In [25]:
```
import numpy as np
import pandas as pd

def getData(filePath):
    data = np.genfromtxt(filePath, delimiter=',')
    x, y = np.array(data[:,0:-1], dtype=float), np.array(data[:,-1],dtype=int)
    y = y.reshape(1,len(y)).T
    return x,y

def sigmoid(x):
    return 1/(1+np.exp(-x))

def trainNeuralNet(synapse0, synapse1, epochs):
    for j in range(epochs):
        l1 = sigmoid(np.dot(X,synapse0))
        l2 = sigmoid(np.dot(l1,synapse1))
        l2_delta = (y - l2)*(l2*(1-l2))
        l1_delta = l2_delta.dot(synapse1.T) * (l1 * (1-l1))
        synapse1 += l1.T.dot(l2_delta) #adjust our synapses up or down as necessary
        synapse0 += X.T.dot(l1_delta)
```

Test our neural network trainer against a simple dataset: X will contain binary tuples and Y will be the XOR result of rows in X.

In [27]:
```
# trivial dataset
X, y = getData('data/prepared/trivial.csv')

# X = np.array([ [0,0],[0,1],[1,0],[1,1] ])
# y = np.array([[0,1,1,0]]).T # XOR(X)

np.random.seed(seed=42)
syn0 = 2*np.random.random((X.shape[1],X.shape[0])) - 1
syn1 = 2*np.random.random((y.shape[0],y.shape[1])) - 1
epochs = 1000

trainNeuralNet(syn0, syn1, epochs)

layer1_transform = sigmoid(np.dot(X,syn0))
result = sigmoid(np.dot(layer1_transform,syn1))

print("Output of predicted y (2nd and 3rd rows should be 1): ",result)
```

```
[[ 0.  0.]
 [ 0.  1.]
 [ 1.  0.]
 [ 1.  1.]]
[[0]
 [1]
 [1]
 [0]]
Output of predicted y (2nd and 3rd rows should be 1):  [[ 0.09335549]
 [ 0.95700552]
 [ 0.91362687]
 [ 0.0701501 ]]
```

Looks good. Now lets load our accute inflamation dataset.

In [84]:
```
X,y = getData('data/prepared/data.csv')

np.random.seed(seed=42)
syn0 = 2*np.random.random((X.shape[1],X.shape[0])) - 1
syn1 = 2*np.random.random((y.shape[0],y.shape[1])) - 1
epochs = 1000

trainNeuralNet(syn0, syn1, epochs)

layer1_transform = sigmoid(np.dot(X,syn0))
result = sigmoid(np.dot(layer1_transform,syn1))

# print("y: ",y)
result = np.double(result>0.99999999)
print("incorrect guesses: ",np.sum(np.abs(y-result)))
print("accuracy: ",1-np.sum(y-result)/X.shape[0])
```

```
incorrect guesses:  10.0
accuracy:  0.916666666667
```

In [ ]:
```

```