```
% Aditya Agre
% SYCOA06
% Delta Learning rule
clear All

% Consider below Input vector without bias
x = [0 0 1; 0 1 1; 1 0 1;1 1 1]
```

```
x = 4×3
     0     0     1
     0     1     1
     1     0     1
     1     1     1
```

```
% Target Output
t = [0 0 1 1]
```

```
t = 1×4
     0     0     1     1
```

```
w = [rand(1,3)] % Three weights as three input
```

```
w = 1×3
    0.5529    0.0185    0.5078
```

```
for i = 1:4     % using each vector from x one by one
    z =0;
    for j= 1:3  % Going over all three values in x and 3 values in w
        z=z+ (x(i,j)*w(j))
    end
    y = 1/(1+exp(-z))
    error = t(i)-y
    diff = derivative(z)
    for k= 1:3   % updating each weight one by one
        w(k) =  w(k) + (error * diff * x(i,k))
    end
end
```

```
z = 0
z = 0
z = 0.5078
y = 0.6243
error = -0.6243
diff = 0.2346
diff = 0.2346
w = 1×3
    0.5529    0.0185    0.5078
w = 1×3
    0.5529    0.0185    0.5078
w = 1×3
    0.5529    0.0185    0.3614
z = 0
z = 0.0185
```

```
z = 0.3799
y = 0.5939
error = -0.5939
diff = 0.2412
diff = 0.2412
w = 1×3
    0.5529    0.0185    0.3614
w = 1×3
    0.5529   -0.1247    0.3614
w = 1×3
    0.5529   -0.1247    0.2182
z = 0.5529
z = 0.5529
z = 0.7711
y = 0.6838
error = 0.3162
diff = 0.2162
diff = 0.2162
w = 1×3
    0.6213   -0.1247    0.2182
w = 1×3
    0.6213   -0.1247    0.2182
w = 1×3
    0.6213   -0.1247    0.2865
z = 0.6213
z = 0.4966
z = 0.7832
y = 0.6864
error = 0.3136
diff = 0.2153
diff = 0.2153
w = 1×3
    0.6888   -0.1247    0.2865
w = 1×3
    0.6888   -0.0572    0.2865
w = 1×3
    0.6888   -0.0572    0.3541
```

```
"\n Final weights "
```

```
ans =
"\n Final weights "
```

```
W
```

```
w = 1×3
    0.6888   -0.0572    0.3541
```

```
function diff =derivative(z)
diff = exp(-z)/((1+exp(-z))^2)
end
```