The Gradient Descent Algorithm

Definitions: Convexity I

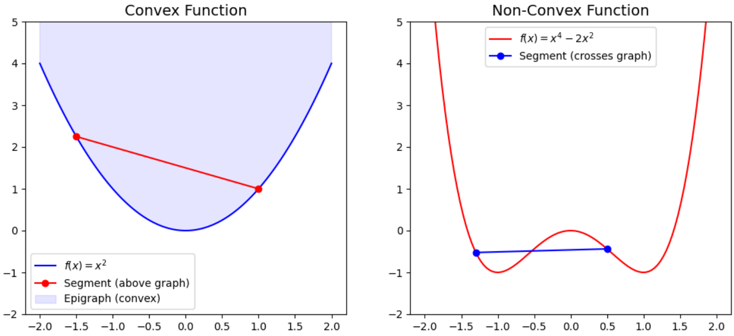
Formal Definition (Convexity)

A function f is convex if its domain $dom(f)$ is a convex set and $\forall \boldsymbol{x}, \boldsymbol{y} \in dom(f), \lambda \in [0,1]$:

$$f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leq \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y})$$

- **Practical Meaning:** The line segment connecting any two points on the function's graph lies *above* or on the graph.
- **Key Implication:** Every local minimum is also a global minimum.

Definitions: Convexity II



Definitions: First-Order Characterization I

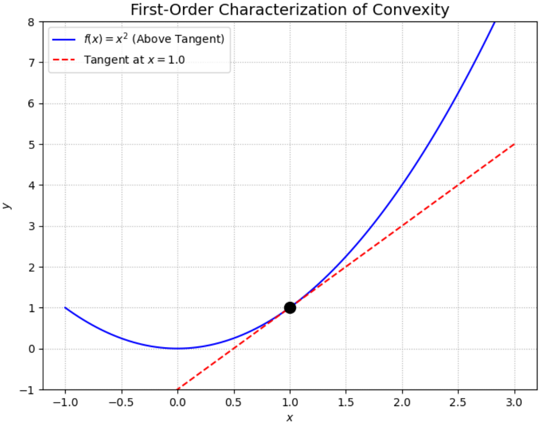
First-Order Characterization

A differentiable f is convex if and only if:

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) \quad \forall \boldsymbol{x}, \boldsymbol{y}$$

Practical Meaning: The tangent hyperplane at any point $\boldsymbol{x}$ lies entirely *below* the graph of the function.

Definitions: First-Order Characterization II



Definitions: Lipschitz and Smoothness

B-Lipschitz

f is B-Lipschitz if its "steepness" is globally bounded:

$$|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq B||\boldsymbol{x} - \boldsymbol{y}|| \quad \forall \boldsymbol{x}, \boldsymbol{y}$$

For convex, differentiable functions, this is equivalent to:

$$||\nabla f(\boldsymbol{x})|| \leq B \quad \forall \boldsymbol{x} \quad \text{(Bounded Gradients)}$$

L-Smoothness (Gradient Lipschitz)

f is L-smooth if its gradient is L-Lipschitz:

$$||\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})|| \leq L||\boldsymbol{x} - \boldsymbol{y}|| \quad \forall \boldsymbol{x}, \boldsymbol{y}$$

Practical Meaning: The function's curvature is bounded *above*. It cannot bend or curve "too sharply".

Definitions: Smoothness & Strong Convexity I

These properties provide tangential quadratic bounds.

L-Smoothness (Quadratic Upper Bound)

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) + \frac{L}{2} ||\boldsymbol{y} - \boldsymbol{x}||^2$$

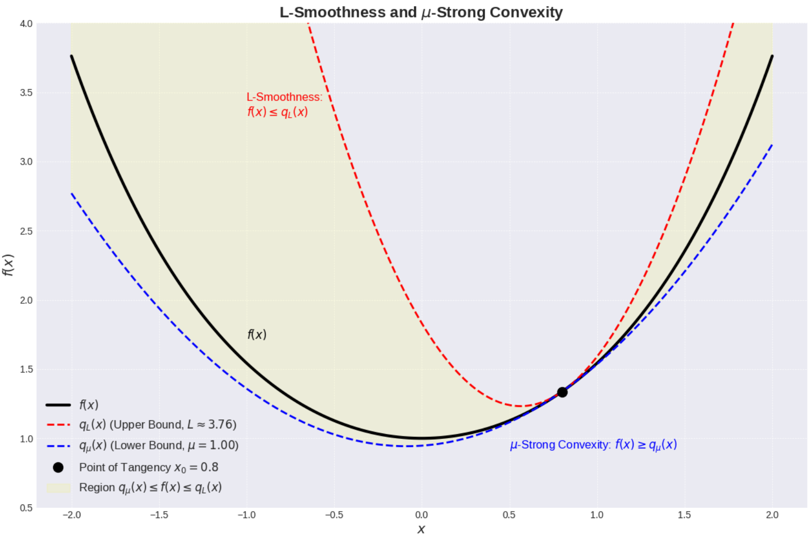
The function always lies *below* a quadratic "bowl" pointing up.
(Prove as exercise)

μ-Strong Convexity (Quadratic Lower Bound)

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}) + \frac{\mu}{2} ||\boldsymbol{y} - \boldsymbol{x}||^2$$

The function always lies *above* a quadratic "bowl" pointing up. It is "at least" quadratic, guaranteeing a unique, sharp minimum.

Definitions: Smoothness & Strong Convexity II



The Gradient Descent Algorithm: idea

Objective

We want to minimize a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$. We seek to find:

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in \mathbb{R}^d} f(\boldsymbol{x})$$

The Gradient Descent Algorithm: idea

Objective

We want to minimize a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$. We seek to find:

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in \mathbb{R}^d} f(\boldsymbol{x})$$

- Gradient Descent (GD) is an iterative method.
- It generates a sequence of solutions $\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$
- The update rule is: $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \boldsymbol{v}_t$.

The Gradient Descent Algorithm: idea

Objective

We want to minimize a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$. We seek to find:

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x})$$

- Gradient Descent (GD) is an iterative method.
- It generates a sequence of solutions $\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \dots$
- The update rule is: $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \boldsymbol{v}_t$.

How to choose the direction $\boldsymbol{v}_t$?

We want $f(\boldsymbol{x}_{t+1}) < f(\boldsymbol{x}_t)$.
Using a first-order Taylor expansion (for small $\boldsymbol{v}_t$):

$$f(\boldsymbol{x}_t + \boldsymbol{v}_t) \approx f(\boldsymbol{x}_t) + \nabla f(\boldsymbol{x}_t)^\top \boldsymbol{v}_t$$

To make f decrease, we must choose $\boldsymbol{v}_t$ such that $\nabla f(\boldsymbol{x}_t)^\top \boldsymbol{v}_t < 0$.

The Gradient Descent Algorithm I

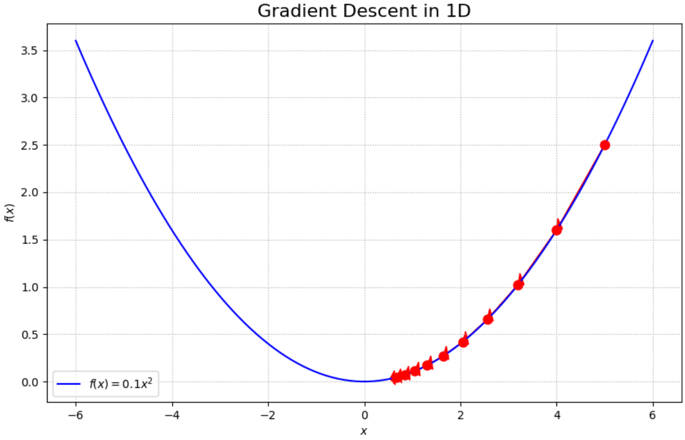
- The direction that maximizes this decrease (for a given step length) is the direction of the **negative gradient**.
- We choose $\boldsymbol{v}_t = -\gamma \nabla f(\boldsymbol{x}_t)$, where $\gamma > 0$ is the **step size** (or **learning rate**).

The Gradient Descent Algorithm

Choose an initial $\boldsymbol{x}_0$. For $t = 0, 1, 2, \ldots$:

$$\boldsymbol{x}_{t+1} := \boldsymbol{x}_t - \gamma \nabla f(\boldsymbol{x}_t)$$

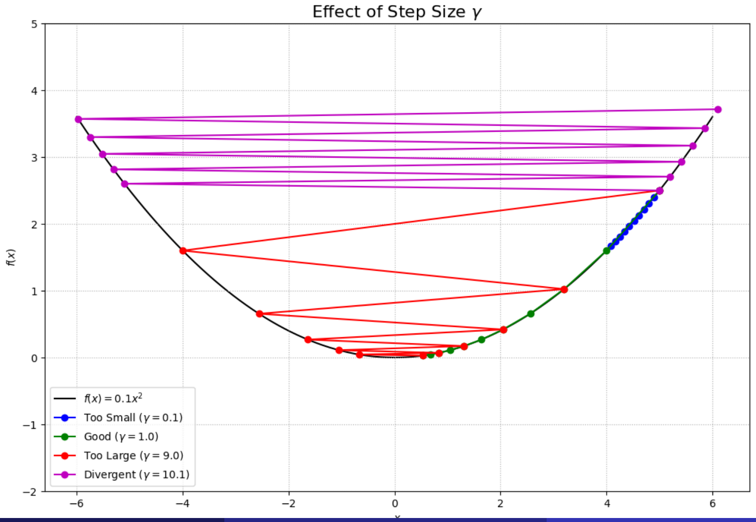
The Gradient Descent Algorithm II



⏩ Colab ⏩ Link

The Importance of the Step Size γ

The choice of γ is critical for performance.



Basic convergence Analysis (Proof)

Our goal is to bound the error $\boxed{f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)}$.

1. **Convexity:** From the first-order characterization, we have:

$$f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*) \leq \nabla f(\boldsymbol{x}_t)^\top (\boldsymbol{x}_t - \boldsymbol{x}^*)$$

Let $\boldsymbol{g}_t = \nabla f(\boldsymbol{x}_t)$. We need to bound $\boldsymbol{g}_t^\top (\boldsymbol{x}_t - \boldsymbol{x}^*)$.

Basic convergence Analysis (Proof)

Our goal is to bound the error $\boxed{f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)}$.

1. **Convexity:** From the first-order characterization, we have:

$$f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*) \leq \nabla f(\boldsymbol{x}_t)^\top (\boldsymbol{x}_t - \boldsymbol{x}^*)$$

   Let $\boldsymbol{g}_t = \nabla f(\boldsymbol{x}_t)$. We need to bound $\boldsymbol{g}_t^\top (\boldsymbol{x}_t - \boldsymbol{x}^*)$.

2. **GD Step:** We can express the gradient using the algorithm's update rule:

$$\boldsymbol{g}_t = \frac{\boldsymbol{x}_t - \boldsymbol{x}_{t+1}}{\gamma}$$

Basic convergence Analysis (Proof)

Our goal is to bound the error $\boxed{f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)}$.

1. **Convexity:** From the first-order characterization, we have:

$$f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*) \leq \nabla f(\boldsymbol{x}_t)^\top (\boldsymbol{x}_t - \boldsymbol{x}^*)$$

   Let $\boldsymbol{g}_t = \nabla f(\boldsymbol{x}_t)$. We need to bound $\boldsymbol{g}_t^\top (\boldsymbol{x}_t - \boldsymbol{x}^*)$.

2. **GD Step:** We can express the gradient using the algorithm's update rule:

$$\boldsymbol{g}_t = \frac{\boldsymbol{x}_t - \boldsymbol{x}_{t+1}}{\gamma}$$

3. **Substitution:**

$$\boldsymbol{g}_t^\top (\boldsymbol{x}_t - \boldsymbol{x}^*) = \frac{1}{\gamma}(\boldsymbol{x}_t - \boldsymbol{x}_{t+1})^\top (\boldsymbol{x}_t - \boldsymbol{x}^*)$$

Basic convergence Analysis (Proof)

4. **Algebraic Identity:** We use the identity
$2\boldsymbol{v}^\top \mathbf{w} = ||\boldsymbol{v}||^2 + ||\mathbf{w}||^2 - ||\boldsymbol{v} - \mathbf{w}||^2$.
Let $\boldsymbol{v} = \boldsymbol{x}_t - \boldsymbol{x}_{t+1}$ and $\mathbf{w} = \boldsymbol{x}_t - \boldsymbol{x}^*$.
Then $\boldsymbol{v} - \mathbf{w} = \boldsymbol{x}^* - \boldsymbol{x}_{t+1}$.

$$2(\boldsymbol{x}_t - \boldsymbol{x}_{t+1})^\top (\boldsymbol{x}_t - \boldsymbol{x}^*) = ||\boldsymbol{x}_t - \boldsymbol{x}_{t+1}||^2 + ||\boldsymbol{x}_t - \boldsymbol{x}^*||^2 - ||\boldsymbol{x}_{t+1} - \boldsymbol{x}^*||^2$$

4. **Algebraic Identity:** We use the identity $2\boldsymbol{v}^\top \mathbf{w} = ||\boldsymbol{v}||^2 + ||\mathbf{w}||^2 - ||\boldsymbol{v} - \mathbf{w}||^2$.
   Let $\boldsymbol{v} = \boldsymbol{x}_t - \boldsymbol{x}_{t+1}$ and $\mathbf{w} = \boldsymbol{x}_t - \boldsymbol{x}^*$.
   Then $\boldsymbol{v} - \mathbf{w} = \boldsymbol{x}^* - \boldsymbol{x}_{t+1}$.

$$2(\boldsymbol{x}_t - \boldsymbol{x}_{t+1})^\top (\boldsymbol{x}_t - \boldsymbol{x}^*) = ||\boldsymbol{x}_t - \boldsymbol{x}_{t+1}||^2 + ||\boldsymbol{x}_t - \boldsymbol{x}^*||^2 - ||\boldsymbol{x}_{t+1} - \boldsymbol{x}^*||^2$$

5. **Reformulation:** Substitute this back and use $||\boldsymbol{x}_t - \boldsymbol{x}_{t+1}||^2 = \gamma^2 ||\boldsymbol{g}_t||^2$:

$$\begin{aligned}\boldsymbol{g}_t^\top (\boldsymbol{x}_t - \boldsymbol{x}^*) &= \frac{1}{2\gamma}\left(\gamma^2 ||\boldsymbol{g}_t||^2 + ||\boldsymbol{x}_t - \boldsymbol{x}^*||^2 - ||\boldsymbol{x}_{t+1} - \boldsymbol{x}^*||^2\right) \\ &= \frac{\gamma}{2}||\boldsymbol{g}_t||^2 + \frac{1}{2\gamma}\left(||\boldsymbol{x}_t - \boldsymbol{x}^*||^2 - ||\boldsymbol{x}_{t+1} - \boldsymbol{x}^*||^2\right) \text{(BA1)}\end{aligned}$$

Basic convergence Analysis (Proof)

6. **Telescoping Sum:** Sum the equality from $t = 0$ to $T - 1$:

$$\sum_{t=0}^{T-1} \boldsymbol{g}_t^\top (\boldsymbol{x}_t - \boldsymbol{x}^*) = \frac{\gamma}{2} \sum_{t=0}^{T-1} ||\boldsymbol{g}_t||^2 + \frac{1}{2\gamma} \sum_{t=0}^{T-1} (||\boldsymbol{x}_t - \boldsymbol{x}^*||^2 - ||\boldsymbol{x}_{t+1} - \boldsymbol{x}^*||^2)$$

The second sum collapses to: $||\boldsymbol{x}_0 - \boldsymbol{x}^*||^2 - ||\boldsymbol{x}_T - \boldsymbol{x}^*||^2$

Basic convergence Analysis (Proof)

6. **Telescoping Sum:** Sum the equality from $t = 0$ to $T - 1$:

$$\sum_{t=0}^{T-1} \boldsymbol{g}_t^\top (\boldsymbol{x}_t - \boldsymbol{x}^*) = \frac{\gamma}{2} \sum_{t=0}^{T-1} ||\boldsymbol{g}_t||^2 + \frac{1}{2\gamma} \sum_{t=0}^{T-1} (||\boldsymbol{x}_t - \boldsymbol{x}^*||^2 - ||\boldsymbol{x}_{t+1} - \boldsymbol{x}^*||^2)$$

   The second sum collapses to: $||\boldsymbol{x}_0 - \boldsymbol{x}^*||^2 - ||\boldsymbol{x}_T - \boldsymbol{x}^*||^2$

7. **Upper Bound:** Since $||\boldsymbol{x}_T - \boldsymbol{x}^*||^2 \geq 0$, we can drop this term:

$$\sum_{t=0}^{T-1} \boldsymbol{g}_t^\top (\boldsymbol{x}_t - \boldsymbol{x}^*) \leq \frac{\gamma}{2} \sum_{t=0}^{T-1} ||\boldsymbol{g}_t||^2 + \frac{1}{2\gamma} ||\boldsymbol{x}_0 - \boldsymbol{x}^*||^2$$

Basic convergence Analysis (Proof)

6. **Telescoping Sum:** Sum the equality from $t = 0$ to $T - 1$:

$$\sum_{t=0}^{T-1} \boldsymbol{g}_t^\top (\boldsymbol{x}_t - \boldsymbol{x}^*) = \frac{\gamma}{2} \sum_{t=0}^{T-1} ||\boldsymbol{g}_t||^2 + \frac{1}{2\gamma} \sum_{t=0}^{T-1} (||\boldsymbol{x}_t - \boldsymbol{x}^*||^2 - ||\boldsymbol{x}_{t+1} - \boldsymbol{x}^*||^2)$$

   The second sum collapses to: $||\boldsymbol{x}_0 - \boldsymbol{x}^*||^2 - ||\boldsymbol{x}_T - \boldsymbol{x}^*||^2$

7. **Upper Bound:** Since $||\boldsymbol{x}_T - \boldsymbol{x}^*||^2 \geq 0$, we can drop this term:

$$\sum_{t=0}^{T-1} \boldsymbol{g}_t^\top (\boldsymbol{x}_t - \boldsymbol{x}^*) \leq \frac{\gamma}{2} \sum_{t=0}^{T-1} ||\boldsymbol{g}_t||^2 + \frac{1}{2\gamma} ||\boldsymbol{x}_0 - \boldsymbol{x}^*||^2$$

8. **Final Result:** Use $f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*) \leq \boldsymbol{g}_t^\top (\boldsymbol{x}_t - \boldsymbol{x}^*)$:

$$\sum_{t=0}^{T-1} (f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)) \leq \frac{\gamma}{2} \sum_{t=0}^{T-1} ||\boldsymbol{g}_t||^2 + \frac{1}{2\gamma} ||\boldsymbol{x}_0 - \boldsymbol{x}^*||^2$$

Convergence: Lipschitz Convex Functions

Theorem

Let f be convex and differentiable. Assume:

- $||\boldsymbol{x}_0 - \boldsymbol{x}^*|| \leq R$ *(bounded initial distance)*
- $||\nabla f(\boldsymbol{x})|| \leq B$ *for all* $\boldsymbol{x}$ *(B-Lipschitz / bounded gradients)*

By choosing a constant step size $\gamma := \frac{R}{B\sqrt{T}}$*, after* T *iterations:*

$$\frac{1}{T}\sum_{t=0}^{T-1}(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)) \leq \frac{RB}{\sqrt{T}}$$

- This implies the error of the *best* iterate $f(\boldsymbol{x}_{best}) - f(\boldsymbol{x}^*)$ is $O(1/\sqrt{T})$.
- To guarantee $f(\boldsymbol{x}_{best}) - f(\boldsymbol{x}^*) \leq \epsilon$, we need $T \geq \frac{R^2B^2}{\epsilon^2}$ iterations.
- This is a $O(1/\epsilon^2)$ convergence rate.

Lipschitz Convex Functions. Proof

1. **Start:** Begin with the "basic analysis" bound:

$$\sum_{t=0}^{T-1}(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)) \leq \frac{\gamma}{2}\sum_{t=0}^{T-1}||\boldsymbol{g}_t||^2 + \frac{1}{2\gamma}||\boldsymbol{x}_0 - \boldsymbol{x}^*||^2$$

2. **Apply Assumptions:** Use the bounds $||\boldsymbol{g}_t|| \leq B$ and $||\boldsymbol{x}_0 - \boldsymbol{x}^*|| \leq R$.

$$\sum_{t=0}^{T-1}(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)) \leq \frac{\gamma}{2}\sum_{t=0}^{T-1}B^2 + \frac{1}{2\gamma}R^2$$

$$\sum_{t=0}^{T-1}(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)) \leq \frac{\gamma B^2 T}{2} + \frac{R^2}{2\gamma}$$

Lipschitz Convex Functions. Proof

3. **Optimize γ:** We want to choose γ to minimize the right-hand side, $q(\gamma)$. We find the minimum by taking the derivative and setting it to 0:
$$q'(\gamma) = \frac{B^2 T}{2} - \frac{R^2}{2\gamma^2} = 0 \implies \gamma^2 = \frac{R^2}{B^2 T}$$
The optimal step size is $\gamma^* = \frac{R}{B\sqrt{T}}$.
4. **Substitute γ^*:** Plug $\gamma = \frac{R}{B\sqrt{T}}$ back into the right-hand side $q(\gamma)$:
$$\begin{aligned} q(\gamma^*) &= \frac{1}{2}\left(\frac{R}{B\sqrt{T}}\right) B^2 T + \frac{1}{2}\left(\frac{B\sqrt{T}}{R}\right) R^2 \\ &= \frac{RB\sqrt{T}}{2} + \frac{RB\sqrt{T}}{2} = RB\sqrt{T} \end{aligned}$$

Lipschitz Convex Functions. Proof

5. **Result:**

$$\sum_{t=0}^{T-1}(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)) \leq RB\sqrt{T}$$

6. **Average Error:** Divide by T to get the average error:

$$\frac{1}{T}\sum_{t=0}^{T-1}(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)) \leq \frac{RB}{\sqrt{T}}$$

7. **Implication:** The average error (and thus the best error) decreases as $O(1/\sqrt{T})$.

Convergence: Smooth Convex Functions

The L-smoothness assumption is stronger and gives a better rate.

Lemma (Sufficient decrease)

Let f be L-smooth. By choosing $\gamma = 1/L$:

$$f(\boldsymbol{x}_{t+1}) \leq f(\boldsymbol{x}_t) - \frac{1}{2L}||\nabla f(\boldsymbol{x}_t)||^2$$

Meaning: *With $\gamma = 1/L$, every single step is guaranteed to decrease the function value.*

Smooth Convex Functions. Proof

Proof.

From the L-smoothness upper bound:

$$f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) + \nabla f(\mathbf{x}_t)^\top (\mathbf{x}_{t+1} - \mathbf{x}_t) + \frac{L}{2}||\mathbf{x}_{t+1} - \mathbf{x}_t||^2$$

$$\text{Substitute } \mathbf{x}_{t+1} - \mathbf{x}_t = -\frac{1}{L}\nabla f(\mathbf{x}_t):$$

$$\leq f(\mathbf{x}_t) + \nabla f(\mathbf{x}_t)^\top \left(-\frac{1}{L}\nabla f(\mathbf{x}_t)\right) + \frac{L}{2}\left\|-\frac{1}{L}\nabla f(\mathbf{x}_t)\right\|^2$$

$$\leq f(\mathbf{x}_t) - \frac{1}{L}||\nabla f(\mathbf{x}_t)||^2 + \frac{L}{2L^2}||\nabla f(\mathbf{x}_t)||^2$$

$$\leq f(\mathbf{x}_t) - \frac{1}{L}||\nabla f(\mathbf{x}_t)||^2 + \frac{1}{2L}||\nabla f(\mathbf{x}_t)||^2$$

$$= f(\mathbf{x}_t) - \frac{1}{2L}||\nabla f(\mathbf{x}_t)||^2$$

□

Convergence: Smooth Convex Functions

Theorem

Let f be convex and L-smooth. Choosing $\gamma = 1/L$:

$$f(\mathbf{x}_T) - f(\mathbf{x}^*) \leq \frac{L}{2T} ||\mathbf{x}_0 - \mathbf{x}^*||^2$$

- This is a much better result. The error decreases as $O(1/T)$.
- To guarantee $f(\mathbf{x}_T) - f(\mathbf{x}^*) \leq \epsilon$, we need $T \geq \frac{LR^2}{2\epsilon}$ iterations (where $R = ||\mathbf{x}_0 - \mathbf{x}^*||$).
- This is a $O(1/\epsilon)$ convergence rate.

Smooth Convex Functions. Proof I

1. **Start:** Begin with the basic result using $\gamma = 1/L$:

$$\sum_{t=0}^{T-1}(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)) \leq \frac{1}{2L}\sum_{t=0}^{T-1}||\boldsymbol{g}_t||^2 + \frac{L}{2}||\boldsymbol{x}_0 - \boldsymbol{x}^*||^2$$

2. **Bound Gradients:** Use the Sufficient Decrease Lemma:

$$\frac{1}{2L}||\boldsymbol{g}_t||^2 \leq f(\boldsymbol{x}_t) - f(\boldsymbol{x}_{t+1})$$

Sum this inequality from $t = 0$ to $T - 1$:

$$\frac{1}{2L}\sum_{t=0}^{T-1}||\boldsymbol{g}_t||^2 \leq \sum_{t=0}^{T-1}(f(\boldsymbol{x}_t) - f(\boldsymbol{x}_{t+1}))$$

Smooth Convex Functions. Proof II

3. **Telescoping Sum:** The right-hand side collapses:

$$\sum_{t=0}^{T-1} (f(\boldsymbol{x}_t) - f(\boldsymbol{x}_{t+1})) = f(\boldsymbol{x}_0) - f(\boldsymbol{x}_T)$$

   So, $\frac{1}{2L} \sum_{t=0}^{T-1} ||\boldsymbol{g}_t||^2 \leq f(\boldsymbol{x}_0) - f(\boldsymbol{x}_T)$.

4. **Substitute:** Plug this bound for the gradient sum back into Step 1:

$$\sum_{t=0}^{T-1} (f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)) \leq (f(\boldsymbol{x}_0) - f(\boldsymbol{x}_T)) + \frac{L}{2} ||\boldsymbol{x}_0 - \boldsymbol{x}^*||^2$$

Smooth Convex Functions. Proof III

5. **Rearrange:** Add $f(\boldsymbol{x}_T) - f(\boldsymbol{x}_0)$ to both sides:

$$\sum_{t=0}^{T-1}(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)) - (f(\boldsymbol{x}_0) - f(\boldsymbol{x}_T)) \le \frac{L}{2}||\boldsymbol{x}_0 - \boldsymbol{x}^*||^2$$

This simplifies to:

$$\sum_{t=1}^{T}(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)) \le \frac{L}{2}||\boldsymbol{x}_0 - \boldsymbol{x}^*||^2$$

6. **Monotonicity:** From Sufficient Decrease, $f(\boldsymbol{x}_{t+1}) \le f(\boldsymbol{x}_t)$. Thus $f(\boldsymbol{x}_T)$ is the smallest value in the sequence $f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_T)$.

$$T(f(\boldsymbol{x}_T) - f(\boldsymbol{x}^*)) \le \sum_{t=1}^{T}(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*))$$

Smooth Convex Functions. Proof IV

7. **Combine:** Combining steps 5 and 6:

$$T(f(\boldsymbol{x}_T) - f(\boldsymbol{x}^*)) \leq \frac{L}{2}||\boldsymbol{x}_0 - \boldsymbol{x}^*||^2$$

$$\Downarrow$$

$$f(\boldsymbol{x}_T) - f(\boldsymbol{x}^*) \leq \frac{L}{2T}||\boldsymbol{x}_0 - \boldsymbol{x}^*||^2$$

Convergence: Smooth & Strongly Convex

This is the "best" class of functions for standard GD.

Theorem

Let f be L-smooth and μ-strongly convex ($\mu > 0$). Choosing $\gamma = 1/L$:

(i) (Distance): *The distance to the optimum decreases geometrically:*

$$||\boldsymbol{x}_{t+1} - \boldsymbol{x}^*||^2 \leq \left(1 - \frac{\mu}{L}\right) ||\boldsymbol{x}_t - \boldsymbol{x}^*||^2$$

(ii) (Function Value): *The function error decreases exponentially:*

$$f(\boldsymbol{x}_T) - f(\boldsymbol{x}^*) \leq \frac{L}{2} \left(1 - \frac{\mu}{L}\right)^T ||\boldsymbol{x}_0 - \boldsymbol{x}^*||^2$$

- Let $\kappa = L/\mu$ be the *condition number*.
- This is **linear convergence** (or geometric convergence).
- The number of steps is $T = O(\kappa \log(1/\epsilon))$.

1. **Start:** We start from the equation derived from the basic analysis plus strong convexity:

$$\boldsymbol{g}_t^\top(\boldsymbol{x}_t - \boldsymbol{x}^*) = \frac{1}{2\gamma}\left(\gamma^2||\boldsymbol{g}_t||^2 + ||\boldsymbol{x}_t - \boldsymbol{x}^*||^2 - ||\boldsymbol{x}_{t+1} - \boldsymbol{x}^*||^2\right)$$
$$= \frac{\gamma}{2}||\boldsymbol{g}_t||^2 + \frac{1}{2\gamma}\left(||\boldsymbol{x}_t - \boldsymbol{x}^*||^2 - ||\boldsymbol{x}_{t+1} - \boldsymbol{x}^*||^2\right) \quad \text{(BA1)}$$

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top(\boldsymbol{y} - \boldsymbol{x}) + \frac{\mu}{2}||\boldsymbol{y} - \boldsymbol{x}||^2 \quad \text{(SC1)}$$

Set $\boldsymbol{x} = \boldsymbol{x}_t, \boldsymbol{y} = \boldsymbol{x}^*$ and $\mathbf{g}_t = \nabla f(\boldsymbol{x}_t)$ then

$$\mathbf{g}_t^\top(\boldsymbol{x}_t - \boldsymbol{x}^*) \geq f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*) + \frac{\mu}{2}||\boldsymbol{x}_t - \boldsymbol{x}^*||^2 \quad \text{(SC2)}$$

Smooth & Strongly Convex. Proof II

We have

$$f(\mathbf{x}_t) - f(\mathbf{x}^*) \leq \frac{1}{2\gamma}(\gamma^2\|\nabla f(\mathbf{x}_t)\|^2 + \|\mathbf{x}_t - \mathbf{x}^*\|^2 + \|\mathbf{x}_{t+1} - \mathbf{x}^*\|^2)$$
$$-\frac{\mu}{2}\|\mathbf{x}_t - \mathbf{x}^*\|^2$$

$$||\mathbf{x}_{t+1} - \mathbf{x}^*||^2 \leq (1 - \mu\gamma)||\mathbf{x}_t - \mathbf{x}^*||^2 + \gamma^2||\mathbf{g}_t||^2 - 2\gamma(f(\mathbf{x}_t) - f(\mathbf{x}^*))$$

Let's call the last two terms the "Noise":

$$\text{Noise} = \gamma^2||\mathbf{g}_t||^2 - 2\gamma(f(\mathbf{x}_t) - f(\mathbf{x}^*))$$

Smooth & Strongly Convex. Proof III

2. **Goal:** Show that Noise ≤ 0 when $\gamma = 1/L$.
3. **Sufficient Decrease:**

$$f(\boldsymbol{x}_{t+1}) \leq f(\boldsymbol{x}_t) - \frac{1}{2L}||\boldsymbol{g}_t||^2$$

4. **Bound:** Since $\boldsymbol{x}^*$ is the minimum, $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}_{t+1})$.

$$f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}_t) - \frac{1}{2L}||\boldsymbol{g}_t||^2$$

$$\Downarrow$$

$$f(\boldsymbol{x}^*) - f(\boldsymbol{x}_t) \leq -\frac{1}{2L}||\boldsymbol{g}_t||^2$$

5. **Show Noise ≤ 0:**

$$\text{Noise} = 2\gamma(f(\boldsymbol{x}^*) - f(\boldsymbol{x}_t)) + \gamma^2||\boldsymbol{g}_t||^2$$

Substitute the inequality from Step 4 and $\gamma = 1/L$:

$$\begin{aligned}\text{Noise} &\leq 2\left(\frac{1}{L}\right)\left(-\frac{1}{2L}||\boldsymbol{g}_t||^2\right) + \left(\frac{1}{L}\right)^2||\boldsymbol{g}_t||^2 \\ &\leq -\frac{1}{L^2}||\boldsymbol{g}_t||^2 + \frac{1}{L^2}||\boldsymbol{g}_t||^2 = 0\end{aligned}$$

Smooth & Strongly Convex. Proof V

6. **Proof of (i):** Since the Noise term is ≤ 0, the inequality from Step 1 becomes:

$$||\boldsymbol{x}_{t+1} - \boldsymbol{x}^*||^2 \leq (1 - \mu\gamma)||\boldsymbol{x}_t - \boldsymbol{x}^*||^2$$

$$||\boldsymbol{x}_{t+1} - \boldsymbol{x}^*||^2 \leq \left(1 - \frac{\mu}{L}\right) ||\boldsymbol{x}_t - \boldsymbol{x}^*||^2$$

Applying this recursively gives (i):

$$||\boldsymbol{x}_T - \boldsymbol{x}^*||^2 \leq \left(1 - \frac{\mu}{L}\right)^T ||\boldsymbol{x}_0 - \boldsymbol{x}^*||^2$$

7. **Proof of (ii):** Use the L-smoothness upper bound, starting from $\boldsymbol{x}^*$:

$$f(\boldsymbol{x}_T) \leq f(\boldsymbol{x}^*) + \nabla f(\boldsymbol{x}^*)^\top (\boldsymbol{x}_T - \boldsymbol{x}^*) + \frac{L}{2}||\boldsymbol{x}_T - \boldsymbol{x}^*||^2$$

Smooth & Strongly Convex. Proof VI

8. **Simplify:** The gradient at the minimum is zero, $\nabla f(\boldsymbol{x}^*) = \mathbf{0}$.

$$f(\boldsymbol{x}_T) - f(\boldsymbol{x}^*) \leq \frac{L}{2}||\boldsymbol{x}_T - \boldsymbol{x}^*||^2$$

9. **Combine:** Substitute the result from part (i) into this inequality:

$$f(\boldsymbol{x}_T) - f(\boldsymbol{x}^*) \leq \frac{L}{2}\left(1 - \frac{\mu}{L}\right)^T ||\boldsymbol{x}_0 - \boldsymbol{x}^*||^2$$

□

Convergence Summary

Table: Gradient Descent Convergence Rates

Function Class	**Error Rate**	**Iterations T for error** ϵ
Lipschitz Convex	$O(1/\sqrt{T})$	$O(1/\epsilon^2)$
Smooth Convex	$O(1/T)$	$O(1/\epsilon)$
Smooth & Strongly Conv.	$O((1-\mu/L)^T)$	$O(\kappa \log(1/\epsilon))$

- Stronger assumptions on the function (e.g., adding smoothness, then strong convexity) lead to exponentially faster convergence guarantees.

The Line Search Challenge

The Gradient Descent update rule is:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \gamma_k \nabla f(\mathbf{x}_k)$$

- We use $\mathbf{g}_k = \nabla f(\mathbf{x}_k)$ as the gradient and $\mathbf{d}_k = -\mathbf{g}_k$ as the descent direction.
- The update becomes $\mathbf{x}_{k+1} = \mathbf{x}_k + \gamma_k \mathbf{d}_k$.

The Core Problem

How do we choose the step size γ_k at each iteration?

- **Plain GD:** Use a fixed γ (e.g., $\gamma_k = 0.01$).
  - Too small $\rightarrow$ very slow convergence.
  - Too large $\rightarrow$ overshooting, oscillation, or divergence.
- **Line Search:** Choose an γ_k intelligently at each step.

The 1D Minimization Problem

Given the current iterate $\mathbf{x}_k$ and direction $\mathbf{d}_k$, we want to find an $\gamma_k > 0$ that minimizes f along that line.

We define a new, 1-dimensional function $\phi(\gamma)$:

$$\phi(\gamma) = f(\mathbf{x}_k + \gamma \mathbf{d}_k)$$

The goal of any line search is to find a good γ_k that minimizes $\phi(\gamma)$.

Two main strategies:

1. **Exact Line Search:** Find the *exact* minimum of $\phi(\gamma)$.
2. **Inexact Line Search:** Find an γ that is "good enough".

Method 1: Exact Line Search I

Idea: Find the γ_k that perfectly minimizes the function along the search direction.

$$\gamma_k = \arg\min_{\gamma > 0} \phi(\gamma) = \arg\min_{\gamma > 0} f(\mathbf{x}_k + \gamma \mathbf{d}_k)$$

How? (Theoretically)

- We solve for $\phi'(\gamma) = 0$.
- Using the chain rule: $\phi'(\gamma) = \nabla f(\mathbf{x}_k + \gamma \mathbf{d}_k)^T \mathbf{d}_k$.
- We need to find γ such that $\nabla f(\mathbf{x}_{k+1})^T \mathbf{d}_k = 0$.
- This means the **new gradient is orthogonal** to the previous search direction.

Method 1: Exact Line Search II

Advantages:

- Makes the most progress possible in the chosen direction.
- Converges in very few iterations (especially for quadratic functions, where it produces a characteristic "zigzag" path).

Drawbacks:

- **Extremely high cost!** Solving $\arg\min_{\gamma>0}$ is often as hard as the original problem.
- Only analytically solvable for simple functions (e.g., quadratics).
- Almost never used in practice for complex problems like deep learning.

Method 2: Backtracking Line Search (Inexact)

Idea: Don't find the *perfect* γ_k. Just find one that guarantees "sufficient decrease" quickly.

Algorithm:

1. Choose parameters:
   - $\bar{\gamma} > 0$ (initial guess, e.g., $\bar{\gamma} = 1.0$)
   - $c \in (0, 1)$ (controls "sufficient decrease", e.g., $c = 10^{-4}$)
   - $\tau \in (0, 1)$ (shrink factor, e.g., $\tau = 0.5$)
2. Set $\gamma = \bar{\gamma}$
3. **While** $f(\mathbf{x}_k + \gamma \mathbf{d}_k) > f(\mathbf{x}_k) + c\gamma \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$:
   - $\gamma \leftarrow \tau\gamma$ (Shrink the step size)
4. **End While**
5. Set $\gamma_k = \gamma$

Note: The condition $f(\mathbf{x}_k + \gamma \mathbf{d}_k) \leq f(\mathbf{x}_k) + c\gamma \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$ is called the **Armijo Condition**. Since $\mathbf{d}_k = -\mathbf{g}_k$ and $\nabla f(\mathbf{x}_k)^T \mathbf{d}_k = -\|\mathbf{g}_k\|^2$, it ensures the new point is sufficiently lower than the old one.

Backtracking: Advantages Drawbacks

Advantages:

- **Practical and efficient:** Much, much cheaper per iteration than exact search. It only requires function evaluations, not solving a new optimization problem.
- **Robust:** Guarantees convergence under mild assumptions.
- **Widely used:** The "default" line search in many serious optimization packages (e.g., for L-BFGS, Newton's method).

Drawbacks:

- Requires tuning parameters (c, τ, $\bar{\gamma}$), though default values (like $c = 10^{-4}, \tau = 0.5$) work well for many problems.
- Can require several function evaluations within one iteration (in the 'while' loop), which can be costly if $f(x)$ is expensive to compute.
- May take more *total iterations* than exact search, but the *total time* is almost always far less.

Comparison: Which Line Search to Use?

Method	Cost per Iteration	Tuning	Practicality
Plain GD (Fixed γ)	Lowest (1 grad)	Requires careful γ tuning	Simple, but can be slow or unstable
Exact LS	Extremely High (Solve $\arg\min$)	None (theoretic)	**Impractical** (except for quadratics)
Backtracking	Low / Medium (Multiple f evals)	Parameters $c, \tau, \bar{\gamma}$	**Very Practical** (Good trade-off)

Table: Comparison of step size strategies.

Key Takeaway

For most optimization problems, an **inexact line search** like backtracking provides the best balance of low computational cost and robust convergence.

The Problem: Constrained Minimization

So far, we have seen *unconstrained* minimization:

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}}\, f(\mathbf{x})$$

But many real-world problems have *constraints*:

$$\underset{\mathbf{x}\in C}{\text{minimize}}\, f(\mathbf{x})$$

- $f(\mathbf{x})$ is the (convex) objective function (e.g., loss function).
- C is a closed, convex set representing our constraints.

Examples of Constraint Sets C:

- **Non-negativity:** $C = \{\mathbf{x} \mid x_i \geq 0 \text{ for all } i\}$
- **Box Constraints:** $C = \{\mathbf{x} \mid l_i \leq x_i \leq u_i\}$
- **Norm Balls:** We want to find a solution $\mathbf{x}$ with a "small" norm.
  - L_2 **Ball:** $C = \{\mathbf{x} \mid \|\mathbf{x}\|_2 \leq R\}$ (The "Unitary Ball" if $R = 1$)
  - L_1 **Ball:** $C = \{\mathbf{x} \mid \|\mathbf{x}\|_1 \leq R\}$

Why Standard Gradient Descent Fails

The standard Gradient Descent (GD) update is:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \gamma_k \nabla f(\mathbf{x}_k)$$

The Problem

Even if $\mathbf{x}_k$ is in the set C (i.e., $\mathbf{x}_k$ is *feasible*), the next step $\mathbf{x}_{k+1}$ may land **outside** of C.

We need a way to move in the direction of the negative gradient while *staying inside* the set C.

The Projection Operator: $\mathcal{P}_C$

Definition: The *projection* of a point $\mathbf{y}$ onto a convex set C, denoted $\mathcal{P}_C(\mathbf{y})$, is the point in C that is closest to $\mathbf{y}$.

$$\mathcal{P}_C(\mathbf{y}) = \arg\min_{\mathbf{x} \in C} \|\mathbf{x} - \mathbf{y}\|_2^2$$

- **If $\mathbf{y} \in C$:** The closest point is $\mathbf{y}$ itself. $\mathcal{P}_C(\mathbf{y}) = \mathbf{y}$.
- **If $\mathbf{y} \notin C$:** $\mathcal{P}_C(\mathbf{y})$ is a point on the boundary of C.

The Projected Gradient Method (PGM) Algorithm

The idea is simple: **Descend, then Project.**
At each iteration k:

1. **Gradient Step (like standard GD):** Take a step in the negative gradient direction.
$$\mathbf{y}_{k+1} = \mathbf{x}_k - \gamma_k \nabla f(\mathbf{x}_k)$$
This $\mathbf{y}_{k+1}$ is our "desired" point, but it might be infeasible.

2. **Projection Step:** Project the result $\mathbf{y}_{k+1}$ back onto the feasible set C.
$$\mathbf{x}_{k+1} = \mathcal{P}_C(\mathbf{y}_{k+1})$$

Single-line Update: $\mathbf{x}_{k+1} = \mathcal{P}_C\Big(\mathbf{x}_k - \gamma_k \nabla f(\mathbf{x}_k)\Big)$

Key Condition

This method is only efficient if the projection $\mathcal{P}_C(\mathbf{y})$ is **easy to compute**.

Case 1: Projection onto the L_2 Ball

Let $C = \{\mathbf{x} \mid \|\mathbf{x}\|_2 \leq R\}$ (the "unitary ball" if $R = 1$).

We have $\mathbf{y} = \mathbf{x}_k - \gamma_k \nabla f(\mathbf{x}_k)$. We need to compute $\mathbf{x}_{k+1} = \mathcal{P}_C(\mathbf{y})$.

The L_2 Projection is simple "shrinking":

$$\mathcal{P}_C(\mathbf{y}) = \begin{cases} \mathbf{y} & \text{if } \|\mathbf{y}\|_2 \leq R \quad \text{(already inside)} \\ R \cdot \frac{\mathbf{y}}{\|\mathbf{y}\|_2} & \text{if } \|\mathbf{y}\|_2 > R \quad \text{(shrink to boundary)} \end{cases}$$

This can be written compactly as:

$$\mathcal{P}_C(\mathbf{y}) = \mathbf{y} \min\left(1, \frac{R}{\|\mathbf{y}\|_2}\right)$$

- This projection is very cheap to compute.
- It scales the entire vector, but *does not* change its direction.
- It does *not* create sparsity.

Case 2: Projection onto the L_1 Ball

Let $C = \{\mathbf{x} \mid \|\mathbf{x}\|_1 \leq R\}$. This is much trickier!

$$\mathcal{P}_C(\mathbf{y}) = \arg\min_{\mathbf{x}\in C} \|\mathbf{x} - \mathbf{y}\|_2^2 \quad \text{s.t.} \quad \sum_i |x_i| \leq R$$

- There is no simple, closed-form formula like the L_2 case.
- **However**, it can be computed efficiently (in $O(n \log n)$ time).
- **Crucial Property:** The L_1 projection is sparsity-inducing. It preferentially sets small components of y to exactly **zero**.

The Link: Constrained vs. Regularized

There is a deep connection in optimization (via Lagrangian duality) between a *constrained* problem and a *regularized* one.

Form 1: Constrained Problem

$$\underset{\mathbf{x} \in C}{\text{minimize}}\, f(\mathbf{x}) \quad \text{where } C = \{\mathbf{x} \mid \Omega(\mathbf{x}) \leq R\}$$

The Link: Constrained vs. Regularized

There is a deep connection in optimization (via Lagrangian duality) between a *constrained* problem and a *regularized* one.

Form 1: Constrained Problem

$$\underset{\mathbf{x}\in C}{\text{minimize}}\, f(\mathbf{x}) \quad \text{where } C = \{\mathbf{x} \mid \Omega(\mathbf{x}) \leq R\}$$

This is solved by **Projected Gradient Descent**:

$$\mathbf{x}_{k+1} = \mathcal{P}_C(\mathbf{x}_k - \gamma_k \mathbf{g}_k)$$

Form 2: Regularized Problem

$$\underset{\mathbf{x}}{\text{minimize}}\, f(V) + \lambda \cdot \Omega(\mathbf{x})$$

The Link: Constrained vs. Regularized

There is a deep connection in optimization (via Lagrangian duality) between a *constrained* problem and a *regularized* one.

Form 1: Constrained Problem

$$\underset{\mathbf{x}\in C}{\text{minimize}}\, f(\mathbf{x}) \quad \text{where } C = \{\mathbf{x} \mid \Omega(\mathbf{x}) \leq R\}$$

This is solved by **Projected Gradient Descent**:

$$\mathbf{x}_{k+1} = \mathcal{P}_C(\mathbf{x}_k - \gamma_k \mathbf{g}_k)$$

Form 2: Regularized Problem

$$\underset{\mathbf{x}}{\text{minimize}}\, f(V) + \lambda \cdot \Omega(\mathbf{x})$$

This is solved by **Proximal Gradient Descent**:

$$\mathbf{x}_{k+1} = \mathbf{prox}_{\gamma_k \lambda \Omega}(\mathbf{x}_k - \gamma_k \mathbf{g}_k)$$

For convex problems, for every $R > 0$, there exists a $\lambda \geq 0$ (and vice-versa) such that these two forms have the same solution.

PGM (L_2 Ball) vs. L_2 Regularization (Ridge)

PGM on L_2 Ball:

$$\underset{\|\mathbf{x}\|_2 \leq R}{\text{minimize}}\, f(\mathbf{x})$$

- **Update:** $\mathbf{x}_{k+1} = \mathcal{P}_{L_2(R)}(\mathbf{x}_k - \gamma_k \mathbf{g}_k)$
- **Mechanism:** "Hard" constraint. If vector is too long, it gets clipped to the boundary.

L_2 **Regularization (Ridge Regression):**

$$\underset{\mathbf{x}}{\text{minimize}}\, f(\mathbf{x}) + \lambda \|\mathbf{x}\|_2^2$$

- **Update:** $\mathbf{x}_{k+1} = \mathbf{prox}_{\gamma_k \lambda \|\cdot\|_2^2}(\mathbf{x}_k - \gamma_k \mathbf{g}_k)$
- This specific *proximal operator* is just **weight decay**:

  $$\mathbf{x}_{k+1} = (1 - \gamma_k \lambda')(\mathbf{x}_k - \gamma_k \mathbf{g}_k)$$

  (assuming $\mathbf{g}_k$ is gradient of f only)
- **Mechanism:** "Soft" penalty. All weights are shrunk (decayed) by a factor at each step.

Connection: Both methods shrink weights to control complexity.

PGM (L_1 Ball) vs. L_1 Regularization (LASSO)

This is the most important connection!

PGM on L_1 Ball:

$$\underset{\|\mathbf{x}\|_1 \leq R}{\text{minimize}}\, f(\mathbf{x})$$

- **Update:** $\mathbf{x}_{k+1} = \mathcal{P}_{L_1(R)}(\mathbf{x}_k - \gamma_k \mathbf{g}_k)$
- **Mechanism:** The L_1 projection creates sparsity by setting small components to 0.

L_1 Regularization (LASSO):

$$\underset{\mathbf{x}}{\text{minimize}}\, f(\mathbf{x}) + \lambda \|\mathbf{x}\|_1$$

- **Update:** $\mathbf{x}_{k+1} = \mathbf{prox}_{\gamma_k \lambda \|\cdot\|_1}(\mathbf{x}_k - \gamma_k \mathbf{g}_k)$
- This proximal operator is the **Soft-Thresholding Operator** S_τ!

$$[S_\tau(\mathbf{y})]_i = \text{sign}(y_i) \cdot \max(0, |y_i| - \tau)$$

  where $\tau = \gamma_k \lambda$.
- **Mechanism:** This operator also creates sparsity by setting components with $|y_i| < \tau$ to 0.

Summary

- **Projected Gradient Method (PGM)** is a simple algorithm for constrained optimization: **Descend, then Project**.

$$\mathbf{x}_{k+1} = \mathcal{P}_C(\mathbf{x}_k - \gamma_k \nabla f(\mathbf{x}_k))$$

- **PGM on L_2 Ball** ("unitary ball")
  - $\mathcal{P}_{L_2(R)}$ is a simple "scaling" or "clipping" operation.
  - It is computationally cheap.
  - It is equivalent to L_2 (Ridge) regularization, as both *shrink* weights to control complexity.
- **PGM on L_1 Ball**
  - $\mathcal{P}_{L_1(R)}$ is more complex, but still efficient.
  - It is the key link to L_1 (LASSO) regularization.
  - Both methods induce sparsity by setting irrelevant features to exactly 0.
- PGM solves the *constrained* problem, while Proximal Gradient solves the *regularized* problem. For L_1 and L_2 norms, these two forms are equivalent.