Neural NLP

NLP master course 2021-2022

Mariano Rico (mariano.rico@upm.es)

Document created on 2022-01-20

Table of contents

1 Installing keras and Tensorflow in R

Keras is an ordinary package, with no special instructions to install. A simple `library(keras)` will do fine to use it. To install it use `install.packages("keras")`.

However, Tensorflow is a little bit special. Follow the instructions here to install TF in your platform.

2 Sentiment classification task: training a bidirectional LSTM on the IMDB reviews dataset

Original source here.

```
library(keras)

# Define maximum number of input features
max_features <- 20000

# Cut texts after this number of words
# (among top max_features most common words)
maxlen <- 100

batch_size <- 32

# Load imdb dataset
imdb <- dataset_imdb(num_words = max_features)

# Define training and test sets
x_train <- imdb$train$x
y_train <- imdb$train$y
x_test <- imdb$test$x
y_test <- imdb$test$y

# Output lengths of testing and training sets
cat(length(x_train), 'train sequences\n')
```

```
25000 train sequences
```

```
cat(length(x_test), 'test sequences\n')
```

```
25000 test sequences
```

```
cat('Pad sequences (samples x time)\n')
```

```
Pad sequences (samples x time)
```

```
# Pad training and test inputs
x_train <- pad_sequences(x_train, maxlen = maxlen)
x_test <- pad_sequences(x_test, maxlen = maxlen)

# Output dimensions of training and test inputs
cat('x_train shape:', dim(x_train), '\n')
```

```
x_train shape: 25000 100
```

```
cat('x_test shape:', dim(x_test), '\n')
```

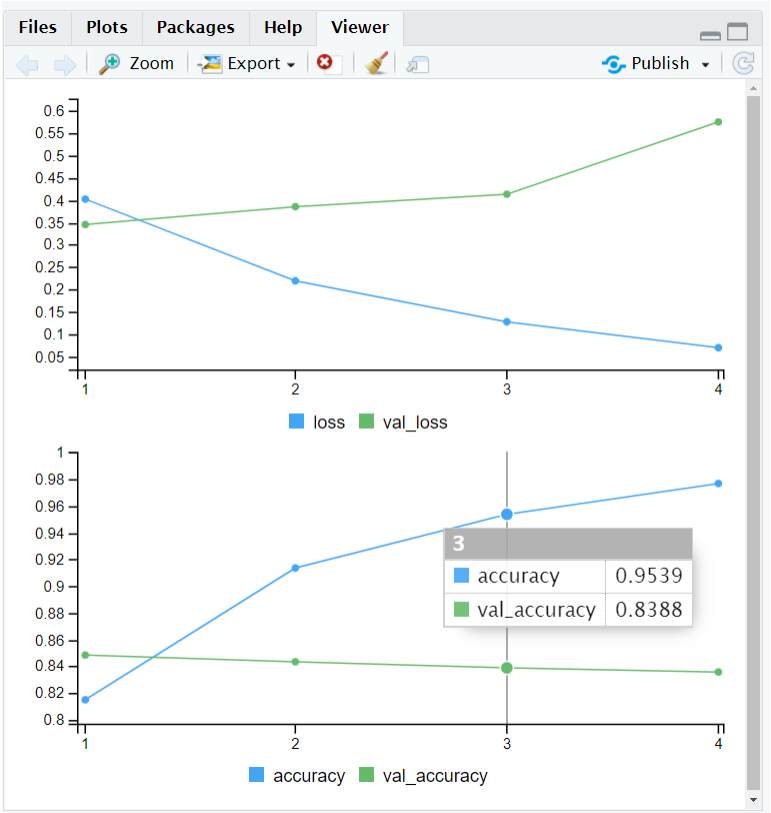
```
x_test shape: 25000 100
```

```
# Initialize model
model <- keras_model_sequential()
model %>%
  # Creates dense embedding layer; outputs 3D tensor
  # with shape (batch_size, sequence_length, output_dim)
  layer_embedding(input_dim = max_features,
                  output_dim = 128,
                  input_length = maxlen) %>%
  bidirectional(layer_lstm(units = 64)) %>%
  layer_dropout(rate = 0.5) %>%
  layer_dense(units = 1, activation = 'sigmoid')

# Try using different optimizers and different optimizer configs
model %>% compile(
  loss = 'binary_crossentropy',
  optimizer = 'adam',
  metrics = c('accuracy')
)

# Train model over four epochs
cat('Train...\n')
```

```
Train...
```

```
model %>% fit(
  x_train, y_train,
  batch_size = batch_size,
  epochs = 4,
  validation_data = list(x_test, y_test)
)
```

If you run this code in RStudio you will get the following interective graph in the Viewer tab.



3 Attention-based Neural Machine Translation with Keras

This is a simple move to a Kaggle notebook of the RStudio post published on July 2018. It is an example of using the R interface to Keras in order to create a NMT (Neural Machine translation) system from scratch.

The post describes step-by-step the implementation code. Here is the Kaggle notebook (notice that requires GPU and that it takes almost 4 hours of GPU time).

4 Transformers from Hugging Face

Original source here.

The Kaggle notebok is here Notice that this code requires a GPU, otherwise you will get and OOV (out of vocabulary) error.