Tarea #6

“Algoritmo de Back Propagation”

INTELIGENCIA ARTIFICIAL

18:00 – 19:00 pm

Profesor:

JOSE MARIO RIOS FELIX

Alumno:

Estrada Valenzuela Jesús Ernesto

7to semestre

ING. En Sistemas Computacionales

25/10/2025

Backpropagation: El Motor del Aprendizaje Profundo

Una investigación concisa sobre el algoritmo que hace posible la inteligencia artificial moderna.

1. ¿Qué es Backpropagation?

En esencia, **Backpropagation** (o Retropropagación) es el algoritmo fundamental que permite a las redes neuronales artificiales **aprender**.

Si una red neuronal es como un cerebro, *backpropagation* es el proceso de "estudio y corrección" que ese cerebro utiliza. Es un método para ajustar eficientemente las "conexiones" (llamadas **pesos**) dentro de la red para que pueda pasar de hacer conjeturas aleatorias a tomar decisiones increíblemente precisas.

Es el motor que impulsa casi todo el *Deep Learning*: desde el reconocimiento de imágenes (como en tu teléfono) hasta los grandes modelos de lenguaje (como yo).

2. El Problema: ¿Cómo Aprende una Red?

Imagina una red neuronal simple que intenta reconocer si una imagen es un "perro" o un "gato".

1. **Conjetura Inicial (Forward Pass):** Le das una foto de un perro. La red, que al inicio tiene conexiones aleatorias, procesa la imagen y dice: "Estoy 60% seguro de que es un gato".
2. **El Error:** Tú sabes que la respuesta correcta es "perro". La red cometió un error.
3. **La Pregunta Clave:** La red tiene miles, quizás millones, de conexiones (pesos). ¿Qué conexiones específicas debe ajustar? ¿Y cuánto debe ajustar cada una para que la próxima vez acierte?

Responder a esta pregunta es extremadamente difícil. Ajustar un peso al azar podría empeorar las cosas. Aquí es donde entra *backpropagation*.

3. La Solución: El Flujo del Algoritmo

Backpropagation es un proceso inteligente de dos pasos que se repite miles de veces. Utiliza el **cálculo (específicamente, la regla de la cadena)** para asignar "culpas" de manera eficiente.

Paso A: El Pase Hacia Adelante (Forward Pass)

Es exactamente lo que vimos en la visualización.

- Los datos de entrada (la imagen del perro) se introducen en la primera capa.
- Cada neurona se activa y pasa su resultado a la siguiente capa.
- Esto continúa hasta que la última capa da una respuesta (la "conjetura").

Paso B: El Pase Hacia Atrás (Backward Pass)

Aquí ocurre la magia. Es el corazón del algoritmo.

1. **Calcular el Error Total:** La red mide qué tan lejos está su conjetura ("gato") de la respuesta correcta ("perro"). A esto se le llama **función de pérdida** (Loss Function).
2. **Propagar el Error Hacia Atrás:** El algoritmo "camina" hacia atrás, desde la capa de salida hasta la de entrada, y se pregunta en cada conexión:

"¿Qué tanto contribuiste tú, conexión específica, al error total?"

3. **Asignación de "Culpa" (El Gradiente):** Usando derivadas (cálculo), la red calcula la **pendiente** (o **gradiente**) del error para cada peso. Este número mágico (el gradiente) te dice dos cosas:
    - **Dirección:** ¿Debo *aumentar* o *disminuir* este peso para reducir el error?
    - **Magnitud:** ¿Qué *tan grande* fue tu contribución al error? (Un gradiente grande significa que ese peso fue muy "culpable" y necesita un gran ajuste).
4. **Actualizar los Pesos:** Una vez que la red sabe la "culpa" (gradiente) de cada peso, los ajusta todos ligeramente en la dirección correcta para minimizar el error. A este paso se le llama **descenso de gradiente** (Gradient Descent).

Y eso es todo. El ciclo (Paso A y Paso B) se repite con miles de imágenes, y con cada ciclo, la red se vuelve un poco más precisa, hasta que puede distinguir perros y gatos con una precisión asombrosa.

4. ¿Por Qué es tan Importante?

Antes de *backpropagation* (en los años 70 y 80), no teníamos una forma eficiente de entrenar redes neuronales con **múltiples capas** (redes "profundas"). Solo podíamos entrenar redes muy simples y superficiales.

Backpropagation fue la clave que "desbloqueó" el entrenamiento de redes profundas. Descubrió cómo asignar la culpa de manera eficiente a través de todas las capas,

permitiendo que las redes aprendieran tareas increíblemente complejas que antes eran imposibles.

5. Conclusion

- **Qué es:** Un algoritmo para entrenar redes neuronales.
- **Cómo funciona:** Calcula el error de la red y luego propaga ese error *hacia atrás* para descubrir cuánto "contribuyó" cada conexión al error.
- **Para qué sirve:** Permite ajustar millones de conexiones de manera inteligente (usando "gradientes") para que la red aprenda y mejore con cada ejemplo.
- **Por qué es crucial:** Es la pieza que hizo posible el *Deep Learning* moderno.