Assignment 4

Mitheysh Asokan, Eungjoo Kim

Question 1: Classification using NNets

1.1. Get the data

```
mnist <- dataset_fashion_mnist()

x_train <- mnist$train$x
y_train <- mnist$train$y
x_test <- mnist$test$x
y_test <- mnist$test$y

dim(x_train)
```

```
## [1] 60000    28    28
```

```
dim(x_test)
```

```
## [1] 10000    28    28
```

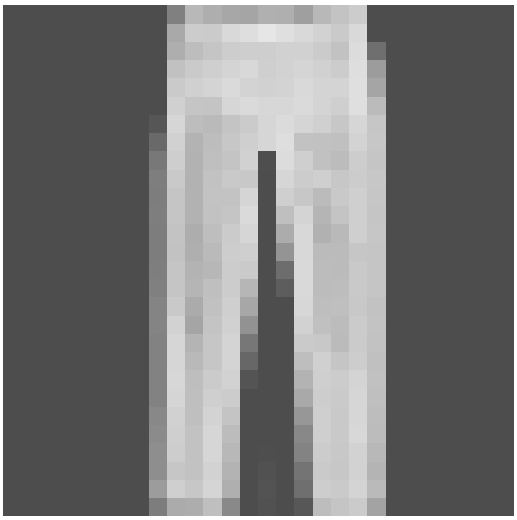
```
y_test  <- y_test[1:2560]
x_train <- x_train[1:10240,,]
x_test  <- x_test[1:2560,, ]

y_train <- y_train[1:10240]
y_test  <- y_test[1:2560]
```

1.2 Plot

Plot of a trouser (class 1)

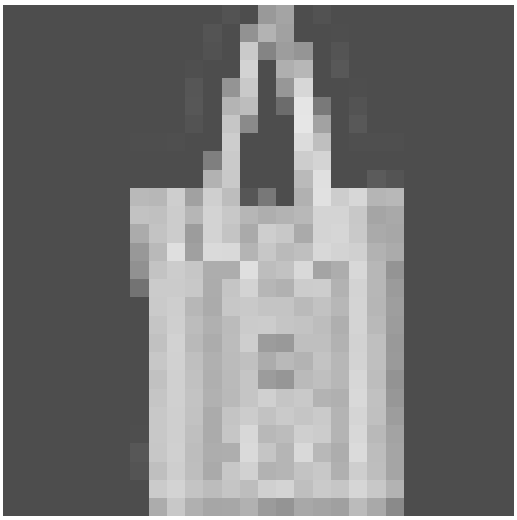
```
digit <- x_train[17,28:1,1:28]
par(pty="s") # for keeping the aspect ratio 1:1
image(t(digit), col = gray.colors(256), axes = FALSE)
```



1.2 Plot

Plot of a bag (class 9)

```
digit <- x_train[58,28:1,1:28]
par(pty="s") # for keeping the aspect ratio 1:1
image(t(digit), col = gray.colors(256), axes = FALSE)
```



1.2 Plot

Plot of an Ankle boot (class 10)

```
digit <- x_train[12,28:1,1:28]
par(pty="s") # for keeping the aspect ratio 1:1
image(t(digit), col = gray.colors(256), axes = FALSE)
```

1.3 Process the dataset

```
# reshape
x_train <- array_reshape(x_train, c(nrow(x_train), 784))
x_test <- array_reshape(x_test, c(nrow(x_test), 784))
# rescale
x_train <- x_train / 255
x_test <- x_test / 255

# convert binary to categorical
y_train <- to_categorical(y_train, 10)
y_test <- to_categorical(y_test, 10)
head(y_train)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    0    0    0    0    0    0    0    0    0     1
## [2,]    1    0    0    0    0    0    0    0    0     0
## [3,]    1    0    0    0    0    0    0    0    0     0
## [4,]    0    0    0    1    0    0    0    0    0     0
## [5,]    1    0    0    0    0    0    0    0    0     0
## [6,]    0    0    1    0    0    0    0    0    0     0
```

1.4 Fit a Shallow Network

32 Hidden neurons with relu

```
set.seed(100)

model <- keras_model_sequential()
model %>%
  layer_dense(units = 32, activation = 'relu', input_shape = c(784)) %>%
  layer_dense(units = 10, activation = 'softmax')

summary(model)
```

```
## Model: "sequential"
## ________________________________________________________________________________
## Layer (type)                        Output Shape                    Param #
## ================================================================================
## dense_1 (Dense)                     (None, 32)                      25120
## ________________________________________________________________________________
## dense (Dense)                       (None, 10)                      330
## ================================================================================
## Total params: 25,450
## Trainable params: 25,450
## Non-trainable params: 0
## ________________________________________________________________________________
```

```
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 = 0.2
)

model %>% evaluate(x_test, y_test)
```

```
##      loss  accuracy
## 0.4517234 0.8488281
```

128 Hidden neurons with relu

```
set.seed(100)

model <- keras_model_sequential()
model %>%
  layer_dense(units = 128, activation = 'relu', input_shape = c(784)) %>%
  layer_dense(units = 10, activation = 'softmax')

summary(model)
```

```
## Model: "sequential_1"
## ________________________________________________________________________________
## Layer (type)                        Output Shape                    Param #
## ================================================================================
## dense_3 (Dense)                     (None, 128)                     100480
## ________________________________________________________________________________
## dense_2 (Dense)                     (None, 10)                      1290
## ================================================================================
## Total params: 101,770
## Trainable params: 101,770
## Non-trainable params: 0
## ________________________________________________________________________________
```

```
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 = 0.2
)

model %>% evaluate(x_test, y_test)
```

```
##      loss  accuracy
## 0.4249544 0.8550781
```

256 Hidden neurons with relu

```
set.seed(100)

model <- keras_model_sequential()
model %>%
  layer_dense(units = 256, activation = 'relu', input_shape = c(784)) %>%
  layer_dense(units = 10, activation = 'softmax')

summary(model)
```

```
## Model: "sequential_2"
## ________________________________________________________________________________
## Layer (type)                        Output Shape                    Param #
## ================================================================================
## dense_5 (Dense)                     (None, 256)                     200960
## ________________________________________________________________________________
## dense_4 (Dense)                     (None, 10)                      2570
## ================================================================================
## Total params: 203,530
## Trainable params: 203,530
## Non-trainable params: 0
```

```
## _________________________________________________________________________
```

```
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 = 0.2
)

model %>% evaluate(x_test, y_test)
```

```
##      loss  accuracy
## 0.4506355 0.8378906
```

32 Hidden neurons with sigmoid

```
set.seed(100)

model <- keras_model_sequential()
model %>%
  layer_dense(units = 32, activation = 'sigmoid', input_shape = c(784)) %>%
  layer_dense(units = 10, activation = 'softmax')

summary(model)
```

```
## Model: "sequential_3"
## _________________________________________________________________________
## Layer (type)                 Output Shape              Param #
## =========================================================================
## dense_7 (Dense)              (None, 32)                25120
## _________________________________________________________________________
## dense_6 (Dense)              (None, 10)                330
## =========================================================================
## Total params: 25,450
## Trainable params: 25,450
## Non-trainable params: 0
## _________________________________________________________________________
```

```
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 = 0.2
```

```
)

model %>% evaluate(x_test, y_test)
```

```
##      loss  accuracy
## 0.5342082 0.8281250
```

128 Hidden neurons with sigmoid

```
set.seed(100)

model <- keras_model_sequential()
model %>%
  layer_dense(units = 128, activation = 'sigmoid', input_shape = c(784)) %>%
  layer_dense(units = 10, activation = 'softmax')

summary(model)
```

```
## Model: "sequential_4"
## ________________________________________________________________________________
## Layer (type)                        Output Shape                    Param #
## ================================================================================
## dense_9 (Dense)                     (None, 128)                     100480
## ________________________________________________________________________________
## dense_8 (Dense)                     (None, 10)                      1290
## ================================================================================
## Total params: 101,770
## Trainable params: 101,770
## Non-trainable params: 0
## ________________________________________________________________________________
```

```
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 = 0.2
)

model %>% evaluate(x_test, y_test)
```

```
##     loss accuracy
## 0.451330 0.846875
```

256 Hidden neurons with sigmoid

```
set.seed(100)

model <- keras_model_sequential()
```

```
model %>%
  layer_dense(units = 256, activation = 'sigmoid', input_shape = c(784)) %>%
  layer_dense(units = 10, activation = 'softmax')

summary(model)
```

```
## Model: "sequential_5"
## ________________________________________________________________________________
## Layer (type)                        Output Shape                    Param #
## ================================================================================
## dense_11 (Dense)                    (None, 256)                     200960
## ________________________________________________________________________________
## dense_10 (Dense)                    (None, 10)                      2570
## ================================================================================
## Total params: 203,530
## Trainable params: 203,530
## Non-trainable params: 0
## ________________________________________________________________________________
```

```
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 = 0.2
)

model %>% evaluate(x_test, y_test)
```

```
##      loss  accuracy
## 0.4433200 0.8453125
```

1.5 Fit a Deep Neural Network

128 and 32 Hidden neurons with relu

```
set.seed(100)

model <- keras_model_sequential()
model %>%
  layer_dense(units = 128, activation = 'relu', input_shape = c(784)) %>%
  layer_dense(units = 32, activation = 'relu') %>%
  layer_dense(units = 10, activation = 'softmax')

summary(model)
```

```
## Model: "sequential_6"
## ________________________________________________________________________________
## Layer (type)                        Output Shape                    Param #
## ================================================================================
## dense_14 (Dense)                    (None, 128)                     100480
## ________________________________________________________________________________
## dense_13 (Dense)                    (None, 32)                      4128
## ________________________________________________________________________________
## dense_12 (Dense)                    (None, 10)                      330
## ================================================================================
## Total params: 104,938
## Trainable params: 104,938
## Non-trainable params: 0
## ________________________________________________________________________________
```

```
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 = 0.2
)

model %>% evaluate(x_test, y_test)
```

```
##      loss  accuracy
## 0.4379967 0.8503906
```

128 and 32 Hidden neurons with sigmoid

```
set.seed(100)

model <- keras_model_sequential()
model %>%
  layer_dense(units = 128, activation = 'sigmoid', input_shape = c(784)) %>%
```

```
  layer_dense(units = 32, activation = 'sigmoid') %>%
  layer_dense(units = 10, activation = 'softmax')

summary(model)
```

```
## Model: "sequential_7"
## ________________________________________________________________________________
## Layer (type)                        Output Shape                    Param #
## ================================================================================
## dense_17 (Dense)                    (None, 128)                     100480
## ________________________________________________________________________________
## dense_16 (Dense)                    (None, 32)                      4128
## ________________________________________________________________________________
## dense_15 (Dense)                    (None, 10)                      330
## ================================================================================
## Total params: 104,938
## Trainable params: 104,938
## Non-trainable params: 0
## ________________________________________________________________________________
```

```
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 = 0.2
)

model %>% evaluate(x_test, y_test)
```

```
##      loss  accuracy
## 0.5073998 0.8292969
```

256 and 128 Hidden neurons with relu

```
set.seed(100)

model <- keras_model_sequential()
model %>%
  layer_dense(units = 256, activation = 'relu', input_shape = c(784)) %>%
  layer_dense(units = 128, activation = 'relu') %>%
  layer_dense(units = 10, activation = 'softmax')

summary(model)
```

```
## Model: "sequential_8"
## ________________________________________________________________________________
## Layer (type)                        Output Shape                    Param #
```

```
## ================================================================================
## dense_20 (Dense)                    (None, 256)                       200960
## ________________________________________________________________________________
## dense_19 (Dense)                    (None, 128)                       32896
## ________________________________________________________________________________
## dense_18 (Dense)                    (None, 10)                        1290
## ================================================================================
## Total params: 235,146
## Trainable params: 235,146
## Non-trainable params: 0
## ________________________________________________________________________________
```

```
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 = 0.2
)

model %>% evaluate(x_test, y_test)
```

```
##      loss  accuracy
## 0.4454214 0.8421875
```

256 and 128 Hidden neurons with sigmoid

```
set.seed(100)

model <- keras_model_sequential()
model %>%
  layer_dense(units = 256, activation = 'sigmoid', input_shape = c(784)) %>%
  layer_dense(units = 128, activation = 'sigmoid') %>%
  layer_dense(units = 10, activation = 'softmax')

summary(model)
```

```
## Model: "sequential_9"
## ________________________________________________________________________________
## Layer (type)                        Output Shape                      Param #
## ================================================================================
## dense_23 (Dense)                    (None, 256)                       200960
## ________________________________________________________________________________
## dense_22 (Dense)                    (None, 128)                       32896
## ________________________________________________________________________________
## dense_21 (Dense)                    (None, 10)                        1290
## ================================================================================
## Total params: 235,146
```

```
## Trainable params: 235,146
## Non-trainable params: 0
## ____________________________________________________________________________
```

```
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 = 0.2
)

model %>% evaluate(x_test, y_test)
```

```
##      loss  accuracy
## 0.4320598 0.8468750
```