

# **SPRAWOZDANIE**

Zajęcia: Analiza Procesów Ucznia

Prowadzący: prof. Dr hab. Vasyl Martsenyuk

## **Laboratorium 3**

3.02.2021

**Temat:** „Użycie sztucznych sieci neuronowych”

**Wariant 8**

Kamil Pająk  
Informatyka II stopień  
Stacjonarne (zaoczne)  
1 semestr

## 1. Polecenie:

Zadanie 1 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) spełniającej warunek  $\text{Error} < 0.01$

Zadanie 2 dotyczy prognozowania ceny urządzeń RTV AGD ( $\text{error} \leq 100 \text{ zł}$ ), określonych na Zajęciu 1. Używając metody sztucznych sieci neuronowych opracować plik w języku R z wykorzystaniem paczki neuralnet

## 2. Wprowadzane dane:

Cały program znajduje się na Githubie:

<https://github.com/vincidaking/APU>

```

library(neuralnet)
library(ggplot2)

function_ <- function(x) (sin(x) ^ cos(x));
stat_point <- 0
end_point <- pi
n <- 1000

inputs <- runif(n, stat_point, end_point)
outputs <- function_(inputs)
plot(inputs, outputs)
scaled_inputs = inputs / max(inputs)
scaled_outputs = outputs / max(outputs)
plot(scaled_inputs, scaled_outputs)

library(neuralnet)
dataset <- data.frame
(inputs=scaled_inputs, outputs= scaled_outputs)
train_data_precent <- 0.7
train_data <- dataset[1:(n * train_data_precent),]
test_data <- dataset[1:(n * (1 -train_data_precent)),]
set.seed(2)
nn <- neuralnet(outputs ~ inputs, data=train_data,
                hidden=c(50, 30, 20),
                threshold=0.0001, linear.output = F)

plot(nn)
result <- compute(nn, test_data[1])
calculated_x <- as.numeric(test_data$inputs)
calculated_y <- result[["net.result"]]
plot(calculated_x, calculated_y)

error <- sqrt(sum(
  (as.numeric(test_data$outputs)
  -calculated_y)^2)/length(calculated_y))

```

```

library(neuralnet)
library(ggplot2)

dataset <- read.csv("monitors.csv");
#train_data <- file_data['wielkosc':'ocena_klientow'];

normalize <- function(x) return (x / max(x));
dataset_normalized <- as.data.frame
(lapply(dataset, normalize))
dataset_normalized <- dataset_normalized[sample
(1:nrow(dataset_normalized)), ]

library(neuralnet)
n <- nrow(dataset_normalized)
train_data_precent <- 0.7
train_data <- dataset_normalized[1:
(n * train_data_precent),]
test_data <- dataset_normalized[1:
(n * (1 -train_data_precent)),]

set.seed(2)
nn <- neuralnet(Cena ~.,
                data=train_data,
                hidden=c(100, 50 ,25),
                threshold=0.001, linear.output = F)

plot(nn)

result <- compute(nn, test_data)
calculated_x <- as.numeric(test_data$Cena)
calculated_y <- result[["net.result"]]
plot(calculated_x, calculated_y)

error <- sqrt(sum(
  (as.numeric
  (test_data$Cena)-calculated_y)^2)
/length(calculated_y))

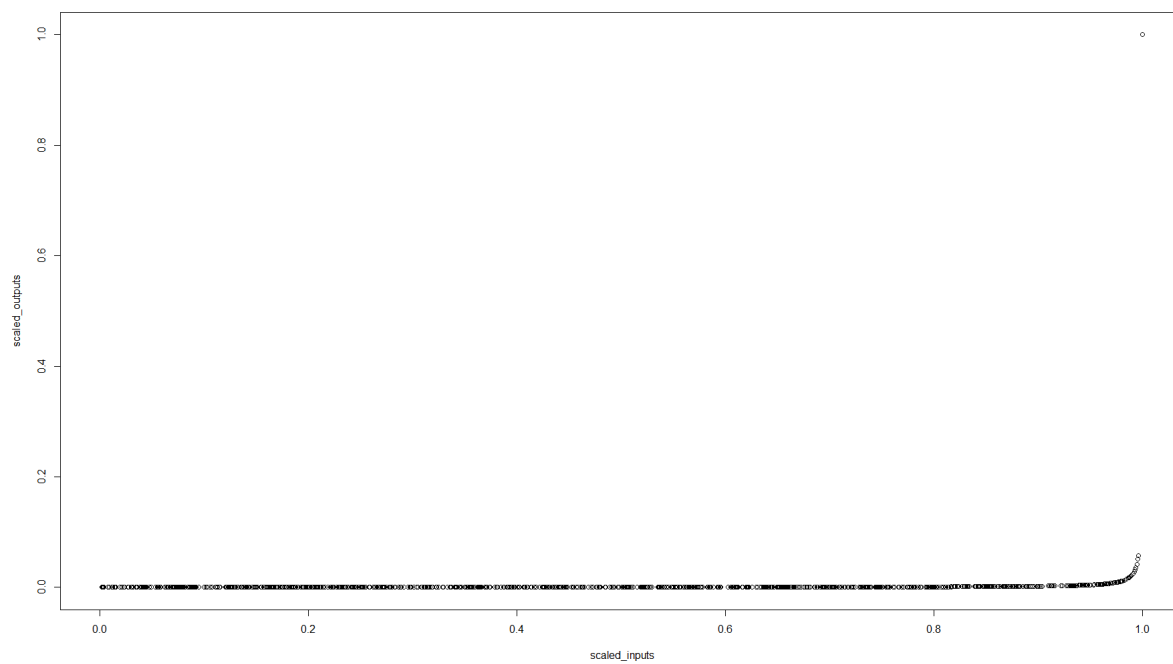
```

monitors

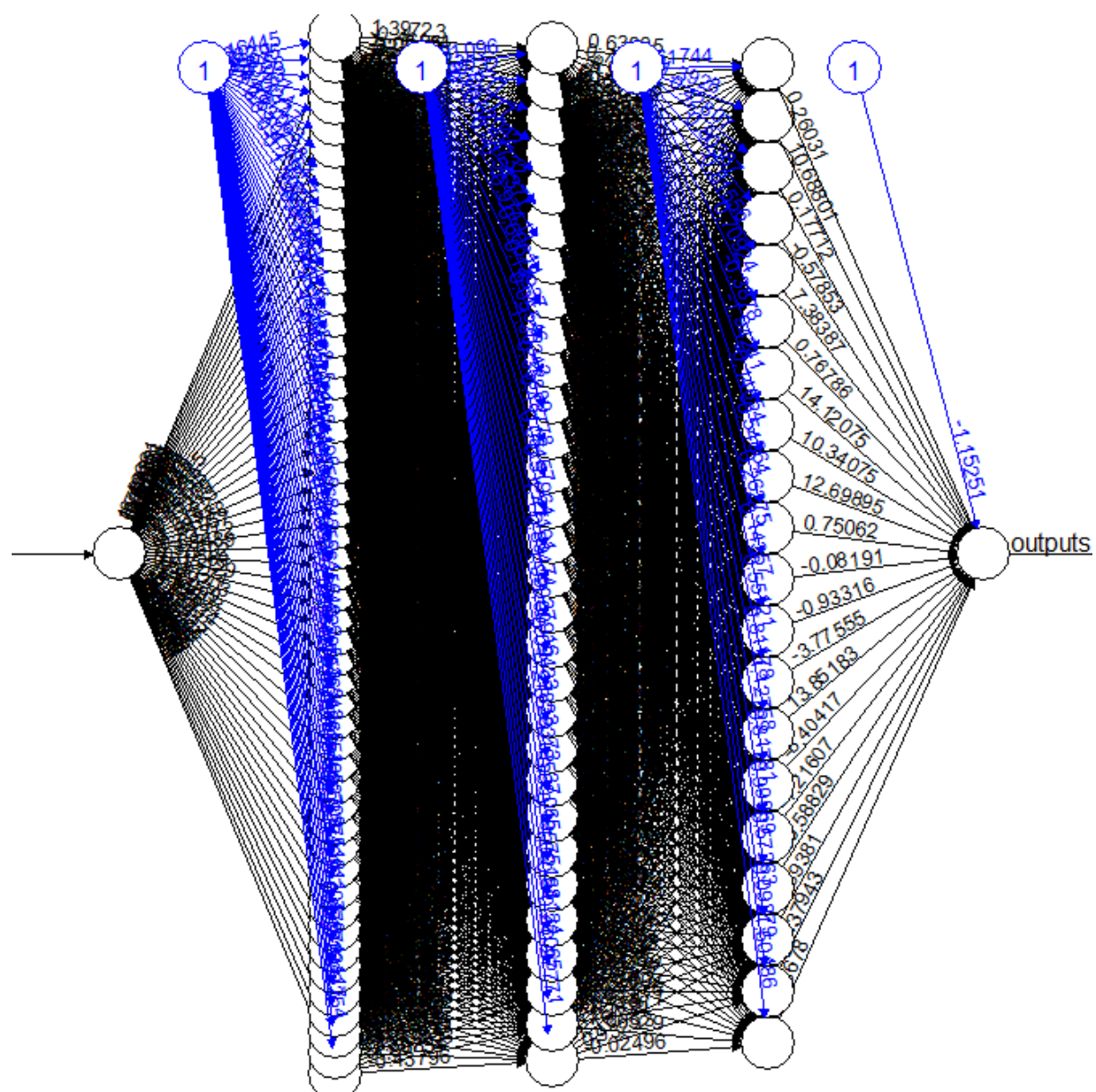
Matryca	Rozdzielczosc_w	Rozdzielczosc_h	Jakosc_ekranu	Czas_reakcji_matrycy	Czestotliwosc_odswiezania	Kontrast_statyczny	Cena	Ocena	Liczba_ocen
24	1920	1080	350	1	144	1000	829	4.8	25
23.6	1920	1080	300	1	144	3000	1099	3	1
27	1920	1080	250	5	60	1000	699	5	53
27	1920	1080	250	4	60	3000	749	4.4	68
24	1920	1080	250	4	60	3000	549	5	1
18.5	1366	768	200	5	60	700	259	4.7	150
24.5	1920	1080	250	1	75	1000	599	5	28
23.6	1920	1080	250	5	60	3000	449	4	3
23.8	1920	1080	250	1	75	3000	599	4.5	45
27	1920	1080	250	4	75	3000	1029	5	9
24	1920	1080	250	1	60	1000	499	5	114
23.8	1920	1080	250	1	75	1000	549	5	107
27	1920	1080	250	4	60	3000	749	5	77
15	2560	1080	250	5	75	1000	649	5	15
34	3440	1440	300	5	75	1000	2999	5	14
24	2560	1440	350	1	165	1000	1839	5	14
27	2560	1440	350	1	144	1000	1999	5	13

plik monitors.csv

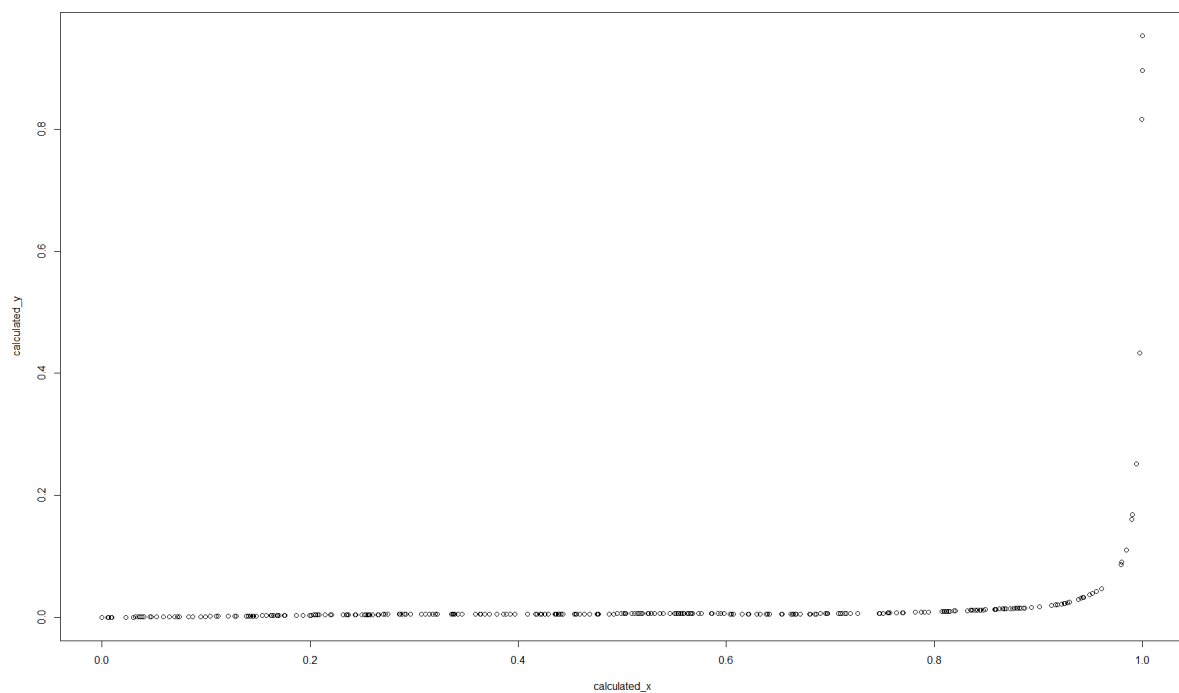
### 3. Wynik działania:



Funkcja aproksymacji

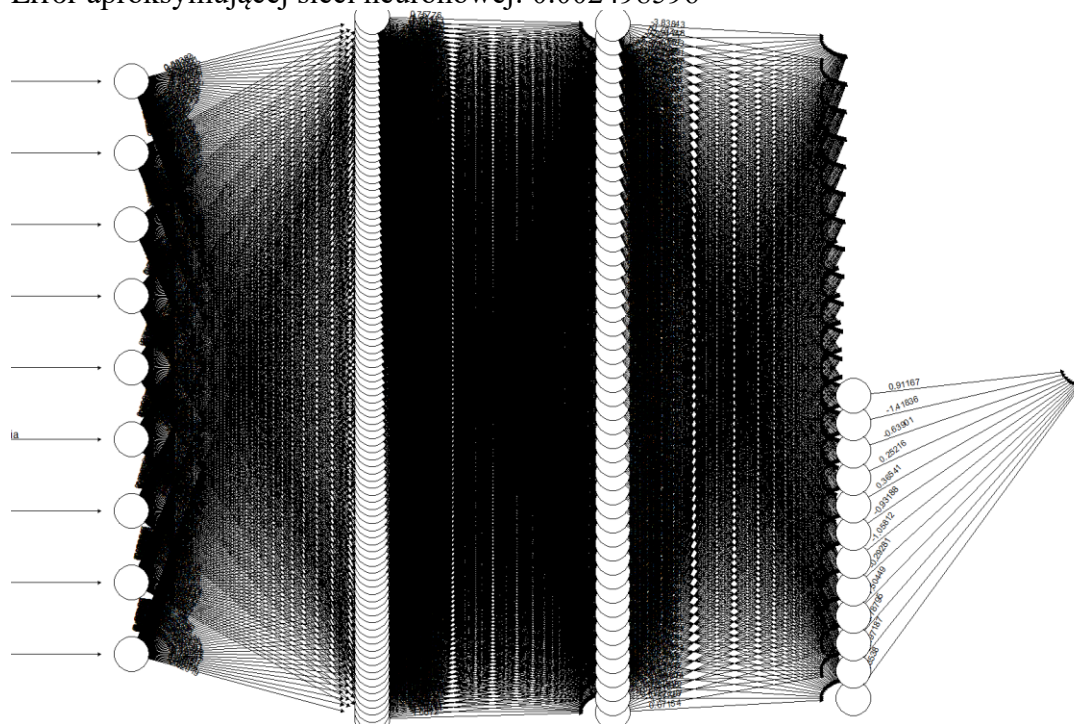


Budowa sieci aproksymującej



Wychodząca funkcja z sieci neuronowej

Error aproksymującej sieci neuronowej: 0.002498596



Sieć neuronowa predykująca cenę  
Error funkcji predykującej cenę: 0.01178325

#### **4. Wnioski:**

Na podstawie otrzymanego wyniku można stwierdzić, że język R oraz paczka neuralnet pozwalają w szybki i wygodny sposób tworzyć sieci neuronowe uczące się w sposób nadzorowany. Dodatkowym atutem paczki neuralnet jest wizualizator sieci neuronowym przedstawiający poszczególne wagi oraz połączenia między neuronami.