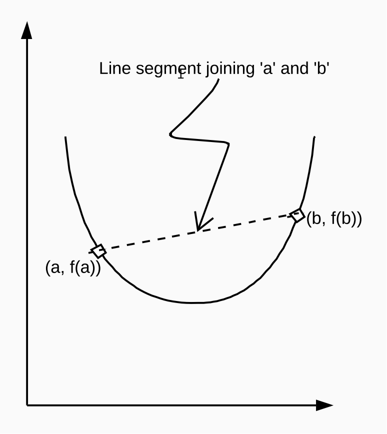
Univariate Convex Functions

Nipun Batra

October 13, 2025

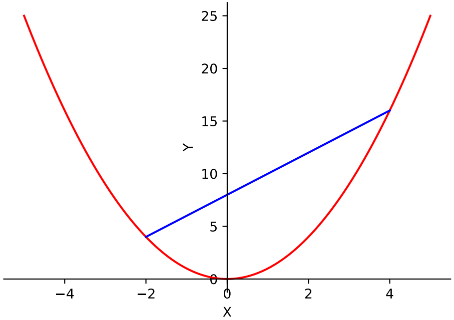
Definition

- Convexity is defined on an interval $[\alpha, \beta]$
- The line segment joining $(a, f(a))$ and $(b, f(b))$ should be *above or on* the function f for all points in interval $[\alpha, \beta]$.



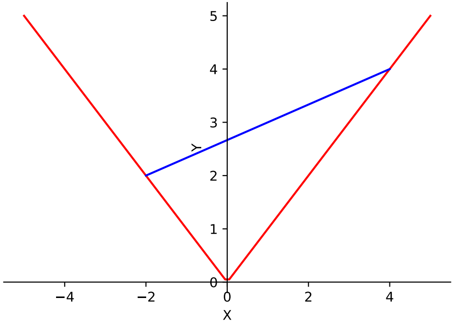
Example: $y = x^2$

Convex on the entire real line i.e. $(-\infty, \infty)$



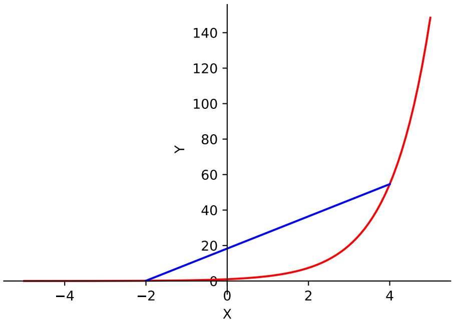
Example: $y = |x|$

Convex on the entire real line i.e. $(-\infty, \infty)$



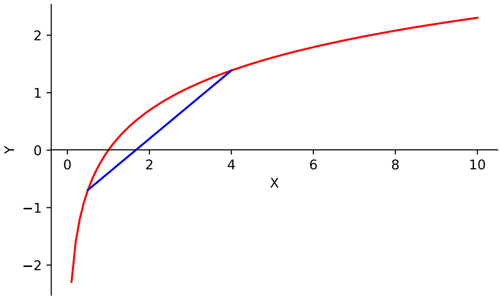
Example: $y = e^x$

Convex on the entire real line i.e. $(-\infty, \infty)$



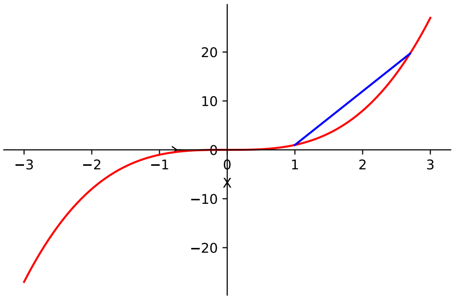
Example: $y = \ln x$

Not convex on the entire real line i.e. $(-\infty, \infty)$



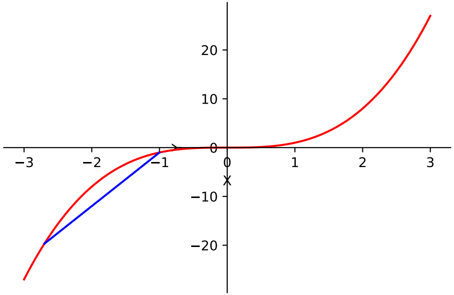
Example: $y = x^3$

It is convex for the interval $[0, \infty)$



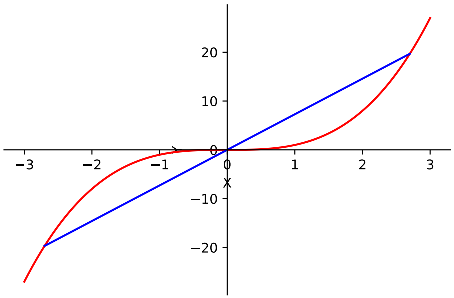
Example: $y = x^3$

It is concave for the interval $(-\infty, 0]$



Example: $y = x^3$

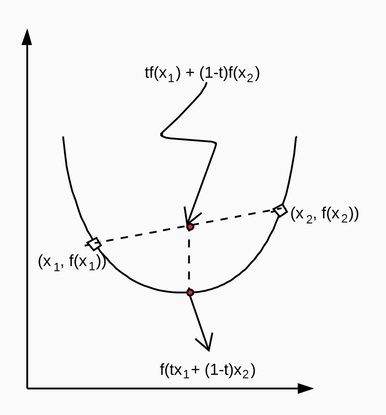
But, it is not convex for the interval $(-\infty, \infty)$



Mathematical Formulation

Function f is convex on set X, if $\forall x_1, x_2 \in X$ and $\forall t \in [0, 1]$

$$f(tx_1 + (1 - t)x_2) \leq tf(x_1) + (1 - t)f(x_2)$$



Question: Prove that $f(x) = x^2$ is convex

Question: Prove that $f(x) = x^2$ is convex

To prove:

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$$

Question: Prove that $f(x) = x^2$ is convex

To prove:

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$$

$$\text{LHS} = f(tx_1 + (1-t)x_2) \quad = t^2x_1^2 + (1-t)^2x_2^2 + 2t(1-t)x_1x_2$$
$$\text{RHS} = tf(x_1) + (1-t)f(x_2) = tx_1^2 + (1-t)x_2^2$$

Question: Prove that $f(x) = x^2$ is convex

To prove:

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$$

$$\text{LHS} = f(tx_1 + (1-t)x_2) \quad = t^2x_1^2 + (1-t)^2x_2^2 + 2t(1-t)x_1x_2$$
$$\text{RHS} = tf(x_1) + (1-t)f(x_2) = tx_1^2 + (1-t)x_2^2$$

Here,

$$\begin{aligned}\text{LHS - RHS} &= (t^2 - t)x_1^2 + [(1-t)^2 - (1-t)]x_2^2 + 2t(1-t)x_1x_2 \\ &= (t^2 - t)x_1^2 + (t^2 - t)x_2^2 - 2(t^2 - t)x_1x_2 \\ &= (t^2 - t)(x_1 - x_2)^2\end{aligned}$$

Question: Prove that $f(x) = x^2$ is convex

To prove:

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$$

$$\begin{aligned}\text{LHS} &= f(tx_1 + (1-t)x_2) &&= t^2x_1^2 + (1-t)^2x_2^2 + 2t(1-t)x_1x_2 \\ \text{RHS} &= tf(x_1) + (1-t)f(x_2) &&= tx_1^2 + (1-t)x_2^2\end{aligned}$$

Here,

$$\begin{aligned}\text{LHS - RHS} &= (t^2 - t)x_1^2 + [(1-t)^2 - (1-t)]x_2^2 + 2t(1-t)x_1x_2 \\ &= (t^2 - t)x_1^2 + (t^2 - t)x_2^2 - 2(t^2 - t)x_1x_2 \\ &= (t^2 - t)(x_1 - x_2)^2\end{aligned}$$

Here, $(t^2 - t) \leq 0$ since $t \in [0, 1]$ and $(x_1 - x_2)^2 \geq 0$

Hence, LHS -RHS ≤ 0

Hence LHS $\leq$ RHS

Hence proved.

Alternative ways to prove convexity

The Double-Derivative Test

If f''(x) > 0, the function is convex.

For example,

$\frac{\partial^2(x^2)}{\partial x^2} = 2 > 0 \Rightarrow x^2$ is a convex function.

Alternative ways to prove convexity

The double derivative test for multi-parameter function is equal to using the Hessian Matrix

A function $f(x_1, x_2, \ldots, x_n)$ is convex iff its $n \times n$ Hessian Matrix is positive semidefinite for all possible values of $(x_1, x_2, \ldots, x_n)$

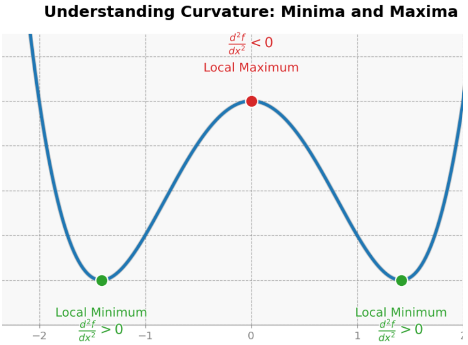
$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

Hessian Matrix Tutorial - Understanding Second-Order Derivatives for Optimization

Palak Gupta

What is Hessian Matrix?

- The **Hessian** is a square matrix containing all **second-order partial derivatives** of a scalar function $f(\mathbf{x})$.
- It's essential for understanding the **curvature** of functions in machine learning optimization.
- **Role in Optimization:** It helps to determine whether an optimization point is a minimum, maximum, or saddle point.



Hessian Matrix Definition and Formula

- For a scalar-valued function $f : \mathbb{R}^n \to \mathbb{R}$ of n variables $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$, the Hessian matrix $\mathbf{H}$ is defined as:

$$\mathbf{H}(f) = \left[\frac{\partial^2 f}{\partial x_i \partial x_j} \right]_{i,j=1}^{n}$$

- The element $\mathbf{H}_{i,j}$ of the matrix is the second-order partial derivative of f with respect to x_i and x_j.
- **Example: Hessian for Two Variables** $f(x, y)$:

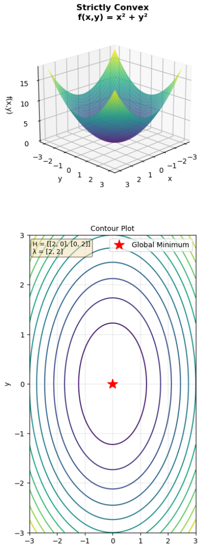
$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

- **Key Property:** If the second partial derivatives are continuous, then by **Clairaut's Theorem** (or Schwarz's theorem), the matrix is **symmetric**:

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x}$$

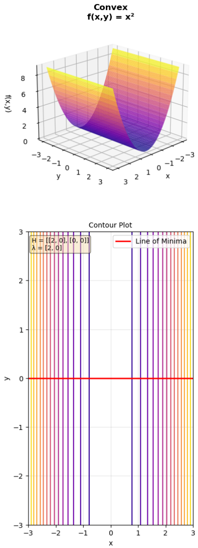
Hessian and Convexity

- **Strictly Convex:** For a point $\mathbf{x}^*$, the properties of $\mathbf{H}(\mathbf{x}^*)$ determine the nature of the function's curvature:
  - **Strictly Convex:** The Hessian $\mathbf{H}$ is **Positive Definite** ($\mathbf{H} \succ 0$).
  - **Positive Definite** $\Leftrightarrow$ **All Eigenvalues** $\lambda_i > 0$**.**
  - **Consequence:** Any local minimum is guaranteed to be the **global minimum**.
  - Optimization algorithms like Gradient Descent are guaranteed to converge to this minimum.



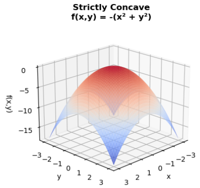
Hessian and Convexity (Cont.)

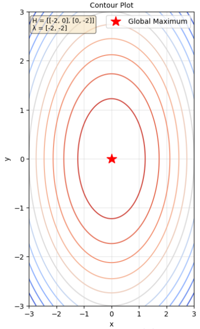
- **Convex:** The Hessian **H** is **Positive Semi-Definite** ($\mathbf{H} \succeq 0$).
- **Positive Semi-Definite** $\Leftrightarrow$ **All Eigenvalues** $\lambda_i \geq 0$.
- **Optimization Implications:**
  - The function still possesses a **global minimum**.
  - It may have **flat regions** or multiple optimal points (e.g., when $\lambda_i = 0$), leading to potential slowdowns for optimization methods.



Hessian and Concavity

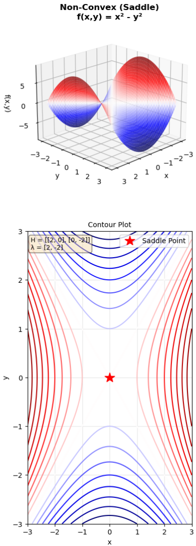
- **Strictly Concave:** The Hessian **H** is **Negative Definite** ($\mathbf{H} \prec 0$).
- **Negative Definite** $\Leftrightarrow$ **All Eigenvalues** $\lambda_i < 0$.
- **Function Shape:**
  - The function curves **downward** everywhere (inverted U-shape).
  - Any local maximum is guaranteed to be the **global maximum**.
- **Optimization Context:** This property is key for **maximization problems**, as any algorithm will converge to the unique global maximum.





Hessian and Non-Convexity

- **Non-Convex:** The Hessian **H** is **Indefinite** (neither positive nor negative semi-definite).
- **Indefinite** $\Leftrightarrow$ **Mixed Eigenvalues** (both positive $\lambda_i > 0$ and negative $\lambda_j < 0$).
- **Function Shape & Optimization Challenges:**
  - **Multiple Local Minima** are possible, complicating optimization.
  - **Saddle Points** are present, where the function curves up in some directions and down in others (Hessian is indefinite).



Hessian: Eigenvalue Summary and Calculation

Eigenvalues	Function Type	Optimization
All positive ($\lambda_i > 0$)	Strictly convex	Global minimum exists
All non-negative ($\lambda_i \geq 0$)	Convex	Minimum exists
All negative ($\lambda_i < 0$)	Strictly concave	Global maximum exists
Mixed signs	Non-convex	Saddle points present

- **Finding Eigenvalues for a 2D Hessian:**
- We solve the characteristic equation: $\det(\mathbf{H} - \lambda \mathbf{I}) = 0$.

$$\mathbf{H} = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$$

$$\det \begin{vmatrix} a-\lambda & b \\ b & c-\lambda \end{vmatrix} = 0$$

$$(a-\lambda)(c-\lambda) - b^2 = 0$$

$$\lambda^2 - \underbrace{(a+c)}_{\text{tr}(\mathbf{H})}\lambda + \underbrace{(ac-b^2)}_{\det(\mathbf{H})} = 0$$

$$\lambda_{1,2} = \frac{\text{tr}(\mathbf{H}) \pm \sqrt{\text{tr}(\mathbf{H})^2 - 4\det(\mathbf{H})}}{2}$$

Alternative Convexity Test: The Quadratic Form $\mathbf{v}^T\mathbf{H}\mathbf{v}$

- **Definition:** Convexity can be defined by the sign of the **quadratic form** $\mathbf{v}^T\mathbf{H}\mathbf{v}$.
- This method is the fundamental mathematical definition of matrix definiteness.

The Quadratic Form $\mathbf{v}^T\mathbf{H}(\mathrm{x})\mathbf{v}$ (Curvature in Direction v)

- **Convex ($\mathbf{H} \succeq 0$):**

$$\mathbf{v}^T\mathbf{H}(\mathbf{x})\mathbf{v} \geq 0$$

  for all $\mathbf{x}$ and all vectors $\mathbf{v} \neq \mathbf{0}$.

- **Strictly Convex ($\mathbf{H} \succ 0$):**

$$\mathbf{v}^T\mathbf{H}(\mathbf{x})\mathbf{v} > 0$$

  for all $\mathbf{x}$ and all vectors $\mathbf{v} \neq \mathbf{0}$.

- **Concave ($\mathbf{H} \preceq 0$):**

$$\mathbf{v}^T\mathbf{H}(\mathbf{x})\mathbf{v} \leq 0$$

  for all $\mathbf{x}$ and all vectors $\mathbf{v} \neq \mathbf{0}$.

- **Intuition:** The sign of this value indicates the function's curvature when moving from $\mathbf{x}$ in the direction $\mathbf{v}$.

Example Q1: Analysis of $f(x, y) = x^2 + y^2$ (Calculations)

- **Function:**

$$f(x, y) = x^2 + y^2$$

- **First Derivatives:**

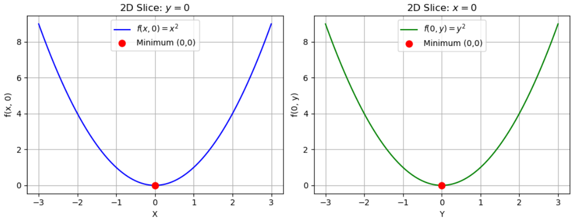
$$\frac{\partial f}{\partial x} = 2x, \quad \frac{\partial f}{\partial y} = 2y$$

- **Second Derivatives:**

$$\frac{\partial^2 f}{\partial x^2} = 2, \quad \frac{\partial^2 f}{\partial y^2} = 2, \quad \frac{\partial^2 f}{\partial x \partial y} = 0$$

- **Hessian Matrix H:**

$$\mathbf{H} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$



Example Q1: Analysis of $f(x, y) = x^2 + y^2$ (Conclusion)

- **Hessian Matrix (from previous slide):**

$$\mathbf{H} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

- **Eigenvalues λ:** The Hessian is a diagonal matrix, so the eigenvalues are the diagonal elements.

$$\lambda_1 = 2, \quad \lambda_2 = 2$$

- **Final Conclusion:**
  - Both eigenvalues are **positive** ($\lambda > 0$).
  - The function $f(x, y)$ is **Strictly Convex**.
  - The critical point (where $\nabla f = \mathbf{0}$, which is $(0, 0)$) is a **Global Minimum**.

Example Q2: Analysis of $f(x, y) = x^2 - y^2$ (Calculations)

- **Function:**

$$f(x, y) = x^2 - y^2$$

- **First Derivatives (Gradient ∇f):**

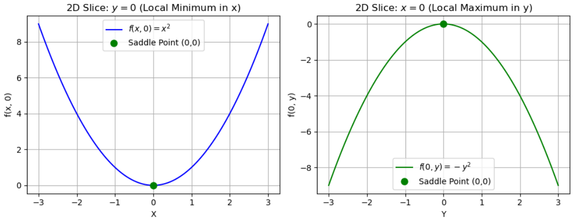
$$\frac{\partial f}{\partial x} = 2x, \quad \frac{\partial f}{\partial y} = -2y$$

- **Second Derivatives:**

$$\frac{\partial^2 f}{\partial x^2} = 2, \quad \frac{\partial^2 f}{\partial y^2} = -2, \quad \frac{\partial^2 f}{\partial x \partial y} = 0$$

- **Hessian Matrix H:**

$$\mathbf{H} = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}$$



Example Q2: Analysis of $f(x, y) = x^2 - y^2$ (Conclusion)

- **Hessian Matrix (from previous slide):**

$$\mathbf{H} = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}$$

- **Eigenvalues λ:**

$$\lambda_1 = 2, \quad \lambda_2 = -2$$

- **Mixed Signs**
  - The critical point $(0, 0)$ is a **Saddle Point** (by setting first derivatives to zero).
  - The Hessian is **Indefinite** (mixed positive and negative eigenvalues).
  - This means the critical point is **NOT** a minimum or a maximum.
  - Saddle points are **very common** in high-dimensional Machine Learning loss landscapes.
  - They can **slow down optimization algorithms** like Gradient Descent, as the gradient near the point is close to zero.

Example Q3: Rosenbrock Function (Calculations)

- **Function:** (Non-convex with a steep, curved valley)

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2$$

- **First Derivatives:**

$$\frac{\partial f}{\partial x} = -2(1 - x) - 400x(y - x^2)$$

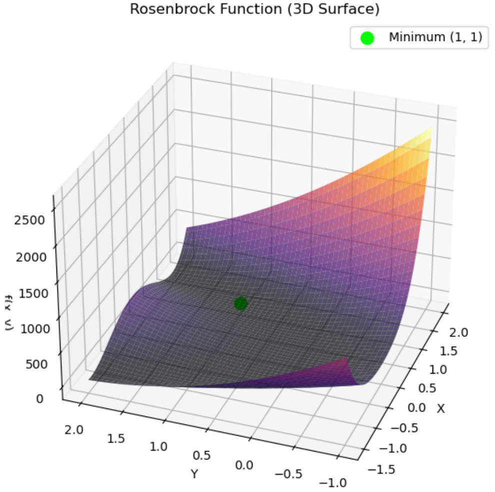
$$\frac{\partial f}{\partial y} = 200(y - x^2)$$

- **Second Derivatives:**

$$\frac{\partial^2 f}{\partial x^2} = 2 + 1200x^2 - 400y$$

$$\frac{\partial^2 f}{\partial y^2} = 200, \quad \frac{\partial^2 f}{\partial x \partial y} = -400x$$

Rosenbrock Function (3D Surface)



Example Q3: Rosenbrock Function (Hessian Analysis)

- **General Hessian Matrix $\mathbf{H}(x, y)$:**

$$\mathbf{H}(x,y) = \begin{pmatrix} 2 + 1200x^2 - 400y & -400x \\ -400x & 200 \end{pmatrix}$$

- **Hessian at Minimum $(1, 1)$:**

$$\mathbf{H}(1,1) = \begin{pmatrix} 802 & -400 \\ -400 & 200 \end{pmatrix}$$

- **Eigenvalues at Minimum:**

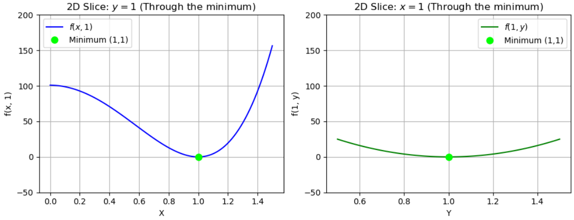
$$\lambda_1 \approx 1001.6, \quad \lambda_2 \approx 0.4$$

  - The **large difference in λ values** ($\approx 2500\times$) indicates severe ill-conditioning.
  - This means the valley is **very steep** in one direction (λ_1) and extremely **shallow** in the orthogonal direction (λ_2).

Example Q3: Rosenbrock Function (Optimization Challenge)

- The **long, narrow, parabolic valley** makes it difficult for simple Gradient Descent (GD) to navigate efficiently.
- GD takes tiny steps along the shallow direction and bounces wildly back-and-forth across the steep walls.
- **Optimization algorithms** must effectively follow the valley floor without bouncing side-to-side.

2D Slices Through the Minimum (1, 1)



Hessian Matrix: Summary and Insights

- **Generalized Form (Hessian H for $f : \mathbb{R}^n \to \mathbb{R}$):**

$$\mathbf{H} = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

- **Eigenvalue (λ) Classification:** The eigenvalues of **H** at a critical point θ_c determine the nature of that point:
  - **Local Minimum:** All $\lambda > 0$
  - **Local Maximum:** All $\lambda < 0$
  - **Saddle Point:** $\exists \lambda^+ > 0$ and $\exists \lambda^- < 0$ (Mixed signs)
- **Eigenvalue Magnitude (Curvature):** The magnitude of the eigenvalues tells us about function steepness along the corresponding eigenvector direction:
  - **Steepness:** Large $|\lambda|$
  - **Flatness:** Small $|\lambda|$

Multivariate Convex Functions

Nipun Batra

Convexity of linear least squares

Prove the convexity of linear least squares i.e. $f(\boldsymbol{\theta}) = ||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}||^2$

Convexity of linear least squares

Prove the convexity of linear least squares i.e. $f(\boldsymbol{\theta}) = ||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}||^2$

We will use the double derivative (Hessian)

Convexity of linear least squares

Prove the convexity of linear least squares i.e. $f(\boldsymbol{\theta}) = ||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}||^2$

We will use the double derivative (Hessian)

$f(\boldsymbol{\theta}) = ||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}||^2 = (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})$

$= \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{X}\boldsymbol{\theta} + \boldsymbol{\theta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\theta}$

Convexity of linear least squares

Prove the convexity of linear least squares i.e. $f(\boldsymbol{\theta}) = ||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}||^2$

We will use the double derivative (Hessian)

$f(\boldsymbol{\theta}) = ||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}||^2 = (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})$

$= \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{X}\boldsymbol{\theta} + \boldsymbol{\theta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\theta}$

$\frac{df}{d\boldsymbol{\theta}} = -2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\boldsymbol{\theta}$

(Using: $\frac{\partial}{\partial\boldsymbol{\theta}}(\mathbf{a}^T\boldsymbol{\theta}) = \mathbf{a}$ and $\frac{\partial}{\partial\boldsymbol{\theta}}(\boldsymbol{\theta}^T\mathbf{A}\boldsymbol{\theta}) = (\mathbf{A} + \mathbf{A}^T)\boldsymbol{\theta}$, with $\mathbf{X}^T\mathbf{X}$ symmetric)

Convexity of linear least squares

Prove the convexity of linear least squares i.e. $f(\boldsymbol{\theta}) = ||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}||^2$

We will use the double derivative (Hessian)

$f(\boldsymbol{\theta}) = ||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}||^2 = (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})$

$= \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{X}\boldsymbol{\theta} + \boldsymbol{\theta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\theta}$

$\frac{df}{d\boldsymbol{\theta}} = -2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\boldsymbol{\theta}$

(Using: $\frac{\partial}{\partial\boldsymbol{\theta}}(\mathbf{a}^T\boldsymbol{\theta}) = \mathbf{a}$ and $\frac{\partial}{\partial\boldsymbol{\theta}}(\boldsymbol{\theta}^T\mathbf{A}\boldsymbol{\theta}) = (\mathbf{A} + \mathbf{A}^T)\boldsymbol{\theta}$, with $\mathbf{X}^T\mathbf{X}$ symmetric)

$\frac{d^2f}{d\boldsymbol{\theta}^2} = \mathbf{H} = 2\mathbf{X}^T\mathbf{X}$

Convexity of linear least squares

Prove the convexity of linear least squares i.e. $f(\boldsymbol{\theta}) = ||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}||^2$

We will use the double derivative (Hessian)

$f(\boldsymbol{\theta}) = ||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}||^2 = (\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})$

$= \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{X}\boldsymbol{\theta} + \boldsymbol{\theta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\theta}$

$\frac{df}{d\boldsymbol{\theta}} = -2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\boldsymbol{\theta}$

(Using: $\frac{\partial}{\partial \boldsymbol{\theta}}(\mathbf{a}^T\boldsymbol{\theta}) = \mathbf{a}$ and $\frac{\partial}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}^T\mathbf{A}\boldsymbol{\theta}) = (\mathbf{A} + \mathbf{A}^T)\boldsymbol{\theta}$, with $\mathbf{X}^T\mathbf{X}$ symmetric)

$\frac{d^2 f}{d\boldsymbol{\theta}^2} = \mathbf{H} = 2\mathbf{X}^T\mathbf{X}$

$\mathbf{X}^T\mathbf{X}$ is positive semidefinite for any $\mathbf{X} \in \mathbb{R}^{m \times n}$.
Hence, linear least squares function is convex.

Properties of Convex Functions

- If $f(x)$ is convex, then $kf(x)$ is also convex, for some constant $k \geq 0$

Properties of Convex Functions

- If $f(x)$ is convex, then $kf(x)$ is also convex, for some constant $k \geq 0$
- If $f(x)$ and $g(x)$ are convex, then $f(x) + g(x)$ is also convex.

Properties of Convex Functions

- If $f(x)$ is convex, then $kf(x)$ is also convex, for some constant $k \geq 0$
- If $f(x)$ and $g(x)$ are convex, then $f(x) + g(x)$ is also convex.

Using this we can say that:

- $(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) + \boldsymbol{\theta}^T\boldsymbol{\theta}$ is convex
- $(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) + ||\boldsymbol{\theta}||_1$ is convex

Second-Order Optimization for Logistic Regression

Abhyudaya Nair

Why Second-Order Methods?

- **GD Problem:** Slow convergence in elongated valleys, struggles with different parameter scales, requires learning rate tuning
- **Second-Order Solution:** Use Hessian $\mathbf{H}$ to capture curvature information
- **Key Benefits:**
  - $\mathbf{H}^{-1}$ acts as automatic, adaptive learning rate
  - Quadratic convergence near optimum (vs linear for GD)
  - Typically 5-10 iterations vs 100s-1000s for gradient descent

Logistic Regression Setup

Model Components:

Data and Predictions:

- Binary labels: $y_i \in \{0, 1\}$
- Features: $\mathbf{x}_i \in \mathbb{R}^d$
- Sigmoid: $\sigma(z) = \frac{1}{1+e^{-z}}$
- Predictions: $\hat{y}_i = \sigma(\boldsymbol{\theta}^T \mathbf{x}_i)$

Loss Function (NLL):

$$J(\boldsymbol{\theta}) = -\sum_{i=1}^{n} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

Key Sigmoid Property:

$$\frac{d\sigma(z)}{dz} = \sigma(z)(1 - \sigma(z))$$

Newton's Method: Core Derivation

Strategy: Locally approximate $f(\theta)$ with a quadratic, jump to its minimum, repeat.

Second-Order Taylor Expansion around θ_k:

$$f_{\text{quad}}(\theta) = \underbrace{f(\theta_k)}_{\text{constant}} + \underbrace{\mathbf{g}_k^T(\theta - \theta_k)}_{\text{linear: slope}} + \underbrace{\frac{1}{2}(\theta - \theta_k)^T \mathbf{H}_k (\theta - \theta_k)}_{\text{quadratic: curvature}}$$

where $\mathbf{g}_k = \nabla f(\theta_k)$ (gradient) and $\mathbf{H}_k = \nabla^2 f(\theta_k)$ (Hessian).

Derivation Steps:

1. Take gradient: $\nabla_\theta f_{\text{quad}}(\theta) = \mathbf{g}_k + \mathbf{H}_k(\theta - \theta_k)$
2. Set to zero: $\mathbf{g}_k + \mathbf{H}_k(\theta - \theta_k) = \mathbf{0}$
3. Solve for θ: $\mathbf{H}_k(\theta - \theta_k) = -\mathbf{g}_k \Rightarrow \theta - \theta_k = -\mathbf{H}_k^{-1}\mathbf{g}_k$
4. **Newton's Update:** $\boxed{\theta_{k+1} = \theta_k - \mathbf{H}_k^{-1}\mathbf{g}_k}$

Gradient Descent: $\theta_{k+1} = \theta_k - \alpha \mathbf{g}_k$

- Needs learning rate α

Newton's Method: $\theta_{k+1} = \theta_k - \mathbf{H}_k^{-1}\mathbf{g}_k$

- $\mathbf{H}_k^{-1}$ acts as adaptive learning rate

Recall Gradient and Hessian

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \mathbf{H}_k^{-1}\mathbf{g}_k$$

What We Need:

- **Gradient vector:** $\mathbf{g} = \nabla J(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial J}{\partial \theta_1} \\ \frac{\partial J}{\partial \theta_2} \\ \vdots \\ \frac{\partial J}{\partial \theta_d} \end{bmatrix} \in \mathbb{R}^d$
- **Hessian matrix:** $\mathbf{H} = \nabla^2 J(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial^2 J}{\partial \theta_1^2} & \frac{\partial^2 J}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 J}{\partial \theta_1 \partial \theta_d} \\ \frac{\partial^2 J}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 J}{\partial \theta_2^2} & \cdots & \frac{\partial^2 J}{\partial \theta_2 \partial \theta_d} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 J}{\partial \theta_d \partial \theta_1} & \frac{\partial^2 J}{\partial \theta_d \partial \theta_2} & \cdots & \frac{\partial^2 J}{\partial \theta_d^2} \end{bmatrix} \in \mathbb{R}^{d \times d}$

Gradient Computation for Logistic Regression

Goal: Compute $\frac{\partial J}{\partial \theta_j}$ where $J(\boldsymbol{\theta}) = -\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$, $\hat{y}_i = \sigma(\boldsymbol{\theta}^T \mathbf{x}_i)$

For single sample i, use chain rule: $\frac{\partial J_i}{\partial \theta_j} = \frac{\partial J_i}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial \theta_j}$

Part 1 - Loss derivative w.r.t. prediction:

$$\frac{\partial J_i}{\partial \hat{y}_i} = \frac{\partial}{\partial \hat{y}_i}[-y_i \log(\hat{y}_i) - (1 - y_i)\log(1 - \hat{y}_i)] = -\frac{y_i}{\hat{y}_i} + \frac{1 - y_i}{1 - \hat{y}_i}$$
$$= \frac{-y_i(1 - \hat{y}_i) + (1 - y_i)\hat{y}_i}{\hat{y}_i(1 - \hat{y}_i)} = \frac{-y_i + y_i\hat{y}_i + \hat{y}_i - y_i\hat{y}_i}{\hat{y}_i(1 - \hat{y}_i)} = \frac{\hat{y}_i - y_i}{\hat{y}_i(1 - \hat{y}_i)}$$

Gradient Computation for Logistic Regression (continued)

Part 2 - Prediction derivative w.r.t. parameter:

$$\frac{\partial \hat{y}_i}{\partial \theta_j} = \frac{\partial \sigma(\boldsymbol{\theta}^T \mathbf{x}_i)}{\partial \theta_j} = \underbrace{\sigma'(\boldsymbol{\theta}^T \mathbf{x}_i)}_{=\hat{y}_i(1-\hat{y}_i)} \cdot \frac{\partial (\boldsymbol{\theta}^T \mathbf{x}_i)}{\partial \theta_j} = \hat{y}_i(1-\hat{y}_i) \cdot x_{ij}$$

Combining (beautiful cancellation!):

$$\frac{\partial J_i}{\partial \theta_j} = \frac{\hat{y}_i - y_i}{\hat{y}_i(1-\hat{y}_i)} \cdot \hat{y}_i(1-\hat{y}_i)x_{ij} = (\hat{y}_i - y_i)x_{ij}$$

Sum over all samples: $\frac{\partial J}{\partial \theta_j} = \sum_{i=1}^{n}(\hat{y}_i - y_i)x_{ij} \quad \Rightarrow \quad \boxed{\mathbf{g}(\boldsymbol{\theta}) = \mathbf{X}^T(\hat{\mathbf{y}} - \mathbf{y})}$

Hessian Computation for Logistic Regression

Goal: Compute second derivatives $H_{jk} = \frac{\partial^2 J}{\partial\theta_j \partial\theta_k}$

Starting from gradient: We know $\frac{\partial J}{\partial\theta_j} = \sum_{i=1}^{n}(\hat{y}_i - y_i)x_{ij}$

Take derivative w.r.t. θ_k:

$$H_{jk} = \frac{\partial}{\partial\theta_k}\left[\sum_{i=1}^{n}(\hat{y}_i - y_i)x_{ij}\right] = \sum_{i=1}^{n} x_{ij}\frac{\partial\hat{y}_i}{\partial\theta_k} \quad \text{(since } y_i \text{ and } x_{ij} \text{ don't depend on } \theta_k\text{)}$$

$$= \sum_{i=1}^{n} x_{ij}\cdot\hat{y}_i(1-\hat{y}_i)\cdot x_{ik} = \sum_{i=1}^{n}\hat{y}_i(1-\hat{y}_i)x_{ij}x_{ik}$$

Matrix Form: Define weight matrix $\mathbf{S} = \text{diag}(\hat{y}_1(1-\hat{y}_1), \ldots, \hat{y}_n(1-\hat{y}_n))$, then: $\boxed{\mathbf{H} = \mathbf{X}^T\mathbf{S}\mathbf{X}}$

Hessian: Key Properties

$$\boxed{\mathbf{H} = \mathbf{X}^T\mathbf{S}\mathbf{X}} \text{ where } \mathbf{S} = \text{diag}(\hat{y}_1(1-\hat{y}_1), \ldots, \hat{y}_n(1-\hat{y}_n))$$

Key Properties:

- **Positive Definite:** Since $\hat{y}_i(1-\hat{y}_i) > 0$ for $\hat{y}_i \in (0,1)$, we have $\mathbf{v}^T\mathbf{H}\mathbf{v} = (\mathbf{X}\mathbf{v})^T\mathbf{S}(\mathbf{X}\mathbf{v}) > 0$ for $\mathbf{v} \neq \mathbf{0} \Rightarrow J(\boldsymbol{\theta})$ is strictly convex!
- **Adaptive Weighting:** Weight $w_i = \hat{y}_i(1-\hat{y}_i)$ is maximal at $\hat{y}_i = 0.5$ (uncertain), minimal near $\hat{y}_i = 0$ or 1 (confident)
- **Fisher Information:** $\mathbf{H}$ equals the Fisher Information Matrix for logistic regression
- **Weighted Least Squares Form:** $\mathbf{X}^T\mathbf{S}\mathbf{X}$ appears in weighted regression problems

Newton's Method Algorithm for Logistic Regression

Complete Algorithm:

Initialize: $\boldsymbol{\theta}_0$ (e.g., $\boldsymbol{\theta}_0 = \mathbf{0}$)

for $k = 0, 1, 2, \ldots$ until convergence **do**

    Compute predictions: $\hat{\mathbf{y}}_k = \sigma(\mathbf{X}\boldsymbol{\theta}_k)$ (apply elementwise)

    Compute gradient: $\mathbf{g}_k = \mathbf{X}^T(\hat{\mathbf{y}}_k - \mathbf{y})$

    Compute weights: $\mathbf{S}_k = \text{diag}(\hat{y}_{1k}(1 - \hat{y}_{1k}), \ldots, \hat{y}_{nk}(1 - \hat{y}_{nk}))$

    Compute Hessian: $\mathbf{H}_k = \mathbf{X}^T\mathbf{S}_k\mathbf{X}$

    Solve linear system: $\mathbf{H}_k\boldsymbol{\delta}_k = -\mathbf{g}_k$ for $\boldsymbol{\delta}_k$

    Update: $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \boldsymbol{\delta}_k$

    Check convergence: if $\|\mathbf{g}_k\| < \epsilon$ or $\|\boldsymbol{\delta}_k\| < \epsilon$, stop

end for

Newton's Method: Practical Considerations

Practical Considerations:

- **Never compute $\mathbf{H}_k^{-1}$ explicitly!** Solve $\mathbf{H}_k \boldsymbol{\delta}_k = -\mathbf{g}_k$ using Cholesky decomposition (exploits $\mathbf{H}_k$ being symmetric positive definite)
- **Computational Cost per iteration:** $O(nd^2 + d^3)$ where n = samples, d = features
- **Memory:** $O(d^2)$ for storing Hessian
- **Convergence:** Typically 5-10 iterations vs 100s-1000s for gradient descent
- **When to use:** Small-to-medium d (features), need high accuracy, well-conditioned problems

Iteratively Reweighted Least Squares (IRLS) Formulation

Motivation: Rewrite Newton's update to reveal connection with weighted least squares. Starting from Newton's update:

$$\begin{aligned}\boldsymbol{\theta}_{k+1} &= \boldsymbol{\theta}_k + \mathbf{H}_k^{-1}(-\mathbf{g}_k) = \boldsymbol{\theta}_k + (\mathbf{X}^T\mathbf{S}_k\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{y} - \hat{\mathbf{y}}_k) \\ &= (\mathbf{X}^T\mathbf{S}_k\mathbf{X})^{-1}[(\mathbf{X}^T\mathbf{S}_k\mathbf{X})\boldsymbol{\theta}_k + \mathbf{X}^T(\mathbf{y} - \hat{\mathbf{y}}_k)] \\ &= (\mathbf{X}^T\mathbf{S}_k\mathbf{X})^{-1}\mathbf{X}^T[\mathbf{S}_k\mathbf{X}\boldsymbol{\theta}_k + (\mathbf{y} - \hat{\mathbf{y}}_k)]\end{aligned}$$

Define adjusted response vector: $\mathbf{z}_k = \mathbf{X}\boldsymbol{\theta}_k + \mathbf{S}_k^{-1}(\mathbf{y} - \hat{\mathbf{y}}_k)$ Then:
$\mathbf{S}_k\mathbf{X}\boldsymbol{\theta}_k + (\mathbf{y} - \hat{\mathbf{y}}_k) = \mathbf{S}_k[\mathbf{X}\boldsymbol{\theta}_k + \mathbf{S}_k^{-1}(\mathbf{y} - \hat{\mathbf{y}}_k)] = \mathbf{S}_k\mathbf{z}_k$

Final IRLS form: $\boxed{\boldsymbol{\theta}_{k+1} = (\mathbf{X}^T\mathbf{S}_k\mathbf{X})^{-1}\mathbf{X}^T\mathbf{S}_k\mathbf{z}_k}$

IRLS: Interpretation

IRLS form:

$$\boxed{\theta_{k+1} = (\mathbf{X}^T\mathbf{S}_k\mathbf{X})^{-1}\mathbf{X}^T\mathbf{S}_k\mathbf{z}_k}$$

Weighted Least Squares:

$$\boxed{\theta = (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\mathbf{y}}$$

This is exactly the **weighted least squares solution** with:

- **Weights:** $\mathbf{S}_k$ (changes each iteration - hence "iteratively reweighted")
- **Response:** $\mathbf{z}_k$ (adjusted to account for current predictions)

Interpretation: Newton's method for logistic regression = iteratively solving weighted least squares problems where weights and responses are updated based on current parameter estimates!