

ニコニコAIスクール「脳型人工知能開発者入門コース」#8  
自己符号化器と生成モデル

# (なるべく)体系的に理解する深層生成モデル

長野 祥大  
2018/03/04

# 生成モデル？

## How Much Information Does the Machine Need to Predict?

Y LeCun

### ■ "Pure" Reinforcement Learning (cherry)

- ▶ The machine predicts a scalar reward given once in a while.
- ▶ **A few bits for some samples**

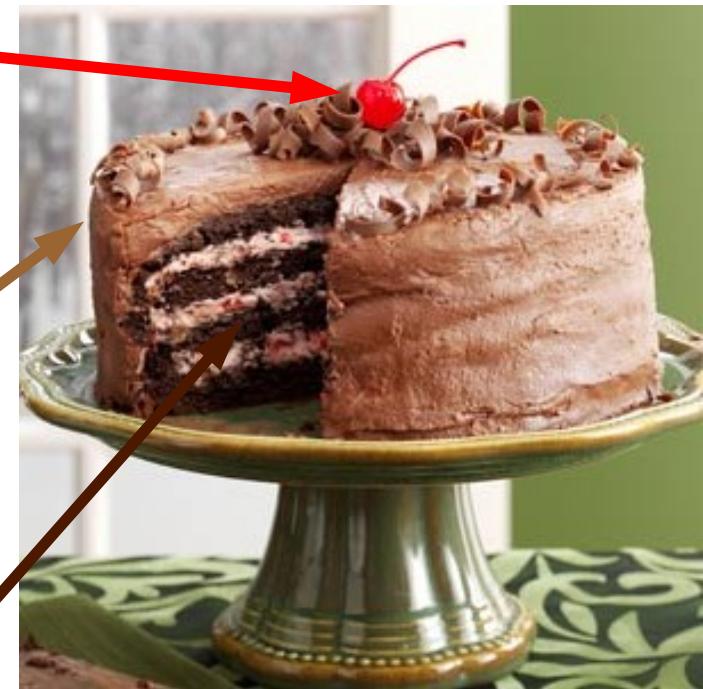
### ■ Supervised Learning (icing)

- ▶ The machine predicts a category or a few numbers for each input
- ▶ Predicting human-supplied data
- ▶ **10→10,000 bits per sample**

### ■ Unsupervised/Predictive Learning (cake)

- ▶ The machine predicts any part of its input for any observed part.
- ▶ Predicts future frames in videos
- ▶ **Millions of bits per sample**

■ (Yes, I know, this picture is slightly offensive to RL folks. But I'll make it up)



# 生成モデルと識別モデル

## 識別モデル

$p(y|x)$ を直接モデル化

## 生成モデル

$p(x)$  or  $p(x, y)$ のモデル化

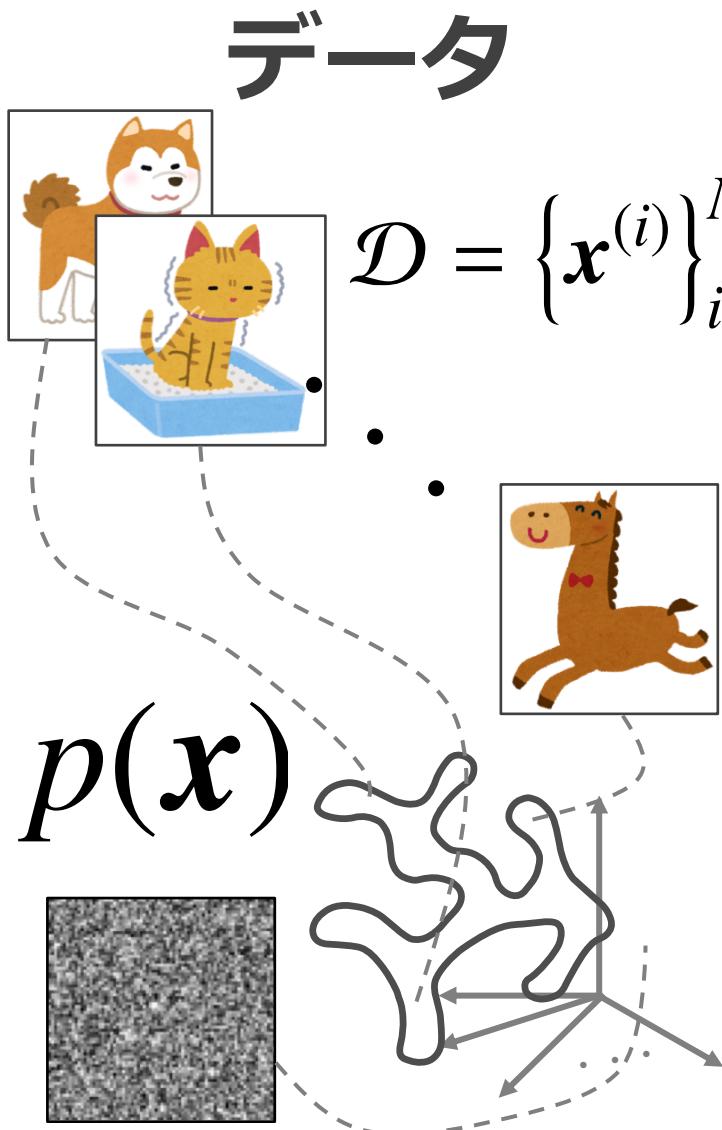
## データの例

$x$ : 画像, 音声波形, 単語系列, ...

$y$ : カテゴリ, 属性, ...

⇒ 今回は主に  $x$  は画像,  $y$  は無いものとして説明する

# 生成モデルとは？



## データ

$N$ 個訓練データ $\mathcal{D}$ が与えられたときに、その背後にある**データを生成した確率分布** $p(\mathcal{D})$ を推定する。

特に、データはi.i.d.で生成されると仮定して、

$$p(\mathcal{D}) = \prod_{i=1}^N p(x^{(i)})$$

としたときの各データの生成分布 $p(x^{(i)})$ を推定する。

# 生成モデルでできること

- 訓練データに近いデータの生成
- 訓練データ間の補完
- データに含まれるノイズ・欠損の除去
- 半教師あり学習への応用

※ 識別モデルでは識別問題そのものしか解けない！

# 深層生成モデルの教師なし学習

## RBM

[Hinton, Neural Comput. 2002]

ボルツマンマシン  
を可視変数と隠れ  
変数で2部グラフ化  
  
サンプリングを途中  
で停止するCD学  
習を提案

## VAE

[Kingma+, ICLR2014]

対数尤度の変分下  
限に関する新しい  
表式を提案  
  
Reparametrization  
trickによって決定  
論的な誤差逆伝搬  
のみで確率モデル  
を学習

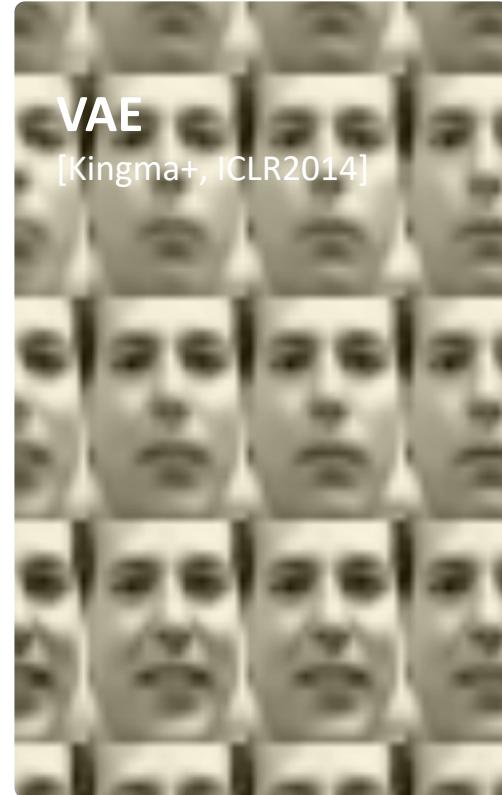
## GAN

[Goodfellow+, NIPS2014]

データについて  
implicitな確率分布を  
与えるモデル  
  
生成モデルと識別  
モデルが互いに相  
手の2者間の敵対的  
学習の枠組み

複雑な確率分布をいかに**単純なサンプリングと勾配法**で学  
習するか？という大きい流れ

# 深層生成モデルの教師なし学習



※: 上画像はAISTATS2009

※: 上画像はDCGAN

複雑な確率分布をいかに**単純なサンプリングと勾配法**で学習するか？という大きい流れ

# ボルツマンマシン

データの分布をあるエネルギー $\Phi(\mathbf{x}, \mathbf{W}, b)$ に従うボルツマン分布でモデル化

データの各次元の間に相互作用 $W_{ij}$ を仮定する

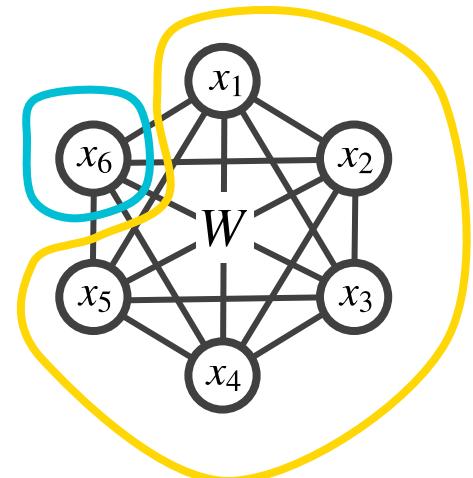
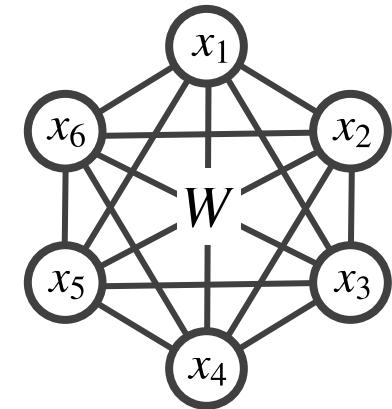
確率分布の学習のため、勾配計算の際に  
**複雑な積分(モデル平均)**が登場する

$$\frac{1}{N} \nabla_{W_{ij}} \mathcal{L}(\boldsymbol{\theta}) = \langle x_i x_j \rangle_{\text{data}} - \langle x_i x_j \rangle_{\text{model}}$$

$$\frac{1}{N} \nabla_{b_i} \mathcal{L}(\boldsymbol{\theta}) = \langle x_i \rangle_{\text{data}} - \langle x_i \rangle_{\text{model}}$$

$i$ 番目の素子をそれ以外からの条件付きで与える  
**Gibbs sampling**で効率的にモデル平均を計算

$$p(x_i | \underline{x}_{\setminus i}, \boldsymbol{\theta}) = \frac{p(\mathbf{x} | \boldsymbol{\theta})}{\sum_{x_i=0,1} p(\mathbf{x} | \boldsymbol{\theta})}$$



# Restricted Boltzmann Machine

各次元間の関係をより柔軟にモデル化するために  
潜在変数  $h$  を導入

可視変数間, 潜在変数間には  
依存関係がないものと仮定する

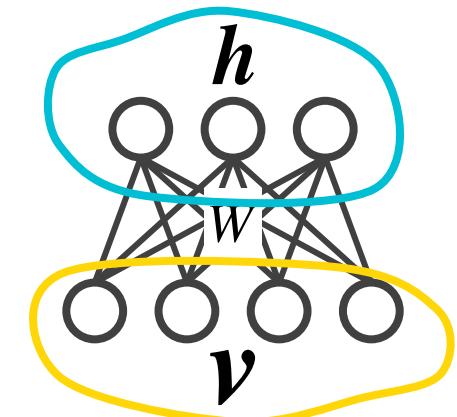
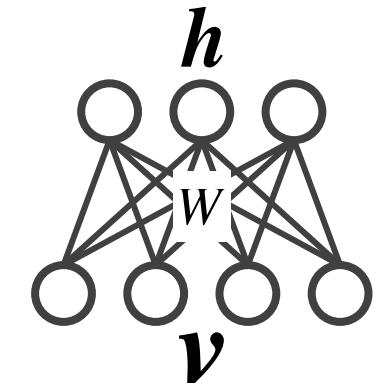
勾配にはボルツマンマシン同様  
**複雑な積分(モデル平均)**が登場する

$$\frac{1}{N} \frac{\partial \log \mathcal{L}(\theta)}{\partial W_{ij}} = \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}}$$

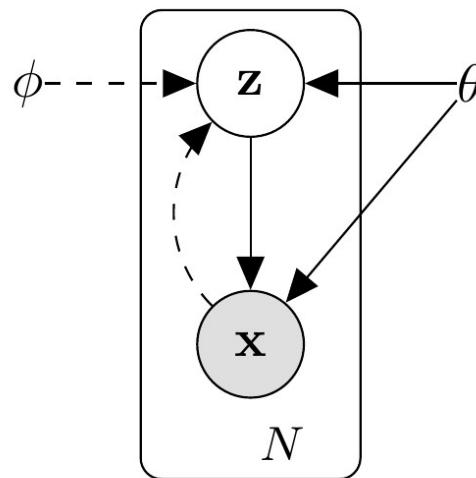
$$\frac{1}{N} \frac{\partial \log \mathcal{L}(\theta)}{\partial a_i} = \langle v_i \rangle_{\text{data}} - \langle v_i \rangle_{\text{model}}$$

$$\frac{1}{N} \frac{\partial \log \mathcal{L}(\theta)}{\partial b_j} = \langle h_j \rangle_{\text{data}} - \langle h_j \rangle_{\text{model}}$$

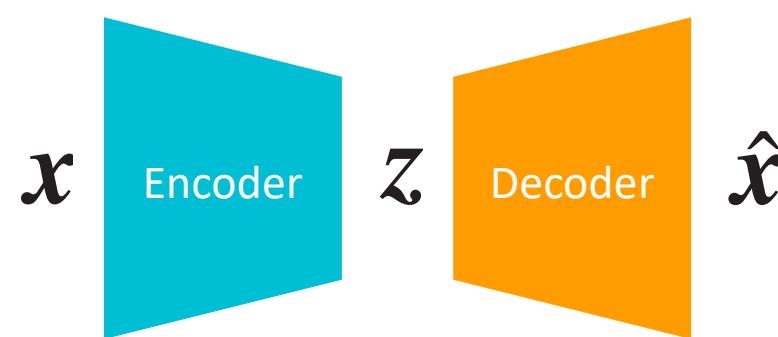
$v$  と  $h$  を交互にサンプリングする contrastive divergence  
(CD) 学習で効率的にモデル平均を計算



## グラフィカルモデル



## ニューラルネットワーク



データ分布を潜在変数 $z$ からの条件付き分布で与える

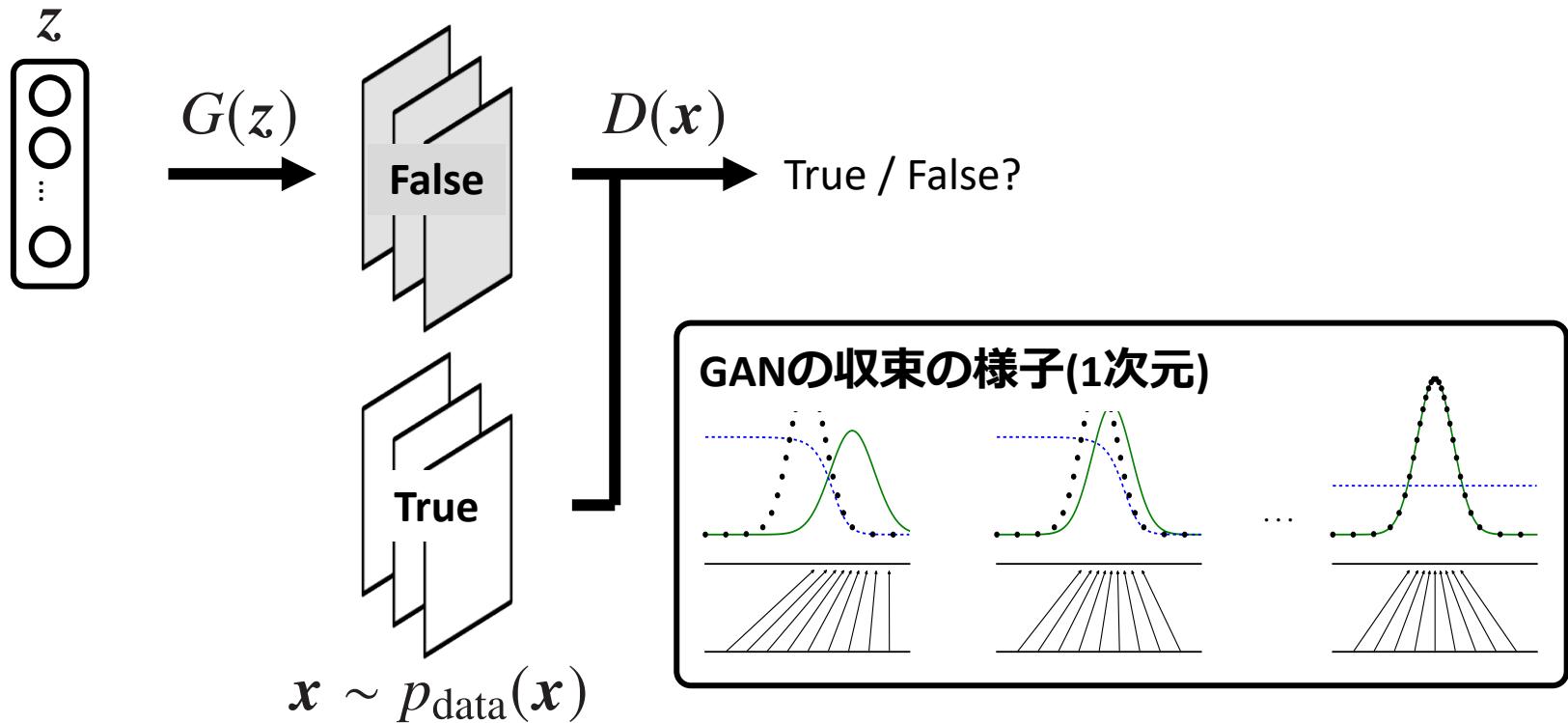
潜在変数からデータ，データから潜在変数の条件付き分布をそれぞれ別々のニューラルネットワークでモデル化

対数尤度の変分下限(variational lower bound)を最大化

確率的な素子 $z$ を経由してエンコーダーに勾配を流すために  
reparameterization trickを使用したSGVBで最適化

# Generative Adversarial Nets (GAN)

26



可視変数 $x$ にimplicit distributionを仮定する生成モデル

ノイズから画像を生成するGeneratorと画像の真偽を見分けるDiscriminatorが互いに互いを騙し合うように分布を学習する

敵対的学習は真のデータ分布 $P$ と生成モデルの分布 $Q$ の間の $f$ -divergenceの変分下限を最小化する

# *Restricted Boltzmann Machine*

解くべき問題

教師なし学習

モデリング

ボルツマンマシン

求解アルゴリズム

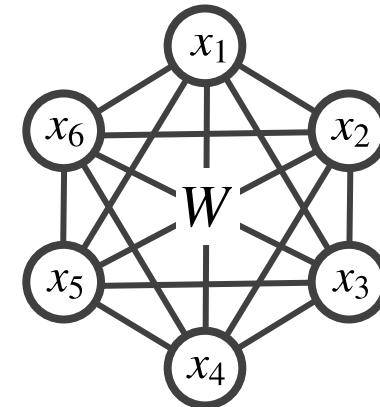
勾配法/ギブスサンプリング

# ボルツマンマシン

## モデル

$$p(x|\theta) = \frac{1}{Z(\theta)} \exp(-\Phi(x, \theta))$$

$$\Phi(x, \theta) = - \sum_{i=1}^M b_i x_i - \sum_{i,j} W_{ij} x_i x_j$$



データの分布をエネルギー $\Phi$ のボルツマン分布でモデル化

データの各次元の間に相互作用 $W_{ij}$ を仮定する

$x$ をバイナリ変数とした場合の分配関数は以下で与えられる:

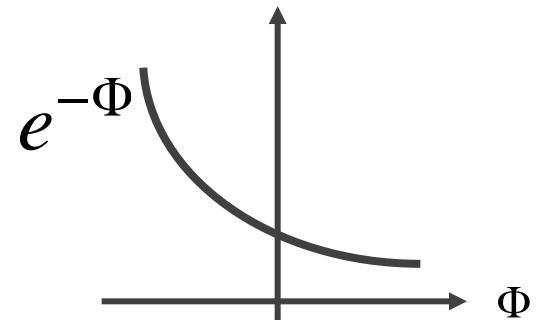
$$\begin{aligned} Z(\theta) &= \sum_x \exp(-\Phi(x, \theta)) \\ &= \sum_{x_1=0,1} \sum_{x_2=0,1} \cdots \sum_{x_M=0,1} \exp(-\Phi(x, \theta)) \end{aligned}$$

# ボルツマンマシン

## モデル

$$p(x|\theta) = \frac{1}{Z(\theta)} \exp(-\Phi(x, \theta))$$

$$\Phi(x, \theta) = - \sum_{i=1}^M b_i x_i - \sum_{i,j} W_{ij} x_i x_j$$



## モデルの解釈

エネルギー $\Phi$ が小さいほど確率が高い

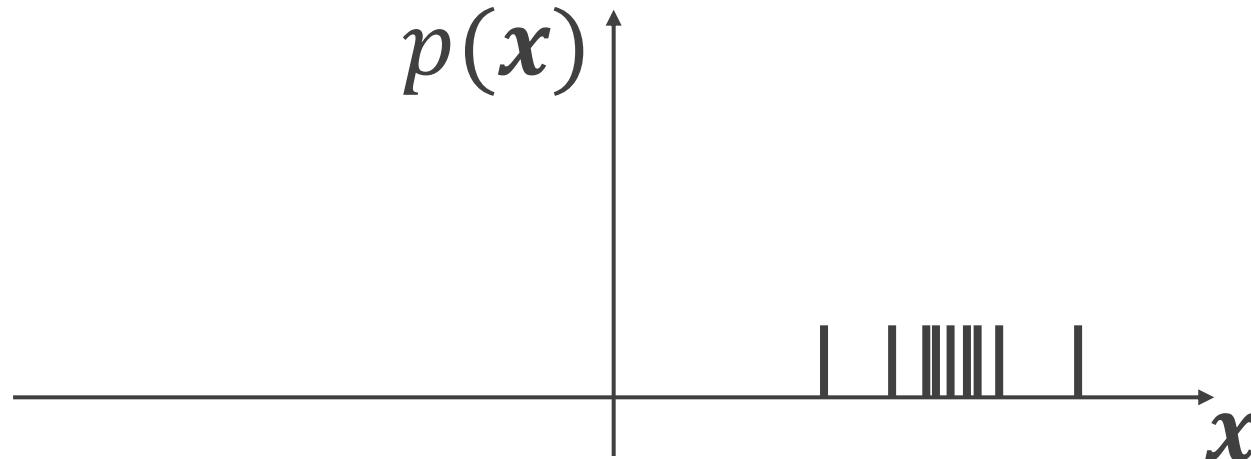
データ $x$ はバイアス $b$ と符号が揃いやすい ( $\Leftarrow$ 磁場)

$W_{ij}$ が正なら $x_i$ と $x_j$ の符号が揃いやすい。負なら反転しやすい (多数決)

⇒ データ $x^{(1)}, x^{(2)}, \dots, x^{(N)}$ からパラメーター $\theta = (W, b)$ を決定する

データ  $x$  に対するパラメーター  $\theta = (\mu, \sigma)$  の対数尤度

$$\log p(x|\theta)$$

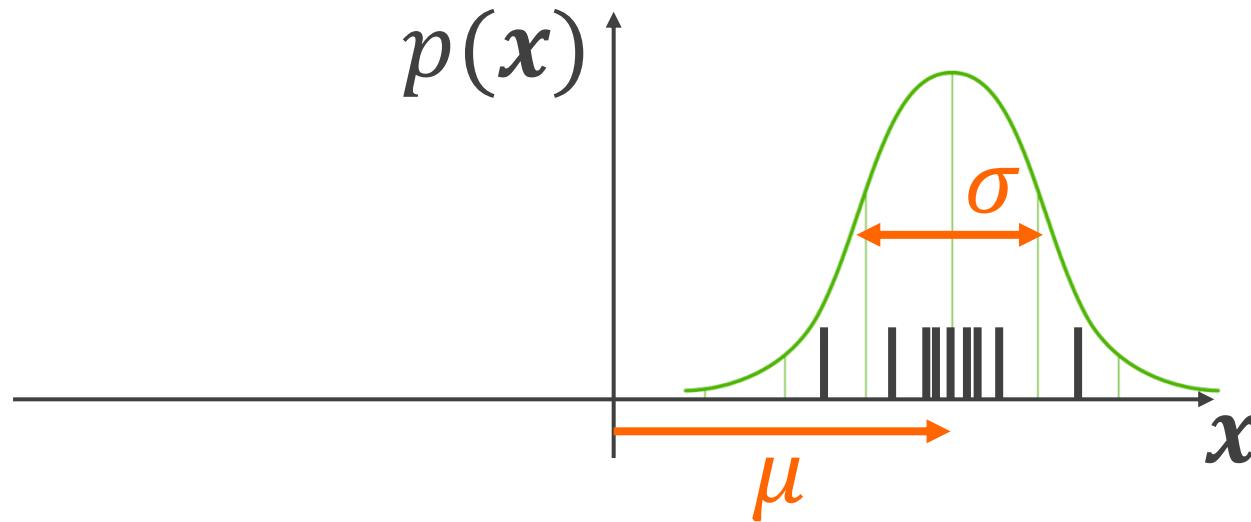


(対数)尤度は与えられたデータに対するモデルのもっともらしさ

⇒ (対数)尤度を最大化を最大化する学習を考える

データ  $x$  に対するパラメーター  $\theta = (\mu, \sigma)$  の対数尤度

$$\log p(x|\theta)$$



(対数)尤度は与えられたデータに対するモデルのもっともらしさ

⇒ (対数)尤度を最大化を最大化する学習を考える

# 対数尤度の勾配

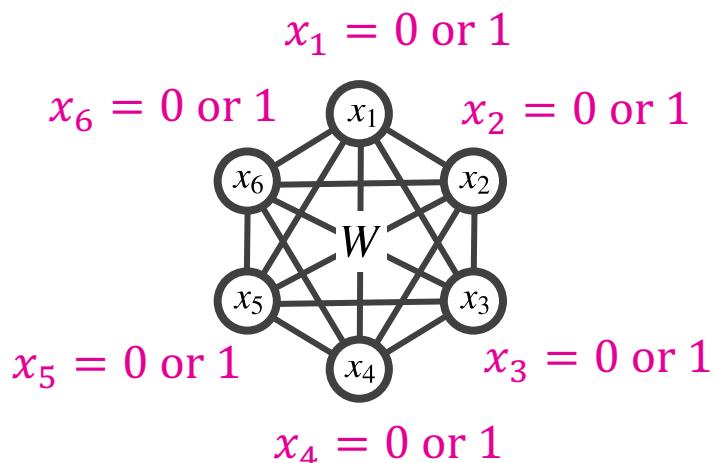
## 対数尤度

$$\log \mathcal{L}(\theta) = \sum_{n=1}^N \log p(x^{(n)}|\theta) = \sum_{n=1}^N \{-\Phi(x^{(n)}, \theta) - \log Z(\theta)\}$$

## 対数尤度のパラメーターに関する勾配

$$\frac{1}{N} \frac{\partial \log \mathcal{L}(\theta)}{\partial W_{ij}} = \langle x_i x_j \rangle_{\text{data}} - \underline{\langle x_i x_j \rangle_{\text{model}}}$$

$$\frac{1}{N} \frac{\partial \log \mathcal{L}(\theta)}{\partial b_i} = \langle x_i \rangle_{\text{data}} - \underline{\langle x_i \rangle_{\text{model}}}$$



データに関する期待値 $\langle \cdot \rangle_{\text{data}}$ は計算が容易

⇒ モデルに関する期待値 $\langle \cdot \rangle_{\text{model}}$ は $2^M$ 通りの和の計算が必要！

# モデル平均の効率的計算

対数尤度の勾配を計算するためには、なんらかの $f(x)$ に関するモデル平均 $\langle f(x) \rangle_{\text{model}} = \sum_{x_1=0,1} \sum_{x_2=0,1} \cdots \sum_{x_M=0,1} f(x)p(x|\theta)$ の計算が必要

## Gibbs sampling

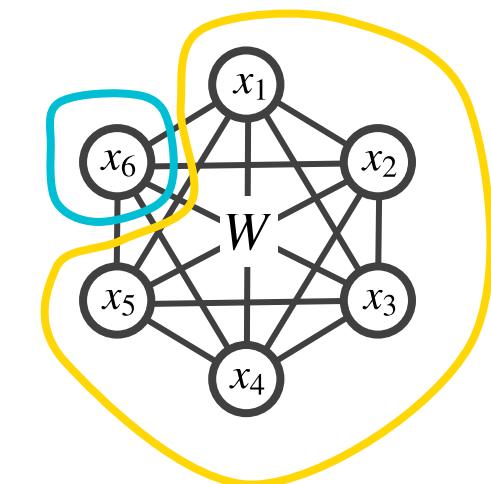
$i$ 番目の素子をそれ以外からの条件付きで与える:

$$p(\underline{x_i} | \underline{x_{\setminus i}}, \theta) = \frac{p(\underline{x} | \theta)}{\sum_{x_i=0,1} p(\underline{x} | \theta)}$$

この時、条件付き確率は以下のシグモイド関数  
 $\sigma(z) = 1/e^{-z}$ で計算できる:

$$p(x_i = 1 | \underline{x_{\setminus i}}, \theta) = \sigma(b_i + \sum_{j=1}^M W_{ij} x_j)$$

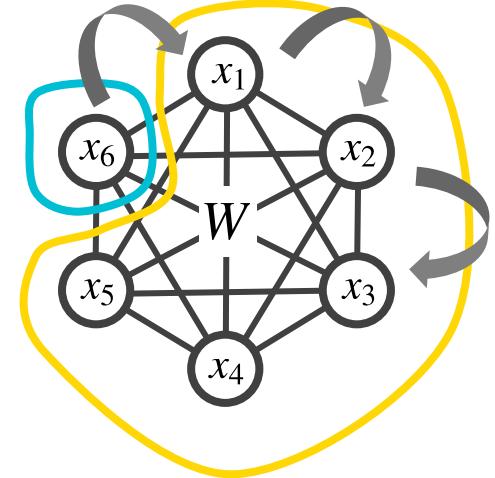
⇒ 活性化関数がシグモイドの再帰NNで確率分布が表現可能！



# Gibbs samplingの手続き

$x(0) \rightarrow x(1) \rightarrow x(2) \rightarrow \dots \rightarrow x(T)$

$x_1$ を変更     $x_2$ を変更     $x_3$ を変更



$p(x|\theta)$ からのサンプルとみなせる

適当な初期値  $x(0)$  から 1 ユニットずつの更新を  $T$  回繰り返す。

十分時間を経た後の  $x(t)$  の系列は元の確率分布  $p(x|\theta)$  からのサンプルとみなせる

$x(t)$  の系列を用いてモデル平均を計算する

# ボルツマンマシンの学習まとめ

データ $\mathcal{D} = \{x^{(n)}\}_{n=1}^N$ を用意する

for epoch in range(nb\_epoch):

Gibbs samplingを用いて対数尤度の勾配を計算:

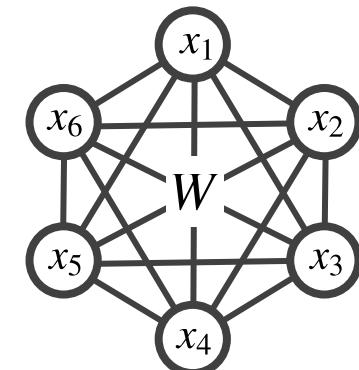
$$\frac{1}{N} \nabla_{W_{ij}} \mathcal{L}(\boldsymbol{\theta}) = \langle x_i x_j \rangle_{\text{data}} - \langle x_i x_j \rangle_{\text{model}}$$

$$\frac{1}{N} \nabla_{b_i} \mathcal{L}(\boldsymbol{\theta}) = \langle x_i \rangle_{\text{data}} - \langle x_i \rangle_{\text{model}}$$

勾配を用いてパラメーターを更新:

$$W_{ij} \leftarrow W_{ij} + \eta \nabla_{W_{ij}} \mathcal{L}(\boldsymbol{\theta})$$

$$b_i \leftarrow b_i + \eta \nabla_{b_i} \mathcal{L}(\boldsymbol{\theta})$$



解くべき問題

教師なし学習



モデリング

RBM



求解アルゴリズム

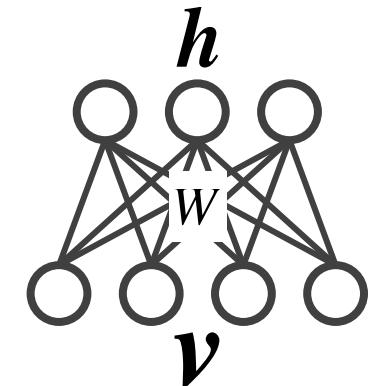
CD学習

# Restricted Boltzmann Machine [Smolensky, Paul 1986; Hinton 2002]

## モデル

$$p(v, h|\theta) = \frac{1}{Z(\theta)} \exp(-\Phi(v, h, \theta))$$

$$\Phi(v, h, \theta) = - \sum_{i=1} a_i v_i - \sum_{j=1} b_j h_j - \sum_{i,j} W_{ij} v_i h_j$$



各次元間の関係をより柔軟にモデル化するために潜在変数  $h$  を導入  
可視変数間, 潜在変数間には依存関係がないものと仮定する

# 対数尤度の勾配

## 対数尤度

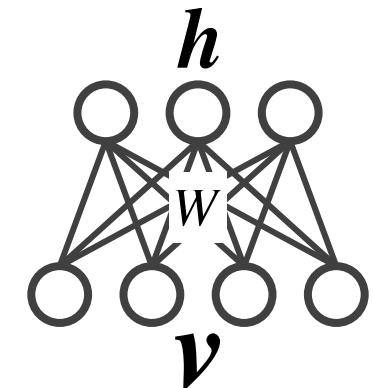
$$\log \mathcal{L}(\theta) = \sum_{n=1}^N \log p(v^{(n)}, h | \theta) = \sum_{n=1}^N \left\{ -\Phi(v^{(n)}, h, \theta) - \log Z(\theta) \right\}$$

## 対数尤度のパラメーターに関する勾配

$$\frac{1}{N} \frac{\partial \log \mathcal{L}(\theta)}{\partial W_{ij}} = \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}}$$

$$\frac{1}{N} \frac{\partial \log \mathcal{L}(\theta)}{\partial a_i} = \langle v_i \rangle_{\text{data}} - \langle v_i \rangle_{\text{model}}$$

$$\frac{1}{N} \frac{\partial \log \mathcal{L}(\theta)}{\partial b_j} = \langle h_j \rangle_{\text{data}} - \langle h_j \rangle_{\text{model}}$$



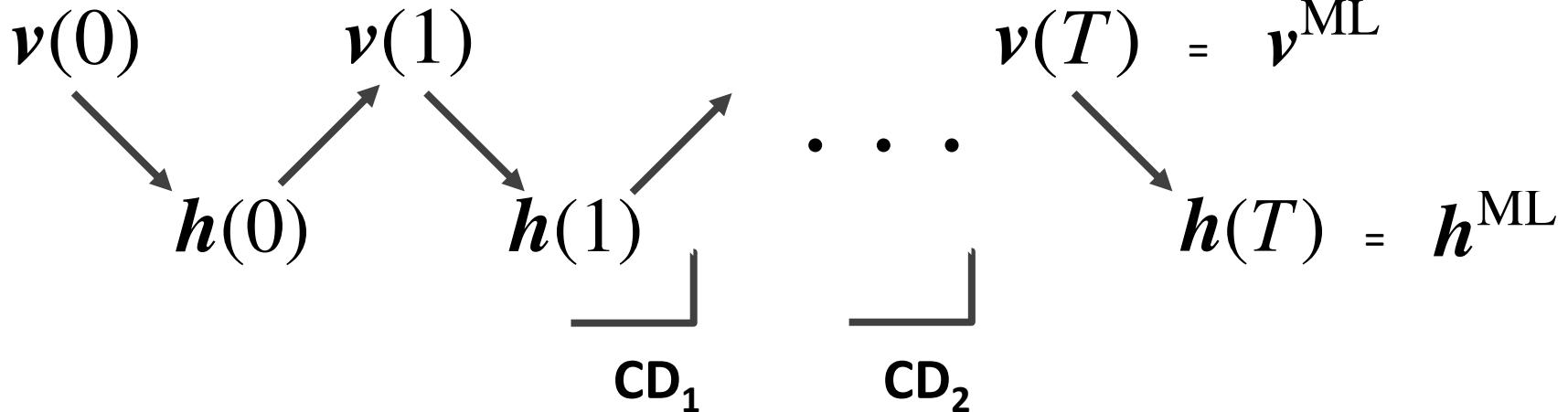
⇒ ボルツマンマシン同様モデル平均を効率的に計算する必要

## 可視変数・潜在変数の条件付き確率

$$p(v_i = 1 | h, \theta) = \sigma(a_i + \sum_j W_{ij} h_j)$$

$$p(h_j = 1 | v, \theta) = \sigma(b_j + \sum_i W_{ij} v_i)$$

### CD学習



可視変数と隠れ変数を交互にサンプリングし, サンプリング回数を  $t$  回  
(典型的には1回)で止めて効率的にモデル平均を計算

# CD学習の妥当性

特定の分布型の下では、CD学習はよい解を与える

## Gaussian-Gaussian RBM

仮定: 入力データの共分散行列  $C = V^T \text{diag}(\lambda_1, \dots, \lambda_N) V$

CD<sub>n</sub>学習の安定平衡点は

$$\bar{W} = U \text{diag} \left( \sqrt{1 - \frac{\sigma^2}{\lambda_1}}, \dots, \sqrt{1 - \frac{\sigma^2}{\lambda_k}}, 0, \dots, 0 \right) V^T$$

任意の直交行列

行列vの各行ベクトル  
= 入力の主成分ベクトル

固有値の大きい主成分のみが抽出される  $\Leftarrow$  **PCAによる次元削減**

最尤解(CD<sub>∞</sub>解)はCD<sub>1</sub>で得ることができる

## Gaussian-Bernoulli RBM

特定の入力の下で、CD<sub>n</sub>学習の**安定平衡点のうちの一つにICA解を含む**

# BM/RBMまとめ

可視変数，潜在変数にボルツマン分布を仮定した生成モデル

学習にモデル平均の計算が必要で，積分が困難

Gibbs sampling (contrastive divergence)を用いて効率的にモデル平均を計算する

BM/RBMはgeneralなモデルである一方，特に多重積分の計算量の問題から実用上はあまり使われない

# *Variational Autoencoder*

# RBMの(個人的)課題

- 確率変数間の関係をexplicitにモデル化している

$$p(\mathbf{v}, \mathbf{h} | \theta) = \frac{1}{Z(\theta)} \exp(-\Phi(\mathbf{v}, \mathbf{h}, \theta))$$

$$\Phi(\mathbf{v}, \mathbf{h}, \theta) = - \sum_{i=1} a_i v_i - \sum_{j=1} b_j h_j - \sum_{i,j} W_{ij} v_i h_j$$

- 確率変数を複数段積むと積分が困難になる

$$p(\mathbf{v} | \mathbf{h}^1, \mathbf{h}^2, \dots, \mathbf{h}^L) = \int p(\mathbf{v} | \mathbf{h}^1) p(\mathbf{h}^1 | \mathbf{h}^2) \cdots p(\mathbf{h}^{L-1} | \mathbf{h}^L) p(\mathbf{h}^L) d\mathbf{h}^1 d\mathbf{h}^2 \cdots d\mathbf{h}^L$$

- 全ての確率変数についてexplicitな分布型を仮定している

$$p(v_i = 1 | \mathbf{h}, \theta) = \sigma(a_i + \sum_j W_{ij} h_j)$$

$$p(h_j = 1 | \mathbf{v}, \theta) = \sigma(b_j + \sum_i W_{ij} v_i)$$

# RBMの(個人的)課題

VAEで(部分的に)解決?

- 確率変数間の関係をexplicitにモデル化している

$$p(\mathbf{v}, \mathbf{h} | \theta) = \frac{1}{Z(\theta)} \exp(-\Phi(\mathbf{v}, \mathbf{h}, \theta))$$

$$\Phi(\mathbf{v}, \mathbf{h}, \theta) = - \sum_{i=1} a_i v_i - \sum_{j=1} b_j h_j - \sum_{i,j} W_{ij} v_i h_j$$

- 確率変数を複数段積むと積分が困難になる

$$p(\mathbf{v} | \mathbf{h}^1, \mathbf{h}^2, \dots, \mathbf{h}^L) = \int p(\mathbf{v} | \mathbf{h}^1) p(\mathbf{h}^1 | \mathbf{h}^2) \cdots p(\mathbf{h}^{L-1} | \mathbf{h}^L) p(\mathbf{h}^L) d\mathbf{h}^1 d\mathbf{h}^2 \cdots d\mathbf{h}^L$$

- 全ての確率変数についてexplicitな分布型を仮定している

$$p(v_i = 1 | \mathbf{h}, \theta) = \sigma(a_i + \sum_j W_{ij} h_j)$$

$$p(h_j = 1 | \mathbf{v}, \theta) = \sigma(b_j + \sum_i W_{ij} v_i)$$

# 機械学習の階層性

解くべき問題

教師なし学習



モデリング

VAE

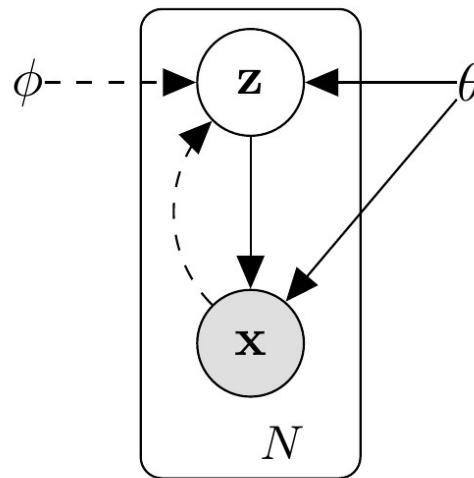


求解アルゴリズム

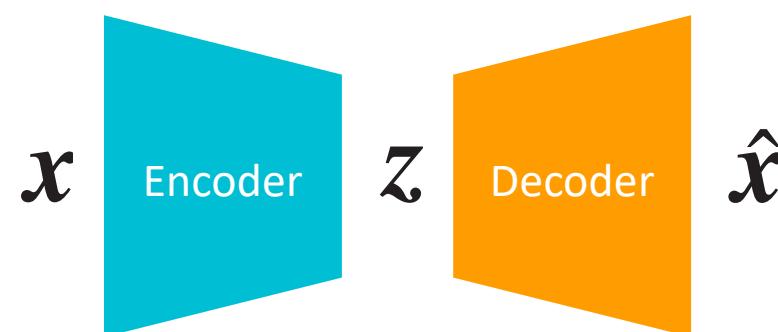
SGVB

[Kingma, Welling, ICLR2014; Rezende+, ICML2014]

## グラフィカルモデル



## ニューラルネットワーク



データ分布を潜在変数 $z$ からの条件付き分布で与える

潜在変数からデータ，データから潜在変数の条件付き分布をそれぞれ別々のニューラルネットワークでモデル化

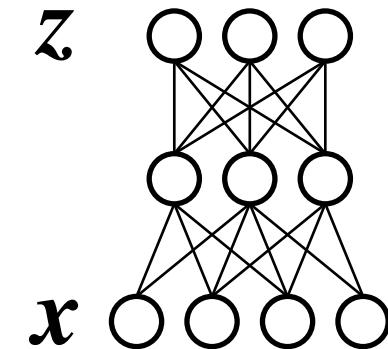
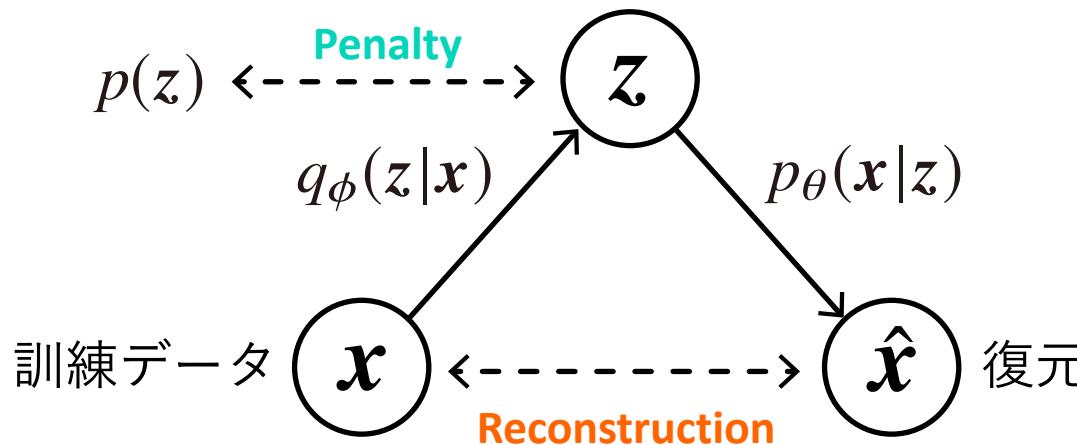
対数尤度の変分下限(variational lower bound)を最大化

# Variational Autoencoder (VAE)

[Kingma, Welling, ICLR2014; Rezende+, ICML2014]

目的関数

$$\mathcal{L}(\theta, \phi; x^{(i)}) = -D_{\text{KL}}(q_{\phi}(z|x^{(i)}) || p_{\theta}(z)) + \mathbb{E}_{q_{\phi}(z|x)} [\log p_{\theta}(x^{(i)}|z)]$$



訓練データ  $\mathcal{D} = \{x^{(i)}\}_{i=1}^N$  を生成する分布  $p(x)$  を推定する  
目的関数として対数尤度  $\log p_{\theta}(x)$  の変分下限を最大化  
 $z$  を特定の分布に押し込む

$$\log p_{\theta}(x^{(i)})$$

$$\log \int p_{\theta}(x^{(i)}, z) dz$$

補助変数(潜在変数)で周辺化する

$$\log \int \frac{q_\phi(z|x^{(i)})}{q_\phi(z|x^{(i)})} \frac{p_\theta(x^{(i)}, z)}{q_\phi(z|x^{(i)})} dz$$

分子と分母に同じ確率分布をかける

$$\geq \int q_{\phi}(z|x^{(i)}) \log \frac{p_{\theta}(x^{(i)}, z)}{q_{\phi}(z|x^{(i)})} dz$$

Jensenの不等式で下から抑える

条件付き確率の公式で分解する

$$\int q_{\phi}(z|x^{(i)}) \log \frac{p_{\theta}(z)p_{\theta}(x^{(i)}|z)}{q_{\phi}(z|x^{(i)})} dz$$

$\log$ の中身の掛け算を $\log$ の足し算に分解する

$$\int q_{\phi}(z|x^{(i)}) \log \frac{p_{\theta}(z)}{q_{\phi}(z|x^{(i)})} dz + \int q_{\phi}(z|x^{(i)}) \log p_{\theta}(x^{(i)}|z) dz$$

logの中身の掛け算をlogの足し算に分解する

$$\int q_{\phi}(z|x^{(i)}) \log \frac{p_{\theta}(z)}{q_{\phi}(z|x^{(i)})} dz + \int q_{\phi}(z|x^{(i)}) \log p_{\theta}(x^{(i)}|z) dz$$

---

Kullback-Leibler divergence

---

Reconstruction error

$$-D_{\text{KL}}(q_{\boldsymbol{\phi}}(z \mid x^{(i)}) \| p_{\boldsymbol{\theta}}(z)) + \mathbb{E}_{q_{\boldsymbol{\phi}}(z|x)}[\log p_{\boldsymbol{\theta}}(x^{(i)} \mid z)]$$

---

Kullback-Leibler divergence

---

Reconstruction error

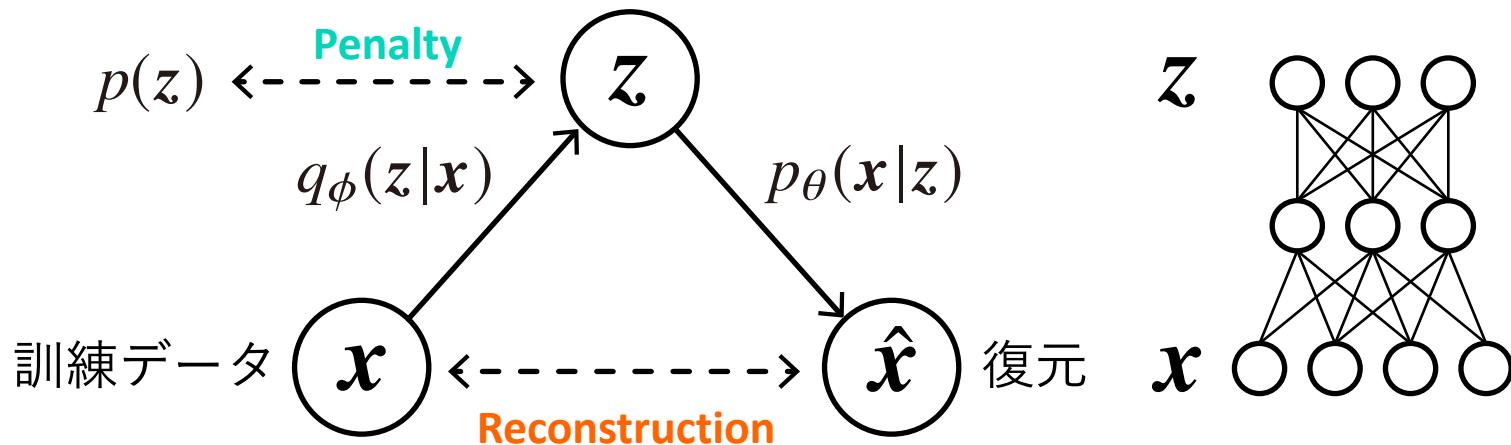
# Variational Autoencoder (VAE)

57

[Kingma, Welling, ICLR2014; Rezende+, ICML2014]

目的関数

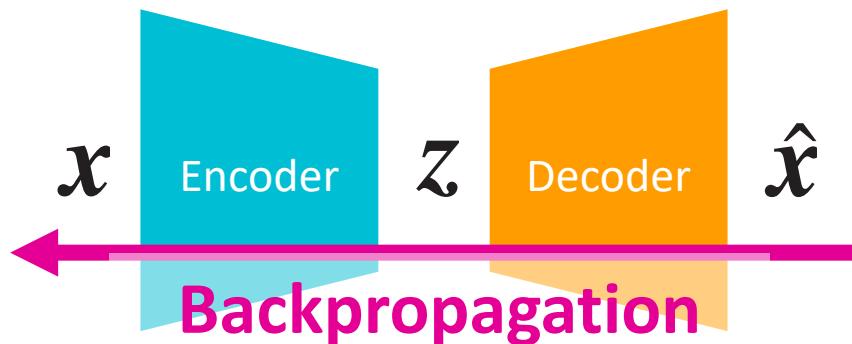
$$\mathcal{L}(\theta, \phi; x^{(i)}) = -D_{\text{KL}}(q_{\phi}(z|x^{(i)}) || p_{\theta}(z)) + \mathbb{E}_{q_{\phi}(z|x)} [\log p_{\theta}(x^{(i)}|z)]$$



訓練データ  $\mathcal{D} = \{x^{(i)}\}_{i=1}^N$  を生成する分布  $p(x)$  を推定する  
目的関数として対数尤度  $\log p_{\theta}(x)$  の変分下限を最大化  
 $z$  を特定の分布に押し込む

目的: 対数尤度の変分下限を最大化したい

手続き: 勾配法でエンコーダー $q_\phi$ とデコーダー $p_\theta$ を最適化



問題点: エンコーダーに勾配を流す際に, 確率的な素子 $z$ を経由する

$$\nabla_{\phi} \mathbb{E}_{q_{\phi}(z|x)}[\log p_{\theta}(x|z)]$$

再構成誤差項の $\phi$ に関する勾配の計算が困難

# Reparameterization trick

VAEでは潜在変数 $z$ を決定論的なパラメーター $\phi$ とパラメターによらない確率変数 $\epsilon$ に分解する

ガウス分布なら $\phi = (\mu, \sigma)$ で

$$z = g_{\phi}(\epsilon) = \mu + \sigma \odot \epsilon$$

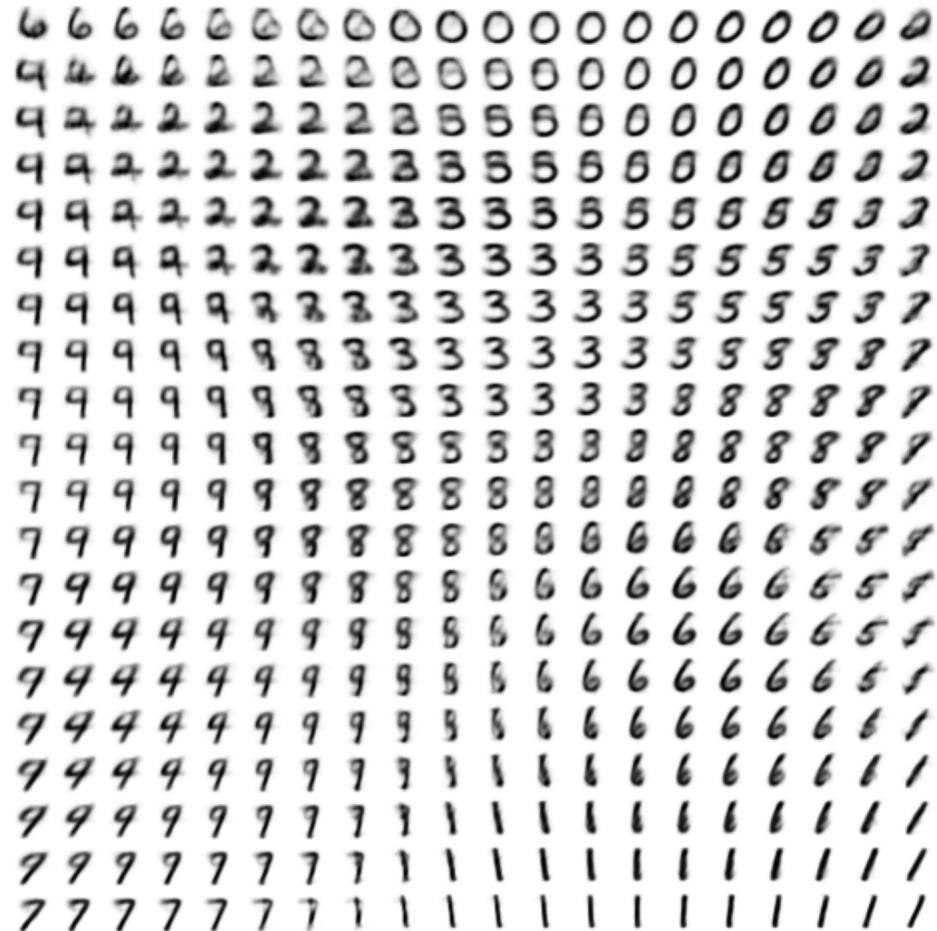
この時、 $\phi$ に関する勾配は以下のMC平均で計算できる:

$$\begin{aligned}\nabla_{\phi} \mathbb{E}_{q_{\phi}(z|x)}[f(z)] &= \nabla_{\phi} \mathbb{E}_{p(\epsilon)}[f(g_{\phi}(\epsilon, x))] \\ &= \mathbb{E}_{p(\epsilon)}[\nabla_{\phi} f(g_{\phi}(\epsilon, x))] \\ &\simeq \frac{1}{L} \sum_{l=1}^L \nabla_{\phi} f(g_{\phi}(\epsilon^{(l)}, x))\end{aligned}$$

# VAEの学習結果



(a) Learned Frey Face manifold



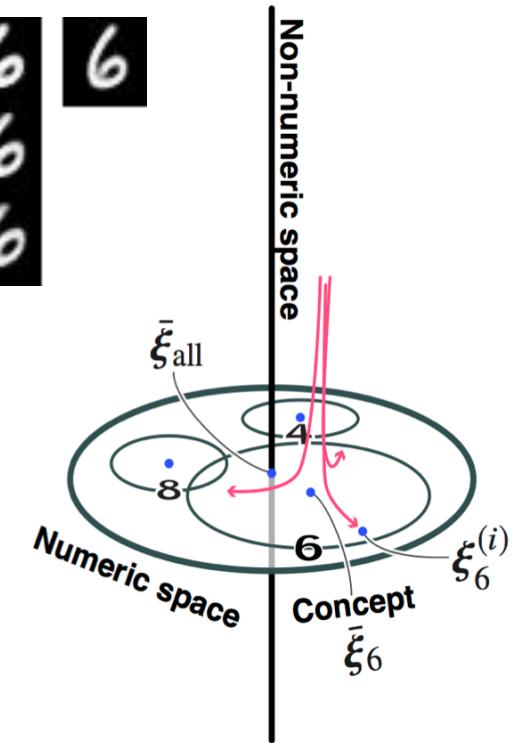
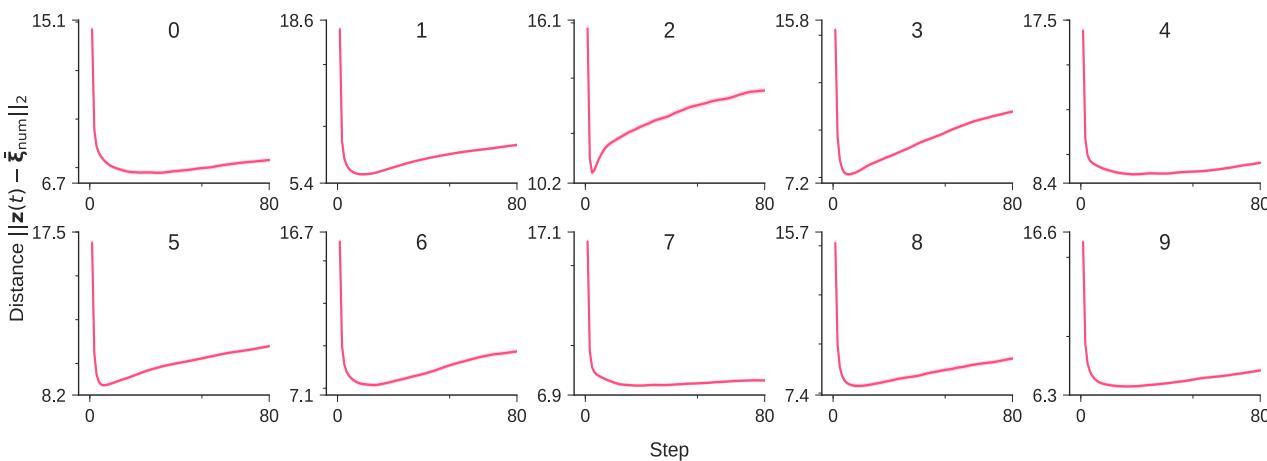
(b) Learned MNIST manifold

[Kingma, Welling, ICLR2014]

NICO2AISCHOOL

# VAEのサンプリングダイナミクスの解析

[Nagano, Karakida, Okada, arXiv:1712.04195]



学習済みのNNでノイズを含む入力から $x \rightarrow z \rightarrow x \rightarrow z \dots$ の順に推論を繰り返す

**潜在空間における発火パターンのダイナミクスはデータに内在するクラスタの中心，“概念”を経由する**

生物の推論の過程と抽象的なレベルで一致

対数尤度の変分下限を最大化するオートエンコーダー型の生成モデル

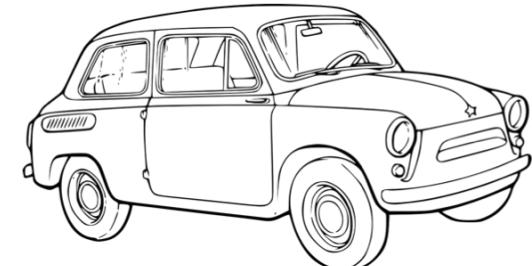
$x$ と $z$ の間に一般的な多層NNを積んでよいので柔軟な分布を表現可能

ただし、特に可視変数 $x$ に関してexplicitな分布型を仮定している

# *Generative Adversarial Network*



[Kingma, Welling, ICLR2014]



[pixabay.com]

RBM/VAEは可視変数についてガウス分布やベルヌーイ分布などのexplicitな分布を仮定  
⇒ 多峰性の分布を表現できない

データの確率分布を  
分布型を陽に仮定しない(**implicit distribution**)で与えたい！



[Kingma, Welling, ICLR2014]

[pixabay.com]

RBM/VAEは可視変数についてガウス分布やベルヌーイ分布などの**explicit**な分布を仮定  
→ 多峰性の分布を表現できない

解くべき問題

教師なし学習



モデリング

GAN



求解アルゴリズム

Adversarial training

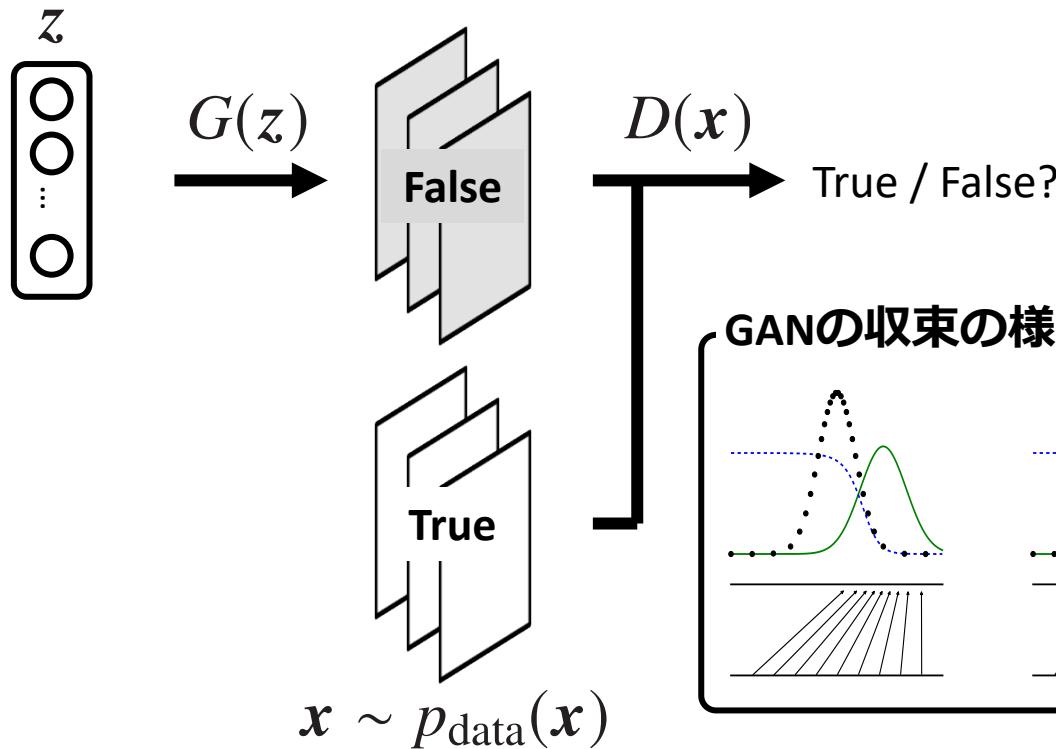
# Generative Adversarial Nets (GAN)

68

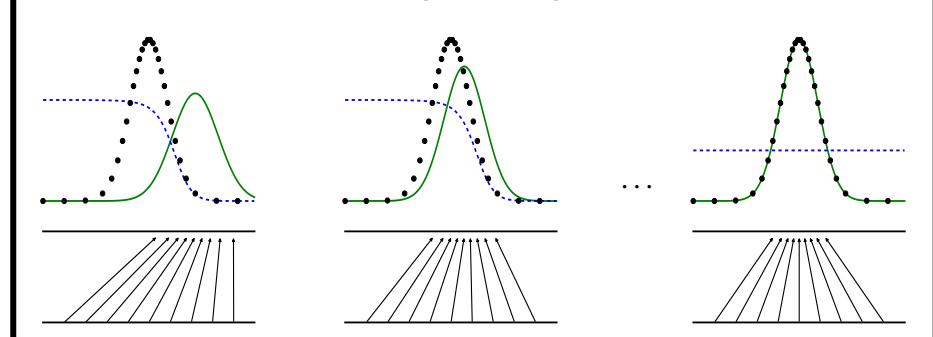
[Goodfellow, ..., Bengio, NIPS2014]

## 目的関数

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



GANの収束の様子(1次元)



[Goodfellow, ..., Bengio, NIPS2014]

1次元ガウシアンでのデモ

[youtu.be/0r3g7-4bMYU](https://youtu.be/0r3g7-4bMYU)

# Generative Adversarial Nets (GAN)

手計算 8

[Goodfellow, ..., Bengio, NIPS2014]

## Optimal discriminator

生成モデル $G$ を固定した下で、最適な識別モデル $D$ は

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

## 証明

任意の $G$ について、最適な $D$ は以下の $V(D, G)$ を最大化することで与えられる

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

任意の $(a, b) \in \mathbb{R}^2 \setminus \{0,0\}$ について関数 $y \rightarrow a \log(y) + b \log(1 - y)$ は最大値 $\frac{a}{a+b}$ を取る。

# Generative Adversarial Nets (GAN)

手計算9

[Goodfellow, ..., Bengio, NIPS2014]

## 定理 1

最適な $D$ の下で生成モデル $G$ のコスト $C(G)$ の最小値は $p_g = p_{\text{data}}$ のみで得られ、その値は $-\log 4$ になる。

## 証明

最適な $D$ を代入すれば、目的関数は

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

と書き直すことができる。

# Generative Adversarial Nets (GAN)

手計算9

[Goodfellow, ..., Bengio, NIPS2014]

## 証明続き

$p_g = p_{\text{data}}$ の時、最適な $D_G^*(x) = \frac{1}{2}$ である。代入すれば

$$C(G) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[-\log 2] + \mathbb{E}_{x \sim p_g(x)}[-\log 2] = -\log 4$$

を得る。これを利用して $C(G)$ を変形すれば

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

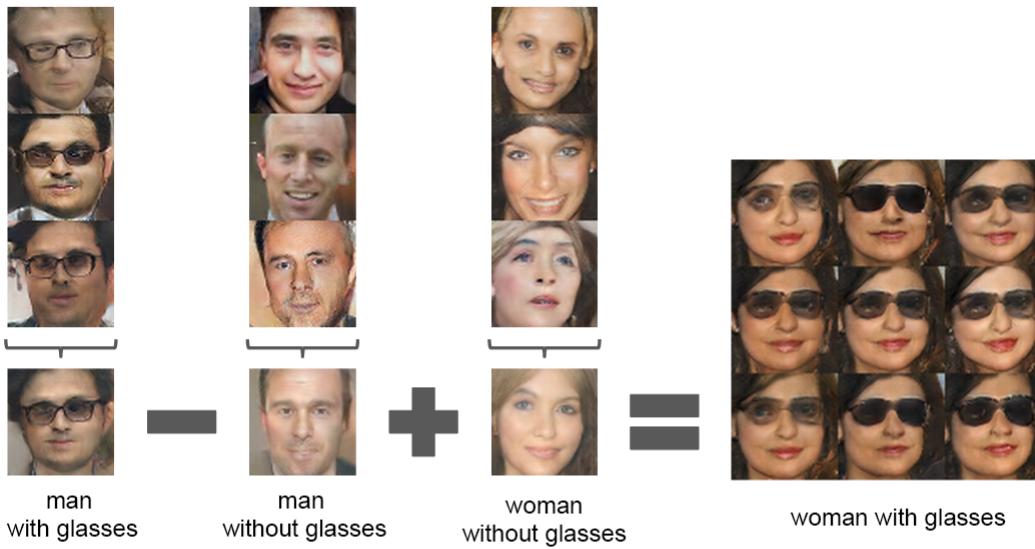
Jensen-Shannonダイバージェンスの非負性より、最小値は $-\log 4$ .

# Generative Adversarial Nets (GAN)

73



z空間上で線形補間



画像同士の加減算

[DCGAN: Radford+, arXiv2015]

NICO2AISCHOOL

# Question

GANは元論文の計算から、**最適な** $D$ の下ではデータ分布 $p_{\text{data}}$ と生成モデル $p_g$ の間のJSダイバージェンスを最小化していることがわかった。

一方で、ゆがんだ多様体上でダイバージェンスは無数に定義可能であり、また最適でない $D$ のときに何が起きるのかよくわからない

**GANにより一般的なダイバージェンスの表式を与えることができるか？**  
**最適化すべき対象からGANのアルゴリズムが演繹的に導出されるか？**

## f-GAN [Nowozin+, NIPS2016]

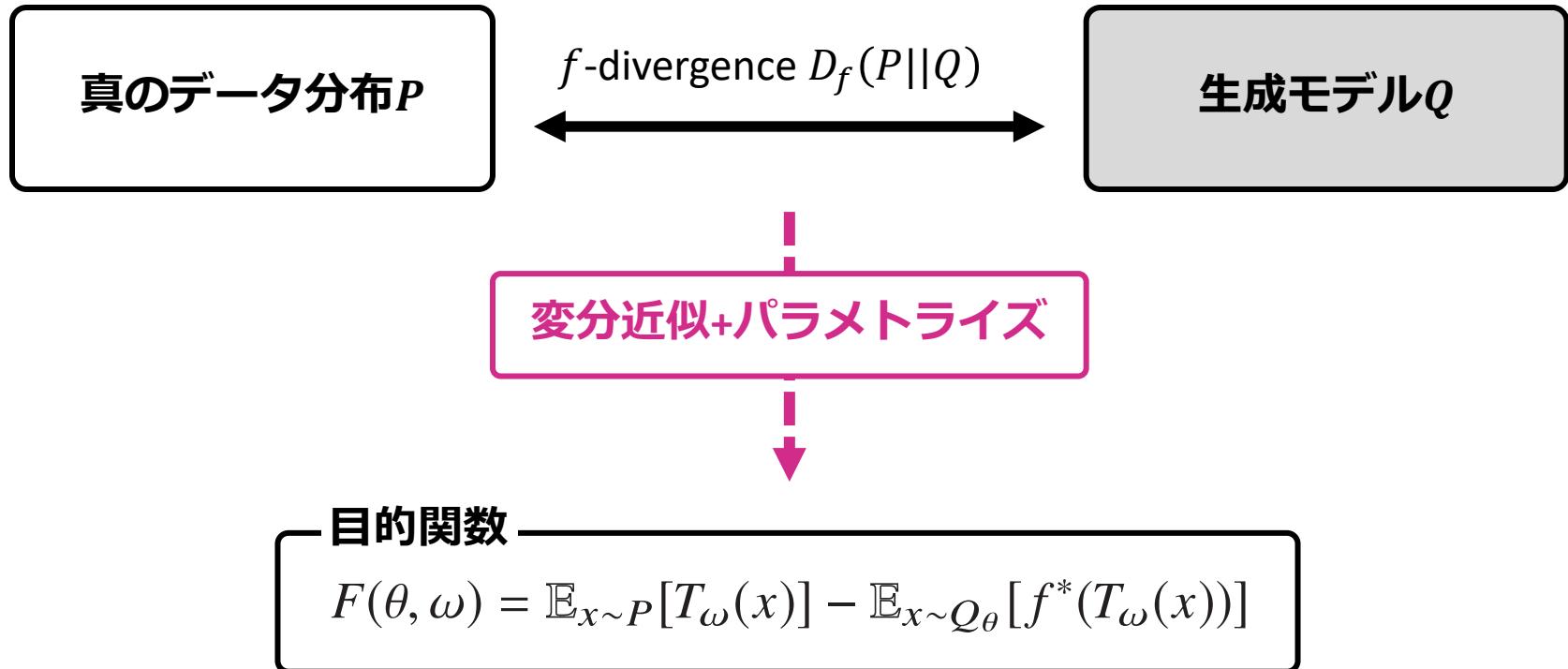
数学的に自然な形でGANの目的関数(を $f$ -divergenceで一般化した形)を導出した。

Goodfellowらの提案するアルゴリズムを単純化し、理論的正当性を与えた。  
自然画像の推定に適切なダイバージェンスに関して実験的な洞察を与えた。

# f-GANの流れ

75

[Nowozin+, NIPS2016]



⇒ 導出される目的関数がGANの一般化に対応

# $f$ -divergence

[Csiszár1963, Morimoto1963, Ali&Silvey1966]

2つの確率分布に $P, Q$ に対して $f(1) = 0$ を満たす凸関数 $f$ を定義した時、以下の式で定義される $f$ -divergenceを最小化することを考える。

$$\begin{aligned} D_f(P\|Q) &= \int_{\chi} f\left(\frac{\mathrm{d}P}{\mathrm{d}Q}\right) \mathrm{d}Q \\ &= \int_{\chi} q(x) f\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x \end{aligned}$$

$f$ の関数型によって様々なダイバージェンスに対応する。右表に代表例を示す。

---

Divergence	$f(u)$
KL-divergence	$u \log u$
Hellinger distance	$(\sqrt{u} - 1)^2, 2(1 - \sqrt{u})$
Total variation distance	$\frac{1}{2}  u - 1 $

---

# $f$ -divergenceの変分下限

Nguyenらによる2つの確率分布 $P, Q$ からの $f$ -divergenceの変分推定法がベース  
 $f$ -divergenceの式をFenchel共役の形で表せば,

$$\begin{aligned}
 D_f(P\|Q) &= \int_{\chi} q(x)f\left(\frac{p(x)}{q(x)}\right)dx \\
 &= \int_{\chi} q(x) \sup_{t \in \text{dom}_{f^*}} \left\{ t \frac{p(x)}{q(x)} - f^*(t) \right\} dx \\
 &\geq \sup_{T \in \mathcal{T}} \left( \int_{\chi} p(x)T(x)dx - \int_{\chi} q(x)f^*(T(x))dx \right) \\
 &= \sup_{T \in \mathcal{T}} (\mathbb{E}_{x \sim P}[T(x)] - \mathbb{E}_{x \sim Q}[f^*(T(x))])
 \end{aligned}$$

となり、これは $f$ -divergenceの変分下限になる。なお、 $\mathcal{T}$ は任意のクラスの関数  
 $T: \chi \rightarrow \mathbb{R}$

# Fenchel共役 (Legendre=Fenchel変換)

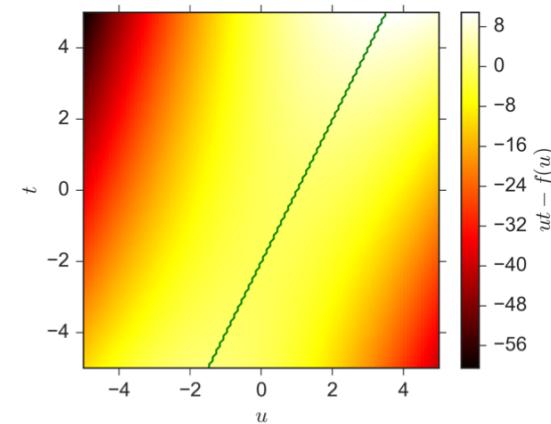
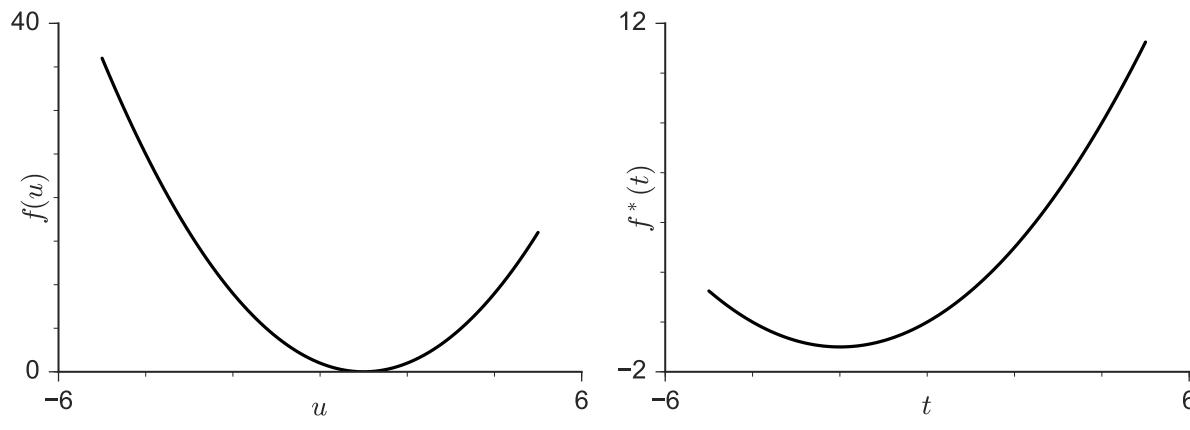
手計算 10

すべての凸で下半連続な関数 $f$ に対する凸共役関数 $f^*$ は

$$f^*(t) = \sup_{u \in \text{dom}_f} \{ut - f(u)\}$$

と書け，これをFenchel共役と呼ぶ。 $f^*$ もまた凸で下半連続であり， $(f, f^*)$ は互いに双対で $f^{**} = f$ 。よって $f(u) = \sup_{t \in \text{dom}_{f^*}} \{tu - f^*(t)\}$ と書ける。

## Legendre=Fenchel変換のイメージ



# $f$ -divergenceの変分下限

Nguyenらによる2つの確率分布 $P, Q$ からの $f$ -divergenceの変分推定法がベース  
 $f$ -divergenceの式をFenchel共役の形で表せば,

$$\begin{aligned}
 D_f(P\|Q) &= \int_{\chi} q(x)f\left(\frac{p(x)}{q(x)}\right)dx \\
 &= \int_{\chi} q(x) \sup_{t \in \text{dom}_{f^*}} \left\{ t \frac{p(x)}{q(x)} - f^*(t) \right\} dx \\
 &\geq \sup_{T \in \mathcal{T}} \left( \int_{\chi} p(x)T(x)dx - \int_{\chi} q(x)f^*(T(x))dx \right) \\
 &= \sup_{T \in \mathcal{T}} (\mathbb{E}_{x \sim P}[T(x)] - \mathbb{E}_{x \sim Q}[f^*(T(x))])
 \end{aligned}$$

となり、これは $f$ -divergenceの変分下限になる。なお、 $\mathcal{T}$ は任意のクラスの関数  
 $T: \chi \rightarrow \mathbb{R}$

# $f$ -divergenceの変分下限

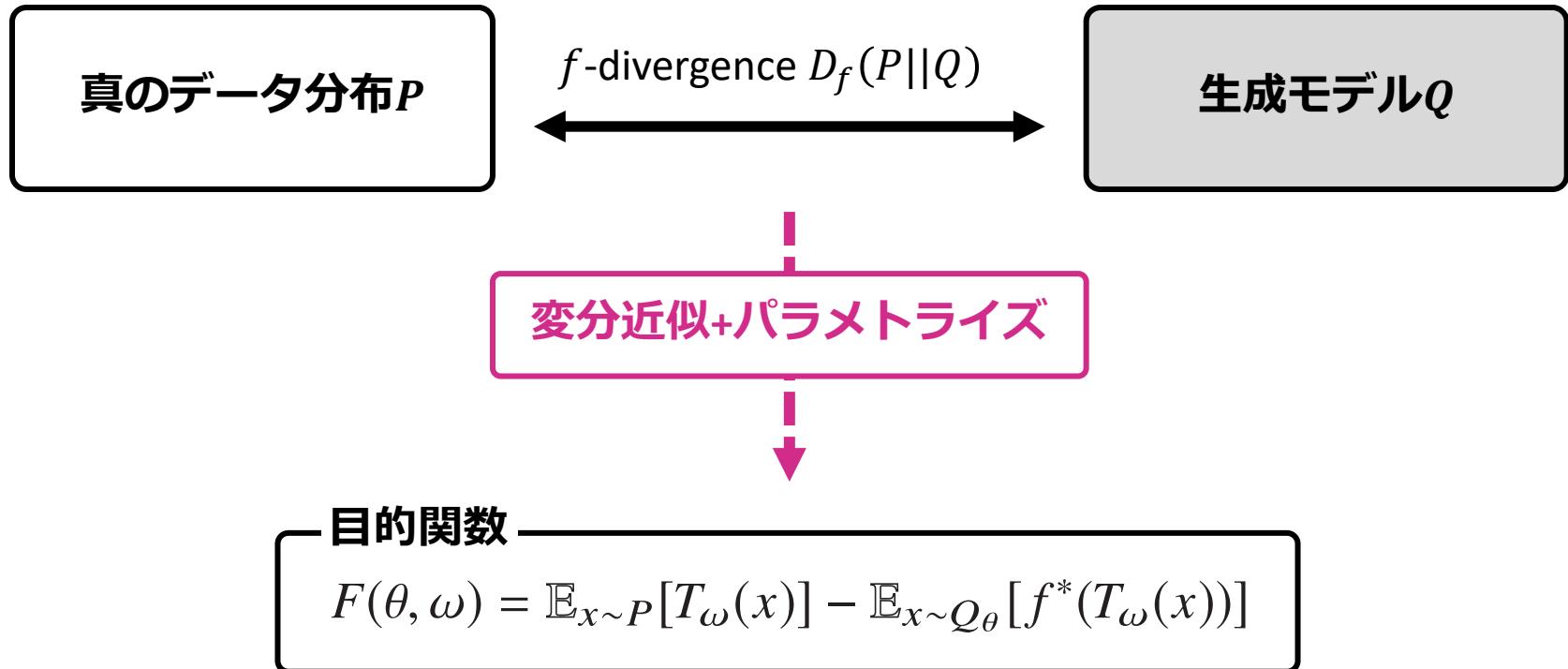
Nguyenらによる2つの確率分布 $P, Q$ からの $f$ -divergenceの変分推定法がベース  
 $f$ -divergenceの式をFenchel共役の形で表せば、

$$\begin{aligned}
 D_f(P\|Q) &= \int_{\chi} q(x)f\left(\frac{p(x)}{q(x)}\right)dx \\
 &= \int_{\chi} q(x) \sup_{t \in \text{dom}_{f^*}} \left\{ t \frac{p(x)}{q(x)} - f^*(t) \right\} dx \\
 &\geq \sup_{T \in \mathcal{T}} \left( \int_{\chi} p(x)T(x)dx - \int_{\chi} q(x)f^*(T(x))dx \right) \\
 &= \sup_{T \in \mathcal{T}} (\mathbb{E}_{x \sim P}[T(x)] - \mathbb{E}_{x \sim Q}[f^*(T(x))])
 \end{aligned}$$

となり、これは $f$ -divergenceの変分下限になる。なお、 $\mathcal{T}$ は任意のクラスの関数  
 $T: \chi \rightarrow \mathbb{R}$

# f-GANの流れ

手計算 10



⇒ 導出される目的関数がGANの一般化に対応

## 目的関数

$$F(\theta, \omega) = \mathbb{E}_{x \sim P}[T_\omega(x)] - \mathbb{E}_{x \sim Q_\theta}[f^*(T_\omega(x))]$$

真の分布  $P$  の下で  $f$ -divergence  $D_f(P||Q)$  を最小にする生成モデル  $Q$  を得るために、生成モデル  $Q$  と変分関数  $T$  をそれぞれ  $\theta$  と  $\omega$  でパラメetrize する。

前述の目的関数を  $\theta$  に関して最小化し、 $\omega$  に関して最大化する。

Name	$D_f(P  Q)$	$f(u)$	$T^*(x)$
Kullback-Leibler	$\int p(x) \log \frac{p(x)}{q(x)} dx$	$u \log u$	$1 + \log \frac{p(x)}{q(x)}$
Reverse KL	$\int p(x) \log \frac{q(x)}{p(x)} dx$	$-u \log u$	$-\frac{q(x)}{p(x)}$
Squared Hellinger	$\int (\sqrt{p(x)} - \sqrt{q(x)})^2 dx$	$(\sqrt{u} - 1)^2$	$\left( \sqrt{\frac{p(x)}{q(x)}} - 1 \right) \cdot \sqrt{\frac{q(x)}{p(x)}}$

# 変分関数の表現

様々な定義域 $\text{dom}_{f^*}$ を持つ $f$ -divergenceに適用するために、変分関数 $T_\omega$ を $T_\omega(x) = g_f(V_\omega(x))$ と変形。

$$F(\theta, \omega) = \mathbb{E}_{x \sim P} [g_f(V_\omega(x))] + \mathbb{E}_{x \sim Q_\theta} [-f^*(g_f(V_\omega(x)))]$$

なお $V_\omega$ は値域に制約を持たない関数で、 $g_f$ は出力活性化関数。ここで元のGANの目的関数

$$F(\theta, \omega) = \mathbb{E}_{x \sim P} [\log D_\omega(x)] + \mathbb{E}_{x \sim Q_\theta} [\log (1 - D_\omega(x))]$$

は識別モデルの最後の非線形関数をシグモイド $D_\omega(x) = 1/(1 + e^{-V_\omega(x)})$ 、対応する出力活性化関数を $g_f(v) = -\log(1 + e^{-v})$ とした特殊ケースに対応する。

可視変数 $x$ にimplicit distributionを仮定する生成モデル  
敵対的学習で分布を学習する  
敵対的学習は真のデータ分布 $P$ と生成モデルの分布 $Q$ の間  
の $f$ -divergenceの変分下限を最小化する

# まとめ

	RBM	VAE	GAN
可視/潜在変数間の関係	Explicit	Implicit	Implicit
潜在→可視変数の条件付き確率	Probabilistic	Probabilistic	Deterministic
可視変数	Explicit	Explicit	Implicit
対数尤度	Exact	Variational lower bound	Intractable

BM/RBM → VAE → GANの流れを個人的解釈に沿って説明した

大枠としては、複雑な確率分布をいかにサンプリングせずに(or 単純な分布からのサンプリングのみで)勾配法で学習するか？という流れ

(個人的な印象として)モデルとしての式の綺麗さ・性質のよさと性能はトレードオフ(?)

⇒ 現実はそんなに甘くない？

## 講義内で飛ばした手計算x10

1. ボルツマンマシン, 対数尤度の勾配の導出
2. ボルツマンマシン, 条件付き確率の導出
3. RBM, 対数尤度の勾配の導出
4. RBM, 条件付き確率の導出
5. VAE, 対数尤度の変分下限の導出
6. VAE, reparameterization trick
7. VAE, 再構成誤差項と罰則項の具体的計算
8. GAN, optimal discriminatorの証明
9. GAN, 生成分布がデータ分布に一致することの証明
10. GAN, f-GANの導出

# ノーテーション

式	意味
$x^{(n)}, v^{(n)}$	$n$ 番目のデータ $x$ (or $v$ )
$x_i$	データ $x$ の $i$ 次元目
$z, h$	潜在変数
$\theta, \phi, W, b$	モデルのパラメーター
$\langle \cdot \rangle_p, \mathbb{E}_p[\cdot]$	ある確率分布 $p$ に関する期待値
$\nabla_{\theta} \mathcal{L}(\theta)$	関数 $\mathcal{L}(\theta)$ の $\theta$ に関する勾配(偏微分)

変数は小文字をスカラー、太字の小文字をベクトル、大文字を行列で表記  
 下付き  $i, j, k$  を行列・ベクトルの次元のイテレーター、上付き  $i, n$  をデータ  
 セットのイテレーターの気持ちで使用 (一部レイヤー数の意味で上付きイテ  
 レーターを使用)