The Korean Journal of Applied Statistics (2019) DOI: http://dx.doi.org/10.5351/KJAS.2019.32.5.693
32(5), 693–702

Initialization by using truncated distributions in artificial neural network

MinJong Kim[a] · Sungchul Cho[a] · Hyerin Jeong[a] · YungSeop Lee[b] · Changwon Lim[a,1]

[a]Department of Applied Statistics, Chung-Ang University;
[b]Department of Statistics, Dongguk University

(Received June 24, 2019; Revised August 9, 2019; Accepted August 20, 2019)

Abstract

Deep learning has gained popularity for the classification and prediction task. Neural network layers become deeper as more data becomes available. Saturation is the phenomenon that the gradient of an activation function gets closer to 0 and can happen when the value of weight is too big. Increased importance has been placed on the issue of saturation which limits the ability of weight to learn. To resolve this problem, Glorot and Bengio (*Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 249–256, 2010) claimed that efficient neural network training is possible when data flows variously between layers. They argued that variance over the output of each layer and variance over input of each layer are equal. They proposed a method of initialization that the variance of the output of each layer and the variance of the input should be the same. In this paper, we propose a new method of establishing initialization by adopting truncated normal distribution and truncated cauchy distribution. We decide where to truncate the distribution while adapting the initialization method by Glorot and Bengio (2010). Variances are made over output and input equal that are then accomplished by setting variances equal to the variance of truncated distribution. It manipulates the distribution so that the initial values of weights would not grow so large and with values that simultaneously get close to zero. To compare the performance of our proposed method with existing methods, we conducted experiments on MNIST and CIFAR-10 data using DNN and CNN. Our proposed method outperformed existing methods in terms of accuracy.

Keywords: initialization, saturation, Xavier initialization, truncated distribution, deep learning

1. 서론

오늘날 과학기술의 발달로 대용량의 데이터들이 급격하게 증가하면서 딥러닝(deep learning)의 구조 역시 점점 더 복잡하고 깊은 층으로 만들어지고 있다. 딥러닝 층이 깊어지면서 학습 속도가 느려지거나 도중에 학습이 되지 않는 포화 현상(saturation)이 자주 발생하기 시작했다. 초기값 설정이란 딥러닝 모델에서 가중치를 학습시킬 때 처음 들어가는 가중치 값을 말하며 최근까지도 활발한 연구가 진행되고 있다

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT (NRF-2017 M3C4A7083281).

[1]Corresponding author: Department of Applied Statistics, Chung-Ang University, 84 Heukseok-ro, Dongjak-gu, Seoul 06974, Korea. E-mail: clim@cau.ac.kr

(Sutskever 등, 2013; Mishkin과 Matas, 2015; Hayou 등, 2015; Humbrid 등, 2018; Hanin과 Rolnick, 2018). Mishkin과 Matas (2015)은 정형화된 매트릭스로 각 층의 가중치를 초기화한 후 첫 번째 층부터 마지막 층의 출력 분산을 동일하게 표준화한 방법인 Layer-sequentialunit-variance (LSUV)를 제안했다. Krahenbuhl 등 (2015)은 K-means 방법과 PCA 방법에 기반하여 데이터에 의존하는 초기값 설정을 통해 폭발(exploding)과 경사감소 소멸(vanishing gradient)을 피하며 모든 유닛이 비슷한 속도에서 학습할 수 있는 초기값 설정 방법을 제안하였다. Humbrid 등 (2018)은 의사결정나무기반의 포워드 신경망을 구축하고 초기값 설정을 위한 자동화된 프로세스를 제시하였다.

초기값 설정에 대해 알려진 사실 중 하나는 초기값을 모두 똑같은 상수로 하거나 0으로 설정하면 안된다는 것이다. 초기값을 모두 똑같은 상수로 설정하면 신경망에 있는 모든 노드(node)들은 같은 신호 값을 받게 된다. 이에따라 각 출력 노드의 출력 값 역시 똑같은 값으로 나오게 된다. 또한 오차를 역전파(backpropagation)함으로써 가중치를 업데이트하는 과정에서 오차는 모두 같은 값으로 나뉘어 전파되고 결국 동일한 가중치 업데이트로 이어지게 된다. 이는 또다시 동일한 값을 가지는 가중치라는 결과로 이어지게 된다. 보통 동일한 가중치를 가지는 것을 대칭이라고 표현한다. 결국 같은 입력에 대해 동일하거나 대칭적인 가중치로 초기값을 설정하면 활성화 함수를 거치면서 같은 방향으로 계속 집중될 수밖에 없고 결국 동일한 계산을 수행하며 네트워크 전체를 대칭으로 만들게 되는 것이다 (Goodfellow 등, 2014).

Sutskever 등 (2013)은 특히 초기값을 0으로 설정하면 동일한 가중치로 업데이트되는 문제도 있지만 입력신호에 의해 좌우되는 가중치 업데이트 값들은 초기값 0으로 인해 모두 0이 되어 버림으로써 가중치를 업데이트하는 능력을 완전히 상실하게 됨을 확인하였다. 따라서 초기값 설정은 대칭성을 깨는 방법으로 설정 한다. LeCun 등 (1998a)은 대칭성을 깨기 위해 가중치의 평균을 0으로 잡고 가중치의 분산을 조정해서 초기값을 설정하는 LeCun initialization 방법을 제안했다. LeCun initialization 방법은 활성화 함수 Relu가 제안되기 전 0 근처에서는 선형함수에 가깝고 위아래 값이 -1 혹은 $+1$로 정해져 있는 Sigmoid에서는 좋은 결과를 보여줬다. 하지만 LeCun initialization 방법은 층이 깊어지면서 포화현상이 발생하는 문제가 발생하였다. 포화 현상을 해결하기 위해 Glorot과 Bengio (2010)은 균일분포의 분산을 층별로 동일하게 적용한 후 초기값 설정을 하는 Xavier initialization 방법을 제안하였다. He 등 (2015)은 활성화 함수 ReLu를 제안하면서 기존 Xaiver initialization 방법은 Relu와 맞지 않기 때문에 ReLu에 적합한 새로운 초기값 설정 방법인 He initialization 방법을 제안하였다. 그들은 Relu가 음수 쪽 신호를 완전히 없애 버린다는 것에 착안해 분산을 두배 해줘야 분산을 유지할 수 있다고 생각했으며 0에 몰려있는 정규분포를 이용해 음수쪽 분산을 층별로 동일하게 적용한 후 초기값을 설정하였다.

정규분포는 0에 몰려있으므로 대부분 초기값이 0에 근사한 작은 값으로 나온다. 비록 작은 확률이지만 양 극값에서 초기값이 설정된다면 포화 현상이 발생할 수 있다. 이런 문제를 방지하기 위해 절단된 분포로부터 초기값을 설정을 하기도 한다. 하지만 절단된 분포의 자르는 위치를 정하는 방법은 없었고 임의로 적당한 위치에서 자르는 방식으로 사용되고 있다. 본 논문에서는 Glorot과 Bengio (2010)가 제안한 초기값 설정 방법을 절단된 분포에서 적용하여 절단된 분포에서 초기값을 설정하는 새로운 방법을 제안하고자 한다. 제안된 방법을 사용함으로써 절단된 분포를 사용할 때에도 포화 현상을 방지할 수 있는 초기치 설정이 가능하게 된다.

본 논문은 총 4장으로 구성되어 있다. 2장에서는 포화 현상에 대한 설명과 본 연구의 배경이 되는 초기값 설정 방법인 Xavier initialization 방법과 본 연구에서 제안하는 방법을 서술하였다. 3장에서는 절단된 분포에서 초기값 설정 방법을 MNIST (LeCun 등, 1998b) 데이터를 이용해 deep neural networks (DNN) 모델에 학습한 실험한 결과와 CIFAR-10 (Krizhevsky와 Hinton, 2009) 데이터를 convolutional neural networks (CNN) 모델에 학습한 실험한 결과를 기존 초기값을 사용했을 때의 성능을 비교하여

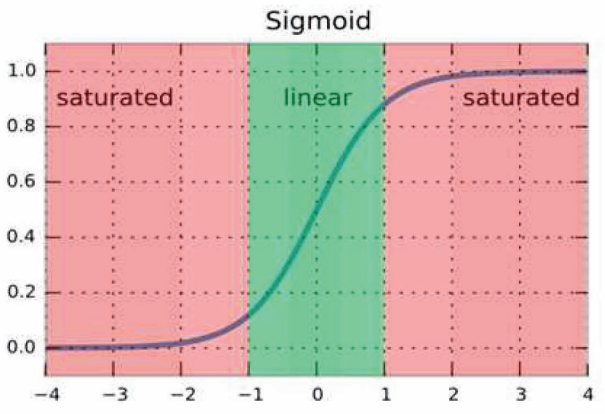

Figure 2.1. Separate sigmoid linear and nonlinear region.

실제로 성능이 좋아지는지를 비교하였다. 4장에서는 본 연구의 결론과 앞으로의 연구 방향에 대해 다루었다.

2. 초기값 설정 방법론

2.1. 포화 현상

포화 현상은 활성화 함수 Sigmoid나 Tanh에서 주로 발생하는 문제이다. Clevert 등 (2015)은 활성화 함수에 들어가는 입력값의 크기가 크면 활성화 함수의 값은 Figure 2.1에서 초록색으로 칠해진 선형영역에서 나오지 않고 빨간색으로 칠해진 비선형 영역에서 나오고 결국 0 혹은 1로 출력됨을 보였다.

Figure 2.1에서 활성화 함수의 출력값이 0, 1이 되면 활성화 함수값의 기울기가 0이 되는 것을 확인할 수 있다. 일반적인 Sigmoid와 그 기울기 식은 각각 다음과 같다.

$$\sigma(x) = \frac{1}{1+e^{-x}}; \qquad \sigma'(x) = \sigma(x)[1-\sigma(x)],$$

여기서 x에 들어가는 입력값의 크기가 커짐에 따라 Sigmoid 기울기가 0에 근사하게 되고 결국 역전파 과정에서 활성화 함수의 기울기 값으로 인해 역전파가 진행되지 않게 된다. 이는 더 나은 가중치로 업데이트해 나가는 학습능력을 저하시킨다. 결국 포화 현상은 기울기 소멸 현상을 야기할 수 있다.

2.2. 기존 초기치 설정 방법

기존에 사용 중인 초기값 설정 방법들은 Table 2.1과 같다. 초기값 설정 방법 중 많이 쓰이는 방법인 Xavier방법은 Glorot과 Bengio (2010)은 층별 활성화 값의 분산이 같다고 가정했을 때 초기값을 어떻게 설정해야 하는지를 이론적으로 유도를 하였고 유도를 하기 위해 몇 가지 가정을 세웠다: (i) 활성화 함수는 선형관계에 있다. (ii) $f'(0) = 1$이다. 이때 f은 활성화 함수이다. (iii) 가중치와 활성화 함수의 평균값은 0이고 서로 독립 관계에 있다. (iv) $f(x)$는 원점을 지나는 함수이다. 이들이 제안한 Xavier initialization 방법론에 사용되는 기본 수식에서 $\mathbf{s}^i$는 $(i+1)$번째 층에서 활성화 함수의 입력값이며 $\mathbf{z}^i$는 i번째 층에서 $(i+1)$번째 층으로 들어가는 입력값이고 $\mathbf{b}^i$는 편향 값을 의미하며 Xavier 방법론과 본 연구에서는 편향 값을 0으로 가정 하여 실험을 하였다. Xavier initialization 방법에서 기본 수

Table 2.1. Feature for popular initialization methods

Popular initialization methods	Feature
LeCun (Uniform)	상한과 하한의 값에 따라 분포 변화가 가능하고 초기값이 모두 똑같은 상수로 설정되는 것을 제한할 수 있음
LeCun (Normal)	평균과 분산 값에 따라 분포 변화가 가능하고 초기값은 주로 평균값 0 주위에서 많이 설정되도록 할 수 있음
Truncated normal	작은 확률이지만 정규분포의 꼬리 부분에서 큰 초기값이 뽑힐 수 있기 때문에 양쪽 꼬리를 절단시킴으로써 큰 초기값이 나오는 것을 제한함
Xavier	각층의 입력과 출력에 대한 분산값을 동일하게 함으로써 포화 현상을 방지할 수 있고 활성화 함수 sigmoid, tanh에서 효과적
He	활성화 함수 relu를 제안하면서 relu가 음수 쪽 신호를 완전히 없애 버린다는 것에서 착안해 분산을 두배 해줌으로써 분산을 유지 시켜준 relu에 최적화된 초기값 설정 방법

식은 다음과 같다.

$$\begin{aligned}\mathbf{z}^{i+1} &= f(\mathbf{s}^i),\\ \mathbf{s}^i &= \mathbf{W}^i\mathbf{z}^i + \mathbf{b}^i,\end{aligned}$$

$$\begin{bmatrix} s_1^i \\ s_2^i \\ \vdots \\ s_{n_{i+1}}^i \end{bmatrix} = \begin{bmatrix} w_{11}^i & \cdots & w_{1n_i}^i \\ \vdots & \ddots & \vdots \\ w_{n_{i+1}1}^i & \cdots & w_{n_{i+1}n_i}^i \end{bmatrix} \begin{bmatrix} z_1^i \\ z_2^i \\ \vdots \\ z_{n_i}^i \end{bmatrix} + \begin{bmatrix} b_1^i \\ b_2^i \\ \vdots \\ b_{n_{i+1}}^i \end{bmatrix},$$

여기에서 $\mathbf{s}^i$는 n_{i+1}차원을 갖는 입력값 벡터이고 $\mathbf{W}^i$는 차원이 $n_{i+1} \times n_i$인 가중치 행렬이다. 이때 가중치 행렬의 원소 w_{kl}^i에서 k는 $(i+1)$번째 층의 노드 인덱스이고 l은 i번째 층의 노드 인덱스를 의미한다. 순전파 과정에서 위의 활성화 함수식은 k번째 노드에 대해서 다음과 같이 표현할 수 있다.

$$\begin{aligned}{z_k}^{i+1} &= f\left(s_k^i\right)\\ &= f\left(w_{k1}^i z_1^i + \cdots + w_{kn_i}^i z_{n_i}^i + b_k^i\right)\\ &= w_{k1}^i z_1^i + \cdots + w_{kn_i}^i z_{n_i}^i + b_k^i\\ &= w_{k1}^i z_1^i + \cdots + w_{kn_i}^i z_{n_i}^i.\end{aligned}$$

위의 식에서 양변에 분산을 취하면 다음과 같다.

$$\begin{aligned}\mathrm{Var}\left[z_k^{i+1}\right] &= \sum_{l=1}^{n_i} \mathrm{Var}\left[w_{kl}^i z_l^i\right]\\ &= \sum_{l=1}^{n_i} \mathrm{Var}\left[w_{kl}^i\right] \mathrm{Var}\left[z_l^i\right]\\ &= \mathrm{Var}\left[z_1^i\right] n_i \mathrm{Var}\left[w_{k1}^i\right], \end{aligned} \tag{2.1}$$

여기에서 첫 번째와 두 번째 등호는 w와 z의 기댓값이 0이고 서로 독립이라는 가정을 했기 때문에 성립한다. Glorot과 Bengio (2010)는 각 층의 출력값에 대한 분산이 입력값에 대한 분산과 같아야 한다고

주장하였고, 따라서 식 (2.1)에서 $\mathrm{Var}[z_k^{i+1}] = \mathrm{Var}[z_1^i]$이 성립하기 위해서는 $n_i\mathrm{Var}[w_{kl}^i] = 1$이 되어야 한다.

본 연구에서 사용된 비용함수는 $-\log P(y|x)$로 negative log-likelihood 함수이다. 여기에서 P는 우리가 가진 데이터의 분포를 뜻하며, 학습데이터의 분포와 예측한 결과의 분포의 차이를 최소화하는 것을 목표로 한다. 결국 두 확률분포 사이의 차이를 측정하는 크로스 엔트로피 함수가 되며 크로스 엔트로피는 비교 대상 확률 분포의 종류를 특정하지 않는 장점이 있기 때문에 본 논문에서 사용하였다. 이때 x는 입력 이미지이고 y는 정답 클래스를 의미한다. 비용함수를 활성화 함수값으로 편미분해준 $\partial\mathrm{Cost}/\partial s_l^i$을 이용하였고 여기에서 Cost는 비용함수 $-\log P(y|x)$를 의미한다. 이때 연쇄법칙을 이용해 식을 $(\partial z_l^{i+1}/\partial s_l^i)(\partial \mathbf{s}^{i+1}/\partial z_l^{i+1})(\partial\mathrm{Cost}/\partial \mathbf{s}^{i+1})$과 같이 표현할 수 있다. 여기에 앞에서 언급한 가정 (i), (ii)를 적용한다면 $\partial z_l^{i+1}/\partial s_l^i = f'(s_l^i) \approx 1$이 되고 다음 식과 같이 표현할 수 있다.

$$\begin{aligned}\frac{\partial\mathrm{Cost}}{\partial s_l^i} &= \frac{\partial z_l^{i+1}}{\partial s_l^i}\frac{\partial \mathbf{s}^{i+1}}{z_l^{i+1}}\frac{\partial\mathrm{Cost}}{\partial \mathbf{s}^{i+1}} \\ &= \left(W_{\cdot,l}^{i+1}\right)^t \frac{\partial\mathrm{Cost}}{\partial \mathbf{s}^{i+1}},\end{aligned} \tag{2.2}$$

여기에서 $W_{\cdot,l}^{i+1} = [w_{1l}^{i+1}, \ldots, w_{n_{i+2}l}^{i+1}]^t$이다. 이후 식 (2.2)의 양변에 분산을 취하면 다음과 같다.

$$\begin{aligned}\mathrm{Var}\left[\frac{\partial\mathrm{Cost}}{\partial s_l^i}\right] &= \mathrm{Var}\left[\left(W_{\cdot,l}^{i+1}\right)^t \frac{\partial\mathrm{Cost}}{\partial \mathbf{s}^{i+1}}\right] \\ &= \mathrm{Var}\left[w_{1l}^{i+1}\frac{\partial\mathrm{Cost}}{\partial s_1^{i+1}} + \cdots + w_{n_{i+2}l}^{i+1}\frac{\partial\mathrm{Cost}}{\partial s_{n_{i+2}}^{i+1}}\right] \\ &= \mathrm{Var}\left[\sum_{k=1}^{n_{i+2}}\frac{\partial\mathrm{Cost}}{\partial s_k^{i+1}}w_{kl}^{i+1}\right] \\ &= \mathrm{Var}\left[\frac{\partial\mathrm{Cost}}{\partial s_1^{i+1}}\right] n_{i+2}\mathrm{Var}\left(w_{1l}^{i+1}\right).\end{aligned} \tag{2.3}$$

Glorot과 Bengio (2010)는 또한 역전파 과정에서 각 층을 통과하기 전과 후의 그래디언트의 분산이 동일해야한다고 주장하였다. 따라서 식 (2.3)에서 $\mathrm{Var}[\partial\mathrm{Cost}/\partial s_l^i] = \mathrm{Var}[\partial\mathrm{Cost}/\partial s_1^{i+1}]$이 성립하기 위해서는 $n_{i+2}\mathrm{Var}[w_{kl}^{i+1}] = 1$이 되어야 한다. 순전파와 역전파에서 가중치의 분산에 대한 결과는 각각 $\mathrm{Var}[w_{kl}^i] = 1/n_i$와 $\mathrm{Var}[w_{kl}^i] = 1/n_{i+1}$로 표현될 수 있고, 이 두 결과를 조화평균을 이용하여 하나의 식으로 표현하면 $\mathrm{Var}[w_{kl}^i] = 2/(n_i + n_{i+1})$과 같이 나타낼 수 있다.

2.3. 제안하는 방법론

Glorot과 Bengio (2010)는 초기값을 설정할 때 균일분포에서 초기값을 설정하였고 He 등 (2015)은 정규분포에서 초기값을 설정하였다. 초기값을 균일분포가 아닌 정규분포에서 설정할 경우 Figure 2.2와 같이 0부분에 더 몰려있기 때문에 균일분포에 비해 더 작은 초기값이 뽑히게 된다. 결국 활성화 함수 Sigmoid값이 선형한 영역에서 나올 확률이 높아지게 된다. 하지만 정규분포의 양 끝 꼬리 부분은 매우 큰 값을 가진다. 작은 확률이지만 이런 양 끝 구간에서 뽑히게 되면 초기값이 매우 커져 버린다는 단점이 있고 이는 포화 현상을 일으키게 된다.

본 연구에서는 초기값을 설정하는 분포로 절단된 정규분포와 절단된 코쉬 분포를 고려하였다. 코쉬 분포는 정규분포보다 0에 더 몰려있고 양쪽 꼬리가 더 얇기 때문에 Figure 2.3과 같이 양쪽 끝을 절단할 경우 초기치가 0 근처에서 뽑힐 확률이 더 높을 수 있다. 절단된 정규분포와 절단된 코쉬 분포를 각각

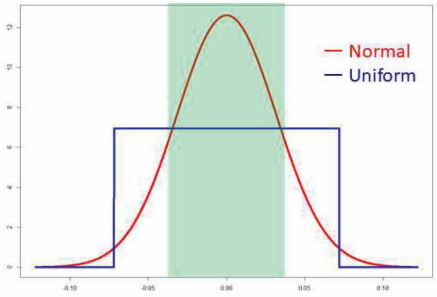

Figure 2.2. Difference between normal distribution and uniform distribution.

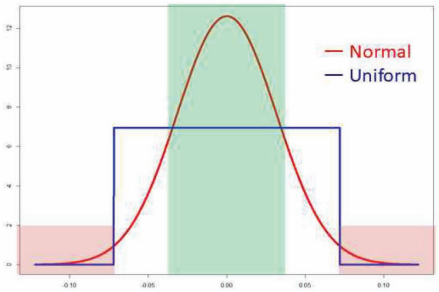

Figure 2.3. Normal distribution and uniform distribution with both ends truncated.

Xavier initialization 방법에 적용 후 초기값을 설정한다면 기존의 한계점을 보완하고 좋은 성능을 보일 것이라 기대한다.

절단된 정규분포의 분산을 Xavier initialization 방법에 적용하는 논리는 다음과 같다. 절단된 정규분포의 분산은 다음과 같이 표현할 수 있다.

$$\sigma^2 \left\{ 1 + \frac{\alpha\phi(\alpha) - \beta\phi(\beta)}{\Phi(\beta) - \Phi(\alpha)} - \left[\frac{\phi(\alpha) - \phi(\beta)}{\Phi(\beta) - \Phi(\alpha)} \right]^2 \right\}, \tag{2.4}$$

여기에서 $\alpha = (a-u)/\sigma$, $\beta = (b-u)/\sigma$로 각각 설정한다. Glorot과 Bengio (2010)의 방법을 적용하기 위해 절단된 정규분포의 분산을 $2/(n_i + n_{i+1})$과 맞춰준다. 여기에서 절단되기 전 정규분포의 분산을 설정해야 하는 문제가 발생한다. 식 (2.4)를 Xavier initialization 방법의 분산과 같게 두고 σ^2에 대하여 정리하면 다음과 같다.

$$\sigma^2 = \frac{2}{(n_i + n_{i+1}) \left\{ 1 + \frac{\alpha\phi(\alpha) - \beta\phi(\beta)}{\Phi(\beta) - \Phi(\alpha)} - \left[\frac{\phi(\alpha) - \phi(\beta)}{\Phi(\beta) - \Phi(\alpha)} \right]^2 \right\}}.$$

최소값(α)과 최대값(β)를 Figure 2.4와 같이 표준편차의 배수로 설정하고 $u = 0$, $a = -k\sigma$, $b = k\sigma$로 설정하게 된다면 절단되기 전 정규분포의 분산은 다음과 같이 표준편차의 배수 k와 레이어의 노드 사이즈에만 의존하게 된다.

$$\sigma^2 = \frac{2}{(n_i + n_{i+1}) \left\{ 1 + \frac{-k\phi(-k) - k\phi(k)}{\Phi(k) - \Phi(-k)} - \left[\frac{\phi(-k) - \phi(k)}{\Phi(k) - \Phi(-k)} \right]^2 \right\}}.$$

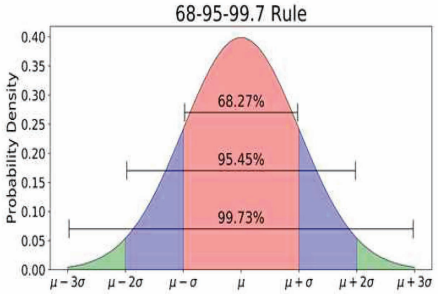

Figure 2.4. Normal distribution was truncated by standard deviation multiple.

절단된 코쉬 분포의 스케일 모수를 Xavier initialization 방법에 적용하는 방법은 다음과 같다. 절단된 코쉬 분포의 스케일 모수는 다음과 같이 표현할 수 있다.

$$\sigma^2\left\{\frac{b+a-\tan^{-1}(b)-\tan^{-1}(a)}{\tan^{-1}(b)+\tan^{-1}(a)}-\left[\frac{\log(1+b^2)-\log(1+a^2)}{\tan^{-1}(b)+\tan^{-1}(a)}\right]^2\right\},$$

여기에서 Y의 범위는 $u-a\sigma<Y<u+b\sigma$로 설정할 수 있다. Glorot과 Bengio (2010)의 논리를 적용하기 위해 절단된 코쉬 분포의 스케일 모수를 $2/(n_i+n_{i+1})$과 맞춰준다. 여기에서 절단되기 전 코쉬 분포의 스케일 모수를 설정해야하는 문제가 발생한다. 절단되기 전 코쉬 분포의 스케일 모수를 설정하기 위해 와 를 같게 설정하면 코쉬 분포의 스케일 모수는 다음과 같이 표현할 수 있다.

$$\sigma^2\left[\frac{a-\tan^{-1}(a)}{\tan^{-1}(a)}\right].$$

코쉬 분포의 스케일 모수를 남기고 식을 이항해서 얻은 코쉬 분포의 스케일 모수의 수식은 다음과 같이 표현할 수 있다.

$$\sigma^2=\frac{\tan^{-1}(a)}{a-\tan^{-1}}\frac{2}{n_i+n_{i+1}}.$$

절단되기 전 코쉬 분포의 스케일 모수는 앞의 식과 같이 a와 레이어의 노드 사이즈에만 의존하게 된다.

3. 실험

3.1. 데이터 설명 및 실험설계

본 연구에서는 MNIST와 CIFAR-10 데이터를 사용하였다. MNIST는 사람이 손으로 쓴 숫자들로 이루어진 대형 데이터베이스이며, 다양한 화상 처리 시스템을 트레이닝 할 때 주로 사용된다. 28×28 크기의 흑백 이미지로 0–9까지 레이블로 구성되어 있고 60,000장의 트레이닝와 10,000장의 테스트 데이터로 이루어져 있다 (LeCun 등, 1998b). CIFAR-10은 기계학습 알고리즘을 학습하는데 일반적으로 사용되는 이미지로 10가지의 클래스로 총 60,000개의 32×32 컬러 이미지로 구성되어 있다 (Krizhevsky와 Hinton, 2009). CIFAR-10은 10가지 클래스로 구분되며 항공기, 자동차, 새, 고양이, 사슴, 개, 개구리, 말, 배, 트럭으로 되어 있다. 또한 60,000개의 이미지가 들어 있으며 50,000장의 트레이닝 데이터와 10,000장의 테스트 테이터로 구성되어 있다.

MNIST 데이터를 DNN을 이용해 실험했을 때 DNN층은 4개로 구성했고 학습 속도는 0.01, 배치 사이즈는 100, 에폭은 200번을 설정한 후 실험을 진행하였다. Xavier initialization 방법을 적용할 때 층

Table 3.1. Result value in DNN according to initialization setting methods

Data type	Model	Method	Max out (%)
MNIST	DNN	Xavier	98.10
		He	98.13
		Truncated Normal	98.21
		Truncated Cauchy	98.14

DNN = deep neural networks.

Table 3.2. Result value in CNN according to initialization setting methods

Data type	Model	Method	Max out (%)
MNIST	CNN	Xavier	64.28
		He	65.51
		Truncated Normal	67.70
		Truncated Cauchy	66.08

CNN = convolutional neural networks.

별 분산을 동일하게 해주기 위해서는 이전 레이어의 노드와 다음 레이어의 노드의 개수를 넣어줘야 한다. CNN에서 이전 레이어의 노드의 개수에 해당하는 값은 Shape×RGB이고 다음 레이어의 노드의 개수에 해당하는 값은 Filter의 개수이다. CIFAR-10 데이터를 CNN을 이용해 실험을 진행할 때 CNN층은 5개, 학습 속도는 0.001, 풀링 레이어는 2개, 커널 사이즈는 3×3, 에폭은 200회 설정한 후 실험을 진행하였다. 이 실험에서는 Xavier initialization 방법과 He initialization 방법과 절단된 분포를 이용한 초기값 설정 방법을 이용해 각각 초기값을 설정한 후 각각의 모델을 학습시키고 분류의 정확도를 구하였다. 본 논문에서는 분류 정확도를 Max out으로 구하였는데 Max out은 실험 결과 중 가장 높은 정확도 값을 의미한다. 초기값에 대한 선행 연구 중 Mishkin과 Matas (2015)은 초기값에 대한 실험 결과를 나타낼 때 Max out을 이용해 실험 결과를 보였다.

3.2. 실험 결과

MNIST 데이터를 DNN을 이용해 분류하였고 그 결과는 Table 3.1과 같다. 활성화 함수는 모두 Relu를 사용하였다. 분석결과 MNIST를 학습시킨 DNN에서는 Xavier initialization 방법은 98.10%의 정확도를 보였고 He initialization 방법은 98.13%의 정확도를 보였다. 절단된 정규분포와 절단된 코쉬 분포를 사용했을 때 실험 결과는 각각 98.21%, 98.14%으로 기존 방법보다 모두 높은 정확도를 보였다. 특히 절단된 분포 중 절단된 정규분포를 이용했을 때 가장 높은 정확도를 보였다. 본 연구에서는 절단된 정규분포에서는 표준편차의 배수 값(k)과 절단된 코쉬 분포에서는 스케일 값(a)에 여러 값들을 넣어서 실험을 진행하였고 그 결과 k를 2, a를 1로 설정할 때 정확도가 가장 높았다.

다음 Table 3.2는 CIFAR-10데이터를 CNN을 이용해 분류한 결과이다. 분석결과 CIFAR-10데이터를 학습시킨 CNN에서는 Xavier initialization 방법은 64.28%의 정확도를 보였고 He initialization 방법은 65.51%의 정확도를 보였다. 절단된 정규분포와 절단된 코쉬 분포를 사용했을 때 실험 결과는 각각 67.7%, 66.08%으로 기존 방법보다 모두 높은 정확도를 보였다. 그 중에서도 절단된 정규분포를 이용했을 때 가장 높은 정확도를 보였다. 앞의 실험과 마찬가지로 k와 a에 여러 가지 값을 넣어서 실험을 진행하였고 이때 절단된 정규분포의 k는 2, 절단된 코쉬 분포의 a는 1로 설정할 때 정확도가 가장 높았다. DNN과 CNN에서 모두 본 연구에서 제안한 초기값 설정 방법이 기존 방법보다 성능이 좋았고 특히 절단된 정규분포를 이용했을 때 성능이 가장 좋았다.

4. 결론

본 연구에서는 기존의 초기값 설정 방법에 대해 살펴보고 기존의 방법을 대체할 수 있는 새로운 초기값 설정 방법을 제시하였다. 기존 방법으로 많이 쓰이는 Xavier initialization 방법과 He initialization 방법을 본 논문에서 제시한 방법과 비교하였는데 DNN과 CNN을 활용한 분류 문제에서 기존 방법보다 좋은 성능을 보여줬다. 초기값 설정할 때 문제가 되었던 포화 현상은 본 연구에서 제안한 방법에서도 발생하지 않았으며 수렴속도 또한 기존 방법보다 빠르게 수렴함을 보여줬다. 또한 기존에 절단된 분포에서 초기값을 설정하는 경우는 많았지만 절단된 분포의 자르는 위치를 제안한 방법은 없었고 본 연구는 자르는 위치를 제안함과 동시에 기존 Xavier 방법의 방법론에 적용시켜 모델의 정확도를 향상시켰다. 본 연구에서 제안한 방법은 연구자가 초기값에 대한 하이퍼 파라미터를 설정할 때 기존에 방법과 함께 고려해 볼 수 있는 방법으로 다양한 딥러닝 모델에서 포화 현상이나 실험 결과가 좋지 않을 때 시도해 볼 수 있는 초기값 선택 방법으로 활용될 수 있을 것이라고 기대된다.

추후에는 정규분포와 비슷한 분포인 Triangle distribution, Epanechnikov distribution, Triweight distribution같은 다양한 분포들의 양 끝이 절단된 분포를 이용해 초기값을 설정해 보는 연구와 함께 절단된 분포에서 성능이 좋은 활성화 함수를 연구할 것이다. 또한 본 연구에서 제안한 방법을 이용하여 다양한 데이터들을 활용해 다양한 기계학습 알고리즘들에 적용시켜 성능을 비교해보는 후속 연구를 진행할 것이다.

References

Clevert, D. A., Unterthiner, T., and Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (pp. 249–256).

Goodfellow, I. J., Vinyals, O., and Saxe, A. M. (2014). Qualitatively characterizing neural network optimization problems. arXiv preprint arXiv:1412.6544.

Hanin, B. and Rolnick, D. (2018). How to start training: The effect of initialization and architecture. In *Advances in Neural Information Processing Systems* (pp. 571–581).

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1026–1034).

Humbird, K. D., Peterson, J. L., and McClarren, R. G. (2018). Deep neural network initialization with decision trees, *EEE Transactions on Neural Networks and Learning Systems*, **30**, 1286–1295.

Hayou, S., Doucet, A., and Rousseau, J. (2018). On the selection of initialization and activation function for deep neural networks. arXiv preprint arXiv:1805.08266.

Krahenbuhl, P., Doersch, C., Donahue, J., and Darrell, T. (2015). Data-dependent initializations of convolutional neural networks. arXiv preprint arXiv:1511.06856.

Krizhevsky, A. and Hinton, G. (2009). *Learning Multiple Layers of Features from Tiny Images* (Vol. 1, No. 4, p. 7) Technical report, University of Toronto.

LeCun, Y., Bottou, L., Orr, G., and Muller, K. (1998a). Efficient backprop in neural networks: Tricks of the trade (Orr, G. and Muller, K., eds.), *Lecture Notes in Computer Science*, **1524(98)**, 111.

LeCun, Y., Cortes, C., and Burges, C. J. (1998b). The MNIST Database of Handwritten Digits.

Mishkin, D. and Matas, J. (2015). All you need is a good init. arXiv preprint arXiv:1511.06422.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013). On the importance of initialization and momentum in deep learning, In *International Conference on Machine Learning* (pp. 1139–1147).

절단된 분포를 이용한 인공신경망에서의 초기값 설정방법

김민종[a] · 조성철[a] · 정혜린[a] · 이영섭[b] · 임창원[a,1]

[a]중앙대학교 응용통계학과, [b]동국대학교 통계학과

(2019년 6월 24일 접수, 2019년 8월 9일 수정, 2019년 8월 20일 채택)

요 약

딥러닝은 대용량의 데이터의 분류 및 예측하는 방법으로 각광받고 있다. 데이터의 양이 많아지면서 신경망의 구조는 더 깊어 지고 있다. 이때 초기값이 지나치게 클 경우 층이 깊어 질수록 활성화 함수의 기울기가 매우 작아지는 포화(Saturation)현상이 발생한다. 이러한 포화현상은 가중치의 학습능력을 저하시키는 현상을 발생시키기 때문에 초기값의 중요성이 커지고 있다.이런 포화현상 문제를 해결하기 위해 Glorot과 Bengio (2010)과 He 등 (2015) 층과 층 사이에 데이터가 다양하게 흘러야 효율적인 신경망학습이 가능하고 주장했다. 데이터가 다양하게 흐르기 위해서는 각 층의 출력에 대한 분산과 입력에 대한 분산이 동일해야 한다고 제안했다. Glorot과 Bengio (2010)과 He 등 (2015)는 각 층별 활성화 값의 분산이 같다고 가정해 초기값을 설정하였다. 본 논문에서는 절단된 코쉬 분포와 절단된 정규분포를 활용하여 초기값을 설정하는 방안을 제안한다. 출력에 대한 분산과 입력에 대한 분산의 값을 동일하게 맞춰주고 그 값이 절단된 확률분포의 분산과 같게 적용함으로써 큰 초기값이 나오는 걸 제한하고 0에 가까운 값이 나오도록 분포를 조정하였다. 제안된 방법은 MNIST 데이터와 CIFAR-10 데이터를 DNN과 CNN 모델에 각각 적용하여 실험함으로써 기존의 초기값 설정방법보다 모델의 성능을 좋게 한다는 것을 보였다

주요용어: 초기값, 포화, Xavier, 절단된 분포, 딥러닝

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단-차세대정보·컴퓨팅기술개발사업의 지원을 받아 수행된 연구임 (NRF-2017M3C4A7083281).

[1]교신저자: (06974) 서울특별시 동작구 흑석로 84, 중앙대학교 응용통계학과. E-mail: clim@cau.ac.kr