

---

# MULTI LAYER PERCEPTRON REPORT

---

27/05/19

Jose Luis Rocabado Rocha  
Octavio Sales  
Lodz University of Technology

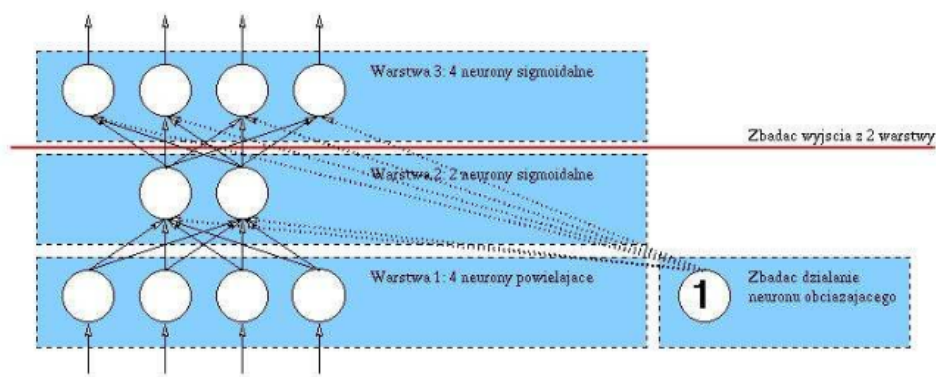
# Contents

0.1	Multilayer Perceptron Introduction . . . . .	2
0.2	Output of Results . . . . .	3

## O.1 MULTILAYER PERCEPTRON INTRODUCTION

The aim of the laboratory is write a program which implements multilayer perceptron with the structure of 3 layers listed below:

- input layer with 4 neurons,
- hidden layer with 2 neurons,
- output layer with 4 neurons.



**Figure 1:** Schema of Multilayer Perceptron

Using **error Backpropagation Method** to train the network.

This method consists of two steps, once the input is already propagated to the output of the neuron, an error signal is calculated for each of the outputs, and finally the error signals are propagated backwards, starting from the output layer, towards all the neurons of the hidden layer that contribute directly to the output of the network.

## 0.2 OUTPUT OF RESULTS

In our code, we are getting all the data from a text file. So we need a text file within patterns which we want to train in the multilayer perceptron network called "patterns.txt". In our case this file contains:

```

INPUTS , OUTPUTS
1 0 0 0 , 1 0 0 0
0 1 0 0 , 0 1 0 0
0 0 1 0 , 0 0 1 0
0 0 0 1 , 0 0 0 1

```

Because **the network is implemented within the BIAS neuron**, we can train all the set of patterns well. The bias neuron allows to converge to the expected output, because it compensates the value of the weights of the hidden layers that can not change their value as much as we would like. This makes it possible to move or "translate" the activation function left or right on the graph.

the result of the training with these specific features:

- number of epochs: 10000,
- value of the number NU=1,

is given below:

```

Pattern 0
expected out: 1, 0, 0, 0
network out :0.941783, 0.0456701, 0.0450885, 0.000326683
hidden neurons out :0.0136974, 0.0211111
Pattern 1
expected out: 0, 1, 0, 0
network out :0.0360071, 0.948842, 9.93859e-05, 0.0566339
hidden neurons out :0.954565, 0.0140412
Pattern 2
expected out: 0, 0, 1, 0
network out :0.0373884, 9.21156e-05, 0.94764, 0.0561032
hidden neurons out :0.0186665, 0.965887
Pattern 3
expected out: 0, 0, 0, 1
network out :6.75883e-05, 0.0344442, 0.0363571, 0.932918
hidden neurons out :0.981997, 0.980237

```

**Figure 2:** Result of training MLP network with BIAS neuron

As you can see, the network is well trained, since the output corresponds to what was expected.