The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 19, No. 2, pp.135-141, Apr. 30, 2019. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2019.19.2.135
JIIBC 2019-2-19

2D 슈팅 게임 학습 에이전트의 성능 향상을 위한 딥러닝 활성화 함수 비교 분석

Comparison of Deep Learning Activation Functions for Performance Improvement of a 2D Shooting Game Learning Agent

이동철*, 박병주**

Dongcheul Lee*, Byungjoo Park**

요 약 최근 강화 학습을 통해 게임을 학습하는 인공지능 에이전트를 만드는 연구가 활발히 진행되고 있다. 게임을 에이전트에게 학습 시킬 때 어떠한 딥러닝 활성화 함수를 사용하는지에 따라 그 학습 성능이 달라진다. 본 논문은 2D 슈팅 게임 환경에서 에이전트가 강화 학습을 통해 게임을 학습할 경우 어떤 활성화 함수가 최적의 결과를 얻는지를 비교 평가 한다. 이를 위해 비교 평가에서 사용할 메트릭을 정의하고 각 활성화 함수에 따른 메트릭 값을 학습 시간에 따라 그래프로 나타내었다. 그 결과 ELU (Exponential Linear Unit) 활성화 함수에 1.0으로 파라미터 값을 설정할 경우 게임의 보상 값이 다른 활성화 함수보다 평균적으로 높은 것을 알 수 있었고, 가장 낮은 보상 값을 가졌던 활성화 함수와의 차이는 23.6%였다.

Abstract Recently, there has been active researches about building an artificial intelligence agent that can learn how to play a game by using re-enforcement learning. The performance of the learning can be diverse according to what kinds of deep learning activation functions they used when they train the agent. This paper compares the activation functions when we train our agent for learning how to play a 2D shooting game by using re-enforcement learning. We defined performance metrics to analyze the results and plotted them along a training time. As a result, we found ELU (Exponential Linear Unit) with a parameter 1.0 achieved best rewards than other activation functions. There was 23.6% gap between the best activation function and the worst activation function.

Keywords : Game, Activation function, Re-enforcement Learning

Ⅰ. 서 론

강화 학습(Re-enforcement learning)으로 비디오 게임을 딥러닝(Deep learning) 에이전트에게 학습 시키는 것은 현재 매우 활발히 연구되는 주제 중의 하나이다. 강화 학습의 목표는 주어진 환경에서 특정 행동을 취하는 정책을 세우고, 그 정책을 따라 행동했을 때 보상을 극대화할 수 있도록 하는 것이다. 이를 위해 fixed Q-targets, DQN (Deep Q-Learning), double DQN, DDQN (Dueling DQN), PER (Prioritized Experience Replay), A3C (Asynchronous Advantage Actor-Critic) 등의 강화 학습 알고리즘이 제시되었으며 이중 A3C가 가장 성능이 뛰어난 것으로 알려져 있다[1]. 비디오 게임을 학습하기 위해서는 게임의 특징 (Feature)을 연구자가 직접 지정해 주는 방법도 있었으나 최근에는 게임 화면 픽셀 자체를 신경망의 입력으로 사용하는 방법이 주된 방법이다.

*종신회원, 한남대학교 멀티미디어공학과
**종신회원, 한남대학교 멀티미디어공학과(교신저자)
접수일자 2019년 1월 31일, 수정완료 2019년 3월 1일
게재확정일자 2019년 4월 5일

Received: 31 January, 2019 / Revised: 1 March, 2019 / Accepted: 5 April, 2019
**Corresponding Author: bjpark@hnu.kr
Dept. of Multimedia Engineering, Hannam University, Korea

신경망에서 이미지 처리를 위해서는 주로 CNN (Convolutional Neural Network)을 사용한다[2]. 또한 게임에 따라 특정 행동을 취한 보상이 즉시 주어지는 것이 아니라, 일정 시간이 흘러야 주어지는 경우가 있다. 예를 들어 슈팅 게임의 경우 총알을 발사하면 일정 시간이 흘러야 적을 맞출 수 있고 이때 점수가 올라가는 것과 같다. 이러한 게임을 학습하기 위해서는 이전 상태를 기억했다가 추후 어떤 행동을 하는 것이 가장 좋을지 결정할 때 사용해야 하는데 신경망에서는 이를 위해 주로 RNN (Recurrent Neural Network)이나 LSTM (Long Short Term Memory)을 사용한다[3, 4].

이와 같이 여러 종류의 신경망을 함께 사용할 경우 신경망 내 여러 층이 존재하게 되는데 이러한 층을 결합하기 위해 사용되는 것이 활성화 함수이다[5]. 활성화 함수도 다양한 종류가 존재하는데 어떤 활성화 함수를 사용하는지에 따라 학습 에이전트의 성능도 달라진다. 그러나 모든 상황에서 뛰어난 결과를 보여주는 활성화 함수는 아직까지 정의되지 않았으며 각 상황에 따라 그 결과가 다르게 나타난다. 본 논문은 2D 슈팅게임을 학습하기 위해 A3C와 LSTM을 기반으로 한 딥러닝 에이전트를 개발하고 어떤 활성화 함수를 사용하는 것이 학습에 유리한지 성능을 비교 평가하고자 한다.

본 논문은 다음과 같은 구성이다. 2장에서는 본 논문을 구현하기 위한 환경인 OpenAI Gym과 활성화 함수 종류에 대해 알아본다. 3장에서는 활성화 함수를 비교 평가하기 위한 방법에 대해 정의하고 4장에서는 비교 평가 결과를 제시하고 결과를 분석한다. 5장에서는 결과를 종합하고 결론을 제시한다.

II. 관련 연구

1. OpenAI Gym

Gym은 OpenAI에서 강화 학습 알고리즘을 개발하고 비교하기 위해 만든 툴킷이다[6]. 이 툴킷을 사용하면 각종 아타리 (Atari) 게임 등을 플레이 할 수 있는 학습 에이전트를 만드는데 도움을 준다. 플레이할 수 있는 게임의 종류는 매우 다양하며, 게임 종류에 상관없이 학습 에이전트에게 표준화된 게임 환경을 제공한다. 게임 환경에서 학습 에이전트가 특정 액션을 취하면 Gym은 프레임 단위로 해당 액션에 대한 결과를 에이전트에게 준다.

표 1. 활성화 함수
Table 1. Activation functions

Name	Equation
Sigmoid	$f(x)=\frac{1}{1+e^{-x}}$
Tanh	$f(x)=\frac{2}{1+e^{-2x}}-1$
ReLU	$f(x)=\begin{cases}0\text{ , for } x<0\\ x\text{ , for } x\geq 0\end{cases}$
Leaky ReLU	$f(x)=\begin{cases}0.01x\text{ , for } x<0\\ x\text{ , for } x\geq 0\end{cases}$
ELU	$f(\alpha,x)=\begin{cases}\alpha(e^{x}-1)\text{ , for } x<0\\ x\text{ , for } x\geq 0\end{cases}$
SELU	$f(\alpha,x)=\lambda\begin{cases}\alpha(e^{x}-1)\text{ , for } x<0\\ x\text{ , for } x\geq 0\end{cases}$ with $\lambda=1.0507$ and $\alpha=1.67326$
PReLU	$f(\alpha,x)=\begin{cases}\alpha x\text{ , for } x<0\\ x\text{ , for } x\geq 0\end{cases}$

해당 액션에 대한 결과는 다음 프레임의 게임 상태, 보상 값 (Reward), 남은 생명 수, 게임 종료 여부 등이다. 학습 에이전트의 목적은 일반적으로 보상 값을 짧은 시간 내에 높이는 것이다. 초기 게임 상태는 임의로 설정되며 게임이 종료될 때 까지 에이전트는 해당 게임을 플레이할 수 있다. 연구자들은 이를 이용하여 표준화된 알고리즘을 작성할 수 있으며 이는 다른 연구자와 알고리즘의 성능을 비교할 수 있는 토대가 된다.

2. 활성화 함수

강화 학습을 위해서는 신경망을 사용해야 한다. 신경망 뉴런 (Neuron)들의 결합으로 구성되고 뉴런은 가중치 (Weight, w), 역치 (Bias, b), 활성화 함수로 구성된다. 신경망에 입력 값이 들어오면 입력 값에 대하여 뉴런의 가중치와 역치는 선형 변환 (Linear Transformation)을 담당한다. 활성화함수는 비선형 변환 (Non-linear Transformation)을 담당하는데 뉴런이 활성화 되어야 할지 아니면 해당 입력을 무시해야 할지를 결정하며 다음과 같이 정의할 수 있다.

$$f(x)=Activation(b+\sum_{i=1}^{n}x_i w_i) \quad (1)$$

활성화 함수의 결과는 다음 뉴런의 입력으로 사용된다. 활성화 함수가 사용되지 않는 신경망은 단순히 선형 회기 모델 (Linear Regression Model)이 되기 때문에 자연어 처리나 이미지 분류와 같은 복잡한 문제를 해결할 수 없다. 또한 가중치와 역치를 학습시킬 때 경사도

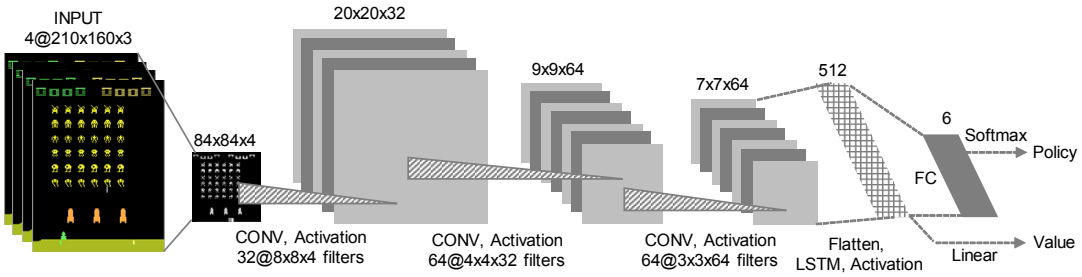

그림 1. 강화 학습 에이전트가 2D 슈팅 게임 방법을 학습하는데 사용한 딥러닝 모델
Fig. 1. Illustration of deep learning model used by the reinforcement learning agent to learn how to play a 2D shooting game

(Gradient)를 제공하여 역전파 (Back-propagation)를 이용할 수 있도록 해준다[7].

활성화 함수의 종류는 표 1과 같다. 초기의 활성화 함수는 시그모이드 (Sigmoid) 함수나 이를 변형한 Tanh 함수가 사용되었다. 두 함수는 비선형이고 모든 구간에서 미분 가능하며, 따라서 역전파를 사용할 수 있게 된다. 그러나 두 함수는 ±3 구간 외에서 기울기가 매우 작게 되어 역전파시 학습이 거의 이루어지지 않는 경사 사라짐 문제 (Vanishing gradient problem)가 있다[8]. 이를 해결하기 위한 함수가 ReLU (Rectified Linear Unit)이다. ReLU는 단순하며 현재 가장 많이 사용되는 활성화 함수이다. 이 함수는 비선형이며 입력이 음수일 경우 뉴런을 활성화하지 않게 하여 신경망을 단순하고 계산이 쉽게 해 준다. 그러나 입력 값이 계속해서 음수로 올 경우 미분 결과가 항상 0이므로 죽은 뉴런 (Dead neuron) 현상을 초래하여 학습이 이루어지지 않게 된다. 이러한 단점을 해결하기 위해 Leaky ReLU가 등장하였다. 이 함수는 입력 값이 음수일 경우 작은 기울기를 가지는 직선으로 결과가 나온다. 따라서 미분 결과가 0이 아니게 되어 죽은 뉴런 현상을 방지할 수 있다. 또 다른 해결 방법으로 제시된 함수가 ELU (Exponential Linear Unit)이다. 이 함수도 입력 값이 음수일 경우 작은 기울기를 가지는 곡선으로 결과가 나오며 α 값에 따라서 곡선의 크기가 결정된다. ELU를 조금 더 확장 시킨 함수가 SELU (Scaled Exponential Linear Unit)이다. 이 함수는 파라미터로 λ를 추가하여 음수 부분뿐만 아니라 양수 부분의 크기도 결정할 수 있도록 한다. PReLU (Parametric Rectified Linear Unit)도 Leaky ReLU와 비슷하게 음수 부분만 α 값에 따라 직선의 크기가 결정되는 함수이다.

Ⅲ. 분석 방법

본 논문은 2장에서 언급된 활성화 함수가 강화 학습에 끼치는 영향을 분석하기 위하여 OpenAI Gym 환경에서 2D 슈팅 게임을 학습하는 강화 학습 에이전트를 만들었다. 본 에이전트는 A3C와 LSTM을 결합한 강화 학습 알고리즘을 사용하며 그림 1과 같은 모델을 기반으로 한다. 최초 입력 값의 복잡도를 줄이기 위하여 컬러 이미지를 회색조로 변환하고 가장자리 부분을 잘라내었다. 또한 4개의 연속된 프레임을 합쳐서 동시에 처리할 수 있도록 하였다. 모델은 3개의 CNN과 1개의 FC (fully-connected), 1개의 LSTM으로 구성된다. CNN에 사용된 필터는 각각 32개, 64개, 64개이며, 필터 크기는 각각 8x8, 4x4, 3x3 이다. LSTM에 사용된 셀은 512개이다. 활성화 함수는 그림 2와 같이 ReLU, Leaky ReLU, ELU, SELU, PReLU가 사용되었으며 각각의 은닉층에 사용되었다.

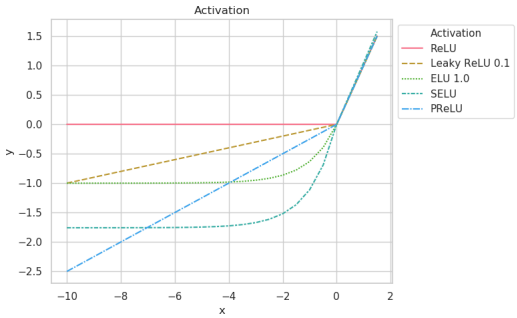

그림 2. 실험에서 사용된 활성화 함수
Fig. 2. Activation functions used in the experiment

표 2. 활성화 함수 별 평균 및 최고 보상 값
Table 2. Mean and max reward for each activation function

Activation Function	Mean	Max
ELU 0.5	321.0	579.5
ELU 1.0	393.8	673.0
ELU 1.5	351.7	633.5
Leaky Relu 0.01	300.8	559.5
Leaky Relu 0.1	325.1	574.5
Leaky Relu 0.3	304.1	536.5
PRelu	331.5	561.0
Relu	359.6	692.5
SELU	350.1	603.0

표 3. 활성화 함수 별 평균 및 최고 $V(s_t)$ 값
Table 3. Mean and max $V(s_t)$ for each activation function

Activation Function	Mean	Max
ELU 0.5	2.709	5.550
ELU 1.0	3.097	6.054
ELU 1.5	2.919	6.384
Leaky Relu 0.01	2.618	5.607
Leaky Relu 0.1	2.672	5.716
Leaky Relu 0.3	2.837	4.302
PRelu	2.494	4.680
Relu	2.513	5.974
SELU	2.873	6.007

표 4. 활성화 함수 별 평균 및 최고 $L(s_t)$ 값
Table 4. Mean and max $L(s_t)$ for each activation function

Activation Function	Mean	Max
ELU 0.5	0.8613	5.2269
ELU 1.0	0.9131	7.4853
ELU 1.5	0.8388	5.9464
Leaky Relu 0.01	0.7966	5.0446
Leaky Relu 0.1	0.8340	4.9982
Leaky Relu 0.3	0.9421	4.5513
PRelu	0.7528	4.5854
Relu	0.7660	5.3489
SELU	0.8429	5.6105

출력층에서는 정책함수 $\pi(s_t)$와 가치함수 $V(s_t)$가 결과 값으로 나온다. 특정 행동을 하였을 때 얻을 수 있는 이득을 추정하는 이득 (Advantage)은 다음과 같이 정의하였으며 할인된 보상 (Discounted reward) R을 사용하였다.

$$A(S_t) = R - V(S_t) \tag{2}$$

신경망을 업데이트하기 위해 사용하는 손실 함수 $L(s_t)$는 가치 손실 함수 $L_v(s_t)$와 정책 손실 함수 $L_\pi(s_t)$의 평균으로 이루어지며 각각 다음과 같이 정의 하였다. 이 때 손실 계수 α는 0.5, 엔트로피 계수 β는 0.01로 설정하였다.

$$L_v(s_t) = \alpha \sum (R - V(s_t))^2 \tag{3}$$

$$L_\pi(s_t) = -\log(\pi(s_t)) \times A(S_t) + \beta \sum (\pi(S_t) \times \log(\pi(S_t) + 10^{-10})) \tag{4}$$

$$L(s_t) = \frac{L_v(s_t) + L_\pi(s_t)}{2} \tag{5}$$

2D 슈팅 게임으로는 OpenAI Gym에서 제공하는 SpaceInvaders-v0을 사용하였다. 이 게임의 특징은 플레이어가 미사일을 발사한 후 일정 시간이 지난 후 적이 맞았을 경우 해당 적 종류에 따른 점수가 올라간다는 것이다. 즉, 사용자가 특정 액션을 취한 후 일정 시간이 지나야 해당 액션에 대한 보상 값이 주어지며, 보상 값이 일정한 것이 아니라 여러 종류가 될 수 있다는 것을 의미한다. 본 논문에서는 여러 종류의 0보다 큰 보상 값을 모두 1로 하고, 보상이 없는 경우는 0, 플레이어가 적에게 맞았을 경우 -1로 통일하여 에이전트가 효과적으로 학습할 수 있도록 하였다.

에이전트가 게임 학습 시 학습 샘플간의 상관관계로 인한 극소값 문제 (Local minimum problem)에 빠지지 않게 하기 위하여 ε-탐욕 알고리즘 (ε-greedy algorithm)을 사용하였다[9]. 즉, ε의 확률로 임의의 행동을 취하고 1-ε의 확률로 최적 정책에 따른 행동을 취하는 것이다. 본 논문에서 ε는 70,000번에 거쳐 0.5에서 0.1로 선형적으로 감소하도록 설정하였다. 또한 A3C를 사용하여 16개의 스레드 (Thread)가 독립적인 환경에서 게임을 수행하도록 하였고, 각 스레드가 신경망 파라미터를 갱신하기 전에 8회의 액션 결과를 축적하였다. 에이전트의 게임 학습을 위해 2만회의 에피소드를 수행하였다.

실험 환경은 OpenAI Gym 0.9.6과 Keras 2.2.2를 통한 python 프로그램을 작성하여 구축하였다. 신경망 파라미터를 초기화시키기 위해 RandomNormal(mean=0.0, stddev=0.01)을 사용하였다. 파라미터 갱신 시 사용한 최적화 알고리즘은 RMSPropOptimizer (learning_rate = 1.0×10^{-4}, decay = 0.99)이다. 작성된 python 프로그램은 Nvidia GeForce GTX 1080ti GPU에서 실행되었다.

Ⅳ. 성능 평가

각 활성화 함수가 성능에 끼치는 영향을 알아보기 위하여 하나의 에피소드가 끝났을 때 총 게임 점수인 총 보상 값(Total reward)과 가치 함수 값, 손실 함수 값을 활성화 함수 별로 비교하였다.

표 2는 각 활성화 함수 별 총 보상 값의 평균 및 최대값을 비교한 표이다. 이를 ELU의 파라미터 별로 나타낸 그래프가 그림 3이다. ELU 1.0을 사용하였을 때 보상 값이 가장 우수하게 나오는 것을 알 수 있다. 또한 LeakyRelu도 파라미터 별로 그래프를 그려보면 그림 4와 같다. LeakyRelu는 파라미터가 0.1일 때 가장 보상 값이 좋을 것을 알 수 있다. 이 두 활성화 함수를 다른 활성화 함수와 비교한 그래프가 그림 5이다. 평균적으로 ELU 1.0이 다른 활성화 함수에 비해 우수한 것을 알 수 있으며 최댓값도 Relu를 제외하고 가장 높게 나왔다. 그러나 Relu는 5500 스텝에서 최고 보상 값을 얻은 후 성능이 계속 하락하여 평균적으로 보상 값이 낮게 나왔다. 그림 6은 각 활성화 함수에 대한 보상 값을 바이올린 그래프로 나타낸 것이다. 각 보상 값의 분포가 평균 값에 모여 분포하는 것을 알 수 있다.

표 3은 각 활성화 함수 별 가치 함수 $V(s_t)$의 평균 및 최댓값을 비교한 표이다. 이를 ELU 파라미터 별로 나타낸 그래프가 그림 7이다. 파라미터가 1.0일 때 평균 값이 가장 높게 나왔으며 최고 값을 가졌던 파라미터도 값이 1.0이었을 때였다. 그림 8은 LeakyRelu를 파라미터 별로 $V(s_t)$의 값을 나타낸 그래프이다. 0.3일 경우 값이 평균적으로 높게 나왔으며 최댓값은 0.1이었을 경우였다. 그림 9는 모든 활성화 함수에 대하여 $V(s_t)$를 나타낸 그래프이다. 보상 값과 마찬가지로 ELU 1.0이 다른 활성화 함수에 비해 높은 값을 가지는 것을 알 수 있다. 그림 10은 그 분포를 나타낸 것으로 보상 값에 비해 큰 분산을 가지는 것을 알 수 있다.

표 4는 각 활성화 함수 별 손실 함수 $L(s_t)$의 평균 및 최댓값을 비교한 표이다. 이를 ELU 파라미터 별로 나타낸 그래프가 그림 11이다. 그림 12는 LeakyRelu의 파라미터 별로 $L(s_t)$의 값을 나타낸 것이다. 그림 13은 모든 활성화 함수에 대하여 $L(s_t)$를 나타낸 그래프이다. 이 그래프에 대한 분포는 바이올린 그래프로 그림 14에 나타냈다. $L(s_t)$는 다른 값들과는 달리 중앙값보다 낮은 곳에 평균이 위치하고 그 중심으로 분포하는 것을 알 수 있다.

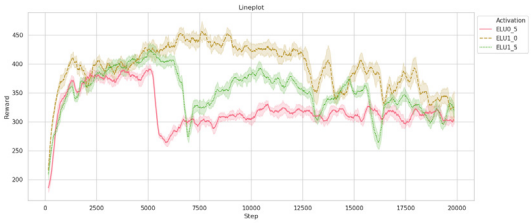

그림 3. ELU 활성화 함수를 사용하였을 때 보상 값
Fig. 3. Rewards for ELU activation functions

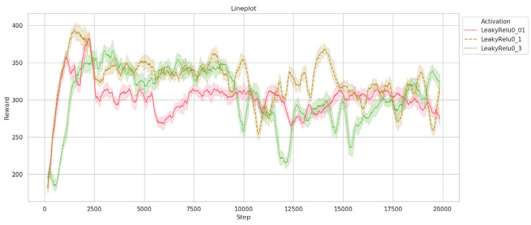

그림 4. LeakyRelu 활성화 함수를 사용하였을 때 보상 값
Fig. 4. Rewards for LeakyRelu activation functions

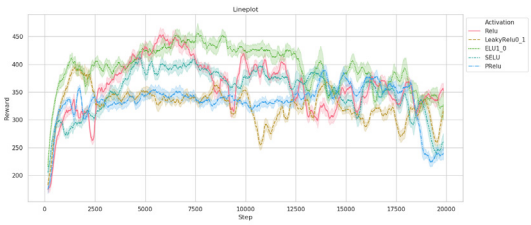

그림 5. 활성화 함수 별 보상 값
Fig. 5. Rewards for activation functions

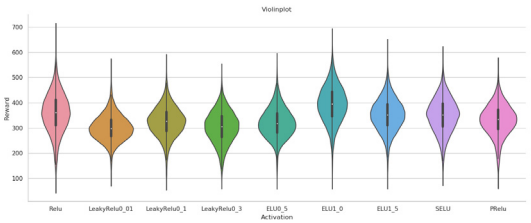

그림 6. 활성화 함수 별 보상 값에 대한 바이올린 그래프
Fig. 6. Violin plot of rewards for each activation function

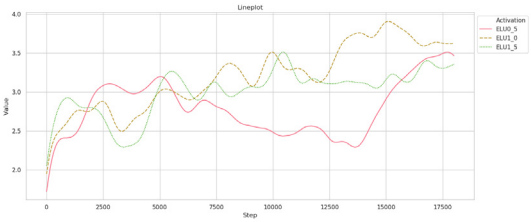

그림 7. ELU 활성화 함수를 사용하였을 때 $V(s_t)$ 값
Fig. 7. $V(s_t)$ for ELU activation functions

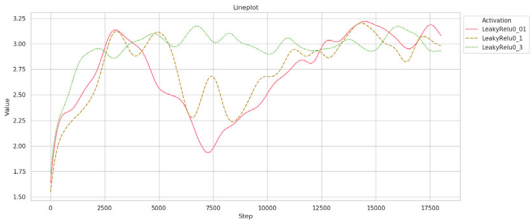

그림 8. LeakyRelu 활성화 함수를 사용하였을 때 $V(s_t)$ 값
Fig. 8. $V(s_t)$ for LeakyRelu activation functions

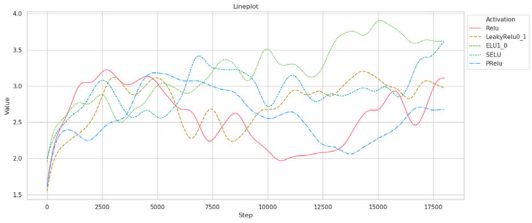

그림 9. 활성화 함수 별 $V(s_t)$ 값
Fig. 9. $V(s_t)$ for activation functions

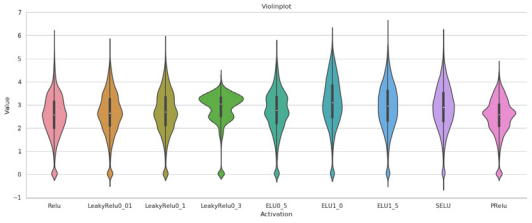

그림 10. 활성화 함수 별 $V(s_t)$ 값에 대한 바이올린 그래프
Fig. 10. Violin plot of $V(s_t)$ for each activation function

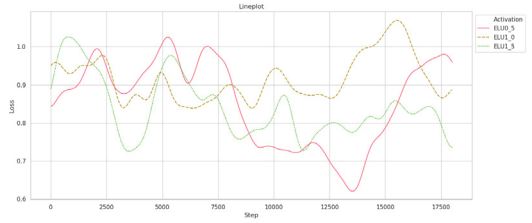

그림 11. ELU 활성화 함수를 사용하였을 때 $L(s_t)$ 값
Fig. 11. $L(s_t)$ for ELU activation functions

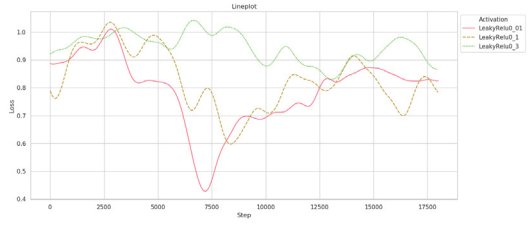

그림 12. LeakyRelu 활성화 함수를 사용하였을 때 $L(s_t)$ 값
Fig. 12. $L(s_t)$ for LeakyRelu activation functions

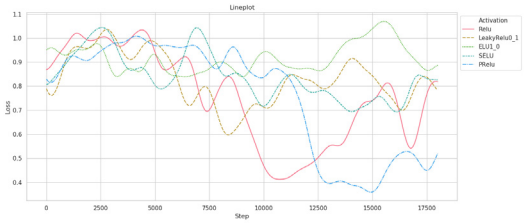

그림 13. 활성화 함수 별 $L(s_t)$ 값
Fig. 13. $L(s_t)$ for activation functions

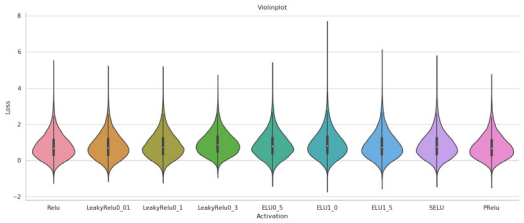

그림 14. 활성화 함수 별 $L(s_t)$ 값에 대한 바이올린 그래프
Fig. 14. Violin plot of $L(s_t)$ for each activation function

V. 결 론

본 논문은 2D 슈팅 게임을 학습하는데 필요한 에이전트가 어떤 활성화 함수를 사용하는지에 따른 성능을 평가하였다. 강화학습에서 가장 많이 사용되는 5가지 활성화 함수인 Relu, PRelu, SELU, LeakyRelu, ELU를 대상으로 하였으며 에이전트를 학습시키기 위해 A3C를 사용하였다. 성능 평가 결과 파라미터 1.0을 사용한 ELU가 다른 활성화 함수에 비해 높은 보상 값을 빠른 시간 내에 달성하였다. 이에 비해 파라미터 0.01을 사용한 LeakyRelu는 가장 적은 보상 값을 달성하였으며, 이는 ELU 1.0에 비해 23.6% 낮은 값이다.

향후 연구로는 딥러닝 에이전트가 게임을 학습하는데 필요한 다른 여러 요인에 대하여 성능 평가를 할 것이다. 이를 통해 게임 스타일 별로, 또는 학습 알고리즘 별로 어떤 요소를 사용하는 것이 효과적인지 알아볼 수 있을 것이다.

References

[1] V. Mnih, A.P. Badia, M. Mirza, A. Graves, T.

Lillicrap, T. Harley, D. Silver and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning”, Proceedings of the International Conference on Machine Learning (ICML), pp. 1928-1937, 2016.

[2] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao and S. Yan, “HCP: A Flexible CNN Framework for Multi-Label Image Classification”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 38, Iss. 9, pp. 1901-1907, 2016.
DOI: 10.1109/TPAMI.2015.2491929

[3] S.-G. Choi and W. Xu, “A Study on Person Re-Identification System using Enhanced RNN”, The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 17, No. 2, 2017.
DOI: 10.7236/JIIBC.2017.17.2.15

[4] I.-T. Joo and S.-H. Choi, "Stock Prediction Model based on Bidirectional LSTM Recurrent Neural Network", The Journal of KIIECT, Vol. 11, No. 2, 2018.
DOI: 10.17661/jkiiect.2018.11.2.204

[5] S.-H. Ryu and J.-B. Yoon, The Effect of regularization and identity mapping on the performance of activation functions, Journal of the Korea Academia-Industrial cooperation Society(JKAIS), Vol. 18, No. 10 pp. 75-80, 2017.
DOI: 10.5762/KAIS.2017.18.10.75

[6] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang and W. Zaremba, “OpenAI Gym”, arXiv:1606.01540, 2016.

[7] I. Arora and A. Saha, “Comparison of back propagation training algorithms for software defect prediction”, Proceedings of the International Conference on Contemporary Computing and Informatics, pp. 51-58, 2016.
DOI: 10.1109/IC3I.2016.7917934

[8] S. Squartini, A. Hussain and F. Piazza, “Preprocessing based solution for the vanishing gradient problem in recurrent neural networks”, Proceedings of the International Symposium on Circuits and Systems, Vol. 5, 2003.
DOI: 10.1109/ISCAS.2003.1206412

[9] T. A. Feo and M. G. C. Resende, “Greedy Randomized Adaptive Search Procedures”, Journal of Global Optimization, Vol. 6, Iss. 2, pp. 109-133, 1995.
DOI: 10.1007/BF01096763

저자 소개

이 동 철(종신회원)

- 2002년 : POSTECH 컴퓨터공학 학사
- 2004년 : POSTECH 전자컴퓨터공학 석사
- 2004년~2012년 : KT 무선연구소 책임연구원
- 2012년 : 한양대학교 전자컴퓨터 통신공학 박사
- 2012년~현재 : 한남대학교 멀티미디어공학과 교수
- 관심분야 : 딥러닝, 게임, 신경망, 알고리즘

박 병 주 (종신회원)

- 2002년 : 연세대학교 전기전자공학 학사
- 2004년 : University of Florida ECE 석사
- 2007년 : University of Florida ECE 박사
- 2007년~2009년 : KT 기술연구소 선임연구원
- 2009년~현재 : 한남대학교 멀티미디어공학과 교수
- 관심분야 : Mobility Management, Proxy Mobile IPv6, IEEE 802.11, IEEE 802.16e, Seamless Handover, Mobile IPTV, NGN, IMS, Service Oriented Architecture