In [1]:

```
from keras.datasets import imdb
from keras import preprocessing
max_features = 10000
maxlen = 500
(x_train, y_train), (x_test, y_test) = imdb.load_data(
num_words=max_features)
x_train = preprocessing.sequence.pad_sequences(x_train, maxlen=maxlen)
x_test = preprocessing.sequence.pad_sequences(x_test, maxlen=maxlen)
```

```
Using TensorFlow backend.
```

In [4]:

```
from keras.layers import Embedding
embedding_layer = Embedding(1000, 64)

from keras.models import Sequential
from keras.layers import Flatten, Dense
model = Sequential()
model.add(Embedding(10000, 8, input_length=maxlen))
model.add(Flatten())
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['acc'
])
model.summary()
history = model.fit(x_train, y_train,
epochs=10,
batch_size=32,
validation_split=0.2)
```

```
WARNING:tensorflow:From C:\ProgramData\Anaconda3\lib\site-packages\keras\back
end\tensorflow_backend.py:541: The name tf.placeholder is deprecated. Please
use tf.compat.v1.placeholder instead.

WARNING:tensorflow:From C:\ProgramData\Anaconda3\lib\site-packages\keras\back
end\tensorflow_backend.py:4432: The name tf.random_uniform is deprecated. Ple
ase use tf.random.uniform instead.

WARNING:tensorflow:From C:\ProgramData\Anaconda3\lib\site-packages\keras\opti
mizers.py:793: The name tf.train.Optimizer is deprecated. Please use tf.compa
t.v1.train.Optimizer instead.

WARNING:tensorflow:From C:\ProgramData\Anaconda3\lib\site-packages\keras\back
end\tensorflow_backend.py:3657: The name tf.log is deprecated. Please use tf.
math.log instead.

WARNING:tensorflow:From C:\ProgramData\Anaconda3\lib\site-packages\tensorflow
\python\ops\nn_impl.py:180: add_dispatch_support.<locals>.wrapper (from tenso
rflow.python.ops.array_ops) is deprecated and will be removed in a future ver
sion.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
embedding_2 (Embedding)      (None, 500, 8)            80000     
_________________________________________________________________
flatten_1 (Flatten)          (None, 4000)              0         
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 4001      
=================================================================
Total params: 84,001
Trainable params: 84,001
Non-trainable params: 0
_________________________________________________________________
WARNING:tensorflow:From C:\ProgramData\Anaconda3\lib\site-packages\keras\back
end\tensorflow_backend.py:1033: The name tf.assign_add is deprecated. Please
use tf.compat.v1.assign_add instead.

Train on 20000 samples, validate on 5000 samples
Epoch 1/10
20000/20000 [==============================] - 3s 154us/step - loss: 0.5845 -
acc: 0.7007 - val_loss: 0.3946 - val_acc: 0.8506
Epoch 2/10
20000/20000 [==============================] - 3s 149us/step - loss: 0.3016 -
acc: 0.8850 - val_loss: 0.3023 - val_acc: 0.8714
Epoch 3/10
20000/20000 [==============================] - 3s 132us/step - loss: 0.2219 -
acc: 0.9163 - val_loss: 0.2685 - val_acc: 0.8916
Epoch 4/10
20000/20000 [==============================] - 3s 127us/step - loss: 0.1828 -
acc: 0.9306 - val_loss: 0.2808 - val_acc: 0.8830
Epoch 5/10
20000/20000 [==============================] - 3s 140us/step - loss: 0.1551 -
acc: 0.9442 - val_loss: 0.2711 - val_acc: 0.8932
Epoch 6/10
```

```
20000/20000 [==============================] - 3s 138us/step - loss: 0.1324 -
acc: 0.9535 - val_loss: 0.2821 - val_acc: 0.8896
Epoch 7/10
20000/20000 [==============================] - 3s 125us/step - loss: 0.1142 -
acc: 0.9601 - val_loss: 0.2908 - val_acc: 0.8930
Epoch 8/10
20000/20000 [==============================] - 3s 127us/step - loss: 0.0979 -
acc: 0.9673 - val_loss: 0.3032 - val_acc: 0.8906
Epoch 9/10
20000/20000 [==============================] - 3s 133us/step - loss: 0.0849 -
acc: 0.9724 - val_loss: 0.3158 - val_acc: 0.8926
Epoch 10/10
20000/20000 [==============================] - 3s 144us/step - loss: 0.0725 -
acc: 0.9768 - val_loss: 0.3373 - val_acc: 0.8876
```

In [5]:

```
from keras.layers import Embedding
embedding_layer = Embedding(10000, 64)

from keras.models import Sequential
from keras.layers import Flatten, Dense
model = Sequential()
model.add(Embedding(100000, 8, input_length=maxlen))
model.add(Flatten())
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['acc'
])
model.summary()
history = model.fit(x_train, y_train,
epochs=10,
batch_size=32,
validation_split=0.2)
```

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
embedding_4 (Embedding)      (None, 500, 8)            800000    
_________________________________________________________________
flatten_2 (Flatten)          (None, 4000)              0         
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 4001      
=================================================================
Total params: 804,001
Trainable params: 804,001
Non-trainable params: 0
_________________________________________________________________
Train on 20000 samples, validate on 5000 samples
Epoch 1/10
20000/20000 [==============================] - 13s 649us/step - loss: 0.5591
- acc: 0.7188 - val_loss: 0.3674 - val_acc: 0.8570
Epoch 2/10
20000/20000 [==============================] - 13s 640us/step - loss: 0.2865
- acc: 0.8885 - val_loss: 0.2851 - val_acc: 0.8826
Epoch 3/10
20000/20000 [==============================] - 15s 733us/step - loss: 0.2191
- acc: 0.9144 - val_loss: 0.2680 - val_acc: 0.8920
Epoch 4/10
20000/20000 [==============================] - 14s 695us/step - loss: 0.1850
- acc: 0.9295 - val_loss: 0.2697 - val_acc: 0.8912
Epoch 5/10
20000/20000 [==============================] - 14s 704us/step - loss: 0.1612
- acc: 0.9401 - val_loss: 0.2742 - val_acc: 0.8926
Epoch 6/10
20000/20000 [==============================] - 14s 699us/step - loss: 0.1418
- acc: 0.9498 - val_loss: 0.2821 - val_acc: 0.8922
Epoch 7/10
20000/20000 [==============================] - 14s 691us/step - loss: 0.1249
- acc: 0.9553 - val_loss: 0.2995 - val_acc: 0.8898
Epoch 8/10
20000/20000 [==============================] - 16s 779us/step - loss: 0.1102
- acc: 0.9616 - val_loss: 0.3232 - val_acc: 0.8826
Epoch 9/10
20000/20000 [==============================] - 16s 817us/step - loss: 0.0970
- acc: 0.9668 - val_loss: 0.3231 - val_acc: 0.8848
Epoch 10/10
20000/20000 [==============================] - 15s 761us/step - loss: 0.0857
- acc: 0.9705 - val_loss: 0.3357 - val_acc: 0.8864
```

In [8]:

```
max_features = 10000
maxlen = 100
(x_train, y_train), (x_test, y_test) = imdb.load_data(
num_words=max_features)
x_train = preprocessing.sequence.pad_sequences(x_train, maxlen=maxlen)
x_test = preprocessing.sequence.pad_sequences(x_test, maxlen=maxlen)
```

In [9]:

```
from keras.layers import Embedding
embedding_layer = Embedding(1000, 64)

from keras.models import Sequential
from keras.layers import Flatten, Dense
model = Sequential()
model.add(Embedding(10000, 8, input_length=maxlen))
model.add(Flatten())
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['acc'
])
model.summary()
history = model.fit(x_train, y_train,
epochs=10,
batch_size=32,
validation_split=0.2)
```

```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_8 (Embedding)      (None, 100, 8)            80000
_________________________________________________________________
flatten_4 (Flatten)          (None, 800)               0
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 801
=================================================================
Total params: 80,801
Trainable params: 80,801
Non-trainable params: 0
_________________________________________________________________
Train on 20000 samples, validate on 5000 samples
Epoch 1/10
20000/20000 [==============================] - 2s 112us/step - loss: 0.6135 -
acc: 0.6897 - val_loss: 0.4618 - val_acc: 0.8060
Epoch 2/10
20000/20000 [==============================] - 2s 84us/step - loss: 0.3662 -
acc: 0.8529 - val_loss: 0.3530 - val_acc: 0.8426
Epoch 3/10
20000/20000 [==============================] - 2s 86us/step - loss: 0.2861 -
acc: 0.8830 - val_loss: 0.3263 - val_acc: 0.8580
Epoch 4/10
20000/20000 [==============================] - 2s 93us/step - loss: 0.2481 -
acc: 0.9012 - val_loss: 0.3210 - val_acc: 0.8588
Epoch 5/10
20000/20000 [==============================] - 2s 113us/step - loss: 0.2223 -
acc: 0.9126 - val_loss: 0.3249 - val_acc: 0.8598
Epoch 6/10
20000/20000 [==============================] - 2s 115us/step - loss: 0.2012 -
acc: 0.9215 - val_loss: 0.3291 - val_acc: 0.8582
Epoch 7/10
20000/20000 [==============================] - 2s 94us/step - loss: 0.1824 -
acc: 0.9295 - val_loss: 0.3350 - val_acc: 0.8588
Epoch 8/10
20000/20000 [==============================] - 2s 93us/step - loss: 0.1657 -
acc: 0.9371 - val_loss: 0.3421 - val_acc: 0.8596
Epoch 9/10
20000/20000 [==============================] - 2s 92us/step - loss: 0.1494 -
acc: 0.9444 - val_loss: 0.3546 - val_acc: 0.8564
Epoch 10/10
20000/20000 [==============================] - 2s 106us/step - loss: 0.1334 -
acc: 0.9510 - val_loss: 0.3616 - val_acc: 0.8560
```

In [ ]: