

SPRAWOZDANIE

Zajęcia: Analiza Procesów Ucznienia

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium 5

Data 31.03.2023r.

Temat: "Użycie sztucznych sieci
neuronowych"

Wariant 5

Jarosław Waliczek
Informatyka II stopień,
stacjonarne
1 semestr

1. Polecenie

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) wypełniającej warunek $Error < 0.01$. Sprawozdania w postaci pliku R oraz obrazu sieci neuronowej zachować w zdalnym repozytorium (np Github), link na który wysłać w mailu z tematem **APU_4_Gr_numer_grupy** na adres mailowy **vmart-senyuk@ath.bielsko.pl**

$$5. f(x) = x^2 + e^{-x}, \quad x \in [1; 10]$$

Zadanie 2. Zadanie dotyczy prognozowania ceny urządzeń RTV AGD ($error \leq 100$ zł), określonych na Zajęciu 1. Używając metody sztucznych sieci neuronowych opracować plik w języku R z wykorzystaniem paczki *neuralnet*. Sprawozdania w postaci pliku R, obrazu sieci neuronowej oraz wyników z konsoli (dowolny plik tekstowy) zachować w zdalnym repozytorium (np Github) link na który wysłać w mailu z tematem **APU_4_Gr_numer_grupy** na adres mailowy **vmartsenyuk@ath.bielsko.pl**

2. Wprowadzane dane

Jako dane treningowe została stworzona ramka danych – wektor *output* to 100 liczb losowych z rozkładu jednostajnego z przedziału 1 do 10; natomiast wektor *input* to wartości funkcji $f(x)$ dla wektora x

3. Wykorzystane komendy

Zadanie 1.

```
install.packages("neuralnet")
library(neuralnet)

#x ∈ [1; 10]
input <- as.data.frame(runif(100, min=1, max=10))
#f(x) = x^2 + e^-x
output <- input^2 + exp(1)^(-input)
#Połączenie danych wejściowych i wyjściowych
trainingdata <- cbind(input,output)
colnames(trainingdata) <- c("Wejscie","Wyjscie")
#Trenowanie sieci neuronowej
net.sqrt <- neuralnet(Wyjscie~Wejscie,trainingdata, hidden=7, threshold=0.01,
stepmax=1e7)

print(net.sqrt)
```

```
plot(net.sqrt, rep = "best")
```

```
#Prognozowanie z pomocą sieci neuronowej  
testdata <- as.data.frame(runif(300, min=1, max=10))  
net.results <- compute(net.sqrt, testdata)  
print(net.results$net.result)
```

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

Zadanie 2

```
df <- read.csv("C:/Users/jaro9/OneDrive/Desktop/apu/zad3/lodowki.csv")  
pojemnosc <- df[["pojemnosc"]]  
cena <- df[["cena"]]
```

```
compare.trainingdata <- cbind(pojemnosc, cena)  
scaled.pojemnosc <- as.data.frame(scale(pojemnosc))  
trainingdata <- cbind(scaled.pojemnosc, cena)
```

```
colnames(trainingdata) <- c("Pojemnosc", "Cena")  
#(error  $\leq 100$  z 1)  
net.price <- neuralnet(Cena~Pojemnosc, trainingdata, hidden=c(7,1), threshold=<= 100, lifesign <- "full")  
plot(net.price)
```

```
testdata <- data.frame(c(20,130))  
scaled.testdata <- as.data.frame(scale(testdata))
```

```
net.results <- compute(net.price, scaled.testdata)  
fixed_cena <- cbind(testdata, as.data.frame(net.results$net.result))  
colnames(fixed_cena) <- c("Pojemnosc", "Cena")
```

```
print(fixed_cena)
```

4. Wynik działania

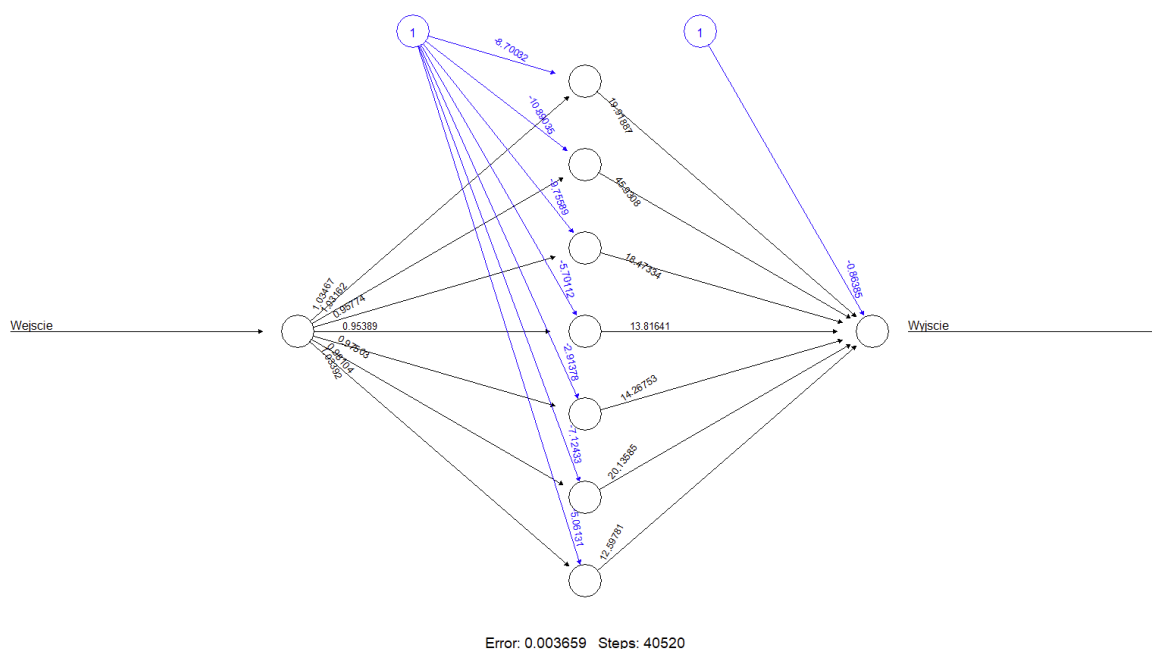
Kod programu dostępny w repozytorium: <https://github.com/Jaro233/APU.git>

Zad 1.

Tabela z wynikami:

wejście	Oczekiwane wyjście	wyjście sieci neuronowej
1 5.621926	3.45334880	31.621178
2 3.252817	2.35904298	10.621047
3 1.599450	0.93931924	2.780771
4 5.478381	3.40161925	30.027152
5 6.235382	3.66047958	38.884536
6 1.932210	1.31732875	3.883053
7 9.278266	4.45534930	86.081774
8 4.427947	2.97587230	19.614095
9 6.752123	3.81971402	45.583532
10 5.038535	3.23423065	25.394716
11 2.195212	1.57255725	4.921713
12 6.428463	3.72147106	41.324170
13 5.991137	3.58056242	35.904649
14 2.040966	1.42684676	4.294185
15 1.014970	0.02971839	1.356835
16 1.598039	0.93755454	2.776580
17 4.584770	3.04548000	21.025295
18 9.550232	4.51313085	91.216010
19 9.714987	4.54733946	94.391344
20 6.146701	3.63183094	37.788984
21 6.386759	3.70845402	40.790882
22 7.268519	3.96710512	52.826791
23 4.471455	2.99542782	20.000501
24 5.685280	3.47576079	32.337411
25 8.960599	4.38567407	80.280593
26 7.748064	4.09488608	60.039437
27 2.725923	2.00561381	7.482712
28 7.076986	3.91369623	50.075992
29 8.980095	4.39002083	80.630347
30 4.738587	3.11147805	22.458952
31 2.667539	1.96231301	7.170764
32 2.555295	1.87633547	6.591785
33 6.333207	3.69161338	40.111222
34 2.630069	1.93402043	6.974437
35 6.856886	3.85050687	47.008405
36 2.076519	1.46138628	4.434182
37 7.118461	3.92538322	50.665262
38 5.722013	3.48864145	32.756251
39 2.065910	1.45114129	4.392114
40 4.655463	3.07608260	21.678103
41 8.784949	4.34607973	77.166073
42 4.746678	3.11489004	22.535719
43 5.561834	3.43185572	30.948998
44 5.606641	3.44790352	31.449533
45 2.768340	2.03649555	7.713943
46 4.065244	2.80494763	16.544276
47 9.353894	4.47158551	87.494663
48 7.513201	4.03332313	56.449624
49 1.849870	1.23023124	3.588748
50 3.402722	2.44915140	11.616486

Schemat sieci neuronowej prezentuje się następująco:



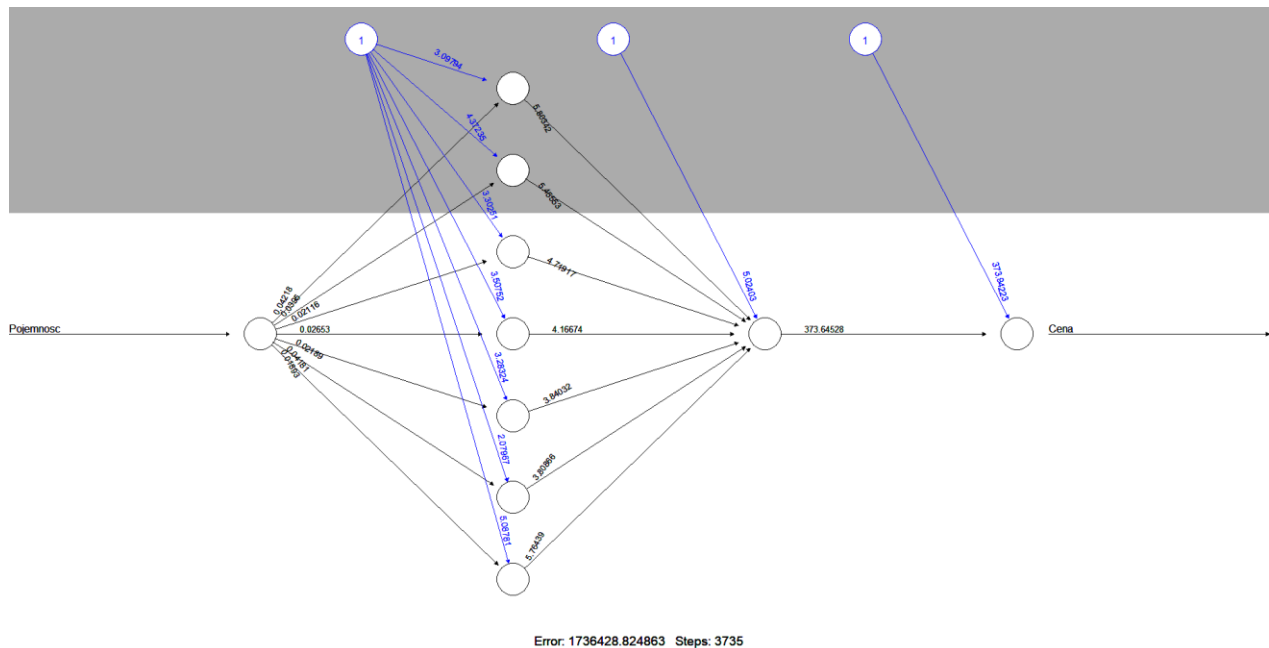
Możemy z niego wyczytać wagi jakie algorytm dobrał dla poszczególnych węzłów. Widzimy także, że błąd wynosi **0.003659** i wynik ten został osiągnięty po **40520** iteracjach.

Zad 2.

Tabela z wynikami:

	Pojemnosc	Cena
1	20	747.5875
2	130	747.5875

Schemat sieci neuronowej z progiem błędu < 100zł wygląda następująco:



5. Wnioski

W trakcie pracy nad ćwiczeniem zaimplementowano modele sieci neuronowych. Dzięki przeprowadzonemu uczeniu zostały ustalone wagi dla poszczególnych węzłów.