UDACITY

PROJECT

Generate Faces

A part of the Deep Learning Nanodegree Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Meets Specifications

This is a very good submission!

In order to get good performance with image generation, there are several different implementations and tricks, which you can check out from https://github.com/soumith/ganhacks

Here are some other important resources for GAN:

http://www.araya.org/archives/1183 for GAN stability.
https://github.com/yihui-he/GAN-MNIST, https://github.com/carpedm20/DCGAN-tensorflow for DCGAN.
https://medium.com/@ageitgey/abusing-generative-adversarial-networks-to-make-8-bit-pixel-art-e45d9b96cee7
It can be seen that a lot of hard work has been put into this.

Congratulations on successfully completing this project!

Required Files and Tests

✓

The project submission contains the project notebook, called "dlnd_face_generation.ipynb".

✓

All the unit tests in project have passed.

Build the Neural Network

✓

The function model_inputs is implemented correctly.

✓

The function discriminator is implemented correctly.

Good work using Batch Normalization and Leaky ReLUs which allow a small non zero gradient when the unit is not active.
Try using different values of alpha parameter between 0.08 and 0.15 and compare your results.

✓

The function generator is implemented correctly.

Good work using Batch Normalization and Leaky ReLUs which allow a small non zero gradient when the unit is not active.
Try using different values of alpha parameter between 0.08 and 0.15 and compare your results.
Since tanh is the last layer of the generator network normalizing the input images is a good step.

✓

The function model_loss is implemented correctly.

Good work using smoothing as it prevents discriminator from being too strong and to generalize in a better way.

✓

The function model_opt is implemented correctly.

Great work updating the training step with dependency to tf.GraphKeys.UPDATE_OPS.

Neural Network Training

✓

The function train is implemented correctly.

- It should build the model using `model_inputs`, `model_loss`, and `model_opt`.
- It should show output of the `generator` using the `show_generator_output` function

Good work keeping batch_z between -1 and 1.
Good work increasing batch size by a factor of two inside the inner for loop.

✓

The parameters are set reasonable numbers.

Good work adjusting the hyper-parameters.

- Try using Batch size as 32 or 64.
- Try using different values of learning rate between 0.0002 and 0.0008 and different values of beta1 between 0.2 and 0.5 and compare your results.

✓

The project generates realistic faces. It should be obvious that images generated look like faces.

The faces generated are quite clear and realistic. To generate clear faces right after one epoch, please follow the suggestions mentioned above.

DOWNLOAD PROJECT

RETURN TO PATH

Student FAQ