COMPUTATIONAL INTELLIGENCE

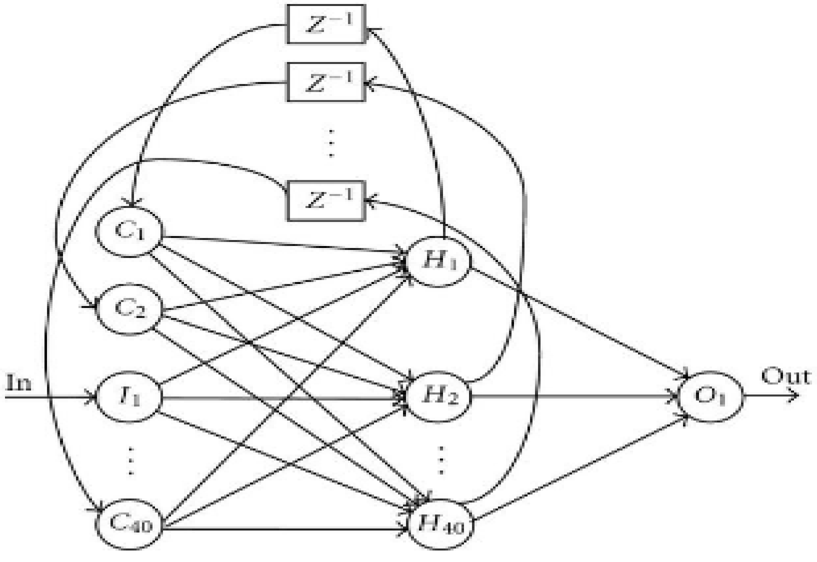
Recurrent Dynamic Hopfield Neural Networks

AGH

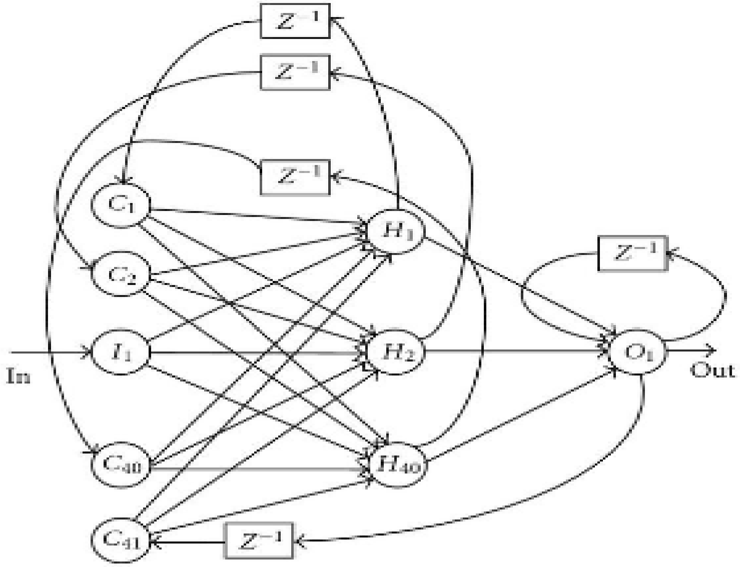
Adrian Horzyk

Recurrent Neural Networks

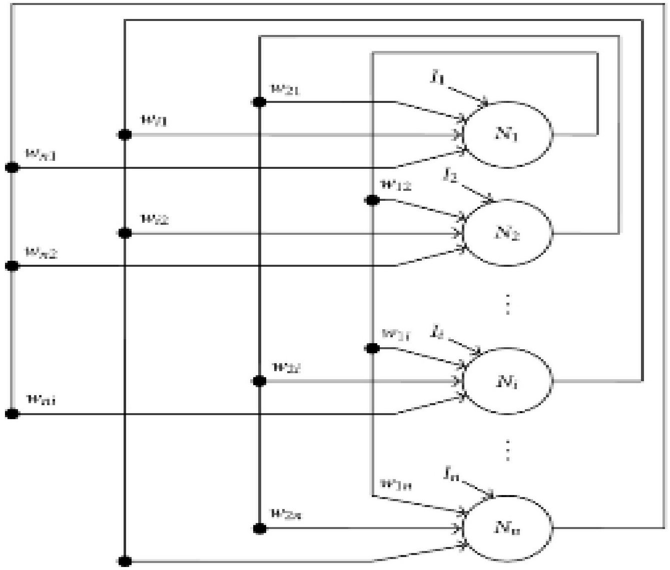
Recurrent Neural Networks (RNN) are a class of dynamic artificial neural networks where connections between units form directed cycles. This enables the network to be in one of many states and allows it to exhibit dynamic temporal behavior. Such networks can continuously process inputs until they find an attractor which makes the process to get stuck in it. The attractor can consists of a single state or a boundary cycle of states between which it jumps to infinity. There are many different recurrent neural networks: Hopfield Networks, Hamming Networks, Elman Networks, Jordan Networks, RTRN, BAM, MAM etc. A few o them can be used as associative memories.



Elman Neural Network



Jordan Neural Network



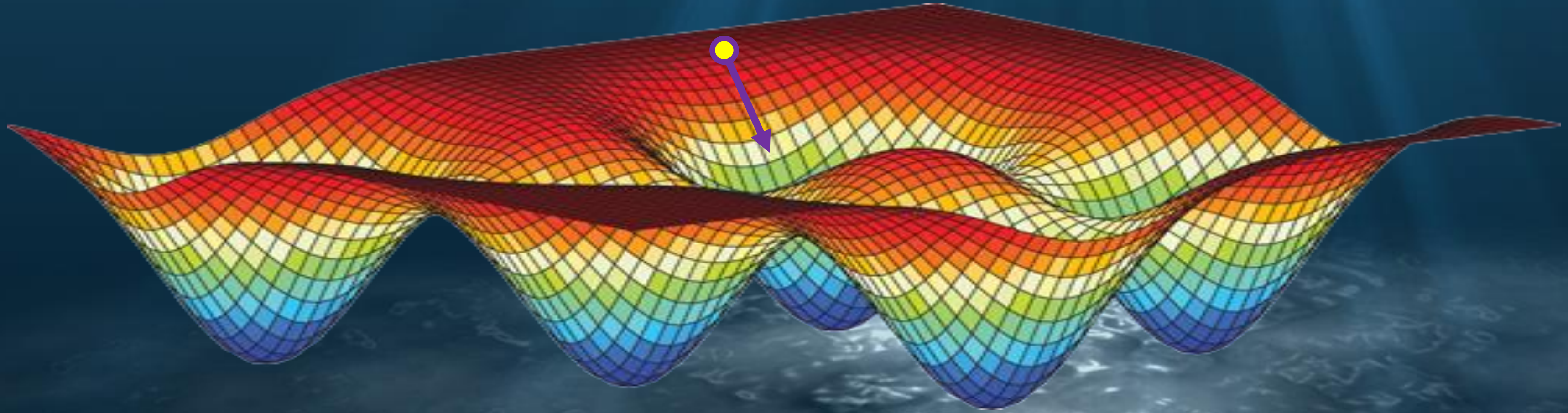
Hopfield Neural Network

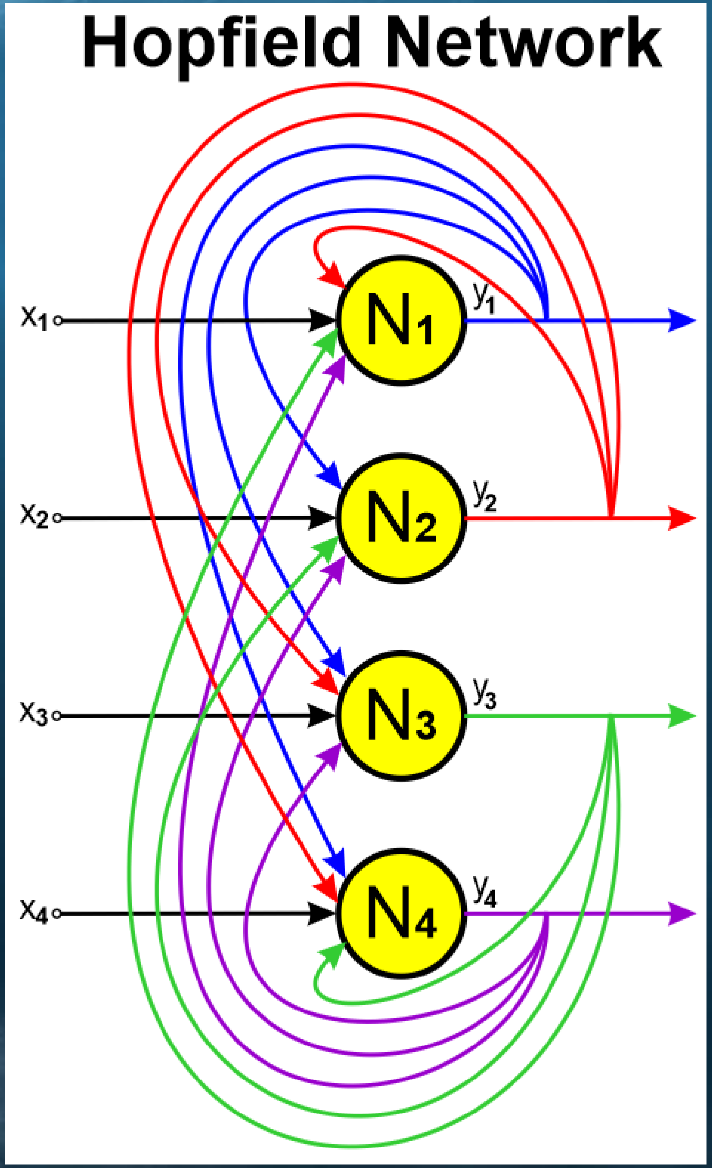
Hopfield Neural Networks

Hopfield Neural Networks is a form of recurrent neural networks. It consists of N binary threshold (hard switch) neurons which are placed in a single layer. The neurons typically take on output values -1 and 1, however they could also take on output values 0 and 1.

The output activation functions are usually modeled using sgn(…).

Each Hopfield NN always converges to a local minimum (attractor) but some of these local minima are „wrong" and represent false patterns which were not been trained, rather than to one of the trained and stored patterns which also have their local minima.





Hopfield Neural Networks

Each neuron in the Hopfield Network has:

- an input connection that provides the input data,
- an output connection that supplies the output data when the network converges to one of the attraction states, and
- N-1 connections to all the other neurons with the exception to itself.

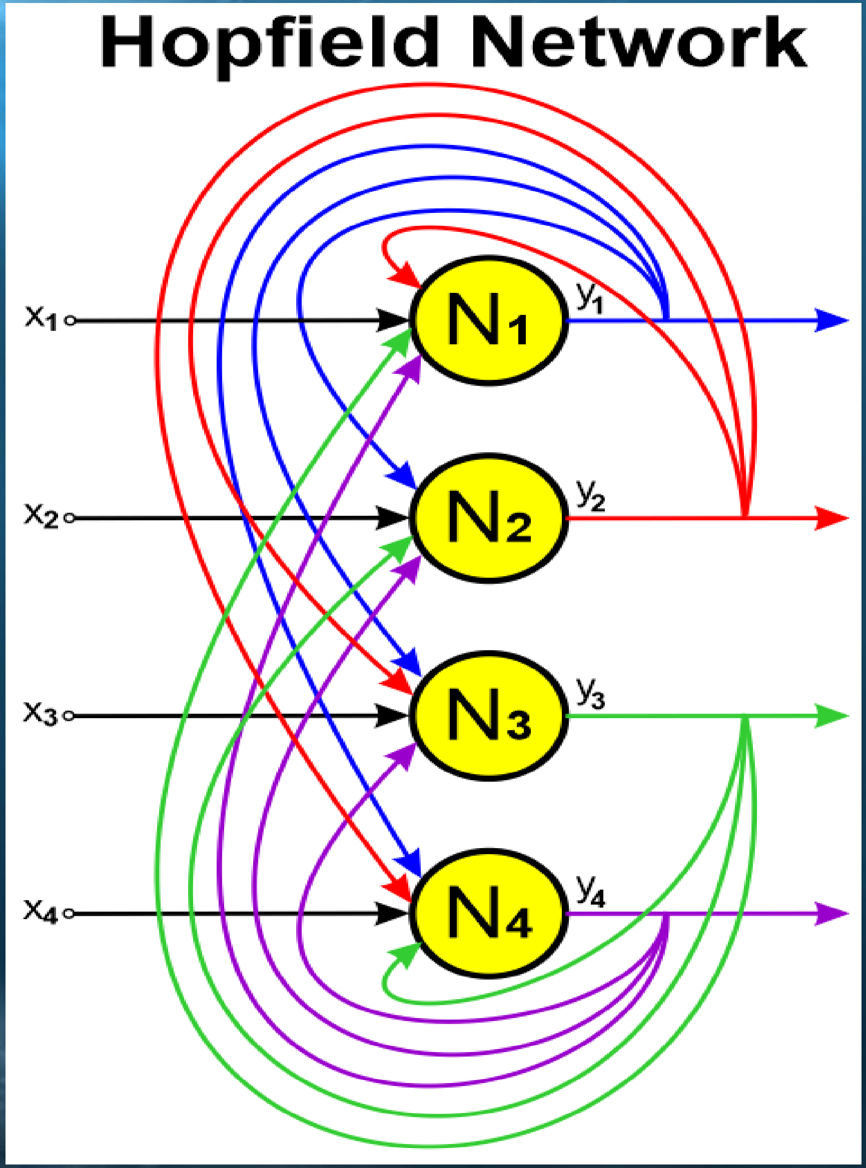
Each neuron has a threshold θ_i which is usually implemented in a form of a bias signal which stimulates each neuron with -1 or +1 input value. Neurons are stimulated both by the external input stimuli and feedback internal stimuli coming from the other neurons in the network.

Consequently, the state of each neuron in t-th period is computed after:

$$y_i^t(t) = sgn\left(x_i + \boldsymbol{w}_i^T \boldsymbol{y}^{t-1} - \theta_i\right)$$

Where x_i is an input signal, θ_i is the threshold of the i-th neuron, $\boldsymbol{w_i}$ is the weight vector (in which $w_{ii} = 0$, $w_{ij} = w_{ji}$), and $\boldsymbol{y}$ is the output vector computed in the previous phase *t-1*.

All weights are symmetric!



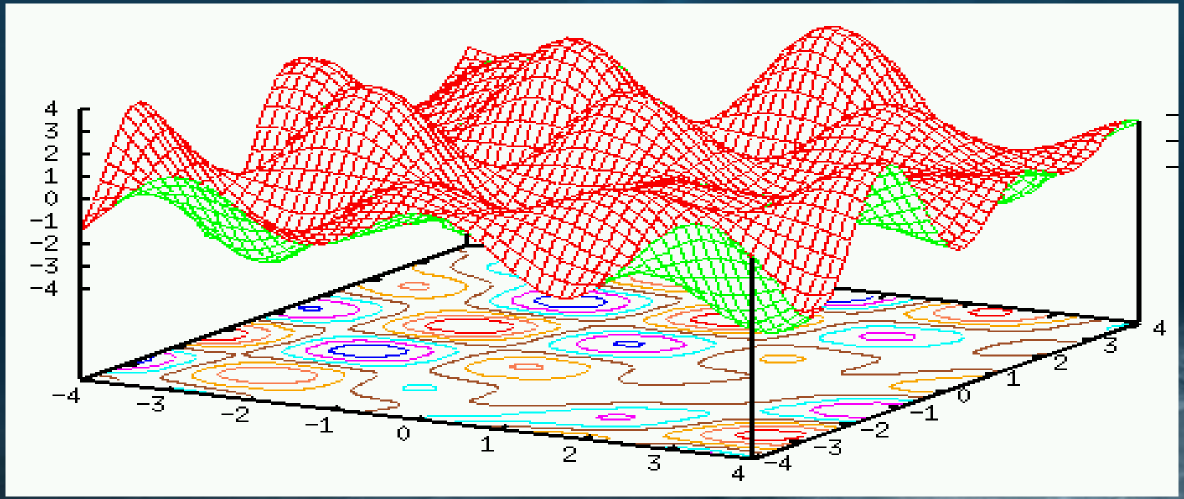
Hopfield Neural Networks

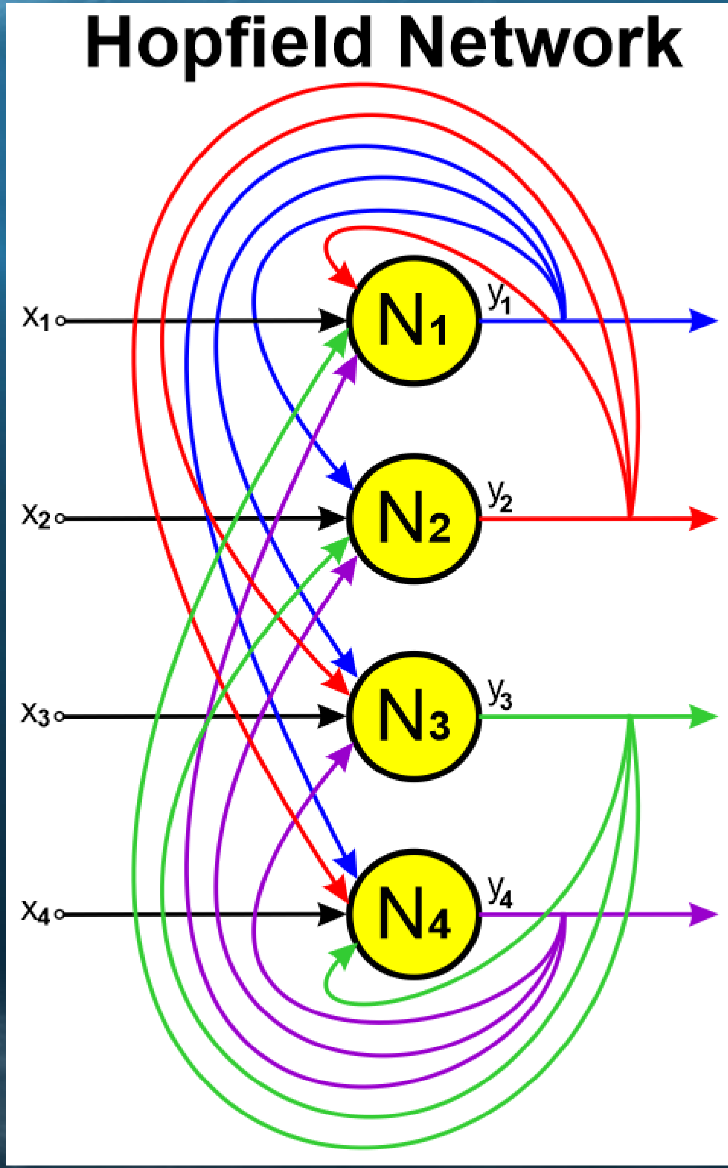
Neurons in the Hopfield Network can work:

- synchronously (all neurons are updated in the same simulation time),
- asynchronously with a random choice of an updated neuron.

The recurrent computational process starts with the external input stimuli.
This process typically finishes in one of the attractors when $y_i^t(t) = y_i^t(t-1)$.
The number of all attractors (including false ones) is usually not known.

The number of false attractors grows with the number of trained patterns. Each attractor represents its attraction area, which are illustrated under the chart of the red 3D function:





Training of Hopfield Neural Networks

Hopfield Neural Networks can be trained using the generalize Hebb rule:

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

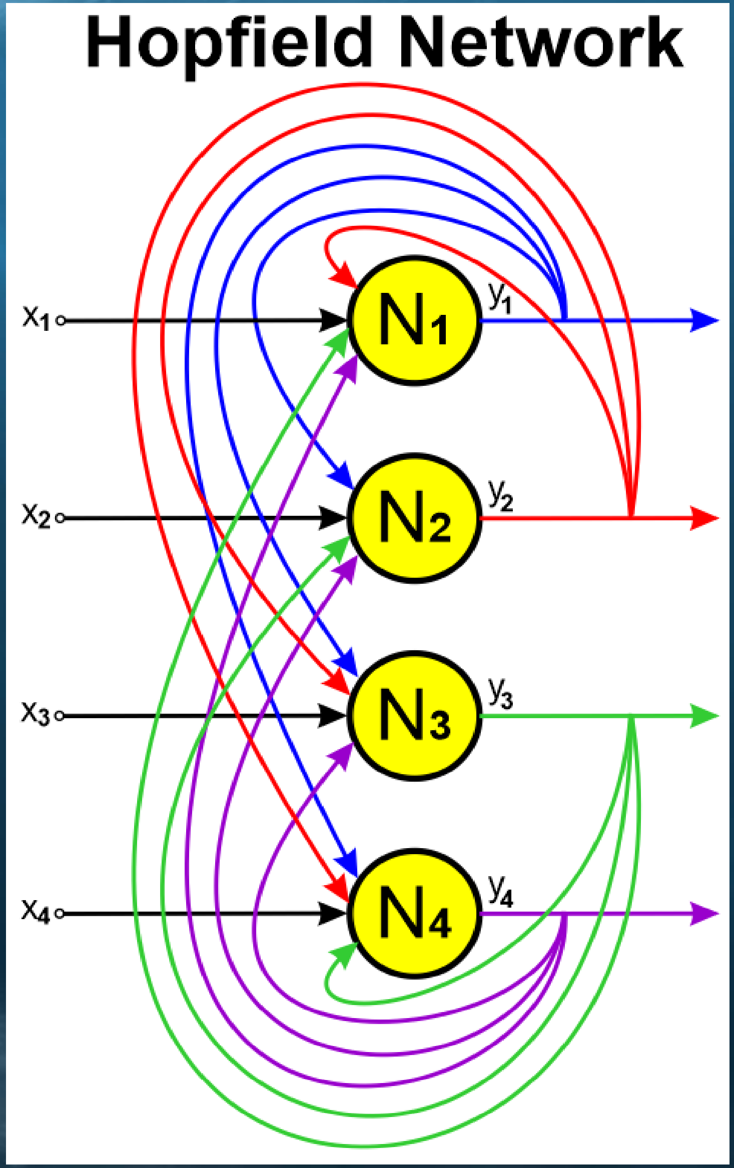
but the number of patterns that can be trained in such a way is only 13.8% of the number of all neurons, so it is better to use the pseudoinverse method which is based on solution of an equation system.

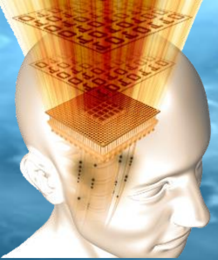
When **X** is a matrix of training vectors $\boldsymbol{X} = [\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^M]$
and **W** is the weight matrix then the solution can be achieved by
the computation of weight parameters to satify the equation: $\boldsymbol{WX} = \boldsymbol{X}$.

It means that we have to solve the system of equations when training vectors are independent:

$$\boldsymbol{W} = \boldsymbol{X}\boldsymbol{X}^+ = \boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T$$

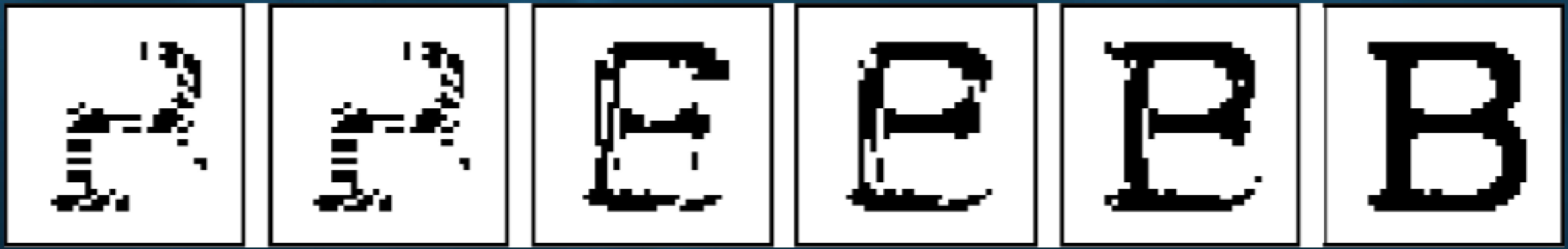
Where $\boldsymbol{X}^+$ is the pseudoinverse matrix of $\boldsymbol{X}$.





Recovering of training patterns

Hopfield networks can successfully recover biased training patterns or to make up their missing parts, but their capacity can be far away from your expectations:



Bibliography and References

http://home.agh.edu.pl/~horzyk/lectures/ahdydci.php

ACADEMIC WEBSITE - ADRIAN HORZYK, PhD, Prof.

AGH University of Science and Technology in Cracow, Poland
Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering
Department of Automatics and Biomedical Engineering, Field of Biocybernetics

Dossier | Research | Publications | Courses | Graduates | Consultations | Contact

AGH

LECTURES

- Introduction to Artificial, Cognitive and Computational Intelligence
- Introduction to Neural Networks
- Models of Neurons
- Construction and Learning Strategies
- Overview of Neural Network Models
- Multilayer Perceptron MLP
- Radial Basis Function Networks RBF
- Support Vector Machine SVM
- Reinforcement Learning
- Deep Learning DL
- Unsupervised Training and Networks
- Self Organizing Maps SOM
- Recurrent Neural Networks
- Associative and Semantic Memories
- Associative Neural Systems

COMPUTATIONAL INTELLIGENCE

is already under construction and will be available at the spring semester in 2016/2017.

This course will include 28 lectures, 14 laboratory classes and 14 project classes.

What is this course about?

This course is intended to give students a broad overview and deep knowledge about popular solutions and efficient neural network models as well as to learn how to construct and train intelligent learning systems in order to use them in everyday life and work. During the course we will deal with the popular and most efficient models and methods of neural networks, fuzzy systems and other learning systems that enable us to find specific highly generalizing models solving difficult tasks. We will also tackle with various CI and AI problems and work with various data and try to model their structures in such a way to optimize operations on them throughout making data available without necessity to search for them. This is a unique feature of associative structures and systems. These models and methods will allow us to form and represent knowledge in a modern and very efficient way which will enable us to mine it and automatically draw conclusions. You will be also able to understand solutions associated with various tasks of motivated learning and cognitive intelligence.