Efficient, private and controllable machine learning & computing systems

Warwick Business School
Imperial College London

Anastasia Borovykh, Nikolas Kantas, Panos Parpas, Greg Pavliotis

The setting

We use more and more **data-collecting devices that have processing power**. In the internet of things setting **devices can interact** with each other.

This opens up **possibilities**:

I. More personalized and data-driven services for individuals and businesses.

II. Better optimize processes in society e.g. logistics, healthcare, energy systems.

III. Better understand natural phenomena in e.g. physics, biology, neuroscience.

However it also poses **challenges**:

I. If devices operate autonomously this can result in wrong decisions being made; e.g. self-driving cars causing traffic accidents

II. By not accounting for privacy, there is a risk of data being compromised by malicious actors.

III. We can face huge energy bills and data storage costs.

Thus we require machine learning and computing systems that are:

I. **Computationally efficient**: require a low computational costs, operate over smartly chosen datasets, minimize the communication costs with other devices.

II. **Robust and controllable**: if the data the device receives changes we want the performance of the devices to be predictable. The decisions of the devices need to be controllable & understandable by humans.

III. **Private**: maintain privacy of the users over whose data the models are trained.

The framework

We propose to use N **interacting agents**: $x^1,\dots,x^N$. Our motivation is to *exploit the collective intelligence* of complex networked systems.

$$dx_t^i = \underbrace{f_i(x_t^i)dt}_{\text{local dynamics}} + \underbrace{\sum_{j=1}^{N}\phi(x_t^i,x_t^j)dt}_{\text{interaction with other agents}} + \underbrace{\sigma^i dW_t^i}_{\text{Brownian noise}} .$$

Goals for learning parameters x:

I. Efficient convergence to parameters such that model has the best performance for the task of interest.

II. The final model needs to be robust and controllable.

III. Minimize ability to extract info on underlying data from the model parameters.

Local dynamics.
The local decision each agent makes.
How do we want to choose it?

i. Few iterations needed for convergence (e.g. preconditioned gradient)

ii. Efficient to compute (e.g. an efficient configuration of the deep learning model)

Interaction.
How agents exchange information.
How do we want to choose it?

i. Learn from knowledge of all devices.

ii. More efficiently explore loss function.

iii. Low communication costs with other devices.

Noise.
Here Brownian - can be more general.
Why do we add noise?

i. Inherent in algorithm (e.g. inaccurate gradient estimates or corrupted communication)

ii. Added into algorithm to converge to more robust optima that leak less private information.

The analysis

Analyse the behavior of these dynamics using **two perspectives:**

I. Stochastic analysis view: $\sim e^{-\frac{2}{\sigma^2}f}$
View the dynamics as Markov processes, convergence to invariant measure, mixing rate of the particles.

II. Dynamical system view: $\min_{x\in\mathcal{X}} f(x)$
Define the right Lyapunov function and identify the stationary states.

Some questions to ask:

I. How many iterations are needed for convergence?

II. How to control the algorithm through the right choice of dynamics?

III. From parameters x can we extract information on data they were computed over?

IV. How to define the interaction structure to maintain low communication costs while it converges fast?

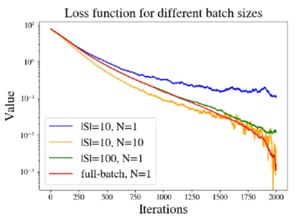
Some results:

A. *Variance reduction*
Interactions can facilitate convergence closer to the optimal solution [1].

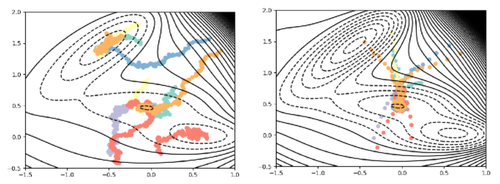
B. *Robustness*
With interactions we may converge to parameters that generalize best to unseen data [1].

C. *Efficiency*
The right local dynamics and interaction results in less iterations needed to converge [2].

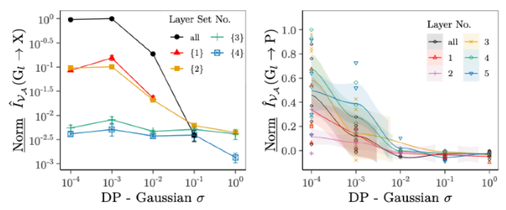
D. *Privacy*
Addition of Brownian noise can result in parameters that leak less private information [3,4].



A linear model where the gradient is noisy as it is computed over batches |Sl| of the data. Using interacting particles allows to use a smaller batch size while still attaining convergence.



A non-convex function (Müller-Brown potential) with N = 10 using i.i.d. (left) and interacting particles with (right). With interaction the particles can converge to the flattest minimum as measured by the Trace of the Hessian.



Adding more noise allows to decrease the privacy risk of extracting sensitive information from layers of a deep neural network.

[1] *A. Borovykh, N. Kantas, P. Parpas, G. Pavliotis, On stochastic mirror descent with interacting particles: convergence properties and variance reduction, Physica D Nonlinear Phenomena 418, 132844, 2021*

[2] *A. Borovykh, N. Kantas, P. Parpas, G. Pavliotis, Stochastic Mirror Descent for Convex Optimization with Consensus Constraints, submitted to SIAM Journal on Optimization 2022*

[3] *M. Malekzadeh, A. Borovykh, D. Gündüz , Honest-but-Curious Nets: Sensitive attributes of private inputs can be secretly encoded into the entropy of classifiers' outputs, ACM Conference on Computer and Communications Security (ACM CCS) 2021*

[4] *M. Fan, A. Borovykh, M. Malekzadeh, S. Demetriou, H. Haddadi, Layer-wise Characterization of Latent Information Leakage in Federated Learning, International Conference on Learning Representations (ICLR) Workshop 'Distributed and Private Machine Learning (DPML)', 2021*

The applications

Finance and healthcare

I. **Market simulation**: each agent is a player in the market. How to choose the dynamics such that market conditions are represented?

II. **Learning across departments**: all departments have valuable data; directly exchanging this data is not possible due to privacy and security constraints. How can we learn from this data while keeping it stored locally?

Security and privacy in internet of things systems

I. Internet-of-things devices (such as phones, smart watches, etc.) have the capacity to store and process information. Ideally the IoT devices would collectively learn a model without transporting data to a central server.

II. One potential solution: exchange model parameters computed locally by each agent [5].

III. How to define the learning dynamics such that from the model parameters it is not possible to extract sensitive information on the local data?

[5] McMahan, Brendan, et al. "Communication-efficient learning of deep networks from decentralized data." *Artificial intelligence and statistics*. PMLR, 2017.

Logistics and energy systems

I. **Vehicles** can be equipped with GPS trackers collecting data on the process. How to optimize logistics processes in the city using forecasted travel or loading times? How do we learn optimal route behaviour from experienced driver?

II. Optimize **distributed renewable energy assets** based on energy need and energy generation predictions.