KIPS Trans. Softw. and Data Eng.
Vol.10, No.10 pp.407~420
ISSN: 2287-5905 (Print), ISSN: 2734-0503 (Online)

https://doi.org/10.3745/KTSDE.2021.10.10.407

Alleviation of Vanishing Gradient Problem Using Parametric Activation Functions

Young Min Ko† · Sun Woo Ko††

ABSTRACT

Deep neural networks are widely used to solve various problems. However, the deep neural network with a deep hidden layer frequently has a vanishing gradient or exploding gradient problem, which is a major obstacle to learning the deep neural network. In this paper, we propose a parametric activation function to alleviate the vanishing gradient problem that can be caused by nonlinear activation function. The proposed parametric activation function can be obtained by applying a parameter that can convert the scale and location of the activation function according to the characteristics of the input data, and the loss function can be minimized without limiting the derivative of the activation function through the backpropagation process. Through the XOR problem with 10 hidden layers and the MNIST classification problem with 8 hidden layers, the performance of the original nonlinear and parametric activation functions was compared, and it was confirmed that the proposed parametric activation function has superior performance in alleviating the vanishing gradient.

Keywords : Deep Neural Network, Vanishing Gradient Problem, Parametric Activation Function, Backpropagation, Learning

파라메트릭 활성함수를 이용한 기울기 소실 문제의 완화

고 영 민† · 고 선 우††

요 약

심층신경망은 다양한 문제를 해결하는데 널리 사용되고 있다. 하지만 은닉층이 깊은 심층신경망을 학습하는 동안 빈번히 발생하는 기울기 소실 또는 폭주 문제는 심층신경망 학습의 큰 걸림돌이 되고 있다. 본 연구에서는 기울기 소실이 발생하는 원인 중 비선형활성함수에 의해 발생할 수 있는 기울기 소실 문제를 완화하기 위해 파라메트릭 활성함수를 제안한다. 제안된 파라메트릭 활성함수는 입력 데이터의 특성에 따라 활성함수의 크기 및 위치를 변환시킬 수 있는 파라미터를 적용하여 얻을 수 있으며 역전파과정을 통해 활성함수의 미분 크기에 제한이 없는 손실함수를 최소화되도록 학습시킬 수 있다. 은닉층 수가 10개인 XOR문제와 은닉층 수가 8개인 MNIST 분류문제를 통하여 기존 비선형활성함수와 파라메트릭활성함수의 성능을 비교하였고 제안한 파라메트릭 활성함수가 기울기 소실 완화에 우월한 성능을 가짐을 확인하였다.

키워드 : 심층신경망, 기울기 소실 문제, 파라메트릭 활성함수, 역전파, 학습

1. 서 론

Sigmoid와 같은 Squashing함수 또는 정류선형단위(Rectified linear unit)와 같은 비선형활성함수를 사용하는 은닉층 수가 하나 이상 있고 선형변환이 있는 순방향 신경망(Feedforward neural network)은 임의의 함수를 0이 아닌 임의의 정확도로 근사화할 수 있다[1]. 많은 경우 비선형활성함수를 가진 은닉층 수가 하나인 단층 순방향 신경망(Single layer forward neural network)보다 비선형활성함수를 가진 은닉층 수가 많은 심층신경망(Deep neural network)이 필요한 파라미터 수가 적고 일반화 오차가 감소한다[1]. 이러한 이유로 심층신경망은 다양한 문제를 해결하는데 널리 사용되고 있다.

심층신경망에서는 선형변환과 비선형활성함수를 사용하는 비선형변환을 반복적 수행한다. 심층신경망의 학습은 선형변환과 비선형변환의 반복을 통해 계산되는 출력값과 레이블의 함수인 손실함수를 최소화하는 선형변환의 파라미터를 구하는 과정이다. Fig. 1과 같은 2개 클래스를 구분하는 분류문제를 통해 심층신경망에서 사용되는 비선형활성함수의 의미를 생각해보자. 입력변수 (x_1, x_2)의 다양한 조건에서 얻어지는 결과치 $y_i, i=1,2,...,5$들이 있고 각 y_i들은 클래스 0 또는 클래스 1에 속한다고 하자. y_i가 어떤 클래스에 속하는지를 구분

† 준 회 원 : 전주대학교 인공지능학과 석사과정
†† 정 회 원 : 전주대학교 인공지능학과 교수
Manuscript Received : March 24, 2021
Accepted : May 11, 2021
* Corresponding Author : Sun Woo Ko(godfriend0@gmail.com)

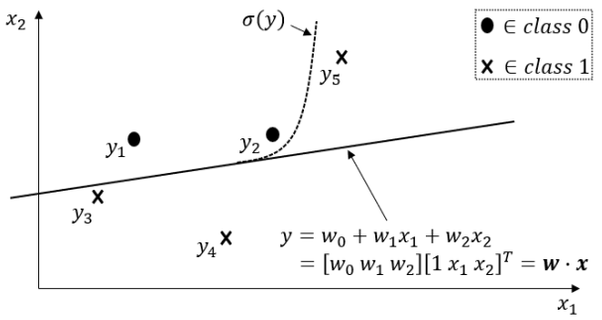


Fig. 1. Nonlinear Classification Problem with 2 Classes (w: parameter, x: input value)

하는 문제이다. 이를 위해 x_1, x_2의 선형변환으로 구해지는 초평면 y를 이용하여 y_i들의 클래스를 최대한 구분한 후에 비선형활성함수 σ를 이용하여 y_i들을 비선형 변환하여 분류성능을 향상시키는 과정이다.

하지만 비선형활성함수를 포함하고 있는 은닉층 수가 많은 심층신경망의 손실함수는 비볼록 손실함수(Non-convex loss function)로써 경사하강법을 사용하여 손실함수의 미분계수가 0인 점을 찾는 경우, 찾아진 점은 전역최솟값(Global minimum)을 보장하지 못하고 극솟값(Local minimum), 극댓값(Local maximum), 안장점(Saddle point) 3가지 경우 중의 한 경우를 찾게 된다. 은닉층 수가 많은 심층신경망의 손실함수 모양은 매우 복잡함이 알려져 있다[2].

은닉층이 깊은 심층신경망이 경사하강법을 사용하여 학습하는 동안 빈번히 나타나는 다른 주요문제는 기울기 소실 문제(Vanishing gradient problem)이다. 기울기 소실 문제는 경사하강법과 같은 기울기를 이용한 학습알고리즘으로 신경망을 학습할 때 작은 기울기 때문에 학습이 제대로 이루어지지 않는 문제로, Hochreiter[3]의 순환신경망(Recurrent neural network) 연구에서 처음 발견되었으며 Hochreiter 등[4]에 의해 연구되었다. 하지만 기울기 소실 문제는 순환신경망 뿐만 아니라 경사하강법을 사용하는 심층신경망[5], 합성곱 신경망(Convolutional neural network)[6] 등 은닉층이 깊은 모든 신경망에서 나타나는 중요한 문제이다.

본 논문은 기울기 소실이 발생하는 원인을 선형변환과 비선형변환에 대한 영향으로 구분하여 비선형활성함수에 의해 발생할 수 있는 기울기 소실을 파라메트릭 활성함수(Parametric activation function)[7]가 완화할 수 있음을 보인다. 제안된 파라메트릭 활성함수는 입력되는 데이터의 특성에 따라 활성함수의 형태를 변화시킬 수 있는 파라미터들을 포함하고 있는 활성함수이다. 파라메트릭 활성함수의 간단한 예로는 기존에 널리 사용되고 있는 임의의 활성함수에 크기와 위치를 변환시킬 수 있는 파라미터를 도입함으로써 얻을 수 있다. 파라메트릭 활성함수에 도입된 크기 및 위치와 관련된 파라미터들은 역전파과정을 통해 손실함수를 최소화되도록 학습시킬 수 있다.

기존에 기울기 소실을 완화하기 위해 연구된 활성함수들은 ReLU, Sigmoid, Tanh 등과 같은 비선형활성함수들의 변형 또는 결합으로 연구되었지만 제안된 파라메트릭 활성함수는 활성함수로 사용할 수 있는 임의의 비선형활성함수에 크기 및 위치와 관련된 파라미터를 간단히 적용하여 비선형활성함수에 의해 발생할 수 있는 기울기 소실을 완화할 수 있음에 의의가 있다.

2. 관련 연구

심층신경망을 학습하는 과정에서 발생하는 기울기 소실 문제를 해결하거나 완화하려는 기존 연구들이 있었으며, 다음과 같이 4가지로 구분할 수 있다. 첫째 파라미터 초기화 전략, 둘째 다양한 경사하강법 알고리즘을 이용한 학습방법, 셋째 순환신경망에서 사용되는 LSTM(Long short-term memory)[8,9]과 GRU(Gated recurrent unit)[10]를 이용한 방법, 마지막으로 다양한 비선형활성함수들을 이용한 연구가 있다.

Glorot와 Bengio는 선형변환층의 가중치 파라미터의 초기화 방법으로 기울기 소실문제를 완화하는 방법을 연구하였다[5]. 모든 은닉층의 비선형변환층의 분산과 모든 은닉층의 입력에 의한 선형변환층의 변화율 분산이 같아지도록 정규화한 초기화(Normalized initialization)방법을 제안하였다. He 등[11]은 활성함수로 ReLU 함수를 사용할 때 선형변환 과정의 파라미터 초기화 방법을 소개하여 30개가 넘는 은닉층을 사용한 모델에 대해서도 수렴함을 보였다. Hinton 등[12]은 심층신뢰신경망(Deep belief network)에서 각 층을 개별적으로 비지도 사전 학습한 뒤 학습된 값을 초기화 값으로 주는 방법을 제시하였다.

다양한 경사하강법 알고리즘을 이용하여 기울기 소실 문제를 해결하기 위한 방법으로 학습률을 학습과정에서 적응시키는 적응형 최적화 방법인 AdaGrad[13], AdaDelta[14], Adam[15]이 연구되었다. 과거의 기울기를 누적하여 반영하는 AdaGrad는 학습시작에서부터 기울기 제곱들을 누적하면 학습률이 필요이상으로 빠르게, 과도하게 감소하는 현상이 나타나는 문제점이 있다[1].

순환신경망에서 순차열(Sequence)이 긴 문장 안에 있는, 서로 멀리 떨어진 두 단어가 서로 중요한 영향을 미칠 때 BPTT (BackPropagation Through Time) 방법을 사용하여 학습하기 때문에 기울기 소실 혹은 폭발 문제가 발생하여 학습에 문제가 발생한다. 이 문제를 해결하기 위해 특정한 순차열이 가지고 있는 정보를 잃지 않고 전달될 수 있는 메모리 셀(Memory cell)을 사용하는 LSTM[8,9]과 GRU[10]가 제안되었다.

마지막으로 다양한 비선형활성함수를 사용하는 방법으로 ReLU[6], Hexpo[16], ISigmoid[17], ReLTanh[18] 등이 있다. ReLU는 양수인 값을 가진 구간에 대해서는 기울기 값이 1을 갖는 함수로 기울기 소실 문제를 피할 수 있지만 기울기 폭발 문제가 발생할 수 있다[19]. 이를 해결하기 위해 기

울기가 특정 임계값을 넘으면 크기를 조정하는 Gradient clipping[20]이 도입되었으며 ReLU의 다양한 함수 변형이 있다[21].

Kong과 Takatsuka[16]에 의해 연구된 Hexpo는 상한과 하한이 존재하는 기울기 크기를 파라미터로 조절하여 기울기 소실 문제를 완화하기 위한 방법으로 소개되었다. Tanh와 기본 형태는 유사하지만 활성함수의 형태를 결정하는 a,b,c,d 4개의 파라미터를 포함하고 있고 각 은닉층의 모든 노드들은 a,b,c,d를 공유하고 있다. Hexpo 활성함수는 4개의 공유 파라미터에 의해 활성함수의 모양이 결정되기 때문에 매우 다양한 비선형변환이 가능한 활성함수이다. MNIST 실험에서 Hexpo는 ReLU 보다 정확도와 학습속도에서 우월한 성능을 보였다.

Qin 등[17]이 연구한 ISigmoid와 Wang 등[18]이 연구한 ReLTanh는 각각 Sigmoid와 Tanh에 LReLU(Leaky ReLU)[21]를 결합한 비선형활성함수로 두 비선형활성함수 모두 중심부분은 Sigmoid와 Tanh을 사용하고 포화가 일어나는 양쪽 끝부분을 두 개의 직선으로 대체하여 구성된다. ISigmoid는 구간을 정하는 파라미터 a와 기울기를 결정하는 α가 있고 ReLTanh는 λ^-,λ^+ 파라미터를 가지고 있다. 특히 ReLTanh의 λ^-,λ^+는 양수 및 음수 임계값으로 직선의 시작위치와 기울기를 결정하며 손실함수를 최소화하는 방향으로 학습된다. Wang등[18]은 ReLTanh에서 중간에 Tanh 모습은 비선형 학습 기능을 제공하고 양쪽 끝부분의 선형 부분이 기울기 소실 문제를 완화할 수 있다고 소개하였다.

이후 최근에 기울기 소실 문제를 해결하기 위해 Basodi 등[22]는 학습 epoch 구간을 나눠 학습률을 다르게 적용하는 전략을 제안하여 모델의 학습시간을 줄일 수 있다고 소개하였다.

3. 파라메트릭 활성함수의 기울기 소실 문제 완화

3.1 기울기 소실의 3가지 원인

기울기 소실 문제를 보다 명확히 정의하기 위해 Fig. 2와 같은 일반적인 심층신경망을 고려하자.

Fig. 2에 제시된 심층신경망은 n_0개 입력 변수 $\boldsymbol{z}^{(1)}=\left[z_1^{(1)}\cdots z_{n_0}^{(1)}\right]^T$를 가지는 입력층이 있고, 전체 K개의 은닉층과 출력층으로 구성되어 있다. l번째 은닉층에서 $(n_{(l-1)}\times 1)$개의 노드 $\boldsymbol{z}^{(l)}=\left[z_1^{(l)}\cdots z_{n_{(l-1)}}^{(l)}\right]^T$를 선형변환하여 $(n_l\times 1)$개의 선형변환된 벡터 $\boldsymbol{y}^{(l)}=\left[y_1^{(l)}\cdots y_{n_l}^{(l)}\right]^T$를 계산할 때, $(n_l\times n_{(l-1)})$개의 가중치 파라미터행렬인 $W^{(l)}=\left[\boldsymbol{w}_1^{(l)}\cdots\boldsymbol{w}_{n_l}^{(l)}\right]^T$과 $(n_l\times 1)$ 절편 파라미터벡터 $\boldsymbol{w}_0^{(l)}=\left[w_{01}^{(l)}\cdots w_{0n_l}^{(l)}\right]^T$를 가진다. 이때 $W^{(l)}$을 구성하는 j번째 벡터 $\boldsymbol{w}_j^{(l)}=\left[w_{1j}^{(l)}\cdots w_{n_{(l-1)}j}^{(l)}\right]^T$이다. 선형변환된 l번째 은닉층의 벡터 $\boldsymbol{y}^{(l)}$는 $(n_l\times 1)$개의 비선형활성함수 벡터 $\boldsymbol{\sigma}^{(l)}=\left[\sigma_1^{(l)}\cdots\sigma_{n_l}^{(l)}\right]^T$를 통해 원소별로 비선형변환

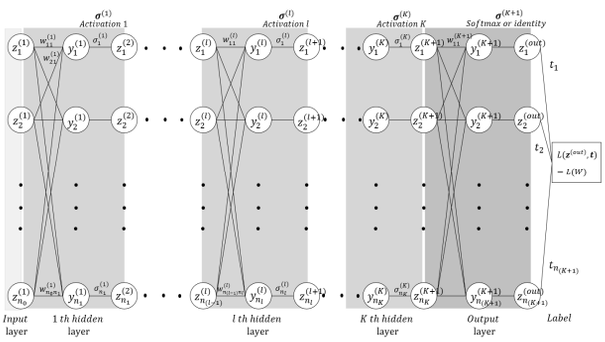


Fig. 2. Structure of Deep Neural Network (the l^{th} hidden layer has n_l nodes and consists of a total of K hidden layers.)

된다. 심층신경망의 모든 선형변환층의 가중치와 절편 파라미터를 W로 나타내고 특별히 언급이 없는 한 마지막 활성함수 $\boldsymbol{\sigma}^{(K+1)}$은 Softmax, 손실함수 L은 Cross-Entropy를 나타낸다.

Equation (1)은 선형변환과 비선형변환을 반복하여 나온 심층신경망의 출력값이고 Equation (2)는 Equation (1)의 출력값과 레이블 $\boldsymbol{t}$에 의해 정의 되는 손실함수이다.

$$\boldsymbol{z}^{(out)}=\boldsymbol{\sigma}^{(K+1)}(W^{(K+1)}\boldsymbol{\sigma}^{(K)}(\cdots\boldsymbol{\sigma}^{(1)}(W^{(1)}\boldsymbol{z}^{(1)}+\boldsymbol{w}_0^{(1)})\cdots)+\boldsymbol{w}_0^{(K+1)}) \quad (1)$$

$$L(W)=L(\boldsymbol{z}^{(out)},\boldsymbol{t}) \quad (2)$$

l번째 은닉층에서 노드 i와 노드 j를 연결하는 파라미터 $w_{ij}^{(l)}$는 Equation (3)과 같이 경사하강법을 통해 최적화된다.

$$w_{ij}^{(l)}=w_{ij}^{(l)}-\rho\nabla L(w_{ij}^{(l)}) \quad (3)$$

여기서 ρ는 학습률, $\nabla L(w_{ij}^{(l)})$는 $w_{ij}^{(l)}$에서 손실함수의 변화율이다.

Fig. 2에서 l번째 은닉층의 선형변환된 벡터 $\boldsymbol{y}^{(l)}$은 Equation (4)에 의해서 계산되고 $\boldsymbol{y}^{(l)}$을 비선형활성함수에 의해 비선형변환된 벡터 $\boldsymbol{z}^{(l+1)}$은 Equation (5)로 계산된다.

$$\boldsymbol{y}^{(l)}=W^{(l)}\boldsymbol{z}^{(l)}+\boldsymbol{w}_0^{(l)},\ l=1,\cdots K,K+1 \quad (4)$$

$$\boldsymbol{z}^{(l+1)}=\boldsymbol{\sigma}^{(l)}(\boldsymbol{y}^{(l)}),\ l=1,\cdots,K,K+1 \quad (5)$$

기울기 소실 문제를 구체적으로 들여다보기 위해 l번째 은닉층에서 파라미터 $w_{ij}^{(l)}$에 의한 손실함수의 변화율 $\nabla L(w_{ij}^{(l)})$을 생각해보자. $w_{ij}^{(l)}$에 의한 손실함수의 변화율 $\nabla L(w_{ij}^{(l)})$은 연쇄법칙에 의해 다음 Equation (6)으로 구할 수 있다.

$$\nabla L(w_{ij}^{(l)})=\frac{\partial L}{\partial\boldsymbol{z}^{(out)}}\frac{\partial\boldsymbol{z}^{(out)}}{\partial\boldsymbol{y}^{(K+1)}}\frac{\partial\boldsymbol{y}^{(K+1)}}{\partial\boldsymbol{z}^{(K+1)}}\frac{\partial\boldsymbol{z}^{(K+1)}}{\partial\boldsymbol{y}^{(K)}}\cdots\frac{\partial z_j^{(l+1)}}{\partial y_j^{(l)}}\frac{\partial y_j^{(l)}}{\partial w_{ij}^{(l)}} \quad (6)$$

Equation (6)에서 $\partial L/\partial\boldsymbol{z}^{(out)}$과 $\partial\boldsymbol{z}^{(out)}/\partial\boldsymbol{y}^{(K+1)}$ 그리고 $\partial y_j^{(l)}/\partial w_{ij}^{(l)}$을 제외하면 $\partial\boldsymbol{y}^{(m)}/\partial\boldsymbol{z}^{(m)}$과 $\partial\boldsymbol{z}^{(m)}/\partial\boldsymbol{y}^{(m-1)}, m=l+1,...,K+1$ 형태

의 Jacobian 행렬이 반복적으로 곱해진 형태이다.

Equation (6)에 따르면, 심층신경망에서 발생하는 기울기 소실문제는 다음과 같은 3가지 유형에 의해 발생할 수 있다.

유형 1) $\left|\left(\partial \boldsymbol{y}^{(m)}/\partial \boldsymbol{z}^{(m)}\right)_{ij}\right|<1, i=1,\cdots,n_m, j=1,\cdots,n_{(m-1)}$의 누적 곱에 의한 발생. 여기서 $\left(\partial \boldsymbol{y}^{(m)}/\partial \boldsymbol{z}^{(m)}\right)_{ij}$는 Jacobian 행렬 $\partial \boldsymbol{y}^{(m)}/\partial \boldsymbol{z}^{(m)}$의 i번째 행, j번째 열의 원소이다.

유형 2) $\left|\left(\partial \boldsymbol{z}^{(m)}/\partial \boldsymbol{y}^{(m-1)}\right)_{ij}\right|<1, i=1,\cdots n_{(m-1)}, j=1,\cdots,n_{(m-1)}$의 누적 곱에 의한 발생. 여기서 $\left(\partial \boldsymbol{z}^{(m)}/\partial \boldsymbol{y}^{(m-1)}\right)_{ij}$는 Jacobian 행렬 $\partial \boldsymbol{z}^{(m)}/\partial \boldsymbol{y}^{(m-1)}$의 i번째 행, j번째 열의 원소이다.

유형 3) 유형 1과 유형 2의 복합적 효과에 의한 발생.

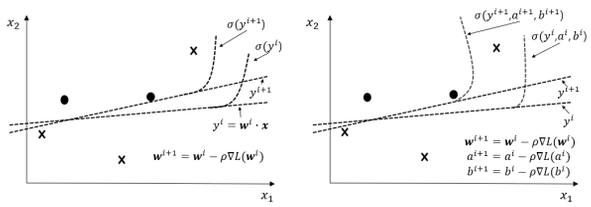
심층신경망 학습에서 기울기 소실은 은닉층의 입력$\boldsymbol{z}^{(m)}$에 의한 선형변환 $\boldsymbol{y}^{(m)}$의 변화율 값이 $\left|\left(\partial \boldsymbol{y}^{(m)}/\partial \boldsymbol{z}^{(m)}\right)_{ij}\right|<1$인 항의 누적 곱에 의한 발생과 $\boldsymbol{y}^{(m-1)}$에 의한 비선형변환 $\boldsymbol{z}^{(m)}$의 변화율 값이 $\left|\left(\partial \boldsymbol{z}^{(m)}/\partial \boldsymbol{y}^{(m-1)}\right)_{ij}\right|<1$인 항의 누적 곱에 의한 발생 그리고 이 두 가지 경우의 복합적 효과에 의한 발생으로 총 3가지로 분류할 수 있다. 즉, l번째 은닉층에서 파라미터 $w_{ij}^{(l)}$에 의한 손실함수의 변화율 $\nabla L(w_{ij}^{(l)})$를 연쇄법칙에 의해 계산하는 과정에서 $\left|\left(\partial \boldsymbol{y}^{(m)}/\partial \boldsymbol{z}^{(m)}\right)_{ij}\right|<1$인 항이 많이 존재하면 활성함수와 무관하게 기울기 소실이 발생할 수 있으며, 활성함수의 함수적 특성 때문에 $\left|\left(\partial \boldsymbol{z}^{(m)}/\partial \boldsymbol{y}^{(m-1)}\right)_{ij}\right|<1$이 많이 존재하면 기울기 소실이 발생할 수 있다.

은닉층이 깊은 심층신경망에서 모든 은닉층에 대한 활성함수를 Sigmoid, Tanh와 같은 비선형활성함수를 사용할 때 $\left|\left(\partial \boldsymbol{z}^{(m)}/\partial \boldsymbol{y}^{(m-1)}\right)_{ij}\right|\leq 1$인 비선형활성함수에 의한 기울기 소실이 발생할 수 있다. 특히 Sigmoid인 경우 $\left|\left(\partial \boldsymbol{z}^{(m)}/\partial \boldsymbol{y}^{(m-1)}\right)_{ij}\right|$의 최댓값이 0.25로 비선형활성함수에 의한 기울기 소실이 크게 발생한다.

3.2 파라메트릭 활성함수

신경망학습은 경사하강법을 이용하여 손실함수를 최소화하는 $\boldsymbol{w}$ 파라미터를 구하는 과정과 비선형활성함수 σ를 이용한 비선형변환을 하는 과정을 반복하는 과정으로 구성된다(Fig. 3(a)참조). Fig. 3(a)에서 $\boldsymbol{w}^i$는 i번째 경사하강법을 통해 학습된 선형변환 파라미터를 의미한다. 이 과정의 문제점은 손실함수를 이용한 비선형변환이 손실함수 값을 최소화한다는 보장이 없다는 것이다.

손실함수와 관계없는 활성함수 문제를 해결하기 위해 공나영 등[7]은 손실함수의 값에 따라 활성함수의 형태를 변화시킬 수 있는 파라메트릭 활성함수를 제안하였다. 제안된 파라메트릭 활성함수는 활성함수의 크기와 위치를 변환시킬 수 있는 2개의 a,b 파라미터를 도입한 활성함수이다. 여기서 도입된 2개의 파라미터 a,b는 역전파과정을 통해 손실함수가 최소화하는 방향으로 학습된다. 이러한 과정이 Fig. 3(b)에 제시되어 있다.



(a) (b)

Fig. 3. Linear Transformation Parameters $(\boldsymbol{w})^i$ Learned through the ith Gradient Descent Method. (a) Conventional Activation Function and (b) Parametric Activation Function

Table 1. Various Parametric Activation Functions

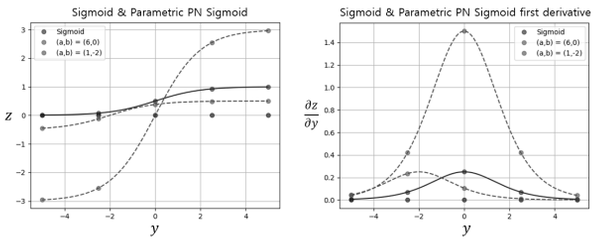
	Parametric activation function formula	Note
Parametric activation function	$z=\sigma_{(a,b)}(y)$	Common form
Parametric ReLU	$z=\begin{cases}a(y-b), & y\geq b\\ 0, & y<b\end{cases}$	Modified ReLU
Parametric Sigmoid	$z=\frac{a}{1+e^{-(y-b)}}$	Modified Sigmoid
Parametric PN Sigmoid	$z=\frac{a}{1+e^{-(y-b)}}-\frac{a}{2}$	Modified Sigmoid
Parametric Tanh	$z=a\left(\frac{e^{2(y-b)}-1}{e^{2(y-b)}+1}\right)$	Modified Tanh

Table 2. Gradients of Various Parametric Activation Functions

	$\frac{\partial z}{\partial y}$	$\frac{\partial z}{\partial a}$	$\frac{\partial z}{\partial b}$
Parametric ReLU	$\begin{cases}a, & y\geq b\\ 0, & y<b\end{cases}$	$\begin{cases}y-b, & y\geq b\\ 0, & y<b\end{cases}$	$\begin{cases}-a, & y\geq b\\ 0, & y<b\end{cases}$
Parametric Sigmoid	$z(1-\frac{z}{a})$	$\frac{1}{1+e^{-(y-b)}}$	$z(\frac{z}{a}-1)$
Parametric PN Sigmoid	$z(1-\frac{z}{a})$	$\frac{1}{1+e^{-(y-b)}}-\frac{1}{2}$	$z(\frac{z}{a}-1)$
Parametric Tanh	$\frac{4ae^{2(y+b)}}{(e^{2y}+e^{2b})^2}$	$\frac{e^{2(y-b)}-1}{e^{2(y-b)}+1}$	$-\frac{4ae^{2(y+b)}}{(e^{2y}+e^{2b})^2}$

파라메트릭 활성함수의 파라미터 a,b의 자세한 역할을 보기 위해 대표적인 비선형활성함수 ReLU와 Sigmoid 그리고 Tanh에 대한 파라메트릭 활성함수로의 변환과 크기 및 위치 파라미터 a,b 그리고 입력 y에 의한 파라메트릭 활성함수의 변화율을 나타내면 각각 Table 1, Table 2와 같다.

Table 1에서 파라미터 a는 활성함수의 크기를, b는 활성함수의 위치를 결정하는 임의의 실수이다. Parametric PN Sigmoid(PPNS)는 Parametric Sigmoid에 $-a/2$를 더함으로써 활성함수를 사용한 변환결과가 양수값과 음수값 모두 가질 수 있도록 만든 활성함수이다. 예를 들어 PPNS를 사용할 경우 Fig. 4(a)에서 다른 a와 b값을 가지는 PPNS들은 서로 다른 비선형변환을 의미한다. Fig. 4(b)는 a와 b값에 따라 y에 의한 z의 변화율이 변하는 경우를 나타낸 것으로 $a=6$일



(a) (b)

Fig. 4. (a) Sigmoid and PPNSs with (a, b)=(6, 0) and (a, b)=(1, −2) (b) Gradients of Sigmoid and PPNSs

때 최대 기울기가 Sigmoid의 0.25보다 6배 큰 1.5가 되고 $b=-2$일 때 y에 대응하는 $\partial z/\partial y$값이 달라짐을 볼 수 있다. 즉 입력 데이터의 특성에 따라 파라메트릭 활성함수의 파라미터 a와 b가 손실함수를 최소화하는 방향으로 학습되며 활성함수의 기울기에 제한이 없음을 알 수 있다.

3.3 파라메트릭 활성함수를 이용한 기울기 소실 문제 완화

은닉층의 수 K가 큰 Fig. 2와 같은 심층신경망 구조에서 $a=1, b=0$인 활성함수 $\sigma_{(1,0)}$을 사용할 경우 기울기 소실 문제가 발생할 수 있다. $\sigma_{(1,0)}$ 대신에 a와 b가 임의의 실수값을 가질 수 있는 $\sigma_{(a,b)}$로 일반화한 파라메트릭 활성함수는 파라미터 a값에 따라 $\sigma_{(1,0)}$의 크기를 조정할 수 있고 파라미터 b값에 따라 $\sigma_{(1,0)}$의 위치를 조정할 수 있는 활성함수이다.

파라메트릭 활성함수 $\sigma_{(a,b)}$의 파라미터 a, b는 Equation (2)에서 정의된 손실함수 $L(W)$를 최소화하는 a와 b로 결정한다. 파라메트릭 활성함수로 PPNS를 사용할 경우, l번째 은닉층의 $w_{ij}^{(l)}$에 의한 손실함수 $L(W)$의 변화율, l번째 은닉층의 j번째 비선형변환 결과인 노드 $z_j^{(l+1)}$의 파라미터 $a_j^{(l)}$와 $b_j^{(l)}$에 의한 손실함수(L)의 변화율은 다음과 같다.

$$\nabla L(w_{ij}^{(l)}) = \frac{\partial L}{\partial z_j^{(l+1)}}\frac{\partial z_j^{(l+1)}}{\partial y_j^{(l)}}\frac{\partial y_j^{(l)}}{\partial w_{ij}^{(l)}} = \frac{\partial L}{\partial z_j^{(l+1)}}\times z_j^{(l+1)}\left(1-\frac{z_j^{(l+1)}}{a_j^{(l)}}\right)\times\frac{\partial y_j^{(l)}}{\partial w_{ij}^{(l)}} \quad (7)$$

$$\nabla L(a_j^{(l)}) = \frac{\partial L}{\partial z_j^{(l+1)}}\frac{\partial z_j^{(l+1)}}{\partial a_j^{(l)}} = \frac{\partial L}{\partial z_j^{(l+1)}}\times\left(\frac{1}{1+e^{-(y_j^{(l)}-b_j^{(l)})}}-\frac{1}{2}\right) \quad (8)$$

$$\nabla L(b_j^{(l)}) = \frac{\partial L}{\partial z_j^{(l+1)}}\frac{\partial z_j^{(l+1)}}{\partial b_j^{(l)}} = \frac{\partial L}{\partial z_j^{(l+1)}}\times z_j^{(l+1)}\left(\frac{z_j^{(l+1)}}{a_j^{(l)}}-1\right) \quad (9)$$

Equation (7), (8), (9)를 이용하여 Equation (10), (11), (12)에 제시된 경사하강법을 이용하여, 모든 l, i, j에 대해 손실함수를 최소화하는 파라미터 $w_{ij}^{(l)}$, $a_j^{(l)}$, $b_j^{(l)}$들을 구할 수 있다.

$$w_{ij}^{(l)} = w_{ij}^{(l)} - \rho_1\nabla L(w_{ij}^{(l)}) \quad (10)$$

$$a_j^{(l)} = a_j^{(l)} - \rho_2\nabla L(a_j^{(l)}) \quad (11)$$

$$b_j^{(l)} = b_j^{(l)} - \rho_3\nabla L(b_j^{(l)}) \quad (12)$$

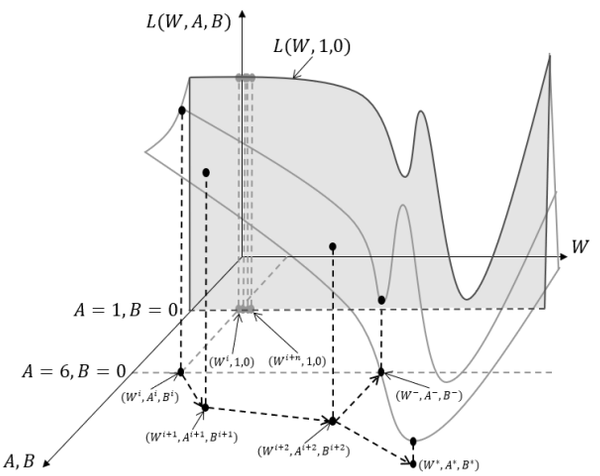


Fig. 5. Loss Function Space for Deep Neural Network Parameters (W, A, B). (W : All linear transformation parameters of the deep neural network, A, B : All hidden layer parametric activation function parameters)

여기서 ρ_1, ρ_2, ρ_3는 각 파라미터에 대응하는 학습률이다. 이와 같은 방법으로 임의의 은닉층 l에서 j번째 활성화된 값 $z_j^{(l+1)}$을 구하는데 사용한 파라메트릭 활성함수 $\sigma_{(a_j^{(l)}, b_j^{(l)})}^{(l)}$의 파라미터 $a_j^{(l)}$, $b_j^{(l)}$들에 대해 손실함수를 최소화 하는 $a_j^{(l)}$, $b_j^{(l)}$을 찾을 수 있다. 이러한 학습과정이 Fig. 5에 제시되어 있다.

Fig. 5에서 W행렬은 모든 은닉층의 선형변환 파라미터들 또는 W의 탐색공간을 의미한다. $A=(a_j^{(l)})$는 파라메트릭 활성함수의 크기 파라미터들로 구성된 행렬 또는 A의 탐색공간을 의미한다. $B=(b_j^{(l)})$는 파라메트릭 활성함수의 위치 파라미터들로 구성된 행렬 또는 B의 탐색공간을 의미한다. (W^i, A^i, B^i)는 i번째 경사하강법을 통해 학습된 심층신경망의 모든 파라미터들을 의미하며 (W^-, A^-, B^-)은 W, A, B 각각에 대해 $\partial L/\partial W=0$, $\partial L/\partial A=0$, $\partial L/\partial B=0$을 만족하는 임계점(critical point)이다. (W^*, A^*, B^*)은 손실함수 값이 국소적으로 최소가 되는 극솟값을 나타낸다.

은닉층이 깊은 심층신경망에서 활성함수로 $\sigma_{(1,0)}$을 사용하는 경우는 Fig. 5에서 W를 최적화하는 과정에서 W의 탐색공간이 $A=1, B=0$으로 제한되는 제한된 탐색공간만을 탐색하는 것을 의미하고 이때의 손실함수는 $L(W, 1, 0)$로 국한되게 된다.

$\sigma_{(1,0)}$을 활성함수로 사용하는 경우, $\partial L/\partial W=0$의 조건이 만족되면 임계점에 도달했음을 의미하고 탐색을 멈추게 된다. 이때, $\partial L/\partial W=0$는 실제 임계점일 수도 있고 Equation (6)에서 도출한 기울기 소실 문제가 발생하는 3가지 유형 중의 하나가 발현된 경우일 수 있다. 하지만 파라메트릭 활성함수 $\sigma_{(a,b)}$을 사용하는 경우 탐색공간이 $W\times AB$로 확대되게 되고 $\partial L/\partial W=0$을 만족하더라도, $\partial L/\partial A=0$와 $\partial L/\partial B=0$을 동시에 만족하는 새로운 임계점을 찾기 위한 추가 탐색이 계속된다. 추가 탐색의 의미는 다음과 같다.

1. $\sigma_{(1,0)}$을 활성함수로 사용할 때, $\partial L/\partial W=0$의 조건이 충족되는 경우 중 Equation (6)을 통해 도출된 기울기 소실이 발생한 2번째 경우라면 $\left|\left(\partial \boldsymbol{z}^{(m)}/\partial \boldsymbol{y}^{(m-1)}\right)_{ij}\right|<1$의 누적 곱에 의해 발생한 기울기 소실을 극복한 경우를 의미하게 된다.
2. 이 경우가 Fig. 5에서 $A=1, B=0$인 탐색 영역을 벗어나 활성함수의 기울기에 자유로운 $W\times AB$ 공간을 탐색한 경우를 의미하고 입력데이터에 따라 손실함수를 최소화하는 방향으로 최적화하여 더 작은 손실함수 값을 가지는 임계점을 탐색하는 것을 의미한다.

예를 들어, PPNS를 활성함수로 사용한 심층신경망에서 A, B의 초기값을 $A=6, B=0$으로 준 경우, Fig. 4(b)에서 $a=6$인 경우처럼 Sigmoid의 기울기 최댓값 0.25보다 6배 커진 최대 기울기를 가짐으로써 학습 최초에 비선형활성함수의 기울기가 작아 기울기 소실이 발생할 수 있는 문제를 완화할 수 있다. 또한 Fig. 5에서와 같이 파라미터 탐색공간이 $W\times AB$로 확대되고 학습과정 (W^i, A^i, B^i)에서 W 뿐 아니라 A, B를 사용하여 손실함수 값을 최소화하는 방향으로 지속적으로 학습할 수 있음을 나타낸다.

하지만 은닉층이 깊어짐에 따라 손실함수의 모양은 Li 등 [2]이 연구한 바와 같이 매우 복잡해지고 Equation (10), (11), (12)의 경사하강법을 사용해 학습하므로 (W^*, A^*, B^*)와 같은 손실함수 값이 국소적으로 최소가 되는 극솟값은 보장하지 못한다.

4. 파라메트릭 활성함수의 성능실험

4.1 은닉층 수가 10개인 $XOR_{(10)}$문제를 이용한 파라메트릭 활성함수의 기울기 소실 완화 성능실험

1) 은닉층 수가 10개인 $XOR_{(10)}$문제를 이용한 파라메트릭 활성함수 PPNS의 기울기 소실 문제 완화 실험

Fig. 6은 $XOR_{(10)}$문제를 해결하기 위한 은닉층의 수가 10개인 인공신경망이다. 실제 $XOR_{(10)}$문제를 해결하기 위해서는 10개의 은닉층이 필요하지 않지만 기울기 소실을 유발할 수 있는 예제를 위해 고안한 것이다. 기울기 소실 문제가 발생할 수 있는 대표적인 비선형활성함수 Sigmoid와 그에 대해 파라메트릭 활성함수 파라미터 a,b을 적용한 PPNS를 다음 Table 3의 실험조건으로 비교하였다. $a=6, b=0$으로 초깃값을 준 이유는 최초 학습 때 활성함수의 기울기의 절댓값이

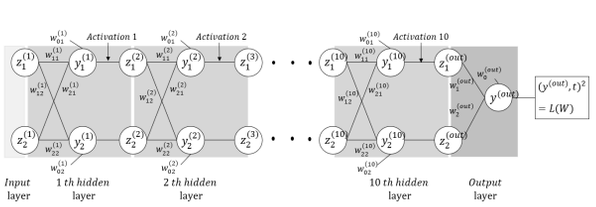


Fig. 6. $XOR_{(10)}$ Problem with 10 Hidden Layers and 2 Nodes

Table 3. Experiment Conditions

Algorithm	Gradient descent
Learning rate	0.05
Iterations	10,000
Loss function	Mean Squared Error
Initialization of W parameters	Standard normal distribution
Initialization of PPNS parameters	$a=6, b=0$

Table 4. Results of experiments conducted 100 times

	Sigmoid	PPNS
number of convergence with Loss=0	0	76

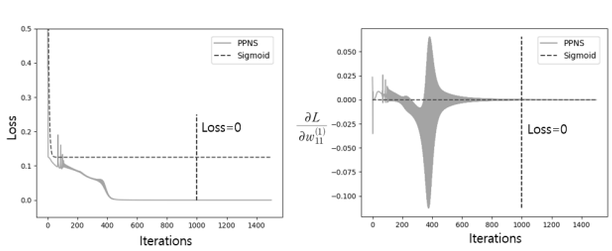
1보다 작아 발생할 수 있는 기울기 소실을 방지하고 추후 학습에 따라 a,b는 경사하강법을 통해 학습되기 때문에 활성함수의 최대 기울기 크기를 조절하는 a를 6으로 설정함으로써 최대 기울기 값이 1.5를 갖도록 하였다.

Fig. 6 $XOR_{(10)}$문제를 Table 3 실험조건으로 Sigmoid와 PPNS에 대해 100회씩 실험하여 학습이 끝난 후 손실함수 값이 0에 수렴하는 경우를 세어본 결과는 Table 4와 같다. 손실함수 값이 0으로 수렴하는 횟수를 평가기준으로 삼은 이유는 파라메트릭 활성함수가 기울기 소실을 완화하여 10,000번의 iteration 안에 손실함수를 감소시키고 파라메트릭 활성함수의 파라미터를 추가함에 따라 복잡해질 수 있는 손실함수에 대해 손실함수 값이 0에 수렴하는 횟수를 세어봄으로써 활성함수의 성능을 확인할 수 있기 때문이다.

Table 4는 Sigmoid의 경우 손실함수 값이 0에 수렴한 횟수가 100회 중 0회임을 보여주는 반면 PPNS의 경우 76회 수렴함을 보여준다.

Sigmoid에 의해 발생하는 기울기 소실을 확인하고 비선형활성함수에 의해 발생하는 기울기 소실을 PPNS가 완화할 수 있는지를 자세히 보기 위해 PPNS의 손실함수 값이 0으로 수렴한 76회 중 한 가지 경우에 대해서 동일한 실험에 사용된 Sigmoid를 함께 비교한 그림은 Fig. 7과 같다.

Fig. 7(a)은 Sigmoid와 PPNS의 손실함수 값을 iteration에 따라 그린 것으로 10,000번의 iteration 중 1,500번



(a) (b)

Fig. 7. Sigmoid and PPNS Compared. (a) Loss Function Value According to Iterations (b) The Gradient of the Loss Function by the Weight Parameter $w_{11}^{(1)}$ According to Iterations.

iteration 이후는 Sigmoid와 PPNS 두 함수 모두 손실함수 값이 변하지 않아 1,500번 iteration까지 그린 것이다. PPNS는 약 1,000번째 iteration에서 손실함수 값이 0에 수렴한 반면 Sigmoid는 학습이 끝난 후에도 손실함수 값이 0에 수렴하지 못한 것을 볼 수 있다.

Sigmoid의 경우 기울기 소실 때문에 손실함수 값이 수렴하지 못한 것을 보기위해 Fig. 7(b)은 Equation (13)과 같은 1번째 은닉층의 가중치 파라미터 $w_{11}^{(1)}$에 의한 손실함수 변화율을 iteration에 따라 그린 것이다. Sigmoid와 PPNS 모두 $w_{11}^{(1)}$을 제외한 나머지 1번째 은닉층의 가중치, 절편 파라미터들도 Fig. 7(b)과 비슷하며 Fig. 7(b)에서 Sigmoid의 경우에 처음 iteration부터 기울기 소실이 발생하여 0에 가까운 $\partial L/\partial w_{11}^{(1)}$값을 가지는 것을 볼 수 있다. 반면 PPNS의 경우 약 1,000번째 iteration까지 $\partial L/\partial w_{11}^{(1)}$값이 Sigmoid보다 상대적으로 큰 것을 볼 수 있다.

$$\frac{\partial L}{\partial w_{11}^{(1)}}=\frac{\partial L}{\partial y^{(out)}}\frac{\partial y^{(out)}}{\partial \boldsymbol{z}^{(out)}}\frac{\partial \boldsymbol{z}^{(out)}}{\partial \boldsymbol{y}^{(10)}}\cdots\frac{\partial \boldsymbol{y}^{(2)}}{\partial z_1^{(2)}}\frac{\partial z_1^{(2)}}{\partial y_1^{(1)}}\frac{\partial y_1^{(1)}}{\partial w_{11}^{(1)}} \tag{13}$$

비선형활성함수에 의한 기울기 소실을 보기위해 $\partial L/\partial y^{(out)}$과 $\partial y_1^{(1)}/\partial w_{11}^{(1)}$을 제외한 Equation (13)을 구성하고 있는 $\partial \boldsymbol{z}^{(m)}/\partial \boldsymbol{y}^{(m-1)}$항과 $\partial \boldsymbol{y}^{(m)}/\partial \boldsymbol{z}^{(m)}, m=2,\cdots,10,out$항 값의 크기를 iteration에 따라 각각 그린 그림은 Fig. 8과 같다. Fig. 8에서 $\partial \boldsymbol{z}^{(m)}/\partial \boldsymbol{y}^{(m-1)}$항과 $\partial \boldsymbol{y}^{(m)}/\partial \boldsymbol{z}^{(m)}$항은 Equation (14-16)과 같은 Jacobian 행렬로 나타나며 $\mathrm{XOR}_{(10)}$ 문제는 4개의 입력 데이터를 가지므로 $\partial \boldsymbol{z}^{(m)}/\partial \boldsymbol{y}^{(m-1)}$항과 $\partial \boldsymbol{y}^{(m)}/\partial \boldsymbol{z}^{(m)}$항은 각각 16개의 값을 가진다.

Sigmoid의 경우 $\partial \boldsymbol{z}^{(m)}/\partial \boldsymbol{y}^{(m-1)}$항은 Equation (14)과 같이 계산된다.

$$\frac{\partial \boldsymbol{z}^{(m)}}{\partial \boldsymbol{y}^{(m-1)}}=\begin{bmatrix} z_1^{(m)}\left(1-z_1^{(m)}\right) & 0 \\ 0 & z_2^{(m)}\left(1-z_2^{(m)}\right)\end{bmatrix} \tag{14}$$

PPNS의 경우 $\partial \boldsymbol{z}^{(m)}/\partial \boldsymbol{y}^{(m-1)}$항은 Equation (15)과 같이 계산된다.

$$\frac{\partial \boldsymbol{z}^{(m)}}{\partial \boldsymbol{y}^{(m-1)}}=\begin{bmatrix} z_1^{(m)}\left(1-\dfrac{z_1^{(m)}}{a_1^{(m-1)}}\right) & 0 \\ 0 & z_2^{(m)}\left(1-\dfrac{z_2^{(m)}}{a_2^{(m-1)}}\right)\end{bmatrix} \tag{15}$$

$\partial \boldsymbol{y}^{(m)}/\partial \boldsymbol{z}^{(m)}$항은 Sigmoid와 PPNS는 동일하게 Equation (16)과 같이 계산된다.

$$\frac{\partial \boldsymbol{y}^{(m)}}{\partial \boldsymbol{z}^{(m)}}=\begin{bmatrix} w_{11}^{(m)} & w_{21}^{(m)} \\ w_{12}^{(m)} & w_{22}^{(m)}\end{bmatrix} \tag{16}$$

Fig. 8을 보면 Sigmoid의 경우 iteration에 따른 $\partial \boldsymbol{z}^{(m)}/\partial \boldsymbol{y}^{(m-1)}$항의 값들이 0.25보다 같거나 작게 분포된 것을 볼 수 있으며 이는 $\partial \boldsymbol{z}^{(m)}/\partial \boldsymbol{y}^{(m-1)}$항이 역전파되어 누적될 때 기울기 소실을 발생시키는 주요 원인임을 알 수 있다. Sigmoid 경우에 Fig. 8(t)인 $\partial y^{(out)}/\partial \boldsymbol{z}^{(out)}$을 제외한 나머지 항들이 기울기 소실이 발생하여 iteration에 따른 값들이 거의 변하지 않는 것을 볼 수 있다. 그 결과로 Fig. 7(b)과 같이 Sigmoid의 $\partial L/\partial w_{11}^{(1)}$값이 소실되는 것을 볼 수 있다.

반면 PPNS의 경우 $\partial \boldsymbol{z}^{(m)}/\partial \boldsymbol{y}^{(m-1)}$항의 값들이 0.25보다 큰 약 1.5까지 분포되어 비선형활성함수에 의한 기울기 소실을 완화하여 그 결과 iteration에 따라 $\partial \boldsymbol{z}^{(m)}/\partial \boldsymbol{y}^{(m-1)}$항과 $\partial \boldsymbol{y}^{(m)}/\partial \boldsymbol{z}^{(m)}$항의 값들이 Sigmoid에 비해 상대적으로 크게 변하는 것을 볼 수 있다. 이 과정을 통해 Fig. 7(b)에서 PPNS에 대한 손실함수가 0에 수렴될 동안 Sigmoid에 비해 $\partial L/\partial w_{11}^{(1)}$값이 큰 값을 가지는 것을 볼 수 있다.

Fig. 7의 실험에서 사용된 모든 가중치 파라미터들이 Sigmoid를 사용했을 때 기울기 소실이 발생하는지 확인하고 PPNS를 사용했을 때 완화할 수 있는지를 전체적으로 보기위해 모든 가중치 파라미터들의 초깃값에서부터 학습이 끝난 후 학습된 값이 얼마나 이동하였는지를 Fig. 9에 나타내었다.

Sigmoid를 사용한 Fig. 9(a)는 출력층의 가중치 파라미터 $(w_1^{(out)}, w_2^{(out)})$을 제외한 모든 가중치 파라미터가 기울기 소실이 발생하여 학습된 후에도 거의 움직이지 않은 것을 확인할 수 있는 반면 PPNS를 사용한 Fig. 9(b)를 보면 상대적으로 Sigmoid보다 학습된 가중치 파라미터가 초깃값으로부터 멀리 떨어진 것을 확인할 수 있다. 이를 Fig. 7(a)과 함께 보면 Sigmoid는 기울기 소실이 발생하여 손실함수 값이 감소하지 못하였지만 PPNS는 기울기 소실을 완화하여 손실함수 값이 0에 수렴하였고 이때 대응되는 모든 가중치 파라미터가 Fig. 9(b)의 학습된 가중치 파라미터임을 알 수 있다.

Fig. 7의 실험에서 사용된 PPNS의 모든 은닉층의 파라메트릭 활성함수 파라미터 a,b가 초깃값 $a=6, b=0$에서 학습된 후 어떻게 분포하는지 나타낸 그림은 Fig. 10과 같다.

Fig. 10에서 PPNS의 파라메트릭 활성함수 파라미터는 Equation (11)과 Equation (12)을 통해 학습하였으며 손실함수를 최소화하는 방향으로 학습되었다. 7번째 은닉층의 파라메트릭 활성함수 파라미터 $(a_2^{(7)}, b_2^{(7)})=(6.046, -0.155)$값으로 모든 은닉층의 파라미터 a,b에서 가장 많이 이동한 것을 알 수 있으며 전체적으로 학습된 a,b의 분포를 보았을 때 초깃값 $a=6, b=0$에 머물지 않고 a는 6보다 큰 값으로, b는 약 −0.15에서0.15까지 이동한 것을 확인할 수 있다.

이상의 은닉층 수가 10개인 $\mathrm{XOR}_{(10)}$ 실험에서 비선형활성함수 Sigmoid에 의해 발생하는 기울기 소실을 확인하였다.

이를 파라메트릭 활성함수 파라미터 a,b를 적용한 PPNS가 파라미터 수가 증가하였음에도 불구하고 손실함수를 최소화하는 방향으로 학습하여 기울기 소실을 완화하였음을 확인하였다.

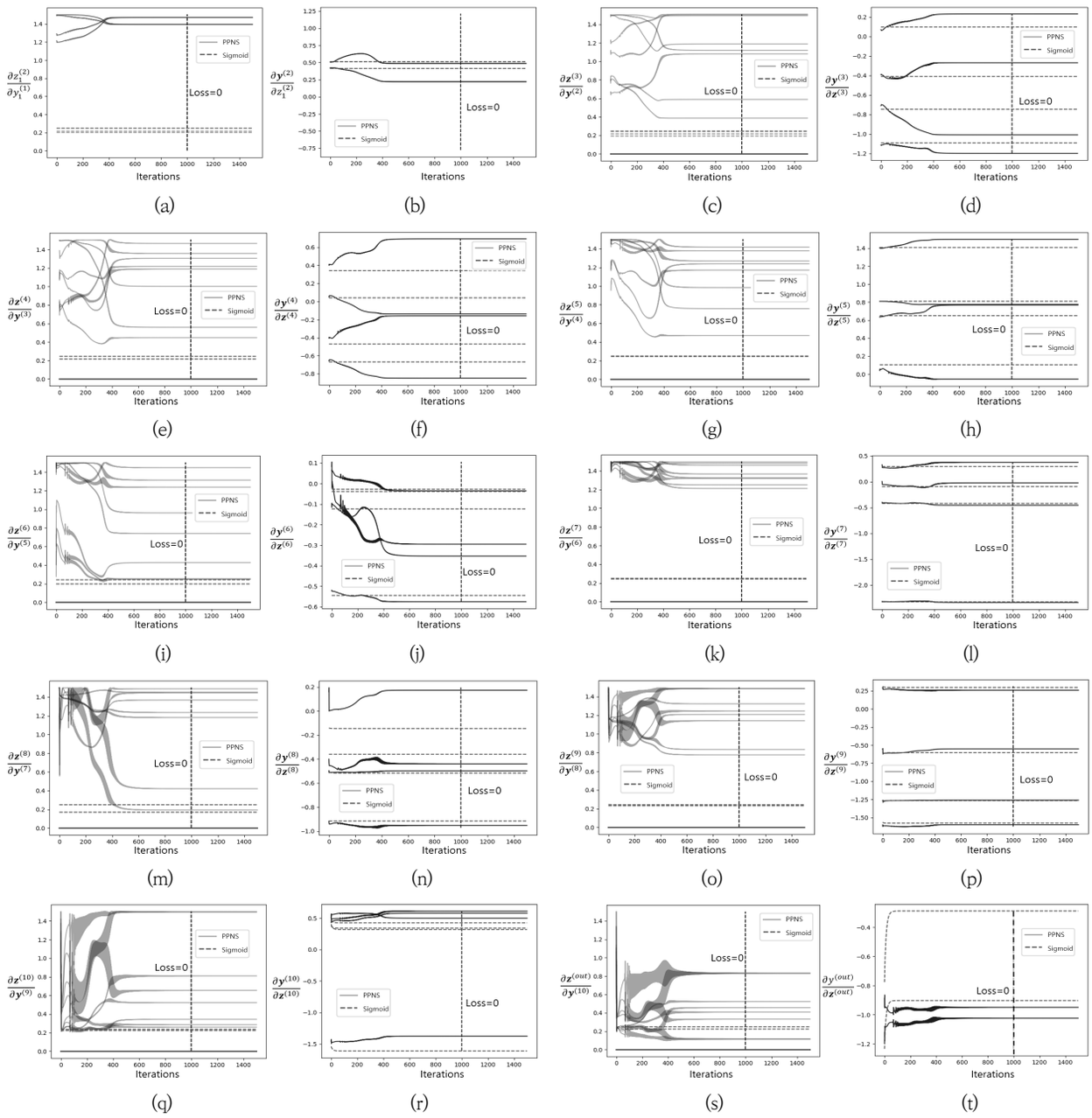


Fig. 8. $\left(\frac{\partial \boldsymbol{z}^{(m)}}{\partial \boldsymbol{y}^{(m-1)}}\right)_{ij}$ and $\left(\frac{\partial \boldsymbol{y}^{(m)}}{\partial \boldsymbol{z}^{(m)}}\right)_{ij}, m = 2, \cdots, 10, out$ Values According to Iterations

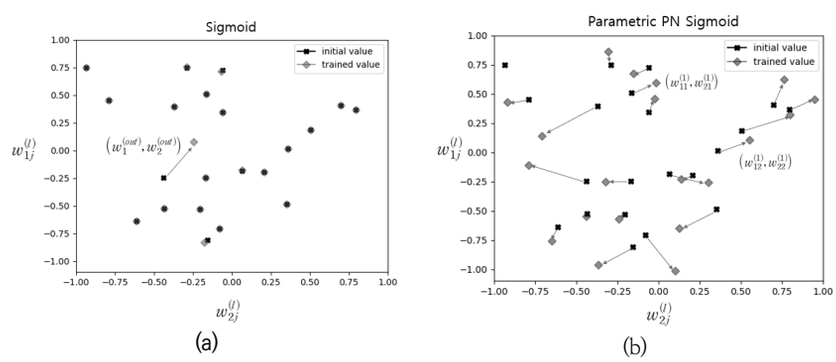


Fig. 9. Initial Value and Learned Value $(w_{1j}^{(l)}, w_{2j}^{(l)}), l = 1, \cdots, out, j = 1, 2$ using (a) Sigmoid (b) PPNS

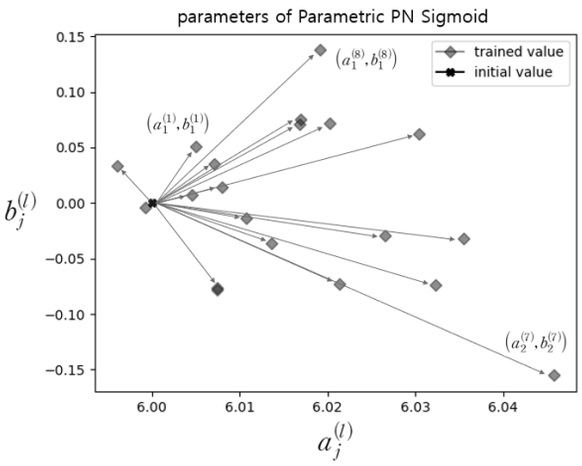


Fig. 10. Distribution of Initial Values and Learned Values of PPNS Parameter $(a_j^{(l)}, b_j^{(l)}), l = 1, \cdots, 10, j = 1, 2$

2) 은닉층 수가 10개인 $XOR_{(10)}$문제를 이용한 기존 활성함수와 파라메트릭 활성함수의 기울기 소실 완화 비교

대표적인 비선형활성함수 ReLU, Sigmoid, Tanh와 Table 1과 같이 파라메트릭 활성함수 파라미터 a,b를 적용한Parametric ReLU, Parametric Sigmoid, PPNS, Parametric Tanh을 Fig. 6과 같은 $XOR_{(10)}$ 문제에 대해 기울기 소실 문제를 완화할 수 있는지 실험하였다.

파라메트릭 활성함수 파라미터 a,b의 초깃값을 제외한 실험조건은 Table 3과 동일하며 각각의 비선형활성함수를 1,000회씩 실험하여 손실함수 값이 0에 수렴한 횟수를 세어본 결과와 그에 대응하는 파라메트릭 활성함수 파라미터 a,b의 초깃값은 Table 5와 같다. Table 5에서 Parametric Sigmoid와 PPNS의 파라메트릭 활성함수 파라미터를 $a=6$, Parametric ReLU와 Parametric Tanh의 파라메트릭 활성함수 파라미터를 $a=1.5$로 초기화한 이유는 초반 iteration에서 비선형활성함수의 기울기 절댓값이 1보다 작아 비선형활성함수에 의해 기울기 소실이 발생하는 것을 완화하고자 모든 비선형활성함수의 기울기 절댓값을 1.5로 만든 것이다.

Table 5. Results of Experiments Conducted 1000 Times

Activation function	Initialization of a,b	number of convergence with Loss=0
Sigmoid		0
Parametric Sigmoid	$a=6, b=0$	29
PPNS	$a=6, b=0$	744
Tanh		628
Parametric Tanh	$a=1.5, b=0$	700
ReLU		11
Parametric ReLU	$a=1.5, b=0$	8

실험 결과 Table 5에서 Sigmoid는 모든 경우에 기울기 소실 문제가 발생하여 한 번도 수렴하지 못한 반면 $a=6$으로 초기화한 Parametric Sigmoid와 PPNS는 Sigmoid보다 많은 각각 29번, 744번 수렴하였다. Parametric Sigmoid와 PPNS의 함수 모양은 $-a/2$ 차이만 있음에도 불구하고 손실함수 값이 0에 수렴한 횟수가 크게 차이가 났다.

Tanh와 $a=1.5$로 초기화한 Parametric Tanh를 비교했을 때 Tanh는 628회, Parametric Tanh는 700회로 Parametric Tanh가 72회 더 많이 손실함수 값이 0에 수렴하였다.

ReLU는 손실함수 값이 0에 수렴한 횟수가 1000회 중에 11회인데 반해 Parametric ReLU는 3회 적은 8회 수렴하였다.

이상의 은닉층이 10개인 $XOR_{(10)}$ 문제에 대해 파라메트릭 활성함수가 기존의 비선형활성함수보다 기울기 소실을 완화할 수 있는지를 10,000번의 iteration안에 얼마나 손실함수 값이 0에 수렴하는 지로 실험해보았다. 실험결과 Tanh와 Sigmoid보다 파라메트릭 활성함수 파라미터 a,b를 적용한 Parametric Tanh, Parametric Sigmoid와 PPNS가 10,000번의 iteration안에 손실함수 값이 0에 수렴한 횟수가 더 많은 것을 확인하였다. 특히 Sigmoid의 경우에 PPNS를 사용할 경우 손실함수 값이 0에 수렴한 횟수가 744회로 한 번도 수렴하지 못한 Sigmoid보다 기울기 소실 완화에 매우 우월한 성능을 가짐을 확인하였다.

4.2 MNIST 분류문제를 이용한 파라메트릭 활성함수의 기울기 소실 완화 성능실험

파라메트릭 활성함수를 보다 다양한 비선형활성함수들과 비교하기 위해 Tanh와 기울기 소실 문제를 해결하기 위해 연구된 비선형활성함수 ReLU, ReLTanh, Hexpo 그리고 파라메트릭 활성함수 파라미터를 적용한 Parametric ReLU, Parametric Tanh, Parametric Hexpo, PPNS 총 8개 함수에 대해 비교하였다(Table 6).

Table 7은 실험에 앞서 비선형활성함수의 파라미터 초기화 방법을 나타낸 것으로 Hexpo와 ReLTanh의 파라미터 초기화는 각 [16,18]에서 사용한 초기화 방법을 사용하였다. Table 7의 Parametric Hexpo는 Hexpo에 파라메트릭 활성함수의 위치를 결정하는 파라미터 h를 적용한 것이다. 이때 Hexpo는 [16]에서 소개한대로 각 은닉층에서 모든 노드들이 비선형활성함수 파라미터 a,b,c,d를 공유하였고 Hexpo를 제외한 나머지 Table 7의 파라메트릭 활성함수들은 각 은닉층의 각 노드들이 파라메트릭 활성함수 파라미터를 가지고 있다.

MNIST 실험조건은 [16]에서 Hexpo가 기울기 소실 문제를 완화할 수 있는지를 확인하기 위해 사용된 MNIST 실험조건을 사용하여 비교했으며 그 조건은 Table 8과 Fig. 11과 같다.

Table 6. Various Activation Functions Used in the MNIST Experiment

Activation function	Formula
ReLU	$\begin{cases} y, y>0 \\ 0, y \le 0 \end{cases}$
Parametric ReLU	$\begin{cases} a(y-b), & y \ge b \\ 0, & y<b \end{cases}$
Tanh	$\frac{e^{2y}-1}{e^{2y}+1}$
Parametric Tanh	$a\left(\frac{e^{2(y-b)}-1}{e^{2(y-b)}+1}\right)$
ReLTanh	$\begin{cases} \tanh'(\lambda^+)(y-\lambda^+)+\tanh(\lambda^+), & y \ge \lambda^+ \\ \tanh(y), & \lambda^- < y < \lambda^+ \\ \tanh'(\lambda^-)(y-\lambda^-)+\tanh(\lambda^-), & y \le \lambda^- \end{cases}$
Hexpo	$\begin{cases} -a(e^{-\frac{y}{b}}-1), & y \ge 0 \\ c(e^{\frac{y}{d}}-1), & y<0 \end{cases}$
Parametric Hexpo	$\begin{cases} -a(e^{-\frac{(y-h)}{b}}-1), & y \ge 0 \\ c(e^{\frac{(y-h)}{d}}-1), & y<0 \end{cases}$
PPNS	$\frac{a}{1+e^{-(y-b)}}-\frac{a}{2}$

Table 7. Initialization of Activation Function Parameters

Activation function	Initialization of parameters
Parametric ReLU	$a=1.5, b=0$
Parametric Tanh	$a=1.5, b=0$
ReLTanh	$\lambda^+=0, \lambda^-=-1.5$
Hexpo	$a,c=\left(\frac{i}{2}+1\right),\ b,d=\left(\frac{i}{8}+1\right),$ $i=i^{th}$ hidden layer
Parametric Hexpo	Same as Hexpo, $h=0$
PPNS	$a=6, b=0$

Table 8. Experiment Conditions

Data preprocessing	Min-Max normalization
Train data	60,000
Test data	10,000
Batch size	64
Algorithm	Gradient descent
Learning rate	0.01
Steps	10,000
Loss function	Cross-Entropy
Initialization of W parameters	Xavier, (ReLU, Parametric ReLU : He initialization)
Number of hidden layers	8
Number of nodes in each hidden layer	128
Regularization	X

실험결과를 나타내는 Fig. 12에서 각 step은 훈련데이터의 배치 크기를 학습했을 때 시험데이터 10,000개에 대한 손실함수 값을 8가지 비선형활성함수에 대해 나타낸 것이다. ReLU가 다른 비선형활성함수들에 비해 손실함수 값이 가장 느리게 감소하였으며 ReLTanh는 Tanh보다 약간 앞서서 손실함수 값이 감소하는 것을 볼 수 있다. Parametric ReLU의 경우 ReLU, Tanh 그리고 ReLTanh보다 손실함수 값이 상대적으로 빠르게 감소한 것을 볼 수 있다.

0번째 step에서 650번째 step 구간을 확대해서 보면 Parametric Tanh, Hexpo, Parametric Hexpo 그리고 PPNS가 다른 비선형활성함수들에 비해 초기 기울기 소실이 일어나지 않아 손실함수 값이 상대적으로 빠르게 감소하는 것을 볼 수 있다. Parametric Hexpo와 Hexpo 그리고 PPNS는 거의 같은 속도로 손실함수 값이 감소하였다. 10,000 step까지 결과 Parametric Hexpo의 손실함수 값이 0.08, Hexpo의

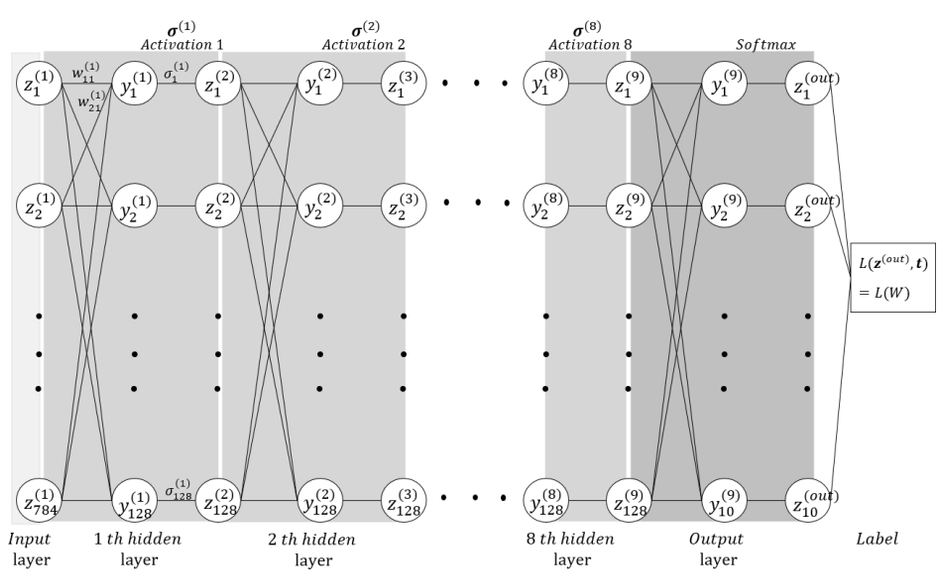


Fig. 11. Deep Neural Network Structure for MNIST Problems with 8 Hidden Layers

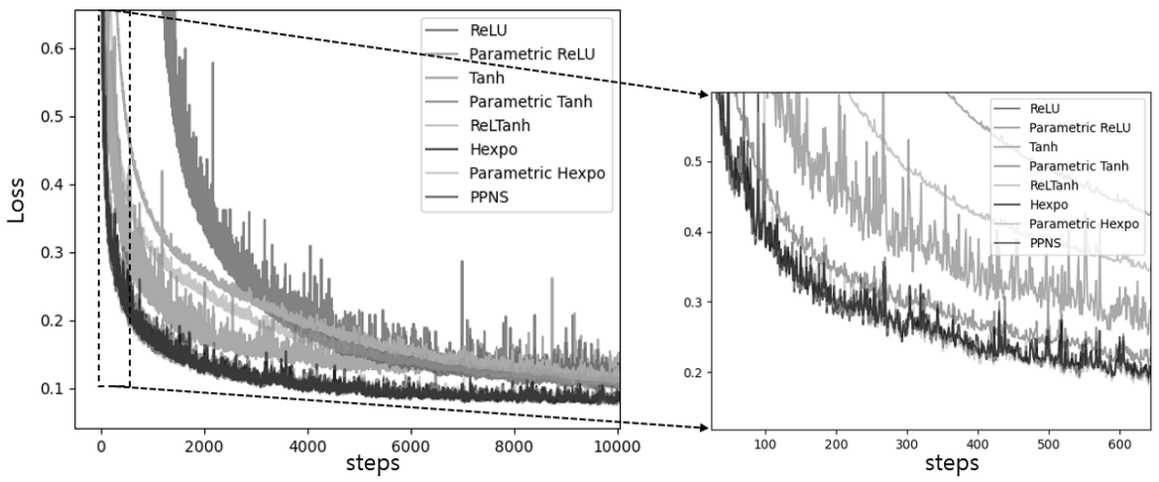


Fig. 12. MNIST Test Results for Various Activation Functions

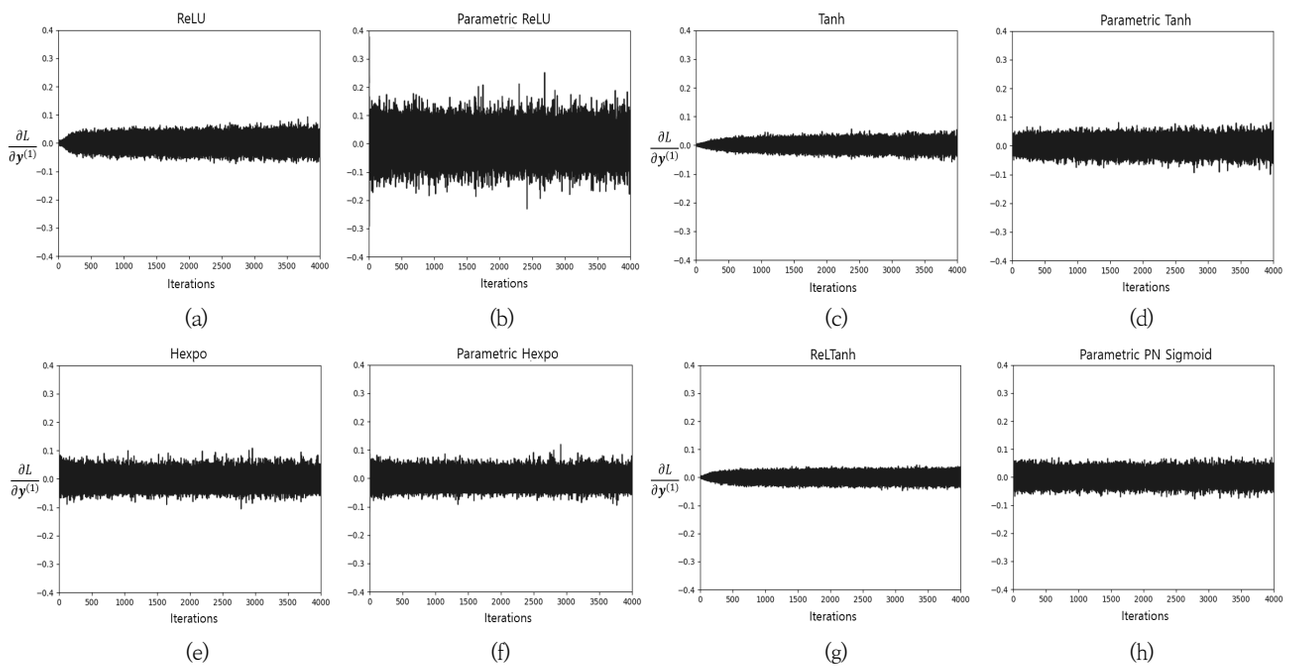


Fig. 13. The Range of the Gradient of the Loss Function with Respect to y of the First Hidden Layer of Various Activation Functions

손실함수 값이 0.077, Parametric Tanh와 PPNS의 손실함수 값이 0.076으로 Parametric Tanh와 PPNS의 손실함수 값이 가장 낮음을 확인하였다.

파라메트릭 활성함수의 학습속도에 대해 평가하기 위해 대표적으로 많이 사용되는 ReLU함수를 파라메트릭 활성함수 PPNS와 비교하였고 60,000개의 훈련데이터에 대한 손실함수 값이 0.1까지 감소하는데 걸린 시간을 비교해본 결과, PPNS함수가 ReLU함수보다 약 2/3정도 학습시간이 줄어든 것을 확인하였다.

Table 6에서 실험한 비선형활성함수들의 기울기 소실 완화를 좀 더 구체적으로 확인하기 위해 상대적으로 기울기 소실이 발생하기 쉬운 1번째 은닉층의 선형변환된 벡터 $\boldsymbol{y}^{(1)}$에 의한 손실함수의 변화율인 Equation (17)을 관찰하였다.

$$\frac{\partial L}{\partial \boldsymbol{y}^{(1)}} = \frac{\partial L}{\partial \boldsymbol{z}^{(out)}} \cdots \frac{\partial \boldsymbol{z}^{(9)}}{\partial \boldsymbol{y}^{(8)}} \cdots \frac{\partial \boldsymbol{z}^{(3)}}{\partial \boldsymbol{y}^{(2)}} \frac{\partial \boldsymbol{y}^{(2)}}{\partial \boldsymbol{z}^{(2)}} \frac{\partial \boldsymbol{z}^{(2)}}{\partial \boldsymbol{y}^{(1)}} \tag{17}$$

8개 비선형활성함수에 대한 Equation (17)을 훈련 데이터에 대한 학습과정 0번째 step에서 4000번째 step까지 관찰하였으며 각 step에서 Equation (17) 기울기의 최댓값과 최솟값을 그렸다(Fig. 13).

Fig. 13에서 ReLU는 0번째 step에서 200번째 step 구간에서 0에 가까운 작은 기울기의 최대최솟값을 가지는 것에

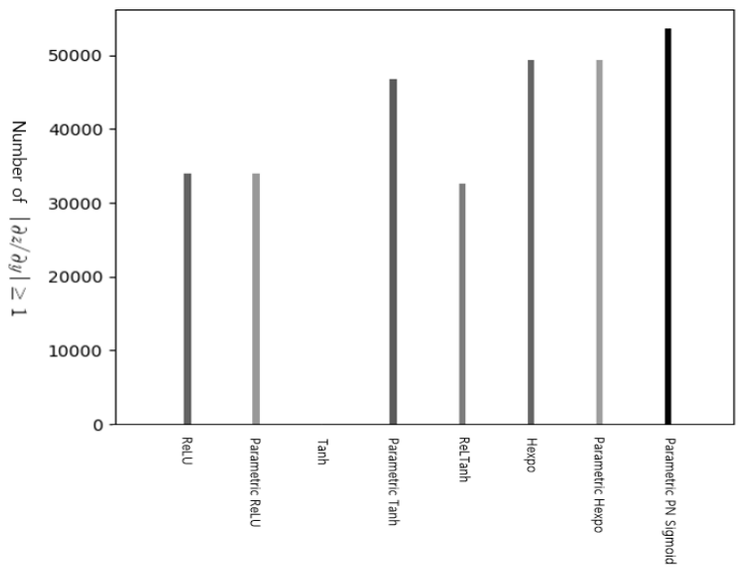


Fig. 14. Number of $\left|\left(\frac{\partial \boldsymbol{z}^{(m)}}{\partial \boldsymbol{y}^{(m-1)}}\right)_{ij}\right| \geq 1$ for Various Activation Functions

반해 Parametric ReLU의 경우 4000번째 step까지 최대최솟값의 범위가 0.4로 상대적으로 큰 것을 확인할 수 있다.

Tanh와 Parametric Tanh을 비교했을 때 Tanh의 경우도 마찬가지로 0번째 step에서 200번째 step 구간에서 작은 최대최솟값 기울기를 나타내는데 반해 Parametric Tanh은 상대적으로 Tanh에 비해 큰 최대최솟값을 가지는 것을 알 수 있다.

Hexpo와 위치를 결정하는 파라미터 h을 적용하고 모든 은닉층의 모든 노드가 각각의 파라미터를 가지는 Parametric Hexpo의 경우 4000번째 step까지 Hexpo와 비슷한 것을 확인할 수 있다.

ReLTanh는 Tanh와 비슷하게 0번째 step에서 200번째 step 구간에서 작은 최대최솟값을 나타낸다. 마지막으로 PPNS의 경우 ReLU, Tanh 그리고 ReLTanh에 비해 0번째 step에서 200번째 step 구간에서 Equation (17)의 최대최솟값이 큰 것을 확인할 수 있다.

Fig. 13을 전체적으로 보면 크기와 위치를 결정하는 파라미터를 적용한 파라메트릭 활성함수가 기존 활성함수에 비해 Equation (17)의 최대최솟값이 비슷하거나 큰 것을 확인할 수 있다.

비선형활성함수의 기울기 크기를 확인하기 위해 8개의 은닉층에 대해 각 은닉층에서 비선형활성함수의 기울기 값을 나타내는 $\left|\left(\partial \boldsymbol{z}^{(m)}/\partial \boldsymbol{y}^{(m-1)}\right)_{ij}\right|, m=2,\cdots,9$값이 1과 같거나 큰 개수를 세어보았다. Fig. 14는 Table 8에서 사용한 MNIST 실험에서 배치 크기 64개 데이터에 대한 0번째 step때 $\left|\left(\partial \boldsymbol{z}^{(m)}/\partial \boldsymbol{y}^{(m-1)}\right)_{ij}\right| \geq 1$ 개수를 나타낸 그림이다.

Fig. 14를 보면 ReLU와 Parametric ReLU는 $\left|\left(\partial \boldsymbol{z}^{(m)}/\partial \boldsymbol{y}^{(m-1)}\right)_{ij}\right| \geq 1$ 개수가 34,030개이다. Tanh는 최대 미분 값이 1이지만 1과 같거나 큰 값이 관측되지 않았으며 Parametric Tanh의 경우 46,694개로 Tanh에 비해 많은 것을 알 수 있다. ReLTanh는 32,606개, Hexpo와 Parametric Hexpo는 49,384개이며 마지막으로 PPNS는 53,566개로 8개의 비선형활성함수 중 가장 많은 것을 확인할 수 있다.

Fig. 14와 Fig. 12를 같이 보면 손실함수 값이 0번째 step부터 가장 빨리 감소하는 비선형활성함수는 Parametric Tanh, Hexpo, Parametric Hexpo 그리고 PPNS이며 이때 다른 4개의 비선형활성함수들보다 $\left|\left(\partial \boldsymbol{z}^{(m)}/\partial \boldsymbol{y}^{(m-1)}\right)_{ij}\right| \geq 1$의 개수가 상대적으로 많은 것을 알 수 있다. 이는 파라메트릭 활성함수가 기존 비선형활성함수에 의해 발생하는 기울기 소실을 완화할 수 있음을 보여준다.

5. 결 론

기울기 소실 문제는 은닉층이 깊은 모든 신경망에서 발생할 수 있는 중요한 문제이다. 기울기 소실에 대한 원인 중 비선형활성함수의 영향으로 발생할 수 있는 부분을 크기와 위치를 결정하는 파라미터를 적용한 파라메트릭 활성함수를 사용함으로써 비선형활성함수의 기울기 크기에 자유로운 손실함수 공간에서 손실함수를 최소화하는 방향으로 최적화하여 기울기 소실 문제를 완화할 수 있다.

파라메트릭 활성함수의 기울기 소실 문제 완화를 확인하기 위해 기울기 소실을 쉽게 구현할 수 있는 인위적으로 만든 은닉층 수가 10개인 $XOR_{(10)}$문제에 대해 실험하였다. 대표적인 비선형활성함수 ReLU, Tanh 그리고 Sigmoid에 파라메트릭 활성함수 파라미터를 적용하여 파라메트릭 활성함수가 기존의 비선형활성함수보다 기울기 소실을 완화할 수 있는지를 10,000번의 iteration안에 얼마나 손실함수 값이 0에 수렴하는 지로 실험해보았다. 1000회를 실험한 결과 Tanh는

628회, Parametric Tanh는 700회 수렴하였으며 Sigmoid는 0회, Parametric Sigmoid는 29회, Parametric PN Sigmoid는 744회로 파라메트릭 활성함수 파라미터를 적용한 파라메트릭 활성함수가 기울기 소실 완화에 우수한 성능을 가짐을 확인하였다. 특히 Sigmoid의 경우 비선형활성함수에 의한 기울기 소실이 주원인임을 확인하였으며 이를 Parametric PN Sigmoid가 완화할 수 있음을 확인하였다.

은닉층 수가 8개인 MNIST 분류문제에서 Tanh와 기울기 소실 문제를 해결하기 위해 연구된 비선형활성함수 ReLU, ReLTanh, Hexpo 그리고 파라메트릭 활성함수 파라미터를 적용한 Parametric ReLU, Parametric Tanh, Parametric Hexpo, Parametric PN Sigmoid 총 8개 함수에 대해 비교하였다. 그 결과 크기와 위치를 결정하는 파라미터를 적용한 파라메트릭 활성함수가 기존 활성함수보다 기울기 소실 완화에 우월한 성능을 가짐을 확인하였다.

특히 기울기 소실이 상대적으로 발생하기 쉬운 1번째 은닉층의 선형변환된 벡터에 대한 손실함수 변화율의 최대최솟값을 학습 step에 따라 관찰한 결과 크기와 위치를 결정하는 파라미터를 적용한 파라메트릭 활성함수가 기존 활성함수보다 큰 값을 가짐을 확인하였다. 또한 위치를 결정하는 파라미터 h을 적용하고 모든 은닉층의 모든 노드가 각각의 파라미터를 가지는 Parametric Hexpo를 Hexpo와 비교한 결과 손실함수 값이 거의 같은 속도로 감소하는 것을 확인하였다.

이상의 $XOR_{(10)}$, MNIST 두 실험 모두 파라메트릭 활성함수 파라미터를 적용함으로써 기존의 비선형활성함수보다 학습해야 하는 파라미터 수가 증가하였음에도 불구하고 입력 데이터에 따라 손실함수를 최소화하는 방향으로 학습하면서 기울기 소실 완화와 손실함수 감소에 모두 우수한 성능을 가짐을 확인하였다.

본 논문은 기울기 소실이 발생할 수 있는 임의의 비선형활성함수에 대해 크기와 위치를 결정하는 파라메트릭 활성함수 파라미터를 간단히 적용하여 기울기 소실 문제를 완화할 수 있다는 점에서 의의가 있다.

References

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

[2] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the Loss Landscape of Neural Nets," *arXiv: 1712.09913*, 2018.

[3] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen netzen," Diploma Thesis, Institut fur Informatik, Lehrstuhl Prof. Brauer, Technische Universit atMunchen, 1991.

[4] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: The difficulty of learning long-term dependencies," IEEE, 2001.

[5] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Artificial Intelligence and Statistics*, Vol.9, 2010.

[6] V. Nair and G. Hinton, "Rectified linear units improve restricted boltzmann machines," *International Conference on Machine Learning*, pp.807-814, 2010.

[7] 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.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, Vol.9, No.8, pp.1735-1780, 1997.

[9] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Computation*, Vol.12, No.10, pp.2451-2471, 2000,

[10] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv:1412.3555*, 2014.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *arXiv:1502.01852*, 2015.

[12] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, Vol.18, No.7, pp.1527-1554, 2006.

[13] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methodsfor online learning and stochastic optimization," *The Journal of Machine Learning Research*, Vol.12, No.61, pp.2121-2159, 2011.

[14] M. D. Zeiler, "ADADELTA: An adaptive learning ratemethod," *arXiv:1212.5701*, 2012.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980, 2014.

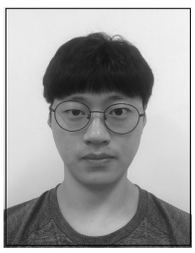
[16] S. Kong and M. Takatsuka, "Hexpo: A vanishing-proof activation function," *International Joint Conference on Neural Networks*, pp.2562-2567, 2017.

[17] 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," *Institute of Electrical and Electronics Engineers*, Vol.66, No.5, pp.3814-3824, 2018.

[18] 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.

[19] R. Pascanu, T. Mikolov, and Y. Bengio, "Understanding the exploding gradient problem," *arXiv:1211.5063*, 2012.

[20] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," *arXiv:1211.5063*, 2013.
[21] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical Evaluation of Rectified Activations in Convolution Networkm," *arXiv: 1505.00853*, 2015.
[22] S. Basodi, C. Ji, H. Zhang, and Y. Pan, "Gradient amplification: An efficient way to train deep neural networks," Big Data mining and Analytics, Vol.3, No.3, pp.196-207, 2020.



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 인공지능학과 교수
관심분야 : Data Science & Artificial Intelligence