딥 러닝에서의 딥 뉴럴 네트워크의 가중치 초기화 방법

홍정하, 여도엽
한국전자통신연구원

jhong@etri.re.kr, yeody@etri.re.kr

Weight initialization method for deep neural network in deep learning

Jungha Hong, Doyeob Yeo
Electronics and Telecommunications Research Institute

요 약

본 논문은 딥 러닝에서의 딥 뉴럴 네트워크의 효율적 학습을 위한 초기 가중치 설정에 관한 것으로, 비지도 학습을 이용하여 학습 데이터의 특징을 추출하는 딥 뉴럴 네트워크를 먼저 학습시킨 후, 학습된 딥 뉴럴 네트워크의 변수들을 지도 학습의 딥 뉴럴 네트워크 초기값으로 이용하는 방법을 제안한다.

Ⅰ. 서 론

딥 러닝에서는 손실함수(loss function 또는 cost function)가 가장 작은 값을 갖도록 딥 뉴럴 네트워크의 가중치(weight)를 학습시키는데, 주로 임의의 초기 가중치를 사용한다. 그런데 같은 구조의 딥 러닝을 학습시켜도, 가중치의 초기값에 따라 손실함수 값이 발산하는 경우도 발생하고, 수렴하더라도 수렴된 손실함수의 값이 다르게 나타날 수 있다. 또한 초기 가중치 설정에 따라 기울기 소실(gradient vanishing), 표현력(representation)의 한계 등 딥 러닝 학습에서 여러 문제를 야기할 수도 있다. 따라서, 딥 러닝 학습에서 초기 가중치 설정은 매우 중요한 역할을 한다.

딥 러닝 학습에서의 손실함수는 비볼록(non-convex)이고 많은 극소점(local minimum)을 갖기 때문에, 초기 가중치를 잘못 설정할 경우 최소점(global minimum)이 아닌 local minimum 으로 수렴할 가능성이 커지게 된다. 특히, 활성화 함수(activation function)가 시그모이드(sigmoid) 또는 정류 선형 유닛(ReLU)인 경우, 가중치 초기값의 절대값이 커지면 그래디언트(gradient) 소실(vanishing) 또는 폭주(exploding)가 일어나게 된다. 따라서, 딥 러닝에서 딥 뉴럴 네트워크가 안정적으로 수렴하기 위해서는 가중치 초기값을 작게 설정해야 하며 동일한 초기값을 갖지 않도록 랜덤하게 설정해야 한다. 가중치 초기값을 설정하는 방법은 다양하게 연구되어 왔으며, 주로 사용하는 방법으로는 Lecun initialization[1], Xavier initialization[2], He initialization[3] 등이 있다.

한편, 많은 데이터를 이용하여 비교적 층수가 얕은 네트워크를 학습시키는 경우에는 모델의 수용력(capacity)이 작기 때문에 과소적합(underfitting) 되는 경향이 있다.[4] 그러나, 네트워크의 층수가 깊어질수록 추론(inference) 시 계산량이 증가하여 계산 시간이 늘어나기 때문에 빠른 추론이 필요한 경우에는 네트워크의 층수를 줄이는 것이 중요하다. 즉, 많은 데이터를 이용하는 경우에도 층수가 얕은 네트워크를 이용하여 학습이 잘 되도록 할 필요가 있다.

따라서, 본 논문에서는 초기 가중치 값을 랜덤 하게 주지 않고, 입력 데이터의 특징이 잘 반영된 값으로 주는 방법을 제안한다. 제안한 가중치 초기화 방법을 사용하면 많은 데이터를 이용하여 학습시키는 경우에도 층수가 얕은 네트워크를 이용하여 학습이 잘 될 수 있도록 하는 것이 가능하게 된다.

Ⅱ. 본론

본 논문에서는 생산적 적대 신경망(Generative Adversarial Network, GAN)과 유사한 방식의 비지도 학습을 이용하여 지도 학습 기반의 딥 뉴럴 네트워크를 보다 안정적으로 학습시키기 위한 초기 가중치 설정 방법으로 학습 데이터의 특징을 추출하는 딥 뉴럴 네트워크를 먼저 학습시킨 다음, 학습된 딥 뉴럴 네트워크의 가중치 값들을 지도 학습 기반의 딥 뉴럴 네트워크의 초기 가중치 값으로 설정하는 방법을 제안한다.

일반적으로 지도 학습 기반의 딥 뉴럴 네트워크 기술에서 특징을 추출하는 네트워크는 그림 1 과 같이 층이 깊어질수록 네트워크의 폭이 점점 줄어드는 방향으로 설계가 된다. 한 예로, 가장 기본적인 딥 뉴럴 네트워크 구조인 다중 퍼셉트론(Multi-layer Perceptron, MLP)를 비롯하여 영상 데이터셋 학습을 위하여 많이 사용되는 합성곱 신경망(Convolutional Neural Network, CNN) 등에서의 네트워크는 대부분 폭이 점점 줄어드는 방향으로 설계가 되어 있다. 그림 1 에서 실선으로 나타낸 부분은 실제로 연산이 이루어지는 딥 뉴럴 네트워크이며, 점선으로 나타낸 부분은 입력 또는 결과 텐서들을 의미한다. 그림 2 는 일반적인 GAN 의 구조를 나타내고 있다.

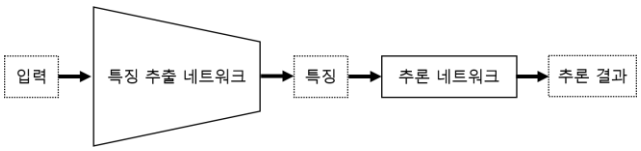

그림 1 지도 학습 기반의 딥 뉴럴 네트워크 구조

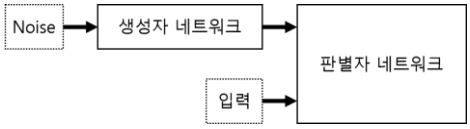

그림 2 Generative Adversarial Network (GAN) 구조

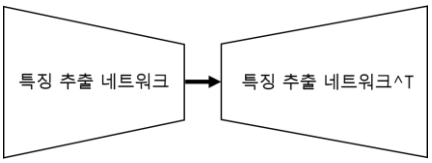

그림 3 생성자 네트워크 구조

본 논문에서는 주어진 데이터셋에 적합한 특징 추출 네트워크의 가중치 초기값을 얻기 위하여 그림 2 에서의 생성자 네트워크는 그림 3 과 같이 구성하고, 판별자 네트워크는 데이터셋에 따라 사용자가 자유롭게 구성하게 한다. 한 예로, 데이터셋이 영상 자료로 구성되어 있을 경우에는 판별자 네트워크를 CNN 으로 구성할 수 있으며, 일반적인 테이블 자료나 벡터로 구성되어 있을 경우에는 MLP 로 구성할 수 있다. 또한 그림 2 에서의 잡음(noise) 대신에 입력 값으로 주어진 데이터셋을 이용한다. 따라서 주어진 데이터셋에 적합한 특징 추출 네트워크의 가중치 초기값을 얻기 위한 장치는 그림 4 와 같이 구성된다. 그림 4 는 그림 2 의 GAN 구조와 유사하지만, GAN 은 생성자 네트워크의 입력으로 noise 를 주로 이용하는데 반해, 본 논문에서는 그림 4 와 같이 입력 데이터를 생성자 네트워크의 입력으로 이용한다는 차이가 있다.

그림 3 에서의 "특징 추출 네트워크"는 그림 1 에 나타낸 지도 학습 기반의 딥 뉴럴 네트워크에서 사용될 특징 추출 네트워크를 의미한다. "특징 추출 네트워크^T"는 특징 추출 네트워크에서의 계산 흐름과 반대로 이루어지는 네트워크이다. 생성자 네트워크를 통해 만들어지는 결과와 입력 데이터의 차원을 맞추기 위하여 그림 3 과 같이 생성자 네트워크를 구성하는 것이다.

그림 4 에서 판별자 네트워크는 입력 데이터로부터 생성자 네트워크를 통하여 나온 결과값인지, 아니면 원래의 입력 데이터인지를 판단하는 역할을 한다. 생성자 네트워크가 주어진 입력 데이터셋의 분포를 잘 학습하기 위하여, GAN 에서 제안된 손실 함수를 사용한다. 생성자 네트워크는 입력과 비슷한 데이터를 생성하도록 학습이 진행되고, 판별자 네트워크는 생성자 네트워크에서 만들어진 데이터인지, 아니면 원래의 입력 데이터인지를 잘 구별해내도록 학습이 진행된다. 생성자 네트워크와 판별자 네트워크 간의 적대적 관계를 통하여 결과적으로 생성자 네트워크는 원래의 입력 데이터들의 분포를 잘 학습할 수 있는 방향으로 학습이 진행된다. 또한, 생성자 네트워크를 구성하고 있는 특징 추출 네트워크에서는 입력 데이터의 특징을 효과적으로 추출해낼 수 있는 방향으로 학습이 진행된다.

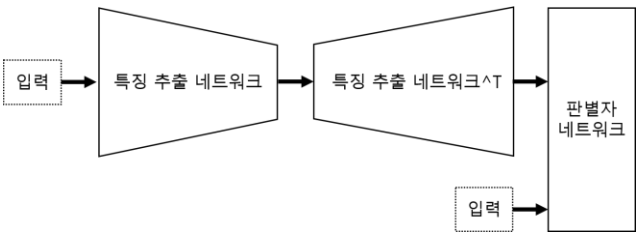

그림 4 딥 러닝에서의 딥 뉴럴 네트워크의 가중치 초기화 장치

따라서, 그림 4 의 장치를 통하여 가중치가 최적화된 특징 추출 네트워크를 그림 1 의 특징 추출 네트워크 초기값으로 설정하여 추론 모델을 학습시키면 층수가 얕은 네트워크를 이용하더라도 학습이 잘 될 수 있다.

Ⅲ. 결론

본 논문에서는 GAN 과 유사한 방식의 비지도 학습을 통하여 학습 데이터의 특징을 먼저 추출한 다음, 학습된 딥 뉴럴 네트워크의 가중치들을 초기값으로 설정하여 지도 학습에 이용하는 방법을 제안하였다.

제안한 가중치 초기화 방법을 사용하면 학습 데이터셋의 분포 및 특징을 먼저 학습한 다음 이를 가중치 초기값으로 설정하기 때문에, 층수가 얕은 네트워크를 이용하더라도 손실 함수가 발산하지 않고 수렴하는 효과가 있다.

딥 러닝에서 딥 뉴럴 네트워크의 층이 깊어지면 가중치들의 작은 변화가 출력 값의 큰 변화로 이어지는 불안정한 현상들이 생기기 때문에 처음부터 최적의 초기 가중치 값들을 사용한다면 출력 값들의 분포를 안정화시킬 수 있으며 학습 횟수가 많지 않더라도 성능이 좋은 모델을 만들 수 있는 효과가 있다.

ACKNOWLEDGMENT

이 논문은 2020 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2018-0-00841, IoT 디바이스를 위한 Lightweight 블록체인 표준개발).

참 고 문 헌

[1] LeCun, Y., et al. "Efficient BackProp," Neural networks: Tricks of the trade, pp. 9-50, Jan. 1998.

[2] Glorot, X. and Bengio, Y. "Understanding the difficulty of training deep feedforward neural networks," Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp. 249-256, 2010.

[3] He, K., et al. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," Proceedings of the IEEE international conference on computer vision, pp. 1026-1034, 2015.

[4] Goodfellow, I., et al. "Deep learning," The MIT press, Cambridge, 2016.