Differentiable Programming à la Moreau

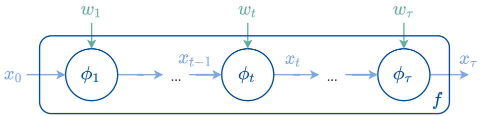
Vincent Roulet, Zaid Harchaoui

International Conference on Acoustics, Speech and Signal Processing
05/09/2022

UNIVERSITY of WASHINGTON

NSF

Optimization Oracles



Oracles for an objective f

Gradient oracle

Rationale: Uses linear approx. of f around param. w
$\rightarrow$ oracle accuracy *fixed* by smoothness properties of f ✗
Implementation: Uses decomposition of f into elementary operations
$\rightarrow$ flexible and fast implementation by automatic differentiation ✓

Moreau envelope based oracle (Moreau 1962, Nesterov 2005, Lin et al. 2018)

Rationale: Uses regularized minimization of f around param. w
$\rightarrow$ oracle accuracy *controlled* by optimization subroutine ✓
Implementation: Requires a priori solving an optimization subproblem
$\rightarrow$ does not exploit decomposition of f into elementary operations ✗

Can we develop approximate computations of Moreau envelopes that exploit the decomposition of the objective?

Moreau Gradients

Moreau Gradient of f on w with stepsize α

$$\nabla \operatorname{env}(\alpha f)(w) = \operatorname*{arg\,min}_{v \in \mathbb{R}^d} \alpha f(w - v) + \|v\|_2^2/2$$

- Well-defined for $0 \leq \alpha < \bar{\alpha}$ s.t. $v \mapsto \bar{\alpha} f(w - v) + \|v\|_2^2/2$ is convex
- Maximal stepsize $\bar{\alpha}$ larger than gradient descent stepsize
- Necessary optimal cond.: $w^* \in \operatorname{arg\,min}_w f(w) \Rightarrow \nabla \operatorname{env}(\alpha f)(w^*) = 0$
- Generally not available in closed form

Approximate Moreau Gradient Optimization

$$w^{(k+1)} = w^{(k)} - \widehat{\nabla} \operatorname{env}(\alpha f)(w^{(k)})$$

for $\widehat{\nabla} \operatorname{env}(\alpha f)(w) \approx \nabla \operatorname{env}(\alpha f)(w)$

- Direct implementation: $\widehat{\nabla} \operatorname{env}(\alpha f)(w) = \mathcal{A}_k \left(\alpha f(w - \cdot) + \|\cdot\|_2^2/2\right)$ for $\mathcal{A}_k(g)$ the k^{th} iterate of algo. $\mathcal{A}$ on g such as gradient descent
- Here: Implement f in a differentiable programming framework that gives access to Moreau gradients in a backward pass like

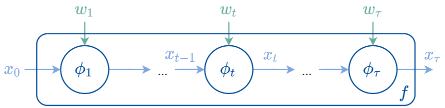
```
out = func(w)     m_grad = auto_m_grad(out, w, alpha)
```

with m_grad$=\nabla \operatorname{env}(\alpha f)(w)$ computed from graph of comput. of f.

Differentiable Programming for Moreau Gradients

Compute Moreau gradient of $h \circ f$ for f with dynamical structure

$$f(w_1, \ldots, w_\tau) = x_\tau,$$
$$\text{s.t. } x_t = \phi_t(w_t, x_{t-1})$$

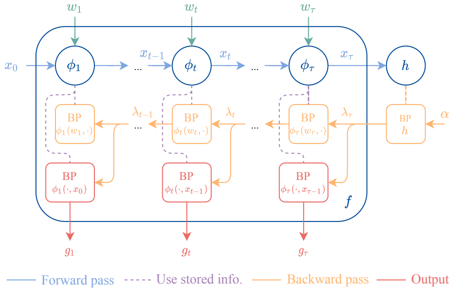


Forward pass

- Compute f through func. ϕ_t
- Store comput. ϕ_t and inputs x_t, w_t

Backward pass

- Back-prop. λ_t using rule BP below on $\phi_t(w_t, \cdot)$ or $\phi_t(\cdot, x_t)$ starting from $\lambda_\tau = \text{BP}(h)(x_\tau, \alpha)$



$\text{BP} \in \{\text{GBP}, \text{MBP}, \text{IBP}\}$

$\diamond$ $\text{GBP}(\phi)(z, \lambda) = \nabla\phi(z)\lambda$
$\rightarrow$ classical backprop. rule in auto.-diff.

$\diamond$ $\text{MBP}(\phi)(z, \lambda) \approx \arg\min_y \lambda^\top \phi(z-y) + \|y\|_2^2/2 = \nabla \operatorname{env}(\lambda^\top \phi)(z)$
$\rightarrow$ generalized Moreau gradient

$\diamond$ $\text{IBP}(\phi)(z, \lambda) \approx \arg\min_y \|\phi(z-y) - \phi(z) + \lambda\|_2^2 + \|y\|_2^2/2$
$\rightarrow$ regularized inverse as in target prop. Lee et al (2015)

Chain Rule

Moreau Gradient Rule for composition $h \circ f$
Under suitable assumptions, comput. of Moreau gradient decomposes as

$$\nabla \operatorname{env}(\alpha h \circ f)(w) = \arg\min_{y} \left\{ {\lambda^*}^\top f(w - v) + \|v\|_2^2/2 \right\}$$
$$\text{where} \quad \lambda^* = \arg\max_{\lambda} -(\alpha h)^*(\lambda) + \operatorname{env}(\lambda^\top f)(w)$$

- Proximal grad. step to compute λ^* gives MBP rule:
  $\to \nabla \operatorname{env}(\alpha h \circ f)(w) \approx \nabla \operatorname{env}(\lambda^\top f)(w)$ for $\lambda = \nabla \operatorname{env}(\alpha h)(f(w))$

Regularized Inverse Rule for composition $h \circ f$
Comput. of Moreau gradient amounts to solve

$$\min_{\lambda} \ \alpha f(g(w) - \lambda) + p(\lambda) \text{ for } p(\lambda) = \min \left\{ \|v\|_2^2/2 : g(w) - g(w-v) = \lambda \right\}$$

- Incremental proximal point to compute λ^* gives IBP rule:
  $\to \nabla \operatorname{env}(\alpha h \circ f)(w) \approx \mathrm{IBP}(f)(w; \lambda)$ for $\lambda = \nabla \operatorname{env}(\alpha h)(f(w))$

Implementation
Use k iterations of algo. $\mathcal{A}$ such as grad.descent to approx. BP rule such as

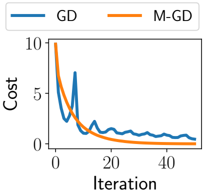
$$\mathrm{MBP}(f)(w, \lambda) \approx \mathcal{A}_k(\lambda^\top f(w - \cdot) + \| \cdot \|_2^2/2)$$

$\mathcal{A} = GD$, $k = 1 \to \mathrm{MBP}(f)(w, \lambda) \approx \nabla f(w) \lambda$

Experiments

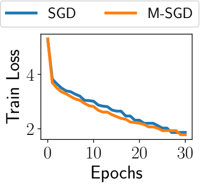
Moreau Gradient Descent (M-GD)

- Nonlinear control: swinging up pendulum
- Use approx. Moreau grad. on output of deterministic dynamical system



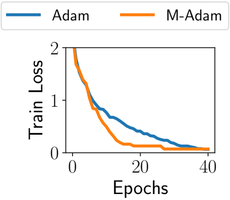
Stoch. Moreau Grad. Desc. (M-SGD)

- MLP on CIFAR10
- Compute oracles on mini-batches S, i.e., $\widehat{\nabla}\,\mathrm{env}(\alpha F_S)(w)$ for $F_S(w) = \sum_{i \in S} f_i(w)$



Adam with Moreau Grad. (M-Adam)

- AllCNN ConvNet on CIFAR10
- Compute oracles on mini-batches S, i.e., $\widehat{\nabla}\,\mathrm{env}(\alpha F_S)(w)$ for $F_S(w) = \sum_{i \in S} f_i(w)$
- Plug oracle directions in Adam optimizer



Lin, H., Mairal, J. & Harchaoui, Z. (2018), 'Catalyst acceleration for first-order convex optimization: from theory to practice', *Journal of Machine Learning Research* **18**(212), 1–54.

Moreau, J. J. (1962), 'Fonctions convexes duales et points proximaux dans un espace hilbertien', *Comptes Rendus de l'Académie des Sciences* **255**.

Nesterov, Y. (2005), 'Smooth minimization of non-smooth functions', *Mathematical programming* **103**(1), 127–152.