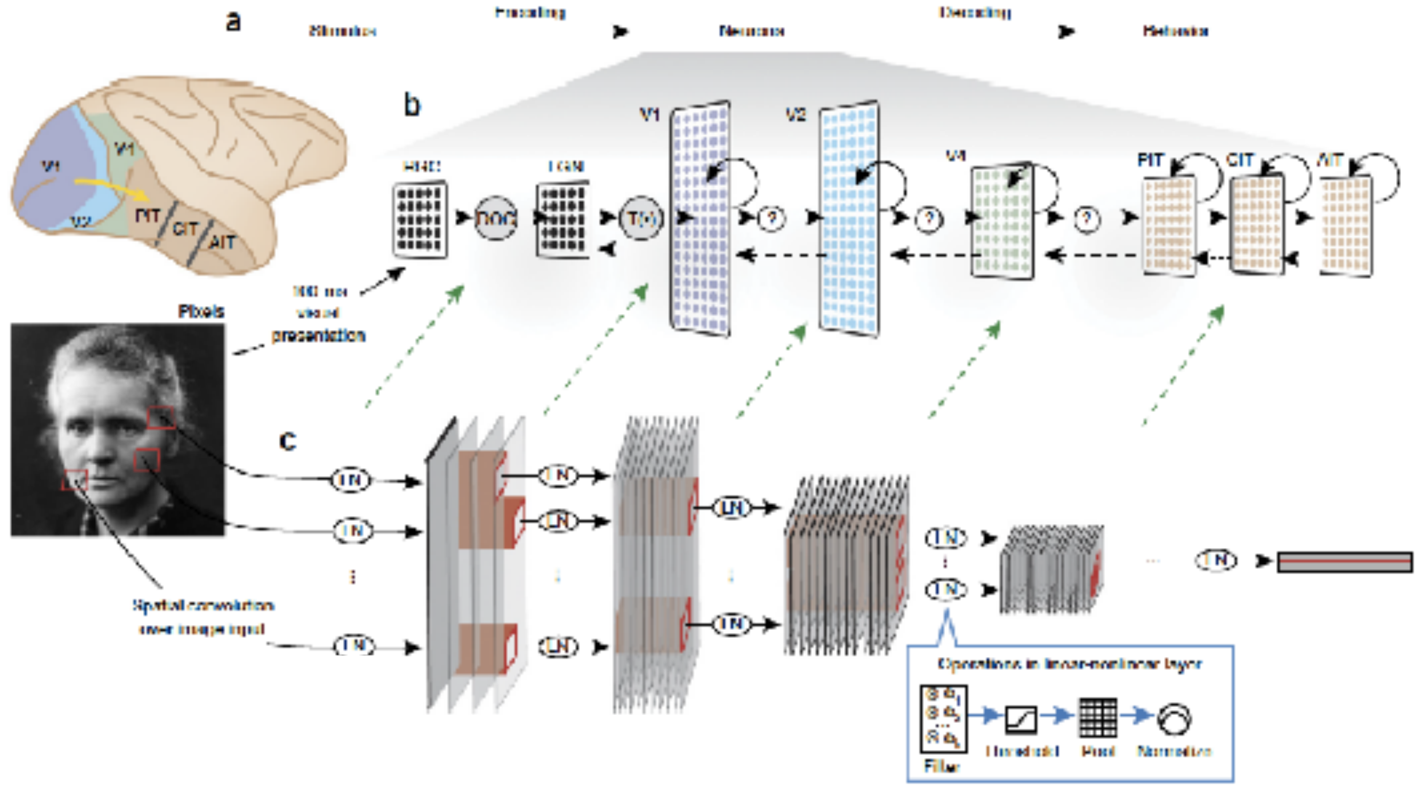
When can Deep Networks avoid the curse of dimensionality and other theoretical puzzles

Tomaso Poggio,
MIT, CBMM

CENTER FOR Brains Minds+ Machines

CBMM's focus is
the <u>Science</u> and the Engineering of Intelligence

We aim to make progress in understanding intelligence, that is in understanding how the brain makes the mind, how the brain works and how to build intelligent machines. We believe that the science of intelligence will enable better engineering of intelligence.



CENTER FOR Brains Minds+ Machines

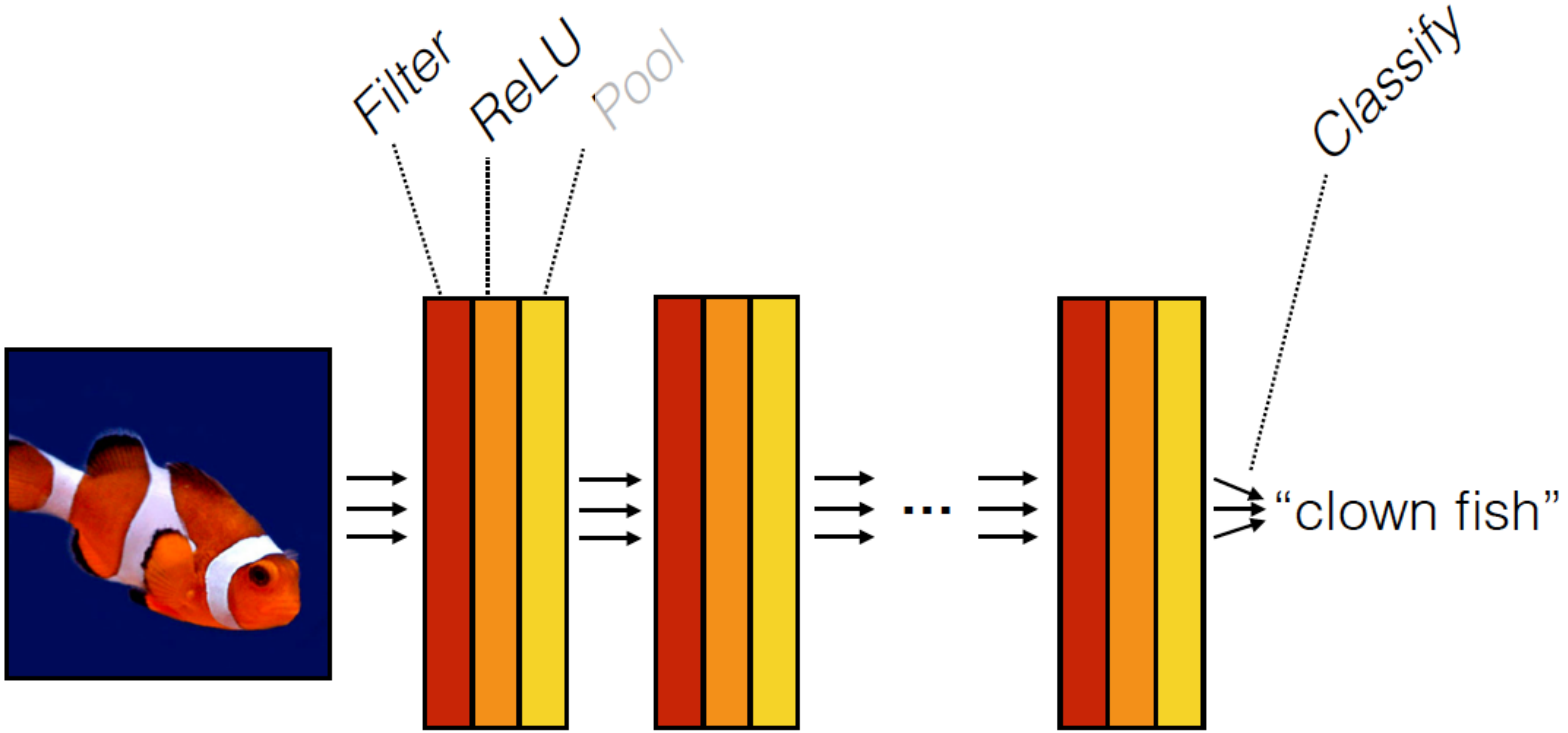
Key role of Machine learning: history

CBMM: one of the motivations

Key recent advances
in the engineering of intelligence
have their roots
in basic research on the brain

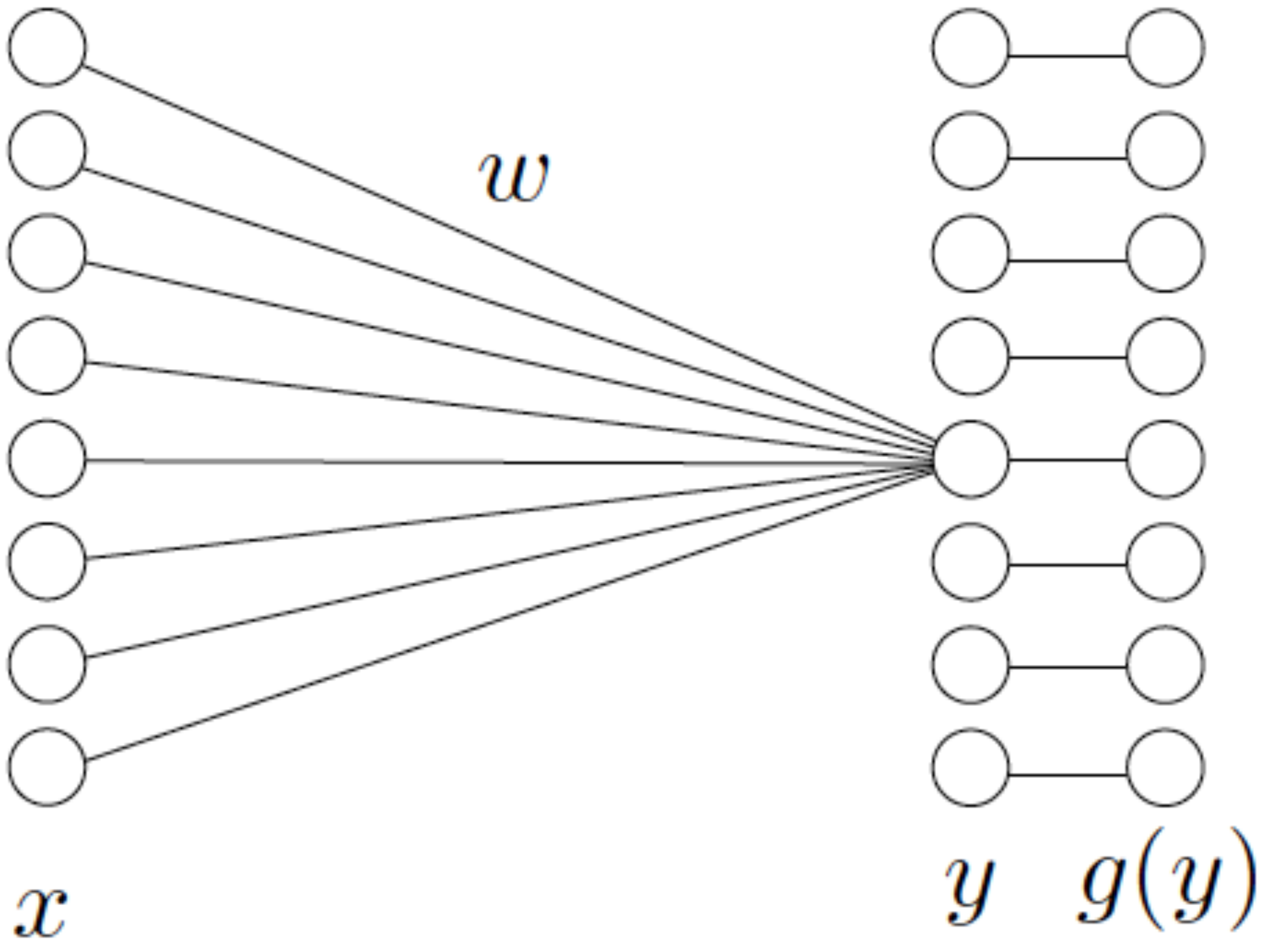
It is time for
a theory of deep learning

Computation in a neural net

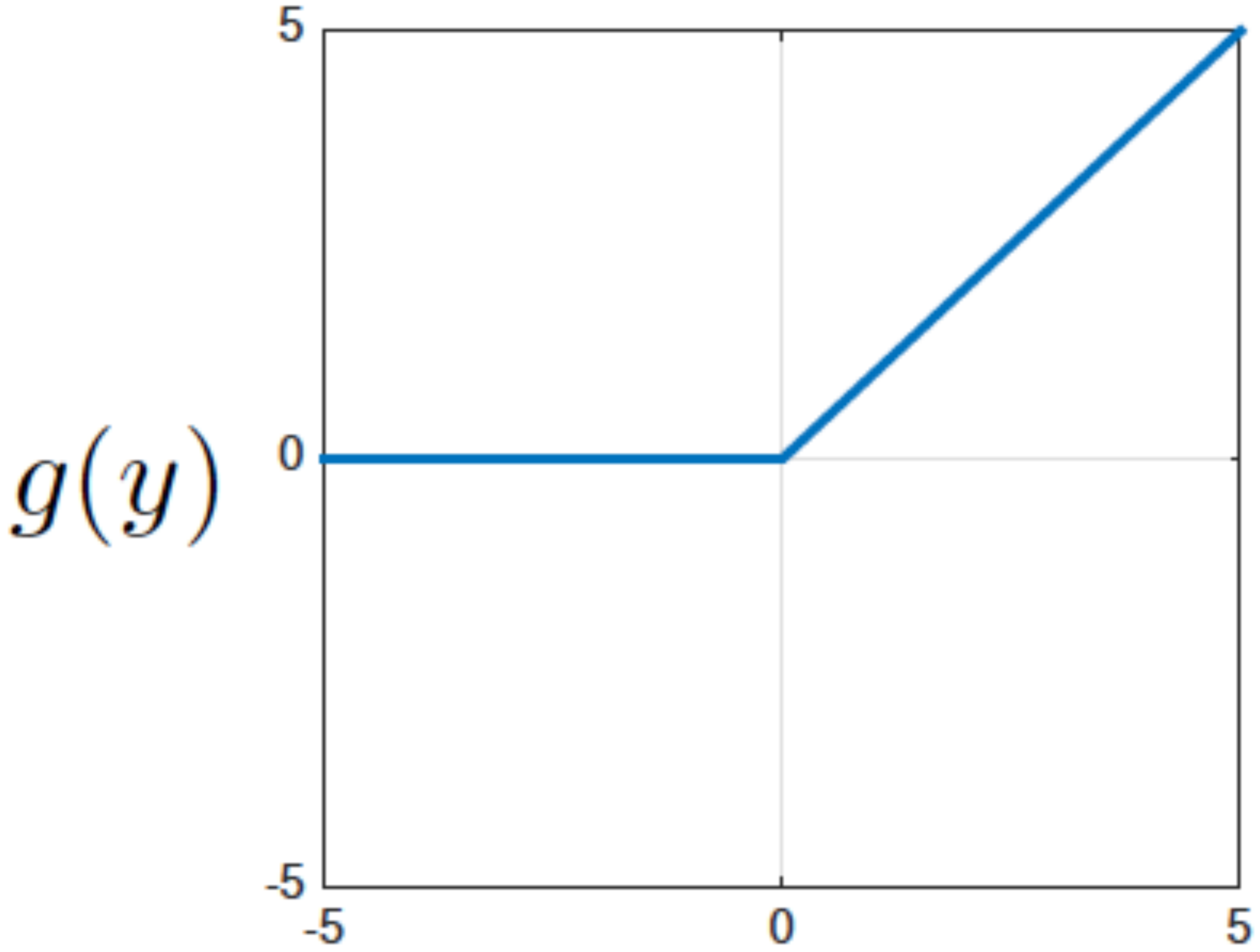


$$f(\mathbf{x}) = f_L(\ldots f_2(f_1(\mathbf{x})))$$

Computation in a neural net

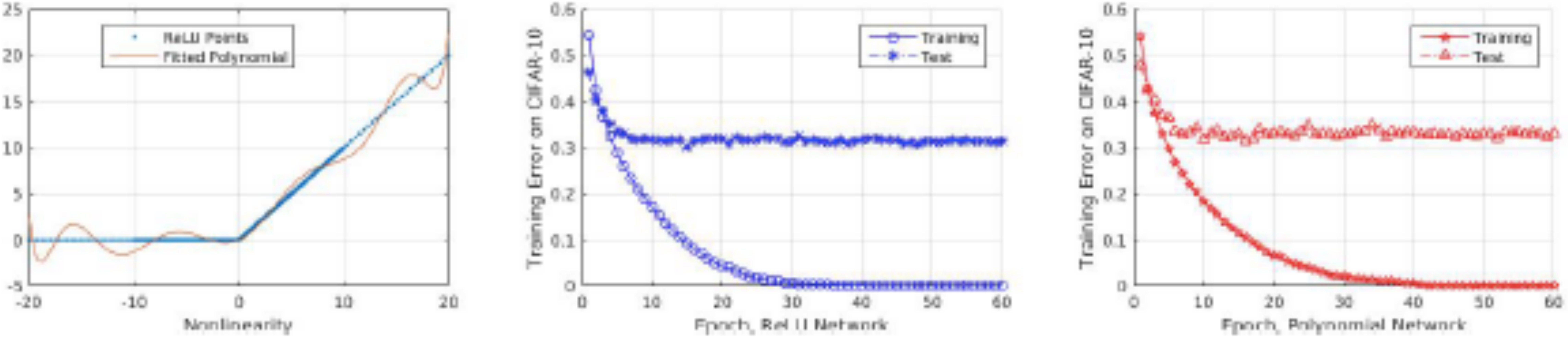


Rectified linear unit (ReLU)



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

RELU approximatinion by univariate polynomial preserves deep nets properties



CENTER FOR Brains Minds + Machines

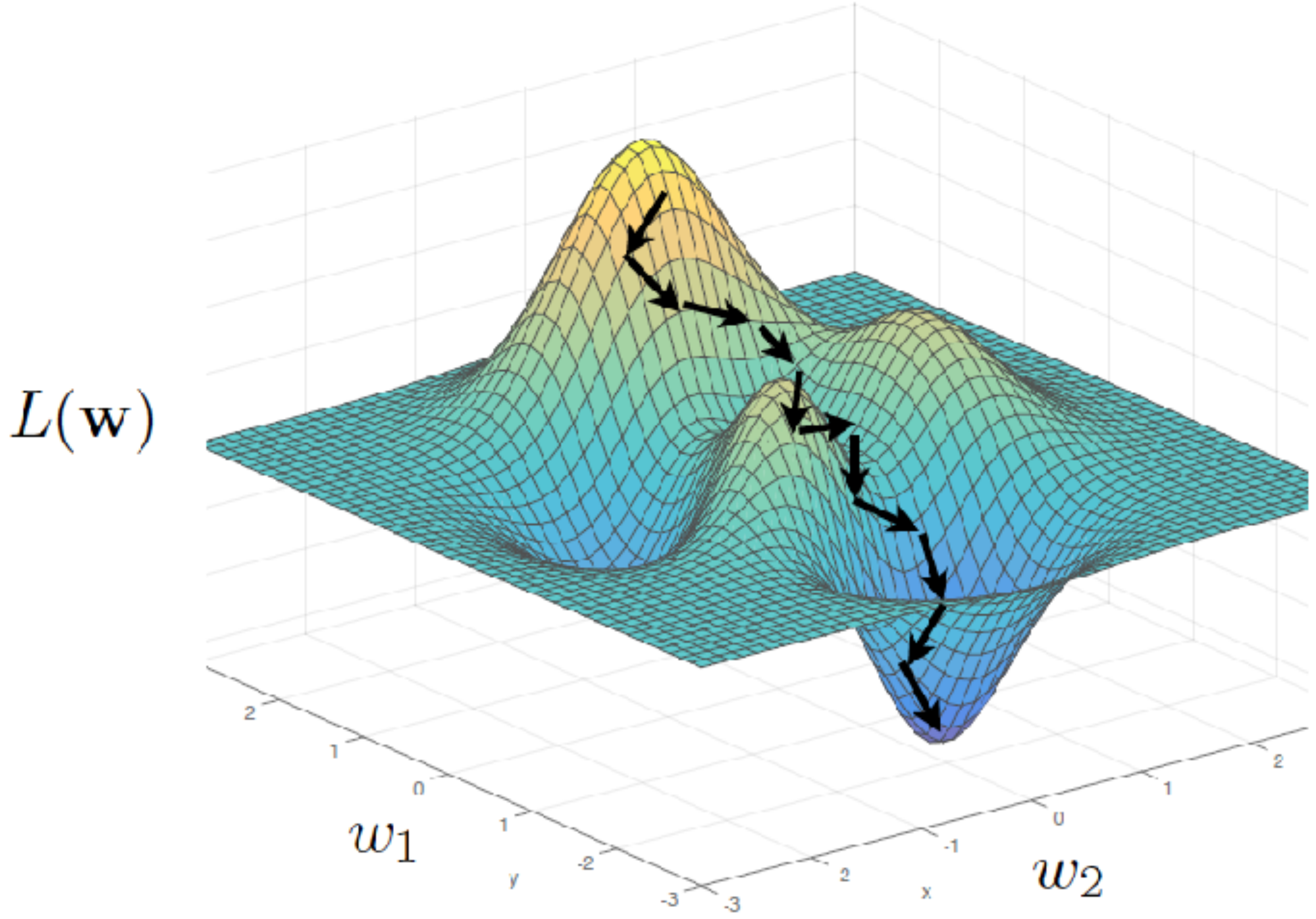
Gradient descent

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

One iteration of gradient de

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

learning rate



Deep Networks:Three theory questions

- *Approximation Theory:* When and why are deep networks better than shallow networks?

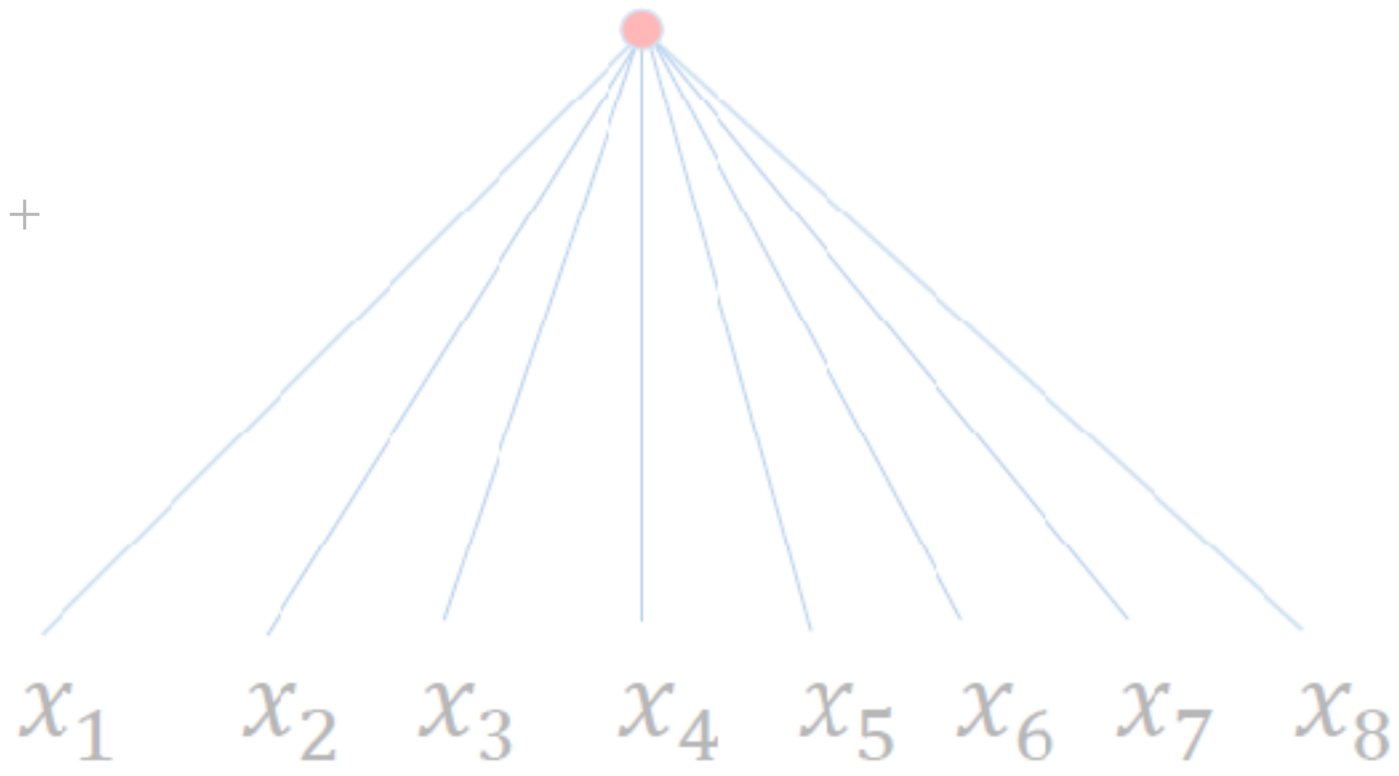
- *Optimization:* What is the landscape of the empirical risk?

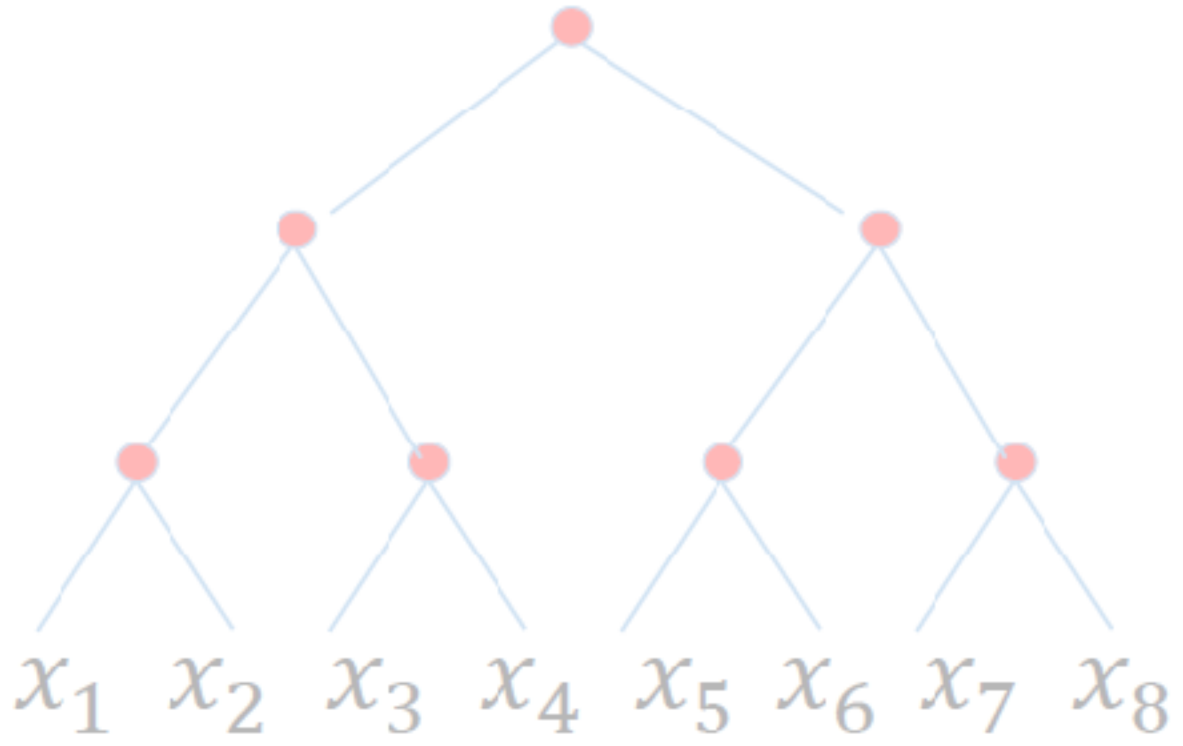
- *Learning Theory:* How can deep learning not overfit?

Theory I:
Why and when are deep networks better than shallow networks?

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

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





Theorem (informal statement)

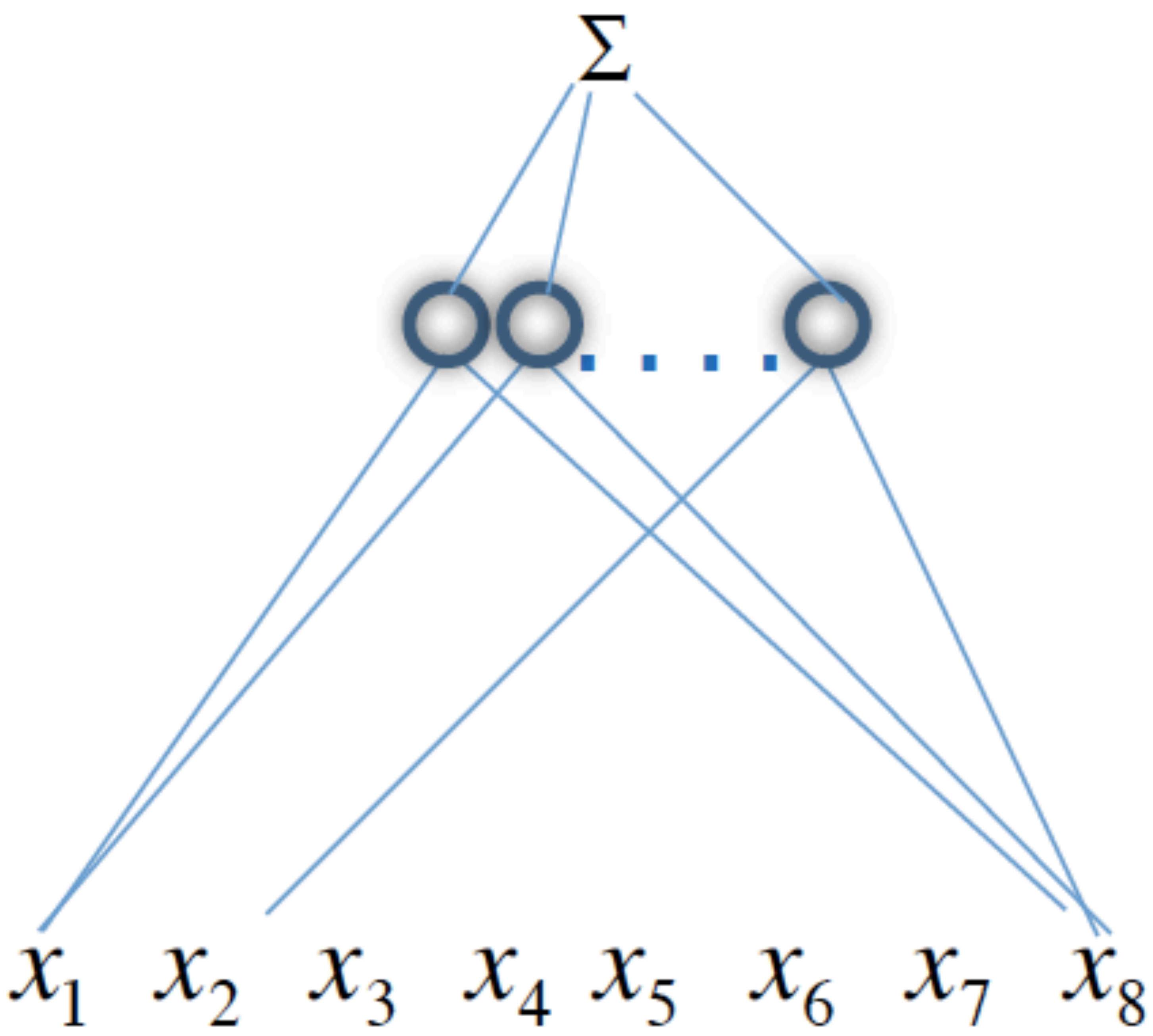
Suppose that a function of d variables is compositional . Both shallow and deep network can approximate f equally well. The number of parameters of the shallow network depends exponentially on d as $O(\varepsilon^{-d})$ with the dimension whereas for the deep network dance is dimension independent, i.e. $O(\varepsilon^{-2})$

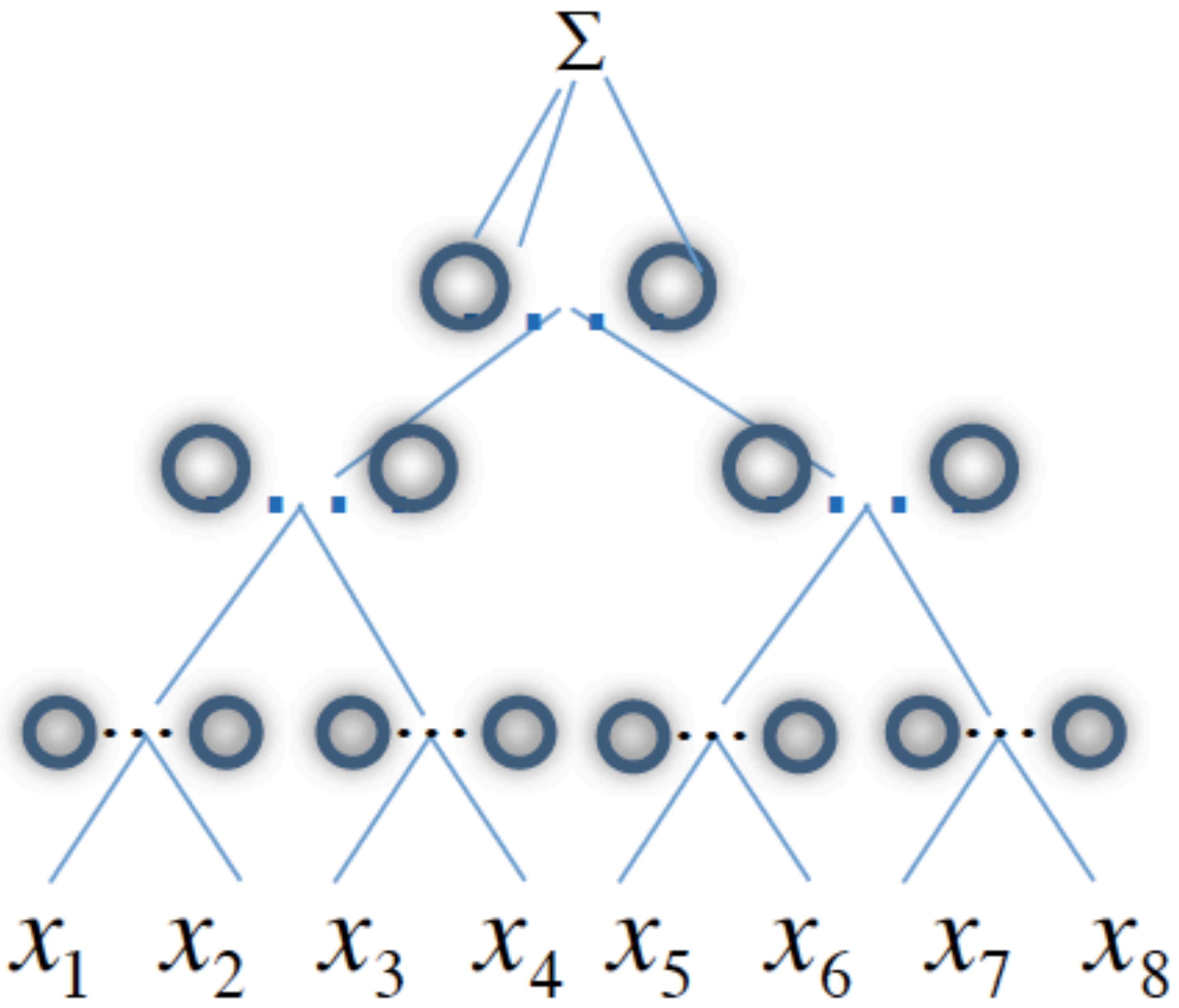
CENTER FOR Brains Minds + Machines

Mhaskar, Poggio, Liao, 2016

Deep and shallow networks: 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, ….

Classical learning theory and Kernel Machines (Regularization in RKHS)

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

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).

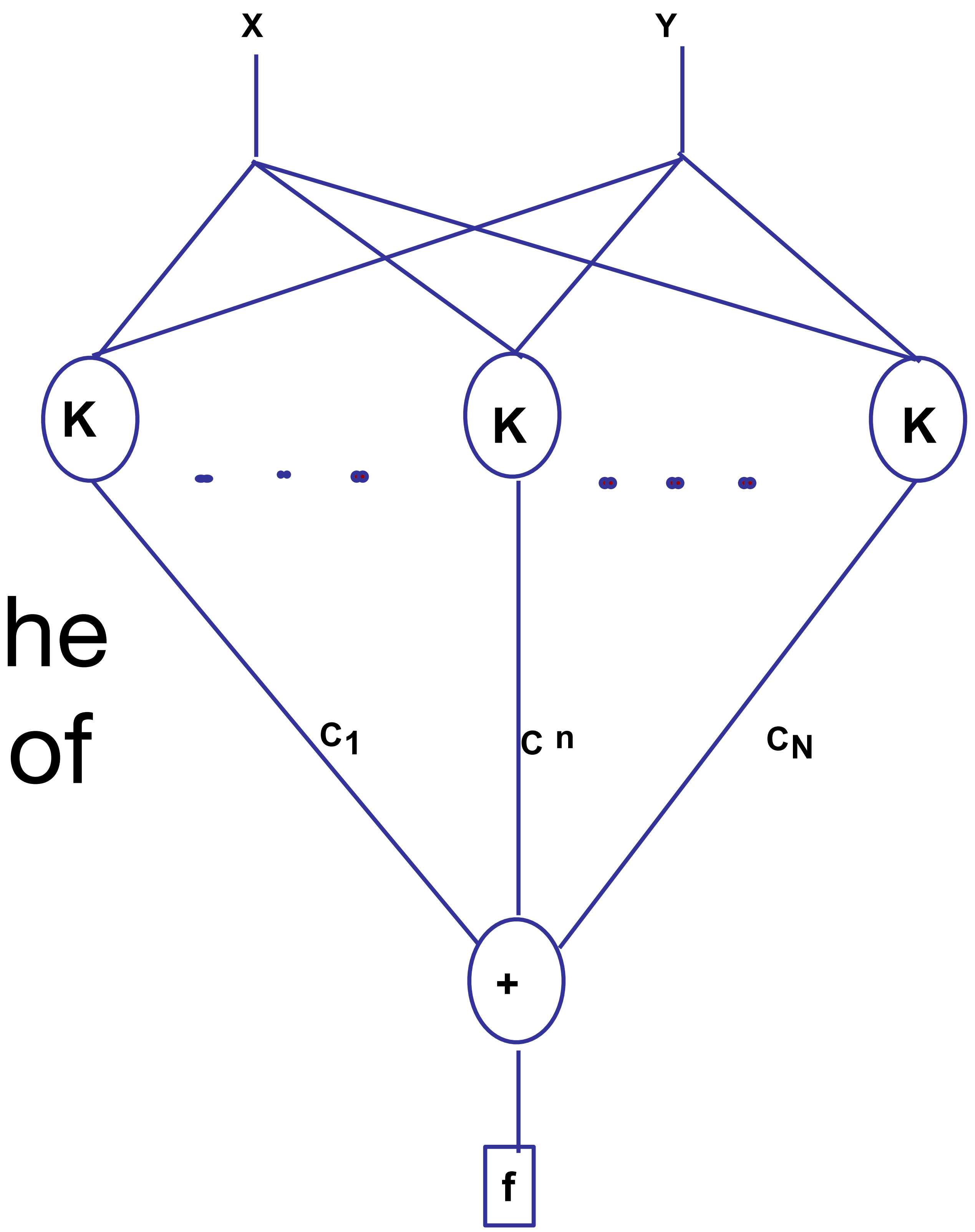
Mhaskar, Poggio, Liao, 2016

Classical kernel machines are equivalent to shallow networks

Kernel machines…

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

can be "written" as shallow networks: the value of K corresponds to the "activity" of the "unit" for the input and the correspond to "weights"



Curse of dimensionality

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

Curse of dimensionality

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

CENTER FOR Brains Minds+ Machines

Mhaskar, Poggio, Liao, 2016

Generic functions

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

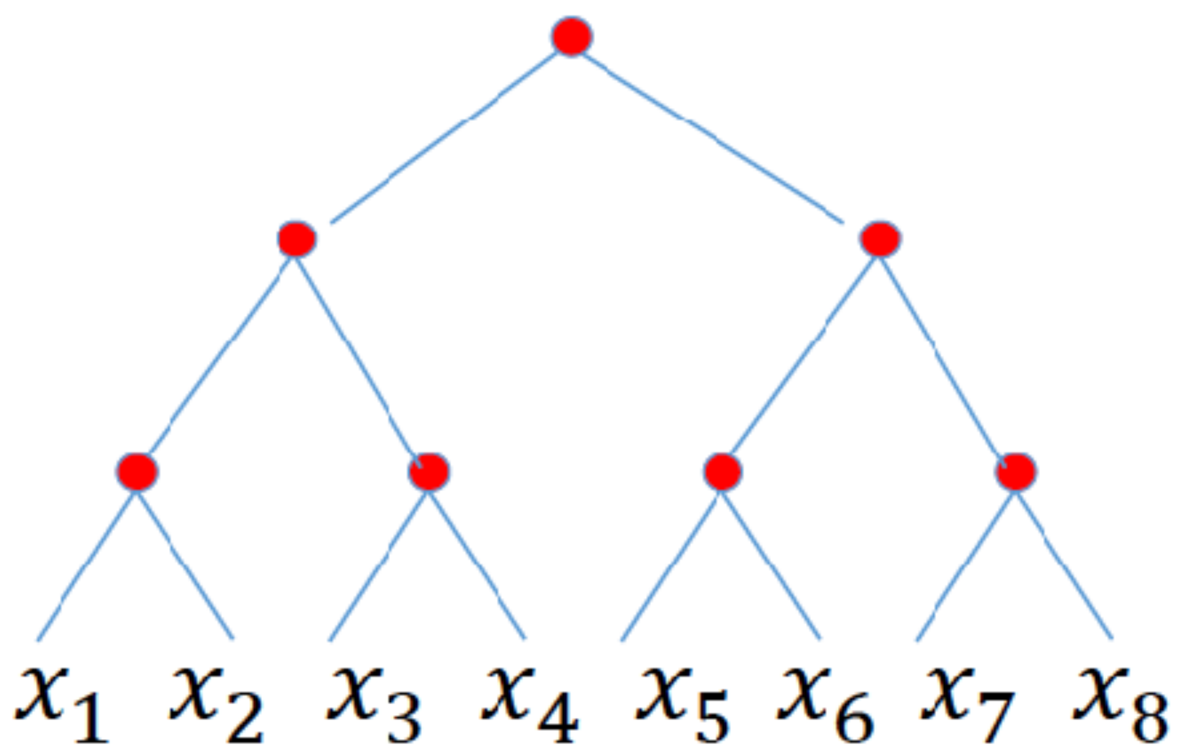
Compositional functions

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

Mhaskar, Poggio, Liao, 2016

Hierarchically local compositionality

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



Theorem (informal statement)

Suppose that a function of d variables is hierarchically, locally, compositional . Both shallow and deep network can approximate f equally well. The number of parameters of the shallow network depends exponentially on d as $O(\varepsilon^{-d})$ with the dimension whereas for the deep network dance is $O(d\varepsilon^{-2})$

CENTER FOR Brains Minds+ Machines

Mhaskar, Poggio, Liao, 2016

Proof

Proof To prove Theorem 2, we observe that each of the constituent functions being in W_m^2, (1) applied with $n = 2$ implies that each of these functions can be approximated from $\mathcal{S}_{N,2}$ up to accuracy $\epsilon = cN^{-m/2}$. Our assumption that $f \in W_m^{N,2}$ 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 $\|h - P\| \leq \epsilon$, $\|h_1 - P_1\| \leq \epsilon$ and $\|h_2 - P_2\| \leq \epsilon$, then $\|h(h_1, h_2) - P(P_1, P_2)\| = \|h(h_1, h_2) - h(P_1, P_2) + h(P_1, P_2) - P(P_1, P_2)\| \leq \|h(h_1, h_2) - h(P_1, P_2)\| + \|h(P_1, P_2) - P(P_1, P_2)\| \leq c\epsilon$ by Minkowski inequality. Thus

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

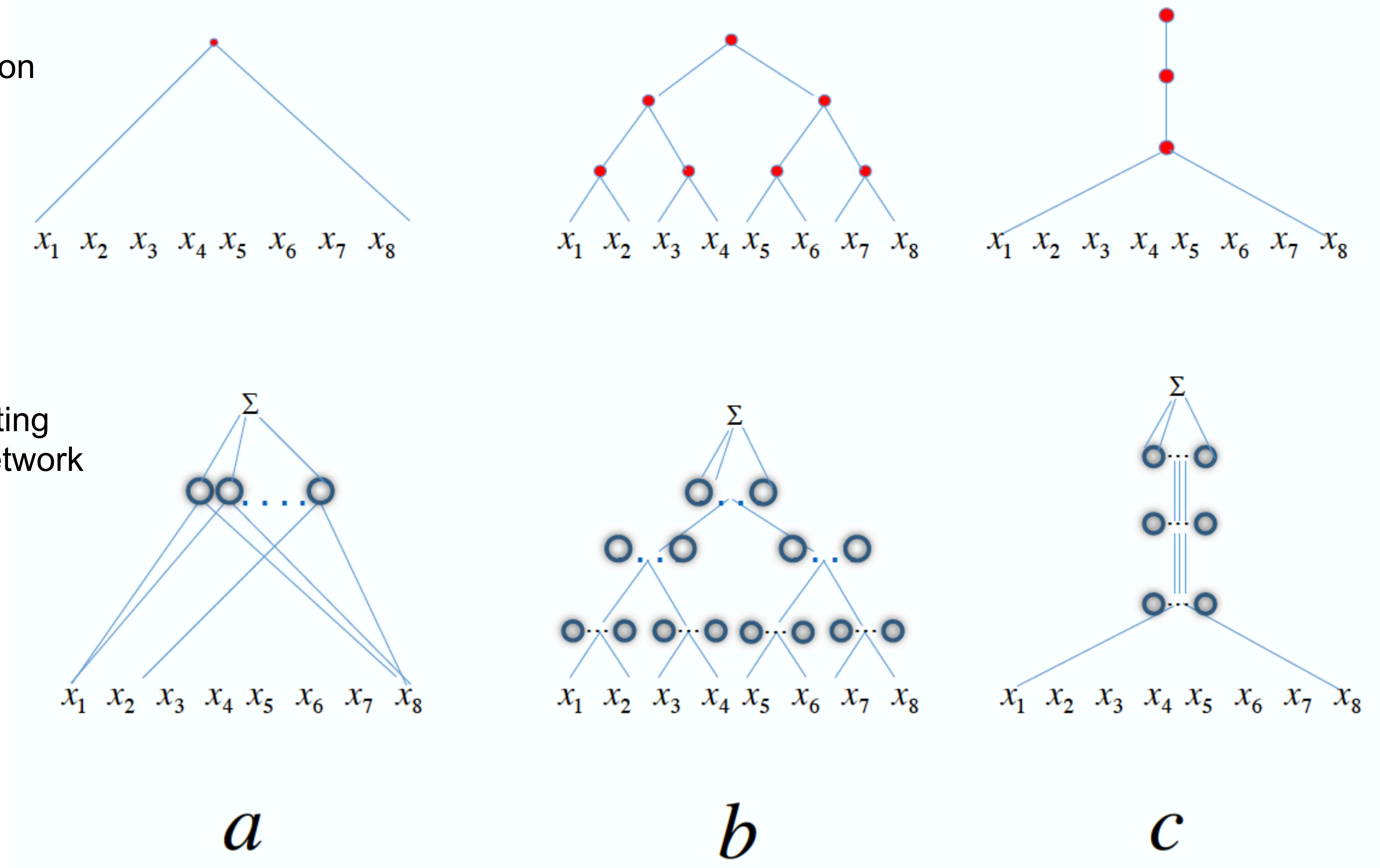
for some constant $c > 0$ independent of the functions involved. This, together with the fact that there are $(n-1)$ nodes, leads to (6). $\square$

CENTER FOR Brains Minds+ Machines

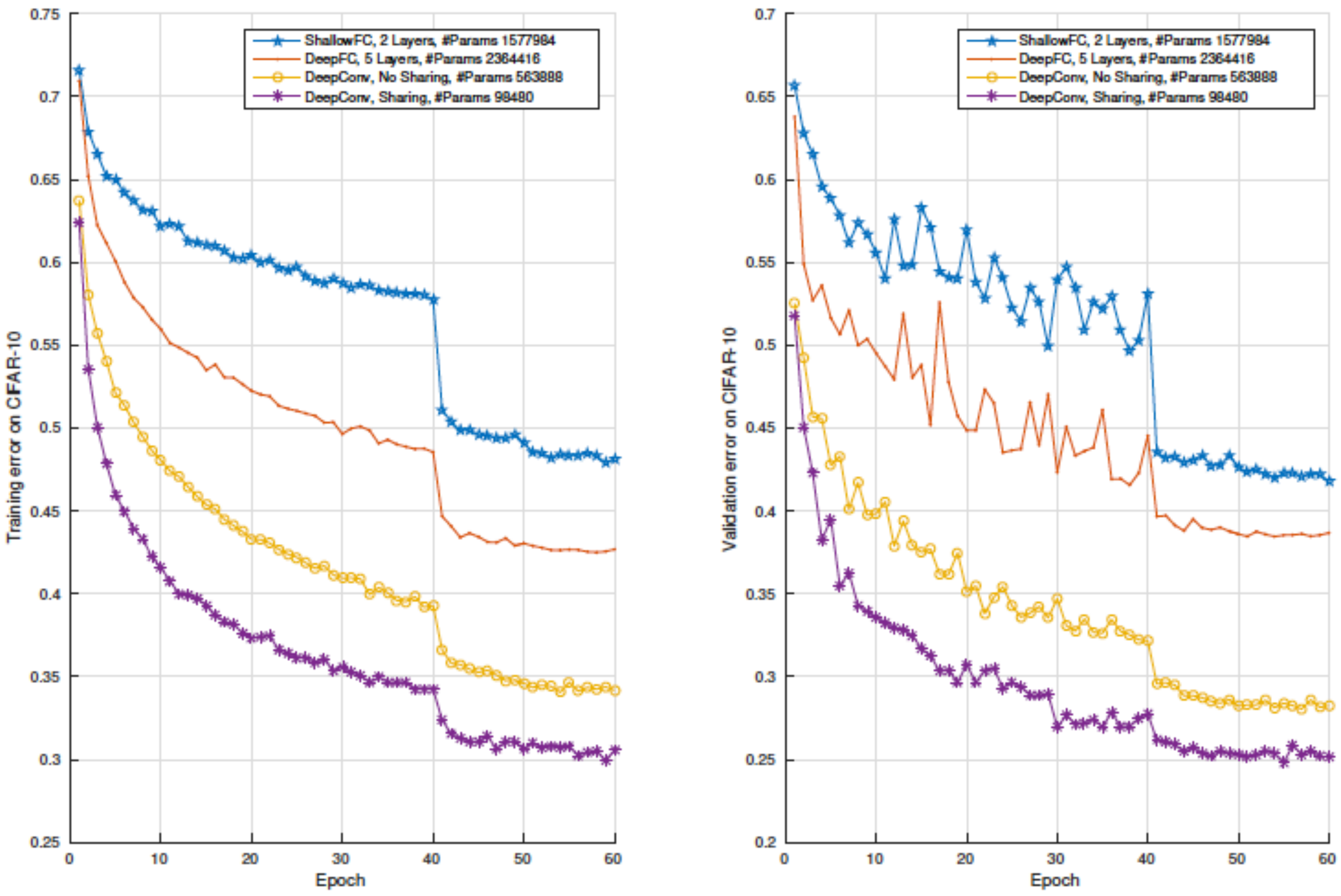
Microstructure of compositionality

target function

approximating function/network



Locality of constituent functions is key: CIFAR



Remarks

Old results on Boolean functions are closely related

- A classical **theorem [Sipser, 1986; Hastad, 1987]** shows that deep circuits are more efficient in representing certain Boolean functions than shallow circuits. Hastad proved that highly-variable functions (in the sense of having high frequencies in their Fourier spectrum) in particular the parity function cannot even be decently approximated by small constant depth circuits

Lower Bounds

- The main **result of [Telgarsky, 2016, Colt]** says that there are functions with many oscillations that cannot be represented by shallow networks with linear complexity but can be represented with low complexity by deep networks.

- Older examples exist: consider a function which is a linear combination of n tensor product Chui–Wang spline wavelets, where each wavelet is a tensor product cubic spline. It was shown by Chui and Mhaskar that is impossible to implement such a function using a shallow neural network with a sigmoidal activation function using O(n) neurons, but a deep network with the activation function $(x_+)^2$ do so. In this case, as we mentioned, there is a formal proof of a gap between deep and shallow networks. Similarly, Eldan and Shamir show other cases with separations that are exponential in the input dimension.

Open problem: why compositional functions are important for perception?

They seem to occur in computations on text, speech, images…why?

Conjecture (with) Max Tegmark

The locality of the hamiltonians of physics induce compositionality in natural signals such as images

or

The connectivity in our brain implies that our perception is limited to compositional functions

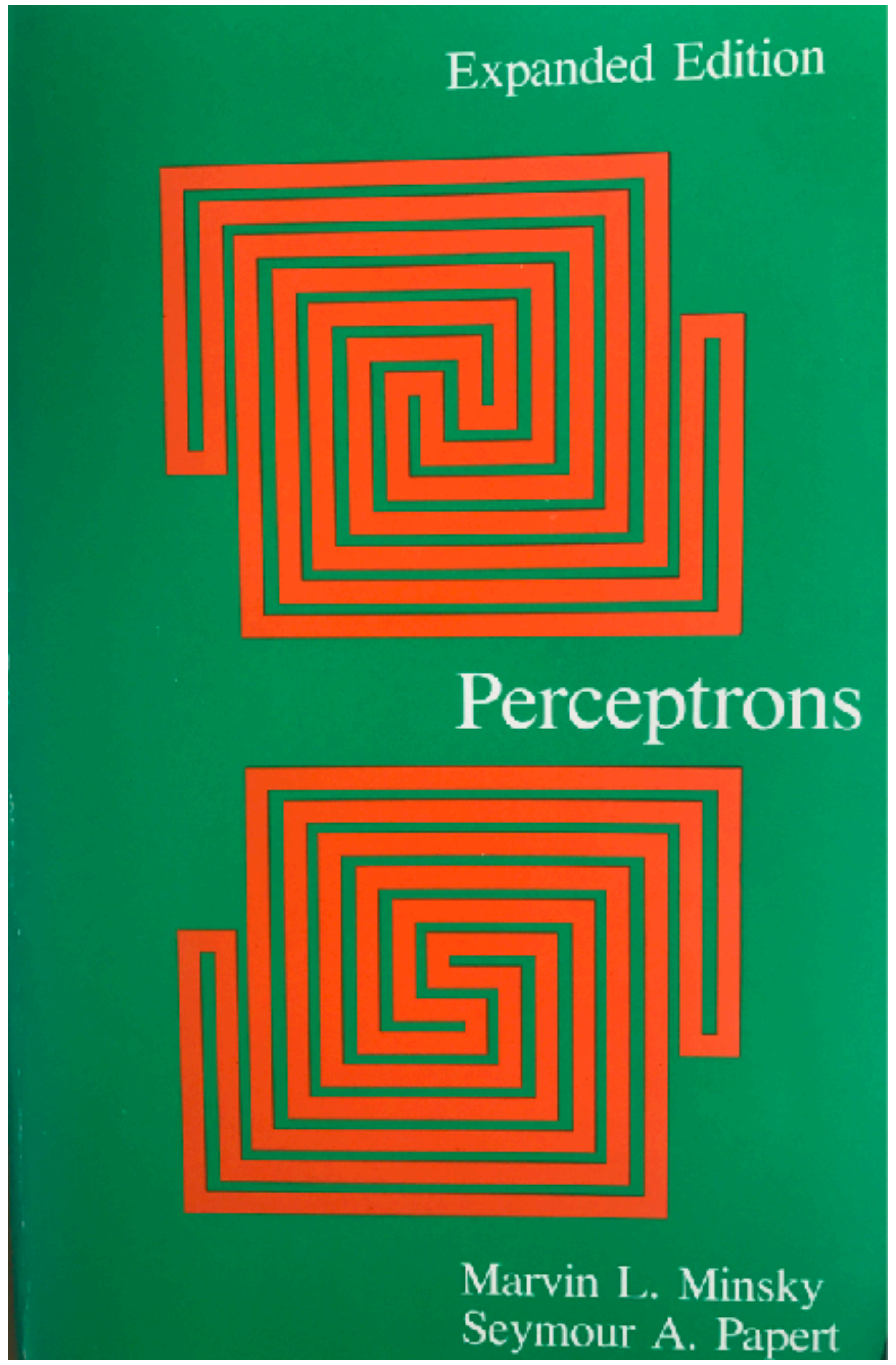
Why are compositional functions important?

Which one of these reasons:
Physics?
Neuroscience? <===
Evolution?

Locality of Computation

What is special about locality of computation?

Locality in “space”?
Locality in “time”?



Deep Networks:Three theory questions

- *Approximation Theory:* When and why are deep networks better than shallow networks?

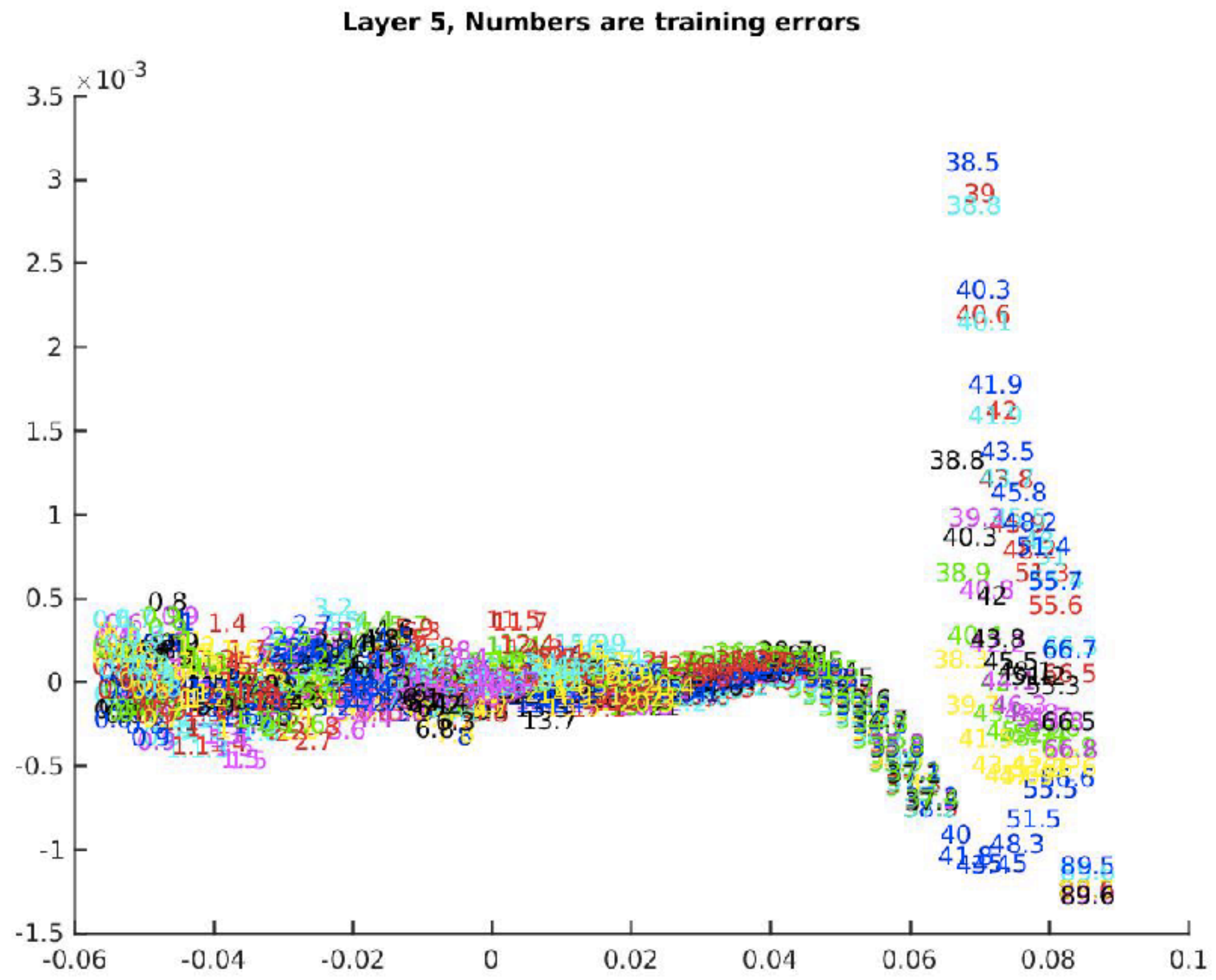
- *Optimization:* What is the landscape of the empirical risk?

- *Learning Theory:* How can deep learning not overfit?

Theory II:
What is the Landscape of the empirical risk?

Observation

Replacing the RELUs with univariate polynomial approximation, Bezout theorem implies that the system of polynomial equations corresponding to zero empirical error has a very large number of degenerate solutions. The global zero-minimizers correspond to flat minima in many dimensions (generically, unlike local minima). Thus SGD is biased towards finding global minima of the empirical risk.



CENTER FOR Brains Minds+ Machines

Liao, Poggio, 2017

Bezout theorem

$$p(x_i) - y_i = 0 \quad \text{for } i = 1,...,n$$

The set of polynomial equations above with *k= degree of p(x)* has a number of distinct zeros (counting points at infinity, using projective space, assigning an appropriate multiplicity to each intersection point, and excluding degenerate cases) equal to

$$Z = k^n$$

the product of the degrees of each of the equations. As in the linear case, when the system of equations is underdetermined – as many equations as data points but more unknowns (the weights) – the theorem says that there are an infinite number of global minima, under the form of Z regions of zero empirical error.

Global and local zeros

$f(x_i) - y_i = 0 \;\; \text{for } i = 1,\ldots,n$ $\qquad$ n equations in W unknowns with $W >> n$

$\nabla_w \sum_{i=1}^{N} (f(x_i) - y_i)^2) = 0$ $\qquad$ W equations in W unknowns

There are a very large number of zero-error minima which are highly degenerate unlike the local non-zero minima.

CENTER FOR Brains Minds + Machines

Langevin equation

$$\frac{df}{dt} = -\gamma_t \nabla V(f(t), z(t)) + \gamma'_t \, dB(t)$$

$$f_{t+1} = f_t - \gamma_n \nabla V(f_t, z_t) + \gamma'_t W_t.$$

with the Boltzmann equation as asymptotic “solution”

$$p(f) \sim \frac{1}{Z} = e^{-\frac{U(x)}{T}}$$

SGD

$$f_{t+1} = f_t - \gamma_t \nabla V(f_t, z_t),$$

$$\nabla V(f_t, z_t) = \frac{1}{|z_t|} \sum_{z \in z_t} \nabla V(f_t, z).$$

We define a noise "equivalent quantity"

$$\xi_t = \nabla V(f_t, z_t) - \nabla I_{S_n}(f_t),$$

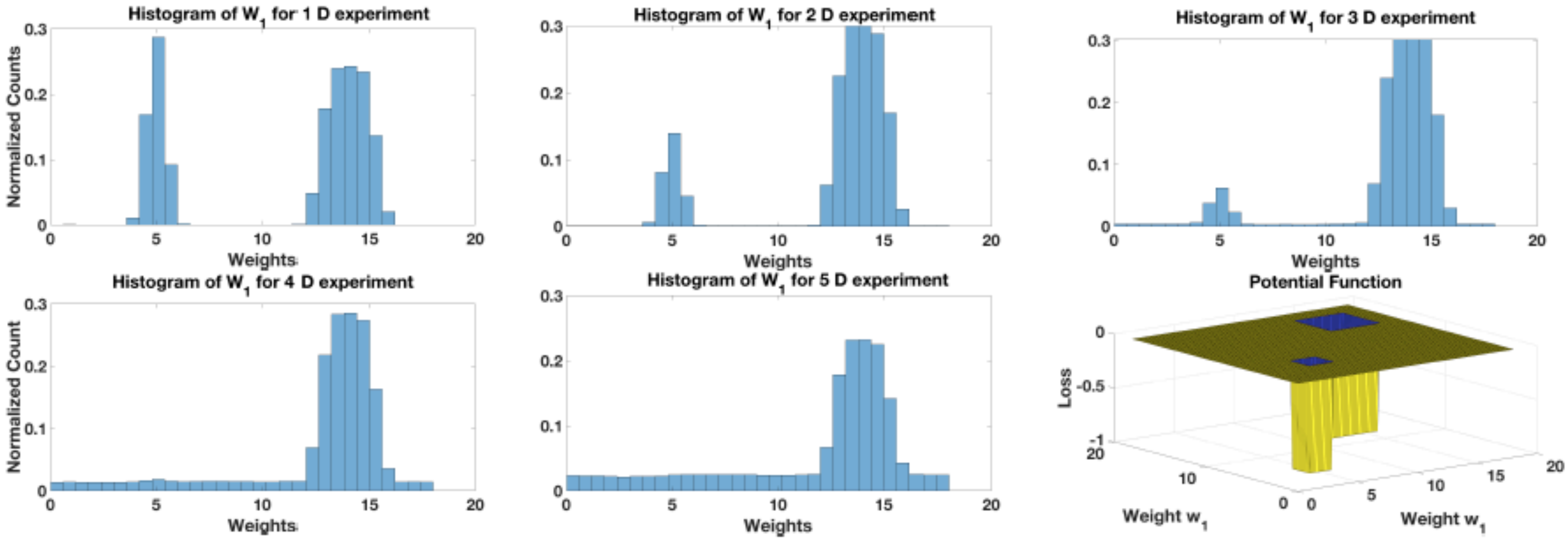
and it is clear that $\mathbb{E}\xi_t = 0$.

We write Equation 6 as

$$f_{t+1} = f_t - \gamma_t(\nabla I_{S_n}(f_t) + \xi_t).$$

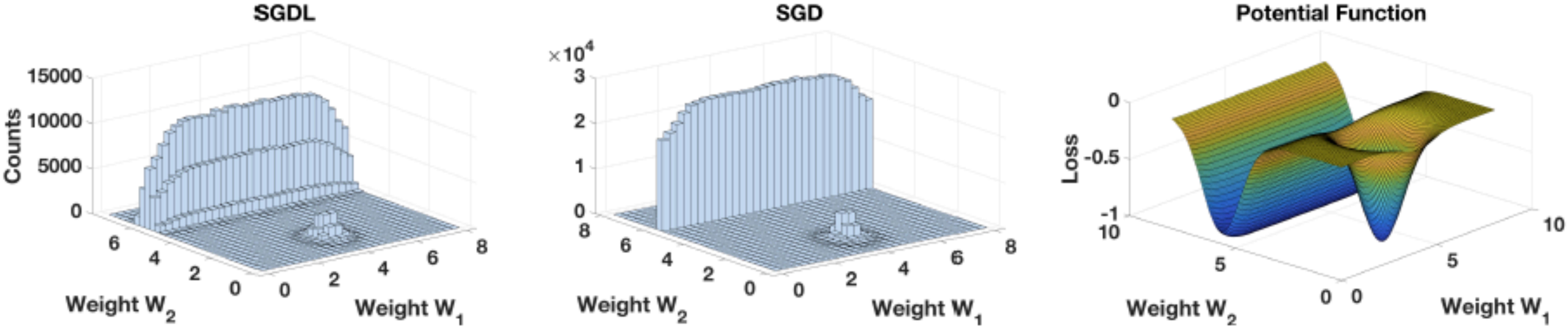
This is an
analogy
NOT a theorem

GDL selects larger volume minima



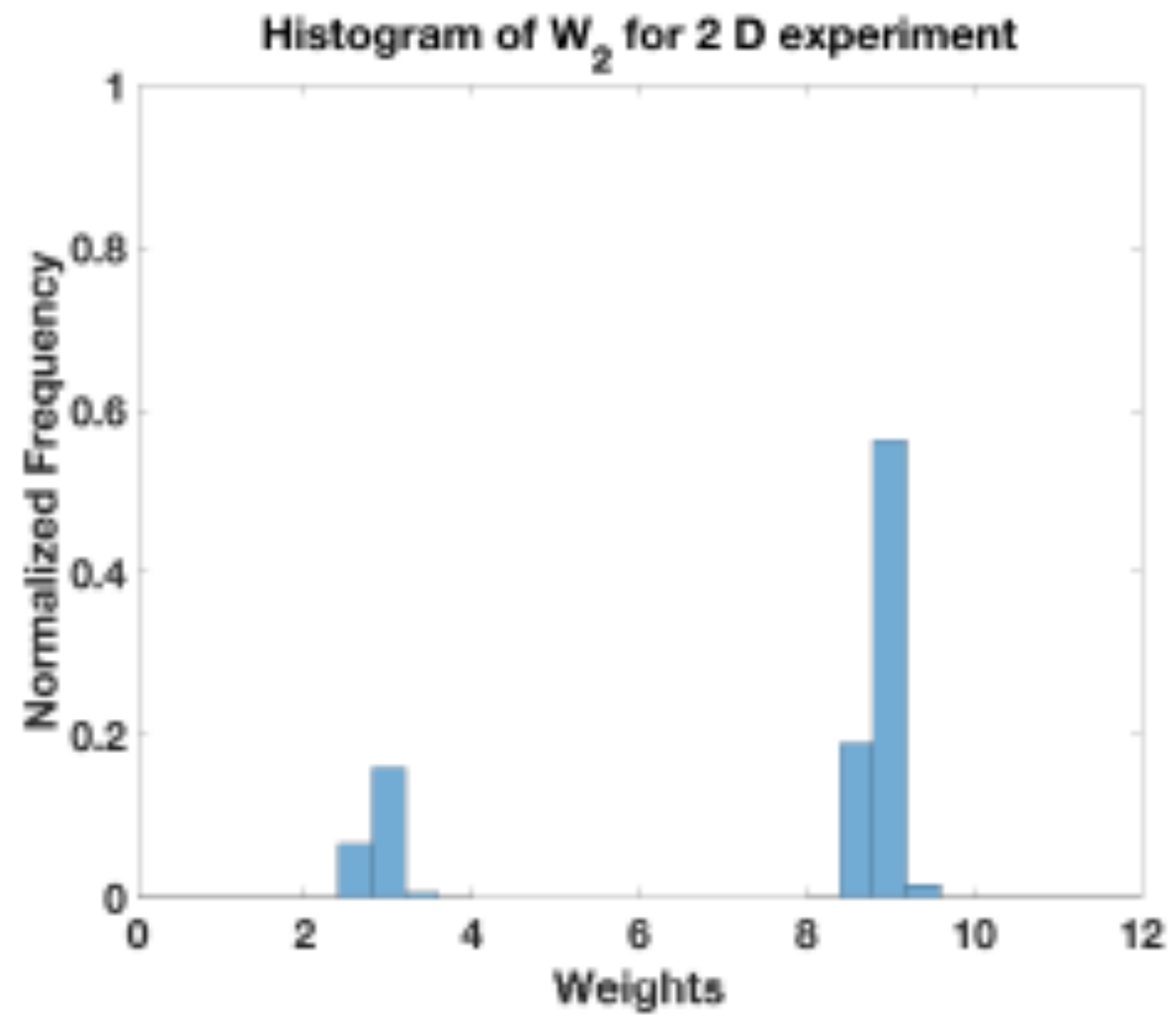
CENTER FOR
Brains
Minds+
Machines

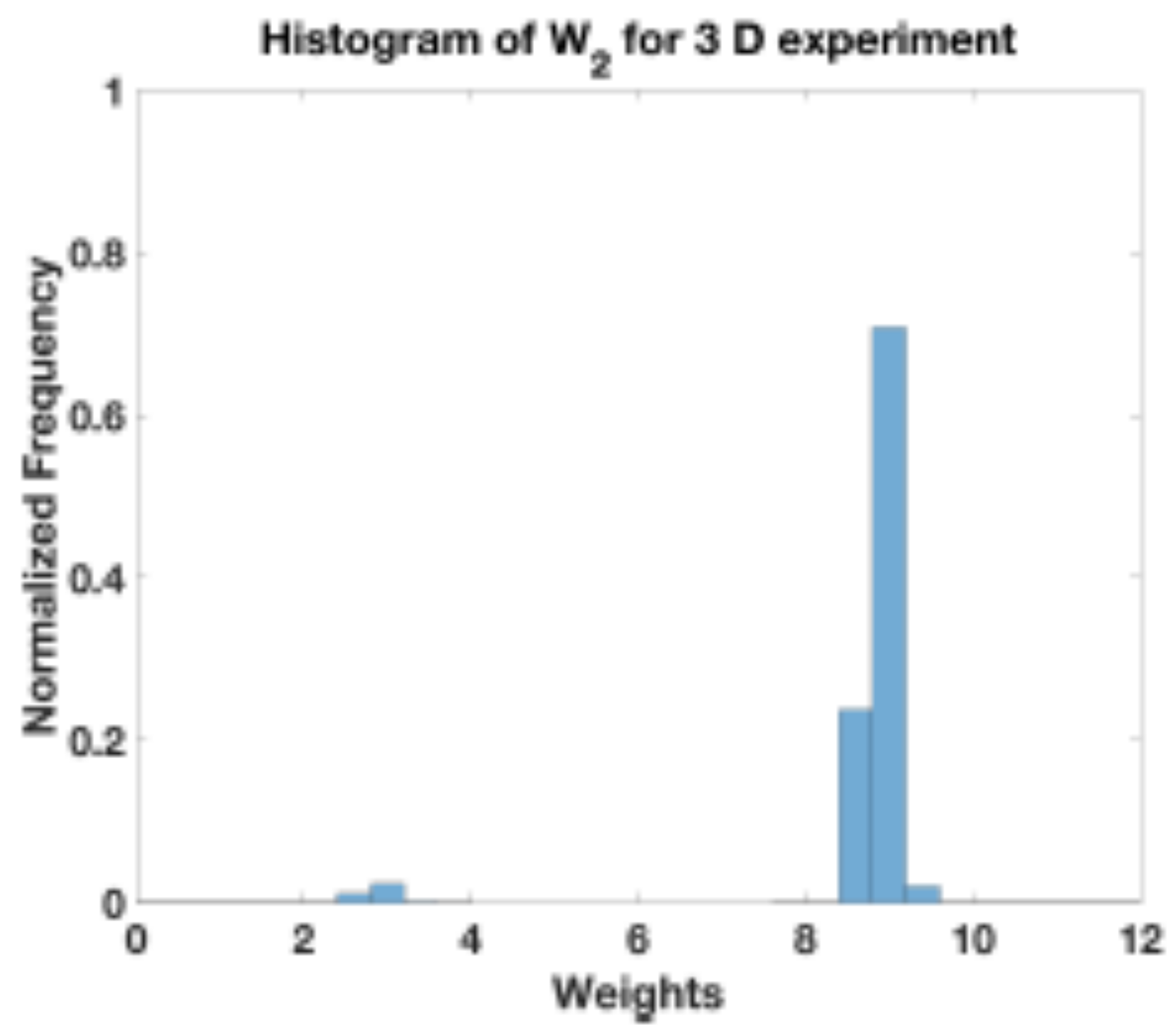
GDL and SGD

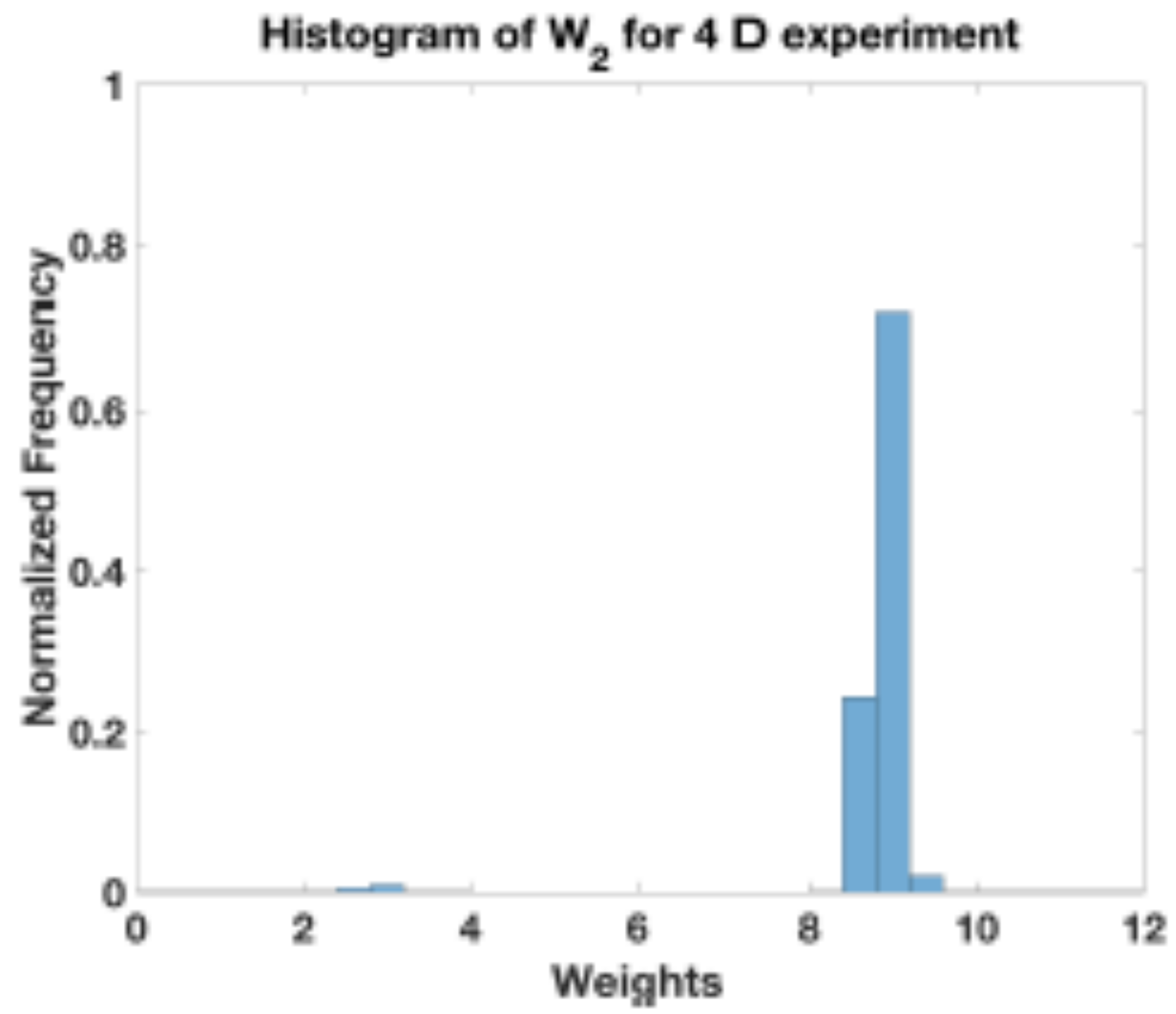


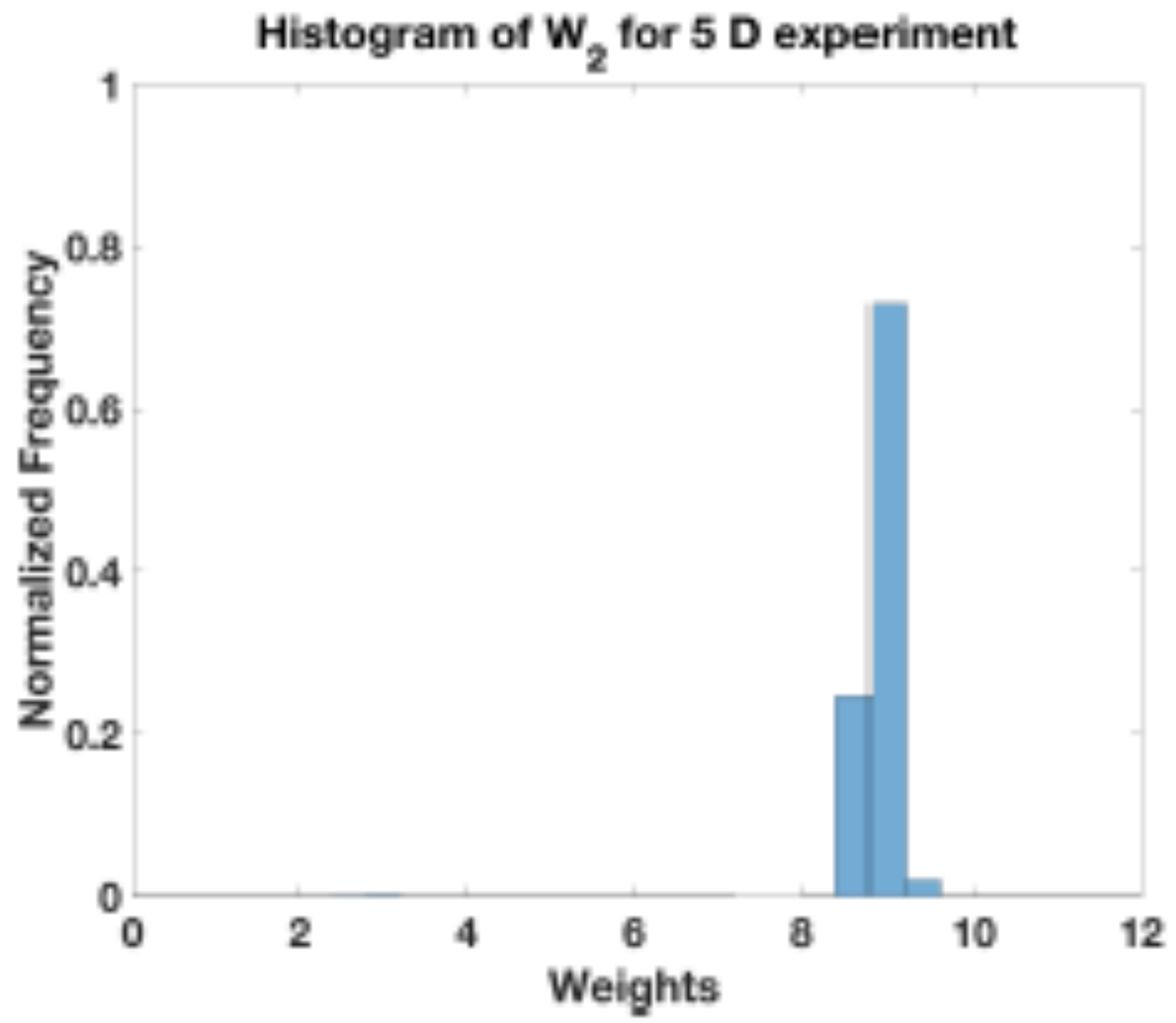
SGDL
Counts
15000
10000
5000
0
Weight W_2
Weight W_1
SGD
$\times 10^4$
3
2
1
0
Weight W_2
Weight W_1
Potential Function
Loss
0
-0.5
-1
Weight W_2
Weight W_1

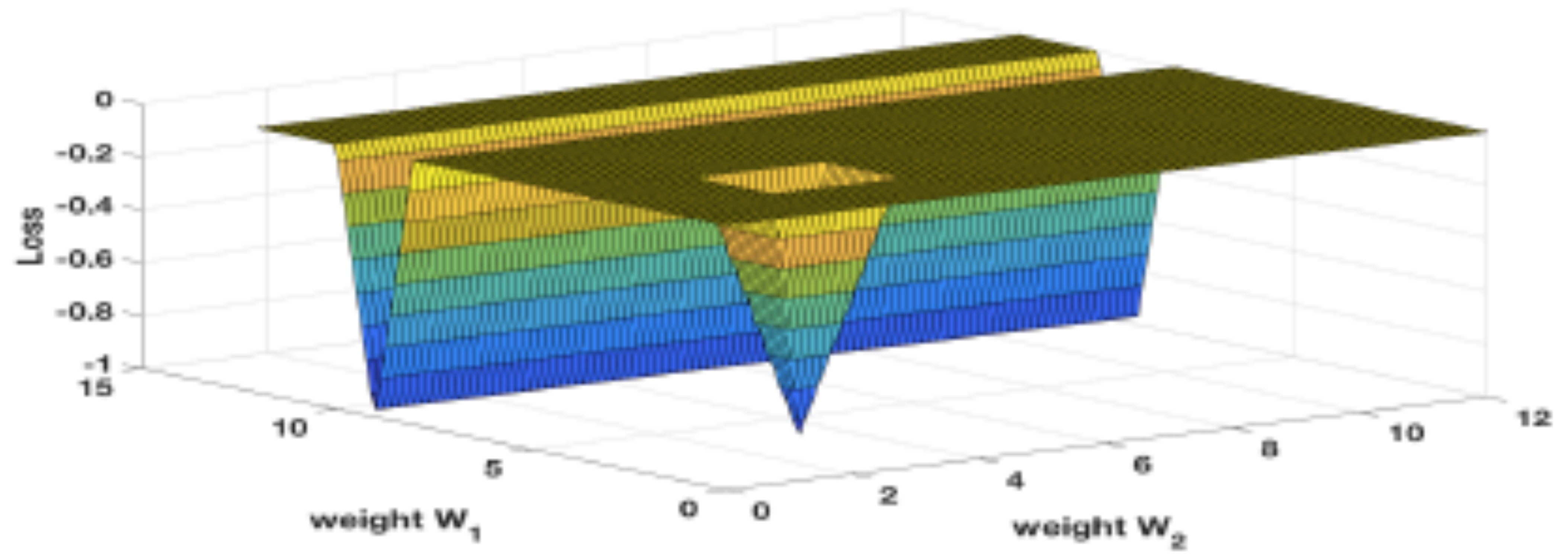
Concentration because of high dimensionality









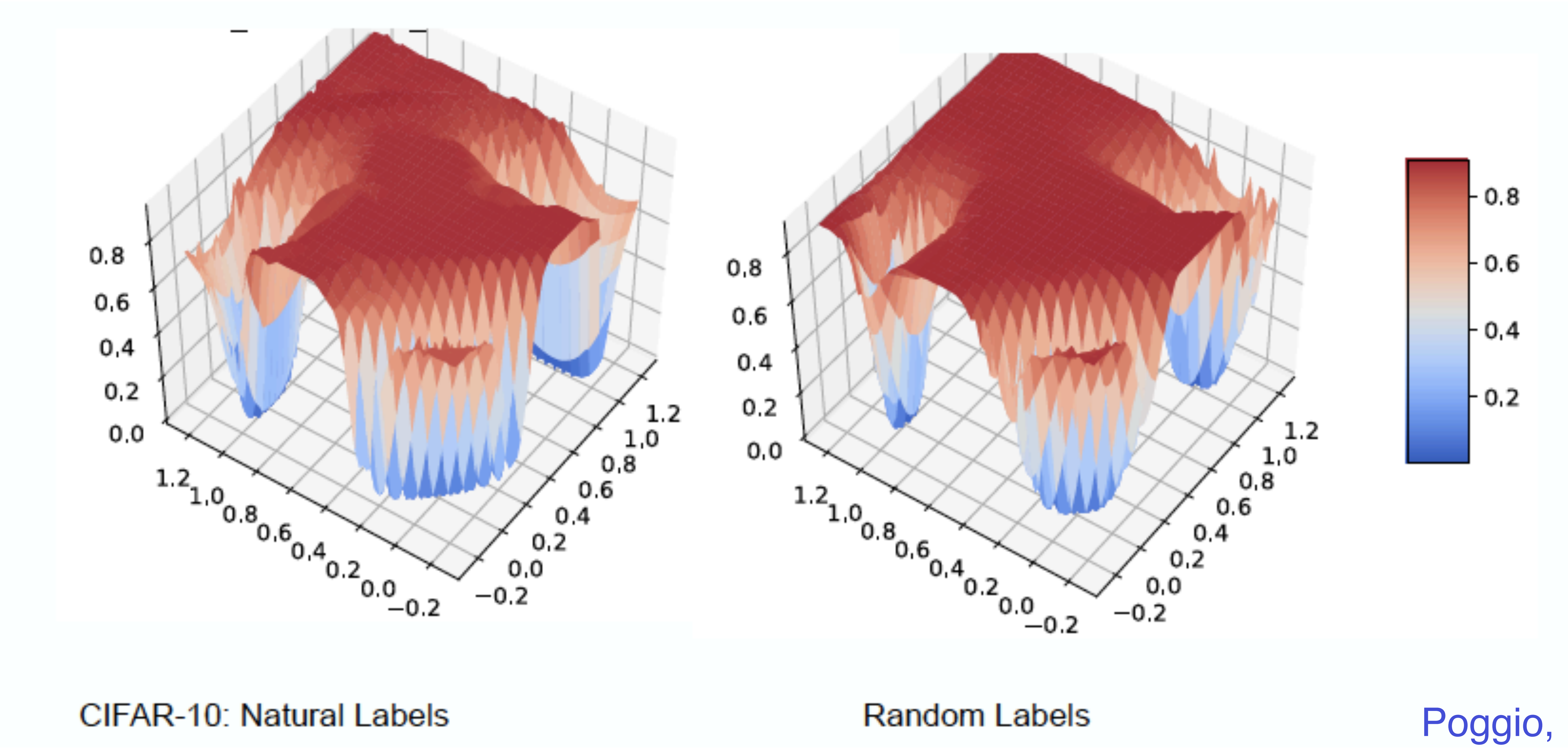


Brains Minds+ Machines

SGDL and SGD observation: summary

- SGDL finds with very high probability large volume, flat zero-minimizers; empirically SGD behaves in a similar way

- Flat minimizers correspond to degenerate zero-minimizers and thus to global minimizers;



Poggio, Rakhlin, Golovitc, Zhang, Liao, 2017

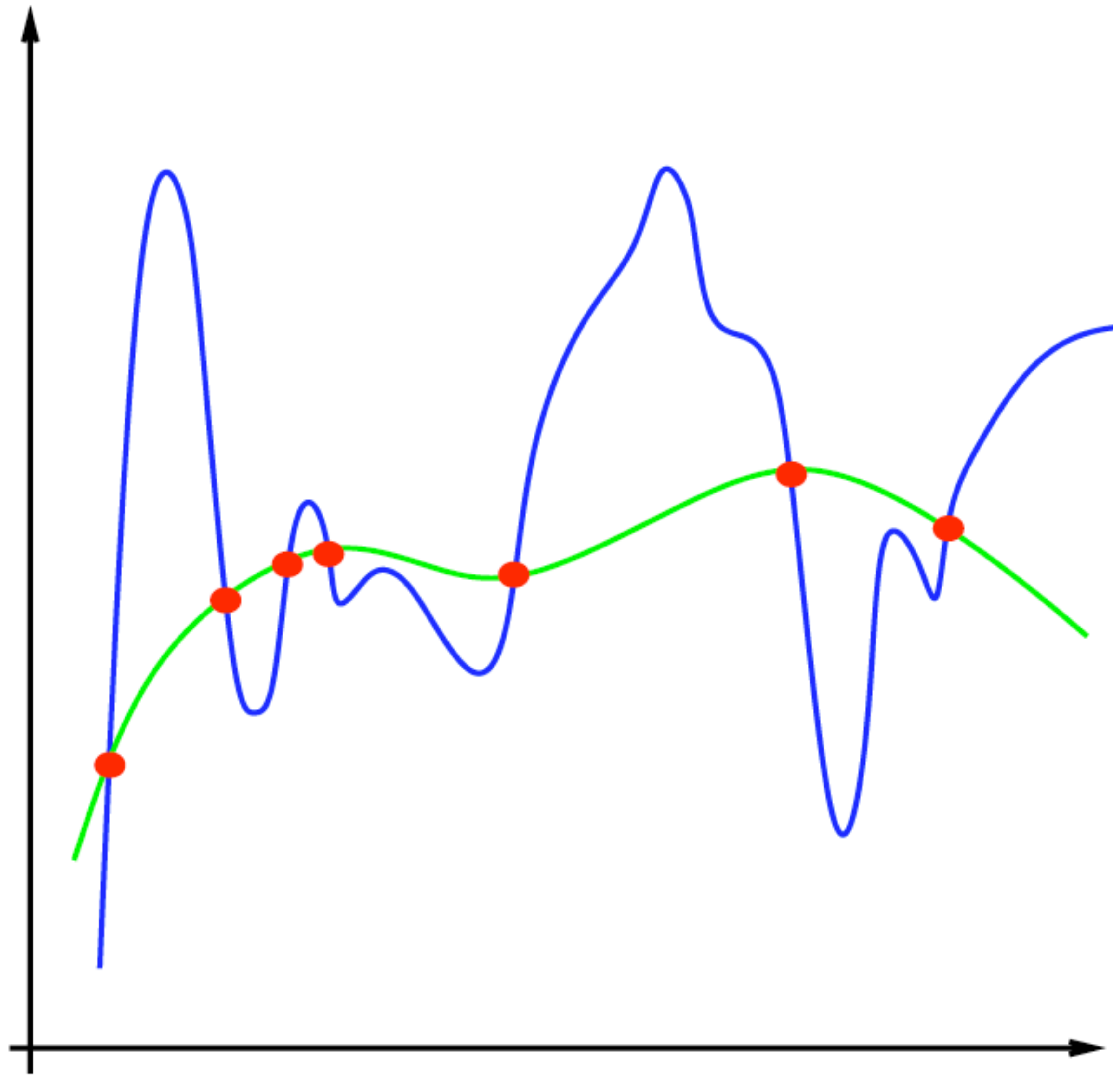
Deep Networks:Three theory questions

- *Approximation Theory:* When and why are deep networks better than shallow networks?

- *Optimization:* What is the landscape of the empirical risk?

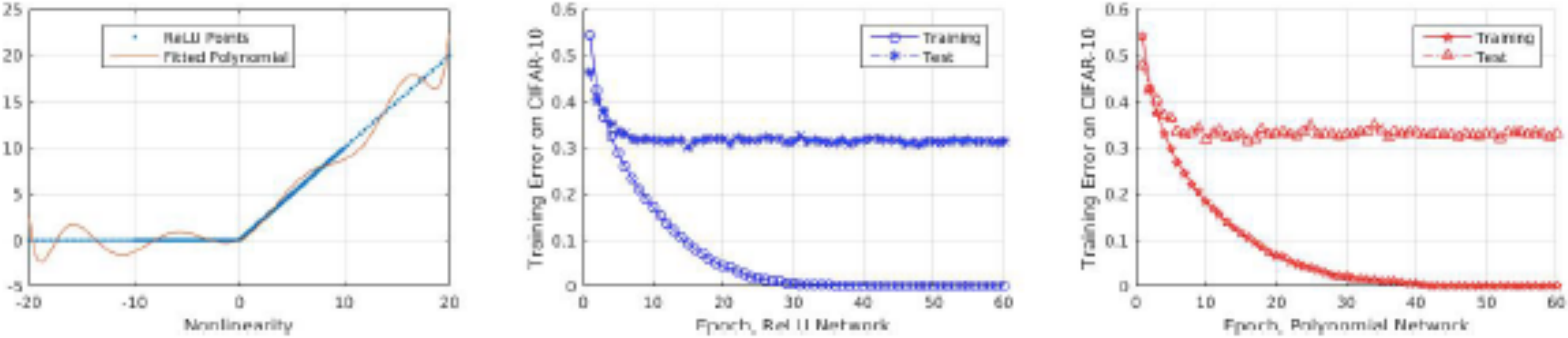
- *Learning Theory:* How can deep learning not overfit?

Problem of overfitting



Regularization or similar to control overfitting

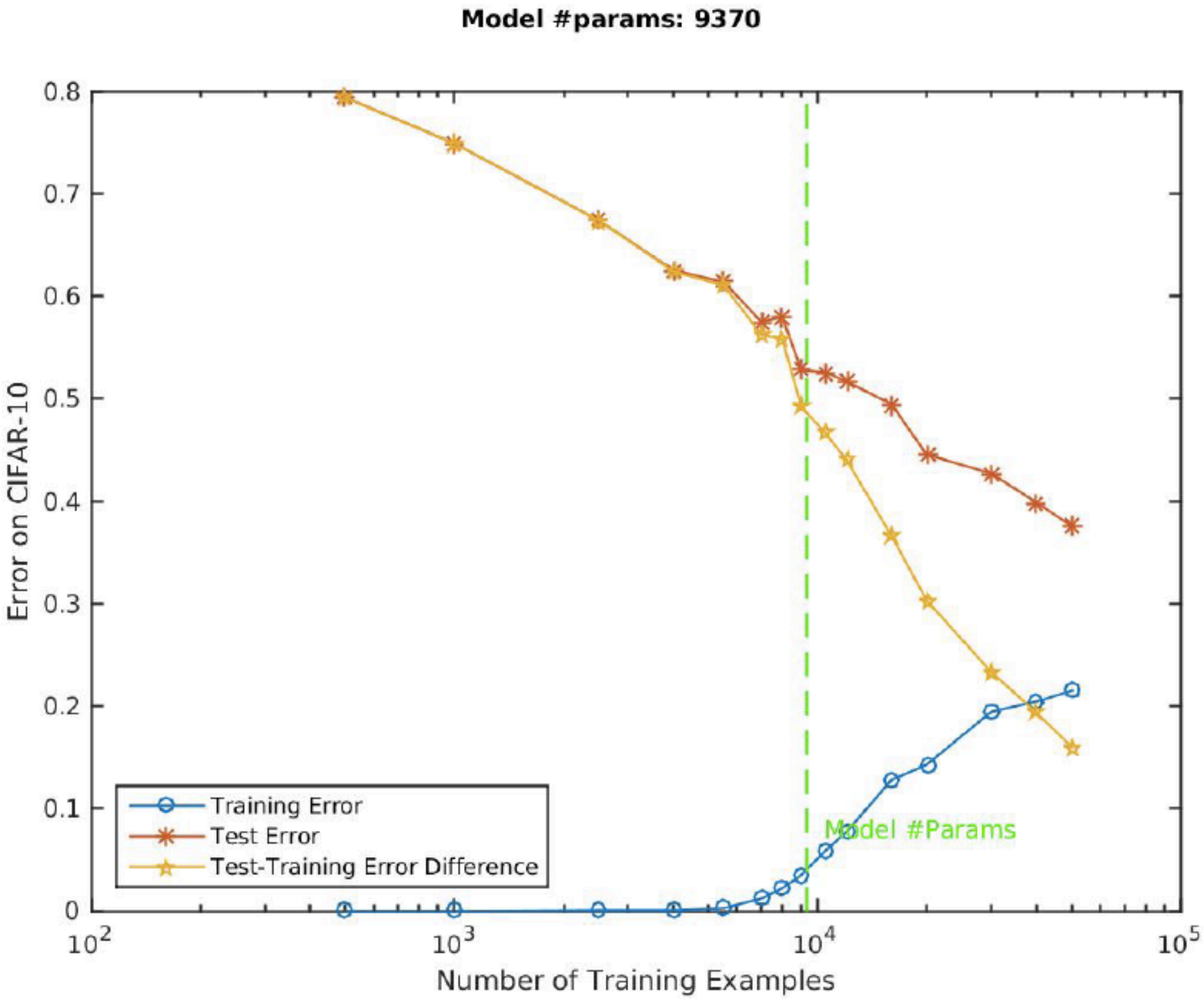
Deep Polynomial Networks show same puzzles



From now on we study polynomial networks!

Poggio et al., 2017

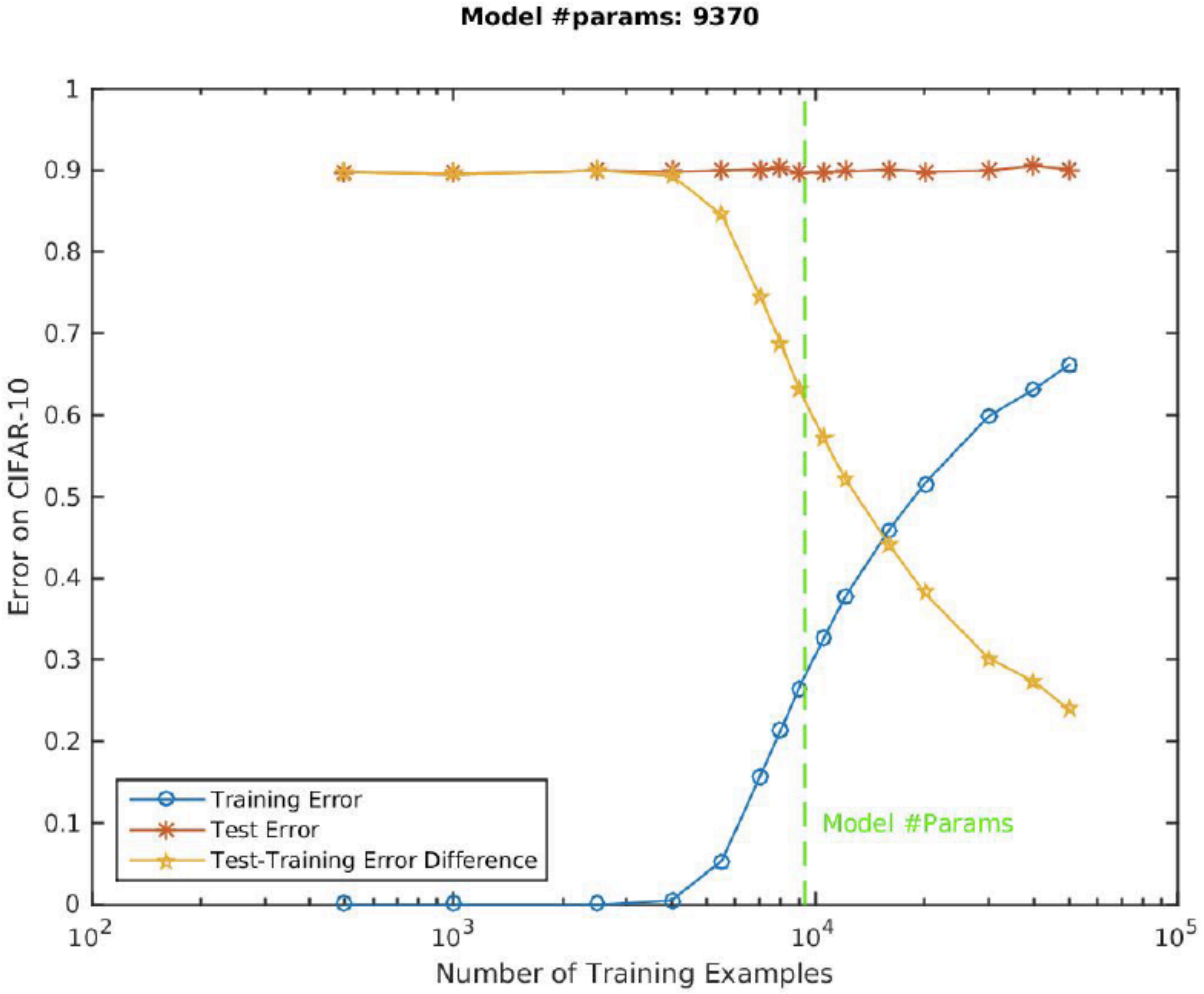
Good generalization with less data than # weights



Poggio et al., 2017

CENTER FOR Brains Minds+ Machines

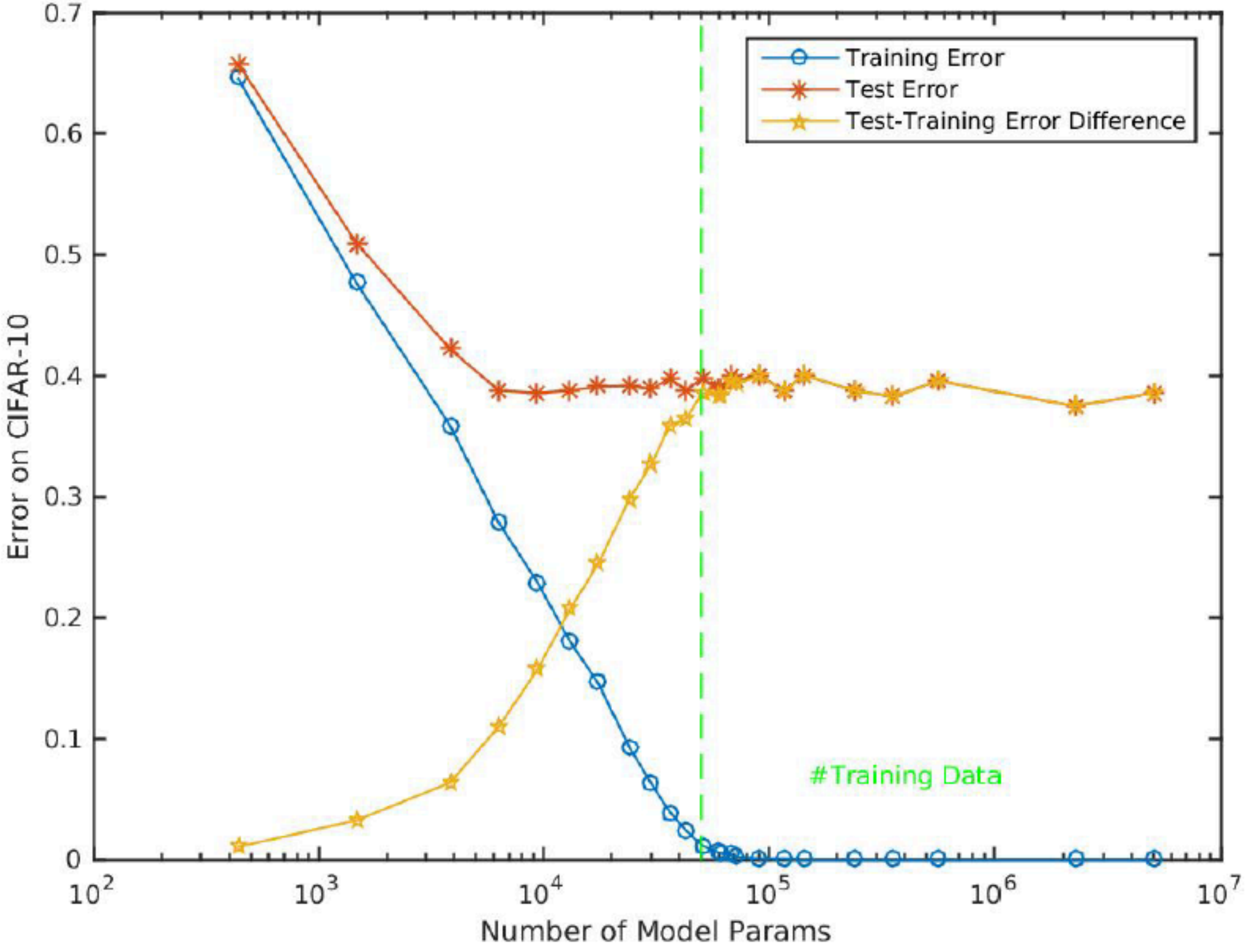
Randomly labeled data



Poggio et al., 2017

following Zhang et al., 2016, ICLR

No overfitting!

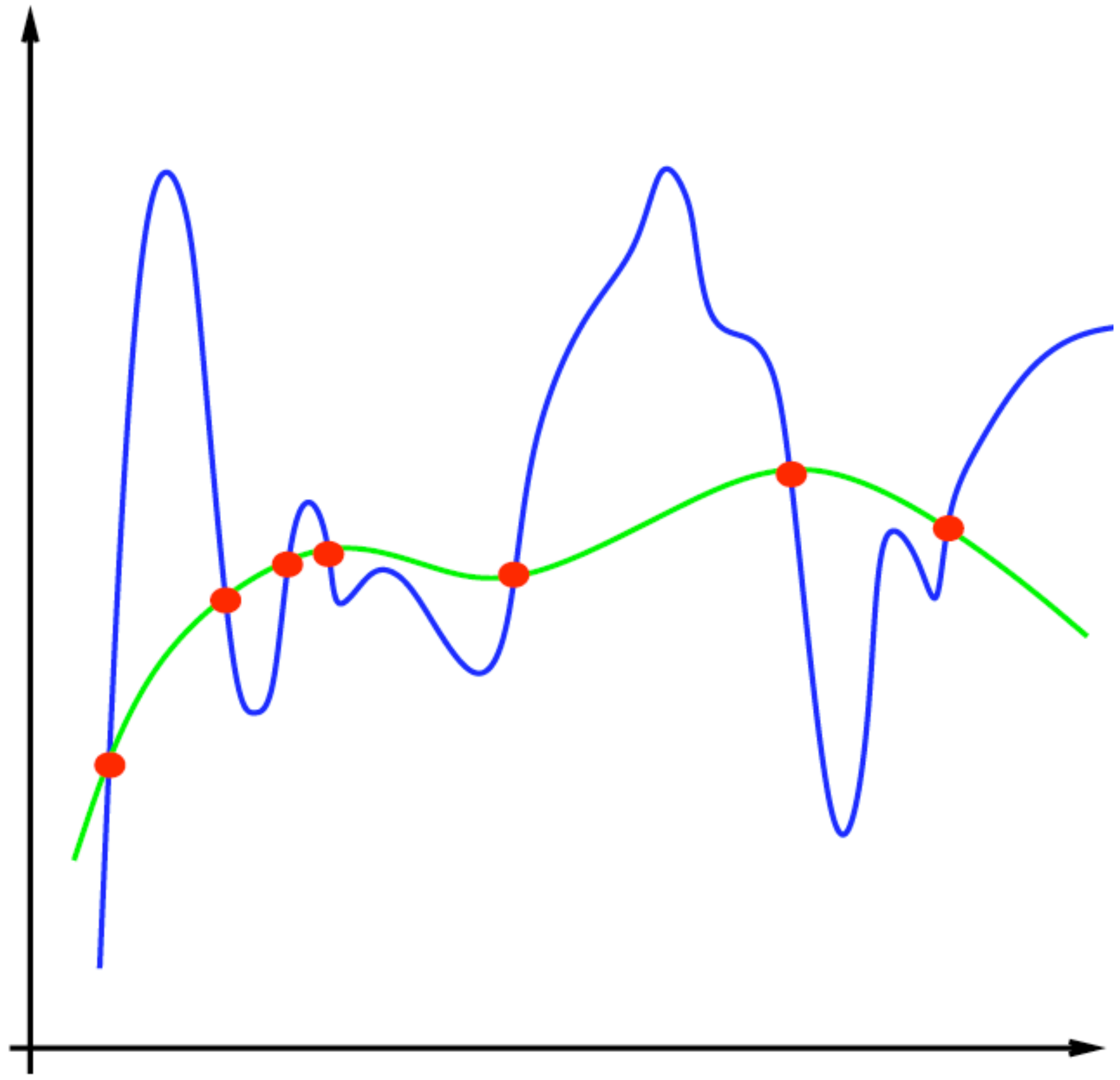


Poggio et al., 2017

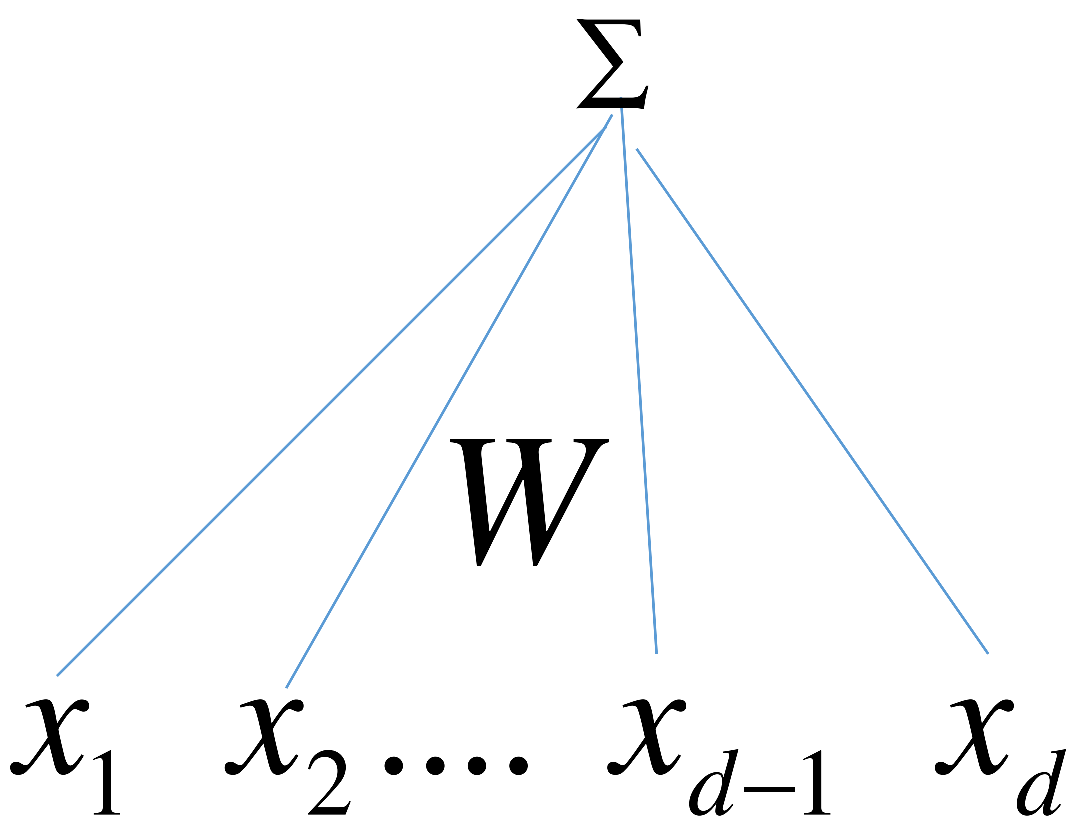
Explaining this figure is our main goal!

CENTER FOR Brains Minds+ Machines

No overfitting with GD



Implicit regularization by GD+SGD (linear case, no hidden layer)



$$W = YX^{\dagger}$$

Corollary 1. *When initialized with zero, both GD and SGD converges to the minimum-norm solution.*

Min norm solution is the limit for $\lambda \to 0$ *of regularized solution*

CENTER FOR Brains Minds+ Machines

Implicit regularization by GD: #iterations controls λ

Theorem 3.1 *In the setting of Section 2, let Assumption 1 hold. Let $\gamma \in]0, \kappa^{-1}]$. Then the following hold:*

(i) *If we choose a stopping rule $t^* : \mathbb{N}^* \to \mathbb{N}^*$ such that*

$$\lim_{n\to+\infty} t^*(n) = +\infty \quad \textit{and} \quad \lim_{n\to+\infty} \frac{t^*(n)^3 \log n}{n} = 0 \tag{9}$$

then

$$\lim_{n\to+\infty} \mathcal{E}(\hat{w}_{t^*(n)}) - \inf_{w\in\mathcal{H}} \mathcal{E}(w) = 0 \quad \mathbb{P}\textit{-almost surely.} \tag{10}$$

(ii) *Suppose additionally that the set $\mathcal{O}$ of minimizers of (1) is nonempty and let $w^\dagger$ be defined as in (2). If we choose a stopping rule $t^* : \mathbb{N}^* \to \mathbb{N}^*$ satisfying the conditions in (9) then*

$$\|\hat{w}_{t^*(n)} - w^\dagger\|_{\mathcal{H}} \to 0 \quad \mathbb{P}\textit{-almost surely.} \tag{11}$$

CENTER FOR
Brains
Minds+
Machines

Rosasco, Villa, 2015

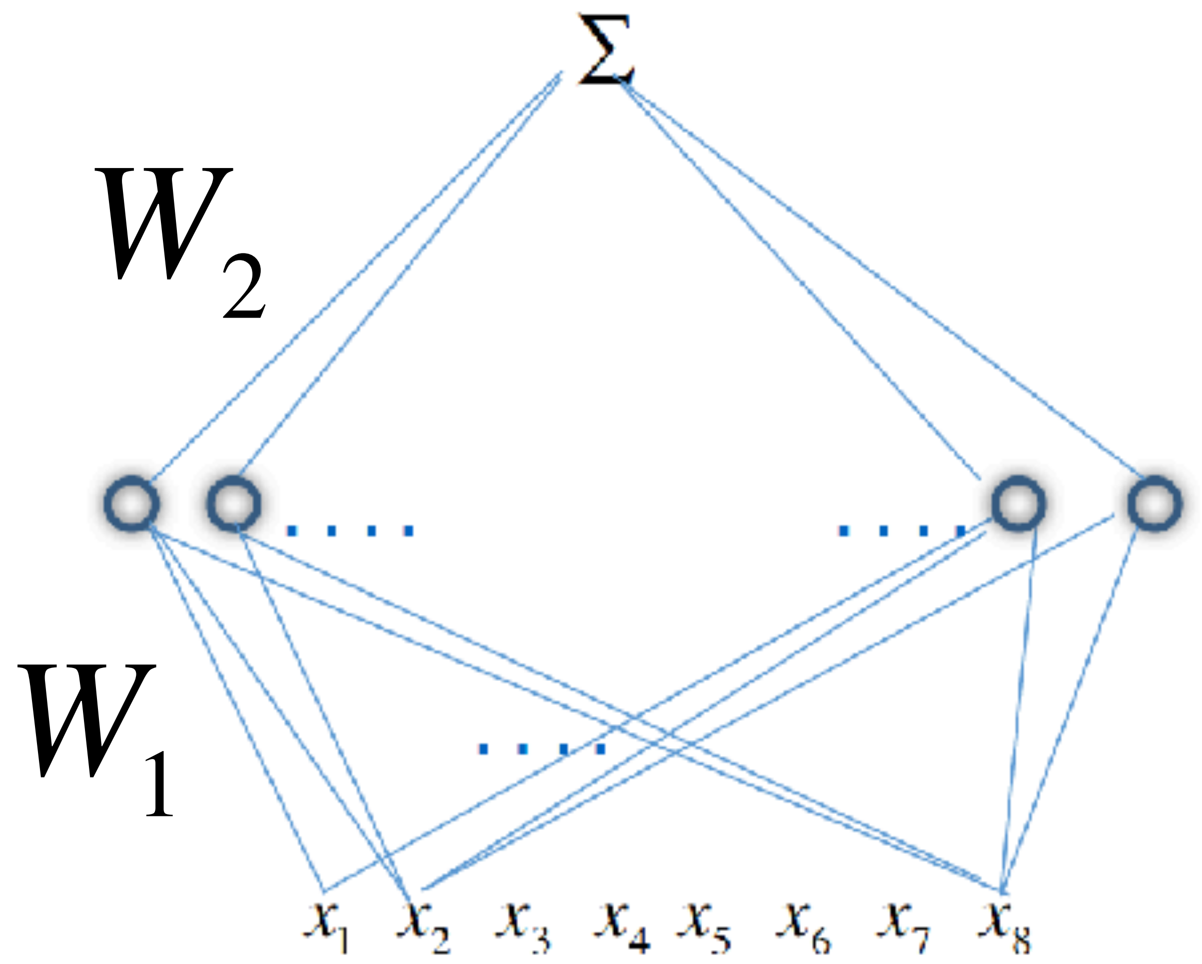
Deep linear network

Dynamical linear systems, training Consider the linear activation case with one hidden layer with d inputs, N hidden *linear* units and d' outputs. We assume $d > n$. We denote the loss with $L(w) = ||W_2 W_1 X - Y||^2$ and define $E = W_2 W_1 X - Y$, $E \in \mathbb{R}^{d',n}$, $W_2 \in \mathbb{R}^{d',N}$, $W_1 \in \mathbb{R}^{N,d}$. We obtain

$$\dot{W}_1 = -\nabla_{W_1} L(w) = -2\gamma W_2^\top E X^\top = -2\gamma(W_2^\top Y X^\top - W_2^\top W_2 W_1 X X^\top) \qquad (3)$$

and similarly

$$\dot{W}_2 = -\gamma E X^\top W_1^\top.$$



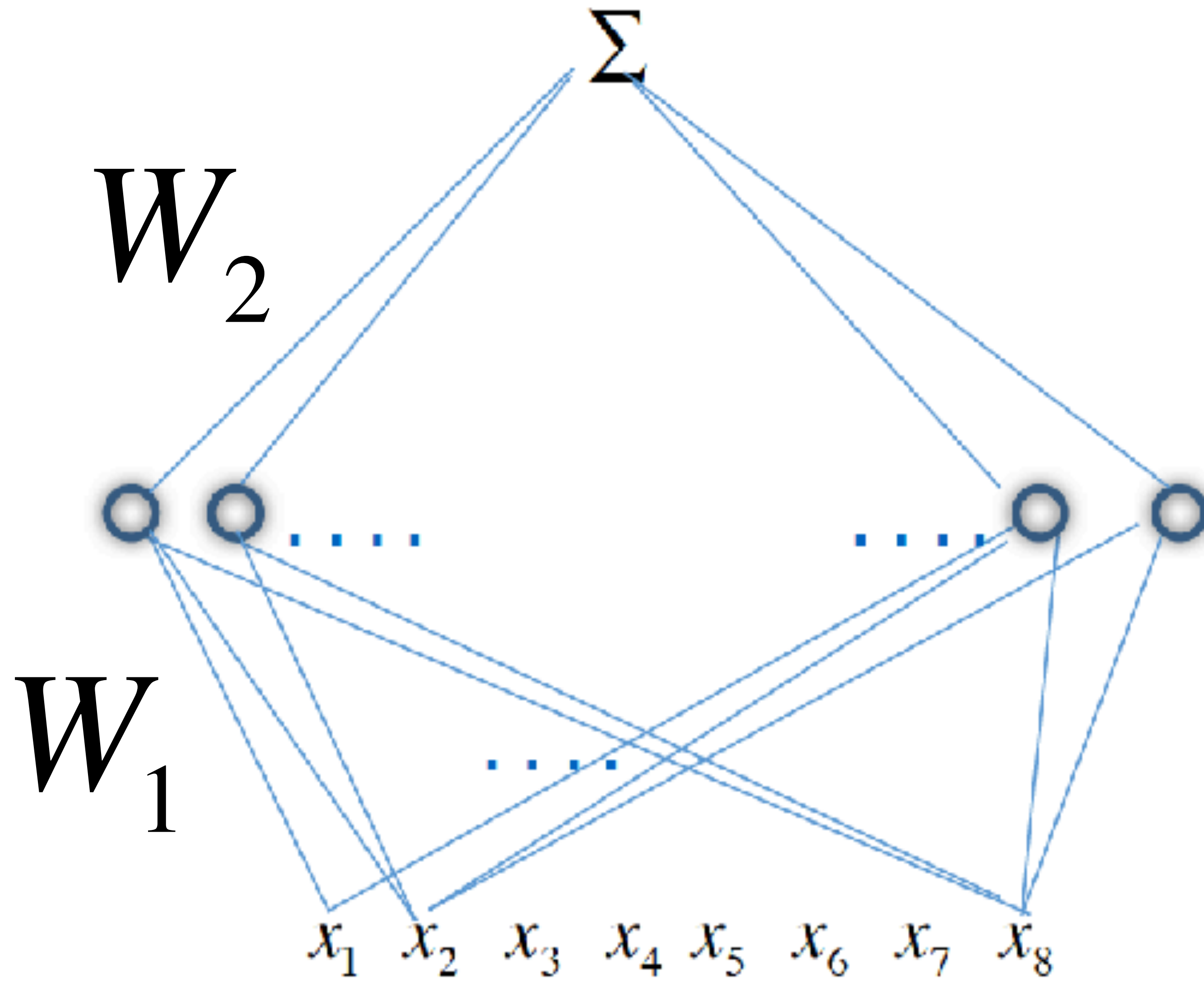
Gangulis, Saxe et al., 2015; Baldi+Hornik, 1989

Deep linear networks

Lemma 3. *For gradient descent and stochastic gradient descent with any mini-batch size,*

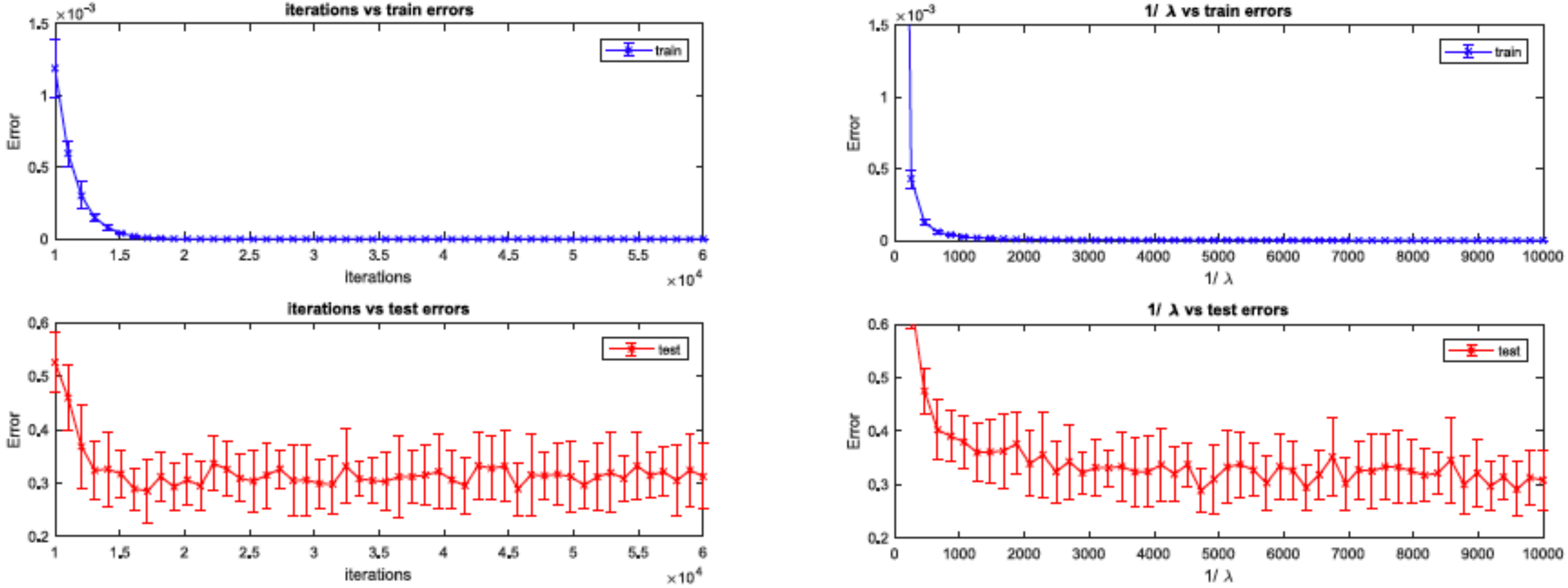
- *any number of the iterations adds no element in* $\mathrm{Null}(X^\top)$ *to the rows of* W_1*, and hence*
- *if the rows of* W_1 *has no element in* $\mathrm{Null}(X^\top)$ *at anytime (including the initialization), the sequence converges to a minimum norm solution if it converges to a solution.*

Lemma 4. *If* $W_2 \neq 0$*, every stationary point w.r.t.* W_1 *is a global minimum.*



Remark: $W_2 W_1 = A$ implies redundant parameters that are controlled if null space is empty

Deep linear network: GD as regularizer



CENTER FOR
Brains
Minds+
Machines

GD regularizes deep linear networks as it does for linear networks

Deep nonlinear (degree 2) networks

Dynamical polynomial multilayer systems, training We now discuss an extension of the above argument to the nonlinear activation case. Consider a polynomial second order (for simplicity and w.l.g) activation function $h(z) = az + bz^2$. The dynamical system (see for notation SI) is given by

$$\dot{W}_1 = -2(aW_2^\top E + 2b[(W_1 X) \circ (W_2^T E)])X^\top \tag{7}$$

and

$$\dot{W}_2 = -2[aEX^\top W_1^\top + bE(((W_1 X)^2)^\top)]. \tag{8}$$

CENTER FOR
Brains
Minds+
Machines

Linearized dynamics to study stable solutions

$$\dot{\delta_{W_1}} = -2\delta_{W_2^\top} Y X^\top + 2\delta_{W_2^\top} W_2^* W_1^* X X^\top + 2W_2^{*\top}\delta_{W_2} W_1^* X X^\top + 2W_2^{*\top} W_2^* \delta_{W_1} X X^\top$$

and similarly

$$\dot{\delta_{W_2}} = -2Y X^\top \delta_{W_1^\top} + 2\delta_{W_2} W_1^* X X^\top W_1^{*\top} + 2W_2^* \delta_{W_1} X X^\top W_1^{*\top} + 2W_2^* W_1^* X X^\top \delta_{W_1^\top}.$$

$$\dot{\delta_{W_1}} = -2\delta_{W_2^\top} Y X^\top$$

If W^* small

$$\dot{\delta_{W_2}} = -2Y X^\top \delta_{W_1^\top}$$

Deep nonlinear networks: conjecture

The conclusion about the extension to multilayer networks with polynomial activation is thus similar to the linear case and can be summarized as follows:

For low-noise data and a degenerate global minimum W^*, GD on a polynomial multilayer network avoids overfitting without explicit regularization, despite overparametrization.

Three theory questions: summary

- *Approximation theorems:* for hierarchical compositional functions deep but not shallow networks avoid the curse of dimensionality because of locality of constituent functions

- *Optimization remarks:* Bezout theorem suggests many global minima that are found by SGD with high probability wrt local minima

- *Learning Theory results and conjectures:* Unlike the case for a linear network the data dictate - because of the regularizing dynamics of GD - the number of effective parameters, which are in general fewer than the number of weights.