MATH 686: HW 8
Text Classification with TF Hub (nnlm-en-dim128)

James DG

Fall 2022

Classify movie reviews as positive or negative using the text of the review. Based on the course handout, retrain the model and evaluate accuracy for the testing data.

- adding an early stopping condition to the training process
- use a pre-trained text embedding model from TensorFlow Hub called google/nnlm-en-dim128/2

```
if (dir.exists("aclImdb/"))
  unlink("aclImdb/", recursive = TRUE)
url <- "https://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz"
dataset <- get_file(
  "aclImdb_v1",
  url,
  untar = TRUE,
  cache_dir = '.',
  cache_subdir = ''
)
```

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

```
unlink("aclImdb/train/unsup/", recursive = TRUE)
```

```
batch_size <- 512
seed <- 42

train_data <- text_dataset_from_directory(
  'aclImdb/train',
  batch_size = batch_size,
  validation_split = 0.2,
  subset = 'training',
  seed = seed
)
validation_data <- text_dataset_from_directory(
  'aclImdb/train',
  batch_size = batch_size,
  validation_split = 0.2,
  subset = 'validation',
  seed = seed
)
test_data <- text_dataset_from_directory(
  'aclImdb/test',
  batch_size = batch_size
)
```

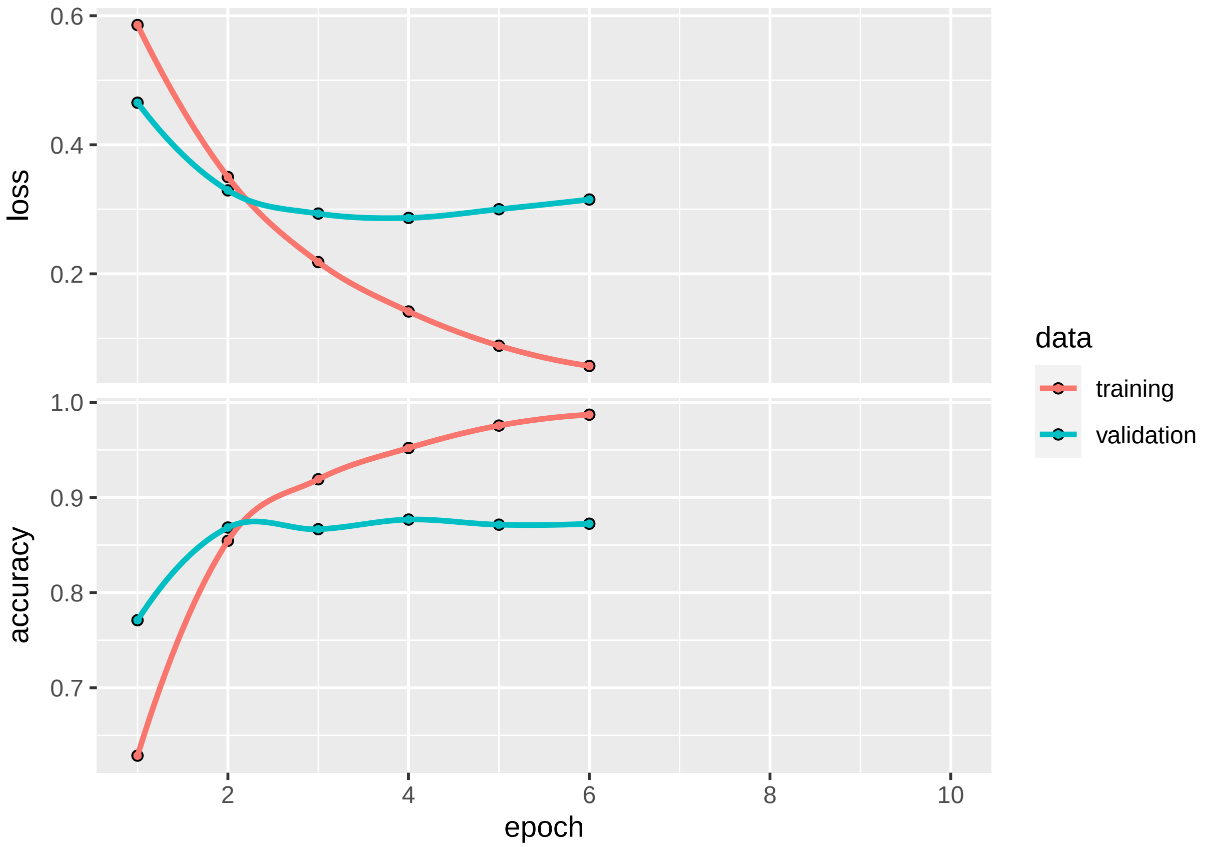
```
embedding <- 'https://tfhub.dev/google/nnlm-en-dim128/2'
hub_layer <- tfhub::layer_hub(handle = embedding, trainable = TRUE)
```

```
model <- keras_model_sequential() %>%
  hub_layer() %>%
  layer_dense(16, activation = 'relu') %>%
  layer_dense(1)

summary(model)
```

```
## Model: <no summary available, model was not built>
```

```
model %>% compile(
  optimizer = 'adam',
  loss = loss_binary_crossentropy(from_logits = TRUE),
  metrics = 'accuracy'
)


history <- model %>% fit(
  train_data,
  epochs = 10,
  validation_data = validation_data,
  callbacks = callback_early_stopping(patience = 2, monitor = 'val_loss'),
  verbose <- 1
)

plot(history)
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : Chernobyl! trL>n 6

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : Chernobyl! trL>n 6

## Warning in sqrt(sum.squares/one.delta): NaNs produced

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : Chernobyl! trL>n 6

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : Chernobyl! trL>n 6

## Warning in sqrt(sum.squares/one.delta): NaNs produced
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : Chernobyl! trL>n 6

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : Chernobyl! trL>n 6

## Warning in sqrt(sum.squares/one.delta): NaNs produced

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : Chernobyl! trL>n 6

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : Chernobyl! trL>n 6

## Warning in sqrt(sum.squares/one.delta): NaNs produced
```



```
results <- model %>% evaluate(test_data, verbose = 2)
results
```

```
##      loss  accuracy
## 0.3469216 0.8579600
```