

抑制層を導入したファジィニューラルネットワークの提案

山本 純司† 松田 聖††

† 日本大学大学院 生産工学研究科 数理情報工学専攻

†† 日本大学 生産工学部 数理情報工学科

本研究で提案するファジィニューラルネットワーク (Fuzzy Neural Network : FNN) は 1) 初期学習過程 2) 追加学習過程の 2 つから構成される。初期学習過程では、入出力空間の特徴を考慮したクラスタリング手法を用いることによって FNN の認識能力の向上を試みた。追加学習過程では、従来の FNN に抑制層を導入することによって追加学習能力の向上を試みた。英大文字実験では従来手法に比べて初期学習、追加学習ともに良好な認識結果を得ることができた。

Fuzzy Neural Network with Inhibition Layer

Junji YAMAMOTO† Satoshi MATSUDA††

† Graduate Department of Mathematical Information Engineering, Postgraduate of Industrial Technology Nihon University

†† Department of Mathematical Information Engineering, College of Industrial Technology Nihon University

The proposed FNN has two phases. 1) Initial Learning phase 2) Additional Learning phase. In the Initial Learning phase, using input-output space considered clustering, we attempted improvement of recognition ability of FNN. In the Additional Learning phase, implementing a inhibition layer, we attempted improvement of additional learning on FNN. In the simulation of Letter Recognition, the proposed FNN obtained excellent results in both Initial Learning and Additional Learning.

1 はじめに

人間の情報処理の仕組みを模範したシステムの開発は、高度情報化社会において重要なテーマである。特にファジィシステムは人間の言葉の概念を数学的に取り扱うことが可能であり、制御やパターン認識など多くの問題に対して応用されている。しかしながらファジィシステムを構築する際、メンバシップ関数の形状などをパラメータとするファジィルールを設計者が定めなければならず、多くの時間と

労力を必要とする。

このような課題を解決するために、入力と出力の関係を近似したい数値データからファジィシステムを自動的に構築するファジィニューラルネットワーク (Fuzzy Neural Network : FNN) の研究が行われてきた [1]~[6]。FNN が学習後に獲得する結合荷重にはファジィルールが保存される。そのため従来の階層型ニューラルネットワークでは困難であったネットワークの解析や推論出力に関するルールの抽出が可能となる。しかしながら、一度構築された

FNN に対して、新たなデータを追加的に学習をさせると FNN の結合荷重は追加データのみに対して近似するように変化し、過去に獲得した知識が失われてしまう問題が発生する。

通常の FNN の学習では、入出力関係を近似するための学習データをあらかじめ用意してそれらに対する二乗誤差を最小とするようにパラメータ (メンバシップ関数の中心値や分散、後件部定数など) の最適化をはかっている。しかし現実では、あらかじめ入出力関係を近似するための全ての学習データを用意することが困難な場合が多く、一度に入手できる学習データは、学習させるべき全領域のごく一部分の領域のデータであることがほとんどである。また、FNN で一度近似した領域に対して、ごく一部の修正が必要となった場合では、その領域を修正させるための学習データを追加する必要がある。このような実環境下では、FNN を使用しつつ、新たな学習データが得られた又は発生した場合に、その学習データを FNN に追加的に学習させる必要がある。しかしながら、追加的に発生した学習データのみに対して二乗誤差を最小とするように FNN のパラメータの更新を行うと、FNN のパラメータは追加的に発生した学習データのみを近似するように更新されてしまうため過去に学習した事柄を忘却させてしまうおそれがある。すなわち新しい情報を学習すると過去の記憶が失われ、過去の記憶の保持を重視しようとするると新しい記憶が困難になるというジレンマが生じる。このようなジレンマを解決するためには、過去の学習データと追加的に発生した学習データの両方を用いて、再度 FNN に学習させる必要がある。しかしこの手法では、学習にかかる計算量も多くなり、動的に環境が変化するような状況に対処することができない。また過去の学習データが存在しているとも限らないため、現実的な手法とはいえない。そのため、追加的に発生した学習データのみを用いて、過去の学習によって得

られた記憶の破壊をなるべく抑えながら学習を行う追加学習法が提案されている [7][8]。しかしこれらの手法は、過去の学習によって得られた領域に追加データが深く入り込んでいくような場合には、追加データを適切に学習することができず結果的に過去の学習によって得られた領域を破壊し、また追加データに対しても対処することが難しい。

本研究では、過去の学習で FNN が生成した領域において、追加データの学習の妨げとなっているであろうと思われる領域を表すファジィルールを抑制し、また追加データの学習によって FNN が生成した領域が、過去に生成された領域を表すファジィルールに悪影響を及ぼすであろうと考えられる場合はそのファジィルールを強める手法を提案する。

2 抑制層を導入したファジィニューラルネットワーク

2.1 FNN の構造

図 1 に抑制層を導入した FNN の構造を示す。FNN は入力層、ルール層、抑制層、出力層の 4 つの層から構成される。入力層とルール層間、抑制層と出力層間は全結合となっている。ルール層と抑制層のノード数は共に等しく、ノード番号が同じノード同士のみが結合されている。FNN の動作を説明するために、第 α 層におけるノード i への入力を $I_i^{(\alpha)}$ 、出力を $O_i^{(\alpha)}$ を以下のように定義する。

$$\begin{aligned} I_i^{(\alpha)} &= a(O_j^{(\alpha-1)}, w_{ji}) \\ O_i^{(\alpha)} &= f(I_i^{(\alpha)}) \end{aligned} \quad (1)$$

但し、 w_{ji} は $\alpha - 1$ 層におけるノード j から α 層におけるノード i への重みである。 $a(\cdot)$ はノード j の出力と重み w_{ji} を結合する関数であり、 $f(\cdot)$ は入力 $I_i^{(\alpha)}$ の活性化関数である。

・入力層（第1層）

入力層上のノード i は、受け取った入力 x_i をそのまま出力する。

$$I_i^{(1)} = x_i \quad \text{かつ} \quad O_i^{(1)} = I_i^{(1)} \quad (2)$$

・ルール層（第2層）

ルール層上のノード j は、 j 番目のファジィルールにおける前件部の計算を行う。

$$I_j^{(2)} = a(O_i^{(1)}, w_{ij}) \quad (i = 1, \dots, N_1)$$

$$= \prod_i \exp\left(-\frac{(O_i^{(1)} - w_{ij})^2}{\sigma_{ij}^2}\right) \quad (3)$$

$$O_j^{(2)} = f(I_j^{(2)}) = I_j^{(2)} \quad (4)$$

・抑制層（第3層）

抑制層上のノード k は、追加学習時にルール層上のノード j の出力を抑制するために使用される。追加学習前の抑制層上のノード k は、接続されている第2層のノード $j (= k)$ の出力をそのまま出力する。

$$I_k^{(3)} = a(O_j^{(2)}) = O_j^{(2)} \quad (j = k) \quad (5)$$

$$O_k^{(3)} = f(I_k^{(3)}) = I_k^{(3)} \quad (6)$$

但し、ルール層上のノード j から抑制層上のノード k への重みは1である。

・出力層（第4層）

出力層上のノード o は、ファジィルールの後件部定数に相当する重み v_{ko} を用いて最終的な推論出力を得る。

$$I_o^{(4)} = a(O_k^{(3)}, v_{ko}) \quad (k = 1, \dots, N_2)$$

$$= \frac{\sum_k v_{ko} O_k^{(3)}}{\sum_k O_k^{(3)}} \quad (7)$$

$$O_o^{(4)} = f(I_o^{(4)}) = I_o^{(4)} \quad (8)$$

FNNの重み上に生成された j 番目のファジィルールは以下のように示される。

*If x_1 is \tilde{w}_{1j} , ... x_i is \tilde{w}_{ij} , ... x_{N_1} is \tilde{w}_{N_1j}
Then y_o is v_{ko} ($k = j$)*

但し、 \tilde{w}_{ij} は、 σ_{ij} の大きさによる w_{ij} の近傍である。また σ_{ij} の初期値を σ_{init} とする。

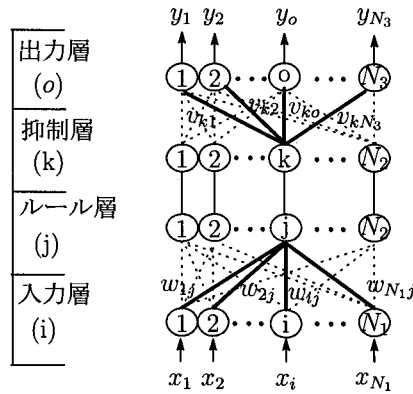


図1：抑制層を導入したFNNの構造

2.2 FNNの学習

入出力関係を表す学習データを近似するFNNの構築は 1) 初期学習過程 2) 追加学習過程の2つからなる。

2.2.1 全学習過程

初期学習過程では、学習データのクラスタリングを行うことによってFNNのルール層および抑制層上のノード数とそれらに接続されている重み w_{ij} , v_{ko} を仮決定する。その後、最急降下法を用いてFNNのパラメータ w_{ij} , σ_{ij} , v_{ko} を調整する。

A. 入出力データのクラスタリング

ファジィ推論の特徴は、入出力関係が同じような位置に分布しているデータの集合を1つのファジィルールで近似している。そのため、近似したい入出力関係を表す学習データにおいて、入力データと出力データを個別にクラスタリングを行う。但し、入力データのクラスタリングでは出力データのクラスタリング状況を踏まえながら入力データの分類を行うことで、適切な入出力関係を表すファジィルールを生成することが可能となる。

学習データにおいて、入出力学習データ \mathbf{x}^p , 出力学習データ \mathbf{y}^p を以下のように記す。

$$\mathbf{x}^p = (x_1^p, x_2^p, \dots, x_{N_1}^p) \quad (9)$$

$$\mathbf{y}^p = (y_1^p, y_2^p, \dots, y_{N_3}^p)$$

但し、 p は学習データ数を表しており、 $p = 1, \dots, L$ とする。また、FNN のルール層上のノード j へ結合されている重みベクトルを \mathbf{w}_j , ノード j と接続されている抑制層上のノード k の重みベクトルを $\mathbf{v}_k (k = j)$ として以下のように記す。

$$\mathbf{w}_j = (w_{1j}, w_{2j}, \dots, w_{N_{1j}}) \quad (10)$$

$$\mathbf{v}_k = (v_{k1}, v_{k2}, \dots, v_{kN_3}) \quad (k = j)$$

入出力学習データ ($\mathbf{x}^p, \mathbf{y}^p$) を用いて FNN のルール層および抑制層のノードを生成する手順を以下に示す。

● 初期設定

FNN のルール層および抑制層上のノード数を $N_2 = 0$ とする。また学習データ数を $p = 0$ とする。

● Step1 : 学習データの提示

$p = p + 1$ として、入力学習データ \mathbf{x}^p と、その対となる出力学習データ \mathbf{y}^p を提示する。 $p = 1$ ならば Step2 へ。そうでなければ Step3 へ。

● Step2 : ノードの初期生成

ルール層および抑制層上のノードを1つ生成する。このとき最初に提示された入出力学習データ $\mathbf{x}^1, \mathbf{y}^1$ を $\mathbf{w}_1, \mathbf{v}_1$ とする。また、 \mathbf{w}_1 を入力空間上の1つのクラスタの代表点みなして、 \mathbf{w}_1 を中心とする半径を $R_1^w = 0$ とする。同様に \mathbf{v}_1 を出力空間上の1つのクラスタの代表点とみなして、 \mathbf{v}_1 を中心とする半径を $R_1^v = 0$ とする。

$$\mathbf{w}_1 \leftarrow \mathbf{x}^1 \quad (11)$$

$$\mathbf{v}_1 \leftarrow \mathbf{y}^1 \quad (12)$$

$N_2 = N_2 + 1$ として Step1 へ。

● Step3 : 出力学習データの判定

提示された出力学習データ \mathbf{y}^p と抑制層上のノード k の重み \mathbf{v}_k とのユークリッド距離を

$$d_k^v = \|\mathbf{y}^p - \mathbf{v}_k\| \quad (13)$$

とすると、 \mathbf{v}_k を中心とする半径 R_k^v の円と \mathbf{y}^p を包含する円の半径は $R_k^{v'} = (d_k^v + R_k^v)/2$ となる (図2)。抑制層上の全てのノード $k (k = 1, \dots, N_2)$ において、 $R_k^{v'} \leq \zeta_v$, または $d_k^v \leq R_k^v$ となるノード $k^* (k^* \in N)$ を検出する。但し、 N は検出されたノード番号の集合である。このような k^* が存在すれば Step4 へ。存在しなければ Step6 へ。

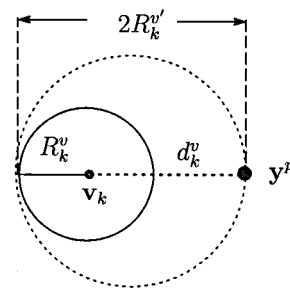


図2: 包含する円の生成

● Step4 : 入力学習データの判定

Step3 で検出された抑制層上の全てのノード

ド k^* において、ノード k^* と結合されているルール層上のノード $j^*(=k^*)$ の重み \mathbf{w}_{j^*} と提示された入力データ \mathbf{x}^p とのユークリッド距離を最小にするルール層上のノード min を検出する。

$$min = \arg \min_{j^*} (\|\mathbf{x}^p - \mathbf{w}_{j^*}\|) \quad (14)$$

$$d_{min}^w = \|\mathbf{x}^p - \mathbf{w}_{min}\| \quad (15)$$

\mathbf{w}_{min} を中心とする半径 R_{min}^w の円と \mathbf{x}^p を包含する円の半径は $R_{min}^{w'} = (d_{min}^w + R_{min}^w)/2$ となる。 $R_{min}^{w'} \leq \zeta_w$, または $d_{min}^w \leq R_{min}^w$ ならば Step5 へ。そうでないなら Step6 へ。

• Step5 : 重みの更新

Step4 で検出されたルール層上のノード min と結合されている抑制層上のノードの重みを \mathbf{v}_{min} とする。 $d_{min}^v > R_{min}^v$ ならば、 \mathbf{v}_{min} を中心とする半径 R_{min}^v の円と \mathbf{y}^p を包含する円の中心に \mathbf{v}_{min} を移動する。また R_{min}^v の値を包含した円の半径に更新する。

$$\mathbf{v}'_{min} \leftarrow \frac{\mathbf{v}_{min}(d_{min}^v + R_{min}^v) + \mathbf{y}^p(d_{min}^v - R_{min}^v)}{2d_{min}^v} \quad (16)$$

$$R_{min}^{v'} \leftarrow \frac{d_{min}^v + R_{min}^v}{2} \quad (17)$$

ルール層上のノード min の重みの更新も同様に、 $d_{min}^w > R_{min}^w$ ならば \mathbf{w}_{min} , R_{min}^w を次式で更新する。

$$\mathbf{w}'_{min} \leftarrow \frac{\mathbf{w}_{min}(d_{min}^w + R_{min}^w) + \mathbf{x}^p(d_{min}^w - R_{min}^w)}{2d_{min}^w} \quad (18)$$

$$R_{min}^{w'} \leftarrow \frac{d_{min}^w + R_{min}^w}{2} \quad (19)$$

Step7 へ。

• Step6 : ノードの追加

\mathbf{x}^p , \mathbf{y}^p をそれぞれ \mathbf{w}_{N_2+1} , \mathbf{v}_{N_2+1} としてルール層および抑制層上のノードを追加する。また \mathbf{w}_{N_2+1} の半径を $R_{N_2+1}^w = 0$, \mathbf{v}_{N_2+1} の半径を $R_{N_2+1}^v = 0$ とする。

$$\mathbf{w}_{N_2+1} \leftarrow \mathbf{x}^p \quad (20)$$

$$\mathbf{v}_{N_2+1} \leftarrow \mathbf{y}^p \quad (21)$$

$N_2 = N_2 + 1$ として Step7 へ。

• Step7 : 終了判定

$p = L$ ならば終了する。そうでなければ Step1 へ。

B. 最急降下法によるパラメータ調整

FNN のパラメータ w_{ij} , σ_{ij} , v_{ko} を最急降下法を用いて調整する。FNN の出力教師信号を T_o , 二乗誤差関数 E を

$$E = \frac{1}{2} \sum_o (T_o - O_o^{(4)})^2 \quad (22)$$

として、各パラメータの値を次式で更新する。

$$w_{ij}(t+1) = w_{ij}(t) + \alpha \left(-\frac{\partial E}{\partial w_{ij}} \right) \quad (23)$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) + \beta \left(-\frac{\partial E}{\partial \sigma_{ij}} \right) \quad (24)$$

$$v_{ko}(t+1) = v_{ko}(t) + \gamma \left(-\frac{\partial E}{\partial v_{ko}} \right) \quad (25)$$

なお α, β, γ は学習定数である。

2.2.2 追加学習過程

追加学習は、一度構築した FNN に新たな学習データがあった場合に行う。既存の FNN に新たな学習データを提示したとき、ルール層の全てのノードの出力値が θ 以下だった場合や、誤認識をした場合には、適切なファジィルールを持っていない可能性が高い。そのため、ルール層上の全てのノードの出力が θ 以下または既存の FNN が誤認識を起こした追加データに対して 2.2.1 の A で示したクラスタリングを行い、新たなファジィルールとして既存の FNN のルール層および抑制層上のノードを追加する (図 3)。このことによって誤認識を起こした追加データの近傍にファジィルールが生成されるため FNN の近似能力を高めることができる。

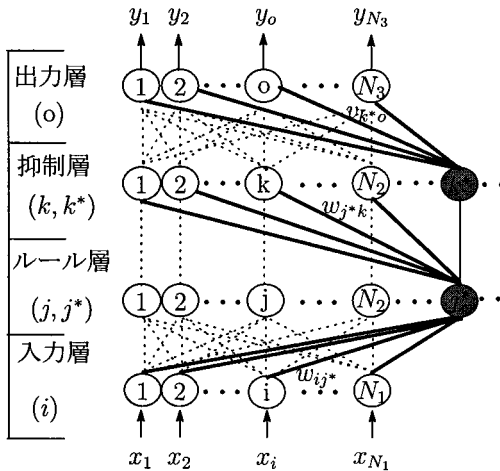


図 3 : 追加学習過程の FNN の構造

新たに追加されたルール層および抑制層上のノードの入出力における動作は 2.1 で示したものと同一である。また、新たに生成されたルール層上のノードを j^* ($j^* = N_2 + 1, \dots, N_2 + N_2^*$)、抑制層上のノードを

k^* ($k^* = N_2 + 1, \dots, N_2 + N_2^*$) として既存の抑制層上のノードの入出力を変更する。

[既存の抑制層の動作変更]

追加データに対して既存の FNN が誤認識を起こした場合、新たなファジィルールを追加することで FNN の近似精度を高めた。しかしながら、FNN が誤認識を起こした追加データに対して既存のルール層上のノード j の出力が高い場合、その出力がそのまま既存の抑制層上のノード k に伝わるため、誤認識を起こす原因となっている既存の抑制層-出力層間の重みが強調される。そのため追加データに対して新たなルールを追加しても正しい認識をすることができない。このことを回避するために新たに生成されたルール層上のノード j^* の出力が高い場合、既存のルール層上のノード j の出力を抑制することによって既存の抑制層-出力層間の重みの影響を抑える。追加されたルール層上のノード j^* の出力値を $O_{j^*}^{(2)}$ 、既存のルール層上のノード j の出力値を $O_j^{(2)}$ として、既存の抑制層上のノード k の入出力を変更する。

まず、ノード k への入力を次式で与える。

$$I_k^{(3)} = a(O_j^{(2)}, O_{j^*}^{(2)}, w_{j^*k}) \\ = \left(\prod_{j^*=N_2+1}^{N_2+N_2^*} S((1 - O_{j^*}^{(2)})w_{j^*k}) \right) O_j^{(2)} \quad (26)$$

ここで、 w_{j^*k} はノード j^* からノード k への重みであり、1 以上の値をとる。また S はシグモイド関数であり、以下のように与えられる。

$$S(u) = \frac{1}{1 + e^{-\frac{2(u-1)}{T}}} \quad (27)$$

$S(u)$ を、 u が 1 以上のときは 1 を、それ以外の場合は 0 をとるステップ関数に近づけるため、 $t = 0.95$, $T = 0.01$ とした (図 4)。

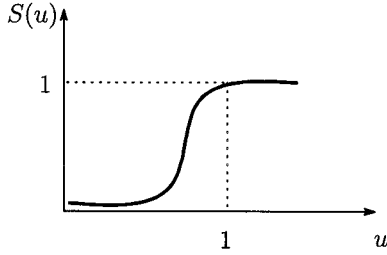


図4：シグモイド関数

式(26)では、新たに追加されたルール層上のノード j^* の出力 $O_{j^*}^{(2)}$ が大きいほど、既存のルール層上のノード j の出力 $O_j^{(2)}$ が抑制される。また、 w_{j^*k} はノード j^* の出力が一定値以下であればノード j の抑制を行わせない働きをする。この理由として、ノード j^* の出力値が低い場合は、認識したいデータが既存のファジィルールに属する可能性が高いからである。例として式(26)でノード j^* の出力 $O_{j^*}^{(2)}$ が0.5以下のときはノード j の出力 $O_j^{(2)}$ を抑制させないための w_{j^*k} の設定例を示す。式(27)において、 $u \geq 1$ のとき $S(u) \doteq 1$ であることを留意して

$$u = (1 - O_{j^*}^{(2)})w_{j^*k} = 1 \quad (28)$$

となるような w_{j^*k} をもとめる。このとき $O_{j^*}^{(2)}$ が0.5以下の場合において、 $w_{j^*k} = 2$ としたときは $S(u) \doteq 1 (u \geq 1)$ となるためノード j の出力 $O_j^{(2)}$ を抑制しない。従って、新たに追加したルール層上の全てのノードの出力が上記の要領で定めた値以下である場合は

$$\prod_{j^*=N_2+1}^{N_2+N_2^*} S\left((1 - O_{j^*}^{(2)})w_{j^*k}\right) \doteq 1 \quad (29)$$

となり、式(26)は既存のルール層上のノード j の出力 $O_j^{(2)}$ を抑制しない。

次にノード k の出力を次のように定める。

$$\begin{aligned} O_k^{(3)} &= f(I_k^{(3)}) \\ &= C \times I_k^{(3)} \end{aligned} \quad (30)$$

但し、 C は1以上の定数である。新たに追加したルール層上のノードからの抑制がないときに限り、既存の抑制層上のノードの出力を高める働きをする。この理由は、新たに追加されたルール層および抑制層上のノードによって、過去の学習で生成されたFNNの忘却を防ぐためである。

最急降下法によるパラメータ調整

(追加学習時)

2.2.1のBと同様に最急降下法によってパラメータを調整する。追加学習時のパラメータ調整では過去に生成されたFNNのパラメータ $w_{ij}, \sigma_{ij}, v_{ko}$ は変更せず、新たに追加された層のパラメータ $w_{j^*}, \sigma_{j^*}, w_{j^*k}, v_{k^*o}$ のみを更新する。但し w_{j^*k} の下限値を w_{min} とする。

$$w_{j^*}(t+1) = w_{j^*}(t) + \alpha \left(-\frac{\partial E}{\partial w_{j^*}}\right) \quad (31)$$

$$\sigma_{j^*}(t+1) = \sigma_{j^*}(t) + \beta \left(-\frac{\partial E}{\partial \sigma_{j^*}}\right) \quad (32)$$

$$v_{k^*o}(t+1) = v_{k^*o}(t) + \gamma \left(-\frac{\partial E}{\partial v_{k^*o}}\right) \quad (33)$$

$$w_{j^*k}(t+1) = w_{j^*k}(t) + \eta \left(-\frac{\partial E}{\partial w_{j^*k}}\right) \quad (34)$$

(但し $w_{j^*k} \geq w_{min}$)

3 計算機実験

提案したFNNを評価するために[9]で公開している英大文字の認識問題(Letter Recognition Database)を例題に評価実験を行った。

3.1 英大文字認識問題

Letter Recognition Databaseが提供するデータは、英大文字から抽出された16の特徴量を用いて、それをAからZに分類する

問題である。本実験では 初期学習過程で構築した FNN を評価するために、20000 データ中 16000 データを学習データとして使用し、残り 4000 データを評価に使用した。

また追加学習法を評価するために、A から J までの 10 種 6114 データを一度学習し、残り 16 種 9886 データを新たに追加学習させることで行った。表 1 に FNN の実験パラメータを示す。

表 1:実験パラメータ

ζ_v	0.0
ζ_w	5.5
σ_{init}	1.6
θ	0.2
$\alpha, \beta, \eta, \gamma$	0.0001
w_{min}	2.0
C	2.5

3.2 実験結果

表 2 に学習用データすべてを初期学習過程の学習にあてた全学習におけるテストデータの認識結果と、初期学習過程で A から J までの 10 種類のデータのみを利用して FNN を学習させて、K から Z までのデータを追加学習過程で学習させた場合のテストデータの認識結果を示した。

表 2:実験結果

手法	全学習	追加学習
提案手法	94.7 %	87.8 %
比較手法 [7]	91.2 %	80.6 %

4 おわりに

ファジィ推論の特徴は入出力関係が同じように分布しているデータを 1 つのファジィルールで近似していることである。そのため初期学習過程では入出力空間を考慮したクラスタリング手法を用いることによって FNN が極小値に陥いる可能性が低くなったと考えることができる。また追加学習過程では良好な認識結果を得ることができ、事前にデータが完全な形で用意できない場合や、時間とともに変化していくデータの分類問題に対して有効であ

る。しかしながら、新たに追加したルール層のノードから既存の抑制層へのノードにおける重みの下限値の設定は場当たり的であり、検討する必要がある。

参考文献

- [1] J.R.Jang, "ANFIS : Adaptive-Network-Based Fuzzy Inference System", IEEE Trans. Systems, Man Cybern. vol.23, pp665-685, May 1993
- [2] C.F.Jang, C.T.Lin, "An On-Line Self-Constructing Neural Fuzzy Inference Network and Its Applications", IEEE Trans. Fuzzy System. Vol.6, pp12-32, 1998.
- [3] J.S.Wang, C.S.George Lee, "Self-Adaptive Neuro-Fuzzy Inference System for Classification Applications", IEEE Trans. Fuzzy System. Vol.10, 2002.
- [4] M.Y.Chen, D.A.Linkens, "A Systematic Neuro-Fuzzy Modeling Framework With Application to Material Property Prediction", IEEE Trans. System , Man Cybern, Vol.31, pp781-790, 2001.
- [5] S.G.Tzafestas, K.C.Zikidis, "NeuroFAST : On-Line Neuro-Fuzzy ART-Based Structure and Parameter Learning TSK Mode", IEEE Trans. System , Man Cybern, Vol.31, pp797-802, 2001.
- [6] H.Iyatomi, M.Hagiwara, "Adaptive Fuzzy Inference Neural Network", pattern recognition, Vol.37, pp2049-2057, 2004
- [7] 居谷 道明, 彌富 仁, 萩原 将文, "追加学習可能なファジィ推論ニューラルネットワークの提案と画像認識への応用", 日本知能情報ファジィ学会, Vol.17, No.1, pp.60-67, 2005
- [8] 山内 康一郎, 山口 暢彦, 石井 直宏, "神経回路による干渉を受けるパターンの想起と追加学習", 電子情報通信学会論文誌, D- , Vol.J80, No.1, pp295-305, 1997.
- [9] UCI MachineLearningRepository, <http://www.ics.uci.edu/mllearn/databases/>