SPRAWOZDANIE

Zajęcia: Analiza Procesów Uczenia
Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium 6
Data 07.06.2023
Temat: "Ucznie głębokie w R. Klasyfikator obrazów za pomocą Keras"
Wariant: 2

Szymon Białek
Informatyka II stopień
stacjonarne
1 semestr,
Gr.1

Wszystkie pliki i komendy można obejrzeć pod linkiem:
https://github.com/NynyNoo/Analiza-procesow-uczenia/tree/main/lab6

Polecenie

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/

Podejście liniowe

Wykorzystane komendy oraz wyniki działania programu

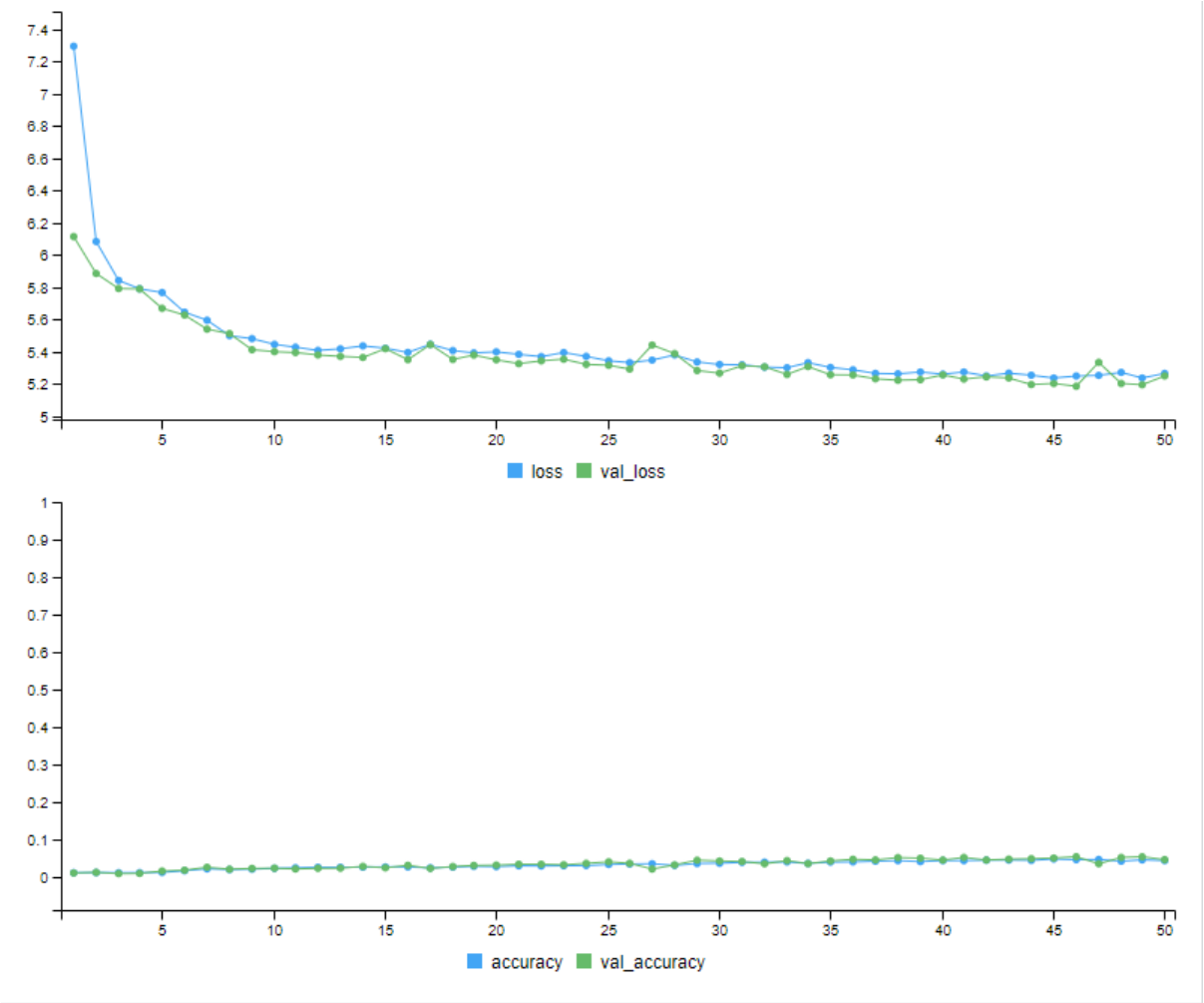
```
> setwd("D:/MGR/APU/lab6")
> library(reticulate)
> use_condaenv("apu")
> library("keras")
> library("tensorflow")
>
> #load data cifar 100
> cifar <- dataset_cifar100()
>
> x_train <- cifar$train$x
> x_test <- cifar$test$x
> y_train <- cifar$train$y
> y_test <- cifar$test$y
>
> #------------------------------------------------- wersja liniowa
>
> #set up data
> #change matrix shape
> x_train <- array_reshape(x_train, c(nrow(x_train), 3072))
> #normalize
> x_train <- x_train / 255
>
> x_test <- array_reshape(x_test, c(nrow(x_test), 3072))
> x_test <- x_test / 255
>
> #set classes
> y_train <- to_categorical(y_train, num_classes = 100)
> y_test <- to_categorical(y_test, num_classes = 100)
>
> #256 neurons, dropout rate 0.25
> 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 = 100, activation = "relu")
```

```
> summary(model)
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_3 (Dense)             (None, 256)               786688
 dropout_2 (Dropout)         (None, 256)               0
 dense_2 (Dense)             (None, 128)               32896
 dropout_1 (Dropout)         (None, 128)               0
 dense_1 (Dense)             (None, 64)                8256
 dropout (Dropout)           (None, 64)                0
 dense (Dense)               (None, 100)               6500
=================================================================
Total params: 834,340
Trainable params: 834,340
Non-trainable params: 0
_________________________________________________________________
```

```
> #set model parameters
> model %>% compile(
+     loss = "categorical_crossentropy",     #calculate loss
+     optimizer = optimizer_adam(),          #optimization
+     metrics = c("accuracy")                #accuracy
+ )
>
> #train model
> history <- model %>%
+     fit(
+         x_train, y_train,          #input
+         epochs = 50,
+         batch_size = 128,          #128 pictures
+         validation_split = 0.15
+     )
```

```
> #check quality
> model %>% evaluate(x_test, y_test)
313/313 [==============================] - 1s 2ms/step - loss: 5.2461 - accuracy: 0.0438
    loss accuracy
5.246055 0.043800
```

Wizualizacja Danych



Podejście spłaszczone

```
> setwd("D:/MGR/APU/lab6")
> library(reticulate)
> use_condaenv("apu")
> library("keras")
> library("tensorflow")
>
> cifar <- dataset_cifar100()
>
> x_train <- cifar$train$x
> x_test <- cifar$test$x
> y_train <- cifar$train$y
> y_test <- cifar$test$y
>
> #set up data
> #normalize
> x_train <- x_train / 255
> x_test <- x_test / 255
>
> #set classes
> y_train <- to_categorical(y_train, num_classes = 100)
> y_test <- to_categorical(y_test, num_classes = 100)
>
> #create model
> model <- keras_model_sequential() %>%
+     layer_flatten(input_shape = c(32, 32, 3)) %>%
+     layer_dense(units = 128, activation = "relu") %>%
+     layer_dense(units = 100, activation = "softmax")

> #print model
> summary(model)
Model: "sequential"
_________________________________________________________________________________
 Layer (type)                        Output Shape                      Param #
=================================================================================
 flatten (Flatten)                   (None, 3072)                      0
 dense_1 (Dense)                     (None, 128)                       393344
 dense (Dense)                       (None, 100)                       12900
=================================================================================
Total params: 406,244
Trainable params: 406,244
Non-trainable params: 0
_________________________________________________________________________________
>
> #set model parameters
> model %>% compile(
+     loss = "categorical_crossentropy",      #calculate loss
+     optimizer = optimizer_adam(),           #optimization
+     metrics = c("accuracy")                 #accuracy
+ )
>
> #train model
> history <- model %>%
+     fit(
+         x_train, y_train,
+         epochs = 50,
+         batch_size = 128,
+         validation_split = 0.15
+     )
```

```
> #check model quality
> model %>% evaluate(x_test, y_test)
313/313 [==============================] - 0s 1ms/step - loss: 3.7560 - accuracy: 0.1353
    loss accuracy
3.756027 0.135300
>
> #predict
> model %>% predict(x_test) %>% k_argmin()
313/313 [==============================] - 0s 1ms/step
tf.Tensor([53 53 68 ... 20 49 52], shape=(10000), dtype=int64)
```

Wizualizacja Danych