広島県立総合技術研究所東部工業技術センター研究報告　No. 36（2023）
技術ノート

深層学習による異常検知手法の簡単な比較（第 4 報）

ボルツマンマシン

廣川　勝久

Anomaly Detections using Deep Learning (Ⅳ)

Boltzmann Machines

HIROKAWA Katsuhisa

正常信号のみの学習から，異常信号を検知できる深層学習生成モデルは機械学習の実用化にとって重要である。現在オートエンコーダを始めとする各種生成モデルが報告されている。本報告では，生成モデルの一種であるボルツマンマシンについて，様々な実装モデルと生成能力の評価を行ったので，その結果について述べる。

キーワード：ボルツマンマシン，Boltzmann Machine, 異常検知，生成 AI，最尤度関数，PyTorch，深層学習，Python

1. 緒　　言

量子コンピュータの実用化が目前に迫り，量子コンピュータの効率的な計算能力を使って，時間短縮ができるような応用問題の調査研究が進められている。特に量子コンピュータでは機械学習のように多数のデータを単純に並列処理するような分野の応用に適しているとされている。ボルツマンマシン[1,2]は量子状態のスピンを模した機械学習法であり，量子コンピュータとの親和性が他の機械学習と比較して高いことが期待されている。ボルツマンマシンを量子コンピュータに実装できれば，高速に大規模な機械学習が可能となる。

また，現在の多くの生成 AI が尤度関数から新しいデータ生成が行われるように，ボルツマンマシンも，確率分布に基づく機械学習であり，量子エネルギー値の最尤度パラメータを学習により求める。よって，ボルツマンマシンも生成 AI の一種であり，異常検知や次元数削減による特徴抽出への応用が可能であると考えられる。今後の生成 AI において一定の基盤技術になることが予想される。

これまでの報告では，オートエンコーダ（**A**uto **E**ncoder）にカルバックライブラーダイバージェンスによる最尤度関数を学習に用いることによって，オートエンコーダの汎化能力が向上することを示した[3]。また，畳み込みニューラルネットワークと最尤度関数によるオートエンコーダ組み合わせ，2 次元データの汎化能力について報告を行った[4]。更にオートエンコーダにより生成される潜在変数の自動クラスタリングについても報告を行った[5]。

本報告では，オートエンコーダと同様にボルツマンマシンの汎化能力による異常データの修正能力について報告する。モデルは，可視ノードが 2 値や連続値，潜在ノードが 2 値や連続値のものについて，また，パラメータの計算についても 3 種類の方法について検討を行った。

2. 生成モデルの概要

2.1 生成モデルとボルツマンマシン

現在，異常検知に利用可能な深層学習による生成モデルには，オートエンコーダや敵対的生成ネットワーク（**G**enerative **A**dversarial **N**etworks：**GAN**）などが提案されている。これらのモデルの学習では実際のデータを直接学習データに用いて損失計算が行われる。一方，ボルツマンマシンは，学習データが取り得る確率分布を学習する生成モデルである。すなわち，このようなモデルでは，学習データが取り得る確率的な領域を求めることができるため，学習データの近傍データからも，学習データに近い典型的なデータ復元が可能と考えられる。

2.2 ボルツマンマシン

従来のニューラルネットワークは，入力から出力への一方向データの流れを持つモデルで表されるが，ボルツマンマシンは，**図 1** のように各ノードが双方向に結合した構造を持つ相互結合モデルである。それぞれのノード

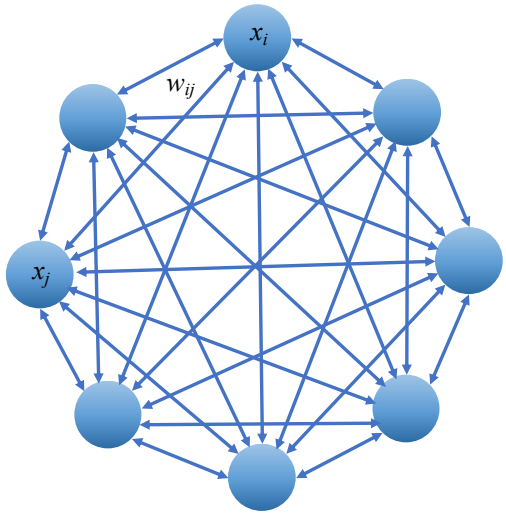

図 1 相互結合のボルツマンマシン

2023. 7. 12　加工技術研究部　受理

は 0 または 1 のバイナリー値を持つ。ここで，各ノード状態$\boldsymbol{x} = (x_1, x_2, \cdots x_i \cdots)$のボルツマンマシンが取り得る確率分布$p(x|\theta)$を，$\theta$をパラーメータとして次式のように表す。

$$p(x|\theta) = \frac{1}{Z(\theta)} \exp[-\Phi(x,\theta)] \tag{1}$$
$$\because Z(\theta) = \sum_x \exp[-\Phi(x,\theta)]$$

ここで$\Phi(x,\theta)$はボルツマンマシンのエネルギー関数

$$\Phi(x,\theta) = -\sum_i b_i x_i - \sum_{i,j} w_{ij} x_i x_j \tag{2}$$

と定義する。b_iはノードのバイアス，w_{ij}はノード間の重みを表す。ボルツマンマシンの学習は，取り得るノード状態に対して，パラメータθ（ここではb_i, w_{ij}）により，発生する確率分布$p(x|\theta)$を真の確率分布$p(x)$に近づけることを意味する。パラメータθに対する尤度関数を$L(\theta)$とすると，$L(\theta)$は

$$L(\theta) = \prod_{n=1}^{M} p(x^n|\theta) \tag{3}$$

により与えられる。ここでは，尤度関数$L(\theta)$を対数化することにより確率分布関数の乗算から積算演算に変換する。

$$\begin{aligned} \log L(\theta) &= \sum_{n=1}^{M} \log[p(x^n|\theta)] \\ &= \sum_{n=1}^{M} -\Phi(x^n,\theta) - \log[Z(\theta)] \end{aligned} \tag{4}$$

パラメータθに対する尤度関数の勾配計算は

$$\begin{aligned} \frac{\log L(\theta)}{\partial b_i} &= \sum_{n=1}^{M} x_i^n - N E_\theta(x_i) \\ \frac{\log L(\theta)}{\partial w_{ij}} &= \sum_{n=1}^{M} x_i^n x_j^n - N E_\theta(x_i x_j) \end{aligned} \tag{5}$$
$$\because E_{p(x|\theta)} f(x) = \sum f(x) p(x|\theta)$$

となる。ここで$E(\cdot)$は期待値を表す。ボルツマンマシンでは，(5)式が 0 の値の時，尤度関数を最大とする極大値をとる。

2.3 制約ボルツマンマシン

図 2 に隠れ層を持つ制約ボルツマンマシンを示す。制約ボルツマンマシンでは，**図 1** の状態が与えられた可視ノードに潜在ノードを加えて構成されており，オートエンコーダと同様に，隠れ層からなる潜在ノードに可視データの特徴を写像することができる。制約ボルツマンマシンのエネルギー関数も (2) 式と同じように次式により与えられる。

$$\Phi(x,h,\theta) = -\sum_i b_i x_i - \sum_j a_j h_j - \sum_{i,j} w_{ij} x_i h_j \tag{6}$$
$$p(x,h|\theta) = \frac{1}{Z(\theta)} \exp[-\Phi(x,h,\theta)] \tag{7}$$

(6), (7)式より，x, θに対する潜在ノードhの確率分布を$p(h|x,\theta)$とすると

$$\begin{aligned} p(h|x,\theta) &= \frac{p(x,h|\theta)}{\sum_h p(x,h|\theta)} \\ &= \frac{\exp\left[h_j\left(a_j + \sum_i w_{ij} x_i\right)\right]}{1 + \exp\left(a_j + \sum_i w_{ij} x_i\right)} \end{aligned} \tag{8}$$

で表され，$h = 1$の確率分布は

$$\begin{aligned} p(h=1|x,\theta) &= \frac{\exp\left(a_j + \sum_i w_{ij} x_i\right)}{1 + \exp\left(a_j + \sum_i w_{ij} x_i\right)} \\ &= \text{sigmoid}\left(a_j + \sum_i w_{ij} x_i\right) \end{aligned} \tag{9}$$

となる。同様に$x = 1$の場合における可視ノードの確率分布は以下の式で与えられる。

$$p(x=1|h,\theta) = \text{sigmoid}\left(b_i + \sum_j w_{ij} h_j\right) \tag{10}$$

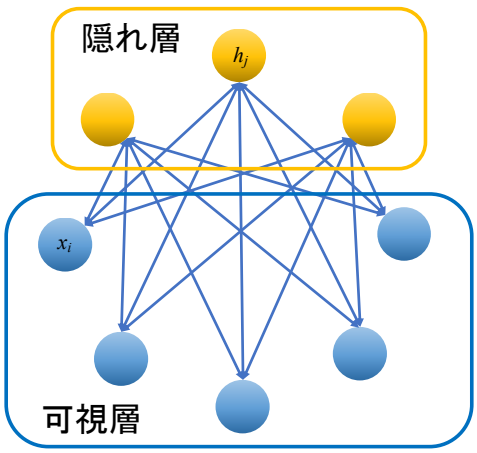

図 2 隠れ層を持つボルツマンマシン

従って,2値の値からなる制約ボルツマンマシンの各ノードが1の値を取る確率は（9），（10）式により与えられる。この時，パラメータθの尤度関数$L(\theta)$は(3)，(4)式より

$$\log L(\theta) = \sum_{n=1}^{M} -\Phi(x^n,h,\theta) - \log[Z(\theta)] \qquad (11)$$

と表され，パラメータθに対する勾配は

$$\frac{1}{N}\frac{\log L(\theta)}{\partial w_{ij}} = \frac{1}{N}\sum_{n=1}^{M} x_i^n p\left(h_j^n = 1 \middle| x^n\right) - \sum_{\boldsymbol{x},\boldsymbol{h}} x_i h_j p(\boldsymbol{x},\boldsymbol{h}|\theta) \quad (12)$$

$$\frac{1}{N}\frac{\log L(\theta)}{\partial b_i} = \frac{1}{N}\sum_{n=1}^{M} x_i^n - \sum_{\boldsymbol{x},\boldsymbol{h}} x_i p(\boldsymbol{x},\boldsymbol{h}|\theta) \qquad (13)$$

$$\frac{1}{N}\frac{\log L(\theta)}{\partial a_j} = \frac{1}{N}\sum_{n=1}^{M} p\left(h_j^n = 1 \middle| x^n\right) - \sum_{\boldsymbol{x},\boldsymbol{h}} h_j p(\boldsymbol{x},\boldsymbol{h}|\theta) \quad (14)$$

となる。(12)，(13)，(14)式より制約ボルツマンマシンにおいても，尤度関数を最大とする極大値を求めることができる。

2.4 コントラスティブダイバージェンス

コントラスティブズダイバージェンス(contrastive divergence)を用いた近似計算により(12)，(13)，(14)式の第2項の計算量の削減が可能である。今，**図3**に示すように可視データ$\boldsymbol{x}$を入力$x^{(0)} = \boldsymbol{x}$とし可視ノードから潜在ノードへデータ伝搬させ，さらに潜在ノードから可視ノード伝搬させる場合を考える。まず伝搬した潜在ノードの値は（9）式より確率分布$p^{(0)}\left(h = 1 \middle| x^{(0)},\theta\right)$を求め，分布からランダムサンプリングにより$h^{(0)}$を決定する。次に$h^{(0)}$を可視ノードへと伝搬させ（10）式より確率分布$p^{(1)}\left(x = 1 \middle| h^{(0)},\theta\right)$から$x^{(1)}$の2値をサンプリングにより求める。これを，次のように繰り返し行う。

$$x^{(0)} \to h^{(0)} \to x^{(1)} \to h^{(1)} \cdots x^{(T)} \to h^{(T)} \qquad (15)$$

これにより得られた近似値$x^{(T)}p^{(T)}$を(12)，(13)，(14)式の第2項として利用することが可能である。近似値により尤度関数を最大にするためのパラメータの更新は次式のように整理でき，ノード数が多い場合でも計算が可能となる。

$$\begin{aligned} \nabla w_{ij} &= \varepsilon\left(\langle x_i^{(0)} p_j^{(0)}\rangle - \langle x_i^{(T)} p_j^{(T)}\rangle\right) \\ \nabla b_i &= \varepsilon\left(\langle x_i^{(0)}\rangle - \langle x_i^{(T)}\rangle\right) \\ \nabla a_j &= \varepsilon\left(\langle p_j^{(0)}\rangle - \langle p_j^{(T)}\rangle\right) \end{aligned} \qquad (16)$$

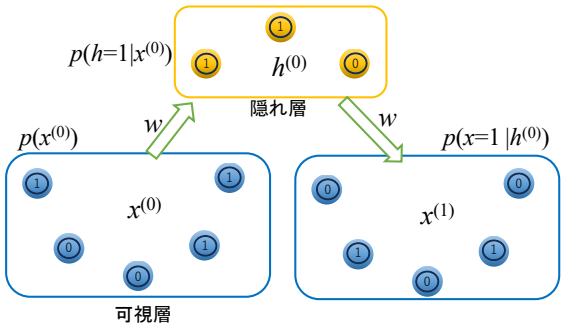

図３　コントラスティブダイバージェンス

ここで$\langle\cdot\rangle$は平均値を表す。εは更新係数を表す。

2.5 ガウシアンベルヌーイ制約ボルツマンマシン

これまでのボルツマンマシンでは，可視ノードの取り得る値は，0，1の2値に限定された。しかし，2値での応用は限られてしまうため，ここでは，可視ノードが連続値となる場合を考える。可視ノードのデータがガウス分布に従うと仮定すると，このボルツマンマシンのエネルギー関数は次式で与えられる。

$$\Phi(x,h,\theta) = \sum_i \frac{(x_i - b_i)^2}{2\sigma_i^2} - \sum_j a_j h_j - \sum_{i,j} w_{ij}\frac{x_i}{\sigma_i} h_j \quad (17)$$

σは可視ノードデータの標準偏差を示す。潜在ノード$\boldsymbol{h}$に対応した可視ノードの確率分布は

$$p(x_i = x|\boldsymbol{h}) = N\left(x|b_i + \sum_j w_{ij} h_j, \sigma^2\right) \qquad (18)$$

で与えられる。$N(\cdot|\mu,\sigma^2)$は平均μ，標準偏差σの正規分布によるサンプリングを表す。可視ノードのデータを標準偏差1に正規化し，コントラスティブダイバージェンスを行った場合，尤度関数を最大にするためのパラメータの更新は2値のボルツマンマシンと同様に(16)式により与えられる。

2.6 エネルギー関数による学習

ボルツマンマシンの計算はそのエネルギーが最大となるパラメータを求めることにある。これまでの計算は，各ノードの値と，それを接続する重みのマトリックス演算により計算可能である。ところが，コントラスティブダイバージェンスはニューラルネットワークの構造となっており，（17）式のエネルギー関数の差を損失関数としたニューラルネットワークによる最適化も可能である。

$$\text{loss} = \Phi^{(0)} - \Phi^{(T)} \qquad (19)$$

また，(17)式は(20)式のように変形出来るため，コントラスティブダイバージェンスの反復時はニューラルネッ

トワークの勾配保持を行わず，$x^{(0)}$と得られた$x^{(T)}$の値のみを用いたエネルギーの損失関数の計算時に，勾配計算を行うことが出来る[6]。

$$\Phi(x,\theta)=\sum_i \frac{(x_i-b_i)^2}{2\sigma_i^2}-\sum_{i,j} softplus\left(w_{ij}\frac{x_i}{\sigma_i}+a_j\right) \quad (20)$$

2.7 カルバックライブラーダイバージェンス関数による学習

2つの離散確率分布の差異を評価する手法としてカルバックライブラーダイバージェンス(Kullback-Leibler divergence)がある。離散確率分布P,Qが与えられた場合，カルバックライブラーダイバージェンスD_{KL}は

$$D_{KL}(P \parallel Q)=\sum_i P(i)\log\frac{P(i)}{Q(i)} \quad (21)$$

により求められる。ボルツマンマシンの学習は可視ノードの実データの確率分布とコントラスティブダイバージェンスにより得られた確率分布を一致させることに他ならない。従って，これらの確率分布のD_{KL}をニューラルネットワークの損失関数と定義する。

$$\text{loss}=D_{KL}\left[p\left(x^{(0)}\right) \parallel p\left(x^{(T)}\right)\right] \quad (22)$$

この方法では，コントラスティブズダイバージェンスの反復時に，ニューラルネットワークの勾配を保持することによって，ボルツマンマシンの最適パラメータが可視ノードの値のみから得ることが可能となる。このことは，潜在のノードに連続値を用いるなどの任意の計算方法に，簡単に拡張できることを意味する。

3. ボルツマンマシンの実装

3.1 学習用1次元データ

MNIST：Modified National Institute of Standards and Technology[7]のデータベースによる手書き数字を用いてボルツマンマシンを学習した。手書き数字は2次元の28×28画素から成るグレースケールの画像データで構成されている。まず，ボルツマンマシンでの可視ノードに対応するためグレースケールの画像データの2値化を行った後，1次元データに展開し728点の学習データとした。フレームワークにはPyTorchを用いて実装を行った。

3.2 2値制約ボルツマンマシン

各種ボルツマンマシンを比較するため，単純なマトリックス演算により計算可能なものについても，できる限り**図4**に示すような同じニューラルネットワークの構造を用いて実装を行った。まず，コントラスティブダイバージェンスの部分については，オートエンコーダに似

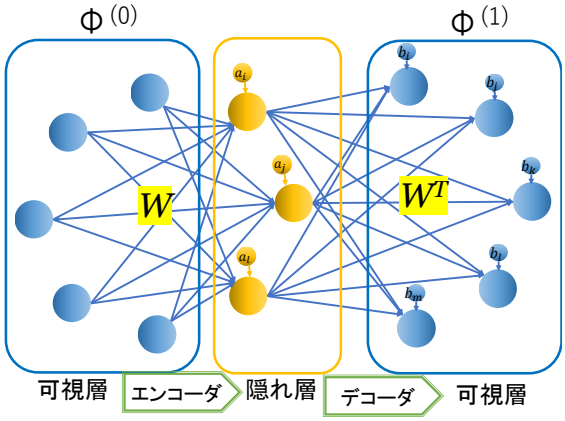

図4　ボルツマンマシンの構造

た構造を用いた。可視ノードから潜在ノードへの伝搬をエンコーダとし，潜在ノードから可視ノードへの伝搬をデコーダとして記述した。両方のネットワークで共通の重みを用いるため，エンコーダの重みの値をデコーダの重みにコピーする構造とした。

```
def encoder(self,v):
    p = self.fc_f(v)
    p = torch.sigmoid(p)
    h = torch.bernoulli(p)
    return h, p

def decoder(self,h,flag):

    self.fc_b.weight.data = self.fc_f.weight.T
    v = self.fc_b(h)
    v = torch.sigmoid(v)
    if flag == 1:
        v = torch.bernoulli(v)
    return v
```

各層のバイアス数は可視ノード数，潜在ノード数のそれに対応するように一致させた。

```
self.fc_f = nn.Linear(features, hsize,bias=True)
self.fc_b = nn.Linear(hsize, features,bias=True)

self.fc_f.bias.data=torch.zeros(hsize)
self.fc_b.bias.data=torch.zeros(features)
```

このボルツマンマシンでは，ニューラルネットワークにより構成されたコントラスティブダイバージェンスに2値化済みのMNISTのデータを与えることより，(16)式の更新データが得られる。得られた更新値から学習係数epsに従って徐々に学習が行われる。このモデルでは，ニューラルネットワークの勾配計算による学習や最適化はなどの手法は用いない。

```
for i in range(EPOCHS2):
    with torch.no_grad():
        v_model=v
        v, p_model = model(v_model, flag)

loss1 = (p0.T @ v0       - p_model.T @ v_model) / batch_size
loss2 = (v0.sum(axis=0) - v_model.sum(axis=0)) / batch_size
loss3 = (p0.sum(axis=0) - p_model.sum(axis=0)) / batch_size

model.fc_f.weight.data += eps * loss1
model.fc_b.bias.data   += eps * loss2
model.fc_f.bias.data   += eps * loss3
```

3.3 ガウシアンベルヌーイ制約ボルツマンマシン

2値制約ボルツマンマシンからガウシアンベルヌーイ制約ボルツマンマシンへの拡張は比較的簡単に可能である。ガウシアンベルヌーイ制約ボルツマンマシンでは，2値制約ボルツマンマシンのデコーダ部分のシグモイド関数とベルヌーイ関数をガウシアン関数(Normal(μ, σ))からのランダムサンプリングに置き換えれば良い。

```
def decoder(self,h,flag):

    self.fc_b.weight.data = self.fc_f.weight.T
    v = self.fc_b(h)
    p_v = Normal(v, self.div)
    if flag == 1:
        v=p_v.sample()
    return v,p_v.loc
```

可視ノードへの入力である MNIST はグレースケール値をそのまま利用可能であるが，(16) 式を用いるには，あらかじめそれぞれの画像の同じ位置の画素値を標準偏差1に規格化する必要がある。また，規格化によりガウシアン関数の標準偏差には $\sigma = 1$ を定数として設定する。

以上により可視ノードが連続値のデータに対する学習についても，(16)式の更新データから学習係数 eps に従った更新により可能となる。

3.4 エネルギー関数による学習

(20) 式より，可視ノードの値が与えられた場合のガウシアンベルヌーイ制約ボルツマンマシンのエネルギーは以下のように計算でき，可視ノードから値をニューラルネットワークの重みに一度伝搬させるのみで勾配保持・計算を行う。従って，潜在ノードの値は学習に用いない。

```
def energy(self,v):
    en_v=torch.sum(0.5*((v-self.fc_b.bias)/stddiv)**2)
    en_h=torch.sum(F.softplus(self.fc_f(v/stddiv)))
    eng= en_v -en_h
    return eng/batch_size
```

学習は，コントラスティブズダイバージェンス処理前後の可視ノードから求めたエネルギー値の差から損失計算が行われ，ニューラルネットワークによる重みとバイアス値の最適化が行われる。

```
    for i in range(EPOCHS2):
        with torch.no_grad():
            v_model=v
            v, _ = model(v_model, flag=1)

    Eng0= model.energy(v0)
    Eng = model.energy(v_model)
    loss =Eng0-Eng

    loss.backward()
    optimizer.step()
```

3.5 カルバックライブラーダイバージェンス関数による学習

カルバックライブラーダイバージェンスを使ったボルツマンマシンでは，可視ノードの確率分布のみを使い学習を行う。学習には損失関数にカルバックライブラーダイバージェンス関数を用い，コントラスティブダイバージェンス処理前後の可視ノードの確率分布を損失関数に入力する。

```
for i in range(EPOCHS2):
    v,p_v = model(v, flag)

loss =torch.distributions.kl_divergence(q_v, p_v).sum(dim=1).mean()
loss.backward()
optimizer.step()
```

従って，これまでのボルツマンマシンとは異なり，学習は，コントラスティブダイバージェンスを実行するニューラルネットワークの伝搬時に勾配を計算し損失関数に応じた最適化を行う必要がある。ここで，ニューラルネットワークにはガウシアン分布関数によるサンプリング時に勾配計算が途切れないよう rsample( )を用いた。

```
def decoder(self,h,flag):
    self.fc_b.weight.data = self.fc_f.weight.T
    v = self.fc_b(h)
    p_v = Normal(v, stddiv)
    if flag == 1:
        v=p_v.rsample()
    return v,p_v
```

また，可視ノードの初期データによる確率分布については，事前計算を行い，損失関数に入力する。

```
    stddiv, _ =calc_std_mean(v)
    q_v = Normal(v, stddiv)
```

カルバックライブラーダイバージェンスによる学習では，可視ノードの確率分布のみから学習が行われるため，エンコーダ部分についてもガウスサンプリングを用いた連続値に変更した場合でも追加の計算は不要であり，これまでと同様の計算が可能である。また，任意の非線形関数を簡単に用いることもできる。

```
def encoder(self,v,flag):
    h = self.fc_f(v)
    p_v = Normal(h, self.div)
    if flag == 1:
        h=p_v.rsample()
    return h
```

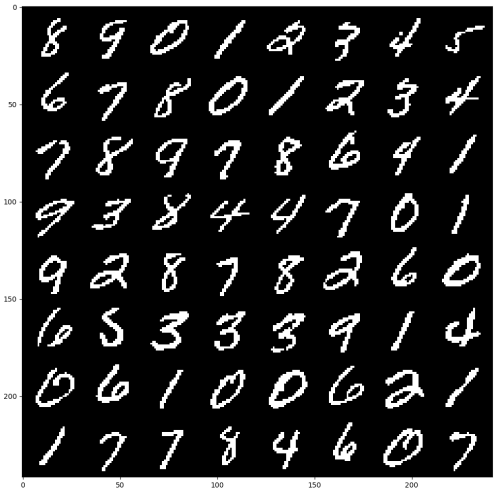

図 5 未学習の入力データ
（バイナリーデータ）

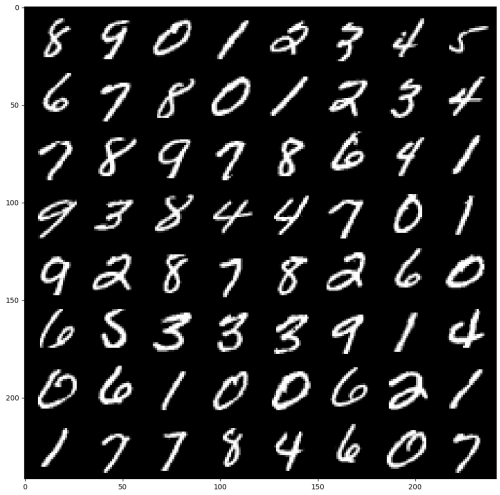

図 7 未学習の入力データ
（グレースケール）

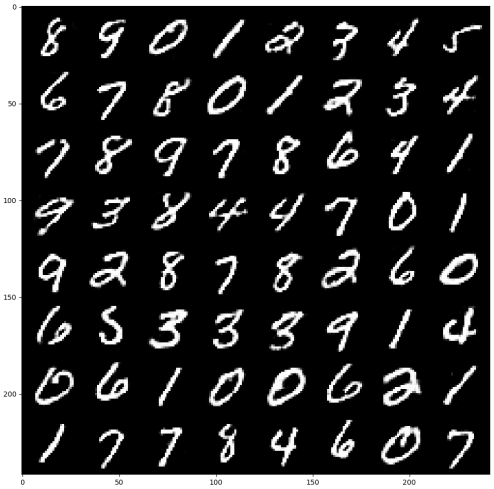

図 6 未学習データからの生成データ
2 値制約ボルツマンマシン

図 8 未学習データからの生成データ
コントラスティブダイバージェンスモデル

4. ボルツマンマシンの能力評価

4.1 学習条件

ボルツマンマシンに6万種類の手書き文字を学習させ，その生成能力について比較検討を行った。ボルツマンマシンの潜在ノードは 2 値の値のみ取り得ることが可能であるため，オートエンコーダと比較すると各ノードの情報量が圧倒的に少なく，入力データの特徴値を極端に低次元化した潜在ノードへ写像することは出来ない。そこで各モデルの生成能力を比較するため，潜在ノード数を 500 個に固定し，評価を行った。

4.2 2 値制約ボルツマンマシンの能力

まず,2値制約ボルツマンマシンにバイナリー化後の手書き文字を学習させた。この時の学習係数 ε は 0.01 とし 400 回の学習を行った。学習終了後，**図 5** に示す未学習のバイナリーデータを与えた場合の,2値制約ボルツマンからの可視ノードの出力を**図 6** に示す（**図 6** はグレースケール）。入力データと比較すると僅かな画像の劣化が確認出来る。但し，各画像についての手書き文字が持つ特異な部分の修正は全く行われていない。

4.3 ガウシアンベルヌーイ制約ボルツマンマシンの生成能力

4.3.1 コントラスティブダイバージェンスによる学習モデル

次に，応用範囲を拡大するため，グレースケールのデータを可視ノードに与える。学習の前処理として，ノード毎に学習データを平均 0，標準偏差 1 に規格化した。処理後，グレースケールの手書き 6 万文字をコントラスティブダイバージェンスに伝搬させ，出力値から（16）式により直接パラーメータの更新を行った。この時，学習係数 ε は 0.01 とした。また，学習に必要な更新回数は過

図 9 未学習データからの生成データ
エネルギー損失関数モデル

学習に陥る直前の 200 回とした。未学習のグレースケール入力画像**図 7** から得られた画像を**図 8** に示す。2 値制約ボルツマンと比較して，ガウシアンベルヌーイ制約ボルツマンマシンでは，画像の劣化が著しい。特異な部分の修正が僅かに行われているのか劣化によるものなのか判断は出来ない。

4.3.2 エネルギー損失関数に基づく学習モデル

次に，ガウシアンベルヌーイ制約ボルツマンマシンの学習を（17）式のエネルギー関数を損失関数に用いて計算を行った。この計算では，最適関数に Adam(…) を用いた。

```
optimizer = optim.Adam(model.parameters(), lr=0.0001, betas=[0.9, 0.999])
```

このモデルでは複数の学習率 lr について学習を行った結果，標準的な値より小さな値（lr=0.0001）の設定が望ましいと明らかとなった。学習回数は 100 回～200 回程度で十分であった。**図 9** に得られた画像を示す。エネルギー関数とニューラルネットワークによる最適化を用いたガウシアンベルヌーイ制約ボルツマンマシンでは，赤丸で囲んだ特異な部分の修正が行われている。また（16）式により直接パラーメータの更新を行ったものと比較して，画像の劣化も少ない結果となった。

4.3.3 カルバックライブラーダイバージェンスに基づく学習モデル

同様にガウシアンベルヌーイ制約ボルツマンマシンの学習に（22）式のカルバックライブラーダイバージェンスを用いて，可視ノードの確率分布から最尤度パラメータの学習を行った。この時，更新パラメータの勾配計算はカルバックライブラーダイバージェンス時のニューラルネットワークで行う。最適関数は Adam(…) を用いた。ガウシアンベルヌーイ制約ボルツマンマシンにカルバックライブラーダイバージェンスを用いて得られた結果を**図 10** に示す。エネルギー損失関数を用いたモデルと比較して画像の劣化が著しい。

図 10 未学習データからの生成データ
カルバックライブラーダイバージェンスモデル

図 11 未学習データからの生成データ
ガウシアン制約ボルツマンマシン

4.4 ガウシアン制約ボルツマンマシンの生成能力

最後に潜在ノードについてもガウス分布の確率に従う可視ノードと潜在ノードの両方が連続値を取るモデルについて評価した。この実験では，潜在ノードのガウス分布の標準偏差を σ =1 に固定している。このモデルでは，潜在ノードが連続値を扱え，情報量が増加した効果によるものか，可視ノードに連続値を入力するモデルの中でもっとも入力データに近い出力が得られた。加えて，丸印の部分に見られるような画像ごとの特異部分についても修正が行われている（**図 11**）。

5. 結　　言

ボルツマンマシンについて, 複数の実装モデルを構築し, 次の結果結果を得た。

1. 2値制約ボルツマンマシンでは, バイナリーデータの学習生成能力は高い。但し，特異な部分の修正は行われない。
2. 2 値制約ボルツマンマシンの可視ノードをガウス確率分布に変更したモデルでは, 生成能力は低い。
3. エネルギー関数を損失関数に用いたモデルでは, 学習生成能力は高く，特異な部分の修正が行われる。
4. カルバックライブラーダイバージェンスを損失関数に用いたガウシアンベルヌーイボルツマンマシンの生成能力はエネルギー損失関数モデルほど高くはなかった。
5. 潜在ノードにガウス確率分布を使ったモデルでは, 生成能力も高く，特異な部分の修正も行われた。

本報告では，可視層と潜在層からなる２層構造のボルツマンマシンについて評価を行った。特に各種モデル実装のポイントに触れ，その生成能力と修正能力を明らかにした。但し，異常検知などに用いるためには，評価に用いた２層構造では，次元数の圧縮が十分では無く，ボルツマンマシンを複数重ね合わせた多層構造にするなどの汎化能力の向上が必要である。また，カルバックライブラーダイバージェンスを損失関数に用いたボルツマンマシンは，学習パラメータの最適化が簡単に行えるため, 応用範囲の拡大や能力向上などへの貢献が期待できる。

文　　献

1) 岡谷貴之：深層学習，講談社（2015）
2) 麻生 英樹, 安田 宗樹, 前田 新一, 岡野原 大輔, 岡谷 貴之, 久保 陽太郎, ボレガラ ダヌシカ：深層学習，近代科学社 (2015).
3) 廣川 勝久：広島県立総合技術研究所東部工業技術センター研究報告，深層学習による異常検知手法の簡単な比較（第 1 報）35（2022）.
4) 廣川 勝久：広島県立総合技術研究所東部工業技術センター研究報告，深層学習による異常検知手法の簡単な比較（第 2 報）36（2023）.
5) 廣川 勝久：広島県立総合技術研究所東部工業技術センター研究報告，深層学習による異常検知手法の簡単な比較（第 3 報）36（2023）.
6) Renjie Liao, Simon Kornblith, Mengye Ren, David J. Fleet, Geoffrey Hinton: Gaussian-Bernoulli RBMs Without Tears, arXiv:2210.10318（2022）.
7) THE MNIST DATABASE of handwritten digits：http://yann.lecun.com/exdb/mnist/