2000080110_ML Skill7

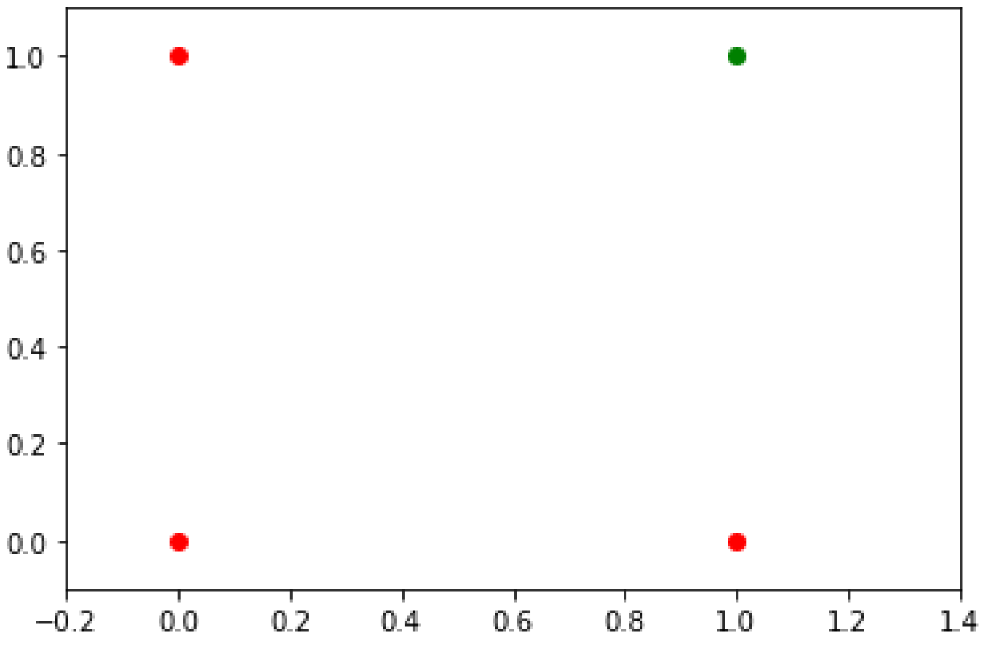
September 2, 2021

[1]:

```
#Graph for AND Gate---if true,false can be seperated through a line then they␣
 ↪are linear seperable
import matplotlib.pyplot as plt
import numpy as np
fig, ax = plt.subplots()
xmin, xmax = -0.2, 1.4
X = np.arange(xmin, xmax, 0.1)
#AND gate
ax.scatter(0, 0, color="r")#0
ax.scatter(0, 1, color="r")#0
ax.scatter(1, 0, color="r")#0
ax.scatter(1, 1, color="g")#1
ax.set_xlim([xmin, xmax])
ax.set_ylim([-0.1, 1.1])
m = -1
#ax.plot(X, m * X + 1.2, label="decision boundary")
plt.plot()
```

[1]:

```
[]
```

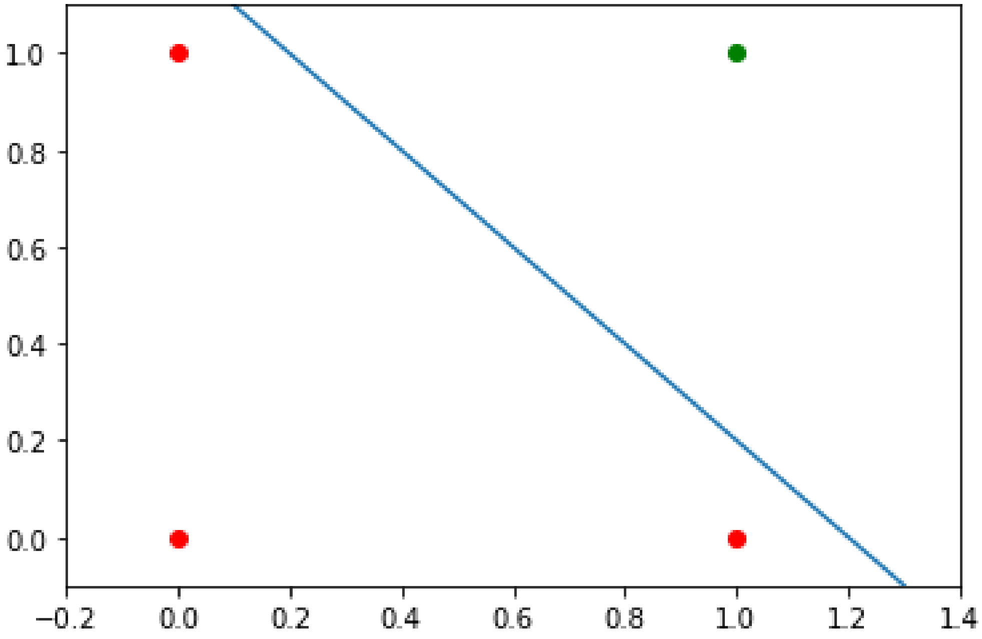


[2]:

```
#linear seperable-AND gate
import matplotlib.pyplot as plt
import numpy as np
fig, ax = plt.subplots()
xmin, xmax = -0.2, 1.4
X = np.arange(xmin, xmax, 0.1)
ax.scatter(0, 0, color="r")
ax.scatter(0, 1, color="r")
ax.scatter(1, 0, color="r")
ax.scatter(1, 1, color="g")
ax.set_xlim([xmin, xmax])
ax.set_ylim([-0.1, 1.1])
m, c = -1, 1.2
ax.plot(X, m * X + c )
plt.plot()
```

[2]:

```
[]
```

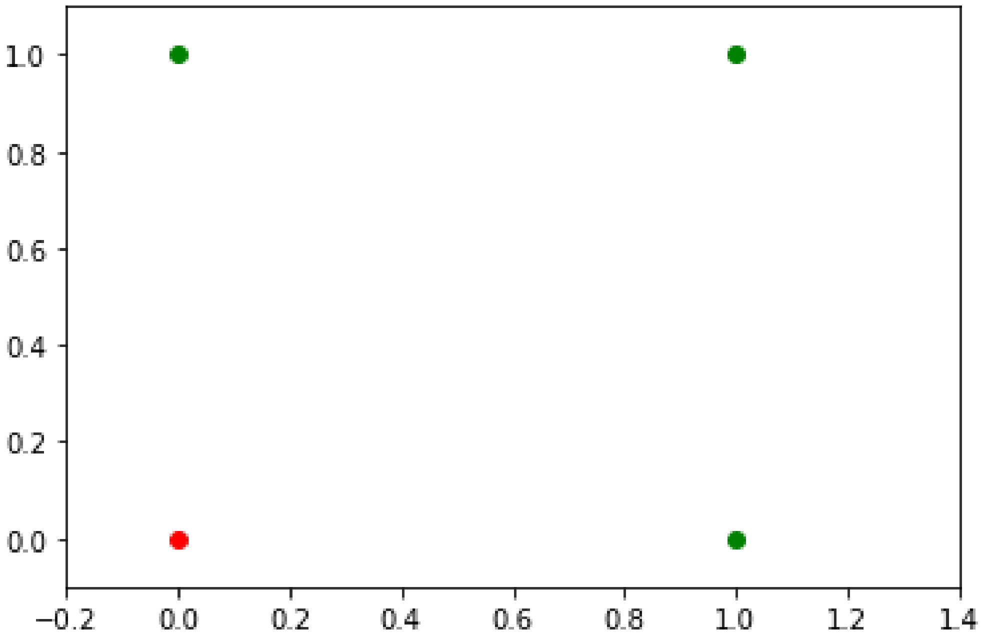


[5]:

```
#Graph for OR Gate---if true,false can be seperated through a line then they␣
 ↪are linear seperable
import matplotlib.pyplot as plt
import numpy as np
fig, ax = plt.subplots()
```

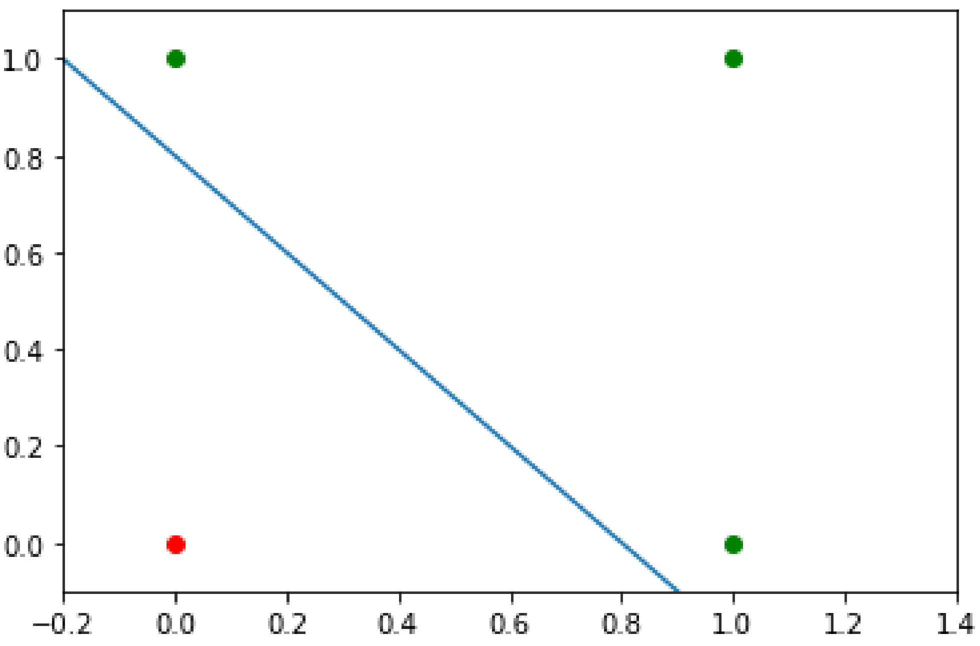
```
xmin, xmax = -0.2, 1.4
X = np.arange(xmin, xmax, 0.1)
#AND gate
ax.scatter(0, 0, color="r")#0
ax.scatter(0, 1, color="g")#1
ax.scatter(1, 0, color="g")#1
ax.scatter(1, 1, color="g")#1
ax.set_xlim([xmin, xmax])
ax.set_ylim([-0.1, 1.1])
m = -1
plt.plot()
```

```
[5]: []
```



```
[20]: #linear seperable-OR gate
import matplotlib.pyplot as plt
import numpy as np
fig, ax = plt.subplots()
xmin, xmax = -0.2, 1.4
X = np.arange(xmin, xmax, 0.1)
ax.scatter(0, 0, color="r")
ax.scatter(0, 1, color="g")
ax.scatter(1, 0, color="g")
ax.scatter(1, 1, color="g")
ax.set_xlim([xmin, xmax])
ax.set_ylim([-0.1, 1.1])
m, c = -1, 0.8
ax.plot(X, m * X + c )
plt.plot()
```

[20]: []



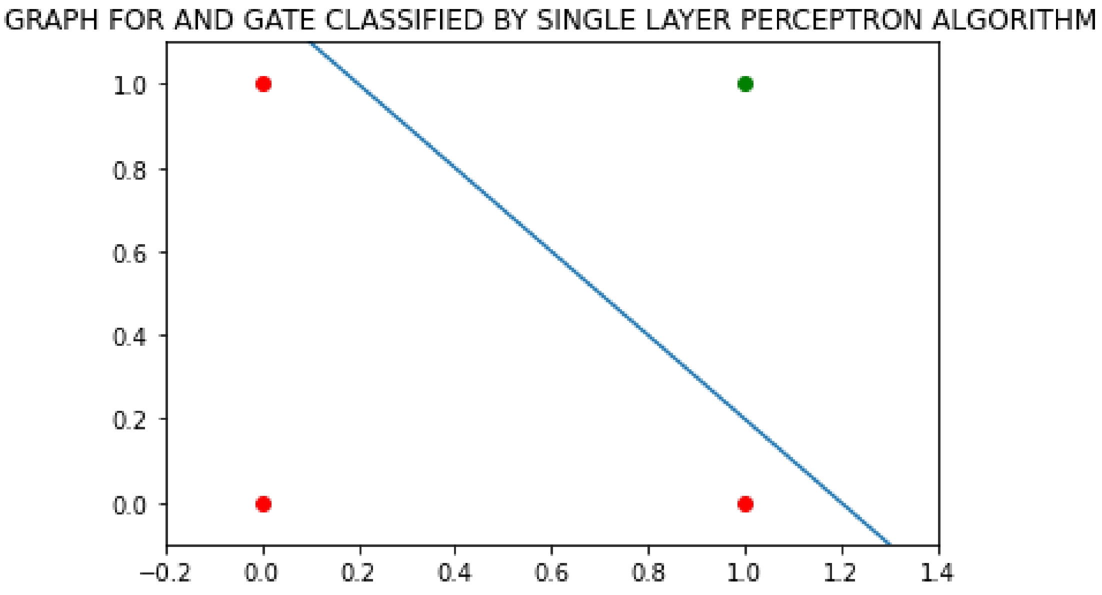
[43]:

```
#apply single layer perceptron algorithm for AND gate
#actual outputs are 0 0 0 1 for AND gate
fig, ax = plt.subplots()
x=[0,1,0,1]
y=[0,0,1,1]
w=[1,1]#let
#from activation func if sum(wixi)<0 output is 0 and if >=0 output is 1
res=[]
bias=-1
for i in range(4):
    res.append((w[0]*x[i]+w[1]*y[i])+bias)
#apply sigmoid fun
for i in range(len(res)):
    if res[i]<=0:
        res[i]=0
    else:
        res[i]=1
print("Outputs will be",res)
print("Equation is x1+x2-1")
xmin, xmax = -0.2, 1.4
for i in range(len(res)):
    if res[i]==0:
        c='r'
    else:
        c='g'
    ax.scatter(x[i],y[i],color=c)
ax.set_xlim([xmin, xmax])
ax.set_ylim([-0.1, 1.1])
m, c = -1, 1.2
ax.plot(X, m * X + c )
plt.plot()
plt.title("GRAPH FOR AND GATE CLASSIFIED BY SINGLE LAYER PERCEPTRON ALGORITHM")
```

```
Outputs will be [0, 0, 0, 1]
Equation is x1+x2-1
```

[43]:

```
Text(0.5, 1.0, 'GRAPH FOR AND GATE CLASSIFIED BY SINGLE LAYER PERCEPTRON
ALGORITHM')
```



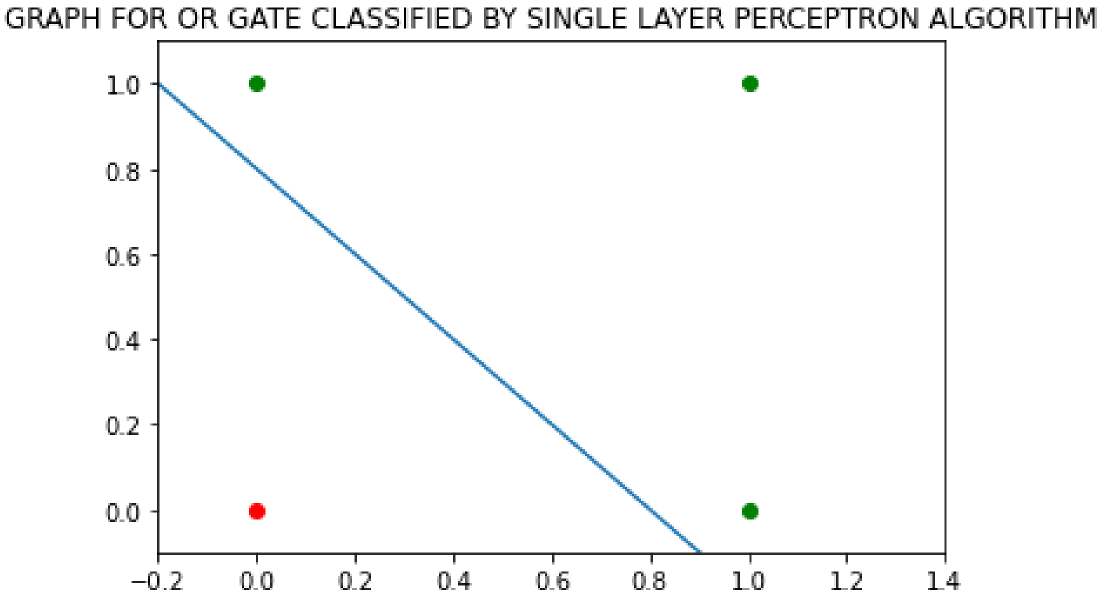
[42]:

```
#apply single layer perceptron algorithm for OR gate
#actual outputs are 0 1 1 1 for OR gate
fig, ax = plt.subplots()
x=[0,1,0,1]
y=[0,0,1,1]
w=[1,1]#let
```

```python
#from activation func if sum(wixi)<0 output is 0 and if >=0 output is 1
res=[]
bias=-1
for i in range(4):
    res.append((w[0]*x[i]+w[1]*y[i])+bias)
#apply sigmoid fun
for i in range(len(res)):
    if res[i]<=0:
        res[i]=0
    else:
        res[i]=1
print("Outputs will be",res,"which are incorrect so we need to update weights to get correct output")
print("Change equation as 2x1+2x2-1")
w=[2,2]
res=[]
bias=-1
for i in range(4):
    res.append((w[0]*x[i]+w[1]*y[i])+bias)
#apply sigmoid fun
for i in range(len(res)):
    if res[i]<=0:
        res[i]=0
    else:
        res[i]=1
xmin, xmax = -0.2, 1.4
for i in range(len(res)):
    if res[i]==0:
        c='r'
    else:
        c='g'
    ax.scatter(x[i],y[i],color=c)
ax.set_xlim([xmin, xmax])
ax.set_ylim([-0.1, 1.1])
m, c = -1, 0.8
ax.plot(X, m * X + c )
plt.plot()
plt.title("GRAPH FOR OR GATE CLASSIFIED BY SINGLE LAYER PERCEPTRON ALGORITHM")
```

```
Outputs will be [0, 0, 0, 1] which are incorrect so we need to update weights to
get correct output
Change equation as 2x1+2x2-1
```

```
[42]: Text(0.5, 1.0, 'GRAPH FOR OR GATE CLASSIFIED BY SINGLE LAYER PERCEPTRON
ALGORITHM')
```


GRAPH FOR OR GATE CLASSIFIED BY SINGLE LAYER PERCEPTRON ALGORITHM
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4