人　工　知　能 94－3
（1994．6．20）

競合的な局所基底ニューラルネットワークへの規則の埋め込みと精密化

岡 夏樹　　金道敏樹
松下技研(株) ヒューマンインタフェース研究所
oka@mrit.mei.co.jp　kindo@mrit.mei.co.jp

学習すべき入出力関係について、あらかじめ分かっているif-then規則の形式の情報を、競合的な局所基底ニューラルネットワークQCNetの初期構造として自然に埋め込むことができることを示す。次に、この初期構造から出発してQCNetにおいて例からの学習を行うことにより、埋め込んだ規則のif部とthen部の修正、必要な規則の追加、冗長な規則や正しくない規則の削除のそれぞれに相当する処理を行えることを準備的な実験によって示す。さらに、分かりやすい学習結果を得、また、ノイズを含む訓練例にも対応できるように、QCNetにおける不要な素子を削除するアルゴリズムを改良する方法を提案する。

Refining, Adding, and Deleting Rules with a Competitive Radial Basis Neural Network

Natsuki Oka　　Toshiki Kindo
Human Interface Research Laboratory
Matsushita Research Institute Tokyo, Inc.
3-10-1, Higashimita, Tama-ku, Kawasaki 214, Japan

Given if-then rules are naturally converted into QCNet, a competitive radial basis neural network, which is then trained with examples. Preliminary experiments demonstrate that the converted rules are refined, new nodes, which correspond to new rules, are added, and redundant or incorrect nodes, which correspond to redundant or incorrect rules, are deleted during training. In order to improve the comprehensibility of the results and to handle noisy examples, proposed is a better algorithm that deletes units in QCNet.

1 まえがき

エキスパートシステムなどの知識システムの構築に際しては、次の2種類の情報が利用できることが多い。1つは専門家からのインタビューなどにより獲得した規則などの形式の一般化された情報であり、もう1つは一般化していない例そのままの情報である。一般には前者は完全に正しいとは限らないし、後者はノイズを含む可能性がある。

このような完全に正しいとは限らない規則の形式の情報とノイズを含む可能性がある例の情報とから学習を行う手法は数多く提案されているが、それらのうちの1種として、まず規則の形式の情報をニューラルネットワークの初期構造に変換し、続いてそれを出発点としてニューラルネットワークで例からの学習を行なうことにより規則を精密化する方法 [14, 8, 3, 11, 6, 7] がある。

これらの方法では、規則の形式の情報を利用してニューラルネットワークの初期構造をうまく作ってやることにより、一般的にニューラルネットワークの欠点であるとされる、訓練例が極めて多数必要になる点や、学習の収束の遅さや、学習が望ましい状態に収束するとは限らない (すなわち、極小解に陥ることがある) といった点を改善できることが報告されている。

しかしながら、それらの方法では、以下に示すように、規則の形式の情報とニューラルネットワークの形式の情報の対応関係が自然でないため、情報の変換が不正確であったり [14, 8, 3, 11]、学習の可能性を制限したりしなければならない [6, 7] という問題点 (問題点 A と呼ぶ) があった。なお、初期の研究では、規則の追加や削除の手段が無かったり貧弱であったりした [14, 8, 6] が、最近の研究では、それらの手段を持つもの [3, 11, 7] が増えている。

これらのうち [14, 8, 3, 11] では、論理積や論理和を階層型ニューラルネットワークの通常の素子1個で表現する (すなわち、線形加重和とシグモイド関数のような微分可能な閾値的関数を用いて表現する)。たとえば論理積であれば、すべての入力が1であるときだけ出力が1になるように各入力の重みや出力の閾値を設定する。しかし、「論理演算」と「線形加重和と閾値的関数による演算」の両者は本来異なる性質を持つため、これは等価な変換ではない。両者の違いが問題になるのは論理積や論理和への入力線の数が多い場合であり、これを線形加重和と閾値的関数で表現すると、入力の値の少しの変動が多数重なって、論理的な出力とは異なる出力が生じる可能性がある (すなわち、入力値の変動への余裕が少くなる)。

これに対して、[6, 7] では、論理演算を不連続な入出力関数 (MIN や MAX 演算) を持つ素子で表現する。この場合上記の意味で変換が不適切であるという問題はなくなるが、入出力関数がすべての領域で微分可能ではないため通常の階層型ニューラルネットワークの特長である最急降下法による学習が使えなくなる。そこで、[6, 7] では、MIN や MAX 演算をする素子への入力の重みは固定して学習しないようにするなどの方法により、誤差逆伝播学習 [13] 風の学習をすることを可能にした。しかし、この方法では、学習により調節可能な重みは少くなるので、与えられたニューラルネットワークの初期構造によっては、最終的に望ましい学習結果に至らない可能性が増えるという問題点が生じる。

一方、Denoeux ら [2] は、もとの入力空間上の入力ベクトルを1次元大きい空間上の単位長のベクトルに正規化することにより、大域的な基底関数 (シグモイド関数) を持つ通常の階層型のニューラルネットワークが、局所的な基底関数を持つニューラルネットワーク [9, 12](以後、局所基底ニューラルネットワークと呼ぶ) と同様の働きをするようにした。すなわち、中間層の各素子が、それぞれの反応中心からの距離に応じた局所的な広がりを持つ出力を出すようにした。また、彼らは、規則の形式の情報を使うのではなく、訓練例から典型的な例を選び出す方法を提案し、その選び出した典型的な例の情報を使って、ニューラルネットワークの初期構造を作った。すなわち、ニューラルネットワークの中間層の各素子の反応中心をそれぞれの典型的な例の位置に初期化した。

彼らの方法は、初期構造を作るための情報 (典型的な例) とニューラルネットワークの構造の対応が自然であるため、上述の問題点 A は有しない。しかし、局所基底ニューラルネットワークは一般的に、学習すべき入出力関係の局所的な構造を素早く捉えることができるが、大域的に出力が一定であるというような単純な構造を捉える能力

は低く (問題点 B と呼ぶ)、彼らの初期構造化法を使った場合でも、この問題点 B は残る。

さて、著者のうちの一人が提案した、競合的な局所基底ニューラルネットワーク QCNet(2節に概要を紹介する) は、通常の局所基底ニューラルネットワークの中間層の素子間に競合関係を導入することにより、学習が速いなどの局所基底ニューラルネットワークの特長を残したままで、上述した問題点 B の軽減を実現したものである [4]。しかしながら [4] は、必要な素子数が訓練例の提示順に依存して変動し、また、冗長な素子も生じるという問題点を有していたので、[5] では冗長な素子を削除する手続きを導入することにより、これらの問題点の軽減を図った。

本論文では、QCNet の構造と if-then 規則の形式の情報とを自然に対応付けることができる点に注目し、学習すべき入出力関係について、あらかじめ分かっている規則の形式の情報を QCNet の初期構造として自然に埋め込む方法を示す (3.1節)。次に、この初期構造から出発して QCNet において例からの学習を行うことにより、埋め込んだ規則の if 部と then 部の修正、必要な規則の追加、冗長な規則や正しくない規則の削除のそれぞれに相当する処理を行う (3.2節) が、適切な初期構造を与えることによって、訓練例が少い時点での正答率を向上できることを準備的な実験により示す (4節)。さらに、分かりやすい学習結果を得、また、ノイズを含む訓練例にも対応できるように、QCNet における不要な素子を削除するアルゴリズムを改良する方法を提案する (5節)。

2 競合的な局所基底ニューラルネットワーク QCNet

2.1 QCNet の構造

競合的な局所基底ニューラルネットワーク QCNet は、入力層、中間層、および出力層の各層に、それぞれ n_i、n_m、n_o 個の素子を持つ。ただし、中間層の素子数は学習により適応的に変化する。入力信号 $\boldsymbol{x}$ の各成分を受け取った入力層の素子は、それをそのまま出力する。

中間層の素子 b は入力信号ベクトル $\boldsymbol{x}$ を受け取り、反応強度

$$B_b(\boldsymbol{x}) = S_b(\boldsymbol{x}) / \sum_{a=1}^{n_m} S_a(\boldsymbol{x}) \tag{1}$$

を出力する。ここで、$S_a(\boldsymbol{x})$ は、

$$S_a(\boldsymbol{x}) = \exp(-\frac{(\boldsymbol{x}-\mu_a)^2}{2\sigma_a^2}) \tag{2}$$

で与えられる、中間層の素子 a が単体時に出力する単体反応強度である。ここに、μ_a は素子 a の反応中心、σ_a は素子 a の反応幅である。式 (1) から分かるように、中間層の各素子は競合関係にあり、それらの反応強度の和は全ての入力に対して 1 に保たれる。

出力層の素子 c は、中間層の素子 b との間の結合加重 v_b^c によって重みづけられた中間層の各素子の反応強度を受け、その線形和を出力する。すなわち、このネットワークの出力信号 y は、

$$y = \sum_{b=1}^{n_m} v_b B_b(\boldsymbol{x}) \tag{3}$$

で与えられる。以上のパラメータ μ、σ、v は学習によって調整されるが、以下ではこれらのパラメータを ξ によってひとまとめに表現する場合がある。

なお、3.1節で後に述べるように、if-then 規則の形式の情報に基づく QCNet の初期構造設定では、1 つの規則は中間層の 1 つの素子に変換される。すなわち、ある if-then 規則の if 部が成立するときに、その規則に対応する素子 b の反応強度 $B_b(\boldsymbol{x})$ が大きくなるように設定し、また、その規則の then 部にかかれた出力の値を結合加重 v_b の値とする。 QCNet の中間層の各素子が上記の意味で競合関係にあることが、規則の形式の情報との自然な対応付けを可能にしている。

2.2 QCNet の学習

2.2.1 確率的降下法

QCNet の出力を 2 乗誤差によって評価することにすれば、損失関数 $r(\boldsymbol{x}, \xi)$ は、

$$r(\boldsymbol{x}, \xi) = \frac{1}{2}\{y(\boldsymbol{x}, \xi) - z(\boldsymbol{x})\}^2 \tag{4}$$

と書ける。ここで、z は教師信号である。この損失関数 $r(\boldsymbol{x}, \xi)$ は、ネットワークのパラメータ ξ によって微分可能である。平均 2 乗誤差を極小にする学習則として確率

的降下法 [1] を採用すると、パラメータの更新則は、各訓練例に対して、

$$\boldsymbol{\xi}(t+1) = \boldsymbol{\xi}(t) - \epsilon \frac{\partial r(\boldsymbol{x}, \boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \quad (5)$$

$$= \boldsymbol{\xi}(t) - \epsilon \sum_{c=1}^{n_o} (y_c - z_c) \nabla_{\boldsymbol{\xi}} y_c \quad (6)$$

で与えられる。ここに ϵ は学習係数 (十分小さい正の定数) である。$\nabla_{\boldsymbol{\xi}}$ は $\boldsymbol{\xi}$ についての gradient であり、$\nabla_{\boldsymbol{\xi}} y_c$ を各パラメータごとに書き下すと、

$$\frac{\partial y_c}{\partial v_b^i} = \delta_{ci} B_b(\boldsymbol{x}), \quad (7)$$

$$\nabla_{\boldsymbol{\mu}_b} y_c = \frac{(\boldsymbol{x} - \boldsymbol{\mu}_b)}{\sigma_b^2} B_b(\boldsymbol{x})(v_b^c - y_c), \quad (8)$$

$$\frac{\partial y_c}{\partial \sigma_b} = \frac{(\boldsymbol{x} - \boldsymbol{\mu}_b)^2}{\sigma_b^3} B_b(\boldsymbol{x})(v_b^c - y_c) \quad (9)$$

となる。

これらの式から分かるように確率的降下法による学習では、中間層の 1 個の素子 b が支配的である領域内の訓練例によっては主にその素子の結合荷重 $\boldsymbol{v}_b$ の学習が進み、中間層の複数の素子が出力に影響を及ぼす領域内の訓練例によってはそれらの素子の結合加重 v、反応中心 μ、および反応幅 σ の学習が進むようになっている。

なお、3.1節で後に述べる方法によって、if-then 規則の形式の情報に基づく QCNet の初期構造設定をした場合を考えると、本節で述べた確率的降下法による学習は、埋め込んだ規則の if 部の条件と then 部の出力値を出力誤差が小さくなる方向に微調整することに相当する。

2.2.2 新しい素子の追加

本節では、必要に応じて中間層に新しい素子を追加する方法を記述する。

新たに追加する $(n_m + 1)$ 個めの素子のパラメータを

$$\boldsymbol{\mu}_{n_m+1} = \boldsymbol{x}, \quad (10)$$

$$\boldsymbol{v}_{n_m+1} = \boldsymbol{y}^{(n_m)}(\boldsymbol{x}) - \frac{\boldsymbol{y}^{(n_m)}(\boldsymbol{x}) - \boldsymbol{z}(\boldsymbol{x})}{B_{n_m+1}^{(n_m+1)}(\boldsymbol{x})} \quad (11)$$

と設定することで、すでにあった n_m 個の素子の状態を変えることなく、入力信号 $\boldsymbol{x}$ に対する誤差を 0 にすることができる。ここに $\boldsymbol{y}^{(n_m)}$ は中間層の素子数が n_m 個である QCNet の出力、$B_{n_m+1}^{(n_m+1)}$ は中間層の素子数が (n_m+1) 個である QCNet の (n_m+1) 番目の素子の反応強度である。

反応幅 σ_{n_m+1} は、誤差 0 の条件だけからは定まらないので、追加する素子の影響が適切な大きさの局所的な範囲に限られるように、中間層の他の素子の位置や反応強度に基づいてヒューリスティックに決める。すなわち、

$$\sigma_{n_m+1}^2 = \gamma \sum_{a=1}^{n_m} (\boldsymbol{x} - \boldsymbol{\mu}_a)^2 B_a^{(n_m)}(\boldsymbol{x}) \quad (12)$$

とする。ここに γ は反応幅の局所性を決める正の定数であり、γ が小さいと、出力値が異なる隣接する素子間における QCNet の出力の変化が急峻になる。

さて、ある訓練例 $(\boldsymbol{x}_{(t)}, \boldsymbol{z}(\boldsymbol{x}_{(t)}))$ に対して、

A) 式 (6)(7)(8)(9) に従って確率的降下法によりパラメータを修正する。

B) 式 (10)(11)(12) に従って新しく素子を追加する。

のいずれを行うかを決定するために、モデルの大きさ (ここでは中間層の素子数 n_m でモデルの大きさを表現する) も考慮した損失関数

$$L(\boldsymbol{x}, \boldsymbol{\xi}, n_m) = \frac{1}{2}\{\boldsymbol{y}(\boldsymbol{x}, \boldsymbol{\xi}) - \boldsymbol{z}(\boldsymbol{x})\}^2 + \eta n_m \quad (13)$$

を考える。この損失関数の第 1 項は誤差項であり、第 2 項はモデルの大きさを示すモデル損失項である。η はモデル損失項の強さを決める正の定数である。

A) と **B)** のうち、損失関数 $L(\boldsymbol{x}, \boldsymbol{\xi}, n_m)$ の減少が大きくなる方を選択すればよいが、確率的降下法による 1 回のパラメータ修正によっては出力誤差は僅かしか変化しないと考えてよいから、新しく素子を追加したときの損失関数 $L(\boldsymbol{x}, \boldsymbol{\xi}, n_m)$ の変化が負である場合に、すなわち、

$$\frac{1}{2}\{\boldsymbol{y}(\boldsymbol{x}_{(t)}, \boldsymbol{\xi}) - \boldsymbol{z}(\boldsymbol{x}_{(t)})\}^2 > \eta \quad (14)$$

の場合に **B)** を行い、そうでない場合に **A)** を行うとしてかまわない。

なお、初期学習は、最初の 2 つの訓練例 $(\boldsymbol{x}_{(1)}, \boldsymbol{z}(\boldsymbol{x}_{(1)}))$ と $(\boldsymbol{x}_{(2)}, \boldsymbol{z}(\boldsymbol{x}_{(2)}))$ を用いて、

$$\boldsymbol{\mu}_1 = \boldsymbol{x}_{(1)}, \quad (15)$$

$$\boldsymbol{v}_1 = \boldsymbol{z}(\boldsymbol{x}_{(1)}), \quad (16)$$

$$\boldsymbol{\mu}_2 = \boldsymbol{x}_{(2)}, \quad (17)$$

$$\boldsymbol{v}_2 = \boldsymbol{y}^{(1)}(\boldsymbol{x}_{(2)}) - \frac{\boldsymbol{y}^{(1)}(\boldsymbol{x}_{(2)}) - \boldsymbol{z}(\boldsymbol{x}_{(2)})}{B_2^{(2)}(\boldsymbol{x}_{(2)})}, \quad (18)$$

$$\sigma_1^2 = \sigma_2^2 = \gamma(\boldsymbol{x}_{(1)} - \boldsymbol{x}_{(2)})^2 \quad (19)$$

とする。ここにγは反応幅の局所性を決める正の定数であって、式(12)に現れるγと同じ値とする。

ところで、3.1節で後に述べる方法によって、if-then規則の形式の情報に基づくQCNetの初期構造設定をした場合を考えると、本節で述べた新しい素子の追加は、あらかじめ分かっていた規則の微調整では説明できない例を扱うための新たな規則の追加に相当する。

2.2.3 冗長な素子の削除

前節までの学習則によって高速な学習を実現できるが、中間層において必要な素子数が学習例の提示順に依存して変動し、冗長な素子も生じるという問題点を有する。そこで[5]では、中間層の冗長な素子を削除する次のような削除アルゴリズムAを提案した。

削除アルゴリズムA: 中間層のn_m個の素子のうちのある素子bの削除による、素子bの反応中心μ_bにおける出力信号の変化$\delta y(\mu_b)$が

$$\frac{1}{2}(\delta y(\mu_b))^2 < \theta \qquad (20)$$

を満たすならば、素子bは冗長であると見なして削除する。ここにθは素子の削除されやすさを決める正の定数である。ただし、この削除を行ったことによるQCNetの出力の変化を小さくするため、素子bの周辺の素子aの結合加重v_aを次のように修正する。

$$v_a \rightarrow v_a + (v_b - v_a)B_b^{n_m}(\mu_a) \qquad (21)$$

なお、素子が1個追加されるごとに上記の削除条件を満たす素子を探し、条件を満足する素子のうち、最初に見つかった1個だけを削除する。

以上の削除アルゴリズムAは、ある素子を削除した場合のその素子の反応中心における出力信号の変化だけに基づいて削除するかどうかを決定するため、必ずしも適切な削除が行われるとは限らない。この問題点については5節において検討する。

なお、3.1節で後に述べる方法によって、if-then規則の形式の情報に基づくQCNetの初期構造設定をした場合を考えると、本節で述べた冗長な素子の削除は、埋め込んだ規則や追加した規則のうち冗長なものを削除することに相当する。

3 QCNetへの規則の埋め込みと精密化

3.1 規則の埋め込み

本節においては、学習すべき入出力関係に関してあらかじめ分かっているif-then規則の形式の情報が存在するとき、この情報を利用してQCNetの初期構造を決め、パラメータを初期設定する方法を提案する。この方法によって適切な規則を埋め込んでそこから学習を始める場合は、白紙の状態から学習を始める場合と比較して、訓練例の少い時点での推論精度が向上することが期待できる。

学習すべき入出力関係に関してあらかじめ分かっている規則がR個存在し、それらのうちk番目の規則が

「入力xが入力空間上の閉領域a_kに属するならば、出力$y(x)$は定ベクトルc_kに等しい」

の形式で表されているとする。これらの規則は完全に正しいものである必要はない。具体的な規則の例を次に2つ示す。ここでは入力xは2次元のベクトル(x_1, x_2)であり、出力$y(x)$はスカラーであるとした。

「$0.2 < x_1 < 0.5$かつ$0.4 < x_2 < 0.8$ならば、$y(x) = 1.0$」

「$x_1^2 + x_2^2 \leq 4.0$ならば、$y(x) = 0.5$」

QCNetの初期構造を決め、パラメータを初期設定するアルゴリズムは次の通りである。

埋め込みアルゴリズムA: 1. QCNetの中間層の素子数は、規則の個数Rと等しくとる。

2. それぞれの規則のif部で言及されている閉領域a_kのそれぞれについて、閉領域a_k内の点の多数の例を一様な分布で発生させ、それらの点の座標の平均値μ_{a_k}[1]と分散σ_{a_k}を求める。

3. $1 \leq k \leq R$の各kに対して、次のようにQCNetのパラメータを初期設定する。

$$\mu_k = \mu_{a_k}, \qquad (22)$$

[1]十分多数の点を発生させれば、これは閉領域a_kの重心と一致する。

$$\sigma_k^2 = \Gamma\sigma_{a_k}^2, \quad (23)$$

$$v_k = c_k \quad (24)$$

ここに Γ は規則を埋め込む場合における素子の反応幅の局所性を決める正の定数であり、Γ が小さいと、出力値が異なる隣接する素子間における QCNet の出力の変化が急峻になる。

なお、以上の説明では、k 番目の規則の then 部が「出力は定ベクトル c_k に等しい」の形式で書かれているとしたが、「出力は入力 x の関数 $f_k(x)$ に等しい」の形式で表現されている場合には、QCNet の結合加重 v_k を入力 x の関数とすることにより、目的関数の良好な近似を実現できる。

また、以上の説明では、k 番目の規則の if 部が「入力 x が入力空間上の閉領域 a_k に属するならば」の形式で表されているとしたが、入力ベクトル x の各要素 x_i が 0,1 の 2 値をとり、規則の条件部の記述が $x_i = 0$ や $x_i = 1$ を含む論理式であるような場合においても、次のような手順でこれを QCNet に埋め込むことができる[2]。

埋め込みアルゴリズム B: 1. $-0.5 \leq x_i \leq 1.5$ の連続的な入力空間を想定する。

2. 条件部の論理式が $x_i = 0$ を含むならば、条件 $-0.5 \leq x_i \leq 0.5$ を含む閉領域 a_k を作り、$x_i = 1$ を含むならば条件 $0.5 \leq x_i \leq 1.5$ を含む閉領域 a_k を作ることによって、「$x_i = 0$ や $x_i = 1$ を含む論理式を満たすならば」という形式を「入力 x が入力空間上の閉領域 a_k に属するならば」という形式に変換する。

3. この変換を施した後に、上述の埋め込みアルゴリズム A に従って埋め込みを行う。

3.2 精密化

以上の初期設定終了後、訓練例に基づき、すでに述べた学習則に従って学習する。この場合、確率的降下法による反応中心 μ、反応幅 σ、および結合加重 v の学習は、埋め込んだ規則の if 部の条件と then 部の出力値を出力誤差が小さくなる方向に微調整することに相当する。また、新しい素子の追加は、あらかじめ分かっていた規則の微調整では説明できない例を扱うための新たな規則の追加に相当する。また、冗長な素子の削除は、埋め込んだ規則や追加した規則のうち冗長なものを削除することに相当する。

あらかじめ分かっていた R 個の規則が完全に正しいものではなかった場合や、これらの R 個の規則に基づく初期構造と初期パラメータの設定によってできた QCNet の入出力関係が、これらの R 個の情報と完全には一致しなかった場合においても、その後の訓練例に基づく学習によって、高速に目的関数の良好な近似を実現できる。

4 準備的な実験

本節では、図 1 に示す入出力関係を QCNet に学習させる実験を行い、規則の形式の情報を利用して QCNet の初期設定を行った場合と行わなかった場合とを比較する。

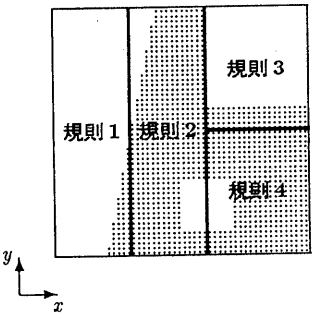

図 1: 学習させた入出力関係と、初めに与えた規則

この学習課題での入力は正方形内の点の直交座標 x と y $(0 < x, y < 1)$ であり、出力は各点における値 (0.9 または 0.1) である。図中の白い領域はその領域内の点における出力が 0.9 であることを、網掛けした領域は 0.1 であることを示す。正方形の領域内で訓練例がランダムに生成される。QCNet に初期構造として埋め込むために用意した規則は 4 つで、規則 1 から 4 と記した長方形の領域内で出力が、それぞれ、0.9(規則 1)、0.1(規則 2)、0.9(規則 3)、0.1(規則 4) であるというものである。図

[2] もちろん、x_i が 3 値以上をとる場合にも同様の手順を使うことができる。また、離散的な値をとる要素についての等式と連続的な値をとる要素についての不等式が混在していても、同様の手順によって QCNet への埋め込みが可能である。

から分かるようにそれぞれの規則はおおよそ正しいが、例外的な領域を含んでいる。

これらの各規則について、3.1節の方法でそれぞれの領域に属する多数の点の位置の平均と分散を求めると、

$$\mu_{a_1} = (0.150, 0.500), \quad \sigma_{a_1} = 0.301, \tag{25}$$

$$\mu_{a_2} = (0.450, 0.500), \quad \sigma_{a_2} = 0.301, \tag{26}$$

$$\mu_{a_3} = (0.800, 0.750), \quad \sigma_{a_3} = 0.185, \tag{27}$$

$$\mu_{a_4} = (0.800, 0.250), \quad \sigma_{a_4} = 0.185, \tag{28}$$

となる。これを用いて、式(22)(23)(24)に従ってQCNetのパラメータを初期設定した。

なお、本実験では次のような各種の定数の値を使用した。

$$\epsilon = 0.01\text{を初期値として} \quad \text{指数関数的に}e^{-2}\text{倍まで減衰}, \tag{29}$$

$$\gamma = 0.3^2, \tag{30}$$

$$\eta = \frac{1}{2}(0.3)^2, \tag{31}$$

$$\theta = \frac{1}{2}(0.05)^2, \tag{32}$$

$$\Gamma = 0.6^2 \tag{33}$$

図2は、規則を埋め込まなかった場合における、訓練例の増加に応じた正答率とモデルの大きさ(中間層の素子数)の変化を示すグラフである。図3は、規則を埋め込んだ場合における同様のグラフである。いずれの場合も、訓練例を1例与えるごとにインクリメンタルに学習を行った。グラフの横軸は学習した訓練例の数、縦軸は各図の上側の図ではテスト例に対する正答率[3]、各図の下側の図では中間層の素子数である。

訓練例を10例与えるごとにテスト例に対する正答率と素子数を計測した。テスト例としては入力空間内に等間隔で分布する1600例を使用した。この計測を、独立に生成した20通りの訓練例に対して行ない、この20回の試行で得られた正答率と素子数の平均値と分散をプロットした。グラフ中の黒点の位置が平均値を表し、縦棒の長さが分散を表す。

[3]教師信号としては数値0.9と0.1を与えており、正答率は、教師信号の0.9に対しては出力が0.6以上なら正答、0.4以下なら誤答としてカウントし、0.4から0.6の間のときは正答誤答それぞれ0.5ポイントずつをカウントして算出した。教師信号の0.1に対しても同様の基準を使用した。

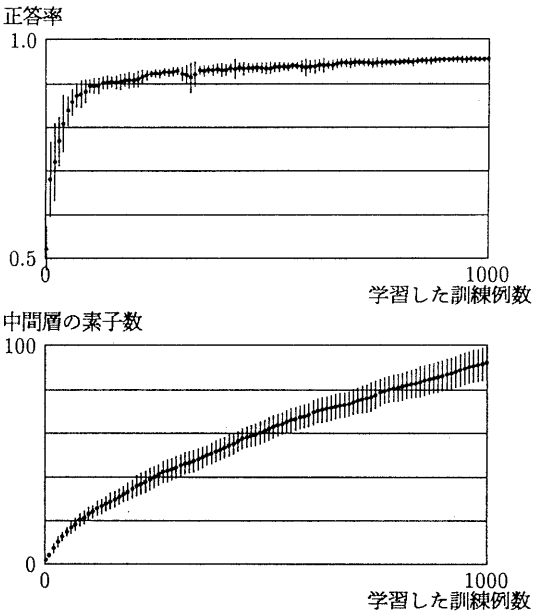

図2: QCNetにおける正答率と中間層の素子数の学習の進行による変化(規則の埋め込みを行わなかった場合)

図2と図3を比べると次のことが分かる。規則を埋め込んだ効果が明瞭に現れているのは、訓練例が少いとき(0例(規則を埋め込んだ場合には、規則を埋め込んだだけで訓練例による学習はまだ行っていない状態であり、グラフでは1番左の黒点)から40例(グラフでは左から5番目の黒点)程度)の正答率だけであって、その他は大きな違いはない。

なお、図3の正答率のグラフの1番左の黒点は、規則を埋め込んだだけの状態のQCNetの正答率を示すが、この値は0.863である。一方、埋め込む前の規則1から規則4自身を使った場合の推論の正答率は0.87であるから、本実験で使用した例題と規則の場合は、3.1節の埋め込みアルゴリズムAによって精度良く埋め込みができたことが分かる。

また、図3の正答率のグラフを見ると、10例から40例の学習を行った時点では、少しであるとはいえ、訓練例による学習の開始前よりもかえって正答率が低下していることが分かる。これは、新しく追加した中間層の素子の副作用(すなわち、追加した素子の反応中心μでは誤差が0になるが、その周辺では誤差がかえって増えるこ

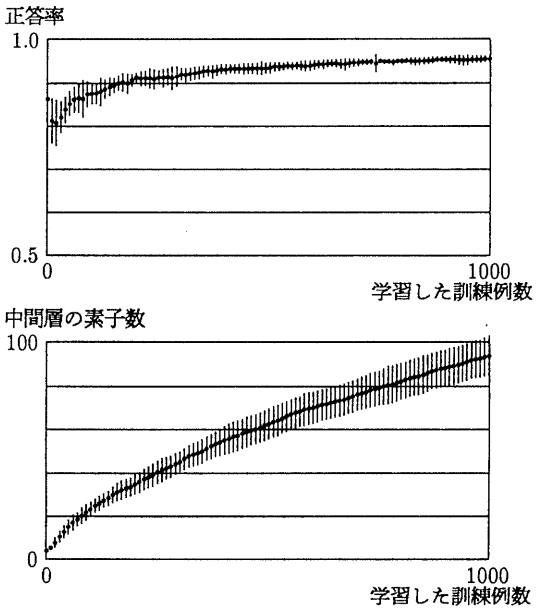

図 3: QCNet における正答率と中間層の素子数の学習の進行による変化 (規則の埋め込みを行った場合)

とがありうる) が原因である。定数 η を大きくして素子の追加を起こりにくくしたり、定数 γ を小さくして追加する素子の反応幅を小さくしたりすることで、この問題を軽減することはできるが、それぞれの定数の変更には副作用があるので注意が必要である。規則の埋め込みによって作られた素子だけを特別扱いして出力を優先したり、それらの素子のパラメータが訓練例からの学習によっては変化しにくいようにするといった対策を検討する価値があるかもしれない。

図2と図3から分かるもう1つの問題点は、正答率の向上にあまり貢献しない素子が学習の進行と共に単調に増加するということであるが、これについては5節で考察する。

5 考察

図2と図3を見ると、正答率の向上にあまり貢献しない素子が増え続けていることが分かる。これは必要な記憶容量の点から見ても、また、学習後の QCNet のパラメータを人が調べて動作を理解したり予測したりしたいという点から見ても問題である。

定数 θ の値を大きくして素子が削除されやすいようにすると、この問題は軽減されるが、正答率が落ちるという副作用がある。また、定数 η を大きくして素子の追加を起こりにくくしても、この問題は軽減されるが、学習速度が遅くなるという副作用がある。このように定数の値の変更は本質的な解決にはならない。

この問題の解決のためには、2節の削除アルゴリズム A を改良することが有効であると考えられる。削除アルゴリズム A では、「ある素子を削除した場合のその素子の反応中心における出力信号の変化」だけに基づいてその素子を削除するかどうかを決定するため、必ずしも適切な削除が行われるとは限らない。実験による確認はまだ行っていないが、訓練例がノイズを含む場合はノイズに汚染された訓練例によって作ってしまった素子を削除できないという問題が生じるはずである。また、4節の実験結果に現れた、正答率の向上にあまり貢献しない素子が増え続けるという現象は、出力の値が不連続に変化する分類課題[4]において一般的に観察されるはずである。

そこで、本論文では、これらの問題点を軽減する以下の削除アルゴリズム B を提案する。

削除アルゴリズム B:

1. 訓練例 $(\boldsymbol{x}, \boldsymbol{z}(\boldsymbol{x}))$ が与えられるごとに、中間層の各々の素子 b のその訓練例に対する貢献度

$$C_b(\boldsymbol{x}, \boldsymbol{z}(\boldsymbol{x})) = |\boldsymbol{y}'(\boldsymbol{x}, b) - \boldsymbol{z}(\boldsymbol{x})| - |\boldsymbol{y}(\boldsymbol{x}) - \boldsymbol{z}(\boldsymbol{x})| \tag{34}$$

を計算し記憶しておく。ここに、$C_b(\boldsymbol{x}, \boldsymbol{z}(\boldsymbol{x}))$ は、訓練例 $(\boldsymbol{x}, \boldsymbol{z}(\boldsymbol{x}))$ に対する中間層の素子 b の貢献度を表し、$\boldsymbol{y}'(\boldsymbol{x}, b)$ は b 番目の素子を削除して $\boldsymbol{x}$ を入力信号としたときの出力信号を表し、$\boldsymbol{z}(\boldsymbol{x})$ は入力信号 $\boldsymbol{x}$ に対応する教師信号を表し、$\boldsymbol{y}(\boldsymbol{x})$ は b 番目の素子を削除せずに $\boldsymbol{x}$ を入力信号としたときの出力信号を表す。

2. 中間層の各々の素子 b ごとに、過去の各訓練例に対する貢献度 $C_b(\boldsymbol{x}, \boldsymbol{z}(\boldsymbol{x}))$ の平均値 $\overline{C}_b$ を求める。

[4] これに対して、[5] での実験のように、出力の値が連続的に変化する関数近似課題では、削除アルゴリズム A による削除の判断基準 (入力空間上の1点だけの情報に基づいて判断する) を使ってもあまり問題は生じない。

3. 貢献度の平均値 $\overline{C}_b$ が

$$\frac{1}{2}(\overline{C}_b)^2 < \Theta \qquad (35)$$

を満たすならば、素子 b は冗長であると見なして削除する。ここに Θ は素子の削除されやすさを決める正の定数であり、その適当な大きさは中間層の素子数 n_m に依存するので、たとえば、

$$\Theta = \frac{1}{2}\frac{1}{n_m}\eta \qquad (36)$$

のように決めるとよい。

この削除アルゴリズムBでは、削除アルゴリズムAと比べて、より総合的な情報に基づいてある素子を削除するかどうかの判定を行うため、訓練例にノイズが含まれている場合や、出力の値が不連続に変化する分類課題に対しても、より適切な削除ができることが期待できる。したがって、必要な記憶容量の点から見ても、また、学習後のQCNetのパラメータを人が調べて動作を理解したり予測したりしたいという点から見ても、削除アルゴリズムBが有効であると予想できる。

6 むすび

本論文では、QCNetの構造とif-then規則の形式の情報とを自然に対応付けることができる点に注目し、学習すべき入出力関係について、あらかじめ分かっている規則の形式の情報をQCNetの初期構造として自然に埋め込む方法を示した。次に、この初期構造から出発してQCNetにおいて例からの学習を行うことにより、埋め込んだ規則のif部とthen部の修正、必要な規則の追加、冗長な規則や正しくない規則の削除のそれぞれに相当する処理を行うが、適切な初期構造を与えることによって、訓練例が少い時点での正答率を向上できることを準備的な実験により示した。さらに、分かりやすい学習結果を得、また、ノイズを含む訓練例にも対応できるように、QCNetにおける不要な素子を削除するアルゴリズムを改良する方法を提案した。

今後、改良した削除アルゴリズムの有効性(結果の分かりやすさとノイズへの対応)や、さまざまな場合における規則の埋め込みアルゴリズムの有効性(元の規則自身の推論精度と、規則を埋め込んだだけのQCNetの推論精度との比較)などを実験的に検討し、さらにこれらのアルゴリズムを改良していく予定である。また、本報告の実験の範囲では、訓練例を多数学習した後には規則を埋め込んだ効果が検証できなかったが、どのような場合に、あるいは、どのような改良によりそこでも効果が検証できるかも興味深い問題である。

将来は、実用的な問題における本方法の有効性を検証する予定である。さらに、規則の形式の情報とニューラルネットワークの形式の情報を並列的に利用する方法[10]との性能を比較する計画がある。

参考文献

[1] Amari, S., Theory of Adaptive Pattern Classifiers, *IEEE Transactions*, EC-16(3), pp. 299–307, 1967.

[2] Denoeux, T. and Lengellé, R., Initializing Back Propagation Networks With Prototypes, *Neural Networks*, Vol. 6, pp. 351–363, 1993.

[3] Fu, L. M., A Connectionist Approach to Rule Refinement, *Journal of Applied Intelligence*, Vol. 2, pp. 93–103, 1992.

[4] 金道敏樹, モデル間遷移を行う疑似競合ネットワーク, 電子情報通信学会論文誌 (掲載予定).

[5] 金道敏樹, 反学習を行う疑似競合ネットワーク, 信学技報, NC94-2, pp. 9–16, 1994.

[6] Lacher, R. C., Hruska, S. I. and Kuncicky, D. C., Back-Propagation Learning in Expert Networks, *IEEE Transactions on Neural Networks*, Vol. 3, No. 1, pp. 62–72, 1992.

[7] Mahoney, J. J. and Mooney, R. J., Combining Connectionist and Symbolic Learning to Refine Certainty Factor Rule Bases, *Connection Science*, Vol. 5, Nos. 3 & 4, pp. 339–364, 1993.

[8] 丸山美奈, 仲林清, 規則埋込み型ニューラルネットのための重み変動を抑制した学習法, 電子情報通信

学会論文誌 D-II, Vol. J75-D-II, No. 5, pp. 982–990, 1992.

[9] Moody, J. and Darken, C., Learning with Localized Receptive Fields, *Proceedings of the 1988 Connectionist Models Summer School*, Morgan Kaufmann, 1988.

[10] 岡 夏樹, 吉田邦夫, 規則モジュール、例記憶モジュール、分散表現モジュールからなる学習 / 推論システム, 人工知能学会全国大会 (第 8 回) 論文集, 1994.

[11] Opitz, D. W. and Shavlik, J. W., Heuristically Expanding Knowledge-Based Neural Networks, *Proceedings of IJCAI-93*, pp. 1360–1365, 1993.

[12] Poggio, T. and Girosi, F., Networks for Approximation and Learning, *Proceedings of the IEEE*, Vol. 78, No. 9, pp. 1481–1497, 1990.

[13] Rumelhart, D. E., Hinton, G. E. and Williams, R. J., Learning Internal Representations by Error Propagation, in D. E. Rumelhart and J. L. McClelland (eds.), *Parallel Distributed Processing*, Vol. 1, pp. 318–362, MIT Press, 1986.

[14] Shavlik, J. W. and Towell, G. G., An Approach to Combining Explanation-Based and Neural Learning Algorithms, *Connection Science*, Vol. 1, No. 3, 1989.