CODE:

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```
dataset=pd.read_csv('Credit_Card_Applications.csv')
X=dataset.iloc[:,:-1].values
Y=dataset.iloc[:,-1].values
dataset.head()
```

Out[2]:

	CustomerID	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	Class
0	15776156	1	22.08	11.46	2	4	4	1.585	0	0	0	1	2	100	1213	0
1	15739548	0	22.67	7.00	2	8	4	0.165	0	0	0	0	2	160	1	0
2	15662854	0	29.58	1.75	1	4	4	1.250	0	0	0	1	2	280	1	0
3	15687688	0	21.67	11.50	1	5	3	0.000	1	1	11	1	2	0	1	1
4	15715750	1	20.17	8.17	2	6	4	1.960	1	1	14	0	2	60	159	1

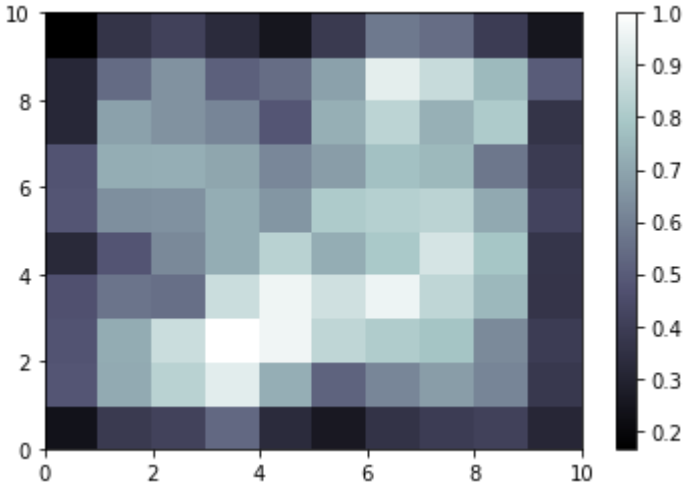
In [3]:
```
from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler(feature_range= (0,1))
X=sc.fit_transform(X)
X
```

Out[3]:
```
array([[8.42681467e-01, 1.00000000e+00, 1.25263158e-01, ...,
        5.00000000e-01, 5.00000000e-02, 1.21200000e-02],
       [6.96090562e-01, 0.00000000e+00, 1.34135338e-01, ...,
        5.00000000e-01, 8.00000000e-02, 0.00000000e+00],
       [3.88981656e-01, 0.00000000e+00, 2.38045113e-01, ...,
        5.00000000e-01, 1.40000000e-01, 0.00000000e+00],
       ...,
       [4.39420332e-01, 0.00000000e+00, 7.63909774e-02, ...,
        5.00000000e-01, 5.00000000e-02, 0.00000000e+00],
       [8.44034934e-01, 0.00000000e+00, 2.05563910e-01, ...,
        5.00000000e-01, 6.00000000e-02, 1.10000000e-04],
       [1.06907888e-01, 1.00000000e+00, 4.09774436e-01, ...,
        0.00000000e+00, 2.80000000e-01, 0.00000000e+00]])
```

In [4]:
```
from minisom import MiniSom
som =MiniSom(x=10 ,y=10 ,input_len=15, sigma=1.0 , learning_rate=0.5)
som.random_weights_init(X)
som.train_random(data=X, num_iteration=100)
```

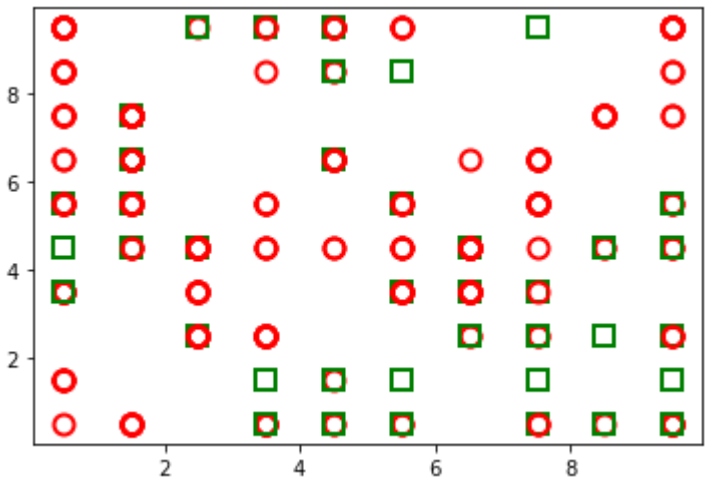
In [6]:
```
from pylab import bone,pcolor, colorbar ,plot , show
bone()
pcolor(som.distance_map().T)
colorbar()
```

Out[6]:
```
<matplotlib.colorbar.Colorbar at 0x7f8be80a6820>
```



In [9]:

```
markers = ['o', 's']
colors=['r','g']
for i,x in enumerate(X):
    w = som.winner(x)
    plot(w[0]+0.5,
         w[1]+0.5,
         markers[Y[i]],
         markeredgecolor = colors[Y[i]],
         markerfacecolor = 'None',
         markersize = 10,
         markeredgewidth = 2)
show()
```



In [10]:

```
mappings = som.win_map(X)
frauds=np.concatenate((mappings[(5,1)],mappings[(6,2)]), axis=0)
frauds = sc.inverse_transform(frauds)
frauds
```

Out[10]:

```
array([[1.5677112e+07, 1.0000000e+00, 2.7750000e+01, 5.8500000e-01,
        1.0000000e+00, 1.3000000e+01, 4.0000000e+00, 2.5000000e-01,
        1.0000000e+00, 1.0000000e+00, 2.0000000e+00, 0.0000000e+00,
        2.0000000e+00, 2.6000000e+02, 5.0100000e+02],
       [1.5597536e+07, 1.0000000e+00, 2.7670000e+01, 2.0000000e+00,
        2.0000000e+00, 1.4000000e+01, 8.0000000e+00, 1.0000000e+00,
        1.0000000e+00, 1.0000000e+00, 4.0000000e+00, 0.0000000e+00,
        2.0000000e+00, 1.4000000e+02, 7.5450000e+03],
       [1.5571284e+07, 1.0000000e+00, 3.2830000e+01, 2.5000000e+00,
        2.0000000e+00, 1.3000000e+01, 8.0000000e+00, 2.7500000e+00,
        1.0000000e+00, 1.0000000e+00, 6.0000000e+00, 0.0000000e+00,
```

```
 2.0000000e+00, 1.6000000e+02, 2.0730000e+03],
[1.5570947e+07, 1.0000000e+00, 2.7250000e+01, 1.6650000e+00,
 2.0000000e+00, 1.3000000e+01, 8.0000000e+00, 5.0850000e+00,
 1.0000000e+00, 1.0000000e+00, 9.0000000e+00, 0.0000000e+00,
 2.0000000e+00, 3.9900000e+02, 8.2800000e+02],
[1.5641715e+07, 1.0000000e+00, 1.9420000e+01, 6.5000000e+00,
 2.0000000e+00, 9.0000000e+00, 8.0000000e+00, 1.4600000e+00,
 1.0000000e+00, 1.0000000e+00, 7.0000000e+00, 0.0000000e+00,
 2.0000000e+00, 8.0000000e+01, 2.9550000e+03],
[1.5706602e+07, 1.0000000e+00, 2.9670000e+01, 1.4150000e+00,
 2.0000000e+00, 9.0000000e+00, 8.0000000e+00, 7.5000000e-01,
 1.0000000e+00, 1.0000000e+00, 1.0000000e+00, 0.0000000e+00,
 2.0000000e+00, 2.4000000e+02, 1.0100000e+02],
[1.5768449e+07, 1.0000000e+00, 4.1920000e+01, 4.2000000e-01,
 2.0000000e+00, 8.0000000e+00, 8.0000000e+00, 2.1000000e-01,
 1.0000000e+00, 1.0000000e+00, 6.0000000e+00, 0.0000000e+00,
 2.0000000e+00, 2.2000000e+02, 9.4900000e+02],
[1.5608595e+07, 0.0000000e+00, 3.0420000e+01, 1.3750000e+00,
 2.0000000e+00, 9.0000000e+00, 8.0000000e+00, 4.0000000e-02,
 0.0000000e+00, 1.0000000e+00, 3.0000000e+00, 0.0000000e+00,
 2.0000000e+00, 0.0000000e+00, 3.4000000e+01],
[1.5635598e+07, 0.0000000e+00, 2.4580000e+01, 6.7000000e-01,
 2.0000000e+00, 6.0000000e+00, 8.0000000e+00, 1.7500000e+00,
 1.0000000e+00, 0.0000000e+00, 0.0000000e+00, 0.0000000e+00,
 2.0000000e+00, 4.0000000e+02, 1.0000000e+00],
[1.5647191e+07, 0.0000000e+00, 3.2330000e+01, 5.4000000e-01,
 2.0000000e+00, 1.3000000e+01, 4.0000000e+00, 4.0000000e-02,
 1.0000000e+00, 0.0000000e+00, 0.0000000e+00, 0.0000000e+00,
 2.0000000e+00, 4.4000000e+02, 1.1178000e+04]])
```

In [ ]:

RESULT:

Hence, we successfully implemented self-organizing maps on given dataset and extracted fraud applications.