Transfer_Learning_Lab

April 30, 2020

1 Lab: Transfer Learning

Welcome to the lab on Transfer Learning! Here, you'll get a chance to try out training a network with ImageNet pre-trained weights as a base, but with additional network layers of your own added on. You'll also get to see the difference between using frozen weights and training on all layers.

1.0.1 GPU usage

In our previous examples in this lesson, we've avoided using GPU, but this time around you'll have the option to enable it. You do not need it on to begin with, but make sure anytime you switch from non-GPU to GPU, or vice versa, that you save your notebook! If not, you'll likely be reverted to the previous checkpoint.

We also suggest only using the GPU when performing the (mostly minor) training below - you'll want to conserve GPU hours for your Behavioral Cloning project coming up next!

In [1]:
```
# Set a couple flags for training - you can ignore these for now
freeze_flag = True  # `True` to freeze layers, `False` for full training
weights_flag = 'imagenet' # 'imagenet' or None
preprocess_flag = True # Should be true for ImageNet pre-trained typically

# Loads in InceptionV3
from keras.applications.inception_v3 import InceptionV3

# We can use smaller than the default 299x299x3 input for InceptionV3
# which will speed up training. Keras v2.0.9 supports down to 139x139x3
input_size = 139

# Using Inception with ImageNet pre-trained weights
inception = InceptionV3(weights=weights_flag, include_top=False,
                        input_shape=(input_size,input_size,3))
```

```
Using TensorFlow backend.


Downloading data from https://github.com/fchollet/deep-learning-models/releases/download/v0.5/in
87916544/87910968 [==============================] - 1s 0us/step
```

We'll use Inception V3 for this lab, although you can use the same techniques with any of the models in Keras Applications. Do note that certain models are only available in certain versions of Keras; this workspace uses Keras v2.0.9, for which you can see the available models here.

In the above, we've set Inception to use an `input_shape` of 139x139x3 instead of the default 299x299x3. This will help us to speed up our training a bit later (and we'll actually be upsampling from smaller images, so we aren't losing data here). In order to do so, we also must set `include_top` to `False`, which means the final fully-connected layer with 1,000 nodes for each ImageNet class is dropped, as well as a Global Average Pooling layer.

1.0.2 Pre-trained with frozen weights

To start, we'll see how an ImageNet pre-trained model with all weights frozen in the InceptionV3 model performs. We will also drop the end layer and append new layers onto it, although you could do this in different ways (not drop the end and add new layers, drop more layers than we will here, etc.).

You can freeze layers by setting `layer.trainable` to False for a given `layer`. Within a `model`, you can get the list of layers with `model.layers`.

```
In [2]: if freeze_flag == True:
            ## TODO: Iterate through the layers of the Inception model
            ##       loaded above and set all of them to have trainable = False
            for layer in inception.layers:
                layer.trainable = False
```

1.0.3 Dropping layers

You can drop layers from a model with `model.layers.pop()`. Before you do this, you should check out what the actual layers of the model are with Keras's `.summary()` function.

```
In [3]: ## TODO: Use the model summary function to see all layers in the
        ##       loaded Inception model
        inception.summary()
```

```
____________________________________________________________________________________________________
Layer (type)                     Output Shape          Param #     Connected to
====================================================================================================
input_1 (InputLayer)             (None, 139, 139, 3)   0
____________________________________________________________________________________________________
conv2d_1 (Conv2D)                (None, 69, 69, 32)    864         input_1[0][0]
____________________________________________________________________________________________________
batch_normalization_1 (BatchNor (None, 69, 69, 32)    96          conv2d_1[0][0]
____________________________________________________________________________________________________
activation_1 (Activation)        (None, 69, 69, 32)    0           batch_normalization_1[0][0]
____________________________________________________________________________________________________
conv2d_2 (Conv2D)                (None, 67, 67, 32)    9216        activation_1[0][0]
____________________________________________________________________________________________________
batch_normalization_2 (BatchNor (None, 67, 67, 32)    96          conv2d_2[0][0]
____________________________________________________________________________________________________
activation_2 (Activation)        (None, 67, 67, 32)    0           batch_normalization_2[0][0]
```

```
conv2d_3 (Conv2D)               (None, 67, 67, 64)   18432       activation_2[0][0]
batch_normalization_3 (BatchNor (None, 67, 67, 64)   192         conv2d_3[0][0]
activation_3 (Activation)       (None, 67, 67, 64)   0           batch_normalization_3[0][0]
max_pooling2d_1 (MaxPooling2D)  (None, 33, 33, 64)   0           activation_3[0][0]
conv2d_4 (Conv2D)               (None, 33, 33, 80)   5120        max_pooling2d_1[0][0]
batch_normalization_4 (BatchNor (None, 33, 33, 80)   240         conv2d_4[0][0]
activation_4 (Activation)       (None, 33, 33, 80)   0           batch_normalization_4[0][0]
conv2d_5 (Conv2D)               (None, 31, 31, 192)  138240      activation_4[0][0]
batch_normalization_5 (BatchNor (None, 31, 31, 192)  576         conv2d_5[0][0]
activation_5 (Activation)       (None, 31, 31, 192)  0           batch_normalization_5[0][0]
max_pooling2d_2 (MaxPooling2D)  (None, 15, 15, 192)  0           activation_5[0][0]
conv2d_9 (Conv2D)               (None, 15, 15, 64)   12288       max_pooling2d_2[0][0]
batch_normalization_9 (BatchNor (None, 15, 15, 64)   192         conv2d_9[0][0]
activation_9 (Activation)       (None, 15, 15, 64)   0           batch_normalization_9[0][0]
conv2d_7 (Conv2D)               (None, 15, 15, 48)   9216        max_pooling2d_2[0][0]
conv2d_10 (Conv2D)              (None, 15, 15, 96)   55296       activation_9[0][0]
batch_normalization_7 (BatchNor (None, 15, 15, 48)   144         conv2d_7[0][0]
batch_normalization_10 (BatchNo (None, 15, 15, 96)   288         conv2d_10[0][0]
activation_7 (Activation)       (None, 15, 15, 48)   0           batch_normalization_7[0][0]
activation_10 (Activation)      (None, 15, 15, 96)   0           batch_normalization_10[0][0]
average_pooling2d_1 (AveragePoo (None, 15, 15, 192)  0           max_pooling2d_2[0][0]
conv2d_6 (Conv2D)               (None, 15, 15, 64)   12288       max_pooling2d_2[0][0]
conv2d_8 (Conv2D)               (None, 15, 15, 64)   76800       activation_7[0][0]
conv2d_11 (Conv2D)              (None, 15, 15, 96)   82944       activation_10[0][0]
```

```
conv2d_12 (Conv2D)              (None, 15, 15, 32)   6144        average_pooling2d_1[0][0]
batch_normalization_6 (BatchNor (None, 15, 15, 64)   192         conv2d_6[0][0]
batch_normalization_8 (BatchNor (None, 15, 15, 64)   192         conv2d_8[0][0]
batch_normalization_11 (BatchNo (None, 15, 15, 96)   288         conv2d_11[0][0]
batch_normalization_12 (BatchNo (None, 15, 15, 32)   96          conv2d_12[0][0]
activation_6 (Activation)       (None, 15, 15, 64)   0           batch_normalization_6[0][0]
activation_8 (Activation)       (None, 15, 15, 64)   0           batch_normalization_8[0][0]
activation_11 (Activation)      (None, 15, 15, 96)   0           batch_normalization_11[0][0]
activation_12 (Activation)      (None, 15, 15, 32)   0           batch_normalization_12[0][0]
mixed0 (Concatenate)            (None, 15, 15, 256)  0           activation_6[0][0]
                                                                 activation_8[0][0]
                                                                 activation_11[0][0]
                                                                 activation_12[0][0]
conv2d_16 (Conv2D)              (None, 15, 15, 64)   16384       mixed0[0][0]
batch_normalization_16 (BatchNo (None, 15, 15, 64)   192         conv2d_16[0][0]
activation_16 (Activation)      (None, 15, 15, 64)   0           batch_normalization_16[0][0]
conv2d_14 (Conv2D)              (None, 15, 15, 48)   12288       mixed0[0][0]
conv2d_17 (Conv2D)              (None, 15, 15, 96)   55296       activation_16[0][0]
batch_normalization_14 (BatchNo (None, 15, 15, 48)   144         conv2d_14[0][0]
batch_normalization_17 (BatchNo (None, 15, 15, 96)   288         conv2d_17[0][0]
activation_14 (Activation)      (None, 15, 15, 48)   0           batch_normalization_14[0][0]
activation_17 (Activation)      (None, 15, 15, 96)   0           batch_normalization_17[0][0]
average_pooling2d_2 (AveragePoo (None, 15, 15, 256)  0           mixed0[0][0]
conv2d_13 (Conv2D)              (None, 15, 15, 64)   16384       mixed0[0][0]
conv2d_15 (Conv2D)              (None, 15, 15, 64)   76800       activation_14[0][0]
```

```
conv2d_18 (Conv2D)              (None, 15, 15, 96)   82944       activation_17[0][0]
--------------------------------------------------------------------------------------------------
conv2d_19 (Conv2D)              (None, 15, 15, 64)   16384       average_pooling2d_2[0][0]
--------------------------------------------------------------------------------------------------
batch_normalization_13 (BatchNo (None, 15, 15, 64)   192         conv2d_13[0][0]
--------------------------------------------------------------------------------------------------
batch_normalization_15 (BatchNo (None, 15, 15, 64)   192         conv2d_15[0][0]
--------------------------------------------------------------------------------------------------
batch_normalization_18 (BatchNo (None, 15, 15, 96)   288         conv2d_18[0][0]
--------------------------------------------------------------------------------------------------
batch_normalization_19 (BatchNo (None, 15, 15, 64)   192         conv2d_19[0][0]
--------------------------------------------------------------------------------------------------
activation_13 (Activation)      (None, 15, 15, 64)   0           batch_normalization_13[0][0]
--------------------------------------------------------------------------------------------------
activation_15 (Activation)      (None, 15, 15, 64)   0           batch_normalization_15[0][0]
--------------------------------------------------------------------------------------------------
activation_18 (Activation)      (None, 15, 15, 96)   0           batch_normalization_18[0][0]
--------------------------------------------------------------------------------------------------
activation_19 (Activation)      (None, 15, 15, 64)   0           batch_normalization_19[0][0]
--------------------------------------------------------------------------------------------------
mixed1 (Concatenate)            (None, 15, 15, 288)  0           activation_13[0][0]
                                                                 activation_15[0][0]
                                                                 activation_18[0][0]
                                                                 activation_19[0][0]
--------------------------------------------------------------------------------------------------
conv2d_23 (Conv2D)              (None, 15, 15, 64)   18432       mixed1[0][0]
--------------------------------------------------------------------------------------------------
batch_normalization_23 (BatchNo (None, 15, 15, 64)   192         conv2d_23[0][0]
--------------------------------------------------------------------------------------------------
activation_23 (Activation)      (None, 15, 15, 64)   0           batch_normalization_23[0][0]
--------------------------------------------------------------------------------------------------
conv2d_21 (Conv2D)              (None, 15, 15, 48)   13824       mixed1[0][0]
--------------------------------------------------------------------------------------------------
conv2d_24 (Conv2D)              (None, 15, 15, 96)   55296       activation_23[0][0]
--------------------------------------------------------------------------------------------------
batch_normalization_21 (BatchNo (None, 15, 15, 48)   144         conv2d_21[0][0]
--------------------------------------------------------------------------------------------------
batch_normalization_24 (BatchNo (None, 15, 15, 96)   288         conv2d_24[0][0]
--------------------------------------------------------------------------------------------------
activation_21 (Activation)      (None, 15, 15, 48)   0           batch_normalization_21[0][0]
--------------------------------------------------------------------------------------------------
activation_24 (Activation)      (None, 15, 15, 96)   0           batch_normalization_24[0][0]
--------------------------------------------------------------------------------------------------
average_pooling2d_3 (AveragePoo (None, 15, 15, 288)  0           mixed1[0][0]
--------------------------------------------------------------------------------------------------
conv2d_20 (Conv2D)              (None, 15, 15, 64)   18432       mixed1[0][0]
--------------------------------------------------------------------------------------------------
conv2d_22 (Conv2D)              (None, 15, 15, 64)   76800       activation_21[0][0]
```

```
conv2d_25 (Conv2D)              (None, 15, 15, 96)   82944       activation_24[0][0]
conv2d_26 (Conv2D)              (None, 15, 15, 64)   18432       average_pooling2d_3[0][0]
batch_normalization_20 (BatchNo (None, 15, 15, 64)   192         conv2d_20[0][0]
batch_normalization_22 (BatchNo (None, 15, 15, 64)   192         conv2d_22[0][0]
batch_normalization_25 (BatchNo (None, 15, 15, 96)   288         conv2d_25[0][0]
batch_normalization_26 (BatchNo (None, 15, 15, 64)   192         conv2d_26[0][0]
activation_20 (Activation)      (None, 15, 15, 64)   0           batch_normalization_20[0][0]
activation_22 (Activation)      (None, 15, 15, 64)   0           batch_normalization_22[0][0]
activation_25 (Activation)      (None, 15, 15, 96)   0           batch_normalization_25[0][0]
activation_26 (Activation)      (None, 15, 15, 64)   0           batch_normalization_26[0][0]
mixed2 (Concatenate)            (None, 15, 15, 288)  0           activation_20[0][0]
                                                                 activation_22[0][0]
                                                                 activation_25[0][0]
                                                                 activation_26[0][0]
conv2d_28 (Conv2D)              (None, 15, 15, 64)   18432       mixed2[0][0]
batch_normalization_28 (BatchNo (None, 15, 15, 64)   192         conv2d_28[0][0]
activation_28 (Activation)      (None, 15, 15, 64)   0           batch_normalization_28[0][0]
conv2d_29 (Conv2D)              (None, 15, 15, 96)   55296       activation_28[0][0]
batch_normalization_29 (BatchNo (None, 15, 15, 96)   288         conv2d_29[0][0]
activation_29 (Activation)      (None, 15, 15, 96)   0           batch_normalization_29[0][0]
conv2d_27 (Conv2D)              (None, 7, 7, 384)    995328      mixed2[0][0]
conv2d_30 (Conv2D)              (None, 7, 7, 96)     82944       activation_29[0][0]
batch_normalization_27 (BatchNo (None, 7, 7, 384)    1152        conv2d_27[0][0]
batch_normalization_30 (BatchNo (None, 7, 7, 96)     288         conv2d_30[0][0]
activation_27 (Activation)      (None, 7, 7, 384)    0           batch_normalization_27[0][0]
```

```
activation_30 (Activation)      (None, 7, 7, 96)     0           batch_normalization_30[0][0]
--------------------------------------------------------------------------------------------
max_pooling2d_3 (MaxPooling2D)  (None, 7, 7, 288)    0           mixed2[0][0]
--------------------------------------------------------------------------------------------
mixed3 (Concatenate)            (None, 7, 7, 768)    0           activation_27[0][0]
                                                                 activation_30[0][0]
                                                                 max_pooling2d_3[0][0]
--------------------------------------------------------------------------------------------
conv2d_35 (Conv2D)              (None, 7, 7, 128)    98304       mixed3[0][0]
--------------------------------------------------------------------------------------------
batch_normalization_35 (BatchNo (None, 7, 7, 128)    384         conv2d_35[0][0]
--------------------------------------------------------------------------------------------
activation_35 (Activation)      (None, 7, 7, 128)    0           batch_normalization_35[0][0]
--------------------------------------------------------------------------------------------
conv2d_36 (Conv2D)              (None, 7, 7, 128)    114688      activation_35[0][0]
--------------------------------------------------------------------------------------------
batch_normalization_36 (BatchNo (None, 7, 7, 128)    384         conv2d_36[0][0]
--------------------------------------------------------------------------------------------
activation_36 (Activation)      (None, 7, 7, 128)    0           batch_normalization_36[0][0]
--------------------------------------------------------------------------------------------
conv2d_32 (Conv2D)              (None, 7, 7, 128)    98304       mixed3[0][0]
--------------------------------------------------------------------------------------------
conv2d_37 (Conv2D)              (None, 7, 7, 128)    114688      activation_36[0][0]
--------------------------------------------------------------------------------------------
batch_normalization_32 (BatchNo (None, 7, 7, 128)    384         conv2d_32[0][0]
--------------------------------------------------------------------------------------------
batch_normalization_37 (BatchNo (None, 7, 7, 128)    384         conv2d_37[0][0]
--------------------------------------------------------------------------------------------
activation_32 (Activation)      (None, 7, 7, 128)    0           batch_normalization_32[0][0]
--------------------------------------------------------------------------------------------
activation_37 (Activation)      (None, 7, 7, 128)    0           batch_normalization_37[0][0]
--------------------------------------------------------------------------------------------
conv2d_33 (Conv2D)              (None, 7, 7, 128)    114688      activation_32[0][0]
--------------------------------------------------------------------------------------------
conv2d_38 (Conv2D)              (None, 7, 7, 128)    114688      activation_37[0][0]
--------------------------------------------------------------------------------------------
batch_normalization_33 (BatchNo (None, 7, 7, 128)    384         conv2d_33[0][0]
--------------------------------------------------------------------------------------------
batch_normalization_38 (BatchNo (None, 7, 7, 128)    384         conv2d_38[0][0]
--------------------------------------------------------------------------------------------
activation_33 (Activation)      (None, 7, 7, 128)    0           batch_normalization_33[0][0]
--------------------------------------------------------------------------------------------
activation_38 (Activation)      (None, 7, 7, 128)    0           batch_normalization_38[0][0]
--------------------------------------------------------------------------------------------
average_pooling2d_4 (AveragePoo (None, 7, 7, 768)    0           mixed3[0][0]
--------------------------------------------------------------------------------------------
conv2d_31 (Conv2D)              (None, 7, 7, 192)    147456      mixed3[0][0]
--------------------------------------------------------------------------------------------
```

```
conv2d_34 (Conv2D)              (None, 7, 7, 192)    172032      activation_33[0][0]
----------------------------------------------------------------------------------------------------
conv2d_39 (Conv2D)              (None, 7, 7, 192)    172032      activation_38[0][0]
----------------------------------------------------------------------------------------------------
conv2d_40 (Conv2D)              (None, 7, 7, 192)    147456      average_pooling2d_4[0][0]
----------------------------------------------------------------------------------------------------
batch_normalization_31 (BatchNo (None, 7, 7, 192)    576         conv2d_31[0][0]
----------------------------------------------------------------------------------------------------
batch_normalization_34 (BatchNo (None, 7, 7, 192)    576         conv2d_34[0][0]
----------------------------------------------------------------------------------------------------
batch_normalization_39 (BatchNo (None, 7, 7, 192)    576         conv2d_39[0][0]
----------------------------------------------------------------------------------------------------
batch_normalization_40 (BatchNo (None, 7, 7, 192)    576         conv2d_40[0][0]
----------------------------------------------------------------------------------------------------
activation_31 (Activation)      (None, 7, 7, 192)    0           batch_normalization_31[0][0]
----------------------------------------------------------------------------------------------------
activation_34 (Activation)      (None, 7, 7, 192)    0           batch_normalization_34[0][0]
----------------------------------------------------------------------------------------------------
activation_39 (Activation)      (None, 7, 7, 192)    0           batch_normalization_39[0][0]
----------------------------------------------------------------------------------------------------
activation_40 (Activation)      (None, 7, 7, 192)    0           batch_normalization_40[0][0]
----------------------------------------------------------------------------------------------------
mixed4 (Concatenate)            (None, 7, 7, 768)    0           activation_31[0][0]
                                                                 activation_34[0][0]
                                                                 activation_39[0][0]
                                                                 activation_40[0][0]
----------------------------------------------------------------------------------------------------
conv2d_45 (Conv2D)              (None, 7, 7, 160)    122880      mixed4[0][0]
----------------------------------------------------------------------------------------------------
batch_normalization_45 (BatchNo (None, 7, 7, 160)    480         conv2d_45[0][0]
----------------------------------------------------------------------------------------------------
activation_45 (Activation)      (None, 7, 7, 160)    0           batch_normalization_45[0][0]
----------------------------------------------------------------------------------------------------
conv2d_46 (Conv2D)              (None, 7, 7, 160)    179200      activation_45[0][0]
----------------------------------------------------------------------------------------------------
batch_normalization_46 (BatchNo (None, 7, 7, 160)    480         conv2d_46[0][0]
----------------------------------------------------------------------------------------------------
activation_46 (Activation)      (None, 7, 7, 160)    0           batch_normalization_46[0][0]
----------------------------------------------------------------------------------------------------
conv2d_42 (Conv2D)              (None, 7, 7, 160)    122880      mixed4[0][0]
----------------------------------------------------------------------------------------------------
conv2d_47 (Conv2D)              (None, 7, 7, 160)    179200      activation_46[0][0]
----------------------------------------------------------------------------------------------------
batch_normalization_42 (BatchNo (None, 7, 7, 160)    480         conv2d_42[0][0]
----------------------------------------------------------------------------------------------------
batch_normalization_47 (BatchNo (None, 7, 7, 160)    480         conv2d_47[0][0]
----------------------------------------------------------------------------------------------------
activation_42 (Activation)      (None, 7, 7, 160)    0           batch_normalization_42[0][0]
```

```
activation_47 (Activation)      (None, 7, 7, 160)    0           batch_normalization_47[0][0]
conv2d_43 (Conv2D)              (None, 7, 7, 160)    179200      activation_42[0][0]
conv2d_48 (Conv2D)              (None, 7, 7, 160)    179200      activation_47[0][0]
batch_normalization_43 (BatchNo (None, 7, 7, 160)    480         conv2d_43[0][0]
batch_normalization_48 (BatchNo (None, 7, 7, 160)    480         conv2d_48[0][0]
activation_43 (Activation)      (None, 7, 7, 160)    0           batch_normalization_43[0][0]
activation_48 (Activation)      (None, 7, 7, 160)    0           batch_normalization_48[0][0]
average_pooling2d_5 (AveragePoo (None, 7, 7, 768)    0           mixed4[0][0]
conv2d_41 (Conv2D)              (None, 7, 7, 192)    147456      mixed4[0][0]
conv2d_44 (Conv2D)              (None, 7, 7, 192)    215040      activation_43[0][0]
conv2d_49 (Conv2D)              (None, 7, 7, 192)    215040      activation_48[0][0]
conv2d_50 (Conv2D)              (None, 7, 7, 192)    147456      average_pooling2d_5[0][0]
batch_normalization_41 (BatchNo (None, 7, 7, 192)    576         conv2d_41[0][0]
batch_normalization_44 (BatchNo (None, 7, 7, 192)    576         conv2d_44[0][0]
batch_normalization_49 (BatchNo (None, 7, 7, 192)    576         conv2d_49[0][0]
batch_normalization_50 (BatchNo (None, 7, 7, 192)    576         conv2d_50[0][0]
activation_41 (Activation)      (None, 7, 7, 192)    0           batch_normalization_41[0][0]
activation_44 (Activation)      (None, 7, 7, 192)    0           batch_normalization_44[0][0]
activation_49 (Activation)      (None, 7, 7, 192)    0           batch_normalization_49[0][0]
activation_50 (Activation)      (None, 7, 7, 192)    0           batch_normalization_50[0][0]
mixed5 (Concatenate)            (None, 7, 7, 768)    0           activation_41[0][0]
                                                                 activation_44[0][0]
                                                                 activation_49[0][0]
                                                                 activation_50[0][0]
conv2d_55 (Conv2D)              (None, 7, 7, 160)    122880      mixed5[0][0]
```

```
batch_normalization_55 (BatchNo (None, 7, 7, 160)    480         conv2d_55[0][0]
__________________________________________________________________________________________________
activation_55 (Activation)      (None, 7, 7, 160)    0           batch_normalization_55[0][0]
__________________________________________________________________________________________________
conv2d_56 (Conv2D)              (None, 7, 7, 160)    179200      activation_55[0][0]
__________________________________________________________________________________________________
batch_normalization_56 (BatchNo (None, 7, 7, 160)    480         conv2d_56[0][0]
__________________________________________________________________________________________________
activation_56 (Activation)      (None, 7, 7, 160)    0           batch_normalization_56[0][0]
__________________________________________________________________________________________________
conv2d_52 (Conv2D)              (None, 7, 7, 160)    122880      mixed5[0][0]
__________________________________________________________________________________________________
conv2d_57 (Conv2D)              (None, 7, 7, 160)    179200      activation_56[0][0]
__________________________________________________________________________________________________
batch_normalization_52 (BatchNo (None, 7, 7, 160)    480         conv2d_52[0][0]
__________________________________________________________________________________________________
batch_normalization_57 (BatchNo (None, 7, 7, 160)    480         conv2d_57[0][0]
__________________________________________________________________________________________________
activation_52 (Activation)      (None, 7, 7, 160)    0           batch_normalization_52[0][0]
__________________________________________________________________________________________________
activation_57 (Activation)      (None, 7, 7, 160)    0           batch_normalization_57[0][0]
__________________________________________________________________________________________________
conv2d_53 (Conv2D)              (None, 7, 7, 160)    179200      activation_52[0][0]
__________________________________________________________________________________________________
conv2d_58 (Conv2D)              (None, 7, 7, 160)    179200      activation_57[0][0]
__________________________________________________________________________________________________
batch_normalization_53 (BatchNo (None, 7, 7, 160)    480         conv2d_53[0][0]
__________________________________________________________________________________________________
batch_normalization_58 (BatchNo (None, 7, 7, 160)    480         conv2d_58[0][0]
__________________________________________________________________________________________________
activation_53 (Activation)      (None, 7, 7, 160)    0           batch_normalization_53[0][0]
__________________________________________________________________________________________________
activation_58 (Activation)      (None, 7, 7, 160)    0           batch_normalization_58[0][0]
__________________________________________________________________________________________________
average_pooling2d_6 (AveragePoo (None, 7, 7, 768)    0           mixed5[0][0]
__________________________________________________________________________________________________
conv2d_51 (Conv2D)              (None, 7, 7, 192)    147456      mixed5[0][0]
__________________________________________________________________________________________________
conv2d_54 (Conv2D)              (None, 7, 7, 192)    215040      activation_53[0][0]
__________________________________________________________________________________________________
conv2d_59 (Conv2D)              (None, 7, 7, 192)    215040      activation_58[0][0]
__________________________________________________________________________________________________
conv2d_60 (Conv2D)              (None, 7, 7, 192)    147456      average_pooling2d_6[0][0]
__________________________________________________________________________________________________
batch_normalization_51 (BatchNo (None, 7, 7, 192)    576         conv2d_51[0][0]
__________________________________________________________________________________________________
batch_normalization_54 (BatchNo (None, 7, 7, 192)    576         conv2d_54[0][0]
__________________________________________________________________________________________________
```

```
batch_normalization_59 (BatchNo (None, 7, 7, 192)    576         conv2d_59[0][0]
--------------------------------------------------------------------------------------------------
batch_normalization_60 (BatchNo (None, 7, 7, 192)    576         conv2d_60[0][0]
--------------------------------------------------------------------------------------------------
activation_51 (Activation)      (None, 7, 7, 192)    0           batch_normalization_51[0][0]
--------------------------------------------------------------------------------------------------
activation_54 (Activation)      (None, 7, 7, 192)    0           batch_normalization_54[0][0]
--------------------------------------------------------------------------------------------------
activation_59 (Activation)      (None, 7, 7, 192)    0           batch_normalization_59[0][0]
--------------------------------------------------------------------------------------------------
activation_60 (Activation)      (None, 7, 7, 192)    0           batch_normalization_60[0][0]
--------------------------------------------------------------------------------------------------
mixed6 (Concatenate)            (None, 7, 7, 768)    0           activation_51[0][0]
                                                                 activation_54[0][0]
                                                                 activation_59[0][0]
                                                                 activation_60[0][0]
--------------------------------------------------------------------------------------------------
conv2d_65 (Conv2D)              (None, 7, 7, 192)    147456      mixed6[0][0]
--------------------------------------------------------------------------------------------------
batch_normalization_65 (BatchNo (None, 7, 7, 192)    576         conv2d_65[0][0]
--------------------------------------------------------------------------------------------------
activation_65 (Activation)      (None, 7, 7, 192)    0           batch_normalization_65[0][0]
--------------------------------------------------------------------------------------------------
conv2d_66 (Conv2D)              (None, 7, 7, 192)    258048      activation_65[0][0]
--------------------------------------------------------------------------------------------------
batch_normalization_66 (BatchNo (None, 7, 7, 192)    576         conv2d_66[0][0]
--------------------------------------------------------------------------------------------------
activation_66 (Activation)      (None, 7, 7, 192)    0           batch_normalization_66[0][0]
--------------------------------------------------------------------------------------------------
conv2d_62 (Conv2D)              (None, 7, 7, 192)    147456      mixed6[0][0]
--------------------------------------------------------------------------------------------------
conv2d_67 (Conv2D)              (None, 7, 7, 192)    258048      activation_66[0][0]
--------------------------------------------------------------------------------------------------
batch_normalization_62 (BatchNo (None, 7, 7, 192)    576         conv2d_62[0][0]
--------------------------------------------------------------------------------------------------
batch_normalization_67 (BatchNo (None, 7, 7, 192)    576         conv2d_67[0][0]
--------------------------------------------------------------------------------------------------
activation_62 (Activation)      (None, 7, 7, 192)    0           batch_normalization_62[0][0]
--------------------------------------------------------------------------------------------------
activation_67 (Activation)      (None, 7, 7, 192)    0           batch_normalization_67[0][0]
--------------------------------------------------------------------------------------------------
conv2d_63 (Conv2D)              (None, 7, 7, 192)    258048      activation_62[0][0]
--------------------------------------------------------------------------------------------------
conv2d_68 (Conv2D)              (None, 7, 7, 192)    258048      activation_67[0][0]
--------------------------------------------------------------------------------------------------
batch_normalization_63 (BatchNo (None, 7, 7, 192)    576         conv2d_63[0][0]
--------------------------------------------------------------------------------------------------
batch_normalization_68 (BatchNo (None, 7, 7, 192)    576         conv2d_68[0][0]
```

```
activation_63 (Activation)      (None, 7, 7, 192)    0           batch_normalization_63[0][0]
activation_68 (Activation)      (None, 7, 7, 192)    0           batch_normalization_68[0][0]
average_pooling2d_7 (AveragePoo (None, 7, 7, 768)    0           mixed6[0][0]
conv2d_61 (Conv2D)              (None, 7, 7, 192)    147456      mixed6[0][0]
conv2d_64 (Conv2D)              (None, 7, 7, 192)    258048      activation_63[0][0]
conv2d_69 (Conv2D)              (None, 7, 7, 192)    258048      activation_68[0][0]
conv2d_70 (Conv2D)              (None, 7, 7, 192)    147456      average_pooling2d_7[0][0]
batch_normalization_61 (BatchNo (None, 7, 7, 192)    576         conv2d_61[0][0]
batch_normalization_64 (BatchNo (None, 7, 7, 192)    576         conv2d_64[0][0]
batch_normalization_69 (BatchNo (None, 7, 7, 192)    576         conv2d_69[0][0]
batch_normalization_70 (BatchNo (None, 7, 7, 192)    576         conv2d_70[0][0]
activation_61 (Activation)      (None, 7, 7, 192)    0           batch_normalization_61[0][0]
activation_64 (Activation)      (None, 7, 7, 192)    0           batch_normalization_64[0][0]
activation_69 (Activation)      (None, 7, 7, 192)    0           batch_normalization_69[0][0]
activation_70 (Activation)      (None, 7, 7, 192)    0           batch_normalization_70[0][0]
mixed7 (Concatenate)            (None, 7, 7, 768)    0           activation_61[0][0]
                                                                 activation_64[0][0]
                                                                 activation_69[0][0]
                                                                 activation_70[0][0]
conv2d_73 (Conv2D)              (None, 7, 7, 192)    147456      mixed7[0][0]
batch_normalization_73 (BatchNo (None, 7, 7, 192)    576         conv2d_73[0][0]
activation_73 (Activation)      (None, 7, 7, 192)    0           batch_normalization_73[0][0]
conv2d_74 (Conv2D)              (None, 7, 7, 192)    258048      activation_73[0][0]
batch_normalization_74 (BatchNo (None, 7, 7, 192)    576         conv2d_74[0][0]
activation_74 (Activation)      (None, 7, 7, 192)    0           batch_normalization_74[0][0]
```

```
conv2d_71 (Conv2D)              (None, 7, 7, 192)    147456      mixed7[0][0]
----------------------------------------------------------------------------------------------------
conv2d_75 (Conv2D)              (None, 7, 7, 192)    258048      activation_74[0][0]
----------------------------------------------------------------------------------------------------
batch_normalization_71 (BatchNo (None, 7, 7, 192)    576         conv2d_71[0][0]
----------------------------------------------------------------------------------------------------
batch_normalization_75 (BatchNo (None, 7, 7, 192)    576         conv2d_75[0][0]
----------------------------------------------------------------------------------------------------
activation_71 (Activation)      (None, 7, 7, 192)    0           batch_normalization_71[0][0]
----------------------------------------------------------------------------------------------------
activation_75 (Activation)      (None, 7, 7, 192)    0           batch_normalization_75[0][0]
----------------------------------------------------------------------------------------------------
conv2d_72 (Conv2D)              (None, 3, 3, 320)    552960      activation_71[0][0]
----------------------------------------------------------------------------------------------------
conv2d_76 (Conv2D)              (None, 3, 3, 192)    331776      activation_75[0][0]
----------------------------------------------------------------------------------------------------
batch_normalization_72 (BatchNo (None, 3, 3, 320)    960         conv2d_72[0][0]
----------------------------------------------------------------------------------------------------
batch_normalization_76 (BatchNo (None, 3, 3, 192)    576         conv2d_76[0][0]
----------------------------------------------------------------------------------------------------
activation_72 (Activation)      (None, 3, 3, 320)    0           batch_normalization_72[0][0]
----------------------------------------------------------------------------------------------------
activation_76 (Activation)      (None, 3, 3, 192)    0           batch_normalization_76[0][0]
----------------------------------------------------------------------------------------------------
max_pooling2d_4 (MaxPooling2D)  (None, 3, 3, 768)    0           mixed7[0][0]
----------------------------------------------------------------------------------------------------
mixed8 (Concatenate)            (None, 3, 3, 1280)   0           activation_72[0][0]
                                                                 activation_76[0][0]
                                                                 max_pooling2d_4[0][0]
----------------------------------------------------------------------------------------------------
conv2d_81 (Conv2D)              (None, 3, 3, 448)    573440      mixed8[0][0]
----------------------------------------------------------------------------------------------------
batch_normalization_81 (BatchNo (None, 3, 3, 448)    1344        conv2d_81[0][0]
----------------------------------------------------------------------------------------------------
activation_81 (Activation)      (None, 3, 3, 448)    0           batch_normalization_81[0][0]
----------------------------------------------------------------------------------------------------
conv2d_78 (Conv2D)              (None, 3, 3, 384)    491520      mixed8[0][0]
----------------------------------------------------------------------------------------------------
conv2d_82 (Conv2D)              (None, 3, 3, 384)    1548288     activation_81[0][0]
----------------------------------------------------------------------------------------------------
batch_normalization_78 (BatchNo (None, 3, 3, 384)    1152        conv2d_78[0][0]
----------------------------------------------------------------------------------------------------
batch_normalization_82 (BatchNo (None, 3, 3, 384)    1152        conv2d_82[0][0]
----------------------------------------------------------------------------------------------------
activation_78 (Activation)      (None, 3, 3, 384)    0           batch_normalization_78[0][0]
----------------------------------------------------------------------------------------------------
activation_82 (Activation)      (None, 3, 3, 384)    0           batch_normalization_82[0][0]
----------------------------------------------------------------------------------------------------
```

```
conv2d_79 (Conv2D)              (None, 3, 3, 384)    442368      activation_78[0][0]
----------------------------------------------------------------------------------------------------
conv2d_80 (Conv2D)              (None, 3, 3, 384)    442368      activation_78[0][0]
----------------------------------------------------------------------------------------------------
conv2d_83 (Conv2D)              (None, 3, 3, 384)    442368      activation_82[0][0]
----------------------------------------------------------------------------------------------------
conv2d_84 (Conv2D)              (None, 3, 3, 384)    442368      activation_82[0][0]
----------------------------------------------------------------------------------------------------
average_pooling2d_8 (AveragePoo (None, 3, 3, 1280)   0           mixed8[0][0]
----------------------------------------------------------------------------------------------------
conv2d_77 (Conv2D)              (None, 3, 3, 320)    409600      mixed8[0][0]
----------------------------------------------------------------------------------------------------
batch_normalization_79 (BatchNo (None, 3, 3, 384)    1152        conv2d_79[0][0]
----------------------------------------------------------------------------------------------------
batch_normalization_80 (BatchNo (None, 3, 3, 384)    1152        conv2d_80[0][0]
----------------------------------------------------------------------------------------------------
batch_normalization_83 (BatchNo (None, 3, 3, 384)    1152        conv2d_83[0][0]
----------------------------------------------------------------------------------------------------
batch_normalization_84 (BatchNo (None, 3, 3, 384)    1152        conv2d_84[0][0]
----------------------------------------------------------------------------------------------------
conv2d_85 (Conv2D)              (None, 3, 3, 192)    245760      average_pooling2d_8[0][0]
----------------------------------------------------------------------------------------------------
batch_normalization_77 (BatchNo (None, 3, 3, 320)    960         conv2d_77[0][0]
----------------------------------------------------------------------------------------------------
activation_79 (Activation)      (None, 3, 3, 384)    0           batch_normalization_79[0][0]
----------------------------------------------------------------------------------------------------
activation_80 (Activation)      (None, 3, 3, 384)    0           batch_normalization_80[0][0]
----------------------------------------------------------------------------------------------------
activation_83 (Activation)      (None, 3, 3, 384)    0           batch_normalization_83[0][0]
----------------------------------------------------------------------------------------------------
activation_84 (Activation)      (None, 3, 3, 384)    0           batch_normalization_84[0][0]
----------------------------------------------------------------------------------------------------
batch_normalization_85 (BatchNo (None, 3, 3, 192)    576         conv2d_85[0][0]
----------------------------------------------------------------------------------------------------
activation_77 (Activation)      (None, 3, 3, 320)    0           batch_normalization_77[0][0]
----------------------------------------------------------------------------------------------------
mixed9_0 (Concatenate)          (None, 3, 3, 768)    0           activation_79[0][0]
                                                                 activation_80[0][0]
----------------------------------------------------------------------------------------------------
concatenate_1 (Concatenate)     (None, 3, 3, 768)    0           activation_83[0][0]
                                                                 activation_84[0][0]
----------------------------------------------------------------------------------------------------
activation_85 (Activation)      (None, 3, 3, 192)    0           batch_normalization_85[0][0]
----------------------------------------------------------------------------------------------------
mixed9 (Concatenate)            (None, 3, 3, 2048)   0           activation_77[0][0]
                                                                 mixed9_0[0][0]
                                                                 concatenate_1[0][0]
                                                                 activation_85[0][0]
```

```
conv2d_90 (Conv2D)              (None, 3, 3, 448)    917504      mixed9[0][0]
batch_normalization_90 (BatchNo (None, 3, 3, 448)    1344        conv2d_90[0][0]
activation_90 (Activation)      (None, 3, 3, 448)    0           batch_normalization_90[0][0]
conv2d_87 (Conv2D)              (None, 3, 3, 384)    786432      mixed9[0][0]
conv2d_91 (Conv2D)              (None, 3, 3, 384)    1548288     activation_90[0][0]
batch_normalization_87 (BatchNo (None, 3, 3, 384)    1152        conv2d_87[0][0]
batch_normalization_91 (BatchNo (None, 3, 3, 384)    1152        conv2d_91[0][0]
activation_87 (Activation)      (None, 3, 3, 384)    0           batch_normalization_87[0][0]
activation_91 (Activation)      (None, 3, 3, 384)    0           batch_normalization_91[0][0]
conv2d_88 (Conv2D)              (None, 3, 3, 384)    442368      activation_87[0][0]
conv2d_89 (Conv2D)              (None, 3, 3, 384)    442368      activation_87[0][0]
conv2d_92 (Conv2D)              (None, 3, 3, 384)    442368      activation_91[0][0]
conv2d_93 (Conv2D)              (None, 3, 3, 384)    442368      activation_91[0][0]
average_pooling2d_9 (AveragePoo (None, 3, 3, 2048)   0           mixed9[0][0]
conv2d_86 (Conv2D)              (None, 3, 3, 320)    655360      mixed9[0][0]
batch_normalization_88 (BatchNo (None, 3, 3, 384)    1152        conv2d_88[0][0]
batch_normalization_89 (BatchNo (None, 3, 3, 384)    1152        conv2d_89[0][0]
batch_normalization_92 (BatchNo (None, 3, 3, 384)    1152        conv2d_92[0][0]
batch_normalization_93 (BatchNo (None, 3, 3, 384)    1152        conv2d_93[0][0]
conv2d_94 (Conv2D)              (None, 3, 3, 192)    393216      average_pooling2d_9[0][0]
batch_normalization_86 (BatchNo (None, 3, 3, 320)    960         conv2d_86[0][0]
activation_88 (Activation)      (None, 3, 3, 384)    0           batch_normalization_88[0][0]
activation_89 (Activation)      (None, 3, 3, 384)    0           batch_normalization_89[0][0]
activation_92 (Activation)      (None, 3, 3, 384)    0           batch_normalization_92[0][0]
```

```
__________________________________________________________________________________________________
activation_93 (Activation)      (None, 3, 3, 384)    0           batch_normalization_93[0][0]
__________________________________________________________________________________________________
batch_normalization_94 (BatchNo (None, 3, 3, 192)    576         conv2d_94[0][0]
__________________________________________________________________________________________________
activation_86 (Activation)      (None, 3, 3, 320)    0           batch_normalization_86[0][0]
__________________________________________________________________________________________________
mixed9_1 (Concatenate)          (None, 3, 3, 768)    0           activation_88[0][0]
                                                                 activation_89[0][0]
__________________________________________________________________________________________________
concatenate_2 (Concatenate)     (None, 3, 3, 768)    0           activation_92[0][0]
                                                                 activation_93[0][0]
__________________________________________________________________________________________________
activation_94 (Activation)      (None, 3, 3, 192)    0           batch_normalization_94[0][0]
__________________________________________________________________________________________________
mixed10 (Concatenate)           (None, 3, 3, 2048)   0           activation_86[0][0]
                                                                 mixed9_1[0][0]
                                                                 concatenate_2[0][0]
                                                                 activation_94[0][0]
==================================================================================================
Total params: 21,802,784
Trainable params: 0
Non-trainable params: 21,802,784
__________________________________________________________________________________________________
```

In a normal Inception network, you would see from the model summary that the last two layers were a global average pooling layer, and a fully-connected "Dense" layer. However, since we set `include_top` to `False`, both of these get dropped. If you otherwise wanted to drop additional layers, you would use:

```
inception.layers.pop()
```

Note that `pop()` works from the end of the model backwards.

It's important to note two things here: 1. How many layers you drop is up to you, typically. We dropped the final two already by setting `include_top` to False in the original loading of the model, but you could instead just run `pop()` twice to achieve similar results. (*Note:* Keras requires us to set `include_top` to False in order to change the `input_shape`.) Additional layers could be dropped by additional calls to `pop()`. 2. If you make a mistake with `pop()`, you'll want to reload the model. If you use it multiple times, the model will continue to drop more and more layers, so you may need to check `model.summary()` again to check your work.

1.0.4 Adding new layers

Now, you can start to add your own layers. While we've used Keras's `Sequential` model before for simplicity, we'll actually use the Model API this time. This functions a little differently, in that instead of using `model.add()`, you explicitly tell the model which previous layer to attach to the current layer. This is useful if you want to use more advanced concepts like skip layers, for instance (which were used heavily in ResNet).

For example, if you had a previous layer named `inp`:

```
x = Dropout(0.2)(inp)
```

is how you would attach a new dropout layer `x`, with it's input coming from a layer with the variable name `inp`.

We are going to use the CIFAR-10 dataset, which consists of 60,000 32x32 images of 10 classes. We need to use Keras's `Input` function to do so, and then we want to re-size the images up to the `input_size` we specified earlier (139x139).

```
In [4]: from keras.layers import Input, Lambda
        import tensorflow as tf

        # Makes the input placeholder layer 32x32x3 for CIFAR-10
        cifar_input = Input(shape=(32,32,3))

        # Re-sizes the input with Kera's Lambda layer & attach to cifar_input
        resized_input = Lambda(lambda image: tf.image.resize_images(
            image, (input_size, input_size)))(cifar_input)

        # Feeds the re-sized input into Inception model
        # You will need to update the model name if you changed it earlier!
        inp = inception(resized_input)

In [5]: # Imports fully-connected "Dense" layers & Global Average Pooling
        from keras.layers import Dense, GlobalAveragePooling2D

        ## TODO: Setting `include_top` to False earlier also removed the
        ##       GlobalAveragePooling2D layer, but we still want it.
        ##       Add it here, and make sure to connect it to the end of Inception
        x = GlobalAveragePooling2D()(inp)

        ## TODO: Create two new fully-connected layers using the Model API
        ##       format discussed above. The first layer should use `out`
        ##       as its input, along with ReLU activation. You can choose
        ##       how many nodes it has, although 512 or less is a good idea.
        ##       The second layer should take this first layer as input, and
        ##       be named "predictions", with Softmax activation and
        ##       10 nodes, as we'll be using the CIFAR10 dataset.
        x = Dense(512, activation = 'relu')(x)
        predictions = Dense(10, activation = 'softmax')(x)
```

We're almost done with our new model! Now we just need to use the actual Model API to create the full model.

```
In [6]: # Imports the Model API
        from keras.models import Model

        # Creates the model, assuming your final layer is named "predictions"
```

```
        model = Model(inputs=cifar_input, outputs=predictions)

        # Compile the model
        model.compile(optimizer='Adam', loss='categorical_crossentropy', metrics=['accuracy'])

        # Check the summary of this new model to confirm the architecture
        model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         (None, 32, 32, 3)         0
_________________________________________________________________
lambda_1 (Lambda)            (None, 139, 139, 3)       0
_________________________________________________________________
inception_v3 (Model)         (None, 3, 3, 2048)        21802784
_________________________________________________________________
global_average_pooling2d_1 ( (None, 2048)              0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               1049088
_________________________________________________________________
dense_2 (Dense)              (None, 10)                5130
=================================================================
Total params: 22,857,002
Trainable params: 1,054,218
Non-trainable params: 21,802,784
_________________________________________________________________
```

Great job creating a new model architecture from Inception! Notice how this method of adding layers before InceptionV3 and appending to the end of it made InceptionV3 condense down into one line in the summary; if you use the Inception model's normal input (which you could gather from `inception.layers.input`), it would instead show all the layers like before.

Most of the rest of the code in the notebook just goes toward loading our data, pre-processing it, and starting our training in Keras, although there's one other good point to make here - Keras callbacks.

1.0.5 Keras Callbacks

Keras callbacks allow you to gather and store additional information during training, such as the best model, or even stop training early if the validation accuracy has stopped improving. These methods can help to avoid overfitting, or avoid other issues.

There's two key callbacks to mention here, `ModelCheckpoint` and `EarlyStopping`. As the names may suggest, model checkpoint saves down the best model so far based on a given metric, while early stopping will end training before the specified number of epochs if the chosen metric no longer improves after a given amount of time.

To set these callbacks, you could do the following:

```
checkpoint = ModelCheckpoint(filepath=save_path, monitor='val_loss', save_best_only=True)
```

This would save a model to a specified `save_path`, based on validation loss, and only save down the best models. If you set `save_best_only` to `False`, every single epoch will save down another version of the model.

```
stopper = EarlyStopping(monitor='val_acc', min_delta=0.0003, patience=5)
```

This will monitor validation accuracy, and if it has not decreased by more than 0.0003 from the previous best validation accuracy for 5 epochs, training will end early.

You still need to actually feed these callbacks into `fit()` when you train the model (along with all other relevant data to feed into `fit`):

```
model.fit(callbacks=[checkpoint, stopper])
```

1.1 GPU time

The rest of the notebook will give you the code for training, so you can turn on the GPU at this point - but first, **make sure to save your jupyter notebook**. Once the GPU is turned on, it will load whatever your last notebook checkpoint is.

While we suggest reading through the code below to make sure you understand it, you can otherwise go ahead and select *Cell* > *Run All* (or *Kernel* > *Restart & Run All* if already using GPU) to run through all cells in the notebook.

In [7]:
```
from sklearn.utils import shuffle
from sklearn.preprocessing import LabelBinarizer
from keras.datasets import cifar10

(X_train, y_train), (X_val, y_val) = cifar10.load_data()

# One-hot encode the labels
label_binarizer = LabelBinarizer()
y_one_hot_train = label_binarizer.fit_transform(y_train)
y_one_hot_val = label_binarizer.fit_transform(y_val)

# Shuffle the training & test data
X_train, y_one_hot_train = shuffle(X_train, y_one_hot_train)
X_val, y_one_hot_val = shuffle(X_val, y_one_hot_val)

# We are only going to use the first 10,000 images for speed reasons
# And only the first 2,000 images from the test set
X_train = X_train[:10000]
y_one_hot_train = y_one_hot_train[:10000]
X_val = X_val[:2000]
y_one_hot_val = y_one_hot_val[:2000]
```

```
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
170500096/170498071 [==============================] - 2s 0us/step
```

You can check out Keras's ImageDataGenerator documentation for more information on the below - you can also add additional image augmentation through this function, although we are skipping that step here so you can potentially explore it in the upcoming project.

```
In [8]: # Use a generator to pre-process our images for ImageNet
        from keras.preprocessing.image import ImageDataGenerator
        from keras.applications.inception_v3 import preprocess_input

        if preprocess_flag == True:
            datagen = ImageDataGenerator(preprocessing_function=preprocess_input)
            val_datagen = ImageDataGenerator(preprocessing_function=preprocess_input)
        else:
            datagen = ImageDataGenerator()
            val_datagen = ImageDataGenerator()
```

```
In [9]: # Train the model
        batch_size = 32
        epochs = 5
        # Note: we aren't using callbacks here since we only are using 5 epochs to conserve GPU
        model.fit_generator(datagen.flow(X_train, y_one_hot_train, batch_size=batch_size),
                            steps_per_epoch=len(X_train)/batch_size, epochs=epochs, verbose=1,
                            validation_data=val_datagen.flow(X_val, y_one_hot_val, batch_size=ba
                            validation_steps=len(X_val)/batch_size)
```

```
Epoch 1/5
313/312 [==============================] - 49s 156ms/step - loss: 1.2783 - acc: 0.5731 - val_los
Epoch 2/5
313/312 [==============================] - 45s 145ms/step - loss: 0.9639 - acc: 0.6675 - val_los
Epoch 3/5
313/312 [==============================] - 46s 148ms/step - loss: 0.9008 - acc: 0.6885 - val_los
Epoch 4/5
313/312 [==============================] - 46s 148ms/step - loss: 0.8352 - acc: 0.7122 - val_los
Epoch 5/5
313/312 [==============================] - 46s 148ms/step - loss: 0.7896 - acc: 0.7287 - val_los
```

```
Out[9]: <keras.callbacks.History at 0x7f916dc36cf8>
```

As you may have noticed, CIFAR-10 is a fairly tough dataset. However, given that we are only training on a small subset of the data, only training for five epochs, and not using any image augmentation, the results are still fairly impressive!

We achieved ~70% validation accuracy here, although your results may vary.

1.2 [Optional] Test without frozen weights, or by training from scratch.

Since the majority of the model was frozen above, training speed is pretty quick. You may also want to check out the training speed, as well as final accuracy, if you don't freeze the weights. Note that this can be fairly slow, so we're marking this as optional in order to conserve GPU time.

If you do want to see the results from doing so, go back to the first code cell and set `freeze_flag` to `False`. If you want to completely train from scratch without ImageNet pre-trained weights, follow the previous step as well as setting `weights_flag` to `None`. Then, go to *Kernel* > *Restart & Run All*.

1.3 Comparison

So that you don't use up your GPU time, we've tried out these results ourselves as well.

Training Mode	Val Acc @ 1 epoch	Val Acc @ 5 epoch	Time per epoch
Frozen weights	65.5%	70.3%	50 seconds
Unfrozen weights	50.6%	71.6%	142 seconds
No pre-trained weights	19.2%	39.2%	142 seconds

From the above, we can see that the pre-trained model with frozen weights actually began converging the fastest (already at 65.5% after 1 epoch), while the model re-training from the pre-trained weights slightly edged it out after 5 epochs.

However, this does not tell the whole story - the training accuracy was substantially higher, nearing 87% for the unfrozen weights model. It actually began overfit the data much more under this method. We would likely be able to counteract some of this issue by using data augmentation. On the flip side, the model using frozen weights could also have been improved by actually only freezing a portion of the weights; some of these are likely more specific to ImageNet classes as it gets later in the network, as opposed to the simpler features extracted early in the network.

1.3.1 The Power of Transfer Learning

Comparing the last line to the other two really shows the power of transfer learning. After five epochs, a model without ImageNet pre-training had only achieved 39.2% accuracy, compared to over 70% for the other two. As such, pre-training the network has saved substantial time, especially given the additional training time needed when the weights are not frozen.

There is also evidence found in various research that pre-training on ImageNet weights will result in a higher overall accuracy than completely training from scratch, even when using a substantially different dataset.