

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/Lec8.zip`

step3: `unzip Lec8.zip`

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

講義ノート更新しました.

1. 深層学習に関する最近の話題

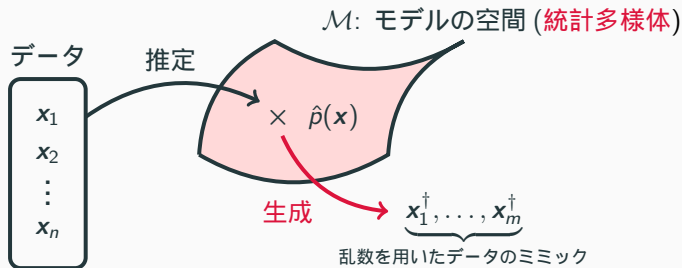
1.1 深層学習と生成モデル

1.2 深層学習の理論

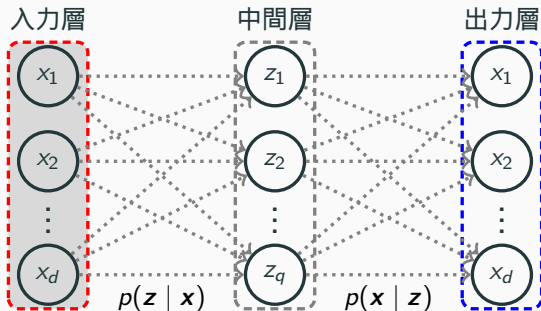
1.1 深層学習と生成モデル

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 w を推定
 - ✓ つまり, 入力と出力関係をモデリング
- 入力 x を生み出す確率モデル $p(x)$ を推定できれば, 推定した $\hat{p}(x)$ から別の標本 x^\dagger をサンプリング可能 $\Rightarrow p(x) = p(x; w)$ 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

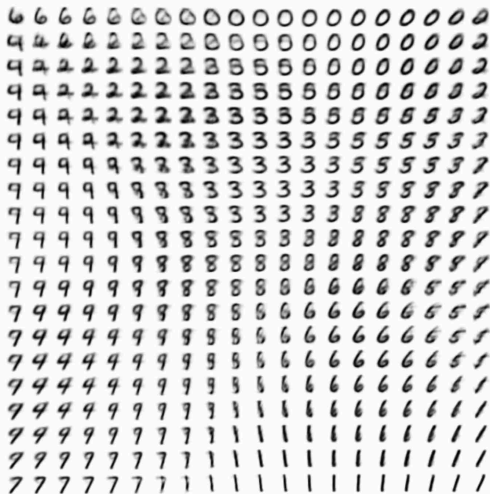
$$p(x) = \int p(x | z) Q(dz) = \mathbb{E}_{z \sim Q}[p(x | z)], \quad \mathbb{E}[x | z] = f(z; \mathbf{w})$$

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

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



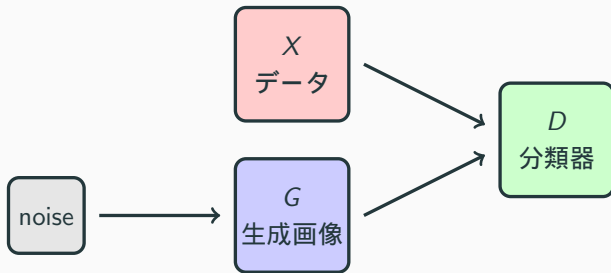
(a) Learned Frey Face manifold



(b) Learned MNIST manifold

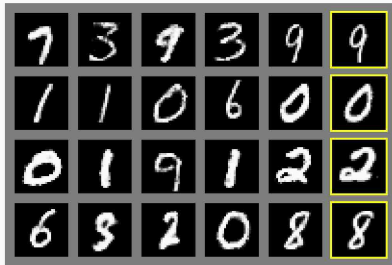
Generative Adversarial Net

- 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

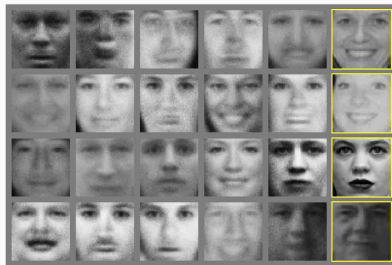


$$\min_G \max_D V(D, G) = \underbrace{\mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x})]}_{\text{本物の画像を本物と判別する確率}} + \underbrace{\mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))]}_{\text{偽物の画像を偽物と判別する確率}}$$

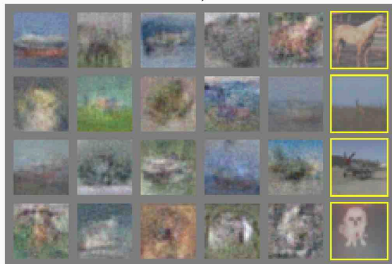
Generative Adversarial Net



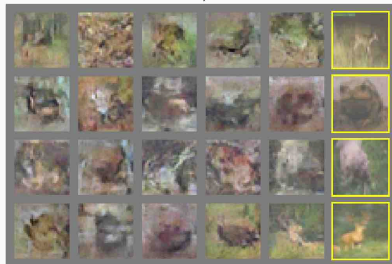
a)



b)



c)



d)

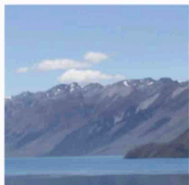
- Goodfellow et al. (2014) 以降, adversarial net の研究が大流行り
 - ✓ Conditional GAN: Mirza and Osindero (2014)
 - ✓ DCGAN: Radford et al. (2015)
 - ✓ Adversarial Autoencoders: Makhzani et al. (2015)
 - ✓ AdaGAN: Tolstikhin et al. (2017)
 - ✓ StackGAN: Zhang et al. (2017)
 - ✓ Wasserstein GAN: Arjovsky et al.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

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

– Yann LeCun

Deep Dream

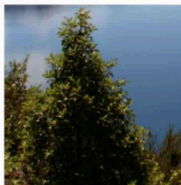
- 深層ニューラルネットワークで、パラメータの代わりにデータを更新することで、不思議な画像を生成することが可能



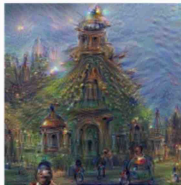
Horizon



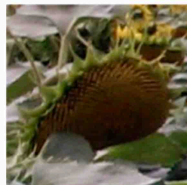
Towers & Pagodas



Trees



Buildings



Leaves



Birds & Insects

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 $\mathbf{x} \sim \pi$ とする
 - ✓ 入力: ノイズを付加した入力 $\mathbf{x}^\dagger = \mathbf{x} + \varepsilon$, $\varepsilon \sim \nu(\varepsilon) = \mathcal{N}(\mathbf{0}, \sigma^2 I)$
 - ✓ 中間層: \mathbf{x}^\dagger の圧縮表現 \mathbf{z} , 出力層: \mathbf{x}^\dagger の復元表現 $\tilde{\mathbf{x}}$
- 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:

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

- 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:

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

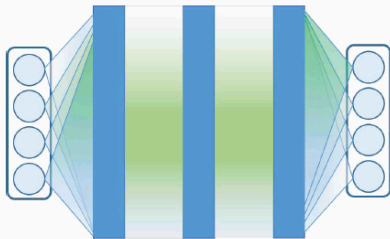
- ✓ \mathbf{x} の最適輸送が g^* の最適解となる: $\sigma^2 \nabla \log(\nu * \pi)(\mathbf{x})$ は \mathbf{x} を最適解へ輸送するためのコスト
- ✓ DAE は Wasserstein GAN を代表とする最適輸送に関連する最も簡単なモデル

¹ g は十分に広い関数を表現出来るとする

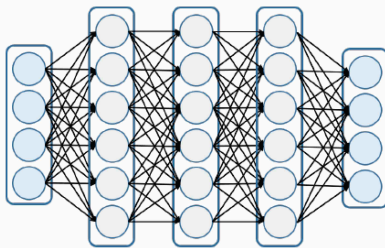
² 変分を 0 と置いたものは Euler-Lagrange 方程式とも呼ばれる

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:

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

ReLU は折り紙

- ReLU の条件分岐は対称的

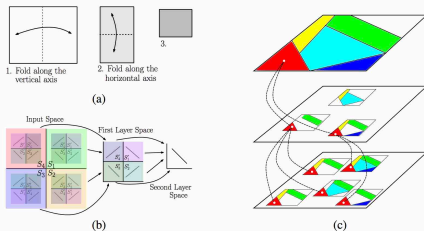


Figure 2: (a) Space folding of 2-D Euclidean space along the two axes. (b) An illustration of how the top-level partitioning (on the right) is replicated to the original input space (left). (c) Identification of regions across the layers of a deep model.

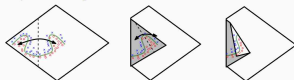


Figure 3: Space folding of 2-D space in a non-trivial way. Note how the folding can potentially identify symmetries in the boundary that it needs to learn.

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