

In [5]:

```
from google.colab import drive
drive.mount('/content/gdrive')
```

Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdqgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%b&scope=email%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdocs.test%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive.photos.readonly%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fpeopleapi.readonly&response_type=code

Enter your authorization code:

.....

Mounted at /content/gdrive

In [0]:

In [16]:

```
!ls "/content/gdrive/My Drive/quora/data/train.csv"
```

```
1.Quora.ipynb          nlp_feat_linear.csv
2.Quora_Preprocessing.ipynb  nlp_features_train.csv
3.Q_Mean_W2V.ipynb      sparse_lr.pckl
4.ML_models.ipynb       sparse_lr_y.pckl
APPLIED_AI_QUORASUBMISSION.ipynb  train.csv
df_fe_without_preprocessing_train.csv  train.db
distance                train_n.txt
Distance-0.1.3.tar.gz   train_p.txt
final_features.csv      wordcloud-1.3.3-cp36-cp36m-win_amd64.whl
```

In [0]:

```
path = "/content/gdrive/My Drive/quora/data/train.csv"
save_path = "/content/gdrive/My Drive/quora/data/"
```

In [0]:

```
import pandas as pd
```

In [0]:

```
data = pd.read_csv(path)
```

In [127]:

```
data.head(2)
```

Out[127]:

	id	qid1	qid2	question1	question2	is_duplicate
0	0	1	2	What is the step by step guide to invest in sh...	What is the step by step guide to invest in sh...	0
1	1	3	4	What is the story of Kohinoor (Koh-i-Noor) Dia...	What would happen if the Indian government sto...	0

In [128]:

```
data.tail(4)
```

Out[128]:

	id	qid1	qid2	question1	question2	is_duplicate
404286	404286	18840	155606	Do you believe there is life after death?	Is it true that there is life after death?	1
404287	404287	537928	537929	What is one coin?	What's this coin?	0
404288	404288	537930	537931	What is the approx annual cost of living while...	I am having little hairfall problem but I want...	0
404289	404289	537932	537933	What is like to have sex with cousin?	What is it like to have sex with your cousin?	0

In [21]:

```
len(data)
```

Out[21]:

404290

In [9]:

```
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
```

Using TensorFlow backend.

In [0]:

```
tokenizer = Tokenizer(num_words=None, filters='!"#$%&()*+,-./:;<=>?@[\\]^_`{|}~ ', lower=True, split=' ', char_level=False, oov_token='unk')
```

In [0]:

```
#slight aside
```

In [0]:

```
texts = data['question1'].astype('str').tolist()
```

In [0]:

```
import numpy as np
texts[0]
labels = np.asarray(data['is_duplicate'])
```

In [0]:

```
labels = labels.tolist()
```

In [18]:

```
labels[0:3]
```

Out[18]:

[0, 0, 0]

In [0]:

```
textsq2 = data['question2'].astype('str').tolist()

equalizedq1 = []
equalizedq2 = []
new_labels = []
for que1, que2, labels in zip(texts, textsq2, labels):
```

```
if len(que1.split()) <= 30 and len(que2.split()) <= 30:
    equalizedq1.append(que1)
    equalizedq2.append(que2)
    new_labels.append(labels)
```

In [23]:

```
len(equalizedq1)
```

Out[23]:

396450

In [0]:

```
tokenizer.fit_on_texts(texts)
```

In [33]:

```
len(new_labels)
```

Out[33]:

396450

In [0]:

```
tokenizer.fit_on_texts(equalizedq1)
sequencesq1 = tokenizer.texts_to_sequences(equalizedq1)
word_indexq1 = tokenizer.word_index

tokenizer.fit_on_texts(equalizedq2)
sequencesq2 = tokenizer.texts_to_sequences(equalizedq2)
word_indexq2 = tokenizer.word_index
```

In [0]:

```
question2 = pad_sequences(sequencesq2, maxlen=30)
```

In [0]:

```
question1 = pad_sequences(sequencesq1, maxlen = 30)
```

In [48]:

```
len(question2)
```

Out[48]:

396450

In [50]:

```
question2.shape
```

Out[50]:

(396450, 30)

In [0]:

```
#let's make a simple network for these
```

```
X_validq1 = question1[:80000,:]
X_validq2 = question2[:80000,:]
question1 = question1[80000:,:]
question2 = question2[80000:,:]
```

```
question2 = question2[00000:,:]

```

```
y_valid = new_labels[:80000]
labels = new_labels[80000:]

```

In [52]:

```
len(y_valid),len(X_validq1),len(question1),len(labels)

```

Out[52]:

```
(80000, 80000, 316450, 316450)

```

In [0]:

```
n_lstm_1 = 100
drop_lstm = 0.3
drop_embed = 0.3

vocab1 = len(word_indexq1)
vocab2 = len(word_indexq2)

qu1_input = Input(shape = (30,))
#drop1 = SpatialDropout1D(drop_embed)(qu1_input)
#drop1 = SpatialDropout1D(drop_embed)(qu1_input)
emb1 = Embedding(vocab1,64)(qu1_input)

qu2_input = Input(shape = (30,))

#drop2 = SpatialDropout1D(drop_embed)(qu2_input)
emb2 = Embedding(vocab2,64)(qu2_input)

core = Bidirectional(LSTM(64, dropout=drop_lstm))

# out_core = Bidirectional(LSTM(n_lstm_1, dropout=drop_lstm))

out1 = core(emb1)

# out_further_1 = out_core(out1)
out2 = core(emb2)

merged = concatenate([out1,out2])
predictions = Dense(1,activation = 'sigmoid')(merged)

model = Model([qu1_input, qu2_input], predictions)

```

In [56]:

```
len(question1),len(question2)

```

Out[56]:

```
(316450, 316450)

```

In [55]:

```
len(labels)

```

Out[55]:

```
316450

```

In [58]:

```
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
history = model.fit([question1,question2],labels,epochs = 12,batch_size = 512,validation_data = ([X_validq1,X_validq2],y_valid))

```

```

Train on 316450 samples, validate on 80000 samples
Epoch 1/12
316450/316450 [=====] - 199s 630us/step - loss: 0.4321 - acc: 0.7967 - va
l_loss: 0.4990 - val_acc: 0.7634
Epoch 2/12
316450/316450 [=====] - 195s 617us/step - loss: 0.3886 - acc: 0.8178 - va
l_loss: 0.5162 - val_acc: 0.7644
Epoch 3/12
316450/316450 [=====] - 195s 618us/step - loss: 0.3455 - acc: 0.8376 - va
l_loss: 0.5488 - val_acc: 0.7613
Epoch 4/12
316450/316450 [=====] - 197s 623us/step - loss: 0.3074 - acc: 0.8571 - va
l_loss: 0.5815 - val_acc: 0.7617
Epoch 5/12
316450/316450 [=====] - 196s 619us/step - loss: 0.2762 - acc: 0.8722 - va
l_loss: 0.6318 - val_acc: 0.7585
Epoch 6/12
316450/316450 [=====] - 196s 620us/step - loss: 0.2498 - acc: 0.8854 - va
l_loss: 0.6909 - val_acc: 0.7589
Epoch 7/12
316450/316450 [=====] - 197s 622us/step - loss: 0.2295 - acc: 0.8953 - va
l_loss: 0.7290 - val_acc: 0.7574
Epoch 8/12
316450/316450 [=====] - 196s 621us/step - loss: 0.2113 - acc: 0.9035 - va
l_loss: 0.7939 - val_acc: 0.7591
Epoch 9/12
316450/316450 [=====] - 196s 621us/step - loss: 0.1953 - acc: 0.9118 - va
l_loss: 0.8121 - val_acc: 0.7557
Epoch 10/12
316450/316450 [=====] - 197s 623us/step - loss: 0.1818 - acc: 0.9185 - va
l_loss: 0.8895 - val_acc: 0.7551
Epoch 11/12
316450/316450 [=====] - 196s 621us/step - loss: 0.1706 - acc: 0.9237 - va
l_loss: 0.9595 - val_acc: 0.7523
Epoch 12/12
316450/316450 [=====] - 195s 617us/step - loss: 0.1595 - acc: 0.9287 - va
l_loss: 1.0038 - val_acc: 0.7550

```

In [0]:

```

sequences = tokenizer.texts_to_sequences(texts)
word_index = tokenizer.word_index

```

In [83]:

```

pd.DataFrame(np.array([len(seq) for seq in sequences]).reshape((len(sequences),1))).describe() #
pad for 20

```

Out[83]:

	0
count	404290.000000
mean	11.033793
std	5.499495
min	0.000000
25%	7.000000
50%	10.000000
75%	13.000000
max	127.000000

In [0]:

```

question1 = pad_sequences(sequences, maxlen=20)

labels = np.asarray(data['is_duplicate'])

```

In [0]:

```
tokenizer2 = Tokenizer(num_words=None, filters='!"#$%&()*+,-./:;<=>?@[\\]^_`{|}~ ', lower=True, split=' ', char_level=False, oov_token='unk')
texts2 = data['question2'].astype('str').tolist()
tokenizer2.fit_on_texts(texts2)

sequences2 = tokenizer2.texts_to_sequences(texts2)
pd.DataFrame(np.array([len(seq) for seq in sequences2]).reshape((len(sequences2),1))).describe() #
pad for 20
word_index2 = tokenizer2.word_index
```

In [86]:

```
pd.DataFrame(np.array([len(seq) for seq in sequences2]).reshape((len(sequences2),1))).describe()
```

Out[86]:

	0
count	404290.000000
mean	11.278891
std	6.377210
min	0.000000
25%	7.000000
50%	10.000000
75%	13.000000
max	237.000000

In [0]:

```
question2 = pad_sequences(sequences2, maxlen=20)
```

In [0]:

```
vocab2 = len(word_index2)
```

In [0]:

```
vocab1 = len(word_index)
```

In [0]:

```
#creating the architecture

from keras import Model

from keras import Input
from keras.datasets import imdb
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Dense, Dropout, Embedding, SpatialDropout1D, LSTM, concatenate
from keras.layers.wrappers import Bidirectional
from keras.callbacks import ModelCheckpoint
import os
from sklearn.metrics import roc_auc_score
import matplotlib.pyplot as plt
%matplotlib inline

def plt_dynamic(x, vy, ty, ax, colors=['b']):
    ax.plot(x, vv, 'b', label="Validation Loss")
```

```
ax.plot(x, ty, 'r', label="Train Loss")
plt.legend()
plt.grid()
fig.canvas.draw()
```

In [0]:

```
n_lstm_1 = 100
drop_lstm = 0.3
drop_embed = 0.3

qu1_input = Input(shape = (20,))
#drop1 = SpatialDropout1D(drop_embed) (qu1_input)
#drop1 = SpatialDropout1D(drop_embed) (qu1_input)
emb1 = Embedding(vocab1,64) (qu1_input)

qu2_input = Input(shape = (20,))

#drop2 = SpatialDropout1D(drop_embed) (qu2_input)
emb2 = Embedding(vocab2,64) (qu2_input)

core = Bidirectional(LSTM(64, dropout=drop_lstm ))

# out_core = Bidirectional(LSTM(n_lstm_1, dropout=drop_lstm))

out1 = core(emb1)

# out_further_1 = out_core(out1)
out2 = core(emb2)

merged = concatenate([out1,out2])
predictions = Dense(1,activation = 'sigmoid') (merged)

model = Model([qu1_input, qu2_input], predictions)
```

In [119]:

```
model.summary()
```

Layer (type)	Output Shape	Param #	Connected to
input_18 (InputLayer)	(None, 20)	0	
input_19 (InputLayer)	(None, 20)	0	
embedding_15 (Embedding)	(None, 20, 64)	4732736	input_18[0][0]
embedding_16 (Embedding)	(None, 20, 64)	4361792	input_19[0][0]
bidirectional_12 (Bidirectional)	(None, 128)	66048	embedding_15[0][0] embedding_16[0][0]
concatenate_2 (Concatenate)	(None, 256)	0	bidirectional_12[0][0] bidirectional_12[1][0]
dense_2 (Dense)	(None, 1)	257	concatenate_2[0][0]
Total params: 9,160,833			
Trainable params: 9,160,833			
Non-trainable params: 0			

In [0]:

```
from sklearn.model_selection import train_test_split
```

In [129]:

404290

In [130]:

```
0.3*len(question1)
```

Out [130]:

121287.0

In [0]:

```
X_validq1 = question1[:100000,:]
X_validq2 = question2[:100000,:]
question1 = question1[100000:]
question2 = question2[100000:]
```

```
y_valid = labels[:100000]
labels = labels[100000:]
```

In [132]:

```
labels[:10]
```

Out [132] :

[illegible]


```
1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1,
1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1,
0, 1, 0, 0, 0, 0, 0, 0, 1, 0])
```

In [0]:

```
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

In [135]:

```
history = model.fit([question1,question2],labels,epochs = 40,batch_size = 512,validation_data = ([X_validq1,X_validq2],y_valid))
```

Train on 304290 samples, validate on 100000 samples

Epoch 1/40

304290/304290 [=====] - 172s 564us/step - loss: 0.3642 - acc: 0.8310 - val_loss: 0.3707 - val_acc: 0.8275

Epoch 2/40

304290/304290 [=====] - 176s 578us/step - loss: 0.3200 - acc: 0.8525 - val_loss: 0.3948 - val_acc: 0.8191

Epoch 3/40

304290/304290 [=====] - 169s 554us/step - loss: 0.2852 - acc: 0.8693 - val_loss: 0.4230 - val_acc: 0.8109

Epoch 4/40

304290/304290 [=====] - 169s 554us/step - loss: 0.2564 - acc: 0.8830 - val_loss: 0.4651 - val_acc: 0.8022

Epoch 5/40

304290/304290 [=====] - 167s 550us/step - loss: 0.2325 - acc: 0.8946 - val_loss: 0.4901 - val_acc: 0.8025

Epoch 6/40

304290/304290 [=====] - 166s 547us/step - loss: 0.2132 - acc: 0.9041 - val_loss: 0.5426 - val_acc: 0.7913

Epoch 7/40

304290/304290 [=====] - 175s 575us/step - loss: 0.1970 - acc: 0.9123 - val_loss: 0.5744 - val_acc: 0.7946

Epoch 8/40

304290/304290 [=====] - 171s 561us/step - loss: 0.1825 - acc: 0.9191 - val_loss: 0.6148 - val_acc: 0.7939

Epoch 9/40

304290/304290 [=====] - 169s 555us/step - loss: 0.1699 - acc: 0.9252 - val_loss: 0.6676 - val_acc: 0.7908

Epoch 10/40

304290/304290 [=====] - 169s 555us/step - loss: 0.1596 - acc: 0.9295 - val_loss: 0.7175 - val_acc: 0.7918

Epoch 11/40

304290/304290 [=====] - 161s 530us/step - loss: 0.1504 - acc: 0.9344 - val_loss: 0.7534 - val_acc: 0.7892

Epoch 12/40

304290/304290 [=====] - 172s 565us/step - loss: 0.1405 - acc: 0.9388 - val_loss: 0.8215 - val_acc: 0.7903

Epoch 13/40

304290/304290 [=====] - 173s 568us/step - loss: 0.1346 - acc: 0.9414 - val_loss: 0.8431 - val_acc: 0.7856

Epoch 14/40

304290/304290 [=====] - 165s 543us/step - loss: 0.1282 - acc: 0.9443 - val_loss: 0.8676 - val_acc: 0.7916

Epoch 15/40

304290/304290 [=====] - 169s 556us/step - loss: 0.1213 - acc: 0.9479 - val_loss: 0.8800 - val_acc: 0.7882

Epoch 16/40

304290/304290 [=====] - 165s 544us/step - loss: 0.1167 - acc: 0.9499 - val_loss: 0.9048 - val_acc: 0.7846

Epoch 17/40

304290/304290 [=====] - 167s 548us/step - loss: 0.1112 - acc: 0.9522 - val_loss: 0.9639 - val_acc: 0.7829

Epoch 18/40

304290/304290 [=====] - 175s 575us/step - loss: 0.1067 - acc: 0.9544 - val_loss: 0.9897 - val_acc: 0.7865

Epoch 19/40

304290/304290 [=====] - 169s 555us/step - loss: 0.1005 - acc: 0.9572 - val_loss: 1.0217 - val_acc: 0.7870

Epoch 20/40

304290/304290 [=====] - 168s 551us/step - loss: 0.0988 - acc: 0.9585 - val_loss: 1.0456 - val_acc: 0.7823

```

Epoch 21/40
304290/304290 [=====] - 166s 547us/step - loss: 0.0929 - acc: 0.9606 - va
l_loss: 1.0933 - val_acc: 0.7832
Epoch 22/40
304290/304290 [=====] - 166s 544us/step - loss: 0.0902 - acc: 0.9620 - va
l_loss: 1.1222 - val_acc: 0.7785
Epoch 23/40
304290/304290 [=====] - 165s 542us/step - loss: 0.0869 - acc: 0.9638 - va
l_loss: 1.1008 - val_acc: 0.7837
Epoch 24/40
304290/304290 [=====] - 165s 543us/step - loss: 0.0839 - acc: 0.9653 - va
l_loss: 1.1354 - val_acc: 0.7849
Epoch 25/40
304290/304290 [=====] - 165s 543us/step - loss: 0.0803 - acc: 0.9666 - va
l_loss: 1.1664 - val_acc: 0.7830
Epoch 26/40
304290/304290 [=====] - 165s 543us/step - loss: 0.0769 - acc: 0.9684 - va
l_loss: 1.2111 - val_acc: 0.7743
Epoch 27/40
304290/304290 [=====] - 164s 540us/step - loss: 0.0746 - acc: 0.9689 - va
l_loss: 1.2179 - val_acc: 0.7805
Epoch 28/40
304290/304290 [=====] - 165s 542us/step - loss: 0.0719 - acc: 0.9705 - va
l_loss: 1.2662 - val_acc: 0.7805
Epoch 29/40
304290/304290 [=====] - 167s 550us/step - loss: 0.0704 - acc: 0.9710 - va
l_loss: 1.2722 - val_acc: 0.7797
Epoch 30/40
304290/304290 [=====] - 163s 537us/step - loss: 0.0710 - acc: 0.9709 - va
l_loss: 1.2531 - val_acc: 0.7770
Epoch 31/40
304290/304290 [=====] - 164s 539us/step - loss: 0.0669 - acc: 0.9727 - va
l_loss: 1.2847 - val_acc: 0.7780
Epoch 32/40
304290/304290 [=====] - 167s 549us/step - loss: 0.0646 - acc: 0.9735 - va
l_loss: 1.3264 - val_acc: 0.7789
Epoch 33/40
304290/304290 [=====] - 170s 560us/step - loss: 0.0616 - acc: 0.9749 - va
l_loss: 1.3271 - val_acc: 0.7737
Epoch 34/40
304290/304290 [=====] - 168s 551us/step - loss: 0.0604 - acc: 0.9757 - va
l_loss: 1.3599 - val_acc: 0.7770
Epoch 35/40
304290/304290 [=====] - 167s 549us/step - loss: 0.0589 - acc: 0.9761 - va
l_loss: 1.3407 - val_acc: 0.7782
Epoch 36/40
304290/304290 [=====] - 168s 552us/step - loss: 0.0570 - acc: 0.9769 - va
l_loss: 1.3818 - val_acc: 0.7782
Epoch 37/40
304290/304290 [=====] - 167s 550us/step - loss: 0.0561 - acc: 0.9774 - va
l_loss: 1.3910 - val_acc: 0.7766
Epoch 38/40
304290/304290 [=====] - 168s 551us/step - loss: 0.0549 - acc: 0.9780 - va
l_loss: 1.4197 - val_acc: 0.7745
Epoch 39/40
304290/304290 [=====] - 170s 559us/step - loss: 0.0544 - acc: 0.9785 - va
l_loss: 1.3854 - val_acc: 0.7783
Epoch 40/40
304290/304290 [=====] - 170s 557us/step - loss: 0.0518 - acc: 0.9792 - va
l_loss: 1.4237 - val_acc: 0.7780

```

In [136]:

```

#trial 2
n_lstm_1 = 100
drop_lstm = 0.3
drop_embed = 0.3

qu1_input = Input(shape = (20,))
#drop1 = SpatialDropout1D(drop_embed)(qu1_input)
#drop1 = SpatialDropout1D(drop_embed)(qu1_input)
emb1 = Embedding(vocab1,128)(qu1_input)

qu2_input = Input(shape = (20,))

#drop2 = SpatialDropout1D(drop_embed)(qu2_input)

```

```

emb2 = Embedding(vocab2,128)(qu2_input)

core = Bidirectional(LSTM(128, dropout=drop_lstm))

# out_core = Bidirectional(LSTM(n_lstm_1, dropout=drop_lstm))

out1 = core(emb1)

# out_further_1 = out_core(out1)
out2 = core(emb2)


merged = concatenate([out1,out2])
predictions = Dense(1,activation = 'sigmoid')(merged)

model_big = Model([qu1_input, qu2_input], predictions)

model_big.compile(loss='binary_crossentropy', optimizer='sgd', metrics=['accuracy'])

history_big = model.fit([question1,question2],labels,epochs = 40,batch_size = 512,validation_data =
([X_validq1,X_validq2],y_valid)) #model big

```

Train on 304290 samples, validate on 100000 samples

```

Epoch 1/40
304290/304290 [=====] - 174s 572us/step - loss: 0.0513 - acc: 0.9793 - va
l_loss: 1.4221 - val_acc: 0.7772
Epoch 2/40
304290/304290 [=====] - 170s 560us/step - loss: 0.0508 - acc: 0.9794 - va
l_loss: 1.4492 - val_acc: 0.7768
Epoch 3/40
304290/304290 [=====] - 171s 561us/step - loss: 0.0493 - acc: 0.9803 - va
l_loss: 1.4428 - val_acc: 0.7766
Epoch 4/40
304290/304290 [=====] - 171s 560us/step - loss: 0.0471 - acc: 0.9809 - va
l_loss: 1.4855 - val_acc: 0.7731
Epoch 5/40
304290/304290 [=====] - 170s 559us/step - loss: 0.0460 - acc: 0.9816 - va
l_loss: 1.4668 - val_acc: 0.7767
Epoch 6/40
304290/304290 [=====] - 171s 563us/step - loss: 0.0456 - acc: 0.9822 - va
l_loss: 1.4932 - val_acc: 0.7776
Epoch 7/40
304290/304290 [=====] - 169s 556us/step - loss: 0.0437 - acc: 0.9826 - va
l_loss: 1.5330 - val_acc: 0.7716
Epoch 8/40
304290/304290 [=====] - 173s 569us/step - loss: 0.0434 - acc: 0.9830 - va
l_loss: 1.5476 - val_acc: 0.7718
Epoch 9/40
304290/304290 [=====] - 171s 561us/step - loss: 0.0427 - acc: 0.9831 - va
l_loss: 1.5717 - val_acc: 0.7705
Epoch 10/40
304290/304290 [=====] - 172s 565us/step - loss: 0.0424 - acc: 0.9835 - va
l_loss: 1.5436 - val_acc: 0.7738
Epoch 11/40
304290/304290 [=====] - 170s 558us/step - loss: 0.0413 - acc: 0.9837 - va
l_loss: 1.5686 - val_acc: 0.7757
Epoch 12/40
304290/304290 [=====] - 168s 553us/step - loss: 0.0394 - acc: 0.9847 - va
l_loss: 1.6120 - val_acc: 0.7755
Epoch 13/40
304290/304290 [=====] - 172s 564us/step - loss: 0.0395 - acc: 0.9844 - va
l_loss: 1.6069 - val_acc: 0.7741
Epoch 14/40
304290/304290 [=====] - 170s 559us/step - loss: 0.0384 - acc: 0.9849 - va
l_loss: 1.6004 - val_acc: 0.7753
Epoch 15/40
304290/304290 [=====] - 171s 562us/step - loss: 0.0394 - acc: 0.9845 - va
l_loss: 1.6457 - val_acc: 0.7746
Epoch 16/40
304290/304290 [=====] - 170s 560us/step - loss: 0.0377 - acc: 0.9853 - va
l_loss: 1.6198 - val_acc: 0.7743
Epoch 17/40
304290/304290 [=====] - 171s 561us/step - loss: 0.0377 - acc: 0.9854 - va

```

```
304290/304290 [=====] - 171s 561us/step - loss: 0.0377 - acc: 0.9854 - va
l_loss: 1.6284 - val_acc: 0.7756
Epoch 18/40
304290/304290 [=====] - 175s 575us/step - loss: 0.0382 - acc: 0.9852 - va
l_loss: 1.6127 - val_acc: 0.7747
Epoch 19/40
304290/304290 [=====] - 174s 573us/step - loss: 0.0360 - acc: 0.9860 - va
l_loss: 1.6317 - val_acc: 0.7753
Epoch 20/40
304290/304290 [=====] - 172s 566us/step - loss: 0.0356 - acc: 0.9864 - va
l_loss: 1.6188 - val_acc: 0.7758
Epoch 21/40
304290/304290 [=====] - 172s 566us/step - loss: 0.0347 - acc: 0.9866 - va
l_loss: 1.6282 - val_acc: 0.7748
Epoch 22/40
304290/304290 [=====] - 172s 567us/step - loss: 0.0337 - acc: 0.9868 - va
l_loss: 1.6511 - val_acc: 0.7732
Epoch 23/40
304290/304290 [=====] - 172s 566us/step - loss: 0.0327 - acc: 0.9874 - va
l_loss: 1.6203 - val_acc: 0.7710
Epoch 24/40
304290/304290 [=====] - 172s 565us/step - loss: 0.0336 - acc: 0.9869 - va
l_loss: 1.7050 - val_acc: 0.7714
Epoch 25/40
304290/304290 [=====] - 171s 564us/step - loss: 0.0334 - acc: 0.9871 - va
l_loss: 1.6916 - val_acc: 0.7721
Epoch 26/40
304290/304290 [=====] - 170s 560us/step - loss: 0.0332 - acc: 0.9872 - va
l_loss: 1.6946 - val_acc: 0.7733
Epoch 27/40
304290/304290 [=====] - 170s 559us/step - loss: 0.0324 - acc: 0.9875 - va
l_loss: 1.7037 - val_acc: 0.7728
Epoch 28/40
304290/304290 [=====] - 170s 560us/step - loss: 0.0308 - acc: 0.9883 - va
l_loss: 1.6984 - val_acc: 0.7727
Epoch 29/40
304290/304290 [=====] - 171s 563us/step - loss: 0.0307 - acc: 0.9883 - va
l_loss: 1.7228 - val_acc: 0.7741
Epoch 30/40
304290/304290 [=====] - 175s 573us/step - loss: 0.0304 - acc: 0.9881 - va
l_loss: 1.7440 - val_acc: 0.7764
Epoch 31/40
304290/304290 [=====] - 172s 565us/step - loss: 0.0305 - acc: 0.9881 - va
l_loss: 1.7383 - val_acc: 0.7714
Epoch 32/40
304290/304290 [=====] - 171s 561us/step - loss: 0.0297 - acc: 0.9885 - va
l_loss: 1.7133 - val_acc: 0.7714
Epoch 33/40
304290/304290 [=====] - 171s 563us/step - loss: 0.0288 - acc: 0.9889 - va
l_loss: 1.7160 - val_acc: 0.7748
Epoch 34/40
304290/304290 [=====] - 171s 561us/step - loss: 0.0302 - acc: 0.9886 - va
l_loss: 1.7197 - val_acc: 0.7691
Epoch 35/40
304290/304290 [=====] - 171s 561us/step - loss: 0.0295 - acc: 0.9885 - va
l_loss: 1.7455 - val_acc: 0.7730
Epoch 36/40
304290/304290 [=====] - 168s 551us/step - loss: 0.0293 - acc: 0.9887 - va
l_loss: 1.7378 - val_acc: 0.7743
Epoch 37/40
304290/304290 [=====] - 170s 560us/step - loss: 0.0293 - acc: 0.9886 - va
l_loss: 1.7633 - val_acc: 0.7702
Epoch 38/40
304290/304290 [=====] - 167s 550us/step - loss: 0.0286 - acc: 0.9891 - va
l_loss: 1.7821 - val_acc: 0.7739
Epoch 39/40
304290/304290 [=====] - 170s 559us/step - loss: 0.0274 - acc: 0.9896 - va
l_loss: 1.7631 - val_acc: 0.7723
Epoch 40/40
304290/304290 [=====] - 175s 574us/step - loss: 0.0276 - acc: 0.9896 - va
l_loss: 1.8035 - val_acc: 0.7742
```