

## SPRAWOZDANIE

Zajęcia: Analiza procesów uczenia

Prowadzący: prof. dr hab. inż. Vasyl Martsenyuk

Laboratorium Nr 6 Data 26.05.2023 Temat: Uczenie głębokie w R. Klasyfikator obrazów za pomocą Keras Wariant 4	Maksymilian Grygiel Informatyka II stopień, stacjonarne, Semestr I, gr.1a
------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------

Link do repozytorium: <https://github.com/Maksiolo20/APU>

### Zadania:

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/>

4. Fashion-MNIST database of fashion articles

### Wykonanie

### zadania:

Instalacja

pakietów:

```
> install.packages("tensorflow")
> library(tensorflow)
> install_tensorflow()
> install.packages("keras")
> library(keras)
> install_keras()
```

Ładowanie danych fashionMNIST:

```
> fashionMnist <- dataset_fashion_mnist()
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-idx1-ubyte.gz
29515/29515 [=====] - 0s 1us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-idx3-ubyte.gz
26421880/26421880 [=====] - 4s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-idx1-ubyte.gz
5148/5148 [=====] - 0s 0s/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-idx3-ubyte.gz
4422102/4422102 [=====] - 1s 0us/step
```

Rozpakowanie

danych:

```
> X_train <- fashionMnist$train$x
> X_test <- fashionMnist$test$x
> Y_train <- fashionMnist$train$y
> Y_test <- fashionMnist$test$y
```

Normalizacja danych:

```
> X_train_norm <- X_train/255
> X_test_norm <- X_test/255
```

Ustawienie klas:

```
> Y_train_class <- to_categorical(Y_train,num_classes = 100)
> Y_test_class <- to_categorical(Y_test,num_classes = 100)
```

Utworzenie

modelu:

```
> model <- keras_model_sequential() %>%
+   layer_flatten(input_shape = c(32,32,3)) %>%
+   layer_dense(units=128,activation = "relu")%>%
+   layer_dense(units=100,activation = "softmax")
```

To enable them in other operations, rebuild tensorflow with the appropriate compiler flags.

```
> 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

Ustawienie

parametrów

modelu:

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

Sprawdzenie

jakości

modelu:

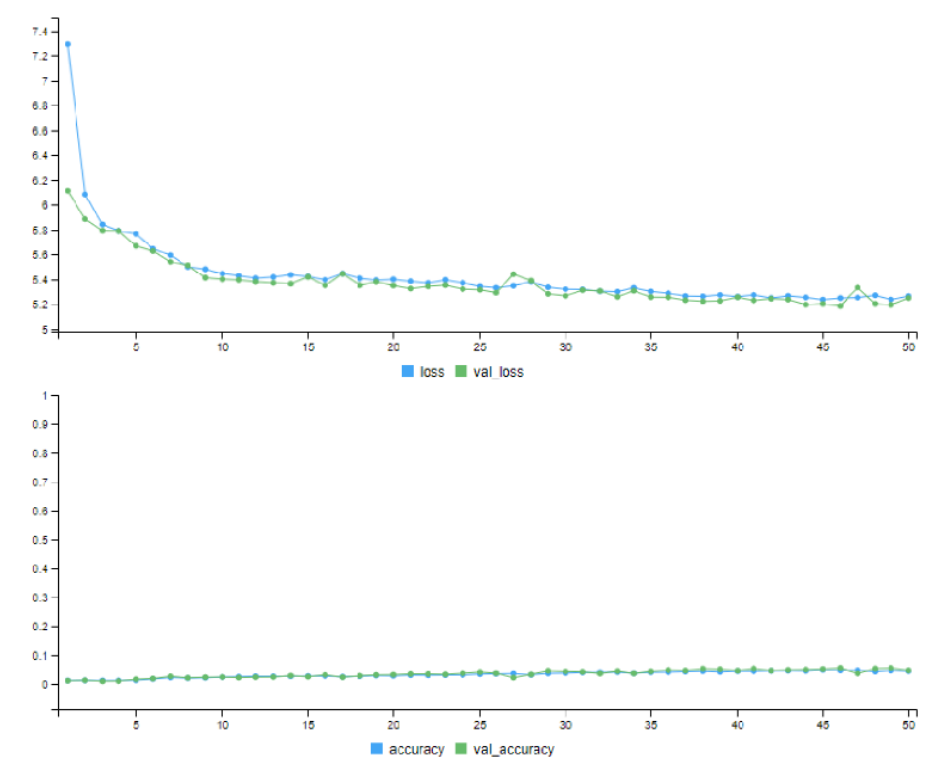
```
> model %>% evaluate(X_test_norm,Y_test_class)
```

```
313/313 [=====] - 1s 2ms/step - loss: 5.2461 - accuracy: 0.043  
loss accuracy  
5.246055 0.043800
```

Przewidzenie:

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

Wizualizacja:



## Wnioski:

Sprawozdanie z R Studio dotyczące łączenia głębokiego w R i klasyfikatora obrazów przy użyciu Kerasa wykazało, że R jest wszechstronnym narzędziem do budowy zaawansowanych modeli uczenia maszynowego. Przy użyciu biblioteki Keras, możliwe jest skuteczne trenowanie klasyfikatorów obrazów, co otwiera nowe perspektywy w analizie wizualnej danych. Integracja tych technologii zapewnia potężne narzędzia do rozwiązywania problemów związanych z przetwarzaniem obrazów przy użyciu języka R.