In [1]:

```
import numpy as np
```

In [ ]:

```
data=[]
res=0
for i in range(0,50):

  res=i

  #print("")
  re=np.binary_repr(res, width=10)
  temp=[]
  for k in re:
      temp.append(int(k))

  data.append(temp)

data=np.reshape(data,(50,10))
print(data)
```

In [ ]:

```
X=data
print(X)
```

In [ ]:

```
y=[]
for i in range(0,8):
    y.append(1)
    #print(i)
for i in range(8,16):
    y.append(2)
for i in  range(16,24):
    y.append(3)
for i in range(24,32):
    y.append(5)
for i in range(32,50):
    y.append(4)
y=np.array(y)
y=y.reshape(50,1)
print(y)
```

In [12]:

```
w=np.random.random((len(X[0]),5))
#w=[]
#w.append([0.2,0.8])
#w.append([0.6,0.4])
#w.append([0.5,0.7])
#w.append([0.9,0.3])
#w=np.array(w)
print(w)
```

```
[[ 0.66315975  0.99516238  0.88824545  0.93582276  0.67794595]
 [ 0.41702515  0.30295343  0.2925622   0.87029285  0.28461974]
 [ 0.62258151  0.95889008  0.3953286   0.23857381  0.93692012]
 [ 0.42415189  0.592749    0.83028495  0.66771101  0.80193451]
 [ 0.44033361  0.65828662  0.51972942  0.79038667  0.19605325]
 [ 0.39070722  0.35783064  0.39657239  0.28051271  0.61308512]
 [ 0.72421657  0.76820599  0.46855211  0.58134371  0.21065411]
 [ 0.22898098  0.6400071   0.11120108  0.46956285  0.36870611]
 [ 0.62718309  0.88558151  0.97930642  0.84952515  0.93984287]
 [ 0.09377905  0.61484148  0.77507     0.01597251  0.81249764]]
```

In [ ]:

```
print(w[:,0])
print(X[0])
alpha=0.6
radius=10.0
```

In [ ]:

```
epoch=10000
distance=[]


def dist(x):
    distance2=[]
    for l in range(0,5):
        s=0
        W=w[:,l]
        for k in range(0,10):
            #print(x[k])
            #print(W[k])
            d=np.power((x[k]-W[k]),2)
            #print(d)
            s=s+d
        #print(s)
        distance.append(s)
        distance2.append(s)

    return distance2


for i in range(0,epoch):
    print("Learning Rate=",alpha)
    for j in range(0,50):
        x=X[j]
        element=(dist(x))
        m=99999999999
        print(element)
        for j in range(0,5):
            if element[j]<m:
                m=element[j]
                key=j
        uw=w[:,key]
        for p in range(0,len(uw)):
            w[p][key]=w[p][key]+alpha*(x[p]-w[p][key])


        #print(w)

    alpha=alpha/2
    radius=radius/2




#print(distance)
print(w)
```

In [ ]:

```
epoch=10000
distance=[]


def dist(x):
    distance2=[]
    for l in range(0,5):
        s=0
        W=w[:,l]
        for k in range(0,10):
            #print(x[k])
            #print(W[k])
            d=np.power((x[k]-W[k]),2)
            #print(d)
```

```
            #print(d)
            s=s+d
        #print(s)
        distance.append(s)
        distance2.append(s)

    return distance2


for i in range(0,epoch):
    print("Learning Rate=",alpha)
    for j in range(0,50):
        x=X[j]
        element=(dist(x))
        m=99999999999
        print(element)
        for j in range(0,5):
            if element[j]<m:
                m=element[j]
                key=j
        uw=w[:,key]
        for p in range(0,len(uw)):
            w[p][key]=w[p][key]+alpha*(x[p]-w[p][key])


        #print(w)

    alpha=alpha/2
    radius=radius/2




#print(distance)
print(w)
```

In [17]:

```
w=ww
cluster=[]
for j in range(0,50):
        x=X[j]
        m=99999999999
        element=dist(x)
        #print(element)
        for k in range(0,5):
            if element[k]<m:
                m=element[k]
                key=k
        cluster.append(key+1)
print(cluster)
```

```
[3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 5, 3,
3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 3, 3]
```

In [ ]:

```
cluster=np.array(cluster)
cluster=cluster.reshape(50,1)
output=np.concatenate((X, cluster), axis=1)
#print(output)
```

In [31]:

```
cnt=0
for i in range(0,50):
        if(cluster[i]==y[i]):
            cnt=cnt+1

accuracy=(cnt/50)*100
```

```
print("Training accuracy=", accuracy,'%')
```

```
Training accuracy= 64.0 %
```

In [ ]:

```
```