UDACITY

‹ Return to Classroom

DISCUSS ON STUDENT HUB

Generate Faces

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Dear excellent student:
This is a very high quality submission. It demonstrates your mastery of GAN. I wish you good luck in the future projects and work!
Here is the great resources as the suggestions for how to train GAN.

Required Files and Tests

The project submission contains the project notebook, called “dlnd_face_generation.ipynb”.

All the unit tests in project have passed.

Data Loading and Processing

The function `get_dataloader` should transform image data into resized, Tensor image types and return a DataLoader that batches all the training data into an appropriate size.

Pre-process the images by creating a `scale` function that scales images into a given pixel range. This function should be used later, in the training loop.

Build the Adversarial Networks

The Discriminator class is implemented correctly; it outputs one value that will determine whether an image is real or fake.

The Generator class is implemented correctly; it outputs an image of the same shape as the processed training data.

This function should initialize the weights of any convolutional or linear layer with weights taken from a normal distribution with a mean = 0 and standard deviation = 0.02.

Weights are initialized correctly.

Optimization Strategy

The loss functions take in the outputs from a discriminator and return the real or fake loss.

Good job using one-sided label smoothing for the discriminator real loss as discussed in this paper.

There are optimizers for updating the weights of the discriminator and generator. These optimizers should have appropriate hyperparameters.

Training and Results

Real training images should be scaled appropriately. The training loop should alternate between training the discriminator and generator networks.

There is not an exact answer here, but the models should be deep enough to recognize facial features and the optimizers should have parameters that help wth model convergence.

You have correctly pieced together all the components to create the model. Also, you have correctly scaled the input images to the same range as the generated images. Well Done !!! I like the idea of executing generator multiple times for one execution of discriminator. I would strongly recommend executing the generator at most 2 or 3 times.

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

Faces generated are real enough!

The question about model improvement is answered.

Great thinking here, AI bias is always an issue to keep in mind when we prepare our training data. Also, learning rate scheduling is a good way to improve model performance with minimum effort.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

START