実践演習9-2

IMDBデータは映画のレビューに対して、P/N(肯定/否定)のラベルが付いた学習データです。学習用に25000事例、評価用に25000事例用意されていて、PNの割合はそれぞれ50%です。 各レビューは単語列ではなく、単語インデックスの系列として表現されています。

ここでは、頻度上位10000語を対象とし、データの大きさは先頭の50単語に限定します。

In [0]:

```
import tensorflow as tf
from tensorflow import keras
```

In [2]:

```
max_features = 10000
maxlen = 50
(X_train, y_train), (X_test, y_test) = keras.datasets.imdb.load_data(num_words=max_features)
X_train = keras.preprocessing.sequence.pad_sequences(X_train, maxlen=maxlen)
X_test = keras.preprocessing.sequence.pad_sequences(X_test, maxlen=maxlen)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/im
db.npz (https://storage.googleapis.com/tensorflow/tf-keras-datasets/imdb.npz)
17465344/17464789 [==============================] - 1s 0us/step
```

単語インデックスを単語に戻して、元のデータを確認します。インデックスは"padding", "start of sequence","unknown"にそれぞれ0,1,2が割り当てられているので、3つずらして対応させます。

In [3]:

```
word_index = keras.datasets.imdb.get_word_index()
reverse_word_index = dict([(value, key) for (key, value) in word_index.items()])
decoded_review = ' '.join([reverse_word_index.get(i - 3, '?') for i in X_train[0]])
decoded_review
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/im
db_word_index.json (https://storage.googleapis.com/tensorflow/tf-keras-datasets/imdb
_word_index.json)
1646592/1641221 [==============================] - 0s 0us/step
```

Out[3]:

```
"grown up are such a big profile for the whole film but these children are amazing a
nd should be praised for what they have done don't you think the whole story was so
lovely because it was true and was someone's life after all that was shared with us
all"
```

単純なRNNを構成して学習させます。

In [4]:

```
model = keras.Sequential([
    keras.layers.Embedding(max_features, 128),
    keras.layers.SimpleRNN(64),
    keras.layers.Dense(1, activation='sigmoid')
])
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
embedding (Embedding)        (None, None, 128)         1280000   
_________________________________________________________________
simple_rnn (SimpleRNN)       (None, 64)                12352     
_________________________________________________________________
dense (Dense)                (None, 1)                 65        
=================================================================
Total params: 1,292,417
Trainable params: 1,292,417
Non-trainable params: 0
_________________________________________________________________
```

In [5]:

```
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['acc'])
model.fit(X_train, y_train, epochs=5, batch_size=200,validation_split=0.2)
```

```
/usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/gradients_impl.py:112:
UserWarning: Converting sparse IndexedSlices to a dense Tensor of unknown shape. Thi
s may consume a large amount of memory.
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "

Train on 20000 samples, validate on 5000 samples
Epoch 1/5
20000/20000 [==============================] - 5s 232us/step - loss: 0.5202 - acc:
0.7340 - val_loss: 0.4436 - val_acc: 0.7994
Epoch 2/5
20000/20000 [==============================] - 4s 181us/step - loss: 0.2958 - acc:
0.8796 - val_loss: 0.4864 - val_acc: 0.7850
Epoch 3/5
20000/20000 [==============================] - 4s 181us/step - loss: 0.1255 - acc:
0.9585 - val_loss: 0.5739 - val_acc: 0.7730
Epoch 4/5
20000/20000 [==============================] - 4s 185us/step - loss: 0.0308 - acc:
0.9934 - val_loss: 0.7463 - val_acc: 0.7634
Epoch 5/5
20000/20000 [==============================] - 4s 180us/step - loss: 0.0065 - acc:
0.9995 - val_loss: 0.8327 - val_acc: 0.7730
```

Out[5]:

```
<tensorflow.python.keras.callbacks.History at 0x7f5378f495c0>
```

In [6]:

```
score = model.evaluate(X_test, y_test, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
Test loss: 0.7796998586845398
Test accuracy: 0.77572
```

RNNユニットをLSTMに変更して性能の変化を確認します。

In [7]:

```
model = keras.Sequential([
    keras.layers.Embedding(max_features, 128),
    keras.layers.LSTM(64),
    keras.layers.Dense(1, activation=tf.nn.sigmoid)
])
model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['acc'])
model.fit(X_train, y_train, epochs=5, batch_size=200,validation_split=0.2)
```

```
/usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/gradients_impl.py:112:
UserWarning: Converting sparse IndexedSlices to a dense Tensor of unknown shape. Thi
s may consume a large amount of memory.
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "

Train on 20000 samples, validate on 5000 samples
Epoch 1/5
20000/20000 [==============================] - 15s 745us/step - loss: 0.5037 - acc:
0.7482 - val_loss: 0.4238 - val_acc: 0.8050
Epoch 2/5
20000/20000 [==============================] - 14s 714us/step - loss: 0.3452 - acc:
0.8500 - val_loss: 0.4239 - val_acc: 0.8024
Epoch 3/5
20000/20000 [==============================] - 15s 728us/step - loss: 0.2997 - acc:
0.8757 - val_loss: 0.4600 - val_acc: 0.8034
Epoch 4/5
20000/20000 [==============================] - 15s 734us/step - loss: 0.2653 - acc:
0.8935 - val_loss: 0.4701 - val_acc: 0.8030
Epoch 5/5
20000/20000 [==============================] - 15s 730us/step - loss: 0.2353 - acc:
0.9081 - val_loss: 0.5012 - val_acc: 0.7924
```

Out[7]:

```
<tensorflow.python.keras.callbacks.History at 0x7f53787a45c0>
```

In [8]:

```
score = model.evaluate(X_test, y_test, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
Test loss: 0.49411070870399476
Test accuracy: 0.79984
```

双方向型にして、ユニットをGRUにします。また、ドロップアウトも加えてみます。

In [9]:

```
model = keras.Sequential([
    keras.layers.Embedding(max_features, 64),
    keras.layers.Bidirectional(keras.layers.GRU(32)),
    keras.layers.Dropout(0.5),
    keras.layers.Dense(1, activation=tf.nn.sigmoid)
])
model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['acc'])
model.fit(X_train, y_train, epochs=5, batch_size=200,validation_split=0.2)
```

```
/usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/gradients_impl.py:112:
UserWarning: Converting sparse IndexedSlices to a dense Tensor of unknown shape. Thi
s may consume a large amount of memory.
  "Converting sparse IndexedSlices to a dense Tensor of unknown shape. "


Train on 20000 samples, validate on 5000 samples
Epoch 1/5
20000/20000 [==============================] - 24s 1ms/step - loss: 0.5917 - acc: 0.
6724 - val_loss: 0.4531 - val_acc: 0.7824
Epoch 2/5
20000/20000 [==============================] - 23s 1ms/step - loss: 0.3716 - acc: 0.
8401 - val_loss: 0.4077 - val_acc: 0.8116
Epoch 3/5
20000/20000 [==============================] - 22s 1ms/step - loss: 0.3097 - acc: 0.
8701 - val_loss: 0.4608 - val_acc: 0.8050
Epoch 4/5
20000/20000 [==============================] - 21s 1ms/step - loss: 0.2695 - acc: 0.
8899 - val_loss: 0.4381 - val_acc: 0.8048
Epoch 5/5
20000/20000 [==============================] - 21s 1ms/step - loss: 0.2395 - acc: 0.
9055 - val_loss: 0.4686 - val_acc: 0.7960
```

Out[9]:

```
<tensorflow.python.keras.callbacks.History at 0x7f5371428f28>
```

In [10]:

```
score = model.evaluate(X_test, y_test, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
Test loss: 0.4524161660003662
Test accuracy: 0.80304
```

In [0]:

```

```