知能プログラミング演習 I

第 8 回: 発展的な話題

梅津 佑太
2 号館 404A: umezu.yuta@nitech.ac.jp

課題のダウンロード

前回作ったディレクトリに移動して今日の課題のダウンロードと解凍

step1: `cd ./DLL`

step2: `wget http://www-als.ics.nitech.ac.jp/~umezu/DLL19/Lec8.zip`

step3: `unzip Lec8.zip`

✓ **まだ** DLL **のフォルダを作ってない人は**, step1 **の前に**
`mkdir -p DLL`
でフォルダを作成する

今日の講義内容

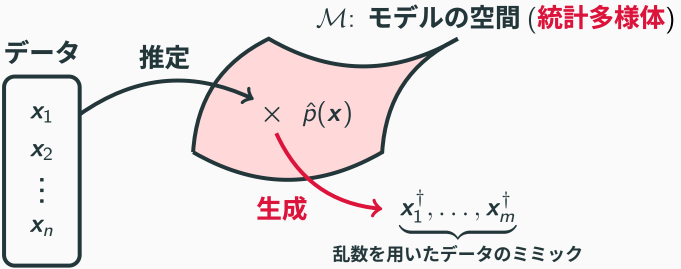
1. **深層学習に関する最近の話題**
    1.1 **深層学習と生成モデル**
    1.2 **深層学習の理論**

1.1 深層学習と生成モデル

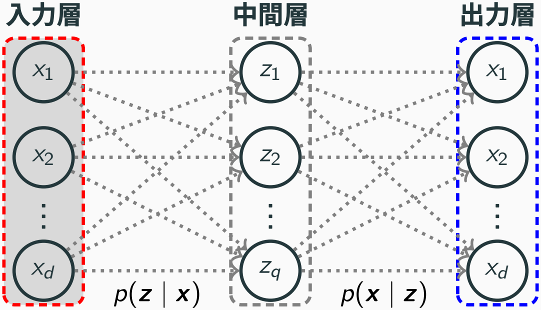
生成モデル

- ニューラルネットワークの学習
  - ✓ 入出力関係 (モデル) をよく記述するパラメータ $\boldsymbol{w}$ を推定
  - ✓ つまり, 入力と出力関係をモデリング

- 入力 $\boldsymbol{x}$ を生み出す確率モデル $p(\boldsymbol{x})$ を推定できれば, 推定した $\hat{p}(\boldsymbol{x})$ から別の標本 $\boldsymbol{x}^\dagger$ をサンプリング可能 $\Rightarrow p(\boldsymbol{x}) = p(\boldsymbol{x}; \boldsymbol{w})$ の**最尤推定**



変分オートエンコーダ



- ニューラルネットワークに z を提示して，入力 x と似た出力を生成

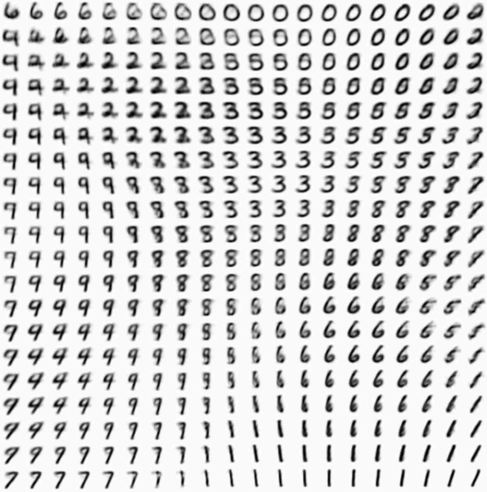
$$p(\boldsymbol{x}) = \int p(\boldsymbol{x} \mid \boldsymbol{z}) Q(\mathrm{d}\boldsymbol{z}) = \mathbb{E}_{\boldsymbol{z} \sim Q}[p(\boldsymbol{x} \mid \boldsymbol{z})], \; \mathbb{E}[\boldsymbol{x} \mid \boldsymbol{z}] = f(\boldsymbol{z}; \boldsymbol{w})$$

- $z \sim Q$ が x を生成するとき $Q(z) \approx P(z \mid x)$ であると予想してみる

$$\Rightarrow \hat{\boldsymbol{w}} = \arg\min \mathrm{KL}[Q(\boldsymbol{z}) \| P(\boldsymbol{z} \mid \boldsymbol{x})]$$

変分オートエンコーダ

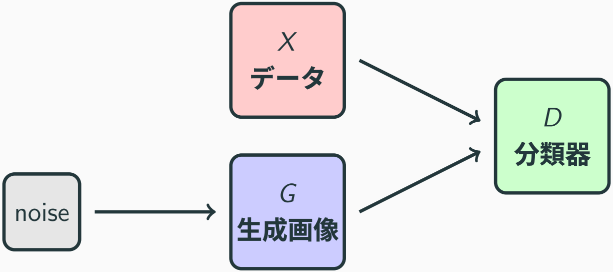
(a) Learned Frey Face manifold



(b) Learned MNIST manifold

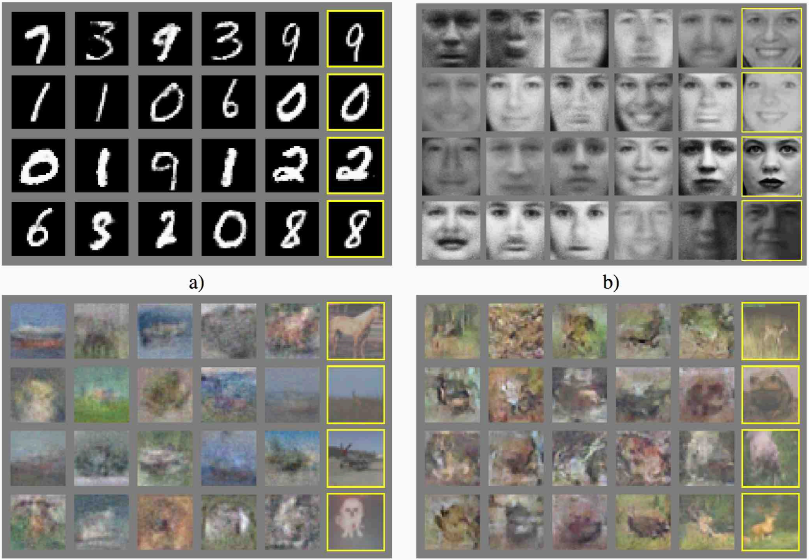
Generative Adversarial Net

- データの生成と識別を繰り返すことで，手元にあるデータと似たようなデータを発生



$$\min_G \max_D V(D,G) = \underbrace{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]}_{\text{本物の画像を本物と判別する確率}} + \underbrace{\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))]}_{\text{偽物の画像を偽物と判別する確率}}$$

Generative Adversarial Net



a) b)

c) d)

Generative Adversarial Net

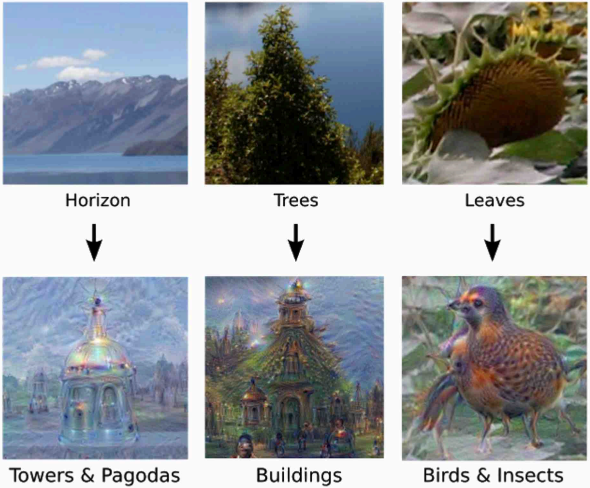
- Goodfellow et al. (2014) **以降**, adversarial net **の研究が大流行り**
  - ✓ Conditional GAN: Mirza and Osindero (2014)
  - ✓ DCGAN: Radford et al. (2015)
  - ✓ Adversarial Autoencoders: Makhzani et al. (2015)
  - ✓ AdaGAN: Tolstikhin et al. (2017)
  - ✓ StackGAN: Zhang et al. (2017)
  - ✓ Wasserstein GAN: Arjovsky et al. (2017)

"The most interesting idea in the last 10 years in ML, in my opinion."

– Yann LeCun

Deep Dream

- 深層ニューラルネットワークで，パラメータの代わりにデータを更新することで，不思議な画像を生成することが可能



1.2 深層学習の理論

デノイジングオートエンコーダ: DAE

- **デノイジングオートエンコーダ**: **入力の分布を** $\boldsymbol{x} \sim \pi$ **とする**
  - ✓ **入力**: **ノイズを付加した入力** $\boldsymbol{x}^\dagger = \boldsymbol{x} + \varepsilon$, $\varepsilon \sim \nu(\varepsilon) = \mathrm{N}(\mathbf{0}, \sigma^2 I)$
  - ✓ **中間層**: $\boldsymbol{x}^\dagger$ **の圧縮表現** z, **出力層**: $\boldsymbol{x}^\dagger$ **の復元表現** $\tilde{\boldsymbol{x}}$
- **デノイジングエンコーダのターゲットは次の問題と等価** [1]:

$$\min_g \mathbb{E}_\pi \mathbb{E}_\nu [\|g(\boldsymbol{x} + \varepsilon) - \boldsymbol{x}\|^2]$$

- g **による変分と** Stein **の等式を用いれば**, **最適解は以下の通り** [2]:

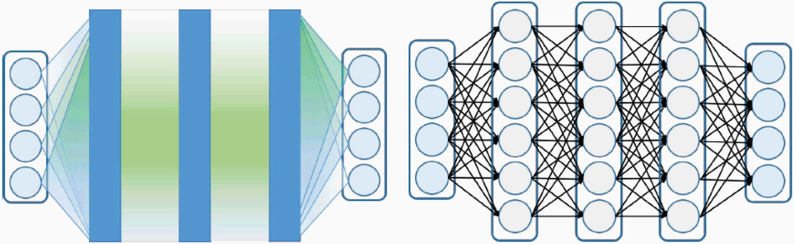
$$g^*(\boldsymbol{x}) = \boldsymbol{x} + \sigma^2 \nabla \log(\nu * \pi)(\boldsymbol{x})$$

  - ✓ $\boldsymbol{x}$ **の最適輸送が** g^* **の最適解となる**: $\sigma^2 \nabla \log(\nu * \pi)(\boldsymbol{x})$ **は** $\boldsymbol{x}$ **を最適解へ輸送するためのコスト**
  - ✓ DAE **は** Wasserstein GAN **を代表とする最適輸送に関連する最も簡単なモデル**

[1] g **は十分に広い関数を表現出来るとする**
[2] **変分を** 0 **と置いたものは** Euler-Lagrange **方程式とも呼ばれる**

深層学習 (DNN) の積分表現

- DNN は背後にある関数空間の有限次元モデルだと考える



(a) 背後にある真のモデル　　(b) モデルの有限次元近似

- 適当な条件のもと，真のモデル f^* と推定したモデル $\hat{f}$ の誤差は，適当な δ と $\varepsilon_n = o(1)$ を用いて，以下のように評価できる:

$$\|\hat{f} - f^*\|_{L^2}^2 \le 2(\delta^2 + \varepsilon_n^2)$$

ReLU は折り紙

- ReLU **の条件分岐は対称的**

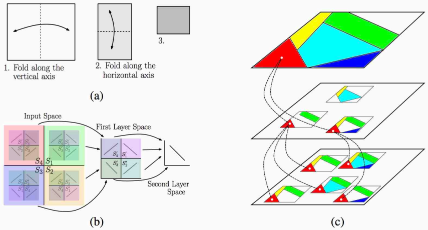


Figure 2: (a) Space folding of 2-D Euclidean space along the two axes. (b) An illustration of how the top-level partitioning (on the right) is replicated to the original input space (left). (c) Identification of regions across the layers of a deep model.

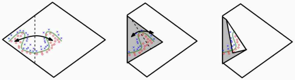


Figure 3: Space folding of 2-D space in a non-trivial way. Note how the folding can potentially identify symmetries in the boundary that it needs to learn.

- Tropical Geometry **と併せれば**, ReLU **の表現力の限界を評価できる**
  - $\Rightarrow$ ReLU **による** DNN **の限界**: $\Theta((n/d)^{(L-1)d} n^d)$