PSO 알고리즘을 이용한 분산 딥 러닝 시스템

조인령*, 김현정**, 유상현**, 원일용*
*서울호서전문학교 사이버해킹보안과
**건국대학교 상허교양대학 소속 초빙교수
e-mail: seeker621@naver.com, nygirl@konkuk.ac.kr, yoosh22@gmail.com, clccclcc@hoseo.ac.kr

A Distribute Deep Learning System Using PSO Algorithm

In-Ryeong Jo *, Hyun-jung Kim**, Sang-hyun Yoo**, il-young Won*
*Dept. of Cyber Hack Security, Seoul Hoseo Technical College
**Dept. of Sang-huh college, Konkuk University

요 약

딥 러닝은 하드웨어의 발전과 데이터 양의 비약적 증가에 힘입어 여러 분야에서 좋은 결과를 보여 주고 있다. 본 연구는 딥 러닝의 많은 시간을 소모하는 학습단계에서 고가의 하드웨어가 아닌 저 사양의 장비를 여러 대 결합한 분산 러닝 시스템에 대한 것이다. 분산 학습 알고리즘의 핵심은 PSO 를 응용한 구조이며, 제안한 시스템의 성능은 실험으로 검증하였다.

1. 서론

딥 러닝은 인간 뇌의 정보처리 과정을 수학적인 모델링을 통해 모사한 모형으로 빅데이터의 시대를 맞이한 최근 다양한 분야에서 연구 및 응용되고 있다. 딥 러닝이 최근 좋은 성과를 내고 있는 이유는 알고리즘의 개선에도 원인이 있지만, 하드웨어의 비약적 발전이 무엇보다도 큰 요인이다[1][2]. 빅데이터를 학습하고 강력한 알고리즘 Layer 들을 적용하기 위해서는 강력한 컴퓨터 자원을 필요로 한다. 특히 값비싼 GPU 들을 요구하는 등의 여전히 좋은 연산 능력을 갖춘 장비가 필요하다[3].

이러한 이유로 오랜 시간과 강력한 하드웨어의 연산 능력을 필요로 하는 딥 러닝에서는 일반적으로 사용하는 PC 수준으로 학습을 하기에는 역부족이나, 만약 대다수를 이루고 있는 PC 들만으로도 학습을 할 수 있다면, 자원 효율성의 증대와 딥 러닝을 대중들이 쉽게 접해 볼 수 있는 기회를 제공할 수 있다.

따라서 본 논문은 일반적인 PC 들만으로 구성된 분산시스템 환경과 중앙에 저장소를 둔 분산 딥 러닝 시스템을 제안한다. 제안 시스템은 저 사양의 하드웨어만으로 분산 처리를 적용함으로써 가치 있는 학습 결과를 만드는 것을 목적으로 한다. 시스템 전체의 분산 학습 알고리즘은 PSO 알고리즘을 응용 및 적용하여 자원의 효율성 확보를 추구한다[4][5].

본 논문의 구성은 다음과 같다. 2 장에서는 딥 러닝의 개요와 PSO 알고리즘관련 연구를 논하고, 3 장에서는 제안하는 분산 딥 러닝의 시스템 구조에 대해 설명한다. 4 장에서는 제안 분산 시스템에 PSO 알고리즘을 적용한 실험에 대하여 기술하고, 마지막 5 장에서는 결과 및 향후 과제로 맺는다.

2. 기존 기술 및 관련 연구

2.1 인공신경망(Artificial Neural Network)

사람의 뇌를 연상시키는 인공신경망은 데이터를 입력하면 자동으로 복잡한 수학식으로 모델링 되는 기법이다[6]. 인공 신경망은 퍼셉트론에 기반한 모델로 각각의 특징을 추출하고 분류하기 위해서는 여러 특징들을 파악하고 여러 요소들을 조합하여 자동으로 검출하는데, 이때 학습하는 데이터를 학습 데이터(Training data)라고 하며 이를 통해 학습된 속성을 기반으로 예측 및 분류작업을 하는 알고리즘을 연구하는 것이 머신 러닝이다[1,10]. 딥 러닝도 머신 러닝의 한 종류로 빅데이터 기술이 도래하면서 인공신경망과 결합하여 많은 데이터를 모델링에 쓸 수 있게 됨으로써 정확성이 더욱 높아졌다.
인공 신경망의 이용은 데이터의 특징을 추출하고 다시 추출한 것들을 다른 머신 러닝 알고리즘의 입력으로 사용하여 분류와 군집화를 가능하게 했다

2.2 딥 러닝(Deep Learning)

딥 러닝은 인공신경망에 많은 수의 Layer 를 만들고 학습시키는 다양한 방법을 제시한다[3]. 특히 신경망을 이용한 패턴인식의 경우 은닉층의 개수를 확장하는 시도를 통해 복수개의 Layer 별로 특징을 자동으로 추출하여 학습하기 때문에 기존에 Layer 별 특징을 수동으로 추출하여 학습시키던 방식에 비해 매우 큰 강점이 있다. 2 개 이상의 은닉층을 가진 신경망을 심층 신경망이라고 하며 이를 학습하기 위한 기법이 딥 러닝이다[1].

2.3 딥 러닝 병렬 분산 처리

2.3.1 딥 러닝 병렬 분산 처리

딥 러닝 시 학습의 가속화는 딥 러닝 시스템에 크게 영향을 미친다. 이러한 학습 가속화를 위한 방식으로는 크게 두가지 방식을 들 수 있다. 첫 번째 방식은 다양한 영역에서의 데이터셋(Data Set)을 여러 대의 컴퓨터가 분배하여 학습하는 데이터 병렬처리(Data Parallelism)방식이다. 이때 각 분산 컴퓨터가 입력 데이터를 나누어 학습 함으로써 발생되는 로컬 가중치를 다른 분산 컴퓨터와 교환하게 된다. 두번째 방식은 각 분산 컴퓨터에 데이터가 아닌 딥 러닝 모델을 분배하여 학습하는 모델 병렬처리(Model Parallelism)방식을 들 수 있다. 이는 딥 러닝 모델이 하나의 컴퓨터에서만 처리 될 수 없을 정도로 방대할 경우 다수의 분산 컴퓨터에서 모델을 나누어서 처리해야 하며, 이때 각 분산 컴퓨터에서는 모든 입력 모델에 대해 학습을 수행하며 부분적으로 계산된 로컬 가중치를 다른 분산된 컴퓨터들과 서로 교환하게 된다[7].

2.3.2 Parallel DNN (Parallel Deep Neural Network)

딥 러닝 Parallel DNN 은 다수의 층을 가진 심층신경망(Deep Neural Network)을 활용하여 머신 러닝을 수행하는 것에 기반을 두고 있다. Parallel DNN 은 각 모델 업데이트를 할 때마다 기울기 및 모델 파라미터 전송을 해야 하는 점을 개선하기 위해 각 노드 간의 오버헤드(overhead)를 최소화하는 새로운 병렬학습구조이다. 이는 학습 데이터(Training Data)를 여러 개의 데이터로 분할하고, 각 심층 신경망에서 분할된 데이터(subset data)로 독립적으로 학습하는 방법이다. Parallel DNN 은 다중 GPU 환경에서 효율적으로 독립학습을 가능하게 해주기 때문에 대용량의 데이터를 빠르게 병렬처리하기에 적합한 구조이다[8].

2.4 PSO 알고리즘

PSO(Particle Swarm Optimization)는 AI 를 기반으로 하는 경험적 전역 최적화기법의 알고리즘으로, 불확실하고 복잡한 영역 문제의 최적화 알고리즘이다. 이 알고리즘은 각종 공학적 문제의 최적화 변수 값을 각 탐색 범위 내에서 조절함으로써 주어진 비용함수(cost function) 값을 최소화 또는 최대화하는 해를 찾아내는 컴퓨터 연산기법이다. 최적화 알고리즘은 찾고자 하는 해의 요구 수준에 따라 크게 전역 최적화 기법(global optimization method)과 지역 최적화 기법(local optimization method)으로 나뉜다. 전역 최적화 기법은 다소 시간이 걸리더라도 전체 탐색영역에서 가장 좋은 해를 찾는 것을 목표로 하며, 지역 최적화 기법은 단시간에 일부 탐색영역 내에서 가장 좋은 해를 찾는 것을 목표로 한다[9].

3. 분산 학습 시스템

3.1 전체 시스템 구조

본 논문에서 제안하는 분산 학습을 위한 전체 시스템의 구성은 Figure 1 과 같다. 전체 시스템에 공통 저장소를 두고, 분산 학습에 참가하는 모든 노드들은 실시간으로 중앙 저장소와 상호작용을 통해 자신의 가중치를 개선하는 구조이다.

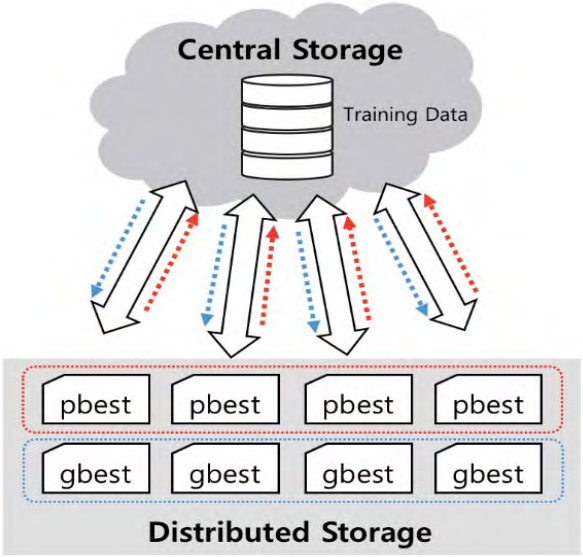

Figure 1. Overall System Architecture

각 노드는 중앙 저장소에서 동일한 학습 데이터(Training data)를 얻고, 랜덤하게 가중치를 초기화한 후 학습을 시작하여 매번 학습이 완료될 때마다 현재까지 가장 좋은 성능을 보인 가중치 값을 저장한다. 일정한 지역 학습시간이 지나면 자신이 가지고 있는 가장 높은 가중치 값을 중앙 저장소에 업데이트하고, 중앙 저장소가 가지고 있는 가장 높은 가중치 값을 내려 받아 자신의 가중치를 재조정하고 다시 학습을 하는 반복하는 구조이다.

각각의 노드에서 시행되는 학습 알고리즘의 전체적인 의사코드는 다음과 같다.

Table 1. Overall pseudo-code of the learning algorithm

```
Do
    Do
        Learning
        Calculate fitness Value
        Update local best weight
    While max learning iteration is not attained
    Upload local best weight to cloud
    Download node's best weight from cloud
    Update local weight using Node Update Algorithm
While maximum iteration or minimum error criteria is not attained
```

3.2 Node Weight update

각각의 노드에서 자신의 신경망 가중치 값 조정은 다음의 공식을 적용한다.

$$v[] = v[] + c1 * rand() * (pbest[] - present[]) + c2 * rand() * (gbest[] - present[]) \quad (1)$$

$$present[] = present[] + v[] \quad (2)$$

식(1)에서 v[]는 현재 가중치의 속도(velocity)를 의미하고, 식(2)에서 present[]는 현재 신경망의 가중치 값을 의미한다. gbest[]는 노드 전체에서 가장 좋은 성능을 보여준 가중치 값을 의미하며, pbest[]는 현재 노드에서 가장 좋은 성능을 보여준 가중치 값을 의미한다. rand()함수는 0~1 사이의 임의의 숫자를 의미하고, c1, c2 은 학습 factor 들이다.

4. 실험 및 결과

중앙 저장소는 웹 인터페이스와 데이터베이스를 이용하여 구현했다. 각각의 노드들은 편의를 위해 가상머신으로 구성했고, 리눅스에서 python 기반의 Keras를 이용하여 구현했다.

학습 데이터(Training data)로는 피마족 인디언의 당뇨병 발생 데이터셋을 사용하였고, 노드는 동일한 데이터에 대하여 PC 개수를 각각 10, 20, 30 개 사용하였다. 실험 결과는 Table 2 와 같다.

Table 2. Pima Indians diabetes Experiment Results

	T (Per epoch)	T+1 (Per epoch+150)	T+2 (Per epoch+300)
10 개	71.61	74.08	75
20 개	75.26	76.82	77.21
30 개	78.12	79.94	81.77

Table 2 에서 가로축은 시간당 epoch 값에 따른 결과값을 나타내며, 세로축은 분산 환경의 PC 개수를 나타낸다. 이때 가로축 epoch 는 시간당 150 번씩 반복 횟수를 증가시켰다. 제안하는 시스템의 성능 평가를 위해 피마족 인디언의 당뇨병 발생 데이터셋을 학습시키는데 적용했다. 사용한 데이터셋은 인스턴스 수가 768 개, 속성이 8 가지가 포함되어 있다

인공신경망으로 쓰이는 알고리즘은 2 개의 은닉층을 가지고 'Keras'가 지원하는 Layer 로 노드가 전달받은 데이터의 가중치를 고려해 합산한 계산값을 상대적으로 학습 속도가 빠른 활성화 함수인 'ReLU'를 이용하여 전달한다.

학습이 완료되면 각기 다른 알고리즘에서 추출한 결과값들을 모델로 저장한다. 기존 중앙 저장소에 저장된 모델이 없으면 노드들의 모델을 불러들여 저장하고 1 시간마다 시스템에서 정확성이 가장 높은 결과를 낸 모델만 받고 다시 반복 학습을 시켜서 가중치를 초기화 시킨 후 중앙 저장소에 저장된 정확성이 가장 높은 모델을 노드들에게 전달한다.

5. 결론 및 향후 과제

고사양이 아닌 계산환경에서 효율적으로 딥 러닝을 하기 위해서는 분산 학습 시스템의 연구가 필수적이다. 본 논문에서는 노드라고 부르는 여러 대의 저 사양 컴퓨터를 연결하고 중앙 저장소를 둔 분산 시스템을 제안했다.

학습 알고리즘의 전체적인 구조는 각각의 노드가 일정시간 동안 지역 학습을 하고, 이후 중앙 저장소에 있는 다른 노드들의 학습결과 중 가장 높은 결과와 자신이 가지고 있는 가장 높은 결과를 낸 신경망의 가중치 값을 이용하여 다시 조정하는 방법이다.

본 논문은 제안한 시스템의 성능을 확인하기 위해 간단한 실험을 실시했고 피마족 인디언의 당뇨병 발생 데이터셋(data set)이 분산 환경에서 훈련시키기에는 적은 데이터라 다양한 결과 값을 내지 못한 한계점이 있어 미비하지만 의미 있는 결과를 얻을 수 있었다.

향후 과제는 검증된 대용량 데이터를 이용하여 10 개 이상의 노드로 구성된 시스템에서 성능을 측정해 보는 추가적인 연구를 해볼 수 있을 것이다.

참고문헌

[1] 이용규, 이일병, "깊은 신경망을 이용한 회전객체 분류 연구", 『한국 지능시스템학회 논문지』, 25.5 (2015): 425-430.

[2] 허민오, 김경민, 장병탁, "딥 러닝 기반 비디오 스토리학습기술", 『한국 멀티미디어 학회지』, 20.3 (2016): 23-40.

[3] 정우근, et al. "딥 러닝을 위한 HW 시스템 및 SW 라이브러리", 『정보과학회지』, 34.9 (2016): 10-20.

[4] 유영선. "딥 러닝을 이용한 저널 추천 방법론". 연세대학교 공학대학원, 2015.

[5] 이민학. "임베디드 환경에서의 딥 러닝 프레임워크 성능 개선과 평가", 인천대학교 대학원 석사 학위 논문, 2017.

[6] 김정혁·김호찬. "PSO 알고리즘을 이용한 건물 실내 온도제어", 『한국 산학 기술학회논문지』, 2013, pp. 2536-2543.

[7] 안신영, 박유미, 임은지, 최완. "딥 러닝 분산처리 기술동향", Electronics and Telecommunications Trends. Vol. 31, No. 3, June 2016, pp. 131-141.

[8] 이은상. "음성 인식을 위한 다중 심층 신경망 병렬 학습", 고려대학교 대학원 석사학위 논문, 2017.

[9] 이민수. "A Bio Chip Data Classification Method using the PSO Algorithm", 『정보과학회 논문지』, 2012.

[10] 최지선. "SVM 과 RVM 을 이용한 분류성능 비교 연구", 인하대학교 대학원 석사 학위 논문, 2012.