ニューラルネットワークとゲームAI

ニューラルネットは、電気うり坊の夢をみるか？

V1.0

Planning
オープランニング
プランナー　大野　功二

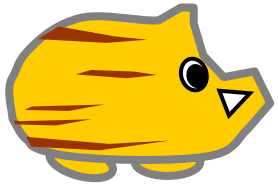
自己紹介

オープランニング・プランナー
大野　功二

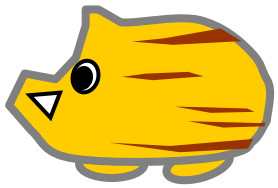
元プログラマーのノウハウを活かして、
現実性の高い企画内容・スケジュール見積もりをモットーに、
大手ディベロッパー様のプランニング作業を
お手伝いさせていただいています。

また、オリジナルゲームのプランニング、
企画応用のためのゲームプログラムの研究開発などもしております。

目次



パート1



ニューラルネットへの挑戦

1.1 これまでの経緯

概要

きっかけは、昨年行われたゲームAI連続セミナーへの参加でした。

開催日　:2007/10/27 (土)
ゲームAI連続セミナー「ゲームAIを読み解く」第5回

上記のグループワークにおいての記録を作成
テーマは「NN，GA，NEATなどを使ったゲーム企画を考える」

開催日　:2007/12/15 (土)
ゲームAI連続セミナー「ゲームAIを読み解く」第6回

第5回グループワークメモを発表
また、その成果物として、シューティングゲームの
オプションにニューラルネットを実装したプログラムを公開

2008/09/10(水) 本講演に至る

グループワークの内容発表

概要：

１・ニューラルネットをシューティングのオプションに実装できないか検討
　(企画テスト用のシューティングプログラムを流用できるため)

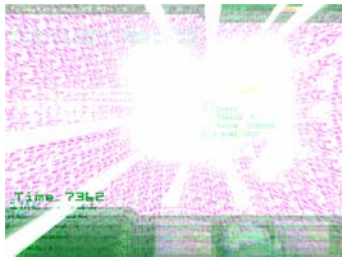
２・しかし、そのまま実装したのでは、
　ニューラルネットの持ち味を活かせない。

３・「AIを活かす」ということは、どういうことなのか？
　　→キャラ化することで活かせるのではないか？

４・キャラ化したオプションを
　実際のシューティングゲームに実装して実演

基本の材料

材料となったシューティングゲームは、下記のような仕様です。

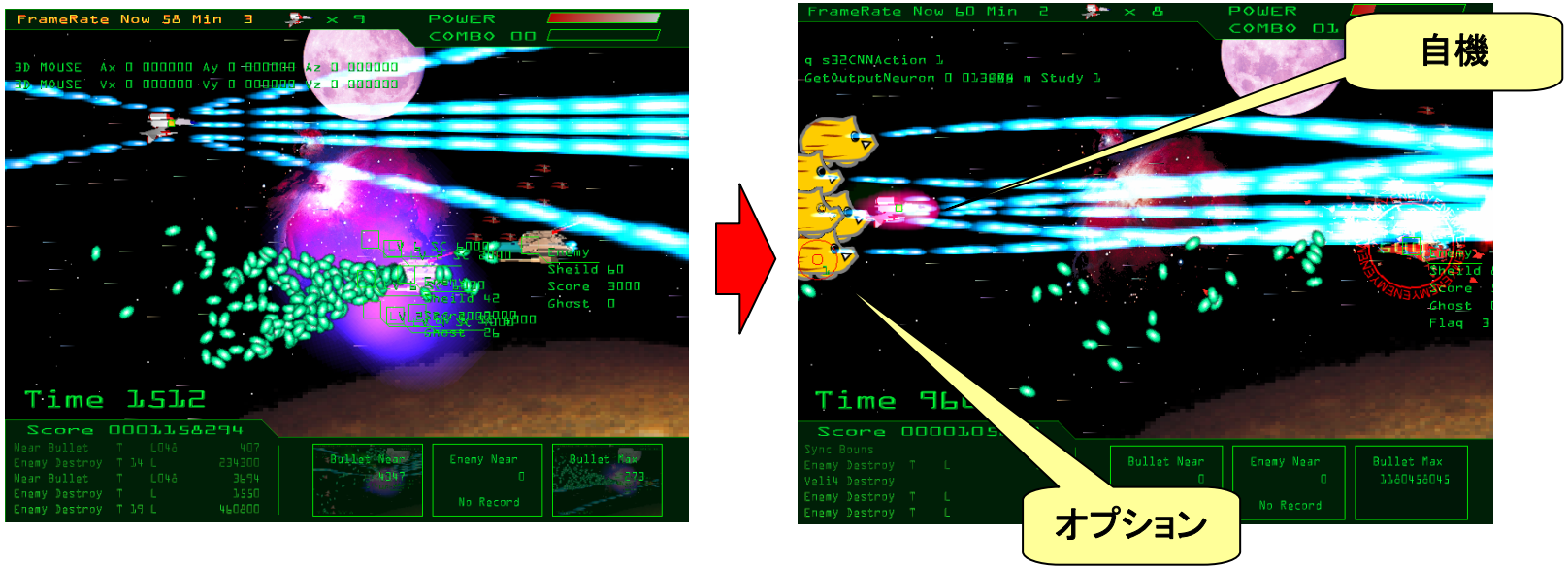


- 2Dシューティング研究用のためのゲームです。
- ゲームシステムは、スタンダードな2Dシューティングのシステムです。
- 元のオリジナル企画では、オプションはありません。
- 直感的なゲームインターフェイスの研究をしており、マウスで操作します。
- Luaをスクリプトの研究開発の目的で実験的に実装しています。
  (ユーザーが作成できるミッションのみLua対応。UCC実験用)

実演

これをAIゲームにしてみる

このゲームでは
「プレイヤーは弾幕を避け、オプションを賢く強化して敵を倒す！」
を表現します。



・自機はプレイヤーが操作する。
・自機は移動は可能だが、弾を発射することはできない。
・自機はオプションを「ほめる」「しかる」ことができる。

・オプションのAIはニューラルネットで実装して、オプションが敵を攻撃する。
・オプションが学習して進化する。
・オプションには当たり判定があり、敵、または敵の弾に当たると死ぬ。

1.2.ニューラルネット基礎知識

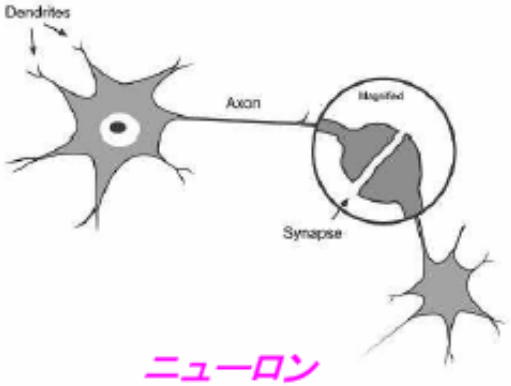
2007年10月のゲームAI連続セミナー「ゲームAIを読み解く」第5回
上記セミナーで学んだこと

ニューラルネットとは？

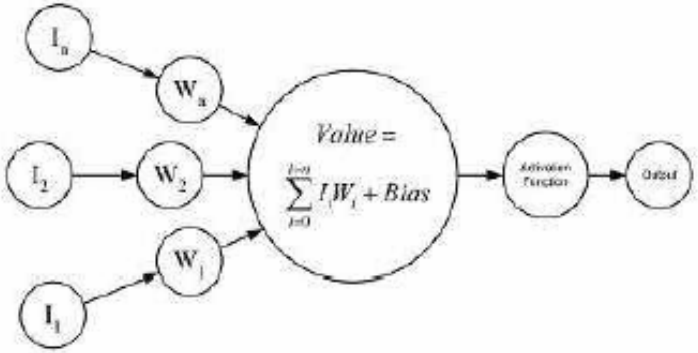
「ニューラルネット」は、生体の神経細胞「ニューロン」を数学的にモデル化し、
相互に接続して構成したネットワークのことです。

医学的知識

数学的モデル

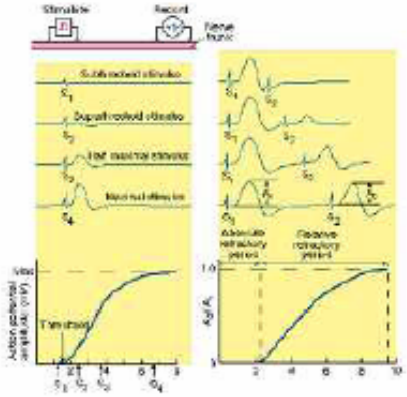


ニューロン

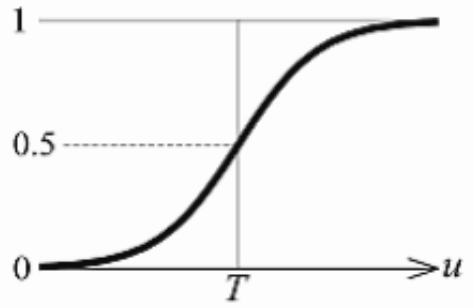


人工ニューロン

モデル化



入出力関係のグラフ



$$f(u) = \frac{1}{1+\exp(-s(u-T))}$$

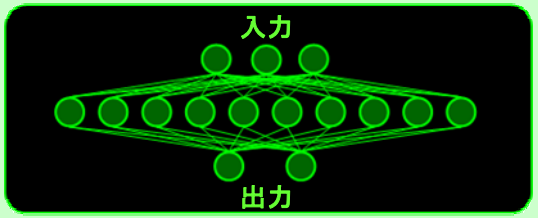
入出力関係の関数（シグモイド関数）

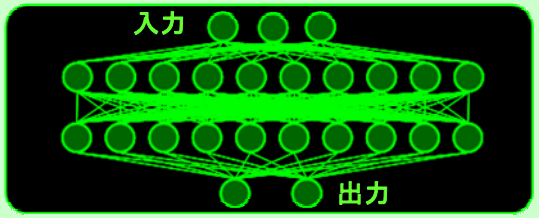
http://www.pri.kyoto-u.ac.jp/brain/brain/11/index-11.html

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

ニューラルネットのいろいろ

フィードフォワード型　　信号が入力側から出力側へ1方向へ流れるニューラルネット方式

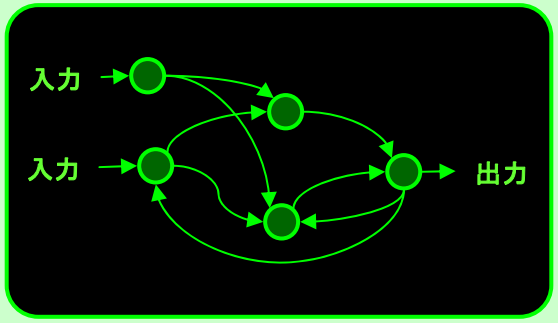




多層のニューラルネット

・入力層・隠れ層・出力層で構成される。
（隠れ層は、何層でも増やすことができる）

リカレント型(相互結合型)　相互に依存しあいながら時間的に値を決定するニューラルネット方式



ボルツマンマシン(マルコフモデルのひとつ)

・相互結合型ネットワークの一種であり、
ネットワーク内部に信号帰還のループを持つ。

本セミナーでは、フィードフォワード型の3層ニューラルネットのみ説明します。

「今までのAI」と「ニューラルネットAI」の違い①

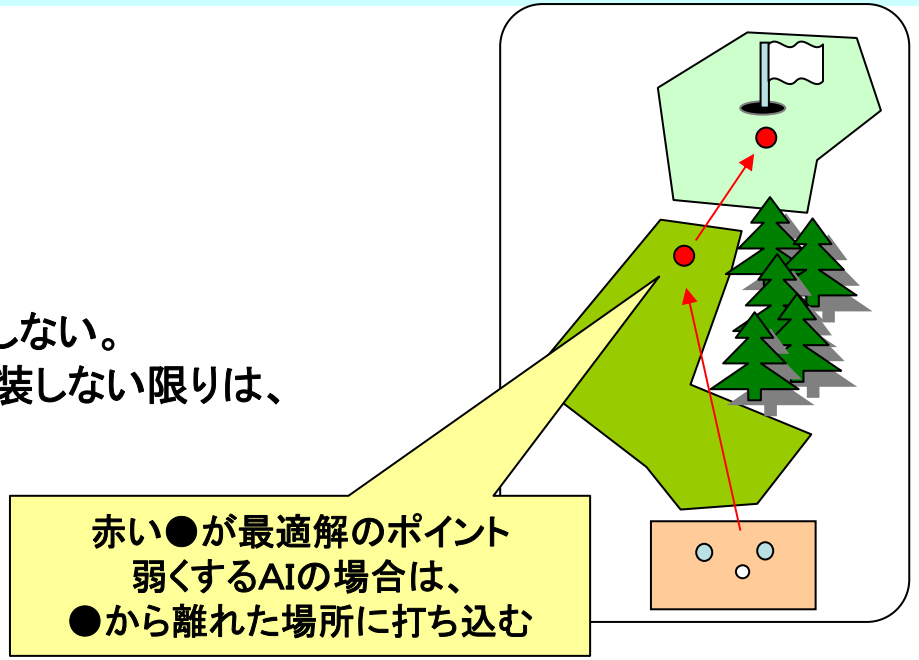
通常のプログラムされたAIの多くは、シンボリズム（記号主義）で作られています。

企画やプログラマーが想定している条件などから「最適解」を最終ゴールとして
そのゴールまでに「ゆらぎ」を作ることでゲームのAIを作っている。

例：ゴルフゲームでAIを作る場合、コースとパー数にあわせて、
　企画側で最適の行動をパラメータとして調整しておく。
　このため、最強のAIの状態からパラメータで誤差を作り、最弱のAIを作っていく。

[メリット]
ゲーム全体のバランスを把握しやすい。
バランス調整がしやすい。

[デメリット]
企画者およびプログラマーの想定内でしかAIは行動しない。
想定外のことは、企画が「想定外のイベント」として実装しない限りは、
ほぼ発生することはない。



「今までのAI」と「ニューラルネットAI」の違い②

これに対しニューラルネットは、学習と経験によって「最適解」を得ようとするので、
企画やプログラマーが想定しなかった「最適解」を求める可能性がある。
これは「意外性(または創発)」を期待できる。・・・にハズなのですが。

創発とは「低位における性質の集積からなる高位で、低位からは予見し得なかった性質が表れること」です。

例：ゴルフゲームのAIとして実装する場合、
最適な解を教師信号として与えて、学習と行動をさせる。

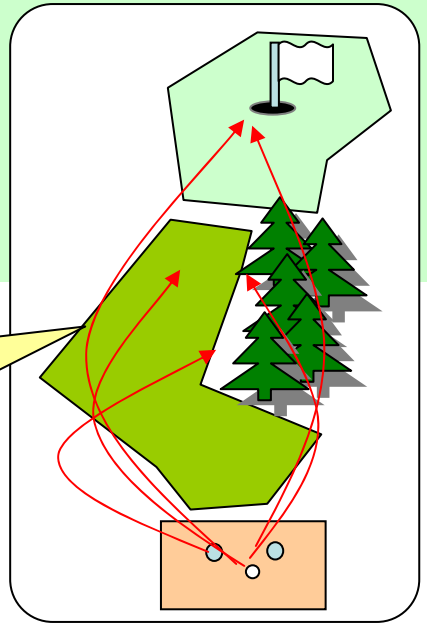
このとき、実際に思考した内容をゲームの環境にも反映するため、
教師信号と違う行動で、高い評価を得る可能性もある。

例えば、教師信号は、フェアウェイにのせるように打ち方などを支持したが、
学習が進んでいないニューラルネットAIは森の中に打ち込む。
しかし、森の木にあたってグリーンにオンして、ホールインワンとなる。
これは高い評価の一つとしてニューラルネットは学習する。

[メリット]
企画の想定を超えた意外性を期待できる。

[デメリット]
調整が大変。最適解を得るまでに時間もかかる。
実装も大変。

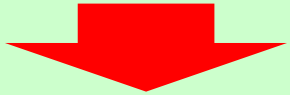
最適解は、さまざまな打ち方をして、
いかに少ない打数でカップインで
きたかで評価される。



夢ふくらむニューラルネット

ニューラルネットを知って、下記のことを勝手に想像しました。

入力された情報を元に、学習を行い、結果を出力をすることで、
「企画やプログラマが想定しえない多様な環境(非線形の事象？)」
に対しても、期待された結果を出力をすることができる。



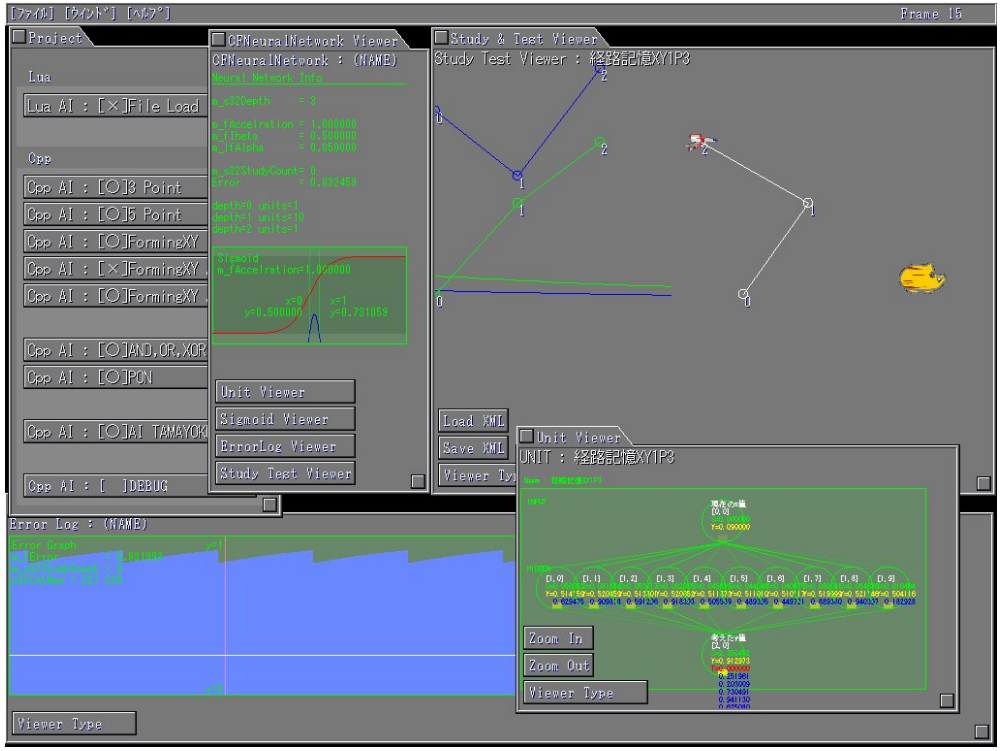
勝手に勉強して賢くなるAIアルゴリズムになる？？？

ニューラルネットがあれば、
アルゴリズムプログラムは必要なくなる？？？

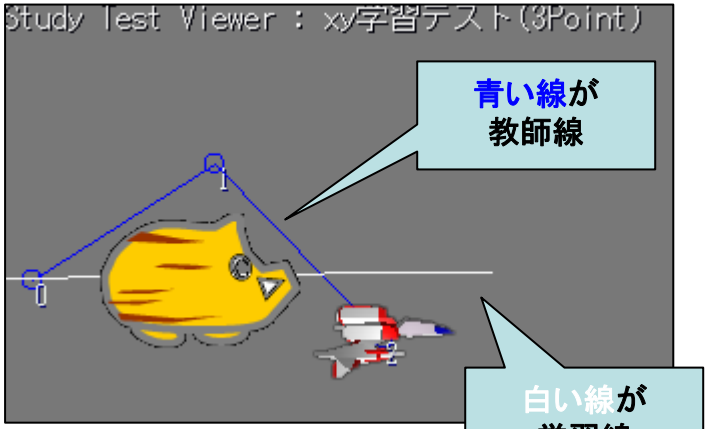
生物の思考をシミュレーションできる
夢のAIを実現できる！！

すごいぞ！ニューラルネットワーク！！

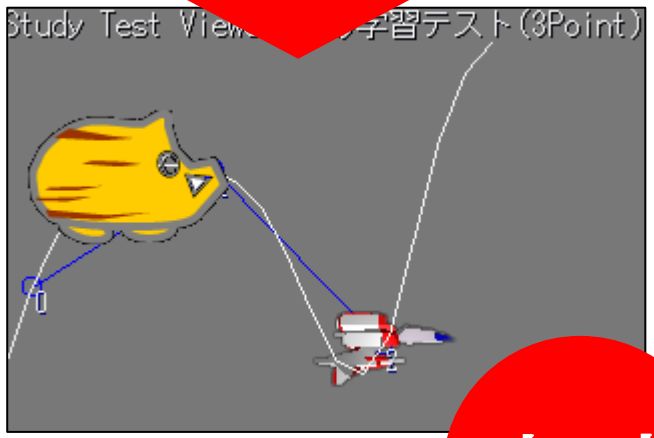
実際にニューラルネットの学習過程を見て、体験しましょう！！！



作成したニューラルネットビュワーで、
３点を通る線を学習するニューラルネットを
見ていただきます。



①学習前



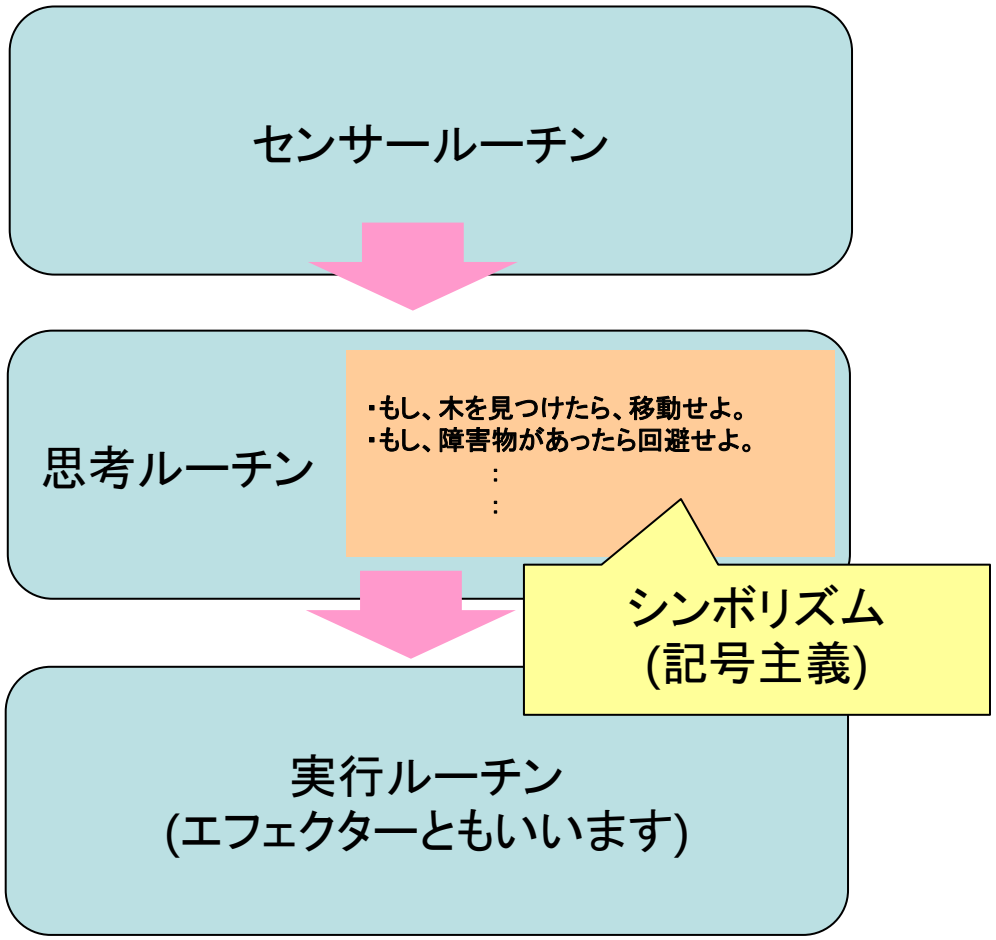
②学習後

実演

1.3ニューラルネットを作ってみる

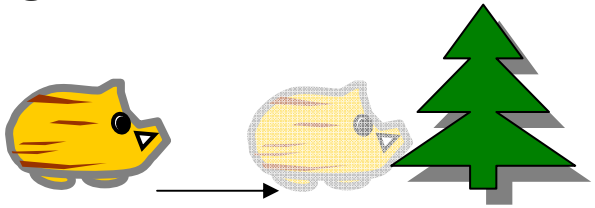
AIの基礎知識

人工知能・AI は、
「センサールーチン」「思考ルーチン」「実行ルーチン」の3つから構成されます。

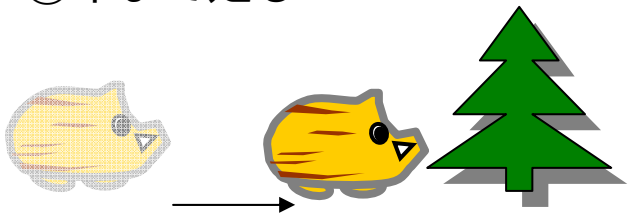


①木があるぞ

②木のところまで行きたい！！

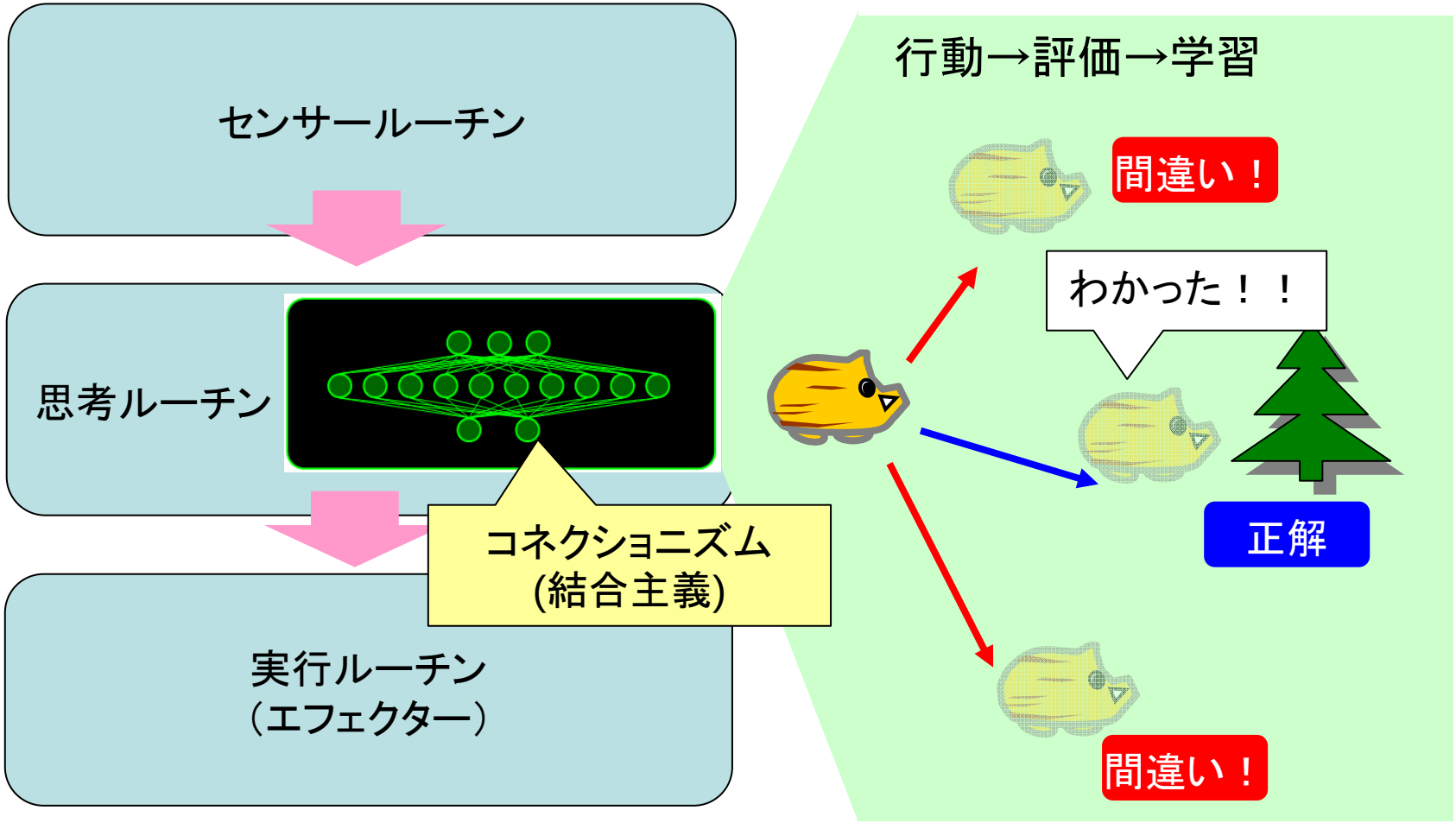


③木まで走る！！



ニューラルネットで置き換える

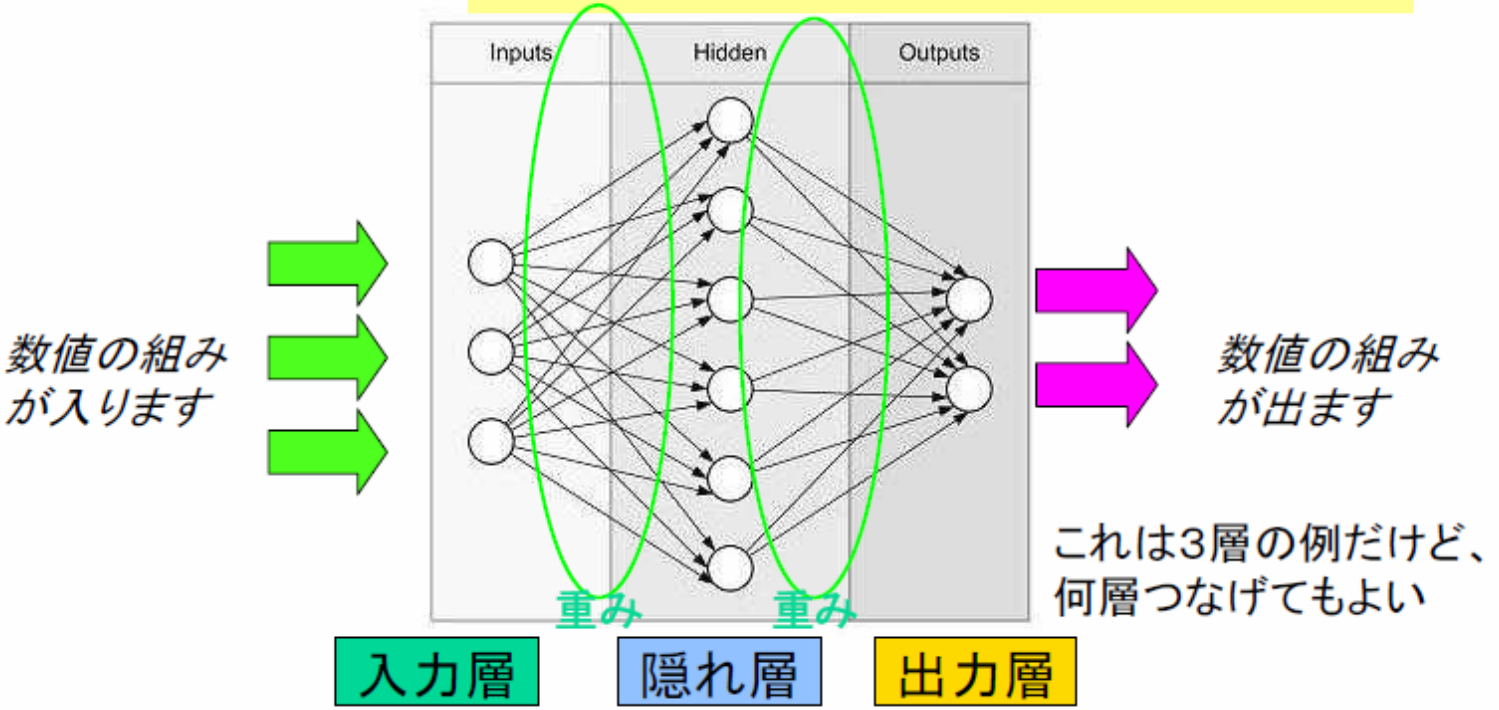
思考ルーチンをニューラルネットに置き換えます。
思考ルーチンは、行動→評価→学習のループを行うことで、
目的を達成するニューラルネットができます。



ニューラルネットの構造

数学的モデル

リカレント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＝ウエイト（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）

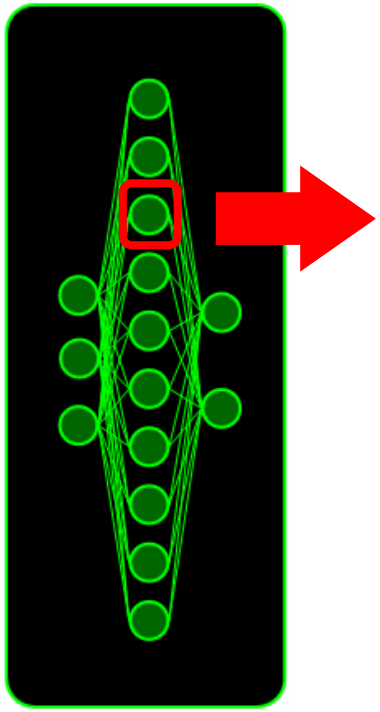
一旦定義してから変えることができないもの…全体の構造
変えることができるもの…ウエイト（重み）

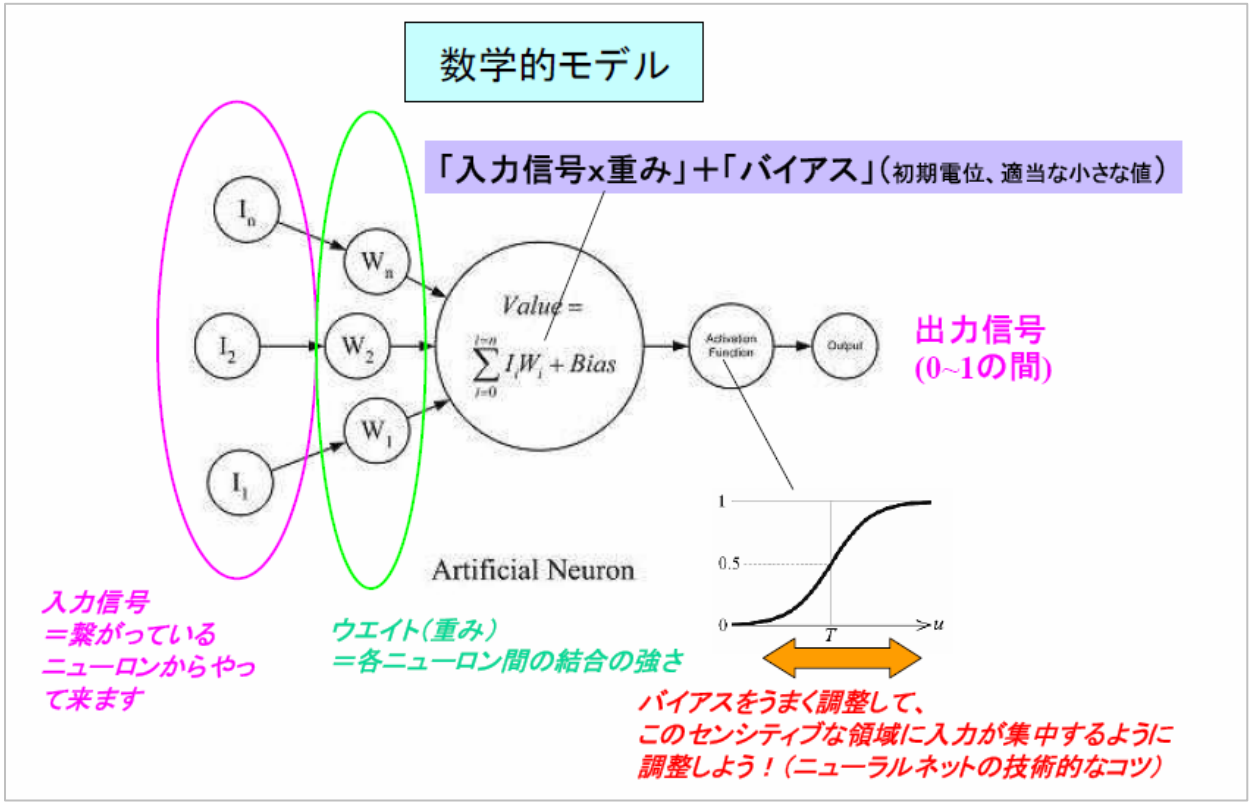
第5回「ニューラルネットによるエージェント」 事前資料-ニューラルネット編より-

ニューラルネットの計算

下記の数学モデルを使って、1ユニットずつ隠れ層から出力層まですべて計算する。

ニューラルネット





第５回「ニューラルネットによるエージェント」 事前資料-ニューラルネット編より-

学習（BP法）①

ニューラルネットの学習には、誤差逆伝搬法(Back Propagation Method)を使います。

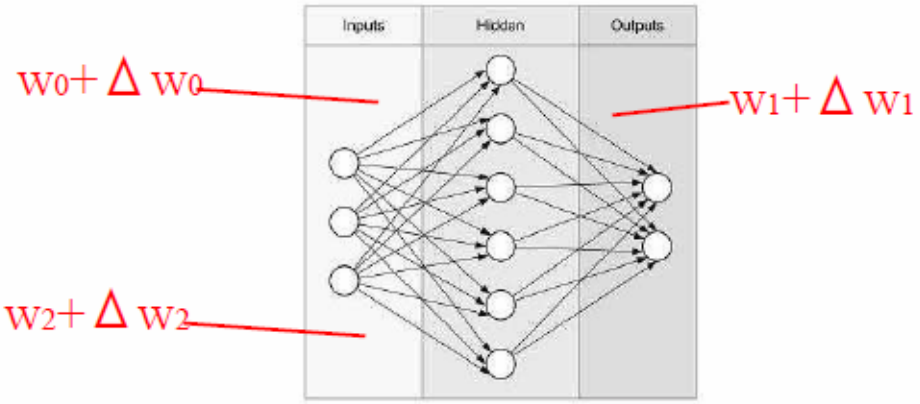
基本的な考え方

「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、一体どれぐらい変えればいいのだろうか？

第5回「ニューラルネットによるエージェント」 事前資料-ニューラルネット編より-

学習（BP法）②

最速降下法

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

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

ニューラルネットの場合（出力層について）

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

$$R = (Out(w) - T)^2 / 2$$

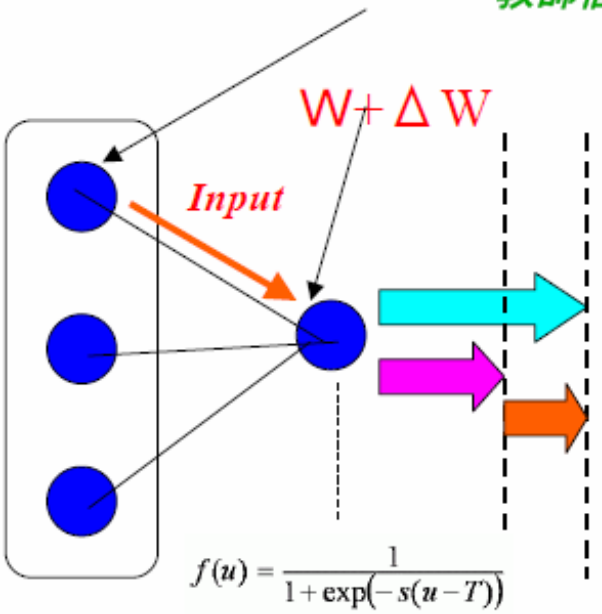
$$\Delta W = -\alpha * Error * Input * Out * (1 - Out)$$

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

教師信号

1step 前の変化分

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

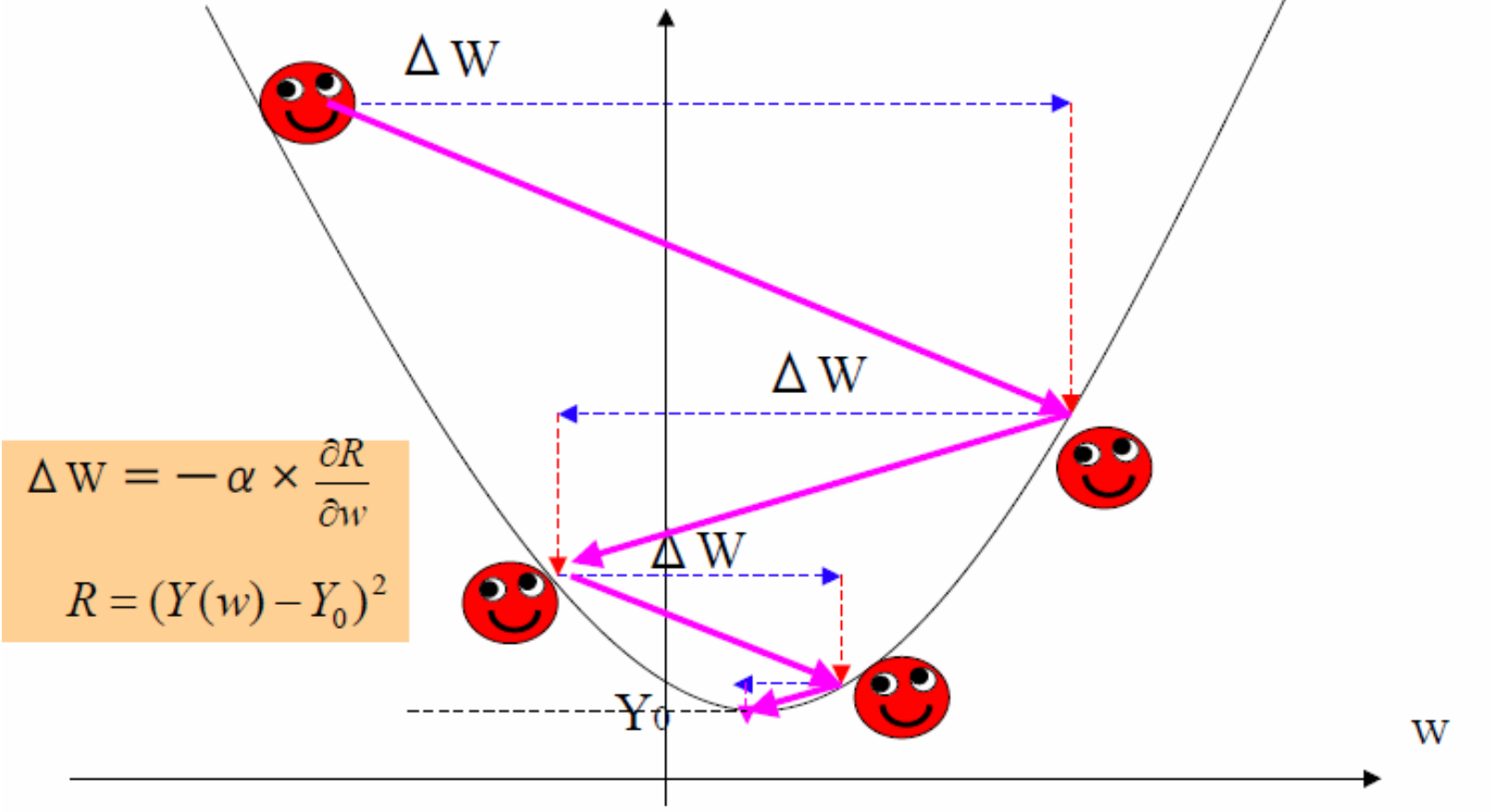


第5回「ニューラルネットによるエージェント」 事前資料-ニューラルネット編より-

学習(BP法)③

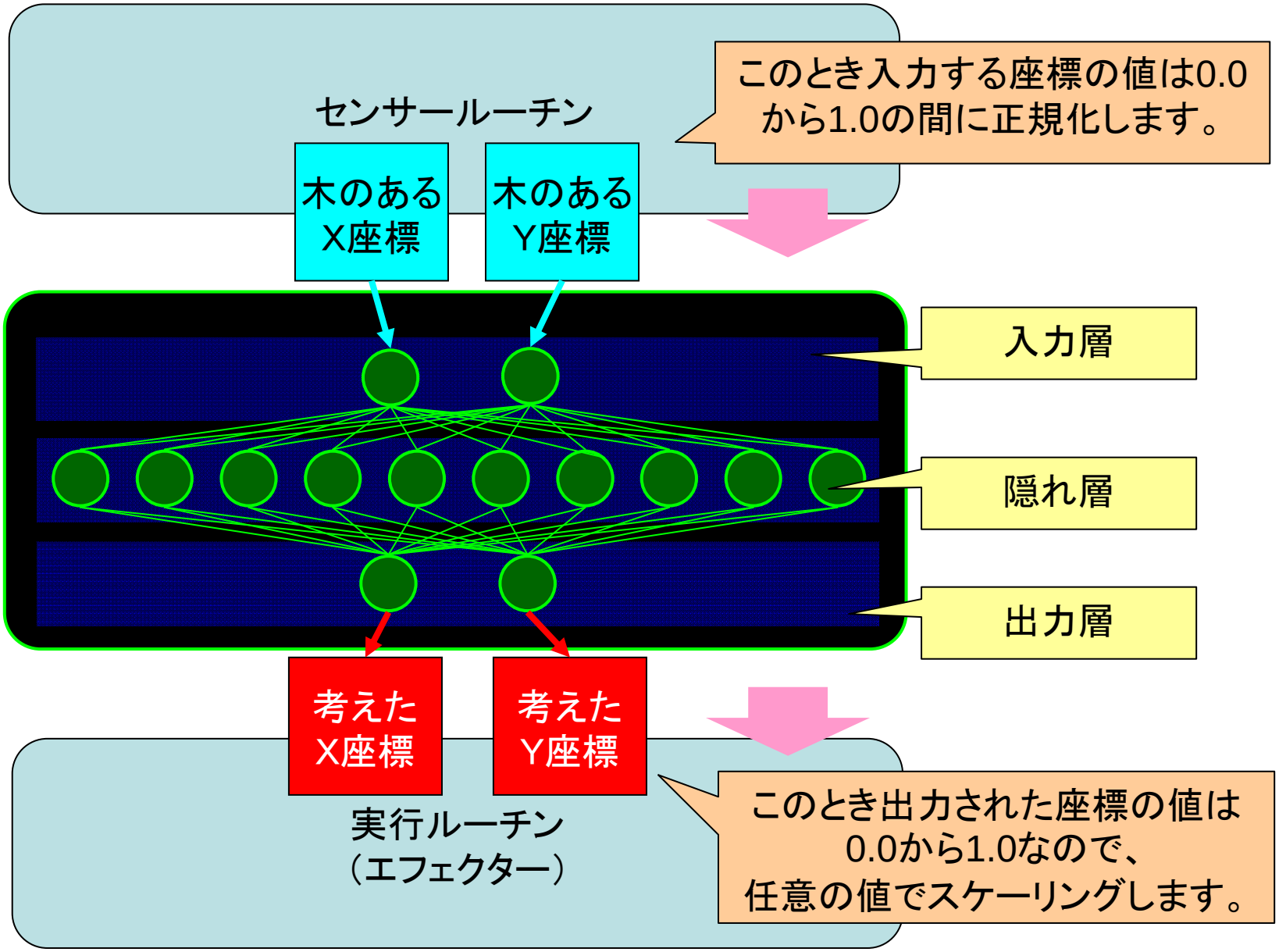
最速降下法

① とりあえず少しずつつずつ進みましょう。(小さい数αを用意。適当なスケール)
② 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。
③ 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。



第5回「ニューラルネットによるエージェント」 事前資料-ニューラルネット編より-

ニューラルネットの実装

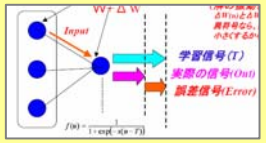


ニューラルネットの実行の流れ

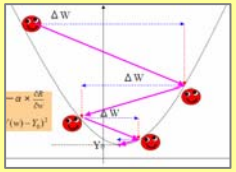
ニューラルネットをプログラムした場合の流れは、下記の通りです。

初期化
ニューラルネットの初期ウェイトを
ランダムに設定

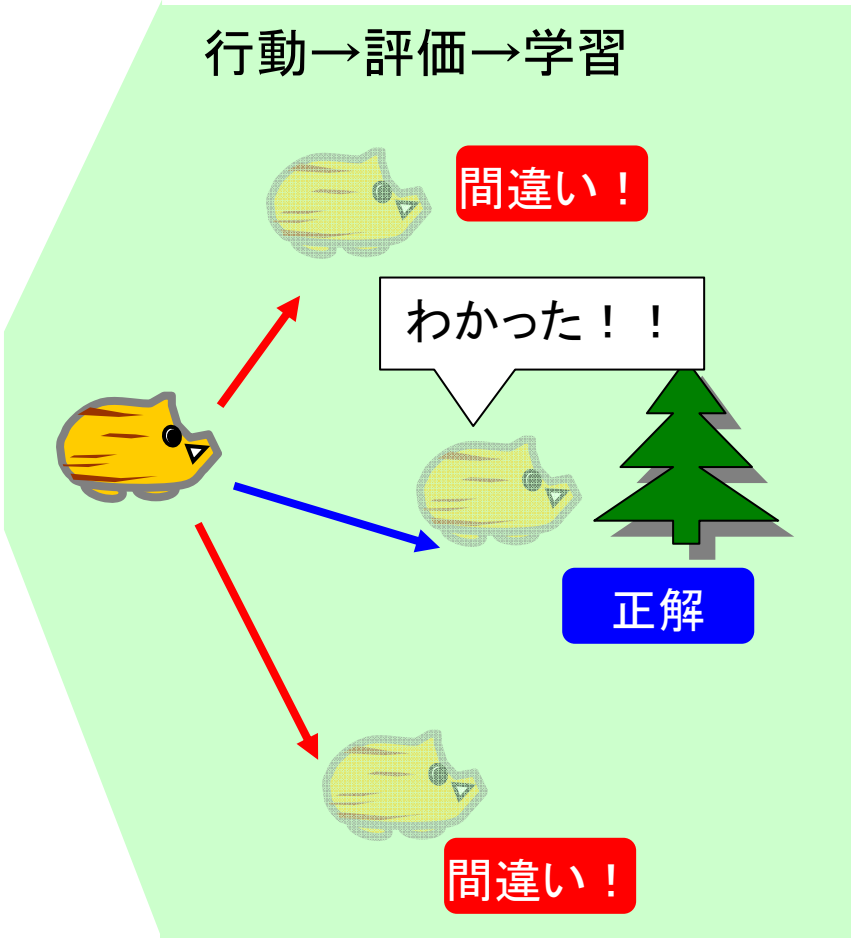
思考→行動
ニューラルネットで思考した
結果をもとに行動

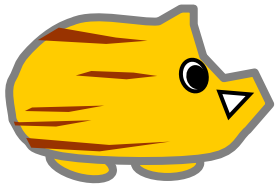


評価
木の座標と出力した座標から
エラー値を計算

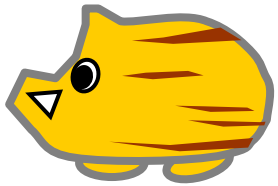


学習
評価で得たエラー値から
BP法でウェイトに反映





パート2



ニューラルネット実践

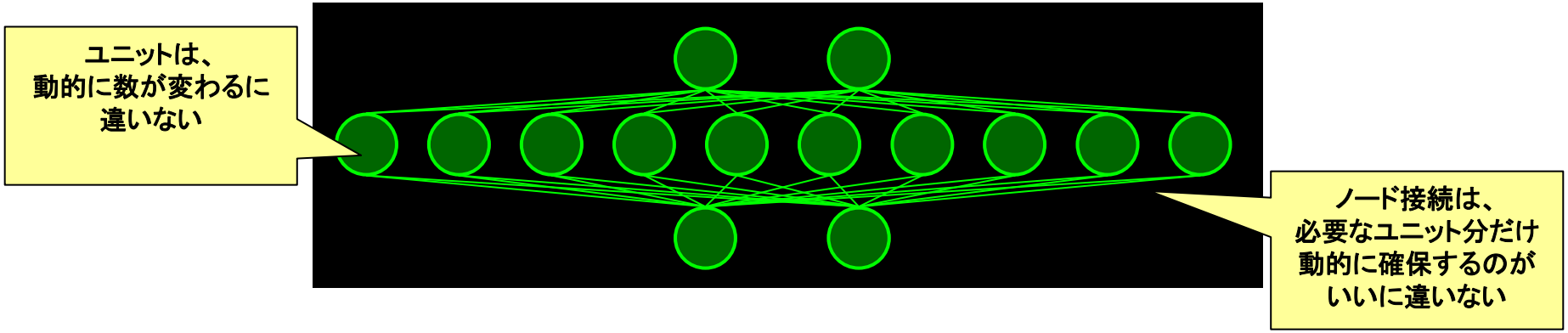
ニューラルネットの実装

2007年12月のゲームAI連続セミナー「ゲームAIを読み解く」第6回
グループワークメモ発表までの経緯

ニューラルネットクラスを作る①

最初に、テストプログラムを作り
ホーミング用のニューラルネットを作成することにしました。

まずは、汎用的にニューラルネットが作れるクラスライブラリを作成することにしました。



①とりあえず、なんの参考資料もなしに、いきなりプログラムに取りかかる。

②ニューラルネットのことを良く分かっていなかったので、
各ユニットをつなぐノードをリアルタイムにメモリ確保しながら、
いかなる変更もできるようなデータ構造クラスとして作成。
(状況にあわせて、ユニット数もノード数も変更できる必要があると考えていました)

③上記構造は、かなり複雑なデータベース構造となり、バグが多発。
単体テストでも、思った通りの結果が出ず。失敗！！

ニューラルネットクラスを作る②

ぜんぜんうまくテストができないことに気がつき、
やっと各種資料を見ることにしました。

その結果・・・

・ニューロンのユニット数は、最初に一度決めてしまえば、動的に変化することはない。

・上記と同様に、階層が動的に変化することもない。

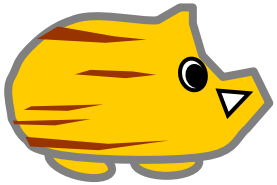
・ユニット同士を接続しているノードは、次の階層にすべて接続されている。
接続のオフはノードのウェイトが0のときに判断する。

・つまり、最初にニューラルネットのデータ構造を動的に確保すれば、
その後はデータ構造を動的に変化させる必要はない。

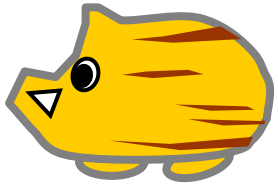
はじめにイメージしていたニューラルネット構造は完全に間違っていた！

無駄な時間を1ヶ月過ごしてしまいました・・・

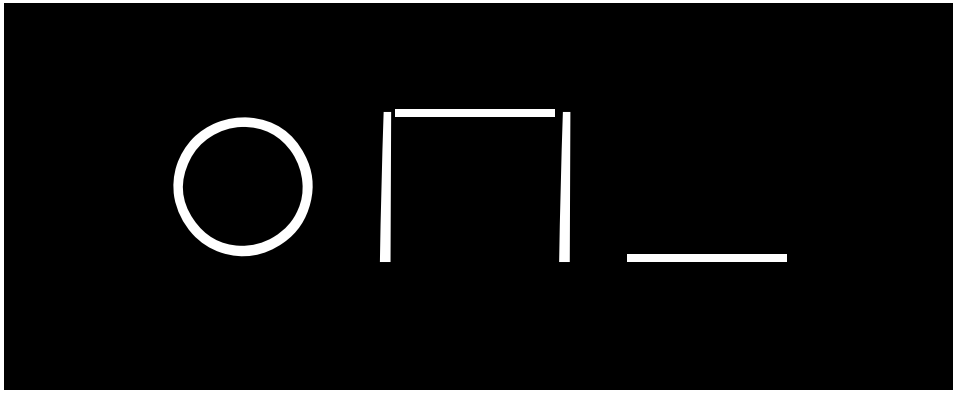
学んだこと・・・



教訓



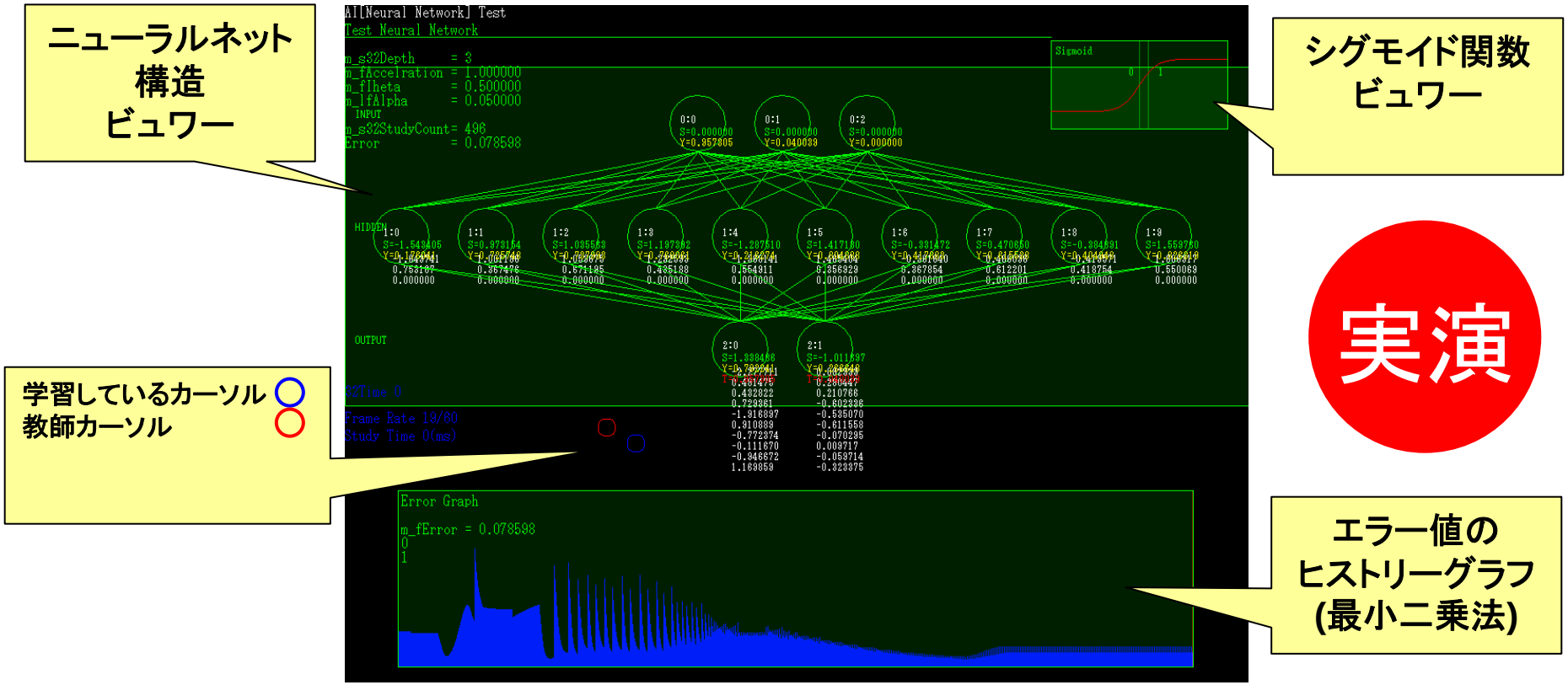
資料は良く読もう！
人の話も良く聞こう！



テストプログラムを作る

反省して、ニューラルネットを視角化してテストできるプログラムを作りました。

ニューラルネット・テストプログラム

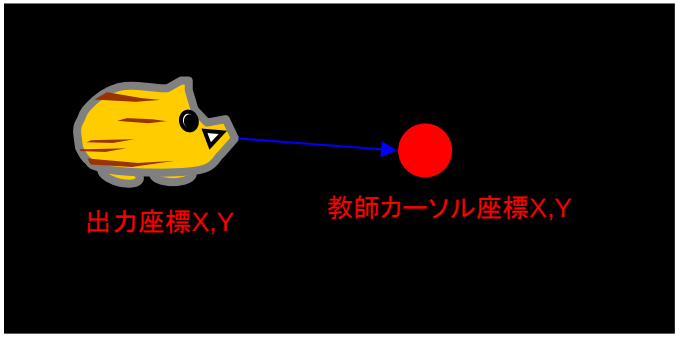


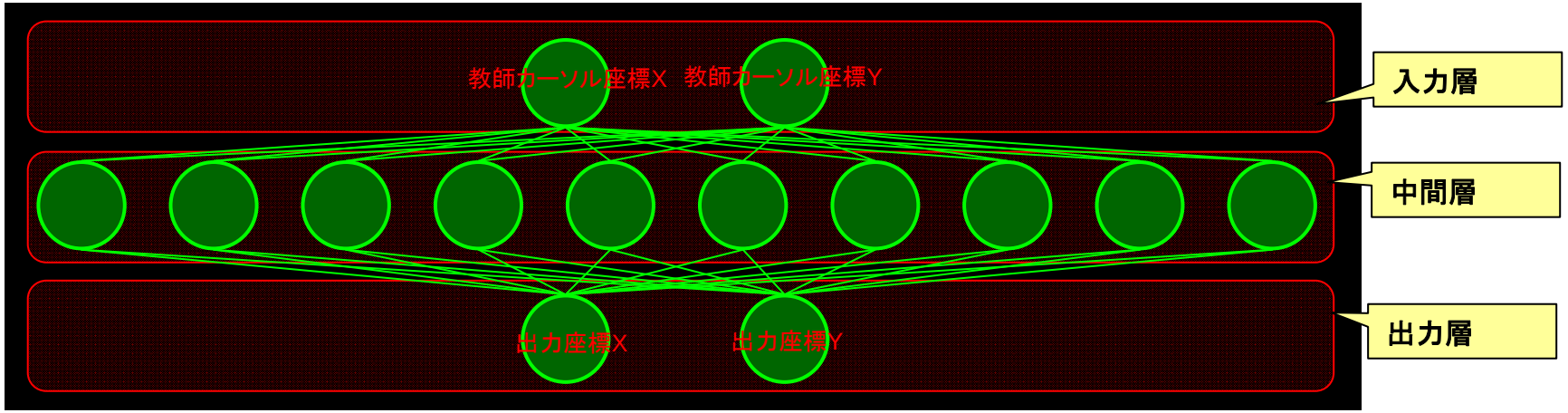
プログラムテストも終了して「ニューラルネットなんて楽勝！！」と
このときは思っていました。

ホーミングニューロン

作成した最初のホーミングアルゴリズムを実装したニューロンは下記の通りです。

入力層：1　　ユニット数：2
中間層：1　　ユニット数：10
出力層：1　　ユニット数：2



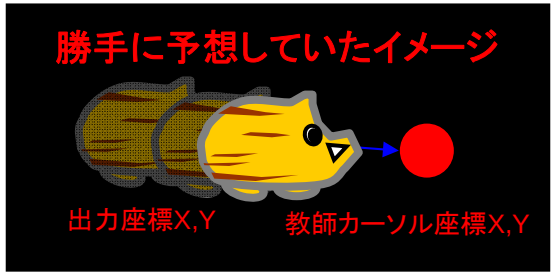


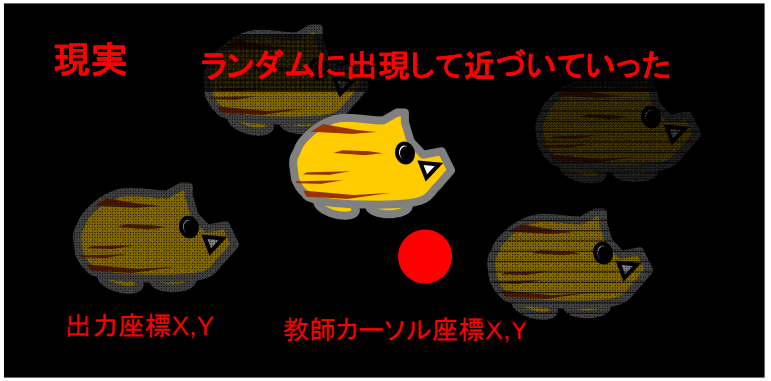
プログラムの単体テストもうまくいき、
さっそく学習させてみました。

問題

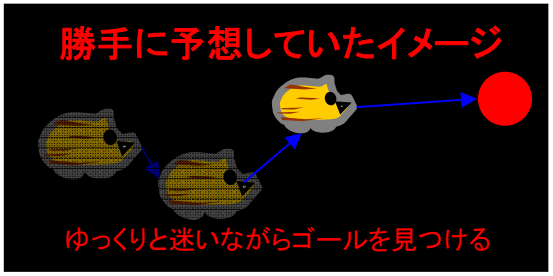
しかし、できたホーミングニューロンは、実際に動作させてみると
下記の点でゲームには使えないことがわかりました。

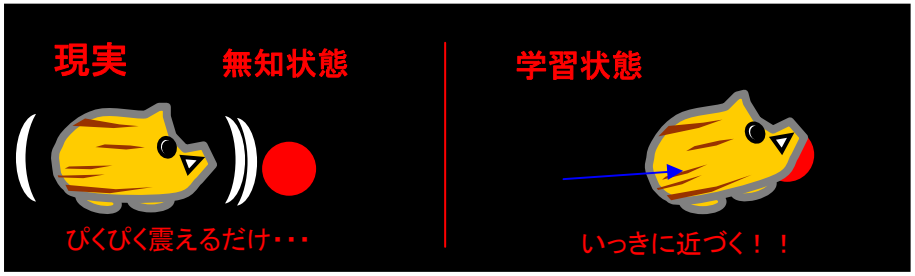
・学習過程の行動が、イメージしていたものと違う。
（ゆっくりと教師信号のカーソルに近づくと思っていた)





・学習の過程の流れが、イメージしていたのと違う。
(無知な状態から、その学習内容を理解する過程がゆっくりと曲線的になると思っていた。
実際には人間の見た目には、デジタル的に「いきなり理解した」ように見える状態になった)





・教師信号のパターン数が思っていたよりも膨大に必要であることが分かった。
(2点か3点、ホーミング先の位置を学習させれば、学習が収束すると思っていた。
実際には、ランダムな座標を数万回のオーダーで教師信号として使わなければならなかった)

問題解決に向けて

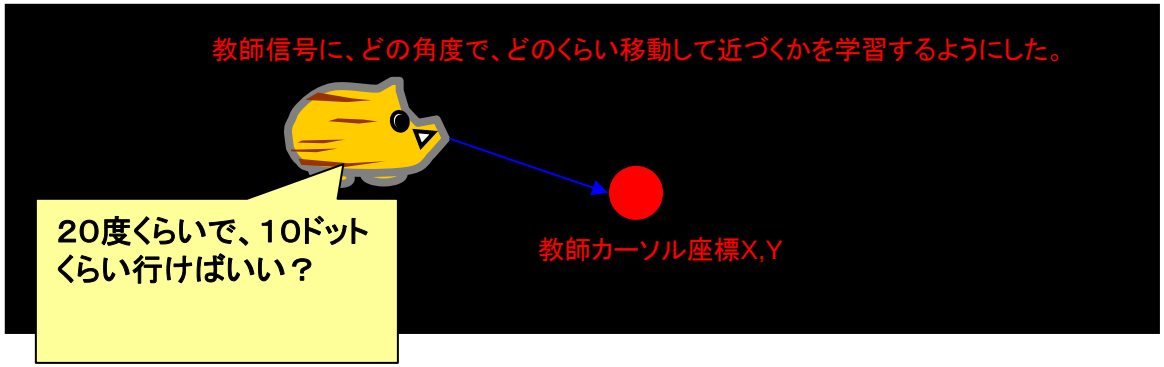
そこで下記のように「企画対応」を行いました。

[問題]

・学習過程の行動が、イメージしていたものと違う。
→ホーミングアルゴリズム部のニューロンを角度センサーと移動分の2個に分割。

最初に作ったホーミングニューラルネットは、座標の値を学習するだけでしたので、
「ホーミングしなければいけない角度」「先に進まなければいけない距離」の
2つに分けてみました。

また、思考のゆらぎを持たせるために、
実験的に入力層に一定時間でループするタイマーの値を入れました。

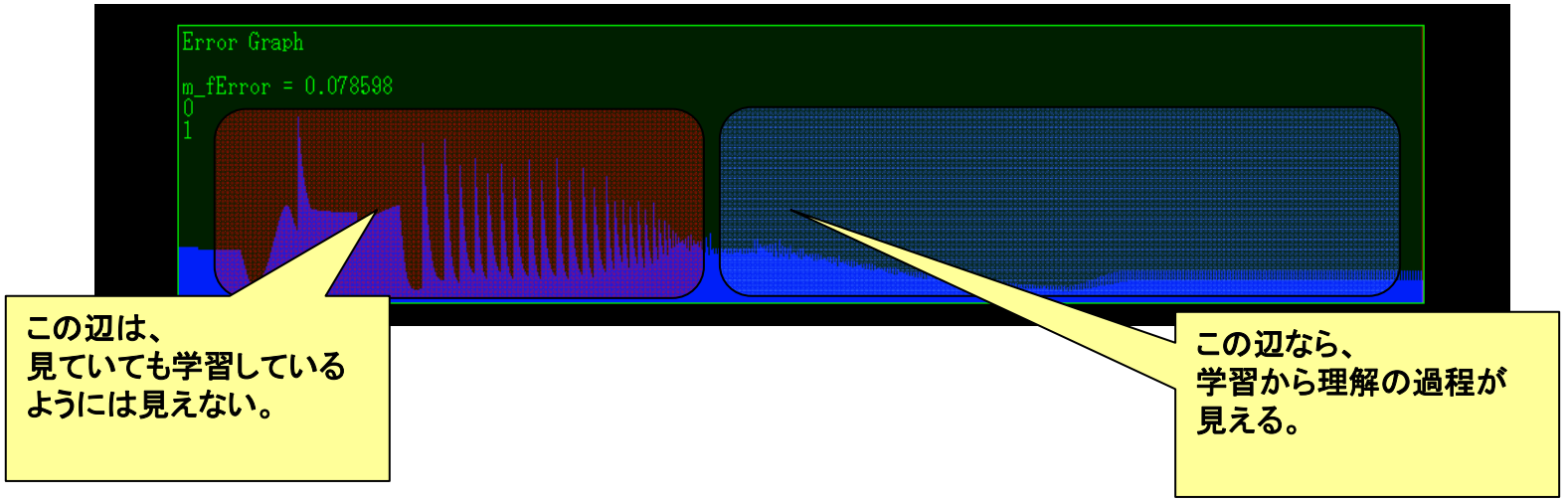


問題解決に向けて

さらにプレゼンで学習効果を見せるために下記の「企画対応」を行いました。

[問題]
・学習の過程の流れが、イメージしていたのと違う。
・教師信号のパターン数が思っていたよりも膨大に必要であることが分かった。
→無学習のニューロンウェイトデータを使うのではなく、
学習曲線が収束する直前のデータを使って対応

発表前日に、私がニューラルネットが理解する直前まで、
繰り返しゲームプレイして、学習曲線が収束するまえのウェイトデータを作りました。



シューティングゲームへの実装

シューティングへの実装は、下記のように行いました。

ルールは簡単です。
自機を移動させてマウスを右クリックをすると、その場所が教師信号になり、
オプションはリアルタイムに学習しながら教師信号をおいかけます。

1フレームごとに、処理落ちしないように時間を計測してBP法で学習しています。
発表会では、2，3分で「学習して理解した」様子が見れるように実演しました。

さらに味付け

しかし、これだけではオプションのすべての「うり坊」が、
同時に学習してしまいます。そこで・・・

いっぱい学習させたエライ子

エライ子より学習量の少ない普通の子

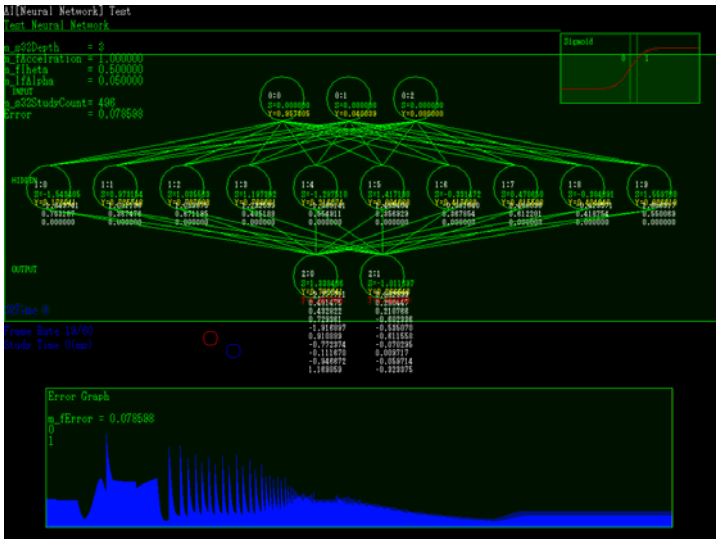
エライ子より学習量の少ない普通の子

学習量の圧倒的に少ないアホな子

というように学習量に変化をつけました。

発表

下記の流れで、ゲームAI連続セミナー第6回にて発表しました。



①ニューラルネットのビュワーツールで、
ホーミングニューラルネットが単体で
学習する過程を説明

②実際にニューラルネットを
オプションに組み込んだ
シューティングゲームをプレイしながら説明

実演

そして・・・発表は終わり・・・

発表は無事に終わり、暖かいご声援をいただきました。
その後、三宅さんから衝撃的なお話を聞くことに・・・・

三宅さん　　：

「よくゲームに実装できましたね。
ニューラルネットはブラックテクノロジーと言われて、
なかなかゲームの実装で成功したものがないんですよ。」

大野　　：

「え？！」

壁にぶつかる

ブラックテクノロジー

なぜ、ニューラルネットは「ブラックテクノロジー」なのか？
改めて考えてみました。

・ニューラルネットは、何でも自動的に学習するわけではない

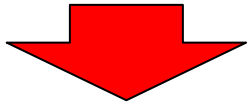
・学習過程は、生命的なものとは限らない（そう見えないことが多い）

・教師信号は適切に分布した値を使う必要がある。
偏った教師信号では、学習が収束した場合に、イメージ通りにならない。

・学習に多大な時間がかかる・・・
また、過学習になってしまうと期待した結果にならない場合がある

・結局、企画側がイメージした生物的な学習過程を見せようと思うと、
「多くの調整期間」と「運」が必要となる。

・しかも、上記がわかっていれば、他のアルゴリズムで代用できる場合が多い。



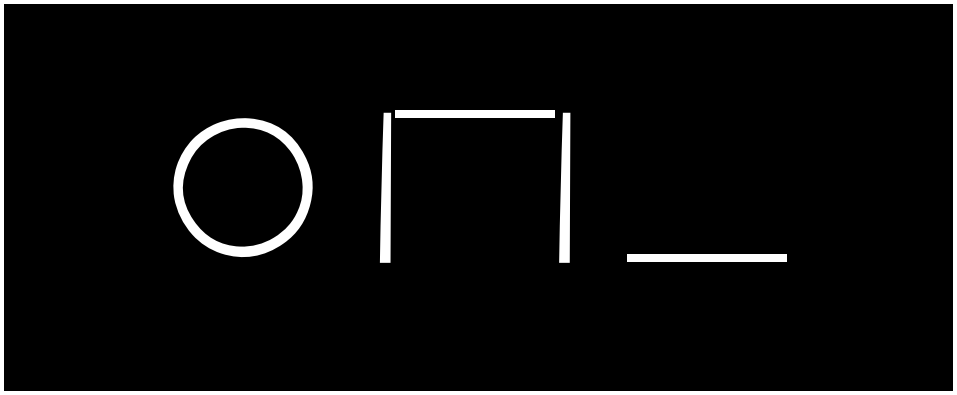
なるほどブラックです・・・

ニューラルネットへの絶望

学習するまでの過程が遅いじゃん！！！

ぜんぜん生物っぽくないじゃん！！！

なんでも学習できると思ったのに、学習できないじゃん！！！



ニューラルネットでなくてもいいじゃん。

ニューラルネットの技術書

そこで、改めてニューラルネットの技術書を読んでみました。

パーセプトロン　　M.ミンスキー/S.パパート著
訳　中野 馨/阪口 豊
パーソナルメディア

訳者プロローグより

「この分野の研究が遅々として進まないのは理由の一つは、
研究者たちがこの分野の歴史をよく知らずに、
ほかの研究者が犯したのと同じ誤りを繰り返しているからである。」

たぶん･･･僕も彼らの仲間入りができたと思っています･･･

ニューラルネットの歴史①

第１次ニューラルネットブーム

1943年	神経細胞(ニューロン)のモデルを提案 マカロック(W.S.McCulloch)とピッツ(W.H.Pitts)
1949年	シナプス強化法則の発表 D.O.Hebb
1958年	学習機械パーセプトロンを提案　その後”Principles of Neurodynamics”にまとめる F.Rosenblatt(心理学者)
1969年	PERCEPTRONS – Expanded Edition(改訂版) Marvin L.Minsky Seymour A. Papert

停滞期

ニューラルネット歴史②

第2次ニューロコンピュータブーム

1986年　バックプロパゲーション(BP法)が登場

1981年にMicrosoftがMS-DOSを発表

1993年　パーセプトロン(改訂版) 日本語翻訳版
訳：中野　馨
　　阪口　豊

1990年からファジィ理論の応用が
家電でブームとなる。

1997年　「がんばれ森川君2号」発売(森川　幸人)
1998年　「アストロノーカ」発売(森川　幸人)

ニューラルネットが実際にコンピューターに実装されはじめたのは、
この20年間である。

ニューラルネットの実例

ニューラルネットの実例には、下記のものがあります。

・画像認識
・文字認識
・指紋認証
・音声認識

・ロボットのセンサーや駆動部などのコントロール

・炊飯器の温度管理
・エアコンの温度管理

・ゲーム
　生物的人工知能への応用
　レースゲームなど

局所的に実装されているものや、実験的なものが多いのが現状です。
ニューラルネットを主軸として使えるのはゲームだけかもしれません。

ニューラルネットの実体

パーセプトロン　　M.ミンスキー/S.パパート著
訳　中野 馨/阪口 豊
パーソナルメディア

「プロローグ」より

筆者らが2章で行う解析からわかるように、何が解けないかという問題は、
実のところ学習方法とは何ら関係なかった。

（中略）

すなわち、Xを表現するための何らかのスキームを潜在的にもっていなければ、
その機械はXを学習認識できないのである。

ニューラルネットを実装する側が、学習させたいXの本質を知っていなければ、
Xを表現するためのニューラルネットを作ることは難しい。
（Xを学習するために必要なXが入る器を作るということ）

ニューラルネットを研究するには・・・

表現したいXを認識させるニューラルネットを作るために、
下記の学問に注目しています。

大脳生理学	大脳の機能に特化した生理学
脳神経科学	脳内の神経機能に特化した生理学
行動（主義）心理学	心の内面よりも行動面を科学的に理解しようとする心理学
生物学	生物の生命現象を研究する自然科学
ロボット工学	ロボット作成のための技術工学 行動するための駆動系だけではなく、AI研究も盛ん。 また、最近ではジェミノイドなど機能面よりも、 人間を理解するための研究もはじまっている。
オントロジー工学	知識の内容を徹底して分析する方法論を研究する工学

大脳生理学から見た数理モデルの問題点

改めて脳について調べてみました。

・脳の神経細胞（ニューラルネット）の数は、数百億個を超えると言われる。
→今回の実験で使用したニューラルネットは、10個くらい

・脳の神経細胞の電気的変化は、
ドーパミンをはじめとする様々な科学物質により変化する。
→今回の実験で使用したニューラルネットは、
入力信号とBP法で変化するだけ。

・人間をはじめとする生物の脳は、
数万年という長い時間をかけて作られたものである。
→今回の実験で使用したニューラルネットは、
1ヶ月ちょいくらいのインスタントニューラルネット

生物の脳の制作規模を考えれば、
ニューラルネットで作るAIは、ミジンコクラスか！？

ニューラルネットと神経細胞の比較

・生体と人工ニューラルネットの間の類似点

項目	生体ニューラルネット	人工ニューラルネット
素子の機能	全か無かの法則に従う(閾値素子)	
素子への入力	興奮性、抑制性作用の重み付け総和	

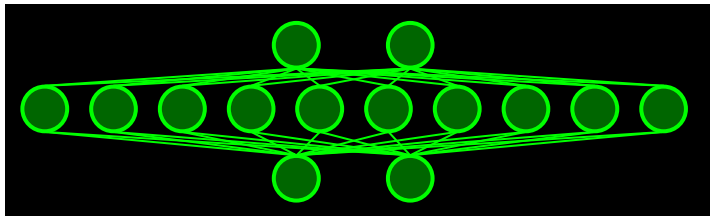
「表2-1 生体と人工ニューラルネットの間の類似点」より

・生体と人工ニューラルネットの間の相違点

項目	生体ニューラルネット	人工ニューラルネット
素子の種別	多種、興奮性と抑制性の別がある	通常一種
素子間相互作用	一方向	双方向の場合がある
素子の動作速度	低速	高速
素子間通信速度	低速	高速
結合素子数	多数(1000以上)	現状技術では小数

「表2-2 生体と人工ニューラルネットの間の相違点」より

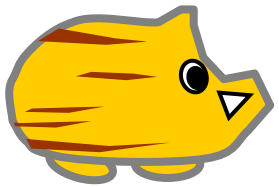
「学習とニューラルネットワーク」
熊沢　逸夫著(森北出版株式会社)



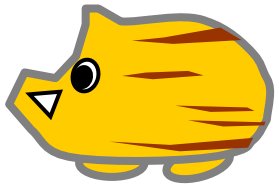
ニューラルネットの実体

ニューラルネットがどんなに発展しても
生物の脳と同じにはならない。

解決方法はないのか！？



パート3



猪突猛進!!

(前に進む)

キャラ化

ニューラルネット研究の楽しさ

実際、ニューラルネットのAIは作っていて楽しいです。

・ニューラルネットの実装の初期段階

「コイツほんとに馬鹿ばっかだなぁ。お、あいつ生き残った？」

・ニューラルネットを学習させて、なんとかそれっぽく動き始めた段階

「お、ゴールしそうなAIあるじゃん。いけるか？」
「あのAIダメそう・・・消えるのかなぁ。ちょっとかわいそう。」
「コイツ頭いいけど、ストレートすぎて面白くないなぁ」

・ニューラルネット調整（末期？）段階

「おおお、ちょっと生き物ぽいかも。」
「まだまだ、怪しい動作するけど、ちょっとかわいいなぁ」

この気持ちをプレイヤーに抱かせるゲームを作った方が面白い！！

AIを使ったゲームでは、そのゲームをプレイすることよりも、
そのゲームのAIを作ることのほうが楽しいと思うことがある。
プログラマーなら、経験されている方が多いのではないでしょうか？

どうしたら、この気持ちを抱かせられるか？

このニューラルネットAIの開発過程をプレイヤーに伝えられないか？

そこで･･･考えました。

プレイヤーの感情移入をより強くするためには、

オプションの「キャラ化」を強く前面に出していくべきではないか？

「キャラ化」については、こちらの書籍が参考になりました。

バンダイキャラクター研究所所長の相原博之氏の著書です。

OLさんがエビちゃん化する理由～『キャラ化するニッポン』
相原博之著
講談社現代新書、700円（税別）

ISBN-10: 4062879107
ISBN-13: 978-4062879101

実験：この犬は、どんな感情？

Q. 下の犬は、今、どんな感情を持っているでしょうか？

結果

A. この犬は、写真ですので、感情はありません。

中の犬などいません。

キャラ化のメリット

「キャラ化」することで、下記のメリットを出すことができます。

・オプションを「ブタ」や「ハムスター」など、動物的なキャラにしたほうが
「キャラ化」できる。これは自然界を模倣したニューラルネットAIの利点になる。

・キャラ化することで、プレイヤーの感情に波を起こすことが出来る。

・ポケモン的要素を持つコンテンツに、
このニューラルネットを使ったゲーム企画はフィットする。
よって、現在、このニューラルネットのゲーム企画に一番良い素材は
「ポケモン」かも？

自機＝サトシ、オプション＝ポケモン

・「キャラ化」することは日本人の得意とする分野である。

キャラ化すると、どうなるか？

では、このシューティングゲームを「キャラ化」すると・・・

・オプションを「ブタ」などキャラとしてデザインすれば、感情移入がしやすくなる。

・プレイヤーから「ほめる」「しかる」などのAIへの評価が、より自然なアクションになる。
また、キャラとしてデザインされたオプションは、そのリアクションを表現しやすい。

例：オプションが自機の前をうろちょろしていて邪魔なので、「しかる」をした。
実は、AIは「自機の前で敵の弾から守ってあげよう」と思っていたので、
プレイヤーの教師信号を受け入れつつも「すねた」リアクションを返した。

・キャラ化されたオプションであれば「ほめてタイミング」も表現しやすく、
プレイヤーにとっても分かりやすいものになる。

・ニューロンAIのキャラ化により「浪花節」を表現できる可能性がある。

例：オプションAのブタばかり「ほめて」いたら、オプションBのブタがすねた。

これは･･･

これまでの考察より、下記のことが言えないでしょうか？

日本人ならではの「キャラ化」の手法を使えば、

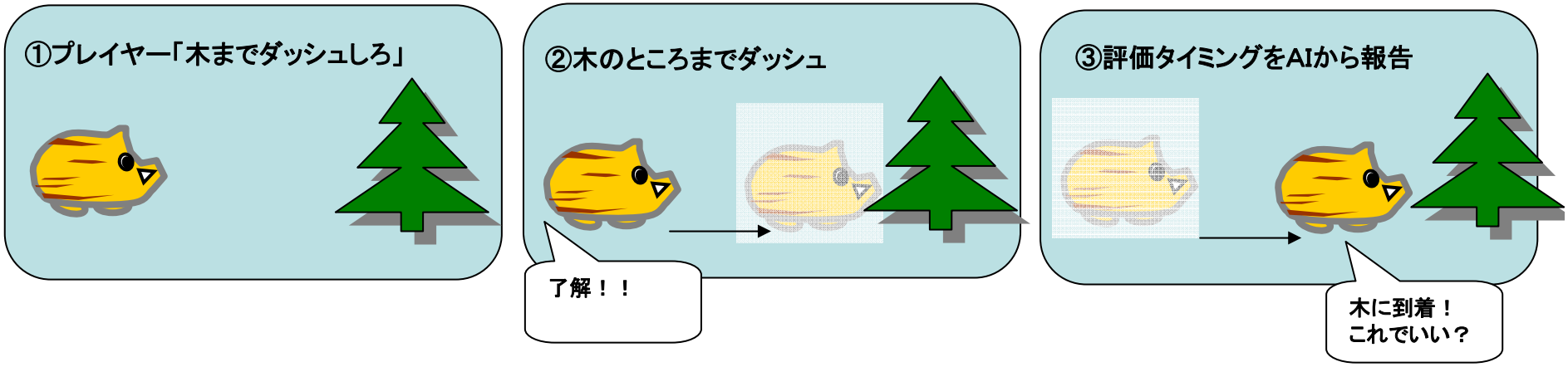
海外ではマネのできないニューロンAIの演出が可能になる。

「浪花節」を演出できるのは、日本人だけである。

これは「AI進化の過程を視角化する」
という新しいゲームを作ることに
つながるのではないか！？

キャラ化のポイント

キャラ化したAIとコミュニケーションを取れるように工夫します。



プレイヤーがキャラ化したAIと積極的にコミュニケーションが取れるように、
AIが何か行動をしたらAI側から「これでいいの？」とプレイヤーに問いかける。
（これが「ほめてタイミング」です）

これにより、プレイヤーは「ほめる」・「しかる」の反応を積極的に行える。

また、AI側から行動の評価タイミングをプレイヤーに教えているので、
AIの取ったどの行動に評価を下しているのか、
プレイヤーにもAIにも分かりやすくなる。

AIの「死」を演出する

さらにAIの「死」を演出することで、
ゲームの面白さにつながらないか考察してみました。

①そもそも「生」「死」は対になっているものであり、相反するものである。
故に、これまでニューロンAI系のゲームで表現されなかった「死」を表現すれば、
より「生」を対比して表現することで面白くなるのではないか？

②進化の過程が面白いのは、この「生」「死」を同時に見ているからではないか？
とくにニューロンAI系を使ったゲームでは、これらの過程を見ているのは、
プログラマーであり、実際、作る楽しみを感じられた。

今回のシューティング企画では、

リアルタイムでオプション（AI）の生と死を見せて、

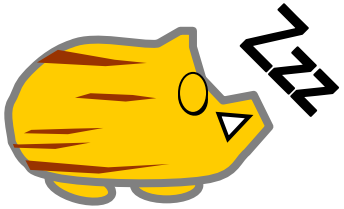
AI進化の過程も見ることができるかも！？

しかし・・・

遺伝的アルゴリズムの実装が間に合わなかったため、
「死」実装は見送りとなりました。

解決方法はないのか！？

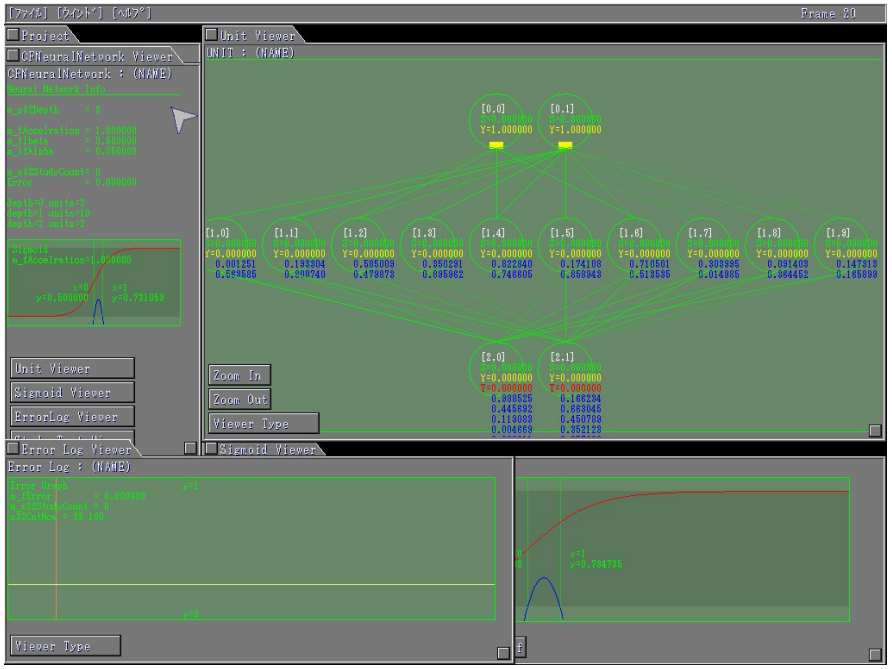
お詫びに「寝る」を実装しました。



さらなる挑戦

ビュワーをバージョンアップ！！

ニューラルネットのテストツールをバージョンアップしました。
C++でのニューラルネット作成だけでなく、Luaにも対応。
ツールから直接ニューラルネット構造データを保存できます。

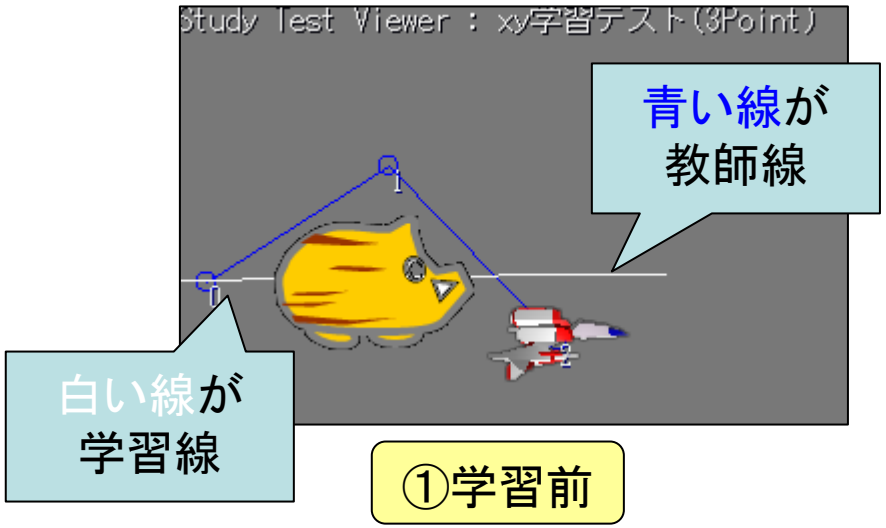


このビュワーで単体テストをしました。

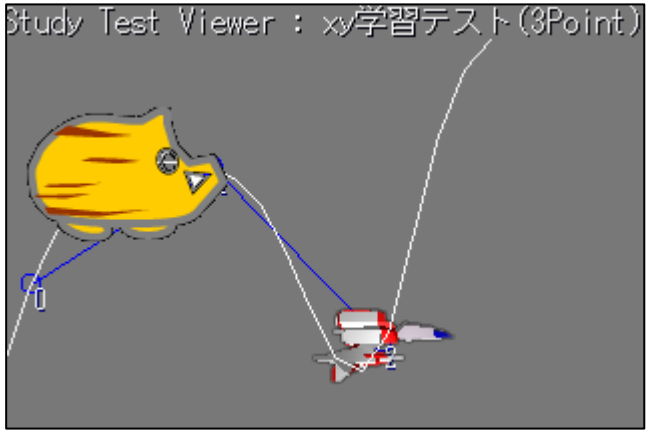
①3ポイントをつなぐ
②ホーミング・XY
③ホーミング・角度（失敗）
④ホーミング・触覚（成功）
⑤弾避け
⑥時間軸で記憶したポイント経由移動

実演

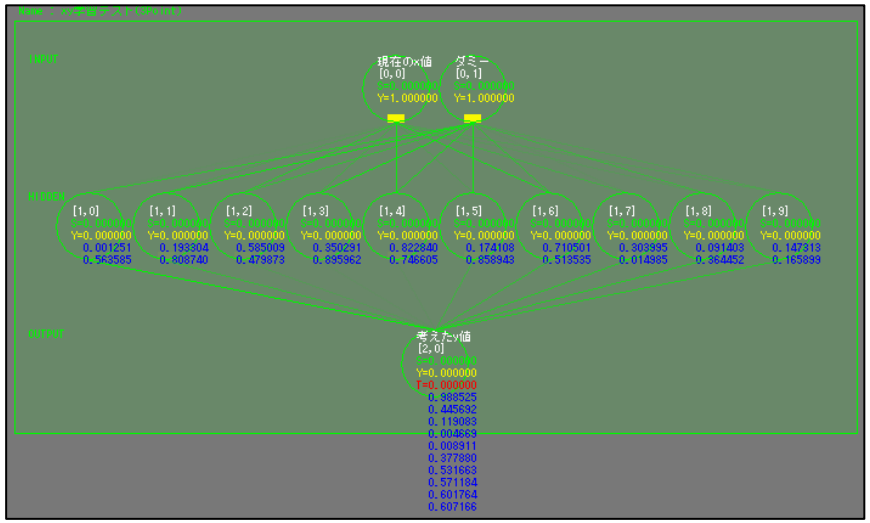
実演①：3ポイントをつなぐ移動



①学習前



②学習後



[ニューラルネットの構成]

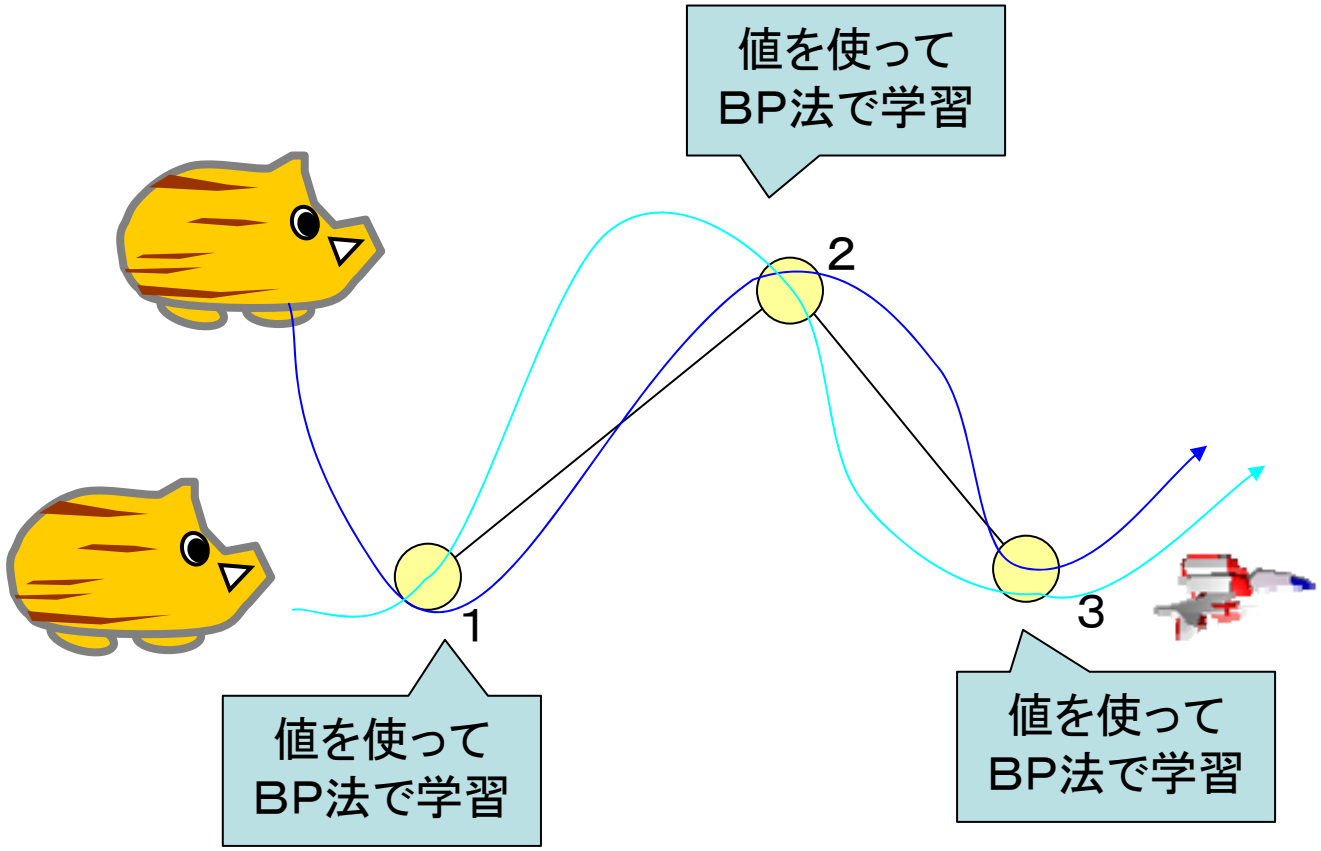
入力層	2	:X値、ダミー値（常に1)
隠れ層	10	:
出力層	1	:期待するX値

実演

解説

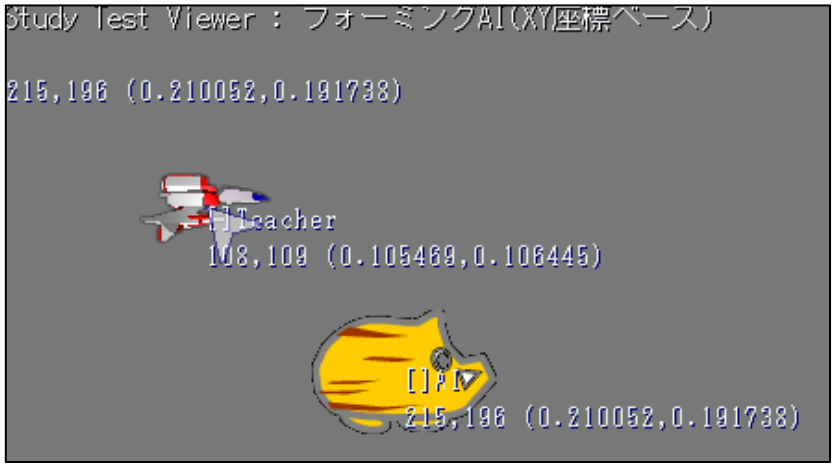
3ポイントと通る線を学習させます。

3ポイントのX座標（入力値）、Y座標（教師値・出力値）として、BP法で学習させます。

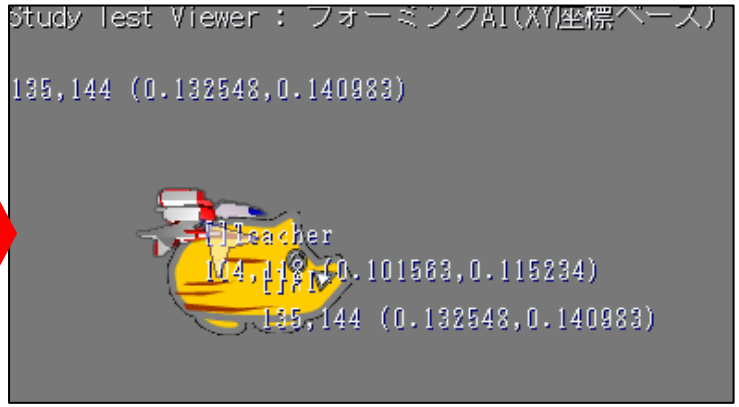


初期ウェイトの違いや、学習時間の違いが、個性を生みます。

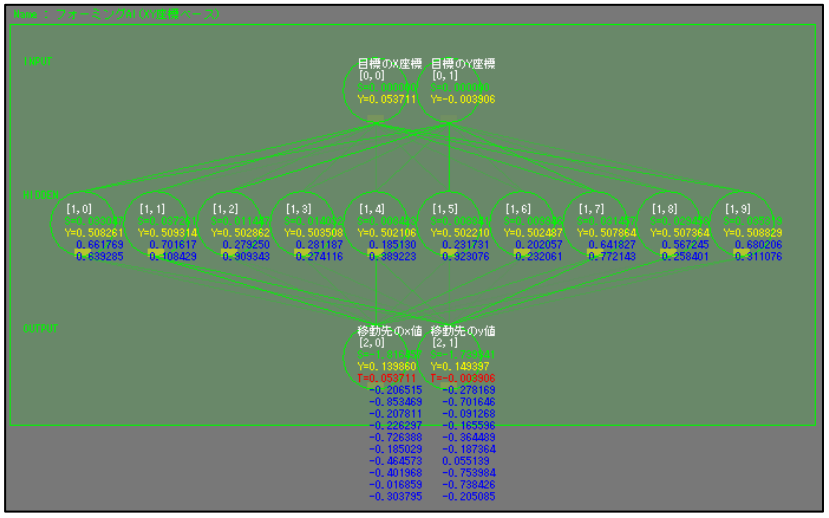
実演②：ホーミングXY



①学習前



②学習後



[ニューラルネットの構成]

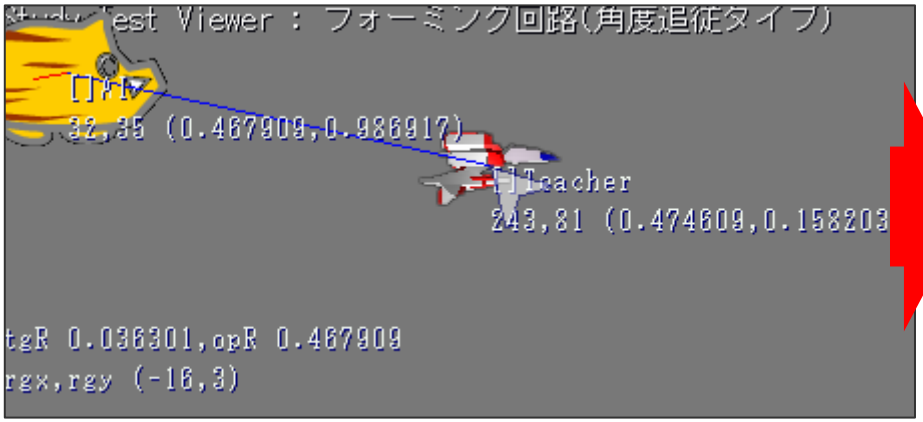
入力層	2	：ターゲットのXY値
隠れ層	10	：
出力層	2	：期待するXY値

実演

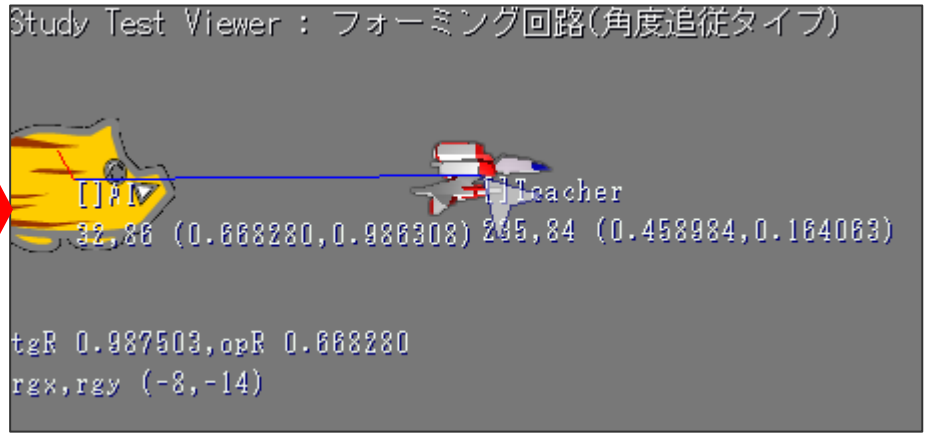
解説

「テストプログラムを作る」のホーミングニューラルネットと同じです。
そちらの解説をご覧ください。

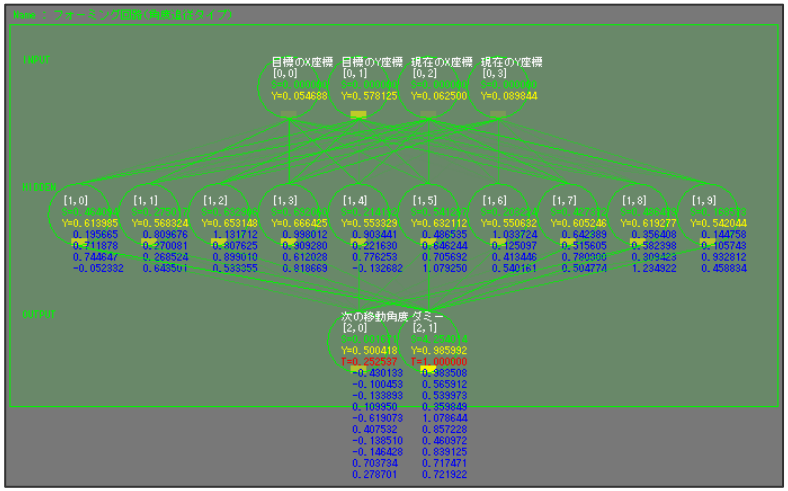
実演③：ホーミング角度（失敗）



①学習前



②学習後



[ニューラルネットの構成]

入力層　4　　：ターゲットのXY値
　　自分のXY値

隠れ層　10　　：

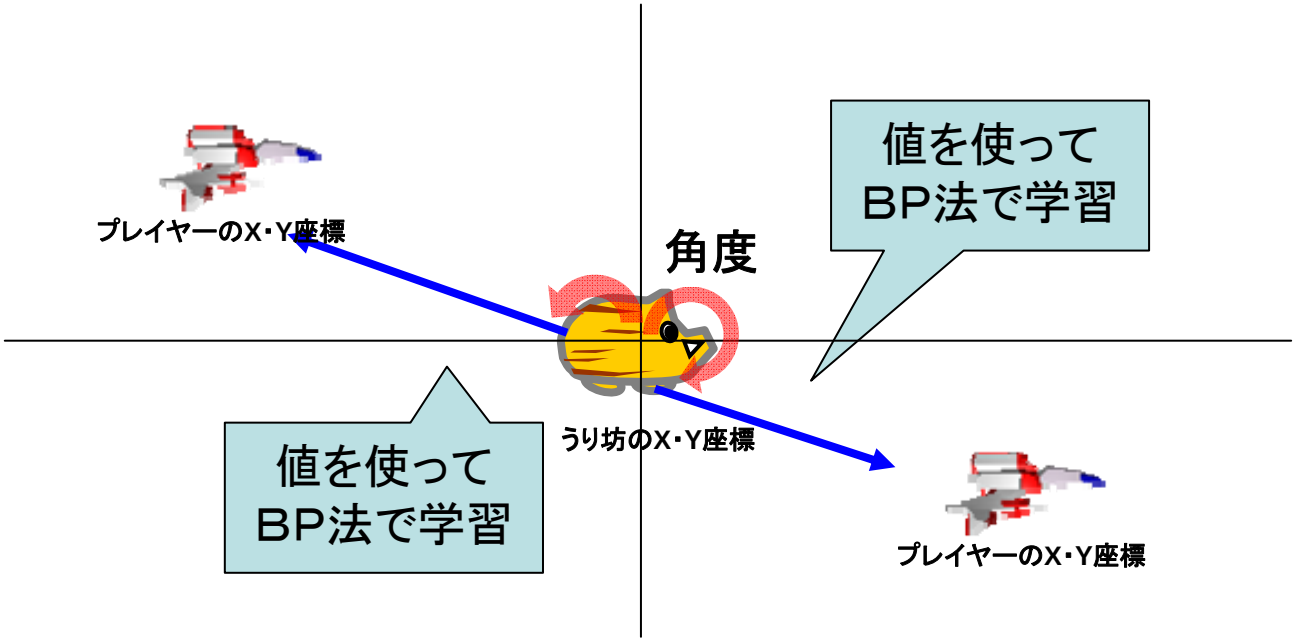
出力層　2　　：期待する角度値
　　ダミー(常に1.0を期待)

実演

解説

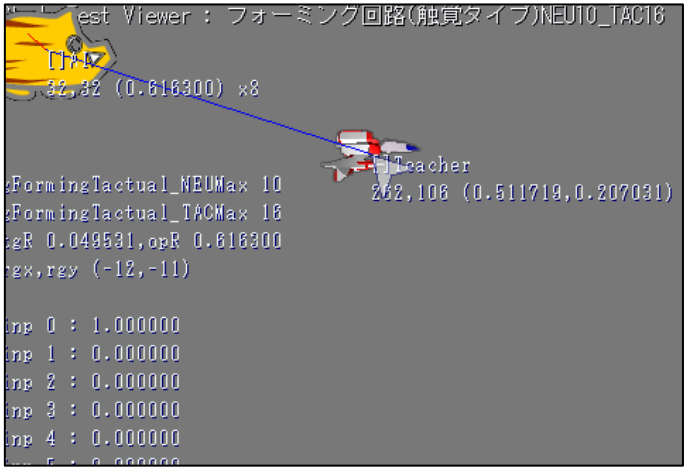
ターゲットの座標とうり坊の座標から角度を学習させます。

プレイヤーのX・Y座標（入力値）、うり坊のX・Y座標（入力値）、
そして角度（教師値・出力値）の3つで、BP法で学習させます。

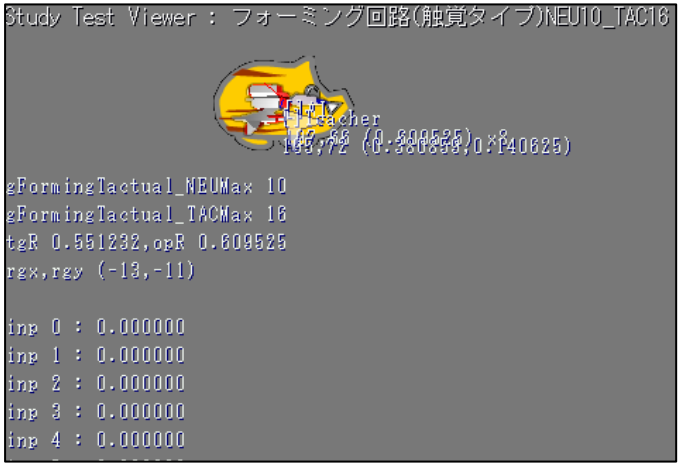


しかし、学習は収束しませんでした。失敗です。
推測なのですが、「角度のパターン数＝ターゲットの座標×うり坊の座標」
となるので、隠れ層のユニット数が足りないのかもしれません。
また、学習量も膨大に必要なのかもしれません。

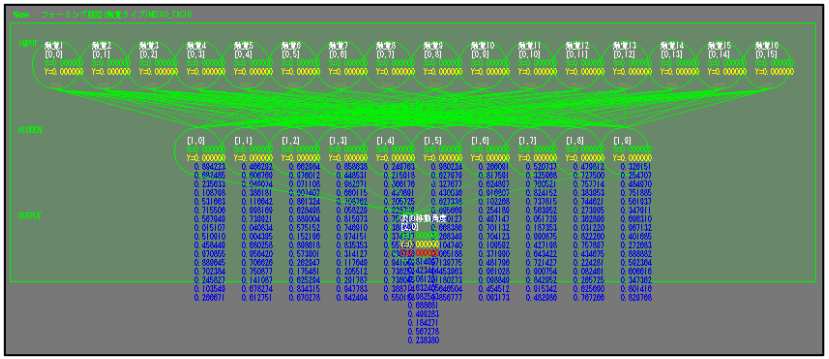
実演④：ホーミング触覚



①学習前



②学習後



[ニューラルネットの構成]

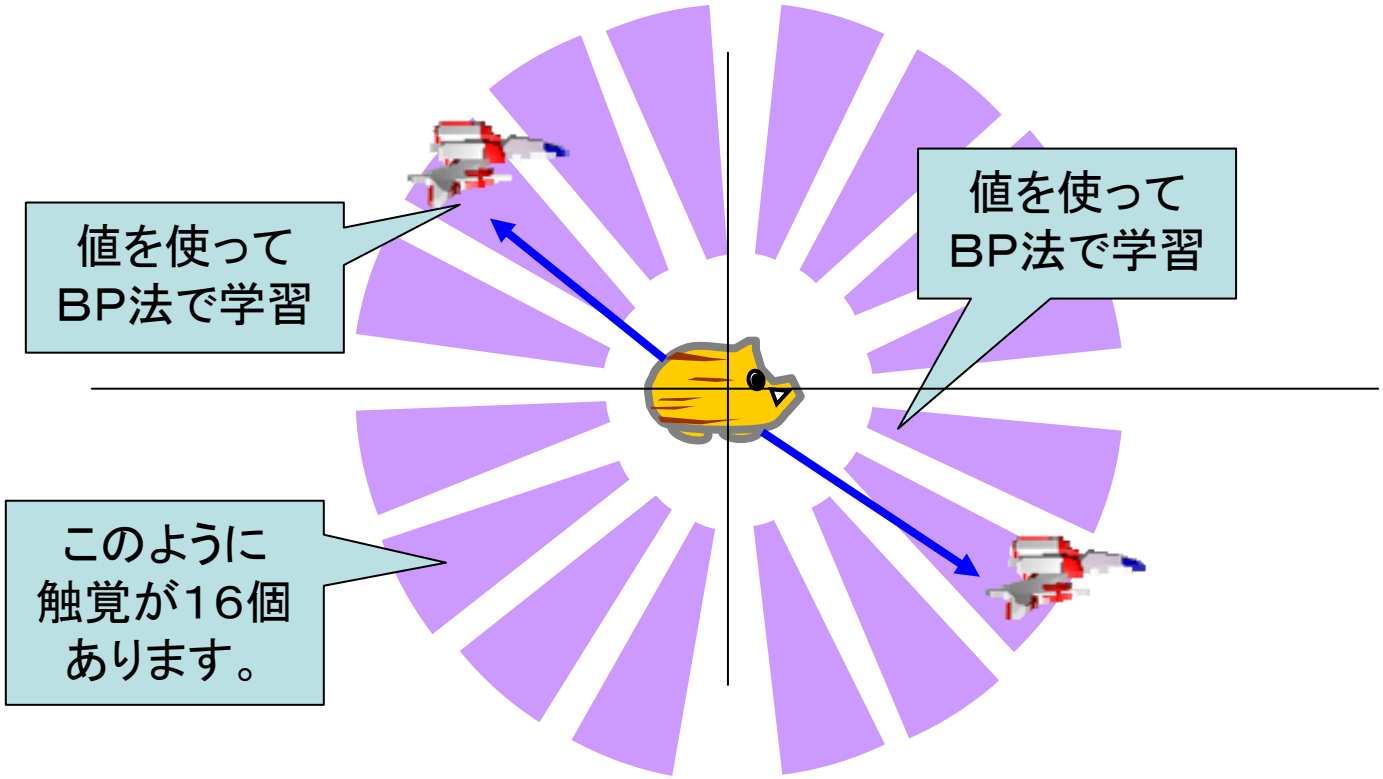
入力層　4　　：触覚16
隠れ層　10　　：
出力層　2　　：期待する角度値

実演

解説

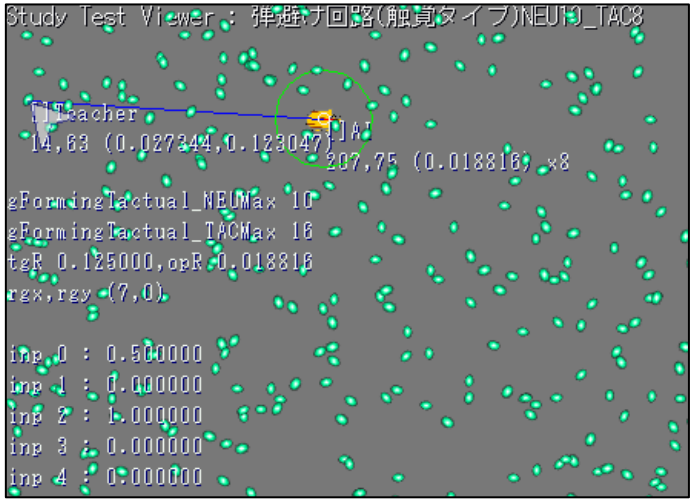
16個の触覚から角度を学習させます。

触覚に反応した角度を、BP法で学習させます。

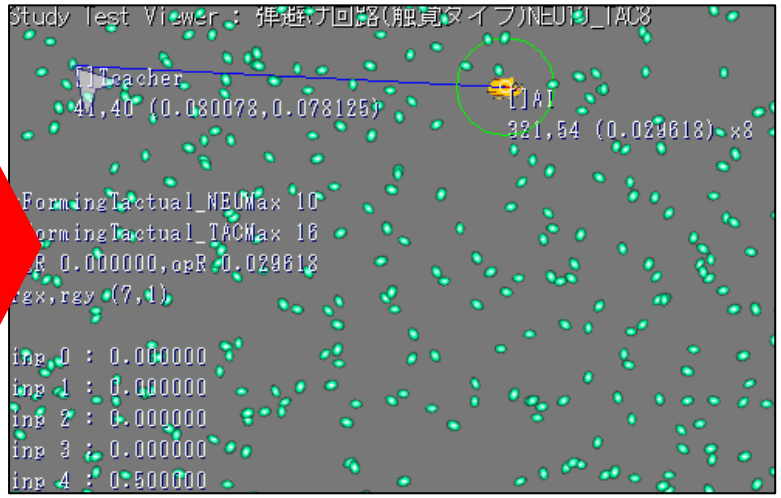


この方法だと、学習の収束が安定しました。
期待する結果のパターンが16個であることが、学習効率を上げていると思われます。

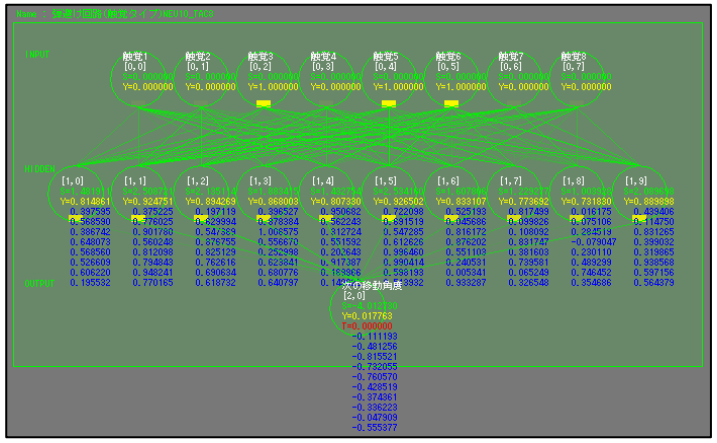
実演⑤：弾避け



①学習前



②学習後



[ニューラルネットの構成]

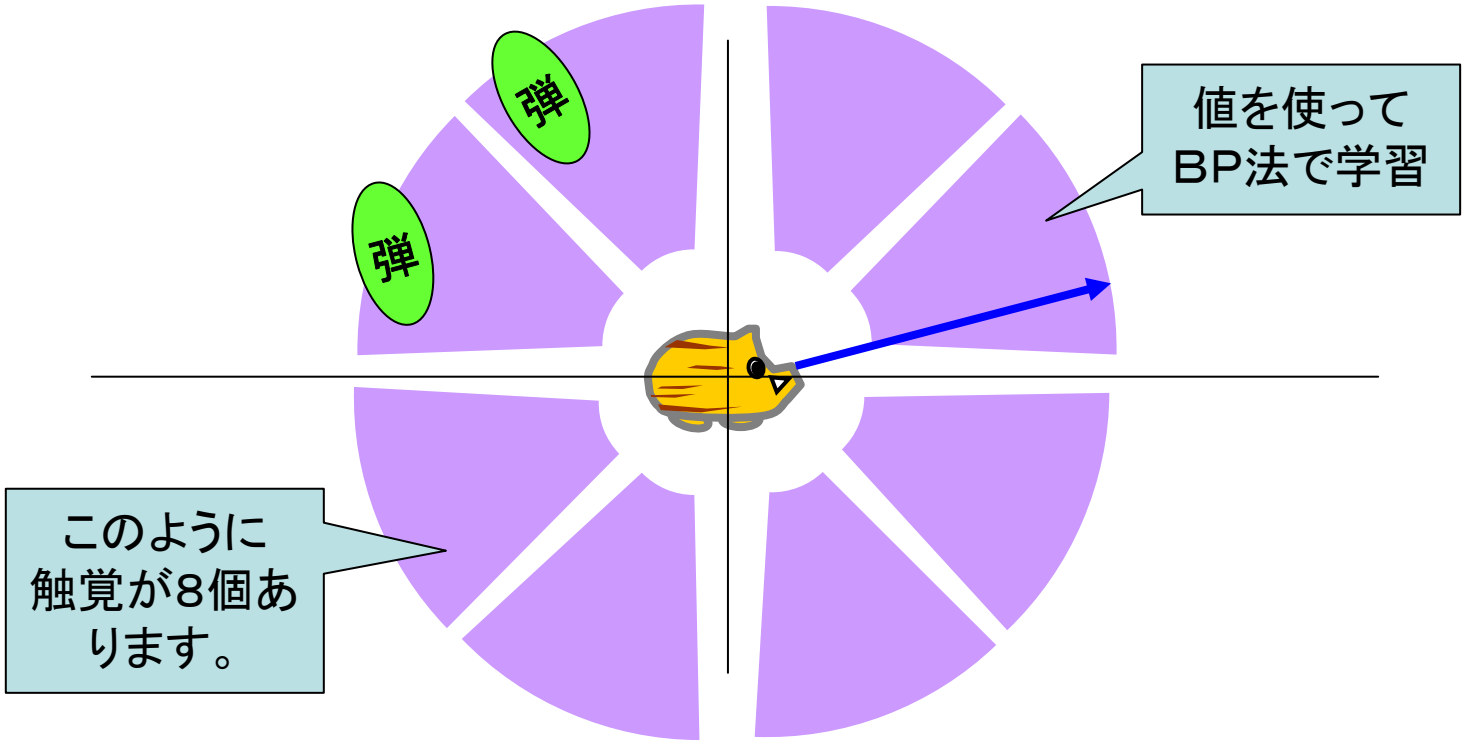
入力層	4	:触覚8
隠れ層	10	:
出力層	2	:期待する角度値

実演

解説

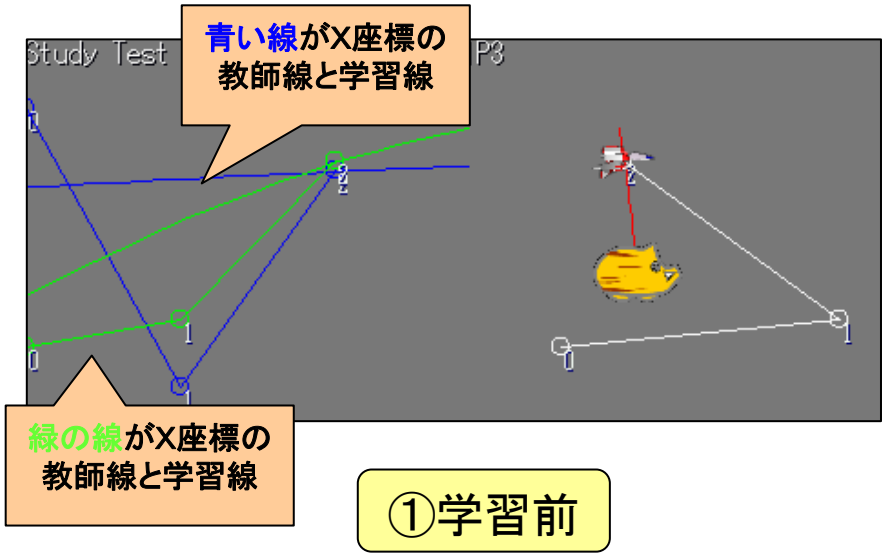
8個の触覚から角度を学習させます。

触覚に反応したら、弾のない方向の角度を教師信号として、BP法で学習させます。

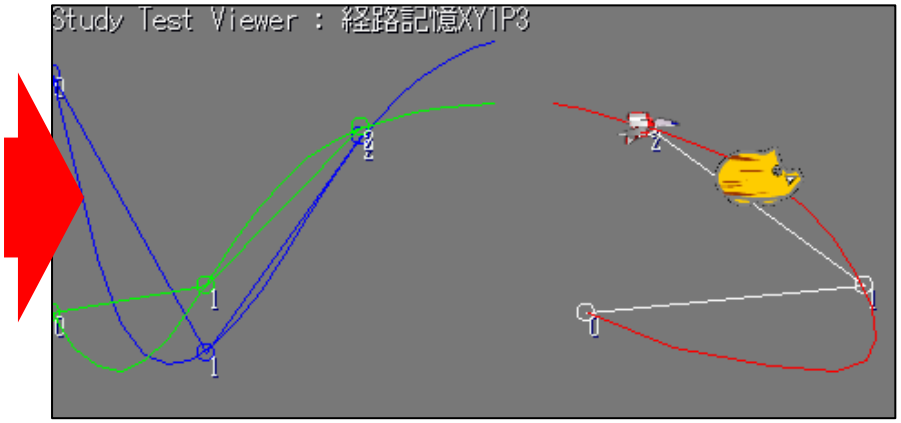


アルゴリズム的には、ホーミングと逆の処理をしています。
完全に弾をよけられるわけではありませんが、
避けている「ふり」は出来ているようです。

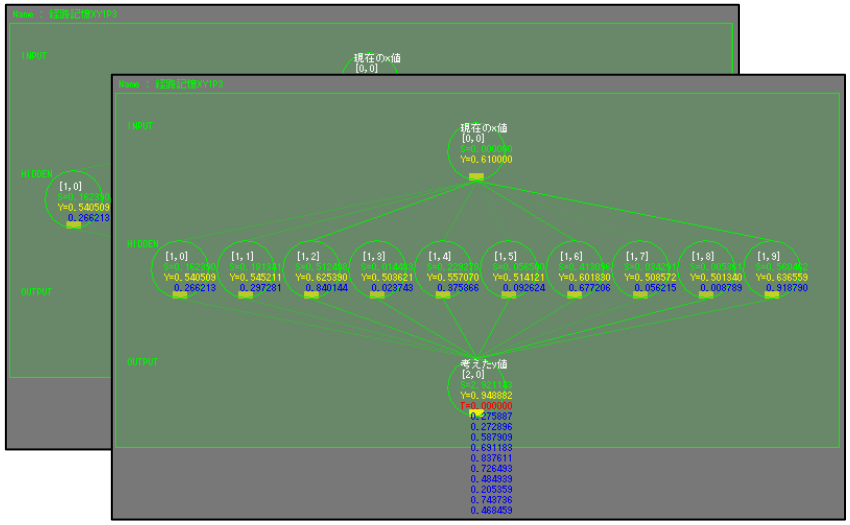
実演⑥：経路の記憶



①学習前



②学習後



[ニューラルネットの構成]
下記の構造のニューラルネットを、
X座標用とY座標用に2つ用意します。

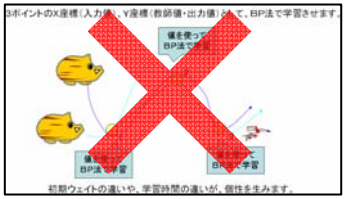
入力層	2	：t値（時間）
隠れ層	10	：
出力層	1	：期待するX値

実演

解説

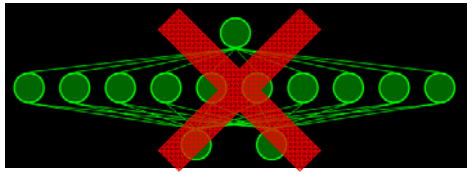
時間にあわせて3ポイントを通過する経路を学習します。

これを実現するためには、時間値・X値・Y値と3つのパラメータが必要になります。



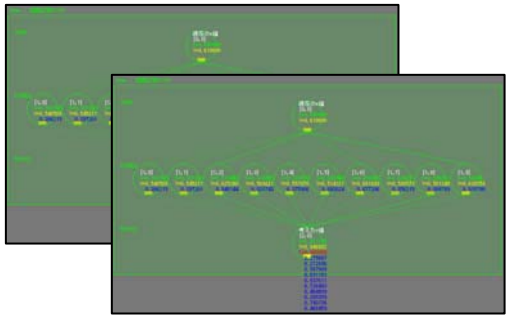
そのため、前に説明した2つのパラメータで動作する
ニューラルネットの方式ではそのまま利用できません。

そこで、入力信号を時間値、出力信号をX値・Y値のニューラルネットを作成しました。



しかし、1つのパラメータから2つのパラメータを出力する学習は
収束しませんでした。表現したいXを理解できなかったようです。

最終的には、単体テストで確認できたニューラルネットを利用しました。



「3ポイントを通る線」を学習するニューラルネットは、
学習が収束することが確認できていますので、これを利用しました。
X座標を時間として入力値に使い、2つのニューラルネットで
対応しました。
期待通り、経路の学習と記憶に成功しました。

単体テストの結果

単体テストで、下記のことが分かりました。

[できること]
・単純な学習
・学習することの「意味（Xの表現）」を、制作者が正しく理解しているものへの学習

[できないこと]
・複雑にからみあう事象の学習
・学習することの「意味（Xの表現）」を、製作者が理解していないものへの学習
・「学習することに意味づけがない事象」に対する学習
（「明日の天気」と「自分の給料がいつアップするか」の関係など。）

[問題点]
・教師信号は効果的な値を効果的な数だけ与える必要がある。
・学習に時間がかかる。
・企画がゲームの調整を行うのに時間がかかる。
・プログラムのデバッグがしにくい(バグの定義が明確でなくなることがある)

ニューラルネットに悩む

これまでの実験で、１つのニューラルネットだけで、
さまざまな絡み合うことをすべて学習させることは、
無理であることが分かりました。

では、どうすれば？

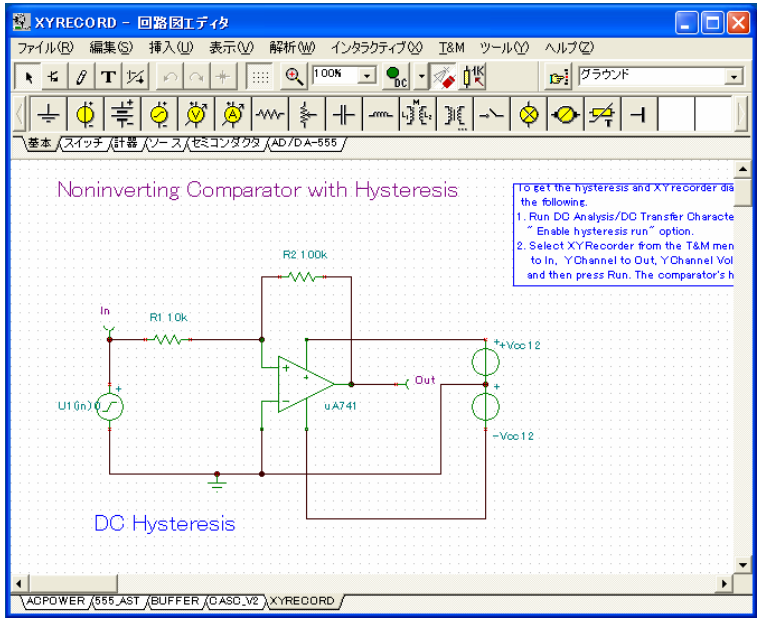
１つでダメなら、いっぱい作ってみたらどうだろう？

そこで考えました

電気回路のように、
複数のニューラルネットのアルゴリズムを組み合わせたら？

Tina

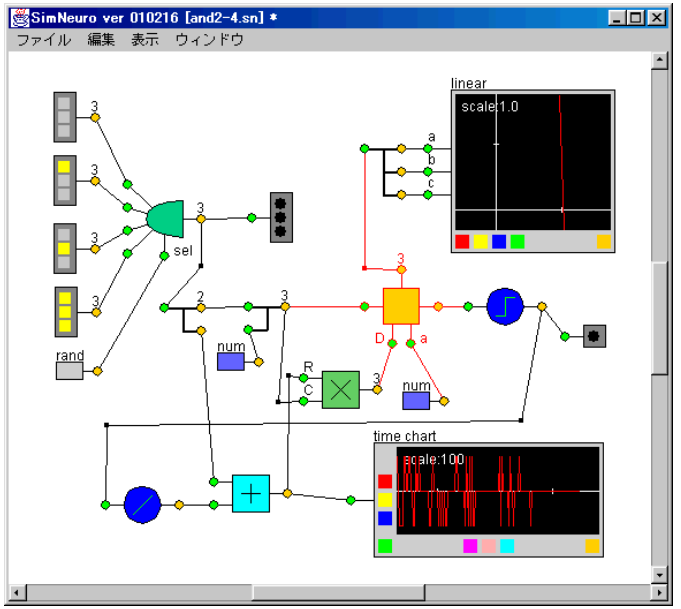
DesignSoft社（ハンガリー）のTINAは高機能かつ大変に使いやすく、その上に「安価」な、統合回路設計ツールです。
回路図入力、Spice3F5/XSpiceをベースにした回路シミュレーション、PCB自動配置配線の3つの機能を統合しています。
プロ用と教育用があります。



http://www.ilink.co.jp/tina/index.html

Sim Neuro

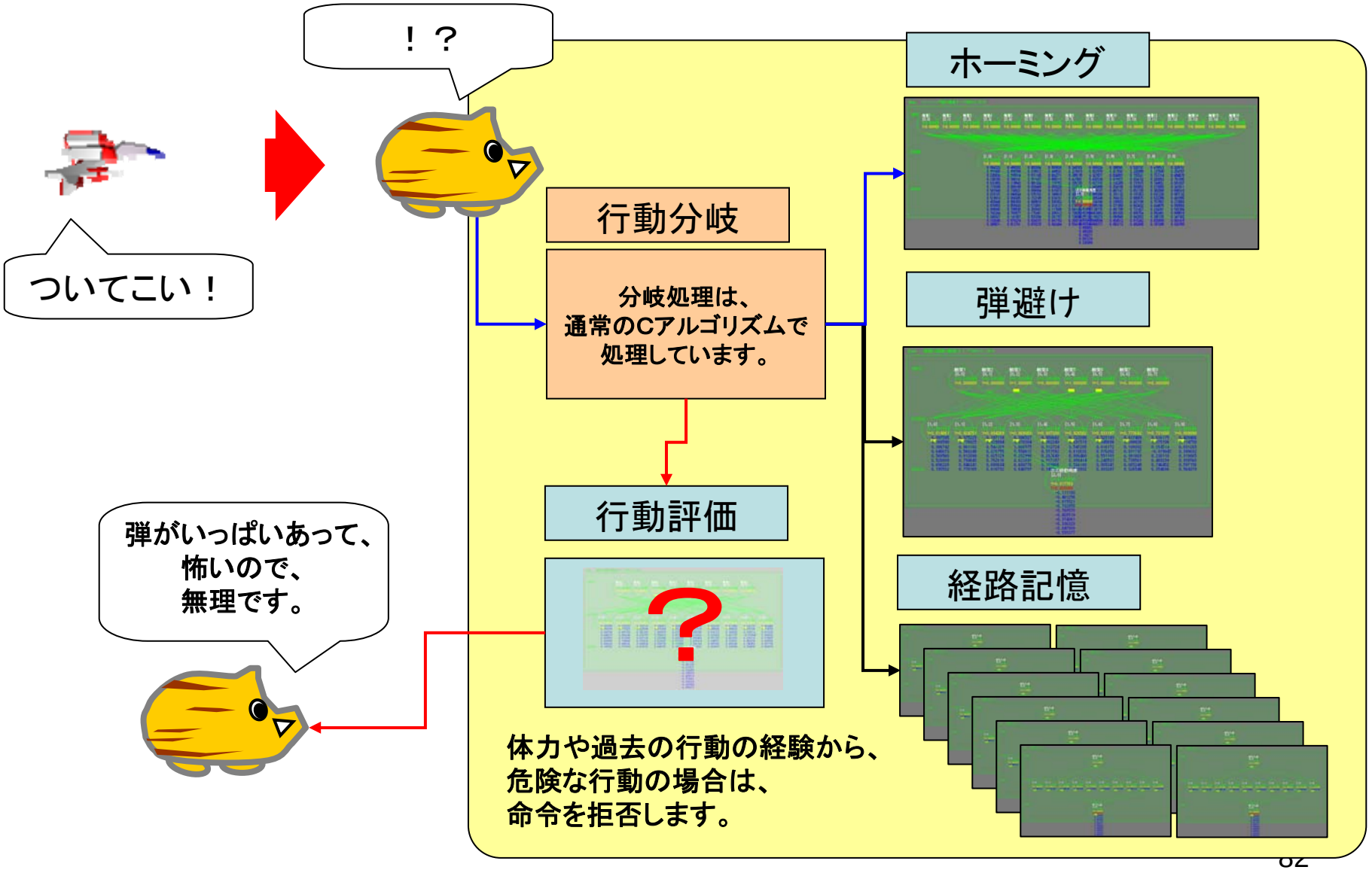
SimNeuro は、Java で書かれた、
小規模なニューラルネット用のシミュレータです。
いくつかのユニットをマウスで並べて接続していくことで、
簡単にニューラルネットワークの実験を行うことが出来ます。



http://suuri.ics.kagoshima-u.ac.jp/research/neuro/SimNeuro/index.html

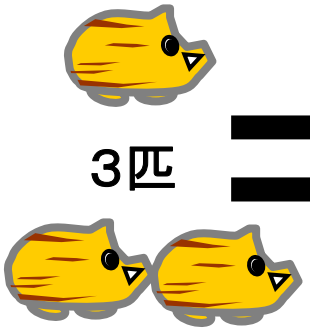
ニューラルネットを組み合わせる

単体テストを元に、下記のようにニューラルネットを組み合わせました。



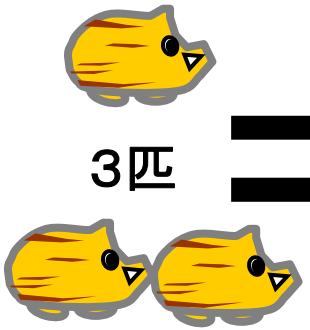
数えてみたら・・・

ホーミングニューラルネット	(1)
弾避けニューラルネット	(1)
行動判定評価ニューラルネット	(1)
経路記憶ニューラルネット	(256)

×  ＝777

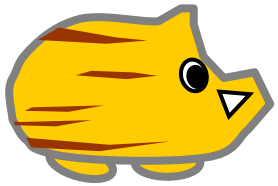
総数777ニューロンになりました。
なんか、ちょっと嬉しい。
・・・ん？

ホーミングニューラルネット	(1)
弾避けニューラルネット	(1)
行動判定評価ニューラルネット	(1)
経路記憶ニューラルネット	(256x2)

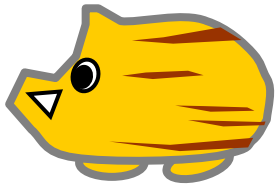
×  ＝1545

よく考えたら、経路記憶ニューラルネットは2個組で使っていました。
777・・・じゃない・・・ちょっと寂しいです。

実演



ニューバージョン
「スペースうり坊」



①ホーミング学習するところまでプレイ
②弾避けを学習
③経路記憶→睡眠→経路再開
④学習したデータでミッションクリア
⑤行動判定ニューロンの「嫌だ」を説明

実演

実演①：ホーミングを学習

まずは敵のいないステージで、「うり坊」に「ついていく」ことを覚えさせます。

①まず、うり坊から離れた位置に移動します。

②「こっちにおいで」命令を出します。
うり坊が、「了解！」と返事します。

③プレイヤーと一定距離まで近づくと
「ほめてタイミング」となり、プレイヤーの反応を伺います。

実演

解説

このホーミングの動作は、ニューラルネットビュワーで見た
ホーミングのニューラルネットとまったく同じです。

実験的に「ほめてタイミング」でプレイヤーが「ほめる」をしておくと、
行動判定評価ニューラルネットで、ストレス値が上昇しにくいように設計しています。

例えば、ホーミングを繰り返すことで、敵の弾にヒットしてしまう場合が続くと、
行動評価ニューロンは、この行動を危険と解釈してホーミングをキャンセルします。

しかし、危険のない状態でのホーミングを「ほめる」と、
安全であることを状況として理解して、
ストレス値を下げる方向へ向かわせることができると思われます。

（まだ実験段階の処理のため、裏付けはとれていません）

実演②：弾避けを学習

弾幕の激しいステージで、「うり坊」に「弾避け」ことを覚えさせます。

①うり坊は、命令を受けていない状態
であれば、弾避け行動を行います。

②弾が自分の危険距離内に入ると、
触覚を使って、弾のない方向へ避ける動作をします。

実演

解説

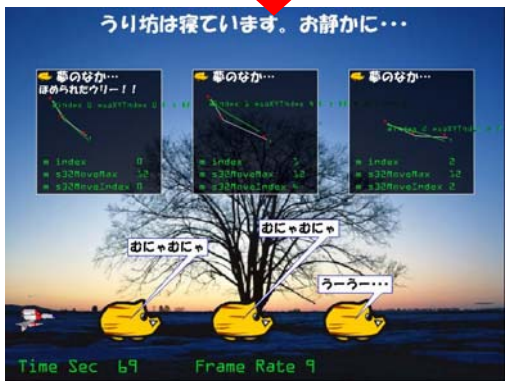
弾避けの学習は、さまざまな状況で学習したほうが、
効率がよいため弾幕ステージで学習させています。
（通常のステージでも学習は可能です）

なお、完全に弾を避けられるわけではなく、
弾を避けているかのような行動ができることを目標としました。

また、弾避け時には体力が減ることが多いため、
適度に「ほめる」をしておくことで、
ストレス値を上げないようにします。

実演③：経路記憶→睡眠→学習

ステージ中に「エサ」を置くことで、うり坊に「攻略経路」を教えます。



①敵の出現時間などにあわせて、敵を効果的に
倒すための「攻略経路」を「エサ」を置いて教えます。
うり坊は、エサにつられて攻略経路を覚えます。
（エサまで移動はホーミングニューロンを使います）

②エサをたべたところで、「ほめてタイミング」となります。
「ほめる」と、後で、その攻略経路を積極敵に学習します。
「しかる」と、適当に学習して終わります。

実演

③ステージを終了させると、うり坊は「睡眠」を取ります。
このとき、うり坊は教わった攻略経路をBP法で学習します。
このときの「ほめられた」「おこられた」かで、
攻略経路の学習量を調整しています。
また、睡眠時間で、攻略経路の記憶度が決まるので、
早く起こしすぎると、経路を正しく学習できません。

解説

攻略経路の記憶は、非常に重たい処理です。
本来ならアルゴリズムの効率化や、
プログラムの高速化などで対応すべきところです。

しかし、「キャラ化」した「うり坊」では、
「睡眠学習」という企画対応ができました。

「寝る子は良く育つ」は本当かもしれません。
脳科学の分野では、「睡眠」は、その日の活動記憶から、
記憶を評価して、重要な記憶は学習して理解（汎化）し、
あまり重要でない記憶は忘れるという説があります。

また、夢は、その学習過程を見ているのではないかとの説もあります。

今回のプログラムでは、この説をビジュアル化してみました。

実演④：経路再生

うり坊は、経路記憶をつねに参照して、
経路記憶の再生時間になると、その経路を辿り始めます

①うり坊は、毎フレーム「経路記憶の再生時間」であるか、
チェックしています。

②経路記憶の再生時間であることを判断したら、
「経路を思い出した！」とプレイヤーに伝えて、
経路記憶を再生します。
ただしく、経路を通れるかは、睡眠学習の結果次第です。

③経路の再生が終わると、「ほめてタイミング」です。
経路の再生がプレイヤーの意図に近ければ「ほめる」。
ぜんぜん違うようであれば「しかる」をします。

実演

解説

攻略経路の再生タイミングは時間で管理しています。
ただし、この最初の記憶再生タイミング時間だけは、
数値データとしてダイレクトに持っています。

この仕様にしたのは、この再生タイミング時間さえもニューラルネットで
学習させてしまうと、誤差が大きすぎてゲームにならないからです。

また、攻略経路の記憶を再生した場合も、
うり坊は「ほめてタイミング」でプレイヤーに評価を求めます。
このときの評価で、さらに「睡眠学習」での経路の学習量が決定します。

また、各うり坊には、好きな「エサ」があります。
ディフォルトでは、3匹分のエサを同時に設置しますが、
好きなエサを個別に設置すれば、
3匹それぞれに違う攻略経路を学習させることもできます。

実演⑤：嫌だ

うり坊は、「行動評価ニューラルネット」で、常に行動に対して評価をしています。

①うり坊は、毎フレーム「行動評価ニューラルネット」で、
うり坊の体力と行動、そしてプレイヤーの行動から、
「ストレス値」を出しています。

②このストレス値は、前フレームと現在のフレームでの
体力差を基準として、出力値の教師値として使っています。
そのため、左図のような弾幕攻撃を受けている状態だと・・・

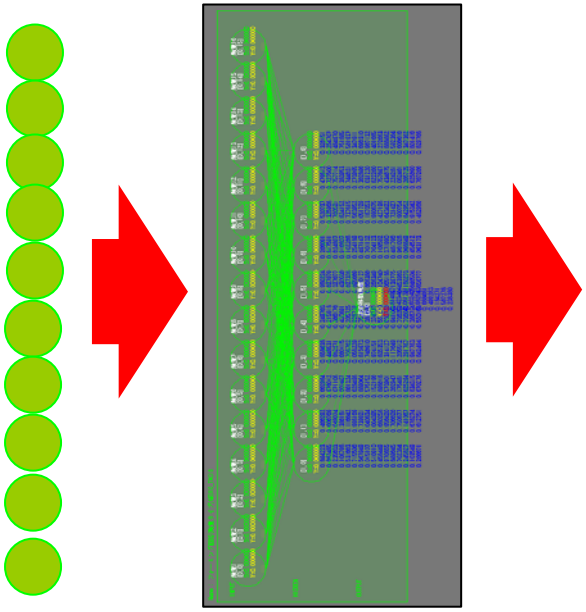
③プレイヤーに「こっちにおでい」といわれても、
体力が減少して、ストレス値が上がっているので、
命令をキャンセルして、その場に止まろうとします。
またこの状態を把握して、うり坊に「嫌だ」と言わせます。

実演

解説

行動評価ニューラルネットは、うり坊の生存力を意志として
持たせるために作成しました。
このニューラルネットでは「死に近い行動を体験するとストレスになる」として扱います。

プレイヤー命令：ついてこいホーミング
プレイヤー命令：えさをたどれ
プレイヤー反応：ほめた
プレイヤー反応：しかった
うり坊行動：ついていく
うり坊行動：えさみっけ
うり坊行動：記憶再生中
うり坊行動：弾避けだ
うり坊情報：現在の体力
うり坊情報：前フレームの体力



ストレス値

教師値は、うり坊が受けた
ダメージ値を使っています。

図は参考用です

このニューラルネットのみ単体テストができませんでした。
そのため、現在、動作の確認、裏付けは出来ていません。
とりあえずそれっぽくは動いているようです。

うり坊のセリフについて

うり坊は、いろいろなセリフを話ますが、
そのタイミングはニューラルネットが行動を実行した後に行っています。

通常のAIでは、企画やプログラマーの意図に基づき
行動を決定する場合がほとんどです。

例えば、キャラが「嫌だ」という場合は、プログラムは先に「嫌だ」と処理を決めて、
行動のアルゴリズムを実行しています。

しかし、ニューラルネットの場合は、局所的にその動作だけを見ても、
その行動が「嫌だ」なのかどうか、分かりません。

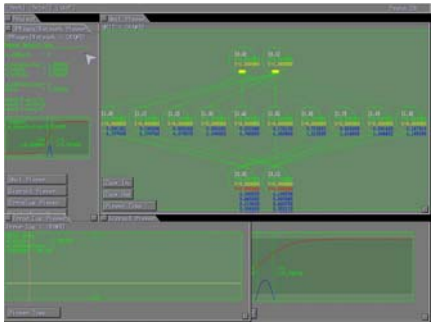
そこで、今回は、一部「嫌だ」などのセリフでは
ニューラルネットが行動をした後に判定を「行動評価ニューラルネット」で行い、
その状況が動物や人間的にあてはめて「どういう状態であるか」
ということを見て選択しています。

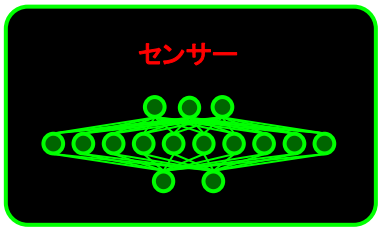
このように、先に行動があり、その後に感情が発生すると考える心理学を
「行動（主義）心理学」といいます。

まとめ

ニューラルネットを複数実装するために、
とにかく単体テストと実装を繰り返しました。

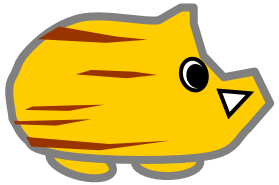
①単体テスト



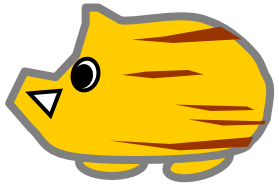


②実装

単体テストで、実際の動作の確認ができたことで、
ニューラルネットの企画的にどう調整すれば良いかわかりました。



パート4



まとめ

ニューラルネット
まとめ

できること・できないこと

ニューラルネットにできること・できないことをまとめました。

[できること]
・単純な学習
・学習することの「意味（Xの表現）」を、制作者が正しく理解しているものへの学習

[できないこと]
・複雑にからみあう事象の学習
・学習することの「意味（Xの表現）」を、製作者が理解していないものへの学習
・「学習することに意味づけがない事象」に対する学習
（「明日の天気」と「自分の給料がいつアップするか」の関係など。）

[問題点]
・教師信号は効果的な値を効果的な数だけ与える必要がある。
・学習に時間がかかる。
・企画がゲームの調整を行うのに時間がかかる。
・プログラムのデバッグがしにくい(バグの定義が明確でなくなることがある)

できるために、どうするか？

もし、ニューラルネットの研究ではなく、
ゲームに製品として実装するのであれば、次のことに注意しましょう。

・学習させる内容を企画的に明確化する。
→何を学習させて、どのようにAIを賢くしたいのか明確化する。

・上記の学習内容が、ニューラルネットにふさわしいか検討する。
→他に代用できるアルゴリズムがあるなら代用する。
製作コスト面での問題も考えておく。
製作コストが極端に高くなるなら、企画自体の変更も検討。

・実装するニューラルネットの単体テストを必ず企画とプログラマーで行う。
→単体テストでバグだけでなく、
実際に「どう使えるのか」「どのように調整するのか」も確認できます。

・複数のアルゴリズムに分割してニューラルネットを組み合わせる。
→複雑な関連性をもつ事象の学習は時間がかかりコストもかかります。
アルゴリズムを分割しておけば、差し替えなどの方法で、
仕様変更や調整にともなうリスクを幾分かは減らせます。

AIを活かすための企画仕様とキャラ化は必須

もし、ニューラルネットのAIを十分に活用したいと思うなら、
そのための企画仕様とキャラ化は、避けては通れません。

例えば、「じゃんけん」は、どんなにプログラマーががんばっても、
表現の種類がグー・チョキ・パーの3種類しかありません。
例えその背後に、人間に匹敵するほどのAIがあったとしても、
プレイヤーはそこに知能を感じるかは疑問です。

しかし、この「じゃんけん」でグーチョキパーの手だけでなく、
顔の表情や体のしぐさを企画仕様として盛り込んでおけば、
「緊張しているこのキャラは、このとき焦ってパーを出す」
などの表現が可能となります。

で、たぶんこうなる

ニューラルネットを正しく理解せずにゲームに使うと・・・

企画　　:「ニューラルネットならなんでも学習してくれるんでしょ？」

プログラマ　　:「よしやってみよう！！」

:　　ニューラルネットやってみた

プログラマ　　:「なんにもできないよう・・・」

企画　　:「でもCEDECでなんでもできるっていってたぞ」

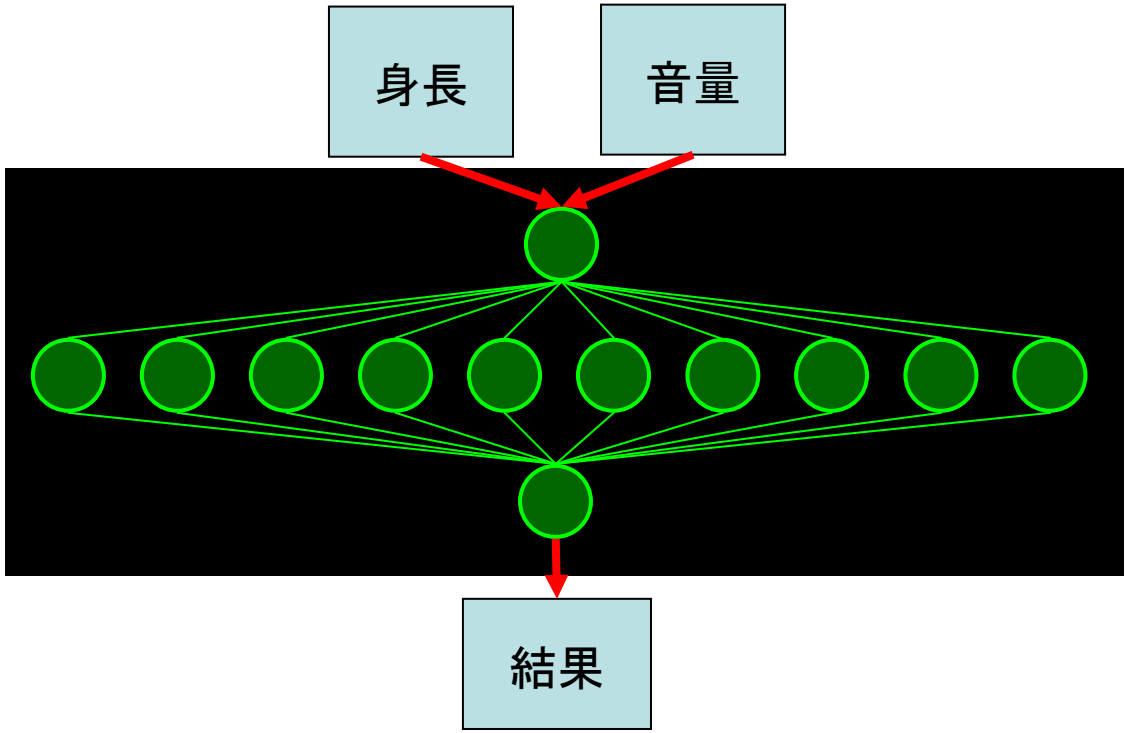
くんずほぐれつのどつきあい

ニューラルネットの将来性

ニューラルネットのもう一つのメリット

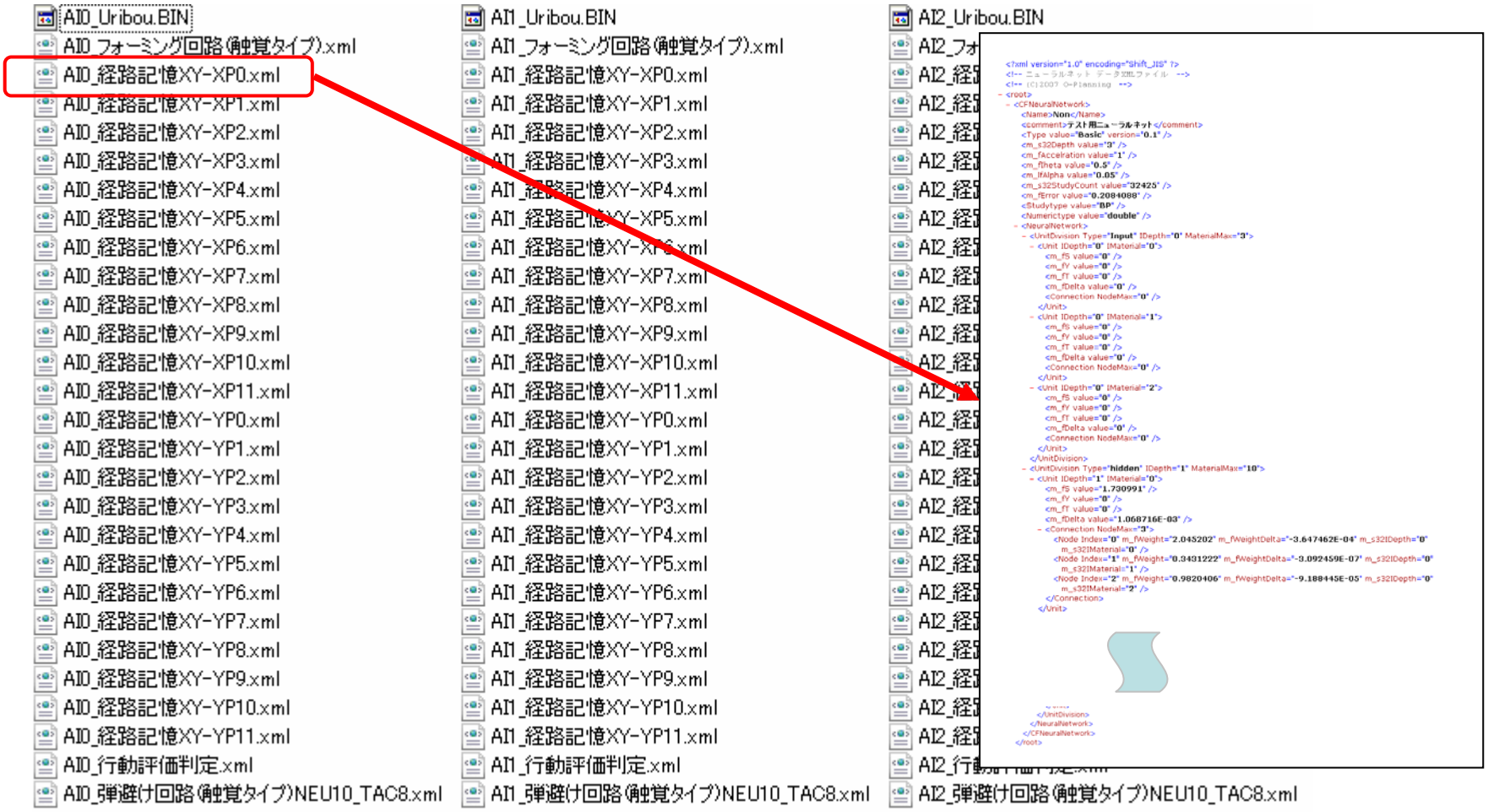
ニューラルネットのもう一つのメリットとして、
入力データと出力データを0から1の数値で扱うため、
固有データへの依存度が非常に低いことがあげられます。

例えば、「大きなものを判定するニューラルネット」を作った場合、
データが「身長」でも、「音量」でも、これらの数値を一度0から1の値に正規化して扱うため、
異なる意味のデータにもかかわらず学習して結果を出すことができます。



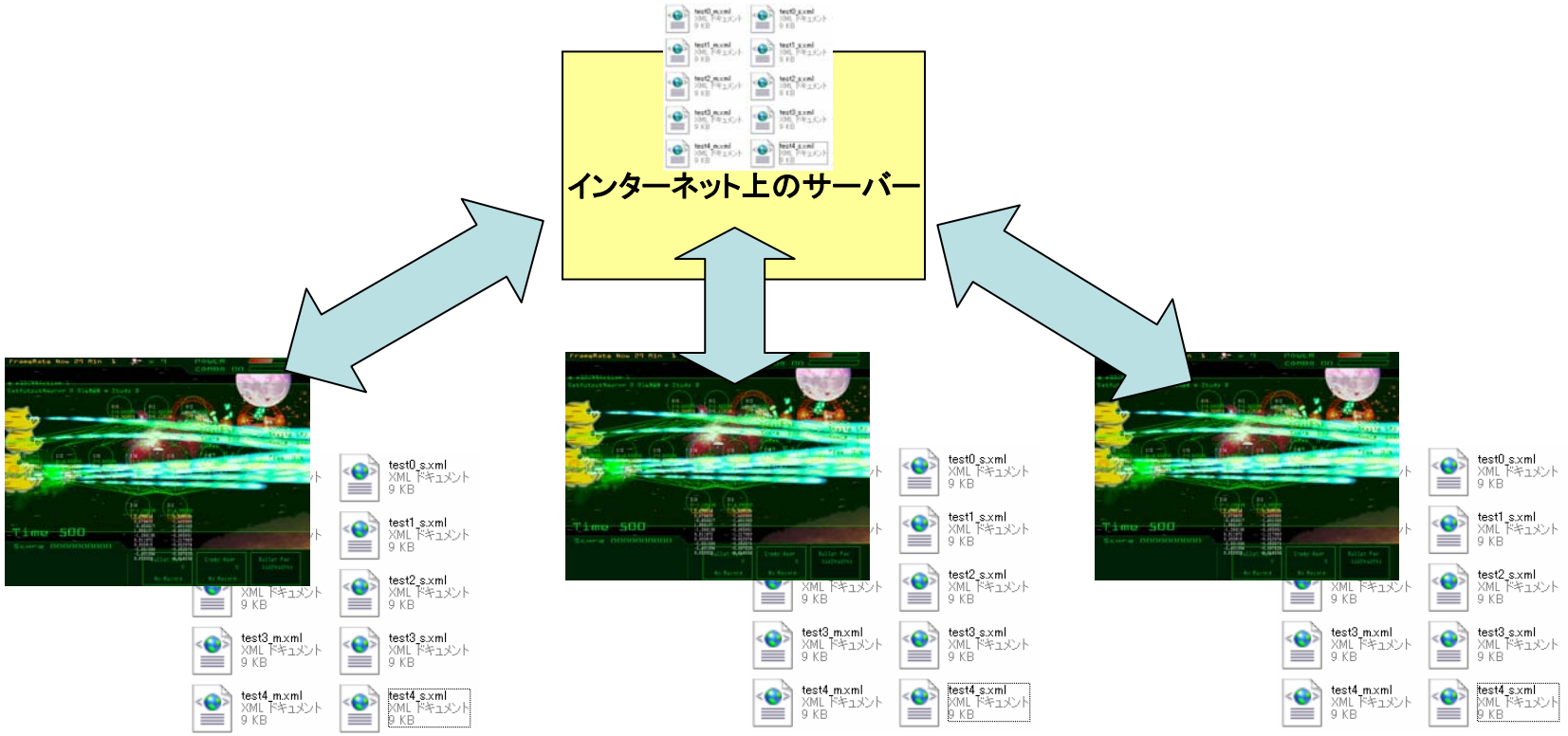
AIデータはXMLで保存

学習させたニューラルネットはXMLで保存しています。
メタデータ化しているので、汎用性が高くなります。



XMLでニューラルネットを共有①

このニューラルネットデータは、MSXMLのDOMパーサーでアクセスしています。
これにより、ローカルデバイス以外にも、
インターネット上のサーバーからニューラルネットデータを取得することが可能です。



ネットのサーバー上で各プレイヤーが学習したニューラルネットデータ共有することで、
「学習したデータの共有化」や「一番かしこい子のダウンロード」「アホな子のダウンロード」ができます。
この技術を敵AIに使えば、ユーザー自身が敵AIを無意識に作って評価することが可能です。

XMLでニューラルネットを共有②

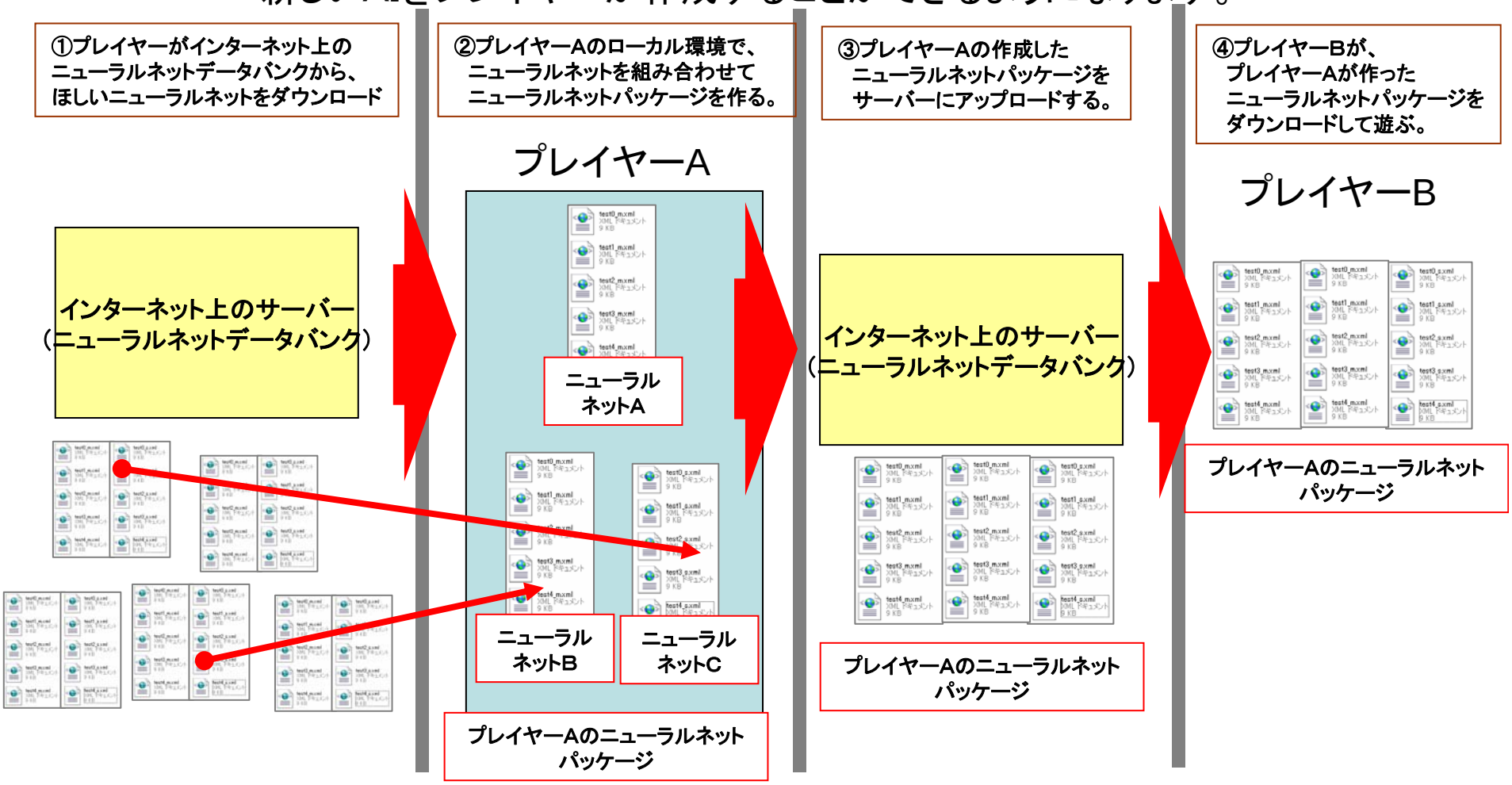
ニューラルネットAIでは、メタ化して扱うことができるため、
違うゲームのアルゴリズムとしても扱うことができます。

プレイヤーA　　プレイヤーB　　プレイヤーC

例えば、同じキャラクターが登場する3つのことなるゲームがあっても、
ニューラルネットAIをXML化しておけば、
共通して使えるアルゴリズム部分を流用することができます。

UCC（ユーザークリエイトコンテンツ）

ニューラルネットAIはUCC（ユーザークリエイトコンテンツ）に向いています。
XMLで保存した様々なニューラルネットAIをプレイヤーが組み合わせることができれば、
新しいAIをプレイヤーが作成することができるようになります。



夢のニューラルネットデータバンク構想

今後の展開

ニューラルネットのメリットをいかに活用するかが鍵になると思います。

今後、プロシージャルを活用したゲームが登場した場合、
膨大な世界環境データに対応したAIが必要となります。

このAIを人間が作成するか、ニューラルネットで学習して作るか。
それはコストで決まるでしょう。

例えば、1万コースあるレースゲームを作った場合、
そのコースの最適なAI用パスを人間が作成するのはコスト的に不可能です。
しかし、ニューラルネットにコースを学習させて、
開発パソコンのハード性能や物量をアップして対応できるのであれば、
現実的にニューラルネットを採用することになるでしょう。

人間のコスト　＞　ニューラルネットのコスト

この関係が成立するように、
ニューラルネットのメリットを活かす方法を考えていければと思います。

AI研究開発における企画的挑戦の「夢」と「現実」と「リスク」

AI研究開発における企画的挑戦の夢と現実

[AI研究開発における企画的挑戦]
・ニューラルネットワークを使った生物らしさをもったAIを作る

[夢]
・なんでも学習できる動物のようなAI
・生きている実感が得られるようなAI

[現実]
・発展途上の技術である。
・学習させたいXの本質を制作者側がよく知っていなければならない。
・実装と調整にコストがかかる。
・他のアルゴリズムで代用できる場合が多い。

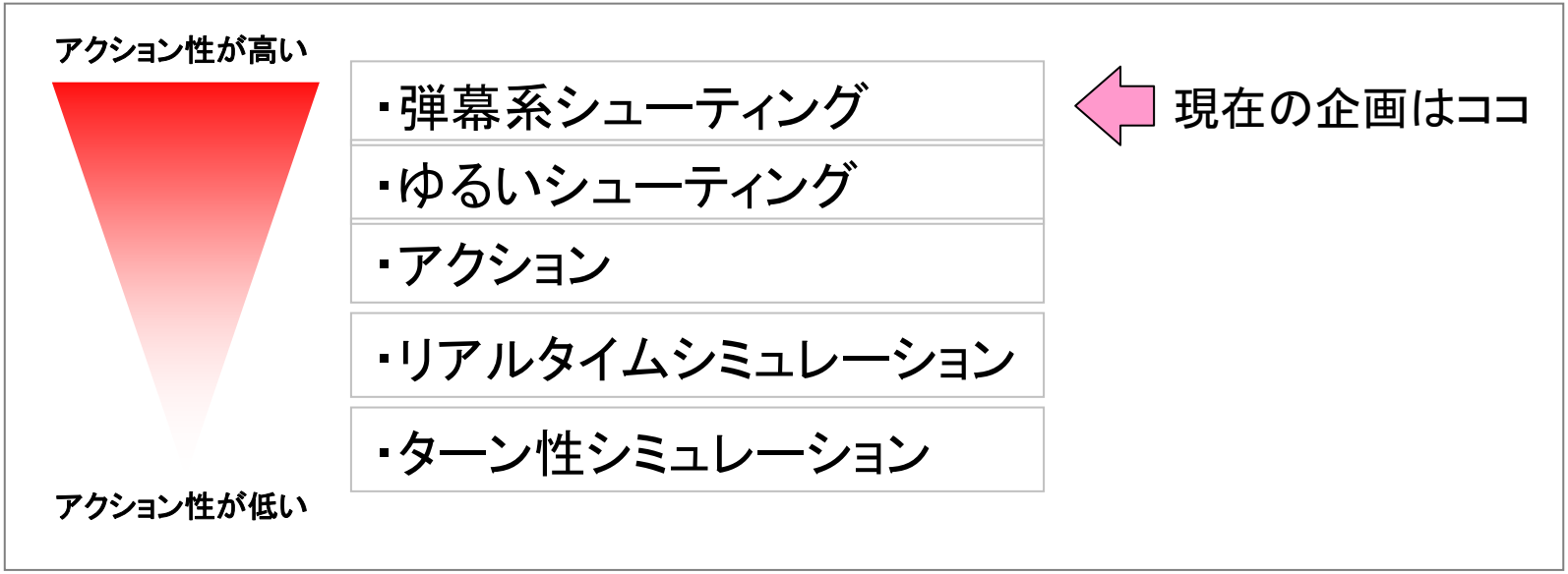
反省点

[問題]

今回のキャラ化した企画には、アクション性が強すぎた。

[解決案]

下記ゲームジャンルの上からテスト実装を行い、
もっともプレイヤーが楽しいと思われるアクション性のジャンルを探します。



反省点

[問題]　いま、あらためてこのゲーム画面をみると、「カオス」すぎる。

[解決案]　ゆっくり寝て休んで、考え直します。

次世代に求められるAIとは・・・

今は、まだニューラルネットはゲームの実装には、早すぎる技術かもしれません。

しかし、今後、ゲームの世界環境データが増大していけば、
現在のAIの作り方では、行き詰まる可能性が高いです。

オンラインゲームや、UCCコンテンツは、
その回避策として登場したのかもしれません。

しかし、そのオンラインゲームでさえも、
膨大な世界環境データを構築しなければならず、
「人」では補えない「敵」や「パートナー機能」がAIに求められていくでしょう。

機が熟したとき、ニューラルネットはその解決手段の一つとなるかもしれません。

ご静聴
ありがとうございました。

質疑応答
どうぞ

参考資料

[IGDAからダウンロードできる資料]
http://www.igda.jp/modules/mydownloads/

・第5回「ニューラルネットによるエージェント」 事前資料-ニューラルネット編より-
・ゲームAI連続セミナー「ゲームAIを読み解く」第5回補足資料

[参考書籍]

・「マッチ箱の脳」
　使える人工知能の話
森川　幸人著
新紀元社

・パーセプトロン
M.ミンスキー/S.パパート　　訳　中野 馨/阪口 豊
パーソナルメディア

・Cで作るニューラルネットワーク
平井　廣美著
パーソナルメディア

・学習とニューラルネットワーク
熊沢　逸夫著
森北出版株式会社

・進化しすぎた脳
池谷　祐二著
BLUEBACKS

・アンドロイドサイエンス
~人間を知るためのロボット研究~
石黒　浩著
毎日コミュニケーションズ

提供

Planning

オープランニング
プランナー　大野　功二
E-mail : skyfox@edit.ne.jp