COMPSCI 689
Lecture 9: Subgradient Learning for SVMs and the Perceptron

Brendan O'Connor

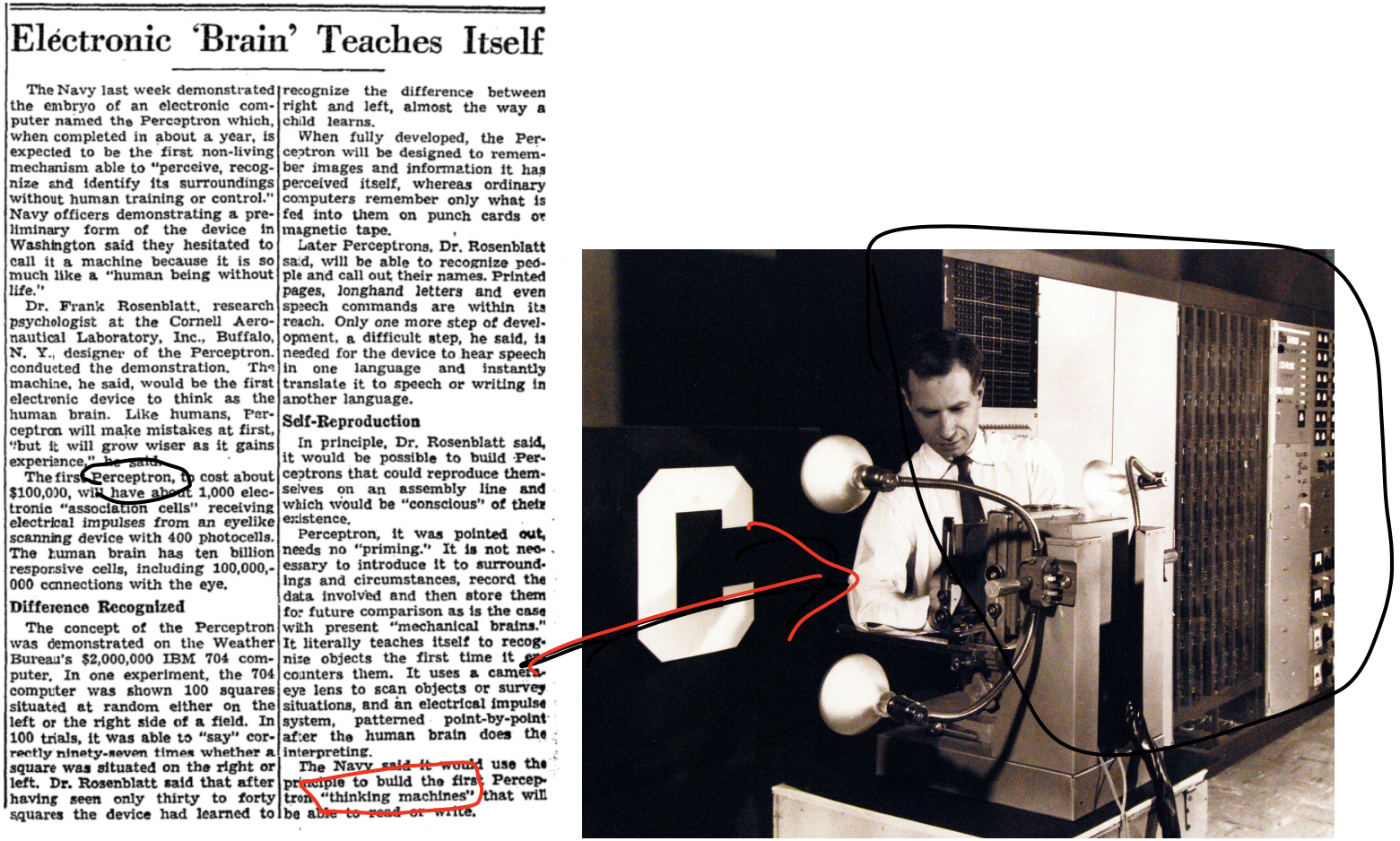
College of Information and Computer Sciences
University of Massachusetts Amherst

Slides by Benjamin M. Marlin (marlin@cs.umass.edu).

Outline

The Perceptron Learning Algorithm

The Perceptron Learning Algorithm



Electronic 'Brain' Teaches Itself

The Navy last week demonstrated the embryo of an electronic computer named the Perceptron which, when completed in about a year, is expected to be the first non-living mechanism able to "perceive, recognize and identify its surroundings without human training or control." Navy officers demonstrating a preliminary form of the device in Washington said they hesitated to call it a machine because it is so much like a "human being without life."

Dr. Frank Rosenblatt, research psychologist at the Cornell Aeronautical Laboratory, Inc., Buffalo, N. Y., designer of the Perceptron, conducted the demonstration. The machine, he said, would be the first electronic device to think as the human brain. Like humans, Perceptron will make mistakes at first, "but it will grow wiser as it gains experience," he said.

The first Perceptron, to cost about $100,000, will have about 1,000 electronic "association cells" receiving electrical impulses from an eyelike scanning device with 400 photocells. The human brain has ten billion responsive cells, including 100,000,000 connections with the eye.

Difference Recognized

The concept of the Perceptron was demonstrated on the Weather Bureau's $2,000,000 IBM 704 computer. In one experiment, the 704 computer was shown 100 squares situated at random either on the left or the right side of a field. In 100 trials, it was able to "say" correctly ninety-seven times whether a square was situated on the right or left. Dr. Rosenblatt said that after having seen only thirty to forty squares the device had learned to recognize the difference between right and left, almost the way a child learns.

When fully developed, the Perceptron will be designed to remember images and information it has perceived itself, whereas ordinary computers remember only what is fed into them on punch cards or magnetic tape.

Later Perceptrons, Dr. Rosenblatt said, will be able to recognize people and call out their names. Printed pages, longhand letters and even speech commands are within its reach. Only one more step of development, a difficult step, he said, is needed for the device to hear speech in one language and instantly translate it to speech or writing in another language.

Self-Reproduction

In principle, Dr. Rosenblatt said, it would be possible to build Perceptrons that could reproduce themselves on an assembly line and which would be "conscious" of their existence.

Perceptron, it was pointed out, needs no "priming." It is not necessary to introduce it to surroundings and circumstances, record the data involved and then store them for future comparison as is the case with present "mechanical brains." It literally teaches itself to recognize objects the first time it encounters them. It uses a camera-eye lens to scan objects or survey situations, and an electrical impulse system, patterned point-by-point after the human brain does the interpreting.

The Navy said it would use the principle to build the first Perceptron "thinking machines" that will be able to read or write.

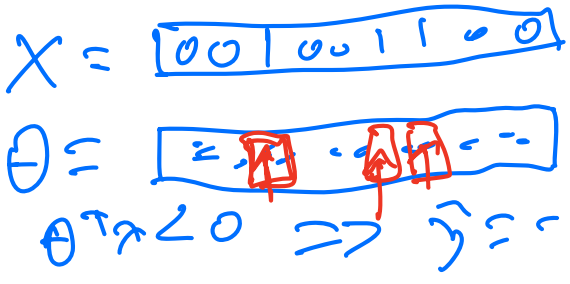
https://timesmachine.nytimes.com/timesmachine/1958/07/13/91396361.html?pageNumber=116

The Perceptron Learning Algorithm

Rosenblatt, 1957

$f(x; \theta) = \text{sign}(\theta^T x)$

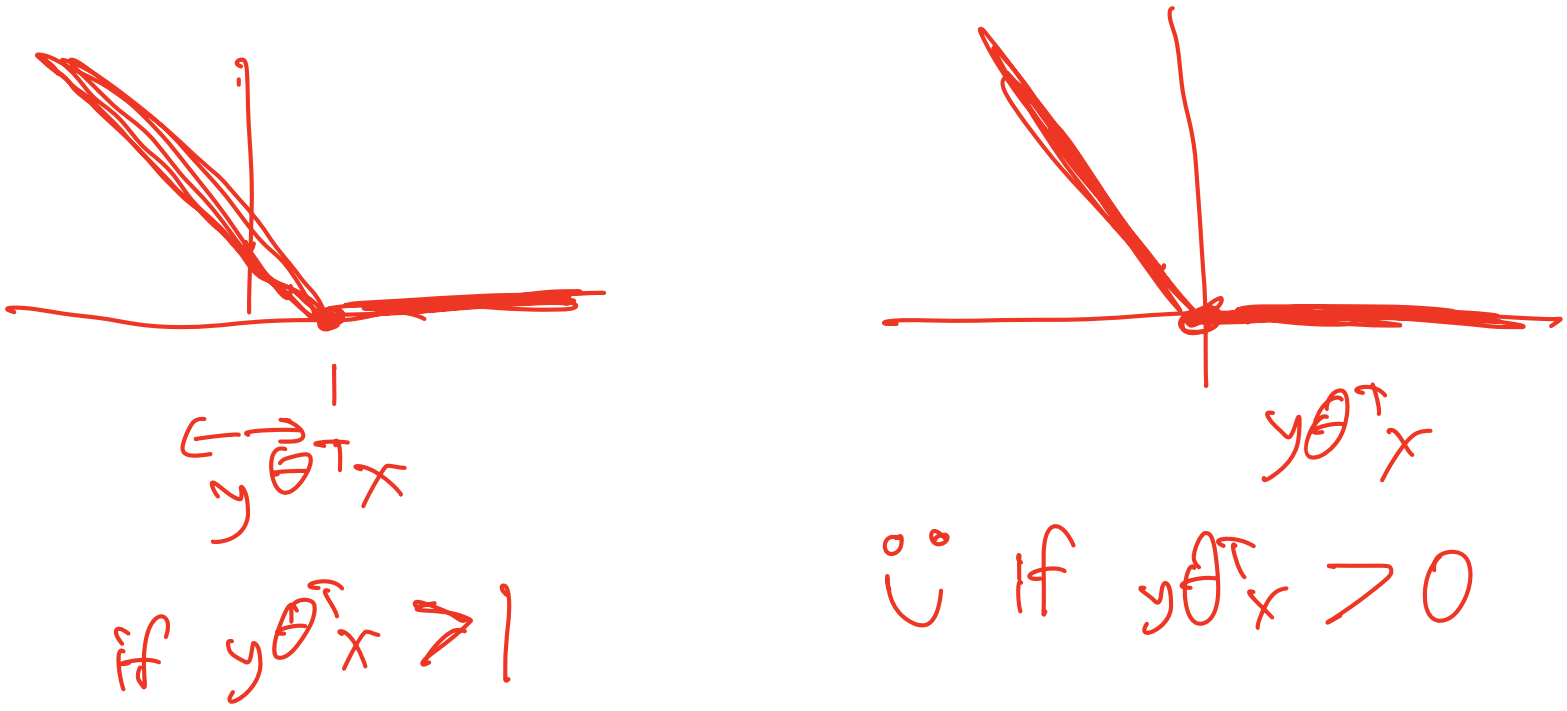
Outer loop (~20 iter):
  for (x_i, y_i) in Dtr:
    $\hat{y} = f(x_i; \theta)$
    if $\hat{y} \neq y_i$:
      $\theta = \theta + y_i x_i$



Convergence: ☹ (in general)

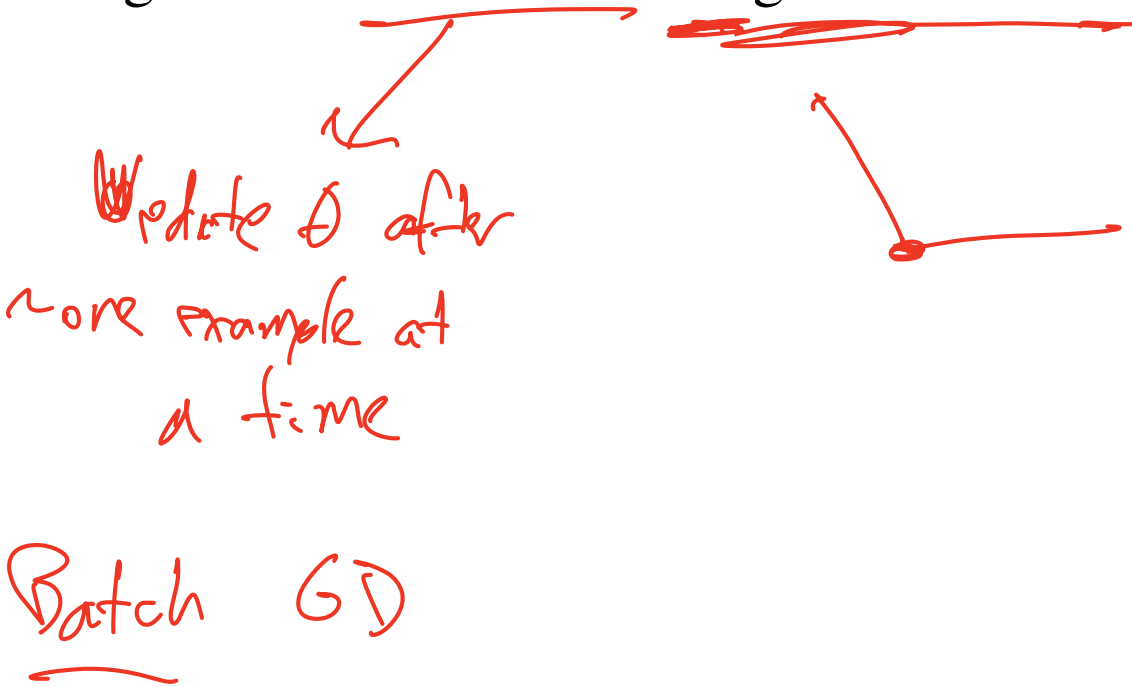
The Perceptron Learning Algorithm

- We view the classic perceptron as an ERM training method for a linear classifier, with the *perceptron loss*, very similar to the SVM hinge loss.



The Perceptron Learning Algorithm

- We view the classic perceptron as an ERM training method for a linear classifier, with the *perceptron loss*, very similar to the SVM hinge loss.
- The algorithm is stochastic subgradient descent for this loss.



Kernelization and Stochastic SubGD

- This isn't the optimal way to learn a kernelized classifier, but it's extremely easy to kernelize/dualize the perceptron.

$i = 1 \ldots N,\ N+1 \ldots 2N, \ldots\ldots NN$

$\theta_i = \theta_{i-1} + y_i x_i 1\{\hat{y}_i \neq y_i\}$

$\theta_i = \sum_{j=1}^{i} y_j x_j 1\{\hat{y}_j \neq y_j\}$

Assume $\theta_0 = \vec{0}$

Warm Start: $\theta_0 \neq \vec{0}$

$f(x^{new}; \theta_i) = \theta_i^T x^{new} = \left(\sum_j y_j x_j 1\{\hat{y}_j = y_j\}\right)^T x^{new}$

$= \sum_j y_j 1\{\hat{y}_j = y_j\}\ x_j^T x^{new}$

$y_j 1\{\hat{y}_j = y_j\} \in \{-1, 0, +1\}$

$x_j^T x^{new} = K(x_j, x^{new})$

Outline

Back to the Hinge Loss Formulation

$$f_\theta(\mathbf{x}) = \text{sign}(g_\theta(\mathbf{x}))$$

$$g_\theta(\mathbf{x}) = \mathbf{x}\theta = \mathbf{x}\mathbf{w} + b$$

$$\hat{\theta} = \arg\min_\theta C \sum_{n=1}^{N} \max(0, 1 - y_n g_\theta(\mathbf{x}_n)) + ||\mathbf{w}||_2^2$$

Global Optimality of Convex Non-Differentiable Functions

- Our existing optimization theory only holds for the case of differentiable functions.

[1] This section mostly follows the presentation in Boyd's EE364b - Convex Optimization II course.

Global Optimality of Convex Non-Differentiable Functions

- Our existing optimization theory only holds for the case of differentiable functions.
- However, it turns out that many results generalize to the case of non-differentiable functions that are convex and we can use them to directly minimize the hinge loss.

[1] This section mostly follows the presentation in Boyd's EE364b - Convex Optimization II course.

Global Optimality of Convex Non-Differentiable Functions

- Our existing optimization theory only holds for the case of differentiable functions.
- However, it turns out that many results generalize to the case of non-differentiable functions that are convex and we can use them to directly minimize the hinge loss.
- To begin, strongly convex non-differentiable functions have a unique global minima, exactly as with convex differentiable functions.

[1] This section mostly follows the presentation in Boyd's EE364b - Convex Optimization II course.

Global Optimality of Convex Non-Differentiable Functions

- Our existing optimization theory only holds for the case of differentiable functions.
- However, it turns out that many results generalize to the case of non-differentiable functions that are convex and we can use them to directly minimize the hinge loss.
- To begin, strongly convex non-differentiable functions have a unique global minima, exactly as with convex differentiable functions.
- We will begin with the characterization of the minimizer of a non-differentiable convex function. [1]

[1] This section mostly follows the presentation in Boyd's EE364b - Convex Optimization II course.

Subgradient

- Let $f : \mathbb{R}^D \to \mathbb{R}$. A vector $\mathbf{g} \in \mathbb{R}^D$ is said to be a sub-gradient of f at a point $\mathbf{x}_o \in \mathbb{R}^D$ if for all $\mathbf{x} \in \mathbb{R}^D$:

$$f(\mathbf{x}) \geq f(\mathbf{x}_o) + \mathbf{g}^T \cdot (\mathbf{x} - \mathbf{x}_o)$$

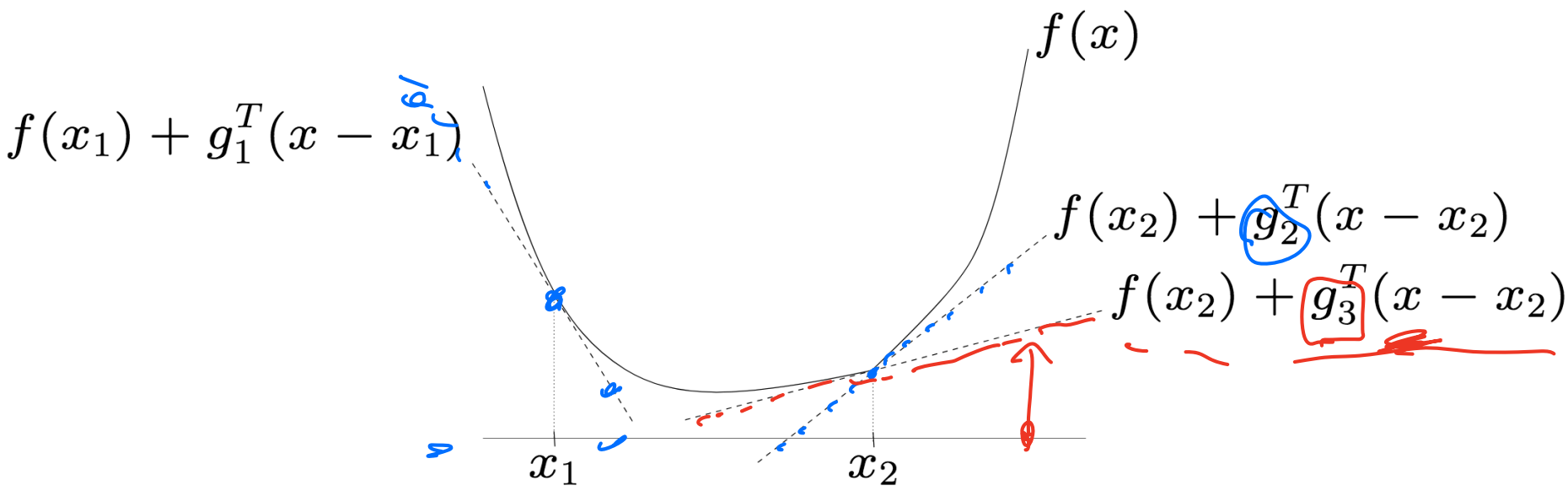
Subgradient

- Let $f : \mathbb{R}^D \to \mathbb{R}$. A vector $\mathbf{g} \in \mathbb{R}^D$ is said to be a sub-gradient of f at a point $\mathbf{x}_o \in \mathbb{R}^D$ if for all $\mathbf{x} \in \mathbb{R}^D$:

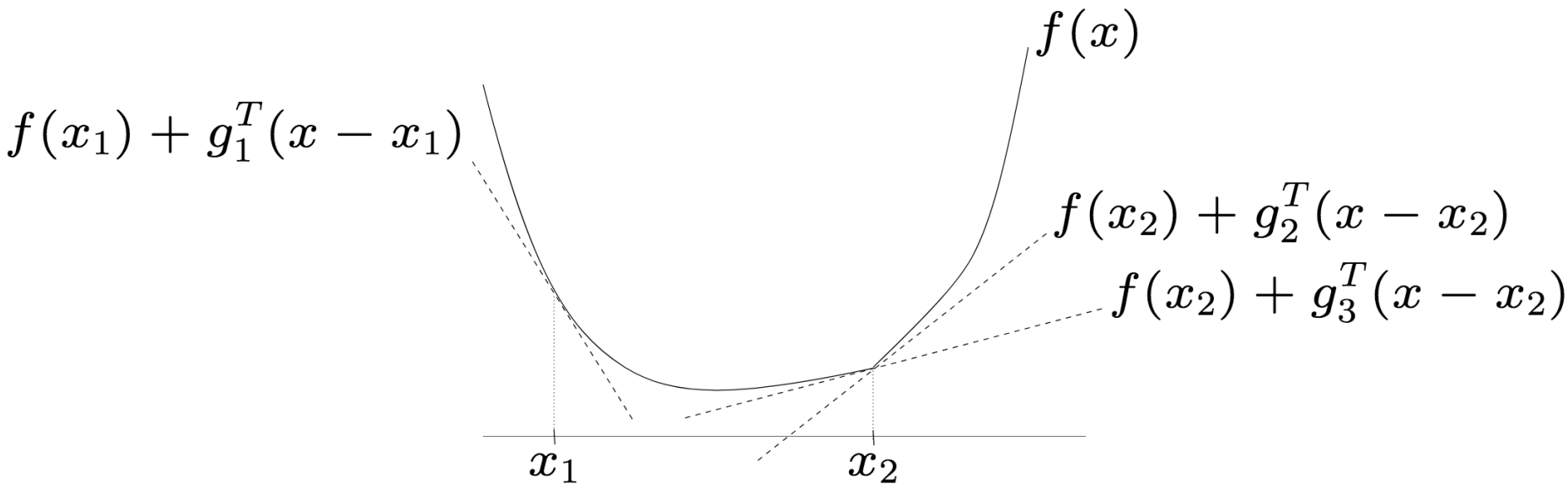
$$f(\mathbf{x}) \geq f(\mathbf{x}_o) + \mathbf{g}^T \cdot (\mathbf{x} - \mathbf{x}_o)$$

- That is to say, the hyperplane defined by $h(\mathbf{x}) = f(\mathbf{x}_o) + \mathbf{g}^T \cdot (\mathbf{x} - \mathbf{x}_o)$ lies at or below below f everywhere and touches f at $\mathbf{x}_o$.

Example: Subgradients



Example: Subgradients



In this example, $\mathbf{g}_1$ is the unique subgradient of f at $\mathbf{x}_1$. Due to f being non-differentiable at $\mathbf{x}_2$, both $\mathbf{g}_2$ and $\mathbf{g}_3$ are subgradients of f at $\mathbf{x}_2$.

Subdifferentials

- If f is convex and is differentiable at $\mathbf{x}_o$, then $\nabla f(\mathbf{x}_o)$ is its unique subgradient at $\mathbf{x}_o$.

Subdifferentials

- If f is convex and is differentiable at $\mathbf{x}_o$, then $\nabla f(\mathbf{x}_o)$ is its unique subgradient at $\mathbf{x}_o$.
- If f is convex and non-differentiable at $\mathbf{x}_o$, it will generally have more than one vector $\mathbf{g}$ satisfying the subgradient property.

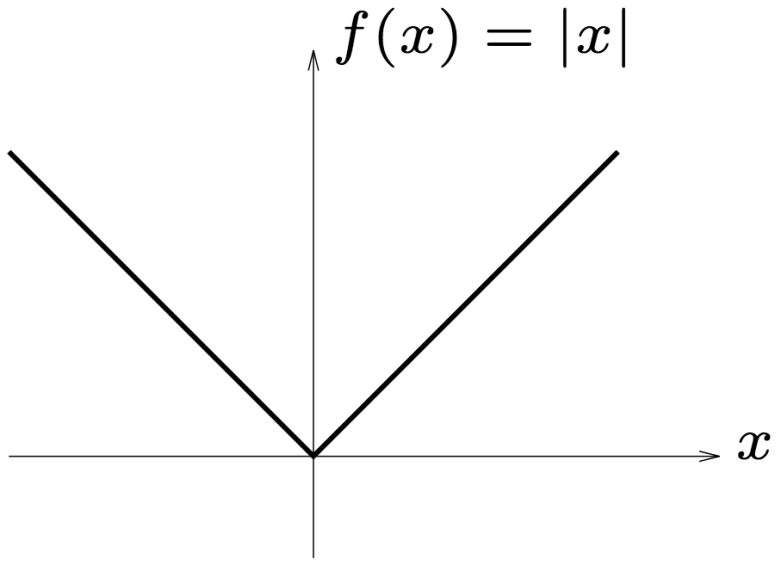
Subdifferentials

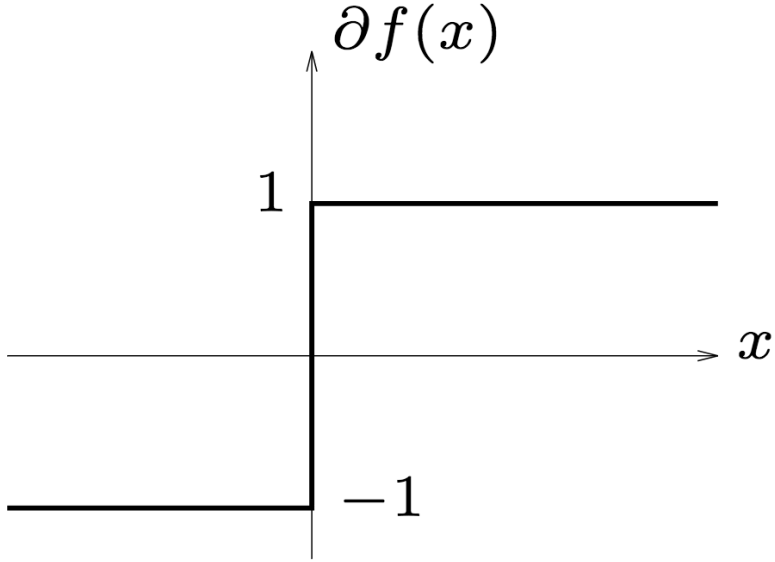
- If f is convex and is differentiable at $\mathbf{x}_o$, then $\nabla f(\mathbf{x}_o)$ is its unique subgradient at $\mathbf{x}_o$.
- If f is convex and non-differentiable at $\mathbf{x}_o$, it will generally have more that one vector $\mathbf{g}$ satisfying the subgradient property.
- The set of all subgradients of f at $\mathbf{x}_o$ is called the subdifferential of f at $\mathbf{x}_o$ denoted by $\partial f(\mathbf{x}_o)$.

Subdifferentials

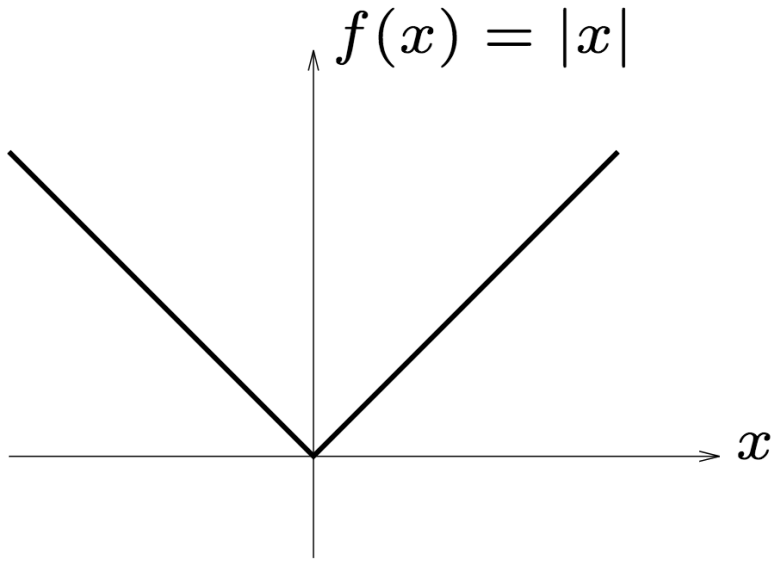
- If f is convex and is differentiable at $\mathbf{x}_o$, then $\nabla f(\mathbf{x}_o)$ is its unique subgradient at $\mathbf{x}_o$.
- If f is convex and non-differentiable at $\mathbf{x}_o$, it will generally have more that one vector $\mathbf{g}$ satisfying the subgradient property.
- The set of all subgradients of f at $\mathbf{x}_o$ is called the subdifferential of f at $\mathbf{x}_o$ denoted by $\partial f(\mathbf{x}_o)$.
- $\partial f(\mathbf{x}_o)$ is a closed, convex set in $\mathbb{R}^D$. If f is convex, $\partial f(\mathbf{x}_o)$ is always non-empty.

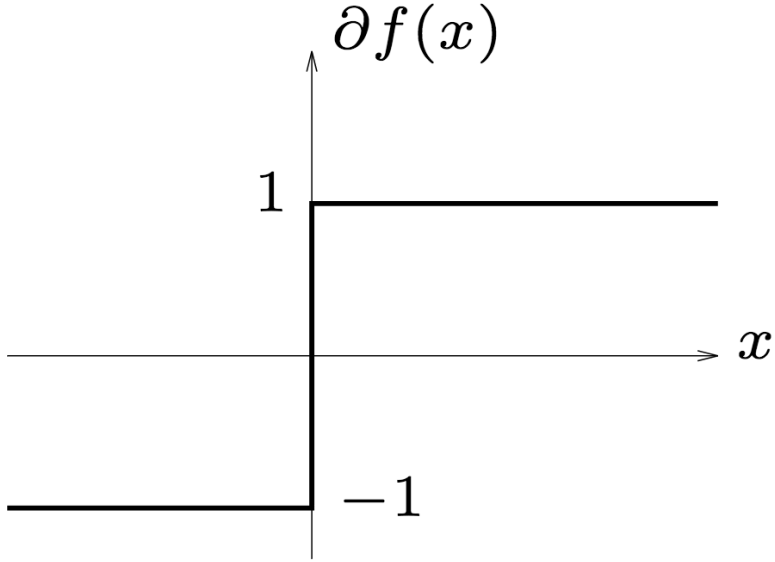
Example: Subdifferentials





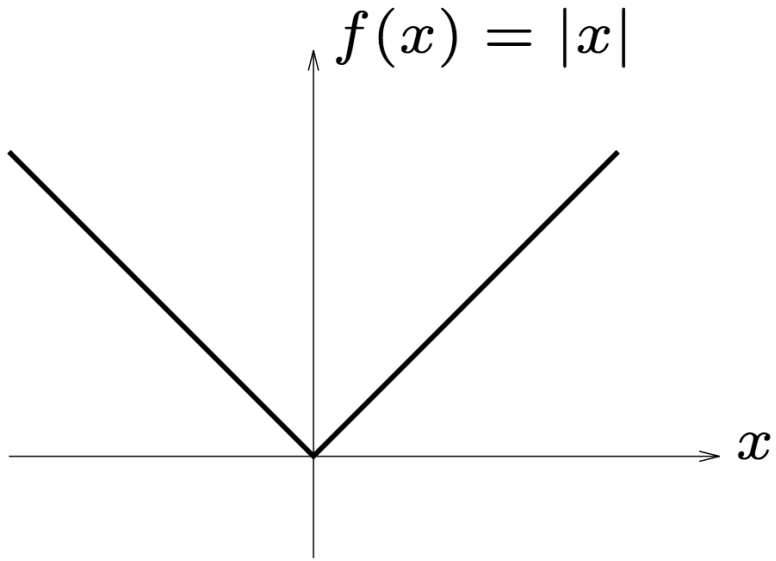
Example: Subdifferentials

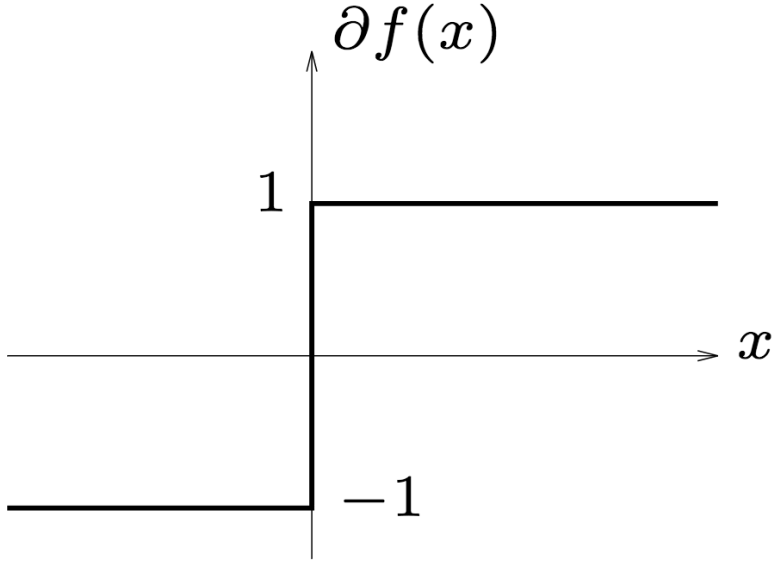




The righthand plot shows $\partial f(x)$ for $f(x) = |x|$.

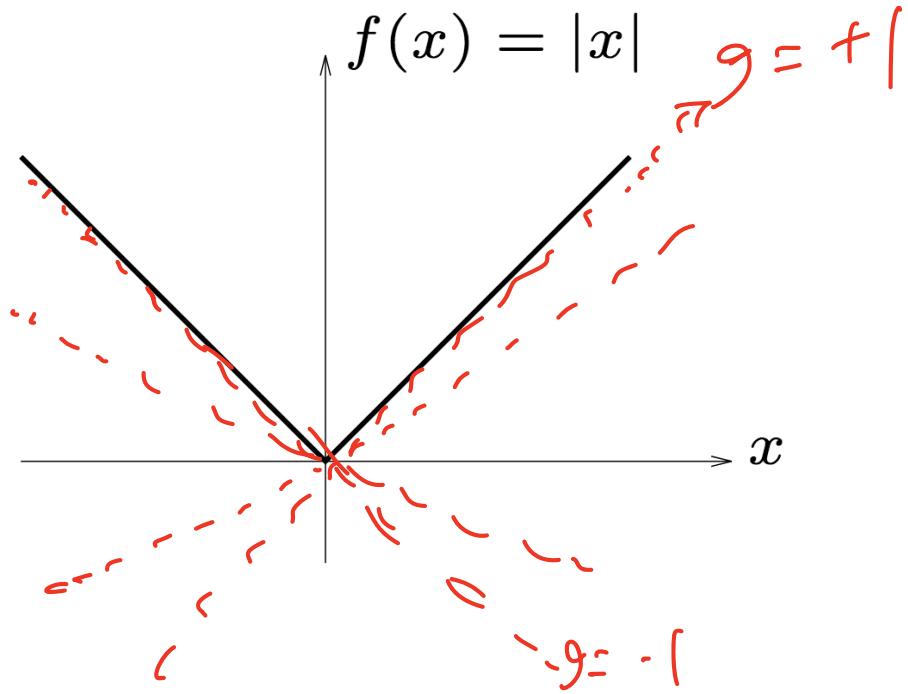
Example: Subdifferentials

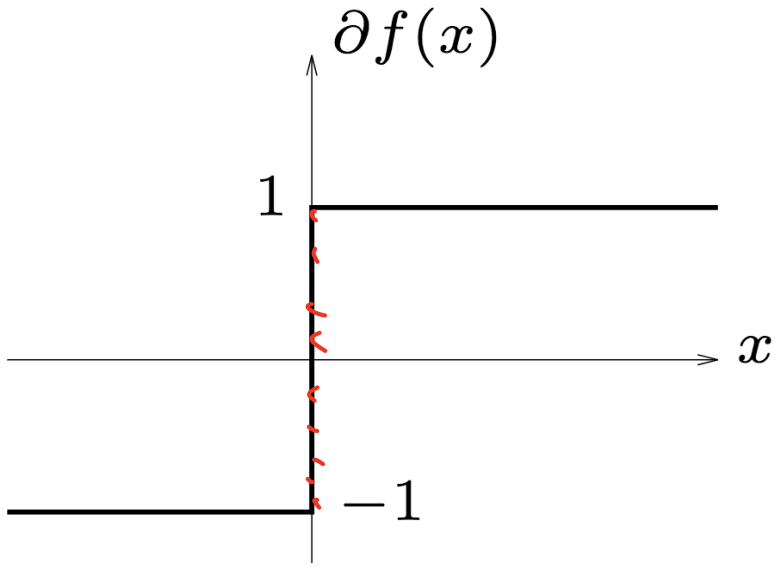




The righthand plot shows $\partial f(x)$ for $f(x) = |x|$.
We have $\partial f(\mathbf{x}) = \{\text{sign}(x)\}$ for $x \neq 0$.

Example: Subdifferentials





The righthand plot shows $\partial f(x)$ for $f(x) = |x|$.
We have $\partial f(\mathbf{x}) = \{\text{sign}(x)\}$ for $x \neq 0$.
When $x = 0$, the line $|0| + g \cdot (x - 0) = g \cdot x$ will lie below f everywhere only if $g \in [-1, 1]$.

Characterizing the Global Minimum

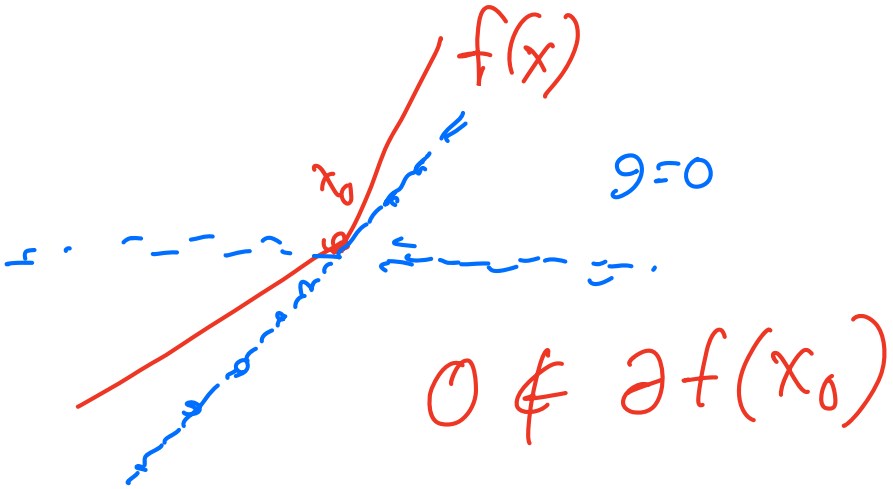
- Let $f : \mathbb{R}^D \to \mathbb{R}$ be a convex function.

Characterizing the Global Minimum

- Let $f : \mathbb{R}^D \to \mathbb{R}$ be a convex function.
- $\mathbf{x}_*$ is the global minimizer of f if and only if $\mathbf{0} \in \partial f(\mathbf{x}_*)$.

Characterizing the Global Minimum

- Let $f : \mathbb{R}^D \to \mathbb{R}$ be a convex function.
- $\mathbf{x}_*$ is the global minimizer of f if and only if $\mathbf{0} \in \partial f(\mathbf{x}_*)$.
- This is a generalization of the idea of a stationary point to include the case of non-differentiable functions.



Finding Subdifferentials

- Suppose that x_0 is a point of non-differentiability for a 1-dimensional convex function $f(x)$.

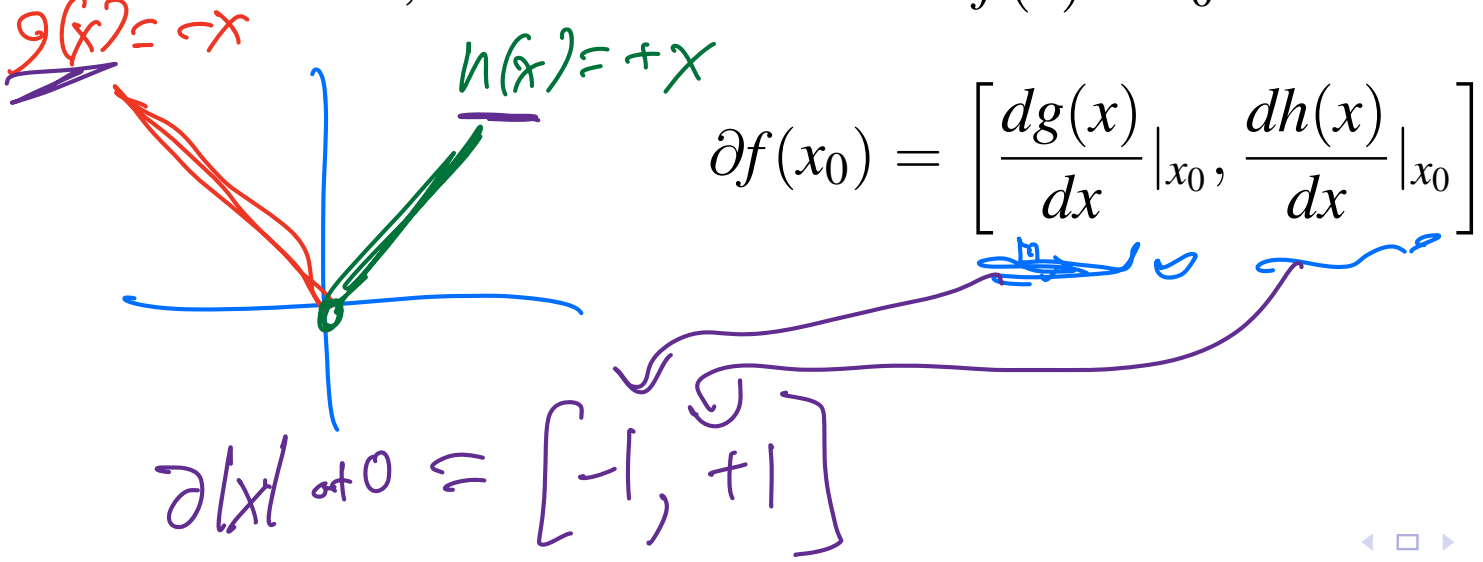
Finding Subdifferentials

- Suppose that x_0 is a point of non-differentiability for a 1-dimensional convex function $f(x)$.
- Suppose that in the neighborhood $[a, x_0]$, for some $a < x_0$, the value of $f(x)$ is given by a differentiable function $g(x)$ and in the neighborhood $[x_0, b]$ for some $b > x_0$, the value of $f(x)$ is given by a differentiable function $h(x)$.

Finding Subdifferentials

- Suppose that x_0 is a point of non-differentiability for a 1-dimensional convex function $f(x)$.
- Suppose that in the neighborhood $[a, x_0]$, for some $a < x_0$, the value of $f(x)$ is given by a differentiable function $g(x)$ and in the neighborhood $[x_0, b]$ for some $b > x_0$, the value of $f(x)$ is given by a differentiable function $h(x)$.
- Then, the subdifferential of $f(x)$ at x_0 is:

$$\partial f(x_0) = \left[\frac{dg(x)}{dx}\Big|_{x_0}, \frac{dh(x)}{dx}\Big|_{x_0} \right]$$



Partial Linearity and Chain Rule

The subdifferential operator satisfies partial linearity and chain rule properties:

- **Scaling:** If $f(\mathbf{x})$ is a convex function and $\alpha > 0$, then if $\mathbf{g} \in \partial f(\mathbf{x})$, $\alpha\mathbf{g} \in \partial(\alpha f(\mathbf{x}))$

Partial Linearity and Chain Rule

The subdifferential operator satisfies partial linearity and chain rule properties:

- **Scaling:** If $f(\mathbf{x})$ is a convex function and $\alpha > 0$, then if $\mathbf{g} \in \partial f(\mathbf{x})$, $\alpha\mathbf{g} \in \partial(\alpha f(\mathbf{x}))$
- **Addition:** If $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are convex functions and $\mathbf{g}_1 \in \partial f_1(\mathbf{x})$ and $\mathbf{g}_2 \in \partial f_2(\mathbf{x})$, then $\mathbf{g}_1 + \mathbf{g}_2 \in \partial(f_1(\mathbf{x}) + f_2(\mathbf{x}))$

Partial Linearity and Chain Rule

The subdifferential operator satisfies partial linearity and chain rule properties:

- **Scaling:** If $f(\mathbf{x})$ is a convex function and $\alpha > 0$, then if $\mathbf{g} \in \partial f(\mathbf{x})$, $\alpha\mathbf{g} \in \partial(\alpha f(\mathbf{x}))$
- **Addition:** If $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are convex functions and $\mathbf{g}_1 \in \partial f_1(\mathbf{x})$ and $\mathbf{g}_2 \in \partial f_2(\mathbf{x})$, then $\mathbf{g}_1 + \mathbf{g}_2 \in \partial(f_1(\mathbf{x}) + f_2(\mathbf{x}))$
- **Chain Rule**: If $f : \mathbb{R} \to \mathbb{R}$ is a convex function and $h : \mathbb{R}^D \to \mathbb{R}$ is a linear function $h(\mathbf{x}) = \mathbf{x}\mathbf{a} + b$, then the sub-differential of $f(h(\mathbf{x}))$ is $\{g\mathbf{a}^T | g \in \partial f(y), y = h(\mathbf{x})\}$.

Partial Linearity and Chain Rule

The subdifferential operator satisfies partial linearity and chain rule properties:

- **Scaling:** If $f(\mathbf{x})$ is a convex function and $\alpha > 0$, then if $\mathbf{g} \in \partial f(\mathbf{x})$, $\alpha\mathbf{g} \in \partial(\alpha f(\mathbf{x}))$
- **Addition:** If $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are convex functions and $\mathbf{g}_1 \in \partial f_1(\mathbf{x})$ and $\mathbf{g}_2 \in \partial f_2(\mathbf{x})$, then $\mathbf{g}_1 + \mathbf{g}_2 \in \partial(f_1(\mathbf{x}) + f_2(\mathbf{x}))$
- **Chain Rule**: If $f : \mathbb{R} \to \mathbb{R}$ is a convex function and $h : \mathbb{R}^D \to \mathbb{R}$ is a linear function $h(\mathbf{x}) = \mathbf{x}\mathbf{a} + b$, then the sub-differential of $f(h(\mathbf{x}))$ is $\{g\mathbf{a}^T | g \in \partial f(y), y = h(\mathbf{x})\}$.

Partial Linearity and Chain Rule

The subdifferential operator satisfies partial linearity and chain rule properties:

- **Scaling:** If $f(\mathbf{x})$ is a convex function and $\alpha > 0$, then if $\mathbf{g} \in \partial f(\mathbf{x})$, $\alpha\mathbf{g} \in \partial(\alpha f(\mathbf{x}))$
- **Addition:** If $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are convex functions and $\mathbf{g}_1 \in \partial f_1(\mathbf{x})$ and $\mathbf{g}_2 \in \partial f_2(\mathbf{x})$, then $\mathbf{g}_1 + \mathbf{g}_2 \in \partial(f_1(\mathbf{x}) + f_2(\mathbf{x}))$
- **Chain Rule**: If $f : \mathbb{R} \to \mathbb{R}$ is a convex function and $h : \mathbb{R}^D \to \mathbb{R}$ is a linear function $h(\mathbf{x}) = \mathbf{x}\mathbf{a} + b$, then the sub-differential of $f(h(\mathbf{x}))$ is $\{g\mathbf{a}^T | g \in \partial f(y), y = h(\mathbf{x})\}$.

These properties can be used to determine the subdifferentials of more some complex functions if they reduce to certain combinations or compositions of simpler functions.

Outline

Sub-gradient Descent

- If f is differentiable at $\mathbf{x}$ then $\partial f(\mathbf{x}) = \{\nabla f(\mathbf{x})\}$ and $-\mathbf{g} = -\nabla f(\mathbf{x})$ is a descent direction unless $\nabla f(\mathbf{x}) = 0$.

Sub-gradient Descent

- If f is differentiable at $\mathbf{x}$ then $\partial f(\mathbf{x}) = \{\nabla f(\mathbf{x})\}$ and $-\mathbf{g} = -\nabla f(\mathbf{x})$ is a descent direction unless $\nabla f(\mathbf{x}) = 0$.
- If f is not differentiable at $\mathbf{x}$, and $\mathbf{0} \notin \partial f(\mathbf{x})$, then there exists at least one $\mathbf{g} \in \partial f(\mathbf{x})$ where $-\mathbf{g}$ is a descent direction.

Sub-gradient Descent

- If f is differentiable at $\mathbf{x}$ then $\partial f(\mathbf{x}) = \{\nabla f(\mathbf{x})\}$ and $-\mathbf{g} = -\nabla f(\mathbf{x})$ is a descent direction unless $\nabla f(\mathbf{x}) = 0$.
- If f is not differentiable at $\mathbf{x}$, and $\mathbf{0} \notin \partial f(\mathbf{x})$, then there exists at least one $\mathbf{g} \in \partial f(\mathbf{x})$ where $-\mathbf{g}$ is a descent direction.
- If f is not differentiable at $\mathbf{x}$, then there may exist some $\mathbf{g} \in \partial f(\mathbf{x})$ where $-\mathbf{g}$ is a **not** descent direction.

Sub-gradient Descent

Despite the fact that not all sub-gradients yield descent directions, it is still possible to minimize a convex non-differentiable function using a fairly basic sub-gradient descent procedure:

Sub-gradient Descent

Despite the fact that not all sub-gradients yield descent directions, it is still possible to minimize a convex non-differentiable function using a fairly basic sub-gradient descent procedure:

- Initialize $\mathbf{x}_0 \in \mathbb{R}^D, f_{min} = \infty, \mathbf{x}_* = 0$

Sub-gradient Descent

Despite the fact that not all sub-gradients yield descent directions, it is still possible to minimize a convex non-differentiable function using a fairly basic sub-gradient descent procedure:

- Initialize $\mathbf{x}_0 \in \mathbb{R}^D, f_{min} = \infty, \mathbf{x}_* = 0$
- For k from 1 to K:

Sub-gradient Descent

Despite the fact that not all sub-gradients yield descent directions, it is still possible to minimize a convex non-differentiable function using a fairly basic sub-gradient descent procedure:

- Initialize $\mathbf{x}_0 \in \mathbb{R}^D, f_{min} = \infty, \mathbf{x}_* = 0$
- For k from 1 to K:
  - Let $\mathbf{g}_k \in \partial f(\mathbf{x}_{k-1})$

Sub-gradient Descent

Despite the fact that not all sub-gradients yield descent directions, it is still possible to minimize a convex non-differentiable function using a fairly basic sub-gradient descent procedure:

- Initialize $\mathbf{x}_0 \in \mathbb{R}^D, f_{min} = \infty, \mathbf{x}_* = 0$
- For k from 1 to K:
  - Let $\mathbf{g}_k \in \partial f(\mathbf{x}_{k-1})$
  - Set $\mathbf{x}_k = \mathbf{x}_{k-1} - \alpha_k \mathbf{g}_k$

Sub-gradient Descent

Despite the fact that not all sub-gradients yield descent directions, it is still possible to minimize a convex non-differentiable function using a fairly basic sub-gradient descent procedure:

- Initialize $\mathbf{x}_0 \in \mathbb{R}^D, f_{min} = \infty, \mathbf{x}_* = 0$
- For k from 1 to K:
  - Let $\mathbf{g}_k \in \partial f(\mathbf{x}_{k-1})$
  - Set $\mathbf{x}_k = \mathbf{x}_{k-1} - \alpha_k \mathbf{g}_k$
  - If $f(\mathbf{x}_k) < f_{min}$ then set $f_{min} = f(\mathbf{x}_k)$ and $\mathbf{x}_* = \mathbf{x}_k$

Sub-gradient Descent

Despite the fact that not all sub-gradients yield descent directions, it is still possible to minimize a convex non-differentiable function using a fairly basic sub-gradient descent procedure:

- Initialize $\mathbf{x}_0 \in \mathbb{R}^D, f_{min} = \infty, \mathbf{x}_* = 0$
- For k from 1 to K:
  - Let $\mathbf{g}_k \in \partial f(\mathbf{x}_{k-1})$
  - Set $\mathbf{x}_k = \mathbf{x}_{k-1} - \alpha_k \mathbf{g}_k$
  - If $f(\mathbf{x}_k) < f_{min}$ then set $f_{min} = f(\mathbf{x}_k)$ and $\mathbf{x}_* = \mathbf{x}_k$
- Return $\mathbf{x}_*$

Sub-gradient Descent

Despite the fact that not all sub-gradients yield descent directions, it is still possible to minimize a convex non-differentiable function using a fairly basic sub-gradient descent procedure:

- Initialize $\mathbf{x}_0 \in \mathbb{R}^D, f_{min} = \infty, \mathbf{x}_* = 0$
- For k from 1 to K:
  - Let $\mathbf{g}_k \in \partial f(\mathbf{x}_{k-1})$
  - Set $\mathbf{x}_k = \mathbf{x}_{k-1} - \alpha_k \mathbf{g}_k$
  - If $f(\mathbf{x}_k) < f_{min}$ then set $f_{min} = f(\mathbf{x}_k)$ and $\mathbf{x}_* = \mathbf{x}_k$
- Return $\mathbf{x}_*$

best solution so far

best value so far

Questions: How to choose α_k? How to choose K?

Subgradient Descent Convergence

- Line search is typically not used for sub-gradient descent procedures. It is more common to use a fixed sequence of step sizes.

Subgradient Descent Convergence

- Line search is typically not used for sub-gradient descent procedures. It is more common to use a fixed sequence of step sizes.
- Common step size rules include $\alpha_k = \alpha/(\beta + k)$ or $\alpha_k = \alpha/\sqrt{k}$.

Subgradient Descent Convergence

- Line search is typically not used for sub-gradient descent procedures. It is more common to use a fixed sequence of step sizes.
- Common step size rules include $\alpha_k = \alpha/(\beta + k)$ or $\alpha_k = \alpha/\sqrt{k}$.
- Under either of these step size rules, we have that the sequence of subgradient descent iterates $\mathbf{x}_k$ satisfies:

$$\lim_{k \to \infty} f(\mathbf{x}_k) = \min_{\mathbf{x}} f(\mathbf{x})$$

Subgradient Descent with Momentum

- Initialize $\mathbf{x}_0 \in \mathbb{R}^D, f_{min} = \infty, \mathbf{x}_* = 0$
- For k from 1 to K:
  - Let $\mathbf{g}_k \in \partial f(\mathbf{x}_{k-1})$
  - Let $\mathbf{d}_k = \gamma \mathbf{d}_{k-1} + \alpha_k \mathbf{g}_k$
  - Set $\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{d}_k$
  - If $f(\mathbf{x}_k) < f_{min}$ then set $f_{min} = f(\mathbf{x}_k)$ and $\mathbf{x}_* = \mathbf{x}_k$
- Return $\mathbf{x}_*$

Typically use with $0 < \gamma < 1$. $\gamma = 0.9$ is a common choice.

Nesterov Accelerated Subgradient Descent

- Initialize $\mathbf{x}_0 \in \mathbb{R}^D, f_{min} = \infty, \mathbf{x}_* = 0$
- For k from 1 to K:
  - Let $\mathbf{g}_k \in \partial f(\mathbf{x}_{k-1} - \gamma \mathbf{d}_{k-1})$
  - Let $\mathbf{d}_k = \gamma \mathbf{d}_{k-1} + \alpha_k \mathbf{g}_k$
  - Set $\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{d}_k$
  - If $f(\mathbf{x}_k) < f_{min}$ then set $f_{min} = f(\mathbf{x}_k)$ and $\mathbf{x}_* = \mathbf{x}_k$
- Return $\mathbf{x}_*$

Typically use with $0 < \gamma < 1$. $\gamma = 0.9$ is a common choice.

Outline

Example: Finding a Subgradient for SVC

- We begin the risk function:

$$
\begin{aligned}
R(\theta, \mathcal{D}) &= C \sum_{n=1}^{N} \max(0, 1 - y_n g_\theta(\mathbf{x}_n)) + ||\mathbf{w}||_2^2 \\
&= C \sum_{n=1}^{N} L(y_n g_\theta(\mathbf{x}_n)) + ||\mathbf{w}||_2^2
\end{aligned}
$$

Example: Finding a Subgradient for SVC

- We begin the risk function:

$$R(\theta, \mathcal{D}) = C \sum_{n=1}^{N} \max(0, 1 - y_n g_\theta(\mathbf{x}_n)) + ||\mathbf{w}||_2^2$$
$$= C \sum_{n=1}^{N} L(y_n g_\theta(\mathbf{x}_n)) + ||\mathbf{w}||_2^2$$

- We are looking for a vector $\mathbf{h}$ such that:

$$\mathbf{h} \in \partial R(\theta, \mathcal{D})$$

Example: Finding a Subgradient for SVC

- By the additivity and scaling properties, we have that if $\mathbf{h}_n \in \partial L(y_n g_\theta(\mathbf{x}_n))$ and $\mathbf{h}_w \in \partial ||\mathbf{w}||_2^2$, then:

$$C \sum_{n=1}^{N} \mathbf{h}_n + \mathbf{h}_w \in \partial R(\theta, \mathcal{D})$$

Example: Finding a Subgradient for SVC

- By the additivity and scaling properties, we have that if $\mathbf{h}_n \in \partial L(y_n g_\theta(\mathbf{x}_n))$ and $\mathbf{h}_w \in \partial ||\mathbf{w}||_2^2$, then:

$$C \sum_{n=1}^{N} \mathbf{h}_n + \mathbf{h}_w \in \partial R(\theta, \mathcal{D})$$

- We will proceed by finding suitable vectors $\mathbf{h}_n$ and $\mathbf{h}_w$ using properties of sub-gradients.

Example: Finding a Subgradient for SVC

- Consider $h_w \in \partial ||\mathbf{w}||_2^2$.

Example: Finding a Subgradient for SVC

- Consider $h_w \in \partial ||\mathbf{w}||_2^2$.
- Since $||\mathbf{w}||_2^2$ is a differentiable function with gradient $2\mathbf{w}$, we have that $h_w = 2[\mathbf{w}; 0]$.

Example: Finding a Subgradient for SVC

- Consider $\mathbf{h}_n \in \partial L(y_n g_\theta(\mathbf{x}_n))$. Recall $L(z) = \max(0, 1 - z)$ and $g_\theta(\mathbf{x}_n) = \mathbf{x}_n\theta$.

Example: Finding a Subgradient for SVC

- Consider $\mathbf{h}_n \in \partial L(y_n g_\theta(\mathbf{x}_n))$. Recall $L(z) = \max(0, 1 - z)$ and $g_\theta(\mathbf{x}_n) = \mathbf{x}_n\theta$.
- By the chain rule, we have that $\mathbf{h}_n \in \partial L(y_n \mathbf{x}_n \theta)$ if and only if:

$$\mathbf{h}_n \in \{k y_n \mathbf{x}_n^T \mid k \in \partial L(z), z = y_n \mathbf{x}_n \theta\}$$

Example: Finding a Subgradient for SVC

- Consider $\mathbf{h}_n \in \partial L(y_n g_\theta(\mathbf{x}_n))$. Recall $L(z) = \max(0, 1 - z)$ and $g_\theta(\mathbf{x}_n) = \mathbf{x}_n\theta$.
- By the chain rule, we have that $\mathbf{h}_n \in \partial L(y_n \mathbf{x}_n \theta)$ if and only if:

$$\mathbf{h}_n \in \{k y_n \mathbf{x}_n^T \mid k \in \partial L(z), z = y_n \mathbf{x}_n \theta\}$$

- We next need to figure out what $\partial L(z)$ is.

Example: Finding a Subgradient for SVC

- Consider $\partial L(z) = \partial \max(0, 1 - z)$. This is a non-differentiable convex function with one point of non-differentiability at $z = 1$.

Example: Finding a Subgradient for SVC

- Consider $\partial L(z) = \partial \max(0, 1 - z)$. This is a non-differentiable convex function with one point of non-differentiability at $z = 1$.
- For $z > 1$ the functions is constant so $\partial L(z) = \{0\}$ for $z > 1$

Example: Finding a Subgradient for SVC

- Consider $\partial L(z) = \partial \max(0, 1 - z)$. This is a non-differentiable convex function with one point of non-differentiability at $z = 1$.
- For $z > 1$ the functions is constant so $\partial L(z) = \{0\}$ for $z > 1$
- For $z < 1$ the functions is $1 - z$ so $\partial L(z) = \{-1\}$ for $z < 1$.

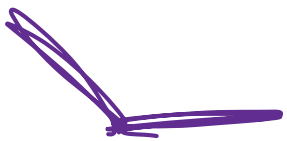
Example: Finding a Subgradient for SVC

- Consider $\partial L(z) = \partial \max(0, 1 - z)$. This is a non-differentiable convex function with one point of non-differentiability at $z = 1$.
- For $z > 1$ the functions is constant so $\partial L(z) = \{0\}$ for $z > 1$
- For $z < 1$ the functions is $1 - z$ so $\partial L(z) = \{-1\}$ for $z < 1$.
- At $z = 0$ we have $\partial L(z) = [-1, 0]$ following subdifferential rules for 1-D convex functions.

Example: Finding a Subgradient for SVC

- Consider $\partial L(z) = \partial \max(0, 1 - z)$. This is a non-differentiable convex function with one point of non-differentiability at $z = 1$.
- For $z > 1$ the functions is constant so $\partial L(z) = \{0\}$ for $z > 1$
- For $z < 1$ the functions is $1 - z$ so $\partial L(z) = \{-1\}$ for $z < 1$.
- At $z = 0$ we have $\partial L(z) = [-1, 0]$ following subdifferential rules for 1-D convex functions.
- We choose the following sub-gradient function for the hinge loss:

$$L'(z) = \begin{cases} 0 & \dots z \geq 1 \\ -1 & \dots z < 1 \end{cases}$$



Example: Finding a Subgradient for SVC

- Consider again, $\mathbf{h}_n \in \partial L(y_n \mathbf{x}_n \theta)$ if and only if:

$$\mathbf{h}_n \in \{k y_n \mathbf{x}_n^T \mid k \in \partial L(z), z = y_n \mathbf{x}_n \theta\}$$

Example: Finding a Subgradient for SVC

- Consider again, $\mathbf{h}_n \in \partial L(y_n \mathbf{x}_n \theta)$ if and only if:

$$\mathbf{h}_n \in \{k y_n \mathbf{x}_n^T \mid k \in \partial L(z), z = y_n \mathbf{x}_n \theta\}$$

- We have shown that $L'(z) \in \partial L(z)$, so we have that:

$$\mathbf{h}_n = L'(y_n \mathbf{x}_n \theta) y_n \mathbf{x}_n^T \in \partial L(y_n \mathbf{x}_n \theta)$$

Example: Finding a Subgradient for SVC

- Consider again, $\mathbf{h}_n \in \partial L(y_n \mathbf{x}_n \theta)$ if and only if:

$$\mathbf{h}_n \in \{k y_n \mathbf{x}_n^T \mid k \in \partial L(z), z = y_n \mathbf{x}_n \theta\}$$

- We have shown that $L'(z) \in \partial L(z)$, so we have that:

$$\mathbf{h}_n = L'(y_n \mathbf{x}_n \theta) y_n \mathbf{x}_n^T \in \partial L(y_n \mathbf{x}_n \theta)$$

- This gives us a final answer for a subgradient of the risk:

$$\mathbf{h} = C \sum_{n=1^N} L'(y_n \mathbf{x}_n \theta) y_n \mathbf{x}_n^T + 2[\mathbf{w}; 0]$$