Dynamical mean-field theory for SGD in non-linear neural networks

Francesca Mignacco
Institut de Physique Théorique, CEA Saclay, France

CLARIPHY seminar
Jan 27, 2021

Physics ↔ ML

[Submitted on 25 Mar 2019 (v1), last revised 6 Dec 2019 (this version, v2)]

Machine learning and the physical sciences

Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, Lenka Zdeborová

Machine learning encompasses a broad range of algorithms and modeling tools used for a vast array of data processing tasks, which has entered most scientific disciplines in recent years. We review in a selective way the recent research on the interface between machine learning and physical sciences. This includes conceptual developments in machine learning (ML) motivated by physical insights, applications of machine learning techniques to several domains in physics, and cross-fertilization between the two fields. After giving basic notion of machine learning methods and principles, we describe examples of how statistical physics is used to understand methods in ML. We then move to describe applications of ML methods in particle physics and cosmology, quantum many body physics, quantum computing, and chemical and material physics. We also highlight research and development into novel computing architectures aimed at accelerating ML. In each of the sections we describe recent successes as well as domain-specific methodology and challenges.

In this talk: **Physics → ML**

[Submitted on 25 Mar 2019 (v1), last revised 6 Dec 2019 (this version, v2)]

Machine learning and the physical sciences

Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, Lenka Zdeborová

Machine learning encompasses a broad range of algorithms and modeling tools used for a vast array of data processing tasks, which has entered most scientific disciplines in recent years. We review in a selective way the recent research on the interface between machine learning and physical sciences. This includes conceptual developments in machine learning (ML) motivated by physical insights, applications of machine learning techniques to several domains in physics, and cross-fertilization between the two fields. After giving basic notion of machine learning methods and principles, we describe examples of how statistical physics is used to understand methods in ML. We then move to describe applications of ML methods in particle physics and cosmology, quantum many body physics, quantum computing, and chemical and material physics. We also highlight research and development into novel computing architectures aimed at accelerating ML. In each of the sections we describe recent successes as well as domain-specific methodology and challenges.

Generative model for the data

+

Typical-case scenario

The supervised learning problem:
e.g., binary classification

-1 $+1$ $+1$ -1

Training set

n pre-labelled samples in d dimensions

$+1$ (dog)

-1 (cat)

The supervised learning problem:
e.g., binary classification

-1 $+1$ $+1$ -1

Training set

n pre-labelled samples in d dimensions

$+1$ (dog)

-1 (cat)

New observation: $\mathbf{x} =$, $y = ?$

Test set

The supervised learning problem:
e.g., binary classification

-1 $+1$ $+1$ -1

Training set

n pre-labelled samples in d dimensions

$+1$ (dog)

-1 (cat)

New observation: $\mathbf{x} =$, $y = ?$

Test set

GOAL: Train the weights $\mathbf{w} \in \mathbb{R}^d$ to estimate the new label $\hat{y} = \sigma\left(\mathbf{w}^\top \mathbf{x}\right)$ (single-layer neural network)

In practice:
empirical risk minimization

Empirical risk: $$\mathscr{L}(\mathbf{w}, b) = \sum_{\mu=1}^{n} \ell\left[y_\mu \Phi\left(\mathbf{w}^\top \mathbf{x}_\mu\right)\right] + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

In practice:
empirical risk minimization

Activation function

Empirical risk: $$\mathscr{L}(\mathbf{w}, b) = \sum_{\mu=1}^{n} \ell\left[y_\mu \Phi\left(\mathbf{w}^\top \mathbf{x}_\mu\right)\right] + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

In practice:
empirical risk minimization

Activation function

Empirical risk:

$$\mathscr{L}(\mathbf{w}, b) = \sum_{\mu=1}^{n} \ell\left[y_\mu \Phi\left(\mathbf{w}^\top \mathbf{x}_\mu\right)\right] + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

Loss function evaluated on each sample in the training set $\{(\mathbf{x}_\mu, y_\mu)\}_{\mu=1}^{n}$

In practice:
empirical risk minimization

Activation function

Empirical risk:

$$\mathscr{L}(\mathbf{w}, b) = \sum_{\mu=1}^{n} \ell\left[y_\mu \Phi\left(\mathbf{w}^\top \mathbf{x}_\mu\right)\right] + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

Loss function evaluated on each sample in the training set $\{(\mathbf{x}_\mu, y_\mu)\}_{\mu=1}^{n}$

Ridge regularisation

In practice:
empirical risk minimization

Activation function

Empirical risk: $$\mathscr{L}(\mathbf{w}, b) = \sum_{\mu=1}^{n} \ell\left[y_\mu \Phi\left(\mathbf{w}^\top \mathbf{x}_\mu\right)\right] + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

Loss function evaluated on each sample in the training set $\{(\mathbf{x}_\mu, y_\mu)\}_{\mu=1}^{n}$

Ridge regularisation

Hopefully minimising also the **generalization error** = fraction of mistakes on the **test** set

Gradient descent

Weights update

$$w_j(t+dt) = w_j(t) - dt \left\{ \sum_{\mu=1}^{n} \partial_{w_j} \ell \left[y_\mu \Phi \left(\mathbf{w}^\top \mathbf{x}_\mu + b \right) \right] + \lambda w_j(t) \right\}, \quad \forall j = 1,\ldots d$$

Gradient descent

Weights update

$$w_j(t+dt) = w_j(t) - dt \left\{ \underline{\sum_{\mu=1}^{n} \partial_{w_j} \ell \left[y_\mu \Phi \left(\mathbf{w}^\top \mathbf{x}_\mu + b \right) \right]} + \lambda w_j(t) \right\}, \quad \forall j = 1,\ldots d$$

Computing the gradient on the whole training set at each time step is computationally expensive

Stochastic gradient descent

Weights update

$$w_j(t+dt) = w_j(t) - dt \left\{ \sum_{\mu \in B} \partial_{w_j} \ell \left[y_\mu \Phi \left(\mathbf{w}^\top \mathbf{x}_\mu + b \right) \right] + \lambda w_j(t) \right\}, \quad \forall j = 1,\ldots d$$

Practitioners only compute the gradient on a **subset** of the training set at each step.
In real applications, this turned out to be beneficial for the generalisation performance!

Understanding the dynamics of SGD

Bad Global Minima Exist and SGD Can Reach Them

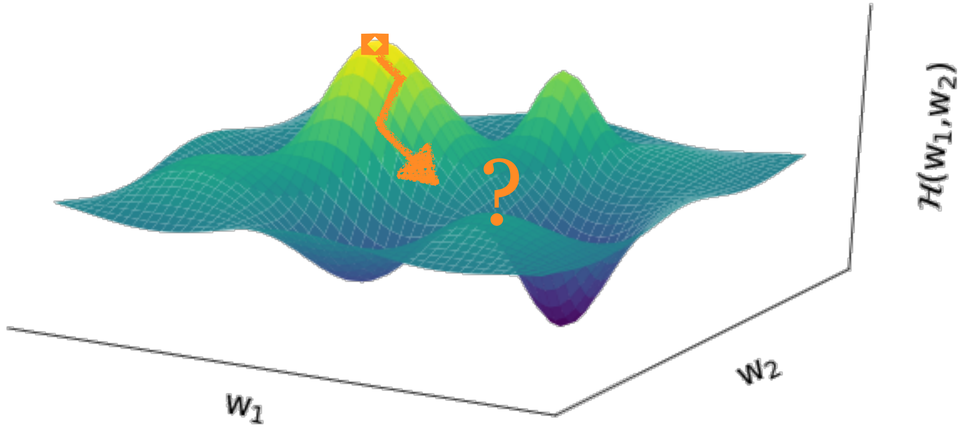
Shengchao Liu, Dimitris Papailiopoulos
University of Wisconsin–Madison

Dimitris Achlioptas
University of California, Santa Cruz

2019

Abstract

Several recent works have aimed to explain why severely overparameterized models, generalize well when trained by Stochastic Gradient Descent (SGD). The emergent consensus explanation has two parts: the first is that there are "no bad local minima", while the second is that SGD performs implicit regularization by having a bias towards low complexity models. We revisit both of these ideas in the context of image classification with common deep neural network architectures. Our first finding is that there exist bad *global* minima, *i.e.*, models that fit the training set perfectly, yet have poor generalization. Our second finding is that given only *unlabeled* training data, we can easily construct initializations that will cause SGD to quickly converge to such bad global minima. For example, on CIFAR, CINIC10, and (Restricted) ImageNet, this can be achieved by starting SGD at a model derived by fitting random labels on the training data: while subsequent SGD training (with the correct labels) will reach zero training error, the resulting model will exhibit a test accuracy degradation of up to 40% compared to training from a random initialization. Finally, we show that regularization seems to provide SGD with an escape route: once heuristics such as data augmentation are used, starting from a complex model (adversarial initialization) has no effect on the test accuracy.



Understanding the dynamics of SGD

Bad Global Minima Exist and SGD Can Reach Them

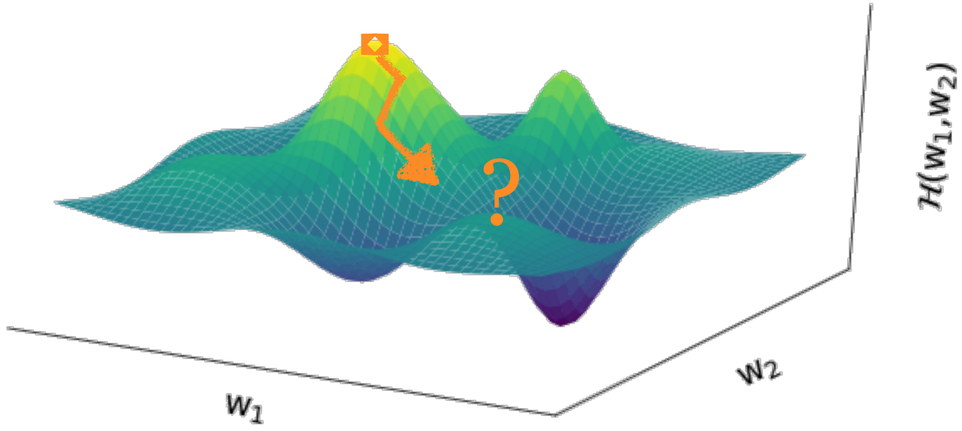
Shengchao Liu, Dimitris Papailiopoulos
University of Wisconsin–Madison

Dimitris Achlioptas
University of California, Santa Cruz

2019

Abstract

Several recent works have aimed to explain why severely overparameterized models, generalize well when trained by Stochastic Gradient Descent (SGD). The emergent consensus explanation has two parts: the first is that there are "no bad local minima", while the second is that SGD performs implicit regularization by having a bias towards low complexity models. We revisit both of these ideas in the context of image classification with common deep neural network architectures. Our first finding is that there exist bad *global* minima, *i.e.*, models that fit the training set perfectly, yet have poor generalization. Our second finding is that given only *unlabeled* training data, we can easily construct initializations that will cause SGD to quickly converge to such bad global minima. For example, on CIFAR, CINIC10, and (Restricted) ImageNet, this can be achieved by starting SGD at a model derived by fitting random labels on the training data: while subsequent SGD training (with the correct labels) will reach zero training error, the resulting model will exhibit a test accuracy degradation of up to 40% compared to training from a random initialization. Finally, we show that regularization seems to provide SGD with an escape route: once heuristics such as data augmentation are used, starting from a complex model (adversarial initialization) has no effect on the test accuracy.



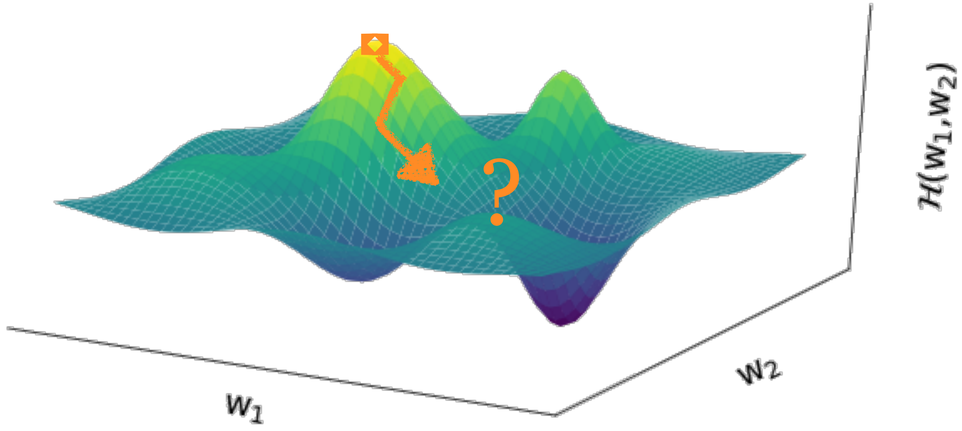
How do gradient-based algorithms manage to achieve good generalisation performance?

Understanding the dynamics of SGD

Related works

(non-exhaustive list)

Modelling the noise of Stochastic Gradient Descent



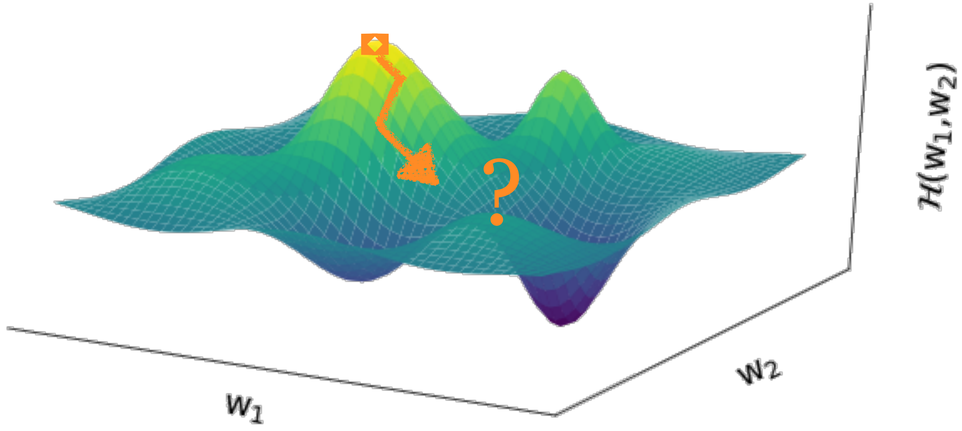
Understanding the dynamics of SGD

Related works

(non-exhaustive list)

Modelling the noise of Stochastic Gradient Descent

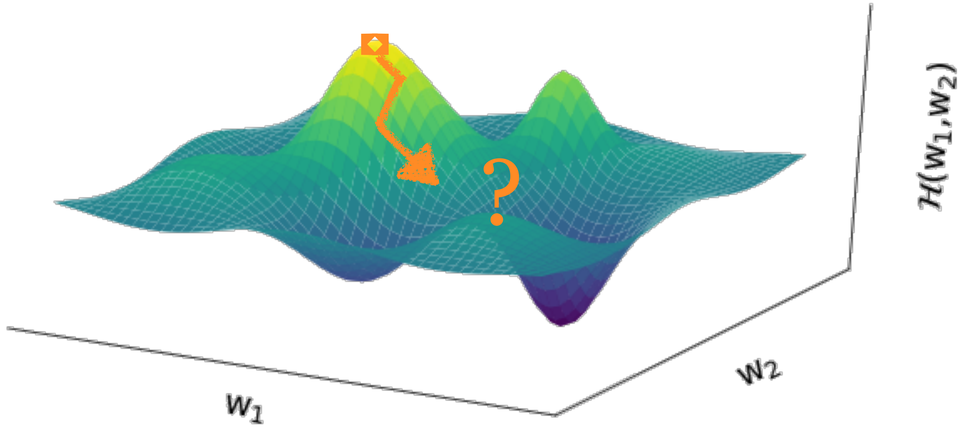
$$w_j(t+dt) = \underbrace{w_j(t) - dt\,\nabla\mathscr{L}(\mathbf{w})}_{\textbf{GD}} + \underbrace{dt\left(\nabla\mathscr{L}(\mathbf{w}) - \tilde{\nabla}_B\mathscr{L}(\mathbf{w})\right)}_{\textbf{Noise}}$$



Understanding the dynamics of SGD

Related works

(non-exhaustive list)



Modelling the noise of Stochastic Gradient Descent

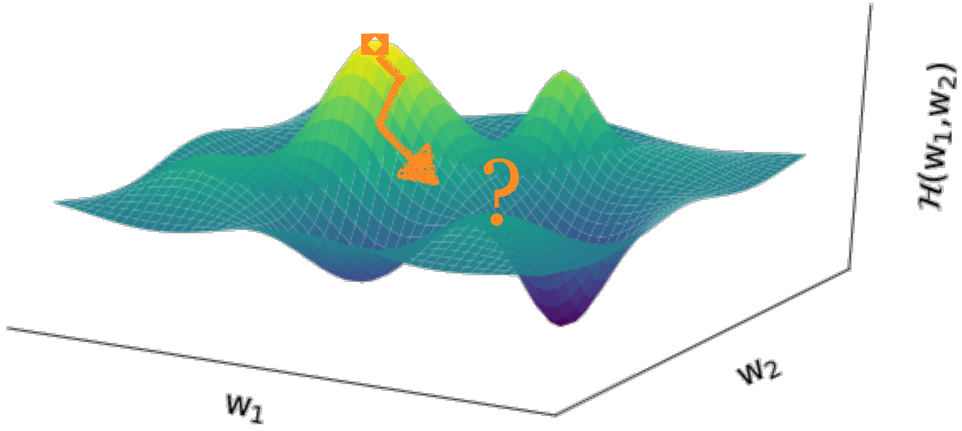
$$w_j(t+dt) = \underbrace{w_j(t) - dt\,\nabla\mathscr{L}(\mathbf{w})}_{\textbf{GD}} + \underbrace{dt\left(\nabla\mathscr{L}(\mathbf{w}) - \tilde{\nabla}_B\mathscr{L}(\mathbf{w})\right)}_{\textbf{Noise}}$$

- **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*]

Understanding the dynamics of SGD

Related works

(non-exhaustive list)



Modelling the noise of Stochastic Gradient Descent

$$w_j(t+dt) = \underbrace{w_j(t) - dt\,\nabla\mathscr{L}(\mathbf{w})}_{\textbf{GD}} + \underbrace{dt\left(\nabla\mathscr{L}(\mathbf{w}) - \tilde{\nabla}_B\mathscr{L}(\mathbf{w})\right)}_{\textbf{Noise}}$$

- **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*]

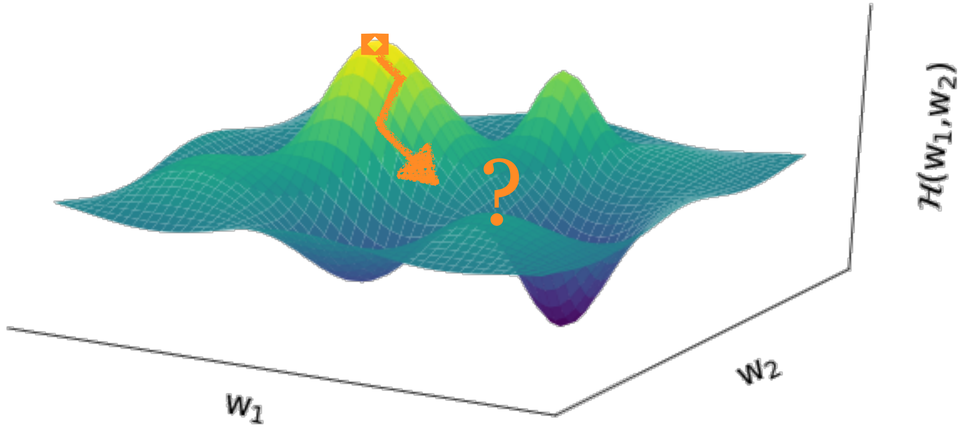
- **Fat-tailed** (Generalized CLT) [*Simsekli et al., '19; Gurbuzbalaban et al., '20; Simsekli et al., '20*]

Understanding the dynamics of SGD

Related works

(non-exhaustive list)

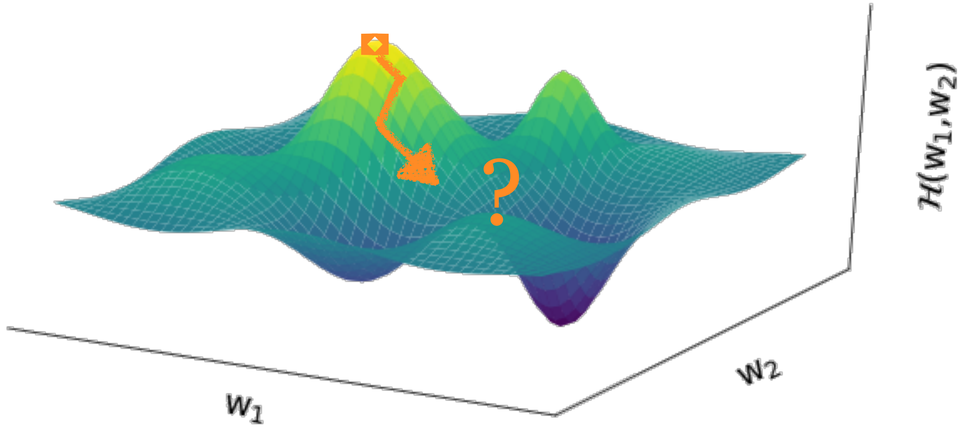
Tracking the whole trajectory of the dynamics



Understanding the dynamics of SGD

Related works

(non-exhaustive list)



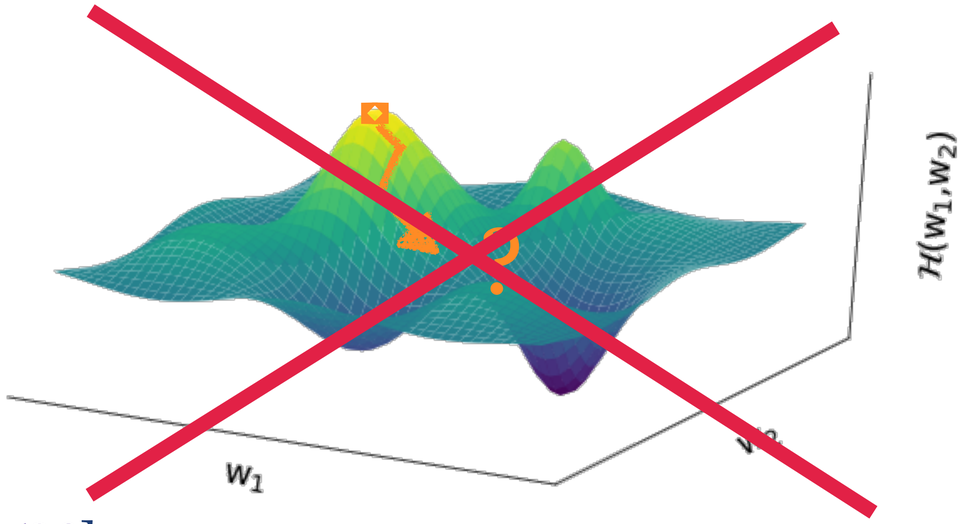
Tracking the whole trajectory of the dynamics

- **Gradient-descent** in **linear** networks [*Bös, Opper, '97 ; Saxe, McClelland, Ganguli, '13; Advani, Saxe, Sompolinksy, '20*]

Understanding the dynamics of SGD

Related works

(non-exhaustive list)



Tracking the whole trajectory of the dynamics

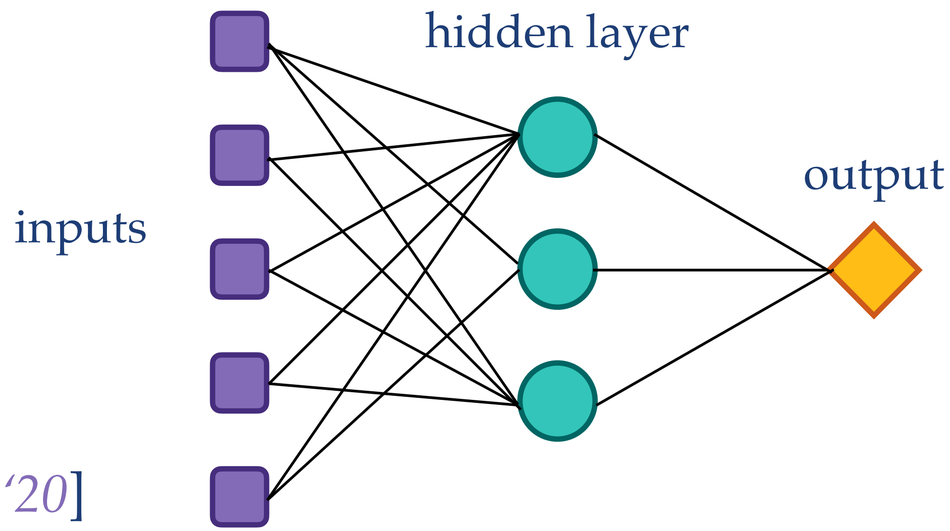
- **Gradient-descent** in **linear** networks [*Bös, Opper, '97 ; Saxe, McClelland, Ganguli, '13; Advani, Saxe, Sompolinksy, '20*]

- **Online** (one-pass) **SGD**

Understanding the dynamics of SGD

Related works

(non-exhaustive list)



Tracking the whole trajectory of the dynamics

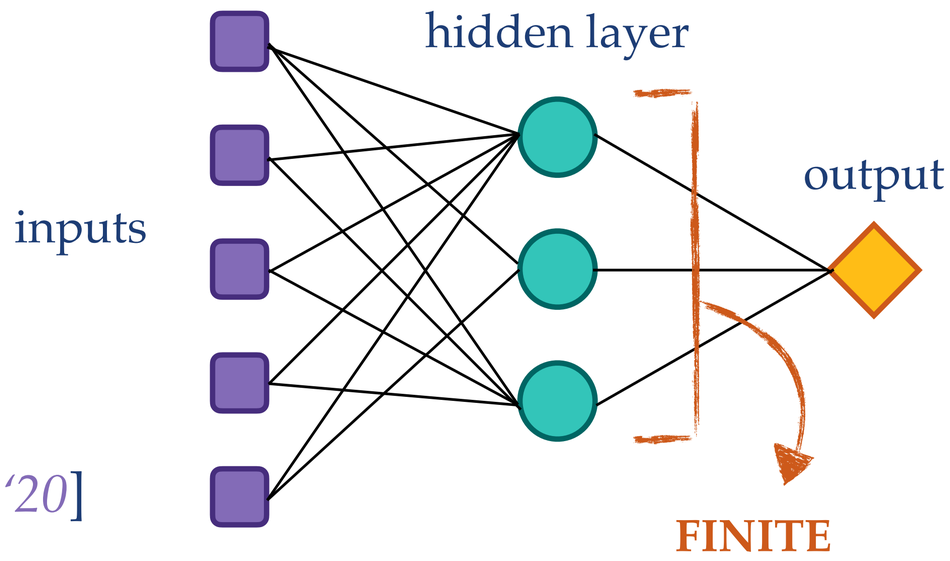
- **Gradient-descent** in **linear** networks [*Bös, Opper, '97 ; Saxe, McClelland, Ganguli, '13; Advani, Saxe, Sompolinksy, '20*]

- **Online** (one-pass) **SGD** in two-layer networks [*in physics: Saad, Solla, '95; Saad, '09; Goldt, Advani, Saxe, Krzakala, Zdeborová, '19; Goldt, Mézard, Krzakala, Zdeborová, '19*]

Understanding the dynamics of SGD

Related works

(non-exhaustive list)



Tracking the whole trajectory of the dynamics

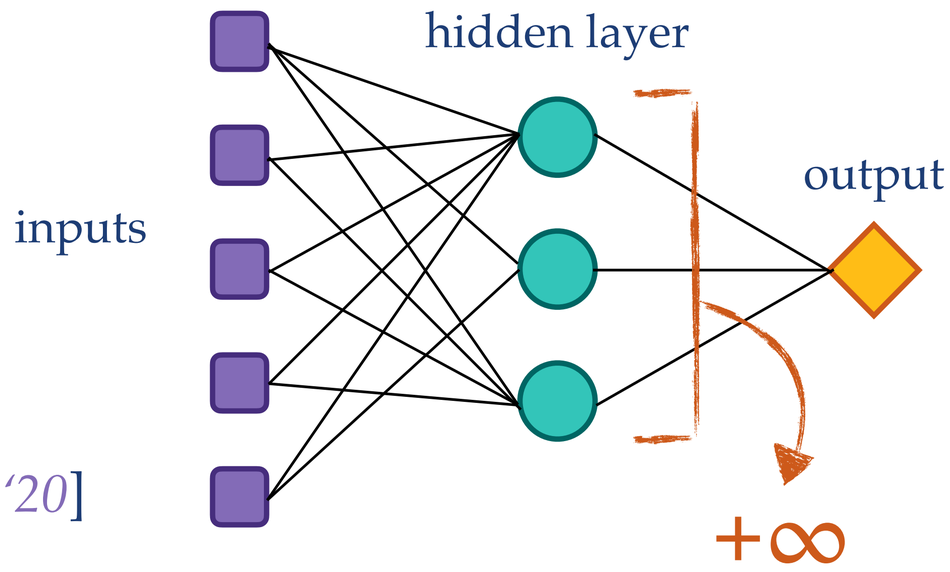
- **Gradient-descent** in **linear** networks [*Bös, Opper, '97 ; Saxe, McClelland, Ganguli, '13; Advani, Saxe, Sompolinksy, '20*]

- **Online** (one-pass) **SGD** in two-layer networks [*in physics: Saad, Solla, '95; Saad, '09; Goldt, Advani, Saxe, Krzakala, Zdeborová, '19; Goldt, Mézard, Krzakala, Zdeborová, '19*]

Understanding the dynamics of SGD

Related works

(non-exhaustive list)



Tracking the whole trajectory of the dynamics

- **Gradient-descent** in **linear** networks [*Bös, Opper, '97 ; Saxe, McClelland, Ganguli, '13; Advani, Saxe, Sompolinksy, '20*]

- **Online** (one-pass) **SGD** in two-layer networks [*in physics: Saad, Solla, '95; Saad, '09; Goldt, Advani, Saxe, Krzakala, Zdeborová, '19; Goldt, Mézard, Krzakala, Zdeborová, '19*]

- **Online SGD** in two-layer networks with **diverging hidden layer** size [*Rotskoff, Vanden-Eijnden, '18; Mei, Montanari, Nguyen, '18; Chizat, Bach, '18; …*]

Understanding the dynamics of SGD

Related works

(non-exhaustive list)

Tracking the whole trajectory of the dynamics

- **Gradient-descent** in **linear** networks [*Bös, Opper, '97 ; Saxe, McClelland, Ganguli, '13; Advani, Saxe, Sompolinksy, '20*]

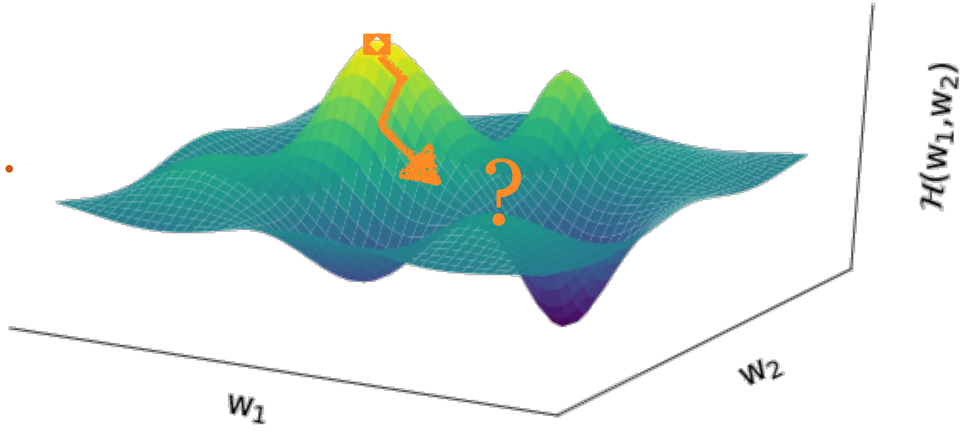
- **Online** (one-pass) **SGD** in two-layer networks [*in physics: Saad, Solla, '95; Saad, '09; Goldt, Advani, Saxe, Krzakala, Zdeborová, '19; Goldt, Mézard, Krzakala, Zdeborová, '19*]

Dynamics of learning with **Direct Feedback Alignment** [*Refinetti, d'Ascoli, Ohana, Goldt, '20*]

Understanding the dynamics of SGD

Our contribution

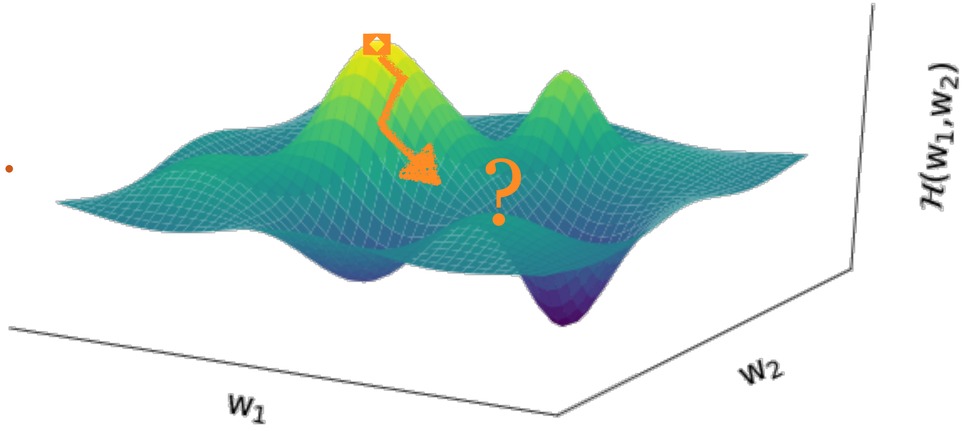
In practice, multi-pass SGD: much less existing theory ...



Understanding the dynamics of SGD

Our contribution

In practice, multi-pass SGD: much less existing theory ...



We analyse in a closed form the learning dynamics of

- **multi-pass** stochastic gradient descent
- trained with a **non-convex** loss function
- in **high dimensions**.

[***FM***, *F. Krzakala, P. Urbani, L. Zdeborová, NeurIPS2020*]

The model for the data:
Gaussian mixture

Data: $\mathbf{x}_\mu = c_\mu \dfrac{\mathbf{v}^*}{\sqrt{d}} + \sqrt{\Delta}\, \mathbf{z}_\mu \in \mathbb{R}^d$

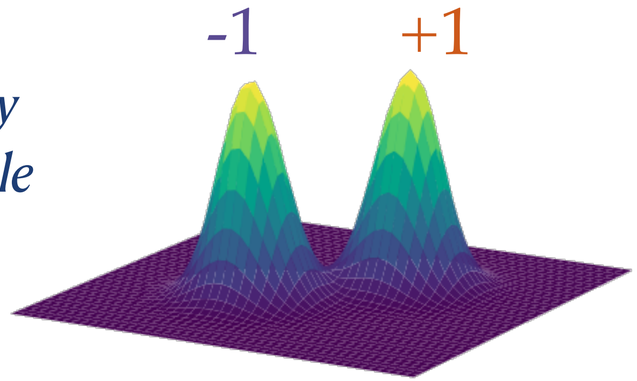
$\mathbf{v}^* \in \mathbb{R}^d, \quad \mathbf{z}_\mu \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}_d\right)$

Labels: $y_\mu = f(c_\mu) \in \{-1, +1\}$,

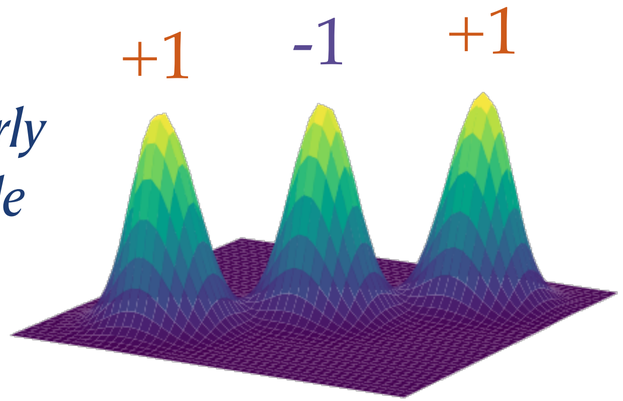
$\mu = 1, \ldots n$

Settings:

Linearly separable



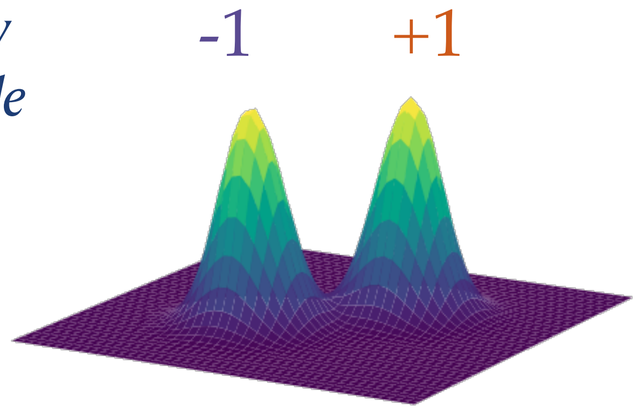
Not linearly separable



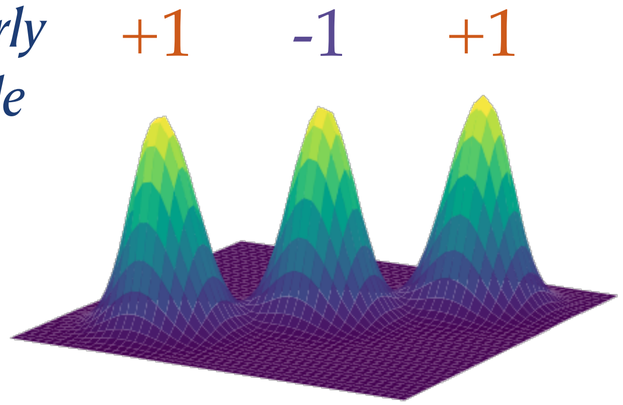
The model for the data:
Gaussian mixture

Settings:

Linearly separable

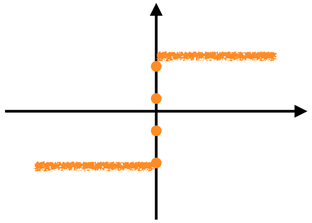


Not linearly separable

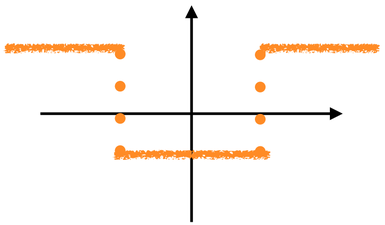


Machines:

$$\hat{y}_\mu(\mathbf{w}) = \mathrm{sgn}\left[\frac{1}{\sqrt{d}}\mathbf{w}^\top \mathbf{x}_\mu\right]$$



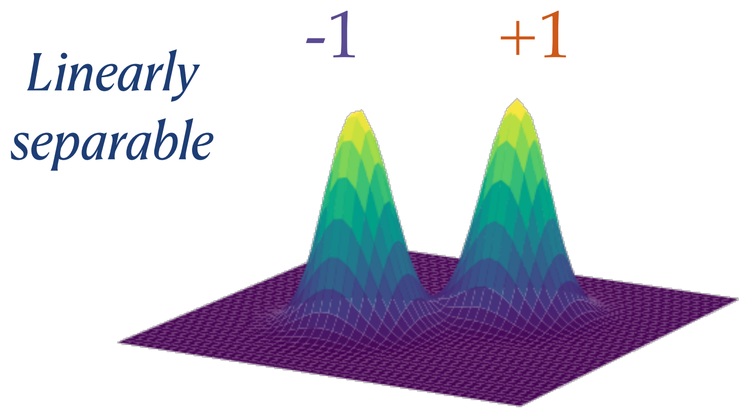
$$\hat{y}_\mu(\mathbf{w}) = \mathrm{sgn}\left[\left(\frac{1}{\sqrt{d}}\mathbf{w}^\top \mathbf{x}_\mu\right)^2 - L^2\right]$$



Asymptotic solution of empirical risk minimisation

- **Linearly separable** setting, $\Phi(x) = x$, and a **convex** cost function ℓ

- High-dimensional regime $d, n \to \infty$ at fixed $\alpha = n/d$



→ **Analytic characterisation** of the **asymptotic performance** of empirical risk minimisation

[*Mai, Liao, Couillet, '19; Deng, Kammoun, Thrampoulidis, '19; **FM**, Lu, Krzakala, Zdeborová, '20*]

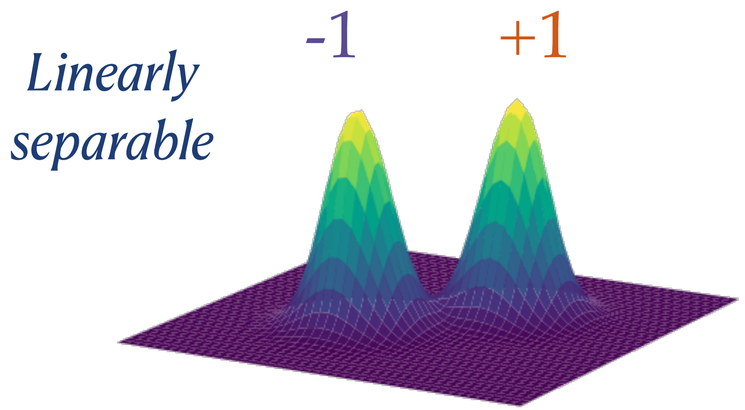
Asymptotic solution of empirical risk minimisation

Alternative methods

Heuristic **replica theory** from statistical physics
[*Mézard, et al., '87; Gardner, '88…*]

or

Rigorous proof using **Gordon's minimax** approach
[*Gordon, '85, '88; Deng, et al., '19…*]



[***FM***, *Lu, Krzakala, Zdeborová, '20*]

Asymptotic solution of empirical risk minimisation

Alternative methods

Heuristic **replica theory** from statistical physics
[*Mézard, et al., '87; Gardner, '88…*]

or

Rigorous proof using **Gordon's minimax** approach
[*Gordon, '85, '88; Deng, et al., '19…*]

Results

- **Closed-form analytic expressions** for the generalisation and training **errors**

$$q = \frac{\hat{q} + \hat{m}^2}{(\lambda + \hat{\gamma})^2}, \quad \hat{q} = \frac{\alpha\Delta}{\gamma^2}\mathbb{E}_{y,s}\left[\left(v(h,y,\gamma) - h\right)^2\right]$$

$$m = \frac{\hat{m}}{\lambda + \hat{\gamma}}, \quad \hat{m} = \frac{\alpha}{\gamma}\mathbb{E}_{y,s}\left[y\left(v(h,y,\gamma) - h\right)\right]$$

$$\gamma = \frac{\Delta}{\lambda + \hat{\gamma}}, \quad \hat{\gamma} = \frac{\alpha\Delta}{\gamma}\left(1 - \mathbb{E}_{y,s}\left[\partial_h v(h,y,\gamma)\right]\right)$$

$$\varepsilon_{\text{gen}} = \rho Q\left(\frac{m+b}{\sqrt{\Delta q}}\right) + (1-\rho) Q\left(\frac{m-b}{\sqrt{\Delta q}}\right) \qquad Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-t^2/2} dt$$

$$\mathrm{L}_{\text{train}} = \lim_{d\to\infty} \frac{\mathcal{L}}{d} = \frac{\lambda}{2} q + \alpha \mathbb{E}_{y,s}\left[\ell\left(v(h,y,\gamma)\right)\right]$$

[***FM**, Lu, Krzakala, Zdeborová, '20*]

Asymptotic solution of empirical risk minimisation

Alternative methods

Heuristic **replica theory** from statistical physics
[*Mézard, et al., '87; Gardner, '88…*]

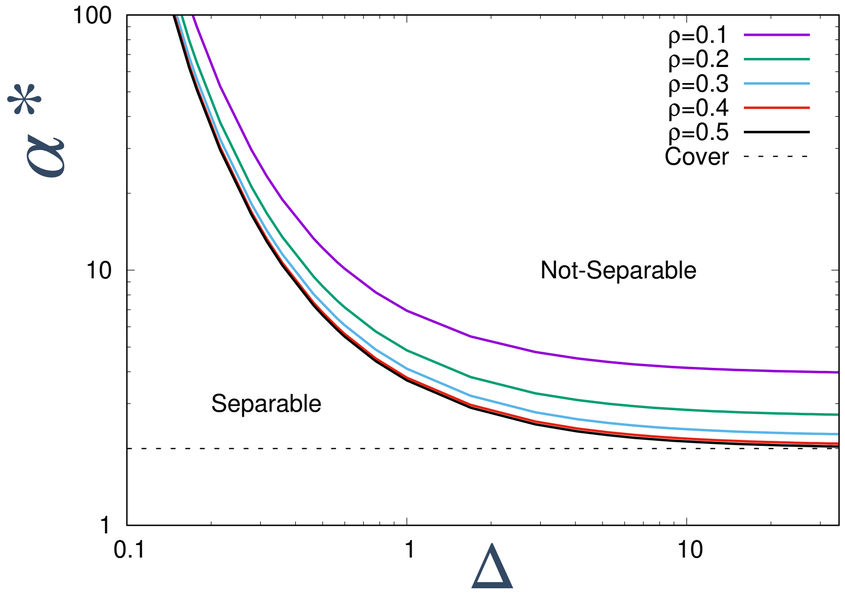
or

Rigorous proof using **Gordon's minimax** approach
[*Gordon, '85, '88; Deng, et al., '19…*]

[***FM**, Lu, Krzakala, Zdeborová, '20*]

Results

- **Closed-form analytic expressions** for the generalisation and training **errors**
- Sharp characterisation of the **data-separability phase transition**



The algorithm

Weight update of multi-pass SGD

$$w_j(t+dt) = w_j(t) - dt \left\{ \sum_{\mu=1}^{n} s_\mu(t)\, \partial_{w_j} \ell \left[y_\mu \Phi\left(\mathbf{w}^\top \mathbf{x}_\mu \right) \right] + \lambda w_j(t) \right\}, \quad \forall j = 1,\ldots d$$

The algorithm

Weight update of multi-pass SGD

$$w_j(t+dt) = w_j(t) - dt \left\{ \sum_{\mu=1}^{n} s_\mu(t)\, \partial_{w_j} \ell \left[y_\mu \Phi\left(\mathbf{w}^\top \mathbf{x}_\mu \right) \right] + \lambda w_j(t) \right\}, \quad \forall j = 1,\ldots d$$

$\in \{0,1\}$ sampling with replacement

batch size $B = bn,\ b \in (0,1]$

The algorithm

Weight update of multi-pass SGD

$$w_j(t+dt) = w_j(t) - dt \left\{ \sum_{\mu=1}^{n} s_\mu(t)\, \partial_{w_j} \ell \left[y_\mu \Phi\left(\mathbf{w}^\top \mathbf{x}_\mu \right) \right] + \lambda w_j(t) \right\}, \quad \forall j = 1,\ldots d$$

$\in \{0,1\}$ sampling with replacement
batch size $B = bn,\ b \in (0,1]$

Continuous-time limit $\eta \to 0$? **Persistent-SGD algorithm**

The algorithm

Weight update of multi-pass SGD

$$w_j(t+dt) = w_j(t) - dt \left\{ \sum_{\mu=1}^{n} s_\mu(t)\, \partial_{w_j} \ell \left[y_\mu \Phi\left(\mathbf{w}^\top \mathbf{x}_\mu \right) \right] + \lambda w_j(t) \right\}, \quad \forall j = 1,\ldots d$$

$\in \{0,1\}$ sampling with replacement
batch size $B = bn,\ b \in (0,1]$

Continuous-time limit $\eta \to 0$? **Persistent-SGD algorithm**

Persistence time τ = average time that each sample spends in the training mini batch

The algorithm

Continuous-time limit: Stochastic gradient flow

$$\dot{\mathrm{w}}_j(t) = -\lambda \mathrm{w}_j(t) - \sum_{\mu=1}^{n} s_\mu(t)\, \partial_{w_j} \ell \left[y_\mu \Phi\left(\mathbf{w}^\top \mathbf{x}_\mu \right) \right], \quad j = 1,\ldots d$$

Continuous-time limit $\eta \to 0$? **Persistent-SGD algorithm**

Persistence time τ = average time that each sample spends in the training mini batch

DMFT

Track the dynamics of disordered systems in the infinite dimensional limit at tractably large times

[*Sompolinsky, Zippelius, '81; Mézard, Parisi, Virasoro, '87; Sompolinsky, Crisanti, Sommers, '88; Cugliandolo, Kurchan, '93; Georges, Kotliar, Krauth, Rozenberg, '96; …*]

DMFT

Track the dynamics of disordered systems in the infinite dimensional limit at tractably large times

Bipartite model: *E. Agoritsas, G. Biroli, P. Urbani, F. Zamponi, '18*

We extend this result to include:

[*Sompolinsky, Zippelius, '81; Mézard, Parisi, Virasoro, '87; Sompolinsky, Crisanti, Sommers, '88; Cugliandolo, Kurchan, '93; Georges, Kotliar, Krauth, Rozenberg, '96; …*]

DMFT

Track the dynamics of disordered systems in the infinite dimensional limit at tractably large times

Bipartite model: *E. Agoritsas, G. Biroli, P. Urbani, F. Zamponi, '18*

We extend this result to include:

1. Multi-pass SGD

[*Sompolinsky, Zippelius, '81; Mézard, Parisi, Virasoro, '87; Sompolinsky, Crisanti, Sommers, '88; Cugliandolo, Kurchan, '93; Georges, Kotliar, Krauth, Rozenberg, '96; …*]

DMFT

Track the dynamics of disordered systems in the infinite dimensional limit at tractably large times

Bipartite model: *E. Agoritsas, G. Biroli, P. Urbani, F. Zamponi, '18*

We extend this result to include:

1. Multi-pass SGD
2. Structured data

[*Sompolinsky, Zippelius, '81; Mézard, Parisi, Virasoro, '87; Sompolinsky, Crisanti, Sommers, '88; Cugliandolo, Kurchan, '93; Georges, Kotliar, Krauth, Rozenberg, '96; …*]

DMFT

$d, n \to \infty$, at fixed $\alpha = n/d, b, \tau$

Markovian dynamics of a strongly coupled system of $d \to \infty$ degrees of freedom

dimensional reduction

Non-Markovian dynamics of one degree of freedom with memory

Stochastic gradient flow

$$\dot{\mathrm{w}}_j(t) = -\lambda \mathrm{w}_j(t) - \sum_{\mu=1}^{n} s_\mu(t)\, \partial_{w_j} \ell \left[y_\mu \Phi\left(\mathbf{w}^\top \mathbf{x}_\mu\right)\right], \quad j = 1,\ldots d$$

Typical gap: $h(t)$ related to $\mathbf{w}(t)^\top \mathbf{z}_\mu \big/ \sqrt{d}$

Magnetisation: $m(t) = \mathbf{w}(t)^\top \mathbf{v}^* \big/ d$

DMFT: the equations

$$\partial_t h(t) = -\left(\lambda + \hat{\lambda}(t)\right) h(t) - \sqrt{\Delta}\, s(t)\, \Lambda'\left(y(c), r(t) - Y(t)\right) + \int_0^t \mathrm{d}t'\, M_R(t,t') h(t') + \xi(t)$$

$$\partial_t m(t) = -\lambda m(t) - \mu(t), \qquad m(0) = 0^+$$

Memory kernels & auxiliary functions

$$\mu(t) = \alpha \left\langle s(t)\left(c + \sqrt{\Delta} h_0\right) \Lambda'\left(y(c), r(t)\right) \right\rangle,$$

$$\hat{\lambda}(t) = \alpha\Delta \left\langle s(t) \Lambda''\left(y(c), r(t)\right) \right\rangle,$$

$$M_C(t,t') = \alpha\Delta \left\langle s(t)s(t')\Lambda'\left(y(c), r(t)\right) \Lambda'\left(y(c), r(t')\right) \right\rangle,$$

$$M_R(t,t') = \alpha\Delta \frac{\delta}{\delta Y(t')} \left\langle s(t)\Lambda'\left(y(c), r(t)\right) \right\rangle \Big|_{Y=0}$$

DMFT: the equations

$$\partial_t h(t) = -\left(\lambda + \hat{\lambda}(t)\right) h(t) - \sqrt{\Delta}\, s(t)\, \Lambda'\left(y(c), r(t) - Y(t)\right) + \int_0^t \mathrm{d}t'\, M_R(t,t') h(t') + \xi(t)$$

Effective stochastic process $h(t)$ related to $\mathbf{w}(t)^\top \mathbf{z}_\mu \big/ \sqrt{d}$

$$\partial_t m(t) = -\lambda m(t) - \mu(t), \qquad m(0) = 0^+$$

Memory kernels & auxiliary functions

$$\mu(t) = \alpha \left\langle s(t)\left(c + \sqrt{\Delta} h_0\right) \Lambda'\left(y(c), r(t)\right) \right\rangle,$$

$$\hat{\lambda}(t) = \alpha\Delta \left\langle s(t) \Lambda''\left(y(c), r(t)\right) \right\rangle,$$

$$M_C(t,t') = \alpha\Delta \left\langle s(t)s(t')\Lambda'\left(y(c), r(t)\right) \Lambda'\left(y(c), r(t')\right) \right\rangle,$$

$$M_R(t,t') = \alpha\Delta \frac{\delta}{\delta Y(t')} \left\langle s(t)\Lambda'\left(y(c), r(t)\right) \right\rangle \Big|_{Y=0}$$

DMFT: the equations

$$\partial_t h(t) = -\left(\lambda + \hat{\lambda}(t)\right) h(t) - \sqrt{\Delta}\, s(t)\, \Lambda'\left(y(c), r(t) - Y(t)\right) + \int_0^t \mathrm{d}t'\, M_R(t,t') h(t') + \xi(t)$$

$$\boxed{\partial_t m(t) = -\lambda m(t) - \mu(t), \qquad m(0) = 0^+}$$

ODE for the *magnetisation*

$$m(t) = \mathbf{w}(t)^\top \mathbf{z}_\mu \Big/ d$$

Memory kernels & auxiliary functions

$$\mu(t) = \alpha \left\langle s(t)\left(c + \sqrt{\Delta} h_0\right) \Lambda'\left(y(c), r(t)\right) \right\rangle,$$

$$\hat{\lambda}(t) = \alpha\Delta \left\langle s(t) \Lambda''\left(y(c), r(t)\right) \right\rangle,$$

$$M_C(t,t') = \alpha\Delta \left\langle s(t)s(t')\Lambda'\left(y(c), r(t)\right) \Lambda'\left(y(c), r(t')\right) \right\rangle,$$

$$M_R(t,t') = \alpha\Delta \frac{\delta}{\delta Y(t')} \left\langle s(t)\Lambda'\left(y(c), r(t)\right) \right\rangle \Big|_{Y=0}$$

DMFT: the equations

$$\partial_t h(t) = -\left(\lambda + \hat{\lambda}(t)\right) h(t) - \sqrt{\Delta}\, s(t)\, \Lambda'\left(y(c), r(t) - Y(t)\right) + \int_0^t \mathrm{d}t'\, M_R(t,t') h(t') + \xi(t)$$

$$\partial_t m(t) = -\lambda m(t) - \mu(t), \qquad m(0) = 0^+$$

Memory kernels & auxiliary functions

$$\mu(t) = \alpha \left\langle s(t)\left(c + \sqrt{\Delta} h_0\right) \Lambda'\left(y(c), r(t)\right) \right\rangle,$$

$$\hat{\lambda}(t) = \alpha\Delta \left\langle s(t) \Lambda''\left(y(c), r(t)\right) \right\rangle,$$

$$M_C(t,t') = \alpha\Delta \left\langle s(t)s(t')\Lambda'\left(y(c), r(t)\right) \Lambda'\left(y(c), r(t')\right) \right\rangle,$$

$$M_R(t,t') = \alpha\Delta \frac{\delta}{\delta Y(t')} \left\langle s(t)\Lambda'\left(y(c), r(t)\right) \right\rangle \Big|_{Y=0}$$

The stochastic equation must be solved self-consistently. [*Eissfeller, Opper, '97; Roy et al., '19; Manacorda et al., '20*]

DMFT: the effective process

$$\partial_t h(t) = -\left(\lambda + \hat{\lambda}(t)\right) h(t) - \sqrt{\Delta}\, s(t)\, \Lambda'\left(y(c), r(t) - Y(t)\right) + \int_0^t \mathrm{d}t'\, M_R(t,t') h(t') + \xi(t)$$

$$h(0) \sim \mathcal{N}(0,R)$$

$$\langle \xi(t) \rangle = 0$$

$$\langle \xi(t)\xi(t') \rangle = M_C(t,t')$$

DMFT: the effective process

$$\partial_t h(t) = -\left(\lambda + \hat{\lambda}(t)\right) h(t) - \sqrt{\Delta}\, s(t)\, \Lambda'\left(y(c), r(t) - Y(t)\right) + \int_0^t \mathrm{d}t'\, M_R(t,t') h(t') + \xi(t)$$

$$\underbrace{h(0) \sim \mathcal{N}(0,R)}_{\text{Random initial condition}}$$

$$\langle \xi(t) \rangle = 0$$

$$\langle \xi(t)\xi(t') \rangle = M_C(t,t')$$

DMFT: the effective process

$$\partial_t h(t) = -\overbrace{\left(\lambda + \hat{\lambda}(t)\right)}^{\text{Regularization}} h(t) - \sqrt{\Delta}\, s(t)\, \Lambda'\left(y(c), r(t) - Y(t)\right) + \int_0^t \mathrm{d}t'\, M_R(t,t') h(t') + \xi(t)$$

$$\underbrace{h(0) \sim \mathcal{N}(0,R)}_{\text{Random initial condition}}$$

$$\langle \xi(t) \rangle = 0$$

$$\langle \xi(t)\xi(t') \rangle = M_C(t,t')$$

DMFT: the effective process

$$\partial_t h(t) = -\overbrace{\left(\lambda + \hat{\lambda}(t)\right) h(t)}^{\text{Regularization}} - \overbrace{\sqrt{\Delta}\, s(t)\, \Lambda'\left(y(c), r(t) - Y(t)\right)}^{\text{Effective stochastic gradient}} + \int_0^t \mathrm{d}t'\, M_R(t,t') h(t') + \xi(t)$$

$$\underbrace{h(0) \sim \mathcal{N}(0,R)}_{\text{Random initial condition}}$$

$$\langle \xi(t) \rangle = 0$$

$$\langle \xi(t)\xi(t') \rangle = M_C(t,t')$$

DMFT: the effective process

$$\partial_t h(t) = -\overbrace{\left(\lambda + \hat{\lambda}(t)\right) h(t)}^{\text{Regularization}} - \overbrace{\sqrt{\Delta}\, s(t)\, \Lambda'\left(y(c), r(t) - Y(t)\right)}^{\text{Effective stochastic gradient}} + \overbrace{\int_0^t \mathrm{d}t'\, M_R(t,t') h(t')}^{\text{Memory term}} + \xi(t)$$

$$\underbrace{h(0) \sim \mathcal{N}(0,R)}_{\text{Random initial condition}}$$

$$\langle \xi(t) \rangle = 0$$

$$\langle \xi(t)\xi(t') \rangle = M_C(t,t')$$

DMFT: the effective process

$$\partial_t h(t) = -\overbrace{\left(\lambda + \hat{\lambda}(t)\right) h(t)}^{\text{Regularization}} - \overbrace{\sqrt{\Delta}\, s(t)\, \Lambda'\left(y(c), r(t) - Y(t)\right)}^{\text{Effective stochastic gradient}} + \overbrace{\int_0^t \mathrm{d}t'\, M_R(t,t') h(t')}^{\text{Memory term}} + \xi(t)$$

$$\underbrace{h(0) \sim \mathcal{N}(0,R)}_{\text{Random initial condition}}$$

$$\boxed{\begin{aligned} \langle \xi(t) \rangle &= 0 \\ \langle \xi(t)\xi(t') \rangle &= M_C(t,t') \end{aligned}}$$

Gaussian effective noise

DMFT: the equations

✔ Correlation $C(t,t') = \frac{1}{d}\mathbf{w}(t)^\top \mathbf{w}(t')$ and response $R(t,t') = \frac{1}{d}\sum_{i=1}^{d} \frac{\delta \mathrm{w}_i(t)}{\delta H_i(t')}$ functions:

$$\partial_t C(t',t) = -\left(\lambda + \hat{\lambda}(t)\right) C(t,t') + \int_0^t \mathrm{d}s\, M_R(t,s) C(t',s) + \int_0^{t'} \mathrm{d}s\, M_C(t,s) R(t',s) - m(t')\left(\int_0^t \mathrm{d}s\, M_R(t,s) m(s) + \mu(t) - \hat{\lambda}(t) m(t)\right) \quad \text{if } t \neq t',$$

$$\frac{1}{2}\partial_t C(t,t) = -\left(\lambda + \hat{\lambda}(t)\right) C(t,t) + \int_0^t \mathrm{d}s\, M_R(t,s) C(t,s) + \int_0^t \mathrm{d}s\, M_C(t,s) R(t,s) - m(t)\left(\int_0^t \mathrm{d}s\, M_R(t,s) m(s) + \mu(t) - \hat{\lambda}(t) m(t)\right),$$

$$\partial_t R(t',t) = -\left(\lambda + \hat{\lambda}(t)\right) R(t,t') + \delta(t-t') + \int_{t'}^t \mathrm{d}s\, M_R(t,s) R(s,t')\,.$$

DMFT: the equations

✔ Correlation $C(t,t') = \frac{1}{d}\mathbf{w}(t)^\top \mathbf{w}(t')$ and response $R(t,t') = \frac{1}{d}\sum_{i=1}^{d} \frac{\delta \mathrm{w}_i(t)}{\delta H_i(t')}$ functions;

✔ Generalization error:

$$\varepsilon_{\text{gen}}(t) \equiv \frac{1}{4}\mathbb{E}_{\mathbf{X},\mathbf{y},\mathbf{x}_{\text{new}},y_{\text{new}}}\left[\left(\hat{y}_{\text{new}} - y_{\text{new}}\right)^2\right] = \begin{cases} \frac{1}{2}\text{erfc}\left(\frac{m(t)}{\sqrt{2\Delta\, C(t,t)}}\right) & \text{Linearly separable} \\ \frac{1}{2}\text{erfc}\left(\frac{L}{\sqrt{2\Delta\, C(t,t)}}\right) + \frac{1}{4}\left(\text{erf}\left(\frac{L-m(t)}{\sqrt{2\Delta\, C(t,t)}}\right) + \text{erf}\left(\frac{L+m(t)}{\sqrt{2\Delta\, C(t,t)}}\right)\right) & \text{Not linearly separable} \end{cases}$$

DMFT: the equations

- ✔ Correlation $C(t,t') = \frac{1}{d}\mathbf{w}(t)^\top \mathbf{w}(t')$ and response $R(t,t') = \frac{1}{d}\sum_{i=1}^{d} \frac{\delta \mathrm{w}_i(t)}{\delta H_i(t')}$ functions;
- ✔ Generalization error;
- ✔ Training loss:

$$e(t) = \alpha \left\langle \ell\left(y\Phi\left(r(t)\right)\right)\right\rangle$$

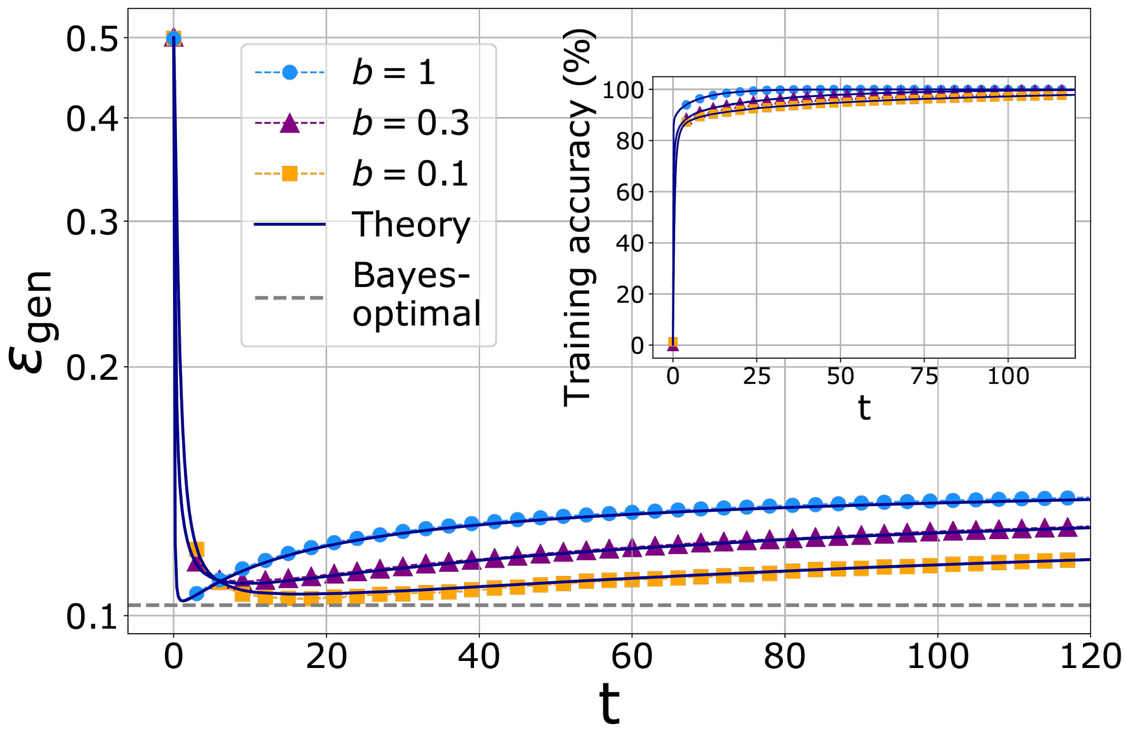
DMFT: the equations

- ✔ Correlation $C(t,t') = \frac{1}{d}\mathbf{w}(t)^\top \mathbf{w}(t')$ and response $R(t,t') = \frac{1}{d}\sum_{i=1}^{d} \frac{\delta \mathrm{w}_i(t)}{\delta H_i(t')}$ functions;
- ✔ Generalization error;
- ✔ Training loss;
- ✔ Training accuracy:

$$a(t) = 1 - \left\langle \theta\left(-y\Phi\left(r(t)\right)\right)\right\rangle$$

Some results*

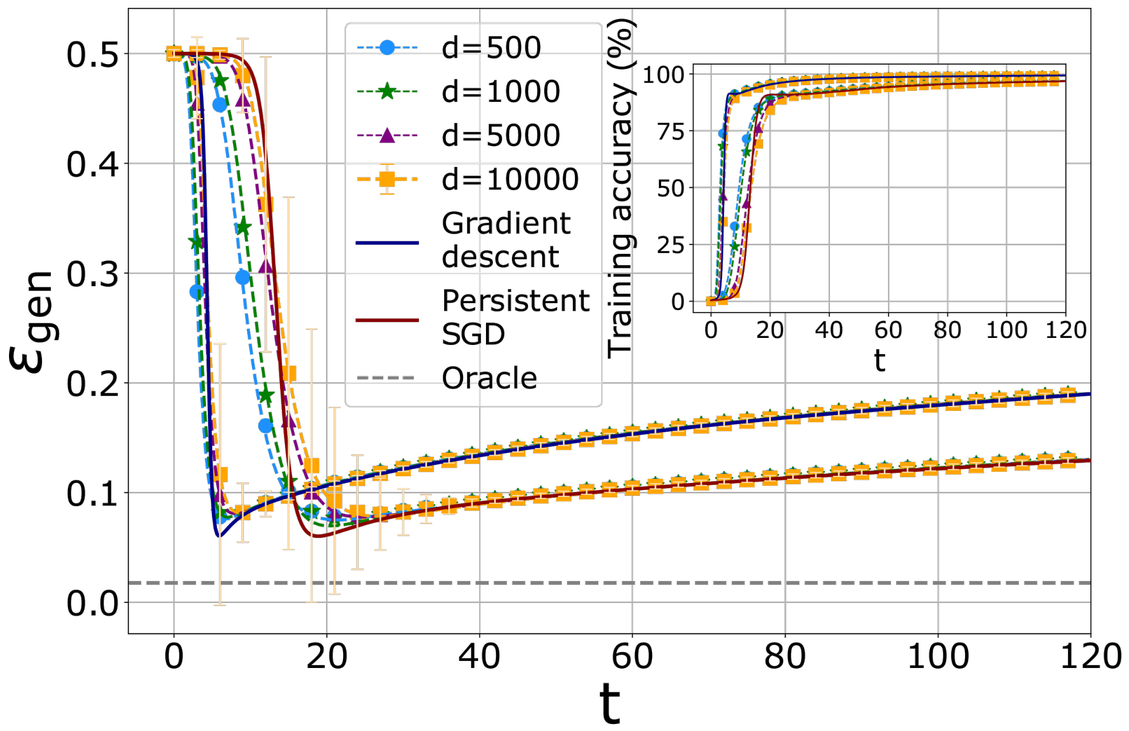
Two-cluster model



$\lambda = 0, \alpha = 2, \Delta = 0.5, 1/\tau = 0.6, \eta = 0.2, R = 0.01$

simulations at $d = 500$

Three-cluster model



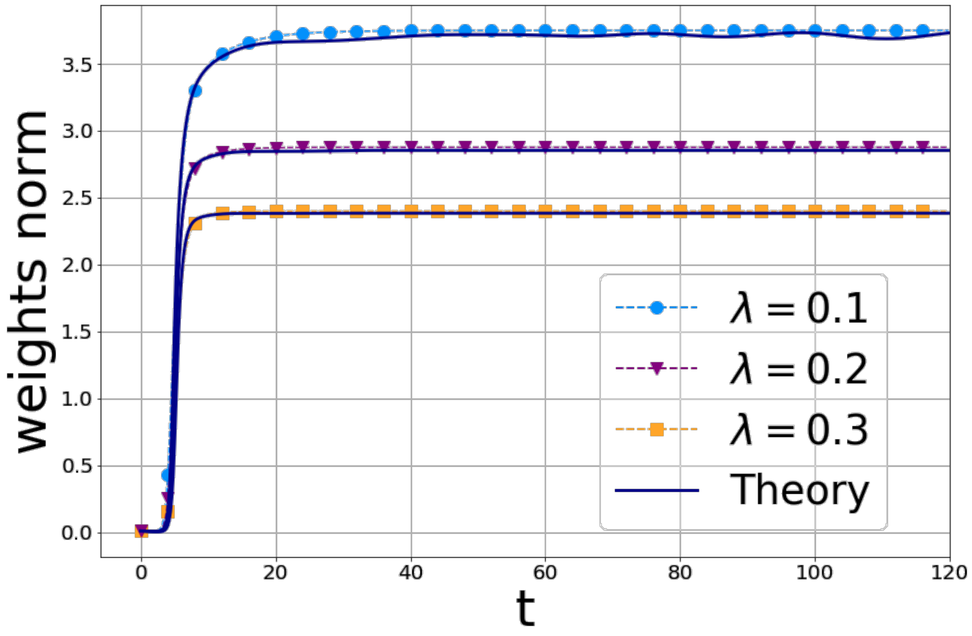
$\lambda = 0, \alpha = 2, \Delta = 0.05, L = 0.7, \eta = 0.2, R = 1$

* using the logistic loss: $\ell(v) = \ln\left(1 + e^{-v}\right)$

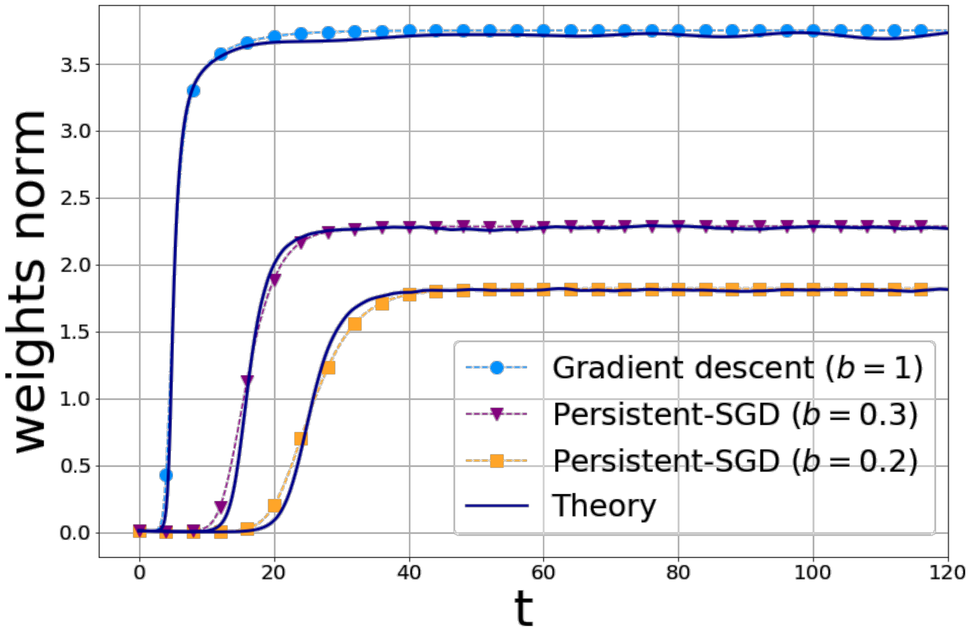
Some results

Three-cluster model

Increasing regularization:



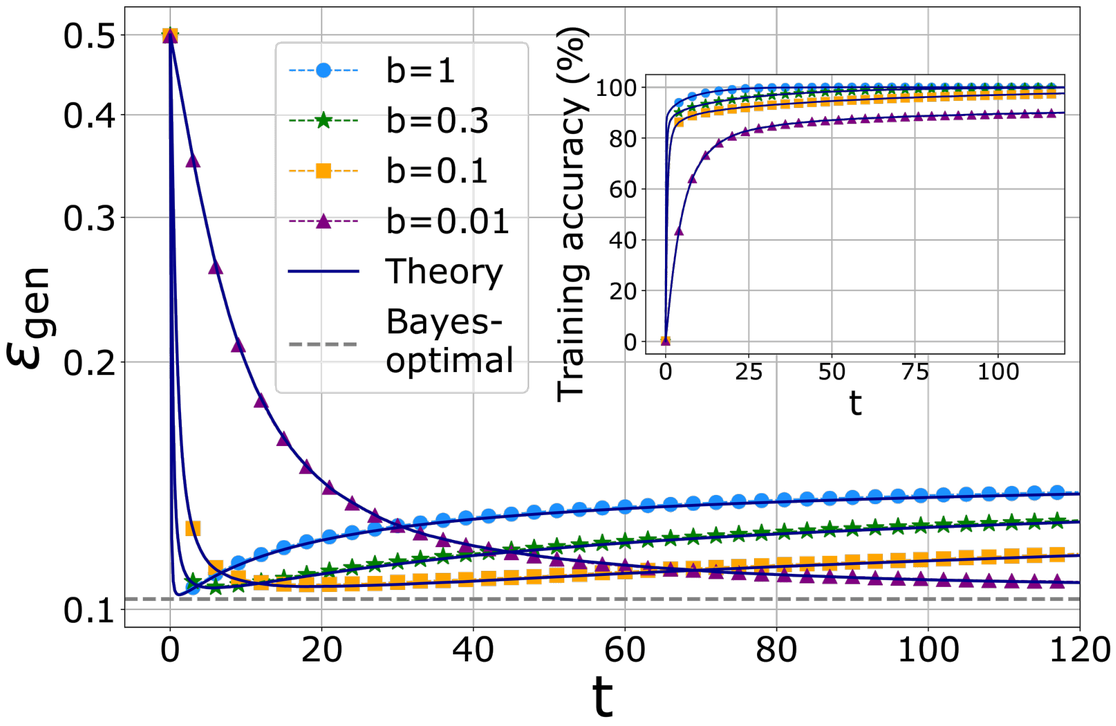
Decreasing batch size:

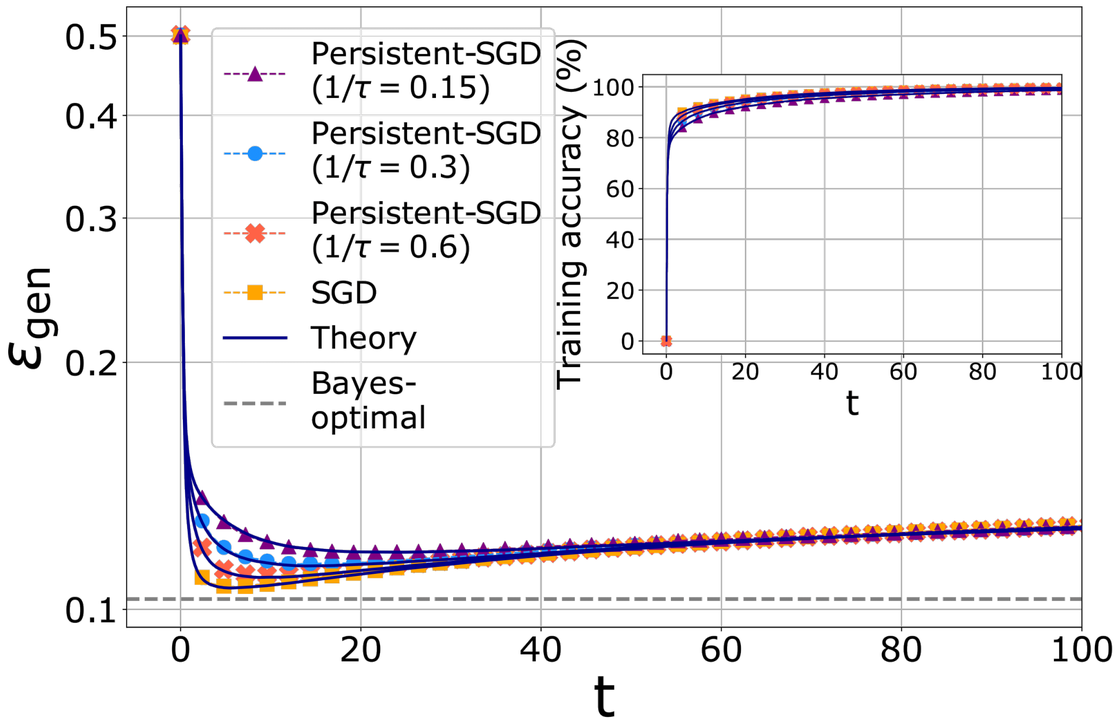


The mini batch acts as an effective regularization.

Some results

Two-cluster model





As τ gets smaller, Persistent-SGD approaches SGD.

Conclusion

DMFT equations track the dynamics of multi-pass SGD in infinite dimensions.

$\longrightarrow$ no finite size effects

Perspectives

- Gaining more insights from DMFT (optimal hyper-parameter setting, nature of SGD noise…).

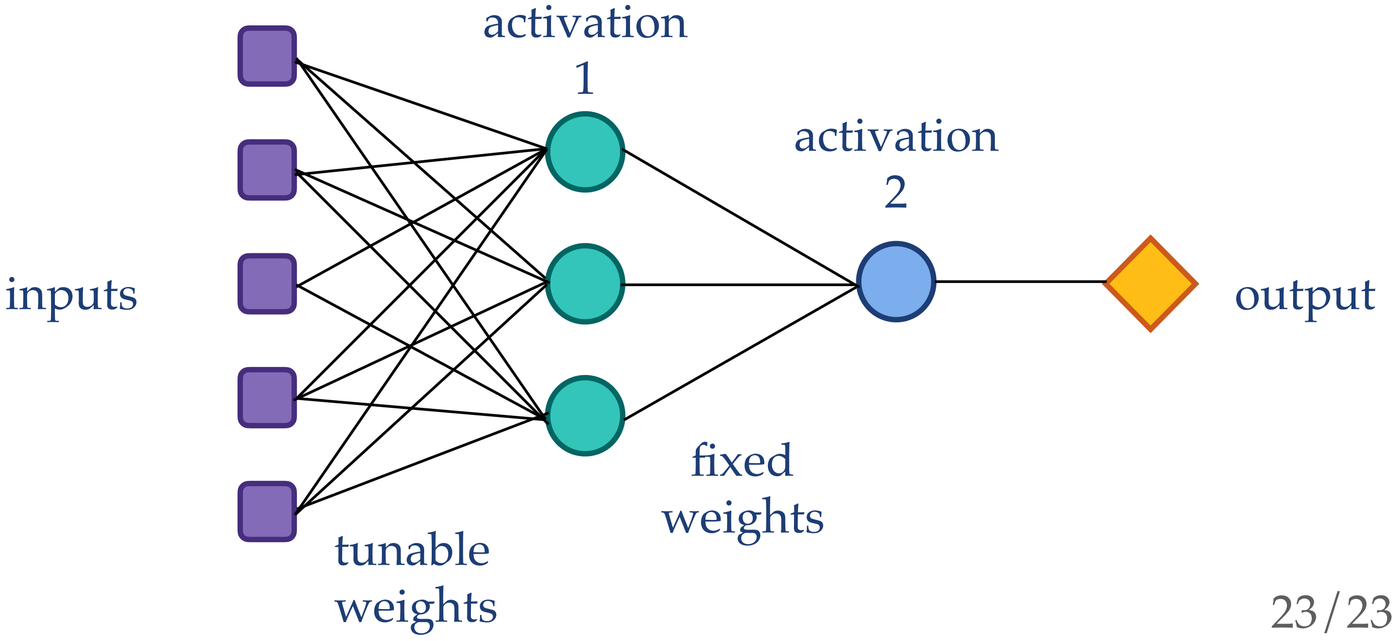
Perspectives

- Gaining more insights from DMFT (optimal hyper-parameter setting, nature of SGD noise…).

- Extension to other data models, networks with hidden units, variants of the training algorithm.

e.g., Committee machine:

[*Monasson, Zecchina, '95 ; Schwarze, '93 ; Schwarze, Hertz, '92, '93 ; Aubin, Maillard, Barbier, Krzakala, Macris, Zdeborová, '19; …*]



Perspectives

- Gaining more insights from DMFT (optimal hyper-parameter setting, nature of SGD noise…).
- Extension to other data models, networks with hidden units, variants of the training algorithm.
- Rigorous proof of the equations. [*related: Ben Arous, Dembo, Guionnet, '06; Zamfir '08; Dembo, Subag, '19*]

Thank you for your attention !