capstone_model2

2022-12-17

Helper packages

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(keras)
library(tfruns)

library(tfestimators)
```

```
## tfestimators is not recomended for new code. It is only compatible with Tensorflow version 1, and is
```

Import MNIST training data

```
mnist <- dataset_mnist()
X_train <- mnist$train$x
X_test <- mnist$test$x
y_train <- mnist$train$y
y_test <- mnist$test$y
```

#reshaping the dataset

```
X_train <- array_reshape(X_train, c(nrow(X_train), 784))
X_train <- X_train / 255

X_test <- array_reshape(X_test, c(nrow(X_test), 784))
X_test <- X_test / 255
```

```r
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 = "sigmoid", input_shape = c(784)) %>%
  layer_dropout(rate = 0.2) %>%
  layer_dense(units = 128, activation = "sigmoid") %>%
  layer_dropout(rate = 0.2) %>%
  layer_dense(units = 128, activation = "sigmoid") %>%
  layer_dropout(rate = 0.2) %>%
  layer_dense(units = 64, activation = "sigmoid") %>%
  layer_dropout(rate = 0.2) %>%
  layer_dense(units = 10, activation = "softmax") %>%
```

# Backpropagation

```r
  compile(
    loss = "categorical_crossentropy",
    optimizer = optimizer_rmsprop(),
    metrics = c("accuracy")
  )
```

#compiling the model

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

history <- model %>%
  fit(X_train, y_train, epochs = 10, batch_size = 128, validation_split = 0.15)
```

#model evaluation

```r
model %>%
  evaluate(X_test, y_test)
```

#model prediction

```r
model %>%
  predict_classes(X_test)
```