KIPS Trans. Softw. and Data Eng.
Vol.9, No.7 pp.213~220 pISSN: 2287-5905


https://doi.org/10.3745/KTSDE.2020.9.7.213

# Performance Improvement Method of Convolutional Neural Network Using Agile Activation Function

Na Young Kong† · Young Min Ko†† · Sun Woo Ko†††

## ABSTRACT

The convolutional neural network is composed of convolutional layers and fully connected layers. The nonlinear activation function is used in each layer of the convolutional layer and the fully connected layer. The activation function being used in a neural network is a function that simulates the method of transmitting information in a neuron that can transmit a signal and not send a signal if the input signal is above a certain criterion when transmitting a signal between neurons. The conventional activation function does not have a relationship with the loss function, so the process of finding the optimal solution is slow. In order to improve this, an agile activation function that generalizes the activation function is proposed. The agile activation function can improve the performance of the deep neural network in a way that selects the optimal agile parameter through the learning process using the primary differential coefficient of the loss function for the agile parameter in the backpropagation process. Through the MNIST classification problem, we have identified that agile activation functions have superior performance over conventional activation functions.

Keywords : Convolutional Neural Network, Agile Activation Function, Backpropagation, Learning

# 민첩한 활성함수를 이용한 합성곱 신경망의 성능 향상

공 나 영† · 고 영 민†† · 고 선 우†††

## 요 약

합성곱 신경망은 합성곱층과 완전연결층으로 구성되어 있다. 합성곱층과 완전연결층의 각 층에서는 비선형 활성함수를 사용하고 있다. 활성함수는 뉴런 간에 신호를 전달할 때 입력신호가 일정 기준 이상이면 신호를 전달하고 기준에 도달하지 못하면 신호를 보내지 않을 수 있는 뉴런의 정보전달 방법을 모사하는 함수이다. 기존의 활성함수는 손실함수와 관계성을 가지고 있지 않아 최적해를 찾아가는 과정이 늦어지는 점을 개선하기 위해 활성함수를 일반화한 민첩한 활성함수를 제안하였다. 민첩한 활성함수의 매개변수는 역전파 과정에서, 매개변수에 대한 손실함수의 1차 미분계수를 이용한 학습과정을 통해 최적의 매개변수를 선택하는 방법으로 손실함수를 감소시킴으로써 심층신경망의 성능을 향상시킬 수 있다. MNIST 분류문제를 통하여 민첩한 활성함수가 기존의 활성함수에 비해 우월한 성능을 가짐을 확인하였다.

키워드 : 합성곱 신경망, 민첩한 활성함수, 역전파, 학습

## 1. 서 론

합성곱 신경망은 그 성능이 입증된 심층신경망의 한 방법으로 이미지분류, 이미지 속에 포함되어 있는 특정 대상물의 탐지, 대상물의 위치 등을 파악하는데 사용되고 있다.

합성곱 신경망은 고양이의 시각피질이 작동하는 방식에 대한 허블과 비셀의 연구에서 이미지의 특정 부분이 시각 뉴런의 특정 부분을 활성화한다는 연구결과에 아이디어를 얻어 만들어진 것이다[1]. Fig. 1은 일반적인 합성곱 신경망을 나타낸 것으로 합성곱층과 완전연결층으로 구성되어 있다. 1개의 이미지 입력층, $K$개의 합성곱층, $M$개의 은닉층을 갖는 완전연결층과 출력층을 가진 합성곱 신경망이다.

합성곱층은 입력층의 입력 $I$의 특성을 추출하는 합성곱 연산을 수행하는 $K$개의 층을 거쳐 특성맵 $P^{(K)}$를 추출한다. 최종 특성맵인 $P^{(K)}$를 1차원 벡터로 변환하여 완전연결층의 입력값으로 사용한다. 완전연결층의 입력데이터는 $M$개의 은닉층을 거치면서 입력으로부터 출력층으로 연결되는 모든 노드 간의 가중치를 이용하여 최종출력 $\hat{y}_i^{(out)}$, $i=1,...,n_{out}$을 계산한다. 이때 $K$개의 합성곱층과 $M$개의 완전연결층에서 활성함

† 정 회 원 : 전주대학교 문화기술학과 박사과정
†† 준 회 원 : 전주대학교 인공지능학과 석사과정
††† 정 회 원 : 전주대학교 스마트미디어학과 교수
Manuscript Received : March 27, 2020
First Revision : May 11, 2020
Accepted : June 1, 2020
* Corresponding Author : Na Young Kong(lindsey0@hanmail.net)

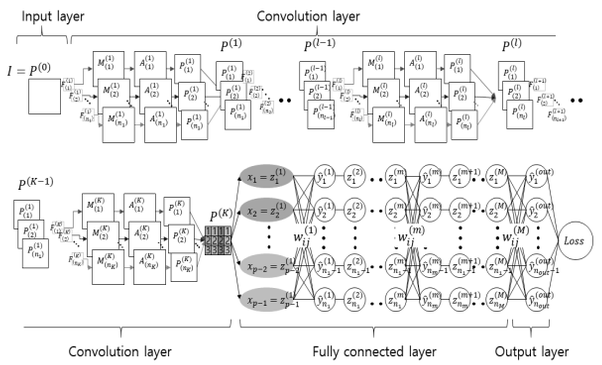



Fig. 1. Structure of Convolutional Neural Network

수를 사용하고 각 합성곱층과 완전연결층에서 비선형 함수인 $ReLU$의 사용이 권장되고 있다[2]. $ReLU$ 활성함수가 속도와 정확도 면에서 다른 활성함수보다 탁월한 성능을 발휘한다고 알려져 있다. 계산속도가 빠른 $ReLU$ 활성함수를 사용하게 되면 더 깊은 층을 가지는 네트워크 모델을 더 많이 학습시킬 수 있다는 점에서 다른 활성함수들을 거의 대체한 상태이다[3].

활성함수에 대한 서베이논문으로 Nwankpa et.al[4]이 있으며 활성함수별 특성을 자세히 설명하고 있어서, 실제 문제 해결 적용시에 어떤 활성함수를 선택할 것인가에 대한 지침이 될 수 있다.

본 논문은 합성곱층과 완전연결층에서 사용되고 있는 활성함수의 의미를 살펴보고 활성함수의 성능을 향상시킬 수 있는 민첩한 활성함수를 제안한다.

## 2. 활성함수

### 2.1 활성함수의 종류

활성함수로 다양한 함수를 사용하고 있다[3]. 일반적으로 널리 사용되는 비선형 변환의 활성함수로는 다음 Equation (1)의 $ReLU$, Equation (2)의 $Sigmoid$가 있다.

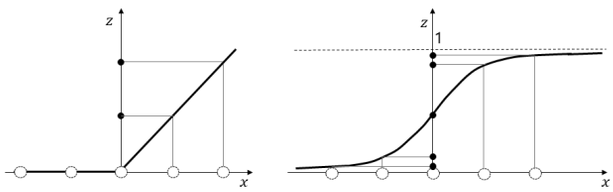
$$ReLU \text{ 활성함수: } z=\begin{cases}x, & x \geq 0\\ 0, & x<0\end{cases} \tag{1}$$

$$Sigmoid \text{ 활성함수: } z=\frac{1}{1+e^{-x}} \tag{2}$$

Fig. 2는 비선형 활성함수 중에 대표적인 $Sigmoid$ 활성함수와 $ReLU$ 활성함수를 나타낸 것이다.

$Sigmoid$ 활성함수는 1차 미분값의 최대값이 0.25이므로 심층신경망에서 은닉층의 깊이가 깊어질수록 기울기가 소멸(Gradient vanishing)되는 문제가 발생할 수 있다. 이러한 기울기 소멸 문제를 해결하고 빠른 연산속도를 보장하는 $ReLU$ 활성함수가 널리 사용되고 있다.

이러한 활성함수의 성능을 개선하기 위해 심층신경망에서 기존 활성함수의 적용결과가 손실함수를 감소하는 방향과 무관하다는 관점에서 활성함수를 매개변수를 이용하여 일반화




(a) (b)

Fig. 2. (a) Activation Function $ReLU$
(b) Activation Function $Sigmoid$

하고 매개변수를 학습하는 방법을 제안하였다[5].

### 2.2 합성곱 신경망과 활성함수의 기능

활성함수는 뉴런의 정보전달과정을 모사한 것으로 뉴런으로 입력되는 정보의 크기에 따라 출력으로 내보낼 것인지, 출력으로 내보낸다면 어떤 크기로 내보낼 것인지를 결정하는 함수이다.

합성곱층의 활성함수는 각 합성곱층에서 계산되는 합성곱 연산의 결과인 특성맵을 활성화시키기 위해 사용되고, 완전연결층의 활성함수는 각 은닉층의 선형 모델에 의해 추정된 결과를 활성화시키기 위해 사용한다. 각 합성곱층에서 특성맵을 활성화한다는 의미와 완전연결층에서 선형 모델의 추정치를 활성화한다는 의미는 서로 다르다.

$l^{th}$ 합성곱층의 특성맵은 Equation (3)과 같이 구해진다.

$$\begin{aligned} M_{(k),(i,j)}^{(l)} &= P_{(i,j)}^{(l-1)} \otimes F_{k}^{(l)} \\ &= \sum_{j=-\lfloor f_l/2 \rfloor}^{\lfloor f_l/2 \rfloor} \sum_{i=-\lfloor f_l/2 \rfloor}^{\lfloor f_l/2 \rfloor} P_{(i,j)}^{(l-1)} F_{(k),(\lfloor f_l/2 \rfloor+1+i,\, \lfloor f_l/2 \rfloor+1+j)}^{(l)} \end{aligned} \tag{3}$$

$P^{(l-1)}$는 $l^{th}$ 합성곱층의 입력데이터이고, 필터 $F_{(k)}^{(l)}$은 $l^{th}$ 합성곱층의 $k^{th}(k=1,...,n_l)$ 필터로 $f_l \times f_l$ 행렬이다.

Equation (3)의 합성곱 연산은 Fig. 3과 같이 필터에 대응하는 입력 영역을 일정 간격씩 이동해 가며 계산된다. 합성곱 연산의 결과는 $P^{(l-1)}$의 특정 영역에 필터 $F_{(k)}^{(l)}$의 검출속성과 일치하는 패턴이 존재할 때 큰 값을 가진다.

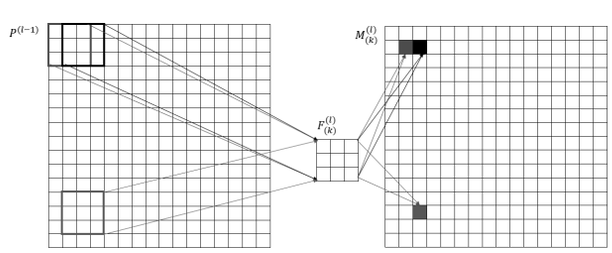



Fig. 3. Process of Convolution

합성곱층의 활성함수는 필터의 속성과 일치하는 입력이 존재하는 경우 이를 출력으로 내보내고 필터의 속성과 일치하지 않아 합성곱 결과가 작을 때 출력을 내보내지 않거나 작은 값을 내보내는 역할을 한다.

완전연결층의 활성함수는 Equation (4)와 같이 계산되는 은닉층의 선형 모델의 추정치 $\hat{y}_i^{(m)}$를 활성화하기 위해 사용된다.

$$\hat{y}_i^{(m)} = \sum_{i=1}^{n_m} \hat{w}_{ij}^{(m)} z_i^{(m)} \tag{4}$$

$z_i^{(m)}$는 $m^{th}$ 은닉층에서 입력값이고 $\hat{w}_{ij}^{(m)}$는 $m^{th}$ 은닉층에서 $z_i^{(m)}$과 $\hat{y}_j^{(m)}$를 연결하는 가중치의 추정치이다.

선형 모델만으로 해결할 수 없는 문제의 경우, 즉 선형 모델 등만을 이용하여 추정한 최종 결과치 $\hat{y}_i^{(M)}$가 참값인 $t_i$에 충분히 접근하지 못하는 경우, 모델의 성능 향상을 위해서 다음과 같은 3가지 방법을 이용하여 모델의 성능을 향상시킬 수 있다.

1) 선형 모델 대신 비선형 모델을 사용하는 방법[6]
2) 커널 함수를 이용하여 차원을 높이는 방법[7]
3) 비선형 변환함수인 활성함수를 사용하는 방법[5]

등을 사용한다.

비선형 모델을 사용할 경우 $y$와 $X$의 관계를 나타내는 모델 $y=f(X,w)$에서 추정해야하는 파라미터 $w$의 원소 수가 급격히 증가하는 문제와 파라미터 추정치의 분산이 증가할 위험이 있다. 커널 함수를 이용하여 데이터의 차원을 높이는 방법을 이용해 문제를 해결하는 방법은 최적의 파라미터를 찾기 위한 탐색 공간의 차원을 높이는 방법을 사용하기 때문에 희소 데이터집합(sparse dataset)의 문제 뿐 아니라 계산에 필요한 메모리의 크기가 급격히 증가하는 문제가 발생한다. 심층신경망에서는 비선형 문제 풀이를 위해 선형 모델과 선형 모델로 추정된 결과값 $\hat{y}_i^{(m)}$에 비선형 활성함수를 반복 적용하여 사용한다. Fig. 4는 선형 모델로는 해결할 수 없는 XOR문제를 은닉층에서 비선형 활성함수를 이용하여 XOR 문제를 해결하는 과정을 그림으로 도식화한 것이다.

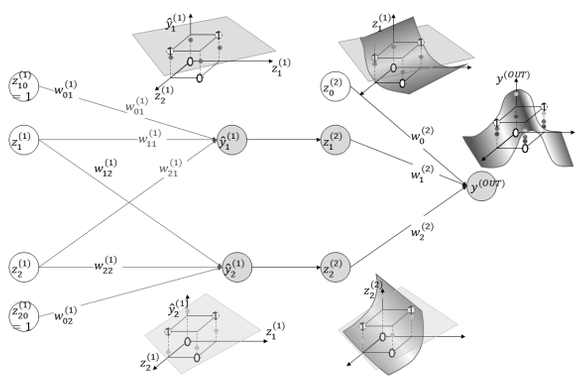


Fig. 4. Nonlinear Problem(XOR) Solution Process using Activation Function

Fig. 4에서 좌측상단의 $\hat{y}_1^{(1)}$의 그림은 선형 모델 $y_1^{(1)} = w_{01}^{(1)} + w_{11}^{(1)} z_1^{(1)} + w_{21}^{(1)} z_2^{(1)} + \epsilon_1^{(1)}$에서, 파라미터 $w_{01}^{(1)}$, $w_{11}^{(1)}$, $w_{21}^{(1)}$들을 최소제곱법으로 추정을 사용했다면 오차항 $\epsilon_1^{(1)}$의 제곱합을 최소화하는 $w_{01}^{(1)}$, $w_{11}^{(1)}$, $w_{21}^{(1)}$의 추정치를 이용하여 추정한 $\hat{y}_1^{(1)}$의 초평면을 나타낸 것이다. 이렇게 추정된 초평면 $\hat{y}_1^{(1)}$를 비선형 활성함수를 적용하여 구한 $z_1^{(2)}$ 초평면이 우측 상단의 그림이다. $z_1^{(2)}$ 초평면은 비선형 활성함수 적용에 의해 휘어진 곡면이다. 하단의 2개의 그림도 같은 방법에 의해 구해진 것이다. 이렇게 구해진 2개의 휘어진 곡면의 선형 결합을 통해 $y^{(OUT)}$이 구해지고 XOR문제를 해결할 수 있다. 이와 같이 비선형 활성함수는 비선형 모델을 이용해야 해결할 수 있는 문제를 선형 모델과 간단한 비선형 함수를 이용하여 해결하는 방법을 제시한 것이다.

## 3. 합성곱 신경망과 민첩한 활성함수

### 3.1 기존 활성함수의 문제점

합성곱층과 완전연결층에서 사용되는 활성함수 적용 시 문제점은 손실함수의 최소화와 어떤 관계도 없는 비선형 변환이라는 점이다.

Fig. 1의 각 합성곱층에 존재하는 필터들 $F_{(k)}^{(l)}$, $l=1,...,K$, $k=1,...,n_k$와 완전연결층의 노드간의 가중치 $w_i^{(m)}$, $m=1,...,M$, $i=1,...,n_m$들은 순전파 과정과 역전파 과정을 통해 Equation (5)과 Equation (6)과 같이 경사하강법을 통해 최적화된다.

$$F_{(k),(i,j)}^{(l)} = F_{(k),(i,j)}^{(l)} - \rho_1 \nabla L(F_{(k),(i,j)}^{(l)}) \tag{5}$$

$$w_{(i,j)}^{(l)} = w_{(i,j)}^{(l)} - \rho_2 \nabla L(w_{(i,j)}^{(l)}) \tag{6}$$

여기서 $\rho_1$과 $\rho_2$는 학습률이고 $\nabla L(F_{(k),(i,j)}^{(l)})$는 필터 파라미터 $F_{(k),(i,j)}^{(l)}$에서 손실함수 $L$의 미분계수이고 $\nabla L(w_{(i,j)}^{(l)})$는 가중치 $w_{(i,j)}^{(l)}$에서 손실함수 $L$의 미분계수이다. Equation (5)와 (6)의 최적화 과정은 손실함수 $L$이 볼록함수일 때, Equation (5)와 (6)을 이용한 최적화 과정에서 손실함수 값을 지속적으로 감소시키게 된다.

Fig. 5는 합성곱층 또는 완전연결층에서 파라미터(필터 또는 가중치)의 최적화 과정과 활성함수의 적용과정을 손실함수 관점에서 나타낸 것이다.

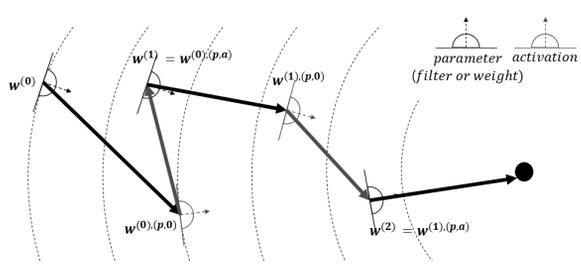



Fig. 5. Loss Value Changes According to Parameter Optimization and Activation Function Application

Fig. 5에서 $w^{(0)}$는 파라미터(합성곱층의 필터 또는 완전연결층의 노드간 가중치)의 초기 값을 의미하고 $w^{(0),(p,0)}$는 파라미터 초기 값에서 파라미터를 Equation (5) 또는 Equation (6)을 이용하여 최적화한 결과이고 $w^{(0),(p,a)}$는 파라미터 최적화 이후 활성함수를 적용한 결과이다. 하지만 기존의 활성함수는 손실함수 $L$과 어떤 함수관계도 가지지 않기 때문에 기존 활성함수 적용이 손실함수를 감소시킨다는 어떤 보장도 없는 비선형 변환일 뿐 이다. Fig. 5에서 $w^{(0),(p,0)}$에서 $w^{(1)=w^{(0),(p,a)}}$로의 변환과정은 활성함수를 적용한 결과를 의미하고 손실함수 값이 증가한 경우를 예시한 것이다. 이와 같이 기존의 활성함수는 특성맵의 각 원소 또는 완전연결층의 각 은닉층에서 선형 함수를 이용한 추정값을 활성화 시킨다는 관점에서는 타당하지만 손실함수를 증가시킬 수 있다는 점에서 개선이 필요하다.

### 3.2 활성함수를 일반화한 민첩한 활성함수의 개발

활성함수는 입력값을 비선형적으로 변환시키는 함수이다. 활성함수는 크기와 위치변화에 따라 입력값을 다양하게 변환할 수 있다. Fig. 6은 *ReLU*와 *Sigmoid* 함수의 크기와 위치변화를 통해 다양한 변환을 할 수 있음을 보여준다.

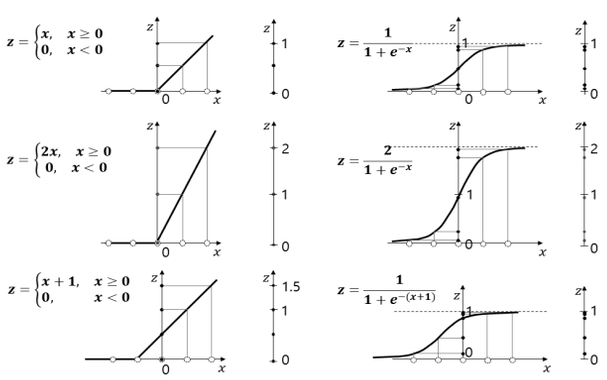



Fig. 6. Various Transformations Depending on Size and Position of the Activation Function

즉, 기존의 활성함수들은 Equation (7)과 (8)의 예와 같이 크기 및 위치를 변화시킬 수 있는 매개변수를 사용하여 입력값들을 보다 다양한 값으로 변환할 수 있는 민첩한 활성함수로 일반화 할 수 있다.

$$\text{민첩한 } Sigmoid \text{ 함수: } z=\frac{a}{1+e^{-(x-b)}} \tag{7}$$

$$\text{민첩한 } ReLU \text{ 함수: } z=\begin{cases}a(x-b), & x\geq b\\ 0, & x<b\end{cases} \tag{8}$$

이때 $a$와 $b$는 임의의 실수값이다. $a$는 민첩한 활성함수의 크기를 결정하는 매개변수이고 $b$는 민첩한 활성함수의 위치를 결정하는 매개변수이다.

즉, 민첩한 활성함수를 사용하면, 합성곱층의 합성곱 연산의 결과인 특성맵의 $(i,j)$원소 $M^{(l)}_{(k),(i,j)}$를 활성화하여 활성화맵의 $(i,j)$원소 $A^{(l)}_{(k),(i,j)}$를 계산할 때 보다 다양한 변환의 자유도를 확보할 수 있고 완전연결층의 선형 모델의 추정치를 활성화할 때 변환의 자유도를 확보할 수 있게 된다.

### 3.3 민첩한 활성함수의 학습

민첩한 활성함수가 손실함수와 연계되기 위해서는, 민첩한 활성함수의 매개변수 $a$와 $b$에 대한 손실함수의 변화율 $\partial L/\partial a$와 $\partial L/\partial b$을 계산하여 역전파 과정에서 $a$와 $b$를 갱신할 새로운 $a$와 $b$값을 찾는 방법을 사용할 수 있다.

Fig. 7은 $l^{th}$ 활성곱층에서 $k^{th}$ 필터를 이용해 구한 특성맵 $M^{(l)}_{(f)}$에 민첩한 활성함수를 적용한 것으로 특성맵 $M^{(l)}_{(f)}$의 각 원소 $(i,j)$에 대해 적용된다.

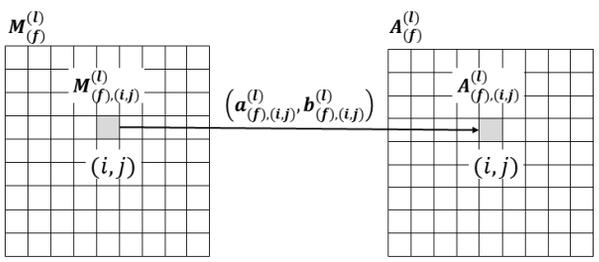



Fig. 7. Activation of the Characteristic Map Applying the $k^{th}$ Filler of the $l^{th}$ Convolutional Layer

Fig. 7은 $l^{th}$ 활성곱층에서 $k^{th}$ 필터를 이용해 구한 특성맵 $M^{(l)}_{(f)}$를 Equation (7)의 민첩한 *Sigmoid* 활성함수를 이용하여 변환하는 경우로 변환한 값 $A^{(l)}_{(k),(i,j)}$은 다음 식과 같다.

$$A^{(l)}_{(f),(i,j)}=\frac{a^{(l)}_{(f),(i,j)}}{1+e^{-(M^{(l)}_{(f),(i,j)}-b^{(l)}_{(f),(i,j)})}} \tag{9}$$

$A^{(l)}_{(f),(i,j)}$에 대한 $M^{(l)}_{(f),(i,j)}$, $a^{(l)}_{(f),(i,j)}$, $b^{(l)}_{(f),(i,j)}$의 미분은 다음과 같다.

$$\frac{\partial A^{(l)}_{(f),(i,j)}}{\partial M^{(l)}_{(f),(i,j)}}=M^{(l)}_{(f),(i,j)}\left(1-\frac{M^{(l)}_{(f),(i,j)}}{a^{(l)}_{(f),(i,j)}}\right) \tag{10}$$

$$\frac{\partial A^{(l)}_{(f),(i,j)}}{\partial a^{(l)}_{(f),(i,j)}}=\frac{1}{1+e^{-(M^{(l)}_{(f),(i,j)}-b^{(l)}_{(f),(i,j)})}} \tag{11}$$

$$\frac{\partial A^{(l)}_{(f),(i,j)}}{\partial b^{(l)}_{(f),(i,j)}}=A^{(l)}_{(f),(i,j)}\left(1-\frac{A^{(l)}_{(f),(i,j)}}{a^{(l)}_{(f),(i,j)}}\right) \tag{12}$$

손실함수 $L$에 대한 $a^{(l)}_{(f),(i,j)}$, $b^{(l)}_{(f),(i,j)}$의 미분계수는 연쇄법칙을 이용하여 Equation (13)과 (14)를 같이 구할 수 있다.

$$\begin{aligned}\frac{\partial L}{\partial a^{(l)}_{(k),(i,j)}}&=\frac{\partial A^{(l)}_{(k),(i,j)}}{\partial a^{(l)}_{(k),(i,j)}}\times\frac{\partial L}{\partial A^{(l)}_{(k),(i,j)}}\\&=\frac{1}{1+e^{-(M^{(l)}_{(f),(i,j)}-b^{(l)}_{(f),(i,j)})}}\times\frac{\partial L}{\partial A^{(l)}_{(f),(i,j)}}\end{aligned} \tag{13}$$

$$\frac{\partial L}{\partial b_{(k),(i,j)}^{(l)}} = \frac{\partial A_{(k),(i,j)}^{(l)}}{\partial b_{(k),(i,j)}^{(l)}} \times \frac{\partial L}{\partial A_{(k),(i,j)}^{(l)}}$$
$$= A_{(f),(i,j)}^{(l)}\left(1 - \frac{A_{(f),(i,j)}^{(l)}}{a_{(f),(i,j)}^{(l)}}\right) \times \frac{\partial L}{\partial A_{(k),(i,j)}^{(l)}} \quad (14)$$

합성곱층에서의 민첩한 활성함수의 매개변수 $a_{(f),(i,j)}^{(l)}$와 $b_{(f),(i,j)}^{(l)}$는 Equation (15)와 (16)과 같이 손실함수 $L$에 대한 미분계수를 이용하여 역전파 과정에서 손실함수 $L$을 감소시키는 방향으로 갱신시킬 수 있다.

$$a_{(k),(i,j)}^{(l)} = a_{(k),(i,j)}^{(l)} - \rho_1\left(\frac{\partial L}{\partial a_{(k),(i,j)}^{(l)}}\right)$$
$$= a_{(l),(i,j)}^{(l)} - \rho_1\left(\frac{z_j^{(K+1)}\left(z_j^{(K+1)}-1\right)}{a_j^{(K)}} \times \frac{\partial L}{\partial z_j^{(K+1)}}\right) \quad (15)$$

$$b_{(k),(i,j)}^{(l)} = b_{(k),(i,j)}^{(l)} - \rho_2\left(\frac{\partial L}{\partial b_{(k),(i,j)}^{(l)}}\right)$$
$$= b_j^{(K)} - \rho_2\left(\frac{z_j^{(K+1)}\left(z_j^{(K+1)}-1\right)}{a_j^{(K)}} \times \frac{\partial L}{\partial z_j^{(K+1)}}\right) \quad (16)$$

완전연결층에서 사용되는 활성함수에 대해서도 민첩한 활성함수로 대체하여 사용할 수 있다. 민첩한 활성함수를 이용하여 각 은닉층의 선형 모델의 추정치 $\hat{y}_i^{(m)}$($m^{th}$ 은닉층의 $i^{th}$ 선형 모델의 추정치)를 활성화하고 동시에 손실함수를 감소시킬 수 있다. 민첩한 활성함수를 만드는 방법은 활성곱층에서와 같이 활성함수에 크기($a$)와 위치($b$)를 변화시킬 수 있는 매개변수를 도입하고 도입된 매개변수는 역전파 과정에서 학습시킴으로써 손실함수 $L$을 감소시킬 수 있다.

Fig. 8은 $m^{th}$ 은닉층의 $i^{th}$ 출력값의 추정치 $\hat{y}_i^{(m)}$를 Equation (7), (8)과 같은 민첩한 활성함수를 이용하여 변환하는 경우를 도식화한 것이다.

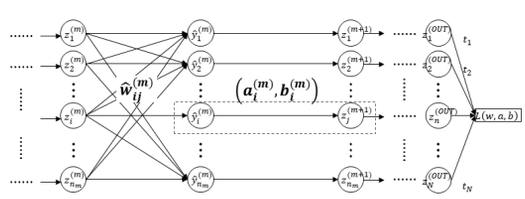



Fig. 8. Application of Agile Activation Function to the Estimate $\hat{y}_i^{(m)}$ of the $i^{th}$ Linear Model of the $m^{th}$ Hidden Layer

Fig. 8의 $m^{th}$ 은닉층의 $i^{th}$ 출력값의 추정치 $\hat{y}_i^{(m)}$를 Equation (7)의 민첩한 *Sigmoid* 활성함수를 이용하여 변환하는 경우 변환한 값 $z_i^{(m+1)}$은 다음 식과 같다.

$$z_i^{(m+1)} = \frac{a_i^{(m)}}{1 + e^{-\left(\hat{y}_i^{(m)} - b_i^{(m)}\right)}} \quad (17)$$

$z_i^{(m+1)}$에 대한 $\hat{y}_i^{(m)}$, $a_i^{(m)}$, $b_i^{(m)}$의 미분은 다음과 같다.

$$\frac{\partial z_i^{(m+1)}}{\partial \hat{y}_i^{(m)}} = \hat{y}_i^{(m)}\left(1 - \frac{\hat{y}_i^{(m)}}{a_i^{(m)}}\right) \quad (18)$$

$$\frac{\partial z_i^{(m+1)}}{\partial a_i^{(m)}} = \frac{1}{1 + e^{-\left(\hat{y}_i^{(m)} - b_i^{(m)}\right)}} \quad (19)$$

$$\frac{\partial z_i^{(m+1)}}{\partial b_i^{(m)}} = -\frac{z_i^{(m+1)}\left(z_i^{(m+1)}-1\right)}{a_i^{(m)}} \quad (20)$$

손실함수 $L$에 대한 $a_i^{(m)}$, $b_i^{(m)}$의 미분계수는 연쇄법칙을 이용하여 Equation (21), (22)와 같이 구할 수 있다.

$$\frac{\partial L}{\partial a_i^{(m)}} = \frac{1}{1 + e^{-\left(\hat{y}_i^{(m)} - b_i^{(m)}\right)}} \times \frac{\partial L}{\partial z_i^{(m+1)}} \quad (21)$$

$$\frac{\partial L}{\partial b_i^{(m)}} = \frac{z_i^{(m+1)}\left(z_i^{(m+1)}-1\right)}{a_i^{(m)}} \times \frac{\partial L}{\partial z_i^{(m+1)}} \quad (22)$$

Equation (23)과 (24)를 이용하여 민첩한 활성함수의 매개변수를 학습시켜 각 학습단계마다 손실함수를 감소시킬 수 있다.

$$a_i^{(m)} = a_i^{(m)} - \rho_3\left(\frac{\partial L}{\partial a_i^{(m)}}\right)$$
$$= a_i^{(m)} - \rho_3\left(\frac{1}{1 + e^{-\left(\hat{y}_i^{(m)} - b_i^{(m)}\right)}} \times \frac{\partial L}{\partial z_i^{(m+1)}}\right) \quad (23)$$

$$b_i^{(m)} = b_i^{(m)} - \rho\frac{\partial L}{\partial b_i^{(m)}}$$
$$= b_j^{(K)} - \rho\left(\frac{z_j^{(K+1)}\left(z_j^{(K+1)}-1\right)}{a_j^{(K)}} \times \frac{\partial L}{\partial z_j^{(K+1)}}\right) \quad (24)$$

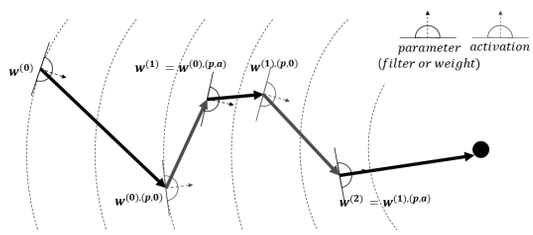



Fig. 9. Loss Value Changes According to Parameter Optimization and Agile Activation Function Application

Fig. 9는 합성곱 신경망에서 합성곱층의 필터 파라미터와 완전연결층의 가중치 파라미터 최적화과정과 민첩한 활성함수의 파라미터를 최적화시킨 경우 손실함수 값의 변화를 나타낸 것이다. Fig. 5와 비교할 때 활성함수를 적용한 단계에서 항상 손실함수를 감소시키는 방향으로 이동하게 된다.

Fig. 1과 같은, 입력층으로 1개의 이미지, $K$개의 합성곱층, $M$개의 은닉층을 갖는 완전연결층과 출력층을 가진 합성곱 신경망에 대해 훈련데이터훈련과정을 마친 후 시험데이터

에 대해 적용하는 방법은 다음과 같다.

(Step 1) 최적 필터 $F_{(k),(i,j)}^{(l)*}$를 구한다. 여기서 $l=1,...,K$는 각 합성곱층을 나타내고 $k=1,...,n_k$는 각 합성곱층에 적용한 필터를 의미한다. Equation (5)의 경사하강법을 통해 합성곱층의 최적필터를 구한다.

(Step 2) 최적특성맵 $M_{(f)}^{(l)*}=P^{(l-1)*}\otimes F_{(f)}^{(l)*}$을 구한다. $P^{(l-1)*}$는 $l^{th}$각 합성곱층의 입력이고 $F_{(k)}^{(l)*}$는 $l^{th}$합성곱층의 $k^{th}$ 최적필터를 합성곱하여 구한다. 이때 $P^{(0)*}$는 입력 이미지 $I$이다.

(Step 3) $A_{(f),(i,j)}^{(l)*}=\sigma_1(M_{(f),(i,j)}^{(l)*},a_{(f),(i,j)}^{(l)*},b_{(f),(i,j)}^{(l)*})$를 계산한다. 여기서 $\sigma_1$는 적용한 활성함수를 나타낸다(Fig. 7 참조).

(Step 4) 풀링을 통해 $P^{(l)*}$를 구한다.

(Step 5) $P^{(l)*}$를 벡터화하여 심층신경망의 입력값 $z^{(1)}=(z_1^{(1)},z_2^{(1)},...,z_{n_1}^{(1)})$으로 변환한다. 이때 $n_1$은 $P^{(l)*}$의 원소의 개수이다.

(Step 6) $\hat{y}_i^{(m)}=z^{(m)}\hat{w}_i^{(m)*}, m=1,...,M$를 계산한다. $\hat{w}_{(i,j)}^{(l)*}$는 Equation (6)을 이용하여 구한 최적 가중치이다. $m=M$이면 즉, $\hat{y}_i^{(M)}=\hat{y}_i^{(OUT)}$이 되고 최종 출력값이 된다.

(Step 7) $z_i^{(m+1)*}=\sigma_2(\hat{y}_i^{(m)},a_i^{(m+1)*},b_i^{(m+1)*})$는 Equation (23)과 (24)을 통해 구한 최적 활성함수의 가중치이다.

(Step 8) $\hat{y}_i^{(OUT)}, i=1,...n_{OUT}$을 이용하여 미리 정해진 $Loss(mean\,square\,error$ 또는 $cross\,entropy)$를 평가한다.

$ReLU$ 등 다른 활성함수에 대해서도 활성함수를 일반화시킬 수 있는 매개변수를 이용하고 매개변수들을 학습시켜 합성곱 신경망의 성능을 향상시킬 수 있다.

## 4. *MNIST*를 이용한 민첩한 활성함수의 성능 실험

28×28 *MNIST* 데이터 5,000개를 각 숫자별로 균등하게 표본추출하여 민첩한 활성함수의 성능을 평가하였다. 평가를 위해 사용한 합성곱 신경망은 Fig. 10과 같다. 입력데이터의 배치크기는 100, 필터 개수가 16, 8, 4인 3개의 합성곱층과 은닉층의 노드수가 30인 1개은 은닉층, 최종 출력수가 10이고 손실함수는 크로스엔트로피를 사용하였다.

활성함수의 성능 평가를 위한 실험을 Table 1과 같이 12개 조건에서 실시하였다. 실험한 12개 조건에서 완전연결층이 *Sigmoid*인 경우를 배제하였는데 그 이유는 완전연결층만 고려한 실험인 Fig. 11에서와 같이 민첩한 *Sigmoid*의 성능이 항상 우월했기 때문이다.

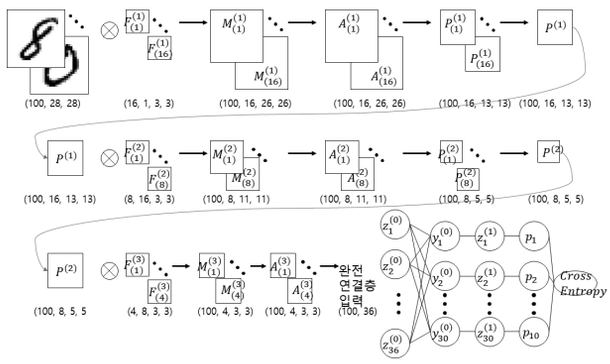



Fig. 10. CNN for Agile Activation Function Performance Evaluation

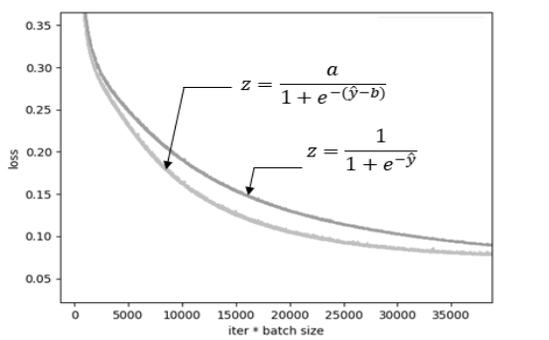



Fig. 11. Performance Comparison of *Sigmoid* and Agile *Sigmoid* in Fully Connected Layer *MNIST* 28×28

Table 1. Activation functions for Performance Evaluation

| n | Convolutional Layer | Fully Connected Layer | Loss Value | Iteration Counts |
|---|---|---|---|---|
| 1 | *Sigmoid* | *ReLU* | 0.18850 | 5030 |
| 2 | *Sigmoid* | Agile *ReLU* | 0.16094 | 5045 |
| 3 | *Sigmoid* | Agile *Sigmoid* | 0.14456 | 9908 |
| 4 | Agile *Sigmoid* | *ReLU* | 0.16259 | 5036 |
| 5 | Agile *Sigmoid* | Agile *ReLU* | 0.16684 | 4317 |
| 6 | Agile *Sigmoid* | Agile *Sigmoid* | 0.15600 | 7808 |
| 7 | *ReLU* | *ReLU* | 0.18127 | 1187 |
| 8 | *ReLU* | Agile *ReLU* | 0.17529 | 1187 |
| 9 | *ReLU* | Agile *Sigmoid* | 0.16031 | 3400 |
| 10 | Agile *ReLU* | *ReLU* | 0.17666 | 1050 |
| 11 | Agile *ReLU* | Agile *ReLU* | 0.18261 | 809 |
| 12 | Agile *ReLU* | Agile *Sigmoid* | 0.15027 | 3261 |

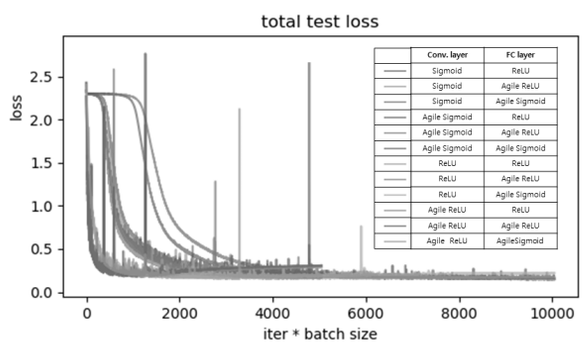



Fig. 12. Loss Function Values in Each Activation Function Combination

Fig. 12는 합성곱층과 완전연결층에서 사용한 활성함수의 12개 조합에서 시험손실(test loss)이다. Fig. 13은 12개 조합 중 상대적으로 성능이 좋은 5개의 조합을 선택하여 손실값을 비교한 것이다.

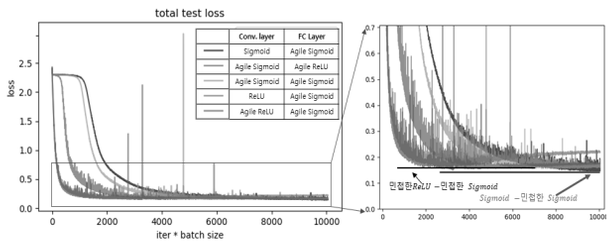



Fig. 13. Loss Function Values in the Top 5 Activation Functions

민첩한 *ReLU*-민첩한 *Sigmoid* 조합이 가장 우수한 성능을 보이고 있음을 알 수 있다. *Sigmoid*-민첩한 *Sigmoid* 조합은 계산속도는 느리지만 최저 손실함수 값을 가지고 있다. 가장 우수한 성능을 보인 활성함수 조합에서 각 매개변수의 값이 얼마나 변동하고 있는가를 나타낸 것이 Fig. 14이다. 가로 축은 활성함수의 크기 파라미터 $a$의 최대값과 최소값을 나타낸 것이고 세로 축은 활성함수 위치변화를 나타내는 매개변수$b$의 최대값과 최소값을 나타낸 것이다.

민첩한 활성함수는 순전파 과정과 역전파 과정에서 합성곱층과 완전연결층의 민첩한 활성함수의 매개변수를 손실함수를 최소화하는 방향으로 끊임없이 최적화함으로써 인공지능망의 성능을 향상시키고 있다.

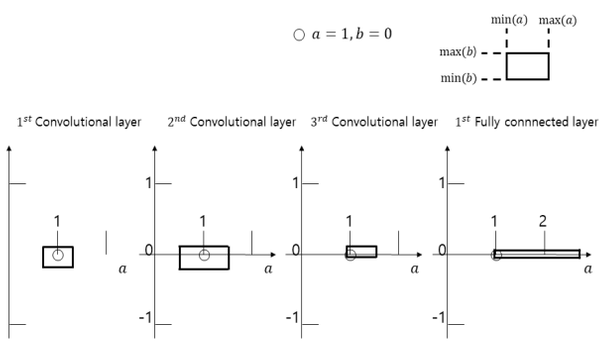



Fig. 14. Variable Range of Parameters in Agile *ReLU*-agile *Sigmoid*

## 5. 결론 및 향후 연구

합성곱 신경망은 합성곱층과 완전연결층에서 활성함수를 사용한다. 합성곱 신경망에서 사용하고 있는 기존의 활성함수를 민첩한 활성함수로 확장하여 심층신경망의 성능을 향상시킬 수 있었다.

민첩한 활성함수는 활성함수로의 입력 데이터를 다양한 형태로 변환할 수 있는 자유도를 부여하는 매개변수를 도입하여 확장한 함수이다. 도입된 매개변수에 대한 손실함수의 미분계수를 계산하고 역전파 계산 과정에서 매개변수를 손실함수 값을 감소시키는 방향으로 최적화함으로써 민첩한 활성함수의 성능을 보장하도록 하였다.

28×28 *MNIST* 데이터를 이용하여 제안된 민첩한 활성함수의 성능을 확인하였다. 실험에 사용한 합성곱 신경망은 합성곱층 3개, 완전연결층 1개, 각 합성곱층의 필터 수는 16, 8, 4를 사용하였고 완전연결층의 노드 수는 30개이다.

민첩한 활성함수는 신경망의 순전파 및 역전파 계산과정에서 각 계산 단계마다 손실함수 값을 지속적으로 감소시키는 방향으로 매개변수의 값을 최적화함으로써 손실함수와 어떤 관계도 갖지 않는 기존 활성함수들에 비해 우수한 성능을 보임을 확인 할 수 있다.

제안한 민첩한 활성함수는 단지 *ReLU*, *Sigmoid* 뿐 아니라 다양한 활성함수에 확장 적용할 수 있다.

합성곱 신경망은 심층심경망과 함께 다양한 문제를 해결하는 기본 신경망으로 적용 범위가 광범위하고 다른 신경망과 연계하여 사용할 수 있는 기본 신경망이다.

민첩한 활성함수를 일반화하는 방법으로 크기와 위치를 다양하게 변화시킬 수 있는 매개변수를 도입하여 활성함수를 일반화하였을 뿐 아니라 도입한 매개변수를 순전파 과정과 역전파 과정에서 최적화시킴으로써 손실함수를 빠르게 감소시킬 수 있다.

## References

[1] D. Hubel and T. Wiesel. “Receptive Fields of Single Neurons in the cat's Striate Cortex,” The Journal of Physiology, Vol.124, No.3, pp.574-591, 1959.

[2] A. Krizhevsky, I. Sutskever and G. Hinton, “Imagenet Classification with Deep Convolution Neural Networks,” NIPS Conference, pp.1097-1107, 2012.

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

[4] Chigozie Enyinna Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall, “Activation Functions: Comparison of Trends in Practice and Research for Deep Learning,” arXiv:1811.03378v1 [cs.LG] 8 Nov. 2018.

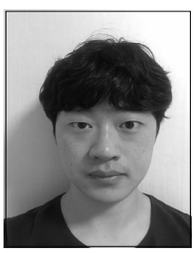
[5] N. Y. Kong and S. W. Ko, “Agile Activation Functions in Deep Neural Networks,” Working Paper, 2020.

[6] Nello Cristianini and John Shawe-Taylor, “An introduction to Support Vector Machines: and other kernel-based learning methods,” Cambridge University Press, New York, NY, USA, 2000.

[7] Bekir Karlik and A Vehbi Olgac, “Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks,” International Journal of Artificial Intelligence And Expert Systems (IJAE), Volume (1): Issue (4), 2011.

공 나 영

https://orcid.org/0000-0002-0245-7827
e-mail : lindsey0@hanmail.net
1988년 이화여자대학교 전자계산학과(학사)
1994년 이화여자대학교 전자계산학과(석사)
2017년 ~ 현 재 전주대학교 문화기술학과 박사과정
관심분야 : Data Science & Artificial Intelligence



고 영 민

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