Newton’s Method for Optimization

🧩 Topic: **Newton’s Method for Optimization**

Main idea:

We want to find the value of θ^* that **minimizes a function** $l(\theta)$ (often the loss function).

In optimization, this is the same as solving:

$l'(\theta) = 0$

because the derivative (slope) is zero at a minimum or maximum.

✳️ Step-by-Step Breakdown of the Slide

1. Goal:

$$\theta^* = \arg\min_{\theta} l(\theta)$$

That means: find the value of θ where $l(\theta)$ is smallest.

To do this, we solve:

$$l'(\theta) = 0$$

i.e., where the slope of the loss function is zero.

---

2. Define

Let:

$g(\theta) = l'(\theta)$

Then the problem becomes solving:

$g(\theta) = 0$

This is a *root-finding* problem (finding where a function crosses zero).

3. Iterative Update Rule

Newton's method updates θ iteratively:

$$\theta_{\text{new}} = \theta_{\text{old}} - \frac{g(\theta_{\text{old}})}{g'(\theta_{\text{old}})}$$

Substitute back $g(\theta) = l'(\theta)$:

$$\theta_{\text{new}} = \theta_{\text{old}} - \frac{l'(\theta_{\text{old}})}{l''(\theta_{\text{old}})}$$

📘 Interpretation:

- $l'(\theta)$ → **gradient** (first derivative)
- $l''(\theta)$ → **curvature** (second derivative / Hessian in higher dimensions)

You stop when:

$$g(\theta) = 0 \quad \text{(i.e., convergence)}$$

4. Visual Intuition (bottom right sketch)

The plot at the bottom shows:

- $g(\theta)$ on y-axis, θ on x-axis
- We're finding where $g(\theta)$ = 0
- The line $g'(\theta)$ (the tangent slope) helps estimate how far to move θ

Each step moves closer to the zero-crossing using tangent approximation.

5. Comments on the method

✅ **Faster convergence:**

It usually reaches the minimum in fewer steps than gradient descent because it uses curvature information.

✅ **Fewer iterations:**

Because each step is more "intelligent" — it accounts for both slope and curvature.

❌ **More expensive iterations:**

Computing $l''(\theta)$ (the second derivative or Hessian matrix) can be **computationally heavy**, especially for large datasets or high-dimensional parameters.

🧠 Applying to Logistic Regression

In **logistic regression**, the cost function is **convex** (bowl-shaped), and we can use Newton's method to find the parameters θ that minimize the loss.

Here:

- $l'(\theta)$ → Gradient (based on the derivative of log-likelihood)
- $l''(\theta)$ → Hessian matrix (second derivative)
- This specific application of Newton's method is called the **Newton-Raphson method** or **Iteratively Reweighted Least Squares (IRLS)** algorithm.

🧾 Summary Table

Concept	Meaning
$l(\theta)$	Objective or loss function
$l'(\theta)$	First derivative (gradient)
$l''(\theta)$	Second derivative (curvature / Hessian)
Update rule	$\theta := \theta - \frac{l'(\theta)}{l''(\theta)}$
Use case	Faster optimization for convex problems like Logistic Regression
Trade-off	Fewer steps but higher computation cost per step

🔑 Intuitive takeaway:

> Newton’s Method improves upon Gradient Descent by adjusting the step size dynamically using curvature (second derivative).
>
> It “jumps” directly toward the optimum more efficiently — but at the cost of heavier computation.
