

第5回セミナーの構成

はじめに

ゲームAI連続セミナーのご紹介

Enjoy AI!

第1部 ゲームAI 技術解説 (100分) (講師: 三宅)

第1章 概論 (10分)

第2章 C4アーキテクチャー (20分)

第3章 遺伝的アルゴリズム (30分)

第4章 ニューラルネットワーク (30分)

第5章 発展 (10分)

各章の終わりに質疑応答を行います。

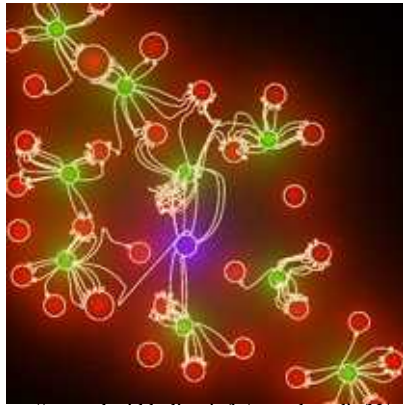
(休憩 20分)

第2部 ディスカッション (2時間) (アイデア: IGDAゲームAI運営委員)

(総合司会 + コーディネート: 長久、グループ司会者)

(I) グループ討論 (80分)

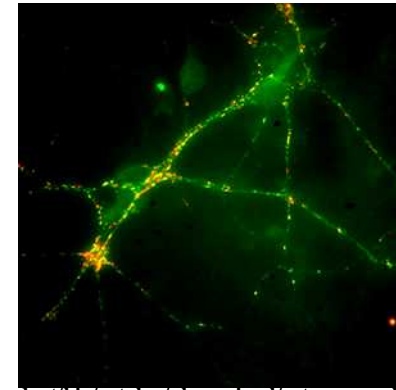
(I) まとめ (30分)



<http://www.daviddarling.info/encyclopedia/N/neuralnet.html>

第4章

ニューラルネットワーク



http://www.sanko-junyaku.co.jp/product/bio/catalog/nhc_animal/rat-neuronal-3striatum.html

1. ニューラルネットへのガイダンス
2. 階層型ニューラルネット
3. ニューラルネットによるエージェント

ニューラルネットで動く3D-PONGのNPC

ニューラルネットで移動する「機雷除去戦車」

2D-シューティングゲームへの応用「バイナリーエイリアン」

衝突回避と探索を行う戦車

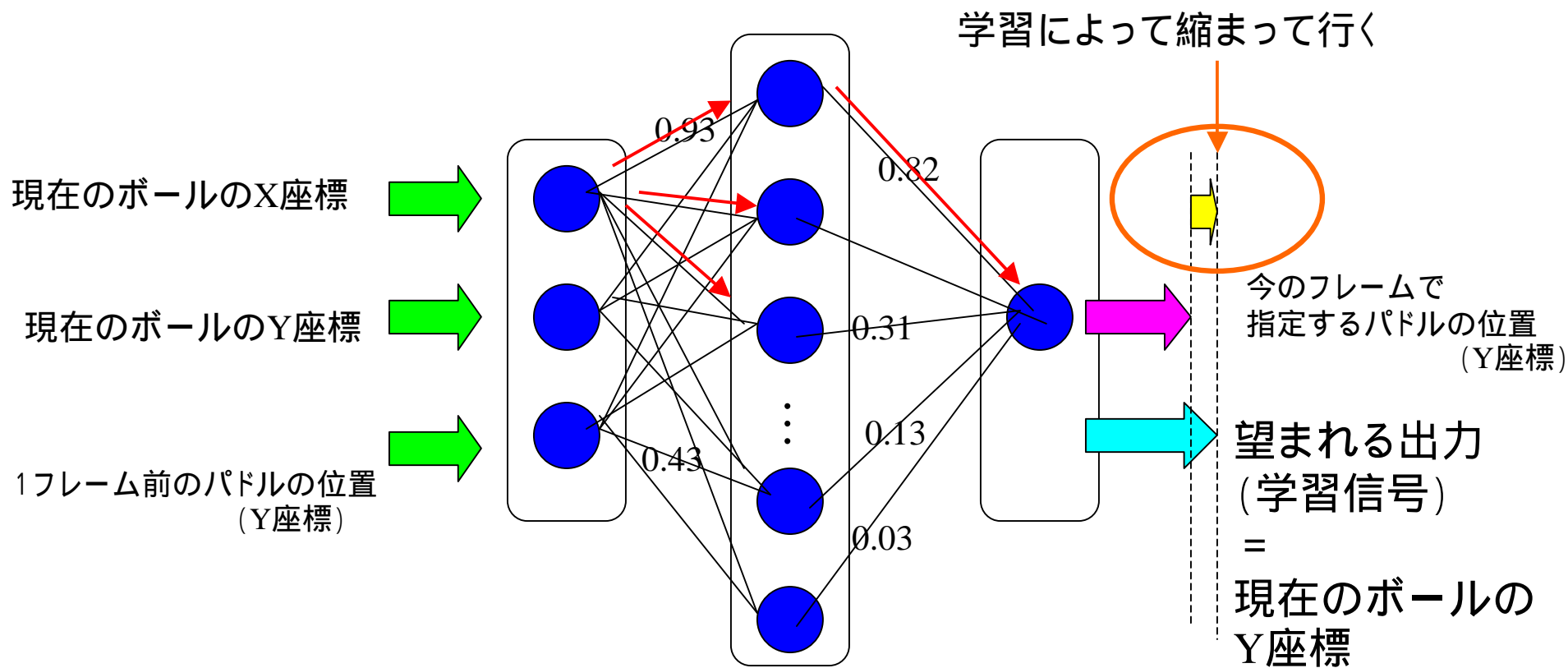
ニューラルネットワークの構造が進化させる「NEAT」の技術

リアルタイム「NEAT」を使ったシミュレーションゲーム「NERO」

ニューラルネットで動くPONGのNPC

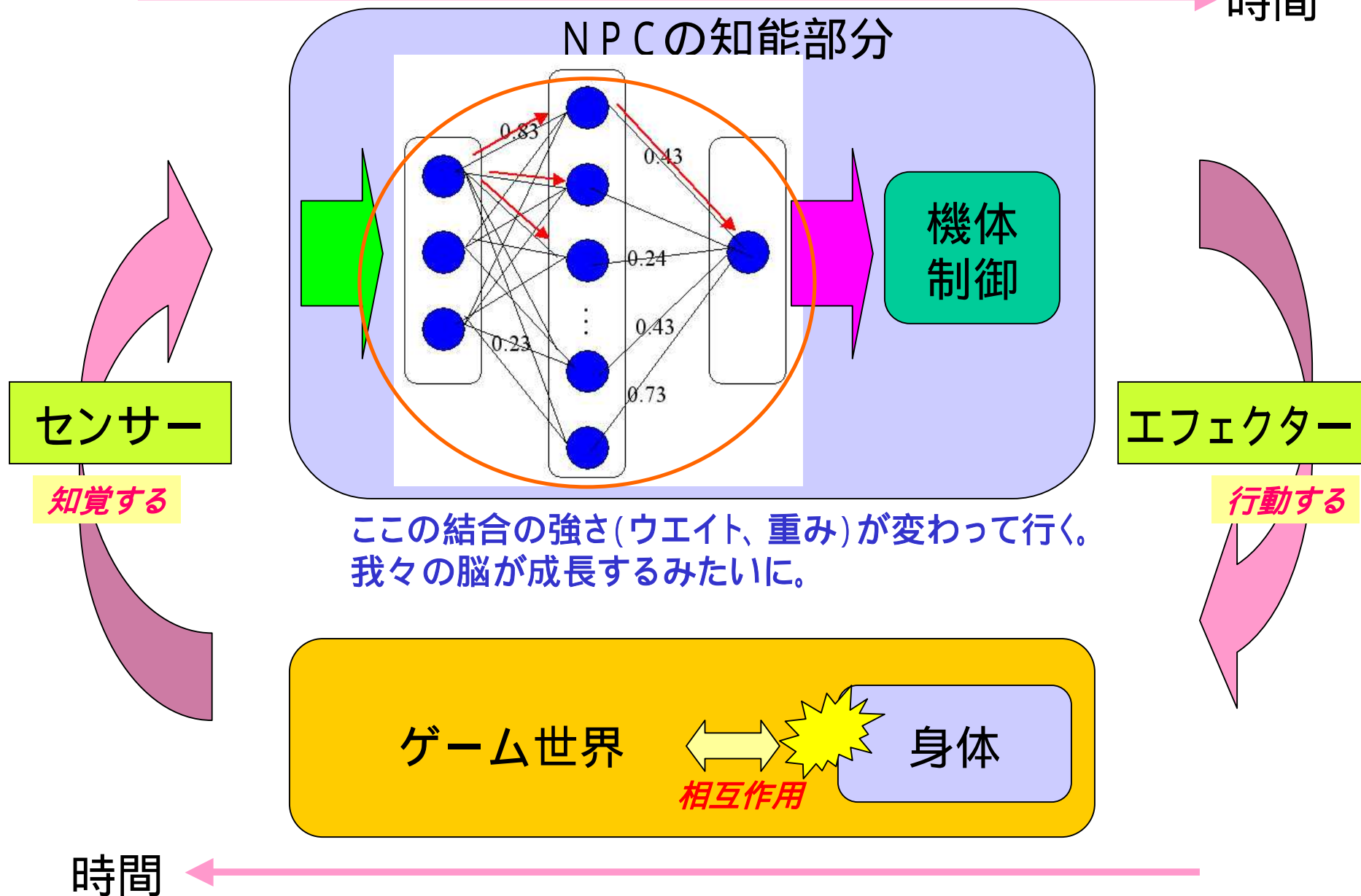
Jeff Meyers (University of Michigan-Dearborn), April 1998

望まれる出力と実際の出力の差から、ウェイトを調整して行く。
(誤差伝播法 = backward propagation of errors)



エージェント・アーキテクチャー

時間 →

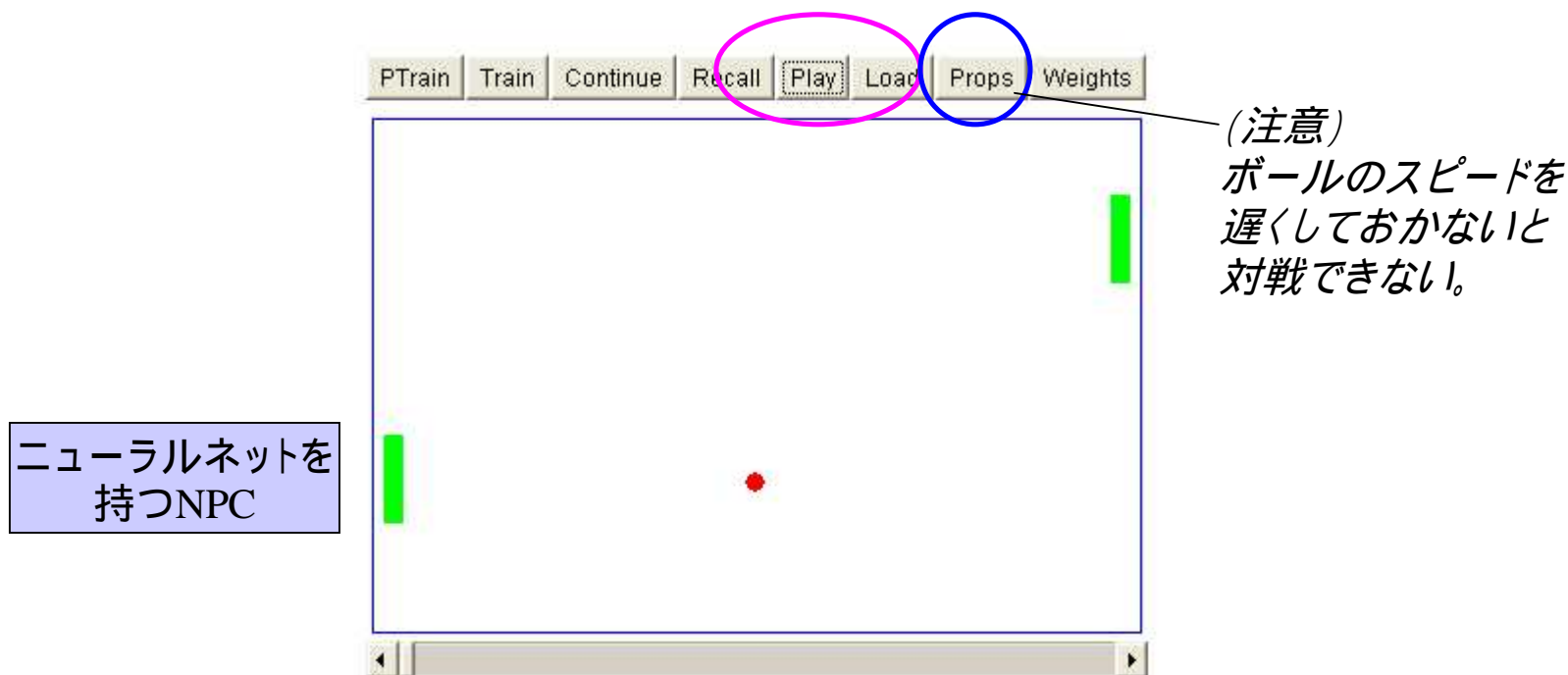


ニューラルネットで動くPONGのNPC

Jeff Meyers (University of Michigan-Dearborn), April 1998

ニューラルネットを持つNPCと対戦する。

ニューラルネットと対戦！

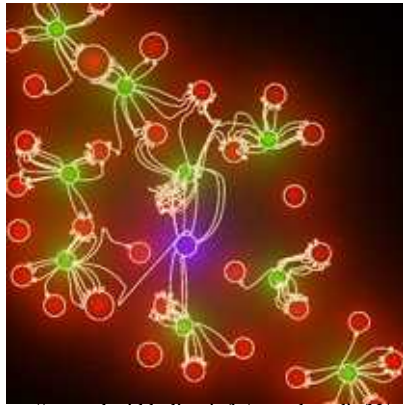


<http://www-personal.engin.umd.umich.edu/~watta/MM/pong/pong5.html>

このNPCは、PONGに対する知識を持っているだろうか？
IF ... then とか、記号で表される形で明示的に持っていない。
PONGに対応した回路という形で持っている。

ガイドンス終了

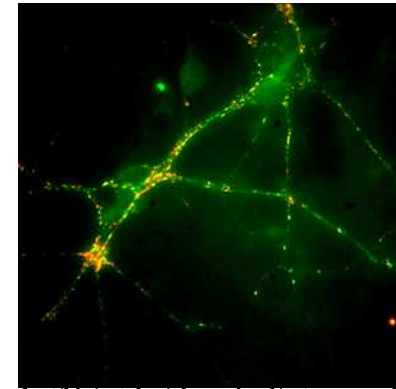
ニューラルネットを簡単に紹介しました。



<http://www.daviddarling.info/encyclopedia/N/neuralnet.html>

第4章

ニューラルネットワーク



http://www.sanko-junyaku.co.jp/product/bio/catalog/nhc_animal/rat-neuronal-3striatum.html

1. ニューラルネットへのガイダンス
2. 階層型ニューラルネット
3. ニューラルネットによるエージェント

ニューラルネットで動く3D-PONGのNPC
ニューラルネットで移動する「機雷除去戦車」
2D-シューティングゲームへの応用「バイナリーエイリアン」
衝突回避と探索を行う戦車
ニューラルネットワークの構造が進化させる「NEAT」の技術
リアルタイム「NEAT」を使ったシミュレーションゲーム「NERO」

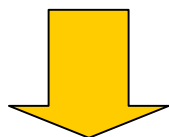
2 . 階層型ニューラルネット

ニューラルネットを理解しよう

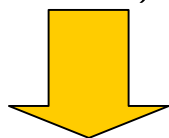
基本思想

コネクショニズム(結合主義)

知性とは脳の活動によって産まれるのだ。



知性とは脳は100億以上の
ニューロン(神経素子)の結合である。

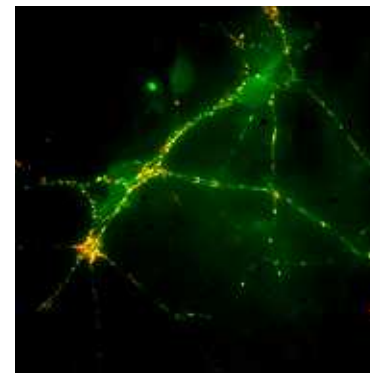


だったら、ニューロン(のモデル)を用いた
回路(ニューラルネットワーク)によって
知能を作ることが可能ではないか！

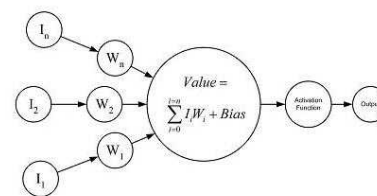
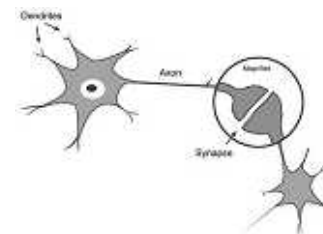
(since 1943)



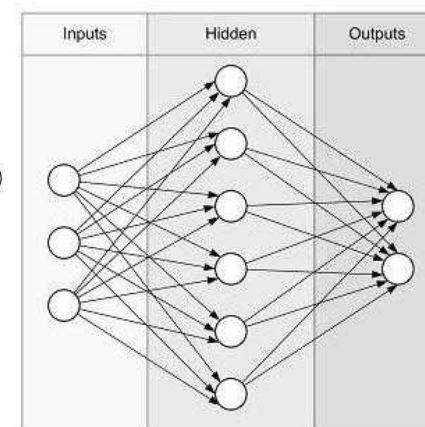
ニューラルネットだけで知性の機能を
全て再現してみよう！



http://www.sanko-junyaku.co.jp/product/bio/catalog/nhc_animal/rat-neuronal-3striatum.html

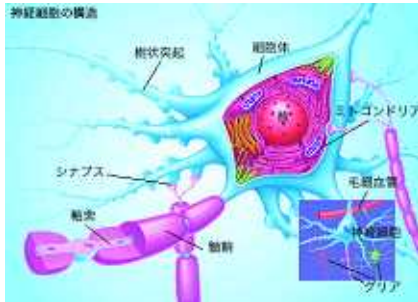


Artificial Neuron

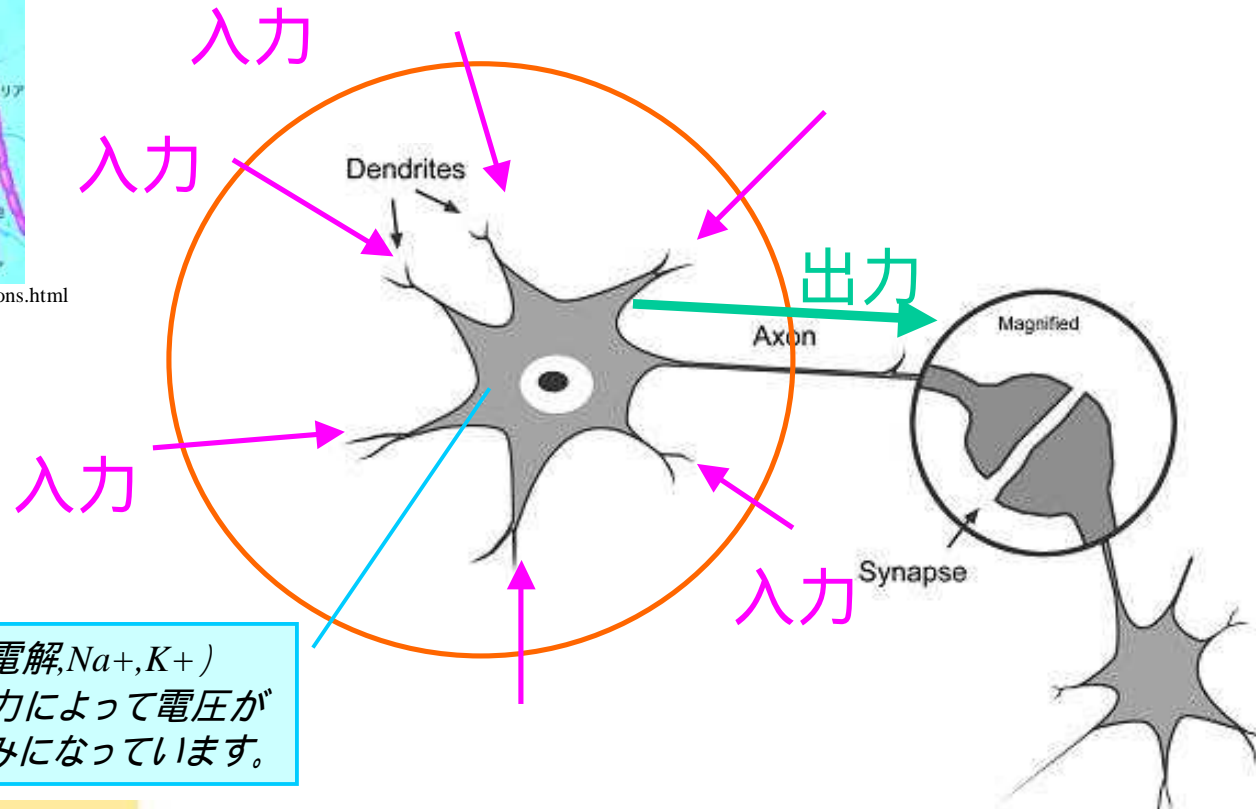


Neural Net

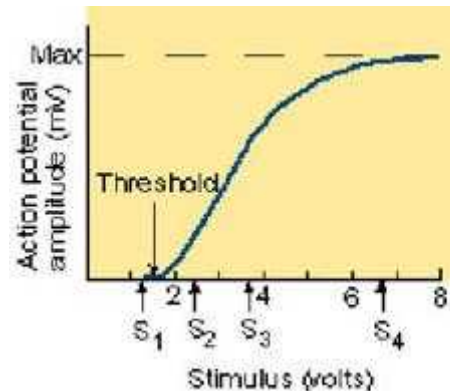
神経素子(ニューロン)とは？



<http://www.brain.riken.go.jp/jp/aware/neurons.html>



この中にはイオン(電解, Na^+ , K^+)
溶液が入っていて、入力によって電圧が
高まると出力する仕組みになっています。

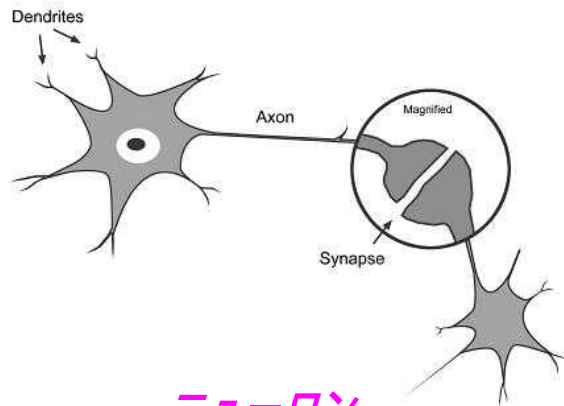


100mV ぐらい

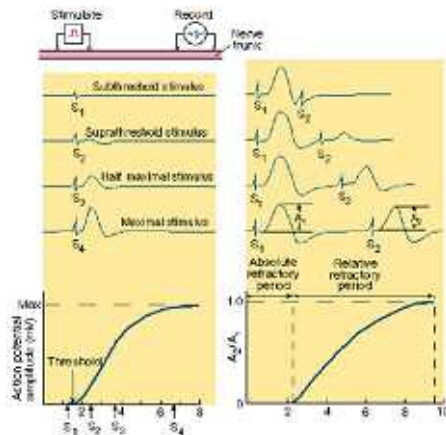
ニューラルネットワーク内シグナル伝達スピード 100(m/sec) ... 案外遅い

ニューラルネットを理解しよう 基本原理

医学的知識

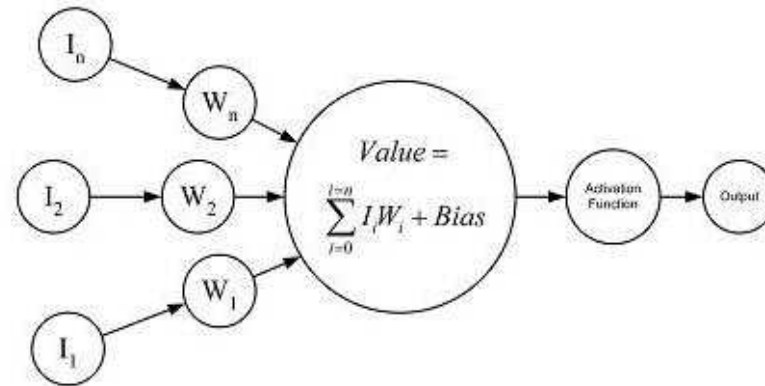


ニューロン



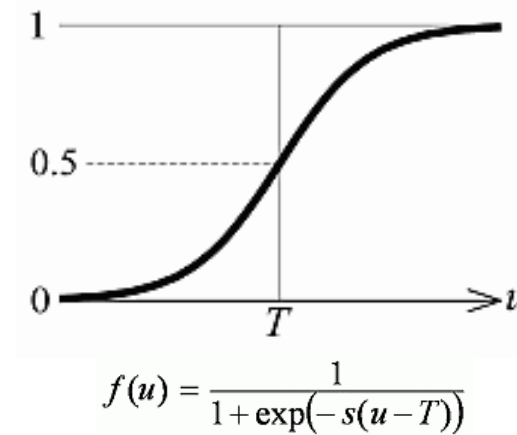
入出力関係のグラフ

数学的モデル



人工ニューロン

モデル化

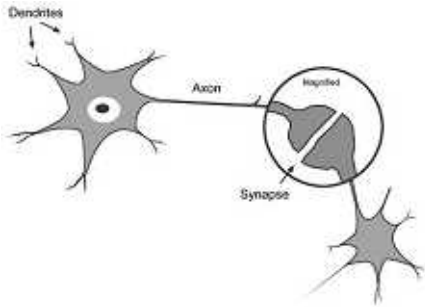


入出力関係の関数(シグモイド関数)

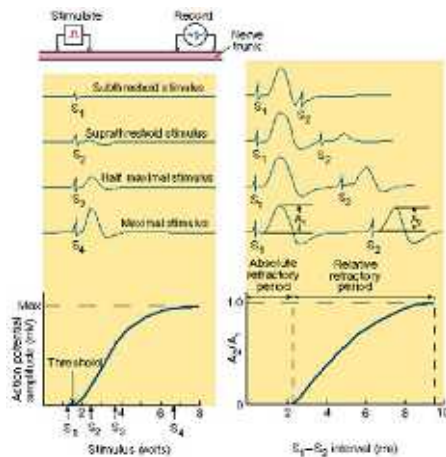
ニューラルネットを理解しよう

数学的原理

医学的知識



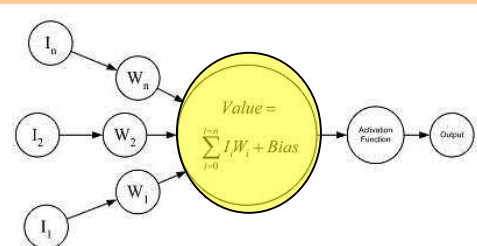
ニューロン



入出力関係のグラフ

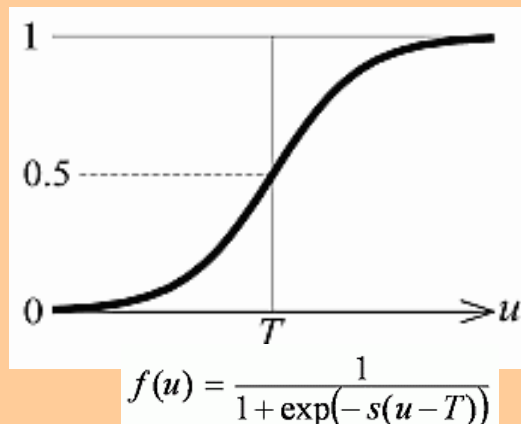
<http://www.pri.kyoto-u.ac.jp/brain/brain/11/index-11.html>

数学的モデル



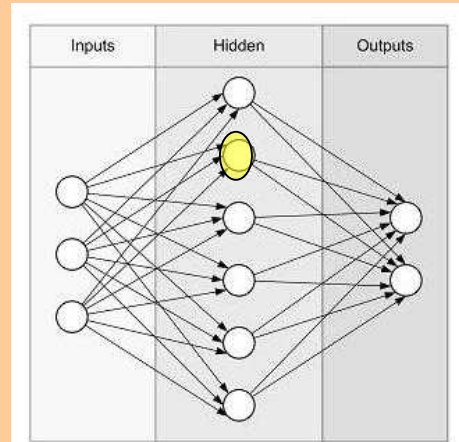
人工ニューロン

モデル化



入出力関係の関数(シグモイド関数)

<http://www.biwako.shiga-u.ac.jp/sensei/mnaka/ut/sozai/ai.html>



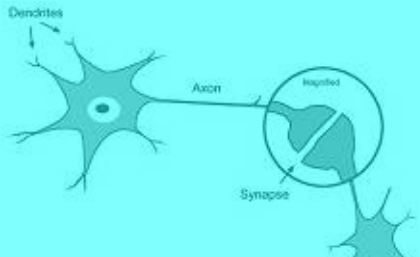
ニューラルネットワーク
(ニューロンをつなげたもの)

道具はこれで全て。これで何ができるだろう？

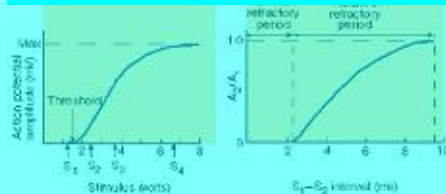
ニューラルネットを理解しよう

基本原理

医学的知識



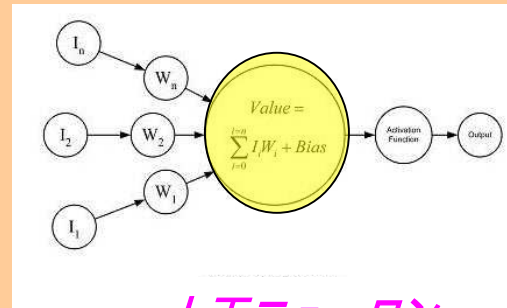
一旦、モデル化したら、元になったモデルはひとまず忘れてよいのです。



入出力関係のグラフ

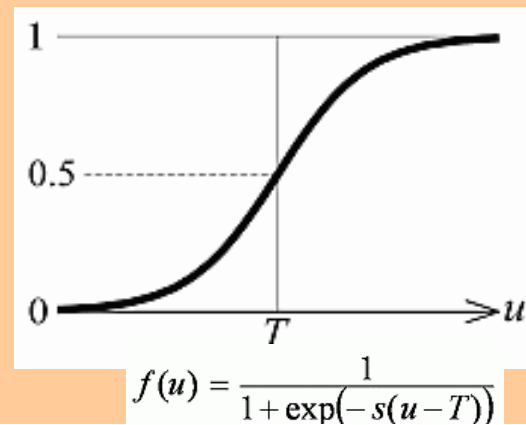
<http://www.pri.kyoto-u.ac.jp/brain/brain/11/index-11.html>

数学的モデル



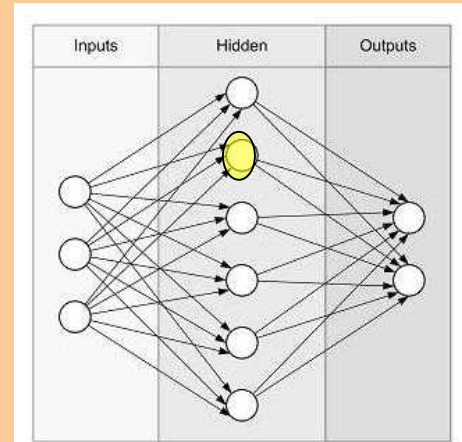
人工ニューロン

モデル化



入出力関係の関数(シグモイド関数)

<http://www.biwako.shiga-u.ac.jp/sensei/mnaka/ut/sozai/ai.html>



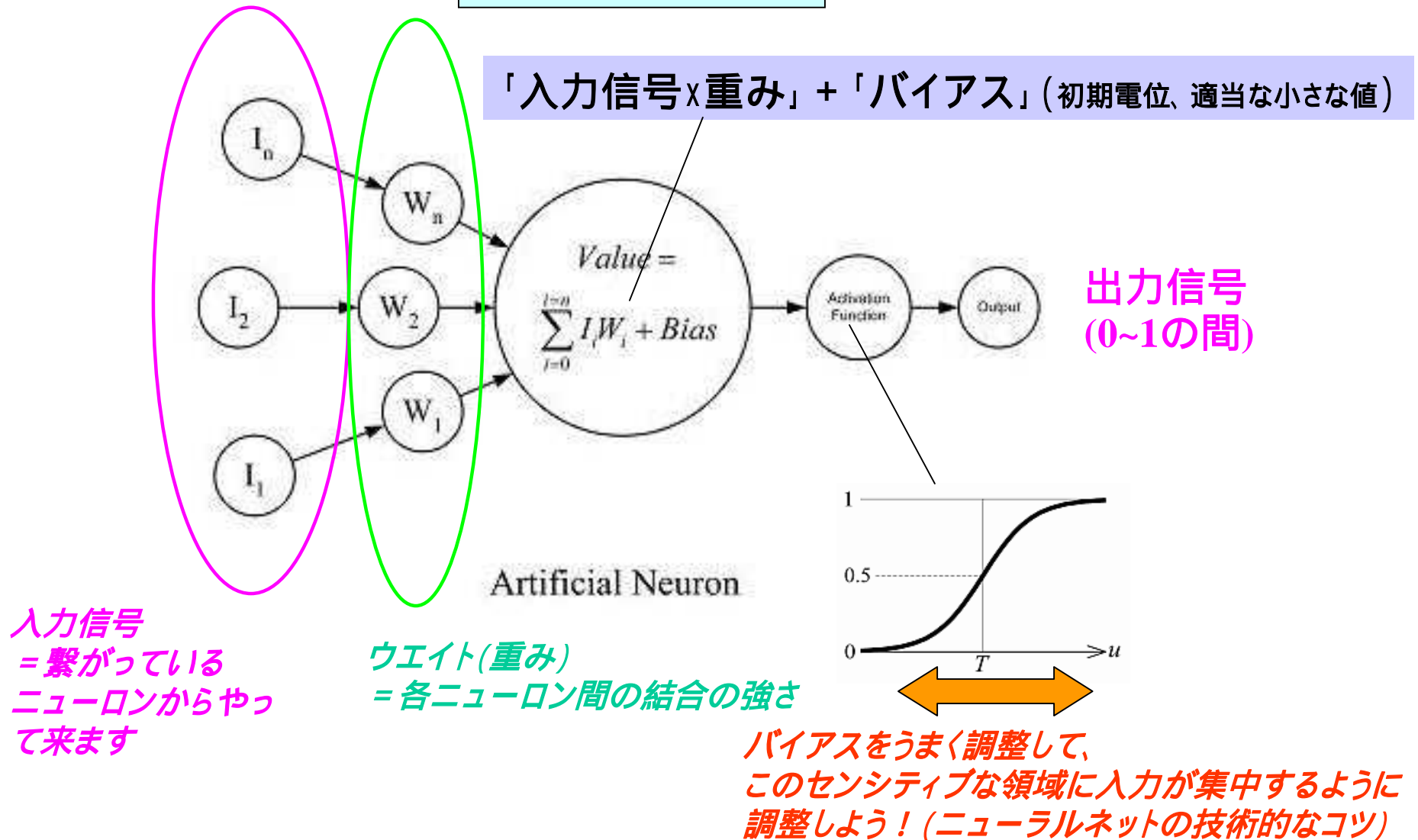
ニューラルネットワーク (ニューロンをつなげたもの)

道具はこれで全て。これで何ができるだろう？

ニューラルネットを理解しよう

数学的原理

数学的モデル



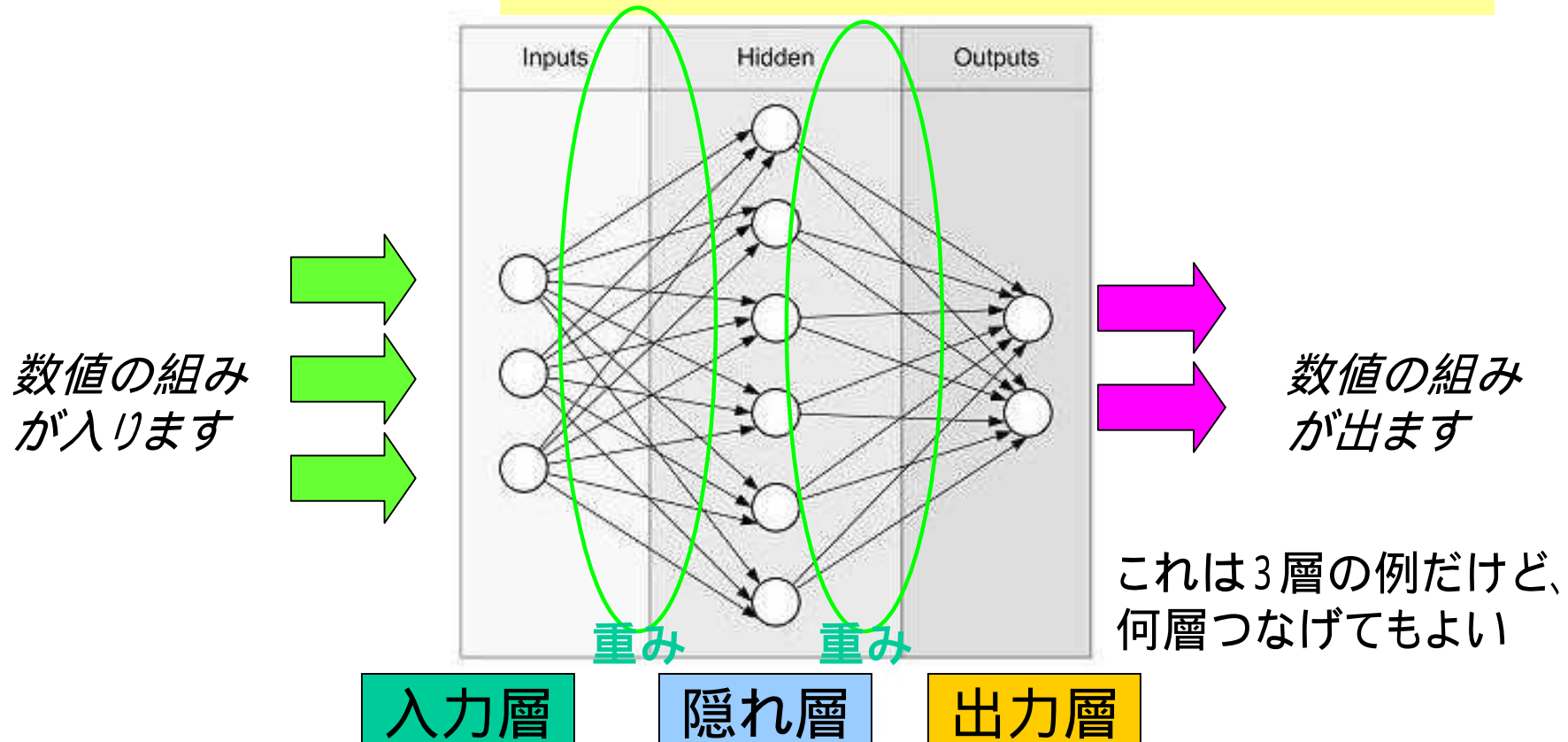
ニューラルネットを理解しよう

数学的原理

数学的モデル

階層型ニューラルネットワーク

... 一方向にニューロンをつなげたもの



最初に定義するもの = ウェイト(重み)、バイアス

とりあえず全ての結合を定義しておく(ニューロン間の重みを0にすれば切れる)

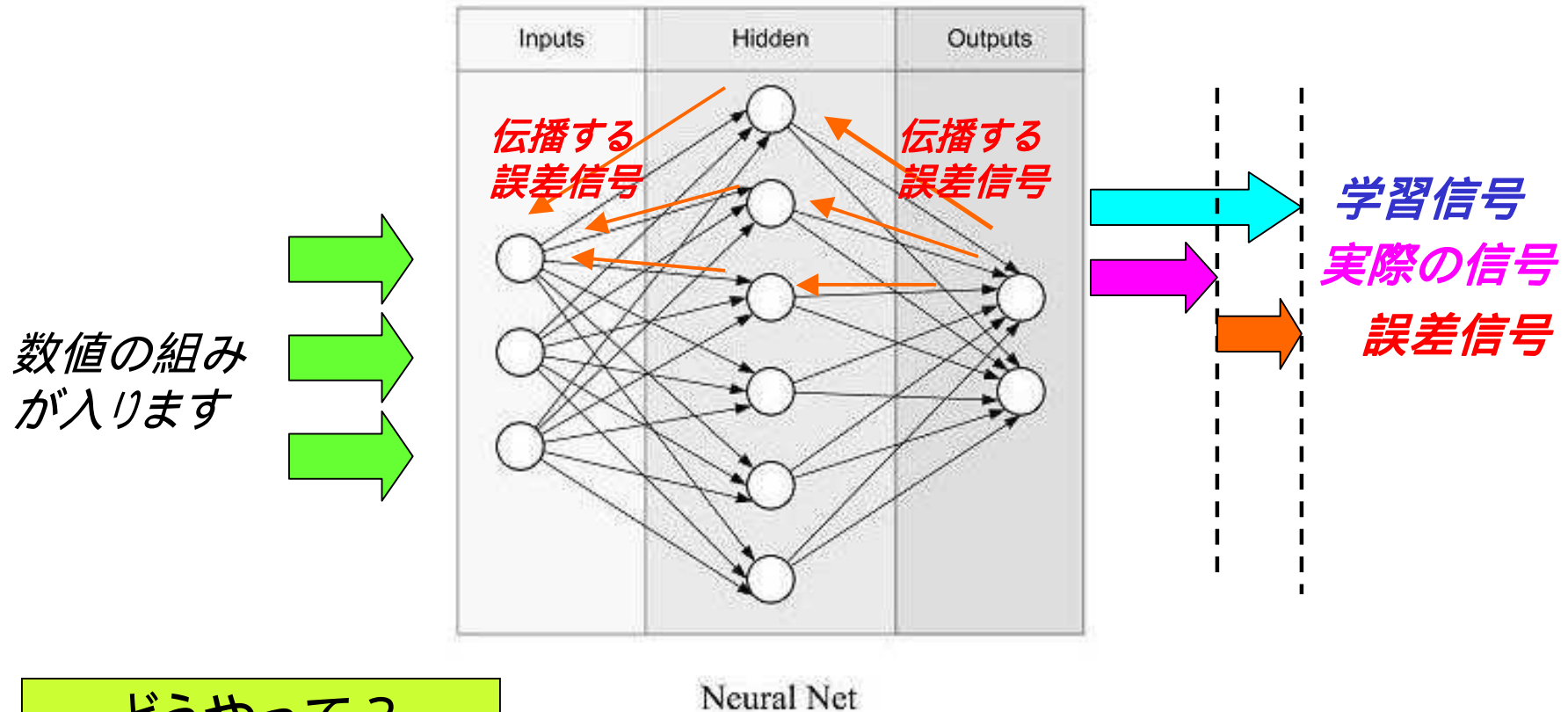
一旦定義してから変えることができないもの... 全体の構造

変えることができるもの... ウェイト(重み)

ニューラルネットを理解しよう 学習

数学的モデル

階層型ニューラルネットでは学習とは、ある入力に対して特定の出力(学習信号)になるようにウェイトを変化させることを言います。



どうやって？

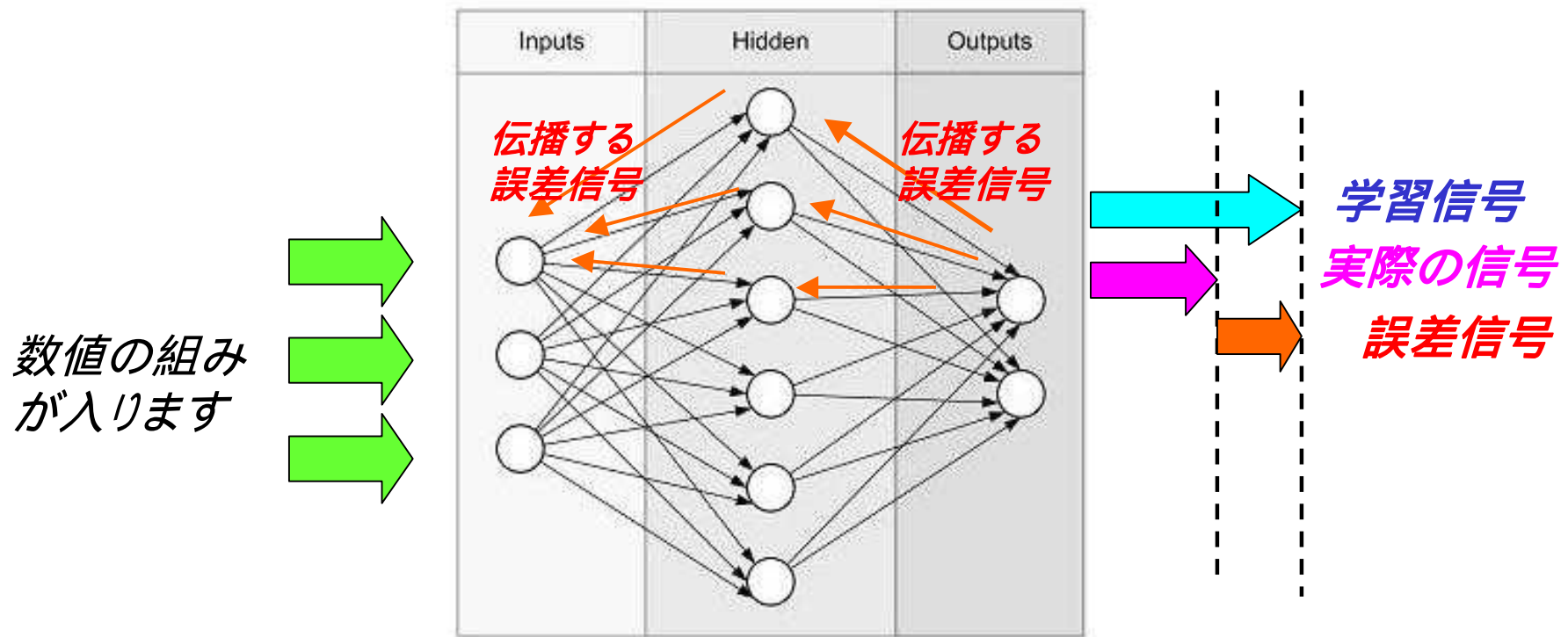
手動で少しずつ勘を頼りに変えて行く。
えらい人が考えた方法を使ってみる。

まず無理
誤差伝播法

ニューラルネットを理解しよう 学習

数学的モデル

教師信号と実際の出力の差を、ウェイトを調整することで、縮めて行く。



誤差伝播法 (Back Propagation Method)

ニューラルネットの出力側から、誤差分を、後ろ側に分担して負担するように、後ろのニューロンへ、そのニューロンが詰める大きさを含んだ情報を伝播して行く。

ここで、誤差伝播法の技術的解説です。
やや煩雑なので、
まずは、サンプルプログラムなどを動かしてみしてから
理解するのがよいと思います。

誤差伝播法を習得しよう

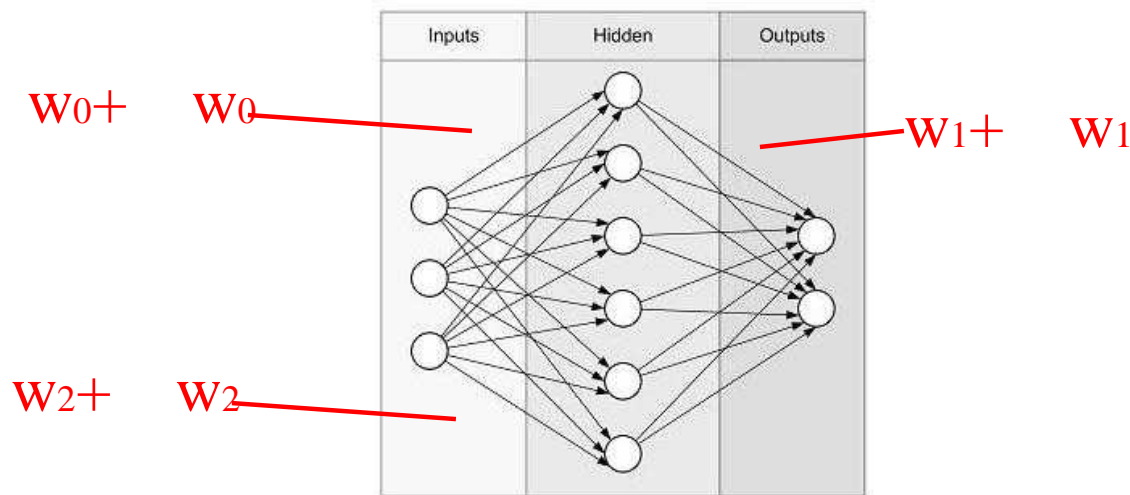
基本的な考え方

「入力層3、隠れ層6、出力層2」なら全部で30個のウエイトがある。

各ウエイトを、ほんの少しずつ変化させて行きます(逐次的近似法)。

$$W + \Delta W$$

これを、同じ教師信号を使って何度もくり返すことで、だんだんと教師信号に近い出力を出させるようにします。



では、一体どれくらい変えればいいのか？

誤差伝播法を習得しよう

最速降下法

変数を操作して、出力を目標値に近付ける方法。

w の関数 $F(w)$ を F_0 に逐次的に近似するには、
 $R = (F(w) - F_0)^2$ として、
十分に小さい数 ϵ によって、

w を $w + \epsilon \times \frac{\partial R}{\partial w}$ としながら近付ける。

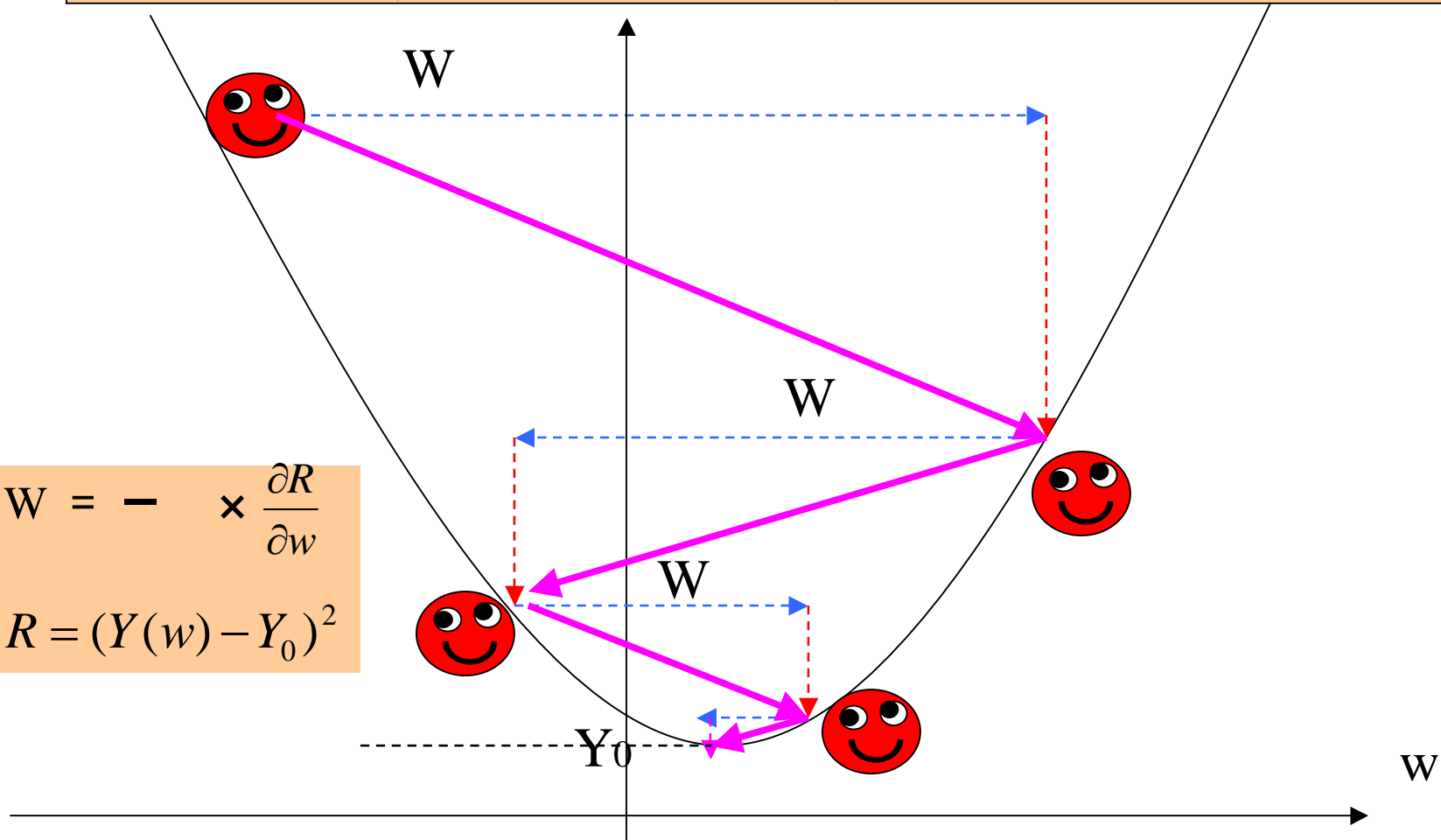
誤差伝播法を習得しよう

最速降下法

とりあえず少しずつずつ進みましょう。(小さい数 を用意。適当なスケール)
傾斜のきついところは最下点より遠いので大胆に進んでしまいましょう。
最下点近くは、傾斜が緩いはずなので、ゆっくり進みましょう。

$$W = - \times \frac{\partial R}{\partial w}$$

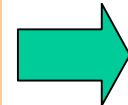
$$R = (Y(w) - Y_0)^2$$



誤差伝播法を習得しよう

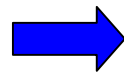
最速降下法

$$W = - \times \frac{\partial R}{\partial w}$$
$$R = (Y(w) - Y_0)^2$$



ニューラルネットの場合(出力層について)

$$W = - \times \frac{\partial R}{\partial w}$$
$$R = (Out(w) - T)^2 / 2$$



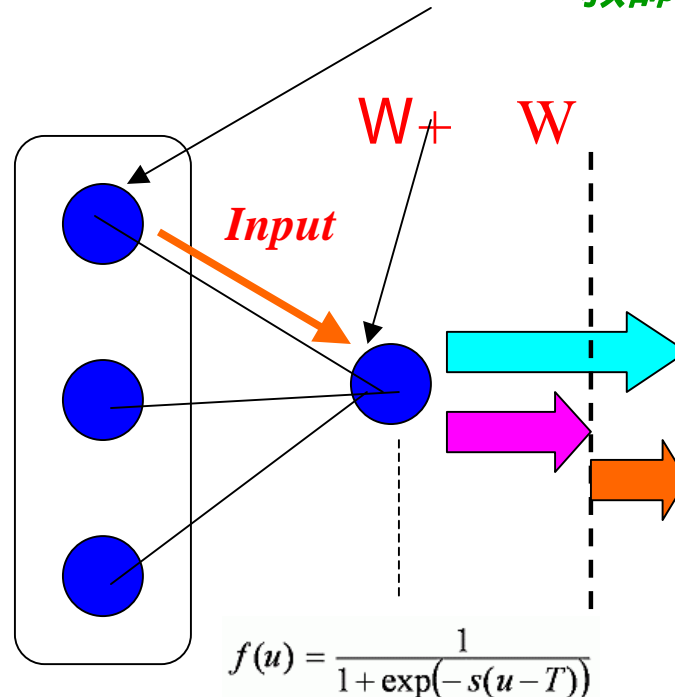
$$\Delta W = -\alpha * Error * Input * Out * (1 - Out)$$

$$\Delta W(n) = -\alpha * Error * Input * Out * (1 - Out) + \eta * \Delta W(n-1)$$

教師信号

1step 前の
変化分

慣性項
(解の振動を抑える)
 $W(n)$ と $W(n-1)$ が
異符号なら、振れ幅を
小さくするから。



学習信号(T)

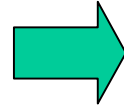
実際の信号(Out)

誤差信号(Error)

誤差伝播法を習得しよう

最速降下法

$$W = - \times \frac{\partial R}{\partial w}$$
$$R = (Y(w) - Y_0)^2$$



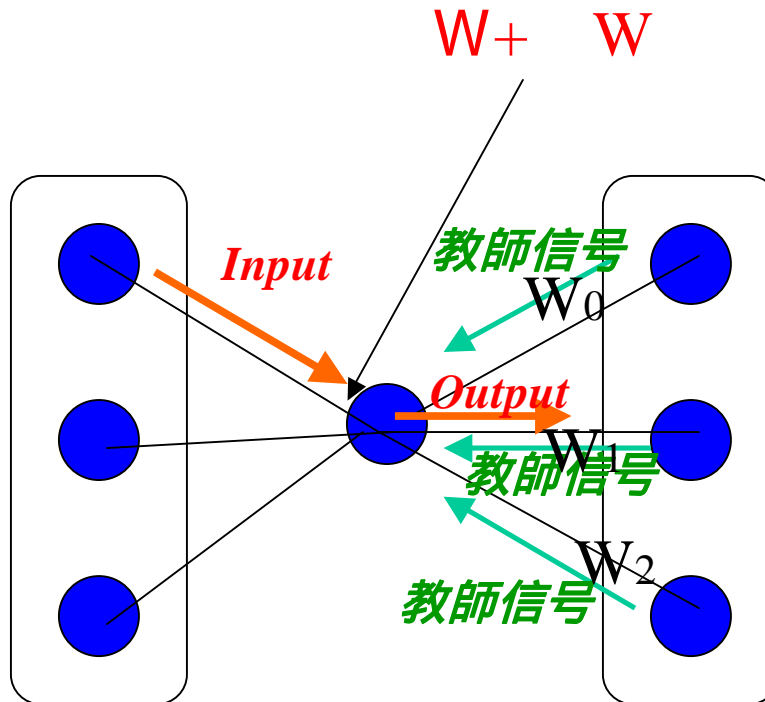
ニューラルネットの場合(出力層でない場合)

$$W = - \times \frac{\partial R}{\partial w}$$
$$R = (Out(w) - T)^2$$

$$\Delta W(n) = -\alpha * Input * Out * (1 - Out) * \sum_i w_i * (教師信号) - \eta * \Delta W(n-1)$$

1step 前の
変化分

慣性項
(解の振動を抑える)



誤差伝播法解説おわり

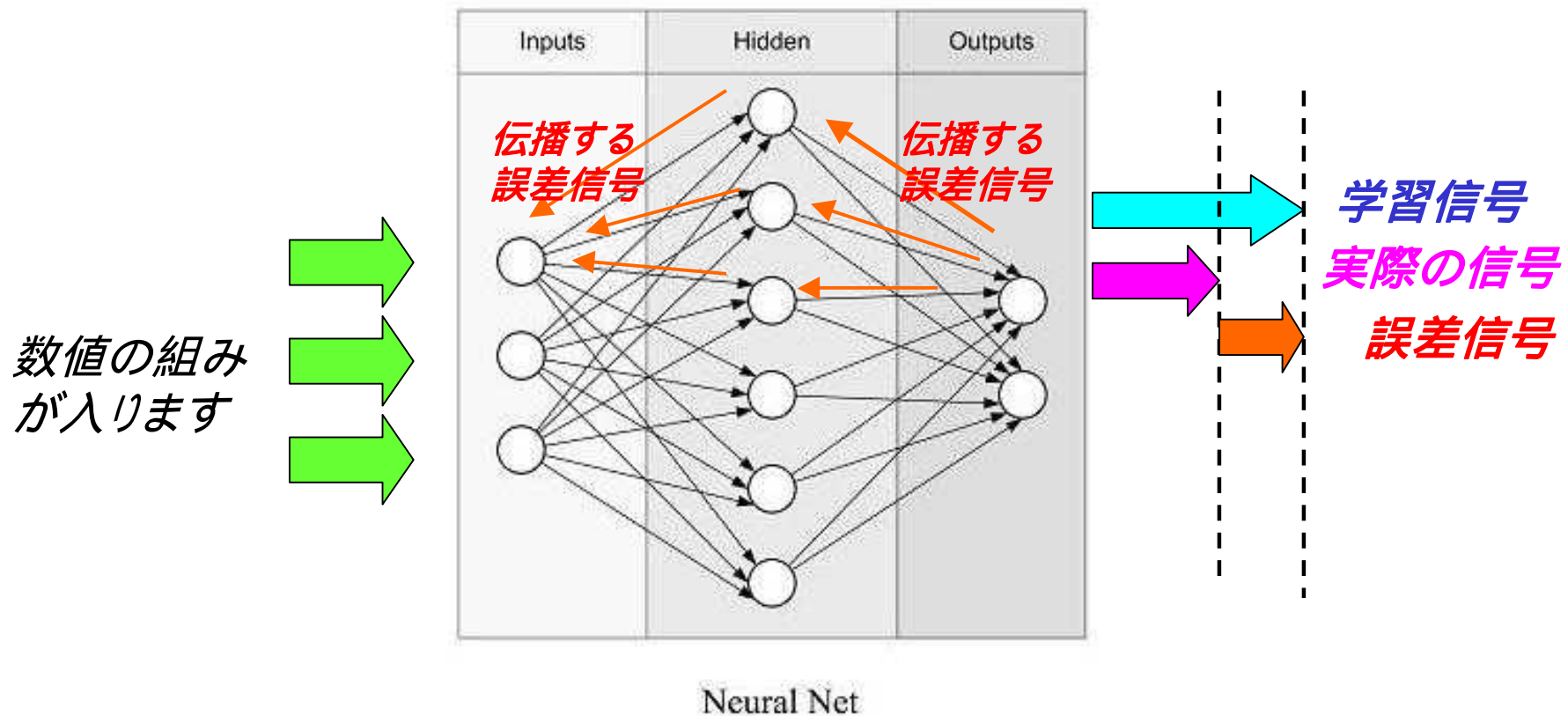
(ここはニューラルネットで唯一難解な部分です。
幾つかの参考書を読まれることをお勧めします。)

(参考文献例) 平野 廣美 著「Cで作るニューラルネットワーク」

ニューラルネットを理解しよう 学習

数学的モデル

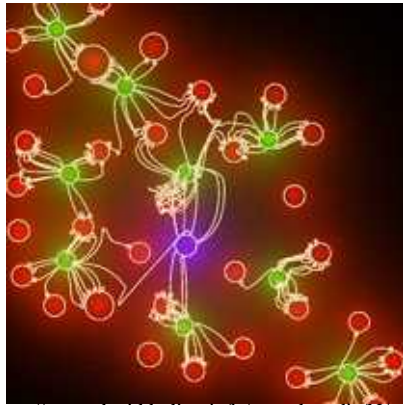
教師信号と実際の出力の差を、ウェイトを調整することで、縮めて行く。



一つの学習信号に対して、誤差伝播法を何度も繰り返してウェイトを変化させて収束させ、学習信号を出力できるようにします。

以上で、ニューラルネットワークの仕組みの
説明はおしまいです。

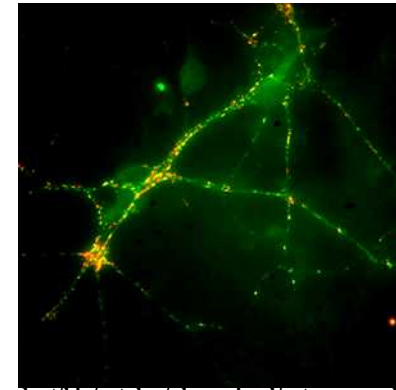
ここからは具体的な使い方を見てみましょう！



<http://www.daviddarling.info/encyclopedia/N/neuralnet.html>

第4章

ニューラルネットワーク



http://www.sanko-junyaku.co.jp/product/bio/catalog/nhc_animal/rat-neuronal-3striatum.html

1. ニューラルネットへのガイダンス
2. 階層型ニューラルネット
3. ニューラルネットによるエージェント

ニューラルネットで動く3D-PONGのNPC
ニューラルネットで移動する「機雷除去戦車」
2D-シューティングゲームへの応用「バイナリーエイリアン」
衝突回避と探索を行う戦車
ニューラルネットワークの構造が進化させる「NEAT」の技術
リアルタイム「NEAT」を使ったシミュレーションゲーム「NERO」

3 . ニューラルネットを持つエージェント

(主要参考文献)

**Mat Buckland, AI techniques for game programming,
Premier Press, 2002**

(実行ファイルとソースコードがCD-ROMにあります)

ニューラルネットを持つエージェント

(1) 学習させる方法 ... 誤差伝播法

(2) 進化させる方法 ... 遺伝的アルゴリズム

~ *NN x GA の手法*

ニューラルネットで動く3D-PONGのNPC

ニューラルネットで移動する「機雷除去戦車」

2D-シューティングゲームへの応用「バイナリーエイリアン」

衝突回避と探索を行う戦車

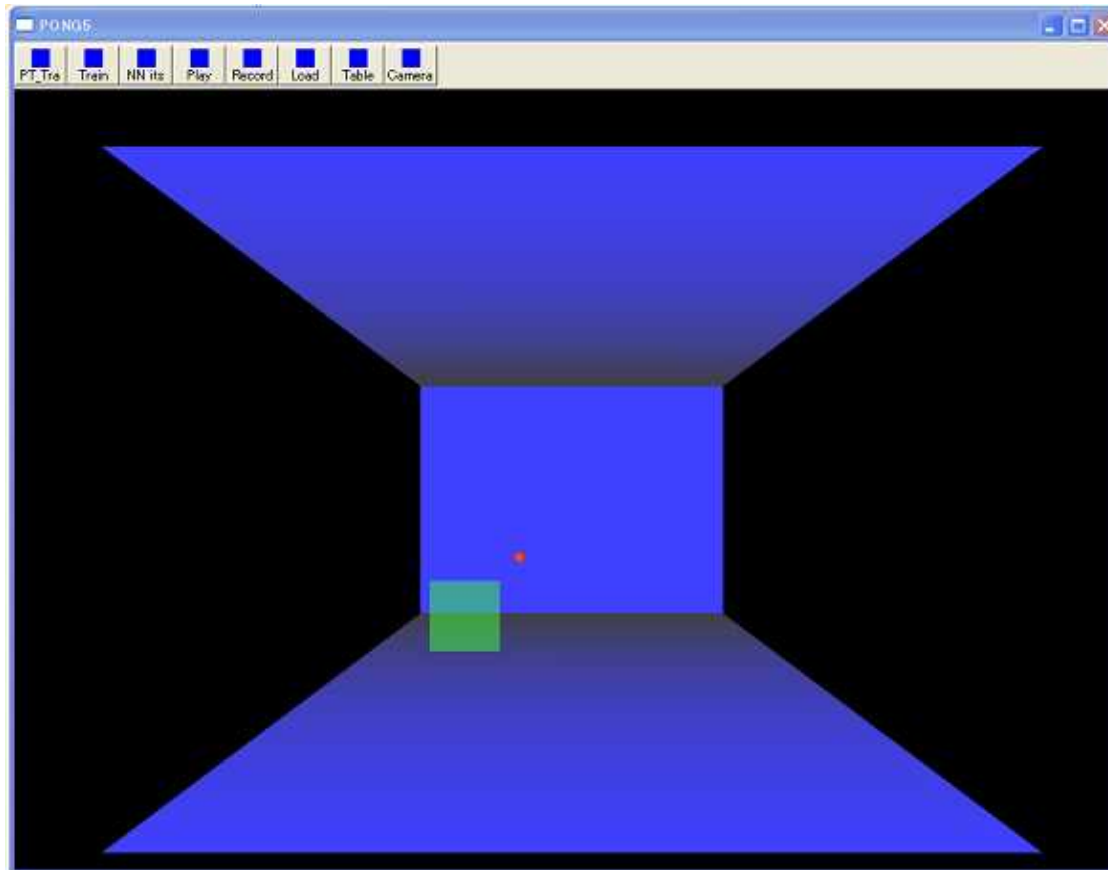
ニューラルネットワークの構造が進化させる「NEAT」の技術

リアルタイム「NEAT」を使ったシミュレーションゲーム「NERO」

ニューラルネットで動く3D-PONGのNPC

Youichiro Miyake, April 2004, Based on Jeff Meyers (University of Michigan-Dearborn), April 1998

<http://www-personal.engin.umd.umich.edu/~watta/MM/pong/pong5.html>



ニューラルネットで
動くNPC

List Pop		
Layer	Input	Output
we_hi[29][0]	10/100	0/100
we_hi[30][0]	-33/100	0/100
we_hi[31][0]	16/100	0/100
we_hi[32][0]	-4/100	0/100
we_hi[33][0]	-14/100	0/100
we_hi[34][0]	-43/100	0/100
we_hi[35][0]	10/100	0/100
we_hi[36][0]	28/100	0/100
we_hi[37][0]	29/100	0/100
we_hi[38][0]	1/100	0/100
we_hi[39][0]	-19/100	0/100
we_hi[40][0]	37/100	0/100
we_hi[41][0]	22/100	0/100
we_hi[42][0]	45/100	0/100
we_hi[43][0]	42/100	0/100
we_hi[44][0]	3/100	0/100
we_hi[45][0]	-35/100	0/100
we_hi[46][0]	-3/100	0/100
we_hi[47][0]	-26/100	0/100
we_hi[48][0]	35/100	0/100
we_hi[49][0]	-28/100	0/100
we_hi[50][0]	27/100	0/100
we_hi[51][0]	33/100	0/100
we_hi[52][0]	49/100	0/100
we_hi[53][0]	49/100	0/100
we_hi[54][0]	11/100	0/100
we_hi[55][0]	-10/100	0/100
we_hi[56][0]	-23/100	0/100
we_hi[57][0]	-20/100	0/100
we_hi[58][0]	33/100	0/100
we_hi[59][0]	-47/100	0/100
we_hi[60][0]	-12/100	0/100
we_hi[61][0]	-40/100	0/100
we_hi[62][0]	17/100	0/100
we_hi[63][0]	-43/100	0/100
we_hi[64][0]	-48/100	0/100
we_hi[65][0]	41/100	0/100
we_hi[66][0]	-22/100	0/100
we_hi[67][0]	-22/100	0/100
we_hi[68][0]	8/100	0/100
we_hi[69][0]	18/100	0/100
we_hi[70][0]	33/100	0/100
we_hi[71][0]	22/100	0/100
we_hi[72][0]	-1/100	0/100
we_hi[73][0]	-29/100	0/100

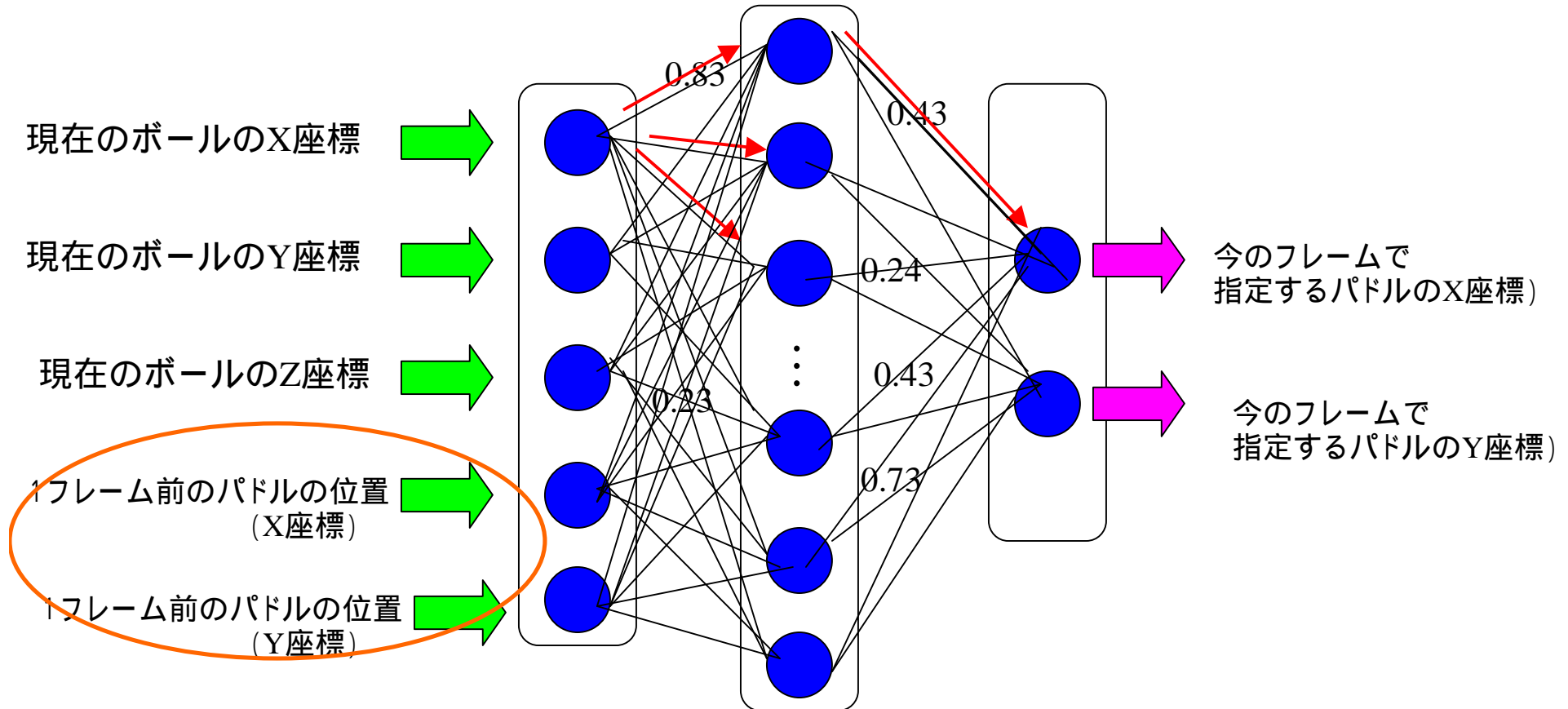
訓練と共に
変化するウェイト

ニューラルネットで動く3D-PONGのNPC

Youichiro Miyake, April 2004, Based on Jeff Meyers (University of Michigan-Dearborn), April 1998

<http://www-personal.engin.umd.umich.edu/~watta/MM/pong/pong5.html>

ボールと自分の位置から、次の瞬間の位置をニューラルネットによって決定する。

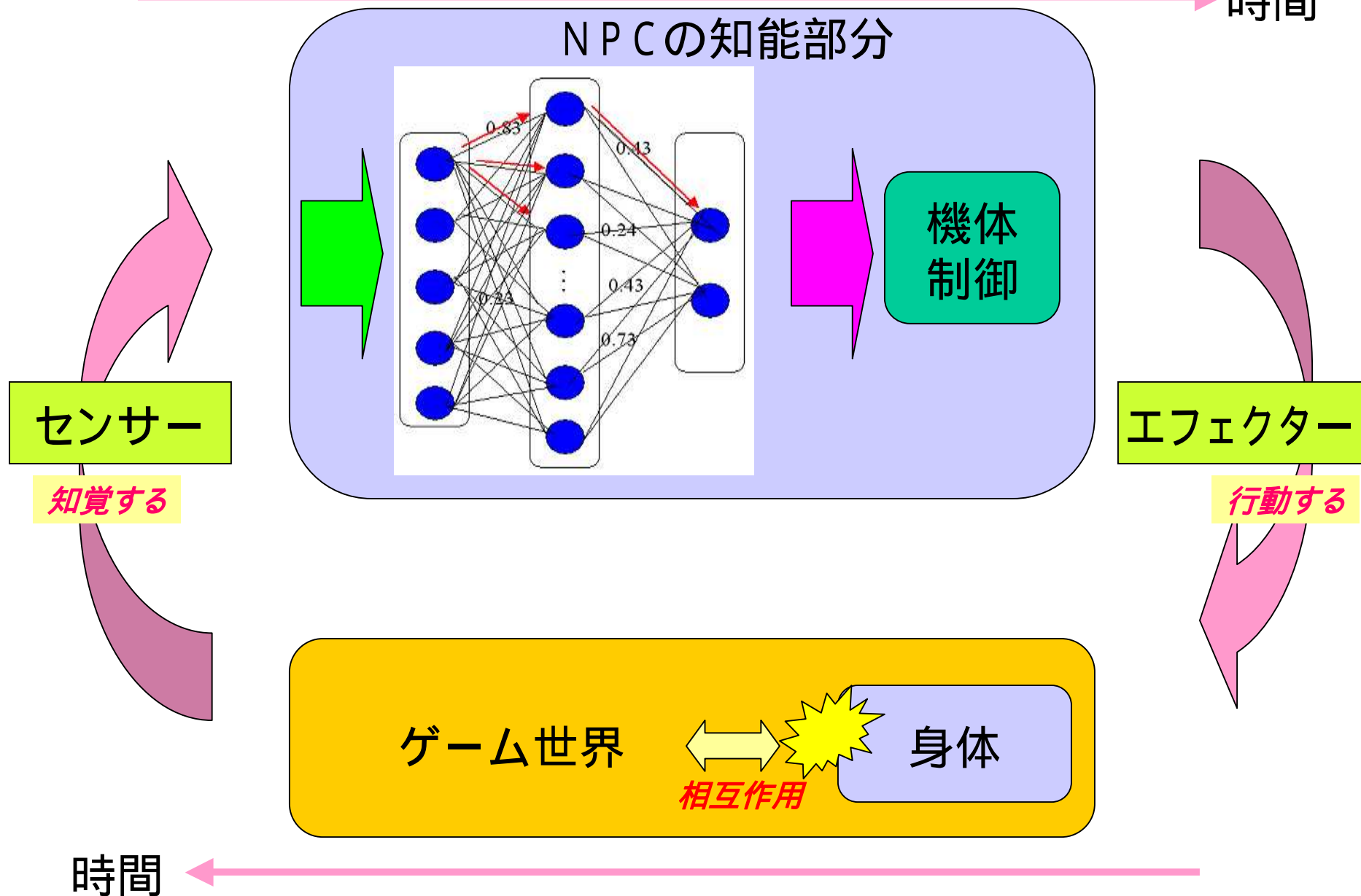


一つ前の状態がインプットとしてフィードバックされている。
このようなニューラルネットを **リカレントニューラルネット** という。

→ フィードバックの機能を実現。アクション制御に利用できる。

エージェント・アーキテクチャ

時間 →

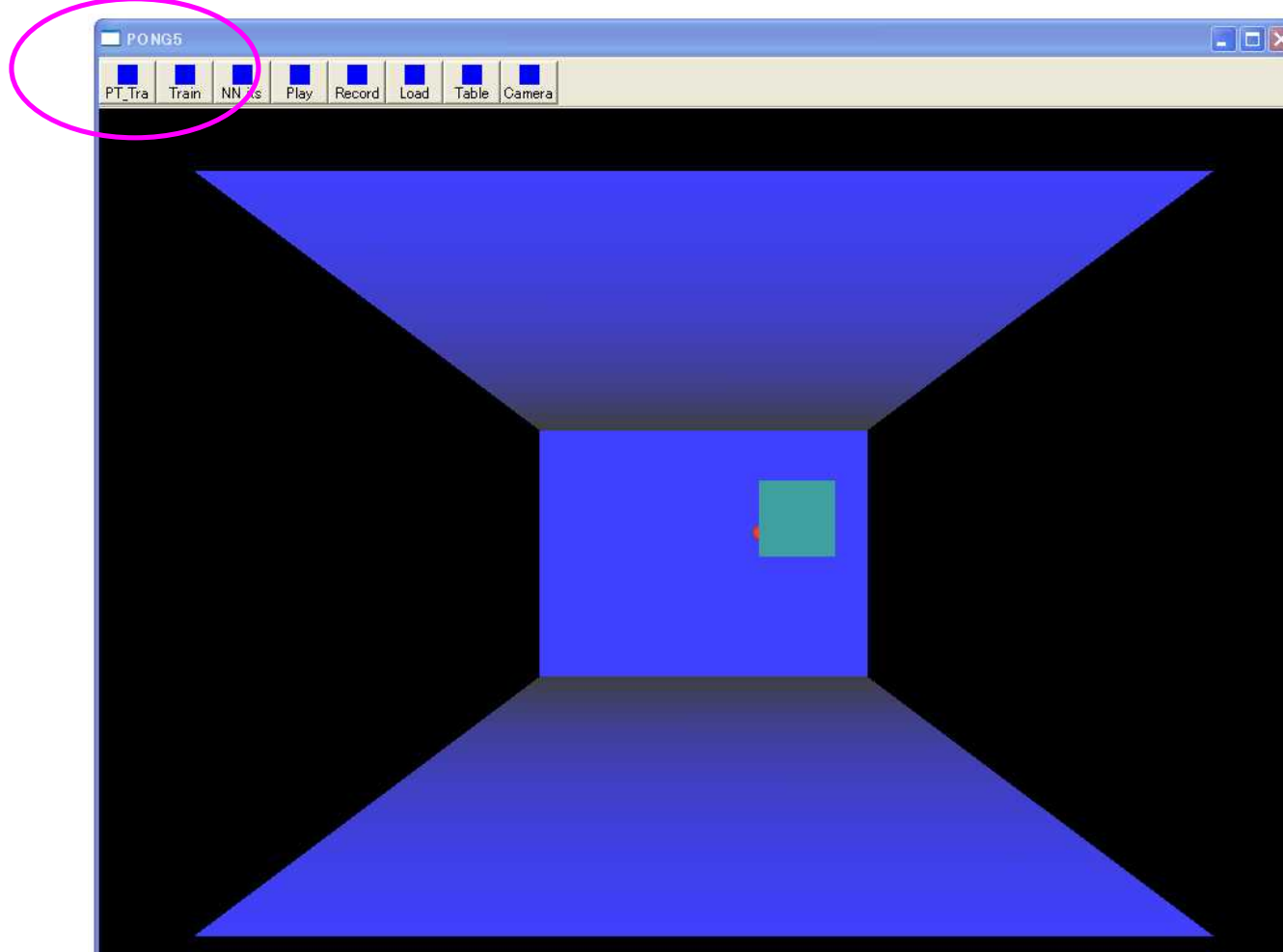


ニューラルネットで動く3D-PONGのNPC

Youichiro Miyake, April 2004, Based on Jeff Meyers (University of Michigan-Dearborn), April 1998

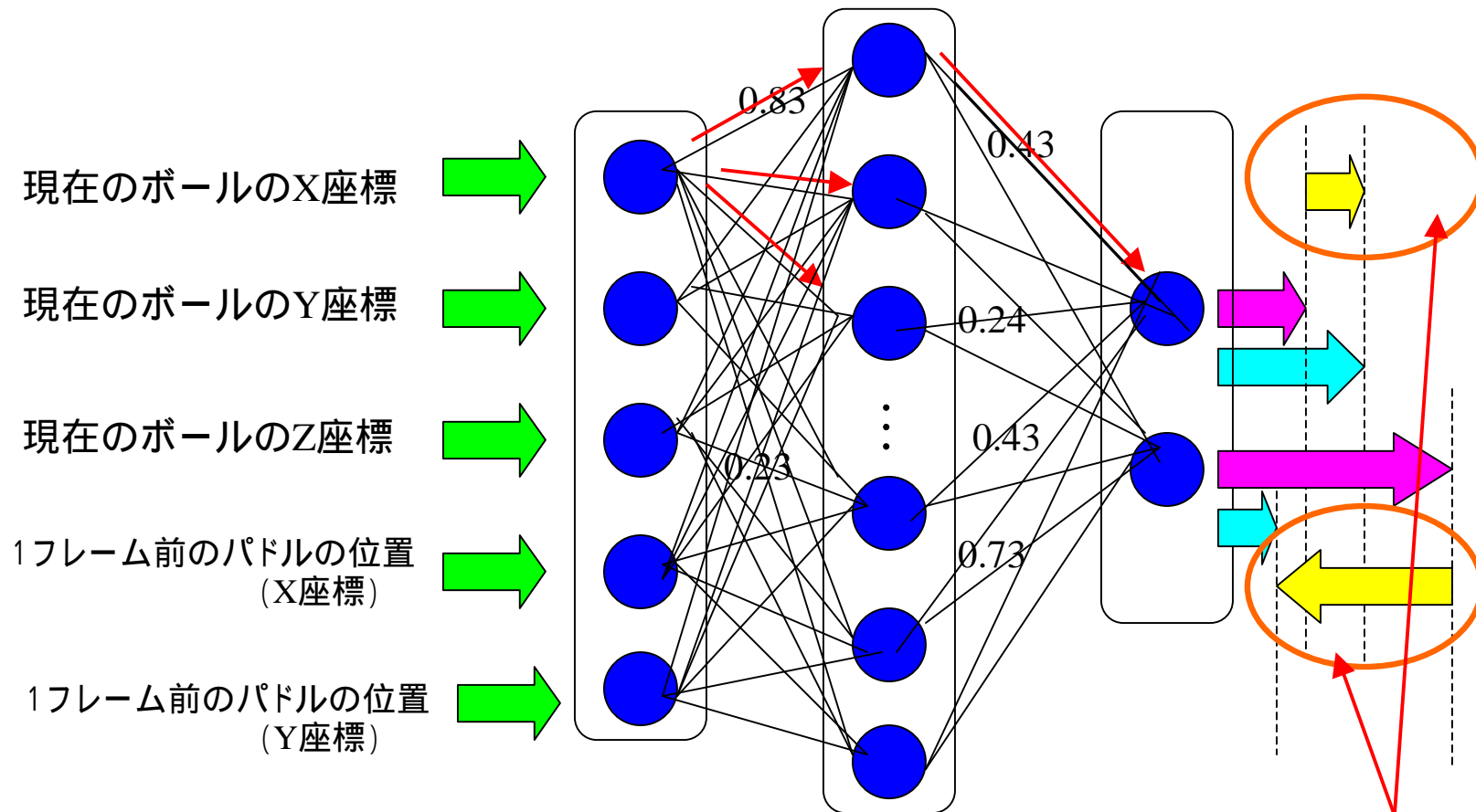
最初は何もできないNPCだが、学習によって賢くすることが出来る。

トレーニングモード



ニューラルネットで動く3D-PONGのNPC

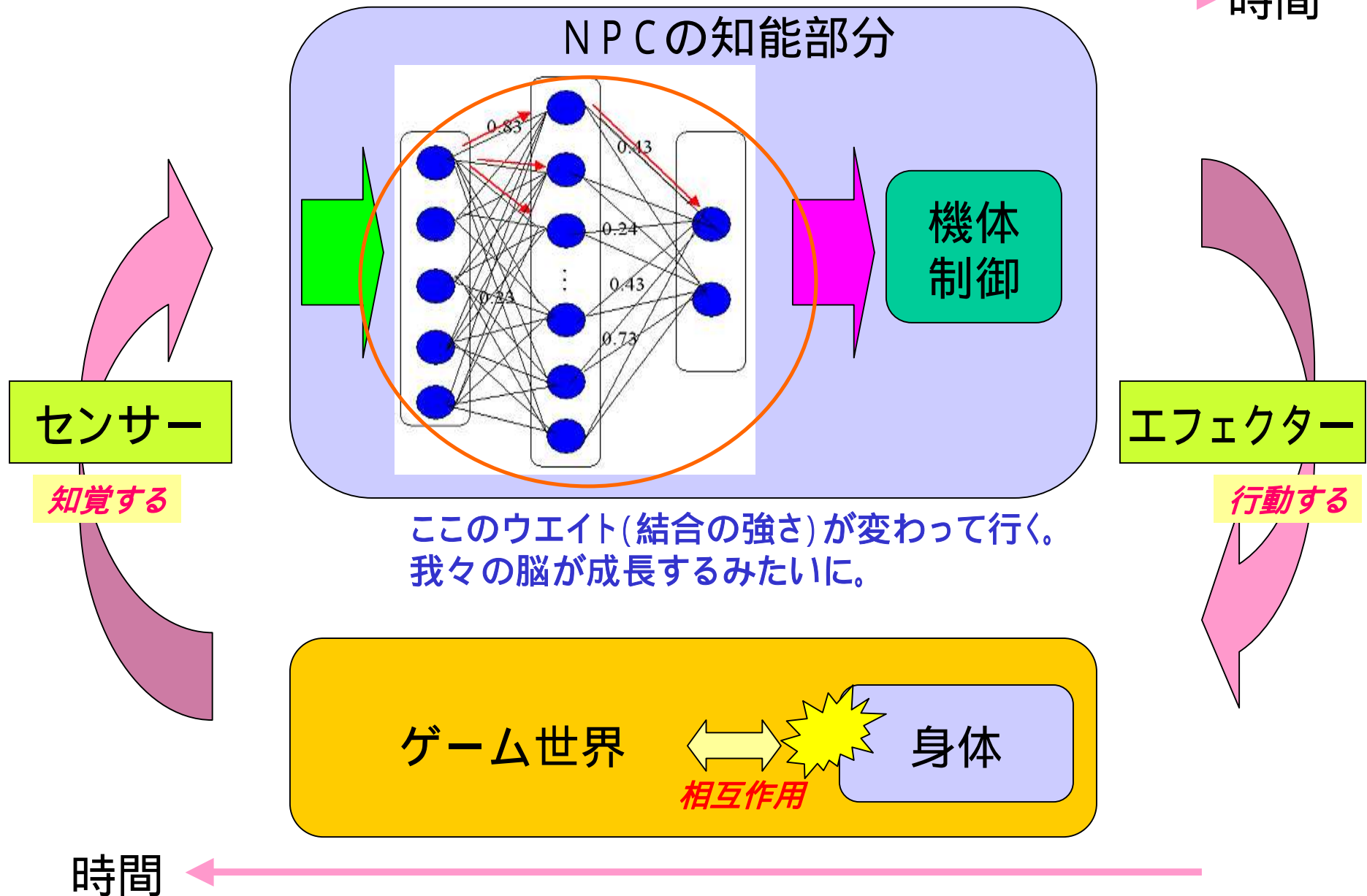
望まれる出力(学習信号)と実際の出力の差から、
ウェイトを調整して行く。(誤差伝播法 = backward propagation of errors)



この両者に対して、誤差伝播法を
多数回、くり返し、NNに学習させる

エージェント・アーキテクチャー

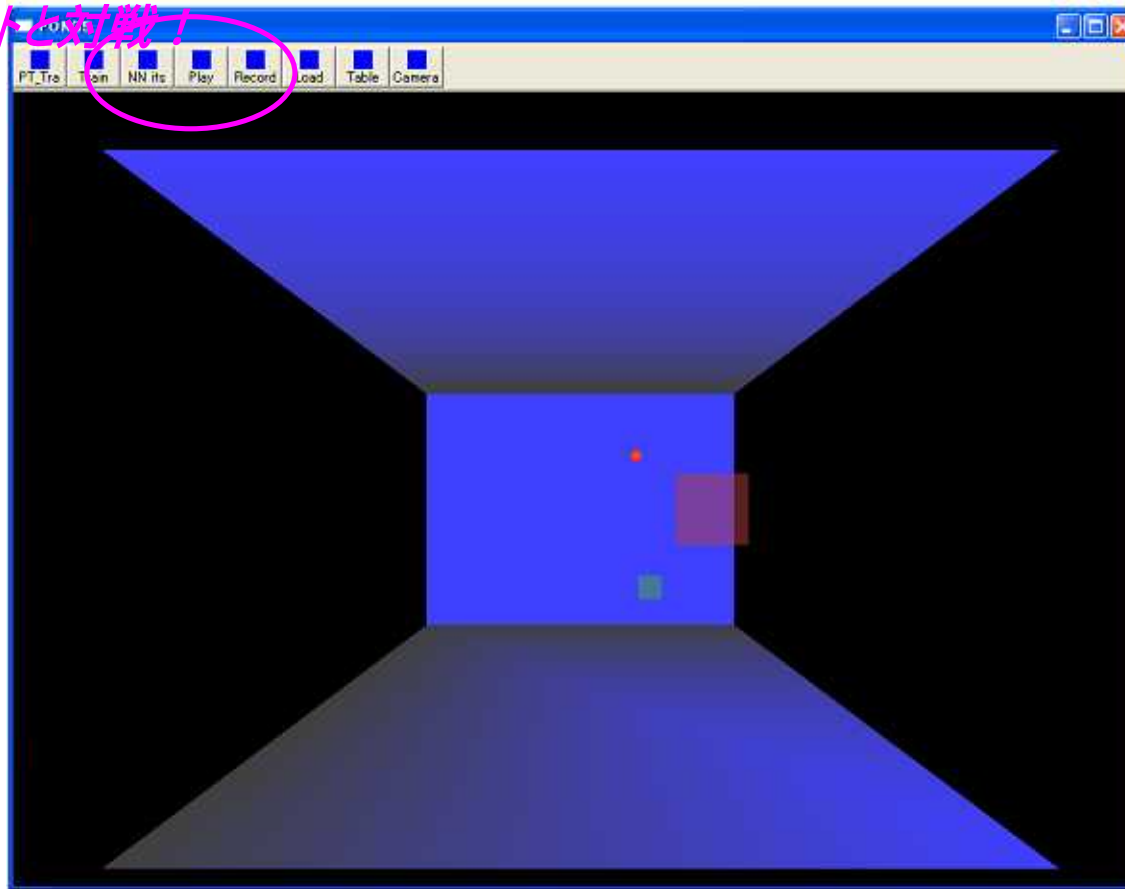
時間 →



ニューラルネットで動く3D-PONGのNPC

ニューラルネットを持つNPCと対戦する。

ニューラルネットと対戦！



このNPCは、PONGに対する知識を持っているだろうか？
IF ... then とか、記号で表される形で明示的に持っていない。
PONGに対応した回路という形で持っている。

ニューラルネットを持つエージェント

(1) 学習させる方法 ... 誤差伝播法

(2) 進化させる方法 ... 遺伝的アルゴリズム

、 、 ~ **NN×GAの手法**

ニューラルネットで動く3D-PONGのNPC

ニューラルネットで移動する「機雷除去戦車」

2D-シューティングゲームへの応用「バイナリーエイリアン」

衝突回避と探索を行う戦車

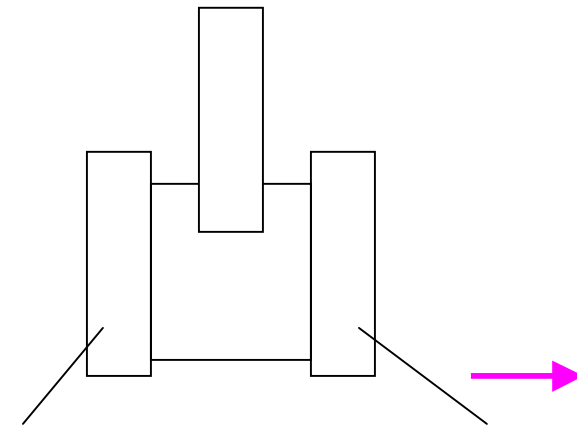
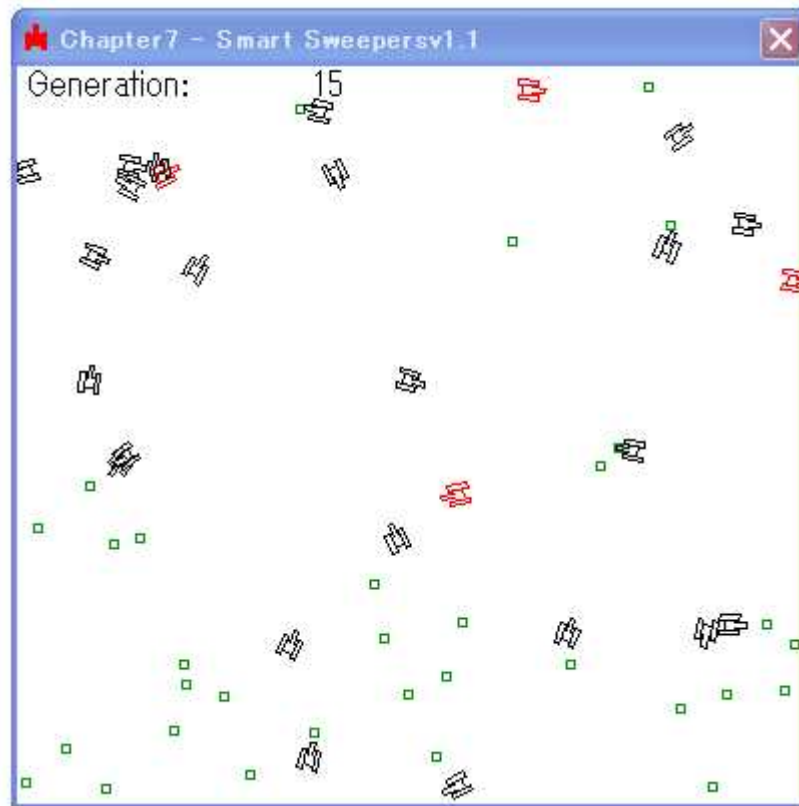
ニューラルネットワークの構造が進化させる「NEAT」の技術

リアルタイム「NEAT」を使ったシミュレーションゲーム「NERO」

GA x NN

ニューラルネットで動く「機雷除去戦車」

Mat Buckland, Chapter 7, AI techniques for game programming, Premier Press, 2002
(実行ファイルとソースコードがCD-ROMにあります)



左のキャタピラの回転数

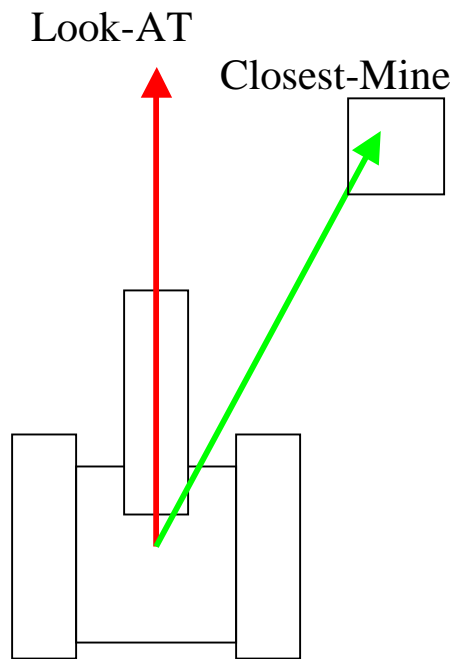
右のキャタピラの回転数

2つの回転数の関係で動く

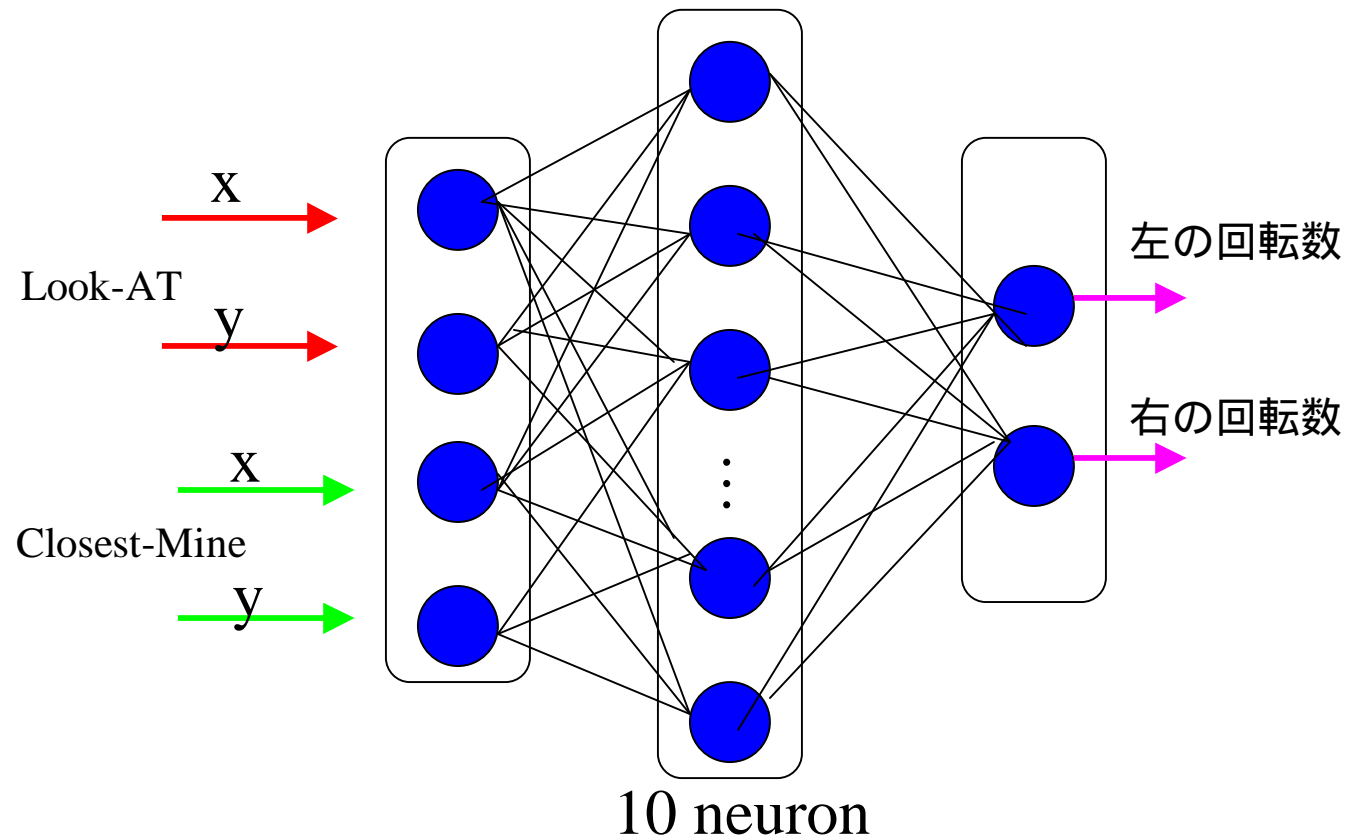
ニューラルネットで動く「機雷除去戦車」

Mat Buckland, Chapter 7, AI techniques for game programming, Premier Press, 2002

自分の向きと、最近接の機雷から、機体制御に必要な左右のキャタピラの回転数を決定する。

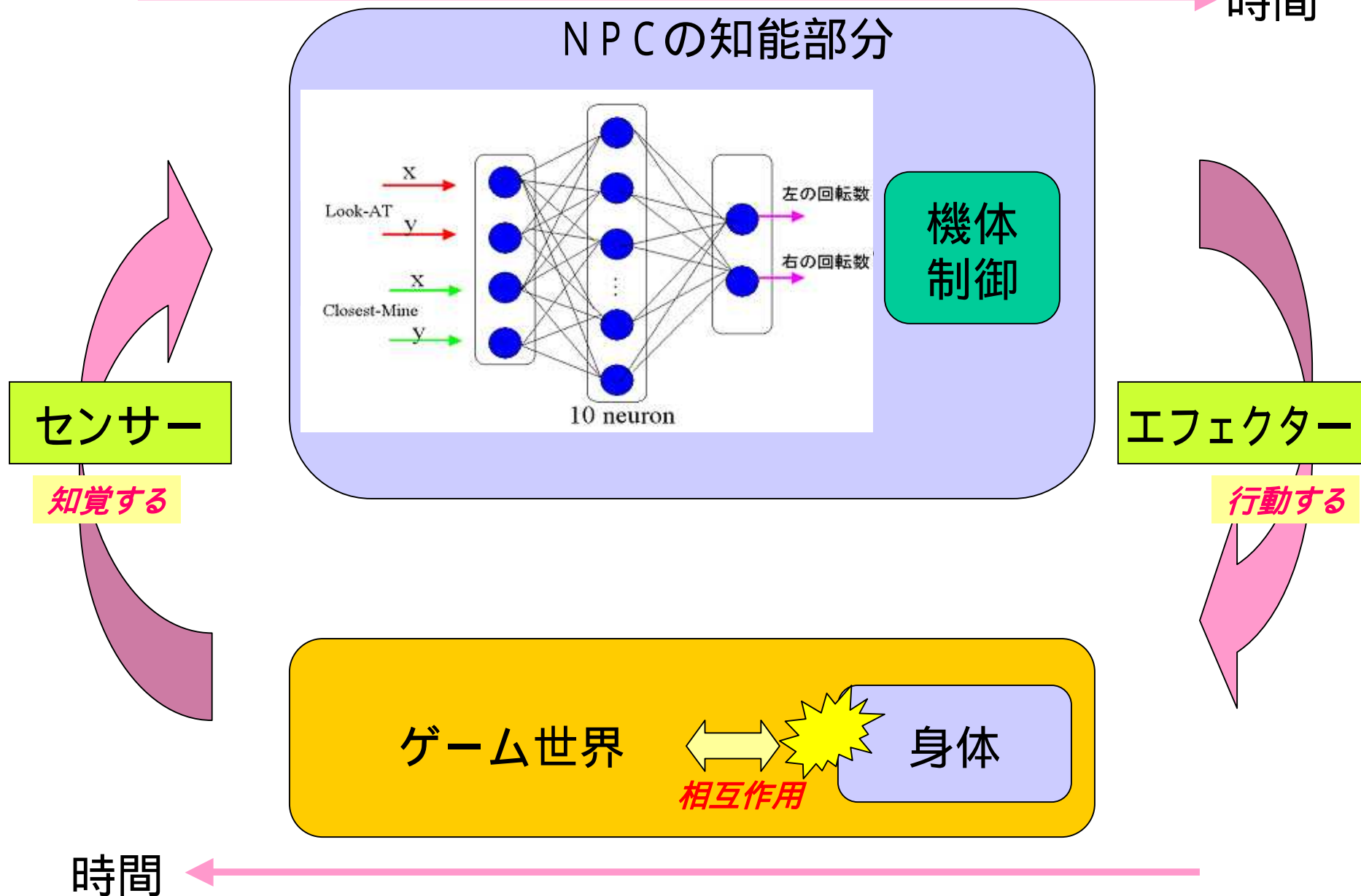


2つのベクトルは
規格化しておく

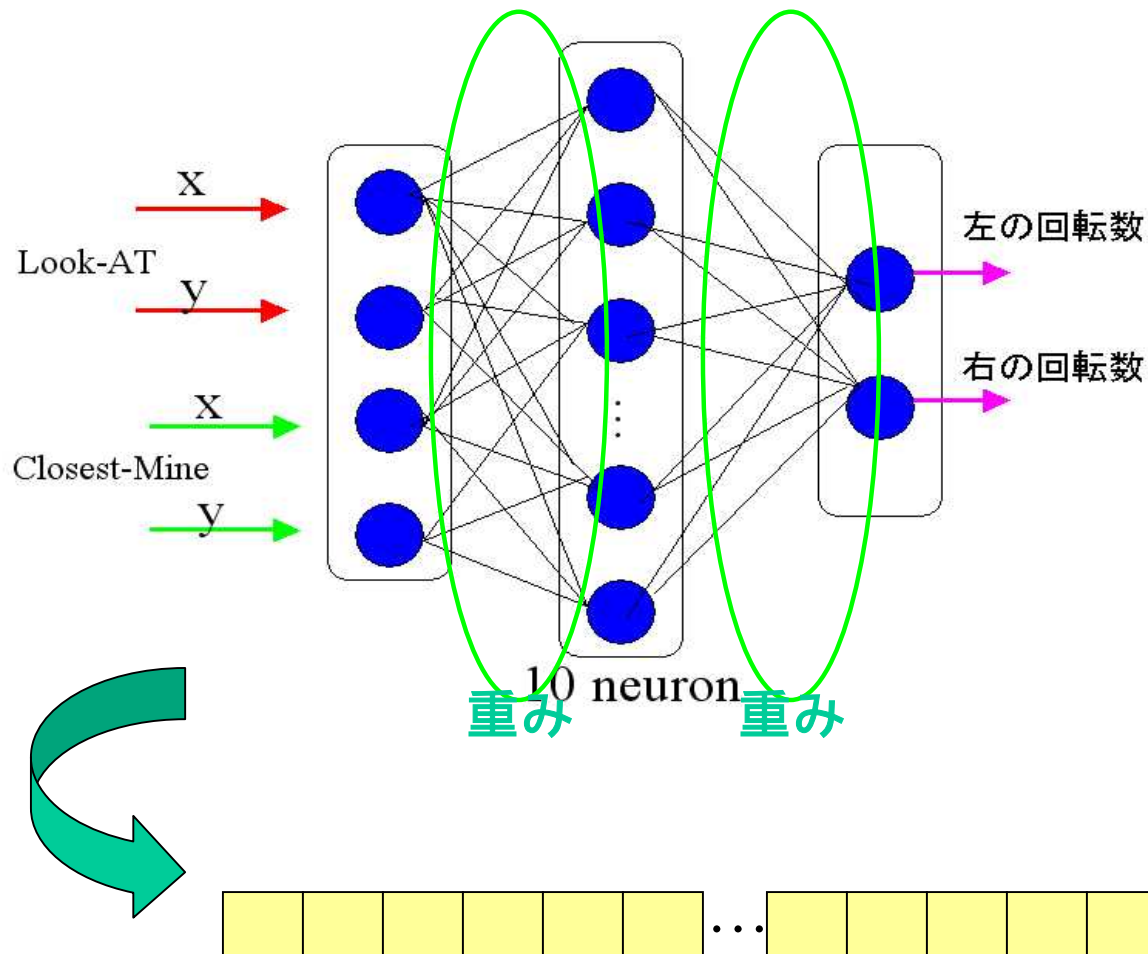


エージェント・アーキテクチャ

時間 →

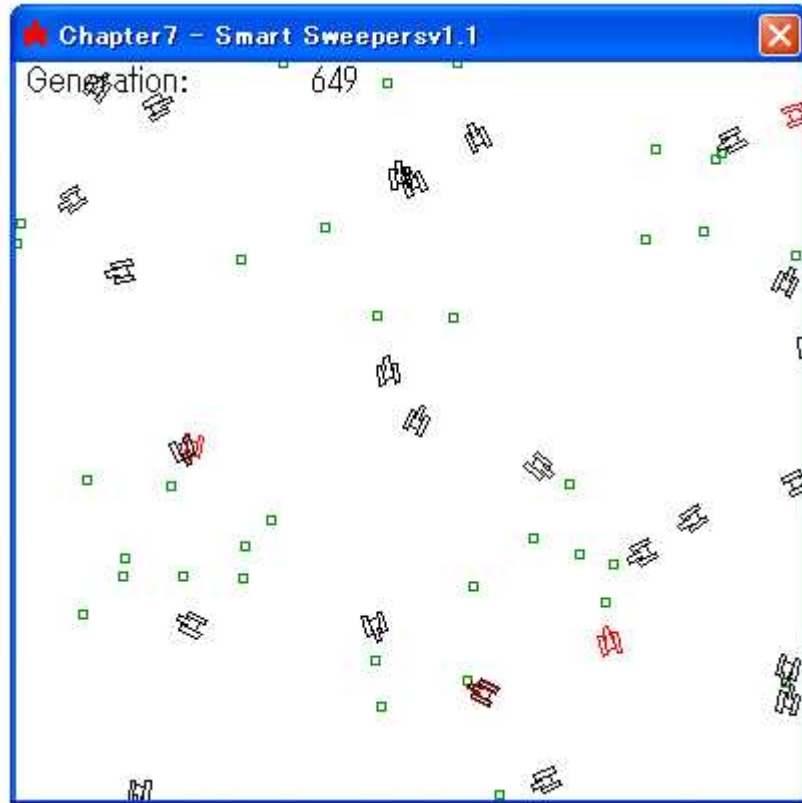


遺伝的アルゴリズムとの併用



重みの実数を $(4 \times 10 + 10 \times 2)$ 要素として遺伝子コードを定義する。

全体の仕組み



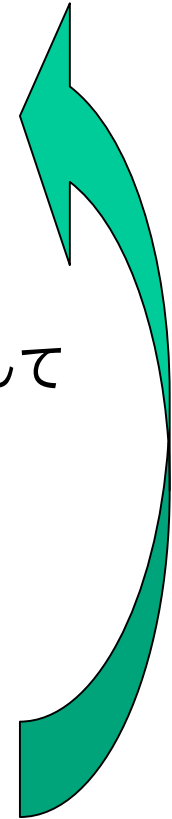
一定時間シミュレーション



機雷除去の戦績を評価関数として
遺伝的アルゴリズムによって、
遺伝子を掛け合わせる。



次の世代を生成

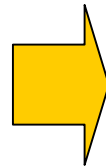
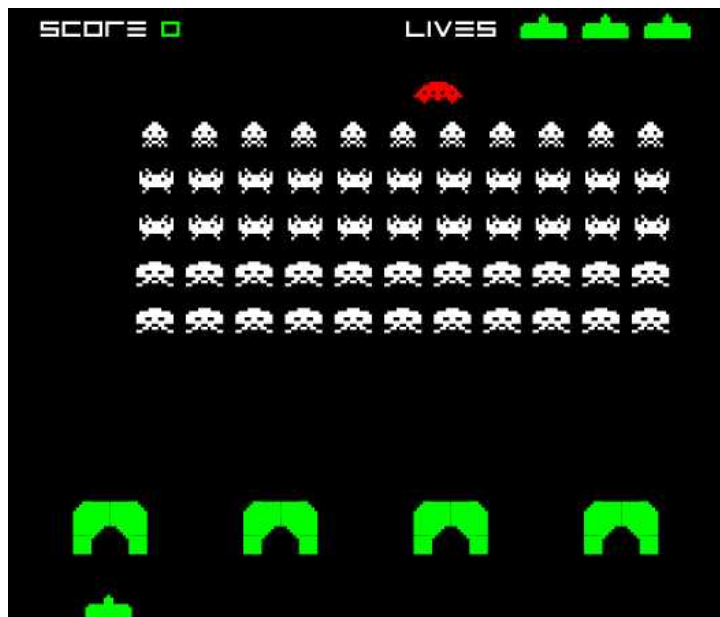


ゲーム内で活動させて成績を見ることで、だんだんと
優秀な個体だけを残して、新しい個体を生成して行く。

2D-シューティングゲームへの応用「バイナリーエイリアン」

Mat Buckland, Chapter 10, AI techniques for game programming, Premier Press, 2002
(実行ファイルとソースコードがCD-ROMにあります)

初期のアーケードゲーム



エイリアン一体一体が、
ニューラルネットの思考を持つ。

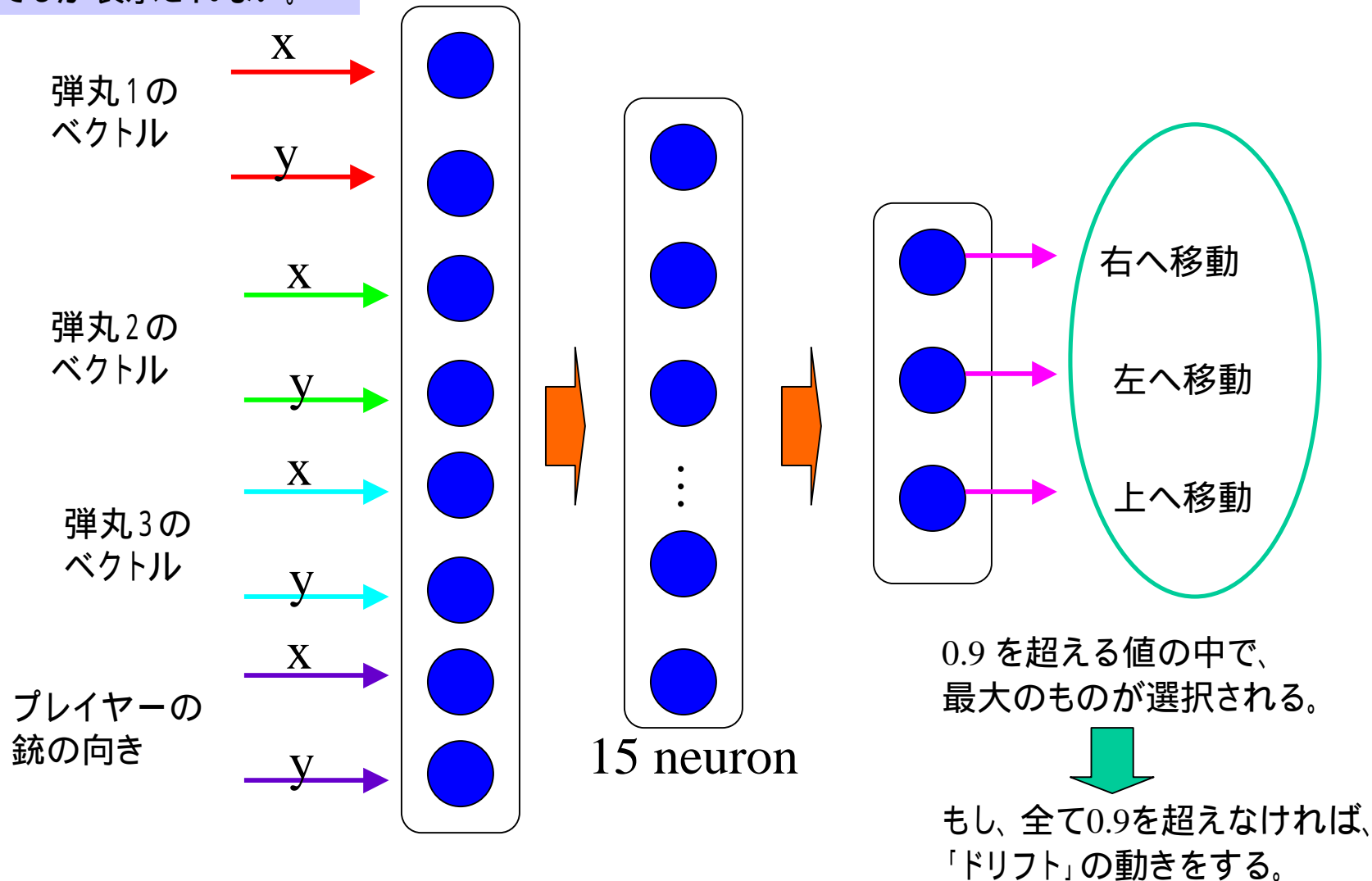


2D-シューティングゲームへの応用「バイナリーエイリアン」

Mat Buckland, Chapter 10, AI techniques for game programming, Premier Press, 2002

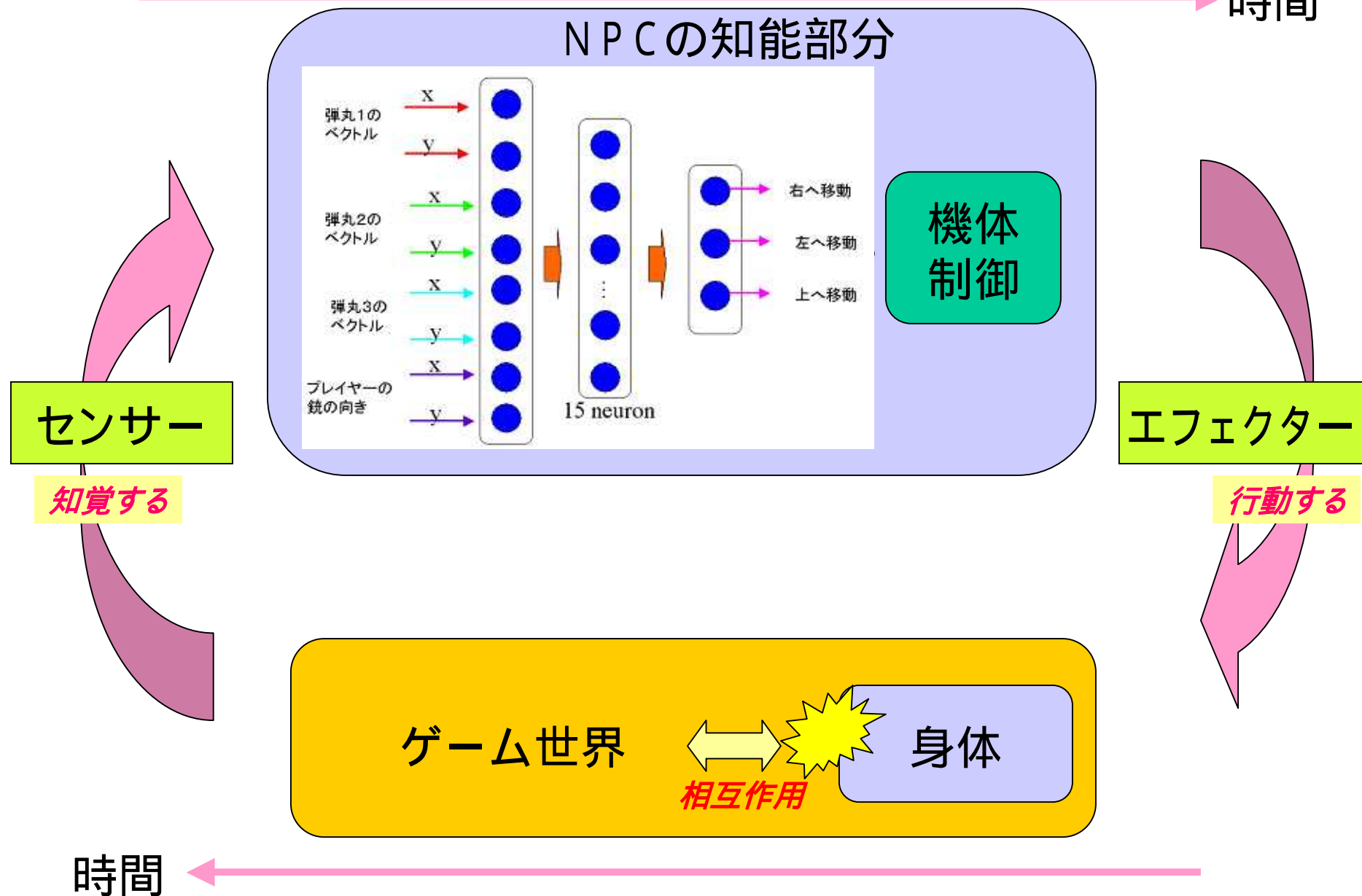
(実行ファイルとソースコードがCD-ROMにあります)

このゲームでは、スクリーンに
3弾までしか表示されない。



エージェント・アーキテクチャ

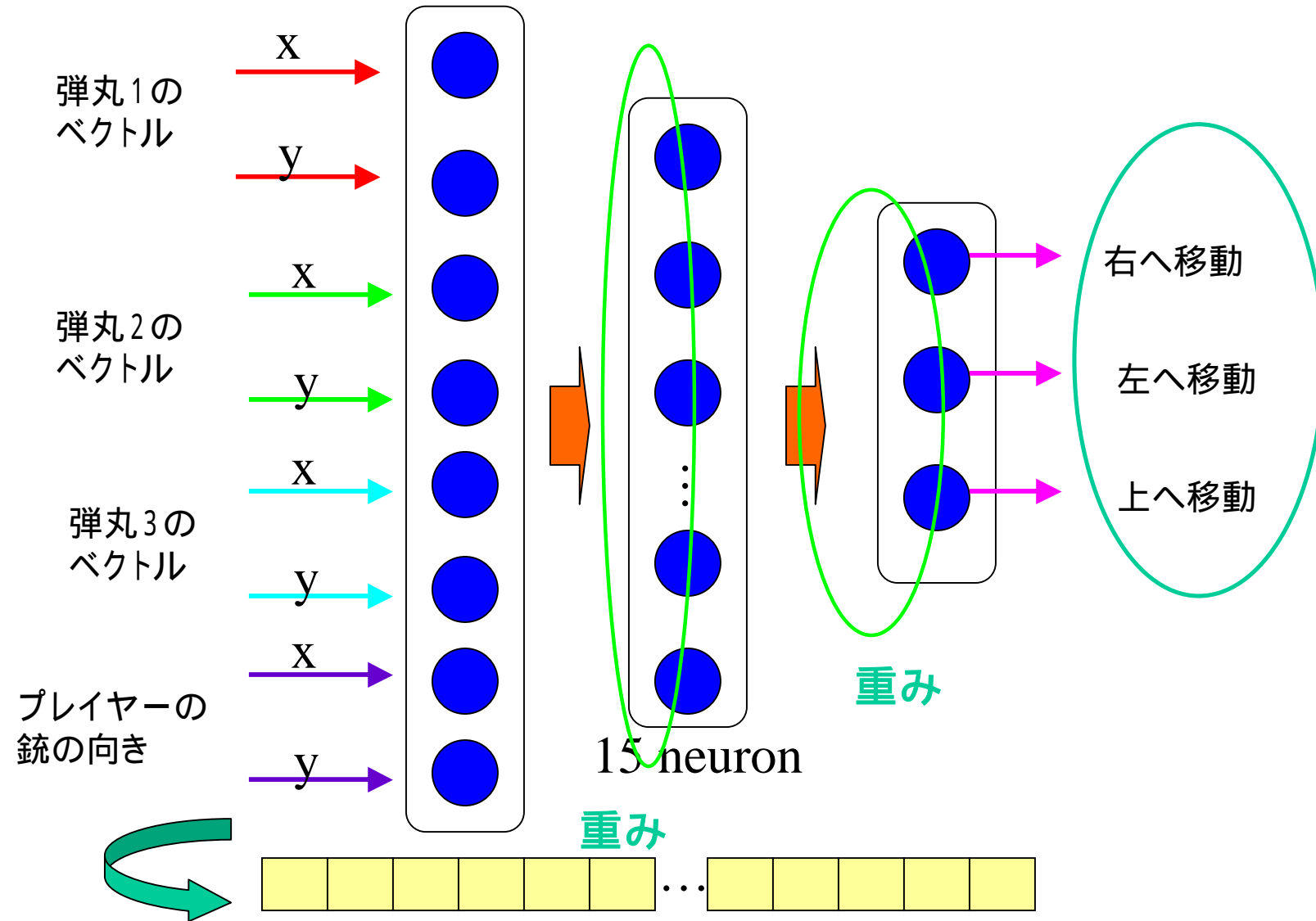
時間 →



2D-シューティングゲームへの応用「バイナリーエイリアン」

Mat Buckland, Chapter 10, AI techniques for game programming, Premier Press, 2002

(実行ファイルとソースコードがCD-ROMにあります)



重みの実数を要素として遺伝子コードを定義する。

全体の仕組み

エイリアンは200体プールしておく。
この数は、ゲーム中、再生産によって常に維持される。



リアルタイムシミュレーション



遺伝的アルゴリズムによって
やっつけられたエイリアンを補充し、
既存のエイリアンに突然変異を施す。



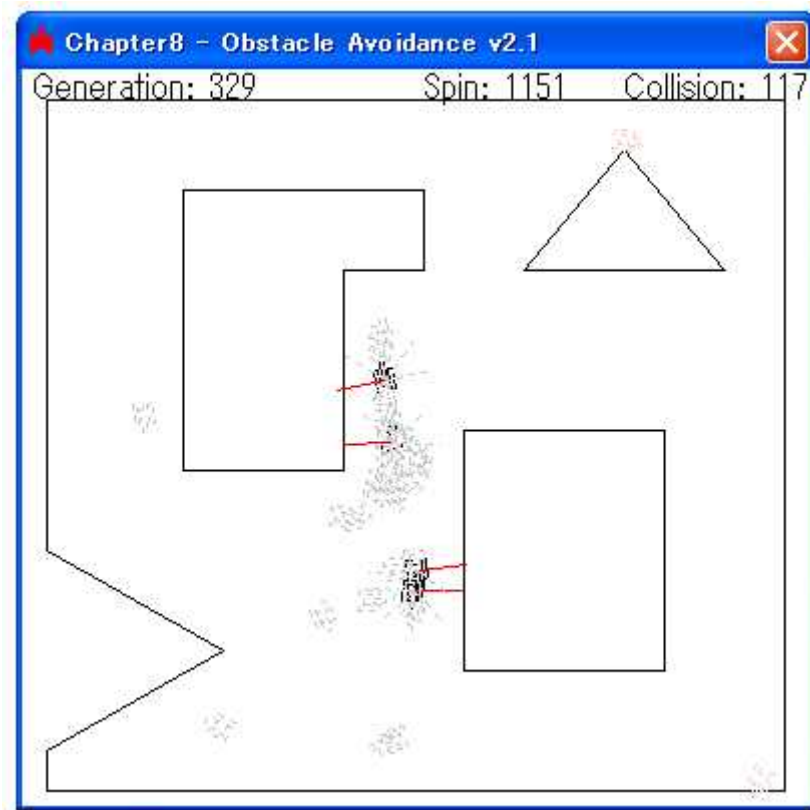
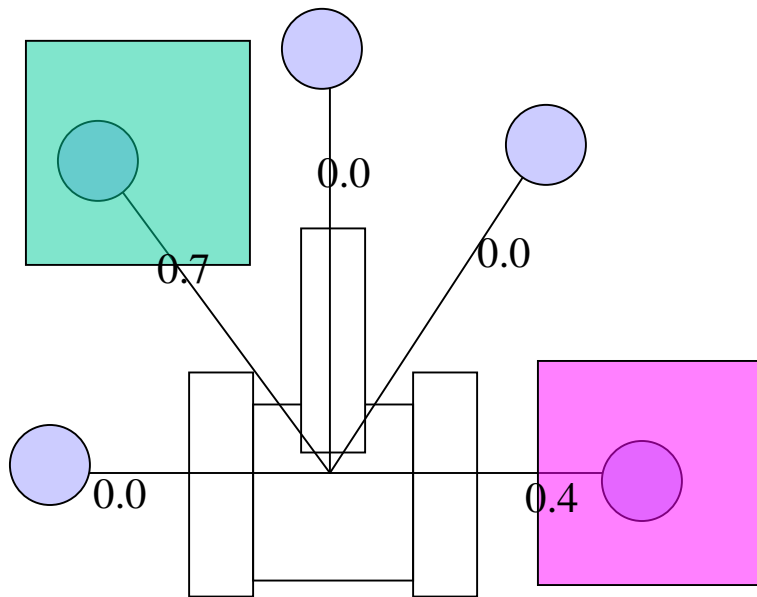
リアルタイム進化

撃墜されたエイリアンの補充は、生き残っているエイリアンの優秀さベスト20%の母集団からトーナメント方式(n 個(ここでは10体)をランダムに選んで、その中から最も優秀な個体をコピーして新しい個体を生成する)で行う。

衝突回避と探索を行う戦車

Mat Buckland, Chapter 8, AI techniques for game programming, Premier Press, 2002
(実行ファイルとソースコードがCD-ROMにあります)

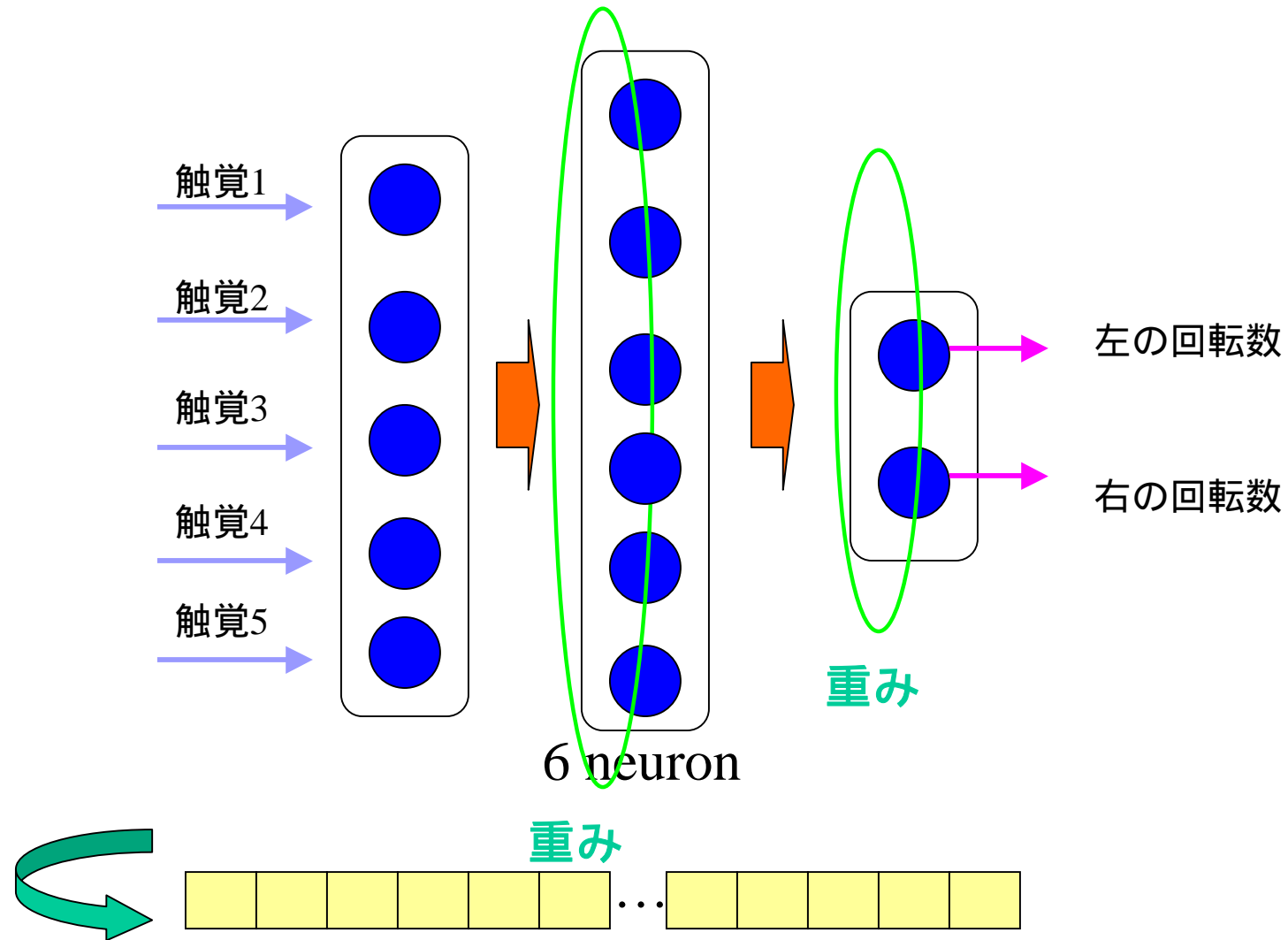
ニューラルネットで衝突を回避



衝突回避のために仮想的な触覚を持つ。
触覚が衝突した場合に、衝突位置を検出する。

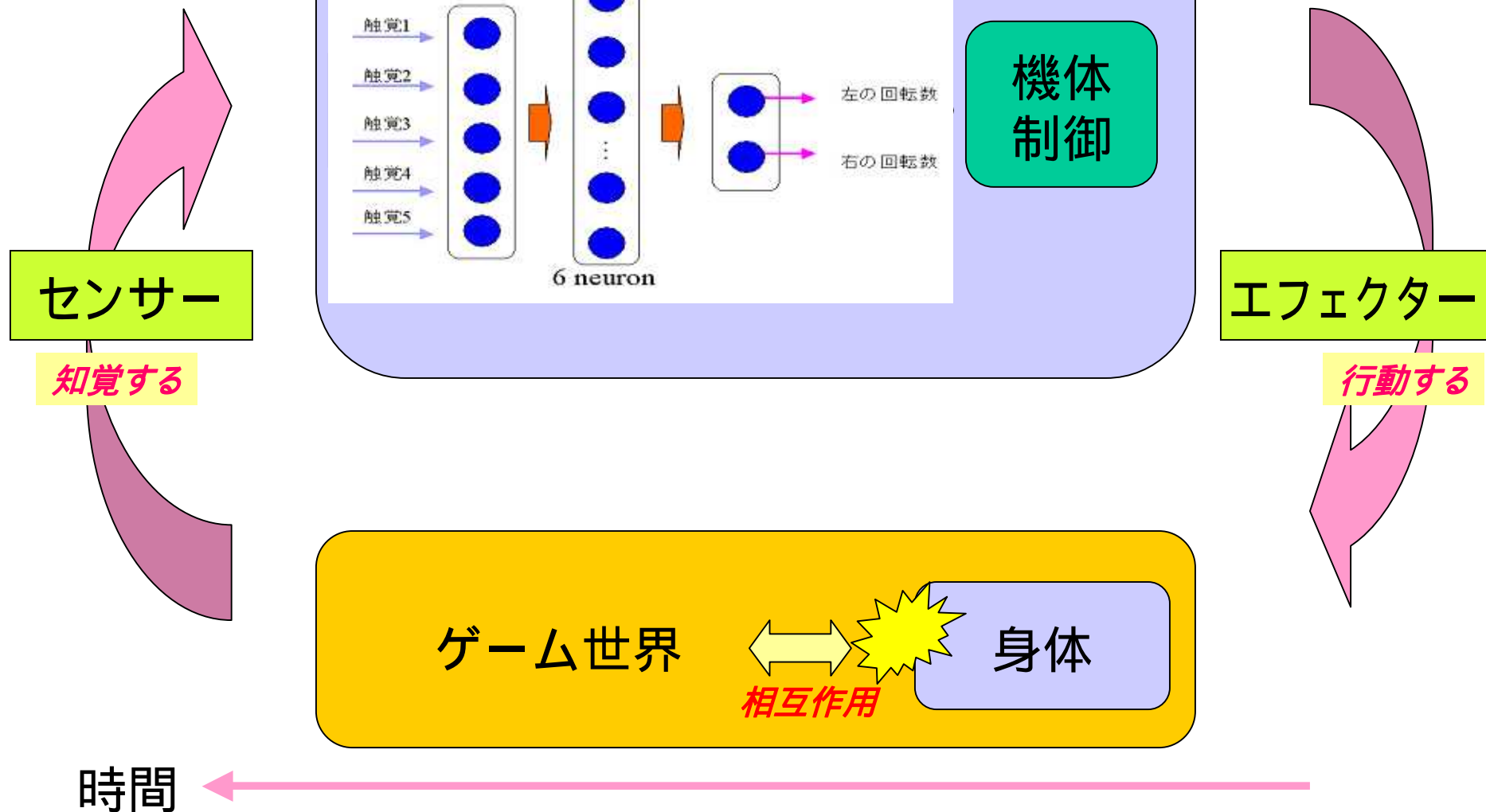
衝突回避を行う戦車

Mat Buckland, Chapter 8, AI techniques for game programming, Premier Press, 2002
(実行ファイルとソースコードがCD-ROMにあります)

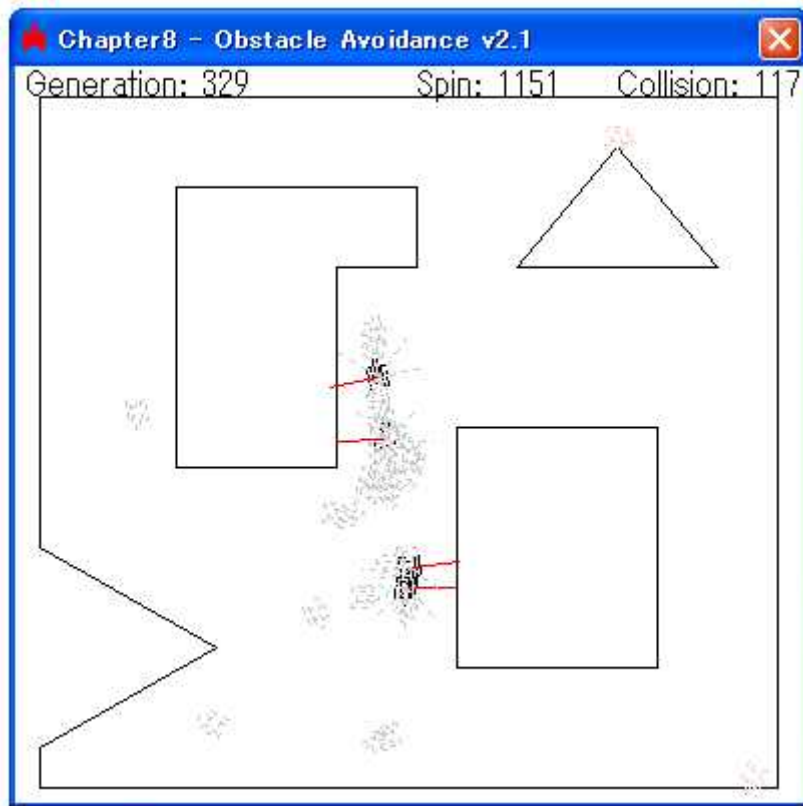


エージェント・アーキテクチャ

時間 →



全体の仕組み



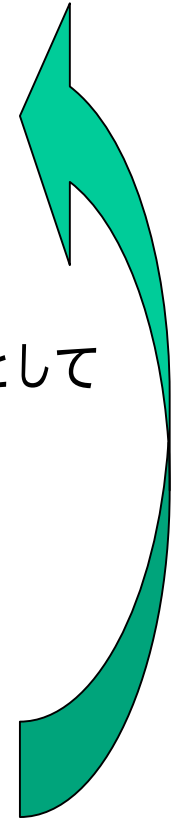
一定時間シミュレーション



衝突の少なさ指標を評価関数として
遺伝的アルゴリズムによって、
遺伝子を掛け合わせる。



次の世代を生成

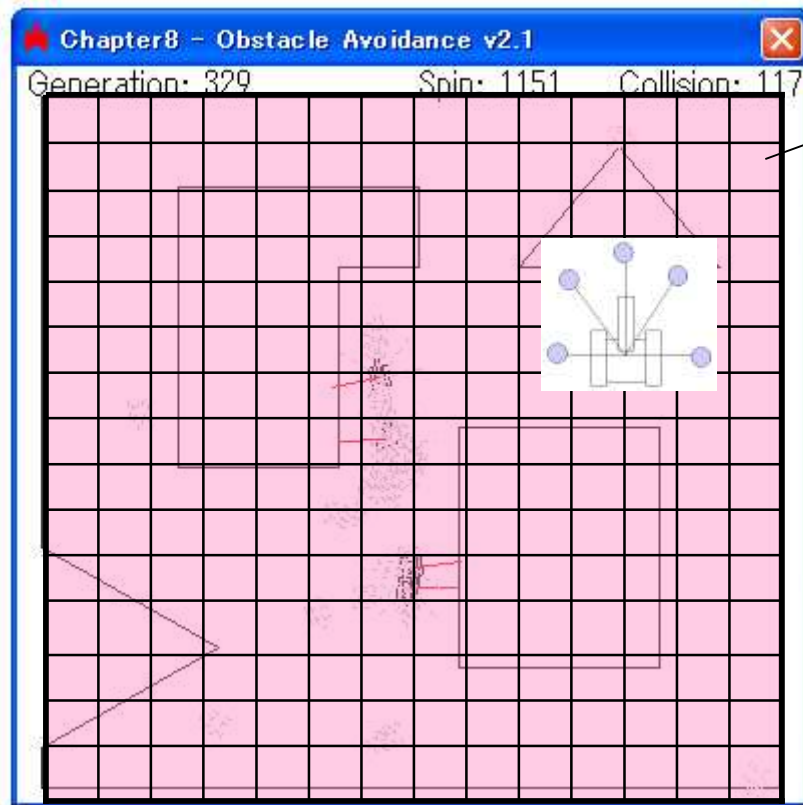


ゲーム内で活動させて成績を見ることで、だんだんと
優秀な個体だけを残して、新しい個体を生成して行く。

衝突回避と探索を行う戦車

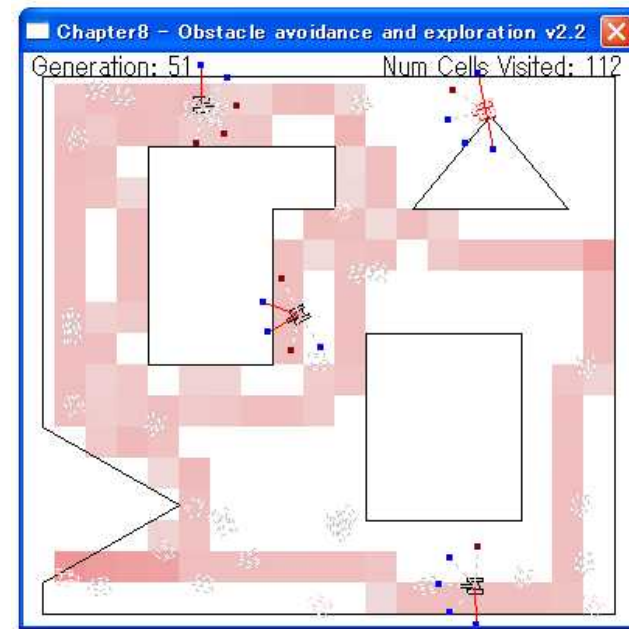
Mat Buckland, Chapter 8, AI techniques for game programming, Premier Press, 2002
(実行ファイルとソースコードがCD-ROMにあります)

探索



NPCを、なるべく未踏野の領域に行くように移動させたい。

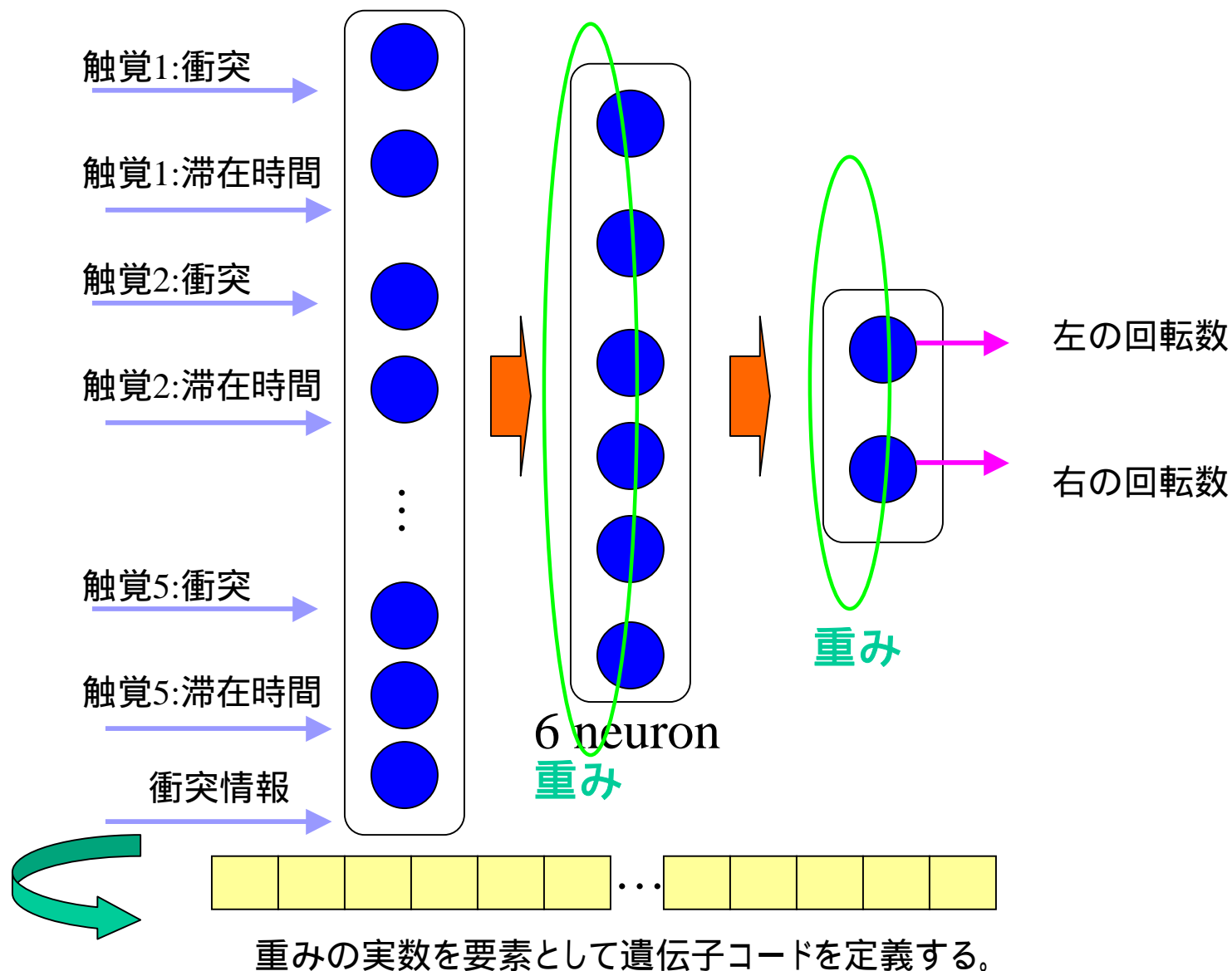
各マスに、NPCがいた通算の時間を規格化(未踏野 - 1、通算時間は0 ~ 1で規格化)して記憶する。



触覚は、自分のいるマスの通算時間を読み取る。

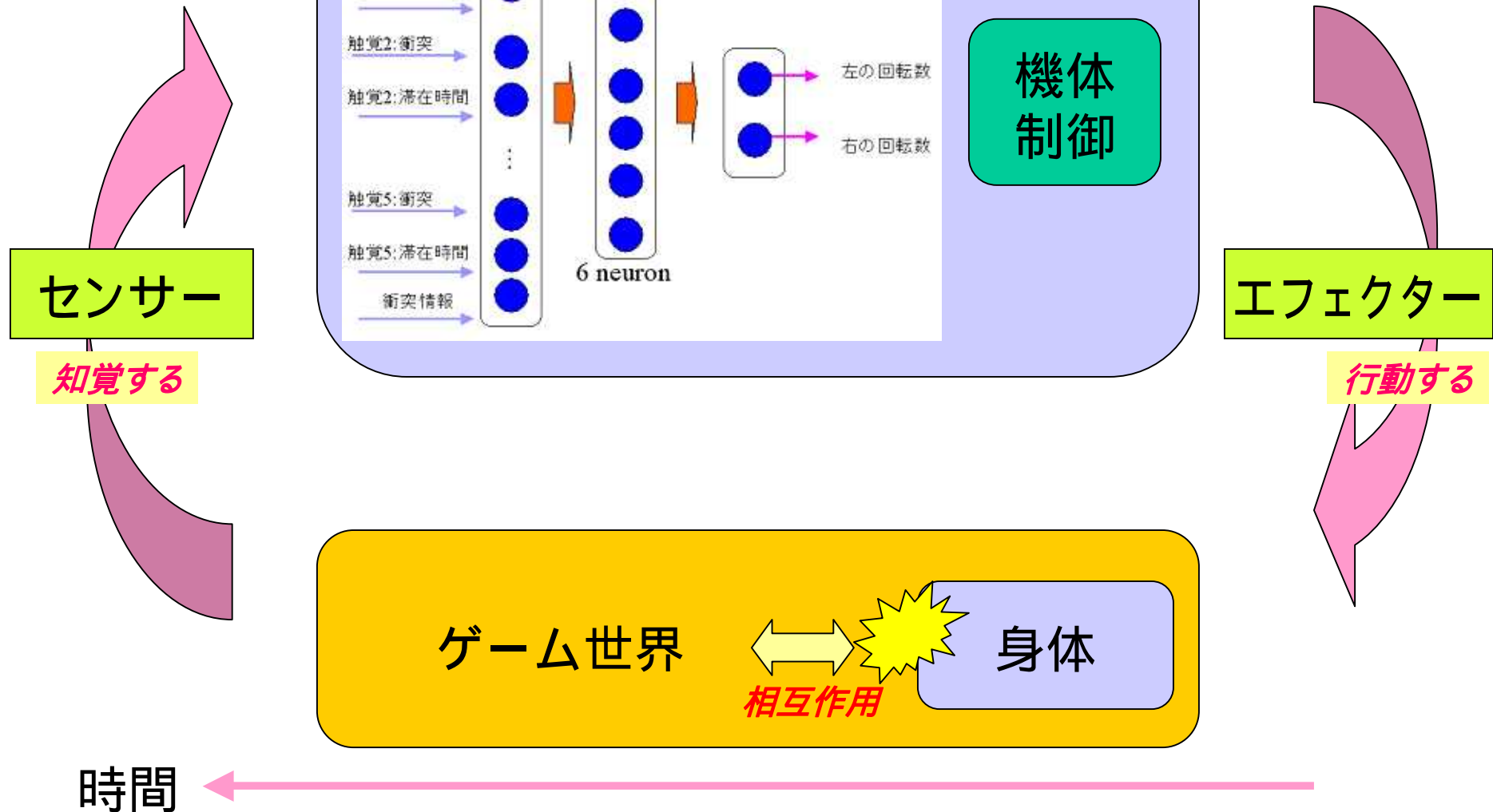
衝突回避と探索を行う戦車

Mat Buckland, Chapter 8, AI techniques for game programming, Premier Press, 2002
(実行ファイルとソースコードがCD-ROMにあります)

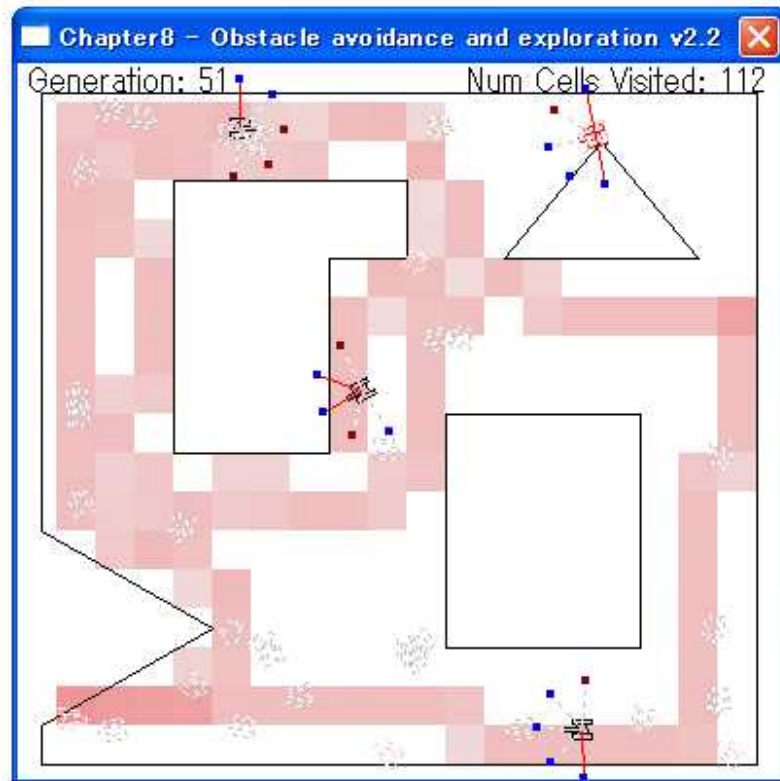


エージェント・アーキテクチャ

時間 →



全体の仕組み



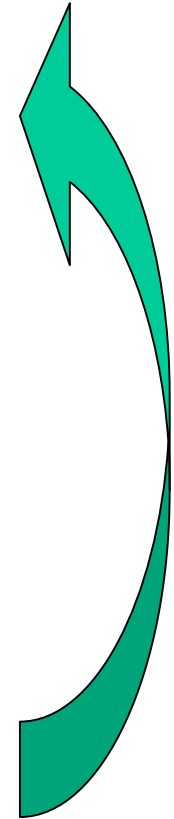
一定時間シミュレーション



探索能力を評価関数として
遺伝的アルゴリズムによって、
遺伝子を掛け合わせる。



次の世代を生成



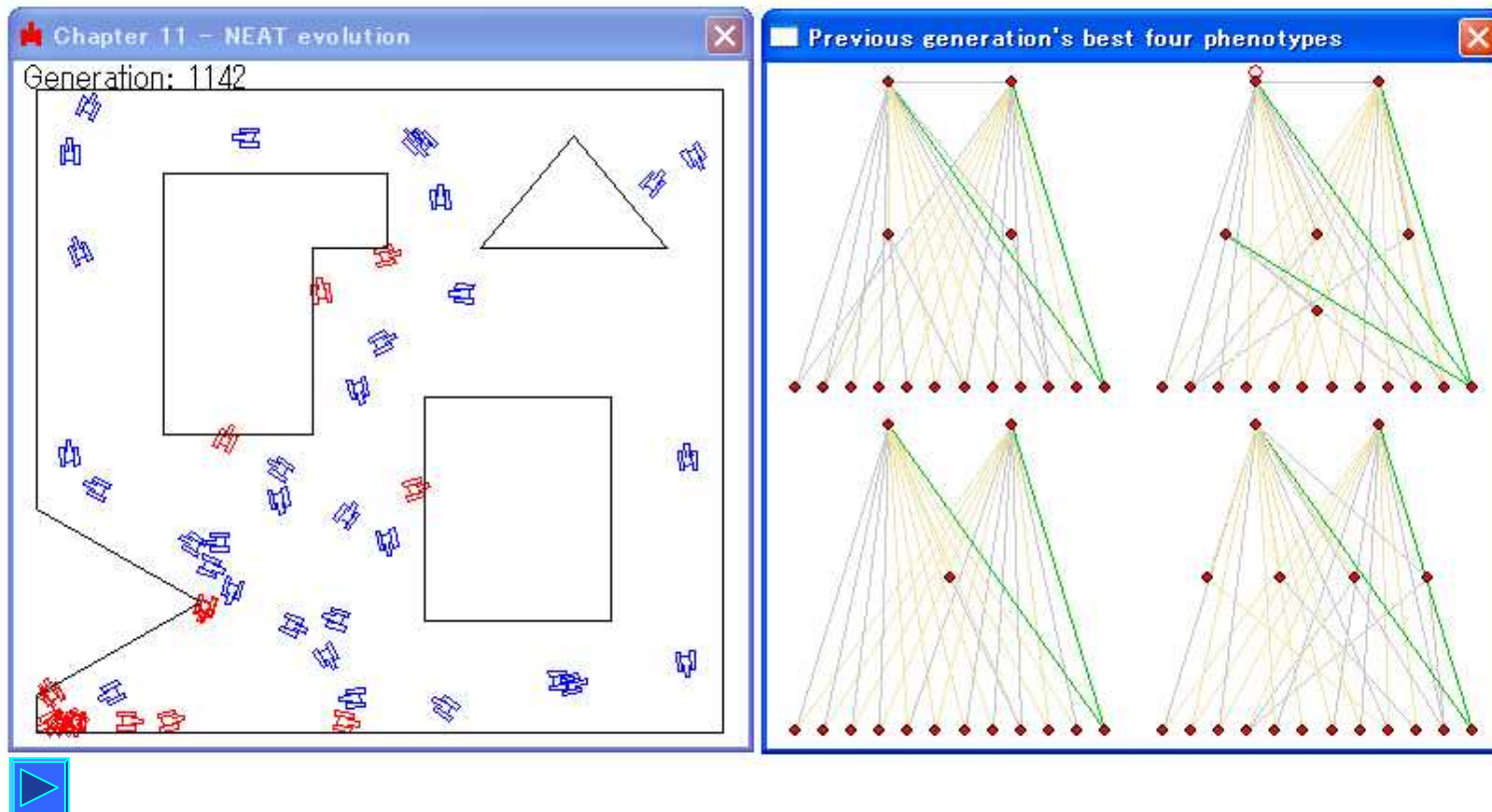
ゲーム内で活動させて成績を見ることで、だんだんと
優秀な個体だけを残して、新しい個体を生成して行く。

ニューラルネットワークの構造が進化させる 「NEAT」の技術

Mat Buckland, Chapter 11, AI techniques for game programming, Premier Press, 2002
(実行ファイルとソースコードがCD-ROMにあります)

これまでニューラルネットは、最初に構造を定義した後は変化しなかった。

➡ 動的にニューラルネットの構造を変化させる技術
Neuron Evolution of Augmenting Topologies (NEAT)

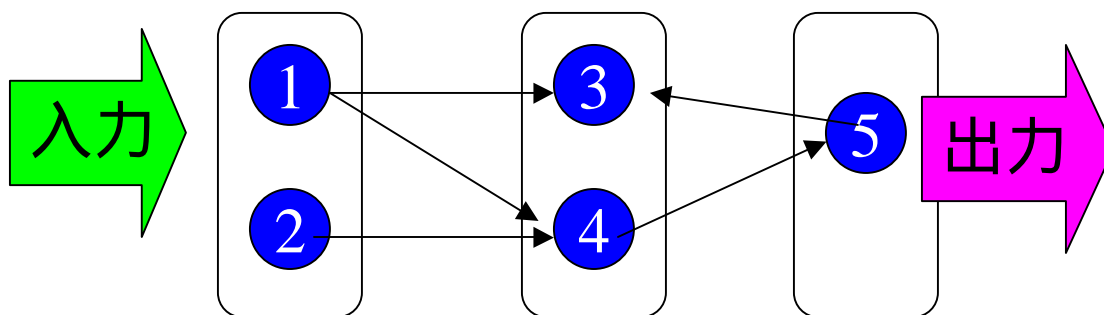


NEAT

回路の構成を遺伝子コードで表現する。

Weight: 1.2	Weight: -3	Weight: 0.7	Weight: -2.1	Weight: 1.1	Weight: 0.8	Weight: -1
From: 1	From: 1	From: 2	From: 3	From: 3	From: 4	From: 5
To: 3	To: 4	To: 4	To: 4	To: 5	To: 5	To: 3
Enabled: Y	Enabled: Y	Enabled: Y	Enabled: Y	Enabled: N	Enabled: Y	Enabled: Y
Recurrent: N	Recurrent: N	Recurrent: N	Recurrent: N	Recurrent: N	Recurrent: N	Recurrent: Y
Innovation: 1	Innovation: 6	Innovation: 2	Innovation: 6	Innovation: 3	Innovation: 4	Innovation: 7

つなぎ方を定義する遺伝子



ニューロンを定義する遺伝子

ID: 1	ID: 2	ID: 3	ID: 4	ID: 5
Type: Input	Type: Input	Type: hidden	Type: hidden	Type: Output

NEAT

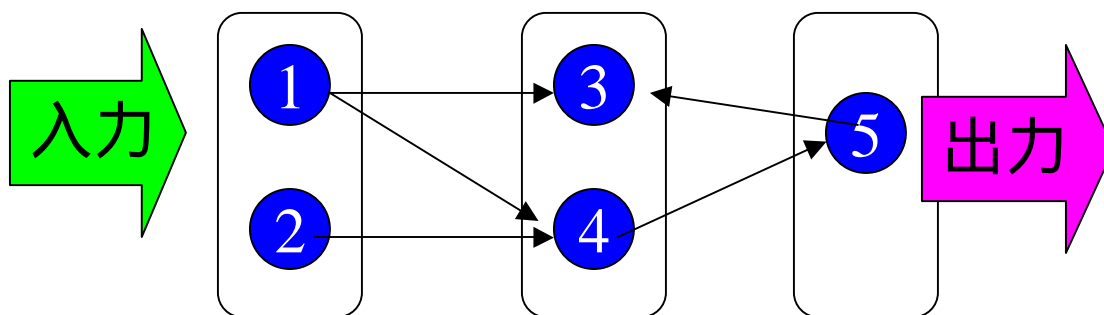
回路の構成を遺伝子コードで表現する。

無効

Weight: 1.2	Weight: -3	Weight: 0.7	Weight: -2.1	Weight: 1.1	Weight: 0.8	Weight: -1
From: 1	From: 1	From: 2	From: 3	From: 3	From: 4	From: 5
To: 3	To: 4	To: 4	To: 4	To: 5	To: 5	To: 3
Enabled: Y	Enabled: Y	Enabled: Y	Enabled: Y	Enabled: N	Enabled: Y	Enabled: Y
Recurrent: N	Recurrent: N	Recurrent: N	Recurrent: N	Recurrent: N	Recurrent: N	Recurrent: Y
Innovation: 1	Innovation: 6	Innovation: 2	Innovation: 6	Innovation: 3	Innovation: 4	Innovation: 7

リンク(つなぎ方)を定義する遺伝子

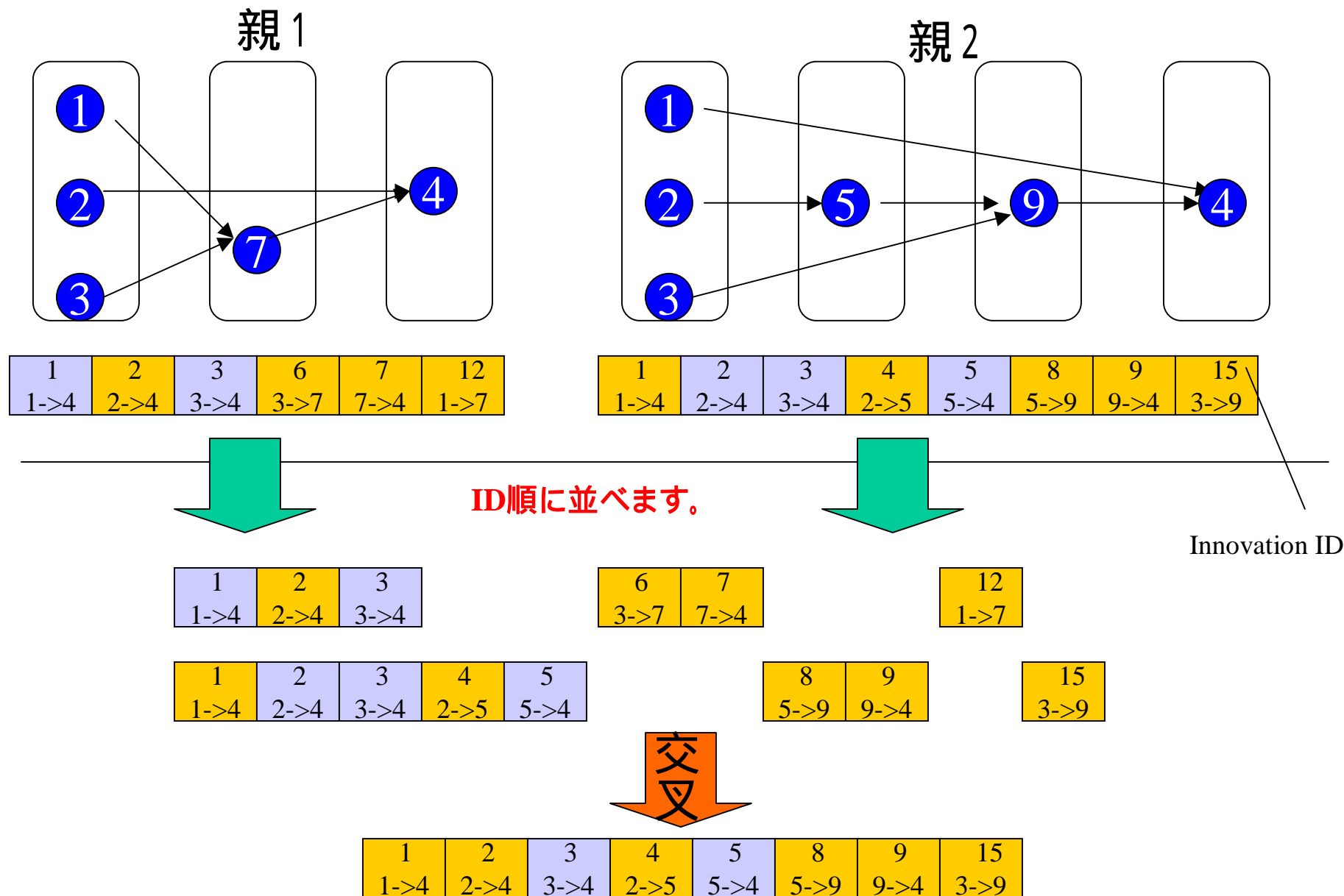
Innovation ID によってリンク、ニューロンを全遺伝子共通の管理する。



ニューロンを定義する遺伝子

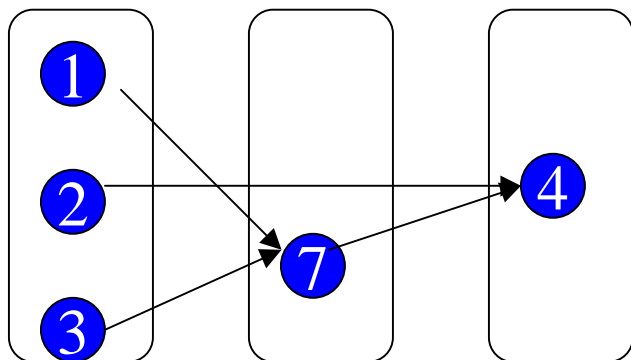
ID: 1	ID: 2	ID: 3	ID: 4	ID: 5
Type: Input	Type: Input	Type: hidden	Type: hidden	Type: Output

NEATにおける交叉



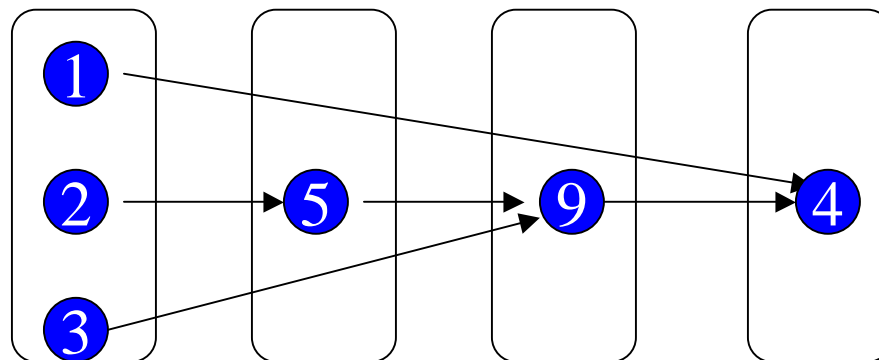
NEATにおける交叉

親 1



1	2	3	6	7	12
1->4	2->4	3->4	3->7	7->4	1->7

親 2

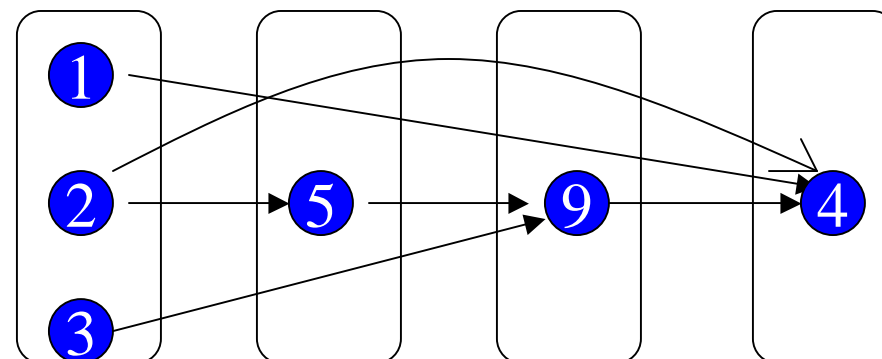


1	2	3	4	5	8	9	15
1->4	2->4	3->4	2->5	5->4	5->9	9->4	3->9

交叉

Innovation ID

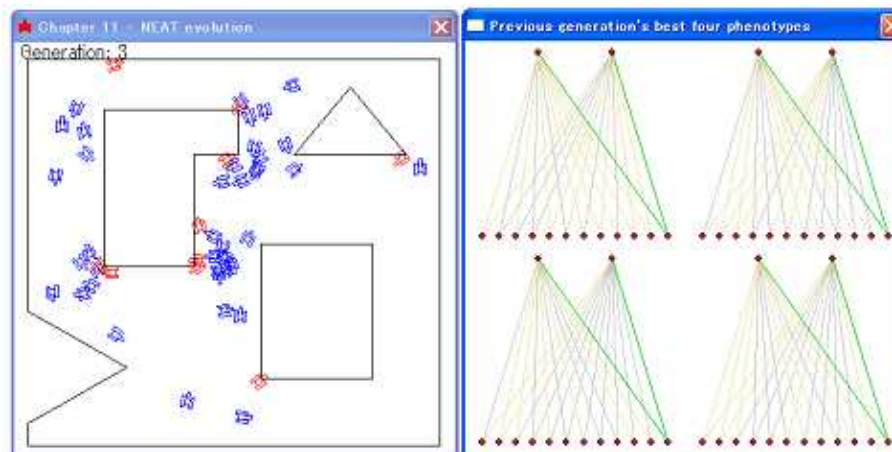
子供 = 新しいニューラルネットワーク



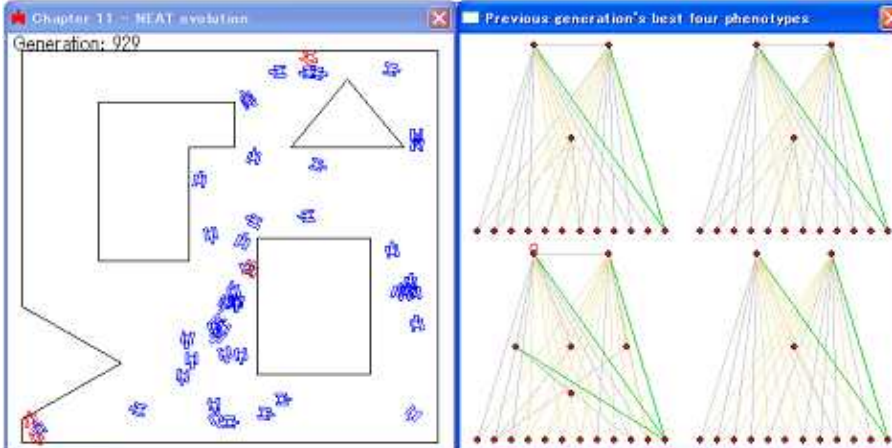
1	2	3	4	5	8	9	15
1->4	2->4	3->4	2->5	5->4	5->9	9->4	3->9

左は俯瞰図(赤は衝突してしまっている)
右は適応度ベスト4のニューラルネット

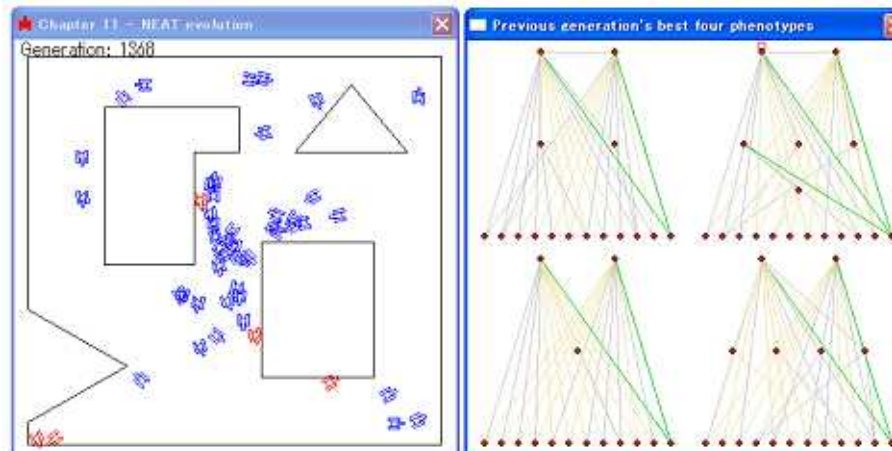
第3世代



第929世代

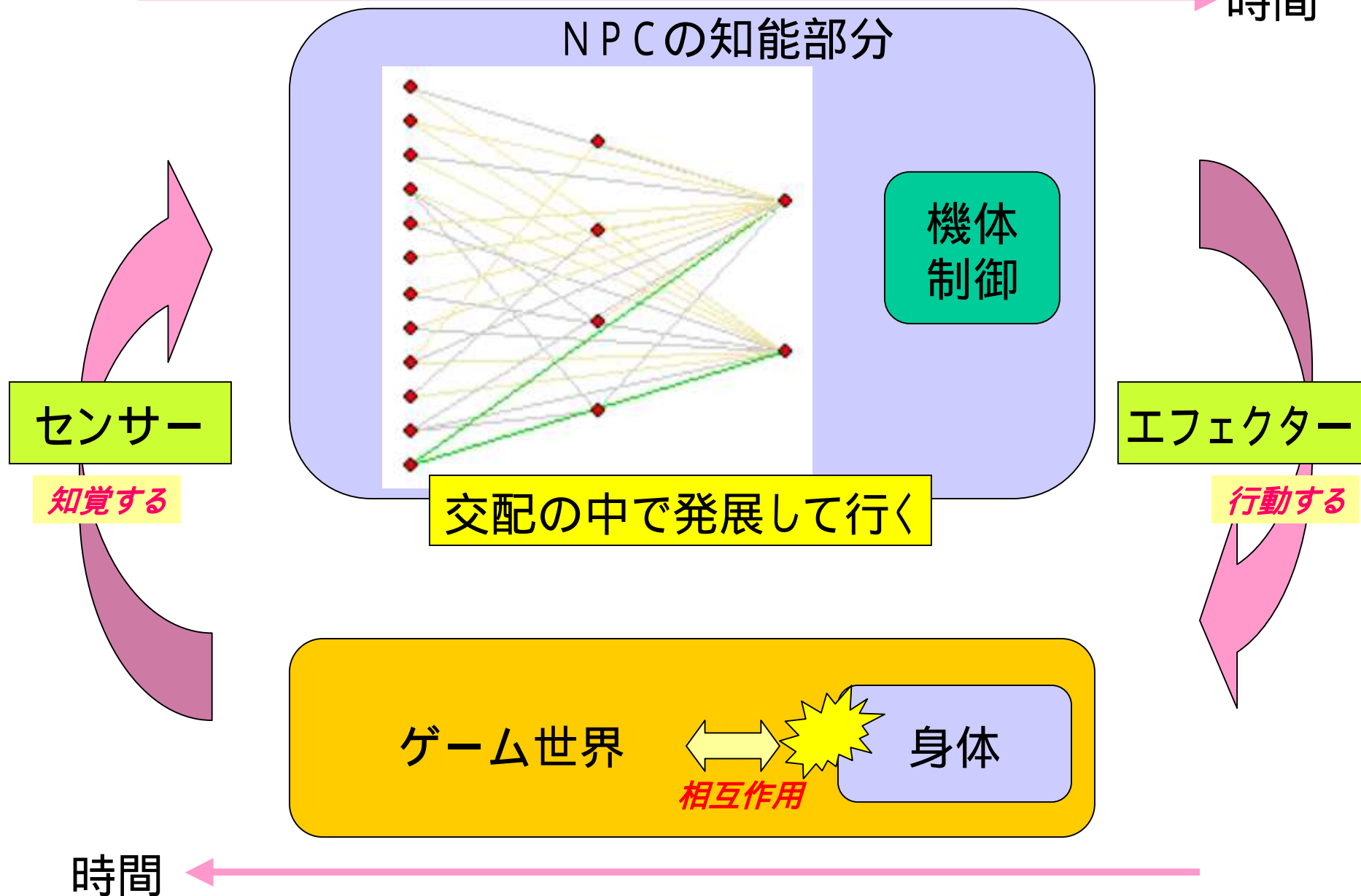


第1368世代



エージェント・アーキテクチャー

▶ 時間



リアルタイム「NEAT」を使ったシミュレーションゲーム 「NERO」

Neural Networks Research Group, Department of Computer Sciences, University of Texas at Austin, Neuro-Evolving Robotic Operatives, <http://www.nerogame.org/>,
(ゲームがあります)

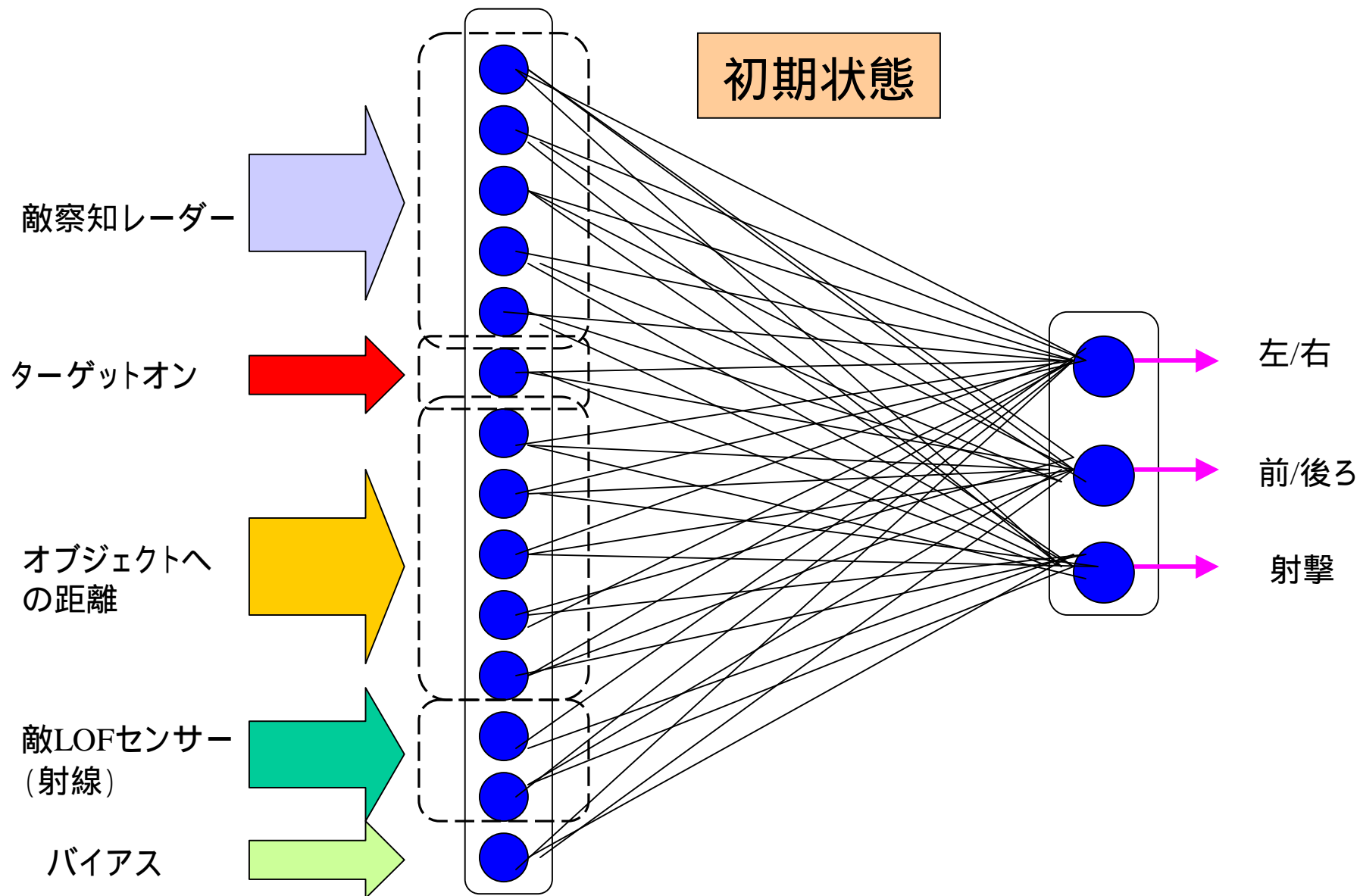


rtNEAT (リアルタイムNEAT) システムの上で、ユーザーが兵士を訓練する。
訓練させた兵士同士を対戦させる。



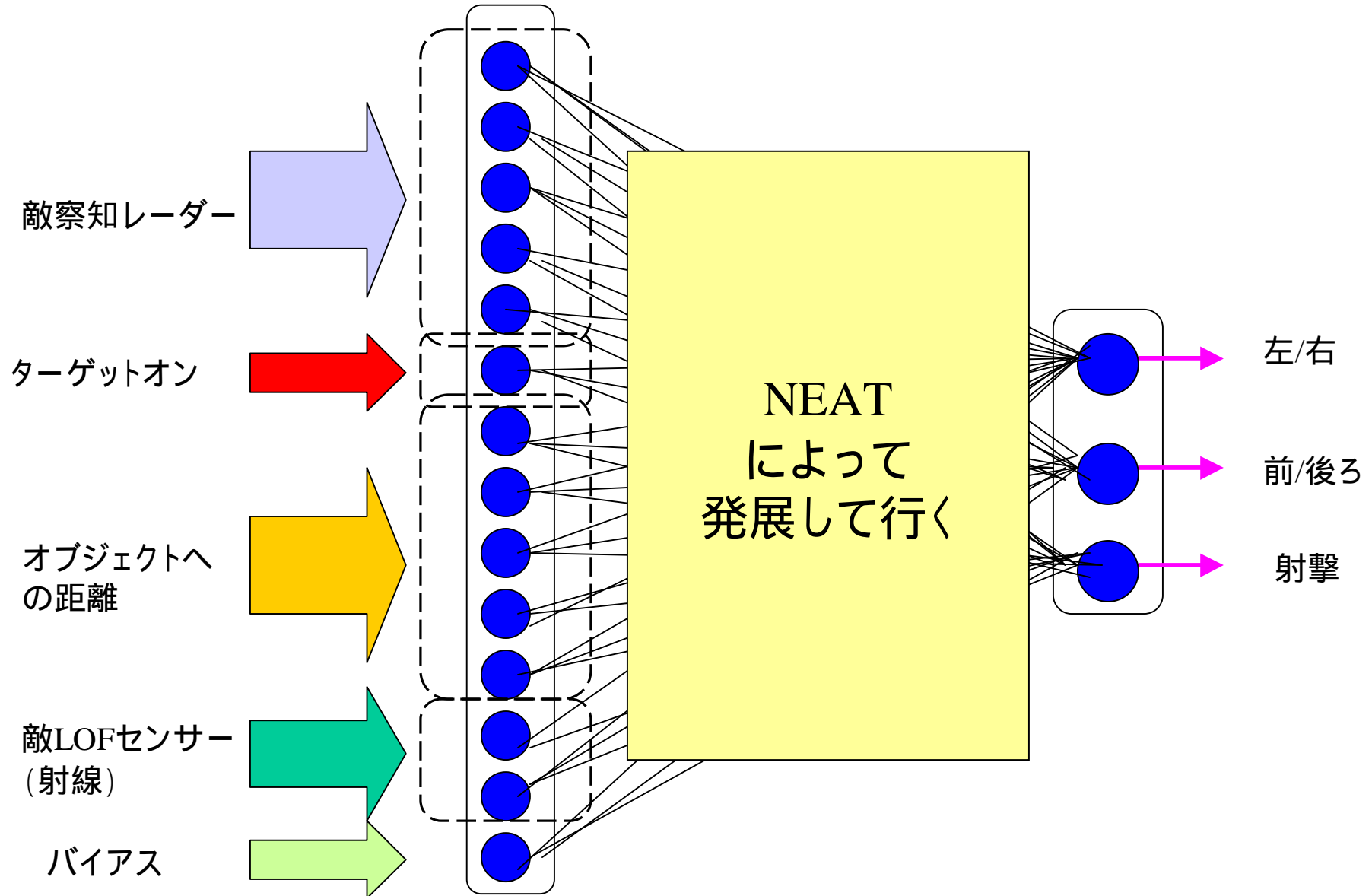
「NERO」

兵士のニューラルネット

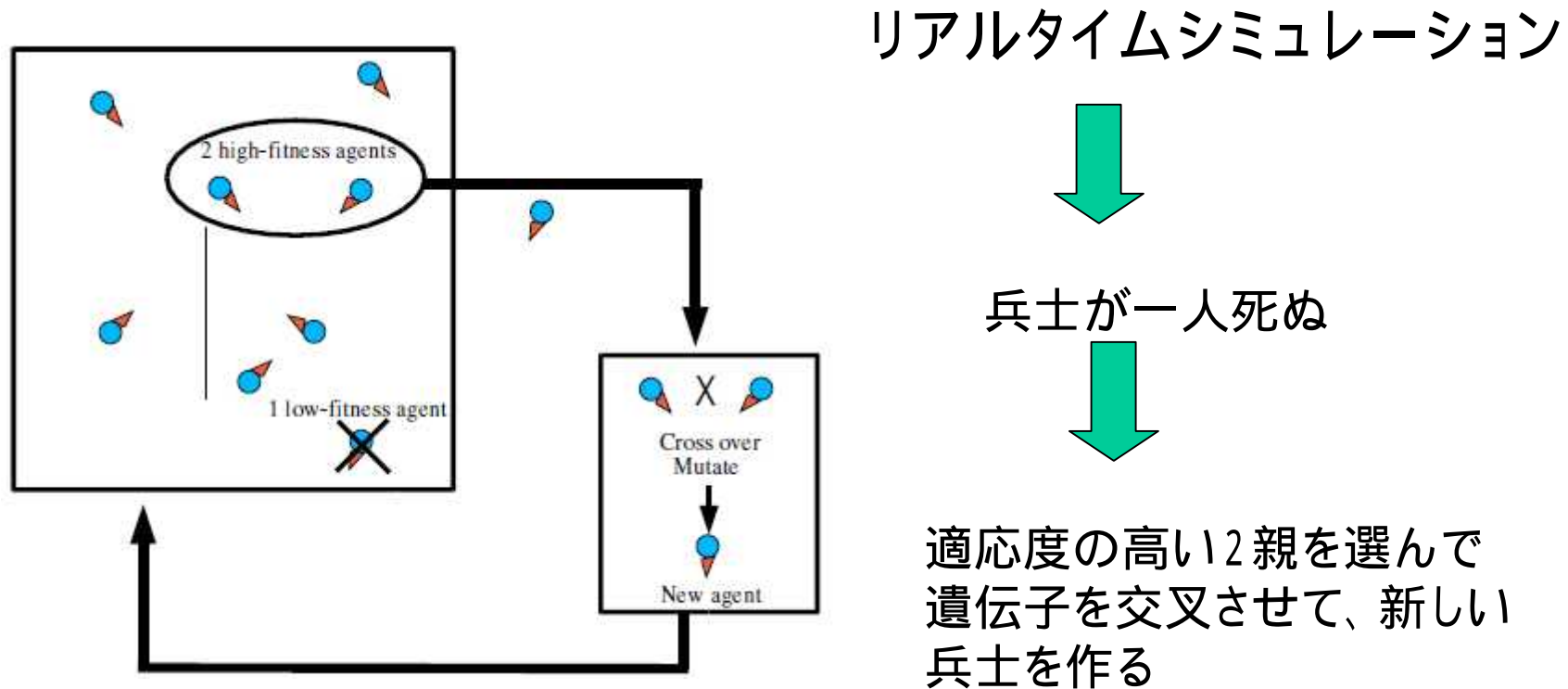


「NERO ver2.0」

兵士のニューラルネット



全体の仕組み



ゲーム内で活動させて成績を見ることで、だんだんと優秀な個体だけを残して、新しい個体を生成して行く。

第4章 まとめ

- (1) ニューラルネットとは、数値の組が入力し、数値の組が出力される。
- (2) 学習法は、誤差伝播法があるが、これは、複雑な回路になればなるほど使いやすすくない。
- (3) 「遺伝的アルゴリズム×ニューラルネットの組み合わせ」はシンプルで有効な方法である。
- (4) ゲームデザインが要である。

第5回セミナーの構成

はじめに

ゲームAI連続セミナーのご紹介

Enjoy AI!

第1部 ゲームAI 技術解説 (100分) (講師:三宅)

第1章 概論 (10分)

第2章 C4アーキテクチャー (20分)

第3章 遺伝的アルゴリズム (30分)

第4章 ニューラルネットワーク (30分)

第5章 発展 (10分)

各章の終わりに質疑応答を行います。

(休憩 20分)

第2部 ディスカッション (2時間) (アイデア:IGDAゲームAI運営委員)

(総合司会 + コーディネート:長久、グループ司会者)

(I) グループ討論 (80分)

(I) まとめ (30分)

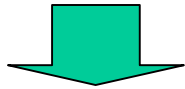
展望

技術的展望

遺伝的アルゴリズム、ニューラルネットは、
既に詳細な研究が為されおり、技術的蓄積や情報、文献も多い。

ゲームデザインの展望

いくつかの例を今回紹介するが、導入例は、多いとは言えない。
これからの分野であり、ゲームデザイナーの手腕にかかっている
ところが大である。



ある程度の**原理**を理解して、その力を利用したゲームデザインを産み出すことが、
この分野の未来、ひいてはデジタルゲームの将来を決定する。

展望

技術導入の展望

まずは技術の上で「遊んでみる」こと

遺伝的アルゴリズム、ニューラルネットは、デモ、サンプルコードが多い(多数紹介)。まずは動かしながらプログラムを発展させ、良いゲームデザインが来るのを待つか、自分で提案すること。



動作させては結果を見て、動作させては結果を見る。気長に実験するゆとりを持つことで、習得でき、いざという時に応用できる。

ゲームデザインの展望

フロンティアを進むための堅実な努力

デモを見て原理を感じとって、そのデモが「しょーもない」と多分思われると思いますが、そこから、自分のゲームデザインを考えたり、全く違う応用を考案する。さらに、先駆者(アストロノーカなど)に学ぶ。



次世代のゲームデザインへ、新しい息吹をゲームに！

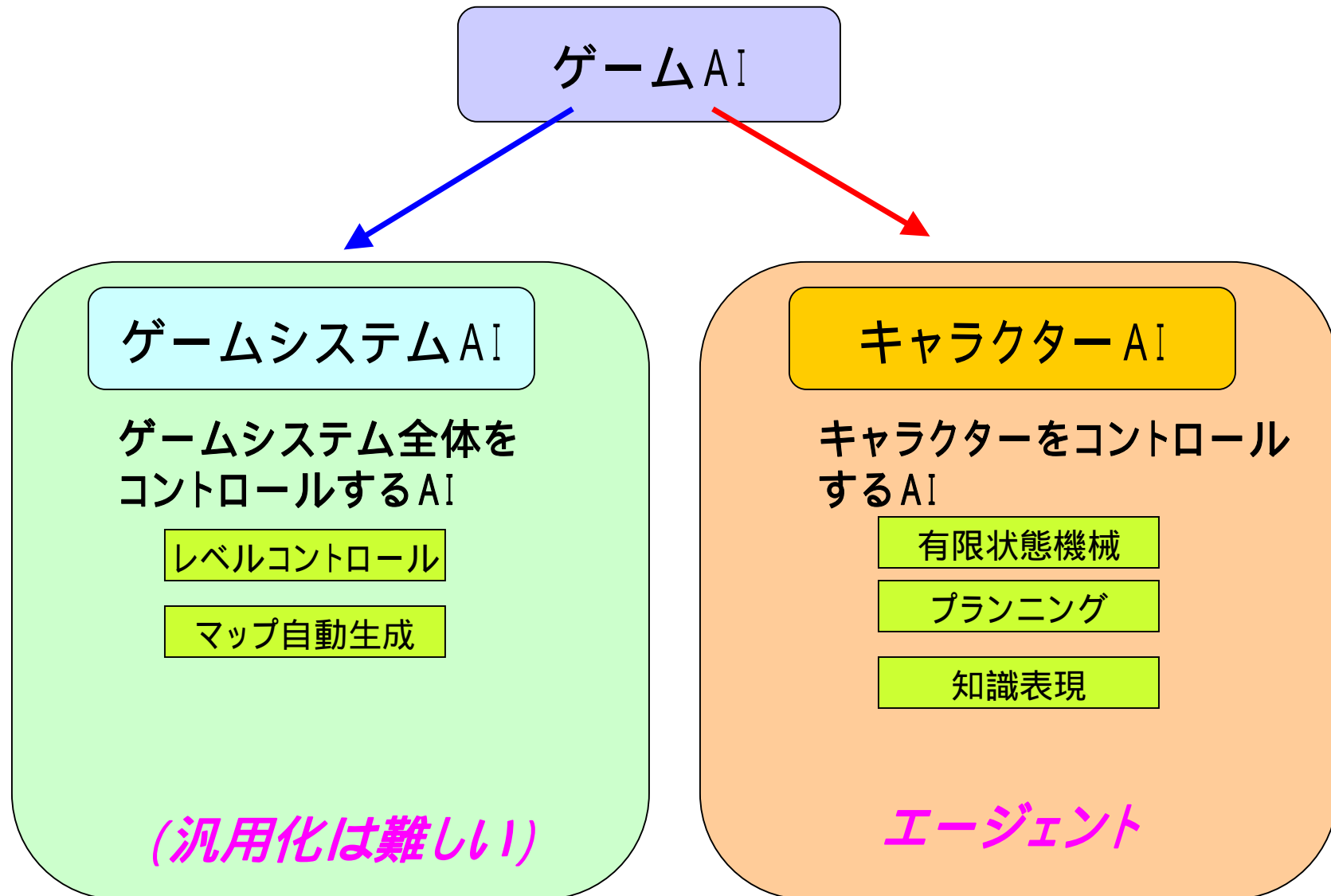
ご清聴ありがとうございました。

質疑応答

これ以外に、意見や質問があれば、メールへ

y_miyake@fromsoftware.co.jp

ゲームAIの2つのカテゴリー



ご清聴ありがとうございました。



これ以外に、意見や質問があれば、メールかアンケートへ

y_miyake@fromsoftware.co.jp

(IGDA Japan登録アドレス yoichi-m@pk9.so-net.ne.jp)

WEB上の意見交換にはIGDA Japanのサイトをご利用ください

<http://www.igda.jp>