



Instituto Tecnológico y de Estudios Superiores de Monterrey

Materia: Implementación de internet de las cosas

Grupo: 501

Actividad evaluativa

“Actividad Integral estructura de datos lineales (Evidencia Competencia)”

Alumnos:

Omar Reyes Barrueta - A01772756

Profesor:

Israel Tabarez Paz

Fecha de entrega: 20/09/2024

Perceptrón y Backpropagation en Redes Neuronales

Introducción

El perceptrón y el algoritmo de backpropagation son los pilares sobre los que se erige el fascinante mundo de las redes neuronales. El primero —el perceptrón— es un modelo sencillo pero poderoso que sentó las bases para la creación de estructuras más complejas. Por otro lado, backpropagation revolucionó el aprendizaje automático al optimizar redes profundas a través del ajuste continuo de pesos. Ambos algoritmos han encontrado múltiples aplicaciones: desde la clasificación binaria hasta el reconocimiento de patrones más sofisticados.

1. Perceptrón

¿Qué es el Perceptrón?

El perceptrón no es más que un modelo de red neuronal que toma múltiples entradas, las combina ponderándolas con pesos, y, mediante una función de activación, decide si dispara una salida específica. En términos más simples: clasifica datos que sean linealmente separables... Y aunque esto pueda sonar simple, su impacto fue enorme en los inicios de la inteligencia artificial.

Parámetros de entrada:

- X : Vector de entradas (las características del conjunto de datos).
- w : Pesos asociados a cada entrada.
- b : Sesgo o *bias* (un valor adicional que permite ajustar la decisión).
- η : Tasa de aprendizaje (indica cuánto deben ajustarse los pesos en cada iteración).

¿Para qué se usa?

El perceptrón se aplica especialmente en problemas de clasificación binaria, como determinar si un correo es spam o no. Aunque su poder es limitado —ya que no puede resolver problemas no linealmente separables, como el clásico XOR—, sigue siendo la base conceptual de las redes neuronales modernas.

Algoritmo del Perceptrón

1. Inicialización: Asignar valores iniciales aleatorios a los pesos y al sesgo.
2. Propagación hacia adelante: Calcular la salida según la función $y = \text{activation}(w \cdot X + b)$.
3. Actualización de los pesos: Modificar los pesos según el error $w = w + \eta \cdot (y_{\text{real}} - y_{\text{predicho}}) \cdot X$.
4. Repetir este proceso para cada entrada y durante varias épocas, hasta minimizar el error.

2. Backpropagation

¿Qué es Backpropagation?

¡Ah, backpropagation! Este algoritmo fue un avance extraordinario. Se trata de una técnica de optimización basada en descenso del gradiente, donde los errores se propagan hacia atrás desde la salida hasta las capas iniciales, permitiendo que cada peso se ajuste de manera precisa.

Parámetros esenciales:

- X: Datos de entrada.
- y: Salidas reales (etiquetas).
- η : Tasa de aprendizaje.
- Número de capas y neuronas.
- Función de activación: ReLU, sigmoide, o softmax según el caso.

Aplicaciones:

Backpropagation se aplica en tareas como clasificación multiclase, regresión, reconocimiento de imágenes, y hasta procesamiento de lenguaje natural. Todo esto hace que algoritmos modernos —como las redes convolucionales o las redes recurrentes— debían su funcionamiento al principio de este método.

Algoritmo de Backpropagation

1. Propagación hacia adelante: Calcular las salidas de todas las capas.
2. Cálculo del error: Comparar la salida predicha con la salida real.
3. Propagación hacia atrás: Determinar los gradientes del error con respecto a los pesos.

4. Actualización de pesos: Ajustar los pesos usando el descenso del gradiente.
5. Iterar hasta minimizar la función de pérdida.

Conclusiones

El perceptrón es el inicio de todo; simple, pero funcional en problemas básicos. Sin embargo, es el algoritmo de backpropagation el que ha permitido que las redes neuronales prosperen en aplicaciones complejas. Ambas técnicas siguen vigentes, ya sea en el ámbito académico o en la industria.

Fuentes Bibliográficas

1. McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4), 115-133.
2. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533-536.
3. Videos en YouTube: FreeCodeCamp, 3Blue1Brown.
4. OpenAI. (2024). ChatGPT (ayuda en gramática). <https://www.openai.com/chatgpt>