CS623: Introduction to Computing with Neural Nets *(lecture-13)*

Pushpak Bhattacharyya
Computer Science and Engineering Department
IIT Bombay

Hopfield Net: an *o(n²) algorithm*

- Consider the energy expression

$$E = -[\, w_{12} x_1 x_2 + w_{13} x_1 x_3 + \ldots + w_{1n} x_1 x_n$$

$$+ w_{23} x_2 x_3 + w_{24} x_2 x_4 + \ldots + w_{2n} x_2 x_n$$

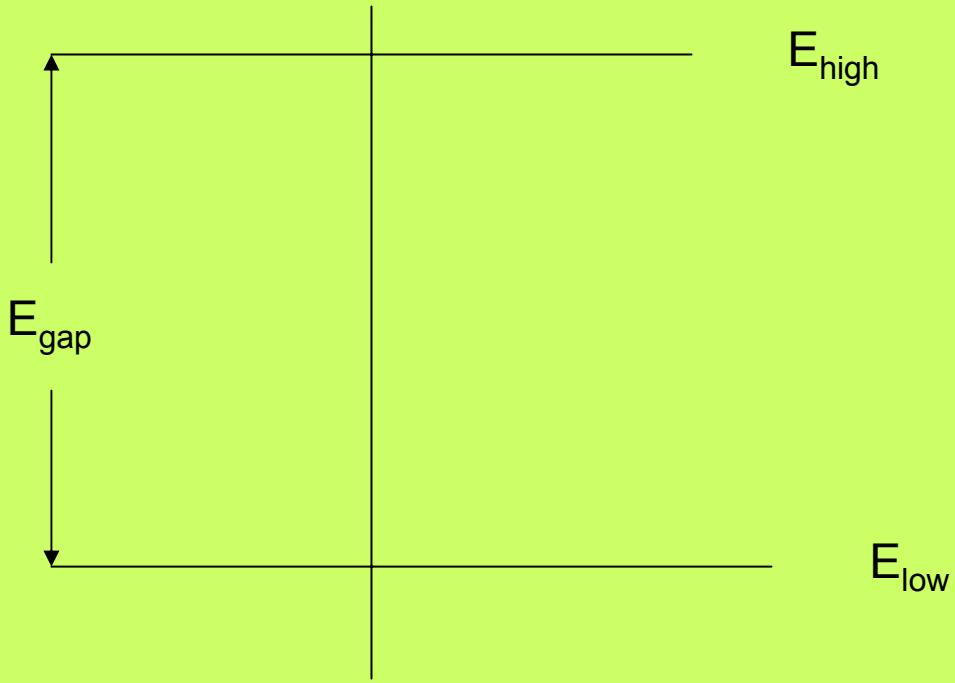
$$\vdots$$

$$+ w_{(n-1)n} x_{(n-1)} x_n \,]$$

- *E* has *[n(n-1)]/2* terms
- Nature of each term
  - w_{ij} is a real number
  - x_i and x_j are each +1 or -1

No. of steps taken to reach stability

- $E_{gap} = E_{high} - E_{low}$



Analysis of the weights and consequent E_{high} and E_{low}

- W_{il} is any weight with upper and lower bounds as W_{max} *and* W_{min} respectively. Suppose

  $w_{min} \leq w_{ij} \leq w_{max}$

  $w_{max} > w_{min}$

- Case 1: $w_{max} > 0$, $w_{min} > 0$

  $E_{high} = (1/2) * w_{max} * n * (n-1)$

  $E_{low} = -(1/2) * w_{max} * n * (n-1)$

Continuation of analysis of E_{high} and E_{low}

- Case 2: $w_{min} < 0$, $w_{max} < 0$

  $E_{high} = (1/2) * w_{min} * n * (n-1)$

  $E_{low} = -(1/2) * w_{min} * n * (n-1)$

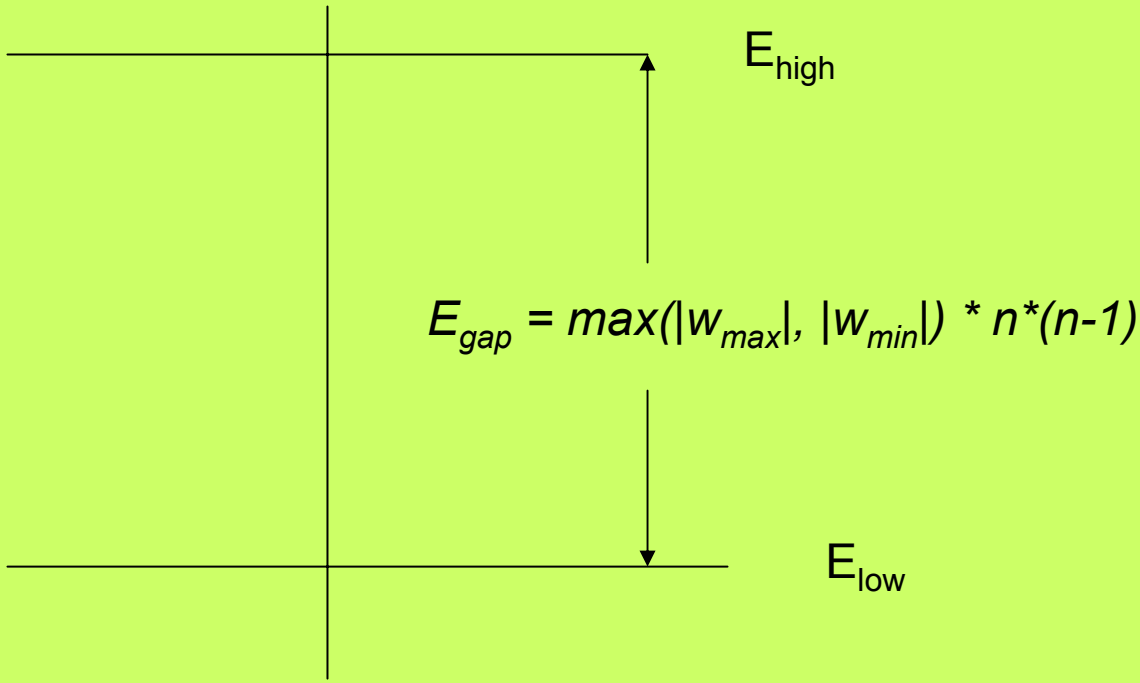
- Case 3: $w_{max} > 0$, $w_{min} < 0$

  $E_{high} = (1/2) * max(|w_{max}|,|w_{min}|) * n*(n-1)$

  $E_{low} = -(1/2) * max(|w_{max}|,|w_{min}|) * n*(n-1)$

The energy gap

- In general,



To find ΔE_{min}

$\Delta E_p = (x_p^{initial} - x_p^{final}) * net_p$

where ΔE_p is the change in energy due to the p^{th} neuron changing activation.

$$| \Delta E_p | = | (x_p^{initial} - x_p^{final}) * net_p |$$
$$= 2 * | net_p |$$

where $net_p = \Sigma^n_{j=1,\, j \neq p} w_{pj} x_j$

To find ΔE_{min}

- $[\Sigma^{n}_{j=1,\, j\neq p} w_{pj}x_j]_{min}$ is determined by the precision of the machine computing it.
- For example, assuming 7 places after the decimal point, the value cannot be lower than *0.0000001 [it can be 0, but that is not of concern to us, since the net will continue in the same state]*
- Thus ΔE_{min} is a constant independent of n, determined by the precision of the machine.

Final observation: $o(n^2)$

- It is possible to reach the minimum independent of *n.*
- Hence in the worst case, the number of steps taken to cover the energy gap is less than or equal to

  $[max(|w_{max}|,|w_{min}|) * n * (n-1)] / constant$
- Thus stability has to be attained in $O(n^2)$ steps

Hopfield Net for Optimization

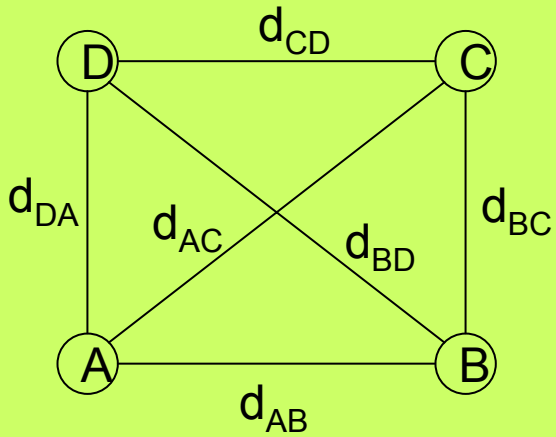
- Optimization problem
  - Maximizes or minimizes a quantity
- Hopfield net used for optimization
  - Hopfield net and Traveling Salesman Problem
  - Hopfield net and Job Scheduling Problem

The essential idea of the correspondence

- In optimization problems, we have to *minimize* a quantity.
- Hopfield net minimizes the energy
- THIS IS THE CORRESPONDENCE

Hopfield net and Traveling Salesman problem

- We consider the problem for *n=4* cities
- In the given figure, nodes represent cities and edges represent the paths between the cities with associated distance.



Traveling Salesman Problem

- Goal
  - Come back to the city A, visiting $j = 2$ to n (n is number of cities) exactly once and minimize the total distance.
- To solve by Hopfield net we need to decide the *architecture*:
  - How many neurons?
  - What are the weights?

Constraints decide the *parameters*

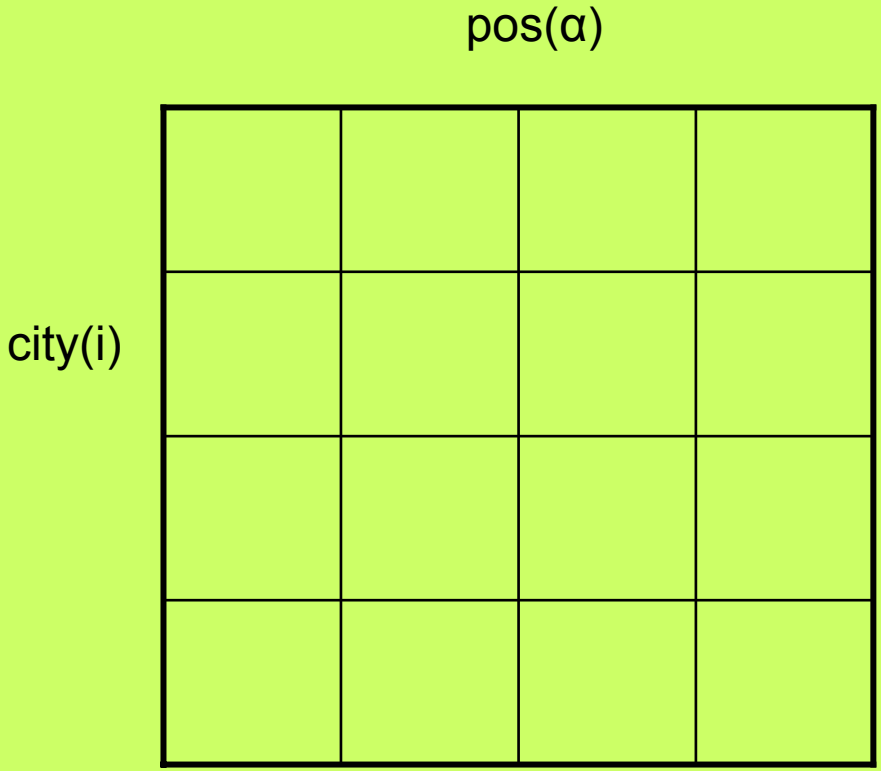
1. For *n* cities and *n* positions, establish city to position correspondence, *i.e*.

   Number of neurons = *n* cities * *n* positions

2. Each position can take <u>one and only one</u> city
3. Each city can be in exactly one position
4. Total distance should be minimum

Architecture

- *n * n* matrix where rows denote cities and columns denote positions
- *cell(i, j) = 1* if and only if i^{th} city is in j^{th} position
- Each cell is a neuron
- n^2 neurons, $O(n^4)$ connections



Expressions corresponding to constraints

1. Each city in one and only one position *i.e*. a row has a single 1.

$$E_1 = \frac{A}{2} \sum_{i=1}^{n} \sum_{\alpha=1}^{n} \sum_{\beta=1, \beta \neq \alpha}^{n} x_{i\alpha} x_{i\beta}$$

- Above equation partially ensures each row has a single 1
- $x_{i\alpha}$ is I/O cost at the *cell(i, α)*

Expressions corresponding to constraints (contd.)

2. Each position has a single city

i.e. each column has at most single 1.

$$E_2 = \frac{B}{2} \sum_{\alpha=1}^{n} \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} x_{i\alpha} \cdot x_{j\alpha}$$

Expressions corresponding to constraints (contd.)

3. All cities MUST be visited once and only once

$$E_3 = \frac{C}{2}\left[\left(\sum_{i=1}^{n}\sum_{\alpha=1}^{n} x_{i\alpha}\right) - n\right]^2$$

Expressions corresponding to constraints (contd.)

- E_1, E_2, E_3 ensure that each row has exactly one 1 and each column has exactly one 1
- If we minimize

  $E_1 + E_2 + E_3$
- Ensures a Hamiltonian circuit on the city graph which is an NP-complete problem.

Constraiant of distance

4. The distance traversed should be minimum

$$E_4 = \frac{D}{2}[\sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} d_{ij} \cdot x_{i\alpha} \cdot (x_{j,(\alpha+1)} + x_{j,(\alpha-1)})]$$

d_{ij} = distance between city *i* and city *j*

Expressions corresponding to constraints (contd.)

- We equate constraint energy:

$$E_{Problem} = E_{network} \qquad (*)$$

Where, $E_{problem} = E_1 + E_2 + E_3 + E_4$

and $E_{network}$ *is the well known energy expression for the Hopfield net*

- Find the weights from (*).