

[Click to Take the FREE GANs Crash-Course](#)

Search...



A Gentle Introduction to Generative Adversarial Networks (GANs)

by **Jason Brownlee** on June 17, 2019 in **Generative Adversarial Networks**

[Tweet](#)[Tweet](#)[Share](#)[Share](#)

Last Updated on July 19, 2019

Generative Adversarial Networks, or GANs for short, are an approach to generative modeling using deep learning methods, such as convolutional neural networks.

Generative modeling is an unsupervised learning task in machine learning that involves automatically discovering and learning the regularities or patterns in input data in such a way that the model can be used to generate or output new examples that plausibly could have been drawn from the original dataset.

GANs are a clever way of training a generative model by framing the problem as a supervised learning problem with two sub-models: the generator model that we train to generate new examples, and the discriminator model that tries to classify examples as either real (from the domain) or fake (generated). The two models are trained together in a zero-sum game, adversarial, until the discriminator model is fooled about half the time, meaning the generator model is generating plausible examples.

GANs are an exciting and rapidly changing field, delivering on the promise of generative models in their ability to generate realistic examples across a range of problem domains, most notably in image-to-image translation tasks such as translating photos of summer to winter or day to night, and in generating photorealistic photos of objects, scenes, and people that even humans cannot tell are fake.

In this post, you will discover a gentle introduction to Generative Adversarial Networks, or GANs.

After reading this post, you will know:

- Context for GANs, including supervised vs. unsupervised learning and discriminative vs. generative modeling.
- GANs are an architecture for automatically training a generative model by treating the unsupervised problem as supervised and using both a generative and a discriminative model.

- GANs provide a path to sophisticated domain-specific data augmentation and a solution to problems that require a generative solution, such as image-to-image translation.

Kick-start your project with my new book [Generative Adversarial Networks with Python](#), including *step-by-step tutorials* and the *Python source code* files for all examples.

Let's get started.



A Gentle Introduction to Generative Adversarial Networks (GANs)

Photo by [Barney Moss](#), some rights reserved.

Overview

This tutorial is divided into three parts; they are:

1. What Are Generative Models?
2. What Are Generative Adversarial Networks?
3. Why Generative Adversarial Networks?

What Are Generative Models?

In this section, we will review the idea of generative models, stepping over the supervised vs. unsupervised learning paradigms and discriminative vs. generative modeling.

Supervised vs. Unsupervised Learning

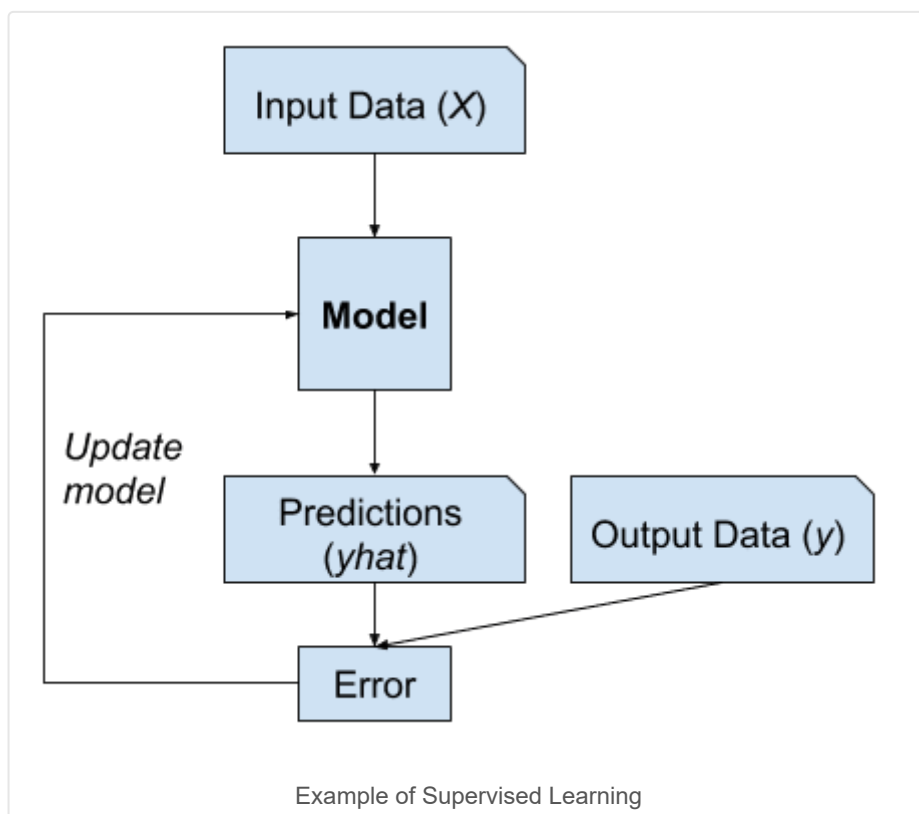
A typical machine learning problem involves using a model to make a prediction, e.g. [predictive modeling](#).

This requires a training dataset that is used to train a model, comprised of multiple examples, called samples, each with input variables (X) and output class labels (y). A model is trained by showing examples of inputs, having it predict outputs, and correcting the model to make the outputs more like the expected outputs.

“ In the predictive or supervised learning approach, the goal is to learn a mapping from inputs x to outputs y , given a labeled set of input-output pairs ...

— Page 2, [Machine Learning: A Probabilistic Perspective](#), 2012.

This correction of the model is generally referred to as a supervised form of learning, or supervised learning.



Examples of supervised learning problems include classification and regression, and examples of supervised learning algorithms include logistic regression and random forest.

There is another paradigm of learning where the model is only given the input variables (X) and the problem does not have any output variables (y).

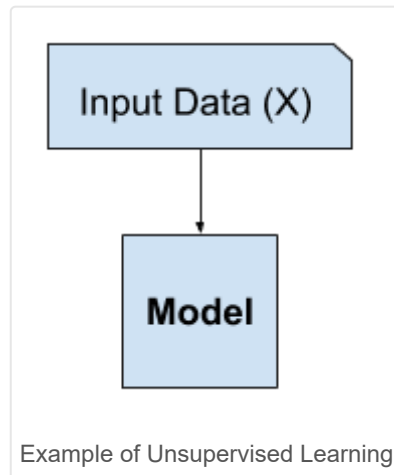
A model is constructed by extracting or summarizing the patterns in the input data. There is no correction of the model, as the model is not predicting anything.

“ The second main type of machine learning is the descriptive or unsupervised learning approach. Here we are only given inputs, and the goal is to find “interesting patterns” in the data. [...] This is a much less well-defined problem, since we are not told what kinds of

patterns to look for, and there is no obvious error metric to use (unlike supervised learning, where we can compare our prediction of y for a given x to the observed value).

— Page 2, [Machine Learning: A Probabilistic Perspective](#), 2012.

This lack of correction is generally referred to as an unsupervised form of learning, or unsupervised learning.



Examples of unsupervised learning problems include clustering and generative modeling, and examples of unsupervised learning algorithms are K-means and Generative Adversarial Networks.

Want to Develop GANs from Scratch?

Take my free 7-day email crash course now (with sample code).

Click to sign-up and also get a free PDF Ebook version of the course.

[Download Your FREE Mini-Course](#)

Discriminative vs. Generative Modeling

In supervised learning, we may be interested in developing a model to predict a class label given an example of input variables.

This predictive modeling task is called classification.

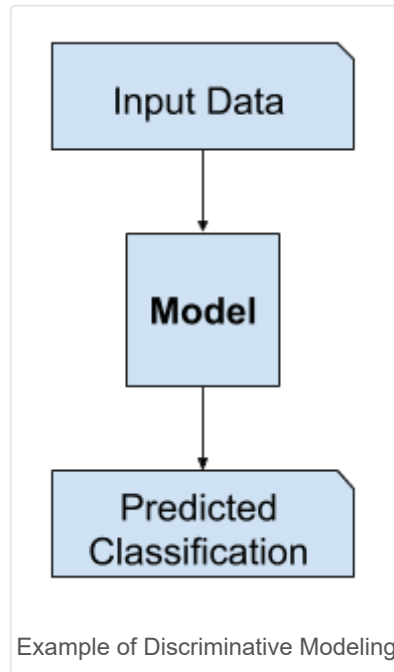
Classification is also traditionally referred to as discriminative modeling.

“... we use the training data to find a discriminant function $f(x)$ that maps each x directly onto a class label, thereby combining the inference and decision stages into a single learning

problem.

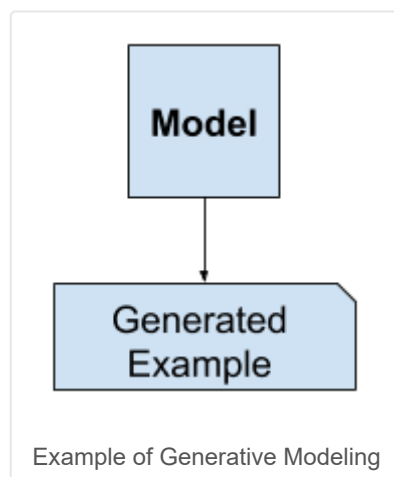
— Page 44, [Pattern Recognition and Machine Learning](#), 2006.

This is because a model must discriminate examples of input variables across classes; it must choose or make a decision as to what class a given example belongs.



Alternately, unsupervised models that summarize the distribution of input variables may be able to be used to create or generate new examples in the input distribution.

As such, these types of models are referred to as [generative models](#).



For example, a single variable may have a known data distribution, such as a [Gaussian distribution](#), or bell shape. A generative model may be able to sufficiently summarize this data distribution, and then be used to generate new variables that plausibly fit into the distribution of the input variable.



Approaches that explicitly or implicitly model the distribution of inputs as well as outputs are known as generative models, because by sampling from them it is possible to generate

synthetic data points in the input space.

— Page 43, [Pattern Recognition and Machine Learning](#), 2006.

In fact, a really good generative model may be able to generate new examples that are not just plausible, but indistinguishable from real examples from the problem domain.

Examples of Generative Models

[Naive Bayes](#) is an example of a generative model that is more often used as a discriminative model.

For example, Naive Bayes works by summarizing the probability distribution of each input variable and the output class. When a prediction is made, the probability for each possible outcome is calculated for each variable, the independent probabilities are combined, and the most likely outcome is predicted. Used in reverse, the probability distributions for each variable can be sampled to generate new plausible (independent) feature values.

Other examples of generative models include Latent Dirichlet Allocation, or LDA, and the Gaussian Mixture Model, or GMM.

Deep learning methods can be used as generative models. Two popular examples include the Restricted Boltzmann Machine, or RBM, and the Deep Belief Network, or DBN.

Two modern examples of deep learning generative modeling algorithms include the Variational Autoencoder, or VAE, and the Generative Adversarial Network, or GAN.

What Are Generative Adversarial Networks?

Generative Adversarial Networks, or GANs, are a deep-learning-based generative model.

More generally, GANs are a model architecture for training a generative model, and it is most common to use deep learning models in this architecture.

The GAN architecture was first described in the 2014 paper by [Ian Goodfellow](#), et al. titled “[Generative Adversarial Networks](#).”

A standardized approach called Deep Convolutional Generative Adversarial Networks, or DCGAN, that led to more stable models was later formalized by [Alec Radford](#), et al. in the 2015 paper titled “[Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks](#)”.

“Most GANs today are at least loosely based on the DCGAN architecture ...

— [NIPS 2016 Tutorial: Generative Adversarial Networks](#), 2016.

The GAN model architecture involves two sub-models: a *generator model* for generating new examples and a *discriminator model* for classifying whether generated examples are real, from the domain, or fake, generated by the generator model.

- **Generator.** Model that is used to generate new plausible examples from the problem domain.
- **Discriminator.** Model that is used to classify examples as real (*from the domain*) or fake (*generated*).



Generative adversarial networks are based on a game theoretic scenario in which the generator network must compete against an adversary. The generator network directly produces samples. Its adversary, the discriminator network, attempts to distinguish between samples drawn from the training data and samples drawn from the generator.

— Page 699, [Deep Learning](#), 2016.

The Generator Model

The generator model takes a fixed-length random vector as input and generates a sample in the domain.

The vector is drawn from randomly from a Gaussian distribution, and the vector is used to seed the generative process. After training, points in this multidimensional vector space will correspond to points in the problem domain, forming a compressed representation of the data distribution.

This vector space is referred to as a latent space, or a vector space comprised of [latent variables](#). Latent variables, or hidden variables, are those variables that are important for a domain but are not directly observable.



A latent variable is a random variable that we cannot observe directly.

— Page 67, [Deep Learning](#), 2016.

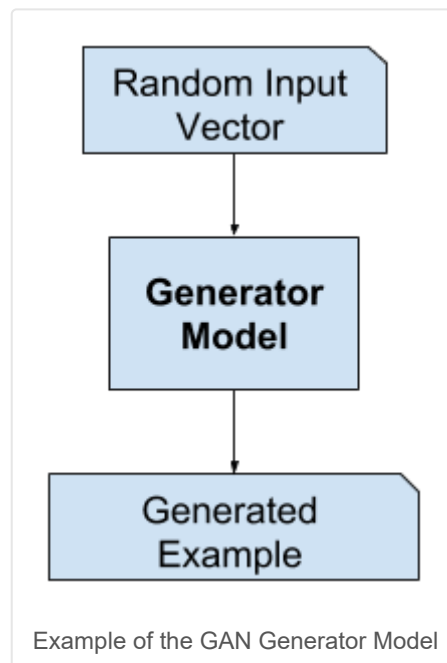
We often refer to latent variables, or a latent space, as a projection or compression of a data distribution. That is, a latent space provides a compression or high-level concepts of the observed raw data such as the input data distribution. In the case of GANs, the generator model applies meaning to points in a chosen latent space, such that new points drawn from the latent space can be provided to the generator model as input and used to generate new and different output examples.



Machine-learning models can learn the statistical latent space of images, music, and stories, and they can then sample from this space, creating new artworks with characteristics similar to those the model has seen in its training data.

— Page 270, [Deep Learning with Python](#), 2017.

After training, the generator model is kept and used to generate new samples.



The Discriminator Model

The discriminator model takes an example from the domain as input (real or generated) and predicts a binary class label of real or fake (generated).

The real example comes from the training dataset. The generated examples are output by the generator model.

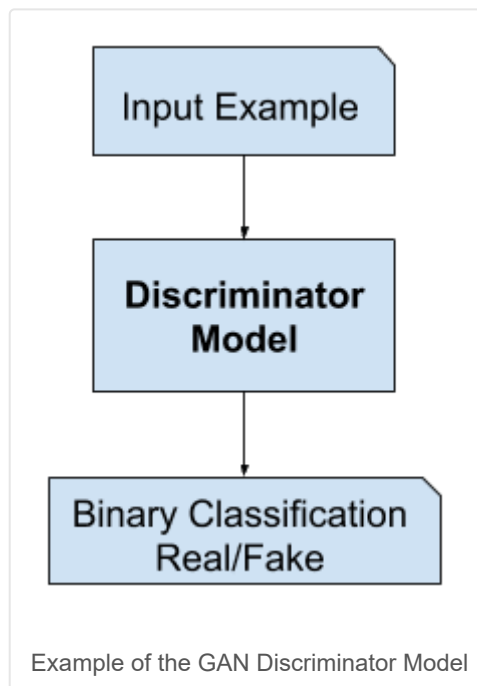
The discriminator is a normal (and well understood) classification model.

After the training process, the discriminator model is discarded as we are interested in the generator.

Sometimes, the generator can be repurposed as it has learned to effectively extract features from examples in the problem domain. Some or all of the feature extraction layers can be used in transfer learning applications using the same or similar input data.

“ We propose that one way to build good image representations is by training Generative Adversarial Networks (GANs), and later reusing parts of the generator and discriminator networks as feature extractors for supervised tasks

— Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2015.



GANs as a Two Player Game

Generative modeling is an unsupervised learning problem, as we discussed in the previous section, although a clever property of the GAN architecture is that the training of the generative model is framed as a supervised learning problem.

The two models, the generator and discriminator, are trained together. The generator generates a batch of samples, and these, along with real examples from the domain, are provided to the discriminator and classified as real or fake.

The discriminator is then updated to get better at discriminating real and fake samples in the next round, and importantly, the generator is updated based on how well, or not, the generated samples fooled the discriminator.

“ We can think of the generator as being like a counterfeiter, trying to make fake money, and the discriminator as being like police, trying to allow legitimate money and catch counterfeit money. To succeed in this game, the counterfeiter must learn to make money that is indistinguishable from genuine money, and the generator network must learn to create samples that are drawn from the same distribution as the training data.

— NIPS 2016 Tutorial: Generative Adversarial Networks, 2016.

In this way, the two models are competing against each other, they are adversarial in the game theory sense, and are playing a zero-sum game.

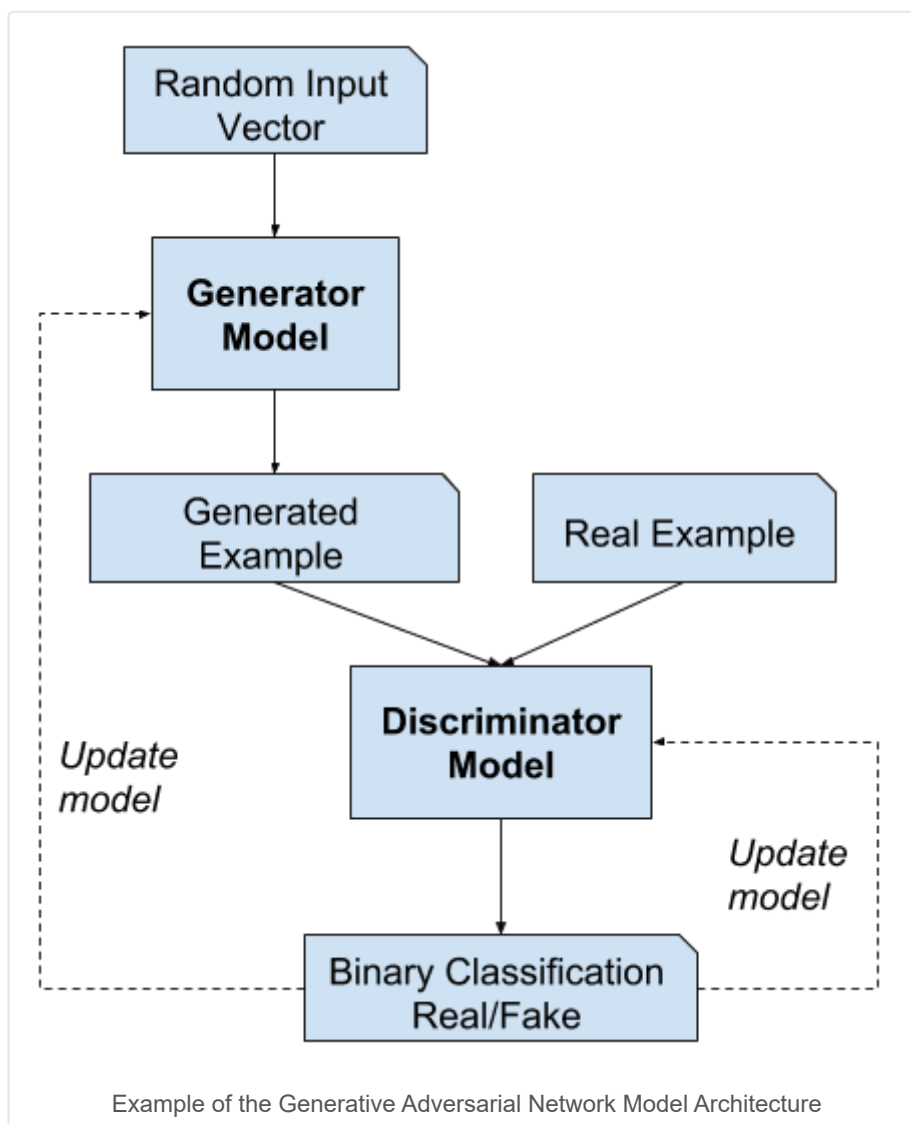
“ Because the GAN framework can naturally be analyzed with the tools of game theory, we call GANs “adversarial”.

— NIPS 2016 Tutorial: Generative Adversarial Networks, 2016.

In this case, zero-sum means that when the discriminator successfully identifies real and fake samples, it is rewarded or no change is needed to the model parameters, whereas the generator is penalized with large updates to model parameters.

Alternately, when the generator fools the discriminator, it is rewarded, or no change is needed to the model parameters, but the discriminator is penalized and its model parameters are updated.

At a limit, the generator generates perfect replicas from the input domain every time, and the discriminator cannot tell the difference and predicts “unsure” (e.g. 50% for real and fake) in every case. This is just an example of an idealized case; we do not need to get to this point to arrive at a useful generator model.



“ [training] drives the discriminator to attempt to learn to correctly classify samples as real or fake. Simultaneously, the generator attempts to fool the classifier into believing its samples are real. At convergence, the generator’s samples are indistinguishable from real data, and the discriminator outputs 1/2 everywhere. The discriminator may then be discarded.

— Page 700, [Deep Learning](#), 2016.

GANs and Convolutional Neural Networks

GANs typically work with image data and use Convolutional Neural Networks, or CNNs, as the generator and discriminator models.

The reason for this may be both because the first description of the technique was in the field of computer vision and used CNNs and image data, and because of the remarkable progress that has been seen in recent years using CNNs more generally to achieve state-of-the-art results on a suite of computer vision tasks such as object detection and face recognition.

Modeling image data means that the latent space, the input to the generator, provides a compressed representation of the set of images or photographs used to train the model. It also means that the generator generates new images or photographs, providing an output that can be easily viewed and assessed by developers or users of the model.

It may be this fact above others, the ability to visually assess the quality of the generated output, that has both led to the focus of computer vision applications with CNNs and on the massive leaps in the capability of GANs as compared to other generative models, deep learning based or otherwise.

Conditional GANs

An important extension to the GAN is in their use for conditionally generating an output.

The generative model can be trained to generate new examples from the input domain, where the input, the random vector from the latent space, is provided with (conditioned by) some additional input.

The additional input could be a class value, such as male or female in the generation of photographs of people, or a digit, in the case of generating images of handwritten digits.

“Generative adversarial nets can be extended to a conditional model if both the generator and discriminator are conditioned on some extra information y . y could be any kind of auxiliary information, such as class labels or data from other modalities. We can perform the conditioning by feeding y into the both the discriminator and generator as [an] additional input layer.

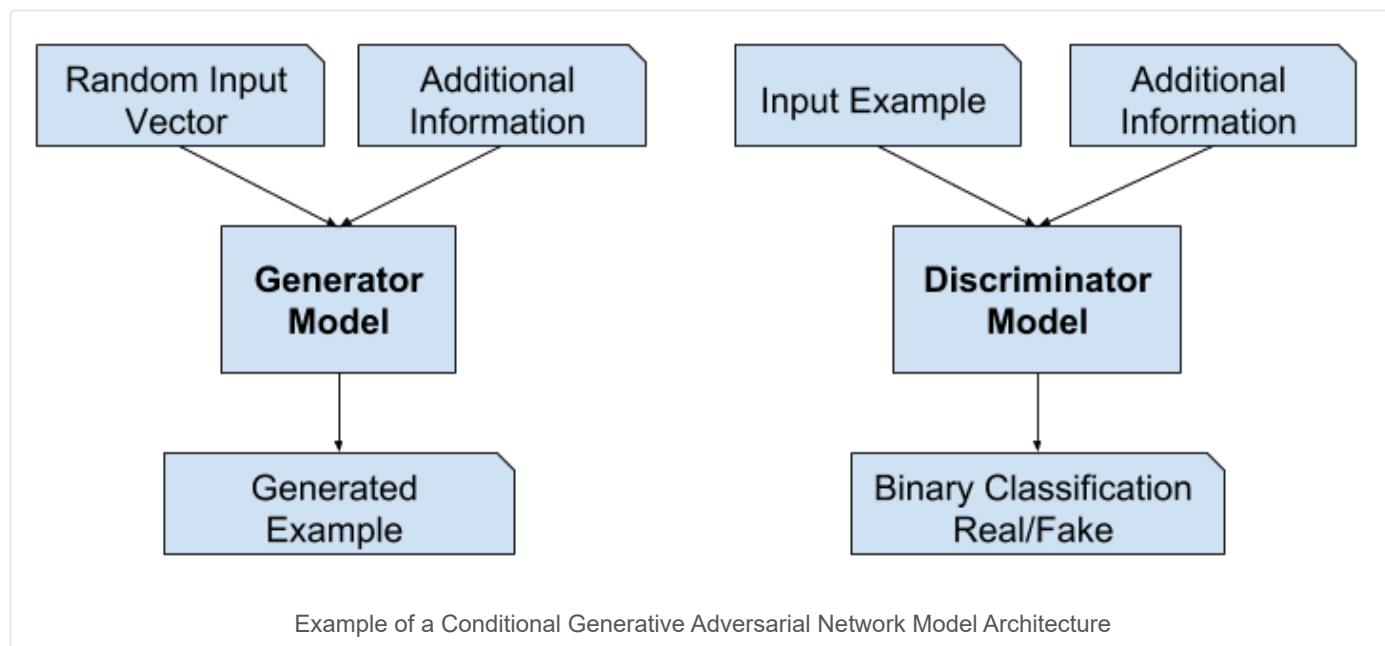
— [Conditional Generative Adversarial Nets](#), 2014.

The discriminator is also conditioned, meaning that it is provided both with an input image that is either real or fake and the additional input. In the case of a classification label type conditional input, the discriminator would then expect that the input would be of that class, in turn teaching the generator to generate examples of that class in order to fool the discriminator.

In this way, a conditional GAN can be used to generate examples from a domain of a given type.

Taken one step further, the GAN models can be conditioned on an example from the domain, such as an image. This allows for applications of GANs such as text-to-image translation, or image-to-image translation. This allows for some of the more impressive applications of GANs, such as style transfer, photo colorization, transforming photos from summer to winter or day to night, and so on.

In the case of conditional GANs for image-to-image translation, such as transforming day to night, the discriminator is provided examples of real and generated nighttime photos as well as (conditioned on) real daytime photos as input. The generator is provided with a random vector from the latent space as well as (conditioned on) real daytime photos as input.



Why Generative Adversarial Networks?

One of the many major advancements in the use of deep learning methods in domains such as computer vision is a technique called [data augmentation](#).

Data augmentation results in better performing models, both increasing model skill and providing a regularizing effect, reducing generalization error. It works by creating new, artificial but plausible examples from the input problem domain on which the model is trained.

The techniques are primitive in the case of image data, involving crops, flips, zooms, and other simple transforms of existing images in the training dataset.

Successful generative modeling provides an alternative and potentially more domain-specific approach for data augmentation. In fact, data augmentation is a simplified version of generative modeling, although it is rarely described this way.

“... enlarging the sample with latent (unobserved) data. This is called data augmentation. [...] In other problems, the latent data are actual data that should have been observed but are missing.”

— Page 276, [The Elements of Statistical Learning](#), 2016.

In complex domains or domains with a limited amount of data, generative modeling provides a path towards more training for modeling. GANs have seen much success in this use case in domains such as deep reinforcement learning.

There are many research reasons why GANs are interesting, important, and require further study. Ian Goodfellow outlines a number of these in his 2016 conference keynote and associated technical report titled “[NIPS 2016 Tutorial: Generative Adversarial Networks](#).”

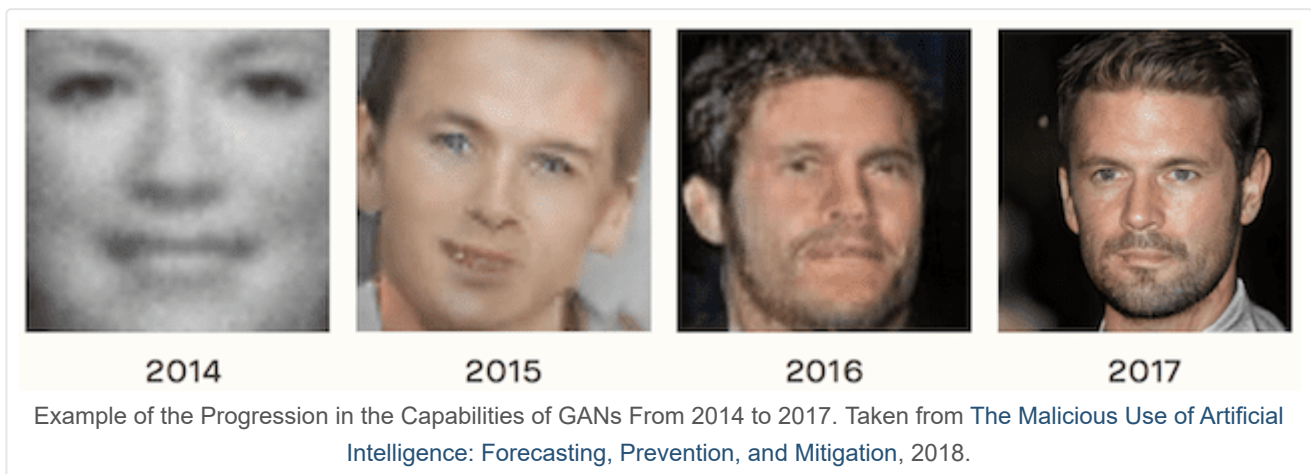
Among these reasons, he highlights GANs’ successful ability to model high-dimensional data, handle missing data, and the capacity of GANs to provide multi-modal outputs or multiple plausible answers.

Perhaps the most compelling application of GANs is in conditional GANs for tasks that require the generation of new examples. Here, Goodfellow indicates three main examples:

- **Image Super-Resolution.** The ability to generate high-resolution versions of input images.
- **Creating Art.** The ability to create new and artistic images, sketches, painting, and more.
- **Image-to-Image Translation.** The ability to translate photographs across domains, such as day to night, summer to winter, and more.

Perhaps the most compelling reason that GANs are widely studied, developed, and used is because of their success. GANs have been able to generate photos so realistic that humans are unable to tell that they are of objects, scenes, and people that do not exist in real life.

Astonishing is not a sufficient adjective for their capability and success.



Further Reading

This section provides more resources on the topic if you are looking to go deeper.

Posts

- [Best Resources for Getting Started With Generative Adversarial Networks \(GANs\)](#)
- [18 Impressive Applications of Generative Adversarial Networks \(GANs\)](#)

Books

- [Chapter 20. Deep Generative Models, Deep Learning](#), 2016.
- [Chapter 8. Generative Deep Learning, Deep Learning with Python](#), 2017.
- [Machine Learning: A Probabilistic Perspective](#), 2012.
- [Pattern Recognition and Machine Learning](#), 2006.
- [The Elements of Statistical Learning](#), 2016.

Papers

- [Generative Adversarial Networks](#), 2014.
- [Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks](#), 2015.
- [NIPS 2016 Tutorial: Generative Adversarial Networks](#), 2016.
- [Conditional Generative Adversarial Nets](#), 2014.
- [The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation](#), 2018.

Articles

- [Generative model](#), Wikipedia.
- [Latent Variable](#), Wikipedia.
- [Generative Adversarial Network](#), Wikipedia.

Summary

In this post, you discovered a gentle introduction to Generative Adversarial Networks, or GANs.

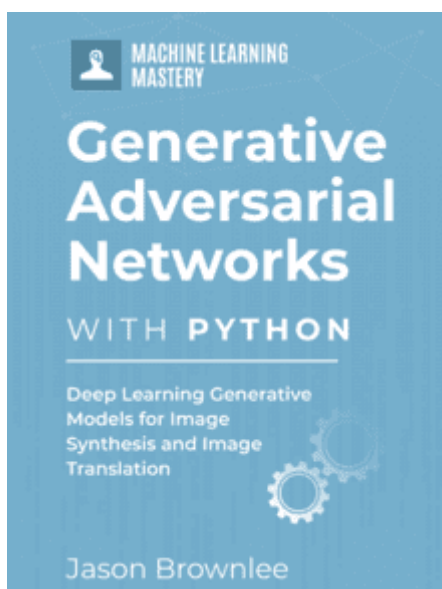
Specifically, you learned:

- Context for GANs, including supervised vs. unsupervised learning and discriminative vs. generative modeling.
- GANs are an architecture for automatically training a generative model by treating the unsupervised problem as supervised and using both a generative and a discriminative model.
- GANs provide a path to sophisticated domain-specific data augmentation and a solution to problems that require a generative solution, such as image-to-image translation.

Do you have any questions?

Ask your questions in the comments below and I will do my best to answer.

Develop Generative Adversarial Networks Today!



Develop Your GAN Models in Minutes

...with just a few lines of python code

Discover how in my new Ebook:

[Generative Adversarial Networks with Python](#)

It provides **self-study tutorials** and **end-to-end projects** on:
DCGAN, conditional GANs, image translation, Pix2Pix, CycleGAN
and much more...

Finally Bring GAN Models to your Vision Projects

Skip the Academics. Just Results.

SEE WHAT'S INSIDE

Tweet

Tweet

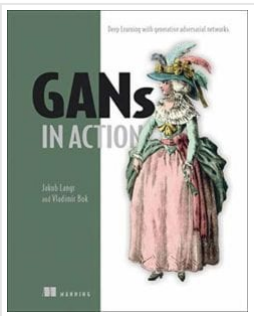
Share

Share

More On This Topic



A Tour of Generative Adversarial Network Models



9 Books on Generative Adversarial Networks (GANs)



Best Resources for Getting Started With GANs



18 Impressive Applications of Generative Adversarial...



Tips for Training Stable Generative Adversarial Networks



How to Develop a Conditional GAN (cGAN) From Scratch



About Jason Brownlee

Jason Brownlee, PhD is a machine learning specialist who teaches developers how to get results with modern machine learning methods via hands-on tutorials.

[View all posts by Jason Brownlee →](#)

< 18 Impressive Applications of Generative Adversarial Networks (GANs)

[Tips for Training Stable Generative Adversarial Networks >](#)

84 Responses to *A Gentle Introduction to Generative Adversarial Networks (GANs)*



Mohammed June 17, 2019 at 1:00 pm #

REPLY ↩

Interesting introduction for GAN models. GAN model is a generative model, and what is the advantage of GAN in feature learning compared with other deep models?



Jason Brownlee June 17, 2019 at 2:10 pm #

REPLY ↩

Thanks.

I'm not sure GANs can be helpful in feature learning exactly.



Hitesh June 21, 2019 at 4:05 pm #

REPLY ↩

Use of transfer learning, from trained discriminator weights may be helpful



Sayak Paul July 10, 2019 at 2:27 pm #

REPLY ↩

Excellent introduction, Jason.

Here are some very interesting kinds of stuff that are GANish that you can consider to include in the concluding section of the article:

- <https://adversarial-ml-tutorial.org>
 - <http://scholarsy.blogspot.com/2018/11/deoldify-freeopen-photo-retoucher-based.html>
 - <https://www.amazon.in/Hands-Unsupervised-Learning-Using-Python/dp/1492035645>
-



Jason Brownlee July 11, 2019 at 9:42 am #

REPLY ↩

Thanks for sharing.

I have a post on “GAN books” scheduled that reviews many examples.



Gregory J Matthews June 17, 2019 at 1:01 pm #

REPLY ↩

This was a great post..



Jason Brownlee June 17, 2019 at 2:10 pm #

REPLY ↩

Thanks!



Matthew Teow June 17, 2019 at 1:14 pm #

REPLY ↩

Hi Jason, are you preparing a GAN book?



Jason Brownlee June 17, 2019 at 2:10 pm #

REPLY ↩

Yes. It is coming together slowly.



Matthew Teow June 19, 2019 at 12:19 pm #

REPLY ↩

Great!



Murwanashyaka Christian June 21, 2019 at 3:29 pm #

REPLY ↩

Hi Jason! Great articule indeed, I can see this GAN models can be use in credit scoring system. Can you recommend me any best deep learning model for credit scoring system. I really like your post. I will appreciate your recommendation Jason. God bless you!



Jason Brownlee June 22, 2019 at 6:30 am #

REPLY ↩

I'm not sure off hand, perhaps I will write about the topic in the future.

Perhaps try searching on google scholar?



Anubhav June 18, 2019 at 3:02 am #

REPLY ↩

Hi Jason! When can we expect some hands on tutorials on GANs to start working on them?



Jason Brownlee June 18, 2019 at 6:42 am #

REPLY ↩

Very soon!

They are all written and scheduled.



Muhammad iqbal bazmi June 18, 2019 at 5:40 pm #

REPLY ↩

Awesome



Jason Brownlee June 19, 2019 at 7:50 am #

REPLY ↩

Thanks!



Falahgs June 18, 2019 at 8:34 pm #

REPLY ↩

Hi Jason....thanks for great information about GAN
do u have tutorial in practice about that techniques...

Thanks for sharing your post

Jason Brownlee June 19, 2019 at 7:52 am #

REPLY ↩



Yes, I have a number of tutorials written and scheduled to come out on the blog very soon.



Oluwaseun Abel Ofinni June 21, 2019 at 5:39 am #

REPLY ↩

Thanks Jason. I look forward to your book on GAN.



Jason Brownlee June 21, 2019 at 6:40 am #

REPLY ↩

Thanks! I'm very excited about it.



OTTE G June 21, 2019 at 5:41 am #

REPLY ↩

are you aware of GAN's that have been used in EEG signal analysis?



Jason Brownlee June 21, 2019 at 6:41 am #

REPLY ↩

Not off hand, sorry.



Asif Khan June 21, 2019 at 12:24 pm #

REPLY ↩

Thank you so much for the nice introduction to GAN's



Jason Brownlee June 21, 2019 at 2:02 pm #

REPLY ↩

You're welcome, I'm glad you found it useful.



Phillip Ha June 21, 2019 at 2:43 pm #

REPLY ↩

They are really interesting techniques. I plan to implement GANs into an experimental project to explore in depth of the models. Thanks very much for introducing GANs.



Jason Brownlee June 22, 2019 at 6:26 am #

REPLY ↩

Thanks.

Excited to hear about your project!



श्रीनिवास आदप्पा June 22, 2019 at 4:25 pm #

REPLY ↩

Jason, which models in deep learning is better for Facial Feature Extraction to do face reading and personality determination using deep learning.



Jason Brownlee June 23, 2019 at 5:31 am #

REPLY ↩

Good question, perhaps start with a face detection algorithm to find the face and a classification algorithm for each main emotion.

This will help for face detection:

<https://machinelearningmastery.com/how-to-perform-face-detection-with-classical-and-deep-learning-methods-in-python-with-keras/>

Also I recommend checking the state of the art on google scholar.



R-ALGO Engineering Big Data July 2, 2019 at 1:08 am #

REPLY ↩

Great details on GAN. I'm just starting to dig into GAN's but hopegulyl over the next 3-6 months can start working on projects to implement GAN.

Awesome post!



Jason Brownlee July 2, 2019 at 7:33 am #

REPLY ↩

Thanks!



Nitish August 29, 2020 at 6:23 am #

REPLY ↩

Hi, Jason.

I have assigned a project, i.e., to generate a face of a person just from eye. Is it possible to implement it with GAN.

If yes please tell me steps.

Can also mail me



Jason Brownlee August 29, 2020 at 8:06 am #

REPLY ↩

Yes, sounds like a great project.

You can approach it as a image-to-image translation problem, e.g. input an eye, output a face. Perhaps explore a modified conditional gan or pix2pix model?



dcart July 4, 2019 at 7:36 pm #

REPLY ↩

GANs is getting my attention now. correct me if I am wrong, GANs can be use to up sample training data and in the process can feature engineered additional features in the training data?

Thanks Jason !



Jason Brownlee July 5, 2019 at 7:53 am #

REPLY ↩

Not as far as I know.

It could be used for data augmentation for computer vision problems.



Alex November 19, 2019 at 8:45 am #

REPLY ↩

Have a question, GANs belongs to semi-supervised learning or unsupervised learning?

Thank you.



Jason Brownlee November 19, 2019 at 1:58 pm #

REPLY ↩

They are a type of unsupervised learning, trained via self-supervised learning.

This might help:

<https://machinelearningmastery.com/types-of-learning-in-machine-learning/>



Midhat December 12, 2019 at 8:55 pm #

REPLY ↩

very good article I want to ask you how can I made face swap with gan



Jason Brownlee December 13, 2019 at 5:58 am #

REPLY ↩

Thanks.

Perhaps start by preparing a dataset of faces then explore using the cyclegan model.

yuwen January 6, 2020 at 4:37 am #

REPLY ↩



this will be a very interesting project



Jason Brownlee January 6, 2020 at 7:13 am #

REPLY ↩

Thanks!



Prateek March 12, 2020 at 12:23 pm #

REPLY ↩

Thanks for sharing such a resourceful content. I need to start a project using GANs and as always, your article has given me a very good understanding about this topic.



Jason Brownlee March 12, 2020 at 1:41 pm #

REPLY ↩

Thanks, I'm happy to hear that.



Felix April 1, 2020 at 3:55 am #

REPLY ↩

Hey Jason, thanks for a lovely and in depth explanation of GANs. I'm a Ph.D. student working on GANs and how to improve the performance of a GAN network. would love to know if you have some tutorials on EBGAN.



Jason Brownlee April 1, 2020 at 5:53 am #

REPLY ↩

See this for improving performance:

<https://machinelearningmastery.com/how-to-code-generative-adversarial-network-hacks/>

Thanks for the suggestion.



Felix April 2, 2020 at 12:48 am #

REPLY ↩

Thank you so much!!!!



Jason Brownlee April 2, 2020 at 6:00 am #

REPLY ↩

You're welcome.



Abdur Rehman April 4, 2020 at 6:37 am #

REPLY ↩

Hi Jason,

I have a kind of out of topic question but I hope you would bear with me. You are good researcher and blogger and I know you read so many research papers to update yourself with deep learning literature. I am also trying to read research papers but one thing which I find difficult to find the prerequisite to study a paper. In paper, there are so many references are given but one can not read all of them. So can you kindly tell me any trick how can I find the prerequisite papers to study and understand a paper in a good way.

Thanks,



Jason Brownlee April 4, 2020 at 8:58 am #

REPLY ↩

There's no trick. You keep looking up terms and concepts until you understand enough to proceed or gloss over the details that are not critical to what you want to know.



sambath parthasarathy April 18, 2021 at 1:31 am #

REPLY ↩

Great article Jason. Thank you !

Abdur:

I agree with Jason's inputs. Also

For many research papers, there will be one foundational, must-read kind of paper. You find out and thoroughly understand that.



Shahar Gino June 6, 2020 at 10:10 pm #

REPLY ↩

Thanks, very educative and and well described 👍



Jason Brownlee June 7, 2020 at 6:24 am #

REPLY ↩

You're welcome.



John Ballesteros July 30, 2020 at 9:27 am #

REPLY ↩

Hello Jason, It's a great post !

I want to ask you, I'm currently working on image to map translation, GANs definitely can help me doing that, however there are many objects in a map, does a GAN can help me discovering objects that have not been used in the training process?



Jason Brownlee July 30, 2020 at 1:45 pm #

REPLY ↩

Thanks!

Hmm, the model might be able to guess if the new objects are very similar to objects seen before. Ideally you want to have very representative training data.



Ramzi August 11, 2020 at 8:34 pm #

REPLY ↩

Thanks,

what is the criteria stopping in gans?
Can I use the early stopping method?



Jason Brownlee August 12, 2020 at 6:10 am #

REPLY ↩

When the generated images meet your desired quality or when it is clear the model has collapsed.



Manohar September 10, 2020 at 3:42 am #

REPLY ↩

Great article. Including cycle GAN will be helping more.



Jason Brownlee September 10, 2020 at 6:36 am #

REPLY ↩

You can find posts on cyclegan on the blog, use the search:
https://machinelearningmastery.com/?s=cyclegan&post_type=post&submit=Search



Manohar September 10, 2020 at 8:08 pm #

REPLY ↩

Sorry for not noticing, and thanks for your reply. I will make best use of it and thank you for such standard blogs.



Jason Brownlee September 11, 2020 at 5:55 am #

REPLY ↩

You're welcome.



Vinay November 13, 2020 at 4:55 pm #

REPLY ↩

Hi Jason

Contents and examples are very well described in the book Generative Adversarial Networks with Python which i have purchased few months back.

Can you share some tutorials on implementation of TABULAR GAN coz its related to real time implementation of 1D data.



Jason Brownlee November 14, 2020 at 6:29 am #

REPLY ↩

Great suggestion, thank you.



Manish December 2, 2020 at 5:13 am #

REPLY ↩

Awesomeee.....



Jason Brownlee December 2, 2020 at 7:49 am #

REPLY ↩

Thanks!



Aurangzeb Alamgeer February 12, 2021 at 1:26 am #

REPLY ↩

Thank you for this awesome article!



Jason Brownlee February 12, 2021 at 5:47 am #

REPLY ↩

You're welcome!



Muhammad Qasim March 10, 2021 at 7:19 pm #

REPLY ↩

Interested, Useful, brief, and nice explanation for beginners to learn.

Regards.



Jason Brownlee March 11, 2021 at 5:09 am #

REPLY ↩

Thanks.



Salem Sheikh June 20, 2021 at 7:48 am #

REPLY ↩

Thank you.



Jason Brownlee June 21, 2021 at 5:35 am #

REPLY ↩

You're welcome.



Rio McMahon July 15, 2021 at 4:45 am #

REPLY ↩

Hi Jason, thanks for putting together such a great blog; I reference it frequently.

I have a clarification question regarding the adversarial game “played” between the generator and the discriminator. In the introduction to this article you state that:

> The two models are trained together in a zero-sum game, adversarial, until the discriminator model is fooled about half the time, meaning the generator model is generating plausible examples.

In the original paper, the (arguable) main goal of training is for the generator to produce output such that when the discriminator evaluates the output of the generator:

> The discriminator is unable to differentiate between the two distributions, i.e. $D(x) = 1/2$

Where $D(x)$ is the probability that the output came from the data rather than the generator. This is discussed later in your article.

I am unclear where your interpretation of “fooling the discriminator about half the time” comes from; my interpretation is that the convergence condition is when $D(x)=1/2$ rather than “fooled half the time”. Could you please clarify? Thanks.



Jason Brownlee July 15, 2021 at 5:34 am #

REPLY ↩

Both descriptions refer to the same thing. Also, it is not “converged” as GANs don’t converge, it is a semi-stable state of a dynamical system.



shankar D October 8, 2021 at 6:08 pm #

REPLY ↩

Hai..

Thanks for gave a nice introduction of GANs and i ll going to use this GANs models for Soil classification

and crop suggestion it will come the better results or not pls give the reply and i can use this models?



Adrian Tam October 13, 2021 at 5:26 am #

REPLY ↩

What is your question?



Mostafa Papen November 3, 2021 at 10:25 pm #

REPLY ↩

Thanks for the great post and introduction.



Akshay s January 14, 2022 at 7:11 pm #

REPLY ↩

really helped for preparing for my seminar



Akshay s January 14, 2022 at 7:12 pm #

REPLY ↩

really helped for preparing for my seminar, Thank you 😊



James Carmichael January 15, 2022 at 11:27 am #

REPLY ↩

Thank you for the feedback Akshay!



Mohammadreza Lahmi January 16, 2022 at 3:51 pm #

REPLY ↩

Hi

I am recently familiar with GAN. I want to run my project with GAN; I have a dataset from pre-and post-operative images of dentists, and I want to generate a post-operative image. Which kind of GAN can I use for this project?



James Carmichael January 17, 2022 at 7:16 am #

REPLY ↩

Hi Mohammadreza...While I am not able to speak specifically to your project, I can offer the following resource as an introduction:

<https://machinelearningmastery.com/how-to-develop-an-information-maximizing-generative-adversarial-network-infogan-in-keras/>



William January 20, 2022 at 12:00 pm #

REPLY ↩

Can you recommend a reference that provides a more gentle introduction to GANs? My education is in Electrical Engineering and Mathematics, and I have training in Machine Learning and Convolutional Neural Networks. However, I'm simply not able to understand the provided explanation. Thanks in advance.



James Carmichael January 25, 2022 at 10:55 am #

REPLY ↩

Hi William...The following may be of interest to you:

<https://machinelearningmastery.com/resources-for-getting-started-with-generative-adversarial-networks/>



Rahul June 21, 2022 at 3:44 pm #

REPLY ↩

Great Post, giving a brief idea how GANs work. Thank you



Catherine November 16, 2022 at 1:37 am #

REPLY ↩

Quite insightful post. Will you be including any articles on implementation of cGANs on non-image data?



James Carmichael November 16, 2022 at 7:23 am #

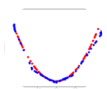
REPLY ↩

Thank you for the feedback and suggestions Catherine! This topic is a great one for consideration of future content! Stay tuned!

Leave a Reply

Name (required)

Email (will not be published) (required)

[SUBMIT COMMENT](#)**Welcome!**I'm *Jason Brownlee* PhDand I **help developers** get results with **machine learning**.[Read more](#)**Never miss a tutorial:****Picked for you:**[How to Develop a Pix2Pix GAN for Image-to-Image Translation](#)[How to Develop a 1D Generative Adversarial Network From Scratch in Keras](#)[How to Develop a CycleGAN for Image-to-Image Translation with Keras](#)



How to Develop a Conditional GAN (cGAN) From Scratch



How to Develop a Wasserstein Generative Adversarial Network (WGAN) From Scratch

Loving the Tutorials?

The GANs with Python EBook is
where you'll find the ***Really Good*** stuff.

>> SEE WHAT'S INSIDE

© 2023 Guiding Tech Media. All Rights Reserved.

[LinkedIn](#) | [Twitter](#) | [Facebook](#) | [Newsletter](#) | [RSS](#)

[Privacy](#) | [Disclaimer](#) | [Terms](#) | [Contact](#) | [Sitemap](#) | [Search](#)