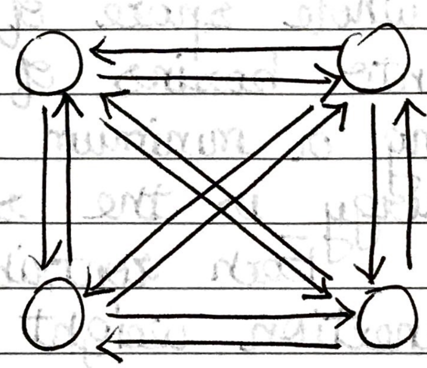
Q]1 Describe Hopfield networks, their functioning and their energy function

Ans ⇒ Hopfield networks was invented by Dr. John Hopfield

⇒ It consists of a single layer which contains one or more fully connected recurrent neurons.

⇒ The hopfield n/w is commonly used for auto-association and optimization tasks.



⇒ A hopfield network operates in a discrete fashion i.e the input & output patterns are discrete vector, which can either binary (0,1) or bipolar (+1,-1) in nature.

⇒ The n/w has symmetrical weights with no-self connection. $w_{ij} = w_{ji}$

⇒ The O/P of each neuron should be the i/p of other neurons but not the input for self.

⇒ The weight can be updated in two ways.

* Synchronous update → where all neuron change their state simultaneously, based on the state of all the other neurons

* Asynchronous update :- where each neuron is updated at a time.

⇒ Given that the connections are symmetric it is possible to build a global energy function. According to it each configuration can be scored.

⇒ It is also possible to look for configurations of minimal energy. In fact the whole space of weights is divided into basins of attraction. Each one containing a minimum of the energy.

⇒ Global energy is the sum of many contributions. Each contribution depends on one connection weight & the binary states of 2 neurons :-

$$E = \frac{-1}{2} \sum_{ij} w_{ij} y_i y_j \quad \frac{-1}{2} \sum_{i} b_i y_i$$

⇒ The simple energy function can make it easier to compute how the states of one neuron affects the global energy.

$$E(y_i = -1) - E(y_i = 1) = \sum_{i} w_{ij} y_j + b_i$$

Q] Discuss how hopfields network can be used as memories, how they can be addressed by partial or corrupted & how many memories are we able to store.

Ans ⇒ Energy minima of the neural network can be used to store memories i.e Memories could be energy minima of a neural net.

⇒ The binary threshold rule (decision) can then be used to clean up partial or corrupted memories.

⇒ If we want to store a set of memories

$$y^{(1)} \ldots y^{(p)} \ldots y^{(P)}$$

$$y^{(p)} = (y_1^{(p)} \ldots y_m^{(p)})$$

⇒ If the states contain/are −1 and 1, then we can use update rule

$$\Delta w_{ji} = \eta \sum_p y_i^{(p)} y_j^{(p)}$$

⇒ But if neurons are 0 and 1, then the rule is

$$\Delta w_{ji} = 4\eta \sum_p (y_i^{(p)} - 1/2)(y_j^{(p)} - 1/2)$$

⇒ The probability of a memory failure retrival increases drastically when the number of stored memory patterns exceeds a fraction of the network size n. by 14%.

$$\frac{P}{N} \text{ exceeds } 0.14$$

Q] Advantages of Boltzman machines compared to hopfield networks

⇒ A Boltzman machine can have hidden units while Hopfield cannot.

⇒ The boltzman machine uses stochastic neurons while the ~~boltzman~~ HF machine uses a deterministic threshold function.

⇒ Hopfield can only support unsupervised learning. While the boltzman can be configured to solve both supervised and unsupervised problems.

Q] 2 Describe Boltzman machines, including their relationships with hopfield n/w

Ans

⇒ A boltzman machine is a network of symetrically connected, neurons like units that make stochastic decisions about whether to be on or off.

⇒ Boltzman machine has a simple learning algorithm that allows them to discover intersting features in dataset composed of binary vectors.

⇒ The boltzman machine can be used to solve two different computational problem. - search problem & learning problem.

⇒ Similar to a hopfield network, a boltzman machine is a n/w of units which are fully interconnected by bidirectional connections with symmetric weights, where no self-connections are allowed. These units have binary values {0, 1} refering to OFF and ON for each unit. That means that the whole system is in a certain state at every time. These states are sometimes referred to as configuration.

⇒ The crucial difference between hopfield n/w & boltzman machine is the way of updating the states of the units which is determined by stochastic decisions.

⇒ Also due to the probabilistic update rule, a BM machine can transit to states on a higher level energy in contrast to HF n/w Thus avoiding getting stuck in local minima of the energy function in minimization problems.

Q] Describe learning in Boltzman machine, include some discussion of how it can be implemented in practice & of the computational challenges it might pose.

Ans ⇒ The learning technique central to boltzmann machine is referred to as simulated annealing.

⇒ The introduction of the notion of temperature introduces a significant divergence from other learning mechanisms.

⇒ The use of temperature introduces a level of randomness.* At high temperature the randomness is high and many different states are possible while at low temperature there is less randomness and the machines is able to settle on and focus on details of learned memory pattern.

⇒ The general idea for learning is :-

1] Starting with a high temperature.

2] At every iteration, selecting a cell at random. If the cell is not clamped then computing a new activation.

AND

Statistics should be collected in order to build up a measure of the probability distribution.

3] Gradually, reducing temperature as iterations proceed. This is the essense of the annealing process.

4] When T is sufficiently small, letting the system reach equilibrium. i.e when no further activations can change. At this point the minimum reached should have the/a high probability of being a global maximum.

⇒ A surprising property of Boltzmann machine is that the learning rule remains the same even with hidden units. i.e

$$\Delta w_{ij} = \eta \sum_{p=1}^{P} \left[y_i^{(p)} y_j^{(p)} - \sum_{y} y_i y_j P(y|w) \right]$$

⇒ * such that the states can be reached by the machine are a function of a probability distribution function. The leve of probability being a ~~texel~~ function of temperature.

⇒ However, computational demands are intensive and in practice this point ma be difficult to reach without compromis

Q] Describe Boltzmann machine with hidden units and RBM, and discuss the difference ~~between~~ with BM with only visible units with respect to expressive power and learning complexity.

Ans

⇒ A BM net seeks the solution by changing the activations of the units (0 or 1) based on a probability distribution & the effect that change would have on the energy function for the net.

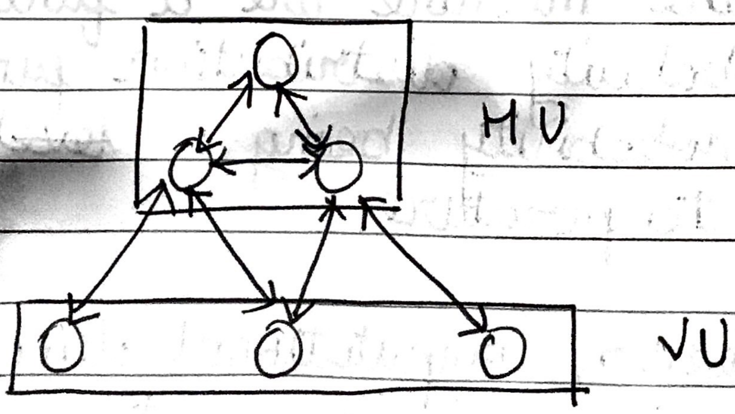
⇒ The neurons in the BM can be categorised into 2 subsets :-

* ~~Input~~ Visible
  - Input
  - Output
* Hidden

⇒ Hidden units are used to represent an interpretation of the inputs.

⇒ Hidden units help us to higher order correlations.

⇒ A BM with hidden units is much harder to train. But it can be powerful when compared with a BM with visible units.



⇒ Learning with BM with HU :-

* The first term is the correlation between neurons i and j when the visible inputs are clamped together.
* The second displays the correlation between neurons i and j when the system is let freely evolve.
* In such cases both terms must be ~~eliminated~~ estimated by letting the n/w involve many times until equilibrium. In one case with the visible units clamped, in the other freely.
* Not only the missing units need to be handled / estimated seperately for each example.
* This process can be computationally very expensive.

⇒ Also hidden units take time to be trained and it is problematic to run a machine with them in any realistic case using a standard algorithm on a CPU.