

テレビゲームへの人工知能技術の利用

Use of Artificial Intelligence for Video Game

森川 幸人*

Yukihito Morikawa

* 有限会社ムームー
Muumuu Co., Ltd., Tokyo 160-0004, Japan.

1998年12月14日 受理

1. はじめに

近年ゲームは、ゲームマシンのスペックの向上に伴い、グラフィックやサウンド、それに、キャラクターの生態や振る舞いにもリアリティーが求められるようになってきました。

一方ゲームの定義はどんどん拡大され、仮想生命を飼うようなものもゲームの1ジャンルとして許容されるようになってきました。

そのような背景から、AI技術（人工知能）を利用した仮想ペットをモニターの中で育てるゲームや仮想生物が自律的に進化するゲームの開発を試みました。

2. ビデオゲームにおけるAI技術

パソコンゲームでは、本格的なAI技術を使ったゲームがいくつか存在します（図1）^{*1}。しかし、コンシューマ・ゲームマシン^{*2}では、これまで本格的なAI技術を利用した事例はないと考えられます。

これは、コンシューマ・ゲームマシンのCPUやメモリでは、そのような「コストの高い」エンジンを走らせることが不可能であったためです。

こうした事情は、いわゆる次世代マシン^{*3}の登場によって変わってきました。一つは、マシンのスペックが飛躍的に向上して、ぎりぎりですが、本格的なAI技術

を組み込むことが可能になったこと。もう一つは、ゲーム内容の多様化とリアリズム嗜好のブームの中、本格的なAI技術に対する期待が、制作者やプレイヤーに生まれてきたことがあげられます。

3. 実用例

ここでは、過去にAI技術を利用して作ったゲーム2作について、具体的な使用例を説明します。

「がんばれ森川君2号」^{*4}

このゲームは、いわゆる「育てゲー」とよばれるカテゴリーに属するゲームで、テレビの中で仮想のペット（このゲームでは、PiT(Pet in TVの略)と呼ばれるロボット型のペット）を飼うというゲームです（図2）^{*5}。ゲーム全体の流れは（図3）のようになります。

このゲームでは、ステージ中に隠されたAI Chipと呼ばれるアイテムを探すことが目的となります。このゲームは通常のゲームと違って、プレイヤーは直接キャラクターを操作することができません。PiTが自律的にステージを移動し、アイテムに対して「正しい」アクションを実行し、目的のアイテムを探し出さなくてはなりません。

ただし、プレイヤーは、PiTに対してアイテムに対する「正しい」アクションを教育することと、調べさせたい、向かわせたいアイテムを指示することができます。

これに対して、PiTはプレイヤーの指示したアクションを学習し、プレイヤーが指示する毎に、そのアイテ

*1 口絵iページ参照。

*2 コンシューマ・ゲームマシン
consumer game machine. 「家庭用ゲーム機」という意味の言葉として使われている。

*3 次世代マシン
プレイステーション、任天堂64、セガ・サターンを称して、そう呼ぶ。

*4 「がんばれ森川君2号」

プレイステーション用ソフト、発売元：SCE 97年5月発売

*5 口絵iページ参照。

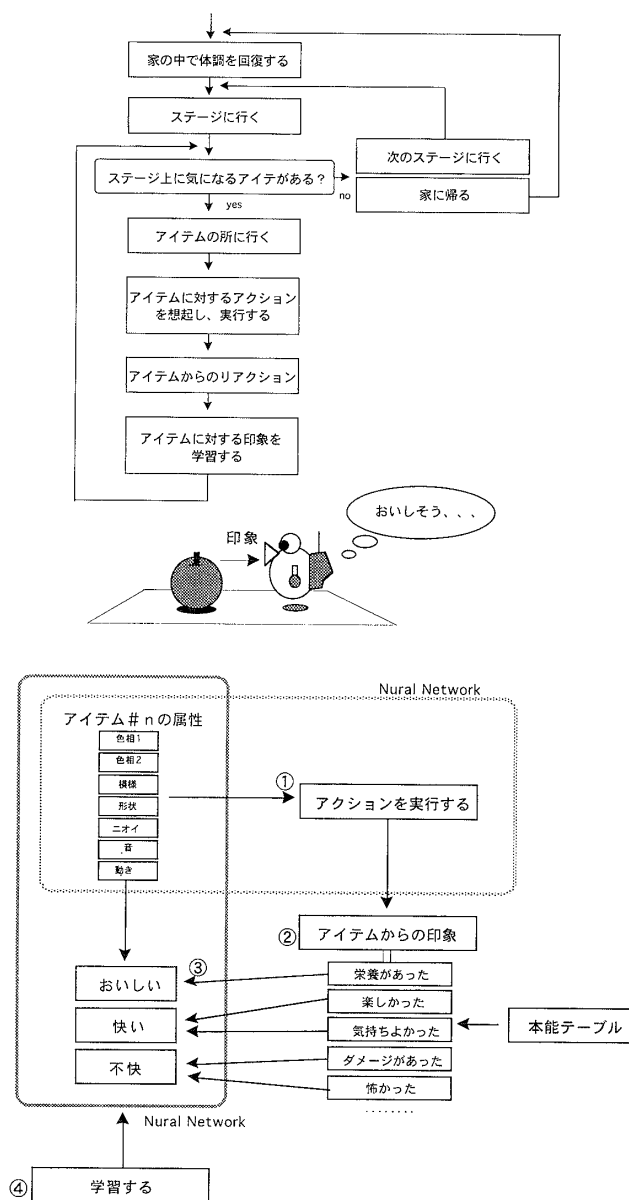


図 3

ムの重要度（指示される回数が多いアイテムは「重要なアイテム」だと認識する）を学習していきます。

また、アイテムに対してアクションした際、その結果から報酬（負の場合もある）を受け取ると、そのアイテムに対する快か不快かの「印象」を学習します（図3）。

こうした学習や判断用のエンジンとして、AIを使っています。

このゲームの場合、初めて遭遇するアイテムに対して、今までの経験からアクションや印象の推測ができることが必要であったこと、プレイヤーが教えるという立場で参加できる必要があったことから、AIモデルとして教師ありのニューラル・ネットワーク・モデルであるバック・プロパゲーション・モデル（以後BP）

を利用しました。

BP を使おうと決めた理由は、入出力のパターンの作成が直感的に理解しやすいこと、このアルゴリズムに関する文献が多いことです（このことは、AI を利用しようとする一般人には、とても重要な問題となります）。

また、ゲーム進行中に学習の追加や修正が行われない本能的な判断にもBPを利用しました。この「本能」としては、出荷前に十分に訓練されたBPが組み込まれています。

本能的な判断には、以下の2つがあります。

1. ステージでの方針
2. 音楽の好み

1. のステージの方針とは、ステージにいるときに、自分の状態（エネルギー残量や疲労度など）と、ステージ中のアイテムの印象（重要度や快度など）から、なにをすべきかを判断することを意味します。

アクション選択のルールは、あらかじめこちらで設定し、BPに十分な訓練を行いました（図4）。

2. の音楽の選択ですが、これは極めてゲーム的な要素です。通常のゲームでは、自動的にBGMが選択されプレイされますが、このゲームではPiTがプレイする、しないの判断と曲の選択を行います。

ステージ中には、音楽（1曲）のデータが入ったCD-ROMが20枚以上落ちています。PiTはステージ中で手に入れた曲をプレイすることが可能なのですが、複数の曲を有している場合は、PiTの好み（そのときの感情やエネルギー残量、疲労度や音楽の種類に対する本能的嗜好性など）を入力としているので、プレイする曲を選択する（場合によってはBGMをかけない）ようになっています。

具体的には、BGMのタイプをあらかじめ5種類に分類しておき、PiTの体調、そのときの気分、またPiT個体の個性に応じて、どのタイプを選択するかを学習させました。

あらかじめ訓練済みのBP（このゲームでは「本能」にあたる部分）を使う場合は、BPはゲーム内でも非常に具合よく機能します。通常のゲームの場合は「エキスパート・ルール」のように、たくさんのルール群を用意しますが、ゲーム制作者は必ずしも数学に強くないため*6、往々にして非線形な問題に対して帳尻のあったルールを設定できないことが起こります。

これに対して、BPのサンプルの作成は、直感的にわ

*6 数学に強くない

例えば、 Σ （シグマ）の意味がわからない、くらいの数学のレベルである。

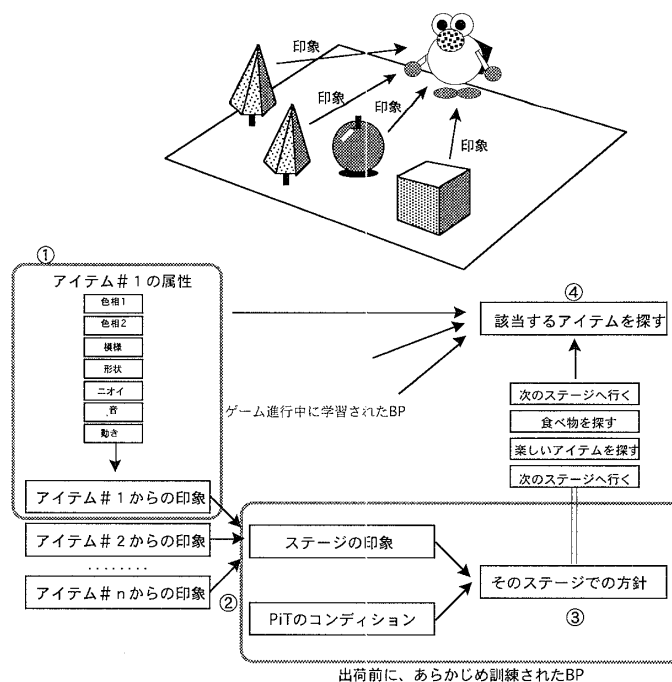


図 4

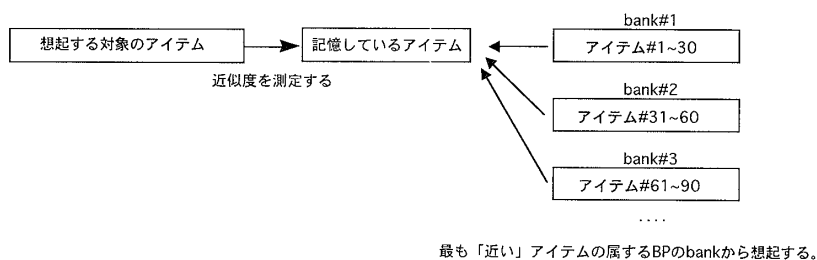


図 5

かりやすく、非線形問題であるかどうかなどを考慮しなくてすむという利点があります。

ただし、ゲームの世界では（そして、おそらく実世界でもそうだと思うのですが）、多くの事象に関して、万人に共通な「唯一の正解」などは存在しません。したがって、唯一絶対の教師信号もなく、あらかじめ学習を済ませておく、ということができません。

このため、ゲーム中でたびたび学習の追加や修正を行わなければならなくなります。これには以下の問題点があります。

1. 再学習のたびに、ゲームが中断する。
2. 再学習のたびに、(少なくとも一時的には) 学習精度が落ちる。
3. 再学習のたびに、(サンプル数が増えるため) 学習時間が長くなり、メモリーも必要になってくる。

これらの問題は、調べた限りでは、実用的な解決策がありませんでした。

そこで、このゲームでは脳にパーテーションを切るという仕組みを組み込みました。あらかじめ、1つのBPは30のサンプルを上限とするように設定します。31番目からのサンプルが発生すると、新たな「脳のパーツ」が生成され、そこで学習されます。

新規の入力パターンの推論の際は、直接BPに入力する前に、すでに学習済みのサンプルとの「近さ」を測定し、最も近似のサンプルについて学習された「脳のパーツ」からの出力を利用します（図5）。

これによって、最も（ゲームとしては）問題な学習時間とメモリーの肥大化という問題を回避しました。それとひきかえに、この方法では、学習の精度は、広くサンプルを学習できないことから、最終的には悪くなることが予測されます。

これを回避するために、ステージ中で出会うだろうアイテムの大まかな順序を想定し、なるべく幅広いジャンルのアイテムに均等に遭遇するように配慮しました。

(この方策は、このゲームではうまくいきましたが、全く一般性のない方策ではあります)。

「アストロノカ」*7

このゲームは、宇宙で農家になるゲームです(図6)*8。

プレイヤーは、与えられた畑で「宇宙作物」を栽培します。「宇宙作物」を収穫した際には、その作物の種も得られますが、この種同士を交配させると新種の作物の種が合成されます。収穫した「宇宙作物」は、そのまま売ってお金に換えることもできますが、新種の作物はコンクール(新種作物の品評会のようなもの)に出すこともできます。

一方、プレイヤーが育てた作物を狙う害獣「バブー」が存在します。このバブーから作物を守るために、畑の前の空き地にトラップ(落とし穴やかかしや扇風機など)を仕掛けます。これを「トラップバトル」と呼んでいます。バブーは最初のうち、簡単なトラップにもひっかかりますが、やがて耐性を持ったり、トラップのかわしかたを学習したりして、ひっかからなくなる進化をみせます。このため、プレイヤーはトラップの配置や組み合わせを替えたり、新種のトラップを購入したりしてバブーを撃退します。

作物の交配と栽培、それとバブーとの攻防がこのゲームの主な遊びです。

このゲームでは、バブーの進化に遺伝的アルゴリズム(以下GA)を利用しました。

GAを使った理由は、バブーの進化を、(通常のゲームのように)あらかじめ仕込んでおくのではなく、プレイヤーの仕掛けるトラップの種類によって、どんどん変わるようにしたかったためです。また、バブーも我々同様遺伝子を持っていて、淘汰と交配、突然変異によって進化していくという、生き物としての「実感」を表現したかったためです。

トラップバトルでの成績を適応度関数として、トラップに対する耐性や回避方法を獲得するように、進化していくような仕組みになっています。

次にGAの仕組みに従って、各要素の詳細を説明します(図7)。

バブーの集団は20個体あり、ゲーム中では3体までしか畑にやってきましたが、バックグラウンドでは、全個体がトラップバトルを行っています。

バブーは、身体的特性(体重や身長や腕力、脚力など)、トラップの特性に対する耐性(電気、熱などに

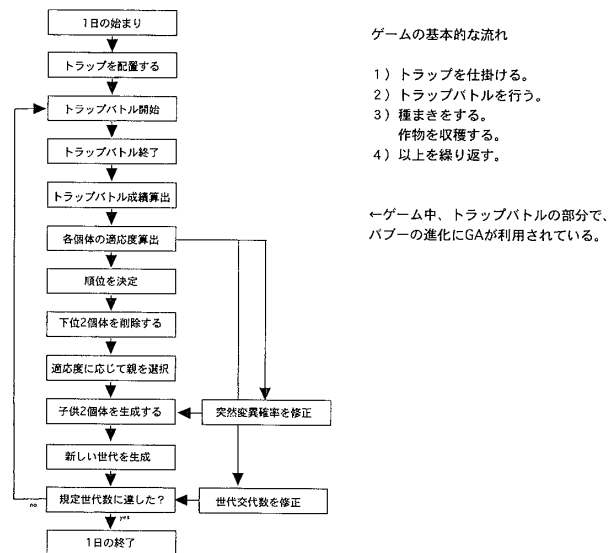


図7

【バブーの属性】総計56

体重	身長	腕力	脚力	耐性_かかし	耐性_快光線
1.87	6.85	16.25	25.03	25.03	16.25
6.85	1.87	25.03	16.25	6.85	25.03

【各ビットの重み】

0	1	2	3	4	5	6	7
1.87	6.85	16.25	25.03	25.03	16.25	6.85	1.87

図8

対する耐性)、障害物に対する回避アクション(迂回する、押す、叩くなど)の情報を遺伝子上に持っています。

各要素の遺伝子の長さは8ビットで、要素の数は56要素ありますから、バブー1体につき448ビット長の遺伝子を持っていることになります。

各要素のビットには、図8のような数値が定義されていて、全てのビットが立ったとき、その要素の値が100になるようになっています。特徴として、獲得した属性が容易に壊れないように、2カ所のビットが同じ数値を持つようになっています。

トラップバトルの成績には、以下のような適応度関数を用いました。適応度=成績+TB時間*0.3+エンジョイ*0.5+トラップ点+安全点+HP*0.5

全ての項の説明は省きますが、例えば、成績とは、トラップバトル終了時のバブーのいる位置に対するボーナスで、ゴールの位置が最も高く、その周囲がその次に高いボーナスが与えられます(図9)。また、TB時間とは、トラップバトルに要した時間、安全点とはトラップに対する耐性の余裕の度合いなどを意味しています。

*7 「アストロノカ」

プレイステーション用ソフト、発売元:エニックス 98年8月発売

*8 口絵iページ参照。

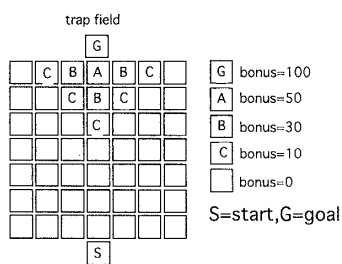


図 9

親選択は、ルーレット方式が取られています。突然変異確率は3%を標準としていますが、親の遺伝子の近さ（ハミング距離で測定）に応じて、突然変異確率が変動します。親が近いほど確率は大きくなり、遠いほど確率は小さくなります。

また、突然変異が起こる箇所は、完全にランダムではなく、そのときの環境（配置されたトラップの種類）によって、おおまかに限定されています。

ゲーム中では、ゲーム中の1日で1世代の世代交代と説明していますが、実際には、バックグラウンドで1日で5世代の世代交代を行っています。これは、GAが必要とする世代数が、ゲーム的には大きすぎるためです。また、世代交代数は、進化の速度（適応度の平均点の伸び）を重みとして調整しています。つまり、進化の速度が遅くなったときは、バックグラウンドでの世代交代数は大きくなり、進化が早くなると世代交代数は小さくなります。これは、バブーの進化の速度が著しく変わるのゲーム的なバランスとして、望ましくないためです。

このゲームの場合、トラップの配置や種類が何度も変更されます。これはGA的には環境がつねに変動することを意味しますから、GAにとっては、とても不都合な条件です。

特に以下の2つの事態が問題となります。

1. 学習が収束しにくい。
2. 過去に獲得した記憶（トラップの攻略方法などが、役に立たない。

1つめの問題は、進化の速度を重みとして世代交代数を逐次調整していくことである程度解決することができました。

2つめの問題は、突然変異を起こす場所を、配置されたトラップの種類によって、大ざっぱに限定することによって解決しました。これにより、直面するトラップに関係しない遺伝子群は、突然変異の影響を強く受けることがなく、過去に獲得した記憶を保存することができます。

4. ま と め

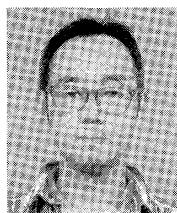
ゲームマシンとその周辺の機器（センサーや入力装置）の進化は、さらに「リアル」な表現を可能にしていくことでしょう。リアリティーの追求は、今のところ、主にビジュアル面（3次元化やフルカラー、フルモーション化など）に集中していますが、今後は、それにとどまらず、ゲーム中のキャラクター達の学習や判断の仕組みにも、それが求められていくと考えられます。

そして、「リアル」な学習や推測、判断、それに進化など「生き物らしさ」を表現するエンジンとしては、本格的なAI技術も有効な技術となるのではないかと考えています。

また、ゲーム内の働きに限らず、特にGAなどは、ゲーム開発の技術としても期待できます。例えば、ロールプレイングゲームなどの場合、配置するモンスターの強さ（パラメータの値）のチューニングは、現在手作業で調整されています。パラメータの数は、昔に比べて非常に多くなってきていますから、この作業はとても負荷の高い作業となっています（通常この作業だけで1~2カ月の時間を要します）。

しかし、GAを使えば、適切な適応度関数さえ設定できれば、とても短時間でバランスの良いパラメータの値を探し出すなどができると思います。

著 者 紹 介



森川 幸人

1959年岐阜生まれ。1982年筑波大学芸術専門学群卒業。1990年有限会社ウルトラ設立 同取締役社長。1995年有限会社ムーム設立 同取締役社長。1990~95年 テレビ用CG制作。代表作「アインシュタイン」、「ウゴウゴルーガ」、1995年プレイステーションソフト「ジャンピングフラッシュ1」プロデュース、1996年プレイステーションソフト「ジャンピングフラッシュ2」プロデュース、1997年プレイステーションソフト「がんばれ森川君2号」制作、1998年プレイステーションソフト「アストロノカ」制作。

<morikawa@muumu.com>