CS623: Introduction to Computing with Neural Nets *(lecture-15)*

Pushpak Bhattacharyya
Computer Science and Engineering Department
IIT Bombay

Finding weights for Hopfield Net applied to TSP

- Alternate and more convenient $E_{problem}$
- $E_P = E_1 + E_2$

where

E_1 is the equation for n cities, each city in one position and each position with one city.

E_2 is the equation for distance

Expressions for E_1 and E_2

$$E_1 = \frac{A}{2}[\sum_{\alpha=1}^{n}(\sum_{i=1}^{n} x_{i\alpha} - 1)^2 + \sum_{i=1}^{n}(\sum_{\alpha=1}^{n} x_{i\alpha} - 1)^2]$$

$$E_2 = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{\alpha=1}^{n} d_{ij} \cdot x_{i\alpha} \cdot (x_{j,\alpha+1} + x_{j,\alpha-1})$$

Explanatory example

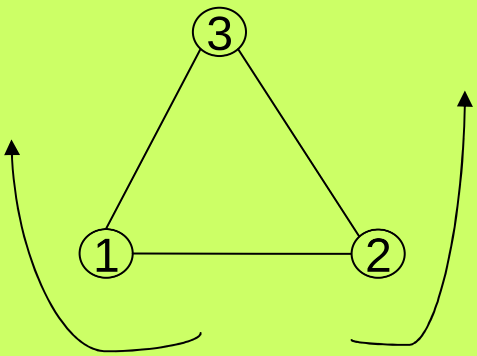

Fig. 1

Fig. 1 shows two possible directions in which tour can take place

pos →

city ↓

	1	2	3
1	x_{11}	x_{12}	X_{13}
2	X_{21}	x_{22}	x_{23}
3	x_{31}	x_{32}	x_{33}

For the matrix alongside, $x_{i\alpha} = 1$, if and only if, i^{th} city is in position α

Expressions of Energy

$$E_{problem} = E_1 + E_2$$

$$\begin{aligned} E_1 = \frac{A}{2}[& (x_{11} + x_{12} + x_{13} - 1)^2 \\ & + (x_{21} + x_{22} + x_{23} - 1)^2 \\ & + (x_{31} + x_{32} + x_{33} - 1)^2 \\ & + (x_{11} + x_{21} + x_{31} - 1)^2 \\ & + (x_{12} + x_{22} + x_{32} - 1)^2 \\ & + (x_{13} + x_{23} + x_{33} - 1)^2] \end{aligned}$$

Expressions (contd.)

$$
\begin{aligned}
E_2 = \frac{1}{2}[\, & d_{12} x_{11}(x_{22} + x_{23}) + \\
& d_{12} x_{12}(x_{23} + x_{21}) + \\
& d_{12} x_{13}(x_{21} + x_{22}) + \\
& d_{13} x_{11}(x_{32} + x_{33}) + \\
& d_{13} x_{12}(x_{33} + x_{31}) + \\
& d_{13} x_{13}(x_{31} + x_{32}) \ldots]
\end{aligned}
$$

$E_{network}$

$$
\begin{aligned}
E_{network} = -[&w_{11,12}x_{11}x_{12} + w_{11,13}x_{11}x_{13} + w_{12,13}x_{12}x_{13} \\
&+ w_{11,21}x_{11}x_{21} + w_{11,22}x_{11}x_{22} + w_{11,23}x_{11}x_{23} \\
&+ w_{11,31}x_{11}x_{31} + w_{11,32}x_{11}x_{32} + w_{11,33}x_{11}x_{33} \ldots]
\end{aligned}
$$

Find row weight

- To find, $w_{11,12}$
  $= -(\text{co-efficient of } x_{11}x_{12}) \text{ in } E_{\text{network}}$
- Search $a_{11}a_{12}$ in $E_{problem}$

$w_{11,12} = -A$ ...from E_1. E_2 cannot contribute

Find column weight

- To find, $w_{11,21}$
  $= -(\text{co-efficient of } x_{11}x_{21}) \text{ in } E_{network}$
- Search co-efficient of $x_{11}x_{21}$ in $E_{problem}$

$w_{11,21} = -A$ ...from E_1. E_2 cannot contribute

Find Cross weights

- To find, $w_{11,22}$

  $= -(\text{co-efficient of } x_{11}x_{22})$
- Search $x_{11}x_{22}$ from $E_{problem.}$ E_1 cannot contribute
- Co-eff. of $x_{11}x_{22}$ in E_2

  $(d_{12} + d_{21}) / 2$

Therefore, $w_{11,22} = -((d_{12} + d_{21}) / 2)$

Find Cross weights

- To find, $w_{11,33}$

  $= -(\text{co-efficient of } x_{11}x_{33})$
- Search for $x_{11}x_{33}$ in $E_{problem}$

  $w_{11,33} = -((d_{13} + d_{31}) / 2)$

Summary

- Row weights = -*A*
- Column weights = -*A*
- Cross weights =

  $-[(d_{ij} + d_{ji})/2], j = i \pm 1$

  $0, j>i+1 \text{ or } j<(i-1)s$
- Threshold = -2A

Interpretation of wts and thresholds

- Row wt. being negative causes the winner neuron to suppress others: one 1 per row.
- Column wt. being negative causes the winner neuron to suppress others: one 1 per column.
- Threshold being -2A makes it possible to for activations to be positive sometimes.
- For non-neighbour row and column ($j>i+1$ or $j<i-1$) neurons, the wt is 0; this is because non-neighbour cities should not influence the activations of corresponding neurons.
- Cross wts when non-zero are proportional to negative of the distance; this ensures discouraging cities with large distances between them to be neighbours.

Can we compare $E_{problem}$ and $E_{network}$?

$$E_1 = \frac{A}{2}[\sum_{\alpha=1}^{n}(\sum_{i=1}^{n} x_{i\alpha} - 1)^2 + \sum_{i=1}^{n}(\sum_{\alpha=1}^{n} x_{i\alpha} - 1)^2]$$

E_1 has square terms $(x_{i\alpha})^2$ which evaluate to 1/0. It also has constants again evaluating to 1.
Sum of square terms and constants

<=n X (1+1+…n times +1) + n X (1+1+…n times +1)

=2n(n+1)

Additionally, there are linear terms of the form *const** $x_{i\alpha}$ which will produce the thresholds of neurons by equating with the linear terms in $E_{network}$.

Can we compare $E_{problem}$ and $E_{network}$? (contd.)

$$E_2 = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{\alpha=1}^{n} d_{ij} \cdot x_{i\alpha} \cdot (x_{j,\alpha+1} + x_{j,\alpha-1})$$

This expressions can contribute only product terms which are equated with the product terms in $E_{network}$

Can we compare $E_{problem}$ and $E_{network}$ (contd)

- So, yes, we CAN compare $E_{problem}$ and $E_{network}$.
- $E_{problem} <= E_{network} + 2n(n+1)$
- When the weight and threshold values are chosen by the described procedure, minimizing $E_{network}$ implies minimizing $E_{problem}$

Principal Component Analysis

Purpose and methodology

- Detect correlations in multivariate data
- Given P variables in the multivariate data, introduce P principal components Z_1, Z_2, Z_3, …Z_P
- Find those components which are responsible for the biggest variation
- Retain them only and thereby reduce the dimensionality of the problem

Example: *IRIS Data (only 3 values out of 150)*

ID	Petal Length (a_1)	Petal Width (a_2)	Sepal Length (a_3)	Sepal Width (a_4)	Classification
001	5.1	3.5	1.4	0.2	Iris-setosa
051	7.0	3.2	4.7	1.4,	Iris-versicolor
101	6.3	3.3	6.0	2.5	Iris-virginica

Training and Testing Data

- Training: 80% of the data; 40 from each class: total 120
- Testing: Remaining 30
- Do we have to consider all the 4 attributes for classification?
- Do we have to have 4 neurons in the input layer?
- Less neurons in the input layer may reduce the overall size of the n/w and thereby reduce training time
- It will also likely increase the generalization performance (Occam Razor Hypothesis: *A simpler hypothesis (i.e., the neural net) generalizes better*

The multivariate data

$$
\begin{array}{cccccccc}
\mathbf{X_1} & \mathbf{X_2} & \mathbf{X_3} & \mathbf{X_4} & \mathbf{X_5} & \cdots & \mathbf{X_p} \\
x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & \cdots & x_{1p} \\
x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & \cdots & x_{2p} \\
x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & \cdots & x_{3p} \\
x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & \cdots & x_{4p} \\
 & & & \cdots & & & \\
 & & & \cdots & & & \\
x_{n1} & x_{n2} & x_{n3} & x_{n4} & x_{n5} & \cdots & x_{np}
\end{array}
$$

Some preliminaries

- Sample mean vector: $<\mu_1, \mu_2, \mu_3, \ldots, \mu_p>$
  For the i^{th} variable: $\mu_i = (\Sigma^n_{j=1} x_{ij})/n$
- Variance for the i^{th} variable:
  $$\sigma_i^{\,2} = [\Sigma^n_{j=1} (x_{ij} - \mu_i)^2] / [n-1]$$
- Sample covariance:
  $$c_{ab} = [\Sigma^n_{j=1} ((x_{aj} - \mu_a)(x_{bj} - \mu_b))] / [n-1]$$

This measures the correlation in the data

In fact, the correlation coefficient

$$r_{ab} = c_{ab} / \sigma_a \sigma_b$$

Standardize the variables

- For each variable x_{ij}

  Replace the values by

$$y_{ij} = (x_{ij} - \mu_i)/\sigma_i^{\,2}$$

Correlation Matrix

$$R = \begin{bmatrix} 1 \;\; r_{12} \;\; r_{13} \ldots \; r_{1p} \\ r_{21} \;\; 1 \;\; r_{23} \cdots \; r_{2p} \\ \vdots \\ r_{p1} \;\; r_{p2} \;\; r_{p3} \cdots 1 \end{bmatrix}$$