

# ファジィ遺伝的アルゴリズムによる ファジィニューラルネットの進化的学習

平成 26 年 4 月 22 日受付

岡田英彦\*

## 要旨

これまでに、結合強度の値を従来の実数からファジィ数に拡張したニューラルネットと、その学習方法が提案されている。しかし、従来の学習方法は誤差逆伝播法を拡張した方法であったため、教師データが必要であった。一方、教師データを必要とせずにニューラルネットを学習させる方法として、進化的アルゴリズムを用いた強化学習方法が研究されており、ニューロエボリューションと呼ばれている。しかし、従来のニューロエボリューションにおいては、学習対象のニューラルネットは結合強度の値が実数である古典的なニューラルネットであり、前記のファジィニューラルネットの学習には適用されていなかった。これに対して本研究ではこれまでに、genotype の値としてファジィ数を利用できるように進化的アルゴリズムを拡張する手法を提案した。この提案手法を用いれば、ファジィニューラルネットの進化的学習が可能になると考えられる。本稿では、ファジィニューラルネットの進化的学習に対する提案手法の有効性評価を行った。実験では、結合強度の値であるファジィ数を対称三角型とし、進化的アルゴリズムとしては遺伝的アルゴリズムを用いた。また、ファジィニューラルネットの学習目標は、実数の入力に対してファジィ数を出力する非線形な関数を近似することとした。実験の結果、教師データが与えられず学習目標の関数も明示的には与えられていないにも関わらず、提案手法によってファジィニューラルネットを自律的に進化させ、目標関数を近似させることができた。また、対称三角型ファジィ数を規定する方法として上限・下限モデルと中心・幅モデルを比較した結果、中心・幅モデルを用いたほうがより適合度の高いファジィニューラルネットが生成され、中心・幅モデルの利用が望ましいことがわかった。

キーワード：進化的アルゴリズム、遺伝的アルゴリズム、ニューラルネット、ファジィ数、ニューロエボリューション

---

\* 京都産業大学コンピュータ理工学部

## 目次

1. 序論
  2. ファジィ数結合強度を持つニューラルネット
  3. 提案手法を用いたファジィニューラルネットの進化的学習
    - 3.1 個体集団の初期化
    - 3.2 適合度評価
    - 3.3 交叉
    - 3.4 突然変異
  4. 実験
    - 4.1 目的および方法
    - 4.2 近似精度の評価
    - 4.3 LU モデルと CW モデルの比較
  5. 結論
- 謝辞
- 参考文献

## 1. 序論

これまでに、結合強度の値を従来の実数からファジィ数に拡張したニューラルネットと、その学習方法が提案されている [1]。このファジィニューラルネットを用いれば、実数ベクトル  $\mathbf{x}$  の入力に対してファジィ数  $Y$  を出力する入出力関数  $Y=F(\mathbf{x})$  をモデリングすることができる。しかし、従来の学習方法は誤差逆伝播法を拡張した方法であったため、教師データ  $\{(\mathbf{x}_1, Y_1), (\mathbf{x}_2, Y_2), \dots\}$  が必要であった。

一方、教師データを必要とせずにニューラルネットを学習させる方法として、進化的アルゴリズムを用いた強化学習方法が研究されており [2-5]、ニューロエボリューションと呼ばれている。ニューロエボリューションにおいては、ニューラルネットの結合強度の値は誤差の逆伝播ではなく進化的操作によってチューニングされる。教師データを必要とせずにニューラルネットを学習できるという長所から、ニューロエボリューションは、例えば自律ロボットの行動制御などに応用可能である。しかし、従来のニューロエボリューションにおいては、学習対象のニューラルネットは結合強度の値が実数である古典的なニューラルネットであり、前記のファジィニューラルネットの学習には適用されていなかった。この理由は、進化的アルゴリズムに含まれる進化的プロセスが、実数ベクトルで表現された genotype (解の遺伝子表現形) しか扱えなかったためである。

これに対して本研究ではこれまでに、genotype の値としてファジィ数を利用できるように進化的アルゴリズムを拡張する手法を提案した [6]。この提案手法を用いれば、ファジィニューラルネットの進化的学習が可能になると考えられる。文献 [6] では、進化的アルゴリズムのなかでも特に代表

的な、進化戦略<sup>[7,8]</sup> および遺伝的アルゴリズム<sup>[9,10]</sup> の2手法について、ファジィ数ベクトルとして表現された genotype を扱うための進化的プロセスの拡張方法を示した。本稿では、遺伝的アルゴリズムに関する提案手法を用いてニューロエボリューションを試み、ファジィニューラルネットの進化的学習に対する提案手法の有効性評価を行った。

## 2. ファジィ数結合強度を持つニューラルネット

本研究で用いるファジィニューラルネットは、文献 [1] で提案された、ファジィ数結合強度を持つ階層型ニューラルネットである(図1)。このファジィニューラルネットの入出力計算を以下に示す。なお、簡単化のためここでは、出力層ユニットの数を1とする。

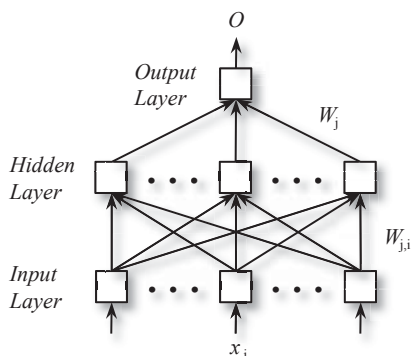


図1：ファジィ数結合強度を持つファジィニューラルネット<sup>[1]</sup>

入力層：

$$o_i = x_i \tag{1}$$

中間層：

$$Net_j = \sum_i W_{j,i} o_i + \theta_j \tag{2}$$

$$O_j = f(Net_j) \tag{3}$$

出力層：

$$Net = \sum_j W_j O_j + \theta \tag{4}$$

$$O = f(Net) \tag{5}$$

式 (1)～(5) において、 $x_i$  および  $o_i$  の値は実数であり、 $Net_j$ ,  $Net$ ,  $W_{j,i}$ ,  $W_j$ ,  $\theta_j$ ,  $\theta$ ,  $O_j$  および  $O$  の値はファジィ数である。式 (3) および式 (5) の  $f(x)$  はユニット内部関数であり、一般的にはシグモイド関数  $f(x)=1/(1+e^x)$  が用いられる。本研究においてもこのシグモイド関数をユニット内部関数

として用いる. 式 (1)~(5) におけるファジィ数の演算は, 拡張原理<sup>[11]</sup> および区間演算<sup>[12]</sup> に従って行われる (詳しくは文献 [1] を参照されたい). シグモイド関数によるファジィ数の写像の例を図 2 に示す.

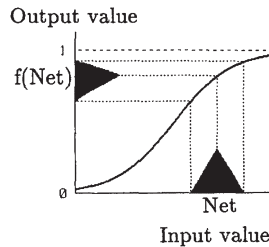


図 2: 中間層および出力層のユニットにおける入出力<sup>[1]</sup>

本研究では文献 [1] と同様に, ファジィニューラルネットの結合強度  $W_{ji}$ ,  $W_j$  およびしきい値  $\theta_j$ ,  $\theta$  の値を対称三角型ファジィ数とする. 対称三角型ファジィ数とは, メンバーシップ関数が図 3 のような左右対称の三角形で表されるファジィ数である. 対称三角型ファジィ数は, サポート集合<sup>[13]</sup> の下限と上限の値 (図 3 の  $L$  および  $U$  の値), もしくは, 中心と幅の値 (図 3 の  $c$  および  $w$  の値) を用いて規定できる. すなわち, 対称三角型ファジィ数  $X_i$  を,  $X_i = [x_i^l, x_i^u]$  もしくは  $X_i = (x_i^c, x_i^w)$  と表すことができる. ここで,  $x_i^l, x_i^u, x_i^c$  および  $x_i^w$  はそれぞれ  $X_i$  の下限, 上限, 中心および幅である. 本稿では, 前者の表現を LU モデル, 後者の表現を CW モデルと呼ぶことにする.

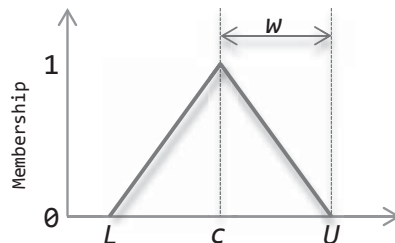


図 3: 対称三角型ファジィ数のメンバーシップ関数と其实数パラメータ

### 3. 提案手法を用いたファジィニューラルネットの進化的学習

図 1 において, 入力層のユニット数を  $n$ , 中間層のユニット数を  $m$  とすると, 図 1 のファジィニューラルネットには  $(mn+m)$  個の結合強度と  $(m+1)$  個のしきい値が含まれ, 全部で  $(mn+2m+1)$  個のファジィ数変数が含まれることになる. 文献 [6] に示した本研究の提案手法を用いれば, この  $(mn+2m+1)$  個のファジィ数を  $(mn+2m+1)$  次元ベクトルの genotype として扱うことができ, 進化的操作によりこれらの結合強度としきい値をチューニング可能である. 本稿では, 進化的アルゴリズムとして遺伝的

アルゴリズムを用いる場合のプロセスを記載する。

### 3.1 個体集団の初期化

遺伝的アルゴリズムにおける個体1つがここではファジィニューラルネット1つに対応し、ある1個体の genotype は  $(mn+2m+1)$  次元ファジィ数ベクトルである。phenotype であるファジィニューラルネットは、この genotype に含まれるファジィ数を、それぞれ対応する場所の結合強度およびしきい値 (式 (1) ~ 式 (5) の  $W_{ji}$ ,  $W_j$ ,  $\theta_j$ ,  $\theta$ ) として用いる。初期化のプロセスにおいては、ある定められた個数の個体の genotype をランダムに初期化する。ある1個体の genotype を  $X_a = (X_{a,1}, X_{a,2}, \dots, X_{a,D})$  と表すことにすると、 $D=mn+2m+1$  であり、 $X_{a,i}=[x_{a,i}^L, x_{a,i}^U]$  もしくは  $X_{a,i}=(x_{a,i}^c, x_{a,i}^w)$  である。また、個体の個数 (集団サイズ) を  $P$  とすると、genotype は  $X_1, X_2, \dots, X_P$  である。この  $P$  個体分の genotype のなかに合計  $PD$  個の対称三角型ファジィ数が含まれ、対称三角型ファジィ数1つは実数2つ (下限と上限、もしくは中心と幅) で規定できるため、 $P$  個体分の genotype のなかに合計  $2PD$  個の実数が含まれている。この  $2PD$  個の実数の値を次のように初期化する。LU モデルを用いる場合は、 $x_{a,i}^L \leq x_{a,i}^U$  でなければならない。そこで、正の数  $\epsilon$  を定めて区間  $[-\epsilon, \epsilon]$  内の一様乱数を2個生成し、小さいほうを  $x_{a,i}^L$  の値、大きいほうを  $x_{a,i}^U$  の値とする ( $i=1,2, \dots, D; a=1,2, \dots, P$ )。一方、CW モデルを用いる場合は、 $0 \leq x_{a,i}^w$  でなければならない。そこで、区間  $[-\epsilon, \epsilon]$  内の一様乱数を2個生成し、一方を  $x_{a,i}^c$  の値、もう一方の絶対値を  $x_{a,i}^w$  の値とする ( $i=1,2, \dots, D; a=1,2, \dots, P$ )。なお、区間  $[-\epsilon, \epsilon]$  内の一様乱数を用いる代わりに例えば  $N(0, \epsilon)$  の正規乱数を用いてもよい。 $\epsilon$  の値は、本研究においてはニューラルネットの結合強度やしきい値の初期値であることから小さめの値が望ましいと考えられる。

### 3.2 適合度評価

genotype  $X_a \in \{X_1, X_2, \dots, X_P\}$  の個体の適合度評価値は、 $X_a$  に対応する phenotype のファジィニューラルネットの性能を評価することで得られる。例えば、このファジィニューラルネットの適用対象問題が自律ロボットの制御であれば、ファジィニューラルネットへの入力値にはロボットのセンサから得られる値を用い、ファジィニューラルネットからの出力をロボットのアクチュエータの制御量として用いる。この場合、 $X_a$  に対応する phenotype のファジィニューラルネットを用いて制御されたロボットが、あらかじめ定められた課題タスクをどれだけうまく達成できたか (例えばタスク達成率、タスク完了時間など) を定量的に求め、その値を  $X_a$  の適合度として用いる。この結果、よりうまく課題タスクを実行できたファジィニューラルネットがより高い評価値を受け取り、その遺伝子を子個体へ継承しやすくなる。

### 3.3 交叉

適合度が評価された現世代の個体  $P$  個を親個体として用い、次世代の子個体を生成する。遺伝的アルゴリズムにおける子個体生成方法は交叉と突然変異である。交叉の方法を以下に示す。

現世代の集団のなかから親個体として選択された2個体の genotype を  $X_a, X_b$  とし、交叉によって生成される子個体の genotype を  $X_z$  とする。 $X_a, X_b$  を現世代の個体集団のなかからサンプリングする方法は従来の遺伝的アルゴリズムと同じでよい。

LU モデルを用いる場合は、

$$X_a = (X_{a,1}, X_{a,2}, \dots, X_{a,D}), X_{a,i} = [x_{a,i}^L, x_{a,i}^U], \quad (6)$$

$$X_b = (X_{b,1}, X_{b,2}, \dots, X_{b,D}), X_{b,i} = [x_{b,i}^L, x_{b,i}^U], \quad (7)$$

$$X_z = (X_{z,1}, X_{z,2}, \dots, X_{z,D}), X_{z,i} = [x_{z,i}^L, x_{z,i}^U]. \quad (8)$$

である。提案手法<sup>[6]</sup>では、 $X_z$  を規定する実数変数  $x_{z,i}^L, x_{z,i}^U$  の値を、従来の実数遺伝的アルゴリズムにおける交叉方法を用いて、 $x_{z,i}^L$  は  $x_{a,i}^L$  と  $x_{b,i}^L$  をもとに、また  $x_{z,i}^U$  は  $x_{a,i}^U$  と  $x_{b,i}^U$  をもとにそれぞれ決定する。その交叉方法としてブレンド交叉<sup>[10]</sup>を用いる場合、 $x_{z,i}^L$  の値は、式(9)で示される区間のなかから一様ランダムにサンプリングする。

$$\begin{aligned} & [\min(x_{a,i}^L, x_{b,i}^L) - \alpha(\max(x_{a,i}^L, x_{b,i}^L) - \min(x_{a,i}^L, x_{b,i}^L)), \\ & \max(x_{a,i}^L, x_{b,i}^L) + \alpha(\max(x_{a,i}^L, x_{b,i}^L) - \min(x_{a,i}^L, x_{b,i}^L))] \end{aligned} \quad (9)$$

ここで、 $\min(x_{a,i}^L, x_{b,i}^L)$  は  $x_{a,i}^L, x_{b,i}^L$  のうち値の小さいほう、 $\max(x_{a,i}^L, x_{b,i}^L)$  は  $x_{a,i}^L, x_{b,i}^L$  のうち値の大きいほうをそれぞれ表し、 $\alpha$  は非負の実数である。

$x_{z,i}^U$  の値も  $x_{z,i}^L$  と同様に、親の値である  $x_{a,i}^U, x_{b,i}^U$  を用いて、式(10)で示される区間のなかから一様ランダムにサンプリングする。

$$\begin{aligned} & [\min(x_{a,i}^U, x_{b,i}^U) - \alpha(\max(x_{a,i}^U, x_{b,i}^U) - \min(x_{a,i}^U, x_{b,i}^U)), \\ & \max(x_{a,i}^U, x_{b,i}^U) + \alpha(\max(x_{a,i}^U, x_{b,i}^U) - \min(x_{a,i}^U, x_{b,i}^U))] \end{aligned} \quad (10)$$

ただし、 $x_{z,i}^L, x_{z,i}^U$  は  $x_{z,i}^L \leq x_{z,i}^U$  の制約を満たす必要がある。そこで、式(9)および式(10)を用いて  $x_{z,i}^L, x_{z,i}^U$  の値を求めた結果、 $x_{z,i}^L > x_{z,i}^U$  となった場合は  $x_{z,i}^L \leq x_{z,i}^U$  の制約が満たされるように  $x_{z,i}^L, x_{z,i}^U$  の値を修正する<sup>[6]</sup>。

一方、CW モデルを用いる場合は、

$$\mathbf{X}_a = (X_{a,1}, X_{a,2}, \dots, X_{a,D}), X_{a,i} = (x_{a,i}^c, x_{a,i}^w), \quad (11)$$

$$\mathbf{X}_b = (X_{b,1}, X_{b,2}, \dots, X_{b,D}), X_{b,i} = (x_{b,i}^c, x_{b,i}^w), \quad (12)$$

$$\mathbf{X}_z = (X_{z,1}, X_{z,2}, \dots, X_{z,D}), X_{z,i} = (x_{z,i}^c, x_{z,i}^w). \quad (13)$$

である。LU モデルの場合と同様にブレンド交叉<sup>[10]</sup>を用いる場合、 $x_{z,i}^c$ の値は、式(14)で示される区間のなかから一様ランダムにサンプリングする。

$$\begin{aligned} & [\min(x_{a,i}^c, x_{b,i}^c) - \alpha(\max(x_{a,i}^c, x_{b,i}^c) - \min(x_{a,i}^c, x_{b,i}^c)), \\ & \max(x_{a,i}^c, x_{b,i}^c) + \alpha(\max(x_{a,i}^c, x_{b,i}^c) - \min(x_{a,i}^c, x_{b,i}^c))] \end{aligned} \quad (14)$$

$x_{z,i}^w$ の値も $x_{z,i}^c$ と同様に、親の値である $x_{a,i}^w, x_{b,i}^w$ を用いて、式(15)で示される区間のなかから一様ランダムにサンプリングする。

$$\begin{aligned} & [\min(x_{a,i}^w, x_{b,i}^w) - \alpha(\max(x_{a,i}^w, x_{b,i}^w) - \min(x_{a,i}^w, x_{b,i}^w)), \\ & \max(x_{a,i}^w, x_{b,i}^w) + \alpha(\max(x_{a,i}^w, x_{b,i}^w) - \min(x_{a,i}^w, x_{b,i}^w))] \end{aligned} \quad (15)$$

ただし、 $x_{z,i}^w$ は $0 \leq x_{z,i}^w$ の制約を満たす必要がある。そこで、式(15)を用いて $x_{z,i}^w$ の値を求めた結果、 $0 > x_{z,i}^w$ となった場合は $0 \leq x_{z,i}^w$ の制約が満たされるように $x_{z,i}^w$ の値を修正する<sup>[6]</sup>。

### 3.4 突然変異

交叉によって生成された子個体の genotype に含まれる実数変数の値を、あらかじめ定められた確率にしたがってランダムな値に書き換える。この書き換えに用いる新しい値の決め方は、初期化プロセスにおいてその変数の初期値を決定する際に用いた方法と同一でよい。ただし、LU モデルを用いる場合において、 $x_{z,i}^L$ あるいは $x_{z,i}^U$ の値を書き換えた結果 $x_{z,i}^L > x_{z,i}^U$ となった場合は、交叉のときと同様に $x_{z,i}^L, x_{z,i}^U$ の値を修正する。CW モデルを用いる場合においても、 $x_{z,i}^w$ の値を書き換えた結果 $0 > x_{z,i}^w$ となった場合は、交叉のときと同様に $x_{z,i}^w$ の値を修正する。

## 4. 実験

### 4.1 目的および方法

提案手法による進化的学習は教師データを用いない学習法であり、結合強度のチューニング方法も、誤差関数の勾配を利用しない確率的探索に基づく方法である。さらに、従来のニューロエボリューションの場合と異なり、学習対象であるニューラルネットの結合強度の値は実数ではなくファジィ数である。このような進化的学習を提案手法によって実際に行えるかどうか評価するために、数値実験を行った。

本実験におけるファジィニューラルネットの学習目標は、明示的には与えられないファジィ関数を近似することである。そのファジィ関数としてここでは3種類の関数  $F_1(x) = [F_1(x)^L, F_1(x)^U]$ ,  $F_2(x) = [F_2(x)^L, F_2(x)^U]$  および  $F_3(x) = [F_3(x)^L, F_3(x)^U]$  を設計した。ここで、 $F_1(x)$ ,  $F_2(x)$ ,  $F_3(x)$  はいずれも実数入力  $x$  に対して対称三角型ファジィ数を出力する関数であり、 $F(x)^L$ ,  $F(x)^U$  は  $F(x)$  の下限および上限を表している。また簡単化のため、文献 [1] と同様に、入力  $x$  の次元数は1とし、その定義域を  $0 \leq x \leq 1$  とする。

$$F_1(x)^L = 0.2\sin(2\pi x) - 0.1x^2 + 0.4 \quad (16)$$

$$F_1(x)^U = 0.2\sin(2\pi x) + 0.1x^2 + 0.6 \quad (17)$$

$$F_2(x)^L = 0.2\sin(2\pi x) + 0.2x^2 + 0.2 \quad (18)$$

$$F_2(x)^U = 0.2\sin(2\pi x) - 0.2x^2 + 0.7 \quad (19)$$

$$F_3(x)^L = 0.15\sin(2\pi x) + 0.3 \quad (20)$$

$$F_3(x)^U = 0.1\sin(3\pi x) - 0.1x + 0.7 \quad (21)$$

目標関数  $F_1(x)$ ,  $F_2(x)$ ,  $F_3(x)$  の図を図 4-6 に示す。図 4-6 において、

- F0.0L および F0.0U はそれぞれ、 $F(x)^L$  および  $F(x)^U$  を（つまり  $F(x)$  のサポート集合の下限と上限を）表している。
- F0.5L および F0.5U はそれぞれ、 $F(x)$  の 0.5 レベル集合 ( $F(x)|_{0.5}$ ) の下限と上限を表している。
- F1.0 は、 $F(x)$  の 1.0 レベル集合 ( $F(x)|_{1.0}$ ) を表している。 $F(x)$  は対称三角型ファジィ数のため、 $F(x)|_{1.0}$  は点（幅がゼロの区間）となる。

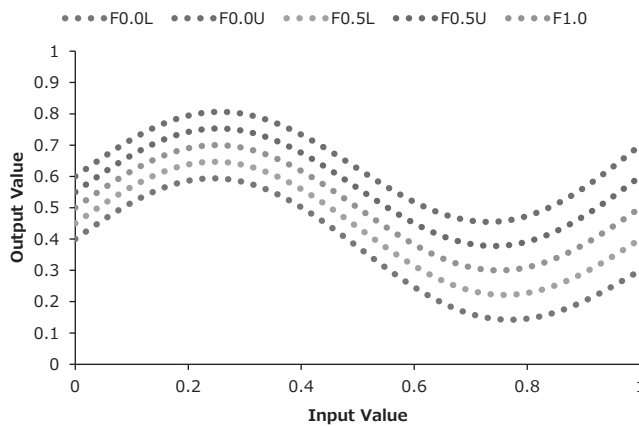


図 4：目標関数  $F_1(x)$



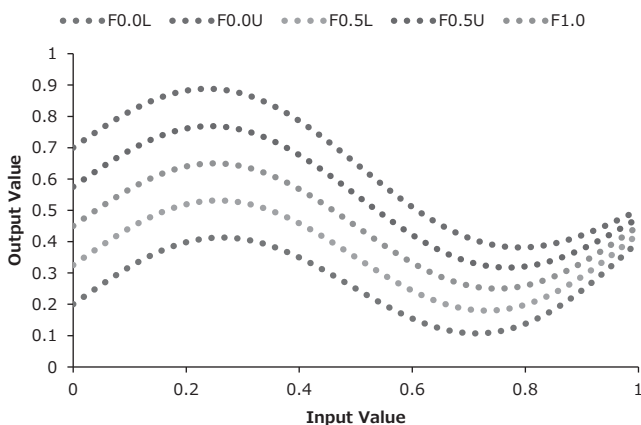


図 5：目標関数  $F_2(x)$

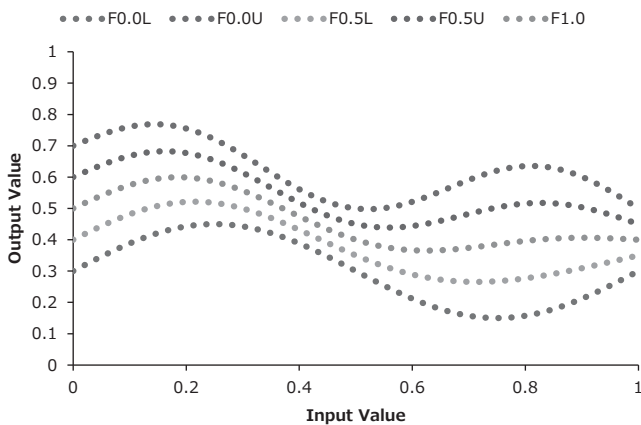


図 6：目標関数  $F_3(x)$

次に、目標関数  $F_1(x)$ ,  $F_2(x)$ ,  $F_3(x)$  を近似させるファジィニューラルネットを次のように設計した。

- ユニット数：入力層 1，中間層 10，出力層 1
- $\epsilon$  : 0.01
- $-10.0 \leq x_{a,i}^L, x_{a,i}^U, x_{a,i}^c \leq 10.0$
- $0.0 \leq x_{a,i}^w \leq 10.0$

また、提案手法であるファジィ遺伝的アルゴリズムは次のように設計した。

- 進化シミュレーション試行 1 回につき生成されるファジィニューラルネットの総数：1,000,000

- 集団サイズおよび総世代数：(100 個体, 10,000 世代) および (500 個体, 2,000 世代)
- ブレンド交叉の定数  $\alpha$  : 0.5
- 突然変異確率 : 0.01
- エリート保存個体数 : 10
- 交叉に用いる親 2 個体の選択方法 : トーナメント選択 (トーナメントサイズは集団サイズの 5%)

集団サイズと総世代数の組合せを、総世代数優先の (100 個体, 10,000 世代) の組合せと、個体数優先の (500 個体, 2,000 世代) の 2 種類設計し、いずれの組合せの場合も進化シミュレーション試行 1 回につき生成される個体数 (= 集団サイズ  $\times$  総世代数) が 1,000,000 となるようにした。

集団内の  $P$  個の個体の適合度は、文献 [1] と同一の評価関数を利用して計算することとした。ただし、文献 [1] における学習は教師データを用いた学習のため、あらかじめ教師データとして  $\{(x_1, F(x_1)), (x_2, F(x_2)), \dots\}$  が与えられている。一方、本研究ではそのような教師データは与えられない。そこで本実験では、学習対象のファジィニューラルネットが実際の課題タスクに応用された場合に、入力値が定義域内において一様にサンプリングされるようなケースを想定し、入力値の定義域  $[0, 1]$  内の値として  $x_r = 0.0, 0.01, 0.02, \dots, 1.0$  をサンプルに用いることとした。サンプル入力値  $x_r$  に対してファジィニューラルネットから出力されるファジィ数を  $O_r$ 、サンプル入力値  $x_r$  に対する目標関数の出力を  $F_r$  とすると、 $O_r$  と  $F_r$  の誤差  $e_r$  は、

$$e_r = \sum_h h((o_{r,h}^L - f_{r,h}^L)^2 + (o_{r,h}^U - f_{r,h}^U)^2) \quad (22)$$

である [1]。ここで、

- $o_{r,h}^L$  および  $o_{r,h}^U$  はそれぞれ  $O_r$  の  $h$  レベル集合の下限および上限であり (つまり  $O_r|_h = [o_{r,h}^L, o_{r,h}^U]$ )、
- $f_{r,h}^L$  および  $f_{r,h}^U$  はそれぞれ  $F_r$  の  $h$  レベル集合の下限および上限である (つまり  $F(x_r)|_h = [f_{r,h}^L, f_{r,h}^U]$ )。

$h$  レベル集合を求めるための  $h$  の値は、文献 [1] と同様、 $h = 0.2, 0.4, \dots, 1.0$  とした。

$X_1, X_2, \dots, X_p$  のそれぞれについて、その genotype の値をデコードして得られたファジィニューラルネットに入力値  $x_r = 0.0, 0.01, 0.02, \dots, 1.0$  をそれぞれ入力し、各入力に対する  $e_r$  の総和をその個体の評価値とする。この評価値は誤差に基づく値のため、より値の小さい個体のほうがより適合度の高い個体である。したがって、得られた評価値の小さい個体ほど集団内でランク上位の個体とする。ここで注意すべきことは、文献 [1] における学習とは異なり、本研究では  $e_r$  の値を結合強度のチューニング量の計算には用いていないという点である。ここでは実験を簡単にするために文献 [1] と同じ誤差  $e_r$  の値を用いているが、実際の応用では、提案手法の場合には課題タスクに対するファジィニューラルネットの性能を評価して集団内の複数の個体間で順位づけできるだけで良く、入力値  $x_r$

に対する望ましい出力値  $F_r$  は未知であっても提案手法を適用可能である。

## 4.2 近似精度の評価

前節に記載した方法で実験を行ったところ、図 7-12 に示す結果が得られた。図 7 は、関数  $F_1(x)$  を学習目標とした進化シミュレーションにおいて生成された全 20,000,000 個体 (= [1 試行あたり 1,000,000 個体] × [5 試行] × [(集団サイズ, 総世代数) の組合せ 2 通り] × [LU モデル, CW モデルの 2 通り]) のなかで最良評価値を得た個体のファジィニューラルネットが出力したファジィ数を表している。また図 8, 9 はそれぞれ、目標関数  $F_2(x)$  および  $F_3(x)$  に対する結果を図 7 と同様に表している。図 7-9 において、

- F0.0L, F0.0U, F0.5L, F0.5U および F1.0 は、いずれも図 4-6 と同様である。
- NN0.0L および NN0.0U はそれぞれ、ファジィニューラルネットが出力したファジィ数のサポート集合の下限および上限を表している。
- NN0.5L および NN0.5U はそれぞれ、ファジィニューラルネットが出力したファジィ数の 0.5 レベル集合の下限および上限を表している。
- NN1.0 は、ファジィニューラルネットが出力したファジィ数の 1.0 レベル集合を表している (一般的には区間だがここでは下限と上限が一致している)。

図 7-9 より、提案手法によって進化的に生成されたファジィニューラルネットは、いずれの目標関数も概ねよく近似できていることがわかる。図 8 および図 9 において、N0.0L, N0.0U がそれぞれ F0.0L, F0.0U とやや離れている箇所があるが、これらはメンバーシップ値が最小の (ファジィ数のサポート集合の) 下限や上限の値であり、誤差評価 (適合度評価) において最も重みの小さいレベル集合である。式 (22) からわかるように、誤差評価 (適合度評価) の値はメンバーシップ値  $h$  で重みづけされており [1], より大きな  $h$  の値におけるレベル集合の誤差がより小さい個体ほど、より高く評価され上位にランク付けされる。また本研究の実験では、文献 [1] における実験と同様に、サポート集合の誤差は誤差評価 (適合度評価) には用いていない (誤差の計算に用いる  $h$  レベル集合の  $h$  の値は 0.2, 0.4, ..., 1.0 であり、0.2 未満のメンバーシップ値  $h$  に対する  $h$  レベル集合の誤差は評価値に含まれない)。つまり、式 (22) の評価によりよく適合するよう (より大きなメンバーシップ値における  $h$  レベル集合の誤差がより小さくなるよう) 自律的に進化が進み、図 7-9 のような出力のファジィニューラルネットが生成されたと考えられる。教師データが与えられず学習目標の関数も明示的には与えられていないにも関わらず、提案手法を用いれば、ファジィニューラルネットを進化的に学習させて目標関数を近似できたことがわかる。

近似の精度をさらに確認するため、進化的に得られた最良ファジィニューラルネットが出力するファジィ数のメンバーシップ関数を、目標関数が出力するファジィ数のメンバーシップ関数と比較す

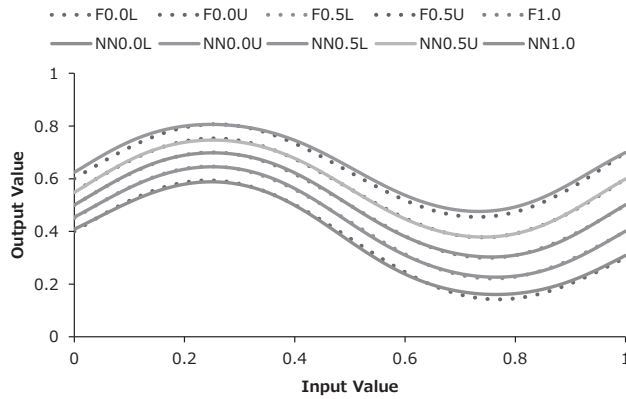


図 7：目標関数  $F_1(x)$  に対する最良個体のファジィニューラルネットの出力 ( $e_r$  総和 =  $4.4E-3$ )

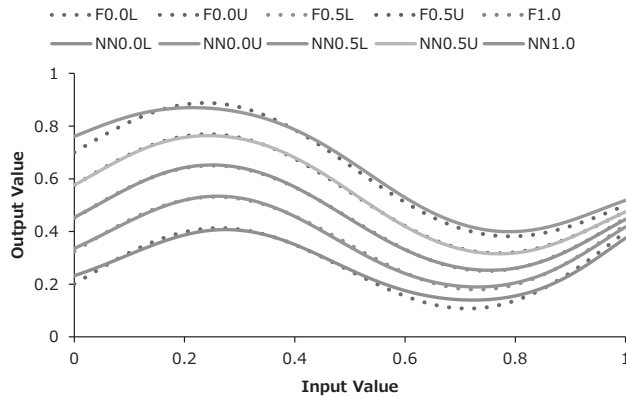


図 8：目標関数  $F_2(x)$  に対する最良個体のファジィニューラルネットの出力 ( $e_r$  総和 =  $9.2E-3$ )

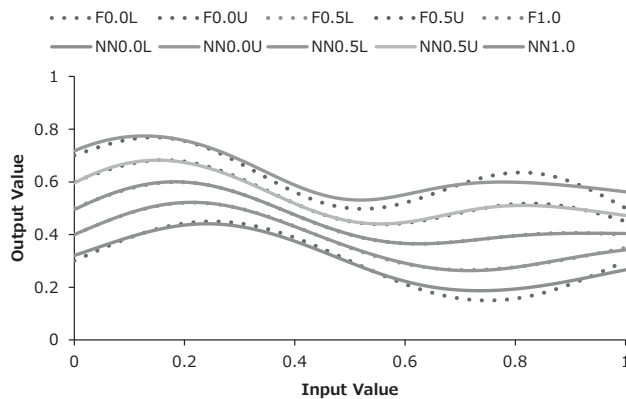


図 9：目標関数  $F_3(x)$  に対する最良個体のファジィニューラルネットの出力 ( $e_r$  総和 =  $9.6E-3$ )

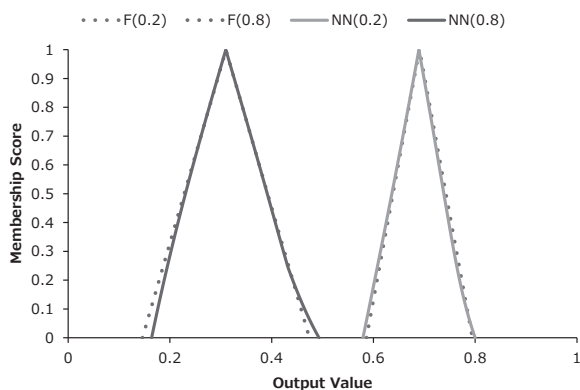


図 10：入力  $x=0.2, 0.8$  に対する目標関数  $F_1(x)$  の出力ファジィ数 ( $F(0.2), F(0.8)$ ) および最良ファジィニューラルネットの出力ファジィ数 ( $NN(0.2), NN(0.8)$ )

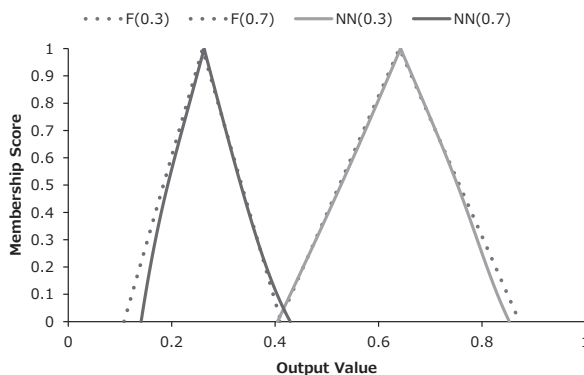


図 11：入力  $x=0.3, 0.7$  に対する目標関数  $F_2(x)$  の出力ファジィ数 ( $F(0.3), F(0.7)$ ) および最良ファジィニューラルネットの出力ファジィ数 ( $NN(0.3), NN(0.7)$ )

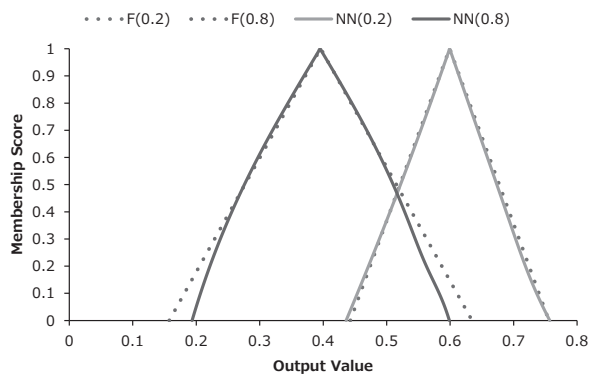


図 12：入力  $x=0.2, 0.8$  に対する目標関数  $F_3(x)$  の出力ファジィ数 ( $F(0.2), F(0.8)$ ) および最良ファジィニューラルネットの出力ファジィ数 ( $NN(0.2), NN(0.8)$ )

る. 図 10 は, 入力  $x=0.2, 0.8$  に対して目標関数  $F_1(x)$  が出力したファジィ数 (図中の  $F_{0.2}, F_{0.8}$ ), および, 同じ入力に対して図 7 の最良ファジィニューラルネットが出力したファジィ数 (図中の  $NN_{0.2}, NN_{0.8}$ ) を表している. 目標関数  $F_1(x)$  が出力するファジィ数は対称三角型だが, ファジィニューラルネットが出力するファジィ数は, ユニット内部関数であるシグモイド関数の写像のため三角型とはならない. しかし, 両者の形状は概ねよく一致しており, 特に, メンバーシップ値が大きい領域ほどよく一致していることがわかる. 図 10 と同様に, 図 11, 12 はそれぞれ目標関数  $F_2(x)$  および  $F_3(x)$  に対する結果を表している. ここで, 図 11 だけ入力  $x$  の値として  $0.2, 0.8$  ではなく  $0.3, 0.7$  を用いた理由は, 図 8 より, 目標関数  $F_2(x)$  に対する最良ニューラルネットの出力は,  $x=0.2, 0.8$  より  $x=0.3, 0.7$  のほうが誤差がやや大きいことがわかったためである.  $F_1(x)$  の結果である図 10 に比べると,  $F_2(x)$  および  $F_3(x)$  の結果である図 11, 図 12 のほうが, 対応する 2 つのファジィ数の誤差がやや大きいことがわかる. しかし図 11, 図 12 においても, メンバーシップ値がより大きい領域ほどよく一致していることは図 10 と同様であり, 式 (22) の評価によりよく適合するよう自律的に進化が進んだことが改めて伺われる.

### 4.3 LU モデルと CW モデルの比較

対称三角型ファジィ数  $X_i$  を規定する方法としては, 2 章に記載した通り, LU モデルおよび CW モデルの 2 種類を利用できる. つまり, 本研究においては genotype の値に, 対称三角型ファジィ数の下限と上限の値, もしくは中心と幅の値を用いることができる. いずれのモデルを利用した場合でも, genotype のファジィ数ベクトルから phenotype のファジィニューラルネットへデコードする方法には変わりがないが, 提案手法によってファジィ数ベクトルの genotype を扱えるように拡張された進化的アルゴリズムにおいては, 利用するモデルに依存して解の探索領域に差が生じる. この理由は, 下限・上限・中心・幅の 4 変数のうち, 下限・上限・中心の値は負の値になりうるが, 幅だけは非負でなければならないためである. 3 章にて報告した実験においては, 下限・上限・中心の値の定義域は  $-10.0 \leq x_{a,i}^L, x_{a,i}^U, x_{a,i}^c \leq 10.0$  であり, 幅の値の定義域は  $0.0 \leq x_{a,i}^w \leq 10.0$  とした. このため, CW モデルのほうが LU モデルより探索空間が小さい. 一般的に, 進化的アルゴリズムによる解の探索は探索空間が小さいほど有利と期待される. ここで, 実験において  $0.0 \leq x_{a,i}^w \leq 10.0$  ではなく  $0.0 \leq x_{a,i}^w \leq 20.0$  と設定すれば, 進化的アルゴリズムが解を探索する空間の大きさは LU / CW モデル間で公平になるが,  $-10.0 \leq x_{a,i}^c \leq 10.0, 0.0 \leq x_{a,i}^w \leq 20.0$  を LU モデルに置き換えると  $-30.0 \leq x_{a,i}^L, x_{a,i}^U \leq 30.0$  となり, phenotype として生成されうるファジィニューラルネットが LU / CW モデル間で大幅に異なってしまう. つまり, CW モデルでは結合強度の下限の値として  $-30.0$ , 上限の値として  $30.0$  を持つようなファジィニューラルネットが生成されうるが, LU モデルでは結合強度の下限の値として  $-10.0$ , 上限の値として  $10.0$  を持つようなファジィニューラルネットしか生成されない. ユニット内部関数として用いたシグモイド関数の形状から, ユニット内部関数に入力される値 (式 (2) の  $Net_i$  や式 (4) の  $Net$  の値) は  $[-10.0, 10.0]$  程度の範囲内で十分と考えられる (これより小さな値や大きな値が

入力されても出力値は $-10.0$ や $10.0$ の入力の場合とほとんど差がないことから、結合強度の値も $[-10.0, 10.0]$ 程度の範囲内で十分であり、過剰に小さな／大きな結合強度の値まで探索範囲に含めると、汎化能力に欠ける局所的最適解へと陥る可能性が高まると考えられる。実験では $0.0 \leq x_{a,i}^w \leq 10.0$ と設定したが、この場合であってもLUモデルに置き換えれば $-20.0 \leq x_{a,i}^L, x_{a,i}^U \leq 20.0$ であり、 $-10.0 \leq x_{a,i}^L, x_{a,i}^U \leq 10.0$ よりも広く設定されていることになる。

以上から、提案手法によるファジィニューラルネットの進化的学習において、genotypeの値の規定方法としてLUモデルおよびCWモデルのいずれがより望ましいかを評価する必要がある。そこで本節では、前節に記載した実験の結果をLUモデルとCWモデルの間で比較する。

図13は、 $F_1(x)$ の近似を目標にして進化的に学習させたファジィニューラルネットのうち、横軸で示された値の個数だけ個体（ファジィニューラルネット）が生成された時点で、それまでに生成された全個体のうち最良の個体に関する誤差評価値（ $e_r$ の総和の値）を示している。例えば、横軸の値が400,000のとき、生成された個体の総数が400,000であり、集団サイズ100の実験では世代数4,000、集団サイズ500の実験では世代数800が経過した時点を表している。図13において、LU100／LU500のグラフはLUモデルで集団サイズ100／500の結果を、CW100／CW500のグラフはCWモデルで集団サイズ100／500の結果をそれぞれ表している。プロットされた誤差評価値は、同じ設定で行った進化シミュレーション5試行分の平均値である。図13と同様に、図14、図15はそれぞれ $F_2(x)$ および $F_3(x)$ を用いた場合の結果を表したものである。

図13-15より、 $F_1(x) \sim F_3(x)$ のいずれの場合も最終的に、CWモデルのほうがLUモデルより適合度の高い（誤差評価値の小さい）個体を獲得できていたことがわかる。このことから、提案手法を用いたファジィニューラルネットの進化的学習において、遺伝子の値であるファジィ数が対称三角型の場合には、LUモデルよりCWモデルを用いるほうが望ましいと考えられる。ただし、このことが他の応用対象の場合にもあてはまるかどうか、今後さらに評価する必要がある。

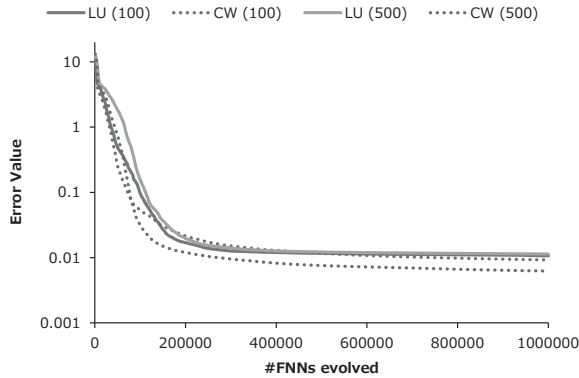


図13：目標関数 $F_1(x)$ に対して各世代までに生成された全個体のうち最良個体の誤差評価値

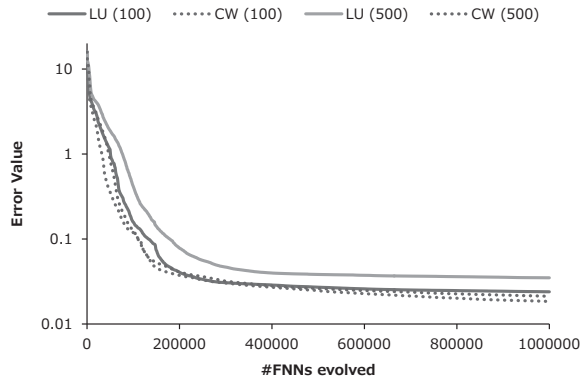


図 14：目標関数  $F_2(x)$  に対して各世代までに生成された全個体のうち最良個体の誤差評価値

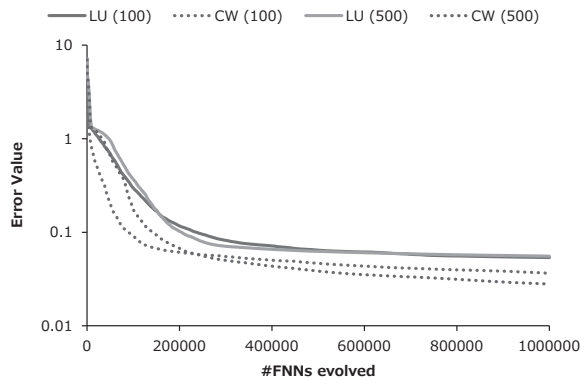


図 15：目標関数  $F_3(x)$  に対して各世代までに生成された全個体のうち最良個体の誤差評価値

## 5. 結論

本稿では、著者が先の研究<sup>[6]</sup>で提案したファジィ進化的アルゴリズムを用いてファジィニューラルネットの進化的学習を試み、その有効性を数値実験によって評価した。提案手法は進化的アルゴリズムにおける genotype の値として実数ではなくファジィ数を利用できるため、結合強度の値がファジィ数に拡張されたニューラルネットであってもその値を genotype の値として用いることで進化的学習が可能と期待できる。実験では、結合強度の値であるファジィ数を対称三角型とし、進化的アルゴリズムとしては遺伝的アルゴリズムを用いた。また、ファジィニューラルネットの学習目標は、非線形な実数入力・ファジィ数出力関数を近似することとした。実験の結果、教師データが与えられず学習目標の関数も明示的には与えられていないにも関わらず、提案手法によってファジィニューラルネットを自律的に進化させ、目標関数を近似させることができた。また、対称三角型ファジィ数を規定する方法として LU モデルと CW モデルを比較した結果、CW モデルを用いたほうがより適合度の高いファジィニューラルネットが生成され、CW モデルの利用が望ましいことがわかった。



今後の課題としては、提案手法<sup>[6]</sup>のさらなる有効性評価を目的として、以下のような評価を実施することが挙げられる。

- 独自に設計した目標関数の近似ではなく、実应用到した具体的な課題タスクを用いた場合の評価
- 結合強度の値であるファジィ数が対称三角型ではなく、非対称三角型、釣鐘型、台形型などの場合の評価
- 遺伝的アルゴリズム以外の進化的アルゴリズムを用いた場合の評価
- ファジィニューラルネットの進化的学習とは異なるファジィ最適化問題に応用した場合の評価

## 謝辞

本研究は京都産業大学特定研究課題 E1207 「ニューロエボリューションのための進化計算手法」の研究の一部として実施された。

## 参考文献

- [1] H. Ishibuchi, H. Tanaka, and H. Okada, Fuzzy neural networks with fuzzy weights and fuzzy biases, *Proc. of IEEE International Conferences on Neural Networks*, pp.1650–1655, 1993.
- [2] D.B. Fogel, L.J. Fogel, and V.W. Porto, Evolving neural networks, *Biological Cybernetics*, vol.63, issue 6, pp.487–493, 1990.
- [3] X. Yao, Evolving artificial neural networks, *Proc. of the IEEE*, vol.87, no.9, pp.1423–1447, 1999.
- [4] K.O. Stanley, and R. Miikkulainen, Evolving neural networks through augmenting topologies, *Evolutionary Computation*, vol.10, no.2, pp.99–127, 2002.
- [5] D. Floreano, P. Durr, and C. Mattiussi, Neuroevolution: from architectures to learning, *Evolutionary Intelligence*, vol.1, no.1, pp.47–62, 2008.
- [6] 岡田, Genotype の値をファジィ数に拡張した進化計算手法の提案, 京都産業大学 総合学術研究所 所報, 第 8 号, pp.33–42, 2013.
- [7] H-P. Schwefel, *Evolution and Optimum Seeking*, Wiley, 1995.
- [8] H-G. Beyer, and H-P Schwefel, Evolution strategies - a comprehensive introduction, *Natural Computing*, vol.1, no.1, pp.3–52, 2002.
- [9] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Professional, 1989.
- [10] L.J. Eshelman and J.D. Schaffer, Real-coded genetic algorithms and interval-schemata, in D.L. Whitley (ed), *Foundation of Genetic Algorithms 2*, pp.187–202, 1993.
- [11] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning - I, II, and III, *Information Sciences*, vol.8, pp.199–249, pp.301–357, and vol.9, pp.43–80, 1975.
- [12] G. Alefeld, and J. Herzberger, *Introduction to Interval Computation*, Academic Press, 1983.
- [13] 古川, ファジィ最適化の数理, 森北出版, 1999.

# Evolutionary Training of Fuzzy Neural Networks by Genetic Algorithm with Fuzzy Genotype Values

Hidehiko OKADA

## Abstract

The author has proposed an extension of genetic algorithm (GA) for solving fuzzy-valued optimization problems. In the proposed GA, genotype values are not real numbers but fuzzy numbers. Evolutionary processes in GA are extended so that GA can handle genotype instances with fuzzy numbers. In this article, the author applies the proposed method to evolving fuzzy neural networks (FNNs). In the FNNs, values of weights and biases are not real numbers but fuzzy numbers. The ordinary GA cannot be applied to the neuroevolution of the FNNs because evolutionary processes in the ordinary GA are not designed to handle fuzzy numbers as genotype values. On the contrary, the proposed GA can adopt fuzzy numbers of FNN weights and biases directly as genotype values: the weights and biases are tuned by evolutionary operations of the proposed GA, not by the traditional back propagation algorithm. Experimental results showed that fuzzy neural networks evolved by the fuzzy GA could model hidden target fuzzy functions well despite the fact that no training data was explicitly provided. Besides, the experimental results revealed that, in the case of adopting symmetric triangular fuzzy numbers as fuzzy genotype values, the center and width model could contribute better than the lower and upper model did for evolving the FNNs.

**Keywords** : evolutionary algorithms, genetic algorithm, fuzzy number, neural network, neuroevolution