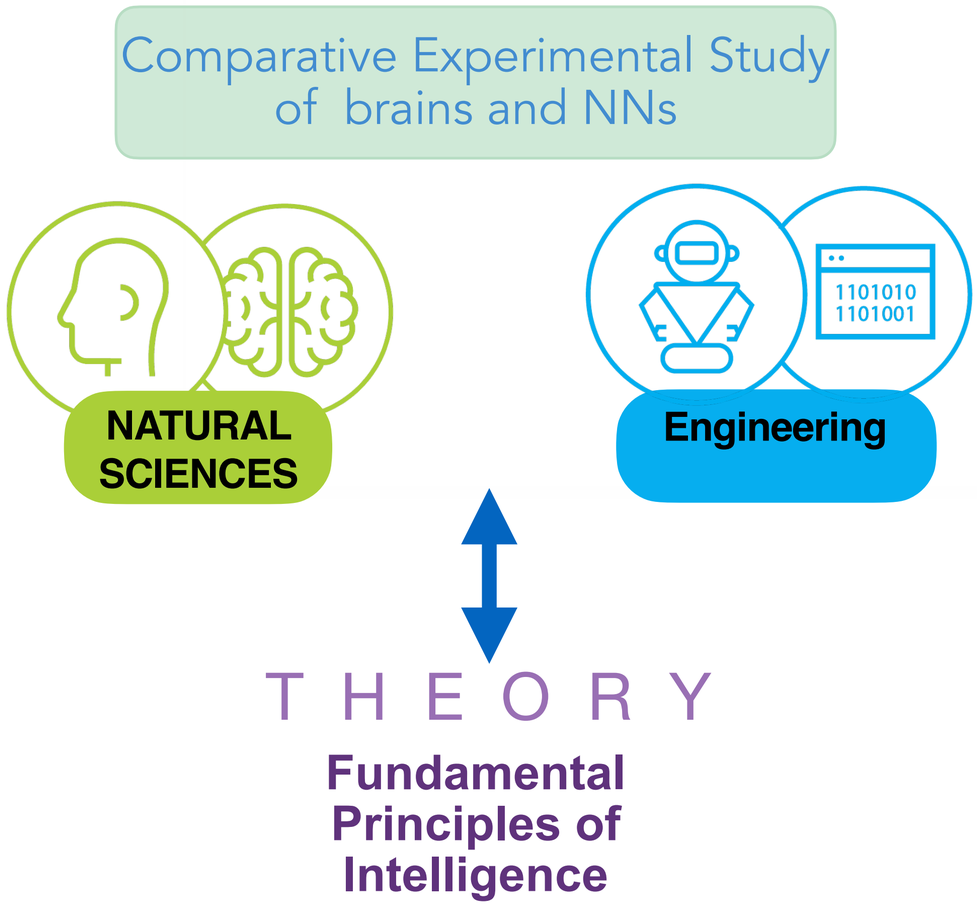
Optimization: implicit regularization in SGD

Pierfrancesco will speak about implicit regularization

Science and Engineering

We should invest not only in engineering (A+M+M+…do that) but also in science. For the first time in human history we have more than just human intelligence to study. This suggests the following research program articulated in 3 thrusts

A research program



- Comparative study of intelligent behavior: what are similarities and differences? What are the roots of the differences?
- Recording from inside the brain and the LLMs, looking for circuits, again similarities and differences.
- Theory looking for fundamental common principles

Optimization by SGD: noise and rank

Intrinsic fluctuations of SGD

All gradient descent methods try to converge to points in the parameter space that has zero or small gradient, *i.e.,* try to minimize $\| \dot{V}_k \|, \forall k \in [L]$ iteratively, over different minibatches (batch size is *B*).

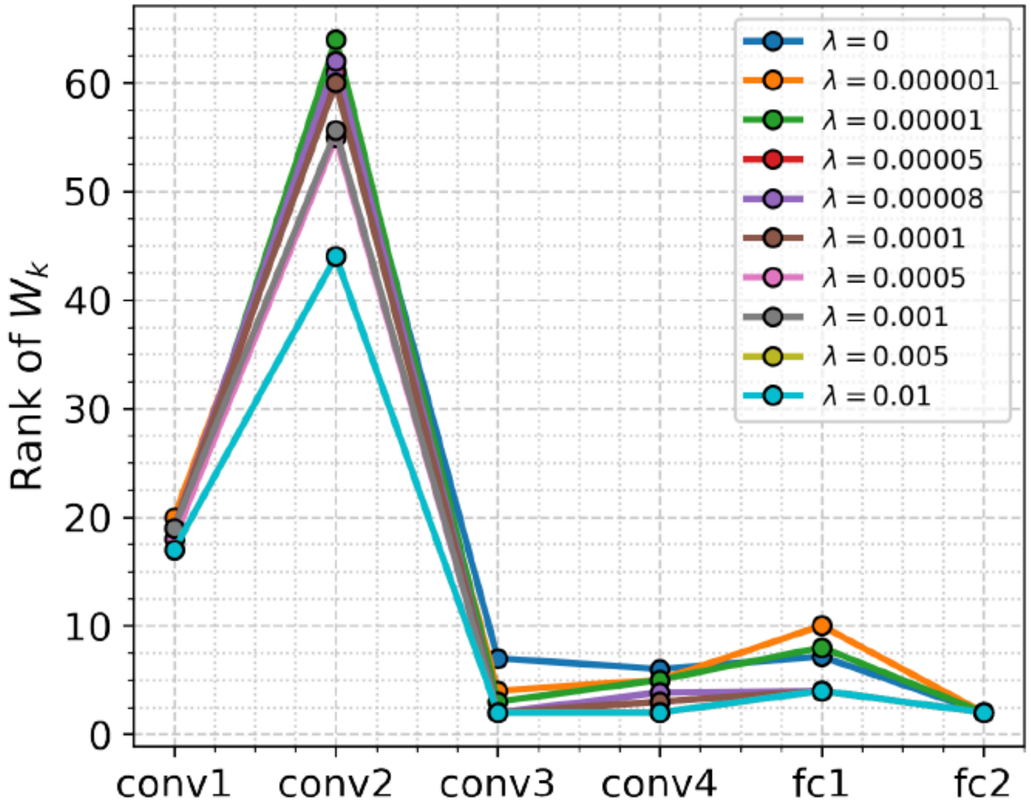
$$\dot{V}_k = \frac{2}{B}\rho \sum_{n \in B} \left[\underbrace{(1 - \rho \bar{f}_n)}_{\text{error}} \left(-V_k \bar{f}_n + \frac{\partial \bar{f}_n}{\partial V_k} \right) \right]$$

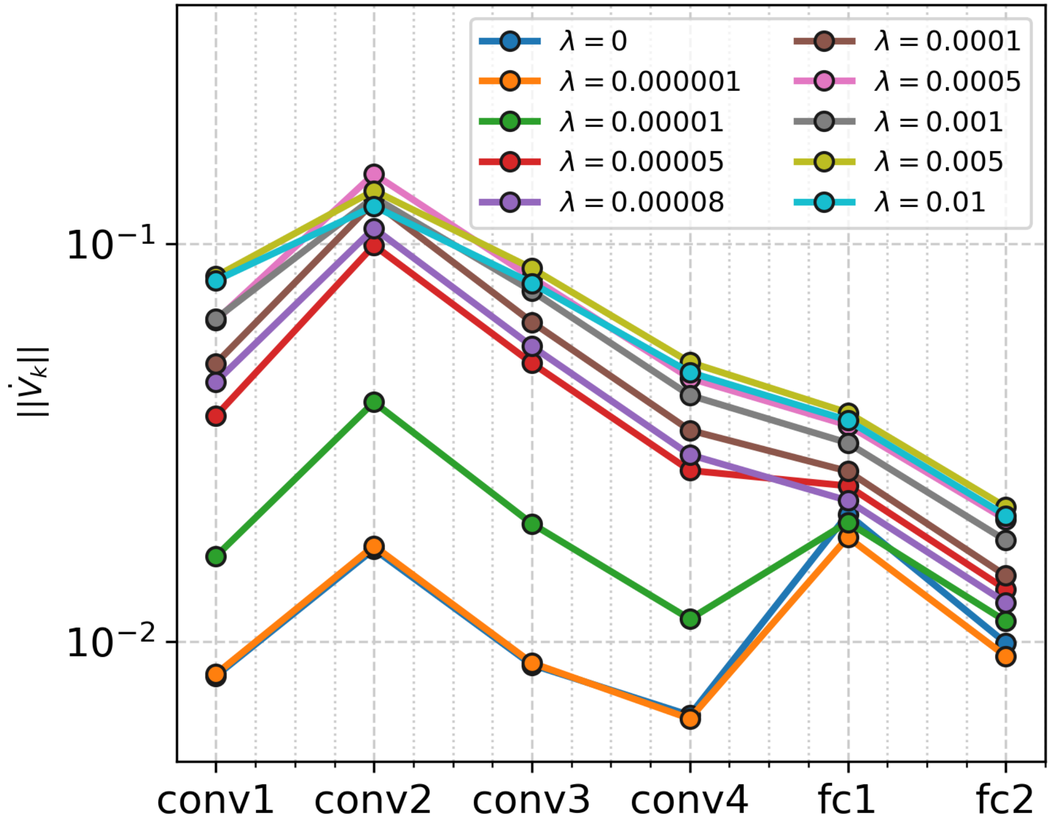
Then, $\|\dot{V}_k\| = \| \frac{2\rho}{B} \sum_{n \in B} \ell_n (\frac{\partial \bar{f}_n}{\partial V_k} - f_n V_k) \|$

Our results suggest that the fluctuations are due to a specific SGD dynamics, arising from the competition between *minimizing the error* and *minimizing the rank*. The dynamics can also be described as switching between slightly different V_k, each one trying to fit just one mini batch.

SGD noise depends on rank of the weight matrices (V_k)

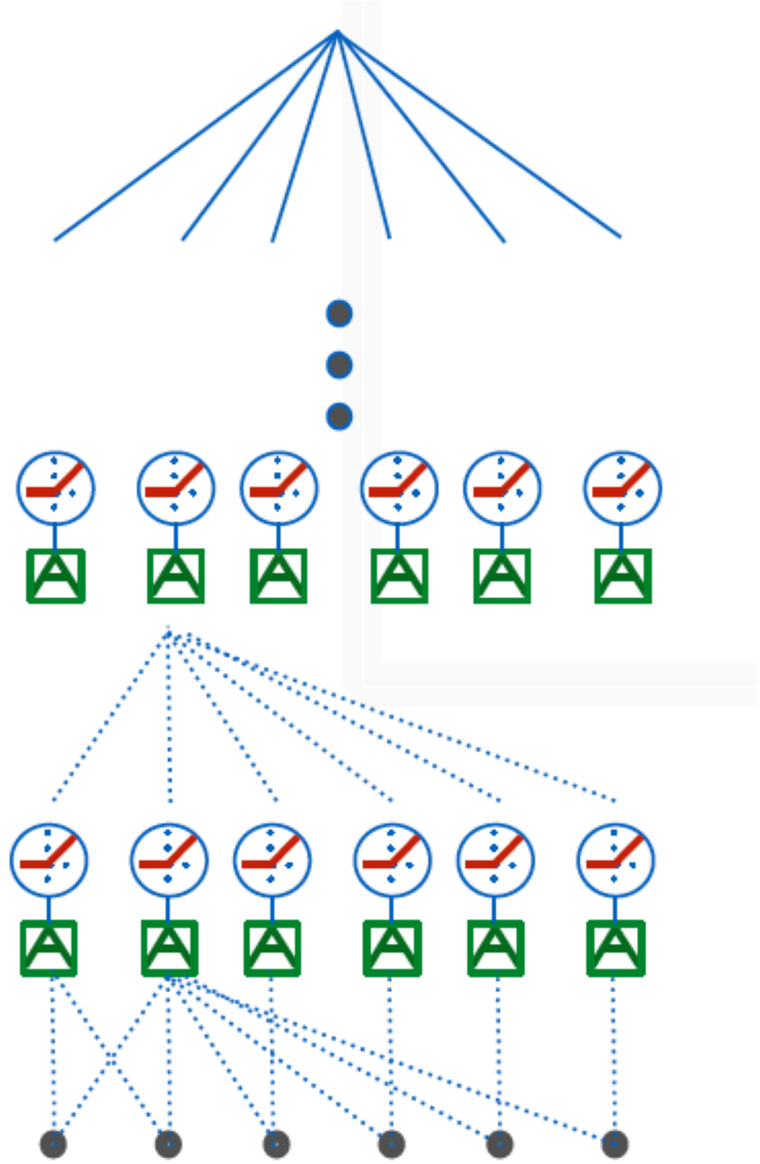
❑ **Layer-wise $||\dot{V}_k||(\forall k \in [L])$ in logarithmic scale by 10 different λ at convergence**. The SGD noise across layers, as measured by $||\dot{V}_k||$, is small when λ is set to 0 or 1e-06. An increase of λ from 1e-05 to 0.01 generates larger fluctuations, especially for the first four convolution layers. The last two fully connected layers tend to consist of low-rank matrices, corresponding to smaller $||\dot{V}_k||$.





Sparsity and Transformers

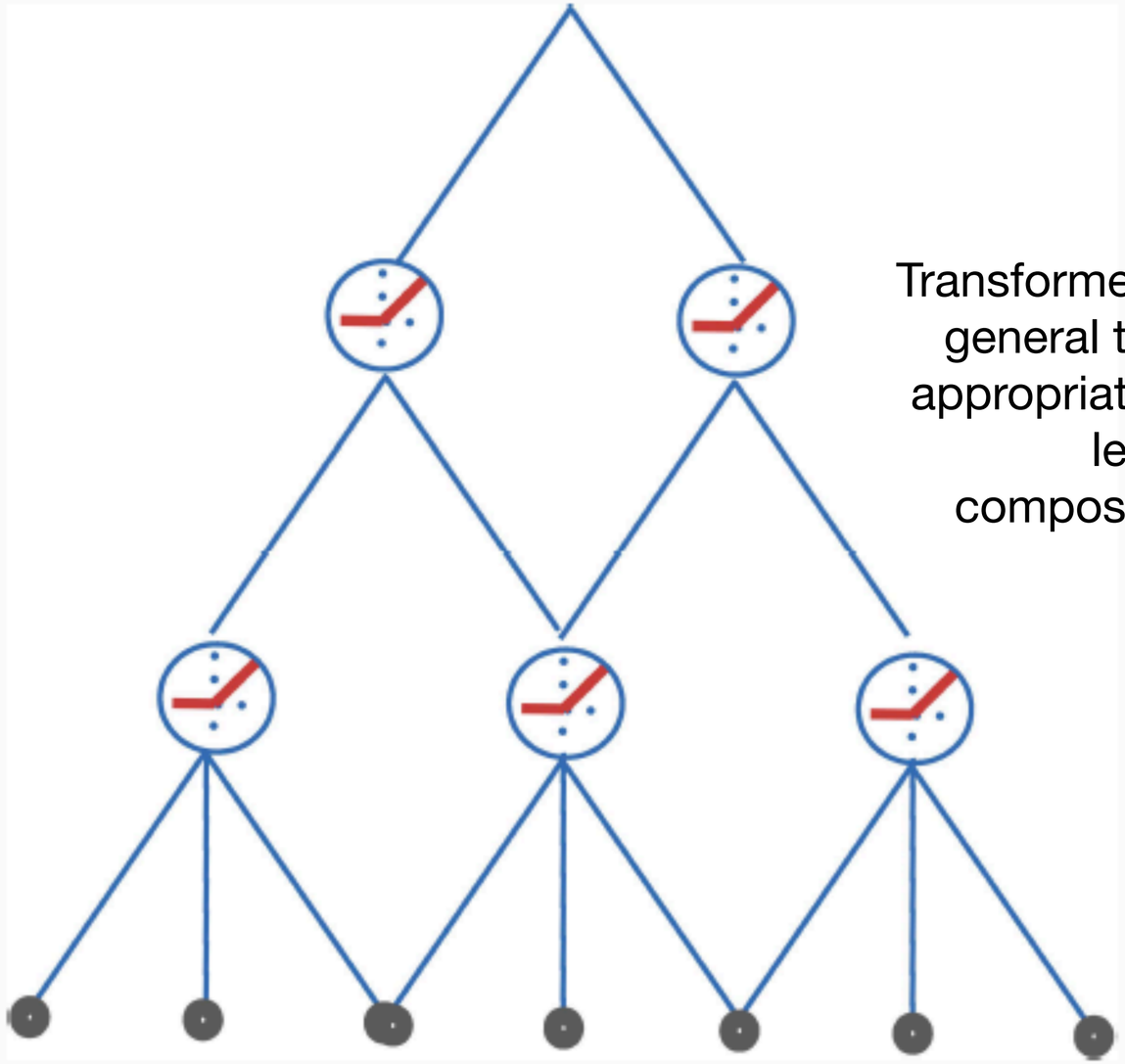
Transformers and Selfattention: a conjecture*



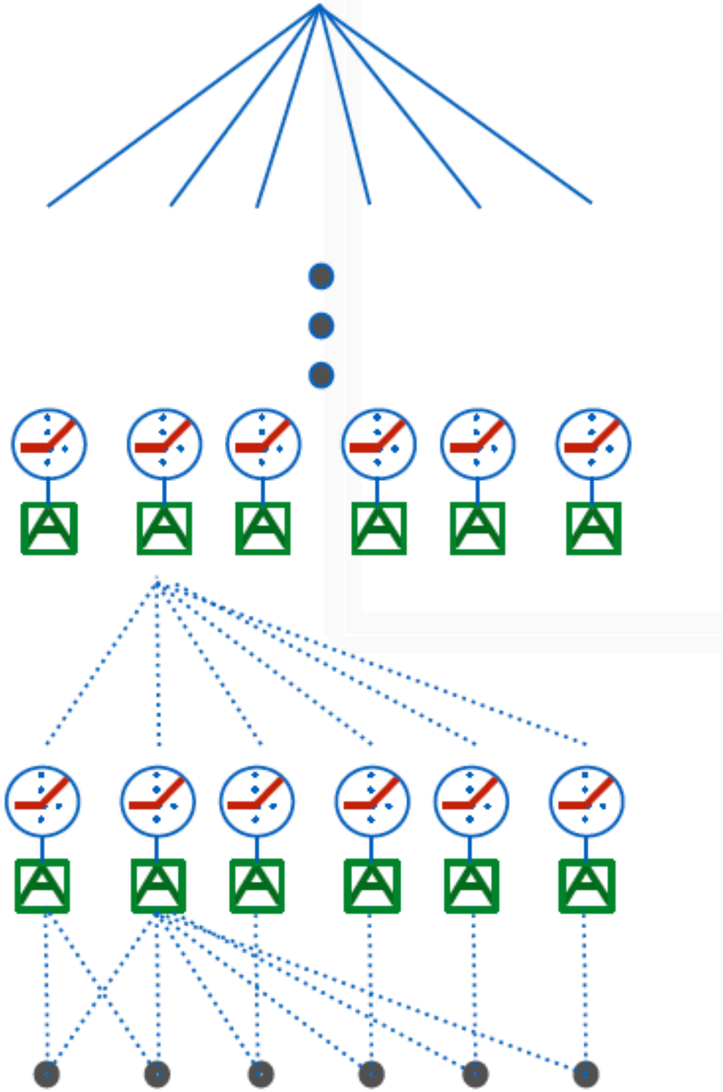
$$node_i = \sum_j softmax(\frac{x_i W_Q (x_j W_K)^T}{norm})$$

$$y = x + MLP(LayerNorm(x + Attention(LayerNorm(x))))$$

A conjecture*: in transformers selfattention selects sparse tokens



Transformers may be a rather general technique to find appropriate sparsity at each level of the compositional hierarchy



Transformers: observations on HyperBF

Neural network architectures:

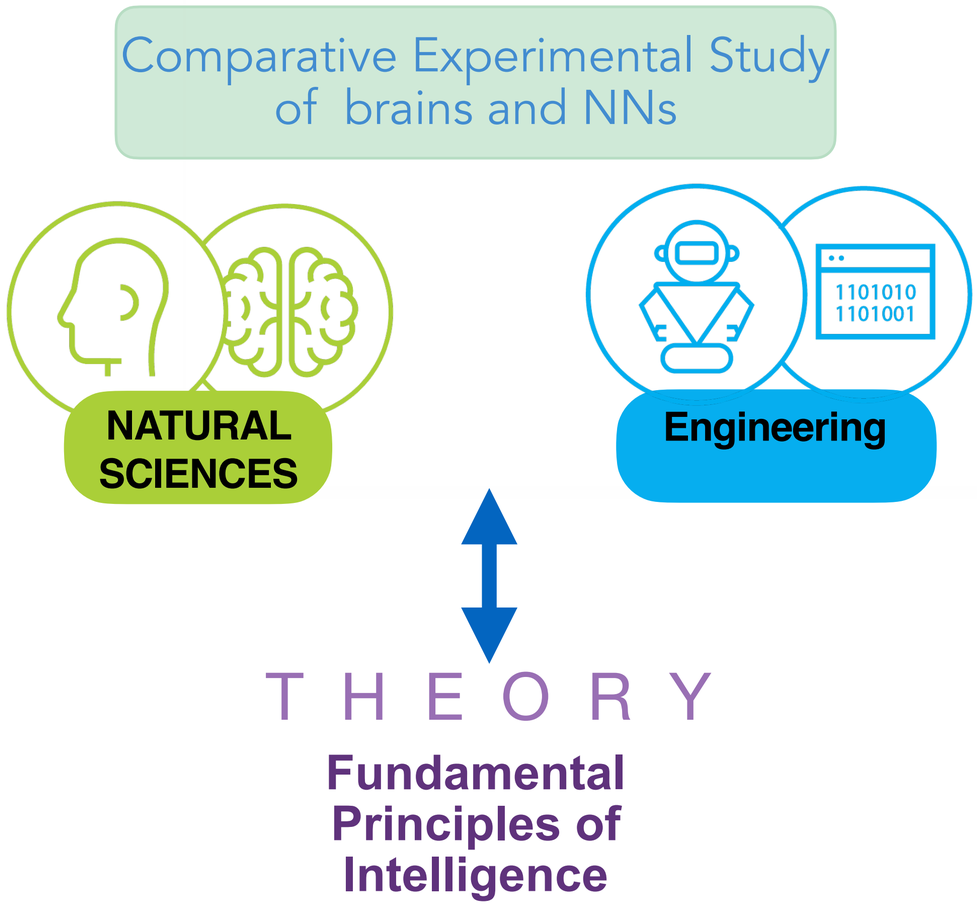
dense, convolutional and residual architectures

Final thoughts

Science and Engineering

We should invest not only in engineering (A+M+M+…do that) but also in science. For the first time in human history we have more than just human intelligence to study. This suggests the following research program articulated in 3 thrusts

A research program, 3 thrusts



- Comparative study of intelligent behavior: what are similarities and differences? What are the roots of the differences?
- Recording from inside the brain and the LLMs, looking for circuits, again similarities and differences.
- Theory looking for fundamental common principles

Tomaso Poggio
Marco Magrini

CERVELLI MENTI ALGORITMI



Il mistero
dell'intelligenza naturale,
gli enigmi
di quella artificiale

Sperling & Kupfer