dlnd_face_generation

July 25, 2017

1 Face Generation

In this project, you'll use generative adversarial networks to generate new images of faces. ### Get the Data You'll be using two datasets in this project: - MNIST - CelebA

Since the celebA dataset is complex and you're doing GANs in a project for the first time, we want you to test your neural network on MNIST before CelebA. Running the GANs on MNIST will allow you to see how well your model trains sooner.

If you're using FloydHub, set `data_dir` to "/input" and use the FloydHub data ID "R5KrjnANiKVhLWAkpXhNBe".

```
In [1]: data_dir = './data'

        # FloydHub - Use with data ID "R5KrjnANiKVhLWAkpXhNBe"
        #data_dir = '/input'


        """
        DON'T MODIFY ANYTHING IN THIS CELL
        """
        import helper

        helper.download_extract('mnist', data_dir)
        helper.download_extract('celeba', data_dir)
```

```
Found mnist Data
Found celeba Data
```

1.1 Explore the Data

1.1.1 MNIST

As you're aware, the MNIST dataset contains images of handwritten digits. You can view the first number of examples by changing `show_n_images`.

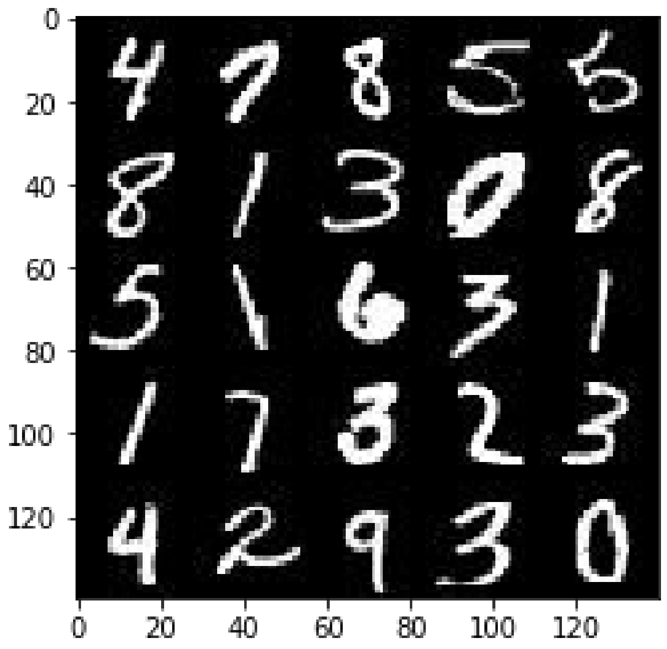
```
In [2]: show_n_images = 25

        """
```

```
DON'T MODIFY ANYTHING IN THIS CELL
"""
%matplotlib inline
import os
from glob import glob
from matplotlib import pyplot

mnist_images = helper.get_batch(glob(os.path.join(data_dir, 'mnist/*.jpg'))[:show_n_imag
pyplot.imshow(helper.images_square_grid(mnist_images, 'L'), cmap='gray')
```

Out[2]:
```
<matplotlib.image.AxesImage at 0x7fa2fa64dac8>
```



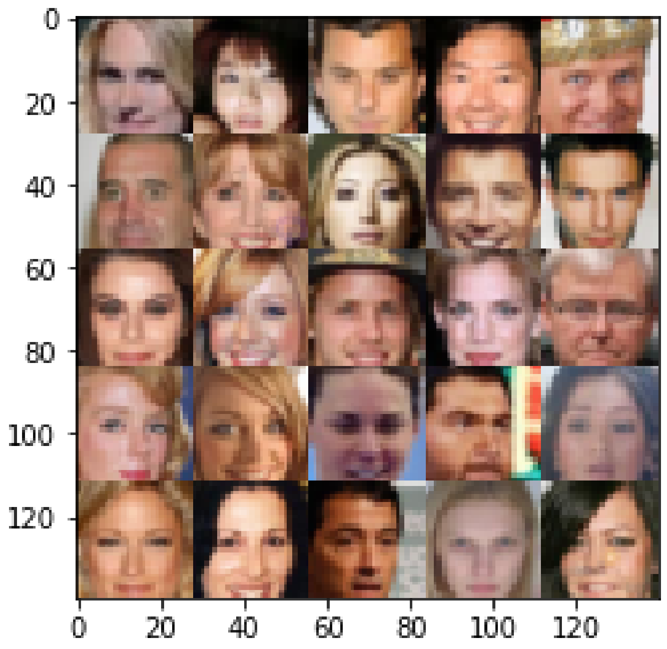
1.1.2 CelebA

The CelebFaces Attributes Dataset (CelebA) dataset contains over 200,000 celebrity images with annotations. Since you're going to be generating faces, you won't need the annotations. You can view the first number of examples by changing `show_n_images`.

In [3]:
```
show_n_images = 25

"""
DON'T MODIFY ANYTHING IN THIS CELL
"""
mnist_images = helper.get_batch(glob(os.path.join(data_dir, 'img_align_celeba/*.jpg'))[:
pyplot.imshow(helper.images_square_grid(mnist_images, 'RGB'))
```

```
Out[3]: <matplotlib.image.AxesImage at 0x7fa2f74575c0>
```



1.2 Preprocess the Data

Since the project's main focus is on building the GANs, we'll preprocess the data for you. The values of the MNIST and CelebA dataset will be in the range of -0.5 to 0.5 of 28x28 dimensional images. The CelebA images will be cropped to remove parts of the image that don't include a face, then resized down to 28x28.

The MNIST images are black and white images with a single [color channel](https://en.wikipedia.org/wiki/Channel_(digital_image%29) while the CelebA images have [3 color channels (RGB color channel)](https://en.wikipedia.org/wiki/Channel_(digital_image%29#RGB_Images). ## Build the Neural Network You'll build the components necessary to build a GANs by implementing the following functions below: - `model_inputs` - `discriminator` - `generator` - `model_loss` - `model_opt` - `train`

1.2.1 Check the Version of TensorFlow and Access to GPU

This will check to make sure you have the correct version of TensorFlow and access to a GPU

```
In [4]: """
        DON'T MODIFY ANYTHING IN THIS CELL
        """
        from distutils.version import LooseVersion
        import warnings
        import tensorflow as tf
        # Check TensorFlow Version
        assert LooseVersion(tf.__version__) >= LooseVersion('1.0'), 'Please use TensorFlow versi
        print('TensorFlow Version: {}'.format(tf.__version__))

        # Check for a GPU
        if not tf.test.gpu_device_name():
            warnings.warn('No GPU found. Please use a GPU to train your neural network.')
        else:
            print('Default GPU Device: {}'.format(tf.test.gpu_device_name()))
```

```
TensorFlow Version: 1.1.0
Default GPU Device: /gpu:0
```

1.2.2 Input

Implement the `model_inputs` function to create TF Placeholders for the Neural Network. It should create the following placeholders: - Real input images placeholder with rank 4 using `image_width`, `image_height`, and `image_channels`. - Z input placeholder with rank 2 using `z_dim`. - Learning rate placeholder with rank 0.

Return the placeholders in the following the tuple (tensor of real input images, tensor of z data)

```
In [14]: import problem_unittests as tests

         def model_inputs(image_width, image_height, image_channels, z_dim):
             """
             Create the model inputs
             :param image_width: The input image width
             :param image_height: The input image height
             :param image_channels: The number of image channels
             :param z_dim: The dimension of Z
             :return: Tuple of (tensor of real input images, tensor of z data, learning rate)
             """
             inputs_real = tf.placeholder(tf.float32, (None, image_width, image_height, image_ch
             inputs_z = tf.placeholder(tf.float32, (None, z_dim), name='input_z')
             learning_rate = tf.placeholder(tf.float32, (None))
             return inputs_real, inputs_z, learning_rate



         """
         DON'T MODIFY ANYTHING IN THIS CELL THAT IS BELOW THIS LINE
         """
         tests.test_model_inputs(model_inputs)
```

```
Tests Passed
```

1.2.3 Discriminator

Implement `discriminator` to create a discriminator neural network that discriminates on `images`. This function should be able to reuse the variables in the neural network. Use `tf.variable_scope` with a scope name of "discriminator" to allow the variables to be reused. The function should return a tuple of (tensor output of the discriminator, tensor logits of the discriminator).

```
In [16]: def discriminator(images, reuse=False):
             """
             Create the discriminator network
             :param images: Tensor of input image(s)
             :param reuse: Boolean if the weights should be reused
             :return: Tuple of (tensor output of the discriminator, tensor logits of the discrim
             """
             alpha=0.2
             x = images
             with tf.variable_scope('discriminator', reuse=reuse):
                 x = tf.layers.conv2d(x, 64, 4, strides=2, padding="same")
                 x = tf.layers.batch_normalization(x, training=True)
                 x = tf.maximum(alpha * x, x)
                 #x = tf.layers.dropout(x, 0.5)

                 x = tf.layers.conv2d(x, 128, 4, strides=2, padding="same")
                 x = tf.layers.batch_normalization(x, training=True)
                 x = tf.maximum(alpha * x, x)
                 #x = tf.layers.dropout(x, 0.5)

                 x = tf.layers.conv2d(x, 256, 4, strides=2, padding="same")
                 x = tf.layers.batch_normalization(x, training=True)
                 x = tf.maximum(alpha * x, x)
                 #x = tf.layers.dropout(x, 0.5)

                 x = tf.reshape(x, (-1, 4 * 4 * 256))
                 logits = tf.layers.dense(x, 1)
                 out = tf.sigmoid(logits)

             return out, logits

         """
         DON'T MODIFY ANYTHING IN THIS CELL THAT IS BELOW THIS LINE
         """
         tests.test_discriminator(discriminator, tf)
```

```
Tests Passed
```

1.2.4 Generator

Implement `generator` to generate an image using `z`. This function should be able to reuse the variables in the neural network. Use `tf.variable_scope` with a scope name of "generator" to allow the variables to be reused. The function should return the generated 28 x 28 x `out_channel_dim` images.

```
In [17]: def generator(z, out_channel_dim, is_train=True):
             """
             Create the generator network
             :param z: Input z
             :param out_channel_dim: The number of channels in the output image
             :param is_train: Boolean if generator is being used for training
             :return: The tensor output of the generator
             """
             reuse = not is_train
             alpha= 0.2
             with tf.variable_scope('generator', reuse=reuse):
                 x = tf.layers.dense(z, 4 * 4 * 512)

                 x = tf.reshape(x, (-1, 4, 4, 512))
                 x = tf.layers.batch_normalization(x, training=is_train)
                 #x = tf.layers.dropout(x, 0.5)
                 x = tf.maximum(alpha * x, x)
                 #print(x.shape)
                 x = tf.layers.conv2d_transpose(x, 256, 4, strides=1, padding="valid")
                 x = tf.layers.batch_normalization(x,training=is_train)
                 x = tf.maximum(alpha * x, x)
                 #print(x.shape)
                 x = tf.layers.conv2d_transpose(x, 128, 4, strides=2, padding="same")
                 x = tf.layers.batch_normalization(x,training=is_train)
                 x = tf.maximum(alpha * x, x)
                 #print(x.shape)
                 x = tf.layers.conv2d_transpose(x, out_channel_dim, 4, strides=2, padding="same"
                 #x = tf.maximum(alpha * x, x)

                 logits = x
                 out = tf.tanh(logits)

             return out



         """
         DON'T MODIFY ANYTHING IN THIS CELL THAT IS BELOW THIS LINE
         """
         tests.test_generator(generator, tf)
```

```
Tests Passed
```

1.2.5 Loss

Implement `model_loss` to build the GANs for training and calculate the loss. The function should return a tuple of (discriminator loss, generator loss). Use the following functions you implemented: - `discriminator(images, reuse=False)` - `generator(z, out_channel_dim, is_train=True)`

```
In [18]: def model_loss(input_real, input_z, out_channel_dim):
             """
             Get the loss for the discriminator and generator
             :param input_real: Images from the real dataset
             :param input_z: Z input
             :param out_channel_dim: The number of channels in the output image
             :return: A tuple of (discriminator loss, generator loss)
             """
             smooth = 0.1
             _, d_logits_real = discriminator(input_real, reuse=False)
             fake = generator(input_z, out_channel_dim, is_train=True)
             d_logits_fake = discriminator(fake, reuse=True)
             # Calculate losses
             d_loss_real = tf.reduce_mean(
                                 tf.nn.sigmoid_cross_entropy_with_logits(logits=d_logits_real,
                                                                         labels=tf.ones_like(d_log
             d_loss_fake = tf.reduce_mean(
                                 tf.nn.sigmoid_cross_entropy_with_logits(logits=d_logits_fake,
                                                                         labels=tf.zeros_like(d_lo
             d_loss = d_loss_real + d_loss_fake

             g_loss = tf.reduce_mean(
                             tf.nn.sigmoid_cross_entropy_with_logits(logits=d_logits_fake,
                                                                     labels=tf.ones_like(d_logits_f
             return d_loss, g_loss


         """
         DON'T MODIFY ANYTHING IN THIS CELL THAT IS BELOW THIS LINE
         """
         tests.test_model_loss(model_loss)
```

```
Tests Passed
```

1.2.6 Optimization

Implement `model_opt` to create the optimization operations for the GANs. Use `tf.trainable_variables` to get all the trainable variables. Filter the variables with names that are in the discriminator and generator scope names. The function should return a tuple of (discriminator training operation, generator training operation).

```
In [19]: def model_opt(d_loss, g_loss, learning_rate, beta1):
             """
             Get optimization operations
             :param d_loss: Discriminator loss Tensor
             :param g_loss: Generator loss Tensor
             :param learning_rate: Learning Rate Placeholder
             :param beta1: The exponential decay rate for the 1st moment in the optimizer
             :return: A tuple of (discriminator training operation, generator training operation
             """
             t_vars = tf.trainable_variables()
             g_vars = [var for var in t_vars if var.name.startswith('generator')]
             d_vars = [var for var in t_vars if var.name.startswith('discriminator')]
             all_update_ops = tf.get_collection(tf.GraphKeys.UPDATE_OPS)

             g_update_ops = [var for var in all_update_ops if var.name.startswith('generator')]
             d_update_ops = [var for var in all_update_ops if var.name.startswith('discriminator

             with tf.control_dependencies(d_update_ops):
                 d_train_opt = tf.train.AdamOptimizer(learning_rate, beta1=beta1).minimize(d_los
             with tf.control_dependencies(g_update_ops):
                 g_train_opt = tf.train.AdamOptimizer(learning_rate, beta1=beta1).minimize(g_los
             return d_train_opt, g_train_opt


         """
         DON'T MODIFY ANYTHING IN THIS CELL THAT IS BELOW THIS LINE
         """
         tests.test_model_opt(model_opt, tf)
```

```
Tests Passed
```

1.3 Neural Network Training

1.3.1 Show Output

Use this function to show the current output of the generator during training. It will help you determine how well the GANs is training.

```
In [20]: """
         DON'T MODIFY ANYTHING IN THIS CELL
         """
         import numpy as np

         def show_generator_output(sess, n_images, input_z, out_channel_dim, image_mode):
             """
```

```
            Show example output for the generator
            :param sess: TensorFlow session
            :param n_images: Number of Images to display
            :param input_z: Input Z Tensor
            :param out_channel_dim: The number of channels in the output image
            :param image_mode: The mode to use for images ("RGB" or "L")
            """
            cmap = None if image_mode == 'RGB' else 'gray'
            z_dim = input_z.get_shape().as_list()[-1]
            example_z = np.random.uniform(-1, 1, size=[n_images, z_dim])

            samples = sess.run(
                generator(input_z, out_channel_dim, False),
                feed_dict={input_z: example_z})

            images_grid = helper.images_square_grid(samples, image_mode)
            pyplot.imshow(images_grid, cmap=cmap)
            pyplot.show()
```

1.3.2 Train

Implement `train` to build and train the GANs. Use the following functions you implemented: - `model_inputs(image_width, image_height, image_channels, z_dim)` - `model_loss(input_real, input_z, out_channel_dim)` - `model_opt(d_loss, g_loss, learning_rate, beta1)`

Use the `show_generator_output` to show `generator` output while you train. Running `show_generator_output` for every batch will drastically increase training time and increase the size of the notebook. It's recommended to print the `generator` output every 100 batches.

```
In [49]: def train(epoch_count, batch_size, z_dim, learning_rate, beta1, get_batches, data_shape
            """
            Train the GAN
            :param epoch_count: Number of epochs
            :param batch_size: Batch Size
            :param z_dim: Z dimension
            :param learning_rate: Learning Rate
            :param beta1: The exponential decay rate for the 1st moment in the optimizer
            :param get_batches: Function to get batches
            :param data_shape: Shape of the data
            :param data_image_mode: The image mode to use for images ("RGB" or "L")
            """
            inputs_real, inputs_z, lr = model_inputs(data_shape[1], data_shape[2], data_shape[3
            d_loss, g_loss = model_loss(inputs_real, inputs_z, data_shape[-1])
            d_train_opt, g_train_opt = model_opt(d_loss, g_loss, learning_rate, beta1)
            batch_num = 0

            with tf.Session() as sess:
                sess.run(tf.global_variables_initializer())
                for epoch_i in range(epoch_count):
                    for batch_images in get_batches(batch_size):
                        batch_num = batch_num+1
                        batch_images = batch_images * 2
                        batch_z = np.random.uniform(-1, 1, size=(batch_size, z_dim))
                        _ = sess.run(d_train_opt, feed_dict={inputs_real: batch_images, inputs_
                        _ = sess.run(g_train_opt, feed_dict={inputs_z: batch_z, lr:learning_rat

                        if batch_num % 100 == 0:
                            train_loss_d = d_loss.eval({inputs_z:batch_z, inputs_real: batch_im
                            train_loss_g = g_loss.eval({inputs_z:batch_z})
                            print("Epoch {}/{} batch {}...".format(epoch_i+1, epoch_count, batc
                              "Discriminator Loss: {:.4f}...".format(train_loss_d),
                              "Generator Loss: {:.4f}".format(train_loss_g))
```

1.3.3 MNIST

Test your GANs architecture on MNIST. After 2 epochs, the GANs should be able to generate images that look like handwritten digits. Make sure the loss of the generator is lower than the loss of the discriminator or close to 0.

```
In [50]: batch_size = 10
         z_dim = 10
         learning_rate = 0.001
         beta1 = 0.3


         """
         DON'T MODIFY ANYTHING IN THIS CELL THAT IS BELOW THIS LINE
         """
         epochs = 2

         mnist_dataset = helper.Dataset('mnist', glob(os.path.join(data_dir, 'mnist/*.jpg')))
         with tf.Graph().as_default():
             train(epochs, batch_size, z_dim, learning_rate, beta1, mnist_dataset.get_batches,
                   mnist_dataset.shape, mnist_dataset.image_mode)
```

```
Epoch 1/2 batch 100... Discriminator Loss: 1.6789... Generator Loss: 0.6887
Epoch 1/2 batch 200... Discriminator Loss: 1.2866... Generator Loss: 0.7252
Epoch 1/2 batch 300... Discriminator Loss: 1.3163... Generator Loss: 0.7115
Epoch 1/2 batch 400... Discriminator Loss: 1.4322... Generator Loss: 0.4245
Epoch 1/2 batch 500... Discriminator Loss: 1.2263... Generator Loss: 0.8442
Epoch 1/2 batch 600... Discriminator Loss: 1.2774... Generator Loss: 0.7829
Epoch 1/2 batch 700... Discriminator Loss: 1.4950... Generator Loss: 0.3994
Epoch 1/2 batch 800... Discriminator Loss: 1.1939... Generator Loss: 0.6612
Epoch 1/2 batch 900... Discriminator Loss: 2.2372... Generator Loss: 0.2341
Epoch 1/2 batch 1000... Discriminator Loss: 1.3250... Generator Loss: 0.5454
Epoch 1/2 batch 1100... Discriminator Loss: 1.6707... Generator Loss: 0.3180
Epoch 1/2 batch 1200... Discriminator Loss: 1.2142... Generator Loss: 0.6209
Epoch 1/2 batch 1300... Discriminator Loss: 1.5464... Generator Loss: 1.3141
Epoch 1/2 batch 1400... Discriminator Loss: 1.1088... Generator Loss: 0.8218
Epoch 1/2 batch 1500... Discriminator Loss: 1.3211... Generator Loss: 1.5244
Epoch 1/2 batch 1600... Discriminator Loss: 1.0463... Generator Loss: 1.2508
Epoch 1/2 batch 1700... Discriminator Loss: 1.3795... Generator Loss: 1.4649
Epoch 1/2 batch 1800... Discriminator Loss: 1.9128... Generator Loss: 0.2970
Epoch 1/2 batch 1900... Discriminator Loss: 1.0022... Generator Loss: 0.8857
Epoch 1/2 batch 2000... Discriminator Loss: 1.2413... Generator Loss: 1.1919
Epoch 1/2 batch 2100... Discriminator Loss: 1.0639... Generator Loss: 1.1489
Epoch 1/2 batch 2200... Discriminator Loss: 1.1747... Generator Loss: 0.7274
Epoch 1/2 batch 2300... Discriminator Loss: 1.2481... Generator Loss: 0.8835
Epoch 1/2 batch 2400... Discriminator Loss: 1.2816... Generator Loss: 0.7128
Epoch 1/2 batch 2500... Discriminator Loss: 1.1665... Generator Loss: 0.6118
Epoch 1/2 batch 2600... Discriminator Loss: 0.9984... Generator Loss: 0.8312
Epoch 1/2 batch 2700... Discriminator Loss: 1.9101... Generator Loss: 0.4434
Epoch 1/2 batch 2800... Discriminator Loss: 0.9848... Generator Loss: 0.8489
Epoch 1/2 batch 2900... Discriminator Loss: 1.0496... Generator Loss: 1.3236
Epoch 1/2 batch 3000... Discriminator Loss: 1.9071... Generator Loss: 0.3629
Epoch 1/2 batch 3100... Discriminator Loss: 0.9061... Generator Loss: 1.0675
Epoch 1/2 batch 3200... Discriminator Loss: 1.3569... Generator Loss: 0.5156
Epoch 1/2 batch 3300... Discriminator Loss: 1.0335... Generator Loss: 0.8499
Epoch 1/2 batch 3400... Discriminator Loss: 1.1435... Generator Loss: 0.7066
Epoch 1/2 batch 3500... Discriminator Loss: 0.9470... Generator Loss: 0.9646
Epoch 1/2 batch 3600... Discriminator Loss: 1.0375... Generator Loss: 0.8048
Epoch 1/2 batch 3700... Discriminator Loss: 1.4081... Generator Loss: 0.5299
Epoch 1/2 batch 3800... Discriminator Loss: 0.9910... Generator Loss: 1.0318
Epoch 1/2 batch 3900... Discriminator Loss: 1.1311... Generator Loss: 0.6772
Epoch 1/2 batch 4000... Discriminator Loss: 1.0767... Generator Loss: 0.7927
Epoch 1/2 batch 4100... Discriminator Loss: 0.8447... Generator Loss: 1.1645
Epoch 1/2 batch 4200... Discriminator Loss: 0.9819... Generator Loss: 0.9927
Epoch 1/2 batch 4300... Discriminator Loss: 0.8791... Generator Loss: 1.3523
Epoch 1/2 batch 4400... Discriminator Loss: 1.1206... Generator Loss: 0.8176
Epoch 1/2 batch 4500... Discriminator Loss: 1.2163... Generator Loss: 0.5978
Epoch 1/2 batch 4600... Discriminator Loss: 0.9421... Generator Loss: 1.1175
Epoch 1/2 batch 4700... Discriminator Loss: 0.9188... Generator Loss: 1.0688
Epoch 1/2 batch 4800... Discriminator Loss: 1.8791... Generator Loss: 0.3082
Epoch 1/2 batch 4900... Discriminator Loss: 1.1410... Generator Loss: 0.8611
Epoch 1/2 batch 5000... Discriminator Loss: 1.2669... Generator Loss: 0.5328
Epoch 1/2 batch 5100... Discriminator Loss: 0.7664... Generator Loss: 1.7048
Epoch 1/2 batch 5200... Discriminator Loss: 1.0096... Generator Loss: 0.8525
Epoch 1/2 batch 5300... Discriminator Loss: 1.2746... Generator Loss: 0.6068
Epoch 1/2 batch 5400... Discriminator Loss: 1.0933... Generator Loss: 0.8574
Epoch 1/2 batch 5500... Discriminator Loss: 1.2650... Generator Loss: 0.6406
Epoch 1/2 batch 5600... Discriminator Loss: 1.0755... Generator Loss: 0.7276
Epoch 1/2 batch 5700... Discriminator Loss: 1.1093... Generator Loss: 0.9769
Epoch 1/2 batch 5800... Discriminator Loss: 1.0757... Generator Loss: 0.7405
Epoch 1/2 batch 5900... Discriminator Loss: 1.3695... Generator Loss: 0.4964
Epoch 1/2 batch 6000... Discriminator Loss: 1.0815... Generator Loss: 0.7260
Epoch 2/2 batch 6100... Discriminator Loss: 1.1787... Generator Loss: 0.6195
Epoch 2/2 batch 6200... Discriminator Loss: 1.0035... Generator Loss: 0.8209
Epoch 2/2 batch 6300... Discriminator Loss: 1.3186... Generator Loss: 0.5390
Epoch 2/2 batch 6400... Discriminator Loss: 1.3125... Generator Loss: 0.5291
Epoch 2/2 batch 6500... Discriminator Loss: 0.9589... Generator Loss: 1.0112
Epoch 2/2 batch 6600... Discriminator Loss: 1.5463... Generator Loss: 0.4200
Epoch 2/2 batch 6700... Discriminator Loss: 1.1023... Generator Loss: 0.7662
Epoch 2/2 batch 6800... Discriminator Loss: 1.0584... Generator Loss: 0.7864
Epoch 2/2 batch 6900... Discriminator Loss: 1.1325... Generator Loss: 0.7401
Epoch 2/2 batch 7000... Discriminator Loss: 1.2239... Generator Loss: 0.6076
Epoch 2/2 batch 7100... Discriminator Loss: 1.3069... Generator Loss: 0.5198
Epoch 2/2 batch 7200... Discriminator Loss: 1.0875... Generator Loss: 0.6984
Epoch 2/2 batch 7300... Discriminator Loss: 1.2818... Generator Loss: 0.5590
Epoch 2/2 batch 7400... Discriminator Loss: 1.0870... Generator Loss: 0.8578
Epoch 2/2 batch 7500... Discriminator Loss: 1.0629... Generator Loss: 1.4216
Epoch 2/2 batch 7600... Discriminator Loss: 0.9137... Generator Loss: 1.0594
Epoch 2/2 batch 7700... Discriminator Loss: 1.1244... Generator Loss: 0.7192
Epoch 2/2 batch 7800... Discriminator Loss: 1.7521... Generator Loss: 0.3595
Epoch 2/2 batch 7900... Discriminator Loss: 1.0527... Generator Loss: 0.7711
Epoch 2/2 batch 8000... Discriminator Loss: 1.1116... Generator Loss: 0.7884
Epoch 2/2 batch 8100... Discriminator Loss: 1.3045... Generator Loss: 0.5273
Epoch 2/2 batch 8200... Discriminator Loss: 1.4329... Generator Loss: 0.5985
Epoch 2/2 batch 8300... Discriminator Loss: 1.1261... Generator Loss: 0.6906
Epoch 2/2 batch 8400... Discriminator Loss: 1.2996... Generator Loss: 0.5546
Epoch 2/2 batch 8500... Discriminator Loss: 1.1625... Generator Loss: 0.6676
Epoch 2/2 batch 8600... Discriminator Loss: 0.9688... Generator Loss: 1.0352
Epoch 2/2 batch 8700... Discriminator Loss: 1.1707... Generator Loss: 0.8839
Epoch 2/2 batch 8800... Discriminator Loss: 1.0193... Generator Loss: 0.8621
Epoch 2/2 batch 8900... Discriminator Loss: 1.1260... Generator Loss: 0.7343
Epoch 2/2 batch 9000... Discriminator Loss: 1.3444... Generator Loss: 0.4853
Epoch 2/2 batch 9100... Discriminator Loss: 1.2939... Generator Loss: 0.5815
Epoch 2/2 batch 9200... Discriminator Loss: 1.3459... Generator Loss: 0.5790
Epoch 2/2 batch 9300... Discriminator Loss: 1.2500... Generator Loss: 0.6416
Epoch 2/2 batch 9400... Discriminator Loss: 0.9796... Generator Loss: 1.0356
Epoch 2/2 batch 9500... Discriminator Loss: 1.1047... Generator Loss: 0.7722
Epoch 2/2 batch 9600... Discriminator Loss: 1.0932... Generator Loss: 0.7045
Epoch 2/2 batch 9700... Discriminator Loss: 1.0777... Generator Loss: 0.7246
Epoch 2/2 batch 9800... Discriminator Loss: 1.0339... Generator Loss: 0.7850
Epoch 2/2 batch 9900... Discriminator Loss: 1.2448... Generator Loss: 0.6127
Epoch 2/2 batch 10000... Discriminator Loss: 1.1927... Generator Loss: 0.6743
Epoch 2/2 batch 10100... Discriminator Loss: 0.9101... Generator Loss: 1.1194
Epoch 2/2 batch 10200... Discriminator Loss: 1.0723... Generator Loss: 0.7304
Epoch 2/2 batch 10300... Discriminator Loss: 0.9820... Generator Loss: 0.9525
Epoch 2/2 batch 10400... Discriminator Loss: 1.0044... Generator Loss: 1.0101
Epoch 2/2 batch 10500... Discriminator Loss: 1.0557... Generator Loss: 0.8258
Epoch 2/2 batch 10600... Discriminator Loss: 1.1662... Generator Loss: 1.1679
Epoch 2/2 batch 10700... Discriminator Loss: 0.9784... Generator Loss: 0.9517
Epoch 2/2 batch 10800... Discriminator Loss: 1.2835... Generator Loss: 0.5464
Epoch 2/2 batch 10900... Discriminator Loss: 1.3608... Generator Loss: 1.2470
Epoch 2/2 batch 11000... Discriminator Loss: 1.5128... Generator Loss: 0.4114
Epoch 2/2 batch 11100... Discriminator Loss: 1.2801... Generator Loss: 0.5226
Epoch 2/2 batch 11200... Discriminator Loss: 1.0780... Generator Loss: 0.7420
Epoch 2/2 batch 11300... Discriminator Loss: 1.5325... Generator Loss: 0.4238
Epoch 2/2 batch 11400... Discriminator Loss: 1.0779... Generator Loss: 0.8510
Epoch 2/2 batch 11500... Discriminator Loss: 0.8783... Generator Loss: 1.1459
Epoch 2/2 batch 11600... Discriminator Loss: 1.3600... Generator Loss: 0.5028
Epoch 2/2 batch 11700... Discriminator Loss: 1.3402... Generator Loss: 0.5630
Epoch 2/2 batch 11800... Discriminator Loss: 1.1713... Generator Loss: 0.8321
Epoch 2/2 batch 11900... Discriminator Loss: 1.6624... Generator Loss: 0.3572
Epoch 2/2 batch 12000... Discriminator Loss: 1.0262... Generator Loss: 1.0220
```

1.3.4 CelebA

Run your GANs on CelebA. It will take around 20 minutes on the average GPU to run one epoch. You can run the whole epoch or stop when it starts to generate realistic faces.

```
In [52]: batch_size = 10
         z_dim = 10
         learning_rate = 0.001
         beta1 = 0.5


         """
         DON'T MODIFY ANYTHING IN THIS CELL THAT IS BELOW THIS LINE
         """
         epochs = 1

         celeba_dataset = helper.Dataset('celeba', glob(os.path.join(data_dir, 'img_align_celeba
         with tf.Graph().as_default():
             train(epochs, batch_size, z_dim, learning_rate, beta1, celeba_dataset.get_batches,
                   celeba_dataset.shape, celeba_dataset.image_mode)
```

```
Epoch 1/1 batch 100... Discriminator Loss: 1.4518... Generator Loss: 0.5030
Epoch 1/1 batch 200... Discriminator Loss: 1.2947... Generator Loss: 0.6405
Epoch 1/1 batch 300... Discriminator Loss: 1.3991... Generator Loss: 0.5135
Epoch 1/1 batch 400... Discriminator Loss: 1.3152... Generator Loss: 0.7704
Epoch 1/1 batch 500... Discriminator Loss: 1.2240... Generator Loss: 0.5945
Epoch 1/1 batch 600... Discriminator Loss: 1.3971... Generator Loss: 0.5664
Epoch 1/1 batch 700... Discriminator Loss: 1.2204... Generator Loss: 0.5688
Epoch 1/1 batch 800... Discriminator Loss: 1.3777... Generator Loss: 0.6233
Epoch 1/1 batch 900... Discriminator Loss: 1.2850... Generator Loss: 0.7947
Epoch 1/1 batch 1000... Discriminator Loss: 1.0381... Generator Loss: 0.8695
Epoch 1/1 batch 1100... Discriminator Loss: 1.1654... Generator Loss: 0.8101
Epoch 1/1 batch 1200... Discriminator Loss: 1.2645... Generator Loss: 0.5823
Epoch 1/1 batch 1300... Discriminator Loss: 1.0629... Generator Loss: 0.9400
Epoch 1/1 batch 1400... Discriminator Loss: 1.4107... Generator Loss: 0.5463
Epoch 1/1 batch 1500... Discriminator Loss: 1.1793... Generator Loss: 0.8156
Epoch 1/1 batch 1600... Discriminator Loss: 1.2861... Generator Loss: 0.5518
Epoch 1/1 batch 1700... Discriminator Loss: 1.2000... Generator Loss: 0.8441
Epoch 1/1 batch 1800... Discriminator Loss: 1.1784... Generator Loss: 0.9347
Epoch 1/1 batch 1900... Discriminator Loss: 1.1997... Generator Loss: 0.8119
Epoch 1/1 batch 2000... Discriminator Loss: 1.2366... Generator Loss: 1.0621
Epoch 1/1 batch 2100... Discriminator Loss: 1.1954... Generator Loss: 0.8081
Epoch 1/1 batch 2200... Discriminator Loss: 1.2167... Generator Loss: 0.8014
Epoch 1/1 batch 2300... Discriminator Loss: 1.1580... Generator Loss: 0.9137
Epoch 1/1 batch 2400... Discriminator Loss: 1.4368... Generator Loss: 0.7410
Epoch 1/1 batch 2500... Discriminator Loss: 1.1658... Generator Loss: 0.7979
Epoch 1/1 batch 2600... Discriminator Loss: 1.3494... Generator Loss: 0.7126
Epoch 1/1 batch 2700... Discriminator Loss: 1.2583... Generator Loss: 0.6873
Epoch 1/1 batch 2800... Discriminator Loss: 1.5089... Generator Loss: 0.4033
Epoch 1/1 batch 2900... Discriminator Loss: 1.4176... Generator Loss: 0.4521
Epoch 1/1 batch 3000... Discriminator Loss: 1.3115... Generator Loss: 0.6286
Epoch 1/1 batch 3100... Discriminator Loss: 1.0428... Generator Loss: 0.8695
Epoch 1/1 batch 3200... Discriminator Loss: 1.1681... Generator Loss: 0.6634
Epoch 1/1 batch 3300... Discriminator Loss: 1.2395... Generator Loss: 0.6535
Epoch 1/1 batch 3400... Discriminator Loss: 1.1512... Generator Loss: 0.7463
Epoch 1/1 batch 3500... Discriminator Loss: 1.0230... Generator Loss: 0.8754
Epoch 1/1 batch 3600... Discriminator Loss: 1.3048... Generator Loss: 0.6681
Epoch 1/1 batch 3700... Discriminator Loss: 1.4227... Generator Loss: 0.4585
Epoch 1/1 batch 3800... Discriminator Loss: 1.1900... Generator Loss: 0.7301
Epoch 1/1 batch 3900... Discriminator Loss: 1.1611... Generator Loss: 0.6672
Epoch 1/1 batch 4000... Discriminator Loss: 1.2233... Generator Loss: 0.8298
Epoch 1/1 batch 4100... Discriminator Loss: 1.5392... Generator Loss: 0.3856
Epoch 1/1 batch 4200... Discriminator Loss: 1.1672... Generator Loss: 0.7501
Epoch 1/1 batch 4300... Discriminator Loss: 1.5527... Generator Loss: 0.4466
Epoch 1/1 batch 4400... Discriminator Loss: 1.1235... Generator Loss: 0.7413
Epoch 1/1 batch 4500... Discriminator Loss: 1.3156... Generator Loss: 0.5666
Epoch 1/1 batch 4600... Discriminator Loss: 1.3068... Generator Loss: 0.6125
Epoch 1/1 batch 4700... Discriminator Loss: 1.1604... Generator Loss: 0.8581
Epoch 1/1 batch 4800... Discriminator Loss: 1.2025... Generator Loss: 0.6685
Epoch 1/1 batch 4900... Discriminator Loss: 1.3145... Generator Loss: 0.5305
Epoch 1/1 batch 5000... Discriminator Loss: 1.0593... Generator Loss: 0.7934
Epoch 1/1 batch 5100... Discriminator Loss: 1.0146... Generator Loss: 0.9371
Epoch 1/1 batch 5200... Discriminator Loss: 1.4040... Generator Loss: 0.4737
Epoch 1/1 batch 5300... Discriminator Loss: 1.3972... Generator Loss: 0.4669
Epoch 1/1 batch 5400... Discriminator Loss: 1.3075... Generator Loss: 0.5326
Epoch 1/1 batch 5500... Discriminator Loss: 1.1474... Generator Loss: 0.7197
Epoch 1/1 batch 5600... Discriminator Loss: 1.2552... Generator Loss: 0.5749
Epoch 1/1 batch 5700... Discriminator Loss: 1.2458... Generator Loss: 0.5999
Epoch 1/1 batch 5800... Discriminator Loss: 1.1727... Generator Loss: 0.6858
Epoch 1/1 batch 5900... Discriminator Loss: 1.1302... Generator Loss: 0.8831
Epoch 1/1 batch 6000... Discriminator Loss: 1.3122... Generator Loss: 0.5844
Epoch 1/1 batch 6100... Discriminator Loss: 1.1496... Generator Loss: 0.6897
Epoch 1/1 batch 6200... Discriminator Loss: 1.2400... Generator Loss: 0.6283
Epoch 1/1 batch 6300... Discriminator Loss: 1.0629... Generator Loss: 1.0641
Epoch 1/1 batch 6400... Discriminator Loss: 1.2356... Generator Loss: 0.6696
Epoch 1/1 batch 6500... Discriminator Loss: 1.3220... Generator Loss: 0.5350
Epoch 1/1 batch 6600... Discriminator Loss: 1.2938... Generator Loss: 0.6353
Epoch 1/1 batch 6700... Discriminator Loss: 1.2078... Generator Loss: 0.5827
Epoch 1/1 batch 6800... Discriminator Loss: 1.4394... Generator Loss: 0.4654
Epoch 1/1 batch 6900... Discriminator Loss: 1.0443... Generator Loss: 0.8153
Epoch 1/1 batch 7000... Discriminator Loss: 1.3070... Generator Loss: 0.6842
Epoch 1/1 batch 7100... Discriminator Loss: 1.1985... Generator Loss: 0.5982
Epoch 1/1 batch 7200... Discriminator Loss: 1.1427... Generator Loss: 0.6938
Epoch 1/1 batch 7300... Discriminator Loss: 1.2015... Generator Loss: 0.5940
Epoch 1/1 batch 7400... Discriminator Loss: 1.0895... Generator Loss: 0.8282
Epoch 1/1 batch 7500... Discriminator Loss: 1.1262... Generator Loss: 0.6546
Epoch 1/1 batch 7600... Discriminator Loss: 1.3608... Generator Loss: 0.4741
Epoch 1/1 batch 7700... Discriminator Loss: 1.1429... Generator Loss: 0.8368
Epoch 1/1 batch 7800... Discriminator Loss: 1.4830... Generator Loss: 0.4223
Epoch 1/1 batch 7900... Discriminator Loss: 1.3208... Generator Loss: 0.5390
Epoch 1/1 batch 8000... Discriminator Loss: 1.3424... Generator Loss: 0.4968
Epoch 1/1 batch 8100... Discriminator Loss: 1.5574... Generator Loss: 0.3738
Epoch 1/1 batch 8200... Discriminator Loss: 1.3273... Generator Loss: 0.4988
Epoch 1/1 batch 8300... Discriminator Loss: 1.3117... Generator Loss: 0.5297
Epoch 1/1 batch 8400... Discriminator Loss: 1.1721... Generator Loss: 0.7253
Epoch 1/1 batch 8500... Discriminator Loss: 1.2796... Generator Loss: 0.9326
Epoch 1/1 batch 8600... Discriminator Loss: 1.0880... Generator Loss: 0.7710
Epoch 1/1 batch 8700... Discriminator Loss: 1.1738... Generator Loss: 0.7001
Epoch 1/1 batch 8800... Discriminator Loss: 1.3867... Generator Loss: 0.4561
Epoch 1/1 batch 8900... Discriminator Loss: 1.0859... Generator Loss: 0.7758
Epoch 1/1 batch 9000... Discriminator Loss: 1.0463... Generator Loss: 0.8541
Epoch 1/1 batch 9100... Discriminator Loss: 1.2322... Generator Loss: 0.5625
Epoch 1/1 batch 9200... Discriminator Loss: 1.1784... Generator Loss: 0.6928
Epoch 1/1 batch 9300... Discriminator Loss: 1.4363... Generator Loss: 0.4366
Epoch 1/1 batch 9400... Discriminator Loss: 0.9825... Generator Loss: 0.9727
Epoch 1/1 batch 9500... Discriminator Loss: 1.0872... Generator Loss: 0.7801
Epoch 1/1 batch 9600... Discriminator Loss: 1.3542... Generator Loss: 0.5074
Epoch 1/1 batch 9700... Discriminator Loss: 1.3030... Generator Loss: 0.5678
Epoch 1/1 batch 9800... Discriminator Loss: 1.1178... Generator Loss: 0.7103
Epoch 1/1 batch 9900... Discriminator Loss: 1.1706... Generator Loss: 0.7964
Epoch 1/1 batch 10000... Discriminator Loss: 1.2505... Generator Loss: 0.6221
Epoch 1/1 batch 10100... Discriminator Loss: 1.0397... Generator Loss: 0.9561
Epoch 1/1 batch 10200... Discriminator Loss: 1.0904... Generator Loss: 0.7029
Epoch 1/1 batch 10300... Discriminator Loss: 1.1373... Generator Loss: 0.7370
Epoch 1/1 batch 10400... Discriminator Loss: 1.1239... Generator Loss: 0.8457
Epoch 1/1 batch 10500... Discriminator Loss: 0.9702... Generator Loss: 0.9053
Epoch 1/1 batch 10600... Discriminator Loss: 1.0001... Generator Loss: 1.0773
Epoch 1/1 batch 10700... Discriminator Loss: 1.3290... Generator Loss: 0.4850
Epoch 1/1 batch 10800... Discriminator Loss: 1.1995... Generator Loss: 0.6574
Epoch 1/1 batch 10900... Discriminator Loss: 1.2705... Generator Loss: 0.5784
Epoch 1/1 batch 11000... Discriminator Loss: 1.1507... Generator Loss: 0.7113
Epoch 1/1 batch 11100... Discriminator Loss: 1.1695... Generator Loss: 0.6296
Epoch 1/1 batch 11200... Discriminator Loss: 1.2028... Generator Loss: 0.6090
Epoch 1/1 batch 11300... Discriminator Loss: 1.0672... Generator Loss: 0.8064
Epoch 1/1 batch 11400... Discriminator Loss: 1.0744... Generator Loss: 0.9813
Epoch 1/1 batch 11500... Discriminator Loss: 1.2361... Generator Loss: 0.6226
Epoch 1/1 batch 11600... Discriminator Loss: 1.1588... Generator Loss: 0.8777
Epoch 1/1 batch 11700... Discriminator Loss: 1.0044... Generator Loss: 0.9526
Epoch 1/1 batch 11800... Discriminator Loss: 1.0636... Generator Loss: 0.9549
Epoch 1/1 batch 11900... Discriminator Loss: 1.1122... Generator Loss: 0.7791
Epoch 1/1 batch 12000... Discriminator Loss: 0.9627... Generator Loss: 0.9537
Epoch 1/1 batch 12100... Discriminator Loss: 1.7299... Generator Loss: 0.3464
Epoch 1/1 batch 12200... Discriminator Loss: 1.2875... Generator Loss: 0.5402
Epoch 1/1 batch 12300... Discriminator Loss: 1.2278... Generator Loss: 0.6930
Epoch 1/1 batch 12400... Discriminator Loss: 0.9943... Generator Loss: 0.8333
Epoch 1/1 batch 12500... Discriminator Loss: 1.1844... Generator Loss: 0.8529
Epoch 1/1 batch 12600... Discriminator Loss: 1.1520... Generator Loss: 0.6768
Epoch 1/1 batch 12700... Discriminator Loss: 1.0450... Generator Loss: 0.8862
Epoch 1/1 batch 12800... Discriminator Loss: 1.2139... Generator Loss: 0.6401
Epoch 1/1 batch 12900... Discriminator Loss: 0.9975... Generator Loss: 0.9049
Epoch 1/1 batch 13000... Discriminator Loss: 1.0615... Generator Loss: 0.8917
Epoch 1/1 batch 13100... Discriminator Loss: 1.0883... Generator Loss: 0.6983
Epoch 1/1 batch 13200... Discriminator Loss: 1.2102... Generator Loss: 0.6002
Epoch 1/1 batch 13300... Discriminator Loss: 1.3336... Generator Loss: 0.5045
Epoch 1/1 batch 13400... Discriminator Loss: 1.0825... Generator Loss: 0.7827
Epoch 1/1 batch 13500... Discriminator Loss: 0.9726... Generator Loss: 0.9367
Epoch 1/1 batch 13600... Discriminator Loss: 1.2872... Generator Loss: 0.5600
Epoch 1/1 batch 13700... Discriminator Loss: 1.1221... Generator Loss: 0.7125
Epoch 1/1 batch 13800... Discriminator Loss: 1.2494... Generator Loss: 0.6135
Epoch 1/1 batch 13900... Discriminator Loss: 1.2141... Generator Loss: 0.6396
Epoch 1/1 batch 14000... Discriminator Loss: 1.1282... Generator Loss: 0.8427
Epoch 1/1 batch 14100... Discriminator Loss: 1.2036... Generator Loss: 0.6998
Epoch 1/1 batch 14200... Discriminator Loss: 1.1529... Generator Loss: 0.7039
Epoch 1/1 batch 14300... Discriminator Loss: 1.1708... Generator Loss: 0.6702
Epoch 1/1 batch 14400... Discriminator Loss: 1.1473... Generator Loss: 0.7449
Epoch 1/1 batch 14500... Discriminator Loss: 1.3820... Generator Loss: 0.5124
Epoch 1/1 batch 14600... Discriminator Loss: 1.0627... Generator Loss: 0.9167
Epoch 1/1 batch 14700... Discriminator Loss: 1.0783... Generator Loss: 0.7720
Epoch 1/1 batch 14800... Discriminator Loss: 1.2221... Generator Loss: 0.5862
Epoch 1/1 batch 14900... Discriminator Loss: 1.4057... Generator Loss: 0.5514
Epoch 1/1 batch 15000... Discriminator Loss: 1.3487... Generator Loss: 0.4999
Epoch 1/1 batch 15100... Discriminator Loss: 1.0399... Generator Loss: 0.8219
Epoch 1/1 batch 15200... Discriminator Loss: 1.0374... Generator Loss: 0.9031
Epoch 1/1 batch 15300... Discriminator Loss: 1.6906... Generator Loss: 0.3558
Epoch 1/1 batch 15400... Discriminator Loss: 1.1468... Generator Loss: 0.6751
Epoch 1/1 batch 15500... Discriminator Loss: 1.2136... Generator Loss: 0.5808
Epoch 1/1 batch 15600... Discriminator Loss: 1.1346... Generator Loss: 0.7763
Epoch 1/1 batch 15700... Discriminator Loss: 1.0840... Generator Loss: 0.7175
Epoch 1/1 batch 15800... Discriminator Loss: 1.3420... Generator Loss: 0.5770
Epoch 1/1 batch 15900... Discriminator Loss: 1.3186... Generator Loss: 0.5232
Epoch 1/1 batch 16000... Discriminator Loss: 1.2274... Generator Loss: 0.6551
Epoch 1/1 batch 16100... Discriminator Loss: 1.1445... Generator Loss: 0.7296
Epoch 1/1 batch 16200... Discriminator Loss: 1.4852... Generator Loss: 0.4352
Epoch 1/1 batch 16300... Discriminator Loss: 1.1951... Generator Loss: 0.7476
Epoch 1/1 batch 16400... Discriminator Loss: 1.1776... Generator Loss: 0.6652
Epoch 1/1 batch 16500... Discriminator Loss: 1.2439... Generator Loss: 0.5919
Epoch 1/1 batch 16600... Discriminator Loss: 1.2269... Generator Loss: 0.7251
Epoch 1/1 batch 16700... Discriminator Loss: 1.1846... Generator Loss: 0.7148
Epoch 1/1 batch 16800... Discriminator Loss: 1.1677... Generator Loss: 0.6436
Epoch 1/1 batch 16900... Discriminator Loss: 1.3108... Generator Loss: 0.5661
Epoch 1/1 batch 17000... Discriminator Loss: 1.1591... Generator Loss: 0.6586
Epoch 1/1 batch 17100... Discriminator Loss: 1.2589... Generator Loss: 0.5431
Epoch 1/1 batch 17200... Discriminator Loss: 1.0813... Generator Loss: 0.7415
Epoch 1/1 batch 17300... Discriminator Loss: 1.1269... Generator Loss: 0.7494
Epoch 1/1 batch 17400... Discriminator Loss: 1.2033... Generator Loss: 0.6792
Epoch 1/1 batch 17500... Discriminator Loss: 1.0750... Generator Loss: 0.8027
Epoch 1/1 batch 17600... Discriminator Loss: 1.2177... Generator Loss: 0.6704
Epoch 1/1 batch 17700... Discriminator Loss: 1.1313... Generator Loss: 0.7399
Epoch 1/1 batch 17800... Discriminator Loss: 1.1676... Generator Loss: 0.6542
Epoch 1/1 batch 17900... Discriminator Loss: 1.2342... Generator Loss: 0.5688
Epoch 1/1 batch 18000... Discriminator Loss: 1.1360... Generator Loss: 0.7992
Epoch 1/1 batch 18100... Discriminator Loss: 1.1233... Generator Loss: 0.7286
Epoch 1/1 batch 18200... Discriminator Loss: 1.1703... Generator Loss: 0.6386
Epoch 1/1 batch 18300... Discriminator Loss: 1.2807... Generator Loss: 0.6158
Epoch 1/1 batch 18400... Discriminator Loss: 0.9692... Generator Loss: 0.9347
Epoch 1/1 batch 18500... Discriminator Loss: 1.2471... Generator Loss: 0.5939
Epoch 1/1 batch 18600... Discriminator Loss: 1.3700... Generator Loss: 0.4690
Epoch 1/1 batch 18700... Discriminator Loss: 1.1420... Generator Loss: 0.7758
Epoch 1/1 batch 18800... Discriminator Loss: 1.2378... Generator Loss: 0.6344
Epoch 1/1 batch 18900... Discriminator Loss: 1.0518... Generator Loss: 0.9917
Epoch 1/1 batch 19000... Discriminator Loss: 1.3019... Generator Loss: 0.5202
Epoch 1/1 batch 19100... Discriminator Loss: 1.2478... Generator Loss: 0.5893
Epoch 1/1 batch 19200... Discriminator Loss: 1.1064... Generator Loss: 0.9350
Epoch 1/1 batch 19300... Discriminator Loss: 1.2153... Generator Loss: 0.6657
Epoch 1/1 batch 19400... Discriminator Loss: 1.0102... Generator Loss: 0.8658
Epoch 1/1 batch 19500... Discriminator Loss: 1.0180... Generator Loss: 0.9146
Epoch 1/1 batch 19600... Discriminator Loss: 1.1114... Generator Loss: 0.7096
Epoch 1/1 batch 19700... Discriminator Loss: 1.0302... Generator Loss: 0.9639
Epoch 1/1 batch 19800... Discriminator Loss: 0.9896... Generator Loss: 1.0479
Epoch 1/1 batch 19900... Discriminator Loss: 1.4311... Generator Loss: 0.4991
Epoch 1/1 batch 20000... Discriminator Loss: 1.1613... Generator Loss: 0.6443
Epoch 1/1 batch 20100... Discriminator Loss: 1.0343... Generator Loss: 0.8532
Epoch 1/1 batch 20200... Discriminator Loss: 1.3041... Generator Loss: 0.5365
```

1.3.5 Submitting This Project

When submitting this project, make sure to run all the cells before saving the notebook. Save the notebook file as "dlnd_face_generation.ipynb" and save it as a HTML file under "File" -> "Download as". Include the "helper.py" and "problem_unittests.py" files in your submission.