Day 3 – Deep Learning Core Math Concepts

Internship: RoboAI Hub
Date: June 10, 2025

1. Vectors & Matrices

- **Vectors:** One-dimensional arrays that represent input features or weights for a single unit.
- **Matrices:** Two-dimensional arrays used to store weights or batches of data.
- **Use in DL:** Represent data and parameters, and enable batch processing through efficient linear algebra operations.
- **Example:** If X is an input vector and W is a weight matrix, the layer output is $Y = XW$.

2. Dot Product & Matrix Multiplication

Dot Product

- Formula: $a \cdot b = \sum_{i=1}^{n} a_i b_i$
- Outputs a scalar and measures the similarity between two vectors.

Matrix Multiplication

- The result is a matrix where each element is a dot product between a row from the first matrix and a column from the second.
- Used in **forward propagation** to calculate neuron outputs.
- **Example:** $Z = XW + b$ where X = input, W = weights, b = bias.

3. Activation Functions

- Introduce **non-linearity** to the model.

ReLU (Rectified Linear Unit)

- Formula: $f(x) = \max(0, x)$
- Zeroes out negative values; fast and widely used.

Sigmoid

- Formula: $f(x) = \frac{1}{1+e^{-x}}$
- Maps input to the (0, 1) range; good for binary classification.

Tanh (Hyperbolic Tangent)

- Formula: $f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$
- Maps input to the (-1, 1) range; better centered than sigmoid.

4. Loss Functions

- Loss functions measure how well the model predictions match the true labels.

Mean Squared Error (MSE)

- Formula: $\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
- Common in regression tasks; penalizes larger errors more heavily.

Cross-Entropy Loss

- Formula (Binary): $L = -[y \log(\hat{y}) + (1-y)\log(1-\hat{y})]$
- Used for classification problems; punishes confident incorrect predictions.