Generative Adversarial Networks and their applications in Finance

Joerg Osterrieder
Zurich University of Applied Sciences, University of Twente
Joint work with

- Prof. Dr. Ali Hirsa, Columbia University
- Weilong Fu, Columbia University
- Dr. Branka Hadji Misheva, Zurich University of Applied Sciences
- Chenxin Nie, University of Zurich

This presentation is purely for educational purposes, not all references might be fully cited.
Op, 24 “I am AI”

AIVA (Artificial Intelligence Virtual Artist)
\[
\min_{\mathcal{G}} \max_{\mathcal{D}} E_x [\log (\mathcal{D}(x))] + E_z [\log (1 - \mathcal{D}(\mathcal{G}(z)))]
\]
A detour --- Neural networks

Fundamental unit of a Neural Network

Activation function

\[\text{output} = \begin{cases} 
1 & \text{if } \sum w_i x_i > 0 \\
-1 & \text{otherwise}
\end{cases}\]

**Inputs**

**Weights**

**Outputs**

Image sources: medium.com; wikipedia.org
A detour --- Neural networks

Neural networks are functions

\[ f : \mathbb{R}^n \rightarrow \mathbb{R}^m \]

Image sources: medium.com; wikipedia.org
A detour --- Neural networks and the universal approximation theorem

Neural networks can approximate (almost) arbitrary functions

Cybenko (1989) states that any continuous function on a compact domain can be approximated with any precision by an appropriate neural network with sufficient width and depth.
(Partial, incomplete) History of Neural Networks

1763 Thinking in numbers – Thomas Bayes

1842 From numbers to poetry – Ada Lovelace

1943 Neurons go artificial

Image source: https://aiartists.org/ai-timeline-art
(Partial, incomplete) History of Neural Networks

1943 Can a machine think? – Edmund Berkeley

1997 Man vs. machine: fight of the 20th century

2002 The first robot for the home

Image source: https://aiartists.org/ai-timeline-art
The protein folding problem
Alphafold – A solution to a 50-year old grand challenge in biology

T1037 / 6vr4
90.7 GDT
(RNA polymerase domain)

T1049 / 6y4f
93.3 GDT
(adhesin tip)

- Experimental result
- Computational prediction
From traditional econometrics to deep neural networks

Traditional econometrics

\[ E(R_i) = R_f + \beta_i (E(R_m) - R_f) \]

Deep Neural Networks

ImageNet Classification with Deep Convolutional Neural Networks
by A. Krizhevsky
Cited by 5,498
Related articles

The neural network, which has 60 million parameters and 600,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling...

Large Scale Distributed Deep Networks - Google Research
by J. Dean
Cited by 3,150
Related articles

we consider the problem of training a deep network with billions of parameters using tens of thousands of CPU cores. We have developed a software framework...

We need more data
A DAY IN DATA

The exponential growth of data is undisputed, but the numbers behind this explosion — fuelled by internet of things and the use of connected devices — are hard to comprehend, particularly when looked at in the context of one day.

4PB

of data created by Facebook, including

500m

tweets are sent every day

Twitter

320bn

emails to be sent each day by 2021

294bn

emails to be sent each day by 2020

Redaktionsgruppende

Facebook Research

350m

photos

100m

hours of video

watch time

3.9bn

people use emails

65bn

messages sent over WhatsApp and two billion minutes of voice and video calls made

442B

ACUMULATED DIGITAL UNIVERSE OF DATA

4.4ZB

2020

44ZB

2021

Demystifying Data Units

From the ancient Sumerian “bar” or “measure”, larger units of measurement are more frequently being used to explore the mass of data.

<table>
<thead>
<tr>
<th>Unit</th>
<th>Suffix</th>
<th>Short Form</th>
<th>Short Form</th>
<th>Full Name</th>
<th>Long Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>bit</td>
<td>bit</td>
<td>bit</td>
<td>bit</td>
<td>bit</td>
</tr>
<tr>
<td>B</td>
<td>byte</td>
<td>B</td>
<td>byte</td>
<td>byte</td>
<td>byte</td>
</tr>
<tr>
<td>KB</td>
<td>kilobyte</td>
<td>KB</td>
<td>kibibyte</td>
<td>1,024 bytes</td>
<td>1,024 bytes</td>
</tr>
<tr>
<td>MB</td>
<td>megabyte</td>
<td>MB</td>
<td>mebibyte</td>
<td>1,024,000 bytes</td>
<td>1,024,000 bytes</td>
</tr>
<tr>
<td>GB</td>
<td>gigabyte</td>
<td>GB</td>
<td>gibibyte</td>
<td>1,024,000,000 bytes</td>
<td>1,024,000,000 bytes</td>
</tr>
<tr>
<td>TB</td>
<td>terabyte</td>
<td>TB</td>
<td>tebibyte</td>
<td>1,024,000,000,000 bytes</td>
<td>1,024,000,000,000 bytes</td>
</tr>
<tr>
<td>PB</td>
<td>petabyte</td>
<td>PB</td>
<td>pebibyte</td>
<td>1,024,000,000,000,000 bytes</td>
<td>1,024,000,000,000,000 bytes</td>
</tr>
<tr>
<td>EB</td>
<td>exabyte</td>
<td>EB</td>
<td>exbibyte</td>
<td>1,024,000,000,000,000,000 bytes</td>
<td>1,024,000,000,000,000,000 bytes</td>
</tr>
<tr>
<td>ZB</td>
<td>zettabyte</td>
<td>ZB</td>
<td>zebibyte</td>
<td>1,024,000,000,000,000,000,000 bytes</td>
<td>1,024,000,000,000,000,000,000 bytes</td>
</tr>
<tr>
<td>YB</td>
<td>yottabyte</td>
<td>YB</td>
<td>yobibyte</td>
<td>1,024,000,000,000,000,000,000,000 bytes</td>
<td>1,024,000,000,000,000,000,000,000 bytes</td>
</tr>
</tbody>
</table>

*Note: “i” is used as an abbreviation for ‘ten to the power of’, while an uppercase “B” represents bytes.

28PB

to be generated from wearable devices by 2020

95m

photos and videos are shared on Instagram

463EB

data will be created every day by 2025

Raconteur
The need for synthetic data in 2021

Data is protected
- Privacy and compliance limit the use of banking/client data
- Regulations prevent data sharing
- Clearance and approvals are inefficient
- The lack of sharing holds back research

Historical data is limited
- Certain events present limited historical data
- Limits statistical analysis and inference
- ML models are crippled by small input sizes

Class imbalance
- Class imbalance is high in rare-event datasets
- Datasets for fraud detection are often imbalanced
- ML and anomaly detection algorithms fail on imbalanced data
Generative Adversarial Networks introduced in 2014

- Groundbreaking work by Ian Goodfellow et al (2014)
- It tried to address the following question: Given a set of data (say, a set of human faces or Van Gogh paintings), can we generate data that are “similar”?
- The authors have proposed GAN, which uses two neural networks “competing against” each other to obtain the desired outcome.
- Yann LeCun has said “this (GAN) and the variations are now the most interesting idea in the last 10 years in ML, in my opinion.”
\[
\min_{\mathcal{G}} \max_{\mathcal{D}} E_x \left[ \log(\mathcal{D}(x)) \right] + E_z \left[ \log(1 - \mathcal{D}(\mathcal{G}(z))) \right]
\]

GANs and Nash Equilibrium

The GAN framework is a non-convex, two-player, non-cooperative game with continuous, high-dimensional parameters, in which each player wants to minimize its cost function.

The optimum of this process takes the name of Nash Equilibrium.

GANs are typically trained using gradient-descent techniques that are designed to find the low value of a cost function and not find the Nash Equilibrium of a game.

A Detour – Nash Equilibrium

What is a Nash Equilibrium?

No participant can gain by a unilateral change of strategy if the strategies of the others remain unchanged.
What else? A zoo of GANs - Different network architectures

Foundation
- Generative Adversarial Network (GAN)
- Deep Convolutional Generative Adversarial Network (DCGAN)

Extensions
- Conditional Generative Adversarial Network (cGAN)
- Information Maximizing Generative Adversarial Network (InfoGAN)
- Auxiliary Classifier Generative Adversarial Network (AC-GAN)
- Stacked Generative Adversarial Network (StackGAN)
- Context Encoders
- Pix2Pix

Advanced
- Wasserstein Generative Adversarial Network (WGAN)
- Cycle-Consistent Generative Adversarial Network (CycleGAN)
- Progressive Growing Generative Adversarial Network (Progressive GAN)
- Style-Based Generative Adversarial Network (StyleGAN)
- Big Generative Adversarial Network (BigGAN)
- Ensembles of GANs

https://github.com/hindupuravinash/the-gan-zoo
Five building blocks
• Generator network
• Discriminator network
• Loss functions
• Regularizations (weights, loss, gradient)
• Optimizers

What else? A zoo of GANs - Different network architectures
Ongoing challenges in GAN training

Mode collapse

Model gets stuck in a mode
Output loses diversity

Lack of proper evaluation metrics

No metric for time series GANs
Hard to evaluate GANs with precision

Proposed solutions

New Loss functions
New Model architectures
Additional regularizations

\(^1\) but see Conditional Sig-Wasserstein GANs for Time Series Generation, Ni et al 2020
Metz et al. 2017
Challenges, tips and tricks when training GANs

- Normalize the inputs
- A modified loss function: \((\min(1-D) \rightarrow \max \log(D))\)
- Sample from Gaussian instead of uniform distribution
- Batch Normalization
- Avoid Sparse Gradients
- LeakyReLU = good (in both G and D)
- For Downsampling, use: Average Pooling, Conv2d + stride
- Use Soft and Noisy Labels
- Use stability tricks from RL
- Track failures early
- D loss goes to 0: failure mode
- Check norms of gradients: if they are over 100, it becomes difficult
- When things are working, D loss has low variance and goes down over time vs having huge variance and spiking
- If loss of generator steadily decreases, then it’s fooling D with garbage

https://github.com/soumith/ganhacks
# GANs in Finance

## Table 1: GANs in finance research

<table>
<thead>
<tr>
<th>Field</th>
<th>Application</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Series Forecasting</td>
<td>Market Prediction</td>
<td>GAN-FD [9], ST-GAN [19], MTS-GAN [20]</td>
</tr>
<tr>
<td></td>
<td>Fine-Tuning of trading models</td>
<td>C-GAN [10], MAS-GAN [21]</td>
</tr>
<tr>
<td>Portfolio Management</td>
<td>Portfolio Optimization</td>
<td>PAGAN [11], GAN-MP [22], DAT-CGAN [23], CorrGAN [12]</td>
</tr>
<tr>
<td>Time Series Generation</td>
<td>Synthetic time series generation and Finance Data Augmentation</td>
<td>TimeGAN [24], WGAN-GP [25], FIN-GAN [3], Quant GAN [14], RA-GAN [26], CDRA-GAN [27], SigCWGAN [28], ST-GAN [19]</td>
</tr>
<tr>
<td>Fraud Detection</td>
<td>Detection of market manipulation</td>
<td>LSTM-GAN [13]</td>
</tr>
<tr>
<td></td>
<td>Detection of Credit Card Fraud</td>
<td>RWGAN [29], LSTM-GAN-2 [30]</td>
</tr>
</tbody>
</table>
The key stylized facts of financial time series

- Absence of autocorrelations
- Fat-tailed distributions
- Volatility clustering
- Gain/loss asymmetry
- Aggregational Gaussianity

Financial time-series data is non-stationary, non-markovian, with non-parametric distributions

https://radhakrishna.typepad.com/rks_musings/2014/06/stylized-facts.html
Results for financial-time series GANs

Quant-GAN\(^1\)

Regularized GAN: an architecture with two discriminators: one as a typical GAN, binary classifier, and the other one as a score function

Conditional Sig-Wasserstein GANs\(^2\)

Signature of a path

Wasserstein GAN with gradient penalty\(^5\)

Improved training of Wasserstein GAN, which pushes the discriminator Lipschitz by gradient penalty

Self-Attention GAN\(^3\)

TransGAN\(^6\)

GAN with transformer blocks but without any convolutional layers

---


### GAN variants implemented for time-series

#### Structural Variants
- DCGAN: Deep Convolutional GAN
- SAGAN: self-attention GAN with dense attention
- BIG GAN deep: larger versions of SAGAN
- YLGAN: your local GAN, with sparse attention
- Transformer GAN: with transformer blocks

#### Loss variants
- Wasserstein GAN
- WGAN with Gradient Penalty
- LS GAN: least-squares
- RAGAN: loss function improvement of DCGAN with realistic factors
- RA LS GAN: loss function improvement of LSGAN with realistic factors
- DRAGAN: deep regret analytic GAN, a loss variant similar to WGAN GP

---

See Hirsa, Fu, Osterrieder, 2021
GAN based on convolutional neural networks, with an added self-attention mechanism that improves learning on long-range dependencies.

**SAGAN**

- **Input**: Batchsize (B) × Length (L) × Channels (C)
- **f(x), g(x), h(x)**: linear transformation of x
- **attention(x)** = softmax(f(x) × g(x)) × h(x)
- **Output v(x)**: linear transformation of attention(x)
GAN modified from SAGAN to fit the task of financial time-series simulation

1. **Innovation**: Combining self-attention mechanism with convolutional networks applied to financial time-series simulation

2. **Main differences from SAGAN**
   a. Dimension changed from Batchsize \((B) \times \text{Width} (W) \times \text{Height} (H) \times \text{Channels} (C)\) for pictures to Batchsize \((B) \times \text{Length} (L) \times \text{Channels} (C)\) for series
   b. Use returns and prices as the real data, such that both the moments of returns and long-range dependency of returns can be well-fitted by the GAN model
   c. Use the loss function of WGAN-GP instead of the hinge loss in the original SAGAN loss to improve training speed
S&P 500 SAGAN-GP, daily data

- **Historical**
  - $s = -0.30$
  - $\kappa = 7.87$

- **Generated**
  - $s = -0.68$
  - $\kappa = 8.72$
Main features:

a. GAN with only transformer blocks and without any convolutional layers
b. Especially good at fitting long-range dependencies and global characteristics
c. Consists of 3-5 transformer blocks
d. Each transformer block is made up by an attention layer and an MLP (fully-connected layers)
S&P 500 TransGAN, daily data
S&P 500 TransGAN, daily data
S&P 500 W-GAN with GP, daily data
S&P 500 W-GAN with GP, daily data
S&P 500 W-GAN with GP, daily data
S&P 500 W-GAN with GP, intraday data
S&P 500 W-GAN with GP, intraday data
S&P 500 W-GAN with GP, intraday data
S&P 500 W-GAN with GP, intraday data
Probability distribution, Kurtosis and Skewness

WGAN-GP

DCGAN

SAGAN
The initial idea was to replicate artists' cuts/patterns utilizing deep convolutional generative adversarial networks with connected components.

From art to finance, the idea can be extended to preserve statistical properties in financial time series.

RegGAN Loss function

\[
\min_G \max_{D_1, D_2} V(G, D_1, D_2) \\
= \mathbb{E}_x \left[ \log(D_1(x)) \right] + \mathbb{E}_z \log(1 - D_1(G(z))) + \mathbb{E}_z \log(1 - D_2(G(z)))
\]

instead of

\[
\min_G \max_D \left\{ \mathbb{E}_{X \sim P_r} \left[ \log D(X) \right] + \mathbb{E}_{Z \sim P_z} \left[ \log(1 - D(G(Z))) \right] \right\}
\]

- Regularized GAN: an architecture with two discriminators: one as a typical GAN, binary classifier, and the other one as a score function to control connected components
- for controlling different statistical properties of time-series such as heavy-tailedness, skewness and autocorrelation
- for introducing no-arbitrage constraints (e.g. for a volatility surface/option prices)
What next?¹

Generate synthetic data

- Apply Wasserstein GANs (and improved versions) to
  - Different asset classes (equity, commodities, FX, ..)
  - Different time-scales (daily to intraday to high-frequency)

Use synthetic data to solve financial problems²

- Scenario generator
- Reinforcement Learning
  - Train trading strategies on artificial data
  - Evaluate them on real data
- Explainable Artificial Intelligence³
  - Explain behaviour of neural networks
  - Accomodate unforeseen data

Improve data generation

- Learn desired features of the data based on applications
- Privacy considerations
- Transfer learning

¹ EU H2020 Fintech Topic: ICT-35-2018
EU H2020 COST CA19130: Fintech and AI in Finance


Conclusion and outlook

- Generating synthetic financial data is achievable and viable with the use of GANs
- Use of synthetic data is gaining traction and new applications
- Stabilizing training is still open to improvements
- Lack of unified quantitative metric still a problem

- A new GAN, based on SAGAN, to generate synthetic financial data, was proposed

- RegGAN: New generation of GANs; Use a second discriminator as a classifier; for controlling different statistical properties of time-series such as heavy-tailedness, skewness and autocorrelation, or for introducing no-arbitrage constraints (e.g. for a vola surface)
- Outlook: Apply GANs in the frequency-domain, combine Quant-GAN with Reg-GAN
38. TJ Horan. Credit card fraud: It’s still a thing (and as big as ever), Jan 2021.