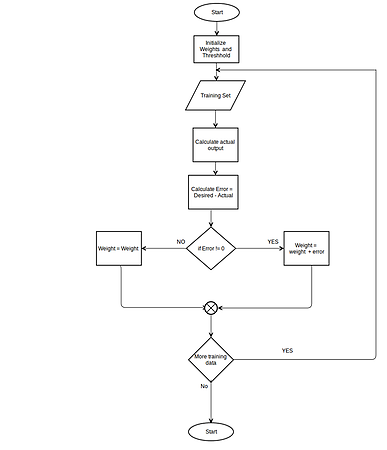
**PERCEPTRON**

**Flowchart**



**Algorithm and implementation**

threshold = 0.5

learning\_rate = 0.1

weights = [0, 0, 0]

training\_set = [((1, 0, 0), 1), ((1, 0, 1), 1), ((1, 1, 0), 1), ((1, 1, 1), 0)]

def dot\_product(values, weights):

return sum(value \* weight for value, weight in zip(values, weights))

while True:

#print the seperation line

print('-' \* 60)

error\_count = 0

for input\_vector, desired\_output in training\_set:

#printing the initial weights

print("Initial Weights: {}". format(weights))

result = dot\_product(input\_vector, weights) > threshold

#calculating error

error = desired\_output - result

print("Error: {}". format(error))

#if there is an error adjust the weights

if error != 0:

error\_count += 1

#for every input in the input vector calculate the new weights

for index, value in enumerate(input\_vector):

weights[index] += learning\_rate \* error \* value

print("Final Weights: {}". format(weights))

#if there was no input with an error terminate the loop

if error\_count == 0:

break

**Screen shots**

