

CONCEPTS



Doing More With Your GAN

- ✓ 1. Introducing Ian Goodfellow
- ✓ 2. What can you do with GANs?
- ✓ 3. How GANs work
- ✓ 4. Games and Equilibria
- ✓ 5. Practical tips and tricks for train...
- ✓ 6. Build a GAN
- ✓ 7. Get started with a GAN
- ✓ 8. Generator Network
- ✓ 9. Discriminator Network
- ✓ 10. Generator and Discriminator ...
- ✓ 11. Building the Network
- ✓ 12. Building the Network Solution
- ✓ 13. Training Losses
- ✓ 14. Training Optimizers
- ✓ 15. Training Losses and Optimizer...
- ✓ 16. A Trained GAN
- ✓ 17. Doing More With Your GAN



Knowledge

Get support and stay on track

Doing More with Your GAN

The GAN I showed you used only one hidden layer in the generator and discriminator. The results of this GAN were pretty impressive, but there were still a lot of noisy images and images that didn't really look like numbers. But, it's completely possible to have the generator make images indistinguishable from the MNIST dataset.



MNIST images generated from a trained GAN

(<https://arxiv.org/pdf/1606.03498.pdf>)

This is from a paper called [Improved Techniques for Training GANs](#). So, how did they make such nice looking images?

As with most neural networks, using more layers will improve performance. Instead of just one layer, try two or three. Try larger



Batch Normalization

Just as a heads up, you probably won't get it to work with three layers. The networks become really sensitive to the initial values of the weights and will fail to train. We can solve this problem using **batch normalization**. The idea is pretty simple. Just like what we do with the inputs to our networks, we can normalize the inputs to each layer. That is, scale the layer inputs to have zero mean and standard deviations of 1. It's been found that batch normalization is necessary to make deep GANs.

We'll cover batch normalization in next week's lesson, as well as using convolutional networks for the generator and discriminator.

[NEXT](#)**Knowledge**

Get support and stay on track