

SPRAWOZDANIE

Zajęcia: Analiza procesów uczenia

Prowadzący: prof. dr hab. inż. Vasyl Martsenyuk

Laboratorium Nr 3 Data 31.03.2023 Temat: Użycie sztucznych sieci neuronowych Wariant 4	Maksymilian Grygiel Informatyka II stopień, stacjonarne, Semestr I, gr.1a
---	--

Link do repozytorium: <https://github.com/Maksiolo20/APU>

Zadania:

Zadanie 1. Zadanie dotyczy modelowania funkcji matematycznych za pomocą sztucznej sieci neuronowej używając paczkę neuralnet. Rozważamy zmienną niezależną x . Celem jest uzyskanie sieci neuronowej (zmieniając zarówno ilość warstw ukrytych jak i ilość neuronów) wypełniające warunek $\text{Error} < 0.01$.

Wariant zadania: 4. $f(x) = \log x^2$, $x \in [1; 10]$

```
> install.packages("neuralnet")
> library(neuralnet)
> input <- as.data.frame(runif(100,min=1,max=10))
> output <- log(input^2)
> trainingdata <- cbind(input,output)
> colnames(trainingdata)<-c("Wejście,Wyjście")
> net.sqrt <- neuralnet(Wyjście~Wejście,trainingdata,hidden=7,threshold=0.01,stepmax=1e7)
> print(net.sqrt)
```

Fragment wyniku powyższej funkcji print (całość wyniku dostępna w repozytorium, w pliku: z1_print:

```
$call  
neuralnet(formula = Wyjście ~ Wejście, data = trainingdata, hidden = 7,  
  threshold = 0.01, stepmax = 1e+07)
```

```
$response  
      Wyjście  
1  85.277715  
2  11.783444  
3   4.888762  
4   2.675981  
5  24.736743  
6  16.668417  
7  52.269785  
8  22.915143  
9  89.101533  
10 52.689216  
11 32.167545|  
12  4.640449  
13 31.769322  
14 21.882288  
15 74.587982  
16 69.050277  
17 58.990672  
18  1.600244  
19 19.002536  
20  7.819909  
21 30.966970  
22 10.172166  
23 68.796644
```

```
91 64.826439
92 22.005558
93 24.442849
94 2.464001
95 42.688949
96 89.262339
97 14.706760
98 64.529275
99 24.867242
100 40.624071
```

```
$covariate
```

```
[1,] 9.234588
[2,] 3.427972
[3,] 2.185484
[4,] 1.571035
[5,] 4.972909
[6,] 4.080627
[7,] 7.229734
[8,] 4.786105
[9,] 9.439357
[10,] 7.258685
```

```
[99,] 4.986021  
[100,] 6.373568
```

```
$model.list  
$model.list$response  
[1] "Wyjście"
```

```
$model.list$variables  
[1] "Wejście"
```

```
$err.fct  
function (x, y)  
{  
  1/2 * (y - x)^2  
}  
<bytecode: 0x000001565d2ba258>  
<environment: 0x000001565a5dc9d8>  
attr(,"type")  
[1] "sse"
```

```
$act.fct  
function (x)  
{  
  1/(1 + exp(-x))  
}  
<bytecode: 0x000001565d2b5458>  
<environment: 0x000001565a5dce70>  
attr(,"type")  
[1] "logistic"
```

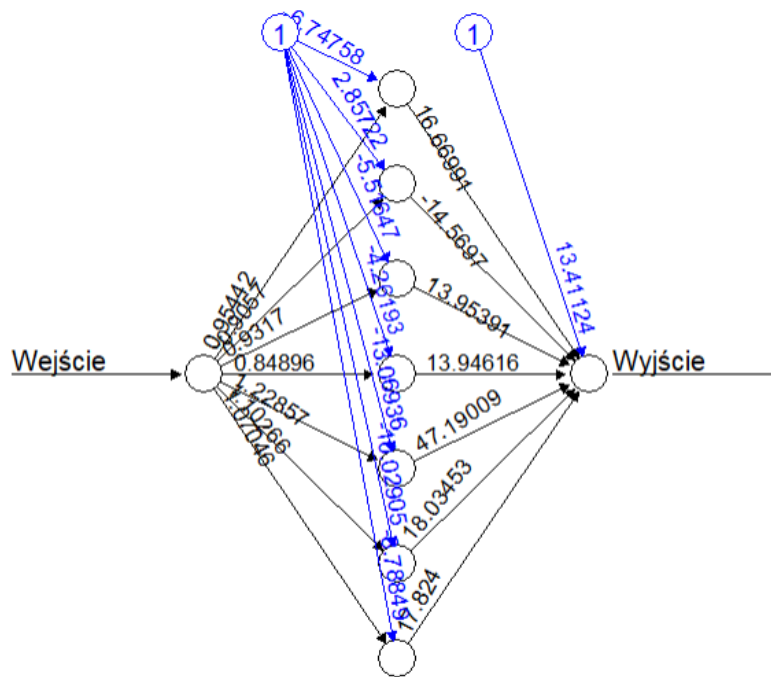
```
$linear.output  
[1] TRUE
```

```
$data  
      Wejście  Wyjście  
1  9.234588 85.277715  
2  3.427972 11.783444  
3  2.185484  4.888762  
4  1.571035  2.675981  
5  4.972909 24.736743
```

Kontynuacja kodu w R:

```
> plot(net.sqr,rep="best")
```

Wynik funkcji plot – wykres przedstawiający sieć neuronową:



Error: 0.004885 Steps: 38132

Kontynuacja kodu w R:

```
> testdata <- as.data.frame(runif(300,min=1,max=20))  
> net.results <- compute(net.sqr, testdata)  
> print(net.results$net.result)
```

Fragment powyższej funkcji print. Tak jak w poprzednim przypadku, całość dostępna jest w repozytorium (z1print_2):

```

|           [,1]
[1,] 138.521819
[2,] 140.840143
[3,] 120.240929
[4,] 140.983858
[5,]  78.220580
[6,] 122.266307
[7,]  31.521218
[8,] 140.858524
[9,] 140.455188
[10,] 119.534246
[11,]  82.203482
[12,]  9.911398
[13,] 35.502518
[14,] 77.659389
[15,] 141.028880
[16,]  2.225974
[17,] 35.744294
[18,] 88.991730
[19,] 124.937234
[20,]  5.543766
[21,] 84.311100
[22,] 141.007463
[23,]  4.298969
[24,] 141.013584
[25,] 11.952532
[26,] 141.023088
[27,]  2.782913
[28,]  2.817876

[291,] 108.545985
[292,] 141.028100
[293,] 141.025507
[294,] 135.742507
[295,]  59.309552
[296,] 140.995329
[297,] 140.985894
[298,]  32.543606
[299,]  82.430420
[300,]  18.793706

```

Kontynuacja kodu w R:

```

> cleanoutput <- cbind(testdata,log(testdata^2),as.data.frame((net.results$net.result)))
> colnames(cleanoutput) <- c("Wejście", "Oczekiwane Wyjście", "Wyjście sieci neuronowej")
> print(cleanoutput)

> save.image("C:/Users/MaksioloLaptop/Desktop/mgr/APU/lab3/Lab3_1.RData")

```

Ostatnia funkcja print – z1print_3 na repozytorium:

	Wejście	Oczekiwane	Wyjście	Wyjście sieci neuronowej
1	13.125108		5.1490541	138.521819
2	15.312502		5.4573392	140.840143
3	11.063065		4.8072241	120.240929
4	16.508486		5.6077491	140.983858
5	8.844167		4.3595162	78.220580
6	11.187608		4.8296136	122.266307
7	5.614115		3.4505679	31.521218
8	15.398324		5.4685174	140.858524
9	14.379318		5.3315818	140.455188
10	11.021171		4.7996362	119.534246
11	9.066781		4.4092346	82.203482
12	3.143142		2.2904457	9.911398
13	5.957623		3.5693433	35.502518
14	8.812350		4.3523083	77.659389
15	19.831919		5.9745854	141.028880
16	1.401548		0.6751552	2.225974
17	5.977853		3.5761229	35.744294
18	9.433700		4.4885767	88.991730
19	11.364755		4.8610337	124.937234
20	2.336591		1.6973861	5.543766
21	9.182356		4.4345676	84.311100
22	17.119449		5.6804303	141.007463
23	2.040830		1.4267137	4.298969
24	17.391467		5.7119594	141.013584
25	3.452484		2.4781882	11.952532
281	9.567837		4.5168143	91.544427
282	2.566560		1.8851328	6.644219
283	12.293158		5.0180857	134.648702
284	17.501704		5.7245965	141.015556
285	3.183917		2.3162247	10.169247
286	5.815897		3.5211899	33.831345
287	16.267569		5.5783469	140.968708
288	6.729828		3.8130993	45.298080
289	17.145175		5.6834336	141.008130
290	5.751661		3.4989772	33.087063
291	10.430179		4.6894068	108.545985
292	19.312959		5.9215527	141.028100
293	18.523859		5.8381192	141.025507
294	12.465573		5.0459413	135.742507
295	7.701814		4.0829117	59.309552
296	16.751608		5.6369885	140.995329
297	16.546813		5.6123871	140.985894
298	5.704300		3.4824406	32.543606
299	9.079299		4.4119939	82.430420
300	4.332693		2.9323786	18.793706

Zadanie 2

Zadanie dotyczy prognozowania ceny urządzeń RTV AGD (error ≤ 100 zł), określonych na Zajęciu 1. Używając metody sztucznych sieci neuronowych opracować plik w języku R z wykorzystaniem paczki neuralnet.

```
> library("neuralnet")
> cena1 <- log(lodowki$cena)
> cena1
```

```
[1] 7.549083 7.494986 7.243513 6.683361 8.342602 8.455105 7.437795 7.649216
[9] 7.377134 7.089243 6.906755 7.862882 8.160232 7.695758 8.318498
```

```
> lodowki <- lodowki[,-5]
> lodowki <- lodowki[,-1]
> lodowki
```

Fragment wywołanych lodówek:

	nazwa	pojemnosc_uzytkowa_chlodziarki	
1	Whirlpool_W7_9210_KAQUA	237	
2	Samsung_RB37J5000SA	269	
3	Sharp_SJBA05DMXW1EU	194	
4	MPM_217CZ19	171	
5	LG_GSJ361DIDV	394	
6	Samsung_RS62R50412C	418	
7	Beko_RCNA406I30ZXB	253	
8	Samsung_RB33N341MSS	217	
9	Samsung_RB29FSRNSA	192	
10	Indesit_LR8S1K	228	
11	Amica_FK2004	109	
12	Whirlpool_W4D7AAAXC	292	
13	Samsung_RS50N3513SA	357	
14	Samsung_RB37J501MB1	255	
15	Samsung_RS50N3913BC	357	

	pojemnosc_uzytkowa_zamrazarki	cena
1	104	1899
2	98	1799
3	70	1399
4	41	799
5	197	4199
6	229	4699
7	109	1699
8	98	2099
9	98	1599
10	111	1199
11	48	999
12	110	2599

Kontynuacja kodu R:

```
> pojemnosc_chlodziarki1 <- log(lodowki$pojemnosc_uzytkowa_chlodziarki)
> pojemnosc_chlodziarki1
```

```
[1] 5.468060 5.594711 5.267858 5.141664 5.976351 6.035481 5.533389 5.379897 5.257495 5.429346 4.691348
[12] 5.676754 5.877736 5.541264 5.877736
```



```
> pojemnosc_zamrazarki1 <- log(lodowki$pojemnosc_uzytkowa_zamrazarki)
> pojemnosc_zamrazarki1

[1] 4.644391 4.584967 4.248495 3.713572 5.283204 5.433722 4.691348 4.584967 4.584967 4.709530 3.871201
[12] 4.700480 4.969813 4.584967 4.969813

> lodowki <- cbind(lodowki, cena1, pojemnosc_chlodziarki1, pojemnosc_zamrazarki1)
> lodowki
```

Fragment z lodówkami:

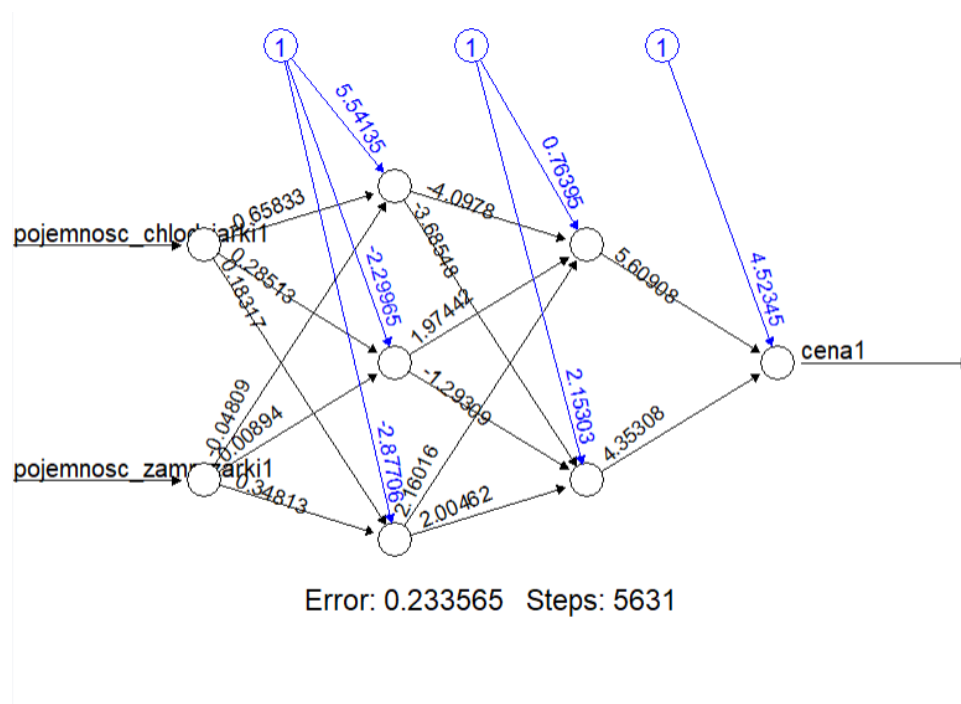
	nazwa	pojemnosc_uzytkowa_chlodziarki	pojemnosc_uzytkowa_zamrazarki	cena	cena1
1	Whirlpool_W7_9210_KAQUA	237	104	1899	7.549083
2	Samsung_RB37J5000SA	269	98	1799	7.494986
3	Sharp_SJBA05DMXW1EU	194	70	1399	7.243513
4	MPM_217CZ19	171	41	799	6.683361
5	LG_GS7361DDV	394	197	4199	8.342602
6	Samsung_RS62R50412C	418	229	4699	8.455105
7	Beko_RCNA406130ZXB	253	109	1699	7.437795
8	Samsung_RB33N341MSS	217	98	2099	7.649216
9	Samsung_RB29FSRNSA	192	98	1599	7.377134
10	Indesit_LR8S1K	228	111	1199	7.089243
11	Amica_FK2004	109	48	999	6.906755
12	Whirlpool_W4D7AAXC	292	110	2599	7.862882
13	Samsung_RS50N3513SA	357	144	3499	8.160232
14	Samsung_RB37J501MB1	255	98	2199	7.695758
15	Samsung_RS50N3913BC	357	144	4099	8.318498

	pojemnosc_chlodziarki1	pojemnosc_zamrazarki1
1	5.468060	4.644391
2	5.594711	4.584967
3	5.267858	4.248495
4	5.141664	3.713572
5	5.976351	5.283204
6	6.035481	5.433722
7	5.533389	4.691348
8	5.379897	4.584967
9	5.257495	4.584967
10	5.429346	4.709530
11	4.691348	3.871201
12	5.676754	4.700480

```
> net.price <- neuralnet(cena1~pojemnosc_chlodziarki1+pojemnosc_zamrazarki1,lodowki,hidden=c(3,2),threshold = 0.01)
> plot(net.price)

> save.image("C:/Users/MaksioloLaptop/Desktop/mgr/APU/lab3/zadanie2/Lab3_2.RData")
```

Poniżej podgląd plot'a sieci neuronowej na lodówkach:



Wnioski:

Język R może posłużyć do przeprowadzania modelowania funkcji matematycznych za pomocą sztucznej sieci neuronowej. Aby to osiągnąć, trzeba skorzystać z funkcji paczki neuralnet. W zadaniach laboratorium uzyskano sieci neuronowe spełniające odpowiednie warunki – na przykład warunek $\text{error} < 0.01$.