Name:KALYANAM VENKATA SREE SAI

ID:2000030439

SEC:01

CourseCode:20cs3026RA

In [1]:
```
import math
import numpy as np
def computeNet(input, weights):
        net = 0
        for i in range(len(input)):
                net = net + input[i]*weights[i]
        print("NET:")
        print(net)
        return net

def computeFNetBinary(net):
        f_net = 0
        if(net>0):
                f_net = 1
        if(net<0):
                f_net = -1
        return f_net

def computeFNetCont(net):
        f_net = 0
        f_net = (2/(1+math.exp(-net)))-1
        return f_net

def hebb(f_net):
        return f_net

def perceptron(desired, actual):
    return (desired-actual)

def widrow(desired, actual):
    return (desired-actual)

def adjustWeights(inputs, weights, last, binary, desired, rule):
    c = 1
    if(last):
        print("COMPLETE")
        return
    current_input = inputs[0]
    inputs = inputs[1:]
    if desired :
        current_desired = desired[0]
        desired = desired[1:]
    if len(inputs) == 0:
        last = True
    net = computeNet(current_input, weights)
    if(binary):
        f_net = computeFNetBinary(net)
```

```
    else:
        f_net = computeFNetCont(net)
    if rule == "hebb":
        r = hebb(f_net)
    elif rule == "perceptron":
        r = perceptron(current_desired, f_net)
    elif rule == "widrow":
        r = widrow(current_desired, net)
    del_weights = []
    for i in range(len(current_input)):
        x = (c*r)*current_input[i]
        del_weights.append(x)
        weights[i] = x
    print("NEW WEIGHTS:")
    print(weights)
    adjustWeights(inputs, weights, last, binary, desired, rule)

if __name__=="__main__":
    total_inputs = 3
    vector_length = 4
    weights = np.array([1,-1,0,0.5]).transpose()
    inputs = [np.array([1,-2,1.5,0]).transpose(),np.array([1,-0.5,-2,-1.5]).transpose(),
    desired = [1,2,1,-1]
    adjustWeights(inputs, [1,-1,0,0.5], False, True, None, "hebb")
    adjustWeights(inputs, [1,-1,0,0.5], False, False, None, "hebb")
    adjustWeights(inputs, [1,-1,0,0.5], False, True, desired, "perceptron")
    adjustWeights(inputs, [1,-1,0,0.5], False, True, desired, "widrow")
```

```
NET:
3.0
NEW WEIGHTS:
[1.0, -2.0, 1.5, 0.0]
NET:
-1.0
NEW WEIGHTS:
[-1.0, 0.5, 2.0, 1.5]
NET:
0.75
NEW WEIGHTS:
[0.0, 1.0, -1.0, 1.5]
COMPLETE
NET:
3.0
NEW WEIGHTS:
[0.9051482536448667, -1.8102965072897335, 1.3577223804673002, 0.0]
NET:
-0.905148253644867
NEW WEIGHTS:
[-0.42401264054072996, 0.21200632027036498, 0.8480252810814599, 0.6360189608110949]
NET:
0.31800948040554744
NEW WEIGHTS:
[0.0, 0.15767814164392502, -0.15767814164392502, 0.23651721246588753]
COMPLETE
NET:
3.0
NEW WEIGHTS:
[0.0, -0.0, 0.0, 0.0]
NET:
0.0
NEW WEIGHTS:
[2.0, -1.0, -4.0, -3.0]
NET:
-1.5
NEW WEIGHTS:
```

```
[0.0, 2.0, -2.0, 3.0]
COMPLETE
NET:
3.0
NEW WEIGHTS:
[-2.0, 4.0, -3.0, -0.0]
NET:
2.0
NEW WEIGHTS:
[0.0, -0.0, -0.0, -0.0]
NET:
0.0
NEW WEIGHTS:
[0.0, 1.0, -1.0, 1.5]
COMPLETE
```