```
import numpy
import random
```

```
#create hidden neurons and their weights
hidden = [[random.randint(0,1)] for _ in range(3)]
for h in hidden:
    h.extend([random.random() for _ in range(2)])
for h in hidden:
    if h[0] == 0:
        h[0] = -1
#h[0] = state of the neuron
#h[1] = weight to first visible neuron
#h[2] = weight tp the second visible neuron
print 'current network:'
for neuron in hidden:
    print neuron


#training data
ts = [(0, 1), (1, 0)]


#energy function
def E(training):
    e = 0
    for h in hidden:
        e = e+(h[0]*h[1]*training[0]
              +h[0]*h[2]*training[1])
    return e

#Temperature value
T = 1

#Chose random neuron to be switched
n = random.randint(0, len(hidden)-1)
#compute current energy
e1 = E(ts[0])

#switch value
hidden[n][0]=hidden[n][0]*-1

#compute new energy value
e2 = E(ts[0])

#switch it back to original state
hidden[n][0]=hidden[n][0]*-1

#calculate propability to switch neuron
p= 1/(1+exp(-1*(abs(e2-e1)/T)))
print 'We switch neuron ',n, ' with a propability of', p, ':'
print ''
```

```
#draw a random value, and maybe flip value
q = random.random()
print 'we rolled a ', q
if q <= p:
    hidden[n][0] = hidden[n][0]*-1

print 'New network:'
for neuron in hidden:
    print neuron
```

```
current network:
[-1, 0.4340291193610335, 0.28557659015505477]
[1, 0.12045096013195544, 0.7401874263537412]
[1, 0.20580381249812996, 0.12691749623449566]
We switch neuron  0  with a propability of 0.639029222885 :

we rolled a  0.449830999313
New network:
[1, 0.4340291193610335, 0.28557659015505477]
[1, 0.12045096013195544, 0.7401874263537412]
[1, 0.20580381249812996, 0.12691749623449566]
```