19 December, 2017

Neural Networks

Course 11: Hopfield nets

Overview

- What is a Hopfield net?
- Energy in a Hopfield net
- Training Hopfield net
- Hopfield problems (Spurious minima)
- Capacity of Hopfield nets

What is a Hopfield Net

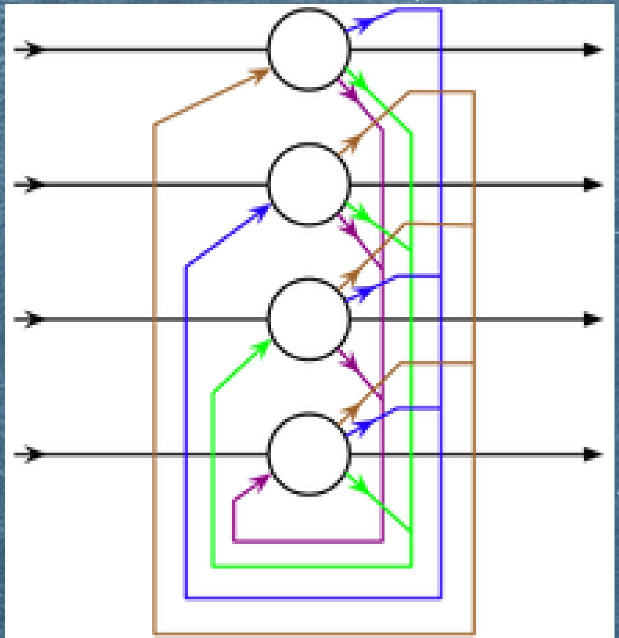
What is a Hopfield net?

- A Hopfield net is a type of recurrent artificial neural network that was popularized by John Hopfield in 1982. It works by first remembering some binary patterns and then returning the one that is most similar to the input

- Due to the way it works, many scientists considered that the Hopfield network could explain the how brain memories are formed and retrieved

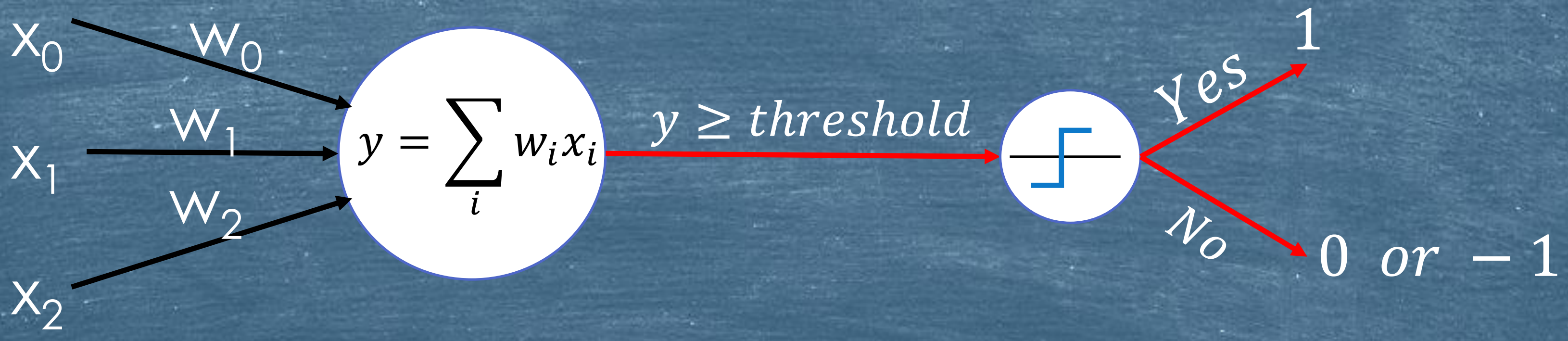
- The simplest recurrent network

What is a Hopfield net?



- It is composed of only one layer of units, binary threshold neurons, where each unit is symmetrically connected with all the other units (except itself).
- So each unit acts as input for other units, but also as output
- Each unit has an output value, known as its binary state, that is usually 0 or 1 (can also be -1 or 1)
- Each unit act as input and output

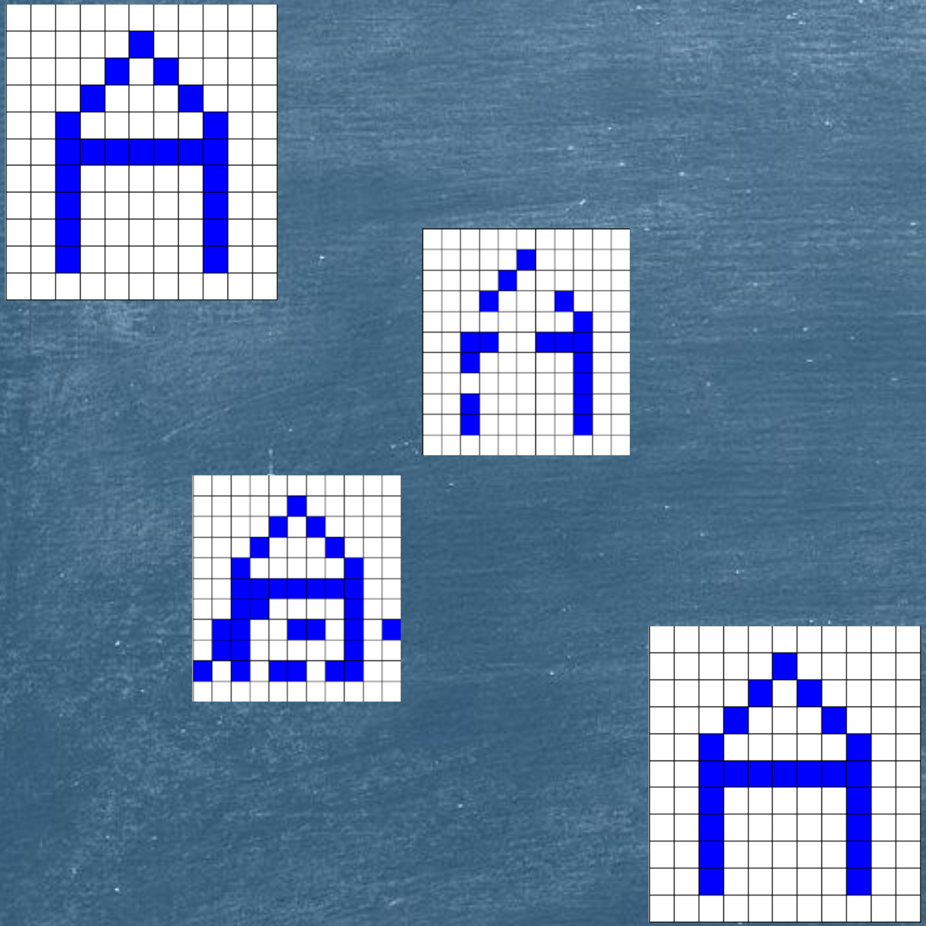
What is a Hopfield net?



if $b = -threshold$, then:

$$s = \begin{cases} 1, if \displaystyle\sum_{i=0}^{n} x_i w_i + b \geq 0 \\ 0 \; or \; -1, otherwise \end{cases}$$

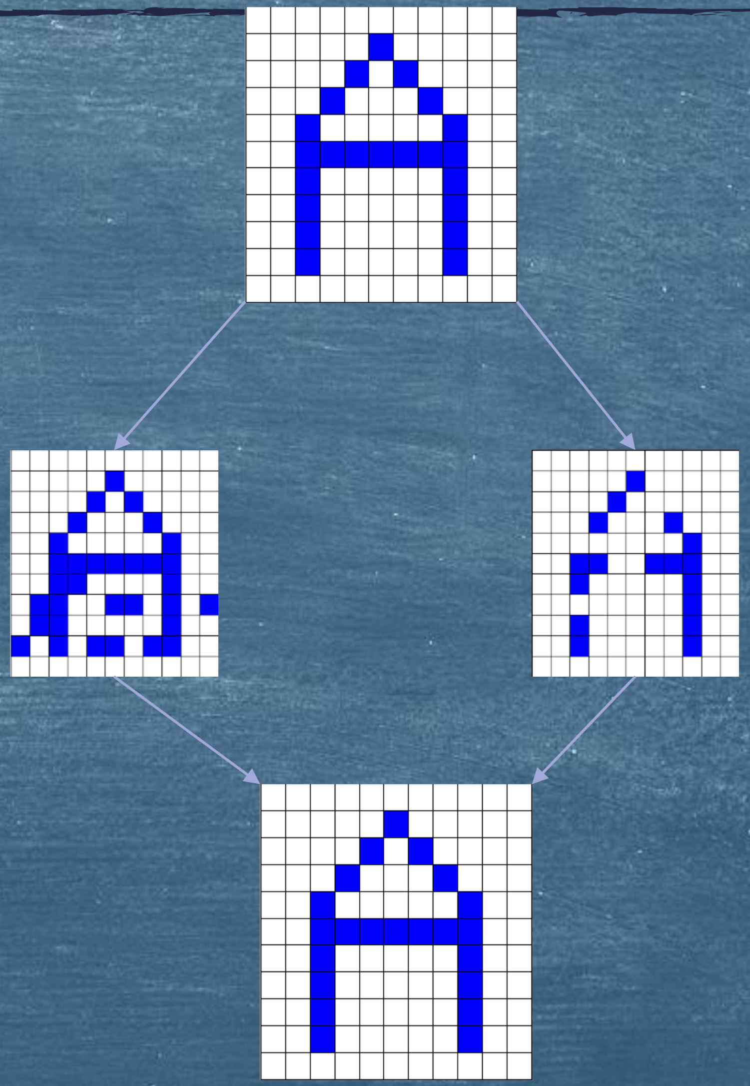
What is a Hopfield net?



- The Hopfield net is considered autoassociative, meaning that it learns to map a noisy input to a learned pattern
- As opposed to indexed memory, associative memory only needs some parts to retrieve a stored item.
- For example, if the letter A was memorized, than you only need a subset of the bits to retrieve it

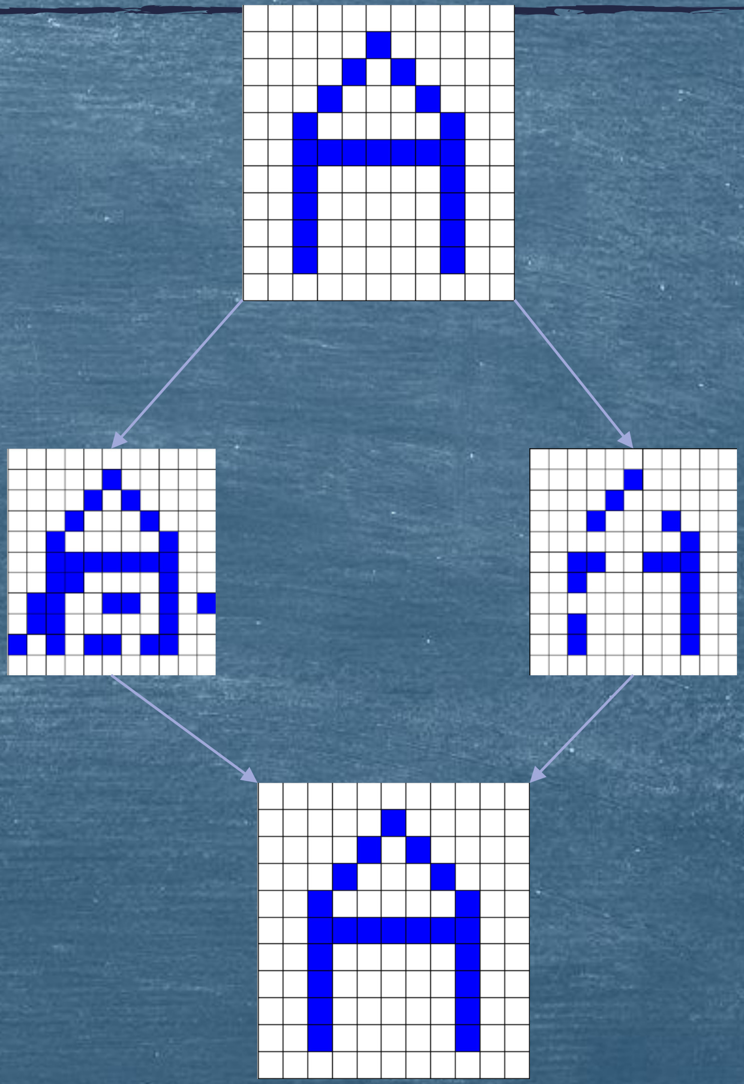
What is a Hopfield net?

- The purpose of a Hopfield net is to rebuild a memory starting from a noisy input:
  - Parts of the memory are missing
  - The input contains data that are not part of the original memory
- The Hopfield net is considered to be autoasociative



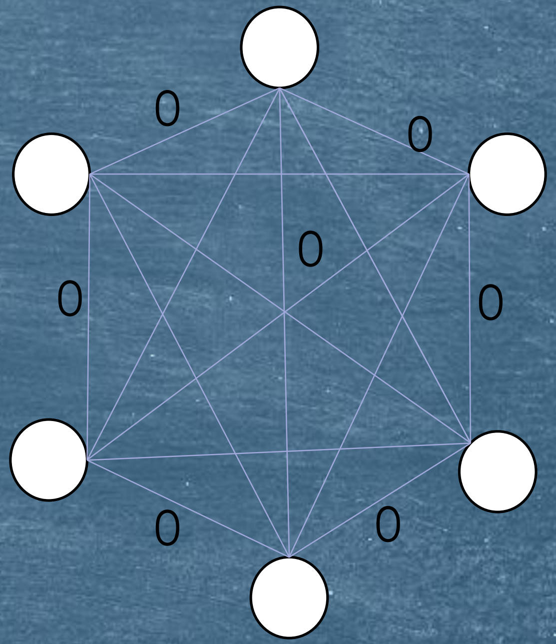
What is a Hopfield net?

- Properties:
  - Information is retrieved based on content (not based on address)
  - It is resistant to damage of some parts
  - Can be used for:
    - Reducing noise
    - Pattern recognition
    - Image reconstruction



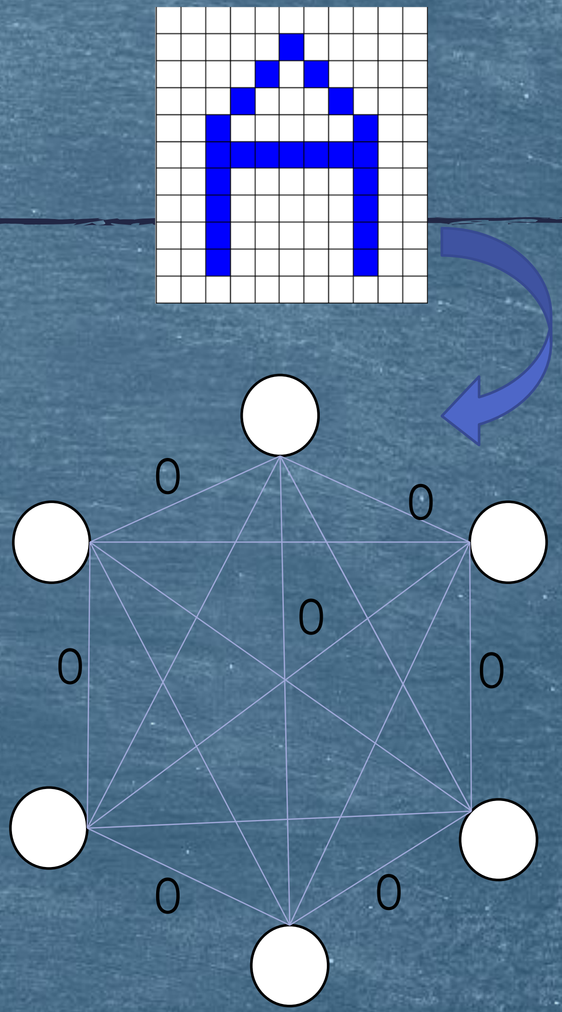
How does it work?

- We start with a network that has as many neurons as the input
- In the beginning all weights are zero



How does it work?

- We start with a network that has as many neurons as the input
- In the beginning all weights are zero
- An input is presented to the network



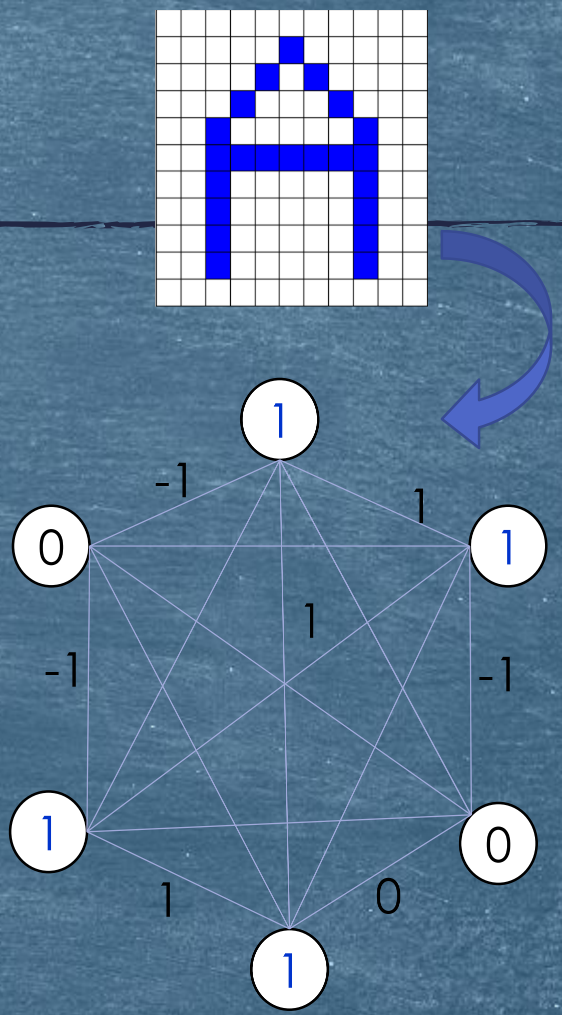
How does it work?

- We start with a network that has as many neurons as the input

- In the beginning all weights are zero

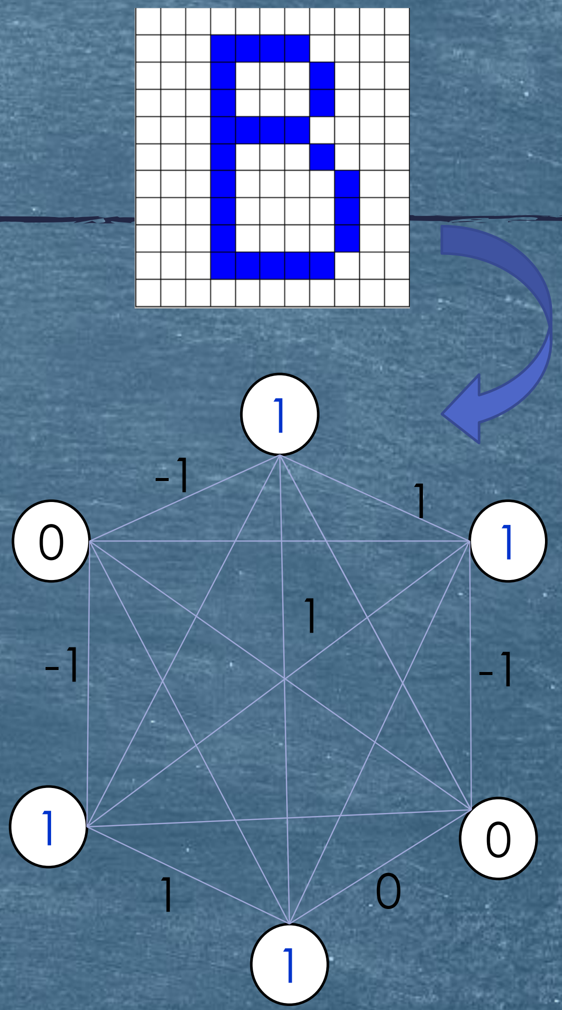
- An input is presented to the network

- Using an algorithm that will be presented later, the weights will adapt such as the input to be memorized



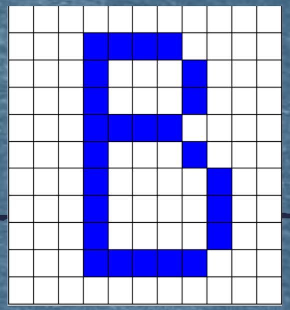
How does it work?

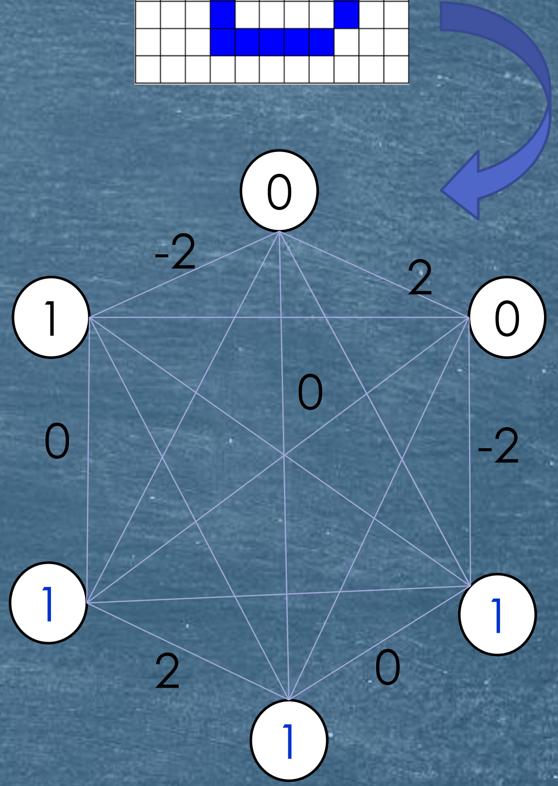
- Another input is presented to the network



How does it work?

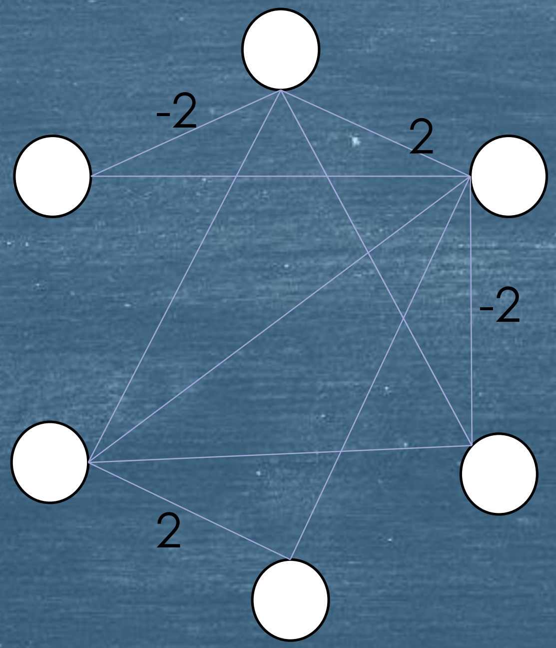
- Another input is presented to the network
- The network will adjust its weights such that the new input will also be retained





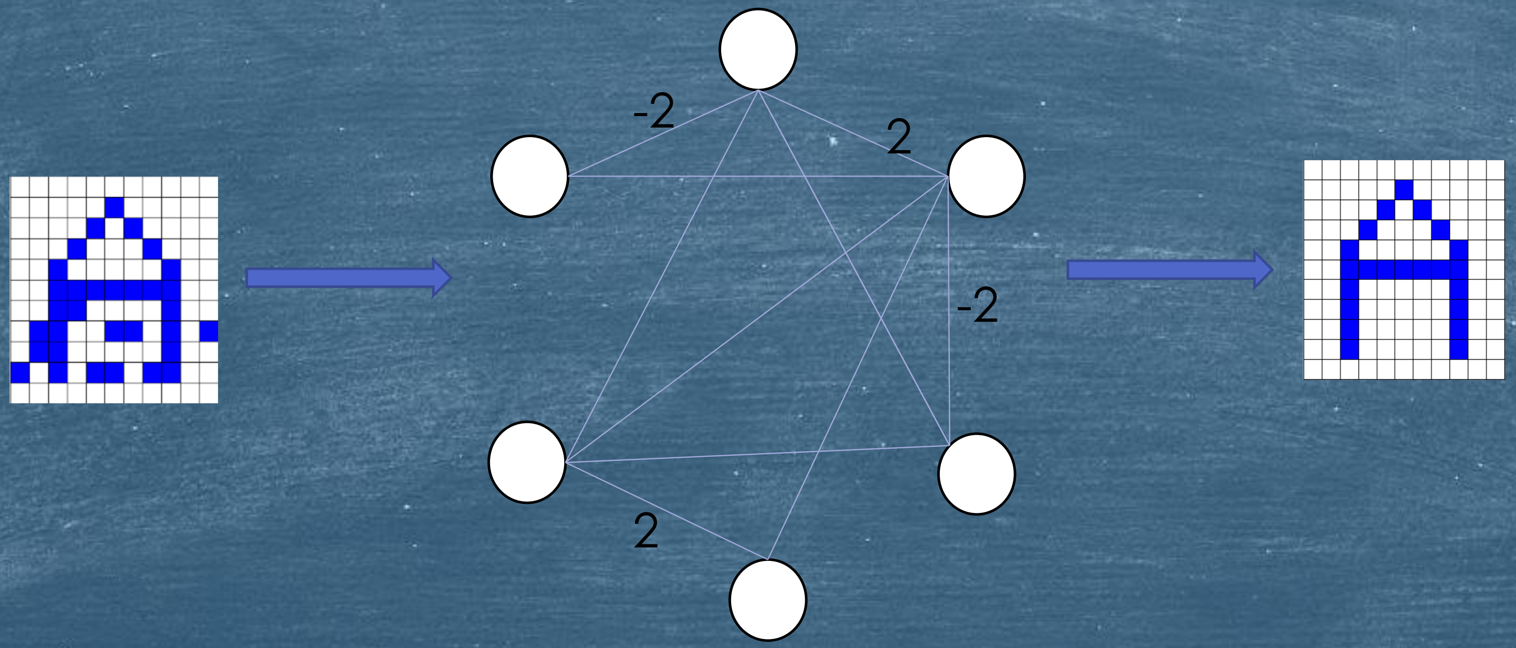
How does it work?

- We have obtained a model that was taught on two separate input instances
- In the representation of the network, the weights that have a weight of value 0 will not be presented anymore



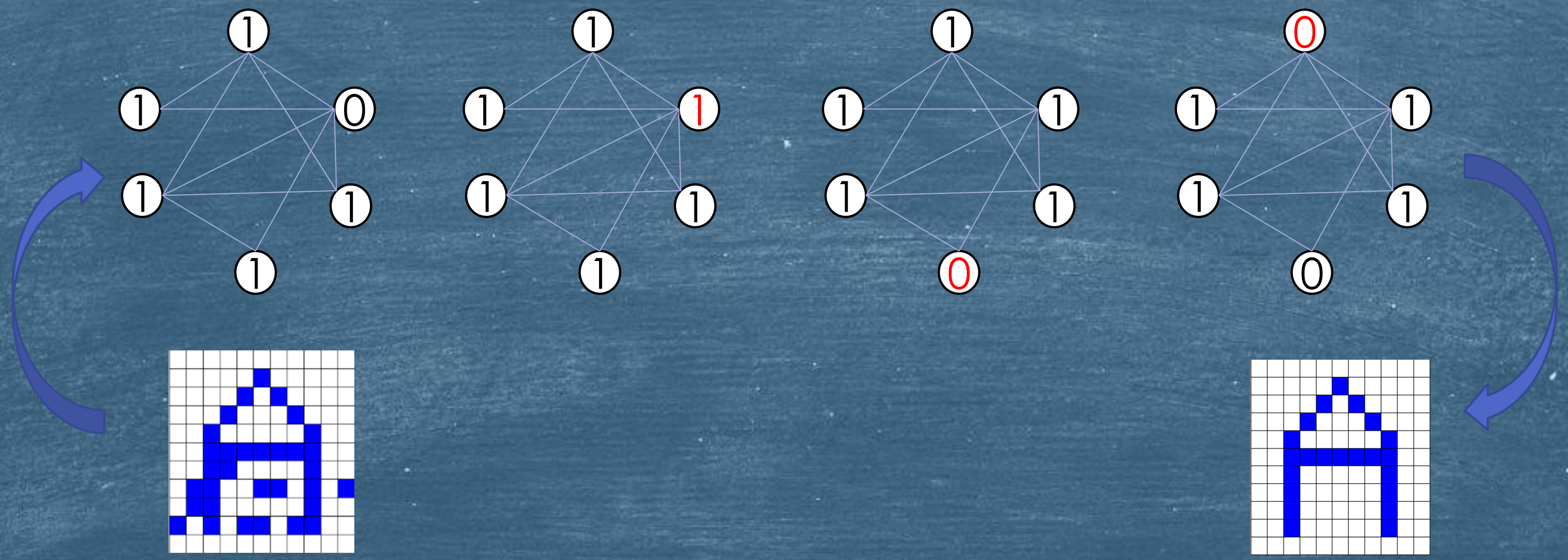
How does it work?

- The resulted network will be further used to transform a noisy input into one of the elements that has remembered during training



How does it work?

- The neurons from the network will be initialized with the values from the input
- Through a series of iterations some of them will change its value
- At the end, the value of each neuron will be read. This should correspond to a value of an element from the training set



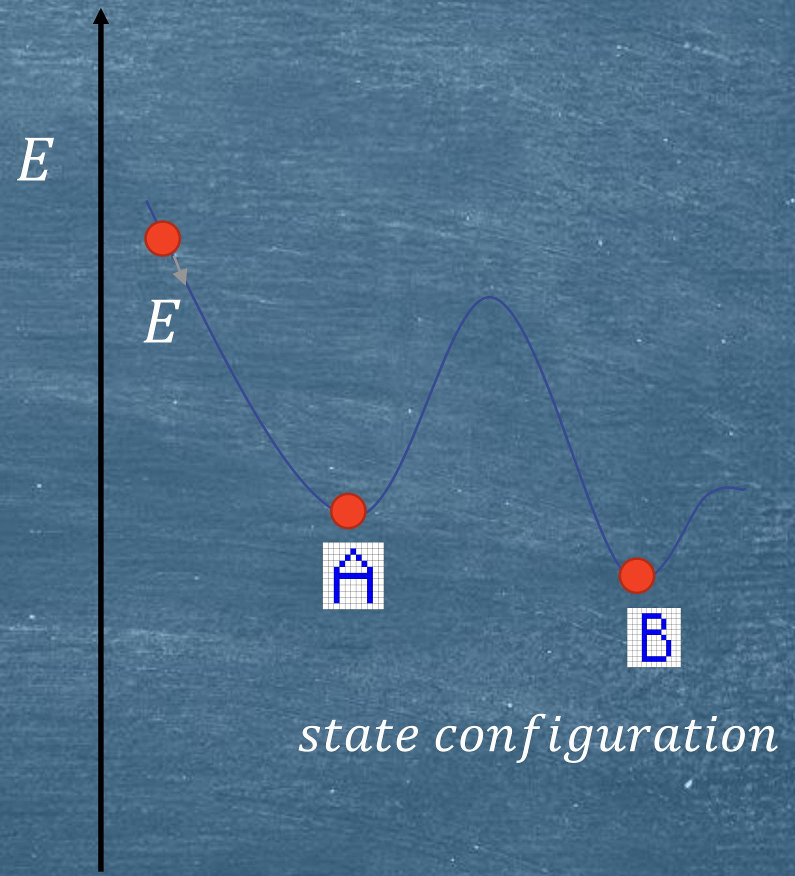
Energy of a Hopfield net

Energy of a Hopfield Net

- Recurrent neural networks are hard to train and analyze, but Hopfield realized that if the weights are symmetric, the learning stabilizes at one point.
- The Hopfield net is also considered an energy based model, since learning is directed by an energy function
- Hopfield realized that if the energy is defined properly, the binary decision rule causes the network to go down in energy and stabilize

Energy of a Hopfield Net

- The main concept that Hopfield networks bring is the Energy:
  - Each configuration of the network has associated an energy
  - The distribution of the energy is given by the weights of the network
  - The inputs learned by the network correspond to points of local minimum in the energy landscape



Energy of a Hopfield Net

- The Energy of the Hopfield network is given by the activation of each neuron and all the interaction between the neurons:

  The global energy is the sum of many contributions, computed locally

$$E = -\sum_{i} s_i b_i - \sum_{i<j} s_i s_j w_{ij}$$

where

$$s_i = activation\ os\ unit\ i$$
$$b_i = the\ bias\ of\ unit\ i$$
$$w_{ij} = the\ symetric\ weight\ between\ unit\ i\ and\ unit\ j$$

Energy of a Hopfield Net

The way energy changes can be computed localy, be each of the neurons (Energy gap)

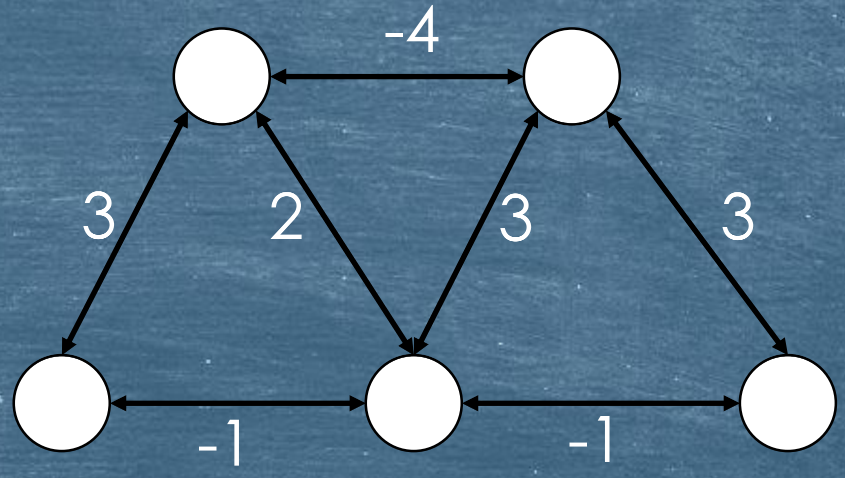
$$\Delta E_i = E(s_i = 0) - E(s_i = 1) = b_i + \sum_j s_j w_{ij}$$

Where
$s_i = activation\ os\ unit\ i$
$b_i = the\ bias\ of\ unit\ i$
$w_{ij} = the\ symetric\ weight\ between\ unit\ i\ and\ unit\ j$

Energy of a Hopfield Net

Finding the Energy minimum (Evaluation):

- Start from a random state
- Update units in random order, one at a time, to the state that gives the lowest global energy
- Stop when there is no unit that changes its output

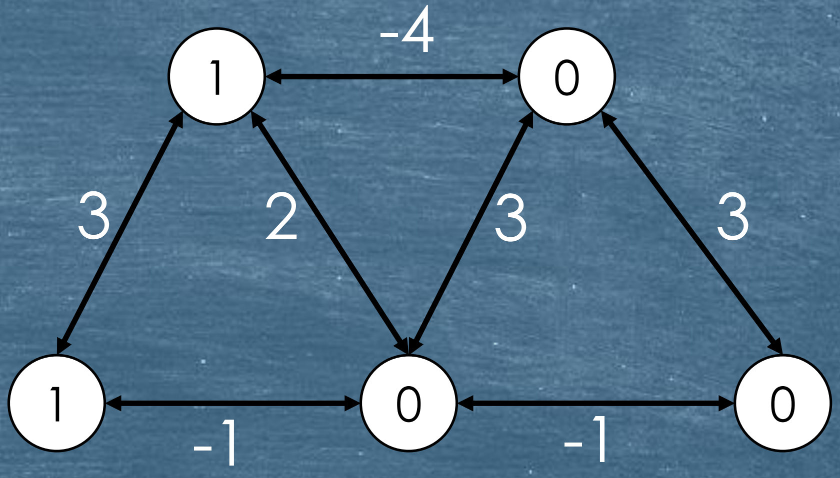


Example taken from http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec11.pdf

Energy of a Hopfield Net

Finding the Energy minimum:

- Start from a random state
- Update units in random order, one at a time, to the state that gives the lowest global energy
- Stop when there is no unit that changes its output



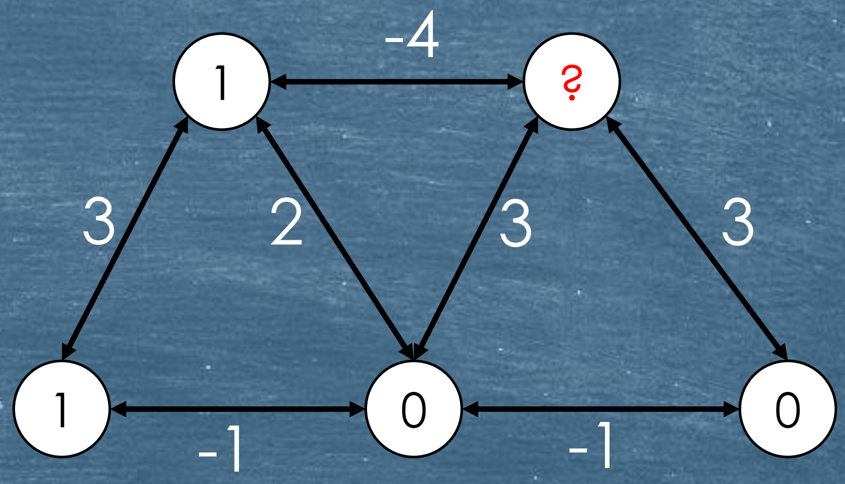
-E=goodness=3

Example taken from http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec11.pdf

Energy of a Hopfield Net

Finding the Energy minimum:

- Start from a random state
- Update units in random order, one at a time, to the state that gives the lowest global energy
- Stop when there is no unit that changes its output



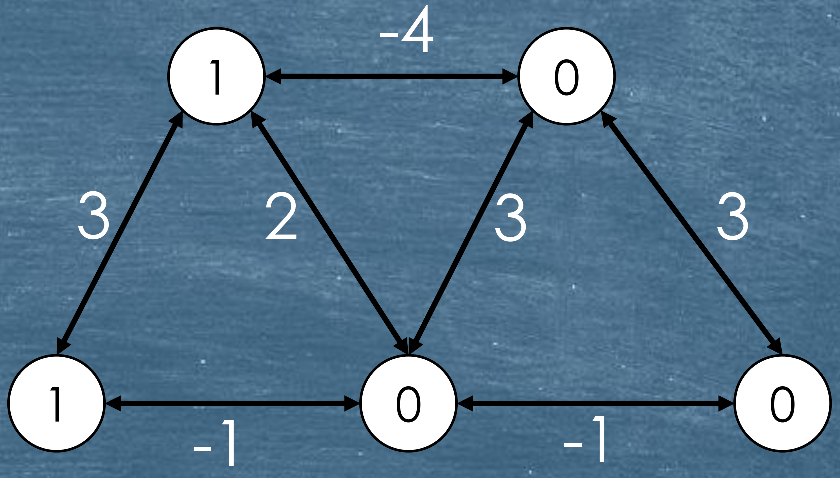
-E=goodness=3

Example taken from http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec11.pdf

Energy of a Hopfield Net

Finding the Energy minimum:

- Start from a random state
- Update units in random order, one at a time, to the state that gives the lowest global energy
- Stop when there is no unit that changes its output



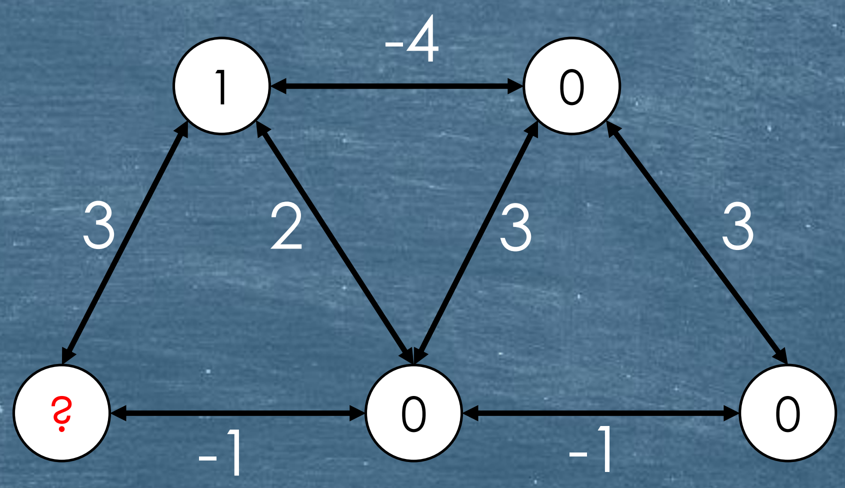
-E=goodness=3

Example taken from http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec11.pdf

Energy of a Hopfield Net

Finding the Energy minimum:

- Start from a random state
- Update units in random order, one at a time, to the state that gives the lowest global energy
- Stop when there is no unit that changes its output



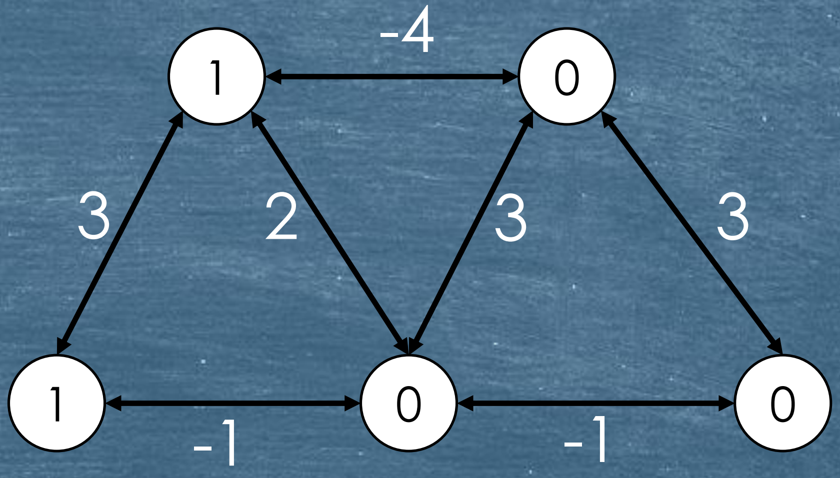
-E=goodness=3

Example taken from http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec11.pdf

Energy of a Hopfield Net

Finding the Energy minimum:

- Start from a random state
- Update units in random order, one at a time, to the state that gives the lowest global energy
- Stop when there is no unit that changes its output



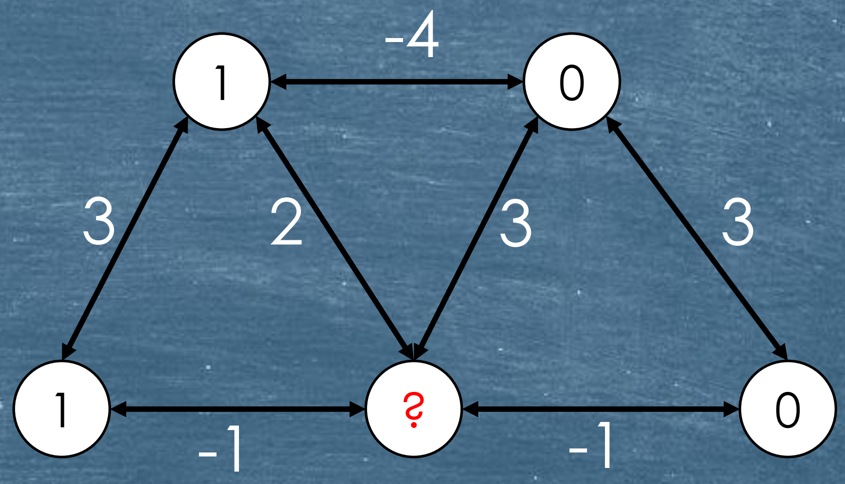
-E=goodness=3

Example taken from http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec11.pdf

Energy of a Hopfield Net

Finding the Energy minimum:

- Start from a random state
- Update units in random order, one at a time, to the state that gives the lowest global energy
- Stop when there is no unit that changes its output



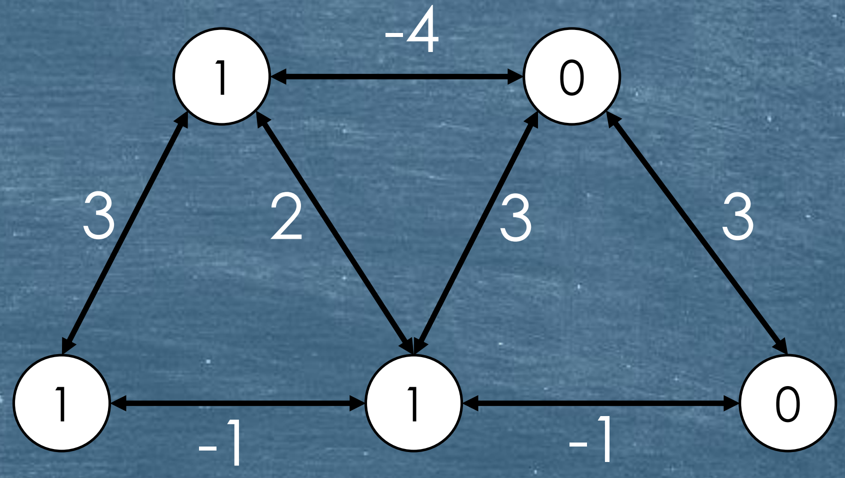
-E=goodness=3

Example taken from http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec11.pdf

Energy of a Hopfield Net

Finding the Energy minimum:

- Start from a random state
- Update units in random order, one at a time, to the state that gives the lowest global energy
- Stop when there is no unit that changes its output



- This is a local minimum. Any unit that we probe, will not change it's activation

-E=goodness=4

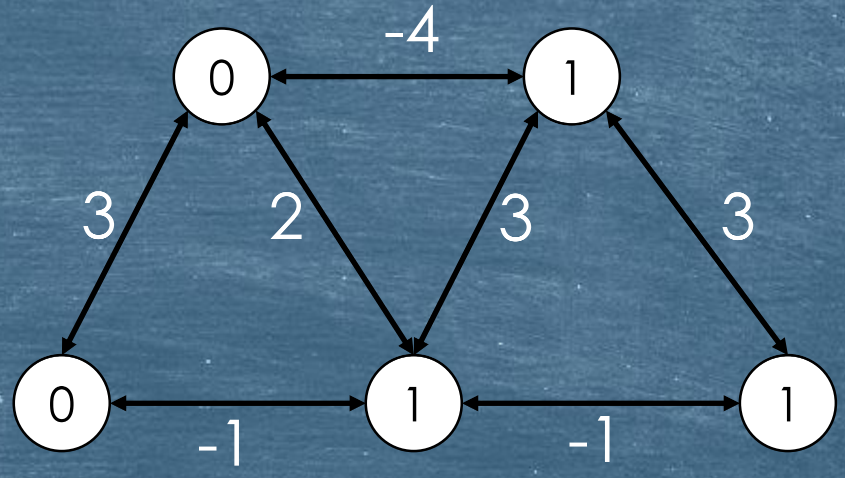
Example taken from http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec11.pdf

Energy of a Hopfield Net

But that is not the global minimum

- The global minimum is given on the right.
- However, to reach this minimum requires to change the activation of two units at the same time

So, why not simultaneous change activations of all units, depending on the other?



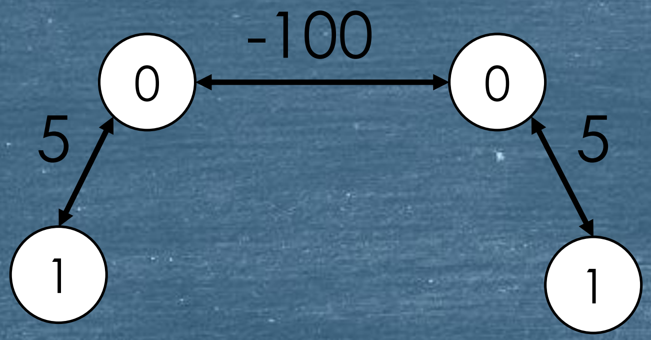
-E=goodness=5

Example taken from http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec11.pdf

Energy of a Hopfield Net

Why not simultaneous change activations?

1. You can. It is called synchronous updating. However, is has some problems:
   - doing this could make the energy go up.
   - You can get oscillations



   - I.e.: Simultaneous updating the top units will make the things worse

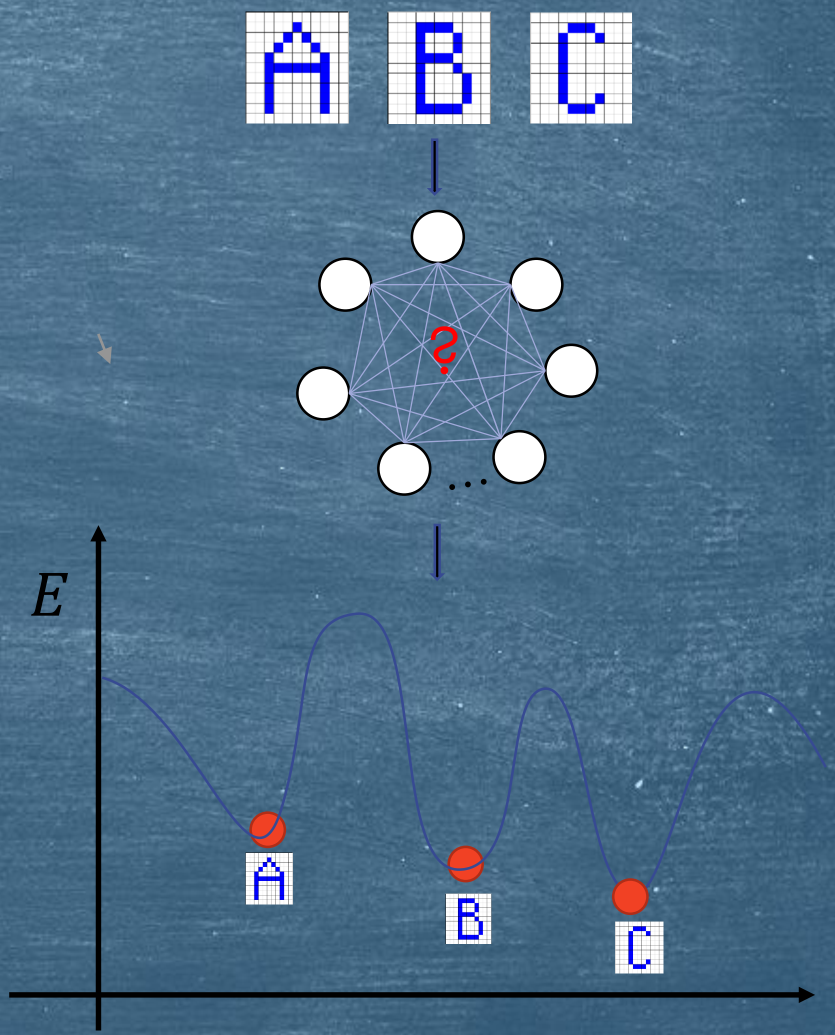
2. So sequential (asynchronous updating) is the one that gives the best results

Example taken from http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec11.pdf

Training a Hopfield Net

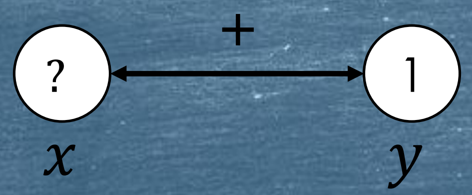
Training a Hopfield Net

- In the previous slides, the weights were already set and we changed the activation until we reached at a minimal energy
- The question is: How do we set the weights, given an input, such that the minimal energy corresponds to the network units having the same activations as the input
- Get the right weights such that a local minimum to correspond to a memory
- We will consider in the following slides that the values outputted by the neurons, through the activations, are 1 and -1



Training a Hopfield Net

- Consider the case of two neurons . (activation is computed using: $s_i = \sum_j w_{ij} s_j$)



- If the weight is positive, then the contribution of unit x to unit y is given by the sign of activation of unit x: unit x attracts unit y to have same activation
- If the weight is negative, then the contribution of unit x to unit y will have the opposite sign of activation of unit x: unit x repels unit y
- So, units that fire together, link together.

Training a Hopfield Net

- So, in order for the network to be in equilibrium, the sign of the weights must depend on the activation value of the units

- The same observation can be obtaine by looking at the formula for the energy (no biases)

$$E = -\sum_{i<j} s_i s_j w_{ij}$$

- The network has the lowest energy (given a pattern s_i^p) when:
  - Units with same activation are linked with a positive weight
  - Units with different activation are linked with a negative weight

Training a Hopfield Net

- Given these observations, the weights will be adjusted in the following way:

$$\Delta w_{ij} = s_i s_j$$
$$w_{ij} = w_{ij} + \Delta w_{ij}$$

- If neurons have activation of (0,1), then equation changes to

$$\Delta w_{ij} = (s_i - \frac{1}{2})(s_j - \frac{1}{2})$$

- Given more patterns, the rule for setting the weight is:

$$w_{ij} = \frac{1}{N} \sum_{n=1}^{N} x_i^n x_j^n$$

This is called Hebbian learning

Training a Hopfield Net

- So the rule through which we change a weight is :

$$\Delta \mathrm{w_{ij}} = \mathrm{s_i s_j}$$
$$w_{ij} = w_{ij} + \Delta w_{ij}$$

This is called Hebbian learning and is plausible from a biologic point of view, since it is:

- Local
- Incremental
- Instant

Training a Hopfield Net

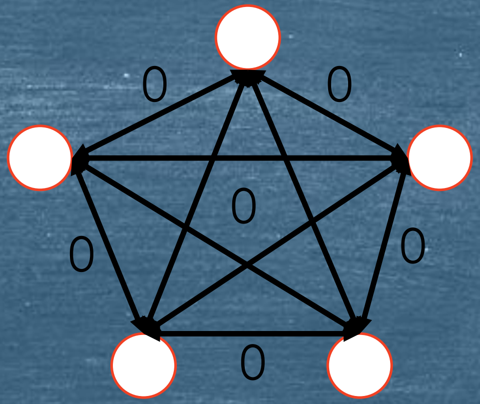
- Example:

Train a Hopfield network to remember the following patterns $V^1 = (-1,1,1,-1,1)$ and $V^2 = (1,-1,1,-1,1)$

So we need to train a network with 5 inputs, where each node is connected with each other, but not itself.

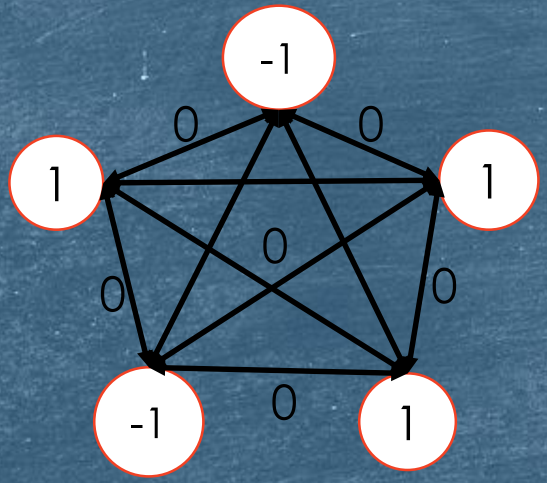
The weights will be represented as symetrical matrix with the elements on the diagonal set to 0

All the weights will be 0 in the beginning



Training a Hopfield Net

- The first sample is presented to the network. $V^1 = (-1,1,1,-1,1)$
- Training for each sample is done separatly (on a separate matrix)



	-1	1	1	-1	1
-1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
-1	0	0	0	0	0
1	0	0	0	0	0

Training a Hopfield Net

- Each cell in the matrix, will be set to -1 or 1: the result of multiplying every two activations.
- Note: this is equivalent to computing $s * s^T$
- The elements on the main diagonal will be set to 0 (since there is no connection from a unit to itself)
- The result will be added (summation) to the global weight matrix

	-1	1	1	-1	1
-1	0	-1	-1	1	-1
1	-1	0	1	-1	1
1	-1	1	0	-1	1
-1	1	-1	-1	0	-1
1	-1	1	1	-1	0

0	-1	-1	1	-1
-1	0	1	-1	1
-1	1	0	-1	1
1	-1	-1	0	-1
-1	1	1	-1	0

Training a Hopfield Net

- The same thing happens with the second pattern: $V^2 = (1, -1, 1, -1, 1)$
- The result will be added (summation) to the global weight matrix
- The values on the weight matrix will be in the interval [-M,M] where M represents the number of patterns

	1	-1	1	-1	1
1	0	-1	1	-1	1
-1	-1	0	-1	1	-1
1	1	-1	0	-1	1
-1	-1	1	-1	0	-1
1	1	-1	1	-1	0

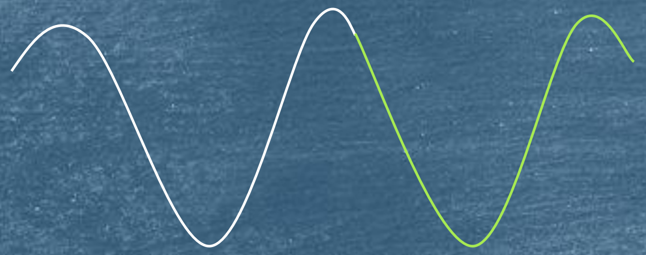
0	-2	0	0	0
-2	0	0	0	0
0	0	0	-2	2
0	0	-2	0	-2
0	0	2	-2	0

Hopfield problem: Spurious minima

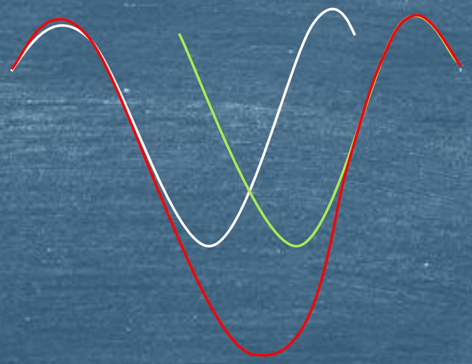
Hopfield problem: Spurious minima

- Spurious minima represents the situation where the network settles to a local minimum that does not represents a learned pattern

- This happens when to patterns, represented by two nearby minima configurations, merge together to create a new minima that has an even lower energy



Trained patterns



Trained patterns with spurious minima

Hopfield problem: Spurious minima

- Solution for spurious minima:
- An interesting solution that was proposed by Hopfield et. all , was to use unlearning:
  - Start from an initial random state
  - Let the network reach a minimum (which is probable to be a spurious pattern)
  - Do the opposite of learning.
    - $\Delta w_{ij} = -s_i s_j$

The authors showed that this worked, but had no analysis. For example, how much unlearning should we do?

Other authors that have tested this idea found that the network ends up working erratically (Cristos 1994)

Hopfield problem: Spurious minima

- Francis Crick and Mitchison, proposed that human brains also do unlearning in order to better store memories

- They proposed that unlearning is what is going on in the brain during REM sleep
  - During the day you do lot of learning and you get spurious minima.
  - At night, you put the network to a random state (dreaming), you settle to a local minima, and then you unlearn (forget your dream).
  - This solves a puzzle: why do we bother dreaming (the brain patterns looks similar during the day as in REM sleep) if we forget what we've dreamt.

Network Capacity

Network Capacity

- The network capacity determines how many memories we can stored and retrieved. Behind this limit spurious states are very probable.
- Hopfield (and many other authors) have showed, that when Hebbian learning is used, the capacity of the Network (using random patterns) is just
  - 0.15N memories (0.138N more exactly)
- Other authors (McEliece *et al.* (1987)) have shown that the capacity can not exceed $\frac{n}{4\ln(n)}$, or $\frac{n}{2\,ln(n)}$ if some errors are tolerable

Network Capacity

- Many authors have tried to increase this limit
- Amos Storkey (1997) modified the Hebbian learnig rule, and achieved a greater capacity:

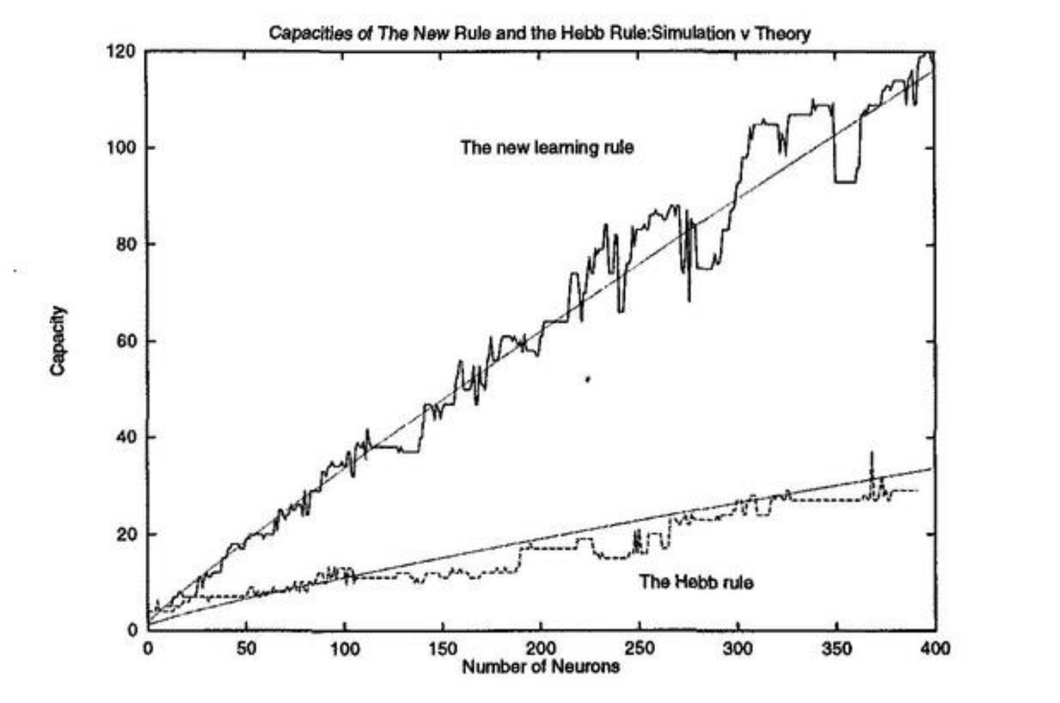
$$w_{ij}^{v} = w_{ij}^{v-1} + \frac{1}{n} s_i^v s_j^v - \frac{1}{n} s_i^v h_{ji}^v - \frac{1}{n} h_{ij}^v s_j^v$$

where

$$h_{ij}^{v} = \sum_{k=1, k!=i,j}^{n} w_{ik}^{v-1} s_k^v$$

Network Capacity

- Amos Storkey (1997)proved that his equation can boost the capacity of the network to $\frac{n}{2\sqrt{\ln(n)}}$
- At 1.000.000 neurons, the Storkey Method stores 5 times more patterns
- Another advantage of Storkey rule is that it has properties that make it biologically plausible:
    - Is incremental
    - Is local



Network Capacity

- Another interesting method was proposed by Elizabeth Gardiner in 1986.
- The idea is to train the network such that every pattern is stable.
  - Initialize the neurons with a pattern
  - For each unit:
    - If it has the right value (given by the pattern) then leave it alone
    - If it doesn't, modify the weight coming to it using the perceptron rule
  - Do this for all the pattern in several iterations
- Obviously, this loses the properties (it is no longer iterative) that would have make it biologically realistic.
- If there is a way to adjust weights (if training converges), than this method provides the highest capacity

Conclusions

- Hopfield network is a way through which you can memorize input patterns and you can obtain it in an associative manner
- It is governed by an Energy function which is modified by every neuron, based on a local operation
- It offers an explaination of how memories are being stored by the human brain
- Its main problem is the sporious minima
- It represents the basis for more complex networks (i.e. Boltzmann Machines)

Questions & Discussion

References

- https://www.doc.ic.ac.uk/~sd4215/hopfield.html
- http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec11.pdf
- https://en.wikipedia.org/wiki/Hopfield_network
- http://homepages.inf.ed.ac.uk/amos/publications/Storkey1997IncreasingtheCapacityoftheHopfieldNetworkwithoutSacrificingFunctionality.pdf
- http://web.cs.ucla.edu/~rosen/161/notes/hopfield.html
- http://neupy.com/2015/09/20/discrete_hopfield_network.html
- http://www.comp.leeds.ac.uk/ai23/reading/Hopfield.pdf
- http://perso.ens-lyon.fr/eric.thierry/Graphes2010/alice-julien-laferriere.pdf
- Humayun Karim Sulehria, Ye Zhang, Study on the Capacity of Hopfield Neural Networks
- Amos Storkey, Increasing the capacity of a Hopfield network without sacrificing functionality, http://homepages.inf.ed.ac.uk/amos/publications/Storkey1997IncreasingtheCapacityoftheHopfieldNetworkwithoutSacrificingFunctionality.pdf