Question 1

In [1]:
```
pip install opencv-python
```

```
Note: you may need to restart the kernel to use updated packages.

'C:\Users\Nupur' is not recognized as an internal or external command,
operable program or batch file.
```

In [3]:
```
import cv2
import numpy as np
import os
from random import shuffle
from tqdm import tqdm
import pandas as pd
```

In [4]:
```
TRAIN_DIR = 'E:\dogs-vs-cats-redux-kernels-edition\train'
TEST_DIR = 'E:\dogs-vs-cats-redux-kernels-edition\test'
IMG_SIZE = 50
LR = 1e-3
MODEL_NAME = 'dogsvscats-{}-{}.model'.format(LR, '2conv-basic')
```

In [5]:
```
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 [6]:
```
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 [7]:
```
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
```

50x50x3 (50 wide, 50 high, 3 color channels) or third dimension of an activation volume, not to the depth of a full Neural Network, which can refer to the total number of layers in a network

Question 4

How to Prevent Overfitting

1. Cross-validation
Cross-validation is a powerful preventative measure against overfitting.
Use your initial training data to generate multiple mini train-test splits. Use these splits to tune your model.
Cross-validation allows you to tune hyperparameters with only your original training set. This allows you to keep your
test set as a truly unseen dataset for selecting your final model.

2. Train with more data
Training with more data can help algorithms detect the signal better.
If we just add more noisy data, this technique won't help. That's why you should always
ensure your data is clean and relevant.

3. Remove features
We can manually improve their generalizability by removing irrelevant input features.

4. Early stopping
Early stopping refers stopping the training process before the learner passes that point.
Today, this technique is mostly used in deep learning while other techniques (e.g. regularization) are preferred for
classical machine learning.

5. Regularization
The method will depend on the type of learner you're using. For example, you could prune a decision tree,
use dropout on a neural network, or add a penalty parameter to the cost function in regression.

6. Ensembling
Ensembles are machine learning methods for combining predictions from multiple separate models.

There are multiple causes for overfitting
a) Too complex model
b) Data has noise i.e. like there are outliers and errors in data
c) Size of data used for training may not be enough