Applied Machine Learning Days (EPFL 2022) — *AI & Physics track*

Statistical physics insights on stochastic gradient descent

Francesca Mignacco
Ph.D. student @ Institute of Theoretical Physics, CEA Saclay
Supervised by of **Lenka Zdeborová** & **Pierfrancesco Urbani**

March 28th, 2022

Puzzles & mysteries of deep learning theory…

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?
- When should one stop the backpropagation and use the current parameters?

Puzzles & mysteries of deep learning theory…

… since almost 30 years !

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?
- When should one stop the backpropagation and use the current parameters?

Reflections after refereeing papers for NIPS,
Leo Breiman, **1995**

Puzzles & mysteries of deep learning theory…

… since almost 30 years !

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

1
- 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?
- When should one stop the backpropagation and use the current parameters?

1 require the understanding of **static** properties

Puzzles & mysteries of deep learning theory...

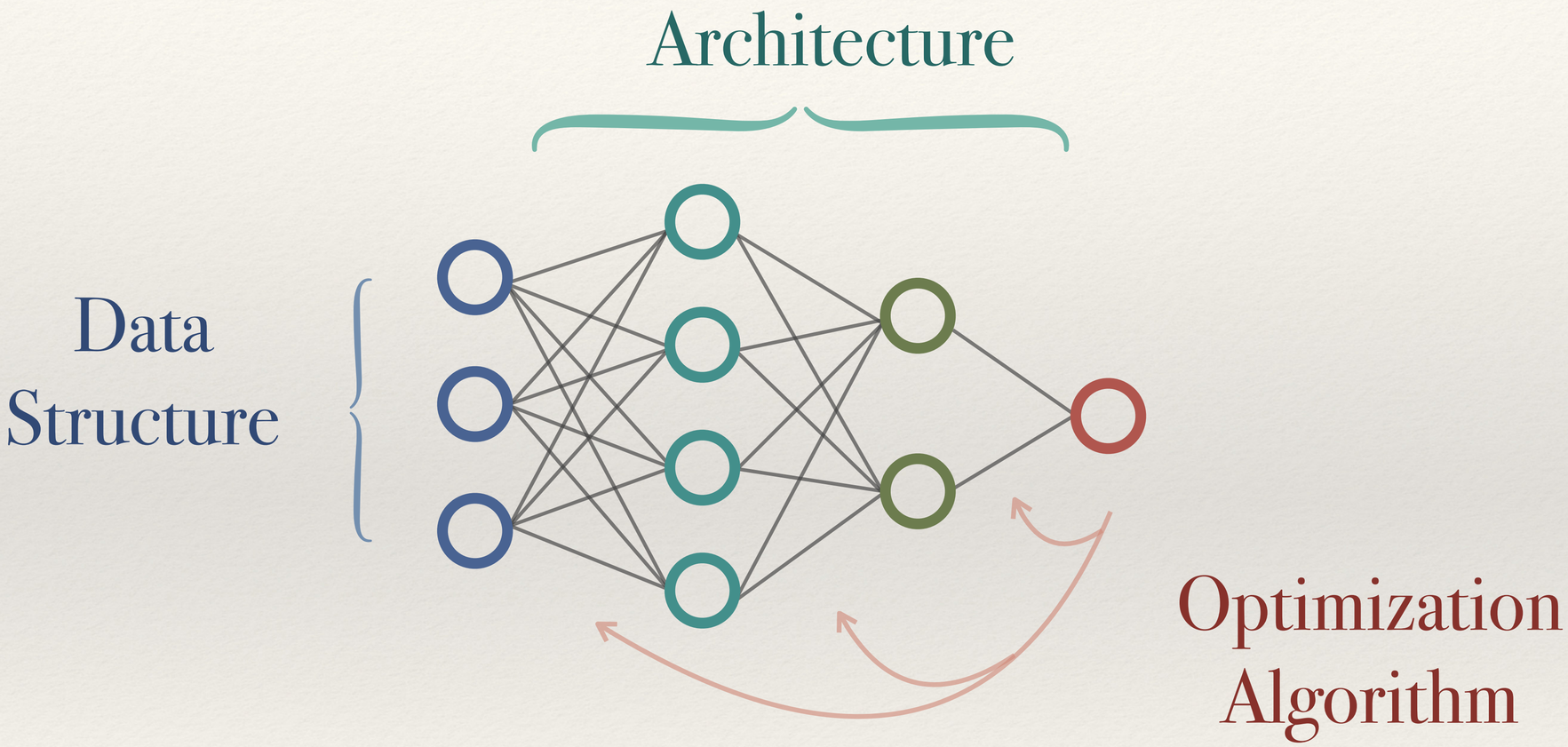
For instance, there are many important questions regarding neural networks which are largely unanswered. There seem to be conflicting stories regarding the following issues:

- Why don't heavily parameterized neural networks overfit the data?
- What is the effective number of parameters?
- 2 Why doesn't backpropagation head for a poor local minima?
- When should one stop the backpropagation and use the current parameters?

1 require the understanding of **static** properties

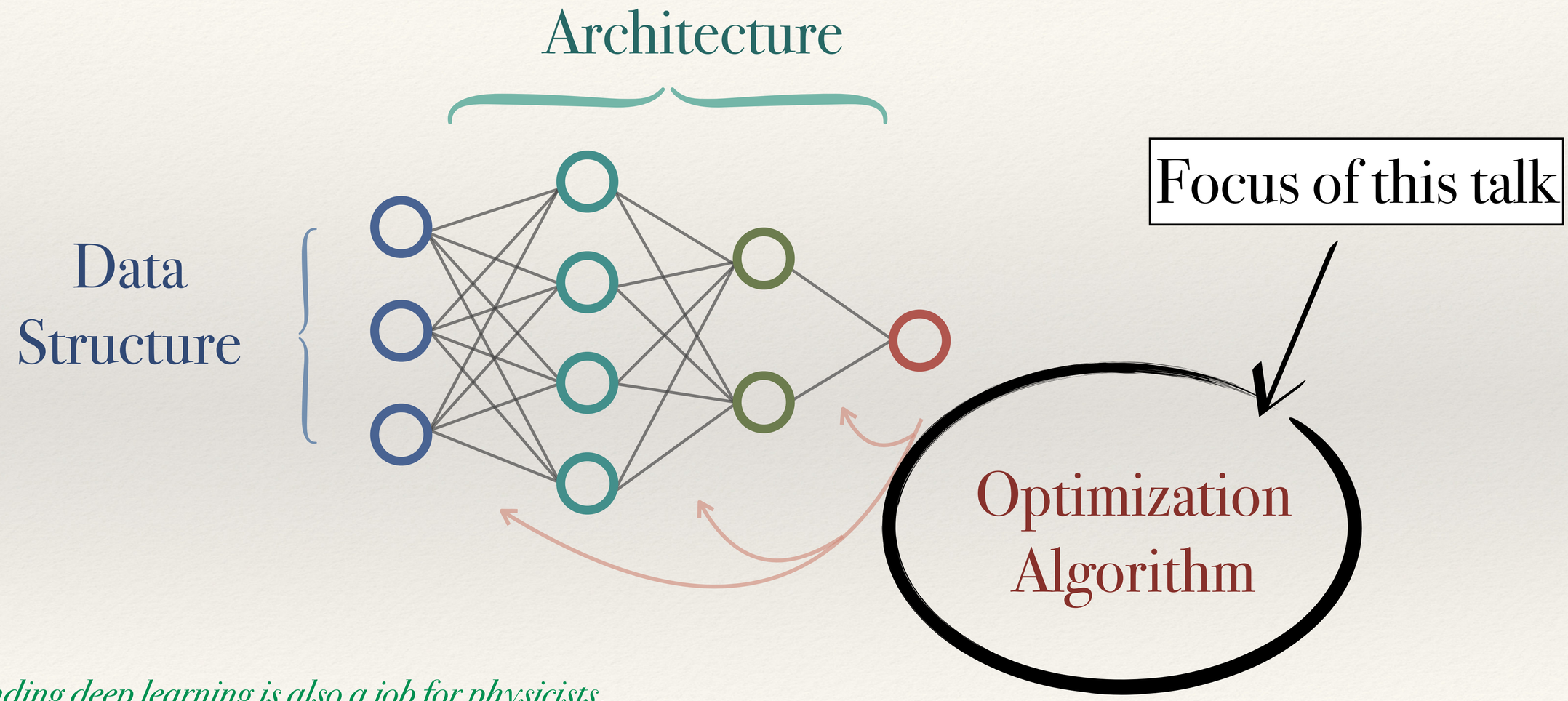
2 require the understanding of **dynamical** properties

The building blocks of deep learning



[*Understanding deep learning is also a job for physicists*, Zdeborová, L. (2020). Nature Physics, 16(6), 602-604]

The building blocks of deep learning



[*Understanding deep learning is also a job for physicists*, Zdeborová, L. (2020). Nature Physics, 16(6), 602-604]

Machine learning *à la physicist*: typical-case results

<u>The problem:</u>

Given some training data and labels $\left\{ \mathbf{x}_\mu, y_\mu \right\}_{\mu=1}^{M}$ sampled independently from $p_{x,y}$

Machine learning *à la physicist*: typical-case results

<u>The problem:</u>

Given some training data and labels $\left\{ \mathbf{x}_\mu, y_\mu \right\}_{\mu=1}^{M}$ sampled independently from $p_{x,y}$

Learn the predictor $\hat{y} = f_{\hat{\mathbf{w}}}(\mathbf{x})$ by minimising the empirical risk:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}}\, \mathscr{L}(\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmin}} \left[\sum_{\mu=1}^{M} \ell\left(y_\mu, f_{\mathbf{w}}(\mathbf{x}_\mu) \right) + r(\mathbf{w}) \right]$$

Machine learning *à la physicist*: typical-case results

The problem:

Given some training data and labels $\left\{ \mathbf{x}_\mu, y_\mu \right\}_{\mu=1}^{M}$ sampled independently from $p_{x,y}$

Learn the predictor $\hat{y} = f_{\hat{\mathbf{w}}}(\mathbf{x})$ by minimising the empirical risk:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}}\, \mathscr{L}(\mathbf{w}) = \underset{\mathbf{w}}{\operatorname{argmin}} \left[\sum_{\mu=1}^{M} \ell\left(y_\mu, f_{\mathbf{w}}(\mathbf{x}_\mu) \right) + r(\mathbf{w}) \right]$$

How can we characterize the **predictor's error**?

$$\varepsilon_{\text{gen}} = \mathbb{E}\left[\left(y_{\text{new}} - f_{\hat{\mathbf{w}}}(\mathbf{x}_{\text{new}}) \right)^2 \right]$$

$$\varepsilon_{\text{train}} = \frac{1}{M} \sum_{\mu=1}^{M} \left(y_\mu - f_{\hat{\mathbf{w}}}(\mathbf{x}_\mu) \right)^2$$

Machine learning *à la physicist*: typical-case results

<u>Data model:</u>

$$\mathbf{x}_\mu \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}_N\right)$$

$$y_\mu = \mathrm{sign}\left(\mathbf{x}_\mu^\top \mathbf{w}^*\right)$$

Machine learning *à la physicist*: typical-case results

Data model:

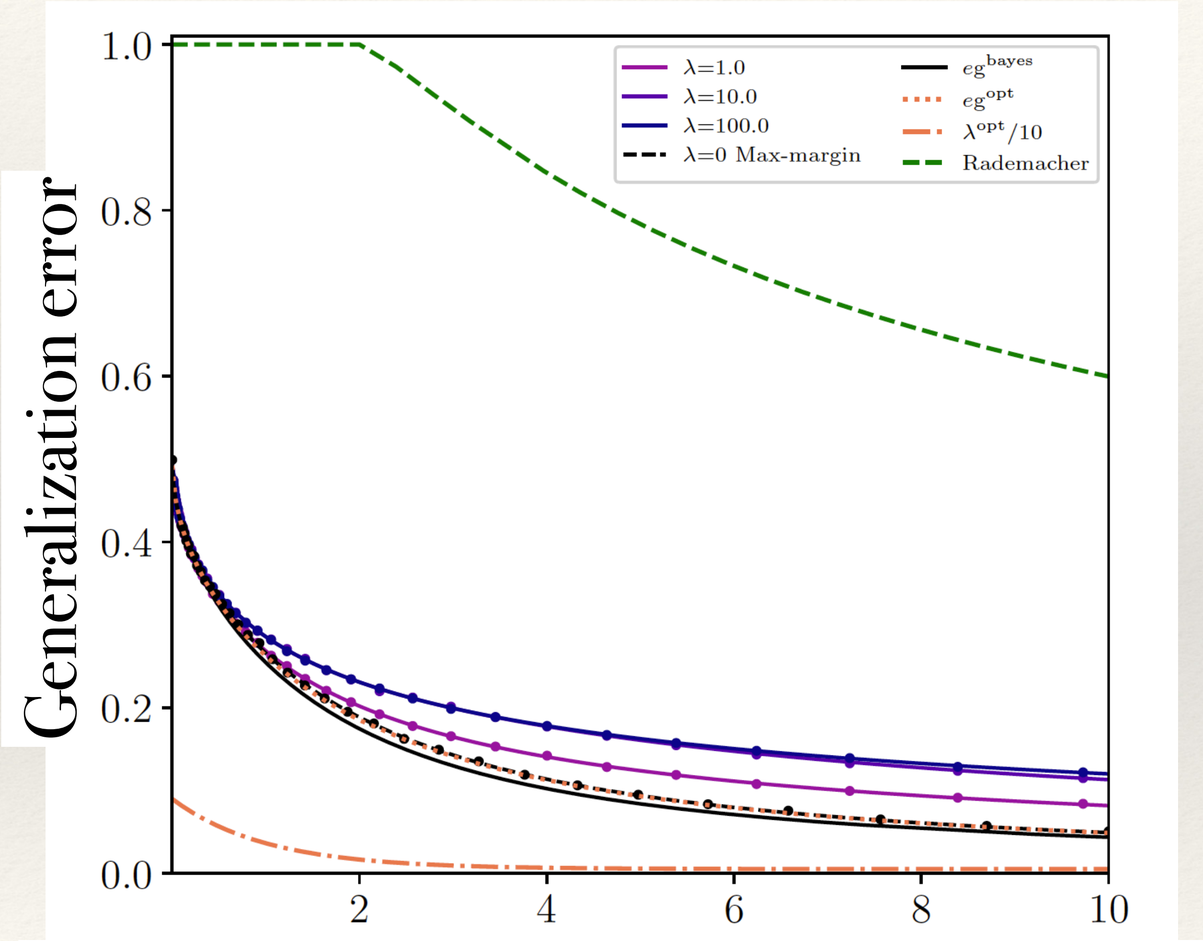
$$\mathbf{x}_\mu \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}_N\right)$$

$$y_\mu = \text{sign}\left(\mathbf{x}_\mu^\top \mathbf{w}^*\right)$$

Teacher-student model:

The labels are generated from a **teacher vector**.

Machine learning *à la physicist*: typical-case results



Sample complexity = # samples / # dimensions

Data model:

$$\mathbf{x}_\mu \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}_N\right)$$

$$y_\mu = \text{sign}\left(\mathbf{x}_\mu^\top \mathbf{w}^*\right)$$

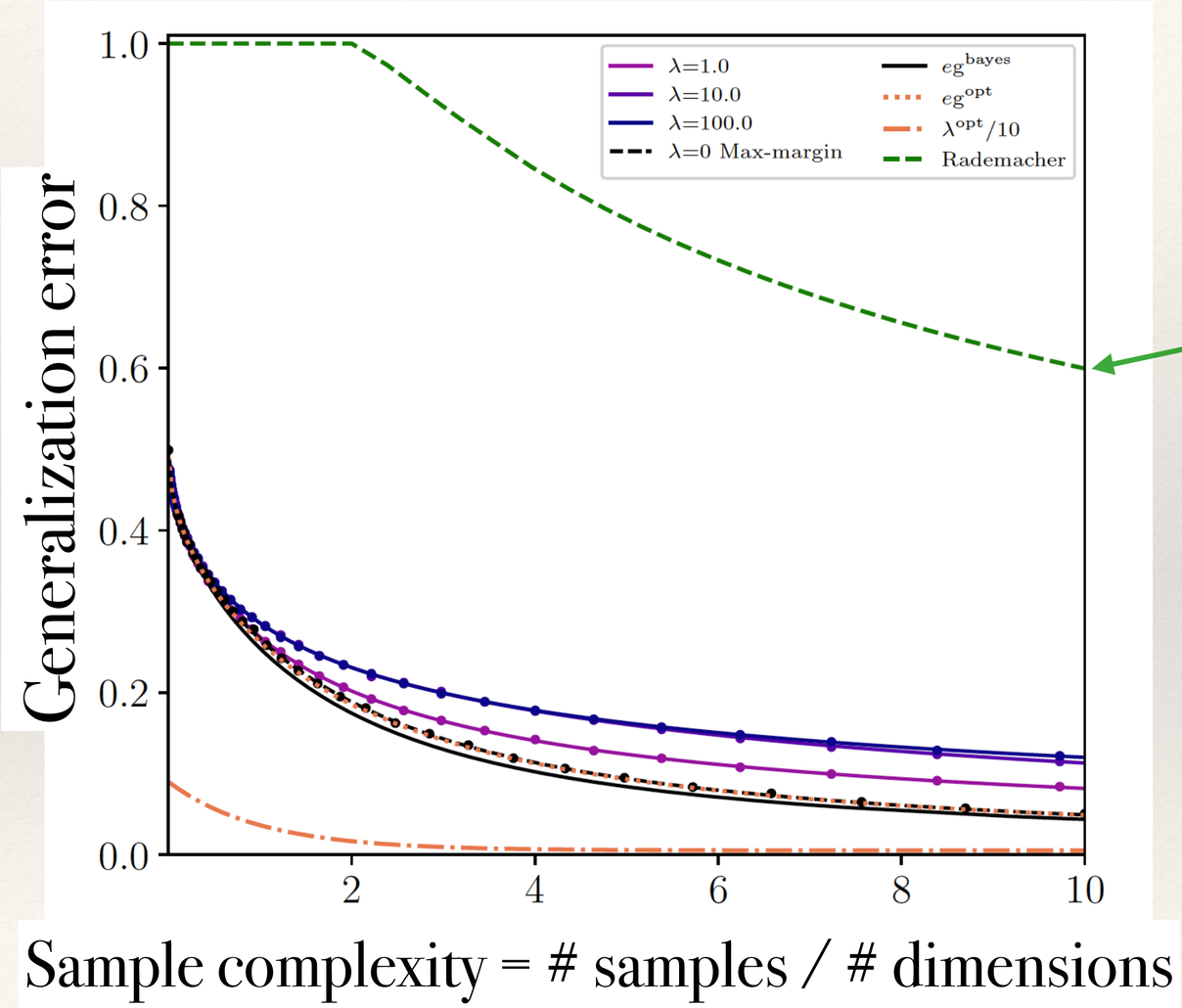
Teacher-student model:

The labels are generated from a **teacher vector**.

[Abbara, Aubin, Krzakala, Zdeborová, *MSML* 2020]

[Aubin, Krzakala, Lu, Zdeborová, *NeurIPS* 2020]

Machine learning *à la physicist*: typical-case results



Data model:

$$\mathbf{x}_\mu \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}_N\right)$$

$$y_\mu = \text{sign}\left(\mathbf{x}_\mu^\top \mathbf{w}^*\right)$$

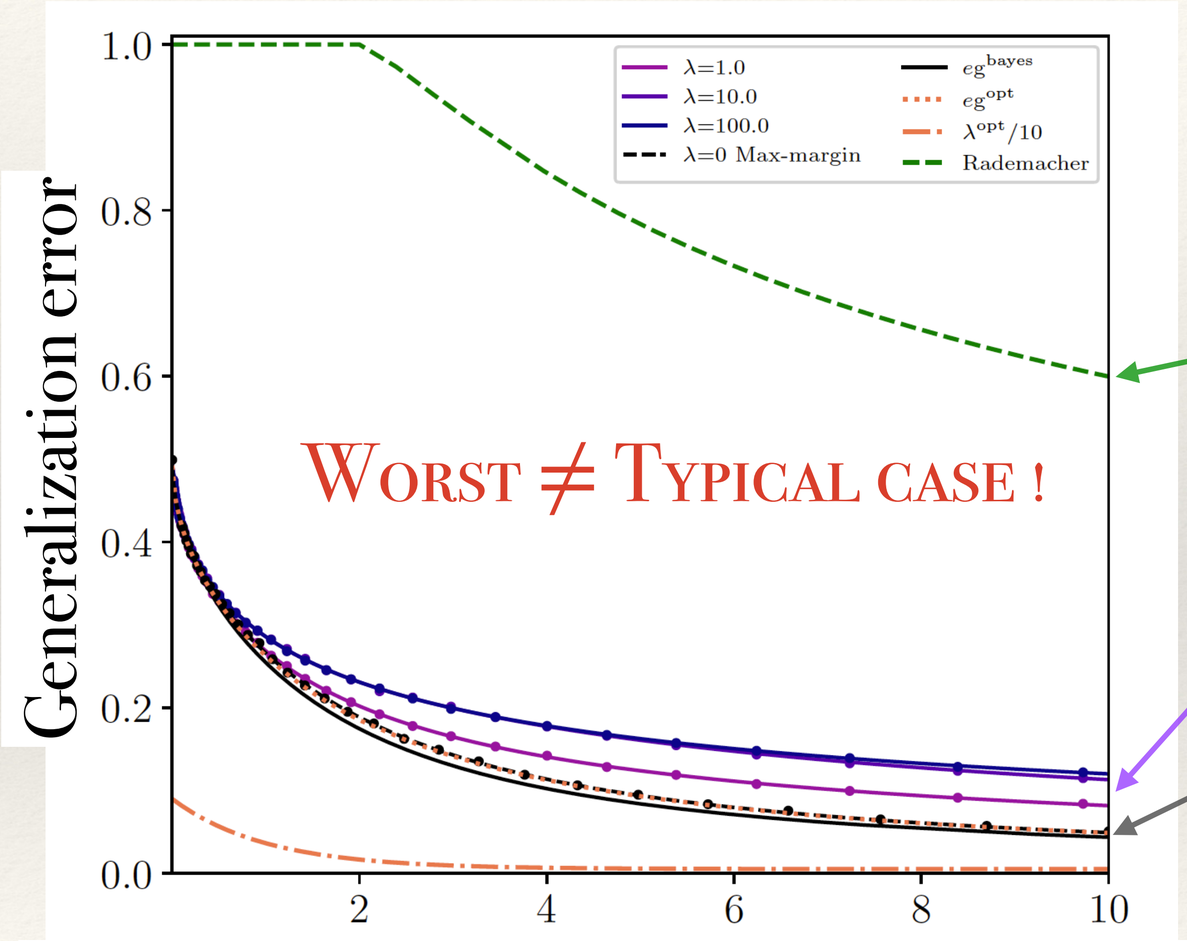
Rademacher bound for function class:

$$f_{\mathbf{w}}(\mathbf{x}) = \text{sign}\left(\mathbf{x}^\top \mathbf{w}\right)$$

[Abbara, Aubin, Krzakala, Zdeborová, *MSML* 2020]

[Aubin, Krzakala, Lu, Zdeborová, *NeurIPS* 2020]

Machine learning *à la physicist*: typical-case results



Sample complexity = # samples / # dimensions

Data model:

$$\mathbf{x}_\mu \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}_N\right)$$

$$y_\mu = \text{sign}\left(\mathbf{x}_\mu^\top \mathbf{w}^*\right)$$

Rademacher bound for function class:
$f_{\mathbf{w}}(\mathbf{x}) = \text{sign}\left(\mathbf{x}^\top \mathbf{w}\right)$

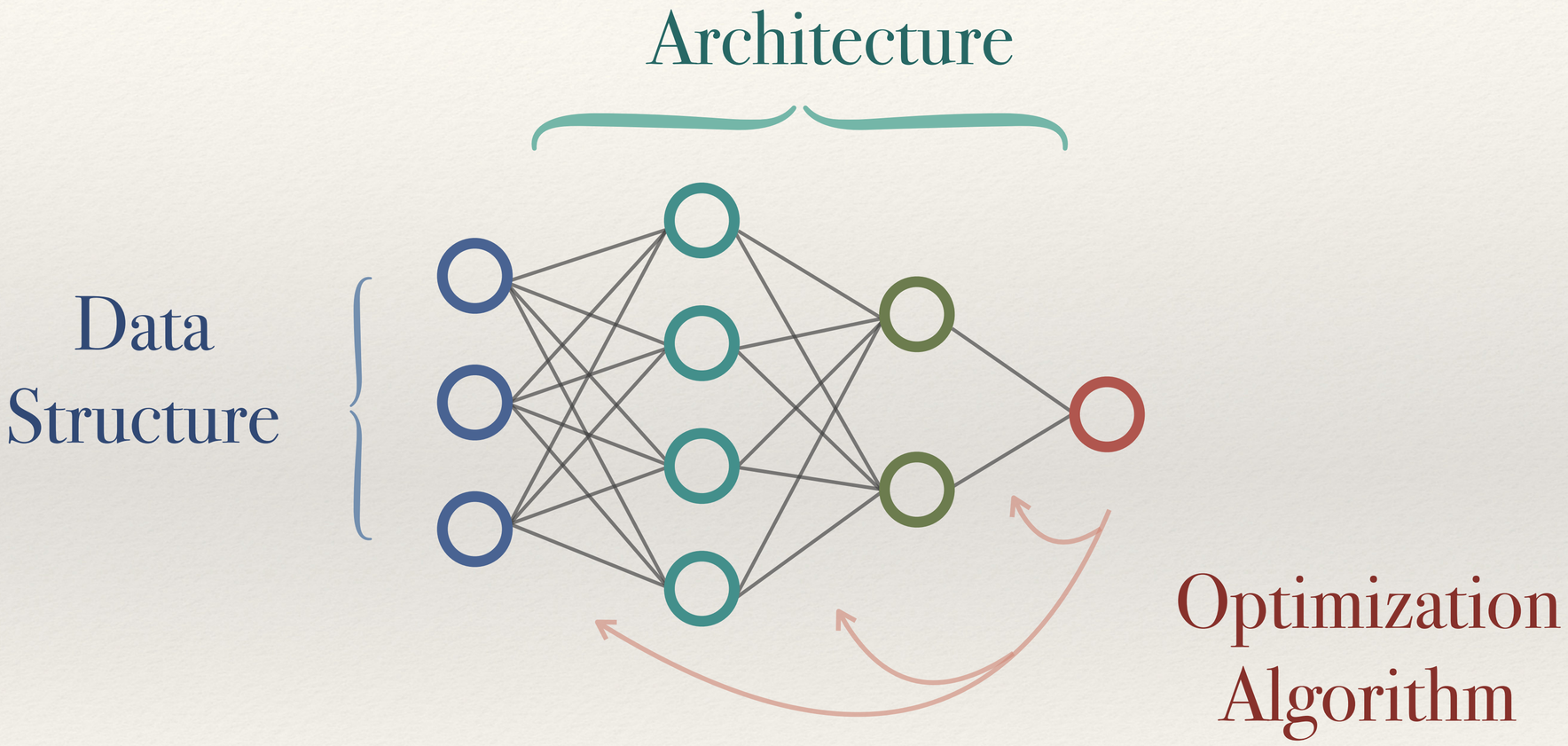
Logistic regression

Bayes-optimal performance

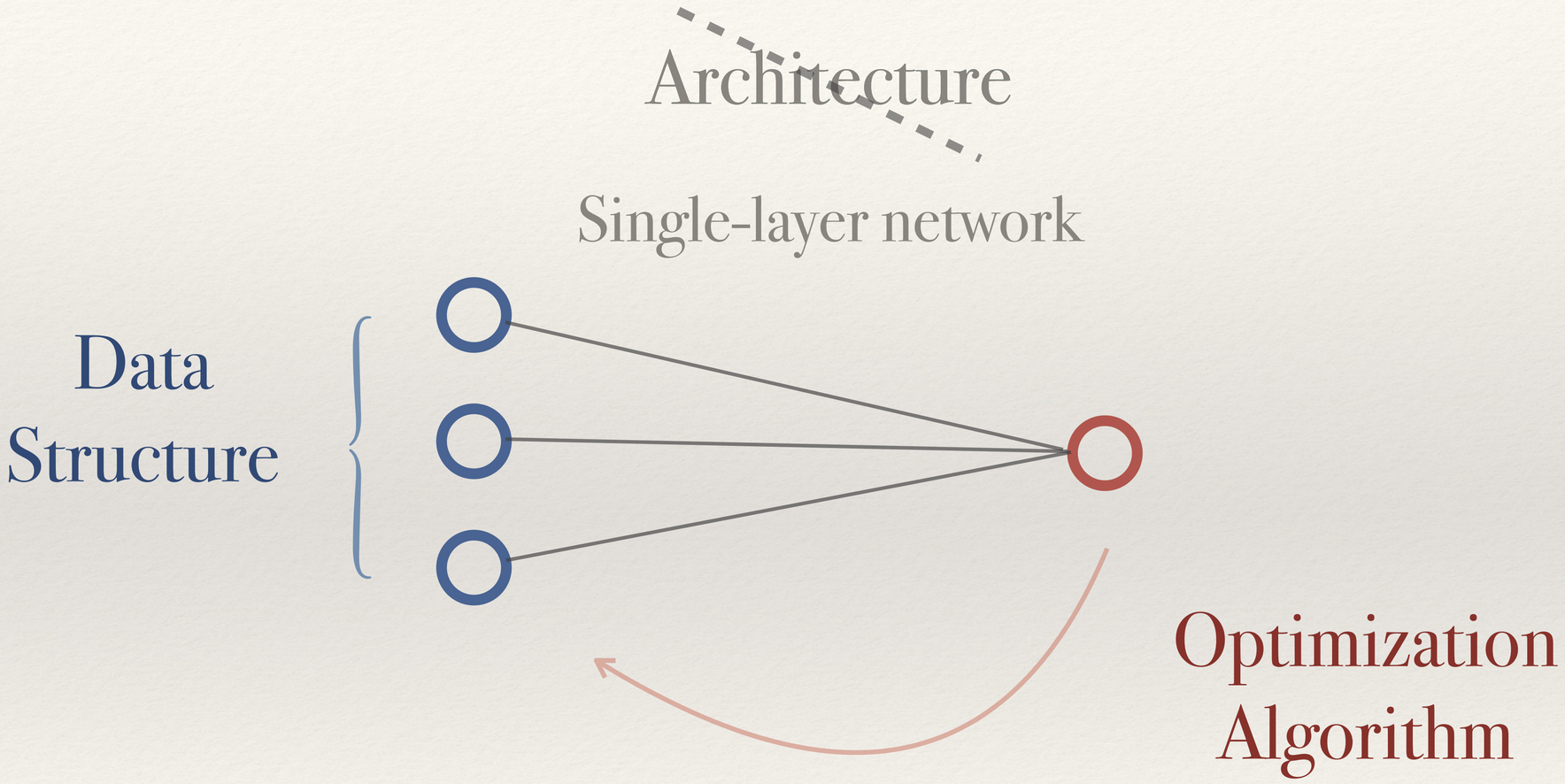
[Abbara, Aubin, Krzakala, Zdeborová, *MSML* 2020]

[Aubin, Krzakala, Lu, Zdeborová, *NeurIPS* 2020]

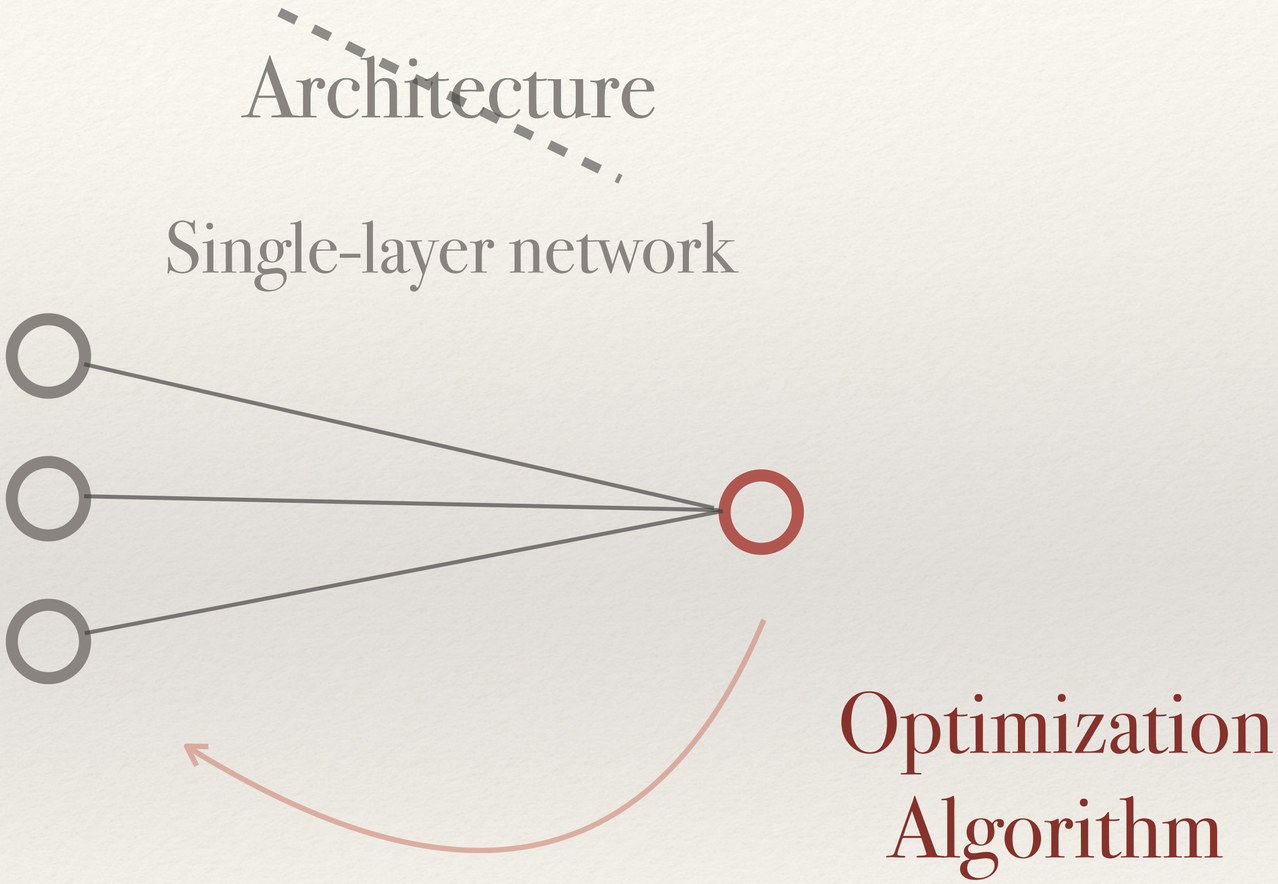
Optimization algorithm: the dynamics of learning



Optimization algorithm: the dynamics of learning



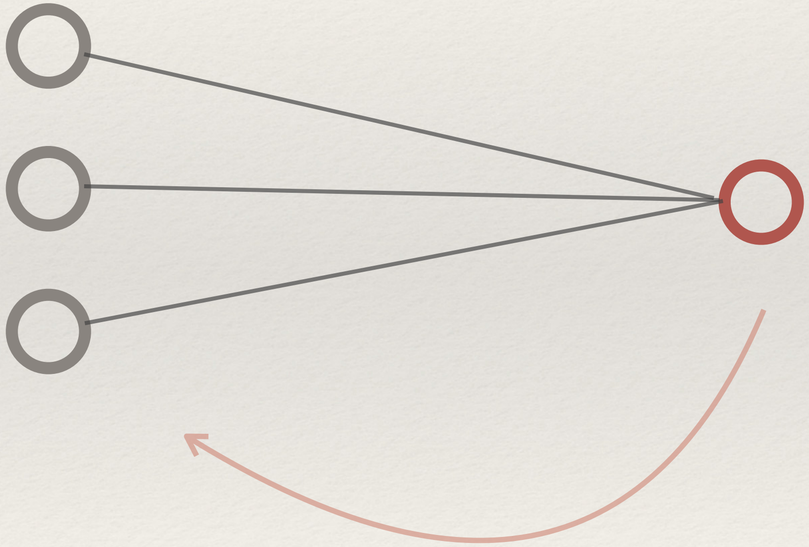
Optimization algorithm: the dynamics of learning



Optimization algorithm: the dynamics of learning

Architecture

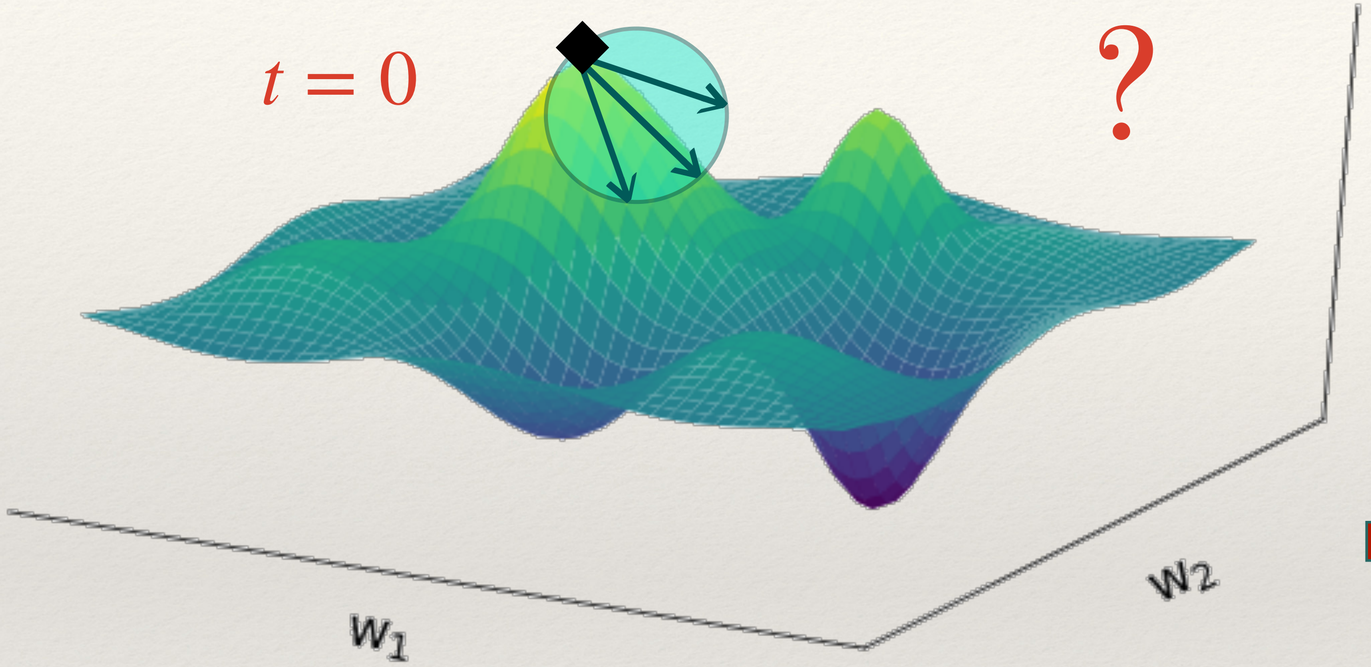
Single-layer network

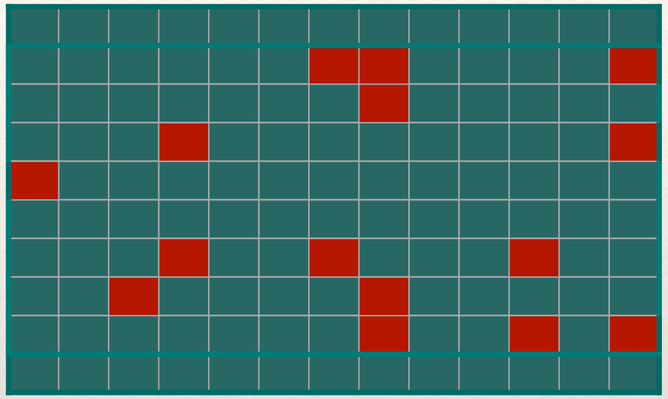


… yet many open questions !

Optimization Algorithm

The interplay of landscape & algorithmic noise





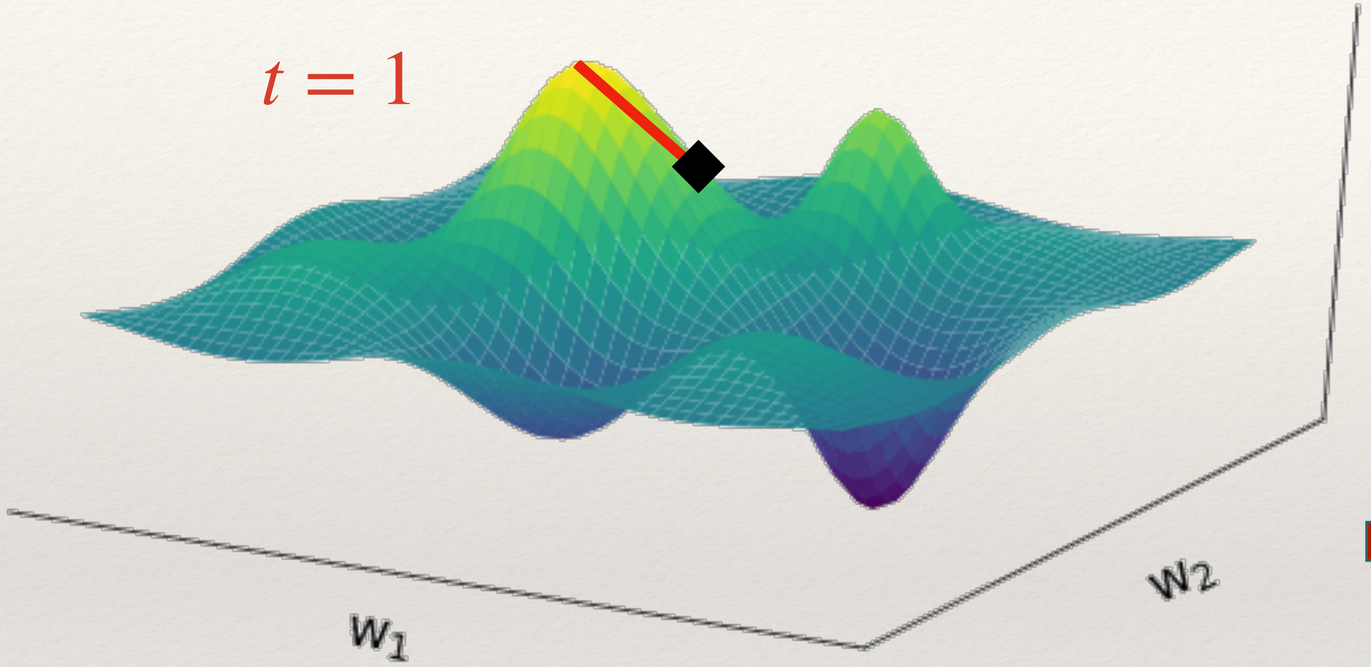
$\mathscr{B}(t = 0)$

MINI BATCH

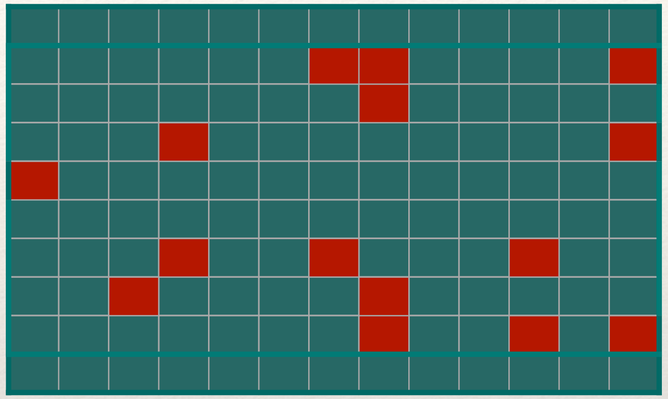
= samples used for training at time t

WEIGHTS UPDATE: $$\mathbf{w}^{t+1} = \mathbf{w}^{t} - \mathrm{d}t \left[\sum_{\mu \in \mathscr{B}(t)} \nabla_{\mathbf{w}} \ell \left(y_\mu, f_{\mathbf{w}(t)}(\mathbf{x}_\mu) \right) + \nabla_{\mathbf{w}} r \left(\mathbf{w}(t) \right) \right]$$

The interplay of landscape & algorithmic noise



TRAINING SET



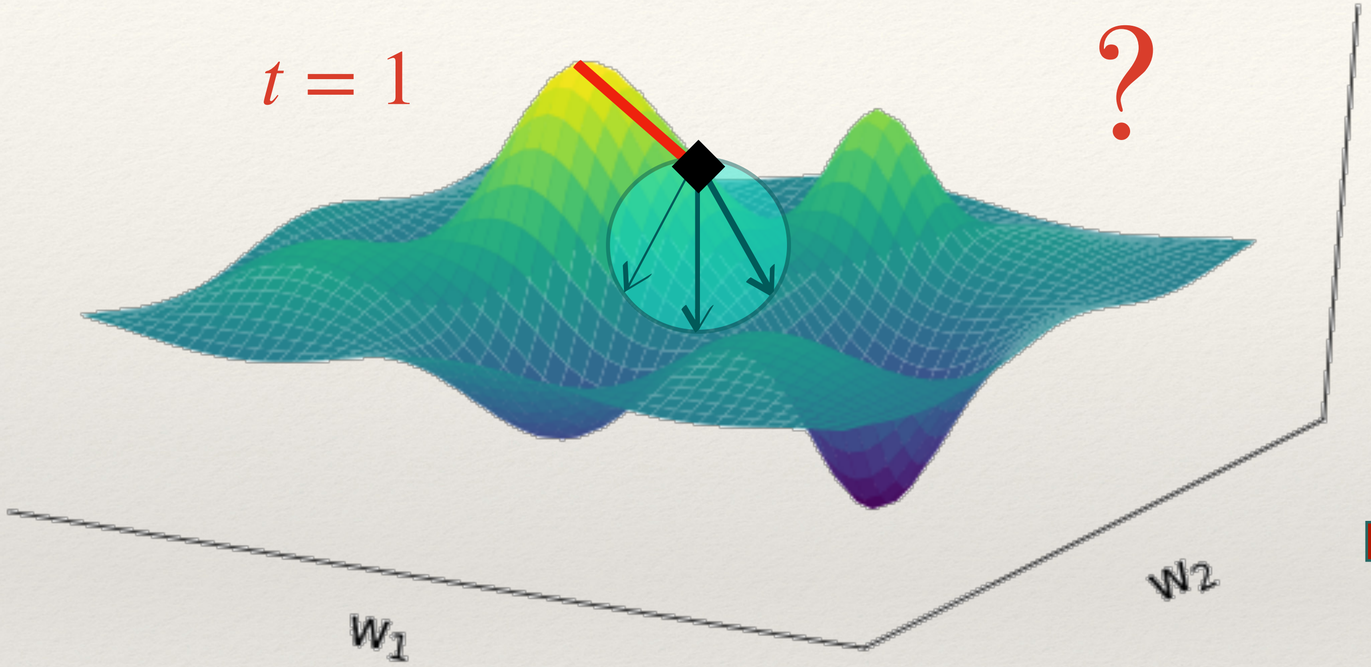
$\mathscr{B}(t = 0)$

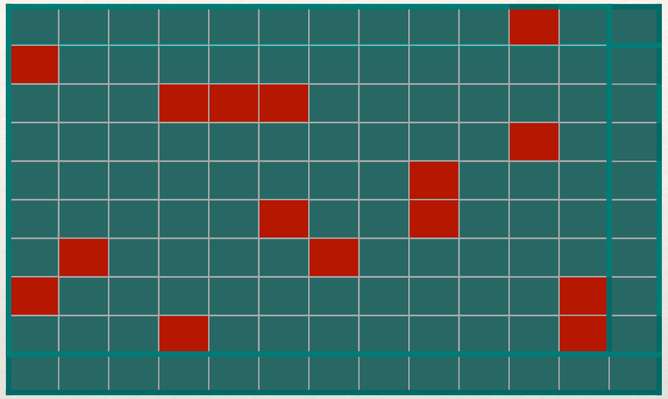
MINI BATCH

= samples used for training at time t

WEIGHTS UPDATE: $$\mathbf{w}^{t+1} = \mathbf{w}^t - \mathrm{d}t \left[\sum_{\mu \in \mathscr{B}(t)} \nabla_{\mathbf{w}} \ell \left(y_\mu, f_{\mathbf{w}(t)}(\mathbf{x}_\mu) \right) + \nabla_{\mathbf{w}} r \left(\mathbf{w}(t) \right) \right]$$

The interplay of landscape & algorithmic noise





$\mathscr{B}(t = 1)$

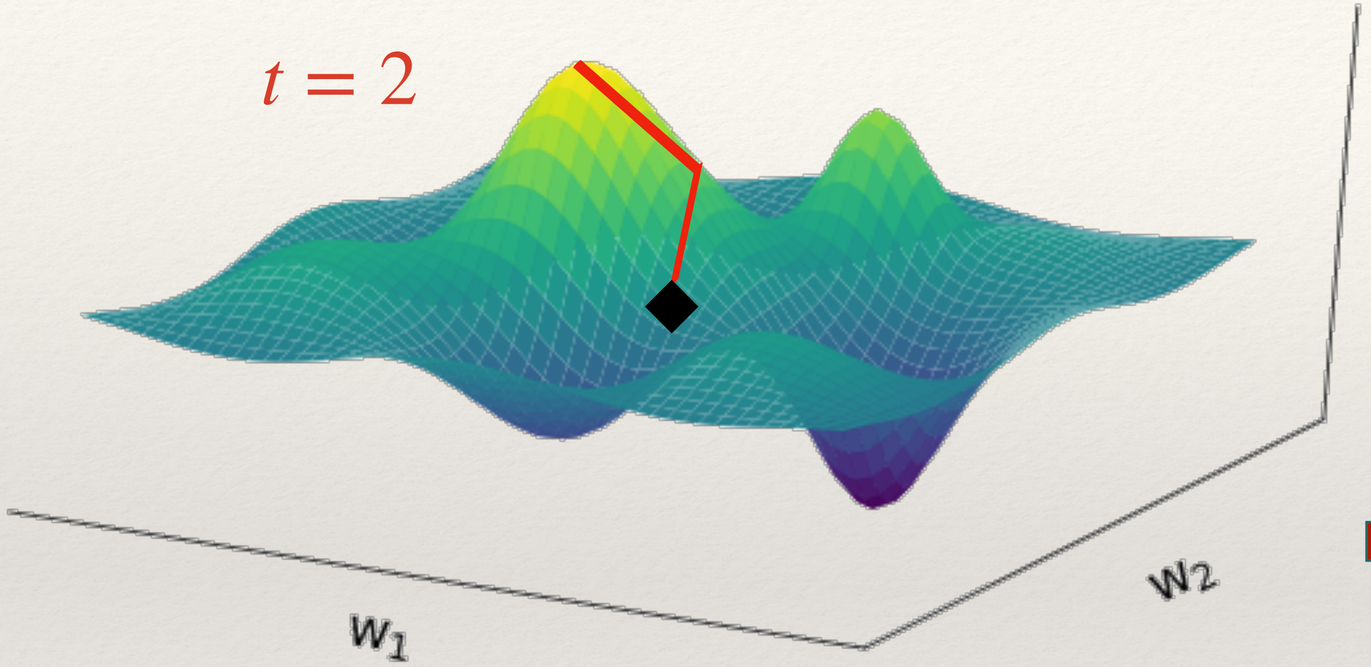
MINI BATCH

= samples used for training at time t

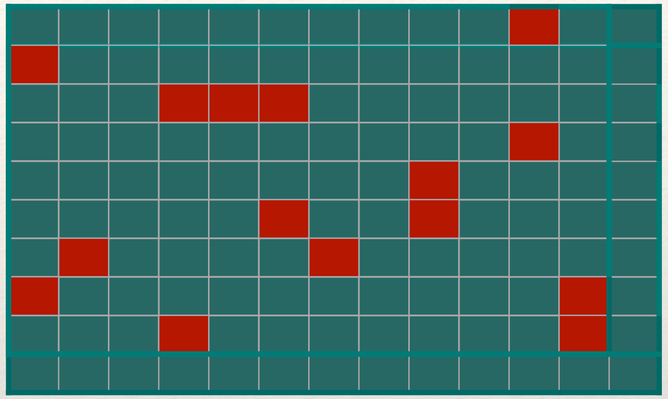
WEIGHTS UPDATE:

$$\mathbf{w}^{t+1} = \mathbf{w}^{t} - \mathrm{d}t \left[\sum_{\mu \in \mathscr{B}(t)} \nabla_{\mathbf{w}} \ell \left(y_\mu, f_{\mathbf{w}(t)}(\mathbf{x}_\mu) \right) + \nabla_{\mathbf{w}} r \left(\mathbf{w}(t) \right) \right]$$

The interplay of landscape & algorithmic noise



TRAINING SET



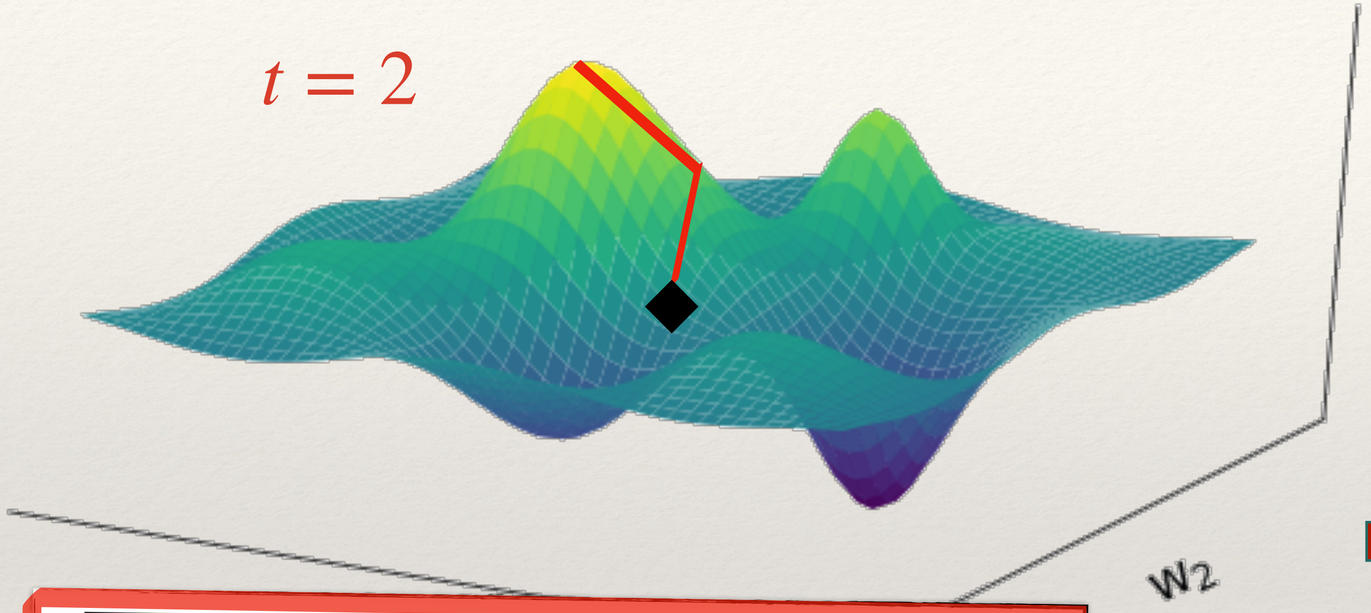
$\mathscr{B}(t = 1)$

MINI BATCH

= samples used for training at time t

WEIGHTS UPDATE: $$\mathbf{w}^{t+1} = \mathbf{w}^{t} - \mathrm{d}t \left[\sum_{\mu \in \mathscr{B}(t)} \nabla_{\mathbf{w}} \ell \left(y_\mu, f_{\mathbf{w}(t)}(\mathbf{x}_\mu) \right) + \nabla_{\mathbf{w}} r \left(\mathbf{w}(t) \right) \right]$$

The interplay of landscape & algorithmic noise



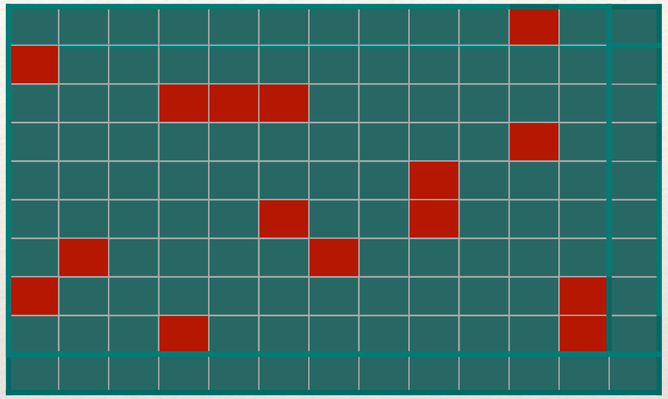
Bad Global Minima Exist and SGD Can Reach Them

Shengchao Liu, Dimitris Papailiopoulos
University of Wisconsin–Madison

Dimitris Achlioptas
University of California, Santa Cruz

[Liu, Papailiopoulos, Achlioptas, D. (2019), arXiv:1906.02613]

TRAINING SET



$\mathscr{B}(t = 1)$

MINI BATCH

= samples used for training at time t

- WHICH BASINS ARE ATTRACTIVE ?
- WHEN/WHY IS SGD BETTER THAN GD ?
- WHAT IS THE ROLE OF SGD EFFECTIVE NOISE ?

Modelling the dynamics of **mini-batch** gradient descent (SGD)

Related works (non-exhaustive list)

$$\mathbf{w}(t + \mathrm{d}t) = \mathbf{w}(t) - \mathrm{d}t\, \tilde{\nabla}_B \mathscr{L}(\mathbf{w})$$

Modelling the dynamics of **mini-batch** gradient descent (SGD)

Related works (non-exhaustive list)

$$\mathbf{w}(t + \mathrm{d}t) = \mathbf{w}(t) - \mathrm{d}t\,\tilde{\nabla}_B \mathscr{L}(\mathbf{w})$$

WEIGHTS

Modelling the dynamics of **mini-batch** gradient descent (SGD)

Related works (non-exhaustive list)

LOSS

$$\mathbf{w}(t + \mathrm{d}t) = \mathbf{w}(t) - \mathrm{d}t\, \tilde{\nabla}_B \mathscr{L}(\mathbf{w})$$

WEIGHTS

Modelling the dynamics of **mini-batch** gradient descent (SGD)

Related works (non-exhaustive list)

LOSS

$$\mathbf{w}(t + \mathrm{d}t) = \mathbf{w}(t) - \mathrm{d}t\, \tilde{\nabla}_B \mathscr{L}(\mathbf{w})$$

WEIGHTS

APPROXIMATE GRADIENT

= computed only on the training samples in the current mini-batch

Modelling the dynamics of **mini-batch** gradient descent (SGD)

Related works (non-exhaustive list)

- Modelling the <u>noise</u> of SGD:

$$\mathbf{w}(t + \mathrm{d}t) = \mathbf{w}(t) - \mathrm{d}t \nabla \mathscr{L}(\mathbf{w}) + \mathrm{d}t \left(\nabla \mathscr{L}(\mathbf{w}) - \tilde{\nabla}_{\mathscr{B}} \mathscr{L}(\mathbf{w}) \right)$$

Modelling the dynamics of **mini-batch** gradient descent (SGD)

Related works (non-exhaustive list)

- Modelling the <u>noise</u> of SGD:

$$\mathbf{w}(t + \mathrm{d}t) = \underbrace{\mathbf{w}(t) - \mathrm{d}t\,\nabla\mathscr{L}(\mathbf{w})}_{\text{Gradient descent}} + \underbrace{\mathrm{d}t\left(\nabla\mathscr{L}(\mathbf{w}) - \tilde{\nabla}_{\mathscr{B}}\mathscr{L}(\mathbf{w})\right)}_{\text{SGD noise}}$$

Modelling the dynamics of **mini-batch** gradient descent (SGD)

Related works (non-exhaustive list)

- Modelling the <u>noise</u> of SGD:

$$\mathbf{w}(t+\mathrm{d}t) = \underbrace{\mathbf{w}(t) - \mathrm{d}t\,\nabla\mathscr{L}(\mathbf{w})}_{\text{Gradient descent}} + \underbrace{\mathrm{d}t\left(\nabla\mathscr{L}(\mathbf{w}) - \tilde{\nabla}_{\mathscr{B}}\mathscr{L}(\mathbf{w})\right)}_{\text{SGD noise}}$$

  - Gaussian (Central limit theorem)
    [Mandt et al., '16; Jastrzebski et al., '17; Li et al.,'17; Hu et al., '17; Zhu et al., '18; Chaudhari, Soatto, '18, ...]

Modelling the dynamics of **mini-batch** gradient descent (SGD)

Related works (non-exhaustive list)

- Modelling the <u>noise</u> of SGD:

$$\mathbf{w}(t+\mathrm{d}t) = \underbrace{\mathbf{w}(t) - \mathrm{d}t\,\nabla\mathscr{L}(\mathbf{w})}_{\text{Gradient descent}} + \underbrace{\mathrm{d}t\left(\nabla\mathscr{L}(\mathbf{w}) - \tilde{\nabla}_{\mathscr{B}}\mathscr{L}(\mathbf{w})\right)}_{\text{SGD noise}}$$

  - Gaussian (Central limit theorem)
    [Mandt et al., '16; Jastrzebski et al., '17; Li et al.,'17; Hu et al., '17; Zhu et al., '18; Chaudhari, Soatto, '18, ...]
  - Fat-tailed (Generalized CLT)
    [Simsekli et al., '19; Gurbuzbalaban et al., '20; Simsekli et al., '20]

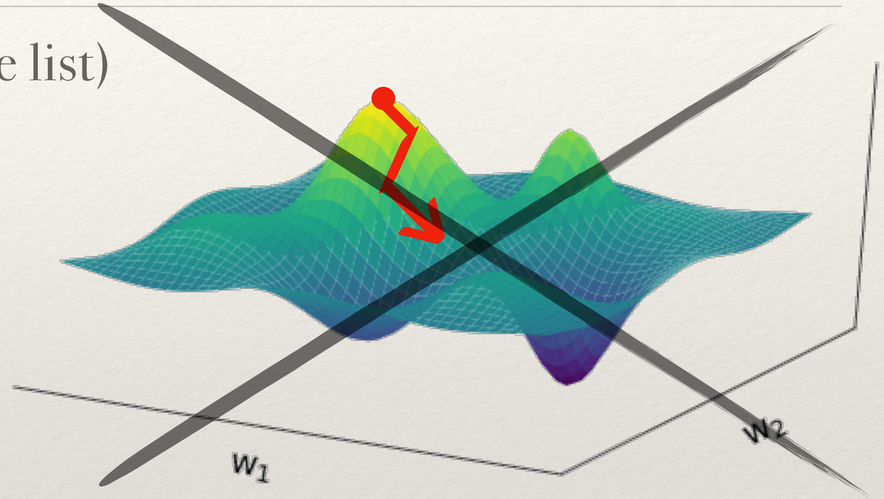
Modelling the dynamics of ~~**mini-batch** gradient descent (SGD)~~

Related works (non-exhaustive list)

- Tracking the <u>whole learning trajectory</u>:
  - **Gradient descent** in (deep) linear networks
    [Bös, Opper, '97 ; Saxe, McClelland, Ganguli, '13; Advani, Saxe, Sompolinksy, '20]

stochastic ~~mini-batch~~ Modelling the dynamics of gradient descent (SGD)

Related works (non-exhaustive list)



- Tracking the <u>whole learning trajectory</u>:
  - Gradient descent in (deep) linear networks
  - **Online** SGD

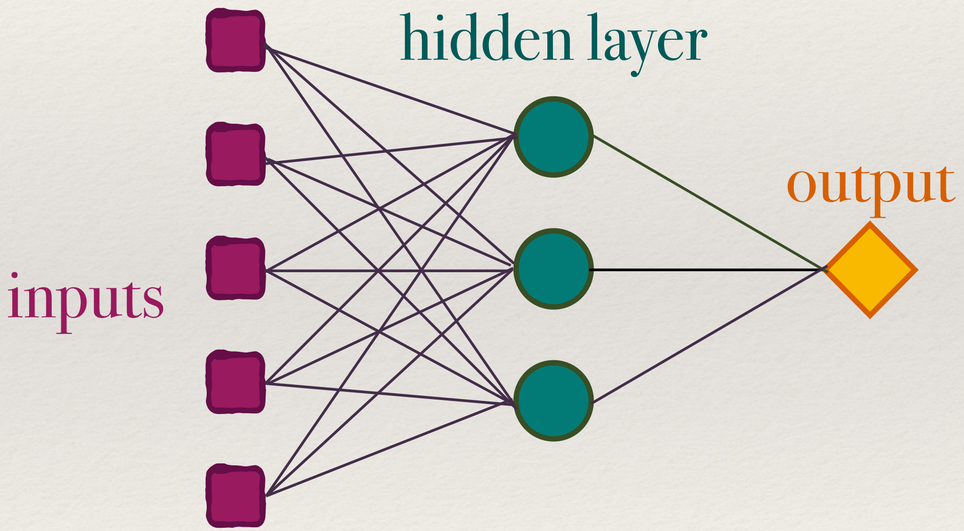
    i.e., a <u>new sample</u> is used to update the weights at each time step

    no distinction between train & test

stochastic ~~mini-batch~~ Modelling the dynamics of gradient descent (SGD)

Related works (non-exhaustive list)

- Tracking the <u>whole learning trajectory</u>:
  - Gradient descent in (deep) linear networks
  - Online SGD for **two-layer** neural networks



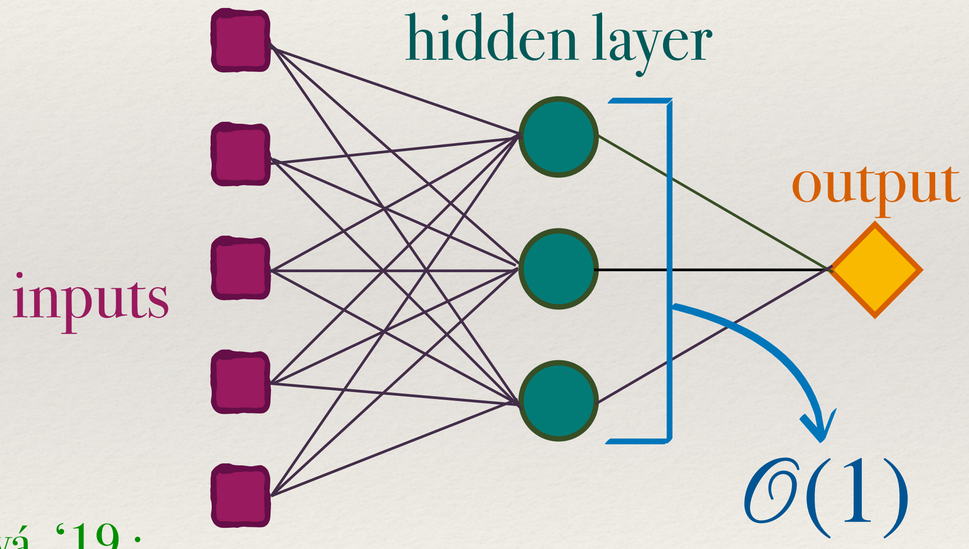
stochastic ~~mini-batch~~ Modelling the dynamics of gradient descent (SGD)

Related works (non-exhaustive list)

- Tracking the whole learning trajectory:
  - Gradient descent in (deep) linear networks
  - Online SGD for **two-layer** neural networks

– with **finite hidden layer**

[in physics: Saad, Solla, '95; Saad, '09; Goldt, Advani, Saxe, Krzakala, Zdeborová, '19; Goldt, Mézard, Krzakala, Zdeborová, '19 ; Refinetti, et al., '20, '21]



stochastic ~~mini-batch~~ Modelling the dynamics of gradient descent (SGD)

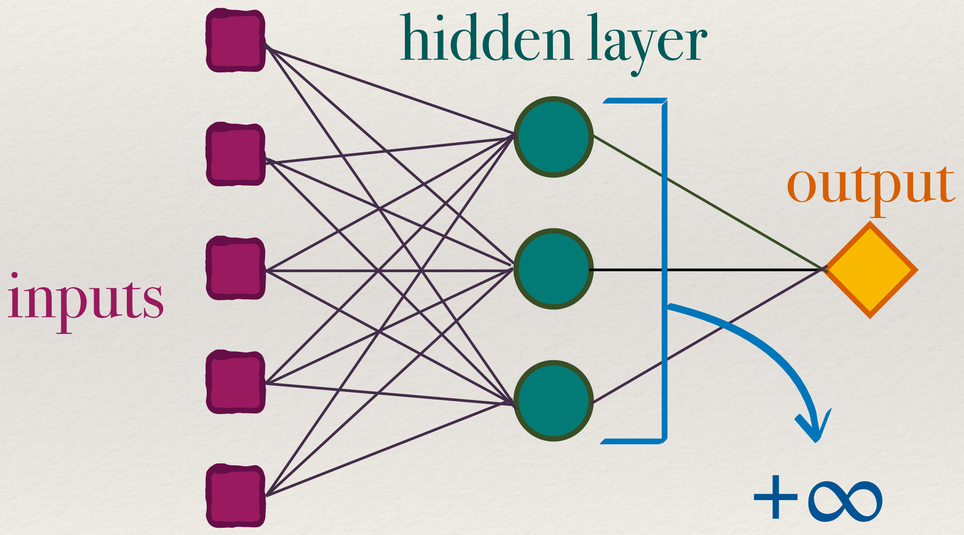
Related works (non-exhaustive list)

- Tracking the whole learning trajectory:
  - Gradient descent in (deep) linear networks
  - Online SGD for **two-layer** neural networks

– with **finite hidden layer**

– with **infinitely wide hidden layer**

[Rotskoff, Vanden-Eijnden, '18; Mei, Montanari, Nguyen, '18; Chizat, Bach, '18; ...]



Modelling the dynamics of **mini-batch** gradient descent (SGD)

- Tracking the <u>whole learning trajectory</u>:
  - **Multi - pass** SGD in shallow neural networks

    [**FM**, Krzakala, Urbani, Zdeborová, '20 ; **FM**, Urbani, Zdeborová, '21 ; **FM**, Urbani, '21 ; Bordelon, Pehlevan, '21]

Modelling the dynamics of **mini-batch** gradient descent (SGD)

- Tracking the <u>whole learning trajectory</u>:
  - **Multi - pass** SGD in shallow neural networks

    [**FM**, Krzakala, Urbani, Zdeborová, '20 ; **FM**, Urbani, Zdeborová, '21 ; **FM**, Urbani, '21 ; Bordelon, Pehlevan, '21]

KEY INGREDIENTS:

Generative model for the data + Typical-case scenario + Thermodynamic limit

(#samples, #dimensions $\to \infty$)

Modelling the dynamics of **mini-batch** gradient descent (SGD)

Based on:

FM, Krzakala, Urbani, Zdeborová, *NeurIPS* 2020 ;
FM, Urbani, Zdeborová, *Machine Learning: Science & Technology* (2020) ;
FM, Urbani, arXiv preprint: *arXiv:2112.10852*

Joint works with:

Pierfrancesco Urbani
@IPhT

Lenka Zdeborová
@EPFL

Florent Krzakala
@EPFL

Modelling the dynamics of **mini-batch** gradient descent (SGD)

Sampling protocols for the **mini batch** $\mathscr{B}(t)$

- **Vanilla** SGD: $\mu \in \mathscr{B}(t)$ with probability $b \in (0,1]$ i.i.d. at each time

$$\frac{1}{M}\langle\, |\mathscr{B}(t)|\, \rangle = b$$ "BATCH SIZE"

Modelling the dynamics of **mini-batch** gradient descent (SGD)

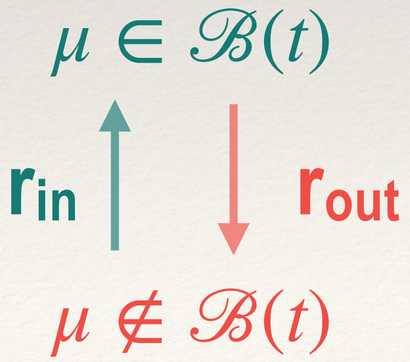
Sampling protocols for the **mini batch** $\mathscr{B}(t)$

- **Vanilla** SGD: $\mu \in \mathscr{B}(t)$ with probability $b \in (0,1]$ i.i.d. at each time

$$\frac{1}{M}\langle\, |\mathscr{B}(t)|\, \rangle = b \qquad \text{“BATCH SIZE”}$$

- **Persistent** SGD: two-state Markov jump process

$$\frac{1}{M}\langle\, |\mathscr{B}(t)|\, \rangle = b, \qquad \tau \quad \text{“PERSISTENCE TIME”}$$



Dynamical mean-field theory for stochastic gradient descent

Thermodynamic limit: #dimensions, #samples $\longrightarrow \infty$, at fixed sample complexity $\alpha = \dfrac{\#\text{samples}}{\#\text{dimensions}}$

Markovian dynamics of strongly coupled degrees of freedom $\to \infty$

Stochastic gradient flow equations

$$\dot{\mathbf{w}}(t) = -\sum_{\mu \in \mathscr{B}(t)} \nabla_{\mathbf{w}} \ell\left(\mathbf{w}(t), y_\mu, \mathbf{x}_\mu\right) - \nabla_{\mathbf{w}} r(\mathbf{w}(t)) \in \mathbb{R}^N$$

[Sompolinsky, Zippelius, '81; Mézard, Parisi, Virasoro, '87; Sompolinsky, Crisanti, Sommers, '88 ... Agoritsas, Birol, Urbani, Zamponi (2018)]

Dynamical mean-field theory for stochastic gradient descent

Thermodynamic limit: #dimensions, #samples $\longrightarrow \infty$, at fixed sample complexity $\alpha = \dfrac{\#\text{samples}}{\#\text{dimensions}}$

Markovian dynamics of strongly coupled degrees of freedom $\to \infty$

dimensional reduction

Non-Markovian dynamics of one effective degree of freedom

Stochastic gradient flow equations

$$\dot{\mathbf{w}}(t) = -\sum_{\mu \in \mathscr{B}(t)} \nabla_{\mathbf{w}} \ell\left(\mathbf{w}(t), y_\mu, \mathbf{x}_\mu\right) - \nabla_{\mathbf{w}} r(\mathbf{w}(t)) \in \mathbb{R}^N$$

One effective stochastic process + one ODE

$$\dot{h}(t) = -\tilde{\lambda}(t)\, h(t) - s(t)\, \Lambda'(y, h(t), m(t)) + \int_0^t \mathrm{d}t'\, M_R(t,t') h(t') + \xi(t)$$

$$\dot{m}(t) = -\lambda m(t) - \mu(t)$$

[Sompolinsky, Zippelius, '81; Mézard, Parisi, Virasoro, '87; Sompolinsky, Crisanti, Sommers, '88 ... Agoritsas, Birol, Urbani, Zamponi (2018)]

Dynamical mean-field theory for stochastic gradient descent

Thermodynamic limit: #dimensions, #samples $\longrightarrow \infty$, at fixed sample complexity $\alpha = \dfrac{\#\text{samples}}{\#\text{dimensions}}$

Markovian dynamics of strongly coupled degrees of freedom $\to \infty$

$\downarrow$ **dimensional reduction**

Non-Markovian dynamics of one effective degree of freedom

Stochastic gradient flow equations

$$\dot{\mathbf{w}}(t) = -\sum_{\mu \in \mathscr{B}(t)} \nabla_{\mathbf{w}} \ell\left(\mathbf{w}(t), y_\mu, \mathbf{x}_\mu\right) - \nabla_{\mathbf{w}} r(\mathbf{w}(t)) \in \mathbb{R}^N$$

One effective stochastic process + one ODE

$$\dot{h}(t) = -\tilde{\lambda}(t)\, h(t) - s(t)\, \Lambda'(y, h(t), m(t)) + \int_0^t \mathrm{d}t'\, M_R(t,t') h(t') + \xi(t)$$

$$\dot{m}(t) = -\lambda m(t) - \mu(t)$$

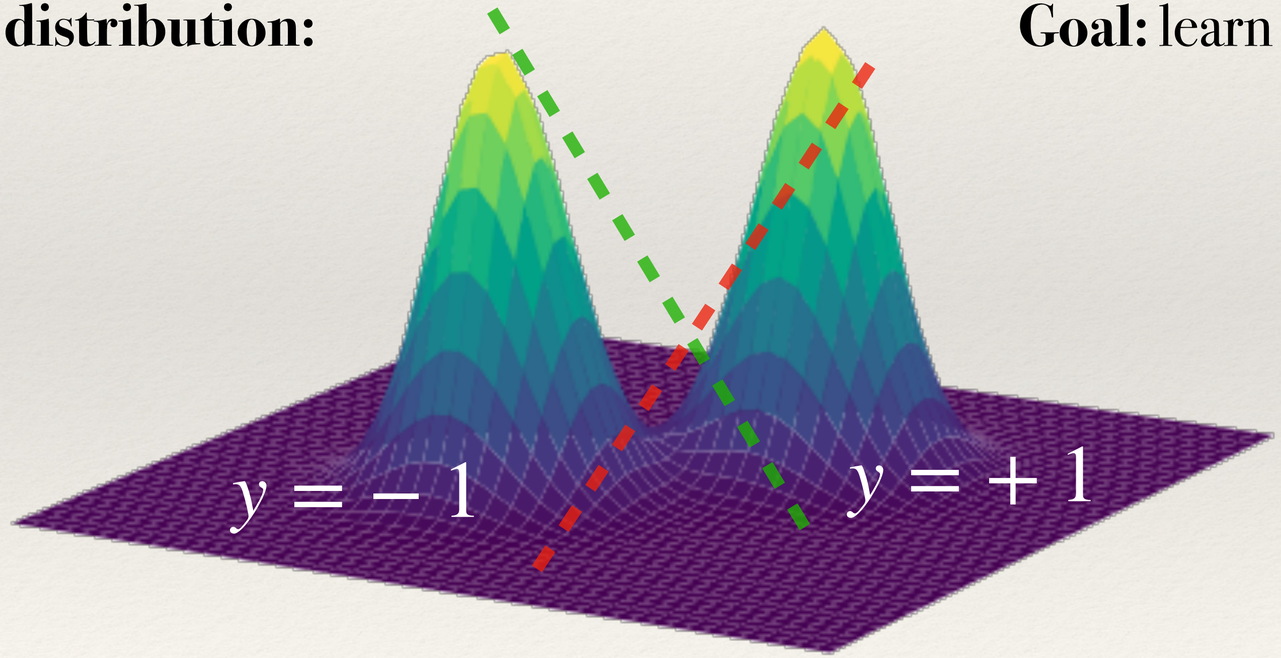
To be solved self-consistently

[Eissfeller, Opper, '97; Roy et al., '19; Manacorda et al., '20]

Dynamical mean-field theory for stochastic gradient descent

The problem: BINARY GAUSSIAN MIXTURES CLASSIFICATION

Data distribution:

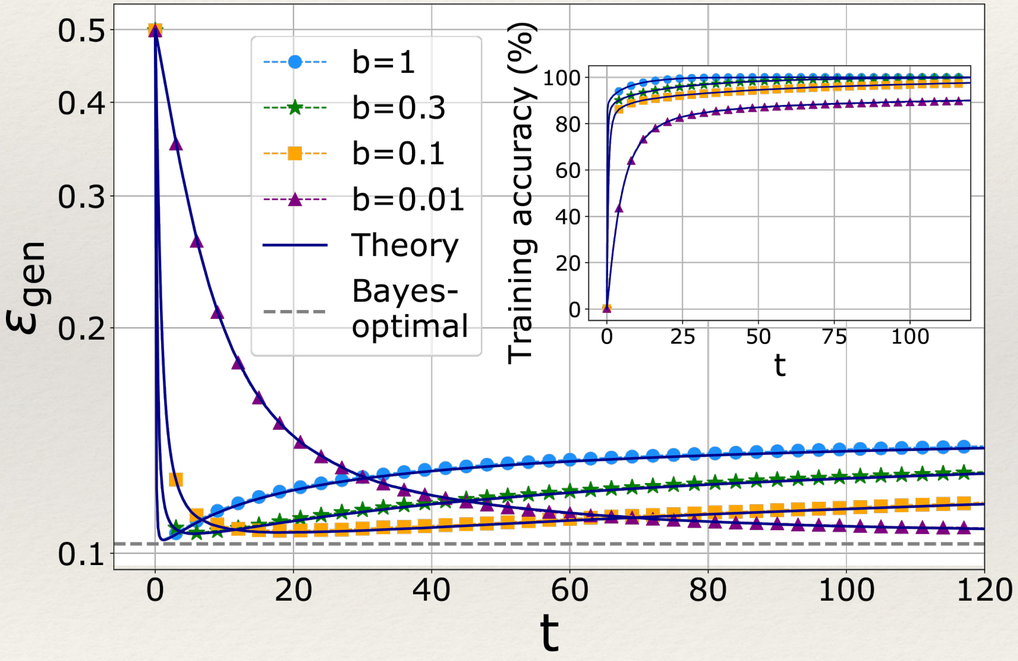


Goal: learn the best separating hyperplane

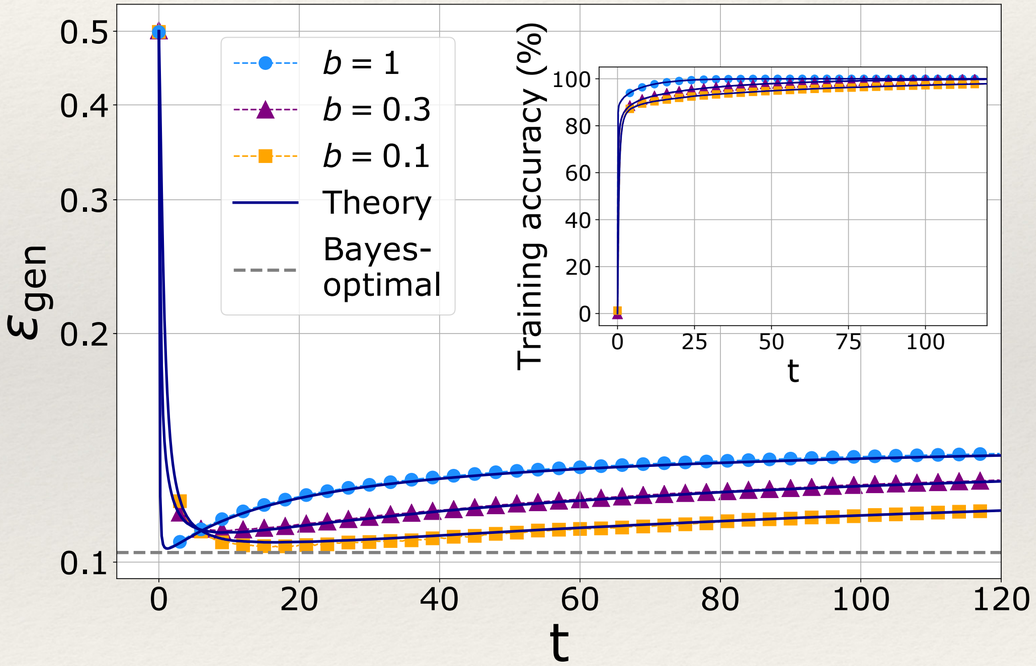
Dynamical mean-field theory for stochastic gradient descent

<u>The problem:</u> BINARY GAUSSIAN MIXTURES CLASSIFICATION

Vanilla SGD



Persistent SGD



[**FM**, F. Krzakala, P. Urbani, L. Zdeborová, NeurIPS (2020)]

$\alpha = 2, 1/\tau = 0.6, \mathrm{d}t = 0.2$, simulations at $N = 500$

Characterizing SGD noise

The problem: Binary Gaussian mixtures classification

If the dynamics was at equilibrium $\longrightarrow$ Fluctuation-Dissipation Theorem

$$R(t,t') = -\frac{1}{T}\partial_t C(t,t')\,\Theta(t-t')$$

Characterizing SGD noise

The problem: BINARY GAUSSIAN MIXTURES CLASSIFICATION

If the dynamics was at equilibrium ⟶ Fluctuation-Dissipation Theorem

$$R(t,t') = -\frac{1}{T}\partial_t C(t,t')\,\Theta(t-t')$$

Linear response

$$R(t,t') = \lim_{\{H_j \to 0\}} \frac{1}{N}\sum_{j=1}^{N} \frac{\delta w_j(t)}{\delta H_j(t')}$$

Correlation

$$C(t,t') = \frac{1}{N}\mathbf{w}(t)^\top \mathbf{w}(t')$$

Integrating on both sides and rescaling:

$$\bar{\chi}(t,t') = \frac{1}{T}\left(1 - \bar{C}(t,t')\right)$$

Characterizing SGD noise

The problem: BINARY GAUSSIAN MIXTURES CLASSIFICATION

If the dynamics was at equilibrium ⟶ Fluctuation-Dissipation Theorem

$$R(t,t') = -\frac{1}{T}\partial_t C(t,t')\,\Theta(t-t')$$

Linear response

$$R(t,t') = \lim_{\{H_j \to 0\}} \frac{1}{N}\sum_{j=1}^{N} \frac{\delta w_j(t)}{\delta H_j(t')}$$

Correlation

$$C(t,t') = \frac{1}{N}\mathbf{w}(t)^\top \mathbf{w}(t')$$

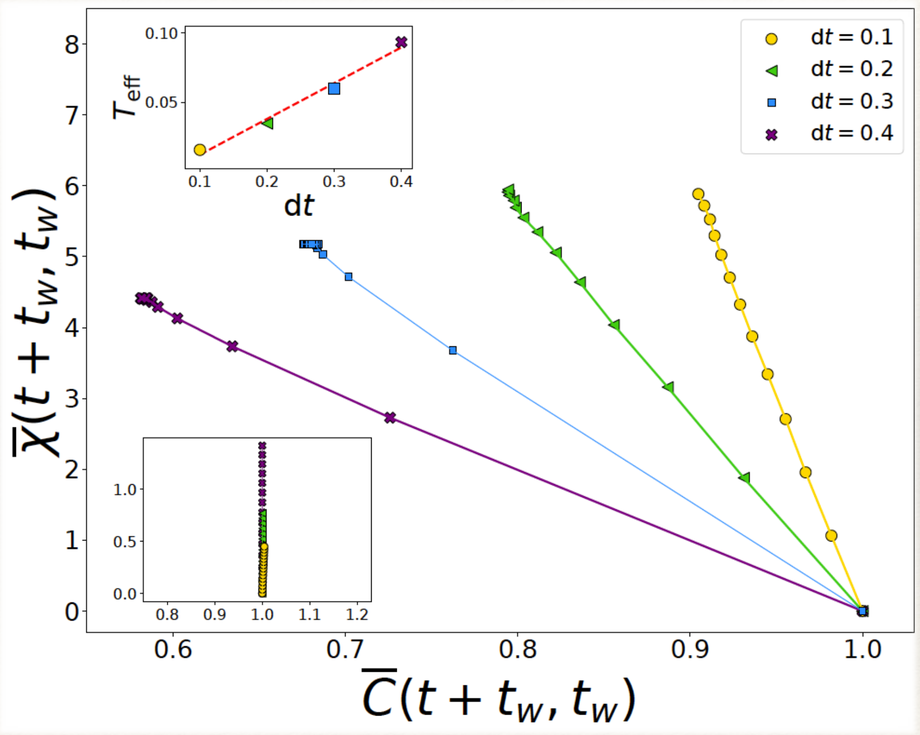
Integrating on both sides and rescaling:

$$\bar{\chi}(t,t') = \frac{1}{T}\left(1 - \bar{C}(t,t')\right)$$

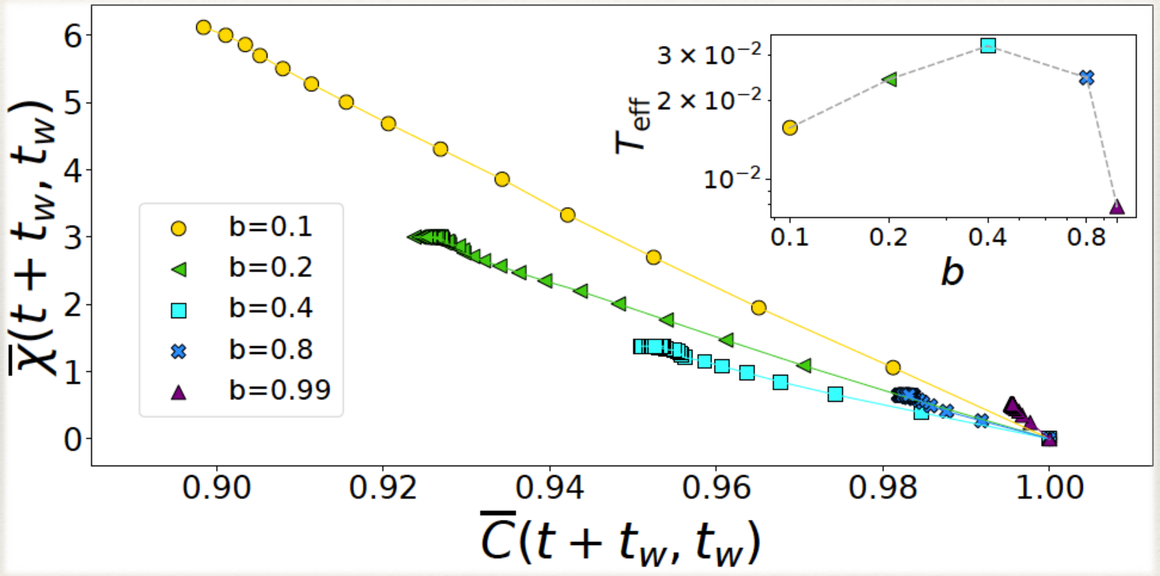
Directly available from DMFT

The effective temperature of vanilla SGD

Changing the time step:



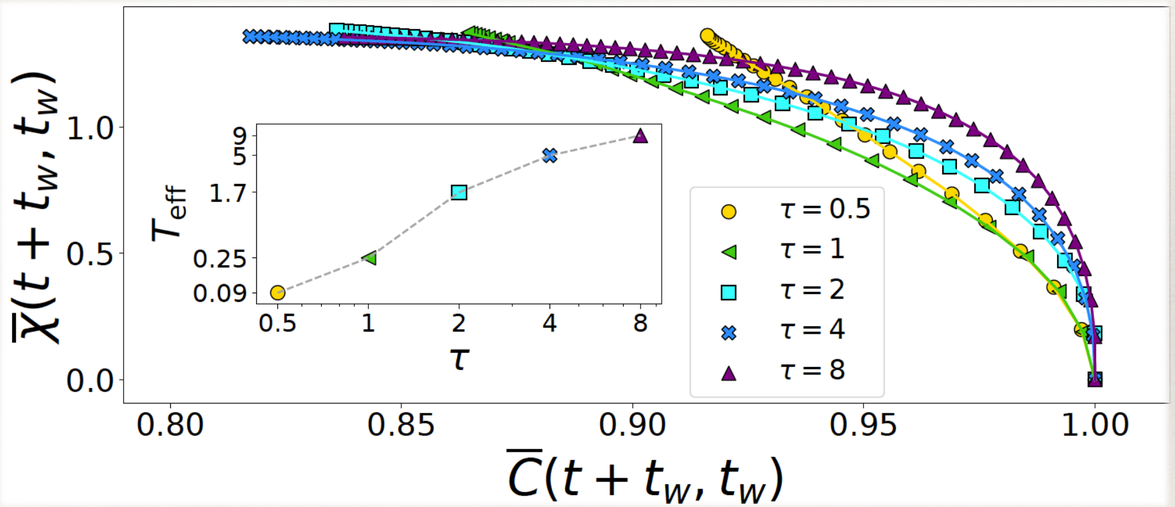
Changing the batch size:



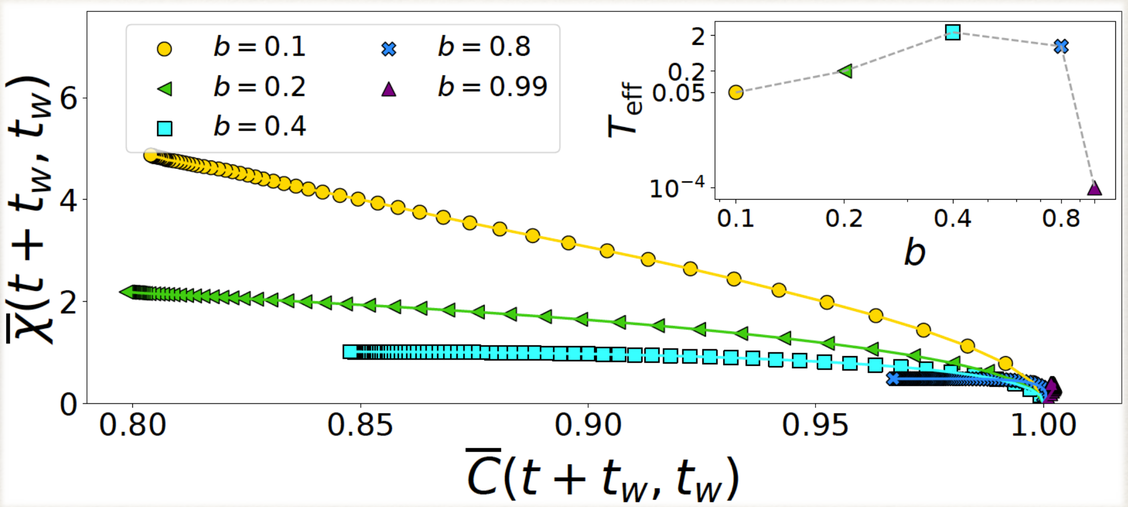
An **effective fluctuation-dissipation theorem** holds for SGD at stationarity in the under-parametrized regime.

The effective temperature of persistent SGD

Changing the persistence time:



Changing the batch size:



An **effective fluctuation-dissipation theorem** holds for SGD at stationarity in the under-parametrized regime.

High-dimensional non-convex losses: the blessing of the noise

THE SIGN-RETRIEVAL PROBLEM :

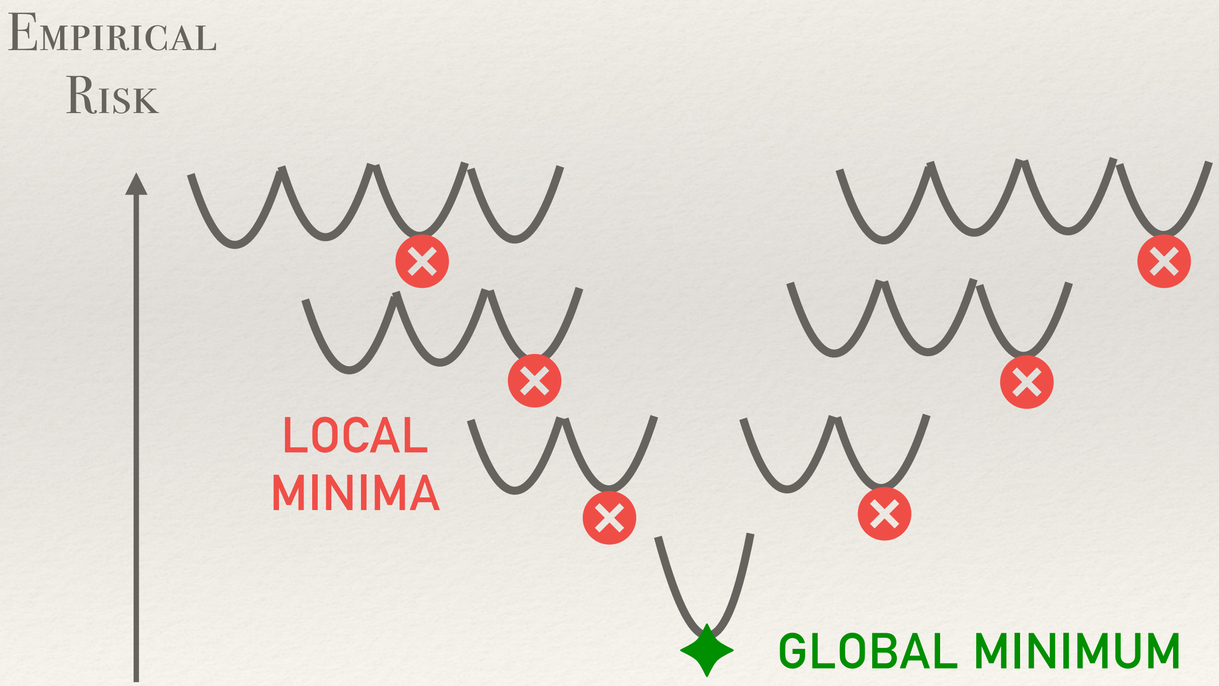
Data model:

$$\mathbf{x}_\mu \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}_N\right)$$

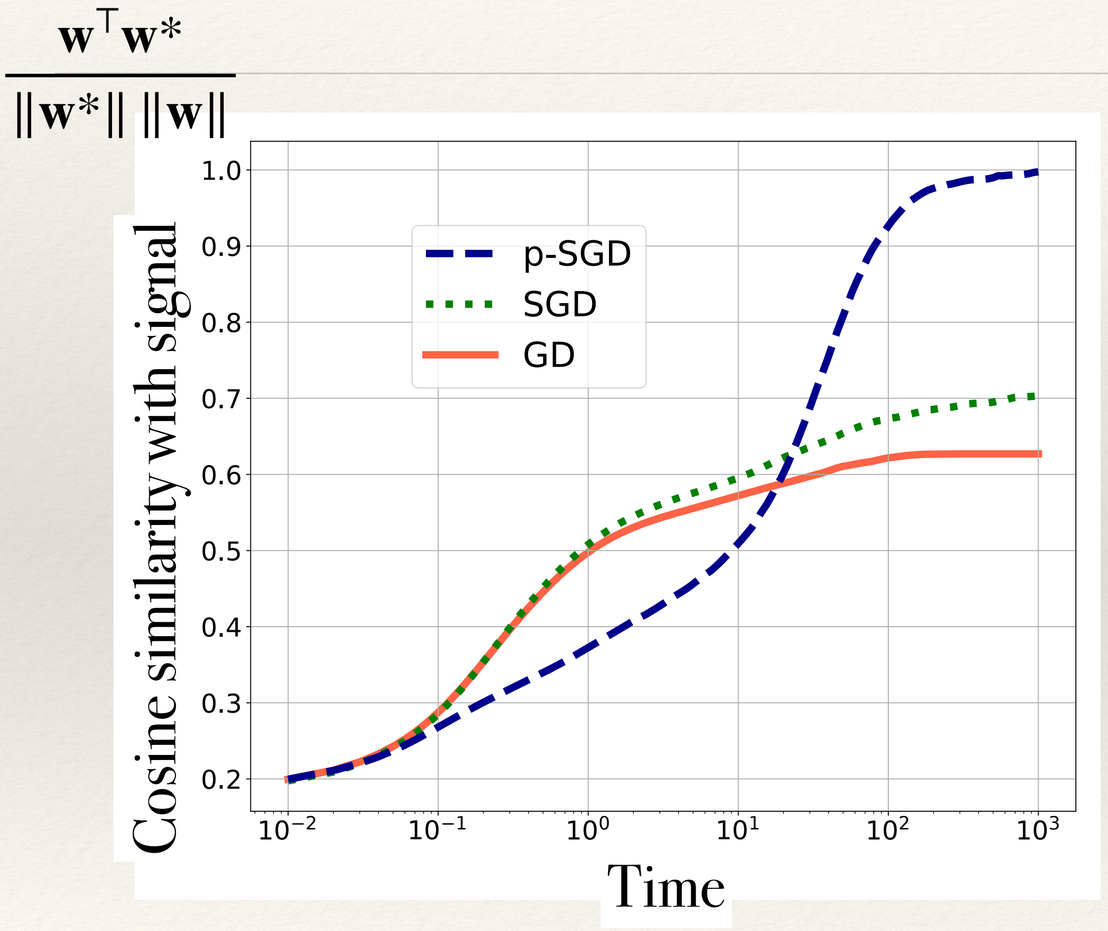
$$y_\mu = |\mathbf{x}_\mu^\top \mathbf{w}^*|$$

Loss function:

$$\ell\left(y_\mu, \mathbf{w}^\top \mathbf{x}_\mu\right) = \left(y_\mu^2 - (\mathbf{w}^\top \mathbf{x}_\mu)^2\right)^2$$

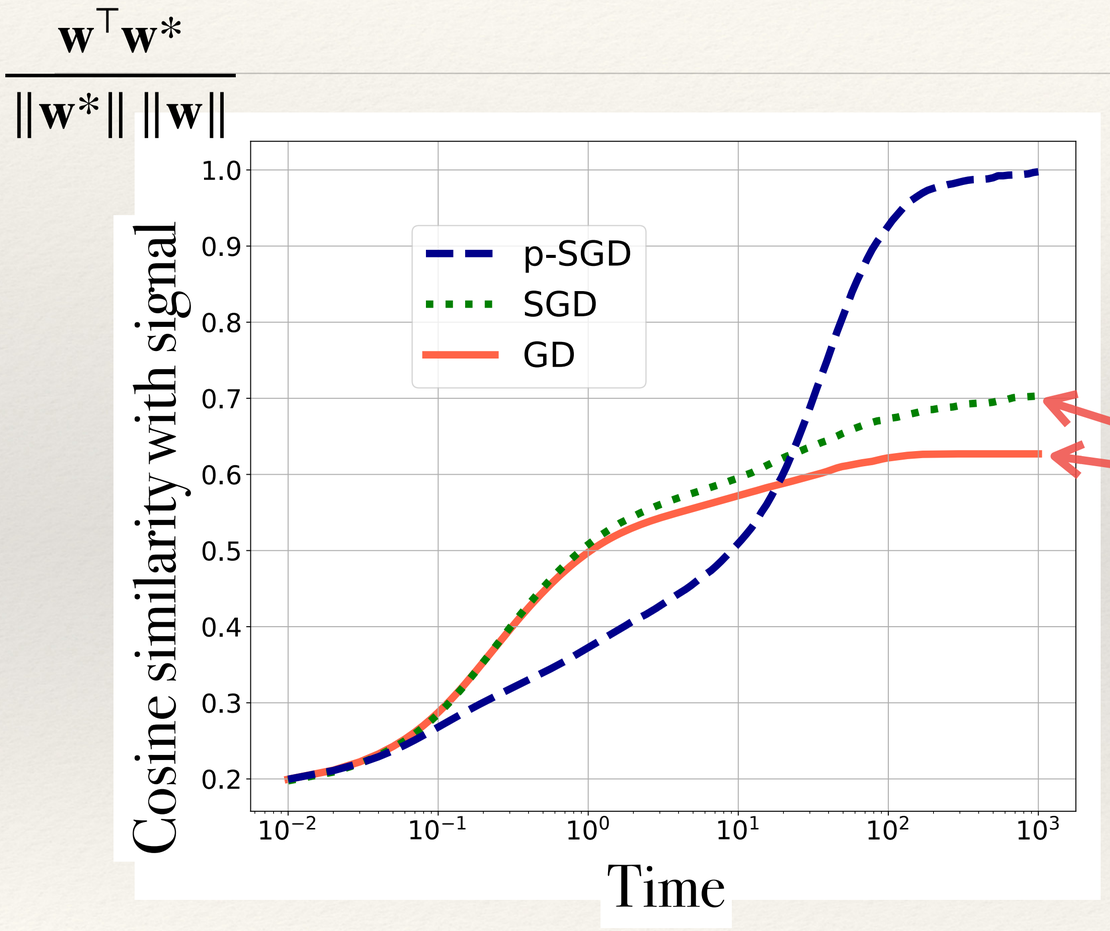


High-dimensional non-convex losses: the blessing of the noise



FM, P. Urbani, L. Zdeborová, Machine Learning: Science and Technology (2021)

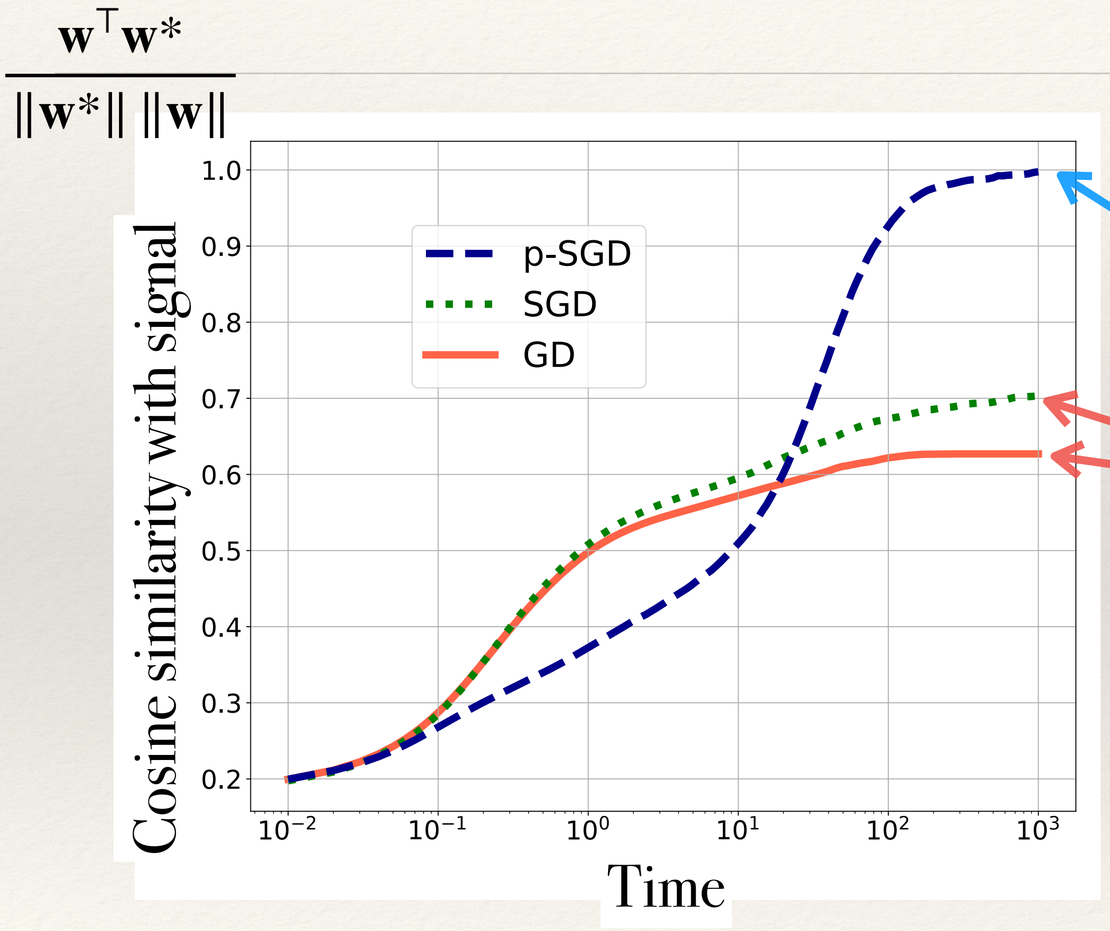
High-dimensional non-convex losses: the blessing of the noise



Vanilla SGD and GD get stuck in local minima

FM, P. Urbani, L. Zdeborová, Machine Learning: Science and Technology (2021)

High-dimensional non-convex losses: the blessing of the noise

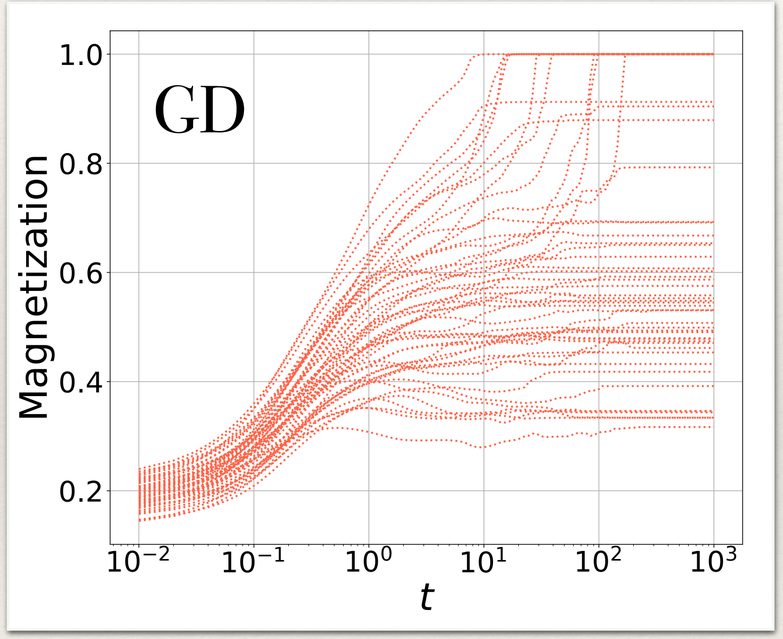


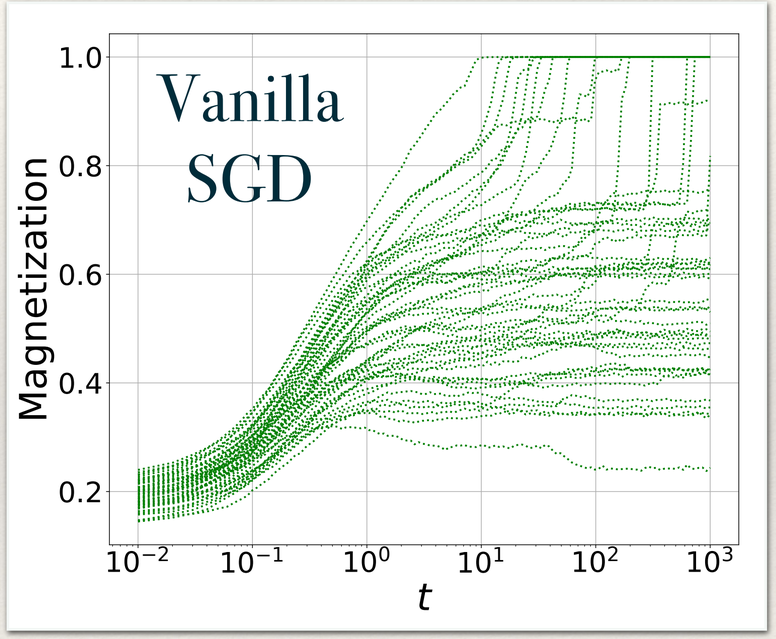
Persistent SGD achieves perfect recovery

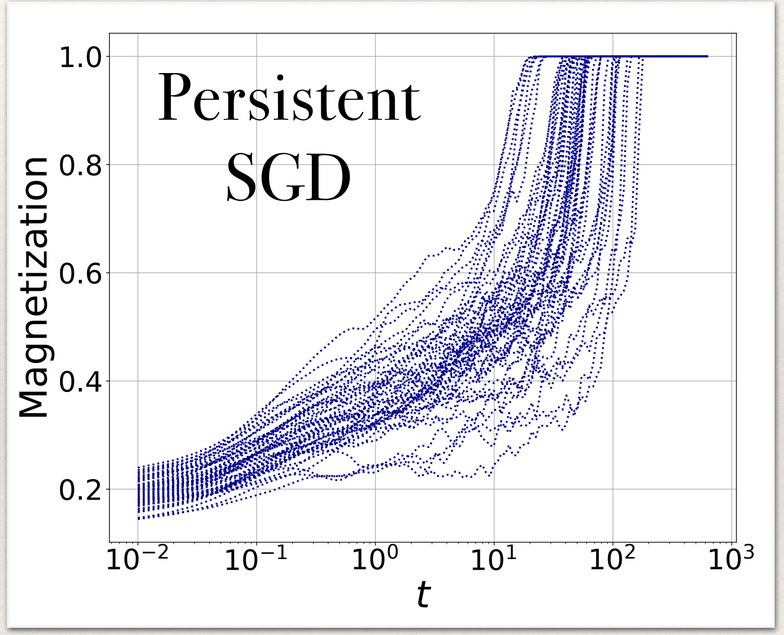
Vanilla SGD and GD get stuck in local minima

FM, P. Urbani, L. Zdeborová, Machine Learning: Science and Technology (2021)

High-dimensional non-convex losses: the blessing of the noise



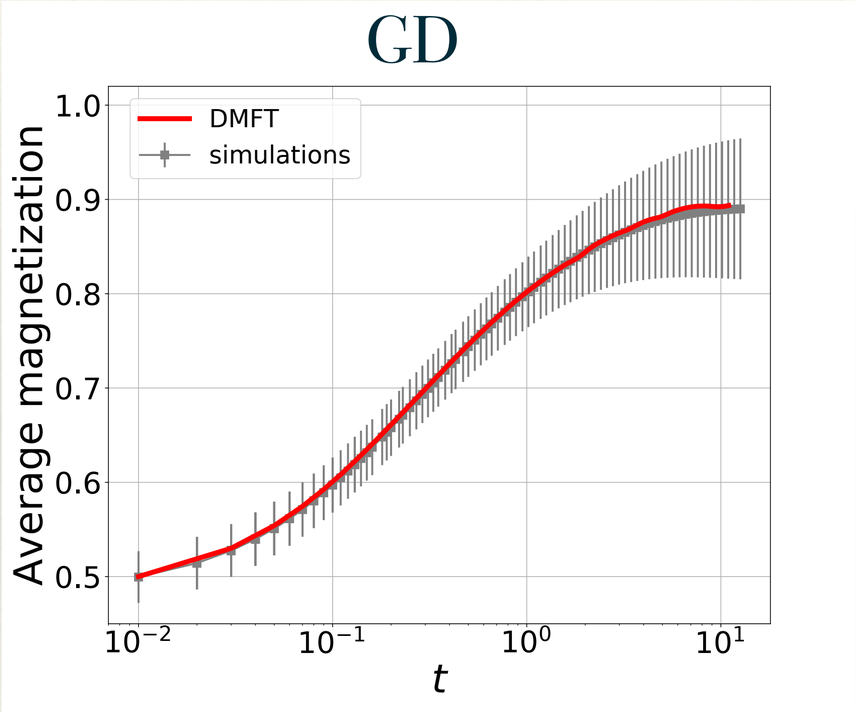


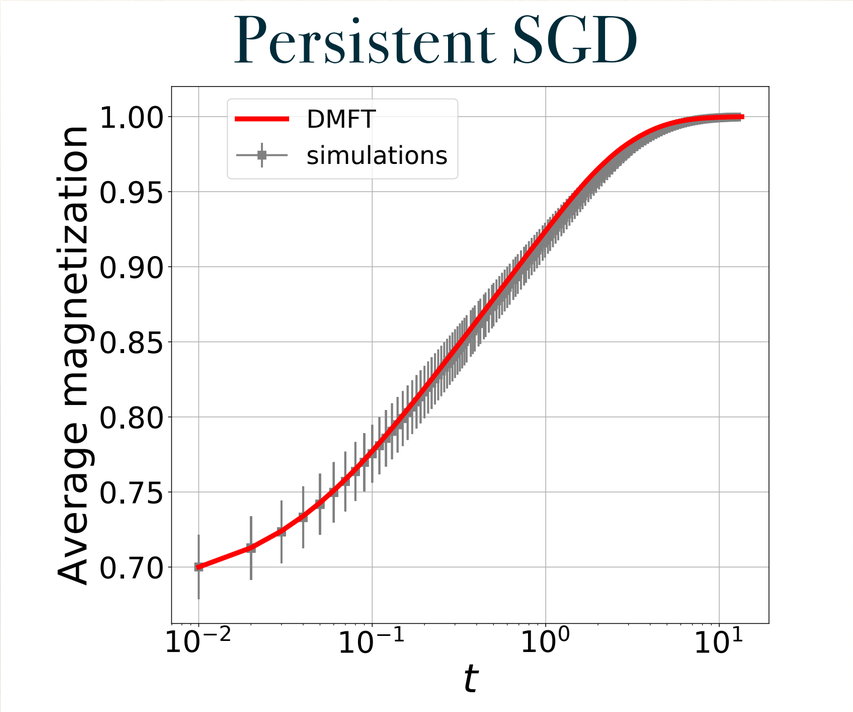


The algorithmic noise produces an **effective self-annealing**.

FM, P. Urbani, L. Zdeborová, Machine Learning: Science and Technology (2021)

High-dimensional non-convex losses: the blessing of the noise





The theory keeps finite-size effects under control.

FM, P. Urbani, L. Zdeborová, Machine Learning: Science and Technology (2021)

Modelling realistic **data structure**

Based on:

B. Loureiro, C. Gerbelot, H. Cui, S. Goldt, F. Krzakala, M. Mezard, L. Zdeborová,
"Learning curves of generic features maps for realistic datasets with a teacher-student model".
Advances in Neural Information Processing Systems 34 (2021).

Modelling realistic **data structure**

Based on:

B. Loureiro, C. Gerbelot, H. Cui, S. Goldt, F. Krzakala, M. Mezard, L. Zdeborová,
"Learning curves of generic features maps for realistic datasets with a teacher-student model".
Advances in Neural Information Processing Systems 34 (2021).

The model:

$$\begin{bmatrix} \boldsymbol{u}^\mu \\ \boldsymbol{v}^\mu \end{bmatrix} \in \mathbb{R}^{p+d} \underset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \begin{bmatrix} \Psi & \Phi \\ \Phi^\top & \Omega \end{bmatrix}\right)$$

Modelling realistic **data structure**

Based on:

B. Loureiro, C. Gerbelot, H. Cui, S. Goldt, F. Krzakala, M. Mezard, L. Zdeborová, "Learning curves of generic features maps for realistic datasets with a teacher-student model". *Advances in Neural Information Processing Systems* 34 (2021).

The model:

$$\begin{bmatrix} \boldsymbol{u}^\mu \\ \boldsymbol{v}^\mu \end{bmatrix} \in \mathbb{R}^{p+d} \underset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \begin{bmatrix} \Psi & \Phi \\ \Phi^\top & \Omega \end{bmatrix}\right)$$

LABELS

$$y^\mu = f_0\left(\frac{1}{\sqrt{p}}\boldsymbol{\theta}_0^\top \boldsymbol{u}^\mu\right)$$

Modelling realistic **data structure**

Based on:

B. Loureiro, C. Gerbelot, H. Cui, S. Goldt, F. Krzakala, M. Mezard, L. Zdeborová, "Learning curves of generic features maps for realistic datasets with a teacher-student model". *Advances in Neural Information Processing Systems* 34 (2021).

The model:

$$\begin{bmatrix} \boldsymbol{u}^\mu \\ \boldsymbol{v}^\mu \end{bmatrix} \in \mathbb{R}^{p+d} \underset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \begin{bmatrix} \Psi & \Phi \\ \Phi^\top & \Omega \end{bmatrix}\right)$$

LABELS

$$y^\mu = f_0\left(\frac{1}{\sqrt{p}} \boldsymbol{\theta}_0^\top \boldsymbol{u}^\mu\right)$$

TEACHER

Modelling realistic **data structure**

Based on:

B. Loureiro, C. Gerbelot, H. Cui, S. Goldt, F. Krzakala, M. Mezard, L. Zdeborová, "Learning curves of generic features maps for realistic datasets with a teacher-student model". *Advances in Neural Information Processing Systems* 34 (2021).

The model:

$$\begin{bmatrix} \boldsymbol{u}^\mu \\ \boldsymbol{v}^\mu \end{bmatrix} \in \mathbb{R}^{p+d} \underset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, \begin{bmatrix} \Psi & \Phi \\ \Phi^\top & \Omega \end{bmatrix}\right)$$

$$y^\mu = f_0\left(\frac{1}{\sqrt{p}}\boldsymbol{\theta}_0^\top \boldsymbol{u}^\mu\right)$$

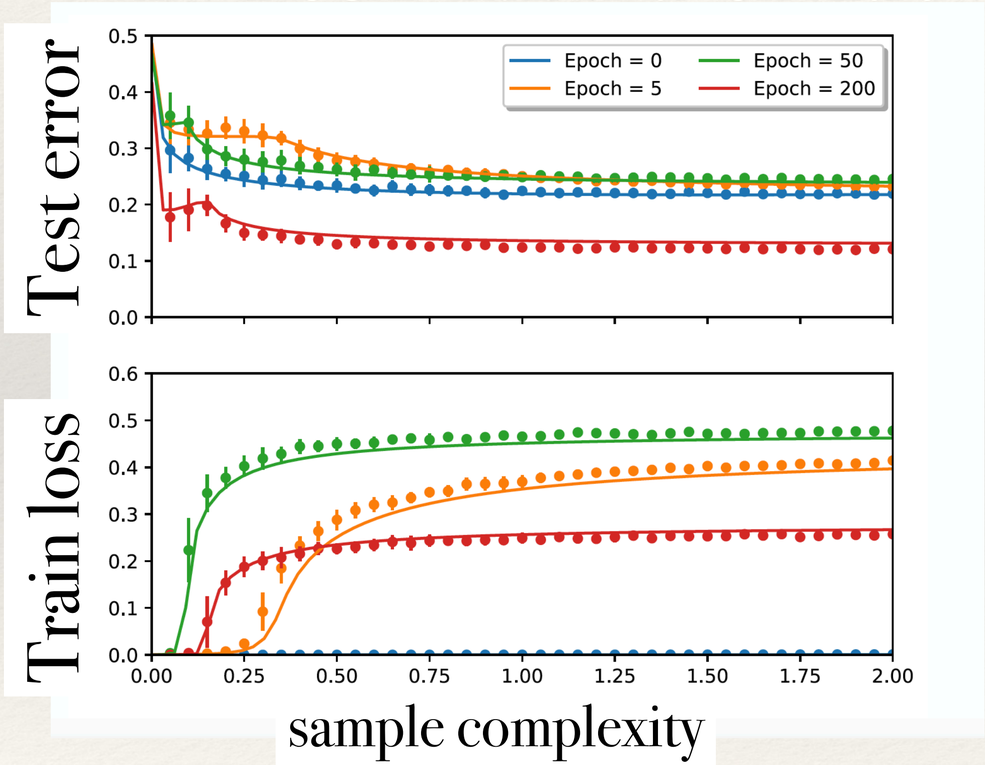
LEARNING

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}\in\mathbb{R}^d}\left[\sum_{\mu=1}^{n} g\left(\frac{\boldsymbol{w}^\top \boldsymbol{v}^\mu}{\sqrt{d}}, y^\mu\right) + r(\boldsymbol{w})\right]$$

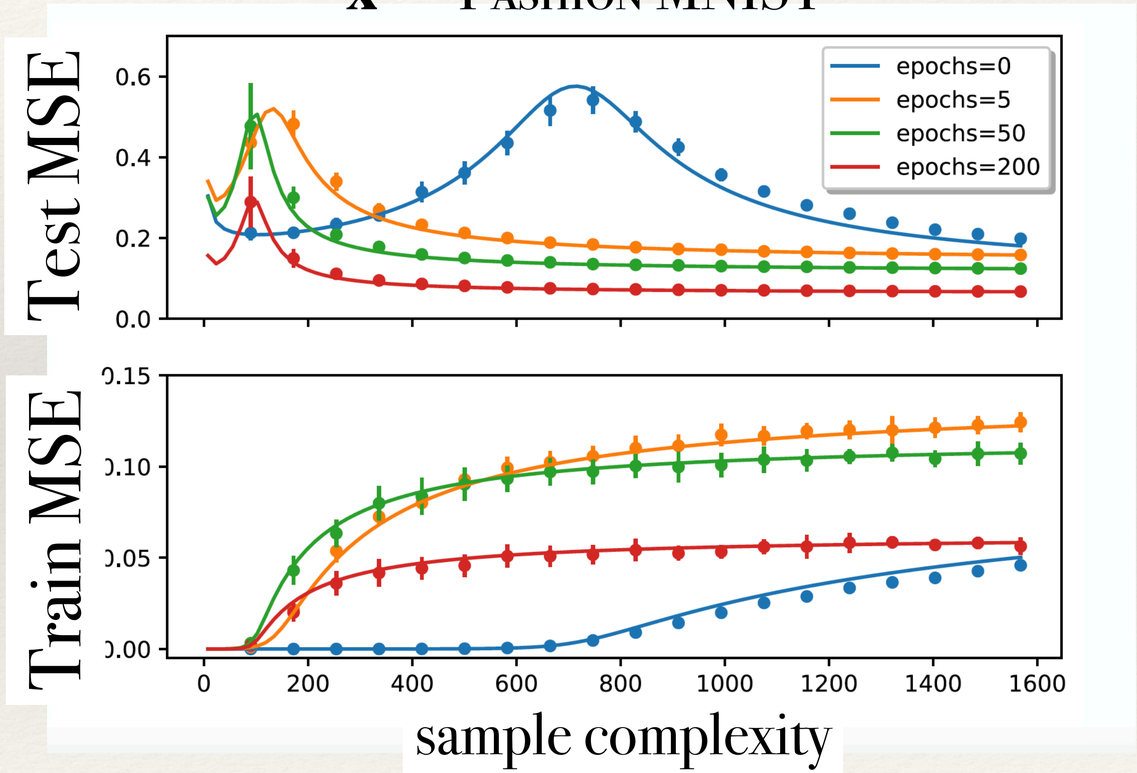
Modelling realistic **data structure**: the asymptotic-time learning curves

[**Loureiro, et al.** *Advances in Neural Information Processing Systems* 34 (2021)]

$\mathbf{x} \sim$ DCGAN TRAINED ON CIFAR10



$\mathbf{x} \sim$ FASHION MNIST



Logistic regression, $f = \text{sign}$, $\mathbf{v} = \text{relu}\left(W_1 \text{relu}\left(W_2 \mathbf{x}\right)\right)$

Ridge regression, $f = \text{id}$, $\mathbf{v} = \text{relu}\left(W_1 \text{relu}\left(W_2 \mathbf{x}\right)\right)$

More open questions:

- How to characterise the noise when the dynamics **stops** ($T_{\text{eff}} \to 0$) ?
- Can we link the noise magnitude to generalisation properties ?
- Exploring SGD with more realistic data models ([Goldt, et al., 2020]), optimisation strategies (e.g., *momentum* [Mannelli, Urbani, NeurIPS 2021]) and architectures (e.g., one-hidden layer networks)

…

Thank you for your attention!