

PROJECT

Generate Faces

A part of the Deep Learning Nanodegree Foundation Program

PROJECT REVIEW

CODE REVIEW

NOTES

Requires Changes

SHARE YOUR ACCOMPLISHMENT



2 SPECIFICATIONS REQUIRE CHANGES

It's a great effort! I am sure that your next submission will be successful. Keep it up!

I suggest you check out the following Ian Goodfellow's [tutorial](#) on GANs. It has a number of great tips.

Congratulations on almost finishing Deep Learning Nanodegree Foundations program! I hope you've thoroughly enjoyed this journey :)

Required Files and Tests

✓	The project submission contains the project notebook, called "dlnd_face_generation.ipynb".
✓	All the unit tests in project have passed.
	Well done! All unit tests have passed!

Build the Neural Network

✓	The function model_inputs is implemented correctly.
✓	The function discriminator is implemented correctly.
	Well done on using batch normalization and a leaky ReLU (rather than a vanilla ReLU). This is important since it helps the gradient flow through the network, which in turn is crucial for the network's ability to learn.
	Kudos on factoring out the leaky ReLU functionality into a separate function.
✓	The function generator is implemented correctly.
	Perfect!
✓	The function model_loss is implemented correctly.
	Well done!
	I would recommend you to multiply <code>labels</code> (for <code>d_loss_real</code>) by a smoothing factor (0.9, for instance). This helps optimizing this loss for the following reason: initially the generator network does not produce anything close to the real input images; hence, the discriminator quickly learns to distinguish between real inputs and generated inputs - outputting a probability close to 1; hence cross-entropy loss will involve the following computation: <code>log(some_very_small_number)</code> , which can be unstable.
✓	The function model_opt is implemented correctly.

Neural Network Training

🔄	The function train is implemented correctly.
	<ul style="list-style-type: none">It should build the model using <code>model_inputs</code>, <code>model_loss</code>, and <code>model_opt</code>.It should show output of the <code>generator</code> using the <code>show_generator_output</code> function
	In the notebook there is a description of the "faces" dataset - it says that the values in the matrix lie in [-0.5, 0.5] range. But your generator produces output in the range [-1, 1] (due to application of <code>tf.tanh</code>). Hence, you need to multiply <code>batch_images</code> by 2 to achieve the same scale. This fix will DRAMATICALLY improve your model's performance.
✓	The parameters are set reasonable numbers.
	It's a great set of hyperparameters! I hope that you played with different values to see how it affects training.
🔄	The project generates realistic faces. It should be obvious that images generated look like faces.
	Once you fix a bug in <code>train</code> function, your should see an improvement in your model's performance.

🔄 RESUBMIT PROJECT

📄 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