CS772: Deep Learning for Natural Language Processing (DL-NLP)

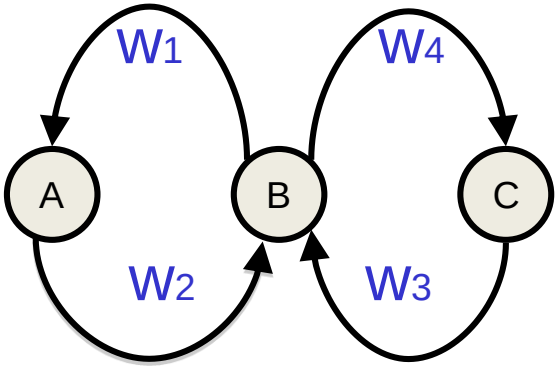
BPTT, Hopfield-net, LSTM

Pushpak Bhattacharyya
Computer Science and Engineering Department
IIT Bombay

Week 7 of 14th Feb, 2022

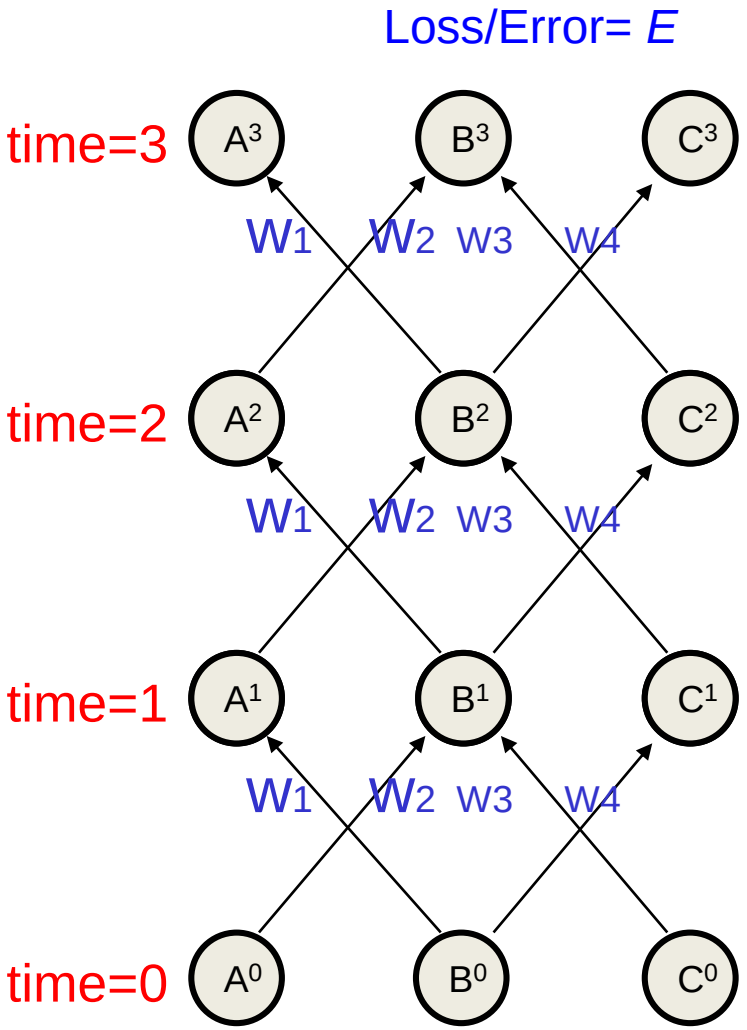
RNN-BPTT

The equivalence between feedforward nets and recurrent nets



Assume that there is a time delay of 1 in using each connection.

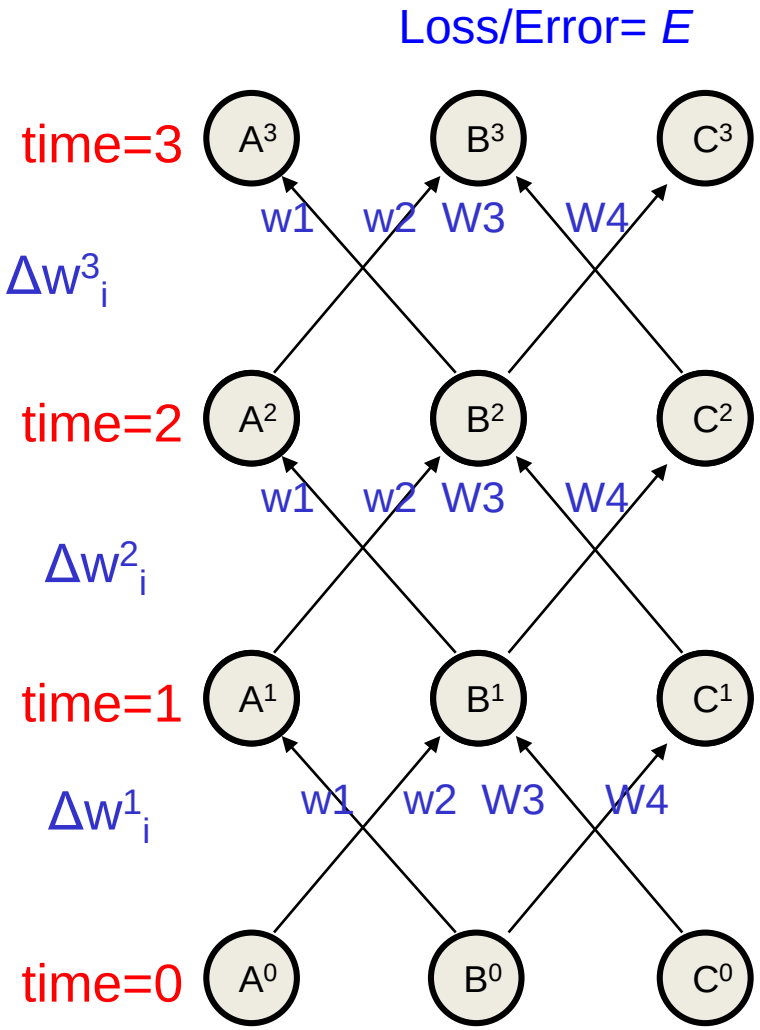
The recurrent net is just a layered net that keeps reusing the same weights.



BPTT illustration

$\Delta w_i = \Delta w^3_i + \Delta w^2_i + \Delta w^1_i$

Vanishing/Exploding
Gradient can strike!!!



An NLP example (1/2)

- The RNN in the previous slide can be used for POS tagging of trigrams.
- “Purchased Videocon Machine”→ VBD NNP NN
- We would know the loss value after this sequence
- Can apply gradient descent
- Obtain all Δws
- Post them

An NLP example (2/2)

- Similarly for “The Blue Sky”→ DT JJ NN
- This way was obtain Δws for each input-output pattern
- At the end the RNN will learn Tri-gram POS tagging

BPTT important points

- The forward pass at each time step.
- The backward pass computes the error derivatives at each time step.
- After the backward pass we add together the derivatives at all the different times for each weight.

Long word sequences

- The famous book by Charles Dickens “A Tale of Two Cities” starts the book with the famous sentence “This was the best of times, this was the worst of times….”
- The sentence has 119 words
- “It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, we were all going direct to Heaven, we were all going direct the other way--in short, the period was so far like the present period that some of its noisiest authorities insisted on its being received, for good or for evil, in the superlative degree of comparison only.

The “best of times…” sentence

- Vanishing gradient will surely strike!!
- Exercise: give an example from NLP, where exploding gradient will strike!!

A Classical RNN: Hopfield Net

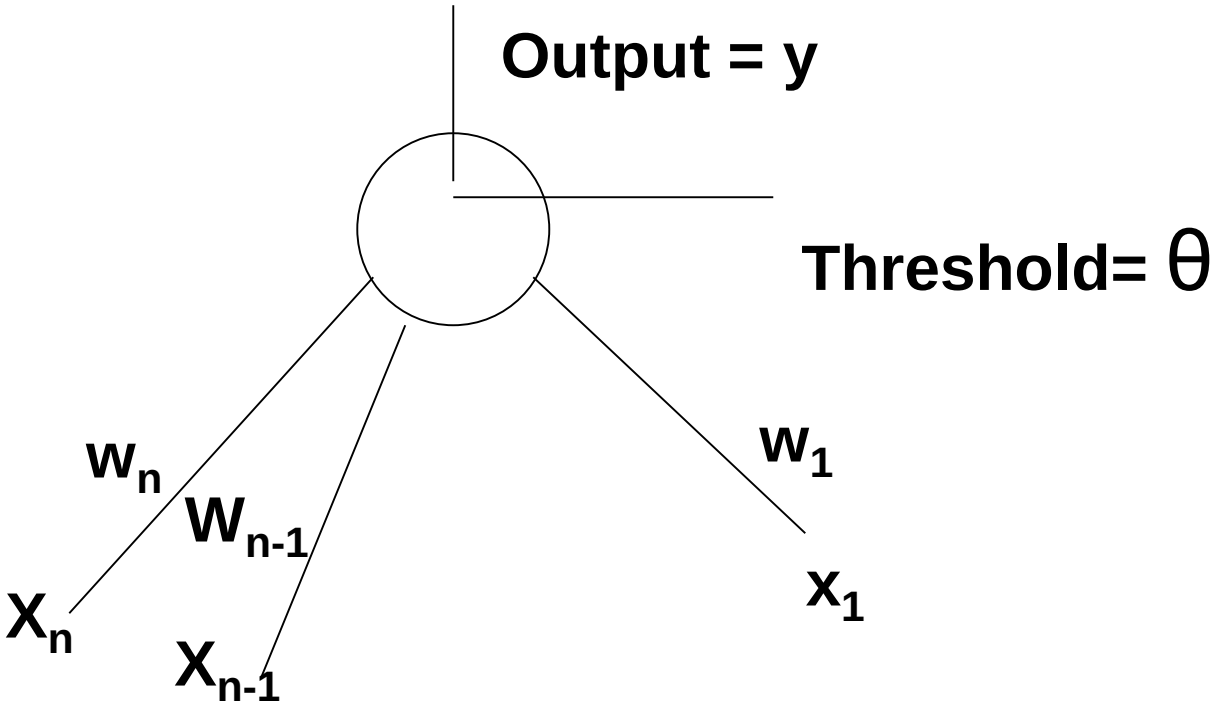
Hopfield net

- Inspired by associative memory which means memory retrieval is not by address, but by part of the data.
- Consists of

  N neurons fully connected with

  symmetric weight strength $w_{ij} = w_{ji}$
- No self connection. So the weight matrix is 0-diagonal and symmetric.
- Each computing element or neuron is a linear threshold element with threshold = 0
- Binary neurons with activations as +1 or -1

Computation



Example

$w_{12} = w_{21} = 5$

$w_{13} = w_{31} = 3$

$w_{23} = w_{32} = 2$

θ=0 for each neuron

At time *t=0*

$s_1(t) = 1$

$s_2(t) = -1$

$s_3(t) = 1$

Unstable state: Neuron 1 will flip.

A stable pattern is called an attractor for the net.

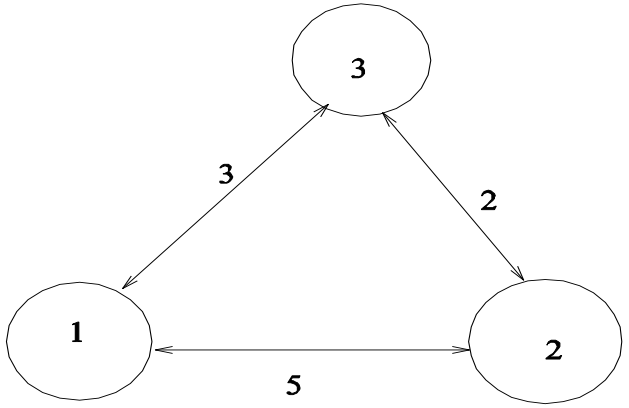


Figure: An example Hopfield Net

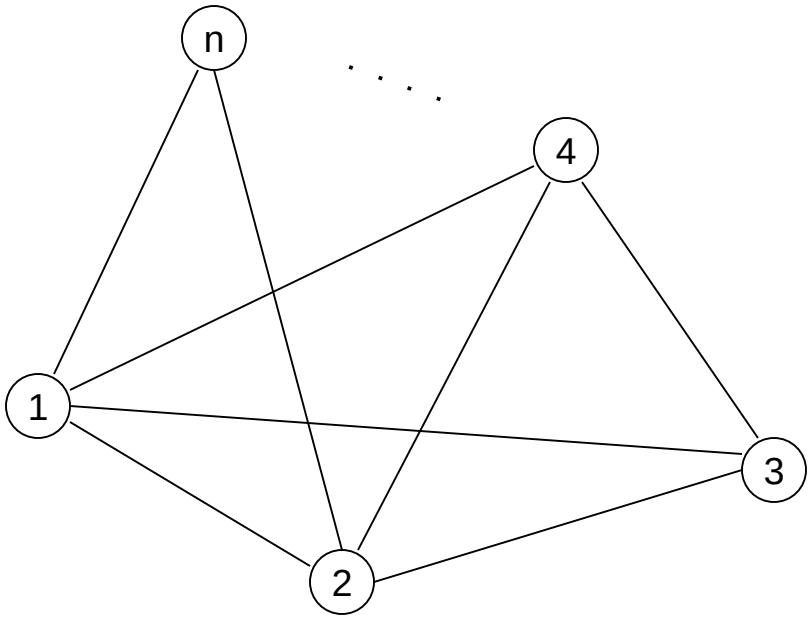
Stability

- Asynchronous mode of operation: at any instant a randomly selected neuron compares the net input with the threshold.
- In the *synchronous* mode of operation all neurons update themselves simultaneously at any instant of time.
- Since there are feedback connections in the Hopfield Net the question of *stability* arises. At every time instant the network evolves and finally settles into a stable state.
- How does the Hopfield Net function as *associative* memory ?
- One needs to store or stabilize a vector which is the memory element.

Energy consideration

- Stable patterns correspond to minimum energy states.
- Energy at state $<x_1, x_2, x_3, \ldots, x_n>$
- $$E = -1/2 \sum_j \sum_{j \neq i} w_{ji} x_i x_j$$
- Change in energy always comes out to be negative in the asynchronous mode of operation. Energy *always* decreases.
- Stability ensured.

Hopfield Net is a fully connected network



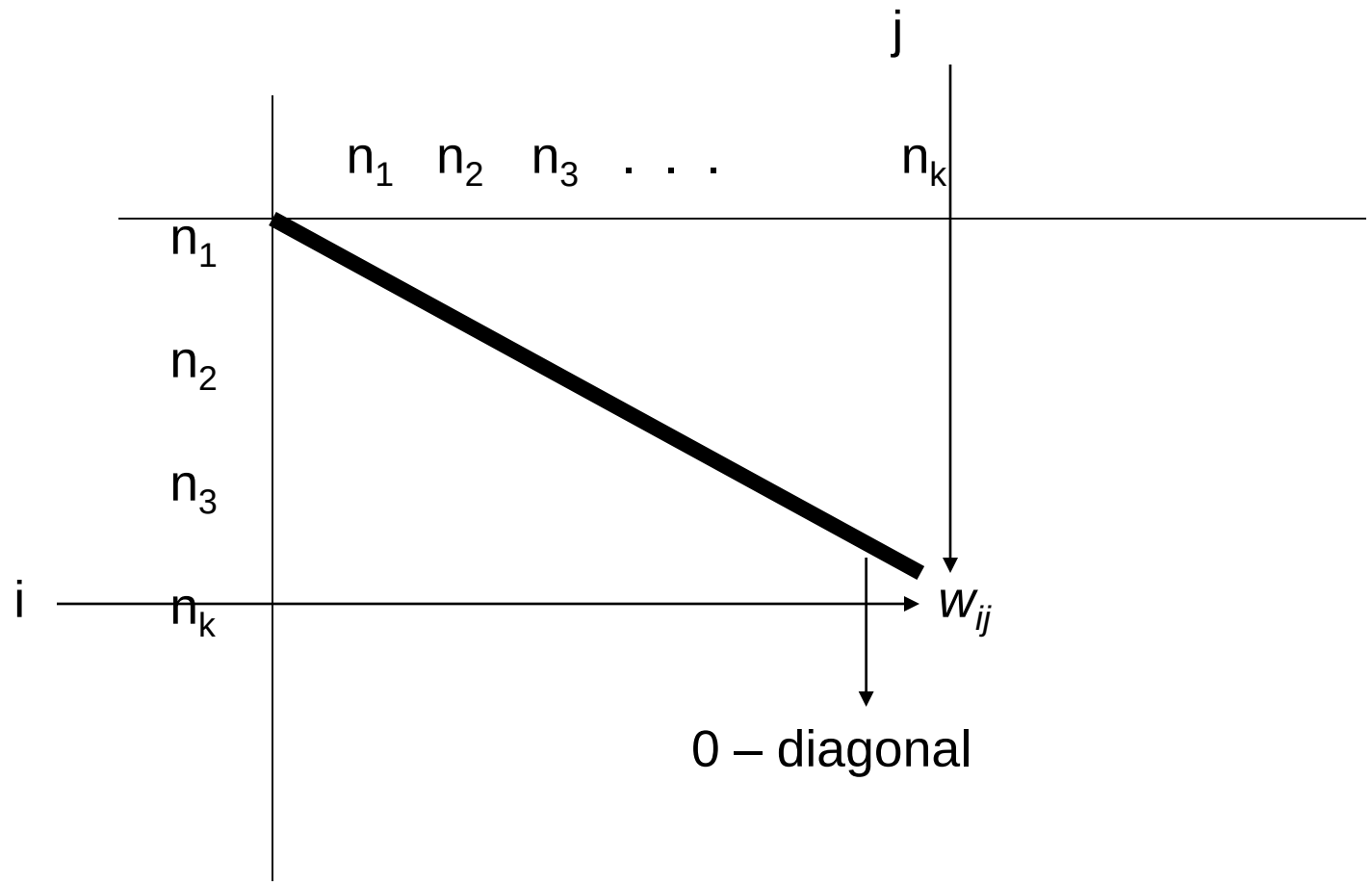
- i^{th} neuron is connected to *(n-1)* neurons

Concept of Energy

- Energy at state s is given by the equation:

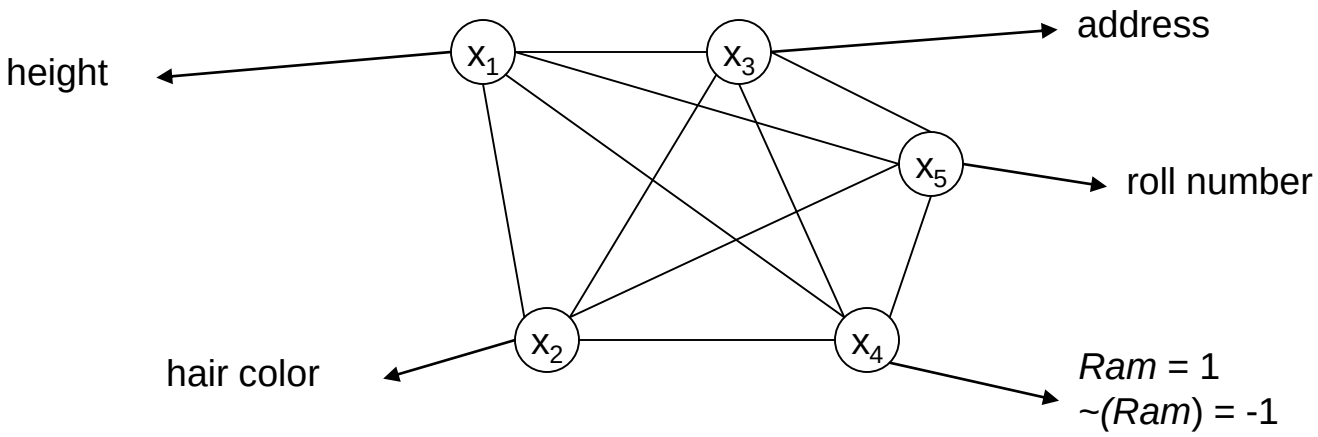
$$E(s) = -\Big[w_{12}x_1x_2 + w_{13}x_1x_3 + \ldots + w_{1n}x_1x_n + w_{23}x_2x_3 + \ldots + w_{2n}x_2x_n + \vdots + w_{(n-1)n}x_{(n-1)}x_n\Big]$$

Connection matrix of the network, 0-diagonal and symmetric



State Vector

- Binary valued vector: value is either 1 or -1, $X = \langle x_n \quad x_{n-1} \quad . \ . \ . \ . \quad x_3 \quad x_2 \quad x_1 \rangle$
- *e.g.* Various attributes of a student can be represented by a state vector



Relation between weight vector W and state vector X

$$W \cdot X^T$$

Weight vector

Transpose of state vector

For example, in figure 1,
At time *t = 0*, state of the neural network is:
s(0) = <1, -1, 1> and corresponding vectors are as shown.

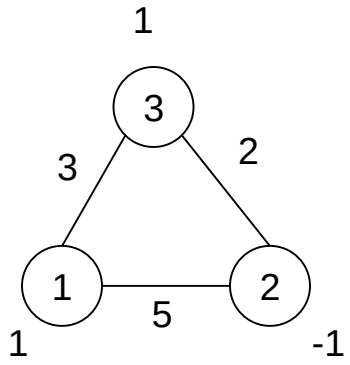


Fig. 1

$$W = \begin{bmatrix} 0 & 5 & 3 \\ 5 & 0 & 2 \\ 3 & 2 & 0 \end{bmatrix} \qquad X^T = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$$

$$W \cdot X^T = \begin{bmatrix} 0 & 5 & 3 \\ 5 & 0 & 2 \\ 3 & 2 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$$

$W.X^T$ gives the inputs to the neurons at the next time instant

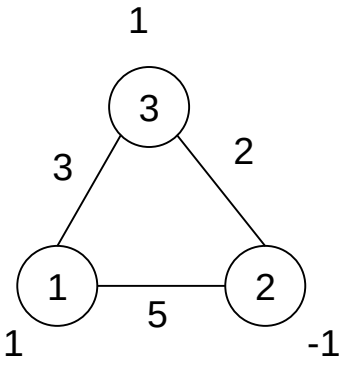
$$W \cdot X^T = \begin{bmatrix} \mathbf{0} & \mathbf{5} & \mathbf{3} \\ \mathbf{5} & \mathbf{0} & \mathbf{2} \\ \mathbf{3} & \mathbf{2} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ \mathbf{-1} \\ \mathbf{1} \end{bmatrix}$$

$$= \begin{bmatrix} -2 \\ 7 \\ 1 \end{bmatrix}$$

$$\text{sgn}(W.X^T) = \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}$$

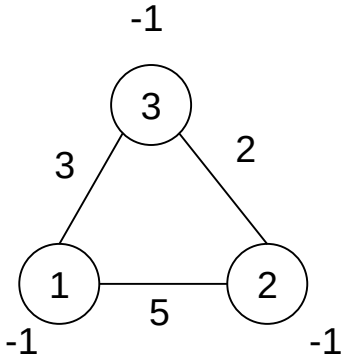
This shows that the n/w will change state

Energy Consideration



At time *t = 0*, state of the neural network is:
s(0) = <1, -1, 1>

- E(0) = -[(5*1*-1)+(3*1* 1)+(2*-1*1)] = 4



The state of the neural network under stability is *<-1, -1, -1>*

E(stable state) = - -[(5*-1*-1)+(3*-1* -1)+(2*-1*-1)] = -10

State Change

- *s(0) = <1, -1, 1…>*
- *s(1) = compute by comparing and summing*
- $x_1(t=1)=sgn[\Sigma_{j=2}^{n} w_{1j}x_j]$

  $= 1 \text{ if } \Sigma_{j=2}^{n} w_{1j}x_j > 0$

  $= -1 \text{ otherwise}$

Theorem

- In the asynchronous mode of operation, the energy of the Hopfield net always decreases.
- Proof:

$$E(t_1) = -\Big[w_{12}x_1(t_1)x_2(t_1) + w_{13}x_1(t_1)x_3(t_1) + \ldots + w_{1n}x_1(t_1)x_n(t_1)$$
$$+ w_{23}x_2(t_1)x_3(t_1) + \ldots + w_{2n}x_2(t_1)x_n(t_1) +$$
$$\vdots$$
$$+ w_{(n-1)n}x_{(n-1)}(t_1)x_n(t_1)\Big]$$

Proof

- Let neuron 1 change state by summing and comparing
- We get following equation for energy

$$E(t_2) = -\Big[w_{12}x_1(t_2)x_2(t_2) + w_{13}x_1(t_2)x_3(t_2) + \ldots + w_{1n}x_1(t_2)x_n(t_2)$$
$$+ w_{23}x_2(t_2)x_3(t_2) + \ldots + w_{2n}x_2(t_2)x_n(t_2) +$$
$$\vdots$$
$$+ w_{(n-1)n}x_{(n-1)}(t_2)x_n(t_2)\Big]$$

Proof: *note that only neuron 1 changes state*

$$\Delta E = E(t_2) - E(t_1)$$

$$= -\{[w_{12}x_1(t_2)x_2(t_2) + w_{13}x_1(t_2)x_3(t_2) + \ldots + w_{1n}x_1(t_2)x_n(t_2)]$$

$$-[w_{12}x_1(t_1)x_2(t_1) + w_{13}x_1(t_1)x_3(t_1) + \ldots + w_{1n}x_1(t_1)x_n(t_1)]\}$$

$$= -\sum_{j=2}^{n} w_{1j}[x_1(t_2)\cdot x_j(t_2) - x_1(t_1)\cdot x_j(t_1)]$$

Since only neuron 1 changes state, $x_j(t_1)=x_j(t_2)$, j=2, 3, 4, …n, and hence

$$= \sum_{j=2}^{n}[w_{1j}\cdot x_j(t_1)][x_1(t_1) - x_1(t_2)]$$

Proof *(continued)*

$$= \sum_{j=2}^{n} \left[w_{1j} \cdot x_j(t_1) \right] \left[x_1(t_1) - x_1(t_2) \right]$$

(S) (D)

- Observations:
  - When the state changes from -1 to 1, *(S) has to be +ve and (D) is –ve;* so ΔE becomes negative.
  - When the state changes from 1 to -1, *(S) has to be -ve and (D) is +ve;* so ΔE becomes negative.
- Therefore, Energy for any state change always decreases.

The Hopfield net has to "converge" in the asynchronous mode of operation

- As the energy *E* goes on decreasing, it has to hit the bottom, since the weight and the state vector have finite values.
- That is, the Hopfield Net has to converge to an energy minimum.
- Hence the Hopfield Net reaches stability.

Probabilistic state change

- If the state change happens with a probability, we have what is called a Boltzmann Machine
- State change is controlled by Boltzmann Distribution
- Randomness controlled by “Temperature”

Boltzmann Machine

Comparative Remarks

Feed forward n/w with BP	**Hopfield net**	**Boltzmann m/c**
Mapping device: (i/p pattern --> o/p pattern), *i.e.* Classification	Associative Memory + Optimization device	Constraint satisfaction. (Mapping + Optimization device)
Minimizes total sum square error, typically	Energy	Entropy (Kullback–Leibler **divergence**)

Comparative Remarks (contd.)

Feed forward n/w with BP	**Hopfield net**	**Boltzmann m/c**
Deterministic neurons	Deterministic neurons	Probabilistic neurons
Learning to associate i/p with o/p *i.e.* equivalent to a function	Pattern	Probability Distribution

Boltzmann Machine

- Hopfield net
- Probabilistic neurons
- Energy expression = $-\Sigma_i \Sigma_{j>i} w_{ij} x_i x_j$
  where x_i = activation of i^{th} neuron
- Used for optimization
- Central concern is to ensure global minimum
- Based on simulated annealing

Meaning of “Learning” a Probability Distribution

- If we have standard distributions like normal, binomial, poisson etc., learning the distribution means estimating the parameters like mean, standard deviation etc.
- A feedforward NN with softmax, also learns a probability distribution, though the there is no closed form expression per se for the learnt distribution

For example

Say, a FFNN with softmax learns POS tags of trigrams

Then the softmax values at the outer layer give

$$P(T|W)$$

Where W is the word sequence, T is the tag sequence

This learning is ensured by fixing the weight values of the FNNN

So P(VBD NNP NN|purchased Videocon machine) will be high after training, and P(JJ NN VB|-same-) will be low

Comparative Remarks (contd.)

Feed forward n/w with BP	**Hopfield net**	**Boltzmann m/c**
Can get stuck in local minimum (Greedy approach)	Local minimum possible	Can come out of local minimum
Credit/blame assignment (consistent with Hebbian rule)	Activation product (consistent with Hebbian rule)	Probability and activation product (consistent with Hebbian rule)

Explaining Hebb's Rule

- Law of nature
- Comes from neurophysiology
- States, "if two connected neurons are both in a state of excitation, the change in weight will be so as to maintain the state of exciation"
- This means two connected neurons in states +1 each, the change in weight will be positive

Theory of Boltzmann m/c

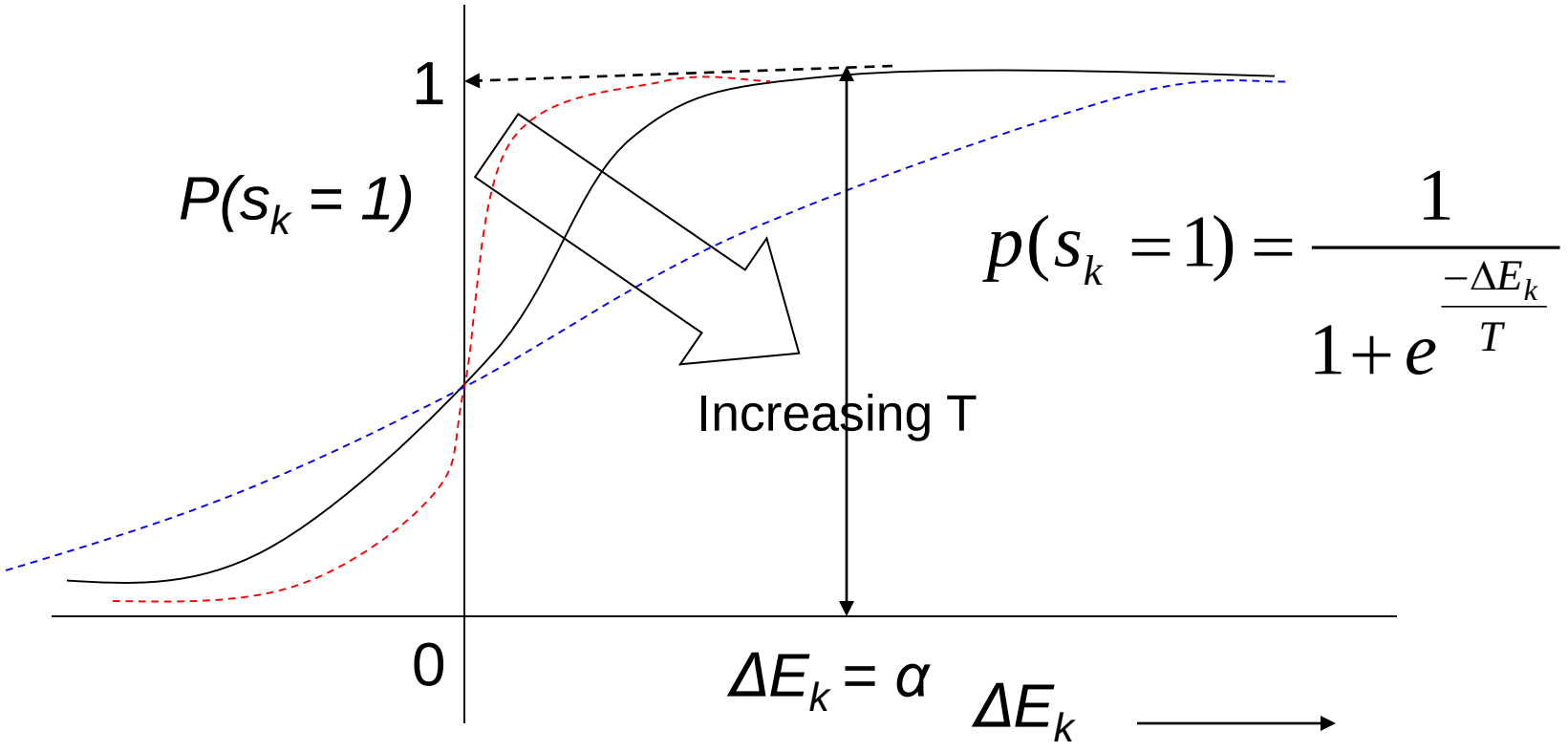
For the m/c the computation means the following: at any time instant, make the state of the k^{th} neuron (s_k) equal to 1, with probability:

$$P(s_k = 1) = \frac{1}{1 + e^{\frac{-\Delta E_k}{T}}}$$

ΔE_k = change in energy of the m/c when the k^{th} neuron changes state

T = temperature which is a parameter of the m/c

Theory of Boltzmann m/c (contd.)



Theory of Boltzmann m/c (contd.)

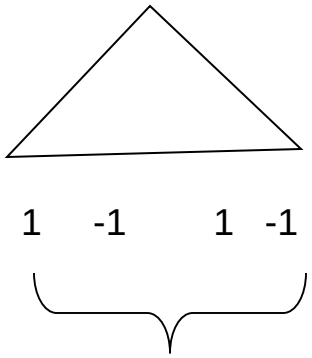
$$\Delta E_k = (E_f - E_i)|_k$$
$$= (S_i - S_k)|_k \cdot \sum_{j \neq k} w_{kj} s_j$$

We observe:

1. The higher the temperature, lower is $P(S_k=1)$
2. at $T = infinity$, $P(S_k=1) = P(S_k=0) = 0.5$, equal chance of being in state 0 or 1. Completely random behavior
3. If $T \rightarrow 0$, then $P(S_k=1) \rightarrow 1$
4. The derivative is proportional $P(S_k=1)*(1 - P(S_k=1))$

Consequence of the form of $P(S_k=1)$

$$P(S_\alpha) = K.e^{-\frac{E_\alpha}{T}}$$



N - bits

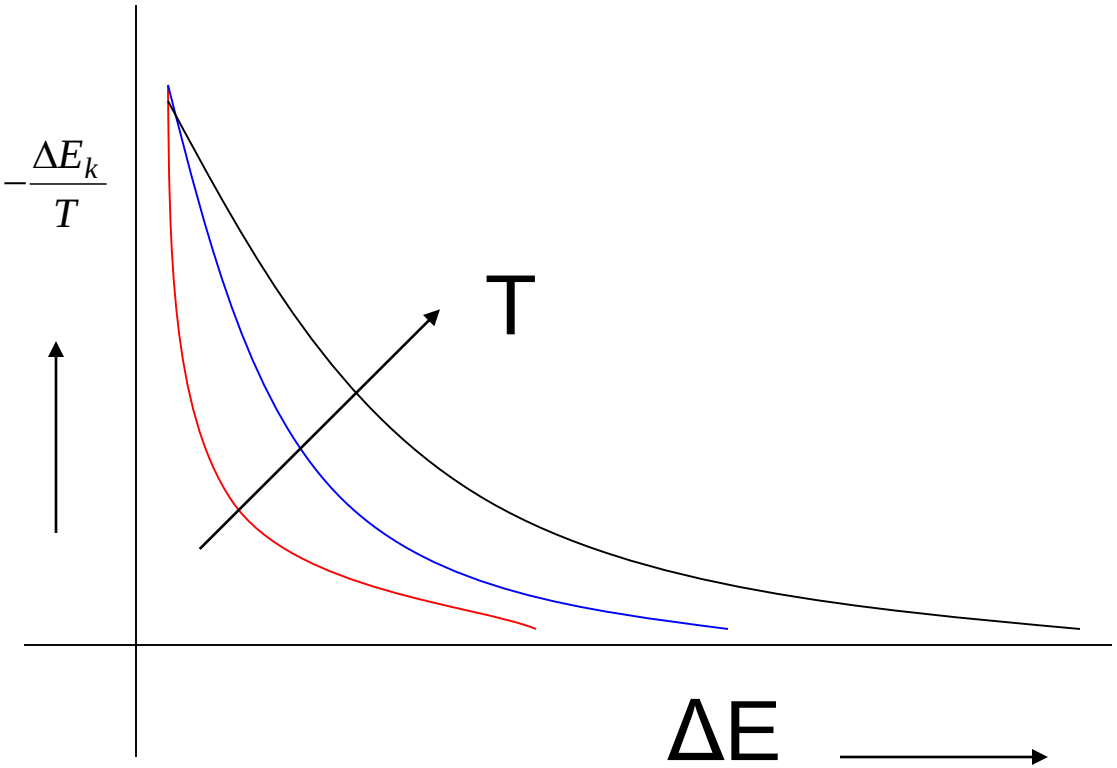
Probability Distribution called as Boltzmann Distribution

$P(S_\alpha)$ is the probability of the state S_α

Local “sigmoid” probabilistic behavior leads to global Boltzmann Distribution behaviour of the n/w

(Ex: try to prove the above)

$$P(S_\alpha) = K.e^{-\frac{\Delta E_k}{T}}$$



Ratio of state probabilities

Normalizing (α and β are states)

$$P(S_\alpha) = K.e^{-\frac{E(S_\alpha)}{T}}$$

$$= \frac{e^{-\frac{E(S_\alpha)}{T}}}{\sum_\beta e^{-\frac{E(S_\beta)}{T}}}$$

Learning a probability distribution

- Digression: Estimation of a probability distribution *Q* by another distribution *P*

- *D* = deviation = $\sum_{sample\ space} Q \ln Q/P$
- *D* >= *0*, which is a required property (just like sum square error >= 0)

To prove $D >= 0$

Lemma: $\ln(1/x) >= (1 - x)$

Let, $x = 1 / (1 - y)$

$\ln(1 - y) = -[y + y^2/2 + y^3/3 + y^4/4 + \ldots]$

Proof (contd.)

$$(1 - x) = 1 - 1/(1 - y)$$
$$= -y(1 - y)^{-1}$$
$$= -y(1 + y + y^2 + y^3 + \ldots)$$
$$= -[y + y^2 + y^3 + y^4 + \ldots]$$

But, $y + y^2/2 + y^3/3 + \ldots <= y + y^2 + y^3 + \ldots$

Lemma proved.

Proof (contd.)

$$D = \sum Q \ln Q / P$$

$$>= \sum_{\text{over the sample space}} Q(1 - P/Q)$$

$$= \sum (Q - P)$$

$$= \sum Q - \sum P$$

$$= 1 - 1 = 0$$