マイクロコンピュータとワークステーション 66－5
（1991. 5. 10）

ファジィ推論におけるメンバシップ関数の高速学習方式について

前田 章　染谷 隆子　船橋 誠壽
（株）日立製作所システム開発研究所

従来のファジィ応用システムの開発では、メンバシップ関数を試行錯誤によって決定しており、多大な工数を必要としていた。本報告では、この作業の自動化を目的とし、教師データを用いた学習によりメンバシップ関数の形状を自動的にチューニングする方式について述べる。本方式では、ファジィ推論過程を計算ネットワークで表現し、ネットワーク算法を用いた高速偏微分係数計算法を利用する。これによって、ニューラルネットワークでよく知られているバックプロパゲーション法と同様の学習アルゴリズムが適用でき、高速な学習が可能となる。実験により直接非線形最適化法のおよそ４０倍の高速化を達成できることを確認した。

A Fast Learning Algorithm for Fuzzy Membership Functions

Akira Maeda Ryuko Someya Motohisa Funabashi
Systems Development Laboratory, Hitachi, Ltd.
1099 Ohzenji, Asao-ku, Kawasaki 215, JAPAN

One of the most time-consuming procedures in developing fuzzy applied systems is to determine the shapes of membership functions. In this paper, a fast supervised learning algorithm for fuzzy membership functions is described. In order to achieve high throughput, the fuzzy inference procedure is represented by a computational flow network and output error is backpropagated through the network to efficiently calculate partial derivative coefficients. In our simulation experiment, the proposed algorithms runs about 40 times faster than a direct nonlinear optimization algorithm.

1．はじめに

ファジィ推論を実用システムに適用する試みが急増している。対象とする分野も、地下鉄の自動運転やトンネル換気制御を代表とするシステム制御から、資産運用計画などの金融・証券分野、さらには家電製品へと、多岐にわたっている。

ファジィ推論の適用が進められている理由には、

（１）エキスパートの言語的な知識を表現できる（高い、小さい、やや低い、など）

（２）少ないルールでモデルを表現できるため、高速性を要求される実時間制御に向く

（３）数値（アナログ）情報を扱いやすい

などがある。これらの特徴は、記号処理を中心とした従来の知識工学的手法の欠点を補完するものであり、ファジィ推論の応用により柔軟なシステムが構築できるというメリットがある。

その反面、ファジィ推論システムの問題点の一つとして、メンバシップ関数の決定に明確な指針が与えられないことがある。本来メンバシップ関数は、主観に基づく曖昧さを表現するものとされているから、関数形状が一意的に決まるということは一般的にはない。しかしながら、ファジィ推論を実システムに適用する場合は、何らかの基準にしたがってメンバシップ関数を決める必要がある。たとえば制御システムへの適用の場合には、制御性能が要求を満たすように決めるというのが一般的な条件である。これに対して従来の方法では、メンバシップ関数は試行錯誤により決定していたのが実状であった。

本報告では、この問題点を解決するために、ファジィ推論の入力値とそれに対する望ましい出力が与えられたときに、メンバシップ関数を学習により自動的に最適化する方式について述べる。この方式では、ファジィ推論過程を計算ネットワークで表現し、ネットワーク算法を用いた高速偏微分係数計算法を利用する。これによって、ニューラルネットワークでよく知られているバックプロパゲーション法と同様の学習アルゴリズムが適用でき、高速な学習が可能となる。

2．問題の定式化

ファジィ推論の定義にはさまざまなバリエーションがあるが、ここではMamdaniのファジィ推論と呼ばれている方法[1)]を例にとって説明する。

図１にファジィ推論過程を図示する。この例は入力変数が２個、ルールが２個の場合で、ルールは

if x_1 is small and x_2 is medium then y is medium
if x_1 is large and x_2 is small then y is large （１）

を仮定した。式（１）に現れるメンバシップ関数は図１の左上の４つのグラフで表現されている。図１の右側の２つのグラフは、if部の出力変数のメンバシップ関数で、斜線部分はそれにif部の適合度を乗算した関数である。右下のグラフは合成されたメンバシップ関数である。斜線部分の重心をとり、そのy座標が出力値Yとなる。

さて、図１のファジィ推論は、結局のところ入力$X=(X_1, X_2, \cdots, X_n)$と出力Yの非線形な関

係式

$$Y = f(X; \alpha) \tag{2}$$

を規定しているのにほかならない。ここでαは関数fを決定するパラメータであり、ファジィ推論の場合は入出力変数に対するメンバシップ関数の形状がこのパラメータに相当する。メンバシップ関数として任意の形状を考える場合もあるが、一般的にはいくつかの関数クラスに限定して自由度を減らすことが多い。ここではメンバシップ関数形状を定義するパラメータを学習の対象と考える。

さて、ファジィ推論における学習とは、入力Xとそれに対する望ましい出力Y_tの組(X, Y_t)（以下、この組のことを教師データと呼ぶ）が与えられたときに、ファジィ推論の出力Yを望ましい出力Y_tにできるだけ近づけるようなパラメータαを求める手続きを指す。（YとY_tとの距離は、２乗誤差$(Y-Y_t)^2$の意味で考えることが多い。）したがって、この問題は非線形最適化問題と等価であり、最急降下法やコンプレックス法など、一般的な最適化手法を用いてメンバシップ関数の学習ができることがわかる。この時の問題点は学習に要する計算量である。逐次的に最適化を進める方法では、繰り返し毎にコスト関数（今の場合は$(Y-Y_t)^2$）の各パラメータによる微分（差分）を求める必要があり、計算量はルール数または学習すべきパラメータ数の２乗に比例して増加する。次章では計算ネットワーク表現による高速偏微分計算法を用いて、ルール数に比例した計算量で学習を行うアルゴリズムを提案する。

３．拡張バックプロパゲーション法による学習アルゴリズムの提案

3.1　計算ネットワーク表現による高速偏微分係数計算法

本節では、提案アルゴリズムのもとになる合成関数の高速偏微分係数計算法[2)]について説明する。

n変数の合成関数 $y = f(x_1, x_2, \cdots, x_n)$ の、各変数x_iごとの偏微分係数$\partial y/\partial x_i$を計算する問題を考えよう。

もっとも直接的な方法は、各変数x_iごとに直接微分（差分）を計算する方法である。この方法では、変数の数が増えるとその２乗に比例して計算時間がかかる。

それに対して高速偏微分係数計算法では、合成関数の計算手順を計算ネットワークで表現し、偏微分係数を出力変数側から入力変数側にたどることにより、すべての変数に対する偏微分係数を同時に求める。この方法では、偏微分係数の計算時間は、関数自体の計算時間の高々定数倍以内であることが示されている。

例をあげて説明しよう。ここでは３変数の合成関数

$$y = f(x_1) \cdot x_2 / g(x_2 + x_3) \tag{3}$$

の、各変数$x_i (i=1,2,3)$に関する偏微分係数$\partial y/\partial x_i$を求めることを考える。

図２は式（３）を計算ネットワークで表現したものである。各ノードは単項または２項演算を表す。図２では、関数f、gを表す単項演算ノード、乗算・加算・除算を表す２項演

算ノードが各１つづつ、合計５個のノードが現れている。またノードとノードを結ぶリンクは変数の入出力関係を表現している。当然であるが、このネットワークは有向グラフであり、ループを含まないことに注意しておく。

偏微分係数を計算する手順は次のようになる。

まず各ノードの入出力関係に関する偏微分係数（要素的偏微分係数）をすべて求めておく。図２の除算ノード $y_5 = y_3/y_4$ に関していえば、$\partial y_5/\partial y_3 = 1/y_4$, $\partial y_5/\partial y_4 = -y_3/y_4^2$ となる。

次に各中間出力y_iに関する偏微分係数$\partial y/\partial y_i$を求める。これは計算ネットワークを出力側から入力変数側にたどったすべての経路に関して、要素的偏微分係数の積和を求めればよい。これが正しいことは、合成関数の微分規則から明かである。

これをすべてのノードについて行えば、入力変数x_iに関する偏微分係数が求められることがわかる。例えば図２の例で$\partial y/\partial x_2$を求める場合、x_2から$y(=y_5)$にいたる経路が２つあるので、それぞれに対する要素的偏微分係数の積$\partial y_3/\partial x_2 \cdot \partial y/\partial y_3$と$\partial y_2/\partial x_2 \cdot \partial y_4/\partial y_2 \cdot \partial y/\partial y_4$を求めることにより、$\partial y/\partial x_2 = y_1/y_4 - g' \cdot y_3/y_4^2 = f/g - f \cdot x_2 \cdot g'/g^2$ が求められる。

ここで重要なことは、上記の方法によれば、入力変数だけではなく、すべての中間出力に関する偏微分係数が求められることである。したがって、計算ネットワークのノードで表現される演算にパラメータが含まれる場合、そのパラメータに関する偏微分係数も容易に計算できる。図２の例で、関数fがパラメータαを含み、$y_1 = f(x_1, \alpha)$の場合、$\partial y/\partial \alpha = \partial y/\partial y_1 \cdot \partial y_1/\partial \alpha = \partial y/\partial y_1 \cdot \partial f/\partial \alpha$から$\alpha$に関する偏微分係数を求めることができる。

3.2　拡張バックプロパゲーション法

前節で述べた高速偏微分係数計算法をファジィ推論パラメータの学習に応用することを考えよう。ここで提案するアルゴリズムを拡張バックプロパゲーション法と呼ぶことにする。

まず２．で述べたファジィ推論方法を計算ネットワークにより表現する。図３は、図１の例のファジィルールによる推論を計算ネットワークで表現したものである。

図３の左端のノードn_1〜n_3はそれぞれ入力変数、出力変数を表す。その右側に位置するノードn_4〜n_9はメンバシップ関数を表すノードで、入力／出力変数の値を入力とし、そのメンバシップ値を出力する。

さて、このようにファジィ推論アルゴリズムを計算ネットワークで表現すれば、３．１で説明した高速偏微分計算法を適用することができる。ただしファジィ推論の場合は、入出力変数ではなく、メンバシップ関数を表すノードn_4〜n_9に含まれるパラメータに関する偏微分係数が必要になる。これが本アルゴリズムにより高速に求められることは前節で述べた。したがって式（２）のパラメータαの各成分α_iによるファジィ推論の出力Yの偏微分係数$\partial Y/\partial \alpha_i$が計算できる。

図３の計算ネットワークは３．１で扱った計算ネットワークと比べると

（１）３つ以上の入力をもつノードを含む。
（２）max,minという微分不可能な点をもつ演算を含む。

という点が異なっている。しかし、前節の方法を拡張してこの場合に適用することは容易である。

上記のアルゴリズムと、ニューラルネットワークの学習アルゴリズムとして有名なバックプロパゲーション法との類似は明かである。バックプロパゲーション法では、計算ネットワークのノードがfをシグモイド関数として$f(\Sigma w_{ij} o_j)$という形に限定された特殊な場合ということができる。これが提案アルゴリズムを拡張バックプロパゲーション法と呼んだ理由である。

拡張バックプロパゲーション法が、前節で指摘した従来の学習法の問題点の１つである計算速度の遅さという問題を改善するものであることも明かである。すなわち、従来の方法では、一組の学習データに対してm個のパラメータを調節するのに要する計算量が、一回のファジィ推論に要する計算量のO(m)倍であるのに対し、提案方式ではO(1)倍で済むからである。

４．シミュレーションによる効果確認実験

本章では提案法の効果を原理確認するため行ったシミュレーションの結果を述べる。

例としてファジィ推論を応用した株価の予測[3)]をとりあげる。

ここではグランビルの法則と呼ばれる８つの株売買ルールを用いる。もともとのグランビルの法則は、株の売買タイミングに関するものであるが、ここでは株価の予測ルールとして解釈をする。すなわち、「買い」は株価の上昇を、「売り」は株価の下落を意味するものとする。そしてグランビルの法則をファジィルールで記述し、株価を予測する。ここで用いた例では、入力変数は１０個、ルールは８個、メンバシップ関数は２６個、メンバシップ関数の形状パラメータは８４個である。

図４に提案法による学習の効果を示す。１００組の教師データを用い、１００回の繰り返しにより学習した。破線がテストデータの株価変動、実線がファジィ推論により予測した株価変動である。図４（ａ）は学習前の結果、図４（ｂ）は学習後の結果である。明らかに推論精度が向上しているのが分かる。

図５に学習処理速度の比較を示す。３．で述べたように、拡張バックプロパゲーション法による学習では、前向きのファジィ推論の約２倍の処理時間しかかからず、これはルールの数に依存しない。したがって、直接８４個のパラメータで最急降下法による最適化をした場合に比べて約４０倍の高速処理を実現できる。ワークステーション２０５０／３２Ｅを用いた場合の学習時間はおよそ３分であり、大型計算機や付加ハードウェアなしで十分対話処理が可能であることを確認した。

５．おわりに

ファジィ応用システムを構築する上で従来最も困難で時間のかかっていたメンバシップ

関数の決定を、教師データを用いて学習する方式を提案した。本方式はニューラルネットワークの学習方式として広く知られているバックプロパゲーションアルゴリズムと同様、計算ネットワークによる高速偏微分係数計算法を利用して計算量を削減するもので、ここでは拡張バックプロパゲーション法と呼んだ。

グランビルの法則を用いた株価予測システムの例について学習実験を行い、直接最適化法の約４０倍の処理速度で学習できることを確認した。

参考文献

1) E.Mamdani: Advances in the Linguistic systhesis of fuzzy controller, Int.J.Man-Machine Studies, vol.8, No.6, pp.669-679(1976)

2) 伊理、ほか：偏導関数計算と丸め誤差推定の自動化の大規模非線形方程式系への応用、情報処理、vol.26、no.11、pp.1411-1420(1985)

3) 染谷、ほか：エキスパートシステム構築ツールＥＳ／ＫＥＲＮＥＬ／Ｗのファジィ推論、日本ファジィ学会誌、Vol.2、No.2、pp.125-131(1990)

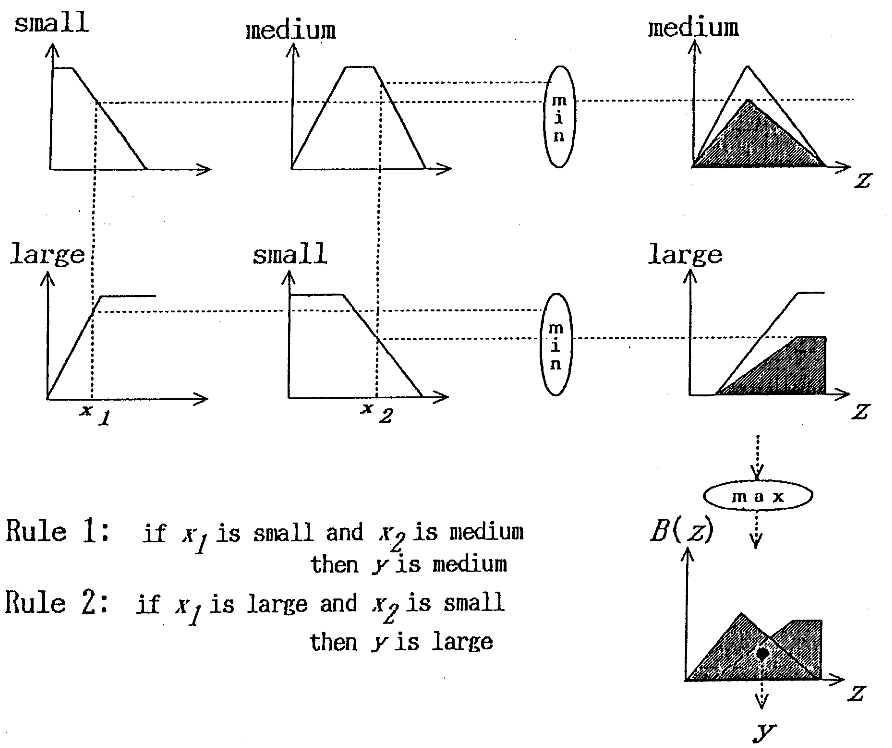
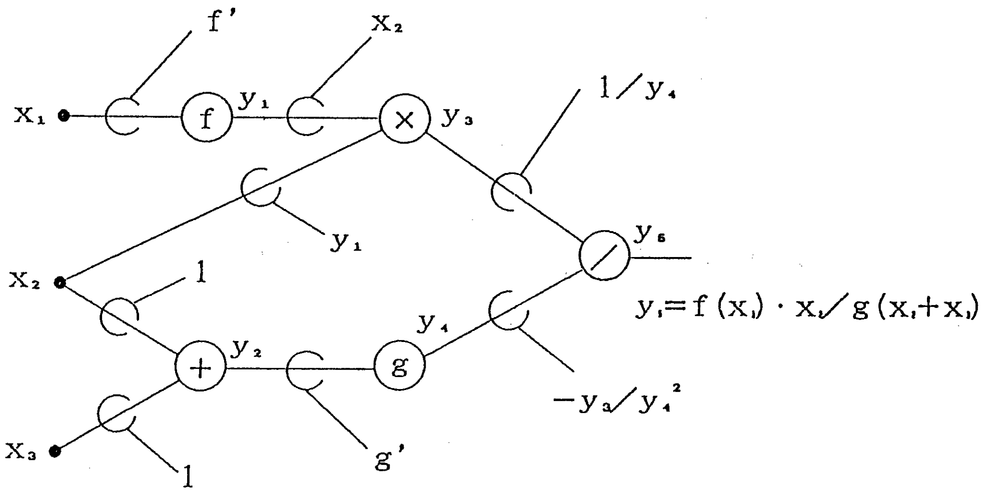


図１　Mamdaniのファジィ推論法



$$\partial y_5/\partial x_1 = f' \cdot x_2/y_4$$
$$\partial y_5/\partial x_2 = y_1/y_4 - g' \cdot y_3/y_4^2$$
$$\partial y_5/\partial x_3 = -g' \cdot y_3/y_4^2$$

図 2　合成関数の高速偏微分係数計算法

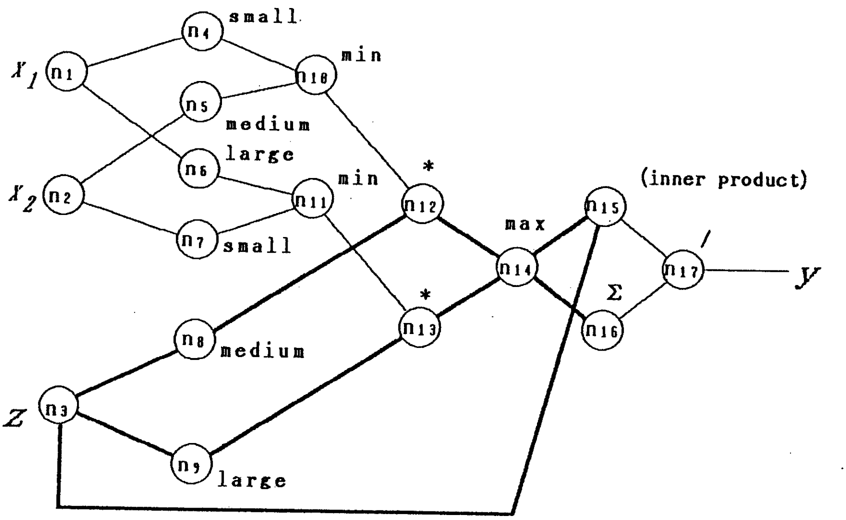
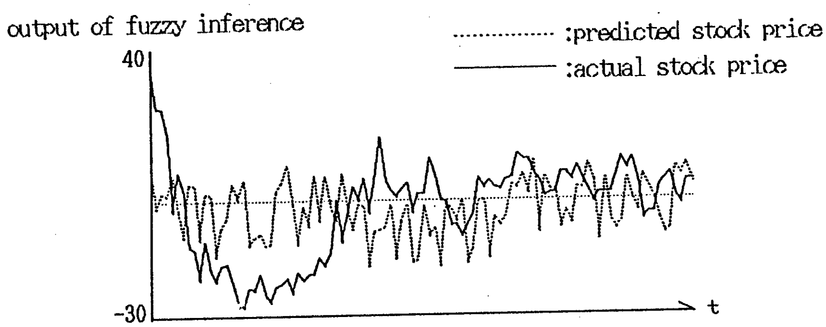
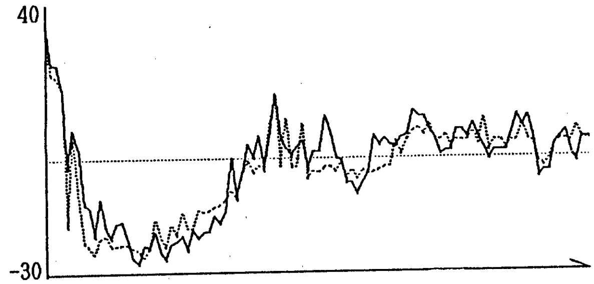


図 3　ファジィ推論の計算ネットワーク表現



(a) before self-tuning



(b) after self-tuning

図４　提案方式によるメンバシップ関数学習の効果

方　式	学習時間（秒）	処理速度（ミリ秒/ケース/繰返し）	速度比
ファジィ推論	７９	７．９	１．０
提案方式	１８１	１８．１	２．３
直接最適化法	６８２０	６８２．０	８６．３

測定条件
・使用マシン：　２０５０／３２Ｅ
・言語：　　　　Ｃ
・ファジィルール：グランビルの法則
（１０入力、１出力、８ルール、２６メンバシップ関数、８４パラメータ）
・教師データ１００ケース、繰返し１００回

図５　学習処理速度の比較