# Лабораторная работа №5. Применение сверточных нейронных сетей (бинарная классификация)

**Данные:** Набор данных DogsVsCats, который состоит из изображений различной размерности, содержащих фотографии собак и кошек. Обучающая выборка включает в себя 25 тыс. изображений (12,5 тыс. кошек: cat.0.jpg, ..., cat.12499.jpg и 12,5 тыс. собак: dog.0.jpg, ..., dog.12499.jpg), а контрольная выборка содержит 12,5 тыс. неразмеченных изображений. Скачать данные, а также проверить качество классификатора на тестовой выборке можно на сайте Kaggle -> https://www.kaggle.com/c/dogs-vs-cats/data (https://www.kaggle.com/c/dogs-vs-cats/data)

In [0]:
```
import os
from keras.preprocessing.image import ImageDataGenerator
import pandas as pd
import PIL
import numpy as np
from tqdm import tqdm
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Conv2D, MaxPooling2D, GlobalAveragePoo
from keras.models import Model
from keras.applications.vgg16 import VGG16
from keras.layers import Input
```

**Задание 1.** Загрузите данные. Разделите исходный набор данных на обучающую, валидационную и контрольную выборки.

In [0]:
```
os.chdir('/content/drive/My Drive/kaggle/dogs-vs-cats/train/tra
df = pd.DataFrame(columns=['animal', 'image'])

items = os.listdir()
for imageName in tqdm(items):
  animal =  0 if imageName.split(".")[0] == "cat" else 1
  df = df.append({'animal': animal, 'image': imageName}, ignore

print(df)
```

In [0]:
```
from google.colab import files

df.to_csv('trainPaths.csv')
files.download('trainPaths.csv')
```

In [9]:
```
df = pd.read_csv('/content/drive/My Drive/kaggle/dogs-vs-cats/t
print(df)
```

```
       Unnamed: 0  animal           image
0               0       0   cat.9578.jpg
1               1       0   cat.9563.jpg
2               2       0   cat.9544.jpg
3               3       0   cat.9574.jpg
4               4       0   cat.9561.jpg
...           ...     ...            ...
24996       24996       1  dog.10151.jpg
24997       24997       1  dog.10121.jpg
24998       24998       1  dog.10117.jpg
24999       24999       1  dog.10147.jpg
25000       25000       1  dog.10138.jpg

[25001 rows x 3 columns]
```

In [8]:

```
train_datagen = ImageDataGenerator(rescale=1./255., validation_
test_datagen = ImageDataGenerator(rescale=1./255., validation_s

train_generator = train_datagen.flow_from_dataframe(
        dataframe=df,
        directory='/content/drive/My Drive/kaggle/dogs-vs-cats/
        x_col="image",
        y_col="animal",
        target_size=(150, 150),
        batch_size=32,
        class_mode='raw')

test_generator = test_datagen.flow_from_dataframe(
        dataframe=df,
        directory='/content/drive/My Drive/kaggle/dogs-vs-cats/
        x_col="image",
        y_col="animal",
        target_size=(150, 150),
        batch_size=32,
        class_mode='raw')
```

```
/usr/local/lib/python3.6/dist-packages/keras_preprocessing/image/d
ataframe_iterator.py:273: UserWarning: Found 3 invalid image filen
ame(s) in x_col="image". These filename(s) will be ignored.
  .format(n_invalid, x_col)

Found 24998 validated image filenames.
Found 25000 validated image filenames.

/usr/local/lib/python3.6/dist-packages/keras_preprocessing/image/d
ataframe_iterator.py:273: UserWarning: Found 1 invalid image filen
ame(s) in x_col="image". These filename(s) will be ignored.
  .format(n_invalid, x_col)
```

**Задание 2.** Реализуйте глубокую нейронную сеть с как минимум тремя сверточными слоями. Какое качество классификации получено?

In [0]:
```
model = Sequential()
model.add(Conv2D(32, (5, 5), activation='relu', padding='same',
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu', padding='same')
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(128, (3, 3), activation='relu', padding='same'
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_initializer='he_
model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer="adam", loss='binary_crossentropy', met

model.fit_generator(
  train_generator,
  steps_per_epoch=20,
  epochs=5,
  validation_data=test_generator,
  validation_steps=8)
```

```
Epoch 1/5
20/20 [==============================] - 435s 22s/step - loss: 0.9
911 - acc: 0.4875 - val_loss: 0.6926 - val_acc: 0.5078
Epoch 2/5
 4/20 [=====>........................] - ETA: 6:02 - loss: 0.6943
- acc: 0.4297

/usr/local/lib/python3.6/dist-packages/keras/utils/data_utils.py:6
10: UserWarning: The input 403 could not be retrieved. It could be
because a worker has died.
  UserWarning)

20/20 [==============================] - 443s 22s/step - loss: 0.6
942 - acc: 0.4766 - val_loss: 0.6925 - val_acc: 0.5625
Epoch 3/5
20/20 [==============================] - 378s 19s/step - loss: 0.6
943 - acc: 0.5000 - val_loss: 0.6934 - val_acc: 0.4805
Epoch 4/5
20/20 [==============================] - 434s 22s/step - loss: 0.6
931 - acc: 0.5250 - val_loss: 0.7036 - val_acc: 0.4883
Epoch 5/5
20/20 [==============================] - 424s 21s/step - loss: 0.6
938 - acc: 0.5156 - val_loss: 0.6937 - val_acc: 0.4727
```

Out[28]:
```
<keras.callbacks.History at 0x7ff5ba687fd0>
```

**Задание 3.** Примените дополнение данных (data augmentation). Как это повлияло на качество классификатора?

In [10]:
```
train_datagen = ImageDataGenerator(rescale=1.0/255.0, width_shi
test_datagen = ImageDataGenerator(rescale=1.0/255.0)

train_generator = train_datagen.flow_from_dataframe(
        dataframe=df,
        directory='/content/drive/My Drive/kaggle/dogs-vs-cats/
        x_col="image",
        y_col="animal",
        target_size=(150, 150),
        batch_size=32,
        class_mode='raw')

test_generator = test_datagen.flow_from_dataframe(
        dataframe=df,
        directory='/content/drive/My Drive/kaggle/dogs-vs-cats/
        x_col="image",
        y_col="animal",
        target_size=(150, 150),
        batch_size=32,
        class_mode='raw')
```

```
/usr/local/lib/python3.6/dist-packages/keras_preprocessing/image/d
ataframe_iterator.py:273: UserWarning: Found 1 invalid image filen
ame(s) in x_col="image". These filename(s) will be ignored.
  .format(n_invalid, x_col)

Found 25000 validated image filenames.
Found 25000 validated image filenames.
```

In [0]:
```
model = Sequential()
model.add(Conv2D(32, (5, 5), activation='relu', padding='same',
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu', padding='same')
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(128, (3, 3), activation='relu', padding='same'
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_initializer='he_
model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer="adam", loss='binary_crossentropy', met

model.fit_generator(
  train_generator,
  steps_per_epoch=20,
  epochs=5,
  validation_data=test_generator,
  validation_steps=8)
```

```
Epoch 1/5
20/20 [==============================] - 365s 18s/step - loss: 0.7
956 - acc: 0.4953 - val_loss: 0.6920 - val_acc: 0.5703
Epoch 2/5
20/20 [==============================] - 350s 17s/step - loss: 0.6
935 - acc: 0.5016 - val_loss: 0.6904 - val_acc: 0.5195
Epoch 3/5
17/20 [========================>.....] - ETA: 52s - loss: 0.6938 -
acc: 0.4816

/usr/local/lib/python3.6/dist-packages/keras/utils/data_utils.py:6
10: UserWarning: The input 772 could not be retrieved. It could be
because a worker has died.
  UserWarning)

20/20 [==============================] - 408s 20s/step - loss: 0.6
935 - acc: 0.4906 - val_loss: 0.6934 - val_acc: 0.4805
Epoch 4/5
20/20 [==============================] - 349s 17s/step - loss: 0.6
920 - acc: 0.5203 - val_loss: 0.6898 - val_acc: 0.5078
Epoch 5/5
20/20 [==============================] - 347s 17s/step - loss: 0.6
919 - acc: 0.5422 - val_loss: 0.6885 - val_acc: 0.5430
```

Out[30]:
```
<keras.callbacks.History at 0x7ff5b9c93dd8>
```

**Задание 4.** Поэкспериментируйте с готовыми нейронными сетями (например, AlexNet, VGG16, Inception и т.п.), применив передаточное обучение. Как это повлияло на качество классификатора? Какой максимальный результат удалось получить на сайте Kaggle? Почему?

In [12]:
```python
from keras.applications import MobileNet

base_model=MobileNet(weights='imagenet',include_top=False)

x=base_model.output
x=GlobalAveragePooling2D()(x)
x=Dense(1024,activation='relu')(x)
x=Dense(1024,activation='relu')(x)
x=Dense(512,activation='relu')(x)
preds=Dense(1,activation='softmax')(x)
model=Model(inputs=base_model.input,outputs=preds)
model.summary()
```

```
/usr/local/lib/python3.6/dist-packages/keras_applications/mobilene
t.py:207: UserWarning: `input_shape` is undefined or non-square, o
r `rows` is not in [128, 160, 192, 224]. Weights for input shape (
224, 224) will be loaded as the default.
  warnings.warn('`input_shape` is undefined or non-square, '

Model: "model_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_5 (InputLayer)         (None, None, None, 3)     0         
_________________________________________________________________
conv1_pad (ZeroPadding2D)    (None, None, None, 3)     0         
_________________________________________________________________
conv1 (Conv2D)               (None, None, None, 32)    864       
_________________________________________________________________
conv1_bn (BatchNormalization (None, None, None, 32)    128       
_________________________________________________________________
conv1_relu (ReLU)            (None, None, None, 32)    0         
```

In [0]:
```python
for layer in model.layers:
    layer.trainable=False
```

In [18]:
```
model.compile(optimizer='Adam',loss='binary_crossentropy',metri

model.fit_generator(
  train_generator,
  steps_per_epoch=20,
  epochs=5,
  validation_data=test_generator,
  validation_steps=8)
```

```
Epoch 1/5
20/20 [==============================] - 273s 14s/step - loss: 8.0
459 - acc: 0.4953 - val_loss: 8.3448 - val_acc: 0.4766
Epoch 2/5
20/20 [==============================] - 249s 12s/step - loss: 7.8
466 - acc: 0.5078 - val_loss: 7.9712 - val_acc: 0.5000
Epoch 3/5
20/20 [==============================] - 233s 12s/step - loss: 7.7
470 - acc: 0.5141 - val_loss: 8.0335 - val_acc: 0.4961
Epoch 4/5
20/20 [==============================] - 214s 11s/step - loss: 7.6
723 - acc: 0.5188 - val_loss: 7.5353 - val_acc: 0.5273
Epoch 5/5
20/20 [==============================] - 253s 13s/step - loss: 7.9
463 - acc: 0.5016 - val_loss: 8.0335 - val_acc: 0.4961
```

Out[18]:
```
<keras.callbacks.History at 0x7f298a378048>
```

In [0]:
```

```