What deep generative models can do for you: Opportunities, challenges, and open questions

Giulia Fanti
Carnegie Mellon University
Common ML tools in networking

- Classification: Classifying network traffic
- Reinforcement learning: Traffic engineering
- Unsupervised methods: Clustering signals
This talk: Generative models

• What are generative models?

• Why are they relevant now?

• How can they be useful in networking?

• What are the limitations?
What is a generative model?

- Models the joint probability distribution $p(x)$ of a dataset
- Example:

$$x[t] = f(x[0:t-1], \theta) + n[t]$$

How do we pick $f$?
How to combine noise?
How are they used in the networking community?

Use **domain knowledge** to extract high-level insights.

Design **parametric model** to model those insights.

Use **data** to populate parameters.

Network traffic has **temporal patterns**.

\[ x[t] = \sin(\theta t) + n[t] \]

\[ \frac{1}{\theta} = 1 \text{ day} \]

Melamed (1993), Denneulin et al (2004), Swing, BURSE, Hierarchical bundling, Di et al (2014), ...
Problems with this approach

- Poor flexibility
  - Requires new design for every type of data

- Poor fidelity
  - Doesn’t capture properties that were not explicitly modeled
Deep generative models

Design **neural network** to produce data of the right dimensionality

Use **data** to populate parameters

\[ \theta \in \mathbb{R}^d \]
Generative Adversarial Networks (GANs): Breakthrough in generative modeling

• Prior approaches
  • Likelihood-based
  • Heavily rely on domain knowledge

• GANs
  • Adversarial learning
  • Limited a priori assumptions
Generative Adversarial Networks (GANs)
How can we use these tools in networking?

• Sharing synthetic data

• Discovering malicious inputs to black-box systems

• Understanding complex datasets
Sharing synthetic data

Use case 1

github.com/fjxmlzn/DoppelGANger
Key stumbling block: Access to data

Enterprises

Division A

Division B

Researchers

Collaborative opportunities go untapped

Unreproducible research
Limited potential
(Not a new) idea: Synthetic data models

Enterprises

Generative Model

Data Clearinghouse (ISAC, ISAO)

Generative Model

Generative Model

Researchers
Two main problems

Fidelity

Privacy

Business secrets

User data

Generative Model

Real

Generated
Existing methods

Fidelity

Privacy

Expert-designed parametric models

Machine-learned models

Anonymized, raw data

DoppelGANger
Generating synthetic time series data with GANs
What kinds of data are we interested in?

Multi-dimensional time series

With metadata

(U.S., mobile traffic)
Datasets: Networking, security, and systems

• Cluster traces
  • Google: task resource usage logs from 12k machines (2011)
  • IBM: resource usage measurements from 100k containers

• Traffic measurements
  • Wikipedia web traffic: # daily views of Wikipedia articles (2016)
  • FCC Measuring Broadband America: Internet traffic and performance measurements from consumer devices around the country
DoppelGANger: Time series generation

Metadata Generator (MLP)

(A₁, ..., Aₘ)

Min/Max Generator (MLP)

(min±max/2)

Auxiliary Discriminator

1: real
0: fake

Discriminator

1: real
0: fake

RNN

R₁,...,Rₛ

Rₜₛ₊₁,...,Rₜ
Part I: RNN + Batched Generation
Challenge: Training on high-dynamic-range time series
Part II: Auto normalization

• Standard normalization: Normalize by global min/max

• DoppelGANger: Normalize each timeseries individually
  • Store min/max as “fake” metadata
Challenge: Complex relationships in metadata

• Need to capture relation between metadata and time series
  • E.g., Cable vs Mobile users

• Straw man: Joint generator of metadata and time series
  • Problem: too hard for a single generator

Before: Single generator

![Graph showing time series min value vs count]

- Real
- DoppelGANger
Part III: Decoupled Generation, Auxiliary Discriminator

- Two stage decoupling
  - Generate metadata (using a standard MLP)
  - Generate measurements conditioned on metadata
- Auxiliary discriminator for metadata alone
Histogram of \( \frac{\text{max} + \text{min}}{2} \) per time series

Without auxiliary discriminator

With auxiliary discriminator
Putting it together

Metadata Generator (MLP) → (A₁, ..., Aₘ)

Min/Max Generator (MLP) → (min±max/2)

RNN → R₁,...,Rₛ → ...

RNN → Rₜ₋ₛ₊₁,...,Rₜ

Noise → Auxiliary Discriminator  → 1: real 0: fake

Noise → Discriminator  → 1: real 0: fake
Temporal Correlations
Microbenchmark
Predicting job failures in a compute cluster

**Downstream task**

- Train on synthetic, test on real
Evaluating privacy

• Protecting business secrets
  • Aggregate functions of the data

• User privacy
  • Differential privacy
  • Robustness against membership inference
Differentially-private SGD kills fidelity in GANs
Open questions: Synthetic data generation

• **Fidelity**
  • Long sequences of data
  • Stateful protocols

• **Privacy**
  • Differentially-private GANs
  • New privacy metrics?
Identifying malicious inputs to black-box systems

Use case 2
Black-box Devices and Systems Abound

IoT Devices  Servers / Routers  Control Units in Vehicles / Manufacturing

*NO* source code / binary / protocol format / design doc
Identifying Attack Packets is Hard

We want to identify attack packets, but do NOT have source code or system description.
Motivating example

- Packet classification
  - Vamanan et al [SIGCOMM 2010]
  - Singh et al [SIGCOMM 2013]
  - Yingchareonthawornchai et al [TON 2018]
  - Liang et al [SIGCOMM 2019]
  - Rashelbach et al [SIGCOMM 2020]
  - Many more…

Can an attacker identify many packets with high classification times?

Classification vs. Time
Random packet generation

- NeuroCuts, Liang et al [SIGCOMM 2019]

Can can we generate many, diverse slow packets?

Classification Time (ms)

Number of packets

Threshold

Fast packets

Slow packets

2,000 total packets

2,000 total packets
Common approaches

• Fuzzing tools
  • Random sampling

• Optimization of black-box functions
  • Bayesian optimization
  • Genetic algorithms
  • Simulated annealing

GANs can help!
Approach 1: Vanilla GAN

- Challenge: too little training data
AmpGAN: Training with Feedback

Random Packets → Classification decision tree

Generate packets with condition="slow" → GAN

Training Dataset
- “Fast” packets
- “Slow” packets

AmpGAN
Results

Number of packets

Number of packets

Threshold

Classification Time (ms)

Random packets

AmpGAN
Results

Fraction of "slow" packets

AmpGAN

10x jump

2.5x jump

System Calls

Genetic Algorithms
Simulated Annealing
Generalized SA
Bayesian Optimization
AmpGAN
Open questions

• Sequences of inputs

• Can we use this to optimize systems as well as finding attacks
  • E.g., CherryPick [NSDI 2017]
Extracting insights from unstructured data

Use case 3

github.com/fjxmlzn/InfoGAN-CR
Disentangled GANs

$d$ input noise

\[ z_1 \quad z_2 \quad \ldots \quad z_d \]

Generator

\[ \Rightarrow \]

$k$ factors

- Hair color
- Rotation
- Background
- Bangs

• How do \( z_i \)'s control the factors?

Vanilla GANs

\[ z_1 \quad z_2 \quad \ldots \quad z_d \]

\[ \Rightarrow \quad \text{Factor}_1 \quad \text{Factor}_2 \quad \ldots \quad \text{Factor}_k \]

Disentangled GANs

\[ \begin{align*}
  c_1 & \quad \Rightarrow \quad \text{Factor}_1 \\
  c_2 & \quad \Rightarrow \quad \text{Factor}_2 \\
  \vdots & \quad \Rightarrow \quad \vdots \\
  c_k & \quad \Rightarrow \quad \text{Factor}_k \\
  z_i & \quad S
\end{align*} \]
Examples of Disentanglement

Changing only:

- $c_1$: hair color
- $c_2$: rotation
- $c_3$: lighting
- $c_4$: background
- $c_5$: bangs

Latent codes

(CelebA dataset)

The remaining noise dimensions

(dSprites dataset)

* CelebA example is generated by InfoGAN-CR. Dsprites example is synthetic for illustration.
Our Solution: Contrastive Regularizer (CR)

- Use two latent codes \((c_1, \ldots, c_i, \ldots, c_k), (c'_1, \ldots, c'_i, \ldots, c'_k)\) to generate a pair of images.
  - Same \(i\)-th latent code
  - Equa
  - \(i = 1\)
  - \(i = 2\)
  - \(i = 3\)
  - Same shape
  - Same x-position
  - Same y-position
Intuition of Contrastive Regularizer (CR)

- Use two latent codes \((c_1, \ldots, c_i, \ldots, c_k), (c'_1, \ldots, c'_i, \ldots, c'_k)\) to generate a pair of images with the same \(i\)-th latent code.

\[
i = 1
\]

\[
i = 2 \quad \text{Generator (} G \text{)}
\]

\[
i = 3
\]

Contrastive Regularizer (CR)

Classification task!
InfoGAN-CR

InfoGAN-CR loss: \[ \min_{G,Q,H} \max_{D} \mathcal{L}_{adv}(G,D) - \lambda I(G,Q) - \alpha L_c(G,H) \]

- Input Noise: \( z \in \mathbb{R}^d \)
- Latent Factors: \( c \in \mathbb{R}^k \)

Key open question

• Can disentanglement help us make sense of networking data?
  • Reverse-engineer protocols
  • Categorize complex data patterns
  • Has not been tried in networking domain

• Time series disentanglement
  • Himberg, Hyvärinen, Esposito (2004)
Take-home messages

• Deep generative models show promise for networking applications
  • Synthetic data generation
  • Identifying malicious packets for black-box systems
  • Extracting structural insights from data

• They often cannot be applied off the shelf
  • New architectures
  • Data pre-processing pipelines
  • Training mechanisms