1.go to https://github.com/miaow1988/SqueezeNet_v1.2 and download the 'symbol.json' and '.params' files (there is not a 'synset.txt' file! so don't use these lines, Hint: just comment these lines).

Install MXNet v1.5 (hint: create a new conda environmet with python 3, pip install mxnet==1.5.1) and follow the same steps of the lecture (part: Using pre-trained models as feature extractors). Find the flatten output layer and create a feature extractor (hint: It should be a numpy array of 1000 elements).

Download the dogs versus cats training folder from https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/data (Remember the number of images is 12500 for each class).

Extract the array of features for different number of images (N: 10, 100, 500, 1000, also 5000 and 12500) and for each value train your favorite binary classifier (only one!!!) using GridSearch to optimize some hyperparameters. Consider to use https://notebooks.csc.fi if you have computational limitations.

Report the accuracy for each value of N and the computational time during the training step.

In [8]:
```
!pip install mxnet==1.5.1
```

```
Collecting mxnet==1.5.1
  Downloading mxnet-1.5.1-py2.py3-none-manylinux1_x86_64.whl (23.1 MB)
     |████████████████████████████████| 23.1 MB 42.9 MB/s eta 0:00:01  |████
████████                        | 8.7 MB 1.9 MB/s eta 0:00:081.9 MB/s eta 0:00:07/s
eta 0:00:06��████████████████                | 13.7 MB 1.9 MB/s eta 0:00:060:04.
5 MB 1.9 MB/s eta 0:00:03��██████████    | 20.4 MB 1.9 MB/s eta 0:00:02�██ | 2
2.8 MB 1.9 MB/s eta 0:00:01
Requirement already satisfied: numpy<2.0.0,>1.16.0 in /opt/conda/lib/python3.7
/site-packages (from mxnet==1.5.1) (1.18.2)
Requirement already satisfied: requests<3,>=2.20.0 in /opt/conda/lib/python3.7
/site-packages (from mxnet==1.5.1) (2.23.0)
Collecting graphviz<0.9.0,>=0.8.1
  Downloading graphviz-0.8.4-py2.py3-none-any.whl (16 kB)
Requirement already satisfied: idna<3,>=2.5 in /opt/conda/lib/python3.7/site-p
ackages (from requests<3,>=2.20.0->mxnet==1.5.1) (2.9)
Requirement already satisfied: chardet<4,>=3.0.2 in /opt/conda/lib/python3.7/s
ite-packages (from requests<3,>=2.20.0->mxnet==1.5.1) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/lib/python3.7/
site-packages (from requests<3,>=2.20.0->mxnet==1.5.1) (2019.11.28)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /opt
/conda/lib/python3.7/site-packages (from requests<3,>=2.20.0->mxnet==1.5.1) (1
.25.7)
Installing collected packages: graphviz, mxnet
  Attempting uninstall: graphviz
    Found existing installation: graphviz 0.13.2
    Uninstalling graphviz-0.13.2:
      Successfully uninstalled graphviz-0.13.2
Successfully installed graphviz-0.8.4 mxnet-1.5.1
```

In [5]:
```
import mxnet as mx

path = 'https://github.com/miaow1988/SqueezeNet_v1.2/'
[
    mx.test_utils.download(path +'/blob/master/model-symbol.json'),
    mx.test_utils.download(path +'blob/master/model-0000.params'),
]
```

Out[5]:
```
['model-symbol.json', 'model-0000.params']
```

In [8]:
```
!wget https://github.com/miaow1988/SqueezeNet_v1.2/raw/master/model-symbol.js
!wget https://github.com/miaow1988/SqueezeNet_v1.2/raw/master/model-0000.para
```

```
--2020-11-07 10:23:47--  https://github.com/miaow1988/SqueezeNet_v1.2/raw/mast
er/model-symbol.json
Resolving github.com (github.com)... 140.82.121.3
Connecting to github.com (github.com)|140.82.121.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/miaow1988/SqueezeNet_v1.2/master/m
odel-symbol.json [following]
--2020-11-07 10:23:47--  https://raw.githubusercontent.com/miaow1988/SqueezeNe
t_v1.2/master/model-symbol.json
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.84.
133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.84
.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24187 (24K) [text/plain]
Saving to: 'model-symbol.json'

model-symbol.json   100%[===================>]  23.62K  --.-KB/s    in 0.01s

2020-11-07 10:23:47 (1.57 MB/s) - 'model-symbol.json' saved [24187/24187]

--2020-11-07 10:23:48--  https://github.com/miaow1988/SqueezeNet_v1.2/raw/mast
er/model-0000.params
Resolving github.com (github.com)... 140.82.121.3
Connecting to github.com (github.com)|140.82.121.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/miaow1988/SqueezeNet_v1.2/master/m
odel-0000.params [following]
--2020-11-07 10:23:49--  https://raw.githubusercontent.com/miaow1988/SqueezeNe
t_v1.2/master/model-0000.params
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.84.
133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.84
.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4945034 (4.7M) [application/octet-stream]
Saving to: 'model-0000.params'

model-0000.params   100%[===================>]   4.72M  12.8MB/s    in 0.4s

2020-11-07 10:23:50 (12.8 MB/s) - 'model-0000.params' saved [4945034/4945034]
```

In [10]:

```
! pip install opencv-python-headless
! pip install --user kaggle --upgrade
```

```
Collecting opencv-python-headless
  Downloading opencv_python_headless-4.4.0.46-cp37-cp37m-manylinux2014_x86_64.
whl (36.7 MB)
     |████████████████████████████████| 36.7 MB 42.8 MB/s eta 0:00:01MB 2.1 M
B/s eta 0:00:13                    | 13.6 MB 2.1 MB/s eta 0:00:12ﾀﾀ████████
| 17.5 MB 2.1 MB/s eta 0:00:10████████████████                | 18.9 MB 2.1 MB
```

```
/s eta 0:00:09████              | 21.1 MB 2.1 MB/s eta 0:00:08▒████████
| 23.7 MB 42.8 MB/s eta 0:00:01101███████████       | 28.9 MB 42.8 MB/s eta 0:
00:01 | 30.8 MB 42.8 MB/s eta 0:00:01|████████████████████████████    | 33.1
MB 42.8 MB/s eta 0:00:01
Requirement already satisfied: numpy>=1.14.5 in /opt/conda/lib/python3.7/site-
packages (from opencv-python-headless) (1.18.2)
Installing collected packages: opencv-python-headless
Successfully installed opencv-python-headless-4.4.0.46
Collecting kaggle
  Downloading kaggle-1.5.9.tar.gz (58 kB)
     |████████████████████████████████| 58 kB 1.5 MB/s eta 0:00:011
Requirement already satisfied, skipping upgrade: six>=1.10 in /opt/conda/lib/p
ython3.7/site-packages (from kaggle) (1.14.0)
Requirement already satisfied, skipping upgrade: certifi in /opt/conda/lib/pyt
hon3.7/site-packages (from kaggle) (2019.11.28)
Requirement already satisfied, skipping upgrade: python-dateutil in /opt/conda
/lib/python3.7/site-packages (from kaggle) (2.8.1)
Requirement already satisfied, skipping upgrade: requests in /opt/conda/lib/py
thon3.7/site-packages (from kaggle) (2.23.0)
Requirement already satisfied, skipping upgrade: tqdm in /opt/conda/lib/python
3.7/site-packages (from kaggle) (4.43.0)
Collecting python-slugify
  Downloading python-slugify-4.0.1.tar.gz (11 kB)
Collecting slugify
  Downloading slugify-0.0.1.tar.gz (1.2 kB)
Requirement already satisfied, skipping upgrade: urllib3 in /opt/conda/lib/pyt
hon3.7/site-packages (from kaggle) (1.25.7)
Requirement already satisfied, skipping upgrade: idna<3,>=2.5 in /opt/conda/li
b/python3.7/site-packages (from requests->kaggle) (2.9)
Requirement already satisfied, skipping upgrade: chardet<4,>=3.0.2 in /opt/con
da/lib/python3.7/site-packages (from requests->kaggle) (3.0.4)
Collecting text-unidecode>=1.3
  Downloading text_unidecode-1.3-py2.py3-none-any.whl (78 kB)
     |████████████████████████████████| 78 kB 3.4 MB/s eta 0:00:011
Building wheels for collected packages: kaggle, python-slugify, slugify
  Building wheel for kaggle (setup.py) ... done
  Created wheel for kaggle: filename=kaggle-1.5.9-py3-none-any.whl size=73265
sha256=d2a32e42c1c9ad13d7feeaae1eac66f910874e3ef556195531f1285a0431c762
  Stored in directory: /home/jovyan/.cache/pip/wheels/09/25/76/1bbe8ad0c423e86
55942b6d3c781f58e0ea2791bf8ee8985b3
  Building wheel for python-slugify (setup.py) ... done
  Created wheel for python-slugify: filename=python_slugify-4.0.1-py2.py3-none
-any.whl size=6767 sha256=28a60d02dcab1924182c050daff80e8dba8e396d89a7d5361aec
9dd8f76d7fa5
  Stored in directory: /home/jovyan/.cache/pip/wheels/48/1b/6f/5c1cfab22eacbe0
095fc619786da6571b55253653c71324b5c
  Building wheel for slugify (setup.py) ... done
  Created wheel for slugify: filename=slugify-0.0.1-py3-none-any.whl size=1908
sha256=33bd827d718d4f5f03d9bf292f89e4e5a74fbaba47dc0df4fa6efb32f1503135
  Stored in directory: /home/jovyan/.cache/pip/wheels/d4/7b/0d/bd65011a1b44284
3bb4043e396f727ab0f1e76050355b9156a
Successfully built kaggle python-slugify slugify
Installing collected packages: text-unidecode, python-slugify, slugify, kaggle
```

```
  WARNING: The script slugify is installed in '/home/jovyan/.local/bin' which
is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this wa
rning, use --no-warn-script-location.
  WARNING: The script kaggle is installed in '/home/jovyan/.local/bin' which i
s not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this wa
rning, use --no-warn-script-location.
Successfully installed kaggle-1.5.9 python-slugify-4.0.1 slugify-0.0.1 text-un
idecode-1.3
```

In [15]:
```
# if "opencv-python" not in pkgs:
#     !pip install opencv-python
# if "opencv-contrib-python" not in pkgs:
#     !pip install opencv-contrib-python
```

Upload your https://github.com/Kaggle/kaggle-api#api-credentials

In [11]:
```
mkdir ~/.kaggle
```

```
mkdir: cannot create directory ‘/home/jovyan/.kaggle’: File exists
```

In [12]:
```
! mv kaggle.json .kaggle/kaggle.json
```

In [13]:
```
! chmod 600 ~/.kaggle/kaggle.json
```

In [17]:
```
! ls -l ~/.kaggle
```

```
total 4
-rw-------. 1 jovyan root 63 Nov  7 10:19 kaggle.json
```

In [20]:
```
! ls -l ~/.local/bin
```

```
total 8
-rwxr-xr-x. 1 jovyan root 217 Nov  7 10:24 kaggle
-rwxr-xr-x. 1 jovyan root 333 Nov  7 10:24 slugify
```

In [19]:
```
!pip show kaggle
```

```
Name: kaggle
Version: 1.5.9
Summary: Kaggle API
Home-page: https://github.com/Kaggle/kaggle-api
Author: Kaggle
Author-email: support@kaggle.com
License: Apache 2.0
Location: /home/jovyan/.local/lib/python3.7/site-packages
Requires: tqdm, certifi, requests, urllib3, six, slugify, python-slugify, pyth
on-dateutil
Required-by:
```

In [21]:
```
! find / -name kaggle 2>/dev/null
```

```
/home/jovyan/.local/lib/python3.7/site-packages/kaggle
/home/jovyan/.local/bin/kaggle
```

In [24]:
```
# !kaggle competitions download -c dogs-vs-cats-redux-kernels-edition
```

In [1]:
```
! unzip dogs-vs-cats-redux-kernels-edition.zip
```

```
Archive:  dogs-vs-cats-redux-kernels-edition.zip
   creating: dogs-vs-cats-redux-kernels-edition/
  inflating: dogs-vs-cats-redux-kernels-edition/test.zip
   creating: __MACOSX/
   creating: __MACOSX/dogs-vs-cats-redux-kernels-edition/
  inflating: __MACOSX/dogs-vs-cats-redux-kernels-edition/._test.zip
  inflating: dogs-vs-cats-redux-kernels-edition/train.zip
  inflating: __MACOSX/dogs-vs-cats-redux-kernels-edition/._train.zip
  inflating: dogs-vs-cats-redux-kernels-edition/sample_submission.csv
  inflating: __MACOSX/dogs-vs-cats-redux-kernels-edition/._sample_submission.c
sv
  inflating: __MACOSX/._dogs-vs-cats-redux-kernels-edition
```

In [5]:
```
! unzip ~/dogs-vs-cats-redux-kernels-edition/train.zip
```

```
Archive:  /home/jovyan/dogs-vs-cats-redux-kernels-edition/train.zip
   creating: train/
  inflating: train/cat.0.jpg
  inflating: train/cat.1.jpg
  inflating: train/cat.10.jpg
  inflating: train/cat.100.jpg
  inflating: train/cat.1000.jpg
  inflating: train/cat.10000.jpg
  inflating: train/cat.10001.jpg
  inflating: train/cat.10002.jpg
  inflating: train/cat.10003.jpg
  inflating: train/cat.10004.jpg
  inflating: train/cat.10005.jpg
  inflating: train/cat.10006.jpg
```

```
 inflating: train/dog.9977.jpg
 inflating: train/dog.9978.jpg
 inflating: train/dog.9979.jpg
 inflating: train/dog.998.jpg
 inflating: train/dog.9980.jpg
 inflating: train/dog.9981.jpg
 inflating: train/dog.9982.jpg
 inflating: train/dog.9983.jpg
 inflating: train/dog.9984.jpg
 inflating: train/dog.9985.jpg
 inflating: train/dog.9986.jpg
 inflating: train/dog.9987.jpg
 inflating: train/dog.9988.jpg
 inflating: train/dog.9989.jpg
 inflating: train/dog.999.jpg
 inflating: train/dog.9990.jpg
 inflating: train/dog.9991.jpg
 inflating: train/dog.9992.jpg
 inflating: train/dog.9993.jpg
 inflating: train/dog.9994.jpg
 inflating: train/dog.9995.jpg
 inflating: train/dog.9996.jpg
 inflating: train/dog.9997.jpg
 inflating: train/dog.9998.jpg
 inflating: train/dog.9999.jpg
```

In [9]:

```
import mxnet as mx
import numpy as np
from mxnet import nd,autograd
from mxnet import gluon

%matplotlib inline
import matplotlib.pyplot as plt
import cv2

# define a simple data batch
from collections import namedtuple
Batch = namedtuple('Batch', ['data'])
```

In [11]:

```
context=mx.cpu()
```

In [13]:

```
mod = mx.mod.Module(symbol=sym, context=mx.cpu(), label_names=None)
mod.bind(for_training=False, data_shapes=[('data', (1,3,224,224))],
         label_shapes=mod._label_shapes)
mod.set_params(arg_params, aux_params, allow_missing=True)
# with open('pretrained/synset.txt', 'r') as f:
#     labels = [l.rstrip() for l in f]
```

In [100…

```
sym, arg_params, aux_params = mx.model.load_checkpoint('model', 0)
```

In [17]:

```
def get_image(url, show=False):
    if url.startswith('http'):
        # download and show the image
        fname = mx.test_utils.download(url)
    else:
        fname = url
    img = cv2.cvtColor(cv2.imread(fname), cv2.COLOR_BGR2RGB)
    if img is None:
         return None
    if show:
         plt.imshow(img)
         plt.axis('off')
    # convert into format (batch, RGB, width, height)
    img = cv2.resize(img, (224, 224))
    img = np.swapaxes(img, 0, 2)
    img = np.swapaxes(img, 1, 2)
    img = img[np.newaxis, :]
    return img

def predict(url):
    img = get_image(url, show=True)
    # compute the predict probabilities
    mod.forward(Batch([mx.nd.array(img)]))
    prob = mod.get_outputs()[0].asnumpy()
    # print the top-5
    prob = np.squeeze(prob)
    a = np.argsort(prob)[::-1]
    for i in a[0:5]:
        print('probability=%f, class=%s' %(prob[i], labels[i]))
```

In [48]:

```
# list the last 10 layers
all_layers = sym.get_internals()
all_layers.list_outputs()[-3:]
```

Out[48]:

```
['flatten0_output', 'softmax_label', 'softmax_output']
```

In [14]:

```
fe_sym = all_layers['flatten0_output']
fe_mod = mx.mod.Module(symbol=fe_sym, context=mx.cpu(), label_names=None)
fe_mod.bind(for_training=False, data_shapes=[('data', (1,3,224,224))])
fe_mod.set_params(arg_params, aux_params)
```

In [15]:
```
def get_features(img):
    fe_mod.forward(Batch([mx.nd.array(img)]))
    features = fe_mod.get_outputs()[0].asnumpy()
    return features
```

In [18]:
```
img = get_image('https://icatcare.org/app/uploads/2018/07/Thinking-of-getting
features = get_features(img)
print("{}\n shape: {}".format(features,features.shape))
```

```
[[ 2.731811    3.9008982   4.8866773   6.902369    5.587853    5.5976596
   3.6299744   5.6956005   5.783274    5.9859447  10.328437    7.249119
  10.089799   11.63578     5.579315   10.29896     9.659985    9.824672
   5.997461    9.134588   10.0658655  10.166415    9.8080845   6.289982
  13.240795    1.9723082   7.6495514   3.712172    2.7841299   6.2934337
   6.9453053   8.631689    7.29645     6.801901    6.52334     5.698734
   8.282388    3.830843   10.181621    8.7683935  12.962755    9.04188
   9.2452     10.77944    11.305762    5.7258263  11.200751   10.964636
   3.4149942   5.8307867   7.5116644   9.0251      3.9738111   3.9767187
   8.057208    4.70411     8.180446    4.212362    7.947       5.900941
   7.274863    9.651376    6.8402615   3.0084639   4.359813    4.1929684
   8.628105    3.9853938   5.851279    7.3413982   7.4339814   6.6312613
   8.691614    8.439708    8.843549    9.059312    9.449075   12.678203
  11.697756    7.8836474   7.3444357   6.451958    6.8108234  11.757459
  10.983695    7.6246867   5.50387     7.03473    12.915622    8.201405
   3.994502    5.4100966   6.856489    6.924605    9.808682    6.971078
   4.193388    3.5059521   1.6024159   4.7404647   3.3018126   2.9084258
   2.415486    5.3541465   8.748269    5.6931176   5.7850704   6.1290493
   5.1263065   5.997225    7.454259    6.640497   11.303702    7.120902
   9.347335    8.703756    4.8624525  12.27395     6.1802354   4.480561
   5.6459785   1.0452362   6.122116    5.4823976  11.299996    4.1871934
   7.719489    2.937037    4.467442    4.9877334   5.112029    6.401343
   9.139223    8.80435     6.144146    2.6501632   4.6860394   3.7774022
  10.005826    8.510313    7.547855    7.3255305   7.330723    3.8226004
   4.816145    6.8668685   4.516519    7.42678     6.5116243   3.608323
   7.267048   10.171126    7.746412    5.281473    8.630499    5.626668
   6.502434   13.557568    9.849358    4.837753    4.3339353   3.7696319
   5.360929    2.2633295   2.9870687   2.431563    3.9928882   2.2981818
   3.342405    6.635318    2.1925945   8.712073    4.962669    8.373492
   4.807753    2.6049435   4.978068    3.2130506   7.113287    1.5388217
   2.2567575   3.4583302   2.7490807   4.909944    4.181       3.6504633
   5.1258836   7.6492605   5.345396    4.6102114   3.7957761   2.6940234
   6.086538    5.387833    1.9049265   8.116254    7.423186    3.3709123
   5.39928     6.287825    3.9034204   6.0577965   4.336672    7.3916006
   5.458247    3.5410912   0.81163764  4.913134    3.077363    2.524287
   3.4502792   5.504821    5.5890636   3.1113174   1.8214495   4.9293275
   4.6584773   5.1635823   3.283465    3.5414634   2.3487635   1.1612074
   4.398783    7.2059193   7.16017     5.134449    3.4099345   7.284354
   1.0771716   3.487802    7.6779857   6.426784    6.5381465   2.5813906
   4.531569    5.8846455   6.2358975   7.270987    3.3144867   5.6216626
   4.9576645   2.6886454   7.497139    3.8266113   2.013793    6.8338447
```

```
  6.7536345  17.710531    8.317595   11.241352    7.7849245  11.963486
  3.5570066   8.755916    5.8202972   6.244552    9.4146185   8.909975
 11.538079    6.4253035  17.255009   13.751663   11.622289   15.469001
  4.237758    7.736162   10.365318   13.24044    18.205471   14.377344
  9.415742   10.363938    3.6586409   9.332913    9.287965    9.225867
 10.726653    2.3643985   3.2395904   2.773296    7.177313   10.607796
 13.280321    7.7800007   3.320212   10.017647    6.189191    6.967745
  5.764682    8.264951    2.065886    9.877997    9.344261   11.44254
  5.910216    4.108808    5.1819353   2.406356    4.4888806   4.482684
  3.3455486  12.267346    6.304007    5.7975836   5.489004    2.9057488
  8.029712    7.311815    5.331028    6.780764    7.2698026   7.03161
  6.9609914   8.205074    6.2120485   9.704169    4.3887997   9.308927
  5.334109    6.4561872   5.7876287   4.3589535   7.1413755   4.1641693
  8.003898    8.619182    2.5742774   2.4626925   3.8087509   5.987408
 10.418253    9.594042   18.342413   11.671421    2.637681    9.242509
  2.5423734   6.47289     5.5148525   5.139831    1.7740538   3.9243221
  5.780046    5.5538406   2.6732502   7.5682645   8.417677    2.763147
 10.0679455   9.356203    5.7668076  13.894629    7.128599    8.015945
  4.268324    7.279767    3.9210467   4.57002     6.2065      5.7564244
  7.1734467   3.3807797  11.519496   16.3085    ]]
shape: (1, 1000)
```

In [19]:

```
# from https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/data
from os import listdir
from os.path import isfile, join
import os


mypath = join(os.getcwd(),'train')

cats_imgs = [join(mypath,f) for f in listdir(mypath) if f.startswith('cat')]
dogs_imgs = [join(mypath,f) for f in listdir(mypath) if f.startswith('dog')]
```

In [20]:

```
print("cats: {} and dogs: {}".format(len(cats_imgs),len(dogs_imgs)))
```

```
cats: 12500 and dogs: 12500
```

In [22]:

```
Nmax = 100 #(N: 10, 100, 500, 1000, also 5000 and 12500)
cats_features = [get_features(get_image(img)).ravel() for img in cats_imgs[:Nm
dogs_features = [get_features(get_image(img)).ravel() for img in dogs_imgs[:Nm
```

In [24]:

```
Y_cats = np.array(Nmax * [1])
Y_dogs = np.array(Nmax * [0])
```

In [25]:

```
X_cvd = np.vstack([cats_features,dogs_features])
Y_cvd = np.vstack([Y_cats,Y_dogs]).ravel()
```

In [26]:
```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_cvd, Y_cvd, random_stat
```

In [27]:
```
from sklearn.linear_model import LogisticRegression

lg = LogisticRegression().fit(X_train, y_train)

print("Test set score: {:.2f}".format(lg.score(X_test, y_test)))
```

```
Test set score: 0.88
/opt/conda/lib/python3.7/site-packages/sklearn/linear_model/_logistic.py:940:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regress
ion
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
```

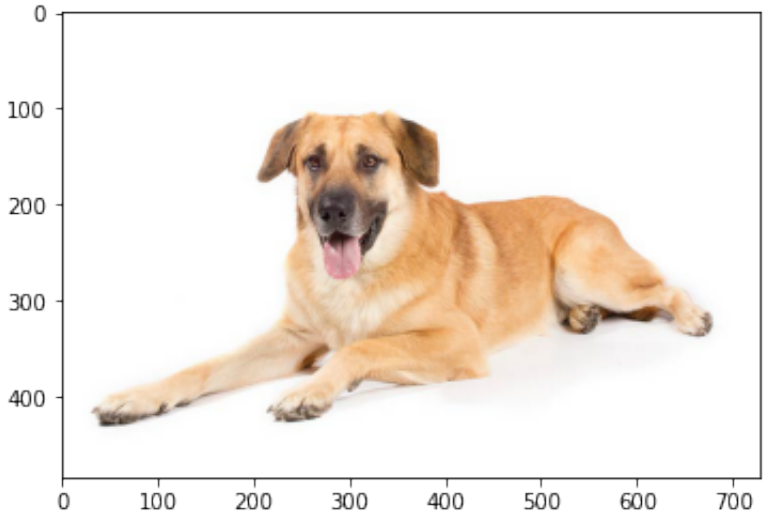
In [30]:
```
img = get_image('https://s3.amazonaws.com/cdn-origin-etr.akc.org/wp-content/u
features = get_features(img)
print("{}\n shape: {}".format(features,features.shape))
```

```
[[ 3.8680282   3.0770736   3.8007517   3.9388447   4.4794497   2.513034
   1.0257548   3.3912466   4.1260457   1.1125107   3.0114415   2.2824717
   3.308097    3.6617973   2.2812922   1.8932449   4.174476    3.710502
   0.45999247  4.1961174   1.2289764   9.014702    6.0741653   5.186479
   4.293989    3.2768373   4.330217    3.7503705   6.209451    4.7506633
   3.809687    8.039387    5.815378    4.479296    4.644457    2.287041
   4.105688    3.3356824   8.155916    4.587722    4.465425    3.855896
   4.043045    5.1172795   4.4702134   3.342398    5.4953494   6.7198224
   3.84865     1.182495    2.0083091   9.924515    2.7754364   2.005786
   3.0824454   1.4339454   2.6161735   1.7753203   1.2789447   2.4843435
   1.5804738   6.9447136   4.049789    2.9182622   1.8067827   1.929847
   2.0441859   2.8852274   2.3605494   5.1194777   2.5484276   5.3415647
   3.779574    3.719067    4.224366    4.0597167   2.472064    2.177067
   6.892461    4.2934427   4.110542    5.4240932   6.0283465   4.938808
   1.0311867   1.9734994   4.873224   10.826043    6.586989   10.874734
   3.7613354   2.369898    2.6786683   1.7927071   7.2485766   1.7895155
   2.0912905   4.3573813   1.0156956   4.274913    1.7799224   8.096338
   3.7537284   7.285345    7.295822    5.0679016   2.7826424   6.1655016
   2.4368806   1.3257196   4.5575867   6.473884   11.895296    6.8320866
   5.361       2.6531858   4.564212    5.8689027   1.9365345   2.6203759
   5.183484    1.7165029   3.0182095   2.274215    4.6858997   5.2946324
   7.652743    1.6650032   1.8852196   4.6898293   2.1779253   1.8760023
   1.9162517   3.1303358   2.3637066   0.35587963  2.2771838   1.5375512
```

```
  5.629182    7.3680015   9.480289    5.6531177   7.636259    4.543409
  2.8130362   2.5628386  10.775148    8.83947     9.697046    8.9554405
  5.122448    2.8479238   6.6367116  10.020451    5.7062306   5.554301
 10.600308   11.314349    5.623864    9.765764    9.736605    8.770255
  6.3605294   4.037713    4.383122    8.11133     3.4699535   5.010916
  9.313533   12.367925    6.7642603   3.3070233   7.72455     3.3671126
  2.723685   13.013277    4.3527923   8.93426     3.0037897   2.9002788
  9.682216    2.4089184  10.428298    5.4077883   3.6294236   0.82561
  8.775564    8.402793    8.439979    2.38039     5.6820807   4.6136727
  8.871776    3.9851723   9.375548    9.182731    3.058218    3.459215
  5.4962754   4.018234    8.839286    8.8314085   9.505224    2.3476098
  8.823815   12.072752    8.601356    7.9471993  11.292472   11.457238
  6.027743    7.071573    7.0377216   2.9373167   7.307343    8.48312
 19.111702    2.9167256   5.7694364   9.05401     7.0445848   5.576129
  2.3362818   8.641402    5.7351575   6.7036204   5.4995666   6.012248
  2.507907    6.9093833   5.001413    4.078641    4.557381    9.426562
  4.4181395   2.9529035   8.198942    2.3040478   2.800392    4.689731
 13.018832    2.6840668   3.003842    6.659847    4.55896     5.778887
 10.350336    5.926081    4.4054236  14.795793    1.2430875   7.5658617
  0.87765217  7.2287545   6.850049    4.951525    8.017592    7.024554
  5.953328    7.7224193   6.606814    6.7617393  11.722911    6.332674
  3.7297676   3.657504   12.947633   12.445496    3.9683583   4.0251756
  5.04758     7.196997    4.534546    5.3412795   9.387548   11.5132
  4.3527474   3.6554115   3.1876783   3.529498    8.873071    4.952435
  6.8690434   2.9971051   2.8173728   6.308332    1.3162148   7.098621
  5.2898364   3.9596841   1.9113549   1.9215572   7.324026    9.437474
  7.8335223   5.1925454   4.6072373   4.2747693   6.805999    3.2848585
  8.721183    7.721843    5.8097773   4.0527506   4.101082    5.9977326
  4.32023     3.37436     6.048842    5.917717    2.8842163   7.8604107
  3.3678865   4.4946995   3.9107037   4.710452    3.8326082   3.741211
  6.5966315   4.096897    7.036597    6.277786    7.1407843   5.955763
  7.9515915   7.523023    5.278676    2.6221385   3.9909554   8.753346
  6.462921    1.9766583   5.333998    3.7846956   2.9319718   6.691585
  6.369771    2.7630775   2.607686    3.20876     3.6895769   5.622818
  5.5627694   3.9689782   2.0877512   5.8020372   7.9343553   2.7381692
  2.6250763   3.7928612   1.1395389   5.831917    5.012505    4.5884542
  4.925548    5.024918    3.991863    6.383615    3.1053205   6.64485
  9.6873665   5.221062    5.8723288  13.390061   ]]
shape: (1, 1000)
```

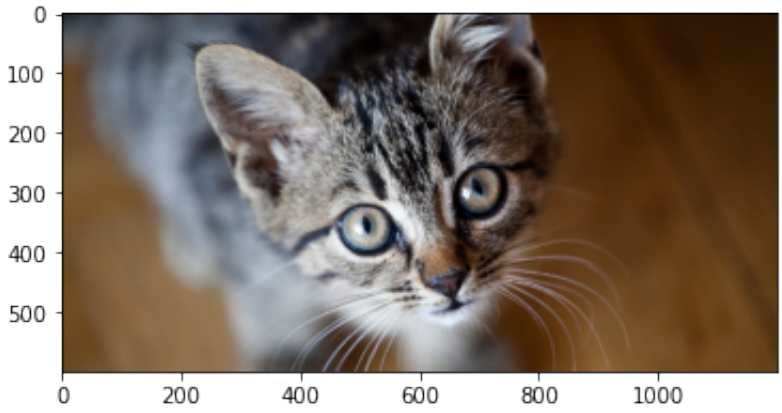
In [31]:

```
import matplotlib.image as mpimg

dog_test_path = join(os.getcwd(),'Chinook-On-White-03.jpg')
cat_test_path = join(os.getcwd(),'Thinking-of-getting-a-cat.png')

img = mpimg.imread(dog_test_path)
imgplot = plt.imshow(img)
plt.show()
```



In [32]:

```
img = mpimg.imread(cat_test_path)
imgplot = plt.imshow(img)
plt.show()
```



In [35]:

```
features_out = get_features(get_image(cat_test_path))

prob = lg.predict_proba(features_out)
pred = lg.predict(features_out)

if pred[0] == 1:
    fpred = 'cat'
else:
    fpred = 'dog'

print("prob: {} and prediction: {}".format(prob,fpred))
```

```
prob: [[7.82527617e-08 9.99999922e-01]] and prediction: cat
```

Repeat all previous steps using MobileNet V2 (https://github.com/KeyKy/mobilenet-mxnet). How the two networks compare?

In [36]:
```
!wget https://github.com/KeyKy/mobilenet-mxnet/raw/master/mobilenet_v2-symbol
!wget https://github.com/KeyKy/mobilenet-mxnet/raw/master/mobilenet_v2-0000.p
```

```
--2020-11-07 12:04:12--  https://github.com/KeyKy/mobilenet-mxnet/raw/master/m
obilenet_v2-symbol.json
Resolving github.com (github.com)... 140.82.121.4
Connecting to github.com (github.com)|140.82.121.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/KeyKy/mobilenet-mxnet/master/mobil
enet_v2-symbol.json [following]
--2020-11-07 12:04:13--  https://raw.githubusercontent.com/KeyKy/mobilenet-mxn
et/master/mobilenet_v2-symbol.json
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.84.
133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.84
.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 108557 (106K) [text/plain]
Saving to: ‘mobilenet_v2-symbol.json’

mobilenet_v2-symbol 100%[===================>] 106.01K  --.-KB/s    in 0.05s

2020-11-07 12:04:13 (1.99 MB/s) - ‘mobilenet_v2-symbol.json’ saved [108557/108
557]

--2020-11-07 12:04:14--  https://github.com/KeyKy/mobilenet-mxnet/raw/master/m
obilenet_v2-0000.params
Resolving github.com (github.com)... 140.82.121.4
Connecting to github.com (github.com)|140.82.121.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/KeyKy/mobilenet-mxnet/master/mobil
enet_v2-0000.params [following]
--2020-11-07 12:04:14--  https://raw.githubusercontent.com/KeyKy/mobilenet-mxn
et/master/mobilenet_v2-0000.params
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.84.
133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.84
.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 14178924 (14M) [application/octet-stream]
Saving to: ‘mobilenet_v2-0000.params’

mobilenet_v2-0000.p 100%[===================>]  13.52M  34.5MB/s    in 0.4s

2020-11-07 12:04:16 (34.5 MB/s) - ‘mobilenet_v2-0000.params’ saved [14178924/1
4178924]
```

In [102…
```
# define a simple data batch
from collections import namedtuple
Batch = namedtuple('Batch', ['data'])
```

In [103…
```
context=mx.cpu()
```

In [104…
```
def get_image(url, show=False):
    if url.startswith('http'):
        # download and show the image
        fname = mx.test_utils.download(url)
    else:
        fname = url
    img = cv2.cvtColor(cv2.imread(fname), cv2.COLOR_BGR2RGB)
    if img is None:
         return None
    if show:
         plt.imshow(img)
         plt.axis('off')
    # convert into format (batch, RGB, width, height)
    img = cv2.resize(img, (224, 224))
    img = np.swapaxes(img, 0, 2)
    img = np.swapaxes(img, 1, 2)
    img = img[np.newaxis, :]
    return img

def predict(url):
    img = get_image(url, show=True)
    # compute the predict probabilities
    mod.forward(Batch([mx.nd.array(img)]))
    prob = mod.get_outputs()[0].asnumpy()
    # print the top-5
    prob = np.squeeze(prob)
    a = np.argsort(prob)[::-1]
    for i in a[0:5]:
        print('probability=%f, class=%s' %(prob[i], labels[i]))
```

In [105…
```
sym, arg_params, aux_params = mx.model.load_checkpoint('mobilenet_v2', 0)
```

In [106…
```
# list the last 10 layers
all_layers = sym.get_internals()
all_layers.list_outputs()[-5:]
```

Out[106… `['fc7_bias', 'fc7_output', 'fc7_flatten_output', 'prob_label', 'prob_output']`

In [107…

```
fe_sym = all_layers['fc7_flatten_output']
fe_mod = mx.mod.Module(symbol=fe_sym, context=mx.cpu(), label_names=None)
fe_mod.bind(for_training=False, data_shapes=[('data', (1,3,224,224))])
fe_mod.set_params(arg_params, aux_params)
```

In [108…

```
def get_features(img):
    fe_mod.forward(Batch([mx.nd.array(img)]))
    features = fe_mod.get_outputs()[0].asnumpy()
    return features
```

In [109…

```
img = get_image('https://icatcare.org/app/uploads/2018/07/Thinking-of-getting
features = get_features(img)
print("{}\n shape: {}".format(features,features.shape))
```

```
[[-8.73116531e+01  2.83228577e+02 -3.59607025e+02 -4.03708984e+02
  -6.46255798e+02  1.48531494e+02 -3.47858490e+02  6.66299515e+01
   2.78768982e+02 -1.49736664e+02  8.38041992e+01 -1.37439499e+02
  -4.69580017e+02 -9.14800415e+01 -2.76609131e+02  1.35187820e+02
   1.80837677e+02 -3.41249298e+02  1.15875607e+01 -5.60398926e+02
  -2.54541214e+02  3.32528320e+02  2.52942486e+01  4.03285645e+02
   2.14639755e+02 -4.70253792e+01 -4.01036255e+02 -2.38460938e+02
  -5.36336288e+01 -6.14872131e+02 -3.40083069e+02 -2.73898651e+02
  -3.69430878e+02 -6.40303528e+02 -2.56124176e+02 -4.16391968e+02
  -1.71422821e+02 -2.47153595e+02 -1.99477921e+02 -1.43629562e+02
  -3.21645264e+02 -4.46027039e+02 -2.44809418e+02  1.03301361e+02
  -1.46011917e+02 -4.25061462e+02 -3.44877625e+02 -5.04795135e+02
  -1.92245743e+02 -4.46315369e+02 -7.29045776e+02  6.43142151e+02
  -5.43736694e+02 -3.97714417e+02 -5.26605774e+02 -7.36024475e+02
  -1.60982224e+02 -2.95890625e+02 -7.81803040e+02 -3.25579132e+02
  -8.40681839e+01 -6.43729919e+02  4.14718666e+01 -3.34889435e+02
  -5.07559692e+02 -1.62592194e+02 -1.88286270e+02 -6.47604553e+02
  -1.27652222e+02 -1.61137512e+02  1.22373108e+02 -4.67065430e+02
  -2.27845261e+02  4.93836746e+01 -8.06280289e+01  5.41381775e+02
  -2.29811203e+02 -4.32604919e+02  1.78805542e+02 -2.30924088e+02
   7.84939880e+02 -3.04035858e+02  2.51537018e+02  5.64096863e+02
  -1.44117874e+02 -2.62628815e+02  2.57276184e+02 -1.06053314e+02
   3.47322205e+02 -1.54493103e+02  5.06014801e+02  4.19897217e+02
   3.65872421e+01  3.79745605e+02 -3.07246521e+02 -9.59804459e+01
   3.53265350e+02 -1.74011169e+02  2.35754410e+02  1.05712608e+02
  -6.54890060e+01 -3.04422974e+02 -1.38272079e+02  2.41708221e+02
   1.80800858e+02 -2.11239624e+02 -2.68215027e+01 -2.94282776e+02
  -2.98940186e+02  2.11309464e+02 -7.89026062e+02 -1.89356308e+02
  -1.41223419e+02 -2.24167953e+02 -2.93801178e+02 -3.43777405e+02
  -4.11104340e+02 -1.95378922e+02 -2.99134857e+02 -3.35203796e+02
  -3.44139679e+02 -5.81374756e+02 -2.53917221e+02 -3.83287689e+02
  -2.50095322e+02 -3.55800873e+02  2.17763214e+02 -5.51281616e+02
   1.30699783e+02  2.94620914e+01 -1.50848785e+02 -4.85231018e+02
  -1.26667007e+02 -2.28896347e+02  1.63994827e+02  2.31013756e+01
  -1.94970383e+02  1.25436058e+02 -4.93182648e+02 -8.36208313e+02
```

```
 -3.89129456e+02  3.36298065e+01 -2.52807846e+02 -3.09167664e+02
 -1.47562958e+02 -4.39925537e+02 -1.87984955e+02 -3.16677673e+02
 -9.73731308e+01 -4.17394531e+02  8.80572357e+01 -2.53235886e+02]]
shape: (1, 1000)
```

In [110…

```
from os import listdir
from os.path import isfile, join
import os


mypath = join(os.getcwd(),'train')

cats_imgs = [join(mypath,f) for f in listdir(mypath) if f.startswith('cat')]
dogs_imgs = [join(mypath,f) for f in listdir(mypath) if f.startswith('dog')]
```

In [111…

```
print("cats: {} and dogs: {}".format(len(cats_imgs),len(dogs_imgs)))
```

```
cats: 12500 and dogs: 12500
```

In [112…

```
Nmax = 100
cats_features = [get_features(get_image(img)).ravel() for img in cats_imgs[:N
dogs_features = [get_features(get_image(img)).ravel() for img in dogs_imgs[:N
```

In [113…

```
Y_cats = np.array(Nmax * [1])
Y_dogs = np.array(Nmax * [0])
```

In [114…

```
X_cvd = np.vstack([cats_features,dogs_features])
Y_cvd = np.vstack([Y_cats,Y_dogs]).ravel()
```

In [115…

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_cvd, Y_cvd, random_stat
```

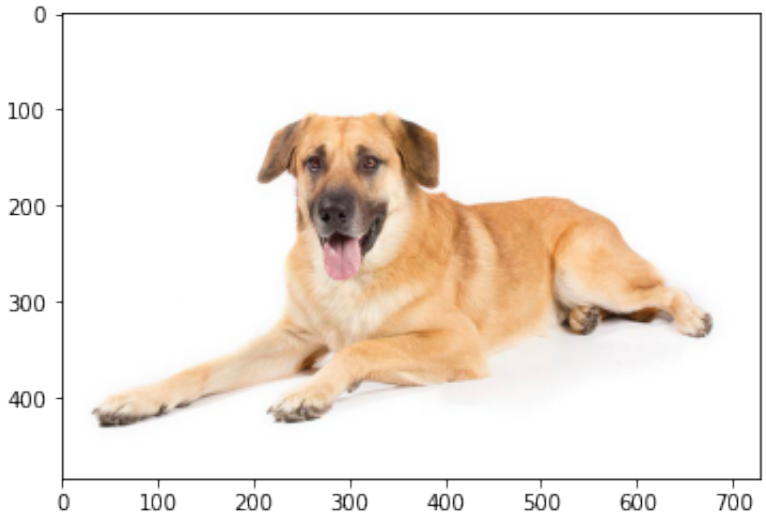
In [116…

```
from sklearn.linear_model import LogisticRegression

lg = LogisticRegression().fit(X_train, y_train)

print("Test set score: {:.2f}".format(lg.score(X_test, y_test)))
```

```
Test set score: 0.54
/opt/conda/lib/python3.7/site-packages/sklearn/linear_model/_logistic.py:940:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regress
ion
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
```

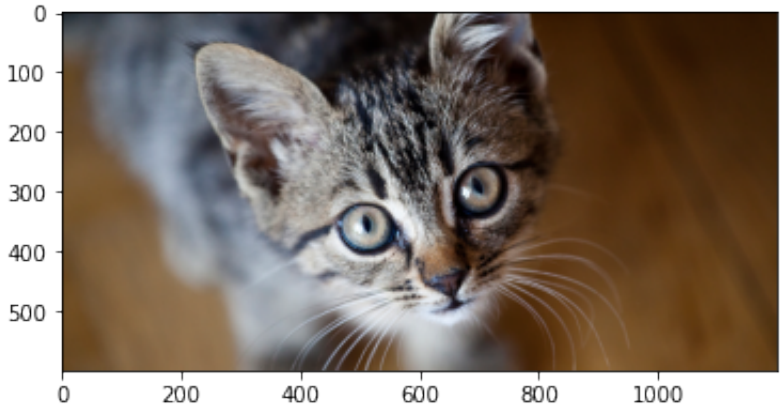
In [117…

```
import matplotlib.image as mpimg

dog_test_path = join(os.getcwd(),'Chinook-On-White-03.jpg')
cat_test_path = join(os.getcwd(),'Thinking-of-getting-a-cat.png')

img = mpimg.imread(dog_test_path)
imgplot = plt.imshow(img)
plt.show()
```



In [118…

```
img = mpimg.imread(cat_test_path)
imgplot = plt.imshow(img)
plt.show()
```



In [119…

```
features_out = get_features(get_image(cat_test_path))

prob = lg.predict_proba(features_out)
pred = lg.predict(features_out)

if pred[0] == 1:
    fpred = 'cat'
else:
    fpred = 'dog'

print("prob: {} and prediction: {}".format(prob,fpred))
```

```
prob: [[0.74842734 0.25157266]] and prediction: dog
```

Different networks apparently produced different results

In [ ]: