**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

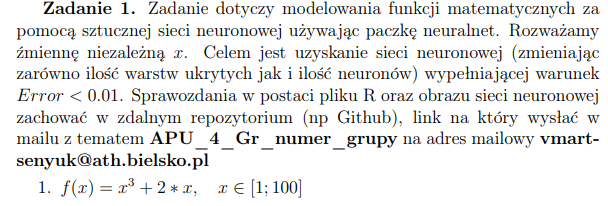
Informatyka II stopień,

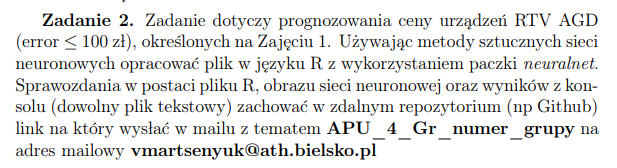
Stacjonarne,

1 semestr,

Gr. a

1. **Polecenie: Wariant 1**





1. **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.

1. **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)

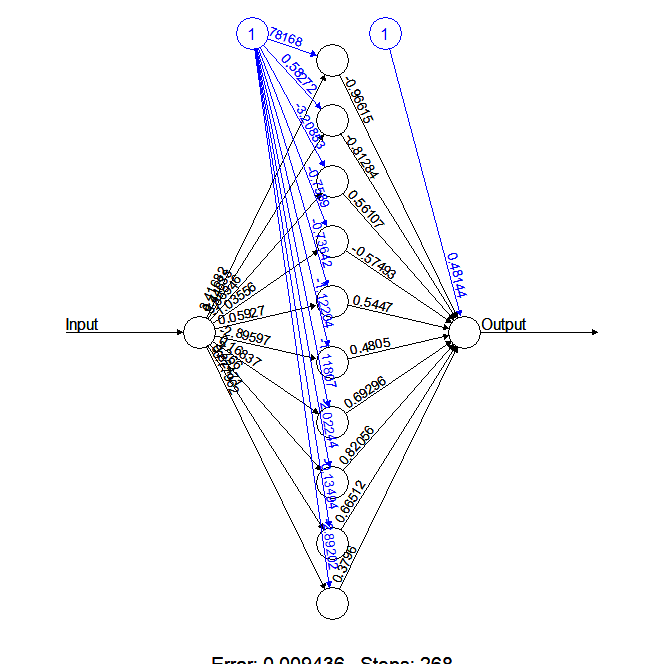
smartfony\_comparison <- data.frame(actual=smartfony\_test[,c('cena')], prediction=smartfony\_results$net.result)

smartfony\_comparison

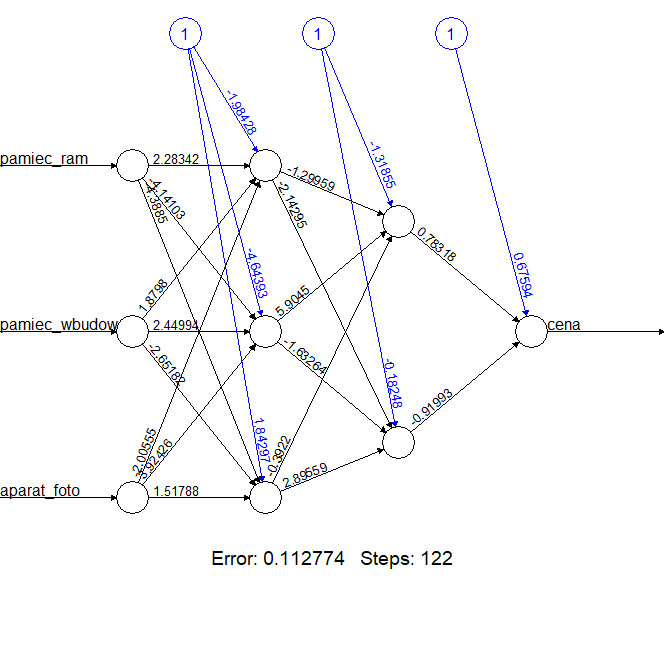
**4. Wynik działania:**

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](https://github.com/Stukeley/APU_Lab3%20)

1. **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).