

COMP9444 Neural Networks and Deep Learning

Term 3, 2019

Solutions to Exercise 7: Hopfield Networks

This page was last updated: 11/18/2019 17:39:08

1.

- a. Compute the weight matrix for a Hopfield network with the two memory vectors $[1, -1, 1, -1, 1, 1]$ and $[1, 1, 1, -1, -1, -1]$ stored in it.

The outer product W_1 of $[1, -1, 1, -1, 1, 1]$ with itself is

$$\begin{bmatrix} 0 & -1 & 1 & -1 & 1 & 1 \\ -1 & 0 & -1 & 1 & -1 & -1 \\ 1 & -1 & 0 & -1 & 1 & 1 \\ -1 & 1 & -1 & 0 & -1 & -1 \\ 1 & -1 & 1 & -1 & 0 & 1 \\ 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}$$

The outer product W_2 of $[1, 1, 1, -1, -1, -1]$ with itself is

$$\begin{bmatrix} 0 & 1 & 1 & -1 & -1 & -1 \\ 1 & 0 & 1 & -1 & -1 & -1 \\ 1 & 1 & 0 & -1 & -1 & -1 \\ -1 & -1 & -1 & 0 & 1 & 1 \\ -1 & -1 & -1 & 1 & 0 & 1 \\ -1 & -1 & -1 & 1 & 1 & 0 \end{bmatrix}$$

The weight matrix W is $(1/6) \times (W_1 + W_2) = (1/3) \times$

$$\begin{bmatrix} 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

- b. Confirm that both these vectors are stable states of this network.

$$\begin{aligned} \text{sgn}(W \cdot [1, -1, 1, -1, 1, 1]) &= \text{sgn}((2/3) \times [1, -1, 1, -1, 1, 1]) \\ &= [1, -1, 1, -1, 1, 1] \end{aligned}$$

so this one is stable. Similarly,

$$\begin{aligned}\text{sgn}(W.[1, 1, 1, -1, -1]) &= \text{sgn}((2/3) \times [1, 1, 1, -1, -1]) \\ &= [1, 1, 1, -1, -1]\end{aligned}$$

so this one is stable too.

2. Consider the following weight matrix W:

$$\begin{array}{ccccc}0.0 & -0.2 & 0.2 & -0.2 & -0.2 \\ -0.2 & 0.0 & -0.2 & 0.2 & 0.2 \\ 0.2 & -0.2 & 0.0 & -0.2 & -0.2 \\ -0.2 & 0.2 & -0.2 & 0.0 & 0.2 \\ -0.2 & 0.2 & -0.2 & 0.2 & 0.0\end{array}$$

a. Starting in the state $[1, 1, 1, 1, -1]$, compute the state flow to the stable state using asynchronous updates.

$W.[1, 1, 1, 1, -1] = [0, -0.4, 0, -0.4, 0]$. Hence:

If neuron 1, 3, or 5 updates first, its total net input is 0, so it does not change state;

If neuron 2 updates first, its total net input is -0.4 , and its current value is $+1$, so it changes state to -1 , and the new state is $[1, -1, 1, 1, -1]$. Call this Case A.

If neuron 4 updates first, its total net input is -0.4 , and its current value is one, so it changes state to -1 , and the new state is $[1, 1, 1, -1, -1]$. Call this Case B.

Case A: $W.[1, -1, 1, 1, -1] = [0.4, -0.4, 0.4, -0.8, -0.4]$. Hence:

If neurons 1, 2, 3, or 5 update first, there is no state change.

If neuron 4 updates first, it flips, and the new state is $[1, -1, 1, -1, -1]$.

$W.[1, -1, 1, -1, -1] = [0.8, -0.8, 0.8, -0.8, -0.8]$. So no matter which neuron updates, there is no change. This is a stable state.

Case B: $W.[1, 1, 1, -1, -1] = [0.4, -0.8, 0.4, -0.4, -0.4]$. Hence:

If neurons 1, 3, 4 or 5 update first, there is no state change.

If neuron 2 updates first, it flips, and the new state is $[1, -1, 1, -1, -1]$.

This is the same state as that reached in case A, and as seen in case A, it is a stable state.

b. Starting in the (same) state $[1, 1, 1, 1, -1]$, compute the next state using synchronous updates.

$W.[1, 1, 1, 1, -1] = [0, -0.4, 0, -0.4, 0]$, so neurons 2 and 4 flip, resulting in a state of $[1, -1, 1, -1, -1]$. (We know from the previous part that this is a stable state.)
