

# Implementación de un Perceptrón Simple en 2D

Documentación Técnica

19 de noviembre de 2024

## 1. Introducción

Este documento describe la implementación de un perceptrón simple para clasificación binaria en un espacio bidimensional. El perceptrón es uno de los algoritmos más básicos de aprendizaje automático, capaz de encontrar un hiperplano (en este caso, una línea) que separe dos clases de puntos linealmente separables.

## 2. Fundamento Teórico

El perceptrón implementa una función de decisión de la forma:

$$f(x) = \text{sign}(w^T x + b) \quad (1)$$

donde:

- $w$  es el vector de pesos (en 2D:  $w = [w_1, w_2]$ )
- $x$  es el vector de entrada (en 2D:  $x = [x_1, x_2]$ )
- $b$  es el término de sesgo (bias)
- $\text{sign}(z)$  retorna  $+1$  si  $z \geq 0$  y  $-1$  en caso contrario

## 3. Implementación

### 3.1. Inicialización

El código comienza inicializando los pesos y el bias de manera aleatoria:

```
1 w = np.random.randn(2)
2 b = np.random.randn()
3 learning_rate = 0.1
```

### 3.2. Algoritmo de Entrenamiento

El entrenamiento sigue el algoritmo del perceptrón:

```
1 for epoch in range(max_epochs):
2     hay_error = False
3     for i in range(len(X)):
4         pred = np.sign(np.dot(current_w, X[i]) +
5                             current_b)
6         if pred != y[i]:
7             hay_error = True
8             current_w += learning_rate * y[i] * X[i]
9             current_b += learning_rate * y[i]
```

La regla de actualización implementada es:

$$w \leftarrow w + \eta y_i x_i \quad (2)$$

$$b \leftarrow b + \eta y_i \quad (3)$$

donde  $\eta$  es la tasa de aprendizaje (learning\_rate).

### 3.3. Visualización

La visualización se realiza mediante una línea que representa el hiperplano de separación. La ecuación de la línea se obtiene de:

$$w_1 x + w_2 y + b = 0$$

Despejando  $y$ :

$$y = -\frac{w_1}{w_2}x - \frac{b}{w_2}$$

Esta transformación se implementa en el código:

```
1 def update_line():
2     x_min, x_max = ax.get_xlim()
3     y_min = (-w[0] * x_min - b) / w[1]
4     y_max = (-w[0] * x_max - b) / w[1]
5     return [x_min, x_max], [y_min, y_max]
```

## 4. Características de la Implementación

- **Animación:** El código genera una animación que muestra la evolución del hiperplano durante el entrenamiento.
- **Visualización de la Ecuación:** En cada iteración se muestra la ecuación actual de la recta en formato  $y = mx + b$ .
- **Convergencia:** El algoritmo se detiene cuando logra clasificar correctamente todos los puntos o alcanza el número máximo de épocas.

## 5. Limitaciones

- Solo funciona con datos linealmente separables
- La solución final puede variar debido a la inicialización aleatoria
- La tasa de aprendizaje fija puede afectar la convergencia

## 6. Conclusión

Esta implementación demuestra el funcionamiento básico del perceptrón en un espacio bidimensional, proporcionando una visualización clara del proceso de aprendizaje. La animación resultante ayuda a entender cómo el algoritmo ajusta iterativamente el hiperplano de separación hasta encontrar una solución satisfactoria.