Residual Networks

Welcome to the second assignment of this week! You will learn how to build very deep convolutional networks, using Residual Networks (ResNets). In theory, very deep networks can represent very complex functions; but in practice, they are hard to train. Residual Networks, introduced by He et al. (https://arxiv.org/pdf/1512.03385.pdf), allow you to train much deeper networks than were previously practically feasible.

In this assignment, you will:

- Implement the basic building blocks of ResNets.
- Put together these building blocks to implement and train a state-of-the-art neural network for image classification.

Updates

If you were working on the notebook before this update...

- The current notebook is version "2a".
- You can find your original work saved in the notebook with the previous version name ("v2")
- To view the file directory, go to the menu "File->Open", and this will open a new tab that shows the file directory.

List of updates

- For testing on an image, replaced `preprocess_input(x)` with `x=x/255.0` to normalize the input image in the same way that the model's training data was normalized.
- Refers to "shallower" layers as those layers closer to the input, and "deeper" layers as those closer to the output (Using "shallower" layers instead of "lower" or "earlier").
- Added/updated instructions.

This assignment will be done in Keras.

Before jumping into the problem, let's run the cell below to load the required packages.

In [1]:
```
import numpy as np
from keras import layers
from keras.layers import Input, Add, Dense, Activation, ZeroPadding2D, BatchNormalization, Flatten, Conv2D, AveragePooling2
from keras.models import Model, load_model
from keras.preprocessing import image
```