A mathematical model for nonsmooth algorithmic differentiation with applications to machine learning

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

Séminaire du CAS
MinesParistech

Novembre 2020

Plan

Motivation: Chain rule of differentiation is widely used outside of its domain of validity. Ex: algorithmic differentiation for deep learning with nonsmooth components.

Differential calculus rules applied to subgradients do not provide subgradients in general.

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

Backpropagation

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

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

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

$$\nabla(f \circ G \circ H)^T = \nabla f^T \times \mathrm{Jac}_G \times \mathrm{Jac}_H$$

Function = program: composition of smooth functions.

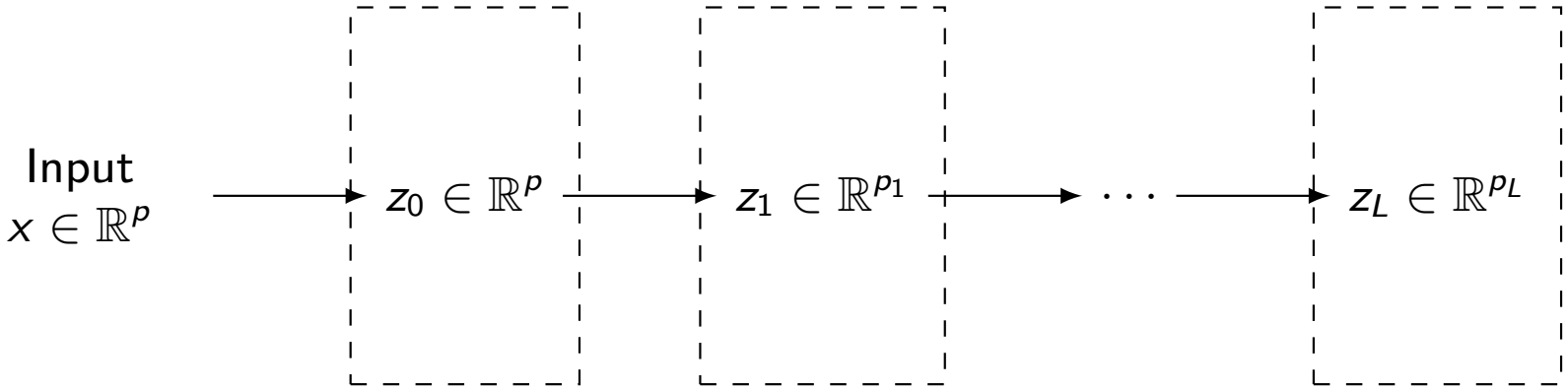
$$x \mapsto (H(x), G(H(x)), f(G(H(x))))$$

Forward mode of AD: $\nabla f^T \times (\mathrm{Jac}_G \times \mathrm{Jac}_H)$.
Backward mode of AD: $(\nabla f^T \times \mathrm{Jac}_G) \times \mathrm{Jac}_H$.

Backpropagation: Backward AD for neural network training.

It computes gradient (provided that everybody is smooth).

Neural network / compositional modeling



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.

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

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

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

Backpropagation and learning

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,

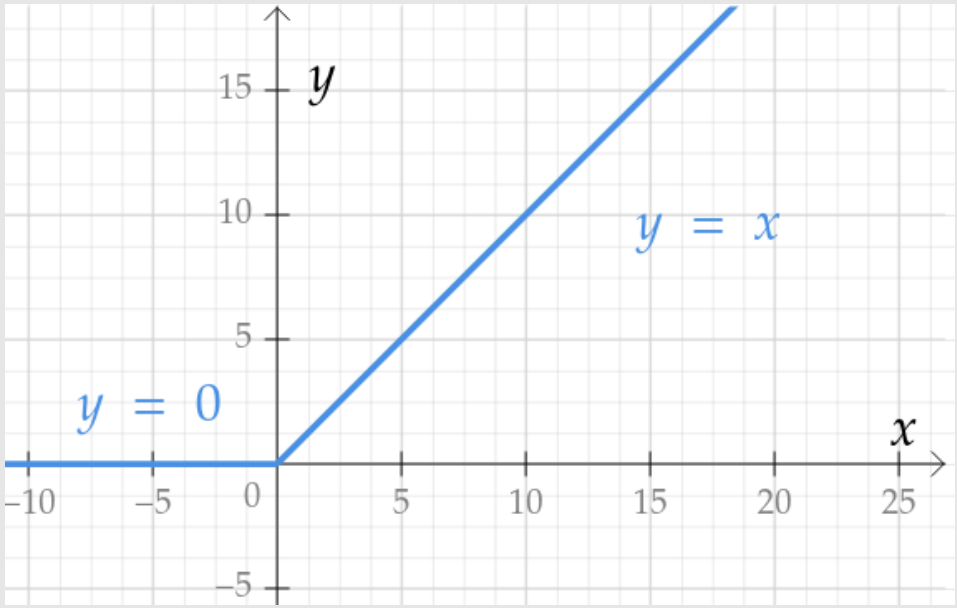
$$\theta_{k+1} = \theta_k - \alpha_k \nabla J_{I_k}(\theta_k).$$

Backpropagation: Backward mode of algorithmic 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).

Nonsmooth activations

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



Less straightforward examples:

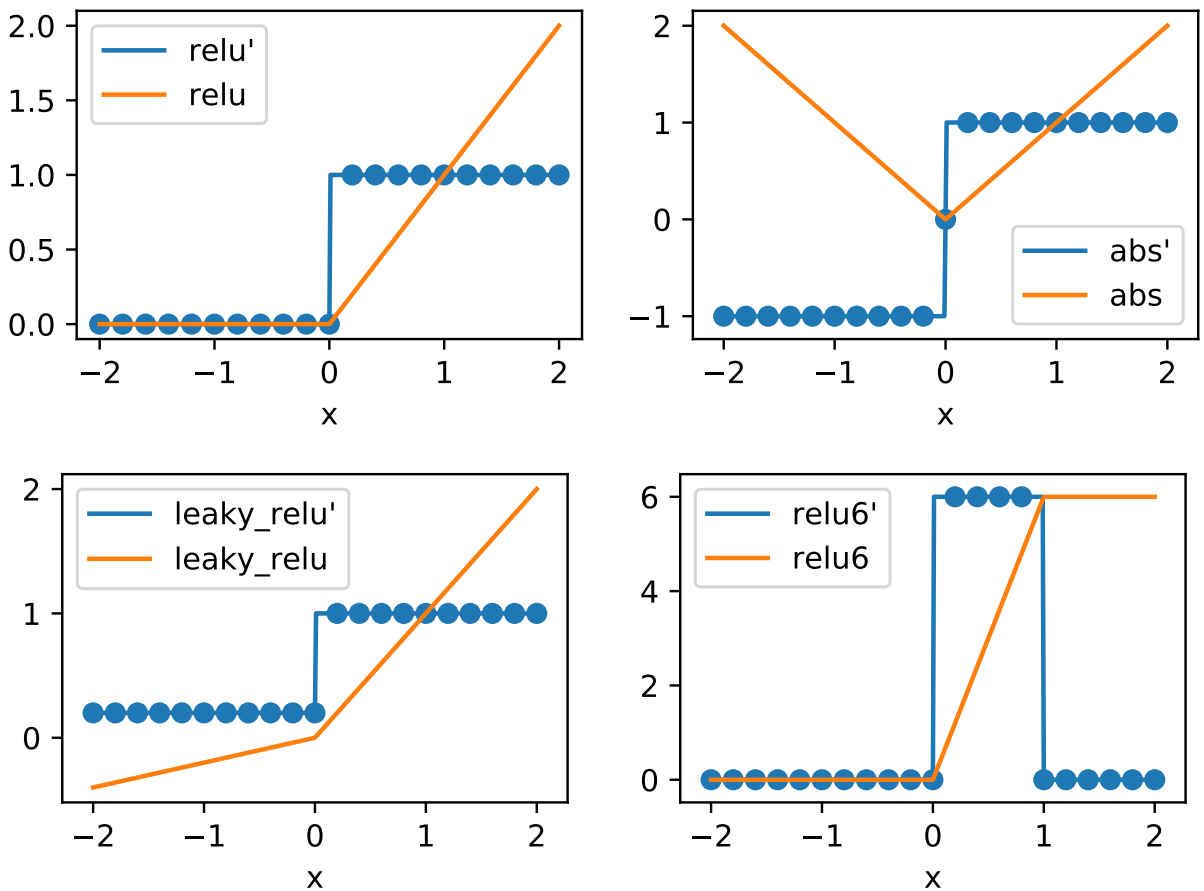
- Max pooling in convolutional networks.
- knn grouping layers, farthest point subsampling layers.
  Qi *et. al.* 2017. PointNet++: Deep Hierarchical Feature Learning on point Sets in a Metric Space.
- Sorting layers.
  Anil *et. al.* 2019. Sorting Out Lipschitz Function Approximation. ICML.

Nonsmooth backpropagation

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

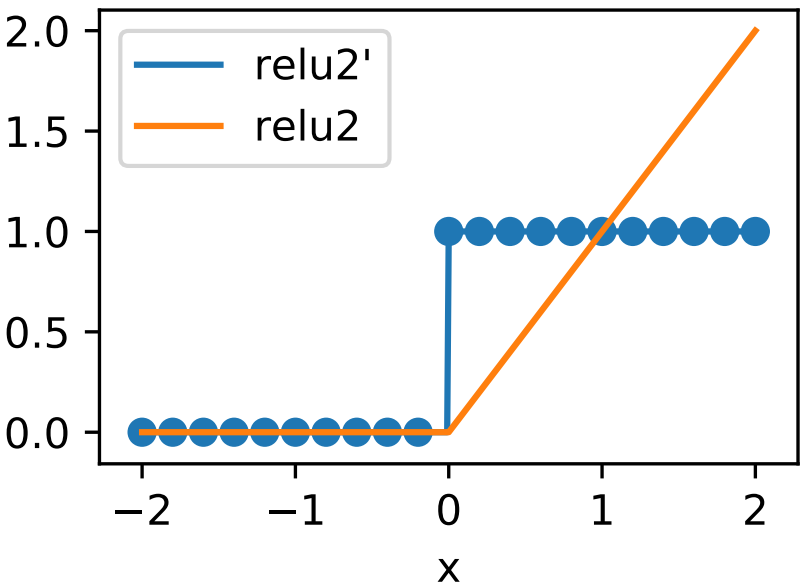
$$(f \circ g)' = g' \times f' \circ g.$$

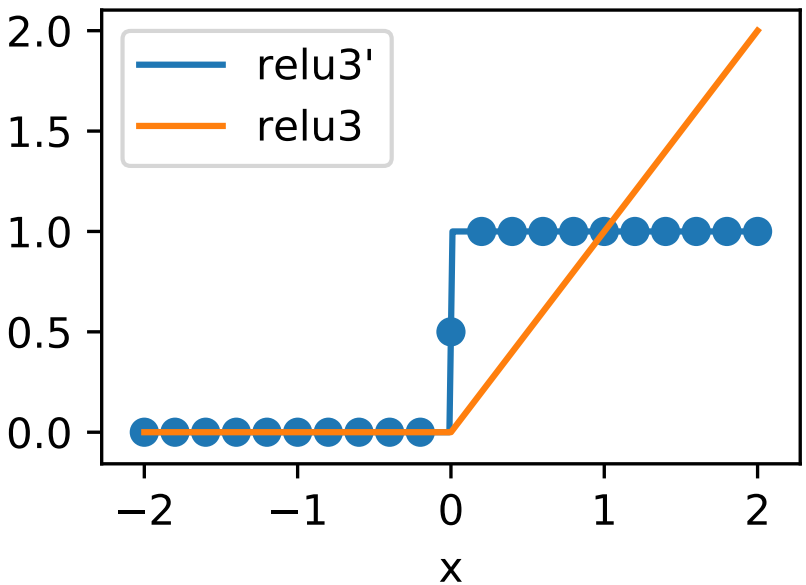
Tensorflow examples:



AD acts on programs, not on functions

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

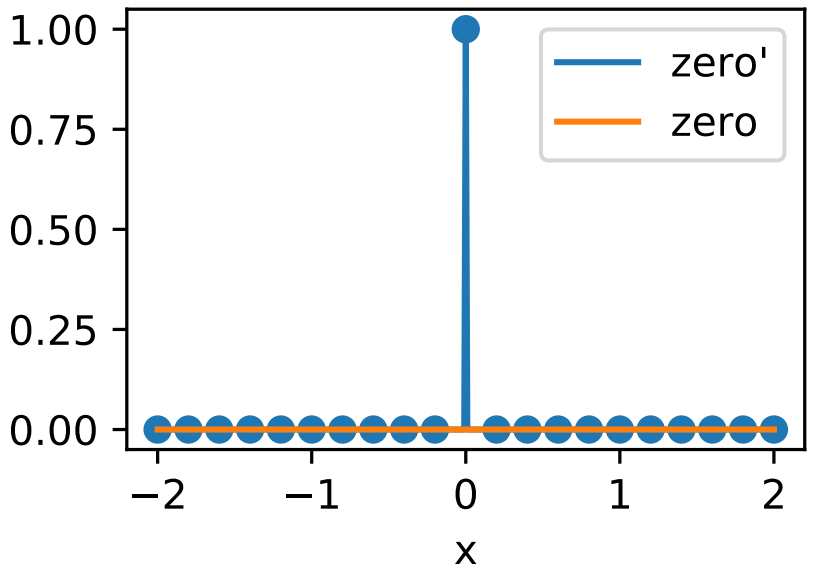




Known from AD litterature (*e.g.* Griewank 2008, Kakade & Lee 2018).

Derivative of zero at 0

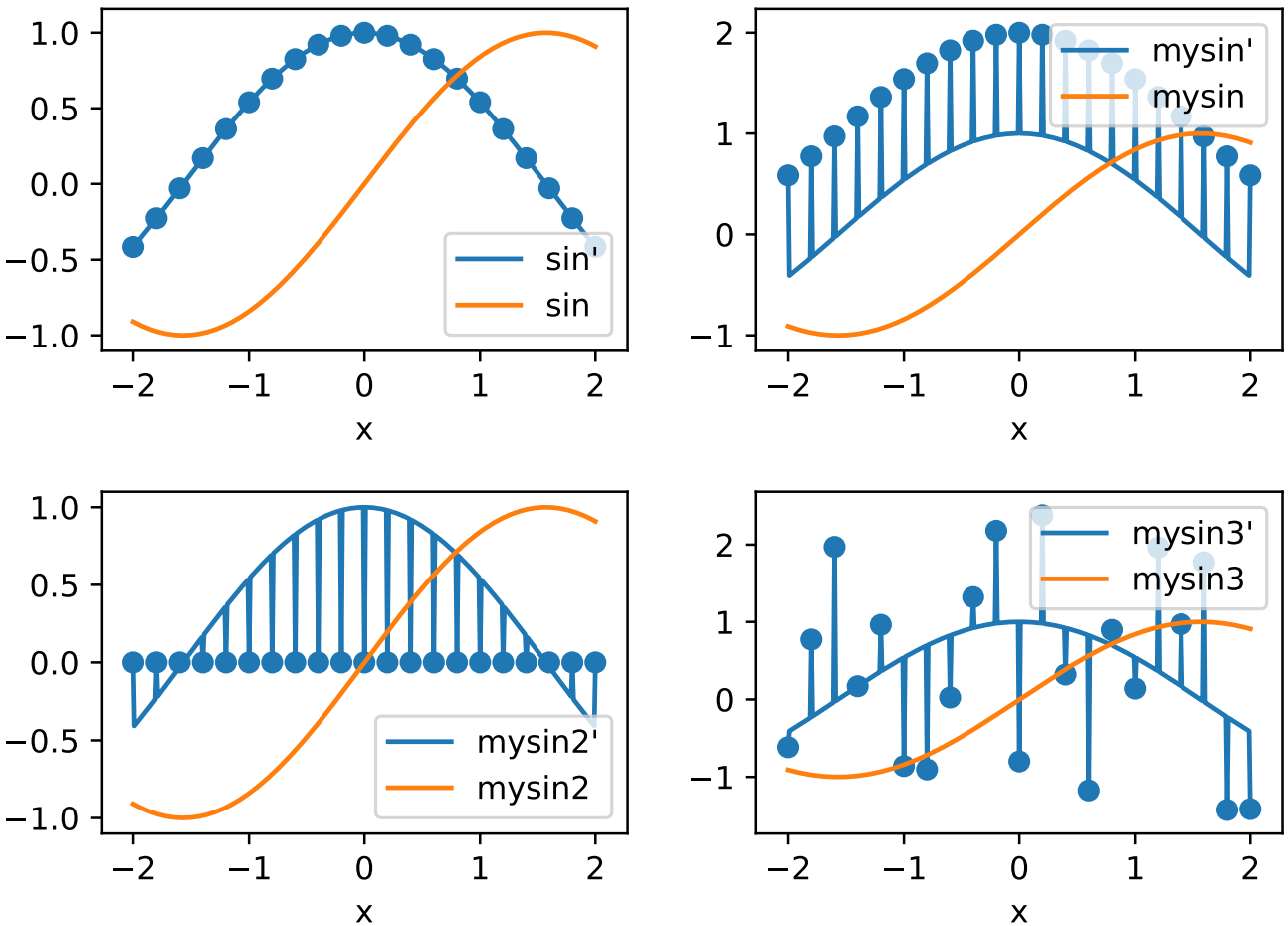
$$\text{zero}(t) = \text{relu2}(t) - \text{relu}(t) = 0.$$



AD acts on programs, not on functions

Derivative of sine at 0:

$$sin' = cos.$$



Consequences for optimization and learning

No convexity, no calculus:

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

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

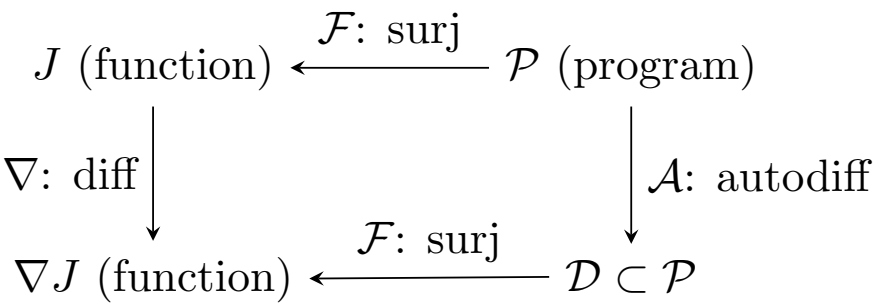
$$\begin{aligned} J(\theta) &= \frac{1}{n}\sum_{i=1}^{n} J_i(\theta) \\ v_i \quad &\in \quad \partial \text{backprop } J_i(\theta) \neq \partial J_i, \quad i = 1, \dots n, \\ \mathbb{E}_I[v_I] \quad &\notin\in \quad \partial J(\theta), \qquad\qquad I \quad \text{uniform on } \{1, \dots, n\}, \end{aligned}$$

Meaning of the chain rule?

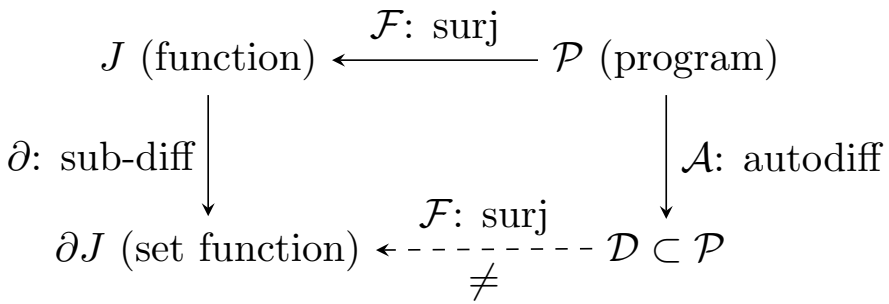
Non uniqueness: Different programs may implement the same function.
No qualification / transverality condition: not a Jacobian of any known sort.

A mathematical model for "nonsmooth algorithmic differentiation"

Smooth:



Nonsmooth:



Outline

1. Conservative set valued field

2. Properties of conservative fields

3. Consequences for deep learning

What is a derivative?

Linear operator:

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

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

An impossibility result

Assume that ∂^A is a subgradient on univariate Lipschitz functions satisfying

- $0 \in \partial^A \mathrm{relu}(0)$.
- sum rule, commutes with translations and multiplication by constants.

Then $\partial^A f(x) = \mathbb{R}$ for any Lipschitz f and any $x \in \mathbb{R}$.

What is a derivative?

Linear operator:

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

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).

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

Technical reminder

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

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

Set valued field: $D : \mathbb{R}^p \rightrightarrows \mathbb{R}^q$ 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

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

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

Closed graph: a notion of continuity for D

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

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

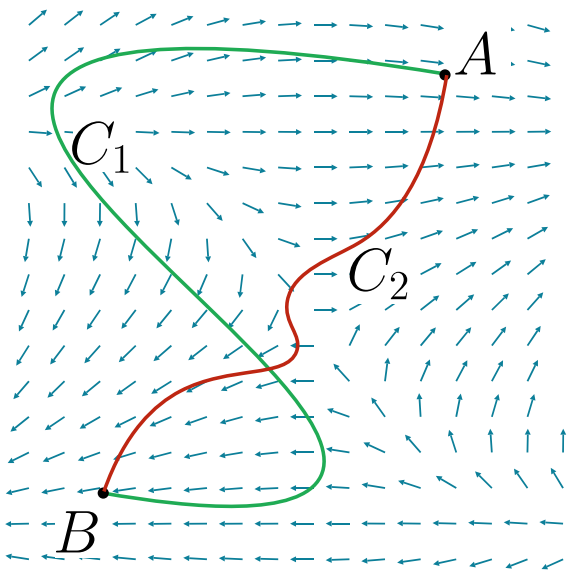
$D\colon \mathbb{R}^p \rightrightarrows \mathbb{R}^p$, set valued, closed graph, non empty values, locally bounded.
Conservative field: For any AC loop $\gamma\colon [0,1] \mapsto \mathbb{R}^p$, $\gamma(0) = \gamma(1)$,

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

Lebsegue integral.

Equivalent forms: set valued (Aumann) integral.

Continuous vector field: null circulation, Poincaré



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.

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

Equivalent forms: With min selection, any measurable selection or set valued integral.

- 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 (they generically don't: Borwein, Moors, Xianfu).

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 $\gamma\colon [0,1] \mapsto \mathbb{R}^p$

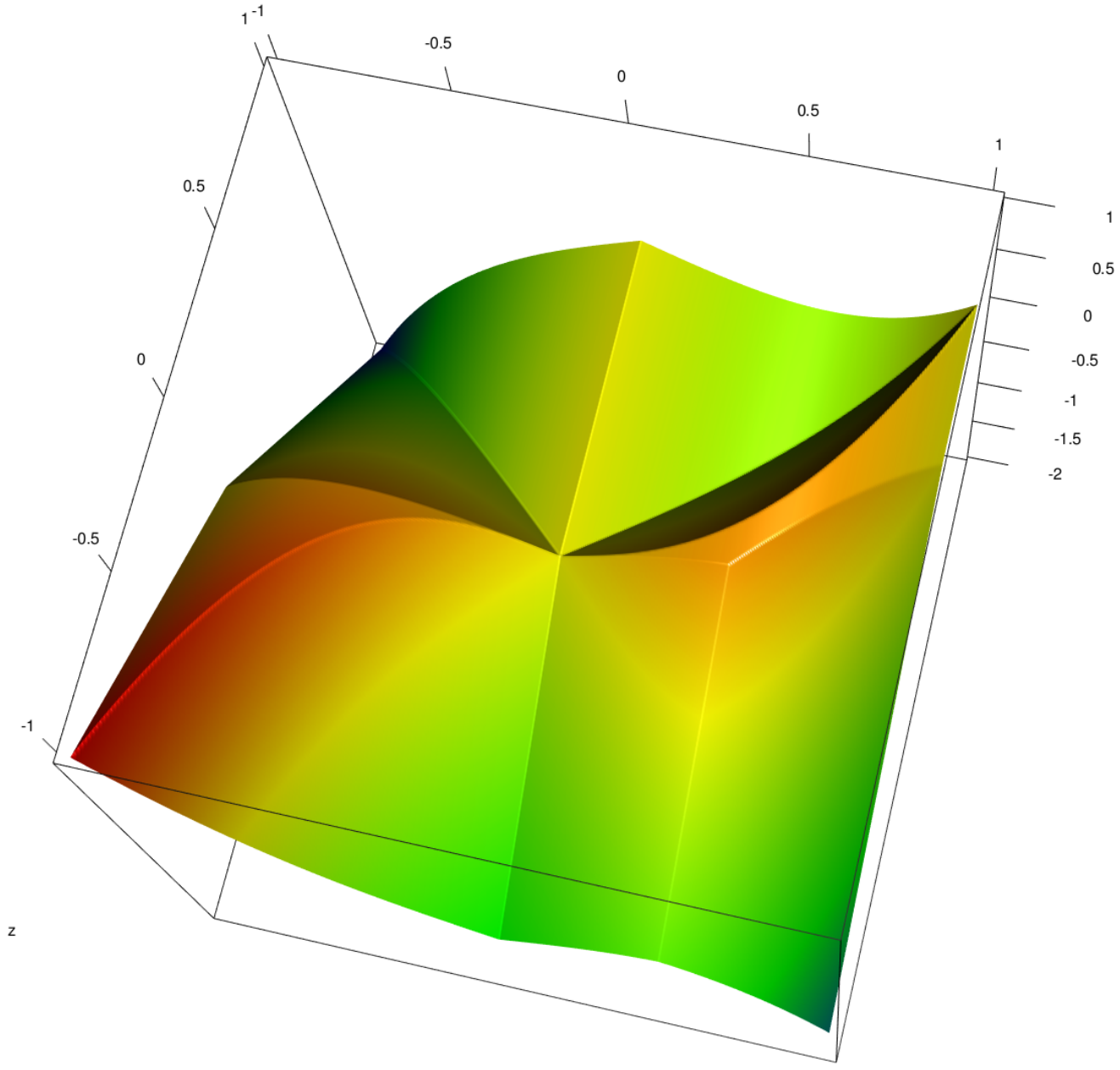
$$\frac{d}{dt} f(\gamma(t)) = \langle v, \dot{\gamma}(t) \rangle \qquad \forall v \in D(\gamma(t)), \qquad \text{a.e.} \quad t \in [0,1].$$

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)].

- Central for Lyapunov analysis of stochastic approximation strategies (minibatch).

Illustration



Outline

1. Conservative set valued field

2. Properties of conservative fields

3. Consequences for deep learning

Relation to gradients

Let $D : \mathbb{R}^p \rightrightarrows \mathbb{R}^p$ be a conservative field for $f : \mathbb{R}^p \mapsto \mathbb{R}$ (implicitely, f admits a conservative field).

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

Compatibility with calculus rules:
Linear combination of conservative fields is conservative
Composition of conservative Jacobian is conservative.

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 $f = \sum_{i=1}^{n} f_i$.

Consequence for AD (informal): A program is a composition of Locally Lipschitz maps.
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

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: Activation functions defining F_θ and ℓ (all nonlinearities) are

- 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 [Dries-Miller 1996].
This structure contains all semi-algebraic sets and the graph of the exponential function [Wilkie (1996)].

Nonsmooth algorithmic differentiation for deep networks

Nonsmooth backpropagation:

- Consider $J\colon \mathbb{R}^p \mapsto \mathbb{R}$ the empirical loss.
- First order mapping,
  $\mathrm{backprop}$ J_i with subgradients in place of derivatives $(\mathrm{relu}'(0) = 0)$.
- Set $D = \frac{1}{n}\sum_{i=1}^{n} \mathrm{backprop}\ J_i$.
- 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.

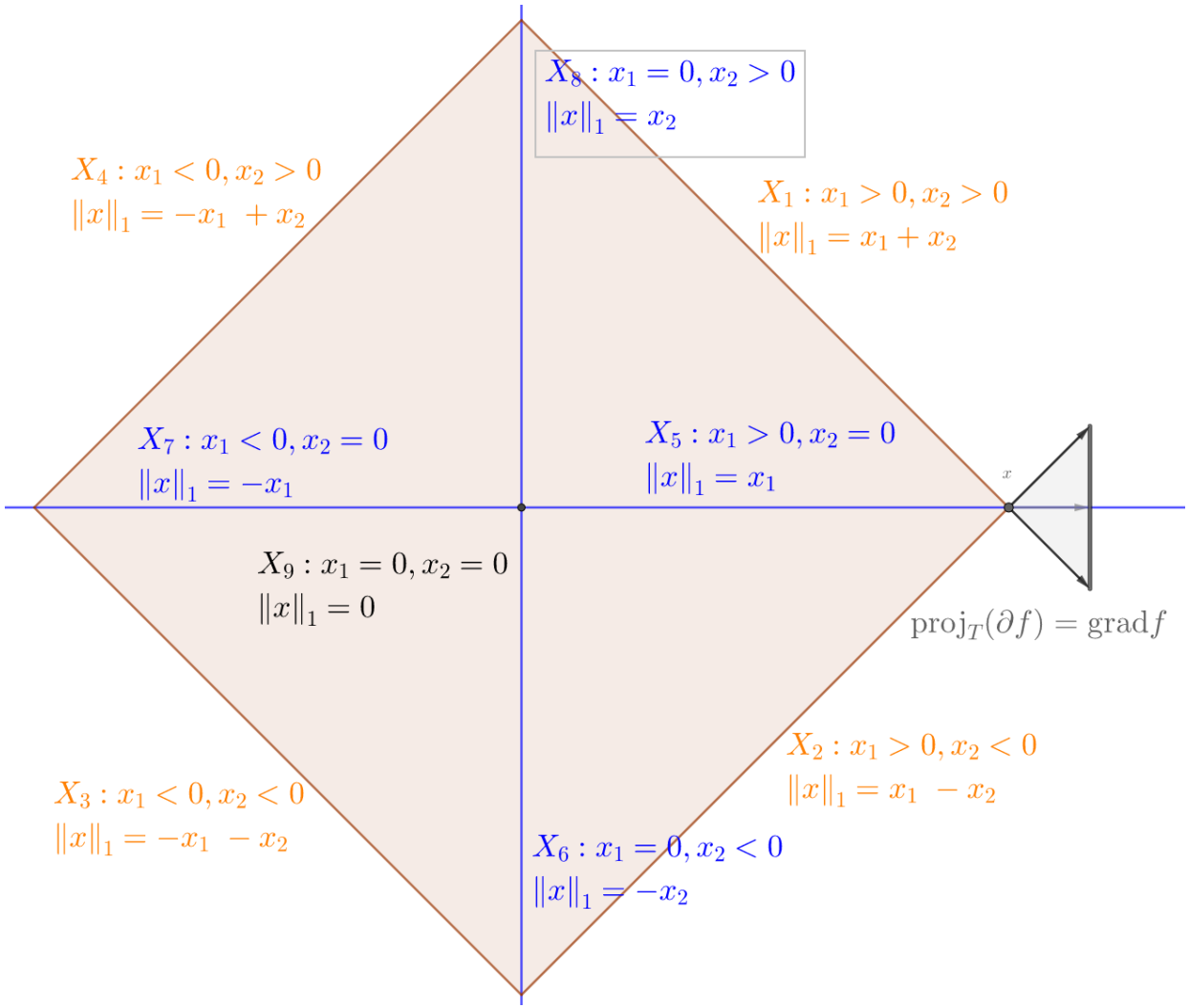
$$J(\mathrm{crit}_J) = \{J(\theta), \quad 0 \in \mathrm{conv}(D(\theta))\}$$

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

Tame characterization: stratification, variational projection

[Bolte-Daniilidis-Lewis (2007)]

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



Minibatch strategies

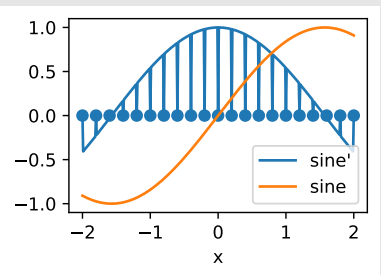
Minibatch stochastic approximation: Given $(I_k)_{k\in\mathbb{N}}$ iid, uniform on $\{1,\dots,n\}$, $(\alpha_k)_{k\in\mathbb{N}}$ positive, iterate,

$$\theta_{k+1} \in \theta_k - \alpha_k \mathrm{backprop}\ J_{I_k}(\theta_k)$$

Convergence: Assume

- $\sum_k \alpha_k = +\infty$ and $\alpha_k = o(1/\log(k))$.
- $\sup_{k\in\mathbb{N}} \|\theta_k\| < +\infty$ with positive probability (call this event E).

Set, $\Theta \subset \mathbb{R}^p$, the set of accumulation points of $(\theta_k)_{k\in\mathbb{N}}$.
Then, almost surely on E, $\emptyset \neq \Theta \subset \mathrm{crit}_J$ and J is constant on Θ.
Spurious critical points:



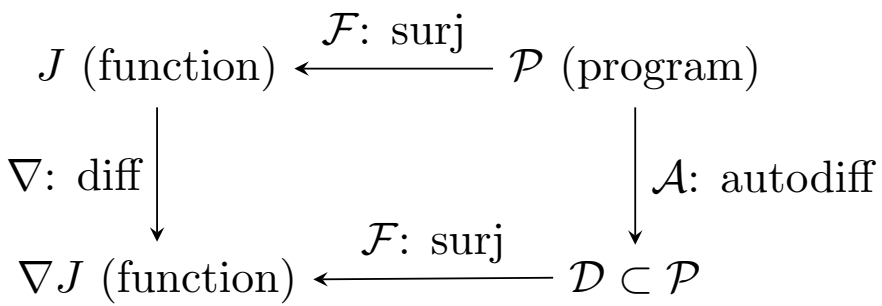
With proper randomization of the initialization, elements of Θ are Clarke critical.

Lyapunov analysis, differential inclusion approach [Benaim-Hofbauer-Sorin (2005)].

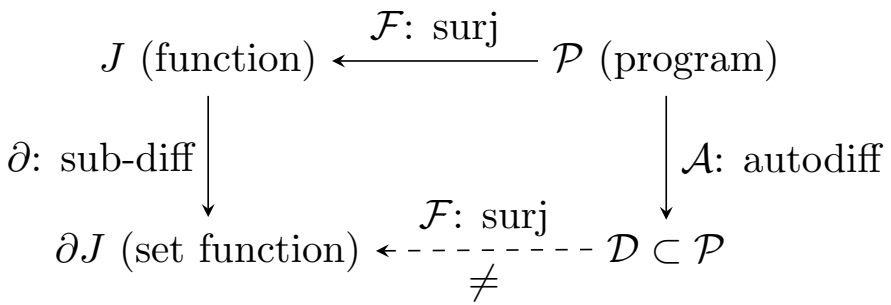
- Conservativity: chain rule along AC curves.
- Tameness: Morse-Sard theorem.

Summary and conclusion: functions, programs and numerics

Smooth:



Nonsmooth:



A mathematical model for nonsmooth algorithmic differentiation.

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


Bolte J., Pauwels E. (2020).
Conservative set valued fields, automatic differentiation, stochastic gradient methods and deep learning.
Mathematical Programming.


Bolte J., Pauwels E. (2020).
A mathematical model for automatic differentiation in machine learning.
Conference on Neural Information Processing systems.

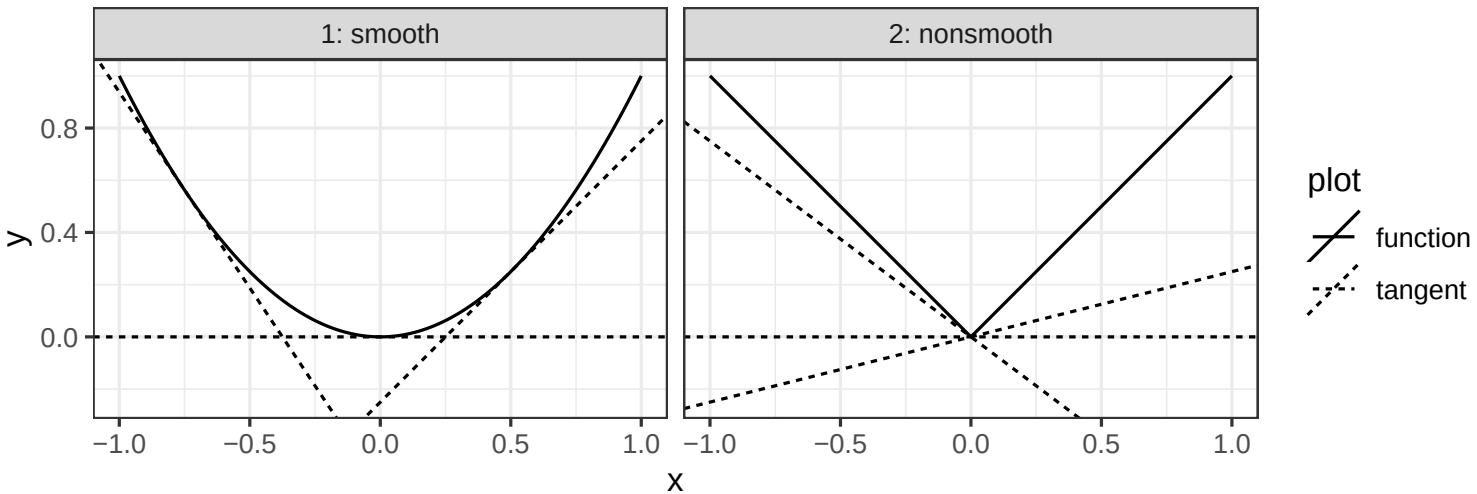

Castera C., Bolte J., Févotte C., Pauwels E. (2019).
An Inertial Newton Algorithm for Deep Learning.
arXiv preprint arXiv:1905.12278.

Subgradients: $F : \mathbb{R}^p \mapsto \mathbb{R}$ Lipschitz continuous

F **convex:** global lower affine tangent

$$F(y) \geq F(x) + \nabla F(x)^T (y - x), \forall y \in \mathbb{R}^p \quad \text{if } F \text{ is differentiable at } x$$

$$\partial_{\text{conv}} F(x) = \left\{ v \in \mathbb{R}^p,\ F(y) \geq F(x) + v^T (y - x),\ \forall y \in \mathbb{R}^p \right\}.$$



Example: $F : x \mapsto |x|$.

$$\partial_{\text{conv}} F(0) = \begin{cases} -1 & \text{if } x < 0 \\ 1 & \text{if } x > 0 \\ [-1, 1] & \text{if } x = 0 \end{cases}.$$

Subgradients: $F : \mathbb{R}^p \mapsto \mathbb{R}$ Lipschitz continuous

F **general:** Rademacher, the set $R \subset \mathbb{R}^p$ where F is differentiable has full measure.

Sequential closure: limits of neighboring gadients.

$$\partial_{\mathrm{cl}} F(x) = \left\{ v \in \mathbb{R}^p,\ \exists \left(y_k, v_k\right)_{k\in\mathbb{N}},\ y_k \underset{k\to\infty}{\to} x,\ v_k \underset{k\to\infty}{\to} v,\ y_k \in R,\ v_k = \nabla F(y_k), k \in \mathbb{N} \right\}.$$

Clarke subgradient: convex closure.

$$\partial_{\mathrm{Clarke}} F(x) = \mathrm{conv}(\partial_{\mathrm{cl}} F(x)).$$

Example: $F : x \mapsto |x|$.

$$\partial_{\mathrm{conv}} F(0) = \begin{cases} -1 & \text{if } x < 0 \\ 1 & \text{if } x > 0 \\ [-1, 1] & \text{if } x = 0 \end{cases}.$$

Subgradient calculus

No convexity, no calculus:

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

- holds with equality if f and g are continuously differentiable.
- holds with equality if f and g are convex (full domain).
- does not hold in general: $f : x \mapsto |x|$

$$\begin{aligned} & \partial(f-f) = \partial(x \mapsto 0) = \{0\} \\ \subset \quad & \partial(f) + \partial(-f) \\ = \quad & \begin{cases} 0 & \text{if } x < 0 \\ 0 & \text{if } x > 0 \\ [-2,2] & \text{if } x = 0 \end{cases}. \end{aligned}$$