ReLU 함수의 예측을 통한 인공 신경망 추론 연산 최적화

박상우, 김한이, 서태원
고려대학교 컴퓨터학과
e-mail : psw0113@korea.ac.kr

Optimization of Artificial Neural Network Inference by ReLU Function Prediction

Sangwoo Park, Hanyee Kim, Taeweon Suh
Dept. of Computer Science, Korea University

요 약

본 연구는 인공 신경망 '추론'과정에서 연산량을 줄이는 아이디어를 고안했고, 이를 구현하여 기존 알고리즘과 성능을 비교 분석하였다. 특정 데이터 셋에 대한 실험을 통해 ReLU (Rectified Linear Unit) 함수의 결과를 분석했고, 그 결과를 통해 ReLU 함수의 결과가 예측가능함을 확인했다. 또한 인공 신경망 알고리즘에 ReLU 함수의 결과 예측 기법을 적용하여 인공 신경망 추론과정을 최적화했다. 이 아이디어를 기반으로 구현된 인공 신경망은 기존 아이디어로 구현된 인공 신경망에 비해 약 3 배 빠른 성능을 보였다.

1. 서론

뉴럴 네트워크는 다양한 어플리케이션 도메인에서 좋은 성능을 보여주고 있으며 그 중에서도 특히 음성, 이미지, 텍스트의 패턴 인식과 분류에 특화되어있다 [1-3]. 뉴럴 네트워크는 인간의 신경망을 모방하여 설계되었으며, 다양한 트레이닝 데이터를 통해 학습하는 것이 가능한 알고리즘이다. 이러한 뉴럴 네트워크는 트레이닝 데이터가 많아지거나 모델의 파라미터 숫자가 증가할수록 분류 정확도에 있어서 좋은 성능을 보여 준다[4,5]. 그리고 이렇게 깊은 레이어 층을 구성하여 많은 파라미터를 가지는 뉴럴 네트워크를 '딥 뉴럴 네트워크'라고 부른다.

하지만 딥 뉴럴 네트워크는 모델의 크기가 커질 수록 많은 파라미터를 가지게 되고, 그만큼 많은 연산량을 가지기 때문에 처리시간이 오래 걸린다는 단점이 존재한다. 따라서 딥 뉴럴 네트워크에서는 속도가 중요한 요소였으며, 그에 따라 인공 신경망을 가속화하는 다양한 연구가 진행되었다. 그 중에서도 네트워크 프루닝은 인공 신경망의 파라미터들 중 0 에 가까운 값들을 걸러내어 효율적으로 연산량을 줄이는 알고리즘이다. 본 연구에서는 기존과는 다르게 활성화 함수의 결과를 예측하여 인공 신경망의 연산량을 줄이는 아이디어를 제시한다.

이 논문은 2018 년도 정부의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임 (NRF-2017R1D1A1B03028926)

다음 섹션에서는 인공 신경망에 대해서 간단히 설명한 후 실험결과를 통해 활성화 함수인 ReLU 의 특성에 대해 분석한다.

2. 본론

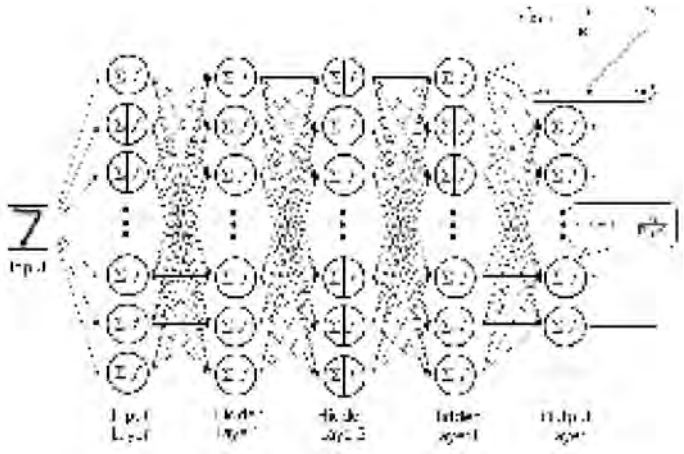
2.1 인공 신경망

인공 신경망의 기본적인 구조는 Input 레이어, Hidden 레이어, Output 레이어로 구성되며 각 레이어들은 특정 개수의 뉴런들이 존재한다. 또한 각 뉴런들은 활성화 함수를 가지는데 비선형 활성화 함수들이 복잡한 문제를 해결하는데 적합하여 활성화 함수에는 주로 비선형 함수들이 사용되고 있다.

인공 신경망은 FP (Forward Phase), BP (Backward Phase), 가중치 갱신의 3 가지 작업을 순차적으로 진행하여 학습한다. FP 에서는 입력을 받아 레이어의 각 뉴런들의 가중치들과 곱셈 연산을 한 후, 그 결과들을 모두 더하고 활성화 함수를 거쳐 다음 레이어로 전파한다. 이 과정을 반복하여 Output 레이어까지 도달하면 FP 가 끝난다. BP 는 경사 하강법을 통해 모든 가중치들의 오차를 계산한다. 그리고 이 계산된 오차를 통해 가중치들을 갱신하여 인공신경망이 학습된다.

본 논문에서는 5 개의 Fully connected 레이어를 가지는 깊은 인공신경망을 구성하여 MNIST 데이터셋 (손글씨, 패션)을 분류하였다. 각 레이어들은 512 개의 뉴런으로 구성되고 마지막 레이어만 10 개의 뉴런으로 구성된다. 활성화 함수로는 ReLU(Rectified

Linear Unit)를 사용했으며 Output 레이어에서 Softmax 함수를 거쳐 최종 결과를 계산한다. (그림 1)은 본 논문에서 실험한 인공 신경망구조를 보여준다.



(그림 1) 인공 신경망 구조

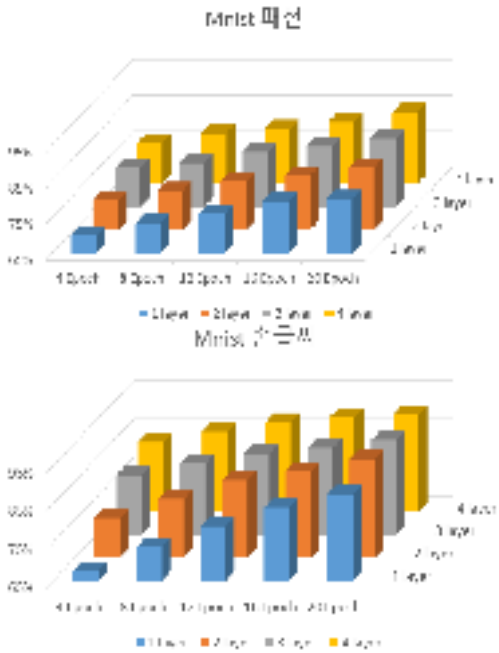
2.2 ReLU 함수 결과의 예측가능성

ReLU 함수는 인공 신경망에서 사용되는 활성화 함수 중 하나로, 입력 값이 음수면 가중치 0 과 곱해져 다음 레이어에 입력 값이 전파되지 않게 만들고, 뉴런의 입력 값이 양수라면 가중치 1 과 곱해져 그 값을 그대로 다음 레이어에 전파되도록 만드는 함수이다. 이러한 ReLU 함수는 다른 활성화 함수에 비해 간단한 구조를 가지기 때문에 빠른 속도를 가져 많은 인공 신경망 모델에서 사용되고있다.

본 논문에서는 (그림 1)의 구조로 구현된 인공 신경망으로 MNIST 손글씨, MNIST 패션 데이터 셋을 각각 20 번 반복 학습시켜 뉴런들의 ReLU 함수의 결과를 분석하였다. 학습이 진행됨에 따라 모델의 정확도는 (표 1)과 같았으며, 과거의 활성화 결과와 미래의 활성화 결과가 같을 확률은 (그림 2)와 같았다. (그림 2)를 보면 두개의 데이터 셋 모두 학습이 진행될수록 과거의 활성화 결과가 미래의 활성화 결과와 같을 확률이 높아지는 것을 볼 수 있다. 하지만 MNIST 손글씨 데이터 셋은 레이어가 깊어질수록 그 확률이 높아졌고, MNIST 패션 데이터 셋은 모든 레이어에서 비슷한 확률을 보였다. 위 실험 결과를 통해 데이터 셋에 따라 조금씩 편차를 보였지만, 과거의 활성화 함수 결과로 미래의 활성화 함수 결과를 높은 확률로 예측 가능함을 확인하였다.

Epoch	4	8	12	16	20
MNIST fashion	85.98 %	87.24 %	88.57 %	89.27 %	89.98 %
MNIST handwritten	94.96 %	95.47 %	96.85 %	97.12 %	98.41 %

<표 1> 학습에 따른 정확도 변화



(그림 2) 뉴런들의 활성화 결과 예측률

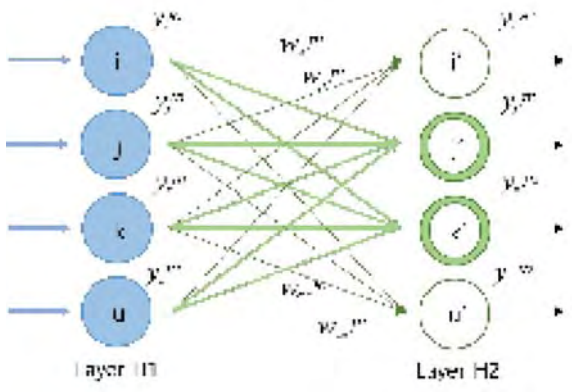
2.3 ReLU 함수의 예측을 통한 인공 신경망 최적화

이번 섹션에서는 과거의 ReLU 함수 결과로 미래의 활성화 함수 결과를 높은 확률로 예측할 수 있다는 점을 이용하여 인공 신경망의 추론 단계에서 계산량을 획기적으로 줄이는 아이디어를 제시한다.

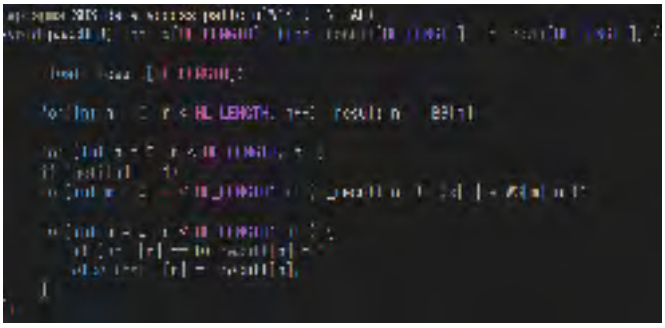
(그림 3)은 레이어 간 뉴런들의 연결 상태와 활성화 함수의 결과를 표현하였다 (i, j, k, u, i', j', k', u' =뉴런의 인덱스, h1, h2=레이어의 인덱스). 그림과 같이 초록색으로 칠해진 y_j^{H2}, y_k^{H2} 뉴런은 과거에 활성화 되었다는 것을 표현하고, 색이 칠해지지 않은 y_i^{H2}, y_u^{H2} 뉴런은 과거에 활성화 되지 않았다는 것을 표현한다. 2.2 섹션에서 다룬 것처럼 ReLU 함수는 그 특성상 특정 뉴런이 활성화 되지 않으면 그 다음 레이어에 전파되는 값은 0 이되어 어떤 값도 전파하지 않는다. 활성화 되지 않는 뉴런들은 복잡한 연산을 하더라도 최종적인 결과가 결국 0 이되기 때문에 이를 사전에 예측할 수만 있다면 이 뉴런들에 연결된 수많은 가중치들을 계산하지 않아도 된다. 과거에 활성화 되었던 뉴런들이 미래에도 활성화 될 확률이 높고, 활성화되지 않았던 뉴런들이 미래에도 활성화 되지 않을 확률이 높다는 것을 보장할수 있다면, (그림 3)의 y_i^{H2} 와 같이 과거에 활성화 되지 않았던 뉴런에 연결된 w_{ii}^{H1}, w_{ji}^{H1}, w_{ki}^{H1}, w_{ii}^{H1} 가중치들을 추론 단계의 계산에서 제외시킬 수 있으므로 추론 단계의 총 연산량을 줄일 수 있다.

(그림 4)는 위 아이디어를 반영하여 추론 단계를 코드로 구현한것으로 x 배열과 acti 배열을 입력으로 받는다. x 는 입력, acti 는 뉴런들의 과거 활성화 함수 결과가 저장 되어있다. for 문으로 뉴런들의 가중치를 계산하기전에 과거에 해당 뉴런이 활성화되었는지 확인하고, 활성화되지 않았다면 그 뉴런들은 계산하지않고 건너뛰도록 코드를 구현하였다((그림 4)의 if(acti[n])부분).

위에서 구현된 코드는 활성화 함수 결과 예측률이 알고리즘 성능에 크게 영향을 미치므로, 활성화 함수 예측률이 높은 3,4 번째 레이어에만(그림 2) 적용시켰다.



(그림 3) 인공 신경망 뉴런들의 활성화 상태



(그림 4) ReLU 함수 예측을 기반으로 최적화된 코드

2.4 실험 환경 및 결과분석

우리는 인공 신경망의 가중치를 학습시키기 위해 MNIST 손 글씨 데이터 셋을 데스크톱 PC 환경에서 학습시켰다. 학습이 끝난 인공 신경망의 가중치들은 98.41%의 정확도를 가졌고, 이를 (그림 1)과 같이 5 개의 레이어를 가지는 인공 신경망 추론 모델을 구현하였다.

우리는 구현된 인공 신경망의 성능을 검증하기 위해 데스크톱 PC (intel core i7-7700cpu 3.60GHz, 16GB ram)에서 MNIST 손 글씨 테스트 데이터셋 10,000 장을 추론하는데 걸리는 소요시간을 측정하였다. (표 2)는 기존 알고리즘과 본 논문에서 제안한 알고리즘과 비교한 결과이다. 본 논문에서 제안된 알고리즘을 적용했을 때 정확도가 1.1%감소하지만, 연산 속도는 성능이 3 배 향상되었다.

5 개의 레이어 중 단 2 개의 레이어에만 본 논문의 아이디어를 적용시켰는데, 전체 연산 속도가 기존에 비해 3 배가 빨라졌다. 그 이유는 활성화 되지 않은 뉴런들의 비율 때문이다. (표 3)에서 볼 수 있듯이 깊은 레이어에서는 90%의 뉴런들이 활성화 되지 않았고, 제안된 알고리즘에 따라 활성화되지 않은 90%의 뉴런들이 연산에서 제외되기 때문에 알고리즘 성능이 극대화 되었던 것이다.

서론에서 언급했듯이 인공 신경망은 깊고 커질수록 좋은 성능을 보이는데[1-3], 인공 신경망이 커지는 만큼 활성화되지 않는 뉴런들도 많아지게된다. 이렇게 활성화 되지 않는 뉴런들이 많은 인공 신경망일수록 본 논문에서 제안한 알고리즘 성능이 극대화되기 때문에, 깊은 인공 신경망일수록 본 논문에서 제안한 알고리즘의 효율성이 증가한다는것을 알 수 있었다.

인공 신경망 모델	정확도	속도
기존의 인공 신경망 모델	98.41 %	45.73 s
제안된 아이디어가 적용된 모델	97.4 %	15.26 s

<표 2> 알고리즘에 따른 결과 차이

	1 레이어	2 레이어	3 레이어	4 레이어
뉴런들이 활성화 되는 비율	67.31 %	44.73 %	15.78 %	10.21 %

<표 3> 레이어에 따른 뉴런들의 활성화되는 비율

3. 결론

본 논문에서는 인공 신경망 추론 단계의 연산량을 줄이는 아이디어를 제안하였고, 이를 구현하여 그 결과를 비교하였다. 우리가 제안한 아이디어는 ReLU 함수의 활성화 확률을 얼마나 정확하게 예측하는지가 관건이었으며, 그 정도에 따라 정확도에 차이가 나는 것을 알 수 있었다. 본 논문의 실험 결과는 인공 신경망의 추론단계에서 일반적인 알고리즘을 사용하는 것 보다 제안된 알고리즘을 사용할 때 더 성능이 좋다는 것을 보였다. 또한 이 알고리즘은 크기가 큰 인공 신경망일수록 성능이 극대화 된다는 것을 보였다.

참고문헌

[1] G. Dahl, D. Yu, L. Deng, and *A. Acero. Context-dependent pre-trained deep neural networks for large vocabulary speech recognition*. IEEE Transactions on Audio, Speech, and Language Processing, 2012.

[2] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber. *Deep big simple neural nets excel on handwritten digit recognition*. CoRR, 2010.

[3] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. *A neural probabilistic language model*. Journal of Machine Learning Research, 3:1137–1155, 2003.

[4] A. Coates, H. Lee, and A. Y. Ng. *An analysis of single-layer networks in unsupervised feature learning*. In AISTATS 14, 2011.

[5] Q.V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A.Y. Ng. *On optimization methods for deep learning*. In ICML, 2011.