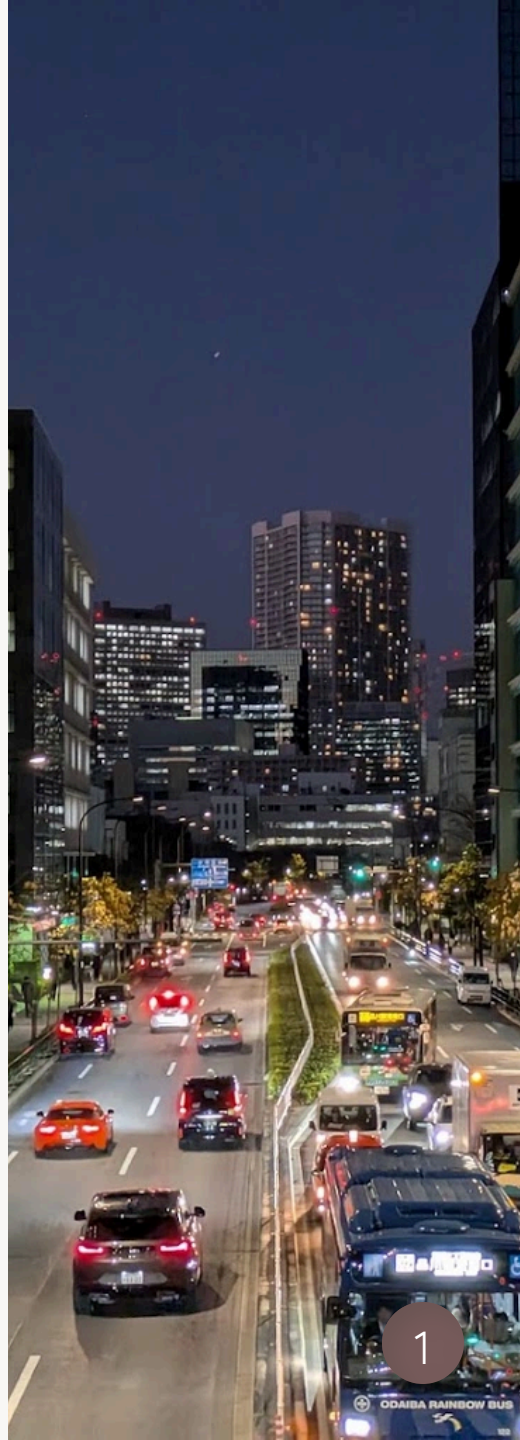
# 論文紹介: Generative Adversarial Nets

島内研究室 2025年3月12日 輪読会

発表者: 多田 瑛貴
(公立はこだて未来大学 複雑系知能学科 複雑系コース)



# 書誌情報

**Generative Adversarial Nets**
Generative Adversarial Networks (GANs, 敵対的生成ネットワーク) の提案

著者: Ian J. Goodfellow et al.

初出: Advances in Neural Information Processing Systems 27 (NIPS 2014)
NeurIPS Proceeding

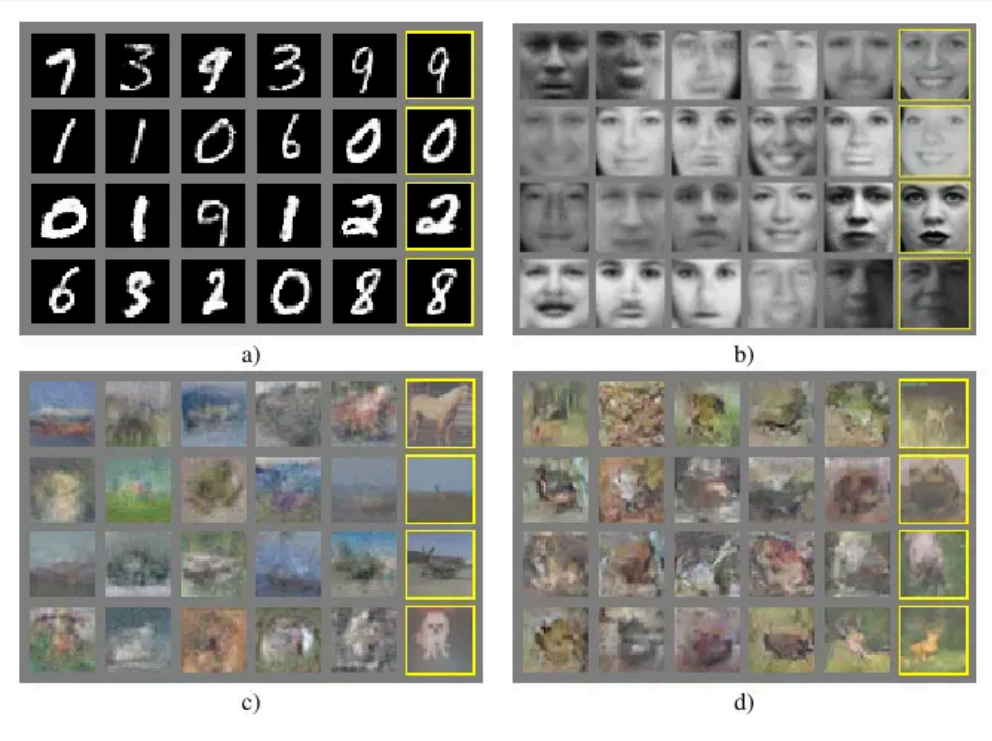
NeurIPS/NIPS (旧称) は、機械学習分野で権威ある国際会議の一つ

Google Scholar Metrics では、Artificial Intelligence #1, Engineering & Computer Science #2 (2025-03-09)

# 提案手法の概要

**Generative Adversarial Nets (GANs)** の提案
2つのニューラルネットワークを同時に学習
対象ドメインのデータ (画像など) の
生成モデルを構築する

- Generative model $G$:
  データの分布を再現する生成モデル
- Discriminative model $D$:
  入力データが$G$から生成されたものか
  教師データから抽出されたのかを
  識別するモデル



## 背景知識: 統計的学習

観測データ $x_{1:n}$ について、その生成源となる確率分布を $p_{\text{data}}$ とする
$x_{1:n}$ から $p_{\text{data}}$ を推定することを**統計的学習**と呼ぶ

一般に$p_{\text{data}}$は知ることができないので
パラメータ $\phi$ をもつ確率分布 $p(x|\phi)$ を仮定し、近似していく

例: 最尤推定 尤度 $p(x_{1:n}|\phi) = \prod_{i=1}^{n} p(x_i|\phi)$ を最大化するパラメータ$\phi$を求める

$$\phi^* = \arg\max_{\phi} \prod_{i=1}^{n} p(x_i|\phi) = \arg\max_{\phi} \sum_{i=1}^{n} \log p(x_i|\phi)$$

## 背景知識: 生成モデル

対象ドメインのデータ (画像、テキストなど) について
その生成源となる確率分布 $p_{\text{data}}$ を近似する $p(x)$ を推定し
新たなデータを生成する**生成モデル**を構築する

$x$ は入力データ、$C$ は条件として関係を示すと

$$x \sim p(x|C)$$

新たなデータを生成することを目的としなくとも
より広義に"データの生成源となる確率分布を近似する"モデルをひと括りに
生成モデルと呼ぶ場合がある (ナイーブベイズ分類器など)

## 提案手法

2つのニューラルネットワークを同時に学習

- Generative model (生成器) $G$: データの分布を再現する生成モデル
- Discriminative model (識別器) $D$: 入力データが$G$から生成されたものか、 wデータから抽出されたのかを判定するモデル

生成モデルとしては、データの確率分布を近似した$G$を利用する

$G$は$D$を騙す ($D$の誤答率を高める) ように学習し
$D$は判別精度を高めるよう学習する

論文中では$G$と$D$をそれぞれ「貨幣偽造者」と「警察」に例えるアナロジーが用いられている
貨幣偽造者$G$ は警察$D$にばれないように貨幣を偽造するが、警察は偽造貨幣を見破ろうとする

## 従来手法 (一部省略)

- データの確率分布をパラメトリックに定義し
  尤度を最大化するモデルを推定する方針は存在していたが
  勾配の計算は(特に高次元空間において)非常に困難
  Deep Boltzmann Machineなど
- そこで、誤差逆伝播により勾配を計算することで
  尤度を明示的に計算せず学習するアプローチも提案されている
  代表として、Generative Stochastic Networksはマルコフ連鎖に基づく手法
  GANsはその中でも、マルコフ連鎖を用いない手法である

- 識別器を用いて生成モデルを構築するアプローチは存在するが
  深層モデルでは適用が難しかった
- 2つのモデルを敵対的に学習するアイデアは存在するが
  目的や最適化問題の方針が大きく異なる

  代表として、Predictability Minimizationはモデルの学習を目的としない

# GANsの手法

## Adversarial Nets の構築

生成器 $G(z;\theta_g)$ (分布を$p_g$とする)

- 多層パーセプトロンによるNNであり、パラメータ $\theta_g$ を学習
- 入力: ノイズ $z$ ただし$z \sim p_z(z)$
- 出力: データ
- 以上の関係は、$G$ がノイズ $p_z(z)$ からデータ空間に写像している、といえる
- $p_z$はデータ空間に対して低次元

識別器: $D(x;\theta_d)$

- 多層パーセプトロンによるNNであり、パラメータ $\theta_d$ を学習
- 入力: データ
- 出力: スカラー値、データの由来が$p_g$ではなく$p_{\text{data}}$である確率

## Adversarial Nets の学習

$G$と$D$の学習は、以下の最適化問題を解くことで行われる

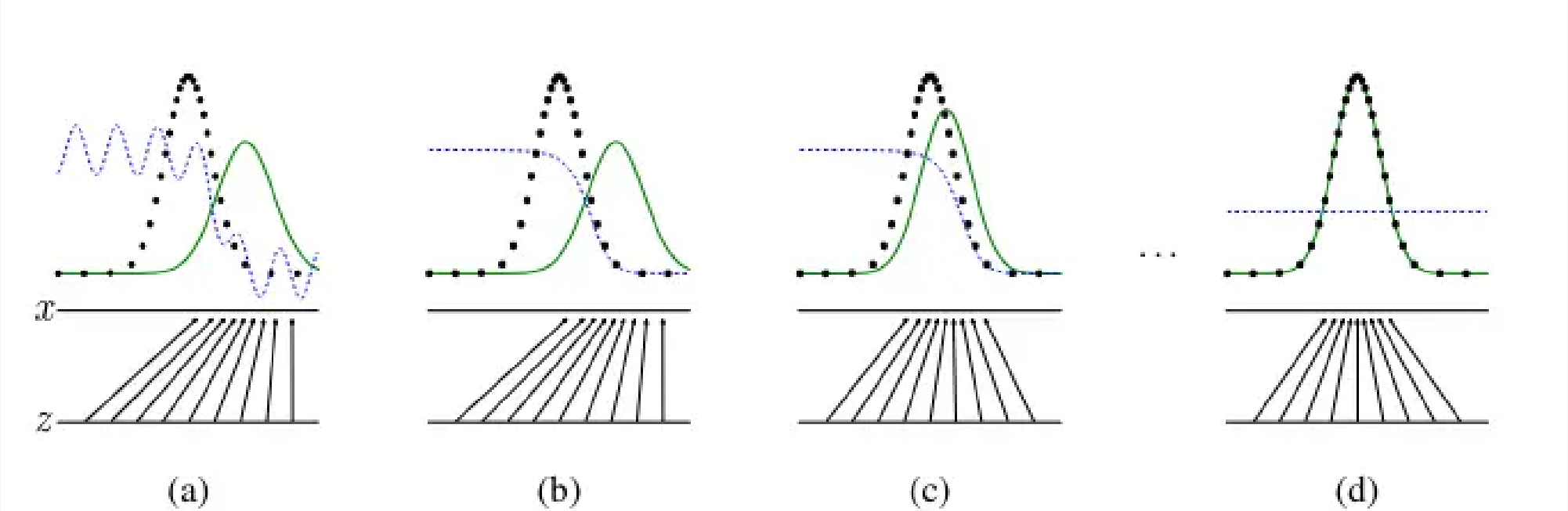
$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$$

右辺は、"$p_{\text{data}}$に対する$D$の対数の期待値" + "$p_g$に対する$(1-D)$の対数の期待値"
$V$は、$D$の精度が高いほど大きくなる

$V$を最大化しようとする$D$と、最小化する$G$で **敵対的(Adversarial)** に学習する
→ $G$の生成能力を高める

この方針は、$D$と$G$によるミニマックスゲームとして解釈できる

# 学習のイメージ




低次元の$z$空間から高次元の$x$空間にマップされる

黒点線: $p_{\text{data}}$, 緑実線: $p_g$, 青点線: $p_g$に対する$D$の出力

## アルゴリズム概要

以下の処理を反復して行う

- $D$のパラメータを**k回**更新する ($G$は固定) k回行う意義は後述
- $G$のパラメータを1回更新する ($D$は固定)

- $D$をk回更新する ($G$は固定)

## $D$の更新

一度の更新につき

- $p_g$から$m$個のミニバッチ $\{z^{(1)}, \ldots, z^{(m)}\}$ を生成
- $p_{\text{data}}$から$m$個のミニバッチ $\{x^{(1)}, \ldots, x^{(m)}\}$ を抽出

パラメータ$\theta_d$に対して以下の勾配を定義し、**上昇**方向に更新する

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[ \log D(x^{(i)}) + \log(1 - D(G(z^{(i)}))) \right]$$

- $G$を1回更新する ($D$は固定)

## $G$の更新

一度の更新につき

- $p_g$から$m$個のミニバッチ $\{z^{(1)}, \ldots, z^{(m)}\}$ を生成

パラメータ$\theta_g$に対して以下の勾配を定義し、**下降**方向に更新する

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(z^{(i)})))$$

# 反復処理について

- $D$をk回更新する ($G$は固定)
- $G$を1回更新する ($D$は固定)

なぜ一度の反復で、$D$を繰り返し更新するのか？
→与えられた$G$に対して、$D$を可能な限り最適化させるため

$D$が追いつかないほど$G$が過剰な学習をしている場合
$z$が少数の$x$にのみマップされ、$p_{\text{data}}$を適切に表現しない可能性がある
(the Helvetica scenarioと呼ばれる)

本来は$D$が完全に最適化されていることが理想的だが
計算コスト面で現実的ではなく、また過学習のリスクが発生しうる

## 理論的背景

最適化問題

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$$

理論的な裏付けのため、以下を確かめる

1. この最適化問題が実際に$p_g = p_{\text{data}}$を導くのか
2. 前述の学習アルゴリズムが実際に最適化問題を解くのか

## 1. 最適化問題が実際に$p_g = p_{\text{data}}$を導くのか

**補題.** $G$ を固定したとき、最適な $D$ は以下のとおりである

$$D^*(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}$$

**証明.** 目的関数は次のように表せる

$$\begin{aligned} V(G, D) &= \int_x p_{\text{data}}(x) \log D(x) dx + \int_x p_g(x) \log(1 - D(x)) dx \\ &= \int_x p_{\text{data}}(x) \log D(x) + p_g(x) \log(1 - D(x)) dx \end{aligned}$$

証明 (続き).

$$V(G,D) = \int_x p_{\text{data}}(x) \log D(x) + p_g(x) \log(1 - D(x)) dx$$

$(a, b) \in \mathbb{R}^2 \backslash \{0, 0\}$ に対して、$f(y) = a \log y + b \log(1 - y)$ (ただし $y \in [0, 1]$) は $y = \frac{a}{a+b}$ で最大値を取ることが知られている

ここでは、$\text{Supp}(p_{\text{data}}) \cup \text{Supp}(p_g)$ である場合のみ考えれば良い
(つまり$p_{\text{data}}(x)$ と $p_g(x)$ が同時に0になる場合は考えなくて良い)

したがって、$V$は$D(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}$ のとき最大化される

**証明終**

ここで、与えられた$G$に対して
解くべき最適化問題は、以下の$C(G)$の最小化問題と等価である

$$
\begin{aligned}
C(G) &= \max_D V(D, G) \\
&= \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D^*(x)] + \mathbb{E}_{x \sim p_g(x)}[\log(1 - D^*(x))] \\
&= \mathbb{E}_{x \sim p_{\text{data}}(x)} \left[ \log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)} \right] + \mathbb{E}_{x \sim p_g(x)} \left[ \log \frac{p_g(x)}{p_{\text{data}}(x) + p_g(x)} \right]
\end{aligned}
$$

## 背景知識: KLダイバージェンス

確率分布 $p$ と $q$ の間のKLダイバージェンスは以下のように定義される
(ただし、$\int p(x)dx = \int q(x)dx = 1$)

$$\begin{aligned}\mathrm{KL}(p||q) &= \int p(x) \log \frac{p(x)}{q(x)} dx \\ &= \mathbb{E}_{x \sim p(x)} \left[ \log \frac{p(x)}{q(x)} \right]\end{aligned}$$

KLダイバージェンスは、2つの分布の違いを定量的に表す指標であり
以下の性質が知られている

- $\mathrm{KL}(p||q) \geq 0$
- $\mathrm{KL}(p||q) = 0 \Leftrightarrow p = q$ (同じ分布なら、違いは0)

**定理.** $G$ および $D$ が最適である、
つまり$C(G)$が最小化されるのは$p_g = p_{\text{data}}$ のときのみであり
そのとき $C(G) = -\log 4$ である

**証明.** $C(G)$は、KLダイバージェンスを導入し以下のように表せる 導出は後述

$$C(G) = -\log 4 + \mathrm{KL}(p_{\text{data}} || \frac{p_{\text{data}} + p_g}{2}) + \mathrm{KL}(p_g || \frac{p_{\text{data}} + p_g}{2})$$

KLダイバージェンスの性質より、$C(G)$は最小値 $-\log 4$ を取るのは
$p_g = p_{\text{data}}$ のときのみである

**証明終**

## 導出の詳細

$$
\begin{aligned}
C(G) &= \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x)+p_g(x)}\right] + \mathbb{E}_{x\sim p_g(x)}\left[\log \frac{p_g(x)}{p_{\text{data}}(x)+p_g(x)}\right] \\
&= \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\log \frac{1}{2}\cdot\frac{p_{\text{data}}(x)}{(p_{\text{data}}(x)+p_g(x))/2}\right] + \mathbb{E}_{x\sim p_g(x)}\left[\log \frac{1}{2}\cdot\frac{p_g(x)}{(p_{\text{data}}(x)+p_g(x))/2}\right] \\
&= \mathbb{E}_{x\sim p_{\text{data}}(x)}[-\log 2] + \mathbb{E}_{x\sim p_g(x)}[-\log 2] \\
&+ \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\log \frac{p_{\text{data}}(x)}{(p_{\text{data}}(x)+p_g(x))/2}\right] + \mathbb{E}_{x\sim p_g(x)}\left[\log \frac{p_g(x)}{(p_{\text{data}}(x)+p_g(x))/2}\right] \\
&= \mathbb{E}_{x\sim p_{\text{data}}(x)}[-\log 2] + \mathbb{E}_{x\sim p_g(x)}[-\log 2] + \mathrm{KL}(p_{\text{data}}||\frac{p_{\text{data}}+p_g}{2}) + \mathrm{KL}(p_g||\frac{p_{\text{data}}+p_g}{2}) \\
&= -\log 4 + \mathrm{KL}(p_{\text{data}}||\frac{p_{\text{data}}+p_g}{2}) + \mathrm{KL}(p_g||\frac{p_{\text{data}}+p_g}{2})
\end{aligned}
$$

平均分布を導入するのは、全体の和を1にするため

## 2. 学習アルゴリズムが実際に最適化問題を解くのか

**定理.** 各反復処理において

- $D$は更新により、与えられた$G$に対する最適な解$D^*_G(x)$に到達可能
- $G$は更新を経て、以下の値$(=V(G,D))$をより小さくしていく

$$\mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D^*_G(x)] + \mathbb{E}_{x\sim p_g(x)}[\log(1-D^*_G(x))]$$

これを経て、$p_g$はいずれ$p_{\text{data}}$に収束する

$$\mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D_G^*(x)] + \mathbb{E}_{x\sim p_g(x)}[\log(1 - D_G^*(x))]$$

**証明.** 以上の式を満たす$p_g$を用いて$V(G, D) = U(p_g, D)$を定義する
この時、$U$は$p_g$に対して凸関数である (省略)

つまり、$U(p_g, D)$は$D$が最適である場合のもとでの$G$に対する凸関数群

よって、最適な$D$のもと、(理想的には) 唯一の解が存在する

**証明終**

実際は、GANsが多層パーセプトロンに基づいている以上
局所解が存在する場合がある

一方で、多層パーセプトロンに基づく手法は
経験的にとても良い性能を示すことから、十分に有効であると言える

# 実験

# 学習するモデル (多層パーセプトロン) の仕様

## 生成器

- 活性化関数: ReLU, Sigmoidの組み合わせ
- ノイズは入力のみから与えられる

  生成器にもDropoutの適用や中間層へのノイズの挿入が可能だが、実験では行っていない

## 識別器

- 活性化関数: Maxout
- Dropoutを適用

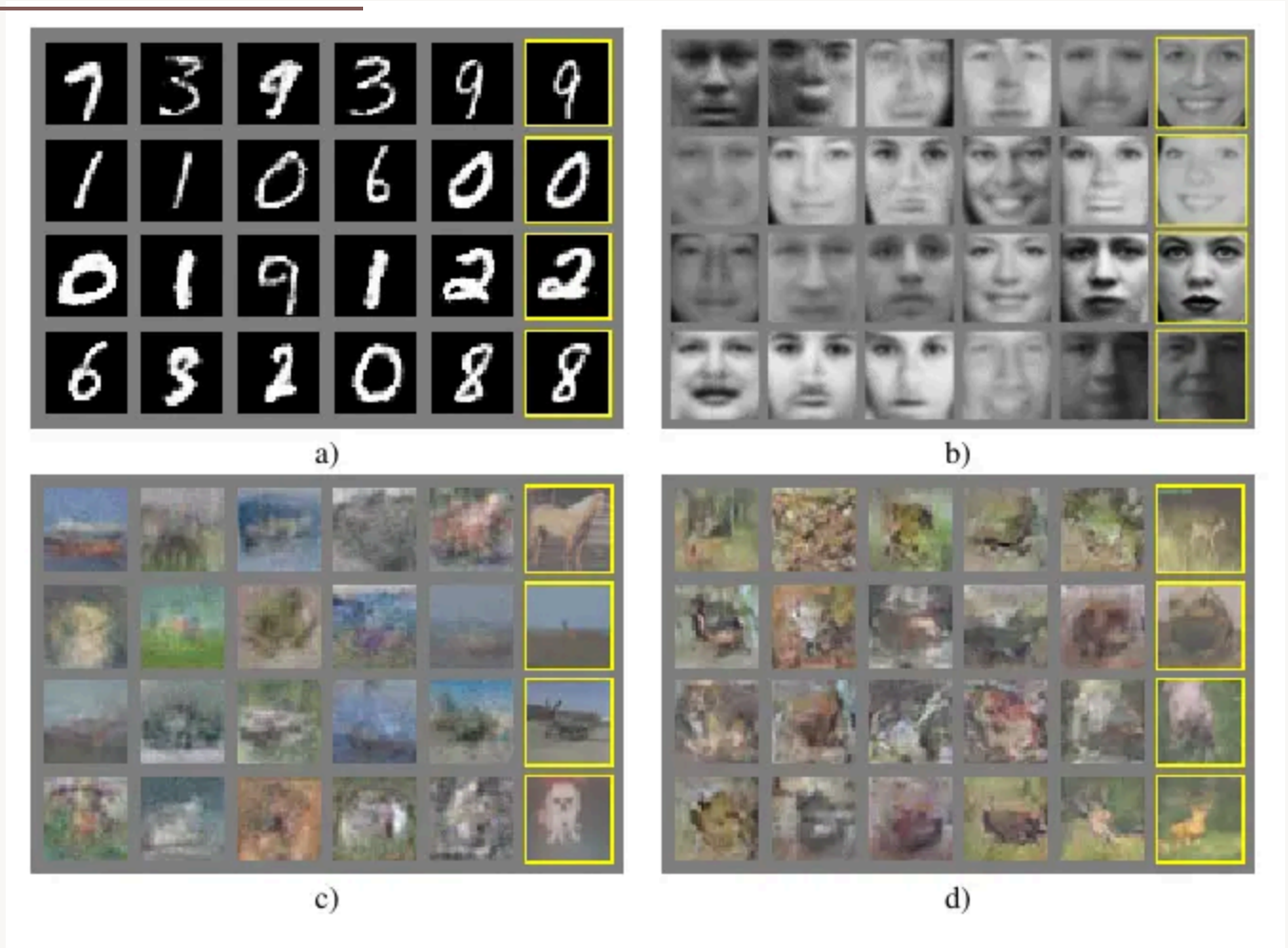
  MaxoutはDropoutを適用したモデルで用いられる手法の一つで、Goodfellowが提案

## 実験1. 実際の生成結果

以下のデータセットに対して、GANsを適用し生成モデルを構築

- MNIST: 手書き数字
- Toronto Face Database (TFD): 顔画像
- CIFAR-10: 10クラスの画像

# 実験1. 実際の生成結果




右の黄色枠の画像は、左隣の生成データと最も近い実データを表す

実データ自体を単に記憶するモデルでないことを示すことが目的

# 実験2. 他生成モデルとの性能比較

$p_g$の分布を推定したのち
実データ$x_{1:n}$に対して、対数尤度$\sum_{i=1}^{n} \log p_g(x_i)$を計算し
$p_g$が実データをよく説明できているかを他モデルと比較

$p_g$の分布は直接算出できないため、$G$から生成されたサンプルから
Gaussian Parzen window を用いて推定

Parzen window (カーネル密度推定): 確率関数の分布を推定するノンパラメトリック手法

Gaussian Parzen window: ガウス分布に基づくParzen window

やや分散が大きく、高次元空間には良い結果を示さないこともあるが、現状はこの手法が最適

以下のデータセットに対して$G$を学習
対数尤度の平均および標準誤差を計算

- MNIST: 手書き数字
- Toronto Face Database (TFD): 顔画像

  TFDにおいては、交差検証を実施

  それぞれの分割での異なる $\sigma$ のもと対数尤度を計算

# 実験2. 他生成モデルとの性能比較

| Model | MNIST | TFD |
|---|---|---|
| DBN [3] | $138 \pm 2$ | $1909 \pm 66$ |
| Stacked CAE [3] | $121 \pm 1.6$ | $\mathbf{2110 \pm 50}$ |
| Deep GSN [5] | $214 \pm 1.1$ | $1890 \pm 29$ |
| Adversarial nets | $\mathbf{225 \pm 2}$ | $\mathbf{2057 \pm 26}$ |

平均対数尤度 ± 標準誤差

# 実験3. 連続的なノイズによる生成

$z$空間上の2点間で線形補間し抽出した連続的なノイズを用いて生成

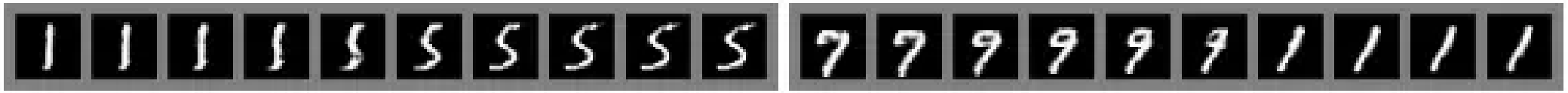


Figure 3: Digits obtained by linearly interpolating between coordinates in $z$ space of the full model.

## 欠点

- $p_g$ を明確に知ることはできない
- $D$が$G$の学習と適切に同期するよう気にかける必要がある

## 利点

計算効率上の利点としては

- マルコフ連鎖を用いない
- 勾配の導出が誤差逆伝播で完結する
- 学習時の推論が不要
- モデルの柔軟性が高い

統計的な利点として

- GANsの生成器は、実データを直接取り入れて学習するわけでなく
  あくまで識別器を通して計算される勾配を用いている
  → 実データがそのままコピーされることがない

- 尖った分布など、極端な分布も表現できる
  マルコフ連鎖由来の手法では表現が難しかった

| | Deep directed graphical models | Deep undirected graphical models | Generative autoencoders | Adversarial models |
|---|---|---|---|---|
| Training | Inference needed during training. | Inference needed during training. MCMC needed to approximate partition function gradient. | Enforced tradeoff between mixing and power of reconstruction generation | Synchronizing the discriminator with the generator. Helvetica. |
| Inference | Learned approximate inference | Variational inference | MCMC-based inference | Learned approximate inference |
| Sampling | No difficulties | Requires Markov chain | Requires Markov chain | No difficulties |
| Evaluating $p(x)$ | Intractable, may be approximated with AIS | Intractable, may be approximated with AIS | Not explicitly represented, may be approximated with Parzen density estimation | Not explicitly represented, may be approximated with Parzen density estimation |
| Model design | Models need to be designed to work with the desired inference scheme — some inference schemes support similar model families as GANs | Careful design needed to ensure multiple properties | Any differentiable function is theoretically permitted | Any differentiable function is theoretically permitted |

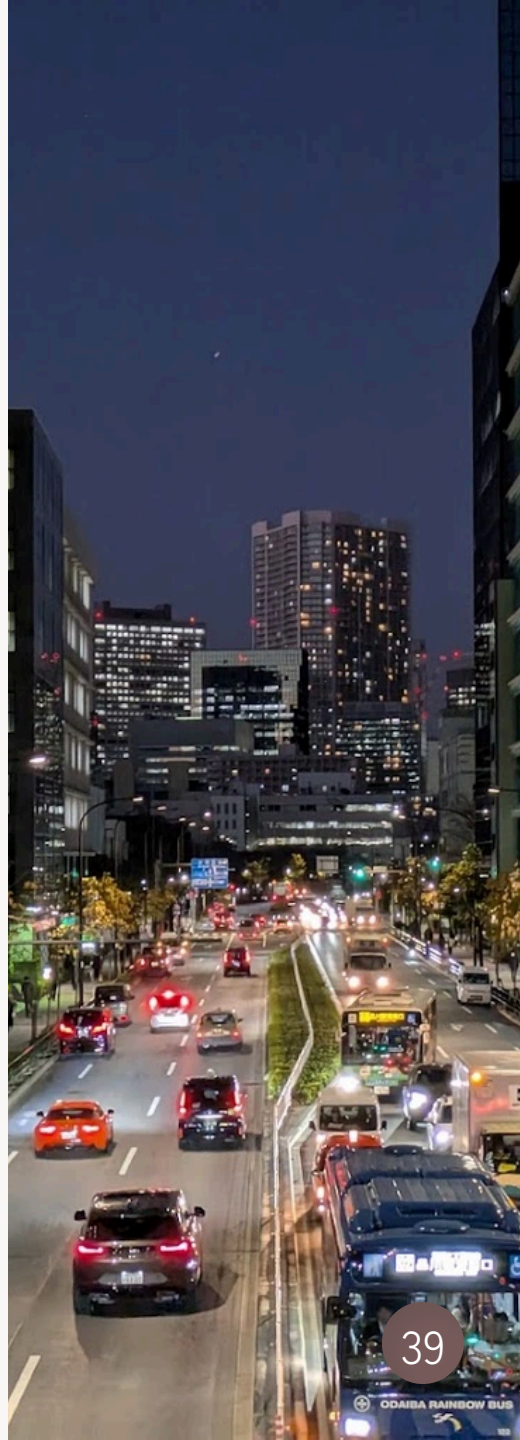
## まとめ

生成器と識別器の **敵対的(Adversarial)** な学習により
生成モデルを構築

- 誤差逆伝播を用いた勾配のみで学習できる点で
  従来手法と比較し学習がシンプルであり
  計算効率の改善が見込める
- 極端な分布への表現力も見込める

実験では、画像に対して適用

- 他の生成モデルに対して優位性があるとは言い切れないが
  対数尤度を用いた性能評価では良い結果を得られている



## 本発表で出来ていないこと・疑問

- Related Worksの詳細な紹介 (第2章)
- GANsの拡張の紹介 (第7章)

## 参考文献 (発表論文を除く)


- 佐藤一誠. ノンパラメトリックベイズ 点過程と統計的機械学習の数理. 講談社, 2016, 160p, (機械学習プロフェッショナルシリーズ).
- 岡野原大輔. 生成モデルは世界をどのように理解しているのか. 統計数理シンポジウム, 株式会社Preferred Networks, 2023-05-25. (参照 2025-03-10).
- 吾妻幸長. はじめてのディープラーニング2. SBクリエイティブ株式会社, 2020, 330p.