■ FÓRMULAS MATEMÁTICAS E EXEMPLOS NUMÉRICOS EM DEEP LEARNING

Este documento apresenta as principais fórmulas matemáticas usadas em Deep Learning, incluindo exemplos numéricos e implementações práticas em Python para facilitar a compreensão.

1■■ Função de Ativação Sigmoid

Fórmula Matemática: σ(x) = 1 / (1 + e^(-x))

Exemplo Numérico: Se x = 2, então σ(2) = 1 / (1 + e^(-2)) ≈ 0.88

Implementação em Python: ```python import math x = 2 sigmoid = 1 / (1 + math.exp(-x)) print(sigmoid) # 0.88 ```

2■■ Função de Ativação ReLU

Fórmula Matemática: f(x) = max(0, x)

Exemplo Numérico: Se x = -3 → f(-3) = 0 Se x = 4 → f(4) = 4

Implementação em Python: ```python def relu(x): return max(0, x) print(relu(-3), relu(4)) # 0 4 ```

3■■ Função Softmax

Fórmula Matemática: Softmax(x■) = e^(x■) / Σ(e^(x■))

Exemplo Numérico: Para x = [1, 2, 3], Softmax(3) = e^3 / (e^1 + e^2 + e^3) ≈ 0.665

Implementação em Python: ```python import numpy as np x = np.array([1, 2, 3]) softmax = np.exp(x) / np.sum(np.exp(x)) print(softmax) ```

4■■ Função de Custo - MSE (Mean Squared Error)

Fórmula Matemática: MSE = (1/n) Σ (y_pred - y_true)²

Exemplo Numérico: y_true = [2, 3], y_pred = [2.5, 2.7] MSE = ((2.5-2)² + (2.7-3)²) / 2 = (0.25 + 0.09)/2 = 0.17

Implementação em Python: ```python import numpy as np y_true = np.array([2, 3]) y_pred = np.array([2.5, 2.7]) mse = np.mean((y_pred - y_true)**2) print(mse) # 0.17 ```

5■■ Função de Perda - Entropia Cruzada (Cross-Entropy)

Fórmula Matemática: L = -Σ [y * log(y■)]

Exemplo Numérico: y = [1, 0], y■ = [0.9, 0.1] L = -(1 * log(0.9) + 0 * log(0.1)) = 0.105

Implementação em Python: ```python import numpy as np y = np.array([1, 0]) y_hat = np.array([0.9, 0.1]) loss = -np.sum(y * np.log(y_hat)) print(loss) # 0.105 ```

6■■ Gradiente Descendente

Fórmula Matemática: $\theta = \theta - \alpha * \partial J/\partial \theta$

Exemplo Numérico: Se $\theta = 2$, $\alpha = 0.1$ e $\partial J/\partial \theta = 0.5 \rightarrow \theta = 2 - 0.1 * 0.5 = 1.95$

Implementação em Python: ```python theta = 2 alpha = 0.1 grad = 0.5 theta = theta - alpha * grad print(theta) # 1.95 ```

7■■ Normalização (Min-Max Scaling)

Fórmula Matemática: x_norm = (x - x_min) / (x_max - x_min)

Exemplo Numérico: x = 7, x_min = 2, x_max = 10 x_norm = (7 - 2) / (10 - 2) = 0.625

Implementação em Python: ```python x, x_min, x_max = 7, 2, 10 x_norm = (x - x_min) / (x_max - x_min) print(x_norm) # 0.625 ```

8■■ Regularização L2 (Ridge)

Fórmula Matemática: $L = \Sigma(y\_pred - y\_true)^2 + \lambda\Sigma\theta^2$

Exemplo Numérico: Erro = 0.2, $\lambda = 0.1$, $\theta = [0.5, 0.3]$ $L = 0.2 + 0.1 * (0.5^2 + 0.3^2) = 0.2 + 0.1 * 0.34 = 0.234$

Implementação em Python: ```python import numpy as np error = 0.2 lmbd = 0.1 theta = np.array([0.5, 0.3]) loss = error + lmbd * np.sum(theta**2) print(loss) # 0.234 ```

■ Conclusão

Essas são as fórmulas e exemplos mais fundamentais em Deep Learning. Dominar esses conceitos é essencial para compreender redes neurais e ajustar modelos de forma eficiente.