Introduction to Neural Networks

Johns Hopkins University
Engineering for Professionals Program
605-447/625-438
Dr. Mark Fleischer

Module 8.4: Hopfield Convergence

What We've Covered So Far…

- Learned about the Hopfield Network
  - Hebbian Learning
    - Recurrent neural networks
    - Matrix/vector representation of a recurrent network
    - Heat death
  - Excitation and Inhibition in Hopfield Networks via bi-polar values
  - Possibility of cycling in Hopfield networks in part because of inhibition and excitation.
- In this sub-module
  - Another modality to the dynamical system.
  - Prove convergence of the Hopfield network using this modality.

A Peek at the Hopfield Convergence Theorem

- In a Hopfield network, with asynchronous updating, the Hopfield net will always converge.
- This is a **sufficient condition**. It is not necessary—there may be other ways to converge even without the Hopfield conditions (no self connections w_{ii} = 0 for all i, symmetric connections $w_{ji} = w_{ij}$).

Convergence

- Want the simplest approach.
- If we simply go down hill, that's easy …
- If there exists some lower-bound to some value --- a bottom of a hill, then we'll eventually reach it.
- In other words, if every possible state has a ranking or metric associated with it (we'll call it energy), then showing that we always reduce that metric is equivalent to showing that we will never get into a cycle.

So What is Asynchronous Updating?

$$\mathbf{x}_{k+1} = F_h\left(\mathbf{W}\mathbf{x}_k\right)$$

$$F_h\left(\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & & \ddots & \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}\right) = \begin{bmatrix} x_1^* \\ x_2^* \\ \vdots \\ x_n^* \end{bmatrix}$$

$$F_h\left(\begin{bmatrix} \boxed{w_{11} \quad w_{12} \quad \cdots \quad w_{1n}} \\ w_{21} \quad w_{22} \quad \cdots \quad w_{2n} \\ \vdots \qquad\qquad \ddots \qquad\quad \\ w_{n1} \quad w_{n2} \quad \cdots \quad w_{nn} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}\right) = \begin{bmatrix} x_1^* \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

So What is Asynchronous Updating?

$$\mathbf{x}_{k+1} = F_h\left(\mathbf{W}\mathbf{x}_k\right)$$

$$F_h\left(\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & & \ddots & \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}\right) = \begin{bmatrix} x_1^* \\ x_2^* \\ \vdots \\ x_n^* \end{bmatrix}$$

$$F_h\left(\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & & \ddots & \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix}\begin{bmatrix} x_1^* \\ x_2^* \\ \vdots \\ x_n \end{bmatrix}\right) = \begin{bmatrix} x_1^* \\ x_2^* \\ \vdots \\ x_n^* \end{bmatrix}$$

Define the Hecht-Nielsen Function:

$$H(\mathbf{x}) = -\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}x_i x_j + \sum_{i=1}^{N}\theta_i x_i$$

Want to prove that

$$\Delta H(\mathbf{x}) \le 0$$

Define an Update Rule

So, apply the update rule to a given node *k* (the one being updated---remember, asynchronously). *x** refers to the current iteration. Now,

$$x_i^* = \begin{cases} x_k^* & i = k \\ x_i & i \neq k \end{cases}$$

where *k* refers to the one we're updating.

The Update Expression

$$\Delta H = H^* - H = -\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}\left(x_i^* x_j^* - x_i x_j\right)$$

Current

Previous Iteration

Remember, the * refers to an updated value.

The Update Expression

$$= \underbrace{-\sum_{i=1}^{N}\sum_{\substack{j\neq k\\ j=1}}^{N} w_{ij}\left(x_i^* x_j^* - x_i x_j\right)}_{A} \underbrace{-\sum_{i=1}^{N} w_{ik}\left(x_i^* x_k^* - x_i x_k\right)}_{B}$$

Simplifying Part *B* first, we get:

$$B = -\sum_{\substack{i=1\\ i\neq k}}^{N} w_{ik}\left(x_i x_k^* - x_i x_k\right) = -\sum_{\substack{i=1\\ i\neq k}}^{N} w_{ik}\left(x_k^* - x_k\right)x_i$$

Why?

The Update Expression

Remember, $$x_i^* = \begin{cases} x_k^* & i = k \\ x_i & i \neq k \end{cases}$$

$$B = -\sum_{\substack{i=1 \\ i\neq k}}^{N} w_{ik}\left(x_i x_k^* - x_i x_k\right) = -\sum_{\substack{i=1 \\ i\neq k}}^{N} w_{ik}\left(x_k^* - x_k\right)x_i$$

First, $w_{kk} = 0$ (no self-connections) and $x_i^* = x_i$ for $i \neq k$.
Which reduces to:

$$= -\sum_{\substack{i=1 \\ i\neq k}}^{N} w_{ik} x_i \Delta x_k$$ for Part B.

The Update Expression

$$= \underbrace{-\cancel{\sum_{i=1}^{N}}\sum_{\substack{j\neq k\\ j=1}}^{N} w_{ij}\left(x_i^* x_j^* - x_i x_j\right)}_{A} - \underbrace{\sum_{i=1}^{N} w_{ik}\left(x_i^* x_k^* - x_i x_k\right)}_{B}$$

Now Part A, recall that for *i or j* $\neq$ *k*, $x_i^* = x_i$ so the sum

$$\left(x_i^* x_j^* - x_i x_j\right) = 0$$

so $i = k$ is the only term left, so

$$A = -\sum_{\substack{j=1\\ j\neq k}}^{N} w_{kj}\left(x_k^* x_j - x_k x_j\right) = -\sum_{\substack{j=1\\ j\neq k}}^{N} w_{kj} x_j \Delta x_k$$

Let's Sum the deltas…

$$\Delta H = A + B = -\sum_{\substack{j=1 \\ j\neq k}}^{N} w_{kj} x_j \Delta x_k - \sum_{\substack{i=1 \\ i\neq k}}^{N} w_{ik} x_i \Delta x_k$$

$$= -\sum_{\substack{j=1 \\ j\neq k}}^{N} w_{kj} x_j \Delta x_k - \sum_{\substack{j=1 \\ j\neq k}}^{N} w_{jk} x_j \Delta x_k$$

$$= -\sum_{\substack{j=1 \\ j\neq k}}^{N} \left(w_{kj} + w_{jk}\right) x_j \Delta x_k = -2 \underbrace{\sum_{\substack{j=1 \\ j\neq k}}^{N} w_{kj} x_j}_{Activity\ a_k} \Delta x_k$$

Implications for cases

$$\Delta H = -2 \underbrace{\sum_{\substack{j=1 \\ j \neq k}}^{N} w_{kj} x_j}_{Activity\ a_k} \Delta x_k$$

Suppose

$$a_k > 0, \quad \text{then} \quad x_k^* = f_h(a_k) = 1 > 0$$

then

$$\Delta x_k = x_k^* - x_k \geq 0.$$

Implications for cases

$$\Delta H = -2 \underbrace{\sum_{\substack{j=1 \\ j \neq k}}^{N} w_{kj} x_j}_{Activity\ a_k} \Delta x_k$$

Activity $a_k > 0$

$$\Downarrow$$

$$\Delta x_k = x_k^* - x_k \geq 0.$$

Why? Well, if $x^* = 1$, then x must have been either 1 (unchanged) or -1 (changed) hence 1-1=0 or 1 - -1 = 2 >0 . Thus, $-2a_k\Delta x_k \leq 0$.

Implications for cases

$$\Delta H = -2 \underbrace{\sum_{\substack{j=1 \\ j \neq k}}^{N} w_{kj} x_j}_{Activity\ a_k} \Delta x_k$$

Suppose $a_k < 0, \quad \text{then} \quad x_k^* = f_h(a_k) = -1 < 0$

Activity $a_k < 0$

$$\Downarrow$$

$$\Delta x_k = x_k^* - x_k \leq 0.$$

JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Implications for cases

$$\Delta H = -2 \underbrace{\sum_{\substack{j=1 \\ j \neq k}}^{N} w_{kj} x_j}_{Activity\ a_k} \Delta x_k$$

Activity $a_k < 0$

$\Downarrow$

$$\Delta x_k = x_k^* - x_k \leq 0.$$

Why? Well, if x^* = -1, then x must have been either -1 (unchanged) or 1 (changed) hence -1- -1=0 or -1 - 1 = -2 < 0 . So again, $-2a_k\Delta x_k \leq 0$.

H-N Function Never Increases

- This means, the *H* value can only decrease and since there is a lower bound, it will eventually converge such that

$$f_h\left(\sum_j w_{ij} x_j\right) = \mathbf{x}$$

Cannot oscillate between solutions. Why?

Now $$\Delta H = -2a_k \Delta x_k = \begin{cases} 0 \\ -4|a_k| \end{cases}$$

For $a_k > 0$:

If $\Delta x_k = 0$, then no change in state, hence no oscillation.

If $\Delta x_k \neq 0$ then $\Delta x_k = 2$ and $\Delta H < 0$ as we showed above. Can't increase *H.*

For $a_k \leq 0$:

In this case, then $\Delta H = 0$ if $a_k = 0$ or $\Delta x_k = 0$.
The latter means no change. If $a_k = 0$ then $\Delta H = 0$ and
Again no change in *H.*