Neural Networks for Erasmus 4

dr hab. inż. Michał Bereta
room 144 / 8, Institute of Computer Science
mbereta@pk.edu.pl
www.michalbereta.pl/nn

Classification problems

Linearly seperable data

- Two classes
  - One perceptron needed
- More than two classes
  - Each class has its own neuron
  - For this neuron, its class is “1”, all other classes are “-1”
  - Multi-class problem is decomposed into several two class problems

Classification problems

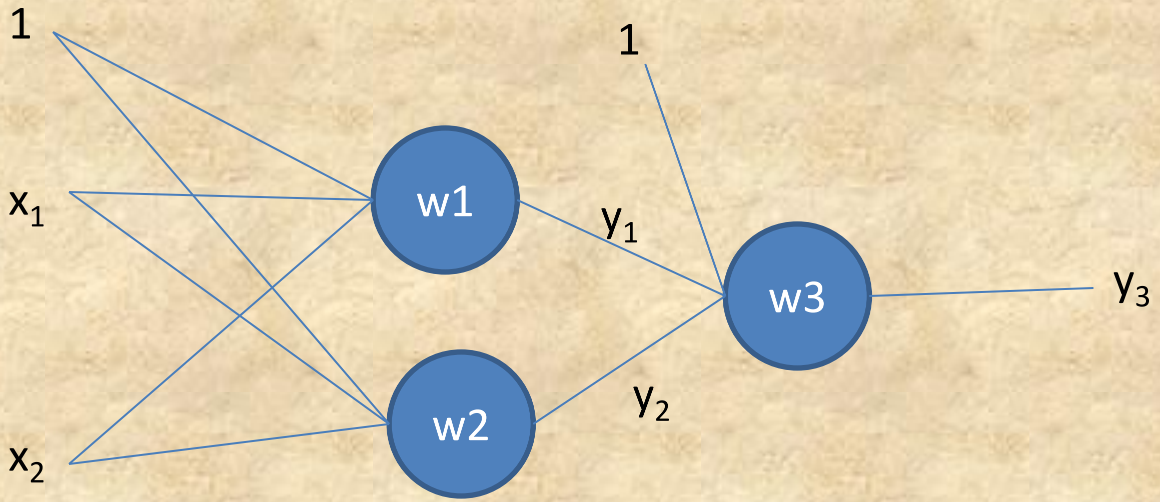
- Linearly nonseparably data (nonlinear problem)
  - You can try one neuron
  - Adding nonlinear features by hand
  - **Use network of neurons**

For regression

- Linear
  - Widrow-Hoff model
- Nonlinear functions
  - Adding nonlinear features by hand
  - **Use network of neurons**

Neural Nets

- Can we use only linear neurons to create a net?



$\mathbf{w}_1=[w_{10}, w_{11}, w_{12}]$ $\mathbf{w}_2=[w_{20}, w_{21}, w_{22}]$ $\mathbf{w}_3=[w_{30}, w_{31}, w_{32}]$

Neural Nets

$y_1 = w_{10}+w_{11}x_1+w_{12}x_2$

$y_2 = w_{20}+w_{21}x_1+w_{22}x_2$

$y_3 = w_{30}+w_{31}y_1+w_{32}y_2 =$

$w_{30}+w_{31}*(w_{10}+w_{11}x_1+w_{12}x_2) + w_{32}*(w_{20}+w_{21}x_1+w_{22}x_2) =$

$(w_{30} + w_{31}w_{10} + w_{32}w_{20}) + x_1*(w_{31}w_{11}+w_{32}w_{21}) + x_2*(w_{31}w_{12}+w_{32}w_{22})$

$= w_{40} + w_{41}x_1 + w_{42}x_2$

Neural Nets

$y_1 = w_{10}+w_{11}x_1+w_{12}x_2$

$y_2 = w_{20}+w_{21}x_1+w_{22}x_2$

$y_3 = w_{30}+w_{31}y_1+w_{32}y_2 =$

$w_{30}+w_{31}*(w_{10}+w_{11}x_1+w_{12}x_2) + w_{32}*(w_{20}+w_{21}x_1+w_{22}x_2) =$

$(w_{30} + w_{31}w_{10} + w_{32}w_{20}) + x_1*(w_{31}w_{11}+w_{32}w_{21}) + x_2*(w_{31}w_{12}+w_{32}w_{22})$

$= w_{40} + w_{41}x_1 + w_{42}x_2$

Linear combination of linear combinations is also a linear combination.

Activation functions

Neurons in hidden layers should have a nonlinear activation functions.

$$f(u_i) = \begin{cases} 1 & u_i > 0 \\ 0 & u_i \leq 0 \end{cases}$$

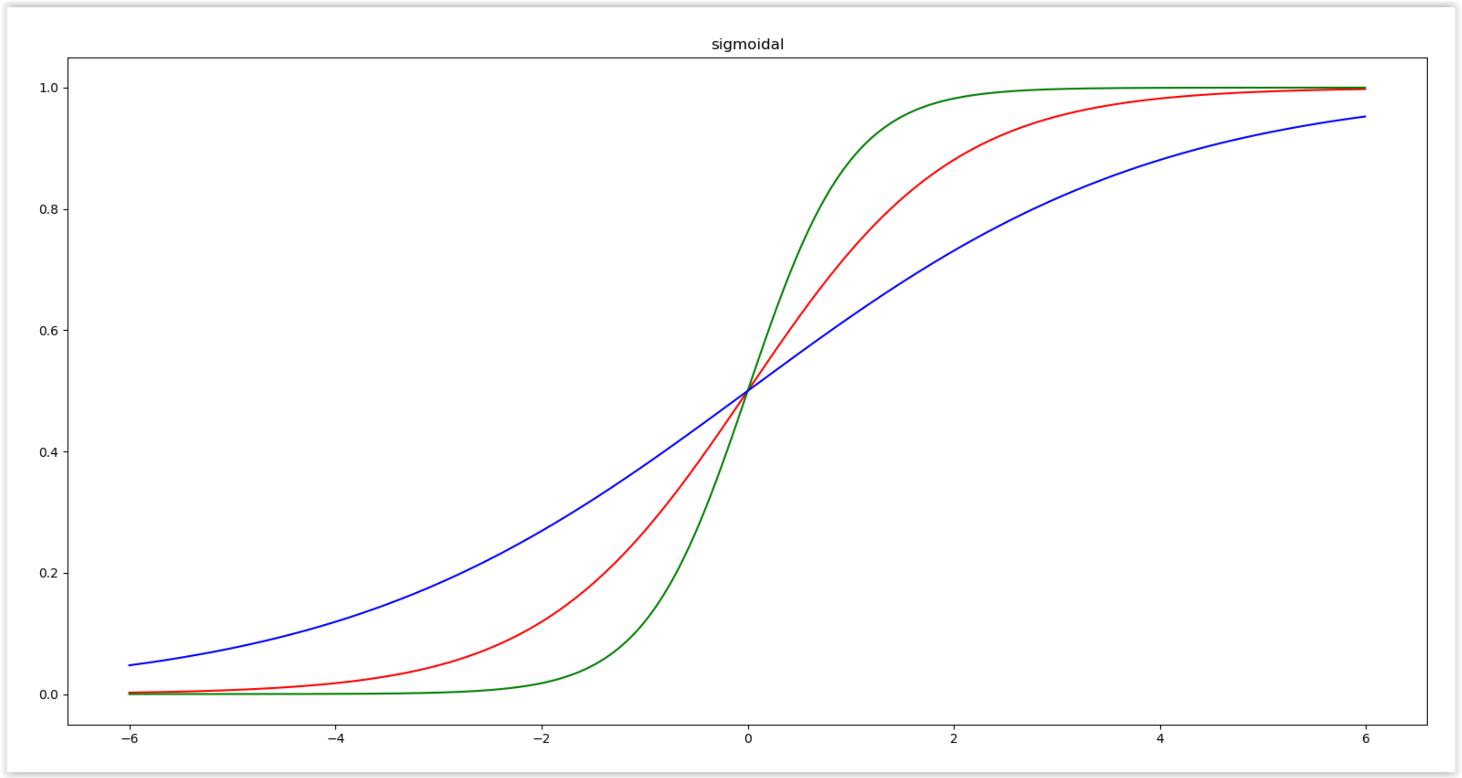
Unipolar activation function

Problem:

•It is not continuous

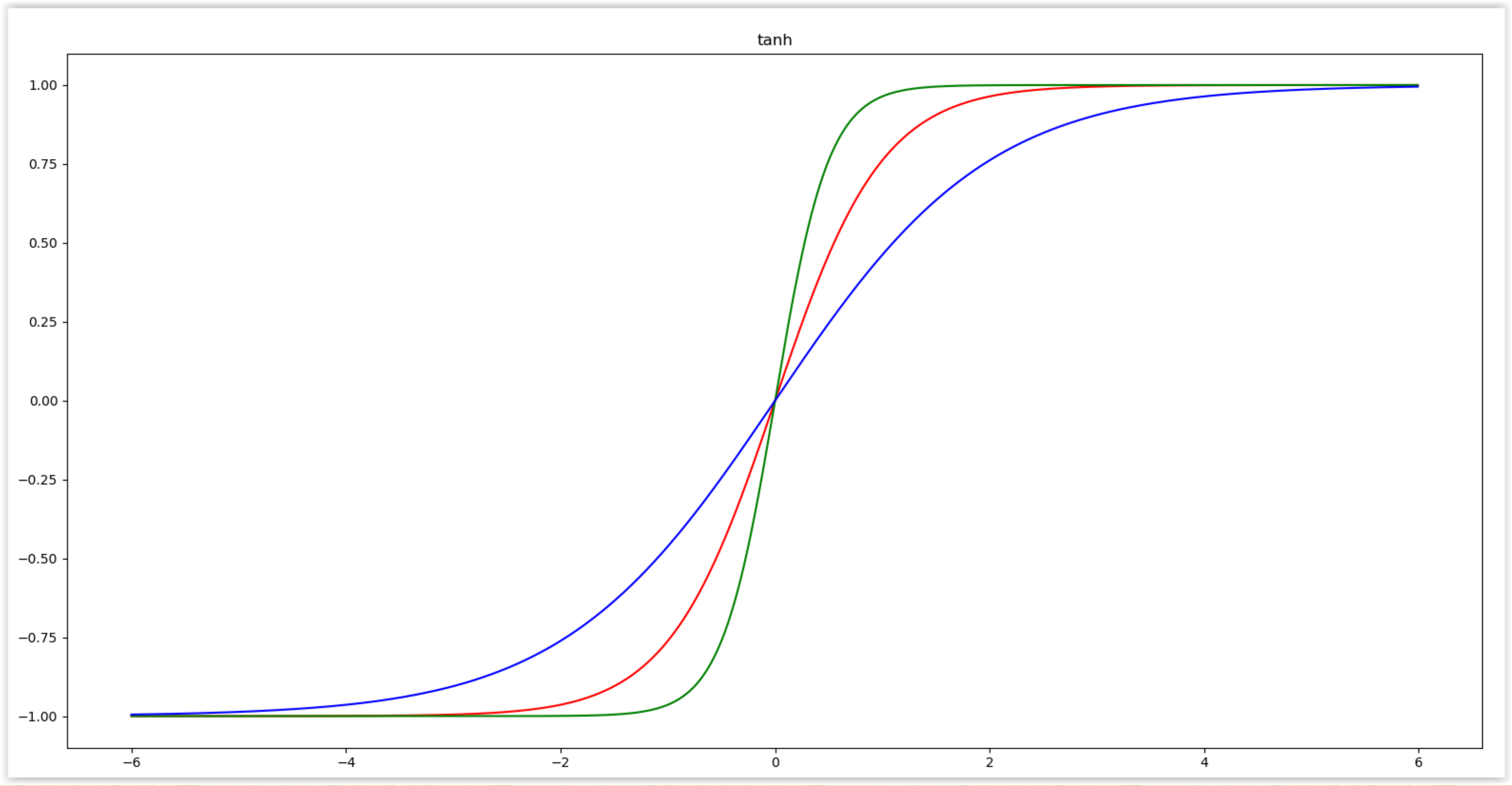
•There will be problems with calculation of gradients

Activation functions



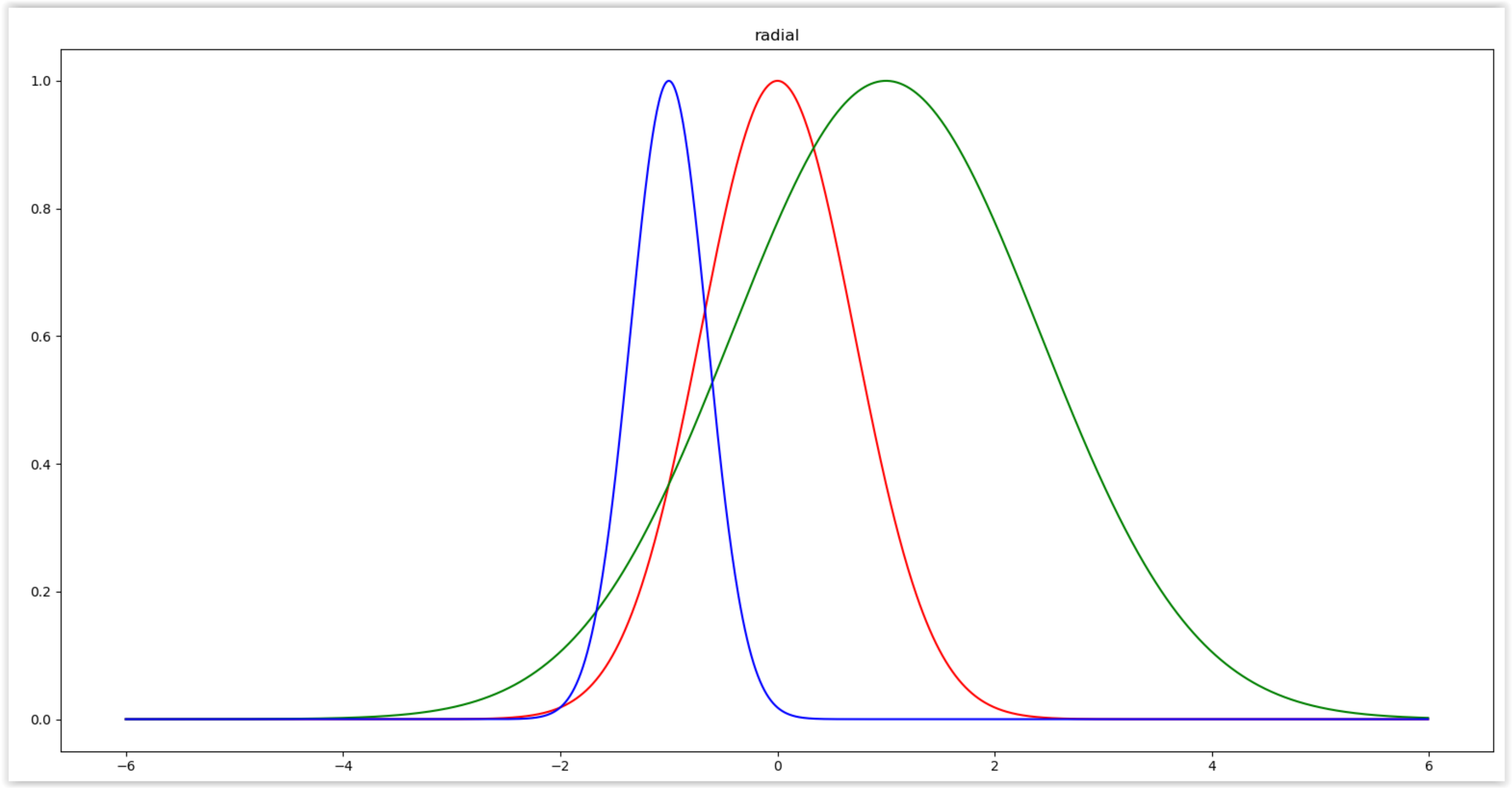
$$f(x) = \frac{1}{1+e^{-\beta x}}$$

Activation functions



$$f(x) = \tanh(\beta x)$$

Activation functions



$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

As in the normal distribution

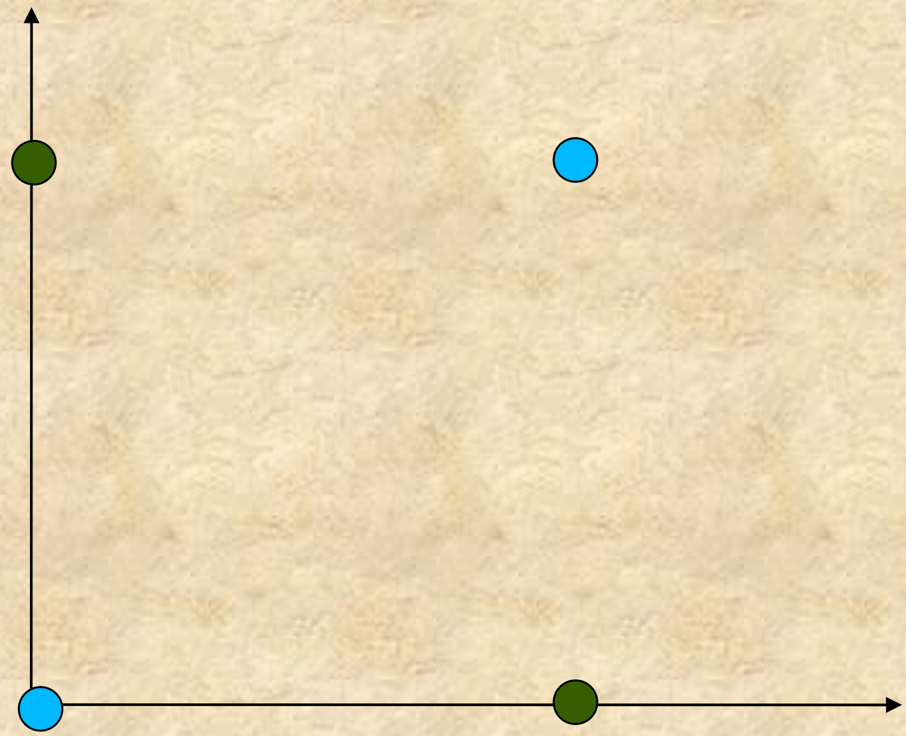
Neural Nets

- Hidden layers, due to the nonlinear transformations, transfer the problem into a new space.
- Linear neurons from output layer solve a linear problem (in the new space)

XOR problem

XOR

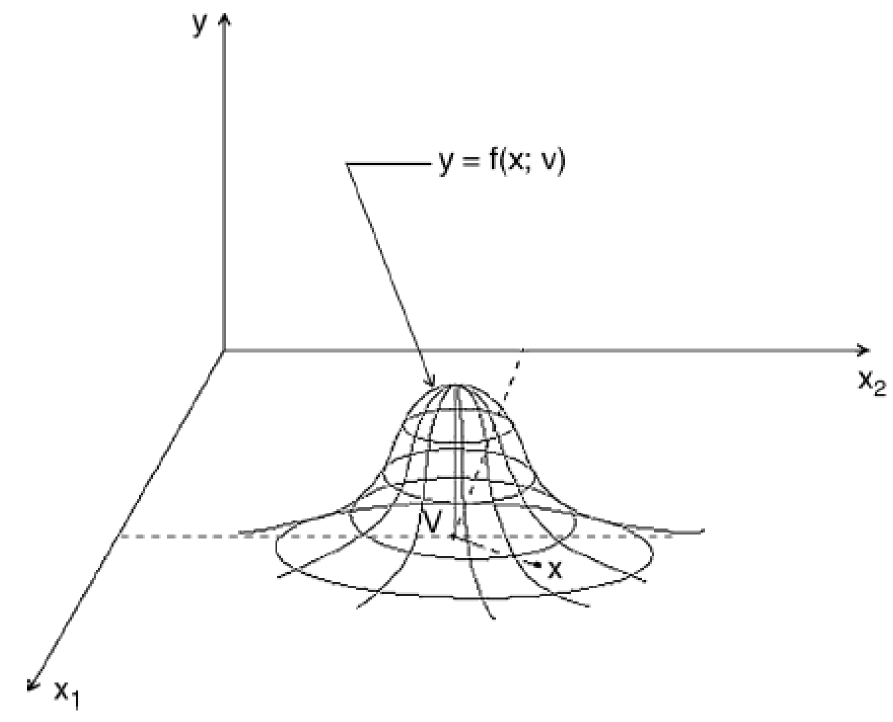
x y	**x xor y**
0 0	0
1 0	1
0 1	1
1 1	0



Cannot be solved with linear model

XOR problem

Nets with radial basis functions (RBF)

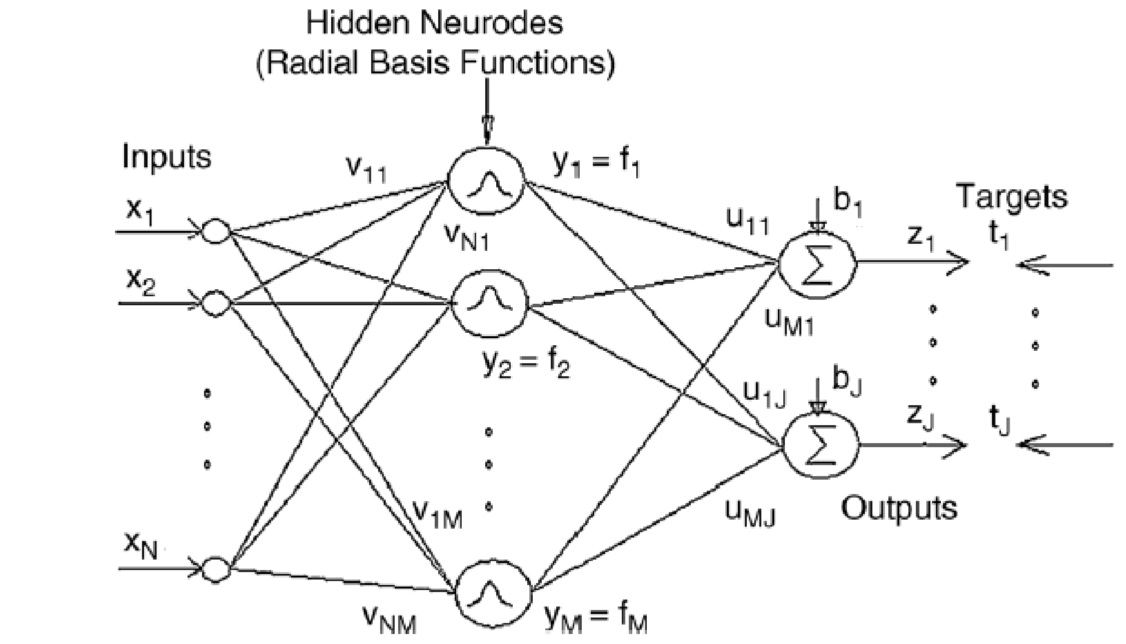


A radial basis function centered on v.

Gaussian Radial basis function: $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$

RBF

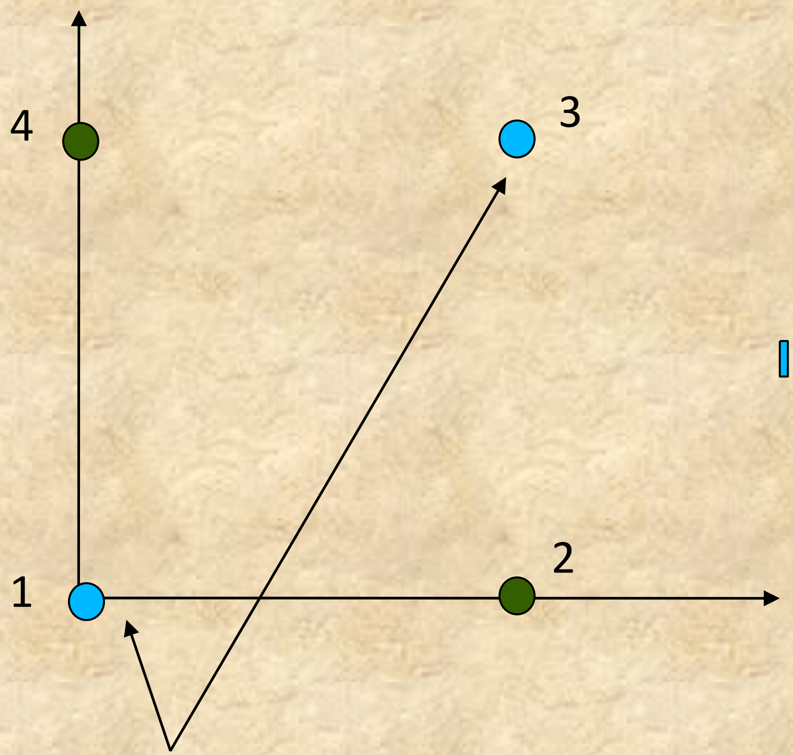
Radial basis functions networks



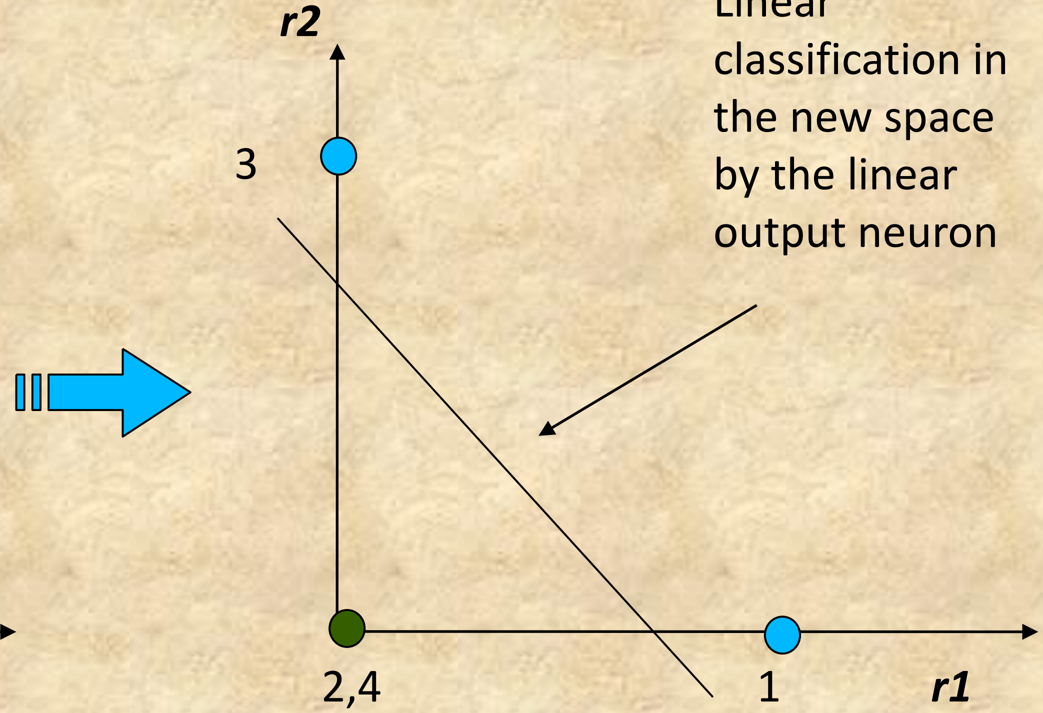
The radial basis function neural network architecture.

XOR problem

Nets with radial basis functions (RBF)



There we place two radial neurons ***r1*** and ***r2***



Linear classification in the new space by the linear output neuron

In the new space, the problem is linear!

Nets with radial basis functions (RBF)

- Also for regression

Gaussian functions

- We can decide where to place them
  - They are fixed, not changed during training of output layer
- Their positions and widths can be learned from data
  - Random initialization
  - Training modyfies the whole network, both layers