National Institute of
Advanced Industrial Science
and Technology
AIST

大脳皮質のアルゴリズム BESOM Ver.2.0

産業技術総合研究所テクニカルレポート

AIST11-J00009

一杉裕志

産業技術総合研究所 ヒューマンライフテクノロジー研究部門

y-ichisugi@aist.go.jp

http://staff.aist.go.jp/y-ichisugi/j-index.html

2011 年 9 月 30 日

独立行政法人 産業技術総合研究所

概 要

脳の中で知能にもっとも深く関与する組織は大脳皮質である。筆者は、近年の計算論的神経科学の進展を踏まえて、大脳皮質の主要な機能を再現させる機械学習アルゴリズム BESOM を開発中である。

BESOM は、ベイジアンネットと自己組織化マップと独立成分分析という 3 つの情報処理技術を組み合わせて、大脳皮質の実行効率と汎化性能を再現させることを目指している。現在のところアルゴリズムは未完成だが、前回のバージョンにおける主要な問題がかなり解決したので、本文書で詳しく説明する。

間違いの指摘、質問、コメントなどを歓迎いたします。

目 次

第1章　はじめに

筆者は、近年における計算論的神経科学と機械学習技術の知見を踏まえて、大脳皮質の主要な機能と性能を計算機上で再現させることを目指している [2][3]。

本文書は、現状において解決された問題と未解決の問題を整理し、次期バージョンの BESOM の実装の足がかりにすることを目的としている。

現在筆者は次の３つを最優先に達成すべき目標として取り組んでいる。

1. **ベイジアンネット** [6] を核として複数の大脳皮質モデルを統合し、必要な機能を一通り持った機械学習アルゴリズムの形にする。具体的には、自己組織化マップ [7]、コラム構造・６層構造との対応がつく認識アルゴリズム [1] [5]、スパース符号化 [11]、独立成分分析（ＩＣＡ）[9]、部品別学習 [12]、注意の正規化モデル [14]、強化学習、時系列学習などを１つのアルゴリズムに統合する。
2. 大規模なネットワークを用いた実験に向けて、１ステップあたり平均 $O(n)$ 程度で動作する**スケーラブル**（大規模化可能）な認識・学習アルゴリズムを設計・実装する。
3. アルゴリズム「開発支援ツール」を開発する。様々な部分アルゴリズムと評価プログラムを、組み合わせやパラメタを変えて実験・評価する作業を、大幅に効率化・高信頼化する。

現時点でこれらの目標はいずれも完全には達成されていないが、前回のテクニカルレポート [3] から大きく進展し、主要な問題はほぼ解決したと考えている。重要な進展には以下のものがある。

1. 認識･学習アルゴリズムの理論的意味付け（2 章）。
2. より妥当と思われる条件付確率表のモデル（5 章）。
3. 他のアルゴリズムとの干渉の問題が解消された側抑制ＩＣＡアルゴリズム（8 章）。
4. 特徴選択に基づくベイジアンネットの構造学習アルゴリズム（9 章）。
5. Belief revision アルゴリズムの採用に向けたいくつかの問題の解決（10 章）。

これらの進展をすべて組み合わせた次期バージョンでは、BESOM の認識・学習アルゴリズムは $O(n)$ で動作し、実データに対する汎化能力の高い機械学習アルゴリズムになると期待している。

また、本文書では述べないが、 BESOM と強化学習との統合のアイデア、オンラインモデル選択のアイデア、より生産性の高い開発支援ツールのアイデアについてもかなり検討が進んでいる。

本文書を読んで、ベイジアンネットに基づく大脳皮質のモデルの有望さを理解し、同じ目的の研究に取り組み始める研究者が増えることを、引き続き期待する。

以後の章では、筆者がこれまでに公開した文章を適宜参照する。その際には、以下の表記を用いる。

- TR2008[2]
  大脳皮質のモデルを作ることにより、高い知能を持ったロボットを実現可能にする、全体構想を述べたテクニカルレポート。
- TR2009[3]
  BESOM アルゴリズムの 2009 年時点での状況を報告するテクニカルレポート。
- IJCNN2007[1]
  近似確率伝播アルゴリズムと大脳皮質の解剖学的構造との対応を述べたもの。
- ICONIP2010[4]
  ベイジアンネットを用いたスパース符号化について述べたもの。
- IJCNN2011[5]
  近似 belief revision の提案と性能評価。

謝辞： 科学技術振興機構／理化学研究所の 細谷晴夫氏には議論を通じ多くのアイデアをいただいております。感謝いたします。

第2章　ＢＥＳＯＭの動作の定式化

この章ではＢＥＳＯＭの動作の定式化について説明する。（内容は ICONIP2010[4] で書いたものと同じだがより詳しく説明する。）

2.1　学習の目的、認識ステップ、学習ステップ

パラメタ θ によって決まる、隠れ変数の値の組 $\mathbf{h}$ と観測変数の値の組 $\mathbf{i}$ との間の同時確率のモデルを $P(\mathbf{h},\mathbf{i}|\theta)$ とする。また、時刻 t における入力変数の値の組を $\mathbf{i}(t)$ とする。各時刻の入力は i.i.d. (独立同分布) に従うと仮定すると、θ のもとで入力データの列 $\mathbf{i}(1),\mathbf{i}(2),\cdots,\mathbf{i}(t)$ が生じる確率は以下のようになる。

$$\begin{aligned}&\prod_{i=1}^{t} P(\mathbf{i}(i)|\theta)\\ =\ &\prod_{i=1}^{t}\sum_{\mathbf{h}} P(\mathbf{h},\mathbf{i}(i)|\theta) \qquad (2.1)\end{aligned}$$

学習の目的は、以下のようにパラメタをＭＡＰ推定すること、すなわちパラメタ θ の事後確率を最大にすることである。

$$\theta^* \ =\ \operatorname*{argmax}_{\theta}\prod_{i=1}^{t}\sum_{\mathbf{h}} P(\mathbf{h},\mathbf{i}(i)|\theta)P(\theta) \quad (2.2)$$

本文書で述べる学習アルゴリズムは複雑だが、その本質は以下に述べるように、認識ステップと学習ステップの動作を表す２つの数式で書ける。

認識ステップでは、まず現在のパラメタ $\theta(t)$ に基づいて、与えられた入力 $\mathbf{i}(t)$ に対する隠れ変数の値の組の最大事後確率推定値 $\hat{\mathbf{h}}(t)$ （すなわち MPE, *most probable explanation*）を次のように求める。

$$\begin{aligned}\hat{\mathbf{h}}(t) \ &=\ \operatorname*{argmax}_{\mathbf{h}} P(\mathbf{h}|\mathbf{i}(t),\theta(t))\\ &=\ \operatorname*{argmax}_{\mathbf{h}} P(\mathbf{h},\mathbf{i}(t)|\theta(t))/P(\mathbf{i}(t))\\ &=\ \operatorname*{argmax}_{\mathbf{h}} P(\mathbf{h},\mathbf{i}(t)|\theta(t)) \qquad (2.3)\end{aligned}$$

次に学習ステップでは、式 (2.2) における隠れ変数の周辺化を推定値 $\hat{\mathbf{h}}(i)$ を用いることで近似してパラメタ推定し、結果を $\theta(t+1)$ とする。

$$\theta(t+1) = \operatorname*{argmax}_{\theta}\prod_{i=1}^{t} P(\hat{\mathbf{h}}(i),\mathbf{i}(i)|\theta)P(\theta) \qquad (2.4)$$

学習ステップでは、確率的勾配法とは違って、過去の経験をすべて踏まえてパラメタを推定しなおしていることに注目されたい。もし式 (2.4) が高い近似精度で実行できるならば、勾配法よりも高いオンライン性能が実現できる可能性がある。もしそれが可能ならば、生まれてからあらゆる瞬間に最適な行動をしなければならない生物に適したアルゴリズムであると言える。

式 (2.3) と式 (2.4) を厳密に実行するには多くの計算量を必要とする。しかし実際の脳は認識と学習を少ない計算量で近似的に実現しているはずである。また、妥当な脳のモデルであるためには、オンラインで実行可能でなければならない。すなわち、$\mathbf{i}(t),\hat{\mathbf{h}}(t),\theta(t)$ のみを使って $\theta(t+1)$ が計算できなければならない。実は、同時確率のモデル $P(\mathbf{h},\mathbf{i}|\theta)$ がベイジアンネットのとき、認識も学習も極めて効率的に近似実行できる。そこが、本定式化に基づく BESOM アルゴリズムが有望だと筆者が考えるもっとも大きな理由である。認識ステップは belief revision アルゴリズム（10 章参照）、学習ステップは（パラメタの事前分布 $P(\theta)$ を無視すれば）カウンティング（2.2 節参照）と呼ばれる方法で非常に簡単に実現できる。

式 (2.4) で、隠れ変数の周辺化を推定値 $\hat{\mathbf{h}}(i)$ を用いることで近似しているが、この近似の妥当性は自明ではなく、実データに学習アルゴリズムを適用して妥当性を検証する必要があるだろう。なお、近傍学習（4 章）には、おそらくこの近似を「補正」する効果がある。

$P(\theta)$ はパラメタの事前分布だが、これは生物が進化によって獲得した、外界に関する生得的知識を意味している。本稿で述べるアルゴリズムには明示的に $P(\theta)$ は出てこないが、近傍関数（4 章）や特徴選択アルゴリズム（9 章）の形で学習アルゴリズムに作り込まれている。

なお、定式化の各構成要素と、大脳皮質の構成要素との対応をまとめると図 2.1 のようになる。

学習の目的：

$$\theta^* = \mathrm{argmax}_{\theta} \prod_{i=1}^{t} \sum_{\mathbf{h}} P(\mathbf{h}, \mathbf{i}(i)|\theta) P(\theta)$$

認識ステップ：

$$\hat{\mathbf{h}}(t) = \mathrm{argmax}_{\mathbf{h}} P(\mathbf{h}, \mathbf{i}(t)|\theta(t))$$

学習ステップ：

$$\theta(t+1) = \mathrm{argmax}_{\theta} \prod_{i=1}^{t} P(\hat{\mathbf{h}}(i), \mathbf{i}(i)|\theta) P(\theta)$$

$P(\mathbf{h}, \mathbf{i}|\theta)$ ： 進化によって獲得された同時確率のモデル。大脳皮質の構造。
$P(\theta)$ ： 進化により獲得された、シナプスの重みに関する生得的知識。
$\mathbf{i}(t)$ ： 一次感覚野への入力。
$\hat{\mathbf{h}}(t)$ ： 連合野の全コラムの活動状態。
$\theta(t)$ ： 全可変シナプスの重み。

図 2.1: BESOM の動作の定式化と、その各構成要素の大脳皮質の構成要素との対応。

また、ＢＥＳＯＭの動作を模式図を使って説明すると図 2.2 のようになる。

2.2 カウンティングによる条件付確率表の学習

観測データにおいて $U = u$ である回数を N_u、$U = u$ かつ $X = x$ である回数を N_x とすると、条件付確率 $w = P(x|u)$ の最尤推定量 $\hat{w}$ は

$$\hat{w} = \frac{N_x}{N_u} \tag{2.5}$$

となることが知られている[1]。つまり、変数が特定の値になった場合の数を数えて割り算するだけでよい。このパラメタ推定方法は**カウンティング**と呼ばれる。

カウンティングを用いれば式 (2.4) の学習ステップは高速に実行可能だが、大規模なベイジアンネットにおいては、別の問題が生じる。**条件付確率表** (CPT, conditional probability table) $P(X|U_1, \cdots, U_m)$ は一般には親ノードの数 m に対し $O(2^m)$ の記憶域を必要とし、ＭＰＥ計算も $O(2^m)$ の計算量を必要とする。これでは効率が悪すぎる上、パラメタ推定が過適合を起こしやすい。

そこで、より少ないパラメタで条件付確率表を近似表現するパラメトリックモデルがよく使われる。（例えば noisy-OR モデル [6] など。）

今、条件付確率表 $P(X|U_1, \cdots, U_m)$ を、X と U_i の値の組ごとにあるパラメタ $\theta(x, u_i)$ の関数で表現することを考える。（X も U_i 値が s 個あるとすれば、パラメタの数は全部で s^2m 個である。）条件付確率 $P(x|u_1, \cdots, u_m)$ は、m 個のパラメタ $w_1, \cdots, w_m$ の関数で表現する。ただし、 $w_i = \theta(x, u_i)$ とする。

$$P(x|u_1, \cdots, u_m) = f(w_1, \cdots, w_m) \tag{2.6}$$

この時、与えられた確率変数 $X, U_1, \cdots, U_m$ の観測データの組から、事後確率最大となるパラメタを推定したい。（ここでは隠れ変数は存在せずすべてが観測変数であると仮定する。）このような場合、（確率的）最急降下法が使われることが多いが、我々は、より効率的な別のアプローチを取る。

パラメタ w_i が自由な値を取るのではなく、２つの値の間の条件付確率 $p(x|u_i)$ と必ず一致する、という制約条件を付けたモデルを採用する。

$$P(x|u_1, \cdots, u_m) = f(w_1, \cdots, w_m)$$
$$\text{ただし } w_i = P(x|u_i) \tag{2.7}$$

このような制約付きのパラメタであれば、カウンティングを用いることで、最尤推定は非常に簡単に行える。

カウンティングは単に計算機上での実装が簡単なだけでなく、生物学的に妥当な神経回路によっても、オンライン学習アルゴリズムとして容易に実現可能である。(IJCNN2007[1] または TR2008[2]p.64 または TR2009[3]p.11 参照)。すると、式 (2.4) の学習ステップもまた、（$P(\theta)$ を無視すれば）神経回路で容易に実現可能ということになる。このことから、式 (2.7) は大脳皮質が採用するＣＰＴモデルの有望な候補と言えるだろう。

制約付きのＣＰＴモデル（式 (2.7)）のパラメタ推定値は、制約のないＣＰＴモデル（式 (2.6)）でのパラメタ推定値と比べて一般に尤度は低いものになるはずである。つまり、少ない計算量と引き換えにフィッティングが悪くなる。しかし、実はフィッティングの悪さ

[1]例えば「エージェントアプローチ 人工知能 第２版」p.723 参照。

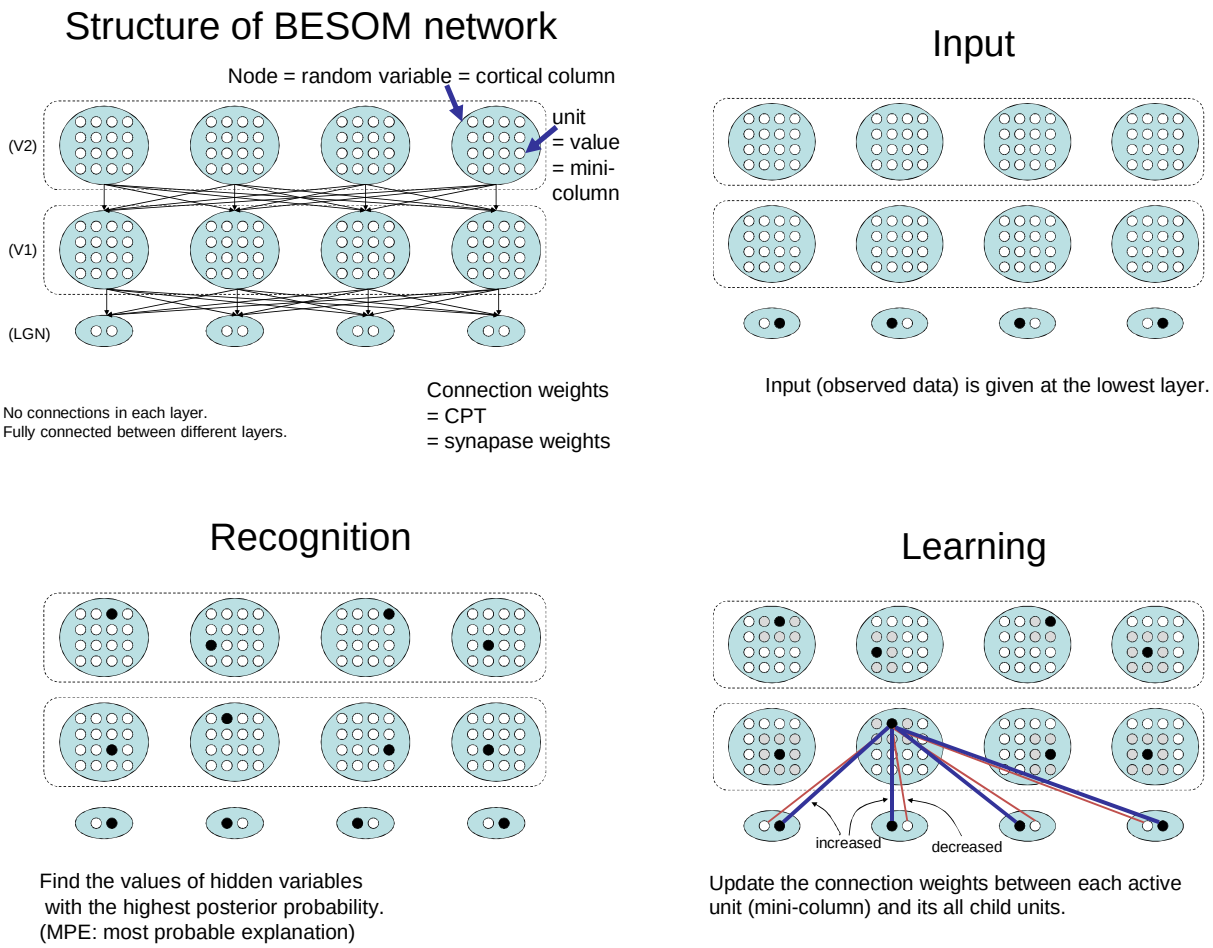

図 2.2: ＢＥＳＯＭの認識ステップと学習ステップの振る舞い。
（左上）ベイジアンネットワークの構造。大きな丸はノード、小さな白い丸はユニットを表す。ネットワークは階層構造をしており、同一層内のノード間にはエッジはない。異なる層に属するノード間には、エッジがあり得る。
（右上）入力。観測データは再下端のノードの値として与えられる。
（左下）認識ステップ。ネットワーク全体がベイジアンネットとして動作し、観測データとの同時確率が最大となる隠れ変数の値の組み合わせ（ＭＰＥ）が計算される。
（右下）学習ステップ。すべての隠れノードがＳＯＭ（自己組織化マップ）[7] として動作し、勝者ユニット（黒丸）と子ノードのユニットとの間の結合の重みが更新される。具体的には、子ノードが勝者ユニットであれば重みは増やされ（太い青線）、そうでなければ重みは減らされる（細い赤線）。これは２ノード間の条件付確率表の要素の更新を意味する。隠れノードの勝者ユニットの近傍のユニット（灰色の丸）も同様に結合を少し更新することにより、近傍学習が行われる。勝者ユニットの近傍にないユニット（白丸）は、子ノードのユニットとの間の重みを更新しない。

は必ずもデメリットにはならない。制約条件が真の生成モデルの特徴に近い場合は、制約条件がパラメタに関する事前知識を与えていることになり、過適合がなくなって汎化能力が向上する可能性がある。

式 (2.7) における関数 f の選び方によって、認識アルゴリズムの効率や学習の汎化能力が決まる。筆者は IJCNN2007[1], ICONIP2010[4], IJCNN2011[5] では f として単純な算術平均のモデル（正規化係数を無視するなら線形和モデル、後述の式 (5.1)）を用いた。TR2009[3] では少し複雑な、あまり根拠のない関数を用いた。本文書では、より外界の性質に近く、認識アルゴリズムの効率も悪くない関数の１つの候補として meanOR model と呼ぶものを用いている。これについては 5 章で述べる。

2.3 定式化の意義と今後

本章で述べた学習の目的、認識ステップ、学習ステップの定式化は、もしそれが正しいならば、「大脳皮質の基本原理」と呼ぶべきものである。大脳皮質のあらゆる機能が、この基本原理に基づいて説明できると考えている。しかし、基本原理の定式化で、脳の解明が完了するわけではない。むしろ、脳の解明の出発点と考えるべきである。

本章で述べた基本原理は、飛行の原理に例えれば、「揚力が機体の重さを上回れば飛べる」という程度のことを言っているにすぎない。それだけではまだ飛行機は作れない。しかし、飛行の原理は、飛行機を作るための重要な設計の指針を与えてくれる。大きな揚力と軽い機体を実現すればよいのである。それには大変な努力が必要だろうが、「鳥の羽毛には飛ぶための神秘的な力があるに違いない」と信じている状態に比べれば、飛躍的な進歩になる[2]。

本章の基本原理は現時点では仮説にすぎないが、それが正しいことを前提にすれば、人間のような高い知能を実現するための重要な設計の指針を与えてくれる。この指針に従って、認識ステップと学習ステップを効率的に実行し、高い汎化能力を実現するための様々な機構が、この後の章で述べられていく。また、今後も新たな工夫が追加されていくだろう。

[2]羽毛に神秘的な力があると信じている人は今日ではいないだろうが、ニューロンにはシリコンやその他の機械にない神秘的な力があると信じている人は多いのではないだろうか。神経科学的には、そのような神秘的な力が存在する気配は見つかっていない。

第3章　認識アルゴリズム

この章ではＭＰＥ計算を行う認識ステップのアルゴリズムと、認識結果に制約条件を与える方法について述べる。

3.1　ＭＰＥ

ＢＥＳＯＭの認識ステップでは、ＭＰＥを計算する。

MPE (most probable explanation) とは、ベイジアンネットにおいて、与えられた観測データを最もうまく説明する変数の値の組のことである。観測データを表す確率変数の組を $\mathbf{i}$ 、隠れ変数（観測データ以外の確率変数）の値の組を $\mathbf{h}$ とすると、MPE となる値の組 $\hat{\mathbf{h}}$ は次の式で与えられる。

$$\begin{aligned}\hat{\mathbf{h}} &= \operatorname*{argmax}_{\mathbf{h}} P(\mathbf{h}|\mathbf{i}) \\ &= \operatorname*{argmax}_{\mathbf{h}} P(\mathbf{h},\mathbf{i})/P(\mathbf{i}) \\ &= \operatorname*{argmax}_{\mathbf{h}} P(\mathbf{h},\mathbf{i}) \qquad (3.1)\end{aligned}$$

ただし $P(\mathbf{h},\mathbf{i})$ は $\mathbf{h}$ と $\mathbf{i}$ との同時確率で、ベイジアンネットが与えられていれば、以下の式で計算できる。

$$\begin{aligned}P(\mathbf{h},\mathbf{i}) &= \prod_{x\in\mathbf{x}} P(x|parents(x)) \\ \text{ただし } \mathbf{x} &= \mathbf{h}\cup\mathbf{i} \qquad (3.2)\end{aligned}$$

ここで $parents(x)$ はノード X の親ノードの値の組である。

3.2　山登り法によるＭＰＥ計算アルゴリズム

図 3.1 はベイジアンネットのＭＰＥの近似解を山登り法を用いて求めるアルゴリズムである。これは TR2009[3] で用いたものと同じものである。本文書での実験も、特に断りのない限り、このアルゴリズムを用いている。

1. すべての隠れノードの値を何らかの値で初期化する。その時の値の組を $\mathbf{h}$ とする。
2. $\mathbf{h}$ の中の高々１つの隠れノードの値を別の値に変更したものを次の状態の候補とする。候補の集合を $\mathbf{H}$ とすると、$\mathbf{H}$ の要素のうち、入力 $\mathbf{i}$ との同時確率が最大のものを $\mathbf{h}'$ とする。
$$\mathbf{h}' = \operatorname*{argmax}_{\mathbf{h}^*\in\mathbf{H}} P(\mathbf{h}^*,\mathbf{i})$$
3. $P(\mathbf{h}',\mathbf{i}) > P(\mathbf{h},\mathbf{i})$ 、すなわち同時確率が大きくなっていれば、 $\mathbf{h} := \mathbf{h}'$ として 2. に戻る。大きくなっていなければ終了。

図 3.1: 素朴な山登り法によってＭＰＥの近似解を求めるアルゴリズム。

3.3　値の組に対する制約条件

本文書では用いるアルゴリズムでは、認識結果（ＭＰＥで求める値の組）に対してスパース性などの制約条件を加えている。この拡張は、以下で述べる方法によって、ベイジアンネットの枠組みを逸脱せずに行える。

図 3.2 のように、全てのノードの共通の子ノードとして制約ノード R が存在すると考える。R は、ＭＰＥの値に対して制約を与える２値の確率変数である。R の条件付確率表は下記のように定義される。

$$P(R=1|\mathbf{h}\cup\mathbf{i}) = \frac{1}{Z}\prod_{x\in\mathbf{x}} R(x,\mathbf{x}) \qquad (3.3)$$

ＭＰＥ計算時において、 $R=1$ は観測値として与えられていると解釈する。Z は正規化定数だが $\operatorname*{argmax}_{\mathbf{h}} P(\mathbf{h},\mathbf{i})$ の値には影響を与えないので、ＭＰＥ計算においては無視することができる。$R(x,\mathbf{x})$ についてはすぐ後で述べる。

同時確率の計算式（式 (3.2)）は、制約ノード R を追加したネットワークに対しては、以下のようになる。

$$P(\mathbf{h},\mathbf{i}) = \left(\frac{1}{Z}\prod_{x\in\mathbf{x}} R(x,\mathbf{x})\right)\prod_{x\in\mathbf{x}} P(x|parents(x))$$

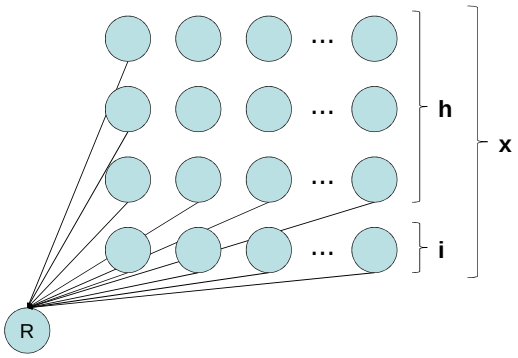

図 3.2: 全てのノードの共通の子ノードとして制約ノード R が存在すると解釈する。$\mathbf{x}$ は隠れノードの値の集合 $\mathbf{h}$ と入力ノードの値の集合 $\mathbf{i}$ の和集合。

$$= \frac{1}{Z}\prod_{x\in\mathbf{x}} R(x,\mathbf{x})\ P(x|parents(x))$$

$$\text{ただし } \mathbf{x} = \mathbf{h}\cup\mathbf{i} \tag{3.4}$$

ただし、$R(x,\mathbf{x})$ はノード X の状態 x に対して与えられるペナルティであり、具体的には式 (4.13)、式 (7.1)、式 (8.5) で定義される値の積である。

$$R(x,\mathbf{x}) = e^{-CU(x)}e^{-\lambda A(x)}e^{-\beta S(x,\mathbf{x})} \tag{3.5}$$

このように制約ノード R を導入することで、ベイジアンネットの枠組みを壊さずにスパース符号化（7 章）や側抑制ＩＣＡ（8 章）の機構が実現できるようになる。この方法には多くの利点がある。様々な機構が１つの統一された目的関数の最適化問題として定式化されるので、各機構の間の干渉が起きにくいと期待できる。また、個々の機構で必要となるパラメタ（たとえば側抑制ＩＣＡの側抑制シナプス）の学習方法も定式化に基づいて素直に決まるので、調整が必要な自由パラメタの数が少なくて済む。また、 belief revision アルゴリズムを用いた認識ステップにペナルティを導入する方法も、素直に導くことができる（10 章）。

なお、$R(x,\mathbf{x})$ はいわゆる正則化項ではなく、モデルの一部である点に注意されたい[1]。制約条件は、式 (2.2) で言えば同時確率のモデル $P(\mathbf{h},\mathbf{i}|\theta)$ の一部であり、パラメタの事前分布 $P(\theta)$ の一部ではない。$R(x,\mathbf{x})$ はパラメタ θ の学習には直接は影響を与えず、認識結果を通してのみ、θ に影響を与える。

[1]TR2009[3] ではスパース符号化や側抑制ＩＣＡの機構を正則化と呼んでいたが、間違いである。ただし、事前知識を使ってパラメタの自由度を下げ、過適合を避けるという効果は、結局は同じである。

3.4 将来のバージョンにおける認識ステップ

将来は認識ステップにおいて belief revision アルゴリズム [6][5] を用い、特徴選択の機構と組み合わせて、１ステップあたり $O(n)$ で動作させることを目指す。詳しくは 10 章で述べる。

第4章　学習アルゴリズム

この章では、本文書でのシミュレーションで使われている学習アルゴリズムの詳細について述べる。この学習アルゴリズムは、2 章の学習ステップの式 (2.4) の近似になっていると考えている。

4.1　学習則

この節で述べる学習則は TR2009[3] で述べたものとほぼ同じだが、近傍関数は正規化しないという点が修正されている。

まず、全てのノード（確率変数）は $s+1$ 個の値のうちのどれかをとるものとする。例えば、ノード X は、x_ϕ か $x_i,\ i=1,\cdots,s$ の値のどれかを取る。

$$X \in \{x_\phi, x_1, x_2, \cdots, x_{s-1}, x_s\} \tag{4.1}$$

（ノードごとに s の値が異なっていてもよいのだが、簡単のため、本稿ではすべてのノードで s は同一とする。）以下、 値 x_ϕ のことを **ϕ 値**、x_ϕ 以外の値のことを**非 ϕ 値**と呼ぶ。

認識ステップで MPE が求まると、学習ステップではそれを用いて結合の重みを更新する。隠れ層のノードにおいては、 MPE の値を表すユニットが**競合学習**の勝者になったと見なし、近傍のユニットとともに**近傍学習** [7] を行う。学習の目標値となる入力ベクトルは、子ノードにおける MPE をぼかしたベクトルになる。以下に、より具体的に説明する。

入力ノードにおいては観測値を表す値、隠れノードにおいては MPE として推定された値を表すユニットを、以下では**勝者ユニット** と呼ぶ。また、ノード X （隠れノード）が子ノード（入力ノード）$Y_l\ (l=1,\cdots,n)$ を持つとする。

ユニット x_i と y_j^l の間の結合の重み w_{ij}^l は、下記の式により更新する。

1. $i=\phi$ の場合：

   学習しない。（$w_{\phi j}^l$ は定義されない。5.2 節参照。）

2. $i\neq\phi, j\neq\phi$ の場合：

$$w_{ij}^l \leftarrow w_{ij}^l + n'(\alpha,i)(v_j^l - w_{ij}^l) \tag{4.2}$$

   n', v_j^l についてはすぐあとで説明する。α は学習率[1]である。

3. $i\neq\phi, j=\phi$ の場合：

$$w_{i\phi}^l = 1 - \sum_{j=1}^{s} w_{ij}^l \tag{4.3}$$

以上のアルゴリズムにより結合の重み w_{ij}^l が学習される。

次に、ユニットと勝者ユニットとの距離について定義しておく。d_{x_i} はノード X における勝者ユニットとユニット x_i の間の距離、$d_{y_j^l}$ はノード Y_l における勝者ユニットとユニット y_j^l の間の距離とする。ノード X と Y_l の勝者ユニットをそれぞれ x_{w_x}, $y_{w_y}^l$ $(w_x, w_y \in \{1,\cdots,s\})$ とすると、d_{x_i}, $d_{y_j^l}$ は下記のように定義される。

$$d_{x_i} = |i - w_x| \tag{4.4}$$

$$d_{y_j^l} = |j - w_y| \tag{4.5}$$

関数 n' は、従来の SOM の意味での**近傍関数** n を拡張したもので、以下のように定義される[2]。なお、近傍関数 n は必ず最大値が 1 であるものとする。

$$n'(\alpha,i) = \begin{cases} 0 & (i=\phi) \\ \alpha n(\alpha, d_{x_i}) & (i\neq\phi) \end{cases} \tag{4.6}$$

子ノード Y_l からの入力ベクトルの要素 v_j^l は、ユニット y_j^l が勝者ユニットであれば 1 そうでなければ 0 となるベクトルを **ぼかし関数** b でぼかしたものである。（なお、 v_j^l は $j\in\{1,\cdots s\}$ に対してのみ定義され、v_ϕ^l は定義されない。）

$$v_j^l = \begin{cases} 0 & (Y_l \text{ の勝者が}\varphi\text{値の時}) \\ \frac{1}{Z_b} b(\alpha, d_{x_i}, d_{y_j^l}) & (Y_l \text{ の勝者が非}\varphi\text{値の時}) \end{cases} \tag{4.7}$$

[1] 本来はユニットごとに学習率を持たせるべきだが、現在はグローバルに 1 つですませている。

[2] $i=\phi$ の場合の値は、現在の学習則では結局使われない。

正規化定数 Z_b は、ぼかし関数の値の総和を１にするためのものである。

$$Z_b = \sum_{y\in\{y_1,\cdots,y_s\}} b(\alpha, d_x, d_y) \tag{4.8}$$

α は学習率だが、同時に、ぼかし半径と近傍半径を決めるパラメタである。学習が進むにつれ α を 0 に近づける。学習率 α はグローバルに１つだけ存在し、t 回目の入力に対し、下記の式で値が決定される。

$$\alpha = 1/(0.001t + 1) \tag{4.9}$$

本章以降の実験における近傍学習では、特に断りのない限り、ぼかし関数 b と 近傍関数 n は以下のものを使っている。

$$b^{SmoothStep}(\alpha, d_x, d_y) = smoothStep(d_y, C_1\alpha + C_2) \tag{4.10}$$

$$n^{SmoothStep}(\alpha, d_x) = smoothStep(d_x, C_1\alpha + C_2) \tag{4.11}$$

$$smoothStep(d, r) = \begin{cases} 1 & (d < 1 \text{ or } d < r - 1) \\ r - d & (r - 1 \le d < r) \\ 0 & (r \le d) \end{cases} \tag{4.12}$$

パラメタの値は $C_1 = s + 1, C_2 = 2$ としている。ただし、 s はノード内の非φ値ユニットの数である。

4.2 ユニット活性度一律化

各ノードにおいて、各ユニットが勝者になる頻度が等しくなるような機構を追加すると、獲得するマップが滑らかになるなどの利点がある。筆者は TR2009[3] の 7.3.4 節[3]でこれを実現するアイデアのみを書いたが、今回はこれを実装した。

実現のアイデアは、高い勝者率のユニットに対して勝者率を下げるようにペナルティを与えるというものである。

ペナルティは、ＭＰＥ計算において、以下の式によって与えられる。（$R(x, \mathbf{x})$ については式 (3.5) 参照。）

$$R(x, \mathbf{x}) \propto e^{-CU(x)} \tag{4.13}$$

C はペナルティの強さを決める定数で、本文書では、すべての実験で C=100 とした。

[3]そこでは「非φ値の等確率の制約」と呼んでいた。

$U(x)$ は以下のように定義される。

$$U(x) = \begin{cases} 0 & (x \in \mathbf{i} \; or \; x = x_\phi) \\ \hat{P}(X = x|X \neq x_\phi) - \frac{1}{s} & (x \in \mathbf{h}, x \neq x_\phi) \end{cases} \tag{4.14}$$

$\hat{P}(X = x|X \neq x_\phi)$ は $P(X = x|X \neq x_\phi)$ の推定値で、条件付確率表と同じように、カウンティングによってオンライン学習する[4]。ただし s は非φ値ユニットの数である。

4.3 勝者ノイズ

一般にオンライン学習は局所解に陥る可能性があり、それを避ける１つのヒューリスティックスとして、ノイズを用いる方法がある。現在の実装では、認識結果にノイズを加える機構を用いており、これを**勝者ノイズ**と呼んでいる。

具体的にはＭＰＥ計算後、各ノードに対して 0.1α（α は学習率）の確率で、勝者ユニットをランダムな値に変更する。（変更後は、等確率で $s + 1$ 個の値のどれかになる。）

[4]推定値 $\hat{P}(X = x|X \neq x_\phi)$ は観測データ $\mathbf{i}(t)$ が与えられるたびに変化するので、本来なら添え字 t を書いて $\hat{P}_t(X = x|X \neq x_\phi)$ とでもすべきだが、ここでは省略している。以下の章でもいくつかの条件付確率の推定値を学習に用いているが、同様に t を省略している。

第5章　条件付確率表のモデル

この章では筆者が従来からしばしば仮定してきた linear-sum model に代わって、計算論的・神経科学的により有望と思われる meanOR model を提案する。

5.1　背景

2.2 節で述べたように、条件付確率表 $P(X|U_1,...,U_m)$ を少ないパラメタの関数で表現する「条件付確率表のモデル」の決め方によって、認識アルゴリズムの効率や学習の汎化能力が決まる。

筆者は以前から式 (5.1) の “linear-sum model” をしばしば仮定してきた。(ただし、$1/m$ は確率の総和を1にするための正規化係数だが、正規化をしなくても最終的に得られる事後確率やＭＰＥの値には影響はないので、アルゴリズムの実装時は無視しても構わない[1]。)

$$P(x|u_1,\cdots,u_m)=\frac{1}{m}\sum_{i=1}^{m}P(x|u_i) \qquad (5.1)$$

このモデルには下記の利点と欠点がある。

利点：

1. 演算が単純である。
2. これを仮定して導いた近似確率伝播アルゴリズムが大脳皮質のコラム構造・6層構造とよく一致する（IJCNN2007[1])。
3. スパース符号化モデル [11] の線形和モデルと近いので、一次視覚野の特性の再現に向いているように思われる。
4. 子ノードの値は定性的に親ノードの値のＯＲで決まるので、noisy-OR model [6] と同様に、外界を比較的自然に表現できると期待できる。

欠点：

1. 生成モデルとして見た時、不活性な親ノードであっても、子ノードの確率分布への影響がある。(混合分布を表現する際のモジュラリティが不完全。)
2. BESOM が仮定する「φ値ユニット」が神経科学的に見つかっていない理由を説明できない。
3. 「φ値ユニット」が持つ重み $w_\phi j$ は0に近い値になり、高い精度で学習するのが難しい。

以上の利点を保存しつつ、欠点を解決するＣＰＴモデルの候補の1つとして、meanOR モデルを提案する。

5.2　meanOR model

meanOR model は以下のように定義される。

$$\begin{aligned} P(x|u_1,\cdots,u_m) &= \frac{1}{Z}\sum_k w(x,u_k) \\ Z &= \sum_x P(x|u_1,\cdots,u_m) \\ w(x,u) &= \begin{cases} 0 & (u=u_\phi) \\ P(x|u) & (u\neq u_\phi) \end{cases} \end{aligned}$$

$1/Z$ は条件付確率表の正規化定数だが、linear-sum model の時と同様、値を変えても認識結果には影響はないので、以下の章および実装では $Z=1$ として正規化定数を無視する。

meanOR モデルは、上で述べた linear-sum model の利点を保存しつつ、欠点を「かなり」なくしたモデルになっている。

meanOR model を用いても解剖学的に妥当な近似確率伝播アルゴリズムが導けることは 10.2 節で示す。

このモデルではφ値ユニットの条件付確率 $P(x|u_\phi)$ を使わないので、φ値ユニットが重みを学習する必要がない。ただし、φ値ユニットから親ノード・子ノードに送るメッセージの計算は依然として必要になる。これについては 10.2.5 節で考察する。

[1]筆者はこのことに長い間気付かず、IJCNN2007[1] で述べた近似確率伝播アルゴリズムは不完全で修正が必要だと考えていたが、修正は不要だったようである。近似確率伝播アルゴリズムの動作確認は行っていないが、近似 belief revision アルゴリズムに関しては、$1/m$ を無視した実装が正しく動作することを確認した(IJCNN2011[5])。

第6章　入力の与え方と条件付確率表の可視化

ここでは、以降の章の実験で使われる、入力データの与え方と条件付確率表の可視化の方法[1]を説明する。

同じ方法は、ICONIP2010[4] および IJCNN2011[5] でも用いている。

6.1　背景

BESOM の基本動作を確認する時、入力データと学習結果が直感的に分かりやすく可視化できると便利である。そのため、動作テストには画像データの学習が適している。また、画像データは２値画像よりも、グレイスケール画像の方が、より現実の脳への入力に近いし、パターン認識の応用にも近い。

しかし、 BESOM は離散値のベイジアンネットなので、グレイスケール画像の情報をなんらかの方法で離散値に変換する必要がある。

TR2009[3] では、入力ノードを多値の確率変数にして、入力画像の１ピクセルを１つの入力ノードに対応付けていた。しかし多値の入力ノードを使う方法では、学習結果の条件付確率表を直接的に画像と対応付けて可視化することはできず、学習結果が正しいかどうかを把握しづらかった。

本章で述べる入力方法と可視化方法は、入力ノードに２値の確率変数を用いており、学習結果が素直に可視化できるという特徴を持つ。

6.2　入力の方法

本文書で述べるすべての実験は、隠れ層と入力層からなる２層 BESOM ネットを用いている。入力画像は、 12x12 =144 ピクセルのグレイスケール画像である。入力層には 144 個の入力ノードがある。入力ノードはすべて２値の確率変数である。

認識ステップでは、入力画像の各ピクセルの濃さに応じた確率で、ランダムに０か１の値を決め、144 個の２値入力ノードに観測値として与える。白いピクセルは確率１、黒いピクセルでは確率０、灰色はその中間の確率で、入力値が１になる。

6.3　可視化の方法

入力ノードを $Y_l \in \{0,1\}$、$(l = 1, \cdots, 144)$ とする。

隠れ層のノード X の各値 x ごとに、144 個の条件付確率 $P(Y_l = 1|X = x)$、$(l = 1, \cdots, 144)$ の値を、１は白、０は黒の濃淡として、１つの**基底画像**として可視化する。

隠れ層のノードの数が n 、各ノードのユニット数が s+1 ならば、n(s+1) 個の基底画像が学習されることになる。

なお、本文書の学習アルゴリズムではφ値ユニットは学習を行う必要がないが、TR2009[3] での実装を引き継いでいる理由からφ値ユニットは入力の平均画像を学習しており、以下の章の図ではそれが可視化されている。

6.4　入力の与え方の今後

この章で述べた入力方法は、生物学的に妥当だろうか。例えばもし、脳がＭＣＭＣをやっていて、ニューロンのパルスの有無が確率変数の離散的状態を表しているとすれば、この章の入力方法は比較的自然に思われる。

が、本文書で述べるバージョンの BESOM モデルではあくまでパルスの頻度がアナログ量を表現しているという解釈に立っており、この解釈のもとでは、大脳皮質への入力もアナログ量であるべきである。

入力がアナログ量だとすると、今後採用する予定の belief revision アルゴリズムを用いた認識ステップにどのようにアナログ量を入力すべきかを決める必要がある。この問題についての今後の方針については、10.5 節で述べる。

[1] この方法は細谷晴夫氏のアイデアである。

第7章　スパース符号化

この章ではスパース符号化の方法と、自然画像の学習結果について述べる。

7.1　背景

ICONIP2010[4] の論文で、スパース符号化を用いた自然画像の学習の結果を示したが、方位選択性は見られるものの、Ｖ１の基底画像とはあまり似ていなかった。その後、画像の前処理の仕方を調整するなどして、Ｖ１と比較的似ている基底画像が得られたので以下に報告する。

7.2　活性ノードに対するペナルティ

スパース符号化の方法は、基本的に TR2009[3] で述べたものと同じである。ただしＣＰＴモデルは 5 章で述べた meanOR model を用いている。(ICONIP2010[4], IJCNN2011[5] では linear-sum model を用いていた。)

式 (3.5) で定義したように、各ノードにはそれが活性ノードであった時に、同時確率の値にペナルティを与えるようにする。

$$R(x, \mathbf{x}) \propto e^{-\beta A(x)} \tag{7.1}$$

ただし、β はスパース性を制御するパラメタ、$A(x)$ は x がφ値のときに０、非φ値の時に１になる関数である。

$$A(x_i) = \begin{cases} 0 & (i = \phi) \\ 1 & (i \neq \phi) \end{cases} \tag{7.2}$$

本章のＢＥＳＯＭによるスパース符号化の各ステップの動作を図 7.1 に示す。

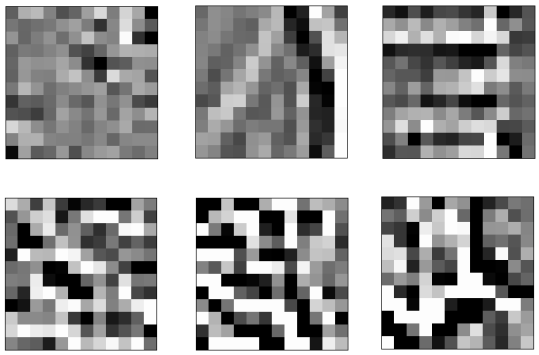

図 7.2: 入力する画像の例。

7.3　実験：自然画像の学習

7.3.1　実験条件

自然画像に対し、下記の 3x3 のラプラシアンフィルタでエッジを強調した画像を入力として用いる。

$$L = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix} \tag{7.3}$$

前処理の手順をより詳しく述べる。[0,255] の範囲の整数値を持つ輝度情報を [-128,127] にずらしたあと、上記のラプラシアンフィルタを適用、その後、再び [0,255] の範囲の値に戻す。(範囲をはみ出た値は最小値、最大値に置きかえられる。) さらに [0,255] の値を [0,1) の浮動小数点の値にスケール変換して、6 章で述べた方法で、12x12=144 個の２値の入力ノードに入力する。

入力する 12x12 ピクセルのラプラシアンフィルタ適用後の画像のサンプルを図 7.2 に示す。

この章の実験における近傍学習では、ぼかし関数 b と 近傍関数 n は以下のものを使っている。(各記号の意味は 4 章参照。) 方位選択性がはっきりでるように入力をぼかし関数でぼかすのをやめるとともに、近傍半径の最低値も他の実験よりも小さめに設定した。

$$b^{Natural}(\alpha, d_x, d_y) = 1 \tag{7.4}$$

$$n^{Natural}(\alpha, d_x) = smoothStep(d_x, (s{+}1)\alpha{+}1) \tag{7.5}$$

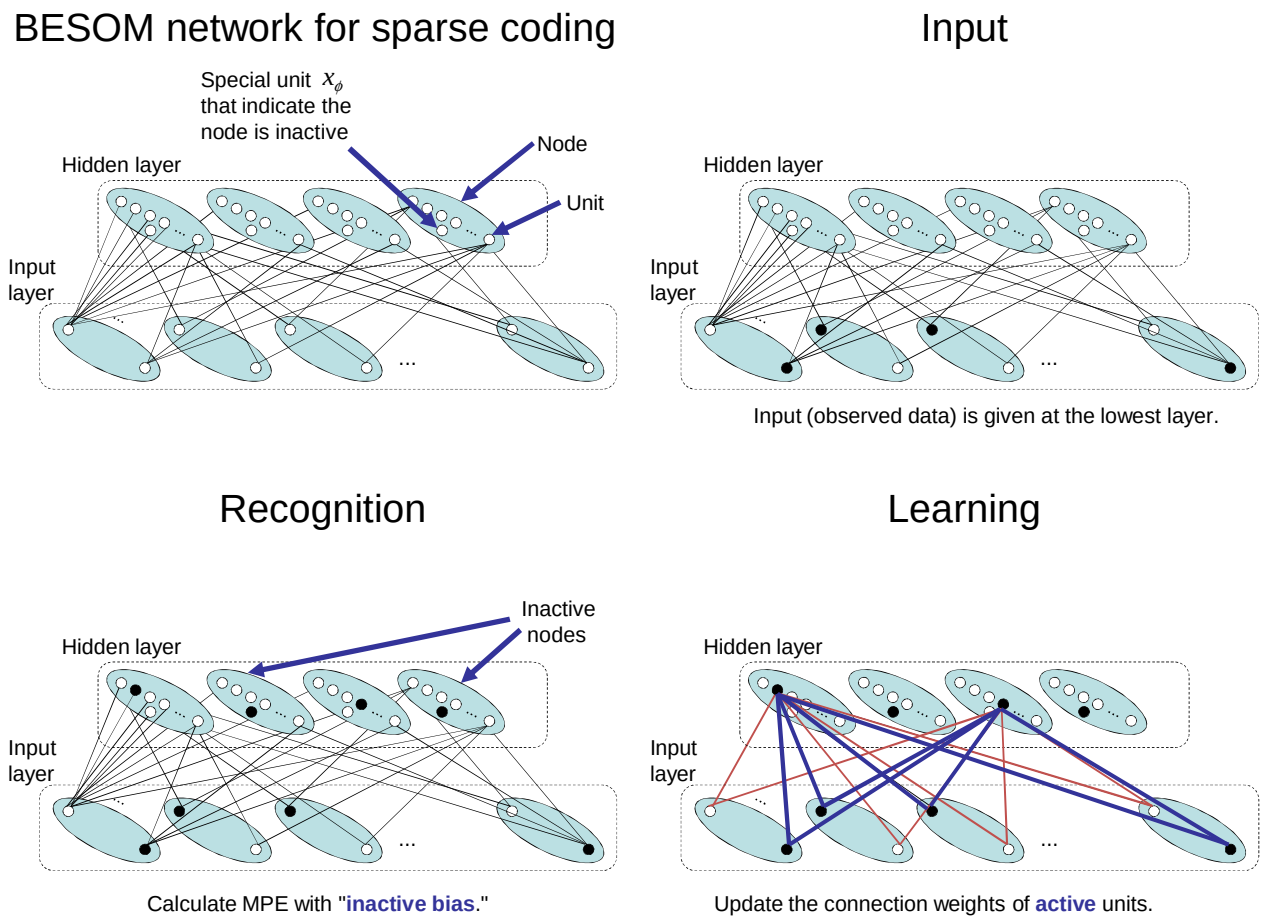

図 7.1: ＢＥＳＯＭによるスパース符号化の各ステップの動作。
（左上）ネットワークの構造。ここでは隠れ層と入力層の２層からなるＢＥＳＯＭネットを用いる。楕円はノード、小さな白い丸はユニットを表す。同一層内のノード間にはエッジはなく、異なる層に属するノード間には、エッジがある。隠れノードはそれぞれ１つだけ特別なユニット（φ値ユニット）を持つ。
（右上）入力。観測データは入力層のノードの値として与えられる。
（左下）認識ステップ。ネットワーク全体がベイジアンネットとして動作し、ＭＰＥが計算される。ただしこの時、できるだけ多くの隠れノードがφ値をとるようにバイアスがかけられる。
（右下）学習ステップ。すべての隠れノードが１次元ＳＯＭとして動作する。勝者ユニット（黒丸）と子ノードのユニットとの間の結合の重みが更新される。ただし、φ値ユニットが勝者になったノードは重みを更新しない。

同時に、近傍学習のない学習則（ベクトル量子化）を用いる実験も行った。この場合は、以下の関数を用いた。

$$b^{VQ}(\alpha, d_x, d_y) = \begin{cases} 1 & (d_y < 1) \\ 0 & (d_y \geq 1) \end{cases} \tag{7.6}$$

$$n^{VQ}(\alpha, d_x) = \begin{cases} 1 & (d_x < 1) \\ 0 & (d_x \geq 1) \end{cases} \tag{7.7}$$

ノード数は４、ユニット数はφ値ユニットを含めて３０、スパース性のパラメタは $\beta = 80$ で自然画像の学習を行った。8 章で述べる側抑制ＩＣＡの機構はここでは使っていない。

7.3.2 学習結果

図 7.3 は、学習結果である。左側は近傍学習を行った場合、右側は近傍学習を行わない場合である。いずれも、方位選択性が獲得されている。

筆者がいろいろな入力画像と認識結果の基底画像の組を見比べてみたところでは、「勝者ユニットの基底画像の平均によって入力画像を近似する」という、意図した学習の目的は、達成されているようである。

しかしながら、Olshausen らのスパース符号化の結果 [11] と比べるといくつか違いがある。

まず、大きな違いとして、近傍学習がある場合もない場合も、２つの方位選択性が重なって「まだら模様」のようになった基底画像が見られる点である[1]。

もう１つの違いは、ガボールフィルタのような空間的局所性が、少しはあるものの、はっきりとは見られない点である。

これらの違いは、ノード数やユニット数を変えても解消できなかった。しかし、入力画像の前処理の仕方を調整したり、9 章で述べる特徴数を制限することで、解消できる可能性がある[2]。

したがってこれらの違いが、 BESOM アルゴリズムと一次視覚野の学習アルゴリズムとの本質的違いを表しているとまでは、現時点では言えないだろう。

[1]同じように２つの方位選択性が重なった基底画像は、下記の本に出ているトポグラフィを入れた自然画像の学習結果でも必ず見られるようである。
「Natural Image Statistics」
`http://www.naturalimagestatistics.net/`

[2]予備実験により、特徴数を制限すると、基底画像の空間的局所性が強くなることは確認済みである。ただし、おそらく前処理が未調整なせいか、まだきれいな基底画像にはなっていない。

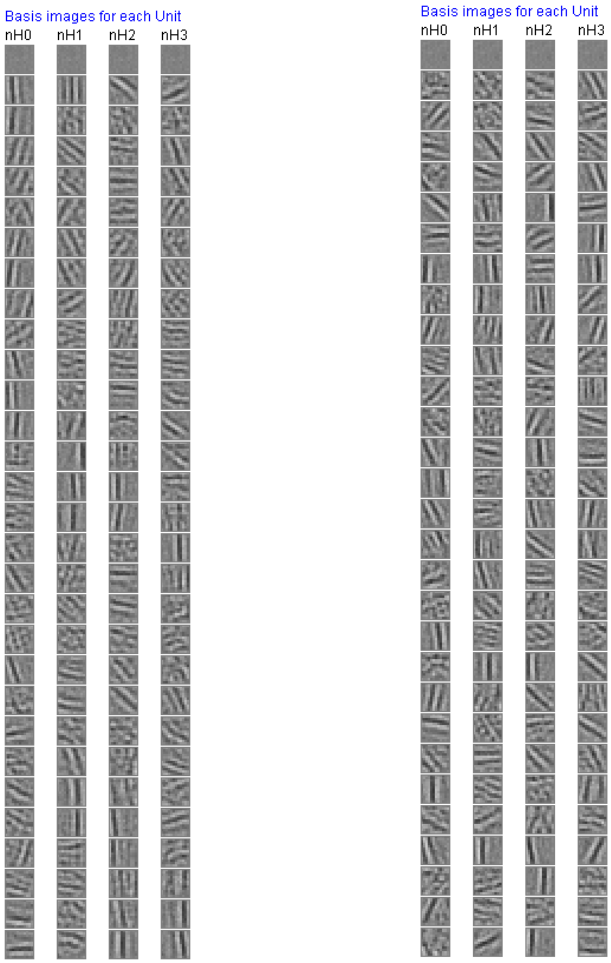

図 7.3: 自然画像の学習の結果得られた基底画像。左は近傍学習あり、右は近傍学習なし。ユニット数が多めなのは近傍学習の効果を見えやすくするため。

第8章　側抑制を用いた非線形ＩＣＡ

本章では、アンチヘブ則で学習する側抑制結合を用いて非線形ＩＣＡを行うアルゴリズムについて述べる。基本的なアイデアは TR2009[3] に書いたものと同じだが、ベイジアンネットの学習則との統合や混合分布への対応などの改良の結果、安定的に動作するようになっている。

8.1　背景

TR2009[3] でも述べたように、BESOM は、感覚器からの入力を非線形**ＩＣＡ（独立成分分析）**[9] することで、外界のモデルを獲得する。

スパース符号化は信号源がすべて優ガウスならば一種のＩＣＡアルゴリズムとして動作するが、生物の外界は必ずしも優ガウスな信号源ばかりとは限らない。サルの側頭葉で顔の向きに応じて応答する場所が変化するコラムが見つかっているが、顔の向きは劣ガウスな分布に従う値の例である。

信号源が劣ガウスの場合のＩＣＡアルゴリズムとして、TR2009[3] では、田尻らのアルゴリズム [13] を BESOM に適用したものを提案した。しかし、TR2009[3] の段階では、スパース符号化と同時に動かすと、学習が振動するなど動作が不安定であった。その理由は、２つのノードが独立になってもペナルティが０になっていなかったせいだと思われる。

本章では、その問題を改良したアルゴリズムとその実験結果について述べる。

基本的な動作は TR2009[3] で述べたものと同じで、隠れ層のノードすべてのユニット間に抑制性結合を持たせ、同時に活性化しやすいユニットには認識ステップでペナルティを与えることで、ユニット間の活性の相関を減らす、というものである。

このアルゴリズムの各ステップの動作を図 8.1 に示す。

8.2　アイデア

今、同一の層内にある２つの兄弟ノードを U, V とする。また、以下、 $i, j \neq \phi$ とする。

今回の学習則では、 U, V どうしを独立にするのではなく U, V がともに活性ノードであるという条件付のもとで、２つの値を独立にすることを目標にする。

すなわち、

$$\begin{aligned} &P(V = v_j | U = u_i, U \neq u_\phi, V \neq v_\phi) \\ &\quad = P(V = v_j | U \neq u_\phi, V \neq v_\phi) \end{aligned} \tag{8.1}$$

という制約が成り立つベイジアンネットの学習を目的とする。

この目的を達成するには、左辺の推定値

$$S_{ij}^{UV} = \hat{P}(V = v_j | U = u_i, U \neq u_\phi, V \neq v_\phi) \tag{8.2}$$

と右辺の推定値

$$T_j^{UV} = \hat{P}(V = v_j | U \neq u_\phi, V \neq v_\phi) \tag{8.3}$$

をカウンティングによりオンライン学習し、ＭＰＥ計算において $S_{ij}^{UV} - T_j^{UV}$ に比例する値をペナルティとして与えればよい。

なお、ユニット活性度一律化（4.2 節）を仮定すれば、T_j^{UV} の値は次の値で代用できる。

$$T_j^{UV} = \hat{P}(V \neq v_\phi | U \neq u_\phi, V \neq v_\phi)/s \tag{8.4}$$

ただし s はノード内の非φ値ユニットの数である。こちらの方が学習すべきパラメタが少し減るので過適合の可能性が減ると期待できる。

ペナルティをベイジアンネットの枠組みの中で実現するためには、スパース符号化の時と同様に、兄弟ノードの値を制約する共通の子ノードが存在すると考えればよい。このように解釈することにより、ＩＣＡの問題が 2 章の定式化の枠内で表現できて見通しがよくなる。例えば厳密解の一意性は自明の性質となる。（もちろん、実際にはオンライン学習による近似解法を用いるので、局所解に陥る可能性はある。）また、 TR2009[3] で問題となった、部分アルゴリズムどうしの干渉が起きにくいことが期待できる。

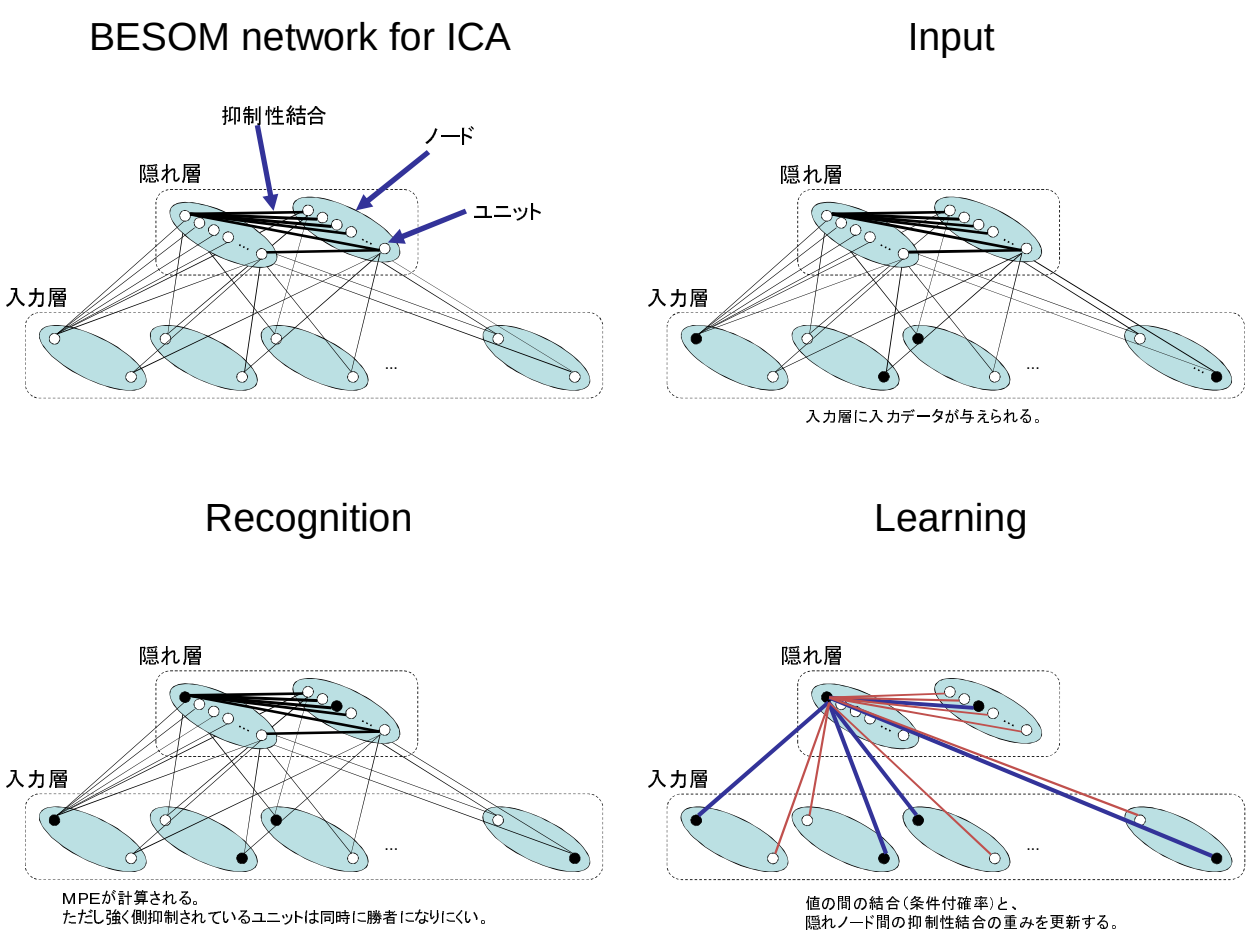

図 8.1: ＢＥＳＯＭによるＩＣＡの各ステップの動作。
（左上）ネットワークの構造。ここでも隠れ層と入力層の２層からなるＢＥＳＯＭネットを用いる。同じ層にある隠れノードのすべてユニット間には側抑制のための結合がある。
（右上）入力。観測データは入力層のノードの値として与えられる。
（左下）認識ステップ。ＭＰＥが計算される。ただし側抑制の結合の重みが大きいユニットの組は、同時に選ばれないように抑制される。
（右下）学習ステップ。通常の結合の更新に加え、側抑制の結合の重みも更新される。

8.3 アルゴリズム

同時確率の計算式に対して、以下のペナルティが与えられる。

$$R(x, \mathbf{x}) \propto e^{-\lambda S(x,\mathbf{x})} \tag{8.5}$$

λ は側抑制の強さを表す定数であり、$S(x, \mathbf{x})$ はユニット x が他のノードのユニットから側抑制を受ける強さである。２層ＢＥＳＯＭの場合は、隠れ層の中の全てのノード間に側抑制が働き、$S(x, \mathbf{x})$ は以下のように定義される。

$$S(u_i, \mathbf{x}) = \begin{cases} 0 & (i = \phi) \\ \sum_{v_j \in \mathbf{h}, v_j \neq u_i, j \neq \phi} (S_{ij}^{UV} - T_j^{UV}) & (i \neq \phi) \end{cases} \tag{8.6}$$

現在の実装では、S_{ij}^{UV} と T_j^{UV} の学習において、条件付確率表の学習と共用のグローバルな学習率を用いている。また、T_j^{UV} は代用版（式 (8.4)）を用いた。

8.4 実験

8.4.1 信号源の推定に失敗する例

図 8.2 は、２次元平面上の扇形の分布から生成される１つの点からなる画像をぼかした画像を入力し、２つの隠れノードで学習させた結果である。（グリッド状の座標の可視化の方法については TR2009[3] の 11.3 節参照。）

この例では２つのノードは独立にはなっているものの、意味のある信号源は獲得していない。この例が示すように、一般に、単に入力をＩＣＡをしただけでは、意図した信号源が獲得されるとは限らない。ＢＥＳＯＭによるＩＣＡは非線形ＩＣＡだが、一般に非線形ＩＣＡは解に一意性がないので、真の信号源を獲得させるには、信号源どうしの独立性に加えて、別の制約条件もしくは信号源に関する何らかの情報が必要となるのである。

8.4.2 ヒントとなる情報を使った信号源の推定

「真の信号源に関する何らかの情報」を入力に与える例の１つとして、信号源の１つにヒントを追加する例を以下に示す。

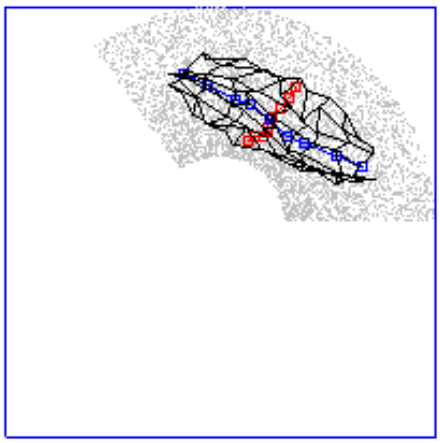

図 8.2: 扇型の分布から生成される１点からなる画像の学習結果。この例では信号源に関する何のヒントも与えられておらず、その結果、意図した信号源が正しく学習されない。

図 8.2 の扇形の分布内の点は、極座標における角度 θ と距離 r という２つの独立な信号源から決まる。このうち、中心からの距離 r と同じ値を y 座標に持つ、もう１つの点を、入力画像に加える。ただし、このヒントとなる点は、入力画像に 1/2 の確率でしか与えられないものとする。このような入力画像のサンプルを図 8.3 に示す。

これを学習した結果の２つのノードの各ユニットの受容野を可視化したものが図 8.4 である。２つのノードが角度 θ と距離 r という２つの独立な信号源を正しく学習している。図 8.5 に、学習結果における各ユニットの基底画像を示す。

なお、学習が収束したあとは、ヒントとなる点が与えられていなくても、扇形上の１点が与えられるだけで、角度 θ と距離 r の値が２つのノードに正しく表現される。つまり、学習が収束し終了した後は、信号源を推定するためにもはやヒントは必要ないのである。

実際の大脳皮質の学習においても、学習時にはヒントを使うが、認識時にはヒントなしで信号源を推定する、ということが、おそらくよく行われているのだろうと筆者は考えている。

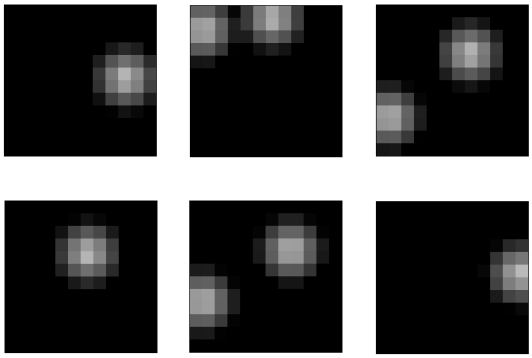

図 8.3: 信号源の１つに関するヒントが与えられる入力画像の例。扇形の中心からの距離 r と同じ値を y 座標に持つもう１つの点が、左端に 1/2 の確率で与えられる。

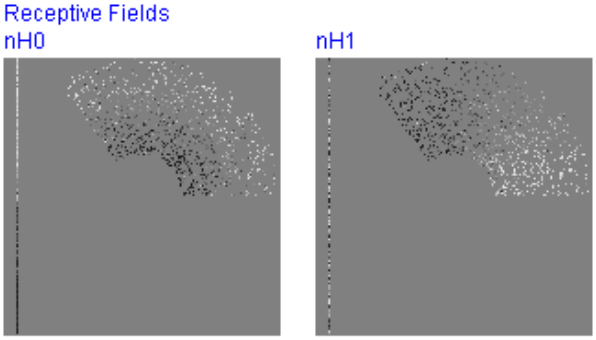

図 8.4: 信号源の１つに関するヒントが与えれた場合の学習結果を可視化したもの。２つのノードの各ユニットの受容野を、ユニットごとに色を変えてプロットした。２つのノードが距離 r と 角度 θ という２つの独立な信号源を正しく学習している。

図 8.5: 学習結果における各ユニットの基底画像。

8.4.3 混合分布の要素のＩＣＡ

この節では、ＩＣＡの機構と混合分布の学習の機構が両立する例を２つ示す。これらの例は、 TR2009[3] で述べたアルゴリズムでは、振動して動作しなかった。

図 8.6 は、２次元平面上の２つの長方形からなる分布から生成される１点を持つ画像を、４つのノードで学習させた例である。常に２つのノードが活性化するようにスパース性を調整した（$\beta = 20$）。２つのノードが表す軸はゆがんではいるが、分離した２つの分布がそれぞれ２つのノードで学習されている。

次の例は、４つの点からなる２種類の動物の顔のような画像の学習例である。それぞれの動物の目の位置と鼻の位置はそれぞれ異なる分布から独立に生成される。図 8.7 は入力画像のサンプルである。図 8.8 は４つの隠れノードでの学習結果で、やはり常に２つのノードが活性ノードになるようにスパース性を調整した（$\beta = 60$）。学習結果では、４つのノードがそれぞれ、１種類の動物の顔の目の位置と顔の位置、という信号源の学習に成功している。図 8.9 に、学習結果の各ユニットの基底画像を示す。

8.5 生物学的妥当性

T_j^{UV} の値は、大規模なＢＥＳＯＭネットワークでは小さな値になり、高い精度での学習が難しくなるかもしれない。しかし、その場合は逆に、学習せずに $T_j^{UV} = 0$

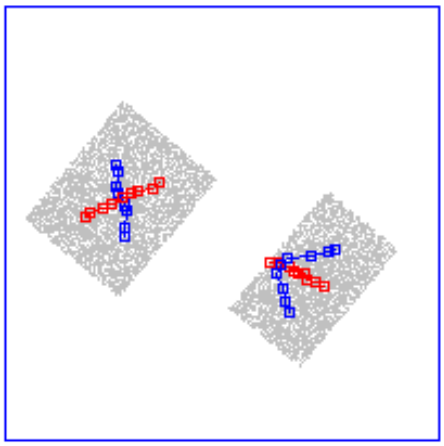

図 8.6: ２次元平面上の２つの長方形からなる分布から生成される１点を持つ画像を、４つのノードで学習させた例。

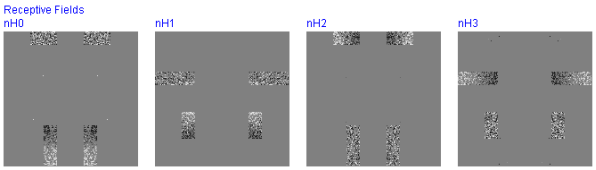

図 8.8: ２種類の動物の顔の学習結果における各ユニットの受容野を可視化したもの。それぞれの動物の目の位置と鼻の位置という信号源が期待通りに獲得されている。

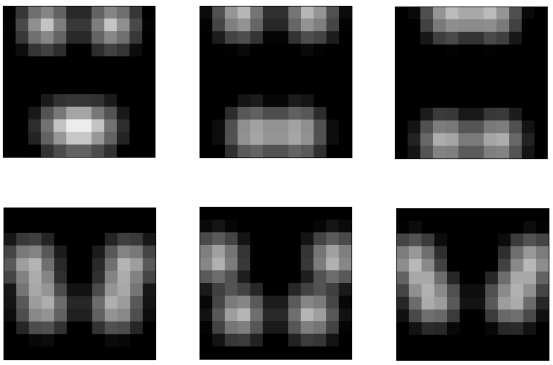

図 8.7: ４つの点からなる２種類の動物の顔のような入力画像の例。

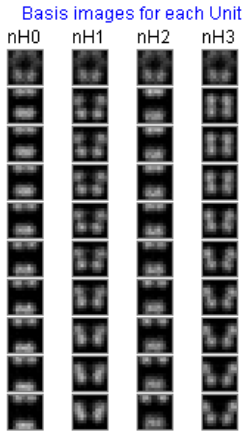

図 8.9: ２種類の動物の顔の学習結果における各ユニットの基底画像。

としてしまってもよいかもしれない。その方が、神経回路での実現も容易である。

$T_j^{UV} = 0$ としても問題なく動作するかどうかについては、将来、大規模ＢＥＳＯＭネットワークで検証したい。

8.6　計算量について

側抑制ＩＣＡは全てノード間の全てのユニット間に結合を持たせる必要があるため、ノード数 n 、ノードあたりの非φ値ユニット数 s とすると、$O(s^2n^2)$ の側抑制シナプスが必要となる。s は定数だが n はヒトで１万～１０万はあると思われるので、このままでは計算量的に問題になり得る。

しかし、おそらく、9 章で述べる「インクリメンタルソート」を用いる方法で計算量を $O(s^2n)$ に減らせるのではないかと考えている。つまり、重みの最も大きい定数個の側抑制のみ計算し、残りは無視することにすればよい。しかし、定数個の側抑制のみで本当にＩＣＡが実行できるかどうかは、実データを使った実験で今後検証していく必要がある。

第9章　特徴選択に基づく構造学習

本章では、特徴選択によってノード間のエッジの数を大幅に少なくする機構について述べる。この機構は、ネットワークを大規模化するために必要となる。

9.1　背景

現在の BESOM は隠れノードの数が数個程度でしか動作せず、顔認識などの複雑なタスクに用いることはできない。大規模化ができない理由は、計算量のオーダーである。そこで現在、1回の入力をノード数 n に対して O(n) 程度の計算量で処理できるスケーラブルな認識・学習アルゴリズムの実現を目指している。

スケーラブルなアルゴリズムの実現の大きな障害になるのが、層間のエッジの数である。1つの層の中のノード数を n とすると、2つの層の間がフルに結合されている場合、結合の本数が $O(n^2)$ なので、そのすべてを利用する認識アルゴリズムの計算量は $O(n^2)$ 以上にならざるを得ない。逆に言えば、スケーラブルなアルゴリズムのためには、1つのノードから見た親・子ノードの数を O(1) 程度に抑える必要がある。

実際の大脳皮質でも、種によってニューロン数は大きく異なるにもかかわらず、ニューロン1個あたりのシナプス数は、およそ1万個であり、種によって変わるという話は聞かない。このことから、脳もまた、ニューロン間の接続数を定数に保つことで、シナプス数の爆発を防ぎ、「脳のスケーラビリティ」を実現していると想像できる。

BESOM モデルのようなベイジアンネットにもとづく大脳皮質モデルにおいては、エッジの数を減らすべきもう1つの大きな理由として、オーバーフロー・アンダーフローの問題がある。確率伝播アルゴリズムや belief revision アルゴリズムでは、子ノードの数だけ λ_{Y_i} メッセージの掛け算を実行する必要がある。子ノードの数が多いと、掛け算すべき数が増え、オーバーフロー・アンダーフローが起きやすくなる。対数を使えばオーバーフローに関しては起きにくくなるが、その場合でも掛け算の数が多いと計算精度で問題が起きるかもしれない。これらは浮動小数点を使う計算機でも問題だが、ダイナミックレンジの小さいニューロンによる演算ではより一層問題になると思われる。

多すぎるエッジは、汎化能力の面でも問題がある。各ノードをパターン認識装置と見なすと、子ノードは認識対象が持つ特徴である。認識装置から見ると、自分の下の層にあるノードが表現するほとんどの特徴は、認識には不要な特徴であろう。不要な特徴があると、過適合の要因になるだけで汎化能力はかえって落ちてしまう。汎化能力を上げるためには、**特徴選択**の技法を使って、不要な特徴を捨て、有用な特徴のみを使うようにする必要がある。なお、脳が特徴選択を行っているとしたら、そのアルゴリズムはオンラインで動作するものでなければならない。

エッジの数の制限は、部品別学習 [12] の効果をもたらし、画像認識の汎化性能を上げることも期待できる。

9.2　アイデア

工学でよく使われている特徴選択の技法を参考に、すべてのノードから見て子ノードの数を定数 E 個に制限する機構を考える。

特徴選択の方法にはいろいろあるが[1]、ここでは、相互情報量を特徴の有用性を測る基準（以下、**スコア**と呼ぶ）として、特徴選択を行う方法を提案する。相互情報量は、ベイジアンネットの構造学習でも、しばしばエッジ選択の基準として用いられる。

エッジの数を制限するためには、特徴選択は当然ノード単位で行われるべきだが、一次視覚野への入力に関する解剖学的知見は、ユニット単位の特徴選択の機構の存在を思わせるため、こちらの検討も必要なように筆者には思われた。そこで次の節では、ノード単位とユニット単位の両方のアルゴリズムについて述べる。

[1]参考：
「特徴選択 - 機械学習の「朱鷺の杜 Wiki」」
http://ibisforest.org/
「Introduction to Information Retrieval」 13.5 章
http://nlp.stanford.edu/IR-book/information-retrieval-book.html

9.3 アルゴリズム

9.3.1 ノード単位の特徴選択

ノード X, Y の間の相互情報量は下記のように定義される。

$$
\begin{aligned}
I(Y,X) &= \sum_i \sum_j p(y_j, x_i) \log \frac{p(y_j, x_i)}{p(x_i)p(y_j)} \\
&= \sum_i \sum_j p(y_j|x_i)p(x_i) \log \frac{p(y_j|x_i)p(x_i)}{p(x_i)p(y_j)} \\
&= \sum_i \sum_j p(y_j|x_i)p(x_i) \log \frac{p(y_j|x_i)}{p(y_j)} \qquad (9.1)
\end{aligned}
$$

これを、ノード X から見た特徴（子ノード） Y のスコアを見なし、スコアの高い特徴を E 個選択すればよい。

より正確には次のようになる。ＢＥＳOMの学習ステップにおいては、直前の認識ステップの結果を用いて、あたかも層の間は完全に結合されているかのように、すべての条件付確率表の値を更新する。認識ステップにおいては、まず現在の条件付確率表および各ノードの $p(y_j), p(x_i)$ の値をもとにスコアを計算し、 X から見て Y のスコアの順位が E より低ければ、X から Y へのエッジが存在しないものと見なした上で、ＭＰＥ計算を行う。

なお、この方法では、認識時に実質的にエッジが減ることでＭＰＥの計算量は減らせるが、条件付確率表の記憶域の量は $O(n^2)$ と変わらない。この問題については 9.7 節で議論する。

特徴選択の機構自身に必要な計算量については 9.5 節で考察する。

9.3.2 ユニット単位の特徴選択

ノード X のユニット x_i が勝者になるかならないかを表す確率変数を考えると、その確率変数と子ノードとの相互情報量は下記の式のようになる。

$$
\begin{aligned}
&\sum_j p(y_j|X = x_i)p(X = x_i) \log \frac{p(y_j|X = x_i)}{p(y_j)} \\
&+ \sum_j p(y_j|X \neq x_i)p(X \neq x_i) \log \frac{p(y_j|X \neq x_i)}{p(y_j)} \qquad (9.2)
\end{aligned}
$$

BESOM では $i \neq \phi$ のとき

$$
p(y_j|X \neq x_i) \approx p(y_j|x_\phi) \approx p(y_j) \qquad (9.3)
$$

が成り立つとすれば（要検証）、２項目は０になる。（なお、 $i = \phi$ の場合は meanOR モデルでは $p(y_j|x_\phi)$ の値を参照しないので、特徴選択をする必要がない。）また、 $p(X = x_i)$ はユニット x_i におけるスコア比較において定数なので不要である。したがって、

$$
\sum_j p(y_j|X = x_i) \log \frac{p(y_j|X = x_i)}{p(y_j)} \qquad (9.4)
$$

をスコアにすればよいことになる。（y_ϕ については特別扱いしない点に注意。）

なお、ユニット単位で特徴選択をする場合、ノードの間のエッジは切れない。現在の実装では、あるユニット x_i から見た特徴 y_j が無効の場合、条件付確率の値 $P(y_j|x_i)$ の代わりに $P(y_j)$ を使うことで、特徴選択を実現している。

9.4 実験

9.4.1 違う場所に独立に発生する３つの点の学習

現在動いている小規模２層ネットワークでは、ノード単位の特徴選択は効果が見えにくいので、今回はユニット単位の特徴選択を実装して、簡単な動作確認を行った。いずれの実験も、側抑制ＩＣＡの機構を働かせている。

最初の実験は、特徴選択の基本動作が動いていることを確認する簡単な例である。違う場所に独立に発生する最大３つの点からなる画像の信号源を学習するのが目的である。３つの点はそれぞれ 0.1 の確率で、互いに重ならない長方形の領域の内部にランダムに発生する。図 9.1 は、入力画像のサンプルである。

スパース性パラメタ $\beta = 150$ 、側抑制の強さのパラメタ $\lambda = 100$ 、隠れ層のノードの数は４で学習を行った。

図 9.2 は、特徴選択をしない場合の４つのノードの受容野を可視化したもの、図 9.3 は、特徴選択をしない場合の基底画像である。ＩＣＡとスパース符号化の機構により、３つの独立な信号源が正しく獲得されていることが分かる。

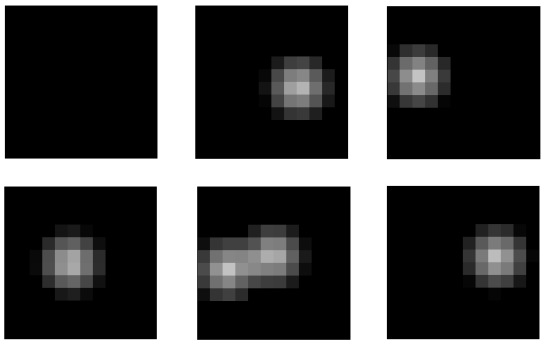

図 9.1: 互いに重ならない違う場所に独立に発生する最大３つの点からなる画像のサンプル。

図 9.4 は、特徴選択の機構により、ユニットごとに２０ピクセルの特徴を選択した場合の学習結果である。基底画像の中の青いピクセルは、その特徴が選択されていないことを意味する。このタスクの場合は、特徴選択があってもなくても獲得される信号源はほぼ同じである。各ユニットの認識に無関係なピクセルが無効になっていることが分かる。

9.4.2 独立でない２つの部品の学習

次の実験は、特徴選択によって部品別学習 [12] の効果が出ることを確かめる簡単な例である。

入力画像は、２つ点のからなる。２つの点はそれぞれ重ならない長方形の領域内でランダムに発生する。ただし、 1/2 の確率で、それぞれの長方形の領域内の同じ相対位置に発生し、残りの 1/2 では、２つの点は、独立にそれぞれの長方形の中のランダムな位置に発生する。図 9.5 は、このタスクの入力画像のサンプルである。上の３つは同じ相対位置に発生する例、下の３つは独立な位置に発生する例である。

スパース性パラメタ $\beta = 0$ 、側抑制の強さのパラメタ $\lambda = 100$ 、隠れ層のノードの数は２で学習を行った。

図 9.6 は、特徴選択を行わない場合の各ノードの受容野を可視化したもの、図 9.7 は、その時の各ユニットの基底画像である。ここでは２つのノードは長方形の x 軸と y 軸の情報を表現している。

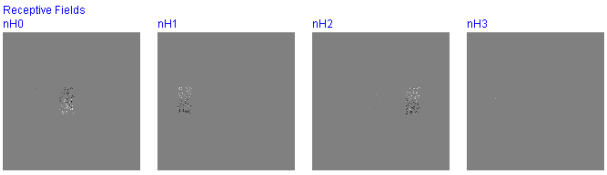

図 9.2: 特徴選択をしない場合の４つのノードの受容野を可視化したもの。

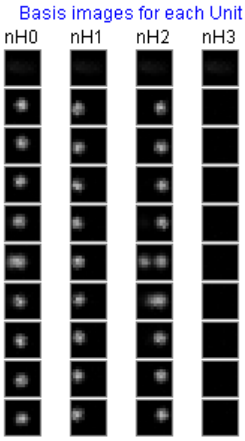

図 9.3: 特徴選択をしない場合の４つのノードの基底画像。

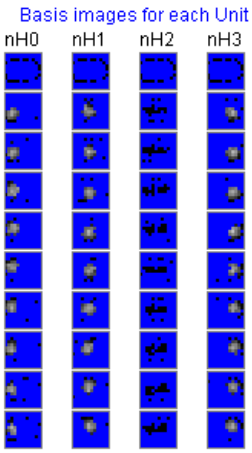

図 9.4: 特徴数を２０ピクセルに制限した場合の４つのノードの基底画像。青いピクセルはその特徴が選択されていないことを意味する。

一方、図 9.8 は、特徴数を２０ピクセルに制限した場合の各ノードの受容野、図 9.9 は、その時の基底画像である。この場合は、２つのノードは２つの点それぞれの x 座標の情報を表現している。これは特徴選択の機構により、「少ないピクセルで表現できる信号源」が獲得されるよう、学習にバイアスがかかった結果と見ることができる。

このバイアスは、実世界のように、「少ないピクセルで表現できる信号源」、すなわち「部品」の組み合わせで入力画像が構成されているときには、真の信号源を獲得するためのヒントとして働き、画像認識の汎化能力を上げる効果があると期待できる。

9.5 １ステップあたり計算量 O(n) のオンライン特徴選択アルゴリズムの実現に向けて

未実装ではあるが、本章で述べたオンライン特徴選択アルゴリズムは各学習ステップにおいてノード数 n に対し $O(n)$ で実行できそうである。具体的方法を以下に示す。

前提として、ＭＰＥで活性ノードとなる隠れノードの数は高々定数個であるとする。（ノード活性のスパース性の仮定。）

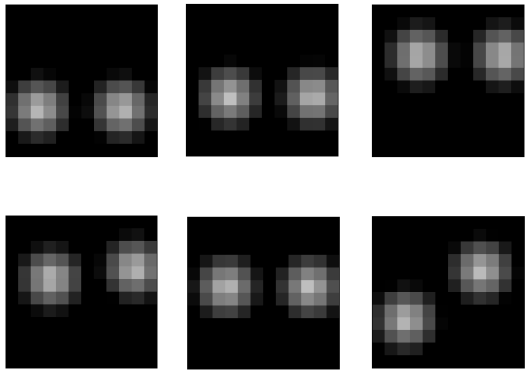

図 9.5: 完全には独立でない２つ点からなる入力画像のサンプル。上の３つは２つの点の相対位置が同じ、下の３つは２つの点の位置は独立な場合の例。

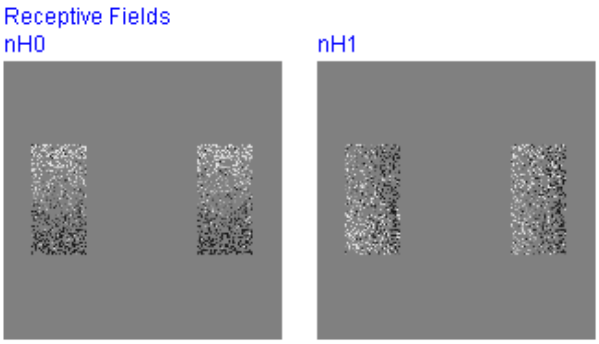

図 9.6: 特徴選択を行わない場合の各ノードの受容野を可視化したもの。２つの点の x 軸と y 軸の情報が表現されている。

図 9.7: 特徴選択を行わない場合の各ノードの基底画像。

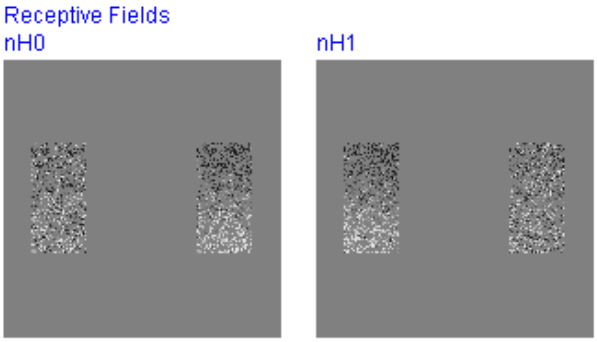

図 9.8: 特徴数を２０ピクセルに制限した場合の各ノードの受容野。それぞれ別の点の x 軸の情報が表現されている。

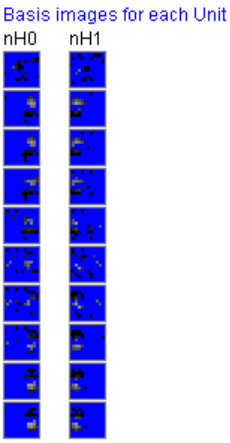

図 9.9: 特徴数を２０ピクセルに制限した場合の各ノードの基底画像。青いピクセルはその特徴が選択されていないことを意味する。

すると、スコア計算で必要となる値 $p(y_j|x_i)$、$p(y_j)$、$p(x_i)$ のうち更新すべきものの数もまた定数個で済むことになる。

次に特徴選択だが、不活性ノードについては、条件付確率表の更新がなく、子ノードのスコアがほぼ変化しないので特徴選択の計算をしなおす必要はない。（厳密には子ノードの $p(y_j)$ が少し変化するが、おそらく無視できる。）一方、特徴選択を計算し直すべき活性ノードの数は仮定により定数個である。

ある活性ノードに注目すると、子ノードの数は $O(n)$ なので、すべてのスコアの再計算に必要な計算量は $O(n)$ である[2]。

最後に $O(n)$ 個の子ノードの中から、スコアの高い E 個を選択する方法だが、挿入ソートのような、インクリメンタルな実行が可能なソートアルゴリズムを使えばよい。挿入ソートでは、前回の特徴選択時におけるソート結果を保持しておけば、そのうちの１個の要素が十分に小さな値だけ変化した場合、ソート結果の更新に必要な計算量は $O(1)$ である。n 個の要素が変化する場合は $O(n)$ である。

以上により、ノード活性のスパース性を仮定すれば、オンライン特徴選択が１ステップあたり $O(n)$ で済む

[2]もし $p(y_\phi|x) = 0$ という条件が成り立つなら、活性ノードのスパース性の仮定からスコア再計算すべき子ノードの数は定数ですむ。すると、特徴選択の計算量も $O(1)$ ですむはずである。ただし、$p(y_\phi|x) = 0$ という条件が成り立つ（あるいはモデルにそのような制約を入れる）ことが妥当かどうかは現時点では分からない。

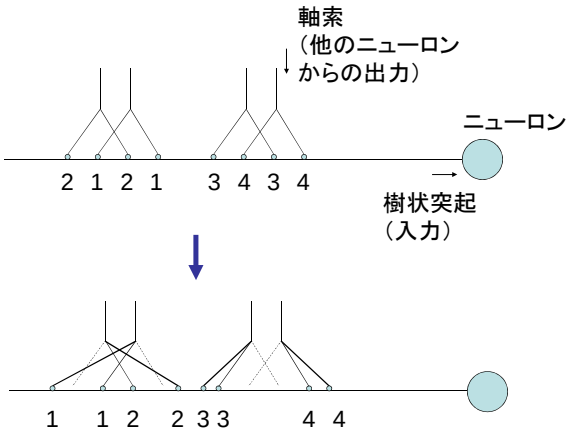

図 9.10: ニューロンによる挿入ソートの実現の１つの可能性。樹状突起上において、周辺のシナプスよりも小さい値のシナプスを持つ軸索はより受け手のニューロンに遠い側に、大きい値のシナプスを持つ軸索は近い側に側枝を伸ばし（図下、太線）、新たなシナプスを作る。古い接続は切る（図下、破線）。

ことが示せた。

ところで、挿入ソートのような比較的複雑な処理が、神経回路で実現可能だろうか。ここで、挿入ソートのニューロンによる実現の１つの可能性について述べよう。図 9.10 のように、樹状突起（他のニューロンからの入力を受け取る突起）上で、シナプスが近傍のシナプスとの重みに基づいて競合すると仮定する。最初、出力側のニューロンは出力先ニューロンの樹状突起上のランダムな位置にシナプスを作る。次に、樹状突起上において、周辺のシナプスよりも小さい値のシナプスを持つ軸索はより受け手のニューロンに遠い側に、大きい値のシナプスを持つ軸索は近い側に側枝を伸ばし新たなシナプスを作る。この動作を繰り返せば、樹状突起上には重みの順にシナプスが並ぶことになる。この動作は局所的な情報だけを用いて実行可能であり、生物学的に無理のない機構であると思われる。

もちろん、仮にニューロンに挿入ソートができるとしても、神経回路で挿入ソートに基づいた特徴選択ができるという主張を正当化するには、まだ大きなギャップがある。しかし、少なくともそれは、決してあり得ないことではない。

9.6　スコアの生物学的妥当性

下記の相互情報量に基づくスコアが、神経回路で簡単に計算可能かどうかを考察する。

$$I(Y,X) = \sum_i \sum_j p(y_j|x_i)p(x_i) \log \frac{p(y_j|x_i)}{p(y_j)} \quad (9.5)$$

まず $i = \phi$ の項について考える。BESOM においてφ値ユニットは子ノードの値とほぼ独立という性質から、$p(y_j|x_\phi) \approx p(y_j)$ である。すると、

$$\begin{aligned} \log \frac{p(y_j|x_\phi)}{p(y_j)} &\approx \log \frac{p(y_j)}{p(y_j)} \\ &= \log 1 \\ &= 0 \end{aligned} \quad (9.6)$$

なので、 $i = \phi$ の項は無視できる。$j = \phi$ の項は一般に無視できない。しかし、近傍学習の効果などによってもし $p(y_\phi|x_i) \approx 0$ が成り立つのなら、$p(y_\phi|x_i)p(x_i) \log p(y_\phi|x_i)/p(y_\phi) \approx 0$ となり、$j = \phi$ の項もやはり無視できるかもしれない。これについては将来、実験により検証する必要がある。

仮に、$i = \phi$、$j = \phi$ の項が無視できるとする。するとユニット活性度一律化（4.2 節）により、$p(x_i)$ は $i \neq \phi$ において同じ値なので、スコア比較において無視できるようになる。したがって、次の値をスコアとすればよい。

$$\sum_i \sum_j p(y_j|x_i) \log p(y_j|x_i)/p(y_j) \quad (9.7)$$

各項 $p(y_j|x_i) \log p(y_j|x_i)/p(y_j)$ は、 $p(y_j)$ が小さいときには下記のように近似できる。

$$\begin{aligned} & p(y_j|x_i) \log p(y_j|x_i)/p(y_j) \\ \approx\ & (-\log p(y_j))p(y_j|x_i) \end{aligned} \quad (9.8)$$

例として、 $y = x \log(x/0.1)$ および $y = x \log(x/0.001)$ のグラフを図 9.11 に示す。なお、この２つのグラフを近似する直線の傾きは $-\log 0.1 = 2.3$、$-\log 0.001 = 6.9$ である。

したがって、スコアは、 $p(y_j|x_i)$ の値を $-\log p(y_j)$ で重みづけした線形和で計算できることになる。この段階ですでに、スコア計算は生物に実現可能になりそうなかなり簡単な計算になった。

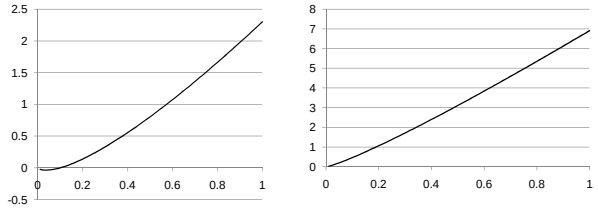

図 9.11: $y = x\log(x/a)$ のグラフ。 a が十分小さければグラフはほぼ直線 $y = -(\log a)x$ となる。

さらに、$p(y_j)$ が小さい値ならば、重み $-\log p(y_j)$ は、ほとんど定数と見なせる可能性がある。例えば $-\log 10^{-4} = 9.2, -\log 10^{-3} = 6.9$ なので、$p(y_j)$ の値の幅が 1 桁あったとしても重みはあまり変わらない。

もし重みが定数と見なせるならば、スコアは $p(y_j|x_i)$ の単なる総和で済むことになる。

$$\sum_i \sum_j p(y_j|x_i) \tag{9.9}$$

さて、ＢＥＳＯＭモデルによれば、$p(y_j|x_i)$ はノード X を表現するマクロコラム近傍に、シナプスの重みとして存在する値である。その総和を生物学的に計算する手段はいろいろ考えられるだろう。例えば、各シナプスが重み $p(y_j|x_i)$ に比例する量の何らかの化学物質を放出すれば、その近辺における化学物質の濃度がスコアを表すことになる。

以上の考察は、いくつか検証が必要な仮定を含んではいるものの、相互情報量に基づく特徴選択が、生物にとって決して実現不可能とは言えないことを示すには十分だろう。

9.7　メモリ量のオーダーの問題

ここで述べたオンライン特徴選択アルゴリズムでは、条件付確率表の記憶域の量は $O(n^2)$ と変わらない。全ての子ノードのスコアを計算してみないと、その中からスコアの高い特徴を選ぶことができるはずがないことを考えると、この記憶域の量は本質的に減らすことができないように筆者には思われる。

BESOM の応用の観点からは、当面は、メモリの量は、計算量ほどは深刻な問題ではない。

しかし、生物学的妥当性の観点からは、「種によらずノードあたりのシナプス数はほぼ一定」という性質と矛盾するので問題である。この問題に対する答えの可能性についていくつか挙げておく。

1. ヒトのシナプスの数は赤ん坊のころに最大になってその後減るらしいが、赤ん坊のころにだいたいの特徴選択を済ませ、その後は特徴選択の候補を制限する、という戦略を脳はとっているのかもしれない。その場合、シナプス数は減らせるが環境の変化への適応力が犠牲になる。これは「臨界期」の原因の１つかもしれない。

2. 何らかのヒューリスティックスを用いて、メモリの量をほぼ $O(n)$ に落とせる可能性はあるように思える。例えば、大多数のスコアの低い子ノードに対して、スコア再計算の精度や頻度を大幅に減らすという工夫が可能かもしれない。この工夫もまた、環境の変化への適応力を多少犠牲にするだろう。なお、学習すべき外界の性質に関する事前知識が、ヒューリスティックスの形で作り込まれるならば、メモリの削減と同時に汎化能力が向上する可能性もある。

9.8　今後の課題

現状の実装では、ユニット単位の特徴選択になっており、計算量の問題もオーバーフロー・アンダーフローの問題も、まだ残ったままである。ノード単位の特徴選択を採用し、実装を全面的に見直すことで、これらの問題が解決するだろう。（ただし、脳の一次視覚野に関しては、ユニット単位での特徴選択を採用している可能性はあると考えている。）

今回提案したオンライン特徴選択アルゴリズムは、局所解におちいる可能性がある。局所解から脱出するための何らかの機構が将来必要になるかもしれない。

特徴選択は汎化能力を向上させると期待しているが、その効果の確認も今後の課題である。

第10章　belief revision アルゴリズムを用いた認識ステップの実現に向けて

認識ステップにおいて、現在の山登り法に代わり belief revision アルゴリズムを採用するには、いくつか解決すべき問題があった。これまで分かっていた問題の多くがほぼ解決されたので、この章で説明する。

10.1　背景

筆者は、大脳皮質は belief revision アルゴリズムの一種を用いてＭＰＥを計算していると考えている。

筆者が導いた近似 belief revision アルゴリズム（以下、**近似ＢＲ**と呼ぶ）は、大脳皮質のコラム構造・６層構造という解剖学的特徴とよく一致するので、生物学的妥当性が高い (IJCNN2011[5])。

また、近似ＢＲを用いた認識アルゴリズムは、以下の理由で、応用上も優れた性質を持つを考えている。(ただし、下記の性質は IJCNN2011[5] の論文で実験した条件の範囲でのみ確認されており、より大規模なネットワークでどうなるかは、今後検証が必要である。)

1. 各ノードが持つ親ノードの数 E が定数ならば、ノード数 n に対してアルゴリズムの１ステップが $O(n)$ の計算量ですむ。
2. 層構造をしたネットワークにおいて、層の深さが一定ならば、層内のノード数が増えても平均収束ステップ数はあまり変わらない。
3. 近似しない belief revision アルゴリズムと比較して、複雑なネットワークでも振動しにくい。

しかし、認識ステップで近似ＢＲを用いるには、以下の問題を解決する必要がある。

1. meaonOR モデルに適用可能か。
2. 制約ノード R のアイデアは導入可能か。
3. 入力ノードへの入力方法はどうすべきか。

以下の節で、これらの問題の解決方法について説明する。

10.2　meanOR　モデルでの近似 belief revision

10.2.1　meanOR モデルでの近似確率伝播アルゴリズム

まず、下記はオリジナルの Pearl の確率伝播アルゴリズムである。

$$
\begin{aligned}
BEL(x) &= \alpha\lambda(x)\pi(x) \\
\pi(x) &= \sum_{u_1,\cdots,u_m} P(x|u_1,\cdots,u_m)\prod_k \pi_X(u_k) \\
\lambda(x) &= \prod_l \lambda_{Y_l}(x) \\
\pi_{Y_l}(x) &= \beta_1\pi(x)\prod_{j\neq l}\lambda_{Y_j}(x) \\
\lambda_X(u_k) &= \beta_2\sum_x \lambda(x) \\
&\qquad \sum_{u_1,\cdots,u_m/u_k} P(x|u_1,\cdots,u_m)\prod_{i\neq k}\pi_X(u_i)
\end{aligned}
$$

ただし α, β_1, β_2 は正規化定数である。以下、正規化定数は省略する。5 章で述べたように meanOR モデルにおける条件付確率表の正規化定数 $1/Z$ も無視できるので、以下、省略する。

このアルゴリズムを、 meanOR モデル（式 (5.2)）を仮定したうえで、近似する。近似の方法は IJCNN2007[1] で述べたものと同じで、親ノードからのメッセージが下記のように正規化されているものとする。

$$\sum_{u_k}\pi_X(u_k) = 1 \tag{10.1}$$

・ $\pi(x)$ の計算式の変形：

$$\pi(x) = \sum_{u_1,\cdots,u_m} P(x|u_1,\cdots,u_m)\prod_i \pi_X(u_i)$$

$$= \sum_{u_1,\cdots,u_m} (\sum_k w(x,u_k)) \prod_i \pi_X(u_i)$$
$$= \sum_{u_1,\cdots,u_m} \sum_k w(x,u_k) \prod_i \pi_X(u_i)$$
$$= \sum_k \sum_{u_k} w(x,u_k)\pi_X(u_k) \sum_{u_1,\cdots,u_m/u_k} \prod_{i\neq k} \pi_X(u_i)$$
$$= \sum_k \sum_{u_k} w(x,u_k)\pi_X(u_k)$$

・ $\pi_{Y_l}(x)$ の近似：

$$\pi_{Y_l}(x) = \pi(x) \prod_{j\neq l} \lambda_{Y_j}(x)$$
$$\approx \pi(x) \prod_j \lambda_{Y_j}(x)$$
$$= \lambda(x)\pi(x)$$

・ $\sum_{u_1,\cdots,u_m/u_k} P(x|u_1,\cdots,u_m) \prod_{i\neq k} \pi_X(u_i)$ の近似：

$$\sum_{u_1,\cdots,u_m/u_k} P(x|u_1,\cdots,u_m) \prod_{i\neq k} \pi_X(u_i)$$
$$= \sum_{u_1,\cdots,u_m/u_k} (\sum_{j\neq k} w(x,u_j) + w(x,u_k)) \prod_{i\neq k} \pi_X(u_i)$$
$$= \sum_{u_1,\cdots,u_m/u_k} \sum_{j\neq k} w(x,u_j) \prod_{i\neq k} \pi_X(u_i)$$
$$+ w(x,u_k) \sum_{u_1,\cdots,u_m/u_k} \prod_{i\neq k} \pi_X(u_i)$$
$$\approx \sum_{u_1,\cdots,u_m} \sum_j w(x,u_j) \prod_i \pi_X(u_i) + w(x,u_k)$$
$$= \pi(x) + w(x,u_k)$$

・ $\lambda_X(u_k)$ の近似：

$$\lambda_X(u_k) = \sum_x \lambda(x) \sum_{u_1,\cdots,u_m/u_k} P(x|u_1,\cdots,u_m) \prod_{i\neq k} \pi_X(u_i)$$
$$\approx \sum_x \lambda(x)(\pi(x) + w(x,u_k))$$
$$= \sum_x (\lambda(x)\pi(x) + \lambda(x)w(x,u_k))$$

以上の結果を、整理すると、近似確率伝播アルゴリズムは以下のようになる。

$$\begin{aligned}
\lambda_{Y_l}^{t+1}(x) &= \sum_{y_l} (\rho^t(y_l) + \lambda^t(y_l) w(y_l,x)) \\
\lambda^{t+1}(x) &= \prod_{l=1}^{n} \lambda_{Y_l}^{t+1}(x) \\
\kappa_{U_k}^{t+1}(x) &= \sum_{u_k} w(x,u_k) BEL^t(u_k) \\
\pi^{t+1}(x) &= \sum_{k=1}^{m} \kappa_{U_k}^{t+1}(x) \\
\rho^{t+1}(x) &= \lambda^{t+1}(x)\pi^{t+1}(x) \\
Z_X^{t+1} &= \sum_x \rho^{t+1}(x) \\
BEL^{t+1}(x) &= \rho^{t+1}(x)/Z_X^{t+1} \qquad (10.2)
\end{aligned}$$

10.2.2 meanOR モデルでの近似 belief revision アルゴリズム

近似確率伝播アルゴリズムから近似ＢＲを得るには、IJCNN2011[5] と同様にアドホックな方法を用いる。ＣＰＴモデルに由来するもの以外の和演算を max 演算に置き換えることで近似ＢＲアルゴリズムを得る。

$$\begin{aligned}
\lambda_{Y_l}^{t+1}(x) &= \max_{y_l} (\rho^t(y_l) + \lambda^t(y_l) w(y_l,x)) \\
\lambda^{t+1}(x) &= \prod_{l=1}^{n} \lambda_{Y_l}^{t+1}(x) \\
\kappa_{U_k}^{t+1}(x) &= \max_{u_k} w(x,u_k) BEL^t(u_k) \\
\pi^{t+1}(x) &= \sum_{k=1}^{m} \kappa_{U_k}^{t+1}(x) \\
\rho^{t+1}(x) &= \lambda^{t+1}(x)\pi^{t+1}(x) \\
Z_X^{t+1} &= \max_x \rho^{t+1}(x) \\
BEL^{t+1}(x) &= \rho^{t+1}(x)/Z_X^{t+1} \\
x^* &= \operatorname*{argmax}_x BEL(x) \qquad (10.3)
\end{aligned}$$

10.2.3 性能評価

IJCNN2011[5] と同様の方法で近似ＢＲの簡単な性能評価を行ったところ、meanOR モデルの場合でも linear-sum モデルと同様の近似精度、速度を持つようである。

10.2.4 和演算を用いた BEL の正規化

トップダウンのメッセージ $BEL(x)$ を正規化するための定数 Z_X は、上で述べたように max 演算を用いて計算している。

$$Z_X = \max_x \rho(x) \tag{10.4}$$

しかし、 max 演算の代わりに和演算で「代用」することも考えられる。

$$Z_X = \sum_x \rho(x) \tag{10.5}$$

実は、この和演算を用いた正規化を行った近似ＢＲでも簡単な性能評価を行ったところ、max 演算の場合と変わらない近似精度、速度のようであった。もし性能があまり変わらないならば、 max 演算よりも和演算の方が生物にとっては実現が容易なので、生物はこちらを採用している可能性が高まる。

さらに、視覚野における多様な電気生理学的現象を説明する「注意の正規化モデル」[14] と同じ現象を近似ＢＲモデルで再現するためには、max による正規化ではなく和演算による正規化が必要だと、筆者は今のところ考えている。

和演算による正規化を用いた場合のより厳密な性能評価と、それによる「注意の正規化モデル」の再現は、今後の課題である。

10.2.5 生物学的妥当性

meanOR モデルに基づく近似ＢＲは、IJCNN2011[5] で述べた linear-sum モデルに基づく近似ＢＲと基本的構造は同じであり、やはり大脳皮質のコラム構造・６層構造とよく一致している。したがって、生物学的妥当性は高いと言える。

meanOR モデルではφ値ユニットが重みを学習する必要はないが、φ値ユニットが他のノードとやりとりするメッセージ計算は依然として必要である。したがって、「φ値ユニットのような応答をするニューロンが大脳皮質に見当たらない」という問題は、依然として残っている。

ただ、φ値ユニットメッセージは、ノード内で局所的に用いられるだけで、上位領野、下位領野には送る必要がない、という可能性がある。少なくとも、$BEL(x_\phi)$ は、 $\mathrm{argmax}_x(BEL(x))$ の計算のために必要だが、子ノードに送る必要はない。子ノードでは $w(y, x_\phi) = 0$ との掛け算に使われるだけだからである。同様に、なんらかの理由で $w(y_\phi, x) = P(y_\phi|x)$ も０なら $\lambda_{Y_l}(x_\phi)$ も親ノードに送らなくて済むことになる。

もしφ値ユニットメッセージは局所的にしか用いられないという性質を持つならば、φ値ユニットメッセージのような応答をするニューロンが見つかりにくい理由になるかもしれない。

なお、神経回路では $w(y, x_\phi)$ というシナプスはノード X 側と Y 側に２重に保持される（参考：IJCNN2011[5]）ので、 $w(y, x_\phi) = 0$ でないと「φ値ユニットは重みを学習する必要がない」と完全には言えない。その意味でも $w(y, x_\phi) = 0$ とする「 meanOR モデルの改良版」の検討が必要であろう。

10.3 belief revision とスパース符号化

10.3.1 ペナルティ付きの belief revision

７章で述べたようなノード活性度へのペナルティを、belief revision アルゴリズムに導入することは、容易である[1]。具体的方法を以下に述べる。

ノード活性度へのペナルティは、他のノードの値とは無関係に決まるので、ペナルティの値を以下、 $R(x, \mathbf{x})$ ではなく $R(x)$ と書くことにする。

まず、下記はオリジナルの Pearl の belief revision である。（Pearl 本とは表記が異なるが、本質的に同じものである。 ）

$$\begin{aligned}
x^* &= \mathrm{argmax}_x BEL(x) \\
BEL(x) &= \alpha\lambda(x)\pi(x) \\
\pi(x) &= \max_{u_1,\cdots,u_m} P(x|u_1,\cdots,u_m)\prod_k \pi_X(u_k) \\
\lambda(x) &= \prod_l \lambda_{Y_l}(x) \\
\pi_{Y_l}(x) &= \beta_1\pi(x)\prod_{j\neq l}\lambda_{Y_j}(x)
\end{aligned}$$

[1]なお、筆者のこれまでの論文では、スパース符号化と belief revision をまだ同時には用いていなかった。ICONIP2010[4] ではスパース符号化のアイデアを実装したが、認識アルゴリズムは山登り法によるＭＰＥ計算であった。IJCNN2011[5] では belief revision を用いた自然画像の学習結果を載せたが、そこでは全ノードが常に活性ノードであった。

$$\begin{aligned}\lambda_X(u_k) &= \beta_2 \max_x \lambda(x) \\ &\quad \max_{u_1,\cdots,u_m/u_k} P(x|u_1,\cdots,u_m)\prod_{i\neq k}\pi_X(u_i)\end{aligned}$$

いま、分かりやすくするために、ノード R の親ノードが X_1, X_2 の２つしかないと仮定する。ノード R からノード X_1 へのメッセージ $\lambda_R(x_1)$ は以下のように書ける。

$$\lambda_R(x_1) = \beta_1 \max_r \lambda(r) \max_{x_2} P(r|x_1,x_2)\pi_R(x_2) \quad (10.6)$$

式 (3.3) で定義されたように、ノード R の条件付確率は以下のようになる。

$$P(R=1|x_1,x_2) = \frac{1}{Z}R(x_1)R(x_2) \quad (10.7)$$

ＭＰＥ計算の際には観測値 $R=1$ が与えられる。Pearl の本 [6] p.256 に従えば、ノード R は、R のダミーの子ノード Z から $\lambda_Z(R=1)=1, \lambda_Z(R=0)=0$ というメッセージを受け取ると考えてよい。すると

$$\lambda(R=1)=1, \lambda(R=0)=0 \quad (10.8)$$

となる。これをもとに $\lambda_R(x_1)$ を整理すると、以下のようになる。

$$\begin{aligned}&\lambda_R(x_1) \\ &= \beta_1 \max_r \lambda(r) \max_{x_2} P(r|x_1,x_2)\pi_R(x_2) \\ &= \beta_1 max(\\ &\lambda(R=0) \max_{x_2} P(R=0|x_1,x_2)\pi_R(x_2), \\ &\lambda(R=1) \max_{x_2} P(R=1|x_1,x_2)\pi_R(x_2)) \\ &= \beta_1 \max_{x_2} P(R=1|x_1,x_2)\pi_R(x_2) \\ &= \beta_1 \max_{x_2}((1/Z)R(x_1)R(x_2))\pi_R(x_2) \\ &= ((\beta_1/Z)\max_{x_2} R(x_2)\pi_R(x_2))R(x_1)\end{aligned} \quad (10.9)$$

ここで、$(\beta_1/Z)\max_{x_2} R(x_2)\pi_R(x_2)$ は x_1 の値によらない定数なので、無視しても belief revision の動作に影響はない。（一般に、 belief propagation でも belief revision でも、あるノードへのメッセージとして送る数値ベクトルは定数倍しても結果は変わらない。結果は最終的に正規化されるからである。）したがって、メッセージ $\lambda_R(x_1)$ は以下のように定義してもかまわないことになる。

$$\lambda_R(x_1) = R(x_1) \quad (10.10)$$

以上の議論は、ノードが２つより多くても全く同様に成り立つ。したがって、メッセージ $\lambda_R(x)$ は一般に以下のようになる。

$$\lambda_R(x) = R(x) \quad (10.11)$$

以上の結果に基づくと、制約条件 $R(x)$ を加えたベイジアンネットにおける belief revision アルゴリズムは、下記のようになる。（追加部分を下線で示した。）

$$\begin{aligned}x^* &= \operatorname*{argmax}_x BEL(x) \\ BEL(x) &= \alpha\lambda(x)\pi(x) \\ \pi(x) &= \max_{u_1,\cdots,u_m} P(x|u_1,\cdots,u_m)\prod_k \pi_X(u_k) \\ \lambda(x) &= \underline{R(x)}\prod_l \lambda_{Y_l}(x) \\ \pi_{Y_l}(x) &= \beta_1\pi(x)\underline{R(x)}\prod_{j\neq l}\lambda_{Y_j}(x) \\ \lambda_X(u_k) &= \beta_2 \max_x \lambda(x) \\ &\quad \max_{u_1,\cdots,u_m/u_k} P(x|u_1,\cdots,u_m)\prod_{i\neq k}\pi_X(u_i)\end{aligned} \quad (10.12)$$

10.2 節で述べた近似ＢＲ（式 (10.3)）に対して同様の考えを適用すると、アルゴリズムは下記のようになる。

$$\begin{aligned}\lambda_{Y_l}^{t+1}(x) &= \max_{y_l}(\rho^t(y_l) + \lambda^t(y_l)w(y_l,x)) \\ \lambda^{t+1}(x) &= \underline{R(x)}\prod_{l=1}^n \lambda_{Y_l}^{t+1}(x) \\ \kappa_{U_k}^{t+1}(x) &= \max_{u_k} w(x,u_k)BEL^t(u_k) \\ \pi^{t+1}(x) &= \sum_{k=1}^m \kappa_{U_k}^{t+1}(x) \\ \rho^{t+1}(x) &= \lambda^{t+1}(x)\pi^{t+1}(x) \\ Z_X^{t+1} &= \max_x \rho^{t+1}(x) \\ BEL^{t+1}(x) &= \rho^{t+1}(x)/Z_X^{t+1} \\ x^* &= \operatorname*{argmax}_x BEL(x)\end{aligned} \quad (10.13)$$

10.3.2 性能評価

このアルゴリズムが正しいかどうかを、IJCNN2011[5] と同様の性能評価をすることで確かめた。

ノードの値に対してスパース性制約が加えられる場合、$R(x)$ の定義は次のようになる。

$$R(x_i) = \begin{cases} 1 & (i = \phi) \\ e^{-\beta} & (i \neq \phi) \end{cases} \tag{10.14}$$

この式に従って厳密解と比較して近似精度を評価したところ、近似しない belief revision アルゴリズム（式 (10.12)）では、ＭＰＥの非常によい近似解が得られることを確認した。

一方、近似ＢＲ（式 (10.13)）については、３層以上のネットワークに対しては、近似しない belief revision よりも精度が悪いものの、あまり悪くない近似解が得られる。しかし、２層のネットワークについては非常に悪くほとんどランダムに見える値が推定されてしまうという現象が観察された。原因はよくわからないが、２層かつスパースだと、「入力ノードの値が複数の親ノードの値から決定される」という、近似ＢＲの仮定が成り立たなくなるせいかもしれない。いずれにせよ、実際の脳は多層なので、２層での近似精度の悪さは問題にならないと思われる。重要なのは、脳のような大規模なネットワークで、脳と似た条件のネットワークにアルゴリズムを適用したときに、実用上十分な精度が出るかどうかである。この点について、今後より大規模なネットワークで評価していく必要がある。

10.3.3 生物学的妥当性

ペナルティを含めた近似ＢＲアルゴリズムを実行する神経回路は、IJCNN2011[5] の神経回路に対するわずかな修正で実現可能なため、IJCNN2011[5] の神経回路と同じ程度の生物学的妥当性を持つと言えるだろう。

10.4 belief revision と側抑制ＩＣＡの統合に向けて

この節では、belief revision アルゴリズムと 8 章で述べた側抑制ＩＣＡアルゴリズムとを統合するアイデアについて述べる。なお、このアイデアはまだ実装・評価を行っていない。

10.4.1 共通子ノードＲを使う方法

まず、8 章で述べた制約ノードＲを素直に用いた場合の $\lambda_R(x)$ を導き、計算量的に問題があることを示す。

ここでは、式を簡単にするためにノードが X1, X2, X3 の３つの場合を考える。これら３つのノードが共通の子ノード R を持つとする。ノード R の条件付確率は以下のようになる。

$$\begin{aligned} & P(R = 1|x_1, x_2, x_3) \\ = \ & \frac{1}{Z} R(x_1, x_2, x_3) R(x_2, x_3, x_1) R(x_3, x_1, x_2) \end{aligned} \tag{10.15}$$

ただし、 $R(x_1, x_2, x_3)$ は、ノード X2, X3 から X1 への側抑制の総和とする。

前節と同様に、 $\lambda_R(x_1)$ を整理すると、以下のようになる。

$$\begin{aligned} & \lambda_R(x_1) \\ = \ & \beta_1 \max_r \lambda(r) \max_{x_2, x_3} P(r|x_1, x_2, x_3) \pi_R(x_2) \pi_R(x_3) \\ = \ & \beta_1 \max_{x_2, x_3} P(r|x_1, x_2, x_3) \pi_R(x_2) \pi_R(x_3) \\ \propto \ & \max_{x_2, x_3} R(x_1, x_2, x_3) R(x_2, x_3, x_1) R(x_3, x_1, x_2) \\ & \pi_R(x_2) \pi_R(x_3) \end{aligned} \tag{10.16}$$

この式はこれ以上簡単にならず、ノード数 n に対しメッセージ１つを計算するために $O(2^n)$ の計算量が必要なり、スケーラブルではない。

10.4.2 ２つのノードごとに共通子ノードを持たせる方法

次に、２つのノードごとに、共通の子ノードを持たせる方法を考える（図 10.1）。互いに側抑制するノードが n 個ある場合、子ノードの数は $n(n-1)/2$ 個になる。

今、ノード X1 と X2 の間の共通子ノードを S_{12} とする。２つのノード間の抑制の強さは対称で、以下の式で表されるとする。

$$P(S_{12} = 1|x_1, x_2) = \frac{1}{Z} S(x_1, x_2) \tag{10.17}$$

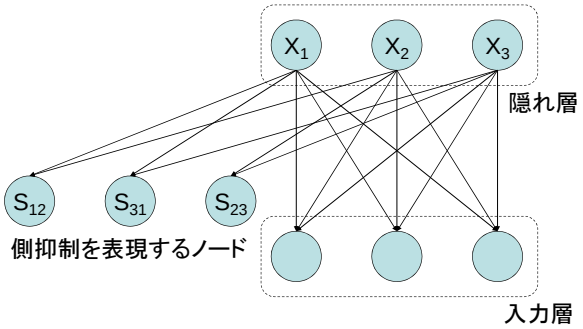

図 10.1: ２つのノード X_i, X_j が互いに側抑制する状況を、２つの間に共通の子ノード S_{ij} があるベイジアンネットで表現する。

このときメッセージ $\lambda_{S_{12}}(x_1)$ は、以下のようになる。

$$\begin{aligned}
&\lambda_{S_{12}}(x_1) \\
&= \beta_1 \max_s \lambda(s) \max_{x_2} P(S_{12} = s|x_1, x_2)\pi_{S_{12}}(x_2) \\
&\propto \max_{x_2} P(S_{12} = 1|x_1, x_2)\pi_{S_{12}}(x_2) \\
&\propto \max_{x_2} S(x_1, x_2)\pi_{S_{12}}(x_2) \qquad (10.18)
\end{aligned}$$

この場合は計算量が極めて少なく、神経回路での実現も容易な計算式となる。

なお、近似ＢＲにこのアイデアを適用する場合は、$\pi_{S_{12}}(x_2)$ の代わりに $BEL(x_2)$ または $\rho(x_2)$ を使う。

$\rho(x_2)$ を使う場合は IJCNN2011[5] の神経回路に対応付けると３層から３層への結合になる。実際の大脳皮質では、３層から同じ方位選択性を持つ（言いかえればヘブ則により結ばれる）別のコラムの３層への結合があることが知られており、それがＩＣＡのための側抑制である可能性がある。

10.4.3 計算量と近似精度に関する考察

共通の子ノード R を持たせる方法では、もとのベイジアンネットの形によっては、ノード R を追加してもネットワークが loop を持たない場合がある。その場合は belief revision は厳密解を計算できる。しかしながら、メッセージ計算の計算量は上に書いたように $O(2^n)$ となる。

一方、２つのノードごとに子ノード S_{ij} を持たせる方法では、必然的にネットワークに loop が生じるため、 loopy belief revision になる。メッセージ１つを計算する計算量は $O(1)$ 、$n(n-1)/2$ 個の子ノードの導入で増えるメッセージの数は $O(n^2)$ 個である。もし収束までに数ステップの反復が必要だとしても、計算量は $O(2^n)$ よりは圧倒的に少ない。もともと BESOM では loop のあるベイジアンネットでの高速な近似解法を目指しているので、こちらの方が好ましい性質を持っていると言える。

問題は、この方法で実用上十分な近似精度が得られるかどうか、である。特に、 loopy belief revision は収束の保証がないため、振動が心配される。これについては、今後実験により検証していく必要がある。

10.5 入力の与え方

6 章で述べた入力データの与え方は、belief revision アルゴリズムにも適用可能であるが、6.4 節で述べたように生物学的妥当性に問題があり、別の方法を考える必要がある。

神経科学的知見によれば感覚器からの入力は単なる中継器である視床[2]を経由して一次感覚野の４層に入力される。このことを素直に BESOM モデルに当てはめれば、感覚器からの入力は $\lambda_{Y_l}(x)$ メッセージとして入力されてくると解釈すべきである。この解釈で入力データの学習・認識が正しく動作するかどうかは今後、実験により検証する必要がある。

10.6 今後

本章で述べた近似ＢＲは、大脳皮質の認識アルゴリズムのモデルの最終版というわけではない。あくまでも第一歩にすぎず、今後も性能向上や生物学的妥当性の向上のために修正される可能性がある。

[2]TR2008[2] の p.27 で、視床は「値」から「値の確率分布」への変換を行うのではないかという予想を書いたが、視床にそのような機能はなく、単純に信号を中継するだけらしい。

第11章　まとめと今後

本文章で述べたすべてのアイデアを、次期バージョンの BESOM で１つの実装にまとめ、さらに強化学習や時系列学習などいくつかの機能を追加すれば、大脳皮質のアルゴリズムのもっとも基本的な機能のモデル（仮説）が一応の完成を見せるのではないかと、今のところは考えている。

しかし、その先にも、人間のような知能を再現させるためにやるべきことは多い。

大脳皮質の基本機能が計算機上で再現できればその次は、大脳皮質が実現する様々な高次機能を再現する必要がある。高次機能とは具体的には、思考や言語理解などである。そのためには、これらの高次機能を実現している前頭前野や言語野の領野間の接続構造を BESOM を使って再現した上、領野ごとに固有の事前知識を作り込む必要があるだろう。

高次機能の再現の見込みが立てばその次は、脳の大脳皮質以外の組織、特に海馬、小脳、扁桃体のモデルと大脳皮質モデルとを統合し、脳全体の機能を再現することが必要になる。これらの組織のモデルの研究はそれなりに進んでいるが、大脳皮質のモデルとの統合の方法は自明ではない。ここでも、機械学習理論の深い知識と多くの神経科学的知見が必要となるだろう。

脳全体の機能が再現できればその次は、知能を持つロボットが「人間の役に立とう」という意思を持つように、情動を設計する必要がある。生物には子孫を残そうとする意思を持つような情動が作り込まれているが、ロボットに作り込むべき情動はそれとは大きく異なることになる。

TR2008[2] の１３章で指摘したように、知能の高いロボットには有用性もあるが危険性もあり、それに対する対策も早いうちに考えておく必要がある。

以上のように「知能の高い役に立つロボット」の実現に向けた道筋はほぼ見えているにも関わらず、そのことが世の中にほとんど知られておらず、この目標に向けて真剣に取り組む研究者が極めて少ない状況が、依然として続いているのは大変残念なことである。

参考文献

[1] Yuuji ICHISUGI, The cerebral cortex model that self-organizes conditional probability tables and executes belief propagation, In Proc. of International Joint Conference on Neural Networks (IJCNN 2007), pp.1065–1070, Aug 2007.
http://staff.aist.go.jp/y-ichisugi/besom/20070509ijcnn-paper.pdf

[2] 一杉裕志、「脳の情報処理原理の解明状況」産業技術総合研究所テクニカルレポート AIST07-J00012, Mar 2008.
http://staff.aist.go.jp/y-ichisugi/besom/AIST07-J00012.pdf

[3] 一杉裕志、「大脳皮質のアルゴリズム BESOM Ver.1.0」, 産業技術総合研究所テクニカルレポート AIST09-J00006, Sep 2009.
http://staff.aist.go.jp/y-ichisugi/besom/AIST09-J00006.pdf

[4] Yuuji Ichisugi, Haruo Hosoya: Computational Model of the Cerebral Cortex that Performs Sparse Coding Using a Bayesian Network and Self-Organizing Maps, In Proc. of 17th International Conference on Neural Information Processing (ICONIP 2010), Part I, LNCS 6443, pp.33–40, Nov 2010.
http://staff.aist.go.jp/y-ichisugi/besom/2010iconip.pdf

[5] Yuuji Ichisugi: "Recognition Model of Cerebral Cortex based on Approximate Belief Revision Algorithm", To appear in Proc. of IJCNN 2011.
http://staff.aist.go.jp/y-ichisugi/besom/2011ijcnn.pdf

[6] J. Pearl , Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, 1988.

[7] T. Kohonen, Self-Organizing Maps. Springer-Verlag, 1995.

[8] T. コホネン, 自己組織化マップ（改訂版）, シュプリンガー・フェアラーク東京, 2005.（[7] の邦訳。）

[9] Aapo Hyvarinen, Juha Karhunen, Erkki Oja, Independent Component Analysis, Wiley-Interscience, 2001.

[10] A. ビバリネン, E. オヤ and J. カルーネン, 詳解独立成分分析, 東京電機大学出版局, 2005.（[9] の邦訳。）

[11] Olshausen BA, Field DJ, Emergence of simple-cell receptive field properties by learning a sparse code for natural images, NATURE 381 (6583): 607-609 JUN 13 1996.

[12] Daniel D. Lee and H. Sebastian Seung, Learning the parts of objects by non-negative matrix factorization Nature 401, 788-791 (21 October 1999).

[13] 田尻 隆, 倉田 耕治: 二つの 1 次元 SOM の結合による独立成分分析と主成分分析, 電子情報通信学会技術研究報告 ニューロコンピューティング研究会, Vol.104, No.139(20040617) pp. 61-66, 2004.

[14] Reynolds JH, Heeger DJ: The normalization model of attention, Neuron. 2009 Jan 29;61(2):168-85.

大脳皮質のアルゴリズム BESOM Ver.2.0
産業技術総合研究所テクニカルレポート AIST11-J00009
2011 年 9 月 30 日
独立行政法人 産業技術総合研究所
〒305-8568　茨城県つくば市梅園 1-1-1　中央第 2
TEL：029-861-2000