Deep Learning 輪読会 2017

第20章 深層生成モデル

2018.01.29, 2018.02.05

システム情報学専攻 M1 東 耕平
工学部システム創成学科 B4 松嶋達也
東京大学 中村藤紀

Deep Learning 輪読会 2017

第20章(~20.4)　深層生成モデル

2018.01.29

システム情報学専攻

M1 東 耕平

構成

20.1 ボルツマンマシン

- 基本的なマルコフ確率場
  - 無向グラフで表されるマルコフ性のある確率変数の集合
    - マルコフ性：過程の将来状態の条件付き確率分布が、現在状態のみに依存し、過去のいかなる状態にも依存しない
- 相互結合型の確率的ニューラルネットワーク
  - D次元の2値確率変数ベクトル $\boldsymbol{x} \in \{0,1\}^d$ を次のエネルギーベースモデルによる同時確率分布で表現

$$P(\boldsymbol{x}) = \frac{\exp\left(-E(\boldsymbol{x})\right)}{Z}.$$

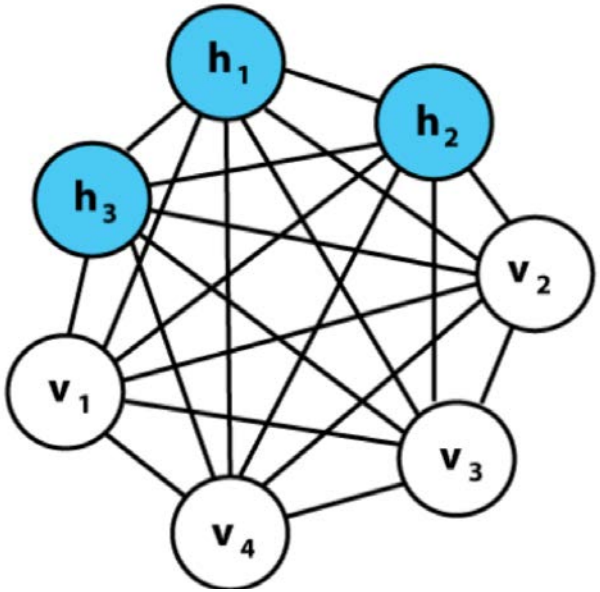
$$Z = \sum_{\boldsymbol{x}} P(\boldsymbol{x}) = 1 \quad \text{(分配関数)}$$

$$E(\boldsymbol{x}) = -\boldsymbol{x}^\top \boldsymbol{U} \boldsymbol{x} - \boldsymbol{b}^\top \boldsymbol{x} \quad \text{(エネルギー関数)}$$

20.1 ボルツマンマシン

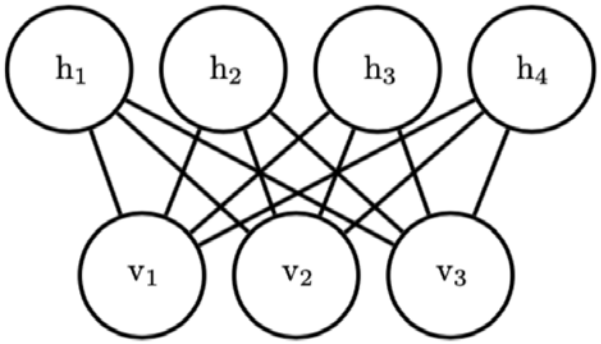
- 隠れ変数(潜在変数)の導入
  - $\boldsymbol{X}$ を可視変数 $\boldsymbol{v}$ と潜在変数 $\boldsymbol{h}$ に分解
    - 可視変数：データに対応した変数
    - 潜在変数：対応するデータが存在しない変数
  - 隠れ変数の追加によりモデルの表現能力が増加

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\boldsymbol{v}^\top \boldsymbol{R}\boldsymbol{v} - \boldsymbol{v}^\top \boldsymbol{W}\boldsymbol{h} - \boldsymbol{h}^\top \boldsymbol{S}\boldsymbol{h} - \boldsymbol{b}^\top \boldsymbol{v} - \boldsymbol{c}^\top \boldsymbol{h}.$$



20.2 制限付きボルツマンマシン

- 可視変数、潜在変数のみに結合があるモデル
  - Restricted Boltzmann Machine (RBM)



$$P(\mathbf{v} = \boldsymbol{v}, \mathbf{h} = \boldsymbol{h}) = \frac{1}{Z} \exp\left(-E(\boldsymbol{v}, \boldsymbol{h})\right).$$

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\boldsymbol{b}^\top \boldsymbol{v} - \boldsymbol{c}^\top \boldsymbol{h} - \boldsymbol{v}^\top \boldsymbol{W} \boldsymbol{h}$$

$$Z = \sum_{\boldsymbol{v}} \sum_{\boldsymbol{h}} \exp\left\{-E(\boldsymbol{v}, \boldsymbol{h})\right\}.$$

Zの計算が困難 [Long and Servedio, 2010]
→ 同時確率P(v) の計算も困難

20.2.1 条件付き分布

- 条件付き分布 P(**h**|**v**) と P(**v**|**h**) は因子分解可能

$$P(\boldsymbol{h} \mid \boldsymbol{v}) = \prod_{j=1}^{n_h} \sigma\left((2\boldsymbol{h} - 1) \odot (\boldsymbol{c} + \boldsymbol{W}^\top \boldsymbol{v})\right)_j .$$

$$P(\boldsymbol{v} \mid \boldsymbol{h}) = \prod_{i=1}^{n_v} \sigma\left((2\boldsymbol{v} - 1) \odot (\boldsymbol{b} + \boldsymbol{W}\boldsymbol{h})\right)_i .$$

- 因子分解可能 → 各変数が統計的に独立 (∵ 条件付き独立の性質)

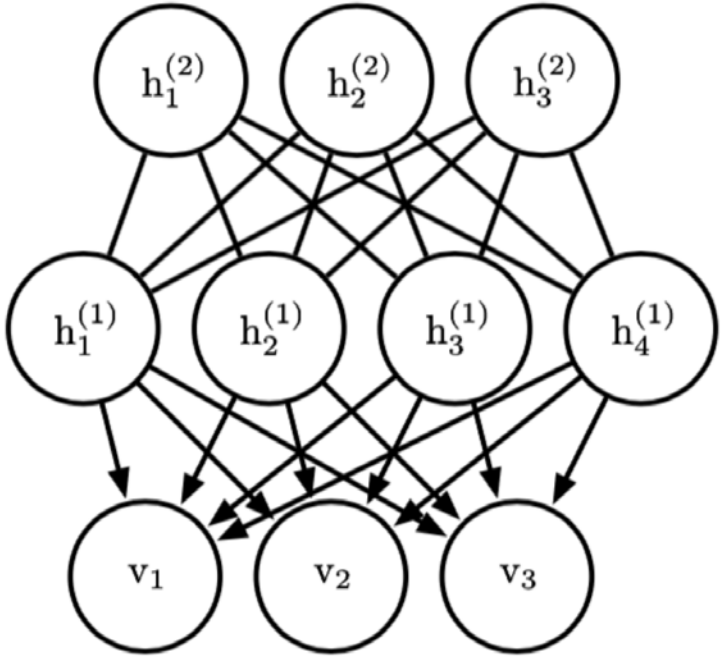
20.2.2 制限付きボルツマンマシンの訓練

- ボルツマンマシンの学習
  - 期待値計算の計算量が $O(2^n)$ なので計算困難
  - 期待値計算はギブスサンプリングによる近似を使用

- 因子分解可能 → 各変数が独立にサンプリング可能
  - 通常は変数間の依存関係によりサンプリングが面倒
  - 変数の値をまとめて更新することが可能(ブロック化ギブスサンプリング)
  - CD法(Contrastive Divergence)に大きく影響

20.3 深層信念ネットワーク

- 深層アーキテクチャの訓練に成功した最初の非畳み込み層のモデル
  - Deep Belief Network (DBN)
  - 潜在変数の層を複数持つ生成モデル
  - 最上位の2層間の結合は無向、それ以外はデータに近い層へ有向



20.3 深層信念ネットワーク

- DBN の訓練：下の層から順番に RBN に見立てて訓練
  - CD法等を用いてRBMを訓練、$\mathbb{E}_{\mathbf{v} \sim p_{\text{data}}} \log p(\boldsymbol{v})$ を最大化
  - 訓練されたRBMのパラメータを使用して二層目のRBMを訓練し、下記を近似的に最大化

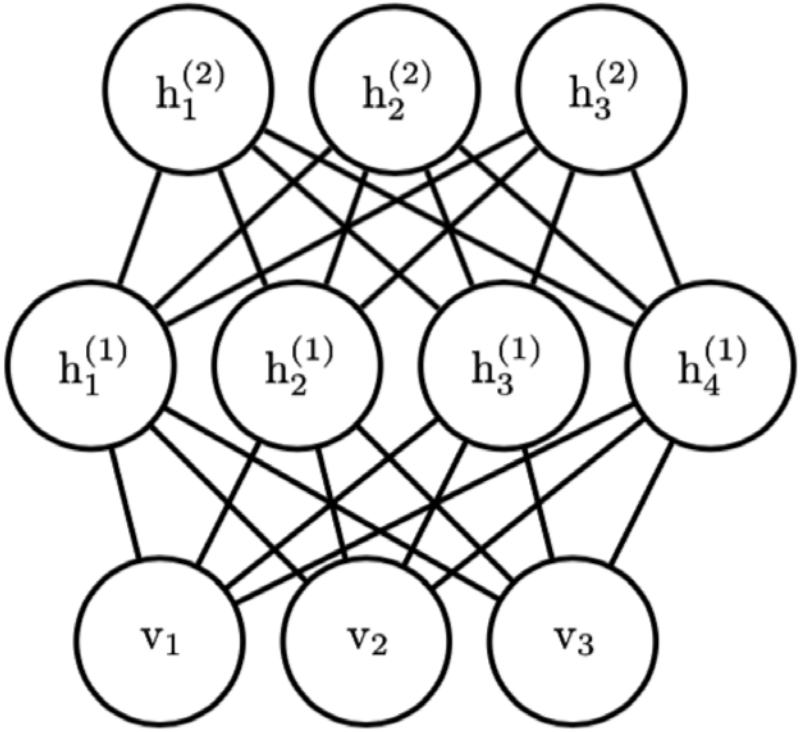
$$\mathbb{E}_{\mathbf{v} \sim p_{\text{data}}} \mathbb{E}_{\mathbf{h}^{(1)} \sim p^{(1)}(\boldsymbol{h}^{(1)} | \boldsymbol{v})} \log p^{(2)}(\boldsymbol{h}^{(1)}),$$

  - この操作の繰り返し (データの対数尤度の変分下限を増加 [Hinton eta al, 2006])

- DBNの訓練で学習されたパラメータでMLPを初期化し分類タスクを実行可
  - ややヒューリスティックな手法

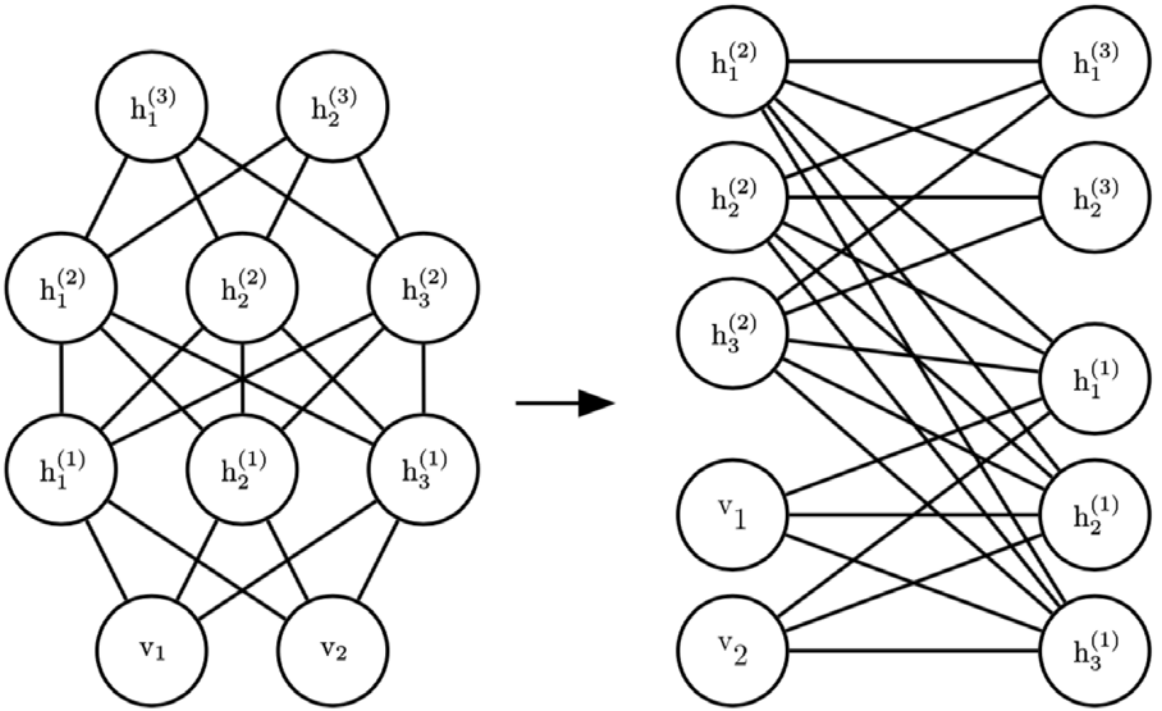
20.4 深層ボルツマンマシン

- 深層生成モデル
  - Deep Boltzmann machine, DBM
  - 潜在変数を複数持つ、完全な無向モデル
  - 各層内の各変数は相互に独立、隣接する層の変数により条件付け



20.4 深層ボルツマンマシン

- 奇数番目の層と偶数番目の層に分割可
  - 2部構造になり、片方の層を条件付けるともう片方の層の分布が因子分解可能
    - 1つのブロックとして同時に独立してサンプリングが可能
    - 効率的なギブスサンプリングが可能



20.4.1 興味深い性質

- DBM は DBN と比較して事後分布 P(**h**|**v**) がより単純
  - DBN によるヒューリスティックな動機付けによる近似推論手続きを用いた分類
    - 求められる下界は明示的には最適化されない
    - 同じ層内の隠れユニット同士の相互作用を考慮することが出来ない
  - DBM では層内の隠れユニットは、他の層が与えられた下で条件付き独立
    - 層内の相互作用が無い
    - 不動方程式を用いて変分下界を最適化、真の最適な平均場の期待値の発見が可能
- DBM のサンプリングは比較的難しい
  - DBN は最上部の二層でのみ MCMC サンプリングを行う必要あり
  - DBM は全ての層に渡って MCMC サンプリングを行う必要あり

20.4.2 DBM の平均場推論

- 全ての隠れ層に関する分布 $P(\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}|\boldsymbol{v})$ は層間の相互作用のため一般的には因子分解不可
  - 隣接する層の下での条件付き確率分布は因子分解可能
- 平均場近似を用いて事後分布を近似
  - 変数間に依存関係があるグラフィカルモデルに対し、それらの変数が互いに独立であると仮定し、各変数の周辺分布を近似的に計算
- 条件付き分布 $P(\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}|\boldsymbol{v})$ を次の分布で近似
  - 違う層の隠れ変数 $\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}$ を互いに独立であると仮定
  - KL距離を最小化するように更新
  - 更新式 (20.33), (20.34) が得られる

$$Q(\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)} \mid \boldsymbol{v}) = \prod_j Q(h_j^{(1)} \mid \boldsymbol{v}) \prod_k Q(h_k^{(2)} \mid \boldsymbol{v})$$

20.4.3 DBM のパラメータ学習

- 計算困難な分配関数、事後分布の課題
  - 20.4.2より変分推論により計算困難な$P(\boldsymbol{h}|\boldsymbol{v})$ を Q($\boldsymbol{h}|\boldsymbol{v}$) で近似
  - 2つの隠れ層を持つ DBM の対数尤度 $\log P(\boldsymbol{v};\boldsymbol{\theta})$ の変分下限 $\mathcal{L}(\boldsymbol{v},Q,\theta)$ は次式

$$\mathcal{L}(Q,\boldsymbol{\theta}) = \sum_i \sum_{j'} v_i W^{(1)}_{i,j'} \hat{h}^{(1)}_{j'} + \sum_{j'} \sum_{k'} \hat{h}^{(1)}_{j'} W^{(2)}_{j',k'} \hat{h}^{(2)}_{k'} - \log Z(\boldsymbol{\theta}) + \mathcal{H}(Q).$$

  - 対数分配関数 $logZ(\theta)$ が含まれる
    - DBM は RBM を含むため、RBM に当てはまる分配関数の計算やサンプリングによる困難が当てはまる → 焼きなまし重点サンプリングや対数分配関数の勾配を近似
  - 通常、確率的最尤法により訓練

20.4.4 層別の事前学習

- 層別の貪欲事前学習の必要性
  - ランダムな初期化で20.3.3の確率的最尤法を使っても良い結果が得られない
  - 大まかな流れは DBN の場合と同じ
- DBM における貪欲事前学習
  - DBN の場合は RBN の重みを直接コピー
  - DBM の場合はボトムアップ入力、トップダウン入力を考慮して値を変更
    [Salakuhutdinov and Hinton, 2009]
    - 最上部と最下部を除く RBM の重みを DBM に挿入する前に半分にする
    - 最上部、最下部は各可視ユニットの2つのコピーと、その2つのコピー間で重みが等しくなるよう制約を課して学習

20.4.5 深層ボルツマンマシンの同時訓練

貪欲事前学習では訓練の後半になるまでハイパーパラメータがどの程度適切かわからない → 同時訓練

- 中心化深層ボルツマンマシン [Montavon and Muiller, 2012]
  - 以下のように再パラメータ化 ($x - u \approx 0$)
    - モデルが表現できる確率分布の集合は変化しないが、尤度に適用される確率的勾配降下法のダイナミクスは変化

$$E(\boldsymbol{x}) = -\boldsymbol{x}^\top \boldsymbol{U}\boldsymbol{x} - \boldsymbol{b}^\top \boldsymbol{x}.$$

$$E'(\boldsymbol{x}; \boldsymbol{U}, \boldsymbol{b}) = -(\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{U}(\boldsymbol{x} - \boldsymbol{\mu}) - (\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{b}.$$

  - コスト関数のヘッセ行列がより良い条件数となる
  - 増強勾配 [Cho et al, 2011] と等価

20.4.5 深層ボルツマンマシンの同時訓練

- 多予測深層ボルツマンマシン [Goodfellow et al, 2013]
  - 平均場方程式 -> 推論問題を近似的に解く回帰型結合ネットワーク族と解釈
    - 回帰型ネットワークが対応する推論問題の答えを得るようにモデルを訓練
  - 近似推論のための計算グラフの誤差逆伝播
    - 実際に使用する方法でモデルが訓練される
      - 特別な修正を加えずに分類器として適切に機能
    - 損失な正確な勾配を計算可能
      - →貪欲事前訓練不要

参考文献

- Deep Learning
  - Ian Goodfellow, Yoshua Bengio, Aaron Courville
  - 日本語版
    https://www.amazon.co.jp/%E6%B7%B1%E5%B1%A4%E5%AD%A6%E7%BF%92-Ian-Goodfellow/dp/4048930621
- 統計的機械学習理論とボルツマン機械学習
  - http://www.aics.riken.jp/labs/cms/workshop/20170322/presentation/yasuda.pdf
- MLP 深層学習
  - https://www.amazon.co.jp/dp/B018K6C99A/ref=dp-kindle-redirect?_encoding=UTF8&btkr=1

Deep Learning 輪読会 2017

第20章　深層生成モデル
(20.5-20.10.6)

2018.02.05

東京大学工学部 松尾研究室
B4 松嶋 達也 (@__tmats__)

構成

- はじめに
- 20.1 ボルツマンマシン
- 20.2 制限付きボルツマンマシン
- 20.3 深層信念ネットワーク
- 20.4 深層ボルツマンマシン
- 20.5 実数値データに対するボルツマンマシン
- 20.6 畳み込みボルツマンマシン
- 20.7 構造化出力や系列出力のためのボルツマンマシン
- 20.8 その他のボルツマンマシン
- 20.9 確率的演算を通る誤差逆伝播
- 20.10 有向生成ネットワーク
- 20.11 自己符号化器からのサンプリング
- 20.12 生成的確率ネットワーク
- 20.13 他の生成スキーム
- 20.14 生成モデルの評価
- 20.15 結論

20.5 実数値データに対するボルツマンマシン

- ボルツマンマシンはもともと2値データで利用するために開発された
  - But 画像・音声では実数値における確率分布を表現する必要

- 実数値が表現できるように拡張したい

20.5.1 ガウス-ベルヌーイ型RBM

- 2値の隠れユニットと実数値の可視ユニットからなるRBM
- 可視ユニットの条件付き分布は平均が隠れユニットの関数となるガウス分布
- 精度行列βを用いた定式化

$$p(\boldsymbol{v} \mid \boldsymbol{h}) = \mathcal{N}(\boldsymbol{v}; \boldsymbol{W}\boldsymbol{h}, \boldsymbol{\beta}^{-1})$$

- エネルギー関数

$$E(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{2}\boldsymbol{v}^{\top}(\boldsymbol{\beta} \odot \boldsymbol{v}) - (\boldsymbol{v} \odot \boldsymbol{\beta})^{\top}\boldsymbol{W}\boldsymbol{h} - \boldsymbol{b}^{\top}\boldsymbol{h}$$

  - βは定数として固定してもいいし，学習することもできる

20.5.2 条件付き共分散の無向モデル

- 画像内の有用な情報のほとんどが存在するのは，ピクセル間の関係であり，ピクセルの絶対値ではないという主張
  - ガウス型RBMは隠れユニットが与えられた下での入力の条件付き平均しかモデリングしないので，条件付き共分散の情報を捉えられない

- 実数値データの共分散をよりよく説明しようとするモデル
  - 平均-共分散型RBM
  - 平均-スチューデントのt分布の積モデル
  - スパイク-スラブ型RBM

20.5.2 条件付き共分散の無向モデル

- **平均-共分散型RBM**(the mean and covariance RBM, mcRBM)
  - 隠れユニットを用いて，全ての観測ユニットの条件付き平均と条件付き共分散を独立に符号化
  - 条件付き平均は単なるガウス型RBMでモデル化
  - 条件付き共分散は**共分散型RBM**(covariance RBM, cRBM)でモデル化
    - 全体のエネルギー関数

$$E_{\mathrm{mc}}\left(\boldsymbol{x}, \boldsymbol{h}^{(m)}, \boldsymbol{h}^{(c)}\right) = E_{\mathrm{m}}\left(\boldsymbol{x}, \boldsymbol{h}^{(m)}\right) + E_{\mathrm{c}}\left(\boldsymbol{x}, \boldsymbol{h}^{(c)}\right)$$

    - ガウス型RBMのエネルギー関数

$$E_{\mathrm{m}}\left(\boldsymbol{x}, \boldsymbol{h}^{(m)}\right) = \frac{1}{2}\boldsymbol{x}^{\top}\boldsymbol{x} - \sum_{j}\boldsymbol{x}^{\top}\boldsymbol{W}_{:,j}h_{j}^{(m)} - \sum_{j}b_{j}^{(m)}h_{j}^{(m)}$$

    - 共分散型RBMのエネルギー関数

$$E_{\mathrm{c}}\left(\boldsymbol{x}, \boldsymbol{h}^{(c)}\right) = \frac{1}{2}\sum_{j}h_{j}^{(c)}\left(\boldsymbol{x}^{\top}\boldsymbol{r}^{(j)}\right)^{2} - \sum_{j}b_{j}^{(c)}h_{j}^{(c)}$$

      - $\boldsymbol{h}^{(m)}$: 2値の平均ユニット, $\boldsymbol{h}^{(c)}$: 2値の共分散ユニット

20.5.2 条件付き共分散の無向モデル

- **平均-共分散型RBM**(the mean and covariance RBM, mcRBM)
  - 同時確率分布

$$p_{\mathrm{mc}}(\boldsymbol{x}, \boldsymbol{h}^{(m)}, \boldsymbol{h}^{(c)}) = \frac{1}{Z} \exp\left\{ -E_{\mathrm{mc}}(\boldsymbol{x}, \boldsymbol{h}^{(m)}, \boldsymbol{h}^{(c)}) \right\}$$

  - $\boldsymbol{h}^{(m)}$と$\boldsymbol{h}^{(c)}$が与えられた下での観測変数における条件付き確率

$$p_{\mathrm{mc}}(\boldsymbol{x} \mid \boldsymbol{h}^{(m)}, h^{(c)}) = \mathcal{N}\left(\boldsymbol{x}; \boldsymbol{C}^{\mathrm{mc}}_{\boldsymbol{x}|\boldsymbol{h}} \left(\sum_j \boldsymbol{W}_{:,j} h^{(m)}_j \right), \boldsymbol{C}^{\mathrm{mc}}_{\boldsymbol{x}|\boldsymbol{h}} \right)$$

    - 多変量ガウス分布
    - 共分散行列: $\boldsymbol{C}^{\mathrm{mc}}_{\boldsymbol{x}|\boldsymbol{h}} = \left(\sum_j h^{(c)}_j \boldsymbol{r}^{(j)} \boldsymbol{r}^{(j)\top} + \boldsymbol{I} \right)^{-1}$

20.5.2 条件付き共分散の無向モデル

- **平均-スチューデントのt分布の積モデル**(mean-product of t-distribution, mPoT)
  - 観測変数における条件付き分布は多変量ガウス分布
  - 隠れ変数における条件付き分布は条件付き独立なガンマ分布を利用
  - エネルギー関数

$$E_{\mathrm{mPoT}}(\boldsymbol{x}, \boldsymbol{h}^{(m)}, \boldsymbol{h}^{(c)}) = E_m(\boldsymbol{x}, \boldsymbol{h}^{(m)}) + \sum_j \left(h_j^{(c)} \left(1 + \frac{1}{2} \left(\boldsymbol{r}^{(j)\top} \boldsymbol{x} \right)^2 \right) + (1 - \gamma_j) \log h_j^{(c)} \right),$$

20.5.2 条件付き共分散の無向モデル

- **スパイク-スラブ型RBM**(spike and slab RBM, ssRBM)
  - 隠れユニット
    - 2値の**スパイク**(spike)ユニット$\boldsymbol{h}$
    - 実数値の**スラブ**(slab)ユニット$\boldsymbol{s}$
  - 隠れユニットによって条件付けられた可視ユニットの平均を$(\boldsymbol{h} \odot \boldsymbol{s})\boldsymbol{W}^{\top}$で与える
  - エネルギー関数

$$E_{\text{ss}}(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{h}) = -\sum_i \boldsymbol{x}^{\top} \boldsymbol{W}_{:,i} s_i h_i + \frac{1}{2}\boldsymbol{x}^{\top}\left(\boldsymbol{\Lambda} + \sum_i \boldsymbol{\Phi}_i h_i\right)\boldsymbol{x} + \frac{1}{2}\sum_i \alpha_i s_i^2 - \sum_i \alpha_i \mu_i s_i h_i - \sum_i b_i h_i + \sum_i \alpha_i \mu_i^2 h_i.$$

  - $\boldsymbol{h}$が与えられた元での観測変数における条件付き分布

$$p_{\text{ss}}(\boldsymbol{x} \mid \boldsymbol{h}) = \frac{1}{P(\boldsymbol{h})}\frac{1}{Z}\int \exp\{-E(\boldsymbol{x}, \boldsymbol{s}, \boldsymbol{h})\}\, d\boldsymbol{s}$$

$$= \mathcal{N}\left(\boldsymbol{x}; \boldsymbol{C}^{\text{ss}}_{\boldsymbol{x}|\boldsymbol{h}} \sum_i \boldsymbol{W}_{:,i}\mu_i h_i\ ,\ \boldsymbol{C}^{\text{ss}}_{\boldsymbol{x}|\boldsymbol{h}}\right)$$

$$\boldsymbol{C}^{\text{ss}}_{\boldsymbol{x}|\boldsymbol{h}} = \left(\boldsymbol{\Lambda} + \textstyle\sum_i \boldsymbol{\Phi}_i h_i - \sum_i \alpha_i^{-1} h_i \boldsymbol{W}_{:,i}\boldsymbol{W}_{:,i}^{\top}\right)^{-1}$$

20.6 畳み込みボルツマンマシン

- 離散畳み込みによる行列計算の置き換え
  - 平行移動不変な空間的・時間的構造を持つ入力の計算量を削減する方法

- **確率的最大プーリング**(probabilistic max pooling)
  - 検出ユニットを制約して，1回につき最大で1つのユニットだけをアクティブにする
    - 合計で$n+1$個の状態を持つ(どれかがアクティブorどれもアクティブでない)
  - 検出ユニットは相互に排他的になる

20.7 構造化出力や系列出力のためのボルツマンマシン

- 条件付き分布$p(\mathbf{y}|\boldsymbol{x})$をモデリング
  - 構造のある出力 ex) 波形
  - 系列モデリング $p(\boldsymbol{x}^{(t)}|\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(t-1)})$

- RNN-RBN
  - 各時間ステップのRBMのパラメータを出力するRNNからなる，フレーム$\boldsymbol{x}^{(t)}$の系列の生成モデル
    - RNNは重みも含めたRBMの全てのパラメータを出力
  - RNNを通じて損失関数の勾配を誤差逆伝播

20.8 その他のボルツマンマシン

- ボルツマンマシンの変種はたくさん考えられる
  - $\log p(y|\boldsymbol{x})$を最大化を目標とする識別的RBM [Larochelle+ 2008]
  - 高次の項をエネルギー関数に持つボルツマンマシン[Sejnowski 1987] … etc

- ボルツマンマシンの枠組みは多くのモデル構造を実現できる
  but 全ての条件付き分布の扱いやすさを維持するエネルギー関数を見つける
  ことが困難
  - 革新的な発見が見つかる余地が残されている

20.9 確率的演算を通る誤差逆伝播

- 従来のNNは入力変数$\boldsymbol{x}$の決定論的な変換

- 生成モデルの場合は$\boldsymbol{x}$の確率的な変換を実装したいことが多い
  - 単純な確率分布からサンプルした入力$\boldsymbol{z}$を追加してNNを拡張する方法が存在
    - 一様分布や確率分布を利用
    - この場合，NNは内部的には決定論的な計算をしながら，
      $\boldsymbol{z}$を直接観測することのできない観測者には$f(\boldsymbol{x}, \boldsymbol{z})$が確率的に現れる
      - fが連続かつ微分可能ならば誤差逆伝播(BP)を用いて勾配を計算できる

20.9 確率的演算を通る誤差逆伝播

- ex) 平均μ，分散σ^2を持つガウス分布からサンプルyを抽出するとき

$$\mathrm{y} \sim N(\mu, \sigma^2)$$

  - このままではBPできない(yはサンプリング過程によって生成されているので)

- $\mathrm{z} \sim N(z; 0,1)$を用いて $y = \mu + \sigma z$ と書き換えられる
  - 追加的な入力zを持つ決定的な演算とみなせるので誤差逆伝播可能
    - **再パラメータ化トリック**

20.9 確率的演算を通る誤差逆伝播

- **再パラメータ化トリック**(reparameterization trick)
  - さっきの例の一般化
  - $p(\mathrm{y};\boldsymbol{\theta})$や$p(\mathrm{y};\boldsymbol{x},\boldsymbol{\theta})$の形式をとる分布を$p(\mathrm{y}|\boldsymbol{\omega})$と表現し($\boldsymbol{\omega}$は$\boldsymbol{\theta}$や$\boldsymbol{x}$を含む変数)
    分布$p(\mathrm{y}|\boldsymbol{\omega})$から値yをサンプリングするとき

$$\mathbf{y} \sim p(\mathbf{y}|\boldsymbol{\omega})$$

    を

$$\mathbf{y} = f(\boldsymbol{z};\boldsymbol{\omega})$$

    に書き換え可能($\boldsymbol{z}$はランダム性の発生源)
    - fがほとんど至るところで連続かつ微分可能なときBPが利用可能
      - $\boldsymbol{y}$は連続値であることが必要
        - » 離散値のサンプルを生成するサンプリング過程を利用するときは,
          REINFORCEアルゴリズムなどを利用して$\boldsymbol{\omega}$の勾配を推定する

20.9.1 離散的な確率的演算を通る誤差逆伝播

- **REINFORCEアルゴリズム**
  - モデルが離散変数$\boldsymbol{y}$を出力する場合，再パラメータ化トリックを利用できない (後注)
    - つまりコスト関数$J(\boldsymbol{y})$を微分してもモデルパラメータ$\boldsymbol{\theta}$の有益な勾配が得られない
  - 強化学習アルゴリズムを使う
  - 期待コスト$\mathbb{E}_z[J(\boldsymbol{y})]$を利用して勾配を推定する

$$\mathbb{E}_z[J(\boldsymbol{y})] = \sum_{\boldsymbol{y}} J(\boldsymbol{y}) p(\boldsymbol{y})$$

  - このとき勾配は

$$\begin{aligned}
\frac{\partial \mathbb{E}[J(\boldsymbol{y})]}{\partial \boldsymbol{\omega}} &= \sum_{y} J(\boldsymbol{y}) \frac{\partial p(\boldsymbol{y})}{\partial \boldsymbol{\omega}} \\
&= \sum_{y} J(\boldsymbol{y}) p(\boldsymbol{y}) \frac{\partial \log p(\boldsymbol{y})}{\partial \boldsymbol{\omega}} \\
&\approx \frac{1}{m} \sum_{\mathbf{y}^{(i)} \sim p(\boldsymbol{y}),\, i=1}^{m} J(\boldsymbol{y}^{(i)}) \frac{\partial \log p(\boldsymbol{y}^{(i)})}{\partial \boldsymbol{\omega}}
\end{aligned}$$

20.9.1 離散的な確率的演算を通る誤差逆伝播

- **分散減少**(variance reduction)法
  - 単純にREINFORCEによって勾配を推定すると分散が大きい
    - 良い勾配の推定量を得るためにはたくさんのサンプルが必要
  - **ベースライン**(baseline)を導入する
    - コスト関数J($\boldsymbol{y}$)の補正
    - 推定勾配の期待値は変わらない

$$E_{p(\boldsymbol{y})}\left[(J(\boldsymbol{y})-b(\boldsymbol{\omega}))\frac{\partial \log p(\boldsymbol{y})}{\partial \boldsymbol{\omega}}\right] = E_{p(\boldsymbol{y})}\left[J(\boldsymbol{y})\frac{\partial \log p(\boldsymbol{y})}{\partial \boldsymbol{\omega}}\right] - b(\boldsymbol{\omega})E_{p(\boldsymbol{y})}\left[\frac{\partial \log p(\boldsymbol{y})}{\partial \boldsymbol{\omega}}\right] \tag{20.66}$$

$$= E_{p(\boldsymbol{y})}\left[J(\boldsymbol{y})\frac{\partial \log p(\boldsymbol{y})}{\partial \boldsymbol{\omega}}\right]. \tag{20.67}$$

    - 最適なベースライン

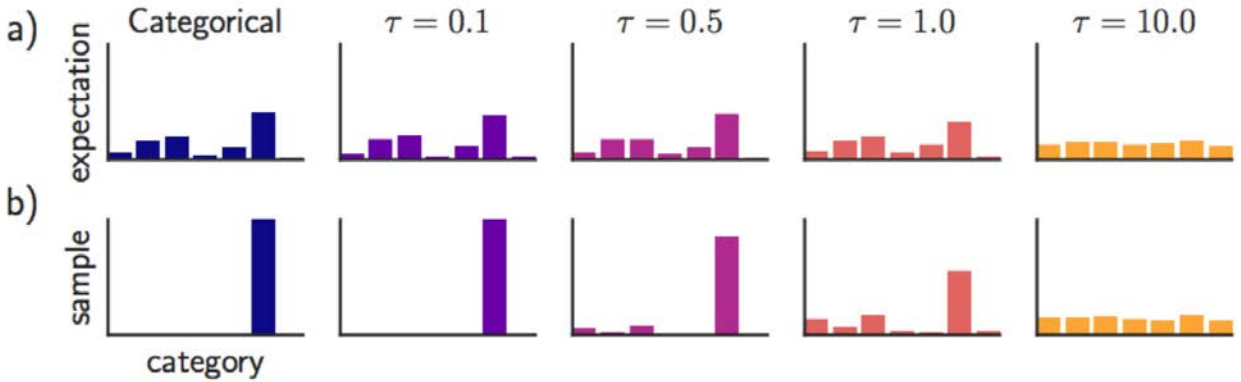
$$b^*(\boldsymbol{\omega})_i = \frac{E_{p(\boldsymbol{y})}\left[J(\boldsymbol{y})\frac{\partial \log p(\boldsymbol{y})}{\partial \omega_i}^2\right]}{E_{p(\boldsymbol{y})}\left[\frac{\partial \log p(\boldsymbol{y})}{\partial \omega_i}^2\right]}$$

20.9.1 離散的な確率的演算を通る誤差逆伝播

- **Gumbel-Softmax** [Jang+ 2016]
  - 微分可能な離散サンプルを近似的に生成できるような分布
    - π_iはニューラルネットの出力，g_iはノイズ，τは温度パラメータ

$$y_i = \frac{\exp((\log(\pi_i) + g_i)/\tau)}{\sum_{j=1}^{k} \exp((\log(\pi_j) + g_j)/\tau)} \qquad \text{for } i = 1, ..., k.$$

  - 温度が低いとone-hotに，温度が高いと一様分布になる



  - http://musyoku.github.io/2016/11/12/Categorical-Reparameterization-with-Gumbel-Softmax/ に詳しい解説あり

20.10 有向生成ネットワーク

- 16章の有向グラフィカルモデルの利用

- 深層学習コミュニティでは2013年ぐらいまでRBMまでの無向グラフィカルモデルに押されて注目されてこなかったらしい

- 鈴木さんのSlideShareがわかりやすいです
  - GAN(と強化学習の関係) https://www.slideshare.net/masa_s/gan-83975514

20.10.1 シグモイド信念ネットワーク

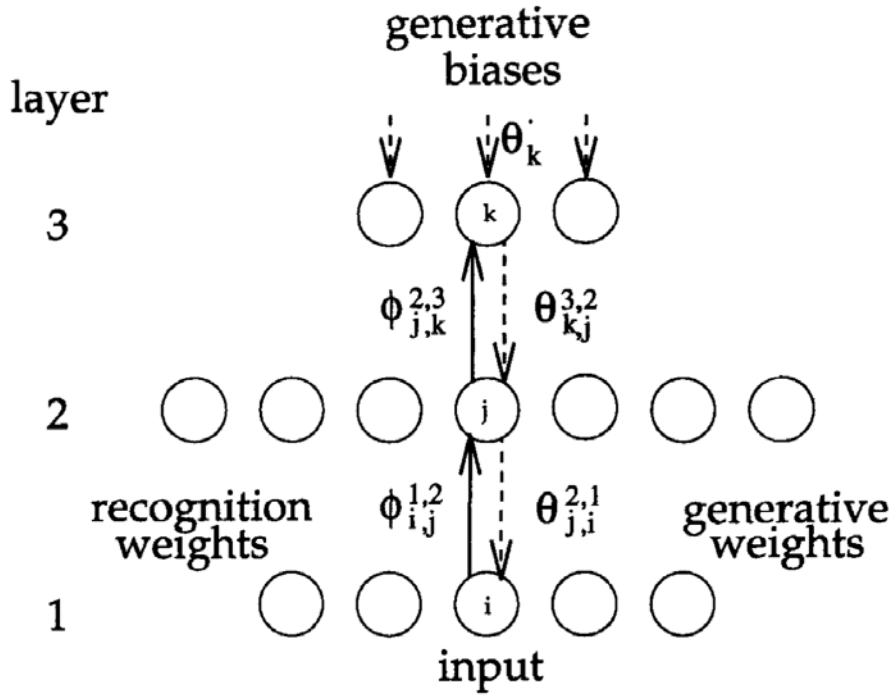
- 特定の種類の条件付き確率分布を持つ単純な形式の有向グラフィカルモデル
  - 2値状態のベクトル$\boldsymbol{s}$を持ち，状態の各要素が先祖からの影響を受ける

$$p(s_i) = \sigma\left(\sum_{j<i} W_{j,i} s_j + b_i\right)$$

  - 多数の隠れ層を通る伝承サンプリングにより，可視層が生成される

20.10.1 シグモイド信念ネットワーク

- **ヘルムホルツマシン** [Dayan+ 1995]
  - 隠れユニットにおける平均場分布のパラメータを予測する推論ネットワークと組み合わせたシグモイド信念ネットワーク



  - https://www.slideshare.net/beam2d/learning-generator

20.10.2 微分可能な生成器ネットワーク

- 多くの生成モデルでは，微分可能な**生成器ネットワーク**(generator network)を用いる
  - 微分可能な関数$g(\mathbf{z}; \boldsymbol{\theta}^{(g)})$を用いて，潜在変数$\mathbf{z}$のサンプルをサンプル$\mathbf{x}$や，サンプル$\mathbf{z}$における分布に変換する
    - 生成器ネットワークと推論ネットワークを組み合わせる→VAE
    - 生成器ネットワークと識別器ネットワークを組み合わせる→GAN

- 生成器ネットワークの定式化
  - ① サンプルを直接出力するアプローチ
  - ② 条件付き分布のパラメータを出すアプローチ

20.10.2 微分可能な生成器ネットワーク

- ① サンプルを直接出力するアプローチ
  - gを用いて$\boldsymbol{x}$のサンプルを直接提供する

$$p_x(\boldsymbol{x}) = \frac{p_z(g^{-1}(\boldsymbol{x}))}{\left|\det\left(\frac{\partial g}{\partial \boldsymbol{z}}\right)\right|}$$

  - 設計者が分布を仮定する必要がない

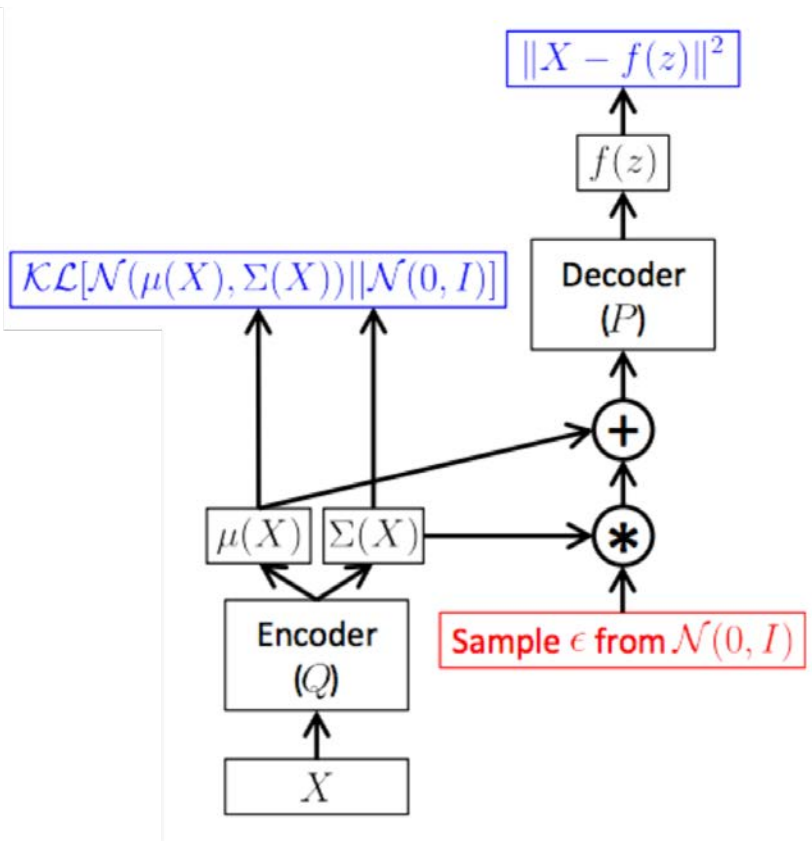
- ② 条件付き分布のパラメータを出すアプローチ
  - gを用いて$\boldsymbol{x}$の条件付き分布を定義する
  - 離散データも生成できる
    - ex) シグモイド出力を持つ生成器ネットワークでベルヌーイ分布の平均パラメータを提供する

$$p(\mathrm{x}_i = 1 \mid \boldsymbol{z}) = g(\boldsymbol{z})_i$$

$$p(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{z}} p(\boldsymbol{x} \mid \boldsymbol{z})$$

20.10.3 変分自己符号化器(VAE)

- **変分自己符号化器**(variational autoencoder, VAE)
  - 学習による近似推論を用い，勾配に基づく方法で訓練できる有向モデル



  - https://arxiv.org/abs/1606.05908

20.10.3 変分自己符号化器(VAE)

- VAEの訓練
  - データ点$\boldsymbol{x}$に関連する変分下界$\mathcal{L}(q)$を最大化することによって訓練できる

$$\begin{aligned}\mathcal{L}(q) &= \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})} \log p_{\text{model}}(\boldsymbol{z}, \boldsymbol{x}) + \mathcal{H}(q(\mathbf{z} \mid \boldsymbol{x})) \\ &= \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})} \log p_{\text{model}}(\boldsymbol{x} \mid \boldsymbol{z}) - D_{\text{KL}}(q(\mathbf{z} \mid \boldsymbol{x}) || p_{\text{model}}(\mathbf{z})) \\ &\leq \log p_{\text{model}}(\boldsymbol{x}).\end{aligned}$$

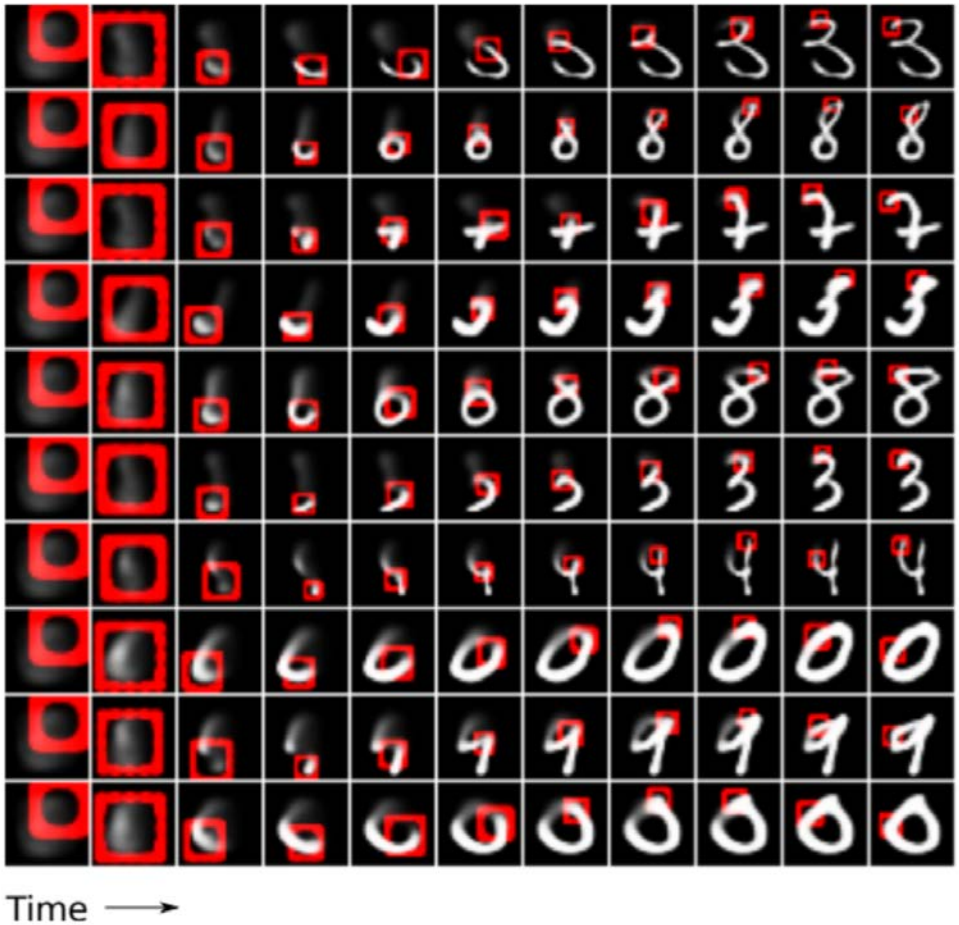
    - 2番目の式
      - 第1項: 再構成による対数尤度
      - 第2項: 近似事後分布$q(\boldsymbol{z}|\boldsymbol{x})$とモデル分布$p_{model}(\boldsymbol{z})$を近づける

20.10.3 変分自己符号化器(VAE)

- VAEの主な欠点
  - 生成画像がぼやける
    - 最尤法の効果
      - $D_{KL}(p_{data}||p_{model})$の最小化

20.10.3 変分自己符号化器(VAE)

- **DRAW**(deep recurrent attention writer)
  - アテンションメカニズムを組み合わせた回帰結合型の符号化器と復号化器を用いる
  - 異なる小さな画像パッチに順次アクセスし，それらの点のピクセルを描く



20.10.4 敵対的生成ネットワーク(GAN)

- 生成器ネットワークが敵対者と競争するゲーム理論的シナリオに基づく
  - 生成器ネットワーク: サンプル$\boldsymbol{x} = g(\boldsymbol{z}; \boldsymbol{\theta}^{(g)})$を直接生成
  - 識別器ネットワーク: $\boldsymbol{x}$が真の訓練事例である確率$d(\boldsymbol{x}; \boldsymbol{\theta}^{(d)})$を出力

- 最も簡単な定式化: ゼロサムゲーム

$$g^* = \arg\min_g \max_d v(g, d)$$

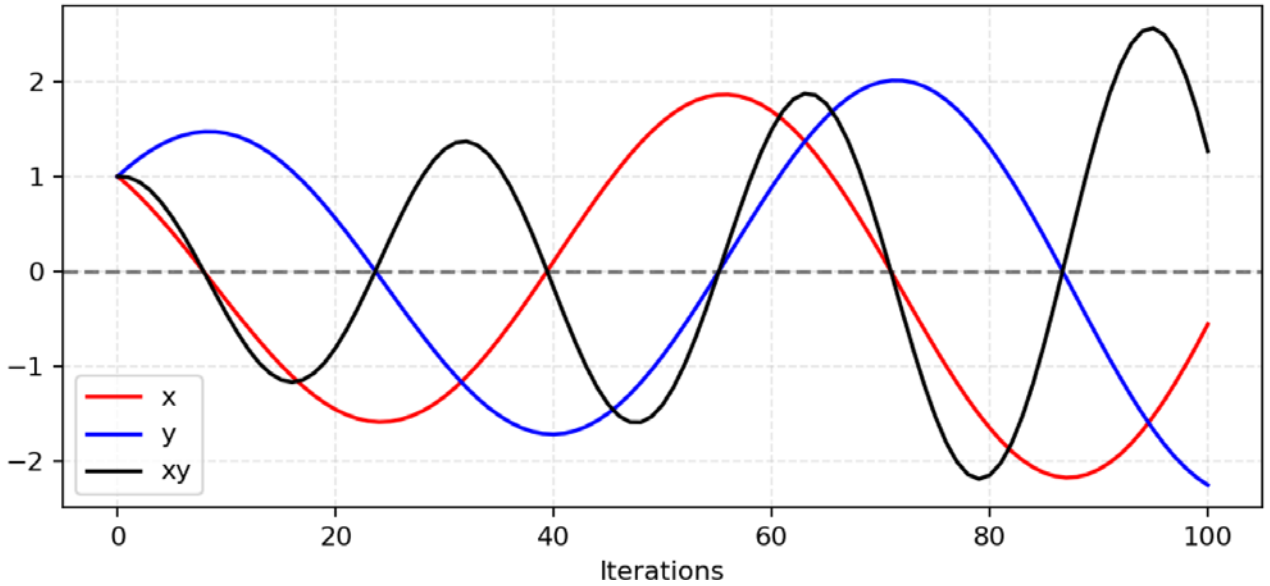
  - 目的関数$v(g, d)$は通常交差エントロピー誤差関数を用いる

$$v(\boldsymbol{\theta}^{(g)}, \boldsymbol{\theta}^{(d)}) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} \log d(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{x} \sim p_{\text{model}}} \log (1 - d(\boldsymbol{x}))$$

- 学習過程で近似推論も分配関数の勾配の近似も必要ないのがgood
  - $\boldsymbol{\theta}^{(g)}$において$\max_d v(g, d)$が凸の場合収束し，漸近的に一致

20.10.4 敵対的生成ネットワーク(GAN)

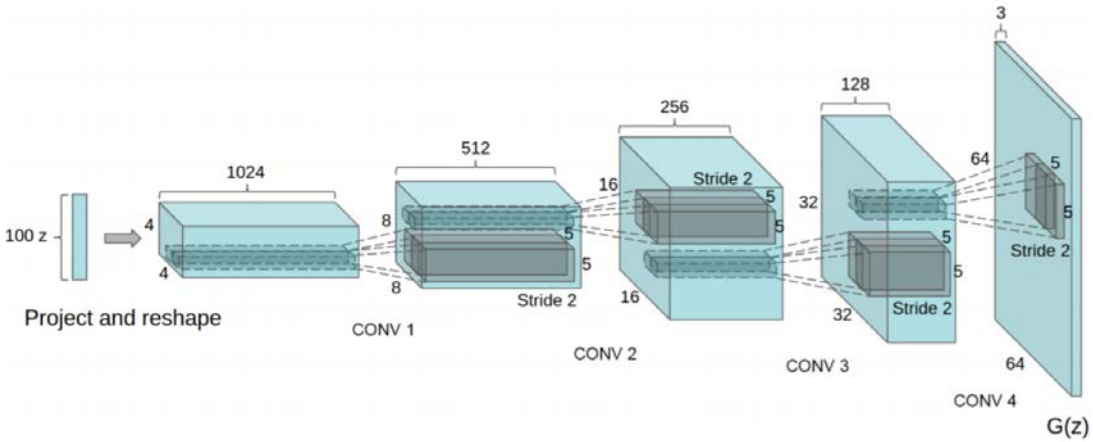
- GANの学習が困難な場合
  - $\max_d v(g,d)$が非凸の場合(NNでは非凸)
    - ex) $v(a,b)=ab$としてGとDで交互に最小化・最大化を繰り返すとき



- https://lilianweng.github.io/lil-log/2017/08/20/from-GAN-to-WGAN.html

20.10.4 敵対的生成ネットワーク(GAN)

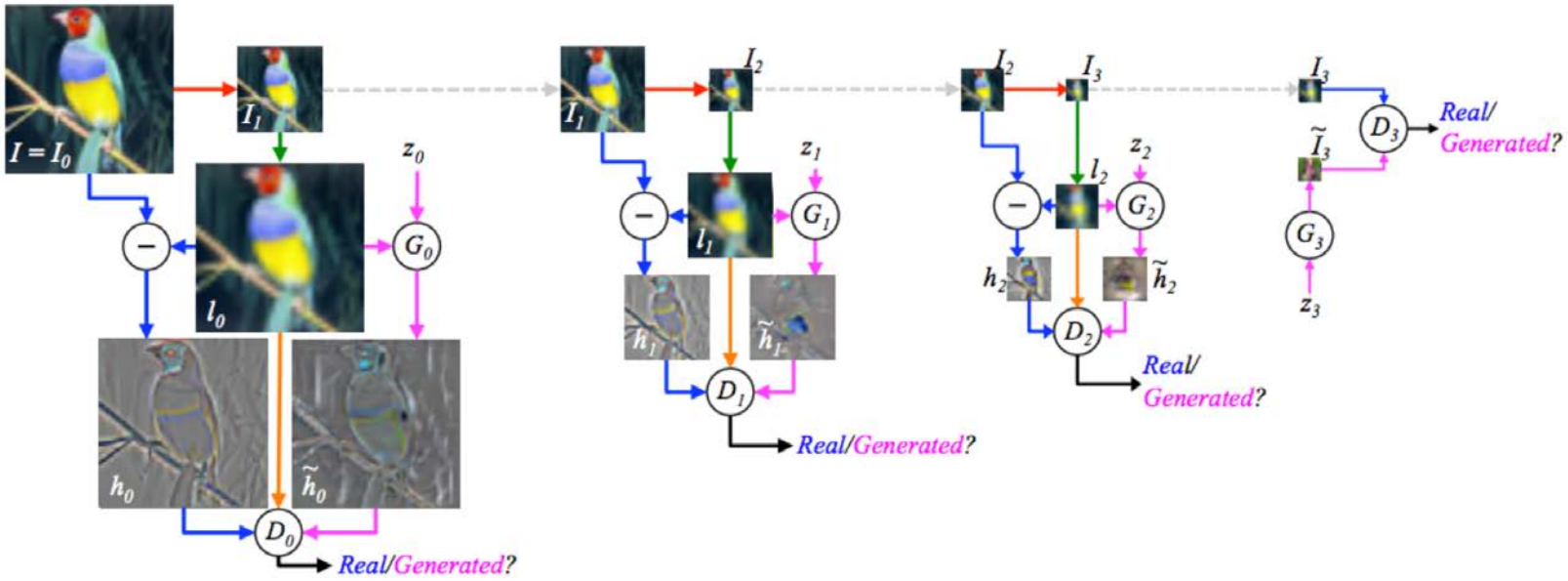
- **深層畳み込みGAN**(deep convolutional GAN, DCGAN) [Radford+ 2015]
  - 潜在表現空間が重要な変動の要因を捉えていることを示した
  - Generatorのネットワーク



  - 生成画像

20.10.4 敵対的生成ネットワーク(GAN)

- LAPGAN [Denton+ 2015]
  - 最初は非常に低解像度な画像を生成し，徐々に画像に詳細を追加するように訓練
  - ラプラシアンピラミッドを用いて様々な段階の詳細を持つ画像を生成



20.10.5 モーメントマッチング生成ネットワーク

- **モーメントマッチング**(moment matching)
  - モデルによって生成されるサンプルの統計量の多くが，訓練集合に含まれる事例の統計量とできるだけ近くなるように生成器を訓練する
    - n次のモーメント: 確率変数のn乗の期待値

$$\mathbb{E}_{\boldsymbol{x}} \prod_i x_i^{n_i}$$

- **maximum mean discrepancy**(MMD)をコスト関数として最小化
  - 無限次元空間における一次モーメントの誤差を測るのに，カーネル関数で定義された特徴量空間への暗黙的な写像を用いることで，無限次元ベクトルの計算を扱いやすくする
    - カーネル法関連の資料を見てください
      - http://www.ism.ac.jp/~fukumizu/OsakaU2014/OsakaU_6kernelMean.pdf

20.10.6 畳み込み生成ネットワーク

- 畳み込み構造を持つ生成器ネットワークを利用することで，パラメータ共有のない全結合層を用いるよりも少ないパラメータ数ですむ

- **アンプーリング**(unpooling)
  - 表現の空間サイズを大きくする
  - 特定の単純化した条件下でのmax-poolingの逆演算に相当

参考文献 1

- 論文
  - [Dayan+ 1995] The Helmholtz Machine
    - http://www.gatsby.ucl.ac.uk/~dayan/papers/hm95.pdf
  - [Denton+ 2015] Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks
    - https://arxiv.org/abs/1506.05751
  - [Gregor+ 2015] DRAW: A Recurrent Neural Network For Image Generation
    - https://arxiv.org/abs/1502.04623
  - [Jang+ 2016] Categorical Reparameterization with Gumbel-Softmax
    - https://arxiv.org/abs/1611.01144
  - [Radford+ 2015] Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks
    - https://arxiv.org/abs/1511.06434

参考文献 2

- Deep Learning
  - Ian Goodfellow, Yoshua Bengio, Aaron Courville
  - 日本語版
    https://www.amazon.co.jp/%E6%B7%B1%E5%B1%A4%E5%AD%A6%E7%BF%92-Ian-Goodfellow/dp/4048930621

Deep Learning 輪読会 2017

第20章 深層生成モデル
(20.10.7 自己回帰ネットワーク～20.15 結論)

2018.02.05

中村藤紀

構成

20.10.7 自己回帰ネットワーク
20.10.8 線形自己回帰ネットワーク
20.10.9 ニューラル自己回帰ネットワーク
20.10.10 NADE

20.11 自己符号化器からのサンプリング
20.11.1 任意の雑音除去自己符号化器に関連付けられるマルコフ連鎖
20.11.2 クランピングと条件付きサンプリング
20.11.3 ウォークバック訓練手続き

20.12 生成的確率ネットワーク
20.12.1 識別 GSN

20.13 他の生成スキーム

20.14 生成モデルの評価

20.15 結論

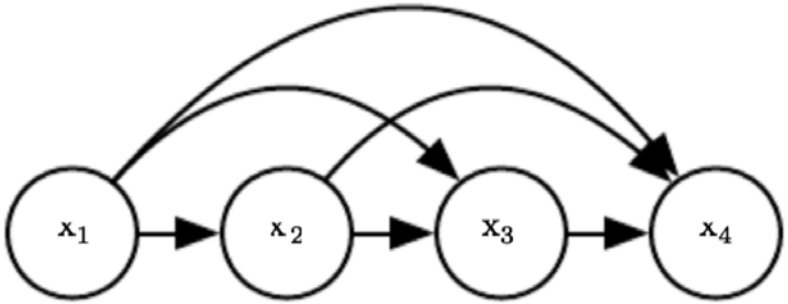
20.10.7 自己回帰ネットワーク

- 自己回帰ネットワーク
  - 潜在確率変数を持たない有向確率モデル
  - 確率の連鎖律を用いて観測変数における同時確率を分解 $P(x_d \mid x_{d-1}, \ldots, x_1)$
  - **完全可視的ベイジアンネットワーク** (すべての変数が可視変数のベイジアンネットワーク, fully-visible Bayes network, FVBN)
  - NADE (20.10.10節) のような一部の自己回帰ネットワークでは，パラメータ共有.
    - 統計的優位性 (より少ない固有のパラメータ)
    - 計算的優位性 (より少ない計算量)

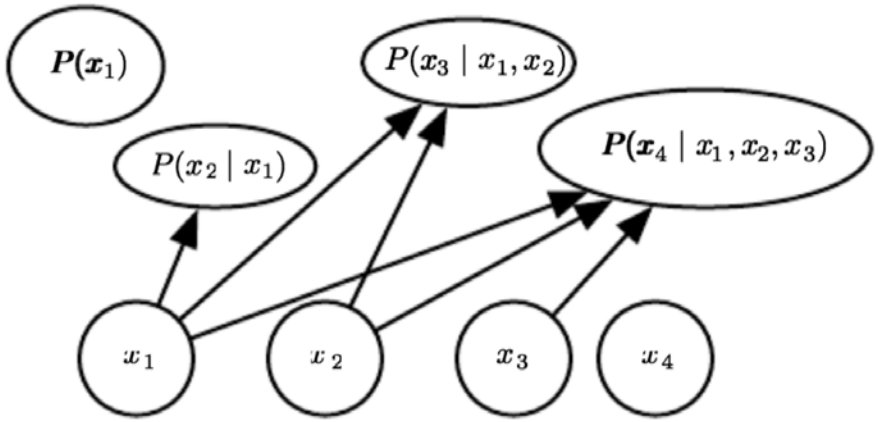
    (→ 特徴量の再利用という深層学習で繰り返されるテーマの例)

20.10.8 線形自己回帰ネットワーク

- 最も単純な形式の自己回帰ネットワーク
  - 隠れユニットなし，パラメータや特徴量の共有なし.
  - i 番目の変数を，その前の $i-1$ 個の変数から予測する完全可視的信念ネットワーク.



FVBN の有効グラフィカルモデル

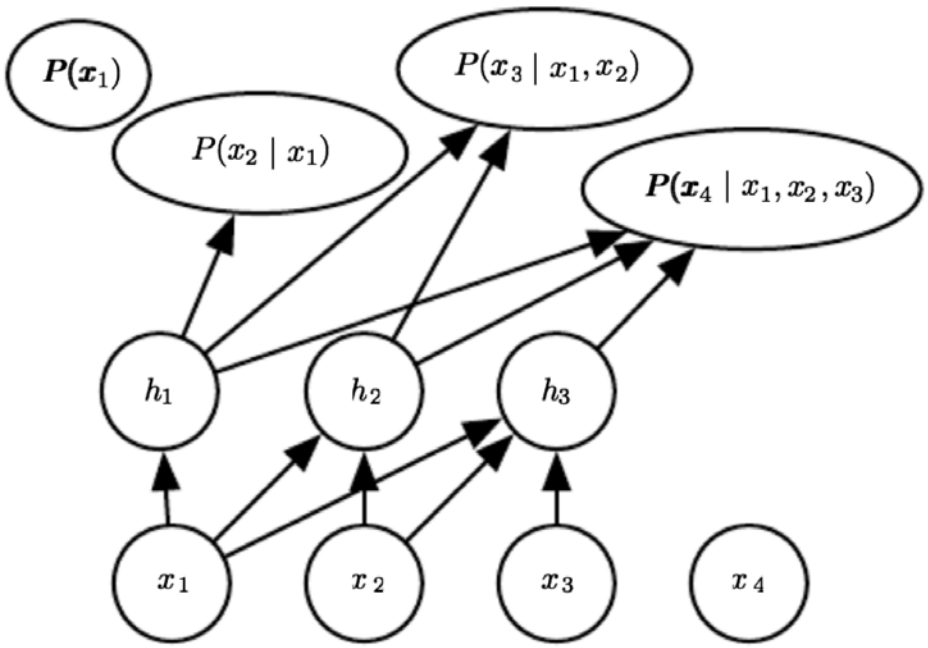


ロジスティック FVBN の計算グラフ

- 線形自己回帰ネットワーク
  - 本質的には線形分類法の生成モデリングへの一般化.
  - したがって，線形分類器と同じ利点と欠点.
    - 凸な損失関数で訓練可能，閉形式の解が許容される.
    - 容量を増やすには，入力の基底展開やカーネルトリックのようなテクニックを用いる.

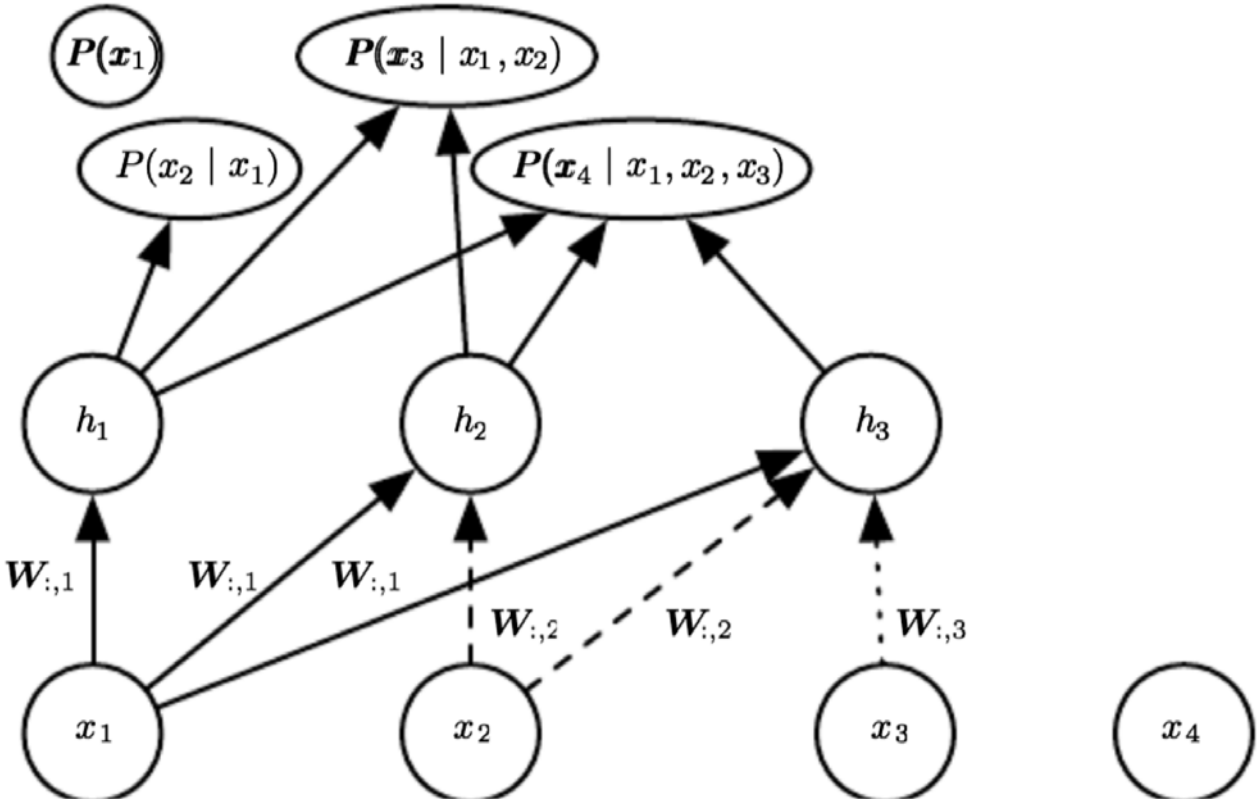
20.10.9 ニューラル自己回帰ネットワーク

- ニューラル自己回帰ネットワーク
  - 容量を必要なだけ増やせ，任意の同時分布を近似できる.
  - 1. 表形式のグラフィカルモデルで生じる次元の呪いを回避.
  - 2. 再利用原理 (reuse principle)
    - 各 x_i の予測に別々のニューラルネットワークを使う代わりに，左から右への接続を使って，すべてのニューラルネットワークを1つに結合することが可能 (下図)



20.10.10 NADE

- **ニューラル自己回帰密度推定器**
  - neural auto-regressive density estimator, NADE
  - ニューラル自己回帰ネットワークのうち，近年非常に成功した形式.
  - 初期のニューラル自己回帰ネットワークに追加のパラメータ共有を導入.



20.10.10 NADE

- ニューラル自己回帰アーキテクチャのもう1つの興味深い拡張
  - 観測変数についての任意の順序を選択する必要性を除外.
  - 自己回帰ネットワークにおいて，順序をランダムにサンプリングし，どの入力が観測され，どれが予測されるかを特定する情報を隠れユニットに与える.
    - → 任意の順序に対応できるようにネットワークを学習.
  - 訓練した自己回帰ネットワークを使って，効率的に任意の推論問題を実行可能.

    (= 変数の任意の部分集合の下で，任意の部分集合における確率分布から予測やサンプリングが可能)
  - 変数の各順序 o によって様々な $p(\boldsymbol{x} \mid o)$ が得られるので，様々な o に対してモデルをアンサンブルできる.

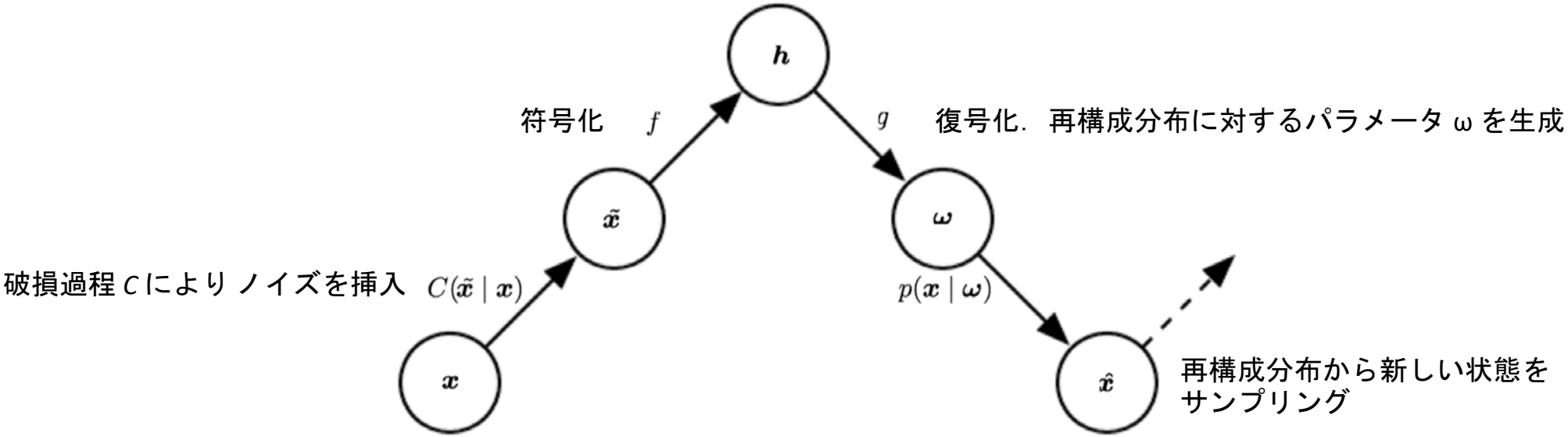
$$p_{\text{ensemble}}(\mathbf{x}) = \frac{1}{k}\sum_{i=1}^{k} p(\mathbf{x} \mid o^{(i)})$$

20.11 自己符号化器からのサンプリング

- 14章で，多くの種類の自己符号化器がデータ分布を学習することを見た.
  - スコアマッチング，denoising AE (雑音除去自己符号化器)，縮小自己符号化器，...
  - このようなモデルからサンプルを抽出する方法は，まだ見ていない.
    - VAE のようなある種の自己符号化器は，明示的に確率分布を表現しており，単純な伝承サンプリングが可能.
    - その他のほとんどの種類の自己符号化器は，MCMC サンプリングが必要.

20.11.1 任意の雑音除去自己符号化器に関連づけられるマルコフ連鎖

- Bengio et al. (2013c) は，**一般化雑音除去自己符号化器** (generalized denoising autoencoders) のためのマルコフ連鎖構築方法を提案.
  - 一般化雑音除去自己符号化器は，破損した入力が与えられた下で，きれいな入力の推定値をサンプリングする雑音除去分布によって規定される.
  - 推定分布から生成されるマルコフ連鎖の各ステップは，以下の通り.



20.11.2 クランピングと条件付きサンプリング

- ボルツマンマシンと同様に，denoising AE やその一般化を用いて，条件付き分布 $p(x_f \mid x_o)$ からサンプリングすることができる.
  - 観測ユニット x_f をクランピング (固定, clamping) し，x_f と (もし存在するならば) サンプリングされた潜在変数が与えられた下で，自由ユニット x_o を再サンプリング.

画像の右半分をクランピングし，各ステップで右半分だけで再サンプリングしてマルコフ連鎖を実行した例

20.11.3 ウォークバック訓練手続き

- ウォークバック訓練手続き (walk-back training procedure)
  - denoising AE の生成的訓練の収束を加速する方法.
  - 1ステップの符号化-復号化再構成を実行する代わりに，(マルコフ連鎖を生成する場合を同様に) 別の複数の確率的な符号化-復号化ステップから成る.
    - 訓練事例で初期化，最後の確率的再構成 (または途中のすべての再構成) にペナルティ.
  - データから離れている偽モードをより効率的に除去できるという利点.

20.12 生成的確率ネットワーク
20.12.1 識別 GSN

- **生成的確率ネットワーク** (generative stochastic network, GSN)
  - denoising AE を一般化.
  - 可視変数 **x** に加えて，生成するマルコフ連鎖における潜在変数 **h** を含む.
  - 1. $p(\mathbf{x}^{(k)} \mid \mathbf{h}^{(k)})$ は，現在の状態が与えられた下で，次の可視変数を生成する方法を閉示す．このような「再構成分布」は，denoising AE, RBM, DBN, DBM にも見られる.
  - 2. $p(\mathbf{h}^{(k)} \mid \mathbf{h}^{(k-1)}, \mathbf{x}^{(k-1)})$ は，前の潜在状態と可視変数の下で，潜在状態変数を更新する方法を示す.

- 識別 GSN
  - GSN の元の定式化は，教師なし学習のモデリングのためのものであり，観測データ **x** について暗黙的に $p(\mathbf{x})$ をモデリング.
  - この枠組みを $p(y \mid x)$ を最適化するように修正可能.

20.13 他の生成スキーム

- **拡散逆変換** (diffusion inversion)
  - 非平衡熱力学に基づいた訓練スキーム.
  - サンプリングしたい確率分布には構造があると仮定.
    - この構造は，確率分布をより多くのエントロピーを持つように少しずつ変化させる拡散過程によって，徐々に破壊される.
    - この過程を逆向きに実行して，構造化されていない分布に対して構造を徐々に修復するようにモデルを訓練.
    - 分布を目標の分布に近づける過程を反復的に適用することで，徐々に目標の分布に近づける.
  - 拡散逆変換により，学習器は，データ点周辺の密度の形をより正確に学習でき，かつ，データ点から遠くに現れる偽モードを取り除ける.

- **近似ベイズ計算** (approximate Bayesian computation, ABC)
  - サンプルの選択された関数のモーメントが，求めたい分布のモーメントに一致するように，サンプルが棄却または修正される.

- 今後，生成モデリングに対する他の多くのアプローチが発見されることを期待.

20.14 生成モデルの評価

- 生成モデルの評価は，評価指標そのものが困難な研究課題.
  - モデルを公平に比較する方法を確立するのは非常に困難.
- 生成モデリングでは，識別モデルとは異なり，前処理の変更はまったく許容されない.
  - 入力データの変更は，捉えられる分布を変え，タスクを根本的に変える.
- MNIST で一般に発生する前処理の問題
  - 実数値モデルは実数値モデルと，二値モデルは二値モデルと比較すること.
- データ分布から本物に近いサンプルを生成できることが生成モデルの目標の1つ
  - サンプルを視覚的に検査して評価.
  - モデルが単に訓練事例をコピーしているだけかどうかを検証.
    - ユークリッド距離に基づく最近傍を示す.
- 生成モデルを使う意図と測定基準の選択が一致しなければならない.
  - 本物に近い点に対して高い確率を割り当てることに優れている生成モデルもあれば，本物から遠い点に対して高い確率を割り当てないことに優れている生成モデルもある.
  - 生成モデルの改善とともに，それを測定する新しい技術を設計することも重要な研究課題.

20.15 結論

- 隠れユニットを持つ生成モデルを訓練することは，与えられた訓練データによって表現される世界を理解するモデルを作るための強力な方法.

- 生成モデルは,
  - モデル $p_{\text{model}}(x)$ と表現 $p_{\text{model}}(h \mid x)$ を学習することによって，x 内の入力変数間の関係についての多くの推論問題に対する答えを提供したり,
  - 階層構造の異なる層での h の期待値を取ることで x を表現するさまざまな方法を提供.
  - AI システムに，理解しなければならないさまざまな直感的な概念の枠組みを提供.

- 読者が，生成モデルのアプローチをより強力にする新手法を発見，学習と知能の基礎となる原理を理解する取り組みを続けてくれることが期待される.

参考文献

- Deep Learning
  - Ian Goodfellow, Yoshua Bengio, Aaron Courville
  - 日本語版
    https://www.amazon.co.jp/%E6%B7%B1%E5%B1%A4%E5%AD%A6%E7%BF%92-Ian-Goodfellow/dp/4048930621