Online Learning and Information Exponents: On The Importance of Batch size, and Time / Complexity Tradeoffs

Luca Arnaboldi[1], Yatin Dandi[1,3], Florent Krzakala[1] Bruno Loureiro[2], Luca Pesce[1], Ludovic Stephan[1]
[1] IdePHICS Labrador, EPFL; [2] ENS Paris; [3] SPOC Labrador, EPFL;

Context & Motivation

The study of training dynamics of *single-index multi-index* models of Gaussian data has been developed recently, leading to the following highlighted results:

- For *single-index models*, the sample complexity of one-pass SGD is determined by the first non-zero Hermite coefficient of the target, also known as the *information exponent* ℓ [1];
- *multi-index model* present a richer behavior in terms of possible dynamics, but the *leap exponent* is determing the sample complexity for escaping the initialization;
- [2] showed that batch sizes $n_b \sim d^\ell$ can improve the number of time steps needed to recover hidden directions up to costant $O_d(1)$.

Aim: *Studying the effect of large batch size in terms of gradient steps needed to recover the target.*

Setting

The exact model we are going to study is the following:

- Input data is generated from independent Gaussian distributions:
$$z \sim \mathcal{N}(0, I_d)$$
Labels are generated by
$$y = f^\star(\boldsymbol{z}) + \sqrt{\Delta}\xi = h^\star(W^\star \boldsymbol{z}) + \sqrt{\Delta}\xi$$
where Δ is the artificial noise.
- We are training a two-layer network with **square activations**:
$$f(\boldsymbol{z}) = \frac{1}{p}\sum_{j=1}^{p} a_j \sigma(\langle \boldsymbol{z}, \boldsymbol{w}_j \rangle)$$
- In most of the cases we are using the **square loss function**
$$\ell(y^\nu, f(z^\nu)) = \frac{1}{2}(y^\nu - f(z^\nu))^2,$$
or whe specified the **correlation loss function**
$$\ell(y^\nu, f(z^\nu)) = 1 - y^\nu f(z^\nu).$$
- We the **projected online SGD** with batch-size n_b:
$$\boldsymbol{g}_j = \frac{1}{n_b}\sum_{\nu=1}^{n_b} \nabla_{\boldsymbol{w}_j} \ell(y^\nu, f(z^\nu)) \qquad \boldsymbol{w}_{j,t+1} = \frac{\boldsymbol{w}_{j,t} - \gamma \boldsymbol{g}_j}{\|\boldsymbol{w}_{j,t} - \gamma \boldsymbol{g}_j\|}$$
meaning that the samples in a batch are used for one single gradient step, and discarded after that.

High dimensional limit

We study the limit where the data dimension is going to infity $d \to +\infty$. Together with the dimension, we also scale:

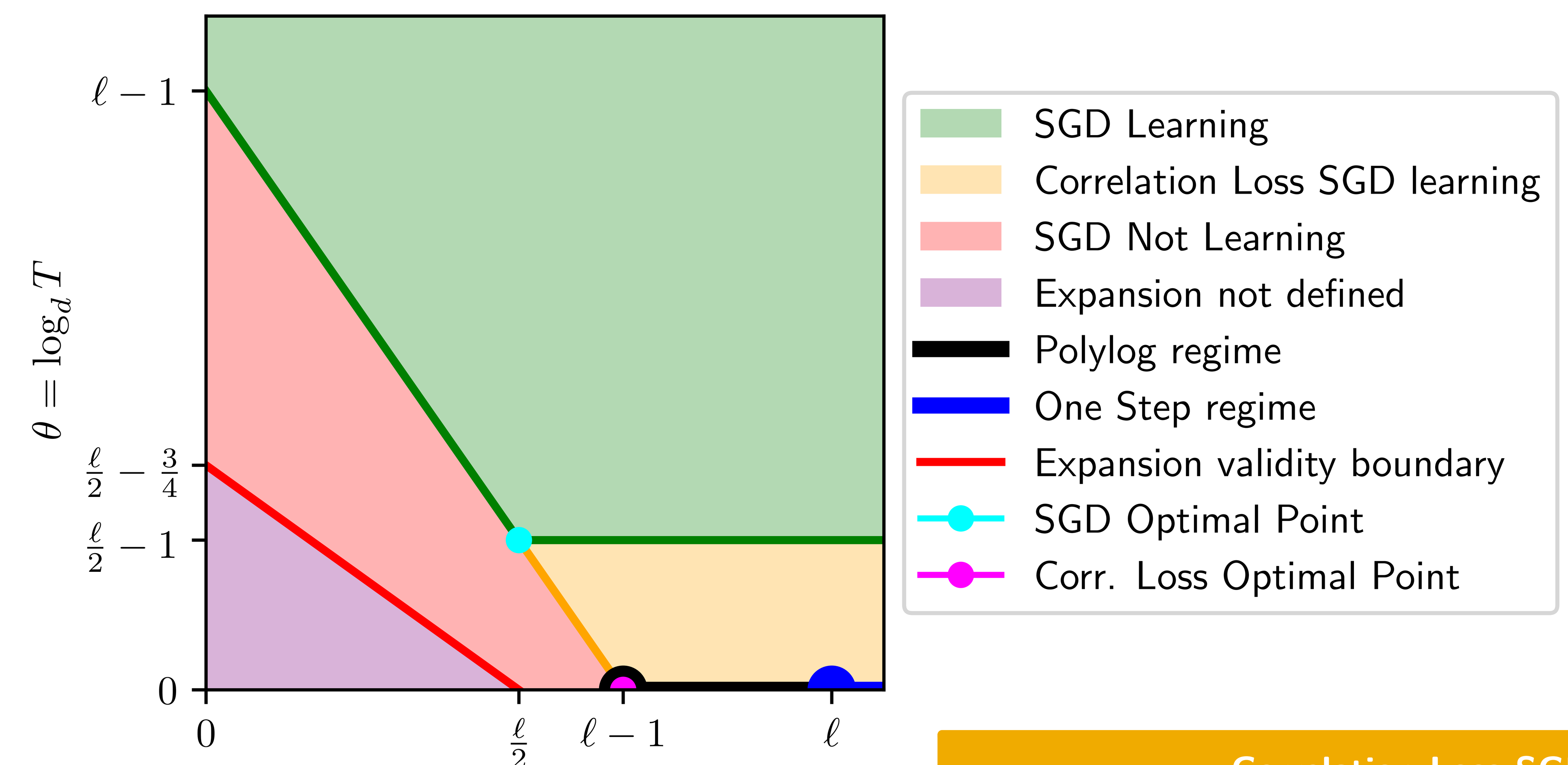
- the learning rate $\gamma = \gamma_0 d^{-\delta}$
- the batch-size $n_b = n_0 d^\mu$

The learning in high-dimensions has happened when the network has *weakly recovered* the target directions, namely the correlation between student and teacher weights is distinguashable from random initialization. The recovery time is
$$T = \min\{t \geq 0 : \|W_t W^{\star\top}\|_F \geq \eta\}$$
for a fixed parameter $\eta \in (0,1)$ independent from d.

We fully characterize the SGD ability to correlate with the target in terms of μ and δ.

Also the weak recovery time scales with the dimension $T \propto d^\theta$.

Time / Batch size Phase diagram



Information Exponent

$$\ell = \min\{k \in \mathbb{N} : \mathbb{E}_{x \sim \mathcal{N}(0,1)}[h^\star(x) H_k(x)] \neq 0\}.$$

The extension to multi-index models introduces an exponent for each target direction. It is referred as the *leap exponent* or *leap index*.

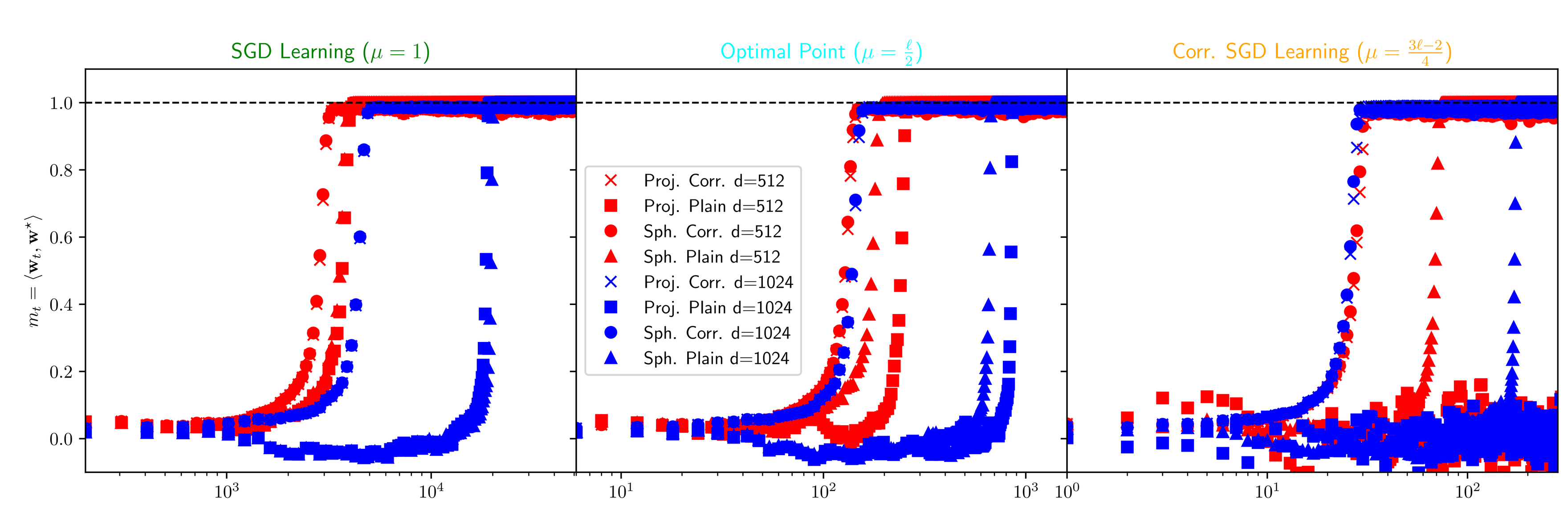
Intuition Using larger batch size reduces the noise in the gradient estimation, ultimately allowing to increase the learning rate and hence the speed of learning.
Results The sample complexity is given by
$$N = n_b \cdot T \propto d^{\theta+\delta}.$$

The phase diagram tells us that:

- All the result are up to a $\log d$ factor.
- The *batch size* **does not affect** the time complexity of the learning: $N = d^{\ell-1}$.
- There is a **tradeoff** between *time* (number of steps̃learning rate) and *memory* (number of samples per batch): Having more computational resources allows to use larger batch sizes, and learn faster.

Projected SGD

- There exists a critical number of time-steps that cannot be reduced by increasing the batch size at $T = O\left(d^{\frac{\ell}{2}-1}\right)$. This corresponds to the transition from vanishing to $O_d(1)$ learning rate.
- There exists an **optimal batch size** that is the smallest one that allows to reach the critical time:
$$n_b = d^{\frac{\ell}{2}}.$$

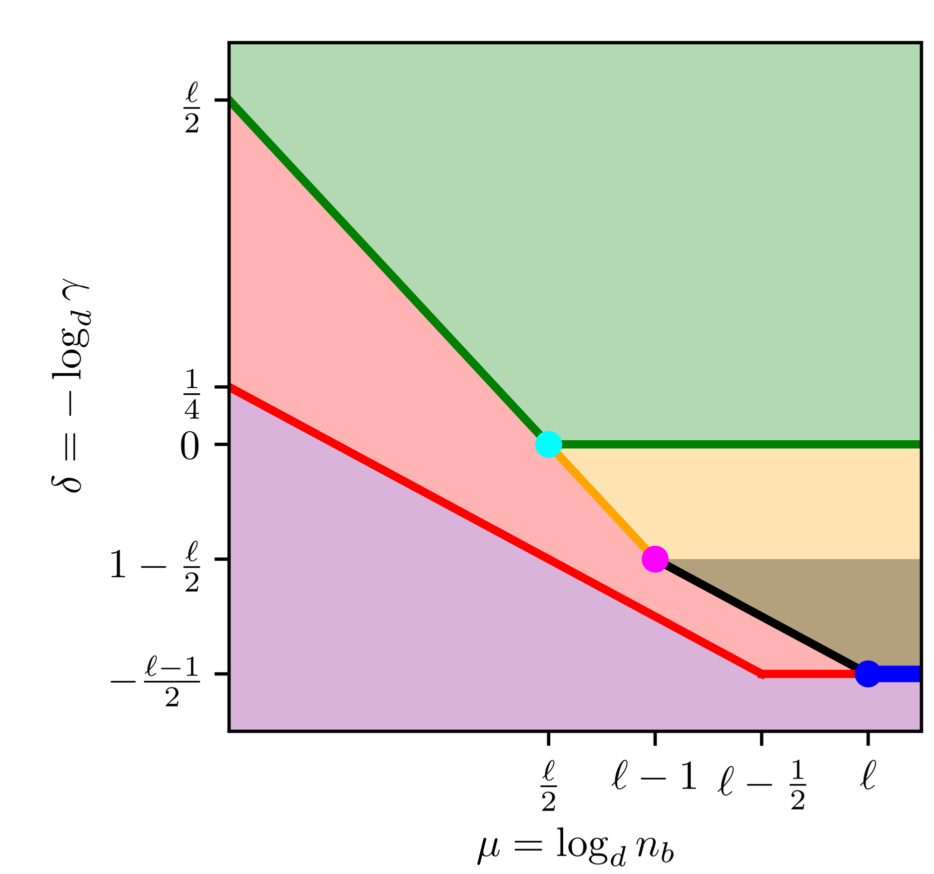


Correlation Loss SGD

Motivation When the learning scale is non-vanishing projected algorithms fail to remain local: the gradient component aligned with student weights, combined with normalization makes the algorithm unstable if doing big jumps.

- The time can be pushed down to $T = O_d(1)$ by increasing the batch size.
- There exists another **optimal point batch size**
$$n_b = d^{\ell-1}.$$
- The *one (giant) step regime*($T = 1$) can be reached when using a sufficiently large batch size and learning rate [2].

Batch-size / Learning rate phase diagram ($\ell \geq 2$)



Spherical SGD

It behaves like the correlation loss, but with different coefficients.

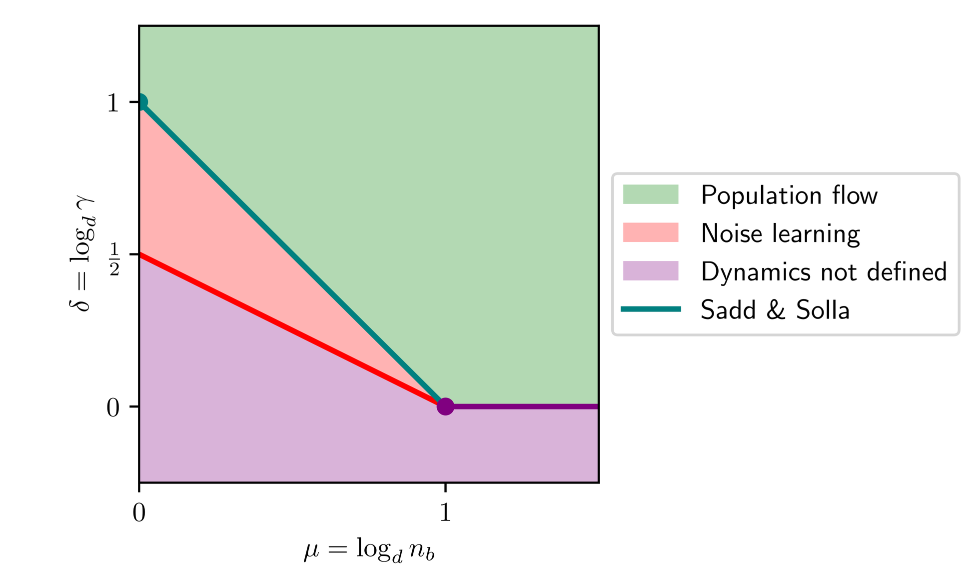
Exact Low-dimensional Asymptotic dynamics

We track the evolution of covariance of pre-activations:
$$\Omega_t := \begin{pmatrix} Q_t & M_t \\ M_t^\top & P \end{pmatrix} = \begin{pmatrix} W_t W_t^\top & W_t W^{\star\top} \\ W^\star W_t^\top & W^\star W^{\star\top} \end{pmatrix}$$

Theorem (Informal)

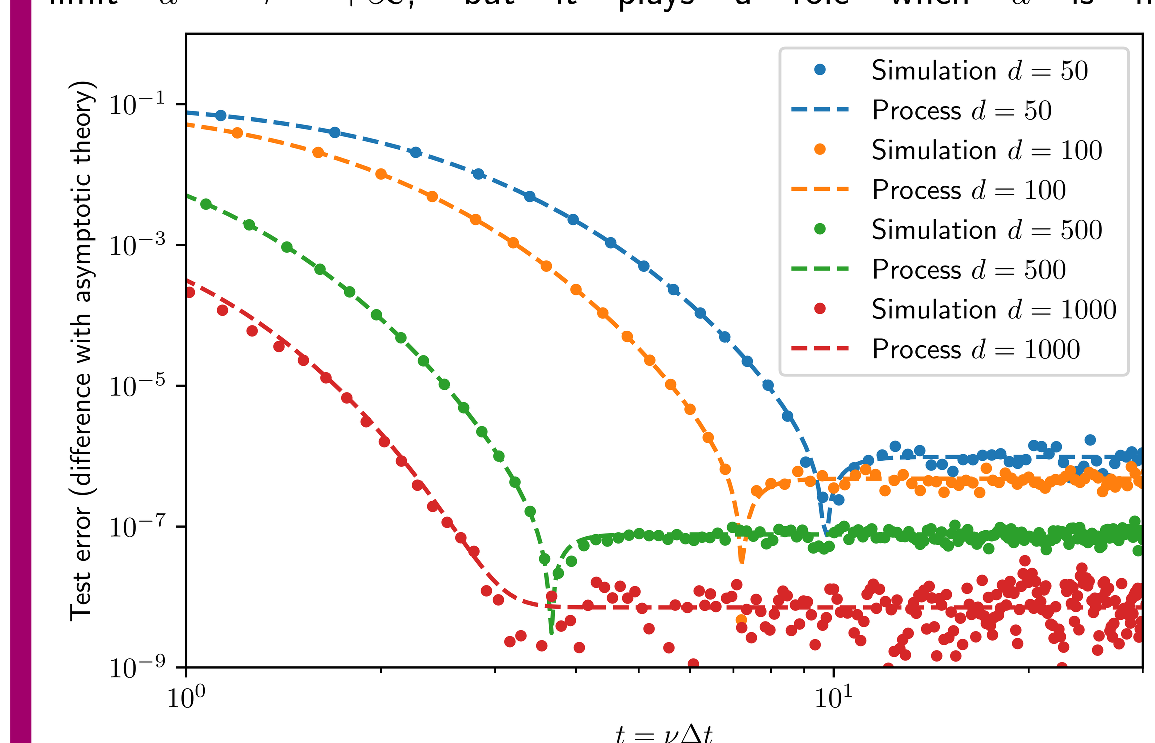
The Projected SGD dynamics of the covariance matrix is approximated in the limit $d \to +\infty$ by the following differential equation:
$$\frac{\mathrm{d}M}{\mathrm{d}t} = \Psi(\Omega; \delta, \mu)$$
$$\frac{\mathrm{d}Q}{\mathrm{d}t} = \Phi(\Omega; \delta, \mu)$$
where Ψ and Φ have the same form as the case $\mu = 0$ [3].

The result is once again summarized by a phase diagram in the $\mu - \delta$ plane.



Non-asymptotic corrections

The presence of intra-batch interaction vanishes in the limit $d \to +\infty$, but it plays a role when d is finite.



References

[1] *On the sample complexity of learning generalized linear models with one-pass stochastic gradient descent*, Gérard Ben Arous, Reza Gheissari, Aukosh Jagannath. The Journal of Machine Learning Research, Volume 22, Issue 1, 2021.

[2] *Learning two-layer neural networks, one (giant) step at a time*, Yatin Dandi, Florent Krzakala, Bruno Loureiro, Luca Pesce, Ludovic Stephan. arXiv preprint arXiv:2305.18270

[3] *Phase diagram of Stochastic Gradient Descent in high-dimensional two-layer neural networks* Rodrigo Veiga, Ludovic Stephan, Bruno Loureiro, Florent Krzakala, Lenka Zdeborová. Advances in Neural Information Processing Systems 35, 2022.