Vol.2013-CVIM-185 No.19
2013/1/23

# ディープラーニング

岡谷 貴之[1,a)]　齋藤 真樹[1,b)]

**概要**：ディープラーニングは，多層のニューラルネットワーク（以下 NN）を用いる機械学習の方法論である．最近，これに基づく方法群が，画像認識のベンチマークテストで従来の記録を次々に塗り替えた他，音声認識やその他の学習・認識の問題に応用されて同様に高い性能を示すに至り，幅広い関心を集めつつある．本稿では，このディープラーニングについて，画像認識への応用を念頭に，現時点で知られている様々な方法をなるべく網羅的に説明する．具体的には，NN の基本構成から，Convolutional NN，プーリング，局所コントラスト正規化，教師なし学習であるプレトレーニング，オートエンコーダ，スパースな特徴表現を可能にする複数の方法，Restricted Boltzmann Machine や Deep Belief Network などの生成モデルに基づくディープラーニングの各手法を，それぞれ説明する．またディープラーニングのためのソフトウェアライブラリにも触れる．

## 1. はじめに

### 1.1 ディープラーニングの隆盛

ディープラーニングとは，多層のニューラルネットワーク（以下 NN）を用いた機械学習の方法論である．最近，画像認識や音声認識の分野で，ディープラーニングの方法が従来の方法を上回る高い性能を見せることが分かり，大きな関心を集めている．

その関心の高さは研究者の間にとどまらず，例えば 2012 年 6 月開催の International Conference on Machine Learning（ICML）で発表された Le らの研究 [32] は，一般各紙にも取り上げられた[*1]．この研究は，YouTube からランダムに切り出した大量の画像を使い，大規模な NN を教師なし学習で訓練した結果，霊長類の脳に見られるのと同様の人や猫の顔などに選択的に反応するニューロンが，自動的に生成されたというものである．

画像認識の研究者にとっては，2012 年 11 月開催の一般物体認識のコンテスト Imagenet Large Scale Visual Recognition Challenge（ILSVRC）[*2] で，ディープラーニングの方法が，従来の方法に大きな差をつけて優勝したことは大きなインパクトがあった[*3]．従来の方法とは，画像から SIFT に代表される局所特徴を多数取り出し，これを Bag-of-features や Fisher ベクトルなどの方法で 1 枚の画像全体を表す特徴に変換するものである．この方法論に基づく参加チームの認識性能はほぼ同じであったのに対し，優勝チームのそれは群を抜いていた．このことは，従来方法の限界を明確に示すと同時に，ディープラーニングの大きな可能性を象徴的に示している．

ディープラーニングを用いた（画像）認識の方法の特徴は，上述のような手動設計した特徴を用いないということである．多層 NN の教師あり，あるいは教師なし学習によって，認識に有効な特徴を自動的に発見できるというのが，上述のような高い性能の秘密であり，また研究者からの期待が高まっている理由でもある．

静止画像からの物体認識だけでなく，動画像を用いた認識へのディープラーニングの応用も既に始まっている [30], [59]．音声認識では，既に画像認識以上の成功を収めている（最新の評論が [20] にある）他，自然言語処理への応用も最近活発化している．その他にも，新薬開発のための化合物活性予測のコンテスト[*4]で優勝するなど，性能だけでなく応用先も広がりを見せ始めている．少なくとも現状を見る限り，近い将来，コンピュータを用いた認識のあらゆる問題に対し，ディープラーニング以外の方法ではトップレベルの性能を達成できない可能性すら想像される．

### 1.2 現在に至る研究の経緯

（人工）ニューラルネットワーク自体の研究はかなり以

[1] 東北大学
Tohoku University
[a)] okatani@fractal.is.tohoku.ac.jp
[b)] msaito@fractal.is.tohoku.ac.jp

[*1] 例えば New York Times, 2012 年 6 月 25 日付記事, “How Many Computers to Identify a Cat? 16,000.”
[*2] http://www.image-net.org/challenges/LSVRC/2012/
[*3] ディープラーニングの方法で，画像認識のベンチマークテストで過去の記録を大幅に塗り替えたという報告はそれ以前にもあった [9], [10].
[*4] https://www.kaggle.com/c/MerckActivity

前からあり，1940 年代にさかのぼる [17], [42], [49]．それ以来，現在までに何度か研究のブームとその終焉を繰り返してきた．そのうち最近のものに，80 年代半ば以降の，NN の学習方法であるバックプロパゲーションの「発見」[50] を契機にするブームがある．このブームは 90 年代後半ころには終焉し，最近に至るまで NN の研究は冬の時代を経験する*5.

この 80 年代半ばのブームが終焉した背景には，3 つほどの理由を挙げることができる．第一に，バックプロパゲーションによる NN の学習は，3 層（隠れ層 1 層）程度の NN ではうまく行くが，それ以上の多層の NN ではあまりうまくいかない*6ことである．第二に，層の数や隠れ層のユニット数，その他学習時のパラメータが，最終的な性能とどのように結びつくかの議論が，他の機械学習の方法と比べて難しいことである．したがって，これらをどのように選ぶかの明確な指針がなく，高い性能を引き出すための知識はノウハウの領域だと見なされた [58]．第三に，90 年代当時の計算機の処理能力が，現実的な問題を扱い得る十分な規模のネットワークを扱えなかったことである [10].

ただし同時に，最初に挙げた理由（多層の NN の学習困難さ）には，当時ひとつの例外があったことは指摘する必要がある．確かに層間のユニットを全結合した多層 NN の学習は困難だったが*7，入力（2 次元画像）にフィルタを畳込む計算を各層で実行する Convolutional Neural Network（以降 CNN）は，80 年代後半に 5 層（隠れ層 3 層）の学習に成功していた [33]．CNN は，Fukushima らの Neocognitron[15] にルーツを持ち，Lecun らが完成させたもので，手書き文字認識において高い性能を達成した [35]．この成果にもかかわらず，恐らく上述の他の理由もあって，他の NN 同様，CNN は最近まであまり注目されなかった．

このような NN に対する低い関心を覆す契機になったのが，Hinton らの Deep Belief Network（DBN）の研究 [13] である．DBN は，その振る舞いが確率的に記述される点で通常の NN とは異なるモデルだが，ネットワークの構造自体は多層 NN とほぼ同じであり，しかもその学習が困難であった点も同じであった．この研究で Hinton らは，DBN の多層構造の中で部分的な層ごとに教師なし学習を実行し，これを層の数だけ繰り返すプレトレーニング（pretraining）という方法を行えば，学習がうまく行えることを示した．そうして得たパラメータを初期値に，全体のネットワークのパラメータの微調整（fine tune）を教師ありあるいは教師なし学習のいずれかで行えば，良い結果を得ることができた．

この直後には，このプレトレーニング＋微調整という方法論が，オートエンコーダ（autoencoder）でも同じように使えることが分かる [3]．オートエンコーダは，NN への入力に対し，順伝播と（仮想的な）逆伝播を順に行ってこれを再現したとき，その再現誤差をなるべく小さくするように学習を行う方法である．Hinton らの DBN 同様，2 層ずつ分離した NN それぞれを下から順番に，オートエンコーダの方法で教師なし学習を行い，その後に全体の NN を例えば誤差逆伝播法によって教師あり学習を行う．以上の結果を要約すると，多層 NN はランダムな初期化から開始すると学習は困難だが，教師なし学習を層ごとに行って得たものを初期値にすればうまく行くということである．

以上の方法論による多層 NN の学習方法は，以前からよく対象とされた文字認識を始め，画像認識，音声認識，自然言語処理の各問題に次々に適用されてきた．画像認識，特に一般物体認識への適用を眺めてみると，2010 年ころまでの数年間は，少なくとも Caltech-101 や NORB に代表されるメジャーなデータセットを使ったベンチマークテストで，非 NN の既存手法を上回るような性能は達成できていなかった．それよりもむしろ，多層の NN によっていかなる画像特徴が学習されるのかということへの関心が高かった．代表的な例が，スタンフォード大の Ng らのグループによる，自然画像を使った NN の教師なし学習により，霊長類の視覚野の V2 領域に見られる特徴 [25] が得られたという報告 [37] や，Caltech-101 を対象に多層 NN を教師なしで学習することで，入力層からの層の高さに対応した階層性を持つ特徴が得られたという報告 [38] である．これらは冒頭に述べた Le らの研究 [32] につながる．

2010 年以降になると，ベンチマークテストでもディープラーニングの方法は次々に過去の記録を書き換えることになる．最新の状況は冒頭に述べた通りである．なおベンチマークテストで良い成績を残している方法の多くが，上述のプレトレーニングを使う方法よりもむしろ，CNN を使ってプレトレーニングを行わず，最初から教師あり学習を行うものであることは，興味深い．

このように近年急に，多層 NN が非 NN の既存手法を上回る性能を達成できるようになった理由はいくつか考えられる．まずひとつには，CNN も含め，一般的な NN の研究ではメジャーな存在とは言えなかったプーリングやコントラスト正規化といった処理が，画像認識にとって重要な構成要素であると認識されたことである [6], [7], [26], [52]．少し長い期間で見れば，GPU を始めとする計算機の処理能力の向上も要因に挙げられる．また以前は多層 NN に対する期待値が低く，試してみる研究者が少なかったというのが案外真相かもしれない．

*5 2000 年の NIPS では，タイトルにニューラルネットワークという言葉を含む投稿は採択率と負の相関があり，逆に SVM やベイジアンネットワークなどをタイトルに含む投稿は正の相関があったという話がある [58].
*6 学習がうまくできないとは，汎化能力が期待を下回るような学習しかできないという意味である．
*7 NN のパラメータをランダムに初期化した状態から学習を始めた場合を指す．

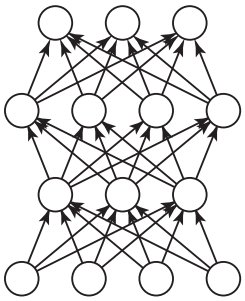

**図 1** 順伝播型の多層ニューラルネットワーク.

### 1.3 本稿の構成

本稿では，ディープラーニングの方法論を構成する各要素について，なるべく網羅的に，かつ簡潔に説明することを試みた．第 2 節では，ネットワークの構成要素，すなわち多層 NN の基本形，Convolutional Neural Network，プーリング，局所コントラスト正規化についてそれぞれ述べる．第 3 節では，多層 NN の教師なし学習の方法を述べる．具体的には，多層 NN のプレトレーニングの方法，そのための部品として機能するオートエンコーダ，そして高い認識性能のためのエッセンスとも考えられる特徴のスパースな表現について，それぞれ述べる．第 4 節では，ボルツマンマシンを主体とする生成モデルをまとめて説明する．第 5 節では，ディープラーニングのためのいくつかのライブラリを短く紹介する．

## 2. ネットワークのアーキテクチャ

### 2.1 多層ニューラルネットワーク

#### 2.1.1 基本構造

ディープラーニングが対象とするのは，図 1 のような有効グラフ構造を持つ多層のニューラルネットワーク（NN）である[*8]．入力が各層を順番に伝搬し出力に至り，同じ層の間では結合はないものとする．隣接層のユニット間の結合は，図のようにすべてが結合する場合を考えることもあれば，後述の CNN のように部分的にしか結合しないものを考えることもある．

最も一般的な隣接層間の計算を，図の最下層から第 1 層への計算を例に示す．入力を $\mathbf{x} = \mathbf{h}^0$ とする．図の有向グラフ構造に従い，第 $k$ 層の各ユニット $\mathbf{h}^k = [h_1^k, \ldots, h_i^k, \ldots]$ への入力は，第 $k-1$ 層のユニットへの入力 $\mathbf{h}^{k-1}$ を元に

$$h_i^k = f(b_i^k + \mathbf{w}_i^{k\top}\mathbf{h}^{k-1}) \tag{1}$$

のように計算される．以降これを簡単に

$$\mathbf{h}^k = f(\mathbf{b}^k + \mathbf{W}^k\mathbf{h}^{k-1})$$

と書く（$\mathbf{W} = [\mathbf{w}_1, \ldots, ]^\top$）．通常，関数 $f$ には何らかの非線形性を持つものが用いられる．

[*8] 多層パーセプトロン（multi-layered perceptron）と呼ぶ場合もある．

この関数 $f$ にはシグモイド関数を用いるのが一般的で，例えば

$$f(x) = \tanh(x) \tag{2}$$

（$f(x) \in [-1:1]$）やロジスティック（シグモイド）関数

$$f(x) = \frac{1}{1+\exp(-x)} \tag{3}$$

（$f(x) \in [0:1]$）がある．なお ILSVRC2012 で優勝したグループの NN では，シグモイド関数よりも学習スピードが優れるとして

$$f(x) = \max(0, x) \tag{4}$$

を選んでいる [28].

NN は，入力 $\mathbf{x}$ から最上位層の状態（あるいは出力）$\mathbf{h}(k)$ に至るひとつの関数を実現する．各層間の結合の重み $\mathbf{W}^k$ およびバイアス $\mathbf{b}^k$ がパラメータとなり，これらは多数の訓練データを用いた学習により決定する．学習の目的は，（訓練データがその一部を表現する）未知の関数を NN に再現させることにある．なお後述するプーリングやコントラスト正規化のように，$\mathbf{W}^k$（およびバイアス $\mathbf{b}^k$）を固定とし，学習の対象としない層や，あるいはそもそも式 (1) では表現できない写像 $\mathbf{h}^{k-1} \to \mathbf{h}^k$ を計算する層を，普通の層の上に積み上げる場合もある．

多層 NN の教師あり学習は，確率的勾配降下法（Stochastic Gradient Descent, SGD）によって行うことが多い [5], [34]. 誤差逆伝播（Back Propagation）法とも呼ばれる．最近の学習最適化の方法については [29] を参照されたい.

最近，学習の方法として dropout と呼ばれるものが提案され [21]，本稿冒頭に述べた ILSVRC2012 で優勝したシステム [28] で使用されている．これは，学習時にサンプルごとにランダムに隠れ層の 50%程度を使わないようにしてしまうというもので，これによって NN の高すぎる自由度を故意に制約し，過学習を防ごうとするものである．操作のランダム性は，複数の異なるモデルを使って学習し，それらモデルの平均を使うのと同様の効果があるとされる.

#### 2.1.2 多層 NN の学習困難さ

層間のユニットがすべて結合する NN の学習を，パラメータをランダムに初期化した状態から開始する場合を考える．浅い（隠れ層の数が 1 かせいぜい 2 個）NN の学習はさほど難しくない一方で，それ以上の深い NN の学習は困難であることが良く知られている [2]．ここで学習が難しいとは，訓練誤差は順当に小さくなるが，汎化誤差が大きくなってしまうという意味で，過学習とも表現できる.

では，なぜ多層 NN の学習は難しいのか？実験的に観察される現象は，下位層ほど学習がうまくできない傾向である [2]．その一つの解釈は，上位層でユニット数が十分あって自由度がたっぷりあれば，そこでのパラメータ調整のみ

で，与えられた学習サンプルをうまく表現できてしまうことによるというものである．また，学習時の誤差逆伝播の計算について考えると，出力層での誤差勾配が，最初の何層かはよいが層を経るにしたがって「分散」し，深い下位層にまで伝わらないからであるとも言われる．これらの解釈は，各層間の結合が局在的で疎な CNN であれば，構造が多層であっても学習できることと整合性がある．CNN は次節で詳しく述べる．

### 2.2 Convolutional Neural Network

#### 2.2.1 歴史

畳込み（convolutional）ニューラルネットワーク（以下 CNN）とは，特定の構造を持つ多層 NN で，Hubel と Wiesel の研究 [22] にルーツがある．この研究で Hubel らは，ネコの初期視覚野 (V1) に，特定の傾きを持つ線分に選択的に反応する単純細胞（simple cells）と，同様の選択性を持ちつつ一定の並進移動不変性（線分と直交する方向に提示位置をずらしても同様に反応する）を持つ複雑細胞（complex cells）があることを発見した．Fukushima らは，この数理的モデルとなる Neocognitron を発表している [15]．Neocognitron は，ある層において同一の結合重みをもつユニットを複数並べ，その出力をさらに上位層で集積する（後述するプーリング）ことで，入力パタンの並進移動に対する不変性を実現している．後に LeCun らは，ほぼ同じ構造を持つ NN を CNN と呼び，その学習に誤差逆伝播法（勾配法）を用いることで，いくつかのパタン認識の問題で当時最高の性能を達成した [33]．特に手書き文字認識を対象とした CNN の LeNet-5[35] は，その高い性能から，同タスクに対するベンチマークとしての役割を長期にわたって果たした．

#### 2.2.2 基本構造

画素数 $n_x \times n_x$ の画像 $\mathbf{x}$ に対する，画素数 $n_w \times n_w$ のフィルタ $\mathbf{w}$ の畳込みを考える．この畳込みを $\mathbf{h} = \mathbf{x} * \mathbf{w}$ と書くと，一般に出力 $\mathbf{h}$ は $n_h \equiv n_x - n_w + 1$ のサイズの画像になる．CNN は通常，このようなフィルタを複数個 $\mathbf{w}^1, \ldots, \mathbf{w}^L$ 持ち，フィルタごとに異なる出力のセット（マップと呼ぶことが多い）$\mathbf{h}^1, \ldots, \mathbf{h}^L$ を持つ．

CNN は，この $\mathbf{x}$ から $\mathbf{h}$ への計算に対応する配線を基本構造として持つ．具体的には図 2 のように，入力画像の画素と 1 対 1 でユニットが対応する入力層に対し，その $n_w \times n_w$ の部分集合と第 1 層のユニットが対応付けられ，両者が結合される．その結合重みは，畳込むフィルタ $\mathbf{w}$ そのものである．このように CNN は，層間の結合が位置ごとに局所的になされ[*9]，その結合重みが（上位層の全ユニット間で）共有されている（=フィルタの係数 $\mathbf{w}^l$ がマップごとに同じ）ことを特徴とする．

[*9] 上位層のユニットの「受容野 (receptive field)」が局所的であるとも表現される

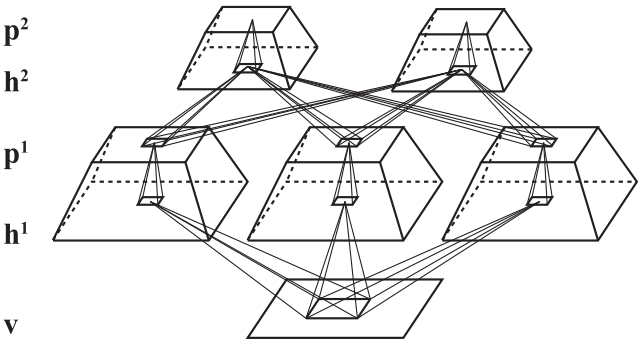


**図 2** 畳込み NN（Convolutional Neural Network，CNN）．

これに加えて CNN では，その次の層でプーリング（あるいはサブサンプリング）と呼ばれる処理が行われる．LeNet-5[35] では，フィルタの出力層の隣接 $2 \times 2$ ブロックの平均値（を定数倍しバイアスを加算した後シグモイド関数を通したもの）が次の層の 1 ユニットの状態となる．この $2 \times 2$ ブロックはフィルタ出力層で重複せず，したがってプーリング層のサイズはフィルタ出力層のサイズ（$n_h \times h_h$）の半分，すなわち $n_h/2 \times n_h/2$ となる．この処理によって，フィルタ出力層の隣接ブロック内ユニットのどれがアクティブとなっても，そのブロックを集積したプーリング層のユニットがアクティブとなるため，冒頭に述べた微小な並進移動に対する不変性が実現される．畳込み層が単純細胞に，プーリング層が複雑細胞にそれぞれ対応する．なおプーリングのやり方は，ここで述べた平均値を求める方法の他にもいくつかあり，それぞれ異なる性質を持つ．これについては 2.3 節で述べる．

CNN はこのフィルタの畳み込みとプーリングをこの順で何度か繰り返す構造をとる．つまりフィルタ出力層，プーリング層の 2 層を，プーリング層を次の層の入力層とする形で積み重ねる．通常，プーリングはユニット数を減少させるので，上にいくほど層のユニット数が小さくなる．最終層には，タスクに応じた数のユニットを持つ出力層を置き[*10]，そこに至る何層かはプーリングを挟まない全結合とすることが多い [9], [10], [35], [58]．CNN は教師あり学習を前提としており，SGD を用いた誤差逆伝播により，パラメータを修正する．

#### 2.2.3 多層 CNN の学習が可能な理由

なお一般の多層 NN と異なり，CNN はプレトレーニングを行わず，ランダムな初期値から学習を開始しても十分な性能を発揮することが知られている [2], [10], [35], [58]．Bengio[2] によれば，これには 2 つの解釈が考えられる．1 つは CNN ではネットワークの結合が疎であるため，仮に多層になっても逆伝播する勾配が拡散せず，上位層から下位層に伝わる過程で無意味なものとならずに済むという理解である．（ただし，疎なだけではだめで，その位置依存の分布，すなわち上位層のユニットはその直下にある下位層

[*10] 問題が $N$ クラス分類であれば $N$ 個のユニットを持つ．

のユニットの集合と結合していることが必要だとも述べている.）もう一つの説明は，CNN の配線の仕方が，特に視覚・パタン認識のタスクに向いているというものである．これは CNN のフィルタをランダムとしても，抽出される特徴に基づく認識を行うと一定の性能を示したと言う結果とも整合する [26], [47], [52].

### 2.3 プーリング（pooling）

#### 2.3.1 目的

プーリングとは，最も一般的には抽出した特徴から認識に余分な情報を捨て，認識に必要な情報を保った新たな表現に変換することである [7]．例えば，パタンの幾何学的変動（位置や回転）を始め，照明条件，背景のクラッタ等に対して不変な特徴を取り出すのが目的である．

プーリングは手動設計した特徴抽出でも不可欠な要素である．局所特徴量の SIFT[40] を例にとれば，特徴記述子（デスクリプタ）は，ある小領域内で最も強い濃淡勾配方向を選び，この小領域をいくつかまとめた領域内で各小領域ごとの勾配方向をまとめてヒストグラムを作った後，さらに大きな領域内でそれらをつなげて出来ている．ひとつめの濃淡勾配の選択は後述するマックスプーリングにあたり，2 つ目のヒストグラムの生成は平均プーリング（あるいは sum pooling）にあたる．非 NN の一般物体認識手法で，この局所特徴の出現頻度をエンコードして画像 1 枚の特徴（mid-level feature とも呼ばれる）を得るのに使われた古典的な BOF（Bag-of-features）表現は，画像全体を対象（受容野）とした平均プーリングである．

これらは共通して，取り出した特徴の位置情報をある程度無視することで，位置変動に対する一定の不変性を実現している．Lezebnik らの Pyramid Matching Kernel（PMK）は，完全に位置の情報を捨てる BOF に対し，画像を部分ブロックに分け，位置の情報をある程度残すようにすることで，認識性能を向上できたと言える．

#### 2.3.2 プーリングの各方法

上述のように CNN では，入力層からフィルタ出力層，プーリング層に至る情報の伝播過程で，同じフィルタが画像の異なる位置 $i$ で返す異なる応答 $h_i$ を，プーリング層のユニット $i$ は直下の小領域 $P_i$ 内のフィルタの応答をまとめて（どこで応答が強いかの情報を捨て）$h'_i$ とし，これにより微小な位置変化に対する不変性を実現していた．フィルタ出力層の小領域 $P_i (i = 1, \ldots)$ を互いに重複を許さずにとれば，プーリング層のユニット数はフィルタ出力層のそれより減少することになる．

プーリングにはいくつものバリエーションがある．**平均プーリング（average pooling）**とは，

$$h'_i = \frac{1}{|P_i|} \sum_{j \in P_i} h_j \tag{5}$$

のように $k-1$ 層での小領域 $P_i$ での応答の平均を $k$ 層のニューロン $i$ の値とする．**マックスプーリング（max pooling）**とは，

$$h'_i = \max_{j \in P_i} h_j \tag{6}$$

のように，$k-1$ 層での領域 $P_i$ 内の応答の最大値を $k$ 層の値とする．

また，平均とマックスプーリングの中間的存在として **Lp プーリング**

$$h'_i = \left( \frac{1}{|P_i|} \sum_{j \in P_i} h_j^P \right)^{\frac{1}{P}} \tag{7}$$

がある（ただし，形式的に両者を連続的につなぐ表現は他にもあり得る）[7]．この他には，平均プーリングで $h_j$ をその絶対値 $|h_j|$ で置き換えた絶対値プーリングもある [26].

#### 2.3.3 プーリング方法の比較

平均プーリングは，Fukushima らの Neocognitron[15], LeCun らの CNN[33], [35]，Pinto らの低次元特徴量で採用されている．一方マックスプーリングは，Serre らの HMAX モデル [57]，Ranzato らの [46] らが初期の採用例と言える．また，局所特徴の出現頻度をエンコードする非 NN の一般物体認識手法を対象に，マックスプーリングがより優れた性能を示すことが示された [7], [64] こともあり，最近の研究では主流である [9], [10], [28]．ただし本稿冒頭に紹介した Le ら [32] の研究では，L2 プーリングが採用されている（後述のように，このプーリングは Topographic ICA[23] にルーツを持つ）．また，情景中の文字認識において，Lp プーリングの $P$ をチューニングして認識率が向上するという報告がある [55].

また，プーリングの違いに焦点を当てた研究はいくつかあり，実験評価に [6], [26], [53] が，理論的比較に [7] がある．

### 2.4 局所コントラスト正規化

#### 2.4.1 処理の詳細

CNN のように，画像に複数のフィルタを畳込んで得られる出力（マップ）を考える．入力画像 $x_{jk}$（$j, k$ は画像座標とする）に，あるフィルタ $(i = 1, \cdots)$ を畳み込んで得られる出力の集合を $h_{ijk}$ と書く．このとき，$i$ 番目のフィルタ出力の画素 $(j, k)$ の出力 $h_{ijk}$ に対し，その近傍における全フィルタ出力 $i$ にわたる平均を求め，これを差し引く処理を**減算正規化（subtractive normalization）**と呼び，すなわち

$$\bar{h}_{ijk} = h_{ijk} - \sum_{ipq} w_{pq} h_{i,j+p,k+q} \tag{8}$$

である．ただし $w_{pq}$ は $\sum_{ipq} w_{pq} = 1$ となる重みである．この操作は，同近傍および全フィルタ出力 $i$ にわたる $\bar{h}_{ijk}$

の平均を 0 にする．画像空間での近傍サイズおよび重み $w_{pq}$ は，Jarrett らは $(j,k)$ をピークに減衰する（画像平面での）ガウス関数とし [26]，Pinto らは近傍を $3\times3$ 画素の小領域とし $w_{pq}=1$ の単純平均としている [45].

さらに，平均を求めたのと同じ領域における $h_{ijk}$ の標準偏差

$$\sigma_{jk}=\sqrt{\sum_{ipq}w_{pq}\bar{h}^2_{i,j+p,k+q}},\tag{9}$$

を求め，$\bar{h}_{ijk}$ をこれで割る：

$$h'_{ijk}=\bar{h}_{ijk}/\max(c,\sigma_{jk}).\tag{10}$$

この処理は**除算正規化（divisive normalization）**と呼ばれる．分母の max 関数を無視すれば，$h'_{ijk}$ は（画像上での近傍および全フィルタ出力にわたる）平均が 0 で分散が 1 に正規化されたことになる．分散は画像のコントラストに相当するから，コントラスト正規化と呼ばれる．ただし max 関数があるので，定数 $c$ の値に応じてコントラストが小さいときにはこの正規化が行われない．$c$ の値は，[26] では $\sigma_{jk}$ の平均が，[45] では入力画像自体を事前に正規化（グレースケールで平均 0，分散 1 となる）しており，その理由もあって $c=1$ としている．

なお式 (10) の代わりに，次のような計算を考えることもある．

$$h'_{ijk}=\bar{h}_{ijk}/\sqrt{c+\sigma^2_{jk}}.\tag{11}$$

コントラスト $\sigma_{jk}$ が十分大きいとき，$h'_{ijk}$ は分散 1 に正規化されるが，逆に非常に小さければ何もされないという振る舞いは式 (10) と同じだが，$\sigma_{jk}$ がその中間の値をとる場合に，両者の切り替えがスムーズに行われる点が異なる．[*11]

一連の処理 $h_{ijk}\to\bar{h}_{ijk}\to h'_{ijk}$ 全体を**局所コントラスト正規化（Local Contrast Normalization）**と呼ぶ．同じものを除算正規化と呼ぶこともある [41], [54], [60].

#### 2.4.2 処理の意味

元々，局所コントラスト正規化ないし除算正規化は，哺乳類の初期視覚野のニューロンが示す振る舞いを説明するモデルとして考案された [60], [63]. [*12] これが何を意味するかは，式 (8), (9) の平均と分散を求める対象が画像平面上の小領域ならば，直観的に理解できる．しかしこれらの式にあるように，一般にはフィルタ空間（全フィルタ）にわたる平均および分散が計算される．

Simoncelli らは，フィルタの集合がスケールと方向の異なるガボールフィルタバンクのとき，この処理には自然画像の統計に関連した特別な意味があることを示した [41], [54]. 自然画像（あるいは写真）は統計的に偏った性質を持ち，例えばホワイトノイズとは全く異なっていることは良く知られる [24]．例えば，ガボールフィルタバンク（あるいは類似の計算処理）に入力して得られる出力には，顕著な性質がある．つまり，各フィルタの出力の統計分布は，0 付近に鋭いピークを持つとともに分布の裾が長く，Student の t 分布でよく表される．また，特にパラメータの近い 2 つのフィルタ出力を $h_1$, $h_2$ とするとき，これらの間には依存関係があり，具体的には，条件付き分布 $p(h_1\mid h_2)$ が蝶ネクタイ（bow-tie）のような形状をとる．式 (9), (11) は，この依存関係をキャンセルするように働いているとされる [41], [54].

## 3. 多層 NN の教師なし学習

### 3.1 プレトレーニング：貪欲法による各層の局所最適化

2.1.2 節に述べたように，多層 NN の誤差逆伝播法による学習は，ランダムに初期値をセットした場合はほとんどうまく行かない．この問題にひとつの解決を与えたのが，ディープラーニングの研究が現在のように隆盛する契機ともなった，Hinton らの Deep Belief Network（DBN）の研究 [13] である．

この研究を要約すれば，目的とする多層ネットワークを連続 2 層に分離し，それぞれを教師なしで学習し，これを層の数だけ実行するというものである．手順は入力層を含む最も下の 2 層から開始し，その結果得られる隠れ層の状態をその上の 2 層の入力に読み替えて教師なし学習を行い，これを繰り返す．

この教師なし学習を**プレトレーニング（pretraining）**と呼ぶ．その後，そこで得たネットワークのパラメータを初期値として，ネットワーク全体の微調整（fine-tune）を行う．微調整には教師あり，あるいは教師なしの学習があり得る．例えば，得られたネットワークをそのまま順方向伝播型の NN に読み換えて，教師あり学習を誤差逆伝播法を行う．その場合，ランダム初期値ではうまくいかない場合でも，プレトレーニングで得た初期値から始めることで，よい局所最適解を見つけられる．

なお，DBN はボルツマンマシンに基礎を置き，ネットワークの振る舞いは確率的に記述され，生成モデルに分類される．上の教師なし学習の最小単位となる 2 層は，Restricted Boltzmann Machine(RBM) が与え，実際の学習は Contrastive Divergence 法により行われる．これらは 4 で改めて説明する．

一方，同じ方法論—2 層ごとの局所的な教師なし学習を下の層から繰り返すプレトレーニング—は，オートエンコーダにも適用することができ，その理解はずっと簡単である．そこで以下ではオートエンコーダを軸に，多層 NN

[*11] 式 (11) と (10) の関係は，ちょうど NN のユニットの非線形入出力に，シグモイド関数とステップ関数のどちらを使うかと相似である．

[*12] 具体的には，特定周期・方向の縞模様に選択的に反応するニューロンについて，その反応の強さはパタンのコントラストに依存するが，その選択性は依存しないというもの [16].

のプレトレーニングの方法を述べる．

### 3.2 オートエンコーダ（Autoencoder）

オートエンコーダは NN に対する教師なし学習の一手法で，ディープラーニングの根幹をなす一要素として，後述する RBM と双璧をなす．なお，オートエンコーダには確率的な概念はないが，RBM との類似性が指摘されている [62].

#### 3.2.1 2 層 NN のオートエンコーダ

入力層（$\mathbf{x}$）と隠れ層（$\mathbf{h}$）の 2 層からなる NN を考え，順方向の伝播

$$\mathbf{h} = f(\mathbf{W}\mathbf{x} + \mathbf{b}) \tag{12}$$

を，入力 $\mathbf{x}$ から符号 $\mathbf{h}$ を得るエンコーダ（encoder）と見なす．これと逆向きの同様の計算

$$\mathbf{y} = f'(\mathbf{W}'\mathbf{h} + \mathbf{b}') \tag{13}$$

を（仮想的に）考え，これをデコーダ（decoder）と見なす．オートエンコーダとは，与えられた入力 $\mathbf{x}$ をエンコーダで符号化した $\mathbf{h}$ を，今度はデコーダで復号化し $\mathbf{x}$ を再現する $\mathbf{y}$ を得たとき，$\mathbf{y}$ が元の $\mathbf{x}$ に近くなるようにパラメータ $\mathbf{W}$, $\mathbf{b}$, $\mathbf{W}'$ および $\mathbf{b}'$ を決定する方法である[*13]．すなわち，与えられたサンプルの集合 $\{\mathbf{x}_1, \ldots\}$ に対し，

$$\min_{\mathbf{W},\mathbf{b},\mathbf{W}',\mathbf{b}'} \sum_i \|\mathbf{x}_i - f'(\mathbf{W}' f(\mathbf{W}\mathbf{x}_i + \mathbf{b}) + \mathbf{b}')\|_2^2 \tag{14}$$

の最小化によってパラメータを決定する．

オートエンコーダの振る舞いは，$f$ および $f'$ に何を選ぶかと，$\mathbf{x}$ と $\mathbf{y}$ の近さを測る損失関数 $L(\mathbf{x}, \mathbf{y})$ の選択によって変わる．また $\mathbf{W}$ と $\mathbf{W}'$ は一般には異なるが，$\mathbf{W}' = \mathbf{W}^\top$ と制約することもある．

$f$ および $f'$ には，シグモイド関数や恒等写像[*14]を用いる．例えばエンコーダ，デコーダともにアフィン変換とした場合，上述の学習は実質的に PCA と等価になることが知られている [1]．（$f$ ないし $f'$ にシグモイド関数を利用すると，その限りでない．詳しい振る舞いは [8].）

式 (14) は，損失関数として最も一般的な 2 乗誤差 $L(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2^2$ を選んだ場合の式である．$\mathbf{x}$ が 2 値の場合には，$f'$ にシグモイド関数を使う前提で

$$L(\mathbf{x}, \mathbf{y}) = -\sum_i x_i \log y_i + (1 - x_i) \log(1 - y_i) \tag{15}$$

をよく用いる [2], [62].

式 (14) の最小化によって期待しているのは，この NN が与えられた学習サンプルの構造をうまく捉えた表現 $\mathbf{h}$ を得ることと，さらに学習サンプル以外の入力にも当てはまる汎化能力をもつことである．$\mathbf{W}'$ および $\mathbf{b}'$ は，学習時に入力を再現するためだけに用いられ，入力からその特徴を求める順方向の計算では使用しない．

行列 $\mathbf{W}$ のサイズを $d' \times d$ とするとき，$d' \geq d$ であれば，最適解は $\mathbf{x} \to \mathbf{y}$ の写像が恒等写像になる（つまり入力はいつも 100%再現できる）というトリビアルなものになってしまう．$d' < d$ の制約を課すか，あるいは $\mathbf{h}$ がなるべくスパースに（＝ベクトルの非零成分の数が少なく）なるようなスパース正則化を行うのが一般的である．

ただし $d' < d$ とした場合，NN の表現力はそうでない場合に比べて低下し，多様な特徴を表現する能力は制限されると考えられる．一方スパース正則化は，陰に $d < d'$ の制約を課すことで，$d' > d$ のままでトリビアルな解を回避でき，特徴の過完備性（overcompleteness）により高い性能を実現できる [11]．これについては 3.3 節で述べる．この他にもトリビアルでない解を得る方法として，デノイジングオートエンコーダ [62] が提案されている．

#### 3.2.2 オートエンコーダによるプレトレーニングと微調整

3.1 節に述べたように，オートエンコーダを用いることで多層 NN のプレトレーニングが行える [3]．具体的な手順は次のようなものである．

1. 入力層を含む最初の 2 層を，上述のオートエンコーダの方法で，学習サンプルを用いて訓練する．
2. 学習結果を用いて計算される，学習サンプルに対する隠れ層の状態を，次の 2 層への入力と見なし，同じくオートエンコーダの方法で訓練する．
3. 2 を層の数だけ繰り返す．

この後は微調整のステップになるが，2 通りのやり方が考えられる．一つは，最上位層のさらに上に，最上位層の出力を入力とする（一般には小さく，浅い）NN を接続し，全体を一つの NN と見なし，教師あり学習を行う．もう一つは，上で訓練した多層 NN を大きなオートエンコーダ（ディープオートエンコーダ）と見なし，教師なしでパラメータの微調整を行う．その場合，最下層への入力サンプル $\mathbf{x}$ に対し，ここから順方向の伝搬によって最上層のユニットの状態 $\mathbf{h}^k$ を計算し，逆にここから $\mathbf{x}$ を再現した $\mathbf{y}$ を求め，その差が小さくなるように学習を行うことになる．パラメータをランダム初期値から始めた場合には，多層 NN の学習困難さと同じ理由でこの方法はうまくいかないが，プレトレーニングのおかげで，学習がうまく行えると期待される．

### 3.3 特徴のスパースな表現

#### 3.3.1 スパースコーディング

特徴の表現をスパースにすることで，認識系の性能を向上させられることが分かってきた．その端緒となったスパースコーディング [43] は，

$$J(\mathbf{x}, \mathbf{h}, \mathbf{W}) = \|\mathbf{x} - \mathbf{W}\mathbf{h}\|_2^2 + \lambda g(\mathbf{h}) \tag{16}$$

なる関数を考え，与えられたサンプル $\{\mathbf{x}_1, \ldots\}$ に対するそ

[*13] 文献によっては Autoassociator とも書かれる．
[*14] このとき $\mathbf{x} \to \mathbf{h}$ ないし $\mathbf{h} \to \mathbf{y}$ は，アフィン変換になる．

の和を次のように最小化する．

$$\min_{\mathbf{W},\{\mathbf{h}_i\}} \sum_i J(\mathbf{x}_i, \mathbf{h}_i, \mathbf{W}) \tag{17}$$

ただし実際の最小化は，$\mathbf{W}$ の各列ベクトルの L2 ノルムがある定数以下であるという不等式制約の下で行う．$\mathbf{W}$ は辞書と呼ばれ，その各列に基底ベクトル（厳密には基底ではないが便宜上そう呼ぶ）を格納した行列である．$\mathbf{h}$ はサンプル $\mathbf{x}$ を符号化したコードと考え，$\mathbf{W}$ を定めた後は，入力された $\mathbf{x}$ に対し $\mathbf{W}$ を固定して $J$ を最小化（ $\min_{\mathbf{h}} J(\mathbf{x}, \mathbf{h}, \mathbf{W})$）し，コード $\mathbf{h}$ を計算する．

与えられた $\mathbf{x}$ に対する $\mathbf{h}$ を計算するとき，$J$ の第 1 項を小さくすることで入力 $\mathbf{x}$ を規定の線形和 $\mathbf{Wh}$ でなるべく正確に表現しようとし，$J$ の第 2 項を小さくすることで $\mathbf{h}$ をなるべくスパースに，つまり非零成分の数を少なくしようとする．この 2 つの一般には相反する要請を満たす解として，$\mathbf{W}$ のなるべく少ない数の基底の組み合わせで $\mathbf{x}$ が表現される結果になる．

関数 $g(\cdot)$ は $\mathbf{h}$ をスパースにする効果のあるものが使用され，圧縮センシング（compressed sensing）でポピュラーな $g(\mathbf{h}) = \sum_j |h_j|$ を始め，$g(\mathbf{h}) = \sum_j -e^{-h_j^2}$[43]，$g(\mathbf{h}) = \sum_j \log(1+h_j^2)$[43], [46] や， $g(\mathbf{h}) = \sum_j \log(\cosh(h_j))$[24], [30] がある．これらの $g(\mathbf{h})$ はいずれも，$\mathbf{h}$ の各成分のうち非零となるものの数が少ないほど小さい効果を持ち（counting norm の $\|\mathbf{h}\|_0$ を近似する効果があるため），使用する基底の数を減らす方向に作用する．

$\mathbf{x}$ を自然画像からランダムに切り出した小さなパッチとし，その集合に対し上の $J$ を最小化したときに得られる $\mathbf{W}$ の各基底は，霊長類の脳の初期視覚野に発見されているもの—空間周波数および方向を変化させて得られるガボールウェーブレット—と酷似したものとなることが知られている [43]．またその後，スパースコーディングは多様な工学的問題，例えば画像のデノイジング，画像復元等に応用され，良い結果が得られている．

スパースコーディングは，過完備（overcomplete）な辞書 $\mathbf{W}$ を考えたときに，その真価を現すと考えられる．この場合，過完備とは $\mathbf{W}$ を横長（サイズを $d \times d'$ としたとき，$d < d'$）を指し，いわば辞書が冗長であるときである．上述のいずれの場合でも，この辞書の冗長性を武器に，自然画像のもつ統計的性質（偏り）をうまく表現できていることが要因であると考えられる．

### 3.3.2 Topographic ICA

CNN の特徴の一つは，入力層とフィルタ出力層間の結合が局所的であることと，結合のウェイト（フィルタ）がフィルタ出力層の全ユニットで同一であることである．この性質と上述のプーリングが合わさって，CNN は入力パタンの並進移動に対する不変性を実現している．（正確には，フィルタのサイズ以下の小パタンが入力層（画像）内で並進移動する場合の不変性.）この不変性は，CNN のネットワークの構成によって最初から実現されているもので，並進不変性以外の不変性，例えば（同じく小パタンの）回転やスケーリング，その他のより一般的な変形への不変性は，少なくともフィルタ層とプーリング層の 2 層では実現できない．

このような多様な変形への不変性を，学習によって獲得したい．その試みが，Hyvärinen らの Topographic ICA（トポグラフィック独立成分分析，以下 TICA）[23], [24]，およびそこから派生した「局所受容野」NN[27], [30], [31], [32] である．

Hyvärinen らの TICA は，Olshausen らのスパースコーディング（式 (16) 参照）に関する研究 [43] をさらに発展させたものである．Olshausen らは，スパースコーディングによって自然画像を対象に基底を学習すると，霊長類の初期視覚野に見られるのと同様のガボールフィルタ様の特徴が得られることを見出した．そこで学習される複数の基底（特徴）の順序には意味はない（基底の順序の入れ替えは全く自由である）．しかるに，霊長類の初期視覚野での特徴の分布は，ガボールフィルタのパラメータ（スケールと回転）の近さが物理的な距離に対応する形になっていることが分かっている．つまり，パラメータの近いフィルタ（特徴）が近くにあるということである．

TICA は，スパースコーディングにはできないこの性質を再現する．図 3 の 3 層 NN を考える．最初の入力層と第 1 層間は全結合で，ここで特徴を抽出し，次の第 1，2 層間の結合は局在的なもので，ここでプーリングを行う．CNN と異なり，入力，第 1 層間の結合は全結合であるので，入力層から抽出される特徴は第 1 層のユニットごとに一般に異なるものになる．プーリングの結合は，第 2 層の各ユニット $(i)$ が第 1 層での直下のユニットを中心とする $N \times N$ のユニット群 $j \in P_i$ のみと結合をもち（この結合を $w'_{ij}$ と書く），かつ $P_i$ どうしがある程度重なるようにする．プーリングの結合重み（$w'_{ij}$）は固定である．以上のネットワーク構造に対し，

$$J(\mathbf{x}, \mathbf{h}, \mathbf{W}) = \|\mathbf{x} - \mathbf{Wh}\|_2^2 + \lambda \sum_{i=1}^{K} \sqrt{\sum_{j \in P_i} w'_{ij} h_j^2} \tag{18}$$

なる目的関数を考える．これをスパースコーディング同様，自然画像から学習サンプルを作り，これを対象に上の目的関数を最小化すると，上述の霊長類の初期視覚野に見られるのと同様の特徴の分布が第 1 層上に出来上がる．すなわち，ガボールフィルタのパラメータの近さが第 1 層上のユニット間の物理的な近さに対応するような分布 (topographic map) が得られる．

なぜそのようなことが上の目的関数によって達成されるかは次のように理解される [27]．まず，$i$ に関する和は，第 2 層のユニットはなるべく少数だけが活性化するように作

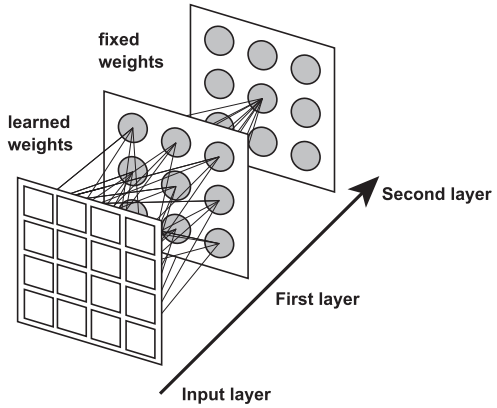


**図 3** Topographic ICA[24] のネットワーク構造.

用する．活性化するユニットが多数であるよりも少数であるほうが目的関数は小さくなるからである（スパースコーディングの $L_1$ ノルムと同じ働き）．次に $j$ に関する和は，$i$ のユニットがプールする範囲 $P_i$ にある第 1 層の複数のユニットが，なるべく均等に活性化するように作用する．なぜなら反応の 2 乗（$h_j^2$）であることから，少数のユニットが大なる反応をとるよりもその逆の方が総和が小さくなるからである．このことは間接的に，$P_i$ の範囲に類似した特徴が集まるように作用する．以上のようにして，上述の特徴の分布が実現される．

第 1 層で近いユニット間で抽出する特徴は類似するから，入力層から取り出した特徴量をプーリングした第 2 層の出力は，一定の不変性を実現することになる．CNN では，第 1 層のユニットが抽出する特徴はすべて同じであり（同一フィルタ＝マップ内での話），それによって並進移動不変性を実現していた．近接するユニットが近い特徴を取り出す今の場合，より柔軟で自由度の高い変換（変形）に対する不変性を実現できる可能性があると言える．本稿冒頭に紹介した Le らの研究 [32] でも，この考え方が取り入れられている．

### 3.3.3 スパースオートエンコーダ

先述のようにオートエンコーダは，$\mathbf{W}$ に何らの制約がなければ恒等写像を学習してしまう．$d' < d$ とする以外に意味のある解を得るもう一つの方法がスパース正則化を行うことである．いわゆるスパース正則化付オートエンコーダの最も単純なものは

$$\min_{\mathbf{W},\mathbf{b},\mathbf{b}'} \sum_i \|\mathbf{x}_i - f'(\mathbf{W}^\top f(\mathbf{W}\mathbf{x}_i + \mathbf{b}) + \mathbf{b}')\|_2^2 + \sum_i g(f(\mathbf{W}\mathbf{x}_i + \mathbf{b})) \quad (19)$$

という最小化で，デコーダの重みをエンコーダのそれと同じにしている（$\mathbf{W}' = \mathbf{W}^\top$）[30].

一方，スパース正則化をオートエンコーダに取り込んだのは Ranzato らの研究 [46], [47] が最初だが，そこでは

$$\alpha_e\|\mathbf{h} - (\mathbf{W}\mathbf{x} + \mathbf{b})\|_2^2 + \|\mathbf{x} - (\mathbf{W}f(\mathbf{h}) + \mathbf{b})\|_2^2 + \alpha_s g(f(\mathbf{h})) + \alpha_r\|\mathbf{W}\|_1 \quad (20)$$

なる形の目的関数がパラメータ（$\mathbf{W}$, $\mathbf{b}$, $\mathbf{b}'$）について最小化された．$\mathbf{h}$ は潜在変数として，最小化の反復過程でパラメータと交互に推定される．$g(\cdot)$ には，当初 sparsifying logistic 関数 [47], [48] が，後に $\sum_i \log(1 + z_i^2)$ が用いられた [46].

式 (16) と (19) を見比べれば分かるように，両者には一定の類似性がある [30]．スパースコーディングには，オートエンコーダにあるエンコーダが陽には存在せず，最適化の計算を通じて陰にエンコードが行われていると見なせる．このためスパースコーディングでは，コード $\mathbf{h}$ を得るのに最適化計算を要するが，オートエンコーダではそれはネットワークの順伝播の計算のみで得られるのが違いと言える．

## 4. 生成モデルに対するディープラーニング

本節では，Ristricted Boltzmann Machine（RBM）や Deep Belief Network（DBM）などの生成モデルに対する，ディープラーニングの学習手法を説明する．

これら生成モデルでは，前節までの NN における入力（あるいは入力層の状態）$\mathbf{v}$，および隠れ層の状態 $\mathbf{h}$ はすべて確率変数として扱われる．そして，それらの同時確率密度関数 $p(\mathbf{v}, \mathbf{h}; \theta)$ によってネットワークの振る舞いを表現し，これに基づいて推論や学習を行う．$\theta$ はネットワークの振る舞いを規定するパラメータの集合で，ユニット間結合の重みやバイアスである．また，ネットワークにサンプル $\mathbf{v}$ が入力されたときの隠れ層の状態 $\mathbf{h}$（あるいは特徴）は，条件付き確率 $p(\mathbf{h}|\mathbf{v}; \theta)$ によって表される．生成モデルでは，このように隠れ層の状態 $\mathbf{h}$ をその分布によって表現する．

生成モデルにはいくつかの利点がある．一つは，入力自体をモデルから生成できることである．つまり，学習後にネットワークがどんな特徴を学習したのかを可視化することができる．また，学習や推論がすべて統計的な枠組みで取り扱えるため，最適性がはっきりすることである．例えば，層ごとに学習最適化を繰り返すプレトレーニングの方法の妥当性を，DBN では示すことができる．

以下では，まず基本となるボルツマンマシンの説明をした後，ディープラーニングで主要な役割を果たす RBM を説明し，その後プレトレーニングの考え方を広めるきっかけになった DBN について述べる．またこれを発展させた Deep Boltzmann Machine（DBM），CNN 同様のネットワーク構造を取り入れた Convolutional DBN についても述べる．

### 4.1 ボルツマンマシン

ボルツマンマシン (Boltzmann machine) は，連想記憶のモデルとしても知られるホップフィールドネットワークを発展させ，統計物理のイジングモデルに基づいて，ネットワークの振る舞いを確率的に記述したもので，Hinton と Sejnowski によって提案された [18].

ボルツマンマシンは，複数のユニットを多数，双方向に結び付けたグラフィカルモデルによって定義される．ユニットは，データが入力される可視ユニット (visible unit) と，それ以外の隠れユニット (hidden unit) の 2 つに分けられる．可視ユニットの数を $N_V$，隠れユニットの数を $V_H$ とするとき，各ユニットは 2 値の値を取り得るとし，その状態を変数 $\mathbf{v} = (v_1, \cdots, v_{N_V})$ および $\mathbf{h} = (h_1, \cdots, h_{N_H})$ によって表す（$v_i \in \{0,1\}$，$h_j \in \{0,1\}$）．これらのユニットは，一般にそのすべてが相互に結合され，NN のように重み $\mathbf{W}$ とバイアス $\mathbf{b}$ によってその結合は記述される．以下これらネットワークのパラメータを $\theta$ と書く．

ボルツマンマシンの挙動は，変数 $\mathbf{v}$ と $\mathbf{h}$ の関数であるエネルギー関数 $E(\mathbf{v}, \mathbf{h}; \theta)$ を指定し，それによって決まる次の確率構造によって記述される（ボルツマン分布）．

$$p(\mathbf{v}, \mathbf{h}; \theta) = \frac{1}{Z(\theta)} \exp(-E(\mathbf{v}, \mathbf{h}; \theta)) \tag{21}$$

ここで，$Z(\theta)$ は正規化定数または分配関数 (partition function) と呼ばれる定数で，次のように与えられる

$$Z(\theta) = \sum_{\mathbf{v}} \sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}; \theta)) \tag{22}$$

可視ユニットにデータを入力したとき，上の確率モデルに従って定まる隠れユニットの状態を，ボルツマンマシンの出力と考えることができる．以降，いくつかの特定のボルツマンマシンについて述べるが，それぞれ対応するエネルギー関数があり，それが与える式 (21) のボルツマン分布に基づいて，パラメータの学習や各種の推論が行われる．

ボルツマンマシンでも，通常の NN 同様にネットワークのパラメータ $\theta$ を学習によって定める．もし，可視ユニットと隠れユニットの状態のペアが学習サンプルとして（多数）与えられたとすると，学習は尤度 $\prod_n p(\mathbf{v}^n, \mathbf{h}^n; \theta)$ を使った最尤推定により行える．

ボルツマンマシンの学習のハイライトは，教師なし学習にある．つまり，可視ユニットの状態 $\mathbf{v}$ の集合のみを元にパラメータを推定する．そのため，$p(\mathbf{v}, \mathbf{h}; \theta)$ を隠れユニットの状態 $\mathbf{h}$ について周辺化：$p(\mathbf{v}; \theta) = \sum_{\mathbf{h}} p(\mathbf{v}, \mathbf{h}; \theta)$ し，この周辺分布を用いてパラメータ $\theta$ を最尤推定する．$\mathbf{v}$ の学習サンプルが $N$ 個 $\{\mathbf{v}^1, \cdots, \mathbf{v}^N\}$ 得られているとき，

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \mathcal{L}(\theta) \tag{23}$$

のようにパラメータを推定する．ただし $\mathcal{L}(\theta)$ は対数尤度で

$$\mathcal{L}(\theta) = \sum_{n=1}^{N} \ln p(\mathbf{v}^n; \theta) = \sum_{n=1}^{N} \ln \sum_{\mathbf{h}} p(\mathbf{v}^n, \mathbf{h}; \theta) \tag{24}$$

のように定められる．

この対数尤度の最大化は，KL 情報量の最小化の観点からも解釈できる．まず，与えられた学習サンプルの集合は，次の式で定義される観測分布 $p_0$ によって生成されたものと考える．

$$p_0(\mathbf{v}) \equiv \frac{1}{N} \sum_{n=1}^{N} \prod_{i=1}^{N_V} \delta(v_i, v_i^n) \tag{25}$$

ここで，$\delta(x, y)$ はクロネッカーのデルタ関数である．この分布 $p_0$ を用いると，式 (23)，(24) は

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \mathcal{D}\left[p_0(\mathbf{v}) \| p(\mathbf{v}; \theta)\right] \tag{26}$$

のように書き直せる．ここで $\mathcal{D}[p\|q]$ は Kullback-Leibler 情報量，もしくは相対エントロピーと呼ばれ，

$$\mathcal{D}[p\|q] = \sum_{x} p(x) \ln \frac{p(x)}{q(x)} \tag{27}$$

のように定義される．KL 情報量は $p(x) = q(x)$ の場合に 0 の値をとり，そうでない場合は非負の値をとるので， KL 情報量は 2 つの確率分布間の近さを表す一種の距離（のようなもの）と見なせる．以上より，対数尤度を最大化する戦略は， $p_0(\mathbf{v})$ と $p(\mathbf{v}; \theta)$ 間の距離 $\mathcal{D}\left[p_0(\mathbf{v}) \| p(\mathbf{v}; \theta)\right]$ を最小化するのと等しいと分かる．この $\mathcal{D}$ は非凸であるので，主に勾配降下法を用いて局所解を探索する．

$\mathcal{D}\left[p_0(\mathbf{v}) \| p(\mathbf{v}; \theta)\right]$ をパラメータ $\theta$ について微分すると

$$\begin{aligned}
\frac{\partial \mathcal{D}}{\partial \theta} &= \sum_{\mathbf{v}} \sum_{\mathbf{h}} \frac{\partial E}{\partial \theta} p(\mathbf{h}|\mathbf{v}; \theta) p_0(\mathbf{v}) \\
&\qquad - \sum_{\mathbf{v}} \sum_{\mathbf{h}} \frac{\partial E}{\partial \theta} p(\mathbf{v}, \mathbf{h}; \theta) \\
&= \left\langle \frac{\partial E}{\partial \theta} \right\rangle_{\text{data}} - \left\langle \frac{\partial E}{\partial \theta} \right\rangle_{\text{model}}
\end{aligned} \tag{28}$$

を得る．式 (28) の第一項，第二項をそれぞれ正項（positive phase），負項（negative phase）と呼ぶ．また，$\langle\cdot\rangle_{\text{data}}$，$\langle\cdot\rangle_{\text{model}}$ はそれぞれ確率分布 $p_{\text{data}}(\mathbf{v}, \mathbf{h}) = p(\mathbf{h}|\mathbf{v}; \theta) p_0(\mathbf{v})$，$p_{\text{model}}(\mathbf{v}, \mathbf{h}) = p(\mathbf{v}, \mathbf{h}; \theta)$ についての期待値を表す．パラメータの推定は，式 (28) の 2 つの項の勾配をそれぞれ計算し，パラメータの値を逐次的に更新して求めることになる．

### 4.2 Restricted Boltzmann Machine

制約付きボルツマンマシン (Restricted Boltzmann Machine, 以下 RBM) は，図 4 のように，可視ユニット同士，隠れユニット同士の間に結合がなく，可視ユニットと隠れユニットの間のみ結合があるボルツマンマシンである [19]. RBM は，後述する DBM(Deep Boltzmann Machine) や DBN(Deep Belief Network) の構成要素でもある．

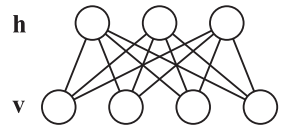


**図 4** Restricted Boltzmann Machine のネットワーク

RBM では，式 (21) のエネルギー分布を以下のように定める．

$$E(\mathbf{v},\mathbf{h};\theta) = -\sum_{i=1}^{N_V} b_i v_i - \sum_{j=1}^{N_H} c_j h_j - \sum_{i,j} v_i W_{ij} h_j \quad (29)$$

RBM のパラメータは，$\theta = (\mathbf{b},\mathbf{c},\mathbf{W})$ である．$\mathbf{b},\mathbf{c}$ はそれぞれ可視ユニット，隠れユニットの持つバイアス，$\mathbf{W}$ は可視ユニットと隠れユニット間の結合の重みである．

RBM には可視ユニット同士，隠れユニット同士での結合がないので，$\mathbf{h}$ が与えられたときの $\mathbf{v}$ の条件付き確率，ならびに $\mathbf{v}$ が与えられたときの $\mathbf{h}$ の条件付き確率を次のように解析的に計算できる．

$$p(v_i = 1|\mathbf{h};\theta) = \sigma\left(b_i + \sum_j W_{ij} h_j\right) \quad (30)$$

$$p(h_j = 1|\mathbf{v};\theta) = \sigma\left(c_j + \sum_i v_i W_{ij}\right) \quad (31)$$

ここで $\sigma(x) = 1/(1+\exp(-x))$ で，これは（ロジスティック）シグモイド関数である．これらの結果から，条件付き確率 $p(\mathbf{h}|\mathbf{v};\theta)$（特徴ベクトル）が計算できるとともに，式 (28) の正項を解析的に計算することが可能になる．式 (25), (30), (31) の関係を式 (28) の正項に代入し，整理すると

$$-\left\langle \frac{\partial E}{\partial W_{ij}} \right\rangle_{\text{data}} = \frac{1}{N}\sum_n v_i^n \sigma(c_j + \sum_{i'} v_{i'}^n W_{i'j}) \quad (32a)$$

$$-\left\langle \frac{\partial E}{\partial b_i} \right\rangle_{\text{data}} = \frac{1}{N}\sum_n v_i^n \quad (32b)$$

$$-\left\langle \frac{\partial E}{\partial c_j} \right\rangle_{\text{data}} = \frac{1}{N}\sum_n \sigma(c_j + \sum_{i'} v_{i'}^n W_{i'j}) \quad (32c)$$

を得る．

#### 4.2.1 ギブスサンプリング

上のように RBM では，式 (28) の正項を解析的に導出できる一方で，負項は期待値を求めるための状態数が爆発してしまうため，その限りでない．後述する Contrastive Divergence（CD）法が登場するまで，負項の計算は，マルコフ連鎖モンテカルロ法の一種である古典的なギブスサンプリングを用いるより他，なかった．CD 法の説明のため，そのやり方をまず説明する．

負項は $p_{model}$ に関する期待値である．$p_{\text{model}}$ を，RBM の状態（可視ユニット ($N_V$ 個) と隠れユニット（$N_H$ 個））を，学習サンプルの数と同じ $N$ 個のサンプルで表し，これらサンプルを用いた離散近似によって近似的に表現することを考える．ギブスサンプリングによって，これらのサンプルを以降の手順に従って $T$ 回反復的に更新し，真の $p_{\text{model}}$ に近い分布 $p_T$ を生成することが目的である．

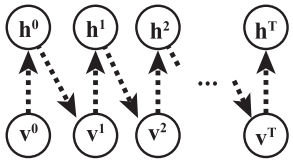


**図 5** ギブスサンプリングによる反復図

まず，反復計算での各サンプルの初期値は次のように生成する．可視ユニットについては，$N$ 個の学習サンプル $(\mathbf{v}^1,\cdots,\mathbf{v}^N)$ をそのまま初期値 $(\mathbf{v}_0^1,\cdots,\mathbf{v}_0^N)$ とする．隠れユニットについては，$\mathbf{v}_0^n$ が与えられたとしたときの $\mathbf{h}_0^n$ の条件付き確率 $p(\mathbf{h}_0^n|\mathbf{v}_0^n;\theta)$（式 (31) から評価可能）に従ってサンプリングし，初期値 $(\mathbf{h}_0^1,\cdots,\mathbf{h}_0^N)$ を生成する．

反復における各回の更新は以下のように行う．可視ユニットのサンプル $\mathbf{v}_t^n$ は，条件付き確率 $p(\mathbf{v}_{t+1}^n|\mathbf{h}_t;\theta)$ に従って $\mathbf{v}_{t+1}^n$ をサンプリングし，これを更新値とする．隠れユニットのサンプル $\mathbf{h}_t^n$ は，条件付き確率 $p(\mathbf{h}_{t+1}^n|\mathbf{v}_{t+1};\theta)$ に従って $\mathbf{h}_{t+1}^n$ をサンプリングし，これを更新値とする．この更新を $T$ 回繰り返す．ただし，隠れユニットのサンプルの最後の更新ではサンプリングを行わずに，条件付き確率 $p(\mathbf{h}_{t+1}^n|\mathbf{v}_{t+1};\theta)$ を求めた後，次のように $p = p_{\text{model}}$ を推定する．

$$\begin{aligned} p(\mathbf{v},\mathbf{h};\theta) &\approx p_T(\mathbf{v})p(\mathbf{h}|\mathbf{v};\theta) \\ &= \frac{1}{N}\sum_{n=1}^{N}\prod_{i=1}^{N_V}\delta(v_i,\hat{v}_i^n)\prod_{j=1}^{N_H}\sigma(c_j + \sum_{i'} v_{i'} W_{i'j}) \end{aligned} \quad (33)$$

ここで，$\hat{v}_i^n$ は，$T$ 回更新後の可視ユニットのサンプルである．これを用いると，式 (28) の負項は次のように評価できる．

$$-\left\langle \frac{\partial E}{\partial W_{ij}} \right\rangle_{\text{model}} = \frac{1}{N}\sum_n \hat{v}_i^n \sigma(c_j + \sum_{i'} \hat{v}_{i'}^n W_{i'j}) \quad (34)$$

$$-\left\langle \frac{\partial E}{\partial b_i} \right\rangle_{\text{model}} = \frac{1}{N}\sum_n \hat{v}_i^n \quad (35)$$

$$-\left\langle \frac{\partial E}{\partial c_j} \right\rangle_{\text{model}} = \frac{1}{N}\sum_n \sigma(c_j + \sum_{i'} \hat{v}_{i'}^n W_{i'j}) \quad (36)$$

#### 4.2.2 Contrastive Divergence

90 年代までは，ギブスサンプリングを用いた上のような計算を行うのが一般的であった．近似の精度を十分なものとするには多くのサンプルの生成を必要とし，それには膨大な計算コストを要するから，大規模なネットワークの学習は現実的でなかった．Contrastive Divergence(CD) 法は，この計算コストの問題を大きく改善できる方法で，Hinton によって提案された [19].

ギブスサンプリングの繰り返し数 $T$ に対する Contrastive Divergence（$T$-CD）とは

$$\mathcal{D}[p_0(\mathbf{v})\|p(\mathbf{v};\theta)] - \mathcal{D}[p_T(\mathbf{v})\|p(\mathbf{v};\theta)] \quad (37)$$

のように定義される．CD 法では，前述の $\mathcal{D}[p_0(\mathbf{v})\|p(\mathbf{v};\theta)]$ の代わりに，これを勾配降下法で最小化する．その際，少なくとも RBM の学習では小さな $T$，具体的には $T=1$，すなわちたった一度の反復だけでも，十分よい精度を実現できることが知られている．なぜ小さい $T$ で十分精度の高いパラメータが得られるかの理由の説明は，[2], [12], [19] を参照されたい．

その後，CD 法よりも精度の高い Persistent Contrastive Divergence(PCD) が Tieleman により提案されている [61]．CD は勾配降下法によるパラメータの更新のたびに，学習サンプルを初期値とする $T$ 回のギブスサンプリングを行う．一方, PCD では初期値として, 前回更新時に用いたギブスサンプリングの結果を用いる．現在, 多くの RBM の学習では CD ではなくこの PCD を用いることが多い [14], [36], [51]．

#### 4.2.3 スパース RBM

RBM の可視ユニットに学習サンプルを入力したとき，活性化する（$h_j=1$）隠れユニットの数を少なくする（＝スパースにする）方法が提案されている．その意図は 3.3 節で述べたものと同じである．

ここでは，最も一般的だと思われる Lee らの手法 [39] を説明する．上で説明した RBM の学習は，元をたどれば負の対数尤度 $\mathcal{L}(\theta)$ を $\theta$ について最小化することに相当する．スパース RBM では，この目的関数 $-\mathcal{L}(\theta)$ に，活性化する隠れノードの数を制限するための項を新たに加えた，

$$-\mathcal{L}(\theta)+\lambda\sum_j\left(\rho-\langle h_j\rangle_{p_0(\mathbf{v})p(h_j|\mathbf{v})}\right)^2 \quad (38)$$

を考え，これを最小化する（$p_0$ は式 (25) で定義）．ここで $\lambda$ はスパース性をどの程度重視するかの重みであり，$\rho$ は $h_j=1$ となる条件付き確率の平均を表す．通常，$\rho$ は十分に小さい正の値を設定する．

実際の最適化の計算は，式 (38) 全体の最小化を勾配法で行わず，対数尤度の最小化とスパース項の最小化を，それぞれ交互にかつ独立に勾配法で行う[*15]．

#### 4.2.4 可視ノードが連続値をとる場合

上では，可視ユニットの状態 $v_i$ は $\{0,1\}$ の離散的な値を取るものとした．問題によっては，$v_i$ は連続値をとるとしたい (例えばグレースケールの画像の各ピクセルの値を可視ユニットに入力するなど)．その場合，次のエネルギー関数を考えればよい [2]．

$$E(\mathbf{v},\mathbf{h};\theta)=\sum_{i=1}^{N_V}\frac{(v_i-b_i)^2}{2\sigma_i^2}-\sum_{j=1}^{N_H}c_jh_j-\sum_{i,j}\frac{v_i}{\sigma_i}W_{ij}h_j \quad (39)$$

RBM の学習に必要となる条件付き確率 $p(\mathbf{v}|\mathbf{h};\theta)$, $p(\mathbf{h}|\mathbf{v};\theta)$ は，

$$p(v_i|\mathbf{h};\theta)=\mathcal{N}\left(v_i;b_i+\sum_jW_{ij}h_j,\sigma_i^2\right) \quad (40a)$$

$$p(h_j=1|\mathbf{v};\theta)=\sigma\left(c_j+\sum_iv_iW_{ij}\right) \quad (40b)$$

のように計算される．ここで，$\mathcal{N}(x;\mu,\sigma^2)$ は 1 次元正規分布を表す．実際の学習最適化は，ギブスサンプリングによって $\mathbf{v}$ を生成する際，離散値の場合に用いた式 (30) を式 (40a) に置き換えれば済む．

なお，正規分布の分散 $\sigma_i^2$ を，学習時，他のパラメータと同様に勾配法で決定することは理論的には可能だが，良い結果を得ることは比較的難しい．そのため，予め入力ベクトルの各成分を平均が 0，分散が 1 の正規化されたベクトルに変換し，かつ式 (39) の $\sigma_i$ を 1 に固定した上で残りのパラメータの学習を行っている [12]．

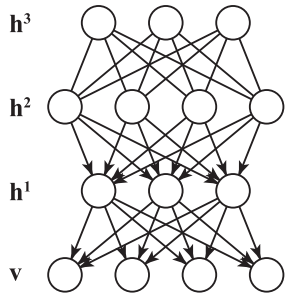

**図 6** Deep Belief Network.

### 4.3 Deep Belief Network

Deep Belief Network（DBN）[13] は，図 6 に示すように，最上層は無向の 2 部グラフからなり，それ以外の層は有向グラフとなっていて，高次から低次の層に向かって情報が伝搬する構造を持つ．この構造から，$L$ 個の隠れ層を持つ DBN の同時確率分布は，次のように与えられる．

$$p(\mathbf{v},\mathbf{h};\theta)=\left(\prod_{l=0}^{L-2}p(\mathbf{h}^l|\mathbf{h}^{l+1})\right)p(\mathbf{h}^{L-1},\mathbf{h}^L) \quad (41)$$

ただし，$\mathbf{h}^0=\mathbf{v}$ とおいた．式 (41) の $p(\mathbf{h}^l|\mathbf{h}^{l+1})$ は，次の式によって表される．

$$p(h_i^l|\mathbf{h}^{l+1})=\sigma\left(c_i^l+\sum_jW_{ij}^{l+1}h_j^{l+1}\right) \quad (42)$$

また，式 (41) の $p(\mathbf{h}^{L-1},\mathbf{h}^L)$ は，RBM のそれ（式 (29)）と同じものになる．

この DBN に対し，4.1 節に述べた勾配降下法を使って全体のパラメータを決定しようとしても，良好な結果は得られない [13]．それはちょうど，ディープオートエンコーダ（多層 NN に対するオートエンコーダ）がランダムな初期値からでは学習がうまくできないのに似ている．そこで，

[*15] そうする理由は不明で，単に便宜的な理由によると思われる．

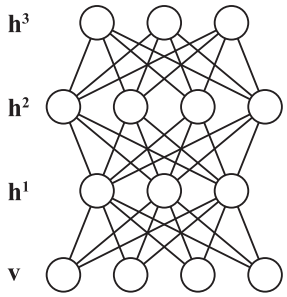


**図 7** Deep Boltzmann Machine のネットワーク.

まず部分的な層の学習を可視層（最下層）から上位層へと順番に行うプレトレーニングを実行する．そしてこれらのパラメータを初期値に用いて，全体の学習を行う．以下では，DBN を使った推論，プレトレーニングについて順に述べる．

#### 4.3.1 推論

DBN は上位層から下位層へ情報が伝搬する構造を持つため，RBM と異なり，条件付き確率 $p(\mathbf{h}|\mathbf{v})$ の周辺分布を解析的に計算できない．

そこで，近似ではあるものの RBM の条件付き確率の式 (式 (31)) を参考に，$p(\mathbf{h}|\mathbf{v})$ を次式の分布 $q$ で近似する [13].

$$q(h_j^l|\mathbf{h}^{l-1}) = \sigma\left(c_j^l + \sum_i h_i^{l-1} W_{ij}^l\right) \tag{43}$$

ただし，$\mathbf{h}^0 = \mathbf{v}$ とおいた．これにより，上位層の状態の条件付き確率 $q(\mathbf{h}^L|\mathbf{v})$ を，可視層から上位層に向って再帰的に計算することで評価できる．

#### 4.3.2 プレトレーニング

プレトレーニングでは，下位層から上位層へ順番に，部分的な 2 層間のパラメータの学習を順番に行う（貪欲法）．すなわち，最初に可視層と第 1 の隠れ層間のパラメータを学習し，次に第 1, 第 2 の隠れ層間のパラメータを学習するという具合である．可視層と第 1 の隠れ層間，つまり $\mathbf{v}$ と $\mathbf{h}^1$ 間では，$\mathbf{v}$ と $\mathbf{h}^1$ で構成される RBM を用いる．こうして得た $\mathbf{v}$, $\mathbf{h}^1$ 間のパラメータを固定し，次に $\mathbf{h}^1$ と $\mathbf{h}^2$ 間のパラメータを学習する．そこでは，$\mathbf{h}^1$ と $\mathbf{h}^2$ で構成される RBM を用いる．このとき $\mathbf{h}^1$ の学習サンプルが必要となるが，これは $q(h_j^l|\mathbf{h}^{l-1})$ によって生成する．あとはこの手順を層の数だけ繰り返し，すべてのパラメータを決める．

### 4.4 Deep Boltzmann Machine

Deep Boltzmann Machine(DBM) は，RBM をそのまま何層か積み重ね，下の RBM の隠れユニットの層が上の RBM の可視ユニットの層と一致するようにしたもので（図 7），Salakhutdinov らによって提案された [51].

DBM は，すべての可視ユニットと隠れユニットの状態を変数とする単一のエネルギー分布 $E(\mathbf{v}, \mathbf{h})$ で表現される．隠れユニットの層が 2 層（RBM が 2 層）である DBM を例にとると，式 (21) のエネルギー関数は

$$E(\mathbf{v}, \mathbf{h}; \theta) = -\sum_{i=1}^{N_V} b_i v_i - \sum_{j=1}^{N_H^1} c_j^1 h_j^1 - \sum_{i,j} v_i W_{ij}^1 h_j^1 - \sum_{k=1}^{N_H^2} c_k^2 h_k^2 - \sum_{j,k} h_j^1 W_{jk}^2 h_k^2 \tag{44}$$

となる．ここで，$N_H^1$, $N_H^2$ はそれぞれ第 1，第 2 の隠れユニット層でのユニット数を表す．

#### 4.4.1 推論

RBM の場合，隠れユニット間の結合が存在しないことから，可視ユニットが与えられた時の隠れユニットの条件付き確率を厳密に求めることができた．一方 DBM にはそのような性質はなく，特徴ベクトルを厳密に計算することはできない．

そのため DBM では，平均場近似 (Mean Field Approximation)[44] を用いて条件付き確率 $p(\mathbf{h}|\mathbf{v};\theta)$ を計算（あるいは推定）するのが一般的である．平均場近似とは，系の中で本来，互いに依存関係のある変数の独立性を仮定した上で，目的の分布を近似計算する方法である．今の場合，各ユニットの状態 $h_j^l$ がそれぞれ独立であることを仮定し，$p(\mathbf{h}|\mathbf{v};\theta)$ を次の分布

$$q(\mathbf{h}|\mathbf{v}) \equiv \prod_l \prod_j q_j^l(h_j^l|\mathbf{v}) \tag{45}$$

で近似する．この独立性の仮定は，元の DBM では一般に成り立たないが（その分だけ近似精度が悪くなる），目的とする $h_j^l$ の条件付き確率を用意に計算できるのが利点である．

このように表現できる $q$ のうち，$p$ に最も近いものを $p$ の近似分布とし，そこから条件付分布を計算する．具体的には，次の KL 情報量 $\mathcal{D}[q\|p]$ を最小化する $q$

$$q^*(\mathbf{h}|\mathbf{v}) = \operatorname*{argmin}_q \mathcal{D}[q(\mathbf{h}|\mathbf{v})\|p(\mathbf{h}|\mathbf{v};\theta)] \tag{46}$$

を求める．右辺の汎関数は一般に，$q$ について非凸の関数であるので，反復計算で局所解を求める．$\mu_j^l \equiv q_j^l(h_j^l = 1|\mathbf{v})$ と定義する．式 (46) に式 (29), (45) を代入し，$\mu_j^l$ の停留点を求める方程式を導出すると，次（固定点方程式）を得る．

$$\mu_j^l = \sigma\left(c_j^l + \sum_i \mu_i^{l-1} W_{ij}^l + \sum_k W_{jk}^{l+1} \mu_k^{l+1}\right) \tag{47}$$

ここで，$\mu_i^0 = v_i$ とした．この式に従って，$\mu_j^l$ を収束するまで更新する．得られる $\mu_j^l$ は，$p(h_j^l|\mathbf{v})$ の推定値となる．

#### 4.4.2 プレトレーニング

DBM のプレトレーニングは，他の多層ネットワーク同様，層ごとの教師なし学習を最下層から上へ向けて順番に繰り返すやり方で行う [13], [51]．しかし，このプレトレーニングの方法は，DBN のそれとは多少異なる．ここでは，DBM の隠れユニットの層が 2 層の場合について説明する．

まず，$\mathbf{v}$–$\mathbf{h}^1$ 間のプレトレーニングについて説明する．$\mathbf{v}$

の層に注目すると，$\mathbf{v}$ は $\mathbf{h}^1$ についての影響だけを受ける．一方，$\mathbf{h}^1$ の層は，$\mathbf{v}$ と $\mathbf{h}^2$ 両方からの影響を受ける．この 2 層間の影響力の差異を擬似的に再現するため， DBM では，$\mathbf{v}$–$\mathbf{h}^1$ 間のプレトレーニング時における条件付き確率を次式で定める．

$$p(h_j^1 = 1|\mathbf{v}) = \sigma\left(\sum_i v_i W_{ij}^1 + \sum_i v_i W_{ij}^1\right) \tag{48}$$

$$p(v_i = 1|\mathbf{h}^1) = \sigma\left(\sum_j W_{ij}^1 h_j\right) \tag{49}$$

パラメータの更新は，上の 2 つの式を用いて正項，負項を計算することで行う．

次に，$\mathbf{h}^1$–$\mathbf{h}^2$ 間のプレトレーニングについて説明する．ここで使用する学習サンプルには，式 (48) の条件付き確率の値を用いる．$\mathbf{h}^1$ の層は $\mathbf{v}$ と $\mathbf{h}^2$ 両方からの影響を受ける一方，$\mathbf{h}^2$ の層は，$\mathbf{h}^1$ だけしか影響を受けない．そのため，DBM は $\mathbf{h}^1$–$\mathbf{h}^2$ 間のプレトレーニング時における条件付き確率を次の式で定める．

$$p(h_k^2 = 1|\mathbf{h}^1) = \sigma\left(\sum_j h_j^1 W_{jk}^2\right) \tag{50}$$

$$p(h_j^1 = 1|\mathbf{h}^2) = \sigma\left(\sum_k W_{jk}^2 h_k^2 + \sum_k W_{jk}^2 h_k^2\right) \tag{51}$$

パラメータの更新は，この 2 つの式に基づいて正項，負項を計算して行う．

最後に, 上のプレトレーニングで得たパラメータ $\mathbf{W}^1, \mathbf{W}^2$ を用いて，DBM の初期パラメータを構成する．

#### 4.4.3 微調整

上のプレトレーニングで得た DBM は，プレトレーニングで得たパラメータを初期値に，4.1 節の方法で微調整することができる．

4.1 節で述べたように，学習は式 (28) に基づく勾配降下法によって行う．式 (28) の正項，負項の計算が必要になるが，正項は平均場近似を用いて条件付き確率を近似計算する．$\mu_j^{ln} \equiv p(h_j^l = 1|\mathbf{v}^n)$ と定義すると，DBM の正項は次のように計算できる．

$$-\left\langle \frac{\partial E}{\partial W_{ij}^l} \right\rangle_{\text{data}} = \frac{1}{N}\sum_n v_i^n \mu_j^{ln} \tag{52}$$

$$-\left\langle \frac{\partial E}{\partial b_i} \right\rangle_{\text{data}} = \frac{1}{N}\sum_n v_i^n \tag{53}$$

$$-\left\langle \frac{\partial E}{\partial c_j^l} \right\rangle_{\text{data}} = \frac{1}{N}\sum_n \mu_j^{ln} \tag{54}$$

一方負項は，RBM と同様に CD 法を用いて近似計算する．ただし，RBM とは異なり DBM は隠れ素子同士が結合していることを踏まえ，ギブスサンプリングの回数 $T$ を RBM より多めにとる必要があるとされる（[51] では，$L = 2, 3$ 層の DBM に対し，CD 法の遷移回数を $T = 5$ と設定している）．

### 4.5 Convolutional DBN

CNN の基本要素であるプーリングは，フィードフォワードのアーキテクチャを前提としている．Lee らはこのプーリング，特にマックスプーリングを RBM に基づく生成モデルに組み込む方法（確率的マックスプーリング（probabilistic max pooling）と呼んだ）を示し，CNN と同様の構造を持つ RBM/DBN（convolutional-RBM/DBN）を提案した [38].

まず，通常の RBM のネットワークを CNN と同様の構造に変更し，これに伴ってエネルギー関数を次のように決める．

$$E(\mathbf{v}, \mathbf{h}) = -\sum_k \sum_{i,j} \left(h_{i,j}^k (\mathbf{W}^k * \mathbf{v})_{i,j} + c_k h_{i,j}^k\right) - b\sum_{i,j} v_{i,j} \tag{55}$$

ここで，$(i, j)$ は各層のユニットのインデックス，$k$ は各フィルタのインデックスであり，$\mathbf{W}^k$ がフィルタを表し，$W^k * \mathbf{v}$ は，フィルタと入力層 $\mathbf{v}$ の畳み込みを表す．隠れ層はフィルタの数だけ並列に存在し，それぞれはフィルタ $W^k$ とバイアス $c_k$ を共有するというモデルである．

隠れ層のさらに一つ上にプーリング層を考える．プーリング層のユニット $p_\alpha$ は，隠れ層の $C \times C$ のサイズのユニット（集合を $B_\alpha$ で表す）に接続されているとする．ここで，$B_\alpha$ の $C^2$ 個のユニットのうち高々 1 つだけがオンとなることができ，どれか一つオンになるものがあれば上位層のユニットも自動的にオンになるという制約を導入する．このようにすると，これら $C^2 + 1$ 個のユニットの状態は，ちょうど $C^2 + 1$ 個[*16]で表せることになる．この制約の下で RBM のエネルギー関数は次のように表せる．

$$E(\mathbf{v}, \mathbf{h}) = -\sum_k \sum_{i,j} \left(h_{i,j}^k (W^k * \mathbf{v})_{i,j} + c_k h_{i,j}^k\right) - b\sum_{i,j} v_{i,j}$$
$$\text{subject to} \sum_{(i,j)\in B_\alpha} h_{i,j}^k \le 1, \ \forall k, \alpha \tag{56}$$

このエネルギー関数を用いて，入力層が与えられたときの隠れ層 $h_{i,j}^k$ およびプーリング層 $p_\alpha$ の条件付き確率および，隠れ層が与えられたときの出力層の条件付き確率を計算し，認識および学習に用いた．

[*16] つまり，下位層 $C^2$ 個のユニットのどれがオンであるか＋どれもオンでないの場合の数．上位層のユニットは前者でオン，後者でオフ．

## 5. ソフトウェア

### 5.1 Theano

Theano[*17]は Montreal 大学の LISA (Laboratoire d'Informatique des Systémes Adaptatifs) で開発された，大規模な数式計算を効率的に行うための Python ライブラリである [4]．Theano 自身はディープラーニングだけでなくその他の汎用的な大規模計算に対しても応用できるよう開発されているが，現状では主にディープラーニングのために用いられていると思われる．

Theano は関数型言語の概念を取り入れており，比較的小さなコードを与えれば, 高度に最適化がなされた CPU/GPU コードが自動的に生成される．自動的に BLAS や CUDA を用いた並列化がなされ，その上，数式の簡約や不要な計算の除去などのより高度な処理も自動的に行われる．また，GPU を使うか使わないかを簡単に選択できる．

反面，Theano は関数型言語の考え方を取り入れているため，事前知識がなければ使い始めるのに一定の努力が必要である．また，並列化を行いやすい処理に重点を置いており，従来の手続き型言語と比較して，書ける処理の範囲に制約がある（例えば，Theano では再帰関数による処理をサポートしていない)．そのため，記述が困難なアルゴリズムもある．

### 5.2 EBlearn

EBlearn[*18]は，主に CNN による学習・推論を行うためのライブラリで，C++で書かれている [56]．簡単に使えるインタフェースが用意されており，CNN の構成を記述した設定ファイルとデータセットを与えるだけで，EBlearn は自動的にこれらのデータから各パラメータを学習し，推論結果を返してくれる．

### 5.3 cuda-convnet

cuda-convnet[*19]は，CNN の学習・推論のためのライブラリで，ILSVRC2012 で優勝したシステム [28] を実装するのにも使われたということである．基本的な機能は EBlearn と同じだが，プーリングの種類が豊富で，使用する色空間を変更でき，また局所コントラスト正規化なども実装されているなど，いくつかの違いがある．

## 6. おわりに

本稿では，ディープラーニングの各方法について，なるべく網羅的に述べたつもりである．本稿冒頭に述べたように，ディープラーニングの方法は，各種のベンチマークテストやコンテストで著しい成果を挙げているものの，そこでは様々な方法が個別に試されている段階であると言え，それらの複数の要素の中で何が本質的に重要であるのかは，まだ明確ではない．例えば，現在のディープラーニングのブームのきっかけを作ったのは，間違いなく多層 NN の学習を可能にするプレトレーニングの方法が発見されたことであり，その重要性を説く論文はいくつもある [2]．しかしながら少なくとも画像認識においては，プレトレーニングを一切行わない CNN がベンチマークで最良の結果を残している [9]．一体プレトレーニングは必要なのだろうか？あるいは，過完備な特徴セットによるスパース表現こそが本質であるという研究（例えば [11]）がある一方，そうではないという主張もある．これらの他にも矛盾する主張はいくつか見られる．今後の研究が進むにつれ，今の混沌とした状況は徐々に整理されていくと予想される．これを進める研究が日本の研究コミュニティから生み出されることを期待して，本稿の結びとする．

[*17] http://deeplearning.net/software/theano/
[*18] http://eblearn.cs.nyu.edu:21991/
[*19] http://code.google.com/p/cuda-convnet/

### 参考文献

[1] Baldi, P. and Hornik, K.: Neural networks and principal component analysis: Learning from examples without local minima, *Neural Networks*, Vol. 2, pp. 53–58 (1989).
[2] Bengio, Y.: Learning Deep Architectures for AI, *Foundations and Trends in Machine Learning*, Vol. 2, No. 1, pp. 1–127 (2009).
[3] Bengio, Y., Lamblin, P., Popovici, D. and Larochelle, H.: Greedy Layer-Wise Training of Deep Networks, *NIPS* (2006).
[4] Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D. and Bengio, Y.: Theano: a CPU and GPU Math Expression Compiler, *Proceedings of the Python for Scientific Computing Conference (Scipy)* (2010).
[5] Bottou, L.: Online Learning and Stocastic Approximations, *Online Learning and Neural Networks*, Cambridge University Press (1998).
[6] Bouerau, Y.-L., Bach, F., LeCun, Y. and Ponce, J.: Learning Mid-Level Features For Recognition, *CVPR* (2010).
[7] Boureau, Y.-L., Ponce, J. and LeCun, Y.: A Theoretical Analysis of Feature Pooling in Visual Recognition, *ICML* (2010).
[8] Bourlard, H. and Kamp, Y.: Auto-association by multilayer perceptrons and singular value deocmposition, *Biological Cybernetics*, Vol. 59, pp. 291–294 (1988).
[9] Cireşan, D. C., Meier, U. and Schmidhuber, J.: Multi-column Deep Neural Networks for Image Classification, *CVPR* (2012).
[10] Cireşan, D. C., Meier, U., Masci, J., Gambardella, L. M. and Schmidhuber, J.: Flexible, High Performance Convolutional Neural Networks for Image Classification, *IJCAI* (2011).
[11] Coates, A., Lee, H. and Ng, A. Y.: An analysis of single-layer networks in unsupervised feature learning, *AISTATS* (2011).
[12] E. Hinton, G.: A Practical Guide to Training Restricted Boltzmann Machines, Technical report (2010).
[13] E. Hinton, G., Osindero, S. and Teh, Y.-W.: A fast learning algorithm for deep belief nets, *Neural Computation*,

Vol. 18, pp. 1527–1544 (2006).
[14] Eslami, S. M. A., Heess, N. and Winn, J.: The Shape Boltzmann Machine : a Strong Model of Object Shape, *CVPR*, Ieee, pp. 406–413 (2012).
[15] Fukushima, K. and Miyake, S.: Neocognitron: A New Algorithm for Pattern Recognition Tolerant of Deformations and Shifts in Position, *Pattern Recognition*, Vol. 15, pp. 455–469 (1982).
[16] Geisler, W. S. and Albrecht, D. G.: Cortical Neurons: Isolation of Contrast Gain Control, *Vision Research*, Vol. 32, No. 8, pp. 1409–1410 (1992).
[17] Hebb, D.: *The Organization of Behavior*, New York: Wiley (1949).
[18] Hinton, G. E. and Sejnowski, T. J.: Parallel distributed processing: explorations in the microstructure of cognition, vol. 1, MIT Press, chapter Learning and relearning in Boltzmann machines, pp. 282–317 (1986).
[19] Hinton, G. E.: Training products of experts by minimizing contrastive divergence., *Neural computation*, Vol. 14, No. 8, pp. 1771–800 (2002).
[20] Hinton, G. E., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A. and Vanhoucke, V.: Deep Neural Networks for Acoustic Modeling in Speech Recognition, *IEEE Signal Processing Magazine*, Vol. 29 (2012).
[21] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.: Improving neural networks by preventing co-adaptation of feature detectors, *CoRR*, Vol. abs/1207.0580 (2012).
[22] Hubel, D. H. and Wiesel, T. N.: Receptive fields, binocular interactions, and functional architecture in the cat's visual cortex, *Journal of Physiology*, Vol. 160, pp. 106–154 (1962).
[23] Hyvärinen, A. and Hoyer, P. O.: A two-layer sparse coding model learns simple and complex cell receptive fields and topography from natural images, *Vision Research*, Vol. 41, No. 18, pp. 2413–2423 (2001).
[24] Hyvärinen, A., Hurri, J. and Hoyer, P. O.: *Natural Image Statistics*, Springer (2009).
[25] Ito, M. and Komatsu, H.: Representation of Angles Embedded within Contour Stimuli in Area V2 of Macaque Monkeys, *The Journal of Neuroscience*, Vol. 24, No. 13, pp. 3313–3324 (2004).
[26] Jarrett, K., Kavukcuoglu, K., Ranzato, M. A. and LeCun, Y.: What is the Best Multi-Stage Architecture for Object Recognition, *ICCV* (2009).
[27] Kavukcuoglu, K., Ranzato, M. A., Fergus, R. and LeCun, Y.: Learning Invariant Features through Topographic Filter Maps, *CVPR* (2009).
[28] Krizhevsky, A., Sutskever, I. and Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks, *NIPS* (2012).
[29] Le, Q. V., Coates, A., Prochnow, B. and Ng, A. Y.: On Optimization Methods for Deep Learning, *ICML*, pp. 265–272 (online), available from ⟨http://ai.stanford.edu/ quocle/publications.html⟩ (2011).
[30] Le, Q. V., Karpenko, A., Ngiam, J. and Ng, A. Y.: ICA with Reconstruction Cost for Efficient Overcomplete Feature Learning, *NIPS* (2011).
[31] Le, Q. V., Ngiam, J., Chen, Z., Chia, D., Koh, P. W. and Ng, A. Y.: Tiled convolutional neural networks, *NIPS* (2010).
[32] Le, Q. V., Ranzato, M. A., Monga, R., Devin, M., Chen, K., Corrado, G. S., Dean, J. and Ng, A. Y.: Building High-level Features Using Large Scale Unsupervised Learning, *ICML* (2012).
[33] Lecun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. and Jackel, L. D.: Backpropagation Applied to Handwritten Zip Code Recognition, *Neural Computation*, Vol. 1, No. 4, pp. 541–551 (1989).
[34] LeCun, Y., Bottou, L., Orr, G. and Muller, K.: Efficient Backprop, *Neural Networks: Tricks of the trade*, Springer (1998).
[35] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P.: Gradient-Based Learning Applied to Document Recognition, *Proc. IEEE* (1998).
[36] Lee, B. H., Grosse, R., Ranganath, R. and Ng, A. Y.: Unsupervised learning of hierarchical representations with convolutional deep belief networks, *Communications of the ACM*, Vol. 54, No. 10, pp. 95–103 (2011).
[37] Lee, H., Ekanadham, C. and Ng, A. Y.: Sparse deep belief net model for visual area V2, *NIPS* (2008).
[38] Lee, H., Grosse, R., Ranganath, R. and Ng, A. Y.: Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations, *ICML* (2009).
[39] Lee, H., Grosse, R., Ranganath, R. and Ng, A. Y.: Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, *ICML*, No. 3, ACM Press (2009).
[40] Lowe, D. G.: Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91–110 (2004).
[41] Lyu, S. and Simoncelli, E. P.: Nonlinear image representation using divisive normalization, *CVPR* (2008).
[42] McCulloch, W. S. and Pitts, W.: A Logical Calculus of Ideas Immanent in Nervous Activity, *Bulletin of Mathematical Biophysics*, Vol. 5, No. 4, pp. 115–133 (1943).
[43] Olshausen, B. A. and Field, D. J.: Emeregnce of simple-cell receptive field properties by learning a sparse code for natural images, *Nature*, Vol. 381, pp. 607–609 (1996).
[44] Opper, M. and Saad, D.: From Naive Mean Field Theory to the TAP Equations, *Advanced Mean Field Methods: Theory and Practice*, The MIT Press, chapter 2 (2001).
[45] Pinto, N., Cox, D. D. and DiCarlo, J. J.: Why is Real-World Visual Object Recognition Hard?, *PLoS computational biology*, Vol. 4, No. 1, pp. 0151–0156 (2008).
[46] Ranzato, M. A., Bouerau, Y.-L. and LeCun, Y.: Sparse Feature Learning for Deep Belief Networks, *NIPS* (2007).
[47] Ranzato, M. A., Huang, F.-J., Boureau, Y.-L. and LeCun, Y.: Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition, *CVPR* (2007).
[48] Ranzato, M. A., Poultney, C., Chopra, S. and LeCun, Y.: Efficient Learning of Sparse Representation with an Energy-Based Model, *NIPS* (2006).
[49] Rosenblatt, F.: The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain, *Psychological Review*, Vol. 65, No. 6, pp. 386–408 (1958).
[50] Rumelhart, D. E. and Mcclelland, J.: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Cambridge: MIT Press (1986).
[51] Salakhutdinov, R. and E. Hinton, G.: Deep Boltzmann Machines, *AISTATS*, Vol. 5, No. 2, pp. 448–455 (2009).
[52] Saxe, A. M., Koh, P. W., Chen, Z., Bhand, M., Suresh, B. and Ng, A. Y.: On Random Weights and Unsupervised Feature Learning, *ICML* (2010).
[53] Scherer, D., Müller, A. and Behnke, S.: Evalutaion of

Pooling Operations in Convolutional Architectures for Object Recognition, *ICANN* (2010).
[54] Schwartz, O. and Simoncelli, E. P.: Natural signal statistics and sensory gain control, *Nature Neuroscience*, Vol. 4, No. 8, pp. 819–825 (2001).
[55] Sermanet, P., Chintala, S. and LeCun, Y.: Convolutional Neural Networks Applied to House Numbers Digit Classification, *ICPR* (2012).
[56] Sermanet, P., Koray Kavukcuoglu and LeCun, Y.: EBLearn : Open-Source Energy-Based Learning in C ++, *IEEE International Conference on Tools with Artificial Intelligence* (2009).
[57] Serre, T., Wolf, L. and Poggio, T.: Object Recognition with Features Inspired by Visual Cortex, *CVPR* (2005).
[58] Simard, P. Y., Steinkraus, D. and Platt, J.: Best Practice for Convolutional Neural Networks Applied to Visual Document Analysis, *ICDAR* (2003).
[59] Taylor, G. W., Fergus, R., LeCun, Y. and Bregler, C.: Convolutional Learning of Spatio-temporal Features, *ECCV* (2010).
[60] Teo, P. C. and Heeger, D. J.: Perceptual image distortion, *Proc. SPIE, Human Vision, Visual Processing, and Digital Display V*, Vol. 2179, pp. 127–141 (1994).
[61] Tieleman, T.: Training restricted Boltzmann machines using approximations to the likelihood gradient, *ICML*, pp. 1064–1071 (2008).
[62] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. and Manzagol, P.-A.: Stacked Denoising Autoencoders : Learning Useful Representations in a Deep Network with a Local Denoising Criterion, *Journal of Machine Learning Research*, Vol. 11, pp. 3371–3408 (2010).
[63] Watson, A. B. and Solomon, J. A.: Model of visual contrast gain control and pattern masking, *J. Opt. Soc. Am. A*, Vol. 14, No. 9, pp. 2379–2391 (1997).
[64] Yang, J., Yu, K., Gong, Y. and Huang, T.: Linear Spatial Pyramid Matching Using Sparse Coding for Image Classification, *CVPR* (2009).