

7. Aprendizaje en Redes Multicapa, Retropropagación

7.1 El Problema: ¿De quién es la culpa?

En un **Perceptrón Simple** (una sola capa, neurona artificial), entrenar es fácil porque sabemos cuál es la respuesta correcta (t) y cuál es la que dio la neurona (y). El error es simplemente $(t - y)$.

En una **Red Multicapa**, tenemos neuronas ocultas en medio.

- Si la red se equivoca en la salida final, ¿qué neurona oculta tuvo la culpa?
- No tenemos un "objetivo correcto" para las capas intermedias.
- **Solución:** Necesitamos un mecanismo para enviar la información del error desde la salida **hacia atrás**, repartiendo la "culpa" entre las neuronas ocultas proporcionalmente a sus pesos. Esto es la **Retropropagación**.

7.2 Fundamento Matemático

El objetivo es el mismo que en la Regresión: Minimizar la función de Error Global (E).

$$E = \frac{1}{2} \sum (t - y)^2$$

- t (**Target / Objetivo**): Es la respuesta correcta (el examen resuelto). Ejemplo: "Esto es un perro (1)".
- y (**Yield / Salida**): Es lo que respondió la red. Ejemplo: "Creo que es un gato (0)".
- $(t - y)$: Es la diferencia. Si debía ser 1 y dijo 0, el error es 1.
- 2 (**Al cuadrado**): Elevamos al cuadrado para que los errores negativos no se cancelen con los positivos (y para castigar más los errores grandes).
- $\frac{1}{2}$: Es un truco matemático. Cuando hagamos la derivada más tarde, el exponente 2 bajará, se multiplicará por $\frac{1}{2}$ y se cancelarán ($2 \cdot 0.5 = 1$). Solo está ahí para facilitar las cuentas después.

Para minimizar este error, usamos el **Descenso por Gradiente**. Necesitamos saber cómo cambiar cada peso (w) para reducir el error. Matemáticamente, esto implica calcular la **Derivada del Error respecto a cada peso** ($\frac{\partial E}{\partial w}$).

La Importancia de la Función Sigmoide

Aquí es donde la función escalón (del Perceptrón antiguo) falla, porque no es derivable (tiene un salto brusco).

Usamos la Función Sigmoide ($f(z) = \frac{1}{1+e^{-z}}$) porque es suave y su derivada es muy fácil de calcular y computacionalmente barata:

$$f'(z) = f(z) \cdot (1 - f(z))$$

(La derivada se calcula usando el propio valor de salida de la neurona, ¡muy eficiente!).

7.3 El Algoritmo: Ciclo de Dos Pasos

El entrenamiento ocurre en un bucle repetitivo con dos fases claramente diferenciadas:

Fase 1: Propagación Hacia Adelante (Forward Pass)

La red actúa normal, como si estuviera prediciendo.

1. Introducimos los datos (x) en la capa de entrada.
2. La señal viaja capa por capa, calculando sumas ponderadas y activaciones.
3. Obtenemos la salida final (y) y calculamos el error comparando con el objetivo (t).

Fase 2: Propagación Hacia Atrás (Backward Pass)

Aquí ocurre el aprendizaje. Calculamos un valor δ (delta) que representa el "error local" de cada neurona.

1. Capa de Salida: Es fácil. El error es la diferencia directa con el objetivo, multiplicada por la derivada de la función de activación.

$$\delta_{salida} = (t - y) \cdot f'(z)$$

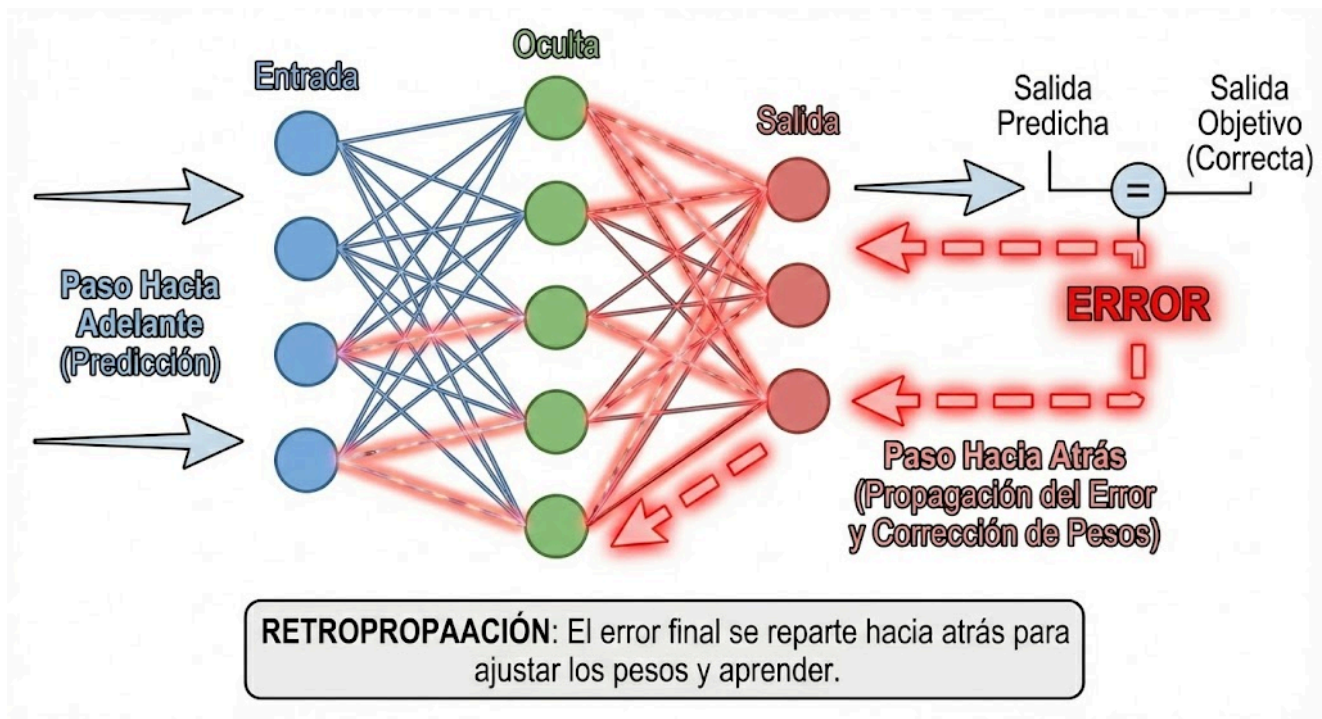
- $(t - y)$: **El Error**. "Te equivocaste por tanto".
- $f'(z)$: **La Derivada**. Esto asusta, pero significa "**Sensibilidad**".
 - Imagina que la neurona estaba muy segura (dio un 1 o un 0 rotundo). Su sensibilidad es baja (cuesta hacerla cambiar de opinión).
 - Si la neurona estaba dudosa (dio 0.5), su sensibilidad es alta.
 - **Traducción**: "Tu culpa es igual al tamaño del error multiplicado por lo fácil que es hacerte cambiar de opinión".

2. Capas Ocultas: Aquí está la magia. Como no tenemos objetivo t , calculamos el error como la suma ponderada de los errores de la capa siguiente (hacia la que enviamos señal).

$$\delta_{oculta} = \left(\sum \delta_{siguiente} \cdot w_{conexion} \right) \cdot f'(z)$$

- $\delta_{siguiente}$: Es la culpa de la neurona a la que le enviamos el dato (la neurona de salida). "Si la de arriba tiene mucha culpa, y yo le hablé, yo tengo culpa".
- $w_{conexion}$: Es el peso de mi conexión con ella. "Si yo le hablé muy alto (peso alto), tengo mucha culpa. Si mi conexión con ella es casi cero, no es mi culpa".
- \sum (**Suma**): Sumo las culpas que me llegan de *todas* las neuronas a las que estoy conectado.

- **Interpretación:** "Si mi conexión con la siguiente neurona es fuerte (w alto) y esa neurona tiene mucho error (δ alto), entonces yo soy muy responsable de ese error".



7.4 Actualización de Pesos

Una vez tenemos los δ (la "culpa") de cada neurona, actualizamos los pesos usando la regla delta generalizada.

$$w_{nuevo} = w_{actual} + \Delta w$$

Donde el cambio (Δw) se calcula como:

$$\Delta w = \eta \cdot \delta \cdot \text{entrada}$$

- η (**Eta - Tasa de aprendizaje**): Es la **Prudencia**. Es un número pequeño (ej. 0.1).
 - "Aunque tengas mucha culpa, no vamos a cambiar el peso de golpe a lo loco, lo cambiaremos poquito a poco para no romper nada".
- δ (**Delta - La Culpa**): Es la **Dirección**. Nos dice si el error fue por exceso o por defecto.
- $x_{entrada}$ (**La entrada**): Es la **Evidencia**.
 - Si la entrada (x) era 0, esa conexión no estaba activa, así que **ese peso** no contribuyó al error. Si x es 0, todo se multiplica por 0 y el peso no se toca. ¡Solo corregimos los pesos que participaron!

7.5 Resumen del Algoritmo

Según el documento [Retropropagación del error.pdf](#), el algoritmo completo es:

1. **Inicializar pesos:** Valores pequeños y aleatorios (¡Importante no ponerlos todos a cero o la red no aprende!).

2. **Repetir** (hasta que el error sea bajo):

- Para cada ejemplo de entrenamiento (X, T) :
 1. **Forward**: Calcular las salidas de todas las capas hasta el final.
 2. **Error Salida**: Calcular δ_k para las neuronas de salida.
 3. **Backward**: Calcular δ_j para las neuronas ocultas (usando los δ_k y los pesos w_{jk}).
 4. **Update**: Actualizar todos los pesos de la red:

$$w_{ij} \leftarrow w_{ij} + \eta \cdot \delta_j \cdot y_i$$

Aquí tienes una versión que mantiene un tono más profesional en general pero incorpora toques informales y críticos del texto original:

7.6 Conclusión de la Asignatura

Una vez más se demuestra que el grado presenta una dejadez considerable por parte de las "vacas sagradas" de la facultad. Y es que no puede ser: estas personas son reconocidísimas en sus ámbitos, publican 300 artículos, acuden a congresos internacionales... pero cuando llega el momento de pensar en sus alumnos y proporcionarles una bibliografía de calidad con la cual puedan aprender los conceptos de forma intuitiva, te sueltan un PDF de 2001 robado de otra universidad o presentaciones sin orden lógico alguno, llenas de palabras sueltas y esquemas incomprensibles.

Eso sí, si te quejas, la alternativa que te ofrecen es el libro de turno de 1990 escrito en alemán por un médico francés. Y apáñate tú para conseguirlo y entenderlo.

La situación es insostenible. Es frecuente encontrarse con:

- Documentación completamente desactualizada
- Materiales de dudosa procedencia sin adaptar al contexto actual
- Presentaciones caóticas sin estructura pedagógica
- Bibliografía obsoleta, inaccesible o en idiomas que nadie domina

¿Tan difícil es hacer las cosas bien? Otras carreras universitarias lo consiguen. No se pide la perfección, pero sí un mínimo de coherencia y actualización en los materiales docentes.

Considero que el grado debería aprender de otras titulaciones y mejorar urgentemente este aspecto. Con la jubilación progresiva de las viejas vacas sagradas de la universidad, espero sinceramente que esta situación mejore con el tiempo y que la nueva generación de profesorado traiga una renovación real en la calidad docente, no solo en la investigadora.