RML\_110.R

darki

2020-04-03

# Rozdział 11 - Sieci neuronowe  
  
rm(list=ls())  
library(neuralnet)  
  
#Definiujemy zbiór treningowy dla zadania XOR  
training\_input\_x <- c(0,0,1,1)  
training\_input\_y <- c(0,1,0,1)  
training\_output <-c(0,1,1,0)  
  
#utworzenie ramki dla zbioru treningowego  
data <- data.frame(training\_input\_x,training\_input\_y,training\_output)  
  
#utworzenie sieci neuronowej  
#linear.output = TRUE dla regresji and FALSE problemu klasyfikacji  
neural\_network <-neuralnet(training\_output~training\_input\_x+training\_input\_y, data, hidden=3, threshold = 0.001, algorithm = "backprop", learningrate = 0.1, linear.output = TRUE)  
  
#wykres archtektury sieci euronowej  
plot(neural\_network)  
  
#prognozowanie danych  
prediction <- compute(neural\_network, data.frame(training\_input\_x=1,training\_input\_y=1))  
prediction$net.result

## [,1]  
 ## [1,] 0.006126514

#zadanie dla danych kredytowych  
credit\_data <- read.csv("./Dane/06 - credit\_data.csv",TRUE,",")  
  
library(neuralnet)  
  
#kilka pierwszych danyh  
head(credit\_data)

## clientid income age loan LTI default  
 ## 1 1 66155.93 59.01702 8106.5321 0.1225367512 0  
 ## 2 2 34415.15 48.11715 6564.7450 0.1907515807 0  
 ## 3 3 57317.17 63.10805 8020.9533 0.1399397997 0  
 ## 4 4 42709.53 45.75197 6103.6423 0.1429105321 0  
 ## 5 5 66952.69 18.58434 8770.0992 0.1309895000 1  
 ## 6 6 24904.06 57.47161 15.4986 0.0006223321 0

#normalizacja zbioru danych  
normalize <- function(x) {return( (x-min(x)) / (max(x)-min(x)) )}  
normalized\_credit\_data <- as.data.frame(lapply(credit\_data[2:4], normalize))  
#dodanie nazw kolumny  
normalized\_credit\_data$default <- credit\_data$default  
  
#podział danych na zbiór treningowy i testowy  
training\_data <- normalized\_credit\_data[1:1500,]  
test\_data <- normalized\_credit\_data[1501:2000,]  
  
#uczenie sieci neuronowej  
neural\_network <- neuralnet(default~income+age+loan, training\_data, hidden=5, act.fct = "logistic", linear.output = FALSE)  
  
#wykres architektury sieci  
plot(neural\_network)  
  
#testowanie  
test\_data2 <- subset(test\_data,select=c("income","age","loan"))  
  
#prognozowanie  
predictions <- compute(neural\_network, test\_data2)  
  
#zaokrąglenie  
predictions <- round(predictions$net.result)  
  
#dokładność modelu  
mean(test\_data$default==predictions)

## [1] 0.988