Keras

Star 52,882

About Keras
Getting started
Developer guides
Keras API reference
Code examples
Computer Vision
Natural Language Processing
Structured Data
Timeseries
Audio Data
Generative Deep Learning
Reinforcement Learning
Graph Data
Quick Keras Recipes
Why choose Keras?
Community & governance
Contributing to Keras
KerasTuner

Search Keras documentation...

» Code examples / Natural Language Processing / Bidirectional LSTM on IMDB

Bidirectional LSTM on IMDB

Author: fchollet
Date created: 2020/05/03
Last modified: 2020/05/03
Description: Train a 2-layer bidirectional LSTM on the IMDB movie review sentiment classification dataset.

View in Colab • **GitHub source**

Setup

```
import numpy as np
from tensorflow import keras
from tensorflow.keras import layers

max_features = 20000  # Only consider the top 20k words
maxlen = 200  # Only consider the first 200 words of each movie review
```

Build the model

```
# Input for variable-length sequences of integers
inputs = keras.Input(shape=(None,), dtype="int32")
# Embed each integer in a 128-dimensional vector
x = layers.Embedding(max_features, 128)(inputs)
# Add 2 bidirectional LSTMs
x = layers.Bidirectional(layers.LSTM(64, return_sequences=True))(x)
x = layers.Bidirectional(layers.LSTM(64))(x)
# Add a classifier
outputs = layers.Dense(1, activation="sigmoid")(x)
model = keras.Model(inputs, outputs)
model.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_1 (InputLayer)         [(None, None)]            0         
_________________________________________________________________
embedding (Embedding)        (None, None, 128)         2560000   
_________________________________________________________________
bidirectional (Bidirectional (None, None, 128)         98816     
_________________________________________________________________
bidirectional_1 (Bidirection (None, 128)               98816     
_________________________________________________________________
dense (Dense)                (None, 1)                 129       
=================================================================
Total params: 2,757,761
Trainable params: 2,757,761
Non-trainable params: 0
_________________________________________________________________
```

Load the IMDB movie review sentiment data

```
(x_train, y_train), (x_val, y_val) = keras.datasets.imdb.load_data(
    num_words=max_features
)
print(len(x_train), "Training sequences")
print(len(x_val), "Validation sequences")
x_train = keras.preprocessing.sequence.pad_sequences(x_train, maxlen=maxlen)
x_val = keras.preprocessing.sequence.pad_sequences(x_val, maxlen=maxlen)
```

```
25000 Training sequences
25000 Validation sequences
```

Train and evaluate the model

```
model.compile("adam", "binary_crossentropy", metrics=["accuracy"])
model.fit(x_train, y_train, batch_size=32, epochs=2, validation_data=(x_val, y_val))
```

```
Epoch 1/2
782/782 [==============================] - 220s 281ms/step - loss: 0.4117 - accuracy: 0.8083
- val_loss: 0.6497 - val_accuracy: 0.6983
Epoch 2/2
726/782 [==========================>...] - ETA: 11s - loss: 0.3170 - accuracy: 0.8683
```