OPTIMIZATION FOR MACHINE LEARNING

November 6, 2023

Today: Last lecture

Thursday: Last year's exam (with Florian Le Bronnec)

↳ Course project is online! Deadline: January 15

REGULARIZATION

Motivation: A typical learning problem

$$\underset{w \in \mathbb{R}^d}{\text{minimize}} \quad \frac{1}{n}\sum_{i=1}^{n} f_i(w) + \lambda \Omega(w)$$

- $\frac{1}{n}\sum_{i=1}^{n} f_i(w)$: "Finite sum" — Every f_i depends on a data point in a dataset of size n
- $\lambda > 0$
- $\Omega : \mathbb{R}^d \to \mathbb{R}$: "Regularization term" that does not depend on the data

1) We have seen how to minimize the finite sum using stochastic gradient when $f_i \in C^1 \ \forall i$:
⇒ But what if the functions are not C^1?

Ex) Neural network with ReLU activations
$t \mapsto \max(t, 0)$ not C^1

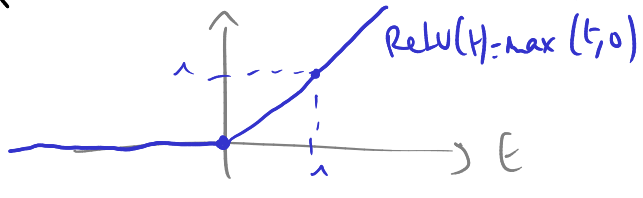
2) What is the function Ω used for?
Do we need to take into account in the optimization process?

(1) Optimization without gradients

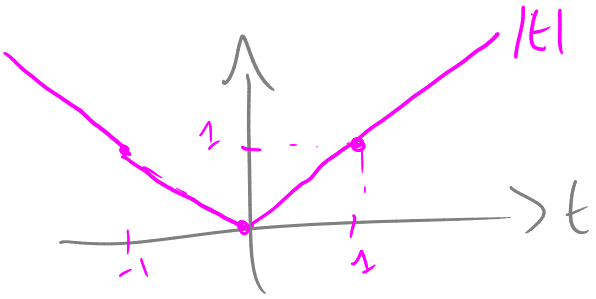
↳ So far all the methods we have seen rely on gradients

↳ But many functions used in ML are nonsmooth, i.e. there are points at which those functions do not have a gradient

Ex)



does not have a gradient at 0



$t \mapsto |t|$ does not have a gradient at 0

$\Rightarrow$ Not having a gradient is an issue for both designing an optimization algorithm and for checking that a point is a minimum

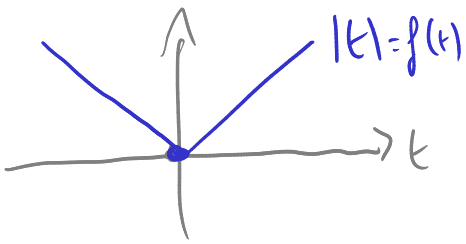
$\Rightarrow$ For convex functions, we can build a tool that plays the role of the gradient in optimization

Def: Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex function and $w \in \mathbb{R}^d$. A vector $g \in \mathbb{R}^d$ is called a subgradient of f at w if $\forall z \in \mathbb{R}^d$, $f(z) \geq f(w) + g^T(z - w)$

The set of all subgradients for f at w is called the subdifferential of f at w, and denoted by $\partial f(w) \subseteq \mathbb{R}^d$

↳ Why is that an appropriate definition?

- If f has a gradient at w, then $\partial f(w) = \{\nabla f(w)\}$
  $\Rightarrow$ More general concept than gradient
- $[\bar{w} \in \mathbb{R}^d$ is a global minimum of $f] \iff 0_{\mathbb{R}^n} \in \partial f(\bar{w})$
  $\Rightarrow$ Optimality condition based on the subdifferential (can check whether a point is a minimum)



$$\forall t \in \mathbb{R}, \quad \partial f(t) = \begin{cases} \{1\} & \text{if } t > 0 \\ \{-1\} & \text{if } t < 0 \\ [-1, 1] & \text{if } t = 0 \end{cases}$$

Remarks

- Using subgradients instead of gradients, we can generalize all gradient-based algorithms (gradient descent, stochastic gradient, ...) so that apply to convex nonsmooth functions

Ex) GD $w_{k+1} = w_k - \alpha_k \nabla f(w_k)$ for $f \in C^1$

Subgradient method $w_{k+1} = w_k - \alpha_k g_k$ $g_k \in \partial f(w_k)$

⚠ The subgradient method is more difficult to implement and analyze

- In practice, gradients and subgradients are computed using automatic differentiation (=backward pass in a neural network) and it works

(2) Regularization terms

↳ In general there are many ways to fit some data, especially with lots of parameters

$\Rightarrow$ minimize $\frac{1}{n}\sum_{i=1}^{n} f_i(w)$ $w \in \mathbb{R}^d$ with d large enough

$\Rightarrow$ there exist many solutions to the problem

↳ Typically we have preferences on the models/parameters we would like to obtain (simplicity / as few parameters as possible, robustness / not too sensitive to data, generalization / model also fits unseen data)

$\Rightarrow$ We can encode these preferences into the objective of the optimization problem using regularization

Two main examples

- **ℓ_2 regularization / ridge regularization**

$$\|w\|^2 = \sum_{j=1}^{d} w_j^2 \qquad \lambda > 0$$

$$\underset{w \in \mathbb{R}^d}{\text{minimize}} \quad \frac{1}{n} \sum_{i=1}^{n} f_i(w) + \frac{\lambda}{2} \|w\|^2$$

($\lambda = 0 \Rightarrow$ No regularization)

- As $\lambda \to +\infty$, the problem gets closer to

$$\underset{w \in \mathbb{R}^d}{\text{minimize}} \quad \frac{\lambda}{2} \|w\|^2$$

, that has solution $\bar{w} = 0$

- With $\lambda > 0$, the solution of the regularized problem is less sensitive to changes in the data than the solution of the problem without regularization

$\Rightarrow$ We say that ℓ_2 regularization reduces the variance of the solution with respect to the data

- Often implemented in ML for Stochastic gradient under the name "Weight decay"

(increasing λ reduces the value of the w_i^2)

$\Rightarrow$ Typically improves generalization

- **ℓ_1 regularization / LASSO**

$$\|w\|_1 = \sum_{j=1}^{d} |w_j| \qquad \lambda > 0$$

$$\underset{w \in \mathbb{R}^d}{\text{minimize}} \quad \frac{1}{n} \sum_{i=1}^{n} f_i(w) + \lambda \|w\|_1$$

Nonsmooth function (but convex!)

↳ For $\lambda > 0$, the solution of the regularized problem has more zero coordinates than the solution without regularization

sparse = lots of zero coefficients

$\Rightarrow$ We say that ℓ_1 regularization promotes sparse solutions

↳ For linear models, ℓ_1 regularization automatically performs feature selection

$\Rightarrow$ More generally, helps in identifying the most important parameters

Optimizing a regularized problem

$$\underset{w \in \mathbb{R}^d}{\text{minimize}} \quad \underbrace{\frac{1}{n}\sum_{i=1}^{n} f_i(w)}_{f(w)} + \lambda \Omega(w)$$

$f : \mathbb{R}^d \to \mathbb{R}$ C^1

$\Omega : \mathbb{R}^d \to \mathbb{R}$ convex

↳ Proximal gradient techniques exploit the structure of regularized problems

Iteration k:

$$w_{k+1} \in \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \left\{ f(w_k) + \nabla f(w_k)^T (w - w_k) + \frac{1}{2\alpha_k}\|w - w_k\|^2 + \lambda \Omega(w) \right\}$$

- $f(w_k) + \nabla f(w_k)^T (w - w_k) \approx f(w)$
- $\frac{1}{2\alpha_k}\|w - w_k\|^2$: "proximal term" ($\approx$ stepsize), $\alpha_k > 0$
- $\lambda \Omega(w)$: regularization (unchanged)

- Every iteration requires to solve an optimization problem $\Rightarrow$ Only worth doing if the "subproblem" (problem at iteration k) is easier to solve than the original one
- Special cases
  - $\Omega(w) = 0 \ \forall w \in \mathbb{R}^d$: the iteration becomes

$w_{k+1} = w_k - \alpha_k \nabla f(w_k)$, which is the GD iteration

* $\Omega(w) = \frac{1}{2}\|w\|^2$ (l_2 regularization)

$$w_{k+1} = w_k - \frac{\alpha_k}{1+\lambda\alpha_k} \nabla f(w_k)$$

"GD with a modified stepsize"

* $\Omega(w) = \|w\|_1$ (l_1 regularization)

→ Explicit formula for w_{k+1}

→ Proximal gradient in that setting corresponds to a method called ISTA from signal processing

NB

Extensions of proximal gradient:

→ acceleration
→ stochastic
→ subgradients