UDACITY

< Back to Deep Learning Nanodegree

Generate Faces

REVIEW

CODE REVIEW

HISTORY

Requires Changes

4 SPECIFICATIONS REQUIRE CHANGES

Required Files and Tests

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

All the files are correctly submitted.

Name	Type	Compressed size	Password p...	Size
dlnd_face_generation	Chrome HTML Document	1,612 KB	No	2,406 KB
dlnd_face_generation	IPYNB File	1,566 KB	No	2,096 KB
floyd_requirements	Text Document	1 KB	No	1 KB
helper	PY File	3 KB	No	8 KB
problem_unittests	PY File	2 KB	No	6 KB

All the unit tests in project have passed.

The unit tests are flawlessly running. Using unit tests is also considered as a great choice. These help us to validate the correctness of the logics

validate the correctness of the logics.

Build the Neural Network

The function model_inputs is implemented correctly.

The placeholders of `input_images` , `input_z` and `learning rate` are rightly created using the tf.placeholder() function.

The function discriminator is implemented correctly.

Good Job!

I appreciate your efforts. I liked the way you coded the `apply_conv()` function but there's an issue. Please consider using the batch normalization.

Required

- Make sure you add batch normalization rightly. This will improve your results alot.
- While creating the convolution layers for discriminator and deconvolution layers for generator, I recommend you to consider adding `Xavier initializer` as the kernel initializer. It will help to initialize weights correctly.
  To further read about it, consider going through this blog post:
  https://prateekvjoshi.com/2016/03/29/understanding-xavier-initialization-in-deep-neural-networks/
- To enhance the power of discriminator, you can try adding 1-2 more hidden layers. Try adding layers with stride=1 and padding=same to maintain the width and height.

The function generator is implemented correctly.

Amazing!

The generator function is perfectly coded up.
Leaky relu, batch normalization and layers are rightly used to build the network. 👏🏽

Suggestion

- As mentioned in the discriminator part, consider using `Xavier initializer` as the kernel intializer.
- Try adding 1 hidden layer to the network.

The function model_loss is implemented correctly.

Nice

The discriminator-generator losses are correctly computed. Although you can consider using the label smoothening as we did in Intro to GANs notebook.

The function model_opt is implemented correctly.

The optimizer part is well coded. You have used the Adam optimizer 👌

You can still try experimenting with other optimizers too like RMSprop, Adagrad etc.

Neural Network Training

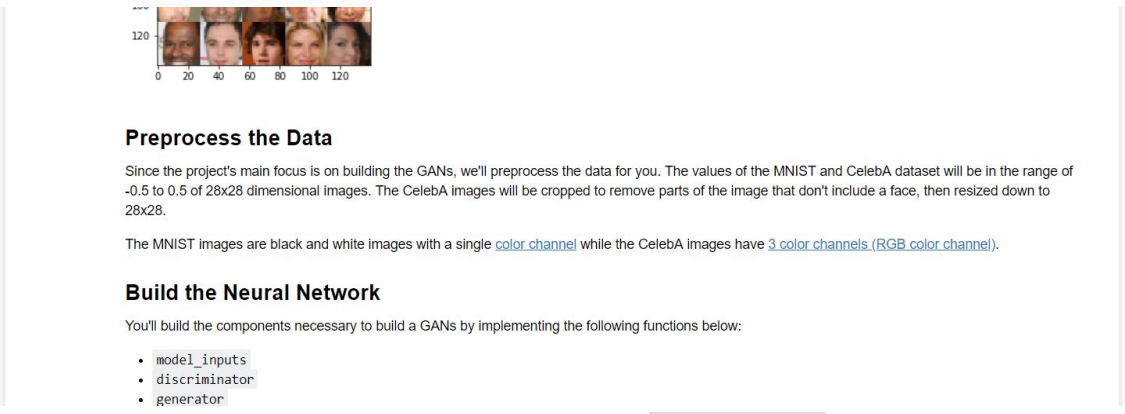
The function train is implemented correctly.

- 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

You have rightly build the network using the above functions. 👍

Required

It's required to rescale the images:



Please consider adding a rescaling step inside the for loop for `batch_images`.

The parameters are set reasonable numbers.

The parameters are set reasonable numbers.

MNIST

```
batch_size = 128
z_dim = 100
learning_rate = 0.0002
beta1 = 0.5
```

Celeb

```
batch_size = 128
z_dim = 100 * 3
learning_rate = 0.0002
beta1 = 0.5
```

The hyperparameter values are well chosen but there is still some space for improvement.

- The batch size should be chosen on the basis of system configuration of training instance. It's value as 128 is acceptable.
- The z_dim is random noise dimesnion. It should be chosen on the basis of complexity of dataset. You have rightly chosen the z_dim values for both the datasets
- The learning rate as 0.0002 seems a good choice but try to tune it a little bit.
- Beta1 i.e, decay rate as 0.5 seems quite high. Consider using a lower decay rate. It's noticed that network performs better with decay rate around 0.1-0.2

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

As you can see, currently the images are blur. Try to work on the things that I have mentioned above and retrain the model. You will surely be able to get great results.

Keep up the great work!

RESUBMIT

DOWNLOAD PROJECT

Best practices for your project resubmission

Ben shares 5 helpful tips to get you through revising and resubmitting your project.

Watch Video (3:01)

RETURN TO PATH

Rate this review