Image Reconstruction

Setup:

$$y = Hx + n$$

- $y \in \mathbb{R}^M$ measured data
- $H \in \mathbb{R}^{M \times N}$ discretized forward model
- $x \in \mathbb{R}^N$ discretization coeffs. of of some analog object $s \in L^2(\mathbb{R}^2)$
- $n \in \mathbb{R}^M$ additive measurement noise.

Goal: Given y, reconstruct x.

Q: How did we try to solve this problem?

A: Least-Squares and variants

Least Squares

$$\min_{x \in \mathbb{R}^N} \underbrace{\|y - Hx\|_2^2}_{J(x)}$$

<u>Q:</u> How did we solve this optimization problem?

<u>A:</u> Iterative approach:

$$x_{k+1} = x_k - \mu H^T(Hx_k - y), \quad k = 0, 1, 2, \ldots$$

where x_0 was arbitrary, $0 < \mu < \frac{2}{\sigma_1^2(H)}$.

<u>Last time:</u> we proved that

$$x_k \rightarrow x^* \quad \text{as } k \rightarrow \infty,$$

where x^* is a solution to

$$\min_{x \in \mathbb{R}^N} \|y - Hx\|_2^2 .$$

Q: How did we derive this iteration?

A: Examined $0 = \frac{\partial J(x)}{\partial x}$.

$$J(x) = (y - Hx)^T (y - Hx)$$

$$= y^T y - y^T Hx - x^T H^T y + x^T H^T H x$$

$$= -2 x^T H^T y + x^T H^T H x$$

$$0 = \frac{\partial J(x)}{\partial x} = -2 H^T y + 2 H^T H x$$

$\Rightarrow$ Every solution must satisfy

$$H^T H x = H^T y$$ "normal equations"

Heuristic-Based Iterative Procedure

Given an initial "guess" x_0, define the error vector:

$$e = H^T H x_0 - H^T y$$

Try to reduce the error by considering

$$x_1 = x_0 - \mu e, \quad \mu > 0 \text{ is the step size (or, learning rate)}$$

Then, we can repeat this process:

$$x_{k+1} = x_k - \mu H^T (H x_k - y)$$

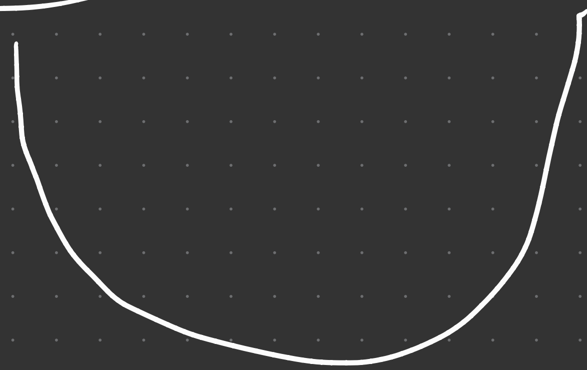
This is called the "Landweber iteration".

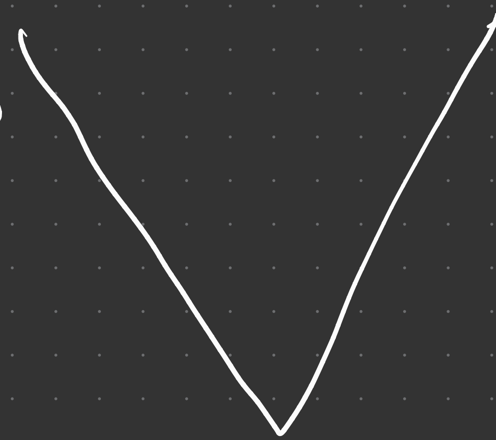
<u>Remark:</u> It turns out that this is a special case of "gradient descent".

<u>Q:</u> How do we minimize general functions $J : \mathbb{R}^N \to \mathbb{R}$?

<u>Q:</u> Can we hope to minimize general functions $J : \mathbb{R}^N \to \mathbb{R}$?

Exercise:

1 

2 

3 

4 

Which ones are "easy" to minimize?

A: 1 & 2

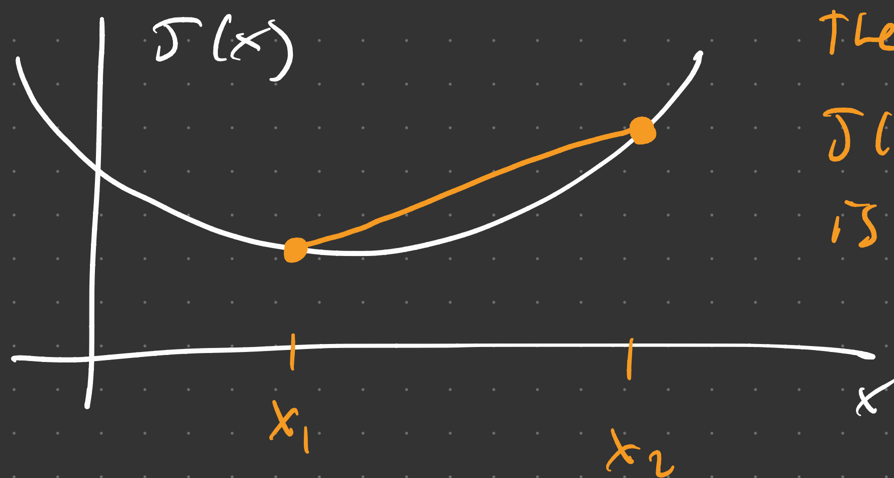
Q: What is the common property?

A: Convexity

Convex Functions

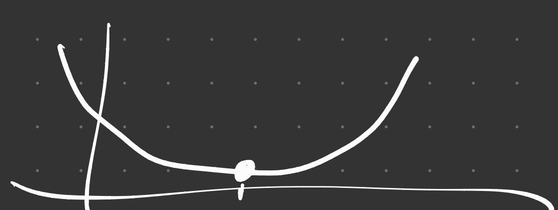
<u>Def$^{\underline{n}}$</u>: $J : \mathbb{R}^N \to \mathbb{R}$ is said to be convex if, for all $x_1, x_2 \in \mathbb{R}^N$ and $\lambda \in [0,1]$

$$\lambda J(x_1) + (1-\lambda) J(x_2) \geq J(\lambda x_1 + (1-\lambda) x_2)$$



The line connecting $J(x_1)$ & $J(x_2)$ is "above" J

All minima of a convex function are global minima



unique global min

non unique global min

Obs: Negative gradient <u>always</u> points towards global minima!

Convexity and Gradients

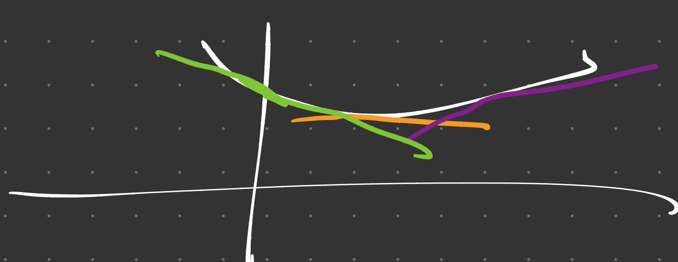
$$\lambda J(x_1) + (1-\lambda) J(x_0) \geq J(\lambda x_1 + (1-\lambda) x_0)$$

$$\Rightarrow J(x_1) \geq J(x_0) + \frac{J(x_0 + \lambda(x_1 - x_0)) - J(x_0)}{\lambda}$$

Q: What happens when $\lambda \to 0$?

$$J(x_1) \geq J(x_0) + \underbrace{\left[\left.\frac{\partial J}{\partial x}\right|_{x=x_0}\right]^T (x_1 - x_0)}_{D_{(x_1 - x_0)} J(x_0)}$$

$\Rightarrow J$ lies "above" all its tangents



Gradient Descent

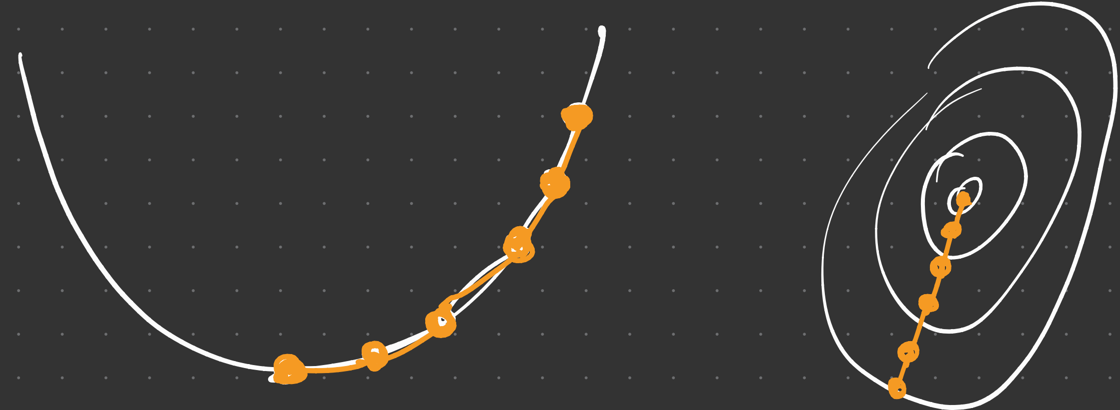
① Start with an initial guess x_0.

② Iterate

$$x_{k+1} = x_k - \gamma \frac{\partial J}{\partial x}\Big|_{x=x_k},$$

where $\gamma > 0$ is the step size / learning rate.

Obs: For convex functions, GD with sufficiently small step sizes will converge to a global min.



Exercise: Write down the GD iteration for the least-squares problem:

$$\min_{x \in \mathbb{R}^n} \underbrace{\|y - Hx\|_2^2}_{J(x)}.$$

$$\frac{\partial J}{\partial x} = -2H^T y + 2H^T H x$$

$$= 2H^T(Hx - y)$$

The GD iteration would be:

$$x_{k+1} = x_k - \gamma\left[2H^T(Hx_k - y)\right]$$

$$= x_k - \underbrace{2\gamma}_{\mu} H^T(Hx_k - y)$$

Obs: Same as the Landweber iteration.

Regularization

$$\min_{x \in \mathbb{R}^N} \|y - Hx\|_2^2 + \lambda R(x), \quad \lambda > 0.$$

- $R(x)$ is called a regularizer and it allows us to inject any prior knowledge about x into the optimization problem.
- λ is the regularization parameter and controls the <u>strength</u> of the regularizer.

<u>Exercise:</u> Suppose we knew, a priori, that x_1 and x_N were always 0. How could we encode that knowledge into $R(x)$?

One possibility:

$$R(x) = R\left(\begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}\right) = \begin{cases} \infty & \text{if } x_1 \text{ or } x_N \neq 0 \\ 0 & \text{else} \end{cases}$$

Three eras of regularizers for image reconstruction

1. $R(x) = \|Lx\|_2^2$ pre 1990

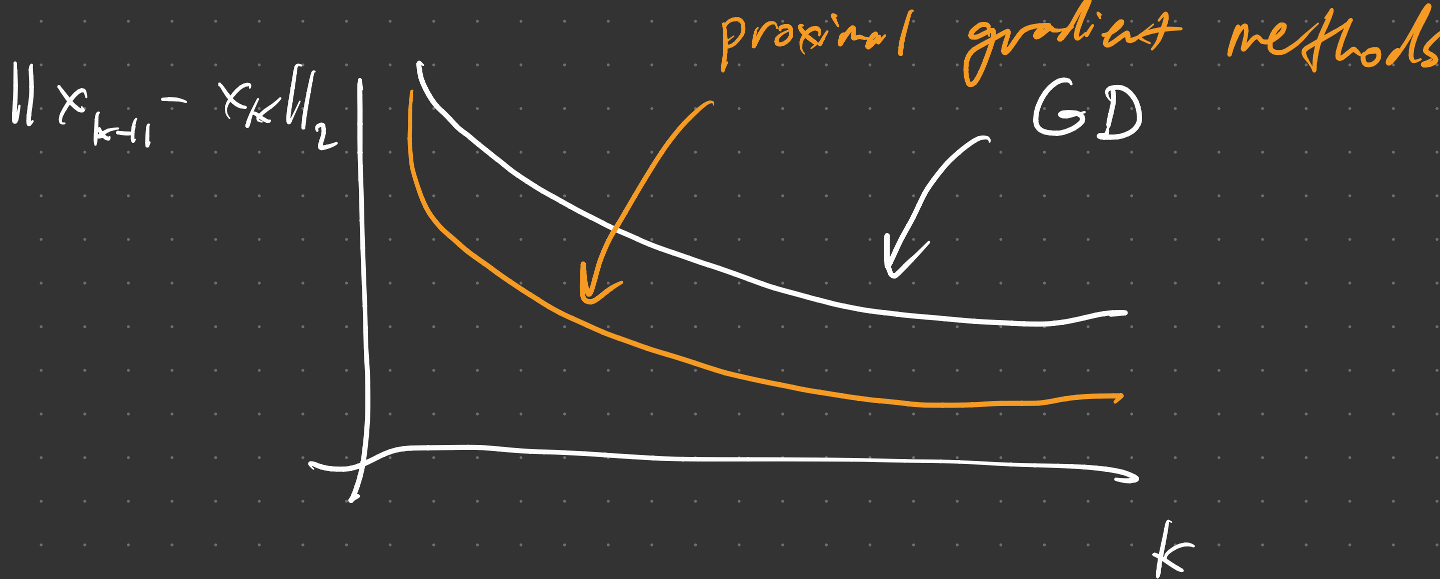
2. $R(x) = \|Lx\|_1$ 1990 – 2015

3. $R(x)$ is learned from data using deep learning now

Remark: Solving

$$\min_{x \in \mathbb{R}^N} \|y - Hx\|_2^2 + \lambda R(x)$$

is very slow ...

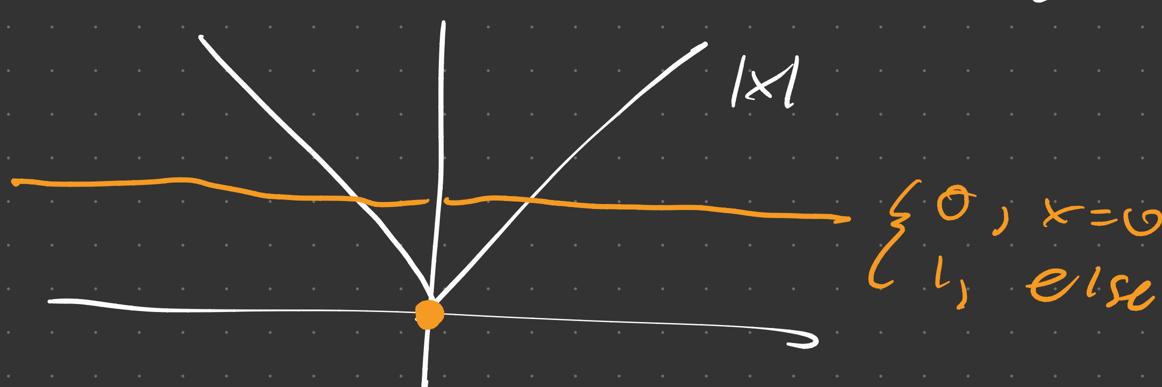


Case Study: ℓ^1-norm Regularization

$$\min_{x \in \mathbb{R}^N} \|y - Hx\|_2^2 + \lambda \|Lx\|_1$$

Thought experiment: Suppose $L = I$.

What is the effect of penalizing $\|x\|_1$?



penalizing the # of nonzero entries of x

"sparsity"

Common choices of L:

1. L is a discrete gradient
   - TV regularization

     $\|\nabla x\|_1$ is the total variation of x

2. L is a discrete wavelet transform (DWT)
   - $\|Wx\|_1$ = good model for natural images
   - ECE 251C

Focus on orthogonal DWT:

$$W^T W = W W^T = I$$

$$\min_{x \in \mathbb{R}^N} \|y - Hx\|_2^2 + \lambda \|Wx\|_1$$

$$\theta = Wx$$

$$x = W^T \theta$$

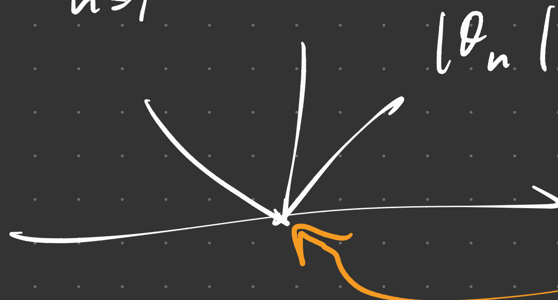
$$A = H W^T$$

We can focus on:

$$\min_{\theta \in \mathbb{R}^N} \|y - A\theta\|_2^2 + \lambda \|\theta\|_1$$

<u>Q:</u> Does something seem weird?

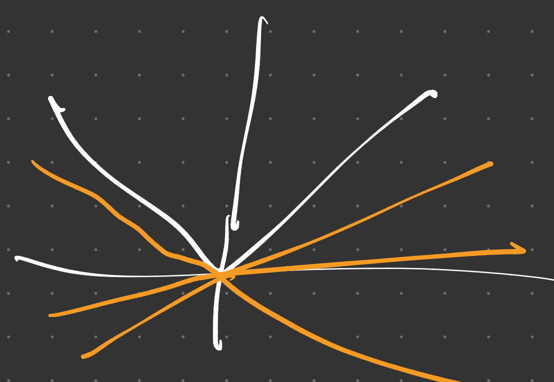
<u>Obs:</u> $\|\theta\|_1 = \sum_{n=1}^{N} |\theta_n|$



not differentiable at 0!

$$\frac{\partial}{\partial \theta_n} |\theta_n| = \begin{cases} 1, & \theta_n > 0 \\ \text{undefined}, & \theta_n = 0 \\ -1, & \theta_n < 0 \end{cases}$$

Subgradient / Subdifferential



$$\partial |\theta_n| = \begin{cases} +1, & \theta_n > 0 \\ \alpha \in [-1, 1], & \theta_n = 0 \\ -1, & \theta_n < 0 \end{cases}$$

any line that touches 0 and lies below $|\theta_n|$ is a subgradient

<u>Remark:</u> Can use subgradients in place of gradients.

<u>Special Case:</u> If $A = I$, then   denoising problem

$$\|y - \theta\|_2^2 + \lambda \|\theta\|_1$$

$$= \sum_{n=1}^{N} \left[(y_n - \theta_n)^2 + \lambda |\theta_n| \right]$$

decoupled $\Rightarrow$ optimize term-by-term

Consider the scalar optimization problem:

$$\min_{\theta \in \mathbb{R}} \underbrace{(y - \theta)^2 + \lambda |\theta|}_{J(\theta)}$$

<u>Exercise:</u> Find an expression for a minimizer $\hat{\theta}$ of this optimization problem.

Need to check when

$$0 \in \partial J(\hat{\theta})$$

<u>Case ①:</u> $\hat{\theta} \neq 0$

$$0 = \left.\frac{\partial J}{\partial \theta}\right|_{\theta = \hat{\theta}} = 2(\hat{\theta} - y) + \lambda \, \mathrm{sgn}(\hat{\theta})$$

if $\hat{\theta} < 0$, then

$$0 = 2\hat{\theta} - 2y - \lambda$$

$$\Rightarrow \hat{\theta} = y + \frac{\lambda}{2} \Rightarrow y < 0 \Rightarrow \hat{\theta} = y - \operatorname{sgn}(y)\frac{\lambda}{2}$$

if $\hat{\theta} > 0$, then

$$0 = 2\hat{\theta} - 2y + \lambda$$

$$\hat{\theta} = y - \frac{\lambda}{2} \Rightarrow y > 0 \Rightarrow \hat{\theta} = y - \operatorname{sgn}(y)\frac{\lambda}{2}$$

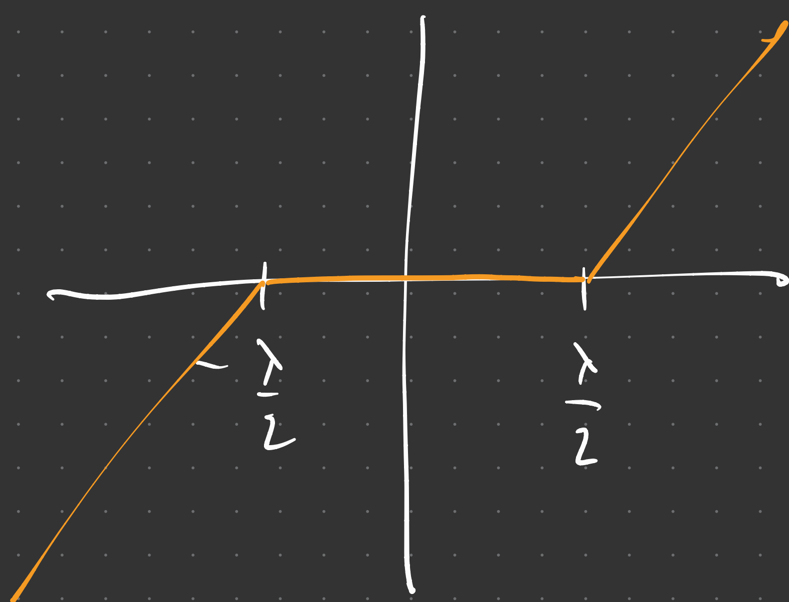
if $\hat{\theta} = 0$, then

$$0 \in -2y + \lambda[-1, 1]$$

$$0 \in -y + \left[-\frac{\lambda}{2}, \frac{\lambda}{2}\right]$$

$$y \in \left[-\frac{\lambda}{2}, \frac{\lambda}{2}\right] \Rightarrow |y| \leq \frac{\lambda}{2}$$

$$\hat{\theta} = \begin{cases} 0, & |y| \le \frac{\lambda}{2} \\ y - \frac{\lambda}{2}\,\text{sgn}(y), & |y| > \frac{\lambda}{2} \end{cases} = \text{sgn}(y)\max\{0, |y| - \tfrac{\lambda}{2}\}$$



soft threshold function

Iterative Soft-Thresholding Algorithm (ISTA)

Goal: Efficiently solve

$$\min_{\theta \in \mathbb{R}^n} \underbrace{\|y - A\theta\|_2^2}_{\text{error}} + \underbrace{\lambda \|\theta\|_1}_{\text{reg}}$$

ISTA:

1. initialize θ_0
2. Iterate:

GD: $\tilde{\theta}_{k+1} = \theta_k - \mu A^T(A\theta_k - y)$

ST: $\theta_{k+1} = \text{sgn}(\tilde{\theta}_{k+1}) \max\{0, |\tilde{\theta}_{k+1}| - \frac{\mu\lambda}{2}\}$

<u>Remark:</u> GD is reducing the error
ST is reducing the regs.

If $\mu < \frac{2}{\sigma_1^2(A)}$, then $\theta_k \to \theta^*$

<u>much faster</u> than GD.

<u>Reformulation of ISTA</u>

① $\tilde{\theta}_{k+1} = \theta_k - \mu A^T(A\theta_k - y)$

② $\theta_{k+1} = \underset{\theta}{\text{argmin}} \|\tilde{\theta}_{k+1} - \theta\|_2^2 + \mu\lambda\|\theta\|_1$

<u>Obs:</u> ① is a gradient step
② is a <u>denoising step</u>

Proximal Gradient Methods

Goal: Efficiently solve

$$\min_{\theta \in \mathbb{R}^N} \|y - A\theta\|_2^2 + R(\theta)$$

Proximal Gradient Descent

① $\tilde{\theta}_{k+1} = \theta_k - \mu A^T(A\theta_k - y)$

② $\theta_{k+1} = \underbrace{\operatorname*{argmin}_\theta \|\tilde{\theta}_{k+1} - \theta\|_2^2 + \mu\lambda R(\theta)}_{\text{prox}_{\mu\lambda R}(\tilde{\theta}_{k+1})}$

denoiser

Obs: ISTA is a special case of proximal gradient descent.

Plug-and-Play (PnP) Methods (2021)

General idea: We have access to lots of training data

$$\{(x_i, y_i)\}_{i=1}^{N},$$

where each x_i is a noise-free image and each y_i is a noisy image (can also generate data easily with noise-free images).

Setup: Train a neural network to learn how to denoise images. Let $D : \mathbb{R}^N \to \mathbb{R}^N$ be the NN.

PnP iteration

$$\tilde{\theta}_{k+1} = \theta_k - \mu A^T (A \theta_k - y)$$

$$\theta_{k+1} = D(\tilde{\theta}_{k+1})$$