RBFN と強化学習的手法による評価関数を用いたゲーム AI

佐藤　力

富山県立大学
u220029@st.pu-toyama.ac.jp

January 21, 2025

はじめに

近年、技術の発展とともにビデオゲームは大きく発展している。
しかしゲームを楽しむうえで深くゲームを楽しまずライトユーザーで終わってしまう人が多い。.

本研究の目的

ライトユーザがゲームを楽しみながらコアユーザへ成長するための仕組みを提供することを目的として、評価関数を自動的に学習するゲーム AI の設計方法を提案しています。特に、SRPG を対象に、機械学習を活用した評価関数の生成方法を研究しています。.

関連研究

TD-GAMMON

特徴として、評価関数に三層のシグモイド型ニューラルネットワークを利用し、その学習に強化学習の手法の一つである TD(λ) が使われている。学習は自分自身を対戦相手として行われ、それを繰り返すことで人間のトッププレイヤーとも渡り合えるようになっている。

RBF ネットワーク

ニューラルネットワークの一種で中間層に RBF を用いて出力層は各 RBF ユニットの重み付き和になる。主にガウス関数がよく使われている。RBF ネットワークは中間層のユニット数が十分に多ければ次のような形で任意の連続関数を近似することが可能である。

$$\phi_i(\mathbf{x}) = \exp\left(-\frac{|\mathbf{c}_i - \mathbf{x}|^2}{\sigma_i^2}\right)$$

図 1: ガウス関数

$$f(\mathbf{x}) = \sum_{i=1}^{N} w_i \phi_i(\mathbf{x})$$

図 2: 連続関数

提案手法 1

評価関数の学習には機械学習の一つである TD(λ) と RBF ネットワークを組み合わせて行います。これはシグモイド型ニューラルネットワークに比べてＲＢＦネットワークのほうが二次利用することが容易であるためである。
RBF ネットワークを評価関数として利用し、ゲーム中の状態を数値化。
TD(λ) を用いて、評価関数のパラメータを学習。

提案手法 2

評価関数の定義

$$V_t(s_k) = \sum_i w_i \exp\left(-\frac{|\mathbf{c}_i - \mathbf{x}_k|^2}{\sigma_i^2}\right)$$

図 3: 評価関数

xk: 状態 Sk の特徴ベクトル（ゲームに関するヒューリスティックな値を含む）
ci: 各 RBF ユニットの中心点
σ i: 各ユニットの分散（特徴空間での影響範囲を制御）
wi : 各ユニットの重み
exp(− ‖ c i − xk ‖ 2/ σ i2): RBF（Radial Basis Function）のガウス関数。

提案手法3 パラメータの更新方法

重み wi の更新式

$$\Delta w_i = \alpha \delta_t \sum_{k=1}^{t} \left\{ \lambda^{t-k} \phi_i(\mathbf{x}_k) \right\}$$

図 4: 重みの更新式

- α：学習率（更新の大きさを決めるパラメータ）。
- δ_t：TD誤差（状態 s_t の評価値と報酬の差）。
- $\phi_i(x_k) = \exp\left(-\frac{\|c_i - x_k\|^2}{\sigma_i^2}\right)$：RBFユニットの出力。
- λ^{t-k}：過去のデータの重要性を指数的に減少させるパラメータ。

図 5: 重みの説明

提案手法3 パラメータの更新方法

中心点の更新式

$$\Delta c_{ij} = \alpha \delta_t \sum_{k=1}^{t} \left\{ \lambda^{t-k} \phi_i(\mathbf{x}_k) \cdot \frac{c_{ij} - x_{kj}}{\sigma_i^2} \right\}$$

図 6: 中心点の更新式

c_{ij}：RBFユニット ϕ_i の j 次元の中心。
x_{kj}：入力ベクトル x_k の j 次元の値。
更新方向は、中心 c_i を入力 x_k に近づけるように調整される。

図 7: 中心点の説明

提案手法3 パラメータの更新方法

分散の更新式

$$\Delta\sigma_i = \alpha\delta_t \sum_{k=1}^{t} \left\{ \lambda^{t-k} \phi_i(\mathbf{x}_k) \cdot \frac{|\mathbf{c}_i - \mathbf{x}_k|^2}{\sigma_i^3} \right\}$$

図 8: 分散の更新式

σ_i：RBFユニット ϕ_i の分散。

分散の更新は、入力 x_k が中心 c_i からどれだけ離れているかを考慮して行われる。

図 9: 分散の説明

TD 誤差の定義

定義

$$\delta_t = r_{t+1} + \gamma V_t(s_{t+1}) - V_t(s_t)$$

図 10: TD 誤差

r t+1: 次の状態で得られる報酬（勝利：+1、敗北：-1、その他：0）
γ : 割引率（未来の報酬をどれだけ考慮するか）。
Vt(st): 現在の状態の評価値

学習の流れ

1. ゲームプレイ中に各状態 sk の特徴ベクトル xk を取得
2. 評価関数 Vt(sk) を計算し、現在の状態の価値を予測
3. ゲーム結果（報酬 rt+1) を受けて TD 誤差を計算
4. 上記の更新式を用いて、重み、中心点、分散を更新
5. これを繰り返し実行して評価関数を徐々に学習