

▼ Gradiente Descendente

O gradiente descendente é um algoritmo de otimização que usa as derivadas da função objetivo para encontrar o ponto com maior inclinação. No processo, as variáveis a otimizar são deslocadas em uma direção negativa o qual reduzirá o valor da função objetivo.

Algoritmo geral para atualizar os pesos com gradiente descendente:



▼ Vamos implementar o algoritmo do Gradiente Descendente!

Importando a biblioteca

```
import numpy as np
```

▼ Função do cálculo da sigmóide

```
def sigmoid(x):  
    return 1/(1+np.exp(-x))
```

▼ Derivada da função sigmóide

```
def sigmoid_prime(x):  
    return sigmoid(x) * (1 - sigmoid(x))
```

▼ Vetor dos valores de entrada e saídas

```
x = np.array([4, 2, 3, 1])  
y = np.array(0.5)  
b = 0.5
```

▼ Pesos iniciais das ligações sinápticas

Nota: Inicializados aleatoriamente

```
w = np.array([-0.5, 0.4, 0.2, 0.1])
```



Calcule um degrau de descida gradiente para cada peso

Critério de parada

- **Epochs:** Número de Épocas
- **MinError:** Erro mínimo estipulado

```
#Número de Épocas  
epochs= 150
```

```
#Inicilizando del_w  
del_w= 0
```

```
for e in range(epochs):  
    # TODO: Calcule a combinação linear de entradas e pesos sinápticos  
    h = np.dot(x, w)  
  
    # TODO: Calcule a saída da Rede Neural  
    output = sigmoid(h)  
  
    # TODO: Calculate error of neural network  
    error = y - output  
  
    # TODO: Calcule o termo de erro  
    error_term = error * sigmoid_prime(h)  
  
    # TODO: Calcule a variação do peso  
    del_w += learnrate * error_term * x  
  
    # TODO: Atualização do Peso  
    w = w + del_w  
  
    print(error)
```

```
0.1224593312018546  
0.017074367991536665  
-0.10523021553925405  
-0.1331610394511129  
-0.04992348602669039  
0.08186288103872164
```

```
0.1369464159069244
0.07636222640918172
-0.05628520346704402
-0.13452085852786644
-0.10091609502426313
0.02400977269931437
0.1252003685701355
0.11944302728834877
0.010083244939308011
-0.10928428170818516
-0.13151220270255815
-0.04340604065366027
0.08713424367679473
0.1367982037433928
0.07352903928797522
-0.05940225344921224
-0.13509161637031208
-0.09866283085165473
0.027449613830077435
0.1264667728191607
0.1178325343000815
0.0065738693031273665
-0.1112126264784643
-0.1305802330570175
-0.040094654418297515
0.08968117287901295
0.1365759856834306
0.07064247716857719
-0.06247406271575895
-0.13558923785605503
-0.0963462305430185
0.030867243128767763
0.1276632885041774
0.11615428671612998
0.0030591034402147055
-0.11307461759661652
-0.1295770278411993
-0.03675137678739049
0.09216763766642
```

```
print('Saída da Rede Neural:')
print(output)
```

