🧠 Retropropagación

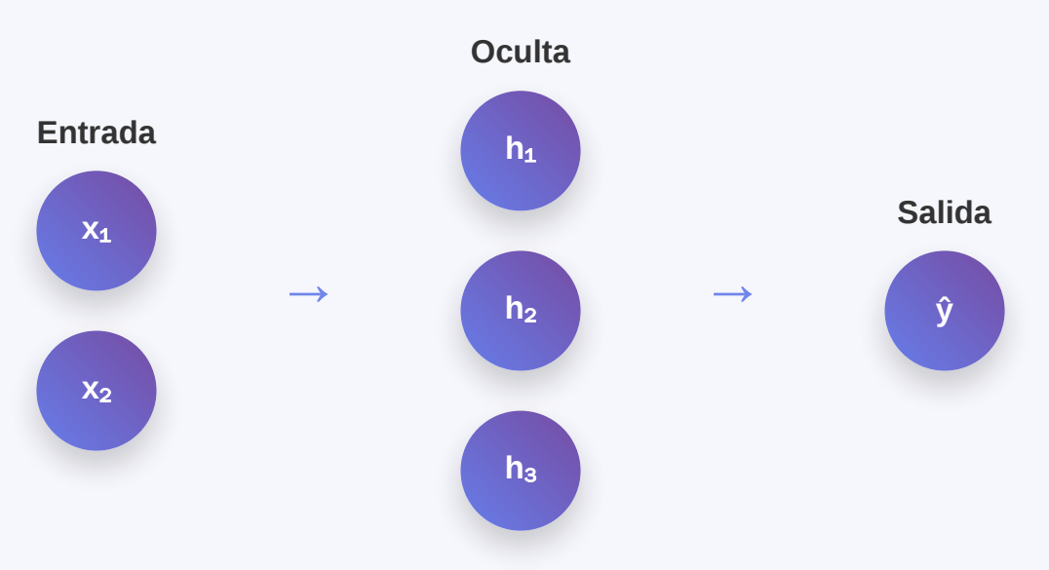
Demostración Matemática con Derivadas Parciales

📚 Contenido de la Presentación

- **Fundamentos teóricos** de la retropropagación
- **Derivación matemática completa** usando cálculo multivariable
- **Ejemplo práctico** con red neuronal simple
- **Implementación algorítmica** paso a paso

🎯 ¿Qué es la Retropropagación?

La **retropropagación** es el algoritmo fundamental para entrenar redes neuronales artificiales. Utiliza la **regla de la cadena** del cálculo diferencial para calcular eficientemente las derivadas parciales de la función de pérdida con respecto a todos los parámetros de la red.



🔑 Idea Clave

Propagamos el error desde la salida hacia la entrada, calculando cómo cada peso contribuye al error total.

🔢 Formulación Matemática Completa

1. Definición de la Red Neuronal

Capa oculta:

$$z_j^{(1)} = \sum_{i=1}^{n} w_{ji}^{(1)} x_i + b_j^{(1)}$$

$$a_j^{(1)} = \sigma(z_j^{(1)})$$

Capa de salida:

$$z_k^{(2)} = \sum_{j=1}^{m} w_{kj}^{(2)} a_j^{(1)} + b_k^{(2)}$$

$$a_k^{(2)} = \sigma(z_k^{(2)}) = \hat{y}_k$$

2. Función de Pérdida

$$L = \frac{1}{2}\sum_{k=1}^{K}(y_k - \hat{y}_k)^2 = \frac{1}{2}\sum_{k=1}^{K}(y_k - a_k^{(2)})^2$$

3. Derivadas Parciales - Capa de Salida

Gradiente respecto a pesos de salida:

$$\frac{\partial L}{\partial w_{kj}^{(2)}} = \frac{\partial L}{\partial a_k^{(2)}} \cdot \frac{\partial a_k^{(2)}}{\partial z_k^{(2)}} \cdot \frac{\partial z_k^{(2)}}{\partial w_{kj}^{(2)}}$$

Donde:

$$\frac{\partial L}{\partial a_k^{(2)}} = -(y_k - a_k^{(2)})$$

$$\frac{\partial a_k^{(2)}}{\partial z_k^{(2)}} = \sigma'(z_k^{(2)})$$

$$\frac{\partial z_k^{(2)}}{\partial w_{kj}^{(2)}} = a_j^{(1)}$$

Resultado:

$$\frac{\partial L}{\partial w_{kj}^{(2)}} = -(y_k - a_k^{(2)}) \cdot \sigma'(z_k^{(2)}) \cdot a_j^{(1)}$$

🔄 Retropropagación a la Capa Oculta

4. Derivadas Parciales - Capa Oculta

Gradiente respecto a pesos ocultos (aplicando regla de la cadena):

$$\frac{\partial L}{\partial w_{ji}^{(1)}} = \sum_{k=1}^{K} \frac{\partial L}{\partial a_k^{(2)}} \cdot \frac{\partial a_k^{(2)}}{\partial z_k^{(2)}} \cdot \frac{\partial z_k^{(2)}}{\partial a_j^{(1)}} \cdot \frac{\partial a_j^{(1)}}{\partial z_j^{(1)}} \cdot \frac{\partial z_j^{(1)}}{\partial w_{ji}^{(1)}}$$

Desarrollando cada término:

$$\frac{\partial z_k^{(2)}}{\partial a_j^{(1)}} = w_{kj}^{(2)}$$

$$\frac{\partial a_j^{(1)}}{\partial z_j^{(1)}} = \sigma'(z_j^{(1)})$$

$$\frac{\partial z_j^{(1)}}{\partial w_{ji}^{(1)}} = x_i$$

Resultado final:

$$\frac{\partial L}{\partial w_{ji}^{(1)}} = \left[\sum_{k=1}^{K} -(y_k - a_k^{(2)}) \cdot \sigma'(z_k^{(2)}) \cdot w_{kj}^{(2)}\right] \cdot \sigma'(z_j^{(1)}) \cdot x_i$$

🎯 Definición del Error Delta

Para simplificar, definimos:

$$\delta_k^{(2)} = -(y_k - a_k^{(2)}) \cdot \sigma'(z_k^{(2)})$$

$$\delta_j^{(1)} = \left[\sum_{k=1}^{K} \delta_k^{(2)} \cdot w_{kj}^{(2)}\right] \cdot \sigma'(z_j^{(1)})$$

⚙️ Algoritmo de Retropropagación

Paso 1: Propagación Hacia Adelante

Calcular todas las activaciones desde la entrada hasta la salida

$$a^{(0)} = x \quad \text{(entrada)}$$

$$z^{(l)} = W^{(l)} a^{(l-1)} + b^{(l)}$$

$$a^{(l)} = \sigma(z^{(l)})$$

Paso 2: Cálculo del Error de Salida

$$\delta^{(L)} = \nabla_a L \odot \sigma'(z^{(L)})$$

Paso 3: Retropropagación del Error

$$\delta^{(l)} = ((W^{(l+1)})^T \delta^{(l+1)}) \odot \sigma'(z^{(l)})$$

Paso 4: Cálculo de Gradientes

$$\frac{\partial L}{\partial W^{(l)}} = \delta^{(l)} (a^{(l-1)})^T$$

$$\frac{\partial L}{\partial b^{(l)}} = \delta^{(l)}$$

Paso 5: Actualización de Parámetros

$$W^{(l)} := W^{(l)} - \eta \frac{\partial L}{\partial W^{(l)}}$$

$$b^{(l)} := b^{(l)} - \eta \frac{\partial L}{\partial b^{(l)}}$$

📊 Ejemplo Numérico Detallado

Red Simple: 2 entradas → 2 neuronas ocultas → 1 salida

🔢 Datos Iniciales

Entrada: $x = [0.5, 0.3]$
Objetivo: $y = 0.8$
Pesos W₁:

$$W^{(1)} = \begin{pmatrix} 0.2 & 0.4 \\ 0.6 & 0.8 \end{pmatrix}$$

Pesos W₂: $W^{(2)} = [0.5, 0.7]$
Sesgos: $b^{(1)} = [0.1, 0.2]$, $b^{(2)} = 0.3$

⚡ Función de Activación

Sigmoide:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Derivada:

$$\sigma'(z) = \sigma(z)(1 - \sigma(z))$$

Cálculos Paso a Paso

1. Propagación Hacia Adelante

Capa oculta:

$$z_1^{(1)} = 0.2 \times 0.5 + 0.4 \times 0.3 + 0.1 = 0.32$$

$$z_2^{(1)} = 0.6 \times 0.5 + 0.8 \times 0.3 + 0.2 = 0.74$$

$$a_1^{(1)} = \sigma(0.32) = 0.579$$

$$a_2^{(1)} = \sigma(0.74) = 0.677$$

Capa de salida:

$$z^{(2)} = 0.5 \times 0.579 + 0.7 \times 0.677 + 0.3 = 1.063$$

$$\hat{y} = a^{(2)} = \sigma(1.063) = 0.743$$

2. Cálculo del Error

$$L = \frac{1}{2}(y - \hat{y})^2 = \frac{1}{2}(0.8 - 0.743)^2 = 0.00162$$

🔄 Retropropagación del Ejemplo

3. Error en Capa de Salida

$$\delta^{(2)} = -(y - \hat{y}) \times \sigma'(z^{(2)})$$

$$\sigma'(z^{(2)}) = a^{(2)}(1 - a^{(2)}) = 0.743 \times (1 - 0.743) = 0.191$$

$$\delta^{(2)} = -(0.8 - 0.743) \times 0.191 = -0.0109$$

4. Gradientes de Pesos de Salida

$$\frac{\partial L}{\partial W_1^{(2)}} = \delta^{(2)} \times a_1^{(1)} = -0.0109 \times 0.579 = -0.0063$$

$$\frac{\partial L}{\partial W_2^{(2)}} = \delta^{(2)} \times a_2^{(1)} = -0.0109 \times 0.677 = -0.0074$$

5. Error en Capa Oculta

$$\delta_1^{(1)} = \delta^{(2)} \times W_1^{(2)} \times \sigma'(z_1^{(1)})$$

$$\sigma'(z_1^{(1)}) = 0.579 \times (1 - 0.579) = 0.244$$

$$\delta_1^{(1)} = -0.0109 \times 0.5 \times 0.244 = -0.00133$$

$$\delta_2^{(1)} = \delta^{(2)} \times W_2^{(2)} \times \sigma'(z_2^{(1)})$$

$$\sigma'(z_2^{(1)}) = 0.677 \times (1 - 0.677) = 0.219$$

$$\delta_2^{(1)} = -0.0109 \times 0.7 \times 0.219 = -0.00167$$

6. Gradientes de Pesos Ocultos

$$\frac{\partial L}{\partial W_{11}^{(1)}} = \delta_1^{(1)} \times x_1 = -0.00133 \times 0.5 = -0.000665$$

$$\frac{\partial L}{\partial W_{12}^{(1)}} = \delta_1^{(1)} \times x_2 = -0.00133 \times 0.3 = -0.000399$$

$$\frac{\partial L}{\partial W_{21}^{(1)}} = \delta_2^{(1)} \times x_1 = -0.00167 \times 0.5 = -0.000835$$

$$\frac{\partial L}{\partial W_{22}^{(1)}} = \delta_2^{(1)} \times x_2 = -0.00167 \times 0.3 = -0.000501$$

🎯 Conclusiones y Puntos Clave

🔑 Conceptos Fundamentales

- **Regla de la Cadena:** Es la base matemática que permite calcular eficientemente las derivadas parciales
- **Propagación del Error:** El error se propaga desde la salida hacia la entrada, capa por capa
- **Gradiente Descendente:** Los gradientes calculados se usan para actualizar los pesos

📈 Ventajas de la Retropropagación

- **Eficiencia Computacional:** O(n) en lugar de O(n²) para el cálculo de gradientes
- **Escalabilidad:** Funciona con redes de cualquier tamaño y profundidad
- **Generalidad:** Aplicable a diferentes arquitecturas y funciones de activación
- **Convergencia:** Garantiza encontrar mínimos locales bajo condiciones apropiadas

🔢 Fórmulas Esenciales para Recordar

Error de salida:
$\delta^{(L)} = \nabla_a L \odot \sigma'(z^{(L)})$
Retropropagación del error:
$\delta^{(l)} = ((W^{(l+1)})^T \delta^{(l+1)}) \odot \sigma'(z^{(l)})$
Gradientes:
$\frac{\partial L}{\partial W^{(l)}} = \delta^{(l)} (a^{(l-1)})^T$

⚠️ Consideraciones Importantes

- **Problema del Gradiente Desvaneciente:** En redes profundas, los gradientes pueden volverse muy pequeños
- **Inicialización de Pesos:** Crucial para evitar simetrías y facilitar el aprendizaje
- **Tasa de Aprendizaje:** Debe ser ajustada cuidadosamente para asegurar convergencia
- **Regularización:** Técnicas como dropout y L2 ayudan a prevenir sobreajuste