Bias Updating

TLU (Threshold Logic Unit)

In NP neuron for any gate, if the summation of their inputs should be greater than the bias for the output to be 1. If this is not the case then we need to modify the bias to get the correct output, we can check the output from the truth table and modify the bias accordingly.

How to modify the bias

1. If the TLU function returns 1 this means that the bias is smaller is than the summation but if for the same inputs the answer in truth table is 0 this means here we need to update the bias. Now, the value of the bias is smaller than the summation so for the answer to be correct, we need to increase the value of the bias.
2. Similiarly if the TLU function returns 0 this means that the bias is greater is than the summation but if for the same inputs the answer in truth table is 1 this means here we need to update the bias. Now, the value of the bias is bigger than the summation so for the answer to be correct, we need to decrease the value of the bias.

image.png

```
import numpy as np

x = np.array([[0,0], [0,1], [1,0], [1,1]])
y = np.array([[0],[1],[1],[1]])

# step 2
# inp = aggregated input summation Wi
def TLU(inp,b):
  if sum(inp) > b:
    return 1
  else:
    return 0

# step 1
b = np.random.randint(-10,10)
print("Initial value of b: ", b)
for epoch in range(15): # take number of epochs between the range
  for i,j in zip(x,y):
    if j==1 and TLU(i,b) == 0: # truth table says 1 aaya hai and TLU says 0 aaya hai, this means bias is bigger than sum so we decreade the bias.
      b = b-1
    elif j==0 and TLU(i,b) == 1:
      b = b+1
  print("epoch = ", epoch +1)

print("Final value of b: ", TLU([0,0],b))
print("Final value of b: ", TLU([0,1],b))
print("Final value of b: ", TLU([1,0],b))
print("Final value of b: ", TLU([1,1],b))
```

```
Initial value of b:  -7
epoch =  1
epoch =  2
epoch =  3
epoch =  4
epoch =  5
epoch =  6
epoch =  7
epoch =  8
epoch =  9
epoch =  10
epoch =  11
epoch =  12
epoch =  13
epoch =  14
epoch =  15
Final value of b:  0
Final value of b:  1
Final value of b:  1
Final value of b:  1
```

```
import numpy as np

# and gate
x = np.array([[0,0], [0,1], [1,0], [1,1]])
y = np.array([[0],[0],[0],[1]])

# step 2
# inp = aggregated input summation Wi
def TLU(inp,b):
  if sum(inp) > b:
    return 1
  else:
    return 0

# step 1
b = np.random.randint(-10,10)
print("Initial value of b: ", b)
for epoch in range(15): # take number of epochs between the range
  for i,j in zip(x,y):
    if j==1 and TLU(i,b) == 0: # truth table says 1 aaya hai and TLU says 0 aaya hai, this means bias is bigger than sum so we decreade the bias.
      b = b-1
    elif j==0 and TLU(i,b) == 1:
      b = b+1
  print("epoch = ", epoch +1)

print("Final value of b: ", TLU([0,0],b))
print("Final value of b: ", TLU([0,1],b))
print("Final value of b: ", TLU([1,0],b))
print("Final value of b: ", TLU([1,1],b))
```

```
Initial value of b:  2
epoch =  1
epoch =  2
epoch =  3
epoch =  4
epoch =  5
epoch =  6
epoch =  7
epoch =  8
epoch =  9
epoch =  10
epoch =  11
epoch =  12
epoch =  13
epoch =  14
epoch =  15
Final value of b:  0
```

```
     Final value of b:  0
     Final value of b:  0
     Final value of b:  1
```

3 input OR Gate

```
import numpy as np

x = np.array([[0,0,0], [0,0,1], [0,1,0], [0,1,1], [1,0,0],[1,0,1],[1,1,0]])
y = np.array([[0],[1],[1],[1],[1],[1]])

# step 2
# inp = aggregated input summation Wi
def TLU(inp,b):
  if sum(inp) > b:
    return 1
  else:
    return 0

# step 1
b = np.random.randint(-10,10)
print("Initial value of b: ", b)
for epoch in range(15): # take number of epochs between the range
  for i,j in zip(x,y):
    if j==1 and TLU(i,b) == 0: # truth table says 1 aaya hai and TLU says 0 aaya hai, this means bias is bigger than sum so we decreade the bias.
      b = b-1
    elif j==0 and TLU(i,b) == 1:
      b = b+1
  print("epoch = ", epoch +1)

print("Final value of b: ", TLU(x[2],b))
```

```
     Initial value of b:  -1
     epoch =  1
     epoch =  2
     epoch =  3
     epoch =  4
     epoch =  5
     epoch =  6
     epoch =  7
     epoch =  8
     epoch =  9
     epoch =  10
     epoch =  11
     epoch =  12
     epoch =  13
     epoch =  14
     epoch =  15
     Final value of b:  1
```

3 input AND Gate

```
import numpy as np

x = np.array([[0,0,0], [0,0,1], [0,1,0], [0,1,1], [1,0,0],[1,0,1],[1,1,1]])
y = np.array([[0],[0],[0],[0],[0],[0], [1]])

# step 2
# inp = aggregated input summation Wi
def TLU(inp,b):
  if sum(inp) > b:
    return 1
  else:
    return 0

# step 1
b = np.random.randint(-10,10)
print("Initial value of b: ", b)
for epoch in range(15): # take number of epochs between the range
  for i,j in zip(x,y):
    if j==1 and TLU(i,b) == 0: # truth table says 1 aaya hai and TLU says 0 aaya hai, this means bias is bigger than sum so we decreade the bias.
      b = b-1
    elif j==0 and TLU(i,b) == 1:
      b = b+1
  print("epoch = ", epoch +1)

print("Final value of b: ", TLU(x[1],b), "Target (y): ", y[1])
print("Final value of b: ", TLU(x[6],b), "Target (y): ", y[6])
```

```
     Initial value of b:  -6
     epoch =  1
     epoch =  2
     epoch =  3
     epoch =  4
     epoch =  5
     epoch =  6
     epoch =  7
     epoch =  8
     epoch =  9
     epoch =  10
     epoch =  11
     epoch =  12
     epoch =  13
     epoch =  14
     epoch =  15
     Final value of b:  0 Target (y):  [0]
     Final value of b:  1 Target (y):  [1]
```