gender_classfication_CNN_1

July 29, 2019

[1]:
```
import re
import os
import math
import nltk
import pickle
import random
import numpy as np
import tensorflow as tf
from nltk.corpus import wordnet
from tensorflow.contrib import rnn
from html.parser import HTMLParser
import xml.etree.ElementTree as ET
from nltk.stem import WordNetLemmatizer
from keras.layers.recurrent import LSTM
from keras.preprocessing import sequence
from keras.layers.embeddings import Embedding
from keras.models import Sequential,load_model
from keras.preprocessing.text import Tokenizer
from keras.layers.convolutional import Conv1D, MaxPooling1D
from keras.layers import Flatten, Dense, Dropout, BatchNormalization
```

```
C:\ProgramData\Anaconda3\lib\site-packages\h5py\__init__.py:72: UserWarning:
h5py is running against HDF5 1.10.2 when it was built against 1.10.3, this may
cause problems
  '{0}.{1}.{2}'.format(*version.hdf5_built_version_tuple)
Using TensorFlow backend.
```

[2]:
```
#using pretrained glove embeddings to embed words
def get_embeddings_index():
    embeddings_index = {}
    f = open(os.path.join('./glove.6B', 'glove.6B.300d.txt')) #TODO try 300␣
 ↪dimensions
    for line in f:
        values = line.split()
        word = values[0]
        coefs = np.asarray(values[1:], dtype='float32')
```

```
        embeddings_index[word] = coefs
    f.close()
    return embeddings_index
embeddings_index = get_embeddings_index()
```

[3]:
```
lemmatizer = WordNetLemmatizer()
def preprocess_words(word_list):
    processed_word_list = []
    for word in word_list.split():
        if wordnet.synsets(word):
            processed_word_list.append(lemmatizer.lemmatize(word))
    return processed_word_list
```

[4]:
```
class CustomHTMLParser(HTMLParser):
    a = ''
    def handle_data(self, data):
        self.a = self.a + str(data)
    def get_raw_text(self):
        self.a = re.sub(r'[0-9_]+', ' ', self.a)
        self.a = re.sub(r'[^\w\s]', ' ', self.a)
        return self.a
```

[5]:
```
#Preprocessing xml files to x train and y train data
def preprocess_data(path = './en/'):
    x_data = []
    y_data = []
    for filename in os.listdir(path):
        root = ET.parse(path + filename).getroot()
        #TODO add for other classifications, ie. age_group and multi-class
        if(root.attrib['gender'] == 'male'):
            y = 1
        elif(root.attrib['gender'] == 'female'):
            y = 0

        for text in root.findall('conversations/conversation'):
            parser = CustomHTMLParser()
            parser.feed(str(text.text))
            removed_tags = parser.get_raw_text()
            word_list = preprocess_words(removed_tags)
            x_data.append(word_list)
            y_data.append(y)
    return x_data, y_data
```

[ ]:
```
x_data, y_data = preprocess_data()
```

[ ]:
```
#store pre-processed input
def save_preprocessed_data(x_data, y_data):
    with open("x_data_all_cnn.txt", "wb") as f:
        pickle.dump(x_data, f)
```

```
    with open("y_data_all_cnn.txt", "wb") as f:
        pickle.dump(y_data, f)
save_preprocessed_data(x_data, y_data)
```

[ ]:
```
def prepare_word_index(x_data):
    tokenizer = Tokenizer(num_words=50000) #max features is 50000
    tokenizer.fit_on_texts(x_data)
    word_index = tokenizer.word_index
    print('Found %s unique tokens.' % len(word_index))
    return word_index
word_index = prepare_word_index(x_data)

def save_word_index(word_index):
    with open("word_index.txt", "wb") as f:
        pickle.dump(word_index, f)
save_word_index(word_index)
```

[12]:
```
max_text_length = 500
embedding_dim = 300 #change to other dim as well
total = 163371
def load_word_index():
    with open("word_index.txt", "rb") as f:
        word_index = pickle.load(f)
        return word_index
word_index = load_word_index()
```

[7]:
```
def get_network_input(x_data, word_index, y_data, max_features=len(word_index)):
    x = []
    for text in x_data:
        text_ids = []
        for word in text:
            word_id = word_index.get(word, -1)
            if word_id != -1 and word_id < max_features:
                text_ids.append(word_id)
        x.append(text_ids)

    #pad sequence length to max_text_length
    x = sequence.pad_sequences(x, maxlen=max_text_length, padding='post')
    y = tf.keras.utils.to_categorical(y_data, num_classes=2)
    return x, y
```

[8]:
```
#load entire batch as training data
def load_all_data(fx="x_data_all_cnn.txt", fy="y_data_all_cnn.txt", shuffle =␣
 ↪False, seed=1000):
    with open(fx, "rb") as f:
        x_data = pickle.load(f)
    with open(fy, "rb") as f:
        y_data = pickle.load(f)
    x, y = get_network_input(x_data, load_word_index(), y_data, 10000)
```

```
    if(shuffle):
        np.random.seed(seed)
        r = np.arange(len(x))
        np.random.shuffle(r)
        x = np.asarray(x)
        y = np.asarray(y)
        x = x[r]
        y = y[r]
    return x,y
```

[9]:
```
#building embedding matrix
def get_embedding_layer(embedding_dim):
    word_index = load_word_index()
    embedding_matrix = np.zeros((len(word_index) + 1, embedding_dim))
    for word, i in word_index.items():
        embedding_vector = embeddings_index.get(word)
        if embedding_vector is not None:
            # words not found in embedding index will be all-zeros.
            embedding_matrix[i] = embedding_vector

    #embedding layer
    embedding_layer = Embedding(len(word_index) + 1, embedding_dim,␣
 ↪weights=[embedding_matrix],
                                input_length=max_text_length, trainable=False)
    return embedding_layer
```

[10]:
```
x, y = load_all_data(fx="x_data_all_cnn.txt", fy="y_data_all_cnn.txt", shuffle␣
 ↪= True)
#train on 140000 data samples out of 163371 samples, validation on 140000 to␣
 ↪150000
train_num = 150000
validation_num = 160000
x_train = x[:train_num]
y_train = y[:train_num]
x_val = x[train_num:validation_num]
y_val = y[train_num:validation_num]
```

[13]:
```
def create_cnn_model():
    #pure cnn model
    model = Sequential()
    model.add(get_embedding_layer(embedding_dim))
    model.add(Conv1D(filters=128, kernel_size=5, padding='same',␣
 ↪activation='relu'))
    model.add(MaxPooling1D(pool_size=5))
    model.add(Dropout(0.2))
    model.add(Conv1D(filters=128, kernel_size=5, padding='same',␣
 ↪activation='relu'))
    model.add(MaxPooling1D(pool_size=5))
```

```
    model.add(Dropout(0.2))
    model.add(Conv1D(filters=128, kernel_size=5, padding='same',␣
 ↪activation='relu'))
    model.add(MaxPooling1D(pool_size=5))
    model.add(Dropout(0.2))
    model.add(Flatten())
    model.add(BatchNormalization())
    model.add(Dropout(0.2))
    model.add(Dense(128, activation='relu'))
    model.add(BatchNormalization())
    model.add(Dense(2, activation='softmax'))
    model.compile(loss='binary_crossentropy', optimizer='adam',␣
 ↪metrics=['accuracy'])#try rmsprop
    print(model.summary())
    return model
model12 = create_cnn_model()
model12.fit(x_train, y_train, epochs=25, batch_size=512,␣
 ↪validation_data=(x_val, y_val))
model12.save('model12_v1.h5')
```

```
WARNING:tensorflow:From C:\ProgramData\Anaconda3\lib\site-
packages\tensorflow\python\framework\op_def_library.py:263: colocate_with (from
tensorflow.python.framework.ops) is deprecated and will be removed in a future
version.
Instructions for updating:
Colocations handled automatically by placer.
WARNING:tensorflow:From C:\ProgramData\Anaconda3\lib\site-
packages\keras\backend\tensorflow_backend.py:3445: calling dropout (from
tensorflow.python.ops.nn_ops) with keep_prob is deprecated and will be removed
in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 -
keep_prob`.
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 500, 300)          23156700
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 500, 128)          192128
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 100, 128)          0
_________________________________________________________________
dropout_1 (Dropout)          (None, 100, 128)          0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 100, 128)          82048
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 20, 128)           0
```

```
_________________________________________________________________
dropout_2 (Dropout)          (None, 20, 128)           0
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 20, 128)           82048
_________________________________________________________________
max_pooling1d_3 (MaxPooling1 (None, 4, 128)            0
_________________________________________________________________
dropout_3 (Dropout)          (None, 4, 128)            0
_________________________________________________________________
flatten_1 (Flatten)          (None, 512)               0
_________________________________________________________________
batch_normalization_1 (Batch (None, 512)               2048
_________________________________________________________________
dropout_4 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               65664
_________________________________________________________________
batch_normalization_2 (Batch (None, 128)               512
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 258
=================================================================
Total params: 23,581,406
Trainable params: 423,426
Non-trainable params: 23,157,980
_________________________________________________________________
None
WARNING:tensorflow:From C:\ProgramData\Anaconda3\lib\site-
packages\tensorflow\python\ops\math_ops.py:3066: to_int32 (from
tensorflow.python.ops.math_ops) is deprecated and will be removed in a future
version.
Instructions for updating:
Use tf.cast instead.
Train on 150000 samples, validate on 10000 samples
Epoch 1/25
150000/150000 [==============================] - 61s 405us/step - loss: 0.7227 -
acc: 0.5359 - val_loss: 0.6888 - val_acc: 0.5160
Epoch 2/25
150000/150000 [==============================] - 61s 406us/step - loss: 0.6887 -
acc: 0.5463 - val_loss: 0.7069 - val_acc: 0.5038
Epoch 3/25
150000/150000 [==============================] - 60s 402us/step - loss: 0.6833 -
acc: 0.5557 - val_loss: 0.6874 - val_acc: 0.5307
Epoch 4/25
150000/150000 [==============================] - 57s 379us/step - loss: 0.6756 -
acc: 0.5679 - val_loss: 0.7003 - val_acc: 0.5271
Epoch 5/25
150000/150000 [==============================] - 57s 378us/step - loss: 0.6671 -
acc: 0.5797 - val_loss: 0.6691 - val_acc: 0.5754
```

```
Epoch 6/25
150000/150000 [==============================] - 57s 379us/step - loss: 0.6564 -
acc: 0.5943 - val_loss: 0.7196 - val_acc: 0.5326
Epoch 7/25
150000/150000 [==============================] - 57s 381us/step - loss: 0.6449 -
acc: 0.6094 - val_loss: 0.6743 - val_acc: 0.5576
Epoch 8/25
150000/150000 [==============================] - 57s 379us/step - loss: 0.6295 -
acc: 0.6261 - val_loss: 0.6693 - val_acc: 0.5851
Epoch 9/25
150000/150000 [==============================] - 57s 378us/step - loss: 0.6112 -
acc: 0.6475 - val_loss: 0.7354 - val_acc: 0.5447
Epoch 10/25
150000/150000 [==============================] - 57s 378us/step - loss: 0.5922 -
acc: 0.6674 - val_loss: 0.6852 - val_acc: 0.5751
Epoch 11/25
150000/150000 [==============================] - 57s 379us/step - loss: 0.5703 -
acc: 0.6882 - val_loss: 0.7032 - val_acc: 0.5871
Epoch 12/25
150000/150000 [==============================] - 57s 379us/step - loss: 0.5477 -
acc: 0.7085 - val_loss: 0.7141 - val_acc: 0.5814
Epoch 13/25
150000/150000 [==============================] - 57s 380us/step - loss: 0.5257 -
acc: 0.7265 - val_loss: 0.7139 - val_acc: 0.5870
Epoch 14/25
150000/150000 [==============================] - 57s 381us/step - loss: 0.5070 -
acc: 0.7420 - val_loss: 0.8462 - val_acc: 0.5498
Epoch 15/25
150000/150000 [==============================] - 57s 380us/step - loss: 0.4876 -
acc: 0.7557 - val_loss: 0.8401 - val_acc: 0.5466
Epoch 16/25
150000/150000 [==============================] - 58s 387us/step - loss: 0.4710 -
acc: 0.7675 - val_loss: 0.9645 - val_acc: 0.5384
Epoch 17/25
150000/150000 [==============================] - 57s 380us/step - loss: 0.4537 -
acc: 0.7792 - val_loss: 0.8541 - val_acc: 0.5464
Epoch 18/25
150000/150000 [==============================] - 57s 378us/step - loss: 0.4377 -
acc: 0.7897 - val_loss: 0.8528 - val_acc: 0.5609
Epoch 19/25
150000/150000 [==============================] - 57s 379us/step - loss: 0.4244 -
acc: 0.7992 - val_loss: 0.8354 - val_acc: 0.5621
Epoch 20/25
150000/150000 [==============================] - 57s 378us/step - loss: 0.4092 -
acc: 0.8069 - val_loss: 1.0705 - val_acc: 0.5382
Epoch 21/25
150000/150000 [==============================] - 57s 378us/step - loss: 0.4001 -
acc: 0.8131 - val_loss: 0.8576 - val_acc: 0.5753
```

```
Epoch 22/25
150000/150000 [==============================] - 57s 377us/step - loss: 0.3871 -
acc: 0.8206 - val_loss: 0.9325 - val_acc: 0.5651
Epoch 23/25
150000/150000 [==============================] - 57s 378us/step - loss: 0.3772 -
acc: 0.8256 - val_loss: 1.1440 - val_acc: 0.5340
Epoch 24/25
150000/150000 [==============================] - 58s 389us/step - loss: 0.3671 -
acc: 0.8324 - val_loss: 1.1923 - val_acc: 0.5362
Epoch 25/25
150000/150000 [==============================] - 53s 357us/step - loss: 0.3587 -
acc: 0.8389 - val_loss: 1.0524 - val_acc: 0.5351
```

[15]:
```
model12.fit(x_train, y_train, epochs=50, initial_epoch=25, batch_size=512,␣
 ↪validation_data=(x_val, y_val))
model12.save('./models/model12_v2.h5')
```

```
Train on 150000 samples, validate on 10000 samples
Epoch 26/50
150000/150000 [==============================] - 54s 359us/step - loss: 0.3478 -
acc: 0.8423 - val_loss: 1.1588 - val_acc: 0.5370
Epoch 27/50
150000/150000 [==============================] - 54s 359us/step - loss: 0.3413 -
acc: 0.8473 - val_loss: 1.0349 - val_acc: 0.5525
Epoch 28/50
150000/150000 [==============================] - 53s 357us/step - loss: 0.3331 -
acc: 0.8517 - val_loss: 0.9330 - val_acc: 0.5703
Epoch 29/50
150000/150000 [==============================] - 54s 357us/step - loss: 0.3266 -
acc: 0.8546 - val_loss: 0.8923 - val_acc: 0.5811
Epoch 30/50
150000/150000 [==============================] - 54s 360us/step - loss: 0.3214 -
acc: 0.8584 - val_loss: 0.9575 - val_acc: 0.5675
Epoch 31/50
150000/150000 [==============================] - 55s 365us/step - loss: 0.3169 -
acc: 0.8603 - val_loss: 0.9780 - val_acc: 0.5645
Epoch 32/50
150000/150000 [==============================] - 55s 364us/step - loss: 0.3077 -
acc: 0.8655 - val_loss: 1.1863 - val_acc: 0.5509
Epoch 33/50
150000/150000 [==============================] - 55s 365us/step - loss: 0.3069 -
acc: 0.8649 - val_loss: 1.0629 - val_acc: 0.5591
Epoch 34/50
150000/150000 [==============================] - 54s 362us/step - loss: 0.2996 -
acc: 0.8694 - val_loss: 1.3185 - val_acc: 0.5349
Epoch 35/50
150000/150000 [==============================] - 55s 365us/step - loss: 0.2925 -
```

```
acc: 0.8726 - val_loss: 1.0998 - val_acc: 0.5509
Epoch 36/50
150000/150000 [==============================] - 55s 364us/step - loss: 0.2896 -
acc: 0.8737 - val_loss: 1.3639 - val_acc: 0.5371
Epoch 37/50
150000/150000 [==============================] - 54s 363us/step - loss: 0.2846 -
acc: 0.8766 - val_loss: 1.0344 - val_acc: 0.5614
Epoch 38/50
150000/150000 [==============================] - 55s 365us/step - loss: 0.2811 -
acc: 0.8776 - val_loss: 1.0777 - val_acc: 0.5540
Epoch 39/50
150000/150000 [==============================] - 55s 364us/step - loss: 0.2767 -
acc: 0.8820 - val_loss: 1.3418 - val_acc: 0.5344
Epoch 40/50
150000/150000 [==============================] - 54s 363us/step - loss: 0.2729 -
acc: 0.8820 - val_loss: 1.1225 - val_acc: 0.5536
Epoch 41/50
150000/150000 [==============================] - 55s 364us/step - loss: 0.2711 -
acc: 0.8834 - val_loss: 1.1711 - val_acc: 0.5524
Epoch 42/50
150000/150000 [==============================] - 54s 363us/step - loss: 0.2661 -
acc: 0.8859 - val_loss: 1.0405 - val_acc: 0.5708
Epoch 43/50
150000/150000 [==============================] - 55s 366us/step - loss: 0.2611 -
acc: 0.8881 - val_loss: 1.1038 - val_acc: 0.5606
Epoch 44/50
150000/150000 [==============================] - 56s 372us/step - loss: 0.2599 -
acc: 0.8894 - val_loss: 1.1650 - val_acc: 0.5559
Epoch 45/50
150000/150000 [==============================] - 55s 369us/step - loss: 0.2586 -
acc: 0.8894 - val_loss: 1.1429 - val_acc: 0.5618
Epoch 46/50
150000/150000 [==============================] - 55s 365us/step - loss: 0.2539 -
acc: 0.8928 - val_loss: 1.2126 - val_acc: 0.5563
Epoch 47/50
150000/150000 [==============================] - 55s 365us/step - loss: 0.2530 -
acc: 0.8923 - val_loss: 1.1762 - val_acc: 0.5473
Epoch 48/50
150000/150000 [==============================] - 56s 370us/step - loss: 0.2502 -
acc: 0.8936 - val_loss: 1.1892 - val_acc: 0.5598
Epoch 49/50
150000/150000 [==============================] - 82s 546us/step - loss: 0.2478 -
acc: 0.8953 - val_loss: 1.1378 - val_acc: 0.5631
Epoch 50/50
150000/150000 [==============================] - 92s 615us/step - loss: 0.2420 -
acc: 0.8980 - val_loss: 1.4277 - val_acc: 0.5360
```

[16]:
```
model12 = load_model('./models/model12_v2.h5')
model12.evaluate(x_val, y_val, verbose=1)
```

```
10000/10000 [==============================] - 3s 324us/step
```

[16]:
```
[1.4276667419433593, 0.536]
```

[14]:
```
model12 = load_model('model12_v1.h5')
model12.evaluate(x_val, y_val, verbose=1)
```

```
10000/10000 [==============================] - 3s 295us/step
```

[14]:
```
[1.0523515672683716, 0.5351]
```

[17]:
```
model12.fit(x_train, y_train, epochs=75, initial_epoch=50, batch_size=512,␣
↪validation_data=(x_val, y_val))
model12.save('./models/model12_v3.h5')
```

```
Train on 150000 samples, validate on 10000 samples
Epoch 51/75
150000/150000 [==============================] - 56s 374us/step - loss: 0.2428 -
acc: 0.8979 - val_loss: 1.1948 - val_acc: 0.5539
Epoch 52/75
150000/150000 [==============================] - 55s 365us/step - loss: 0.2369 -
acc: 0.9003 - val_loss: 1.2008 - val_acc: 0.5575
Epoch 53/75
150000/150000 [==============================] - 54s 361us/step - loss: 0.2374 -
acc: 0.8996 - val_loss: 1.3477 - val_acc: 0.5420
Epoch 54/75
150000/150000 [==============================] - 57s 382us/step - loss: 0.2374 -
acc: 0.8999 - val_loss: 1.2039 - val_acc: 0.5558
Epoch 55/75
150000/150000 [==============================] - 55s 365us/step - loss: 0.2332 -
acc: 0.9025 - val_loss: 1.2457 - val_acc: 0.5560
Epoch 56/75
150000/150000 [==============================] - 55s 365us/step - loss: 0.2307 -
acc: 0.9028 - val_loss: 1.3552 - val_acc: 0.5453
Epoch 57/75
150000/150000 [==============================] - 55s 365us/step - loss: 0.2304 -
acc: 0.9033 - val_loss: 1.1214 - val_acc: 0.5659
Epoch 58/75
150000/150000 [==============================] - 55s 367us/step - loss: 0.2282 -
acc: 0.9043 - val_loss: 1.4507 - val_acc: 0.5464
Epoch 59/75
150000/150000 [==============================] - 55s 366us/step - loss: 0.2271 -
acc: 0.9051 - val_loss: 1.2581 - val_acc: 0.5535
Epoch 60/75
```

```
150000/150000 [==============================] - 56s 371us/step - loss: 0.2250 -
acc: 0.9058 - val_loss: 1.3838 - val_acc: 0.5480
Epoch 61/75
150000/150000 [==============================] - 63s 421us/step - loss: 0.2242 -
acc: 0.9060 - val_loss: 1.4442 - val_acc: 0.5463
Epoch 62/75
150000/150000 [==============================] - 92s 612us/step - loss: 0.2232 -
acc: 0.9063 - val_loss: 1.1889 - val_acc: 0.5611
Epoch 63/75
150000/150000 [==============================] - 92s 615us/step - loss: 0.2184 -
acc: 0.9090 - val_loss: 1.2345 - val_acc: 0.5608
Epoch 64/75
150000/150000 [==============================] - 92s 615us/step - loss: 0.2187 -
acc: 0.9090 - val_loss: 1.2225 - val_acc: 0.5610
Epoch 65/75
150000/150000 [==============================] - 92s 617us/step - loss: 0.2153 -
acc: 0.9105 - val_loss: 1.2740 - val_acc: 0.5550
Epoch 66/75
150000/150000 [==============================] - 92s 615us/step - loss: 0.2148 -
acc: 0.9111 - val_loss: 1.2163 - val_acc: 0.5644
Epoch 67/75
150000/150000 [==============================] - 92s 614us/step - loss: 0.2144 -
acc: 0.9111 - val_loss: 1.3450 - val_acc: 0.5544
Epoch 68/75
150000/150000 [==============================] - 92s 616us/step - loss: 0.2111 -
acc: 0.9121 - val_loss: 1.4500 - val_acc: 0.5475
Epoch 69/75
150000/150000 [==============================] - 93s 617us/step - loss: 0.2099 -
acc: 0.9125 - val_loss: 1.4242 - val_acc: 0.5454
Epoch 70/75
150000/150000 [==============================] - 93s 618us/step - loss: 0.2106 -
acc: 0.9121 - val_loss: 1.5493 - val_acc: 0.5407
Epoch 71/75
150000/150000 [==============================] - 92s 615us/step - loss: 0.2077 -
acc: 0.9145 - val_loss: 1.2992 - val_acc: 0.5599
Epoch 72/75
150000/150000 [==============================] - 92s 615us/step - loss: 0.2057 -
acc: 0.9151 - val_loss: 1.3499 - val_acc: 0.5512
Epoch 73/75
150000/150000 [==============================] - 92s 611us/step - loss: 0.2046 -
acc: 0.9153 - val_loss: 1.1791 - val_acc: 0.5717
Epoch 74/75
150000/150000 [==============================] - 92s 616us/step - loss: 0.2068 -
acc: 0.9144 - val_loss: 1.5558 - val_acc: 0.5415
Epoch 75/75
150000/150000 [==============================] - 92s 614us/step - loss: 0.2042 -
acc: 0.9153 - val_loss: 1.2200 - val_acc: 0.5647
```

[18]:
```
model12 = load_model('./models/model12_v3.h5')
model12.evaluate(x_val, y_val, verbose=1)
```

```
10000/10000 [==============================] - 3s 333us/step
```

[18]:
```
[1.2199890647888183, 0.5647]
```

[19]:
```
def create_cnn_model_1():
    #pure cnn model
    model = Sequential()
    model.add(get_embedding_layer(embedding_dim))
    model.add(Conv1D(filters=128, kernel_size=5, padding='same',␣
 ↪activation='relu'))
    model.add(MaxPooling1D(pool_size=5))
    model.add(Dropout(0.3))
    model.add(Conv1D(filters=128, kernel_size=5, padding='same',␣
 ↪activation='relu'))
    model.add(MaxPooling1D(pool_size=5))
    model.add(Dropout(0.3))
    model.add(Conv1D(filters=128, kernel_size=5, padding='same',␣
 ↪activation='relu'))
    model.add(MaxPooling1D(pool_size=5))
    model.add(Dropout(0.3))
    model.add(Flatten())
    model.add(BatchNormalization())
    model.add(Dropout(0.3))
    model.add(Dense(128, activation='relu'))
    model.add(BatchNormalization())
    model.add(Dense(2, activation='softmax'))
    model.compile(loss='binary_crossentropy', optimizer='adam',␣
 ↪metrics=['accuracy'])#try rmsprop
    print(model.summary())
    return model
model13 = create_cnn_model_1()
model13.fit(x_train, y_train, epochs=25, batch_size=512,␣
 ↪validation_data=(x_val, y_val))
model13.save('model13_v1.h5')
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, 500, 300)          23156700
_________________________________________________________________
conv1d_4 (Conv1D)            (None, 500, 128)          192128
_________________________________________________________________
max_pooling1d_4 (MaxPooling1 (None, 100, 128)          0
_________________________________________________________________
```

```
dropout_5 (Dropout)          (None, 100, 128)          0
_________________________________________________________________
conv1d_5 (Conv1D)            (None, 100, 128)          82048
_________________________________________________________________
max_pooling1d_5 (MaxPooling1 (None, 20, 128)           0
_________________________________________________________________
dropout_6 (Dropout)          (None, 20, 128)           0
_________________________________________________________________
conv1d_6 (Conv1D)            (None, 20, 128)           82048
_________________________________________________________________
max_pooling1d_6 (MaxPooling1 (None, 4, 128)            0
_________________________________________________________________
dropout_7 (Dropout)          (None, 4, 128)            0
_________________________________________________________________
flatten_2 (Flatten)          (None, 512)               0
_________________________________________________________________
batch_normalization_3 (Batch (None, 512)               2048
_________________________________________________________________
dropout_8 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_3 (Dense)              (None, 128)               65664
_________________________________________________________________
batch_normalization_4 (Batch (None, 128)               512
_________________________________________________________________
dense_4 (Dense)              (None, 2)                 258
=================================================================
Total params: 23,581,406
Trainable params: 423,426
Non-trainable params: 23,157,980
_________________________________________________________________
None
Train on 150000 samples, validate on 10000 samples
Epoch 1/25
150000/150000 [==============================] - 56s 373us/step - loss: 0.7233 -
acc: 0.5328 - val_loss: 0.6823 - val_acc: 0.5484
Epoch 2/25
150000/150000 [==============================] - 62s 411us/step - loss: 0.6902 -
acc: 0.5416 - val_loss: 0.6863 - val_acc: 0.5265
Epoch 3/25
150000/150000 [==============================] - 57s 378us/step - loss: 0.6857 -
acc: 0.5493 - val_loss: 0.6950 - val_acc: 0.5248
Epoch 4/25
150000/150000 [==============================] - 56s 373us/step - loss: 0.6806 -
acc: 0.5619 - val_loss: 0.6788 - val_acc: 0.5497
Epoch 5/25
150000/150000 [==============================] - 56s 372us/step - loss: 0.6737 -
acc: 0.5715 - val_loss: 0.6869 - val_acc: 0.5363
Epoch 6/25
```

```
150000/150000 [==============================] - 56s 371us/step - loss: 0.6676 -
acc: 0.5810 - val_loss: 0.6774 - val_acc: 0.5461
Epoch 7/25
150000/150000 [==============================] - 56s 372us/step - loss: 0.6594 -
acc: 0.5917 - val_loss: 0.6677 - val_acc: 0.5715
Epoch 8/25
150000/150000 [==============================] - 56s 372us/step - loss: 0.6508 -
acc: 0.6053 - val_loss: 0.6770 - val_acc: 0.5569
Epoch 9/25
150000/150000 [==============================] - 56s 372us/step - loss: 0.6404 -
acc: 0.6181 - val_loss: 0.6641 - val_acc: 0.5768
Epoch 10/25
150000/150000 [==============================] - 56s 372us/step - loss: 0.6276 -
acc: 0.6337 - val_loss: 0.6620 - val_acc: 0.5915
Epoch 11/25
150000/150000 [==============================] - 56s 374us/step - loss: 0.6133 -
acc: 0.6506 - val_loss: 0.6681 - val_acc: 0.5880
Epoch 12/25
150000/150000 [==============================] - 56s 374us/step - loss: 0.5987 -
acc: 0.6653 - val_loss: 0.6680 - val_acc: 0.5877
Epoch 13/25
150000/150000 [==============================] - 56s 375us/step - loss: 0.5832 -
acc: 0.6802 - val_loss: 0.6767 - val_acc: 0.5929
Epoch 14/25
150000/150000 [==============================] - 57s 380us/step - loss: 0.5686 -
acc: 0.6944 - val_loss: 0.7071 - val_acc: 0.5806
Epoch 15/25
150000/150000 [==============================] - 57s 378us/step - loss: 0.5525 -
acc: 0.7072 - val_loss: 0.7051 - val_acc: 0.5810
Epoch 16/25
150000/150000 [==============================] - 58s 383us/step - loss: 0.5375 -
acc: 0.7203 - val_loss: 0.8325 - val_acc: 0.5421
Epoch 17/25
150000/150000 [==============================] - 57s 382us/step - loss: 0.5260 -
acc: 0.7314 - val_loss: 0.7118 - val_acc: 0.5857
Epoch 18/25
150000/150000 [==============================] - 57s 379us/step - loss: 0.5093 -
acc: 0.7433 - val_loss: 0.7685 - val_acc: 0.5660
Epoch 19/25
150000/150000 [==============================] - 57s 379us/step - loss: 0.4974 -
acc: 0.7521 - val_loss: 0.7549 - val_acc: 0.5795
Epoch 20/25
150000/150000 [==============================] - 57s 377us/step - loss: 0.4853 -
acc: 0.7606 - val_loss: 0.7404 - val_acc: 0.5810
Epoch 21/25
150000/150000 [==============================] - 57s 377us/step - loss: 0.4747 -
acc: 0.7678 - val_loss: 0.7573 - val_acc: 0.5825
Epoch 22/25
```

```
150000/150000 [==============================] - 57s 379us/step - loss: 0.4640 -
acc: 0.7745 - val_loss: 0.7917 - val_acc: 0.5747
Epoch 23/25
150000/150000 [==============================] - 57s 380us/step - loss: 0.4540 -
acc: 0.7810 - val_loss: 0.7574 - val_acc: 0.5865
Epoch 24/25
150000/150000 [==============================] - 57s 379us/step - loss: 0.4447 -
acc: 0.7870 - val_loss: 0.7667 - val_acc: 0.5895
Epoch 25/25
150000/150000 [==============================] - 57s 379us/step - loss: 0.4358 -
acc: 0.7922 - val_loss: 0.7690 - val_acc: 0.5910
```

[20]:
```
model13 = load_model('model13_v1.h5')
model13.evaluate(x_val, y_val, verbose=1)
```

```
10000/10000 [==============================] - 3s 334us/step
```

[20]:
```
[0.7689918890953064, 0.591]
```

[21]:
```
model13.fit(x_train, y_train, epochs=50, initial_epoch=25, batch_size=512,␣
↪validation_data=(x_val, y_val))
model13.save('./models/model13_v2.h5')
```

```
Train on 150000 samples, validate on 10000 samples
Epoch 26/50
150000/150000 [==============================] - 56s 374us/step - loss: 0.4279 -
acc: 0.7989 - val_loss: 0.8079 - val_acc: 0.5712
Epoch 27/50
150000/150000 [==============================] - 56s 376us/step - loss: 0.4182 -
acc: 0.8040 - val_loss: 0.7865 - val_acc: 0.5817
Epoch 28/50
150000/150000 [==============================] - 57s 380us/step - loss: 0.4113 -
acc: 0.8089 - val_loss: 0.7827 - val_acc: 0.5809
Epoch 29/50
150000/150000 [==============================] - 54s 363us/step - loss: 0.4068 -
acc: 0.8106 - val_loss: 0.7880 - val_acc: 0.5816
Epoch 30/50
150000/150000 [==============================] - 55s 366us/step - loss: 0.4014 -
acc: 0.8151 - val_loss: 0.7815 - val_acc: 0.5870
Epoch 31/50
150000/150000 [==============================] - 53s 354us/step - loss: 0.3940 -
acc: 0.8183 - val_loss: 0.8074 - val_acc: 0.5871
Epoch 32/50
150000/150000 [==============================] - 53s 354us/step - loss: 0.3891 -
acc: 0.8221 - val_loss: 0.8466 - val_acc: 0.5750
Epoch 33/50
150000/150000 [==============================] - 53s 355us/step - loss: 0.3830 -
```

```
acc: 0.8257 - val_loss: 0.8169 - val_acc: 0.5860
Epoch 34/50
150000/150000 [==============================] - 53s 355us/step - loss: 0.3757 -
acc: 0.8298 - val_loss: 0.8739 - val_acc: 0.5721
Epoch 35/50
150000/150000 [==============================] - 53s 355us/step - loss: 0.3725 -
acc: 0.8315 - val_loss: 0.8875 - val_acc: 0.5670
Epoch 36/50
150000/150000 [==============================] - 53s 356us/step - loss: 0.3687 -
acc: 0.8343 - val_loss: 0.8543 - val_acc: 0.5743
Epoch 37/50
150000/150000 [==============================] - 53s 355us/step - loss: 0.3633 -
acc: 0.8355 - val_loss: 0.8350 - val_acc: 0.5900
Epoch 38/50
150000/150000 [==============================] - 53s 356us/step - loss: 0.3606 -
acc: 0.8380 - val_loss: 0.9856 - val_acc: 0.5600
Epoch 39/50
150000/150000 [==============================] - 53s 357us/step - loss: 0.3546 -
acc: 0.8422 - val_loss: 0.8359 - val_acc: 0.5852
Epoch 40/50
150000/150000 [==============================] - 54s 357us/step - loss: 0.3503 -
acc: 0.8439 - val_loss: 0.8467 - val_acc: 0.5881
Epoch 41/50
150000/150000 [==============================] - 54s 357us/step - loss: 0.3469 -
acc: 0.8452 - val_loss: 0.8587 - val_acc: 0.5823
Epoch 42/50
150000/150000 [==============================] - 54s 357us/step - loss: 0.3458 -
acc: 0.8461 - val_loss: 0.8673 - val_acc: 0.5836
Epoch 43/50
150000/150000 [==============================] - 54s 357us/step - loss: 0.3406 -
acc: 0.8490 - val_loss: 0.8671 - val_acc: 0.5930
Epoch 44/50
150000/150000 [==============================] - 54s 359us/step - loss: 0.3371 -
acc: 0.8505 - val_loss: 0.9828 - val_acc: 0.5576
Epoch 45/50
150000/150000 [==============================] - 53s 357us/step - loss: 0.3328 -
acc: 0.8536 - val_loss: 0.8867 - val_acc: 0.5827
Epoch 46/50
150000/150000 [==============================] - 54s 357us/step - loss: 0.3306 -
acc: 0.8538 - val_loss: 0.9367 - val_acc: 0.5763
Epoch 47/50
150000/150000 [==============================] - 54s 358us/step - loss: 0.3310 -
acc: 0.8539 - val_loss: 0.8883 - val_acc: 0.5839
Epoch 48/50
150000/150000 [==============================] - 54s 357us/step - loss: 0.3256 -
acc: 0.8558 - val_loss: 0.9072 - val_acc: 0.5779
Epoch 49/50
150000/150000 [==============================] - 54s 357us/step - loss: 0.3228 -
```

```
acc: 0.8584 - val_loss: 0.9020 - val_acc: 0.5843
Epoch 50/50
150000/150000 [==============================] - 61s 405us/step - loss: 0.3180 -
acc: 0.8607 - val_loss: 0.9006 - val_acc: 0.5818
```

[22]:
```
model13 = load_model('./models/model13_v2.h5')
model13.evaluate(x_val, y_val, verbose=1)
```

```
10000/10000 [==============================] - 3s 326us/step
```

[22]:
```
[0.9006036964416504, 0.5818]
```

[23]:
```
def create_cnn_model_2():
    #pure cnn model
    model = Sequential()
    model.add(get_embedding_layer(embedding_dim))
    model.add(Conv1D(filters=128, kernel_size=5, padding='same',␣
 ↪activation='relu'))
    model.add(MaxPooling1D(pool_size=5))
    model.add(Dropout(0.5))
    model.add(Conv1D(filters=128, kernel_size=5, padding='same',␣
 ↪activation='relu'))
    model.add(MaxPooling1D(pool_size=5))
    model.add(Dropout(0.5))
    model.add(Conv1D(filters=128, kernel_size=5, padding='same',␣
 ↪activation='relu'))
    model.add(MaxPooling1D(pool_size=5))
    model.add(Dropout(0.5))
    model.add(Flatten())
    model.add(BatchNormalization())
    model.add(Dropout(0.5))
    model.add(Dense(128, activation='relu'))
    model.add(BatchNormalization())
    model.add(Dense(2, activation='softmax'))
    model.compile(loss='binary_crossentropy', optimizer='adam',␣
 ↪metrics=['accuracy'])#try rmsprop
    print(model.summary())
    return model
model14 = create_cnn_model()
model14.fit(x_train, y_train, epochs=25, batch_size=512,␣
 ↪validation_data=(x_val, y_val))
model14.save('model14_v1.h5')
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_3 (Embedding)      (None, 500, 300)          23156700
```

```
_________________________________________________________________
conv1d_7 (Conv1D)            (None, 500, 128)          192128
_________________________________________________________________
max_pooling1d_7 (MaxPooling1 (None, 100, 128)          0
_________________________________________________________________
dropout_9 (Dropout)          (None, 100, 128)          0
_________________________________________________________________
conv1d_8 (Conv1D)            (None, 100, 128)          82048
_________________________________________________________________
max_pooling1d_8 (MaxPooling1 (None, 20, 128)           0
_________________________________________________________________
dropout_10 (Dropout)         (None, 20, 128)           0
_________________________________________________________________
conv1d_9 (Conv1D)            (None, 20, 128)           82048
_________________________________________________________________
max_pooling1d_9 (MaxPooling1 (None, 4, 128)            0
_________________________________________________________________
dropout_11 (Dropout)         (None, 4, 128)            0
_________________________________________________________________
flatten_3 (Flatten)          (None, 512)               0
_________________________________________________________________
batch_normalization_5 (Batch (None, 512)               2048
_________________________________________________________________
dropout_12 (Dropout)         (None, 512)               0
_________________________________________________________________
dense_5 (Dense)              (None, 128)               65664
_________________________________________________________________
batch_normalization_6 (Batch (None, 128)               512
_________________________________________________________________
dense_6 (Dense)              (None, 2)                 258
=================================================================
Total params: 23,581,406
Trainable params: 423,426
Non-trainable params: 23,157,980
_________________________________________________________________
None
Train on 150000 samples, validate on 10000 samples
Epoch 1/25
150000/150000 [==============================] - 65s 437us/step - loss: 0.7136 -
acc: 0.5357 - val_loss: 0.6891 - val_acc: 0.5224
Epoch 2/25
150000/150000 [==============================] - 63s 418us/step - loss: 0.6861 -
acc: 0.5494 - val_loss: 0.6859 - val_acc: 0.5442
Epoch 3/25
150000/150000 [==============================] - 61s 409us/step - loss: 0.6792 -
acc: 0.5645 - val_loss: 0.6805 - val_acc: 0.5490
Epoch 4/25
150000/150000 [==============================] - 212s 1ms/step - loss: 0.6698 -
```

```
acc: 0.5768 - val_loss: 0.6675 - val_acc: 0.5788
Epoch 5/25
150000/150000 [==============================] - 67s 445us/step - loss: 0.6607 -
acc: 0.5892 - val_loss: 0.6796 - val_acc: 0.5589
Epoch 6/25
150000/150000 [==============================] - 65s 431us/step - loss: 0.6503 -
acc: 0.6037 - val_loss: 0.7077 - val_acc: 0.5406
Epoch 7/25
150000/150000 [==============================] - 61s 409us/step - loss: 0.6358 -
acc: 0.6215 - val_loss: 0.6722 - val_acc: 0.5861
Epoch 8/25
150000/150000 [==============================] - 61s 408us/step - loss: 0.6176 -
acc: 0.6436 - val_loss: 0.6783 - val_acc: 0.5683
Epoch 9/25
150000/150000 [==============================] - 62s 411us/step - loss: 0.5967 -
acc: 0.6649 - val_loss: 0.6728 - val_acc: 0.5785
Epoch 10/25
150000/150000 [==============================] - 62s 410us/step - loss: 0.5750 -
acc: 0.6860 - val_loss: 0.6944 - val_acc: 0.5803
Epoch 11/25
150000/150000 [==============================] - 62s 411us/step - loss: 0.5528 -
acc: 0.7056 - val_loss: 0.7417 - val_acc: 0.5669
Epoch 12/25
150000/150000 [==============================] - 62s 412us/step - loss: 0.5306 -
acc: 0.7235 - val_loss: 0.7261 - val_acc: 0.5832
Epoch 13/25
150000/150000 [==============================] - 62s 412us/step - loss: 0.5090 -
acc: 0.7402 - val_loss: 0.7766 - val_acc: 0.5664
Epoch 14/25
150000/150000 [==============================] - 62s 413us/step - loss: 0.4900 -
acc: 0.7544 - val_loss: 0.7927 - val_acc: 0.5781
Epoch 15/25
150000/150000 [==============================] - 62s 416us/step - loss: 0.4729 -
acc: 0.7673 - val_loss: 0.7938 - val_acc: 0.5635
Epoch 16/25
150000/150000 [==============================] - 63s 418us/step - loss: 0.4571 -
acc: 0.7775 - val_loss: 1.0106 - val_acc: 0.5417
Epoch 17/25
150000/150000 [==============================] - 63s 420us/step - loss: 0.4404 -
acc: 0.7871 - val_loss: 0.8610 - val_acc: 0.5616
Epoch 18/25
150000/150000 [==============================] - 65s 434us/step - loss: 0.4260 -
acc: 0.7966 - val_loss: 0.9699 - val_acc: 0.5522
Epoch 19/25
150000/150000 [==============================] - 67s 446us/step - loss: 0.4115 -
acc: 0.8063 - val_loss: 0.9493 - val_acc: 0.5458
Epoch 20/25
150000/150000 [==============================] - 80s 534us/step - loss: 0.4011 -
```

```
acc: 0.8123 - val_loss: 0.8363 - val_acc: 0.5828
Epoch 21/25
150000/150000 [==============================] - 96s 640us/step - loss: 0.3902 -
acc: 0.8194 - val_loss: 0.8979 - val_acc: 0.5622
Epoch 22/25
150000/150000 [==============================] - 96s 638us/step - loss: 0.3790 -
acc: 0.8259 - val_loss: 1.0164 - val_acc: 0.5468
Epoch 23/25
150000/150000 [==============================] - 96s 639us/step - loss: 0.3697 -
acc: 0.8304 - val_loss: 0.9734 - val_acc: 0.5505
Epoch 24/25
150000/150000 [==============================] - 94s 627us/step - loss: 0.3607 -
acc: 0.8356 - val_loss: 1.1400 - val_acc: 0.5381
Epoch 25/25
150000/150000 [==============================] - 94s 628us/step - loss: 0.3506 -
acc: 0.8416 - val_loss: 0.8950 - val_acc: 0.5737
```

[24]:
```
model14 = load_model('model14_v1.h5')
model14.evaluate(x_val, y_val, verbose=1)
```

```
10000/10000 [==============================] - 4s 361us/step
```

[24]:
```
[0.8950166385650635, 0.5737]
```

[25]:
```
model14.fit(x_train, y_train, epochs=50, initial_epoch=25, batch_size=512,␣
↪validation_data=(x_val, y_val))
model14.save('./models/model14_v2.h5')
```

```
Train on 150000 samples, validate on 10000 samples
Epoch 26/50
150000/150000 [==============================] - 95s 635us/step - loss: 0.3428 -
acc: 0.8465 - val_loss: 1.0043 - val_acc: 0.5561
Epoch 27/50
150000/150000 [==============================] - 95s 631us/step - loss: 0.3375 -
acc: 0.8486 - val_loss: 1.1272 - val_acc: 0.5496
Epoch 28/50
150000/150000 [==============================] - 93s 618us/step - loss: 0.3316 -
acc: 0.8532 - val_loss: 0.9154 - val_acc: 0.5734
Epoch 29/50
150000/150000 [==============================] - 94s 627us/step - loss: 0.3216 -
acc: 0.8580 - val_loss: 0.9897 - val_acc: 0.5594
Epoch 30/50
150000/150000 [==============================] - 94s 628us/step - loss: 0.3179 -
acc: 0.8599 - val_loss: 1.0805 - val_acc: 0.5502
Epoch 31/50
150000/150000 [==============================] - 95s 636us/step - loss: 0.3126 -
acc: 0.8622 - val_loss: 1.3086 - val_acc: 0.5307
```

```
Epoch 32/50
150000/150000 [==============================] - 94s 629us/step - loss: 0.3071 -
acc: 0.8651 - val_loss: 1.0463 - val_acc: 0.5607
Epoch 33/50
150000/150000 [==============================] - 95s 636us/step - loss: 0.3020 -
acc: 0.8674 - val_loss: 1.0102 - val_acc: 0.5624
Epoch 34/50
150000/150000 [==============================] - 96s 638us/step - loss: 0.2968 -
acc: 0.8710 - val_loss: 1.2716 - val_acc: 0.5349
Epoch 35/50
150000/150000 [==============================] - 93s 619us/step - loss: 0.2902 -
acc: 0.8749 - val_loss: 1.1630 - val_acc: 0.5501
Epoch 36/50
150000/150000 [==============================] - 96s 641us/step - loss: 0.2874 -
acc: 0.8754 - val_loss: 1.0707 - val_acc: 0.5630
Epoch 37/50
150000/150000 [==============================] - 96s 641us/step - loss: 0.2827 -
acc: 0.8783 - val_loss: 1.1375 - val_acc: 0.5511
Epoch 38/50
150000/150000 [==============================] - 96s 641us/step - loss: 0.2794 -
acc: 0.8797 - val_loss: 0.9941 - val_acc: 0.5795
Epoch 39/50
150000/150000 [==============================] - 96s 641us/step - loss: 0.2754 -
acc: 0.8811 - val_loss: 1.1029 - val_acc: 0.5577
Epoch 40/50
150000/150000 [==============================] - 96s 643us/step - loss: 0.2714 -
acc: 0.8836 - val_loss: 1.1654 - val_acc: 0.5510
Epoch 41/50
150000/150000 [==============================] - 97s 644us/step - loss: 0.2683 -
acc: 0.8860 - val_loss: 1.1722 - val_acc: 0.5468
Epoch 42/50
150000/150000 [==============================] - 96s 641us/step - loss: 0.2648 -
acc: 0.8865 - val_loss: 1.2254 - val_acc: 0.5440
Epoch 43/50
150000/150000 [==============================] - 99s 660us/step - loss: 0.2646 -
acc: 0.8874 - val_loss: 1.1579 - val_acc: 0.5589
Epoch 44/50
150000/150000 [==============================] - 97s 645us/step - loss: 0.2592 -
acc: 0.8899 - val_loss: 1.1551 - val_acc: 0.5593
Epoch 45/50
150000/150000 [==============================] - 96s 642us/step - loss: 0.2582 -
acc: 0.8903 - val_loss: 1.5894 - val_acc: 0.5333
Epoch 46/50
150000/150000 [==============================] - 97s 644us/step - loss: 0.2533 -
acc: 0.8924 - val_loss: 1.2611 - val_acc: 0.5459
Epoch 47/50
150000/150000 [==============================] - 97s 644us/step - loss: 0.2520 -
acc: 0.8926 - val_loss: 1.1668 - val_acc: 0.5547
```

```
Epoch 48/50
150000/150000 [==============================] - 96s 639us/step - loss: 0.2495 -
acc: 0.8941 - val_loss: 1.1591 - val_acc: 0.5591
Epoch 49/50
150000/150000 [==============================] - 94s 627us/step - loss: 0.2465 -
acc: 0.8961 - val_loss: 1.2692 - val_acc: 0.5466
Epoch 50/50
150000/150000 [==============================] - 96s 638us/step - loss: 0.2431 -
acc: 0.8971 - val_loss: 1.1512 - val_acc: 0.5609
```

[26]:
```
model14 = load_model('./models/model14_v2.h5')
model14.evaluate(x_val, y_val, verbose=1)
```

```
10000/10000 [==============================] - 3s 304us/step
```

[26]:
```
[1.1512007453918458, 0.5609]
```

[ ]:
```

```