



Instituto Tecnológico de Culiacán

Inteligencia Artificial Perceptrón

Unidad: 4

Maestro: Zuriel Dathan Mora Felix

Alumno: Miguel Angel Barraza Leon

Carrera: Ingeniería en Sistemas Computacionales

Numero Control: 20170601

Culiacán, Sinaloa.

27/05/25

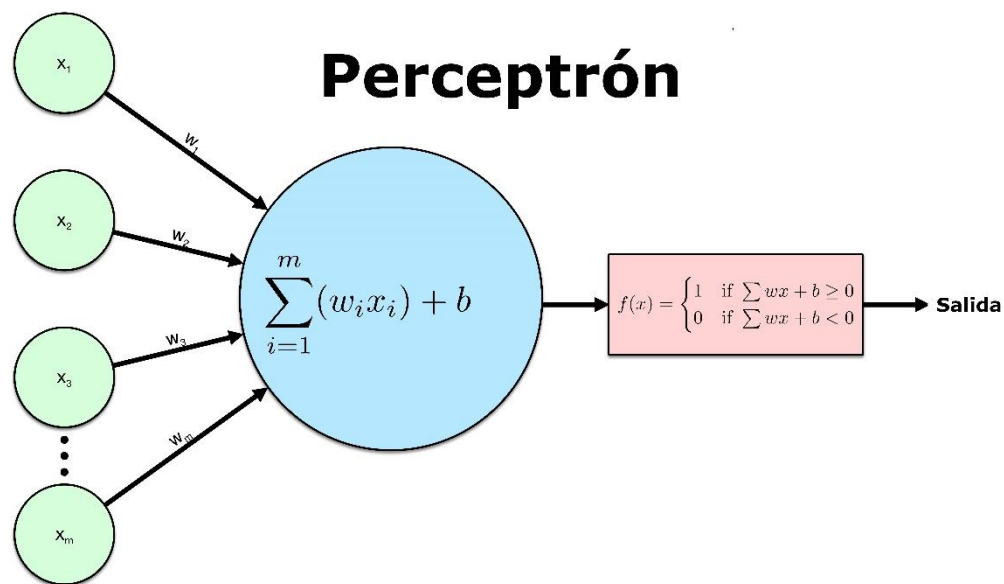
Introducción

En este documento se va a explicar lo que es un perceptrón, y el intento de que realice las 4 operaciones aritméticas y explicar si es posible y no y por qué.

¿Qué es un Perceptrón?

El perceptrón es el modelo más simple de una red neuronal. Fue propuesto por Frank Rosenblatt en 1958 y es la base de las redes neuronales artificiales modernas.

Un perceptrón toma varias entradas, las pondera, las suma y pasa el resultado por una función de activación (en el caso clásico, una función escalón o lineal).



Características del Perceptrón Clásico

- Una sola capa de neuronas
- No tiene capas ocultas
- Utiliza funciones de activación lineales o escalón
- Solo puede resolver problemas linealmente separables

Ventajas del Perceptrón

- Modelo simple y fácil de entender
- Útil para problemas lineales
- Rápido en tareas sencillas como clasificación básica
- Implementación directa con librerías como NumPy o TensorFlow



Desventajas del Perceptrón

- No puede resolver problemas no lineales (como XOR)
- No aprende funciones no lineales como multiplicación o división
- Bajo poder de generalización
- Es una base teórica, pero insuficiente para problemas reales complejos



¿Por qué NO puede aprender multiplicación y división?

Multiplicación y división son operaciones no lineales. Un perceptrón clásico solo puede aprender funciones que se pueden expresar como una combinación lineal de las entradas.

Un perceptrón puede aprender bien suma y resta (funciones lineales), pero no puede aprender multiplicación ni división, ni siquiera de forma aproximada.

Código realizado en Python con tensorflow (solo un perceptron).

```
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

# Datos de entrada (A, B)
X = np.array([
    [0, 0],
    [0, 1],
    [1, 0],
    [1, 1]
], dtype=np.float32)

# Salida esperada: [suma, resta, multiplicación, división]
Y = np.array([
    [0, 0, 0, 0],
    [1, -1, 0, 0],
    [1, 1, 0, 99],
    [2, 0, 1, 1]
], dtype=np.float32)

# Modelo tipo perceptrón: sin capas ocultas, salida lineal (no va a dar
respuestas tan acertadas)
model = Sequential([
    Dense(4, input_dim=2, activation='linear') # 4 salidas: suma, resta,
multiplicación, división
])
model.compile(optimizer='adam', loss='mse')
model.fit(X, Y, epochs=1000, verbose=0)

# Evaluación
print("Resultados del perceptrón (TensorFlow):")
predicciones = model.predict(X)

# Imprimir resultados corregidos
for i in range(len(X)):
    a, b = X[i]
    suma, resta, mult, div = map(float, predicciones[i])
    print(f"A={int(a)} B={int(b)} => Suma: {suma:.2f}, Resta: {resta:.2f},
Mult: {mult:.2f}, Div: {div:.2f}")
```