# Sprawozdanie laboratorium 2 (sztuczne sieci neuronowe)

**Autorzy:** Remigiusz Nowakowski  
**Grupa:** INMN1(hybryda)\_1.1, INMN1(hybryda)\_sp\_wszyscy  
**Data:** 02.12.2023  
**Przedmiot:** Inteligencja obliczeniowa

|  |  |
| --- | --- |
| **Cel zadania (co chcemy klasyfikować)** | Przewidywanie liczby cylindrów silnika na podstawie osiągów pojazdu, czyli liczby mil, jakie pojazd może przejechać na jednym galonie paliwa. |
| **Skąd pobrano zbiór danych?** | data(mtcars) |
| **Jakie dane zawiera każda kolumna zbioru danych?** | Baza danych "mtcars" zawiera informacje dotyczące różnych cech i parametrów 32 różnych modeli samochodów. Oto opis niektórych kolumn w tej bazie:  - mpg: Miles/(US) gallon - Oszczędność paliwa, liczba mil, jakie pojazd może przejechać na jednym galonie paliwa.  - cyl: Number of cylinders - Liczba cylindrów.  - disp: Displacement (cu.in.) - Pojemność skokowa, objętość silnika w calach sześciennych.  - hp: Gross horsepower - Moc silnika w koniach mechanicznych.  - drat: Rear axle ratio - Stosunek przekładni osi tylnego mostu.  - wt: Weight (1000 lbs) - Waga pojazdu w tysiącach funtów.  - qsec: 1/4 mile time - Czas przejazdu ćwiartki mili (mierzone w sekundach).  - vs: V/S - Rodzaj układu napędowego (0 = silnik z przodu, napęd tylny; 1 = silnik z przodu, napęd na wszystkie koła).  - am: Transmission (0 = Automatic, 1 = Manual) - Rodzaj skrzyni biegów (0 = automatyczna, 1 = manualna).  - gear: Number of forward gears - Liczba biegów do przodu.  - carb: Number of carburetors - Liczba gaźników.  To są podstawowe kategorie danych w tej bazie. Każdy wiersz reprezentuje inny model samochodu, a kolumny zawierają różne parametry związane z jego wydajnością i specyfikacjami. |
| **Jak jest dokładność klasyfikacji?** | 90,90909% |
| **Wnioski i podsumowanie (analiza otrzymanych wyników)** | Na podstawie otrzymanych wyników możemy określić z dokładnością ponad 90% ilość cylindrów silnika na podstawie osiągów pojazdu. |
| **Bibliografia** | [1] Biblioteka AMORE  [2] Moodle  [3] data(faithful)  [4] |

## Kod źródłowy rozwiązania wraz z dodatkowymi (szczegółowymi) komentarzami, (w szczególności proszę opisać parametry użytych funkcji z pakietu AMORE)

|  |
| --- |
| # Instalacja pakietu AMORE (odkomentuj linię poniżej, jeśli nie jest zainstalowany)  # install.packages("AMORE")  library(AMORE)  # Załaduj dane  data(mtcars)  # Określenie funkcji target  target <- function(x) {  n <- length(x)  values <- levels(x)  l <- length(values)  T <- matrix(0, nrow = n, ncol = l)  for (i in 1:l)  T[, i] <- (x == values[i])  colnames(T) <- values  return(T)  }  # Zastosowanie funkcji target dla kategorii "cyl"  wZadane <- target(as.factor(mtcars$cyl))  # Ustawienie ziarna dla powtarzalności wyników  set.seed(434)  # Podział danych na dane trenujące i testowe (2/3 - 1/3)  ile <- nrow(mtcars)  idxTren <- sample(1:ile, 2 \* ile / 3)  idxTest <- setdiff(1:ile, idxTren)  # Tworzymy strukturę sieci  siec <- newff(n.neurons = c(1, 5, length(unique(mtcars$cyl))),  learning.rate.global = 0.01,  momentum.global = 0.3,  hidden.layer = "sigmoid",  output.layer = "purelin",  method = "ADAPTgdwm",  error.criterium = "LMS")  # Trenujemy sieć  wynik <- train(siec,  mtcars[idxTren, "mpg"],  wZadane[idxTren, ],  error.criterium = "LMS",  report = TRUE,  show.step = 10,  n.shows = 800)  # Wyświetlamy wartości błędów  plot(wynik$Merror, type = "l", xlab = "Iteracja (x10)",  ylab = "Błąd", col = "darkred")  # Stosujemy wytrenowaną sieć do danych testowych  y <- sim(wynik$net, mtcars[idxTest, "mpg"])  # Wyniki klasyfikacji  wynik <- testKlasyfikacji(wZadane[idxTest, ], y)  # Określamy dokładność klasyfikacji  cat("Dokładność klasyfikacji:", sum(diag(wynik)) / sum(wynik) \* 100, "%\n") |