

<https://doi.org/10.7236/JIIBC.2022.22.6.75>
JIIBC 2022-6-12

자율주행 자동차의 주차를 위한 강화학습 활성화 함수 비교 분석

A Comparative Analysis of Reinforcement Learning Activation Functions for Parking of Autonomous Vehicles

이동철*

Dongcheul Lee*

요 약 주차 공간의 부족함을 획기적으로 해결할 수 있는 자율주행 자동차는 심층 강화 학습을 통해 큰 발전을 이루고 있다. 심층 강화 학습에는 활성화 함수가 사용되는데, 그동안 다양한 활성화 함수가 제안되어 왔으나 적용 환경에 따라 그 성능 편차가 심했다. 따라서 환경에 따라 최적의 활성화 함수를 찾는 것이 효과적인 학습을 위해 중요하다. 본 논문은 자율주행 자동차가 주차를 학습하기 위해 심층 강화 학습을 사용할 때 어떤 활성화 함수를 사용하는 것이 가장 효과적인지 비교 평가하기 위해 강화 학습에 주로 사용되는 12개의 함수를 분석하였다. 이를 위해 성능 평가 환경을 구축하고 각 활성화 함수의 평균 보상을 성공률, 에피소드 길이, 자동차 속도와 비교하였다. 그 결과 가장 높은 보상은 GELU를 사용한 경우였고, ELU는 가장 낮았다. 두 활성화 함수의 보상 차이는 35.2%였다.

Abstract Autonomous vehicles, which can dramatically solve the lack of parking spaces, are making great progress through deep reinforcement learning. Activation functions are used for deep reinforcement learning, and various activation functions have been proposed, but their performance deviations were large depending on the application environment. Therefore, finding the optimal activation function depending on the environment is important for effective learning. This paper analyzes 12 functions mainly used in reinforcement learning to compare and evaluate which activation function is most effective when autonomous vehicles use deep reinforcement learning to learn parking. To this end, a performance evaluation environment was established, and the average reward of each activation function was compared with the success rate, episode length, and vehicle speed. As a result, the highest reward was the case of using GELU, and the ELU was the lowest. The reward difference between the two activation functions was 35.2%.

Key Words : Autonomous Vehicle, Parking, Reinforcement Learning

*중신회원, 한남대학교 멀티미디어공학과
접수일자 2022년 11월 2일, 수정완료 2022년 11월 30일
게재확정일자 2022년 12월 9일

Received: 2 November, 2022 / Revised: 30 November, 2022 /
Accepted: 9 December, 2022
Corresponding Author: jackdlee@hnu.kr
Hannam University, Korea

I. 서 론

일반적인 차량은 대부분의 시간을 주차장에서 소비한다. 이를 위해 도시는 많은 공간을 주차장으로 할애해야 하고 이로 인해 삶에 필수적인 주거 공간 및 녹지 공간은 줄어드는 실정이다. 이러한 주차장 팽창 문제를 자율주행 자동차 (Autonomous Vehicle)를 통해 해결할 수 있다^[1]. 자율주행차는 스스로 주차하므로 주차 시 문을 열고 걸어 나갈 공간이 필요 없다. 타워 식 주차장의 경우 승객을 실어 나를 엘리베이터와 계단도 필요 없다. 승객들은 주차장에 내린 후 목적지까지 걸어갈 필요 없이 목적지에 바로 내린 후 차량이 스스로 주차하도록 하면 된다. 또한 사용하지 않는 차량은 주차 공간을 절약하기 위해 다른 차량과 겹쳐서 주차할 수도 있다. 나중에 출차가 필요한 경우 막고 있는 다른 차량들을 자동으로 이동시키고 내 차가 나오도록 하면 되기 때문이다. 자율주행차는 일반적인 사람보다 더 빠르게 반응하고 더 정확히 주행할 수 있어서 좁은 도로에서도 주위 환경과 여유 공간 없이 주행할 수 있으며 주차장을 빠져나올 때도 최소한의 공간만 제공되면 된다.

자율주행차는 인지, 의사 결정, 제어의 세 단계로 스스로 차량을 움직인다. 인지 단계에서는 카메라, 레이더 (Radar), 라이다 (Lidar)와 같은 외부환경 인지 센서를 사용한다. 의사 결정 단계에서는 인지 단계에서 인식한 개체에 대하여 어떤 액션을 취할지 결정한다. 핸들 방향을 바꿀지, 속도를 변경할지, 후진할지, 멈출지 등이 고려된다. 의사 결정 알고리즘으로는 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 일일이 정해주는 방식이다. 그러나 주행의 모든 상황을 고려하는 규칙을 만들기가 사실상 어렵기 때문에 이 방식 단독으로는 사용되지 않는다. 지도 학습 방식은 대량의 운전 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 학습 시키는 것이다. 보상을 어떻게 계산할지 설정하느냐에 따라 특정 목적에 맞는 정책을 학습시키는 것이 가능하다. 제어 단계에서는 의사 결정 단계에서 결정된 결과에 따라 실제로 차량을 제어한다.

지도 학습이나 강화 학습은 신경망 (Neural Network)을 이용한다. 특히 많은 계층 (Layer)을 사용한 심층 신경망

의 경우 각 계층을 결합하기 위해 활성화 함수 (Activation Function)가 필요하다. 지금까지 제안된 활성화 함수는 여러 가지가 있는데 어떠한 활성화 함수를 사용하느냐에 따라 학습의 결과가 크게 달라진다. 모든 환경에서 일반적으로 우수한 활성화 함수는 아직 발견되지 않았으며 각 상황에 맞게 적절한 활성화 함수를 선택해야 한다^[4]. 본 논문은 자율주행 자동차가 주차 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

활성화 함수는 신경망 계층의 선형 입력에 비선형성을 도입하는 역할을 한다. 따라서 적절한 활성화 함수를 선택하는 것은 강화 학습의 성능을 올리는 데 큰 역할을 한다. 신경망의 계층에서 활성화 함수를 적용하는 방법은 다음 식을 적용하는 것이다.

$$g(y) = g\left(\sum_{i=0}^n w_i x_i + b\right) \quad (1)$$

여기서 x_i 는 신경망 계층의 입력값이고, $g(y)$ 는 출력값이 된다.

신경망 학습 초기에 사용된 활성화 함수는 역전파 (Backpropagation)을 사용할 수 있도록 미분이 가능한 시그모이드 (Sigmoid) 함수와 하이퍼볼릭 탄젠트 (Tanh) 함수였다. 그러나 신경망이 깊어짐에 따라 기울기 소멸 문제 (Vanishing gradient problem)이 발생하여 더는 활용되지 못하였다^[5].

이 문제를 해결하기 위해 표 1과 같은 함수들이 제안되었으며 그중에서도 가장 먼저 Rectified Linear Unit (ReLU) 함수가 제안되었다. 이 함수는 계산하기 효율적이며 비선형성으로 인해 네트워크를 빨리 수렴할 수 있게 한다. 그러나 입력이 0에 가까워지거나 음수가 되면 기울기 정보가 손실되어 신경망이 역전파를 수행하기 어렵고 학습을 더는 할 수 없게 되는 Dying ReLU 문제가 발생한다^[6]. 이 문제를 해결하기 위해 다음과 같은 활성화 함수들이 제안되었다.

Softplus 함수는 로지스틱 함수로 부드러운 버전의 ReLU라고 할 수 있으며 출력을 항상 양수로 제한한다^[7].

표 1. 활성화 함수 종류

Table 1. Types of activation functions

Activation Function	Equation
ReLU	$f(x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases}$
Softplus	$f(x) = \ln(1 + \exp^x)$
Leaky ReLU	$f(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha x, & \text{if } x \leq 0 \end{cases}$
ELU	$f(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha \exp(x) - 1, & \text{if } x \leq 0 \end{cases}$
PReLU	$f(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha x, & \text{if } x \leq 0 \end{cases}$
GELU	$f(x) = \frac{x}{2} \left[1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right) \right]$
SELU	$f(x) = \begin{cases} \lambda x, & \text{if } x \geq 0 \\ \lambda \alpha (\exp(x) - 1), & \text{if } x < 0 \end{cases}$
CELU	$f(x) = \begin{cases} x, & \text{if } x \geq 0 \\ \alpha (\exp(\frac{x}{\alpha}) - 1), & \text{if } x < 0 \end{cases}$
SiLU	$f(x) = \frac{x}{1 + e^{-x}}$
Mish	$f(x) = x \operatorname{Tanh}(\ln(1 + e^x))$
Hardswish	$f(x) = \begin{cases} 0, & \text{if } x \leq -3 \\ x, & \text{if } x \geq 3 \\ x(x+3)/6, & \text{otherwise} \end{cases}$
HardTanh	$f(x) = \begin{cases} -1, & \text{if } x < -1 \\ 1, & \text{if } x > 1 \\ x, & \text{otherwise} \end{cases}$

Leaky ReLU 함수는 ReLU의 변형으로 음의 영역에 약간의 경사를 더해 Dying ReLU 문제를 해결하여 음의 입력값에도 역전파가 가능하다^[8]. Exponential Linear Unit (ELU) 함수는 ReLU와 Leaky ReLU의 장점을 결합하였다. 배치 정규화와 같이 큰 음의 값을 비활성화시켜 평균 단위 활성화를 0에 가깝게 할 수 있지만, 계산 복잡도는 더 낮은 장점을 가진다^[9]. Parametric ReLU (PReLU) 함수는 가장 간결하게 Dying ReLU 문제를 해결할 수 있는 방법 중 하나이며 Leaky ReLU와 비슷한 형태를 지닌다. 이 활성화 함수는 ReLU를 발전시켜 서로 다른 신경망 계층은 서로 다른 비선형 활성화 함수를 필요로 할 것이라는 의미에서 제안되었다^[10]. 표준 가우스 누적 분포 함수를 사용하는 Gaussian Error Linear Unit (GELU) 함수는 Google의 BERT나 OpenAI의

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

Table 2. Ranking of activation functions based on average reward and maximum reward while learning

Ranking	Average Reward	Max Reward
1	CELU	Mish
2	SELU	GELU
3	ELU	CELU
4	Hardswish	ELU
5	Mish	PReLU
6	SiLU	Hardswish

GPT-2에서 사용된다^[11]. 비선형성은 ReLU처럼 부호로 결정하는 것이 아닌 백분위 수로 입력 가중치를 부여한다. Scaled Exponential Linear Unit (SELU) 함수는 자기 정규화를 유도하는 특징을 지니므로 각 계층이 이전 계층으로부터 평균과 분산을 보존할 수 있다^[12]. 또한 ReLU에서는 불가능했던 양의 값과 음의 값을 모두 가지므로 평균을 이동하는 것이 가능하다. Continuously Differentiable Exponential Linear Units (CELU) 함수는 C1 연속하며 알파가 1이 아니면 연속적으로 미분 가능하다^[13]. Sigmoid-Weighted Linear Units (SiLU) 함수는 시그모이드 함수의 입력값을 곱한 값으로 계산한다^[14]. Mish 함수는 스스로 정규화하는 비단조 함수로써 입력 값과 softplus 함수를 tanh 함수 입력으로 사용한 결과를 곱하여 활성화를 계산한다^[15]. Hardswish 함수는 비선형 버전의 swish로써 계산이 더 빠르고 메모리 접근 수를 줄여 지연 비용을 낮추는 장점이 있다^[16]. Hard Tanh 함수는 Tanh 함수의 변형으로 계산이 보다 빠른 장점이 있다.

III. 성능 평가 방법

자율주행 자동차가 주차를 학습할 때 신경망에 사용할 활성화 함수를 성능 평가하기 위해 Truncated Quantile Critics (TQC)를 학습 알고리즘으로 사용하는 심층 강화 학습 에이전트를 제작하였다. TQC는 Off-policy 학습 시 발생하는 과대평가 편향 (Overestimation bias) 문제를 완화할 수 있는 알고리즘으로 값 함수에 대한 분포를 예측하는데 평균값 대신 양자 회귀 분석을 사용한다^[17]. TQC 목적 분포 함수 정의를 위해 사용할 표기법은 다음과 같이 정의한다. 에이전트는 연속적 상태 $s \in S$ 에서 정책 π 에 따라 특정 액션 $a \in A$ 를 취하여 보상

$r \in R$ 을 받고 다음 상태로 $p(s' | s)$ 확률로 이동하고, 할인 인자 (Discount factor) γ 는 $[0,1]$ 의 범위를 가지는 (S, A, P, R, γ) 튜플의 Markov 결정 과정을 거친다. N 개의 근사치 $Z_{\psi_1}, \dots, Z_{\psi_N}$ 를 학습할 때 목적 분포 함수는 다음과 같다.

$$Y(s, a) = \frac{1}{kN} \sum_{i=1}^{kN} \delta(y_i(s, a)) \quad (2)$$

여기서 $y_i(s, a)$ 는 다음과 같이 정의한다.

$$y_i(s, a) = r(s, a) + \gamma[z_{(i)}(s', a') - \alpha \log \pi_{\emptyset}(a' | s')] \quad (3)$$

여기서 \emptyset 는 정책 파라미터이며 α 는 엔트로피 온도 계수이다. $z_{(i)}(s', a')$ 는 각 근사치 Z_{ψ_n} 를 오름차순으로 정렬했을 때 각 원소를 뜻하며 Z_{ψ_n} 은 다음과 같이 정의한다.

$$Z_{\psi_n}(s, a) = \frac{1}{M} \sum_{m=1}^M \delta(\theta^m(s, a)) \quad (4)$$

정책 파라미터 \emptyset 를 최적화할 Loss 함수는 다음과 같이 정의한다.

$$J_{\pi}(\emptyset) = E_{D, \pi} \left[\alpha \log \pi_{\emptyset}(a | s) - \frac{1}{NM} \sum_{m,n=1}^{M,N} \theta^m(s, a) \right] \quad (5)$$

이 에이전트는 주차 환경을 학습하기 위해 Highway-env의 parking-v0를 사용하였다^[18]. 학습 기간은 각 활성화 함수마다 5만 번의 타임스텝을 거쳤다. 이 환경에서 사용할 수 있는 액션은 연속적 액션으로, 가속페달 조절과 운전대 조절이다. 에피소드 종료 조건은 자동차를 적절한 방향으로 지정된 위치에 위치 시켜 성공하거나, 주위에 충돌하여 실패하는 것이다. 에이전트의 상태 s , 목표 상태 s_g , 충돌 횟수 c , 충돌 계수 b 가 주어질 경우 p-norm을 이용한 보상 $R(s, a)$ 는 다음과 같이 정의하였다.

$$R(s, a) = - \left[\sum_i^n (W_i |s - s_g|)^p \right] - b \times c \quad (6)$$

IV. 성능 평가

자동차가 자동 주차 학습 시 활성화 함수가 끼치는 영향을 비교 평가하기 위해 그림 1과 같이 학습하는 동안 타임스텝에 따른 보상을 그래프로 비교하였다. 학습되는 속도는 활성화 함수마다 차이가 크게 발생하였으나

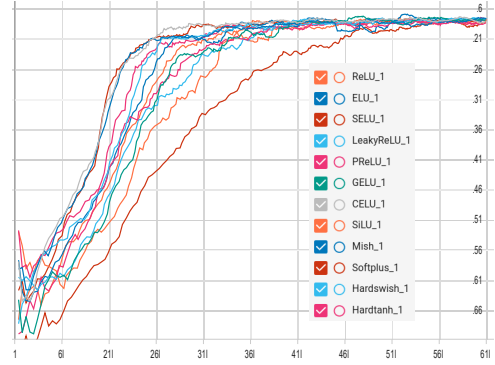


그림 1. 에이전트가 학습하는 동안 타임스텝에 따른 보상을 나타낸 그래프

Fig. 1. A plot showing rewards according to timestamps while the agent was learning

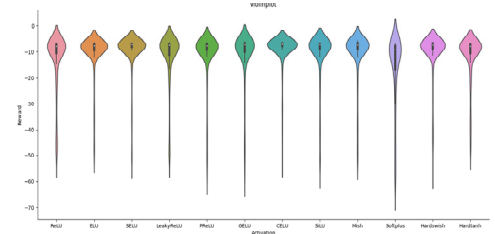


그림 2. 에이전트가 학습하는 동안 타임스텝에 따른 보상을 나타낸 바이올린 그래프

Fig. 2. A violin plot showing rewards according to timestamps while the agent was learning

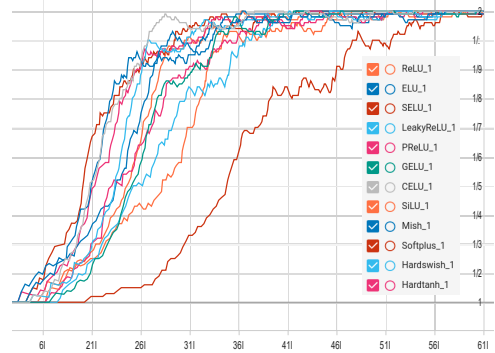


그림 3. 에이전트가 학습하는 동안 타임스텝에 따른 성공률을 나타낸 그래프

Fig. 3. A plot showing the success rate according to timestamps while the agent was learning

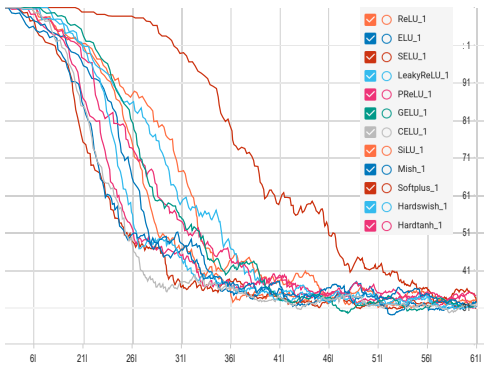


그림 4. 에이전트가 학습하는 동안 타임스텝에 따른 에피소드 길이를 나타낸 그래프
 Fig. 4. A plot showing the episode length according to timestamps while the agent was learning

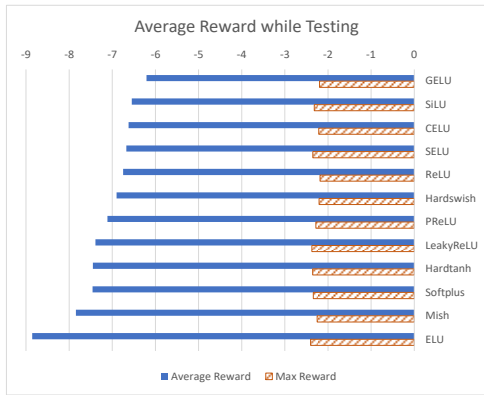


그림 5. 에이전트가 테스트하는 동안 평균 및 최대 보상을 나타낸 그래프
 Fig. 5. A plot showing mean and max rewards while the agent was testing

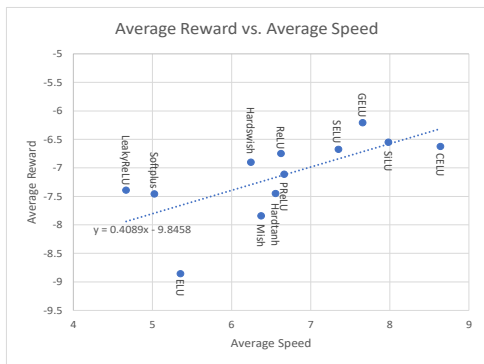


그림 6. 에이전트가 테스트하는 동안 평균 보상과 평균 속도를 나타낸 그래프
 Fig. 6. A plot showing the number of collisions while the agent was testing

표 3. 에이전트가 테스트하는 동안 평균 보상과 최대 보상에 따른 활성화 함수 순위

Table 3. Ranking of activation functions based on average reward and maximum reward during the test

Ranking	Average Reward	Max Reward
1	GELU	ReLU
2	SiLU	GELU
3	CELU	Hardswish
4	SELU	CELU
5	ReLU	Mish
6	Hardswish	PRelu

학습이 끝나는 50k 타임스텝에서는 대부분 비슷한 결과를 보여주었다. 표 2는 평균 및 최대 보상이 가장 높은 6개 함수를 보여준다. 평균 보상은 CELU가 가장 높았으며 Softplus가 가장 낮았고 둘의 차이는 6.74 (51.7%)였다. 최대 보상은 근소한 차이로 Mish가 가장 높았고 Hardtanh가 가장 낮았으며 둘의 차이는 1.08 (16.6%)이었다. 그림 2는 그림 1의 데이터를 바이올린 플롯으로 보여준다. 대부분의 함수는 분산이 작게 분포하나 가장 평균 보상이 낮았던 Softplus의 경우 분산이 크게 분포하는 것을 알 수 있다.

에이전트의 학습 시 보상에 영향을 주는 요인으로 주차 성공률과 한 에피소드를 끝내는 데 들었던 시간의 길이를 들 수 있다. 그림 3과 그림 4는 각각 활성화 함수별 주차 성공률과 에피소드 길이를 나타낸 그래프이다. 성공률의 경우 활성화 함수 차이에 따라 그 차이가 보상보다 더 명확하게 드러났으며 그림 1과 그림 2에서 가장 낮은 결과를 보여준 Softplus와 ReLU가 성공률에서도 가장 낮은 결과를 보여주었다. 그러나 학습이 종료되는 50k 타임스텝 부근에서는 대부분 비슷한 결과를 보여주었다. 에피소드 길이에서도 Softplus가 다른 함수들보다 학습이 늦게 되는 결과를 보여주었으며 학습 종료 시점에는 대부분이 비슷한 결과를 보였다. 이를 통해 활성화 함수 종류에 따라 학습하는 속도는 서로 다르지만, 일정한 타임스텝을 거둬하면 서로 비슷한 학습 결과를 내는 것을 알 수 있다.

그림 5는 각 활성화 함수별로 학습을 완료한 에이전트에 대하여 100번 주차 테스트 시 평균 보상과 최대 보상을 나타낸 그래프이다. 여기서 평균 및 최대 보상이 가장 높은 6개 함수가 표 3에 정리되었다. 평균 보상은 GELU가 가장 높았으며 ELU가 가장 낮았고 둘의 차이는 2.65 (35.2%)였다. 최대 보상은 ReLU가 가장 높았으며 ELU

가 가장 낮았고 둘의 차이는 0.22 (9.6%)였다. 학습을 빨리 시켰던 활성화 함수와 테스트 시 보상이 우수한 함수 간에 상관관계는 발견하기 어려웠다.

그림 6은 테스트 시 에이전트의 평균 보상과 평균 자동차 속도 간의 상관관계를 나타내는 그래프이다. 평균 속도는 전진 속도와 후진 속도의 절대값을 평균하여 구하였다. 대체로 속도가 빠른 에이전트가 보상이 높은 결과를 얻을 수 있었으며 추세선의 기울기는 0.4였다.

V. 결 론

본 논문은 자율주행 자동차가 강화 학습을 통해 주차를 학습할 때 어떤 활성화 함수를 사용하는 것이 유리한지 비교 평가하였다. 강화 학습에 주로 사용되는 12개의 활성화 함수를 주차 학습 시 사용하여 비교해 본 결과 학습을 가장 빠른 함수는 CELU였고, Softplus가 가장 늦었다. 둘의 평균 보상 차이는 6.74 (51.7%)였다. 학습 완료된 에이전트의 평균 보상은 GELU가 가장 높았으며 ELU가 가장 낮았고 둘의 차이는 2.65 (35.2%)였다.

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

References

- [1] D. Shim, J. Yang, J. Son, S. Han, H. Lee, "Smart Parking Guidance System based on IoT Car-stoppers," *The Journal of The Institute of Internet, Broadcasting and Communication*, Vol. 17, No. 3, pp. 137-143, 2017.
DOI: <https://doi.org/10.7236/IIBC.2017.17.3.137>
- [2] D. Jeon, D. Park, "Malware Detection in Encrypted TLS Traffic using Machine Learning Techniques," *Journal of KIIT*, Vol. 19, No. 10, pp. 125-136, 2021.
DOI: <http://dx.doi.org/10.14801/jkiit.2021.19.10.125>
- [3] H. Cho, H. Shin, "Trading Strategies Using Reinforcement Learning," *Journal of the Korea Academia-Industrial cooperation Society*, Vol. 22, No. 1, pp. 123-130, 2021.
DOI: <https://doi.org/10.5762/KAIS.2021.22.1.123>
- [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] M. Roodschild, J.G. Sardiñas, A. Will, "A new approach for the vanishing gradient problem on sigmoid activation," *Progress in Artificial Intelligence*, Vol. 9, pp. 351-360, 2020.
DOI: <https://doi.org/10.1007/s13748-020-00218-y>
- [6] A. Isac, C. J. Frederico, D. Kragic, J.A. Stork, "The effect of Target Normalization and Momentum on Dying ReLU," *Proceedings of the 32nd annual workshop of the Swedish Artificial Intelligence Society*, 2020.
DOI: <https://doi.org/10.48550/arXiv.2005.06195>
- [7] X. Glorot, A. Bordes, Y. Bengio, "Deep sparse rectifier neural networks," *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, PMLR, Vol. 15, pp. 315-323, 2011.
- [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. Clevert, T. Unterthiner, S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," *Proceedings of 4th International Conference on Learning Representations (ICLR)*, 2016.
DOI: <https://doi.org/10.48550/arXiv.1511.07289>
- [10] K. He, X. Zhang, S. Ren, J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *Proceedings of IEEE International Conference on Computer Vision*, 2015.
DOI: <https://doi.org/10.1109/iccv.2015.123>
- [11] D. Hendrycks, K. Gimpel, "Gaussian Error Linear Units," *arXiv:1606.08415*, 2016.
DOI: <https://doi.org/10.48550/arXiv.1606.08415>
- [12] G. Klambauer, T. Unterthiner, A. Mayr, S. Hochreiter, "Self-Normalizing Neural Networks," *Proceedings of 31st Conference on Neural Information Processing Systems (NIPS)*, 2017.
DOI: <https://doi.org/10.48550/arXiv.1706.02515>
- [13] B. Jonathan, "Continuously Differentiable Exponential Linear Units," *arXiv:1704.07483*, 2017.
DOI: <https://doi.org/10.48550/arXiv.1704.07483>
- [14] S. Elfving, 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>
- [15] D. Misra, "Mish: A Self Regularized Non-Monotonic Activation Function", *arXiv:1908.08681*, 2020.
DOI: <https://doi.org/10.48550/arXiv.1908.08681>
- [16] 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," *Proceedings of IEEE/CVF international conference on*

computer vision, 2019.

DOI: <https://doi.org/10.48550/arXiv.1905.02244>

- [17] A. Kuznetsov, P. Shvechikov, A. Grishin, D. Vetrov, "Controlling Overestimation Bias with Truncated Mixture of Continuous Distributional Quantile Critics," Proceedings of the 37 th International Conference on Machine Learning (PMLR), 2020.
DOI: <https://doi.org/10.48550/arXiv.2005.04269>
- [18] Leurent, Edouard, "An Environment for Autonomous Driving Decision-Making," GitHub repository: <https://github.com/eleurent/highway-env>, 2018.

저 자 소 개

이 동 철(중신회원)



- 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

※ 이 논문은 2022학년도 한남대학교 학술연구비 지원에 의하여 연구되었음