知能プログラミング演習 I

第 4 回: オートエンコーダ

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/Lec4.zip`

step3: `unzip Lec4.zip`

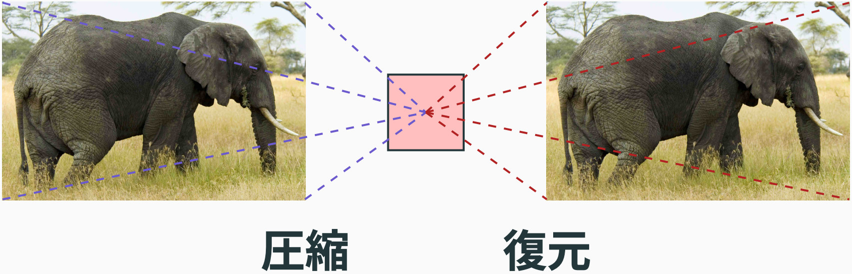
- ✓ **まだ** DLL **のフォルダを作ってない人は**, step1 **の前に**
  `mkdir -p DLL`
  でフォルダを作成する

今日の講義内容

1. **オートエンコーダと** adam **によるパラメータ推定**

圧縮と復元

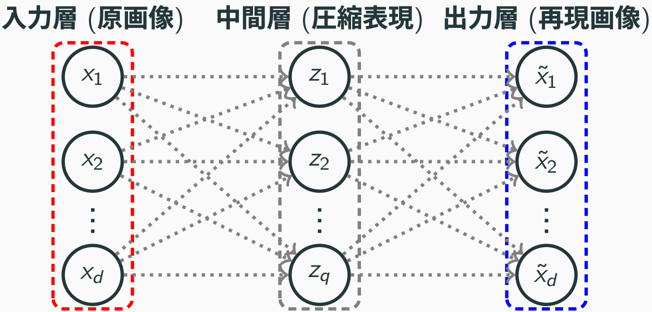
- jpeg (非可逆圧縮) や png (可逆圧縮) などは, 実際には原画像を圧縮し, 適当なサイズで復元したもの [1]
- "圧縮" とは, データの低次元表現を得るための方法, c.f., 主成分分析



[1] ちなみに, jpeg では離散コサイン変換とその逆変換, jpeg 2000 は離散ウェーブレット変換とその逆変換が用いられている

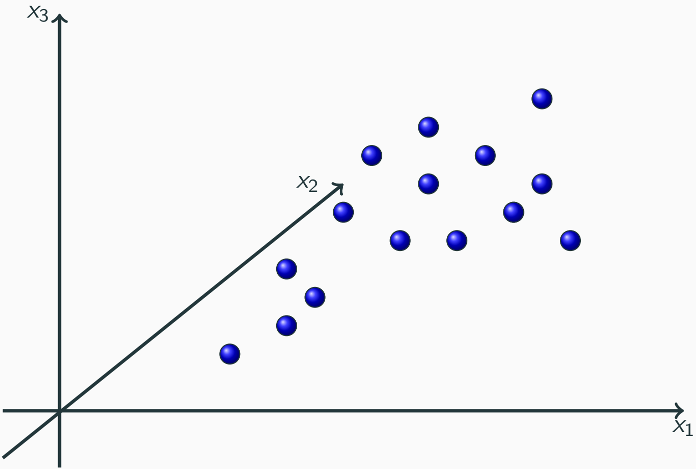
自己符号化器

- 圧縮と復元をニューラルネットワークで表すのが自己符号化器[2]
- 構造は 3 層ニューラルネットワークと同じに見えるが，出力がないので**教師なし学習**



[2]非可逆圧縮なので，当然，圧縮前 (原画像) と復元後 (再現画像) では誤差が生じる

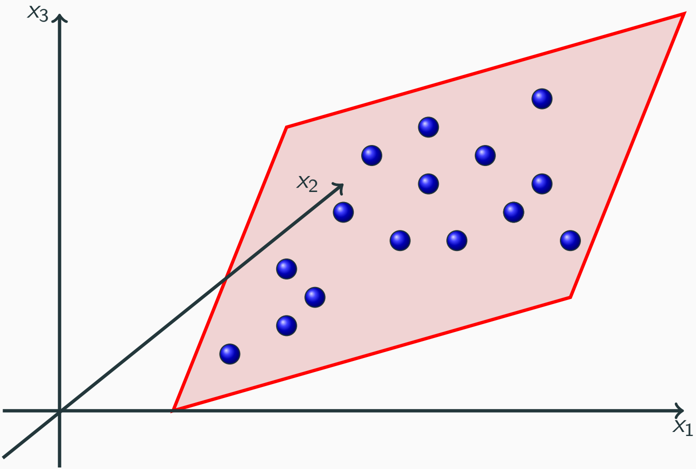
主成分分析



- **データの (線形な) 低次元表現を学習**:

$$\min_{W \in \mathbb{R}^{q \times d}} \|\boldsymbol{x} - W^\top W \boldsymbol{x}\|^2$$

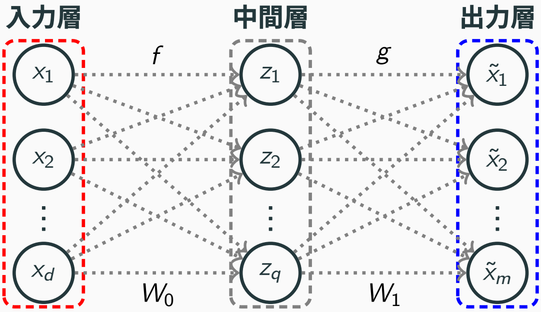
主成分分析



- **データの (線形な) 低次元表現を学習**:

$$\min_{W \in \mathbb{R}^{q \times d}} \|\boldsymbol{x} - W^\top W \boldsymbol{x}\|^2$$

オートエンコーダ



- **砂時計型ネットワーク** $(q < d)$ **によって，原画像を復元するネットワークを学習**
  - ✓ **主成分分析は** f, g **が恒等写像である場合に対応**

$$\min_{W_0, W_1} \|\boldsymbol{x} - g(W_1 f(W_0 \boldsymbol{x}))\|^2$$

オートエンコーダ

原画像をベクトルで $\boldsymbol{x} \in [0,1]^d$ と表したとき, 切片項を考慮して $(1, \boldsymbol{x}^\top)^\top$ を改めて $\boldsymbol{x}$ とすれば,

- 入力層 $\to$ 中間層 $(q < d)$

$$\boldsymbol{z} = f(W_0^\top \boldsymbol{x})$$

- 中間層 $\to$ 出力層: $(1, \boldsymbol{z}^\top)^\top$ を改めて $\boldsymbol{z}$ として,

$$\tilde{\boldsymbol{x}} = g(W_1^\top \boldsymbol{z})$$

- 誤差関数

$$E(W_0, W_1) = \|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|^2, \quad \boldsymbol{x} \in [0,1]^d, \quad (\text{ただし}, g = \mathrm{Id})$$

✓ 原画像が $\boldsymbol{x} \in \{0,1\}^d$ の 2 値画像であれば, g として softmax 関数, 誤差関数としてクロスエントロピーを用いる

✓ 逆伝播は 3 層ニューラルネットワークと同様

いろいろなオートエンコーダ

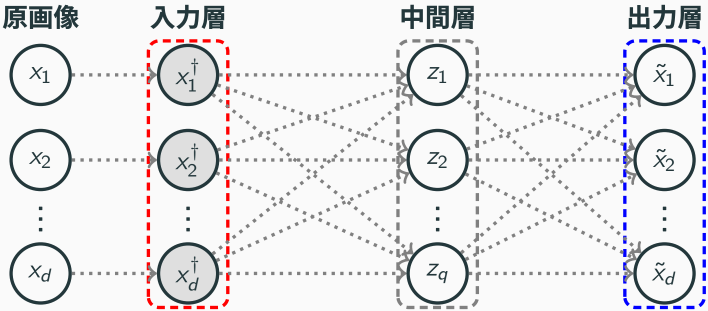
- **特徴抽出のためのオートエンコーダ**
  - ✓ **深層オートエンコーダ**
  - ✓ **積層オートエンコーダ**
  - ✓ **デノイジングオートエンコーダ**
- **生成モデルとしてのオートエンコーダ**
  - ✓ **変分オートエンコーダ** (VAE)
  - ✓ Generative Adversarial Network (GAN)

デノイジングオートエンコーダ

原画像 x に加法的ノイズを付加したものを入力として再現画像を生成

- **ノイズ** (e.g., $\nu \sim \mathrm{N}(0, \sigma^2)$) **を用いて，入力として次を用いる**:

$$\boldsymbol{x}^{\dagger} = \boldsymbol{x} + \nu \boldsymbol{x}$$



- **誤差関数は原画像と再現画像で評価する**
- **原画像の形状に対して頑健にパラメータを学習可能**

誤差逆伝播法

適当な誤差関数 $E(W_0, W_1)$ と活性化関数 g を用いれば, 逆伝播は

$$\text{出力層から中間層}: \boldsymbol{\delta}_2 = \tilde{\boldsymbol{x}} - \boldsymbol{x} \Rightarrow \frac{\partial E}{\partial W_1} = \boldsymbol{\delta}_2 \tilde{\boldsymbol{x}}^\top$$

$$\text{中間層から入力層}: \boldsymbol{\delta}_1 = \tilde{W}_1^\top \boldsymbol{\delta}_2 \odot \nabla f(W_0 \boldsymbol{x}) \Rightarrow \frac{\partial E}{\partial W_0} = \boldsymbol{\delta}_1 \boldsymbol{x}^\top$$

となる [3]. この逆伝播を利用してパラメータを更新する: 例えば, 通常の確率的勾配降下法なら

$$W_1 \leftarrow W_1 - \eta \frac{\partial E}{\partial W_1}$$

$$W_0 \leftarrow W_0 - \eta \frac{\partial E}{\partial W_0}$$

[3] $\tilde{W}_1$ は W_1 から 1 列目を取り除いた行列

adam によるパラメータの更新

- **パラメータを効率的に更新する方法**
  - ✓ **目的関数の 2 階微分を利用**: e.g., **ニュートン法**, **準ニュートン法**
  - ✓ **目的関数を評価する点の修正**: e.g, **ネステロフの加速法**
  - ✓ **パラメータの更新にデータのばらつきを利用**: e.g, **モーメンタム法**

- **adam** (Adaptive Moment Estimation): **準ニュートン法に近い形で**, **目的関数の勾配の** 2 **次モーメントを考慮したパラメータの更新方法**. **学習率を自動的に決定できる**.

パラメータ W **を以下のように更新**:

$$
\begin{aligned}
m &\leftarrow \beta_1 m + (1-\beta_1)\frac{\partial E}{\partial W} \\
v &\leftarrow \beta_2 v + (1-\beta_2)\frac{\partial E}{\partial W} \odot \frac{\partial E}{\partial W} \\
\hat{m} &\leftarrow \frac{m}{1-\beta_1^t} \\
\hat{v} &\leftarrow \frac{v}{1-\beta_2^t} \\
W &\leftarrow W - \alpha \frac{\hat{m}}{\sqrt{\hat{v}} + \varepsilon}
\end{aligned}
$$

演算は全て成分ごとに行い, m **と** v **の初期値は** 0, t **はエポック番号** $+1$ **(つまり**, $t \geq 1$**) とする** [4].

[4] **パラメータの推奨値は** $\alpha = 0.01$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$.

adam によるパラメータの更新 II

パラメータ更新の意味:

- $\alpha/(\sqrt{\hat{v}}+\varepsilon)$ **でデータのばらつきを考慮した学習率を評価**:

$$W \leftarrow W - \alpha \frac{\hat{m}}{\sqrt{\hat{v}}+\varepsilon}$$

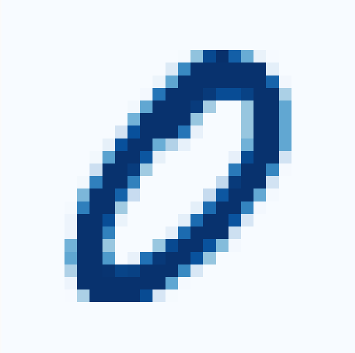
- **過去の履歴を考慮して勾配** m, v **更新**:

$$m \leftarrow \beta_1 m + (1-\beta_1)\frac{\partial E}{\partial W}$$
$$v \leftarrow \beta_2 v + (1-\beta_2)\frac{\partial E}{\partial W} \odot \frac{\partial E}{\partial W}$$

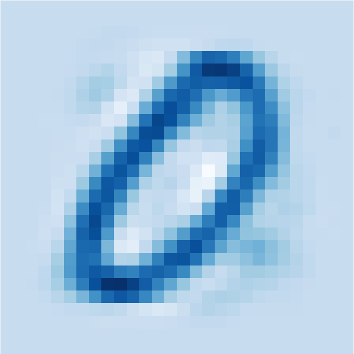
- **重み付きの更新によって得られるバイアスを修正**;

$$\hat{m} \leftarrow \frac{m}{1-\beta_1^t}, \quad \hat{v} \leftarrow \frac{v}{1-\beta_2^t}$$

オートエンコーダによる手書き文字の再現



(a) オリジナルの入力



(b) オートエンコーダで再現した入力

中間層で学習された特徴

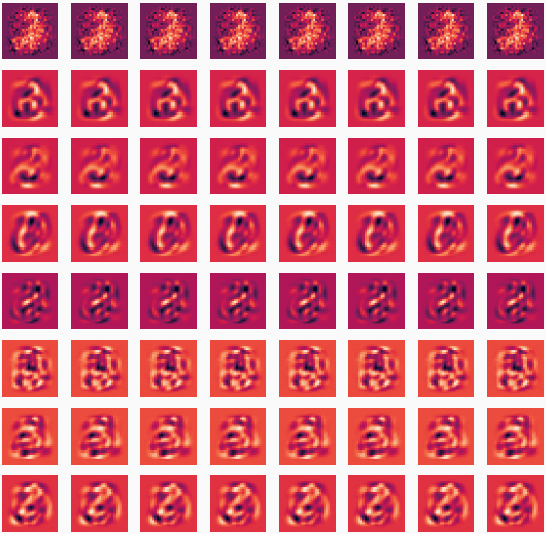


Figure 1: 中間層から出力層へのパラメータ $W_1 \in \mathbb{R}^{d\times(q+1)}$ **の可視化**.