Neural networks and physical systems with emergent collective computational abilities
By - J. J. HOPFIELD

Anurag Kumar
Associative Memory

• “Remembering”?
  – Associating something with sensory cues
• Cues in terms of text, picture or anything
• Modeling the process of memorization
• The minimum requirements of a content addressable memory
Associative Memory

• Able to store multiple independent patterns
• Able to recall patterns with reasonable accuracy
• Should be able to recall partial or noisy input patterns
• Other expectations – Speed and biological similarities
Associative Memory

Original ‘T’

half of image corrupted by noise

20% corrupted by noise (whole image)
Modeling Memory

• Analogy with physical systems
• Systems that flow towards locally stable points
• Ball in a bowl
• Think in terms of “remembering” the bottom of bowl

• Initial position is now our sensory cue
Modeling Memory

• Surface with several stable points -> storing several patterns

• Depending on initial position (sensory cue) the ball will end up in one of the stable points (stored pattern)

• Stable point closest to initial position is reached
Modeling Memory

• Two important points
  – Stores a set patterns and recalls the one closest to initial position
  – Minimum energy state is reached

• Properties of network
  – Described by state vectors
  – Has a set of stable states
  – Network evolves to reach stable state closest to initial state resulting in decrease of energy

notes from http://www.cs.ucla.edu/~rosen/
Hopfield Networks

• A recurrent neural network which can function as a associative memory
• A node is connected to all other nodes (except with itself)
• Each node behaves as an input as well as output node
• Network’s current state defined by outputs of nodes
Hopfield Networks

• Nodes are not self connected \( w_{ii} = 0 \)
• Strength of connections are symmetric meaning \( w_{ji} = w_{ij} \)
Hopfield Networks

- Updating the node outputs
- Output of ith node is given by
  - $O(i) = 1$ if $\sum_j W_{ij} x_j \geq \alpha_i$
  - $O(i) = 0$ if $\sum_j W_{ij} x_j < \alpha_i$

- Each node can randomly and asynchronously evaluate if it is above or below the threshold
- Interesting results due to the recurrent nature of network
- Asynchronous updating of nodes is more similar to biological neurons
Hopfield Networks

- Biological Similarity of the model
- Mean Firing Rate or output of a neuron is smooth function of mean membrane potential
- On- Off model of neuron

Fig. 1. Firing rate versus membrane voltage for a typical neuron (solid line), dropping to 0 for large negative potentials and saturating for positive potentials. The broken lines show approximations used in modeling.
Storing the patterns

- Hopfield used
  \[ W_{ij} = \sum_s (2x^s_i - 1)(2x^s_j - 1) \]
- Rules from the Hebb’s family can be used
- Assume two nodes on average take same value (0,0) or (1,1)
- Positive weight would reinforce the pairing
- If the take opposite values on average – a negative weight would reinforce the pairing
- Mathematically, with change of symbol this can be simply seen as
  \[ W_{ij} = \sum_s x^s_i x^s_j \]
Updating the outputs

• Asynchronous: Nodes operate asynchronously
  – Select node and compute output using the weighted sum
  – Selection can be random (as proposed by Hopfield) but a fixed order also possible

• Synchronous: All nodes are updated in one go based on the current values
  - Unrealistic to biological neurons
Energy

- For a single pair of neurons
  \[ e_{ij} = -w_{ij}x_ix_j \]
- The overall energy of network is the sum over all pairs
  \[ E = -\frac{1}{2} \sum_{ij} w_{ij}x_ix_j \]
- What happens to the energy when a node fires?
Energy

• Let node $k$ fires then

$$E = -\frac{1}{2} \sum_{i \neq k, j \neq k} w_{ij} x_i x_j - \frac{1}{2} \sum_i w_{ki} x_k x_i - \frac{1}{2} \sum_i w_{ik} x_i x_k$$

• Using connection symmetry $w_{ik} = w_{ki}$

$$E = -\frac{1}{2} \sum_{i \neq k, j \neq k} w_{ij} x_i x_j - \sum_i w_{ki} x_k x_i$$

• Thus, the energy is

$$E = S - x_k \sum_i w_{ki} x_i$$
Energy

\[ E = S - x_k a^k \]

After node k updates,

\[ E' = S - x'_k a^k \]

• Change in energy is given by

\[ \Delta E = -\Delta x_k a^k \]
Energy

- If $a^k \geq 0$, this means the output has stayed at one or changed to 1 from 0. In any case the change in energy is $\leq 0$.

- If $a^k < 0$, output stayed to 0 or changed to 0 from 1. In any case again the change in energy is again $\leq 0$.

- So updating nodes always results in decrease of energy or it stays the same – essential for the network to finally evolve to a stable state.
Energy

• But, the network can still change ($E - E' = 0$)

• Only finite $N$ possibilities

• The network will thus settle down to a local stable state with local minimum energy
Hopfield Networks

- If $s (101110)$ is a stable state then $\hat{s} (010001)$ is also a stable state.
- Storage Capacity of Hopfield Networks – Hopfield gave a bound of $0.15N$ before severe error in recall.
- A later 1987 paper by McEliece, showed theoretically that storage capacity is $N/(2 \log N)$ to be almost without errors.
Hopfield Networks

• Removing the symmetry criterion $w_{ij} = w_{ji}$
• Simulations showed that in most cases the settling was at a stable state
• The Algorithm, even for non-symmetric case, changes $E$ in similar fashion and stable limit points still persists
• However, non-symmetric weights can lead to metastable minima
• On simulations - starting from any random state network leads to a stable assigned memories with high probability
Hopfield Networks

• Further Applications – Solving Combinatorial Optimization Problems such as travelling salesman problem (1985 paper by Hopfield and Tank)
• Simple Structure with elementary properties in nodes can store and recall pattern
• Energy helps in visualizing the overall activity in the network
• Only one input processes at one time
• Overall gives highly intuitive formation of associative memories
References

• Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational properties. Proceedings of the National Academy of Sciences of the USA,


• Neural Networks by Simon Haykin
• http://www.comp.leeds.ac.uk/ai23
• http://www.cs.ucla.edu/~rosen/161/NeuralNotes/NeuralNotes.html
• http://www.cs.ucla.edu/~rosen/161/