Adaptation and Universality in First Order Optimization

Volkan Cevher
https://lions.epfl.ch

Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)
Switzerland

OPT-ML 2020

[11 December 2020]

Joint work with

Ahmet Alacaoglu, Francis Bach, Ali Kavis,
Kfir Levy, Yurii Malitskyi, Panayotis Mertikopoulos, Alp Yurtsever

lions@epfl

One formula to rule all machine learning problems

$$f^\star = \min_{x : x \in \mathcal{X}} f(x) \quad (\text{argmin} \to x^\star)$$

◦ Growing interest in first-order gradient methods[1] due to their scalability and generalization performance

[1] Lan, Guanghui. First-order and Stochastic Optimization Methods for Machine Learning. Springer Nature, 2020.

One formula to rule ~~all~~ some machine learning problems ...and one algorithm to solve them.

$$f^\star = \min_{x : x \in \mathcal{X}} f(x) \quad (\text{argmin} \to x^\star)$$

- Growing interest in first-order gradient methods[1] due to their scalability and generalization performance
- In the sequel,
  - the set $\mathcal{X}$ is convex and has a tractable projection operator $P_\mathcal{X}$
  - all convergence characterizations are with feasible iterates $x^k \in \mathcal{X}$
  - gradient mapping means $G_\eta(x^k) = \frac{1}{\eta}[x^k - P_\mathcal{X}(x^k - \eta \nabla f(x^k))]$, where η is the step-size
  - L-smooth means $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \forall x, y \in \mathcal{X}$
  - ∂ may refer to the generalized subdifferential, and $\delta_\mathcal{X}$ refers to the indicator function for the set $\mathcal{X}$

[1] Lan, Guanghui. First-order and Stochastic Optimization Methods for Machine Learning. Springer Nature, 2020.

Worst-case iteration complexities of classical projected first-order methods[1,2]

$f(x)$	gradient oracle	L-smooth	Stationarity measure	**GD/SGD**	**Accelerated GD/SGD**
Convex	stochastic	yes	$f(x^k) - f^\star =$	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$
Convex	deterministic	yes	$f(x^k) - f^\star =$	$\mathcal{O}\left(\frac{1}{k}\right)$	$\mathcal{O}\left(\frac{1}{k^2}\right)$
Convex	stochastic	no	$f(x^k) - f^\star =$	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$
Nonconvex	stochastic	yes	$\Vert G_\eta(x^k)\Vert^2 =$	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$[3]	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$[3]
Nonconvex	deterministic	yes	$\Vert G_\eta(x^k)\Vert^2 =$	$\mathcal{O}\left(\frac{1}{k}\right)$[4]	$\mathcal{O}\left(\frac{1}{k}\right)$[4]
Nonconvex	stochastic	no	$\mathrm{dist}(0, \partial(f(x^k) + \delta_{\mathcal{X}}(x^k)))^2 =$	?[356]	?[356]

- Basic structures, such as smoothness or strong convexity, help, but there are more structures that can be used:
  - $\max$-form, metric subregularity, Polyak-Lojasiewicz, Kurdyka-Lojasiewicz, weak convexity,[3] growth cond...

[1] Y. Nesterov, "Introductory lectures on convex optimization: A basic course," Springer Science, 2013.

[2] Y. Carmon, J.C. Duchi, O. Hinder, and A. Sidford, "Lower bounds for finding stationary points I–II." Mathematical Programming, 2019.

[3] D. Davis and D. Drusvyatskiy, "Stochastic model-based minimization of weakly convex functions," SIOPT, 2019.

[4] S. Ghadimi and G. Lan, "Accelerated gradient methods for nonconvex nonlinear and stochastic programming," MathProg, 2016.

[5] J. Zhang, et al., "On complexity of finding stationary points of nonsmooth nonconvex functions," arXiv:2002.04130, 2020.

[6] O. Shamir, "Can We Find Near-Approximately-Stationary Points of Nonsmooth Nonconvex Functions?" arXiv:2002.11962, 2020.

Worst-case iteration complexities of classical projected first-order methods[1,2]

$f(x)$	gradient oracle	L-smooth	Stationarity measure	**GD/SGD**	**Accelerated GD/SGD**
Convex	stochastic	yes	$f(x^k) - f^\star =$	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$
Convex	deterministic	yes	$f(x^k) - f^\star =$	$\mathcal{O}\left(\frac{1}{k}\right)$	$\mathcal{O}\left(\frac{1}{k^2}\right)$
Convex	stochastic	no	$f(x^k) - f^\star =$	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$
Nonconvex	stochastic	yes	$\Vert G_\eta(x^k)\Vert^2 =$	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$[3]	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$[3]
Nonconvex	deterministic	yes	$\Vert G_\eta(x^k)\Vert^2 =$	$\mathcal{O}\left(\frac{1}{k}\right)$[4]	$\mathcal{O}\left(\frac{1}{k}\right)$[4]
Nonconvex	stochastic	no	dist$(0, \partial(f(x^k) + \delta_{\mathcal{X}}(x^k)))^2 =$	?[3,5,6]	?[3,5,6]

at the end of the presentation

- Basic structures, such as smoothness or strong convexity, help, but there are more structures that can be used:
  - $\max$-form, metric subregularity, Polyak-Lojasiewicz, Kurdyka-Lojasiewicz, weak convexity,[3] growth cond...

[1] Y. Nesterov, "Introductory lectures on convex optimization: A basic course," Springer Science, 2013.

[2] Y. Carmon, J.C. Duchi, O. Hinder, and A. Sidford, "Lower bounds for finding stationary points I–II." Mathematical Programming, 2019.

[3] D. Davis and D. Drusvyatskiy, "Stochastic model-based minimization of weakly convex functions," SIOPT, 2019.

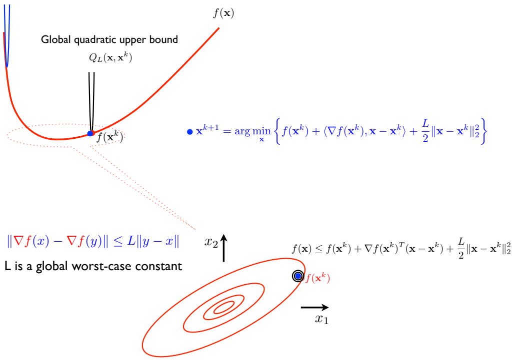
[4] S. Ghadimi and G. Lan, "Accelerated gradient methods for nonconvex nonlinear and stochastic programming," MathProg, 2016.

[5] J. Zhang, et al., "On complexity of finding stationary points of nonsmooth nonconvex functions," arXiv:2002.04130, 2020.

[6] O. Shamir, "Can We Find Near-Approximately-Stationary Points of Nonsmooth Nonconvex Functions?" arXiv:2002.11962, 2020.

Worst-case is often too pessimistic

◦ GD: $x^{k+1} = x^k - \frac{1}{L}\nabla f(x^k)$



◦ Rates are not everything!

- overall computational effort is what matters
- constants &implementations are key

◦ Knowledge of smoothness, the value of L,...

- challenging

◦ Must "somehow" adapt to a "different" function

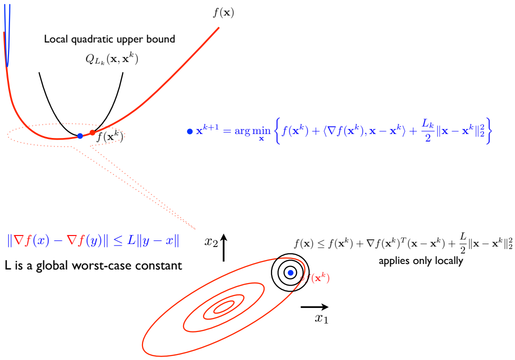
- online and without knowing L
- can reduce overall computational effort!

Warmup: f is convex

$$f^\star = \min_{x : x \in \mathcal{X}} f(x) \quad (\text{argmin} \to x^\star)$$

A classical approach: Line-search

◦ Long history: Backtracking, Armijo, steepest descent...



◦ Universal accelerated gradient method[1]

$$f(x^k) - f^\star = \mathcal{O}\left(\frac{L_\nu \|x^0 - x^\star\|^{1+\nu}}{k^{\frac{1+3\nu}{2}}} \right)$$

- adapts to Hölder smoothness ($\nu \in [0, 1]$)

$$\boxed{\|\nabla f(x) - \nabla f(y)\|_2 \leq L_\nu \|x - y\|_2^\nu}$$

- has extensions to primal-dual optimization[2]
- sets accuracy a priori & monotonic step-sizes

◦ Not as universal as we wish it to be

- different procedures for stochastic gradients[3]

[1] Y. Nesterov, "Universal Gradient Methods for Convex Optimization Problems," Mathematical Programming, 2015.
[2] A. Yurtsever, Q. Tran-Dinh, and V. Cevher, "A Universal Primal-Dual Convex Optimization Framework," NeurIPS, 2015.
[3] S. Vaswani et al., "Painless Stochastic Gradient: Interpolation, Line-Search, and Convergence Rates," NeurIPS, 2019.

A contemporary approach: Online convex optimization (OCO)

Algorithm: A basic online learning problem[1,2,3]

1: **for** $t = 1, \ldots, k$ **do**
2: Player chooses some action $x^t \in \mathcal{X} \subset \mathbb{R}^p$
3: Environment reveals a convex loss $f_t(\cdot)$
4: Player suffers the loss $f_t(x^t)$
5: **end for**

- Minimize the total loss vs the best action in hindsight:

$$R(k) = \sum_{t=1}^{k} f_t(x^t) - \min_{x \in \mathcal{X}} \sum_{t=1}^{k} f_t(x).$$

  - "somehow" adapts to a "different" function!

- For general convex f_t, optimal regret is sublinear:

$$R(k) = \mathcal{O}\left(\sqrt{k}\right).$$

- We can trivially convert regret to rate via $f_t = f$:

$$f\left(\frac{1}{k}\sum_{t=1}^{k} x^t\right) - f^\star \le \frac{R(k)}{k}.$$

[1] N. Cesa-Bianchi and G. Lugosi, "Prediction, learning, and games," Cambridge University Press, 2006.
[2] S. Shalev-Shwartz, "Online learning and online convex optimization," Found. Trends Mach. Learn., 2012.
[3] E. Hazan, "Introduction to online convex optimization," arXiv:1909.05207, 2019.

A contemporary approach: Online convex optimization (OCO)

Algorithm: A basic online learning problem[1,2,3]

1: **for** $t = 1, \ldots, k$ **do**
2: Player chooses some action $x^t \in \mathcal{X} \subset \mathbb{R}^p$
3: Environment reveals a convex loss $f_t(\cdot)$
4: Player suffers the loss $f_t(x^t)$
5: **end for**

- One procedure to rule them all...
  - smooth, non-smooth, stochastic!
- Not as adaptive as we like in optimization
  - The "offline" fast rate $1/k^2$ is not immediate

- Minimize the total loss vs the best action in hindsight:

$$R(k) = \sum_{t=1}^{k} f_t(x^t) - \min_{x \in \mathcal{X}} \sum_{t=1}^{k} f_t(x).$$

  - "somehow" adapts to a "different" function!

- For general convex f_t, optimal regret is sublinear:

$$R(k) = \mathcal{O}\left(\sqrt{k}\right).$$

- We can trivially convert regret to rate via $f_t = f$:

$$f\left(\frac{1}{k}\sum_{t=1}^{k} x^t\right) - f^\star \leq \frac{R(k)}{k}.$$

[1] N. Cesa-Bianchi and G. Lugosi, "Prediction, learning, and games," Cambridge University Press, 2006.
[2] S. Shalev-Shwartz, "Online learning and online convex optimization," Found. Trends Mach. Learn., 2012.
[3] E. Hazan, "Introduction to online convex optimization," arXiv:1909.05207, 2019.

The curious case of AdaGrad[1]

Algorithm: AdaGrad (scalar)[2]

Input: Iterations k; $x_0 \in \mathcal{X}$
for $t = 0, ..., k-1$ **do**
    Obtain a gradient estimate g_t
    $\eta_t = D / \left(2 \sum_{i=1}^{t} \|g_t\|^2\right)^{1/2}$
    $x^{t+1} = P_{\mathcal{X}}\left(x^t - \eta_t g_t\right)$
end for
Output: $\bar{x}_k = \frac{1}{k} \sum_{t=1}^{k} x^t$

◦ AdaGrad does not need to know smoothness

1. $g_t \in \partial f(x^t)$
2. $g_t = \nabla f(x^t)$
3. $\mathbb{E} g_t = \nabla f(x^t)$ & $\mathbb{E}[\|g - \nabla f(x)\|^2 | x] \leq \sigma^2$

◦ AdaGrad adapts and achieves optimal regret[1]

$$R(k) \leq \sqrt{2D^2 \sum_{t=1}^{k} \|g_t\|_2^2},$$

where $D = \sup_{x,y \in \mathcal{X}} \|x - y\|_2$.

◦ When f is L-smooth, AdaGrad output satisfies[2]

$$\mathbb{E}\left[f(\bar{x}_k)\right] - f^\star = \mathcal{O}\left(\frac{LD^2}{k} + \frac{\sigma D}{\sqrt{k}}\right).$$

[1] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," JMLR, 2011.
[2] K.Y. Levy, A. Yurtsever, and V. Cevher, "Online adaptive methods, universality and acceleration," NeurIPS 2018.

The curious case of AdaGrad[1]

Algorithm: AdaGrad (scalar)[2]

Input: Iterations k; $x_0 \in \mathcal{X}$
for $t = 0, \ldots, k-1$ **do**
  Obtain a gradient estimate g_t
  $\eta_t = D / \left(2 \sum_{i=1}^{t} \|g_t\|^2\right)^{1/2}$
  $x^{t+1} = P_{\mathcal{X}}\left(x^t - \eta_t g_t\right)$
end for
Output: $\bar{x}_k = \frac{1}{k} \sum_{t=1}^{k} x^t$

- Is it an adaptive optimization method?
- Is it a universal optimization method?

- AdaGrad does not need to know smoothness
  1. $g_t \in \partial f(x^t)$
  2. $g_t = \nabla f(x^t)$
  3. $\mathbb{E} g_t = \nabla f(x^t)$ & $\mathbb{E}[\|g - \nabla f(x)\|^2 | x] \leq \sigma^2$
- AdaGrad adapts and achieves optimal regret[1]

$$R(k) \leq \sqrt{2D^2 \sum_{t=1}^{k} \|g_t\|_2^2},$$

where $D = \sup_{x,y \in \mathcal{X}} \|x - y\|_2$.

- When f is L-smooth, AdaGrad output satisfies[2]

$$\mathbb{E}\left[f(\bar{x}_k)\right] - f^\star = \mathcal{O}\left(\frac{LD^2}{k} + \frac{\sigma D}{\sqrt{k}}\right).$$

[1] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," JMLR, 2011.
[2] K.Y. Levy, A. Yurtsever, and V. Cevher, "Online adaptive methods, universality and acceleration," NeurIPS 2018.

Enter AcceleGrad:[1] Exploiting the linear coupling idea[2]

Algorithm: AcceleGrad for unconstrained optimization

Input: Iterations k; $y_0, z_0 \in \mathbb{R}^p$
for $t = 0, \ldots, k-1$ **do**
  Obtain a gradient estimate g_t
  $\alpha_t = \max\left(1, \frac{t+1}{4}\right)$
  $\eta_t = \frac{2D}{\sqrt{G^2 + \sum_{i=0}^{t} \alpha_i^2 \|g_i\|^2}}$
  $x^{t+1} = \frac{1}{\alpha_t} y_t + (1 - \frac{1}{\alpha_t}) z_t,$
  $z_{t+1} = P_{\mathcal{X}}\left(z_t - \alpha_t \eta_t g_t\right)$
  $y_{t+1} = x^{t+1} - \eta_t g_t$
end for
Output: $\bar{y}_k \propto_\alpha \sum_{t=1}^{k} \alpha_{t-1} y_t$

- AcceleGrad does not need to know smoothness
  1. $g_t \in \partial f(x^t)$
  2. $g_t = \nabla f(x^t)$
  3. $\mathbb{E} g_t = \nabla f(x^t)$ & $\|g\| \leq G$
- AcceleGrad output satisfies:[1] $\mathbb{E} f(\bar{y}_k) - f^\star =$
  1. $\mathcal{O}\left(\frac{GD\sqrt{\log(k)}}{\sqrt{k}}\right)$
  2. $\mathcal{O}\left(\frac{DG + LD^2 \log(LD/G)}{k^2}\right)$
  3. $\mathcal{O}\left(\frac{GD\sqrt{\log k}}{\sqrt{k}}\right)$
- Caveats:
  - needs a bound G on the subgradient norms
  - needs a bound D on $\mathcal{X}$ where the solution lives
  - cannot handle constraints!

[1] K.Y. Levy, A. Yurtsever, and V. Cevher, "Online adaptive methods, universality and acceleration," NeurIPS 2018.
[2] L. Orecchia and Z. Allen-Zhu, "Linear coupling: An ultimate unification of gradient and mirror descent," arXiv:1407.1537, 2014.

Enter AcceleGrad:[1] Exploiting the linear coupling idea[2]

Algorithm: AcceleGrad for unconstrained optimization

Input: Iterations k; $y_0, z_0 \in \mathbb{R}^p$
for $t = 0, \ldots, k-1$ **do**
  Obtain a gradient estimate g_t
  $\alpha_t = \max\left(1, \frac{t+1}{4}\right)$
  $\eta_t = \frac{2D}{\sqrt{\cancel{G^2} + \sum_{i=0}^{t} \alpha_i^2 \|g_i\|^2}}$
  $x^{t+1} = \frac{1}{\alpha_t} y_t + (1 - \frac{1}{\alpha_t}) z_t,$
  $z_{t+1} = P_{\mathcal{X}}\left(z_t - \alpha_t \eta_t g_t\right)$
  $y_{t+1} = x^{t+1} - \eta_t g_t$
end for
Output: $\bar{y}_k \propto_\alpha \sum_{t=1}^{k} \alpha_{t-1} y_t$

- AcceleGrad does not need to know smoothness
  1. $g_t \in \partial f(x^t)$
  2. $g_t = \nabla f(x^t)$
  3. $\mathbb{E} g_t = \nabla f(x^t)$ & $\|g\| \leq G$
- AcceleGrad output satisfies:[1] $\mathbb{E} f(\bar{y}_k) - f^\star =$
  1. $\mathcal{O}\left(\frac{GD\sqrt{\log(k)}}{\sqrt{k}}\right)$
  2. $\mathcal{O}\left(\frac{\cancel{DG} + LD^2 \log(LD/\|g_0\|)}{k^2}\right)$
  3. $\mathcal{O}\left(\frac{GD\sqrt{\log k}}{\sqrt{k}}\right)$
- Caveats:
  - needs a bound G on the subgradient norms
  - needs a bound D on $\mathcal{X}$ where the solution lives
  - cannot handle constraints!

[1] K.Y. Levy, A. Yurtsever, and V. Cevher, "Online adaptive methods, universality and acceleration," NeurIPS 2018.
[2] L. Orecchia and Z. Allen-Zhu, "Linear coupling: An ultimate unification of gradient and mirror descent," arXiv:1407.1537, 2014.

Enter AcceleGrad:[1] Exploiting the linear coupling idea[2]

Algorithm: AcceleGrad for unconstrained optimization

1: **Input:** Iterations k; $y_0, z_0 \in \mathbb{R}^p$
2: **for** $t = 0, \ldots, k-1$ **do**
3: Obtain a gradient estimate g_t
4: $\alpha_t = \max\left(1, \frac{t+1}{4}\right)$
5: $\eta_t = \frac{2D}{\sqrt{\cancel{G^2} + \sum_{i=0}^{t} \alpha_i^2 \|g_i\|^2}}$
6: $x^{t+1} = \frac{1}{\alpha_t} y_t + (1 - \frac{1}{\alpha_t}) z_t,$
7: $z_{t+1} = P_{\mathcal{X}}\left(z_t - \alpha_t \eta_t g_t\right)$
8: $y_{t+1} = x^{t+1} - \eta_t g_t$
9: **end for**
10: **Output:** $\bar{y}_k \propto_\alpha \sum_{t=1}^{k} \alpha_{t-1} y_t$

- Is it an adaptive optimization method?
- Is it a universal optimization method?

- AcceleGrad does not need to know smoothness
  1. $g_t \in \partial f(x^t)$
  2. $g_t = \nabla f(x^t)$
  3. $\mathbb{E} g_t = \nabla f(x^t)$ & $\|g\| \leq G$

- AcceleGrad output satisfies:[1] $\mathbb{E} f(\bar{y}_k) - f^\star =$
  1. $\mathcal{O}\left(\frac{GD\sqrt{\log(k)}}{\sqrt{k}}\right)$
  2. $\mathcal{O}\left(\frac{\cancel{DG} + LD^2 \log(LD/\|g_0\|)}{k^2}\right)$
  3. $\mathcal{O}\left(\frac{GD\sqrt{\log k}}{\sqrt{k}}\right)$

- Caveats:
  - needs a bound G on the subgradient norms
  - needs a bound D on $\mathcal{X}$ where the solution lives
  - cannot handle constraints!

[1] K.Y. Levy, A. Yurtsever, and V. Cevher, "Online adaptive methods, universality and acceleration," NeurIPS 2018.
[2] L. Orecchia and Z. Allen-Zhu, "Linear coupling: An ultimate unification of gradient and mirror descent," arXiv:1407.1537, 2014.

UniXGrad:[1] Universal eXtra Gradient method

Algorithm: UniXGrad

Input: Iterations k; $y_0 \in \mathcal{X}$; $\alpha_t = t$

1: **for** $t = 0, \ldots, k-1$ **do**

2: $\tilde{y}_t \propto_\alpha \alpha_t y_{t-1} + \sum_{i=1}^{t-1} \alpha_i x_i$

3: Obtain a gradient estimate $g_t^{(1)} = g_t(\tilde{y}_t)$

4: $$\eta_t = 2D \Big/ \sqrt{1 + \sum_{i=1}^{t-1} \alpha_i^2 \left\| g_i^{(1)} - g_i^{(2)} \right\|_*^2}$$

5: $$x^t = P_{\mathcal{X}}\left(y_{t-1} - \alpha_t \eta_t g_t^{(1)} \right)$$

6: $\bar{x}_t \propto_\alpha \alpha_t x^t + \sum_{i=1}^{t-1} \alpha_i x_i \to$ **output**

7: Obtain a gradient estimate $g_t^{(2)} = g_t(\bar{x}_t)$

8: $$y_t = P_{\mathcal{X}}\left(y_{t-1} - \alpha_t \eta_t g_t^{(2)} \right)$$

9: **end for**

[1] A. Kavis, K.Y. Levy, F. Bach, and V. Cevher, "Unixgrad: A universal, adaptive algorithm with optimal guarantees for constrained optimization," NeurIPS 2019.

[2] A. Nemirovski, " Prox-method with rate of convergence ... smooth convex-concave saddle point problems," SIOPT, 2005.

[3] A. Rakhlin and K. Sridharan, "Optimization, learning, and games with predictable sequences," NeurIPS 2013.

[4] A. Cutkosky, "Anytime online-to-batch, optimism and acceleration," ICML 2019.

UniXGrad:[1] Universal eXtra Gradient method

Algorithm: UniXGrad

Input: Iterations k; $y_0 \in \mathcal{X}$; $\alpha_t = t$

1: **for** $t = 0, \ldots, k-1$ **do**

2: $\tilde{y}_t \propto_\alpha \alpha_t y_{t-1} + \sum_{i=1}^{t-1} \alpha_i x_i$

3: Obtain a gradient estimate $g_t^{(1)} = g_t(\tilde{y}_t)$

4: $\eta_t = 2D \Big/ \sqrt{1 + \sum_{i=1}^{t-1} \alpha_i^2 \left\| g_i^{(1)} - g_i^{(2)} \right\|_*^2}$

5: $x^t = P_{\mathcal{X}} \left(y_{t-1} - \alpha_t \eta_t g_t^{(1)} \right)$

6: $\bar{x}_t \propto_\alpha \alpha_t x^t + \sum_{i=1}^{t-1} \alpha_i x_i \to$ **output**

7: Obtain a gradient estimate $g_t^{(2)} = g_t(\bar{x}_t)$

8: $y_t = P_{\mathcal{X}} \left(y_{t-1} - \alpha_t \eta_t g_t^{(2)} \right)$

9: **end for**

- UniXGrad does not need to know smoothness
  1. $g_t(\cdot) \in \partial f(\cdot)$
  2. $g_t(\cdot) = \nabla f(\cdot)$
  3. $\mathbb{E} g_t(\cdot) = \nabla f(\cdot)$ & $\mathbb{E}[\|g_t(x) - \nabla f(x)\|^2 | x] \leq \sigma^2$
- UniXGrad output satisfies:[1] $\mathbb{E} f(\bar{x}_k) - f^\star =$
  1. $\frac{6D}{k^2} + \frac{14GD}{\sqrt{k}}$
  2. $\frac{20\sqrt{7}D^2 L}{k^2}$
  3. $\frac{224\sqrt{14}D^2 L}{k^2} + \frac{14\sqrt{2}\sigma D}{\sqrt{k}}$
- First universal and adaptive algorithm
  - optimal rates in the "offline" setting
  - builds on mirror-prox[2] & optimistic MD[3]
  - new online-to-offline conversion lemma[14]

[1] A. Kavis, K.Y. Levy, F. Bach, and V. Cevher, "Unixgrad: A universal, adaptive algorithm with optimal guarantees for constrained optimization," NeurIPS 2019.

[2] A. Nemirovski, " Prox-method with rate of convergence ... smooth convex-concave saddle point problems," SIOPT, 2005.

[3] A. Rakhlin and K. Sridharan, "Optimization, learning, and games with predictable sequences," NeurIPS 2013.

[4] A. Cutkosky, "Anytime online-to-batch, optimism and acceleration," ICML 2019.

f is nonconvex

$$f^\star = \min_{x : x \in \mathcal{X}} f(x) \quad (\text{argmin} \rightarrow x^\star)$$

Detour: Weak convexity (WeCo) & approximate stationarity[1]

- Smooth: Gradient mapping norm
  - $\|G_\eta(x^k)\|^2 = \frac{1}{\eta^2}\|x^k - P_{\mathcal{X}}(x^k - \eta\nabla f(x^k))\|^2$
  - possible to compute
- Non-smooth: Generalized subdifferential distance
  - $\mathrm{dist}(0, \partial(f(x^k) + \delta_{\mathcal{X}}(x^k)))^2$
  - hard in general (even approximately)[2][3]

- f is ρ-weakly convex if $f(x) + \frac{\rho}{2}\|x\|^2$ is convex.

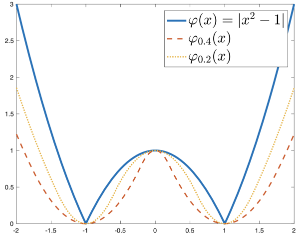


Figure: ME with $f(x) = |x^2 - 1|$, $\mathcal{X} = \mathbb{R}$, and $\hat{v}_t = \mathbb{I}$.[1]

- Moreau envelope (ME):

$$\varphi_{1/\rho}(x) = \min_{y\in\mathcal{X}} \left\{ f(y) + \frac{\rho}{2}\|y - x\|^2 \right\}$$

$$\hat{x} \leftarrow \arg\min$$

$$\nabla\varphi_{1/\rho}(x) = \rho(x - \hat{x})$$

- Small $\|\nabla\phi_{1/\rho}(x)\|$ implies near-stationarity:[1]

$$\mathrm{dist}(0, \partial(f(x^k) + \delta_{\mathcal{X}}(x^k)))^2 \le \|\nabla\phi_{1/\rho}(x^k)\|^2$$

  - also implies small $\|G_\eta(x^k)\|^2$ if f is smooth

[1] D. Davis and D. Drusvyatskiy, "Stochastic model-based minimization of weakly convex functions," SIOPT, 2019.
[3] J. Zhang, et al., "On complexity of finding stationary points of nonsmooth nonconvex functions," arXiv:2002.04130, 2020.
[3] O. Shamir, "Can We Find Near-Approximately-Stationary Points of Nonsmooth Nonconvex Functions?" arXiv:2002.11962, 2020.

The King of all optimization algorithms: Adam[1] (60K+ citations)

Algorithm: (variable metric) Adam

1: **Input:** Iterations k; $x_0 \in \mathcal{X}$, $\beta_{1,2} \in [0,1]$
2: **for** $t = 0, \ldots, k-1$ **do**
3: Obtain a gradient estimate g_t
4: $m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$
5: $\hat{v}_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$
6: $x^{t+1} = P_{\mathcal{X}}^{\hat{v}_t^{1/2}} \left(x^t - \alpha_t \hat{v}_t^{-1/2} m_t \right)$
7: **end for**
8: **Output:** $x^{t_*(k)}$: $t_*(k)$ is randomly chosen in $\{1, \ldots, k\}$.

- The King does not need to know smoothness
  1. $g_t \in \partial f(x^t)$
  2. $g_t = \nabla f(x^t)$
  3. $\mathbb{E} g_t = \nabla f(x^t)$ & $\mathbb{E}[\|g - \nabla f(x)\|^2 | x] \leq \sigma^2$

- The King adapts and achieves optimal regret[3]

$$R(k) = \mathcal{O}\left(\sqrt{k}\right),$$

with constant β_1 in OCO.

- The King's output satisfies for WeCo[4]

$$\mathbb{E}\|\nabla \phi_{1/\rho}^t (x^{t_*(k)})\|^2 = \mathcal{O}\left(\frac{1}{\sqrt{k}}\right).$$

[1] D.P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980, 2014.

[2] S.J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," arXiv:1904.09237, 2019.

[3] A. Alacaoglu, Y. Malitsky, P. Mertikopoulos, and V. Cevher, "A new regret analysis for adam-type algorithms," ICML 2020

[4] A. Alacaoglu, Y. Malitsky, and V. Cevher, "Convergence of adaptive algorithms for weakly convex constrained optimization," arXiv:2006.06650, 2020.

The King of all optimization algorithms: Adam[1] (60K+ citations)

Algorithm: (variable metric) Adam-type

Input: Iterations k; $x_0 \in \mathcal{X}$, $\beta_{1,2} \in [0,1]$
for $t = 0, \ldots, k-1$ **do**
  Obtain a gradient estimate g_t
  $m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$
  $\hat{v}_t = \phi(g_t)$
  $x^{t+1} = P_{\mathcal{X}}^{\hat{v}_t^{1/2}} \left(x^t - \alpha_t \hat{v}_t^{-1/2} m_t \right)$
end for
Output: $x^{t_*(k)}$: $t_*(k)$ is randomly chosen in $\{1, \ldots, k\}$.

- The King is naked:[2] AMSGrad
  - $\phi(g_t) = \max(\hat{v}_{t-1}, v_t)$, and $v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$
  - F. Orabona: parameterfree.com (Dec 6)

- ~~The King~~ An Adam-type algorithm does not need to know smoothness
  1. $g_t \in \partial f(x^t)$
  2. $g_t = \nabla f(x^t)$
  3. $\mathbb{E} g_t = \nabla f(x^t)$ & $\mathbb{E}[\|g - \nabla f(x)\|^2 | x] \leq \sigma^2$

- ~~The King~~ An Adam-type algorithm adapts and achieves optimal regret[3]

$$R(k) = \mathcal{O}\left(\sqrt{k}\right),$$

with constant β_1 in OCO.

- ~~The King~~ An Adam-type algorithms' 's output satisfies for WeCo[4]

$$\mathbb{E}\|\nabla \phi_{1/\rho}^t(x^{t_*(k)})\|^2 = \mathcal{O}\left(\frac{1}{\sqrt{k}}\right).$$

---

[1] D.P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980, 2014.
[2] S.J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," arXiv:1904.09237, 2019.
[3] A. Alacaoglu, Y. Malitsky, P. Mertikopoulos, and V. Cevher, "A new regret analysis for adam-type algorithms," ICML 2020
[4] A. Alacaoglu, Y. Malitsky, and V. Cevher, "Convergence of adaptive algorithms for weakly convex constrained optimization," arXiv:2006.06650, 2020.

A comparison of algorithms

	GD/SGD	Accelerated GD/SGD	AdaGrad	AcceleGrad/UniXgrad	Adam/AMSGrad
Convex, stochastic	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$ [1]	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$ [1]	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$ [2]	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$ [3,4]	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$ [5]
Convex, deterministic, L-smooth	$\mathcal{O}\left(\frac{1}{k}\right)$ [1]	$\mathcal{O}\left(\frac{1}{k^2}\right)$ [1]	$\mathcal{O}\left(\frac{1}{k}\right)$ [3]	$\mathcal{O}\left(\frac{1}{k^2}\right)$ [3,4]	$\mathcal{O}\left(\frac{1}{k}\right)$ [6]
Nonconvex, stochastic, L-smooth	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$ [1]	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$ [1]	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$ [7]	?	$\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$ [8]
Nonconvex, deterministic, L-smooth	$\mathcal{O}\left(\frac{1}{k}\right)$ [1]	$\mathcal{O}\left(\frac{1}{k}\right)$ [1]	$\mathcal{O}\left(\frac{1}{k}\right)$ [7]	?	$\mathcal{O}\left(\frac{1}{k}\right)$ [6]

1 Lan, First-order and Stochastic Optimization Methods for Machine Learning. Springer Nature, 2020.

2 Duchi, Hazan, Singer, Adaptive subgradient methods for online learning and stochastic optimization, JMLR, 2011

3 Levy, Yurtsever, Cevher, Online adaptive methods, universality and acceleration, NeurIPS 2018

4 Kavis, Levy, Bach, Cevher, UniXGrad: A Universal, Adaptive Algorithm with Optimal Guarantees for Constrained Optimization, NeurIPS, 2019

5 Reddi, Kale, Kumar, On the convergence of adam and beyond, ICLR, 2018.
Alacaoglu, Malitsky, Mertikopoulos, Cevher, A new regret analysis for Adam-type algorithms, ICML 2020.

6 Barakat, Bianchi, Convergence Rates of a Momentum Algorithm with Bounded Adaptive Step Size for Nonconvex Optimization, ACML, 2020

7 Ward, Xu, Bottou, AdaGrad stepsizes: Sharp convergence over nonconvex landscapes, ICML 2019.

8 Alacaoglu, Malitsky, Cevher, Convergence of adaptive algorithms for weakly convex constrained optimization, arXiv, 2020.
Chen, Zhou, Tang, Yang, Cao, Gu, Closing the generalization gap of adaptive gradient methods in training deep neural networks, IJCAI 2020.
Chen, Liu, Sun, Hong, On the convergence of a class of adam-type algorithms for non-convex optimization, ICLR 2018.

Conclusions

- Simple algorithms automatically adapt to strong convexity under broad assumptions
  - GD achieves linear rate with $\eta = 1/L$[1]
  - SGD achieves $\mathcal{O}(1/k)$-rate with $\eta_k = \mathcal{O}(1/k)$[2]
  - PDHG achieves linear rate under metric subregularity[3,4,5]

- Adaptive methods are promising but are not yet truly universal...
  - Accelegrad/UniXgrad does not adapt to strong convexity
  - AdaGrad needs a different step-size policy
  - Adam-type does not adapt to strong convexity
  - MetaGrad comes close but is not universal yet[6]

- Still seeking one algorithm to rule them all!

---

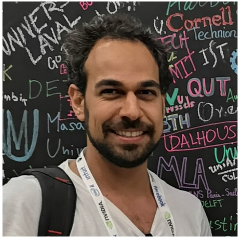
[1] G. Lan, "First-order and Stochastic Optimization Methods for Machine Learning," Springer Nature, 2020.
[2] P. Mertikopoulos, N. Hallak, A. Kavis, and V. Cevher, "On the almost sure convergence of stochastic gradient descent in non-convex problems," NeurIPS, 2020.
[3] P. Latafat, N.M. Freris, and P. Patrinos, "A new randomized block-coordinate primal-dual proximal algorithm for distributed optimization," IEEE TAC, 2019.
[4] A. Alacaoglu, O. Fercoq, and V. Cevher, "Random extrapolation for primal-dual coordinate descent," ICML, 2020.
[5] J. Liang, J. Fadili, and G. Peyré, "Convergence rates with inexact non-expansive operators." MathProg, 2016.
[6] T. van Erven, and W.M. Koolen, "Metagrad: Multiple learning rates in online learning." NeurIPS 2016.

Acknowledgements

- Faculty: Kfir Levy, Francis Bach, Yura Malitsky, Panayotis Mertikopoulos.
- PhD:

Ahmet Alacaoglu
ahmet.alacaoglu@epfl.ch

Ali Kavis
ali.kavis@epfl.ch



Alp Yurtsever
alpy@mit.edu

- Postdoc positions available at LIONS. Email: volkan.cevher@epfl.ch

Logistic regression

- Data: a4a
- Oracle: Deterministic

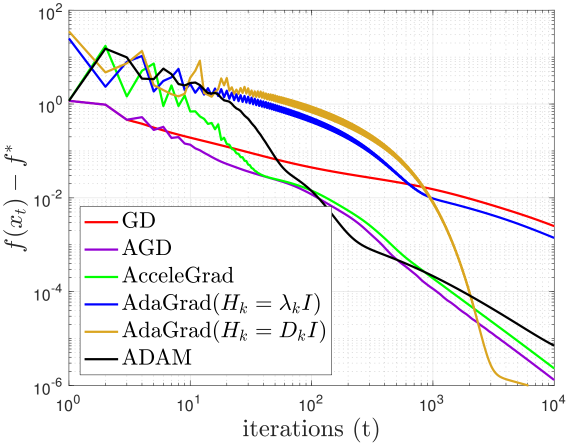


Figure: Logistic regression on a4a

Neural network training: ADAM vs. AcceleGrad

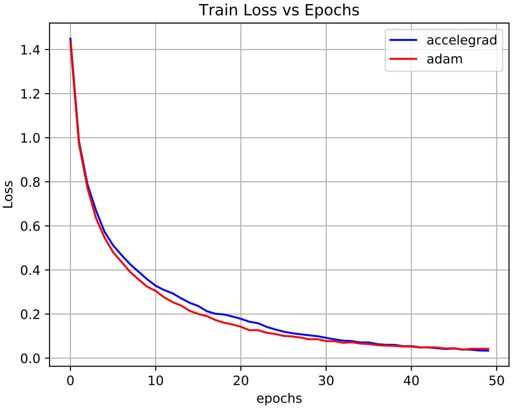


Figure: Resnet classifier optimization (train loss)

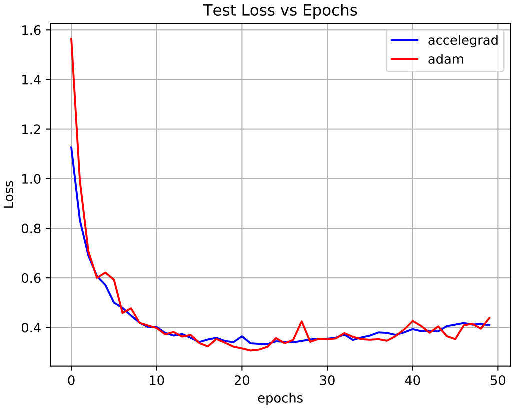


Figure: Resnet classifier optimization (test loss)