# 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)** | Rodzaj karmy na podstawie masy kurczaka |
| **Skąd pobrano zbiór danych?** | Data(chickewts) |
| **Jakie dane zawiera każda kolumna zbioru danych?** | 1. weight: Waga kurczaka (w gramach) 2. feed: Rodzaj karmy dostarczonej kurczakom. |
| **Jak jest dokładność klasyfikacji?** | 33,33% |
| **Wnioski i podsumowanie (analiza otrzymanych wyników)** | Klasyfikator jest dokładny w 33,33%. Istnieje możliwość poprawy tego wyniku. |
| **Bibliografia** | [1] Biblioteka AMORE  [2] Moodle  [3]data(chickewts)  [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(chickwts)  # Cel klasyfikacji - "feed"  cel\_klasyfikacji <- "feed"  # 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 celu klasyfikacji "feed"  wZadane <- target(chickwts[[cel\_klasyfikacji]])  # Ustawienie ziarna dla powtarzalności wyników  set.seed(434)  # Podział danych na dane trenujące i testowe (2/3 - 1/3)  ile <- nrow(chickwts)  idxTren <- sample(1:ile, 2 \* ile / 3)  idxTest <- setdiff(1:ile, idxTren)  # Tworzymy strukturę sieci  siec <- newff(n.neurons = c(12, 6, length(unique(chickwts[[cel\_klasyfikacji]]))),  learning.rate.global = 0.01,  momentum.global = 0.05,  hidden.layer = "sigmoid",  output.layer = "purelin",  method = "ADAPTgdwm",  error.criterium = "LMS")  # Trenujemy sieć  wynik <- train(siec,  chickwts[idxTren, 2], # Waga kurczaka jako druga kolumna  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, chickwts[idxTest, 2]) # Przewidywanie na podstawie wagi kurczaka  # Definiujemy funkcję oceny klasyfikacji (zamieniamy liczby na etykietę)  testKlasyfikacji <- function(zad, wy) {  zadane <- max.col(zad)  rozpoznane <- max.col(wy)  print(table(zadane, rozpoznane))  }  # Wyniki klasyfikacji  wynik <- testKlasyfikacji(wZadane[idxTest, ], y)  # Określamy dokładność klasyfikacji  cat("Dokładność klasyfikacji:",  sum(diag(wynik)) / sum(wynik) \* 100, "%\n") |