

カオスとニューロ、分散処理

ー現実世界の複雑性とコンピューティングー

豊橋創造大学 情報ビジネス学部

教授 今井 正文

E-mail:mimai@sozo.ac.jp

URL:<http://theoria.sozo.ac.jp/mimai/>

1.コンピュータのイメージと実際

- ◆ 一般的にコンピュータは、非常に高速に緻密な計算ができるとか、色々な問題を高速に解いてくれる、ある意味で万能な機械のような説明をされている。
- ◆ 一方、家庭におけるパソコンは、ワープロ、表計算、電子メール、インターネット、CD、DVD鑑賞、画像編集、データベース、etc、として使われている。(ソフトさえあれば様々な機能を果すという意味で万能・多機能だが。。。)
- ◆ イメージと実際の利用が異なったり混乱する原因は、コンピュータの色々な使い方の説明があまりされていない。
- ◆ 高速な計算能力は元々は数値計算や、制御に使われていたが、個人利用のためのパソコンとソフトウェアが発展・普及する段階で大きく利用方法が変わった。

本日の内容

- ◆ 本来のコンピュータの使い方「シミュレーション」について
 1. シミュレーションの基本と応用
(システムやモデルの概念)
 2. 複雑性の科学(カオス理論)
 3. ソフトコンピューティング
 - 3.1 ニューラルネットワーク
 - 3.2 遺伝的アルゴリズム(GA)
- ◆ 現実の問題の自動シミュレーションへのアプローチ
 4. GA-ニューラルネット
 5. 多段システムと分散処理
 5. コンピューティングと適応(エージェントシステム)

2. 1コンピューティングの基礎

基本的な用語

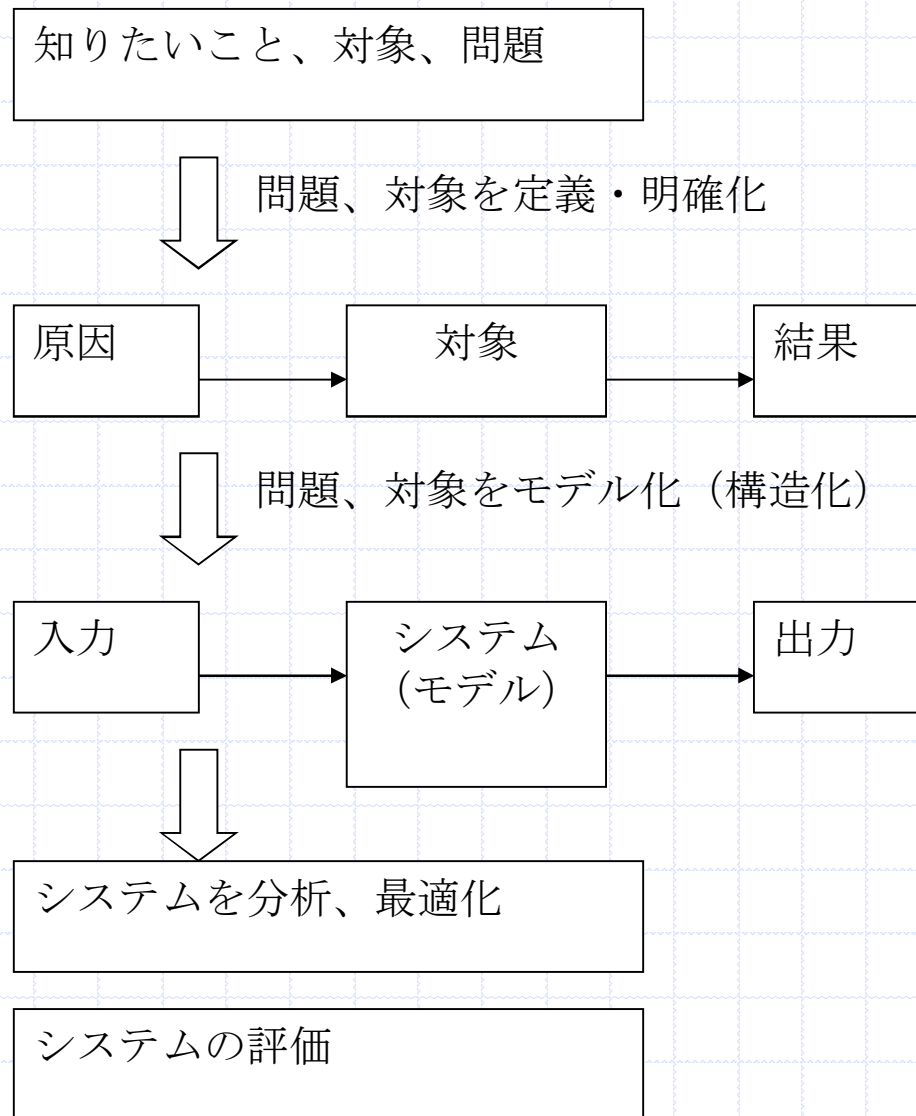
- ◆ コンピューティングとは、
「狭義では計算する、広義ではコンピュータを利用して何らかの処理をすること」
- ◆ シミュレーションとは
「複雑なシステムの仮想実験(狭義ではコンピュータ上で)を行うこと」(モデルが必要)
- ◆ ここでいうシステムは、
「(全体として特定の目的を果たす、)相互に結合した多数の構成要素の有機的結合体」

2. 2モデルとは

基本は入出力の関係だが、対象に対して単純化して関係を整理したもの（簡単にいうと構造化したもの）。

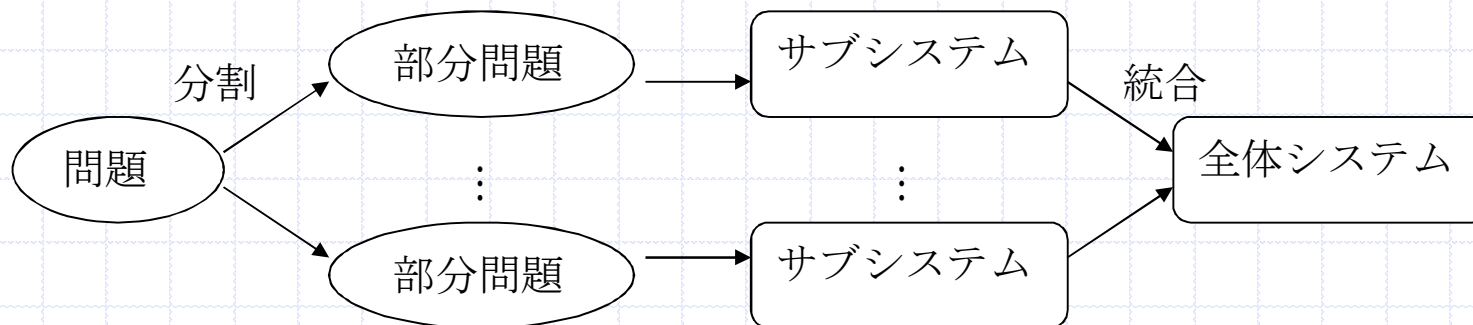
「システム工学の基本は、モデル化」

(ちなみに、環境は外部システム、対象は内部システムとして捉える。)



全体システムと部分システム

問題がすぐに単純化できない場合は、問題を部分問題に分割してサブシステム(部分システム)を作り、サブシステムを統合して全体システムを得る。(要素還元的な考え方ともいえる。)



2.3 順問題と逆問題

◆ 順問題: 入力を与えて出力を求める演算的な問題(コンピュータの得意分野)

例題: $1+2=?$ 解: $1+2=3$ で、解は1個。

◆ 逆問題: 結果から原因を探る問題(ばかばかしい話だが、難しいことは確か。コンピュータも逆問題は苦手)

例題: $?+?=3$ 解: 0も含めた正の整数なら解は4個。

$0+3=3, 1+2=3, 2+1=3, 3+0=3$

小数を含めると解けなくなる(解の候補が無限になる)。

$0+3=3, 0.01+2.99=3, 0.02+2.98=3, \dots, 1+2=3, \dots$

負の整数を含めると解けなくなる。

$100+(-97)=3, 99+(-96)=3, \dots, 4+(-1)=3, 3+0=3, \dots$

逆問題は、条件や情報次第。例の場合、 $1+?=3$ なら?

カオスとニューロ、分散処理

ー現実世界の複雑性とコンピューティングー

2.4モデル化の問題点(ここまでのまとめ)

我々は、多くの人や物事から影響を受け、また、多くの人や物事に対して影響を与える。さらに、システムを構成する要素同士が互いに影響を与えるつまり相互作用することによりシステム全体の挙動は決定される。つまり、複数の構成要素が相互作用することが重要、言い換えると要素間の関係が重要。

◆ 問題点

モデル化で重要でない部分は単純化して大丈夫か？
サブシステムは統合で全体システム(現実を映したもの)になるのか？

◆ 基本的な解答

相互作用(要素間の関係)が線形(足し算)なら大丈夫
⇔ 関係が非線形(掛け算がある)の場合が問題となる。

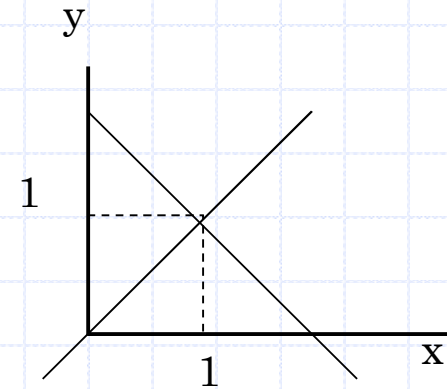
3. 非線形が問題な理由(複雑性の科学)

- ◆ 線形なら、答えを求めるのは比較的簡単。
(機械であれば安全であると保障できる。微分方程式なら、安定、安定限界、不安定の3種類)
- ◆ 非線形は大部分の問題が解けない。
(解が存在するか分からない、機械であれば安全かどうか
いえない)。
非線形部分(掛け算の成分)は、値が小さければ無視で
きるが、それ以外では大きな値になってしまう。
また、カオスと呼ばれる難しい現象が生じやすい。
⇒複雑なものは複雑なまま考えられないか？
(複雑性の科学といわれる分野、まだ定義は微妙)

3.1 方程式の話 (線形の場合)

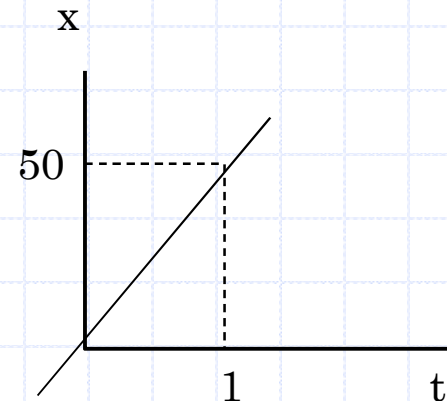
◆ 連立方程式: 答えは時間で変わらない

$$\begin{cases} y = x \\ y = -x + 2 \end{cases} \quad \text{答え} \quad \begin{cases} x = 1 \\ y = 1 \end{cases}$$



◆ 微分方程式: 時間 t で解が動く (動特性)
積分すると答

$$\frac{dx}{dt} = 50 \quad \begin{array}{l} \text{えは } c=0, t=1 \\ \text{なら } x=50 \end{array} \quad x = 50t + c$$



(例: 時速 50km/h の車は、最初の地点を 0 とすると 1 時間後に 50km の場所に移動する。微分 \div 時速。グラフでは傾き)

線形の連立微分方程式の場合

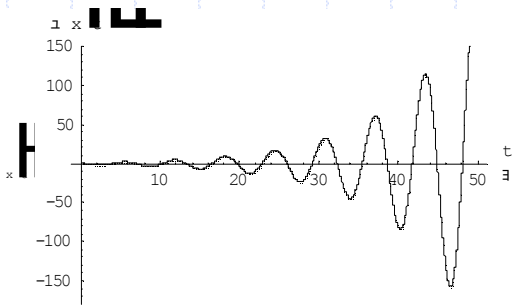
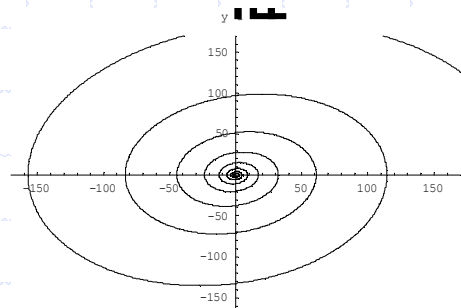
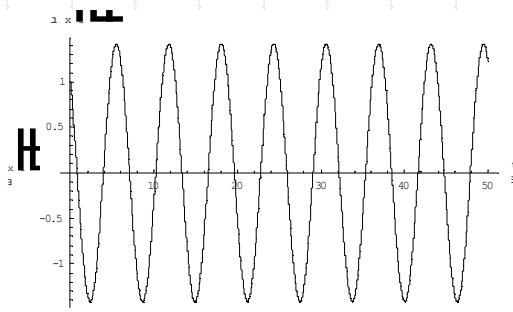
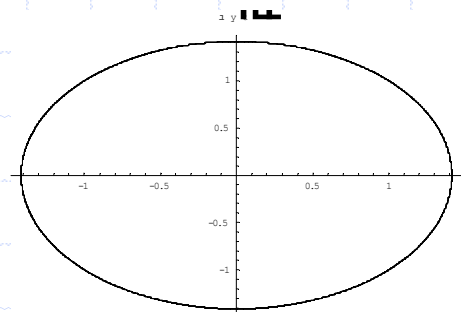
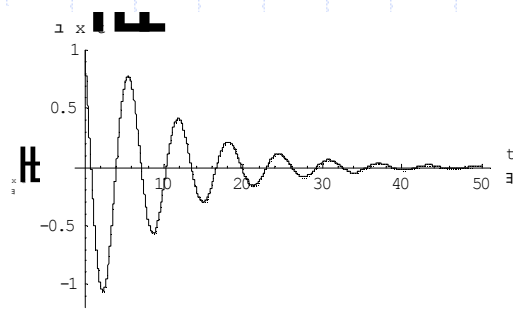
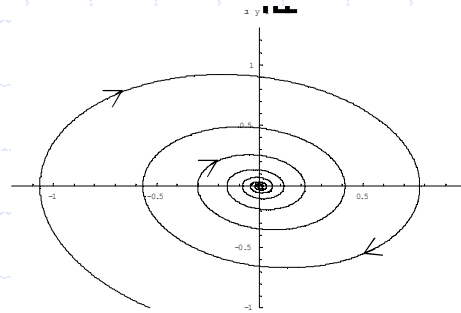
連立微分方程式の解は

$$\begin{cases} \frac{dx}{dt} = -y \\ \frac{dy}{dt} = x - ay \end{cases}$$

安定(平衡点): 一点に収束する解軌道

安定限界(周期軌道): ループ上になる解軌道を周期軌道

不安定(不安定平衡点): 一点より発散する解軌道

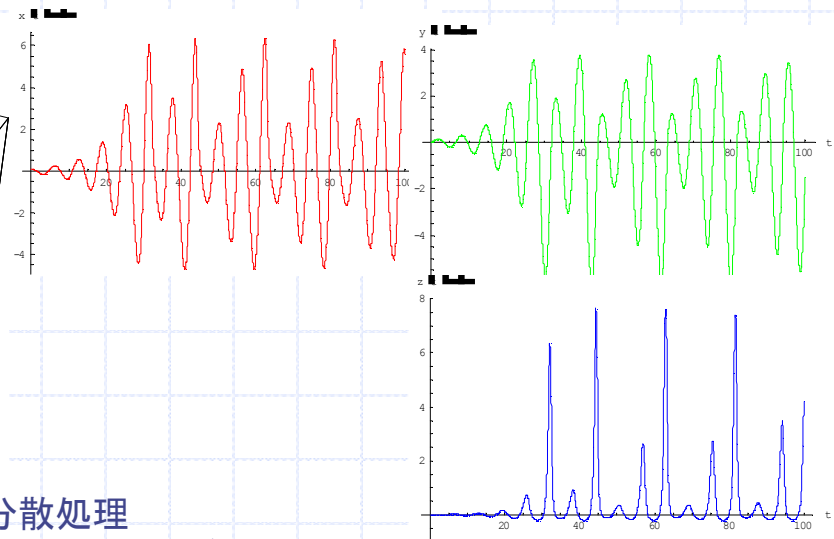
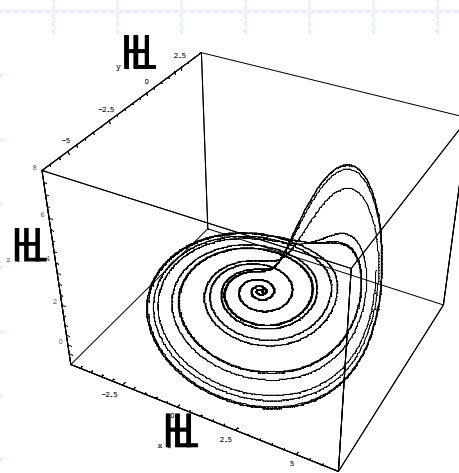


4. 非線形とカオス理論

- ◆ 連立微分方程式が非線形の項を持つと、理論的に解けない上にカオス性を持つ場合がある。
- ◆ カオスとは規則に従って発生しているにも関わらず、不規則にみえる振る舞いを示す現象の事である。
- ◆ カオス特性を持つ時系列データを特にカオス時系列データと言う。

レスラーモデル

$$\begin{cases} \frac{dx}{dt} = -y - z \\ \frac{dy}{dt} = x + ay \\ \frac{dz}{dt} = bx - (c - x)z \end{cases}$$



2010/8/2

カオスとニューロ、分散処理
ー現実世界の複雑性とコンピューティングー

4.1カオス(非線形)システムを解く

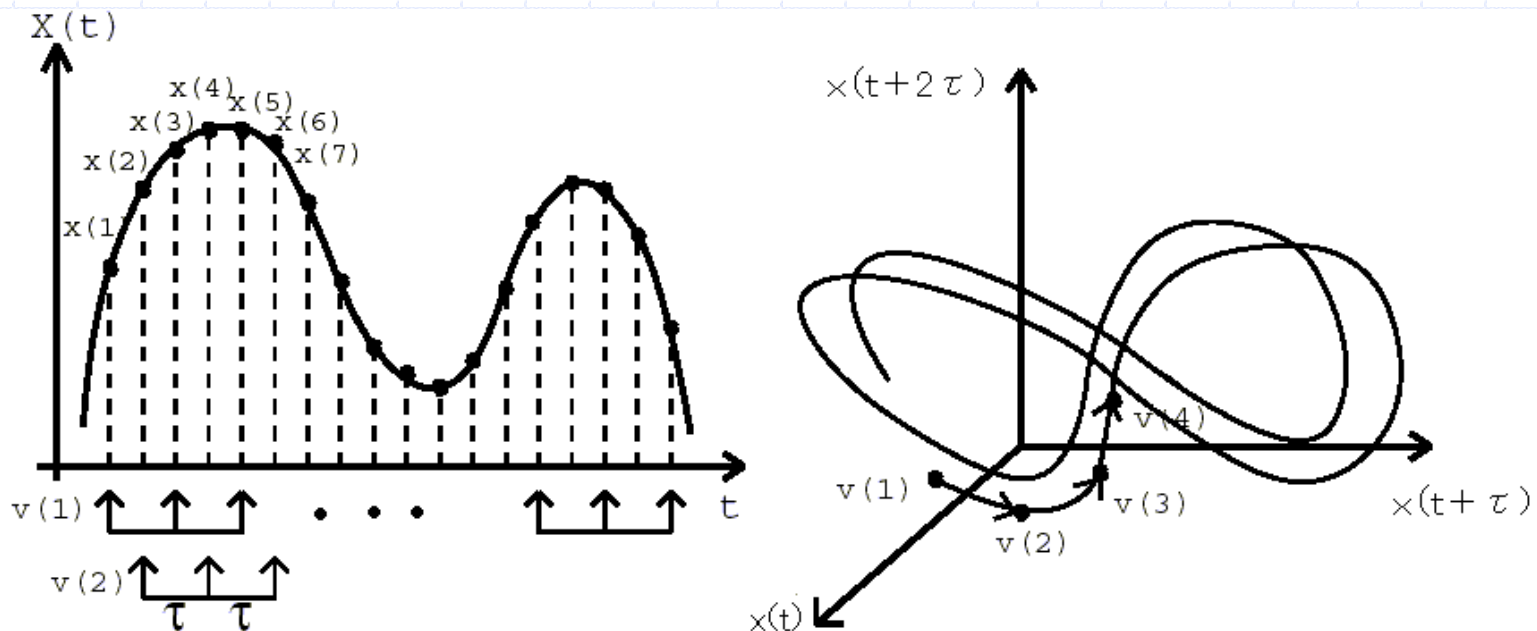
- ◆ 非線形は大部分の問題が解けない(難しい)。
- ◆ 制御も難しい(機械であれば安全かどうかいえない)。
- ◆ できるだけ全要素を観測しないと正確に解析、同定、予測できない。
- ◆ カオス現象(複雑な動き)が生じやすい。
⇒一般的な解析的な方法では、対処が難しい。

4.2カオスと短期予測

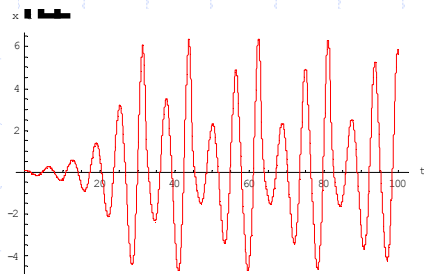
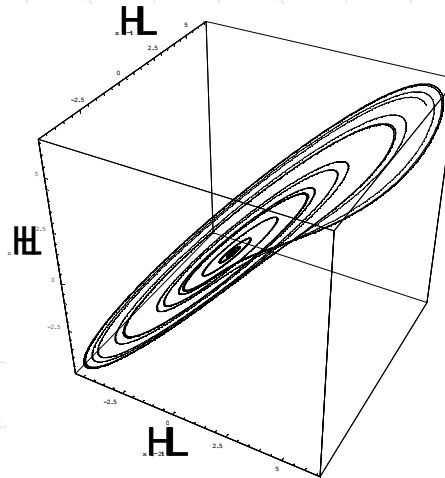
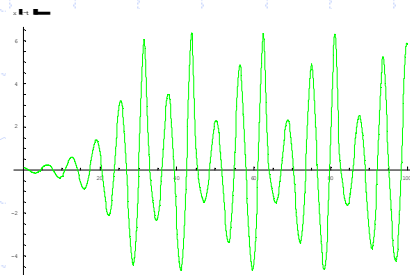
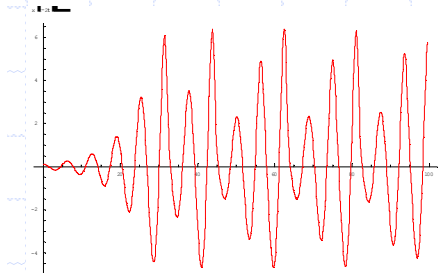
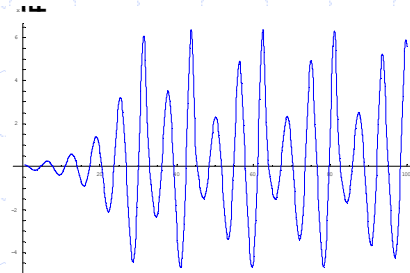
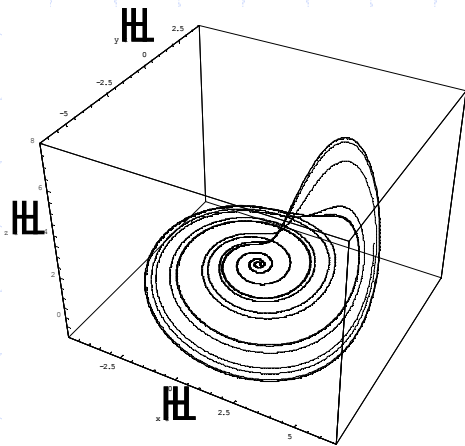
- ◆ 時系列データがカオス性を持つ場合、そのデータは決定論的な法則に従っている事が考えられる。
- ◆ もし、その非線形的な決定論的規則性を推定する事が出来れば、規則性を失うまでの近未来なら予測する事が可能である。
- ◆ ただし、カオスの特徴の一つである鋭敏な初期値依存性により、因果性を失うまでの短期予測しか出来ない。

再構成 (takensの埋め込み定理)

- ◆ また、アトラクタの再構成を利用して一部のデータから他のデータを推定する事が出来る。



再構成(レスラーモデル)



1つのデータしか得られない

1つのデータを遅れ時間 $\tau, 2\tau, 3\tau$ ずらす

ある程度以上の次元に再構成すれば元のと同じ特性を持つ

5. ソフトコンピューティング

- ◆ ハードコンピューティング → 厳密な計算 (計算時間やコストが増大する傾向がある)
- ◆ ソフトコンピューティング → 過度の厳密性を避ける事により実用性を向上させる。

代表的な理論:

ファジィ理論、ニューロコンピューティング、
遺伝的アルゴリズム、カオス理論、etc

ソフトコンピューティングに属する理論

◆ ファジィ理論

人間の言語のような曖昧さを取り扱う事ができる(柔らかい制御などが得意)。

◆ ニューロコンピューティング

神経回路網の応用、学習や予測を行うことができる。理論的には何でも学習ができる

◆ 遺伝的アルゴリズム(GA: Genetic Algorithms)

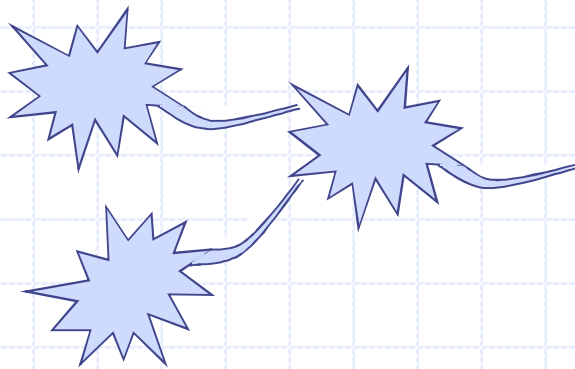
生物進化のメカニズムを利用、解の発見などに有効。

◆ カオス理論、etc

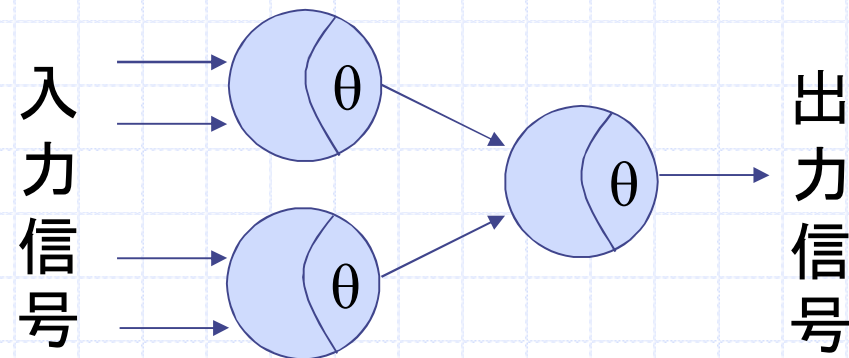
自然界にみられる複雑な現象を取り扱う。非線形理論。

5.1 神経回路網とニューラルネット

- ◆ ニューラルネットワークとは
人間の脳を構成する神経回路網をニューロンとそのつながりであるニューラルネットワークとしてコンピュータ上に再現し、問題に適用しようとする新しいアルゴリズムである。



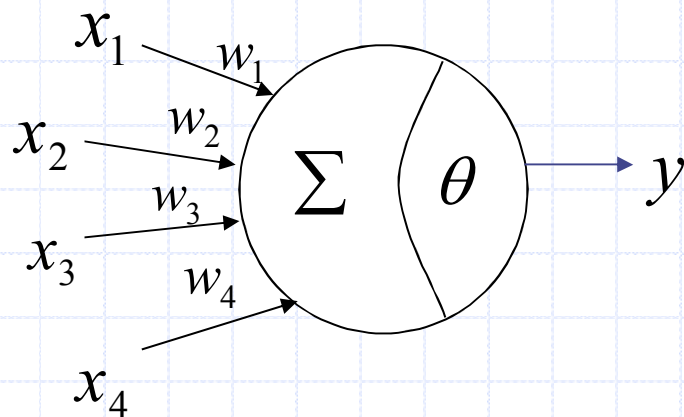
実際の神経回路網
(生物の脳や神経)



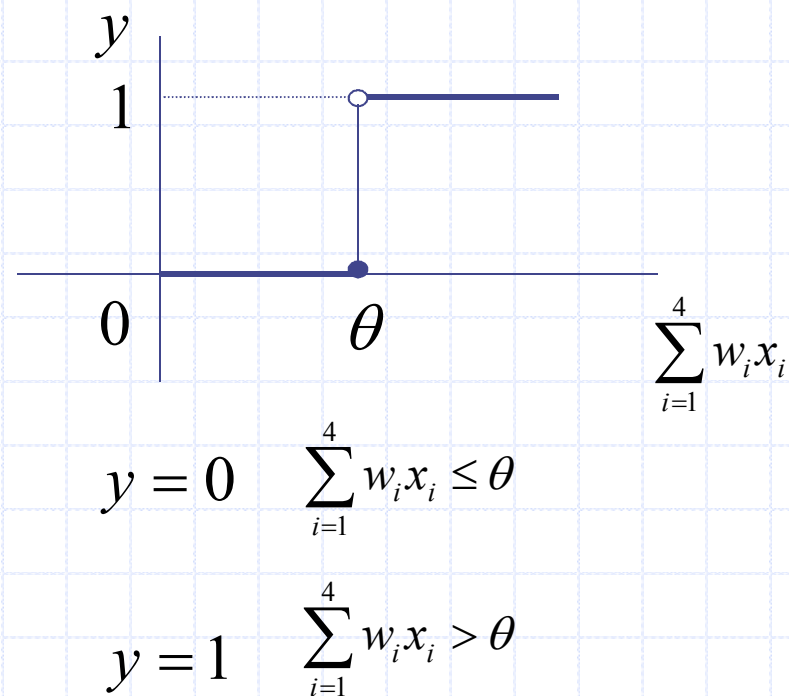
ニューラルネットワーク(コン
ピュータ上に神経網を再現)

ニューロンモデルの基本動作

- ◆ ニューロンは基本的に多入力多出力(ここでは簡単のため4入力1出力かつ閾値モデルの場合を考える)

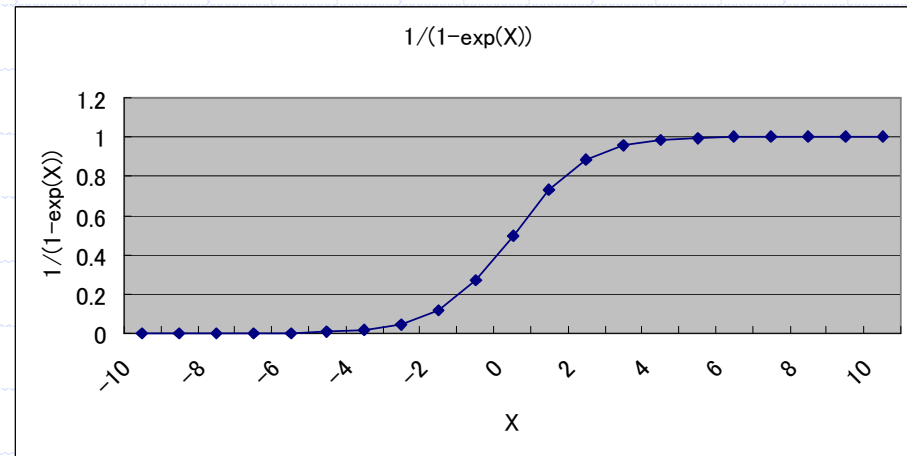
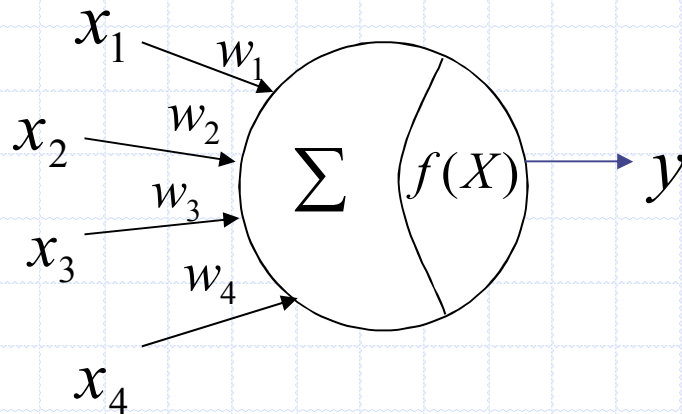


注:実際の生物ニューロンは生体パルスの頻度で発火し、閾値も個々に異なる。



ニューロンモデル

他にもいろいろなニューロンのモデル(応答関数、出力関数の型)がある。以下はシグモイド関数



$$X = \sum_{i=1}^n w_i x_i \quad \text{として}$$
$$y = f(X) = \frac{1}{1 + \exp(-X)}$$

学習は、教師信号と出力との誤差から結合係数を修正して行われる。

(一般デルタ則、BP法)

ニューラルネットワークの特徴

◆ 長所

与えられた数値データから学習や認識を行う。非線形関係や対象の構造が未知でも学習できる可能性を持つ。
(応用例: パターン認識、音声認識や文字認識、家電製品等の制御など)

◆ 短所

学習や認識の過程、または学習した結果の構造の解析が困難である。

分類について

代表的なニューロンのタイプ

- 閾値関数、線形関数、シグモイド関数、etc

代表的なニューラルネットワークのタイプ

- パーセプトロン、ポップフィールドネットワーク、ボルツマンマシンなど。
- ネットワークトポロジー(形態)による分類も注意

代表的な学習則

- 教師なしと教師あり学習、標準デルタ則と一般デルタ則、学習の方向(前向き後向き)など。

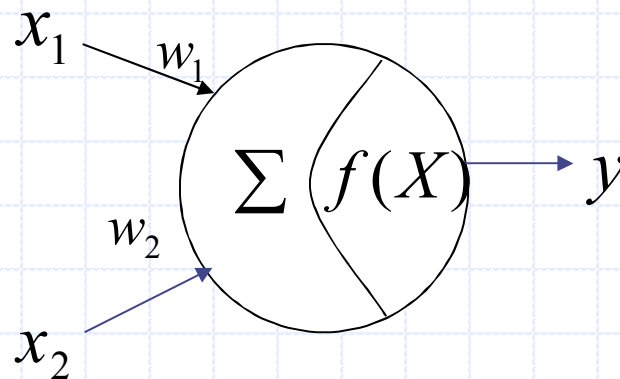
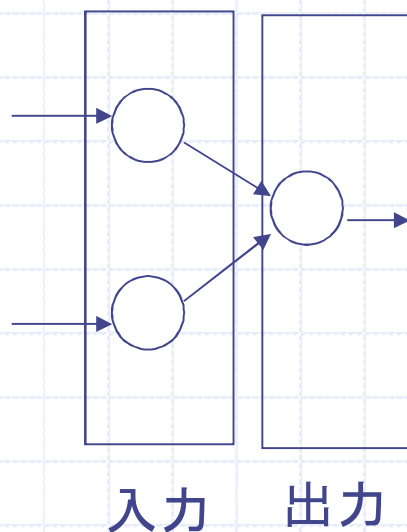
特に教師ありの一般デルタ則やフィードバック型のバックプロパゲーション法(BP法)が有名

カオスとニューロ、分散処理

ー現実世界の複雑性とコンピューティングー

パーセプトロン

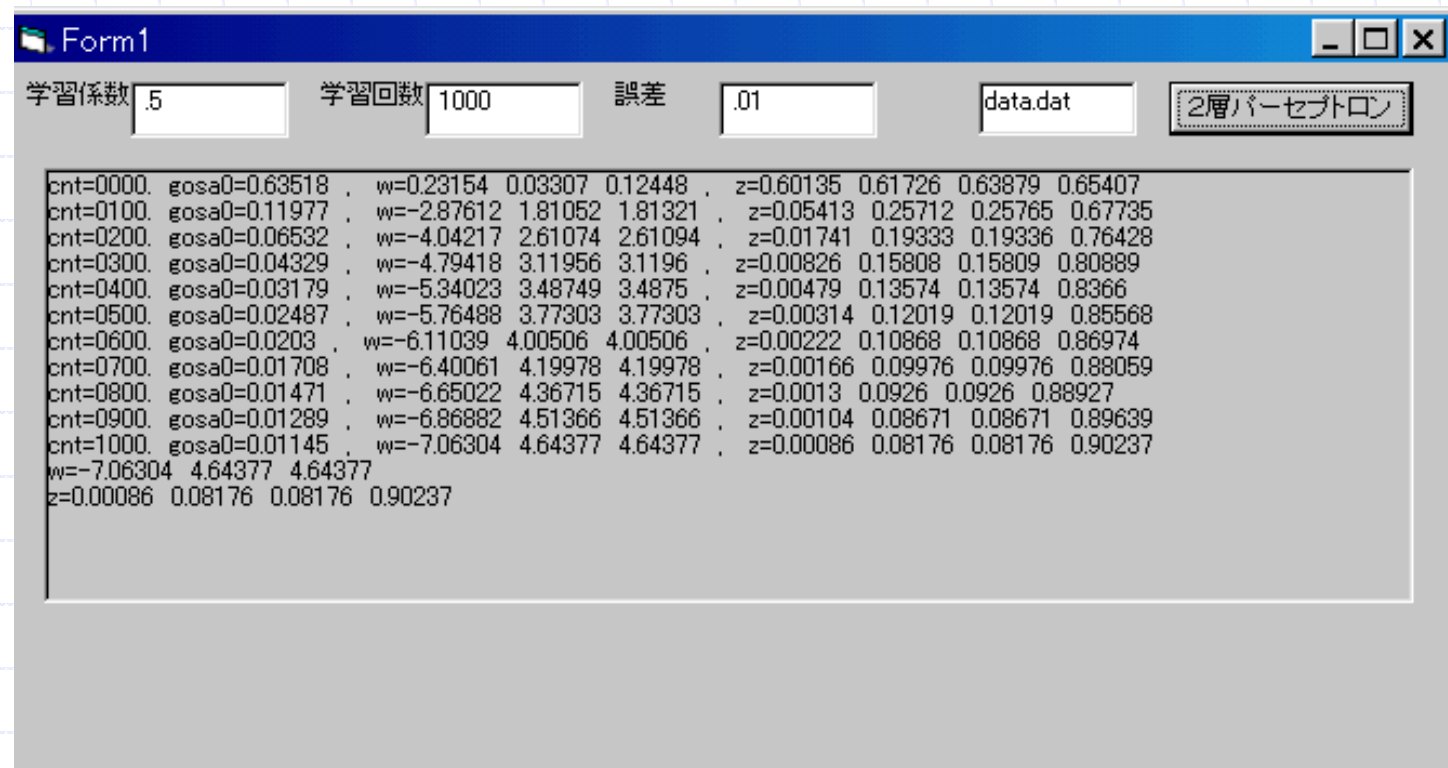
- ◆ 最初に提案されたニューラルネットワークのモデル
- ◆ 基本は2層のニューラルネットワーク(本来は3層)
例:2入力のANDを学習するには以下のような2層のニューラルネットワークを考えれば良い



x1	x2	Y
0	0	0
1	0	0
0	1	0
1	1	1

パーセプトロン(ANDの学習)

◆ 結果の例

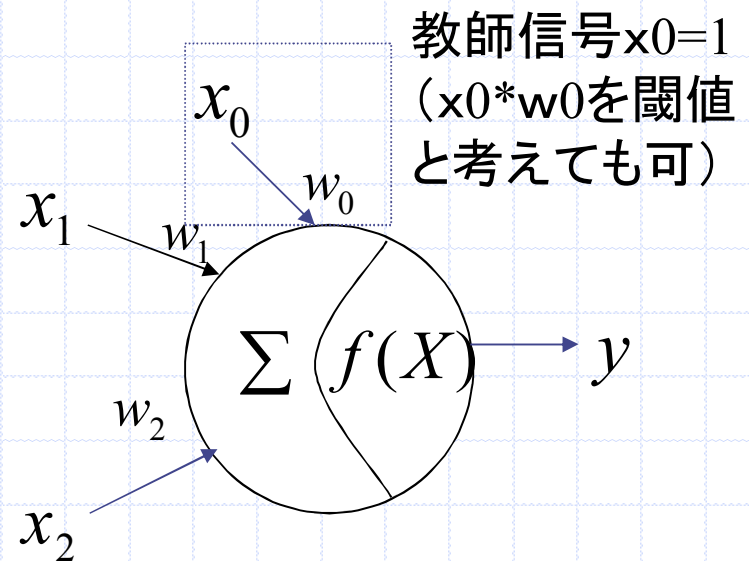
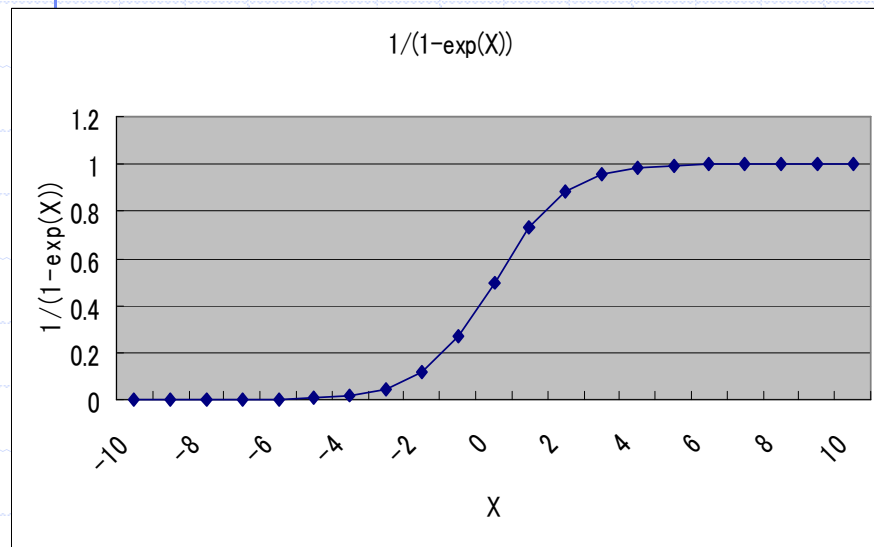


```
cnt=0000. gosa0=0.63518 . w=0.23154 0.03307 0.12448 . z=0.60135 0.61726 0.63879 0.65407
cnt=0100. gosa0=0.11977 . w=-2.87612 1.81052 1.81321 . z=0.05413 0.25712 0.25765 0.67735
cnt=0200. gosa0=0.06532 . w=-4.04217 2.61074 2.61094 . z=0.01741 0.19333 0.19336 0.76428
cnt=0300. gosa0=0.04329 . w=-4.79418 3.11956 3.1196 . z=0.00826 0.15808 0.15809 0.80889
cnt=0400. gosa0=0.03179 . w=-5.34023 3.48749 3.4875 . z=0.00479 0.13574 0.13574 0.8366
cnt=0500. gosa0=0.02487 . w=-5.76488 3.77303 3.77303 . z=0.00314 0.12019 0.12019 0.85568
cnt=0600. gosa0=0.0203 . w=-6.11039 4.00506 4.00506 . z=0.00222 0.10868 0.10868 0.86974
cnt=0700. gosa0=0.01708 . w=-6.40061 4.19978 4.19978 . z=0.00166 0.09976 0.09976 0.88059
cnt=0800. gosa0=0.01471 . w=-6.65022 4.36715 4.36715 . z=0.0013 0.0926 0.0926 0.88927
cnt=0900. gosa0=0.01289 . w=-6.86882 4.51366 4.51366 . z=0.00104 0.08671 0.08671 0.89639
cnt=1000. gosa0=0.01145 . w=-7.06304 4.64377 4.64377 . z=0.00086 0.08176 0.08176 0.90237
w=-7.06304 4.64377 4.64377
z=0.00086 0.08176 0.08176 0.90237
```

パーセプトロン(結果の検討)

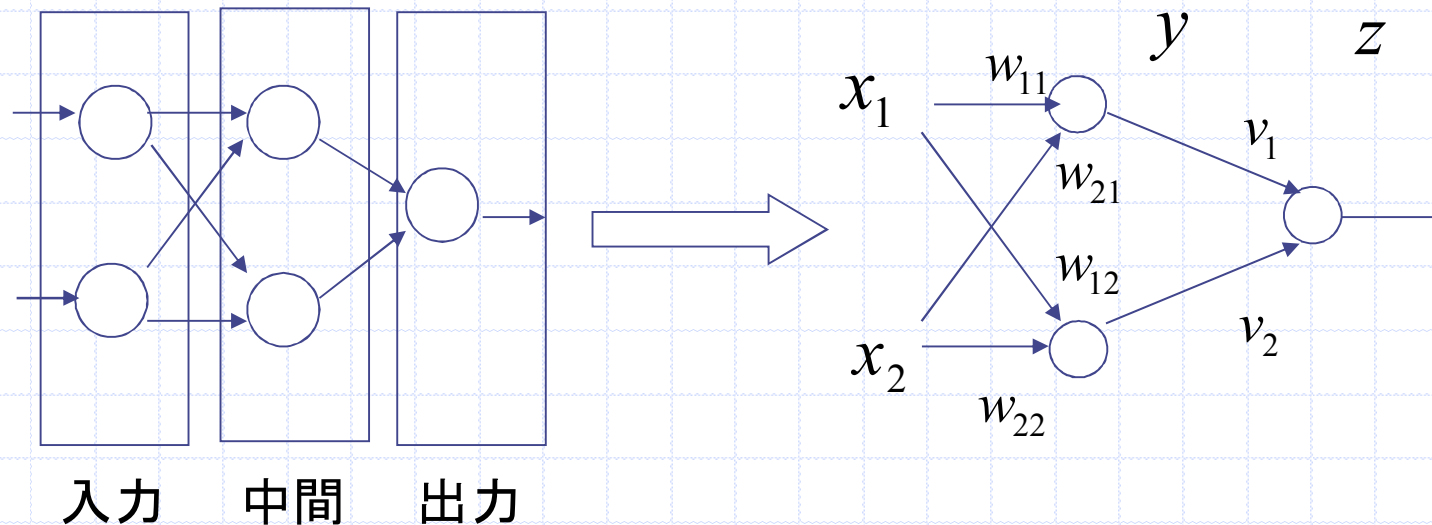
$W_0 = -7.06304$ 、 $W_1 = 4.64377$ 、 $W_2 = 4.64377$

	-7.06304	4.64377	4.64377	$x_0 * w_0$	$x_1 * w_1$	$x_2 * w_2$	$X = \text{sum}(x)$	$\exp(-X)$	$1 + \exp(-X)$	$1 / (1 + \exp(-X))$
pattern1	1	0	0	-7.06304	0	0	-7.06304	1167.99	1168.99	0.000855439
pattern2	1	1	0	-7.06304	4.64377	0	-2.41927	11.23765	12.23765	0.081715016
pattern3	1	0	1	-7.06304	0	4.64377	-2.41927	11.23765	12.23765	0.081715016
pattern4	1	1	1	-7.06304	4.64377	4.64377	2.2245	0.108121	1.108121	0.902428146



3層ニューラルネットワーク

パーセプトロンは簡単なルールしか学習できない。入力層と中間層、出力層の3階層構造のネットワークを使うとより複雑な学習ができるようになる。



3層ニューロ (EXORの学習)

```
Form1
学習係数 .5   学習回数 10000   誤差 .01   dataexor.dat   3層パーセptron

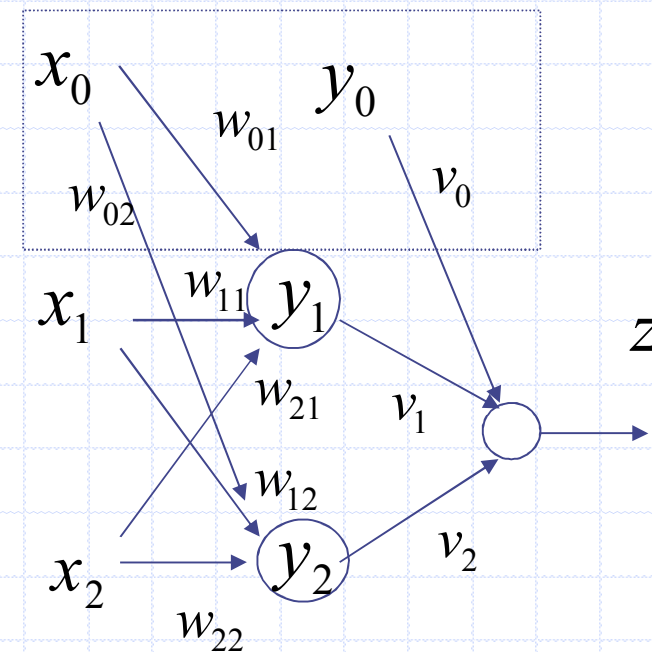
cnt=0000. gosa0=0.50156
u=-0.42043 -0.39428 0.54789 . y=0.39631 0.30641 0.53211 0.43352
u=-0.97247 0.52156 0.62878 . y=0.27449 0.38924 0.41508 0.5445
v=0.42162 -0.90651 -0.17098 . z=0.50263 0.51813 0.46577 0.48257
cnt=1000. gosa0=0.46774
u=-0.28424 -0.57548 -0.48423 . y=0.4295 0.29825 0.31778 0.20821
u=-2.81684 1.48843 1.47059 . y=0.05677 0.20985 0.2069 0.53513
v=0.70351 -0.84351 -1.70631 . z=0.56087 0.52364 0.52079 0.40539
cnt=2000. gosa0=0.01053
u=1.7167 -4.86196 -4.86072 . y=0.84765 0.04129 0.04134 0.00033
u=-6.54516 4.18368 4.18313 . y=0.00144 0.08618 0.08614 0.86073
v=3.39311 -7.11279 -6.90458 . z=0.06627 0.92441 0.9244 0.07232
cnt=3000. gosa0=0.0039
u=1.95287 -5.29384 -5.29304 . y=0.87574 0.0342 0.03422 0.00018
u=-7.05523 4.52652 4.5262 . y=0.00086 0.07388 0.07386 0.88052
v=3.89018 -8.05241 -7.91615 . z=0.04038 0.9539 0.9539 0.04388
u=1.952866 -5.293839 -5.293039
u=-7.055225 4.526517 4.526201
v=3.890177 -8.052414 -7.916148
z=0.04038 0.9539 0.9539 0.04388
```

3層ニューラルネット(結果の検討2)

正解	x1	x2	z
	0	0	0
	1	0	1
	0	1	1
	1	1	0

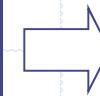
ニューラルネットの解答	x1	x2	y1	Y2	z
	0	0	0.87575	0.00086	0.04036
	1	0	0.03419	0.07386	0.95391
	0	1	0.03421	0.07384	0.95391
	1	1	0.00017	0.88053	0.04386

教師信号

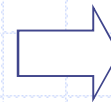
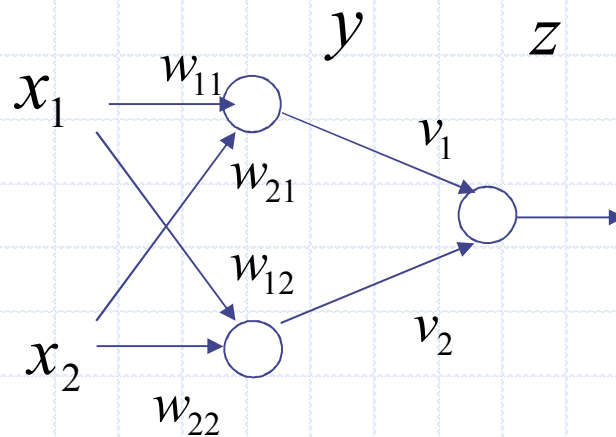


3層ニューラルネット(結果の検討3)

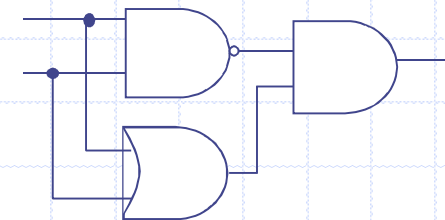
x1	X2	y1	Y2	z
0	0	0.8757588	0.0008621	0.0403638
1	0	0.0341920	0.0738679	0.9539155
0	1	0.0342184	0.0738463	0.9539137
1	1	0.0001779	0.8805304	0.0438699



x1	X2	y1	Y2	z
0	0	1	0	0
1	0	0	0	1
0	1	0	0	1
1	1	0	1	0

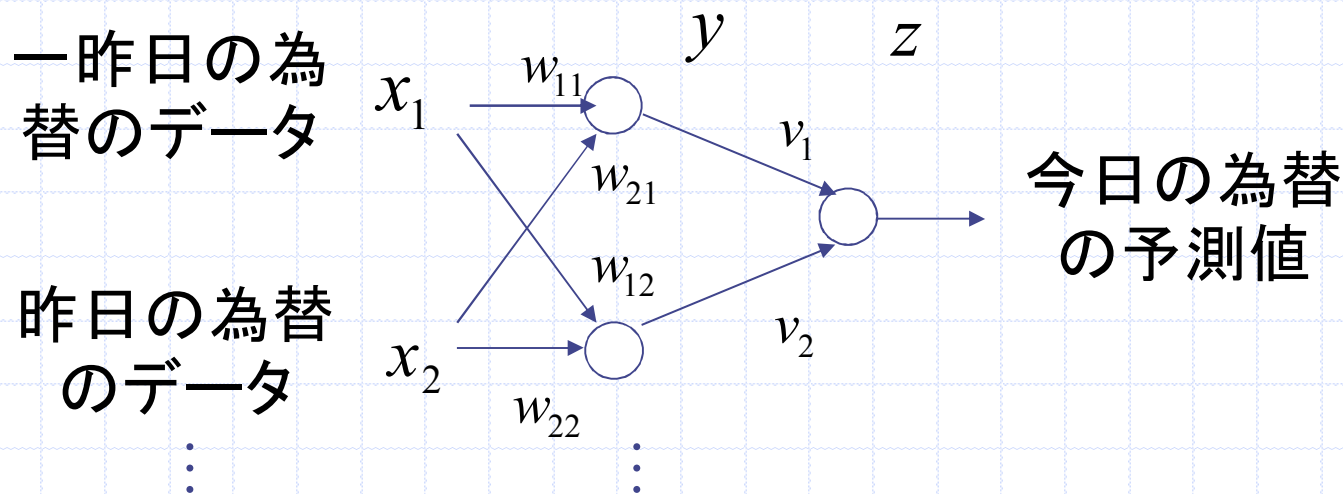


電子回路に直すと



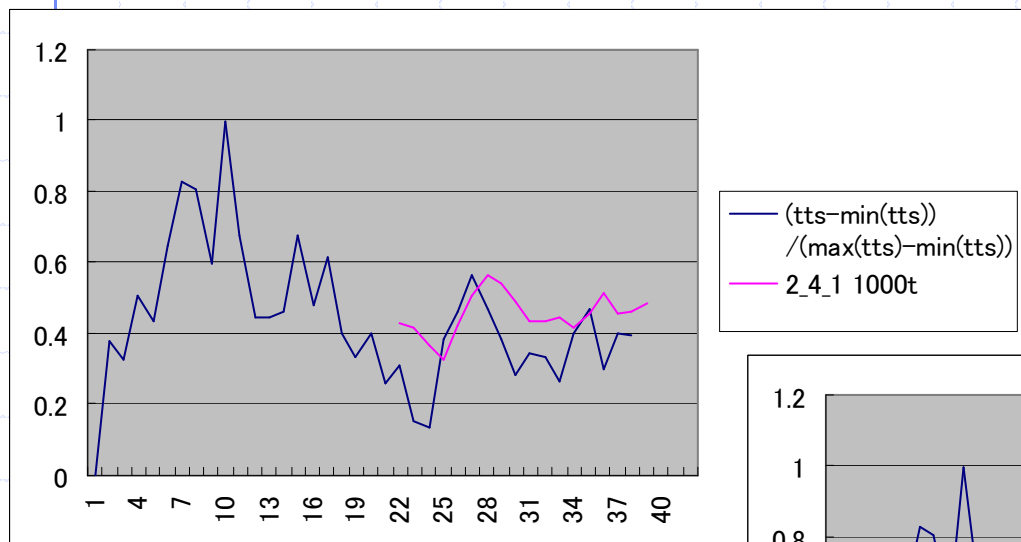
時系列データに対する適用

一般的な3層ニューロとバックプロパゲーションだけで時系列データの予測が可能



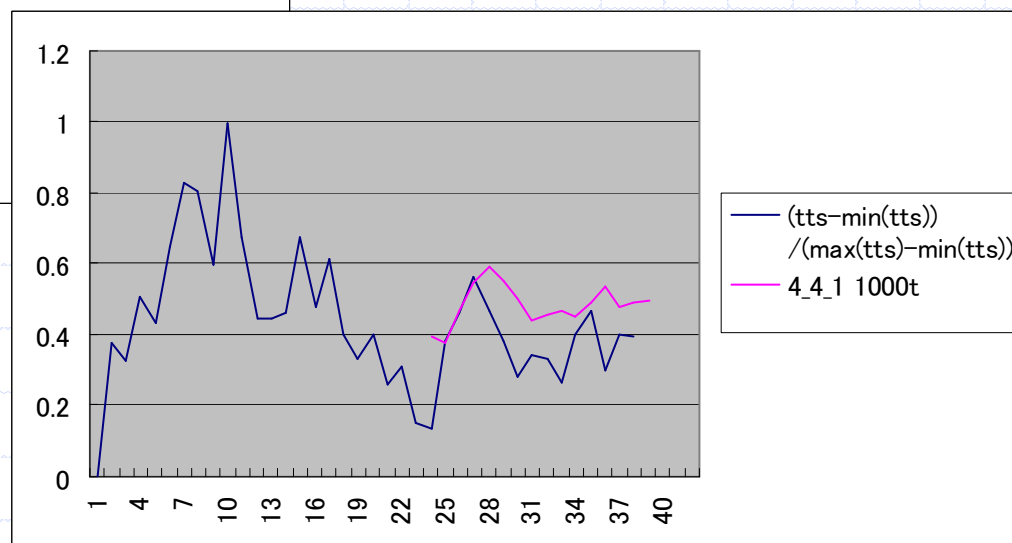
時系列データに対する適用例

入力層や中間層の数によって予測が変化



← 入力2、中間層4、
出力1、学習1000回

入力4、中間層4、
出力1、学習1000回 →



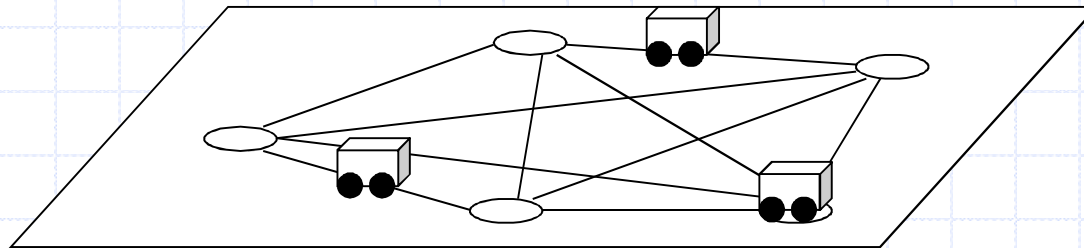
ニューラルネットワークのまとめ

- ◆ 神経回路網をニューラルネットワークとしてコンピュータ上に再現し、問題に適用する新しいアルゴリズム。
- ◆ 与えられた数値データから学習や認識を行う。
- ◆ 非線形関係や対象の構造が未知でも学習できる可能性を持つ。
- ◆ 学習や認識の過程、または学習した結果の構造の解析が困難である。
- ◆ 入力層や中間層の数によって(つまりはニューラルネットの構造)によって性能が変化

5.2 遺伝的アルゴリズム (GA: Genetic Algorithms)

- ◆ 遺伝と自然淘汰という生物進化のメカニズムを利用、コンピュータ上で擬似生命を繁殖させて答えを探す。

遺伝的アルゴリズムのイメージ

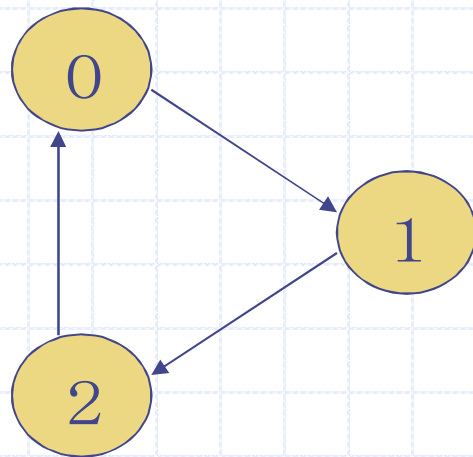


GAの例題(巡回セールスマン)

- ◆ いくつかの都市を最短で効率良く回りたい。都市0から出発して戻るには？(3都市の場合)

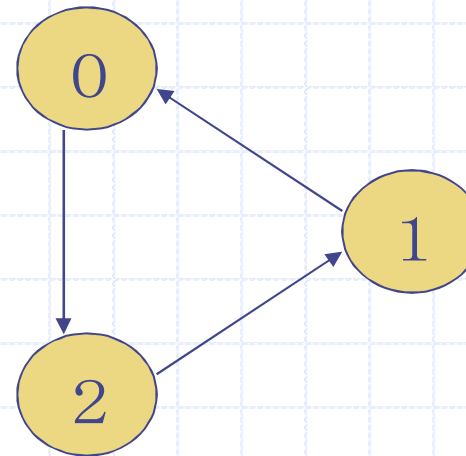
巡回ルート1

順序(0→1→2→0)



巡回ルート2

順序(0→2→1→0)



GAの例題の続き

- ◆ もし、10都市、20都市で実際の走行距離を考えて計算しなければならないなら？

(スタートする都市が固定されているとして)

5都市の場合、 $4 \times 3 \times 2 \times 1 = 24$ 通り

6都市の場合、 $5 \times 4 \times 3 \times 2 \times 1 = 120$ 通り

7都市の場合、 $6 \times 5 \times 4 \times 3 \times 2 \times 1 = 720$ 通り

10都市の場合、 $9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 362880$ (36万通り)

遺伝的アルゴリズムによる探索(その 1)

- ◆ 生物は一般に遺伝によって、親から子に子から孫に能力を継承する。また、優れたものは生き残りやすいという自然淘汰という原理も働く。

「運動能力の高い遺伝子を持つねずみは、すばしっこいので生き延びて子孫を残す」

「運動能力の低い遺伝子を持つねずみは、捕まって子孫を残せない」



「だんだん、ねずみはすばしっこくなる」

(運動能力の高い遺伝子を持つねずみが増える)

遺伝的アルゴリズムによる探索(その2)

- ◆ 「運動能力が高い」という部分を「短い走行距離で巡回する」におきかえてコンピュータ上に再現し生活させてやれば、生物が環境に適応してきたように、いつかは

「短い距離で巡回できる人工的なねずみが繁殖する。」

プログラムは人間よりは知的ではないが、力技で永久に計算しているよりはまし。

12都市計算例

TSP

```

route map=
  0, 12, 63, 11.1, 92, 6, 7, 36, 162, 10,
  12, 0, 122, 156, 42, 6, 144, 94, 12, 22,
  63, 122, 0, 5, 8, 72, 22, 41, 11.1, 10,
  11.1, 156, 5, 0, 11.4, 11.7, 44, 9, 10, 136,
  92, 42, 8, 11.4, 0, 36, 102, 6, 94, 22,
  6, 6, 72, 11.7, 36, 0, 92, 36, 125, 4,
  7, 144, 22, 44, 102, 92, 0, 58, 126, 122,
  36, 94, 41, 9, 6, 36, 58, 0, 127, 72,
  162, 12, 11.1, 10, 94, 125, 126, 127, 0, 111,
  10, 22, 10, 136, 22, 4, 122, 72, 11.1, 0,
  82, 5, 72, 108, 1, 28, 94, 5, 98, 28,
  144, 9, 101, 104, 6.7, 10, 12, 108, 3, 82,
  
```

Minimum distance= 50.3
 Minimum running route=
 0 - 2 - 6 - 3 - 8 - 11 - 1 - 9 - 4 - 10 - 5 - 7 - 0
 Calculation time= 39.71875

data.dat データファイル

図 厳密な解

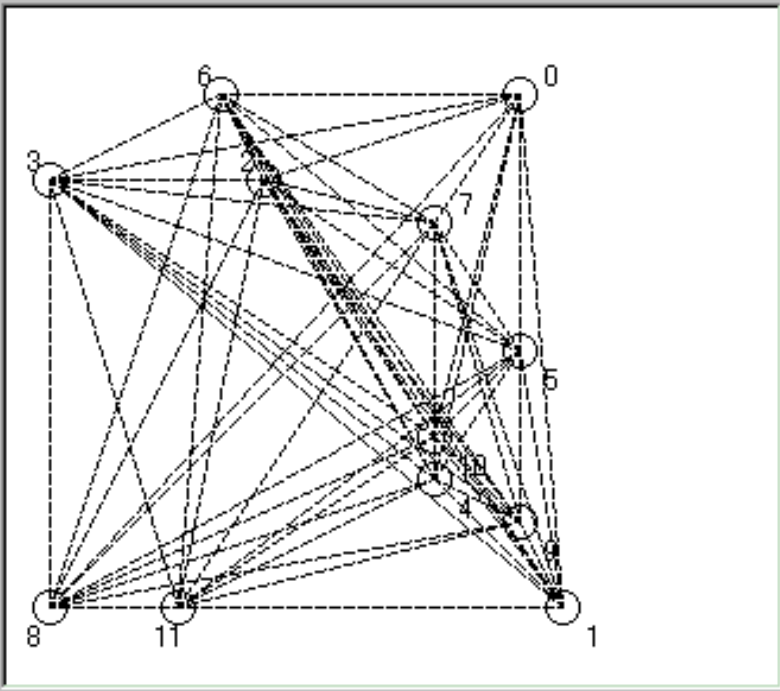
make route map

都市数
12

data.dat

データファイル

データ作成



```

generation= 1, distance= 66.1
generation= 2, distance= 65.4
generation= 3, distance= 65.4
generation= 4, distance= 65.4
generation= 5, distance= 63.8
generation= 6, distance= 63.7
generation= 7, distance= 58.3
generation= 8, distance= 58.3
generation= 9, distance= 58.3
generation= 10, distance= 58.3

10 thMinimum distance= 58.3
Minimum running route=
  0 - 5 - 7 - 6 - 2 - 10 - 9 - 1 - 11 - 8 - 11 - 4 - 0
Calculation time= 26.78125
  
```

data.dat データファイル

経路の探索

図 GAの解

GAのまとめ

- ◆ 遺伝と自然淘汰という生物進化のメカニズムを利用、コンピュータ上で擬似生命を繁殖させて答えを探す。
- ◆ 生物が環境に適応してきたように、条件に適応した求めたい解がいつかは見つかる。
- ◆ プログラムは人間よりは知的ではないが、力技で永久に計算しているよりはまし。

5.3 GA-ニューラルネットワーク

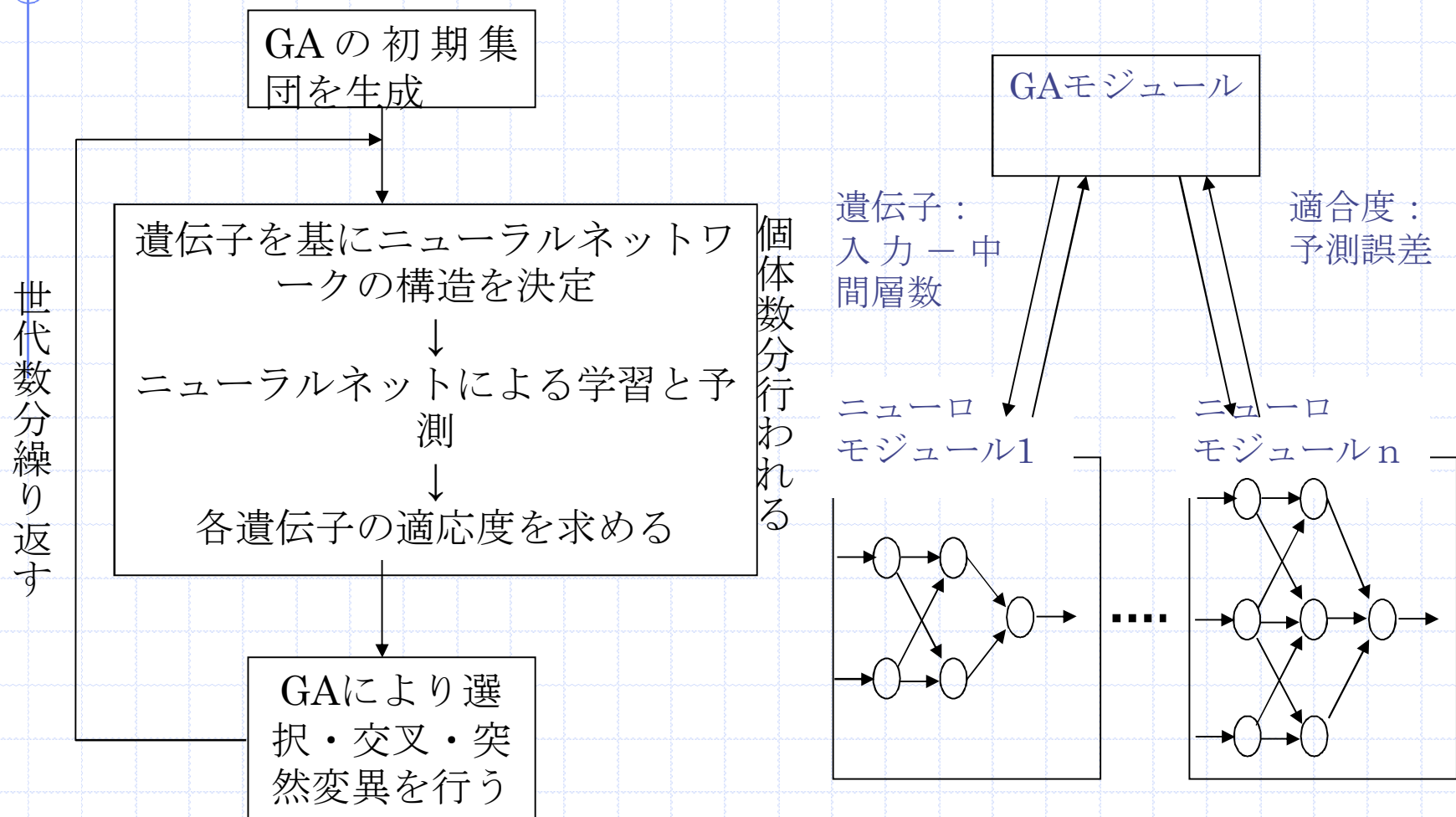
ニューラルネットワーク

- ◆ 神経回路網をニューラルネットワークとしてコンピュータ上に再現し、問題に適用する新しいアルゴリズム。
- ◆ 与えられた数値データから、非線形関係や対象の構造が未知でも学習できる。
- ◆ 入力層や中間層の数によって(つまりはニューラルネットの構造)によって性能が変化

GA

- ◆ 遺伝と自然淘汰という生物進化のメカニズムを利用、コンピュータ上で擬似生命を繁殖させて答えを探す。
 - ◆ 生物の適応のように、条件に適応した解が見つかる。
- ⇒ニューラルネットワークの構造の決定をGAで行なえば？

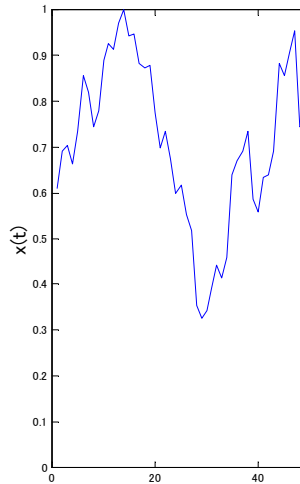
GA-ニューロの処理の流れ



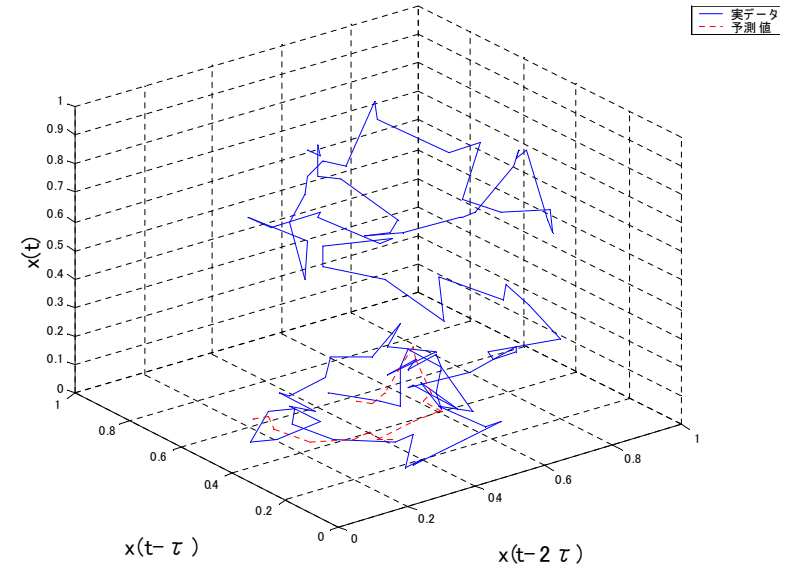
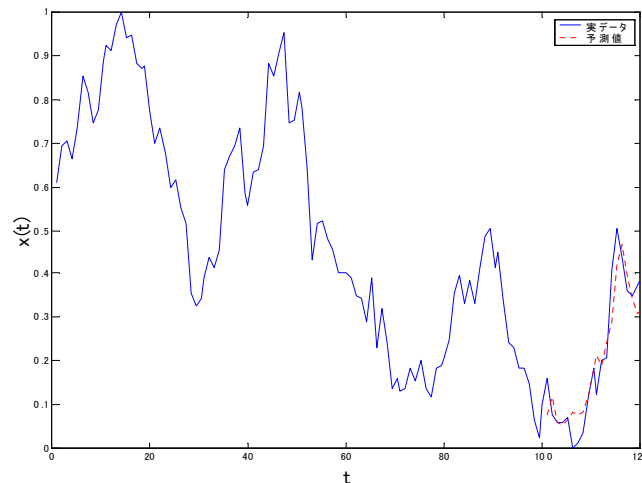
5.3.1 直接予測、5.3.2 再構成空間で予測

◆ 比較のため直接および再構成空間で通常のニューロで予測

データを直接予測

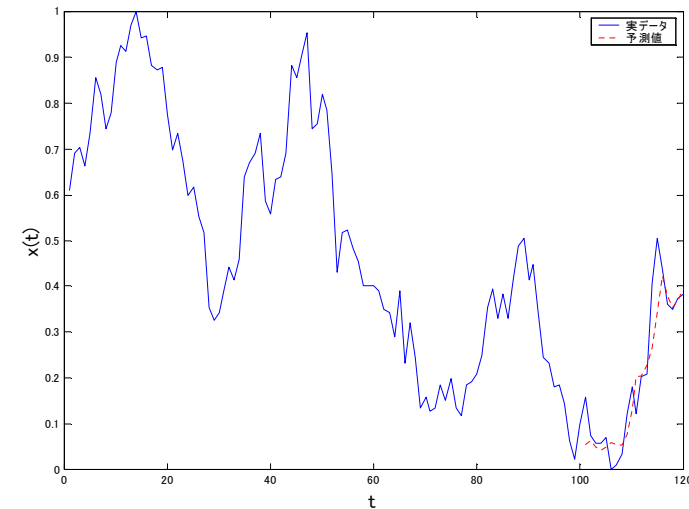
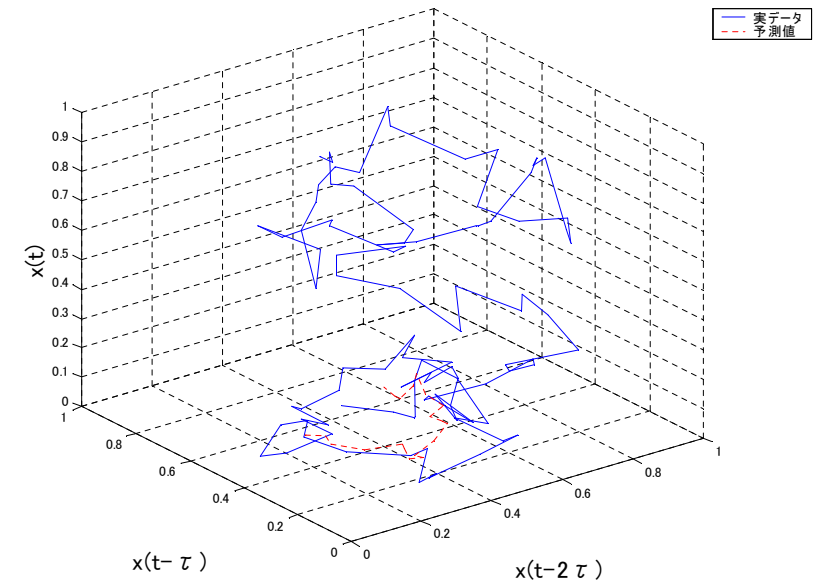


データを再構成して予測



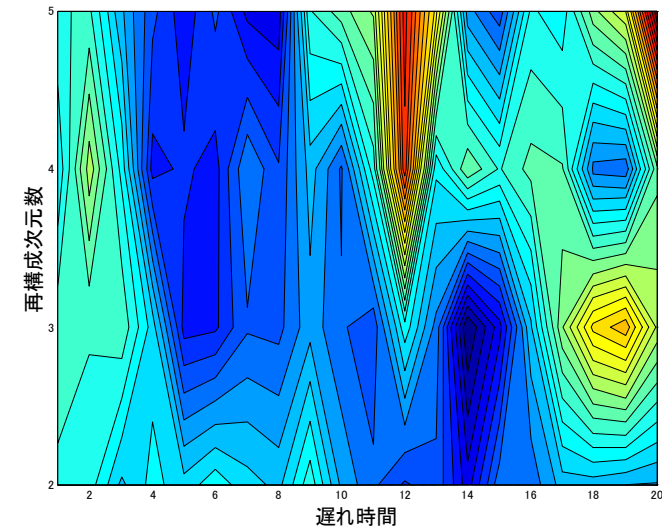
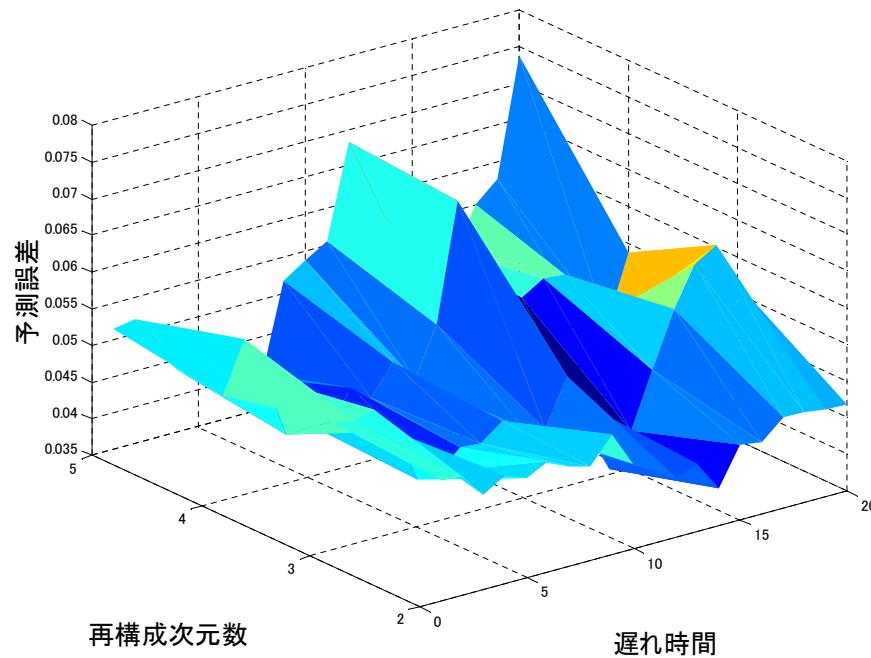
5.3.3 GA-ニューロによる予測

- ◆ 外国為替データをGA-ニューロで予測
- ◆ ニューロの構造はGAで決定される



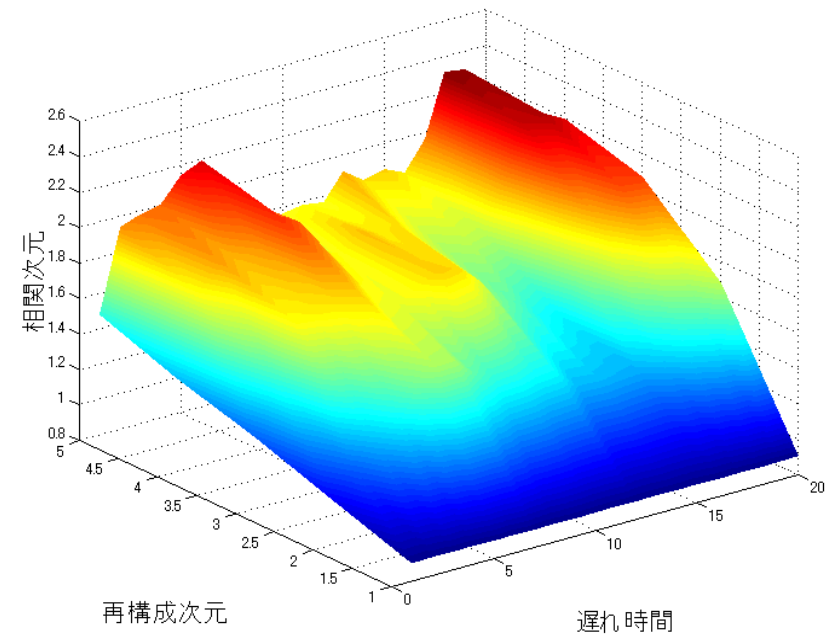
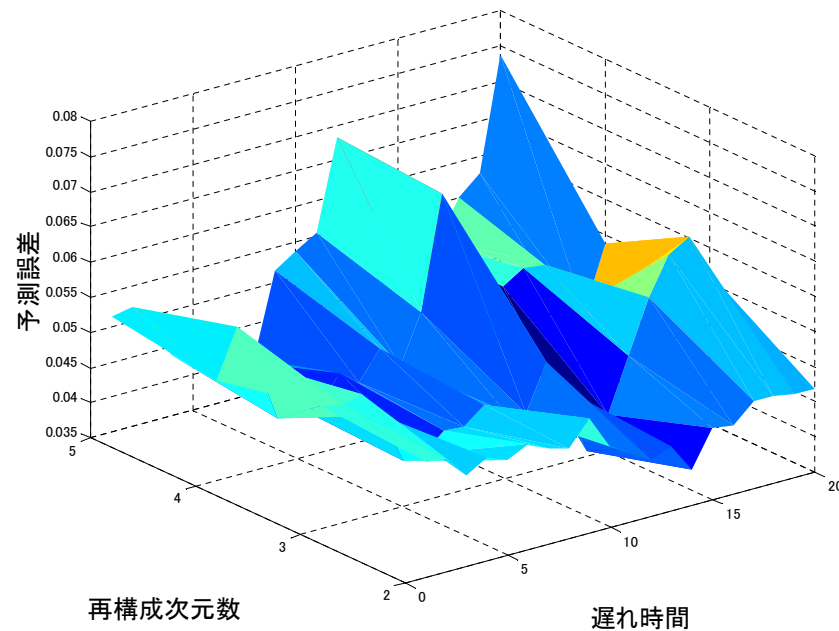
GA-ニューロによる予測(結果の検討)

- ◆ 再構成次元と遅れ時間を変えてGA-ニューロで予測した場合の誤差



GA-ニューロによる予測(結果の検討2)

◆ 再構成次元と遅れ時間とGA-ニューロの予測誤差およびカオス性(相関次元)の関係



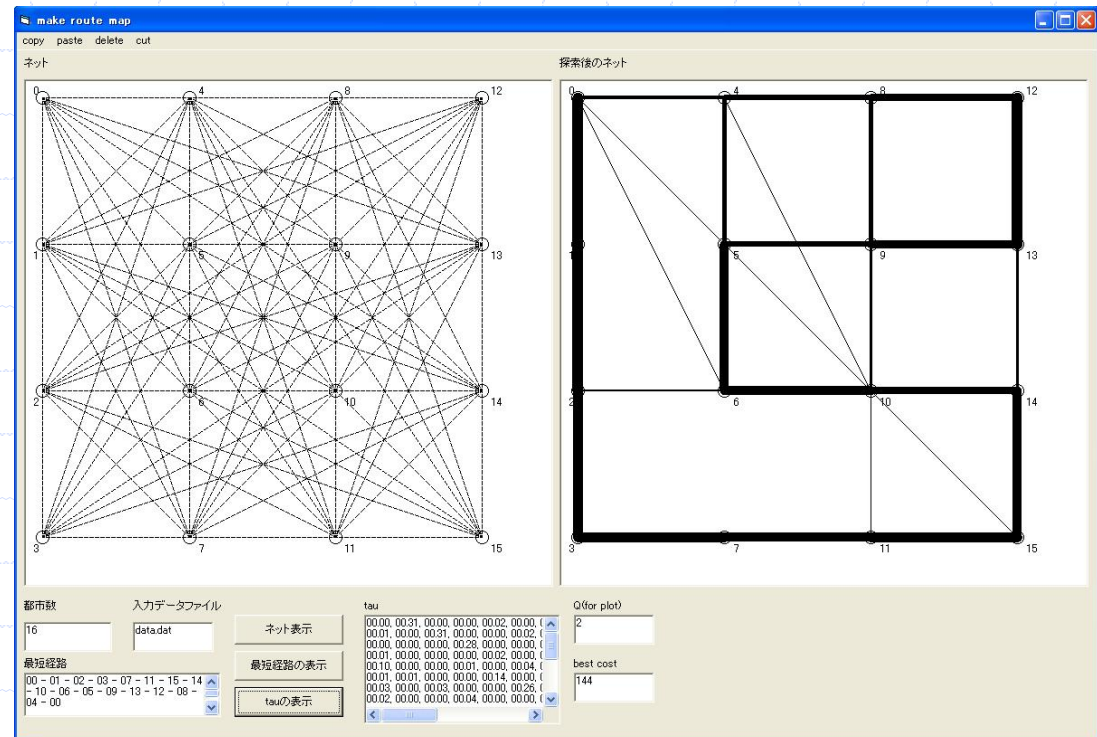
6. 多段システムと分散処理

- ◆ Ant SystemはM.Dorigoらによって提案されたACOメタヒューリスティクスに基づくアルゴリズム
- ◆ Ant Systemにおいてantは組み合わせ最適化問題のような分散的な空間で並列的に行動するエージェントである。
- ◆ フェロモンの揮発性の特徴を利用して最適経路を維持しつつ動的な変化に適応するモデルが構築できる。
- ◆ TSPに対するAnt Systemでは、複数のantを各都市に配置し次の都市までの距離と経路に置かれているフェロモン量に従って、未訪問都市から確率的に選択しながら全ての都市を巡回して経路を生成する。

Ant Systemの例

Ant-Systemのパラメータ

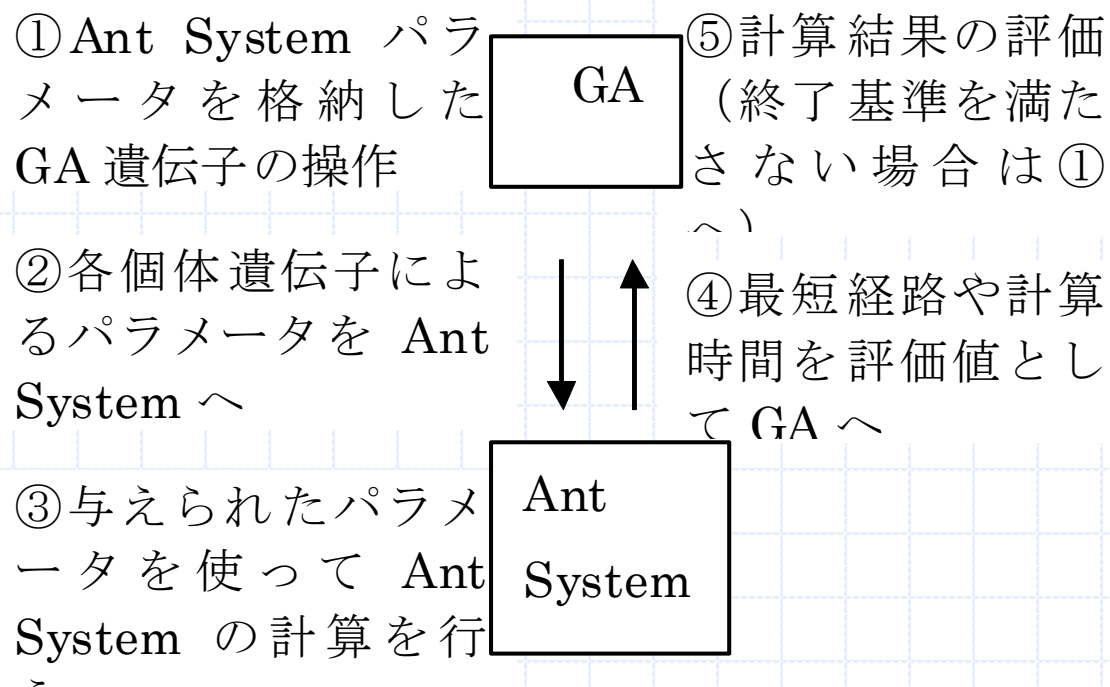
ant数	50
ステップ数	20
ρ (蒸発率)	0.9
Q (フェロモン量)	1.0
α (フェロモンの重要度)	1.0
β (問題依存の重要度)	1.0



16都市の問題に対してAnt-Systemを適用した計算例

GA-Ant System

◆ TSPにおけるAnt Systemのパラメータ設定の難しさを解決するために、Ant SystemのパラメータをGAで探索するGA-Ant Systemを提案し、Ant Systemパラメータを最適化することを考える。



6.1 分散処理

- ◆ グリッドコンピューティングは、分散処理の方式の一つ。
- ◆ であるインターネット等の広域ネットワークを通じて複数のコンピュータ資源を1つの仮想的なコンピュータとして扱う。

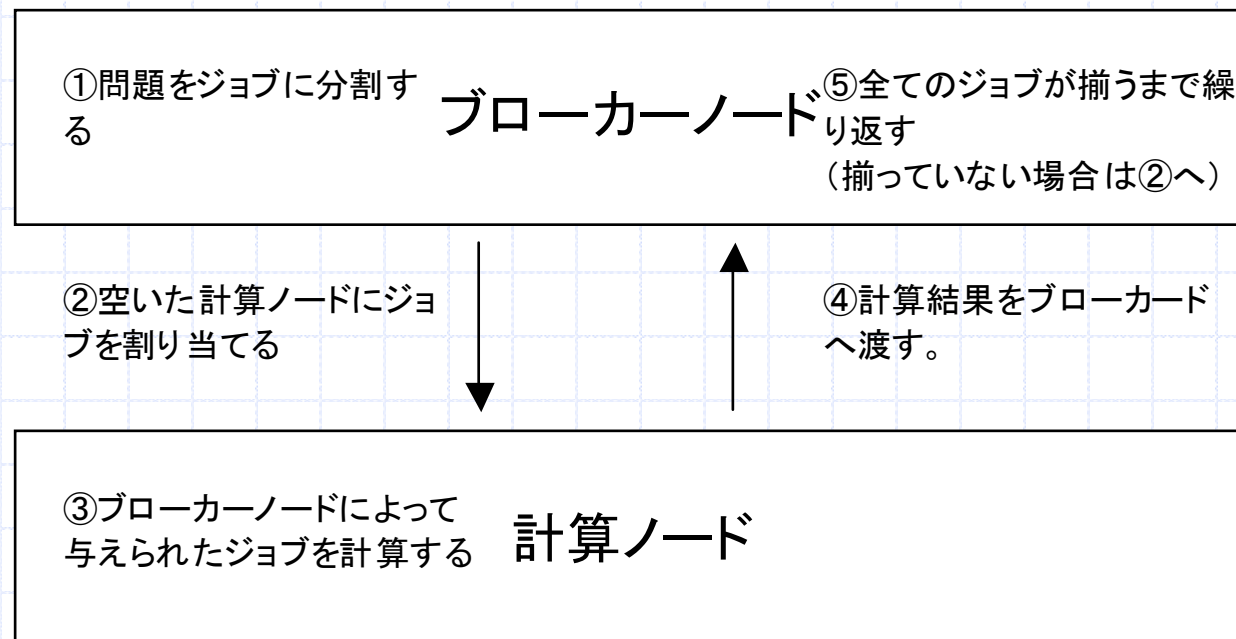


図2-1 グリッドコンピューティングのイメージ

グリッドコンピューティング

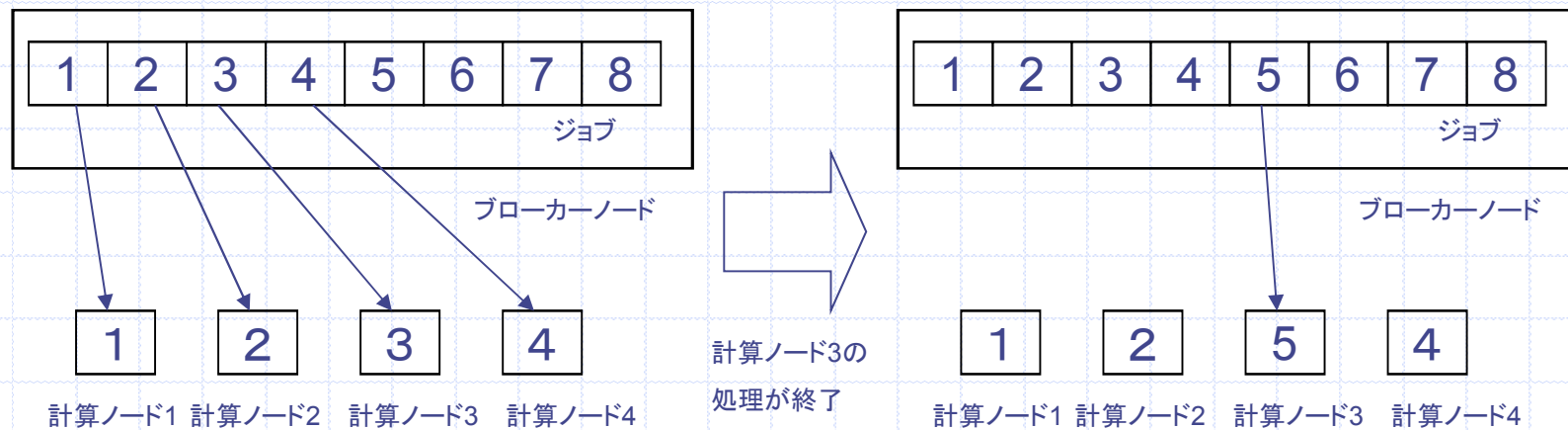


図2-2 グリッドコンピューティングの動作の具体例

TSPの計算例(グリッド100都市)

Table6.1パラメータの探索範囲計10bit)

	ant数	ρ	Q	α	β
数	100~400	0.2~0.8	1~4	1~4	1~4
刻み	100	0.2	1	1	1

Table6.2 GA-ASのパラメータ

GA				AS
遺伝子数	遺伝子長	世代数	突然変異率	ステップ数
20	10	5	0.5	24

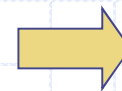
Table6.3処理時間の平均

台数	1台	2台	3台	4台
平均	14127.7	8086.6	5368.1	4558.7
比率	100%	57%	38%	32%

7. コンピューティングと適応

- ◆ 生物は、全体を理解して最良を選んでいるわけではない。実際の生物は、それぞれに限られた情報をもとに個々が良いと思う、自分の判断した方法に従って行動している。
- ◆ 結果的に状況に適した個体が生き延びていく。つまり、個々が多様ゆえに全体として適応する。
- ◆ 変化する環境では今良い事が次もそうである保障は無く、構成要素自体も変化しなくてはならない。そして、構成要素の変化において相互作用は重要な役割を果たす。

多数が相互作用することが重要
多様でなければ適応できない
構成要素は相互作用で変化する



(マルチ)エー
ジェントシステム

8.まとめ

- ◆ 経営情報のような複合的分野では対象自体が複雑で、従来の方法では計算できない場合がある。
- ◆ 特に非線形な場合は、モデルの構造化そのものが難しく、カオス特性を持つ場合は注意が必要。
- ◆ 伝統的なハードコンピューティング（厳密な解を求めるための計算）に対して、ソフトコンピューティングという概念が提案されている。
- ◆ 理由は、現実の世界が不精密、不確実であり従来の方法ではうまく計算できないことによる。
- ◆ 複雑なものをそのまま考える「複雑性の科学」やソフトコンピューティングが現在のところ適用可能で有用な手法。
- ◆ 人工生命やエージェントシステムという分野も適応と言う観点から、将来的には有用。