韓國IT서비스學會誌
第15卷 第4號
2016年 12月, pp.85-95

Journal of Information Technology Services
https://doi.org/10.9716/KITS.2016.15.4.085

픽셀 데이터를 이용한 강화 학습 알고리즘 적용에 관한 연구*

문새마로** · 최용락***

A Study on Application of Reinforcement Learning Algorithm Using Pixel Data*

Saemaro Moon** · Yonglak Choi***

■ Abstract ■

Recently, deep learning and machine learning have attracted considerable attention and many supporting frameworks appeared. In artificial intelligence field, a large body of research is underway to apply the relevant knowledge for complex problem-solving, necessitating the application of various learning algorithms and training methods to artificial intelligence systems. In addition, there is a dearth of performance evaluation of decision making agents. The decision making agent that can find optimal solutions by using reinforcement learning methods designed through this research can collect raw pixel data observed from dynamic environments and make decisions by itself based on the data. The decision making agent uses convolutional neural networks to classify situations it confronts, and the data observed from the environment undergoes preprocessing before being used. This research represents how the convolutional neural networks and the decision making agent are configured, analyzes learning performance through a value-based algorithm and a policy-based algorithm : a Deep Q-Networks and a Policy Gradient, sets forth their differences and demonstrates how the convolutional neural networks affect entire learning performance when using pixel data. This research is expected to contribute to the improvement of artificial intelligence systems which can efficiently find optimal solutions by using features extracted from raw pixel data.

Keyword : Artificial Intelligence, Reinforcement Learning, CNN(Convolutional Neural Networks), DQN(Deep Q-Networks), PG(Policy Gradient)

Submitted : October 17, 2016 1st Revision : October 26, 2016 Accepted : October 28, 2016

* 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 SW특성화대학원 지원사업의 연구결과로 수행되었음(과제번호 : R0346-16-1010).

** 숭실대학교 소프트웨어특성화대학원 석사과정, 교신저자
*** 숭실대학교 소프트웨어특성화대학원 교수

1. 서 론

80년대 인공 지능분야의 확산과 함께 많은 투자와 연구들이 진행되었지만, 당시 진행되었던 프로젝트들의 기대에 미치지 않는 성과로 인해 인공 지능 연구의 침체기를 맞이했다. 그러나 기계 학습, 컴퓨터 비전 등 인공 지능의 하위 영역에 대한 집중적인 연구와 발전으로, 최근 인공 지능 분야에 대한 활발한 연구가 다시 진행되고 있다. 구글, 페이스북 등 글로벌 IT 기업들은 이미 이를 위한 연구소를 설립해 인공 지능 시스템 개발 및 연구에 집중하고 있고, 실제로 인공 지능을 도입한 뛰어난 성능의 서비스를 제공하고 있다. 또한, 사물인식과 음성인식 등 인공 지능분야 전반에 걸쳐 선구적 연구를 주도하고 있다. 이러한 연구결과로 Deep Mind 사는 2013년 인공 신경망(Artificial Neural Networks)과 Q-Learning을 활용한 Agent를 첫 선보였고, 2015년에 발표 된 논문을 통해 DQN(Deep Q-Networks)을 소개했다(Mnih et al., 2015). 2014년 본격적으로 AlphaGo가 개발되었고, 프로 바둑 기사들을 상대로 연이은 승리를 거두는 성과를 보였다. 심층 강화 학습(Deep Reinforcement Learning) 알고리즘 DQN이 발표 된 후, 이를 기반으로 한 여러 형태의 알고리즘이 제안되었다. 그러나 여전히 다양한 학습기법 및 알고리즘을 적용한 인공 지능 Agent의 학습 성능에 대한 평가연구는 부족한 실정이다.

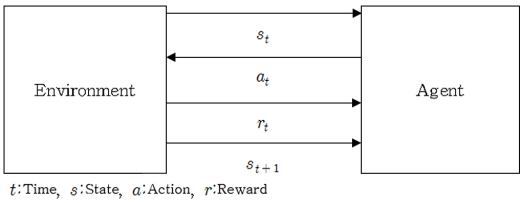
본 논문에서는 픽셀 데이터를 이용해 학습하는 인공 지능 Agent를 구현하고, 가치 기반 학습과 정책 기반 학습의 성능 차이를 분석한다. 이를 통해 Agent의 적절한 기반 학습 선택과 그 성능을 평가하는 연구에 기여할 수 있을 것으로 기대한다.

논문의 구성은 다음과 같다. 제 2장은 강화 학습과 DQN, PG(Policy Gradient) 학습 알고리즘에 대해 설명한다. 제 3장에서 심화 학습을 가능하게 하는 인공 신경망 및 CNN(Convolutional Neural Networks) 모형을 설명한다. 제 4장은 설계된 Agent의 데이터의 처리 방법과 인공 신경망 구성에 대해 기술한다. 제 5장은 학습 알고리즘에 따른 학습 구조 설계 및 동작 과정에 대해 다룬다. 제 6장에서 개발된 인공 지능 Agent의 성능측정실험을 진행하고, 그 결과를 분석한다. 마지막으로 제 7장에서 연구의 결론 및 향후 과제에 대하여 논한다.

2. 강화 학습

2.1 Markov Decision Process

현대 강화 학습 분야의 권위자 Richard S. Sutton 교수는, 강화 학습은 행동 심리학에서 영감을 얻어 발전된 기계 학습의 한 분야라고 설명한다(Sutton, 1998). On-line Learning의 특징을 갖고, 환경과 무수한 상호작용을 통해 학습하는 강화 학습의 기본적인 구조는 Markov Decision Process로 표현된다. <Figure 1>은 이를 나타낸 그림이다.



〈Figure 1〉 Markov Decision Process

환경을 탐색하는 의사결정 Agent는 현재의 상태(s_t)를 인식해 특정 행동(a_t)을 취한다. 이후 Agent는 다음 상태(s_{t+1})로 전이됨과 동시에 환경으로부터 보상(r_t)을 받는다. 이때 주어지는 보상 값은, Agent가 취한 행동이 학습목표를 달성하는데 긍정적일 경우 상점을 부여해 다음에 선택될 가능성을 높이는 반면, 부정적일 경우 벌점을 부여해 다음에 선택될 가능을 낮춘다. 연속되는 보상 $r_1, r_2, r_3, r_4 \cdots$이 존재 할 때, 누적 보상은 $r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots + \gamma^{k-1} r_k + \cdots$라고 할 수 있다. γ는 Discount Factor라고 하며 $0 \le \gamma < 1$의 범위를 갖는다. 이 조건을 통해 미래 보상의 중요도가 결정된다. γ가 1에 가까

울수록 미래 보상 누적 값이 증가하며, 이는 Agent가 미래 보상이 큰 방향으로 정책을 수립하도록 만든다.

2.2 Q-Learning

Q-Learning은 강화 학습의 대표적인 알고리즘이며 Off-Policy 형태의 가치 기반 학습 방법론이다. 이 방법론은 가치 함수를 기반으로, 최적의 정책에 대한 탐색을 진행한다. 환경을 모델링 할 수 없는 대다수의 현실 문제에서 Q-Learning은 행동 가치 함수를 사용해 최적의 정책을 찾아 낼 수 있다. 행동 가치 함수 $Q^{\pi}(s,a)$는 정책 π을 따라 상태 s에서 행동 a을 취함으로써 받게 될 보상치를 구해낸다. 최적의 행동 함수는 고려할 수 있는 모든 정책을 따라 상태 s에서 수행 한 행동 a에 대한 보상 중 최대치를 구해낸다. 이와 유사하게, 상태 가치 함수 $V^{\pi}(s)$가 존재한다. 정책 π을 따르는 상태 가치 함수는 정책 업데이트를 위해 환경 모델을 필요로 한다. 환경을 모델링 할 수 있는 경우 Dynamic Programming을 통해 최적의 정책을 구할 수 있다.

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha[r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})]$$

위의 식은 Q-Learning을 정리한 것으로, 상태 s에서 가장 높게 평가된 행동을 수행하고, 받은 보상 값을 학습 비율 α에 따라 Q-테이블을 업데이트함을 의미한다(Watkins, 1992).

2.3 Deep Q-Networks

가치 기반의 강화 학습 방법론에서 Q-Learning의 목표는 최적의 정책 함수를 찾는 것이다. Q-Learning은 가능한 모든 상태공간을 계산해, 최적의 행동 함수를 찾아낸다. 이때 넓은 상태공간의 문제로 학습수렴에 오랜 시간이 소요되거나 최악의 경우 학습이 불가능한 경우가 발생한다. 또한, Q-테이블을 업데이트 할 때 작은 변화에도 정책의 큰 변화가 발생하므로 학습 속도가 저하되거나 안정적인 학습이 어려운 경우가 많다. DQN은 Q-Learning에 인공 신경망을 도입한 개념으로, Experience Replay를 통해 학습의 안정성 및 성능의 향상을 가능하게 한다(Mnih et al., 2015). 경험 재생이란 학습 과정에서 발생하는 경험들을 $<s, a, r, s'>$의 형태로 Replay Memory에 저장하고, 무작위로 추출한 경험들을 사용해 신경망 가중치를 조정하는 것을 의미한다. 이것은 연속된 학습 데이터에 대한 상관관계를 줄여 학습이 Local Minimum에 수렴하지 않도록 유도하는 효과가 있다.

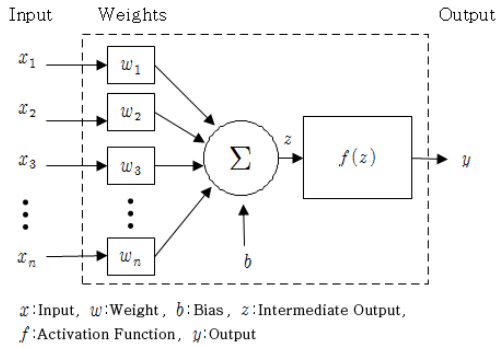
2.4 Actor-Critic Policy Gradient

정책 기반 강화 학습은 정책 자체를 평가해 신경망으로부터 최적 행동에 대한 확률을 구한다. 정책을 함수화한 학습은 점진적인 정책 개선을 통해 보다 안정적인 학습을 진행할 수 있다는 장점을 갖는다(Sutton, 1999). 그러나 여전히 높은 다양성 문제를 갖는다. Actor-Critic은 신경망을 두 개로 구성해 이와 같은 문제를 해결한다. Actor는 정책을 근사하고, 여기서 출력 된 정책은 Critic을 통해 평가된다. Critic은 행동 가치 함수를 근사해, 현재 정책이 얼마나 우수한지 평가한다.

3. 인공 신경망

3.1 인공 뉴런

인공 신경망은 이를 구성하는 인공 뉴런들의 네트워크라고 할 수 있다. 실제 뉴런의 정보처리 과정을 수학적으로 모방한 인공 뉴런은 시냅스의 결합 세기 조절을 통해 문제 해결 능력을 학습할 수 있다. 이렇게 구성된 인공 신경망은 학습을 통해 비선형 함수도 모델링 할 수 있다는 특징을 갖는다(Hsu, 1995). <Figure 2>는 인공 뉴런의 내부 구조를 나타낸 것이다.

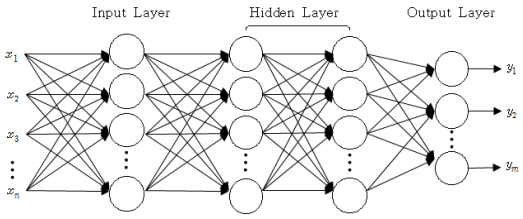


〈Figure 2〉 Internal Structure of An Artificial Neuron

각 뉴런은 입력 벡터와 가중치 벡터, Bias, 활성 함수(Activation Function)를 갖는다. 활성 함수는 대표적으로 Sigmoid나 ReLU를 이용한다. 0 또는 1의 단순 이진출력 함수를 사용하지 않는 이유는 Multi Layered Perceptron을 구성해 복잡한 문제 해결 능력을 학습 할 수 있기 때문이다.

3.2 Multi Layered Perceptron

인공 뉴런들로 구성된 인공 신경망의 가장 기본적인 형식을 Multi Layered Perceptron이라고 하며, <Figure 3>은 이를 나타낸다.



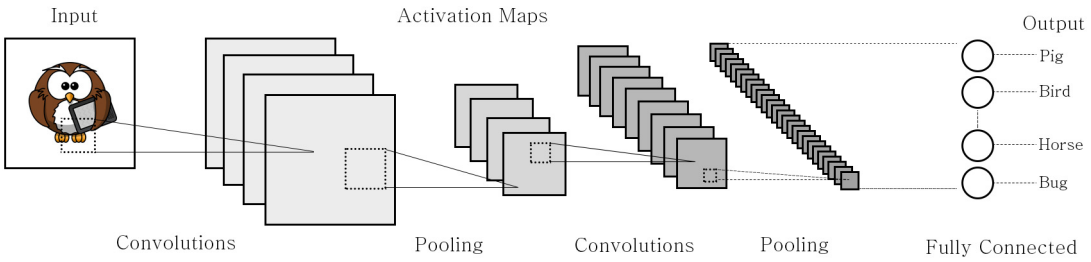
〈Figure 3〉 Structure of A Multi Layered Perceptron

네트워크는 Input Layer와 Hidden Layer, Output Layer로 구성되어 있고 각 레이어는 모두 연결되어 있다. 입력된 데이터는 Hidden Layer를 거쳐 하나의 Feature로 정의된다. Hidden Layers의 층이 깊어질수록 Feature Space도 복잡해진다. 충분한 학습이 진행되면, 형성된 Feature Space에 따라 입력 데이터를 적절히 분류할 수 있다(Bengio, 2007).

3.3 Convolutional Neural Networks

CNN은 동물의 시각정보 처리 과정을 모방한 인공 시신경 구조 모형으로 Convolution과 Pooling, Fully Connected라는 세 개의 핵심 요소를 가지고 있다(Fukushima, 1980). <Figure 4>는 CNN 구조를 나타낸다.

다음과 같이 구성된 신경망은 Input 이미지로부터 Activation Maps를 만드는 과정을 거친다. Convolution은 Input 이미지를 작게 나누는 것을 의미하고, Pooling은 Sampling을 통한 사이즈를 변경을 의미한다. Convolution시 특이점 추출에 사용하는 Filter는 Input 이미지보다 작은 크기를 갖으며 일정 간격(Stride)으로 이미지에 대한 Feature 추출을 진행한다. 이 과정에서 사용되는 Filter의 수는 해당 Convolution에서 사용되는 뉴런수를 의미한다. 최종 출력은 Fully Connected Layer에 연결되어 추출된 정보를 기반으로 Input 이미지를 분류한다.



〈Figure 4〉 Structure of CNN

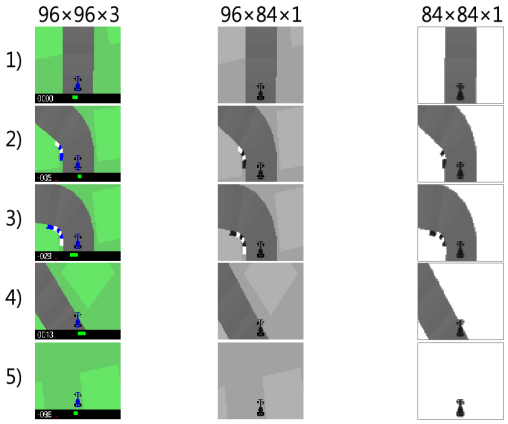
4. Agent 설계

4.1 학습 환경

Agent는 차량 조작 방법에 대해 무지한 상태에서 핸들, 가속, 브레이크를 적절히 조작해 트랙을 완주해야 한다. 핸들의 조작범위는 -1.0에서 +1.0 사이의 실수 구간이며 -1.0에 가까울수록 크게 핸들을 왼쪽으로 꺾으며, +1.0에 가까울수록 오른쪽으로 꺾는다. 가속과 브레이크는 0부터 +1.0의 가동범위를 갖으며 수치가 +1.0에 가까울수록 입력에 크게 반응한다. 회전구간에서 적절한 조작이 이루어지지 않으면, 차량은 균형을 잃고 미끄러진다. 매 프레임마다 -0.1의 보상을 받고 트랙을 따라 바르게 전진할 때만 +3의 보상을 받는다. 트랙을 크게 벗어나면 -100의 보상을 받고 총 -100 이하의 누적보상이 발생 했을 때 Episode를 종료하고 출발지점으로 돌아가 새로운 Episode를 시작한다.

4.2 데이터 전처리

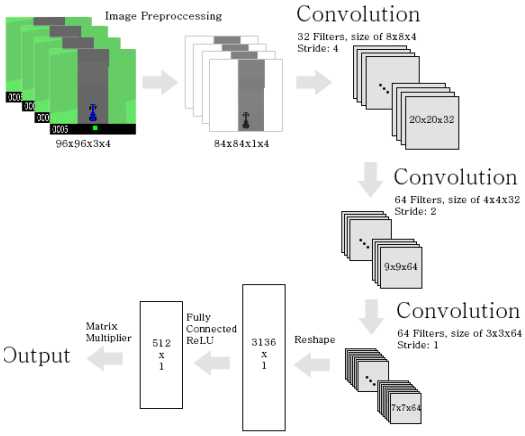
신경망은 가상 물리환경에서 관측한 화면 정보를 학습에 이용한다. 환경에서 관측한 화면은 96×96 픽셀 이미지고 RGB 채널을 고려하면 96×96×3의 데이터가 된다. <Figure 5>는 이 데이터를 효율적으로 사용하기 위한 전처리 과정을 나타낸다. 1)부터 5)는 상황에 따라 다른 입력화면을 나타낸다. 학습에 불필요한 부분을 Crop하고 Gray Scale로 색상을 변경하는 과정을 거친다. 최종적으로 차량 및 주행코스 이외 영역에 대한 색상 대체를 통해 보다 명확한 84×84×1의 학습용 흑백 이미지를 완성한다. 전처리를 통한 이점으로 27,648 Bytes의 원본 이미지의 크기는 $\frac{1}{4}$수준으로 줄어드는 효과를 나타낸다.



〈Figure 5〉 Image Preprocessing

4.3 네트워크 구성

CNN을 구성해 연속적으로 관찰되는 화면에 대한 정보를 파악할 수 있도록 한다. 관측 순서에 따라 전처리 과정을 거쳐 출력된 4개의 84×84×1의 이미지를 학습의 기본 데이터 셋으로 사용한다. 한 프레임의 이미지를 학습에 사용할 경우, 차량의 방향 또는 코스의 변화와 같은 정보는 추출 할 수 있지만, 차량의 주행 속도나 이동 경로와 같은 시간성을 포함하는 정보는 얻을 수 없다. 그러므로 연속된 4프레임의 이미지를 학습의 기본 단위로 구성해, Agent가 스스로 환경변화를 인지하도록 유도한다. <Figure 6>은 이 네트워크를 나타낸 것이다.



〈Figure 6〉 Configuration of the Designed Network

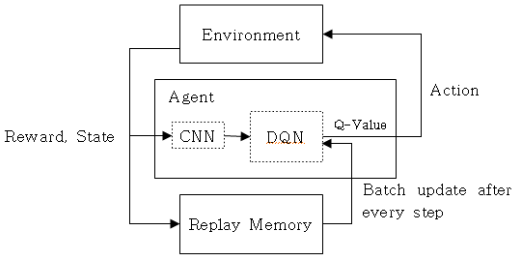
3번의 Convolution을 통해 7×7×64의 이미지로 분류된 입력 데이터는 Fully Connected Layer를 거치기 전 3136×1의 모양으로 변환된다. 512개의

인공 뉴런을 사용하는 Fully Connected Layer는 이를 입력으로 받아 ReLU 활성 함수를 통해 결과를 출력한다.

5. 학습 프로세스

5.1 DQN을 이용한 학습

DQN을 이용한 학습구조는 <Figure 7>과 같이 나타낼 수 있다. Agent는 DQN을 거쳐 출력된 Q-Value를 기반으로, 미래보상을 최대로 하는 행동을 선택한다.



〈Figure 7〉 Architecture of Learning Using DQN

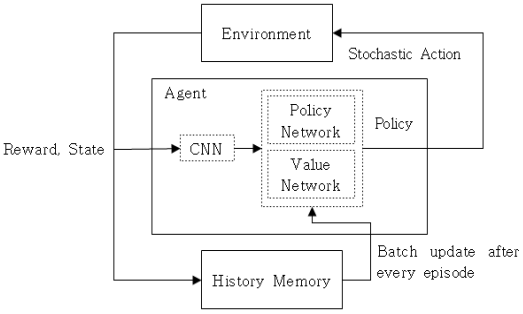
Agent는 환경으로부터 수행한 행동에 따른 보상과 다음 전이 상태를 관측한다. 이것이 하나의 학습 데이터 셋이 되어 Replay Memory에 최대 20,000개까지 저장된다. $<s, a, r, s'>$로 구성된 이 학습 데이터 셋은 학습 초기 20,000회 동안 Replay Memory에 채워지는 과정을 거치고 이후 Replay Memory가 가득차면 가장 오래된 학습 데이터 셋을 빼고, 새로운 학습 데이터를 추가한다. 매 행동선택 Step마다 그 중 임의로 학습 데이터 셋 16개를 추출해 DQN 내부의 가중치조정에 사용한다. <Table 1>은 DQN 학습에서 설정된 Parameters를 나타낸다. 높은 다양성 문제를 보완하기 위해 DQN은 ε-Greedy한 행동 선택을 한다. ε은 무작위 행동 비율을 의미하고, 학습 초기 무작위 행동 비율을 높게 설정해 다양한 경험을 할 수 있도록 유도한다. 관측 회수에 따라 무작위 행동 비율을 줄이고 학습 성과에 따른 일관적인 행동을 하도록 한다. Discount Factor는 1에 가깝게 두어 미래 보상이 큰 쪽으로 학습이 수렴하도록 한다.

〈Table 1〉 Parameters Configuration of the DQN

Parameter	Value
Learning rate(α)	0.0001%
Batch size	16 steps
Discount factor(γ)	99%
Early random action rate(ε)	30%
Final random action rate(ε)	0.01%
Number of observation	20,000 times
Replay memory size	20,000 units

5.2 PG를 이용한 학습

PG를 이용한 학습은 Q-Value를 출력하는 DQN과 달리 신경망에서 Stochastic한 Policy를 출력한다. <Figure 8>은 이를 나타낸다.



〈Figure 8〉 Architecture of Learning Using PG

매 행동선택 Step마다 관측된 정보는 History Memory에 저장되고, Episode를 마칠 때마다 신경망 가중치 조정에 사용된다. Policy Network를 통해 출력된 정책은 핸들, 가속, 브레이크에 대한 우선순위 선택 확률을 포함한다. Agent는 0과 1사이에서 구한 임의의 확률과 정책을 따른 행동 확률을 비교해 Stochastic Action을 수행한다. Value Network는 정책에 대한 평가치를 출력하고, 정책이 얼마나 유용한지 판단한다. <Table 2>는 PG 학습의 Parameters 설정을 나타낸다.

〈Table 2〉 Parameters Configuration of the PG

Parameter	값
Learning rate of policy network(α)	0.0001%
Learning rate of value network(α)	0.0001%
Batch size	1 episode
Discount factor(γ)	0.99

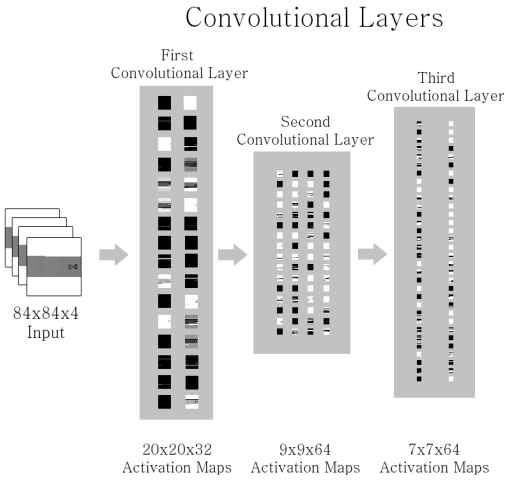
Policy Network과 Value Network는 모두 0.0001%의 학습 비율로 학습한다. 적절한 신경망 가중치 조정을 위해 Agent는 각 Step마다 계산된 미래 보상 벡터와 Value Network의 출력 벡터를 구하고, 그 차이를 Value Network의 손실 함수로 정의한다. Value Network는 이 손실 함수를 최소화하는 방향으로 신경망 가중치를 조정한다. Policy Network도 가중치 조정을 위해 매 Step 계산된 미래 보상과 Value Network의 값의 차이를 이용한다. 이 두 값의 차를 이용해 정책의 진행 방향과 변화 세기를 결정한다.

6. 실험 및 결과 분석

6.1 CNN 학습 결과

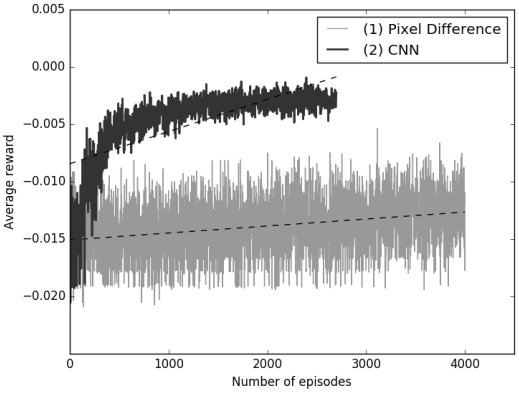
입력된 이미지는 3번의 Convolution을 거치면서 각각 20×20×32, 9×9×64, 7×7×64의 Activation Maps를 출력한다. <Figure 9>는 구성한 CNN이 입력되는 학습 이미지에 따라 다르게 반응하는 Activation Maps을 나타낸다. 구현한 CNN은 3개의 Hidden Layer를 갖는다. 각 Layer에 있는 인공 뉴런 수는 Convolution시 사용한 Filter의 수와 같다. 학습을 통해 뉴런의 가중치가 적절히 조절되면, 입력으로 들어온 이미지에 대해 각 뉴런들은 다르게 반응한다. 특정 뉴런들은 차량의 위치에 반응하는 반면, 몇 뉴런들은 주행 코스에 반응하거나 환경 변화를 감지하지 않는 등 각각 다른 반응 양상을 보인다.

CNN 기반의 강화 학습 성능 평가를 위해 CNN을 이용한 학습과 프레임 간 단순 픽셀 이미지 대조를 기반으로 한 학습의 학습 성능 비교했다.

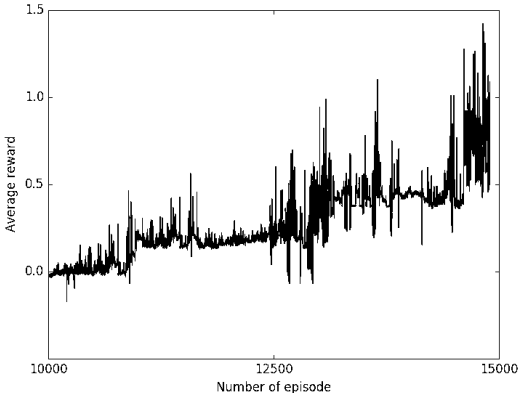


〈Figure 9〉 Activation Maps of the CNN

<Figure 10>의 가로축은 학습한 Episode의 수를 나타내고, 세로축은 한 Episode에서 발생한 보상의 평균을 의미한다. (1)은 이미지 프레임 간 픽셀 차이를 사용한 이미지 대조 기반 학습의 성능을 나타낸다. (2)는 CNN을 사용한 학습의 성능을 나타낸다. 두 영역에 표시한 점선은 각 학습 결과에 Linear Regression을 적용해 두 학습의 성능을 선형적으로 나타낸 모습이다. 강화 학습을 위해 PG 학습 알고리즘을 사용했고 입력 이미지를 처리하는 부분 외의 나머지 요소들은 모두 같게 설정했다. 학습 환경으로는 선택 행동의 수가 작고 행동 범위에 제한이 있어, 단순 이미지 대조만으로도 비교적 학습이 용이한 환경을 선택했다. 학습 환경은 2D 환경에서 진행되는 일대일 대결 형식의 탁구 게임으로, Agent는 일정 수준의 성능을 갖춘 인공 지능 플레이어와 대결을 통해 승점을 획득해야 한다. 상대의 공을 받지 못하면 -1의 감점을 받고, 반대로 득점에 성공하면 +1의 승점 얻는다. 랠리가 이어지면 0의 보상을 받는다. <Figure 10>은 위와 같은 환경에서 4,000여 개의 Episode를 학습한 결과이다. 이미지 대조 기반 학습은 학습 초기부터 불안정하고 편차가 큰 학습 양상을 보인 반면, CNN을 이용한 학습은 전반적으로 안정적



〈Figure 10〉 Performance of the CNN and Pixel Difference-Based Learning



〈Figure 11〉 Performance of Learning Using DQN

인 학습의 모습을 보였다. 뿐만 아니라 1,000개의 Episode를 학습하기도 전에 이미 4,000여 개의 Episode를 학습한 이미지 대조 기반 학습의 성능을 뛰어넘는 결과를 보였다. 전반적으로 CNN을 이용한 학습의 장점이 두드러졌지만 단점도 존재했다. 증가한 데이터 처리 양으로 인해 이미지 대조 기반 학습보다 느린 연산 속도를 보였다. CNN을 사용한 학습은 학습에 필요한 기본 데이터의 크기가 84×84×4로 한 Step마다 처리해야 할 데이터가 28,224 Bytes인 반면, 이미지 대조를 이용한 학습은 두 프레임간의 픽셀 차이를 이용하므로 84×84×1의 데이터가 필요하다. 이는 Step당 처리 데이터 량이 4배 증가함을 의미한다.

6.2 Agent 학습 결과

DQN과 PG 알고리즘을 적용한 Agent의 학습 성능을 평가하기 위해 매 Episode마다 발생한 보상의 평균을 그래프에 나타냈다. 그래프에 나타난 학습 곡선 및 변화 양상을 통해 각 학습의 특징을 분석할 수 있었다.

DQN 알고리즘을 적용한 학습 Agent는 근소하지만 점진적으로 학습 성능을 높이는 양상을 보였다. 그러나 Episode 간 평균 보상 값의 편차가 급격하게 커지는 구간이 빈번히 등장했고 이는 그래프 상에서도 큰 진폭을 형성했다. DQN 학습이 이러한 양상을 보이는 이유는 Q-Value의 변동이 급격한 정책 변화를 유도했기 때문이다. 학습의 속도 측면에서 DQN 학습을 분석해 보면, 매 Step마다 진행되는 Random Batch Update로 인해 단위 Step당 계산해야 할 연산량이 크게 증가해 전체적인 학습 진행 속도를 느리게 만든다. 뿐만 아니라, Batch 크기가 증가 할수록 학습 속도는 더욱 저하되는 현상이 나타났다. <Table 3>은 간단한 실험을 통해 나타낸 Batch 크기와 학습 속도에 대한 표이다.

〈Table 3〉 Operation Speed of Learning Using DQN by Batch Size

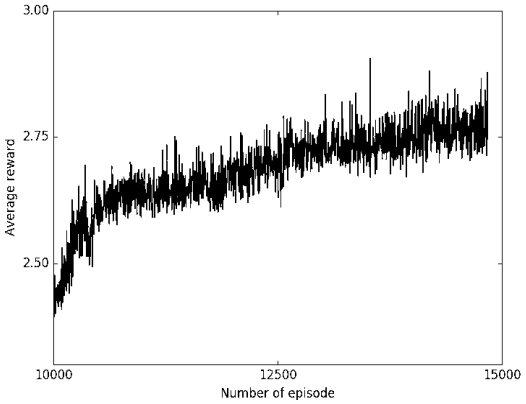
Unit : Step/Sec.

Type of step / Batch size	Training Step
16 steps	0.09600701
32 steps	0.16169815
64 steps	0.29857897

Batch 크기에 따른 연산속도비교를 위해 서로 다른 Batch 크기를 갖는 학습에서 1,000 Steps 연산에 소요되는 시간을 측정했다. 표의 값은 1 Step 진행에 필요한 평균 소요시간을 의미한다. Batch 크기가 증가 할수록 신경망의 가중치를 조정하는 Training Step에서 소요되는 시간이 크게 증가하는 것을 확인 할 수 있다. DQN 학습에서 Replay Memory를 이용한 Random Batch Update는 학습

이 Local Minimum에 수렴하지 않도록 유도하는 역할을 한다. 이를 위해서는 적당한 규모의 Batch 크기가 설정되어야 한다. 이 크기가 작을 경우 빠른 학습은 가능하나, 의도했던 학습 효과는 떨어질 수 있다. 그러므로 DQN을 이용한 학습은 Batch 크기와 학습 속도간의 적절한 조정이 이루질 필요가 있다고 판단된다.

PG를 이용한 학습 역시 학습 성능 그래프를 통해 학습의 특징을 분석할 수 있다. <Figure 12>는 PG를 이용한 학습 그래프를 나타낸다.



〈Figure 12〉 Performance of Learning Using PG

PG 기반 학습은 DQN을 이용한 학습에 비해 비교적 안정정인 학습 곡선을 나타낸다. 또한, 각 Episode마다 획득한 평균 보상의 편차가 비교적 낮고, 이에 따른 학습 성과가 안정적으로 증가하는 양상을 보인다. 획득한 평균 보상 역시 DQN을 이용한 학습보다 더 높은 수치를 기록했다, DQN이 15,000개의 Episode를 학습함으로써 이루어낸 학습 성과를 PG를 이용한 학습은 10,000개의 Episode를 채 학습하기도 전에 이룰 수 있었다. PG 학습의 신경망 조정은 매 Episode가 종료 될 때 진행된다. 각 학습기법에 따른 학습 속도 비교를 위해 Episode에서 발생한 행동선택 Step 단위 PG 학습의 평균 연산 속도를 계산했다. 비교가 되는 대상은 16개의 Batch 크기를 갖는 DQN 학습이다. <Table 4>는 이를 나타낸다.

〈Table 4〉 Comparison of Operation Speed by Learning Algorithm

Unit : Step/Sec.

Learning algorithm / Number of episode	PG	DQN
10 episodes	0.00235855	0.09583281
100 episodes	0.00234981	0.09609522
1,000 episodes	0.00236903	0.09361807

두 학습 알고리즘 모두 진행된 학습 Episode 수와는 관계없이 일정한 수준의 학습 속도를 나타냈다. 결과적으로 PG 학습은 16개의 Batch 크기를 갖는 DQN 학습에 비해 Step 단위 평균 연산 소요 시간이 약 40배 빠름을 확인할 수 있다.

7. 결 론

본 연구에서는 가상의 물리 환경으로부터 관측한 픽셀 데이터를 이용해, 스스로 학습하는 인공지능 Agent를 구현했다. 학습을 통한 올바른 차량 주행 능력 습득을 목표로, 가치 기반 학습과 정책 기반 학습을 각각 진행했다. 각 학습 알고리즘에 따른 학습 Parameters 조정과 학습 구조의 변경을 통한 여러 실험을 진행했다. 이를 통해, 각 기반 학습에 따른 학습 성능 및 특징을 분석하고, 픽셀 데이터를 이용한 학습에서 CNN 모형이 전체 학습 성능에 미치는 영향을 측정하는 연구를 수행했다.

실험을 통해 학습 Parameters 조정이 인공 지능 Agent의 학습 성능을 개선할 수 있음을 확인했다. 특히, 신경망 가중치 조정에 사용되는 Batch 크기는 학습 속도에 직접적인 영향을 미치므로 적절한 조정이 필요했다. 픽셀 데이터를 사용하는 강화 학습 Agent 설계에 CNN 모형을 적용함으로써 얻은 결론은 다음과 같다. CNN 모형을 사용하는 학습은 이미지 대조 기반 학습보다 전반적인 학습 성능 및 수렴 속도에서 우위를 보였다. 그러나 단순 이미지 대조 기법보다 더 큰 학습 데이터를 사

용하므로 많은 연산 량을 요구했다. 학습 방법론적 차이와 그 성능에 대한 결론은 다음과 같다. 정책 기반 학습인 PG는 가치 기반 학습 DQN에 비해 더 우월한 학습 결과를 나타냈다. 두 개의 신경망 사용으로 인해 연산이 더 복잡해 졌음에도 불구하고 학습 속도에서도 우위를 보였다. 가치 기반 학습은 반드시 하나의 최적의 행동으로 수렴하는 반면, PG를 이용한 학습은 Stochastic한 정책을 배울 수 있었다. 높은 다양성으로 인한 Local Minimum 문제를, 이를 보완하는 Actor-Critic 형태의 학습을 통해 학습 안정성 및 수렴 속도 개선을 가능하게 했다.

본 연구에서는 강화 학습에서 대표적으로 사용되는 학습 알고리즘인 DQN과 PG를 이용했지만, 최근 활발한 연구를 통해 개선되고 있는 다양한 강화 학습 알고리즘을 적용한다면 보다 뛰어난 성능을 발휘하는 인공 지능 Agent를 구현할 수 있을 것이다. 또한 단일 머신이 아닌 분산 클러스터 환경이나 Multi-Agent를 이용한 환경 구성을 통해 보다 빠르게 학습하고 고성능을 발휘하는 인공 지능 시스템 연구가 가능할 것으로 기대한다.

References

Bengio, Y., P. Lamblin, D. Popovici, and H. Larochelle, “Greedy Layer-wise Training of Deep Networks”, *Advances in Neural Information Processing Systems*, Vol.19, 2007, 153.

Fukushima, K., “Neocognitron : A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position”, *Biological Cybernetics,* Vol.34, No.4, 1980, 193-202.

Hsu, K., H.V. Gupta, and S. Sorooshian, “Artificial Neural Network Modeling of the Rainfall Runoff Process”, *Water Resources Research*, Vol.31, No.10, 1995, 2517-2530.

Mnih, V., K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-level Control Through Deep Reinforcement Learning”, *Nature,* Vol. 518, No.7540, 2015, 529-533.

Sutton, R.S. and A.G. Barto, *Reinforcement Learning : An Introduction*, MIT press, Cambridge, 1998.

Sutton, R.S., D. McAllester, S. Singh, and Y. Mansour, “Policy Gradient Methods for Reinforcement Learning with Function Approximation”, *NIPS,* Vol.99, 1999, 1057-1063.

Watkins, C.J.C.H. and P. Dayan, “Q-learning”, *Machine Learning*, Vol.8, No.3, 1992, 279-292, doi : 10.1007/BF00992698(Downloaded April 2, 2016).

◈ About the Authors ◈

Saemaro Moon (saemaromoon@gmail.com)

Saemaro Moon is currently in a master's course in Software Engineering from the Graduate School of Software, Soongsil University. He received a B.S. degree in Computer Science and Engineering from Soongsil University in 2015. His current research interests include artificial intelligence, big data, IoT and etc.

Yonglak Choi (ylchoi58@ssu.ac.kr)

Yonglak Choi is currently working as a professor for the Graduate School of Software, Soongsil University in Seoul, Korea. He received a Ph.D. from Soongsil University. His research and main interest areas include databases, data modeling, big data, software engineering, information strategy planning, and open source architecture. And, he developed careers that ERP, financial information systems, marketing system, insurance system, etc. He has served as an advisor to the Ministry of National Defense, Ministry of Strategy and Finance, and The Board of Audit and Inspection of Korea.