https://doi.org/10.5392/JKCA.2021.21.03.616

파라메트릭 활성함수를 이용한 심층신경망의 성능향상 방법

Performance Improvement Method of Deep Neural Network Using Parametric Activation Functions

공나영*, 고선우**

전주대학교 문화기술학과*, 전주대학교 스마트미디어학과**

Nayoung Kong(lindsey0@hanmail.net)*, Sunwoo Ko(godfriends0@gmail.com)**

요약

심층신경망은 임의의 함수를 근사화하는 방법으로 선형모델로 근사화한 후에 비선형 활성함수를 이용하여 추가적 근사화를 반복하는 근사화 방법이다. 이 과정에서 근사화의 성능 평가 방법은 손실함수를 이용한다. 기존 심층학습방법에서는 선형근사화 과정에서 손실함수를 고려한 근사화를 실행하고 있지만 활성함수를 사용하는 비선형 근사화 단계에서는 손실함수의 감소와 관계가 없는 비선형변환을 사용하고 있다. 본 연구에서는 기존의 활성함수에 활성함수의 크기를 변화시킬 수 있는 크기 파라메터와 활성함수의 위치를 변화시킬 수 있는 위치 파라미터를 도입한 파라메트릭 활성함수를 제안한다. 파라메트릭 활성함수를 도입함으로써 활성함수를 이용한 비선형 근사화의 성능을 개선시킬 수 있다. 각 은닉층에서 크기와 위치 파라미터들은 역전파 과정에서 파라미터들에 대한 손실함수의 1차 미분계수를 이용한 학습과정을 통해 손실함수 값을 최소화시키는 파라미터를 결정함으로써 심층신경망의 성능을 향상시킬 수 있다. MNIST 분류 문제와 XOR 문제를 통하여 파라메트릭 활성함수가 기존의 활성함수에 비해 우월한 성능을 가짐을 확인하였다.

■ 중심어 : | 심층신경망 | 분류 | 파라메트릭 활성함수 | 역전파 | 학습 |

Abstract

Deep neural networks are an approximation method that approximates an arbitrary function to a linear model and then repeats additional approximation using a nonlinear active function. In this process, the method of evaluating the performance of approximation uses the loss function. Existing in-depth learning methods implement approximation that takes into account loss functions in the linear approximation process, but non-linear approximation phases that use active functions use non-linear transformation that is not related to reduction of loss functions of loss. This study proposes parametric activation functions that introduce scale parameters that can change the scale of activation functions and location parameters that can change the location of activation functions. By introducing parametric activation functions based on scale and location parameters, the performance of nonlinear approximation using activation functions can be improved. The scale and location parameters in each hidden layer can improve the performance of the deep neural network by determining parameters that minimize the loss function value through the learning process using the primary differential coefficient of the loss function for the parameters in the backpropagation. Through MNIST classification problems and XOR problems, parametric activation functions have been found to have superior performance over existing activation functions.

■ keyword : | Deep Neural Network | Classification | Parametric Activation Function | Backpropagation | Learning |

접수일자 : 2020년 10월 14일
수정일자 : 2020년 11월 09일
심사완료일 : 2020년 11월 09일
교신저자 : 공나영, e-mail : lindsey0@hanmail.net

I. 서 론

선형모델을 이용한 지도학습인 퍼셉트론(Perceptron)은 입력 데이터인 X에 대응하는 출력값 y의 추정치 $\hat{y}= X\hat{w}$을 참값 t로 매핑하는 방법이다[1]. 즉, 주어진 X에 대해 출력값 y의 관계를 $y = Xw+\epsilon$인 선형모델을 가정하여 출력값 y의 추정치 $\hat{y} = X\hat{w}$와 참값 t와의 차이를 최소화하는 $\hat{w}$을 구하는 방법이다.

선형모델로 추정한 $\hat{y}$가 참값인 t에 충분히 접근하지 못하는 경우, 모델의 성능 향상을 위해서 다음과 같은 3가지 방법을 이용하여 모델의 성능을 향상시킬 수 있다.

1) 선형모델을 비선형 모델로 확장하는 방법[2]

2) 커널 함수를 이용하여 차원을 높이는 방법[3]

3) 비선형 변환함수인 활성함수를 사용하는 방법[4] 등을 사용한다.

비선형 모델로 확장할 경우 y와 X의 관계를 나타내는 모델 $y=f(X, W)$에서 추정해야하는 파라미터 w의 수가 급격히 증가하는 문제와 파라미터 추정치의 분산이 증가할 위험이 있다.

커널 함수를 이용하여 데이터의 차원을 높이는 방법으로 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.

신경망을 이용한 심층신경망(Deep Neural Network, DNN)은 선형모델과 활성함수의 적용을 반복 적용하여 문제를 해결하는 방법을 사용한다.

K개의 은닉층을 갖는 심층신경망의 기본 구조는 [그림 1]과 같다. l^{th}은닉층$(l = 1,\dots, K)$에서 선형성을 가정한 어파인(Affine) 변환에서, $z_i^{(l)}(i = 1, \dots, n_{l-1})$에서 $y_j^{(l)}$ $(j = 1, \dots, n_l)$를 연결하는 가중치 파라미터 $w_{ij}^{(l)}$를 추정하고 추정된 $\hat{w}_{ij}^{(l)}$을 이용하여 $\hat{y}_j^{(l)}$를 계산한다. 계산된 $\hat{y}_j^{(l)}$는 비선형 변환인 활성함수(Activation function)를 이용하여 $z_j^{(l+1)}$을 계산한다. 즉, 심층신경망에서 입력층의 데이터는 어파인과 활성함수 적용을 반복해서 출력층까지 전달하는 구조이다.

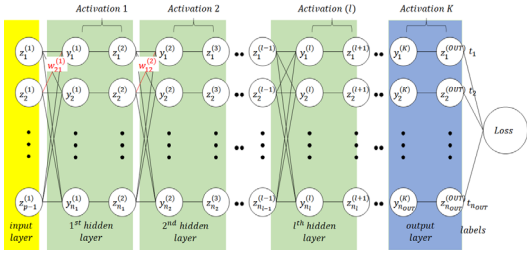


그림 1. 심층신경망의 기본 구조

본 논문은 선형 모델의 단순성을 이용하면서 심층신경망의 성능을 향상시키기 위한 활성함수의 의미를 살펴보고 활성함수의 성능을 향상시킬 수 있는 파라메트릭 활성함수를 제안한다.

II. 기존 연구에서 활성함수의 의미

1. 활성함수의 기능

심층신경망에서의 학습은 어파인 변환과 활성함수 변환을 거쳐 손실함수 값을 계산하는 순전파(Forward propagation) 과정과 연쇄법칙을 이용하여 각 파라미터에 대한 손실함수의 미분계수를 계산하여 파라미터의 최적값을 찾아가는 역전파(Back propagation) 과정을 반복하며 최종 출력값의 추정치 $\hat{y}$를 참값 t로 근사하게 된다.

이때 활성함수의 핵심 기능은 어파인 변환을 거쳐 계산되는 출력 중 중요한 특성치를 강조하고 중요하지 않은 특성치를 무시하거나 약화시키는 기능이다[5].

2. 활성함수의 종류

활성함수에 대한 연구는 크게 2개 방향으로 구분할 수 있다. 한 방향은 임의의 연속함수를 근사화하는 주제와 관련한 연구이고 다른 방향은 다양한 활성함수의 성능비교에 관한 연구이다.

1980년대부터 진행되고 있는 임의의 연속함수의 근사화 주제와 관련해 1987년 Wieland 와 Leighton[6]은 1개 또는 2개의 은닉층으로 구성된 네트워크의 근사화 역량에 대해 연구하였다. Chen과 그의 동료들의 연구[7][8]에서는 Sigmodal 함수의 유계성

(Boundedness)이 은닉층에서 활성함수로써 핵심적인 역할을 한다는 점을 밝혔다. Sigmodal 함수 이외에도 많은 다른 함수들이 은닉층에서 활성함수로써 사용될 수 있음이 밝혀졌다. 예를 들면, Hornik[9]는 유계 비상수 연속함수(Bounded nonconstant continuous function)들이 활성함수로 사용될 수 있음을 증명하였다. 또한 Mhaskar와 Micchelli[10]는 무한에 가까운 연속함수의 크기에 대해 몇몇의 제약을 가하면 임의의 비 다항식 함수를 활성함수로 사용할 수 있음을 증명하였다.

활성함수의 성능비교에 관한 연구의 목적은 어떤 활성함수를 선택할 것인가에 관한 연구이다. Sibi[11]는 선형 활성함수, Sigmoid 활성함수, Sigmoid Stepwise 활성함수, Sigmoid Symmetric 활성함수, Sigmoid Symmetric Stepwise 활성함수, Gaussian 활성함수, Gaussian Symmetric 활성함수, Elliot 활성함수, Elliot Symmetric 활성함수, Linear Piecewise 활성함수, Linear Piecewise Symmetric 활성함수에 대해 각 활성함수의 특징 뿐 아니라 심층신경망에 따라 적절한 활성함수 선택을 위한 연구를 진행하였다. 이 연구는 3개의 은닉층을 가지고 각 은닉층에서 각각 4, 5, 6개의 노드를 가지는 네트워크에 대한 성능 비교를 통해 이루어졌다. 적절한 활성함수를 선택하는 것이 중요한 문제이지만 연구 결과에 따르면 네트워크의 적절한 훈련을 위해서는 훈련 알고리즘, 네트워크 크기 및 학습률과 같은 다른 요소가 더 중요하다고 주장하고 있다. Chigozie Enyinna Nwankpa 및 그의 동료들[12]은 실제 심층신경망을 이용한 문제해결에 사용되고 있는 23 종류의 활성함수에 대한 사용 경향 과 적용 가능한 분야에 대한 연구를 진행하였으며 각 활성함수가 가지는 특성을 설명하고 있어, 실제 심층신경망을 활용하는 경우 어떤 활성함수를 사용하는 것이 적절한 것인가에 대한 지침이 될 수 있다. 이 연구에 의하면 *ReLU*와 *Softmax*가 압도적 비율로 사용되고 있음을 알 수 있다. Garrett Bingham and Risto Miikkulainen[13]은 활성함수를 데이터의 특성을 변경시킬 수 있는 진화탐색방법에 기반한 파라메트릭 방법을 제안하였다.

활성함수로 *Sigmoid*계열 활성함수와 *ReLU*계열의 활성함수가 널리 사용되어 왔으며 l^{th}층에서 *ReLU*, *Sigmoid*, *Tanh*의 함수식은 다음과 같다.

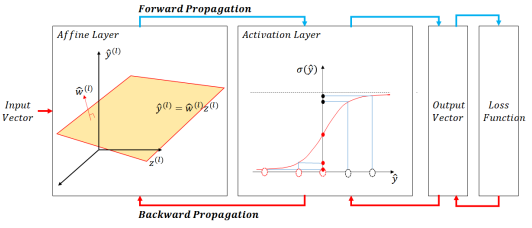
$$\textit{ReLU 활성함수: } z^{(l+1)} = \begin{cases} \hat{y}^{(l)}, & \hat{y}^{(l)} \geq 0 \\ 0, & \hat{y}^{(l)} < 0 \end{cases} \quad (1)$$

$$\textit{Sigmoid 활성함수: } z^{(l+1)} = \frac{1}{1+e^{-\hat{y}^{(l)}}} \quad (2)$$

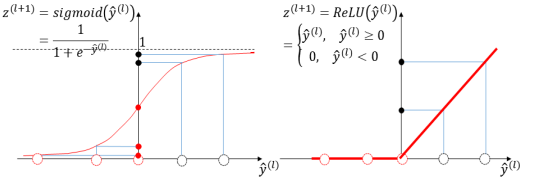
$$\textit{Tanh 활성함수: } z^{(l+1)} = \frac{(e^{\hat{y}^{(l)}} - e^{-\hat{y}^{(l)}})}{(e^{\hat{y}^{(l)}} - e^{-\hat{y}^{(l)}})} \quad (3)$$

여기서 $\hat{y}^{(l)}$는 어파인 변환의 출력값이다.

[그림 2(a)]는 l^{th} 은닉층의 입력값이 어파인 변환을 거치면서 $\hat{y}^{(l)} = X\hat{w}^{(l)}$가 계산되고 $z^{(l+1)}$는 *Sigmoid* 활성함수를 이용하여 변환한 예이다. [그림 2(b)]는 *Sigmoid* 활성함수와 *ReLU* 활성함수를 이용한 변환을 나타낸 것이다.



(a) 심층신경망의 동작 원리



(b) 활성함수 *Sigmoid*, *ReLU*의 데이터 변환

그림 2. (a)심층신경망의 동작 원리와 (b)활성함수 *Sigmoid*, *ReLU*의 데이터 변환

Sigmoid 활성함수는 연속함수이고 (0, 1) 사이의 출력값을 갖는다. 역전파 과정에서 계산되는 *Sigmoid* 활성함수의 1차 미분값의 최대값이 0.25이므로 은닉층의 깊이가 깊어질수록 기울기가 소멸(Gradient Vanishing)되는 문제가 발생한다. 이러한 기울기 소멸 문제를 해결하고 빠른 연산속도를 보장하는 *ReLU* 활성함수를 사용할 것을 권장하고 있다[14].

Ⅲ. 활성함수의 일반화와 파라메트릭 활성함수

1. 기존 활성함수의 문제점

어파인 변환인 선형모델로 손실함수 값을 충분히 최소화 시킬 수 없을 때, 어파인 변환의 출력 데이터를 입력데이터로 사용하는 비선형 함수인 활성함수를 사용한다.

기존 활성함수의 문제점 중의 하나는 어파인 변환의 출력값들을 비선형 변환할 때 손실함수를 최소화 시키는 방향과 관련성 없는 일방적인 비선형 변환이라는 점이다.

최적화 과정은 나타낸 [그림 3]은 l^{th} 은닉층의 어파인 변환에서 노드 i와 노드 j를 연결하는 파라미터 $w^{(l)_{ij}}$의 최적화 과정과 활성함수를 이용한 비선형 과정 두 개의 단계를 나타낸 것이다.

이때 어파인 변환의 파라미터 $w^{(l)_{ij}}$는 경사하강법을 통해 다음 식 (4)와 같이 개선된 파라미터를 찾게 된다.

$$w^{(l)_{ij}} = w^{(l)_{ij}} - \rho \nabla L(w^{(l)_{ij}}) \tag{4}$$

여기서 ρ는 학습률이고 $\nabla L(w^{(l)_{ij}})$는 $\nabla L(w_{ij}^{(l)}) = \partial L / \partial w_{ij}^{(l)}$로 계산되는 파라미터 $w^{(l)_{ij}}$에서 손실함수 L의 미분계수이다.

식 (4)의 최적화 과정은 학습률 ρ를 너무 크게 설정하여 오버슈팅 되지 않는 한 식 (4)를 이용한 $w^{(l)_{ij}}$의 최적화 과정은 지속적으로 손실함수 값을 감소시키게 된다. 여기서 $\nabla L(w_{ij}^{(l)}) = \partial L / \partial w_{ij}^{(l)}$는 연쇄법칙에 따라 계산할 수 있고 역전파 과정을 통해 모든 $w_{ij}^{(l)}$의 개선된 값을 구한다[15]. 역전파 과정을 통해 얻어진 $w_{ij}^{(l)}$와 입력값을 순전파 과정을 통해 최종 손실함수 값을 평가할 수 있다.

[그림 3]의 (a)는 l^{th} 은닉층에서 파라미터 $w^{(l)} = (w_{11}^{(l)}, w_{12}^{(l)}, \ldots, w_{n_{l-1}n_l}^{(l)})$을 $\hat{w}^{(l)}$로 최적화할 때의 손실값의 변화와 $\hat{y}^{(l)} (= X\hat{w}^{(l)})$에 활성함수 $\sigma(.)$를 적용하여 $z^{(l+1)} (= \sigma(\hat{y}^{(l)}))$로 변환할 때 손실함수 변화를 나타낸 것이다.

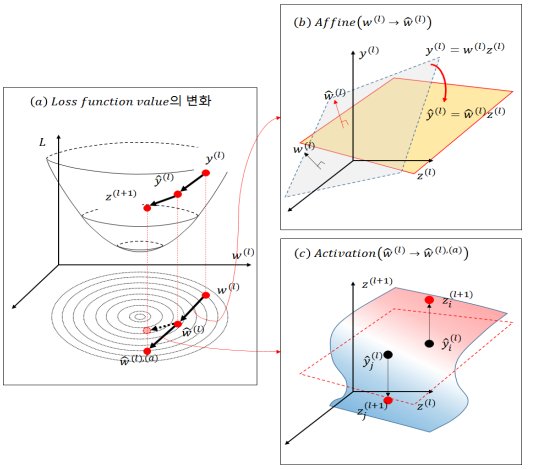


그림 3. 어파인변환과 활성함수에 의한 손실값의 변화

[그림 3]의 (b)에서, 식 (4)를 이용하여 어파인 파라미터 $w^{(l)}$를 $w^{(l)} \rightarrow \hat{w}^{(l)}$로 최적화할 때 손실함수는 지속적으로 감소한다. 반면, [그림 3]의 (c)는 어파인 변환된 $\hat{y}^{(l)}$을 활성함수를 적용하여 $z^{(l+1)}$로 변환했을 때 손실함수가 감소한다는 보장도 없다. 심층신경망에서 널리 사용되고 있는 활성함수인 식 (1), (2), (3)등은 손실함수 L과 어떤 함수관계도 가지지 않는 비선형변환일 뿐이다. [그림 3]의 (c)에서 활성함수의 적용했을 때, [그림 3]의 (a)의 손실함수 값은 증가할 수도 감소할 수도 있음을 나타낸 것이다.

2. 활성함수를 일반화한 파라메트릭 활성함수의 개발

활성함수 $\sigma_i^{(l)}(.)$에 파라미터를 도입하는 경우 다음과 같은 몇가지 기준이 필요하다. (1) $\hat{y}_i^{(l)}$를 다양한 값으로 변환할 수 있어야 하고 (2) 파라미터의 수는 최소여야 하고 (3) 파라미터들은 손실함수와 연계시킬 수 있어야 한다.

[그림 4]와 같이 활성함수의 크기를 결정하는 크기 파라미터(scale parameter) $a_j^{(l)}$와 활성함수의 위치를 결정하는 위치 파라미터(location parameter) $b_j^{(l)}$를 도입하여 $(a_j^{(l)}, b_j^{(l)})$를 동시에 고려한 변환으로 다양한 변환을 만들어 낼 수 있다.

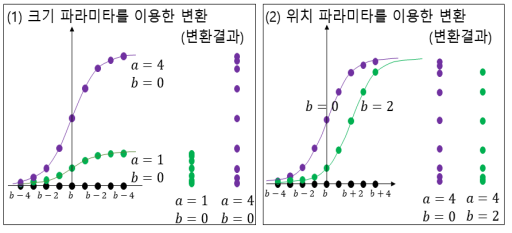
파라메트릭 활성함수를 도입한 경우 손실함수 값을 최소화시킬 수 있는 각 파라미터 최적화 방법은 다음과

같다.

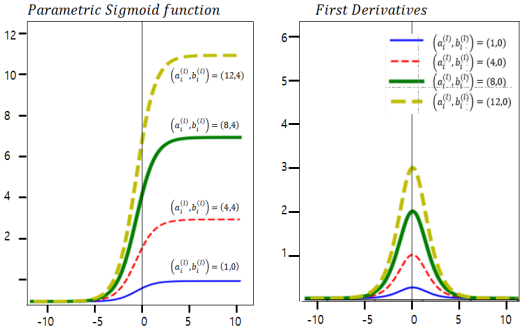
파라메트릭 활성함수의 도입 및 최적화 방법

1. 활성함수 $\sigma_i^{(l)}(.)$에 크기와 위치 파라미터 $a_i^{(l)}$, $b_i^{(l)}$를 동시에 도입: $z_i^{(l+1)} = \sigma_i^{(l)}(\hat{y}_i^{(l)} | a_i^{(l)}, b_i^{(l)})$
2. 크기 파라메터 $a_i^{(l)}$와 위치 파라메터 $b_i^{(l)}$가 손실함수 L에 미치는 영향도 $\partial L/\partial a_i^{(l)}$와 $\partial L/\partial b_i^{(l)}$를 계산
3. 경사하강법을 사용하여 $a_i^{(l)}$와 $b_i^{(l)}$의 최적 변환 $a_i^{(l)*}$와 $b_i^{(l)*}$를 결정
4. 최적 변환 $a_i^{(l)*}$와 $b_i^{(l)*}$를 이용하여 $z_i^{(l+1)} = \sigma_i^{(l)}(y_i^{(l)} | a_i^{(l)*}, b_i^{(l)*})$을 계산

[그림 4]의 파라메트릭 *Sigmoid* 활성함수는 $a^{(l)_i}$값에 따라 최대값, 최소값 및 1차 미분값이 달라진다. [그림 4(a)]는 $a^{(l)_i}$과 $b_i^{(l)}$값의 변화에 따라 어파인 변환의 출력값인 $\hat{y}_i^{(l)}$들이 어떻게 변환되는지를 보여준다. [그림 4(a)(1)]은 크기 파라미터 $a_i^{(l)}$를 1과 4로 각각 변환하였을 때 다양한 $z_i^{(l+1)}$ 변환 값을 만들어 낼 수 있음을 보인 예이고 [그림 4(a)(2)]는 위치 파라미터 $b_i^{(l)}$를 0과 2로 각각 변환하였을 때 $z_i^{(l+1)}$변환 값이 달라질 수 있음을 보인 예이다. [그림 4(b)]에서 $a^{(l)_i}$값이 커지게 되면 입력 $\hat{y}_i^{(l)}$들에 대해 파라메트릭 *Sigmoid* 활성함수를 적용한 결과인 $z_i^{(l+1)}$값들도 전체적으로 커지게 되고 $z_i^{(l+1)}$들 간의 간격도 변화하게 된다. 같은 $a^{(l)_i}$에서도 $b_i^{(l)}$값이 변화하게 되면 $z_i^{(l+1)}$들의 최대값 뿐 아니라 $z_i^{(l+1)}$들의 간격도 변화하게 된다.



(a) (1)$a^{(l)_i}$=[1,4]인 경우와 (2)$b_i^{(l)}$=[0, 2]인 경우 파라메트릭 Sigmoid 활성화 후 데이터 분포



(b) $a^{(l)_i}$=[1,4,8,12]이고 모든 $b_i^{(l)}$가 0인 경우의 파라메트릭 Sigmoid 함수와 1차 미분값

그림 4. 다양한 $a^{(l)_i}$, $b_i^{(l)}$값의 파라메트릭 Sigmoid 활성함수

파라메트릭 활성함수를 이용하여 어파인 변환의 결과인 $\hat{y}_i^{(l)}$를 활성함수를 적용한 결과는 다음 식 (5)로 구할 수 있다.

$$z_i^{(l+1)} = \sigma_i^{(l)}(y_i^{(l)} | a_i^{(l)}, b_i^{(l)}) \tag{5}$$

식 (5)의 매개변수 $a_i^{(l)}$와 $b_i^{(l)}$는 $w_{ij}^{(l)}$를 최적화하는 방법과 동일한 방법인 연쇄법칙과 역전파를 통해 다음 식 (6), (7)과 같이 구할 수 있다.

$$a_i^{(l)} = a_i^{(l)} - \rho_S \nabla L(a_i^{(l)}) \tag{6}$$

$$b_i^{(l)} = b_i^{(l)} - \rho_L \nabla L(b_i^{(l)}) \tag{7}$$

여기서 ρ_S과 ρ_L은 각각 크기 매개변수의 학습률과 위치 매개변수의 학습률이다. 각 매개변수에 대한 손실함수의 변화율$\nabla L(a_i^{(l)})$과 $\nabla L(b_i^{(l)})$은 연쇄법칙에 의해 다음과 같이 구할 수 있다.

$$\frac{\partial L}{\partial a_i^{(l)}} = \frac{\partial \sigma_i^{(l)}}{\partial a_i^{(l)}} \times \frac{\partial L}{\partial \sigma_i^{(l)}} \tag{8}$$

$$\frac{\partial L}{\partial b_i^{(l)}} = \frac{\partial \sigma_i^{(l)}}{\partial b_i^{(l)}} \times \frac{\partial L}{\partial \sigma_i^{(l)}} \tag{9}$$

일반적으로 널리 사용되고 있는 *Sigmoid* 활성함수와 *ReLU* 활성함수의 파라메트릭 활성함수는 다음 식 (10) 및 (11)과 같다.

$$z_i^{(l+1)} = \frac{a_i^{(l)}}{1 + e^{-(\hat{y}_i^{(l)} - b_i^{(l)})}} \tag{10}$$

$$z_i^{(l+1)} = \begin{cases} a_i^{(l)}(\hat{y}_i^{(l)} - b), & \hat{y}_i^{(l)} \geq b_i^{(l)} \\ 0, & \hat{y}_i^{(l)} < b_i^{(l)} \end{cases} \tag{11}$$

이때 식 (10)과 식 (11)에서 $a_i^{(l)}$와 $b^{(l)_i}$는 임의의 실수

값이고 $l=1,...,K,\ i=1,...,n_l,\ j=1,...,n_l$이다.

파라메트릭 *Sigmoid* 활성함수의 또 다른 장점 중의 하나는 [그림 4(b)]의 예와 같이 $|a^{(l)}_i|>1$인 경우 기존 *Sigmoid* 활성함수의 단점이었던 역전파 계산 과정중 기울기 소멸 문제가 완화될 수 있다는 것이다.

3. 파라메트릭 활성함수의 학습

파라메트릭 활성함수가 손실함수와 연계되기 위해서는, 파라메트릭 활성함수의 매개변수 $a^{(l)_i}$와 $b_i^{(l)}$에 대한 손실함수의 변화율 $\partial L/\partial a_i^{(l)}$와 $\partial L/\partial b^{(l)_i}$을 계산하여 역전파 과정에서 $a^{(l)_i}$와 $b_i^{(l)}$를 갱신할 새로운 $a^{(l)_i}$와 $b_i^{(l)}$값을 찾는 방법을 사용한다.

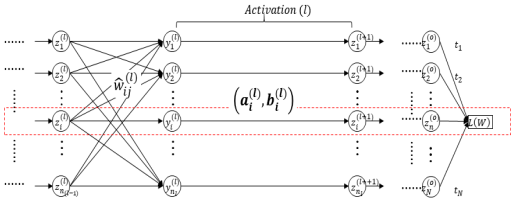


그림 5. l번째 은닉층 i번째 노드값의 파라메트릭 활성함수 적용

[그림 5]는 l^{th} 은닉층의 i^{th} 어파인 출력값 $\hat{y}_i^{(l)}$를 식 (5)의 파라메트릭 *Sigmoid* 활성함수를 이용하여 변환하는 경우로 변환한 값 $z_i^{(l+1)}$은 식 (10)과 같다.

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

$$\frac{\partial z_i^{(l+1)}}{\partial \hat{y}_i^{(l)}} = z_i^{(l+1)}\left(1-\frac{z_i^{(l+1)}}{a_i^{(l)}}\right) \tag{12}$$

$$\frac{\partial z_i^{(l+1)}}{\partial a_i^{(l)}} = \frac{1}{1+e^{-(\hat{y}_i^{(l)}-b_i^{(l)})}} \tag{13}$$

$$\frac{\partial z_i^{(l+1)}}{\partial b_i^{(l)}} = -\frac{(z_i^{(l+1)})^2}{a_i^{(l)}}\times e^{-(\hat{y}_i^{(l)}-b_i^{(l)})} = \frac{z_i^{(l+1)}(z_i^{(l+1)}-1)}{a_i^{(l)}} \tag{14}$$

손실함수 L에 대한 $a_i^{(l)}$, $b_i^{(l)}$의 미분계수는 연쇄법칙을 이용하여 식 (15)와 식 (16)과 같이 구할 수 있다.

$$\frac{\partial L}{\partial a_i^{(l)}} = \frac{\partial z_i^{(l+1)}}{\partial a_i^{(l)}}\times\frac{\partial L}{\partial z_i^{(l+1)}} = \frac{1}{1+e^{-(\hat{y}_i^{(l)}-b_i^{(l)})}}\times\frac{\partial L}{\partial z_i^{(l+1)}} \tag{15}$$

$$\frac{\partial L}{\partial b_i^{(l)}} = \frac{\partial z_i^{(l+1)}}{\partial b_i^{(l)}}\times\frac{\partial L}{\partial z_i^{(l+1)}} = \frac{z_i^{(l+1)}(z_i^{(l+1)}-1)}{a_i^{(l)}}\times\frac{\partial L}{\partial z_i^{(l+1)}} \tag{16}$$

식 (15), 식 (16)과 경사하강법을 이용하여 파라메트릭 *Sigmoid* 활성함수의 매개변수를 다음과 같이 학습시킬 수 있으며 각 학습단계마다 손실함수를 감소시키는 방향으로 $\hat{y}_i^{(l)}$ 를 변환할 수 있다.

$$a_i^{(l)} = a_i^{(l)} - \rho_S\left(\frac{\partial L}{\partial a_i^{(l)}}\right) = a_i^{(l)} - \rho_S\left(\frac{1}{1+e^{-(\hat{y}_i^{(l)}-b_i^{(l)})}}\times\frac{\partial L}{\partial z_i^{(l+1)}}\right) \tag{17}$$

$$b_i^{(l)} = b_i^{(l)} - \rho_L\frac{\partial L}{\partial b_i^{(l)}} = b_i^{(l)} - \rho_L\left(\frac{z_i^{(l+1)}(z_i^{(l+1)}-1)}{a_i^{(l)}}\times\frac{\partial L}{\partial z_i^{(l+1)}}\right) \tag{18}$$

심층신경망에서 파라메트릭 활성함수를 이용한 가중치 파라미터의 학습 알고리즘은 다음과 같다.

알고리즘: 파라메트릭 *Sigmoid* 경우

파라메트릭 활성함수 $\sigma(\hat{y}_i^{(l)}|a_i^{(l)},b_i^{(l)}) = \frac{a_i^{(l)}}{1+e^{-(\hat{y}_i^{(l)}-b_i^{(l)})}}$을 적용한 심층신경망의 학습 알고리즘. 손실함수 $L(y)$에 대한 순전파와 오류 역전파를 시행하고 파라메트릭 활성함수의 매개변수 $A^{(l)}=(a_1^{(l)},...,a_{n_l}^{(l)})$, $B^{(l)}=(b_1^{(l)},...,b_{n_l}^{(l)})$ $j=1,...,n_l$, $l=1,...,K$과 가중치 파라미터 $W^{(l)}=(w_{ij}^{(l)})$, $i=1,...,n_{(l-1)}$, $j=1,...,n_l$를 추정한다.

```
# 입력 : 입력벡터 X, label값 y, 학습률 ρ
# 출력 : 가중치 파라미터 W=(W^(1),...,W^(K)), 매개변수 A=(A^(1),...,A^(K)), B=(B^(1),...,B^(K))
# 순전파 계산:  W, A, B의 초기화
z^(0) = X
for k = 1 to K do
  for j = 1 to n do
    y_j^(k) = W_j^(k) z_j^(k-1)
    z_j^(k+1) = a_j^(k) / (1 + e^-(y_j^(k) - b_j^(k)))
ŷ^(OUT) = z^(OUT)
L = loss(ŷ^(OUT), y)
# 역전파 계산
for k = K-1 to 1 do
  for j = 1 to n do
```

$$\frac{\partial L(W,A,B)}{\partial w_j^{(k)}} = z_j^{(k)} \frac{\partial L(W,A,B)}{\partial z_j^{(k)}}$$

$$\frac{\partial L(W,A,B)}{\partial a_j^{(k)}} = \frac{1}{1+e^{-(\hat{y}_j^{(k)} - b_j^{(k)})}} \times \frac{\partial L(W,A,B)}{\partial z_j^{(k+1)}}$$

$$\frac{\partial L(W,A,B)}{\partial b_j^{(k)}} = \frac{z_j^{(k+1)}(z_j^{(k+1)}-1)}{a_j^{(k)}} \times \frac{\partial L(W,A,B)}{\partial z_j^{(k+1)}}$$

update W, A, B

for k = 1 *to* K **do**

  for j = 1 *to* n **do**

$$w_j^{(k)} = w_j^{(k)} - \rho\left(\frac{\partial L(W,A,B)}{\partial w_j^{(k)}}\right)$$

$$a_j^{(k)} = a_j^{(k)} - \rho\left(\frac{\partial L(W,A,B)}{\partial a_j^{(k)}}\right)$$

$$b_j^{(k)} = b_j^{(k)} - \rho\left(\frac{\partial L(W,A,B)}{\partial b_j^{(k)}}\right)$$

return W, A, B

Ⅳ. 파라메트릭 활성함수의 성능실험

기존 *Sigmoid* 활성함수, a, b 2개의 매개변수를 가지는 파라메트릭 *Sigmoid* 활성함수, 기존 *ReLU* 활성함수, a, b 2개의 매개변수를 가지는 파라메트릭 *ReLU* 활성함수에 대해 MNIST 데이터와 XOR 문제에 적용하여 각 활성함수의 성능을 비교하였다.

표 1. 성능평가를 위한 활성함수들

Activation Function		Function Formula
Sigmoid	기존	$z = \frac{1}{1+e^{-\hat{y}}}$
	Parametric	$z = \frac{a}{1+e^{-(\hat{y}-b)}}$
ReLU	기존	$z = \begin{cases} \hat{y}, & \hat{y} \geq 0 \\ 0, & \hat{y} < 0 \end{cases}$
	Parametric	$z = \begin{cases} a(\hat{y}-b), & \hat{y} \geq b \\ 0, & \hat{y} < b \end{cases}$

1. XOR 문제를 이용한 활성함수 성능 비교

파라메트릭 활성함수의 성능평가를 위해 1개의 은닉층과 은닉층은 2개의 노드만을 가지는 가장 간단한 [그림 6]의 네트워크를 사용하였다.

성능비교를 위하여 손실함수 값이 0.01에 도달할 때까지 필요한 실행 횟수를 비교하였다.

[표 2]와 [그림 8]을 보면 10회의 실험에서 *ReLU*는 발산하는 경우가 빈번히 발생하고 있지만 파라메트릭 *Sigmoid* 및 파라메트릭 *ReLU* 활성함수가 일반 *Sigmoid*, *ReLU* 활성함수에 비해 우수한 성능을 보이고 있음을 확인할 수 있다.

[그림 8]에서 좌표(1, 0)는 $a=1, b=0$인 *Sigmoid* 함수를 적용했을 때를 의미하며 파라메트릭 *Sigmoid*의 10회 실험에서 매개변수 a, b의 값을 나타낸 그림이다. [그림 6]의 $z_i^{(j)}(i=1,\ldots,10, j=1,2)$는 i^{th}실험에서 j^{th} 활성함수의 매개변수 $(a_i^{(j)}, b_i^{(j)})$ 값을 나타낸 것이다.

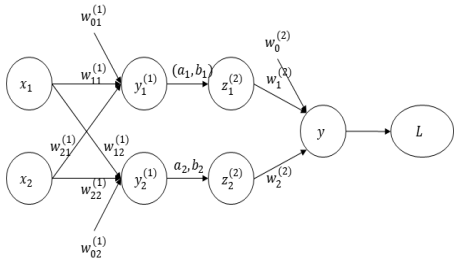


그림 6. 활성함수 평가를 위한 XOR 네트워크

표 2. XOR 네트워크에서의 활성함수 성능비교표

Exp.	Sigmoid		ReLU	
	기존	Parametric	기존	Parametric
수렴비율(%)	100	100	20	20
평균	4678.2	3670.4	871	810
표준편차	1903.0	1588.8	1033.8	980.1
최대	8592	6471	1602	1503
최소	2130	1368	140	117

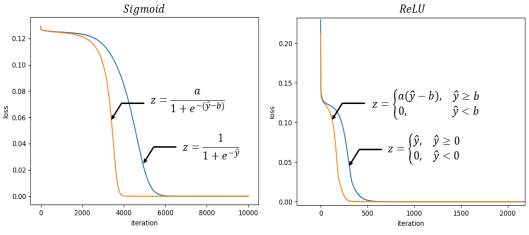


그림 7. 활성함수별 손실함수

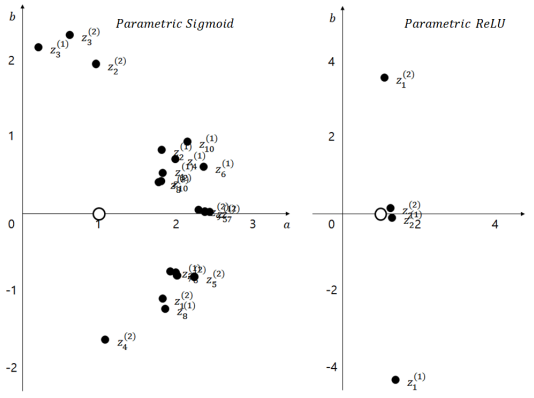


그림 8. 10회 실험에서의 a, b 매개변수 값

즉, 파라메트릭 *Sigmoid* 활성함수 또는 파라메트릭 *ReLU*를 이용하여 $\hat{y}_j^{(1)}$를 $z_j^{(2)}$로 비선형 변환할 때 $(a, b) = (1, 0)$로 고정된 값을 사용하지 않고 손실함수 L을 감소시키는 최적의 $(a^{(1)_j}, b^{(1)_j}), j = 1, 2$를 선택함으로써 손실함수를 감소시킴을 확인할 수 있다.

[그림 9]는 *ReLU* 활성함수가 많은 경우 수렴하지 않았지만 수렴한 경우, 각 활성함수를 사용했을 때 손실함수를 비교해본 것이다. 파라메트릭 *Sigmoid*와 파라메트릭 *ReLU*의 경우 순전파 계산과 역전파 계산을 1회 수행했을 때 손실함수 값이 매 계산 단계마다 매끄럽게 감소하는 반면 일반 *Sigmoid*와 일반 *ReLU*를 사용하는 경우 활성함수가 손실함수를 감소시키는 방향과 관계가 없이 손실함수 계산 도중 손실함수 값이 증가하는 현상이 빈번하게 발생함을 확인할 수 있다.

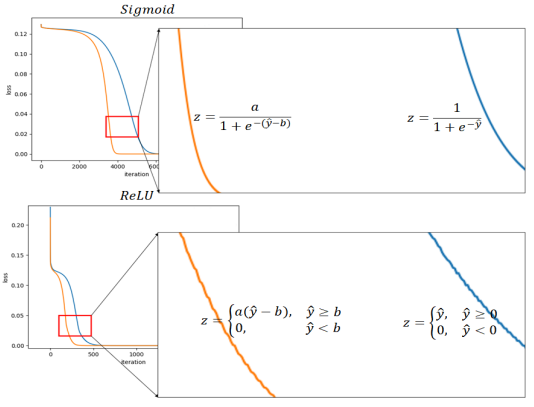


그림 9. 활성함수 종류에 따른 손실함수

2. 28x28 MNIST 데이터를 이용한 활성함수 성능 비교

각 활성함수 성능비교를 위해 28×28 MNIST 데이터에 대해 입력층 784개, 은닉층 1개, 은닉층의 노드수 200개, 출력층 10개의 [그림 10]의 간단한 네트워크를 이용하여 성능을 평가하였다.

MNIST 데이터 집합에서 훈련 데이터는 60000개, 테스트 데이터는 10000개로 구성하고 학습률(Learning rate)은 0.1, 배치 크기는 200, 손실함수는 크로스 엔트로피를 적용하였다.

[그림 11]은 *ReLU*, *Sigmoid* 각각에 대해 파라메트릭 *ReLU*와 파라메트릭 *Sigmoid*와 성능을 비교한 것이다. 파라메트릭 활성함수가 일반 활성함수에 비해 항상 성능이 좋음을 확인할 수 있다.

역전파 계산과정에서 파라메트릭 *Sigmoid*의 경우 $a_i^{(l)} \in [1.07,\ 4.49]$, $b_i^{(l)} \in [-0.45,\ 0.63]$의 범위의 값을 가진다. 한편 파라메트릭 *ReLU*의 경우 $a_i^{(l)} \in [1.00,\ 2.61]$, $b_i^{(l)} \in [-2.4,\ 0.23]$의 범위의 값을 가지며 손실함수를 최소화하는 과정을 거친다.

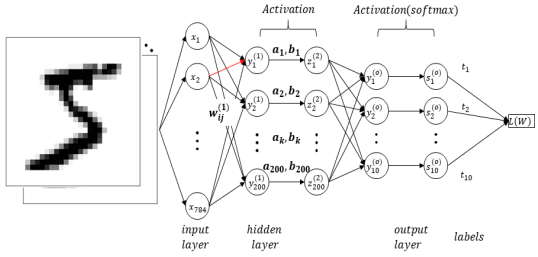


그림 10. MNIST 데이터 성능실험을 위한 네트워크

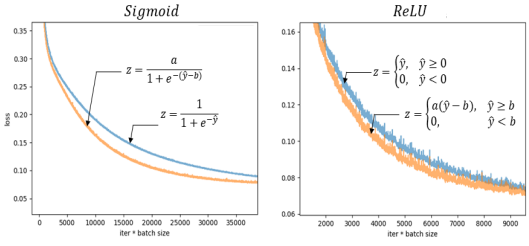


그림 11. 각 활성함수별 손실함수

V. 결론 및 향후 연구

심층신경망에서 사용하고 있는 기존의 활성함수를 파라메트릭 활성함수로 확장하여 심층신경망의 성능을 향상시킬 수 있었다. 파라메트릭 활성함수는 활성함수의 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 파라메트릭 활성함수의 성능을 보장하도록 하였다.

제안된 파라메트릭 활성함수의 성능을 확인하기 위하여 최소한의 은닉층과 각 은닉층의 노드 수를 최소화한 네트워크를 이용하여 XOR 문제와 MNIST 데이터에 적용하였다. 파라메트릭 활성함수는 각 계산 단계마다 손실함수 값을 지속적으로 감소시키는 방향으로 비선형 변환함으로써 기존의 *ReLU*, *Sigmoid*에 비해 우수한 성능을 보임을 확인 할 수 있다. 제안한 파라메트릭 활성함수는 단지 *ReLU*, *Sigmoid* 뿐 아니라 다양한 활성함수에 확장 적용할 수 있다.

심층신경망은 신경망 기반의 다양한 네트워크 모델의 기본 모델로 사용될 뿐 아니라 다른 신경망과 연계하여 사용할 수 있는 기본 신경망이다.

본 논문에서 제안한 파라메트릭 활성함수는 기본 신경망의 성능 개선에 기여할 수 있다는 점과 기존 신경망에서 사용되는 활성함수의 의미를 확대하였다는 점에서 의의가 있다.

논문에서 제안한 파라메트릭 활성함수는 크기와 위치를 변환할 수 있는 2 종류의 매개변수를 도입하는 개념으로 활성함수를 일반화한 것이다. 임의의 활성함수는 크기와 위치 뿐 아니라 다른 특성을 가질 수 있다는 관점에서 보면 파라메트릭 활성함수는 확대된 개념의 활성함수로 확장될 수 있다. 또한 현재 활성함수는 *ReLU*, *Sigmoid*, *Tanh* 등 특정 함수 중심으로 논의되고 있으나 보다 일반적인 활성함수를 정의하고 그 정의에 기반한 다양한 활성함수를 찾아내는 노력이 필요할 뿐 아니라 각 활성함수가 가지는 다양한 의미를 연구하는 것이 필요한 시점이다.

참 고 문 헌

[1] Stephen I. Gallant, "Perceptron-Based Learning Algorithms," IEEE Transactions on Neural Networks, Vol.1. No.2, 1990.

[2] Kevin P. Murphy, *Machine Learning: A Probabilistic Perspective,* The MIT Press, New York, NY, USA, 2012.

[3] 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.

[4] Charu C. Aggarwal, *Neural Networks and Deep Learning: A Textbook,* Springer International Publishing AG, 2018.

[5] Leonid Datta, "A Survey on Activation Functions and their relation with Xavier and He Normal Initialization," *arXiv:* 2004.06632v1 [cs.NE] 18 Mar 2020.

[6] A. Wieland and R. Leighton, "Geometric analysis of neural network capacity," *Proc. IEEE First ICNN,* 1, 1987.

[7] T. Chen, H. Chen, and R. Liu, "A constructive proof of Cybenko's approximation theorem and its extensions," Proc. 22nd Symp. Interface, 1990.

[8] T. Chen, H. Chen, and R. Liu, "Approximation capability in CR by multilayer feedforward networks and related problems," *IEEE Trans. Neural Networks*, Vol.6, No.1, Jan. 1995.

[9] K. Homik, M. Stinchcombe, and H. White, "Multi-layer feedforward networks are universal approximators," Neural Networks, Vol.2, 1989.

[10] H. N. Mhaskar and C. A. Micchelli, "Approximations for nonlinear functionals," IEEE Trans. Circuits Syst., Vol.39, No.1, pp.65-67, Jan. 1992.

[11] P. Sibi, S. A. Jones, and P. Siddarth, "Analysis of different Activation functions," Journal of Theoretical and Applied Information Technology, Vol.47, No.3, 2013.

[12] 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.

[13] Garrett Bingham and Risto Miikkulainen, "Discovering parametric activation functions," *arXiv: 2006.03179v2 [csLG] 6 Oct. 2020*

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Neural Information processing Systems, Conference,* 2012.

[15] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams, "Learning representations by back-propagating errors," Nature, Vol.323, 1986.

저 자 소 개

공 나 영(Nayoung Kong) 정회원

- 1988년 2월 : 이화여자대학교 전자계산학과(이학사)
- 1994년 2월 : 이화여자대학교 전자계산학과(이학석사)
- 2017년 9월 ~ 현재 : 전주대학교 문화기술학과(박사과정)

〈관심분야〉 : Data Science & Artificial Intelligence

고 선 우(Sunwoo Ko) 정회원

- 1985년 2월 : 고려대학교 산업공학과학과(공학사)
- 1998년 2월 : 한국과학기술원 산업공학과(공학석사)
- 1992년 2월 : 한국과학기술원 산업공학과(공학박사)
- 2005년 3월 ~ 현재 : 전주대학교 스마트미디어학과 교수

〈관심분야〉 : Data Science & Artificial Intelligence