ЛАБОРАТОРНАЯ РАБОТА №14

Использование глубокого обучения для классификации изображений с помощью keras

Цель работы: научиться работать с библиотекой keras для построения нейронных сетей для классификации изображений.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Пакет **EBImage**. Пакет предназначен для работы с изображениями, он не входит в состав официального репозитория CRAN, однако содержится в проекте Bioconductor. Для работы с пакетами проекта необходимо установить пакет BiocManager из репозитория CRAN, дальнейшая установка пакетов производится с помощью функции install().

Пакет **keras**. Пакет представляет собой R интерфейс к библиотеке Keras. Пакет поддерживает сложные сетевые структуры, используемые для реализации глубокого обучения. Более подробная информация о пакете представлена на сайте https://keras.rstudio.com.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Задание 1 – Используя тестовый набор данных повторите задачу распознавания объектов с помощью программного кода, представленного ниже. Дайте письменно объяснение параметров, используемых в примере. Для понимания параметров обратитесь к документации пакетов EBImages и keras.

```
# Load Packages
library(EBImage)
library(keras)

# Read images
pics <- c('p1.jpg', 'p2.jpg', 'p3.jpg', 'p4.jpg', 'p5.jpg', 'p6.jpg',
          'c1.jpg', 'c2.jpg', 'c3.jpg', 'c4.jpg', 'c5.jpg', 'c6.jpg')
mypic <- list()
for (i in 1:12) {mypic[[i]] <- readImage(pics[i])}

# Explore
print(mypic[[1]])
display(mypic[[1]])
summary(mypic[[1]])
hist(mypic[[2]])
str(mypic)

# Resize
for (i in 1:12) {mypic[[i]] <- resize(mypic[[i]], 28, 28)}

# Reshape
for (i in 1:12) {mypic[[i]] <- array_reshape(mypic[[i]], c(28, 28, 3))}

# Row Bind
trainx <- NULL
for (i in 1:5) {trainx <- rbind(trainx, mypic[[i]])}
str(trainx)
for (i in 7:11) {trainx <- rbind(trainx, mypic[[i]])}
str(trainx)
testx <- rbind(mypic[[6]], mypic[[12]])
trainy <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
testy <- c(0, 1)

# One Hot Encoding
trainLabels <- to_categorical(trainy)
```

```
testLabels <- to_categorical(testy)

# Model
model <- keras_model_sequential()
model %>%
        layer_dense(units = 256, activation = 'relu', input_shape = c(2352)) %>%
        layer_dense(units = 128, activation = 'relu') %>%
        layer_dense(units = 2, activation = 'softmax')
summary(model)

# Compile
model %>%
        compile(loss = 'binary_crossentropy',
                optimizer = optimizer_rmsprop(),
                metrics = c('accuracy'))

# Fit Model
history <- model %>%
        fit(trainx,
            trainLabels,
            epochs = 30,
            batch_size = 32,
            validation_split = 0.2)

# Evaluation & Prediction - train data
model %>%
  evaluate(trainx, trainLabels)
pred <- model %>%
  predict_classes(trainx)
table(Predicted = pred, Actual = trainy)
prob <- model %>%
  predict_proba(trainx)
cbind(prob, Prected = pred, Actual = trainy)

# Evaluation & Prediction - test data
model %>%
  evaluate(testx, testLabels)
pred <- model %>%
  predict_classes(testx)
```

Задание 2 – Постройте нейронную сеть для классификации изображений. Повторите первое задание для вашего набора данных.

Контрольные вопросы:

1. Опишите параметры, используемые в первом задании.
2. Опишите процесс подготовки изображений для второго задания.