In [1]:
```
import cv2                  # working with, mainly resizing, images
import numpy as np          # dealing with arrays
import os                   # dealing with directories
from random import shuffle # mixing up or currently ordered data that m
ight lead our network astray in training.
from tqdm import tqdm       # a nice pretty percentage bar for tasks. Th
anks to viewer Daniel BA1/4hler for this suggestion
import warnings
```

In [2]:
```
TRAIN_DIR = 'C:/Users/Abir Khan/Desktop/traincatdog'
TEST_DIR = 'C:/Users/Abir Khan/Desktop/testcatdog'
IMG_SIZE = 50
LR = 1e-3
MODEL_NAME = 'dogsvscats-{}-{}.model'.format(LR, '2conv-basic') # just
 so we remember which saved model is which, sizes must match
```

In [3]:
```
def label_img(img):
    word_label = img.split('.')[-3]
    # conversion to one-hot array [cat,dog]
    #                            [much cat, no dog]
    if word_label == 'cat': return [1,0]
    #                             [no cat, very doggo]
    elif word_label == 'dog': return [0,1]
```

In [4]:
```
def create_train_data():
    training_data = []
    for img in tqdm(os.listdir(TRAIN_DIR)):
        label = label_img(img)
        path = os.path.join(TRAIN_DIR,img)
        img = cv2.imread(path,cv2.IMREAD_GRAYSCALE)
        img = cv2.resize(img, (IMG_SIZE,IMG_SIZE))
        training_data.append([np.array(img),np.array(label)])
    shuffle(training_data)
    np.save('train_data.npy', training_data)
    return training_data
```

In [5]:
```
def process_test_data():
    testing_data = []
    for img in tqdm(os.listdir(TEST_DIR)):
        path = os.path.join(TEST_DIR,img)
        img_num = img.split('.')[0]
        img = cv2.imread(path,cv2.IMREAD_GRAYSCALE)
        img = cv2.resize(img, (IMG_SIZE,IMG_SIZE))
        testing_data.append([np.array(img), img_num])

    shuffle(testing_data)
    np.save('test_data.npy', testing_data)
    return testing_data
```

In [6]:
```
#train_data = create_train_data()
# If you have already created the dataset:
train_data = np.load('train_data.npy')
```

In [7]:
```
import tflearn
from tflearn.layers.conv import conv_2d, max_pool_2d
from tflearn.layers.core import input_data, dropout, fully_connected
from tflearn.layers.estimator import regression

convnet = input_data(shape=[None, IMG_SIZE, IMG_SIZE, 1], name='input')

convnet = conv_2d(convnet, 32, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)

convnet = conv_2d(convnet, 64, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)

convnet = fully_connected(convnet, 1024, activation='relu')
convnet = dropout(convnet, 0.8)

convnet = fully_connected(convnet, 2, activation='softmax')
convnet = regression(convnet, optimizer='adam', learning_rate=LR, loss=
'categorical_crossentropy', name='targets')
```

```
model = tflearn.DNN(convnet, tensorboard_dir='log')
```

```
curses is not supported on this machine (please install/reinstall curse
s for an optimal experience)
WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\helpers\summarizer.py:9: The name tf.summary.merge is deprecat
ed. Please use tf.compat.v1.summary.merge instead.

WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\helpers\trainer.py:25: The name tf.summary.FileWriter is depre
cated. Please use tf.compat.v1.summary.FileWriter instead.

WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\collections.py:13: The name tf.GraphKeys is deprecated. Please
use tf.compat.v1.GraphKeys instead.

WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\config.py:123: The name tf.get_collection is deprecated. Pleas
e use tf.compat.v1.get_collection instead.

WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\config.py:129: The name tf.add_to_collection is deprecated. Pl
ease use tf.compat.v1.add_to_collection instead.

WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\config.py:131: The name tf.assign is deprecated. Please use t
f.compat.v1.assign instead.

WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\layers\core.py:81: The name tf.placeholder is deprecated. Plea
se use tf.compat.v1.placeholder instead.

WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\layers\conv.py:73: The name tf.variable_scope is deprecated. P
lease use tf.compat.v1.variable_scope instead.

WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\initializations.py:119: calling UniformUnitScaling.__init__ (f
rom tensorflow.python.ops.init_ops) with dtype is deprecated and will b
```

```
e removed in a future version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing it
to the constructor
WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tensorflow_core\python\util\deprecation.py:507: UniformUnitScaling.__i
nit__ (from tensorflow.python.ops.init_ops) is deprecated and will be r
emoved in a future version.
Instructions for updating:
Use tf.initializers.variance_scaling instead with distribution=uniform
to get equivalent behavior.
WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\layers\conv.py:552: The name tf.nn.max_pool is deprecated. Ple
ase use tf.nn.max_pool2d instead.

WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\initializations.py:174: calling TruncatedNormal.__init__ (from
tensorflow.python.ops.init_ops) with dtype is deprecated and will be re
moved in a future version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing it
to the constructor
WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\layers\core.py:239: calling dropout (from tensorflow.python.op
s.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`.
WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\optimizers.py:238: The name tf.train.AdamOptimizer is deprecat
ed. Please use tf.compat.v1.train.AdamOptimizer instead.

WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\objectives.py:66: calling reduce_sum_v1 (from tensorflow.pytho
n.ops.math_ops) with keep_dims is deprecated and will be removed in a f
uture version.
Instructions for updating:
keep_dims is deprecated, use keepdims instead
```

```
WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\objectives.py:70: The name tf.log is deprecated. Please use t
f.math.log instead.

WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\layers\estimator.py:189: The name tf.trainable_variables is de
precated. Please use tf.compat.v1.trainable_variables instead.

WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\helpers\trainer.py:571: The name tf.get_default_graph is depre
cated. Please use tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\helpers\trainer.py:115: The name tf.Session is deprecated. Ple
ase use tf.compat.v1.Session instead.

WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\summaries.py:46: The name tf.summary.scalar is deprecated. Ple
ase use tf.compat.v1.summary.scalar instead.

WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tensorflow_core\python\ops\math_grad.py:1424: where (from tensorflow.p
ython.ops.array_ops) is deprecated and will be removed in a future vers
ion.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\helpers\trainer.py:134: The name tf.train.Saver is deprecated.
Please use tf.compat.v1.train.Saver instead.

WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\helpers\trainer.py:164: The name tf.global_variables_initializ
er is deprecated. Please use tf.compat.v1.global_variables_initializer
instead.

WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\helpers\trainer.py:165: The name tf.local_variables_initialize
r is deprecated. Please use tf.compat.v1.local_variables_initializer in
stead.
```

```
WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\helpers\trainer.py:166: The name tf.variables_initializer is d
eprecated. Please use tf.compat.v1.variables_initializer instead.

WARNING:tensorflow:From C:\Users\Abir Khan\Anaconda3\lib\site-packages
\tflearn\helpers\trainer.py:167: The name tf.get_collection_ref is depr
ecated. Please use tf.compat.v1.get_collection_ref instead.
```

if os.path.exists('{}.meta'.format(MODEL_NAME)): model.load(MODEL_NAME) print('model loaded!')

In [8]:
```
train = train_data[:-500]
test = train_data[-500:]
```

In [9]:
```
X = np.array([i[0] for i in train]).reshape(-1,IMG_SIZE,IMG_SIZE,1)
Y = [i[1] for i in train]
```

In [10]:
```
test_x = np.array([i[0] for i in test]).reshape(-1,IMG_SIZE,IMG_SIZE,1)
test_y = [i[1] for i in test]
```

In [11]:
```
model.fit({'input': X}, {'targets': Y}, n_epoch=3, validation_set=({'in
put': test_x}, {'targets': test_y}),
    snapshot_step=500, show_metric=True, run_id=MODEL_NAME)
```

```
Training Step: 1148  | total loss: 11.84704 | time: 60.278s
| Adam | epoch: 003 | loss: 11.84704 - acc: 0.4855 -- iter: 24448/24500
Training Step: 1149  | total loss: 11.84960 | time: 61.438s
| Adam | epoch: 003 | loss: 11.84960 - acc: 0.4854 | val_loss: 11.78924
- val_acc: 0.4880 -- iter: 24500/24500
--
```

In [12]:
```
import tensorflow as tf
tf.reset_default_graph()
convnet = input_data(shape=[None, IMG_SIZE, IMG_SIZE, 1], name='input')
```

```
convnet = conv_2d(convnet, 32, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)

convnet = conv_2d(convnet, 64, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)

convnet = conv_2d(convnet, 128, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)

convnet = conv_2d(convnet, 64, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)

convnet = conv_2d(convnet, 32, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)

convnet = fully_connected(convnet, 1024, activation='relu')
convnet = dropout(convnet, 0.8)

convnet = fully_connected(convnet, 2, activation='softmax')
convnet = regression(convnet, optimizer='adam', learning_rate=LR, loss=
'categorical_crossentropy', name='targets')

model = tflearn.DNN(convnet, tensorboard_dir='log')



if os.path.exists('{}.meta'.format(MODEL_NAME)):
    model.load(MODEL_NAME)
    print('model loaded!')

train = train_data[:-500]
test = train_data[-500:]

X = np.array([i[0] for i in train]).reshape(-1,IMG_SIZE,IMG_SIZE,1)
Y = [i[1] for i in train]

test_x = np.array([i[0] for i in test]).reshape(-1,IMG_SIZE,IMG_SIZE,1)
test_y = [i[1] for i in test]
```

```
model.fit({'input': X}, {'targets': Y}, n_epoch=3, validation_set=({'in
put': test_x}, {'targets': test_y}),
    snapshot_step=500, show_metric=True, run_id=MODEL_NAME)
```

```
Training Step: 4978  | total loss: 0.26317 | time: 65.448s
| Adam | epoch: 003 | loss: 0.26317 - acc: 0.8948 -- iter: 24448/24500
Training Step: 4979  | total loss: 0.25763 | time: 66.618s
| Adam | epoch: 003 | loss: 0.25763 - acc: 0.8960 | val_loss: 0.62711 -
val_acc: 0.7680 -- iter: 24500/24500
--
```

In [13]:
```
import tensorflow as tf
tf.reset_default_graph()
```

In [14]:
```
convnet = input_data(shape=[None, IMG_SIZE, IMG_SIZE, 1], name='input')

convnet = conv_2d(convnet, 32, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)

convnet = conv_2d(convnet, 64, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)

convnet = conv_2d(convnet, 128, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)

convnet = conv_2d(convnet, 64, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)

convnet = conv_2d(convnet, 32, 5, activation='relu')
convnet = max_pool_2d(convnet, 5)

convnet = fully_connected(convnet, 1024, activation='relu')
convnet = dropout(convnet, 0.8)

convnet = fully_connected(convnet, 2, activation='softmax')
convnet = regression(convnet, optimizer='adam', learning_rate=LR, loss=
'categorical_crossentropy', name='targets')
```

```
model = tflearn.DNN(convnet, tensorboard_dir='log')



if os.path.exists('C:/Users/H/Desktop/KaggleDogsvsCats/{}.meta'.format(
MODEL_NAME)):
    model.load(MODEL_NAME)
    print('model loaded!')

train = train_data[:-500]
test = train_data[-500:]

X = np.array([i[0] for i in train]).reshape(-1,IMG_SIZE,IMG_SIZE,1)
Y = [i[1] for i in train]

test_x = np.array([i[0] for i in test]).reshape(-1,IMG_SIZE,IMG_SIZE,1)
test_y = [i[1] for i in test]

model.fit({'input': X}, {'targets': Y}, n_epoch=10, validation_set=({'i
nput': test_x}, {'targets': test_y}),
    snapshot_step=500, show_metric=True, run_id=MODEL_NAME)

model.save(MODEL_NAME)
```

```
Training Step: 3829  | total loss: 0.35186 | time: 62.576s
| Adam | epoch: 010 | loss: 0.35186 - acc: 0.8478 -- iter: 24448/24500
Training Step: 3830  | total loss: 0.35344 | time: 63.740s
| Adam | epoch: 010 | loss: 0.35344 - acc: 0.8458 | val_loss: 0.62508 -
val_acc: 0.7400 -- iter: 24500/24500
--
INFO:tensorflow:C:\Users\Abir Khan\Desktop\data\dogsvscats-0.001-2conv-
basic.model is not in all_model_checkpoint_paths. Manually adding it.
```

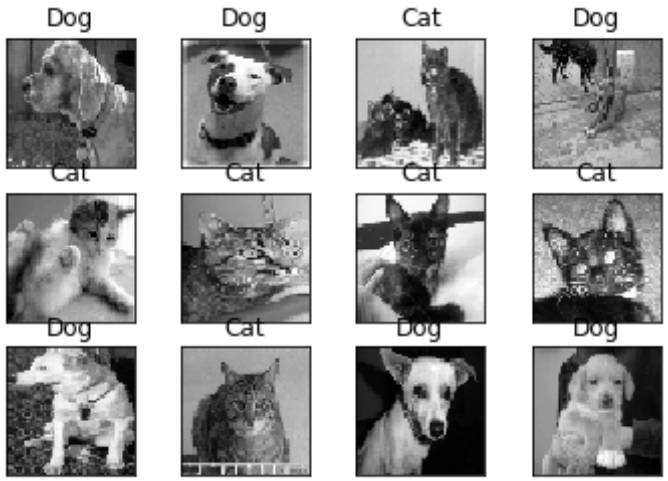
In [15]:
```
!pip install numpy==1.16.2
import numpy as np
print(np.__version__)
```

```
Requirement already satisfied: numpy==1.16.2 in c:\users\abir khan\anac
onda3\lib\site-packages (1.16.2)
```

```
1.16.2
```

In [20]:
```
import matplotlib.pyplot as plt

# if you need to create the data:
test_data = process_test_data()
# if you already have some saved:
#test_data = np.load('test_data.npy')


fig=plt.figure()

for num,data in enumerate(test_data[:12]):
    # cat: [1,0]
    # dog: [0,1]

    img_num = data[1]
    img_data = data[0]

    y = fig.add_subplot(3,4,num+1)
    orig = img_data
    data = img_data.reshape(IMG_SIZE,IMG_SIZE,1)

    #model_out = model.predict([data])[0]
    model_out = model.predict([data])[0]

    if np.argmax(model_out) == 1: str_label='Dog'
    else: str_label='Cat'

    y.imshow(orig,cmap='gray')
    plt.title(str_label)
    y.axes.get_xaxis().set_visible(False)
    y.axes.get_yaxis().set_visible(False)
plt.show()
```

```
100%|██████████| 18/18 [00:00<00:00, 620.25it/s]
```



In [ ]:

In [ ]: