Introduction to generative adversarial networks (GANs)

Yulia Newton, Ph.D.

CS156, Introduction to Artificial Intelligence

San Jose State University

Spring 2021

What are GANs?

- Generative adversarial networks (GANs) - generative ANN models
  - Often used for generating synthetic data
  - Similarly to autoencoders, they find the latent space representation of the data from which new data can be generated
    - Use neural networks to learn this latent space representation
  - Can learn to mimic any distributions and patterns of the data
    - With great power comes great responsibility!
- First introduced in 2014
  - https://arxiv.org/abs/1406.2661
- Widely used in image generation, video generation, and voice generation, music generation, text generation, art generation
  - GANs gave a rise to deepfakes - not something to brag about!

GANs are useful for synthetic data generation

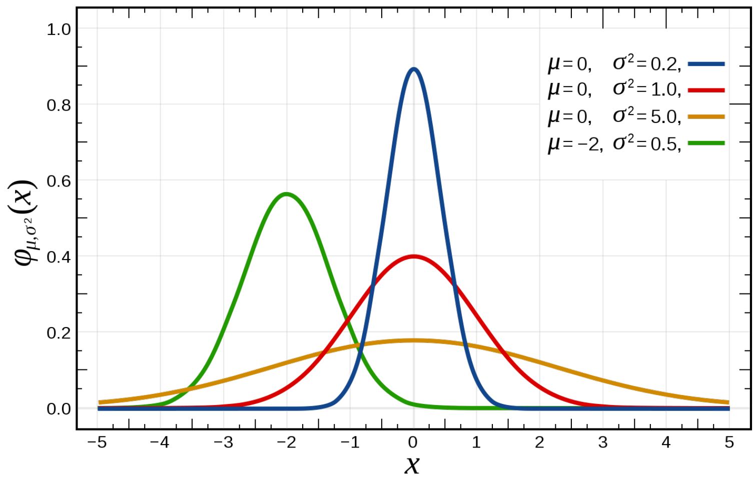
- We have some images of cats
- We want to generate new images of cats that look like cats but are new and are not pictures of any real cats
- Can we use our training set of cat images to generate a “cat probability distribution” to generate new cat images?

- GAN models are a good answer for this type of need

Probability distributions provide a way to produce new values

- Once we have a probability distribution for a random variable we can generate new values of that random variable from this distribution

If our random variable comes from a Gaussian distribution we can generate new data points using the probability density function, given the parameters for the correct distribution



https://en.wikipedia.org

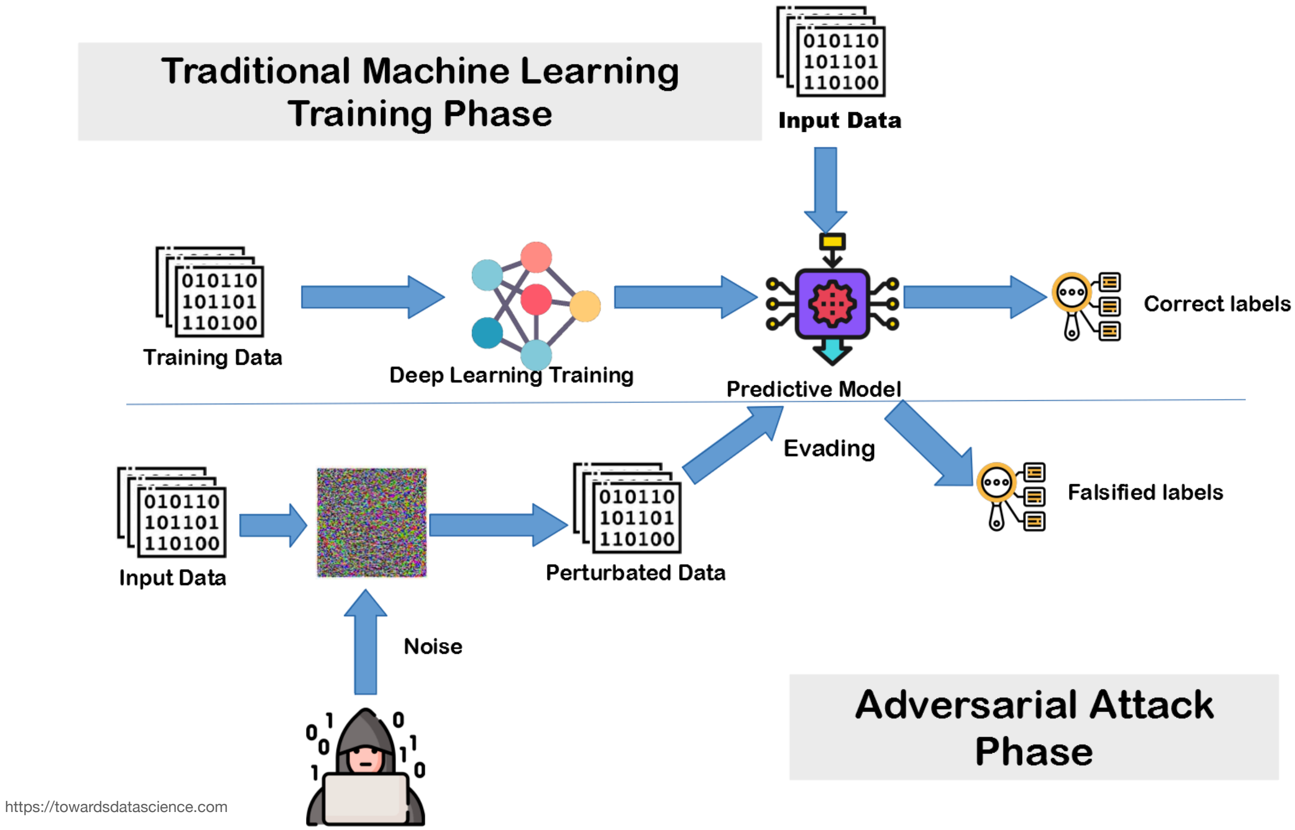
How can we train a model to obtain the correct distribution(s)?

- Two similar ideas:
  - Direct training
    - Generating probability distributions and comparing them directly to the true probability distributions, then backpropagating the difference between the two as an error
      - Example: generative matching networks (GMN)
  - Indirect training
    - No direct comparison between the distributions happens
    - A downstream classification task is involved to assess how “good” the learned distributions are
      - Example: generative adversarial networks (GAN)

The idea of adversarial training

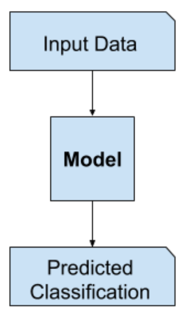
- ML technique that produces models that are trained on deceptive input
  - Robustness of the model
  - Prepares the model for adversarial attacks/situations/data
    - E.g. spammers get more and more creative, so spam filters are forced to as well

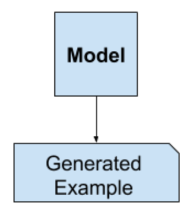
Traditional ML in the context of adversarial input



Discriminative vs. generative models

- Two types of ML models
  - Discriminative - supervised models that discriminate between (predict) class labels
  - Generative - unsupervised models that learn distributions and patterns within the input data, from which a new synthetic observation can be generated
    - This synthetic observation will share some properties with the existing input data

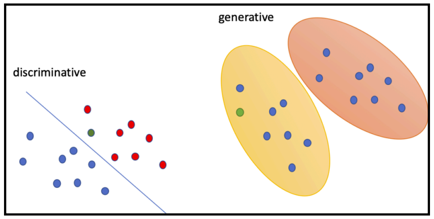




https://machinelearningmastery.com

What does it mean mathematically?

- The terms come from the fields of probability theory and statistics
- Discriminative models
  - Given data X model output Y
  - Model P(Y|X)
  - Parameters for P(Y|X) are estimated directly from the input data
- Generative models
  - There is a probability distribution for output P(Y)
  - There is a probability distribution for input data P(X|Y)
  - These give you a joint probability P(X,Y)
  - Use Bayes rule to compute P(Y|X)
    - Estimate from the joint probability P(X,Y)
  - Parameters for P(Y) and P(X|Y) are estimated directly from the input data, while P(Y|X) is computed from those distributions



http://mbenhaddou.com

Bayes' theorem

- Also referred to as "Bayes' law" and "Bayes' rule"
- Named after Thomas Bayes (1701 - 1761), British statistician
- Gives a rule for how to calculate a probability of an event, given prior knowledge
- In a context of a classification problem, prior knowledge is the probability distribution of the output variable, the conditional probability of the data observations given a label

Bayes' theorem:

$$\boxed{P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}}$$

We can use proportionality of the posterior probability to numerator

$$P(Y|X) \propto P(X|Y)P(Y)$$

P(X|Y)P(Y) = P(X, Y)

Baye's rule example

- Pancreatic cancer example (https://en.wikipedia.org/wiki/Bayes%27_theorem)
- Not everyone who has the symptoms associated with pancreatic cancer actually have pancreatic cancer
- Known prior knowledge about pancreatic cancer
  - Incidence = 1/100,000
  - 1/10000 of healthy people will experience symptoms consistent with pancreatic cancer

Symptom \ Cancer	Yes	No	Total
Yes	1	10	11
No	0	99989	99989
Total	1	99999	100000

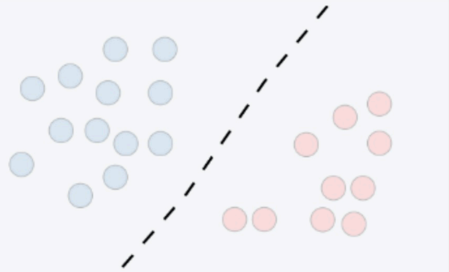
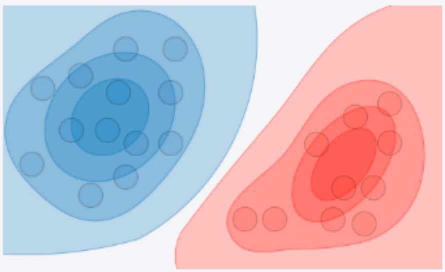
$$
\begin{aligned}
P(\text{Cancer}|\text{Symptoms}) &= \frac{P(\text{Symptoms}|\text{Cancer})P(\text{Cancer})}{P(\text{Symptoms})} \\
&= \frac{P(\text{Symptoms}|\text{Cancer})P(\text{Cancer})}{P(\text{Symptoms}|\text{Cancer})P(\text{Cancer}) + P(\text{Symptoms}|\text{Non-Cancer})P(\text{Non-Cancer})} \\
&= \frac{1 \times 0.00001}{1 \times 0.00001 + (10/99999) \times 0.99999} = \frac{1}{11} \approx 9.1\%
\end{aligned}
$$

Probability of having pancreatic cancer, given the symptoms

90.9% will not have the diagnosis

Differences between what these models capture

- Discriminative - capture conditional probability P(Y|X)
- Generative - capture joint probability P(X,Y) or just P(X) in the absence of output labels

	Discriminative model	Generative model
Goal	Directly estimate $P(y\|x)$	Estimate $P(x\|y)$ to then deduce $P(y\|x)$
What's learned	Decision boundary	Probability distributions of the data
Illustration		
Examples	Regressions, SVMs	GDA, Naive Bayes

https://github.com/mainkoon81/ooo-Minkun-Model-Collection-II

Examples of these types of models

- Discriminative
  - Logistic regression, SVM, decision tree, multi-layer perceptron (MLP), traditional CNN, etc.
- Generative
  - Variational autoencoders (VAE), generative adversarial networks (GAN), Naïve Bayes, hidden Markov models (HMM), Markov random fields, etc.

How does all this relate to GANs?

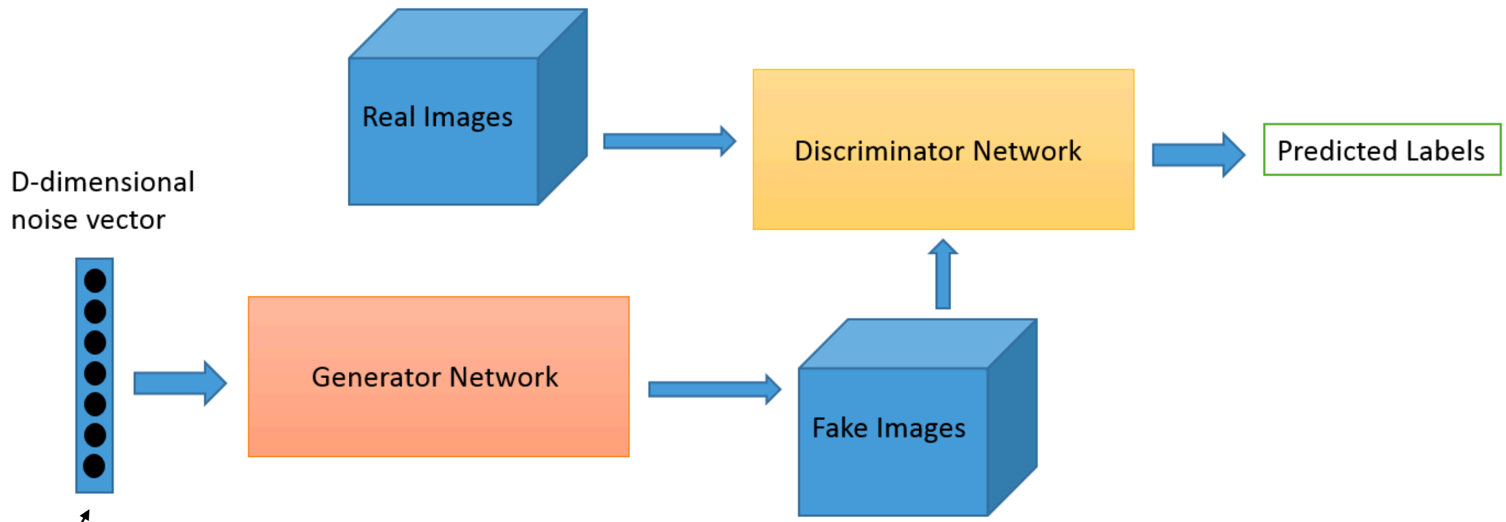
- GANs have two parts
  - Generator
    - Generates new synthetic data
  - Discriminator
    - Evaluates likelihood of the new synthetic data
      - How authentic is the newly generated data?

  - Both discriminator and generator are ANNs

The goal

- The goal of the GAN model is to produce synthetic data that the discriminator has hard time discriminating from real data

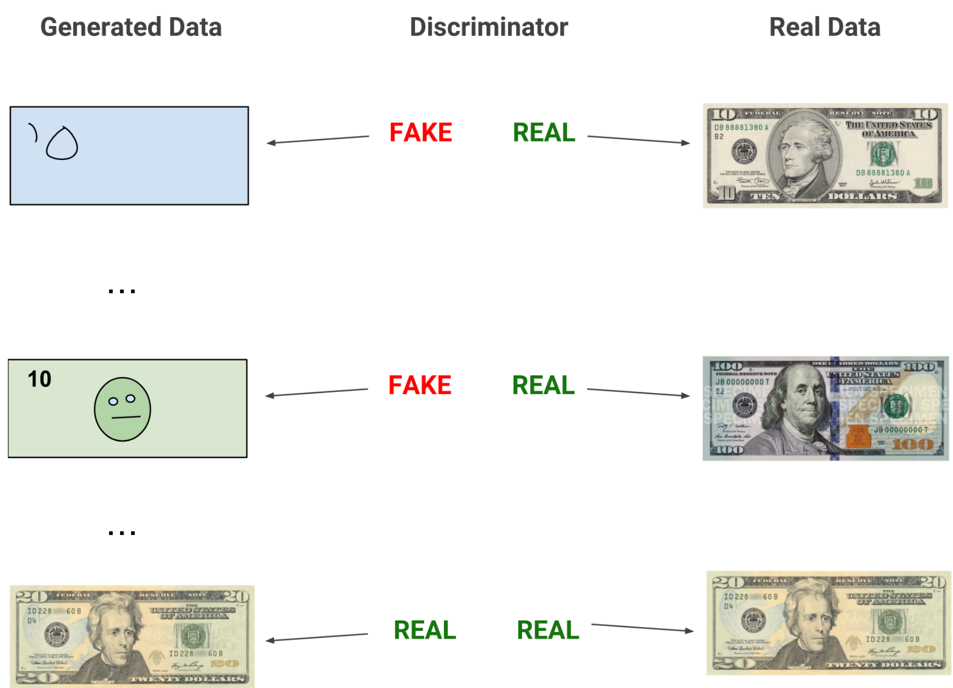
Generator vs. discriminator parts



https://wiki.pathmind.com/generative-adversarial-network-gan

Usually drawn randomly from a Gaussian distribution

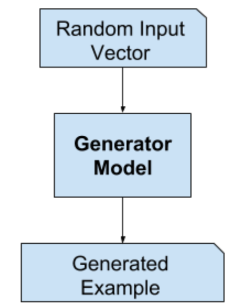
Toy example: counterfeit money generator



https://developers.google.com/machine-learning/gan/gan_structure

Generator model

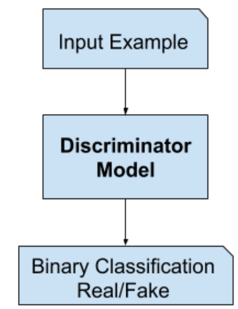
- Takes a noise vector, usually generated from a Gaussian distribution
  - Seeds generative process
  - This vector is the random input space for the generator
  - Just like with VAE models, this space is a compression/ projection of the input data that uses probability distributions to represent hidden variables
- Can be thought of as feature extraction models
- The generator model includes the latent space representation vector as well as the ANN that produces new data



https://machinelearningmastery.com

Discriminator model

- Takes in an observation, real or synthetically generated, and predicts the output label/ class
  - Real observations come from the training input data
  - Synthetically generated observations come from the generator model
- The discriminator is a standard classifier
- This is the downstream component of the training that assesses how “good” the probability distributions are in the latent space
  - Remember we mentioned the “indirect” way to learn these distributions? Discriminator model is the indirect component.
- The desire is for the discriminator to fail as much as possible (misclassify synthetic data as real)
- Normally after a GAN is trained the discriminator model is discarded as the purpose of a GAN is generating new data points



https://machinelearningmastery.com

Training the discriminator model

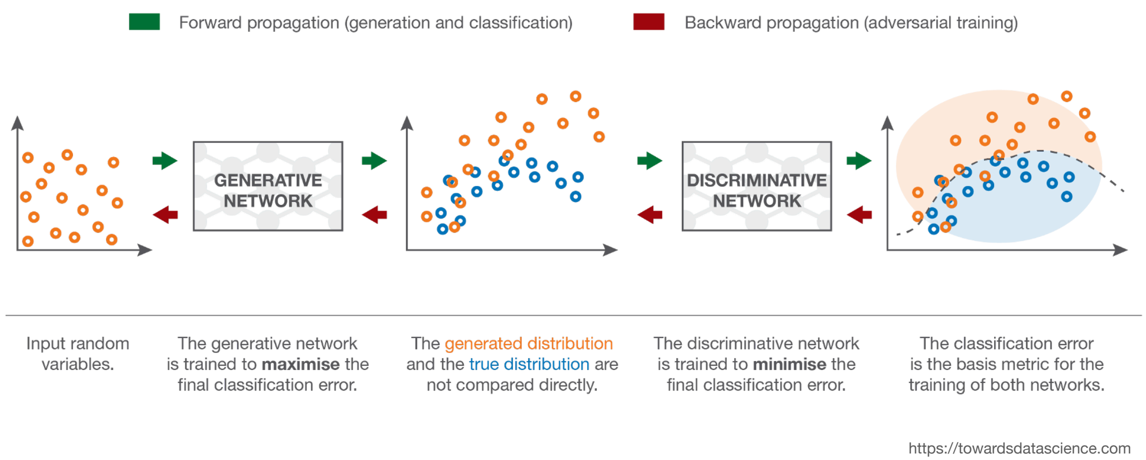
- The discriminator classifies real and synthetic data
- The loss function penalizes discriminator for misclassifications
  - Real observations predicted to be synthetic
  - Synthetic observations predicted to be real
- The discriminator updates the weights using backpropagation from this loss function

Training the generator model

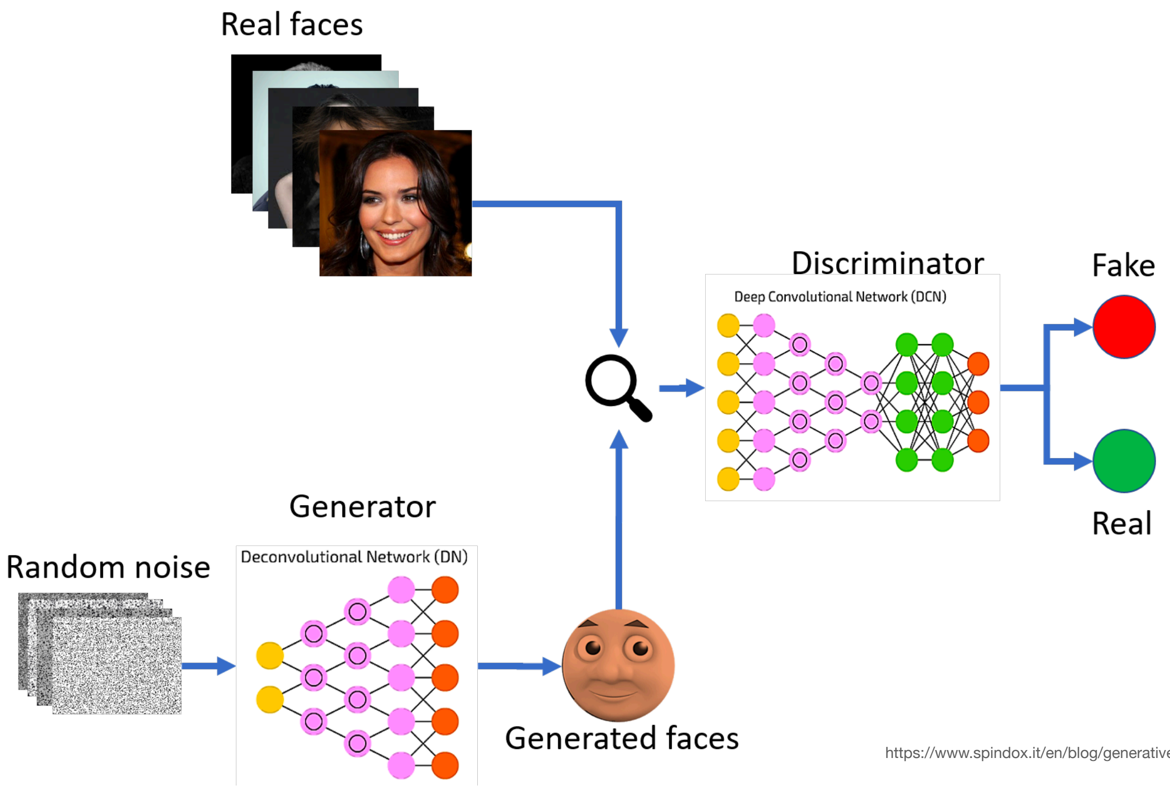
- Produce generated output from the latent space
- Get the classification results from the discriminator model (real or synthetic label)
- The loss function penalizes misclassifications from the discriminator
  - Synthetic observations predicted to be real
- Backpropagate through both generator and discriminator to obtain gradients
- Update only the generator weights

Discriminator and generator have different goals

- The goal of generator is to trick/fool the discriminator
  - Trained to maximize the final classification error
- The goal of discriminator is to detect synthetic data
  - Trained to minimize the final classification error



After training the generative model produces data that appears to resemble real data



https://www.spindox.it/en/blog/generative-adversarial-neural-networks

Competing goals

- Competing goals means the two networks (generator and discriminator) are in the race to beat each other
- This race is what is making both networks better as they are learning
- In game theory this type of setup is called minimax two-players game
  - The equilibrium state is where the generator produces data from distributions similar to real distributions and the discriminator discriminates these observations as real or synthetic with 0.5 probability
- Two feedback loops
  - The discriminator has a feedback loop to the real data
  - The generator has a feedback loop with the discriminator

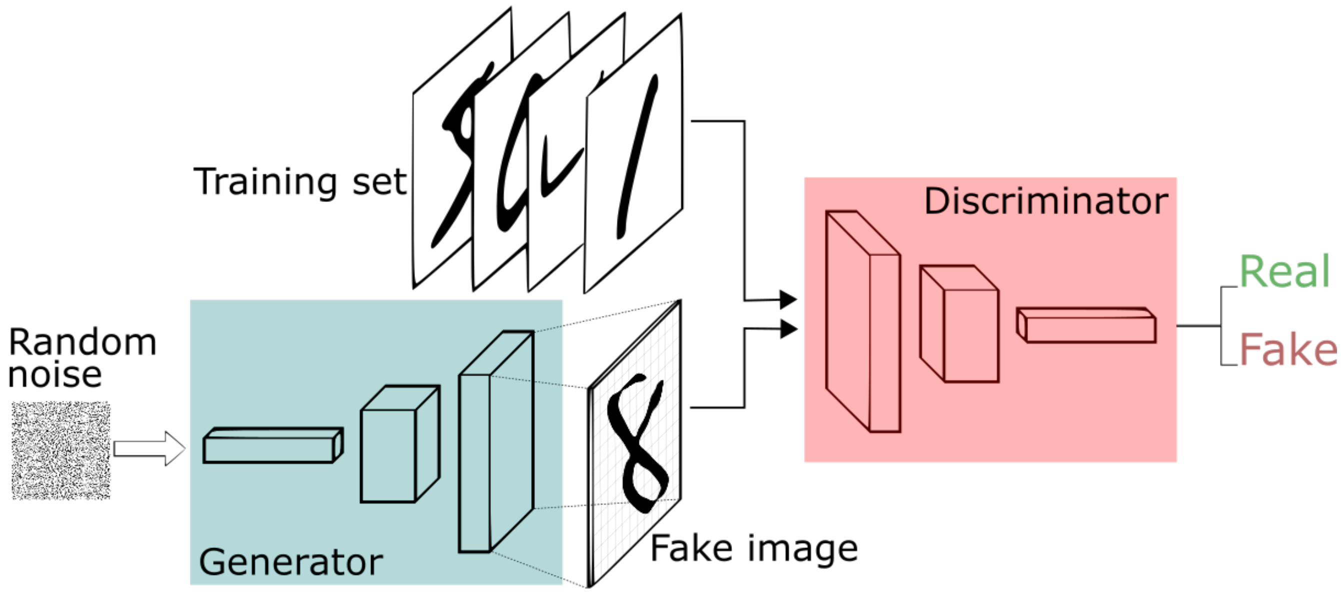
Training a GAN model

- GAN training joggles training two models at the same time
- The training occurs in alternating manner
  - The discriminator trains for one or more epochs
  - The generator trains for one or more epochs
  - Repeat until the convergence
- The generator is kept constant during the discriminator training
- The discriminator is kept constant during the generator training
- Convergence in GAN training is extremely difficult to identify
  - As the generator improves the discriminator performance gets worse
  - As the discriminator improves the generator performance gets worse
  - At equilibrium the discriminator prediction accuracy of 0.5 on synthetic data (discriminator flips a coin to make a prediction about the data authenticity)

Few words about the loss function

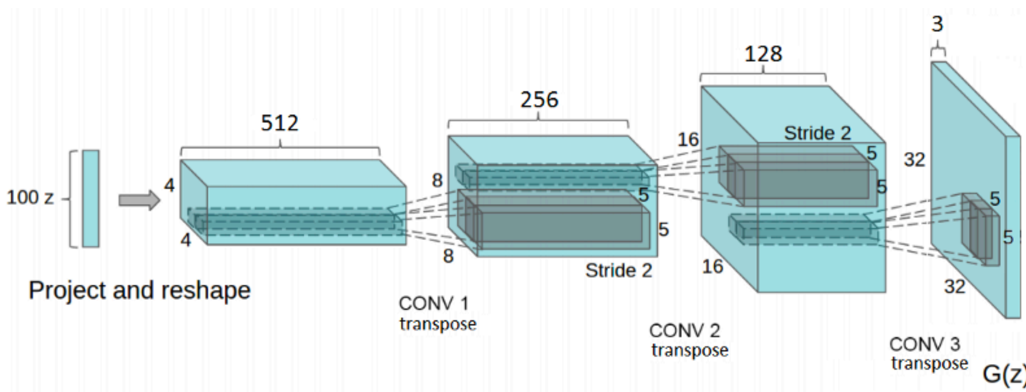
- Several loss functions are often used in training a GAN model
  - Minimax loss
    - Introduced in the original GAN paper (https://arxiv.org/abs/1406.2661)
  - Wasserstein loss
    - Introduced in 2017 paper “Wasserstein GAN” by Arjovsky et al. (https://arxiv.org/abs/1701.07875)
    - GAN models that use it are often called “WGAN” or “Wasserstein GAN”
    - Default loss in tensorflow

Obtaining the generator model is the objective of training a GAN



Real life example of a generator model

- Deep convolutional generative adversarial networks (DCGANs)
  - "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks" Radford et. al 2015 (https://arxiv.org/abs/1511.06434)
  - Image generator



GAN vs VAE

- Both GAN and VAE models learn a latent space representation for the training data
- VAE’s latent space is normalized (conforms to Gaussian distribution) while GAN’s is not

GANs and VAE produce similar results

- Which type of model you want to use depends on the type of task you want to perform
- VAE models have been shown to outperform GANs in face generation
  - https://syncedreview.com/2019/06/06/going-beyond-gan-new-deepmind-vae-model-generates-high-fidelity-human-faces/

Let’s look at some code examples

- GAN application to generating handwritten digits
  - *GAN.MNIST.ipynb*