KIPS Trans. Softw. and Data Eng.
Vol.11, No.9 pp.371~380
ISSN: 2287-5905 (Print), ISSN: 2734-0503 (Online)

https://doi.org/10.3745/KTSDE.2022.11.9.371

Performance Improvement Method of Convolutional Neural Network Using Combined Parametric Activation Functions

Young Min Ko† · Peng Hang Li†† · Sun Woo Ko†††

ABSTRACT

Convolutional neural networks are widely used to manipulate data arranged in a grid, such as images. A general convolutional neural network consists of a convolutional layers and a fully connected layers, and each layer contains a nonlinear activation functions. This paper proposes a combined parametric activation function to improve the performance of convolutional neural networks. The combined parametric activation function is created by adding the parametric activation functions to which parameters that convert the scale and location of the activation function are applied. Various nonlinear intervals can be created according to parameters that convert multiple scales and locations, and parameters can be learned in the direction of minimizing the loss function calculated by the given input data. As a result of testing the performance of the convolutional neural network using the combined parametric activation function on the MNIST, Fashion MNIST, CIFAR10 and CIFAR100 classification problems, it was confirmed that it had better performance than other activation functions.

Keywords : Convolutional Neural Network, Nonlinear Activation Function, Combined Parametric Activation Function, Loss function

결합된 파라메트릭 활성함수를 이용한 합성곱 신경망의 성능 향상

고 영 민† · 이 붕 항†† · 고 선 우†††

요 약

합성곱 신경망은 이미지와 같은 격자 형태로 배열된 데이터를 다루는데 널리 사용되고 있는 신경망이다. 일반적인 합성곱 신경망은 합성곱층과 완전연결층으로 구성되며 각 층은 비선형활성함수를 포함하고 있다. 본 논문은 합성곱 신경망의 성능을 향상시키기 위해 결합된 파라메트릭 활성함수를 제안한다. 결합된 파라메트릭 활성함수는 활성함수의 크기와 위치를 변환시키는 파라미터를 적용한 파라메트릭 활성함수들을 여러 번 더하여 만들어진다. 여러 개의 크기, 위치를 변환하는 파라미터에 따라 다양한 비선형간격을 만들 수 있으며, 파라미터는 주어진 입력데이터에 의해 계산된 손실함수를 최소화하는 방향으로 학습할 수 있다. 결합된 파라메트릭 활성함수를 사용한 합성곱 신경망의 성능을 MNIST, Fashion MNIST, CIFAR10 그리고 CIFAR100 분류문제에 대해 실험한 결과, 다른 활성함수들보다 우수한 성능을 가짐을 확인하였다.

키워드 : 합성곱 신경망, 비선형활성함수, 결합된 파라메트릭 활성함수, 손실함수

1. 서 론

격자 형태로 배열된 데이터를 다루는데 특화된 합성곱 신경망(Convolutional neural network)은 [1] 입력이 이미지인 경우, 합성곱 연산을 통해 이미지에 포함된 특정 대상의 위치를 파악하거나 탐지하는데 유용하게 사용된다. 합성곱 신경망의 합성곱 연산은 파라미터를 공유하여 파라미터 수를 감소시킬 수 있어 통계적 효율성이 높아진다[1].

일반적인 합성곱 신경망 구조를 표현하면 Fig. 1과 같다. 합성곱 신경망은 크게 합성곱층과 완전연결층 두 단계로 구분할 수 있으며 Fig. 1과 같이 입력층과 F개의 합성곱층, K개의 완전연결신경망의 은닉층 그리고 출력층을 갖는다. 합성곱 신경망의 출력값 $\boldsymbol{z}^{(out)}$과 레이블 $\boldsymbol{t}$의 함수로 정의되는 손실함수는 Equation (1)과 같다.

$$L(\boldsymbol{z}^{(out)},\boldsymbol{t}) = L(W) \tag{1}$$

Equation (1)에서 W는 합성곱 신경망의 모든 선형변환 파라미터를 나타낸다.

† 정 회 원 : 전주대학교 인공지능연구소 연구교수
†† 비 회 원 : 전주대학교 인공지능학과 석사
††† 정 회 원 : 전주대학교 인공지능학과 교수
Manuscript Received : December 16, 2021
First Revision : February 22, 2022
Accepted : March 18, 2022
* Corresponding Author : Sun Woo Ko(godfriend0@gmail.com)

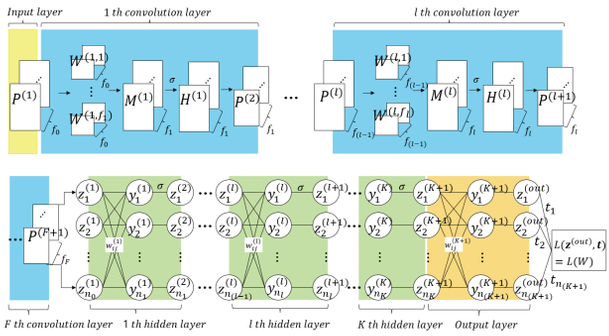


Fig. 1. Structure of a General Convolutional Neural Network (Consisting of F Convolutional Layers and K Hidden Layers of Fully Connected Neural Networks)

일반적으로 합성곱층은 특징맵(Feature map) M, 활성화맵(Activation map) H, 풀링맵(Pooling map) P로 구성되며 예를 들어 l번째 합성곱층의 연산 과정은 다음과 같다.

1) l번째 합성곱층의 입력데이터 $P^{(l)}$과 l번째 합성곱층의 필터 파라미터들 $W^{(l,i)}, i=1,\cdots,f_l$의 합성곱 연산을 통해 계산되는 특징맵 $M^{(l)}$.

2) l번째 합성곱층의 특징맵 $M^{(l)}$에 비선형활성함수 σ을 적용한 활성화맵 $H^{(l)}$.

3) 활성화맵 $H^{(l)}$의 크기를 줄이기 위한 풀링 연산을 통해 계산된 풀링맵 $P^{(l+1)}$.

합성곱 신경망의 입력데이터 $P^{(1)}$은 위와 같은 합성곱층의 연산을 통해 특징을 추출하여 완전연결신경망의 입력 $(z_1^{(1)},\cdots,z_{n_0}^{(1)})$으로 전달되며 출력값 $\boldsymbol{z}^{(out)}$을 계산한다.

한편 모든 신경망 구조가 가지고 있는 비선형활성함수의 의미를 이해하기 위해 분류문제를 예로 생각해보자. Fig. 2는 2차원 입력변수 (x_1,x_2)에 대응하는 $y_i, i=1,2,...,5$들이 클래스 0 혹은 클래스 1에 분류되게 하는 문제이다. 만약 선형모델만을 사용할 경우 경사하강법을 이용해 찾아진 선형변환 파라미터 w와 입력변수 x의 선형변환을 통해 만들어진 초평면 y만으로는 Fig. 2와 같은 비선형 분류문제를 나눌 수 없다. 이때 비선형활성함수 σ을 포함한 신경망을 사용할 경우 y_i들을 비선형 변환하여 향상된 분류성능을 가질 수 있다.

이와 같이 비선형성은 비선형활성함수가 가지는 중요한 성질[2]이며 본 논문은 입력데이터에 따라 다양한 비선형변환을 할 수 있는 결합된 파라메트릭 활성함수(Combined parametric activation function : CPAF)[3]가 합성곱 신경망의 성능을 향상시킬 수 있음을 보인다.

기존에 크기와 위치를 변환하는 단일한 파라미터 a, b을 가지는 파라메트릭 활성함수[4-6]들의 연구로 완전연결 신경망, 합성곱 신경망의 성능 향상과 기울기 소실 문제를 완화할 수 있었다. 이에 비선형성을 증가시키기 위해 크기와 위치를 변환하는 파라미터를 여러 개 결합한 파라메트릭 활성함수[3]가 심층신경망의 성능을 향상시킬 수 있음을 제시하였다.

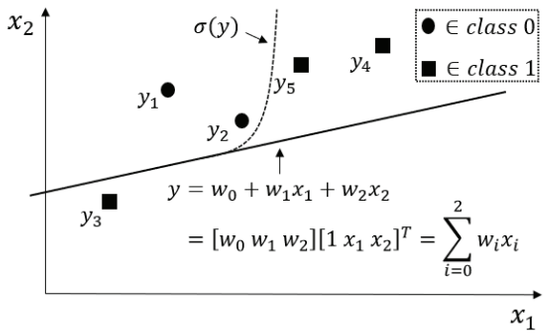


Fig. 2. Classification Problem
(x: Input Variable, w: Linear Transformation Parameter)

본 논문에서는 결합된 파라메트릭 활성함수가 합성곱 구조에서 가지는 의미와 실험을 통해 합성곱 신경망의 분류성능을 올릴 수 있음을 보인다.

여러 개의 파라메트릭 활성함수를 통해 다양한 비선형간격을 만들어낼 수 있어 비선형성을 키울 수 있고 각각의 파라미터들은 손실함수를 최소화하는 방향으로 학습하여 성능을 올릴 수 있다.

2. 관련된 연구

Fig. 2와 같은 분류문제를 선형모델만으로 해결할 수 없는 경우 신경망을 사용할 때 비선형활성함수는 초평면 y을 비선형 변환하여 분류 성능을 높일 수 있는 중요한 역할을 한다. 비선형활성함수에 대한 연구는 크게 파라미터의 유무에 따라 구분할 수 있다[7]. 그 중 파라미터가 없는 대표적인 활성함수는 Sigmoid, Tanh 그리고 ReLU(Rectified linear unit) 함수[8]가 있다.

Sigmoid, Tanh함수의 경우 치역의 범위가 각각 (0,1), (-1,1)을 갖는 유계함수로써 미분 가능하고 연속이며 일대일 대응하는 성질을 가지는 단조함수이다[9]. 하지만 비선형성 증가를 위해 Sigmoid, Tanh함수를 가지는 은닉층 수가 늘어나는 경우 기울기 소실 문제[10]가 발생할 수 있다. 이를 해결하기 위해 파라미터를 도입한 활성함수 Hexpo[11], ISigmoid[12], ReLTanh[13] 등이 연구되었다.

Kong과 Takatsuka[11]이 제시한 Hexpo함수는 Tanh함수와 비슷한 형태로 기울기를 조절하는 파라미터 4개를 가지고 다양한 활성함수 형태를 만들 수 있다. Qin 등[12]이 제시한 ISigmoid와 Wang 등[13]이 제시한 ReLTanh함수는 Sigmoid와 Tanh함수의 수렴하는 양쪽 끝부분에 직선을 적용한 함수로써 직선을 적용하는 구간에 파라미터를 도입하여 학습할 수 있다.

Nair 등[8]이 제안한 ReLU함수는 합성곱층이 깊은 합성곱 신경망에서 발생하는 기울기 소실 문제를 해결하기 위해 소개되었고 이후 파라미터를 적용한 ReLU함수의 변형들이

합성곱 신경망의 학습을 향상시키기 위해 연구되고 비교되었다[14,15]. 대표적으로 He 등[16]이 제시한 PReLU(Parametric ReLU)와 Clevert 등[17]이 제시한 ELU(Exponential linear unit) 등이 있다. PReLU와 ELU는 0보다 작은 입력 y값에 대해 0을 출력하는 ReLU함수를 각각 αy, $\alpha(e^{y}-1)$로 변환한 함수로 임의의 실수값을 가지는 α는 경사하강법을 사용해 학습된다.

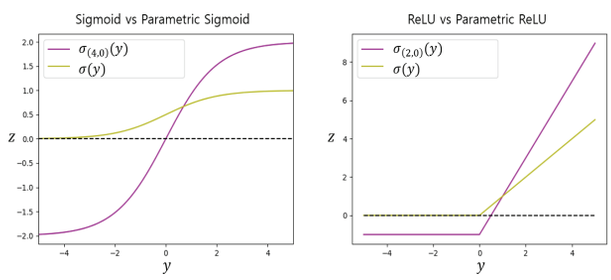
파라미터를 적용한 활성함수 연구 중 고영민 등[6]은 손실함수와 무관한 비선형활성함수의 문제점을 개선한 파라메트릭 활성함수를 소개하였다. 제시된 파라메트릭 활성함수는 입력데이터에 맞게 활성함수 σ의 크기와 위치를 변환하는 파라미터를 적용하여 경사하강법을 이용해 학습할 수 있으며[6], Sigmoid와 ReLU함수에 파라메트릭 활성함수를 적용한 Equation은 각각 Equation (2), (3)와 같다.

$$z=\sigma_{(a,b)}(y)=\frac{a}{1+e^{-(y-b)}}-\frac{a}{2} \tag{2}$$

$$z=\sigma_{(a,b)}(y)=\begin{cases}a(y-b)-\dfrac{a}{2}, & y>b\\ -\dfrac{a}{2}, & y\le b\end{cases} \tag{3}$$

Equation (2), (3)의 입력값 y에 대한 비선형변환 z값은 크기를 결정하는 파라미터 a와 위치를 결정하는 파라미터 b에 따라 다양한 z값을 가질 수 있으며 $-a/2$을 더함으로써 음수값을 취할 수 있어 다양한 비선형변환 값의 범위를 가질 수 있다[6]. Fig. 3은 Sigmoid와 ReLU함수에 대해 파라메트릭 활성함수를 적용한 경우와 비교한 것으로 Fig. 3(a)같이 $a=4, b=0$을 가지는 파라메트릭 활성함수를 적용한 Sigmoid함수는 Sigmoid함수 보다 다양한 비선형변환을 할 수 있다.

이후 더욱 다양한 비선형간격을 만들어내기 위해 파라메트릭 활성함수를 결합한 결합된 파라메트릭 활성함수를 제안하였고 심층신경망에서 다른 함수들보다 우수한 성능을 가짐을 확인하였다[3].



(a) (b)

Fig. 3. Nonlinear Activation Function $\sigma(y)$ and Parametric Activation Function $\sigma_{(a,b)}(y)$. (a): Sigmoid, (b): ReLU

3. 결합된 파라메트릭 활성함수를 이용한 합성곱 신경망

3.1 합성곱 신경망의 활성함수

Fig. 1과 같은 일반적인 합성곱 신경망의 출력값은 입력데이터가 합성곱층의 연산을 반복하여 특징을 추출한 뒤 완전연결신경망의 입력으로 전달되어 선형변환과 비선형변환을 반복하여 계산된 $\boldsymbol{z}^{(out)}$값이다. 합성곱 신경망의 학습은 Equation (1)과 같이 출력값과 레이블의 함수로 정의되는 손실함수를 모든 합성곱 신경망의 선형변환 파라미터 W에 대해 경사하강법을 이용하여(Equation (4)) W에 대한 손실함수 변화율이 동시에 0을 만족하는 점을 찾아가는 과정이다.

$$W=W-\rho\frac{\partial L}{\partial W} \tag{4}$$

ρ는 학습률을 의미하며 Equation (4)의 $\partial L/\partial W$는 W에 대한 손실함수의 변화율을 나타낸다.

합성곱층의 연산을 자세히 보기 위해 Fig. 1에서 $f_{(l-1)}$개의 채널을 가지는 임의의 l번째 합성곱층의 입력 $P^{(l)}$과 높이와 너비가 (3×3)사이즈의 l번째 합성곱층의 c번째 필터 파라미터 $W^{(l,c)}$의 합성곱 연산을 통해 나온 특징맵 $M^{(l)}$의 c번째 채널의 원소 i,j는 Fig. 4의 $m_{i,j,c}^{(l)}$과 같으며 계산은 Equation (5)과 같다(이때 스트라이드는 1 패딩은 0을 가정하였다).

$$m_{i,j,c}^{(l)}=\sum_{u=1}^{3}\sum_{v=1}^{3}\sum_{r=1}^{f_{(l-1)}}w_{u,v,r}^{(l,c)}\,p_{i+u-1,j+v-1,r}^{(l)} \tag{5}$$

Equation (5)에서 $w_{u,v,r}^{(l,c)}$의 u,v,r는 $W^{(l,c)}$의 높이, 너비, 채널의 위치를 의미하며 $P^{(l)}$의 원소 $p^{(l)}$도 마찬가지이다. Equation (5)은 Fig. 2의 y 같이 선형변환을 의미한다.

활성화맵은 특징맵이 비선형활성함수를 통해 원소별로 비선형 변환되어 만들어지며 예를 들어 Equation (5)에 대해 비선형활성함수를 Sigmoid함수로 사용할 경우 Equation (6)과 같다.

$$h_{i,j,c}^{(l)}=\sigma(m_{i,j,c}^{(l)})=\frac{1}{1+e^{-m_{i,j,c}^{(l)}}} \tag{6}$$

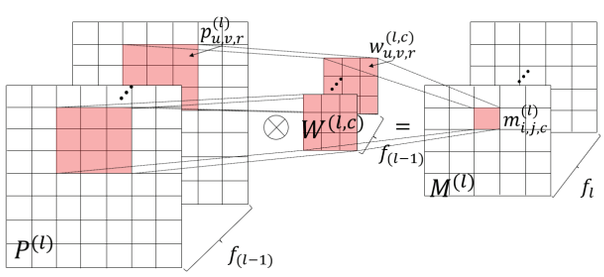


Fig. 4. Convolution Operation of the lth Convolution Layer ($P^{(l)}$ of the lth Convolution Layer : Input Data, $W^{(l,c)}$: cth Filter, $M^{(l)}$: Feature Map)

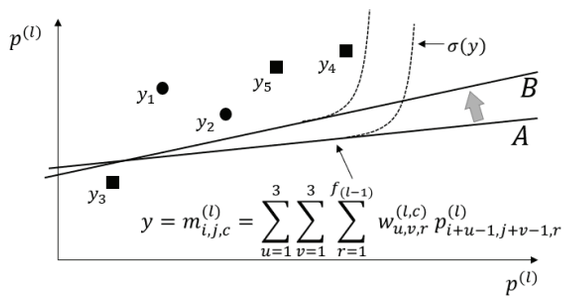


Fig. 5. Classification problem with two classes ($p^{(l)}$: input data, $w^{(l,c)}$: linear transformation parameters)

Equation (6)의 $h_{i,j,c}^{(l)}$는 l번째 합성곱층의 활성화맵 $H^{(l)}$의 원소이다.

Equation (6)과 같이 계산된 활성화맵은 풀링을 통해 계산되며 이 과정을 반복하여 완전연결신경망의 입력으로 전달되고 손실함수 값을 계산하여 Equation (4)과 같이 학습하게 된다.

합성곱 신경망의 합성곱층에서 활성함수의 의미를 이해하기 위해 Fig. 2와 같은 분류문제를 Equation (5)에서 사용된 $p^{(l)}$을 입력데이터로 사용하는 문제로 변환하여 생각해보자 (Fig. 5). Fig. 5에서 $p^{(l)}$은 고차원이지만 선형변환의 초평면의 의미를 전달하기 위해 2차원에 투영하였다. 법선벡터 $w^{(l,c)}$에 의해 만들어지는 초평면 y은 $w^{(l,c)}$가 경사하강법을 이용하여 학습되면서 초평면 y가 손실함수를 최소화하는 방향으로 이동한다. 이 과정이 Fig. 5의 A에서 B로 손실함수 값을 감소시키는 방향으로 변환된 초평면을 의미한다.

이후 초평면은 비선형활성함수 σ에 의해 비선형변환 $\sigma(y)$을 계산하게 되는데 Equation (6)의 Sigmoid함수를 사용하는 경우 제한된 비선형변환의 치역(0,1) 안에서 손실함수와 관련된 파라미터 없이 비선형 변환을 하여 손실함수를 최소화할 수 있다는 보장이 없다. 이 의미는 Fig. 5의 A에서 B로 이동된 초평면 다음에 이루어지는 비선형변환이 손실함수와 무관하게 진행되는 것을 의미한다. 이는 합성곱 신경망의 모든 비선형변환에서 Sigmoid 혹은 ReLU함수와 같은 파라미터가 없는 비선형활성함수를 사용할 경우 문제가 될 수 있다. 이러한 문제로 손실함수가 증가할 수 있고 그 영향으로 경사하강법을 이용한 학습 횟수가 늘어날 가능성이 있다.

3.2 결합된 파라메트릭 활성함수

위와 같이 합성곱 신경망에서 손실함수와 무관한 비선형활성함수에 의해 발생할 수 있는 문제를 개선하여 합성곱 신경망의 성능을 향상시키기 위해 결합된 파라메트릭 활성함수 [3]을 제안한다. 입력 y에 대해 결합된 파라메트릭 활성함수를 적용한 비선형 변환 z는 다음 Equation (7)과 같다.

$$z = \sum_{i=1}^{k} z_i = \sum_{i=1}^{k} \sigma_{(a_i,b_i)}(y) \tag{7}$$

Equation (7)의 $\sigma_{(a_i,b_i)}$는 i번째 파라메트릭 활성함수를 나타내며 a_i는 파라메트릭 활성함수의 크기를, b_i는 위치를 변환할 수 있는 파라미터로써 모두 실수값을 갖는다. Equation (7)의 z는 k개의 파라메트릭 활성함수를 더해 만들어진다.

예를 들어 입력값 y을 Sigmoid와 ReLU 함수에 결합된 파라메트릭 활성함수를 적용한 결합된 파라메트릭 Sigmoid와 결합된 파라메트릭 ReLU 함수는 각각 Equation (8), (9)과 같다.

$$z = \sum_{i=1}^{k} z_i = \sum_{i=1}^{k}\left(\frac{a_i}{1+e^{-(y-b_i)}} - \frac{a_i}{2}\right) \tag{8}$$

$$z = \sum_{i=1}^{k} z_i = \begin{cases} \sum_{i=1}^{k} a_i(y-b_i) - \frac{a_i}{2}, & y > b_i \\ \sum_{i=1}^{k} -\frac{a_i}{2}, & y \le b_i \end{cases} \tag{9}$$

결합된 파라메트릭 활성함수 z의 비선형성을 보기위해 Fig. 6과 같이 같은 간격 y값에 대응하는 Sigmoid함수 값과 $k=2$이고 $(a_1=4, b_1=4)$, $(a_2=4, b_2=-4)$인 결합된 파라메트릭 Sigmoid의 값을 나타내었다. Sigmoid함수를 나타내는 Fig. 6(a)는 치역이 (0,1)인 범위에서 제한적인 비선형성을 가지는 반면, Fig. 6(b)는 a_i,b_i값에 따라 다양한 비선형간격을 만들어 비선형변환을 할 수 있다. 이로 인해 비선형성을 키우기 위해 필요한 은닉층과 노드의 개수를 줄일 수 있다.

각 a_i,b_i들은 입력데이터를 보고 손실함수를 최소화하는 방향으로 학습할 수 있으며 결합된 파라메트릭 활성함수 z을 구성하는 a_i,b_i을 학습하기 위해 a_i,b_i에 대한 z의 변화율 (Equation (10,11))과 입력값 y에 대한 z의 변화율을 계산할 수 있다(Equation (12)).

$$\frac{\partial z}{\partial a_i} = \frac{1}{1+e^{-(y-b_i)}} - \frac{1}{2} \tag{10}$$

$$\frac{\partial z}{\partial b_i} = \frac{4z_i^2 - a_i^2}{4a_i} \tag{11}$$

$$\frac{\partial z}{\partial y} = \sum_{i=1}^{k} \frac{a_i^2 - 4z_i^2}{4a_i} \tag{12}$$

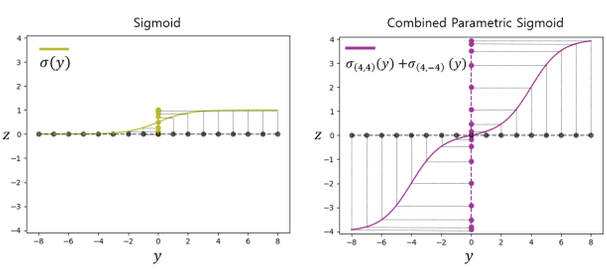


Fig. 6. Sigmoid $\sigma(y)$ and Combined Parametric Sigmoid $\sum_{i=1}^{2}\sigma_{(a_i,b_i)}(y)$

3.3 합성곱 신경망에서의 결합된 파라메트릭 활성함수 학습

Fig. 1과 같은 일반적인 합성곱 신경망에 결합된 파라메트릭 활성함수를 적용하여 학습할 수 있으며 비선형활성함수를 사용하는 합성곱층의 활성화맵 H과 완전연결신경망의 비선형변환 z에 적용된다.

예를 들어 합성곱 신경망에 결합된 파라메트릭 Sigmoid 함수를 사용할 때 임의의 l번째 특징맵 $M^{(l)}$의 c번째 채널의 원소 $m_{i,j,c}^{(l)}$의 활성화맵 원소 $h_{i,j,c}^{(l)}$ 연산은 Fig. 7과 같이 계산되며 Equation (13)과 같다.

$$h_{i,j,c}^{(l)}=\sum_{u=1}^{k}h_{i,j,c,u}^{(l)}=\sum_{u=1}^{k}\left(\frac{a_{i,j,c,u}^{(l)}}{1+e^{-(m_{i,j,c}^{(l)}-b_{i,j,c,u}^{(l)})}}-\frac{a_{i,j,c,u}^{(l)}}{2}\right) \quad (13)$$

Equation (13)에서 $a_{i,j,c,u}^{(l)}$과 $b_{i,j,c,u}^{(l)}$는 $h_{i,j,c}^{(l)}$의 각각 u번째 파라메트릭 활성함수의 크기와 위치를 변환하는 파라미터이다.

이후 완전연결신경망의 l번째 은닉층의 j번째 선형변환 $y_j^{(l)}$에 대한 결합된 파라메트릭 Sigmoid 함수의 변환 $z_j^{(l+1)}$은 Fig. 8과 같고 계산은 Equation (14)과 같다.

$$z_j^{(l+1)}=\sum_{u=1}^{k}z_{j,u}^{(l+1)}=\sum_{u=1}^{k}\left(\frac{a_{j,u}^{(l)}}{1+e^{-(y_j^{(l)}-b_{j,u}^{(l)})}}-\frac{a_{j,u}^{(l)}}{2}\right) \quad (14)$$

Equation (14)에서 $a_{j,u}^{(l)}$와 $b_{j,u}^{(l)}$는 $z_j^{(l+1)}$을 구성하는 u번째 파라메트릭 활성함수의 크기와 위치를 변환하는 파라미터이다.

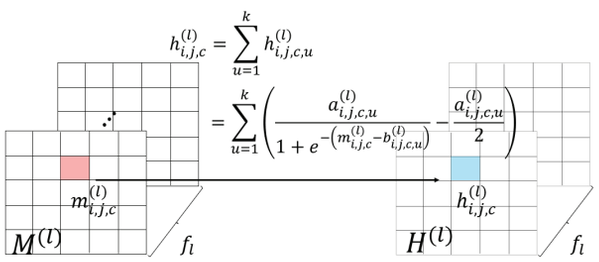


Fig. 7. Activation Map $H^{(l)}$ Using Combined Parametric Sigmoid Function on Feature Map $M^{(l)}$

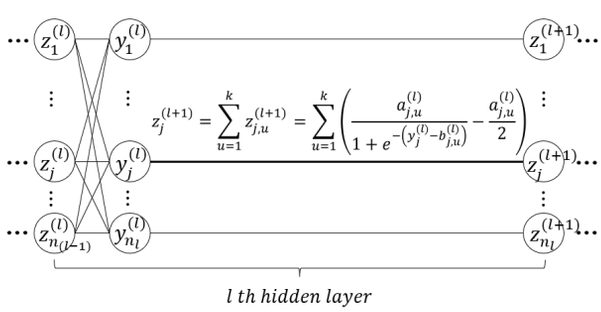


Fig. 8. Nonlinear Transformation $z_j^{(l+1)}$ Using Combined Parametric Sigmoid Function on Linear Transformation $y_j^{(l)}$

이렇게 합성곱 신경망의 모든 비선형활성함수에 적용할 수 있으며 합성곱 신경망의 손실함수(L)에 대해 결합된 파라메트릭 Sigmoid 함수의 파라미터들은 경사하강법을 이용하여 학습될 수 있다.

먼저 Equation (13)과 같은 활성화맵 원소 $h_{i,j,c}^{(l)}$의 $a_{i,j,c,u}^{(l)}$, $b_{i,j,c,u}^{(l)}$ 그리고 $m_{i,j,c}^{(l)}$에 대한 손실함수의 변화율은 다음과 같이 구해진다.

$$\frac{\partial L}{\partial a_{i,j,c,u}^{(l)}}=\frac{\partial L}{\partial h_{i,j,c}^{(l)}}\frac{\partial h_{i,j,c}^{(l)}}{\partial a_{i,j,c,u}^{(l)}}=\frac{\partial L}{\partial h_{i,j,c}^{(l)}}\times\left(\frac{1}{1+e^{-\left(m_{i,j,c}^{(l)}-b_{i,j,c,u}^{(l)}\right)}}-\frac{1}{2}\right) \quad (15)$$

$$\frac{\partial L}{\partial b_{i,j,c,u}^{(l)}}=\frac{\partial L}{\partial h_{i,j,c}^{(l)}}\frac{\partial h_{i,j,c}^{(l)}}{\partial b_{i,j,c,u}^{(l)}}=\frac{\partial L}{\partial h_{i,j,c}^{(l)}}\times\left(\frac{4\left(h_{i,j,c,u}^{(l)}\right)^2-\left(a_{i,j,c,u}^{(l)}\right)^2}{4a_{i,j,c,u}^{(l)}}\right) \quad (16)$$

$$\frac{\partial L}{\partial m_{i,j,c}^{(l)}}=\frac{\partial L}{\partial h_{i,j,c}^{(l)}}\frac{\partial h_{i,j,c}^{(l)}}{\partial m_{i,j,c}^{(l)}}=\frac{\partial L}{\partial h_{i,j,c}^{(l)}}\times\left(\sum_{u=1}^{k}\frac{\left(a_{i,j,c,u}^{(l)}\right)^2-4\left(h_{i,j,c,u}^{(l)}\right)^2}{4a_{i,j,c,u}^{(l)}}\right) \quad (17)$$

그리고 Equation (14)과 같은 완전연결신경망의 비선형변환 $z_j^{(l+1)}$을 구성하는 $a_{j,u}^{(l)}$, $b_{j,u}^{(l)}$ 그리고 $y_j^{(l)}$에 대한 손실함수의 변화율은 다음과 같다.

$$\frac{\partial L}{\partial a_{j,u}^{(l)}}=\frac{\partial L}{\partial z_j^{(l+1)}}\frac{\partial z_j^{(l+1)}}{\partial a_{j,u}^{(l)}}=\frac{\partial L}{\partial z_j^{(l+1)}}\times\left(\frac{1}{1+e^{-\left(y_j^{(l)}-b_{j,u}^{(l)}\right)}}-\frac{1}{2}\right) \quad (18)$$

$$\frac{\partial L}{\partial b_{j,u}^{(l)}}=\frac{\partial L}{\partial z_j^{(l+1)}}\frac{\partial z_j^{(l+1)}}{\partial b_{j,u}^{(l)}}=\frac{\partial L}{\partial z_j^{(l+1)}}\times\left(\frac{4\left(z_{j,u}^{(l+1)}\right)^2-\left(a_{j,u}^{(l)}\right)^2}{4a_{j,u}^{(l)}}\right) \quad (19)$$

$$\frac{\partial L}{\partial y_j^{(l)}}=\frac{\partial L}{\partial z_j^{(l+1)}}\frac{\partial z_j^{(l+1)}}{\partial y_j^{(l)}}=\frac{\partial L}{\partial z_j^{(l+1)}}\times\left(\sum_{u=1}^{k}\frac{\left(a_{j,u}^{(l)}\right)^2-4\left(z_{j,u}^{(l+1)}\right)^2}{4a_{j,u}^{(l)}}\right) \quad (20)$$

이상의 수식들을 통해 각각의 파라미터들은 경사하강법을 통해 최종적으로 손실함수를 최소화하는 방향으로 움직일 수 있다(Equation (4, 21-24)).

$$a_{i,j,c,u}^{(l)} = a_{i,j,c,u}^{(l)} - \rho \frac{\partial L}{\partial a_{i,j,c,u}^{(l)}} \tag{21}$$

$$b_{i,j,c,u}^{(l)} = b_{i,j,c,u}^{(l)} - \rho \frac{\partial L}{\partial b_{i,j,c,u}^{(l)}} \tag{22}$$

$$a_{j,u}^{(l)} = a_{j,u}^{(l)} - \rho \frac{\partial L}{\partial a_{j,u}^{(l)}} \tag{23}$$

$$b_{j,u}^{(l)} = b_{j,u}^{(l)} - \rho \frac{\partial L}{\partial b_{j,u}^{(l)}} \tag{24}$$

Table 1. Activation Functions used in the Experiment

	Sigmoid	ReLU
AF $\sigma(y)$	$\frac{1}{1+e^{-y}}$	$\begin{cases} y, y>0 \\ 0, y \le 0 \end{cases}$
CPAF $(k=2)$ $\sum_{i=1}^{2}\sigma_{(a_i,b_i)}(y)$	$\sum_{i=1}^{2}\left(\frac{a_i}{1+e^{-(y-b_i)}}-\frac{a_i}{2}\right)$, $a_1=4, b_1=2,$ $a_2=4, b_2=-2$	$\begin{cases} \sum_{i=1}^{2} a_i(y-b_i)-\frac{a_i}{2}, y>b_i \\ \sum_{i=1}^{2}-\frac{a_i}{2}, y \le b_i \end{cases}$, $a_1=1, b_1=1,$ $a_2=1, b_2=-1$

4. 결합된 파라메트릭 활성함수의 합성곱 신경망 성능 실험

결합된 파라메트릭 활성함수를 적용한 합성곱 신경망의 성능을 확인하기 위해 3가지 실험을 진행하였다.

1. 합성곱층의 특징 추출 성능을 보기위한 실험.
2. 파라미터가 적용된 다른 활성함수들과 성능 비교 실험.
3. Tanh, ReLU 함수와 결합된 파라메트릭 활성함수의 성능 비교 실험.

합성곱 구조에서 활성함수의 파라미터 적용으로 인한 과적합을 관찰하기 위해 결합된 파라메트릭 활성함수의 파라미터 수의 사용으로 4.1,2절의 실험에서는 이미지의 채널×가로×세로 수만큼 적용하였고, 4.3절에서는 채널 수만 적용하였다.

4.1 합성곱층의 특징 추출 성능 비교 실험

합성곱층의 특징 추출이 잘 되었는지 확인하기 위해 MNIST 숫자 데이터셋을 사용하였고 Fig. 9와 같이 합성곱 출력층을 2차원 $[z_1^{(1)} z_2^{(1)}]$으로 시각화할 수 있는 구조를 사용하였다. 세부적으로 필터 개수가 16, 8, 4, 2인 4개의 합성곱층을 사용하고 완전연결신경망에서는 선형변환만 사용하였다.

실험에 비교된 활성함수는 총 4가지로 일반적으로 사용되는 활성함수인 Sigmoid, ReLU 함수와 $k=2$인 결합된 파라메트릭 활성함수를 사용하였다(Table 1).

합성곱 신경망의 학습은 Adam 알고리즘을 사용하여 훈련데이터를 128개의 배치크기로 10에폭 학습하였다. 손실함수는 크로스엔트로피를 사용하였으며 모든 선형변환 파라미터 W는 동일하게 초기화하였다.

학습 결과를 나타내는 Fig. 10은 배치크기로 훈련데이터를 학습할 때마다 학습에 전혀 개입하지 않은 시험데이터 10,000개에 대한 손실함수를 나타낸 것으로 Sigmoid와 ReLU함수보다 결합된 파라메트릭 Sigmoid와 결합된 파라메트릭 ReLU함수가 학습과정에서 손실함수가 빠르게 감소하는 것을 볼 수 있다. 학습이 끝난 후 시험데이터에 대한 손실함수 값은 Sigmoid는 0.687, ReLU는 0.627, 결합된 파라메트릭 Sigmoid와 결합된 파라메트릭 ReLU는 각각 0.352, 0.297로 파라미터를 적용하지 않은 활성함수보다 월등히 낮은 것을 확인할 수 있다.

학습 후 합성곱층을 통해 추출된 특징 $[z_1^{(1)} z_2^{(1)}]$이 Table 1의 4가지 활성함수에 따라 시험데이터가 어떤 분포를 형성하는지 보기위해 MNIST 시험데이터 0~9까지 숫자를 각각

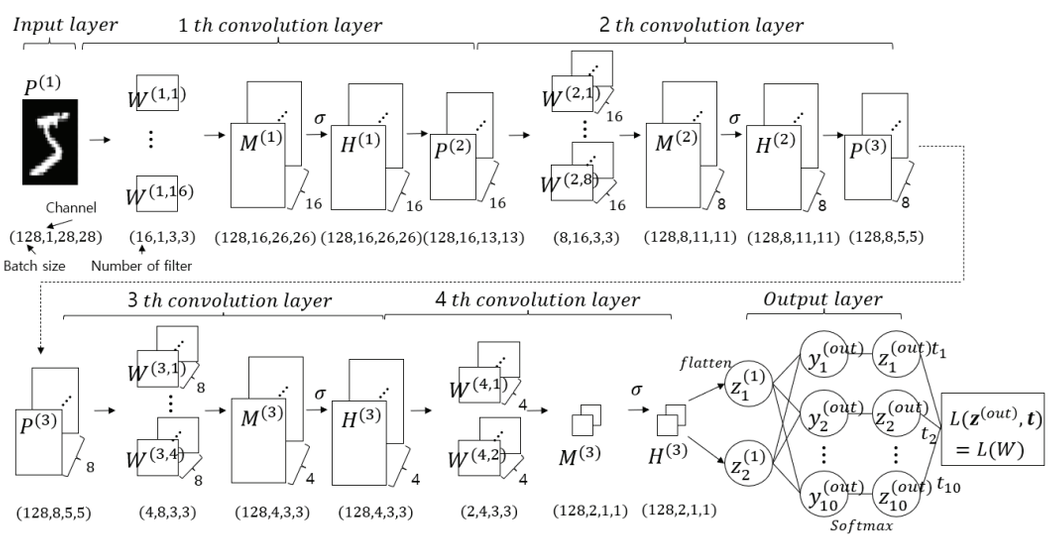


Fig. 9. Convolutional Neural Network Structure used in the Experiment

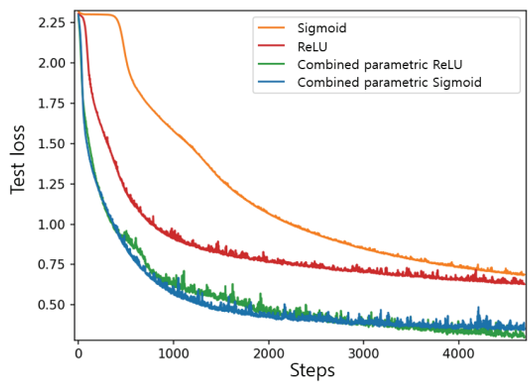


Fig. 10. Loss Function Values for Test Data

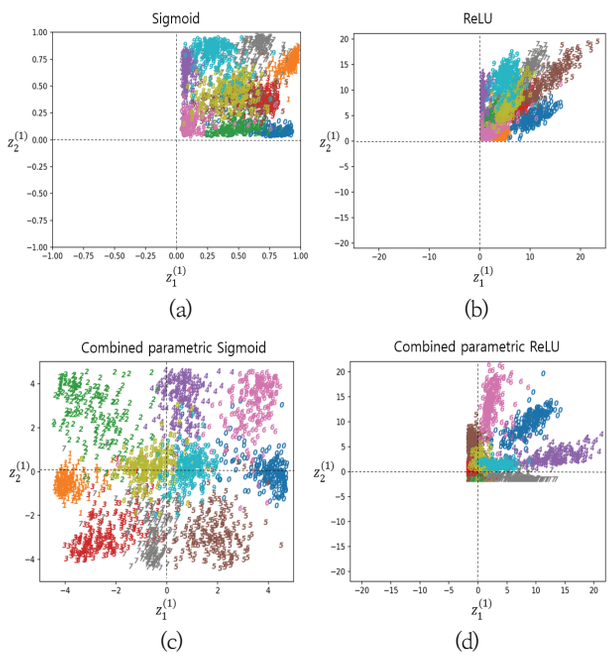


Fig. 11. Distribution of 2,000 MNIST Test Data Corresponding to $[z_1^{(1)} z_2^{(1)}]$

200개씩 총 2,000개를 시각화하였다(Fig. 11).

Sigmoid(Fig. 11(a))와 ReLU함수(Fig. 11(b))를 사용한 경우 $[z_1^{(1)} z_2^{(1)}]$에 대응하는 값들은 모두 1사분면인 양수를 취하는 반면 결합된 파라메트릭 Sigmoid(Fig. 11(c))와 결합된 파라메트릭 ReLU(Fig. 11(d))는 모든 사분면에 분포되어 있으며 시각적으로 보다 구분이 뚜렷한 것을 확인할 수 있다. 이후 Fashion MNIST을 사용하는 실험에서 결합된 파라메트릭 활성함수의 파라미터들이 학습된 값들을 자세히 제시하였다.

4.2 파라미터를 적용한 활성함수들과 결합된 파라메트릭 활성함수의 성능 비교

파라미터를 적용한 활성함수들과 결합된 파라메트릭 활성함수의 합성곱 신경망의 성능을 비교하기 위해 ReLU 함수의 변형인 He 등[16]이 연구한 PReLU와 Clevert 등[17]이 연구한 ELU 함수를 실험에 사용하였다.

Fashion MNIST 데이터셋을 사용하였으며 결합된 파라메트릭 활성함수는 Table 1과 동일하게 사용하였고 PReLU와 ELU의 파라미터 α값의 초기화는 해당 논문에 기재된 초기화 값을 사용하였다.

실험에 사용된 합성곱 신경망의 구조는 필터 개수가 16, 32, 4인 3개의 합성곱층과 노드 수가 30개인 은닉층 하나를 사용한 완전연결신경망을 사용하였다(Fig. 12).

실험결과, 학습에 전혀 개입하지 않은 시험데이터 10,000개에 대한 Step마다 손실함수를 나타낸 Fig. 13에서 PReLU와 ELU보다 결합된 파라메트릭 Sigmoid와 결합된 파라메트릭 ReLU가 각각 0.325, 0.336으로 낮은 손실함수 값을 가진다. 특히 0~1,000번째 Step에서 PReLU와 ELU에 비해 결합된 파라메트릭 ReLU는 상대적으로 빠른 속도로 손실함수 값이 감소하였다.

Table 2는 학습이 끝난 후 시험데이터 10,000개에 대한 정확도를 나타낸 것으로 결합된 파라메트릭 Sigmoid와 결합된 파라메트릭 ReLU가 각각 87.92%, 88.12%로 86.69%인 PReLU와 87.05%인 ELU보다 높은 정확도를 가진다.

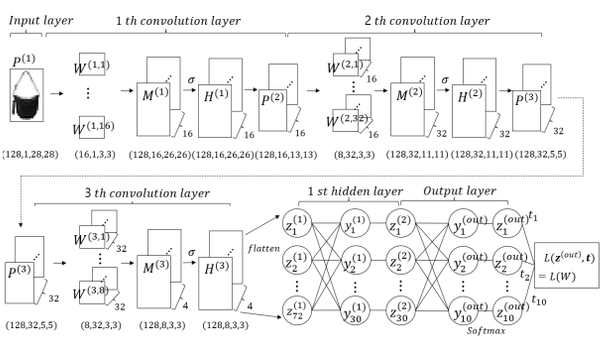


Fig. 12. Convolutional Neural Network Structure used in the Experiment

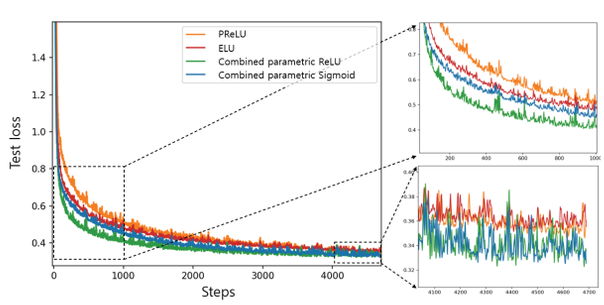


Fig. 13. Loss Function Value for Test Data

Table 2. Test Data Accuracy

Activation function	Test accuracy
PReLU	86.69%
ELU	87.05%
CPAF Sigmoid	87.92%
CPAF ReLU	88.12%

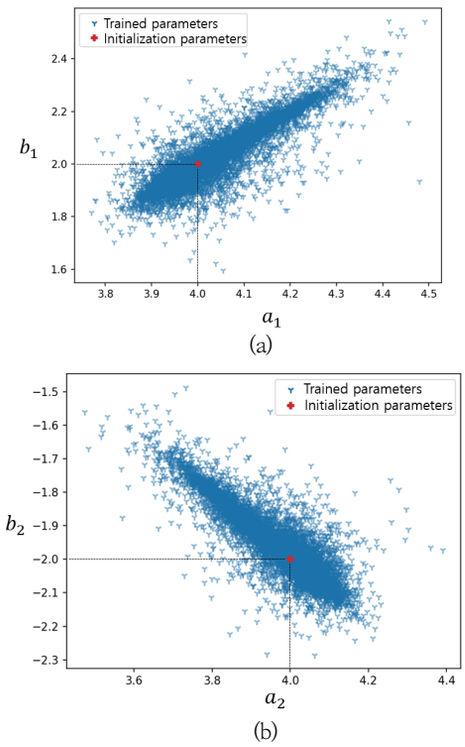


Fig. 14. Distribution of Parameter Values (a_i,b_i) of Combined Parametric Sigmoid (a) : (a_1,b_1), (b) : (a_2,b_2)

이처럼 낮은 손실함수 값을 갖는 결합된 파라메트릭 활성함수를 구성하고 있는 파라미터 a_i,b_i의 학습된 값을 보기위해 실험에 사용된 $k=2$인 결합된 파라메트릭 Sigmoid의 파라미터 a_i,b_i을 Fig. 14에 나타내었다. Fig. 14는 Fig. 12와 같은 합성곱 신경망 구조의 모든 합성곱층과 은닉층의 결합된 파라메트릭 활성함수 파라미터 $(a_1,b_1),(a_2,b_2)$쌍을 그린 것으로 각각 14,790개를 가진다.

Fig. 14(a)에서 (a_1,b_1)의 초기값 (4,2)이 Equation (21-24) 같이 경사하강법을 통해 학습 후 a_1값이 약 3.75~4.5, b_1값이 약 1.6~2.55로 분포된 것을 볼 수 있다. Fig. 14(b)도 마찬가지로 초기값(4,-2)에서 학습하여 (a_2,b_2)값들이 이동하였으며 이는 손실함수를 최소화하는 방향으로 학습된 서로 다른 $(a_1,b_1),(a_2,b_2)$쌍이 Fig. 6(b)의 a_i,b_i값에 따라 다양한 비선형 변환을 만들어내는 것을 의미하며 결과적으로 PReLU, ELU보다 우수한 분류성능을 가짐을 확인하였다.

4.3 Tanh, ReLU 함수와 결합된 파라메트릭 활성함수의 성능 비교

마지막으로 일반적으로 많이 사용되는 함수인 Tanh, ReLU 함수를 결합된 파라메트릭 활성함수와 비교하는 실험을 진행하였다.

CIFAR10, CIFAR100 데이터셋에 대해 Table 3과 같은 임의의 3가지 합성곱 신경망 구조를 4가지 활성함수에 대해 학습하였다. 앞서 진행했던 실험들과 다르게 이번 실험에서는 과적합을 우려하여 합성곱 구조에서 채널별로 파라미터 수를 적용하여 파라미터 수를 감소시켜 사용하였다.

Table 3. Structures used in Experiments

Structure 1	Structure 2	Structure 3
input$(N,32,32,3)$		
conv(32) BN	conv(32)	conv(32) BN
conv(32) BN	conv(32)	conv(32) BN
maxpool dropout		
conv(64) BN	conv(64)	conv(64) BN
conv(64) BN	conv(64)	conv(64) BN
maxpool dropout		
conv(128) BN	conv(64)	conv(64) BN
conv(128) BN	conv(64)	conv(64) BN
maxpool dropout		
Flatten(2048)	Flatten(1024)	Flatten(1024)
softmax	FC(512) dropout	FC(512) dropout
	FC(512) dropout	FC(512) dropout
	softmax	softmax

결합된 파라메트릭 활성함수의 초기화는 Table 1과 같고 학습은 RMSprop 알고리즘, 배치사이즈는 128개, 100에폭을 진행하였다.

Table 4,5는 시험 데이터에 대한 CIFAR10, CIFAR100 데이터셋의 정확도를 보여준다. Tanh와 결합된 파라메트릭 Sigmoid를 비교하고, ReLU와 결합된 파라메트릭 ReLU을 비교하였을 때, 두 데이터셋에 대해 3가지 합성곱 신경망 구조 모두 결합된 파라메트릭 활성함수의 시험 정확도가 높은 것을 확인할 수 있다. 특히 Table 4의 Structure 2에서 ReLU와 결합된 파라메트릭 ReLU의 정확도 차이는 약 18%로 크게 차이나며 이에 대한 시험 데이터 정확도 그래프는 Fig. 15와 같다.

Table 4. Test Data Accuracy for the CIFAR10 Dataset

	Tanh	CPAF Sigmoid	ReLU	CPAF ReLU
Structure 1	82.60%	83.20%	85.80%	86.50%
Structure 2	76.34%	78.93%	67.82%	85.72%
Structure 3	82.03%	82.35%	81.40%	85.60%

Table 5. Test Data Accuracy for the CIFAR100 Dataset

	Tanh	CPAF Sigmoid	ReLU	CPAF ReLU
Structure 1	53.90%	55.50%	57.70%	58.90%
Structure 2	39.43%	47.79%	37.16%	56.25%
Structure 3	48.01%	51.31%	47.96%	58.17%

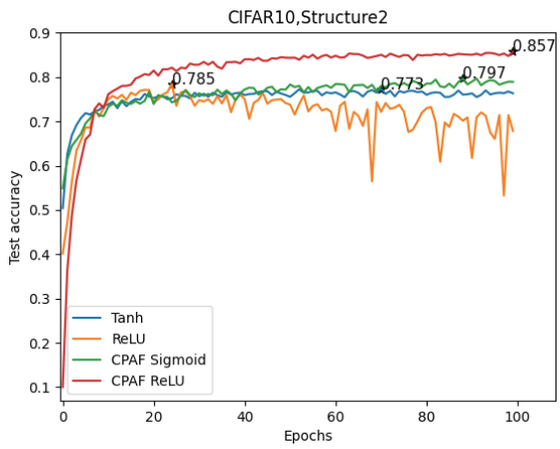


Fig. 15. Result of Structure 2 using CIFAR10 Dataset

Table 6. The Learning Time(Seconds) of Structure 1

		Tanh	CPAF Sigmoid	ReLU	CPAF ReLU
CIFAR 10	Time taken up to 100 epochs	610	875	601	866
	Time taken when test accuracy is 80%	195	140	60	77
CIFAR 100	Time taken up to 100 epochs	617	865	601	881
	Time taken when test accuracy is 50%	111	173	60	79

실험에 사용된 활성함수들의 계산 시간을 보기위해 Structure 1에 대해 CIFAR10, 100을 학습한 시간을 Table 6에 나타내었다. 100에폭을 진행하는데 걸린 시간은 모두 Tanh, ReLU 함수가 적게 걸렸지만, 성능 관점에서 CIFAR 10의 경우 시험 정확도가 80%을 도달하는데 걸린 시간은 오히려 결합된 파라메트릭 Sigmoid가 Tanh보다 적게 걸린 것을 볼 수 있다.

5. 결 론

본 논문은 합성곱 신경망에서 사용되는 Sigmoid 혹은 ReLU함수와 같은 파라미터가 없는 비선형활성함수가 손실함수와 무관한 비선형변환을 하는 문제를 개선하여 합성곱 신경망의 성능을 향상시키기 위해 결합된 파라메트릭 활성함수를 제안하였다.

활성함수의 크기와 위치를 변환하는 파라미터를 도입한 파라메트릭 활성함수들을 여러 번 더하여 만들어진 결합된 파라메트릭 활성함수는 여러 개의 크기, 위치를 변환하는 파라미터에 따라 다양한 비선형간격을 만들 수 있으며, 파라미터는 주어진 입력데이터에 의해 계산된 손실함수를 최소화하는 방향으로 학습할 수 있다.

결합된 파라메트릭 활성함수를 사용한 합성곱 신경망의 성능을 다른 활성함수들과 비교하기 위해 MNIST, Fashion MNIST, CIFAR10 그리고 CIFAR100 데이터셋을 사용하였다.

Sigmoid와 ReLU함수에 대해 결합된 파라메트릭 활성함수를 적용한 경우를 MNIST 분류문제에 대해 비교하였고 실험결과 파라미터를 적용하지 않은 활성함수보다 결합된 파라메트릭 활성함수가 손실함수 감소 속도와 낮은 손실함수에서 모두 우수한 성능을 가짐을 확인하였다. 특히 합성곱 신경망 구조에서 합성곱층을 통해 추출된 특징 $[z_1^{(1)} z_2^{(1)}]$을 시험데이터 2,000개에 대해 시각화한 결과 결합된 파라메트릭 Sigmoid가 Sigmoid, ReLU함수에 비해 뚜렷하게 구분되었다.

파라미터를 적용한 다른 활성함수와 결합된 파라메트릭 활성함수를 비교하기 위해 PReLU, ELU와 결합된 파라메트릭 Sigmoid, 그리고 결합된 파라메트릭 ReLU를 Fashion MNIST 분류문제에 대해 실험하였고 시험데이터에 대한 정확도가 결합된 파라메트릭 Sigmoid와 결합된 파라메트릭 ReLU가 각각 87.92%, 88.12%로 86.69%를 가지는 PReLU, 87.05%를 가지는 ELU보다 높은 정확도를 가졌다.

마지막으로 CIFAR10, CIFAR100 데이터셋에 대해 3가지 합성곱 신경망 구조를 Tanh, ReLU 그리고 결합된 파라메트릭 활성함수에 대해 비교하였고 그 결과, 결합된 파라메트릭 활성함수가 Tanh, ReLU 보다 높은 시험 데이터 정확도를 가지는 것을 확인하였다.

이상의 실험들에 대해서 결합된 파라메트릭 활성함수가 합성곱 신경망의 성능을 향상시킬 수 있음을 보았다.

References

[1] Y. Bengio, I. Goodfellow, and A. Courville, “Deep learning,” MIT Press, 2017.

[2] C. A. Charu, “Neural Networks and Deep Learning: A Textbook,” Springer International Publishing AG, 2018.

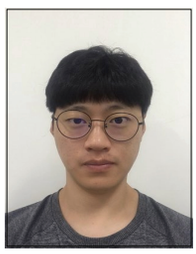
[3] Y. M. Ko, P. H. Li, and S. W. Ko, “Performance improvement method of fully connected neural network using combined parametric activation functions,” *KIPS Transactions on Software and Data Engineering*, Vol.11, No.1, pp.1-10, 2022.

[4] N. Y. Kong and S. W. Ko, “Performance improvement method of deep neural network using parametric activation functions,” *Journal of the Korea Contents Association*, Vol.21, No.3, pp616-625, 2021.

[5] N. Y. Kong, Y. M. Ko, and S. W. Ko, “Performance improvement method of convolutional neural network using agile-activation function,” KIPS Transactions on Software and Data Engineering, Vol.9, No.7, pp.213-220, 2020.

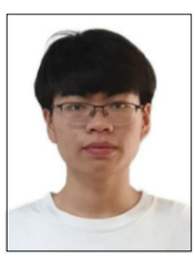
[6] Y. M. Ko and S. W. Ko, “Alleviation of vanishing gradient problem using parametric activation functions,” *KIPS Transactions on Softward and Data Engineering*, Vol.10, No. 10, pp.407-420, 2021.

[7] A. Apicella, F. Donnarumma, F. Isgro, and R. Prevete, "A survey on modern trainable activation functions," *Neural Networks*, Vol.138, pp.14-32, 2021.
[8] V. Nair and G. Hinton, "Rectified linear units improve restricted boltzmann machines," In *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML)*, pp.807-814, 2010.
[9] M. Roodschild, J. Gotay Sardiñas, and A. Will, "A new approach for the vanishing gradient problem on sigmoid activation," *Springer Nature*, Vol.20, Iss.4, pp.351-360, 2020.
[10] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol.6, No.2, pp.107-116, 1998.
[11] S. Kong and M. Takatsuka, "Hexpo: A vanishing-proof activation function," *International Joint Conference on Neural Networks*, pp.2562-2567, 2017.
[12] Y. Qin, X. Wang, and J. Zou, "The optimized deep belief networkswith improved logistic Sigmoid units and their application in faultdiagnosis for planetary gearboxes of wind turbines," *IEEE Transactions on Industrial Electronics*, Vol.66, No.5, pp.3814-3824, 2018.
[13] X. Wang, Y. Qin, Y. Wang, S. Xiang, and H. Chen, "ReLTanh: An activation function with vanishing gradient resistance for SAE-based DNNs and its application to rotating machinery fault diagnosis," *Neurocomputing*, Vol.363, pp.88-98, 2019.
[14] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv: 1505.00853*, 2015.
[15] S. Qian, H. Liu, C. Liu, S. Wu, and H. Wong, "Adaptive activation functions in convolutional neural networks," *Neurocomputing*, Vol.272, pp.204-212, 2017.
[16] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *arXiv:1502.01852*, 2015.
[17] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," *arXiv:1511.07289*, 2016.



고 영 민

https://orcid.org/0000-0003-2779-3170
e-mail : gjtrj55@naver.com
2020년 전주대학교 경영학과(학사)
2022년 전주대학교 인공지능학과(석사)
2022년 ~ 현 재 전주대학교 인공지능연구소 연구교수
관심분야 : Data Science & Artificial Intelligence



이 붕 항

https://orcid.org/0000-0001-8496-8067
e-mail : lipenghang@jj.ac.kr
2020년 전주대학교 스마트미디어학과(학사)
2022년 전주대학교 인공지능학과(석사)
관심분야 : Digital Image Processing & Artificial Intelligence

고 선 우

https://orcid.org/0000-0002-6328-5440
e-mail : godfriend0@gmail.com
1985년 고려대학교 산업공학과(학사)
1988년 한국과학기술원 산업공학과(석사)
1992년 한국과학기술원 산업공학과(박사)
2005년 ~ 현 재 전주대학교 인공지능학과 교수, 전주대학교 인공지능연구소 연구소장
관심분야 : Data Science & Artificial Intelligence