

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	PIHII (Russia) = 0.207	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 4.102	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

SOI: [1.1/TAS](#) DOI: [10.15863/TAS](#)

**International Scientific Journal
Theoretical & Applied Science**

p-ISSN: 2308-4944 (print) e-ISSN: 2409-0085 (online)

Year: 2018 Issue: 06 Volume: 62

Published: 01.06.2018 <http://T-Science.org>

Sagat Zhunisbekov
prorector, doctor of Science,
Taraz state University named after M.Kh. Dulati

Alexandr Shevtsov
candidate of technical sciences,
member of PILA (USA),
Taraz state University named after M.Kh. Dulati

Nurlan Karymsakov
candidate of technical sciences,
Taraz state University named after M.Kh. Dulati

SECTION 2. Applied mathematics. Mathematical modeling.

DEVELOPMENT OF THE SIMPLEST PERCEPTRON FOR ARTIFICIAL INTELLIGENCE

(Part 1)

Abstract: This study analyzes the existing algorithms for creating artificial intelligence and develops a simple perceptron capable of self-learning on the example of a simple game.

Key words: perceptron, game, delphi.

Language: English

Citation: Zhunisbekov S, Shevtsov A, Karymsakov N (2018) DEVELOPMENT OF THE SIMPLEST PERCEPTRON FOR ARTIFICIAL INTELLIGENCE. ISJ Theoretical & Applied Science, 06 (62): 1-10.

Soi: <http://s-o-i.org/1.1/TAS-06-62-1> Doi: [crossref https://dx.doi.org/10.15863/TAS.2018.06.62.1](https://dx.doi.org/10.15863/TAS.2018.06.62.1)

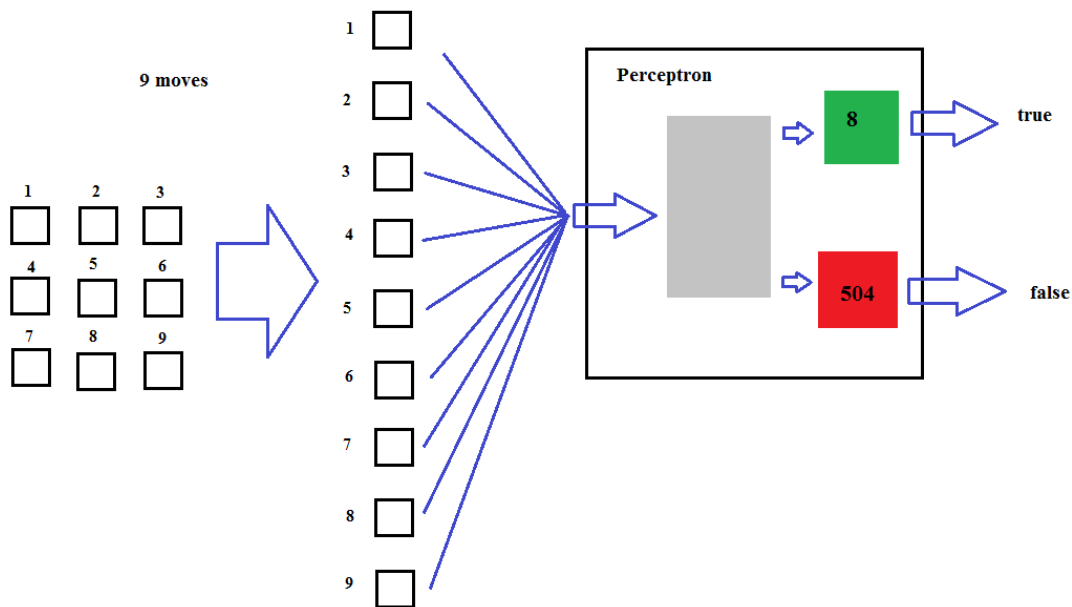
Introduction

Perceptron creates a lot of questions. How does the brain really function? How does he build connections within himself? How does the process of neural network learning happen? Neurons are random number and are connected by accident. You want to build a connection algorithm, after which the model will act expediently. The perceptron (from

the word perception) can be considered as a variant of neural network realization [1].

Materials and Methods

Block diagram of the game with perceptron will be presented in the following form Picture 1.



Picture 1 - Block diagram.

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	PIHHI (Russia) = 0.207	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 4.102	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

The learning algorithm of a neural network:

d_i - is the desired output of some of the i -th effector.

y_i -the current state of the effector.

E - the error between the desired and the real. If it is minimal, the training was successful.

The structure, input and output signals cannot be changed.

E can be minimized by changing w_{ij} .

$$E = \frac{1}{2} \sum_i (y_i - d_i)^2$$

$$EA_i = \frac{\partial E}{\partial y_i} = (y_i - d_i); \quad (1)$$

$$EI_i = \frac{\partial E}{\partial x_i} = \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial x_i} = EA_i \cdot y_i (1 - y_i); \quad (2)$$

$$EW_{ij} = \frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial x_i} \frac{\partial x_i}{\partial w_{ij}} = EI_i \cdot y_j; \quad (3)$$

$$EA_j = \frac{\partial E}{\partial y_j} = \sum_i \frac{\partial E}{\partial x_i} \frac{\partial x_i}{\partial y_j} = \sum_j EI_i \cdot w_{ij}; \quad (4)$$

1-dependence of the error rate on the output signal of the perceptron.

2-dependence of the error rate on the input signal of the perceptron.

3-the dependence of the error rate of change on the weight of the connection.

Back-calculation algorithm

We know what should be on the output and gradually calculate the input from layer to layer by the chain of formulas 1 - 2 - 4 - 2 - 4 - 2 - 4 - : after determining (2) in each layer, you can calculate the

formula (3) ∂w and then w by the formula: $w = w + \partial w$. Implemented as a gradient method.

The task is as follows: find all w_{ij} , that is, adjust the weight of all connections so that the perceptron gave the desired output signal to the corresponding input. To set up (train) a perceptron for a task, it is necessary to implement many iterations. The goal is to reduce error E to zero. As a result, all the best w_{ij} values are found. Learning is exponential. If the error E does not come to zero, it means that the complexity of the perceptron is not enough to teach this example (examples), the number of layers or neurons in the layers should be increased.

The General features of the technology are as follows. The system begins to detect patterns in the input information. The system doesn't know how it learns - it doesn't care about the subject of reasoning. The system can easily complete their education, and relearn.

64 games were submitted for input. As a training, the system had to give an answer to the question – can you win in this game. On games she studied. Then gave the expert image – a game that still she couldn't see. She answered her question exactly.

Thus, the experiment took place in two stages-training (perceptron gave a number of examples) and examination (verification of the degree of training). Each perceptron was trained exponentially. the perceptrons were trained in languages, reading English text, recognition of spoken letters, etc. the Perceptron can be "for underfeeding" examples, but you can "overfeed". The number of perceptrons should be increased hierarchically, until the complexity of the system is equal to the complexity of the problem.

$$\begin{pmatrix} 1 & * & * \\ * & 1 & * \\ * & * & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 & 1 \\ * & * & * \\ * & * & * \end{pmatrix} \quad \begin{pmatrix} * & * & * \\ 1 & 1 & 1 \\ * & * & * \end{pmatrix} \quad \begin{pmatrix} * & * & * \\ * & * & * \\ 1 & 1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} * & * & 1 \\ * & 1 & * \\ 1 & * & * \end{pmatrix} \quad \begin{pmatrix} 1 & * & * \\ 1 & * & * \\ 1 & * & * \end{pmatrix} \quad \begin{pmatrix} * & 1 & * \\ * & 1 & * \\ * & 1 & * \end{pmatrix} \quad \begin{pmatrix} * & * & 1 \\ * & * & 1 \\ * & * & 1 \end{pmatrix}$$

Picture 2 - Winning combination.

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	PIHHI (Russia) = 0.207	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 4.102	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

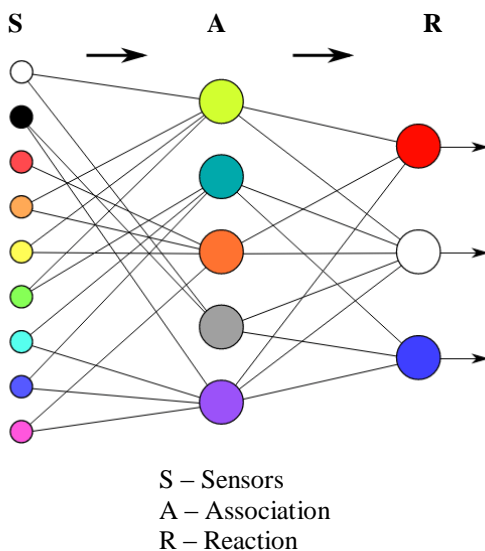
Single-layer perceptron

Similar names. But this is not the same as a perceptron with one hidden layer, although it may seem so. This type of perceptron as an elementary perceptron, too often mean when they talk about the perceptron at all.

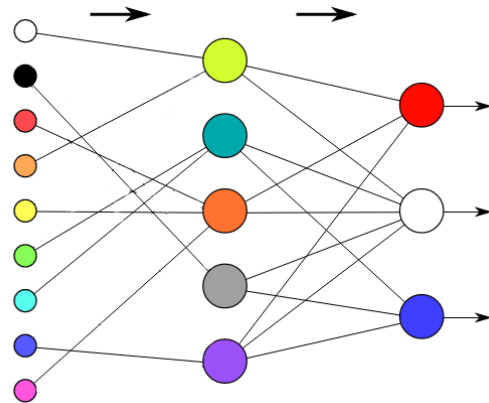
Its key feature is that each S-element uniquely corresponds to one A-element, all S-A connections have a weight equal to +1, and the threshold of elements is 1.

I'll explain. Take a picture of the perceptron in the General sense and convert it into a picture of a single-layer perceptron.

Initially, the perceptron in the General sense looks like this [2-4]:



Based on the key features of a single-layer perceptron sensor can be uniquely associated with only one associative element. Look at the white sensor in the picture (upper left corner). It transmits a signal to the light green (first) and gray (fourth) associative elements. Disorder. The sensor can only transmit a signal to one a-element. Remove the excess communication. The same operation is carried out with other sensors.



Make sure you understand the phrase "each S-element uniquely matches one A-element". This means that each sensor can only transmit a signal to one A-element. However, this statement does not prohibit the situation when several sensors transmit a signal to one A-element, as shown in the picture above (1, 2 and 3 A-elements).

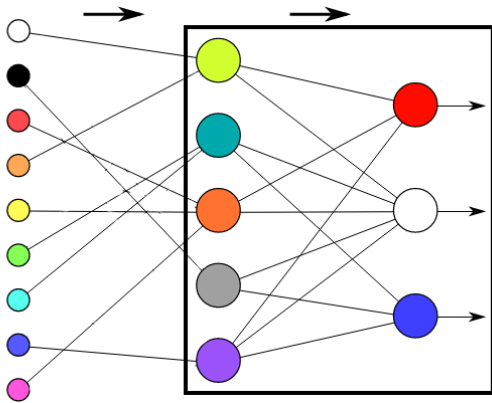
Next, s-a links always have a weight equal to one, and the threshold of A-elements is always +1. On the other hand, we know that sensors can only signal 0 or 1.

Consider the first S-element in the last picture. Let it generate a signal equal to one. The signal passes through The s-A link and does not change, since any number multiplied by 1 is equal to itself. The threshold of any element is equal to 1. Since the sensor produced a signal equal to 1, the a-element was definitely excited. This means that it has a signal equal to 1 (since it can also generate only 1 or 0 at its output). Further, this single signal is multiplied by an arbitrary weight of a-R connection and enters the corresponding R-element, which sums up all the received weighted signals, and if they exceed its threshold, gives +1. Otherwise, the output of this R-element is -1.

Not counting the touch-elements and S-A relations we described the scheme of the artificial neuron. It's no coincidence A single-layer perceptron is indeed an artificial neuron with a slight difference. Unlike an artificial neuron, the single-layer perceptron inputs can take fixed values: 0 or 1. The artificial neuron can be applied to the input of any value.

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	PIHHI (Russia) = 0.207	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 4.102	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	



In the perceptron, R-elements sum the weighted inputs and, if the weighted sum is above a certain threshold, give out 1. Otherwise, the outputs of the R-elements would be -1.

It is easy to guess that this behavior is easily set by the activation function called the single jump function, which we have already considered in Chapter 2. The difference is that the function of a single jump gives 0 if the threshold is not exceeded, and here gives -1, but it is not essential.

Thus it becomes clear that part of the single-layer perceptron (highlighted in black rectangle in the picture above) can be represented as an artificial neuron, but in any case do not confuse these two concepts. First, no one has canceled S-elements, which in the artificial neuron simply do not exist. Secondly, in a single-layer perceptron, S-elements and A-elements can only take fixed values 0 and 1,

whereas in an artificial neuron there are no such restrictions.

A single-layer perceptron is a perceptron, each S-element of which uniquely corresponds to one A-element, s-a connections are always equal to 1, and the threshold of any A-element is equal to 1.

Part of the single-layer perceptron corresponds to the model of an artificial neuron.

A single-layer perceptron can also be an elementary perceptron with only one layer of S, a,R-elements.

Multilayer perceptron

Under a multilayer perceptron understand two different types: multilayer perceptron Rosenblatt and multilayer perceptron Rumelhart.

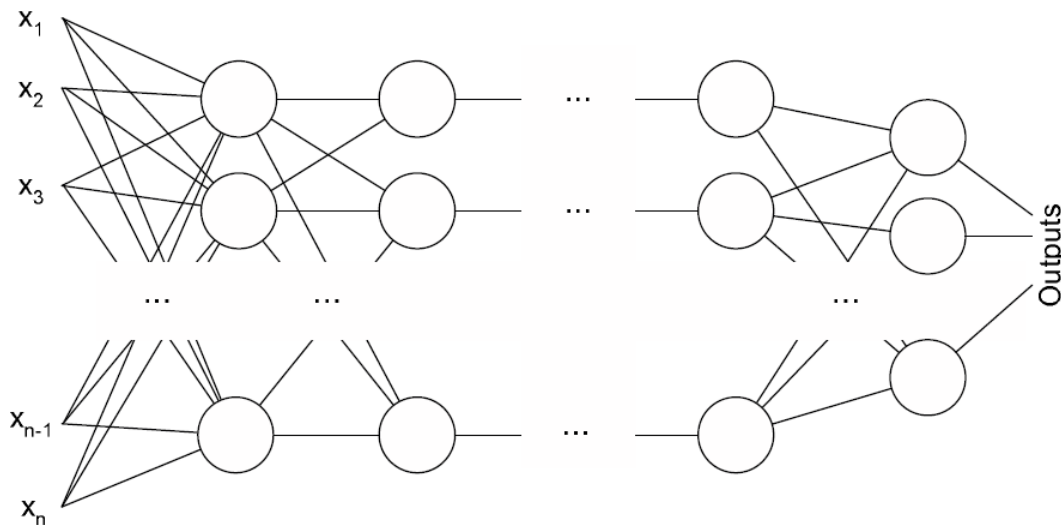
Rosenblatt multilayer perceptron contains more than 1 layer of a-elements.

The multilayer perceptron by Rumelhart is a special case of the multilayer perceptron by Rosenblatt, with two features:

S-A relationships can be of arbitrary weight and learn along with the A-R relations.

Training is performed by a special algorithm, which is called training by the method of back propagation of the error.

This method is the cornerstone of learning all multilayer ins. Thanks to him, the interest in neural networks has been renewed. But we will discuss it in other chapters.



Picture 3 - Multilayer perceptron.

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	PIHHI (Russia) = 0.207	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 4.102	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

Multilayer perceptron **Roseblatt-perceptron**, which has more than 1 layer Of a-elements.

Multi-layer perseptron by **Rumelhart** is a multi-layer perseptron by Rosenblatt, in which s-a connections are also subject to training, as well as the training itself is made by the method of back propagation of the error.

Classification task

Is it possible to classify Boolean functions? Yes, and besides, this problem will perfectly illustrate such a classification.

These are functions from some number of variables. Moreover, both the variables themselves and the values of logical functions can take only fixed (discrete) values: 0 or 1.

We have two binary variables (that is, they can only be 0 or 1). The value of the logical "And" function will be 1 only when the values of both variables are also 1. In all other cases, the value of this logical function is 0.

In order to better understand the principle of the logical function, often use truth tables, where the first two columns have possible combinations of variables, and the third value of the function in this case.

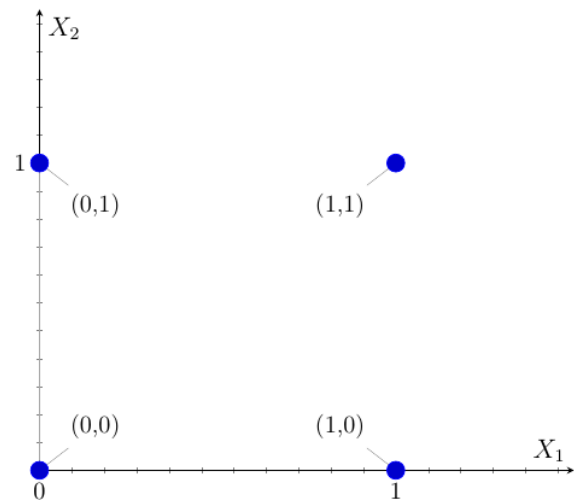
Here is, for example, a truth table for logical And.

x1	x2	function Value
0	0	0
1	0	0
0	1	0
1	1	1

And there is a logical "OR". The truth table for logical OR looks like this.

x1	x2	function Value
0	0	0
1	0	1
0	1	1
1	1	1

Logic functions illustrate the idea of classification very nicely. Any such function takes two arguments as input. By luck points on the plane are given by two numbers (x and y)! But logical functions can only accept discrete arguments (0 or 1). As a result, we obtain that for the image of any logical function on the plane, 4 points (with coordinates (0,0) (1,0) (0,1) (1,1)). This is what it looks like:

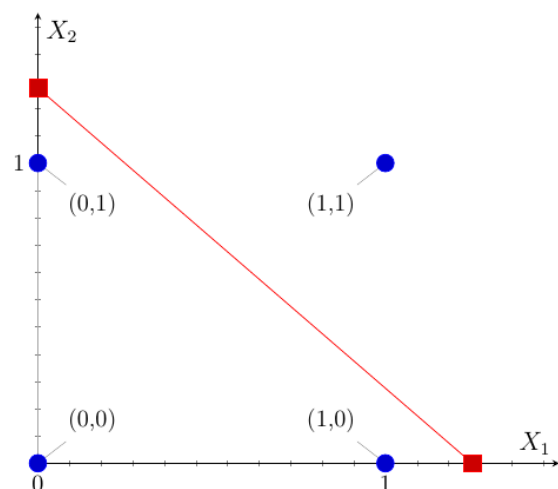


Let's consider the logical function of I. It is equal to zero for any set of input arguments, except for a set of (1,1).

x1	x2	The Logical And
0	0	0
1	0	0
0	1	0
1	1	1

There is a problem of classification: we have 4 points. We need to draw a straight line so that on one side we have points for which the values of the logical And is equal to 1, and on the other, for which this value is 0.

In the case of the logical and this line, for example, you can draw as shown in the figure below. All points below this line result in a 0 value of this function. A single point above this line results in a logical and equal to 1.



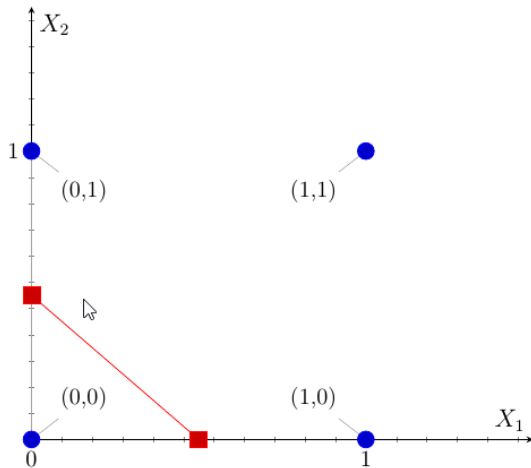
In a similar way behaves in a logical OR with the following truth table:

x1	x2	Logical OR
0	0	0
1	0	1
0	1	1
1	1	1

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	PIHII (Russia) = 0.207	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 4.102	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

For this function, the graphical representation will look like this:



It is easy to notice that this picture is a graphical representation of the logical and, but Vice versa (also one point, but for which the value of the function is 0 and already under the line).

On this we complete the classification tasks. I think now you can imagine that many problems can be solved if you can reformulate them in the form of classification problems.

Now we move to the cornerstone of neural networks — their training. After all, without this property, they do not make any sense.

Training a perceptron

You know what they are and what tasks they can solve. It is time to talk about their training. As you remember, by learning neural network means the process of adjusting the weight coefficients of the connection so that as a result, when the network input of a certain signal, it gave us the correct answer.

Simplify to the limit

Let's start training our neural networks from the simplest case. To do this, we will greatly simplify the already simple single-layer perceptron with one hidden layer:

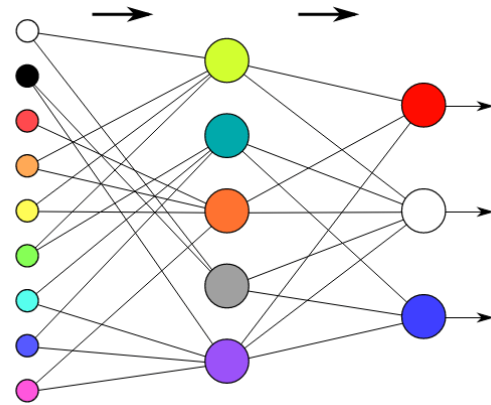
1. We will assume that its A-R connections can only take integer values (... , -2, -1, 0, 1, 2, ...).

2. Moreover, each A-element can have only one S-element.

3. And we will only have 1 R-element.

In words, such a large number of simplifications can look difficult. Let us explain what was said in the diagram. Take the already used image of a single-layer perceptron with one hidden layer and convert it.

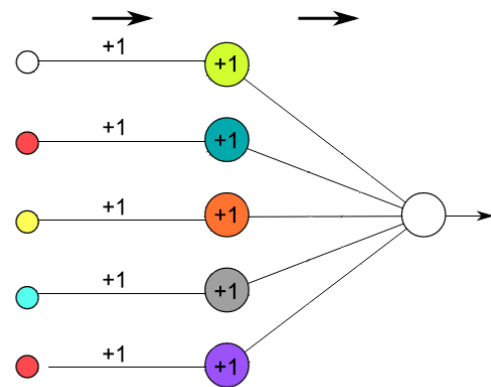
Initially, we have the following perceptron.



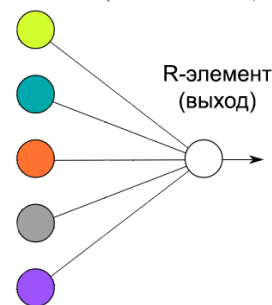
We have to simplify it. Now A-elements can be connected to only one S-element. We're taking out all the extra connections.

In the picture above 3 R-element. Only one left. S-A weights and thresholds of a elements are now +1. Note this in the figure.

As a result, we get the following picture.



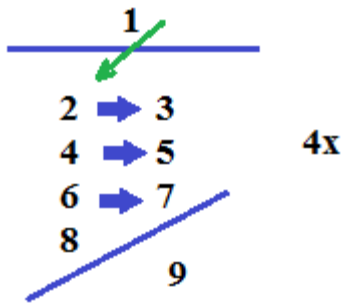
However, it turns out that we have a layer of A-elements does not perform any functions. It is equivalent to S-layer. Therefore, we are making the following simplification. Throw away the sensor layer. Now the role of sensors we will perform associative elements (or Vice versa, no difference).



So, we just made a very simple single-layer perceptron with one hidden layer.

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	PIHHI (Russia) = 0.207	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 4.102	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	



Conclusion

Each turn is a total of nine squares, with only two possible colors. As mentioned in the previous section, the white square is 0, and the black square is 1. So our four moves from 1 to 4 in string format will look like this:

To record every game we have a used 4 stroke with 9 characters each. Now remove all line breaks to get one long line of 36 characters for each move from 1 to 4.

1 – 001001001001001

.....

4 – 111101111001111

Numbers in this string format can already be used to work with the neural network.

Learning algorithm

Finally we got to the main thing: how to train the network. In General, the process is clear. We will randomly select a number and run it through the network, modifying its weight. But how to modify them?

We know that the importance of certain inputs (in our case – S-elements) is given by weights that connect them to the R-element. Thus, the more strongly influenced some weight of the connection on the result, the more it is necessary to change it.

Therefore, we must consider the following important points:

If our neural network correctly recognized / rejected move 1, then we do not do anything (everything is great!).

If the neural network made a mistake and recognized the wrong move as 1, then we must punish it—we reduce the weight of those connections through which the signal has passed. In other words, the weights associated with the excited inputs are reduced.

If the neural network made a mistake and did not recognize the move 1, then we must increase all the weights through which the signal passed. Thus if we say network such and so and related inputs – correct.

Now we write down the learning algorithm, which we will implement in the program:

Submit to the inputs of the neural network number in string format.

If the number detected/rejected by the right, then go to step 1.

If the network has made a mistake and recognized the wrong number, then subtract one from all the links associated with the excited S-elements.

If the network made a mistake and rejected move 1, then add one to all links associated with the initiated S-elements.

Why in the algorithm we add or subtract 1. In fact, this value can be set any. It is clear that this value affects the effectiveness of training.

References:

- (2018) Back-calculation algorithm <https://intellect.ml/perseptron-5281>
- (2018) perseptrons <http://neuralnet.info/chapter/перцептронны/>
- Bryukhomitskiy, Yu. A. (2005) Neyrosetevye modeli dlya sistem informatsionnoy bezopasnosti: Uchebnoe posobie. — Taganrog: Izd-vo TRTU, 2005. — 160 p.
- Mak-Kallok, U. S., Pitts, V. (1956) Logicheskoe ischislenie idey, odnosyashchikhsya k nervnoy aktivnosti = A logical calculus of the ideas immanent in

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	PIHHI (Russia) = 0.207	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 4.102	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

- nervous activity // Avtomaty : Sb. — M., 1956. — p. 363—384.
5. Minskiy, M., Peypert, S. (1971) Perseptrony = Perceptrons. — M.: Mir, 1971. — 261 p.
 6. Rozenblatt, F. (1965) Printsipy neyrodinamiki: Pertseptrony i teoriya mekhanizmov mozga = Principles of Neurodynamic: Perceptrons and the Theory of Brain Mechanisms. — M.: Mir, 1965. — 480 p.
 7. Uossermen, F. (1992) Neyrokomp'yuternaya tekhnika: Teoriya i praktika = Neural Computing. Theory and Practice. — M.: Mir, 1992. — 240 p. — ISBN 5-03-002115-9.
 8. Khaykin, S. (2006) Neyronnye seti: Polnyy kurs = Neural Networks: A Comprehensive Foundation. — 2-e izd. — M.: «Vil'yams», 2006. — 1104 p. — ISBN 0-13-273350-1.
 9. Yakovlev S.S. (2004) Sistema raspoznavaniya dvizhushchikh ob"ektov na baze iskusstvennykh neyronnykh setey // ITK NANB. — Minsk, 2004. — p. 230—234.
 10. Kussul E., Baidyk T., Kasatkina L., Lukovich V. (2001) Pertseptrony Rozenblatta dlya raspoznavaniya rukopisnykh tsifr = Rosenblatt Perceptrons for Handwritten Digit Recognition // IEEE. — 2001. — p. 1516—1520. — ISBN 0-7803-7044-9. (angl.)
 11. Stormo G. D., Schneider T. D., Gold L., Ehrenfeucht A. (1982) Ispol'zovanie pertseptrona dlya vydeleniya saytov initsiatsii v E. coli = Use of the 'Perceptron' algorithm to distinguish transational initiation sites in E. coli // Nucleic Acids Research. — 1982. -p. 2997—3011. (angl.)

Annex 1

unit Unit1;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics, Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.ButtonGroup;

type

```
TForm1 = class(TForm)
  ButtonGroup1: TButtonGroup;
  Button1: TButton;
  Memo1: TMemo;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  procedure Button1Click(Sender: TObject);
  procedure ButtonGroup1Items0Click(Sender: TObject);
  procedure ButtonGroup1Items1Click(Sender: TObject);
  procedure ButtonGroup1Items2Click(Sender: TObject);
  procedure ButtonGroup1Items3Click(Sender: TObject);
  procedure ButtonGroup1Items4Click(Sender: TObject);
  procedure ButtonGroup1Items5Click(Sender: TObject);
  procedure ButtonGroup1Items6Click(Sender: TObject);
  procedure ButtonGroup1Items7Click(Sender: TObject);
  procedure ButtonGroup1Items8Click(Sender: TObject);
```

private

{ Private declarations }

public

{ Public declarations }

end;

var



Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	PIHHI (Russia) = 0.207	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 4.102	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

```
Form1: TForm1;
a:array[1..9]of integer;
t:integer;

implementation

{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
var i:integer;
begin
label1.Visible:=false;label2.Visible:=false;label3.Visible:=false;label4.Visible:=false;
for i:=1 to 9 do begin
    ButtonGroup1.Items[i-1].Caption:="";
    a[i]:=0;
end;
end;

function check():boolean;
begin
if
(a[1]+a[2]+a[3]=3) or
(a[4]+a[5]+a[6]=3) or
(a[7]+a[8]+a[9]=3) or
(a[1]+a[4]+a[7]=3) or
(a[2]+a[5]+a[8]=3) or
(a[3]+a[6]+a[9]=3) or
(a[1]+a[5]+a[9]=3) or
(a[3]+a[5]+a[7]=3) then begin
    form1.label1.Visible:=true;
    form1.label2.Visible:=true;
end;
if
(a[1]+a[2]+a[3]=-3) or
(a[4]+a[5]+a[6]=-3) or
(a[7]+a[8]+a[9]=-3) or
(a[1]+a[4]+a[7]=-3) or
(a[2]+a[5]+a[8]=-3) or
(a[3]+a[6]+a[9]=-3) or
(a[1]+a[5]+a[9]=-3) or
(a[3]+a[5]+a[7]=-3) then begin
    form1.label3.Visible:=true;
    form1.label4.Visible:=true;
end;
end;

function xodcomp():boolean;
begin
k:=false;
s:=random
a[t]:=1;
check;
end;

function upd(t:integer):boolean;
begin
form1.ButtonGroup1.Items[t-1].Caption:=' X';
```

Impact Factor:

ISRA (India) = 1.344	SIS (USA) = 0.912	ICV (Poland) = 6.630
ISI (Dubai, UAE) = 0.829	PIHHI (Russia) = 0.207	PIF (India) = 1.940
GIF (Australia) = 0.564	ESJI (KZ) = 4.102	IBI (India) = 4.260
JIF = 1.500	SJIF (Morocco) = 2.031	

```
a[t]:=1;  
check;  
end;
```

```
procedure TForm1.ButtonGroup1Items0Click(Sender: TObject);begin upd(1); end;  
procedure TForm1.ButtonGroup1Items1Click(Sender: TObject);begin upd(2); end;  
procedure TForm1.ButtonGroup1Items2Click(Sender: TObject);begin upd(3); end;  
procedure TForm1.ButtonGroup1Items3Click(Sender: TObject);begin upd(4); end;  
procedure TForm1.ButtonGroup1Items4Click(Sender: TObject);begin upd(5); end;  
procedure TForm1.ButtonGroup1Items5Click(Sender: TObject);begin upd(6); end;  
procedure TForm1.ButtonGroup1Items6Click(Sender: TObject);begin upd(7); end;  
procedure TForm1.ButtonGroup1Items7Click(Sender: TObject);begin upd(8); end;  
procedure TForm1.ButtonGroup1Items8Click(Sender: TObject);begin upd(9); end;
```

```
end.
```

