

simplest-ever-multicause-RBM

June 26, 2015

```
In [14]: %matplotlib inline
import numpy as np
import numpy.random as rng
from pylab import *

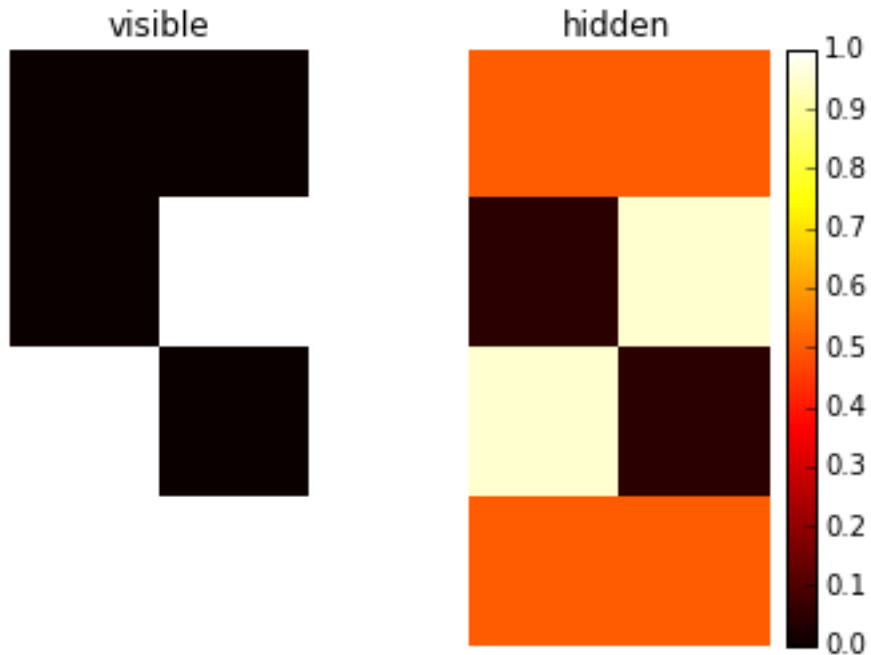
def sigmoid(x):
    return 1.0/(1+np.exp(-x))
```

Two inputs. two rbms, both the same, having two units each!

I will set weights such that patterns 01 and 10 on the visible units get piles of sand (are “memories”), and 00 and 11 aren’t.

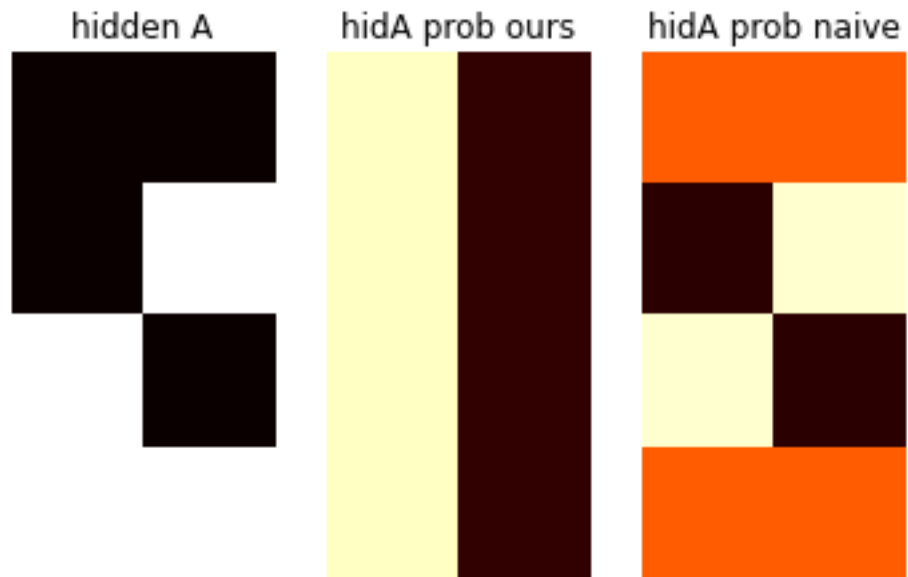
```
In [103]: w = 3.0 * np.array([[1,-1],[-1,1]])
pats = np.array([[0,0],[0,1],[1,0],[1,1]])
```

```
In [104]: psiA = np.dot(pats, w)
subplot(121)
imshow(pats, interpolation='nearest', cmap='hot', vmin=0, vmax=1)
title('visible')
ax = axis('off')
subplot(122)
imshow(sigmoid(psiA), interpolation='nearest', cmap='hot', vmin=0, vmax=1)
title('hidden')
colorbar()
ax = axis('off')
```



```
In [131]: v = np.array([1,1])
          hB = np.array([0,1])
          phiA = np.dot(pats, w) - np.dot((2*pats-1),w/2)
          phiB = np.dot(hB, w)
          sigA = sigmoid(phiA)
          sigAB = sigmoid(phiA + phiB)
          effective_vis = v + sigA - sigAB
          our_psiA = np.dot(effective_vis, w)

In [140]: subplot(131)
          imshow(pats, interpolation='nearest',cmap='hot', vmin=0, vmax=1)
          title('hidden A')
          ax = axis('off')
          subplot(132)
          imshow(sigmoid(our_psiA), interpolation='nearest',cmap='hot', vmin=0, vmax=1)
          title('hidA prob ours')
          ax = axis('off')
          subplot(133)
          naive_psiA = np.dot(v, w)
          imshow(sigmoid(psiA), interpolation='nearest',cmap='hot', vmin=0, vmax=1)
          title('hidA prob naive')
          #colorbar()
          ax = axis('off')
          savefig('the_way')
```



In []: