In [1]:

```
# Importing the Keras libraries and packages

from keras.models import Sequential

from keras.layers import Conv2D

from keras.layers import MaxPooling2D

from keras.layers import Flatten

from keras.layers import Dense,Dropout
```

```
/home/user/anaconda3/lib/python3.6/site-packages/h5py/__init__.py:36: FutureWarning: Conversion
of the second argument of issubdtype from `float` to `np.floating` is deprecated. In future, it
will be treated as `np.float64 == np.dtype(float).type`.
  from ._conv import register_converters as _register_converters
Using TensorFlow backend.
```

In [8]:

```
classifier = Sequential()
#Add Convolution layer
classifier.add(Conv2D(32, (3, 3),padding='same', input_shape = (64, 64, 3), activation = 'relu'))
#Add Max Pooling layer
classifier.add(MaxPooling2D(pool_size = (2, 2)))

#Add Convolution layer
classifier.add(Conv2D(64, (3, 3),padding='same', activation = 'relu'))
#Add Max Pooling layer
classifier.add(MaxPooling2D(pool_size = (2, 2)))


#Add Convolution layer
classifier.add(Conv2D(128, (3, 3),padding='same', activation = 'relu'))
#Add Max Pooling layer
classifier.add(MaxPooling2D(pool_size = (2, 2)))


#Add Convolution layer
classifier.add(Conv2D(64, (3, 3), padding='same', activation = 'relu'))
#Add Max Pooling layer
classifier.add(MaxPooling2D(pool_size = (2, 2)))

#Add Convolution layer--
classifier.add(Conv2D(32, (3, 3), padding='same', activation = 'relu'))
#Add Max Pooling layer
classifier.add(MaxPooling2D(pool_size = (2, 2)))



#Perform Flattening
classifier.add(Flatten())
print(classifier.layers[10].output_shape)
#Add Fully Connected Network
classifier.add(Dense(units = 32, activation = 'relu'))
# classifier.add(Dropout(0.8))

#Final Output layer
classifier.add(Dense(units = 1, activation = 'sigmoid'))
#compile the classsifier
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
```

```
(None, 128)
```

In [9]:

```
from keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(rescale = 1./255,shear_range = 0.2,zoom_range = 0.2,horizontal_flip = True)
test_datagen = ImageDataGenerator(rescale = 1./255)
training_set = train_datagen.flow_from_directory('training_set',target_size = (64, 64),batch_size = 32,class_mode = 'binary')
validation_set = train_datagen.flow_from_directory('validation_set',target_size = (64, 64),batch_size = 32,class_mode = 'binary')
test_set = test_datagen.flow_from_directory('test_set',target_size = (64, 64),batch_size = 32,class_mode = 'binary')
```

```
Found 1600 images belonging to 2 classes.
Found 400 images belonging to 2 classes.
Found 2023 images belonging to 2 classes.
```

In [10]:

```
classifier.fit_generator(training_set,steps_per_epoch = 1600,epochs = 10,validation_data = validation_set,validation_steps = 400,max_queue_size=7,workers=3)
```

```
Epoch 1/10
1600/1600 [==============================] - 651s 407ms/step - loss: 0.4601 - acc: 0.7671 - val_loss: 0.6186 - val_acc: 0.7609
Epoch 2/10
1600/1600 [==============================] - 619s 387ms/step - loss: 0.1278 - acc: 0.9491 - val_loss: 0.9159 - val_acc: 0.7611
Epoch 3/10
1600/1600 [==============================] - 639s 399ms/step - loss: 0.0599 - acc: 0.9786 - val_loss: 1.0098 - val_acc: 0.7803
Epoch 4/10
1600/1600 [==============================] - 633s 396ms/step - loss: 0.0421 - acc: 0.9847 - val_loss: 1.1338 - val_acc: 0.7976
Epoch 5/10
1600/1600 [==============================] - 647s 404ms/step - loss: 0.0365 - acc: 0.9870 - val_loss: 1.1252 - val_acc: 0.7837
Epoch 6/10
1600/1600 [==============================] - 649s 406ms/step - loss: 0.0294 - acc: 0.9896 - val_loss: 1.2232 - val_acc: 0.7843
Epoch 7/10
1600/1600 [==============================] - 552s 345ms/step - loss: 0.0293 - acc: 0.9895 - val_loss: 1.2512 - val_acc: 0.7784
Epoch 8/10
1600/1600 [==============================] - 320s 200ms/step - loss: 0.0227 - acc: 0.9922 - val_loss: 1.4969 - val_acc: 0.7717
Epoch 9/10
1600/1600 [==============================] - 316s 198ms/step - loss: 0.0219 - acc: 0.9925 - val_loss: 1.2861 - val_acc: 0.7856
Epoch 10/10
1600/1600 [==============================] - 329s 206ms/step - loss: 0.0211 - acc: 0.9927 - val_loss: 1.2015 - val_acc: 0.7884
```

Out[10]:

```
<keras.callbacks.History at 0x7f49f64d5f98>
```

In [12]:

```
import numpy as np
from keras.preprocessing import image
import os
import glob
data_path = os.path.join('./test_set/dogs/','*jpg')
files = glob.glob(data_path)
dog,cat,total=0,0,0
for f in files:
    test_image = image.load_img(f, target_size = (64, 64))
    test_image = image.img_to_array(test_image)
    test_image = np.expand_dims(test_image, axis = 0)
    result = classifier.predict(test_image)
    training_set.class_indices
    if result[0][0] == 1:
        dog+=1
    else:
        cat+=1
    total+=1
print("Total Dogs: ",total)
print("Dogs : ",dog)
print("Cats : ",cat)
print('----------------------')

data_path = os.path.join('./test_set/cats/','*jpg')
files = glob.glob(data_path)
dog,cat,total=0,0,0
for f in files:
    test_image = image.load_img(f, target_size = (64, 64))
    test_image = image.img_to_array(test_image)
    test_image = np.expand_dims(test_image, axis = 0)
    result = classifier.predict(test_image)
    training_set.class_indices
    if result[0][0] == 1:
        dog+=1
    else:
        cat+=1
    total+=1
print("Total Cats: ",total)
print("Dogs : ",dog)
print("Cats : ",cat)
print('----------------------')
```

```
Total Dogs:  1012
Dogs :  863
Cats :  149
----------------------
Total Cats:  1011
Dogs :  360
Cats :  651
----------------------
```