UDACITY

‹ Return to "Deep Learning" in the classroom

DISCUSS ON STUDENT HUB

Generate Faces

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Congratulations on completing this project! You did a great job so far.
I'd like to suggest you watch this video and the corresponding document. It has some great advice to improve DCGANs performance.

Required Files and Tests

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

Perfect job. Your notebook is named as required.

All the unit tests in project have passed.

All the test units are passing perfectly. Well done.

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.

Good job loading image data.

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.

Nice job with the scaling function.

Build the Adversarial Networks

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

Great job with the discriminator!

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

Great job with the generator as well.

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.

Great job initializing weights.

Optimization Strategy

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

Well done creating real and fake loss functions.

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

Well done.

Training and Results

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

Nice results on converging.

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.

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

Great job! Faces are clearly recognizable.

The question about model improvement is answered.

DOWNLOAD PROJECT

RETURN TO PATH