HW 5 MATH 868

James DG

Fall 2022

Homework problems

1. In the example, sparse_categorical_crossentropy is used. Develop a new model, by keeping everything of the model the same except using categorical_crossentropy. Note that the data need to be processed, in order for categorical_crossentropy to work. Train the model and print out the training history.

```
library(keras)
```

```
## Warning: package 'keras' was built under R version 4.1.2
```

```
fashion_mnist <- dataset_fashion_mnist()
```

```
## Loaded Tensorflow version 2.9.2
```

```
c(train_images, train_labels) %<-% fashion_mnist$train
c(test_images, test_labels) %<-% fashion_mnist$test

class_names = c('T-shirt/top',
                'Trouser',
                'Pullover',
                'Dress',
                'Coat',
                'Sandal',
                'Shirt',
                'Sneaker',
                'Bag',
                'Ankle boot')

par(mfrow=c(5,5), mar=c(0,0,1,0))
for(i in sample(1:60000,25,replace=FALSE)){
    dat <- train_images[i, ,]
    image(t(dat)[, 28:1], axes=FALSE,col=grey(seq(1,0,length=256)), ylab='', xlab='',
          main=paste(class_names[train_labels[i] + 1])
           )
}
```

```
model <- keras_model_sequential()
model %>%
  layer_flatten(input_shape = c(28, 28)) %>%
  layer_dense(units = 128, activation = 'relu') %>%
  layer_dense(units = 10, activation = 'softmax')
```

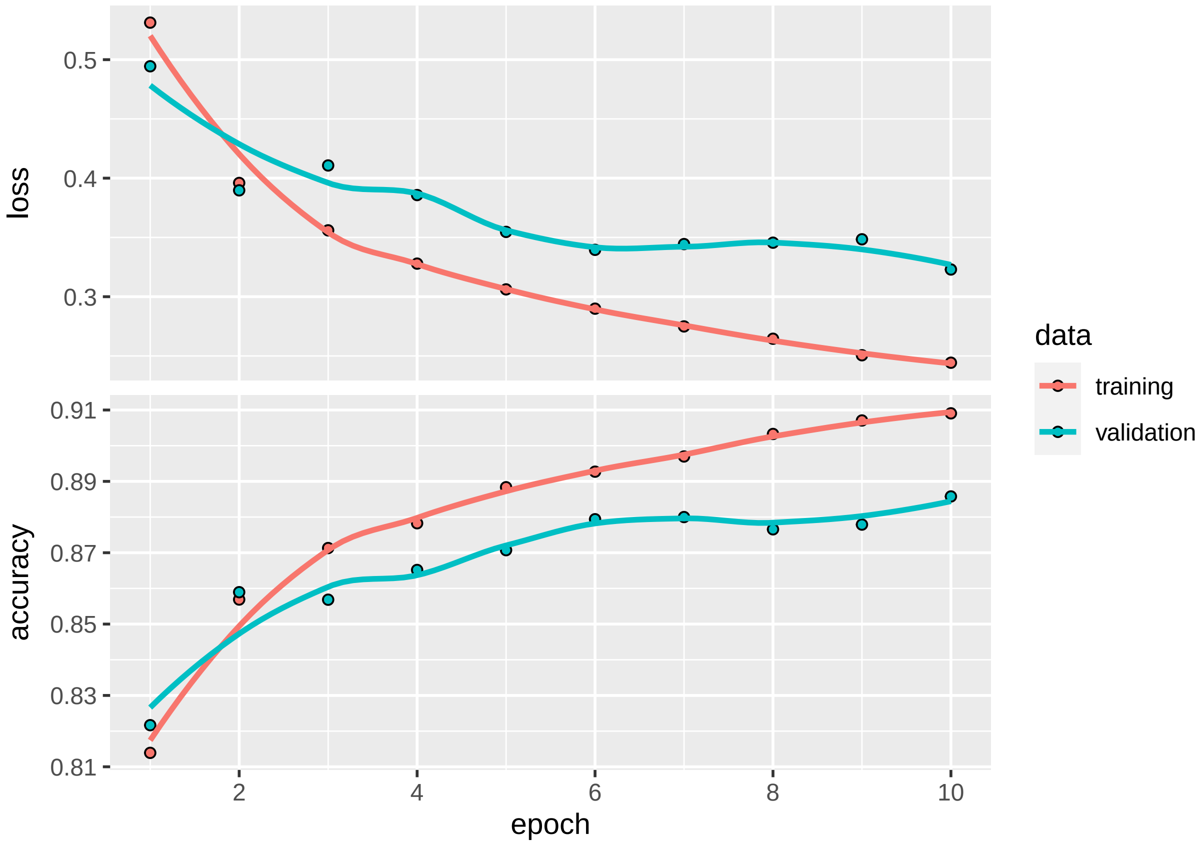
The labels are converted to one-hot encoding by to_categorical() for which the matching loss is categorical_crossentropy.

```
train_images <- train_images / 255
test_images <- test_images / 255

y_train = to_categorical(train_labels)
y_test = to_categorical(test_labels)
```

```
model %>% compile(
  optimizer = 'adam',
  loss = 'categorical_crossentropy',
  metrics = c('accuracy'))

history <- model %>% fit(train_images, y_train, epochs = 10, verbose = 2, validation_split = 0.3)
plot(history)
```



```
##     loss accuracy
## 0.353598 0.875400
```

2. Apply the newly trained model to the last 10 graphs of the testing data and report accuracy rate.

```
score <- model %>% evaluate(tail(test_images,10),tail(y_test,10), verbose = 0)

cat('Test accuracy:', score["accuracy"]*100, "% \n")
```

```
## Test accuracy: 100 %
```