Signal mixing using neural network

Michal Chovanec
michal.chovanec@yandex.com

Keywords: neural network, multiplexer, signal processing
Abstract: This paper presents multiplexer realized using neural network. This can be usefull when two signals correlation computation is required in neural network (or for problems with signals mixing, vector rotation or Fourier transform). Problems with classical implementation using McCulloch-Pitts neuron model are also discussed.

1 Introduction

It is the well known fact that feed forward neural network can aproximate any continuous function. Using three hidden layers it is possibe to separe any data sets or aproximate any continuous function [1]. In this paper we describe multiplexer problem, solved using neural network with one hidden layer. With common McCulloch-Pitts neuron model it is diffucult to aproximate this function.

Common used neuron transfer function is McCulloch-Pitts neuron model [2]

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

where
x is input vector
w is weights vector
φ is activation function, common used $\tanh(q)$, sigmoid or linear function.

Single layer in feed forward neural network can be written as [3]

$$Y(n) = \sum_{j=0}^{M-1} \varphi(\sum_{i=0}^{N-1} x_i(n) W_{ij}(n)), \tag{2}$$

where
M is neurons count in layer

W is weight matrix (NxM)
$Y(n)$ is output vector of size M.

Some common problems, like data separation or clasification can be solved effectively using this neuron model. For problems like signal multiplications, signals mixing or conditional switches can be difficult to learn network with one hidden layer. Network with more than one hidden layer is difficult to learn, especially in real time.

2 Multiplexer problem

Simple boolean multiplexer using in logic circuit with two inputs A, B and select S has following truth table

Table 1: Multiplexer truth table

Input A	Input B	Select S	Output Y
A	B	0	A
A	B	1	B

Using boolean logic we can write

$$y = A\neg S + BS \tag{3}$$

When $A, B, S \in \mathbb{R}$ and $S \in \langle 0, 1\rangle$ we can write nothing else than

$$y = A(1 - S) + BS \tag{4}$$

Considering (1) we can see that realization of this function in neural network can be difficult, because of multiplication of inputs is required. Note : inputs A, B, S corresponds to x_0, x_1, x_2 input vector.

3 Neuron model modification

Let us define following neuron model with two inputs multiplication capability

$$y(n) = \sum_{i=0}^{N-1} x_i(n)w_i(n) + \sum_{j=0}^{N-1}\sum_{i=j}^{N-1} x_i(n)x_j(n)v_{ij}(n), \tag{5}$$

where
$v(n)$ is matrix representing influence of product of each input pair.

We can define error by

$$e(n) = y_r(n) - y(n), \tag{6}$$

where $y_r(n)$ is required output and $y(n)$ neuron calculated output, and using gradient descent method we can modify weights in learning process as in [4]

$$w_i(n+1) = w_i(n) + \eta_1 e(n) x_i(n) \tag{7}$$

and similarly for $v(n)$ matrix

$$v_{ij}(n+1) = v_{ij}(n) + \eta_2 e(n) x_i(n) x_j(n) \tag{8}$$

where η_1, η_2 are learning rate constants, and can be changed during learning process.

4 Experimental results

For testing multiplexer we use two neurons models, 1 and 5. There were four inputs defined as $x_0 = A$, $x_1 = B$, $x_2 = S$, $x_3 = 1$ where x_3 represents bias input. Inputs were normalized into $\langle -1, 1 \rangle$ interval. In multiplication neuron model single hidden layer were used (with 4 neurons), for McCulloch-Pitts (using hyperbolic tangent activation function) during learning process random inputs were selected, processed output and backpropagation learning in real time. In both cases, constant learning rate is used $\eta_1 = \eta_2 = 0.01$. We choose 4000 random input vectors, and corresponding required values (using 4). After 4000 iterations learning has been turned off. On figure 1 we can see resulting errors on testing data (randomly selected inputs).

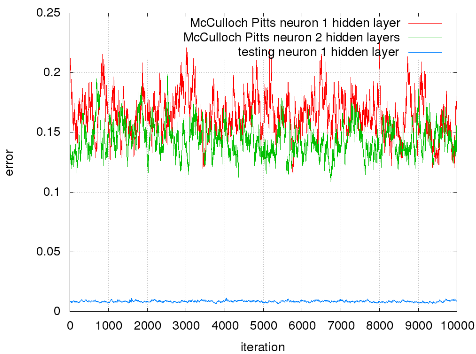

Figure 1: Multiplexer approximation error

5 Conclusion

In this paper we described neuron model improvement for two inputs multiplications. Using only single hidden layer some difficult problems for McCulloch-Pitts neural network can be solved. It is necessary to normalise input and output vectors into $\langle -1, 1 \rangle$ interval, beacuse using high order powering in 5 can be learning algorithm unstable.
Model has been tested on multiplexer problem learning using backpropagation. In future, more tests of these models are required. Some problems need to be solved : error backpropagation, hybrid learning using stochastic optimization and more application examples and tests.

References

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

[2] McCulloch Pitts neuron model, http://www.mind.ilstu.edu/curriculum

[3] Comparsion of feedforward and reccurent neural network language models, M. Sundermeyer, I. Oparin 2, J.-L. Gauvain, B. Freiberg 1, R. Schluter, H. Ney, 2013 IEEE

[4] Backpropagation algorithm, http://page.mi.fu-berlin.de/rojas/neural/chapter/K7.pdf