

Game Competition과 기계학습

세종대학교 | 김경중*
연세대학교 | 조성배**

1. 서론

최근 게임지능 국제학술대회를 중심으로 카레이싱, Ms. 팩맨, 슈퍼마리오, 언리얼 토너먼트, 스타크래프트 등의 다양한 게임 Competition이 열리고 있으며, 이를 통해 게임 인공지능 기술의 발전을 도모하고 있다(표 1). Competition에서는 제공되는 게임 플랫폼의 소프트웨어를 토대로 참가자들이 개발한 인공지능 프로그램을 대회 날에 수행시켜 가장 우수한 성능을 낸 것을 우승자로 결정하는데, 비교적 짧은 역사에 비해 비디오 게임과 같은 실제 게임을 위한 인공지능 기술 발전에 크게 기여하고 있다. 본 논문은 이러한 게임 Competition에서 널리 사용되는 기계학습 기법들을 각 대회별로 우수한 프로그램이 사용했던 방법을 중심으로 소개하고 그 가능성을 조망한다.

표 1 2010년도 게임지능 경진대회 현황 및 주요 웹 사이트

2010년도 대회	AIIDE	WCCI	CIG	EvoStar	GECCO	ICE-GIC
Car Racing		○	○	○	○	
Ms. Pac-Man		○	○			
Super Mario		○	○	○		○
UT			○			
StarCraft	○		○			

	웹사이트
Car Racing	http://cig.ws.dei.polimi.it/
Ms. Pac-Man	http://cswww.essex.ac.uk/staff/sml/pacman/PacManContest.html
Super Mario	http://www.marioai.org/
UT	http://www.botprize.org/
StarCraft	http://eis.ucsc.edu/StarCraftAICompetition

* 정회원

** 종신회원

† 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(2010-0012876) 및 뇌과학원천기술개발사업(2010-0018948)의 지원에 의한 것임.

더 나아가 최근의 새로운 Competition 동향과 기계학습 기술 변화를 다루도록 한다. 이전에는 각 게임에서 가장 높은 성적을 얻는 것을 목표로 대회를 개최해 왔지만, 최근에는 “사람과 얼마나 유사한가”, 혹은 “사람을 얼마나 즐겁게 하는가”, “새로운 환경에 가장 잘 적응하는 것은 무엇인가” 등 보다 높은 수준의 새로운 문제를 제시하고 있다. 이러한 경향에 따라 기계학습 기술이 단순히 전문지식으로 작성한 시스템의 매개변수를 최적화하는 수준이 그치지 않고, 사람의 게임 플레이 행태를 토대로 모방학습을 수행하는 등의 보다 고차원적인 수준으로 활용되고 있다. 이를 통해 현실 세계의 문제를 해결하는데 기계학습이 어떻게 공헌할 수 있는지 논한다.

2. 게임 인공지능 경진대회

2.1 카레이싱

이 경진대회의 목적은 다양한 센서를 갖춘 경주용 자동차를 운전하는 프로그램을 개발하는 것이다[1]. 이 경진대회는 2005년부터 시작되었으며, 초창기에는 무선으로 조정하는 카메라 기반 장난감 자동차를 대상으로 하였다. 그 이후 2차원 자동차 경주 시뮬레이터를 이용하여 2007년도까지 진행되었고, 2008년부터는 오픈 소스 프로젝트인 TORCS를 활용하여 보다 사실감

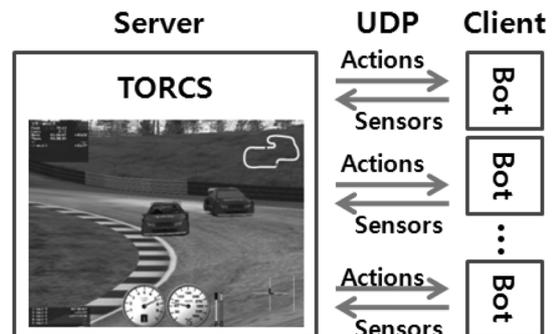


그림 1 자동차 경주 경진대회 서버-클라이언트 SW구조

있는 경진대회를 열고 있다. 현재 진행 중인 경진대회 중에서 가장 많은 참가자를 확보하고 있다.

경진대회 프로그램은 서버-클라이언트 구조로 되어 있다(그림 1). 참가자들은 자신만의 클라이언트 프로그램을 개발하여 제출한다. 각각의 클라이언트 프로그램은 서버로부터 경주 시뮬레이션 환경에 대한 각종 정보를 전달받고, 자신의 자동차를 어떻게 주행할지에

Sensors (79)		Actuators (7)
Gear	Current Lap Time	Changing Gear
Last Lap Time	Distance from Start	
Race Position	Distance Raced	
RPM	Track Position	
Speed X/Y/Z	Wheel Spin Velocity (4)	
Track (19)	Opponents (36)	
Damage		
Angle		Steering the wheel
Focus (5)		Restart the race
Fuel		Focusing
Z		Clutch
		Break
		Accelerate

그림 2 Simulated 자동차 경주에서 센서와 실행기

```

// SimpleDriver.cpp
int SimpleDriver::getGear(CarState &cs){
    // 기어변경 코드
    return gear;}
float SimpleDriver::getSteer(CarState &cs){
    // 방향 변경 코드
    return steer;}
float SimpleDriver::getAccel(CarState &cs){
    // 가속도 및 브레이크 조절 코드
    return accel_and_brake; }
CarControl SimpleDriver::wDriver(CarState cs){
    // Stuck 상황여부 판단
    If(Stuck){ // Stuck 상황 회복 코드 }
    Else {
        float accel_and_brake = getAccel(cs);
        int gear = getGear(cs);
        float steer = getSteer(cs);
    }
    clutching(cs,&clutch);
    CarControl cc(accel,breake, gear, steer, clutch);
    Return cc;}
void SimpleDriver::clutching(CarState &cs, float &clutch){
    // 클러치 조절 코드 }
void SimpleDriver::init(float *angles){
    // 트랙센서 위치 세팅 코드 }

```

그림 3 자동차 경주 API (C++ 버전)

대한 명령 정보를 전송한다(그림 2). 서버는 전송받은 명령들을 시뮬레이션 환경에 반영하여 경주를 진행한다. 그림 3은 제공하는 API를 보여준다.

대회가 진행되면서, 우승을 차지하는 프로그램들이 사용하는 전략에도 많은 변화가 있어왔다. 2008년도 대회에서는 진화 신경망을 사용한 제어 프로그램이 우승을 차지하였다[2]. 자동차 경주에 대한 전문지식을 전혀 활용하지 않고, 진화 신경망이 학습을 통해 자동차를 운전하도록 만들었다. 2009년도부터는 전문지식을 활용하여 설계한 경우, 기계학습만을 활용한 경우, 둘을 함께 사용한 경우 등 다양한 참가자가 등장하였다. 2009년도의 경우 전문지식을 주로 활용한 팀이 우승하였다[3]. 2010년도의 경우 기계학습을 사용한 경우와 전문지식을 사용한 경우의 경합이 치열하였는데, 근소한 차이로 전문지식을 사용한 팀이 우승할 정도로 기계학습 방법의 발전이 괄목할만하다.

2.2 Ms. 팩맨

팩맨은 세계적으로 인기를 끌어난 아케이드 게임인데, Ms. 팩맨은 기존 팩맨과 달리 고스트들의 움직임에 불확실성이 존재한다. 팩맨은 하얀 점으로 표시되어 있는 약을 먹으며 미로를 돌아다닌다. 네 모퉁이에는 “파워필”이라고 불리는 반짝이는 큰 점이 있는데, 이 약을 먹으면 잠깐 동안 고스트가 팩맨을 잡아먹을 수 있는 모드로 바뀐다. 이 모드에서 고스트를 연속으로 잡을수록 큰 점수를 얻을 수 있다.

Ms. 팩맨 경진대회는 2007년부터 시작되었으며, 대회의 목적은 팩맨을 조정하여 가장 높은 점수를 얻는 것이다[4]. 현재까지 사람이 올린 가장 높은 팩맨 점수는 921,360점이다. 2007년도부터 시작된 경진대회를 통해 컴퓨터가 획득한 가장 높은 점수는 30,010점으로 아직까지 컴퓨터와 사람 사이에 큰 격차가 존재한다.

경진대회는 웹기반 또는 Microsoft Revenge of Arcade의 Ms. 팩맨을 사용한다. 경진대회에서 제공하는 소프트웨어는 지속적으로 게임의 실행 화면을 Capture하

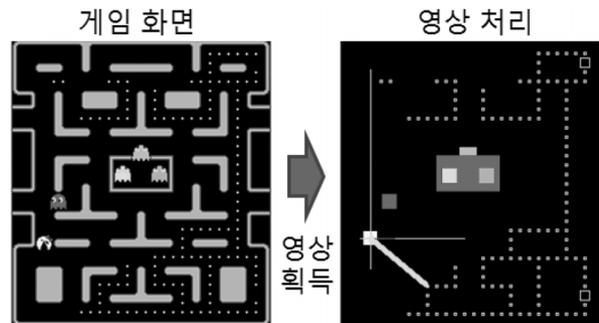


그림 4 화면 Capture를 통한 게임 정보 획득

```

public override Direction Think(GameState gs) {
    Direction bestDirection = Direction.None;
    double[] dist = { Double.PositiveInfinity,
                    Double.PositiveInfinity,
                    Double.PositiveInfinity,
                    Double.PositiveInfinity };
    foreach (Ghost ghost in gs.Ghosts)
    {
        if (ghost.Node == null) { continue;}
        Node.PathInfo path = gs.Pacman.Node.ShortestPath[ghost.Node.X, ghost.Node.Y];
        if (path != null){
            dist[(int)path.Direction] = path.Distance;
        }
    }
    double bestDistance = 0.0;
    for (int i = 0; i < dist.Length; i++)
    {
        if (gs.Pacman.PossibleDirection((Direction)i) && dist[i] > bestDistance)
        {
            bestDirection = (Direction)i;
            bestDistance = dist[i];
        }
    }
    return bestDirection;
}

```

그림 5 팩맨 제어를 위한 C#프로그램 예 (가장 안전한 방향, 즉 고스트와 가장 멀어지는 방향으로 이동하는 코드)

여 제공해주며, 이를 분석하여 게임 오브젝트의 위치 및 미로 형태를 파악한다(그림 4). 참가자가 제출한 프로그램은 이러한 정보를 토대로 팩맨이 어느 방향으로 이동하는 것이 좋을지를 결정한다. 프로그램이 결정한 팩맨의 이동방향은 실행중인 팩맨 게임에 전달되어 반영된다. 팩맨 프로그래밍은 C#과 Java버전이 지원되고 있으며, 각종 함수 API를 제공하고, 사용자는 의사결정 부분을 작성한다(그림 5).

2009년도 대회에서 우승한 일본의 리츠메이칸 대학 팀은 총 9개의 규칙을 사용하였다[5]. 각 규칙의 파라미터는 경험적으로 결정되었다. 고스트들로부터 가장 멀어지는 방향으로 이동하고, “파워필”을 잘 활용하여 많은 고스트를 잡은 것이 성적 상승의 원인이었다. 2010년도 대회에서는 리츠메이칸 대학팀이 진화연산을 이용하여 파라미터를 최적화 하고 나왔으나, 우승은 개미최적화 알고리즘을 사용한 팀에게 돌아갔다[6].

2.3 슈퍼마리오

슈퍼마리오는 닌텐도의 가장 성공적인 비디오 게임 중의 하나이다. 게임에서 마리오는 오른쪽 방향으로 진행하며 아이템을 획득하고, 적들을 피해 출구로 나가야 한다. 본 경진대회는 슈퍼마리오를 변형한 Infi-

nite Mario를 토대로 이루어진다. 이 버전은 기존 슈퍼마리오와 유사하나, 매 단계별 게임의 난이도가 임의로 결정되고, 자바 소스코드가 공개되어 있다. 슈퍼마리오는 자신을 중심으로 $N \times N$ 격자 공간을 구성하여 해당 셀에 어떤 게임 객체가 있는지 알 수 있다. 이를 이용하여 “왼쪽”, “오른쪽”, “자세 낮추기”, “점프”, “스피드”키 중 어떤 것을 누를지 결정하며, JAVA 기반의 API를 이용한다(그림 6, 7).

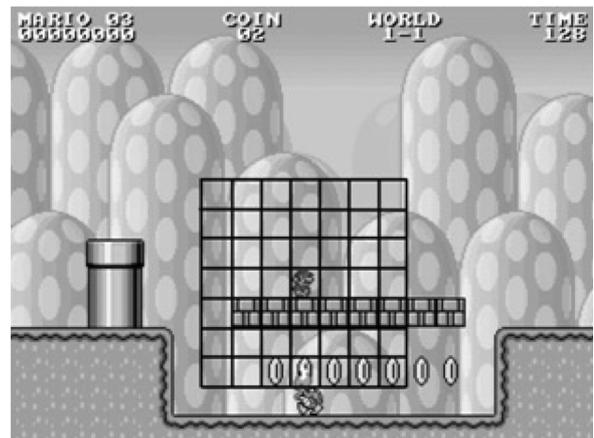


그림 6 슈퍼마리오의 센서 모델(슈퍼마리오를 중심으로 7X7 격자 공간에 대해 객체정보 반환)

```

public boolean[] getAction()
{
    // 의사결정을 위한 코드
    return action;
}

public void integrateObservation(Environment environment)
{
    levelScene = environment.getLevelSceneObservationZ(zLevelScene);
    enemies = environment.getEnemiesObservationZ(zLevelEnemies);
    mergedObservation = environment.getMergedObservationZZ(1, 0);
    this.marioFloatPos = environment.getMarioFloatPos();
    this.enemiesFloatPos = environment.getEnemiesFloatPos();
    this.marioState = environment.getMarioState();
    receptiveFieldWidth = environment.getReceptiveFieldWidth();
    receptiveFieldHeight = environment.getReceptiveFieldHeight();
    marioStatus = marioState[0]; marioMode = marioState[1];
    isMarioOnGround = marioState[2] == 1; isMarioAbleToJump = marioState[3] == 1;
    isMarioAbleToShoot = marioState[4] == 1; isMarioCarrying = marioState[5] == 1;
    getKillsTotal = marioState[6]; getKillsByFire = marioState[7];
    getKillsByStomp = marioState[8]; getKillsByShell = marioState[9];
}

```

그림 7 슈퍼마리오 프로그래밍 API(JAVA버전)

2009년부터 경진대회가 개최되었으며, 대회의 목적은 마리오가 죽지 않으면서 가장 멀리까지 이동하는 것이다. 매번 단계를 완료하면, 새로운 단계가 생성된다. 이러한 과정을 총 50번 반복하고, 마리오가 가장 멀리 이동한 프로그램이 우승한다[7]. 자동차 경주와 유사한 방식으로 매년 통합 챔피언을 결정한다. 2010년도 12월에는 사람이 플레이하는 것과 가장 유사한 프로그램을 만드는 것을 대상으로 경진대회가 열리기도 하였다.

2009년도 대회에서 우승한 프로그램은 기본적인 탐색 알고리즘 중의 하나인 A* 알고리즘을 사용하였다. 마리오가 택할 수 있는 여러 가지 경로 중에서 마리오가 피해를 입지 않으면서 출구와 가장 가까워지는 것을 택하도록 하였다. 이 대회 우승자의 동영상은 Youtube에 공개되어 폭발적인 관심을 받았다¹⁾.

3. 기계학습 기술 동향

3.1 전문지식과 기계학습 기술의 통합

현재까지의 경향을 보면, 좋은 성적을 거두는 팀의 경우 전문적인 게임 지식과 기계학습 기술을 혼합해서 사용하는 경우가 많았다. 전문지식만으로 게임 제어를 설계하는 경우에 많은 파라미터의 설정이라는

어려움에 직면하게 되고, 경우에 따라 사람이 설계하기 불가능한 부분도 있다. 순수하게 기계학습만을 활용하여 모든 것을 설계하는 것도 가능한 하지만, 처음부터 모든 것을 학습으로 설계하는 것은 비효율적인 경우도 많다. 간단한 규칙 하나로 해결될 것을 학습을 통해 설계하면, 매우 오랜 시간이 걸려야 하거나 실패하는 경우도 종종 있다.

이러한 이유로 게임지능 경진대회에 출전하는 프로그램의 많은 경우는 게임 지식과 기계학습 기술을 함께 사용하여 좋은 성적을 올려왔다. 자동차 경주분야에서 가장 좋은 성적을 올리고 있는 두 팀을 예로 들어보자. Onieva & Pelta 팀은 모듈형 구조로 자동차 제어를 설계하였다. 각 모듈을 어떻게 설계할지 각각의 모듈이 어떤 연결 관계를 가지는지 등에 전문지식이 주로 활용되었다[3]. 전문지식은 퍼지규칙 형태로 표현되었으며, 최근에는 유전자 알고리즘을 이용하여 규칙을 최적화 하는 시도를 하고 있다[8]. COBOSTAR 팀의 경우 기계학습 기술을 적극적으로 활용한다[9]. 제어 프로그램에서 정의되는 모든 파라미터들을 CMA-진화 전략을 이용하여 최적화하였다. 제어 프로그램을 트랙 안쪽에 있을 때와 바깥쪽에 있을 때로 구분하여 설계하는 등 다양한 휴리스틱이 들어가 있다.

Ms. 팩맨의 경우 규칙기반 기법이 좋은 성과를 올리고 있다. 2009년에 가장 높은 성적을 올린 일본의 리

1) <http://www.doc.ic.ac.uk/~rb1006/projects/marioai>

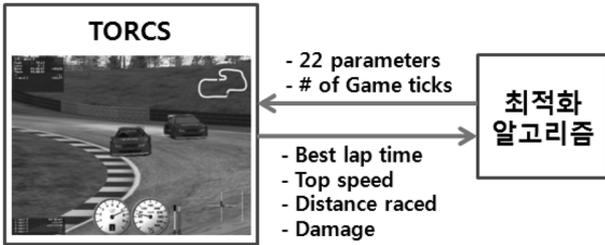


그림 8 자동차 경주기반 최적화 알고리즘 경진대회

츠메이칸 팀의 경우 9개의 규칙으로 프로그램을 작성하였다. 프로그램의 파라미터는 모두 경험적으로 최적화 하였다. 2010년의 경우 경험적으로 최적화 했던 파라미터를 진화전략을 이용하여 자동으로 결정하였다[10]. 2010년의 경우 개미집단 최적화 방법을 적용한 팩맨 제어가 우승을 차지하였다[6]. 2010년에 들어 팩맨 분야에 기계학습의 필요성이 커지고 있음을 보여주었다. 팩맨 분야의 경우 기계학습이 매우 필요하지만, 대회방식이 화면 Capture를 통해 이루어지기 때문에 학습이 어려운 측면이 있다. 가장 큰 어려움은 화면 Capture 및 영상처리로 인해 평가하는데 들어가는 시간이 매우 크다는 점이다. 화면 Capture를 사용하지 않는 Ms. 팩맨 시뮬레이터를 사용하는 것도 한 방법이지만, 실제 대회 방식으로 평가할 때와 차이가 크기 때문에 한계가 있다.

3.2 학습기술 중심의 경진대회

기존의 게임 인공지능 경진대회들의 참가자들이 기계학습 기법을 많이 활용해 왔지만, 순수하게 기계학습만으로 프로그램 전체를 구성하는 경우보다는 전문 지식을 활용하는 쪽이 유리했다. 이러한 이유로, 일반적인 게임 플레이 경진대회와 함께 순수한 학습 능력만을 평가할 수 있는 새로운 형태의 대회를 개최하기 시작했다.

자동차 경주의 경우 2009년도부터 Car Setup Optimization 경진대회를 개최해 오고 있다(그림 8). 이 대회의 목적은 주어진 시간 안에 경주용 자동차의 다양한 파라미터를 최적화하는 학습 알고리즘을 만드는 것이다. 참가자들에게는 2시간의 CPU사용 가능 시간을 주고 22가지 경주용 자동차의 파라미터를 최적화 하도록 한다. API함수를 통해 새로운 파라미터 조합이 해당 트랙에서 어떤 결과를 보였는지 전달 받을 수 있다. 이러한 정보를 토대로 학습 알고리즘은 지속적으로 최적화를 수행한다.

2009년 대회의 경우 총 5팀이 참여하였으며, Multi-Objective Evolutionary Algorithm(MOEA), Particle Swarm Optimization(PSO), Simple Genetic Algorithm(SGA), Co-

variance Matrix Adaptation Evolutionary Strategy(CMA-ES) 등의 기법이 사용되었다. 제한된 자원 아래에서 22개의 파라미터를 최적화 하는 문제의 경우 CMA_ES > MOEA > SGA > PSO 순으로 성능을 보였다. 상대적으로 SGA의 성능이 높게 나와, 성능향상의 여지가 많이 있음을 보여주었다.

슈퍼마리오 경진대회에서도 2010년부터 학습 능력을 평가하는 대회를 개최하였다. 총 5개의 난이도를 가진 레벨이 주어지며, 각 프로그램은 각 레벨별로 10,000개의 에피소드를 실행해 볼 수 있다. 10,001번째 성적으로 해당 프로그램의 학습 능력을 평가한다.

2010년 대회에서는 총 4팀이 참가하였으며, Bio-inspired 알고리즘이 많이 활용되었다. 우승팀은 마리오의 행동을 결정하는 규칙들을 진화 연산으로 학습하였다. 2위 팀은 Cuckoo 탐색을 활용하여 마리오의 의사결정을 학습하였다. 이 알고리즘은 빠꾸기가 다른 새의 등지에 알을 놓은 행동을 모방하였다. 3위 팀은 진화연산의 일종인 유전자 프로그래밍을 사용하였다.

기계학습 분야에서는 UCI Machine Learning Repository²⁾가 관련 기술의 수준을 발전시키는데 큰 기여를 해왔다. 이곳에 있는 데이터를 이용하면 새로운 기계학습 기술이 우수한지를 기존의 방법과 객관적으로 비교할 수 있다. 하지만, 기계학습 기술에는 UCI Machine Learning Repository로 평가하기 어려운 부분이 많이 있다. 특히 최적화와 강화학습 등이 그 예이다. 이러한 분야를 평가하기 위해 게임은 매우 좋은 플랫폼이 될 수 있다. 경진대회를 통해 잘 갖추어진 소프트웨어 평가기준이 널리 활용될 수 있다면, 기계학습 분야의 기술발전에 기여할 수 있을 것이다.

3.3 모방학습

모방학습은 다른 프로그램이나 사람이 게임을 하는 것을 보고 배우는 것을 말한다. 게임 인공지능을 설계하는 것은 많은 노력을 필요로 한다. 게임 자체에 대한 전문적인 지식을 가지고 있어야 하거나, 기계학습에 대한 전문성이 요구된다. 하지만, 이미 잘 만들어진 프로그램이나 전문가 수준의 사람 플레이어가 게임을 하는 과정을 지켜보는 것만으로 프로그램이 학습을 수행할 수 있다면 매우 유용할 것이다.

모방학습에 관한 연구가 가장 활발한 곳은 자동차 경주 분야이다. Hoorn은 MOEA를 활용하여 게임을 잘 하면서 사람의 플레이와 유사한 신경망을 탐색하도록 하였다[11]. Munoz는 두 개의 프로그램(WCCI 2008

2) www.ics.uci.edu/~mllearn/

우승프로그램과 Hand-Coded 프로그램)과 한명의 사람으로부터 게임 플레이 데이터를 수집했다[12]. 이 연구의 특이한 점은 프로그램과 사람의 게임 플레이로부터 획득한 데이터를 함께 사용해서 모방학습을 했다는 점이다. 실험결과 사람의 행동을 모방하는 것은 매우 어렵지만, 프로그램의 행동을 모방하는 것은 가능성이 있음을 보였다. 사람과 프로그램의 플레이 데이터를 섞어서 사용하는 경우 모방학습은 성공적이지 못했다. Cardamone은 저수준의 정보만으로 모방학습이 성공하기 어렵다고 판단하여 고수준의 정보를 활용하는 방법을 제안하였다[13]. 즉 여러 개의 저수준 센서정보를 병합하여 사람이 판단할 때 인식하는 정도의 고수준 정보를 생성하고 이를 학습 데이터로 사용하려 하였다. 실험결과 TORCS에서 가장 잘 달리는 자동차의 약 85% 정도 수준의 프로그램을 모방학습을 통해 만들 수 있음을 보였다. Munoz는 사람이 TORCS를 플레이한 데이터로부터 경로와 속도를 예측하는 신경망을 학습하였다[14].

모방학습이 성공하기 위해서는 여러 가지 측면을 고려해야 한다. 일반적으로 사람이 플레이하는 것을 모방하는 것이 프로그램을 모방하는 것보다 어렵다고 알려져 있다. 그 이유는 사람은 게임을 할 때 프로그램과 달리 매우 고차원적인 정보들을 활용하기 때문이다. 단순히 현재 게임에서 수집 가능한 저수준의 센서 정보만으로 사람이 인식하는 것을 모방하기에는 그 격차가 매우 크다. 또한 사람은 유사한 상황에서도 매우 다른 형태로 플레이를 할 수 있기 때문에 제한된 양의 데이터로부터 일반화 하는 것이 어려울 수 있다.

프로그램을 모방하는 경우에 고려할 점은 학습에 도움이 될 수 있는 다양한 데이터를 수집하는 것이다. 여러 가지 다양한 상황에서 프로그램의 행동을 수집할 수록 일반화가 용이하기 때문이다. 모방에만 치우치면 플레이어의 데이터와 매우 유사한 결과를 보이지만, 실제 게임에서는 아주 낮은 성능을 보이는 경우가 있다. 이러한 문제를 해결하기 위해 플레이어와의 유사성과 함께 실제 플레이 수준도 함께 최적화 하는 MOEA 기법 등이 필요하다.

3.4 온라인학습

게임 지능 경진대회의 경우 대회에서 사용할 트랙 또는 맵은 사전에 공개되지 않는다. 즉, 대회 당일에 공개하여 특정 트랙이나 맵에 최적화하는 것이 불가능하도록 해놓았다. 이러한 이유로 사전에 다양한 환경에서 제어 프로그램을 테스트하여 일반성을 높이는 것이 필요하다. 하지만, 모든 경우를 대비할 수 없기

때문에 실제 경기를 진행하는 중에 실시간으로 학습하는 과정이 필요하다. 특히, 수십 바퀴를 도는 자동차 경주의 경우, 처음에는 최적화가 안 되어 있어 결과가 좋지 않더라도 시간이 지나가면서 성능이 좋아진다면, 좋은 결과를 기대할 수 있다.

Seo는 단순 기억기반의 기계학습을 이용하여 점차적으로 성능을 향상시키는 방법을 사용했다. 초반 레이스에서 사고가 난 지점을 기억하고 있다가, 그 지점을 다음번에 다시 만나게 되면 속도를 감속하는 방법을 사용하였다[15]. 아무리 최적의 제어를 사전에 설계한다고 하더라도 사고는 발생할 수 있기 때문에 실행하는 동안 발생한 사고를 활용하는 기술이 필요하다.

Cardamone은 실시간 진화 신경망기술을 TORCS에 적용하였다[16]. 이 연구에서는 NeuroEvolution with Augmenting Topology(NEAT)의 온라인 버전 알고리즘과 rtNEAT(Real-Time NEAT)을 사용하였다. 대표적인 진화 신경망 모델인 NEAT에 실시간 학습 기술을 적용한 모델이다³⁾. 실험 결과 온라인 알고리즘을 사용할 경우 전통적인 오프라인 기법과 유사한 성능을 보였으며, 매우 어려운 트랙에서는 더 뛰어난 성능을 보여주었다.

4. 결론

게임 지능 경진대회는 기계학습 기술을 발전시킬 수 있는 좋은 플랫폼을 제공해 주고 있다. 기존의 전통적인 보드게임을 벗어나 보다 현실적이고 흥미로운 비디오 게임에 기계학습 기술을 적용함으로써 새로운 기술을 개발할 수 있는 가능성을 열어주고 있다. 자동차 경주, 팩맨, 슈퍼마리오, 언리얼 토너먼트, 스타크래프트 등이 최근에 활발히 열리고 있는 게임지능 경진대회 종목이다.

경진대회의 목적이 초창기에는 단순히 좋은 성적을 거두는 프로그램을 찾는 데 있었다면, 최근 들어서는 학습 성능이 우수한 프로그램, 사람과 닮은 행동을 보이는 프로그램, 흥미로운 게임 콘텐츠를 생성하는 프로그램 등을 찾는 방향으로 확장되고 있다. 이러한 추세는 기계학습 분야에도 새로운 문제들을 제기하고 있으며, 기계학습의 좋은 응용 분야로 등장하고 있다.

게임 인공지능 경진대회 플랫폼은 대부분 무료이고, 규칙도 사전에 정의되어 있으며, 참가자들의 소스코드도 공개되어 있는 경우가 많다. 이러한 자원을 잘 활용한다면, 기계학습 연구에 유익할 수 있고, 또한 교

3) <http://nn.cs.utexas.edu/?rtNEAT>

육적인 측면에서 기계학습을 학생들이 손쉽게 익히도록 하는 좋은 재료가 될 수 있다.

현재 활발히 진행 중인 5가지 게임 인공지능 경진 대회 종목과 관련한 각종 정보는 인터넷 사이트 (<http://dasan.sejong.ac.kr/~kimkj/cim/>)를 참고하기 바란다.

참고문헌

- [1] D. Loiacono et al. "The 2009 simulated car racing championship," IEEE Trans. on Computational Intelligence and AI in Games, vol. 2, no. 2, pp. 131-147, 2010.
- [2] D. Loiacono et al. "The WCCI 2008 simulated car racing competition," Proceedings of the IEEE Symposium on Computational Intelligence and Games, pp. 119-126, 2008.
- [3] E. Onieva, D. A. Pelta, J. Alonso, V. Milanés, and J. Perez, "A modular parametric architecture for the TORCS racing engine," IEEE Symposium on Computational Intelligence and Games, pp. 256-262, 2009.
- [4] S. Lucas, "Ms Pac-Man competition," SIGEvolution, vol. 2, no. 4, pp. 37-38, 2007.
- [5] R. Thawonmas and H. Matsumoto, "Automatic controller of Ms. Pac-Man and its performance: Winner of the IEEE CEC 2009 software agent Ms. Pac-man competition," CD-ROM Proc. of Asia Simulation Conference, 2009.
- [6] M. Emilio, M. Moises, R. Gustavo, and S. Yago, "Pac-mAnt: Optimization based on ant colonies applied to developing an agent for Ms. Pac-Man," IEEE International Conference on Computational Intelligence and Games, pp. 458-464, 2010.
- [7] J. Togelius, S. Karakovskiy, and R. Baumgarten, "The 2009 Mario AI competition," IEEE Congress on Evolutionary Computation, 2010.
- [8] E. Onieva, J. Alonso, J. Perez, V. Milanés, and T. de Pedro, "Autonomous car fuzzy control modeled by iterative genetic algorithms," IEEE International Conference on Fuzzy Systems, pp. 1615-1620, 2009.
- [9] M. V. Butz and T. D. Lonnerker, "Optimized sensory-motor couplings plus strategy extensions for the TORCS car racing challenge," IEEE Symposium on Computational Intelligence and Games, pp. 317-324, 2009.
- [10] R. Thawonmas, and T. Ashida, "Evolution strategy for optimizing parameters in Ms Pac-Man controller ICE Pambush 3," IEEE International Conference on Computational Intelligence and Games, pp. 235-240, 2010.
- [11] N. V. Hoorn, J. Togelius, D. Wierstra, and J. Schmidhuber, "Robust player imitation using multiobjective evolution," Proceedings of the 11th Conference on Congress on Evolutionary Computation, pp. 652-659, 2009.
- [12] J. Munoz, G. Gutierrez and A. Sanchis, "Controller for TORCS created by imitation," IEEE Symposium on Computational Intelligence and Games, pp. 271-278, 2009.
- [13] L. Cardamone, D. Loiacono and P. L. Lanzi, "Learning drivers for TORCS through imitation using supervised methods," IEEE Symposium on Computational Intelligence and Games, pp. 148-155, 2009.
- [14] J. Munoz, G. Gutierrez, and A. Sanchis, "A human-like TORCS controller for the simulated car racing championship," IEEE International Conference on Computational Intelligence and Games, pp. 473-480, 2010.
- [15] J.-H. Seo, J.-G. Park, J.-H. Lee, and K.-J. Kim, "Designing robust robotic car controllers based on artificial neural network," International Conference on Convergence & Hybrid Information Technology, pp. 183-190, 2010.
- [16] L. Cardamone, D. Loiacono, and P. L. Lanzi, "Learning to drive in the open racing car simulator using on-line neuroevolution," IEEE Transactions on Computational Intelligence and AI in Games, 2010 (in print).

약 력



김경중

2007 연세대학교 컴퓨터과학과(박사)
 2007~2009 코벨대학교 박사후 연구원
 2009~현재 세종대학교 컴퓨터공학과 조교수
 관심분야 : 진화 연산, 로봇 지능, 게임 지능, 3차원 프린터
 E-mail : kimkj@sejong.ac.kr



조성배

1993 한국과학기술원 전산학과(박사)
 1993~1995 일본 ATR 연구소 연구원
 1995~현재 연세대학교 컴퓨터과학과 교수
 관심분야 : 인공생명, 신경망, 진화 연산, 소프트웨어 컴퓨팅
 E-mail : sbcho@cs.yonsei.ac.kr