Intelligence Artificielle
Recherche Opérationnelle

Jérôme MALICK

CNRS, Lab. Jean Kuntzmann & MIAI (Institut IA de Grenoble)

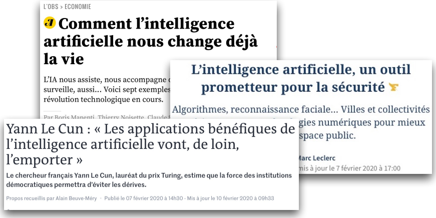
Tutoriel du GdR RO

ROADEF – Feb. 2020 – Montpellier

Fact: AI is everywhere...

On the news

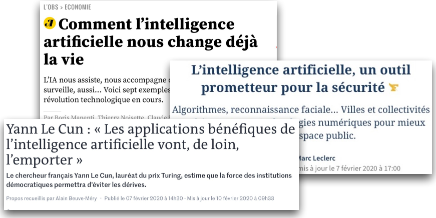
...and people's discussions



Fact: AI is everywhere...

On the news

...and people's discussions


L'OBS > ECONOMIE

Comment l'intelligence artificielle nous change déjà la vie

L'IA nous assiste, nous accompagne
surveille, aussi... Voici sept exemples
révolution technologique en cours.

Par Boris Manenti, Thierry Noisette, Claude

L'intelligence artificielle, un outil prometteur pour la sécurité

Algorithmes, reconnaissance faciale... Villes et collectivités
gies numériques pour mieux
space public.

arc Leclerc
mis à jour le 7 février 2020 à 17:00

Yann Le Cun : « Les applications bénéfiques de l'intelligence artificielle vont, de loin, l'emporter »

Le chercheur français Yann Le Cun, lauréat du prix Turing, estime que la force des institutions démocratiques permettra d'éviter les dérives.

Propos recueillis par Alain Beuve-Méry · Publié le 07 février 2020 à 14h30 - Mis à jour le 10 février 2020 à 09h33

In our universities

e.g. AI institutes

INVESTIR L'AVENIR

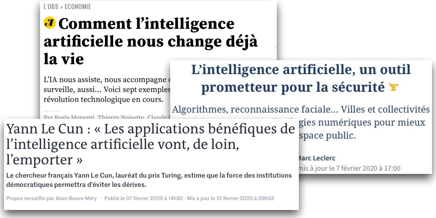
MIAI Grenoble Alpes

Multidisciplinary Institute In Artificial Intelligence

Fact: AI is everywhere...

On the news

...and people's discussions


L'OBS > ECONOMIE

Comment l'intelligence artificielle nous change déjà la vie

L'IA nous assiste, nous accompagne … surveille, aussi… Voici sept exemples … révolution technologique en cours.

Par Boris Manenti, Thierry Noisette, Claude …

L'intelligence artificielle, un outil prometteur pour la sécurité

Algorithmes, reconnaissance faciale… Villes et collectivités … gies numériques pour mieux … space public.

…arc Leclerc

…mis à jour le 7 février 2020 à 17:00

Yann Le Cun : « Les applications bénéfiques de l'intelligence artificielle vont, de loin, l'emporter »

Le chercheur français Yann Le Cun, lauréat du prix Turing, estime que la force des institutions démocratiques permettra d'éviter les dérives.

Propos recueillis par Alain Beuve-Méry · Publié le 07 février 2020 à 14h30 - Mis à jour le 10 février 2020 à 09h33

In our universities

e.g. AI institutes

In the scientific community

e.g. Turing award 2019

to deep learning pioneers

Let's play together !

Various, fruitful interactions AI $\leftrightarrows$ OR

Let's play together !

Various, fruitful interactions AI $\leftrightarrows$ OR

- E.g. last year ROADEF/EURO Challenge

Let's play together !

Various, fruitful interactions AI $\leftrightarrows$ OR

- E.g. last year ROADEF/EURO Challenge

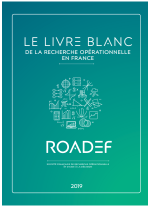
New perspectives in decision-making ☺

- E.g. design and control of smart infrastructure (grids, transportation,...)
- Hence the title of this "tutorial" (...actually more like a bird-eye overview)

Vast domain at the interface AI/OR

A couple of pointers in a jungle of references

A. Parmentier's talk

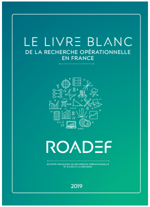


B. Rottembourg

Vast domain at the interface AI/OR

A couple of pointers in a jungle of references

A. Parmentier's talk



B. Rottembourg

A complementary viewpoint:

This tutorial : modest goals

- recall basics of machine learning
- mention (important) general ideas
- illustrate related theoretical research topics

Recent research in the team DAO @LJK

Y.-G. Hsieh, F. Iutzeler, J. Malick, P. Mertikopoulos,
On the Convergence of Single-Call Stochastic Extra-Gradient Methods
NeurIPS, 2019

M. Grishchenko, F. Iutzeler, J. Malick
Subspace Descent Methods with Identification-Adapted Sampling
Submitted to: Maths of OR, 2019

Y. Laguel, J. Malick, Z. Harchaoui
Superquantile Minimization: Oracles and First-order Algorithms
Submitted to: Optimization Methods and Software, 2019

Outline

1. Back to basics: learning is optimizing
2. Discussion: (some) perspectives, (deep) questions, and (personal) thoughts
3. Highlight: fresh interest on min-max
4. Highlight: collaborative/federative learning

Outline

Supervised learning set-up

Data: n observations $(a_i, y_i) \in \mathbb{R}^m \times \mathcal{Y}$

Task: e.g. binary classification $\quad (\mathcal{Y} = \{0, 1\})$

$a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5$

$y_1 = 0 \quad y_2 = 1 \quad y_3 = 0 \quad y_4 = 1 \quad y_5 = 0$

Supervised learning set-up

Data: n observations $(a_i, y_i) \in \mathbb{R}^m \times \mathcal{Y}$

Task: e.g. binary classification $(\mathcal{Y} = \{0, 1\})$

a_1 a_2 a_3 a_4 a_5

$y_1 = 0$ $y_2 = 1$ $y_3 = 0$ $y_4 = 1$ $y_5 = 0$

Model: for a new a, prediction $h(a, x) \in \mathcal{Y}$ parameterized by $x \in \mathbb{R}^d$

usually $x = \beta$ (in stats) $x = \omega$ (in learning) or $x = \theta$ (in deep learning)

Standard prediction functions :

- Linear prediction: $h(a, x) = a^\top x$

Supervised learning set-up

Data: n observations $(a_i, y_i) \in \mathbb{R}^m \times \mathcal{Y}$

Task: e.g. binary classification $(\mathcal{Y} = \{0,1\})$

a_1 a_2 a_3 a_4 a_5

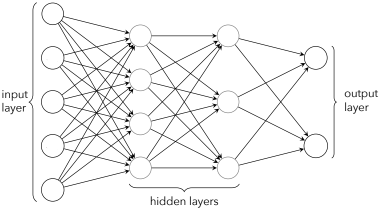
$y_1 = 0$ $y_2 = 1$ $y_3 = 0$ $y_4 = 1$ $y_5 = 0$

Model: for a new a, prediction $h(a, x) \in \mathcal{Y}$ parameterized by $x \in \mathbb{R}^d$

usually $x = \beta$ (in stats) $x = \omega$ (in learning) or $x = \theta$ (in deep learning)

Standard prediction functions :

- Linear prediction: $h(a, x) = a^\top x$
- Artificial neural networks:
  $h(a, x) = {x_m}^\top \sigma({x_{m-1}}^\top \cdots \sigma({x_1}^\top a))$



Optimization comes into play

- Learning = finding the "best" parameter $\bar{x}$

  = finding $\bar{x}$ such that $h(a_i, \bar{x}) \simeq y_i$ (and generalizes well on unseen data)

  = solving an optimization problem

- Regularized empirical risk minimization

$$\min_{x \in \mathbb{R}^d} \quad f(x) \;=\; \frac{1}{n}\sum_{i=1}^{n} \ell\big(y_i, h(a_i, x)\big)$$

Optimization comes into play

- Learning = finding the "best" parameter $\bar{x}$

  = finding $\bar{x}$ such that $h(a_i, \bar{x}) \simeq y_i$ (and generalizes well on unseen data)

  = solving an optimization problem

- Regularized empirical risk minimization

  (regularization avoids overfitting, helps numerically, or imposes structure to x)

$$\min_{x \in \mathbb{R}^d} \quad f(x) \;=\; \frac{1}{n}\sum_{i=1}^{n} \ell\big(y_i, h(a_i, x)\big) \quad + \quad \lambda\, r(x)$$

Optimization comes into play

- Learning = finding the "best" parameter $\bar{x}$

  = finding $\bar{x}$ such that $h(a_i, \bar{x}) \simeq y_i$ (and generalizes well on unseen data)

  = solving an optimization problem

- Regularized empirical risk minimization

  (regularization avoids overfitting, helps numerically, or imposes structure to x)

$$\min_{x \in \mathbb{R}^d} \quad f(x) \;=\; \frac{1}{n}\sum_{i=1}^{n} \ell\big(y_i, h(a_i, x)\big) \;+\; \lambda\, r(x)$$

- Example: linear model $h(a, x) = a^\top x$ and least-squares loss

$$\min_{x \in \mathbb{R}^d} \quad \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}\big(y_i - a_i^\top x\big)^2 + \frac{\lambda}{2}\|x\|_2^2 \;=\; \frac{1}{2n}\|Ax - y\|_2^2 + \frac{\lambda}{2}\|x\|_2^2$$

Optimization comes into play

- Learning $=$ finding the "best" parameter $\bar{x}$
  $=$ finding $\bar{x}$ such that $h(a_i, \bar{x}) \simeq y_i$ (and generalizes well on unseen data)
  $=$ solving an optimization problem

- Regularized empirical risk minimization
  (regularization avoids overfitting, helps numerically, or imposes structure to x)

$$\min_{x \in \mathbb{R}^d} \quad f(x) \;=\; \frac{1}{n}\sum_{i=1}^{n} \ell\big(y_i, h(a_i, x)\big) \;+\; \lambda\, r(x)$$

- Example: linear model $h(a, x) = a^\top x$ and least-squares loss

$$\min_{x \in \mathbb{R}^d} \quad \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}\big(y_i - a_i^\top x\big)^2 + \frac{\lambda}{2}\|x\|_2^2 \quad = \quad \frac{1}{2n}\|Ax - y\|_2^2 + \frac{\lambda}{2}\|x\|_2^2$$

$$\min_{x \in \mathbb{R}^d} \quad \frac{1}{2n}\|Ax - y\|_2^2 + \frac{\lambda}{2}\|x\|_1$$ ℓ_1-norm promotes sparse solutions

Stochastic gradient rules

(in the simple case $r = 0$)

- Basic optimization algorithm : stochastic gradient descent (SGD)

Draw random i_k

$$x_{k+1} = x_k - \gamma_k\, g_k \qquad \text{with} \quad g_k = \nabla \ell\big(y_{i_k}, h(a_{i_k}, x_k)\big)$$

$$\text{with} \quad \mathbb{E}\,[g_k] = \nabla f(x_k)$$

Stochastic gradient rules

(in the simple case $r = 0$)

- Basic optimization algorithm : stochastic gradient descent (SGD)

$$\text{Draw random } i_k$$

$$x_{k+1} = x_k - \gamma_k\, g_k \qquad \text{with} \quad g_k = \nabla \ell\big(y_{i_k}, h(a_{i_k}, x_k)\big)$$

$$\text{with} \quad \mathbb{E}\left[g_k\right] = \nabla f(x_k)$$

- We can often compute the gradient
  E.g. back-propagation for neural networks to derivate $x \mapsto \ell\big(y, h(a, x)\big)$

- Tuning of γ_k is the key of efficiency

- Zoology: SAG, SDCA, Miso, SVRG, SAGA, Adam, AdaGrad, Eve,...
  + mini-batch + prox-versions + accelerated versions + 2nd order

Bottomline

Stochastic first-order optimization (training) methods work great!

Stochastic gradient rules

(in the simple case $r = 0$)

- Basic optimization algorithm : stochastic gradient descent (SGD)

$$
\begin{aligned}
&\text{Draw random } i_k \\
&x_{k+1} = x_k - \gamma_k\, g_k \qquad \text{with} \quad g_k = \nabla \ell\big(y_{i_k}, h(a_{i_k}, x_k)\big) \\
&\qquad\qquad\qquad\qquad\;\; \text{with} \quad \mathbb{E}\,[g_k] = \nabla f(x_k)
\end{aligned}
$$

- We can often compute the gradient
  E.g. back-propagation for neural networks to derivate $x \mapsto \ell\big(y, h(a, x)\big)$

- Tuning of γ_k is the key of efficiency

- Zoology: SAG, SDCA, Miso, SVRG, SAGA, Adam, AdaGrad, Eve,...
  + mini-batch + prox-versions + accelerated versions + 2nd order

Bottomline

Stochastic first-order optimization (training) methods work great!

...except when they don't
- non-convex landscape, very few guarantees...
- leaving aside i.i.d. train/test data, biased data,...

More than the just workhorse...

Optimization plays a fundamental role in learning

E.g. Test of Time Award

NeurIPS 2018

The Tradeoffs of Large Scale Learning

Léon Bottou
NEC laboratories of America
Princeton, NJ 08540, USA
leon@bottou.org

Olivier Bousquet
Google Zürich
8002 Zurich, Switzerland
olivier.bousquet@m4x.org

Abstract

This contribution develops a theoretical framework that takes into account the effect of approximate optimization on learning algorithms. The analysis shows distinct tradeoffs for the case of small-scale and large-scale learning problems. Small-scale learning problems are subject to the usual approximation–estimation tradeoff. Large-scale learning problems are subject to a qualitatively different tradeoff involving the computational complexity of the underlying optimization algorithms in non-trivial ways.

1 Motivation

The computational complexity of learning algorithms has seldom been taken into account by the learning theory. Valiant [1] states that a problem is "learnable" when there exists a probably approximatively correct learning algorithm *with polynomial complexity*. Whereas much progress has been made on the statistical aspect (e.g., [2, 3, 4]), very little has been told about the complexity side of this proposal (e.g., [5].)

Computational complexity becomes the limiting factor when one envisions large amounts of training data. Two important examples come to mind:

[Bottou & Bousquet '07]

NeurIPS 2019

Dual Averaging Method for Regularized Stochastic Learning and Online Optimization

Lin Xiao
Microsoft Research, Redmond, WA 98052
lin.xiao@microsoft.com

Abstract

We consider regularized stochastic learning and online optimization problems, where the objective function is the sum of two convex terms: one is the loss function of the learning task, and the other is a simple regularization term such as -norm for promoting sparsity. We develop a new online algorithm, the *regularized dual averaging* (RDA) method, that can explicitly exploit the regularization structure in an online setting. In particular, at each iteration, the learning variables are adjusted by solving a simple optimization problem that involves the running average of all past subgradients of the loss functions and the whole regularization term, not just its subgradient. Computational experiments show that the RDA method can be very effective for sparse online learning with -regularization.

1 Introduction

In machine learning, online algorithms operate by repetitively drawing random examples, one at a time, and adjusting the learning variables using simple calculations that are usually based on the single example only. The low computational complexity (per iteration) of online algorithms is often associated with their slow convergence and low accuracy in solving the underlying optimization problems. As argued in [1, 2], the combined low complexity and low accuracy, together with other

[Xiao '09]

ICML 2019

Online Dictionary Learning for Sparse Coding

Julien Mairal JULIEN.MAIRAL@INRIA.FR
Francis Bach FRANCIS.BACH@INRIA.FR
INRIA,[1] 45 rue d'Ulm 75005 Paris, France

Jean Ponce JEAN.PONCE@ENS.FR
Ecole Normale Supérieure,[1] 45 rue d'Ulm 75005 Paris, France

Guillermo Sapiro GUILLE@UMN.EDU
University of Minnesota - Department of Electrical and Computer Engineering, 200 Union Street SE, Minneapolis, USA

Abstract

Sparse coding—that is, modelling data vectors as sparse linear combinations of basis elements—is widely used in machine learning, neuroscience, signal processing, and statistics. This paper focuses on *learning* the basis set, also called dictionary, to adapt it to specific data, an approach that has recently proven to be very effective for signal reconstruction and classification in the audio and image processing domains. This paper proposes a new online optimization algorithm for dictionary learning, based on stochastic approximations, which scales up gracefully to large datasets with millions of training samples. A proof of convergence is presented, along with experiments with natural images demonstrating

like decompositions based on principal component analysis and its variants, these models do not impose that the basis vectors be orthogonal, allowing more flexibility to adapt the representation to the data. While learning the dictionary has proven to be critical to achieve (or improve upon) state-of-the-art results, effectively solving the corresponding optimization problem is a significant computational challenge, particularly in the context of the large-scale datasets involved in image processing tasks, that may include millions of training samples. Addressing this challenge is the topic of this paper.

Concretely, consider a signal $\mathbf{x}$ in $\mathbb{R}^m$. We say that it admits a sparse approximation over a *dictionary* $\mathbf{D}$ in $\mathbb{R}^{m \times k}$, with k columns referred to as *atoms*, when one can find a linear combination of a "few" atoms from $\mathbf{D}$ that is "close" to the signal $\mathbf{x}$. Experiments have shown that modelling a

[Mairal *et al* '09]

Outline

A history of success: ML and IA

- First Generation ('90-'00): the background
  e.g., fraud detection, search engines
- Second Generation ('00-'10)
  e.g., recommendation systems
- Third Generation ('10-now): rebirth of deep learning
  e.g., speech recognition, computer vision, translation...

A history of success: ML and IA

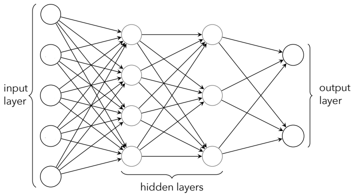
- First Generation ('90-'00): the background
  e.g., fraud detection, search engines
- Second Generation ('00-'10)
  e.g., recommendation systems
- Third Generation ('10-now): rebirth of deep learning
  e.g., speech recognition, computer vision, translation...
- Fourth Generation (emerging): markets
  not just one agent making decisions but multi-agents...
  towards interconnected web of data, agents, decisions

[Jordan '18] "Artificial Intelligence: The Revolution Hasn't Happened Yet"

OK but keep in mind current limits

Success of deep learning

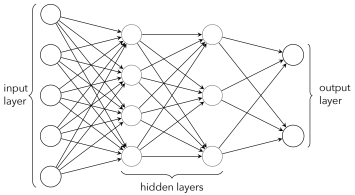
for image recognition



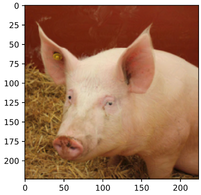
OK but keep in mind current limits

Success of deep learning

for image recognition



Example:

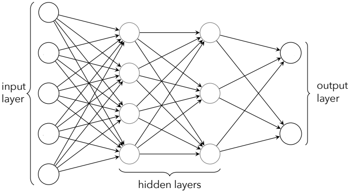


pig (99%)

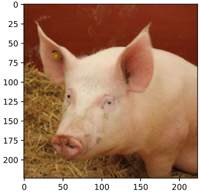
OK but keep in mind current limits

Success of deep learning

for image recognition

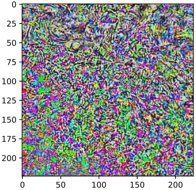


Example: (notebooks of NeurIPS 2018 tutorial on Adversarial Robustness)

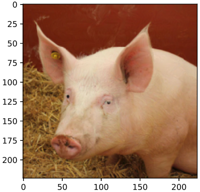


pig (99%)

$+\ \mathcal{E}\times$



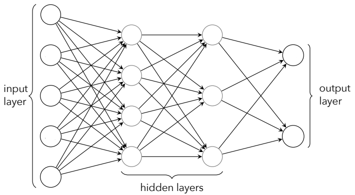
$=$



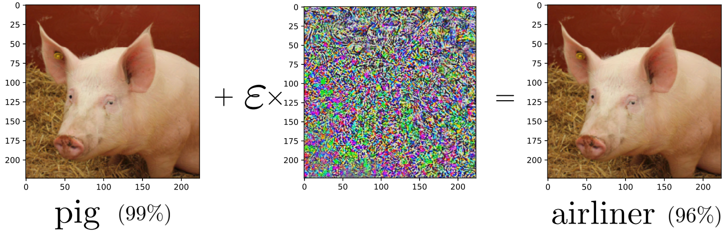
OK but keep in mind current limits

Success of deep learning

for image recognition



Example: (notebooks of NeurIPS 2018 tutorial on Adversarial Robustness)



Warning: fragile approach, and more work needed

Beyond this: political thoughts

1. Deep learning needs a lot of energy

Beyond this: political thoughts

1. Deep learning needs a lot of energy

2. Deep learning needs a lot of data

Beyond this: political thoughts

1. Deep learning needs a lot of energy

2. Deep learning needs a lot of data
   (confidentiality issues)

Facebook a laissé Netflix et Spotify accéder à la messagerie privée de ses utilisateurs

Par Lucie Ronfaut | Mis à jour le 19/12/2018 à 12:01 / Publié le 1

Google sait où vous êtes, même si vous désactivez l'historique des positions

Les services Google suivent et stockent vos données de localisation, même si vous désactivez l'historique des positions dans vos paramètres de confidentialité.

Par Marc Zaffagni avec CNET.com / mardi 14 août 2018 à 10:19 / Mis à jour mardi 14 août 2018 à 16:38

Beyond this: political thoughts

1. Deep learning needs a lot of energy

2. Deep learning needs a lot of data
   (confidentiality issues)

Facebook a laissé Netflix et Spotify accéder à la messagerie privée de ses utilisateurs

Par Lucie Ronfaut | Mis à jour le 19/12/2018 à 12:01 / Publié le 1

Google sait où vous êtes, même si vous désactivez l'historique des positions

Les services Google suivent et stockent vos données de localisation, même si vous désactivez l'historique des positions dans vos paramètres de confidentialité.

Par Marc Zaffagni avec CNET.com / mardi 14 août 2018 à 10:19 / Mis à jour mardi 14 août 2018 à 16:38

3. Manual data treatment (in poor countries)

amazon mechanical turk beta

Beyond this: political thoughts

1. Deep learning needs a lot of energy

2. Deep learning needs a lot of data
   (confidentiality issues)

Facebook a laissé Netflix et Spotify accéder à la messagerie privée de ses utilisateurs

Par Lucie Ronfaut | Mis à jour le 19/12/2018 à 12:01 / Publié le

Google sait où vous êtes, même si vous désactivez l'historique des positions

Les services Google suivent et stockent vos données de localisation, même si vous désactivez l'historique des positions dans vos paramètres de confidentialité.

Par Marc Zaffagni avec CNET.com / mardi 14 août 2018 à 10:19 / Mis à jour mardi 14 août 2018 à 16:38

3. Manual data treatment (in poor countries)

amazon mechanical turk beta

4. AI may be biased

Amazon : l'intelligence artificielle qui n'aimait pas les femmes

Fil info

09:55

09:55

09:40

Accélérer le recrutement en faisant analyser les CV par une IA : l'idée semblait prometteuse à Amazon. Mais elle s'est mise à sous-noter les femmes candidates à des postes tech.

Beyond this: political thoughts

1. Deep learning needs a lot of energy

2. Deep learning needs a lot of data
   (confidentiality issues)

Facebook a laissé Netflix et Spotify accéder à la messagerie privée de ses utilisateurs

Par Lucie Ronfaut | Mis à jour le 19/12/2018 à 12:01 / Publié le 1

Google sait où vous êtes, même si vous désactivez l'historique des positions

Les services Google suivent et stockent vos données de localisation, même si vous désactivez l'historique des positions dans vos paramètres de confidentialité.

Par Marc Zaffagni avec CNET.com / mardi 14 août 2018 à 10:19 / Mis à jour mardi 14 août 2018 à 16:38

3. Manual data treatment (in poor countries)

amazon mechanical turk beta

4. AI may be biased

Amazon : l'intelligence artificielle qui n'aimait pas les femmes

Accélérer le recrutement en faisant analyser les CV par une IA : l'idée semblait prometteuse à Amazon. Mais elle s'est mise à sous-noter les femmes candidates à des postes tech.

Fil info

09:55 Le cri de révolte de N

09:55 Municipales à Paris: l'éponge après des "a

09:40 En Inde, des applicat intimité précieuse au

Tout

Machine Bias

There's software used across the country to predict future criminals. And it's biased against blacks.

by Julia Angwin, Jeff Larson, Surya Mattu and Lauren Kirchner, ProPublica

May 23, 2016

5. Bad bots: fake news and massive people manipulation

Zoom on first point: deep learning needs a lot of energy

How much energy spent on computational experiments of this paper ?

Capacity and Trainability in Recurrent Neural Networks

Jasmine Collins*, Jascha Sohl-Dickstein & David Sussillo
Google Brain
Google Inc.
Mountain View, CA 94043, USA
{jlcollins, jaschasd, sussillo}@google.com

Abstract

Two potential bottlenecks on the expressiveness of recurrent neural networks (RNNs) are their ability to store information about the task in their parameters, and to store information about the input history in their units. We show experimentally that all common RNN architectures achieve nearly the same per-task and per-unit capacity bounds with careful training, for a variety of tasks and stacking depths.

Zoom on first point: deep learning needs a lot of energy

How much energy spent on computational experiments of this paper ?

Energy measure unit: (from J. Duchi, Stanford)

how many Toyota Camrys from Montpellier to Paris ? (approximately)

CAPACITY AND TRAINABILITY IN RECURRENT NEURAL NETWORKS

Jasmine Collins,[*] Jascha Sohl-Dickstein & David Sussillo
Google Brain
Google Inc.
Mountain View, CA 94043, USA
{jlcollins, jaschasd, sussillo}@google.com

ABSTRACT

Two potential bottlenecks on the expressiveness of recurrent neural networks (RNNs) are their ability to store information about the task in their parameters, and to store information about the input history in their units. We show experimentally that all common RNN architectures achieve nearly the same per-task and per-unit capacity bounds with careful training, for a variety of tasks and stacking depths.

1?

Zoom on first point: deep learning needs a lot of energy

How much energy spent on computational experiments of this paper ?

Energy measure unit: (from J. Duchi, Stanford)

how many Toyota Camrys from Montpellier to Paris ? (approximately)

CAPACITY AND TRAINABILITY IN RECURRENT NEURAL NETWORKS

Jasmine Collins*, Jascha Sohl-Dickstein & David Sussillo
Google Brain
Google Inc.
Mountain View, CA 94043, USA
{jlcollins, jaschasd, sussillo}@google.com

ABSTRACT

Two potential bottlenecks on the expressiveness of recurrent neural networks (RNNs) are their ability to store information about the task in their parameters, and to store information about the input history in their units. We show experimentally that all common RNN architectures achieve nearly the same per-task and per-unit capacity bounds with careful training, for a variety of tasks and stacking depths.

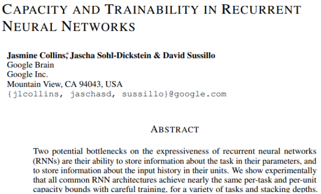
10?

Zoom on first point: deep learning needs a lot of energy

How much energy spent on computational experiments of this paper ?

Energy measure unit: (from J. Duchi, Stanford)

how many Toyota Camrys from Montpellier to Paris ? (approximately)



CAPACITY AND TRAINABILITY IN RECURRENT NEURAL NETWORKS

Jasmine Collins*, Jascha Sohl-Dickstein & David Sussillo
Google Brain
Google Inc.
Mountain View, CA 94043, USA
{jlcollins, jaschasd, sussillo}@google.com

ABSTRACT

Two potential bottlenecks on the expressiveness of recurrent neural networks (RNNs) are their ability to store information about the task in their parameters, and to store information about the input history in their units. We show experimentally that all common RNN architectures achieve nearly the same per-task and per-unit capacity bounds with careful training, for a variety of tasks and stacking depths.

100?

Zoom on first point: deep learning needs a lot of energy

How much energy spent on computational experiments of this paper ?

Energy measure unit: (from J. Duchi, Stanford)

how many Toyota Camrys from Montpellier to Paris ? (approximately)

CAPACITY AND TRAINABILITY IN RECURRENT NEURAL NETWORKS

Jasmine Collins*, Jascha Sohl-Dickstein & David Sussillo
Google Brain
Google Inc.
Mountain View, CA 94043, USA
{jlcollins, jaschasd, sussillo}@google.com

ABSTRACT

Two potential bottlenecks on the expressiveness of recurrent neural networks (RNNs) are their ability to store information about the task in their parameters, and to store information about the input history in their units. We show experimentally that all common RNN architectures achieve nearly the same per-task and per-unit capacity bounds with careful training, for a variety of tasks and stacking depths.

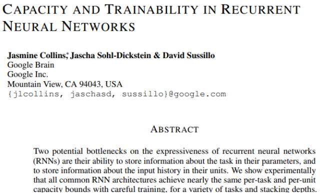
1000?

Zoom on first point: deep learning needs a lot of energy

How much energy spent on computational experiments of this paper ?

Energy measure unit: (from J. Duchi, Stanford)

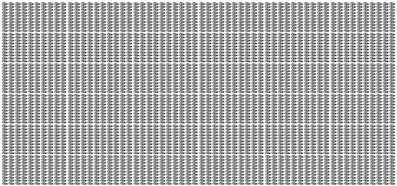
how many Toyota Camrys from Montpellier to Paris ? (approximately)


CAPACITY AND TRAINABILITY IN RECURRENT NEURAL NETWORKS

Jasmine Collins*, Jascha Sohl-Dickstein & David Sussillo
Google Brain
Google Inc.
Mountain View, CA 94043, USA
{jlcollins, jaschasd, sussillo}@google.com

ABSTRACT

Two potential bottlenecks on the expressiveness of recurrent neural networks (RNNs) are their ability to store information about the task in their parameters, and to store information about the input history in their units. We show experimentally that all common RNN architectures achieve nearly the same per-task and per-unit capacity bounds with careful training, for a variety of tasks and stacking depths.



4200

Deep learning has a terrible carbon footprint...

⟶ work needed towards energy-efficient learning models

⟶ for us, in particular: efficient optimization algorithms

(fast convergence, automatic tuning of parameters,...)

Outline

Min-Max & friends

In OR/Optimization/Games,

we are used to deal with min-max, (Nash) equilibrium, or saddle-points...

$$\min_{x\in X} \max_{y\in Y} F(x,y) \quad \text{or} \quad F(x^\star,y) \leqslant F(x^\star,y^\star) \leqslant F(x,y^\star)$$

Examples: nonconvex problems (with X discrete)

- Lagrangian duality or relaxation
  e.g. [Lemaréchal '01] "the omnipresence of Lagrange"

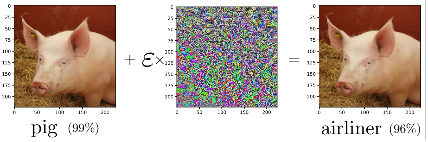
$$\min_{u} \max_{x\in X} L(x,u) = p^\top x - u^\top c(x)$$

- Robust optimization
  e.g. the work of our conference chair [Poss '18], [Poss *et al* '16]

$$\min_{x\in X} \max_{\xi\in\Delta_x} f(x,\xi)$$

Examples in AI: non-convex too

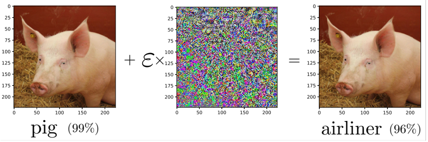
Adversarially robust models [Kolter & Madry '18]



$$\min_{x\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \max_{\delta\in\Delta} \ell\big(y_i, h(x, a_i+\delta)\big)$$

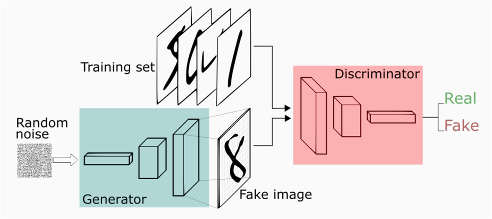
Examples in AI: non-convex too

Adversarially robust models [Kolter & Madry '18]



$$\min_{x\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \max_{\delta\in\Delta} \ell\big(y_i, h(x, a_i+\delta)\big)$$

Generate data with GANs [Goodfellow *et al* '14]



$$\min_{x_G}\max_{x_D} \ \mathbb{E}_{a\sim\text{data}}\Big[\log h_D\big(x_D, a\big)\Big] + \mathbb{E}_{z\sim\text{noise}}\Big[\log\Big(1 - h_D\big(x_D, h_G(x_G, z)\big)\Big)\Big]$$

GANs: sucesses and failures

Question: who is real, who isn't ?

GANs: sucesses and failures

Question: who is real, who isn't ?

Answer: both are fake !
[https://thispersondoesnotexist.com]


GANs: sucesses and failures

Question: who is real, who isn't ?

Answer: both are fake !
[https://thispersondoesnotexist.com]

But the story far from being over...

GANs: sucesses and failures

Question: who is real, who isn't ?

Answer: both are fake !
[https://thispersondoesnotexist.com]

But the story far from being over...

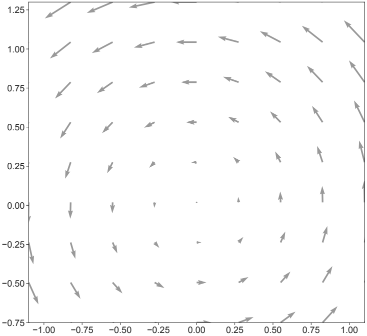
Coupling of two neural networks gives rise to strange behaviors

Even when solved with state-of-the-art stochastic gradient (extra-gradient variants)

Example of strange phenomena... and a simple fix

Non-convergent phenomena are observed even in very simple problems

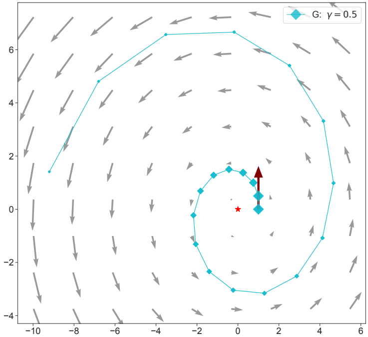
Example: $\min_x \max_y \; x\,y$ solution (0,0)



Example of strange phenomena... and a simple fix

Non-convergent phenomena are observed even in very simple problems

Example: $\min_x \max_y \; x\,y$ solution (0,0)

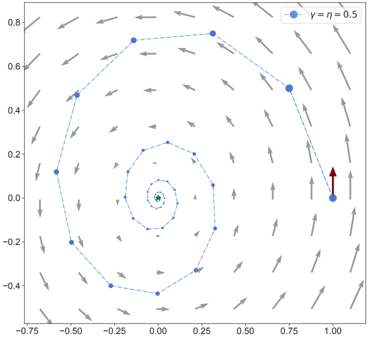


- Gradient algorithm diverges...

Example of strange phenomena... and a simple fix

Non-convergent phenomena are observed even in very simple problems

Example: $\min_{x} \max_{y} \; x\,y$ solution (0,0)

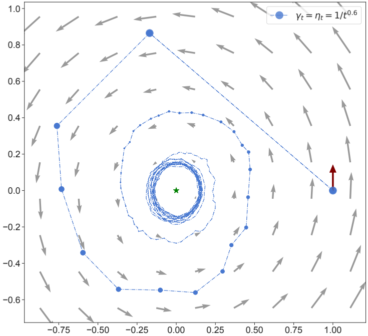


- Gradient algorithm diverges...
- Extra-gradient algorithm converges [Korpelevich '76] ($\to$ GANs)

Example of strange phenomena... and a simple fix

Non-convergent phenomena are observed even in very simple problems

Example: $\min_{x} \max_{y} \; x\, y$ solution (0,0)

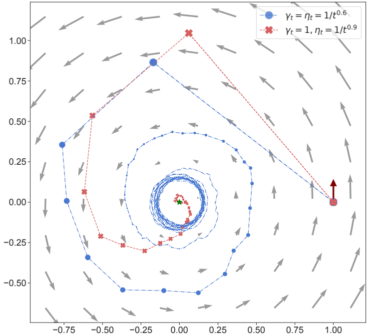


- Gradient algorithm diverges...
- Extra-gradient algorithm converges [Korpelevich '76] ($\rightarrow$ GANs)
- Stochastic extra-gradient never converges...

Example of strange phenomena... and a simple fix

Non-convergent phenomena are observed even in very simple problems

Example: $\min_x \max_y \; x\,y$ solution (0,0)

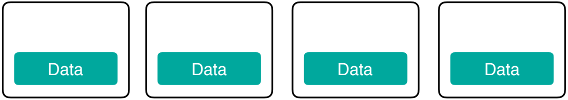


- Gradient algorithm diverges...
- Extra-gradient algorithm converges [Korpelevich '76] ($\rightarrow$ GANs)
- Stochastic extra-gradient never converges...
- A remedy: use double stepsize [Hsieh, Iutzeler, M., Mertikopoulos '20]

Outline

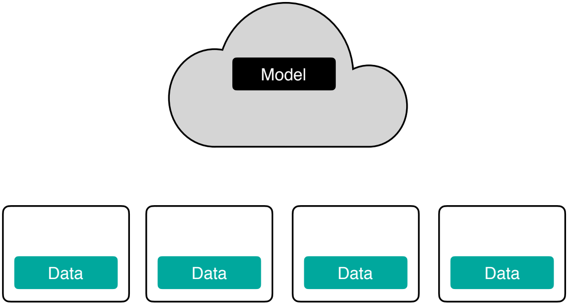
Collaborative learning

Set-up:



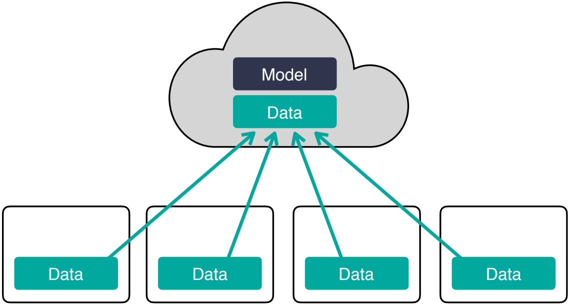
Collaborative learning

Set-up: (standard) centralized learning



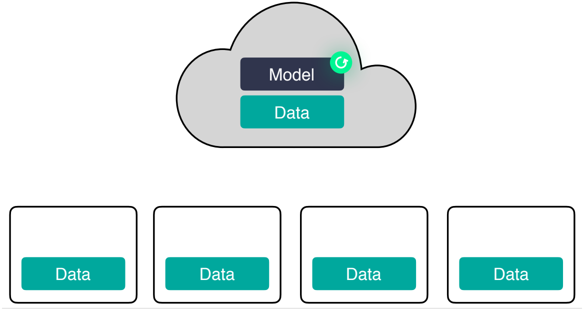
Collaborative learning

Set-up: (standard) centralized learning



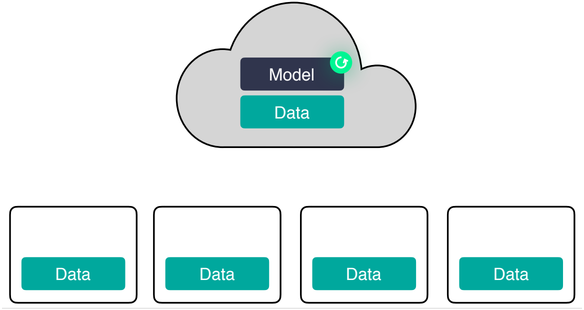
Collaborative learning

Set-up: (standard) centralized learning



Collaborative learning

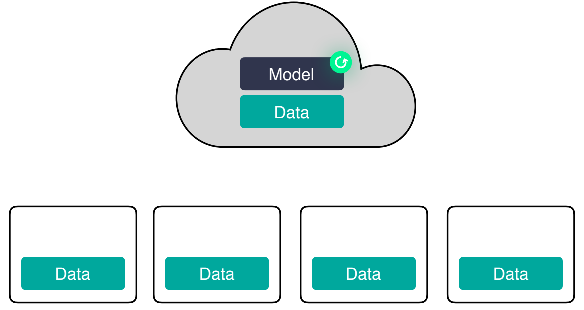
Set-up: (standard) centralized learning



- needs of lot of storage ☹... but efficient ☺

Collaborative learning

Set-up: (standard) centralized learning

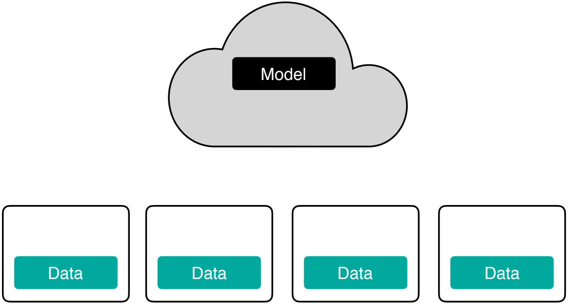


- needs of lot of storage ☹... but efficient ☺
- is highly privacy invasive (e.g. phones) ☹
- jeopardy on confidentiality/strategy (e.g. hospitals/companies) ☹

Move the model, not the data

Different set-up: Federative Learning

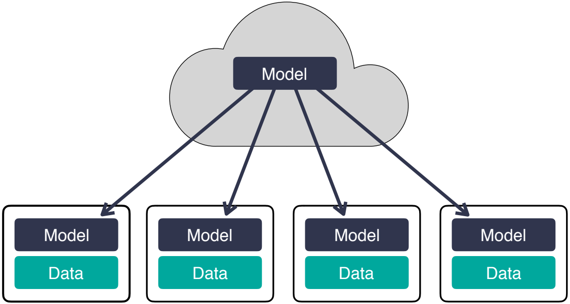
Decoupling the ability to learn a global model from moving local data



Move the model, not the data

Different set-up: Federative Learning

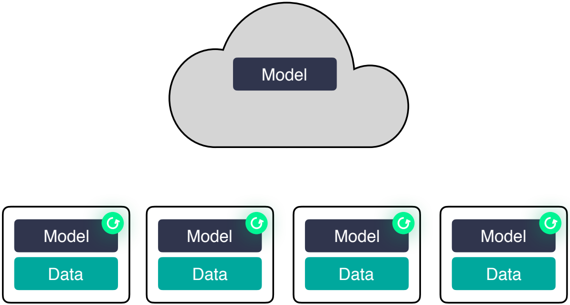
Decoupling the ability to learn a global model from moving local data



Move the model, not the data

Different set-up: Federative Learning

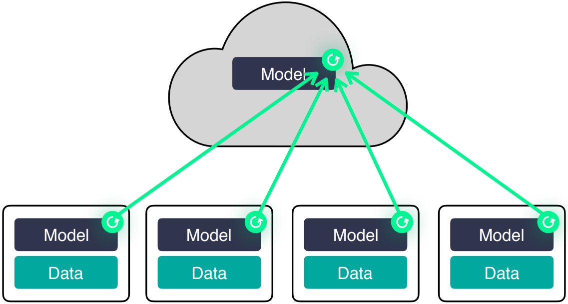
Decoupling the ability to learn a global model from moving local data



Move the model, not the data

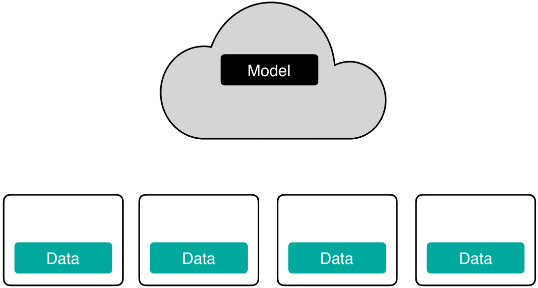
Different set-up: Federative Learning

Decoupling the ability to learn a global model from moving local data

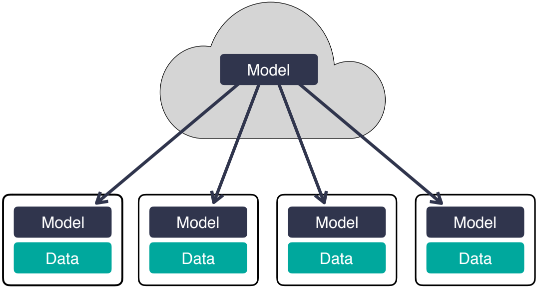


New (optimization) algorithm able to deal with heterogenous data/systems

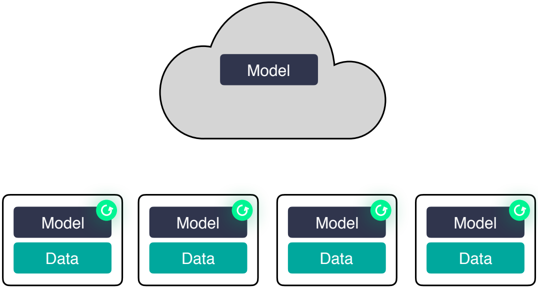
Zoom on a research topic: harnessing communications



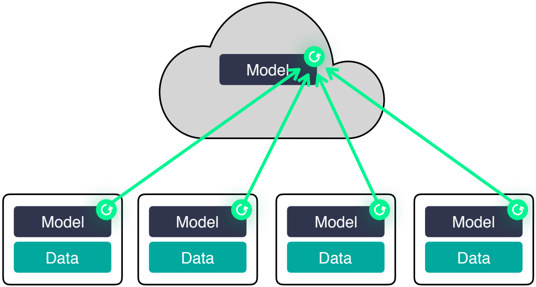
Zoom on a research topic: harnessing communications



Zoom on a research topic: harnessing communications

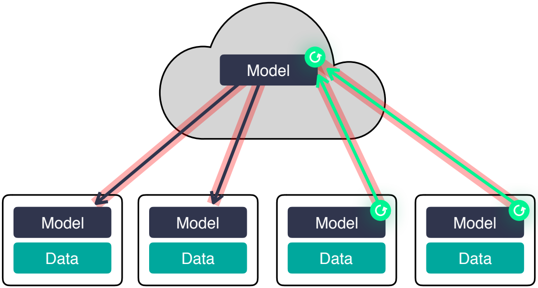


Zoom on a research topic: harnessing communications



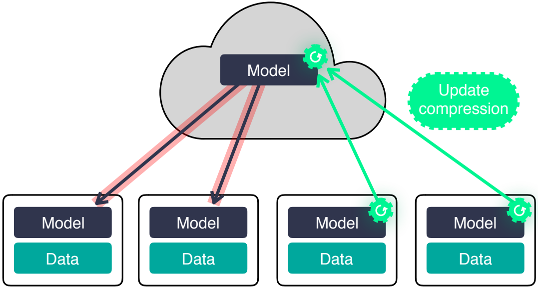
Zoom on a research topic: harnessing communications

Communication is the bottleneck ☹



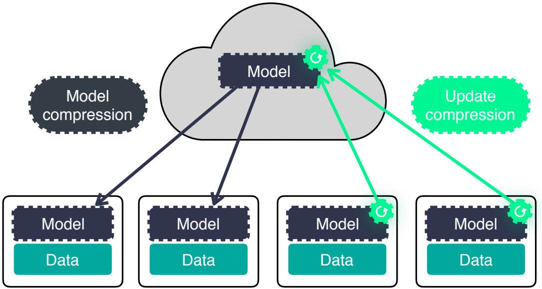
Zoom on a research topic: harnessing communications

Communication is the bottleneck ☹



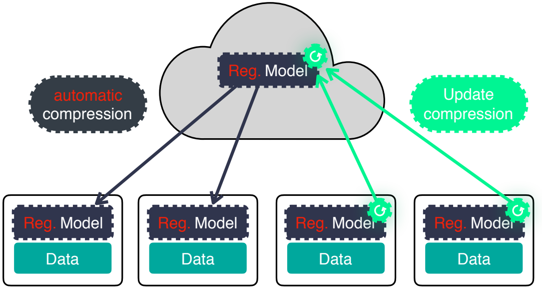
Zoom on a research topic: harnessing communications

Communication is the bottleneck ☹



Zoom on a research topic: harnessing communications

Communication is the bottleneck ☹

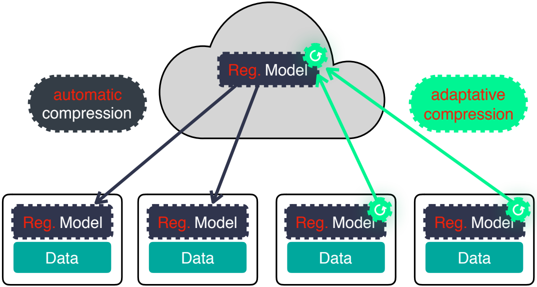


Our contribution: compression by nonsmooth regularization

- Observation: nonsmooth regularization gives automatic model compression
  E.g. for $r = \|\cdot\|_1$, model becomes sparse... just communicate nonzeros!

Zoom on a research topic: harnessing communications

Communication is the bottleneck ☹



Our contribution: compression by nonsmooth regularization

- Observation: nonsmooth regularization gives automatic model compression
  E.g. for $r = \|\cdot\|_1$, model becomes sparse... just communicate nonzeros!
- [Grishchenko, Iutzeler, M. '19] uses it for update comp.
  E.g. for $r = \|\cdot\|_1$, select current support + random entries

Illustration of compression by a nonsmooth regularizer

On an instance of TV-regularized logistic regression (a1a dataset on 10 machines)

$$\min_{x \in \mathbb{R}^d} \quad \frac{1}{n}\sum_{j=1}^{n} \log\left(1+\exp(-y_j\, a_j^\top x)\right) \;+\; \lambda\,\mathrm{TV}(x)$$

Total Variation

$$\mathrm{TV}(x) = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$$

Comparison of
- Usual algorithm (black)
- Our variant with compression (red)

Illustration of compression by a nonsmooth regularizer

On an instance of TV-regularized logistic regression (a1a dataset on 10 machines)

$$\min_{x\in\mathbb{R}^d} \quad \frac{1}{n}\sum_{j=1}^{n} \log\left(1+\exp(-y_j\, a_j{}^\top x)\right) \;+\; \lambda\,\mathrm{TV}(x)$$

Total Variation

$$\mathrm{TV}(x) = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$$

Comparison of
- Usual algorithm (black)
- Our variant with compression (red)

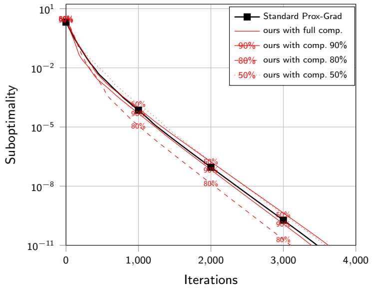


Illustration of compression by a nonsmooth regularizer

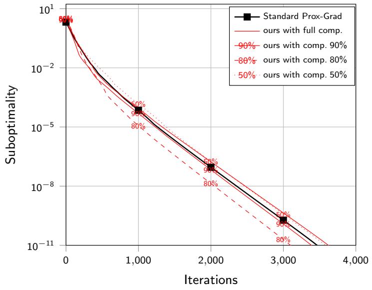
On an instance of TV-regularized logistic regression (a1a dataset on 10 machines)

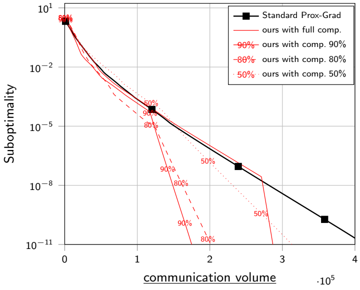
$$\min_{x\in\mathbb{R}^d} \quad \frac{1}{n}\sum_{j=1}^{n} \log\left(1+\exp(-y_j\, a_j^\top x)\right) \; + \; \lambda\,\mathrm{TV}(x)$$

Total Variation

$$\mathrm{TV}(x) = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$$

Comparison of
- Usual algorithm (black)
- Our variant with compression (red)

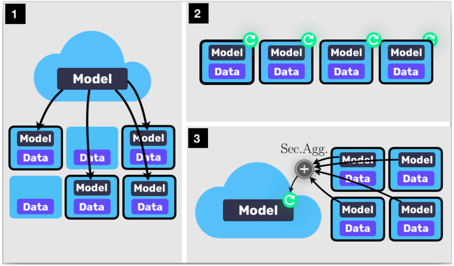




Acceleration... with respect to communication volume !

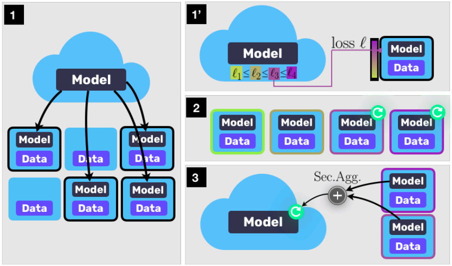
Zoom on a second topic: improving worst-case FedAvg

Federated Learning by Google = FedAvg



Zoom on a second topic: improving worst-case FedAvg

Federated Learning by Google = FedAvg vs Robust FedAvg

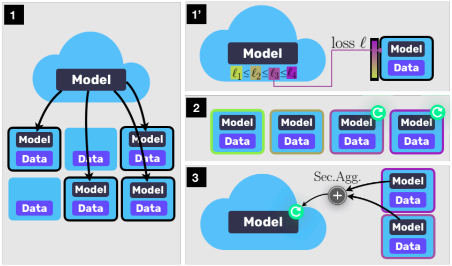


Our contribution: improve worst-case performance over users

- by adaptive filtering [Laguel, M., Harchaoui '19]

Zoom on a second topic: improving worst-case FedAvg

Federated Learning by Google = FedAvg vs Robust FedAvg



Our contribution: improve worst-case performance over users

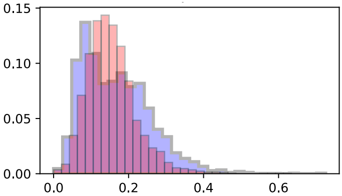
- by adaptive filtering [Laguel, M., Harchaoui '19]

Example: classification task

Standard vs Ours

(using ConvNet on EMNIST dataset)

Histogram over users
of test misclassification error



Conclusion

Take-home message

- AI is a rich, active, visible field of research and developments
- Next step: Operational AI (?) (not much discussed today... except in the title!)
- There are many problems involving uncertainty, decision-making, robustness and scale... far from being solved

  (not to mention economic, social and legal issues...)

Two theoretical questions

- understanding convergence towards good equilibrium
- models and algorithms in federated learning

  compression by nonsmooth regularization, improvement of worst-case

Conclusion

Take-home message

- AI is a rich, active, visible field of research and developments
- Next step: Operational AI (?) (not much discussed today... except in the title!)
- There are many problems involving uncertainty, decision-making, robustness and scale... far from being solved

  (not to mention economic, social and legal issues...)

Two theoretical questions

- understanding convergence towards good equilibrium
- models and algorithms in federated learning

  compression by nonsmooth regularization, improvement of worst-case

thanks !!