Basic Image Classification with TensorFlow

R Sample Code

Libraries

Load TensorFlow and Keras libraries.

```r
library(tensorflow)
library(keras)
```

Dataset

Load the MNIST dataset and scale the pixel values to the range of [0,1].

```r
c(c(x_train, y_train), c(x_test, y_test)) %<-% keras::dataset_mnist()
x_train <- x_train / 255
x_test <-  x_test / 255
```

Artificial Neural Network Model

Define a fully-connected feedforward artificial neural network architecture with a single hidden layer composed of 128 neurons. Set the dropout rate at 20%.

```r
model <- keras_model_sequential(input_shape = c(28, 28)) %>%
  layer_flatten() %>%
  layer_dense(128, activation = "relu") %>%
  layer_dropout(0.2) %>%
  layer_dense(10)
```

Define the loss function, the optimizer, and the evaluation metric for the training.

```r
loss_fn <- loss_sparse_categorical_crossentropy(from_logits = TRUE)
```

```r
model %>% compile(
  optimizer = "adam",
  loss = loss_fn,
  metrics = "accuracy"
)
```

Model Training and Evaluation

Fit the model on the training set.

```
model %>% fit(x_train, y_train, epochs = 5)
```

```
## Epoch 1/5
## 1875/1875 - 3s - loss: 0.2953 - accuracy: 0.9134 - 3s/epoch - 1ms/step
## Epoch 2/5
## 1875/1875 - 2s - loss: 0.1406 - accuracy: 0.9578 - 2s/epoch - 1ms/step
## Epoch 3/5
## 1875/1875 - 3s - loss: 0.1050 - accuracy: 0.9682 - 3s/epoch - 1ms/step
## Epoch 4/5
## 1875/1875 - 2s - loss: 0.0871 - accuracy: 0.9728 - 2s/epoch - 1ms/step
## Epoch 5/5
## 1875/1875 - 2s - loss: 0.0753 - accuracy: 0.9768 - 2s/epoch - 1ms/step
```

Evaluate the classification performance of the model on the test set.

```
model %>% evaluate(x_test,  y_test, verbose = 2)
```

```
## 313/313 - 0s - loss: 0.0701 - accuracy: 0.9790 - 321ms/epoch - 1ms/step
```

```
##       loss   accuracy
## 0.07006038 0.97899997
```