**SPRAWOZDANIE**

Zajęcia: Analiza Procesów Uczenia

Prowadzący: prof. dr hab. Vasyl Martsenyuk

**Laboratorium 6**

Data 12.05.2023

**Temat:**​ Ucznie głębokie w R.

**Wariant 1**

Rafał Klinowski

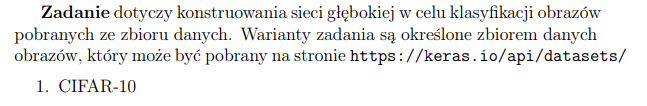
Informatyka II stopień,

Stacjonarne,

1 semestr,

Gr. a

1. **Polecenie: Wariant 1**



1. **Wprowadzane dane:**

Dane zostały zaczerpnięte z setu ‘Cifar10’ w pakiecie ‘keras’.

cifar <- dataset\_cifar10()

1. **Wykorzystane komendy:**

Poniżej można znaleźć wszystkie wykorzystane komendy:

# Autor: Rafal Klinowski, wariant: 1.

setwd('C:\\Users\\klino\\Pulpit\\Studia magisterskie\\APU\\Lab6')

# Instalacja TensorFlow

install.packages("tensorflow")

library(tensorflow)

install\_tensorflow()

# Instalacja Keras

install.packages("keras")

library(keras)

install\_keras()

# Pobranie danych wejsciowych

library(keras)

cifar <- dataset\_cifar10()

X\_train <- cifar$train$x

X\_test <- cifar$test$x

y\_train <- cifar$train$y

y\_test <- cifar$test$y

# Konwersja wartosci pikseli do zakresu [0, 1]

# Liczba klas = 10

X\_train <- X\_train / 255

X\_test <- X\_test / 255

# Konwersja etykiet na kategorie

y\_train <- to\_categorical(y\_train, num\_classes = 10)

y\_test <- to\_categorical(y\_test, num\_classes = 10)

# Tworzenie modelu zgodnie z instrukcja laboratoryjna

model <- keras\_model\_sequential()

model <- model %>%

layer\_dense(units = 256, activation = "relu", input\_shape = c(32,32,3)) %>%

layer\_dropout(rate = 0.25) %>%

layer\_dense(units = 128, activation = "relu") %>%

layer\_dropout(rate = 0.25) %>%

layer\_dense(units = 64, activation = "relu") %>%

layer\_dropout(rate = 0.25) %>%

layer\_flatten(input\_shape = c(32,32,3)) %>%

layer\_dense(units = 10, activation = 'softmax')

# Kompilowanie modelu

model %>% compile(

loss = "categorical\_crossentropy",

optimizer = optimizer\_adam(),

metrics = "accuracy"

)

# Informacje o modelu

summary(model)

# Trenowanie modelu

# epochs=10 by zaoszczedzic czas (trenowanie trwa bardzo dlugo)

history <- model %>%

fit(X\_train, y\_train, epochs = 10, batch\_size = 32,

validation\_data = list(X\_test,y\_test), shuffle=TRUE)

# Ocena modelu

model %>%

evaluate(X\_test, y\_test)

# Zapisanie do pliku

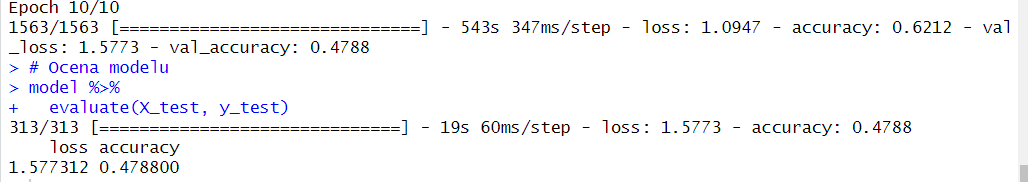
save\_model\_tf(model, "saved\_model/model")

**4. Wynik działania:**

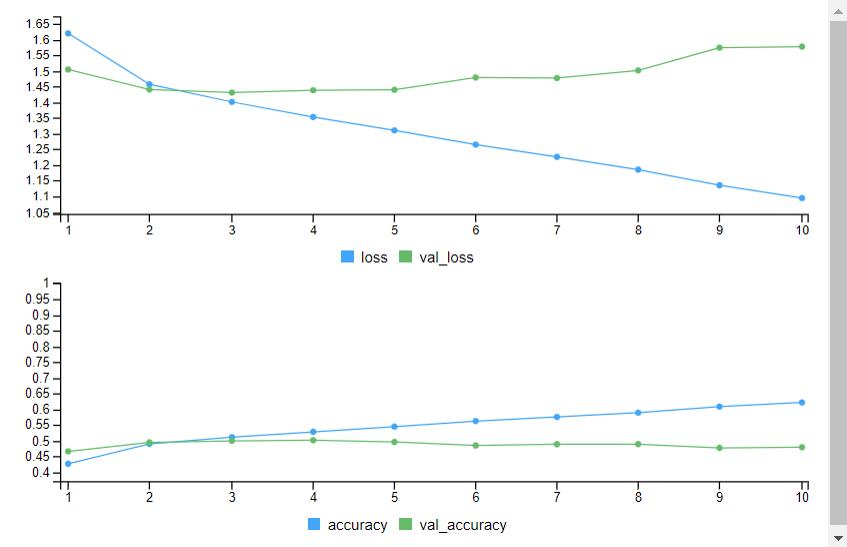
Wyniki poleceń w konsoli można znaleźć w pliku „wyniki\_z\_konsoli.txt”, link do repozytorium poniżej.



Rysunek . Model podczas trenowania.



Rysunek . Uzyskane wyniki na secie treningowym (góra) i testowym (dół).

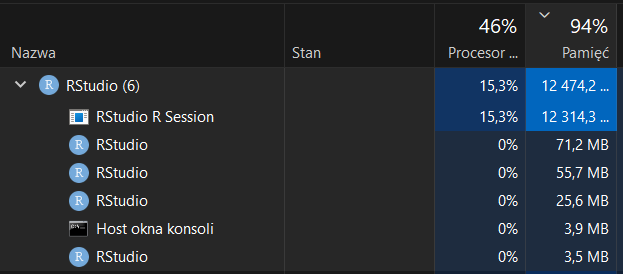


Rysunek . Dokładność i strata modelu podczas trenowania (na osi OX epoka trenowania).

Link do repozytorium: <https://github.com/Stukeley/APU_Lab6>

1. **Wnioski:**

Podczas realizacji laboratorium operowano na bardzo dużych zestawach danych (50000 obrazów), wobec czego środowisko R wykorzystywało ogromną ilość pamięci. Było konieczne regularne, manualne czyszczenie niepotrzebnych zmiennych i wartości tak, aby możliwe było dalsze korzystanie ze środowiska (brak automatycznego GC). Konieczne było również sporadyczne restartowanie środowiska, gdyż nie wszystkie zmienne były usuwane (coraz większe wykorzystanie pamięci z czasem).



Rysunek . Zużycie pamięci rzędu 12GB podczas realizacji ćwiczenia.

Ostateczny model jest efektem kilku eksperymentów związanych z ilością i parametrami warstw. Początkowo testowano model przedstawiony w instrukcji laboratoryjnej, zawierający kilka warstw Dense i Dropout, jednak ostatecznie został on zmodyfikowany, co dało lepsze efekty.

Czas trenowania był dość wysoki, każde pokolenie z ostatecznymi parametrami trenowało się około 10 minut. Standardową wartością ilości epok jest 50, jednak w celu uzyskania wyników szybciej, zastosowano tylko 10 epok.

Uzyskany model jest satysfakcjonujący, gdyż jest znacznie lepszy od modelu losowego (takiego, który losowo wybiera klasę dla danych), mającego w tej sytuacji dokładność w okolicach 10%.