

# **SPRAWOZDANIE**

Zajęcia: Analiza Procesów Ucznienia

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

**Zadanie** dotyczy konstruowania sieci głębokiej w celu klasyfikacji obrazów pobranych ze zbioru danych. Warianty zadania są określone zbiorem danych obrazów, który może być pobrany na stronie <https://keras.io/api/datasets/>

### 1. CIFAR-10

## 2. Wprowadzane dane:

Dane zostały zaczerpnięte z setu 'Cifar10' w pakiecie 'keras'.

```
cifar <- dataset_cifar10()
```

## 3. Wykorzystane komendy:

Poniżej można znaleźć wszystkie wykorzystane komendy:

# Autor: Rafał 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 wejściowych
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 wartości 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 instrukcją laboratoryjną
```

```

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.

```

Epoch 1/10
1563/1563 [=====] - 517s 331ms/step - loss: 1.6196 - accuracy: 0.4261 - val
_loss: 1.5044 - val_accuracy: 0.4657

```

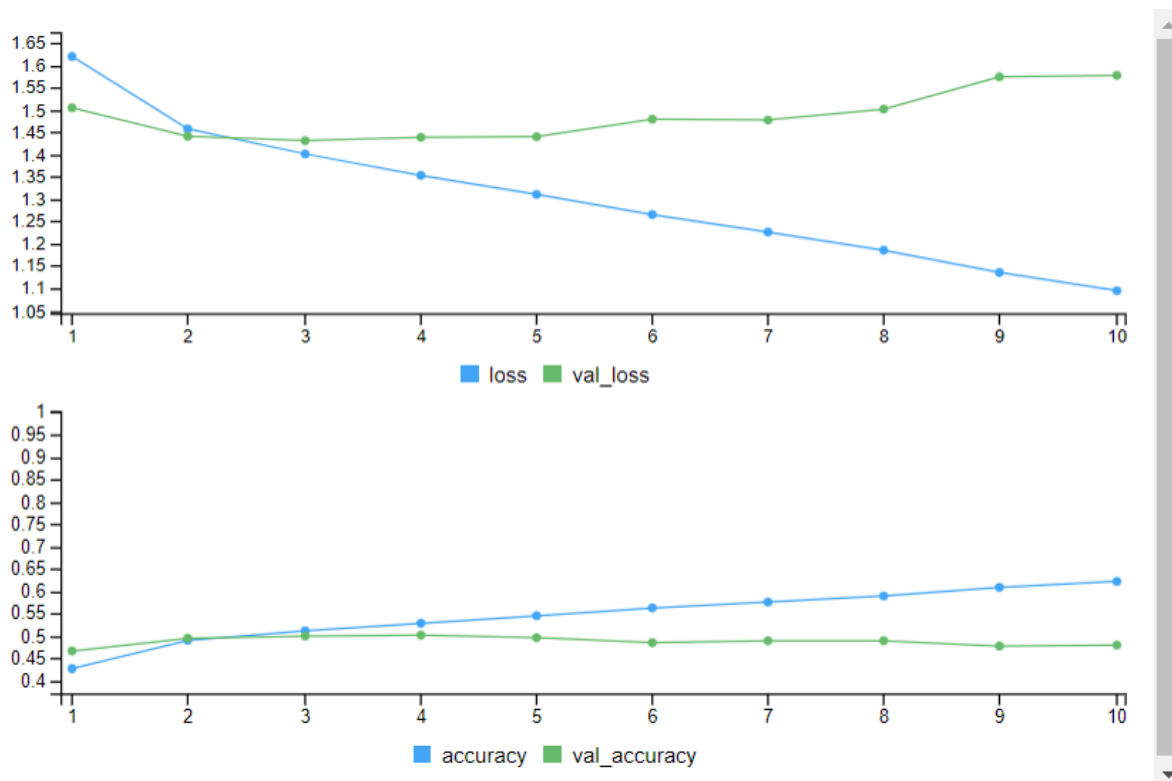
*Rysunek 1. Model podczas trenowania.*

```

Epoch 10/10
1563/1563 [=====] - 543s 347ms/step - loss: 1.0947 - accuracy: 0.6212 - val
_loss: 1.5773 - val_accuracy: 0.4788
> # Ocena modelu
> model %>%
+   evaluate(X_test, y_test)
313/313 [=====] - 19s 60ms/step - loss: 1.5773 - accuracy: 0.4788
      loss accuracy
1.577312 0.478800

```

*Rysunek 2. Uzyskane wyniki na sieci treningowym (góra) i testowym (dół).*



Rysunek 3. Dokładność i strata modelu podczas trenowania (na osi OX epoka trenowania).

Link do repozytorium: [https://github.com/Stukeley/APU\\_Lab6](https://github.com/Stukeley/APU_Lab6)

## 5. 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).

		46%	94%
Nazwa		Procesor ...	Pamięć
<div> <div></div> <div>RStudio (6)</div> </div> <div> <div></div> <div>RStudio R Session</div> </div> <div> <div></div> <div>RStudio</div> </div> <div> <div></div> <div>RStudio</div> </div> <div> <div></div> <div>RStudio</div> </div> <div> <div></div> <div>Host okna konsoli</div> </div> <div> <div></div> <div>RStudio</div> </div>	Stan	15,3%	12 474,2 ...
		15,3%	12 314,3 ...
		0%	71,2 MB
		0%	55,7 MB
		0%	25,6 MB
		0%	3,9 MB
		0%	3,5 MB

Rysunek 4. 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%.