Deep learning theory
for power-efficient algorithms

Sébastien Loustau
j.w.w. A. Chee (Cornell, Ithaca), and P. Gay (team member)

November, 29th, 2021
Team ApproxBayes, RIKEN AIP

Deep learning theory
for power-efficient algorithms

Sébastien Loustau
j.w.w. A. Chee (Cornell, Ithaca), and P. Gay (team member)

GreenAI
U · P · P · A

November, 29th, 2021

Outlines

1. Gentle start with gradient and mirror descent
2. First application: how to learn sparse deep nets
3. Extension to the power metrical task problem

Outlines

Convexity and gradient

Let $f : K \subset \mathbb{R}^p \to \mathbb{R}$ a convex function on a convex body.

Convexity and gradient

Let $f : K \subset \mathbb{R}^p \to \mathbb{R}$ a convex function on a convex body.
If f is differentiable, $\forall x, y \in K$,

$$f(y) \geq f(x) + \nabla f(x) \cdot (y - x).$$

Convexity and gradient

Let $f : K \subset \mathbb{R}^p \to \mathbb{R}$ a convex function on a convex body.
If f is differentiable, $\forall x, y \in K$,

$$f(y) \geq f(x) + \nabla f(x) \cdot (y - x).$$

For $y = \arg\min_K f(x)$, we have :

$$-\nabla f(x) \cdot (y - x) \geq 0.$$

Convexity and gradient

Let $f : K \subset \mathbb{R}^p \to \mathbb{R}$ a convex function on a convex body.
If f is differentiable, $\forall x, y \in K$,

$$f(y) \geq f(x) + \nabla f(x) \cdot (y - x).$$

For $y = \arg\min_K f(x)$, we have :

$$-\nabla f(x) \cdot (y - x) \geq 0.$$

The gradient flow $\frac{d}{dt} x_t = -\nabla f(x_t)$ is suitable for convex opt

Gradient descent

Theorem

Under the previous assumption, the discretized version

$$x_{t+1} = x_t - \eta \nabla f(x_t),\; t = 1, \ldots, T, \tag{1}$$

satisfies:

$$\frac{1}{T}\sum_{t=1}^{T} f(x_t) - f(y) \leq \frac{\|y - x_1\|^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T} \|\nabla f(x_t)\|^2.$$

Theorem

Under the previous assumption, the discretized version

$$x_{t+1} = x_t - \eta \nabla f(x_t),\ t = 1, \dots, T, \tag{1}$$

satisfies:

$$\frac{1}{T}\sum_{t=1}^{T} f(x_t) - f(y) \leq \frac{\|y - x_1\|^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T} \|\nabla f(x_t)\|^2.$$

Proof.

The drop at time t satisfies:

$$\|x_{t+1} - y\|^2 - \|x_t - y\|^2 = -2\eta(x_t - y)\nabla f(x_t) + \eta^2\|\nabla f(x_t)\|^2.$$

□

Extension to non-euclidean settings

Gradient descent (1) can be written as:

$$x_{t+1} := \arg\min_{x \in K} \left\{ \eta \nabla f(x_t) \cdot x + \frac{\|x - x_t\|^2}{2} \right\}.$$

Extension to non-euclidean settings

Gradient descent (1) can be written as:

$$x_{t+1} := \arg\min_{x \in K} \left\{ \eta \nabla f(x_t) \cdot x + \frac{\|x - x_t\|^2}{2} \right\}.$$

$\Rightarrow$ no localization and pure Euclidean setting

Mirror descent

Mirror descent solves:

$$x_{t+1} := \arg\min_{x \in K} \left\{ \eta \nabla f(x_t) \cdot x + \mathcal{B}_\Phi(x, x_t) \right\}, \tag{2}$$

Mirror descent

Mirror descent solves:

$$x_{t+1} := \arg\min_{x \in K} \left\{ \eta \nabla f(x_t) \cdot x + \mathcal{B}_\Phi(x, x_t) \right\}, \tag{2}$$

- Right dual form $\nabla \Phi(x_{t+1}) = \nabla \Phi(x_t) - \eta \nabla f(x_t)$,

Mirror descent

Mirror descent solves:

$$x_{t+1} := \arg\min_{x \in K} \left\{ \eta \nabla f(x_t) \cdot x + \mathcal{B}_\Phi(x, x_t) \right\}, \qquad (2)$$

- Right dual form $\nabla \Phi(x_{t+1}) = \nabla \Phi(x_t) - \eta \nabla f(x_t)$,
- For $\Phi(x) = \frac{\|x\|^2}{2}$, (2) $\Leftrightarrow$ (1),

Mirror descent

Mirror descent solves:

$$x_{t+1} := \arg\min_{x \in K} \left\{ \eta \nabla f(x_t) \cdot x + \mathcal{B}_\Phi(x, x_t) \right\}, \tag{2}$$

- Right dual form $\nabla \Phi(x_{t+1}) = \nabla \Phi(x_t) - \eta \nabla f(x_t)$,
- For $\Phi(x) = \frac{\|x\|^2}{2}$, (2) $\Leftrightarrow$ (1),
- $\mathcal{B}_\Phi(x, x_t) = \|x - x_t\|^2_{\nabla^2 \Phi(\omega_t)}$ by Taylor approximation,

Mirror descent

Mirror descent solves:

$$x_{t+1} := \arg\min_{x \in K} \left\{ \eta \nabla f(x_t) \cdot x + \mathcal{B}_\Phi(x, x_t) \right\}, \tag{2}$$

- Right dual form $\nabla\Phi(x_{t+1}) = \nabla\Phi(x_t) - \eta\nabla f(x_t)$,
- For $\Phi(x) = \frac{\|x\|^2}{2}$, (2) $\Leftrightarrow$ (1),
- $\mathcal{B}_\Phi(x, x_t) = \|x - x_t\|^2_{\nabla^2\Phi(\omega_t)}$ by Taylor approximation,
- Next: (2) with $\Phi(\rho) = \int \rho \log \rho$, then $\mathcal{B}_\Phi(\rho, \pi) = \mathcal{K}(\rho, \pi)$ and we get for instance Bayesian updating.

Outlines

Considering a deterministic set $\{z_t, t = 1, \ldots, T\}$, a set of experts $\mathcal{G}$ and a loss function, we want to build a **sequence of distributions** $(\rho_t)_{t=1}^T$ on $\mathcal{G}$ satisfying:

$$\sum_{t=1}^{T} \mathbb{E}_{g\sim\rho_t} \ell(g, z_t) \leq \inf_{g\in\mathcal{G}} \left\{ \sum_{t=1}^{T} \ell(g, z_t) + \text{pen}(g) \right\} + \Delta_T,$$

where

- $\text{pen}(g)$ measures the **complexity** of the network,
- $\Delta_T > 0$ is at least sublinear.

- $z = (x, y)$, $x \in \mathcal{X}$ input space of images, time series, network,
- the **cross-entropy** loss function $\ell(\hat{y}, y)$,
- $\mathcal{G} := \{g_{\mathbf{w}} : \mathcal{X} \rightarrow \mathcal{Y}, \mathbf{w} \in \mathcal{W}\}$, where **w** are the weights of a given CNNs architecture or set of architectures,

- $z = (x, y)$, $x \in \mathcal{X}$ input space of images, time series, network,
- the **cross-entropy** loss function $\ell(\hat{y}, y)$,
- $\mathcal{G} := \{g_{\mathbf{w}} : \mathcal{X} \to \mathcal{Y}, \mathbf{w} \in \mathcal{W}\}$, where **w** are the weights of a given CNNs architecture or set of architectures,
- $\mathcal{G} := \{g_{\mathbf{w}} : \mathcal{X} \to \mathcal{Y}, \mathbf{w} \in \mathcal{W}\}$ is a set of XNOR-nets architecture. For XNOR-nets convolutions are approximated by bitwise operations:

$$x_k = \left(\mathbf{w}_k^{\text{bin}} \bigoplus \text{sign} \circ \text{BNorm}\left(x_{k-1}\right)\right) \bigodot \mathbf{w}_k^{\text{scale}}.$$

Theorem

Considering inputs $\{(x_t, y_t), t = 1, \ldots, T\}$*, the decision space* $\mathcal{G}$*, and cross-entropy loss, there exists a* ***sequence of distributions*** $(\rho_t)_{t=1}^T$ *on* $\mathcal{G}$ *such that:*

$$\sum_{t=1}^{T} \mathbb{E}_{g' \sim \rho_t} \ell(y_t, g'(x_t)) \leq \inf_{\mathbf{w} \in \mathcal{W}} \left\{ \sum_{t=1}^{T} \ell(y_t, g_{\mathbf{w}}(x_t)) + pen(g_{\mathbf{w}}) \right\} + \Delta_T,$$

where $\Delta_T > 0$ *is optimal and* $pen(g_{\mathbf{w}})$ *measures the complexity of the network as follows:*

$$pen(g_{\mathbf{w}}) = 4\|\mathbf{w}\|_0 \log \left(1 + \frac{\|\mathbf{w}\|_1}{\tau \|\mathbf{w}\|_0} \right)$$

Sparsity regret bound

Proof.

The proof is based on two facts:

- A PAC-Bayesian bound due to [Audibert, 2009]:

$$\sum_{t=1}^{T} \mathbb{E}_{g\sim\rho_t} \ell(g, z_t) \leq \inf_{\rho\in\mathcal{P}(\mathcal{G})} \left\{ \mathbb{E}_{g\sim\rho} \sum_{t=1}^{T} \overline{\ell}(g, z_t) + \frac{\mathcal{K}(\rho,\pi)}{\lambda} \right\},$$

  where $\overline{\ell}(y, g(x)) = \ell(y, g(x)) + \frac{\lambda}{2} \left(\ell(y, g(x) - \ell(y, \hat{g}_t(x))\right)^2$ satisfies a mixability condition,

Proof.

The proof is based on two facts:

- A PAC-Bayesian bound due to [Audibert, 2009]:

$$\sum_{t=1}^{T} \mathbb{E}_{g \sim \rho_t} \ell(g, z_t) \leq \inf_{\rho \in \mathcal{P}(\mathcal{G})} \left\{ \mathbb{E}_{g \sim \rho} \sum_{t=1}^{T} \bar{\ell}(g, z_t) + \frac{\mathcal{K}(\rho, \pi)}{\lambda} \right\},$$

  where $\bar{\ell}(y, g(x)) = \ell(y, g(x)) + \frac{\lambda}{2} \left(\ell(y, g(x) - \ell(y, \hat{g}_t(x)) \right)^2$ satisfies a mixability condition,

- The choice of a power law π such that:

$$\mathcal{K}(\pi_{\mathbf{w}}, \pi) = 4 \|\mathbf{w}\|_0 \log \left(1 + \frac{\|\mathbf{w}\|_1}{\tau \|\mathbf{w}\|_0} \right),$$

  where $\pi_{\mathbf{w}}$ is a translated version of π.

□

Sparsity regret bound

XNOR-Nets case

Theorem

Considering inputs $\{(x_t, y_t), t = 1, \ldots, T\}$, the decision space $\mathcal{G}$, and cross-entropy loss, there exists a sequence of distributions $(\rho_t)_{t=1}^{T}$ on $\mathcal{G}$ such that:

$$\sum_{t=1}^{T} \mathbb{E}_{g' \sim \rho_t} \ell(y_t, g'(x_t)) \leq \inf_{\mathbf{w} \in \mathcal{W}_{\mathrm{XNOR}}} \left\{ \sum_{t=1}^{T} \ell(y_t, g_{\mathbf{w}}(x_t)) + pen(g_{\mathbf{w}}) \right\} + \Delta_T,$$

where $\Delta_T > 0$ is optimal and $pen(g_{\mathbf{w}})$ measure the complexity of the network as follows:

$$pen(g_{\mathbf{w}}) = 4 \sum_{\mathbf{w} \in \{\mathbf{w}^{\mathrm{real}}, \mathbf{w}^{\mathrm{scale}}\}} \|\mathbf{w}\|_0 \log \left(1 + \frac{\|\mathbf{w}\|_1}{\tau \|\mathbf{w}\|_0} \right) + p_{\mathrm{bin}} \log 2$$

Hyper-parameters : sparsity prior $\pi \in \mathcal{P}(\mathcal{G})$. Parameter $\lambda > 0$.

- Observe x_1 and draw $\hat{y}_1 = g_{\hat{\mathbf{w}}_1}(x_1)$ where $\hat{\mathbf{w}}_1 \sim \rho_1 := \pi$.
- For $t = 1, \ldots, T-1$:
    - Observe y_t and draw $\hat{y}_{t+1} = g_{\hat{\mathbf{w}}_{t+1}}(x_{t+1})$ where:

$$\hat{\mathbf{w}}_{t+1} \sim \exp\left\{-\lambda \sum_{u=1}^{t} \bar{\ell}(y_u, g_{\mathbf{w}}(x_u))\right\} d\pi(\mathbf{w}).$$

Challenging sampling problem

From the theoretical part, we want to sample from:

$$d\rho_T(\mathbf{w}) \sim \exp\left\{-\lambda \sum_{t=1}^{T} \ell(y_t, g_{\mathbf{w}}(x_t))\right\} d\pi(\mathbf{w}),$$

where prior $\pi \in \mathcal{P}(\mathcal{W})$ is a mixture of sparsity priors related with CNNs architectures.

Problem dimension of $\mathcal{W}$ is huge (from 60k to 150M parameters)

Greedy (RJ)-MCMC algorithm

Initialization : $\mathbf{w}_1 \sim \pi$. Parameter $\lambda > 0$.

For $m = 1, \dots M$ **do**

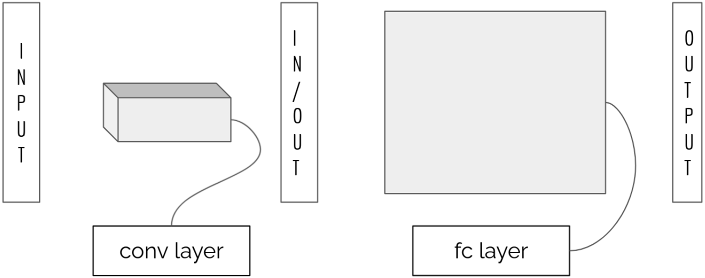
For $k = 1, \dots, N$ **do**

- Pick a layer $\ell \in \{1, \dots, L\}$ at random,
- Propose $\tilde{\mathbf{w}} \sim p(\cdot|\mathbf{w}_k)$,
- Accept $\mathbf{w}_{k+1} = \tilde{w}$ with proba:

$$\rho = \frac{\exp\{-\lambda \sum_{t \in \mathcal{I}_m} \ell(y_t, g_{\tilde{\mathbf{w}}}(x_t))\}}{\exp\{-\lambda \sum_{t \in \mathcal{I}_m} \ell(y_t, g_{\mathbf{w}_k}(x_t))\}} \frac{\pi(\tilde{\mathbf{w}})}{\pi(\mathbf{w}_k)}.$$

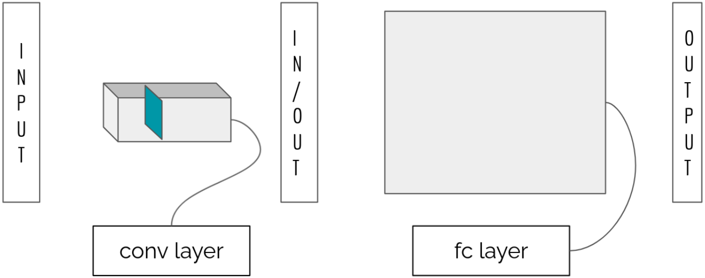
Greedy (RJ)-MCMC algorithm

Example on a simple CNN



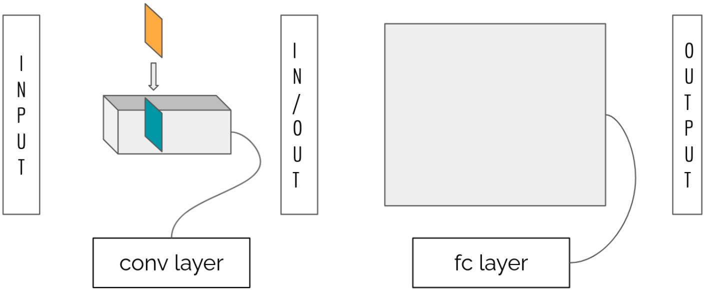
Greedy (RJ)-MCMC algorithm

Example on a simple CNN



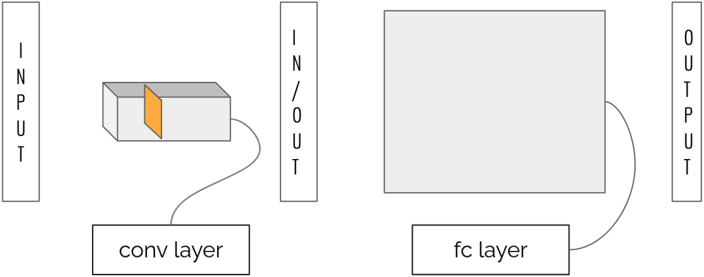
Greedy (RJ)-MCMC algorithm

Example on a simple CNN



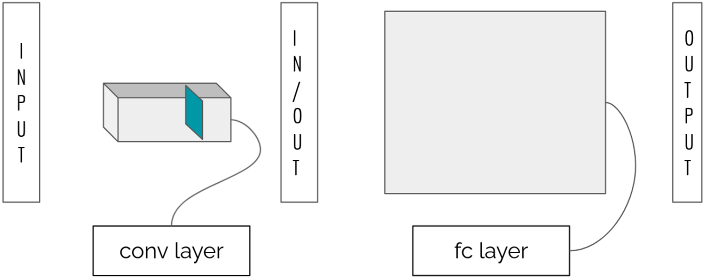
Greedy (RJ)-MCMC algorithm

Example on a simple CNN



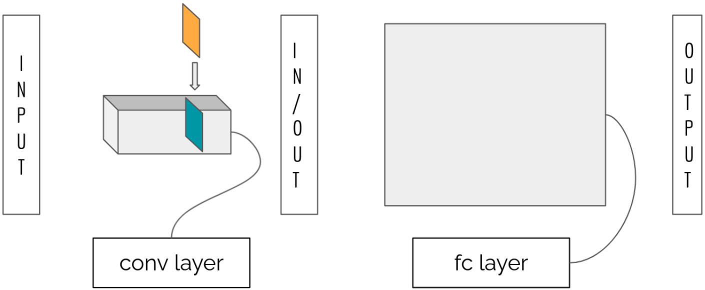
Greedy (RJ)-MCMC algorithm

Example on a simple CNN



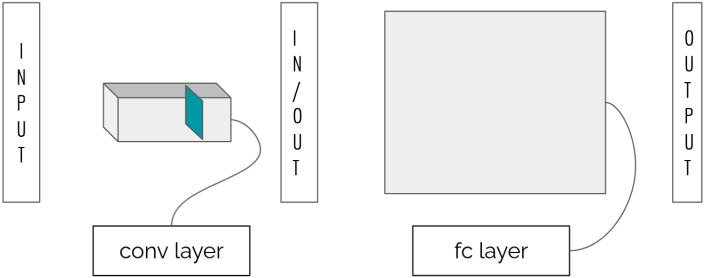
Greedy (RJ)-MCMC algorithm

Example on a simple CNN



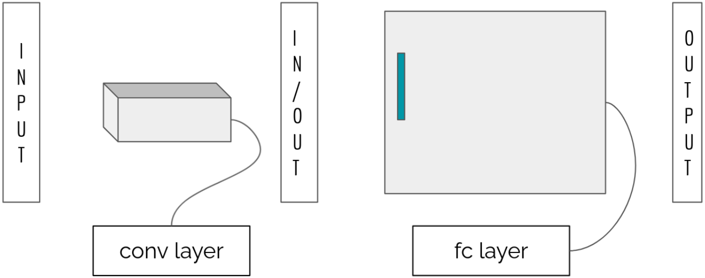
Greedy (RJ)-MCMC algorithm

Example on a simple CNN



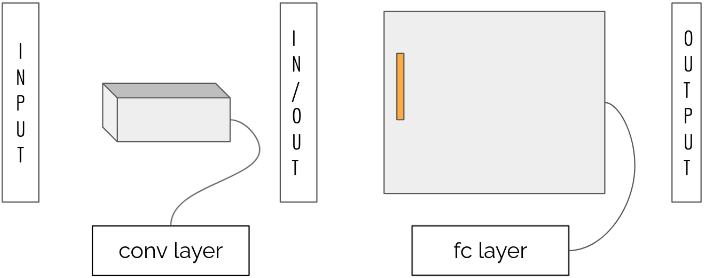
Greedy (RJ)-MCMC algorithm

Example on a simple CNN



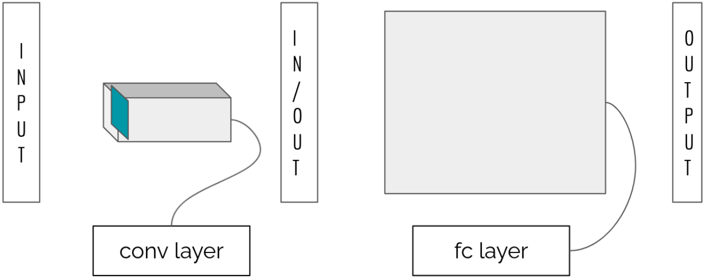
Greedy (RJ)-MCMC algorithm

Example on a simple CNN



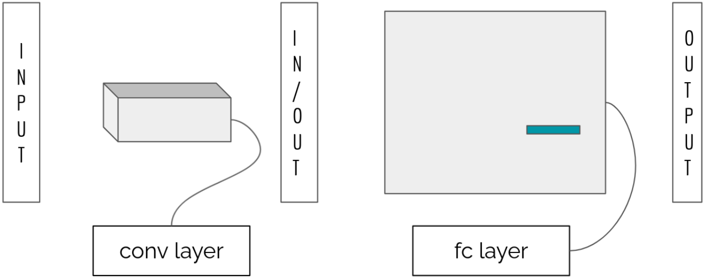
Greedy (RJ)-MCMC algorithm

Example on a simple CNN

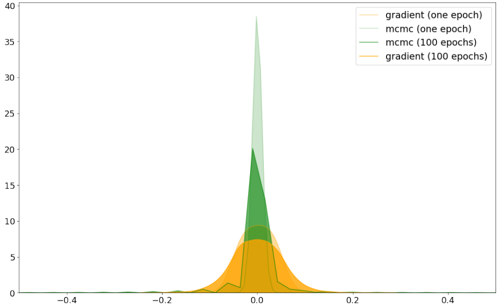


Greedy (RJ)-MCMC algorithm

Example on a simple CNN

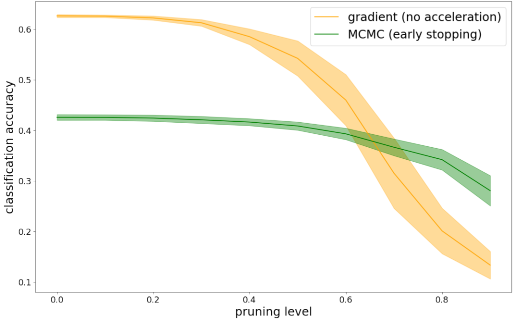


Resistence to pruning on CIFAR-10



- CNN with 60,000 params,
- SGD with batch size 256 and no acceleration,
- MCMC with 200 iterations by epoch.

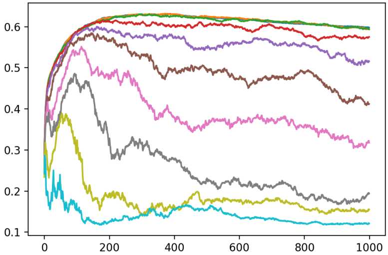
Resistence to pruning on CIFAR-10



- CNN with 60,000 params,
- SGD with batch size 256 and no acceleration,
- MCMC with 200 iterations by epoch.

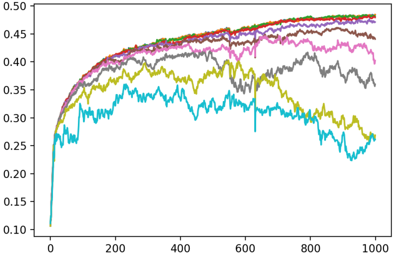
Resistence to pruning on CIFAR-10

stochastic gradient descent



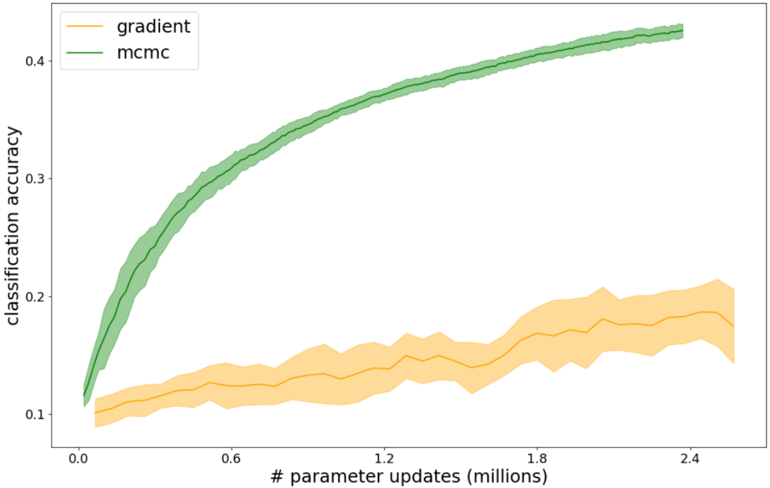
Resistence to pruning on CIFAR-10

mcmc algorithm



Lazy regime

gradient descent VS mcmc



Outlines

Motivation

How to consider a new metrical task ?

Motivation

How to consider a new metrical task ?

- add a cost to the loss function $\Rightarrow$ possible by non-differentiable programming,

Motivation

How to consider a new metrical task ?

- add a cost to the loss function $\Rightarrow$ possible by non-differentiable programming,
- put it directly at the core of the online decision,

Motivation

How to consider a new metrical task ?

- add a cost to the loss function $\Rightarrow$ possible by non-differentiable programming,
- put it directly at the core of the online decision,
- link with metrical task systems and power management.

Motivation

How to consider a new metrical task ?

- add a cost to the loss function $\Rightarrow$ possible by non-differentiable programming,
- put it directly at the core of the online decision,
- link with metrical task systems and power management.

From mirror descent to Optimal transport

Mirror descent solves:

$$\rho_{t+1} := \arg \min_{\rho \in \mathcal{P}(\mathcal{G})} \left\{ \eta \langle \nabla f(\rho_t), \rho \rangle + \mathcal{B}_\Phi(\rho, \rho_t) \right\} .$$

From mirror descent to Optimal transport

Mirror descent solves:

$$\rho_{t+1} := \arg \min_{\rho \in \mathcal{P}(\mathcal{G})} \left\{ \eta \langle \nabla f(\rho_t), \rho \rangle + \mathcal{B}_\Phi(\rho, \rho_t) \right\} .$$

- $\Phi(x) \approx \|x\|^2 \Rightarrow$ no localization,

From mirror descent to Optimal transport

Mirror descent solves:

$$\rho_{t+1} := \arg\min_{\rho \in \mathcal{P}(\mathcal{G})} \left\{ \eta \langle \nabla f(\rho_t), \rho \rangle + \mathcal{B}_\Phi(\rho, \rho_t) \right\}.$$

- $\Phi(x) \approx \|x\|^2 \Rightarrow$ no localization,
- $\Phi(\rho) = \int \rho \log \rho \Rightarrow$ sparsity,

From mirror descent to Optimal transport

Mirror descent solves:

$$\rho_{t+1} := \arg\min_{\rho \in \mathcal{P}(\mathcal{G})} \left\{ \eta \langle \nabla f(\rho_t), \rho \rangle + \mathcal{B}_{\Phi}(\rho, \rho_t) \right\}.$$

- $\Phi(x) \approx \|x\|^2 \Rightarrow$ no localization,
- $\Phi(\rho) = \int \rho \log \rho \Rightarrow$ sparsity,

$$\rho_t \overset{\text{Joules?}}{\to} \rho_{t+1}$$

From mirror descent to Optimal transport

Mirror descent solves:

$$\rho_{t+1} := \arg \min_{\rho \in \mathcal{P}(\mathcal{G})} \left\{ \eta \langle \nabla f(\rho_t), \rho \rangle + \mathcal{B}_\Phi(\rho, \rho_t) \right\}.$$

- $\Phi(x) \approx \|x\|^2 \Rightarrow$ no localization,
- $\Phi(\rho) = \int \rho \log \rho \Rightarrow$ sparsity.

$$\rho_t \overset{\text{Joules?}}{\longrightarrow} \rho_{t+1}$$

Optimal transport

Consider the sequence $(\rho_t)_{t=1}^{T}$ defined as:

$$\rho_{t+1} := \arg\min_{\rho \in \mathcal{P}(\mathcal{G})} \left\{ \mathbb{E}_{g \sim \rho} \bar{\ell}(g, z_t) + \frac{\mathcal{W}_\alpha(\rho, \rho_t)}{\lambda} \right\}, \qquad (3)$$

where $\bar{\ell}(g, z_t) = \ell(g, z_t) + \delta_t(\alpha, \lambda)$.

Optimal transport

Consider the sequence $(\rho_t)_{t=1}^T$ defined as:

$$\rho_{t+1} := \arg\min_{\rho \in \mathcal{P}(\mathcal{G})} \left\{ \mathbb{E}_{g \sim \rho} \bar{\ell}(g, z_t) + \frac{\mathcal{W}_\alpha(\rho, \rho_t)}{\lambda} \right\}, \qquad (3)$$

where $\bar{\ell}(g, z_t) = \ell(g, z_t) + \delta_t(\alpha, \lambda)$.
Idea : replace $\mathcal{B}_\Phi(\rho, \pi)$ by a $\mathcal{W}_\alpha(\rho, \pi)$, strictly convex perturbation of the original optimal transport defined as:

$$\mathcal{W}_\alpha(\rho, \pi) := \min_{\Lambda \in \Delta(\rho, \pi)} \left\{ \int_{\mathcal{G} \times \mathcal{G}} C(g, g') d\Lambda(g, g') - \alpha \mathcal{H}(\Lambda) \right\},$$

for some $\alpha > 0$ and cost $C : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$.

Optimal transport theorem

Theorem

Assume $\mathcal{G}$ is finite and let $T, \lambda > 0$. Let $z_1, \dots, z_T$ deterministic data. Then $\forall \pi \in \mathcal{P}(\mathcal{G})$, $(\rho_t)_{t=1}^T$ based on (3) is such that :

$$\sum_{t=1}^T \mathbb{E}_{g \sim \Pi(\rho_t)} \ell(g, z_t) \leq \inf_{\rho \in \mathcal{P}(\mathcal{G})} \left\{ \mathbb{E}_{g \sim \rho} \sum_{t=1}^T \bar{\ell}(g, z_t) + \frac{\mathcal{W}_\alpha(\rho, \pi)}{\lambda} \right\} + \Delta_{T,\lambda},$$

where $\Delta_{T,\lambda} > 0$ and $\Pi : \mathcal{P}(\mathcal{G}) \to \mathcal{P}(\mathcal{G})$ is defined as:

$$d\Pi(\rho_t)(g) = A(\rho_t) \mathbb{E}_{g' \sim \rho_t} \exp\left\{ -\frac{C(g, g')}{\alpha} \right\}.$$

Proof.

- **new mixability condition** $\exists \delta_{\lambda,\alpha} : \forall \pi, \exists \Pi(\pi) : \forall z,$

$$\mathbb{E}_{g' \sim \Pi(\rho)} \ell(g', z) \leq \mathbb{E}_{g' \sim \Pi(\rho)} \min_{\rho} \left\{ \mathbb{E}_{g \sim \rho} \bar{\ell}(g, z) + \frac{\mathcal{W}_\alpha(\rho, \pi)}{\lambda} \right\},$$

where $\bar{\ell} = \ell(g, z) + \delta_{\lambda,\alpha}(g, g')$.

Proof.

- **new mixability condition** $\exists \delta_{\lambda,\alpha} : \forall \pi, \exists \Pi(\pi) : \forall z$,

$$\mathbb{E}_{g' \sim \Pi(\rho)} \ell(g', z) \leq \mathbb{E}_{g' \sim \Pi(\rho)} \min_{\rho} \left\{ \mathbb{E}_{g \sim \rho} \bar{\ell}(g, z) + \frac{\mathcal{W}_\alpha(\rho, \pi)}{\lambda} \right\},$$

  where $\bar{\ell} = \ell(g, z) + \delta_{\lambda,\alpha}(g, g')$.
- generalized PAC-Bayesian bound with $\mathcal{B}_\Phi$,

Proof.

- **new mixability condition** $\exists \delta_{\lambda,\alpha} : \forall \pi, \exists \Pi(\pi) : \forall z$,

$$\mathbb{E}_{g' \sim \Pi(\rho)} \ell(g', z) \leq \mathbb{E}_{g' \sim \Pi(\rho)} \min_{\rho} \left\{ \mathbb{E}_{g \sim \rho} \bar{\ell}(g, z) + \frac{\mathcal{W}_\alpha(\rho, \pi)}{\lambda} \right\},$$

  where $\bar{\ell} = \ell(g, z) + \delta_{\lambda,\alpha}(g, g')$.
- generalized PAC-Bayesian bound with $\mathcal{B}_\Phi$,
- applied for $\Phi(\cdot) = \mathcal{W}_\alpha(\cdot, \nu)$.

□

Corollary

Let $\pi = \delta_{g_\eta^\star}$ *the Dirac measure on the unique minimizer:*

$$g_\eta^\star := \arg\min_{g\in\mathcal{G}} \left\{ \mathrm{Err}_i + \eta \mathrm{Env}_i \right\}.$$

Consider minimization (3) *with* $C(g_i, g_j) := C(\mathrm{Env_i}, \mathrm{Env}_j)$ *we have:*

$$\sum_{t=1}^{T} \mathbb{E}_{\hat{g}_t\sim\tilde{\rho}_t} \ell(\hat{g}_t, z_t) \leq \min_{g\in\mathcal{G}} \left\{ \sum_{t=1}^{T} \bar{\ell}(g, z_t) + \frac{C(g, g_{i^\star})}{\lambda} \right\} + \Delta_T.$$

Link with metrical task systems

Let X a finite metric space of size n.
At every time step $t = 1, \dots, T$:

- the player receive a task function $c_t : X \to \mathbb{R}_+$,
- moove from state s_{t-1} to s_t and pay the movement cost $d(s_{t-1}, s_t)$ and the service cost $c_t(s_t)$.

Link with metrical task systems

Let X a finite metric space of size n.
At every time step $t = 1, \dots, T$:

- the player receive a task function $c_t : X \to \mathbb{R}_+$,
- moove from state s_{t-1} to s_t and pay the movement cost $d(s_{t-1}, s_t)$ and the service cost $c_t(s_t)$.

$\Rightarrow$ Many applications like power management,

Link with metrical task systems

Let X a finite metric space of size n.
At every time step $t = 1, \ldots, T$:

- the player receive a task function $c_t : X \to \mathbb{R}_+$,
- moove from state s_{t-1} to s_t and pay the movement cost $d(s_{t-1}, s_t)$ and the service cost $c_t(s_t)$.

$\Rightarrow$ Many applications like power management,
$\Rightarrow$ Link with PEA,

Link with metrical task systems

Let X a finite metric space of size n.
At every time step $t = 1, \dots, T$:

- the player receive a task function $c_t : X \to \mathbb{R}_+$,
- moove from state s_{t-1} to s_t and pay the movement cost $d(s_{t-1}, s_t)$ and the service cost $c_t(s_t)$.

$\Rightarrow$ Many applications like power management,
$\Rightarrow$ Link with PEA,
$\Rightarrow$ New competitive ratios instead of regret (OPT is mooving)

Link with metrical task systems

Let X a finite metric space of size n.
At every time step $t = 1, \dots, T$:

- the player receive a task function $c_t : X \to \mathbb{R}_+$,
- moove from state s_{t-1} to s_t and pay the movement cost $d(s_{t-1}, s_t)$ and the service cost $c_t(s_t)$.

$\Rightarrow$ Many applications like power management,
$\Rightarrow$ Link with PEA,
$\Rightarrow$ New competitive ratios instead of regret (OPT is mooving)
$\Rightarrow$ Optimal bound for stochastic algorithm is an open problem.

Concluding remarks

Summary

- a new optimizer based on theoretical framework,
- uses sparsity to get robustness to pruning,
- extend previous PAC-Bayesian approach to Bregman and Optimal Transport.

Concluding remarks

Summary

- a new optimizer based on theoretical framework,
- uses sparsity to get robustness to pruning,
- extend previous PAC-Bayesian approach to Bregman and Optimal Transport.

Open problems

- scale this new optimizer to imagenet,
- propose a power managed deep learning method at inference,
- introduce step by step the electricity constraints into the online decision.

THANK YOU

- AIPowerMeter software,
- More materials on ACML workshop organized here,
- Mathematical contents [1] for sparsity and [2] for Bregman and Optimal Transport,
- Website of the whole project.

THANK YOU

- AIPowerMeter software,
- More materials on ACML workshop organized here,
- Mathematical contents [1] for sparsity and [2] for Bregman and Optimal Transport,
- Website of the whole project.