📌 **General Understanding**

- **Loss Function** is computed when a **single data record** is passed through the model.
- **Cost Function** is computed when a **batch of data** is passed.
- **Error Function** can be used interchangeably with **Loss/Cost Function**.
- Different problems use different types of loss functions:
  - **Regression Problems** → Squared Error, Absolute Error, Huber Loss
  - **Classification Problems** → Cross Entropy (Binary or Multi-class)

🧮 Regression Loss Functions

1. Squared Error Loss (MSE - Mean Squared Error)

Formula:

$$L = (y - \hat{y})^2 \quad \text{(for one sample)}$$

$$J = \frac{1}{t} \sum_{i=1}^{t} (y_i - \hat{y}_i)^2 \quad \text{(for batch of size } t\text{)}$$

Advantages:

- It forms a **quadratic function**: $ax^2 + bx + c$
  - When plotted, this provides a **single global minimum**—no local minima.
  - Works well with **gradient descent optimization**.
- Penalizes **large errors more heavily** by squaring them.

Disadvantages:

- **Not robust to outliers**. Since errors are squared, a single large error can dominate the loss.

2. Absolute Error Loss (MAE - Mean Absolute Error)

Formula:

$$L = |y - \hat{y}|$$

$$J = \frac{1}{t} \sum_{i=1}^{t} |y_i - \hat{y}_i|$$

Advantages:

- **More robust to outliers** compared to MSE because it doesn't square the error.

Disadvantages:

- **Computationally more difficult** (non-differentiable at 0).
- Can lead to **multiple local minima**, which makes optimization harder.

3. Huber Loss

A **hybrid** of MSE and MAE. Introduces a hyperparameter δ to control transition from MSE to MAE.

Formula:

$$\text{Loss} = \begin{cases} \frac{1}{2}(y - \hat{y})^2, & \text{if } |y - \hat{y}| \leq \delta \\ \delta \cdot |y - \hat{y}| - \frac{1}{2}\delta^2, & \text{otherwise} \end{cases}$$

Key Points:

- Uses **MSE for small errors** (quadratic).
- Uses **MAE for large errors** (linear).
- Handles **outliers** better than MSE.
- Avoids the **local minima** issue of MAE.
- δ is a **hyperparameter** to tune.

🧠 Classification Loss Functions

4. Binary Cross Entropy Loss

Used in **binary classification** problems.

Formula:

$$\text{Loss} = -y\log(\hat{y}) - (1-y)\log(1-\hat{y})$$

Output Activation:

- Uses **Sigmoid Activation:**

$$\hat{y} = \frac{1}{1+e^{-z}}$$

Behavior:

- When $y = 0$ → Loss = $-\log(1-\hat{y})$
- When $y = 1$ → Loss = $-\log(\hat{y})$

5. Multi-class Cross Entropy Loss

Used for **multi-class classification** problems.

Formula:

$$L(x_i, y_i) = -\sum_{j=1}^{c} y_{ij}\log(\hat{y}_{ij})$$

Key Concepts:

- **One-hot Encoding**: Converts class labels into vectors.
  - e.g., "Good" = $[1, 0, 0]$, "Bad" = $[0, 1, 0]$
- $y_{ij} = 1$ if the class is correct, 0 otherwise.

Output Activation:

- Uses **Softmax Activation**:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}$$

- Converts raw model outputs (logits) into **probabilities** that sum to 1.

✅ **Summary Table**

Problem Type	Loss Function	Robust to Outliers	Local Minima	Differentiability	Notes
Regression	Squared Error (MSE)	❌ No	✅ No	✅ Yes	Simple, global minimum
Regression	Absolute Error (MAE)	✅ Yes	❌ Yes	❌ No at 0	Harder to optimize
Regression	Huber Loss	✅ Yes	✅ No	✅ Yes	Best of both MSE & MAE
Classification	Binary Cross Entropy	✅ Yes	✅ No	✅ Yes	For binary tasks
Classification	Multi-class Cross Entropy	✅ Yes	✅ No	✅ Yes	Uses Softmax