Sprawozdanie

Zajęcia: Analiza procesów uczenia
Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium 5
Data: 10.05.2024

Temat: Uczenie głębokie w R. Klasyfikator obrazów za pomocą Keras

Wariant: 1

Agnieszka Białecka
Informatyka II stopień,
stacjonarne,
1 semestr,
Gr.1a

1. Cel ćwiczenia

Celem ćwiczenia było zapoznanie się z konceptem uczenia głębokiego za pomocą R i pakietu keras.

2. Wstęp teoretyczny

Uczenie głębokie (deep learning) jest zaawansowaną dziedziną sztucznej inteligencji, która naśladuje sposób, w jaki ludzki mózg przetwarza informacje, dzięki czemu maszyny mogą uczyć się z dużych zbiorów danych. W odróżnieniu od tradycyjnych algorytmów uczenia maszynowego, które zazwyczaj wymagają ręcznego tworzenia cech, modele głębokiego uczenia automatycznie odkrywają hierarchiczne reprezentacje danych, co czyni je niezwykle potężnymi w rozwiązywaniu skomplikowanych problemów, takich jak rozpoznawanie obrazów, analiza języka naturalnego czy predykcja szeregów czasowych.

3. Przebieg ćwiczenia

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/.

Wariant: CIFAR-10

Repozytorium GitHub:

https://github.com/Delisolara/APU

```
install.packages("keras")
library("keras")

install.packages("tensorflow")
library("tensorflow")
tensorflow::install_tensorflow()

library(reticulate)
library(keras)

virtualenv_create("myenv")
install_keras(method="virtualenv", envname="myenv")

cifar <- dataset_cifar10()

x_train <- cifar$train$x
x_test <- cifar$test$x
y_train <- cifar$train$y
y_test <- cifar$test$y
```

Zdjęcie 1. Skrypt

```
# wersja liniowa
x_train <- array_reshape(x_train, c(nrow(x_train), 3072))
x_train <- x_train / 255

x_test <- array_reshape(x_test, c(nrow(x_test), 3072))
x_test <- x_test / 255

y_train <- to_categorical(y_train, num_classes = 10)
y_test <- to_categorical(y_test, num_classes = 10)

model <- keras_model_sequential() %>%
  layer_dense(units = 256, activation = "relu", input_shape = c(3072)) %>%
  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_dense(units = 10, activation = "relu")

summary(model)
```

Zdjęcie 2. Skrypt - cd.

```
model %>% compile(
  loss = "categorical_crossentropy",
  optimizer = optimizer_adam(),
  metrics = c("accuracy")
)

history <- model %>%
  fit(
    x_train, y_train,
    epochs = 50,
    batch_size = 128,
    validation_split = 0.15
  )

model %>% evaluate(x_test, y_test)
```

Zdjęcie 3. Skrypt - cd.

```r
#wersja spłaszczona
cifar <- dataset_cifar10()

x_train <- cifar$train$x
x_test <- cifar$test$x
y_train <- cifar$train$y
y_test <- cifar$test$y

x_train <- x_train / 255

x_test <- x_test / 255

y_train <- to_categorical(y_train, num_classes = 10)
y_test <- to_categorical(y_test, num_classes = 10)

model <- keras_model_sequential() %>%
  layer_flatten(input_shape = c(32, 32, 3)) %>%
  layer_dense(units = 128, activation = "relu") %>%
  layer_dense(units = 10, activation = "softmax")

summary(model)
```

Zdjęcie 4. Skrypt - cd.

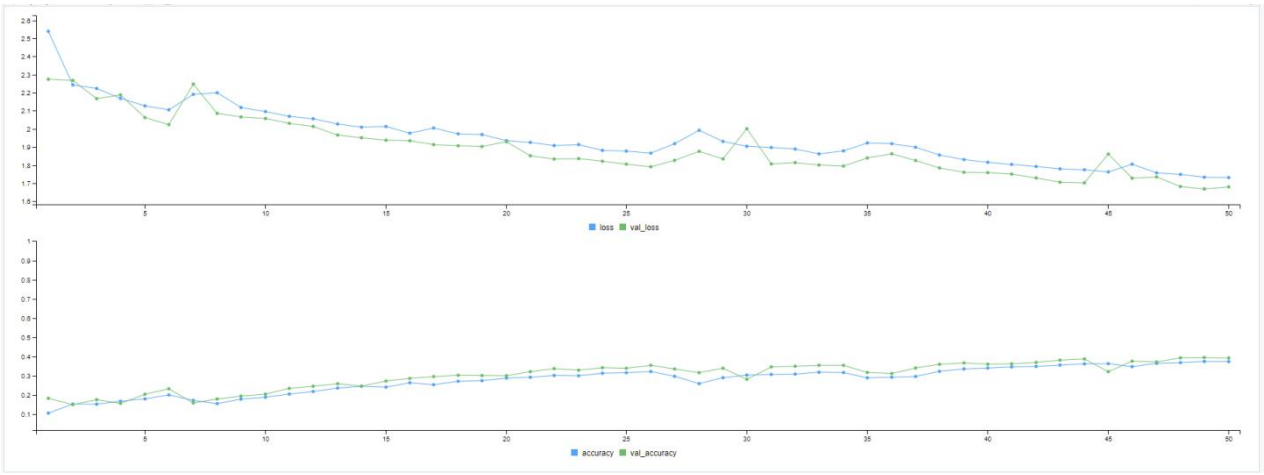
```r
model %>% compile(
  loss = "categorical_crossentropy",
  optimizer = optimizer_adam(),
  metrics = c("accuracy")
)

history <- model %>%
  fit(
    x_train, y_train,
    epochs = 50,
    batch_size = 128,
    validation_split = 0.15
  )

model %>% evaluate(x_test, y_test)

model %>% predict(x_test) %>% k_argmin()
```

Zdjęcie 5. Skrypt - cd.

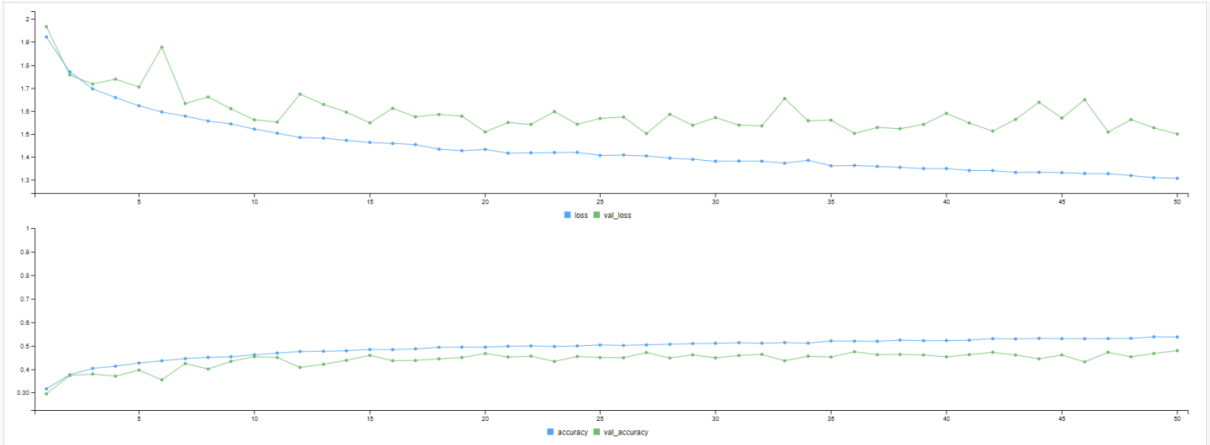


Zdjęcie 6. Model wersji liniowej

```
> model %>% evaluate(x_test, y_test)

  1/313 [..............................] - ETA: 6s - loss: 1.4592 - accuracy: 0.4375
 31/313 [=>............................] - ETA: 0s - loss: 1.5896 - accuracy: 0.4204
 62/313 [====>.........................] - ETA: 0s - loss: 1.5896 - accuracy: 0.4385
 94/313 [========>.....................] - ETA: 0s - loss: 1.6118 - accuracy: 0.4192
124/313 [==========>...................] - ETA: 0s - loss: 1.6032 - accuracy: 0.4226
152/313 [=============>................] - ETA: 0s - loss: 1.6032 - accuracy: 0.4196
182/313 [================>.............] - ETA: 0s - loss: 1.6096 - accuracy: 0.4181
211/313 [====================>.........] - ETA: 0s - loss: 1.6180 - accuracy: 0.4156
240/313 [======================>.......] - ETA: 0s - loss: 1.6160 - accuracy: 0.4173
267/313 [========================>.....] - ETA: 0s - loss: 1.6188 - accuracy: 0.4161
284/313 [==========================>...] - ETA: 0s - loss: 1.6185 - accuracy: 0.4164
311/313 [============================>.] - ETA: 0s - loss: 1.6182 - accuracy: 0.4161
313/313 [==============================] - 1s 2ms/step - loss: 1.6176 - accuracy: 0.4157
    loss accuracy
1.617603 0.415700
```

Zdjęcie 7. Jakość modelu wersji liniowej



Zdjęcie 8. Model wersji spłaszczonej

```
> model %>% evaluate(x_test, y_test)

  1/313 [..............................] - ETA: 5s - loss: 1.1480 - accuracy: 0.5625
 45/313 [===>..........................] - ETA: 0s - loss: 1.4609 - accuracy: 0.4875
 89/313 [=======>......................] - ETA: 0s - loss: 1.4804 - accuracy: 0.4807
131/313 [===========>..................] - ETA: 0s - loss: 1.4678 - accuracy: 0.4907
176/313 [===============>..............] - ETA: 0s - loss: 1.4699 - accuracy: 0.4883
221/313 [====================>.........] - ETA: 0s - loss: 1.4717 - accuracy: 0.4900
266/313 [========================>.....] - ETA: 0s - loss: 1.4747 - accuracy: 0.4884
311/313 [============================>.] - ETA: 0s - loss: 1.4738 - accuracy: 0.4888
313/313 [==============================] - 0s 1ms/step - loss: 1.4738 - accuracy: 0.4888
    loss accuracy
1.473762 0.488800
```

Zdjęcie 9. Jakość modelu wersji spłaszczonej

4. Podsumowanie

Przeprowadzone ćwiczenie pozwoliło na dogłębne zapoznanie się z kluczowymi koncepcjami głębokiego uczenia. W ramach tego ćwiczenia zostały zaimplementowane modele głębokiego uczenia, korzystając z biblioteki Keras w języku R.