Homework 4 Part 2

In [0]:
```
import numpy as np
import pandas as pd
from keras import optimizers
from keras.preprocessing.image import ImageDataGenerator, load_img
from keras.utils import to_categorical
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Dropout, Flatten, Dense, Activa
tion, BatchNormalization
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import random
import cv2
```

In [121]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

In [122]:
```
from google.colab import files
files.upload()
```

Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving kaggle.json to kaggle (1).json
```

Out[122]:
```
{'kaggle.json': b'{"username":"willywu","key":"dfad6221ebc97ed6ba2322f41aaf76
0c"}'}
```

In [0]:
```
!chmod 600 /root/.kaggle/kaggle.json
```

```
chmod: cannot access '/root/.kaggle/kaggle.json': No such file or directory
```

In [0]:
```
!pip install Pillow==4.1.1
!pip install fastai=0.7.0
!pip install torchtext==0.2.3
!pip install blosc
```

In [0]:
```
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
!kaggle competitions download -c dogs-vs-cats-redux-kernels-edition
```

```
Warning: Your Kaggle API key is readable by other users on this system! To fi
x this, you can run 'chmod 600 /root/.kaggle/kaggle.json'
Warning: Looks like you're using an outdated API Version, please consider upd
ating (server 1.5.6 / client 1.5.4)
Downloading test.zip to /content
 96% 261M/271M [00:01<00:00, 144MB/s]
100% 271M/271M [00:01<00:00, 152MB/s]
Downloading train.zip to /content
 98% 531M/544M [00:02<00:00, 214MB/s]
100% 544M/544M [00:02<00:00, 203MB/s]
Downloading sample_submission.csv to /content
  0% 0.00/111k [00:00<?, ?B/s]
100% 111k/111k [00:00<00:00, 42.9MB/s]
```

In [0]:
```
!unzip train.zip
```

In [123]:
```
!ls
```

```
 drive              kaggle.json   sample_data             test       train
'kaggle (1).json'   model.h5      sample_submission.csv   test.zip   train.zi
p
```

In [0]:
```
filenames = os.listdir("/content/train")
categories = []
for filename in filenames:
    category = filename.split('.')[0]
    if category == 'dog':
        categories.append(1)
    else:
        categories.append(0)

catdog = pd.DataFrame({
    'filename': filenames,
    'category': categories
})
```

In [192]:
```
train, test = train_test_split(catdog, test_size= 0.2, random_state=1)
print(train)
print(test)
```

```
            filename  category
6655    dog.7721.jpg         1
6085     dog.933.jpg         1
21848  dog.12285.jpg         1
5106    dog.4850.jpg         1
21856   cat.4997.jpg         0
...              ...       ...
10955   dog.9181.jpg         1
17289   dog.1214.jpg         1
5192    dog.7438.jpg         1
12172   cat.6003.jpg         0
235    dog.10374.jpg         1

[20000 rows x 2 columns]
           filename  category
21492  cat.3312.jpg         0
9488   dog.4630.jpg         1
16933   cat.503.jpg         0
12604  dog.3911.jpg         1
8222   cat.8832.jpg         0
...             ...       ...
11139  dog.3309.jpg         1
19053  dog.7769.jpg         1
7037   dog.1121.jpg         1
17119  dog.2241.jpg         1
20477  cat.1185.jpg         0

[5000 rows x 2 columns]
```

In [193]:
```
train["category"] = train["category"].replace({0: 'cat', 1: 'dog'})
test["category"] = test["category"].replace({0: 'cat', 1: 'dog'})
print(train)
```

```
            filename category
6655    dog.7721.jpg      dog
6085     dog.933.jpg      dog
21848  dog.12285.jpg      dog
5106    dog.4850.jpg      dog
21856   cat.4997.jpg      cat
...              ...      ...
10955   dog.9181.jpg      dog
17289   dog.1214.jpg      dog
5192    dog.7438.jpg      dog
12172   cat.6003.jpg      cat
235    dog.10374.jpg      dog

[20000 rows x 2 columns]

/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:1: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/st
able/user_guide/indexing.html#returning-a-view-versus-a-copy
  """Entry point for launching an IPython kernel.
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:2: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/st
able/user_guide/indexing.html#returning-a-view-versus-a-copy
```

In [0]:
```
width = 256
height= 256
channels=3
batch_size = 32
size_train = 20000
size_test = 5000
```

In [217]:
```
train_datagen = ImageDataGenerator(
    rescale=1./255.,
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True
    )

train_gen = train_datagen.flow_from_dataframe(
    train,
    "/content/train/",
    x_col='filename',
    y_col='category',
    target_size=(width,height),
    class_mode='categorical',
    batch_size=batch_size
)
test_datagen = ImageDataGenerator(rescale=1./255)
test_gen = test_datagen.flow_from_dataframe(
    test,
    "/content/train/",
    x_col='filename',
    y_col='category',
    target_size=(width,height),
    class_mode='categorical',
    batch_size=batch_size
)
```

```
Found 20000 validated image filenames belonging to 2 classes.
Found 5000 validated image filenames belonging to 2 classes.
```

In [218]:
```
model1 = Sequential()

model1.add(Conv2D(32, (3, 3), activation='relu', input_shape=(width, height, c
hannels)))
model1.add(BatchNormalization())
model1.add(MaxPooling2D(2, 2))

model1.add(Conv2D(64, (3, 3), activation='relu'))
model1.add(BatchNormalization())
model1.add(MaxPooling2D(2, 2))

model1.add(Conv2D(128, (3, 3), activation='relu'))
model1.add(BatchNormalization())
model1.add(MaxPooling2D(2, 2))

model1.add(Flatten())
model1.add(Dense(512, activation='relu'))
model1.add(BatchNormalization())
model1.add(Dense(2, activation='sigmoid'))

model1.compile(loss='binary_crossentropy', optimizer='rmsprop', metrics=['accu
racy'])

model1.summary()
```

```
Model: "sequential_33"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_85 (Conv2D)           (None, 254, 254, 32)      896
_________________________________________________________________
batch_normalization_109 (Bat (None, 254, 254, 32)      128
_________________________________________________________________
max_pooling2d_83 (MaxPooling (None, 127, 127, 32)      0
_________________________________________________________________
conv2d_86 (Conv2D)           (None, 125, 125, 64)      18496
_________________________________________________________________
batch_normalization_110 (Bat (None, 125, 125, 64)      256
_________________________________________________________________
max_pooling2d_84 (MaxPooling (None, 62, 62, 64)        0
_________________________________________________________________
conv2d_87 (Conv2D)           (None, 60, 60, 128)       73856
_________________________________________________________________
batch_normalization_111 (Bat (None, 60, 60, 128)       512
_________________________________________________________________
max_pooling2d_85 (MaxPooling (None, 30, 30, 128)       0
_________________________________________________________________
flatten_27 (Flatten)         (None, 115200)            0
_________________________________________________________________
dense_53 (Dense)             (None, 512)               58982912
_________________________________________________________________
batch_normalization_112 (Bat (None, 512)               2048
_________________________________________________________________
dense_54 (Dense)             (None, 2)                 1026
=================================================================
Total params: 59,080,130
Trainable params: 59,078,658
Non-trainable params: 1,472
_________________________________________________________________
```

In [220]:
```
epochs=50
history = model1.fit_generator(
    train_gen,
    epochs=epochs,
    validation_data=test_gen,
    validation_steps=size_test//batch_size,
    steps_per_epoch=size_train//batch_size,
)
```

```
Epoch 1/50
625/625 [==============================] - 318s 508ms/step - loss: 0.5763 - a
cc: 0.6975 - val_loss: 0.5287 - val_acc: 0.7318
Epoch 2/50
625/625 [==============================] - 314s 503ms/step - loss: 0.5171 - a
cc: 0.7437 - val_loss: 0.5363 - val_acc: 0.7338
Epoch 3/50
625/625 [==============================] - 315s 504ms/step - loss: 0.4836 - a
cc: 0.7678 - val_loss: 0.4719 - val_acc: 0.7715
Epoch 4/50
625/625 [==============================] - 316s 505ms/step - loss: 0.4518 - a
cc: 0.7893 - val_loss: 0.4066 - val_acc: 0.8193
Epoch 5/50
625/625 [==============================] - 315s 505ms/step - loss: 0.4264 - a
cc: 0.8051 - val_loss: 0.7722 - val_acc: 0.7122
Epoch 6/50
625/625 [==============================] - 310s 495ms/step - loss: 0.3949 - a
cc: 0.8260 - val_loss: 0.4152 - val_acc: 0.8139
Epoch 7/50
625/625 [==============================] - 308s 493ms/step - loss: 0.3745 - a
cc: 0.8353 - val_loss: 0.3762 - val_acc: 0.8329
Epoch 8/50
625/625 [==============================] - 311s 497ms/step - loss: 0.3499 - a
cc: 0.8498 - val_loss: 1.4729 - val_acc: 0.7158
Epoch 9/50
625/625 [==============================] - 308s 493ms/step - loss: 0.3359 - a
cc: 0.8563 - val_loss: 0.3646 - val_acc: 0.8444
Epoch 10/50
625/625 [==============================] - 306s 490ms/step - loss: 0.3204 - a
cc: 0.8620 - val_loss: 0.3071 - val_acc: 0.8786
Epoch 11/50
625/625 [==============================] - 304s 487ms/step - loss: 0.3083 - a
cc: 0.8680 - val_loss: 0.7969 - val_acc: 0.7487
Epoch 12/50
625/625 [==============================] - 303s 485ms/step - loss: 0.2980 - a
cc: 0.8717 - val_loss: 0.3328 - val_acc: 0.8578
Epoch 13/50
625/625 [==============================] - 303s 484ms/step - loss: 0.2912 - a
cc: 0.8761 - val_loss: 0.3115 - val_acc: 0.8668
Epoch 14/50
625/625 [==============================] - 302s 483ms/step - loss: 0.2796 - a
cc: 0.8818 - val_loss: 0.3273 - val_acc: 0.8760
Epoch 15/50
625/625 [==============================] - 305s 487ms/step - loss: 0.2811 - a
cc: 0.8830 - val_loss: 0.2707 - val_acc: 0.9007
Epoch 16/50
625/625 [==============================] - 307s 491ms/step - loss: 0.2727 - a
cc: 0.8867 - val_loss: 0.6059 - val_acc: 0.7972
Epoch 17/50
625/625 [==============================] - 310s 496ms/step - loss: 0.2636 - a
cc: 0.8886 - val_loss: 0.2870 - val_acc: 0.8950
Epoch 18/50
625/625 [==============================] - 318s 509ms/step - loss: 0.2616 - a
cc: 0.8893 - val_loss: 0.2525 - val_acc: 0.9039
Epoch 19/50
625/625 [==============================] - 312s 500ms/step - loss: 0.2563 - a
cc: 0.8928 - val_loss: 0.4316 - val_acc: 0.8044
```

```
Epoch 20/50
625/625 [==============================] - 310s 496ms/step - loss: 0.2560 - a
cc: 0.8938 - val_loss: 0.4405 - val_acc: 0.8395
Epoch 21/50
625/625 [==============================] - 308s 493ms/step - loss: 0.2514 - a
cc: 0.8933 - val_loss: 0.2370 - val_acc: 0.9102
Epoch 22/50
625/625 [==============================] - 308s 492ms/step - loss: 0.2409 - a
cc: 0.8987 - val_loss: 0.2420 - val_acc: 0.9175
Epoch 23/50
625/625 [==============================] - 308s 493ms/step - loss: 0.2477 - a
cc: 0.8979 - val_loss: 0.2334 - val_acc: 0.9012
Epoch 24/50
625/625 [==============================] - 306s 490ms/step - loss: 0.2410 - a
cc: 0.8990 - val_loss: 0.1998 - val_acc: 0.9242
Epoch 25/50
625/625 [==============================] - 304s 487ms/step - loss: 0.2393 - a
cc: 0.9023 - val_loss: 0.2608 - val_acc: 0.8923
Epoch 26/50
625/625 [==============================] - 304s 486ms/step - loss: 0.2382 - a
cc: 0.9036 - val_loss: 0.3624 - val_acc: 0.8759
Epoch 27/50
625/625 [==============================] - 303s 484ms/step - loss: 0.2277 - a
cc: 0.9036 - val_loss: 0.2462 - val_acc: 0.9177
Epoch 28/50
625/625 [==============================] - 307s 491ms/step - loss: 0.2281 - a
cc: 0.9051 - val_loss: 0.2506 - val_acc: 0.9098
Epoch 29/50
625/625 [==============================] - 317s 506ms/step - loss: 0.2241 - a
cc: 0.9062 - val_loss: 0.3205 - val_acc: 0.8896
Epoch 30/50
625/625 [==============================] - 317s 507ms/step - loss: 0.2304 - a
cc: 0.9053 - val_loss: 0.2183 - val_acc: 0.9148
Epoch 31/50
625/625 [==============================] - 317s 508ms/step - loss: 0.2239 - a
cc: 0.9067 - val_loss: 0.2494 - val_acc: 0.9180
Epoch 32/50
625/625 [==============================] - 318s 509ms/step - loss: 0.2235 - a
cc: 0.9076 - val_loss: 0.2586 - val_acc: 0.9010
Epoch 33/50
625/625 [==============================] - 318s 508ms/step - loss: 0.2169 - a
cc: 0.9103 - val_loss: 0.2981 - val_acc: 0.8702
Epoch 34/50
625/625 [==============================] - 319s 510ms/step - loss: 0.2172 - a
cc: 0.9108 - val_loss: 0.2513 - val_acc: 0.9014
Epoch 35/50
625/625 [==============================] - 318s 509ms/step - loss: 0.2158 - a
cc: 0.9108 - val_loss: 0.2600 - val_acc: 0.8945
Epoch 36/50
625/625 [==============================] - 317s 507ms/step - loss: 0.2166 - a
cc: 0.9097 - val_loss: 0.3367 - val_acc: 0.8815
Epoch 37/50
625/625 [==============================] - 318s 508ms/step - loss: 0.2144 - a
cc: 0.9131 - val_loss: 0.2167 - val_acc: 0.9209
Epoch 38/50
625/625 [==============================] - 319s 511ms/step - loss: 0.2123 - a
cc: 0.9135 - val_loss: 0.2373 - val_acc: 0.9097
```

```
Epoch 39/50
625/625 [==============================] - 319s 510ms/step - loss: 0.2170 - a
cc: 0.9131 - val_loss: 0.2286 - val_acc: 0.9118
Epoch 40/50
625/625 [==============================] - 320s 511ms/step - loss: 0.2115 - a
cc: 0.9153 - val_loss: 0.2710 - val_acc: 0.9104
Epoch 41/50
625/625 [==============================] - 319s 511ms/step - loss: 0.2069 - a
cc: 0.9161 - val_loss: 0.3120 - val_acc: 0.8754
Epoch 42/50
625/625 [==============================] - 321s 514ms/step - loss: 0.2083 - a
cc: 0.9161 - val_loss: 0.2036 - val_acc: 0.9211
Epoch 43/50
625/625 [==============================] - 319s 510ms/step - loss: 0.2067 - a
cc: 0.9157 - val_loss: 0.2087 - val_acc: 0.9274
Epoch 44/50
625/625 [==============================] - 318s 510ms/step - loss: 0.2047 - a
cc: 0.9162 - val_loss: 0.2387 - val_acc: 0.9126
Epoch 45/50
625/625 [==============================] - 320s 512ms/step - loss: 0.2058 - a
cc: 0.9165 - val_loss: 0.2221 - val_acc: 0.9183
Epoch 46/50
625/625 [==============================] - 317s 507ms/step - loss: 0.1996 - a
cc: 0.9184 - val_loss: 0.2313 - val_acc: 0.9052
Epoch 47/50
625/625 [==============================] - 318s 509ms/step - loss: 0.2070 - a
cc: 0.9159 - val_loss: 0.1959 - val_acc: 0.9282
Epoch 48/50
625/625 [==============================] - 320s 512ms/step - loss: 0.1979 - a
cc: 0.9197 - val_loss: 0.2051 - val_acc: 0.9220
Epoch 49/50
625/625 [==============================] - 322s 515ms/step - loss: 0.1971 - a
cc: 0.9206 - val_loss: 0.2747 - val_acc: 0.8922
Epoch 50/50
625/625 [==============================] - 320s 511ms/step - loss: 0.2007 - a
cc: 0.9172 - val_loss: 0.1921 - val_acc: 0.9295
```

In [0]:
```
model1.save_weights("model.h5")
```

In [0]:
```
!ls
```

```
drive        sample_data            test.zip  train.zip
kaggle.json  sample_submission.csv  train
```

In [0]:
```
!unzip test.zip
!ls
```

In [224]:
```
test_data = os.listdir('/content/test')
test_df = pd.DataFrame({
    'id': test_data
})
nTest = test_df.shape[0]

test_datagen = ImageDataGenerator(rescale = 1.0/255.)
test_gen = train_datagen.flow_from_dataframe(
    test_df,
    '/content/test',
    x_col='id',
    y_col=None,
    class_mode=None,
    target_size=(height, width),
    batch_size=batch_size,
)

predict = model1.predict_generator(test_gen, steps= np.round(nTest / batch_siz
e))
```

```
Found 12500 validated image filenames.
```

In [225]:
```
predict
```

Out[225]:
```
array([[1.07061476e-01, 8.95302176e-01],
       [9.74257112e-01, 2.66921818e-02],
       [9.99999762e-01, 2.38418579e-07],
       ...,
       [9.99975204e-01, 2.47061253e-05],
       [9.99316335e-01, 7.16328621e-04],
       [9.90611315e-01, 9.98291373e-03]], dtype=float32)
```

In [226]:
```
predict1 = np.argmax(predict, axis=-1)
predict1.shape[0]
```

Out[226]:
```
12500
```

In [0]:
```
test_df['label'] = predict1
test_df['image'] = test_df['id']
test_df['id'] = test_df.index + 1
submission = test_df[['id', 'label']]
submission.to_csv('submission2.csv', index=False)
```

In [230]:
```
submission
```

Out[230]:

	id	label
0	1	1
1	2	0
2	3	0
3	4	0
4	5	1
...	...	...
12495	12496	0
12496	12497	1
12497	12498	0
12498	12499	0
12499	12500	0

12500 rows × 2 columns

In [205]:
```
!ls
```

```
 drive              model.h5                submission.csv   train
'kaggle (1).json'   sample_data             test             train.zip
 kaggle.json        sample_submission.csv   test.zip
```

In [0]:
```
files.download("submission2.csv")
```