

hopfield

April 15, 2019

1 Hopfield Neural Network

A Hopfield network is a form of recurrent artificial neural network popularized by John Hopfield in 1982, but described earlier by Little in 1974. Hopfield nets serve as content-addressable (“associative”) memory systems with binary threshold nodes. They are guaranteed to converge to a local minimum and, therefore, may converge to a false pattern (wrong local minimum) rather than the stored pattern (expected local minimum). Hopfield networks also provide a model for understanding human memory.

```
In [1]: # importing modules
```

```
import numpy as np
from neupy import algorithms
from neupy import plots
import matplotlib.pyplot as plt
```

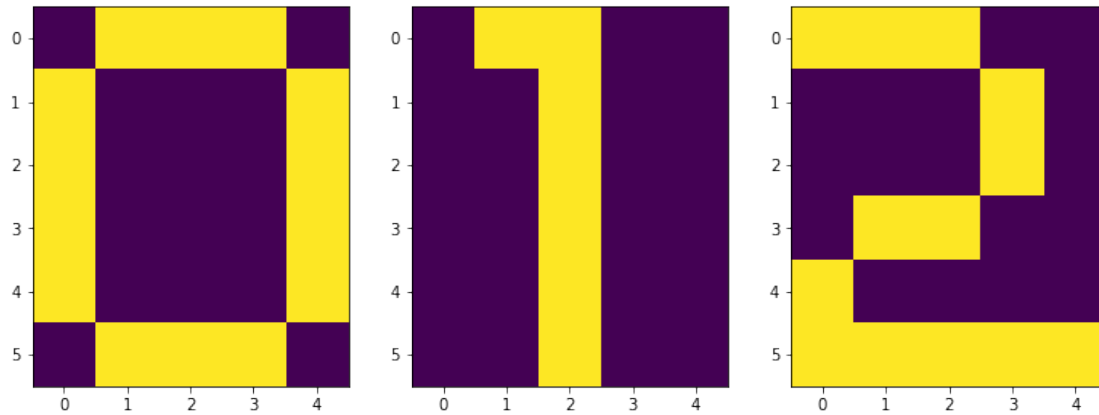
```
In [2]: # Training Data
```

```
zero = np.matrix([ 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0,
one = np.matrix([0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
two = np.matrix([1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0,
```

```
data = np.concatenate([zero, one, two], axis=0)
```

```
plt.figure(figsize=[12,12])
plt.subplot(131)
plt.imshow(zero.reshape(6,5))
plt.subplot(132)
plt.imshow(one.reshape(6,5))
plt.subplot(133)
plt.imshow(two.reshape(6,5))
```

```
Out[2]: <matplotlib.image.AxesImage at 0x7fe8fafc0278>
```



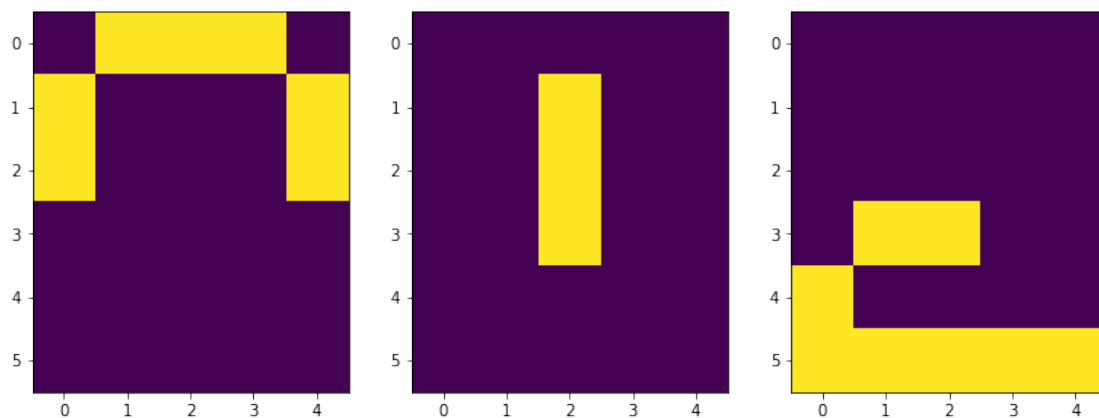
In [3]: *# model training*

```
hop = algorithms.DiscreteHopfieldNetwork(mode='sync')
hop.train(data)
```

In [4]: *# Testing Data*

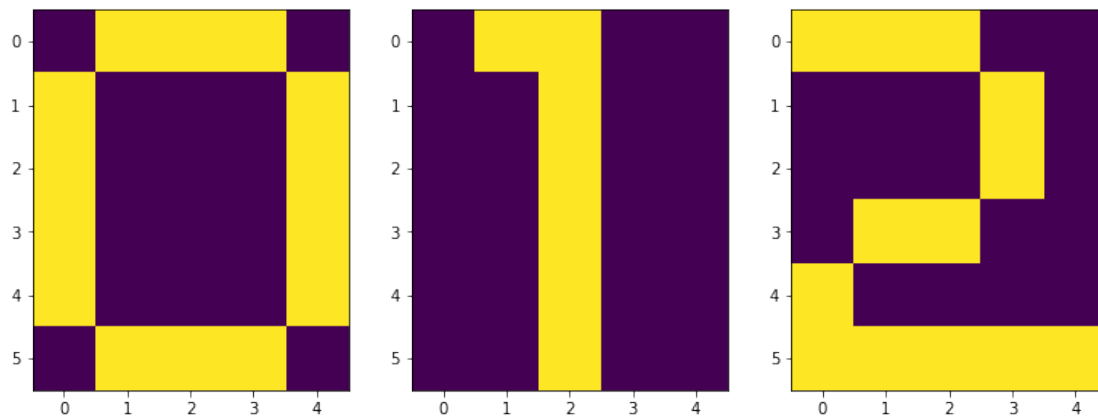
```
half_zero = np.matrix([0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0])
half_two = np.matrix([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0])
half_one = np.matrix([0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0])
plt.figure(figsize=[12,12])
plt.subplot(131)
plt.imshow(half_zero.reshape(6,5))
plt.subplot(132)
plt.imshow(half_one.reshape(6,5))
plt.subplot(133)
plt.imshow(half_two.reshape(6,5))
```

Out[4]: <matplotlib.image.AxesImage at 0x7fe8faf02198>



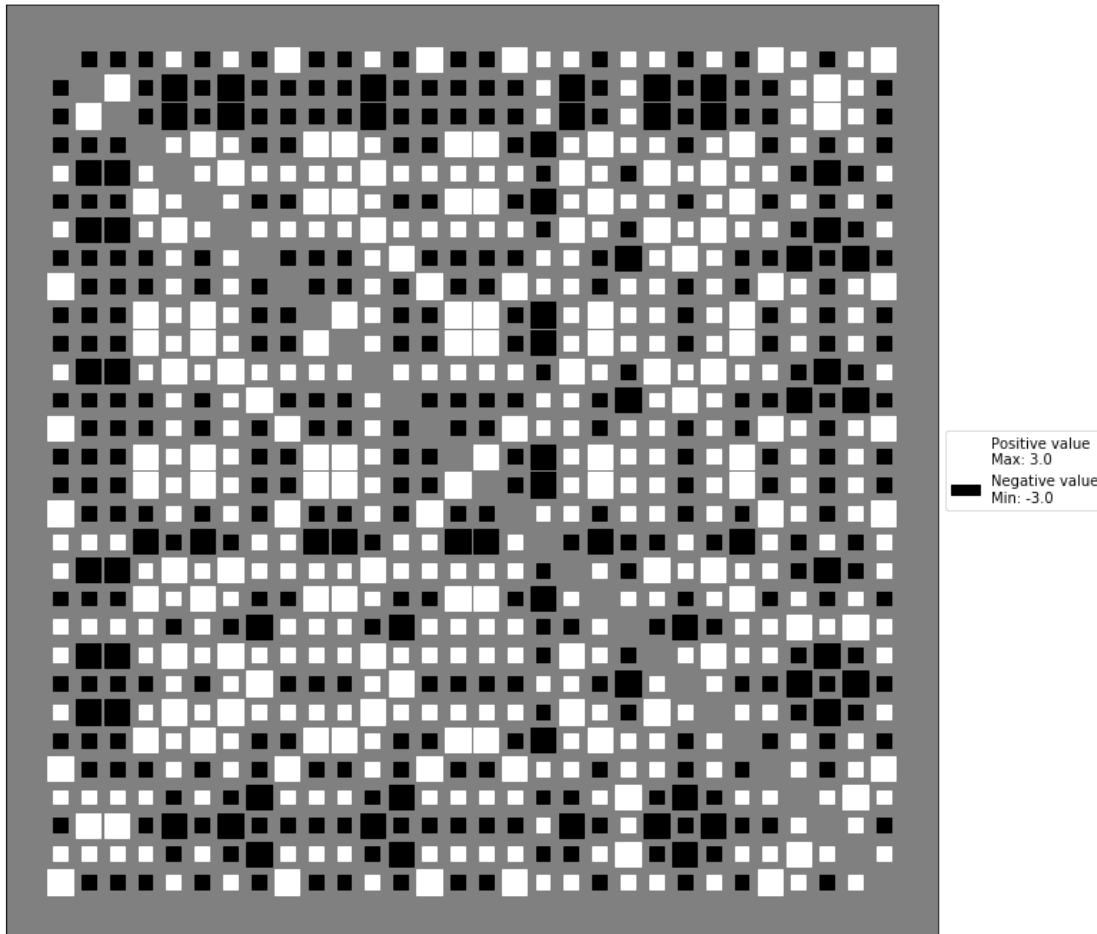
```
In [5]: # pattern reconstruction
recon_zero = hop.predict(half_zero)
recon_one = hop.predict(half_one)
recon_two = hop.predict(half_two)
plt.figure(figsize=[12,12])
plt.subplot(131)
plt.imshow(recon_zero.reshape(6,5))
plt.subplot(132)
plt.imshow(recon_one.reshape(6,5))
plt.subplot(133)
plt.imshow(recon_two.reshape(6,5))
```

Out[5]: <matplotlib.image.AxesImage at 0x7fe8fae73be0>



```
In [6]: plt.figure(figsize=(14, 12))
plots.hinton(hop.weight)
```

Out[6]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe8fadc2208>



1.1 References

1. http://neupy.com/2015/09/20/discrete_hopfield_network.html#discrete-hopfield-network
2. https://en.wikipedia.org/wiki/Hopfield_network