Assignment № 1
The delta rule

Intelligent Data Processing Laboratory

Maciej Socha, 239709 № and Michał Darowny, 239644 №

27.10.2021

1 Main goal

Our main goal was to implement and train with delta rule a linear neuron. To train neuron we used a simple and multiple training patterns.

2 Theoretical background

Every single linear neuron consist the following parts [1]:

1. Vector of neurons inputs, $x = [x_1, x_2, x_3, ..., x_n] \in R^N$
2. Vector of weights, $w = [w_1, w_2, w_3, ..., w_n] \in R^N$
3. Output value, calculated: $y = \sum w_i x_i$

In Figure 1 there is a graphical representation of linear neuron.

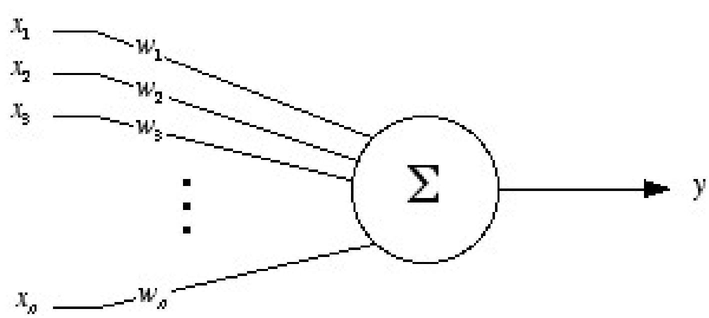

Figure 1: Linear neuron model

The process of training can be carried out using single or multiple training patterns. In single our training set Ω contains only one training pattern – vector of inputs and desired output. With multiple training patterns Ω contains many pairs of input vectors and desired outputs.

3 Experiments and results

We carry out experiments using single and multiple training patterns.

3.1 Experiment № 1

3.1.1 Description

In this experiment we measured how changing of training step affects to results. We used a neuron with two inputs. Our training pattern was: $\Omega = [10, 10], 20$. Number of epochs was 100.

3.1.2 Results

Table 1 presents calculated output for input vector $x = [10, 10]$.

Training step	Run 1	Run 2	Run 3	Run 4	Run 5
0.1	-3.85e+128	-1.29e+129	-1.13e+129	-6.20e+128	-3.07e+128
0.01	13.57	12.63	8.93	4.95	9.33
0.001	19.99	19.99	19.99	19.99	19.99

Table 1: Changing training step

As we can se, changing training step to lower number has positive impact to results. With $\eta = 0.001$ neuron calculates output properly.

3.2 Experiment № 2

3.2.1 Description

In this experiment we measured how changing number of epochs affects to results. We used a neuron with two inputs. Our training pattern was: $\Omega = \{[10, 10], 20\}$. Training step was set to 0.001.

3.2.2 Results

Table 2 presents calculated output for input vector $x = [10, 10]$.

Number of epochs	Run 1	Run 2	Run 3	Run 4	Run 5
5	17.455	16.574	16.883	16.146	17.452
25	19.954	19.942	19.943	19.928	19.949
50	19.999	19.999	19.999	19.999	19.999

Table 2: Changing number of epochs

Higher number of epochs can help produce better results. In this example just 25 epochs give a quite good result.

3.3 Experiment № 3

3.3.1 Description

We show how neuron works if we change number of inputs. We set training step to 0.001 and number of epochs to 50. In every run we used a training set:

1. Run - $\Omega = \{[5, 10], 15\}$

2. Run - $\Omega = \{[5, 10, 15], 30\}$
3. Run - $\Omega = \{[5, 10, 15, 20], 50\}$
4. Run - $\Omega = \{[5, 10, 15, 20, 25], 75\}$

3.3.2 Results

Run	Output
1	14.995119228763564
2	29.999999992876774
3	50.0
4	74.99999999999999

Table 3: Changing number of inputs

As table 3 shown, our implementation of neuron works with different numbers of input. The parameters we set work well with all of those training sets.

3.4 Experiment № 4

3.4.1 Description

We set a number of epochs to 500 and training step to 0.0001. The neuron has 5 inputs. In this example we use a training set with three patterns $\Omega = \{[[18, 16, 17, 19, 13], [21, 28, 23, 25, 24], [10, 20, 19, 28, 12]], [16.6, 24.2, 17.8]\}$

3.4.2 Results

Input vector	Calculated output	Desired output
[18, 16, 17, 19, 13]	16.546514148348194	16.6
[21, 28, 23, 25, 24]	24.210605977811213	24.2
[10, 20, 19, 28, 12]	17.81649840767928	17.8
[31, 39, 33, 32, 38]	34.31044976790747	34.6
[39, 15, 27, 15, 34]	25.209572373860027	26
[45, 48, 43, 44, 41]	43.9761854716063	44.2
[11, 50, 48, 35, 24]	26.490185478860788	33.6

Table 4: 3 training patterns

3.5 Experiment № 5

3.5.1 Description

We set a number of epochs to 500 and training step to 0.0001. The neuron has 5 inputs. In this example we use a training set with three patterns $\Omega = \{[[18, 16, 17, 19, 13], [21, 28, 23, 25, 24], [10, 20, 19, 28, 12], [31, 39, 33, 32, 38], [39, 15, 27, 15, 34]], [16.6, 24.2, 17.8, 34.6, 26]\}$

3.5.2 Results

Input vector	Calculated output	Desired output
[18, 16, 17, 19, 13]	17.04363976937711	16.6
[21, 28, 23, 25, 24]	24.308869335485923	24.2
[10, 20, 19, 28, 12]	17.587225164066165	17.8
[31, 39, 33, 32, 38]	34.67210754010774	34.6
[39, 15, 27, 15, 34]	25.91407388836371	26
[45, 48, 43, 44, 41]	45.32044001263306	44.2
[11, 50, 48, 35, 24]	41.35612400604051	33.6

Table 5: 5 training patterns

3.6 Experiment № 6

3.6.1 Description

We set a number of epochs to 500 and training step to 0.0001. The neuron has 5 inputs. In this example we use a training set with three patterns $\Omega = \{[[18, 16, 17, 19, 13], [21, 28, 23, 25, 24], [10, 20, 19, 28, 12], [31, 39, 33, 32, 38], [39, 15, 27, 15, 34], [45, 48, 43, 44, 41], [11, 50, 48, 35, 24]], [16.6, 24.2, 17.8, 34.6, 26, 44.2, 33.6]\}$

3.6.2 Results

Input vector	Calculated output	Desired output
[18, 16, 17, 19, 13]	16.5911648193495	16.6
[21, 28, 23, 25, 24]	24.175929550231153	24.2
[10, 20, 19, 28, 12]	17.722949498661375	17.8
[31, 39, 33, 32, 38]	34.55228799663332	34.6
[39, 15, 27, 15, 34]	25.927789977519232	26
[45, 48, 43, 44, 41]	44.20265384994138	44.2
[11, 50, 48, 35, 24]	33.59956763752777	33.6

Table 6: 7 training patterns

4 Summary and conclusions

- Higher number of epochs has positive impact on results
- Lower training step help to give a more accurate results
- More training patterns give us a better results. Results for testing patterns which was not used to lear, are much worse.

References

[1] Labolatory instruction № 1 the delta rule. *Institute of Information Technology, Technical University of Lodz.*