Machine Learning - 1100-ML0ENG (Ćwiczenia informatyczne Z-23/24)

Neural networks - intro

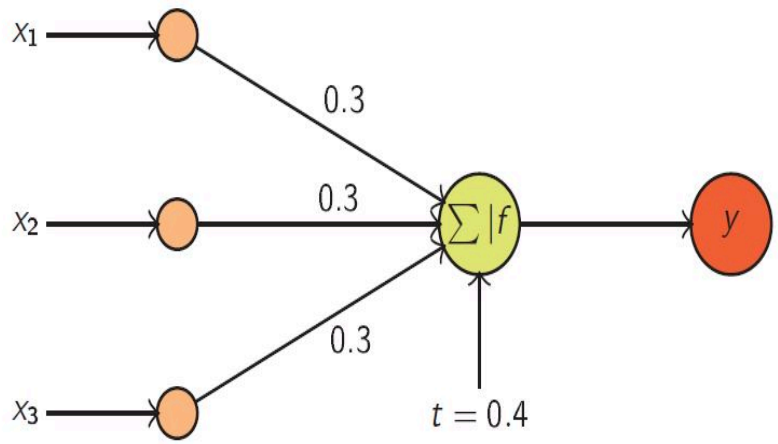
Artificial Neural Networks (ANN) models are powerful classification tools, which can be applied to several different areas. Unlike tree-based methods, the process of how neura networks transform from input to output is less clear and can be hard to interpret. As a result neural networks are referred to as **black box methods**.

Analogous to human brain structure, an ANN is composed of an interconnected assemb of nodes and directed links.

Perceptron

Let us consider the data set containing three boolean variables x_1, x_2, x_3 and an outp variable y.

X_1	X_2	X_3	y
1	0	0	-1
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	-1
0	1	0	-1
0	1	1	1
0	0	0	-1



Here we have **the simplest neural network architecture known as a perceptron**. The perceptron consists of two types of nodes: input nodes, which are used to represent the input attributes, and an output node, which is used to represent the model output.

In a perceptron, each input node is connected via a weighted link to the output node. T weighted link is used to emulate the strength of synaptic connection between neurons.

As in biological neural systems, training a perceptron model amounts to adapting the weights of the links until they fit the input-output relationships of the underlying data.

A perceptron computes its output value $\hat{y}$ by performing a weighted sum on its inputs, subtracting a bias factor t from the sum, and then examining the sign of the result.

$$\hat{y} = \begin{cases} 1 & , 0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4 > 0 \\ -1 & , 0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4 \leq 0 \end{cases}$$

For example, if $x_1 = 0, x_2 = 1, x_3 = 0$, then $\hat{y} = -1$.

The output node is a mathematical device that computes the weighted sum of its inputs, subtracts the bias term, and then produces an output that depends on the **sign** of the resulting sum. More specifically, the output of a perceptron model can be expressed mathematically as follows:

$$\hat{y} = sgn(w_n x_n + \ldots + w_1 x_1 + w_0 x_0)$$

where $w_1, w_2, \ldots, w_n$ are the weights of the input links and $x_1, x_2, \ldots, x_n$ are the input attribute values. The **sgn function**, which acts as an activation function for the output neuron, outputs a value **+1** if its argument is positive and **-1** if its argument is negative, $w_0 = -t$ and $x_0 = 1$.

Learning Perceptron Model

The **learning rate** λ a parameter whose value is between 0 and 1, can be used to control the amount of adjustments made in each iteration. If λ is close to 0, then the new weig is mostly influenced by the value of the old weight. On the other hand, if λ is close to 1, then the new weight is sensitive to the amount of adjustment performed in the current iteration.

The perceptron learning algorithm is guaranteed to converge to an optimal solution (as long as the learning rate is sufficiently small) for linearly separable classification problems.

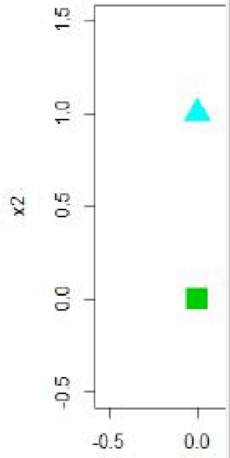
$D = \{(x_i, y_i)\},\ i = 1, ..., n$ - the set of training example
initialize the weight vector with random values w
repeat
for each training example $(x_i, y_i) \in D$
  do
    compute the predicted output $\hat{y}_i^{(k)}$
    for each weight $w_j^{(k)}$
      do
        update the weight $w_j^{(k+1)} = w_j^{(k)} + \lambda(y_i - \hat{y}_i^{(k)})x_{ij}$

until stop

If the problem not linearly separab the algorith fails to converg An example of nonlinea y separa e data i given by the **XOR function**. Perceptron cannot find the right solution for this data becau there is no linear hyperplane that can perfectly separate the training instances.

XOR

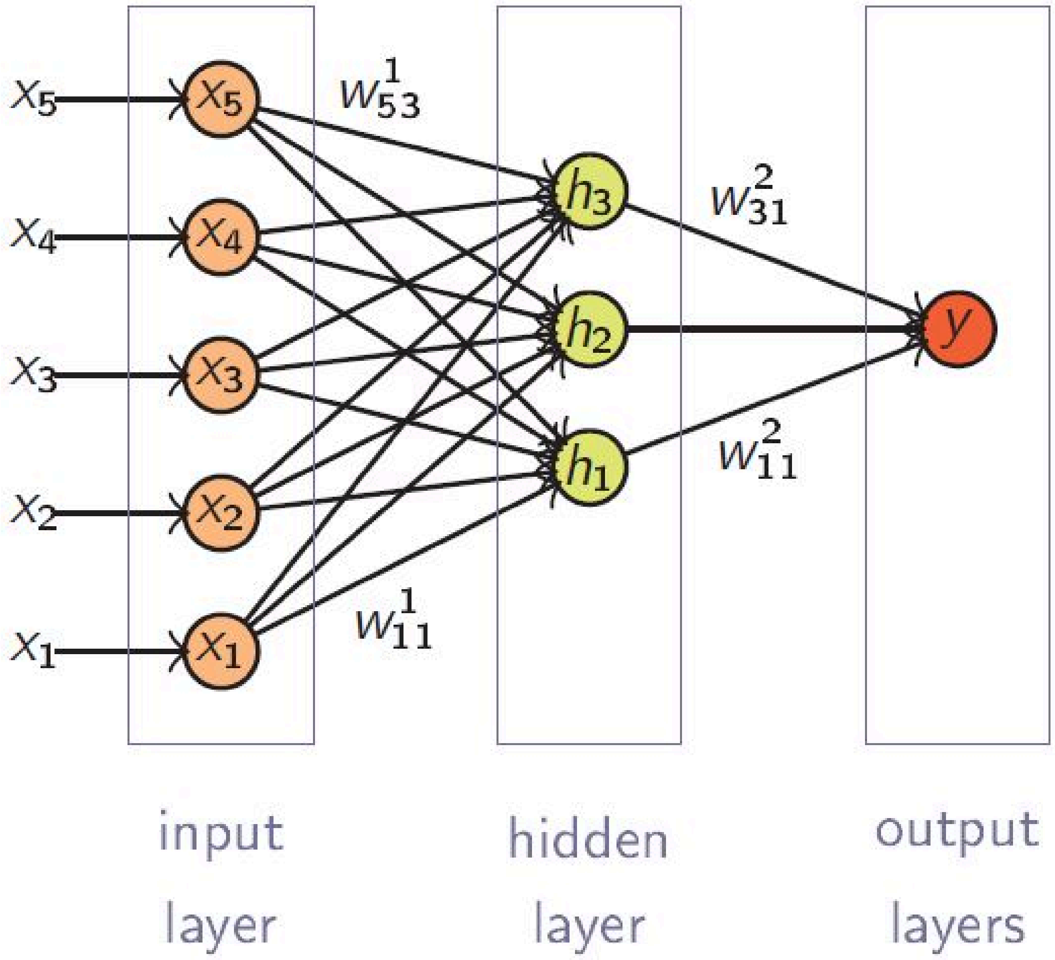
x_1	x_2	y
0	0	-1
1	0	1
0	1	1
1	1	-1



Multilayer Artificial Neural Network

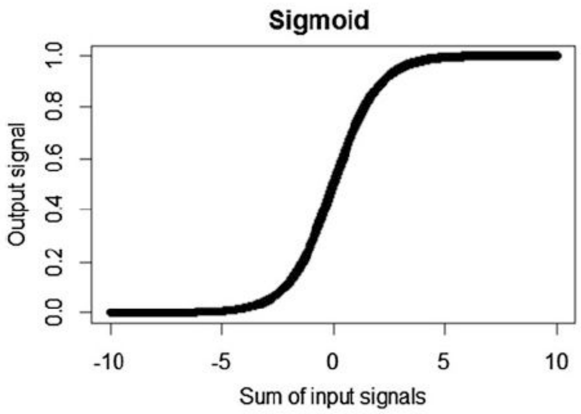
An artificial neural network has a more complex structure than that of a perceptron model.

The network may contain several intermediary layers between its input and output laye Such intermediary layers are called **hidden layers** and the nodes embedded in these layers are called **hidden nodes**.



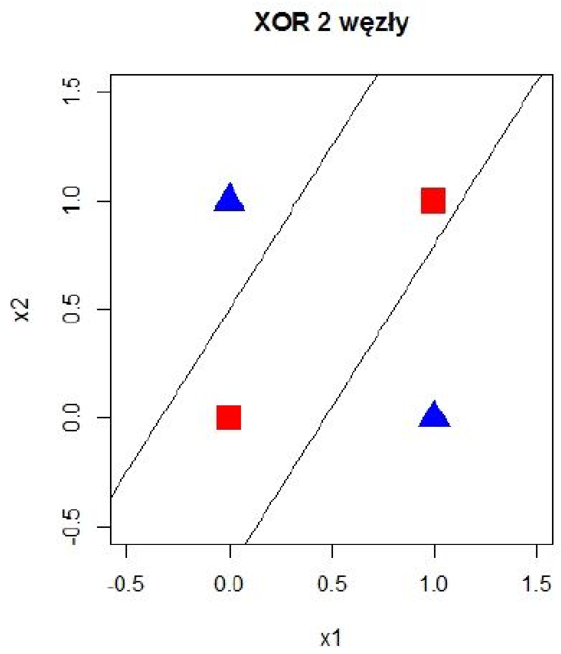
The network may use types of activation functions other than the sign function - linear, sigmoid (logistic), and hyperbolic tangent functions and others.

The most commonly used function is the sigmoid activation function, where $f(x) == \frac{1}{1+e^{-x}}$, where **e** is Euler's natural number, which is also the base of the natur logarithm function.



These activation functions allow the hidden and output nodes to produce output value that are nonlinear in their input parameters.

Let us consider the **XOR problem again**. Because a perceptron can create only one hyperplane, it cannot find the optimal solution. This problem can be solned using a twc layer, feed-forward neural network. Intuitively, we can think of each hidden node as a perceptron that tries to construct one of the two hyperplanes, while the output node simply combines the results of the perceptrons to yield the decision boundary.



The goal of the ANN learning algorithm is to determine a set of weights w that minimize the total sum of souared errors:

$E(w) = \frac{1}{2}\sum_{i=1}^{n}(y_i - \hat{y})^2$

Last modified: środa, 26 maja 2021, 7:43