Instituto Tecnológico de Culiacán

Ingeniería en Sistemas Computacionales

Materia:

Inteligencia Artificial.

Tarea:

Back Propagation.

Alumno:

Flores Saldaña Martin Alejandro

Docente:

Jose Mario Rios Felix

Grupo:

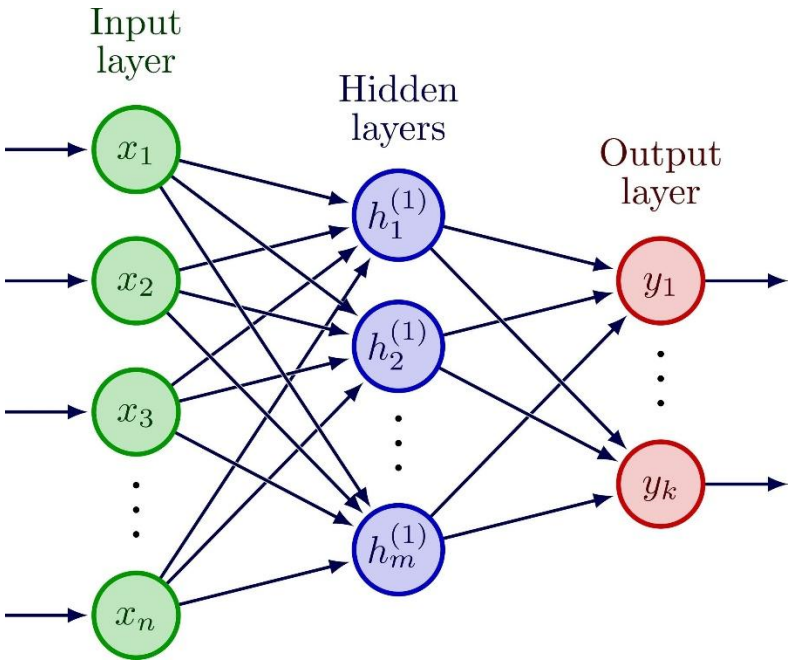
18:00-19:00

1. Introducción y Definición

El Backpropagation (retropropagación) es el método dominante para entrenar redes neuronales artificiales. Matemáticamente, es una técnica eficiente para calcular el gradiente de la función de pérdida (error) con respecto a los pesos de la red.

Su objetivo principal es responder a la pregunta: "¿Cómo debe cambiar cada peso en la red para que el error final disminuya?".

Nota histórica: Aunque sus raíces matemáticas datan de la década de 1960, se popularizó en 1986 gracias al artículo de Rumelhart, Hinton y Williams, marcando el inicio del renacimiento de las redes neuronales.



2. Fundamento Matemático: La Regla de la Cadena

El corazón del algoritmo es la Regla de la Cadena del cálculo diferencial. Dado que una red neuronal es esencialmente una función compuesta gigante (una función dentro de otra función, dentro de otra...), necesitamos descomponer la derivada del error final capa por capa.Si tenemos una función compuesta $y = f(g(x))$, la derivada de y con respecto a x es:

$$\frac{dy}{dx} = \frac{dy}{du} \cdot \frac{du}{dx}$$

3. El Ciclo de Entrenamiento

El algoritmo ocurre en un ciclo de dos fases principales que se repite iterativamente:

Fase A: Forward Pass (Propagación hacia adelante)

1. Los datos de entrada entran en la red.
2. Se multiplican por los pesos (w) y se suman los sesgos (b).
3. Pasan a través de una función de activación (como ReLU o Sigmoide).
4. La red produce una predicción ($\hat{y}$).
5. Se calcula el Error (Loss) comparando la predicción con el valor real usando una función de costo (ej. Error Cuadrático Medio).

Fase B: Backward Pass (Retropropagación)

Aquí ocurre el aprendizaje real. El error calculado al final se "envía de vuelta" a través de la red:

1. Se calcula el gradiente de la última capa.
2. Usando la regla de la cadena, se calcula el gradiente de las capas ocultas anteriores de forma recursiva.
3. Se determina la "responsabilidad" de cada neurona en el error total.

4. Actualización de Pesos (Descenso del Gradiente)

Una vez que tenemos los gradientes (que nos dicen la dirección en la que aumenta el error), actualizamos los pesos en la dirección opuesta para minimizar ese error.

La fórmula de actualización para un peso w es:

$$w_{nuevo} = w_{actual} - \eta \cdot \frac{\partial E}{\partial w}$$

Donde:

- η (Eta): Es la Tasa de Aprendizaje (Learning Rate). Define qué tan grandes son los pasos que damos.

- $\frac{\partial E}{\partial w}$: Es el Gradiente. Indica cuánto cambia el error total E si cambiamos ligeramente el peso w.

5. Importancia y Aplicaciones

Sin Backpropagation, el "Deep Learning" (Aprendizaje Profundo) no sería viable. Es el motor detrás de:

1. Visión por Computadora (CNNs): Ajusta los filtros para detectar bordes, texturas y objetos.
2. Procesamiento de Lenguaje Natural (Transformers): Permite a modelos como GPT entender la relación entre palabras distantes.