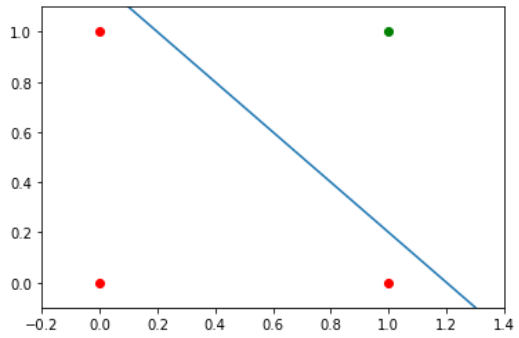
In [2]:
```
"""Implementing AND Gate through perceptron"""

import numpy as np
import matplotlib.pyplot as plt

# Define unit step function
def unit_step(v):
    if v > 0:
        return 1
    else:
        return 0

# Design Perceptron Model
def perceptron_model(x,w,b):
    v = np.dot(w,x) + b
    y = unit_step(v)
    return y

# AND Logic Function
# w1 = 1, w2 = 1 and b = -1
def AND(x):
    w = np.array([1,1])
    b = -1
    return perceptron_model(x,w,b)

# Perceptron model
test1 = np.array([0,0])
test2 = np.array([0,1])
test3 = np.array([1,0])
test4 = np.array([1,1])

print("AND({} , {} ) = {}".format(0,0, AND(test1)))
print("AND({} , {} ) = {}".format(0,1, AND(test2)))
print("AND({} , {} ) = {}".format(1,0, AND(test3)))
print("AND({} , {} ) = {}".format(1,1, AND(test4)))

# Plotting the perceptron
fig, ax = plt.subplots()
X = np.arange(-0.2, 1.4, 0.1)
ax.scatter(0,0,color = "red")
ax.scatter(0,1,color = "red")
ax.scatter(1,0,color = "red")
ax.scatter(1,1,color = "green")
ax.set_xlim([-0.2, 1.4])
ax.set_ylim([-0.1 , 1.1])
m = -1
c = 1.2
ax.plot(X, m * X + c)
plt.plot()
```

```
AND(0 , 0 ) = 0
AND(0 , 1 ) = 0
AND(1 , 0 ) = 0
AND(1 , 1 ) = 1
```

Out[2]:
```
[]
```



In [ ]: