

논문 2020-15-30

인공신경망과 유전 알고리즘을 이용한 팩맨 게임 강화학습

(Pacman Game Reinforcement Learning Using Artificial Neural-network and Genetic Algorithm)

박진수, 이호정, 황두연, 조수선*

(Jin-Soo Park, Ho-Jeong Lee, Doo-Yeon Hwang, Soosun Cho)

Abstract : Genetic algorithms find the optimal solution by mimicking the evolution of natural organisms. In this study, the genetic algorithm was used to enable Pac-Man's reinforcement learning, and a simulator to observe the evolutionary process was implemented. The purpose of this paper is to reinforce the learning of the Pacman AI of the simulator, and utilize genetic algorithm and artificial neural network as the method. In particular, by building a low-power artificial neural network and applying it to a genetic algorithm, it was intended to increase the possibility of implementation in a low-power embedded system.

Keywords : Reinforcement learning, Artificial neural network, Genetic algorithm, Pacman game, Simulator

I. 서론

1. 연구 배경

강화학습은 시행착오를 통해 수집되는 수많은 데이터 속에 숨어 있는 패턴을 학습을 통해 찾아내는 방법이다. 또한 강화학습은 주어진 환경 안에서 현재 상황을 인식하여, 실행 가능한 행동 중 최대의 보상을 얻을 수 있는 행동을 선택하는 기계 학습 분야 중 하나이다 [1]. 강화학습이 이용되는 분야로서는 게임, 금융, 자율주행자동차 등이 있으며, 그 외에도 다양한 분야에서 활용되고 있다 [2].

특히 게임 분야에서는 강화학습을 이용하여 게임의 특정목적을 달성하였을 때마다 보상을 주면서 캐릭터가 학습을 하게 만든다 [3]. 오랫동안 게임

캐릭터의 제어를 위해 다양한 방법의 강화학습이 연구되어 왔다 [3-5]. 강화학습 분야에는 상태공간의 문제, 캐릭터의 자동화 등의 문제가 있는데, 캐릭터의 자동화를 다룬 연구들은 서양보드 게임인 오เทล로, 틱택토 외에 슈팅게임 등을 다루고 있다. 이렇듯 게임 분야에서 활발하게 쓰이는 캐릭터의 자동화를 위한 강화학습을 본 논문에서는 고전 게임 팩맨에 적용하고자 한다.

2. 유전 알고리즘과 인공신경망

유전 알고리즘이란 자연세계의 생물 진화를 모방한 알고리즘이다. 교배 연산 (crossover)과 변이 연산 (mutation)을 거쳐 최적해를 찾아내는 기법으로, 이 연산들은 유전자 (gene)로 이루어진 한 세대 (generation) 안에서 이루어진다 [6]. 특히 변이 연산 기법은 지역 최적해에 빠지는 것을 막을 수 있는데, 다른 최적화 알고리즘과의 차이이자 큰 장점으로 작용할 수 있다. 또한 유전 알고리즘은 탐색공간이 매우 크거나 일반적인 해를 찾기 힘든 문제에서 뛰어난 성능을 발휘하며, 반드시 최적해가 필요한 경우가 아닐 때 유용하다 [7]. 본 연구에서는 이 유전자가 각각 팩맨 AI가 되어 게임을 반복하며 진행한다. 각 유전자가 마친 게임의 결과를 토대로 교

*Corresponding Author (sscho@ut.ac.kr)

Received: Jul. 28, 2020, Revised: Aug. 14, 2020, Accepted: Sep. 8, 2020.

Jin-Soo Park: Korea National University of Transportation (B.S. Student)

Ho-Jeong Lee: KNUT (B.S. Student)

Doo-Yeon Hwang: KNUT (B.S. Student)

Soosun Cho: KNUT (Prof.)

배 연산과 변이 연산을 거치면 더 좋은 게임 결과를 이룰 수 있는 새로운 세대가 만들어지는 것이다.

인공신경망은 생물학적 신경망을 모방한 기계 학습 알고리즘이다. 신경망과 같이 구성되어 시냅스의 결합으로 네트워크를 형성한 인공 뉴런이 학습을 통해 문제 해결 능력을 가진다. 본 연구에서 인공신경망이 유전 알고리즘의 유전자가 된다. 즉 팩맨 AI의 판단을 내려주는 근거가 되는 것인데, 현재 상황을 인식하여 어떠한 판단을 내리는 알고리즘으로서 적합하다. 특히 본 연구에서는 인공신경망을 구축하여 유전 알고리즘에 적용함으로써 임베디드 시스템에서의 구현 가능성을 높이하고자 하였다. 이러한 유전 알고리즘과 인공신경망은 구조의 결합을 시도한 이유는 둘 다 자연계 생물의 진화와 학습 원리를 모방한 구조이고 이를 사용하면 팩맨이 살아있는 유기체처럼 학습을 할 것으로 기대하였기 때문이다.

본 연구의 목표는 팩맨 게임 내에서 강화학습이 가능한 팩맨 AI를 개발하여 사람을 능가하는 플레이를 가능토록하고 그 차이를 사용자가 실감하게 하는 것이다. 팩맨 게임의 룰은 유명을 피하여 오래 생존하고 포인트를 먹어나가는 것이다. 이러한 게임 규칙에 따라 강화학습을 진행할 환경과 인식해야 할 대상이 명확하여 간단한 인공지능을 설계해보기에 적합하므로 팩맨 게임을 선택하였다. 또한 인공신경망의 강화학습엔 일반적인 방식인 역전파 (Back propagation) 알고리즘이 있는데 본 연구에서는 인공신경망에 역전파 알고리즘을 적용하지 않은 단순한 다층 퍼셉트론 구조를 사용하여 유전 알고리즘과 결합하였을 때, 역전파 알고리즘만큼의 효율이 나오는지 알아보고 향후 연구를 계획할 수 있는 근거를 만들고자 하였다.

II. 관련 연구

본 논문의 목적은 게임 내 AI를 강화학습시키는 것이고, 그 방법으로 유전 알고리즘과 인공신경망을 활용한다. 게임 분야에서 강화학습이 자주 쓰이는 만큼 참고할 수 있는 관련 연구 또한 아주 많다. 먼저, 절차적 콘텐츠의 생성을 유전 알고리즘을 이용해 연구한 논문 [8]이 있다. 게임에서 생성되는 몬스터의 주요 속성을 유전 알고리즘의 유전자로 구성하고, 사용자와 몬스터의 전투 시뮬레이션으로 유전자를 평가한다. 그 평가를 토대로 유전자들 교배하여 플레이어 적응형 몬스터를 생성한다. 비슷한

주제로 게임 몬스터 생성에 적합한 유전 알고리즘을 연구한 논문 [9]이 있는데, 기존 유전 알고리즘과 달리 상동염색체 구조를 갖도록 하였다. 각 개체가 하나의 유전자를 갖는 것이 아닌 한 쌍의 대립 유전자를 지니게 하였고, 실험 결과 이러한 구조가 성능에 더 좋은 영향을 미치는 것으로 밝혀졌다.

앞선 두 논문과 같이 게임에서 플레이어에게 맞는 몬스터를 생성한다는 것은 게임의 난이도를 조절한다는 의미와 같다. 이러한 게임 난이도 조절에 대해 유전 알고리즘을 이용해 연구한 논문 [10]이 있다. 테트리스 게임에 유전 알고리즘을 적용하여 게임 플레이 중 사용자의 성향과 게임 적응도를 분석하고 그에 맞추어 적당한 난이도를 제공한다.

또한 함정 교전 시뮬레이션을 연구한 논문 [11]에서는 함정 교전에서 보다 적합한 대응을 찾기 위해 환경변화에 능동적으로 대처할 수 있도록 강화 학습 기능을 가지며, 또한 유전 알고리즘을 이용하여 세대별 진화학습 기능을 갖는 에이전트 모델링 방법론을 제안하였다.

인공신경망과 유전 알고리즘을 적용한 파라미터 설계 방법론을 연구한 논문 [12]에서는 기계학습 기반의 파라미터 설계 절차를 3단계로 구성하는데, 첫 번째로 인공신경망 구축에 필요한 데이터를 수집하고, 두 번째로 수집한 데이터를 기반으로 인공신경망을 구축한다. 마지막으로 최적의 공정조건을 도출하기 위해 유전 알고리즘을 적용한다. 이때 일반적인 문제에서는 적합도 함수로서 평균 및 산포의 반응함수를 사용하지만, 연구 [12]에서는 반응함수를 대신해 인공신경망을 사용한다. 유전 알고리즘과 인공신경망의 적절한 조합 예시라 할 수 있다.

본 논문에서는 팩맨 게임의 강화학습을 위하여 유전 알고리즘 뿐 아니라 다층 퍼셉트론 구조의 인공신경망 또한 활용한다. 미로와 같은 팩맨 게임의 맵 구조상, 주어진 환경을 인식하고 판단을 내릴 알고리즘이 필요하기 때문이다. 이와 비슷한 연구로 게임 환경에서 유전 알고리즘과 인공신경망을 이용한 경로탐색에 대한 연구 [13]이 있다. 이 연구에서는 유전 알고리즘과 인공신경망 두 가지 알고리즘을 각각 사용하여 실험했는데, 유전 알고리즘보다 인공신경망을 활용한 경우가 경로탐색 문제해결에 더 적합한 결과를 보여주었다. 본 논문에서 유전 알고리즘과 인공신경망을 결합한 연구를 진행하는데 참고하였다.

RPG 게임 환경에서 유전 알고리즘과 인공신경망을 결합하여 학습 및 적응 능력을 부여하는 연구 [14]에서는 유전 알고리즘에서 사용할 유전자로 인

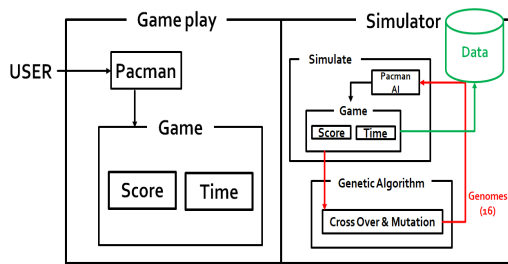


그림 1. 시스템 구성도
Fig. 1 System configuration

공신경망 노드의 각 가중치를 사용하였는데, 선택압의 설정과 비균등 변이에 집중하여 100개체씩 1000세대 동안 AI를 학습시켰다. 선택압은 우수한 해들과 열등한 해들 사이의 적합도 차이를 말하며, 비균등 변이는 세대가 진화할수록 변이율을 줄여나가는 것을 말한다.

본 논문에서는 연구 [14]와 달리 선택압의 설정 등에 집중하지 않고 유전 알고리즘과 인공신경망을 단순하게 결합하였다. 자연계에서 모방한 알고리즘들의 단순한 결합만으로 강화학습의 수준이 어느 정도 되는가를 확인하고 필요하면 향후 연구에서 역전파 알고리즘을 이용한 보완을 계획하기 위해서이다.

III. 본 론

1. 시스템 구성

시스템의 구성은 그림 1과 같이 게임 플레이어와 시뮬레이터로 나뉜다.

게임 플레이 기능에서는 직접 팩맨 게임을 해볼 수 있으며, 시뮬레이터 기능에서는 팩맨 AI가 유전 알고리즘으로 강화학습이 진행되는 과정을 관찰할 수 있다. 시뮬레이터뿐만 아니라 직접 게임을 플레이해봄으로써 진화되는 AI를 관찰할 때 그 차이를 실감할 수 있다.

시뮬레이터 기능에서는 한 세대의 유전자들이 각각 순서대로 팩맨이 되어 게임을 진행한다. 한 세대의 개체군 (population)은 16으로 설정하여, 16개의 각기 다른 유전자들이 게임을 모두 마치면 그 결과값을 토대로 유전 알고리즘을 통해 진화시킨다. 진화되어 도출된 다음 세대 또한 이 과정을 반복하여 사용자가 종료할 때까지 진화가 계속된다. 각 유전자가 게임을 마칠 때 마다 그 값의 분석을 위해 데이터로 저장된다.

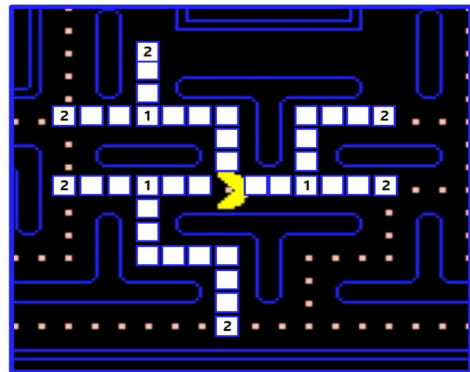


그림 2. 팩맨의 인식 범위
Fig. 2 Recognition range in Pacman

강화학습은 CPU i7-3537, RAM 8GB, 윈도우 10 64비트 환경에서 진행되었으며, 3시간의 학습이 진행되었다. 팩맨 게임은 오픈소스를 활용해 pygame으로 구성하였다. 개발 언어로는 파이썬을 사용하였다. 본 논문에서는 연구의 실질적인 내용인 시뮬레이터 기능을 주로 서술한다.

2. 강화학습을 위한 환경인식

팩맨 게임의 룰은 유령을 피하며 포인트를 먹는 것이다. 포인트를 먹는 것보다도 중요한 것이 생존인데, 유령에게 닿는 순간 게임 오버되기 때문이다. 따라서 팩맨이 강화학습을 위해 현재 환경에서 인식해야 할 개체는 유령이다.

그림 2는 팩맨이 유령을 인식해야 하는 범위를 보여준다. 시뮬레이터 기능에서 팩맨 AI는 멀리 있는 유령까지 인식할 필요는 없다. 팩맨은 항상 자신을 기점으로 일정 범위까지의 길을 인식하는데, 벽으로 막혀져있는 방향이 아닌 진행 가능한 방향으로 인식 센서가 뻗어나간다. 이때 커브길을 만나면 방향을 꺾어나가며 탐색을 하고 교차로 (진행 가능 방향이 3갈래 이상)를 만나면 탐색을 종료한다. 그리고 각각 탐색이 종료된 첫 번째 교차로를 기점으로 다시 한 번 교차로를 만날 때까지 뻗어나가며 탐색한다. 이렇게 구현한 유령 인식 함수는 게임이 진행되는 동안 항상 작동하는데, 유령이 범위 내에서 인식되면 그 위치를 4크기의 배열로 출력한다. 배열의 원소는 순서대로 오른쪽, 왼쪽, 위쪽, 아래쪽 방향을 의미한다. 예를 들어 유령이 오른쪽 방향에서 뻗어 나온 센서에 인식되었을 때, 유령 인식 함수는 [1,0,0,0]을 출력한다.

센서와 맵의 구조상 두 번째 교차로까지 탐색하는 것이 가장 이상적이다. 세 번째 교차로까지 탐색한다면 넓게 인식할 수는 있으나, 길을 따라 뻗어나온 센서끼리 겹칠 수 있다. 만약 겹치는 부분에서 유령이 인식되었을 때, 두 방향 이상에서 인식된 것처럼 결과를 출력할 수 있고 팩맨의 활동범위를 제약할 수 있기 때문에 두 번째 교차로까지 탐색한다.

3. 인공신경망의 적용

팩맨 AI의 강화학습을 위해 유전 알고리즘과 인공신경망을 사용하여 구현한다. 한 세대의 개체군은 16이며, 각각의 유전자는 인공신경망으로 이뤄진다. 실제 유전자에 적용한 인공신경망의 구조는 다층 퍼셉트론 구조를 사용했다.

인공신경망의 심층학습은 학습 (training)과 추론 (inference)이라는 두 가지 단계로 구성되는데 먼저 인공신경망이 주어진 상태에서, 학습용 데이터를 활용해 인공신경망을 학습시키는 것이 학습단계이다. 학습된 인공신경망 모델을 활용해 새로운 데이터를 입력하여 출력을 얻는 것이 추론단계이다. 인공신경망을 폭넓게 활용하기 위해서는 막대한 계산량에 필수적으로 뒤따르는 많은 전력소비 문제를 해결해야 하는데 이를 위해 인공신경망과 알고리즘을 최적화함으로써 소프트웨어적인 문제 해결을 모색하고자 한다.

본 논문의 인공신경망은 입력층, 은닉층, 출력층의 3가지 층과 각 층을 연결하는 시냅스 역할의 가중치 (weights)를 가지고 있다. 계산량을 줄여 유전 알고리즘 알고리즘을 구현하기 위해 가장 간단한 형태의 인공신경망을 구축한다. 입력층과 출력층은 노드 4개로 한 층씩 두고, 은닉층은 노드 10개, 20개, 10개로 세 층을 둔다. 입력층과 출력층의 노드 개수가 4개인 것은 인식해야 할 것도 방향, 판단을 내려야 할 것도 방향이기 때문이다. 4크기의 배열로 반환된 유령의 위치가 강화학습을 위한 현재 환경 인식에 대한 값이 되며, 이를 기반으로 판단을 내리기 위해 인공신경망의 입력층에 삽입된다.

이렇게 삽입된 환경 인식 값이 인공신경망 안에서 가중치 망과 연산되며 다음 층의 노드 값을 만들어낸다. 최종적으로 출력층에 다다르며 이는 인식 값을 바탕으로 내린 판단이 된다. 인공신경망은 마지막으로 4크기의 배열을 반환하며, 이 배열이 유령 방향과 같은 원리로 팩맨의 이동 방향이 된다.

하지만 실질적으로 유전 알고리즘을 통해 진화가 될 부분, 즉 염색체 (chromosome)가 될 부분은 각 층의 노드들이 아닌 가중치이다 [7]. 각 가중치

는 행렬 데이터로 구성이 되는데, 각 연결된 층의 노드 개수를 행과 열로 가진다. 예를 들어 입력층과 은닉층 사이의 가중치 행렬은 4행 10열이 된다. 가중치의 행렬 데이터와 이전 층 노드의 1차원 배열 값을 행벡터로 취급하여 행렬 곱을 하면 다음 층의 노드 개수와 맞는 행벡터, 즉 1차원 배열 값을 연산할 수 있다. 이때 단순히 행렬 곱하여 연산할 값을 다음 층 노드의 값으로 주지 않고 활성화 함수 (activation function)를 한번 거친 후 넘겨준다. 주로 비선형 함수를 사용하는데, 여기선 ReLU 함수를 거친 후 값을 넘겨준다. ReLU 함수의 수식은 다음과 같다.

$$f(x) = \begin{cases} 0 & (x \leq 0) \\ x & (x \geq 1) \end{cases} \quad (1)$$

각 층의 노드 값들은 연결된 가중치와의 행렬 곱을 마치고 ReLU 함수를 이용해 0이하의 값들을 전부 0으로 치환하며 다음 노드의 값으로 연산된다. 이 과정이 은닉층을 통과할 때까지 반복되고 마지막 출력층의 노드 값을 연산할 때는 ReLU 함수가 아닌 softmax 함수를 사용한다. softmax 함수의 수식은 다음과 같다.

$$P = \frac{\exp(a_k)}{\sum_{i=1}^n \exp(a_i)} \quad (2)$$

softmax 함수가 최종 출력층의 노드 값 중 어떤 것이 가장 적합한 판단인지 확률 분포로 정규화해 준다. 따라서 출력층이 입력층과 같이 방향을 의미하는 것은 같으나, 각 방향을 의미하는 배열의 원소들이 가지는 값은 적합한 판단일 확률, 0에서 1사이의 값을 가진다.

각각 게임을 반복하는 팩맨 AI는 이러한 구성의 인공신경망을 유전자로 가지며, 출력층 노드의 값 중 가장 큰 값을 가진 원소의 방향을 움직일 방향으로 선택한다.

4. 유전 알고리즘의 적용

강화학습을 위한 환경 인식과 그에 따른 판단을 내릴 인공신경망을 구현하였으므로, 인공신경망이 판단을 올바르게 할 수 있도록 학습을 시켜야한다. 그 방법으로 유전 알고리즘을 사용하였다. 이는 심층학습에서 추론단계의 소비전력을 최적화시키는 문제와도 연관이 있다. 본 연구에서 사용된 인공신경망은 노드 10, 20, 10개의 은닉층과 노드 4개의

입력, 출력층으로 구성된 간단한 구조로서 복잡한 가중치를 가지는 깊은 인공신경망이 아니므로 추론 단계에서 최적화된 유전 알고리즘 사용이 가능한 것이다.

본 프로그램에서는 한 세대의 개체군을 16으로 설정하였으므로, 16개의 유전자가 모두 각각의 게임을 마치면 교배 연산과 변이 연산을 통해 진화가 되어 새로운 세대가 탄생한다. 개체군을 16으로 설정한 이유는 개체군이 너무 많으면 학습에 오랜 시간이 걸리기 때문이다. 플레이어가 상대해야 할 유령이 4개나 있는 팩맨 게임 특성상 한 게임당 한 개체를 학습시킬 수밖에 없으므로 개체군을 늘리면 그 만큼 학습시간이 늘어날 수 있다.

교배 연산은 일정 지점을 기준으로 두 유전자를 단순히 섞는 방법이다. 실질적으로 진화시킬 부분은 인공신경망의 가중치의 행렬 데이터 부분이므로, 랜덤으로 행렬의 열을 고른다. 그 열을 기점으로 두 행렬을 섞는 것이다. 이 과정이 한 인공신경망의 모든 가중치에서 반복된다. 이때 교배 연산을 위해 부모 유전자 두 개를 골라야 하는데 그 기준을 적합도 (Fitness)라고 한다.

적합도는 각 유전자가 문제 해결에 대해서 얼마나 적합한 지를 나타내는 수치이다. 따라서 이 팩맨 게임을 수행하는 데에 얼마나 적합한 지 알 수 있도록 설정해야 한다. 단순히 팩맨 게임의 결과인 포인트로 설정하지 않고, 게임의 가장 중요한 요소인 생존을 위해 유령 회피 횟수를 적합도로 선정했다. 팩맨이 진행하는 방향이 유령의 방향과 같으면 적합도는 감소하고, 유령의 방향과 다르면 적합도는 증가한다. 이렇게 설정한 적합도를 기준으로 상위 유전자를 4개 뽑는다. 이 4개 중 랜덤으로 2개씩 뽑아서 교배 연산을 한 후 자식 유전자 1개를 탄생시킨다. 이 과정을 총 4번 반복하여 기존에 존재하던 4개의 상위 유전자와 자식 유전자 4개, 총 8개의 유전자를 가지고 변이 연산을 진행한다.

변이 연산은 교배 연산으로만 진행할 시 생기는 지역 최적점을 돌파하기 위해 사용되는 연산이다. 앞서 생성했던 8개 유전자들의 각 가중치 값을 20% 확률로 변이시킨다. 20% 이하의 확률로 테스트를 해봤지만, 지역 최적점을 돌파하지 못하고 너무 오랜 세대동안 머무는 문제가 발생했기에 20%로 맞추었다. 변이 방법은 변이시킬 가중치 행렬 데이터와 같은 크기의 행렬을 난수로 생성하여 기존 가중치와 행렬 곱을 해서 새로운 행렬 데이터를 생성하는 것이다. 8개의 유전자에서 이 과정을 2번 반복하여 다음 세대인 16개의 유전자가 생성된다.

	Generation	Genomes	Evasion	Time	Score
0	1	0	1	7.83	510
1	1	1	-2	3.55	300
2	1	2	5	3.29	190
3	1	3	8	2.13	180
4	1	4	-2	3.64	270
5	1	5	10	4.21	230
6	1	6	14	2.35	140
7	1	7	2	1.23	70
8	1	8	-11	3.99	260
9	1	9	-5	5.66	330
10	1	10	7	5.29	390
11	1	11	9	3.75	170
12	1	12	1	2.42	130
13	1	13	-4	1.26	110
14	1	14	-2	0.96	70
15	1	15	9	3.09	210

그림 3. 저장된 게임 결과 데이터 파일

Fig. 3 Game result data file

새로운 세대의 유전자들은 다시 팩맨이 되어 각각 게임을 진행하고, 모든 게임이 끝나면 계산된 적합도를 기준으로 교배 연산과 변이 연산이 일어난다. 이러한 과정이 사용자가 종료할 때까지 계속해서 반복된다.

이와 같은 알고리즘을 프로그램 구현에 잘 적용하고, 특히 적합도를 올바르게 선정했다면, 시뮬레이터 기능 내에서 팩맨은 세대가 지날수록 게임 플레이를 능숙하게 할 것이다. 그에 따른 판단은 각 유전자가 마친 게임의 결과 데이터를 기반으로 분석할 것이다.

5. 적합도에 따른 데이터 변화 분석

이하 데이터 분석은 테스트 결과 중 하나를 분석한 것이다. 시뮬레이터 내에서 반복되는 각 게임이 끝날 때마다 해당하는 유전자의 세대와 유전자 번호, 게임의 목적에 해당하는 결과 값 (총 점수, 생존시간) 그리고 산출된 적합도를 그림 3과 같이 csv 파일로 저장한다. Evasion 컬럼이 적합도를 의미한다. 이렇게 한 세대에 유전자 번호가 0부터 15까지 존재하며, 세대가 지날수록 데이터가 어떻게 변하는지 분석했다.

그림 4는 세대가 지남에 따라 적합도와 생존시

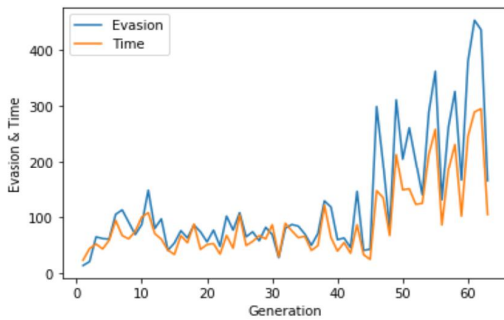


그림 4. 세대에 따른 적합도와 생존시간

Fig. 4 Fitness and survival time by generation

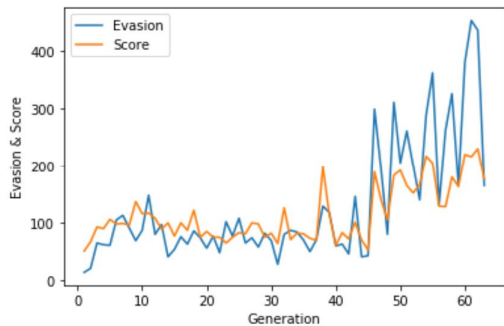


그림 5. 세대에 따른 적합도와 총 점수

Fig. 5 Fitness and total score by generation

간이 어떻게 변하는지 두 그래프를 겹쳐서 보여준다. 그림 5는 세대가 지남에 따라 적합도와 게임의 결과인 총 점수가 어떻게 변하는지 두 그래프를 겹쳐서 보여준다. X축은 세대를 나타내며, Y축은 각 세대가 가진 적합도와 게임 결과의 최대값이다.

그림 4와 그림 5를 보면 10세대 정도까지는 적합도와 함께 게임의 결과 값인 생존시간 및 총 점수가 증가하는 것을 볼 수 있으나, 40세대 정도까지 지역 최적점을 벗어나지 못하는 것을 확인할 수 있다. 하지만 40세대 중반쯤부터 적합한 변이가 일어나 지역 최적점을 돌파하고 높은 적합도와 게임 결과 값을 달성하는 것을 확인할 수 있다.

6. 플레이어와 AI의 게임 결과 비교 분석

강화학습이 가능한 팩맨 AI가 사람을 능가했는지 확인하기 위해 테스트 플레이어 10명의 게임 결과 데이터와 AI의 게임 결과 데이터를 비교, 분석하였다. 테스트 플레이어들은 최소 15번, 최대 22번의 게임을 진행하였으며 최대 점수와 평균 점수를

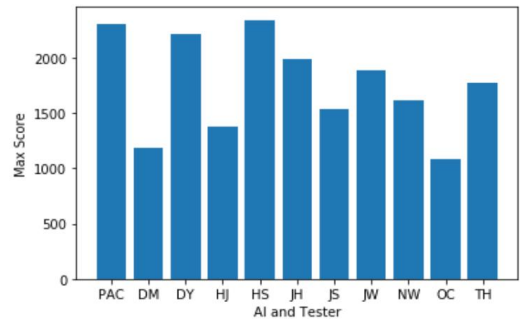


그림 6. AI의 최대 점수와 플레이어별 최대 점수

Fig. 6 Max score of AI and max score per player

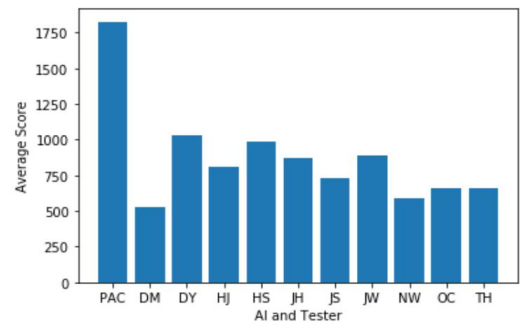


그림 7. AI의 평균 점수와 플레이어별 평균 점수

Fig. 7 Average score of AI and average score per player

계산하여 AI와 비교하였다.

분석 결과, 그림 6과 같이 팩맨 AI의 최대 점수보다 더 높거나 비슷한 점수를 기록한 플레이어들이 몇몇 있었지만 대체로 플레이어들이 더 낮은 점수를 기록하였다. 그러나 최대 점수에서 AI와 플레이어의 확연한 차이는 눈에 띄지 않았다.

적합도 200 이상을 달성한 팩맨 AI와 평균 점수를 비교한 그림 7을 보면 최대 점수 비교와 달리 AI와 플레이어의 평균 점수 차이가 확연하다는 것을 알 수 있다. 또한 적합도 400 이상을 달성한 팩맨 AI는 사람을 능가하는 플레이를 보여주었다. 팩맨 게임 규칙 상 플레이어는 정지할 수 없고 진행 방향으로 계속 나아갈 수밖에 없다. 하지만 팩맨 AI는 유령들에게 포위되어 어느 쪽으로든 움직일 수 없을 때, 사람과 다른 플레이를 보였다. 포위망이 뚫릴 때까지 정지해 있을 수 없으므로 사람이 조작할 수 없는 빠른 속도로 방향을 전환하여 마치 정지한 것처럼 기다렸다가 틈이 생겼을 때 포위망을 탈출하는 모습을 보였다.

IV. 결론 및 향후 연구

본 논문에서는 고성능의 팩맨 AI를 구현할 강화 학습을 위하여 유전 알고리즘 뿐 아니라 다층 퍼셉트론 구조의 간단한 인공신경망을 활용하였다. 미로와 같은 팩맨 게임의 맵 구조상, 주어진 환경을 인식하고 판단을 내릴 알고리즘으로 인공신경망이 적합하였다. 또한 유전 알고리즘의 핵심인 적합도를 설정함에 있어서 팩맨 게임의 점수인 포인트를 고려하지 않고, 게임의 가장 중요한 요소인 생존을 위해 유령을 회피한 횟수를 사용하여 시간에 구애받지 않고 높은 점수를 달성할 수 있게 하였다. 10여명의 플레이어로 테스트한 결과, 적합도 200이상을 달성한 팩맨 AI가 모든 참여자들보다 높은 평균 점수를 기록하는 것을 확인하였다. 또한 적합도 400 이상을 달성한 팩맨 AI는 유령으로 포위된 상황을 탈출함으로써 사람으로서는 불가능한 플레이를 보여주었다.

그러나 인공신경망에 유전 알고리즘을 단순히 결합한 형태의 진화속도 자체는 예상보다 느렸고 지역 최적점 문제 등의 한계로 인해 기존 강화학습 기법인 역전파 알고리즘보다 효율적인지는 의심스럽다. 그러므로 향후 연구에서는 각 세대의 신경망을 역전파 알고리즘으로 최적화시킨 뒤 이 신경망들을 유전 알고리즘으로 제차 선별하는 구조를 적용해 볼 필요가 있다.

References

- [1] R.S. Sutton, A.G. Barto, Reinforcement Learning :An Introduction, 2nd Edition, MIT Press, Cambridge, MA, 2018.
- [2] H.H. Lee, T.Y. Kim, M.J. Choi, "Deep Reinforcement Learning Application in Aerospace Filed with Unity," KSAS, Proc. of Conference, pp. 522-523, 2019 (in Korean).
- [3] S.Y. Jang, H.J. Yoon, N.S. Park, J.K. Yun, Y.S. Son, "Research Trends on Deep Reinforcement Learning," ETRI Electronics and Telecommunications Trends, Vol. 34, No. 4, pp. 1-14, 2019(in Korean).
- [4] Y.W. Shin, "Control of Intelligent Characters Using Reinforcement Learning," JICS, Vol. 8, No. 5, pp. 91-97, 2007 (in Korean).
- [5] Y.W. Shin, T.C. Chung, "Improvement of Sequential Prediction Algorithm for Player's Action Prediction," JICS, Vol. 11, No. 3, pp. 25-32, 2010 (in Korean).
- [6] S.Y. Park, P.W. Park, J.M. Kim, J.H. Borm, S.W. Lee, "A Study on Searching Optimal Path for Robot Using Genetic Algorithm," J. Korean Soc. Precis. Eng., Vol. 35, No. 12, pp. 1147-1155, 2018 (in Korean).
- [7] J.N. Kim, H.T. Kim, C.W. Ahn, "Routing Algorithm for Multiple Environment Using Genetic Algorithm," Proceedings of KIISE Conference, 38(2A), pp. 333-336, 2011 (in Korean).
- [8] J.M. Kim, S.J. Kim, S.M. Hong, "Players Adaptive Monster Generation Technique Using Genetic Algorithm," JIC, Vol. 18, No. 2, pp. 43-51, 2017 (in Korean).
- [9] S.W. Park, W.H. Lee, "Genetic Algorithm for Game Monster Generation," Proceedings of the Korea Contents Association Conference, Vol. 4, No. 2, pp. 811-814, 2006 (in Korean).
- [10] T.Y. Kim, J.S. Choi, "Game Difficulty Controlling Using Evolutionary Algorithm," Journal of the Korean Society for Computer Game, No. 11, pp. 20-27, 2007 (in Korean).
- [11] C.H. Jung, C.Y. Park, S.D. Chi, J. Kim, "The Battle Warship Simulation of Agent-based with Reinforcement and Evolutionary Learning," Journal of the Korea Society for Simulation, Vol. 21, Issue 4, pp. 65-73, 2012 (in Korean).
- [12] T.R. Arungpadang, Y.J. Kim, "A Study on Dual Response Approach Combining Neural Network and Genetic Algorithm," Journal of the Korean Institute of Industrial Engineers, Vol. 39, No. 5, pp. 361-366, 2013 (in Korean).
- [13] D.H. Oh, "A Study on Pathfinding in Game Environment Using Genetic Algorithm and Neural Network," Proceedings of the Korea Information Processing Society Conference, Vol. 23, No. 2, pp. 607-608, 2016 (in Korean).
- [14] O.K. Kwun, J.K. Park, "Control of RPG Game Characters using Genetic Algorithm and Neural Network," Journal of Korea Game Society, Vol. 6, No. 2, pp. 13-22, 2006 (in Korean).

Jin-Soo Park (박진수)

He is a prospective graduate of the Dept. of Computer Science and Information Engineering,

He will be received B.E. degree in 2021. He is interested in artificial intelligence and game simulator.

Email: jsorry0130@naver.com

Doo-Yeon Hwang (황두연)

He is a prospective graduate of the Dept. of Computer Science and Information Engineering,

He will be received B.E. degree in 2021. He is interested in artificial intelligence and game simulator.

Email: dyrhl456@naver.com

Ho-Jeong Lee (이호정)

He is a prospective graduate of the Dept. of Computer Science and Information Engineering,

He will be received B.E. degree in 2021. He is interested in artificial intelligence and game simulator.

Email: turtle8076@naver.com

Soosun Cho (조수선)

She is a professor in the Dept. of Computer Science and Information Engineering, She received BS and MS degrees from Seoul National University,

Dep. of Computer Science & Statistics and Ph.D. from

Dept. of Computer Science. She has been a senior researcher at ETRI for 10 years. She is interested in statistical machine learning and related fields.

Email: