What does backpropagation compute?

Edouard Pauwels (IRIT, Toulouse 3)
joint work with Jérôme Bolte (TSE, Toulouse 1)

Optimization for machine learning
CIRM

March 2020

Plan

Motivation: There is something that we do not understand in backpropagation for deep learning.

Plan

Motivation: There is something that we do not understand in backpropagation for deep learning.

Nonsmooth analysis is not really compatible with calculus.

Plan

Motivation: There is something that we do not understand in backpropagation for deep learning.

Nonsmooth analysis is not really compatible with calculus.

Contribution: Conservative set valued fields. Analytic, geometric and algorithmic properties.

Automatic differentiation (AD, 70s):

Automatic differentiation (AD, 70s):
Automatized numerical implementation of the chain rule:

$$H\colon \mathbb{R}^p \mapsto \mathbb{R}^p, \quad G: \mathbb{R}^p \mapsto \mathbb{R}^p, \quad f: \mathbb{R}^p \to \mathbb{R}, \qquad \text{(differentiable)}.$$

$$f \circ G \circ H\colon \mathbb{R}^p \mapsto \mathbb{R}.$$

$$\nabla(f \circ G \circ H)^T = \nabla f^T \times J_G \times J_H$$

Automatic differentiation (AD, 70s):
Automatized numerical implementation of the chain rule:

$$H\colon \mathbb{R}^p \mapsto \mathbb{R}^p, \quad G\colon \mathbb{R}^p \mapsto \mathbb{R}^p, \quad f\colon \mathbb{R}^p \to \mathbb{R}, \qquad \text{(differentiable)}.$$

$$f \circ G \circ H\colon \mathbb{R}^p \mapsto \mathbb{R}.$$

$$\nabla(f \circ G \circ H)^T = \nabla f^T \times J_G \times J_H$$

Function = program: smooth elementary operations, combined smoothly.

$$x \mapsto (H(x), G(H(x)), f(G(H(x))))$$

Automatic differentiation (AD, 70s):
Automatized numerical implementation of the chain rule:

$$H\colon \mathbb{R}^p \mapsto \mathbb{R}^p, \quad G: \mathbb{R}^p \mapsto \mathbb{R}^p, \quad f: \mathbb{R}^p \to \mathbb{R}, \qquad \text{(differentiable).}$$

$$f \circ G \circ H\colon \mathbb{R}^p \mapsto \mathbb{R}.$$

$$\nabla(f \circ G \circ H)^T = \nabla f^T \times J_G \times J_H$$

Function = program: smooth elementary operations, combined smoothly.

$$x \mapsto (H(x), G(H(x)), f(G(H(x))))$$

Forward mode of AD: $\nabla f^T \times (J_G \times J_H)$.
Backward mode of AD: $(\nabla f^T \times J_G) \times J_H$.

Automatic differentiation (AD, 70s):
Automatized numerical implementation of the chain rule:

$$H\colon \mathbb{R}^p \mapsto \mathbb{R}^p, \quad G : \mathbb{R}^p \mapsto \mathbb{R}^p, \quad f : \mathbb{R}^p \to \mathbb{R}, \qquad \text{(differentiable)}.$$
$$f \circ G \circ H\colon \mathbb{R}^p \mapsto \mathbb{R}.$$
$$\nabla(f \circ G \circ H)^T = \nabla f^T \times J_G \times J_H$$

Function = program: smooth elementary operations, combined smoothly.

$$x \mapsto (H(x), G(H(x)), f(G(H(x))))$$

Forward mode of AD: $\nabla f^T \times (J_G \times J_H)$.
Backward mode of AD: $(\nabla f^T \times J_G) \times J_H$.

Backpropagation: Backward AD for neural network training.

It computes gradient (provided that everybody is smooth).

Neural network / compositional modeling

$$\begin{array}{c}\text{Input}\\ x \in \mathbb{R}^p\end{array} \longrightarrow \boxed{z_0 \in \mathbb{R}^p} \longrightarrow \boxed{z_1 \in \mathbb{R}^{p_1}} \longrightarrow \cdots \longrightarrow \boxed{z_L \in \mathbb{R}^{p_L}}$$

Input
$x \in \mathbb{R}^p$

$$\longrightarrow z_0 \in \mathbb{R}^p \longrightarrow z_1 \in \mathbb{R}^{p_1} \longrightarrow \cdots \longrightarrow z_L \in \mathbb{R}^{p_L}$$

For $i = 1, \ldots, L$:

- $z_i \in \mathbb{R}^{p_i}$ "layer".
- $z_i = \phi_i(W_i z_{i-1} + b_i)$
- $\phi_i : \mathbb{R}^{p_i} \mapsto \mathbb{R}^{p_i}$ "activation functions", nonlinear.
- $W_i \in \mathbb{R}^{p_i \times p_{i-1}}$, $b_i \in \mathbb{R}^{p_i}$, $\theta = (W_1, b_1, \ldots, W_L, b_L)$, model parameters.

Input
$x \in \mathbb{R}^p$

$$\longrightarrow z_0 \in \mathbb{R}^p \longrightarrow z_1 \in \mathbb{R}^{p_1} \longrightarrow \cdots \longrightarrow z_L \in \mathbb{R}^{p_L}$$

For $i = 1, \dots, L$:

- $z_i \in \mathbb{R}^{p_i}$ "layer".
- $z_i = \phi_i(W_i z_{i-1} + b_i)$
- $\phi_i : \mathbb{R}^{p_i} \mapsto \mathbb{R}^{p_i}$ "activation functions", nonlinear.
- $W_i \in \mathbb{R}^{p_i \times p_{i-1}}$, $b_i \in \mathbb{R}^{p_i}$, $\theta = (W_1, b_1, \dots, W_L, b_L)$, model parameters.

$$\begin{aligned} F_\theta(x) &= z_L \\ &= \phi_L\left(W_L \quad \phi_{L-1}\left(W_{L-1}\left(\dots \quad \phi_1\left(W_1 x + b_1\right) \quad \right) \quad + b_{L-1}\right) + b_L\right) \end{aligned}$$

Neural network / compositional modeling

$$\begin{array}{c}\text{Input}\\ x \in \mathbb{R}^p\end{array} \longrightarrow \boxed{z_0 \in \mathbb{R}^p} \longrightarrow \boxed{z_1 \in \mathbb{R}^{p_1}} \longrightarrow \cdots \longrightarrow \boxed{z_L \in \mathbb{R}^{p_L}}$$

For $i = 1, \ldots, L$:

- $z_i \in \mathbb{R}^{p_i}$ "layer".
- $z_i = \phi_i(W_i z_{i-1} + b_i)$
- $\phi_i \colon \mathbb{R}^{p_i} \mapsto \mathbb{R}^{p_i}$ "activation functions", nonlinear.
- $W_i \in \mathbb{R}^{p_i \times p_{i-1}}$, $b_i \in \mathbb{R}^{p_i}$, $\theta = (W_1, b_1, \ldots, W_L, b_L)$, model parameters.

$$\begin{aligned} F_\theta(x) = \; & z_L \\ = \; & \phi_L\left(W_L \quad \phi_{L-1}\left(W_{L-1}\left(\ldots \quad \phi_1\left(W_1 x + b_1\right) \quad \right) \quad + b_{L-1}\right) + b_L\right) \end{aligned}$$

Training set: $\{(x_i, y_i)\}_{i=1}^n$ in $\mathbb{R}^p \times \mathbb{R}^{p_L}$, loss $\ell \colon \mathbb{R}^{p_L} \times \mathbb{R}^{p_L} \to \mathbb{R}_+$.

$$\min_\theta \quad J(\theta) \quad := \quad \frac{1}{n} \sum_{i=1}^n \ell(F_\theta(x_i), y_i) \quad = \quad \frac{1}{n} \sum_{i=1}^n J_i(\theta).$$

Stochastic (minibatch) gradient algorithm: Given $(I_k)_{k\in\mathbb{N}}$ iid, uniform on $\{1,\ldots,n\}$, $(\alpha_k)_{k\in\mathbb{N}}$ positive, iterate,

$$\theta_{k+1} = \theta_k - \alpha_k \nabla J_{I_k}(\theta_k).$$

Backpropagation: Backward mode of automatic differentiation used to compute ∇J_i

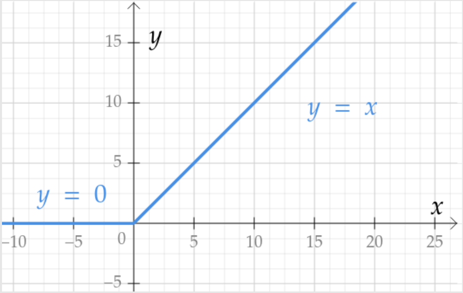
Stochastic (minibatch) gradient algorithm: Given $(I_k)_{k\in\mathbb{N}}$ iid, uniform on $\{1,\dots,n\}$, $(\alpha_k)_{k\in\mathbb{N}}$ positive, iterate,

$$\theta_{k+1} = \theta_k - \alpha_k \nabla J_{I_k}(\theta_k).$$

Backpropagation: Backward mode of automatic differentiation used to compute ∇J_i

Profusion of numerical tools: *e.g.* Tensorflow, Pytorch. Democratized the usage of these models. Goes beyond neural nets (differentiable programming).

Positive part: $\mathrm{relu}(t) = \max\{0, t\}$,



Less straightforward examples:

- Max pooling in convolutional networks.
- knn grouping layers, farthest point subsampling layers.
  Qi *et. al.* 2017. PointNet++: Deep Hierarchical Feature Learning on point Sets in a Metric Space.
- Sorting layers.
  Anil *et. al.* 2019. Sorting Out Lipschitz Function Approximation. ICML.

Nonsmooth backpropagation

Set $\mathrm{relu}'(0) = 0$ and implement the chain rule of smooth calculus.

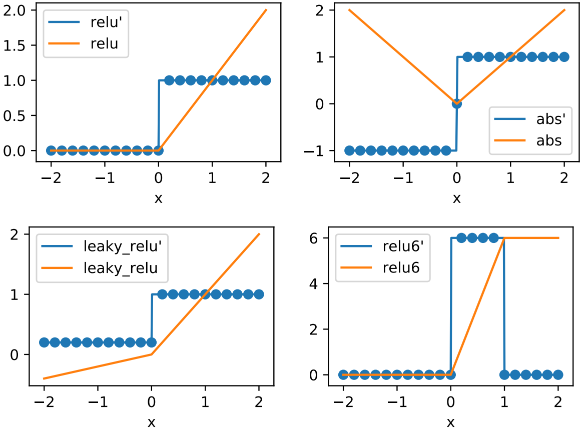
$$(f \circ g)' = g' \times f' \circ g.$$

Nonsmooth backpropagation

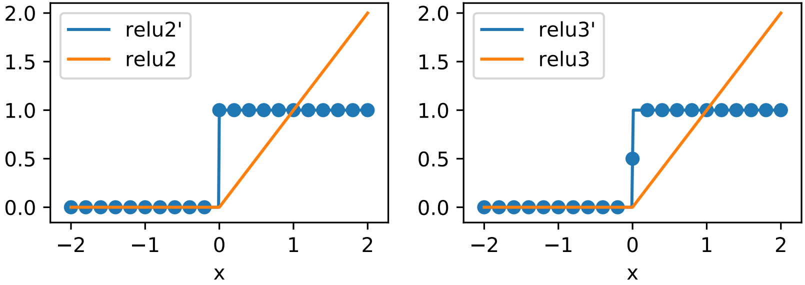
Set $\mathrm{relu}'(0) = 0$ and implement the chain rule of smooth calculus.

$$(f \circ g)' = g' \times f' \circ g.$$

Tensorflow examples:



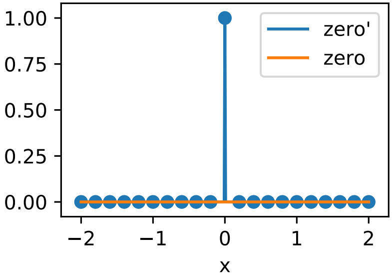
$$\mathrm{relu2}(t) = \mathrm{relu}(-t) + t = \mathrm{relu}(t)$$
$$\mathrm{relu3}(t) = \frac{1}{2}(\mathrm{relu}(t) + \mathrm{relu2}(t)) = \mathrm{relu}(t).$$



Known from AD litterature (*e.g.* Griewank 08, Kakade & Lee 2018).

Derivative of zero at 0

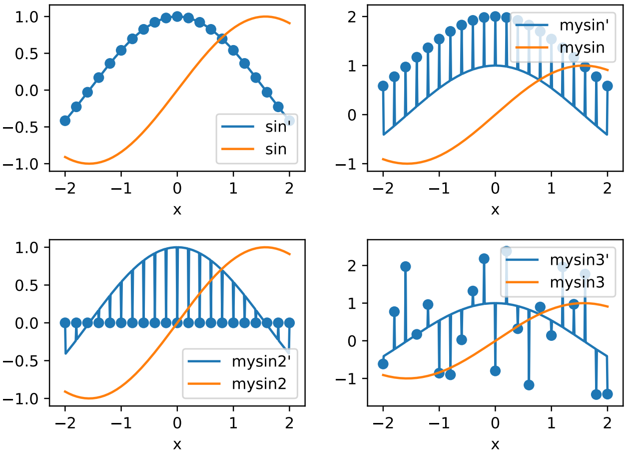
$$\mathrm{zero}(t) = \mathrm{relu2}(t) - \mathrm{relu}(t) = 0.$$



AD acts on programs, not on functions

Derivative of sine at 0:

$$sin' = cos.$$



Consequences for optimization and learning

No convexity, no calculus:

$$\partial(f+g) \subset \partial f + \partial g.$$

Consequences for optimization and learning

No convexity, no calculus:

$$\partial(f+g) \subset \partial f + \partial g.$$

Minibatch + subgradient: locally Lipschitz, convex,

$$\begin{aligned} J(\theta) &= \frac{1}{n}\sum_{i=1}^{n} J_i(\theta) \\ v_i \quad &\in \quad \partial J_i(\theta), \quad i = 1, \ldots n, \\ \mathbb{E}_I[v_I] \quad &\in \quad \partial J(\theta), \qquad\qquad I \quad \text{uniform on } \{1, \ldots, n\}, \end{aligned}$$

No convexity, no calculus:

$$\partial(f+g) \subset \partial f + \partial g.$$

Minibatch + subgradient: locally Lipschitz, no sum rule,

$$\begin{aligned} J(\theta) &= \frac{1}{n}\sum_{i=1}^{n} J_i(\theta) \\ v_i \quad &\in \quad \partial J_i(\theta), \quad i = 1, \ldots n, \\ \mathbb{E}_I[v_I] \quad &\notin \quad \partial J(\theta), \qquad\qquad I \quad \text{uniform on } \{1, \ldots, n\}, \end{aligned}$$

Consequences for optimization and learning

No convexity, no calculus:

$$\partial(f+g) \subset \partial f + \partial g.$$

Minibatch + subgradient: locally Lipschitz, no sum rule, auto differentiation.

$$
\begin{array}{lll}
J(\theta) = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} J_i(\theta) & & \\
v_i \quad \notin \quad \partial J_i(\theta), \quad i = 1, \dots n, & & \\
\mathbb{E}_I[v_I] \quad \notin \quad \partial J(\theta), & I & \text{uniform on } \{1, \dots, n\},
\end{array}
$$

Consequences for optimization and learning

No convexity, no calculus:

$$\partial(f+g) \subset \partial f + \partial g.$$

Minibatch + subgradient: locally Lipschitz, no sum rule, auto differentiation.

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n} J_i(\theta)$$

$$v_i \quad \notin \quad \partial J_i(\theta), \quad i = 1, \ldots n,$$

$$\mathbb{E}_I[v_I] \quad \notin \quad \partial J(\theta), \qquad\qquad I \quad \text{uniform on } \{1, \ldots, n\},$$

Discrepancy:

Analyse: $\theta_{k+1} = \theta_k - \alpha_k(v_k + \epsilon_k), \quad v_k \in \partial J(\theta_k),$

$(\epsilon_i)_{i\in\mathbb{N}}$ zero mean (martingale increments).

(Davis *et. al.* 2018. Stochastic subgradient method converges on tame functions. FOCM.)

Implement: $\theta_{k+1} = \theta_k - \alpha_k D_{I_k}(\theta_k)$

Smooth:

$$\begin{tikzcd} J \arrow[r, "\mathrm{num}"] \arrow[d, "\mathrm{diff}"'] & \mathcal{P} \arrow[d, "\mathrm{autodiff}"] \\ \nabla J \arrow[r, "\mathrm{num}"'] & D \end{tikzcd}$$

Nonsmooth:

$$\begin{tikzcd} J \arrow[r, "\mathrm{num}"] \arrow[d, "\mathrm{diff}"'] & \mathcal{P} \arrow[d, "\mathrm{autodiff}"] \\ \partial J & D \end{tikzcd}$$

A mathematical model for "nonsmooth automatic differentiation"?

Outline

What is a derivative?

What is a derivative?

Linear operator:

$$\begin{aligned} \text{derivative}\colon C^1(\mathbb{R}) &\mapsto C^0(\mathbb{R}) \\ f &\mapsto f' \end{aligned}$$

What is a derivative?

Linear operator:

$$\begin{aligned} \text{derivative}\colon C^1(\mathbb{R}) &\quad\mapsto\quad C^0(\mathbb{R}) \\ f &\quad\mapsto\quad f' \end{aligned}$$

Notions of subgradients inherited from calculus of variation follow the "operator" view.

What is a derivative?

Linear operator:

$$\begin{aligned} \text{derivative}: C^1(\mathbb{R}) &\mapsto C^0(\mathbb{R}) \\ f &\mapsto f' \end{aligned}$$

Notions of subgradients inherited from calculus of variation follow the "operator" view.

Lebesgue differentiation theorem: If $f : \mathbb{R} \mapsto \mathbb{R}$ is integrable, then

$$F : x \mapsto \int_{-\infty}^{x} f(t)dt$$

is differentiable for almost all x, with $F'(x) = f(x)$ (F is absolutely continuous).

What is a derivative?

Linear operator:

$$\begin{aligned} \mathrm{derivative}\colon C^1(\mathbb{R}) &\mapsto C^0(\mathbb{R}) \\ f &\mapsto f' \end{aligned}$$

Notions of subgradients inherited from calculus of variation follow the "operator" view.

Lebesgue differentiation theorem: If $f\colon \mathbb{R} \mapsto \mathbb{R}$ is integrable, then

$$F\colon x \mapsto \int_{-\infty}^{x} f(t)dt$$

is differentiable for almost all x, with $F'(x) = f(x)$ (F is absolutely continuous).

Linear map *versus* relation / equivalence class in L^1.

Technical reminder

Absolutely continuous path (AC): $\gamma\colon [0,1] \mapsto \mathbb{R}^p$ is called absolutely continuous if

- γ is differentiable almost everywhere with integrable derivative $\gamma'\colon [0,1] \mapsto \mathbb{R}^p$.
- $\gamma(t) - \gamma(0) = \int_0^t \gamma'(s)ds$, for all $t \in [0,1]$.

Technical reminder

Absolutely continuous path (AC): $\gamma\colon [0,1] \mapsto \mathbb{R}^p$ is called absolutely continuous if

- γ is differentiable almost everywhere with integrable derivative $\gamma'\colon [0,1] \mapsto \mathbb{R}^p$.
- $\gamma(t) - \gamma(0) = \int_0^t \gamma'(s)ds$, for all $t \in [0,1]$.

Set valued field: $D\colon \mathbb{R}^p \rightrightarrows \mathbb{R}^q$ is a function from $\mathbb{R}^p$ to the set of subsets of $\mathbb{R}^q$.

Technical reminder

Absolutely continuous path (AC): $\gamma \colon [0,1] \mapsto \mathbb{R}^p$ is called absolutely continuous if

- γ is differentiable almost everywhere with integrable derivative $\gamma' \colon [0,1] \mapsto \mathbb{R}^p$.
- $\gamma(t) - \gamma(0) = \int_0^t \gamma'(s) ds$, for all $t \in [0,1]$.

Set valued field: $D \colon \mathbb{R}^p \rightrightarrows \mathbb{R}^q$ is a function from $\mathbb{R}^p$ to the set of subsets of $\mathbb{R}^q$.

- ∂f, the subgradient of a convex function f.
- $\partial^c f$, the Clarke subgradient of a locally Lipschitz function f

$$\partial^c f(x) = \mathrm{conv}\left\{ v \in \mathbb{R}^p, \exists y_k \underset{k \to \infty}{\to} x \text{ with } y_k \in R,\, v_k = \nabla f(y_k) \underset{k \to \infty}{\to} v \right\}.$$

where R is the (full measure set) where f is differentiable.

Absolutely continuous path (AC): $\gamma \colon [0,1] \mapsto \mathbb{R}^p$ is called absolutely continuous if

- γ is differentiable almost everywhere with integrable derivative $\gamma' \colon [0,1] \mapsto \mathbb{R}^p$.
- $\gamma(t) - \gamma(0) = \int_0^t \gamma'(s) ds$, for all $t \in [0,1]$.

Set valued field: $D \colon \mathbb{R}^p \rightrightarrows \mathbb{R}^q$ is a function from $\mathbb{R}^p$ to the set of subsets of $\mathbb{R}^q$.

- ∂f, the subgradient of a convex function f.
- $\partial^c f$, the Clarke subgradient of a locally Lipschitz function f

$$\partial^c f(x) = \mathrm{conv} \left\{ v \in \mathbb{R}^p,\, \exists y_k \underset{k \to \infty}{\to} x \text{ with } y_k \in R,\, v_k = \nabla f(y_k) \underset{k \to \infty}{\to} v \right\}.$$

where R is the (full measure set) where f is differentiable.

Closed graph: a notion of continuity for D

$$\mathrm{graph}\, D = \{(x,z),\, x \in \mathbb{R}^p,\, z \in D(x)\} \subset \mathbb{R}^{p+q},$$

Technical reminder

Absolutely continuous path (AC): $\gamma\colon [0,1] \mapsto \mathbb{R}^p$ is called absolutely continuous if

- γ is differentiable almost everywhere with integrable derivative $\gamma'\colon [0,1] \mapsto \mathbb{R}^p$.
- $\gamma(t) - \gamma(0) = \int_0^t \gamma'(s)ds$, for all $t \in [0,1]$.

Set valued field: $D\colon \mathbb{R}^p \rightrightarrows \mathbb{R}^q$ is a function from $\mathbb{R}^p$ to the set of subsets of $\mathbb{R}^q$.

- ∂f, the subgradient of a convex function f.
- $\partial^c f$, the Clarke subgradient of a locally Lipschitz function f

$$\partial^c f(x) = \mathrm{conv}\left\{ v \in \mathbb{R}^p,\ \exists y_k \underset{k\to\infty}{\to} x \text{ with } y_k \in R,\ v_k = \nabla f(y_k) \underset{k\to\infty}{\to} v \right\}.$$

where R is the (full measure set) where f is differentiable.

Closed graph: a notion of continuity for D

$$\mathrm{graph}\, D = \{(x,z),\ x \in \mathbb{R}^p,\ z \in D(x)\} \subset \mathbb{R}^{p+q},$$

If $v_k \in D(x_k)$ for all $k \in \mathbb{N}$, $\lim_{k\to\infty} v_k \in D(\lim_{k\to\infty} x_k)$ (provided limits exist).

Conservative set valued fields

$D\colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$, set valued, closed graph, non empty compact values.

Conservative set valued fields

$D\colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$, set valued, closed graph, non empty compact values.
Conservative field: For any AC loop $\gamma\colon [0,1] \mapsto \mathbb{R}^p$, $\gamma(0) = \gamma(1)$,

$$\int_0^1 \max_{v \in D(\gamma(t))} \langle \dot{\gamma}(t), v \rangle \, dt = 0$$

Lebsegue integral.

Conservative set valued fields

$D \colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$, set valued, closed graph, non empty compact values.
Conservative field: For any AC loop $\gamma \colon [0,1] \mapsto \mathbb{R}^p$, $\gamma(0) = \gamma(1)$,

$$\int_0^1 \max_{v \in D(\gamma(t))} \langle \dot\gamma(t), v \rangle \, dt = 0$$

Lebsegue integral.

Equivalent forms: With min or set valued (Auman) integral.

Conservative set valued fields

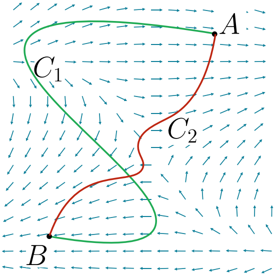
$D\colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$, set valued, closed graph, non empty compact values.
Conservative field: For any AC loop $\gamma\colon [0,1] \mapsto \mathbb{R}^p$, $\gamma(0) = \gamma(1)$,

$$\int_0^1 \max_{v \in D(\gamma(t))} \langle \dot{\gamma}(t), v \rangle \, dt = 0$$

Lebsegue integral.

Equivalent forms: With min or set valued (Auman) integral.

Links with physics:



Locally Lipschitz potentials

Potential: $D\colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ a conservative field. Define $f\colon \mathbb{R}^p \mapsto \mathbb{R}$,

$$f(x) = f(0) + \int_0^1 \max_{v \in D(\gamma(t))} \langle \dot{\gamma}(t), v \rangle \, dt$$

where $\gamma\colon [0,1] \mapsto \mathbb{R}^p$ is any AC path with $\gamma(0) = 0$, $\gamma(1) = x$.

Locally Lipschitz potentials

Potential: $D\colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ a conservative field. Define $f\colon \mathbb{R}^p \mapsto \mathbb{R}$,

$$f(x) = f(0) + \int_0^1 \max_{v \in D(\gamma(t))} \langle \dot{\gamma}(t), v \rangle \, dt$$

where $\gamma\colon [0,1] \mapsto \mathbb{R}^p$ is any AC path with $\gamma(0) = 0$, $\gamma(1) = x$.
f is well and uniquely defined up to a constant.

Locally Lipschitz potentials

Potential: $D\colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ a conservative field. Define $f\colon \mathbb{R}^p \mapsto \mathbb{R}$,

$$f(x) = f(0) + \int_0^1 \max_{v \in D(\gamma(t))} \langle \dot{\gamma}(t), v \rangle \, dt$$

where $\gamma\colon [0,1] \mapsto \mathbb{R}^p$ is any AC path with $\gamma(0) = 0$, $\gamma(1) = x$.
f is well and uniquely defined up to a constant.

- f is a *potential for* D.
- *D is a conservative field for f.*

Locally Lipschitz potentials

Potential: $D\colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ a conservative field. Define $f\colon \mathbb{R}^p \mapsto \mathbb{R}$,

$$f(x) = f(0) + \int_0^1 \max_{v \in D(\gamma(t))} \langle \dot{\gamma}(t), v \rangle \, dt$$

where $\gamma\colon [0,1] \mapsto \mathbb{R}^p$ is any AC path with $\gamma(0) = 0$, $\gamma(1) = x$.
f is well and uniquely defined up to a constant.

- f is a *potential for* D.
- D *is a conservative field for* f.

Equivalent forms: With min or set valued (Auman) integral.

Locally Lipschitz potentials

Potential: $D \colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ a conservative field. Define $f \colon \mathbb{R}^p \mapsto \mathbb{R}$,

$$f(x) = f(0) + \int_0^1 \max_{v \in D(\gamma(t))} \langle \dot{\gamma}(t), v \rangle \, dt$$

where $\gamma \colon [0,1] \mapsto \mathbb{R}^p$ is any AC path with $\gamma(0) = 0$, $\gamma(1) = x$.
f is well and uniquely defined up to a constant.

- f is a *potential for* D.
- D *is a conservative field for* f.

Equivalent forms: With min or set valued (Auman) integral.

D is locally bounded (by assumption) and f is locally Lipschitz.

Potential: $D\colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ a conservative field. Define $f\colon \mathbb{R}^p \mapsto \mathbb{R}$,

$$f(x) = f(0) + \int_0^1 \max_{v \in D(\gamma(t))} \langle \dot\gamma(t), v \rangle \, dt$$

where $\gamma\colon [0,1] \mapsto \mathbb{R}^p$ is any AC path with $\gamma(0) = 0$, $\gamma(1) = x$.
f is well and uniquely defined up to a constant.

- f is a *potential for* D.
- *D is a conservative field for f.*

Equivalent forms: With min or set valued (Auman) integral.

D is locally bounded (by assumption) and f is locally Lipschitz.

- f C^1: $\{\nabla f\}$ is conservative for f (not unique).
- f convex locally Lipschitz: ∂f is conservative for f.
- Not all locally Lipschitz f admit a conservative field.

An operational chain rule

Lemma: The following are equivalent

- $D\colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ is conservative for $f : \mathbb{R}^p \mapsto \mathbb{R}$.
- For any AC $\gamma\colon [0,1] \mapsto \mathbb{R}^p$

$$\frac{d}{dt} f(\gamma(t)) = \langle v, \dot{\gamma}(t) \rangle \qquad \forall v \in D(\gamma(t)), \qquad \text{a.e.} \quad t \in [0,1].$$

An operational chain rule

Lemma: The following are equivalent

- $D\colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ is conservative for $f\colon \mathbb{R}^p \mapsto \mathbb{R}$.
- For any AC $\gamma\colon [0,1] \mapsto \mathbb{R}^p$

$$\frac{d}{dt} f(\gamma(t)) = \langle v, \dot\gamma(t)\rangle \qquad \forall v \in D(\gamma(t)), \qquad \text{a.e.} \quad t \in [0,1].$$

Affine span of $D(\gamma(t))$ is "orthogonal" to $\dot\gamma$ for almost all t and any γ.

Lemma: The following are equivalent

- $D\colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ is conservative for $f\colon \mathbb{R}^p \mapsto \mathbb{R}$.
- For any AC $\gamma\colon [0,1] \mapsto \mathbb{R}^p$

$$\frac{d}{dt} f(\gamma(t)) = \langle v, \dot{\gamma}(t) \rangle \qquad \forall v \in D(\gamma(t)), \qquad \text{a.e.} \quad t \in [0,1].$$

Affine span of $D(\gamma(t))$ is "orthogonal" to $\dot{\gamma}$ for almost all t and any γ.

Theorem: If f is locally Lipschitz and tame then $\partial^c f$ is conservative for f.
Davis *et. al.* 2019. Stochastic subgradient method converges on tame functions. FOCM.

An operational chain rule

Lemma: The following are equivalent

- $D \colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ is conservative for $f \colon \mathbb{R}^p \mapsto \mathbb{R}$.
- For any AC $\gamma \colon [0,1] \mapsto \mathbb{R}^p$

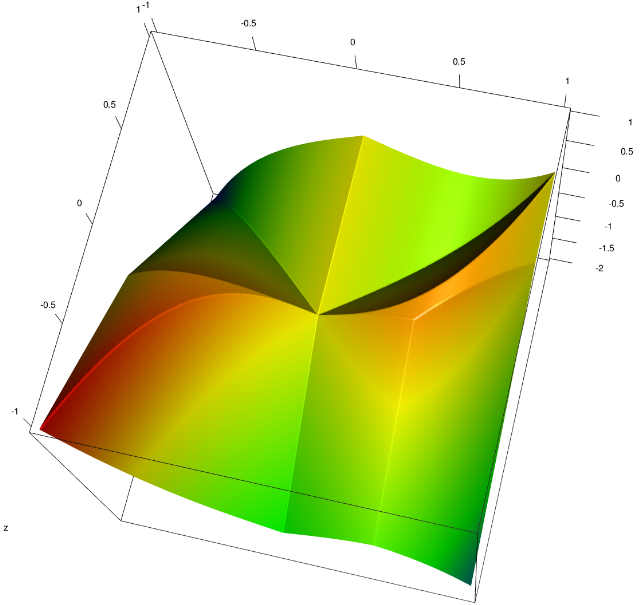
$$\frac{d}{dt} f(\gamma(t)) = \langle v, \dot{\gamma}(t) \rangle \qquad \forall v \in D(\gamma(t)), \qquad \text{a.e.} \quad t \in [0,1].$$

Affine span of $D(\gamma(t))$ is "orthogonal" to $\dot{\gamma}$ for almost all t and any γ.

Theorem: If f is locally Lipschitz and tame then $\partial^c f$ is conservative for f.
Davis *et. al.* 2019. Stochastic subgradient method converges on tame functions. FOCM.

- Chain rule is central for Lyapunov analysis of minibatch strategies.

Illustration



Outline

1. Conservative set valued field

2. Properties of conservative fields

3. Consequences for deep learning

Relation to gradients

Let $D \colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ be a conservative field for $f \colon \mathbb{R}^p \mapsto \mathbb{R}$.

Gradient almost everywhere: $D = \{\nabla f\}$ Lebesgue almost everywhere.

Relation to gradients

Let $D \colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ be a conservative field for $f \colon \mathbb{R}^p \mapsto \mathbb{R}$.

Gradient almost everywhere: $D = \{\nabla f\}$ Lebesgue almost everywhere.

Consequence: $\partial^c f$ is conservative for f, and for all $x \in \mathbb{R}^p$,

$$\partial^c f(x) \subset \mathrm{conv}(D(x)).$$

Fermat rule: $0 \in \mathrm{conv}(D)$ for local minima.

Relation to gradients

Let $D \colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ be a conservative field for $f \colon \mathbb{R}^p \mapsto \mathbb{R}$.

Gradient almost everywhere: $D = \{\nabla f\}$ Lebesgue almost everywhere.

Consequence: $\partial^c f$ is conservative for f, and for all $x \in \mathbb{R}^p$,

$$\partial^c f(x) \subset \mathrm{conv}(D(x)).$$

Fermat rule: $0 \in \mathrm{conv}(D)$ for local minima.

Remark: Conservativity is much stronger than "gradient almost everywhere".

Take $f = \|\cdot\|^2$ and set $D = \{\nabla f\}$ and $D = \{\nabla f, 0\}$ on a segment $[x, y]$,
D is compact valued with closed graph, gradient almost everywhere but not conservative.

Conservative fields and calculus

Informal: Conservative set valued fields are compatible with the compositional rules of differential calculus.

Informal: Conservative set valued fields are compatible with the compositional rules of differential calculus.

Sum rule: Let $f_1, \ldots, f_n$ be locally Lipschitz continuous functions and $D_1, \ldots, D_n$ respective conservative fields. Then $D = \sum_{i=1}^n D_i$ is conservative for $f = \sum_{i=1}^n f_i$.

Conservative fields and calculus

Informal: Conservative set valued fields are compatible with the compositional rules of differential calculus.

Sum rule: Let $f_1, \ldots, f_n$ be locally Lipschitz continuous functions and $D_1, \ldots, D_n$ respective conservative fields. Then $D = \sum_{i=1}^n D_i$ is conservative for $f = \sum_{i=1}^n f_i$.

Chain rule along AC curves + sum rule for derivatives + union of zero measure sets has zero measure:

$$\frac{d}{dt}(f_1(\gamma(t)) + f_2(\gamma(t))) = \langle v_1, \dot{\gamma}(t)\rangle + \langle v_2, \dot{\gamma}(t)\rangle = \langle v_1 + v_2, \dot{\gamma}(t)\rangle$$
$$\forall v_1 \in D_1(\gamma(t)),\ v_2 \in D_2(\gamma(t))$$

Informal: Conservative set valued fields are compatible with the compositional rules of differential calculus.

Sum rule: Let $f_1, \ldots, f_n$ be locally Lipschitz continuous functions and $D_1, \ldots, D_n$ respective conservative fields. Then $D = \sum_{i=1}^n D_i$ is conservative for $f = \sum_{i=1}^n f_i$.

Chain rule along AC curves + sum rule for derivatives + union of zero measure sets has zero measure:

$$\frac{d}{dt}(f_1(\gamma(t)) + f_2(\gamma(t))) = \langle v_1, \dot{\gamma}(t)\rangle + \langle v_2, \dot{\gamma}(t)\rangle = \langle v_1 + v_2, \dot{\gamma}(t)\rangle$$
$$\forall v_1 \in D_1(\gamma(t)),\ v_2 \in D_2(\gamma(t))$$

Consequence for AD (informal): A program combines locally Lipschitz elementary functions in a locally Lipschitz way.
AD with conservative fields in place of gradients, output a conservative field for the implemented function.

Outline

1. Conservative set valued field
2. Properties of conservative fields
3. Consequences for deep learning

Deep networks and tamness

Training: Given $\{(x_i, y_i)\}_{i=1}^{n}$ in $\mathbb{R}^p \times \mathbb{R}^{p_L}$ and a loss $\ell \colon \mathbb{R}^{p_L} \times \mathbb{R}^{p_L} \to \mathbb{R}_+$.

$$\min_{\theta} \qquad J(\theta) \quad := \quad \frac{1}{n}\sum_{i=1}^{n} \ell(F_\theta(x_i), y_i) \quad = \quad \frac{1}{n}\sum_{i=1}^{n} J_i(\theta).$$

Training: Given $\{(x_i, y_i)\}_{i=1}^n$ in $\mathbb{R}^p \times \mathbb{R}^{p_L}$ and a loss $\ell \colon \mathbb{R}^{p_L} \times \mathbb{R}^{p_L} \to \mathbb{R}_+$.

$$\min_\theta \qquad J(\theta) \quad := \quad \frac{1}{n}\sum_{i=1}^n \ell(F_\theta(x_i), y_i) \quad = \quad \frac{1}{n}\sum_{i=1}^n J_i(\theta).$$

Assumption: ℓ and the activation functions defining F_θ are

- Univariate (applied coordinatewise).
- Locally Lipschitz.
- Defined piecewise (finitely many pieces).
- Expressed with, polynomials, quotients, exponential, logarithms.

Training: Given $\{(x_i, y_i)\}_{i=1}^n$ in $\mathbb{R}^p \times \mathbb{R}^{p_L}$ and a loss $\ell \colon \mathbb{R}^{p_L} \times \mathbb{R}^{p_L} \to \mathbb{R}_+$.

$$\min_\theta \qquad J(\theta) \quad := \quad \frac{1}{n}\sum_{i=1}^n \ell(F_\theta(x_i), y_i) \quad = \quad \frac{1}{n}\sum_{i=1}^n J_i(\theta).$$

Assumption: ℓ and the activation functions defining F_θ are

- Univariate (applied coordinatewise).
- Locally Lipschitz.
- Defined piecewise (finitely many pieces).
- Expressed with, polynomials, quotients, exponential, logarithms.

Tameness: Then J is locally Lipschitz and "tame", *i.e.* definable in an o-minimal structure (contains all semi-algebraic sets and the graph of the exponential function [Wilkie]).

Nonsmooth automatic differentiation for deep networks

Nonsmooth backpropagation:

- Consider $J\colon \mathbb{R}^p \mapsto \mathbb{R}$ the empirical loss.
- Set $D_i\colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$,
  AD on J_i using Clarke subgradient in place of derivatives $(\mathrm{relu}'(0) = 0)$.
- Set $D = \frac{1}{n}\sum_{i=1}^n D_i$.
- Set $\mathrm{crit}_J = \{\theta \in \mathbb{R}^p, \quad 0 \in \mathrm{conv}(D(\theta))\}$.

Nonsmooth backpropagation:

- Consider $J \colon \mathbb{R}^p \mapsto \mathbb{R}$ the empirical loss.
- Set $D_i \colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$,
  AD on J_i using Clarke subgradient in place of derivatives $(\mathrm{relu}'(0) = 0)$.
- Set $D = \frac{1}{n} \sum_{i=1}^n D_i$.
- Set $\mathrm{crit}_J = \{\theta \in \mathbb{R}^p, \quad 0 \in \mathrm{conv}(D(\theta))\}$.

Then:

Conservativity: D is conservative for J.

$$\{J(\theta_2) - J(\theta_1)\} = \int_0^1 \langle D((1-t)\theta_1 + t\theta_2), \theta_2 - \theta_1 \rangle \, dt,$$

Nonsmooth backpropagation:

- Consider $J \colon \mathbb{R}^p \mapsto \mathbb{R}$ the empirical loss.
- Set $D_i \colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$,
  AD on J_i using Clarke subgradient in place of derivatives $(\mathrm{relu}'(0) = 0)$.
- Set $D = \frac{1}{n} \sum_{i=1}^n D_i$.
- Set $\mathrm{crit}_J = \{\theta \in \mathbb{R}^p, \quad 0 \in \mathrm{conv}(D(\theta))\}$.

Then:

Conservativity: D is conservative for J.

$$\{J(\theta_2) - J(\theta_1)\} = \int_0^1 \langle D((1-t)\theta_1 + t\theta_2), \theta_2 - \theta_1 \rangle \, dt,$$

Gradient: $D = \{\nabla J\}$ except on a finite union of smooth manifolds of dimension $< p$.

Nonsmooth backpropagation:

- Consider $J \colon \mathbb{R}^p \mapsto \mathbb{R}$ the empirical loss.
- Set $D_i \colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$,
  AD on J_i using Clarke subgradient in place of derivatives $(\mathrm{relu}'(0) = 0)$.
- Set $D = \frac{1}{n} \sum_{i=1}^n D_i$.
- Set $\mathrm{crit}_J = \{\theta \in \mathbb{R}^p, \quad 0 \in \mathrm{conv}(D(\theta))\}$.

Then:

Conservativity: D is conservative for J.

$$\{J(\theta_2) - J(\theta_1)\} = \int_0^1 \langle D((1-t)\theta_1 + t\theta_2), \theta_2 - \theta_1 \rangle \, dt,$$

Gradient: $D = \{\nabla J\}$ except on a finite union of smooth manifolds of dimension $< p$.

Morse-Sard: The set of critical values is finite.

$$J(\mathrm{crit}_J) = \{J(\theta), \quad 0 \in \mathrm{conv}(D(\theta))\}$$

Nonsmooth automatic differentiation for deep networks

Nonsmooth backpropagation:

- Consider $J\colon \mathbb{R}^p \mapsto \mathbb{R}$ the empirical loss.
- Set $D_i\colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$,
  AD on J_i using Clarke subgradient in place of derivatives $(\mathrm{relu}'(0) = 0)$.
- Set $D = \frac{1}{n}\sum_{i=1}^n D_i$.
- Set $\mathrm{crit}_J = \{\theta \in \mathbb{R}^p, \quad 0 \in \mathrm{conv}(D(\theta))\}$.

Then:

Conservativity: D is conservative for J.

$$\{J(\theta_2) - J(\theta_1)\} = \int_0^1 \langle D((1-t)\theta_1 + t\theta_2), \theta_2 - \theta_1\rangle \, dt,$$

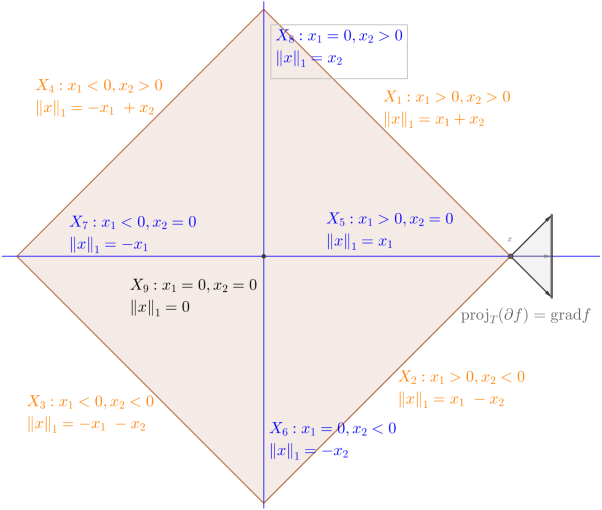
Gradient: $D = \{\nabla J\}$ except on a finite union of smooth manifolds of dimension $< p$.

Morse-Sard: The set of critical values is finite.

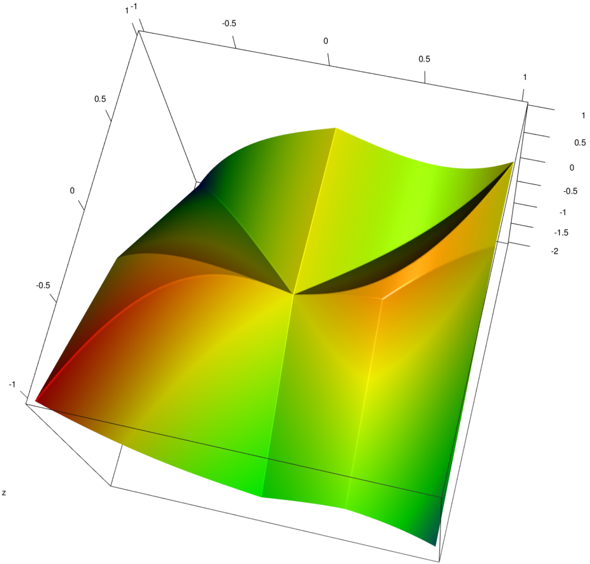
$$J(\mathrm{crit}_J) = \{J(\theta), \quad 0 \in \mathrm{conv}(D(\theta))\}$$

KL inequality: There is a Kurdyka-Łojasiewicz inequality for D and J.

Example: Projection formula $f(x_1, x_2) = |x_1| + |x_2|$.



Example: Projection formula .



Minibatch strategies

Minibatch stochastic approximation: Given $(I_k)_{k\in\mathbb{N}}$ iid, uniform on $\{1,\ldots,n\}$, $(\alpha_k)_{k\in\mathbb{N}}$ positive, iterate,

$$\theta_{k+1} \in \theta_k - \alpha_k D_{I_k}(\theta_k)$$

Minibatch stochastic approximation: Given $(I_k)_{k\in\mathbb{N}}$ iid, uniform on $\{1,\ldots,n\}$, $(\alpha_k)_{k\in\mathbb{N}}$ positive, iterate,

$$\theta_{k+1} \in \theta_k - \alpha_k D_{I_k}(\theta_k)$$

Convergence:
Assume that $\sum_k \alpha_k = +\infty$ and $\alpha_k = o(1/\log(k))$.
Fix any $M > 0$, condition on the event $\sup_{k\in\mathbb{N}} \|\theta_k\| \leq M$.
Set, $\Theta \subset \mathbb{R}^p$, the set of accumulation points of $(\theta_k)_{k\in\mathbb{N}}$.

Then, almost surely, $\emptyset \neq \Theta \subset \mathrm{crit}_J$ and J is constant on Θ.

Minibatch stochastic approximation: Given $(I_k)_{k\in\mathbb{N}}$ iid, uniform on $\{1,\dots,n\}$, $(\alpha_k)_{k\in\mathbb{N}}$ positive, iterate,

$$\theta_{k+1} \in \theta_k - \alpha_k D_{I_k}(\theta_k)$$

Convergence:
Assume that $\sum_k \alpha_k = +\infty$ and $\alpha_k = o(1/\log(k))$.
Fix any $M > 0$, condition on the event $\sup_{k\in\mathbb{N}} \|\theta_k\| \leq M$.
Set, $\Theta \subset \mathbb{R}^p$, the set of accumulation points of $(\theta_k)_{k\in\mathbb{N}}$.

Then, almost surely, $\emptyset \neq \Theta \subset \mathrm{crit}_J$ and J is constant on Θ.

Differential inclusion approach [Benaim-Hofbauer-Sorin (2005)].

- Conservativity: chain rule along AC curves.
- Tameness: Morse-Sard theorem.

Summary and conclusion: functions, programs and numerics

Smooth:

$$\begin{tikzcd} J \arrow[r, "\mathrm{num}"] \arrow[d, "\mathrm{diff}"'] & \mathcal{P} \arrow[d, "\mathrm{autodiff}"] \\ \nabla J \arrow[r, "\mathrm{num}"'] & D \end{tikzcd}$$

Nonsmooth:

$$\begin{tikzcd} J \arrow[r, "\mathrm{num}"] \arrow[d, "\mathrm{diff}"'] \arrow[dr, dash, dashed, "\mathrm{conservative}"] & \mathcal{P} \arrow[d, "\mathrm{autodiff}"] \\ \partial J \arrow[r, dash, dashed, "\text{``}\subset\text{''}"'] & D \end{tikzcd}$$

Smooth:

$$
\begin{tikzcd}
J \arrow[r, "\text{num}"] \arrow[d, "\text{diff}"'] & \mathcal{P} \arrow[d, "\text{autodiff}"] \\
\nabla J \arrow[r, "\text{num}"'] & D
\end{tikzcd}
$$

Nonsmooth:

$$
\begin{tikzcd}
J \arrow[r, "\text{num}"] \arrow[d, "\text{diff}"'] \arrow[dr, dashed, no head, "\text{conservative}" sloped] & \mathcal{P} \arrow[d, "\text{autodiff}"] \\
\partial J \arrow[r, dashed, no head, "\text{“}\subset\text{”}"'] & D
\end{tikzcd}
$$

A mathematical model for nonsmooth automatic differentiation.

- **Algorithms:** Nonsmooth AD $+$ minibatching deep nets $\sim$ smooth case.

Abadi M., Barham P., Chen J., Chen Z., Davis A., Dean J., Devin M., Ghemawat S., Irving G., Isard M., Kudlur M., Levenberg J., Monga R., Moore S., Murray D., Steiner B., Tucker P., Vasudevan V., Warden P., Wicke M., Yu Y. and Zheng X. (2016).
Tensorflow: A system for large-scale machine learning.
In Symposium on Operating Systems Design and Implementation.

Aliprantis C.D., Border K.C. (2005)
Infinite Dimensional Analysis (3rd edition)
Springer

Attouch H., Goudou X. and Redont P. (2000).
The heavy ball with friction method, I. The continuous dynamical system: global exploration of the local minima of a real-valued function by asymptotic analysis of a dissipative dynamical system.
Communications in Contemporary Mathematics, 2(01), 1-34.

Aubin, J. P., Cellina, A. (1984).
Differential inclusions: set-valued maps and viability theory (Vol. 264). Springer.

Aubin, J.-P., and Frankowska, H. (2009).
Set-valued analysis. Springer Science & Business Media.

Baydin A., Pearlmutter B., Radul A. and Siskind J. (2018).
Automatic differentiation in machine learning: a survey.
Journal of machine learning research, 18(153).

Benaïm, M. (1999).
Dynamics of stochastic approximation algorithms.
In Séminaire de probabilités XXXIII (pp. 1-68). Springer, Berlin, Heidelberg.

Benaïm, M., Hofbauer, J., Sorin, S. (2005).
Stochastic approximations and differential inclusions.
SIAM Journal on Control and Optimization, 44(1), 328-348.

Bolte, J., Daniilidis, A., Lewis, A., Shiota, M. (2007).
Clarke subgradients of stratifiable functions.
SIAM Journal on Optimization, 18(2), 556-572.

Bolte J., Sabach S., and Teboulle M. (2014).
Proximal alternating linearized minimization for nonconvex and nonsmooth problems.
Mathematical Programming, 146(1-2), 459-494.

Borkar, V. (2009).
Stochastic approximation: a dynamical systems viewpoint (Vol. 48). Springer.

Borwein J. and Lewis A. S. (2010).
Convex analysis and nonlinear optimization: theory and examples.
Springer Science & Business Media.

Borwein J. M. and Moors W. B. (1997).
Essentially smooth Lipschitz functions.
Journal of functional analysis, 149(2), 305-351.

Borwein J. M. and Moors, W. B. (1998).
A chain rule for essentially smooth Lipschitz functions.
SIAM Journal on Optimization, 8(2), 300-308.

Borwein, J., Moors, W. and Wang, X. (2001).
Generalized subdifferentials: a Baire categorical approach.
Transactions of the American Mathematical Society, 353(10), 3875-3893.

Bottou L. and Bousquet O. (2008).
The tradeoffs of large scale learning.
In Advances in neural information processing systems (pp. 161-168).

Bottou L., Curtis F. E. and Nocedal J. (2018).
Optimization methods for large-scale machine learning.
Siam Review, 60(2), 223-311.

Castera C., Bolte J., Févotte C., Pauwels E. (2019).
An Inertial Newton Algorithm for Deep Learning.
arXiv preprint arXiv:1905.12278.

Clarke F. H. (1983).
Optimization and nonsmooth analysis.
Siam.

Chizat, L., and Bach, F. (2018).
On the global convergence of gradient descent for over-parameterized models using optimal transport.
In Advances in neural information processing systems, 3036-3046.

Corliss G., Faure C., Griewank A., Hascoet L. and Naumann U. (Editors) (2002).
Automatic differentiation of algorithms: from simulation to optimization.
Springer Science & Business Media.

Correa R. and Jofre, A. (1989).
Tangentially continuous directional derivatives in nonsmooth analysis.
Journal of optimization theory and applications, 61(1), 1-21.

Coste M., *An introduction to o-minimal geometry* (1999).
RAAG notes, Institut de Recherche Mathématique de Rennes.

Davis, D., Drusvyatskiy, D., Kakade, S., Lee, J. D. (2018).
Stochastic subgradient method converges on tame functions.
Foundations of Computational Mathematics.

van den Dries L. and Miller C. (1996).
Geometric categories and o-minimal structures.
Duke Math. J, 84(2), 497-540.

Evans, L. C. and Gariepy, R. F. (2015).
Measure theory and fine properties of functions.
Revised Edition. Chapman and Hall/CRC.

Glorot X., Bordes A. and Bengio Y. (2011).
Deep sparse rectifier neural networks.
In Proceedings of the fourteenth international conference on artificial intelligence and statistics (pp. 315-323).

Griewank, A., Walther, A. (2008).
Evaluating derivatives: principles and techniques of algorithmic differentiation (Vol. 105).
SIAM.

Griewank A. (2013).
On stable piecewise linearization and generalized algorithmic differentiation.
Optimization Methods and Software, 28(6), 1139-1178.

Griewank A., Walther A., Fiege S. and Bosse T. (2016).
On Lipschitz optimization based on gray-box piecewise linearization.
Mathematical Programming, 158(1-2), 383-415.

Ioffe A. D. (1981).
Nonsmooth analysis: differential calculus of nondifferentiable mappings.
Transactions of the American Mathematical Society, 266(1), 1-56.

Ioffe, A. D. (2017).
Variational analysis of regular mappings.
Springer Monographs in Mathematics. Springer, Cham.

Kakade, S. M. and Lee, J. D. (2018).
Provably correct automatic sub-differentiation for qualified programs.
In Advances in Neural Information Processing Systems (pp. 7125-7135).

Kurdyka, K. (1998).
On gradients of functions definable in o-minimal structures.
In Annales de l'institut Fourier 48(3), 769-783.

Kurdyka, K., Mostowski, T. and Parusinski, A. (2000).
Proof of the gradient conjecture of R. Thom.
Annals of Mathematics, 152(3), 763-792.

Kushner H. and Yin, G. G. (2003).
Stochastic approximation and recursive algorithms and applications (Vol. 35).
Springer Science & Business Media.

LeCun Y., Bengio Y., Hinton, G. (2015).
Deep learning.
Nature, 521(7553).

Ljung L. (1977).
Analysis of recursive stochastic algorithms.
IEEE transactions on automatic control, 22(4), 551-575.

Majewski, S., Miasojedow, B. and Moulines, E. (2018).
Analysis of nonsmooth stochastic approximation: the differential inclusion approach.
arXiv preprint arXiv:1805.01916.

Mohammadi, B. and Pironneau, O. (2010).
Applied shape optimization for fluids.
Oxford university press.

Moulines E. and Bach, F. (2011).
Non-asymptotic analysis of stochastic approximation algorithms for machine learning.
In Advances in Neural Information Processing Systems (pp. 451-459).

Moreau J.-J. (1963).
Fonctionnelles sous-différentiables.

Mordukhovich B. S. (2006).
Variational analysis and generalized differentiation I: Basic theory.
Springer Science & Business Media.

Paszke A., Gross S., Chintala S., Chanan G., Yang E., DeVito Z., Lin Z., Desmaison A., Antiga L. and Lerer A. (2017).
Automatic differentiation in pytorch.
In NIPS workshops.

Robbins H. and Monro, S. (1951).
A stochastic approximation method.
The annals of mathematical statistics, 400-407.

Rockafellar R. T. (1963).
Convex functions and dual extremum problems.
Doctoral dissertation, Harvard University.

Rockafellar R. (1970).
On the maximal monotonicity of subdifferential mappings.
Pacific Journal of Mathematics, 33(1), 209-216.

Rockafellar, R. T., Wets, R. J. B. (1998).
Variational analysis.
Springer.

Rumelhart E., Hinton E., Williams J. (1986).
Learning representations by back-propagating errors.
Nature 323:533-536.

Speelpenning, B. (1980).
Compiling fast partial derivatives of functions given by algorithms (No. COO-2383-0063; UILU-ENG-80-1702; UIUCDCS-R-80-1002).
Illinois Univ., Urbana (USA). Dept. of Computer Science.

Thibault, L. (1982).
On generalized differentials and subdifferentials of Lipschitz vector-valued functions.
Nonlinear Analysis: Theory, Methods & Applications, 6(10), 1037-1053.

Thibault, L. and Zagrodny, D. (1995).
Integration of subdifferentials of lower semicontinuous functions on Banach spaces.
Journal of Mathematical Analysis and Applications, 189(1), 33-58.

Thibault, L. and Zlateva, N. (2005).
Integrability of subdifferentials of directionally Lipschitz functions.
Proceedings of the American Mathematical Society, 2939-2948.

Valadier, M. (1989).
Entraînement unilatéral, lignes de descente, fonctions lipschitziennes non pathologiques.
Comptes rendus de l'Académie des Sciences, 308, 241-244.

Wang X. (1995).
Pathological Lipschitz functions in $\mathbb{R}^n$.
Master Thesis, Simon Fraser University.