Exactly solvable models for statistical machine learning:

A study of over-parametrization

Florent Krzakala

Institut Universitaire de France

Empirical Risk Minimisation

$$(\mathbf{X}_i \in \mathbb{R}^d, y_i \in \mathbb{R}), i = 1, \ldots, n \qquad \mathscr{R} = \frac{1}{n} \sum_{i=1}^{n} \mathscr{L}(y_i, f_\theta(\mathbf{X}_i))$$

Loss : $\mathscr{L}(y, h)$ **Square Loss** $L(f(\vec{x}), y) = (1 - yf(\vec{x}))^2$

Logistic loss/Cross-entropy $L(f(\vec{x}), y) = \frac{1}{\ln 2} \ln(1 + e^{-yf(\vec{x})})$

Ex: linear network

Model: $f_\theta(\mathbf{X}) = \theta \cdot \mathbf{X}$

Empirical Risk Minimisation

$$(\mathbf{X}_i \in \mathbb{R}^d, y_i \in \mathbb{R}), i = 1, \ldots, n \qquad \mathscr{R} = \frac{1}{n} \sum_{i=1}^{n} \mathscr{L}(y_i, f_\theta(\mathbf{X}_i))$$

Loss : $\mathscr{L}(y, h)$ **Square Loss** $L(f(\vec{x}), y) = (1 - yf(\vec{x}))^2$

Logistic loss/Cross-entropy $L(f(\vec{x}), y) = \frac{1}{\ln 2} \ln(1 + e^{-yf(\vec{x})})$

Ex: neural networks

Model: $$f_\theta(\mathbf{X}) = \eta^{(0)} \left(\mathbf{W}^{(0)} \eta^{(1)} \left(\mathbf{W}^{(1)} \ldots \eta^{(L)} \left(\mathbf{W}^{(L)} \cdot \mathbf{X} \right) \right) \right)$$

$$\theta = \{\mathbf{W}^{(0)}, \mathbf{W}^{(1)}, \ldots, \mathbf{W}^{(L)}\}$$

Statistical learning 101

Supervised Binary classification

- Dataset, *m* examples $\{y^{(\mu)}, \vec{x}^{(\mu)}\}_{\mu=1}^{m}$
- Function class $f_{\vec{w}} \in \mathcal{F}$

Rademacher complexity

Theorem: Uniform convergence

With probability at least 1-δ,

$$\forall f_{\vec{w}} \in \mathcal{F}, \quad \epsilon_{\text{gen}}(f_{\vec{w}}) - \epsilon_{\text{train}}^{m}(f_{\vec{w}}) \leq \mathfrak{R}_m(\mathcal{F}) + \sqrt{\frac{\log(1/\delta)}{m}}$$

Generalization error

Train error

Classical result

Rademacher Complexity is bounded by VC dimension for some constant C

$$\mathfrak{R}_m(\mathcal{F}) \leq C\sqrt{\frac{d_{\text{VC}}(\mathcal{F})}{m}}$$

UNDERSTANDING MACHINE LEARNING
WORST CASE ANALYSIS IS NOT ENOUGH

- Deep learning brought unprecedented empirical/engineering progress into many applications, including fundamental sciences.

- Some theory open questions:

For instance, there are many important questions regarding neural networks which are largely unanswered. There seem to be conflicting stories regarding the following issues:

- Why don't heavily parameterized neural networks overfit the data?
- What is the effective number of parameters?
- Why doesn't backpropagation head for a poor local minima?

UNDERSTANDING MACHINE LEARNING
WORST CASE ANALYSIS IS NOT ENOUGH

- Deep learning brought unprecedented empirical/engineering progress into many applications, including fundamental sciences.

- Some theory open questions:

For instance, there are many important questions regarding neural networks which are largely unanswered. There seem to be conflicting stories regarding the following issues:

- Why don't heavily parameterized neural networks overfit the data?
- What is the effective number of parameters?
- Why doesn't backpropagation head for a poor local minima?

UNDERSTANDING MACHINE LEARNING
WORST CASE ANALYSIS IS NOT ENOUGH

- Deep learning brought unprecedented empirical/engineering progress into many applications, including fundamental sciences.

- Some theory open questions:

For instance, there are many important questions regarding neural networks which are largely unanswered. There seem to be conflicting stories regarding the following issues:

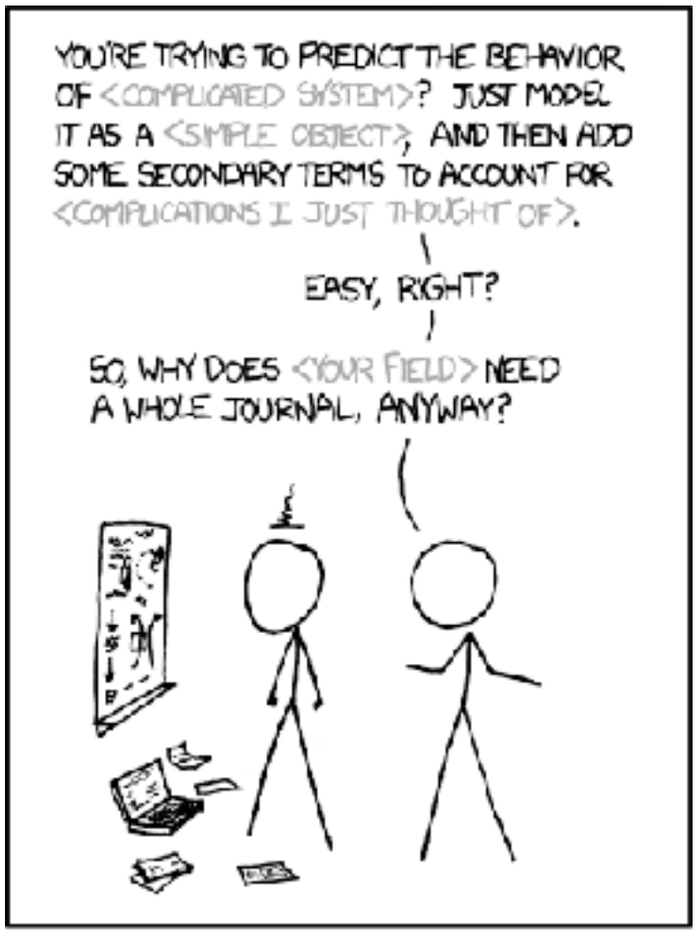
- Why don't heavily parameterized neural networks overfit the data?
- What is the effective number of parameters?
- Why doesn't backpropagation head for a poor local minima?

From "Reflections after refereeing papers for NIPS", Leo Breiman, 1995.

Still not answered!

Physicists like models of data

Instead of worst case analysis, we could instead study models of data



credit: XKCD

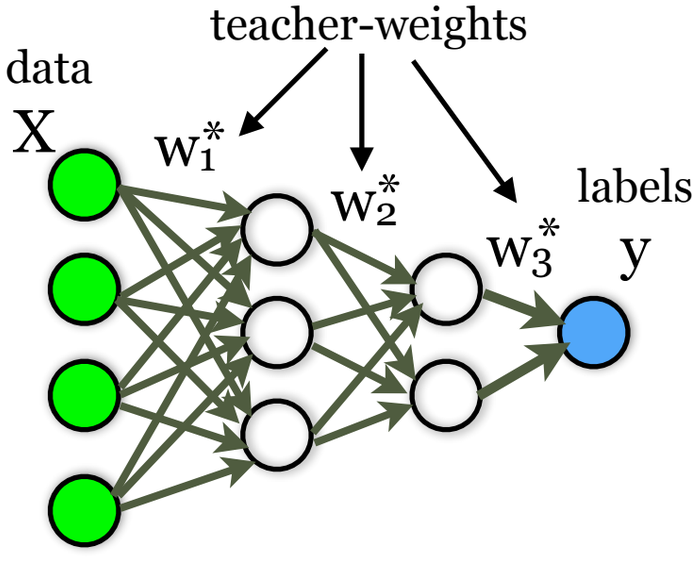
Teacher - Student Framework

[P. Carnevali & S. Patarnello (1987)
N. Tishby, E. Levin, & S. Solla (1989)
E. Gardner, B. Derrida (1989)]

Can a neural network learn a neural network?

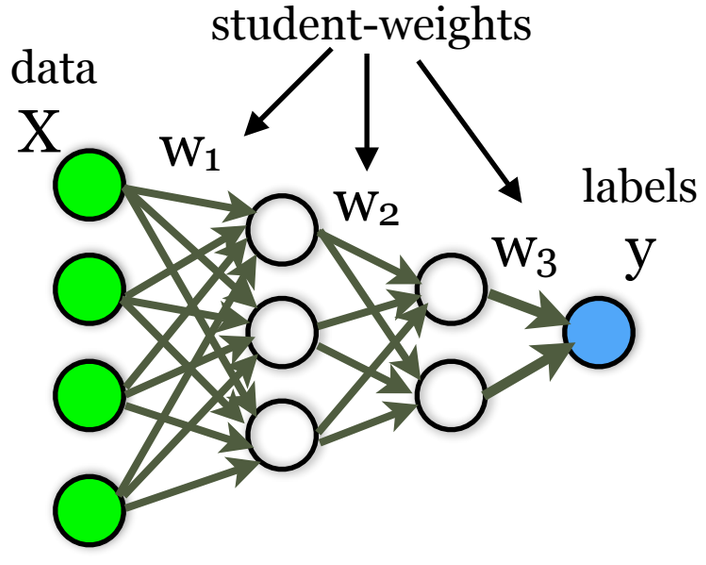
Teacher-network

- Generates data X, n samples of p dimensional data, e.g. random input vectors.
- Generates weights w*, e.g. iid random.
- Generates labels y.



Student-network

- Observes X, y
- How does the generalisation error depend on the number of samples n?



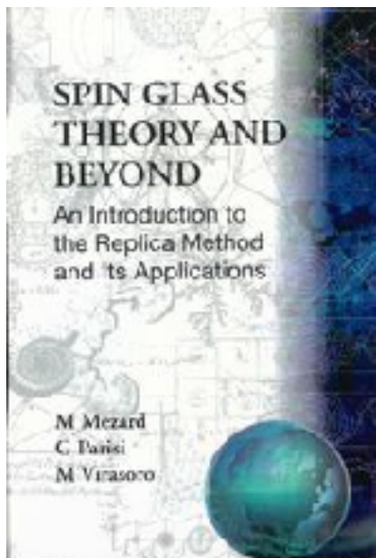
Statistical mechanics

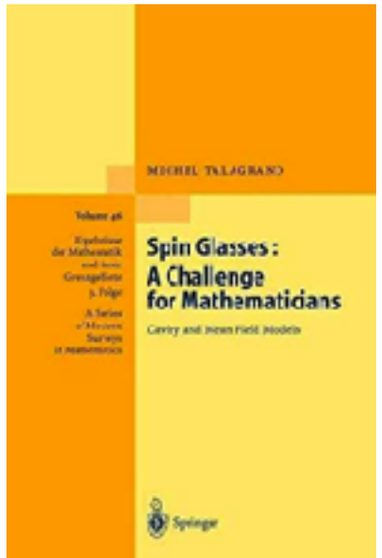
$$\hat{\theta} = \text{argmin}\mathscr{R}\left(\theta, \{\mathbf{X}, y\}\right) \qquad \mathscr{R}\left(\theta, \{\mathbf{X}, y\}\right) = \frac{1}{n}\sum_{i=1}^{n}\mathscr{L}(y_i, f_\theta(\mathbf{X}_i))$$

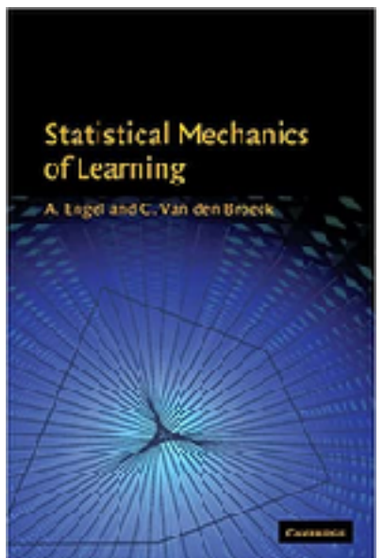
$$P_{\text{Boltzmann}}(\theta, \{\mathbf{X}, y\}) = \frac{1}{Z(\{\mathbf{X}, y\})}e^{-\beta\mathscr{R}\left(\theta, \{\mathbf{X}, y\}\right)}$$

Effective Statistical Mechanics problem, with disordered interaction depending on $\{\mathbf{X}, y\}$

Need to study the zero-temperature limit of the averaged "free energy" $-\mathbb{E}\log Z(\{\mathbf{X}, y\})$

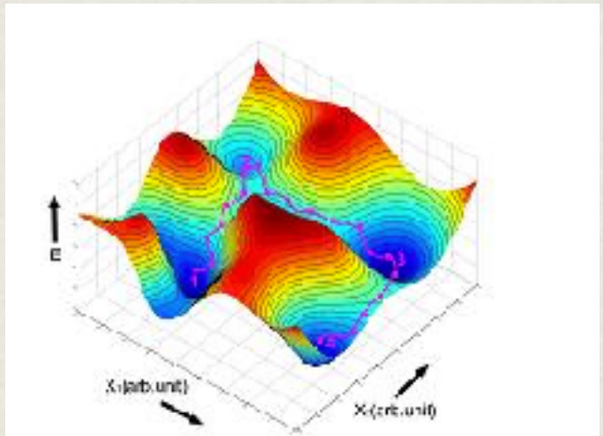




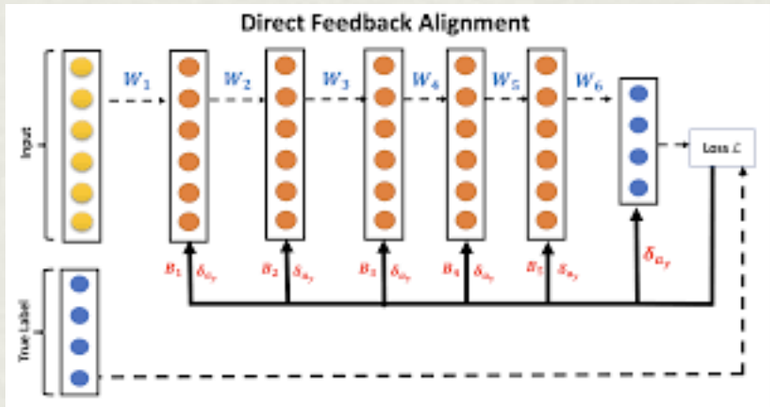


MANY DIRECTIONS EXPLORED IN MY GROUP

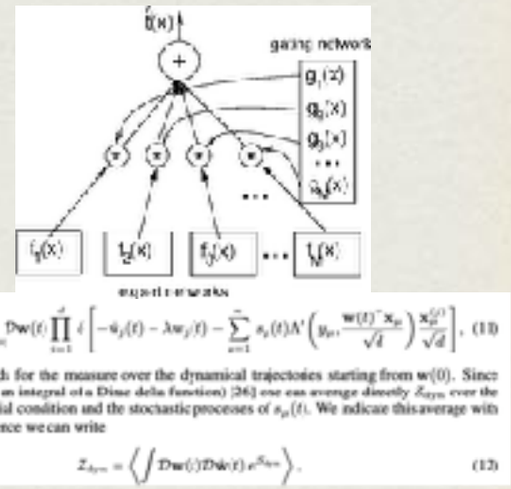
Study of energy landscape

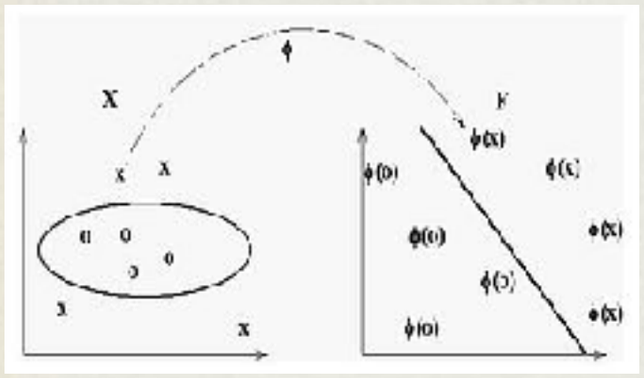


Alternative to back-propagation



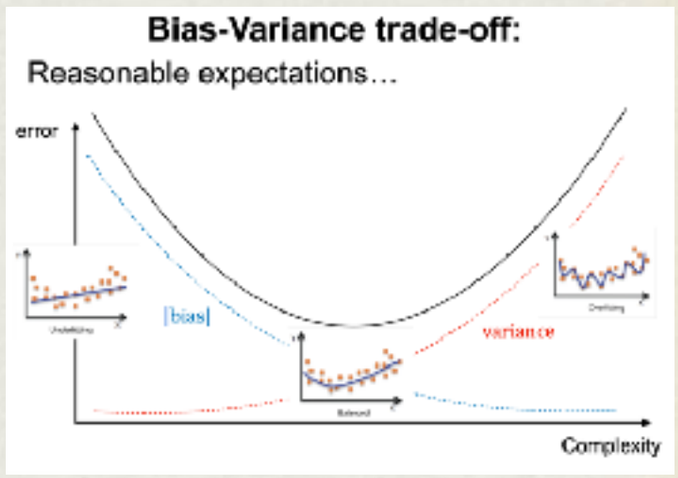
Dynamics of learning in NN





Kernel vs Neural nets

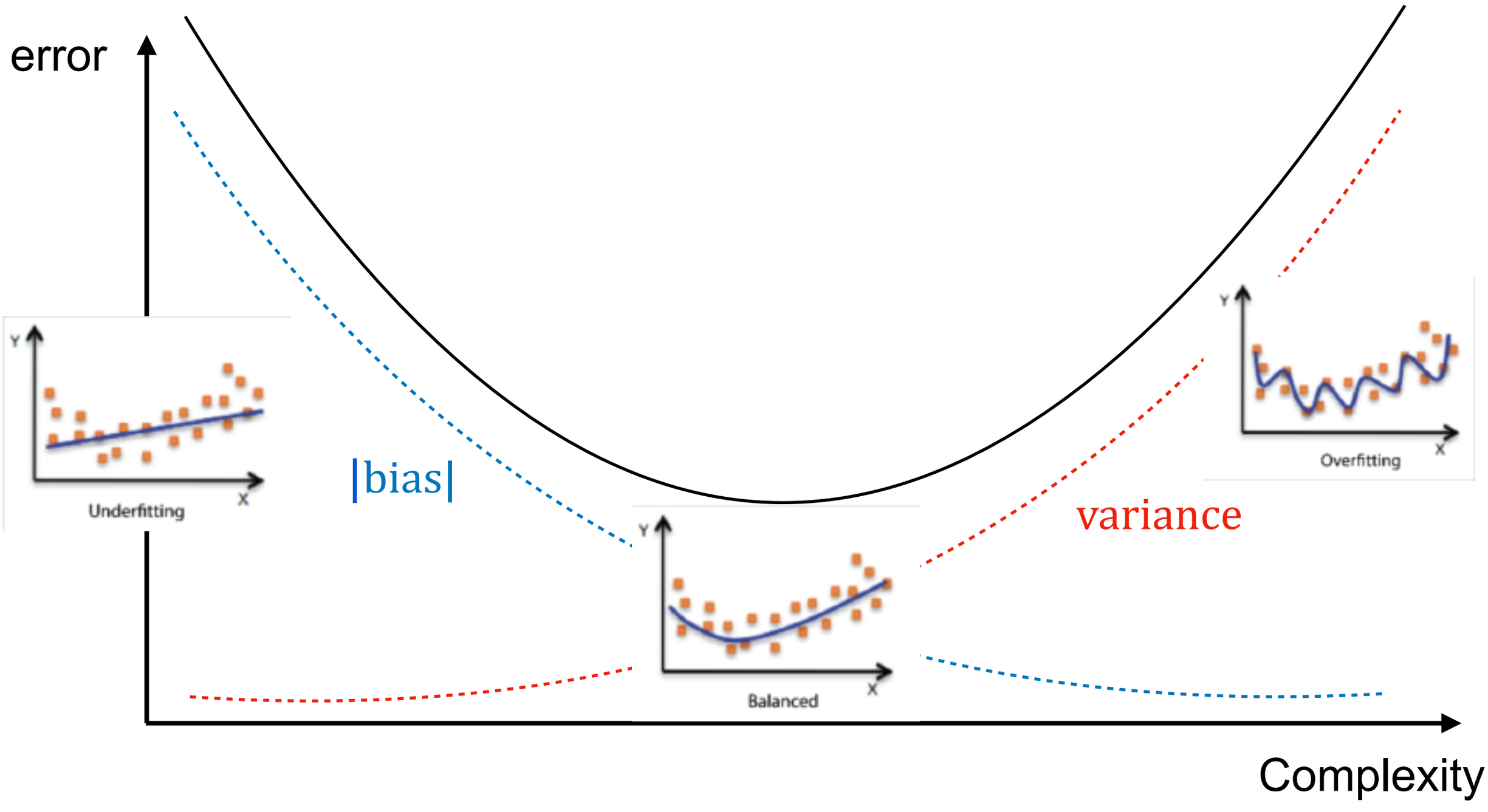
Rigorous approach to replica method



Bias-Variance trade-off

Bias-Variance trade-off:

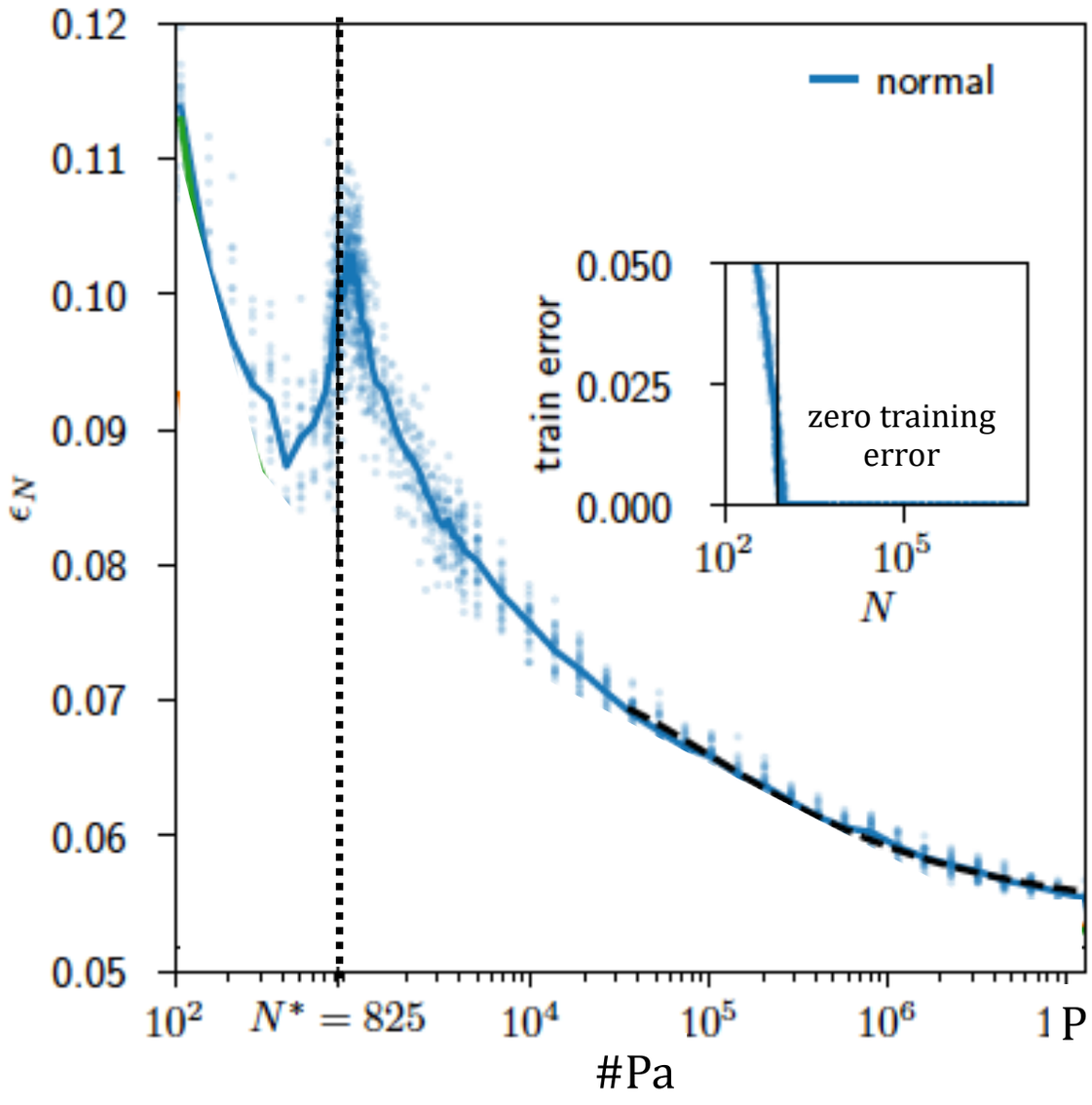
Reasonable expectations…



Bias-Variance trade-off:

… versus reality

error



Complexity

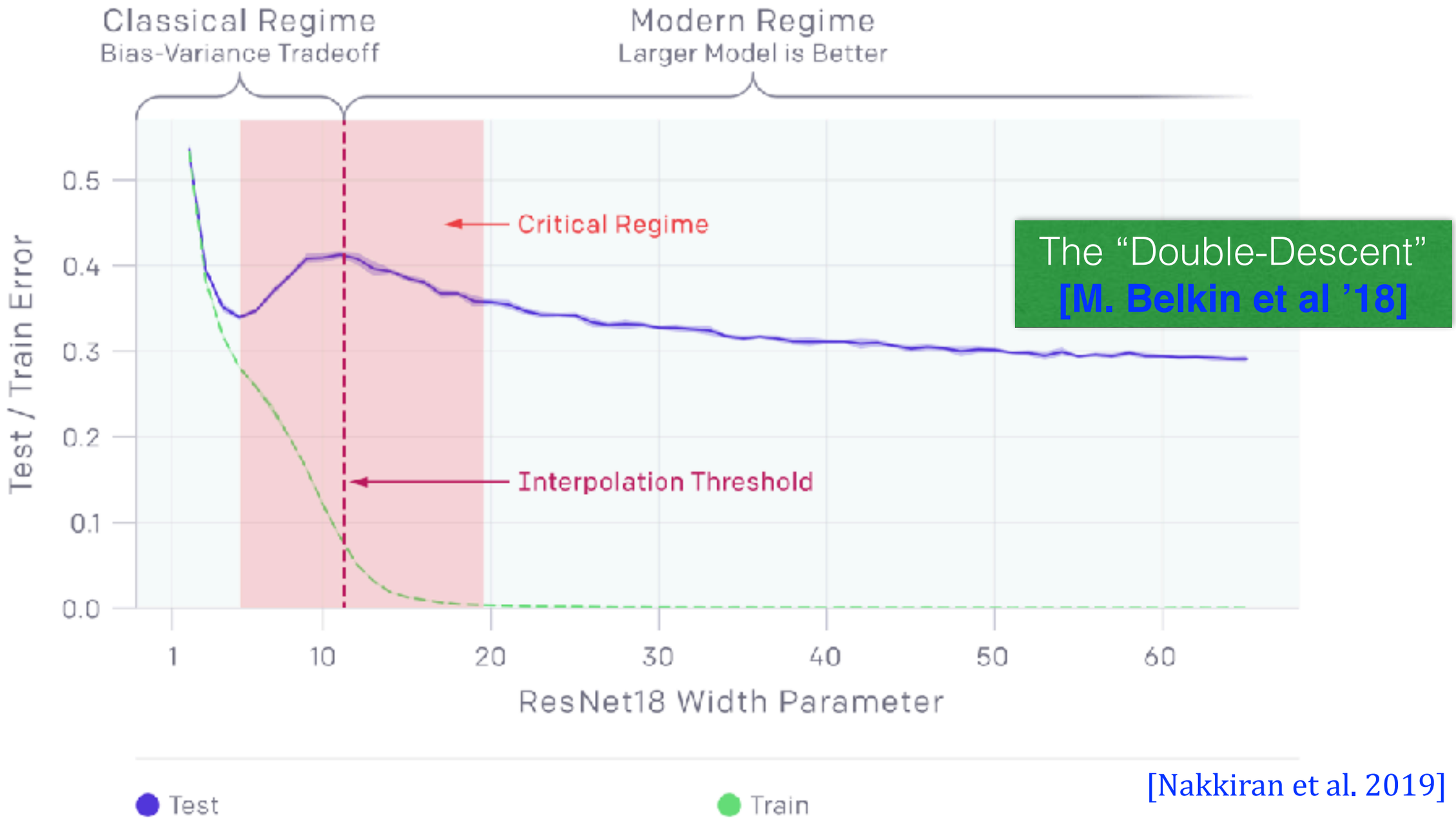
Parity-MNIST, 5 layers, FCN, hinge loss, no regularisation

[Geiger et al. ’18]

See also [Geman et al. ’92; Opper ’95; Neyshabur, Tomyoka, Srebro, 2015; Advani-Saxe 2017; Belkin, Hsu, Ma, Soumik, Mandal 2019; Nakkiran et al. 2019]

Bias-Variance trade-off:

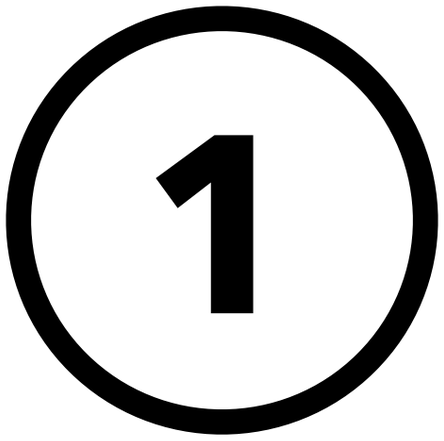
… versus reality



The "Double-Descent"
[M. Belkin et al '18]

[Nakkiran et al. 2019]

See also [Geman et al. '92; Opper '95; Neyshabur, Tomyoka, Srebro, 2015; Advani-Saxe 2017; Belkin, Hsu, Ma, Soumik, Mandal 2019; Nakkiran et al. 2019]



Learning with a simple one-layer network

Teacher-student perceptron

J. Phys. A: Math. Gen. 22 (1989) 1983-1994. Printed in the UK

1989

Three unfinished works on the optimal storage capacity of networks

E Gardner and B Derrida

The Institute for Advanced Studies, The Hebrew University of Jerusalem, Jerusalem, Israel and Service de Physique Théorique de Saclay†, F-91191 Gif-sur-Yvette Cedex, France

Received 13 December 1988

Abstract. The optimal storage properties of three different neural network models are studied. For two of these models the architecture of the network is a perceptron with $\pm J$ interactions, whereas for the third model the output can be an arbitrary function of the inputs. Analytic bounds and numerical estimates of the optimal capacities and of the minimal fraction of errors are obtained for the first two models. The third model can be solved exactly and the exact solution is compared to the bounds and to the results of numerical simulations used for the two other models.

Simplest version

⭐ ***Data*** $\begin{cases} \vec{x}^{(\mu)} \in \mathbb{R}^d, \mu = 1 \dots m \\ P_X(\vec{x}) = \mathcal{N}(0, \mathbf{1}_d) \end{cases}$

⭐ ***Labels*** $\begin{cases} y_\mu \equiv \text{sign}(\vec{x}_\mu \cdot \vec{W}^*) \\ \vec{W}^* \sim \mathcal{N}(0, \mathbf{1}_d) \end{cases}$

High-dimensional limit n, d $\to \infty$,
with $\alpha = n/d$ fixed

Physics literature

[…, Opper Kinzel '90, Kleinz, Seung '90, Opper, Haussler '91, Seung Sompolinsky, Tishby '92, Watkin Rau & Biehl '93, Opper, Kinzel '95,…]

Rigorous proofs

[…., Barbier, **FK**, Macris, Miolane, Zdeborova '17
Trampoulidis, Abbasi, Hassbi '18
Montanari, Ruan, Sohn, Yan '20
Aubin, **FK**, Lu, Zdeborova '20,
Gerbelot, Abbara, **FK** '20….]

Teacher-student perceptron

J. Phys. A: Math. Gen. 22 (1989) 1983-1994. Printed in the UK

1989

Three unfinished works on the optimal storage capacity of networks

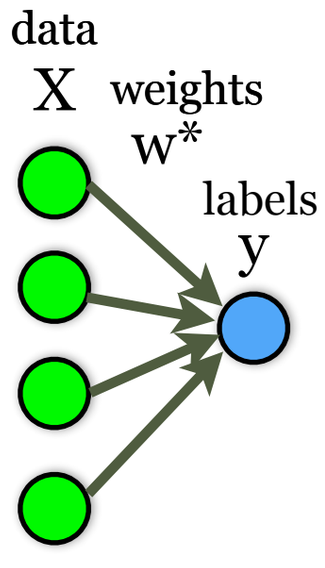
E Gardner and B Derrida

The Institute for Advanced Studies, The Hebrew University of Jerusalem, Jerusalem, Israel and Service de Physique Théorique de Saclay†, F-91191 Gif-sur-Yvette Cedex, France

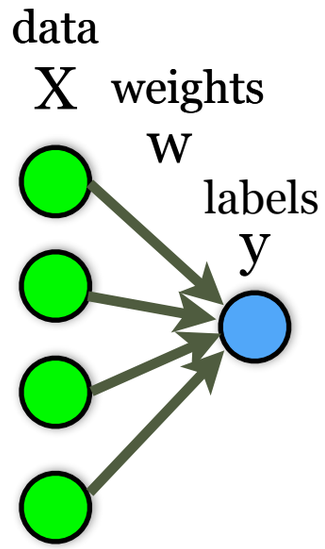
Received 13 December 1988

Abstract. The optimal storage properties of three different neural network models are studied. For two of these models the architecture of the network is a perceptron with $\pm J$ interactions, whereas for the third model the output can be an arbitrary function of the inputs. Analytic bounds and numerical estimates of the optimal capacities and of the minimal fraction of errors are obtained for the first two models. The third model can be solved exactly and the exact solution is compared to the bounds and to the results of numerical simulations used for the two other models.

Teacher



Student



Physics literature

[…, Opper Kinzel '90, Kleinz, Seung '90, Opper, Haussler '91, Seung Sompolinsky, Tishby '92, Watkin Rau & Biehl '93, Opper, Kinzel '95,…]

Rigorous proofs

[…., Barbier, **FK**, Macris, Miolane, Zdeborova '17
Trampoulidis, Abbasi, Hassbi '18
Montanari, Ruan, Sohn, Yan '20
Aubin, **FK**, Lu, Zdeborova '20,
Gerbelot, Abbara, **FK** '20….]

A rigorous solution

For binary classification ($y \in \pm 1$), the generalization error of ERM is given by

$$e_g^{\text{erm}}(\alpha) = \frac{1}{\pi} acos(\sqrt{\eta}) \quad \textit{with} \quad \eta \equiv \frac{m^2}{\rho_{w^\star} q}$$

and $\rho_{w^\star} \equiv \frac{1}{d}\mathbb{E}\left[\|\mathbf{w}^\star\|_2^2\right]$.

Theorem (Gordon's minimax - ℓ_2 - classification)

As $n, d \to \infty$ with $n/d = \alpha = \Theta(1)$, for ℓ_2 regularization $r(w) = \frac{\lambda}{2}w^2$:

$$m \underset{d\to\infty}{\longrightarrow} \sqrt{\rho_{w^\star}}\mu^*\,, \qquad q \underset{d\to\infty}{\longrightarrow} (\mu^*)^2 + (\delta^*)^2\,, \qquad \Sigma \underset{d\to\infty}{\longrightarrow} \tau^\star$$

where parameters μ^* and δ^* are solutions of

$$(\mu^*, \delta^*) = \underset{\mu,\delta\geq 0}{\arg\min} \sup_{\tau>0} \left\{ \frac{\lambda(\mu^2+\delta^2)}{2} - \frac{\delta^2}{2\tau} + \alpha \mathbb{E}_{g,s} \mathcal{M}_\tau[\delta g + \mu s y] \right\},$$

The saddle point equations yield

$$\Rightarrow \begin{cases} \mu^* & = \frac{\alpha}{\lambda\tau^* + \alpha}\mathbb{E}_{g,s}[s \cdot y \cdot \mathcal{P}_{\tau^*}(\delta^* g + \mu^* s y)]\,, \\ \delta^* & = \frac{\alpha}{\lambda\tau^* + \alpha - 1}\mathbb{E}_{g,s}[g \cdot \mathcal{P}_{\tau^*}(\delta^* g + \mu^* s y)]\,, \\ (\delta^*)^2 & = \alpha \mathbb{E}_{g,s}[((\delta^* g + \mu^* s y) - \mathcal{P}_{\tau^*}(\delta^* g + \mu^* s y))^2] \end{cases}$$

with $y = \varphi_{\text{out}^\star}(\sqrt{\rho_{w^\star}} s)$, $\mathcal{M}$ and $\mathcal{P}$ the Moreau-Yosida regularization and the proximal map.

[Aubin, **FK**, Lu, Zdeborova '20]

L2 loss

Simplest version

⭐ ***Data*** $\begin{cases} \overrightarrow{x}^{(\mu)} \in \mathbb{R}^d, \mu = 1 \ldots m \\ P_X(\overrightarrow{x}) = \mathcal{N}(0, \mathbf{1}_d) \end{cases}$

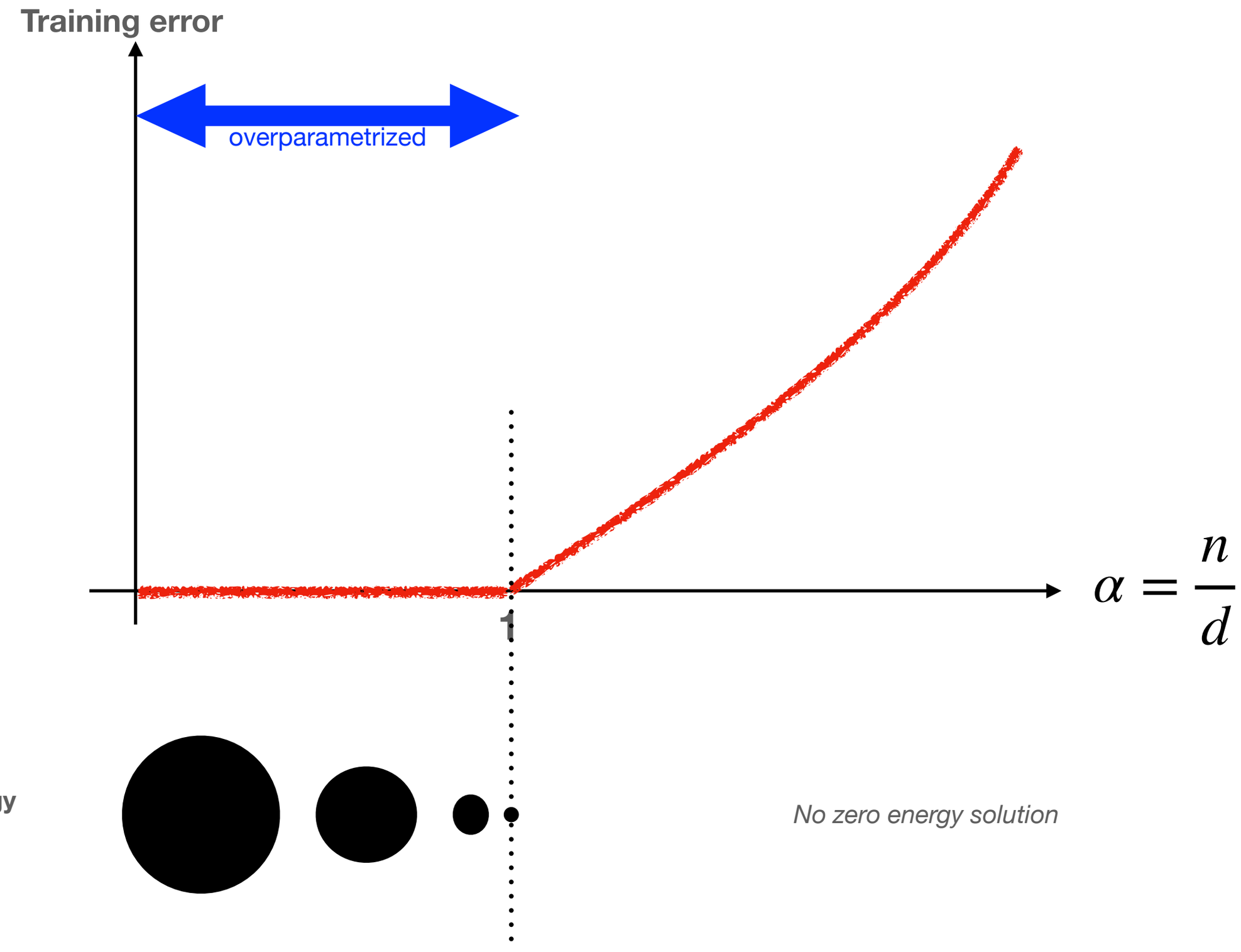
⭐ ***Labels*** $\begin{cases} y_\mu \equiv \text{sign}(\overrightarrow{\text{x}}_\mu \cdot \overrightarrow{\text{W}}^*) \\ \overrightarrow{W}^* \sim \mathcal{N}(0, \mathbf{1}_d) \end{cases}$

$$\hat{\theta} = \text{argmin} \mathscr{R}(\theta, \{\mathbf{X}, y\})$$

High-dimensional limit n, d $\rightarrow \infty$,
with $\alpha = n/d$ fixed

$$\mathscr{R}(\theta, \{\mathbf{X}, y\}) = \frac{1}{n} \sum_{i=1}^{n} \| y_i - \theta X_i \|_2^2$$

L2 loss: what to expect?



Ordinary least square

$$\hat{\theta} = \text{argmin}(||\mathbf{Y} - \mathrm{A}\theta||_2^2)$$

$$||\mathbf{Y} - A\theta||_2^2 = (\mathbf{Y} - A\theta)^T(\mathbf{Y} - A\theta) = \mathbf{Y}^T\mathbf{Y} + \theta^T A^T A\theta - 2\mathbf{Y}^T A\theta$$

Taking the extremum yields the normal equations:

d x d d x n

$$A^T A\theta = A^T\mathbf{Y}$$

d x 1 n x 1

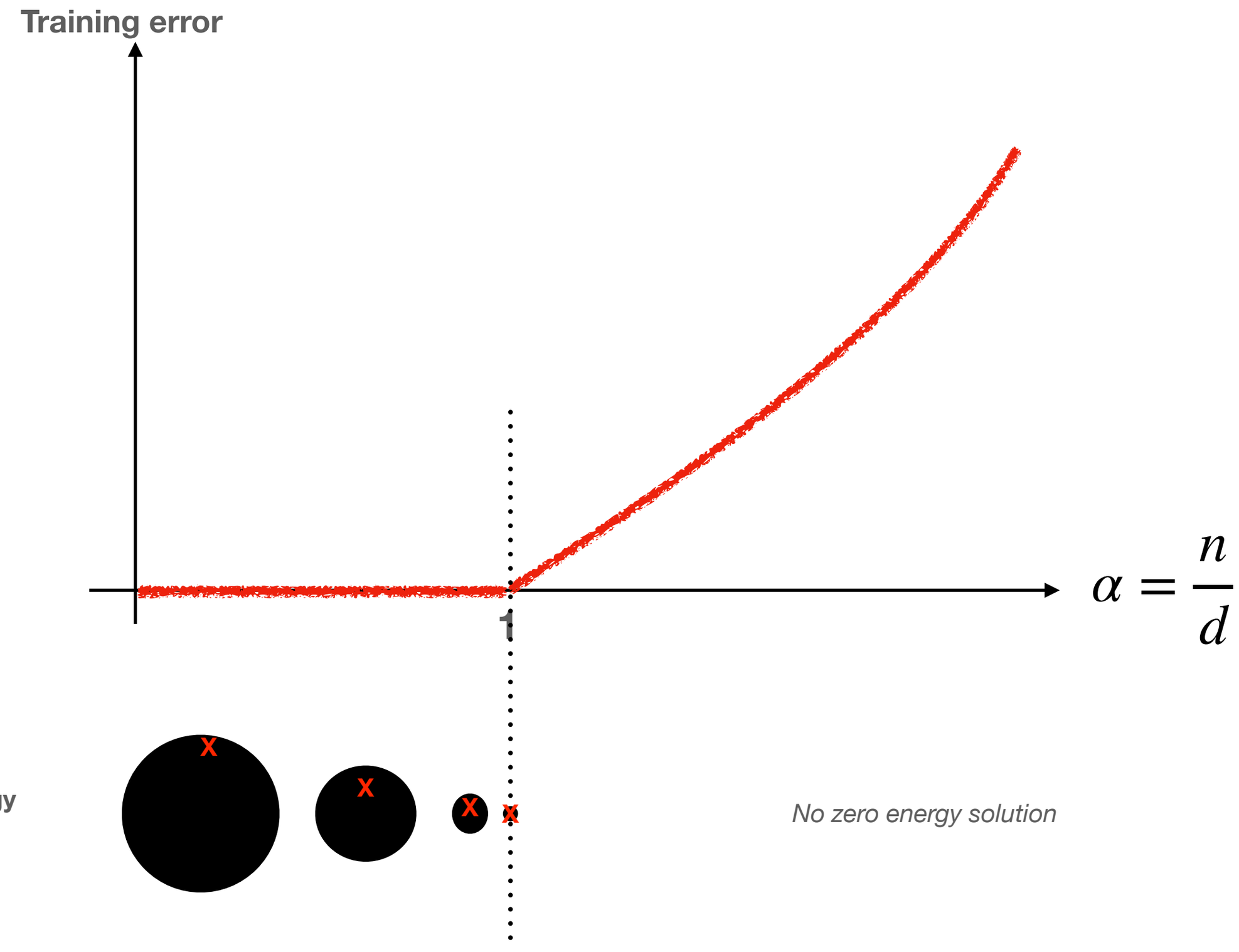
Unique solution if A^TA is full rank
This requires (*at least*) n>p
(no more unknown than datapoints)

$$\hat{\theta} = (A^T A)^{-1} A^T\mathbf{Y}$$

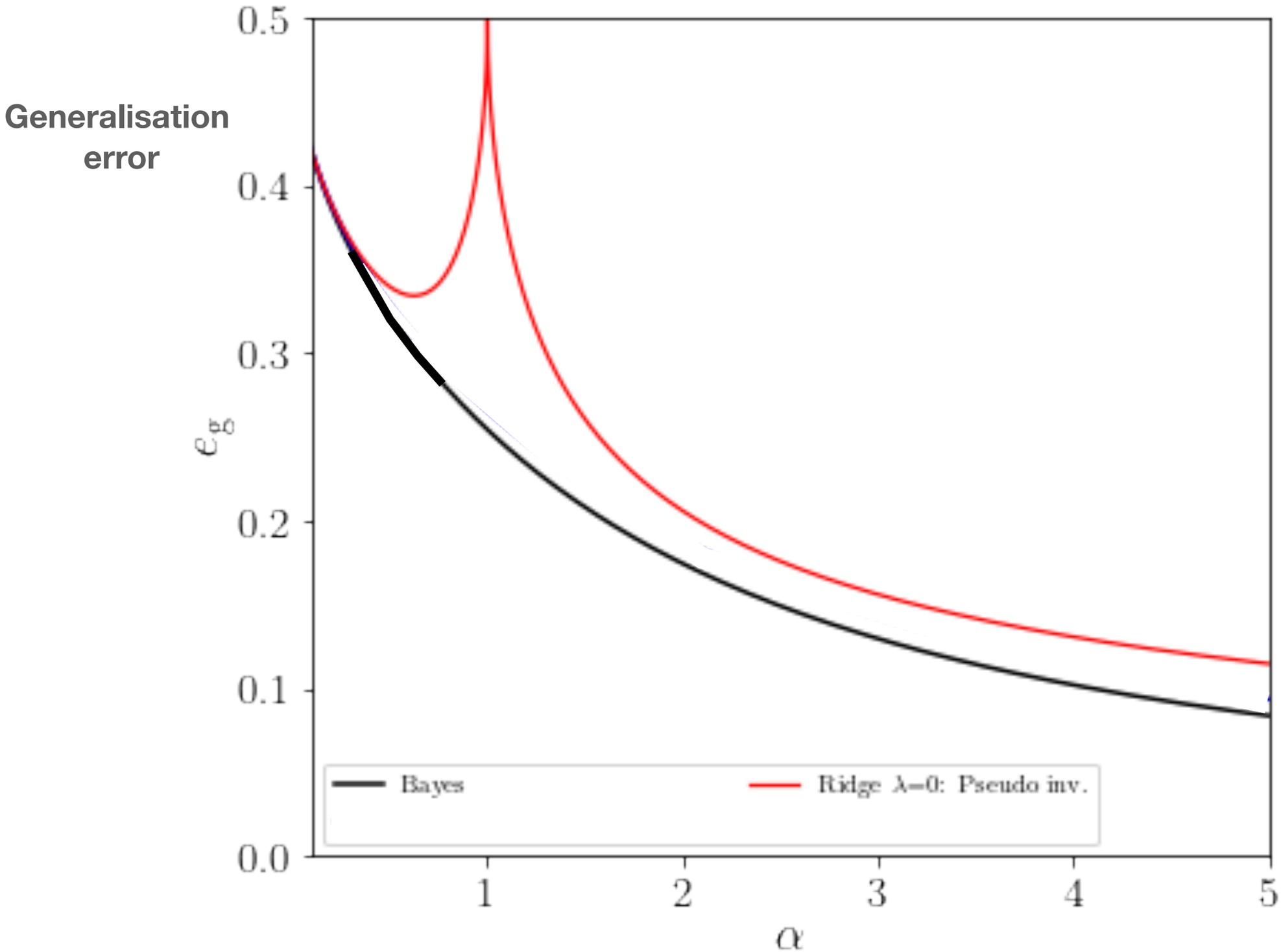
Otherwise, many solutions may exists.
A popular choice leads
to the least-norm (*in l2 norm*) solution:

$$\hat{\theta}_{\text{ln}} = A^T(AA^T)^{-1}\mathbf{Y}$$

Least norm solution

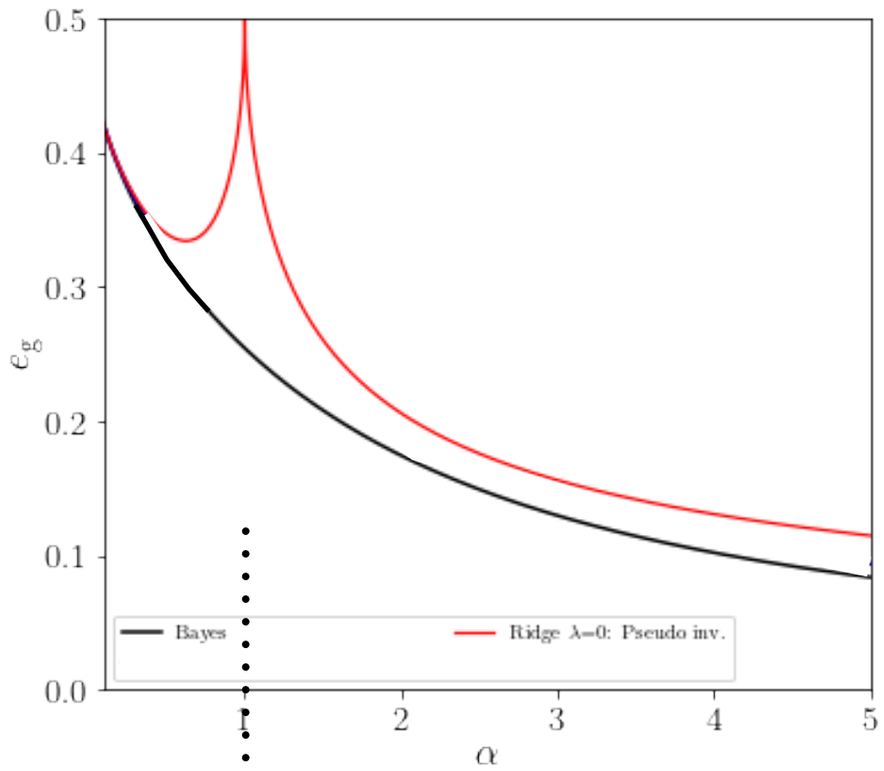


Least norm solution yields the “double descent”



Rigorous result from [Aubin, FK, Lu, Zdeborova ’20], but first discussed by Manfred Opper in ’95!

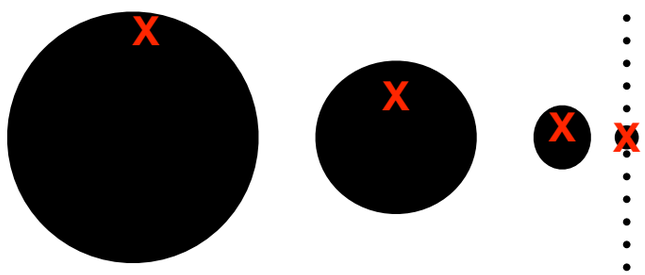
Least norm solution yields the “double descent”



Biasing to low l2 norm solutions is good!

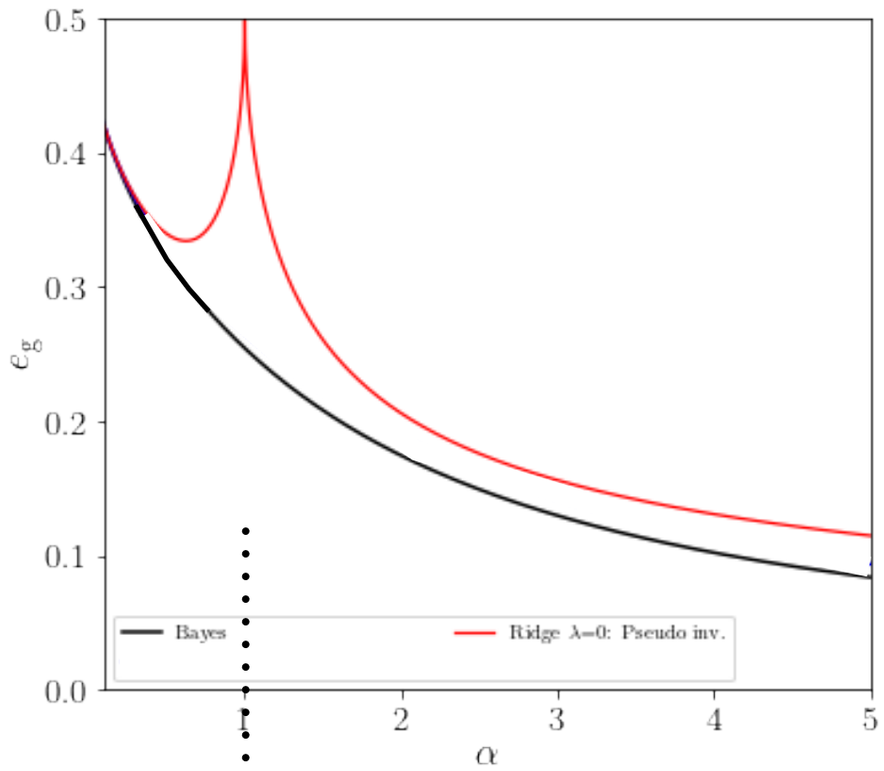
This explains the non-monotonic curve

Zero energy Solutions



No zero energy solution

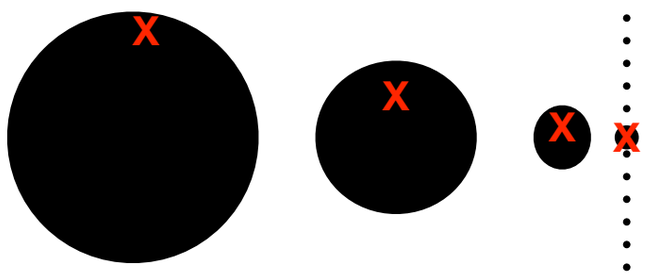
What if I do gradient descent?



Biasing to low l2 norm solutions is good!

This explains the non-monotonic curve

Zero energy Solutions



No zero energy solution

A lesson from representer theorem

$$\mathscr{R} = \frac{1}{n}\sum_{i=1}^{n} \mathscr{L}(y_i, \theta \cdot \mathbf{x}_i)$$

$$\mathbb{R}^d = \text{span}(\{X\}) + \text{null}(\{X\})$$

$$\theta = \sum_{i=1}^{n} \beta_i \mathbf{X}_i + \overrightarrow{\mathscr{N}}$$

Simple mathematical fact:

If you do gradient descent, $\overrightarrow{\mathscr{N}}$ is ever never updated!

Initialising weights close to zero implies $\overrightarrow{\mathscr{N}} = 0$ at all times:

In this case gradient descent converges to the least norm solution

This is an example of “implicit regularisation”

See also Advani-Saxe ’17, N. Sbrero *et al ’18*

Ridge loss (now with explicit regularisation)

$$\hat{\theta} = \text{argmin}\mathscr{R}(\theta, \{\mathbf{X}, y\})$$

$$\mathscr{R} = \frac{1}{n}\sum_{i=1}^{n} \|y_i - \theta X_i\|_2^2 + \lambda\|\theta\|_2^2$$

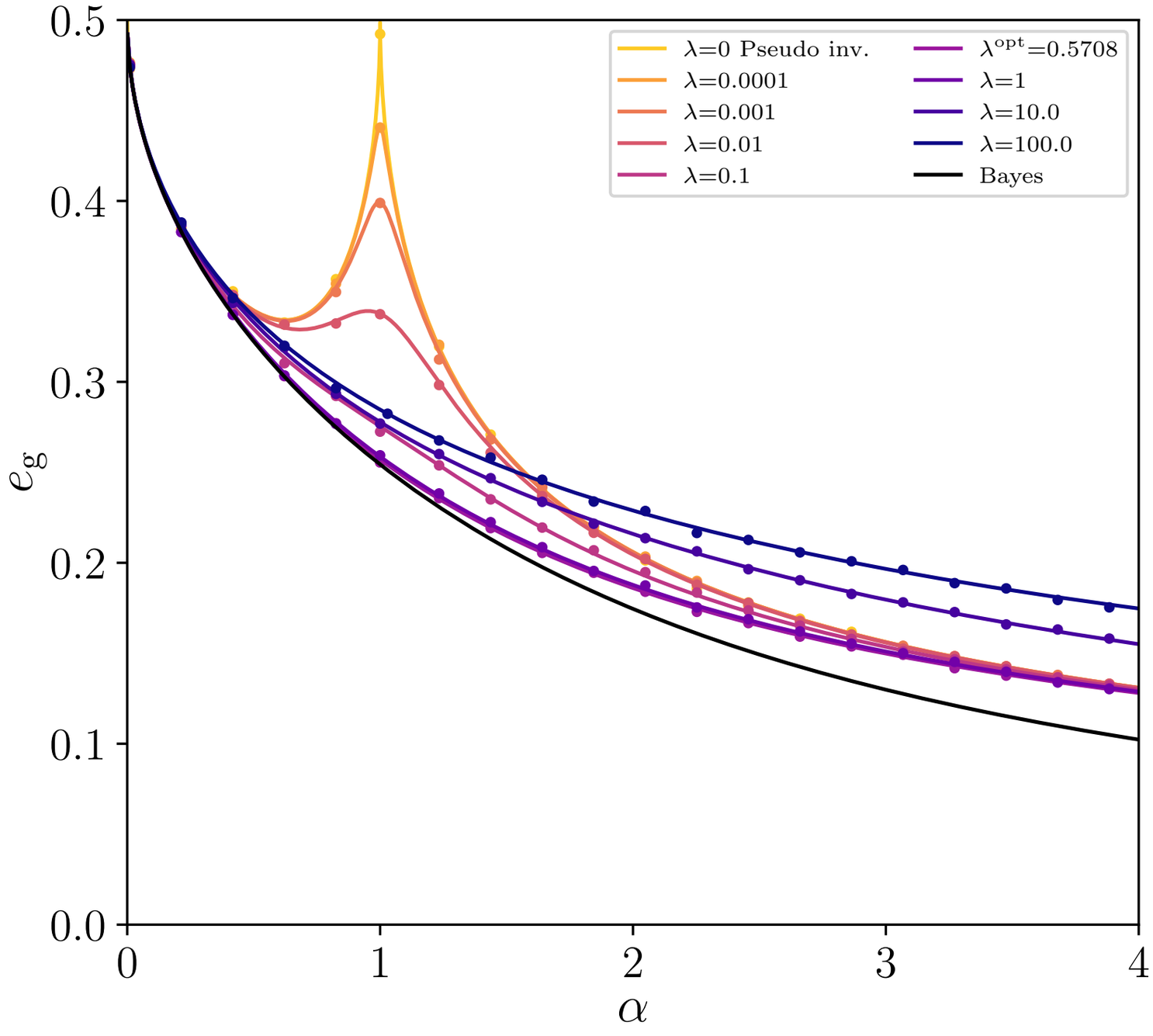
Simplest version

⭐ ***Data*** $\begin{cases} \vec{x}^{(\mu)} \in \mathbb{R}^d, \mu = 1 \ldots m \\ P_X(\vec{x}) = \mathcal{N}(0, \mathbf{1}_d) \end{cases}$

⭐ ***Labels*** $\begin{cases} y_\mu \equiv \text{sign}(\vec{\text{x}}_\mu \cdot \vec{\text{W}}^*) \\ \vec{W}^* \sim \mathcal{N}(0, \mathbf{1}_d) \end{cases}$

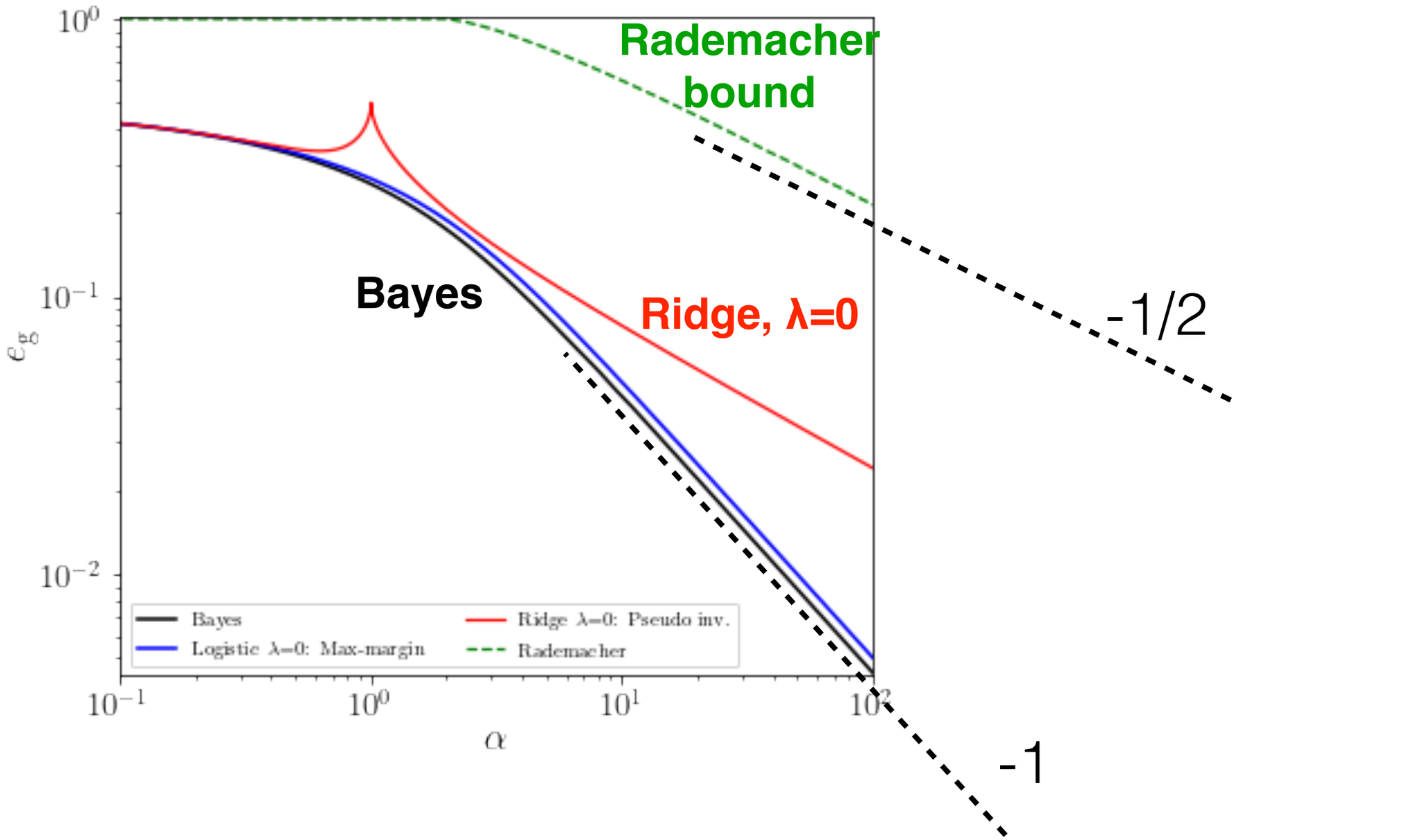
High-dimensional limit n, d $\to \infty$,
with $\alpha = n/d$ fixed

Ridge loss (now with explicit regularisation)



[Aubin, Krzakala, Lu, Zdeborova'20]

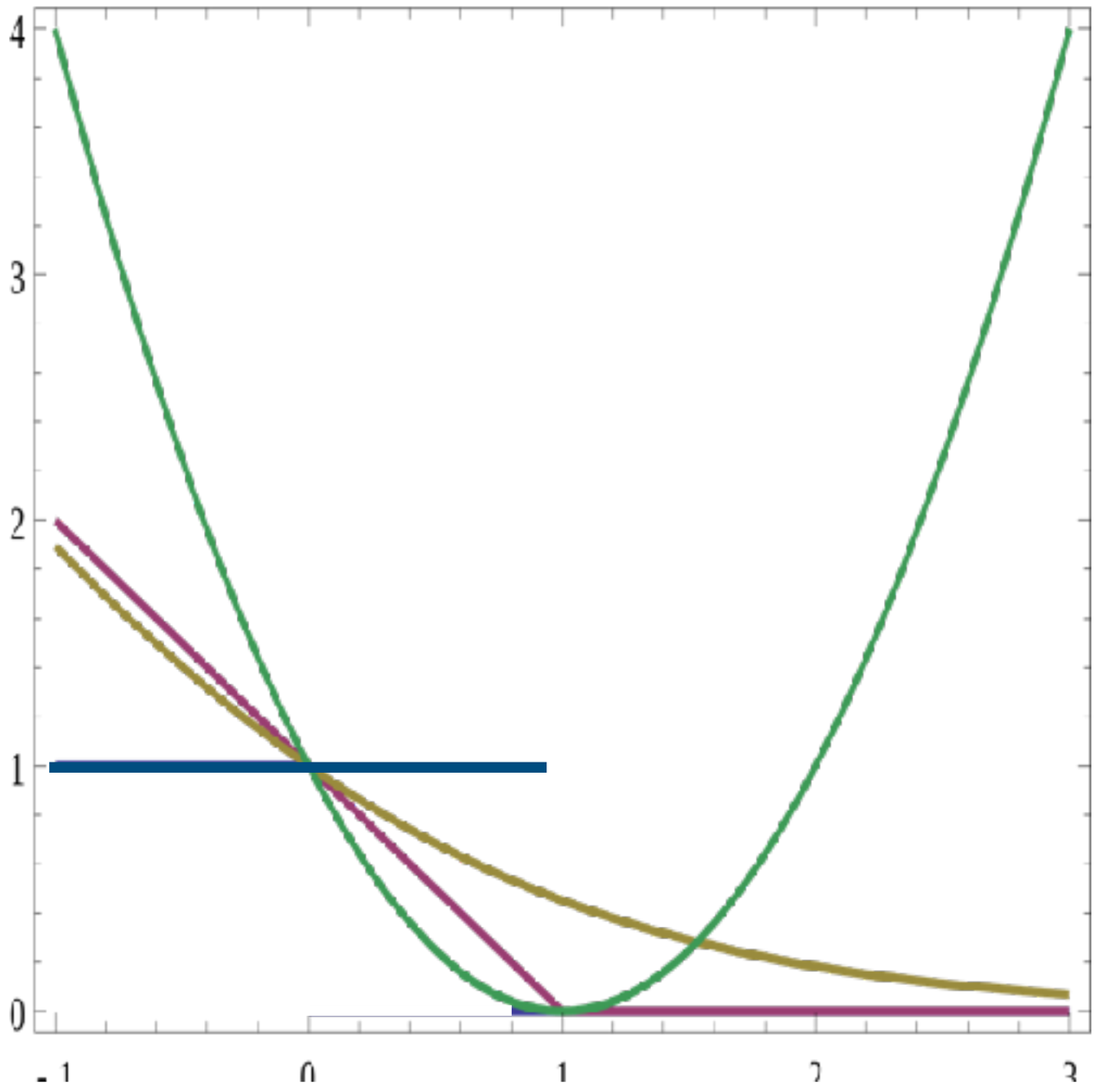
Still far from Bayes!



[Aubin, Krzakala, Lu, Zdeborova'20]

Cover you Losses!

$$f(\vec{x}) = \theta \cdot \vec{x} + \alpha$$



Square Loss

$$L(f(\vec{x}), y) = (1 - yf(\vec{x}))^2$$

Hard margin

$$L(f(\vec{x}), y) = \mathbf{1}(yf(\vec{x}) > 1)$$

Hinge loss

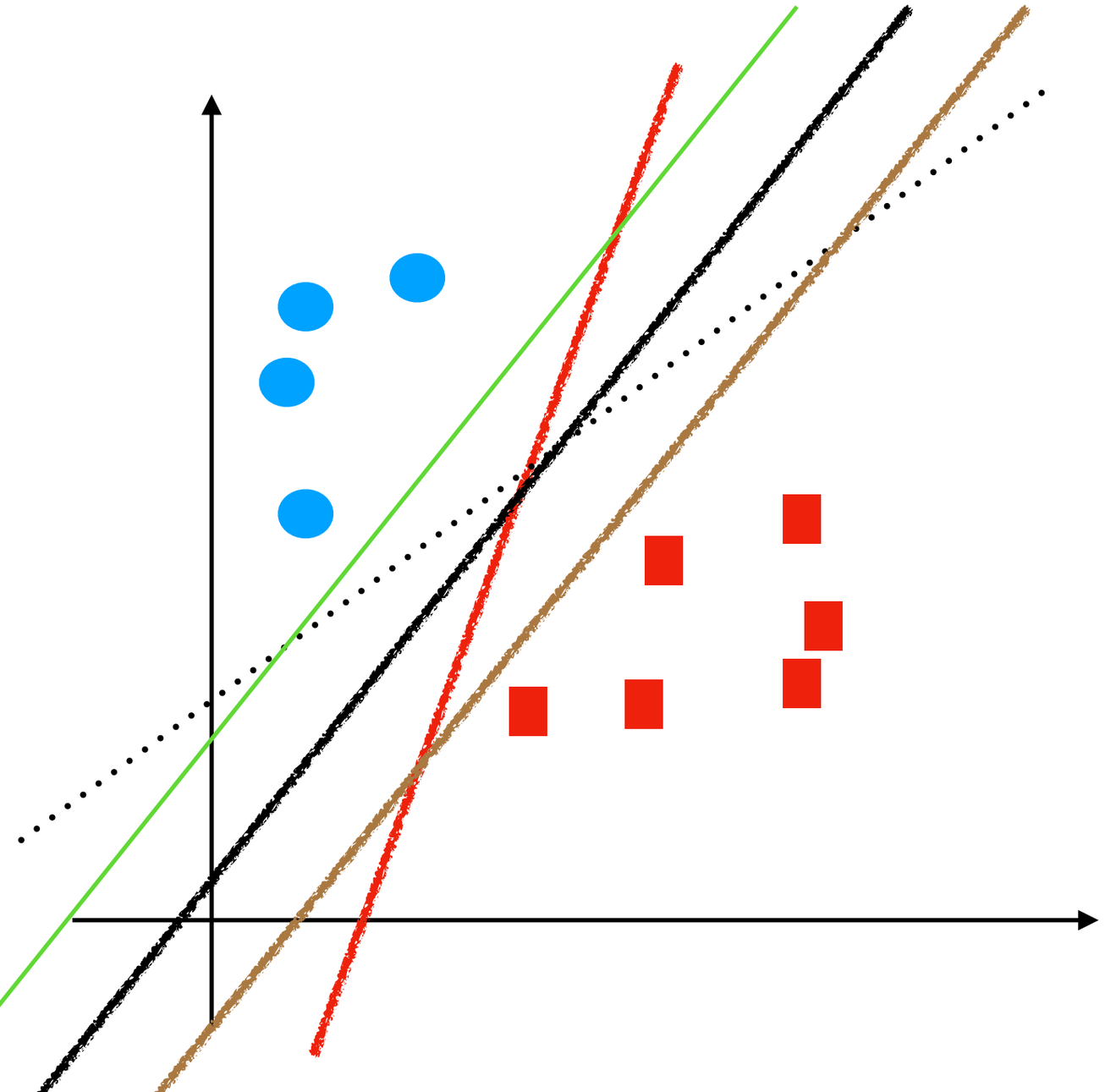
$$L(f(\vec{x}), y) = \max(0, 1 - yf(\vec{x}))$$

Logistic loss/Cross-entropy

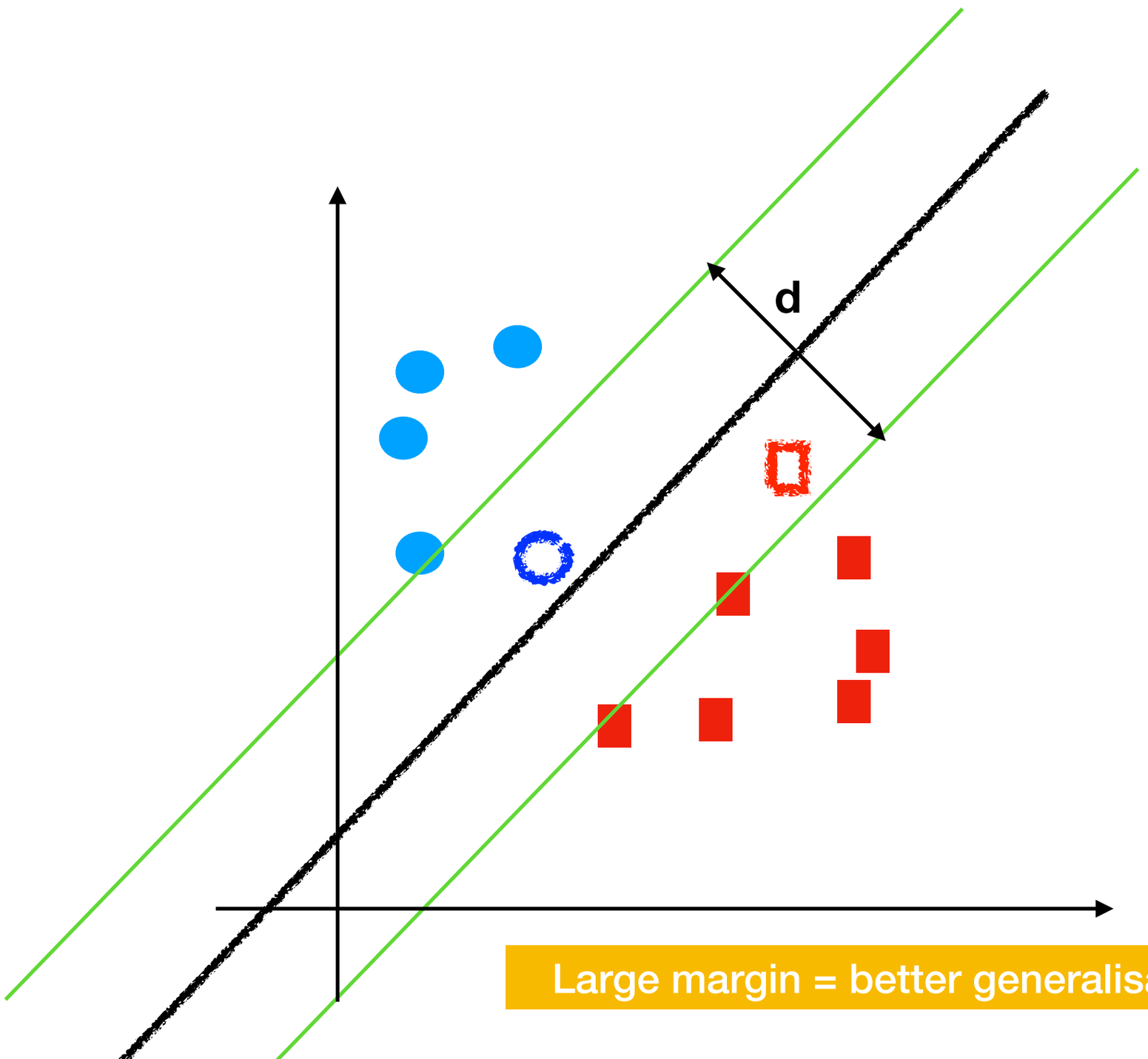
$$L(f(\vec{x}), y) = \frac{1}{\ln 2} \ln(1 + e^{-yf(\vec{x})})$$

Pushing the boundaries

Which frontier should we choose?



Pushing the boundaries



Large margin = better generalisation properties!

Implicit regularization (again)

Margin Maximizing Loss Functions

2006

Saharon Rosset
Watson Research Center
IBM
Yorktown, NY, 10598
srosset@us.ibm.com

Ji Zhu
Department of Statistics
University of Michigan
Ann Arbor, MI, 48109
jizhu@umich.edu

Trevor Hastie
Department of Statistics
Stanford University
Stanford, CA, 94305
hastie@stat.stanford.edu

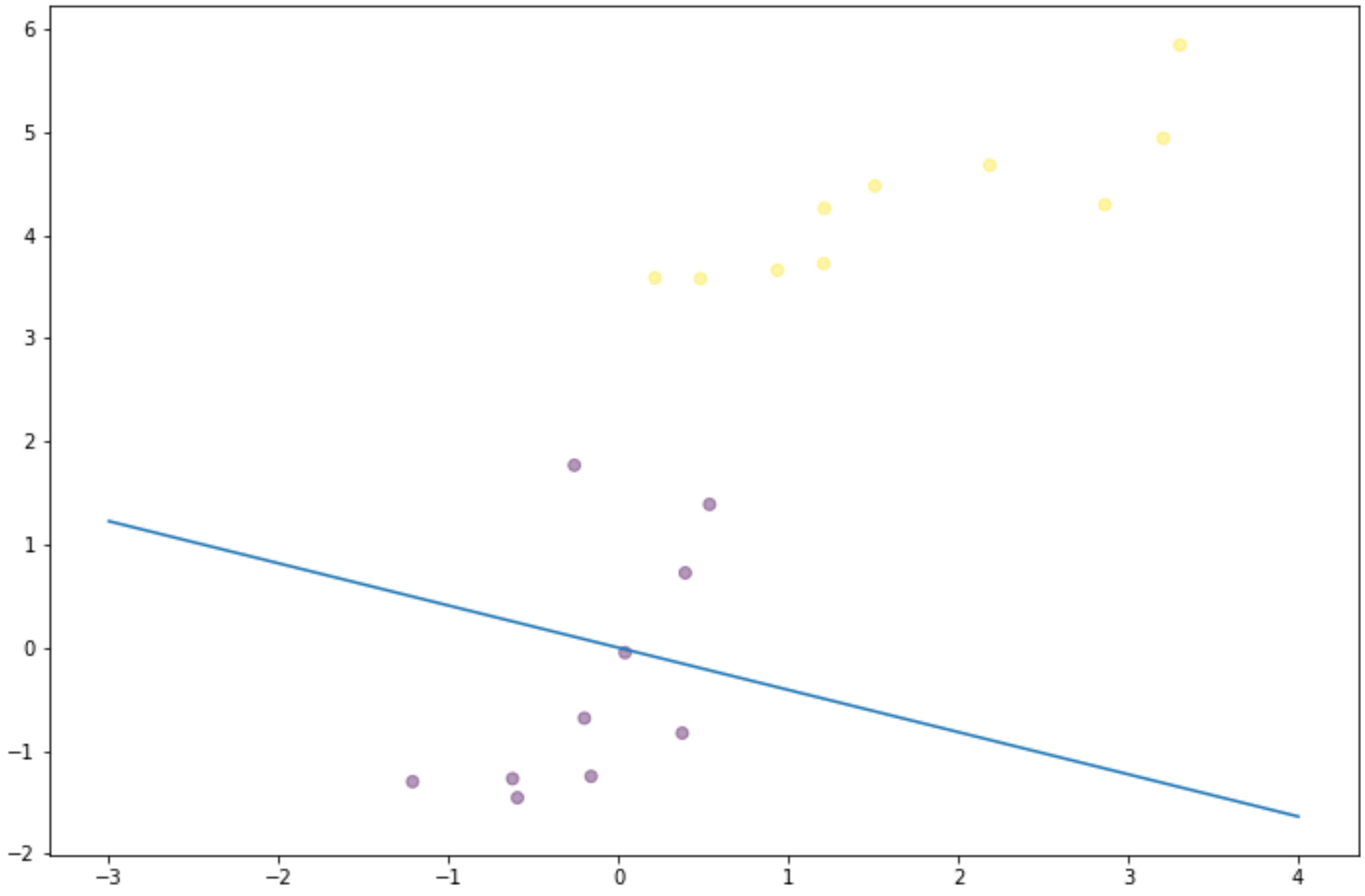
Abstract

Margin maximizing properties play an important role in the analysis of classi£-cation models, such as boosting and support vector machines. Margin maximization is theoretically interesting because it facilitates generalization error analysis, and practically interesting because it presents a clear geometric interpretation of the models being built. We formulate and prove a suf£cient condition for the solutions of regularized loss functions to converge to margin maximizing separators, as the regularization vanishes. This condition covers the hinge loss of SVM, the exponential loss of AdaBoost and logistic regression loss. We also generalize it to multi-class classi£cation problems, and present margin maximizing multi-class versions of logistic regression and support vector machines.

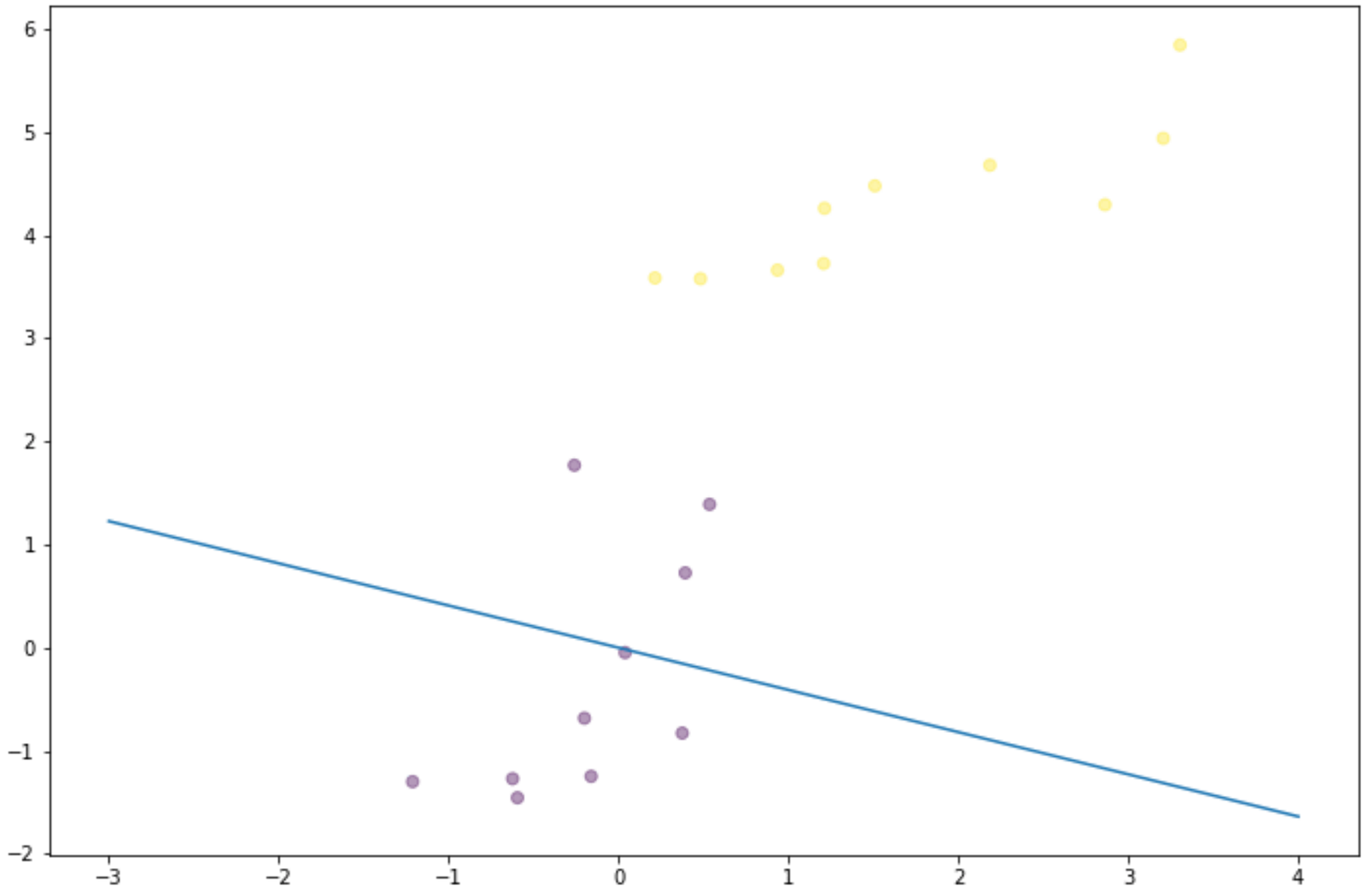
As λ goes to zero, many losses
Converges to the max-margin solution!

$$\mathscr{R} = \frac{1}{n}\sum_{i=1}^{n}\mathscr{L}(y_i, f_\theta(\mathbf{X}_i)) + \lambda\|\theta\|_2^2$$

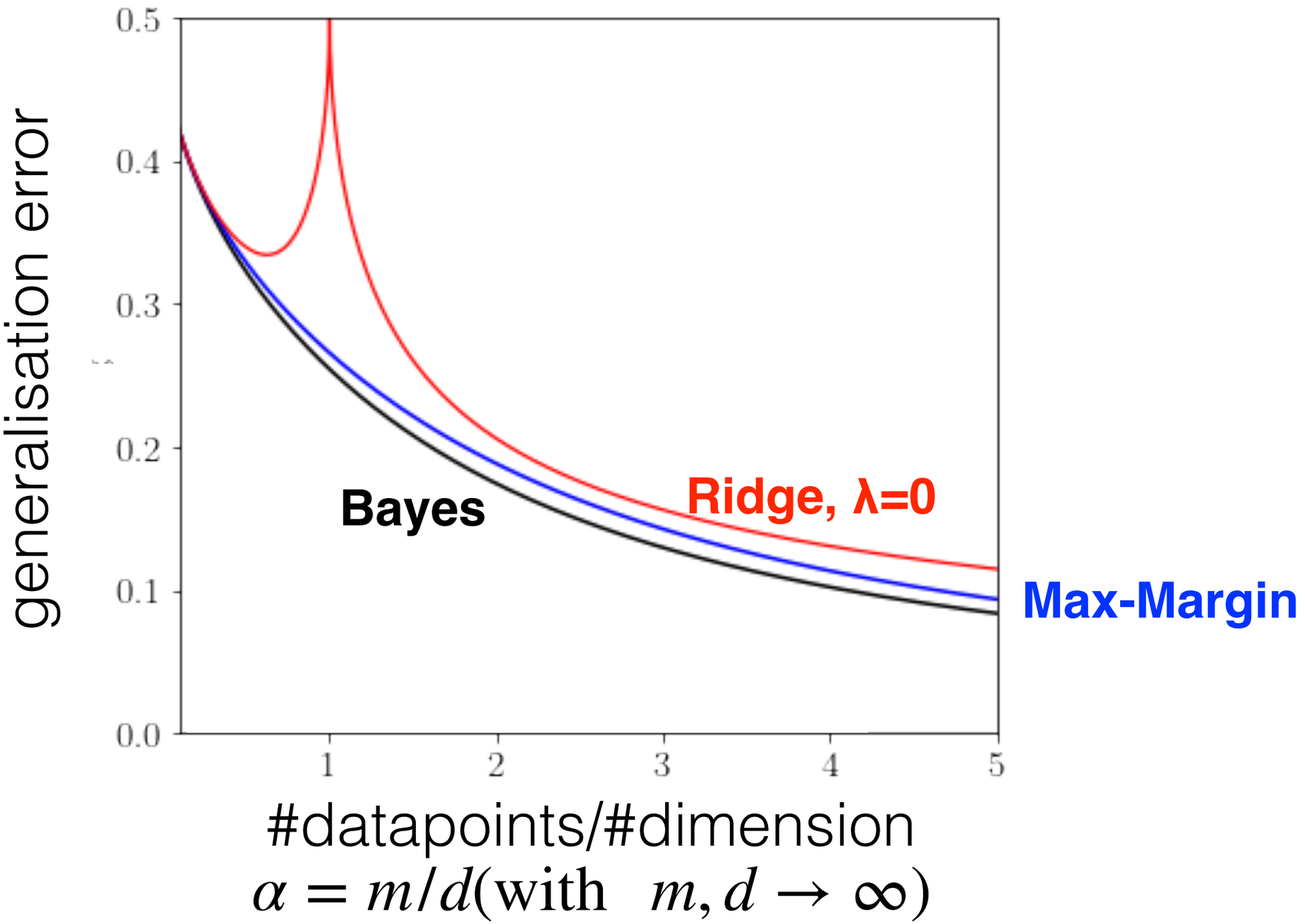
Ex: l2 LOSS



Ex: logistic LOSS

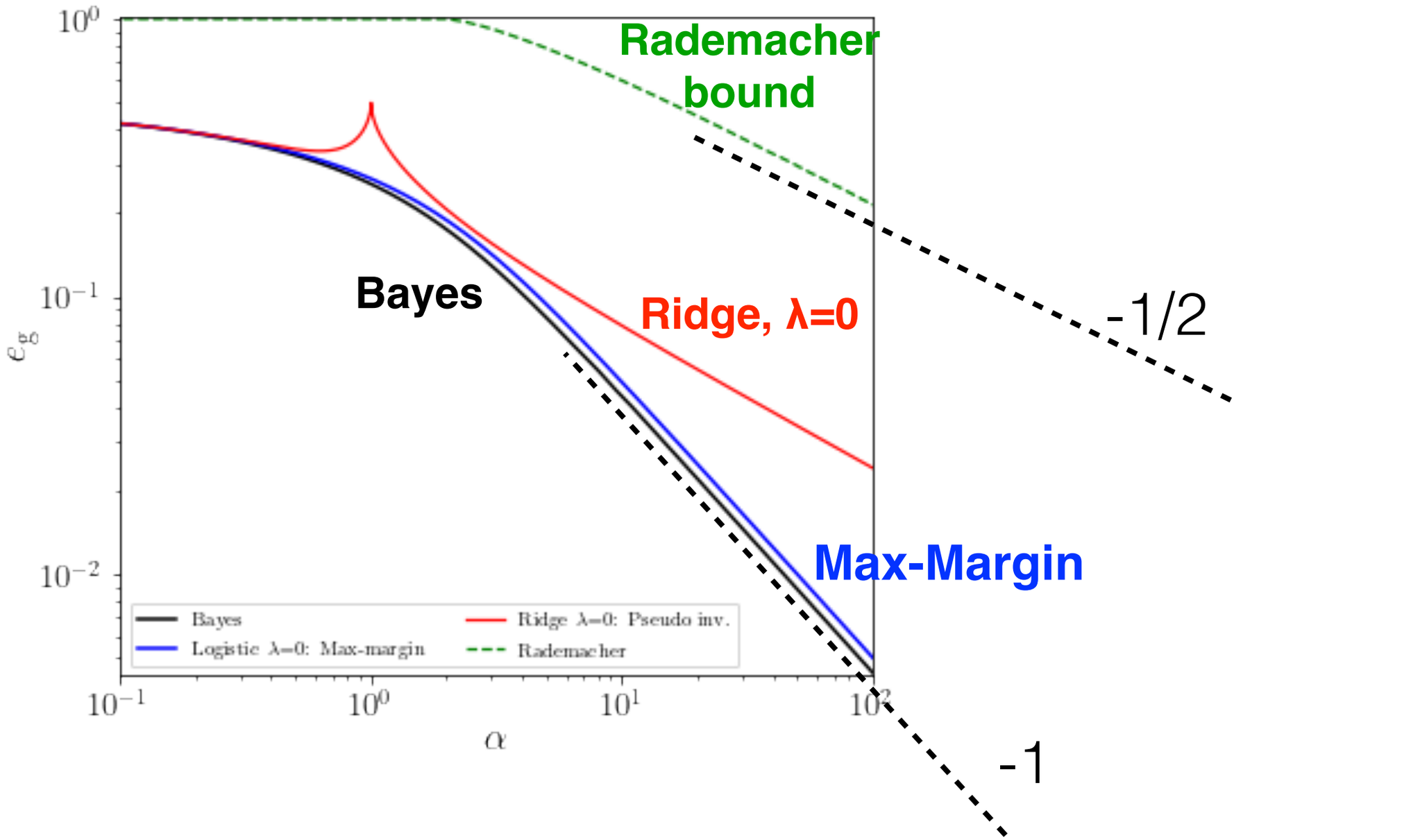


Max-margin is good



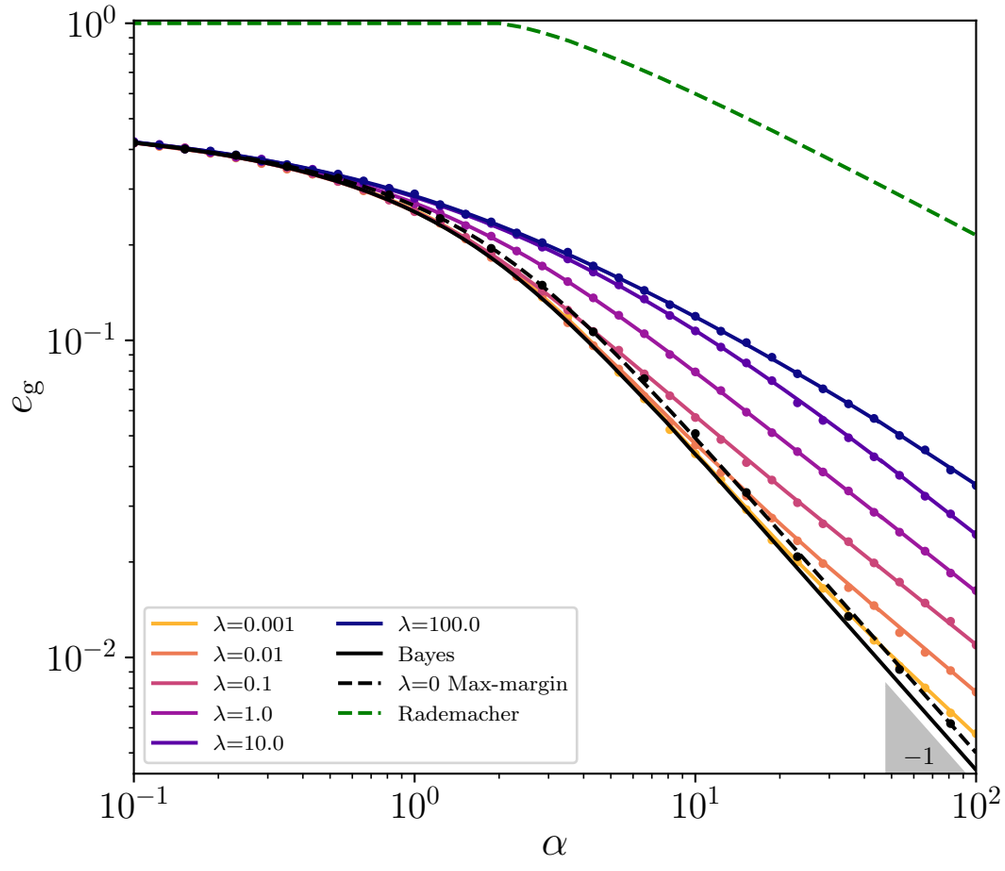
[Aubin, Krzakala, Lu, Zdeborova'20]

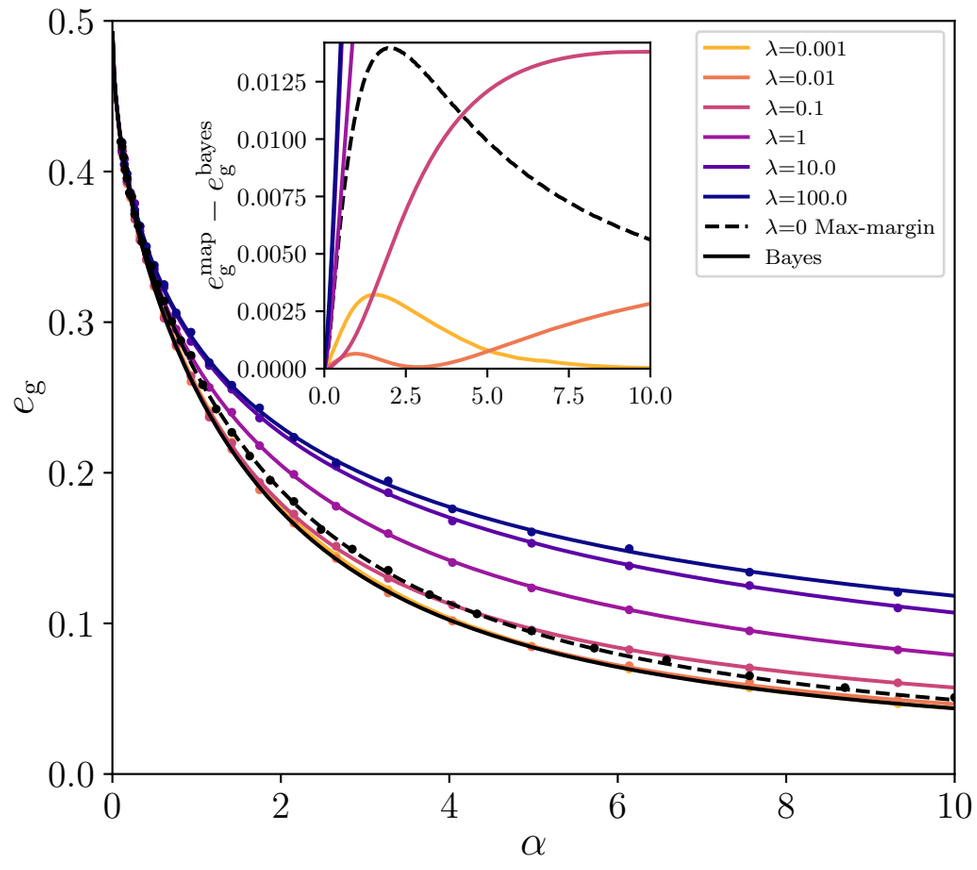
Reaching Bayes rates



[Aubin, Krzakala, Lu, Zdeborova'20]

Chasing the Bayes optimal result





Regularised logistic losses (almost) achieve Bayes optimal results!
(And specially designed losses do achieve it)

[Aubin, Krzakala, Lu, Zdeborova'20]

Ok, does this explain double-descent?

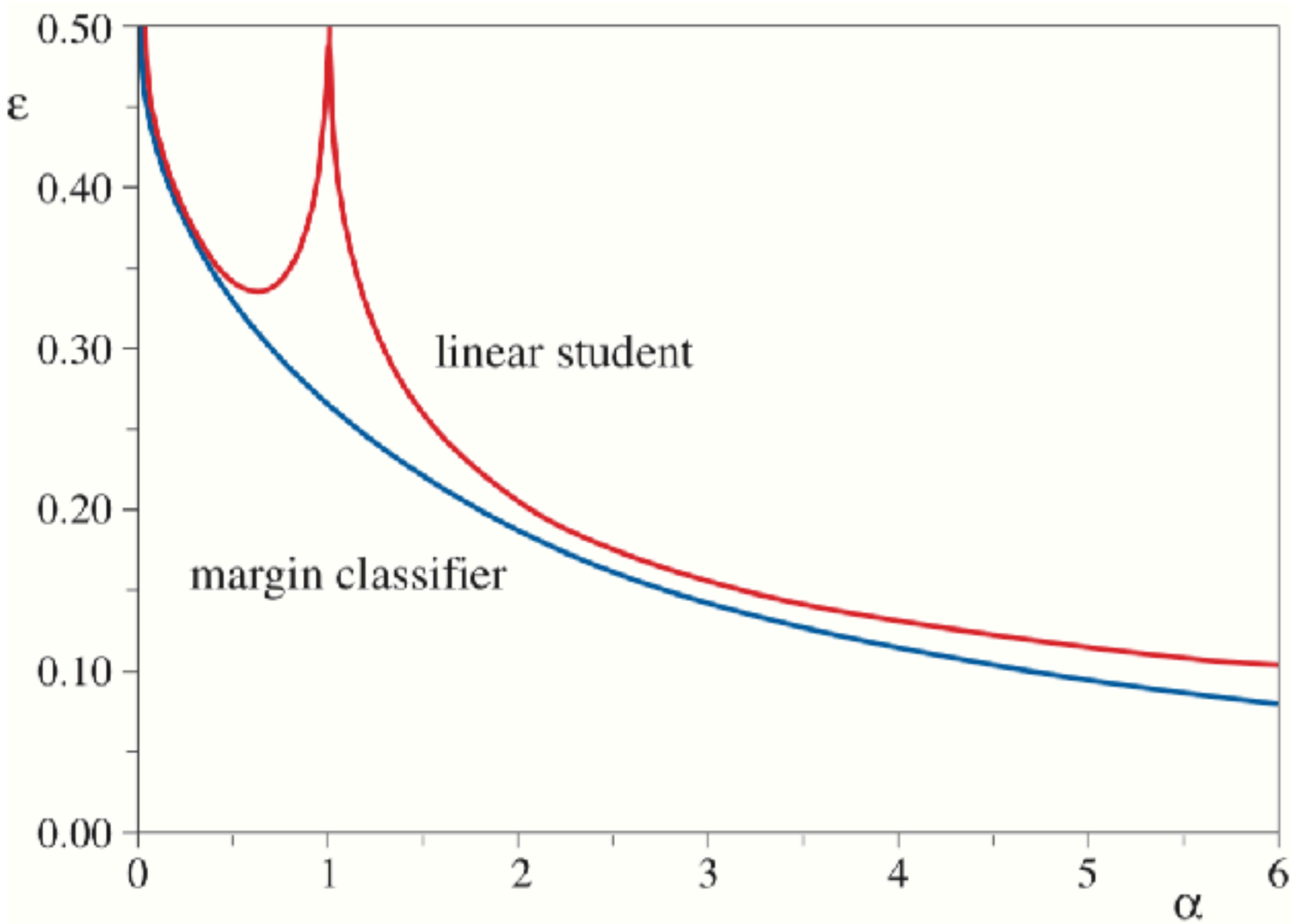


FIGURE 10 Learning curves for a linear student and for a margin classifier. $\alpha = m/N$.

Ok, does this explain double-descent?

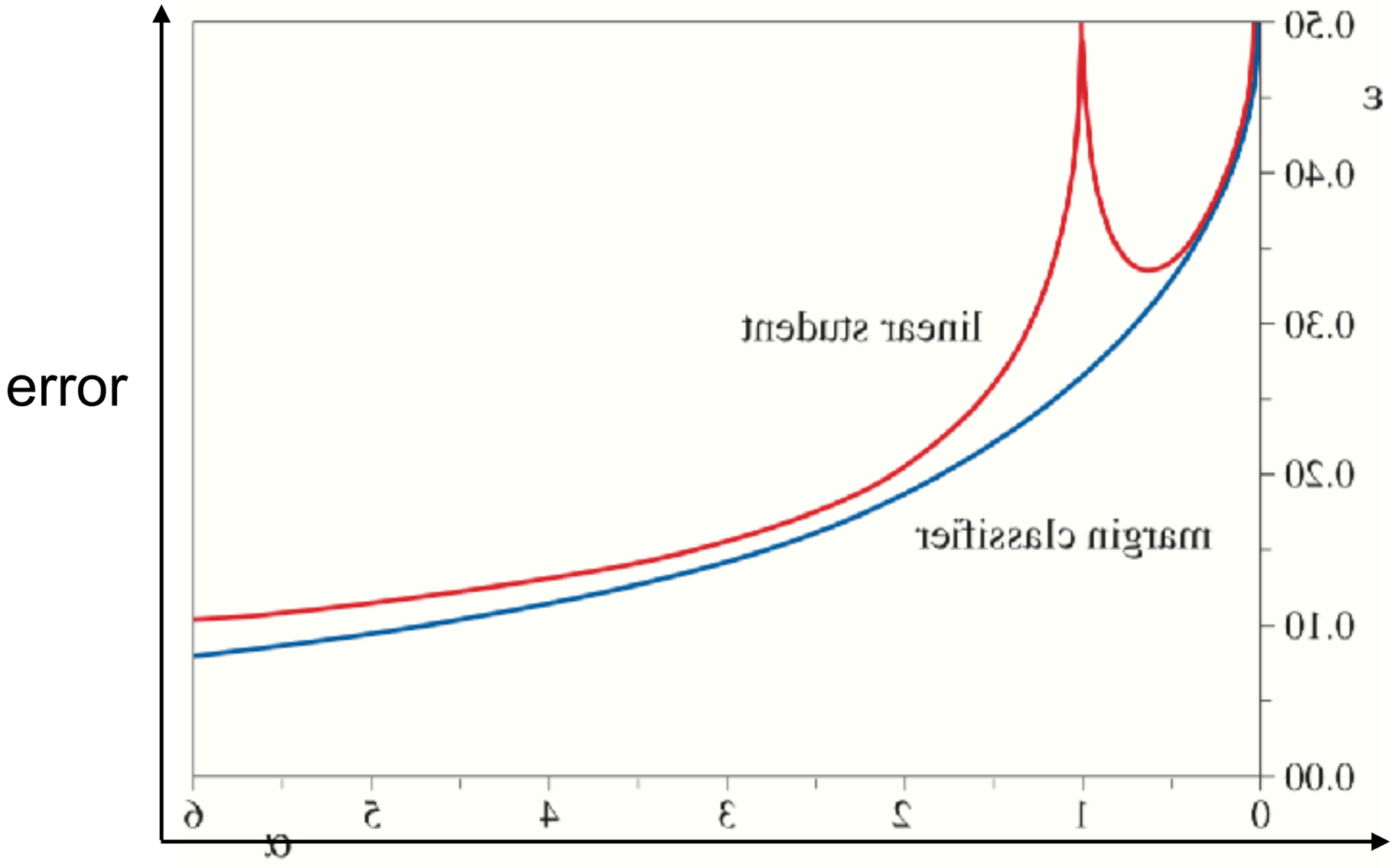
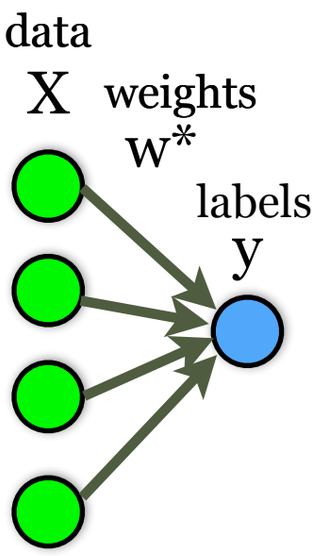


FIGURE 10 Learning curves for a linear student and for a margin classifier. $\alpha = m/N$.

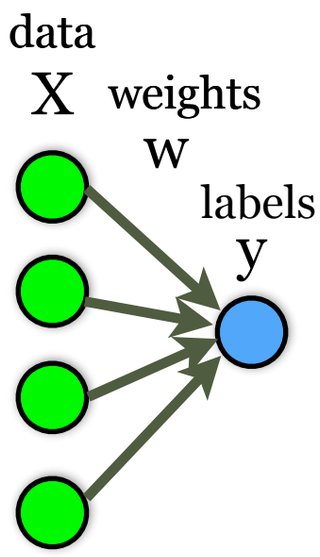
Ok, does this explain double-descent?

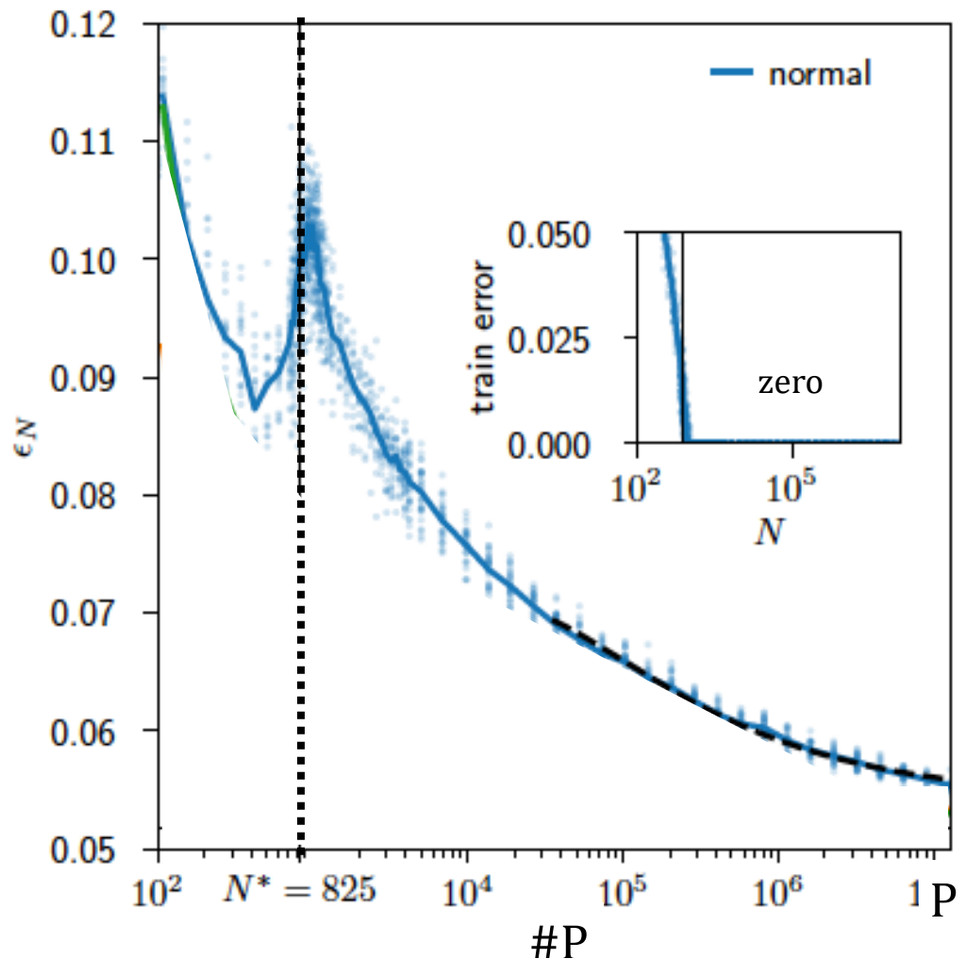
error

Teacher



Student





Parity-MNIST, 5 layers, [Geiger et al.

Complexity

While such models explains non-monotonicity, and the pick close to the exact interpolation threshold, they do not explain the lack of overfitting

Learning with random feature neural networks

Generalisation error in learning with random features and the hidden manifold model

Federica Gerace†, Bruno Loureiro†,

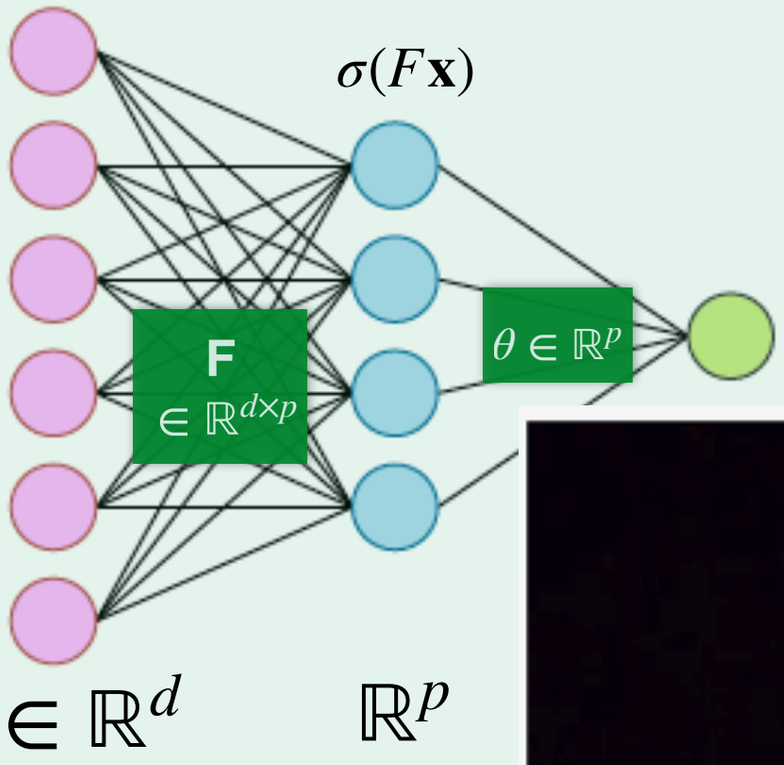
Florent Krzakala*, Marc Mézard *, Lenka Zdeborová†

ICML | 2020

Thirty-seventh International Conference on Machine Learning

Random features neural net

Architecture: Two-layers neural network with *fixed* first layer **F**



Random Features for Large-Scale Ker

Ali Rahimi
Intel Research Seattle
Seattle, WA 98105
ali.rahimi@intel.com

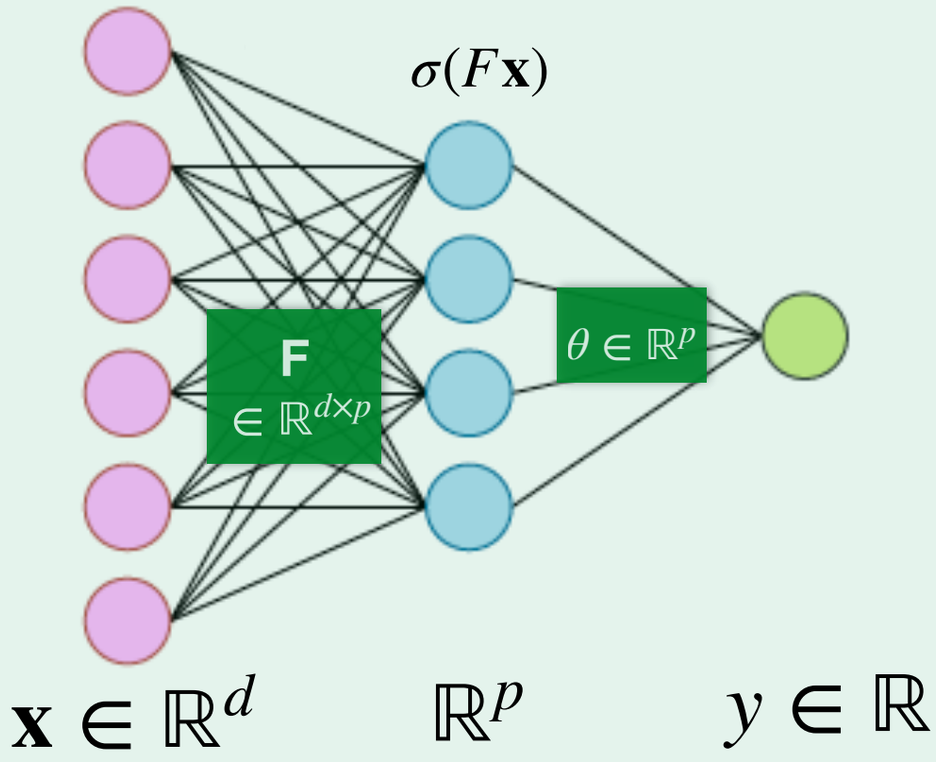
Benjamin
Caltech
Pasadena, C
brecht@ist.c

NIPS ‘07

Ali Rahimi - NIPS 2017 Test-of-Time Award presentation

Random features neural net

Architecture: Two-layers neural network with *fixed* first layer **F**



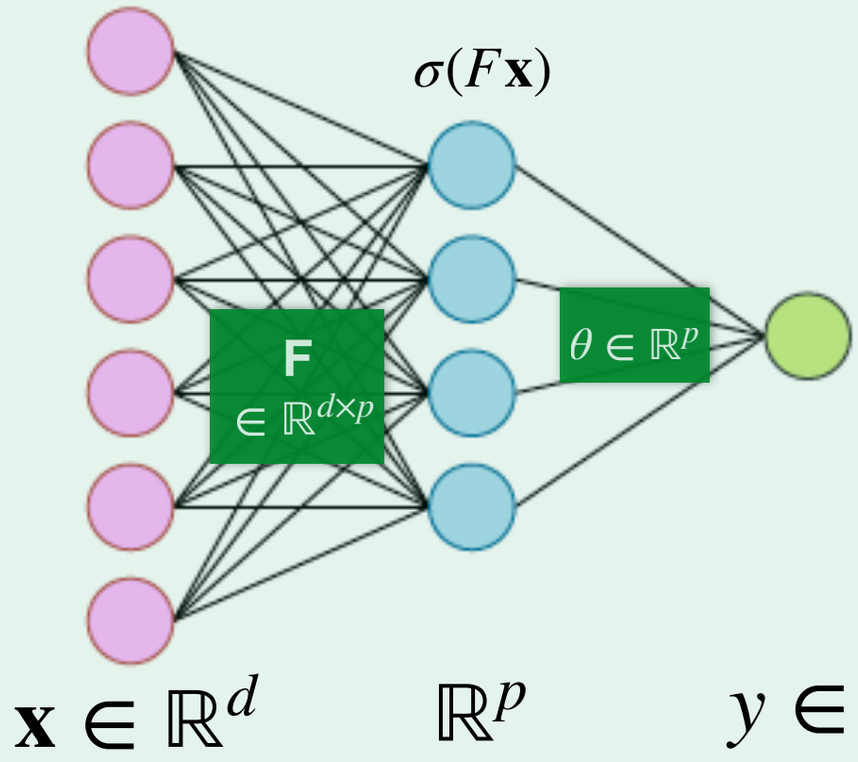
Deep connections with genuine neural networks in the “Lazy regime”
[Jacot, Gabriel, Hongler ’18; Chizat, Bach ’19; Geiger et al. ’19]

Random feature model…

Dataset:

- n vector $\mathbf{x}_i \in \mathbb{R}^d$, drawn randomly from $\mathcal{N}(0, \mathbf{1}_d)$
- n labels y_i given by a function $y_i^0 = f^0(\mathbf{x} \cdot \theta^*)$

Architecture: Two-layers neural network with fixed first layer **F**



Cost function:

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, y_i^0) + \lambda \|\theta\|_2^2 \qquad \ell(\,.\,) = \begin{matrix} \textit{Logistic loss} \\ \textit{Hinge loss} \\ \textit{Square loss} \\ \dots \end{matrix}$$

What is the training error & the generalisation error in the high dimensional limit $(d, p, n) \to \infty$?

… and its solution

[Loureiro, Gerace, **FK**, Mézard, Zdeborova, '20]

Definitions:

Consider the unique fixed point of the following system of equations

$$\begin{cases}
\hat{V}_s = \frac{\alpha}{\gamma}\kappa_1^2 \mathbb{E}_{\xi,y}\left[\mathcal{Z}\left(y,\omega_0\right)\frac{\partial_\omega \eta(y,\omega_1)}{V}\right], \\
\hat{q}_s = \frac{\alpha}{\gamma}\kappa_1^2 \mathbb{E}_{\xi,y}\left[\mathcal{Z}\left(y,\omega_0\right)\frac{\left(\eta(y,\omega_1)-\omega_1\right)^2}{V^2}\right], \\
\hat{m}_s = \frac{\alpha}{\gamma}\kappa_1 \mathbb{E}_{\xi,y}\left[\partial_\omega \mathcal{Z}\left(y,\omega_0\right)\frac{(\eta(y,\omega_1)-\omega_1)}{V}\right], \\
\hat{V}_w = \alpha\kappa_\star^2 \mathbb{E}_{\xi,y}\left[\mathcal{Z}\left(y,\omega_0\right)\frac{\partial_\omega \eta(y,\omega_1)}{V}\right], \\
\hat{q}_w = \alpha\kappa_\star^2 \mathbb{E}_{\xi,y}\left[\mathcal{Z}\left(y,\omega_0\right)\frac{\left(\eta(y,\omega_1)-\omega_1\right)^2}{V^2}\right],
\end{cases}
\quad
\begin{cases}
V_s = \frac{1}{\hat{V}_s}\left(1 - z\ g_\mu(-z)\right), \\
q_s = \frac{\hat{m}_s^2+\hat{q}_s}{\hat{V}_s}\left[1-2zg_\mu(-z)+z^2 g'_\mu(-z)\right] \\
\quad -\frac{\hat{q}_w}{(\lambda+\hat{V}_w)\hat{V}_s}\left[-zg_\mu(-z)+z^2 g'_\mu(-z)\right], \\
m_s = \frac{\hat{m}_s}{\hat{V}_s}\left(1 - z\ g_\mu(-z)\right), \\
V_w = \frac{\gamma}{\lambda+\hat{V}_w}\left[\frac{1}{\gamma}-1+zg_\mu(-z)\right], \\
q_w = \gamma\frac{\hat{q}_w}{(\lambda+\hat{V}_w)^2}\left[\frac{1}{\gamma}-1+z^2 g'_\mu(-z)\right], \\
\quad +\frac{\hat{m}_s^2+\hat{q}_s}{(\lambda+\hat{V}_w)\hat{V}_s}\left[-zg_\mu(-z)+z^2 g'_\mu(-z)\right],
\end{cases}
\quad
\begin{cases}
\eta(y,\omega) = \underset{x\in\mathbb{R}}{\text{argmin}}\left[\frac{(x-\omega)^2}{2V} + \ell(y,x)\right] \\
\mathcal{Z}(y,\omega) = \int \frac{dx}{\sqrt{2\pi V^0}} e^{-\frac{1}{2V^0}(x-\omega)^2}\delta\left(y - f^0(x)\right)
\end{cases}$$

where $V = \kappa_1^2 V_s + \kappa_\star^2 V_w$, $V^0 = \rho - \frac{M^2}{Q}$, $Q = \kappa_1^2 q_s + \kappa_\star^2 q_w$, $M = \kappa_1 m_s$, $\omega_0 = M/\sqrt{Q}\xi$, $\omega_1 = \sqrt{Q}\xi$ and g_μ is the Stieltjes transform of FF^T

$\kappa_0 = \mathbb{E}\left[\sigma(z)\right], \kappa_1 \equiv \mathbb{E}\left[z\sigma(z)\right], \kappa_\star \equiv \mathbb{E}\left[\sigma(z)^2\right] - \kappa_0^2 - \kappa_1^2$, and $\vec{z}^\mu \sim \mathcal{N}(\vec{0}, \mathbf{I_p})$

Then in the high-dimensional limit:

$$\epsilon_{gen} = \mathbb{E}_{\lambda,\nu}\left[\left(f^0(\nu) - \hat{f}(\lambda)\right)^2\right]$$

with $(\nu,\lambda) \sim \mathcal{N}\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}\rho & M^\star \\ M^\star & Q^\star\end{pmatrix}\right)$

$$\mathscr{L}_{\text{training}} = \frac{\lambda}{2\alpha}q_w^\star + \mathbb{E}_{\xi,y}\left[\mathcal{Z}\left(y,\omega_0^\star\right)\ell\left(y,\eta(y,\omega_1^\star)\right)\right]$$

with $\omega_0^\star = M^\star/\sqrt{Q^\star}\xi, \omega_1^\star = \sqrt{Q^\star}\xi$

Agrees with **[Louart , Liao , Couillet'18 & Mei-Montanari '19]** who solved a particular case using random matrix theory: linear function f^0, $\ell(x,y) = \|x-y\|_2^2$ & Gaussian random weights **F**…

… and its solution

[Loureiro, Gerace, **FK**, Mézard, Zdeborova, ICML'20]

Definitions:

Consider the unique fixed point of the following system of equations

$$\begin{cases}
\hat{V}_s = \frac{\alpha}{\gamma}\kappa_1^2 \mathbb{E}_{\xi,y}\left[\mathcal{Z}\left(y,\omega_0\right)\frac{\partial_\omega \eta(y,\omega_1)}{V}\right], \\
\hat{q}_s = \frac{\alpha}{\gamma}\kappa_1^2 \mathbb{E}_{\xi,y}\left[\mathcal{Z}\left(y,\omega_0\right)\frac{\left(\eta(y,\omega_1)-\omega_1\right)^2}{V^2}\right], \\
\hat{m}_s = \frac{\alpha}{\gamma}\kappa_1 \mathbb{E}_{\xi,y}\left[\partial_\omega \mathcal{Z}\left(y,\omega_0\right)\frac{(\eta(y,\omega_1)-\omega_1)}{V}\right], \\
\hat{V}_w = \alpha\kappa_\star^2 \mathbb{E}_{\xi,y}\left[\mathcal{Z}\left(y,\omega_0\right)\frac{\partial_\omega \eta(y,\omega_1)}{V}\right], \\
\hat{q}_w = \alpha\kappa_\star^2 \mathbb{E}_{\xi,y}\left[\mathcal{Z}\left(y,\omega_0\right)\frac{\left(\eta(y,\omega_1)-\omega_1\right)^2}{V^2}\right],
\end{cases}
\qquad
\begin{cases}
V_s = \frac{1}{\hat{V}_s}\left(1 - z\ g_\mu(-z)\right), \\
q_s = \frac{\hat{m}_s^2+\hat{q}_s}{\hat{V}_s}\left[1-2zg_\mu(-z)+z^2 g'_\mu(-z)\right] \\
\quad -\frac{\hat{q}_w}{(\lambda+\hat{V}_w)\hat{V}_s}\left[-zg_\mu(-z)+z^2 g'_\mu(-z)\right], \\
m_s = \frac{\hat{m}_s}{\hat{V}_s}\left(1 - z\ g_\mu(-z)\right), \\
V_w = \frac{\gamma}{\lambda+\hat{V}_w}\left[\frac{1}{\gamma}-1+zg_\mu(-z)\right], \\
q_w = \gamma\frac{\hat{q}_w}{(\lambda+\hat{V}_w)^2}\left[\frac{1}{\gamma}-1+z^2 g'_\mu(-z)\right], \\
\quad +\frac{\hat{m}_s^2+\hat{q}_s}{(\lambda+\hat{V}_w)\hat{V}_s}\left[-zg_\mu(-z)+z^2 g'_\mu(-z)\right],
\end{cases}
\qquad
\begin{cases}
\eta(y,\omega) = \underset{x\in\mathbb{R}}{\text{argmin}}\left[\frac{(x-\omega)^2}{2V} + \ell(y,x)\right] \\
\mathcal{Z}(y,\omega) = \int \frac{dx}{\sqrt{2\pi V^0}} e^{-\frac{1}{2V^0}(x-\omega)^2}\delta\left(y - f^0(x)\right)
\end{cases}$$

where $V = \kappa_1^2 V_s + \kappa_\star^2 V_w$, $V^0 = \rho - \frac{M^2}{Q}$, $Q = \kappa_1^2 q_s + \kappa_\star^2 q_w$, $M = \kappa_1 m_s$, $\omega_0 = M/\sqrt{Q}\xi$, $\omega_1 = \sqrt{Q}\xi$ and g_μ is the Stieltjes transform of FF^T

$\kappa_0 = \mathbb{E}\left[\sigma(z)\right], \kappa_1 \equiv \mathbb{E}\left[z\sigma(z)\right], \kappa_\star \equiv \mathbb{E}\left[\sigma(z)^2\right] - \kappa_0^2 - \kappa_1^2$ and $\vec{z}^\mu \sim \mathcal{N}(\vec{0}, \mathbf{I_p})$

Then in the high-dimensional limit:

$$\epsilon_{gen} = \mathbb{E}_{\lambda,\nu}\left[\left(f^0(\nu) - \hat{f}(\lambda)\right)^2\right]$$

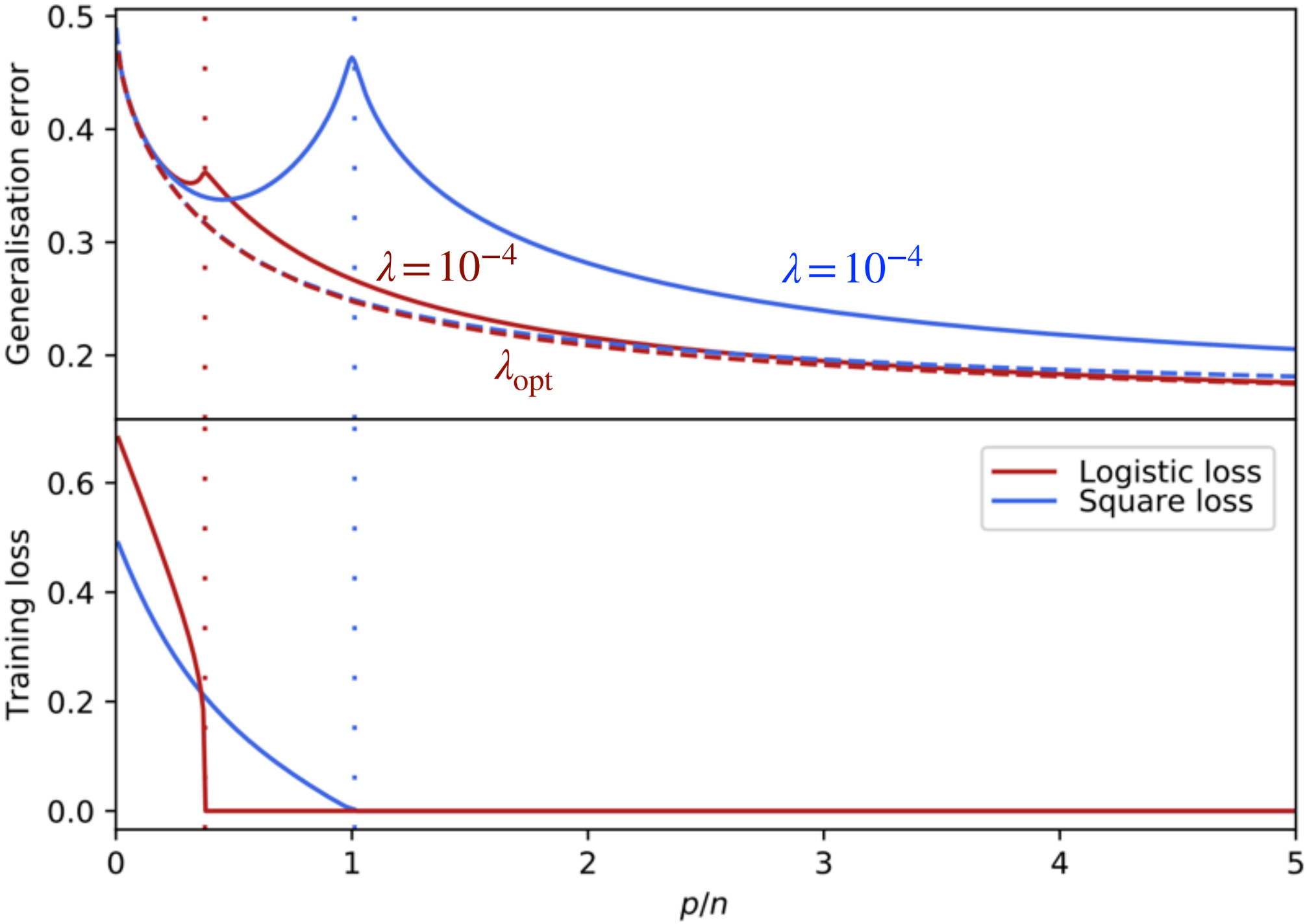
with $(\nu,\lambda) \sim \mathcal{N}\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}\rho & M^\star \\ M^\star & Q^\star\end{pmatrix}\right)$

$$\mathscr{L}_{\text{training}} = \frac{\lambda}{2\alpha}q_w^\star + \mathbb{E}_{\xi,y}\left[\mathcal{Z}\left(y,\omega_0^\star\right)\ell\left(y,\eta(y,\omega_1^\star)\right)\right]$$

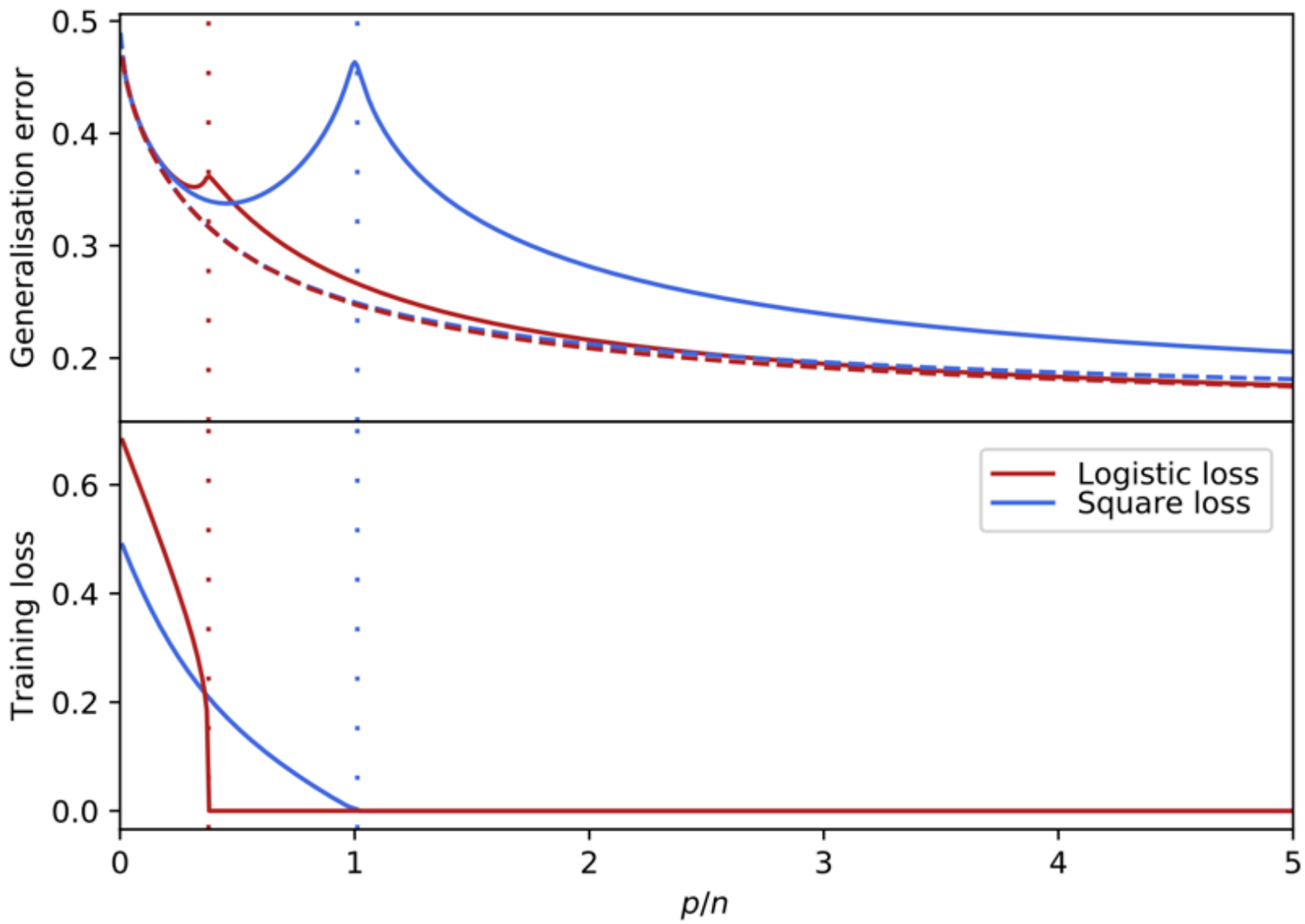
with $\omega_0^\star = M^\star/\sqrt{Q^\star}\xi, \omega_1^\star = \sqrt{Q^\star}\xi$

… and recently proven in full generality by **[Dhifallah, Lu, '20]**

A classification task



A classification task



Implicit regularisation of gradient descent [Rosset, Zhy, Hastie, '04] [Neyshabur, Tomyoka, Srebro, '15]

As $\lambda \to 0$*, in the overparametrized regime,*
Logistic converges to max-margin, ℓ_2 *converges to least norm*

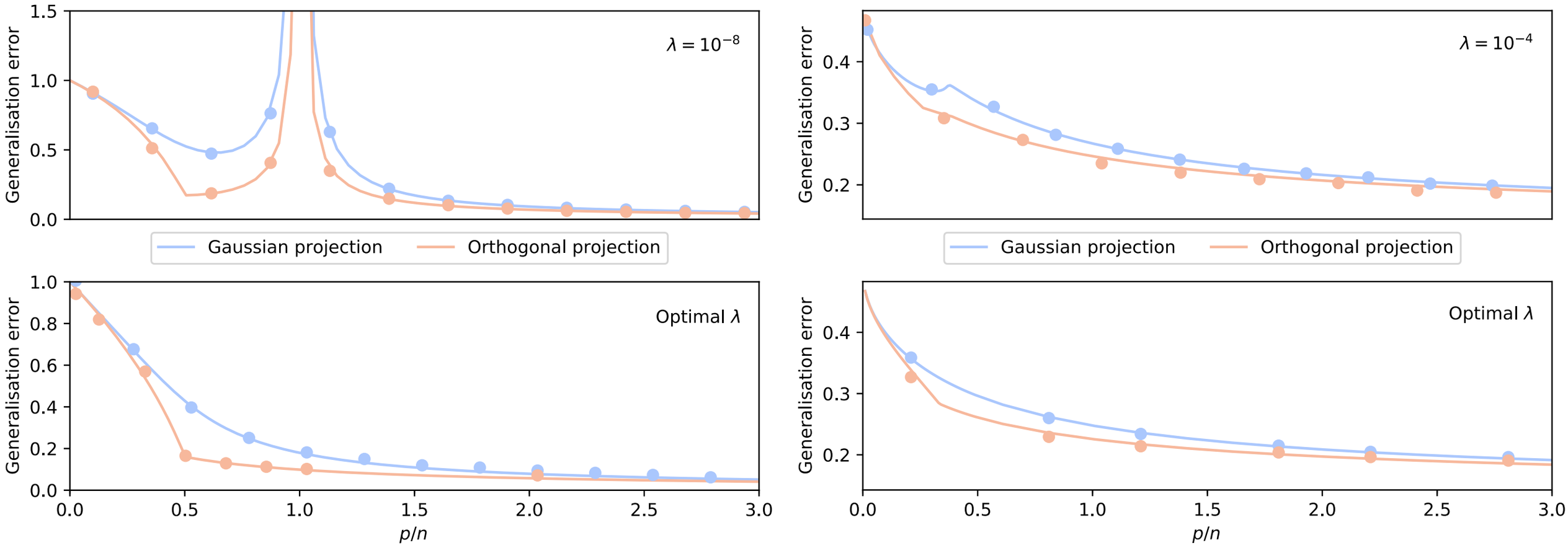
Asymptotics accurate even at d=200!

Regression task

ℓ_2 loss

Classification task

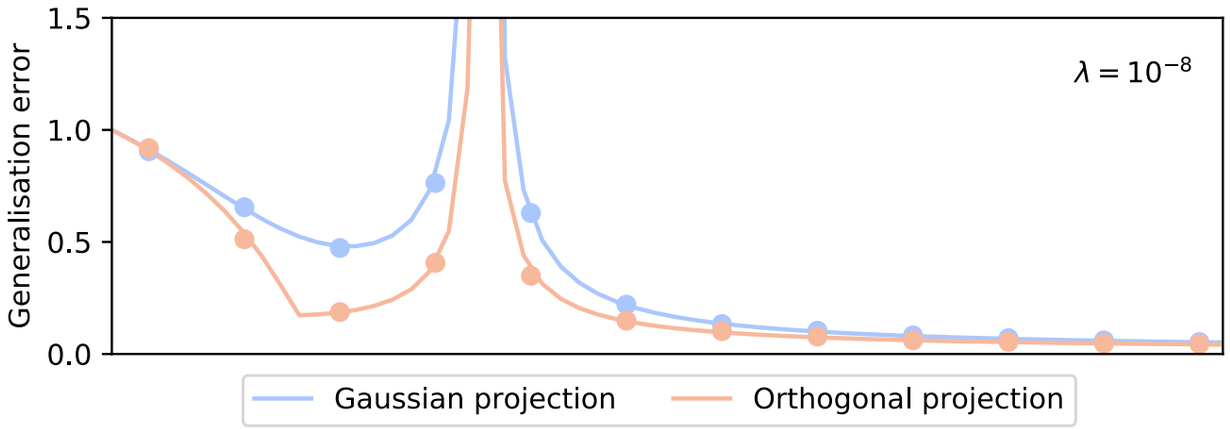
logistic loss

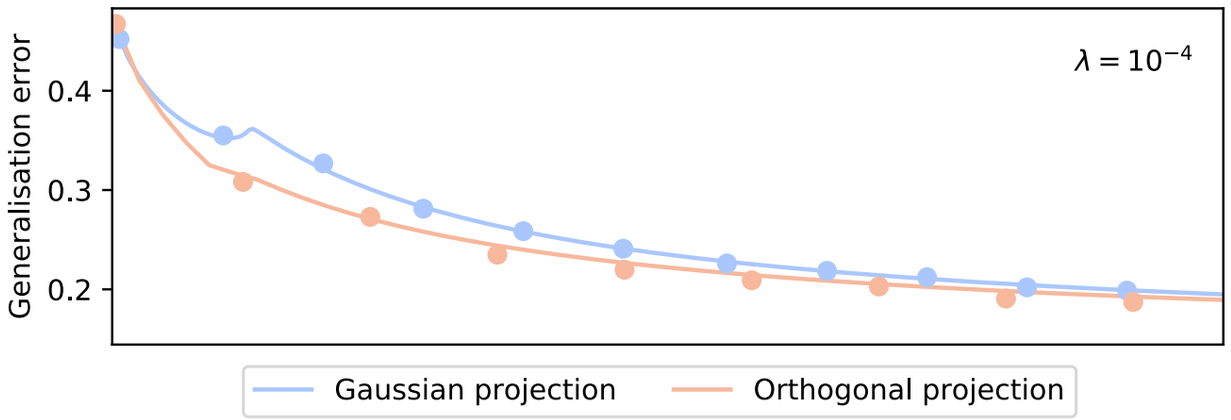


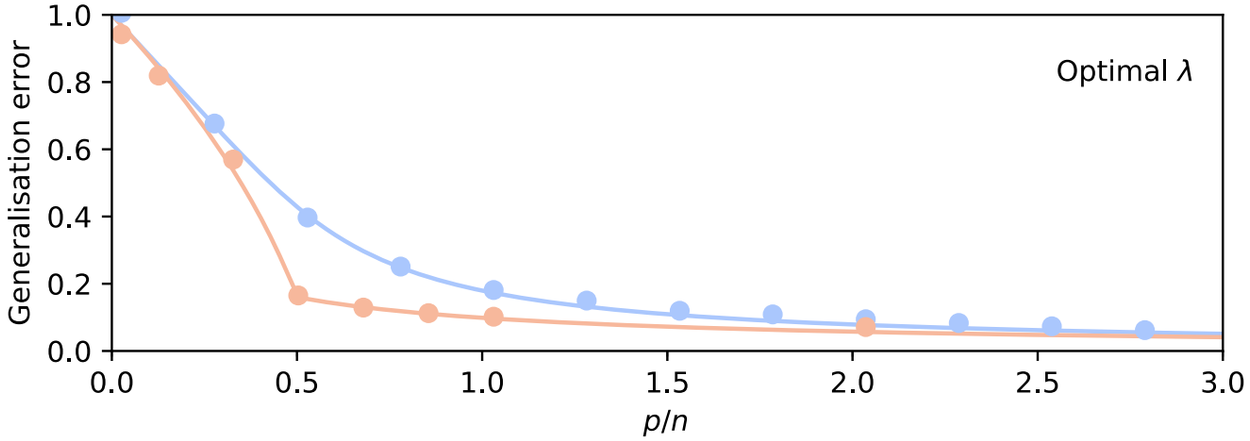
First layer: random Gaussian Matrix

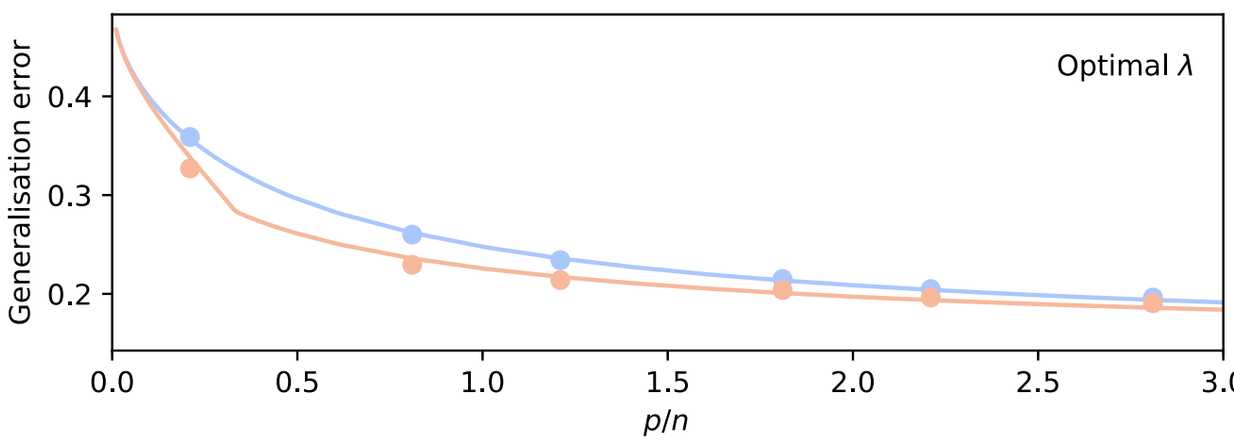
First layer: subsampled Fourier matrix

Regularisation & different First Layer









The Unreasonable Effectiveness of Structured Random Orthogonal Embeddings

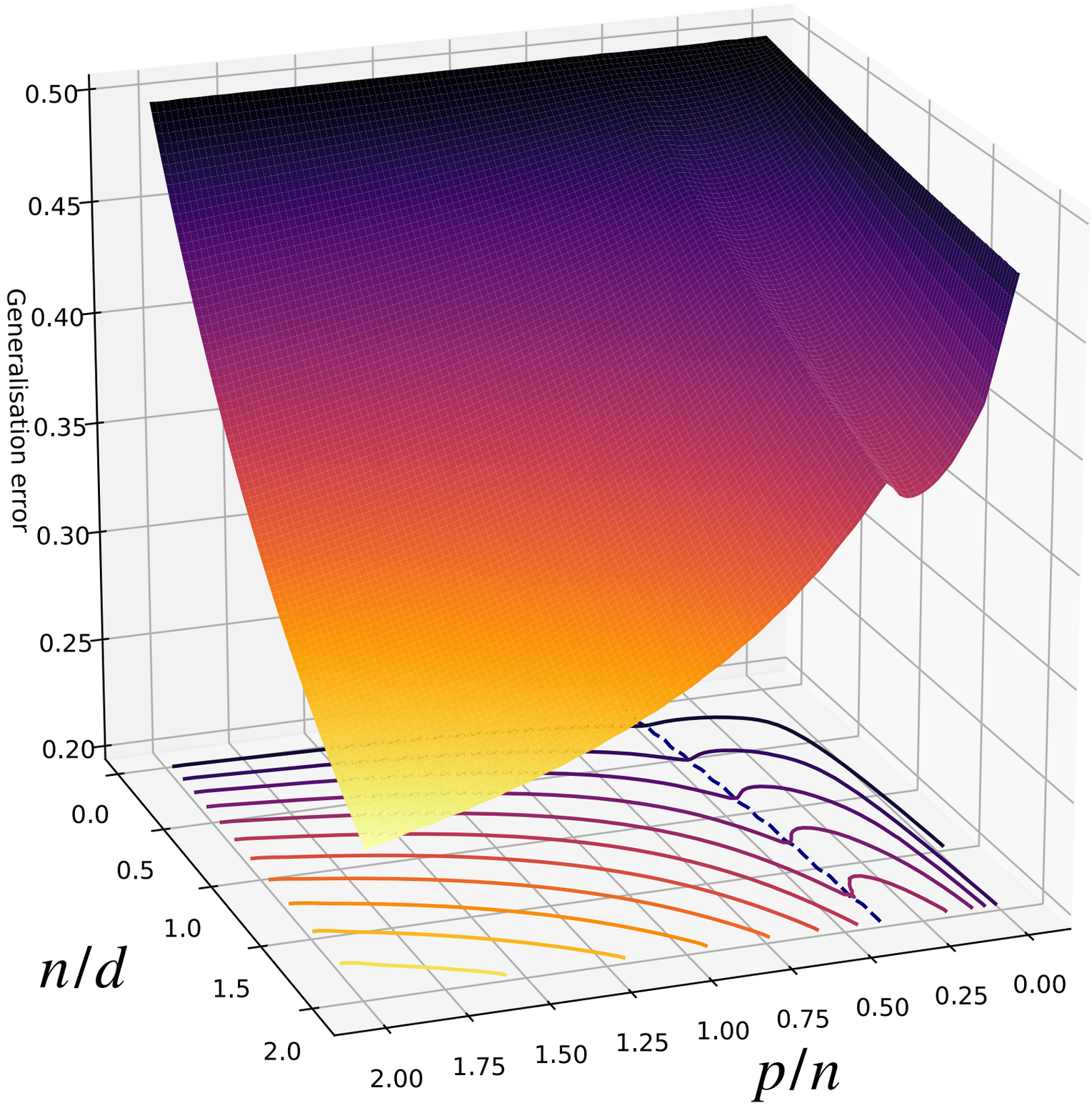
Krzysztof Choromanski *
Google Brain Robotics
kchoro@google.com

Mark Rowland *
University of Cambridge
mr504@cam.ac.uk

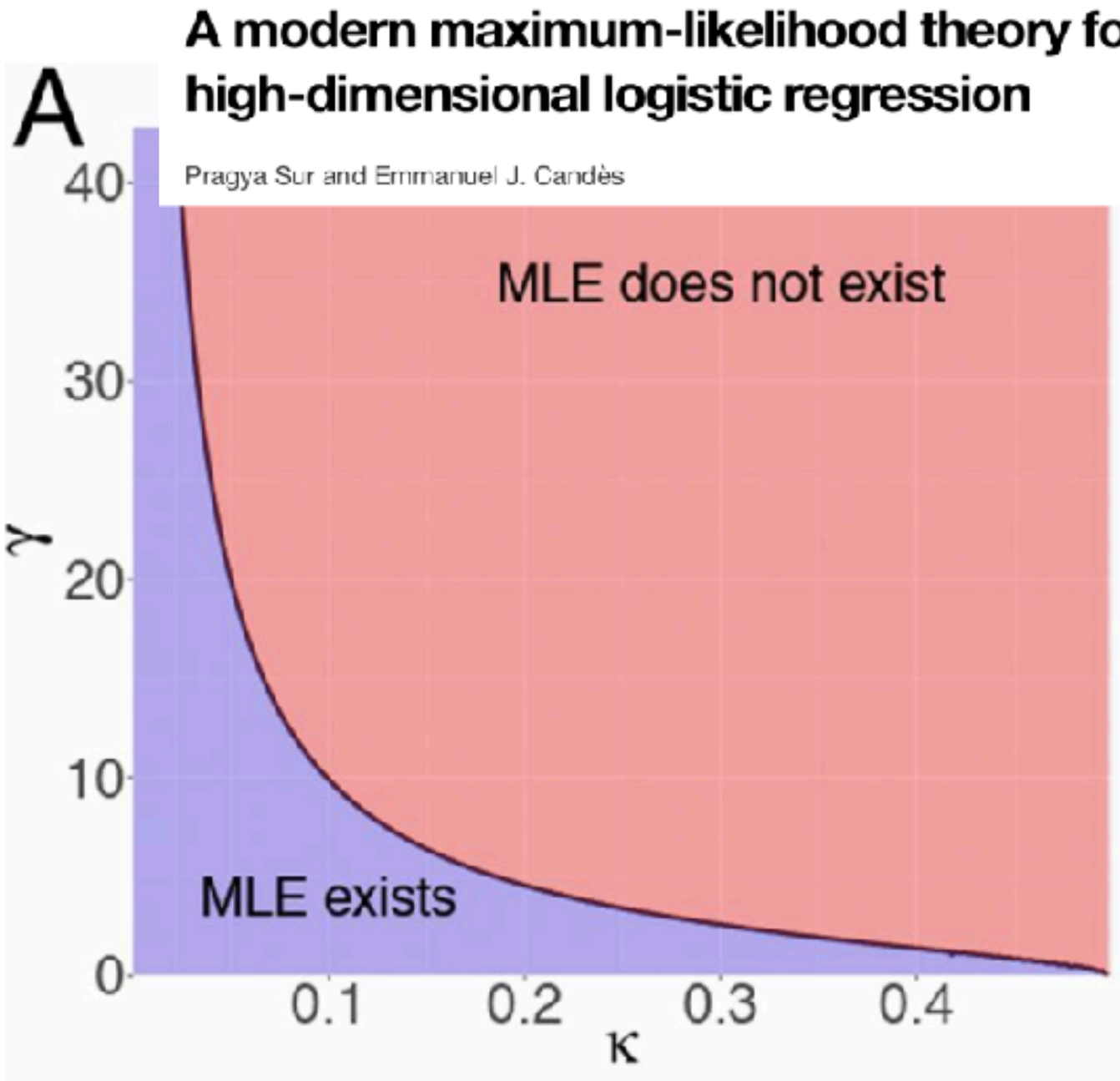
Adrian Weller
University of Cambridge and Alan Turing Institute
aw665@cam.ac.uk

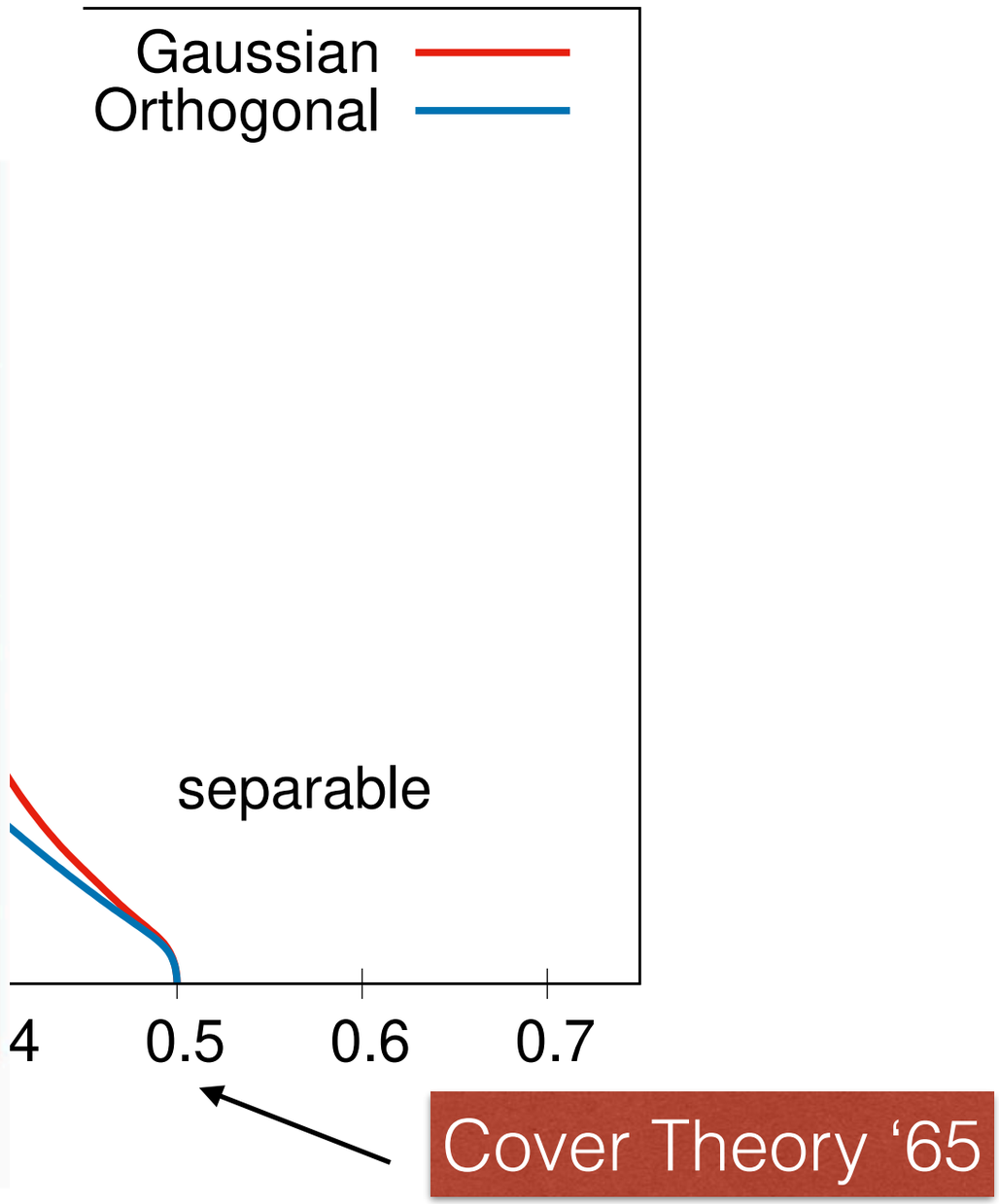
Nips '17

Logistic loss, no regularisation



Phase transition of perfect separability





Generalize a phase transition discussed
by [Cover ’65; Gardner ’87; Sur & Candes, ’18]

We now see over-parametrisation does not hurt, but why?

Bias-Variance reloaded…

Double Trouble in Double Descent: Bias and Variance(s) in the Lazy Regime

Stéphane d'Ascoli [*1] Maria Refinetti [*1] Giulio Biroli [1] Florent Krzakala [1]

ICML | 2020

Thirty-seventh International Conference
on Machine Learning

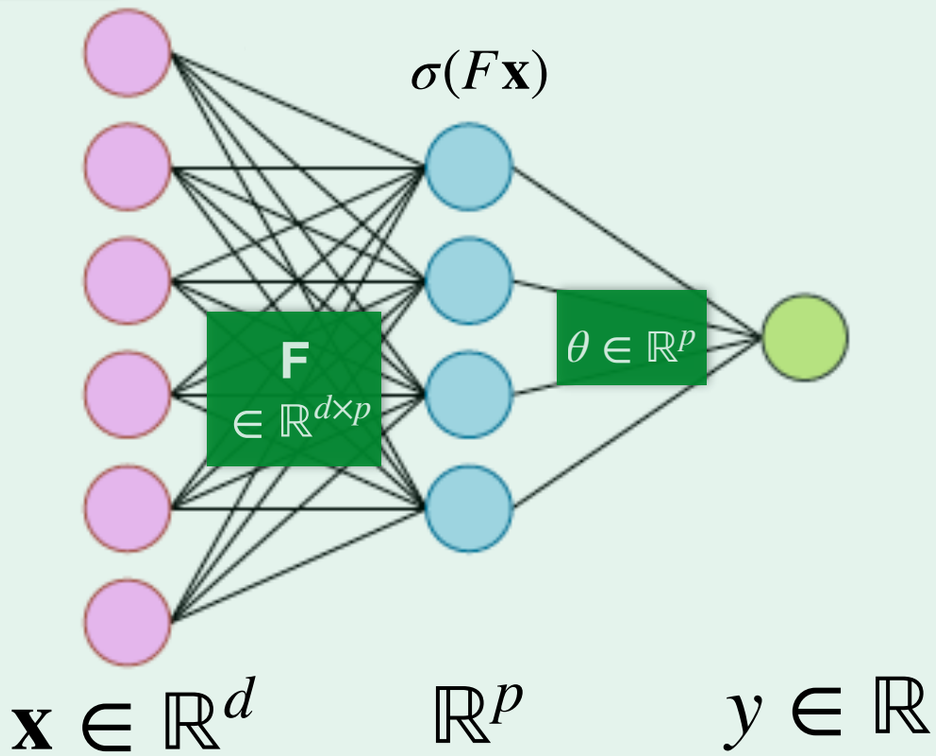
Double Trouble in Double Descent: Bias and Variance(s) in the Lazy Regime

Stéphane d'Ascoli [*1] Maria Refinetti [*1] Giulio Biroli [1] Florent Krzakala [1]

Dataset:

- n vector $\mathbf{x}_i \in \mathbb{R}^d$, drawn randomly from $\mathcal{N}(\mathbf{0},\mathbf{1}_d)$
- n labels y_i given by a function $y_i^0 = f^0(\mathbf{x} \cdot \theta^*)$

Architecture: Two-layers neural network with fixed first layer **F**



Cost function:

$$\mathscr{L} = \frac{1}{n}\sum_{i=1}^{n} \|y_i - y_i^0\|_2^2 + \lambda\|\theta\|_2^2$$

Square loss

Generalization Error = Bias + Variance

$$\mathcal{E}_{\text{Bias}} = \mathop{\mathbb{E}}_{\boldsymbol{x}} \left[\left(\langle \boldsymbol{\beta}, \boldsymbol{x} \rangle - \mathop{\mathbb{E}}_{\boldsymbol{X}, \boldsymbol{\Theta}, \boldsymbol{\varepsilon}} \left[\hat{f}(\boldsymbol{x}) \right] \right)^2 \right]$$

Bias
(Unavoidable error)

Generalization Error = Bias + Variance

$$\mathcal{E}_{\text{Bias}} = \mathop{\mathbb{E}}_{\boldsymbol{x}}\left[\left(\langle \boldsymbol{\beta}, \boldsymbol{x}\rangle - \mathop{\mathbb{E}}_{\boldsymbol{X},\boldsymbol{\Theta},\boldsymbol{\varepsilon}}\left[\hat{f}(\boldsymbol{x})\right]\right)^2\right]$$

Bias
(Unavoidable error)

Generalization Error = Bias + Variance

Variance

Noise
labels

Variance

Sampling
(Finite training set)

Variance

Initialization:
Choice of the
random features

$$\mathcal{E}_{\text{Noise}} = \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{X},\boldsymbol{\Theta}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}}\left[\hat{f}(\boldsymbol{x})^2\right] - \left(\mathop{\mathbb{E}}_{\boldsymbol{\varepsilon}}\left[\hat{f}(\boldsymbol{x})\right]\right)^2\right]$$

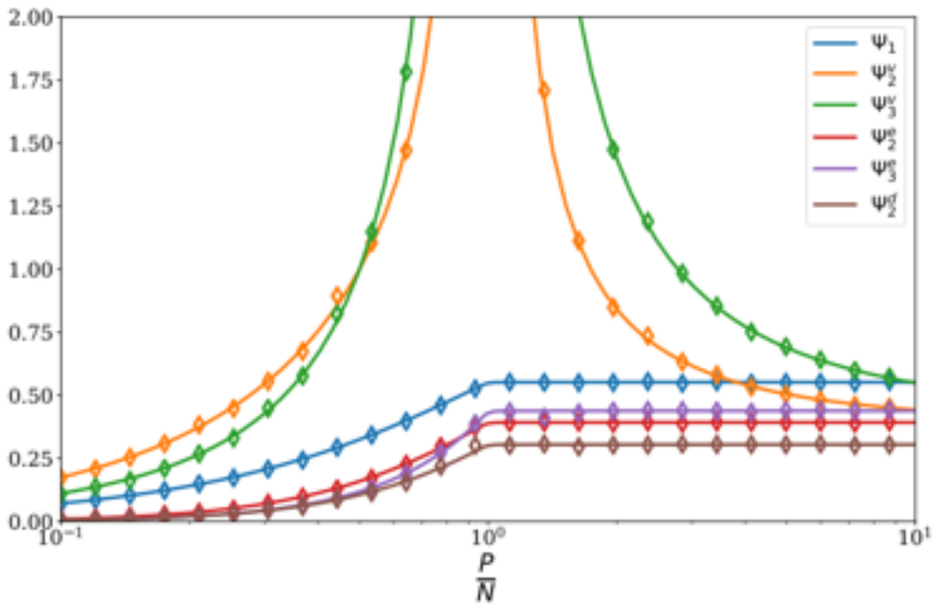
$$\mathcal{E}_{\text{Init}} = \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{X}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\Theta}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}}\left[\hat{f}(\boldsymbol{x})\right]^2\right] - \mathop{\mathbb{E}}_{\boldsymbol{\Theta},\epsilon}\left[\hat{f}(\boldsymbol{x})\right]^2\right]$$

$$\mathcal{E}_{\text{Samp}} = \mathop{\mathbb{E}}_{\boldsymbol{x}}\left[\mathop{\mathbb{E}}_{\boldsymbol{X}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\Theta},\boldsymbol{\varepsilon}}\left[\hat{f}(\boldsymbol{x})\right]^2\right] - \mathop{\mathbb{E}}_{\boldsymbol{X},\boldsymbol{\Theta},\boldsymbol{\varepsilon}}\left[\hat{f}(\boldsymbol{x})\right]^2\right]$$

Double Trouble in Double Descent: Bias and Variance(s) in the Lazy Regime

Stéphane d'Ascoli [*1] Maria Refinetti [*1] Giulio Biroli [1] Florent Krzakala [1]

All these terms can be computed exactly using random metric theory & statistical physics methods (See paper)



$$
\begin{aligned}
\mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{\varepsilon},\boldsymbol{\Theta},\boldsymbol{X}}\left[\langle\boldsymbol{\beta},\boldsymbol{x}\rangle\hat{f}(\boldsymbol{x})\right] &= F^2\Psi_1,\\
\mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{\Theta},\boldsymbol{X}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\varepsilon}}\left[\hat{f}(\boldsymbol{x})\right]^2\right] &= F^2\Psi_2^v,\\
\mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{\Theta},\boldsymbol{X}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\varepsilon}}\left[\hat{f}(\boldsymbol{x})^2\right]-\mathop{\mathbb{E}}_{\boldsymbol{\varepsilon}}\left[\hat{f}(\boldsymbol{x})\right]^2\right] &= \tau^2\Psi_3^v,\\
\mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{X}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\varepsilon},\boldsymbol{\Theta}}\left[\hat{f}(\boldsymbol{x})\right]^2\right] &= F^2\Psi_2^e,\\
\mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{X}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\varepsilon},\boldsymbol{\Theta}}\left[\hat{f}(\boldsymbol{x})^2\right]-\mathop{\mathbb{E}}_{\boldsymbol{\varepsilon},\boldsymbol{\Theta}}\left[\hat{f}(\boldsymbol{x})\right]^2\right] &= \tau^2\Psi_3^e,\\
\mathop{\mathbb{E}}_{\boldsymbol{x}}\left[\mathbf{E}_{\boldsymbol{\varepsilon},\boldsymbol{\Theta},\boldsymbol{X}}\left[\hat{f}(\boldsymbol{x})\right]^2\right] &= F^2\Psi_2^d,
\end{aligned}
$$

It was shown in [11] that the random features model is equivalent, in the high-dimensional limit of Assumption 2, to a Gaussian covariate model in which the activation function σ is replaced as:

$$
\sigma\left(\frac{\boldsymbol{\Theta}_{hi}^{(k)}\boldsymbol{X}_{\mu i}}{\sqrt{D}}\right) \to \mu_0 + \mu_1\frac{\boldsymbol{\Theta}_{hi}^{(k)}\boldsymbol{X}_{\mu i}}{\sqrt{D}} + \mu_*\boldsymbol{W}_{\mu h}^{(k)}, \tag{49}
$$

with $\boldsymbol{W}^{(k)} \in \mathbb{R}^{N\times P}$, $\boldsymbol{W}_{\mu h}^{(k)} \sim \mathcal{N}(0,1)$ and μ_0, μ_1 and μ_* defined in (29). To simplify the calculations, we take $\mu_0 = 0$, which amounts to adding a constant term to the activation function σ.
This powerful mapping allows to express the quantities U, V. We will not repeat their calculations here: the only difference here is $\boldsymbol{U}^{kl}$, which carries extra indices k, l due to the different initialization of the random features $\boldsymbol{\Theta}^{(k)}$. In our case,

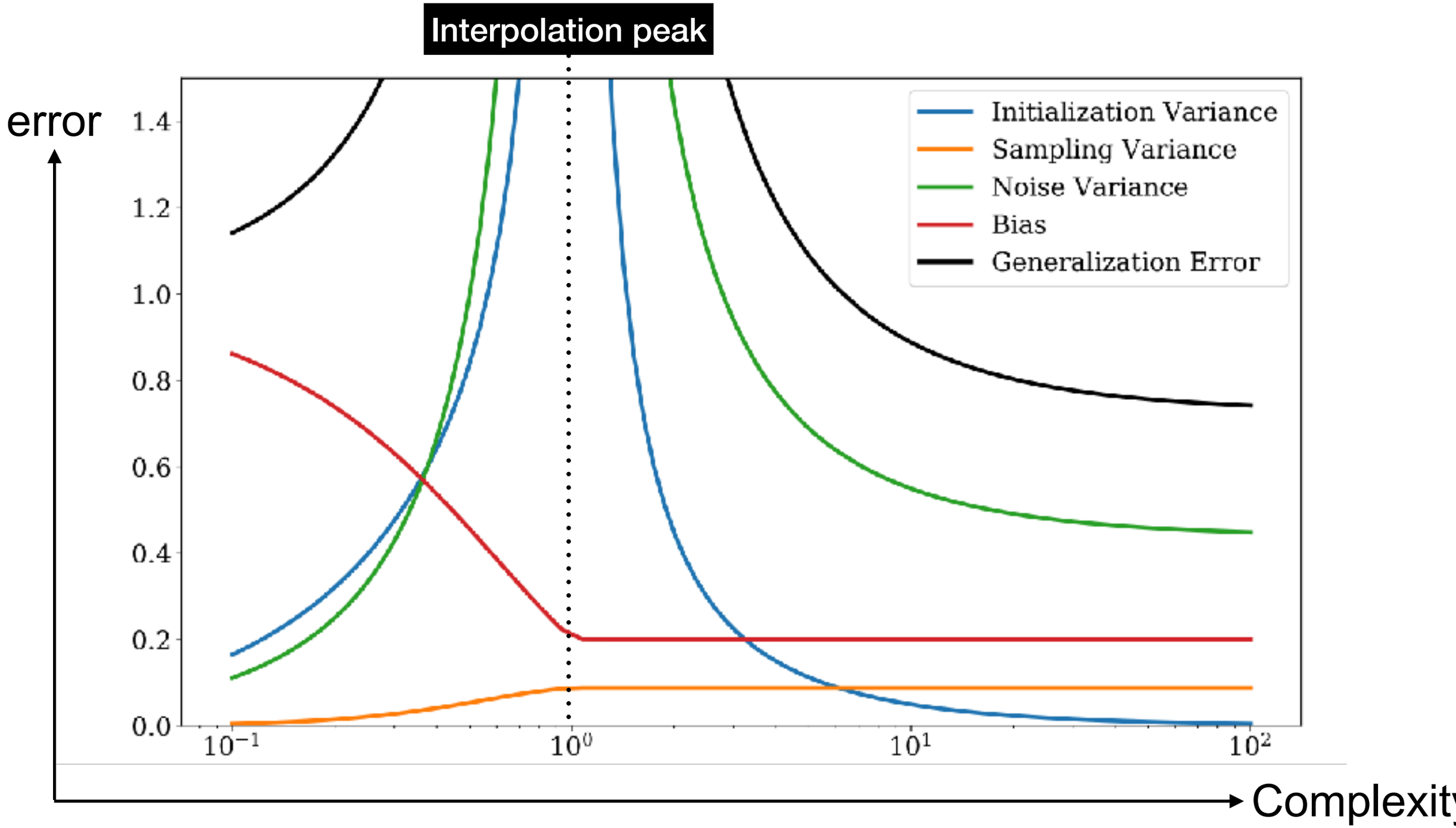
$$
\boldsymbol{U}_{hh'}^{(kl)} = \frac{\mu_1^2}{D}\boldsymbol{\Theta}_{hi}^{(k)}\boldsymbol{\Theta}_{h'i}^{(l)} + \mu_*^2\hat{\psi}_{kl}\delta_{hh'}. \tag{50}
$$

Hence we can rewrite the generalization error as

$$
\mathop{\mathbb{E}}_{(\boldsymbol{\Theta}^{(k)}),\boldsymbol{X},\boldsymbol{\varepsilon}}\left[\mathcal{R}_{\mathrm{RF}}\right] = F^2\left(1 - 2\Psi_1^v\right) + \frac{1}{K}\left(F^2\Psi_2^v + \tau^2\Psi_3^v\right) + \left(1 - \frac{1}{K}\right)\left(F^2\Psi_2^e + \tau^2\Psi_3^e\right), \tag{51}
$$

where $\Psi_1, \Psi_2^v, \Psi_2^e, \Psi_3^v, \Psi_3^e$ are given by:

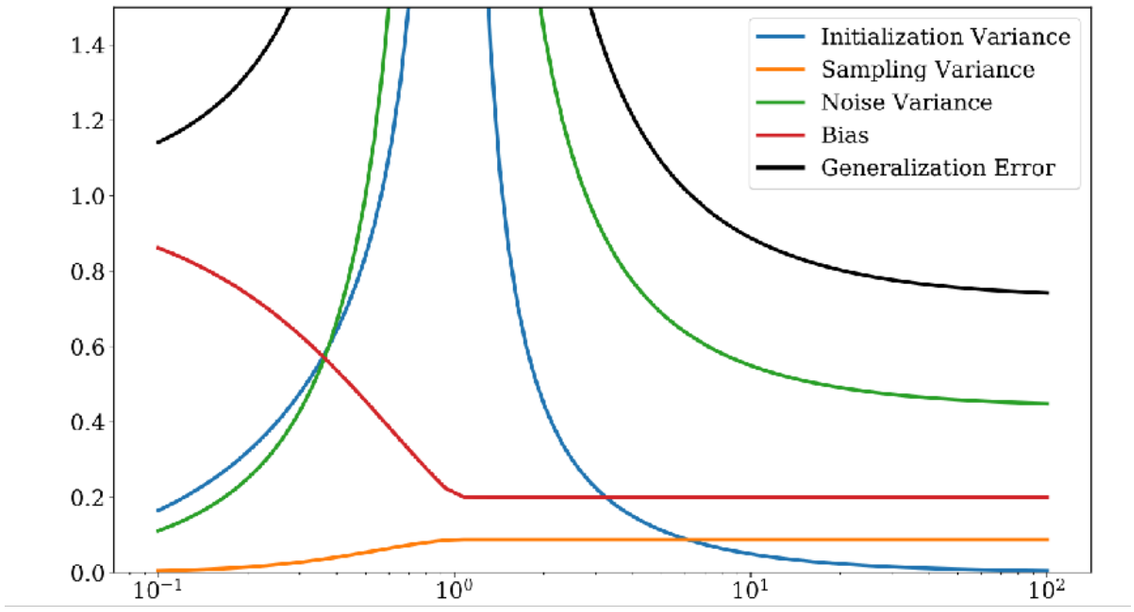
$$
\begin{aligned}
\Psi_1 &= \frac{1}{D}\mathrm{Tr}\left[\left(\frac{\mu_1}{D}\boldsymbol{X}\boldsymbol{\Theta}^{(1)\top}\right)^\top \boldsymbol{Z}^{(1)}\left(\boldsymbol{Z}^{(1)\top}\boldsymbol{Z}^{(1)} - \psi_1\psi_2\lambda\mathbf{I}_N\right)^{-1}\right],\\
\Psi_2^v &= \frac{1}{D}\mathrm{Tr}\left[\left(\boldsymbol{Z}^{(1)\top}\boldsymbol{Z}^{(1)} + \psi_1\psi_2\lambda\mathbf{I}_N\right)^{-1}\left(\frac{\mu_1^2}{D}\boldsymbol{\Theta}^{(1)}\boldsymbol{\Theta}^{(1)\top} + \mu_*^2\mathbf{I}_N\right)\left(\boldsymbol{Z}^{(1)\top}\boldsymbol{Z}^{(1)} - \psi_1\psi_2\lambda\mathbf{I}_N\right)^{-1}\boldsymbol{Z}^{(1)\top}\left(\frac{1}{D}\boldsymbol{X}\boldsymbol{X}^\top\right)\boldsymbol{Z}^{(1)}\right]\\
\Psi_3^v &= \frac{1}{D}\mathrm{Tr}\left[\left(\boldsymbol{Z}^{(1)\top}\boldsymbol{Z}^{(1)} + \psi_1\psi_2\lambda\mathbf{I}_N\right)^{-1}\left(\frac{\mu_1^2}{D}\boldsymbol{\Theta}^{(1)}\boldsymbol{\Theta}^{(1)\top} + \mu_*^2\mathbf{I}_N\right)\left(\boldsymbol{Z}^{(1)\top}\boldsymbol{Z}^{(1)} - \psi_1\psi_2\lambda\mathbf{I}_N\right)^{-1}\boldsymbol{Z}^{(1)\top}\boldsymbol{Z}^{(1)}\right],\\
\Psi_2^e &= \frac{1}{D}\mathrm{Tr}\left[\left(\boldsymbol{Z}^{(1)\top}\boldsymbol{Z}^{(1)} + \psi_1\psi_2\lambda\mathbf{I}_N\right)^{-1}\left(\frac{\mu_1^2}{D}\boldsymbol{\Theta}^{(1)}\boldsymbol{\Theta}^{(2)\top}\right)\left(\boldsymbol{Z}^{(2)\top}\boldsymbol{Z}^{(2)} - \psi_1\psi_2\lambda\mathbf{I}_N\right)^{-1}\boldsymbol{Z}^{(2)\top}\left(\frac{1}{D}\boldsymbol{X}\boldsymbol{X}^\top\right)\boldsymbol{Z}^{(1)}\right],\\
\Psi_3^e &= \frac{1}{D}\mathrm{Tr}\left[\left(\boldsymbol{Z}^{(1)\top}\boldsymbol{Z}^{(1)} + \psi_1\psi_2\lambda\mathbf{I}_N\right)^{-1}\left(\frac{\mu_1^2}{D}\boldsymbol{\Theta}^{(1)}\boldsymbol{\Theta}^{(2)\top}\right)\left(\boldsymbol{Z}^{(2)\top}\boldsymbol{Z}^{(2)} - \psi_1\psi_2\lambda\mathbf{I}_N\right)^{-1}\boldsymbol{Z}^{(2)\top}\boldsymbol{Z}^{(1)}\right].
\end{aligned}
$$



Bias decreases and has a phase transition at interpolation

Noise & Initialitation variances diverges at the peak…

… but decay later on (self-averaging!)



Over-parametrization here helps because of self-averaging effect for large networks!

In a nutshell: the networks learn “many time” the same sub-network plus fluctuations, and averaging these networks leads to reduced variance

See also: [Geiger, et al, ’19, Spigler, Geiger, Ascoli, Sagun, Biroli, Wyart, ’19, Ascoli, Sagun, Biroli ’20, Lin & Dobriban ’20, Adlam, Pennington ’20]

Neural networks and the **bias/variance** dilemma

S **Geman**, E Bienenstock, R Doursat - Neural computation, 1992 - MIT Press

Feedforward neural networks trained by error backpropagation are examples of nonparametric regression estimators. We present a tutorial on nonparametric inference and its relation to neural networks, and we use the statistical viewpoint to highlight strengths and …

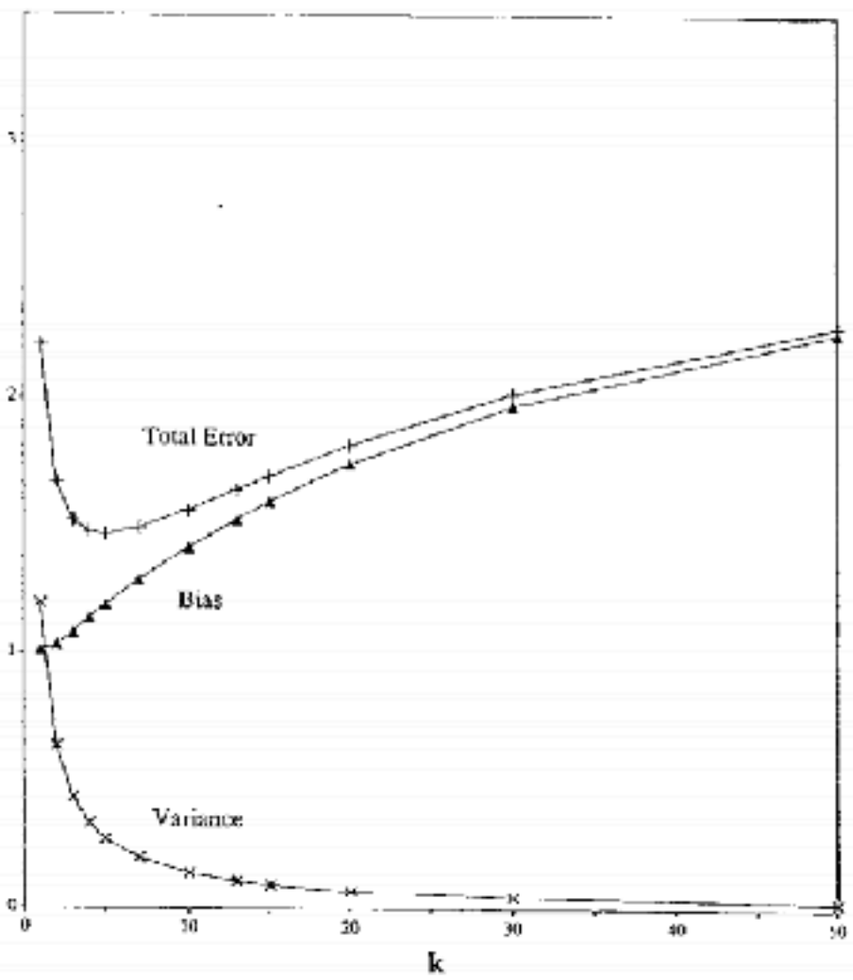


Figure 13: Nearest-neighbor regression for handwritten numeral recognition. Bias, variance, and total error as a function of number of neighbors.

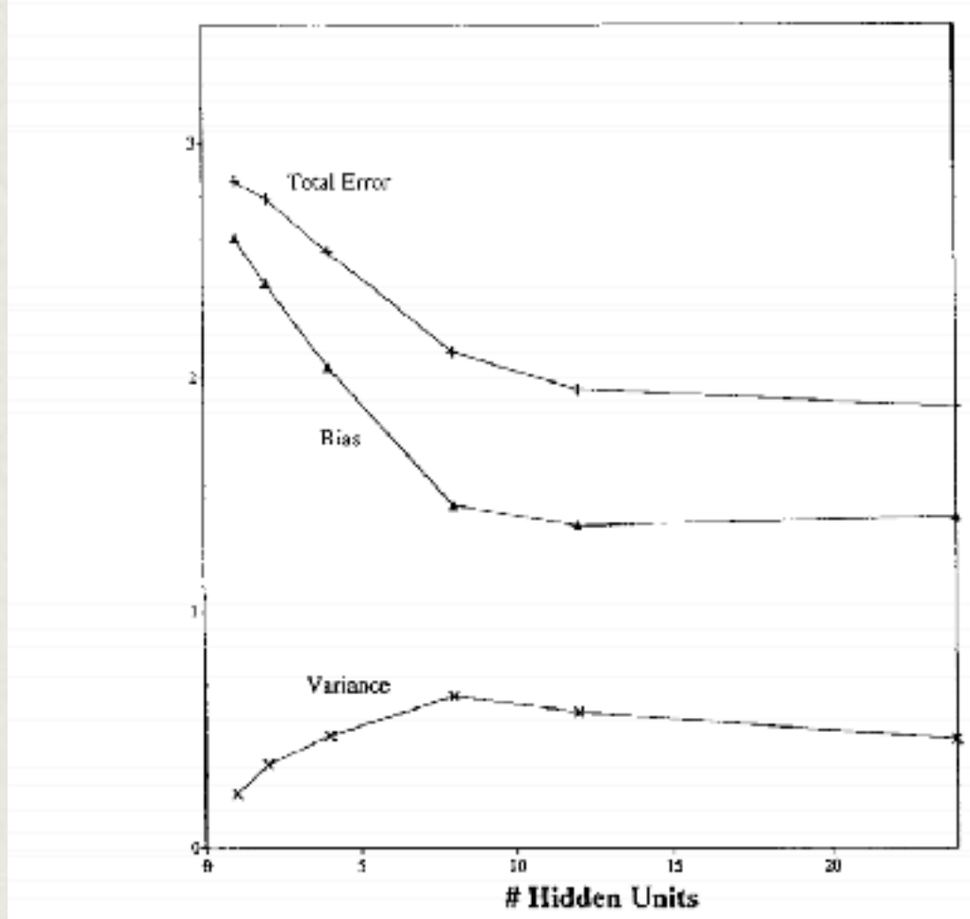
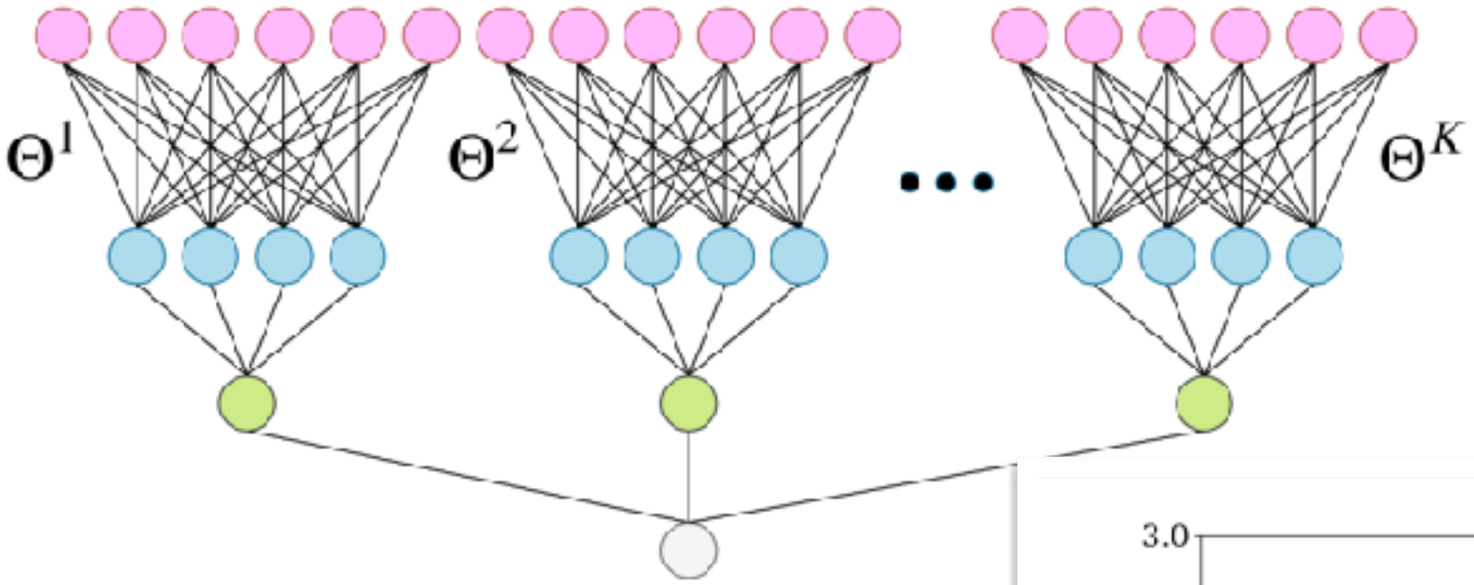


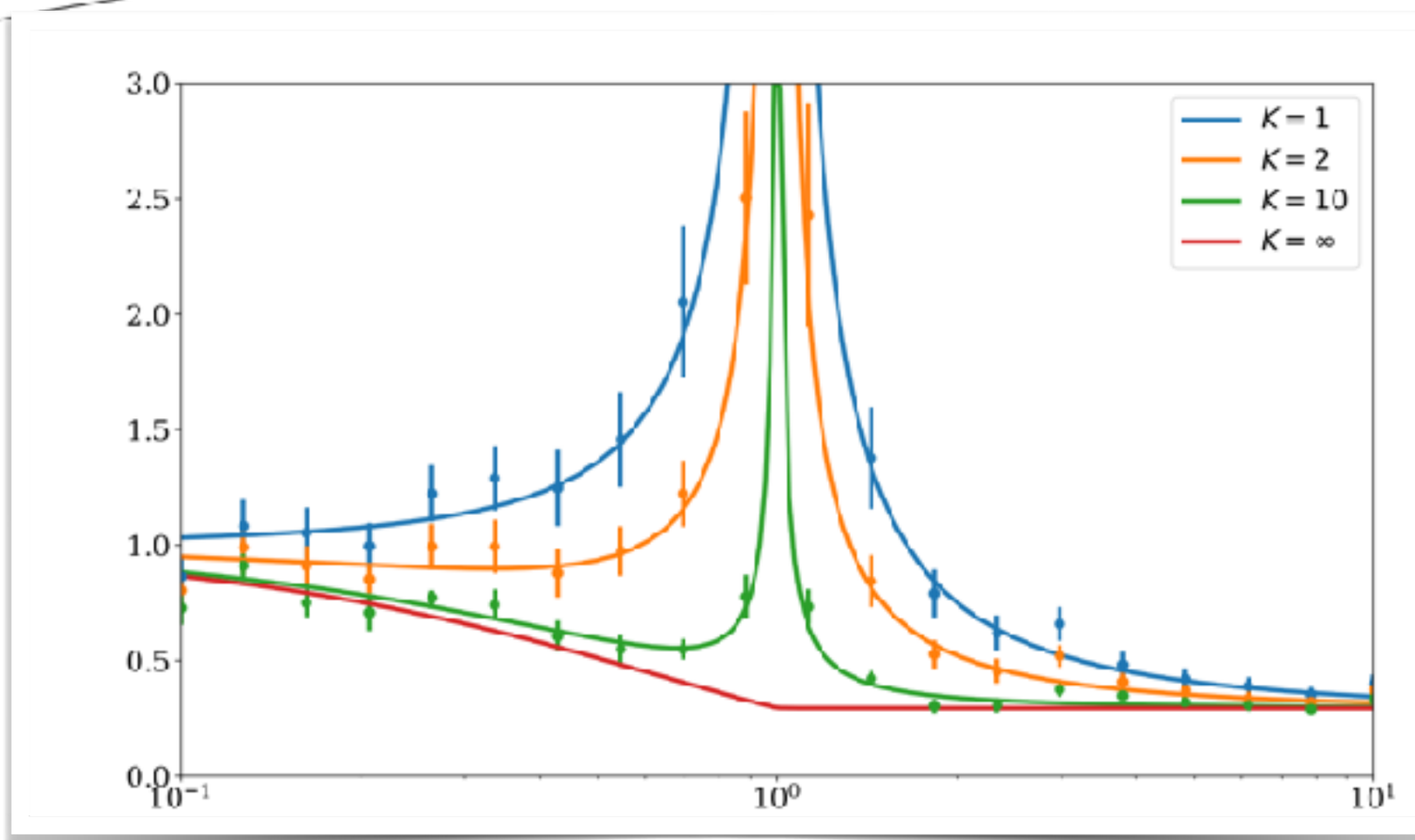
Figure 16: Total error, bias, and variance of feedforward neural network as a function of the number of hidden units. Training is by error backpropagation. For a fixed number of hidden units, the number of iterations of the backpropagation algorithm is chosen to minimize total error.

Reducing the variances by ensembling

Averaging many two-layer networks
with different random features should reduce the variance!



$$f(x) = \frac{1}{K} \sum_{k=1}^{K} f_{\Theta^k}(x)$$



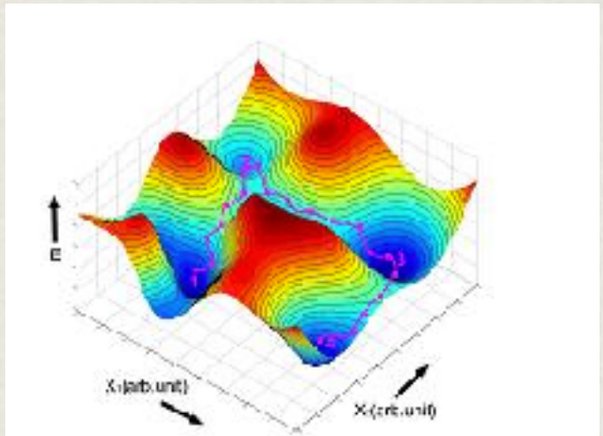
[See also Spigler, Geiger, Ascoli, Sagun, Biroli, Wyart, ’19]

3

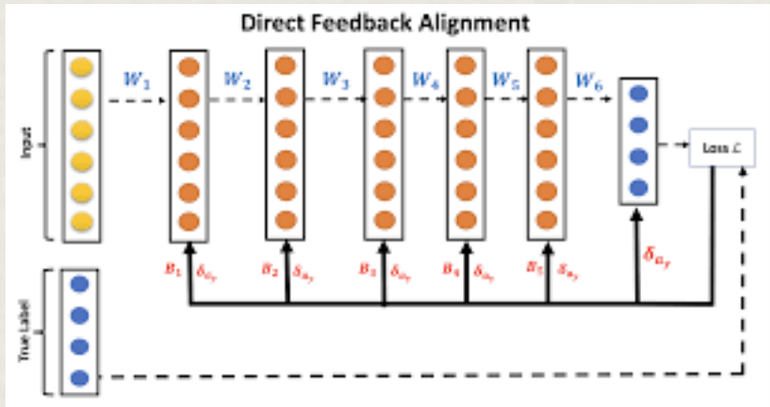
Stochastic Gradient Descent

MANY DIRECTIONS EXPLORED IN MY GROUP

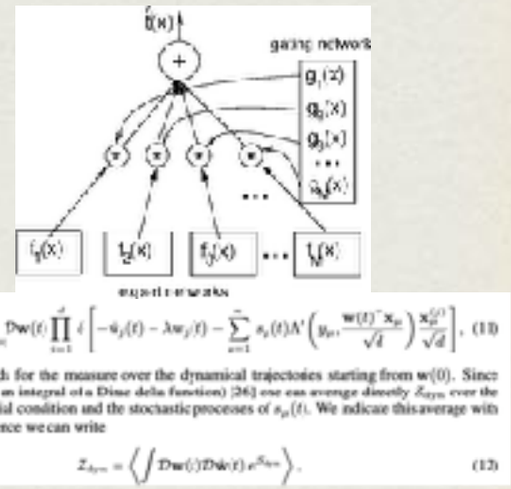
Study of energy landscape

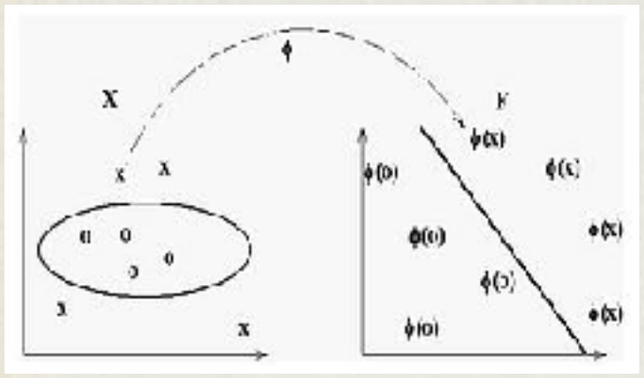


Alternative to back-propagation



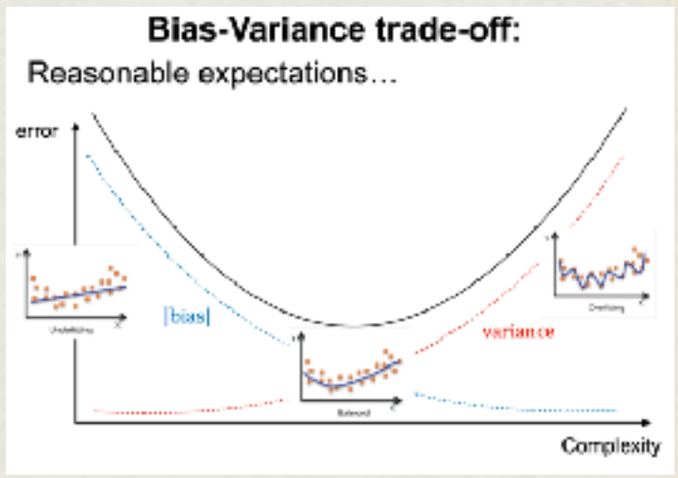
Dynamics of learning in NN





Kernel vs Neural nets

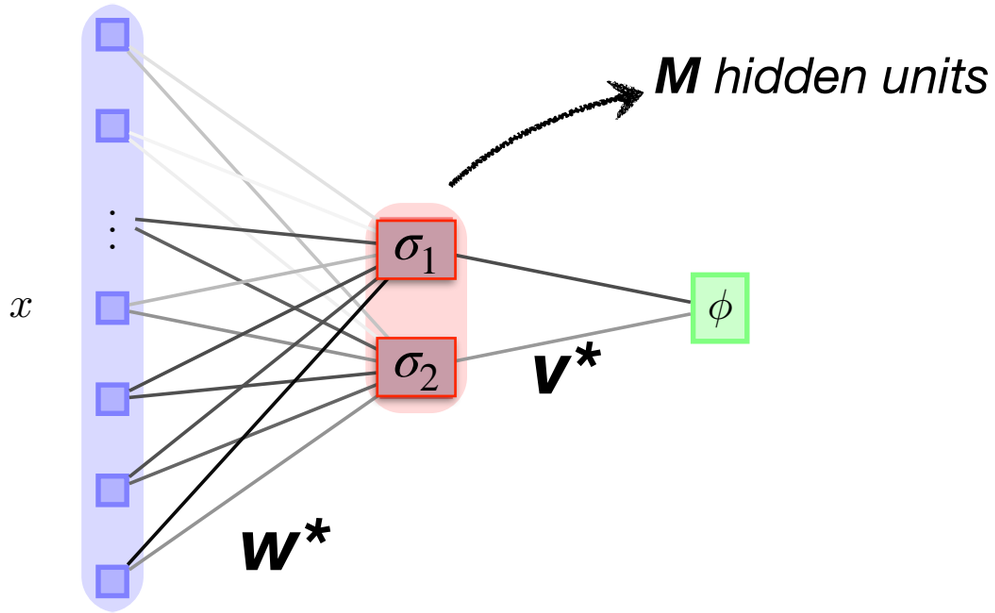
Rigorous approach to replica method



Bias-Variance trade-off

Two-layers teacher-student problem

The teacher generates a dataset…



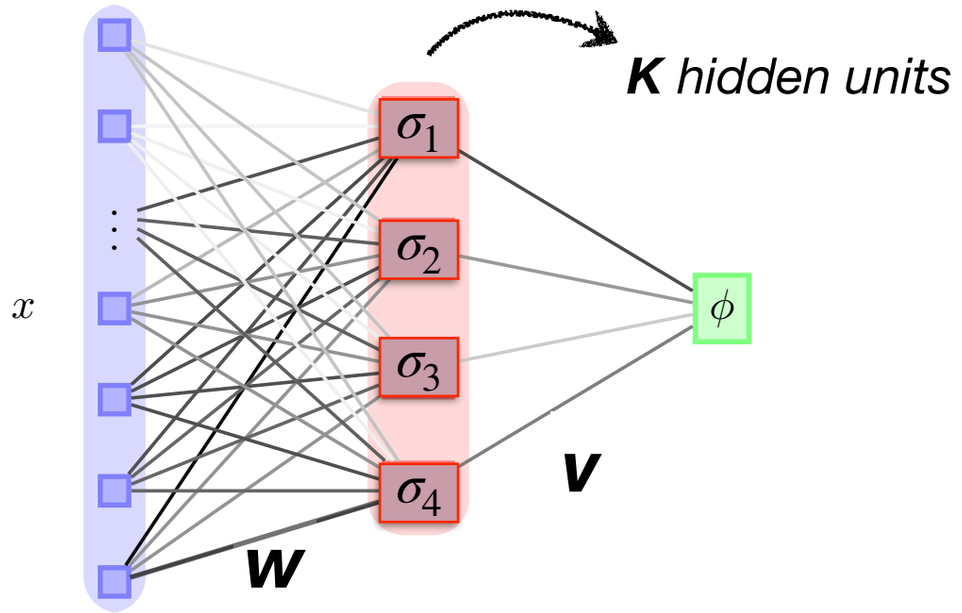
Inputs $\mathbf{x}^\mu \in \mathbb{R}^d$ are i.i.d. Gaussians

Output $$\phi(x, \theta^*) = \sum_m^M v_v^* \sigma\left(\frac{w_m^* x}{\sqrt{d}}\right)$$

Label $$y^\mu \equiv \phi(x^\mu, \theta^*) + \sigma\zeta^\mu$$

Additive output noise

… and the student learns from it



Trained by SGD on the quadratic error:

$$E(\theta, x) = \frac{1}{2}(\phi(x, \theta) - y)^2$$

$d \to \infty$
Large dimensional vector

$M, K = O(1)$
Finite-size hidden layer

On-line gradient descent, one sample at a time…

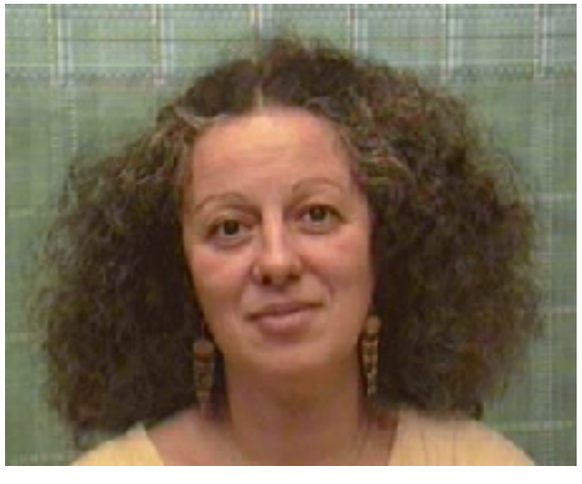
PHYSICAL REVIEW E VOLUME 52, NUMBER 4 OCTOBER 1995

On-line learning in soft committee machines

David Saad[1] and Sara A. Solla[2]
[1]*Department of Physics, University of Edinburgh, King's Buildings, Mayfield Road, Edinburgh EH9 3JZ, United Kingdom*
[2]*CONNECT, The Niels Bohr Institute, Blegdamsdvej 17, Copenhagen 2100, Denmark*
(Received 4 April 1995)

The problem of on-line learning in two-layer neural networks is studied within the framework of statistical mechanics. A fully connected committee machine with K hidden units is trained by gradient descent to perform a task defined by a teacher committee machine with M hidden units acting on randomly drawn inputs. The approach, based on a direct averaging over the activation of the hidden units, results in a set of first-order differential equations that describes the dynamical evolution of the overlaps among the various hidden units and allows for a computation of the generalization e
provide a power
learning scenari
convergence of t

PACS number(s



Dynamics of On-Line Gradient Descent Learning for Multilayer Neural Networks

David Saad*
Dept. of Comp. Sci. & App. Math.
Aston University
Birmingham B4 7ET, UK

Sara A. Solla†
CONNECT, The Niels Bohr Institute
Blegdamsdvej 17
Copenhagen 2100, Denmark

Abstract

We consider the problem of on-line gradient descent learning for general two-layer neural networks. An analytic solution is presented and used to investigate the role of the learning rate in controlling the evolution and convergence of the learning process.

… our rigorous proof just took 25 years (NeurIPS ’19)

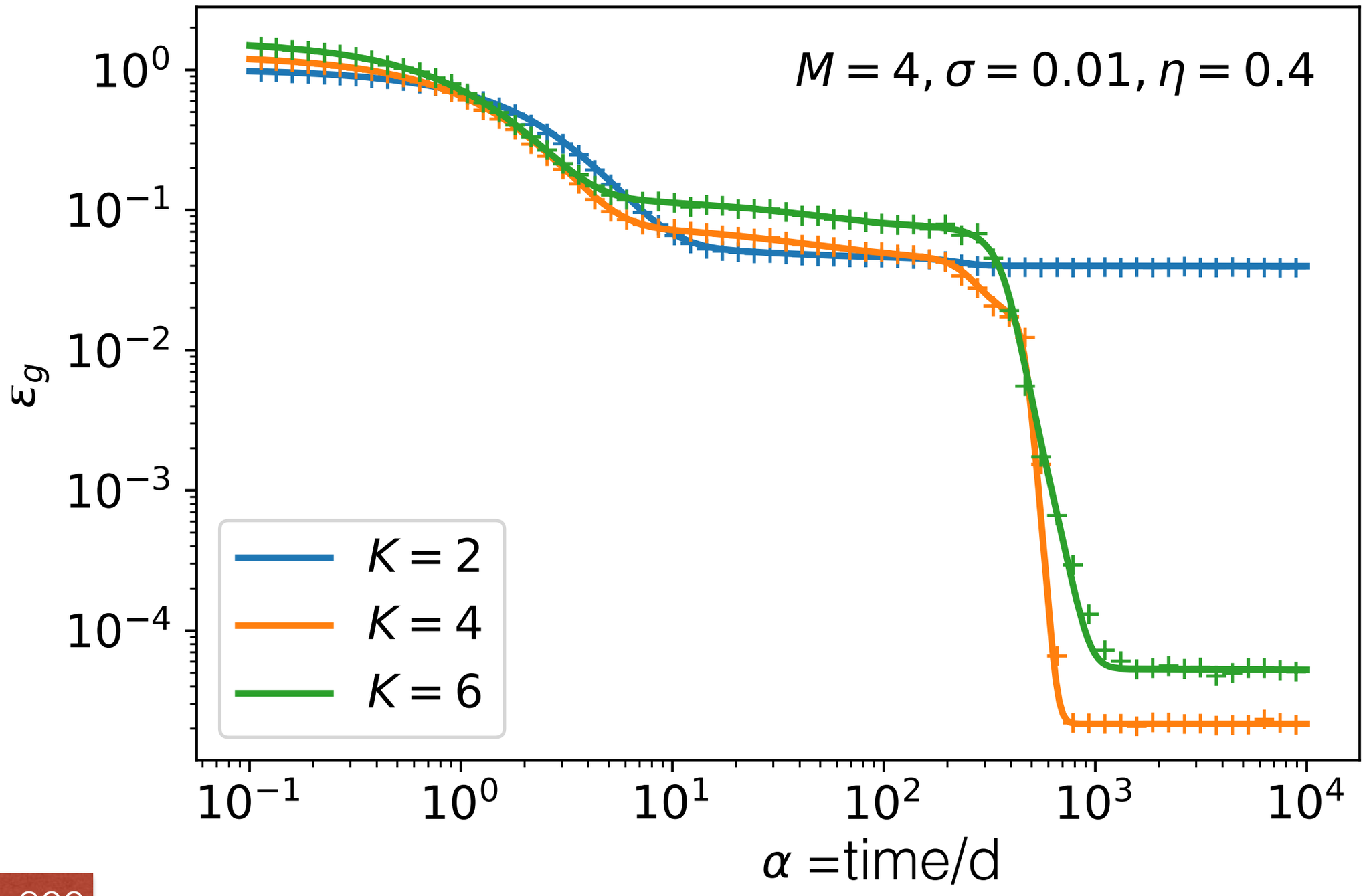
Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup

Sebastian Goldt, Madhu S. Advani, Andrew M. Saxe, Florent Krzakala, Lenka Zdeborová

(Submitted on 18 Jun 2019 (v1), last revised 27 Oct 2019 (this version, v2))

Deep neural networks achieve stellar generalisation even when they have enough parameters to easily fit all their training data. We study this phenomenon by analysing the dynamics and the performance of over-parameterised two-layer neural networks in the teacher-student setup, where one network, the student, is trained on data generated by another network, called the teacher. We show how the dynamics of stochastic gradient descent (SGD) is captured by a set of differential equations and prove that this description is asymptotically exact in the limit of large inputs. Using this framework, we calculate the final generalisation error of student networks that have more parameters than their teachers. We find that the final generalisation error of the student increases with network size when training only the first layer, but stays constant or even decreases with size when training both layers. We show that these different behaviours have their root in the different solutions SGD finds for different activation functions. Our results indicate that achieving good generalisation in neural networks goes beyond the properties of SGD alone and depends on the interplay of at least the algorithm, the model architecture, and the data set.

ODEs are accurate even for small dimension



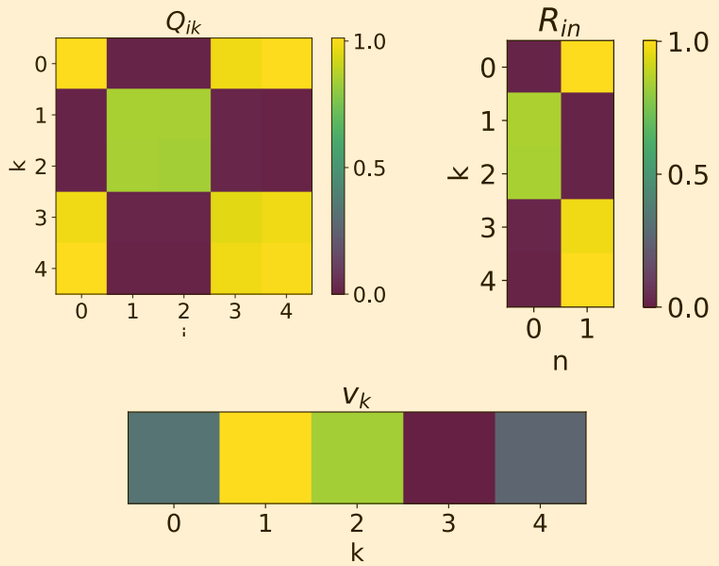
d~800

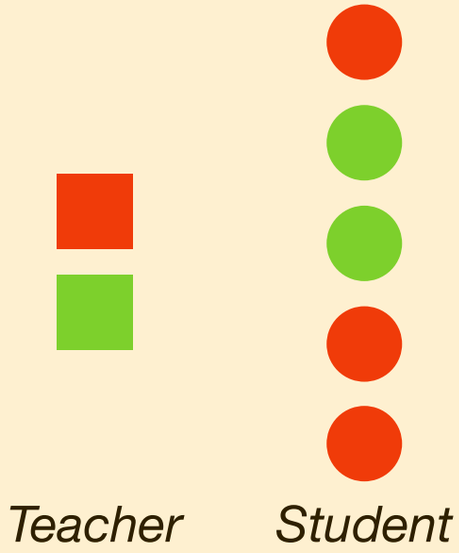
Implicit regularisation of SGD

No overfitting even when K>M !
(Student larger than teacher)

All neurons find relevants features, and the second later average them!
(As in the previous section)

Which neurons learn which neurons an example with M=2, K=5:

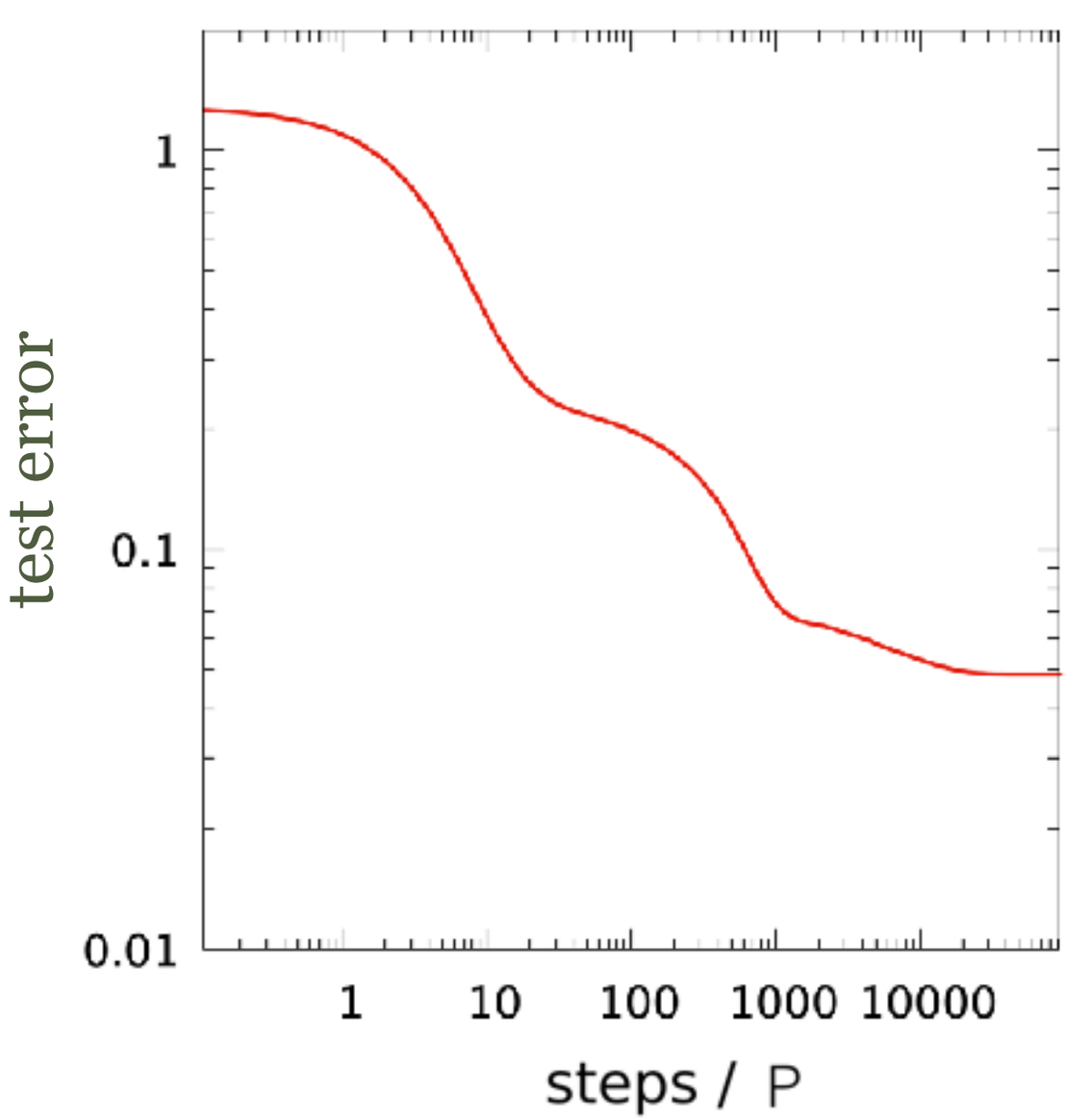


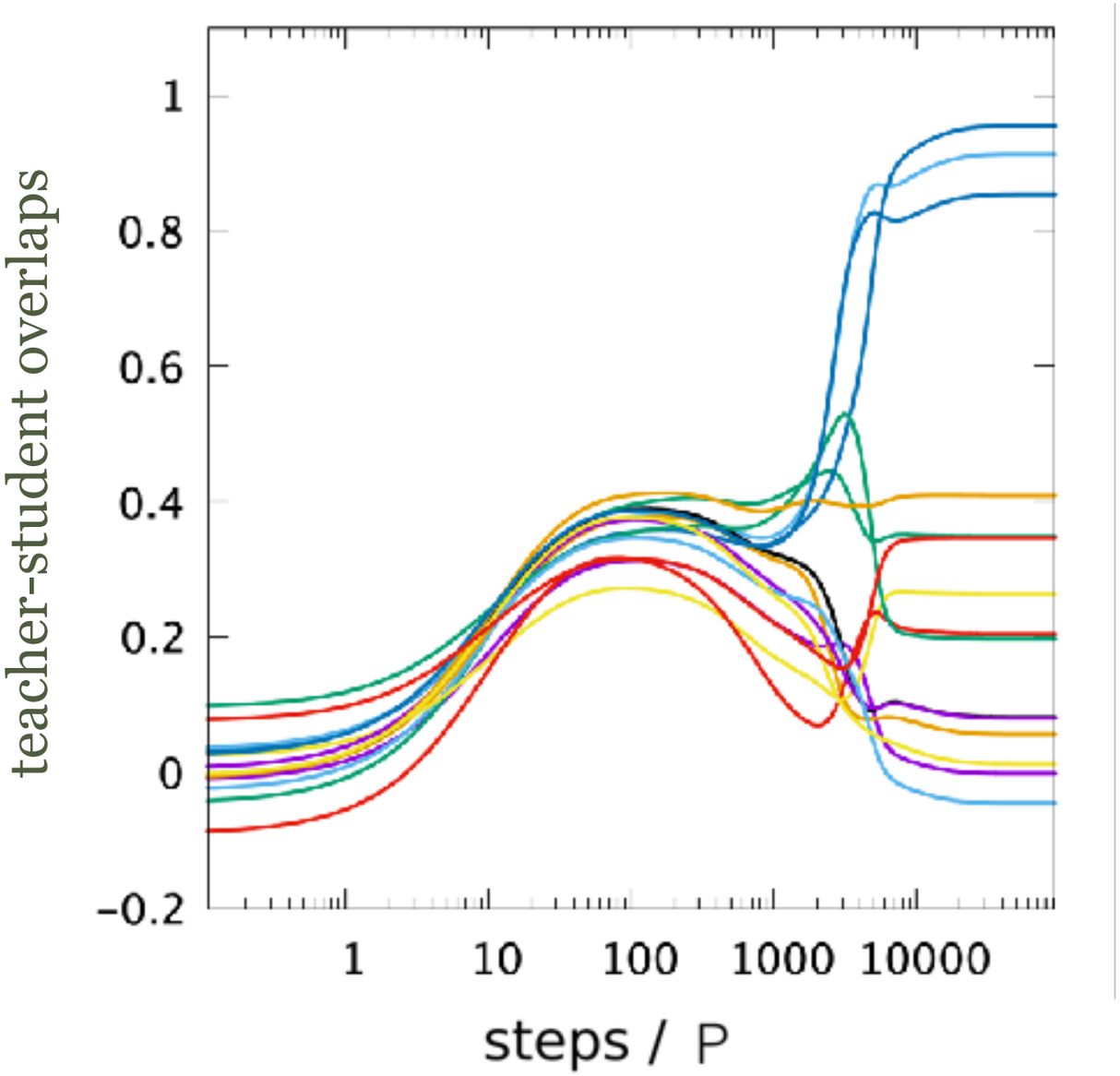


http://github.com/sgoldt/pyscm

SGD learns more complicated functions over time

(Aka “the specialisation phenomenon”)

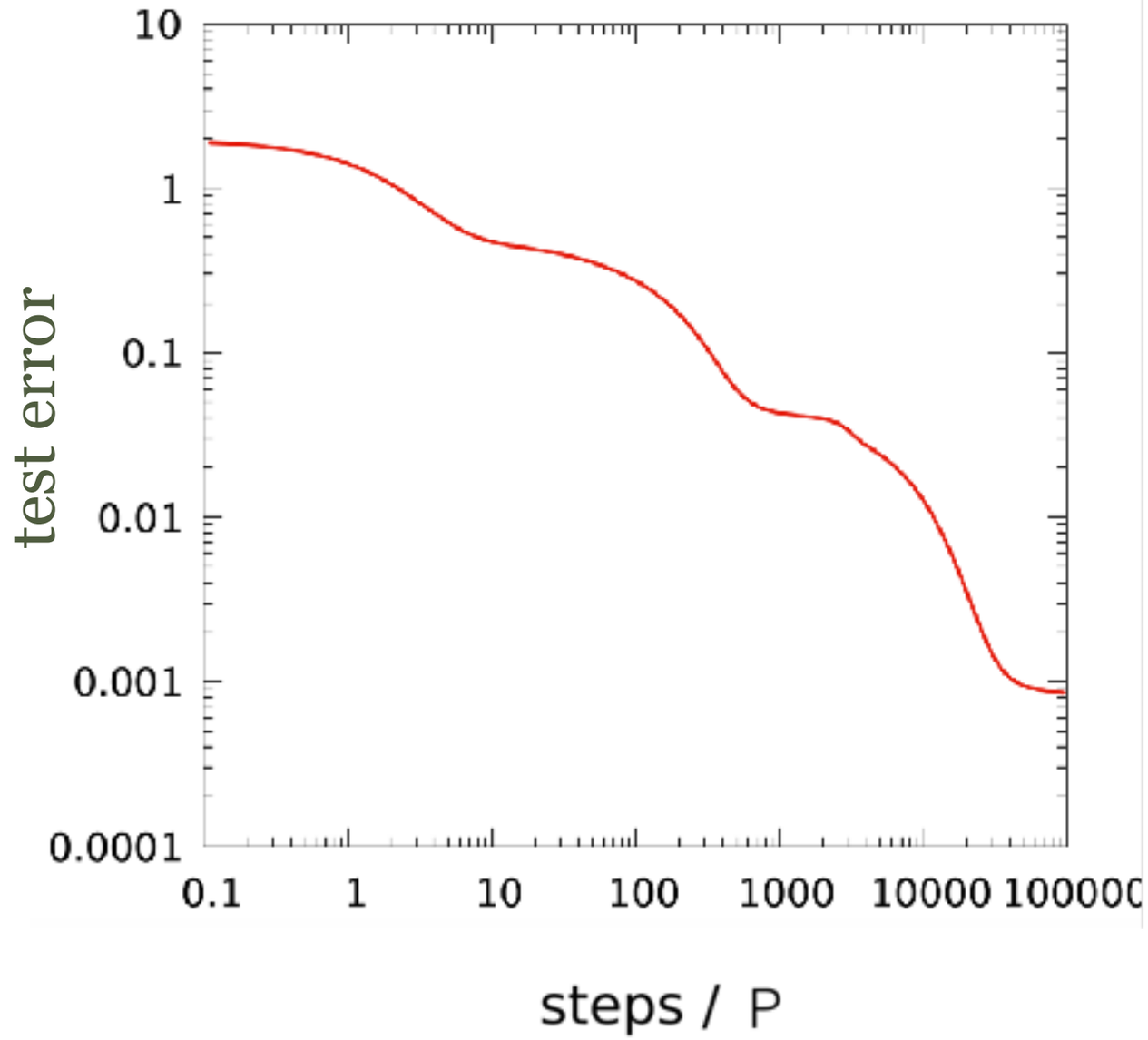


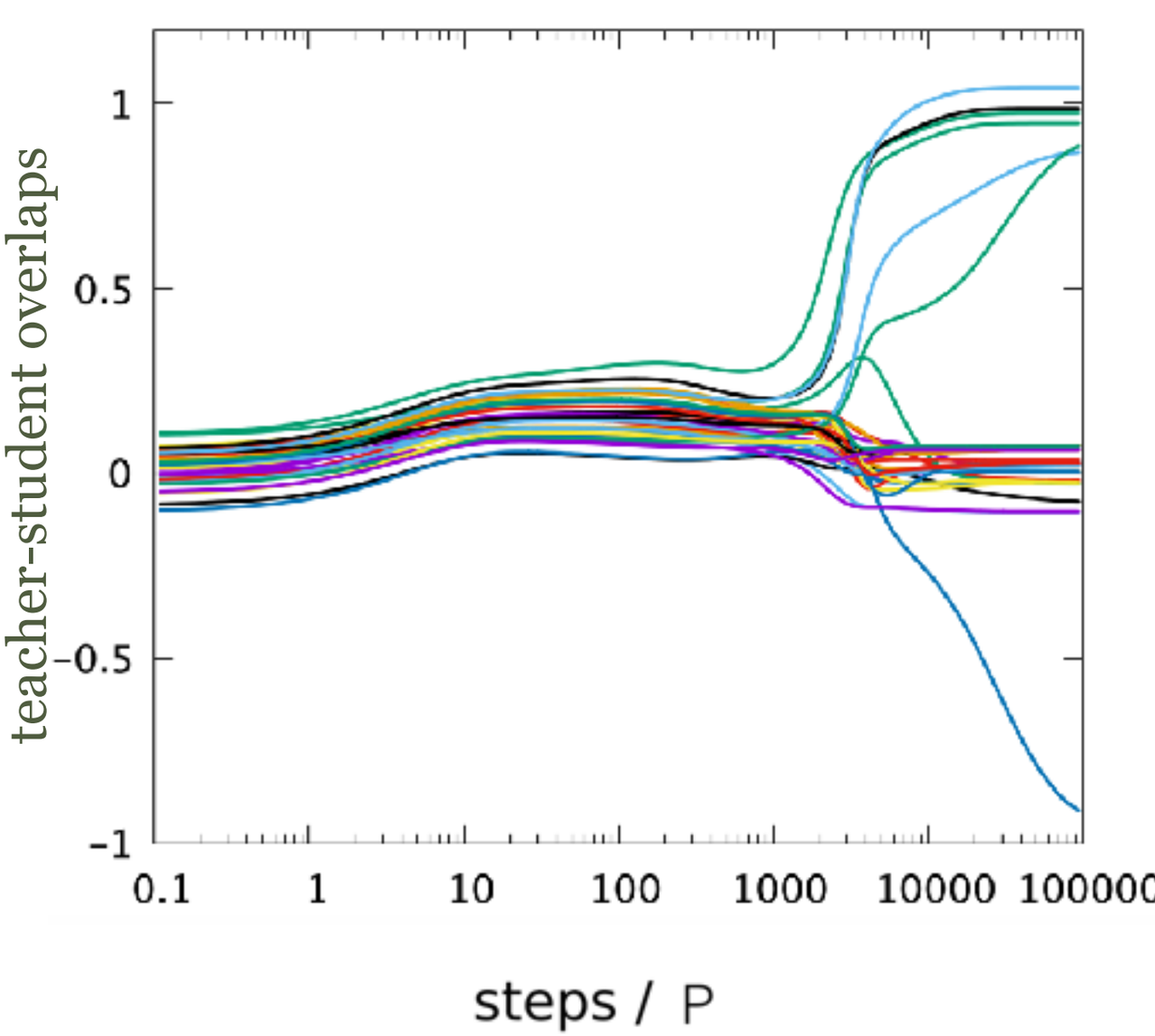


$$f(x) = \text{sign}(x)\,, g(x) = \tilde{g}(x) = \text{erf}(x/\sqrt{2}), M = 5, K = 3, \eta = 0.005$$

SGD learns more complicated functions over time

(Aka “the specialisation phenomenon”)





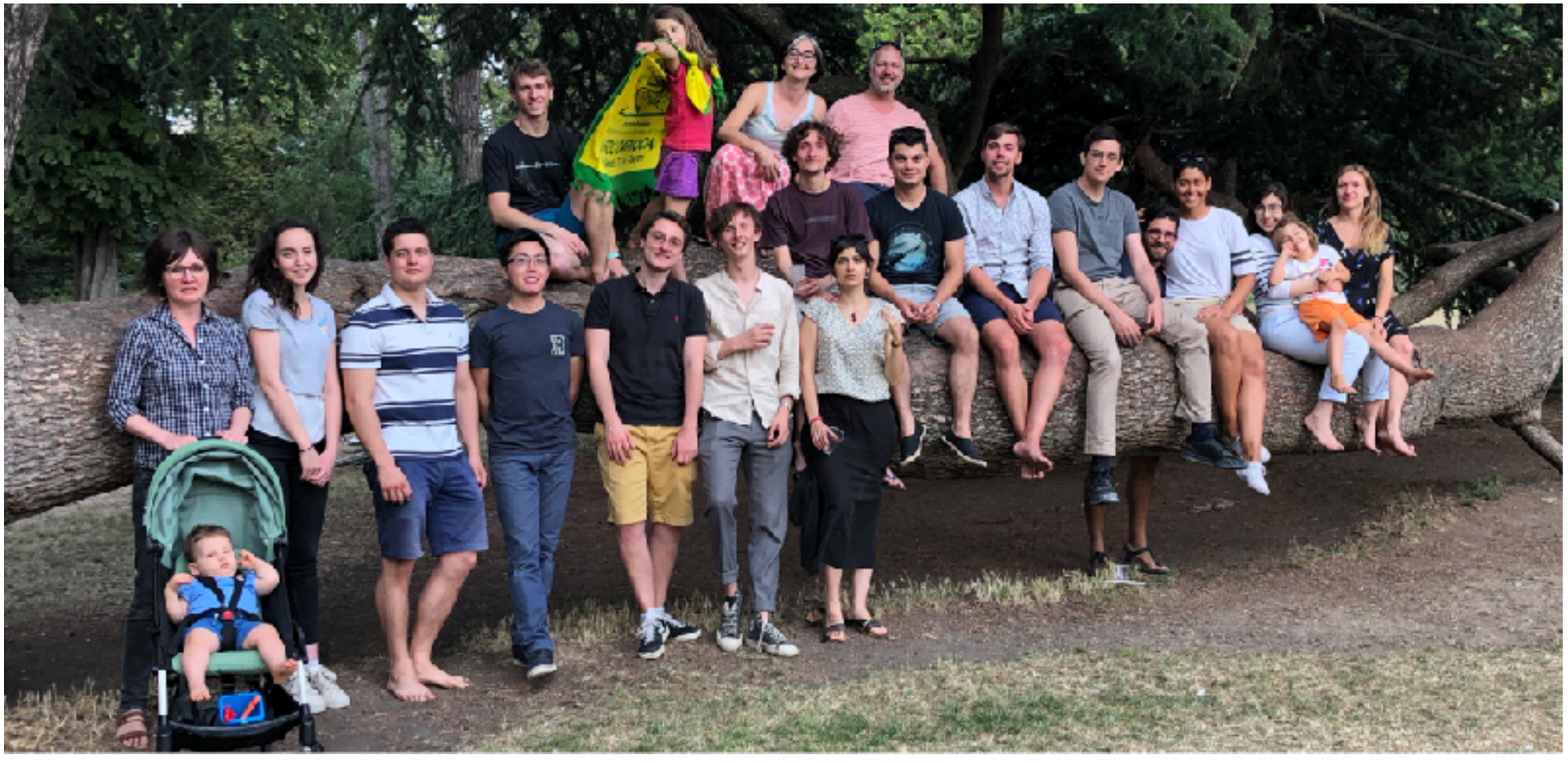
$$f(x) = \text{sign}(x)\,, g(x) = \tilde{g}(x) = \text{erf}(x/\sqrt{2}), M = 5, K = 7, \eta = 0.005$$

Conclusions

Some examples of exactly solvable models of statistical machine learning
(Using rigorous and non-rigorous methods from statistical mechanics)
These models shed lights on many interesting phenomena

- Worst case versus typical
- Implicit regularisation (small l2 norm solutions, large margin)
- Divergence of the generalisation at the *interpolation* peak
- Large 2-layer networks display some “self-averaging” effects:
  Over-paramerization helps in reducing the variance
- Complete & explicit Bias-Variance decompositions

Many thanks to the team(s)....



We want you in Switzerland!

IdEφIX

INFORMATION, IEARNING & PHYSICS LAB.

I just created the IdePHICS lab. in EPFL

We are looking for talented postdocs & students, send me a mail!