

```

In [1]: # Instalar la Librería micrograd
# !pip install micrograd # Install the micrograd library

# Importar los módulos necesarios
import random
from micrograd.engine import Value
from micrograd.nn import MLP

# Definir los datos de entrada (Xs) y las salidas deseadas (ys)
Xs = [
    [2.5, 3.5, -0.5],
    [4.0, -1.0, 0.5],
    [0.5, 1.5, 1.0],
    [3.0, 2.0, -1.5]
]
ys = [1.0, -1.0, -1.0, 1.0] # Valores deseados

# Crear una red neuronal con 3 entradas, 2 capas ocultas de 4 neuronas y una salida de 1 neurona
random.seed(42) # Para tener resultados reproducibles
n = MLP(3, [4, 4, 1])

# Definir la tasa de aprendizaje
learning_rate = 0.01

# Número de épocas (iteraciones de entrenamiento)
epochs = 200

# Función para imprimir los pesos actuales
def print_weights(n, etapa):
    print(f"\nPesos en la {etapa}:")
    for i, p in enumerate(n.parameters()):
        print(f"Peso {i+1}: {p.data:.4f}")

# Entrenamiento de la red neuronal
for epoch in range(epochs):
    # Inicializar la pérdida
    loss = Value(0.0)

    # Forward pass y cálculo de la pérdida
    for x, y in zip(Xs, ys):
        y_pred = n(x) # Predicción de la red
        loss += (y_pred - Value(y))**2 # Pérdida cuadrática

    # Mostrar los pesos en la penúltima y última iteración (forward propagation)
    if epoch in [epochs - 2, epochs - 1]:
        print(f'\nEpoch {epoch+1}, Forward pass')
        print(f'Pérdida: {loss.data / len(Xs)}')
        print_weights(n, f"forward propagation en la iteración {epoch+1}")

    # Backward pass (retropropagación)
    n.zero_grad() # Inicializar los gradientes a cero antes de la retropropagación
    loss.backward()

    # Mostrar los gradientes (backward propagation)
    if epoch in [epochs - 2, epochs - 1]:
        print(f'\nEpoch {epoch+1}, Backward pass')
        print("Gradientes después de backward propagation:")
        for i, p in enumerate(n.parameters()):
            print(f"Gradiente del peso {i+1}: {p.grad:.4f}")

```

```
# Actualización de Los pesos
for p in n.parameters():
    p.data -= learning_rate * p.grad # Actualizar los parámetros (pesos) con gradiente desc

# Mostrar los pesos después de la actualización (backward propagation)
if epoch in [epochs - 2, epochs - 1]:
    print_weights(n, f"backward propagation en la iteración {epoch+1}")

# Predicciones finales después del entrenamiento
print("\nPredicciones finales:")
for x, y in zip(Xs, ys):
    y_pred = n(x)
    print(f"Entrada: {x}, Predicción: {y_pred.data:.4f}, Objetivo: {y}")
```

Epoch 199, Forward pass
Pérdida: 0.0017649445005884878

Pesos en la forward propagation en la iteración 199:

Peso 1: 0.5862
Peso 2: -1.1550
Peso 3: -0.3241
Peso 4: 0.0652
Peso 5: -0.1636
Peso 6: 0.9073
Peso 7: -0.3656
Peso 8: -0.4145
Peso 9: 1.0388
Peso 10: -0.8407
Peso 11: -0.5563
Peso 12: 0.0681
Peso 13: -0.9404
Peso 14: -0.5627
Peso 15: 0.0107
Peso 16: 0.0000
Peso 17: -1.2822
Peso 18: -0.5240
Peso 19: 0.8567
Peso 20: 0.0899
Peso 21: 0.2560
Peso 22: -0.4538
Peso 23: 0.0426
Peso 24: 0.6874
Peso 25: -0.9870
Peso 26: 0.0758
Peso 27: 0.5643
Peso 28: 0.8755
Peso 29: -0.3881
Peso 30: -0.6890
Peso 31: -0.4369
Peso 32: 0.9144
Peso 33: -0.3268
Peso 34: -0.8145
Peso 35: -0.8066
Peso 36: 0.0000
Peso 37: 1.3485
Peso 38: -0.0595
Peso 39: 1.0784
Peso 40: 0.4595
Peso 41: -0.9130

Epoch 199, Backward pass

Gradientes después de backward propagation:

Gradiente del peso 1: -0.0046
Gradiente del peso 2: 0.0011
Gradiente del peso 3: -0.0006
Gradiente del peso 4: -0.0011
Gradiente del peso 5: -0.0054
Gradiente del peso 6: -0.0113
Gradiente del peso 7: -0.0008
Gradiente del peso 8: -0.0028
Gradiente del peso 9: 0.0044
Gradiente del peso 10: 0.0001
Gradiente del peso 11: 0.0036
Gradiente del peso 12: 0.0012
Gradiente del peso 13: 0.0000
Gradiente del peso 14: 0.0000

Gradiente del peso 15: 0.0000
Gradiente del peso 16: 0.0000
Gradiente del peso 17: 0.0000
Gradiente del peso 18: -0.0043
Gradiente del peso 19: -0.0069
Gradiente del peso 20: 0.0000
Gradiente del peso 21: -0.0030
Gradiente del peso 22: 0.0086
Gradiente del peso 23: -0.0040
Gradiente del peso 24: 0.0124
Gradiente del peso 25: 0.0000
Gradiente del peso 26: -0.0068
Gradiente del peso 27: 0.0000
Gradiente del peso 28: -0.0115
Gradiente del peso 29: -0.0000
Gradiente del peso 30: 0.0000
Gradiente del peso 31: -0.0045
Gradiente del peso 32: 0.0000
Gradiente del peso 33: 0.0000
Gradiente del peso 34: 0.0000
Gradiente del peso 35: 0.0000
Gradiente del peso 36: 0.0000
Gradiente del peso 37: -0.0033
Gradiente del peso 38: -0.0664
Gradiente del peso 39: -0.0075
Gradiente del peso 40: 0.0000
Gradiente del peso 41: 0.1135

Pesos en la backward propagation en la iteración 199:

Peso 1: 0.5862
Peso 2: -1.1550
Peso 3: -0.3241
Peso 4: 0.0652
Peso 5: -0.1636
Peso 6: 0.9074
Peso 7: -0.3656
Peso 8: -0.4145
Peso 9: 1.0388
Peso 10: -0.8407
Peso 11: -0.5564
Peso 12: 0.0681
Peso 13: -0.9404
Peso 14: -0.5627
Peso 15: 0.0107
Peso 16: 0.0000
Peso 17: -1.2822
Peso 18: -0.5239
Peso 19: 0.8567
Peso 20: 0.0899
Peso 21: 0.2560
Peso 22: -0.4539
Peso 23: 0.0426
Peso 24: 0.6872
Peso 25: -0.9870
Peso 26: 0.0758
Peso 27: 0.5643
Peso 28: 0.8756
Peso 29: -0.3881
Peso 30: -0.6890
Peso 31: -0.4369
Peso 32: 0.9144
Peso 33: -0.3268

Peso 34: -0.8145
Peso 35: -0.8066
Peso 36: 0.0000
Peso 37: 1.3485
Peso 38: -0.0588
Peso 39: 1.0785
Peso 40: 0.4595
Peso 41: -0.9141

Epoch 200, Forward pass

Pérdida: 0.0017316279062231852

Pesos en la forward propagation en la iteración 200:

Peso 1: 0.5862
Peso 2: -1.1550
Peso 3: -0.3241
Peso 4: 0.0652
Peso 5: -0.1636
Peso 6: 0.9074
Peso 7: -0.3656
Peso 8: -0.4145
Peso 9: 1.0388
Peso 10: -0.8407
Peso 11: -0.5564
Peso 12: 0.0681
Peso 13: -0.9404
Peso 14: -0.5627
Peso 15: 0.0107
Peso 16: 0.0000
Peso 17: -1.2822
Peso 18: -0.5239
Peso 19: 0.8567
Peso 20: 0.0899
Peso 21: 0.2560
Peso 22: -0.4539
Peso 23: 0.0426
Peso 24: 0.6872
Peso 25: -0.9870
Peso 26: 0.0758
Peso 27: 0.5643
Peso 28: 0.8756
Peso 29: -0.3881
Peso 30: -0.6890
Peso 31: -0.4369
Peso 32: 0.9144
Peso 33: -0.3268
Peso 34: -0.8145
Peso 35: -0.8066
Peso 36: 0.0000
Peso 37: 1.3485
Peso 38: -0.0588
Peso 39: 1.0785
Peso 40: 0.4595
Peso 41: -0.9141

Epoch 200, Backward pass

Gradientes después de backward propagation:

Gradiente del peso 1: -0.0045
Gradiente del peso 2: 0.0011
Gradiente del peso 3: -0.0006
Gradiente del peso 4: -0.0011
Gradiente del peso 5: 0.0707

Gradiente del peso 6: 0.2168
Gradiente del peso 7: 0.1512
Gradiente del peso 8: 0.1492
Gradiente del peso 9: 0.0044
Gradiente del peso 10: 0.0001
Gradiente del peso 11: 0.0035
Gradiente del peso 12: 0.0012
Gradiente del peso 13: 0.0000
Gradiente del peso 14: 0.0000
Gradiente del peso 15: 0.0000
Gradiente del peso 16: 0.0000
Gradiente del peso 17: 0.0000
Gradiente del peso 18: -0.0042
Gradiente del peso 19: -0.0067
Gradiente del peso 20: 0.0000
Gradiente del peso 21: -0.0029
Gradiente del peso 22: 0.0085
Gradiente del peso 23: -0.0039
Gradiente del peso 24: 0.0122
Gradiente del peso 25: 0.0000
Gradiente del peso 26: -0.0066
Gradiente del peso 27: 0.0000
Gradiente del peso 28: 0.0754
Gradiente del peso 29: -0.0000
Gradiente del peso 30: 0.0000
Gradiente del peso 31: 0.1691
Gradiente del peso 32: 0.0000
Gradiente del peso 33: 0.0000
Gradiente del peso 34: 0.0000
Gradiente del peso 35: 0.0000
Gradiente del peso 36: 0.0000
Gradiente del peso 37: -0.0032
Gradiente del peso 38: -0.0659
Gradiente del peso 39: -0.0073
Gradiente del peso 40: 0.0000
Gradiente del peso 41: 0.1124

Pesos en la backward propagation en la iteración 200:

Peso 1: 0.5863
Peso 2: -1.1550
Peso 3: -0.3241
Peso 4: 0.0652
Peso 5: -0.1643
Peso 6: 0.9053
Peso 7: -0.3671
Peso 8: -0.4160
Peso 9: 1.0387
Peso 10: -0.8408
Peso 11: -0.5564
Peso 12: 0.0680
Peso 13: -0.9404
Peso 14: -0.5627
Peso 15: 0.0107
Peso 16: 0.0000
Peso 17: -1.2822
Peso 18: -0.5239
Peso 19: 0.8568
Peso 20: 0.0899
Peso 21: 0.2560
Peso 22: -0.4539
Peso 23: 0.0427
Peso 24: 0.6871

Peso 25: -0.9870
Peso 26: 0.0759
Peso 27: 0.5643
Peso 28: 0.8749
Peso 29: -0.3881
Peso 30: -0.6890
Peso 31: -0.4386
Peso 32: 0.9144
Peso 33: -0.3268
Peso 34: -0.8145
Peso 35: -0.8066
Peso 36: 0.0000
Peso 37: 1.3486
Peso 38: -0.0581
Peso 39: 1.0786
Peso 40: 0.4595
Peso 41: -0.9152

Predicciones finales:

Entrada: [2.5, 3.5, -0.5], Predicción: 0.9837, Objetivo: 1.0
Entrada: [4.0, -1.0, 0.5], Predicción: -1.0210, Objetivo: -1.0
Entrada: [0.5, 1.5, 1.0], Predicción: -0.9209, Objetivo: -1.0
Entrada: [3.0, 2.0, -1.5], Predicción: 1.0031, Objetivo: 1.0