Machine Learning from a Statistical Physics Perspective

with an appendix on

Smart Inference for Covid19 tracing

Marc Mézard
Ecole normale supérieure - PSL University

Prairie's e-Colloquium
May 6, 2020

Machine Learning, supervised

Input → Machine → Output

Machine Learning, supervised

Input

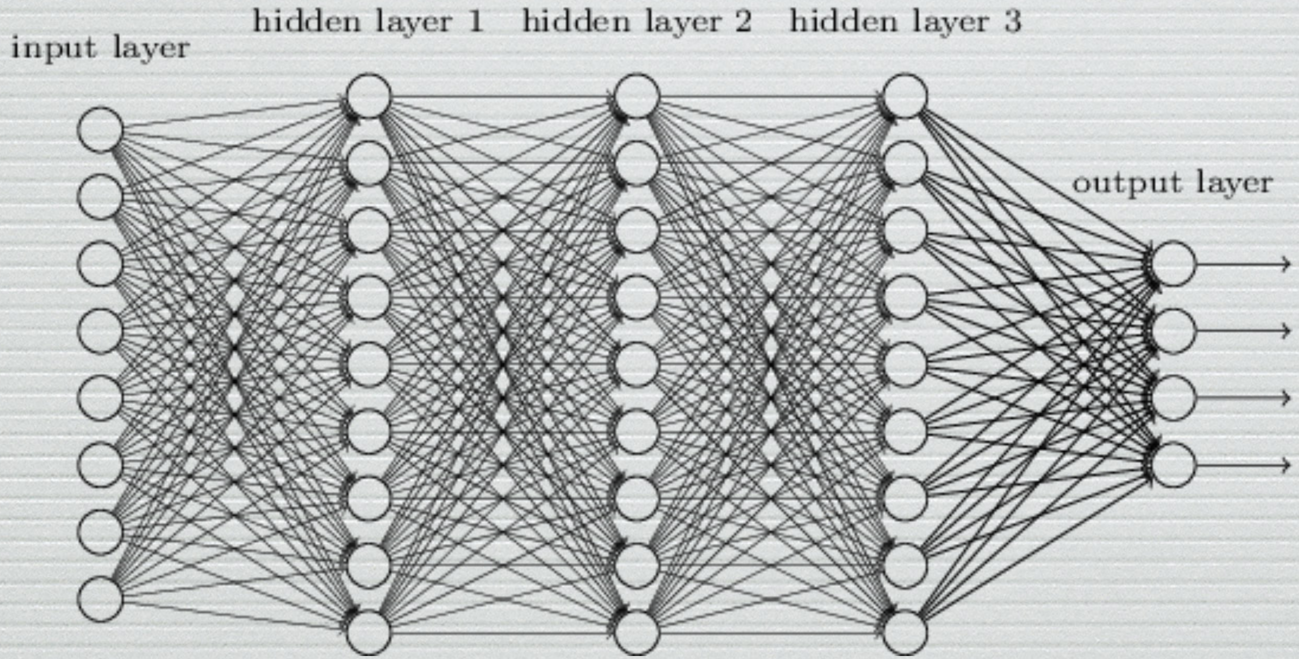
Machine

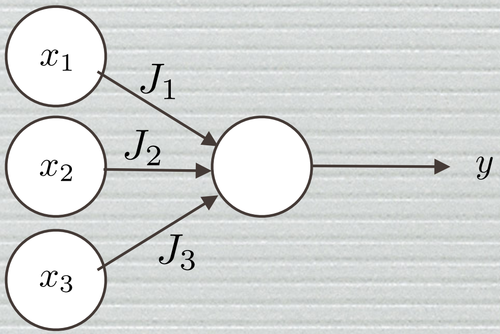
Output

CAT

DOG

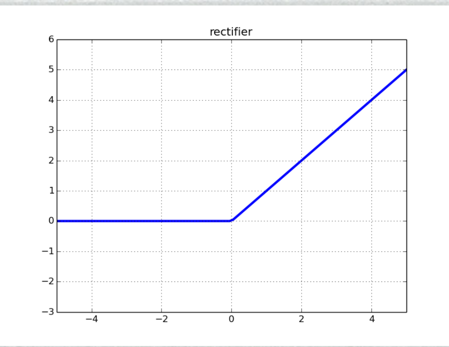
Deep neural networks





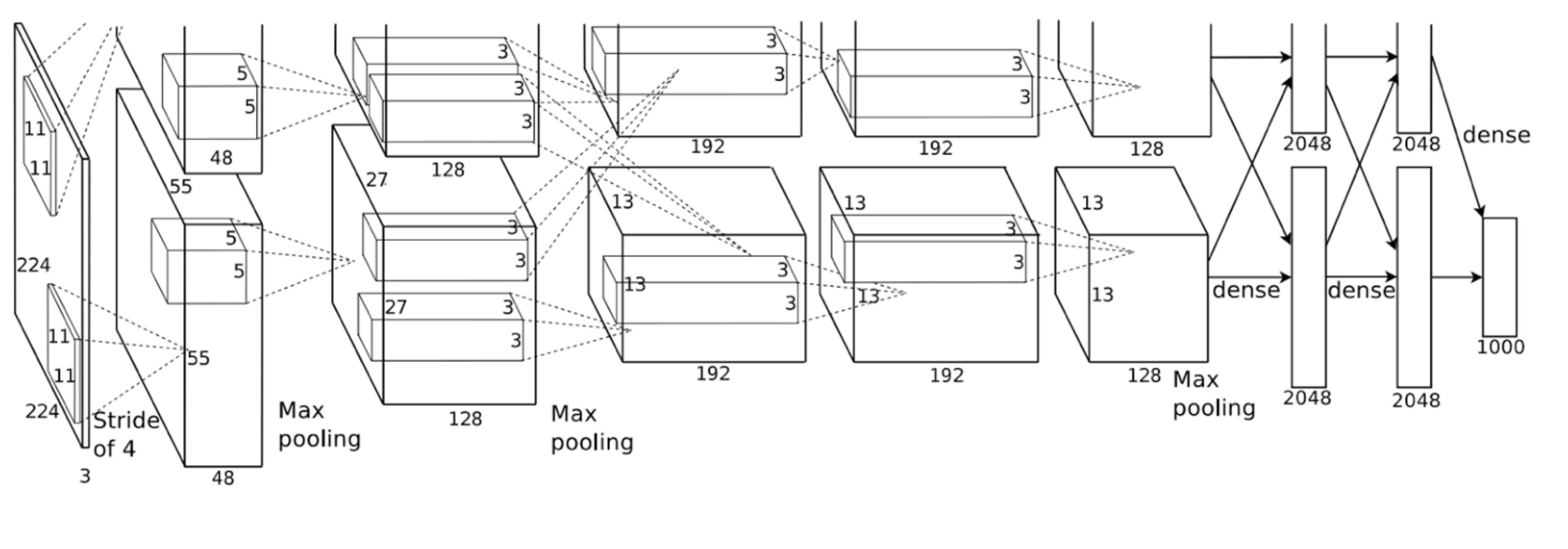
$$y = f(J_0 + J_1 x_1 + J_2 x_2 + J_3 x_3)$$

Can have 100,000 neurons, 100 layers, more than 1,000,000 parameters



Trained on huge databases, by simple gradient-descent type algorithms

Do we understand how it works?



One knows everything (the dream of the neuroscientist)
One understands very little. Accumulated practical knowledge.

No big theoretical progress in the last 25 years

Machine learning: training

$$y = f(W, \xi)$$

Database = M examples of input-output (ξ_μ, y_μ)

Optimization

Find W that minimizes $\sum_\mu [f(W, \xi_\mu) - y_\mu]^2$

(or other « loss function »)

Bayesian inference :

$$P(W | \{\xi_\mu, y_\mu\}) = \frac{1}{Z} P^0(W) \exp\left(-\beta \sum_\mu [f(W, \xi_\mu) - y_\mu)]^2 \right)$$

Unknown Data Prior Inverse temperature

Machine learning: training

$$\boxed{\xi} \longrightarrow \boxed{W} \longrightarrow \boxed{y} \qquad y = f(W$$

Database = M examples of i ... ut (ξ_μ, y_μ)

Other big « prior »: architecture!

Optimization

Find W that ... s $\sum_\mu [f(W, \xi_\mu) - y_\mu]^2$

(or other « loss ... nction »)

Bayesian inference :

$$\underset{\text{Unknown}}{P(W|\{\xi_\mu, y_\mu\})} = \underset{\text{Data}}{\frac{1}{Z}} \underset{\text{Prior}}{P^0(W)} \exp\left(\underset{\text{Inverse temperature}}{-\beta} \sum_\mu [f(W, \xi_\mu) - y_\mu)]^2 \right)$$

Machine learning: generalization

$$y = f(W, \xi)$$

$$(\xi_\mu, y_\mu)$$

Database = M examples of input-output

Machine learning: generalization

$$\boxed{\xi} \longrightarrow \boxed{W} \longrightarrow \boxed{y}$$

$$y = f(W, \xi)$$

$$(\xi_\mu, y_\mu)$$

Database = M examples of input-output

Generalization: having found the best (a « typical ») set of parameters W^*, compute the performance of the machine on some **new data**

$$E_g = \sum_\nu \left[y_\nu - f(W^*, \xi_\nu)\right]^2$$

Machine learning: training and generalization

Learning: $$P(W|\{\xi_\mu, y_\mu\}) = \frac{1}{Z} P^0(W) \; \exp\left(-\beta \sum_\mu [f(W, \xi_\mu) - y_\mu)]^2\right)$$

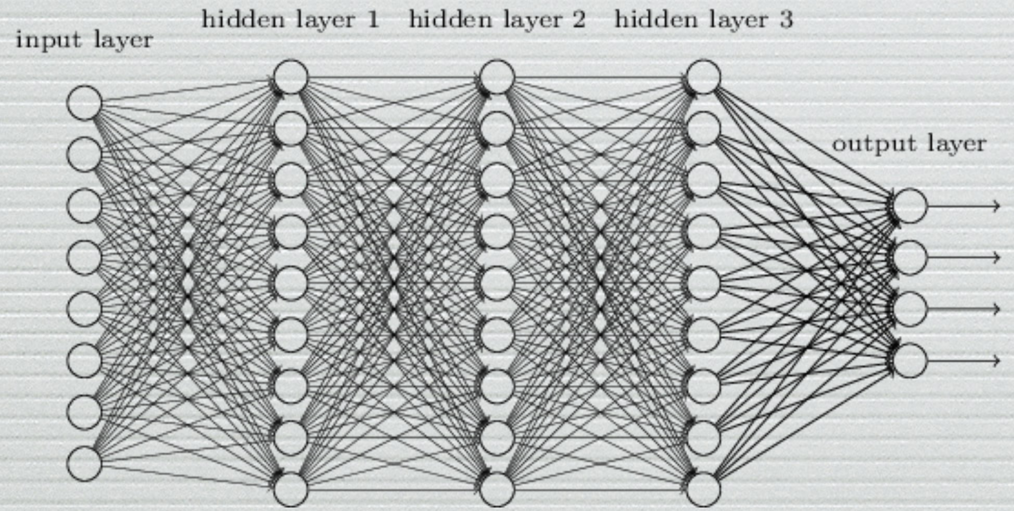
Generalization: $$E_g = \sum_\nu [y_\nu - f(W^*, \xi_\nu)]^2$$

Two main issues:

- Algorithmic
- Theoretical

Machine learning: training and generalization

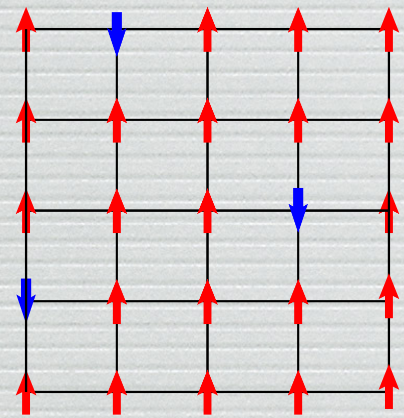
Learning: $P(W|\{\xi_\mu, y_\mu\}) = \frac{1}{Z} P^0(W) \exp\left(-\beta \sum_\mu [f(W,\xi_\mu) - y_\mu)]^2\right)$



Learning problem: optimization or sampling in a large dimensional space, with a disordered « energy function ». Typical of statistical physics.

→ **Statistical physics. Models, disorder, ensembles, replicas, message-passing equations…**

Magnets and Ising Model



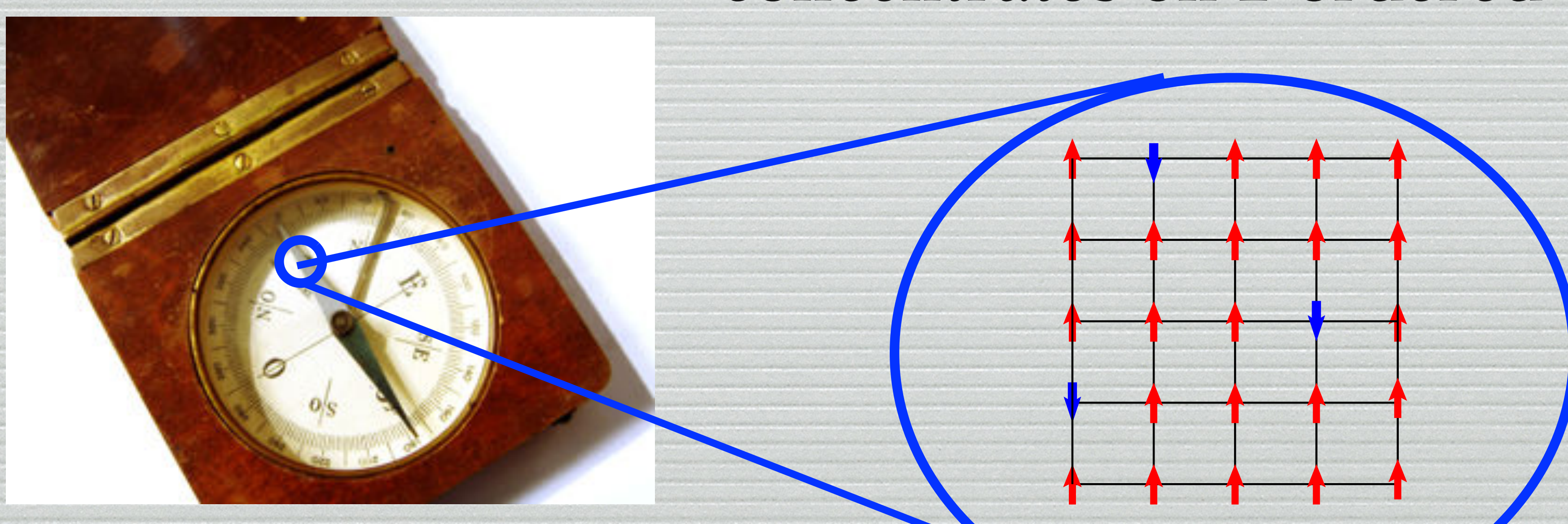
$$s_i \in \{\pm 1\}$$

$$E = -\sum_{ij} J_{ij} s_i s_j$$

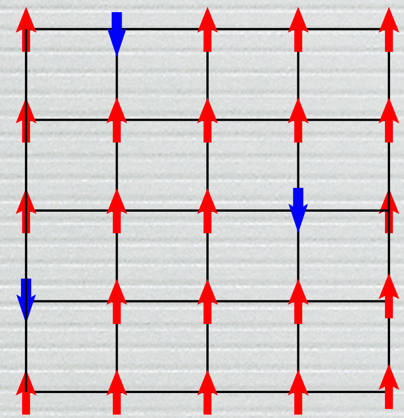
Equilibrium: $P(s_1, \ldots, s_N) = \frac{1}{Z} e^{-E/T}$

Ferromagnet: $J_{ij} > 0$

At low T: spins align, P concentrates on 2 ordered states



Magnets and Ising Model



$$s_i \in \{\pm 1\}$$

$$E = -\sum_{ij} J_{ij} s_i s_j$$

Equilibrium: $P(s_1, \ldots, s_N) = \frac{1}{Z} e^{-E/T}$

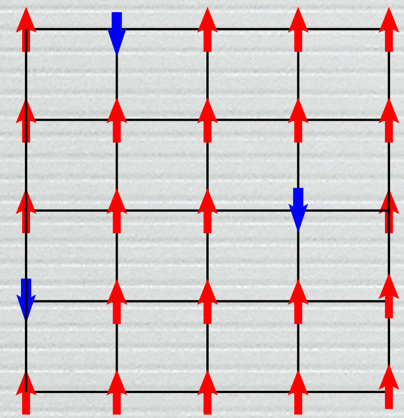
Ferromagnet: $J_{ij} > 0$

Phases : $\langle s_i \rangle = M$

$$M = \tanh\left(\sum_j J_{ij} s_j\right) \simeq \tanh\left(zJM\right)$$

« Mean Field » (Weiss 1907)

Magnets and Ising Model



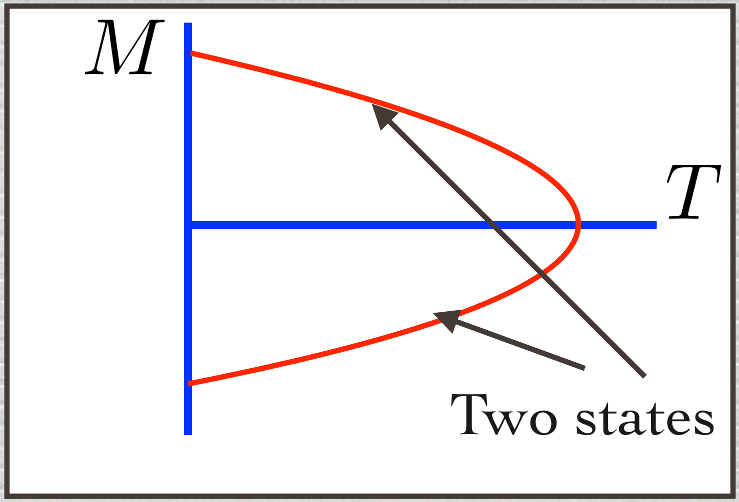
$$s_i \in \{\pm 1\}$$

$$E = -\sum_{ij} J_{ij} s_i s_j$$

Equilibrium: $P(s_1, \ldots, s_N) = \frac{1}{Z} e^{-E/T}$

Ferromagnet: $J_{ij} > 0$

Phases : $\langle s_i \rangle = M$

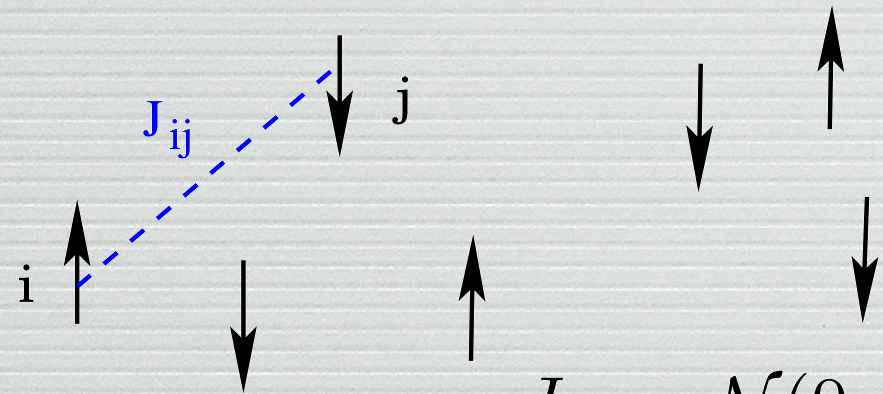


$$M = \tanh\left(\sum_j J_{ij} s_j\right) \simeq \tanh\left(zJM\right)$$

« Mean Field » (Weiss 1907)

Random Magnets: Spin glasses
disorder ensemble

CuMn



$J_{ij} \sim \mathcal{N}(0, 1/N)$

Strongly disordered system:

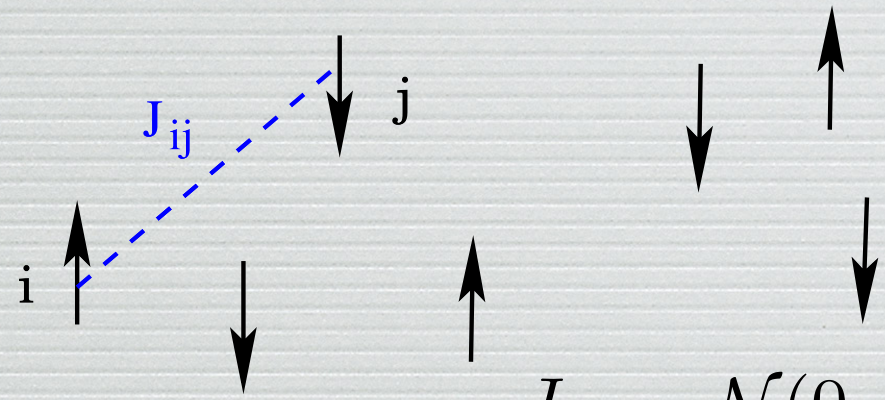
$$s_i \in \{\pm 1\}$$

$$E_J(s) = -\sum_{ij} J_{ij} s_i s_j$$

$$P_J(s) = \frac{1}{Z_J} e^{-\beta E_J(s)}$$

Random Magnets: Spin glasses
disorder ensemble

CuMn



$$J_{ij} \sim \mathcal{N}(0, 1/N)$$

$$s_i \in \{\pm 1\}$$

$$E_J(s) = -\sum_{ij} J_{ij} s_i s_j$$

$$P_J(s) = \frac{1}{Z_J} e^{-\beta E_J(s)}$$

Strongly disordered system:

Spin glass sample described by the whole set of J_{ij}

e.g. (« SK model »): $$J_{ij} \sim \mathcal{N}\left(\frac{J_0}{N}, \frac{1}{N}\right)$$

Ensemble:

drawn from a probability distribution. eg iid

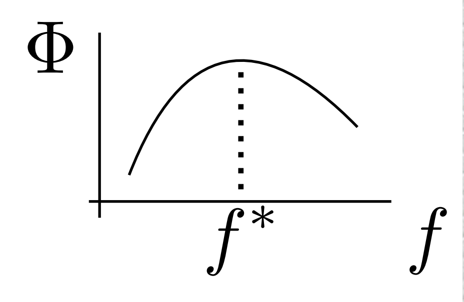
Replicas, large deviations

Free energy of sample J: $\quad f_J = -\frac{1}{\beta N} \log Z_J$

Probability of finding a sample with $f_J = f$: $\quad e^{N\Phi(f)}$

Almost all samples have $f_J = f^*$

and the same thermodynamic properties



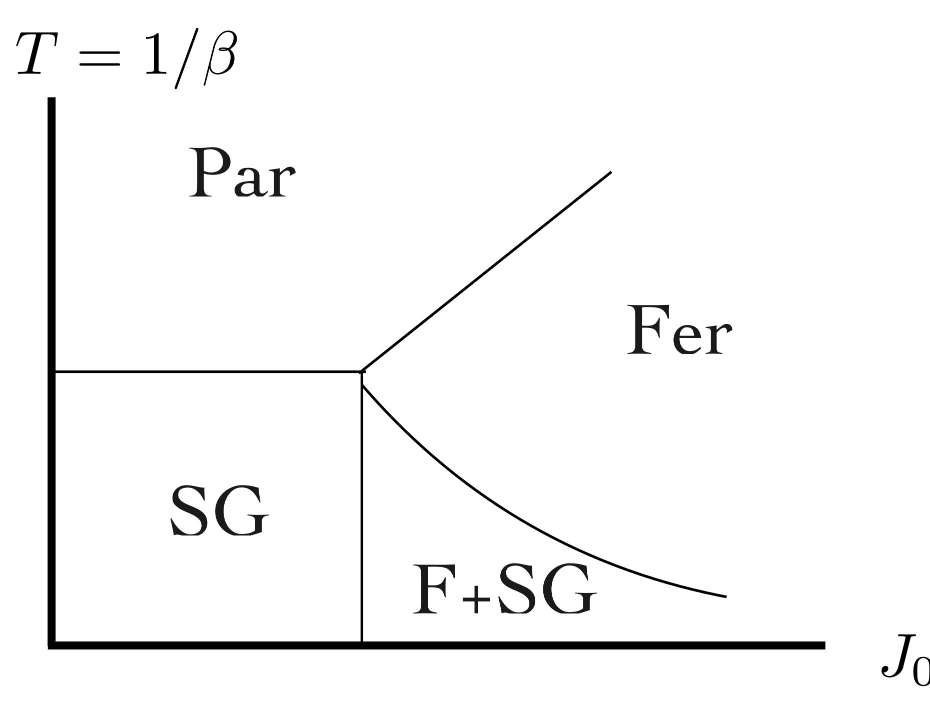
Reconstruct the large deviation function $\Phi(f)$ and find f^*

$$\mathcal{E}(Z_J^n) = \int df \; e^{N[-n\beta f + \Phi(f)]} \left[\int df \; e^{N\Phi(f)} \right]^{-1} \simeq e^{-nN\beta f^*}$$

when $n \to 0$

studied in the thermodynamic limit with the Laplace method

Phase Diagram and Message Passing   eg SK model

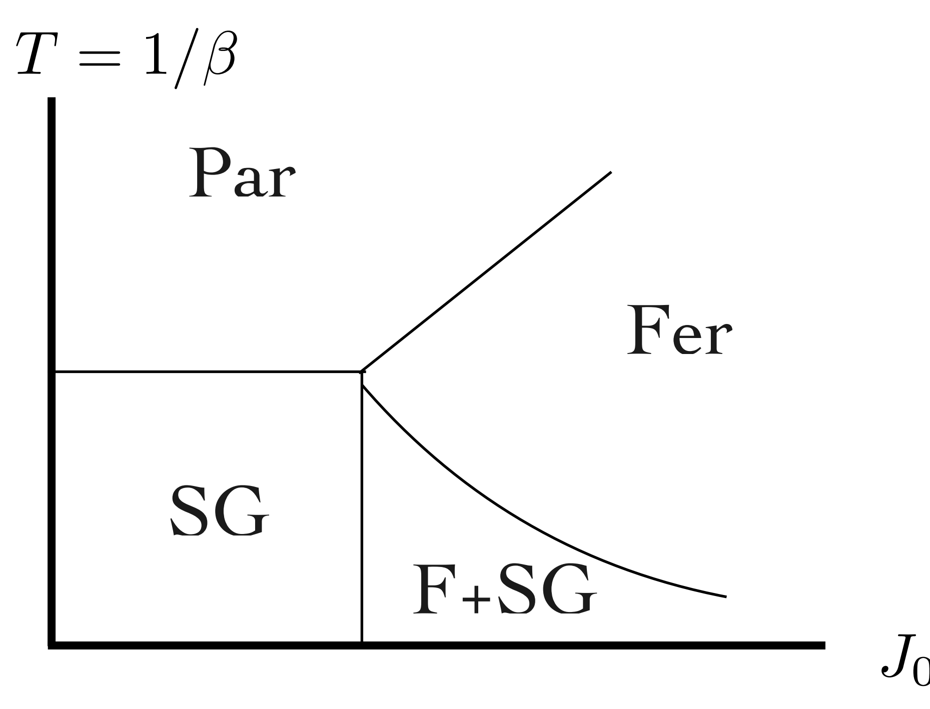


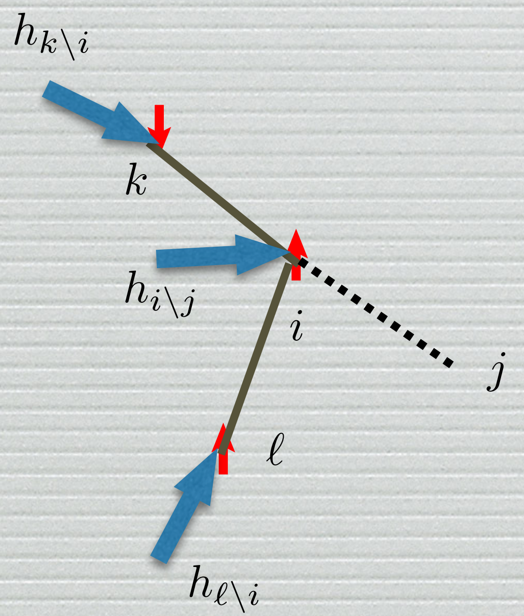
Phase Diagram and Message Passing

eg SK model

Inhomogeneous Mean Field (cavity) ➔ **message passing algorithms** BP, AMP, GAMP, VAMP

Built up in the last 40 years, a very fruitful connexion…

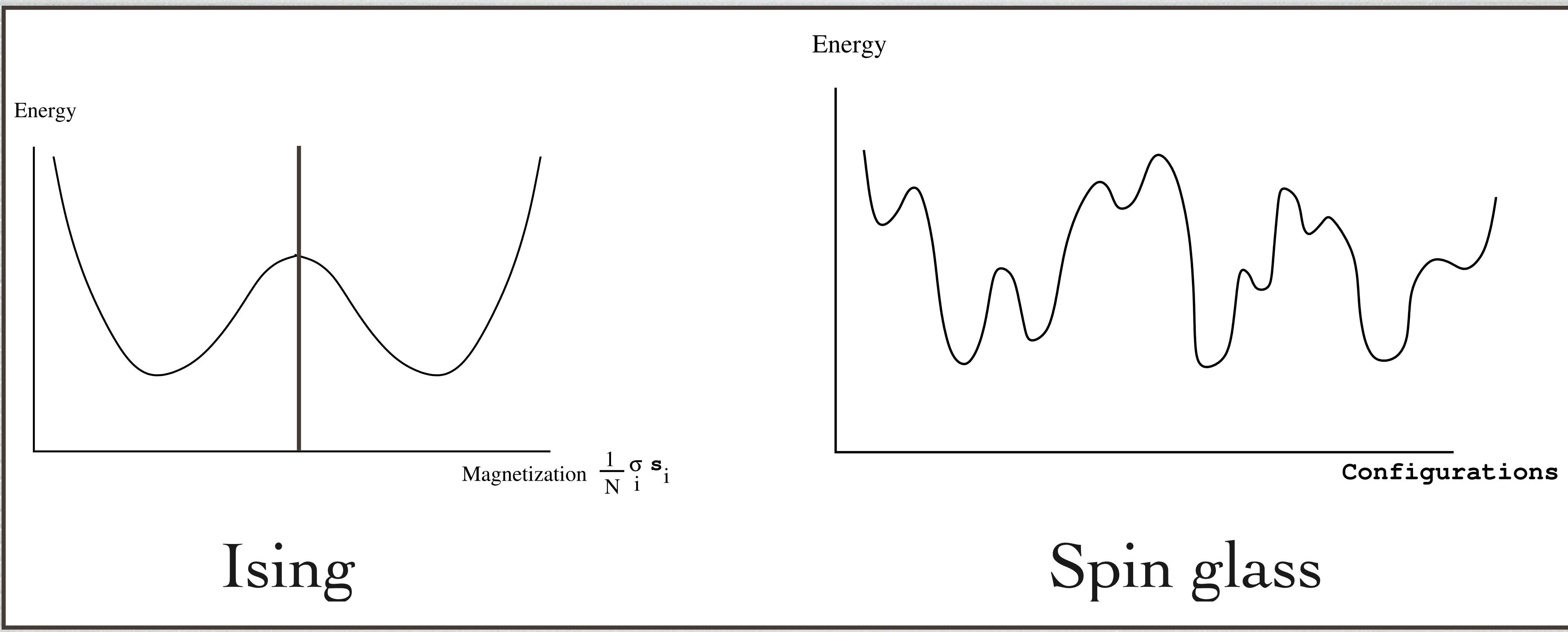




$$h_{i\setminus j} = \frac{1}{\beta}\text{atanh}[\tanh(\beta J_{ki})\tanh(\beta h_{k\setminus i})] + \frac{1}{\beta}\text{atanh}[\tanh(\beta J_{\ell i})\tanh(\beta h_{\ell\setminus i})]$$

SG phase = many solutions

Mean-field lessons



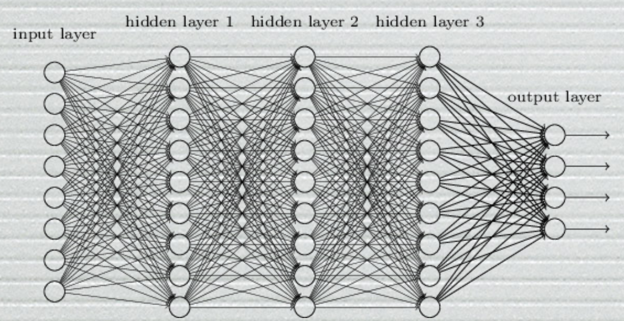
1- Glass « phase » : Many pure states, unrelated by symmetry, organized in a hierarchical « ultrametric » structure

2- Many metastable states, unrelated by symmetry

3- « True » ground state : fragile to perturbation!

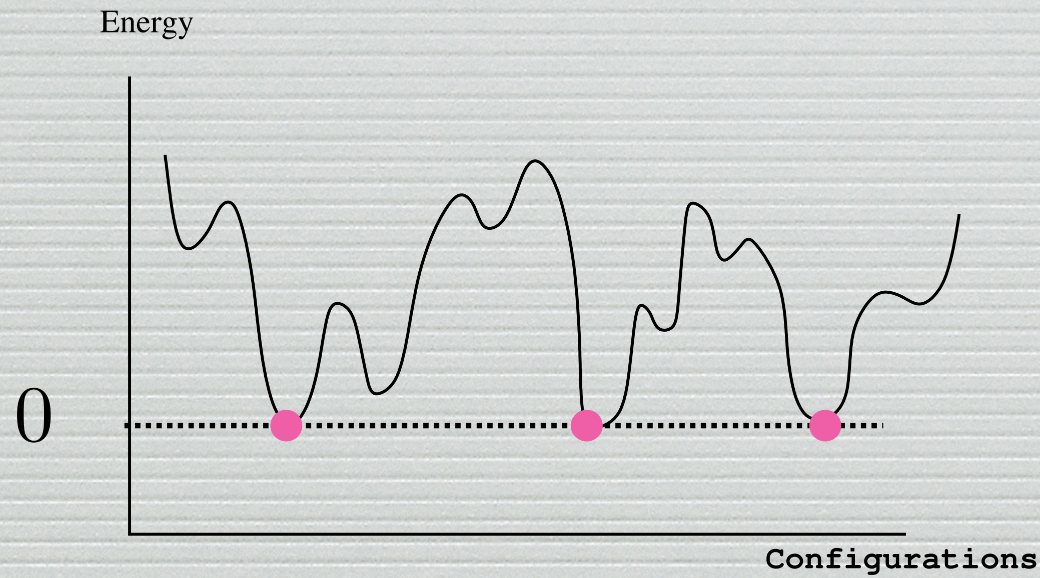
Machine learning: training

Learning: $P(W|\{\xi_\mu, y_\mu\}) = \frac{1}{Z} P^0(W) \, \exp\left(-\beta \sum_\mu [f(W,\xi_\mu) - y_\mu)]^2\right)$



Variables = W .

Disorder in the sample = data base ξ_μ, y_μ



Zero energy = perfect performance of the machine on the training set

Landscape? How does it depend on the problem, the database, the architecture?

Stochastic gradient descent often find a solution. Depends on details, but good…

Machine learning: generalization

Learning: $P(W|\{\xi_\mu, y_\mu\}) = \frac{1}{Z} P^0(W) \; \exp\left(-\beta \sum_\mu [f(W, \xi_\mu) - y_\mu)]^2\right)$

Generalization: $E_g = \sum_\nu [y_\nu - f(W^*, \xi_\nu)]^2$

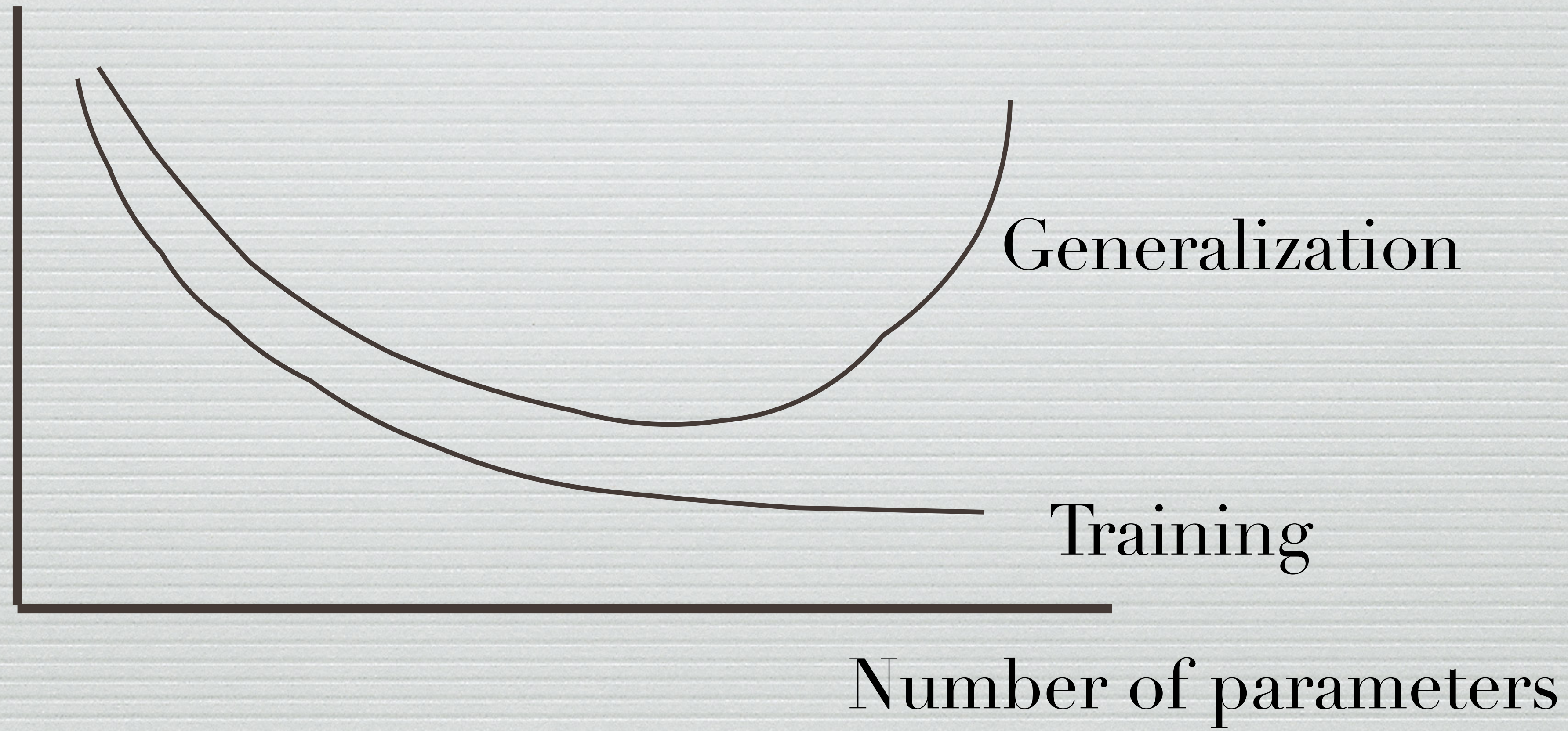
Deep networks work in an « over-parametrized » regime

Makes learning easier

Should degrade generalization: why training with so many parameters does not lead to overfitting ?

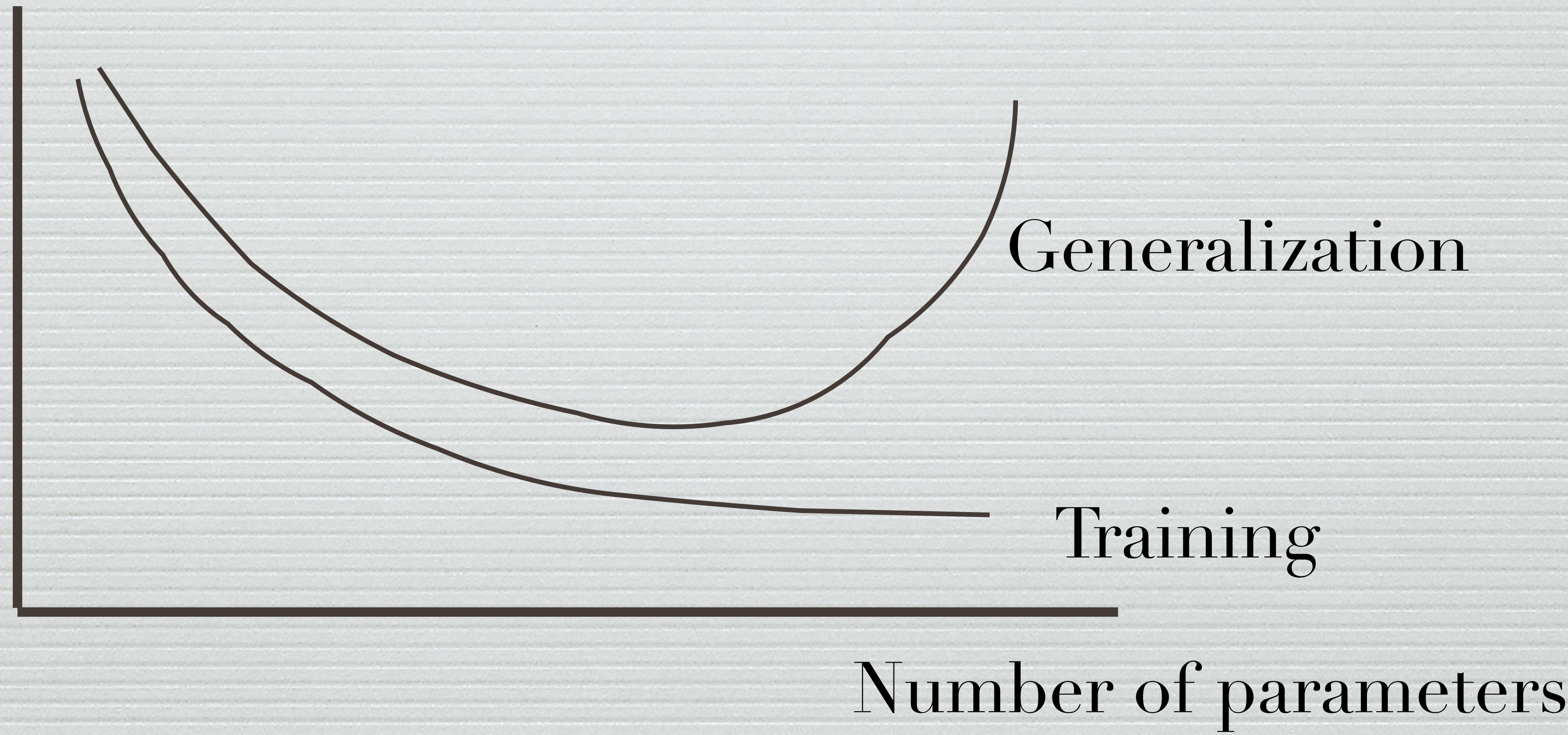
Machine learning: training and generalization

Usual behavior in statistics

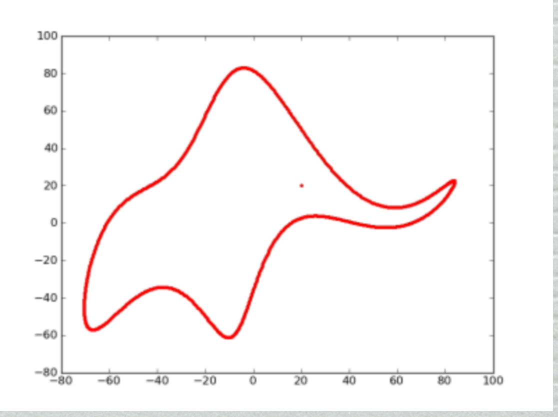


Machine learning: training and generalization

Usual behavior in statistics

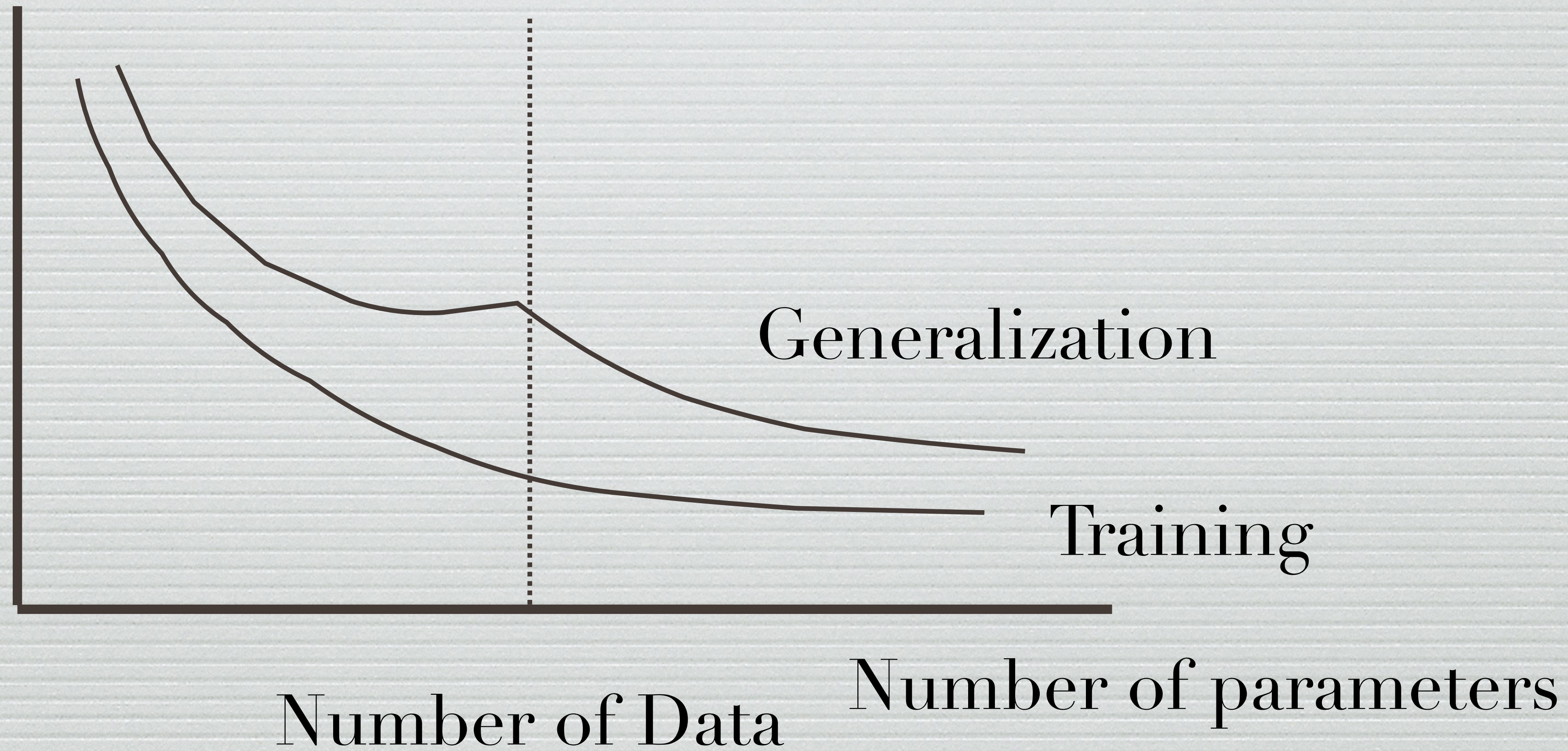


With 4 parameters I can fit
an elephant (J. von Neumann)



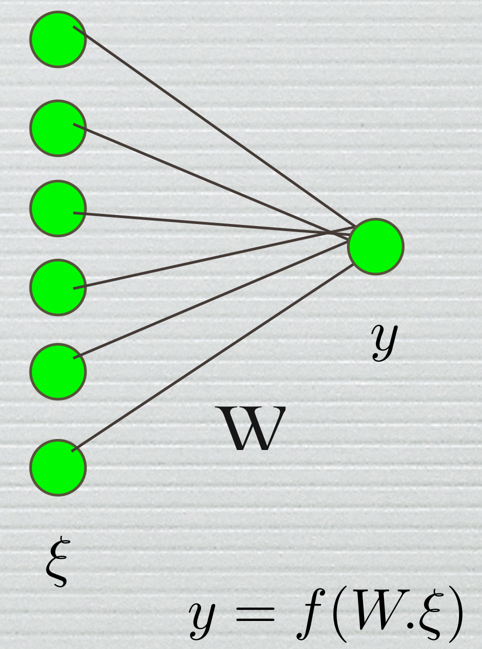
Machine learning: training and generalization

Deep networks



Machine learning Theory

Some important general statements.



Two layers limited to linearly separable problems
More than two: universal computer

Complexity results. Bounds on the difference between training energy and generalization energy, for different classes of functions

Simple settings. Convex optimization.
Linear problems

Machine learning Theory

What statistical physics can bring:

Tools and concepts for empirical analysis (landscape, learning dynamics)

Precise statements for the asymptotic regime (thermodynamic limit). Phase transitions, learning dynamics.

Requires an ensemble to model the data

Model of data: ensemble

Learning:

$$P(W|\{\xi_\mu, y_\mu\}) = \frac{1}{Z} P^0(W) \ \exp\left(-\beta \sum_\mu [f(W,\xi_\mu) - y_\mu)]^2\right)$$

Algorithmic studies typically uses one (or several) databases for $\{\xi_\mu, y_\mu\}$: data = quenched disorder

$\xi_\mu =$  $y_\mu =$ CAT

Theoretical analysis relies on a probabilistic « **model of the world** ». Independent patterns drawn from $P_\xi(\xi) P_y(y|\xi)$

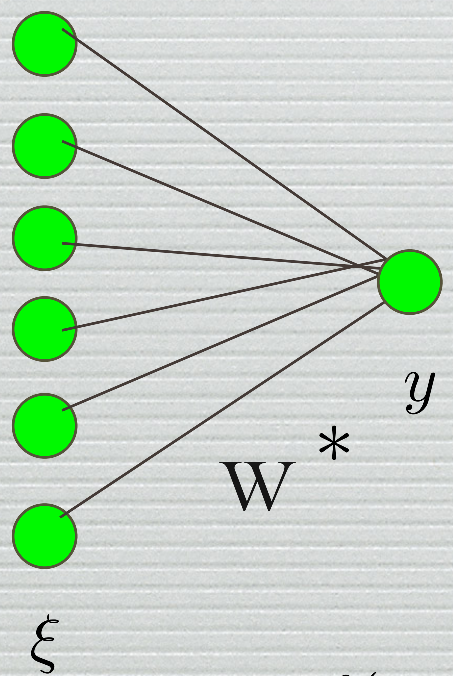
Quenched disorder

Examples from the 80's: iid patterns' entries $\xi_{i\mu} \sim \mathcal{N}(0,1)$

Example from the 90's:
« teacher-student perceptron »

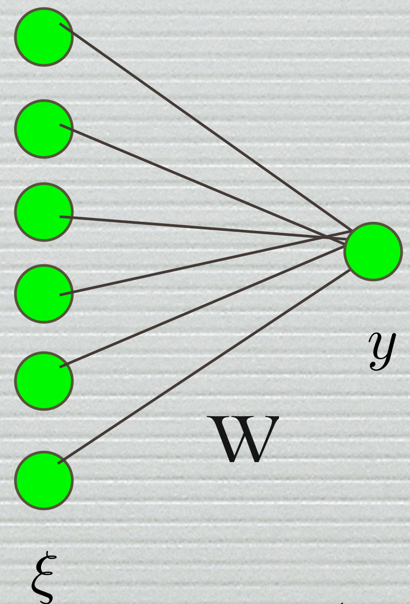
Teacher: generates parameters w^* from teacher prior $w_i^* = \pm 1$

generates data $\xi_i^\mu = \pm 1$ and target $y^\mu = Sign(\sum_i w_i^* \xi_i^\mu)$



$y = f(W.\xi)$

Student: same architecture, machine-learning for finding its weights w_i

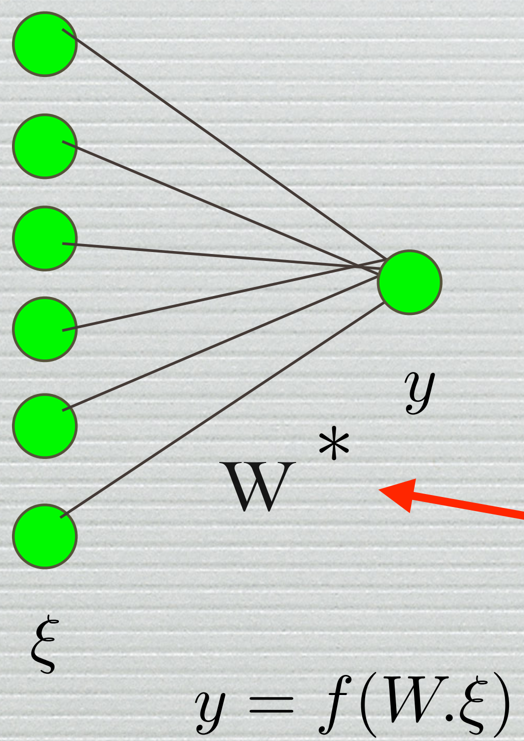


$y = f(W.\xi)$

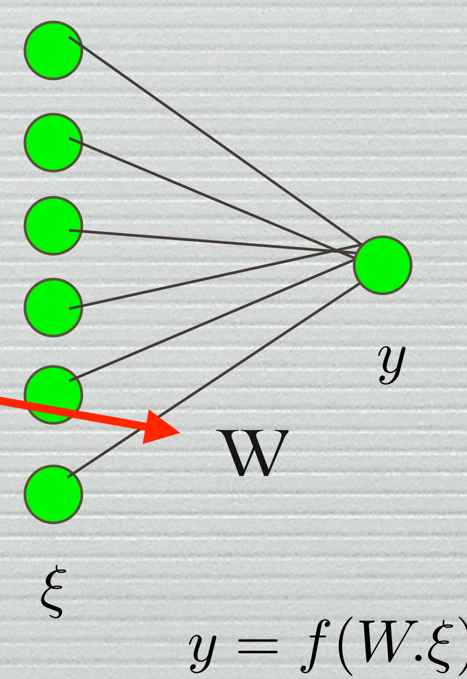
Example from the 90's:
« teacher-student perceptron »

Teacher: generates parameters w^* from teacher prior $w_i^* = \pm 1$

generates data $\xi_i^\mu = \pm 1$ and target $y^\mu = Sign(\sum_i w_i^* \xi_i^\mu)$



Student: same architecture, machine-learning for finding its weights w_i



Example from the 90's: « teacher-student perceptron »

$$E_\xi(W) = \sum_{\mu=1}^{P} \left[\mathrm{Sign}(W.\xi_\mu) - \mathrm{Sign}(W^*.\xi_\mu)\right]^2$$

Binary weights. Discrete optimization

Replica analysis= properties of the Gibbs measure

Gardner 88, Gardner Derrida 89 ,Gyorgyi 90
Barbier et al. 2019

$$P(W) = \frac{1}{Z} e^{-\beta E_\xi(W)}$$

Example from the 90's: « teacher-student perceptron »

$$E_\xi(W) = \sum_{\mu=1}^{P} \left[\mathrm{Sign}(W.\xi_\mu) - \mathrm{Sign}(W^*.\xi_\mu)\right]^2$$

Binary weights. Discrete optimization

Replica analysis= properties of the Gibbs measure

Gardner 88, Gardner Derrida 89 ,Gyorgyi 90
Barbier et al. 2019

$$P(W) = \frac{1}{Z} e^{-\beta E_\xi(W)}$$

Generalization error depends on the typical angle between W and W^*

Phase diagram in the thermodynamic limit

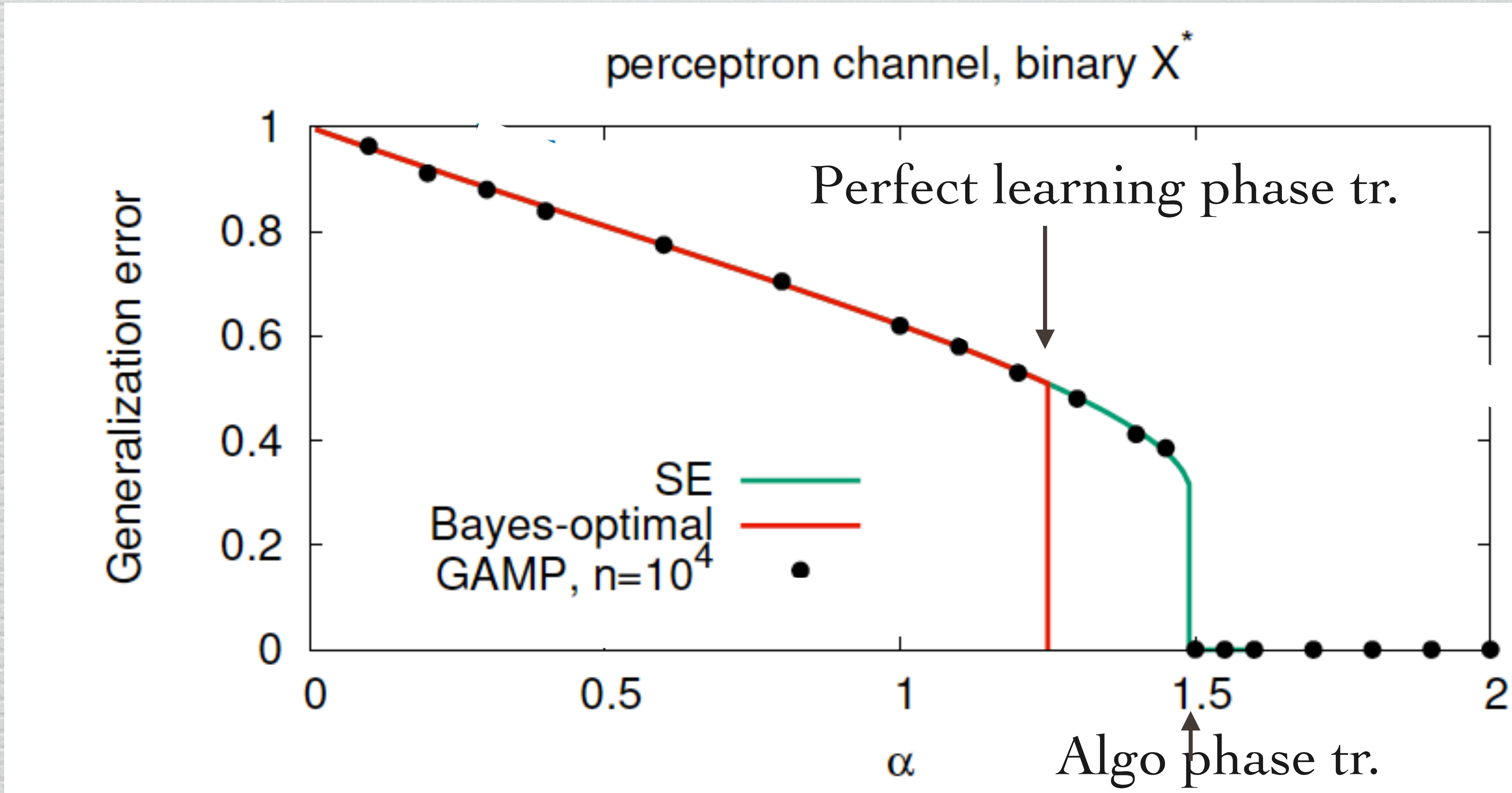
$$N \to \infty \quad P \to \infty \quad \alpha = P/N$$

Example. Binary perceptron $w_i \in \{\pm 1\}$

Algorithm: TAP-AMP equations

MM 89, Rangan 2011, Krzakala et al. 2012

Teacher-student binary perceptron



- Algorithm based on mean field equations

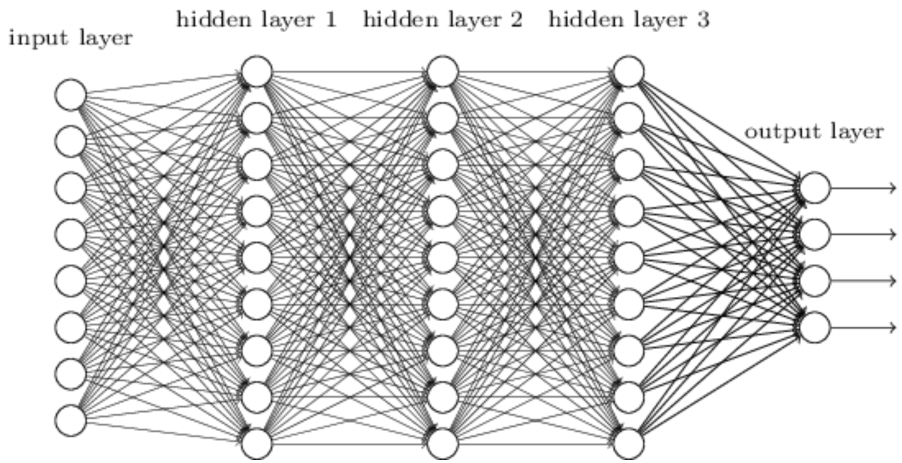
Theory Gyorgyi 1990

With discrete weights: Phase transition to perfect generalization when the size of the database reaches the threshold $\alpha_c = 1.245$. Fast message passing algorithm for $\alpha > 1.49$

Conjectured 30 years ago. Proof: Barbier et al. 2019

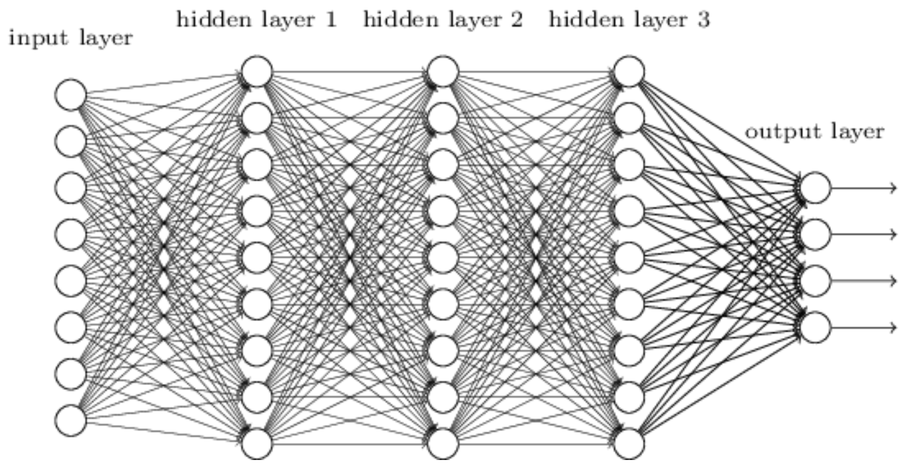
With continuous weights $w_i \in \mathbb{R}$ **:**
Generalization error of maximally stable W decreases like $.5005/\alpha$

Nice results, but of little use for understanding realistic networks. Decoupling between theoretical results and practical engineering applications…



Why does it work?

Architecture
Algorithms
Data structure



Why does it work?

Architecture
Algorithms
Data structure

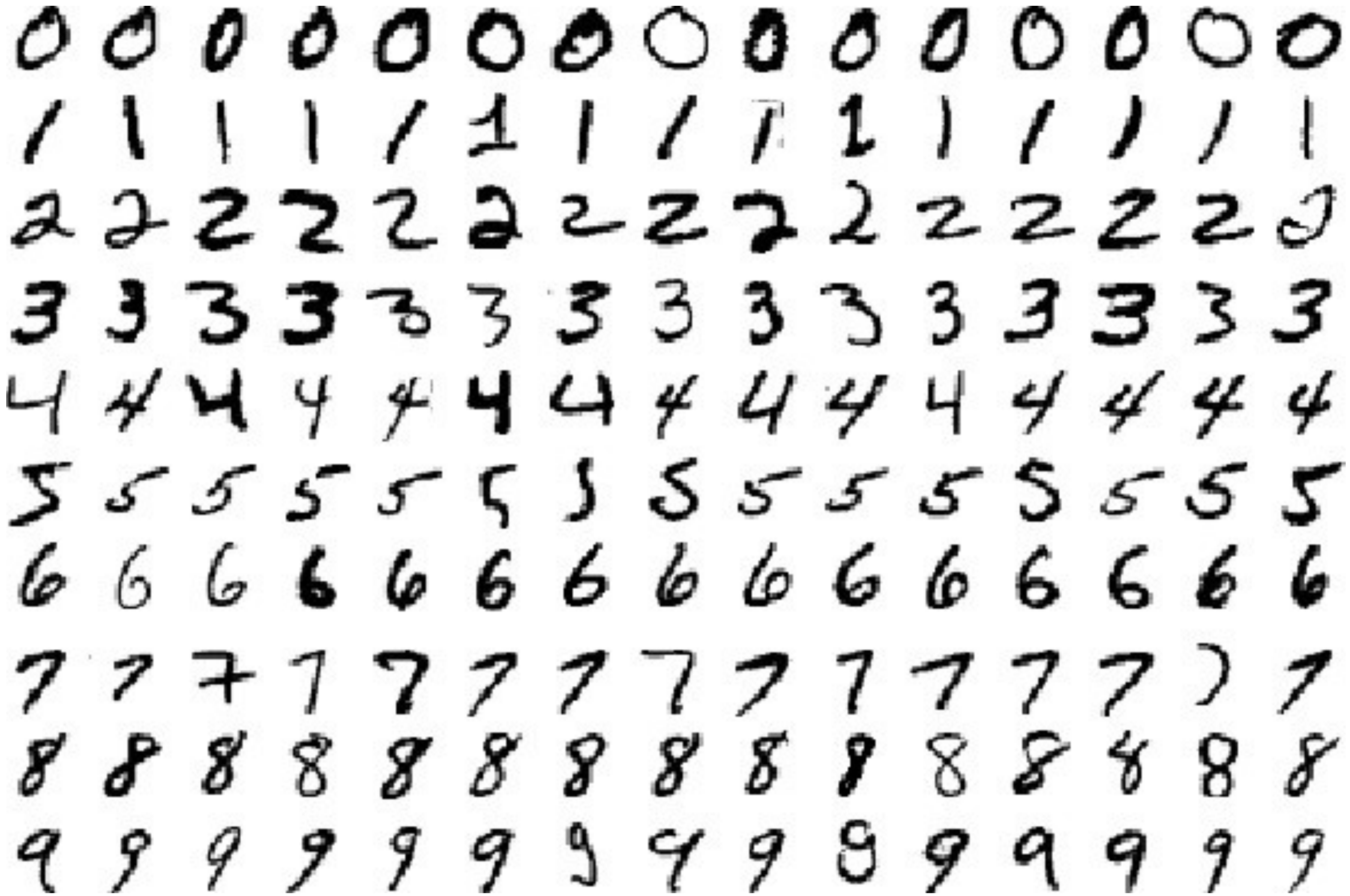
Data structure

- Hidden manifolds and sub manifolds
- Combinatorial structure
- Euclidean correlations

- Analyse data
- Build generative models that can be analyzed fully in some large size limit
- Understand mechanisms

The hidden manifold of data

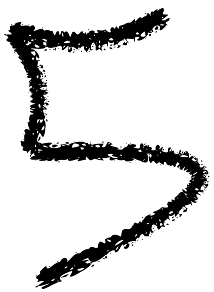
MNIST



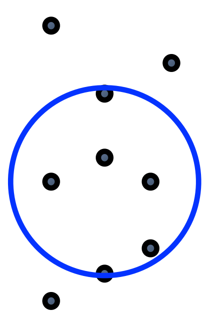
Input space: dimension $28^2 = 784$

The hidden manifold of data

Input space: dimension $28^2 = 784$



Manifold of handwritten digits in MNIST:



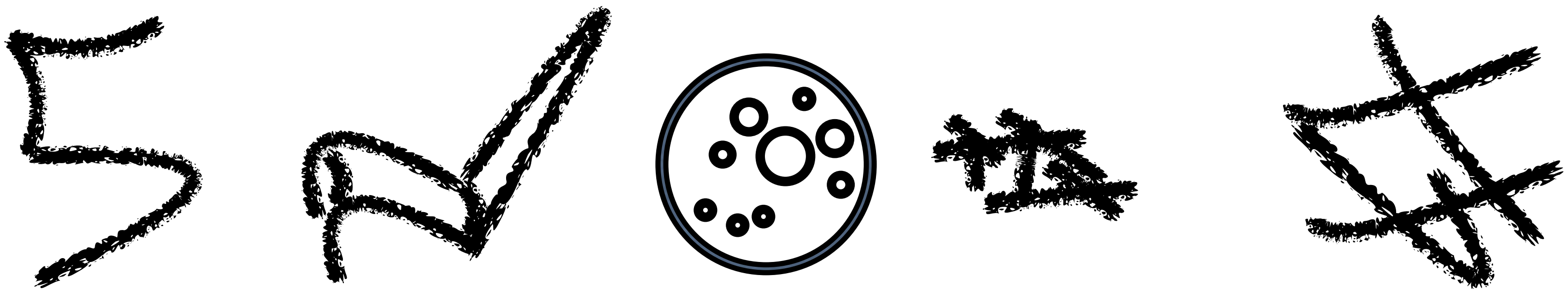
$p \simeq cR^d$

Nearest neighbors' distance : $R_{nn} \simeq p^{-1/d}$

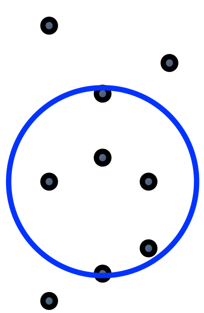
Grassberger Procaccia 83, Costa Hero 05, Heinz Audibert 05, Ansuini et al. 19, Spigler et al. 19…

The hidden manifold of data

Input space: dimension $28^2 = 784$



Manifold of handwritten digits in MNIST:

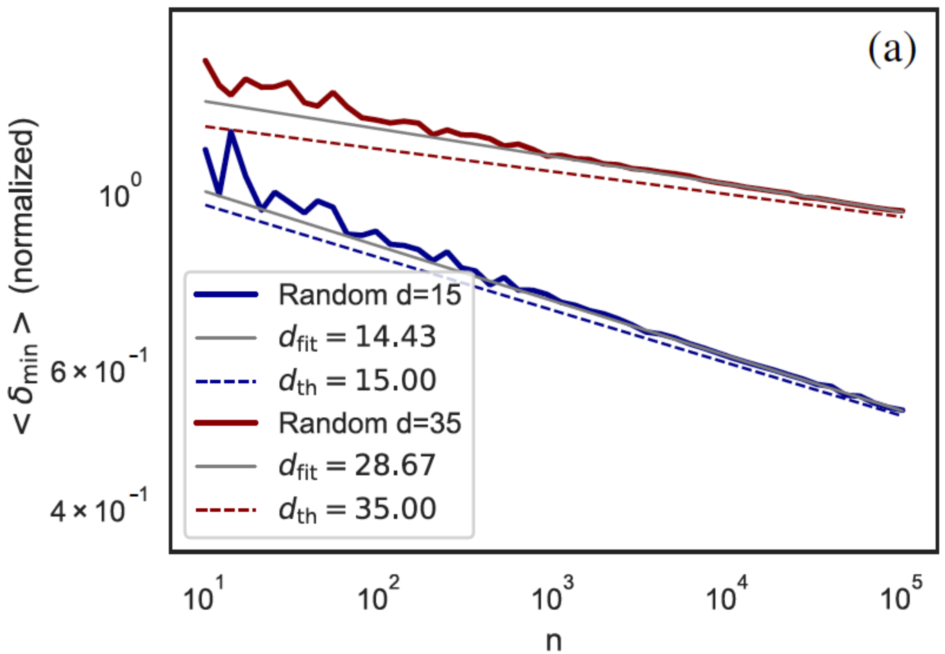


$p \simeq cR^d$

Nearest neighbors' distance : $R_{nn} \simeq p^{-1/d}$

Grassberger Procaccia 83, Costa Hero 05, Heinz Audibert 05, Ansuini et al. 19, Spigler et al. 19…

The hidden manifold of data



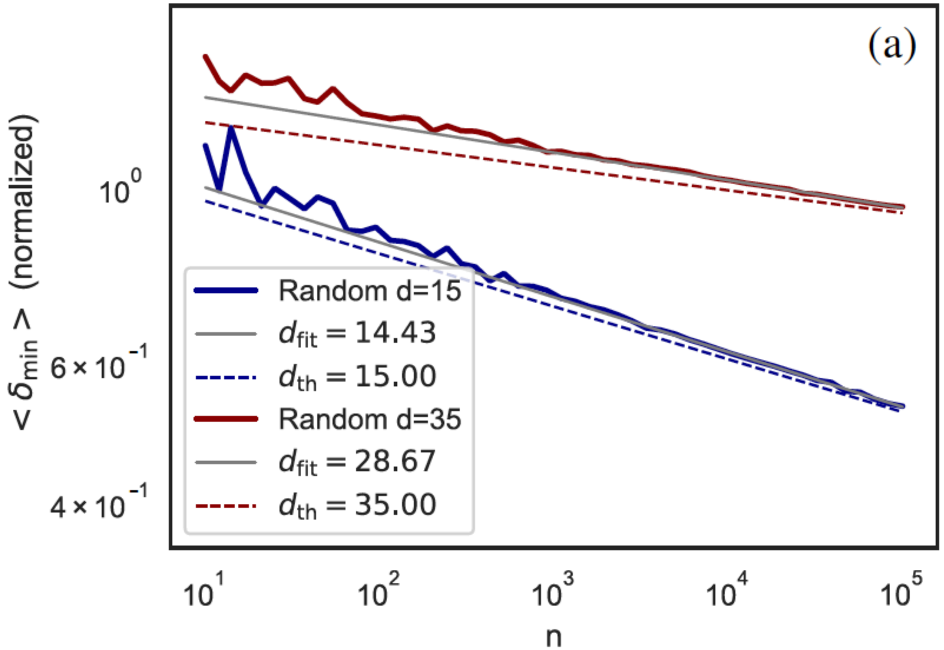
MNIST: $d = 784$

$d_{\text{eff}} \simeq 15$

Spigler et al. 19

Nearest neighbors' distance : $R_{nn} \simeq p^{-1/d}$

The hidden manifold of data

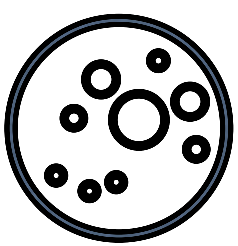


MNIST: $d = 784$

$d_{\text{eff}} \simeq 15$

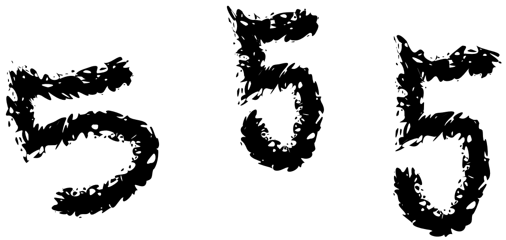
Spigler et al. 19

Nearest neighbors' distance : $R_{nn} \simeq p^{-1/d}$



The neural net should answer: this image does not seem to be a handwritten digit

Structure of the task: perceptual sub-manifolds



$$d_{\text{eff}}(5) \simeq 12$$

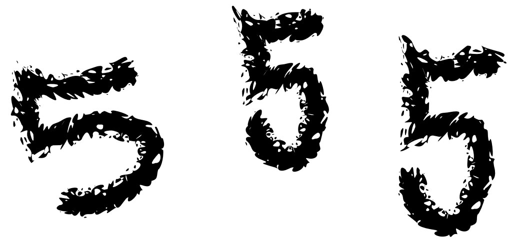
Hein Audibert 05

Table 7. Number of samples and estimated intrinsic dimensionality of the digits in MNIST.

1	2	3	4	5
7877	6990	7141	6824	6903
8/7/7	13/12/13	14/13/13	13/12/12	12/12/12
6	7	8	9	0
6876	7293	6825	6958	6903
11/11/11	10/10/10	14/13/13	12/11/11	12/11/11

MNIST problem: in the 15**-dim manifold** of handwritten digits, identify the 10 **perceptual sub manifolds** associated with each digit, of **dimensions between** 7 **and** 13…

Structure of the task: perceptual sub-manifolds



$d_{\text{eff}}(5) \simeq 12$

Hein Audibert 05

Table 7. Number of samples and estimated intrinsic dimensionality of the digits in MNIST.

1	2	3	4	5
7877	6990	7141	6824	6903
8/7/7	13/12/13	14/13/13	13/12/12	12/12/12
6	7	8	9	0
6876	7293	6825	6958	6903
11/11/11	10/10/10	14/13/13	12/11/11	12/11/11

MNIST problem: in the 15**-dim manifold** of handwritten digits, identify the 10 **perceptual sub manifolds** associated with each digit, of **dimensions between** 7 **and** 13…

… from an input in 784 dimensions!

An ensemble for the hidden manifold

Pattern $\mu:\quad X_{\mu i} = f\left[\frac{1}{\sqrt{R}}\sum_{r=1}^{R} C_{\mu r} F_{ir}\right]$

arXiv:1909.11500

S. Goldt, F. Krzakala MM L. Zdeborova

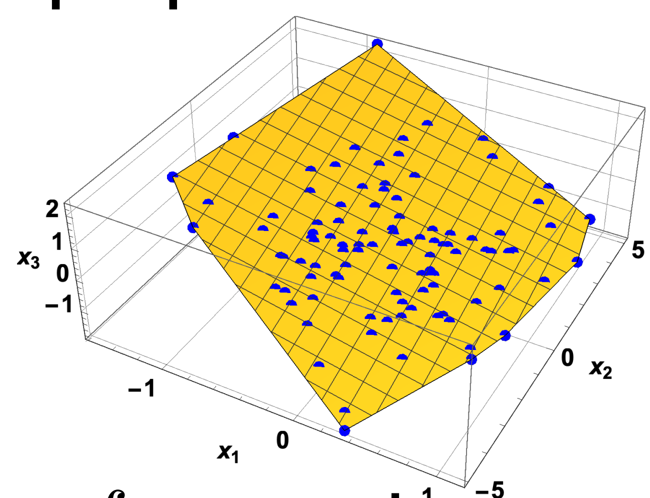
Data = input patterns built from R features $\vec{F}_r$

A feature is a N component vector in the input space

Each pattern is built from a weighted superposition of features (feature r has weight C_r):

$$\sum_{r=1}^{R} C_r \vec{F}_r$$

latent representation

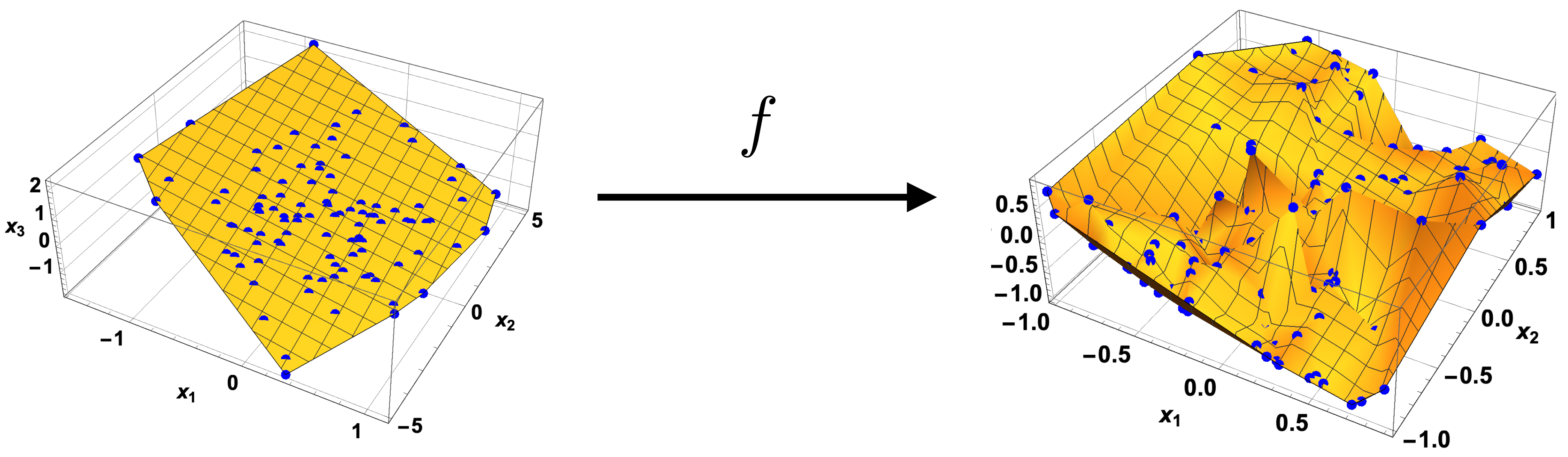


Then apply a nonlinear folding function f to each component

An ensemble for the hidden manifold

$$X_{\mu i} = f\left[\frac{1}{\sqrt{R}}\sum_{r=1}^{R} C_{\mu r}F_{ir}\right]$$

The R-dimensional data manifold is folded by applying the non-linear function f



An ensemble for the task

$$\vec{X}^{\mu} = f\left[\frac{1}{\sqrt{R}}\sum_{r=1}^{R} C_r^{\mu}\vec{F}_r\right]$$

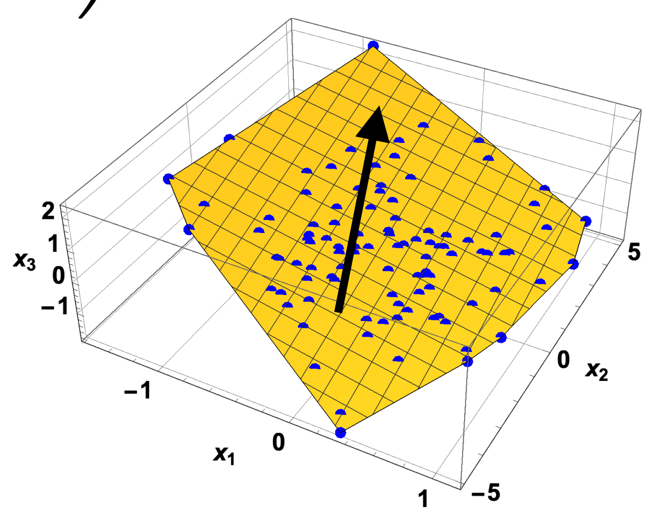
« Latent representation »: $\{C_r\}$

iid

Desired output = **function of latent representation**

Example: $$y^{\mu} = g\left(\sum_{r=1}^{R}\tilde{w}_r C_r^{\mu}\right)$$

(perceptron in hidden manifold)



An ensemble for the task

$$\vec{X}^{\mu} = f\left[\frac{1}{\sqrt{R}}\sum_{r=1}^{R} C_r^{\mu}\vec{F}_r\right]$$

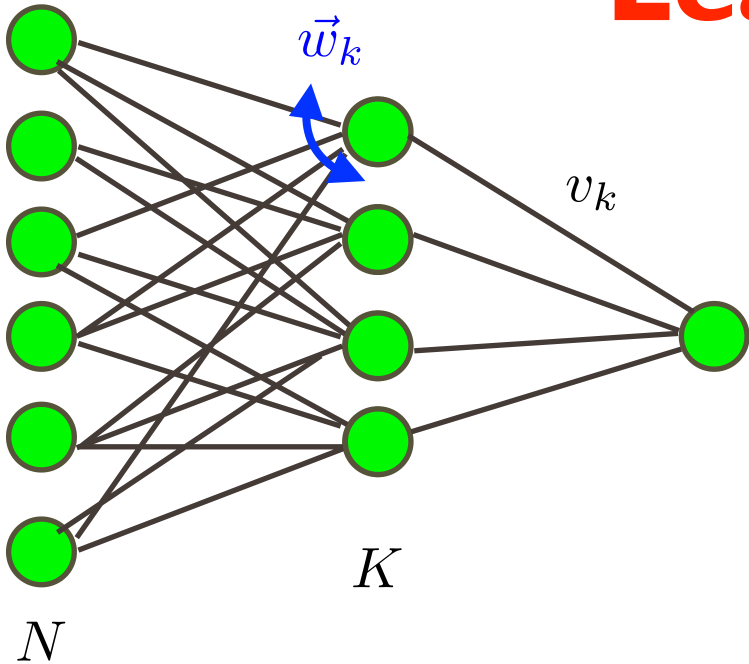
« Latent representation »: $\{C_r\}$

Desired output = **function of latent representation**

Examples: $$y^{\mu} = g\left(\sum_{r=1}^{R}\tilde{w}_r C_r^{\mu}\right)$$ (perceptron in latent space)

$$y^{\mu} = \sum_{m=1}^{M}\tilde{v}_m\, g\left(\sum_{r=1}^{R}\tilde{w}_{mr} C_r^{\mu}\right)$$ (2 layers nn in latent space)

Learning from HMM data



Learn using a 2-layer neural net, K hidden units

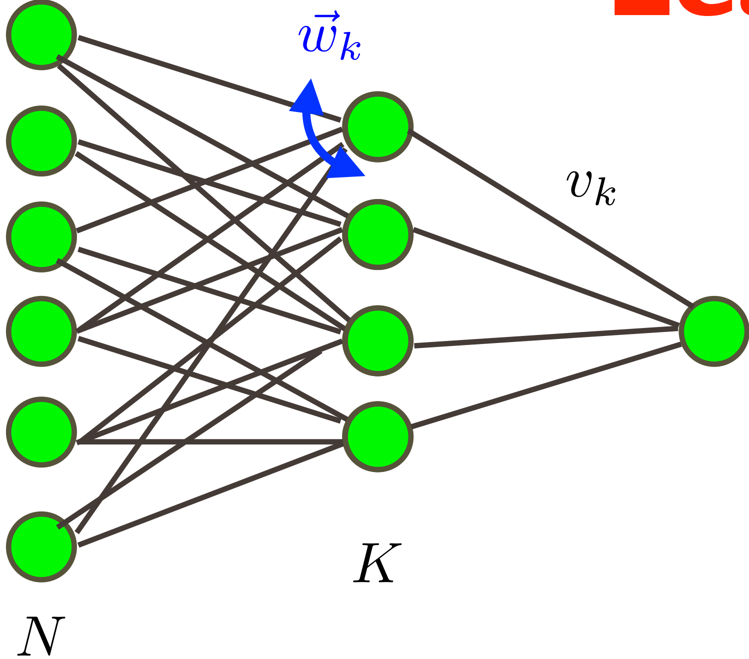
$$\phi(\vec{X}) = \sum_{k}^{K} v_k g\left(\vec{w}_k . \vec{X}/\sqrt{N}\right) ,$$

Training error $$\frac{1}{2P}\sum_{\mu=1}^{P}\left[\Phi(\vec{X}^{\mu}) - y^{\mu}\right]^2$$

or other loss function

Target label: generated from latent representation C_r^{μ}

Learning from HMM data



Learn using a 2-layer neural net, K hidden units

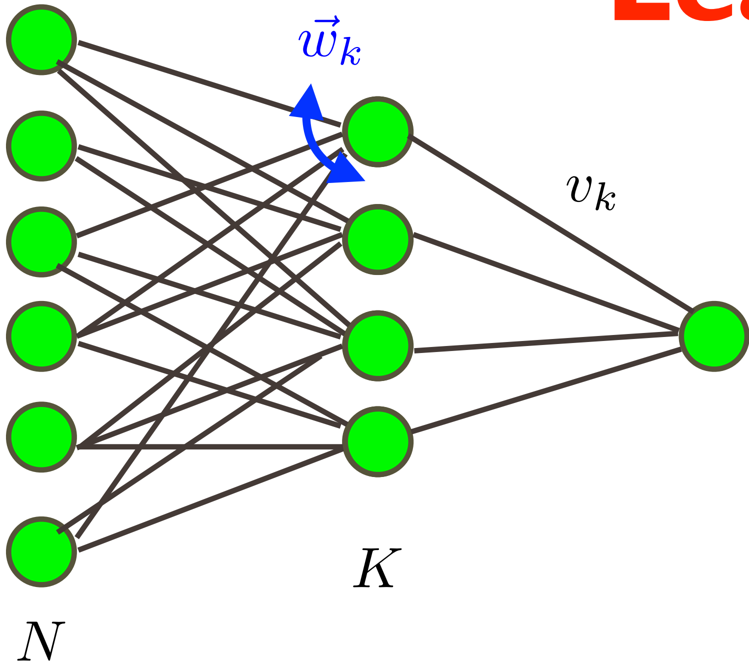
$$\phi(\vec{X}) = \sum_{k}^{K} v_k g\left(\vec{w}_k.\vec{X}/\sqrt{N}\right),$$

Training error $$\frac{1}{2P}\sum_{\mu=1}^{P}\left[\Phi(\vec{X}^{\mu}) - y^{\mu}\right]^2$$

or other loss function

Target label: generated from latent representation C_r^{μ}

Learning from HMM data



Learn using a 2-layer neural net, K hidden units

$$\phi(\vec{X}) = \sum_{k}^{K} v_k g\left(\vec{w}_k.\vec{X}/\sqrt{N}\right),$$

Training error $\displaystyle \frac{1}{2P}\sum_{\mu=1}^{P}\left[\Phi(\vec{X}^{\mu}) - y^{\mu}\right]^2$

or other loss function

Target label: generated from latent representation C_r^{μ}

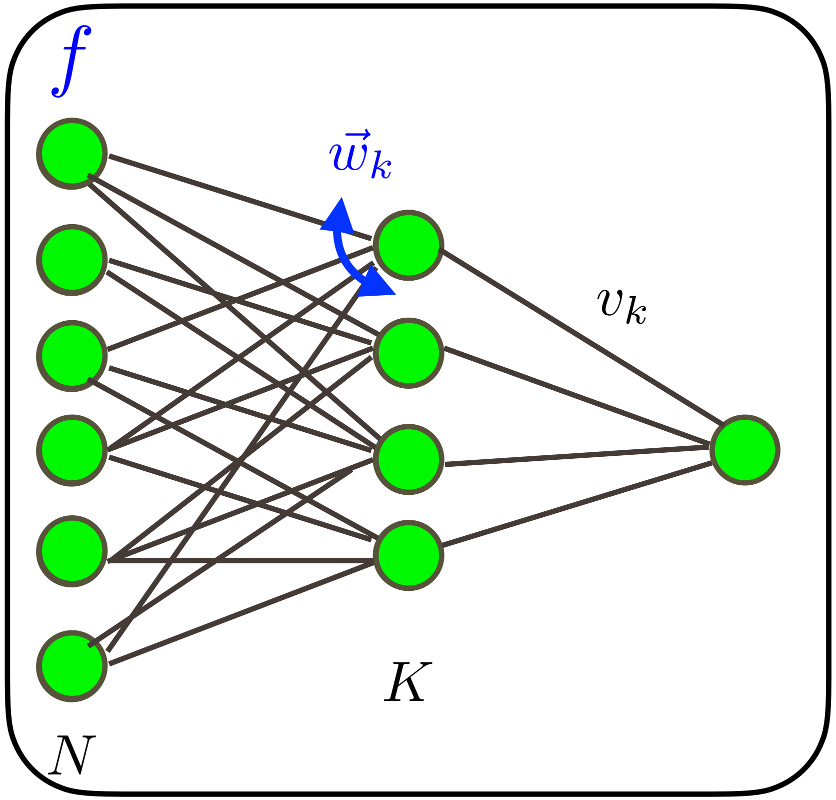
Generalization error: same with P^* new patterns

NB: Hidden manifold and random features

$$\vec{X}^{\mu} = f\left[\frac{1}{\sqrt{R}}\sum_{r=1}^{R} C_r^{\mu}\vec{F}_r\right]$$

Correlated components

iid



Learning a task with a iid database in R dimensions C_r^{μ}

In general not linearly separable. Embed it into a larger dimensional space of features $C_r^{\mu} \to X_i^{\mu}$

Embedding through quenched (or « lazily learnt ») matrix F, and nonlinearity f. Special case of HMM

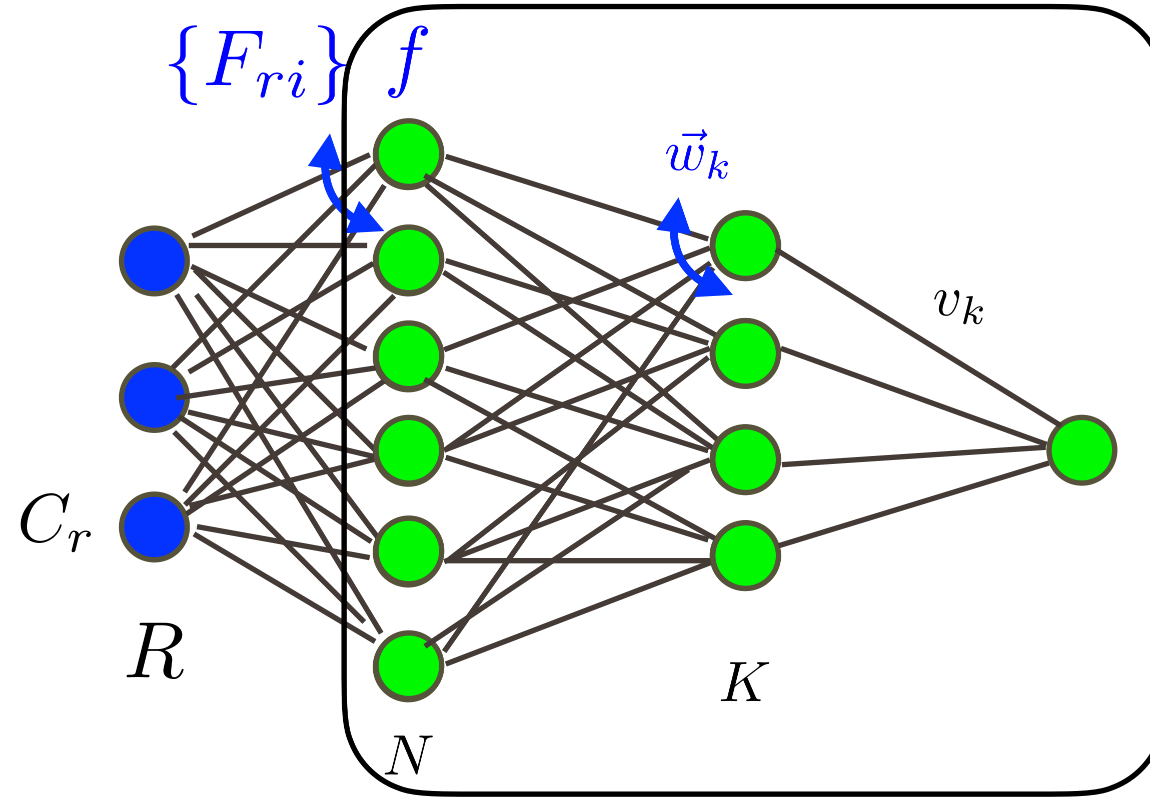
Connexion to Montanari Mei arXiv:1908.05335 and 1911.01544

NB: Hidden manifold and random features

$$\vec{X}^{\mu} = f\left[\frac{1}{\sqrt{R}}\sum_{r=1}^{R} C_r^{\mu}\vec{F}_r\right]$$

Correlated components

iid



Learning a task with a iid database in R dimensions C_r^{μ}

In general not linearly separable. Embed it into a larger dimensional space of features $C_r^{\mu} \to X_i^{\mu}$

Embedding through quenched (or « lazily learnt ») matrix F, and nonlinearity f. Special case of HMM

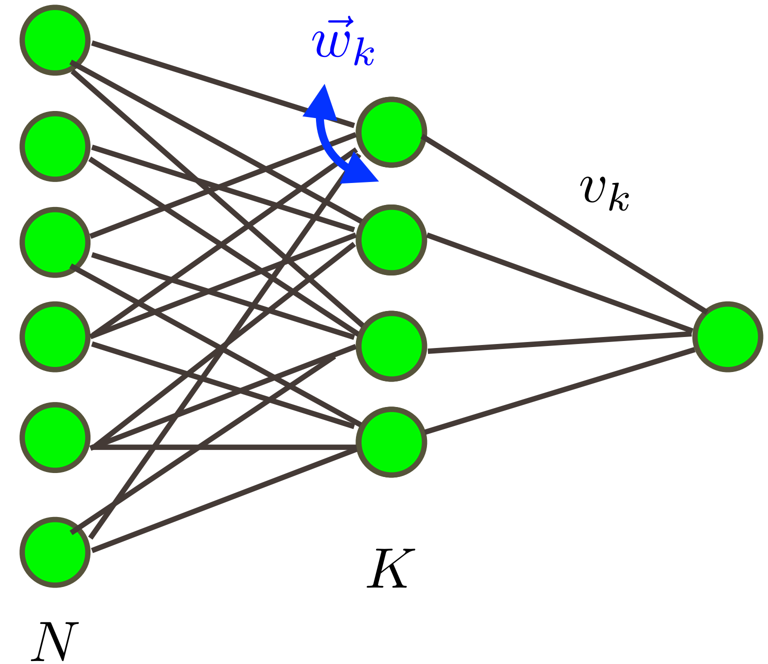
Connexion to Montanari Mei arXiv:1908.05335 and 1911.01544

Analytic study of the hidden manifold model

$$\vec{X} = f\left[\frac{1}{\sqrt{R}}\sum_{r=1}^{R} C_r \vec{F}_r\right]$$

Correlated components

iid



Solvable limit = **thermodynamic limit** with extensive latent dimension $N \to \infty$, $R \to \infty$, $P \to \infty$

With fixed $R/N = \gamma\,, \quad P/N = \alpha\,, \quad K$

Analytic study of the hidden manifold model

$$\vec{X} = f\left[\frac{1}{\sqrt{R}}\sum_{r=1}^{R} C_r \vec{F}_r\right]$$

Correlated components

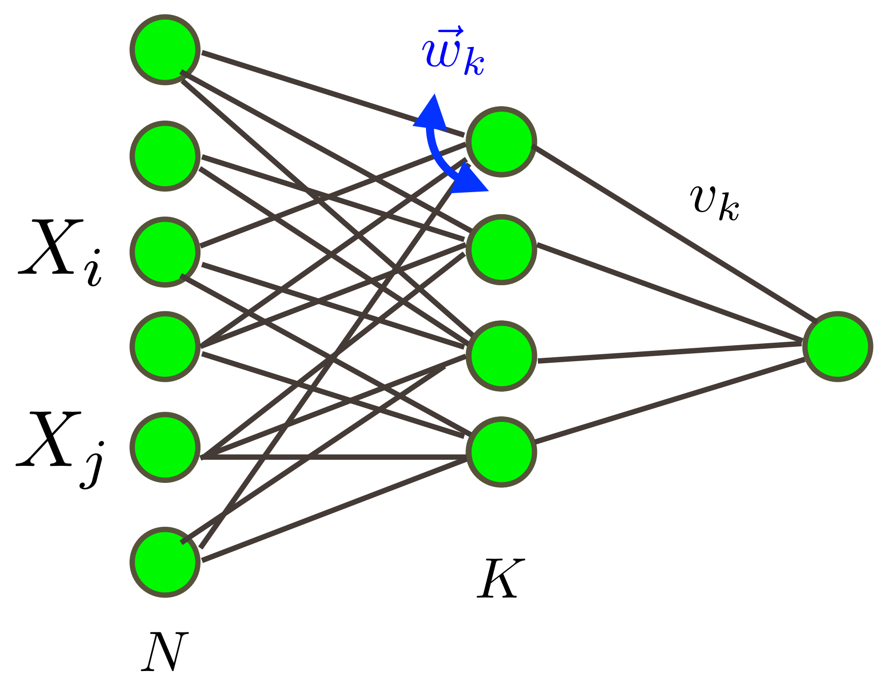
iid

balanced:

$$F_{ri} = O(1)$$

$$\frac{1}{N}\sum_i F_{ri}F_{si} = O(1/\sqrt{N})$$

$$\frac{1}{N}\sum_i F_{ri}F_{ri} = 1$$

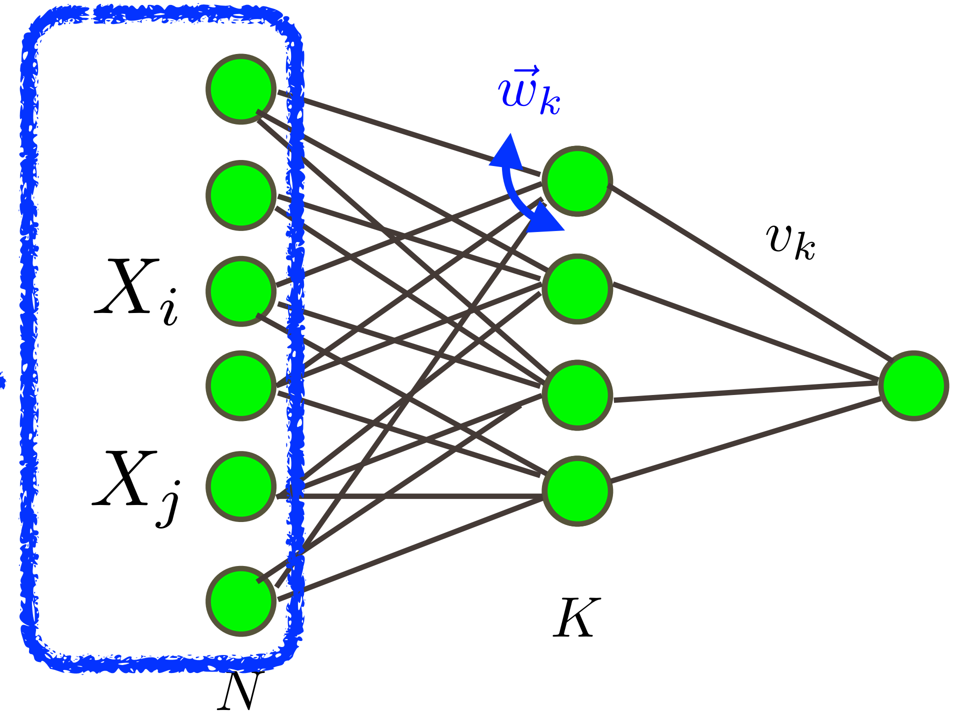


Analytic study of the hidden manifold model

$$\vec{X} = f\left[\frac{1}{\sqrt{R}}\sum_{r=1}^{R} C_r \vec{F}_r\right]$$

Correlated components

iid



$$X_i = f[u_i]$$

$$u_i = \frac{1}{\sqrt{R}}\sum_{r=1}^{R} C_r F_{ri}$$

Gaussian, weakly correlated $O(1/\sqrt{N})$ when F_{ri} are balanced and $O(1)$

$$\mathbb{E}\left(f[u_i]f[u_j]\right) = \langle f(u)\rangle^2 + \langle u f(u)\rangle^2 \mathbb{E}\left(u_i u_j\right)$$

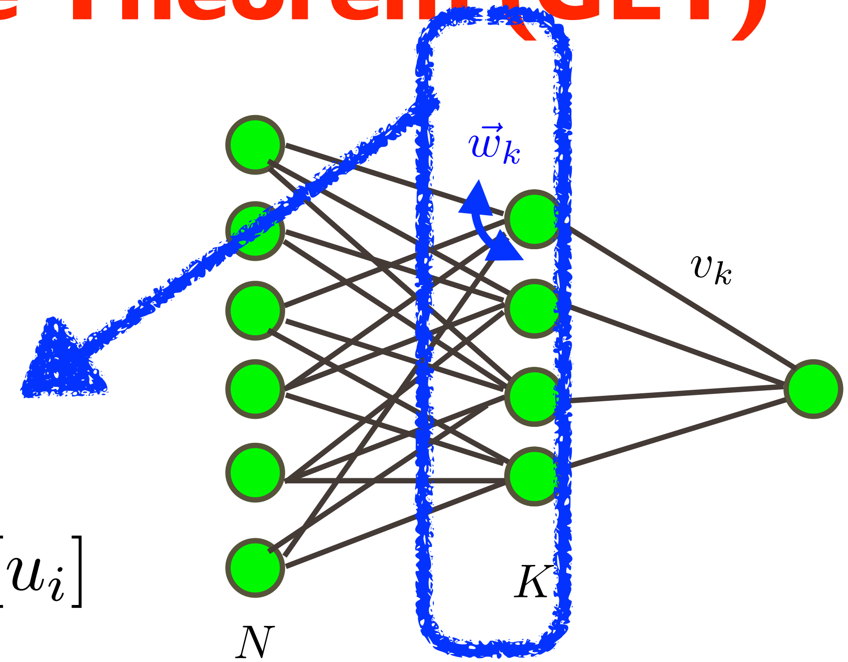
u Gaussian $\mathcal{N}(0,1)$

Gaussian Equivalence Theorem (GET)

$$u_i = \frac{1}{\sqrt{R}} \sum_{r=1}^{R} \boxed{C_r} F_{ri}$$

iid

$$X_i = f[u_i]$$



Inputs of hidden units: $\lambda^k = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} w_i^k f[u_i]$

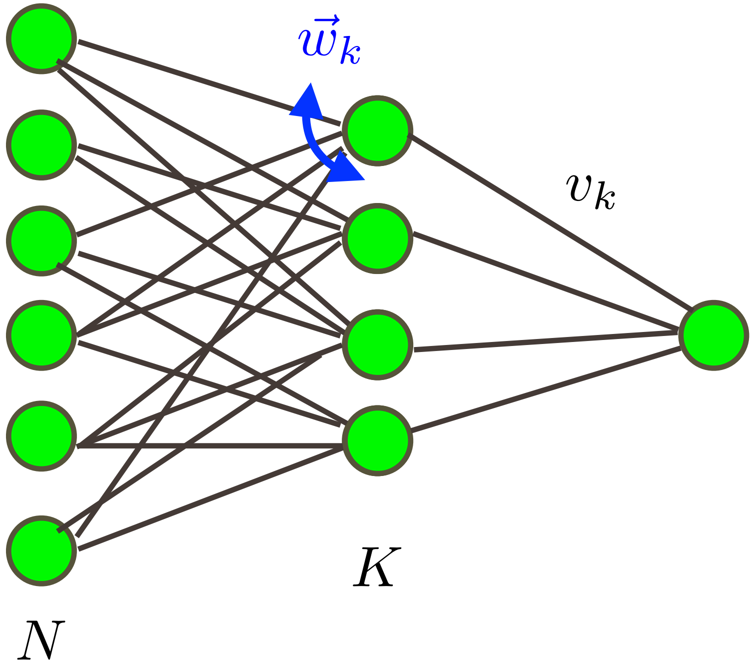
GET: In the thermodynamic limit, the variables λ^k have a Gaussian distribution, with covariance

$$\mathbb{E}[\tilde{\lambda}^k \tilde{\lambda}^\ell] = (c - a^2 - b^2) W^{k\ell} + b^2 \Sigma^{k\ell}$$

$$W^{k\ell} \equiv \frac{1}{N} \sum_{i=1}^{N} w_i^k w_i^\ell \qquad \Sigma^{k\ell} \equiv \frac{1}{R} \sum_{r=1}^{R} S_r^k S_r^\ell \qquad S_r^k \equiv \frac{1}{\sqrt{N}} \sum_{i=1}^{N} w_i^k F_{ir}$$

$c = \langle f(u)^2 \rangle \qquad a = \langle f(u) \rangle \qquad b = \langle u f(u) \rangle \qquad u$ Gaussian $\mathcal{N}(0,1)$

Online learning of Hidden Manifold Model



Learn using a 2-layer neural net, K hidden units

$$\Phi\left(\vec{X}\right) = \sum_{k=1}^{K} g\left(\vec{w}^k.\vec{X}/\sqrt{N}\right)$$

$$\vec{X} = f\left[\frac{1}{\sqrt{R}}\sum_{r=1}^{R} C_r \vec{F}_r\right]$$

$\vec{X}$ = inside hidden R-dimensional manifold, folded by function f

Desired output given constructed from latent representation

$$\Phi_t(\vec{X}) = \sum_{m=1}^{M} \tilde{g}\left(\sum_{r=1}^{R} \tilde{w}_r^m C_r\right)$$

Online learning: ODE for SGD

Evolution of the weights during learning

D Saad and S Solla 95, Biehl and Schwarze 95, …

$$\left(w_i^k\right)^{\mu+1} - \left(w_i^k\right)^{\mu} = -\frac{\eta}{\sqrt{N}}\Delta g'(\lambda^k) f(u_i)$$

$$\Delta = \sum_{\ell=1}^{K} g(\lambda^\ell) - \sum_{m=1}^{N} \tilde{g}(\nu^m)$$

New pattern (and therefore new latent representation C_r) at each time

GET: λ^k and ν^m are Gaussian, and the learning dynamics can be analyzed by ordinary differential equations for order parameters like

$$W^{k\ell} \equiv \frac{1}{N}\sum_{i=1}^{N} w_i^k w_i^\ell$$

Order parameters

$$W^{k\ell} = \frac{1}{N} \sum_i w_i^k w_i^\ell$$

$$\Sigma^{k\ell} = \frac{1}{R} \sum_{r=1}^{R} S_r^k S_r^\ell$$

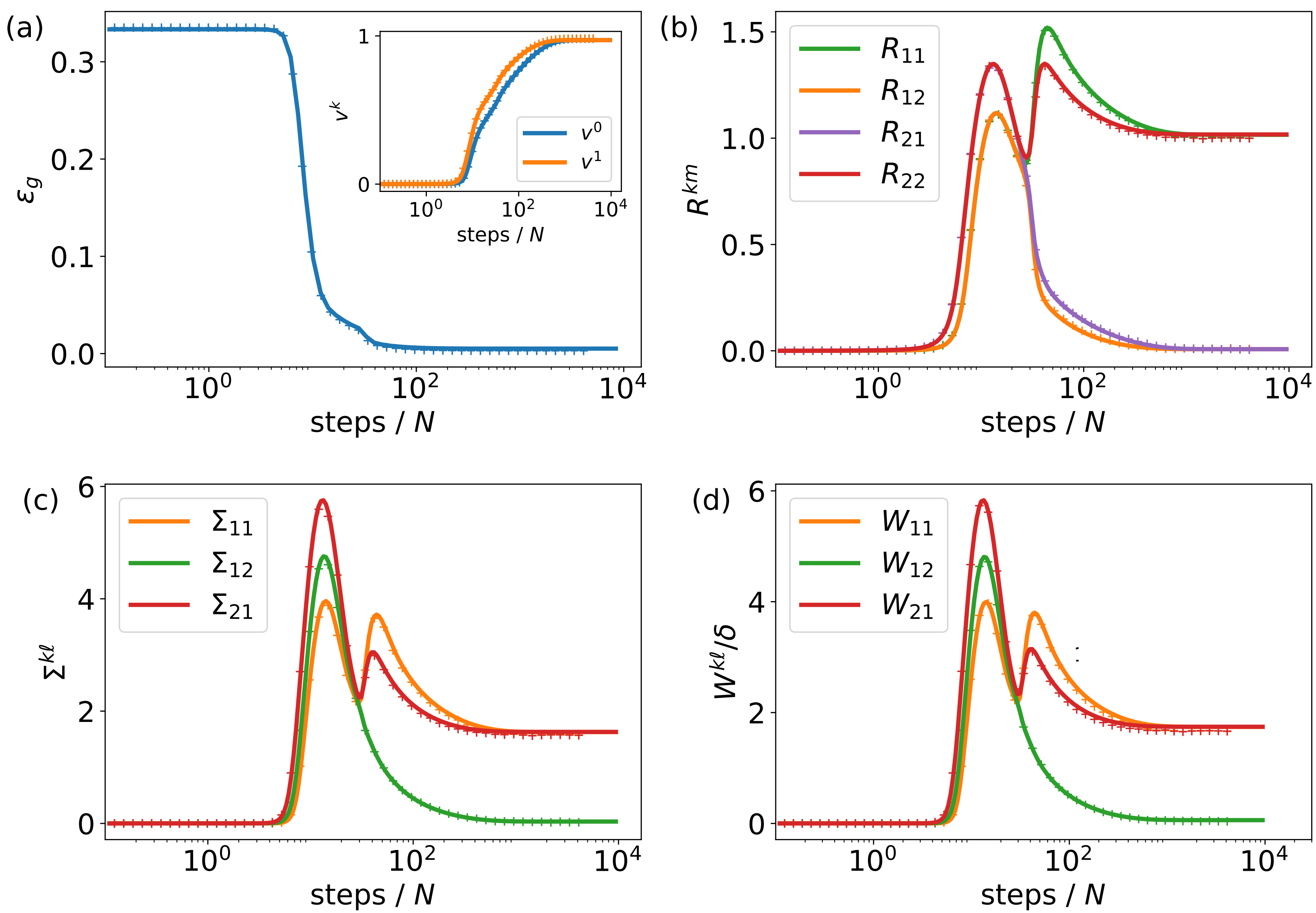
$$R^{km} = \frac{1}{R} \sum_{r=1}^{R} S_r^k \tilde{w}_r^m$$

where

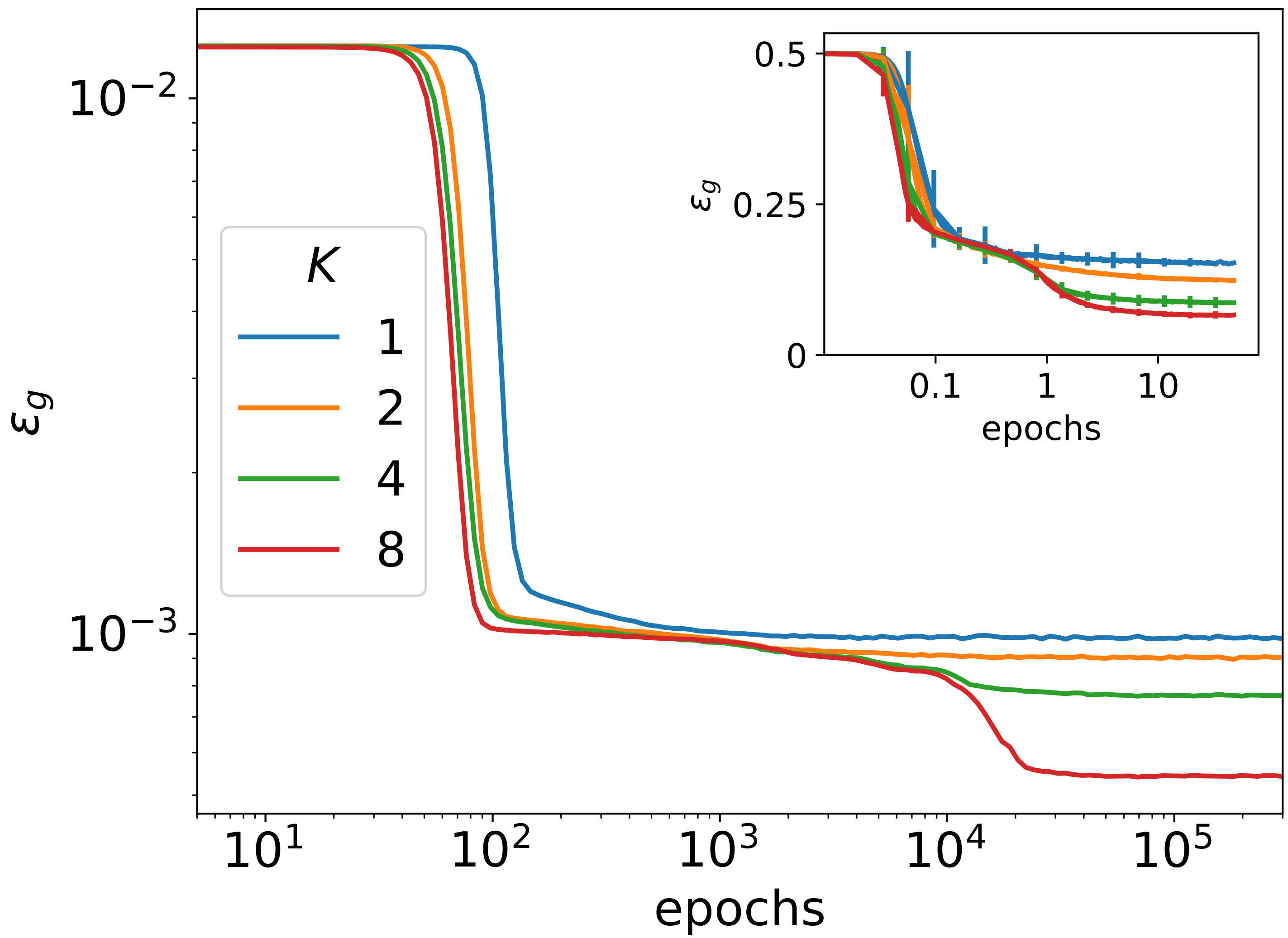
$$S_r^k = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} w_i^k F_{ir}$$

NB: mixed order parameter R^{km} measures the correlation of pre-activation of neuron k in the student and the weight m in the latent task. First project student's weight to latent space (S_r^k), then measure overlap to teacher

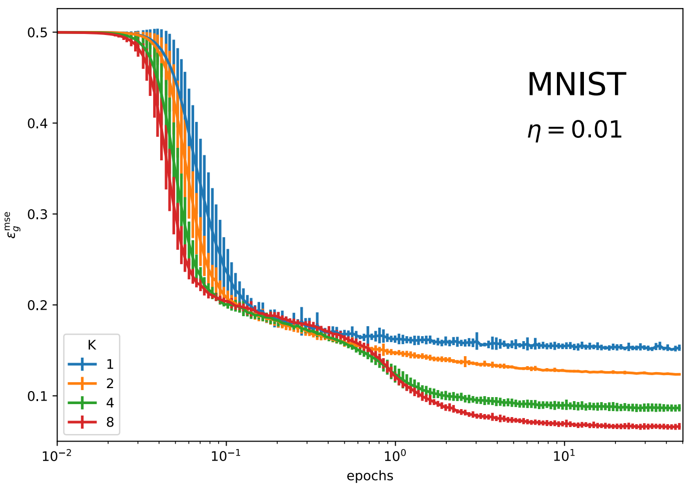
ODE Theory vs simulations N=10000, D=100, M=2, K=2 specializes after 50.N steps

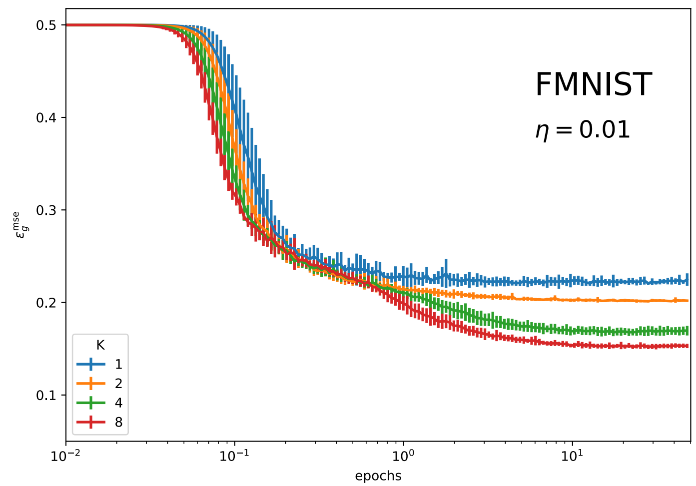


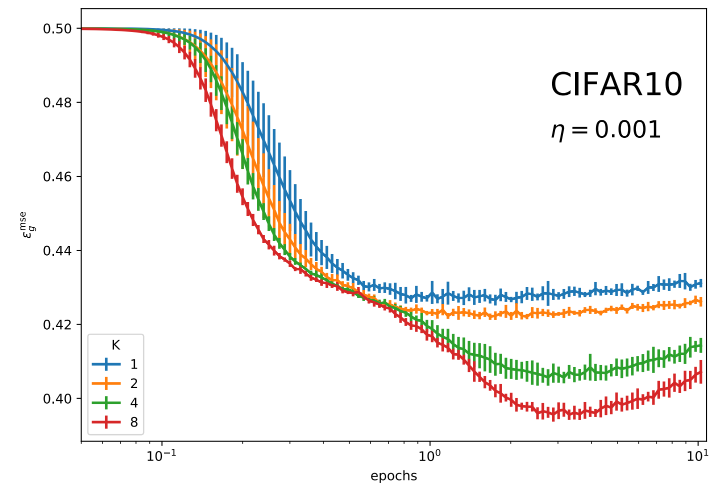
Larger second layer allows better learning after specialization

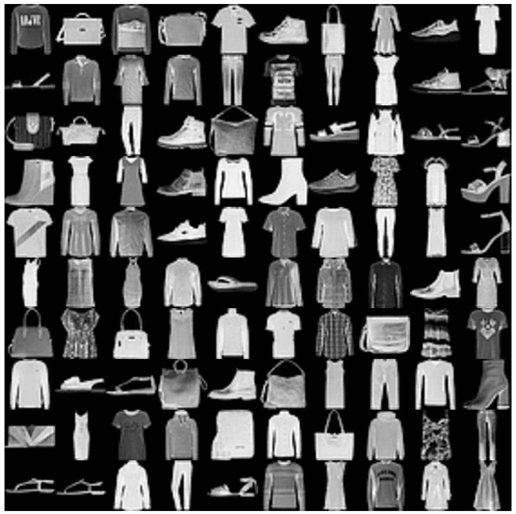


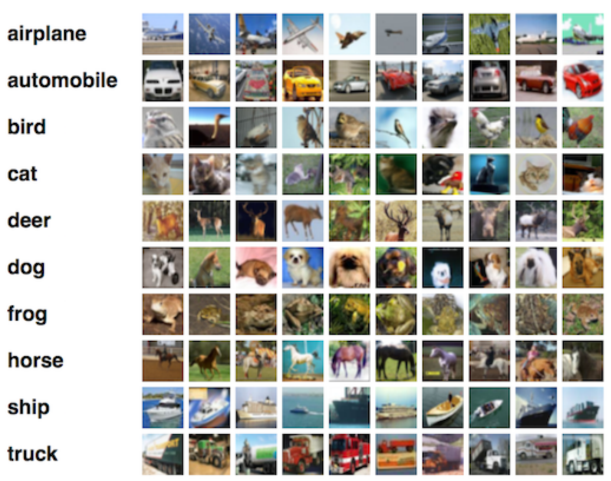
Larger second layer allows better learning: experiments on databases (erf)



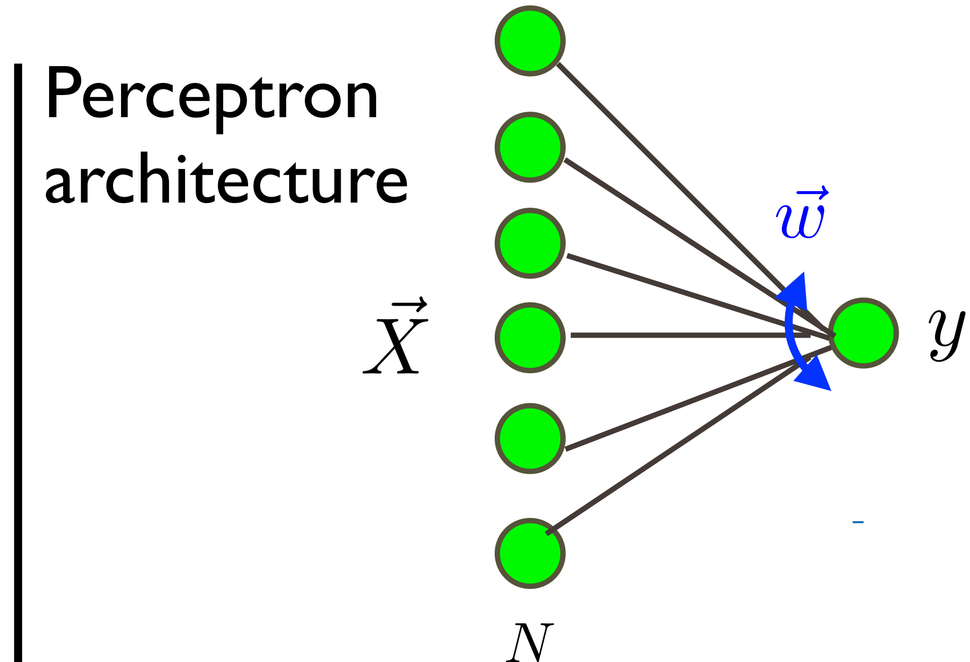








« Full batch »: perceptron learning and generalized linear regression



$$y_w\left(\vec{X}\right) = g\left(\vec{w}.\vec{X}\right)$$

« Full batch »: perceptron learning and generalized linear regression

$$\vec{X}^{\mu} = f\left[\frac{1}{\sqrt{R}}\sum_{r=1}^{R} C_r^{\mu}\vec{F}_r\right]$$

Learn from database of P patterns lying in a hidden manifold

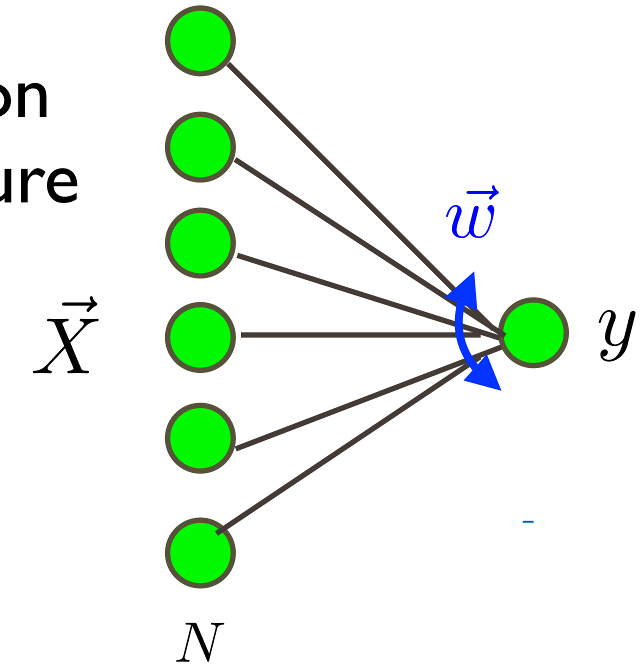
Target output = latent task

$$y^{\mu} = \Phi_t(\vec{X}^{\mu}) = \tilde{g}\left(\sum_{r=1}^{R}\tilde{w}_r\, C_r^{\mu}\right)$$

- Classification $\tilde{g}(z) = \mathrm{Sign}(z)$

- Regression $\tilde{g}(z) = z$

Perceptron architecture



$$y_w\left(\vec{X}\right) = g\left(\vec{w}.\vec{X}\right)$$

« Full batch »: perceptron learning and generalized linear regression

$$\vec{X}^{\mu} = f\left[\frac{1}{\sqrt{R}}\sum_{r=1}^{R} C_r^{\mu}\vec{F}_r\right]$$

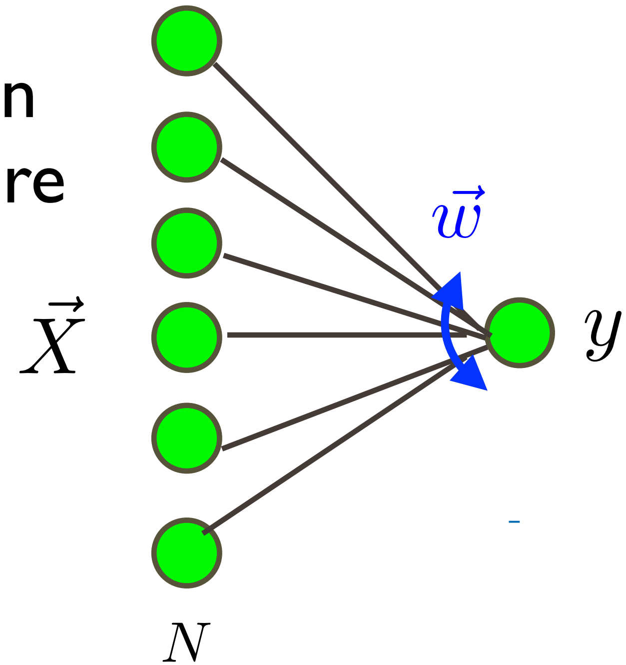
Learn from database of P patterns lying in a hidden manifold

Target output = latent task

$$y^{\mu} = \Phi_t(\vec{X}^{\mu}) = \tilde{g}\left(\sum_{r=1}^{R}\tilde{w}_r\, C_r^{\mu}\right)$$

- Classification $\tilde{g}\left(z\right) = \mathrm{Sign}(z)$
- Regression $\tilde{g}\left(z\right) = z$

Perceptron architecture



$$y_w\left(\vec{X}\right) = g\left(\vec{w}.\vec{X}\right)$$

Learning = minimize « loss»

$$E = \sum_{\mu=1}^{P}\epsilon\left(y^{\mu}, g(\vec{w}.\vec{X}^{\mu})\right) + \frac{\lambda}{2}\vec{w}^2$$

Replica study

F. Gerace, B. Loureiro, F. Krzakala, MM, L. Zdeborova

In short

Gardner's computation: typical volume of weight space compatible with the data $\left\{\vec{X}_\mu, \Phi_t(\vec{x}_\mu)\right\}$. Evaluated with replicas

The volume can be written in terms of the local input fields to the hidden variables, $\lambda_\mu{}^a$.

GET: these are Gaussian variables,
independent for different patterns,
correlated for one given pattern.

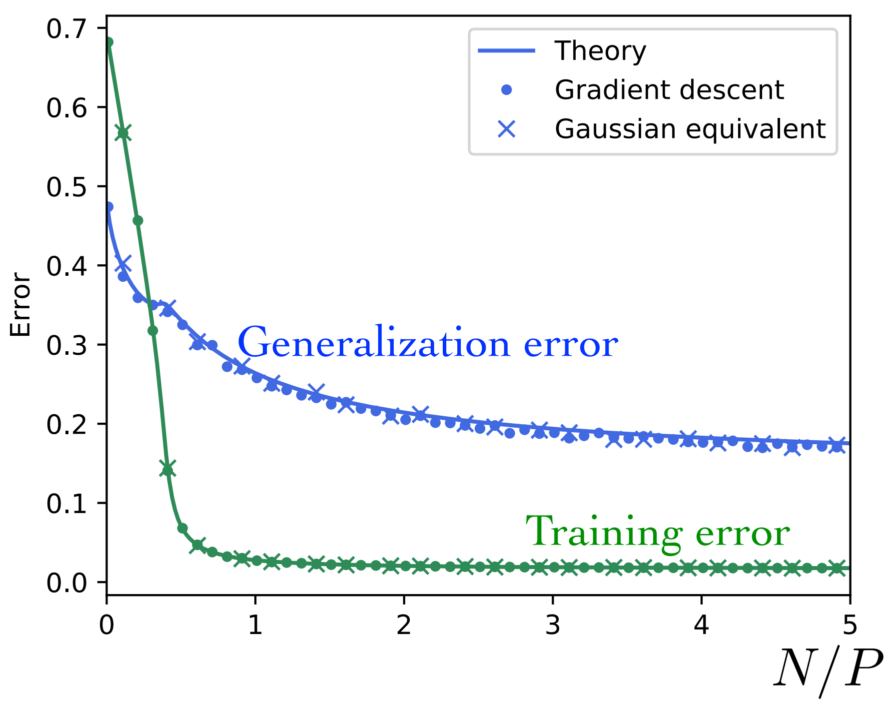
In short

Gardner's computation: typical volume of weight space compatible with the data $\left\{\vec{X}_\mu, \Phi_t(\vec{x}_\mu)\right\}$. Evaluated with replicas

The volume can be written in terms of the local input fields to the hidden variables, $\lambda_\mu{}^a$.

GET: these are Gaussian variables, independent for different patterns, correlated for one given pattern.

Theory vs simulations. Classification, logistic loss, « sign » non-linearities,
$R = 200 \quad P = 600 \qquad \lambda = 10^{-3}$

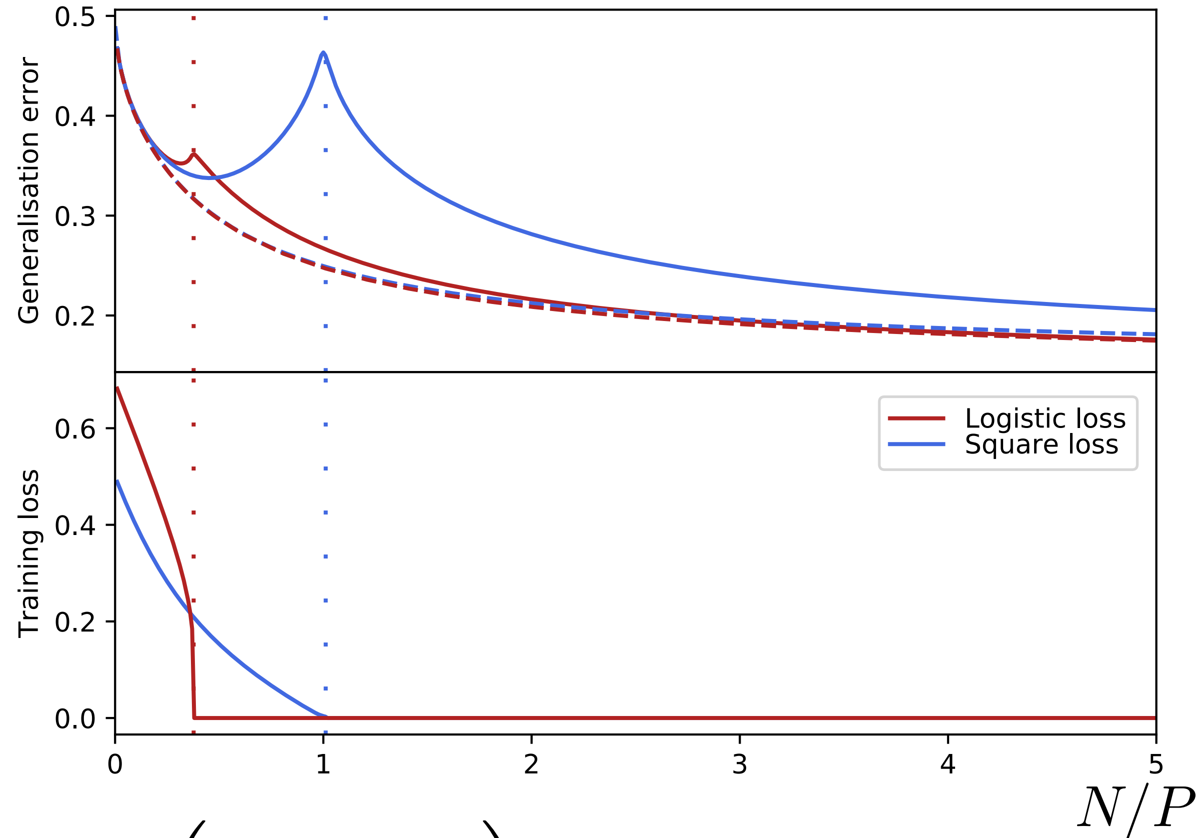


« Double descent »

Opper and Kinzel 1995
Spigler et al. 2019
Belkin et al. 2019

$$R/N = 1/3$$

$$\lambda = 10^{-4}$$



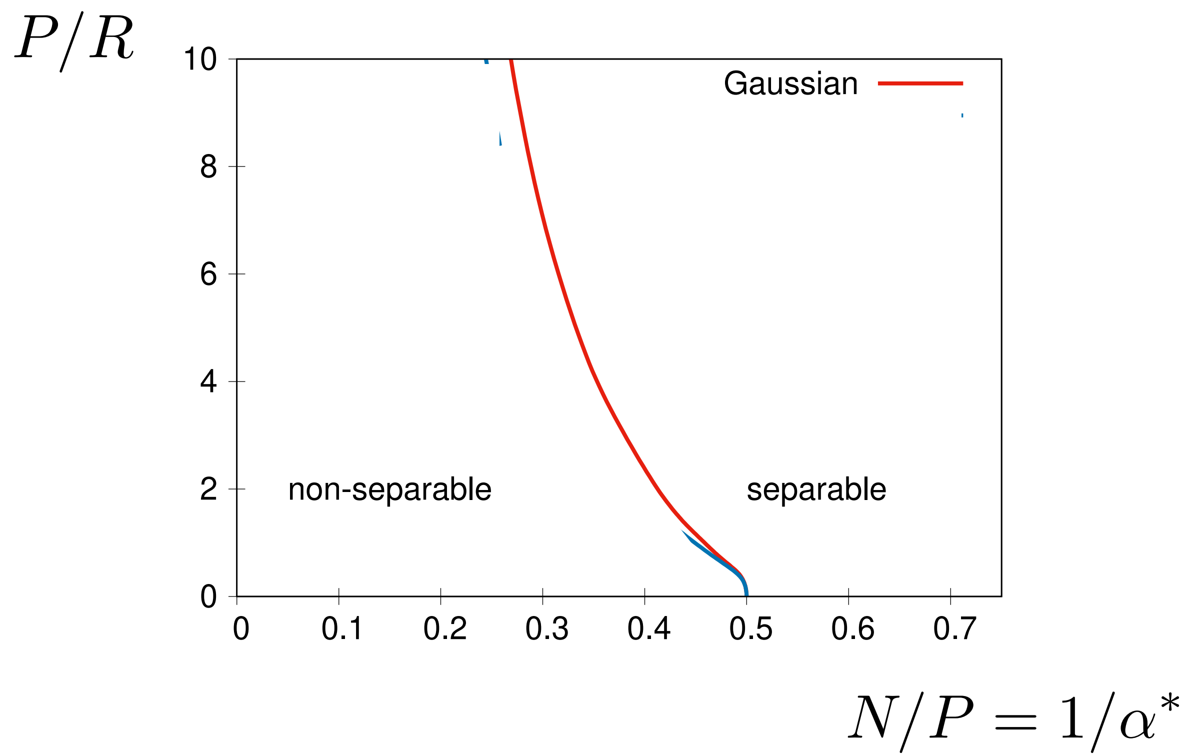
Classification task: $y^\mu = \mathrm{Sign}\left(\sum_r \tilde{w}_r C_r^\mu\right)$

Square loss: minimize $\sum_\mu (y^\mu - \vec{w}.\vec{X}^\mu)^2$ zero for $P < N$

« capacity » $\alpha^* = P/N = 1$

Logistic loss: « capacity » $\alpha^* = P/N > 2$

NB Phase diagram for learning: Threshold of linear separability



NB: at large R, the data matrix entries are close to iid. Cover's result $\alpha^* = 2$

Statistical physics for machine learning: Requires better ensembles for data

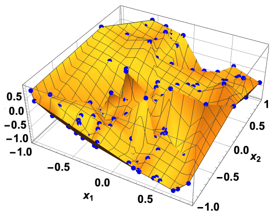
- Data in submanifolds
- Combinatorial structure

Hidden Manifold Model

Data has « Latent representation »: $\{C_r\}$

Desired output (task) = function of latent representation

Example $y = g\left(\sum_{r=1}^{R} \tilde{w}_r C_r\right)$ $\vec{X} = f\left[\frac{1}{\sqrt{R}} \sum_{r=1}^{R} C_r \vec{F}_r\right]$



- Good learning and generalization phenomenology
- Can be studied analytically : online learning and full batch in the limit where $R = O(N)$, thanks to a Gaussian Equivalence Theorem

Smart Inference for Covid19 tracing using message passing

CO-AUTHORS: A. BAKER, F. KRZAKALA, M. MÉZARD, M. REFINETTI, S. SARAO MANNELLI, L ZDEBOROVA, (ENS -PSL AND UNIV PARIS SACLAY)

COLLABORATING WITH: A. BRAUNSTEIN, LUCA DALL ASTA, ALESSANDRO INGROSSO, INDACO BIAZZO, ANNA PAOLA MUNTONI, (TORINO)

DISCUSSIONS WITH: YOSHUA BENGIO, IRINA RISH, LUCA FERRETTI, IVAN BESTVINA, (MILA)

Information about individuals (age, symptoms,…), known by each individual

Information about contacts (time, duration), known by the two individual in contact

Pb: Infer the probability that each individual be infected

Message-passing provides an efficient solution, without need for a central system, and based on simple exchange of messages that can be encrypted

Smart Inference of People At Risk (SIPAR)

Risk can be estimated more accurately than the list of contacts. Every individual should account for increased risk of his recent contacts and spread the information to other contacts

Basic tool: Susceptible-Infected-Recovered (SIR) model for individuals

Susceptible individuals (S) → Can be infected

Infected individuals (I) → Can infect others

Removed individuals (R) → Cannot spread or be infected

Parameters: $\lambda_{ij}(t)$ Infection rate from j to i at time t (contact)

μ_i Recovery rate

Estimate probabilities $P_S^j(t)$ $P_I^j(t)$ $P_R^j(t)$

Smart Inference of People At Risk (SIPAR)

Mean-field equations:

Can incorporate the knowledge from contacts' infections (backtrack)

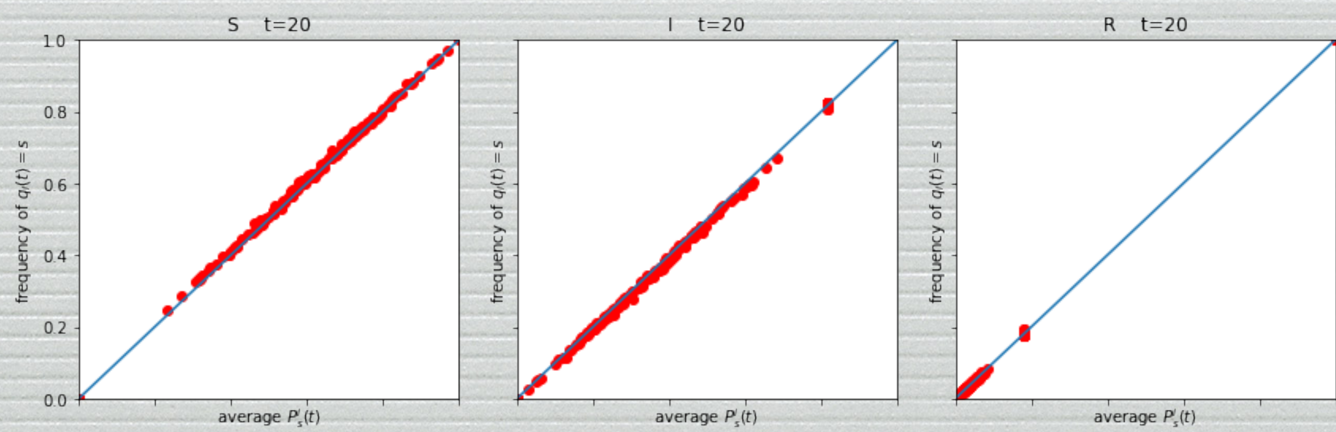
$$P_S^i(t+1) = P_S^i(t)\left(1 - \sum_{j \in \partial i(t)} P_I^j(t)\lambda_{ij}(t)\right)$$

$$P_R^i(t+1) = P_R^i(t) + \mu_i P_I^i(t)$$

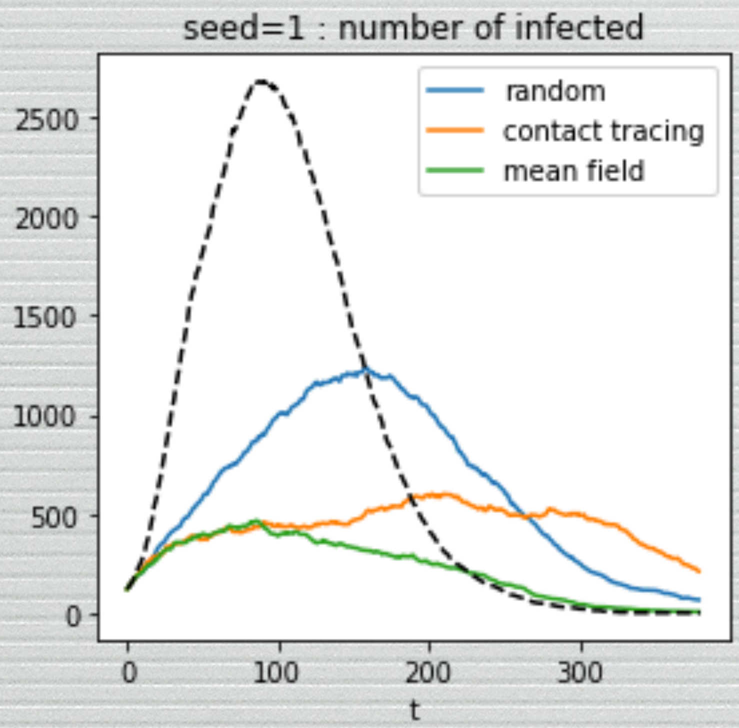
$$P_I^i(t+1) = P_I^i(t) + P_S^i(t) \sum_{j \in \partial i(t)} P_I^j(t)\lambda_{ij}(t) - \mu_i P_I^i(t)$$

More elaborate = dynamic message passing (cavity equations)

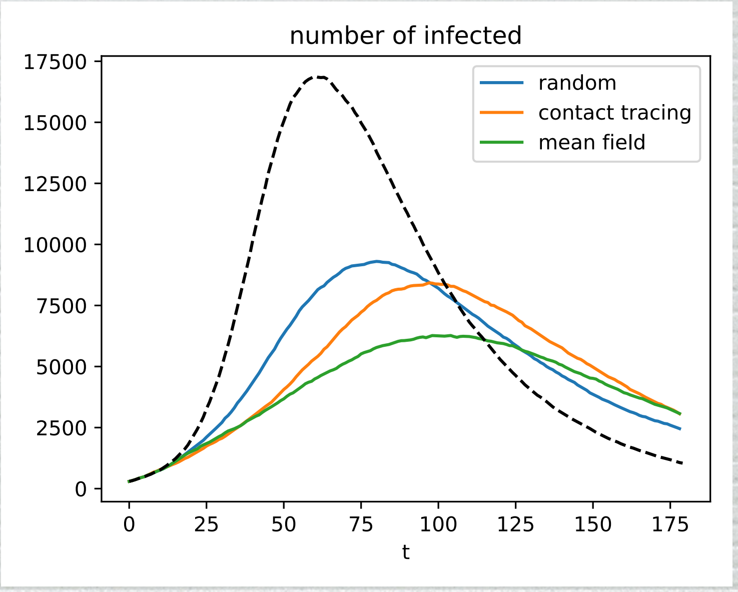
Lokhov, MM, Ohta, Zdeborova PRE 2014, PRE 2015



Comparing tracing and SIPAR : quarantine all symptomatic tested positive and test more according to tracing, or smart inference ranking



Random geometric contact graph in 2D, scale 1.1,
daily on average 7.4 contacts.
Population size= 10 000,
λ=0.02 , μ=0.03. Initially 20 infected τ=5, δ=15.
Tests 7-21



Contact graph provided by Ferretti and Hinch.
daily on average 12.7 contacts.
Population size= 50 000,
λ=0.01 , μ=0.03. Initially 10 infected τ=5, δ=15.
Tests 50-100

The End

Thanks to *Florent Krzakala and Lenka Zdeborova*, as well as *Sebastian Goldt, Federica Gerace, Bruno Loureiro, Galen Reeves, Antoine Baker, Maria Refinetti, Stefano Sarao…*

MM, Phys.Rev. E 95 (2017),022117

S. Goldt, F. Krzakala, MM, L. Zdeborova, arXiv:1909.11500

F. Gerace, B. Loureiro, F. Krzakala, MM, L. Zdeborova, arXiv:2002.09339

F. Gerace, B. Loureiro, F. Krzakala, MM, L. Zdeborova, arXiv:2002.09339

A Baker, F. Krzakala, MM, M. Refinetti, S. Sarao Mannelli, in progress. https://github.com/sphinxteam/sir_inference