The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 21, No. 6, pp.117-122, Dec. 31, 2021. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2021.21.6.117
JIIBC 2021-6-17

교차로에서 자율주행을 위한 심층 강화 학습 활성화 함수 비교 분석

Comparison of Activation Functions using Deep Reinforcement Learning for Autonomous Driving on Intersection

이동철*

Dongcheul Lee*

요 약 자율주행은 자동차가 사람 없이 운전할 수 있도록 해 주며 최근 인공지능 기술의 발전에 힘입어 매우 활발히 연구되고 있다. 인공지능 기술 중에서도 특히 심층 강화 학습이 가장 효과적으로 사용되는데 이를 위해서는 적절한 활성화 함수를 이용한 신경망 구축이 필수적이다. 여태껏 많은 활성화 함수가 제시됐으나 적용 분야에 따라 서로 다른 성능을 보여주었다. 본 논문은 교차로에서 자율주행을 학습하기 위해 심층 강화 학습을 사용할 때 어떤 활성화 함수를 사용하는 것이 효과적인지 성능을 비교 평가한다. 이를 위해 평가에서 사용할 성능 메트릭을 정의하고 각 활성화 함수에 따른 메트릭의 값을 그래프로 비교하였다. 그 결과 Mish를 사용할 경우 보상이 다른 활성화 함수보다 평균적으로 높은 것을 알 수 있었고 보상이 가장 낮은 활성화 함수와의 차이는 9.8%였다.

Abstract Autonomous driving allows cars to drive without people and is being studied very actively thanks to the recent development of artificial intelligence technology. Among artificial intelligence technologies, deep reinforcement learning is used most effectively. Deep reinforcement learning requires us to build a neural network using an appropriate activation function. So far, many activation functions have been suggested, but different performances have been shown depending on the field of application. This paper compares and evaluates the performance of which activation function is effective when using deep reinforcement learning to learn autonomous driving on highways. To this end, the performance metrics to be used in the evaluation were defined and the values of the metrics according to each activation function were compared in graphs. As a result, when Mish was used, the reward was higher on average than other activation functions, and the difference from the activation function with the lowest reward was 9.8%.

Keywords : Autonomous Driving, Deep Learning, Reinforcement Learning

*종신회원, 한남대학교 멀티미디어공학과
접수일자 2021년 9월 16일, 수정완료 2021년 11월 16일
게재확정일자 2021년 12월 10일

Received: 16 September, 2021 / Revised: 16 November, 2021 / Accepted: 10 December, 2021
*Corresponding Author: jackdclee@hnu.kr
Dept. of Multimedia Engineering, Hannam University, Korea

Ⅰ. 서 론

자율주행(Autonomous driving)은 자동차가 사람 없이 목적을 수행하도록 해 준다[1]. 자율주행은 오랜 시간 동안 발전되기 어려운 분야로 여겨졌으나 최근 인공지능 기술의 발전에 힘입어 매우 활발히 연구되고 있으며 자동차 제조사들도 상용화에 힘쓰고 있는 분야이다[2]. 자율주행은 인지, 의사 결정, 계획, 제어의 네 가지의 단계로 이루어진다. 이중 의사 결정 단계는 아주 중요한 단계인데 사람의 경험에 의한 룰 (Rule) 기반의 정책으로 만들어지거나 지도 학습 (Supervised learning)을 통한 모델 기반으로 만들어져왔다[3]. 최근에는 심층 강화 학습 (Deep reinforcement learning)을 이용하여 의사 결정 정책을 학습하며 가장 효과적으로 사용되는 방법이다[4]. 특히 자동차의 차선 유지, 충돌 방지, 연료 절감, 주차, 이동 속도, 차선 변경, 교차로 통과 등 다양한 연구 분야에서 심층 강화 학습이 자율주행 정책 결정 도구로 사용되고 있다. 심층 강화 학습의 주된 목적은 특정 상황에서 에이전트가 가장 높은 보상을 낼 수 있도록 액션을 선택하는 정책을 만드는 것이다. 따라서 보상을 어떻게 설정하냐에 따라 특정 목적에 맞는 정책을 만드는 것이 가능하다. 예를 들어 교차로를 통과하는 상황에서는 다른 차와 충돌 없이 빠르게 통과할 때 보상이 주어질 수 있다.

인지 단계에서 사용되는 기술에는 여러 가지가 혼합되어 사용되는데 카메라, 레이더 (Radar), 라이다 (Lidar)와 같은 외부환경 인지 센서가 사용된다. 최근에는 카메라만 이용하여 거리 및 속도, 각도, 객체 탐지하는 연구도 활발히 진행 중이다. 카메라를 이용한 인지를 위해서는 심층 강화 학습 시 일반적으로 CNN (Convolutional Neural Network)으로 신경망을 구성한다[5]. CNN은 픽셀 데이터를 입력받아 데이터 내의 다양한 개체에 대하여 중요성을 부여하고 이를 통해 다른 개체와 구별해 내는 방법을 제공한다.

심층 신경망 구성 시 많은 계층 (Layer)을 사용하게 되면 각 계층을 결합하기 위해 활성화 함수 (Activation Function)를 사용하게 된다. 어떤 활성화 함수를 사용하냐에 따라 에이전트의 성능에 큰 영향을 끼치게 되며 성능 향상을 위해 그동안 연구자들은 많은 활성화 함수를 제안해왔다[6]. 그러나 모든 환경에서 일반적으로 우수한 활성화 함수는 아직 발견되지 않았으며 각 환경에 맞게 활성화 함수를 비교 선택하여 사용해 왔다. 본 논문은 교차로에서 자율주행을 학습하기 위한 심층 강화 학습 에이전트를 제시하며 신경망에 다양한 활성화 함수를 사용하여 어떤 함수를 사용하는 것이 학습에 유리한지 성능을 비교 평가하고자 한다.

본 논문은 다음과 같은 구성이다. 2장에서는 본 논문에서 사용할 자율주행 시뮬레이션 환경과 신경망 구성 시 사용된 활성화 함수에 대하여 알아본다. 3장에서는 에이전트의 성능을 비교할 방법에 대하여 정의하고 4장에서는 성능 평가 결과를 분석한다. 마지막으로 5장에서는 결과를 종합하고 결론을 제시한다.

Ⅱ. 관련 연구

1. Highway-env

Highway-env는 자율주행을 시뮬레이션하기 위한 여러 환경을 제공해주고 안전한 운행 정책을 만드는 데 도움을 주는 도구다[7]. 5개의 환경을 제공하며 각각 단순한 고속도로 환경, 도로가 합쳐지는 환경, 회전 교차로 환경, 주차 환경, 교차로 환경을 제공한다. 주차 환경을 제외한 모든 환경에서 시뮬레이션 대상 자동차를 제외한 다른 차들의 개수와 움직이는 방식을 조정할 수 있다.

또한, 주어진 환경이 어떻게 동작하고 있는지 관측하는 방법을 4가지로 제공한다. 첫 번째 방법은 Kinematics로 근처의 자동차들 V에 대한 특징 (Feature) F를 $V \times F$ 배열로 나타낸다. 두 번째 방법은 Grayscale image로 $W \times H$ 화면에 대한 회색 이미지를 나타낸다. 세 번째 방법은 Occupancy grid로 자동차를 감싸는 특정 사각형 셀 공간 안을 $W \times H$의 비선형 수치로 나타내고 각 셀은 F 특징을 가지는 $W \times H \times F$ 배열로 나타낸다. 마지막 방법은 Time to collision으로 $V \times L \times H$ 배열로 자동차와 같은 도로의 다른 자동차가 부딪치는데 걸리는 시간을 나타낸다. 여기서 V는 자동차의 속도, L은 현재 차선 주위의 차선 수, H는 부딪칠 가능성을 1초마다 one-hot 인코딩으로 표현하는 개수를 뜻한다.

2. 활성화 함수

일반적으로 강화 학습을 위해 신경망이 사용되는데, 신경망은 여러 뉴런 (Neuron)으로 구성되어 있다. 뉴런은 입력값을 가중치 w와 곱하고 변형 b를 더하여 활성화되는 함수이다. 해당 뉴런의 활성화는 활성화 함수를 거쳐 다른 뉴런으로 전달되며 이러한 과정이 마지막 뉴런

표 1. 강화 학습에서 사용되는 활성화 함수
Table 1. Activation functions used in the reinforcement learning

Name	Equation
ReLU	$f(x)=\begin{cases}0, & \text{for } x<0\\ x, & \text{for } x\geq 0\end{cases}$
LeakyReLU	$f(x,a)=\begin{cases}ax, & \text{for } x<0\\ x, & \text{for } x\geq 0\end{cases}$
PReLU	$f(x)=\begin{cases}ax, & \text{for } x<0\\ x, & \text{for } x\geq 0\end{cases}$
GELU	$f(x)=x\Phi(x)$,where $\Phi(x)=x\frac{1}{2}\left[1+erf(\frac{x}{\sqrt{2}})\right]$
SiLU	$f(x)=x\sigma(x)$,where $\sigma(x)=\frac{1}{1+e^{-x}}$
Mish	$f(x)=x\,Tanh(\ln(1+e^{x}))$

까지 반복된다. 이렇게 활성화 함수는 신경망으로 입력값이 어떤 범위까지 계속 전달될지를 결정하는 역할을 한다. 일반적인 형태의 활성화 함수는 다음과 같이 정의된다.

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

표 1은 본 논문에서 사용된 활성화 함수를 나타낸다. 강화 학습 초기에 사용된 활성화 함수는 시그모이드 (Sigmoid) 함수였다. 이 활성화 함수의 출력은 0에서 1 사이에 존재하기 때문에 확률을 출력으로 사용할 수 있으므로 사용되었다. 또한, 미분 가능하므로 역전파 (Backpropagation)를 사용할 수 있었다. 그러나 x 값이 매우 크거나 매우 작을 경우 예측에 변화가 거의 없으므로 기울기 소멸 문제 (Vanishing gradient problem)가 발생한다. 이로 인해 모델을 학습하기 너무 느려지거나 더 이상 학습이 안 되는 문제가 발생한다.

이를 해결한 것이 ReLU (Rectified Linear Unit)이며 현재 가장 많이 사용되는 활성화 함수이다. 이 함수는 계산하기 효율적이며 비선형성으로 인해 네트워크를 빨리 수렴할 수 있게 하며 역전파를 가능하게 한다. 그러나 이 함수도 입력이 0에 가까워지거나 음수가 되면 함수의 미분값이 0이 되어 네트워크가 역전파를 수행할 수 없고 학습을 더 이상할 수 없게 되는 한계가 있다.

LeakyReLU는 음의 영역에서도 작은 양의 기울기를 가지므로 음의 입력값에도 역전파가 가능하다[8]. 본 논문에서는 음의 기울기 a로 $1e^{-2}$를 사용하였다. PReLU는 학습용 파라미터 a를 사용한다. 본 논문에서는 그 초기값으로 0.25를 사용하였다. GELU (Gaussian Error Linear Unit)는 입력값에 Gaussian 분포에 대한 누적 분포함수를 곱한 값을 결과로 사용한다[9]. SiLU (Sigmoid-weighted Linear Unit)은 입력 값에 시그모이드 함수를 곱한 값을 결과값으로 사용한다[10]. Mish는 스스로 정규화되는 비단조 함수로써 입력 값 x와 x의 softplus 함수를 tanh 함수 입력으로 사용한 결과를 곱한 값이다[11].

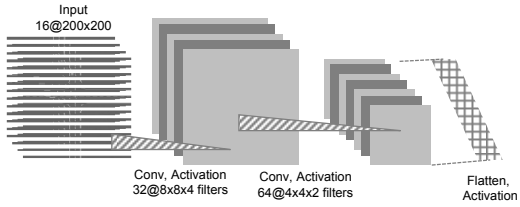
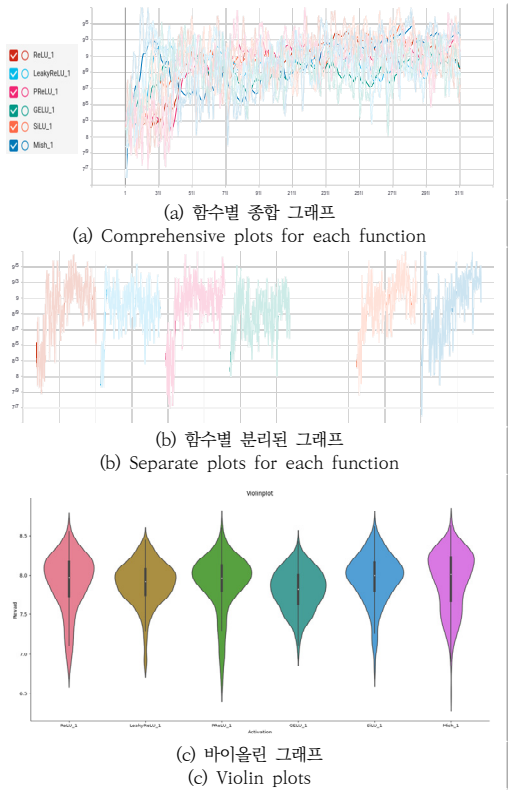


그림 1. 교차로에서 자율주행을 위한 강화 학습 에이전트가 사용하는 신경망 구성도
Fig. 1. Illustration of the neural network used by RL agent to learn how to learn self-driving on a intersection

Ⅲ. 성능 평가 방법

본 논문은 교차로에서 자율주행을 학습하기 위한 심층 강화 학습 에이전트를 제시한다. 이 에이전트는 학습을 위해 PPO (Proximal Policy Optimization) 알고리즘을 이용하였고 신경망으로 CNN을 사용하였다[12]. 그림 1은 에이전트가 사용한 신경망을 보여준다. 신경망에 사용될 입력값은 200x200의 화면을 캡쳐한 회색 화면의 이미지 16장이 한 번에 사용된다. 네트워크는 이 입력 이미지의 픽셀 데이터를 읽어서 어떤 액션을 취할지 결정한다. 네트워크는 2개의 CNN 계층과 1개의 완전연결계층 (Fully-connected network)을 사용한다. CNN에 사용된 필터는 각각 32, 64개이며, 필터 크기는 각각 8x8과 4x4, 스트라이드 (Stride)는 각각 4와 2이다. 활성화 함수는 각 은닉층 (Hidden layer)에 사용되었다.

에이전트가 학습할 교차로를 시뮬레이션하기 위해 Highway-env의 intersection-v0를 사용하였다. 자동차가 취할 수 있는 액션은 감속, 등속, 가속의 3가지이다. 하나의 교차로 에피소드는 자동차가 교차로를 통과하거나 다른 자동차와 충돌했을 경우 종료된다. 교차로 환경에서 자동차의 목적은 두 가지다. 첫째로 교차로를 빨리 통과하는 것이고 두 번째는 다른 자동차와 충돌을 피하는 것이다. 이를 위해 보상 함수 (Reward function)는 다음과 같이 정의하였다.

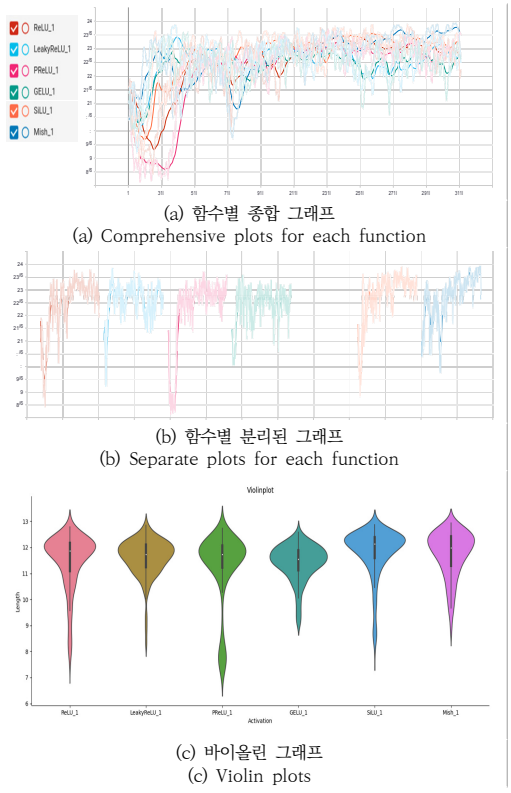


(a) 함수별 종합 그래프
(a) Comprehensive plots for each function

(b) 함수별 분리된 그래프
(b) Separate plots for each function

(c) 바이올린 그래프
(c) Violin plots

그림 2. 에이전트가 학습하는 동안 타임스탬프에 따른 보상을 나타낸 그래프
Fig. 2. A plot showing rewards according to timestamps while the agent is learning



(a) 함수별 종합 그래프
(a) Comprehensive plots for each function

(b) 함수별 분리된 그래프
(b) Separate plots for each function

(c) 바이올린 그래프
(c) Violin plots

그림 3. 에이전트가 학습하는 동안 타임스탬프에 따른 에피소드를 끝내는 시간을 나타낸 그래프
Fig. 3. A plot showing the time it takes to end the episode according to timestamps while the agent is learning

$$R(a,b) = a\frac{v - v_{\min}}{v_{\max} - v_{\min}} - b \times c \qquad (2)$$

여기서 v는 현재 자동차 속도, $v_{\min}$과 $v_{\max}$는 각각 최저 속도와 최고 속도이다. c는 충돌 횟수이며 a와 b는 계수이다. 보상 함수의 결과는 [0, 1] 범위로 정규화하였다.

에이전트는 타임스텝 t에 주행 상태 s_t를 픽셀 데이터로 관찰할 수 있고 정책 파라미터 θ를 기반으로 한 확률적 정책 π_θ에 따라 어떤 액션 a_t을 취할지 결정한다. PPO는 on-policy 알고리즘이므로 새로운 정책이 이전 정책에서 멀어지는 것을 막기 위해 목적 함수(Objective function)에 클리핑 (Clipping)을 사용하며 목적 함수는 다음과 같이 정의한다.

$$L^{CLIP}(\theta) = \hat{E}_t\left[\min\begin{matrix}(r_t(\theta)\hat{A}_t, clip(r_t(\theta),\\ 1-\epsilon, 1+\epsilon)\hat{A}_t\end{matrix}\right] \qquad (3)$$

$\hat{E}_t$는 샘플링과 최적화를 번갈아 수행하는 알고리즘에서 타임스텝 t에 유한 표본 배치에 대한 경험적 기댓값(Empirical expectation)을 의미한다. $\hat{A}_t$는 타임스텝 t에 Advantage 함수의 추정량(Estimator)을 의미한다. ϵ는 하이퍼 파라미터로 0.1 또는 0.2의 값을 갖는다. 확률비 $r_t(\theta)$는 다음과 같이 정의한다.

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \qquad (4)$$

에이전트는 Intel i7-9700 CPU와 Nvidia GeForce RTX 2060 GPU, 32G 메모리를 탑재한 우분투 18.04 머신에서 학습하였다. 에이전트는 Python 3.9, PyTorch 1.9, Stable-baselines 3, OpenAI Gym 0.18, Highway-env 1.2를 이용하여 만들었다.

Ⅳ. 성능 평가

각 활성화 함수가 교차로에서 자율주행을 학습할 때

표 2. 테스트 환경에서 각 활성화 함수별 평균 보상과 충돌 횟수
Table 2. Mean reward and crashed count for each activation function during the testing

Activation Function	Reward	Crashed Count
ReLU	7.915	437
LeakyReLU	8.126	303
PReLU	8.255	297
GELU	7.994	330
SiLU	7.746	388
Mish	8.503	303

끼치는 영향을 평가하기 위해 각 활성화 함수를 사용하여 학습할 때 얻게 되는 보상과 에피소드를 끝내는데 걸리는 시간을 비교하였다.

그림 2는 에이전트가 학습하는 동안 타임스탬프에 따른 보상을 나타낸 그래프이다. 그래프는 smoothing factor를 0.9로 설정하여 그려졌으며 원본 데이터는 흐린 색으로 표시하였다. 모든 활성화 함수가 공통으로 타임스탬프가 지날수록 보상이 늘어나다가 특정 시점에서 늘어나는 속도가 급격히 줄어드는 것을 알 수 있다. (a)에서 학습이 끝난 타임스탬프 기준으로 Mish가 가장 높은 보상을 기록하였으며 ReLU가 가장 낮았다. (b)를 보면 Mish를 사용하였을 경우 보상이 증가하는 기울기가 다른 그래프에 비해 높은 것을 확인할 수 있다. (c)를 통해 가장 높은 분포를 띄는 보상이 어떤 값인지 확인할 수 있는데 Mish가 가장 높은 보상에서 높은 분포를 보였으며 GELU는 가장 낮은 보상에서 높은 분포를 보였다. 실제로 (a)에서 GELU의 경우 대부분의 값이 다른 활성화 함수보다 낮은 경향을 보인다.

그림 3은 에이전트가 학습하는 동안 타임스탬프에 따른 에피소드를 끝내는 시간을 비교한 그래프이다. (a)에서 학습이 끝난 시점을 기준으로 Mish의 시간이 가장 길고 GELU가 가장 짧았다. (b)를 보면 GELU의 경우 가장 빨리 해당 시간에 도달한 후 거의 증감이 없었으며 Mish의 경우 조금씩 계속 증가하는 것을 알 수 있다. (c)를 통해 가장 높은 분포를 띄는 시간을 확인할 수 있는데 SiLU의 경우 가장 긴 시간에서 높은 분포를 보였으며 GELU의 경우 가장 낮은 시간에서 분포가 높았다.

표 2는 학습이 종료된 테스트 환경에서 각 활성화 함수별 평균 보상과 충돌 횟수를 나타낸다. 평균 보상이 가장 높은 함수는 Mish였고, 가장 낮은 함수는 SiLU였으며 두 값의 차이는 9.8%였다. 충돌 횟수가 가장 적었던 함수는 PReLU였고 가장 많았던 함수는 ReLU였으며 두 값의 차이는 47.1%였다. Mish의 경우 보상이 가장 높았지만 충돌 횟수도 두 번째로 적었으며 PReLU의 경우도 충돌 횟수가 가장 적었으나 보상도 두 번째로 높았다. 따라서 Mish와 PReLU 모두 좋은 성능을 보여준다고 할 수 있다.

V. 결 론

본 논문은 교차로에서 자율주행 학습 시 어떤 활성화 함수를 사용하는 것이 심층 강화 학습에 유리한지 성능을 평가하였다. 성능 평가 시 비교했던 활성화 함수는 일반적으로 많이 사용되는 ReLU, LeakyReLU, PReLU, GELU, SiLU, Mish를 사용하였다. 성능 평가 결과 Mish를 사용할 경우 가장 높은 보상을 얻을 수 있었으며 가장 낮은 보상을 기록한 SiLU와의 차이는 9.8% 였다. 두 번째로 높은 보상은 얻은 함수는 PReLU였고 충돌 횟수는 가장 적었다.

향후 연구로는 심층 강화 학습 에이전트가 자율주행을 학습하는데 필요한 다른 여러 요인에 대하여 성능 평가를 할 것이다. 이를 통해 자율주행 상황별로, 또는 학습 알고리즘별로 어떤 요소를 사용하는 것이 효과적인지 알아볼 수 있을 것이다.

References

[1] E. Jang, J. Kim, “Proposal of New Information Processing Model for Implementation of Autonomous Mobile System”, The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 19, No. 2, pp. 237-242, 2019.
DOI: https://doi.org/10.7236/JIIBC.2019.19.2.237

[2] A. Rasouli, J. K. Tsotsos, “Autonomous vehicles that interact with pedestrians: A survey of theory and practice”, IEEE Trans. Intell. Transp. Syst., Vol. 21, No. 3, pp. 900-918, 2020.
DOI: https://doi.org/10.1109/TITS.2019.2901817

[3] P. Hart, A. Knoll, “Using counterfactual reasoning and reinforcement learning for decision-making in autonomous driving”, arXiv:2003.11919, 2020.

[4] M. Kim, S. Lee, J. Lim, J. Choi, S. Kang, “Unexpected collision avoidance driving strategy using deep reinforcement learning”, IEEE Access, Vol. 8, pp. 17243-17252, 2020.
DOI: https://doi.org/10.1109/ACCESS.2020.2967509

[5] G. Shu, W. Liu, X. Zheng, J. Li, “IF-CNN: Image-Aware Inference Framework for CNN With the Collaboration of Mobile Devices and Cloud”, IEEE

Access, Vol 6, pp. 68621-68633, 2018.
DOI: https://doi.org/10.1109/ACCESS.2018.2880196

[6] M. Lau, K. Lim, "Review of Adaptive Activation Function in Deep Neural Network", 2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences, pp. 686-690, 2018.
DOI: https://doi.org/10.1109/IECBES.2018.8626714.

[7] L. Edouard, "An Environment for Autonomous Driving Decision-Making", GitHub, https://github.com/eleurent/highway-env, 2018.

[8] T. Jiang, J. Cheng, "Target Recognition Based on CNN with LeakyReLU and PReLU Activation Functions", 2019 International Conference on Sensing, Diagnostics, Prognostics, and Control, pp. 718-722, 2019.
DOI: https://doi.org/10.1109/SDPC.2019.00136

[9] D. Hendrycks, K. Gimpel, "Gaussian Error Linear Units", arXiv:1606.08415, 2020.

[10] S. Elfwing, E. Uchibe, K. Doya. "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning", Neural Networks, Vol. 107, pp. 3-11, 2018.
DOI: https://doi.org/10.1016/j.neunet.2017.12.012

[11] D. Misra, "Mish: A Self Regularized Non-Monotonic Activation Function", arXiv:1908.08681, 2020.

[12] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, "Proximal Policy Optimization Algorithms", arXiv:1707.06347, 2017.

저 자 소 개

이 동 철(종신회원)

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