

Sprawozdanie

Zajęcia: Analiza procesów uczenia

Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium 3

Data: 12.04.2024

Temat: Użycie sztucznych sieci neuronowych

Wariant: 1

Agnieszka Białecka
Informatyka II stopień,
stacjonarne,
1 semestr,
Gr.1a

1. Cel ćwiczenia

Celem ćwiczenia było uczenie maszynowe nadzorowane za pomocą procedury nauczania sztucznej sieci neuronowej.

2. Wstęp teoretyczny

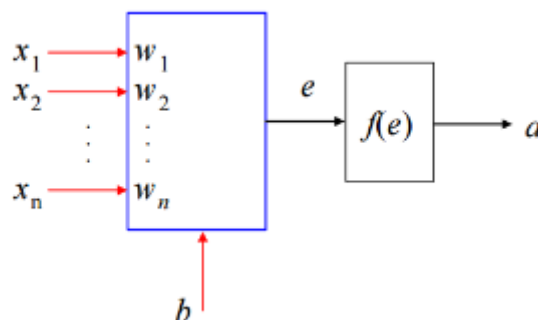
Sztuczne sieci neuronowe są techniką inteligentną powstałą z inspiracji strukturą i możliwościami analitycznymi ludzkiego mózgu. Zastosowanie sieci neuronowych wiąże się z możliwością ich uczenia do konkretnego zadania, tak że sieć taka staje się matematycznym modelem systemu czy procesu, który podlega analizie. Niezależnie od konkretnej struktury, każda sieć neuronowa zbudowana jest z pojedynczych modułów obliczeniowych, zwanych neuronami.

Pojedynczy neuron realizuje obliczenie sumy ważonej sygnałów wejściowych z nałożoną nieliniową funkcją aktywacji. Jego wyjście określone jest zależnością:

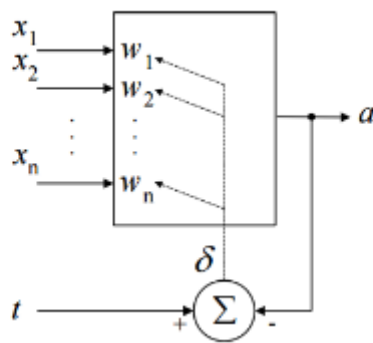
$$a = f(X * W + b)$$

gdzie: X – wektor sygnałów wejściowych o rozmiarze $[n \times 1]$, W – wektor wag synaptycznych o rozmiarze $[1 \times n]$, $f(e)$ – funkcja aktywacji neuronu, zwykle nieliniowa, b – współczynnik przesunięcia (ang. bias).

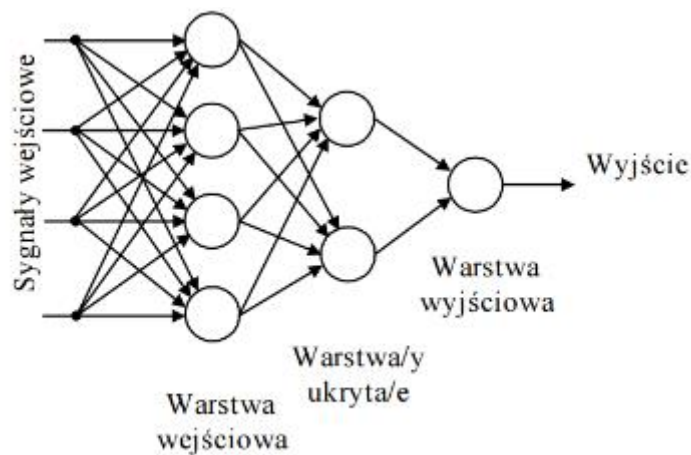
Projektowanie neuronowego układu rozpoznającego lub decyzyjnego zarówno jedno-neuronowego, jak i wielowarstwowego, jest możliwe poprzez uczenie sieci neuronowej z wykorzystaniem odpowiednio przygotowanych wzorców uczących i stosownego algorytmu uczenia. Uczenie polega na dostosowaniu/dostrojeniu współczynników wagowych (wag synaptycznych) neuronu/neuronów, aby uzyskać jak największą zgodność wyjścia neuronu/sieci a z wymaganym sygnałem wyjściowym t .



Zdjęcie 1. Model pojedynczego neuronu z nieliniową funkcją aktywacji



Zdjęcie 2. Ilustracja procesu uczenia pojedynczego neuronu



Zdjęcie 3. Struktura wielowarstwowa perceptronu

Istnieją również metody uczenia bez nauczyciela (tzn. bez określania i podawania żdanego wyjścia neuronu/sieci), oparte na idei samoorganizacji danych.

Do najczęściej stosowanych sieci neuronowych należy zaliczyć:

- sieci typu wielowarstwowy perceptron – trój- lub czterowarstwowe sieci jednokierunkowe, sieci z radialną funkcją bazową,
- sieci typu Hopfielda – maszyny Boltzmann, Gaussa, sieci chaotyczne,
- sieci Kohonena – dwu- lub trójwymiarowe sieci kratowe,
- inne – rzadziej stosowane.

3. Przebieg ćwiczenia

3.1. Zadanie 1

Zadanie dotyczy modelowania funkcji matematycznej $f(x) = x^3 + 2 * x$ 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 $\text{Error} < 0.01$.

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

set.seed(42)
input <- as.data.frame(runif(100, min=1, max=100))
output <- input^3 + 2*input

normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}

input <- as.data.frame(normalize(input))
output <- normalize(output)

trainingdata <- cbind(input, output)
colnames(trainingdata) <- c("wejscie", "wyjscie")
net.sqrtn <- neuralnet(wyjscie ~ wejscie, trainingdata, hidden=3, threshold=0.01, stepmax=500000)
print(net.sqrtn)
plot(net.sqrtn, rep = "best")

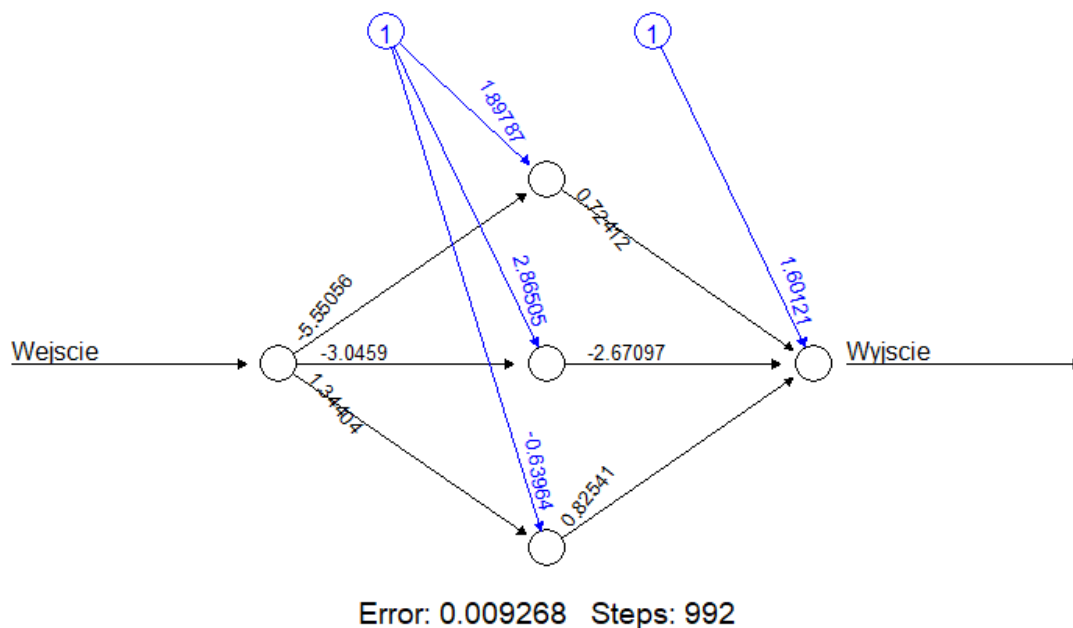
testdata <- as.data.frame(runif(300, min=1, max=10))
testdata <- normalize(testdata)
net.results <- compute(net.sqrtn, testdata)
print(net.results$net.result)

cleanoutput <- cbind(testdata, normalize(testdata^3 + 2*testdata), as.data.frame(net.results$net.result))
colnames(cleanoutput) <- c("wejscie", "oczekiwane wyjscie", "wyjscie sieci neuronowej")
print(cleanoutput)
```

Zdjęcie 4. Kod skryptu

	wejscie	oczekiwane wyjscie	wyjscie sieci neuronowej
1	0.35642019	0.25270612	4.300327e-02
2	0.41064923	0.29684913	6.207180e-02
3	0.57444418	0.44614898	1.858636e-01
4	0.59074152	0.46254579	2.045223e-01
5	0.72148213	0.60617400	3.945924e-01
6	0.39489537	0.28379055	5.563719e-02
7	0.92219802	0.87622620	7.889615e-01
8	0.96581849	0.94418588	8.828243e-01
9	0.23249983	0.15918924	2.174492e-02
10	0.72635080	0.61197124	4.029313e-01
11	0.90653739	0.85269210	7.555941e-01
12	0.60461814	0.47675412	2.213343e-01
13	0.63281561	0.50634859	2.580718e-01

14	0.94048648	0.90428239	8.281835e-01
15	0.85307417	0.77565357	6.439179e-01
16	0.58082635	0.45253328	1.930301e-01
17	0.82382495	0.73558988	5.848154e-01
18	0.11199291	0.07513016	8.291003e-03
19	0.76659540	0.66123159	4.747897e-01
20	0.62487552	0.49791527	2.473822e-01
21	0.14692440	0.09900681	1.264919e-02
22	0.07834274	0.05238877	3.397301e-03
23	0.46439675	0.34298244	9.055173e-02
24	0.78154288	0.68015313	5.027113e-01
25	0.73543407	0.62287914	4.187009e-01
26	0.81962702	0.72995666	5.764758e-01
27	0.16876752	0.11411398	1.508415e-02
28	0.94786393	0.91577748	8.440663e-01
29	0.29295231	0.20368203	2.954141e-02
30	0.14755351	0.09943985	1.272199e-02
31	0.72120179	0.60584125	3.941147e-01
32	0.32359291	0.22702334	3.510413e-02
33	0.78098094	0.67943551	5.016504e-01
34	0.39436021	0.28335044	5.543237e-02
35	0.68017708	0.55834396	3.272288e-01
36	0.77797900	0.67561027	4.959977e-01
37	0.18657783	0.12655023	1.696234e-02
38	0.02686421	0.01791593	-5.587240e-03
39	0.13411698	0.09021546	1.112667e-02
40	0.68175760	0.56013055	3.296903e-01
41	0.93790857	0.90028983	8.226409e-01
42	0.55132770	0.42341273	1.614268e-01
43	0.60290028	0.47498268	2.192072e-01
44	0.19575675	0.13300501	1.790965e-02
45	0.53598082	0.40864526	1.465203e-01
46	0.17821582	0.12069731	1.608921e-02
47	0.45214231	0.33223909	8.310577e-02
48	0.31651910	0.22158282	3.368435e-02
49	0.11446336	0.07680880	8.621731e-03
50	0.18480059	0.12530412	1.677778e-02



Zdjęcie 5. Uzyskany schemat

3.2. Zadanie 2

Zadanie dotyczy prognozowania ceny urządzeń RTV AGD ($\text{error} \leq 100 \text{ zł}$), określonych na Zajęciach 1. Używając metody sztucznych sieci neuronowych opracować plik w języku R z wykorzystaniem paczki neuralnet.

```
df <- read.csv("D:/Studia/APU/smartfony.csv")
ram <- df[["pamiec_RAM"]]
cena <- df[["cena"]]

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

colnames(trainingdata) <- c("Pamięć_Ram", "Cena")
#(error ≤ 100 zł)
net.price <- neuralnet(Cena~Pamięć_Ram,trainingdata, hidden=c(7,1), threshold=-100, lifesign <- "full")
plot(net.price)

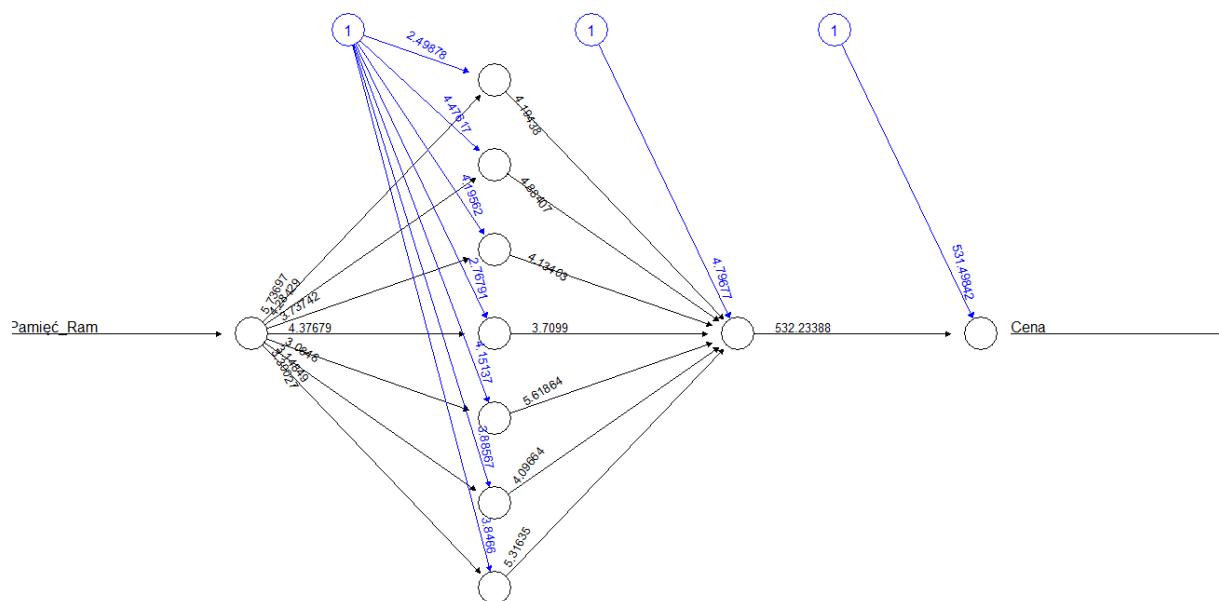
testdata <- data.frame(c(20,130))
scaled.testdata <- as.data.frame(scale(testdata))

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

print(fixed_cena)
```

Zdjęcie 6. Kod skryptu

Pamięć_Ram	Cena
1	20 1063.732
2	130 1063.732



Error: 321354.093202 Steps: 5328

Zdjęcie 7. Uzyskany schemat

4. Podsumowanie

Przeprowadzone ćwiczenia umożliwiło zapoznanie się z zaawansowanym konceptem sieci neuronowych, które odgrywają kluczową rolę we współczesnej analizie danych i sztucznej inteligencji. W ramach ćwiczenia został opracowany model matematyczny, który wykorzystuje algorytmy uczenia maszynowego do analizy wzorców w danych.