G.RAKSHANA

Artificial Neural Network (ANN) Documentation

1. Activation Functions in ANN

a. Sigmoid Activation Function

Formula: f(x) = 1 / (1 + e^(-x))

Output range: (0,1)

Used in binary classification problems.

Disadvantage: Causes vanishing gradient problems.

b. Tanh (Hyperbolic Tangent) Activation Function

Formula: f(x) = (e^x - e^(-x)) / (e^x + e^(-x))

Output range: (-1,1)

Better than Sigmoid but still suffers from vanishing gradient issues.

c. ReLU (Rectified Linear Unit)

Formula: f(x) = max(0, x)

Output range: [0, infinity)

Helps with vanishing gradient issues.

Can suffer from 'dying ReLU' problem.

d. Leaky ReLU

Formula: f(x) = max(0.01x, x)

Similar to ReLU but allows small negative values.

e. Softmax Activation Function

Formula: f(x_i) = e^(x_i) / sum(e^(x_j))

Used in multi-class classification.

Converts outputs into probability distributions.

2. Types of Layers in ANN

a. Input Layer

The first layer in ANN. Receives raw input data.

b. Hidden Layer(s)

Intermediate layers between input and output.

Extracts features and learns patterns.

c. Output Layer

The final layer that provides predictions.

Uses appropriate activation functions.

3. Types of Optimizers

a. Gradient Descent (GD)

Updates weights in the direction of the steepest loss reduction. Slow for large datasets.

b. Stochastic Gradient Descent (SGD)

Updates weights after each training sample. Faster but more noisy updates.

c. Mini-Batch Gradient Descent

Combines GD and SGD by updating weights in small batches. More stable than SGD.

d. Adam (Adaptive Moment Estimation)

Combines momentum and RMSprop for adaptive learning rates. Widely used optimizer.

e. RMSprop (Root Mean Square Propagation)

Adjusts learning rates for different parameters to prevent oscillations.

f. Adagrad (Adaptive Gradient Algorithm)

Adjusts learning rates individually for parameters. Slows down learning over time.

4. Types of Loss Functions

a. Mean Squared Error (MSE)

Formula: MSE = (1/n) * sum((y_true - y_pred)^2)

Used in regression tasks. Penalizes large errors.

b. Mean Absolute Error (MAE)

Formula: MAE = (1/n) * sum(|y_true - y_pred|)

Measures absolute differences.

c. Binary Cross-Entropy

Formula: L = -(1/n) * sum(y * log(p) + (1 - y) * log(1 - p))

Used for binary classification.

d. Categorical Cross-Entropy

Formula: L = -sum(y_true * log(y_pred))

Used for multi-class classification.

e. Hinge Loss

Used in SVM-based classification.