

INTELIGENCIA ARTIFICIAL GENERATIVA (IAGen) Programa Python O.O del perceptrón

Vamos a generar paso a paso, usando programación orientada a objetos con Python y la herramienta ChatGPT, el programa del **Perceptrón**.

Dialogo con el ChatGPT:

Primera entrada:

"Por favor, necesito generar un programa orientado a objetos con Python, redes neuronales, aprendizaje automático para resolver el OR con las siguientes características. Es el perceptrón con una neurona de salida y varias de entrada, para entrenar una red neuronal que tiene como entrada 00, 01, 10 y 11. Y como salida 0,1,1,1. El programa no debe tener librerías especiales, solo con Python O.O, ejemplo de uso, etc..."

Segunda entrada:

"Todo está muy bien, pero tengo una pregunta: ¿Por qué no veo la función escalón de salida?"

Tercera entrada:

"Presenta este aviso en la pantalla:
`print(f"Epoch{epoch+1}\")` SyntaxError: unexpected character
after line

ChatGPT generó el programa correcto de Perceptrón usando el lenguaje Python orientado a objetos

```

1  # -*- coding: utf-8 -*-
2  """
3  Creado Febrero 28 de 2025
4  Nombre: miprimerAR
5  Que hace: red neuronal perceptron
6  Autores: Ms. Hernando lvarez R.
7           Ms. Leonardo lvarez V.
8           Ms. Luis Fernando lvarez V.
9  Universidad Tecnol gica de Pereira
10 C digo construido en ChatGPT
11 """
12
13 class Perceptron:
14     def __init__(self, input_size, learning_rate=0.1, epochs=10):
15         """Inicializa el perceptr n.
16         Args:
17             input_size (int): N mero de entradas.
18             learning_rate (float): Tasa de aprendizaje.
19             epochs (int): N mero de iteraciones para el entrenamiento.
20         """
21         self.weights = [0.0] * (input_size + 1) # Pesos, incluyendo el sesgo
22         self.learning_rate = learning_rate
23         self.epochs = epochs
24
25     def step_function(self, x):
26         """Funci n escal n."""
27         return 1 if x >= 0 else 0
28
29     def predict(self, inputs):
30         """Realiza una predicci n para una entrada dada.
31         Args:
32             inputs (list): Lista de entradas.
33         Returns:
34             int: Salida (0 o 1).
35         """
36         summation = sum(w * x for w, x in zip(self.weights[:-1], inputs)) + self.weights[-1]
37         return self.step_function(summation)
38
39     def train(self, training_inputs, labels):
40         """Entrena el perceptr n utilizando los datos de entrenamiento.
41         Args:
42             training_inputs (list): Lista de listas con las entradas de entrenamiento.
43             labels (list): Lista con las salidas deseadas.
44         """
45         for _ in range(self.epochs):
46             for inputs, label in zip(training_inputs, labels):
47                 prediction = self.predict(inputs)
48                 error = label - prediction
49
50                 # Actualizaci n de pesos y sesgo
51                 self.weights[:-1] = [
52                     w + self.learning_rate * error * x
53                     for w, x in zip(self.weights[:-1], inputs)
54                 ]
55                 self.weights[-1] += self.learning_rate * error
56
57                 print(f"Entrada: {inputs}, Predicci n: {prediction}, "
```

```
58         f"Error: {error}, Pesos: {self.weights}")
59
60 # Ejemplo de uso
61 def main():
62     # Entradas y salidas para la operaci n OR
63     training_inputs = [[0, 0], [0, 1], [1, 0], [1, 1]]
64     labels = [0, 1, 1, 1]
65
66     # Crear el perceptr n
67     perceptron = Perceptron(input_size=2, learning_rate=0.1, epochs=10)
68
69     # Entrenar el perceptr n
70     perceptron.train(training_inputs, labels)
71
72     # Probar el perceptr n
73     print("\nResultados de predicci n:")
74     for inputs in training_inputs:
75         print(f"Entrada: {inputs}, "
76               f"Salida predicha: {perceptron.predict(inputs)}")
77
78 if __name__ == "__main__":
79     main()
```