

컨벌루션 신경망에서 활성화 함수가 미치는 영상 분류 성능 비교

박성욱*, 김도연**

Comparison of Image Classification Performance by Activation Functions in Convolutional Neural Networks

Sung-Wook Park^{*}, Do-Yeon Kim^{**}

ABSTRACT

Recently, computer vision application is increasing by using CNN which is one of the deep learning algorithms. However, CNN does not provide perfect classification performance due to gradient vanishing problem. Most of CNN algorithms use an activation function called ReLU to mitigate the gradient vanishing problem. In this study, four activation functions that can replace ReLU were applied to four different structural networks. Experimental results show that ReLU has the lowest performance in accuracy, loss rate, and speed of initial learning convergence from 20 experiments. It is concluded that the optimal activation function varied from network to network but the four activation functions were higher than ReLU.

Key words: Deep Learning, Computer Vision, Convolutional Neural Network, Activation Function

1. 서 론

2016년 구글 딥마인드(Google DeepMind)가 개발한 인공지능(Artificial Intelligence, AI) 바둑프로그램 알파고(AlphaGo)[1]의 등장 이후 일반에 딥러닝(Deep Learning)[2]이라는 단어가 급부상하기 시작했다. 딥러닝이 바로 알파고의 핵심 기술이기 때문이다. 그런데 딥러닝이 비단 바둑프로그램에만 사용되는 것은 아니다. 딥러닝은 매우 다양한 분야에 사용되고, 그 중 가장 폭넓게 활용되는 분야는 패턴 인식(Pattern Recognition)[3]을 통한 컴퓨터 비전(Computer Vision)[4]이다.

패턴을 파악해 여러 클래스로 구분 짓는 작업을

분류라고 한다. ILSVRC는 이미지넷(ImageNet)[5]이 제공하는 1,000여 클래스로 분류된 100만 개의 이미지를 인식하여 그 정확도를 겨루는 대회이다. 이 대회에서 2011년까지는 이미지 인식률이 75%를 넘지 못하였는데, 2012년 대회에서 딥러닝 알고리즘의 기반을 둔 컨벌루션 신경망(Convolutional Neural Network, CNN)[6], AlexNet이 무려 Top-5 정확도 84.7%라는 놀라운 인식률을 달성했다. 그 이후부터는 CNN을 이용한 팀들이 상위 랭크를 모두 차지하였고, 매년 인식률은 높아졌다. 최근에는 상당수의 도전자가 97%에 육박하는 인식률을 기록하고 있으며 이는 인간의 인식률인 95%를 웃도는 수준이다.

CNN을 구성하고 있는 컨벌루션 계층(Convolu-

* Corresponding Author : Do-Yeon Kim, Address: (57922) Jungang-ro 255, Suncheon, Jeonnam, Korea, TEL : +82-61-750-3628, FAX : +82-61-750-3620, E-mail : dykim@sunchon.ac.kr

Receipt date : Jul. 25, 2018, Revision date : Aug. 28, 2018
Approval date : Aug. 29, 2018

^{*} Dept. of Computer Eng., Sunchon National University
(E-mail : park7231654@naver.com)

^{**} Dept. of Computer Eng., Sunchon National University
* This paper was supported by Sunchon National University Research Fund III in 2018.

tion Layer), 풀링 계층(Pooling Layer) 그리고 완전 연결 계층(Fully-Connected Layer)의 개념은 매우 간단하다. 컨벌루션 계층과 풀링 계층은 2차원 평면 행렬에서 지정 영역의 값들을 단일값으로 압축하는 것이다. 단, 단일값으로 압축할 때 컨벌루션 계층은 가중치(Weight)와 바이어스(Bias)를 적용하고, 풀링 계층은 단순히 후보값들 중 하나를 선택해서 가져 오는 방식을 취한다. 풀링 기법으로는 주로 최대 풀링(Max Pooling)을 사용한다. 완전연결 계층은 인접한 계층의 모든 뉴런과 상호 연결된 계층을 의미한다.

CNN 역시 완벽한 분류 성능은 내지 못한다. CNN의 성능이 저하되는 이유 중 하나는 기울기 소실(Gradient Vanishing)에 있다. 기울기 소실 문제를 해결하기 위해, 활성 함수(Activation Function) 측면에서 정규화 선형 유닛(Rectified Linear Unit, ReLU)[7]이 등장하였지만 이 함수도 문제점은 존재한다. 이를 해결하기 위해, 최근 여러 활성 함수가 제안되었다. 하지만 아직도 대부분의 네트워크 설계자들은 활성 함수로 ReLU를 채택하여 사용하고 있다. 이에 본 연구에서는 최근 발표된 4가지 활성 함수 및 ReLU를 서로 다른 4개 구조 네트워크에 적용하여 실험해보고 위 함수 각각이 분류 성능에 어떤 영향을 미치는지 알아보려고 한다.

본 논문에서는 네트워크 및 활성 함수에 따라 CNN의 분류 성능이 어떻게 변화하는지 비교하고 분석하였다. 학습에 사용된 네트워크 모델은 Lin et al.[8]의 네트워크 속의 네트워크(Network In Network, NIN), Ioffe et al. [9]의 인셉션-배치정규화(Inception-Batch Normalization, Inception-BN), Zagoruyko et al.[10]의 넓은 잔류 네트워크(Wide Residual Network, WRN) 그리고 Huang et al. [11]의 밀집하게 연결된 컨벌루션 망(Dense Connected Convolutional Network, DenseNet)이고, 각 네트워크 은닉 계층(Hidden Layer)의 활성 함수 및 분류 계층(Classification Layer)의 출력 클래스 수를 변형시켜 사용하였다. 구동을 위한 딥러닝 프레임워크로 공개 소스 Caffe[12]를 이용하였다.

논문의 2장에서는 연구된 활성 함수 구조에 대한 설명을 기술하였고 3장에서는 학습 데이터 세트 및 실험 환경 구성에 대한 설명을 기술하였다. 4장에서는 실험 결과에 대한 설명을 기술하였고 마지막 5장에서는 결론 및 향후과제에 대하여 기술하였다.

2. 관련 연구

2.1 Leaky ReLU

ReLU는 함수가 활성화 되지 않을 때 가중치를 조정하지 않기 때문에 기울기가 0이 된다. 따라서 학습 속도가 느려지는 현상이 발생한다. 이런 문제를 완화시키기 위해, Mass et al.[13]은 다음과 같이 누출 정규화 선형유닛(Leaky Rectified Linear Unit, Leaky ReLU)을 제안했다.

$$a_{i,j,k} = \max(z_{i,j,k}, 0) + \lambda \min(z_{i,j,k}, 0) \quad (1)$$

식 (1)에서 λ 는 0부터 1의 범위를 가지는 하이퍼파라미터(Hyper-Parameter)이다. Leak ReLU는 음수 부분을 ReLU처럼 상수 '0'으로 매핑하지 않고 압축하므로 함수가 비활성화 상태여도 '1'보다 작은 양수 값이 출력된다.

2.2 Parametric ReLU

Leaky ReLU에서 λ 는 학습 이전에 사용자가 정의해주는 하이퍼파라미터이지만, 매개변수 정규화 선형유닛(Parametric Rectified Linear Unit, PReLU)에서 λ 는 스스로 학습하여 값을 조정하는 내부 파라미터이다. He et al.[14]은 PReLU를 다음과 같이 제안했다.

$$a_{i,j,k} = \max(z_{i,j,k}, 0) + \lambda_k \min(z_{i,j,k}, 0) \quad (2)$$

식 (2)에서 λ_k 는 k 번째 채널에서 학습된 매개변수이다. PReLU는 훈련에 매개변수를 추가로 도입한다. 이때, 추가된 매개변수의 수는 해당 네트워크 채널수와 동일하다. [14]논문 실험에 의하면 매개변수 수가 늘어난다 하여, 오버피팅(Overfitting) 위험이 커지는 것은 아니며 추가 계산 비용 역시 무시할 수 있는 정도이다.

2.3 Randomized ReLU

무작위 누출 정규화 선형유닛(Randomized Leaky Rectified Linear Unit, RReLU)[15]의 음수부분 매개변수는 학습 시 균일 분포에서 무작위로 샘플링된 후 테스트에서 고정된다. RReLU 함수는 다음과 같이 정의된다.

$$a_{i,j,k}^{(n)} = \max(z_{i,j,k}^{(n)}, 0) + \lambda_k \min(z_{i,j,k}^{(n)}, 0) \quad (3)$$

식 (3)에서 $z_{i,j,k}^{(n)}$ 는 n 번째 예제 k 번째 채널에서 위치 (i,j) 활성 함수로의 입력을 나타내고, $\lambda_{i,j,k}^{(n)}$ 는 샘플링 된 매개변수를 나타내며, $a_{i,j,k}^{(n)}$ 은 최종 출력을 나타낸다.

2.4 ELU

Clevert et al. [16]는 심층 신경망(Deep Neural Network)의 학습 속도 및 분류 정확도를 높이는 지수함수형 선형 유닛(Exponential Linear Unit, ELU)을 제안한다. ELU는 양수부분에서 입력값을 정제 없이 출력하여 기울기 소실 문제를 방지 한다. LReLU, PReLU, RReLU는 함수 그래프의 음수 부분이 포화화(Unsaturation) 형태인 반면 ELU는 포화(Saturation) 형태를 띈다.

$$a_{i,j,k} = \max(z_{i,j,k}, 0) + \min(\lambda(e^{z_{i,j,k}} - 1), 0) \quad (4)$$

식 (4)에서 λ 는 음수값에 대해 해당 함수의 포화현상을 제어하기 위해 정의된 하이퍼파라미터이다.

3. 학습 데이터베이스 및 실험을 위한 환경 구성

3.1 훈련 및 검증 데이터베이스 구성

실험데이터는 주변에서 쉽게 볼 수 있는 산림 곤충을 대상으로 선정했다. 이후 ImageNet에서 27개를 선택하고 각 클래스에 해당하는 이미지들을 수집하여 데이터셋으로 구성하였다. 이미지 세부 항목은 Table 1과 같다.

본 논문에선 총 35,100장의 이미지를 가지고 실험을 진행하였고, 학습 데이터 세트(Training Data Set)로 29,700장, 검증 데이터 세트(Validation Data Set)로 5,400장이 사용되었다.

3.2 실험 환경 구성

네 가지 CNN 모델(NIN, Inception-BN, WRN, DenseNet)과 다섯 가지 활성 함수(ReLU, LReLU, PReLU, RReLU, ELU)를 조합하여 총 20가지의 실험

Table 1. Image subsection

Image Size	Image Cleaning	Train Set	Validation Set
256 × 256	Crop	1,300	200

환경을 구성하였다. NIN은 분류 계층의 완전연결 구조를 전역 평균 풀링(Global Average Pooling) 구조로 대체하여 오버피팅 및 많은 계산량 문제를 해결한 모델이고 Inception-BN은 초기 GoogLeNet[17]의 배치 정규화(Batch Normalization)를 적용한 모델이다. WRN은 ResNet[14]의 핵심 기술을 채용하면서, 인셉션 모듈처럼 계층의 폭을 넓힌 모델이고 DenseNet은 모든 레이어에서 후속 레이어로의 직접 연결을 추가하여 설계된 모델이다. 특유의 학습 기술이 없는 깊은 컨벌루션 신경망(Deep Convolutional Neural Network, DCNN)[18], 즉 단순히 컨벌루션 및 완전연결 계층을 적층시킨 AlexNet과 VGGNet은 비교대상으로 적절치 않다고 판단하여 실험에서 제외하였다. 활성 함수 측면에서, 시그모이드(Sigmoid), 쌍곡 탄젠트(Hyperbolic Tangent) 함수는 ReLU보다 성능이 좋지 않다고 입증된바[19-21]가 있어 실험에서 제외하였다. 매개변수 갱신 방법의 경우 네스테로프 모멘텀(Nesterov Momentum)[22]을 사용하였다.

솔버(Solver) 파일의 하이퍼파라미터 값 설정은 초기 학습률(Base Learning Rate)='0.01', 가중치 감쇠(Weight Decay)='0.0005', 모멘텀(Momentum)='0.9', 감마(Gamma)='0.94', 평균 손실(Average Loss)='1,000', 학습률 방침(Learning Rate Policy)="Step", 보폭 크기(Step Size)='10,000'과 같고, 네트워크 파일의 활성 함수 하이퍼파라미터 값 설정은 Table 2.와 같다. 본 논문에서는 위와 같은 설정을, 도출한 결론의 객관성 확보를 위해 모든 실험에 동일하게 적용하였다.

Table 2. Hyper-parameter lambda value of each activation function

Activation Function	Hyper-Parameter(λ)
LReLU	0.1
RReLU	$3.0 < \lambda < 8.0$
ELU	1.0

4. 실험 결과

실험에서 사용된 하드웨어 사양은 Table 3과 같다.

에폭(Epoch)은 '250'으로 지정하였으며 솔버 및 네트워크 파라미터 설정은 앞서 언급했던 내용과 동일하다. 본 논문은 검증 데이터 세트를 사용하여 출

Table 3. Hardware specifications used in the experiment

	specifications
CPU	Intel I7-7700K 4.20GHz
GPU	Geforce Titan Xp 12GB
RAM	삼성 DDR4 32GB
SSD	삼성전자 850 Pro 512GB

력 된 Top-1 오류율(Error Rate)과 각 활성화 함수들의 학습 수렴속도(Training Convergence Speed) 및 손실률(Loss Rate)을 통해 성능을 평가한다.

4.1 NIN

Table 4의 5가지 활성화 함수 중에서 PReLU가 Top-1 Error Rate 20.39%로 가장 낮은 오류율을 보여주었다. RReLU가 20.71%, ELU가 22.00%, LReLU가 22.18%로 그 뒤를 이었다. ReLU는 23.37%로 가장 높은 오류율을 보였고 PReLU와 2.98%의 성능 차이를 보였다. 테스트 손실률의 경우는 RReLU가 1.13%로 가장 낮았으며 PReLU와 0.05%의 근소한 차이를 보였다. ELU가 1.21%, LReLU가 1.30%로 그 뒤를 이었다.

Fig. 1에서 PReLU와 RReLU는 다른 활성화 함수보다 학습 수렴속도가 빠르며 5 Epoch이후 PReLU가 RReLU보다 1.18% 더 높은 정확도를 기록했다. 가장 낮은 정확도를 기록한 ReLU는 43.5%로 1위인 PReLU와 13.89%의 차이를 보였고, 5 Epoch때는 4 Epoch때보다 오히려 성능이 떨어졌다. LReLU는 1 Epoch때 두 번째로 낮은 정확도로 학습을 시작하였지만 2 Epoch때부터 ELU보다 높은 정확도를 보여주며 최종 순위 3위를 기록했고 ELU가 그 뒤를 이었다. 분석 결과, NIN 모델에서 PReLU와 RReLU는 초기 학습

Table 4. Top-1 error rate and softmax loss rate of each activation function in nin

NIN			
Activation Function	Top-1 Error Rate[%]	Softmax Loss Rate[%]	Epoch
ReLU	23.37	1.51	239
LReLU	22.18	1.30	229
PReLU	20.39	1.18	214
RReLU	20.71	1.13	223
ELU	22.00	1.21	205

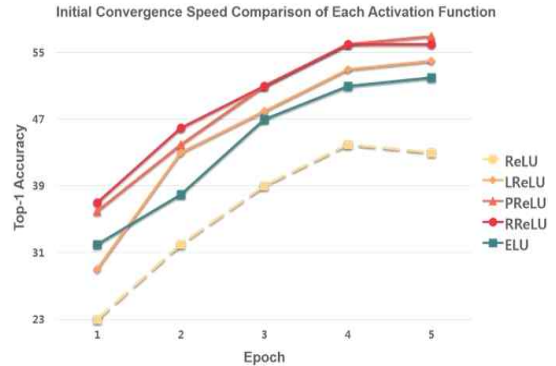


Fig. 1. Initial convergence speed comparison of each activation function in nin.

수렴속도가 다른 활성화 함수들보다 빨랐고 정확도 역시 높았다. 결과적으로 PReLU가 5 Epoch를 기준으로 가장 높은 정확도를 보여주었지만 RReLU와의 초기 1.18%의 차이는 관련 하이퍼파라미터 조정으로 그 순위가 역전될 수도 있다고 판단된다. 따라서 둘 중 더 우수한 활성화 함수가 무엇인지 결정 짓기 어렵다. LReLU 및 ELU는 ReLU 보다는 학습 수렴속도가 빠르나, 위 두 함수와 정확도 차이가 있다. 그래프 개형을 보면 ReLU를 제외한 나머지 활성화 함수들은 초기 5 Epoch까지 상승세를 보였다.

4.2 Inception-BN

Table 5의 5가지 활성화 함수 중에서 ELU가 Top-1 Error Rate 17.18%로 가장 낮은 오류율을 보여주었다. RReLU가 19.02% PReLU가 19.03%, LReLU가 19.57%로 그 뒤를 이었다. ReLU는 22.73%로 가장 높은 오류율을 보였고 PReLU와 5.55%의 성능 차이를 보였다. PReLU와 RReLU는 오류율이 0.01% 차이 밖에 나지 않기 때문에 사실, 동일하다고 봐도 무

Table 5. Top-1 error rate and softmax loss rate of each activation function in inception-bn

Inception-BN			
Activation Function	Top-1 Error Rate[%]	Softmax Loss Rate[%]	Epoch
ReLU	22.73	1.26	242
LReLU	19.57	1.05	181
PReLU	19.03	0.97	170
RReLU	19.02	1.00	247
ELU	17.18	0.94	184

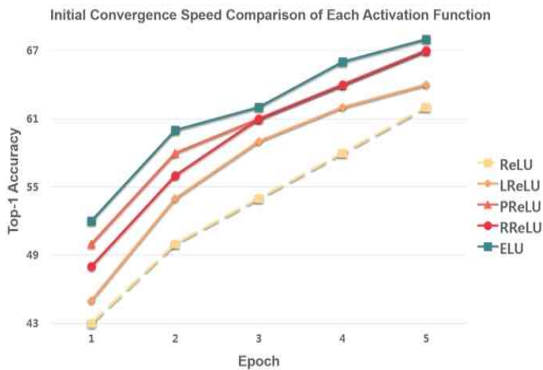


Fig. 2. Initial convergence speed comparison of each activation function in inception-bn.

방하다. 테스트 손실률의 경우도 ELU가 0.94%로 가장 낮았으며 PReLU와 0.03%의 근소한 차이를 보였다. RReLU가 1.00%, LReLU가 1.05%로 그 뒤를 이었다.

Fig. 2에서 ELU는 다른 활성화 함수보다 학습 수렴 속도가 빠르며 5 Epoch이후 68.25%의 가장 높은 정확도를 기록했다. 가장 낮은 정확도를 기록한 ReLU는 62.02%로 1위인 PReLU와 6.23%의 차이를 보였고, NIN 모델과 달리 5 Epoch때 성능이 떨어지지 않았다. PReLU는 2 Epoch까지 RReLU 보다 학습 수렴속도가 빠르지만 3 Epoch부터 RReLU와 속도 면에서 비슷한 양상을 보인다. LReLU는 64.64%로 최종 순위 4위를 기록했다. 분석 결과, Inception-BN 모델에서 ELU는 초기 학습 수렴속도가 다른 활성화 함수들보다 빨랐고 정확도 역시 높았다. 결과적으로, PReLU가 5 Epoch를 기준으로 가장 높은 정확도를 보여주었다. 5 Epoch기준, PReLU와 RReLU의 정확도 차이는 0.60%이기 때문에 관련 하이퍼파라미터 조정으로 그 순위가 역전될 수도 있다고 판단된다. 따라서 둘 중 더 우수한 활성화 함수가 무엇인지 결정 짓기 어렵다. ReLU의 경우 5 Epoch까지 상승세는 보이지만 다른 함수들과의 정확도 차이가 있다.

4.3 Wide_ResNet

Table 6의 5가지 활성화 함수 중에서 PReLU가 Top-1 Error Rate 17.84%로 가장 낮은 오류율을 보여주었다. RReLU가 18.47%, LReLU가 19.00%, ELU가 19.21%로 그 뒤를 이었다. ReLU는 21.02%로 가장 높은 오류율을 보였고 PReLU와 3.18%의 성능

Table 6. Top-1 error rate and softmax loss rate of each activation function in wrn-28-10

WRN-28-10			
Activation Function	Top-1 Error Rate[%]	Softmax Loss Rate[%]	Epoch
ReLU	21.02	1.02	219
LReLU	19.00	0.89	238
PReLU	17.84	0.79	238
RReLU	18.47	0.85	245
ELU	19.21	0.75	247

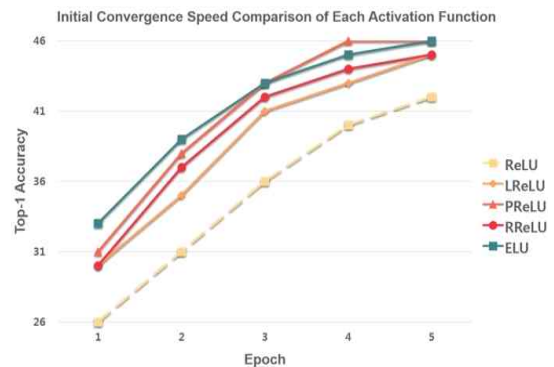


Fig. 3. Initial convergence speed comparison of each activation function in wrn-28-10.

차이를 보였다. 테스트 손실률의 경우는 ELU가 0.75%로 가장 낮았으며 PReLU와 0.04%의 근소한 차이를 보였다. RReLU가 0.85%, LReLU가 0.89%로 그 뒤를 이었다.

Fig. 3에서 PReLU와 ELU는 다른 활성화 함수보다 학습 수렴속도가 빠르며 5 Epoch이후 PReLU가 ELU보다 0.09% 더 높은 정확도를 기록했다. 가장 낮은 정확도를 기록한 ReLU는 42.00%로 1위인 PReLU와 4.94%의 차이를 보였다. 5 Epoch이전의 성능 차이는 있지만, LReLU와 RReLU의 5 Epoch정확도는 각각 45.75%, 45.90%로 비슷했다. 분석 결과, WRN-28-10 모델에서 1 Epoch정확도는 ELU가 가장 높지만, 3 Epoch이후 PReLU에 역전을 허용했다. ReLU를 제외한 나머지 활성화 함수들의 학습 수렴속도는 큰 차이를 보이지 않았다.

4.4 DenseNet

Table 7의 5가지 활성화 함수 중에서 ELU가 Top-1 Error Rate 15.13%로 가장 낮은 오류율을 보여주었

Table 7. Top-1 error rate and softmax loss rate of each activation function in densenet-161

DenseNet-161			
Activation Function	Top-1 Error Rate[%]	Softmax Loss Rate[%]	Epoch
ReLU	18.44	0.93	187
LReLU	16.41	0.83	218
PReLU	15.48	0.78	219
RReLU	16.23	0.77	233
ELU	15.13	0.69	249

다. PReLU가 15.48%, RReLU가 16.23%, LReLU가 16.41%로 그 뒤를 이었다. ReLU는 18.44%로 가장 높은 오류율을 보였고 ELU와 3.31%의 성능 차이를 보였다. 테스트 손실률의 경우도 ELU가 0.69%로 가장 낮았으며 RReLU와 0.08%의 근소한 차이를 보였다. PReLU가 0.78%, LReLU가 0.83%로 그 뒤를 이었다. RReLU와 PReLU는 손실률이 0.01% 차이 밖에 나지 않기 때문에 사실, 동일하다고 봐도 무방하다.

Fig. 4에서 RReLU는 1 Epoch정확도가 다른 함수들보다 높지만, 5 Epoch정확도는 52.91%로 2위를 기록했다. 반면 PReLU는 1 Epoch정확도는 RReLU보다 낮지만, 5 Epoch정확도는 53.30%로 가장 높았다. 가장 낮은 정확도를 기록한 ReLU는 42.15%로 1위인 PReLU와 11.15%의 차이를 보였고, 5 Epoch 때는 4 Epoch때보다 오히려 성능이 떨어졌다. LReLU와 ELU는 3, 4 Epoch때 이전보다 성능이 떨어지는 모습을 보였지만, 이후 상승세를 보였다. 분석 결과, DenseNet-161 모델에서도 ReLU를 제외한 나머지 활성화 함수들의 초기 학습 수렴속도는 ReLU보다 빨

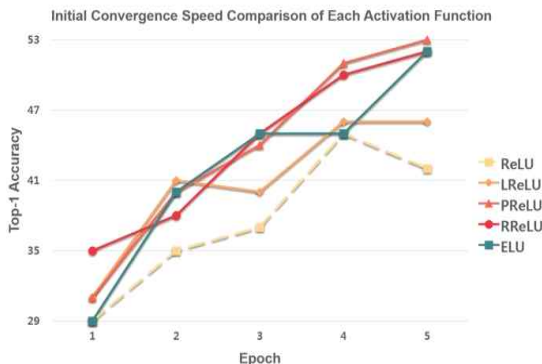


Fig. 4. Initial convergence speed comparison of each activation function in densenet-161.

Table 8. The highest performance activation function of each network

Network	Activation Function	Top-1 Error Rate
NIN	PReLU	20.39
Inception-BN	ELU	17.18
WRN-28-10	PReLU	17.84
DenseNet-161	ELU	15.13

랐고 정확도 역시 높았다.

5. 결 론

본 논문에서는 CNN의 기울기 소실 문제를 완화하기 위해 제안된 다양한 활성화 함수들을, 서로 다른 구조의 CNN 모델에서 실험하였다. 활성화 함수로 ReLU, LReLU, PReLU, RReLU, ELU를 사용하였고 CNN 모델로 NIN, Inception-BN, WRN-28-10, DenseNet-161을 사용하였다. 27종의 소규모 곤충 데이터세트를 20가지 실험에서 학습시켰을 때 가장 높은 성능을 보여준 활성화 함수는 Table 8과 같다.

NIN에서 PReLU가, Inception-BN에서 ELU, WRN-28-10에서 PReLU 그리고 DenseNet-161에서는 ELU가 가장 높은 성능을 보여주었다. ReLU는 20가지 실험에서 정확도, 손실률, 초기 학습 수렴속도 모두 성능이 가장 낮았다. 이를 바탕으로, 0보다 작은 음수가 활성화값(Active-Value)으로 입력되었을 때 0이 아닌 양수로 출력하는 방법이 더 좋은 성능을 얻을 수 있다는 것을 알 수 있다.

끝으로, 실험에 사용된 결과들은 추가 학습 및 솔버의 하이퍼파라미터 값 조정을 통해 더 높은 정확도를 얻을 수 있을 것으로 기대된다. 향후 연구 과제로 빠르고 정확한 활성화 함수의 개발도 중요하지만, ReLU 이후 발표된 나머지 활성화 함수들의 상관관계를 명확하게 밝힐 수 있는 연구가 필요할 것으로 사료된다.

REFERENCE

- [1] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, et al., "Mastering the Game of Go without Human Knowledge," *Nature*, Vol. 550, pp. 354-359, 2017.

- [2] L. Yann, B. Yoshua, and H. Geoffrey, "Deep leaning," *Nature*, Vol. 521, No. 7553, pp. 436-444, 2015.
- [3] H. Huang, J. Joseph, M. Huang, and T. Margolina, "Automated Detection and Identification of Blue and Fin Whale Foraging Calls by Combining Pattern Recognition and Machine Learning Techniques," *Proceeding of OCEANS 2016 MTS/IEEE Monterey*, pp. 1-7, 2016.
- [4] N. Akhtar and A. Mian, "Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey," *arXiv preprint*, arXiv:1801.00553, 2018.
- [5] R. Olga, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, Vol. 115, No. 3, pp. 211-252, 2015.
- [6] K. Alex, S. Ilya, and H. Geoffrey, "ImageNet Classification with Deep Convolutional Neural Networks," *Proceeding of Advances in Neural Information Processing System*, pp. 1097-1105, 2012.
- [7] V. Nair and G. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," *Proceedings of the 27th International Conference on Machine Learning*, pp. 807-814, 2010.
- [8] M. Lin, Q. Chen, and S. Yan, "Network In Network," *arXiv*, arXiv:1312.4400v3, 2014.
- [9] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *arXiv preprint*, arXiv:1502.03167v3, 2015.
- [10] S. Zagoruyko and N. Komodakis, "Wide Residual Networks," *arXiv preprint*, arXiv:1605.07146, 2016.
- [11] G. Huang, Z. Liu, L.V.D. Maaten, and K.Q. Weinberger, "Densely Connected Convolutional Networks," *Proceeding of 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2261-2269, 2017.
- [12] J. Yangqing, "Caffe: An Open Source Convolutional Architecture for Fast Feature Embedding," *Proceeding of the 22nd ACM International Conference on Multimedia*, pp. 675-678, 2014.
- [13] A.L. Maas, A.Y. Hannun, and A.Y. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models," *Proceeding of International Conference on Machine Learning*, 2013.
- [14] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep Into Rectifiers: Surpassing Human-Level Performance On Imagenet Classification," *Proceeding of International Conference on Computer Vision*, pp. 1026-1034, 2015.
- [15] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical Evaluation of Rectified Activations in Convolutional Network," *arXiv*, arXiv:1505.00853, 2015.
- [16] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," *arXiv*, arXiv:1511.07289, 2015.
- [17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., "Going Deeper with Convolutions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9, 2015.
- [18] Y. Jeong, I. Ansari, J. Shim and J. Lee, "A Car Plate Area Detection System Using Deep Convolution Neural Network," *Journal of Korea Multimedia Society*, Vol. 20, No. 8, pp. 1166-1174, 2017.
- [19] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the Best Multi-stage Architecture for Object Recognition?," *Proceeding of IEEE 12th International Conference on Computer Vision*, pp. 2146-2153, 2009.
- [20] V. Nair and G. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," *2010 27th International Conference on Machine Learning*, pp. 807-814, 2010.
- [21] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," *Proceedings of the Fourteenth International Confer-*

ence on Artificial Intelligence and Statistics (AISTATS), *PMLR*, Vol. 15, pp. 315-323, 2011.

- [22] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the Importance of Initialization and Momentum in Deep Learning," *Proceedings of the 30th International Conference on Machine Learning*, Vol. 28, pp. 1139-1147, 2013.



박 성 욱

2018년 순천대학교 컴퓨터공학과
(공학사)
2018년~현재 순천대학교 컴퓨터
공학과 석사과정
관심분야: 컴퓨터비전, 기계학습



김 도 연

1986년 충남대학교 계산통계학과
졸업(이학사)
2000년 충남대학교 대학원 정보
통신공학과 졸업(공학석
사)
2003년 충남대학교 대학원 컴퓨
터공학과 졸업(공학박사)

1986년~1996년 한국원자력연구원 선임연구원
1997년~2008년 한국전력기술(주) 책임연구원
2008년~현재 순천대학교 컴퓨터공학과 교수
관심분야: 컴퓨터비전, 컴퓨터보안, 기계학습