ez

1 + ez

x1

x2

x3

+1

z

z = wT . x + b

y = sigmoid(z) =

11

w1

w2

w3

b1

Σ

Summing Function

Sigmoid Function

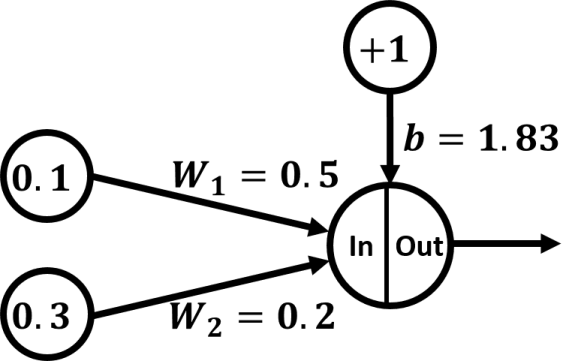
Multiply by weights

ŷ = P(Y=1|x,w)

**X1=0.1**

**X2=0.3**

**Desired Output** 0.03



Train the network 🡺 how the network will predict the output (feed +backward)

**1.Feed forward**

z=X1\* W1+ X2\*W2+b

z=0.1\* 0.5+ 0.3\*0.2+1.83

z=1.94

The value 1.94 is then applied to the activation function (sigmoid), which results in the value 0.874352143.

🡺 the predicted is 0.874352143

The desired is 0.03

🡺Error =1/2(0.03-0.874352143) 2

🡺Error = 0.357

|  |  |
| --- | --- |
| *W(n+1)=W(n)+η[d(n)-Y(n)]X(n)*  Where:   * n: Training step (0, 1, 2, …). * W(n): Parameters in current training step. Wn=[bn,W1(n),W2(n),W3(n),…, Wm(n)] * *η*: Learning rate with a value between 0.0 and 1.0. * d(n): Desired output. * Y(n): Predicted output. * X(n): Current input at which the network made false prediction. | For our network, these parameters have the following values(for step 0):   * n: 0 * W(n): [1.83, 0.5, 0.2] * *η*: Because it is a hyperparameter, then we can choose it 0.01 for example. * d(n): [0.03]. * Y(n): [0.874352143]. * X(n): [+1, 0.1, 0.3]. First value (+1) is for the bias. |

*W(n+1)=W(n)+η[d(n)-Y(n)]X(n)*

*=[1.83, 0.5, 0.2]+0.01[0.03-0.874352143][+1, 0.1, 0.3]*

*=[1.83, 0.5, 0.2]+0.01[-0.844352143][+1, 0.1, 0.3]*

*=[1.83, 0.5, 0.2]+-0.00844352143[+1, 0.1, 0.3]*

*=[1.83, 0.5, 0.2]+[-0.008443521, -0.000844352, -0.002533056]*

*=[1.821556479, 0.499155648, 0.197466943]*

The new parameters are listed in the next table:

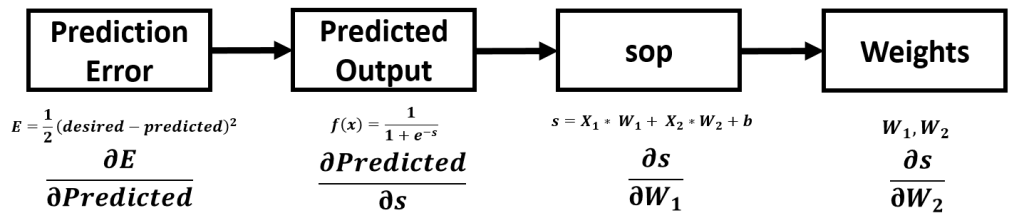
**W1new=**0.197466943

**W2new=**0.499155648

**Bnew=**1.821556479

**2-using backpropagation**

using the backpropagation algorithm🡺 how each single weight correlates with the error.



∂E/∂Predicted***,***∂Predicted/∂s***,***∂s/W1***and***∂s/W2

To calculate the derivative of the error W.R.T the weights, simply multiply all the derivatives in the chain from the error to each weight

∂E/W1=∂E/∂Predicted\* ∂Predicted/∂s\* ∂s/W1

∂EW2=∂E/∂Predicted\* ∂Predicted/∂s\* ∂s/W2

For the derivative of the error W.R.T the predicted output:

*∂E/∂Predicted=∂/∂Predicted(1/2(desired-predicted)2)*

*=2\*1/2(desired-predicted)2-1\*(0-1)*

*=(desired-predicted)\*(-1)*

*=predicted-desired*

By substituting by the values:

*∂E/∂Predicted=predicted-desired=0.874352143-0.03*

*∂E/∂Predicted=0.844352143*

For the derivative of the predicted output W.R.T the SOP:

*∂Predicted/∂s=∂/∂s(1/(1+e-s))*

Remember: the quotient rule can be used to find the derivative of the sigmoid function as follows:

*∂Predicted/∂s=1/(1+e-s)(1-1/(1+e-s))*

By substituting by the values:

*∂Predicted/∂s=1/(1+e-s)(1-1/(1+e-s))*=*1/(1+e-1.94)(1-1/(1+e-1.94))*

*=1/(1+0.143703949)(1-1/(1+0.143703949))*

*=1/1.143703949(1-1/1.143703949)*

*=0.874352143(1-0.874352143)*

*=0.874352143(0.125647857)*

*∂Predicted/∂s=0.109860473*

For the derivative of SOP W.R.T W1:

*∂s/W1=∂*/*∂W1(X1\* W1+ X2\*W2+b)*

*=1\*X1\*(W1)(1-1)+ 0+0*

*=X1\*(W1)(0)*

*=X1(1)*

*∂s/W1=X1*

By substituting by the values:

∂s/W1=X1=0.1

====================================================

For the derivative of SOP W.R.T W2:

*∂s/W2=∂*/*∂W2(X1\* W1+ X2\*W2+b)*

*=0+1\*X2\*(W2)(1-1)+0*

*=X2\*(W2)(0)*

*=X2(1)*

*∂s/W2=X2*

By substituting by the values:

*∂s/W2=X2=0.3*

*Multiply …..*

For the derivative of the error W.R.T W1:

*∂E/W1=0.844352143\*0.109860473\*0.1*

*∂E/W1=0.009276093*

For the derivative of the error W.R.T W2:

*∂E/W2=0.844352143\*0.109860473\*0.3*

*∂E/W2=0.027828278*

*===============================🡺*the prediction error changes with respect to the weights:

*0.009276093 for W1*

*0.027828278 for W2*

* Because the result of the ∂E/W1 derivative is positive, this means if W1 increases by 1, then the total error increases by 0.009276093.
* Because the result of the ∂E/W2 derivative is positive, this means that if W2 increases by 1 then the total error increases by 0.027828278.

**Updating weights**

For W1:

*W1new=W1-η\*∂E/W1*

*=0.5-0.01\*0.009276093*

*W1new=0.49990723907*

For W2:

*W2new=W2-η\*∂E/W2*

*=0.2-0.01\*0.027828278*

*W2new= 0.1997217172*

**Training with new w values**

Here are the new forward pass calculations:

*s=X1\* W1+ X2\*W2+b*

*s=0.1\*0.49990723907+ 0.3\*0.1997217172+1.821556479*

*s=1.931463718067*

*f(s)=1/(1+e-s)*

*f(s)=1/(1+e-1.931463718067)*

*f(s)=0.873411342830056*

*E=1/2(0.03-0.873411342830056)2*

*E=0.35567134660719907*