GANdinsky

Course: Neural Networks & Deep Learning
Instructor: Dr. Farid Alizadeh
Group Members: Elnaz A. Torkamani, Ruiwen Zhang, Sunit Nair

RUTGERS

Contents

RUTGERS

1: Introduction

Project scope and limitations

Project scope

- Create a Generative Adversarial Network to create art from portrait and abstract datasets.
- Set up project to read data from Google Drive and save output to Google Drive.
- Compress raw dataset into readable and reusable files.
- Extended scope: explore additional types of GAN (WGAN)

Limitations

- Objective evaluation of model is difficult compared to classification/regression problems.
- Good results require large number of epochs and/or complex modeling.
- Long session running times and limitations on hardware availability (Colab Pro).
- Subjective nature and interpretation of “art” (more about this in the next slide)

Definition of art

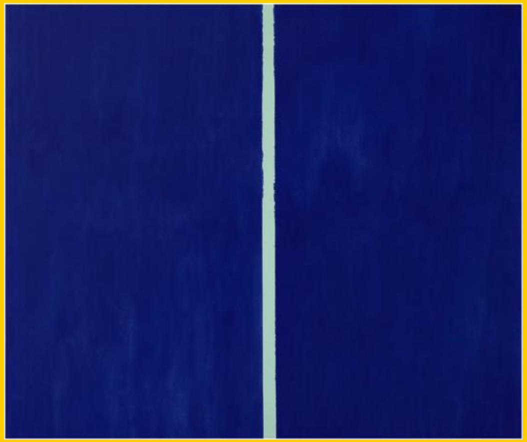
- For the purpose of this project, any image generated that resembles the style of the dataset to a recognizable degree without introducing overfitted elements or blatant patterns is considered generated art.

1: Introduction

Subjective nature of art

The blood-red mirror
by Gerhard Richter
$1.1 million

Untitled
by Cy Twombly
$2.3 million



Onement VI
by Barnett Newman
$43.84 million

Orange, Red, Yellow
by Mark Rothko
$86 million

Another reason for this might be the controversial status quo of the *"art world"* as explained in this video.

Source: iloboyou.com

RUTGERS

2: Generative Adversarial Networks

Concepts and overview

Definition

Generative Adversarial Networks (GANs) are algorithmic architectures that use two neural networks, pitting against the other (thus the "adversarial") in order to generate new, synthetic instances of data that can pass for real data.

History

GANs were introduced in a paper by Ian Goodfellow and other researchers at the University of Montreal, including Yoshua Bengio, in 2014.

Applications

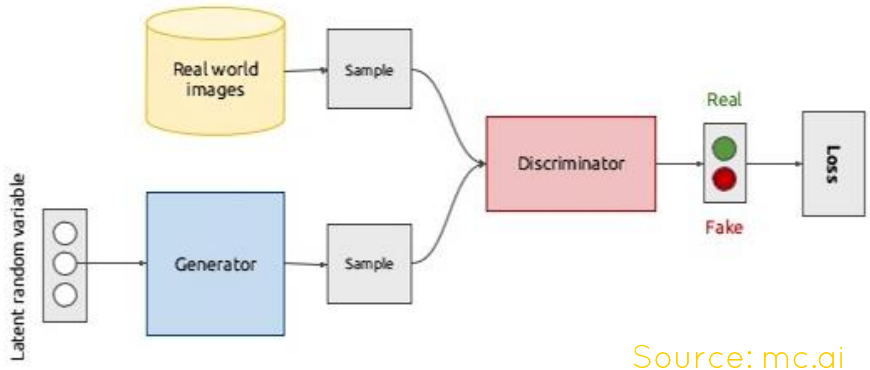
Widely used in the generation of images, video, and voice.

Some types of GANs

- Deep Convolutional GANs (DCGANs)
- Conditional GANs (cGANs)
- StackGANs
- InfoGANs
- Wasserstein GANs (WGANs)

2: Generative Adversarial Networks

Generative vs Discriminative Networks



Source: mc.ai

Discriminative Network

Tries to classify input data. Given the features of an instance of data, predict a label or category to which it belongs. In this instance, differentiate between a real and fake painting.

Generative Network

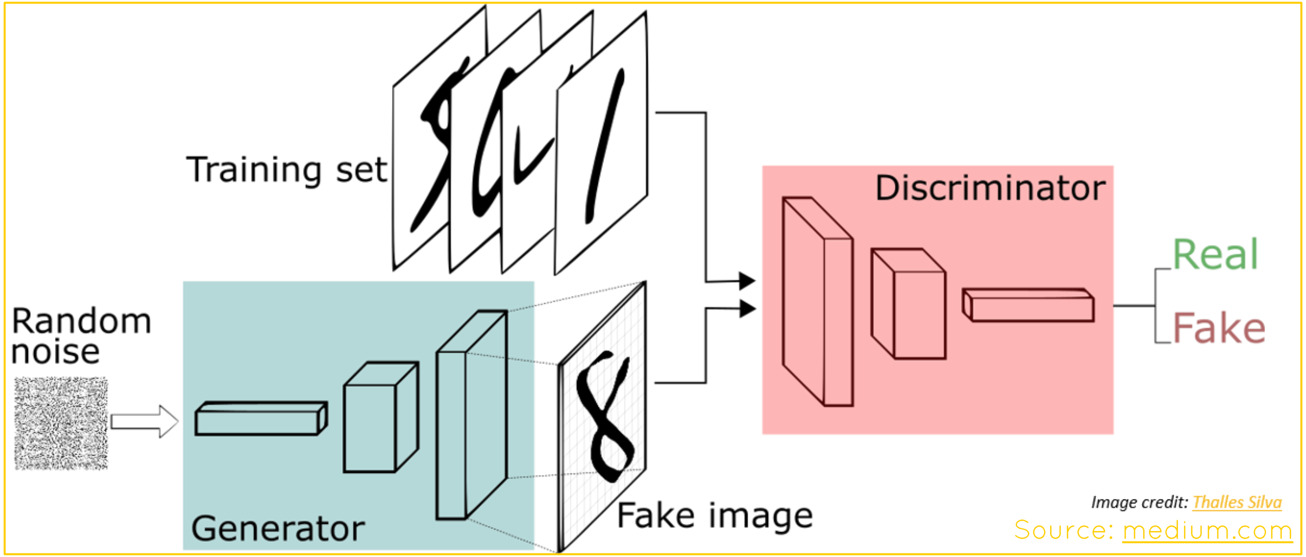
Attempt to predict features given a certain label or category to which that data belongs.

These opposing objectives and competition between the two networks is the hallmark of GANs.

RUTGERS

2: Generative Adversarial Networks

How GANs work



Source: medium.com

How GANs work

- The generator generates and returns an image
- The generated image is fed into the discriminator alongside a stream of images taken from the actual dataset
- The discriminator takes in both real and fake images and returns probabilities

3: Project Architecture

Architecture, Libraries, and Tools used

Python Libraries Used

Keras, numpy, Pillow (PIL), google.colab, time, os

Platforms/IDEs used

- Google Colaboratory
- Spyder

Input

Images resized to 128 x 128, and compressed into a single .npy file

Output

- Generated images in a configurable grid
- Model state/weights in .h5 files

Configurable Parameters

No. of images to preview/save, number of epochs after which to save previews, noise size, batch size, generated resolution, image size,

RUTGERS

3: Project Architecture

Architecture, Libraries, and Tools used

GAN Model: DCGAN (Deep Convolutional Generative Adversarial Networks)

Discriminator

- Model: Sequential
- Activation: LeakyRelU(alpha=0.2)
- Last Layer Activation: Sigmoid

Generator

- Model: Sequential
- Activation: RelU
- Last Layer Activation: tanh

Optimizer

Adam:

Learning Rate: 0.00015

Beta: 0.5

Loss/Metrics

- Binary cross-entropy
- Accuracy

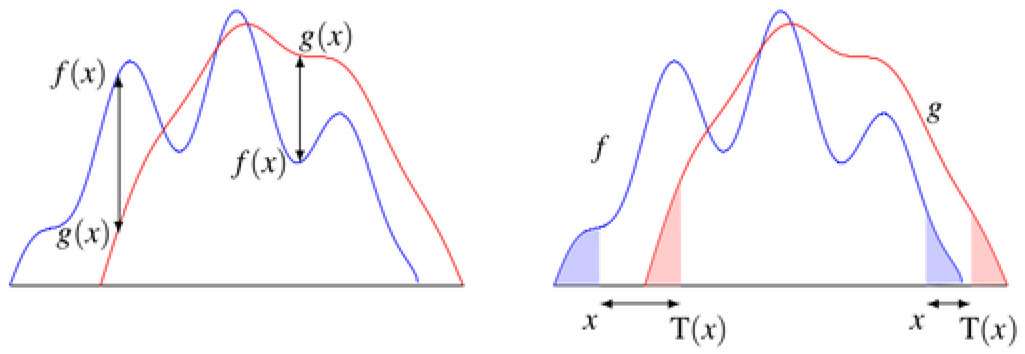
3: Project Architecture (Aside)

Experiments with WGAN

Mathematical understanding

- Wasserstein or Kantorovich–Rubinstein metric or distance is a distance function defined between probability distributions on a given metric space M.
- Consider two functions f and g. Usual distances inevitable measure distance (difference) between f(x) and g(x) i.e. along the y-axis for certain values of x or use maximums/averages.
- However, if f and g are similarly distributed but only translated linearly along x-axis, these distances may differ vastly.
- Also, traditional distances fail to measure the similarity between the functions that is lost due to the translation.
- Wasserstein distances attempt to capture these similarities.

Source: Optimal Transport for Applied Mathematicians, by F. Santambrogio



3: Project Architecture (Aside)

Experiments with WGAN

Intuitive Understanding

- A special way to compare probability distributions.
- Earth-mover problem.

In the context of GANs

- A meaningful loss metric that correlates with the generator's convergence and sample quality.
- Improved stability of the optimization process.
- Intuitively, convert the discriminator network from a forgery expert into an art critic.

Results with WGANs

- Shows improved convergence in LSUN and MNIST datasets (max 64 x 64).
- Mixed results and convergence issues with other datasets (CelebA).
- Failed to converge with our datasets from WikiArt (128 x 128 portrait/abstract datasets).

Converting DCGAN to WGAN

- Ensure last layer of Discriminator is linear (not sigmoid).
- Add a clipper constraint to discriminator layers.
- Train discriminator (5x) more than generator.
- Optimizer: RMSProp
- Loss function: Wasserstein Loss (mean(sum(y_true * y_pred))

4: Results

Portrait dataset vs generated images

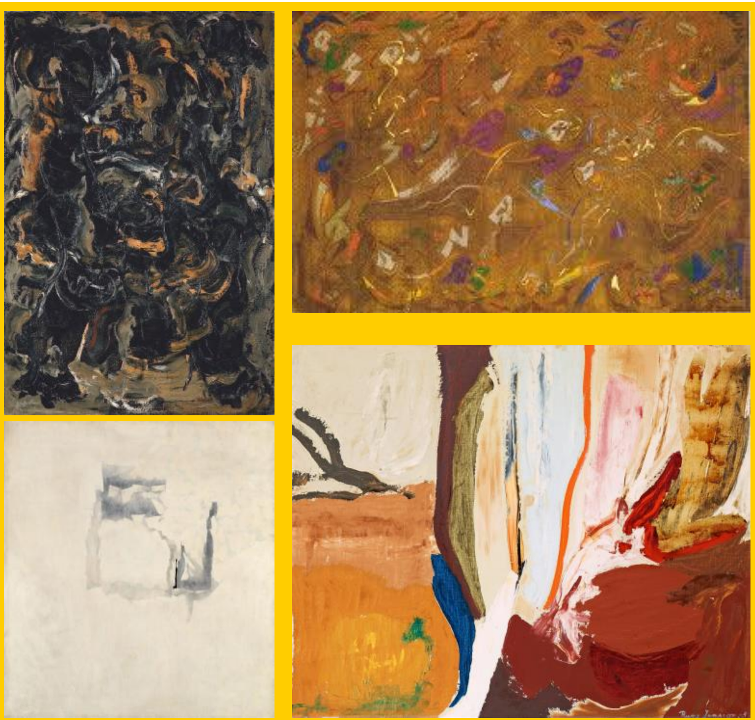
Sample images from dataset (input)

Sample images from results (output)

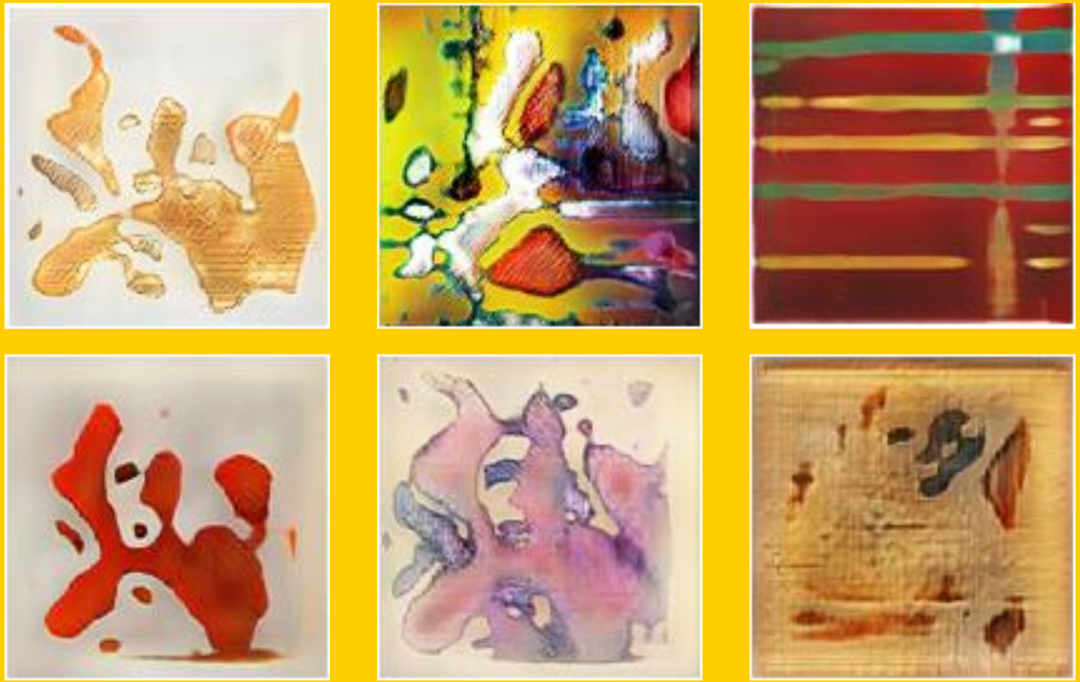
RUTGERS

4: Results

Abstract dataset vs generated images



Sample images from dataset (input)



Sample images from results (output)

5: References

Datasets, Repositories, and References

Datasets

- WikiArt.org
- Repository on Google Drive sourced from GitHub

Repositories

- GitHub: nndl-groupc/gandinsky
- GANdinsky Public Repository: Google Drive

Online Resources Used

- https://towardsdatascience.com/
- https://heartbeat.fritz.ai/
- https://medium.com/

Wasserstein GAN Research Paper

- https://arxiv.org/pdf/1701.07875.pdf

5: Conclusions

Future work and potential

Project features and limits

- GANdinsky was developed and tested on Colab (free).
- Maximum epochs were run on Colab Pro account (97000 epochs in 24 hours).
- Relatively new (2014) WGAN model doesn't work well with the chosen dataset.
- Saved models can be used to continue training.

Possible refinements/Future work

- Implement WGAN or other better discriminator/generator models.
- Lease hardware (Amazon EC2) to run more epochs.
- Use larger datasets (9000+ images).
- Larger inputs and output resolutions (256 x 256).

Potential

- AI art sold for $432,500.
- A trend is emerging with AI generated content being sold as art.
- GANs and their creative applications have expanded the applications of AI from problem solving to creativity and artistic areas.
- The potential for such models is still being discovered: games, security, entertainment.

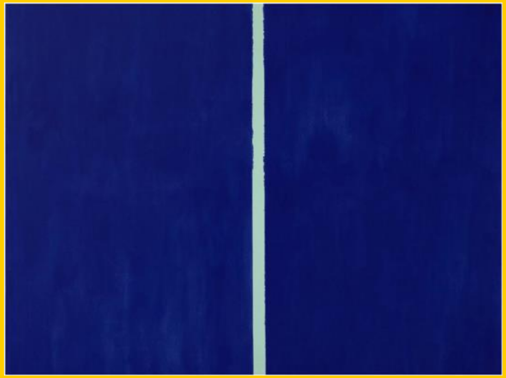
RUTGERS

5: Conclusions

Comparing GANdinsky to AI generated and human art

Edmond de Belamy
by French art collective
Obvious (AI generated)
$432,500

portrait_epoch_46000[0,1]
by GANdinsky



Onement VI
by Barnett Newman
$43.84 million

abstract_epoch_19000[2,6]
by GANdinsky

RUTGERS

Thank You!

“