About cross synaptic neuron model

Michal Chovanec
michal.chovanec@yandex.com

Keywords: neural network, neuron model, modification
Abstract: In this paper, experimental neuron model with ability multiply inputs is described. Many tests an comparsion with other common models has been processed on funcion aprxoimation problem.

1 Introduction

Well know McCulloch Pitts neuron model

$$y(n) = \varphi(\sum_{i=0}^{N-1} x_i(n)w_i(n)) \tag{1}$$

where
$x(n)$ is input vector
$y(n)$ is neuron output
$w(n)$ is weight vector
$\varphi(t)$ is neuron transfer function.

Common used transfer functions are (citovat)
linear $\varphi(t) = t$
tanh $\varphi(t) = tanh(t)$
step $\varphi(t) = sgn(t)$

very popular is also rectified neuron model $\varphi(t) = max(0, t)$

Multilayer neural network using this model can be used as universal function aproximator [1]. Usually many hidden layers need to be used, which is dificult to learn using backpropagation algorithm - local minima.

2 Proposed model

We define following neuron model with ability to multiply two signals

$$y(n) = \varphi(\sum_{i=0}^{N-1} x_i(n)w_i(n) + \sum_{j=0}^{N-1}\sum_{i=j}^{N-1} x_i(n)x_j(n)v_{ji}(n)) \tag{2}$$

where v_{ji} is matrix representing weights for multiplied inputs.

For learning process we can use common backpropagation algorithm

$$\delta w_i(n) = \eta e(n)x_i(n) \tag{3}$$

$$e_i(n) = w_{ji}(n)e_j(n)\varphi'(y_j(n)) \tag{4}$$

For our experiments linear and tanh transfer function has been used, when linear transfer function is used, we can write

$$e_i(n) = w_{ji}(n)e_j(n) \tag{5}$$

and finally for tanh

$$e_i(n) = w_{ji}(n)e_j(n)y_j(n)(1 - y_j(n)) \tag{6}$$

3 Experimental results

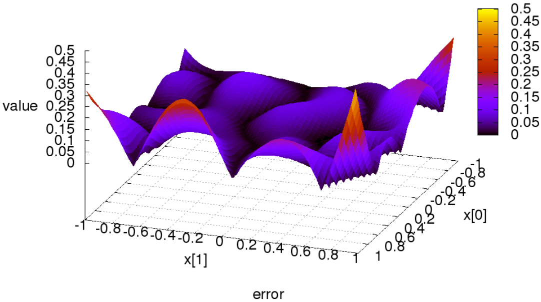
yeah, fucking awesome, many pictures == many pages to be taken

3.1 Experiment 1

4 Zaver

Literatúra

[1] Kolmogorov's Theorem, http://neuron.eng.wayne.edu/tarek/MITbook/chap2/2_3.html



Obr. 1: experiment schematic