The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 22, No. 5, pp.63-68, Oct. 31, 2022. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2022.22.5.63
JIIBC 2022-5-10

고속도로 자율주행 시 보상을 최대화하기 위한 강화 학습 활성화 함수 비교

Comparison of Reinforcement Learning Activation Functions to Maximize Rewards in Autonomous Highway Driving

이동철*

Dongcheul Lee*

요 약 자율주행 기술은 최근 심층 강화학습의 도입으로 큰 발전을 이루고 있다. 심층 강화 학습을 효과적으로 사용하기 위해서는 적절한 활성화 함수를 선택하는 것이 중요하다. 그 동안 많은 활성화 함수가 제시되었으나 적용할 환경에 따라 다른 성능을 보여주었다. 본 논문은 고속도로에서 자율주행을 학습하기 위해 강화 학습을 사용할 때 어떤 활성화 함수를 사용하는 것이 효과적인지 12개의 활성화 함수 성능을 비교 평가한다. 이를 위한 성능 평가 방법을 제시하였고 각 활성화 함수의 평균 보상 값을 비교하였다. 그 결과 GELU를 사용할 경우 가장 높은 평균 보상을 얻을 수 있었으며 SiLU는 가장 낮은 성능을 보여주었다. 두 활성화 함수의 평균 보상 차이는 20%였다.

Abstract Autonomous driving technology has recently made great progress with the introduction of deep reinforcement learning. In order to effectively use deep reinforcement learning, it is important to select the appropriate activation function. In the meantime, many activation functions have been presented, but they show different performance depending on the environment to be applied. This paper compares and evaluates the performance of 12 activation functions to see which activation functions are effective when using reinforcement learning to learn autonomous driving on highways. To this end, a performance evaluation method was presented and the average reward value of each activation function was compared. As a result, when using GELU, the highest average reward could be obtained, and SiLU showed the lowest performance. The average reward difference between the two activation functions was 20%.

Keywords : Autonomous Driving, Deep Learning, Reinforcement Learning

*종신회원, 한남대학교 멀티미디어공학과
접수일자 2022년 8월 25일, 수정완료 2022년 9월 25일
게재확정일자 2022년 10월 7일

Received: 25 August, 2022 / Revised: 25 September, 2022 / Accepted: 7 October, 2022
*Corresponding Author: jackdclee@hnu.kr
Department of Multimedia Engineering, Hannam University, Korea

Ⅰ. 서 론

자율주행(Autonomous driving) 기술은 운전 미숙에 의한 사고를 방지하고 운전 관련 스트레스를 줄일 수 있는 차세대 기술이다[1]. 이 기술은 오랜 시간 동안 연구되어왔으나 최근에서야 인공지능 기술의 발달로 인해 비약적으로 발전하고 있다. 자율주행은 인지, 의사 결정, 제어의 세 단계로 이루어진다. 인지 단계에서는 카메라, 레이더 (Radar), 라이다 (Lidar)와 같은 외부환경 인지 센서가 사용된다. 의사 결정 단계에서는 인지 단계에서 인식한 개체에 대하여 차선을 계속 유지할지, 충돌을 방지하기 위해 멈출지, 주차할지, 차선을 변경할지, 속도를 변경할지 등을 결정한다. 이 결정 과정에서 사용되었던 방식은 규칙 기반 (Rule-based) 방식, 지도 학습 (Supervised Learning) 방식, 강화 학습 (Reinforcement Learning) 방식 등이 있다[2, 3]. 규칙 기반 방식은 가장 기초적인 방식으로 모든 상황에 대한 규칙을 정해주는 방식이다. 구현하기 가장 쉽지만, 주행의 모든 상황을 고려하는 규칙을 만들기가 사실상 어렵기 때문에 거의 사용되지 않는다. 지도 학습 방식은 다량의 사람 운전자의 데이터를 이용하여 신경망을 학습시키는 것을 말한다. 오프라인 방식으로 학습할 수 있으나 일반화 성능이 떨어지는 단점이 있다. 강화 학습 방식은 가장 높은 보상을 낼 수 있도록 어떤 액션을 취할지 선택하는 정책을 만드는 것이다. 보상을 어떻게 계산할지 설정하느냐에 따라 특정 목적에 맞는 정책을 만드는 것이 가능하다.

지도 학습이나 강화 학습 방법을 사용할 때에는 신경망 (Neural Network)을 구성하게 되는데, 신경망을 깊게 구성하는 심층 신경망에서는 많은 계층 (Layer)을 사용한다. 이때 각 계층을 결합하기 위해 활성화 함수 (Activation Function)을 사용하게 되는데 어떤 활성화 함수를 사용하느냐에 따라 해당 계층의 활성화가 결정되므로 학습 결과에 매우 큰 영향을 끼친다. 그동안 많은 연구자가 학습 성능 향상을 위해 다양한 활성화 함수를 제안해왔으나 모든 환경에서 일반적으로 우수한 활성화 함수는 아직 발견되지 않았으며 각 상황에 맞게 활성화 함수를 비교 평가해 가며 사용해 왔다[4]. 본 논문은 고속도로에서 자율주행 시 어떤 활성화 함수를 사용하는 것이 학습에 유리한지 비교 평가하고자 한다.

본 논문은 다음과 같이 구성되었다. 2장은 본 논문에서 비교 평가할 활성화 함수에 대하여 알아본다. 3장은 각 활성화 함수의 성능을 비교할 방법에 대하여 정의하고 4장은 성능 평가 결과를 분석한다. 마지막으로 5장은 결과를 종합하고 결론을 제시한다.

Ⅱ. 관련 연구

강화 학습에서 사용되는 신경망을 활성화할 것인지는 활성화 함수에 달려있다. 따라서 어떠한 활성화 함수를 선택하느냐에 따라 학습 성능이 크게 좌우된다. 강화 학습 초기에 사용된 활성화 함수는 시그모이드(Sigmoid) 함수였다. 이 활성화 함수는 미분 가능하므로 역전파 (Backpropagation)를 사용할 수 있어 초기에 활용되었으나 신경망이 깊어짐에 따라 기울기 소멸 문제 (Vanishing gradient problem)가 발생하여 더는 활용되지 못하였다. 본 논문에서는 시그모이드 활성화 함수의 문제를 개선한 12개의 활성화 함수를 비교 평가하였으며 각 활성화 함수의 특징은 다음과 같다.

1. Rectified Linear Unit (ReLU)

이 함수는 계산하기 효율적이며 비선형성으로 인해 네트워크를 빨리 수렴할 수 있게 하여 현재까지 가장 많이 사용되는 활성화 함수이다. 그러나 입력이 0에 가까워지거나 음수가 되면 네트워크가 역전파를 수행하기 어렵고 학습을 더 이상 할 수 없게 되는 한계가 있다. ReLU는 다음과 같이 정의한다.

$$f(x)=\begin{cases}x, \text{if } x \geq 0\\ 0, \text{if } x < 0\end{cases} \tag{1}$$

2. Softplus

이 함수는 부드러운 버전의 ReLU라고 할 수 있으며 다음과 같이 정의한다[5].

$$f(x)=\log(1+\exp^{x}) \tag{2}$$

3. Leaky Rectified Linear Unit (Leaky ReLU)

이 함수는 ReLU의 변형으로 음의 영역에서도 작은 양의 기울기를 가지므로 음의 입력값에도 역전파가 가능하다[6]. Leaky ReLU는 다음과 같이 정의하며 a는 일반적으로 0.01로 설정한다.

$$f(x)=\begin{cases}x, & \text{if } x > 0\\ \alpha x, & \text{if } x \leq 0\end{cases} \tag{3}$$

4. Exponential Linear Unit (ELU)

이 함수는 ReLU와 달리 배치 정규화와 같이 평균 단

위 활성화를 0에 가깝게 할 수 있지만, 계산 복잡성은 낮다[7]. ELU는 다음과 같이 정의하며 일반적으로 a는 1.0으로 설정한다.

$$f(x)=\begin{cases}x, & \text{if } x>0\\ \alpha\exp(x)-1, & \text{if } x\le 0\end{cases} \quad (4)$$

5. Parametric Rectified Linear Unit (PReLU)

이 함수는 ReLU를 발전시켜 서로 다른 신경망 계층은 서로 다른 비선형 활성화 함수를 필요로 할 것이라는 의미에서 제안되었다[8]. PReLU는 다음과 같이 정의하며 일반적으로 a는 0.25로 설정한다.

$$f(x)=\begin{cases}x, & \text{if } x>0\\ \alpha x, & \text{if } x\le 0\end{cases} \quad (5)$$

6. Gaussian Error Linear Unit (GELU)

표준 가우스 누적 분포 함수를 사용하는 이 함수는 ReLU보다 더 부드러운 곡선을 띈다[9]. 비선형성은 ReLU처럼 부호로 결정하는 것이 아닌 백분위수로 입력 가중치를 부여하며 다음과 같이 정의한다.

$$f(x)=\frac{x}{2}\left[1+erf(\frac{x}{\sqrt{2}})\right] \quad (6)$$

7. Scaled Exponential Linear Unit (SELU)

이 함수는 자기 정규화 특성을 유도하는 기능을 하며 다음과 같이 정의한다[10].

$$f(x)=\begin{cases}\lambda x, & \text{if } x\ge 0\\ \lambda\alpha(\exp(x)-1), & \text{if } x<0\end{cases} \quad (7)$$

일반적으로 a는 1.6733, λ는 1.0507의 값을 가진다.

8. Continuously Differentiable Exponential Linear Units (CELU)

이 함수는 모든 곳에서 미분 가능한 함수이며 다음과 같이 정의한다[11].

$$f(x)=\begin{cases}x, & \text{if } x\ge 0\\ \alpha(\exp(\frac{x}{\alpha})-1), & \text{if } x<0\end{cases} \quad (8)$$

일반적으로 a는 1.0의 값을 가진다.

9. Sigmoid-Weighted Linear Units (SiLU)

이 함수의 활성화는 시그모이드 함수의 입력값을 곱한 값으로 계산하며 다음과 같이 정의한다[12].

$$f(x)=\frac{x}{1+e^{-x}} \quad (9)$$

10. Mish

이 함수는 스스로 정규화하는 비단조 함수로써 입력값과 softplus 함수를 tanh 함수 입력으로 사용한 결과를 곱하여 활성화를 계산하며 다음과 같이 정의한다[13].

$$f(x)=x\,Tanh(\ln(1+e^{x})) \quad (10)$$

11. Hardswish

이 함수는 비선형 버전의 swish로써 계산이 더 빠르고 메모리 접근 수를 줄여 지연 비용을 낮추는 장점이 있으며 다음과 같이 정의한다[14].

$$f(x)=\begin{cases}0, & \text{if } x\le -3\\ x, & \text{if } x\ge 3\\ x(x+3)/6, & otherwise\end{cases} \quad (11)$$

12. Hard Hiperbolic Function (HardTanH)

이 함수는 Tanh 함수의 변형으로 계산이 보다 빠른 장점이 있으며 다음과 같이 정의한다.

$$f(x)=\begin{cases}-1, & \text{if } x<-1\\ 1, & \text{if } x>1\\ x, & otherwise\end{cases} \quad (12)$$

Ⅲ. 성능 평가 방법

고속도로에서 자율주행을 학습할 때 신경망에 사용되는 활성화 함수 성능 평가를 위해 심층 강화 학습 에이전트를 제작하였다. 이 에이전트는 신경망으로 CNN (Convolutional Neural Network)을 사용하며 학습 알고리즘으로 PPO (Proximal Policy Optimization)를 사용하였다[15]. 에이전트는 타임스텝 t에 주행 상태 s_t를 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 적용하며 목적 함수는 다음과 같이 정의하였다.

$$L^{CLIP}(\theta)=\hat{E}_t\left[\min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right] \quad (13)$$

$\hat{E}_t$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.

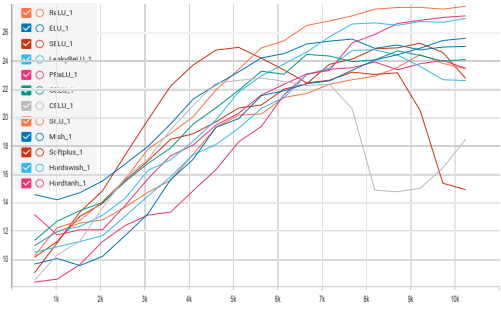

그림 1. 에이전트가 학습하는 동안 타임스탬프에 따른 보상을 나타낸 그래프
Fig. 1. A plot showing rewards according to timestamps while the agent was learning

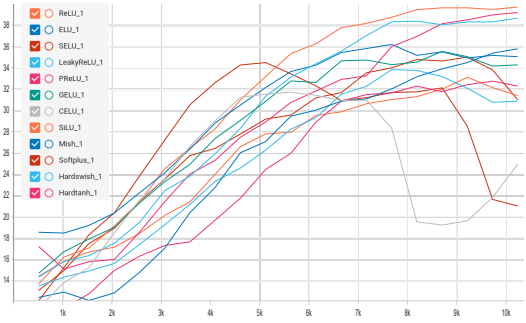

그림 3. 에이전트가 학습하는 동안 타임스탬프에 따른 에피소드 길이를 나타낸 그래프
Fig. 3. A plot showing the episode length according to timestamps while the agent was learning

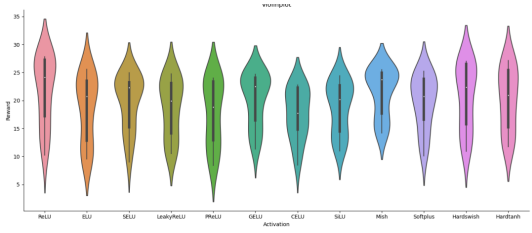

그림 2. 에이전트가 학습하는 동안 타임스탬프에 따른 보상을 나타낸 바이올린 그래프
Fig. 2. A violin plot showing rewards according to timestamps while the agent was learning

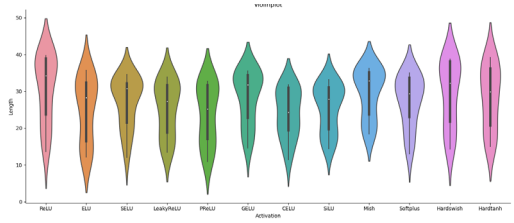

그림 4. 에이전트가 학습하는 동안 타임스탬프에 따른 에피소드 길이 나타낸 바이올린 그래프
Fig. 4. A violin plot showing the episode length according to timestamps while the agent was learning

ϵ는 하이퍼 파라미터로 0.1 또는 0.2의 값을 갖는다. 확률비 $r_t(\theta)$는 다음과 같이 정의한다.

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \tag{14}$$

또한, 에이전트가 학습할 고속도로를 시뮬레이션하기 위해 Highway-env의 highway-v0를 사용하였다. 이 환경에서 자동차가 취할 수 있는 액션은 감속, 가속, 등속, 차선 변경 4가지이다. 고속도로 환경에서 하나의 에피소드는 자동차가 목적지까지 도달하거나 다른 자동차와 충돌했을 때 종료된다. 보상은 속도가 빠를수록 증가하고 다른 차와 많이 충돌할수록 작아진다. 자동차 속도 v, 최저 속도 $v_{\min}$, 최고 속도 $v_{\max}$, 충돌 횟수 c, 계수 a와 b가 주어질 경우 보상은 다음과 같이 정의하였다.

$$R(a,b) = a\frac{v - v_{\min}}{v_{\max} - v_{\min}} - b \times c \tag{15}$$

이 보상 함수의 결과는 [0, 1] 범위로 정규화하였다.

표 1. 에이전트가 학습하는 동안 평균 보상과 최대 보상에 따른 활성화 함수 순위
Table 1. The best activation functions based on average rewards and maximum rewards while learning

Ranking	Average Reward	Max Reward	Final Reward
1	ReLU	ReLU	ReLU
2	Mish	HardTanh	HardTanh
3	Hardswish	Hardswish	Hardswish
4	GELU	ELU	ELU
5	HardTanh	Mish	Mish
6	Softplus	Softplus	GELU

Ⅳ. 성능 평가

각 활성화 함수가 고속도로 자율주행 학습 시 미치는 영향을 평가하기 위해 각 활성화 함수를 사용하여 학습할 때 얻게 되는 보상을 비교하였다.

그림 1은 에이전트가 고속도로 자율주행을 학습하는 동안 10k 타임스탬프에 따른 보상을 나타낸 그래프이다. CELU와 SELU의 경우 10k 타임스탬프 지점에서 이전보다 보상이 하락하여 오버피팅 (Overfitting)된 경향을 보인다. 이 함수들은 5k 타임스탬프에서 가장 큰 보상을 보

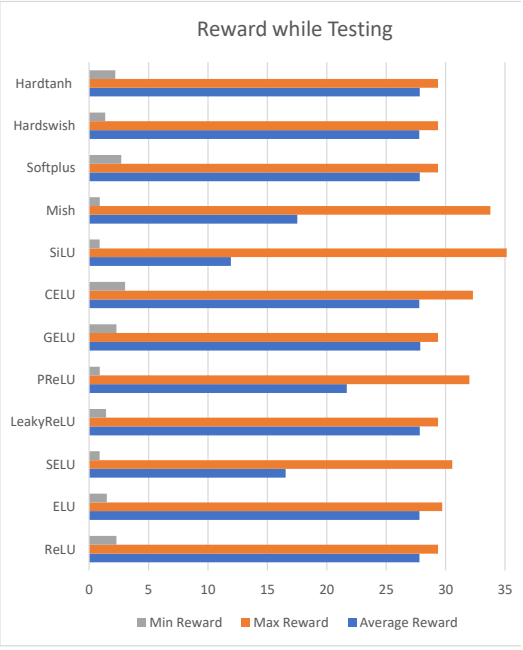

그림 5. 에이전트가 테스트하는 동안 타임스탬프에 따른 보상을 나타낸 그래프

Fig. 5. A plot showing rewards according to timestamps while the agent was testing

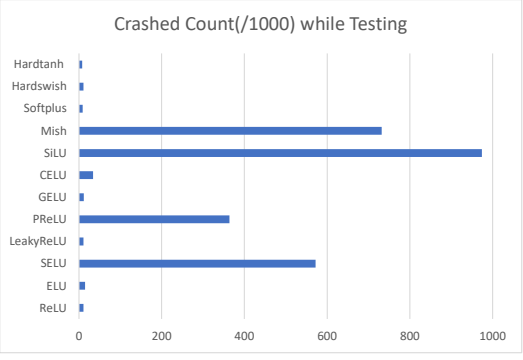

그림 6. 에이전트가 테스트하는 동안 충돌 횟수를 나타낸 그래프

Fig. 6. A plot showing the number of collisions while the agent was testing

이므로 이때 학습을 멈추었더라면 좀 더 좋은 결과를 보였을 것이다. 그림 1의 그래프를 분석한 결과, 표 1과 같이 학습하는 동안 평균 보상과 최대 보상, 최종 보상에 따른 활성화 함수 순위를 구할 수 있었다. 그 결과 평균 보상 값은 ReLU와 Mish가 가장 좋은 결과를 보여주었으며 최대 보상 값은 ReLU와 HardTanh가 우수하였다. 학습이 종료된 시점에서 최종 보상 값은 ReLU와

표 2. 에이전트가 테스트하는 동안 평균 보상과 최대 보상에 따른 활성화 함수 순위

Table 2. The best activation functions based on average rewards and maximum rewards while testing

Ranking	Average Reward	Max Reward
1	GELU	SiLU
2	Softplus	Mish
3	LeakyReLU	CELU
4	HardTanh	PReLU
5	ReLU	SELU
6	ELU	ELU

HardTanH가 우수하였다. 최대 보상과 최종 보상 순위는 5 순위까지 모두 동일하였다. 모든 경우에서 ReLU가 가장 우수한 결과를 보여주었다. 그림 2는 그림 1의 데이터를 바이올린 플롯으로 표현한 것이다. Mish의 경우 가장 분산이 적으면서 좋은 결과를 보여주었다.

그림 3과 그림 4는 학습하는 동안 한 에피소드 당 에이전트가 고속도로를 주행한 시간을 그래프와 바이올린 그래프로 나타낸 것이다. 보상을 나타낸 그래프와 거의 유사한 것을 알 수 있다. 이는 보상을 결정하는데 주행 시간이 큰 역할을 한다는 것을 알게 해 준다.

그림 5는 학습이 종료된 에이전트에 대하여 같은 고속도로 환경에서 테스트를 했을 경우 평균 보상과 최대 보상, 최소 보상을 나타낸 그래프이다. SiLU의 경우 최대 보상이 가장 컸지만, 평균 보상은 가장 작았다. 따라서 최대 보상에 대해서는 우연히 충돌 없이 좋은 환경이 생성되어 최대 보상을 얻었을 뿐 특별한 의미를 찾을 수 없었다. 표 2는 그림 5를 기반으로 평균 보상과 최대 보상의 상위 6개 활성화 함수를 나열한 것이다. 평균 보상에서 가장 좋은 성능을 보여준 활성화 함수는 GELU와 Softplus 였다. 학습할 때와 테스트 시 모두 6위 내에 들었던 활성화 함수는 GELU, Softplus, HardTanh, ReLU 였다.

그림 6은 에이전트가 테스트하는 동안 충돌 횟수를 나타낸 그래프이다. Mish, SiLU, PReLU, SELU의 경우 충돌 횟수가 다른 활성화 함수보다 현저하게 많은데 그 결과 평균 보상 순위에서도 가장 낮은 순위를 차지하게 되었다. SELU를 제외한 나머지 활성화 함수는 학습 시 오버피팅된 현상도 발생하지 않았는데 이렇게 결과가 안 좋게 나온 것은 추후 연구 및 분석이 필요하다.

V. 결 론

본 논문은 고속도로에서 자율주행 학습 시 어떤 활성화 함수를 사용하는 것이 강화 학습에 유리한지 비교 평가하였다. 12가지 활성화 함수의 성능을 평가해본 결과 GELU와 Softplus가 평균적으로 높은 보상을 보여주었으며 SELU와 SiLU는 가장 낮은 보상을 보여주었다. 가장 높은 보상을 주었던 GELU와 가장 낮았던 SiLU와의 차이는 20%였다.

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. 또한, 일부 활성화 함수에서 충돌이 과도하게 발생하는 이유에 대하여 알아볼 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] M. Ok, "A Simulator Implementation of Highway Driving Guidance System for Longitudinal Autonomous Driving of ADAS-Driving Vehicles", The Journal of Korean Institute of Information Technology, Vol 17, No. 11, pp. 27-35, 2019.
DOI : https://doi.org/10.14801/jkiit.2019.17.11.27

[3] 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

[4] 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.

[5] X. Glorot, A. Bordes, Y. Bengio, "Deep sparse rectifier neural networks", International Conference on Artificial Intelligence and Statistics, 2011.

[6] 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

[7] D. Clevert, T. Unterthiner, S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)", arXiv:1511.07289, 2015.

[8] K. He, X. Zhang, S. Ren, J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification", IEEE International Conference on Computer Vision, 2015.
DOI: https://doi.org/10.1109/iccv.2015.123

[9] D. Hendrycks, K. Gimpel, "Gaussian Error Linear Units", arXiv:1606.08415, 2016.

[10] G. Klambauer, T. Unterthiner, A. Mayr, S. Hochreiter, "Self-Normalizing Neural Networks". Advances in Neural Information Processing Systems", arXiv:1706.02515, 2017.

[11] B. Jonathan, "Continuously Differentiable Exponential Linear Units", arXiv:1704.07483, 2017.

[12] 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

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

[14] H. Andrew, S. Mark, G. Chu, L. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q.V. Le, H. Adam, "Searching for MobileNetV3", IEEE/CVF international conference on computer vision, 2019.

[15] 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