

Overview of GAN Structure ▢

A generative adversarial network (GAN) has two parts:

- The **generator** learns to generate plausible data. The generated instances become negative training examples for the discriminator.
- The **discriminator** learns to distinguish the generator's fake data from real data. The discriminator penalizes the generator for producing implausible results.

When training begins, the generator produces obviously fake data, and the discriminator quickly learns to tell that it's fake:



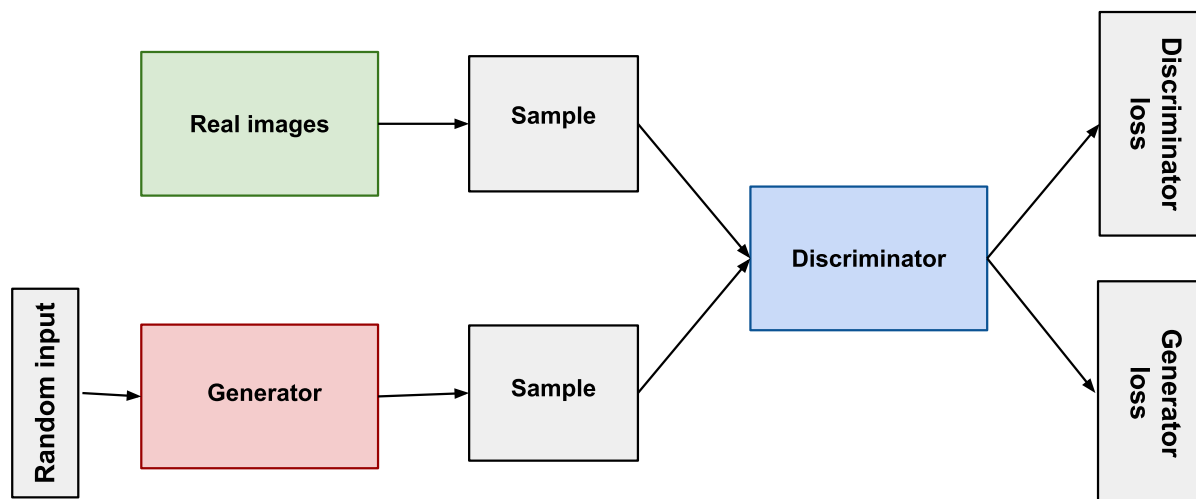
As training progresses, the generator gets closer to producing output that can fool the discriminator:



Finally, if generator training goes well, the discriminator gets worse at telling the difference between real and fake. It starts to classify fake data as real, and its accuracy decreases.



Here's a picture of the whole system:



Both the generator and the discriminator are neural networks. The generator output is connected directly to the discriminator input. Through [backpropagation](#) (/machine-learning/glossary#backpropagation), the discriminator's classification provides a signal that the generator uses to update its weights.

Let's explain the pieces of this system in greater detail.

[Previous](#)

← [Generative Models](#) (/machine-learning/gan/generative)

[Next](#)

[Discriminator](#) (/machine-learning/gan/discriminator)



Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](#) (<https://creativecommons.org/licenses/by/4.0/>), and code samples are licensed under the [Apache 2.0 License](#) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see the [Google Developers Site Policies](#) (<https://developers.google.com/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2019-05-24 UTC.