# SPRAWOZDANIE

Zajęcia: Analiza Procesów Uczenia
Prowadzący: prof. dr hab. Vasyl Martsenyuk

**Laboratorium 3**
Data 31.03.2023
**Temat:** Użycie sztucznych sieci neuronowych
**Wariant 1**

Rafał Klinowski
Informatyka II stopień,
Stacjonarne,
1 semestr,
Gr. a

## 1. Polecenie: Wariant 1

**Zadanie 1.** Zadanie dotyczy modelowania funkcji matematycznych za pomocą sztucznej sieci neuronowej używając paczkę neuralnet. Rozważamy źmienną 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 **vmartsenyuk@ath.bielsko.pl**

1. $f(x) = x^3 + 2 * x, \quad x \in [1; 100]$

**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 konsolu (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:

Dane do pierwszego z zadań zostały wygenerowane „ręcznie” w programie Microsoft Excel (pary {x, f(x)} dla różnych wartości x z przedziału [1,100]).

Dane do drugiego z zadań zostały zaczerpnięte z zestawu końcowego po pierwszym z ćwiczeń laboratoryjnych.

## 3. Wykorzystane komendy:

Poniżej można znaleźć wszystkie wykorzystane komendy:

```
# Autor: Rafal Klinowski, wariant: 1.

setwd('C:\\Users\\klino\\Pulpit\\Studia magisterskie\\APU\\Lab3')

# Instalacja pakietu neuralnet
install.packages('neuralnet')
library('neuralnet')

# Zadanie 1.
# Funkcja: f(x) = x^3 + 2x, x < [1; 100]

# Wczytanie wczesniej utworzonego pliku z danymi
dane <- read.csv('dane1.csv')
```

```
# Bez normalizacji
# scaled <- dane

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

scaled <- as.data.frame(lapply(dane, normalize))

# Normalizacja
# scaled <- scale(dane)

# Podzial danych na set treningowy i walidacyjny
trainset <- scaled[1:80, ]
testset <- scaled[81:100, ]

# Utworzenie sieci neuronowej
# wyjscie ~ wejscie1 + wejscie2 + ...
# Testowane wartości hidden: od 5 do 15, oraz c(2,1), c(2,2), c(5,5)
# Najlepsze wyniki dla hidden=10
nn <- neuralnet(Output ~ Input, data=trainset, hidden=10, threshold=0.01, linear.output=TRUE)

print(nn)
#plot(nn)

results <- compute(nn, testset)

ls(results)
comparison <- data.frame(actual=testset[,2], prediction=results$net.result)
comparison

# Zadanie 2.

# Wczytanie pliku zawierajacego dane o smartfonach z Lab1.
smartfony <- read.csv('smartfony.csv')

# Wybor tylko interesujacych nas kolumn
smartfony_reduced <- smartfony[, c('pamiec_ram', 'pamiec_wbudowana', 'aparat_foto', 'cena')]

# Normalizacja danych
smartfony_scaled <- as.data.frame(lapply(smartfony_reduced, normalize))

smartfony_train <- smartfony_scaled
smartfony_test <- smartfony_scaled[1:3, ]

# Stworzenie modelu
# hidden=c(3,2)
smartfony_nn <- neuralnet(cena ~ pamiec_ram + pamiec_wbudowana + aparat_foto,
data=smartfony_train,
                          hidden=c(3,2), threshold=0.01)

print(smartfony_nn)
plot(smartfony_nn)
```

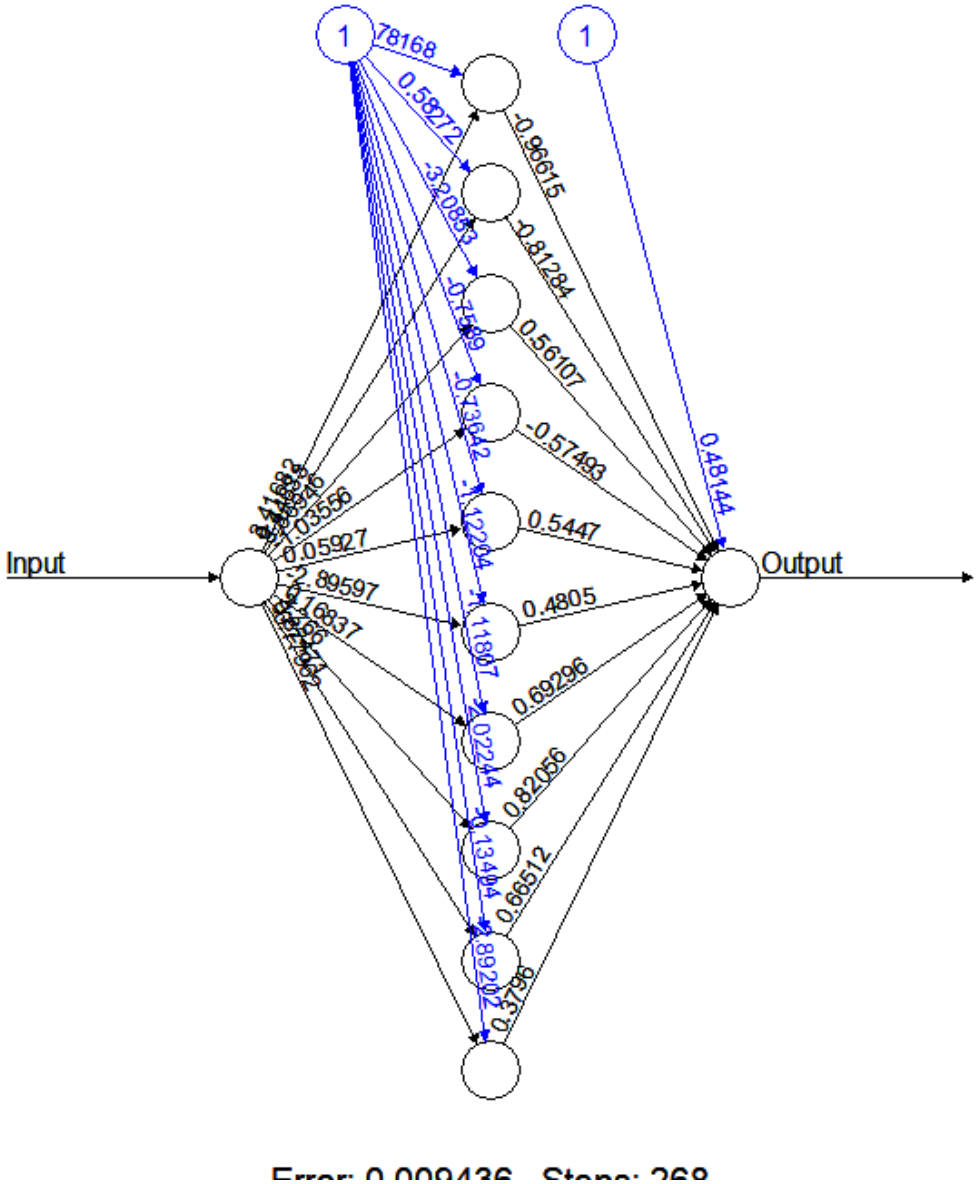
```
smartfony_results <- compute(smartfony_nn, smartfony_test)

ls(smartfony_results)
smartfony_comparison <- data.frame(actual=smartfony_test[,c('cena')],
prediction=smartfony_results$net.result)
smartfony_comparison
```

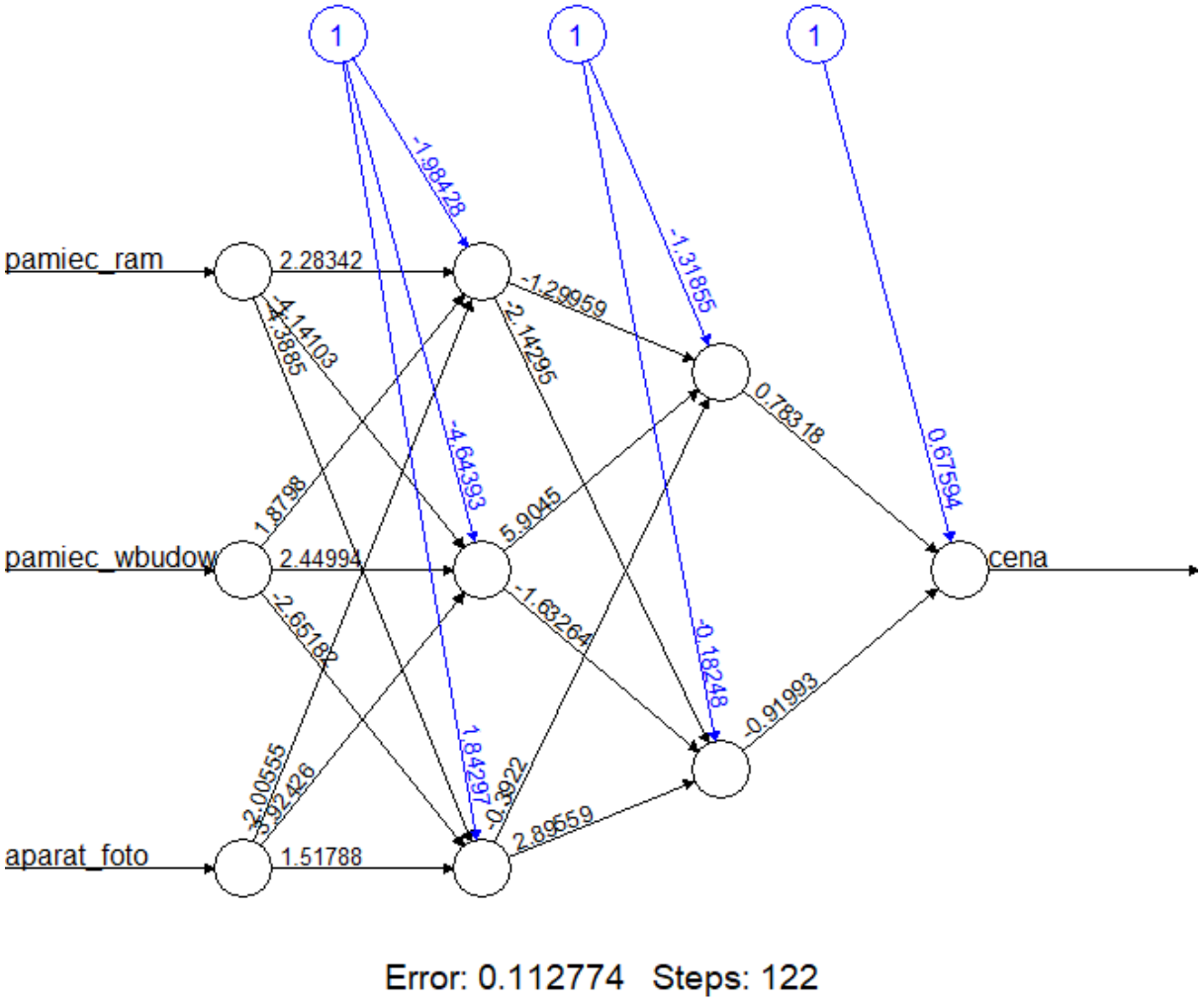
## 4. Wynik działania:

Wyniki poleceń w konsoli można znaleźć w pliku „wyniki z konsoli.txt”, link do repozytorium poniżej.

Zrzuty ekranu przedstawiające sieci uzyskane w pierwszym i drugim zadaniu:




*Rysunek 1. Pierwsza z sieci (zadanie 1).*




*Rysunek 2. Druga z sieci (zadanie 2).*

Zgodnie z poleceniem, link do repozytorium GitHub zawierający niezbędne pliki znajduje się tutaj: https://github.com/Stukeley/APU_Lab3

**5. Wnioski:**

Warto zauważyć, że wygląd sieci czy uzyskane wyniki różnią się przy każdym uruchomieniu programu.
Mimo dość niewielkiej ilości danych wejściwych w drugim zadaniu, udało się uzyskać dość dobre wyniki (zbliżone wartości oczekiwane i otrzymane).