An Introduction to Neural Networks

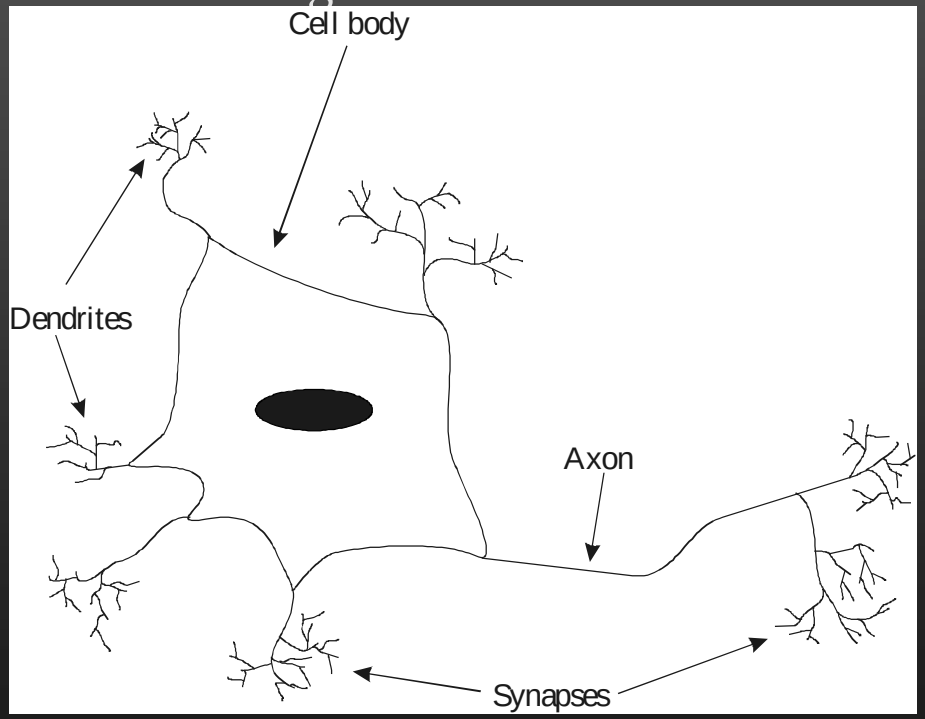
Reference: B.J.A. Kröse and P.P. van der Smagt (1994): An Introduction to Neural Networks, Poglavja 1-5, 6.1, 6.2, 7-8.

- B.J.A. Kröse and P.P. van der Smagt (1994): An Introduction to Neural Networks, Seventh edition, The University of Amsterdam. http://staff.science.uva.nl/~krose/publications/index_en.html
- ! ! J. Sjöberg et al. (1995):Non-linear Black-Box Modeling in System Identification: a Unified Overview, Automatica, Vol. 31, 12, 1691-1724. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.27.81
- M. Agarwal(1997): A systematic classification of neural-network-based control, IEEE Control Systems Magazine, Vol. 17, No. 2, 75-93.
- M. Nørgaard (1995): Neural Network Based System Identification Toolbox User's Guide, Technical Report 95-E-773, Institute of Automation, Technical University of Denmark,Lyngby. http://www.iau.dtu.dk/research/control/nnsysid.html
- M. Nørgaard (1995):Neural Network Based Control System Design Toolkit User's Guide, Technical Report 96-E-830, Institute of Automation, Technical University of Denmark,Lyngby. http://www.iau.dtu.dk/research/control/nnctrl.html

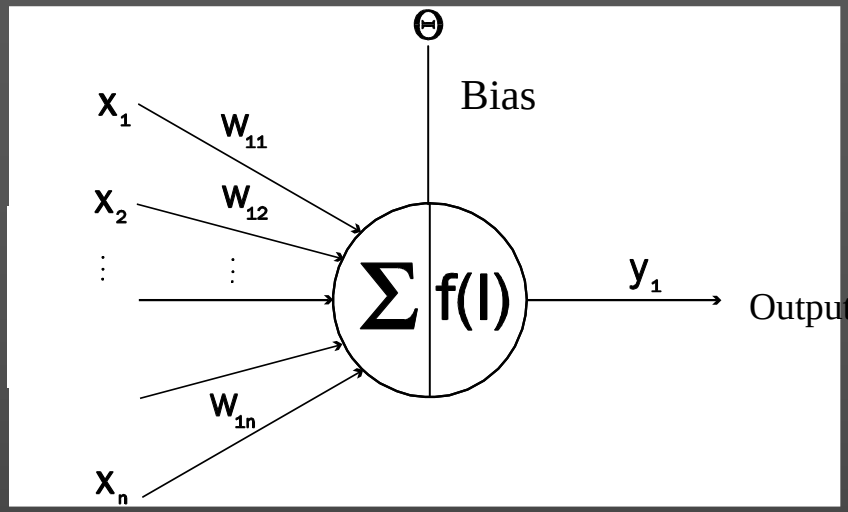
Introduction: historical overview

Milestones

- **1943 – dawn of Artificial Neural Networks (ANN)**
  McCulloch in Pitts – formal representation of neuron, mathematical abstraction – a model - of biological neuron



A biological neuron

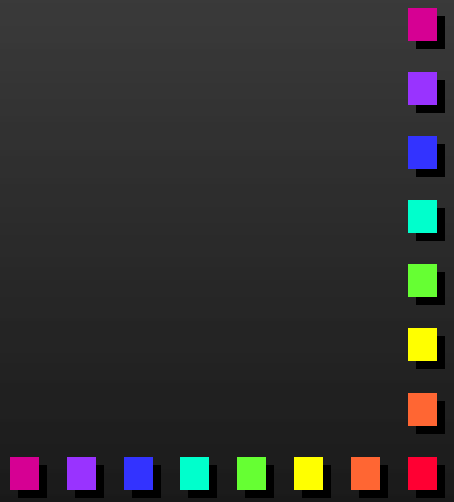


$$y_i(t) = f\left(\sum_j w_{ij}(t)x_j(t) + \theta_i(t)\right)$$

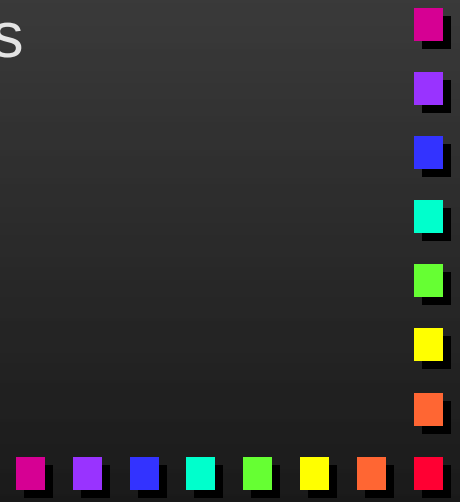
The neuron of McCulloch and Pitts

- Treshold activation function: elements of logic functions
- Conections between units into networks
- Formal model of nevron – unchanged until present days

- **1949 - Hebb:** psychologist, the first learning rule
- **1959 - Rosenblatt:** the first who used "*perceptron*" for single layer network
- **1960 - Widrow, Hoff:** *adaline* (ADAptive LINear Element) – single layer,
  the first ***analiticaly*** derived learning rule (least squares),
  until then only heuristical learning, in analogy with BIO systems
- **1963 - Widrow, Smith:** inverted pendulum, adaline



- Perceptron (1959)
  - Activation function: switching function
  - Single layer network
  - Can not represent XOR function
- Adaline (Adaptive linear element - 1960)
  - Activation function: linear function
  - Learning with “delta rule” (least squares optimisation method)
  - linear models



Explanations

- Regression
- Classification

- Learning: optimising weights of ANN

Learning methods

- Hebbs rule

$$\Delta w_{ij} = \gamma y_i x_j$$

- Delta rule (Widrow-Hoff rule)

$$\Delta w_{ij} = \gamma x (d_i - y_i) x_j$$

  - Least squares optimisation method

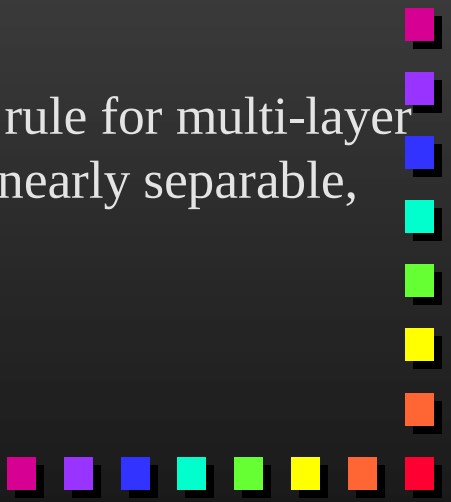
- 1949 - Hebb: psychologist, the first learning rule
- 1959 - Rosenblatt: the first who used "*perceptron*" for single layer network
- 1960 - Widrow, Hoff: *adaline* (ADAptive LINear Element) – single layer, the first *analiticaly* derived learning rule (least squares), until then only heuristical learning, in analogy with BIO systems
- 1963 - Widrow, Smith: inverted pendulum, adaline
- **1969 -** Minsky, Papert show the limitations of perceptrona, that can not be used for classification of elements that are not linearly separable -XOR function can not be represented

  - nothing much happens until app. 1982

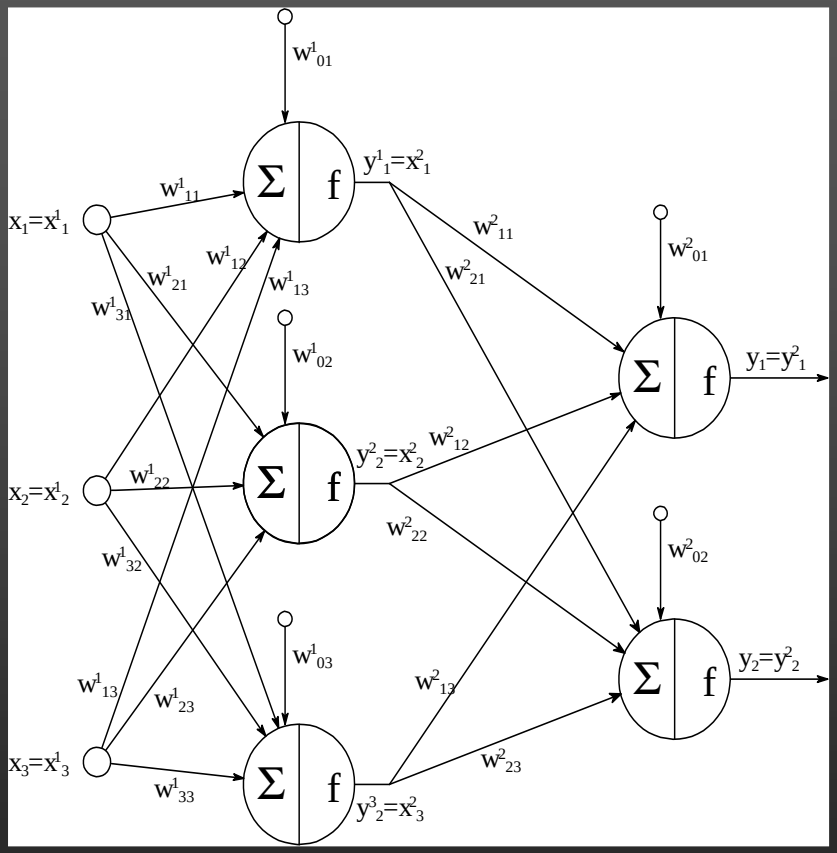
  The end of single layer networks period!

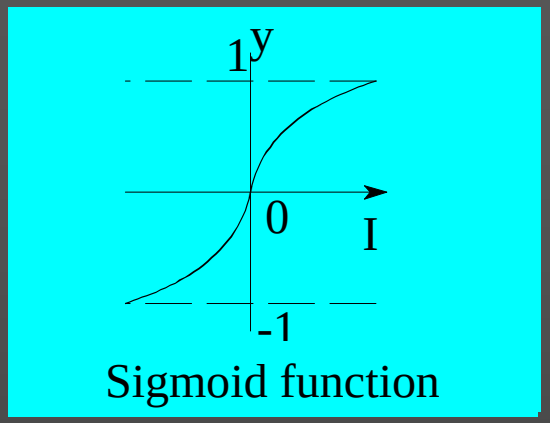
- **1986 - Rumelhart and co.:** "*backpropagation*" – learning rule for multi-layer perceptron – classification of elements that are not linearly separable, the nonlinear mapping!!!

  The renaissance of ANN



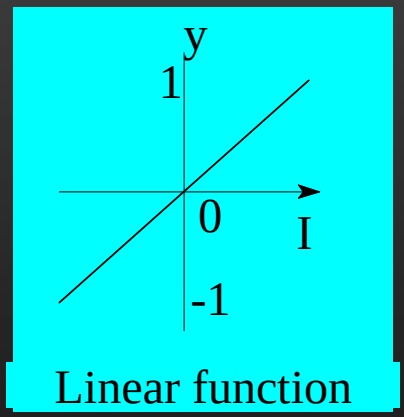
Multi-layer perceptron





Sigmoid function

$$f(I) = \frac{1}{1+e^{-aI}}$$

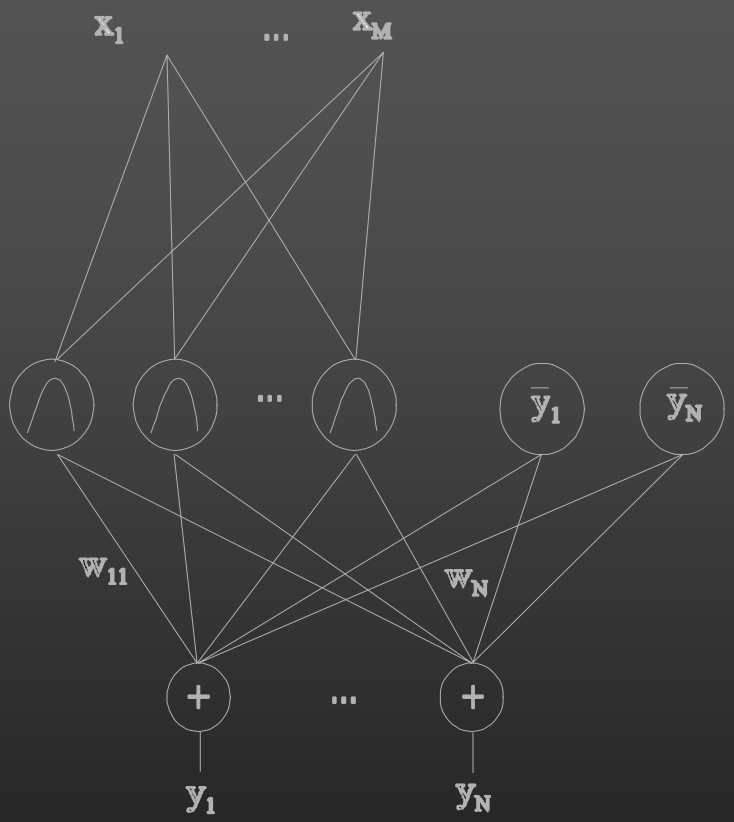


Linear function

$$y_i = f_i\left(\sum_j w_{ij}\, f_j\left(\sum_k w_{jk} x_k + w^1_{0k}\right) + w^2_{0j}\right)$$

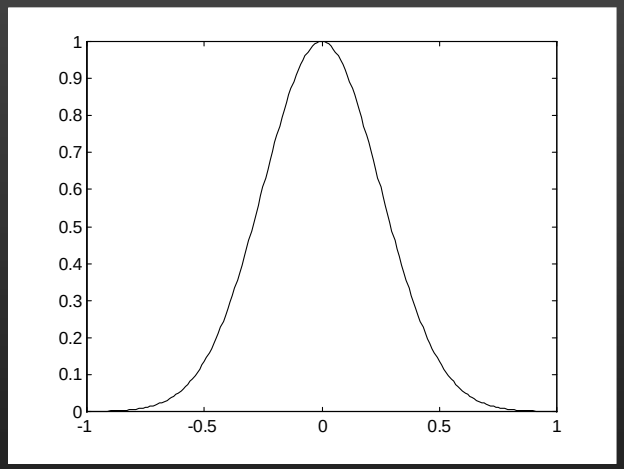
Radial basis function (RBF) network

- all weights between inputs and hidden layer ≡ 1

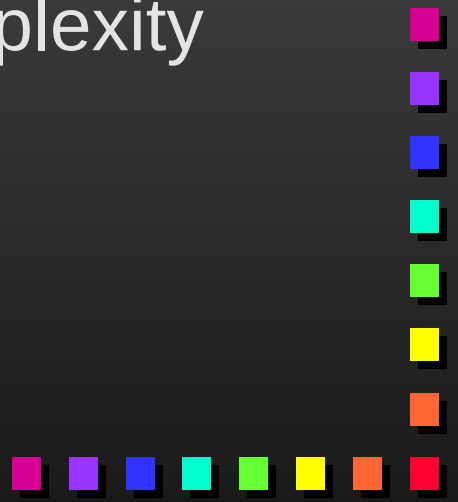


$$y_j = \sum_{i=1}^{N} w_{ij} z_i(\mathbf{x}, c_i)$$

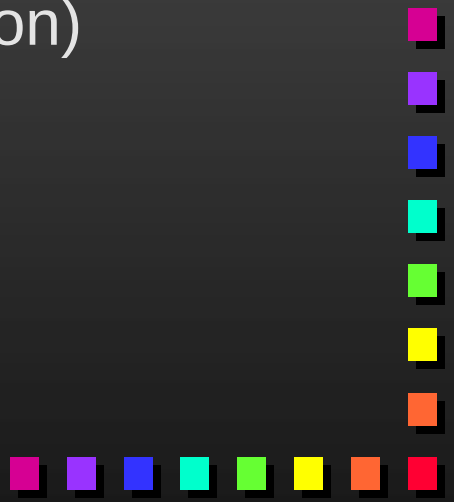
$$z_i = \exp\left[-\frac{1}{2}\left(\frac{(x_1 - c_{i1})^2}{\sigma_{i1}^2} + \frac{(x_2 - c_{i2})^2}{\sigma_{i2}^2} + \ldots + \frac{(x_n - c_{in})^2}{\sigma_{in}^2}\right)\right]$$



- Back-propagation method (BP)
- multi-layer feed-forward networks
- delta rule generalised for nonlinear problems (gradient optimisation method)
- BP algorithm is used for the calculation of cost function gradienta
- Applicable for all sorts of nonlinear systems, but differences in computation complexity
- improvements:
  - Learning rate with momentum
  - Learning per pattern

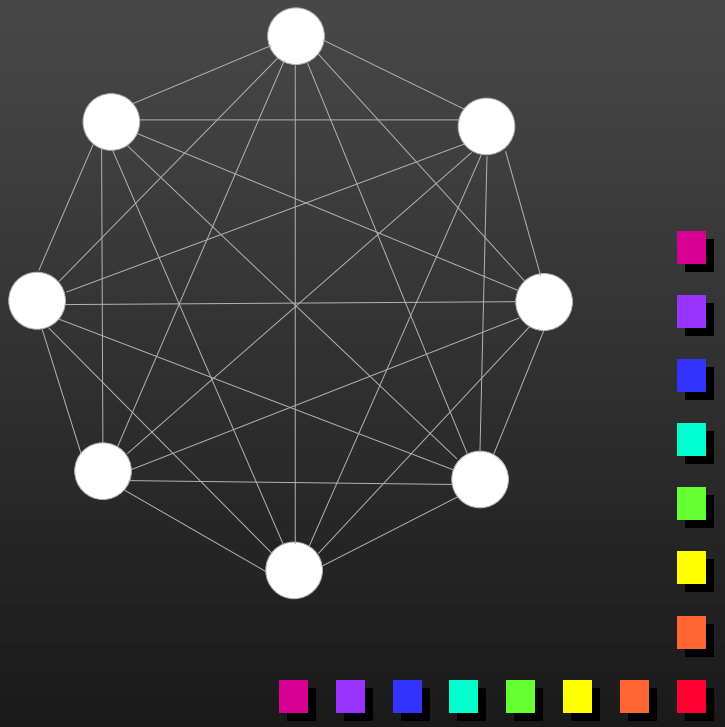


- ANN topology
  - Feed-forward networks
  - Recurrent networks
- ANN learning
  - Supervised (associative) learning
  - Unsupervised learning (self-organisation)
- ANN application
  - Regression
  - Classification



Recurrent networks

- The main difference from feed-forward networks is that they contain feedback connections
- Mainly self-organised networks
- Rekursive neural networks
- Hopfield neural network
- Boltzmann machines

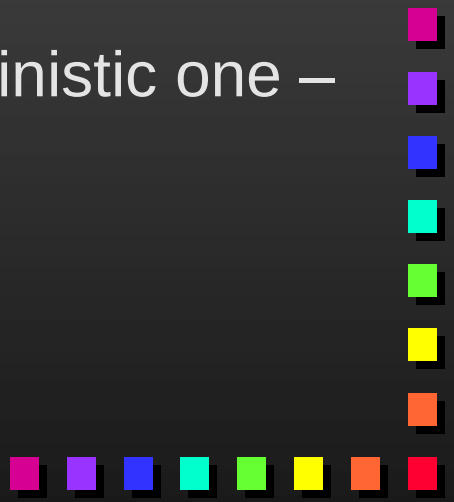


Hopfield neural network

- Signum activation function or linear function with saturation
- Feed-back connections
- Hopfield network as associative memory
- Theoreticaly intriguing,but less interesting for practice

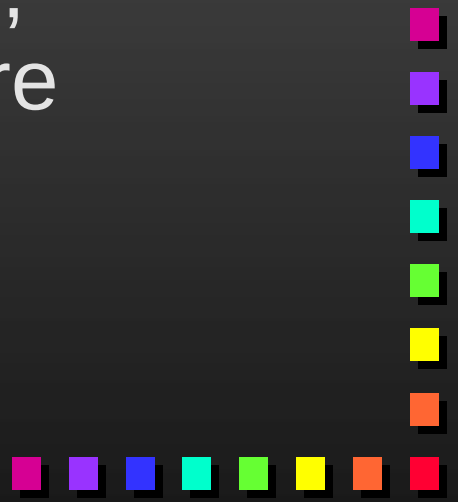
Boltzmann machines

- Hopfield network with hidden layers
- Stochastic update rule instead of deterministic one – principle of annealing



Self-organised neural networks

- Instead of input/output data pairs as in supervised learning, self-organised NN use only input data
- Tipically used for classification: clustering, vector quantification, dimensionality reduction, feature extraction (pattern recognition)
- Kohonen network

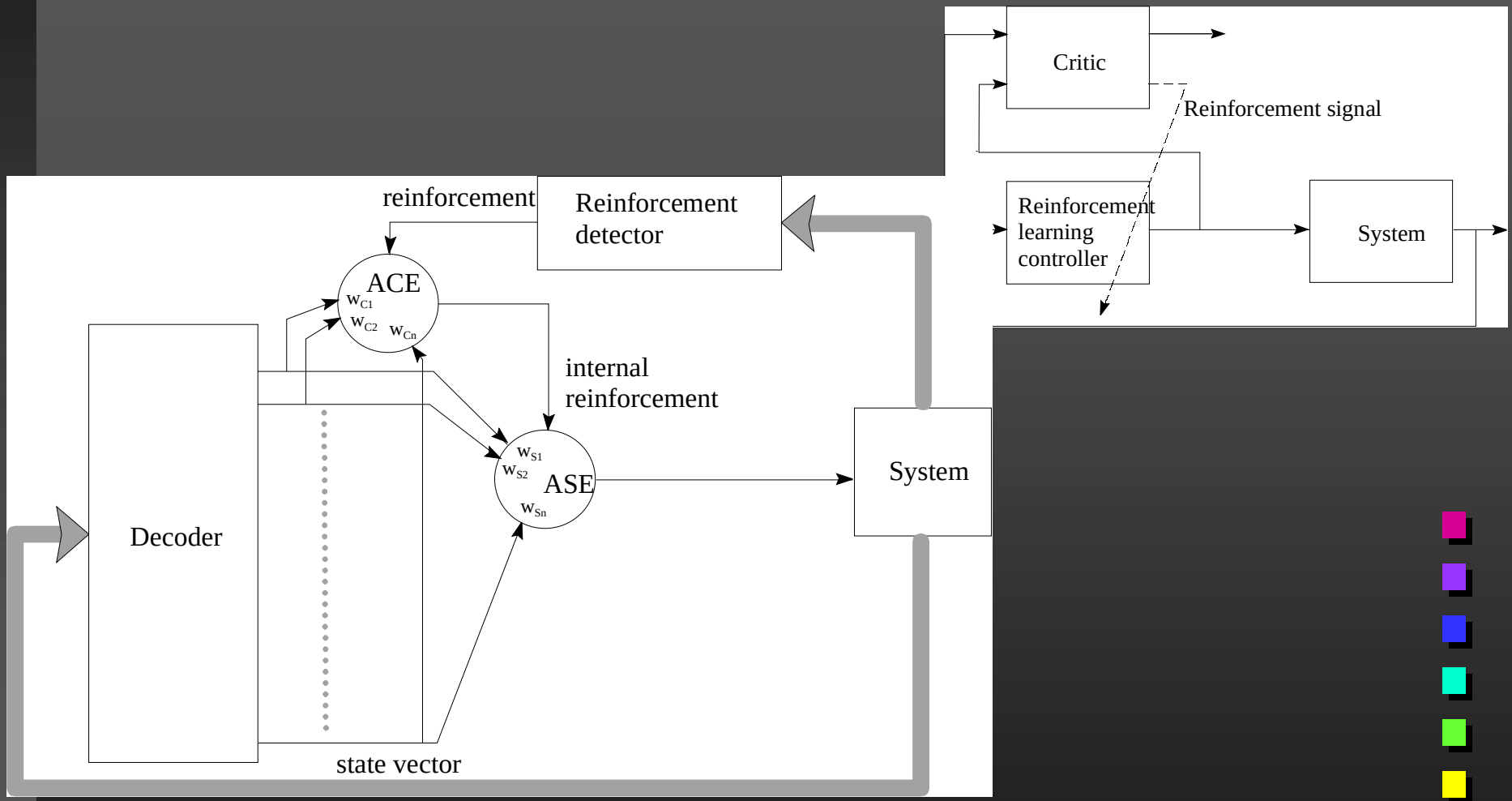


- 1949 - Hebb: psychologist, the first learning rule
- 1959 - Rosenblatt: the first who used “perceptron” for single layer network
- 1960 - Widrow, Hoff: adaline (ADAptive LINear Element) – single layer,
  - the first analiticaly derived learning rule (least squares),
  - until then only heuristical learning, in analogy with BIO systems
- 1963 - Widrow, Smith: inverted pendulum, adaline
- 1969 - Minsky, Papert show the limitations of perceptrona, that can not be used
  - for classification of elements that are not linearly separable -XOR function
  - can not be represented
  - nothing much happens until app. 1982
  - The end of single layer networks period!

- 1986 - Rumelhart and co.: “backpropagation” – learning rule for multi-layer
  - perceptron – classification of elements that are not linearly separable,
  - the nonlinear mapping!!!
- **1988 - Psaltis:** ANN used as controller
- **1990 - Narendra in Parthasarathy:** system identification and control with ANN
- **1995 - Sjöberg et co.:** nonlinear system identification - nonlinear regression

- Newton optimisation methods (2nd order gradient optimisation method)
  - Gauss-Newton modification
  - Levenberg-Marquardt modification
- Other kinds of learning (various determenistic and stochastic optimisation methods)

Reinforcement learning



- Approximation of Dynamic Programming for controller learning

ANN today

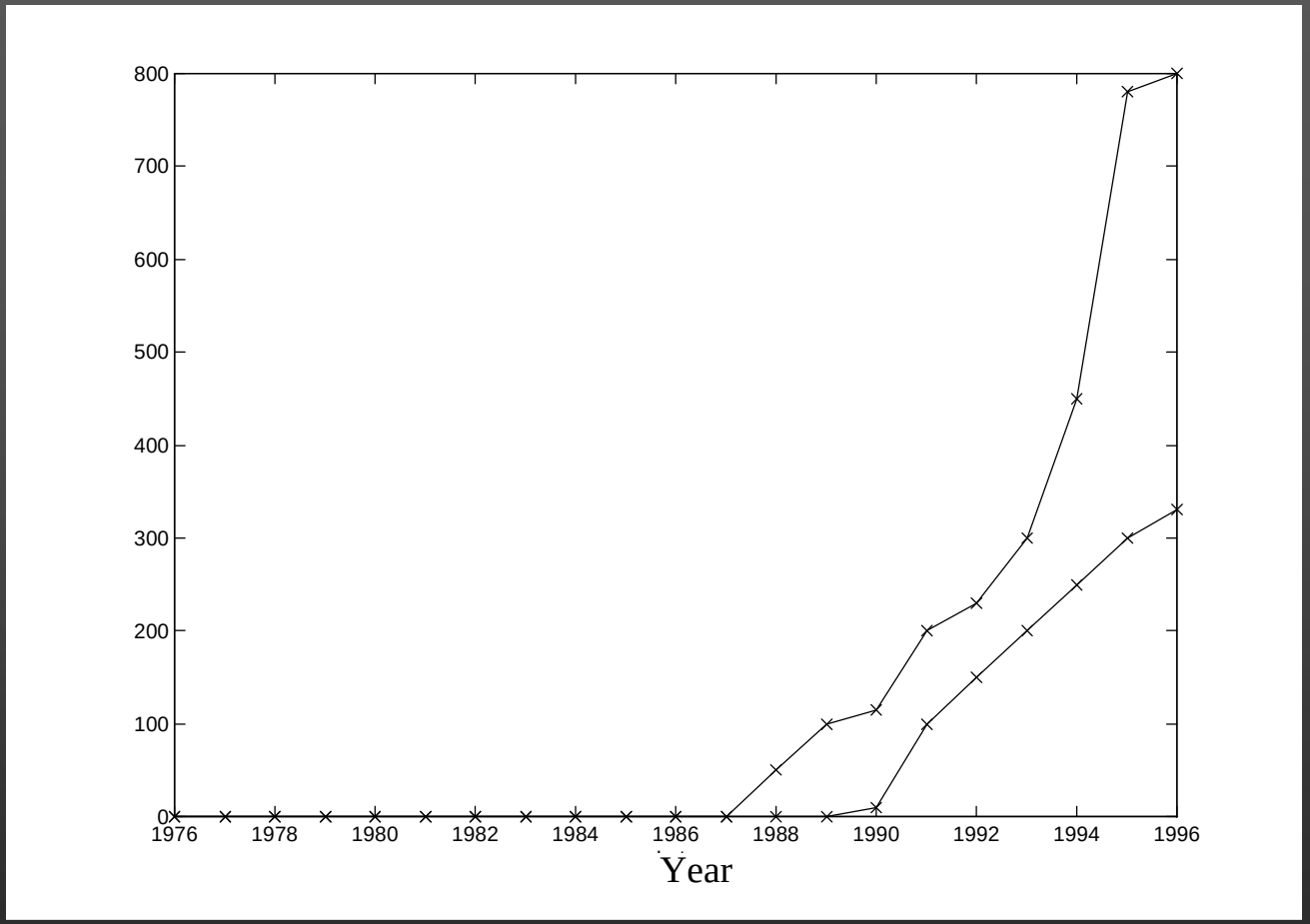
- **Definition of ANN:** does not exist, only *common main properties*:

  - composed of large number of simple and interconnected elements
  - in general adaptive topology
    interconnections change – learning rules change “weights”
    number of network elements is changing
  - topology enables parallel information processing (?)
  - data are processed according to ANN states and inputs

- **Have ANN fulfilled expectations?**

- *Early days*: comprehension of BIO systems with mathematical modelling,

strong connection between BIO research and ANN, ***large*** *expectations*



Growth of published conference papers in previous century – ANN in control loop

(from: G.W. Ng: Application of Neural Networks to Adaptive Control of Nonlinear Systems)

Applications of ANN

- Aerospace (automatic pilots, fault detection etc.)
- Automotive industry (automatic control etc.)
- Finance sector (document recognition etc.)
- Electronics (control, pattern recognition etc.)
- Artificial speach (source recognition etc.)
- Production systems (control, forecasting etc.)
- Medicine (signal processing applications etc.)
- Accountancy (trends forecasting etc.)
- Robotics (control, signals processing etc.)
- Telecommunications (data compression etc.)
- Transportation sciences (diagnosis systems etc.)
- Military industry (radar and sonar signals processing etc.)
- Entertainment industry (animation, special effects etc.)
- Insurance (assets optimisation etc.)

- ANN – **intelligent systems** ???
  pattern recognition
  systems learned from examples
  generalisation ability

- ***Paper 1992*** - (Kohonen, Neural Networks, 1988): "ANNs are parallel inter-connected networks of simple computational elements **which are intended to interact with the objects of the real world in a similar way to biological nervous systems**."

• ANN development in two separate directions:
  - classification, pattern recognition – very close to ANN
  - regression (ANN for systems identification and control, time-series identification)

differences: methodology, topology, learning rules, fields of applications

Important properties of ANN

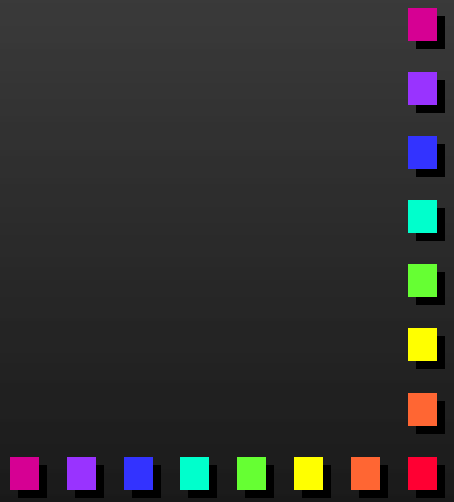
a) ANN as nonlinear mapping – at least two network topologies that are *universal approximators*
unknown function $f(\underline{x})$, $f(\underline{x})$: $\Re^{nx} \rightarrow \Re^{ny}$, $\underline{x}$ ó $\underline{y}$
ANN aproximates $f(\underline{x})$

b) Ability to learn from examples ($\underline{v}$, $\underline{y}$)
ANN parameter optimisation based on selected cost function, e.g.

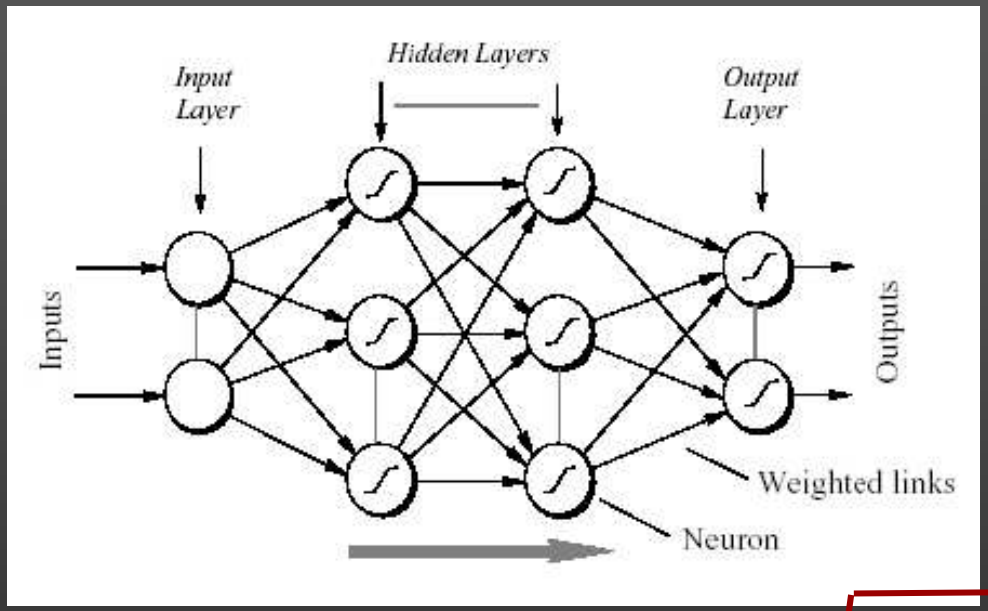
$$E(k)=\frac{1}{2}\sum_{j=1}^{n}(d_j(k)-y_j(k))^2$$

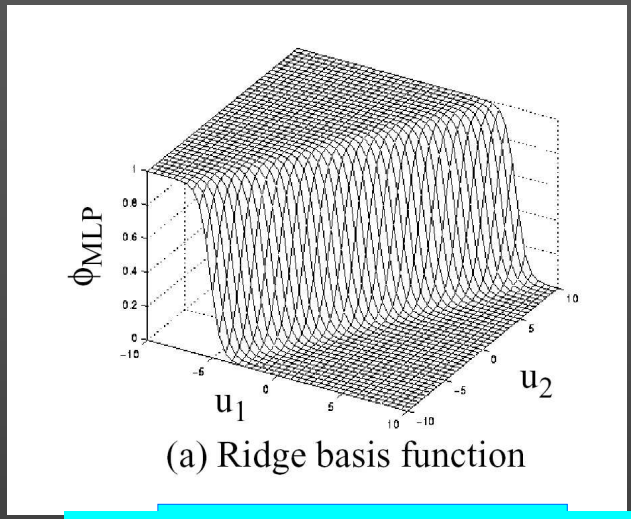
with selected optimisation algorithm (learning rule)

c) Generalisation ability (comes from a))



Multi-layer perceptron



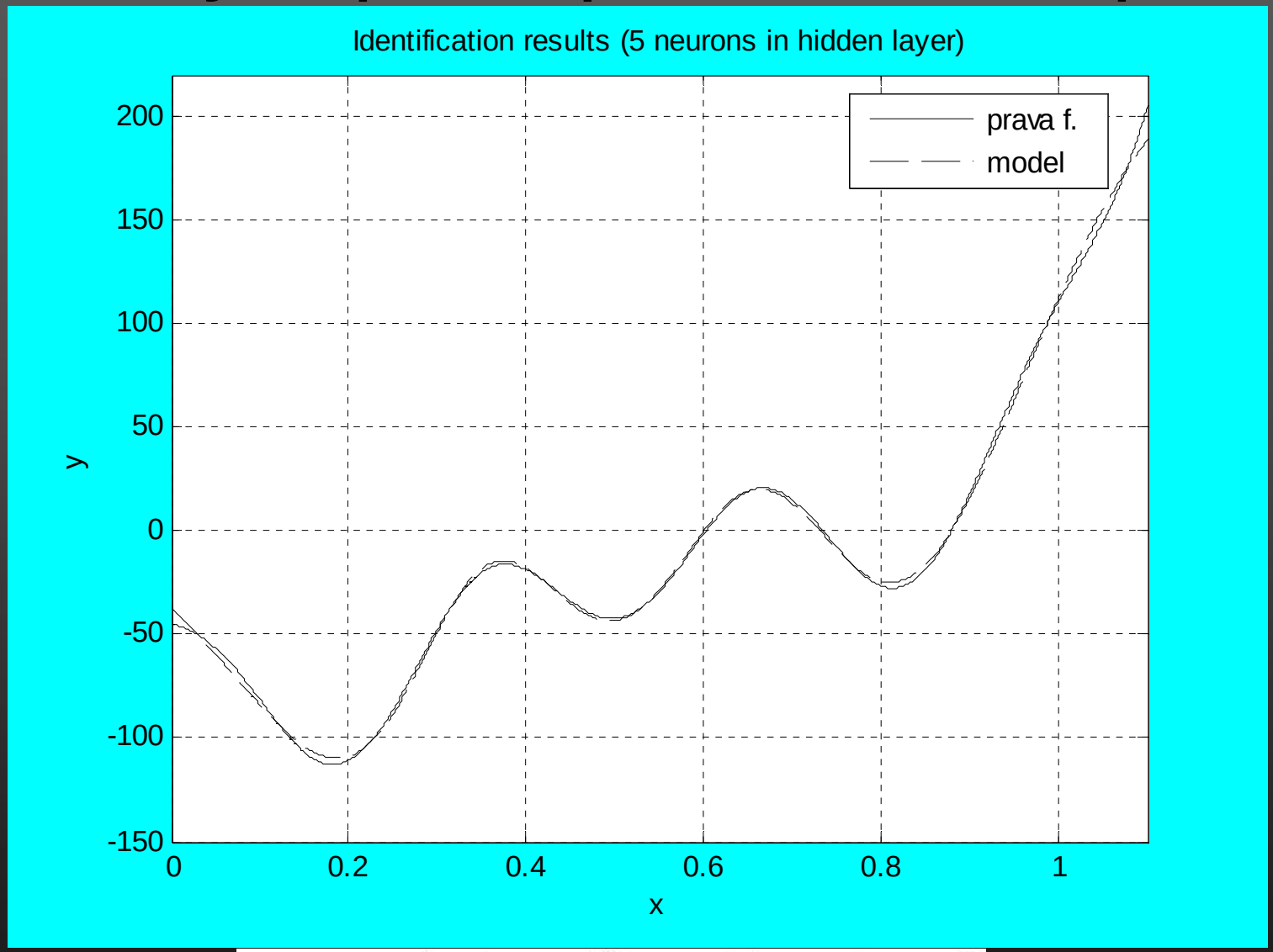


(a) Ridge basis function

Sigmoid or tanh activation function

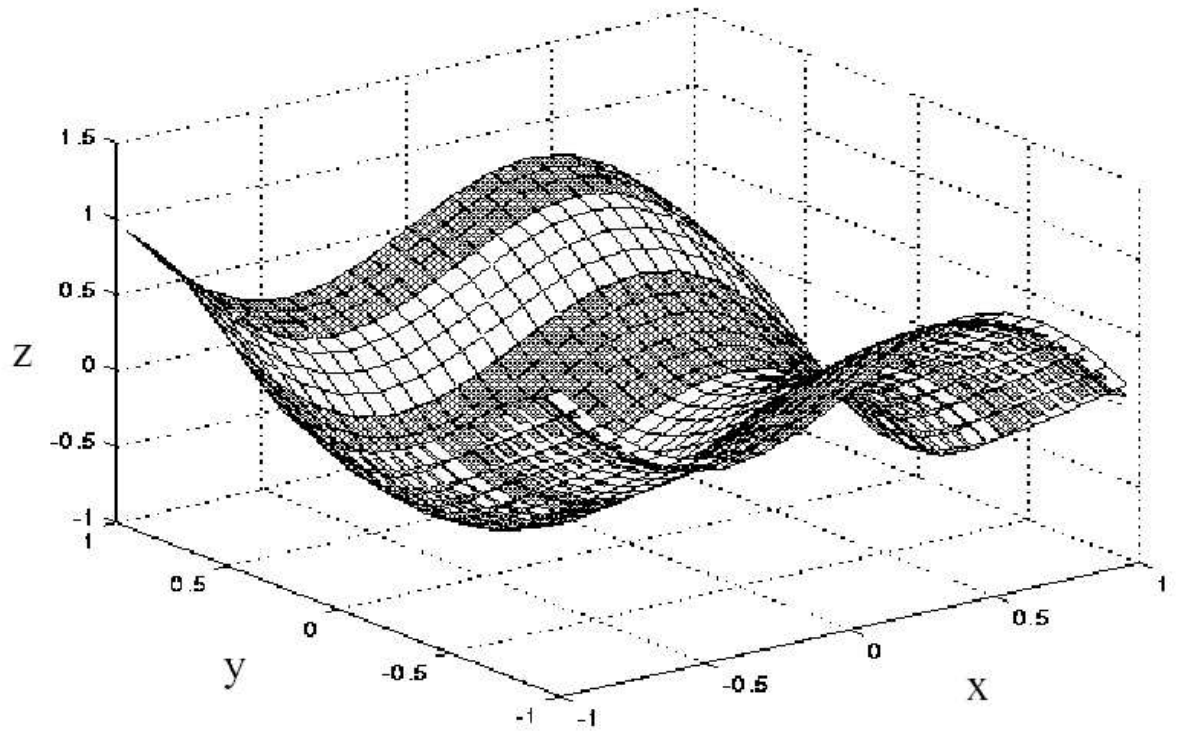
$$y = g(\mathbf{u}, \mathbf{w}) = \sum_{j=1}^{N_h} c_j a\left(\sum_{i=1}^{N_i} w_{ij} u_i + b_j \right) + d$$

Multi-layer perceptron – Example 1



$$y = 30\left[100(x-0.6)^2(x-0.1)(x-0.8) - 5e^{-5x} + 0.9\sin(20x+\frac{\pi}{4})\right] \quad ($$

Multi-layer perceptron – Example 2

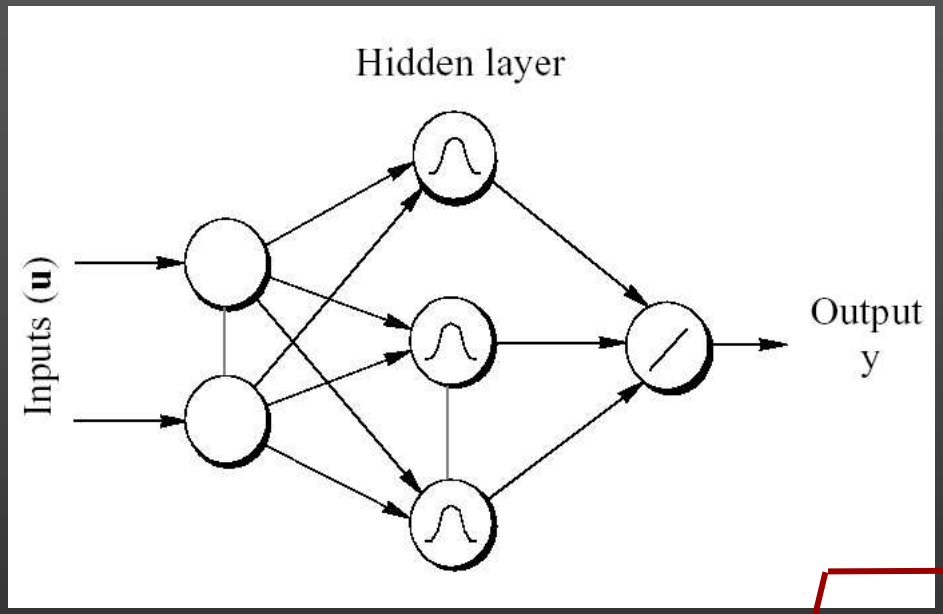


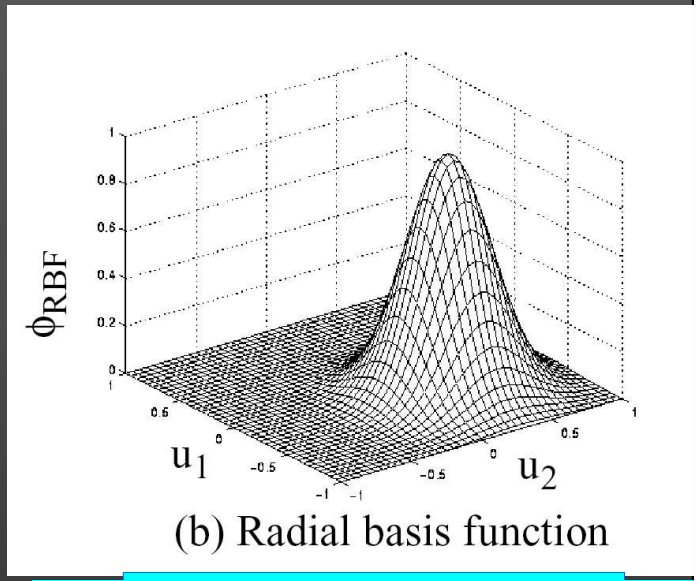
(c) Best approximation using an MLP(2,4,1)

Key: Approximation : Desired :

$$z(x, y) = x\cos(2x) + y\sin(2y) - 0.75$$

The Radial Basis Function Network



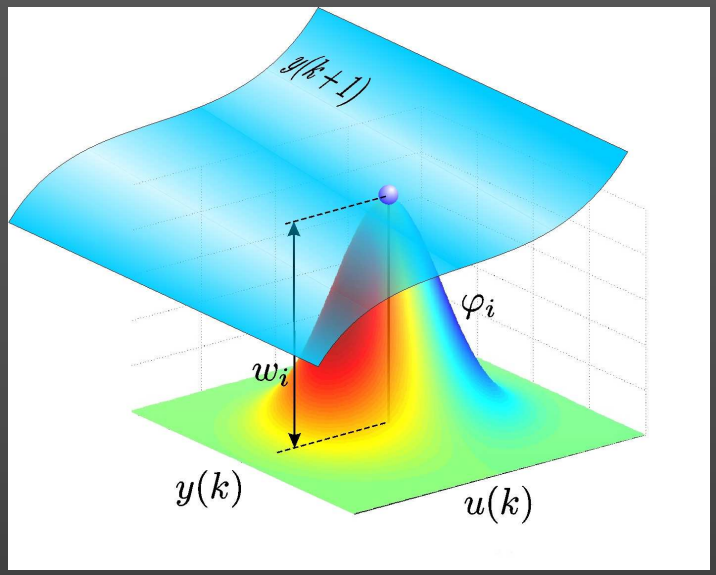


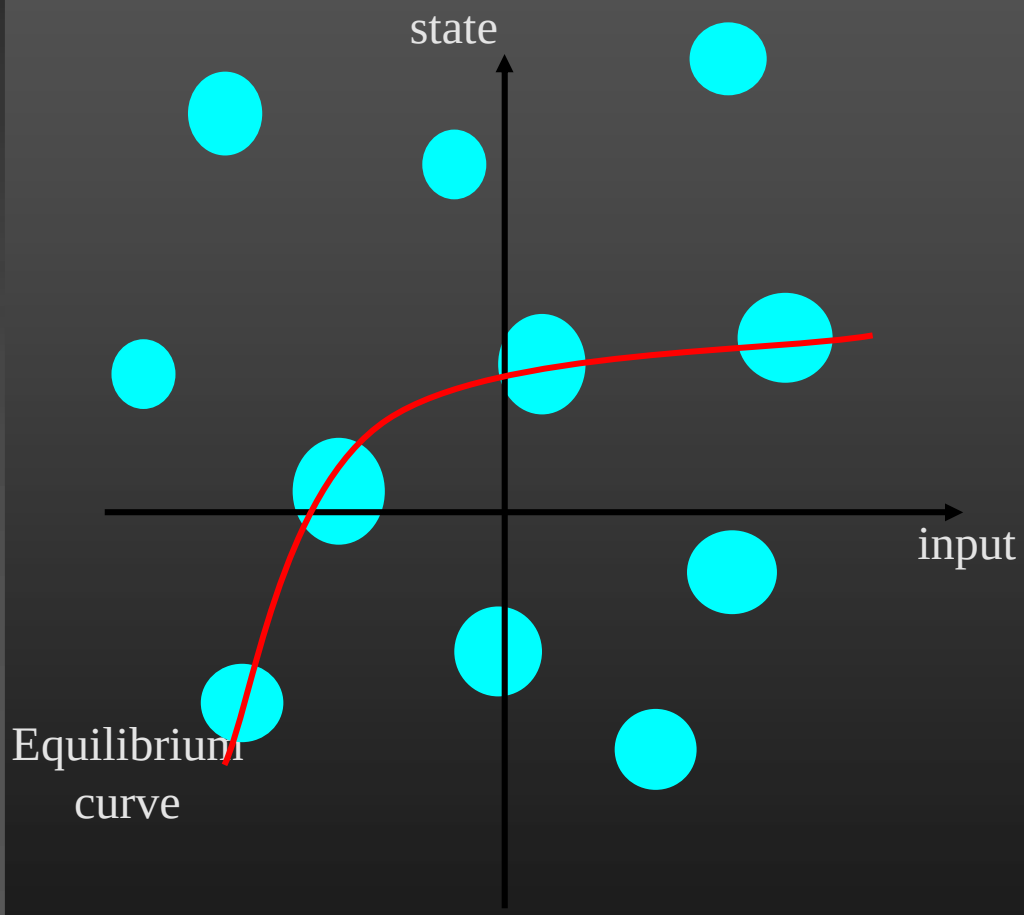
(b) Radial basis function

Gaussian function

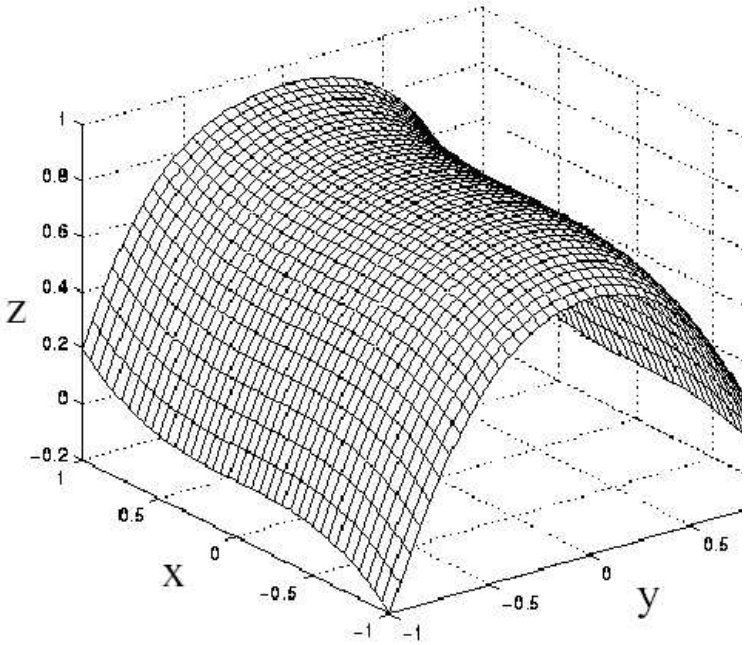
$$y = \sum_{i=1}^{N_h} h_i \exp\left(-\frac{\left(\sum_{j=1}^{N_i} (u_j - c_{ij})^2 \right)}{w_i^2} \right)$$

Radial basis function network principle

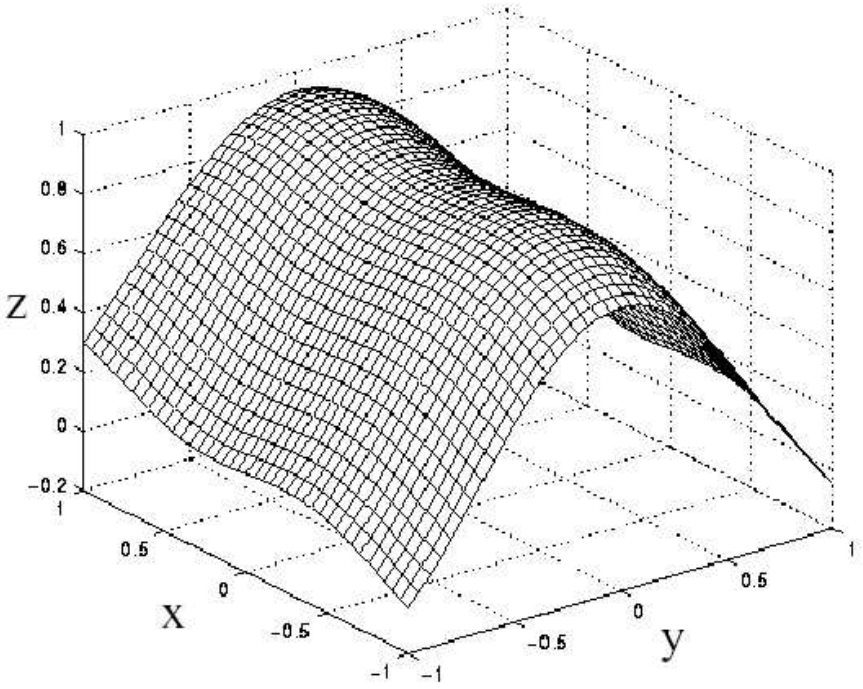




RBF network – Example



(a) Original function



(d) Approximation for w=0.8

$$z(x, y) = \sin(0.4x^3 - 1.6y^2 + 0.5)$$