Retropropagación en Redes Neuronales: El Motor del Aprendizaje Profundo

Descubre cómo la retropropagación ajusta los pesos de una red neuronal, permitiendo que aprenda y mejore con cada iteración. Es el corazón del aprendizaje profundo y la clave para entrenar modelos sofisticados.

Edwin Felipe Pinilla Peralta

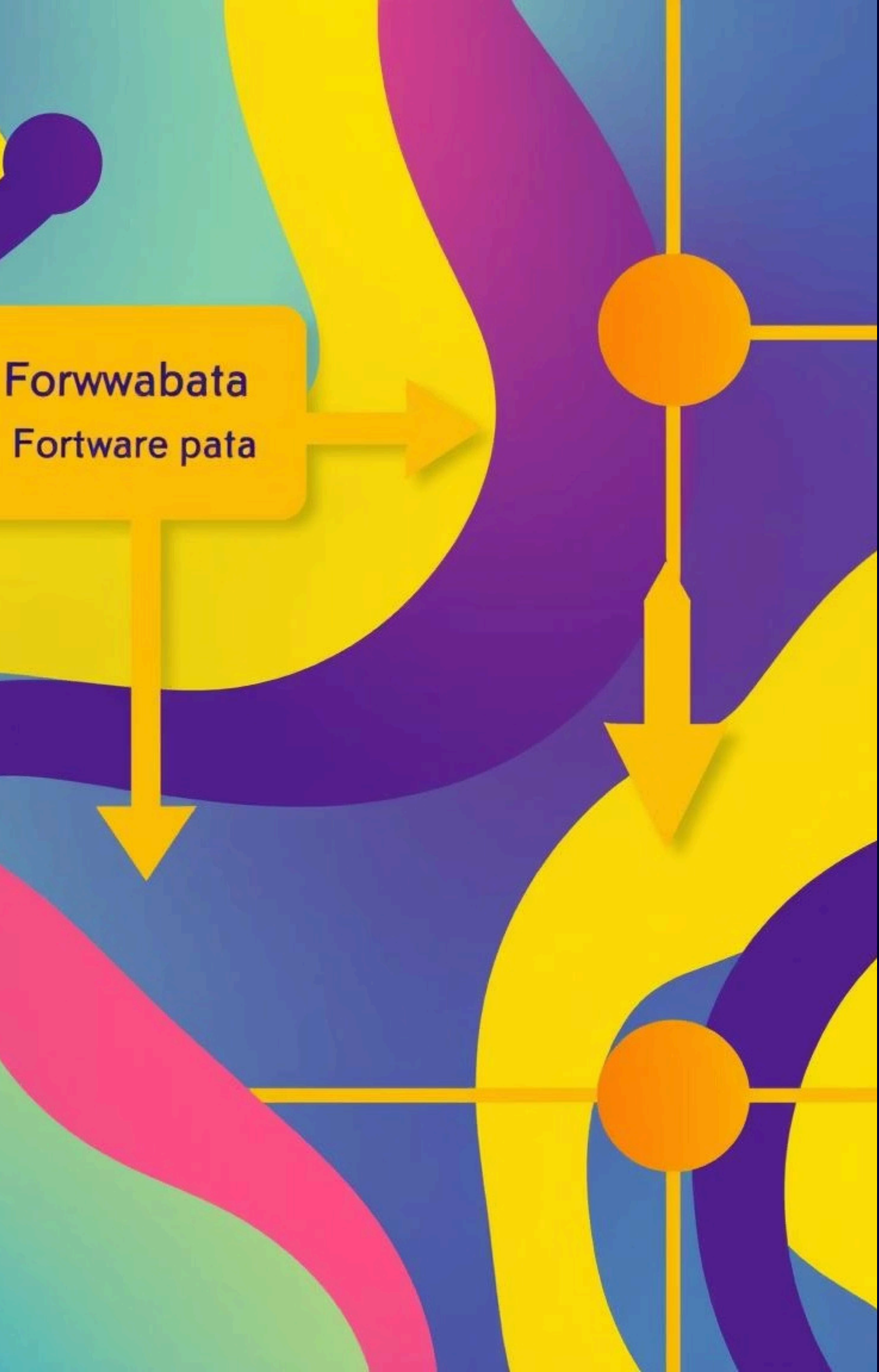
Cristian Alejandro Beltran Rojas

Javier Santiago Vargas Parra

El Objetivo: Ajustar Pesos y Sesgos

- Entrenar redes neuronales: Optimizar el rendimiento del modelo.
- Minimizar la función de pérdida: Reducir el error entre predicciones y valores reales.
- Ajuste iterativo: Refinar pesos y sesgos con cada ciclo de aprendizaje.

La retropropagación permite a las redes neuronales aprender de sus errores. Piensa en ello como ajustar un termostato: si la temperatura es incorrecta, lo ajustas para acercarte al valor deseado.



El Flujo Básico del Aprendizaje

1. Propagación Hacia Adelante (Forward Pass)

La entrada atraviesa la red, produciendo una salida.

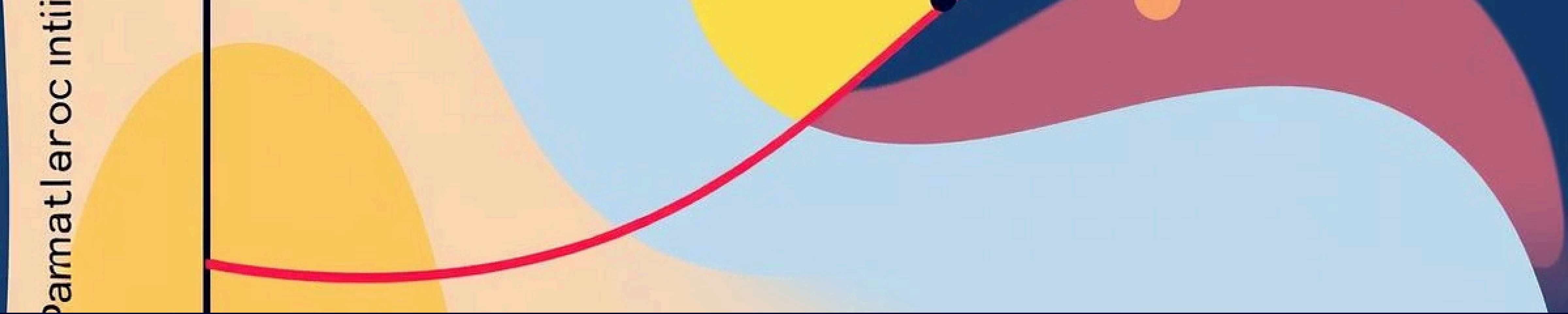
2. Cálculo del Error

Comparamos la salida de la red con el valor real.

3. Retropropagación (Backward Pass)

El error se propaga hacia atrás para actualizar los pesos.

Este ciclo de tres pasos se repite miles o millones de veces. Cada iteración acerca a la red a una solución óptima, permitiendo que aprenda de manera efectiva.

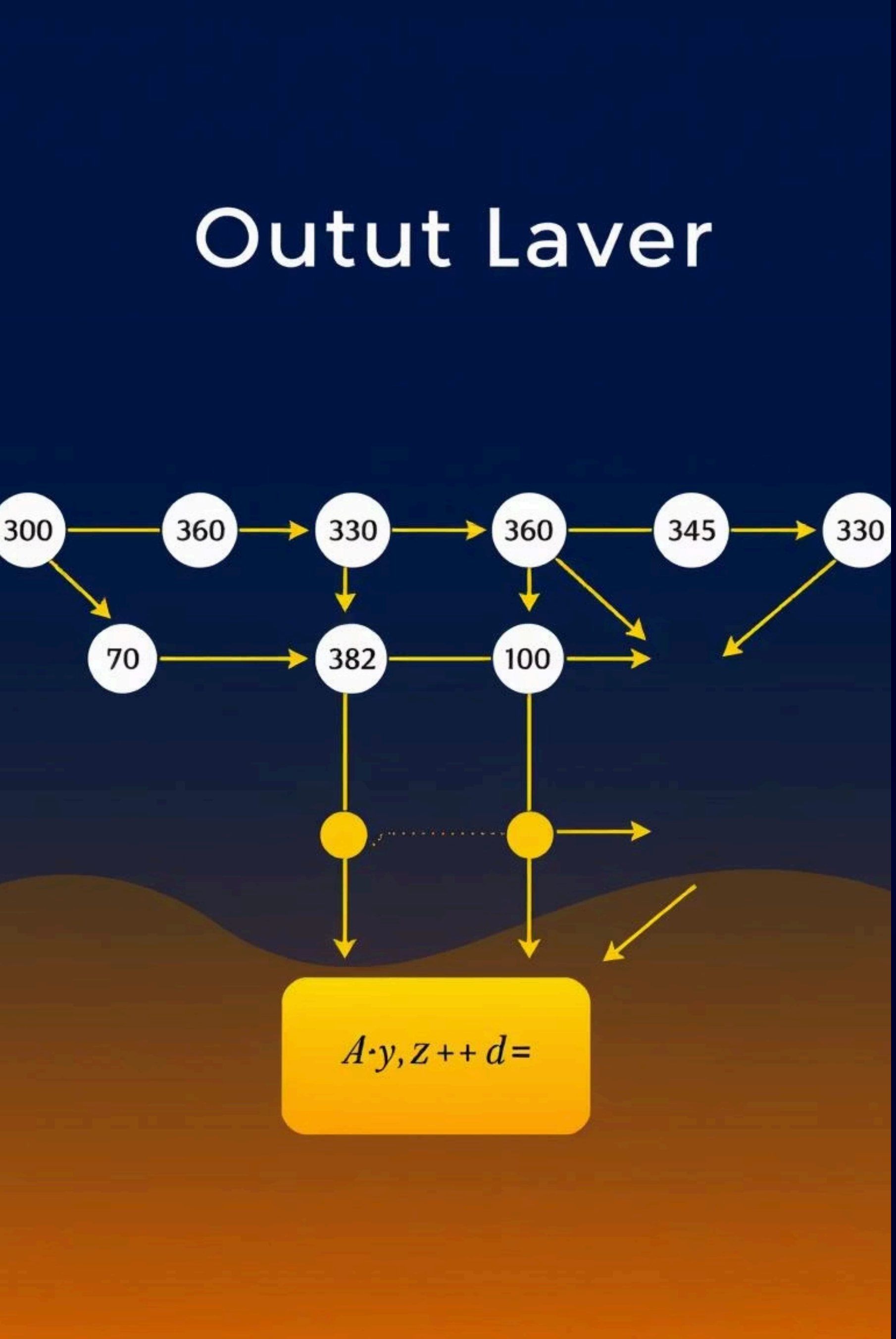


La Función de Pérdida: Medir el Error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

El Error Cuadrático Medio (MSE) cuantifica la diferencia entre las predicciones de la red y los valores reales. Es una métrica fundamental para evaluar el rendimiento.

Una función de pérdida baja indica que el modelo está haciendo predicciones precisas. El objetivo del entrenamiento es minimizar esta función.



Gradiente en la Capa de Salida

$$\delta^{(L)} = \nabla_{\hat{y}^{(L)}} L \odot f'(z^{(L)})$$

El gradiente de la capa de salida ($\delta^{(L)}$) es crucial. Combina la derivada de la función de pérdida con respecto a la salida y la derivada de la función de activación de la capa.

Representa el error inicial que se propaga hacia atrás. Es el punto de partida para el ajuste de pesos en todas las capas anteriores.

Propagación del Gradiente en Capas Ocultas

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

El gradiente de una capa oculta ($\delta^{(l)}$) se calcula en función del gradiente de la capa siguiente, los pesos de esa capa y la derivada de su propia función de activación.

Esto permite que el error "sepa" cómo afecta a las capas anteriores. Es un proceso de distribución del error.

Actualización de Pesos y Sesgos

Para los pesos:

$$W^{(l)} \leftarrow W^{(l)} - \eta \cdot \delta^{(l)} (a^{(l-1)})^T$$

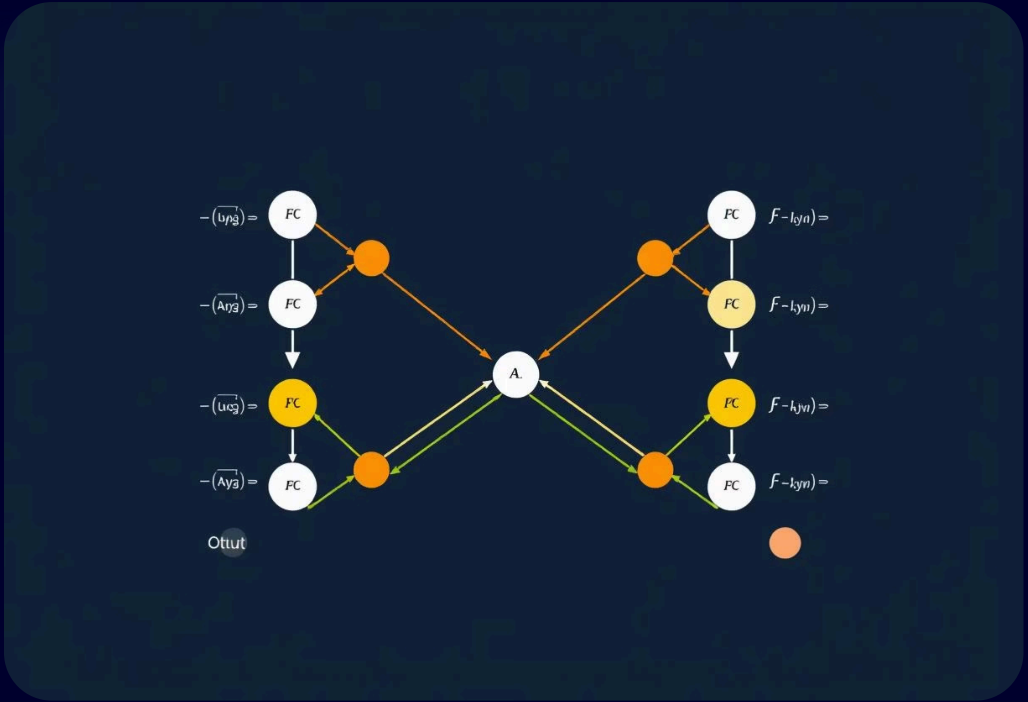
Para los sesgos:

$$b^{(l)} \leftarrow b^{(l)} - \eta \cdot \delta^{(l)}$$

Aquí, η es la tasa de aprendizaje, $\delta^{(l)}$ es el gradiente de la capa, y $a^{(l-1)}$ es la activación de la capa anterior.

Estas fórmulas son el corazón de la retropropagación, permitiendo que cada peso y sesgo se ajuste en la dirección correcta para minimizar la pérdida.

Diagrama: Grafo Computacional



Este grafo visualiza el flujo de datos (flechas hacia adelante) y la propagación de errores (flechas hacia atrás) durante la retropropagación.

Cada nodo representa una operación matemática, y las flechas indican dependencias. Facilita la comprensión del cálculo de gradientes.

Es una herramienta poderosa para el diseño y análisis de redes neuronales.