

블록체인을 활용한 양질의 기계학습용 데이터 수집 방안 연구

김영량¹ · 우정훈¹ · 이재환² · 신지선^{3*}

High-quality data collection for machine learning using block chain

Youngrang Kim¹ · Junghoon Woo¹ · Jaehwan Lee² · Ji Sun Shin^{3*}

¹Graduate student, School of Electronics and Information Engineering, Korea Aerospace University, Goyang-city, 10540 Korea.

²Assistant professor, School of Electronics and Information Engineering, Korea Aerospace University, Goyang-city, 10540 Korea.

^{3*}Assistant professor, Department of Computer and Information Security, Sejong University, Seoul, 05006 Korea.

요 약

기계학습의 정확도는 학습용 데이터의 양과 데이터의 품질에 많은 영향을 받는다. 기존의 웹을 기반으로 학습용 데이터를 수집하는 것은 실제 학습과 무관한 데이터가 수집 될 수 있는 위험성이 있으며 데이터의 투명성을 보장할 수가 없다. 본 논문에서는 블록체인구조에서 블록들이 직접 병렬적으로 데이터를 수집하게 하고 각 블록들이 수집한 데이터를 타 블록의 데이터와 비교하여 양질의 데이터만을 선별하는 방안을 제안한다. 제안하는 시스템은 각 블록들은 데이터를 서로 블록체인을 통해 공유하며 All-reduce 구조의 Parallel-SGD를 활용하여 다른 블록들의 데이터와 비교를 통해 양질의 데이터만을 선별하여 학습용 데이터셋을 구성할 수가 있다. 또한 본 논문에서는 제안한 구조의 성능을 확인하기 위해 실험을 통해 기존의 벤치마크용 데이터셋의 이미지를 활용하여 변조된 이미지 사이에서 원본 이미지만을 양질의 데이터로 판별함을 확인하였다.

ABSTRACT

The accuracy of machine learning is greatly affected by amount of learning data and quality of data. Collecting existing Web-based learning data has danger that data unrelated to actual learning can be collected, and it is impossible to secure data transparency. In this paper, we propose a method for collecting data directly in parallel by blocks in a block - chain structure, and comparing the data collected by each block with data in other blocks to select only good data. In the proposed system, each block shares data with each other through a chain of blocks, utilizes the All-reduce structure of Parallel-SGD to select only good quality data through comparison with other block data to construct a learning data set. Also, in order to verify the performance of the proposed architecture, we verify that the original image is only good data among the modulated images using the existing benchmark data set.

키워드 : 블록체인, 데이터 수집, 기계학습, 병렬 딥러닝

Key word : Block chain, Data collection, Machine learning, Parallel-SGD

Received 12 November 2018, Revised 3 December 2018, Accepted 4 December 2018

* Corresponding Author Ji Sun Shin(E-mail:jsshin@sejong.ac.kr, Tel : +82-2-3408-3888)

Assistant professor, Department of Computer and Information Security, Sejong University, Seoul, 05006 Korea.

Open Access <http://doi.org/10.6109/jkiice.2019.23.1.13>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

기계학습은 최근 다양한 인공지능 분야에서 활용되고 있다. 특히 딥러닝(Deep learning)은 기계학습의 대표적인 알고리즘으로 이미지 분류와 자연어 처리 분야에서 높은 성능을 보이고 있다. 딥러닝을 활용하여 특정 기능을 구현할 때, 해당 기능에 사용되는 데이터셋을 구성하여 학습을 해야 한다. 하지만, 다수의 데이터를 수집하는데 어려움이 있을 뿐만 아니라 무관한 데이터들이 수집될 수 있다는 한계가 있다.

Parallel-SGD[1]는 딥러닝에서 다수의 서버를 사용하여 데이터 병렬화를 수행하며 매개변수 학습을 위해 사용된다. 딥러닝 분산 처리시 Aggregation을 수행하는 방법에는 Parameter Server 방식과 All-reduce 방식이 있다 [2]. Parameter Server 방식은 Server-client 구조이며 All-reduce 방식은 Peer-to-Peer 구조이다. All-reduce 방식의 Peer-to-Peer 구조는 블록체인과 유사한 구조를 갖고 있다.

블록체인 또한 기존의 Server-client 구조의 중앙 서버에 의존하는 모델과 다르게 블록체인 네트워크를 이루는 각 노드들이 블록체인 데이터를 가지고 있다 [3]. 블록체인은 각 블록을 가지고 있는 객체들이 동일한 블록체인망에 참여중인 블록들의 데이터만을 공유한다. 또한, 이 망 안에서 집단 연산을 수행하기 때문에 보안성이 보장되는 Peer-to-Peer 기반의 분산처리 시스템으로 활용이 가능하다.

본 논문에서는 기계학습에 사용할 양질의 학습용 데이터셋을 구성하기 위하여 블록체인 구조로 구성된 분산처리 시스템을 활용하여 데이터를 수집하는 시스템을 제안하였다. 블록체인을 사용하면 네트워크에 참여한 사용자들이 직접 학습용 데이터를 입력하여 빠른 시간에 데이터를 수집할 수 있지만, 특정 데이터에 과적화가 이루어질 수 있다. 이를 해결하기 위하여 All-reduce 방식의 Parallel-SGD를 추가로 사용하여 각 사용자의 데이터를 마다 우선순위를 부여하여 양질의 데이터를 추출하였다.

제안한 시스템의 성능을 확인하기 위해 본 논문에서는 MNIST[4] 데이터셋을 활용하였다. 각 블록으로 하여금 동일한 이미지에 대하여 각각 원본 이미지와 변조된 이미지를 입력 받게 하여 양질의 이미지를 입력 받은 블록에게 Reward를 부여하게끔 구현하였으며 양질의

이미지인 원본 이미지를 입력 받은 블록이 가장 많은 Reward를 받았음을 확인하였다.

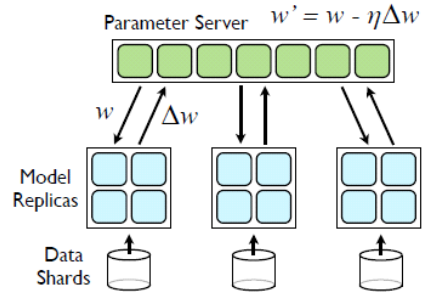


Fig. 1 Structure of a Parameter Server system consist with parameter server and workers including model replica

II. 관련 연구

특정 주제에 대한 이미지를 수집하는 방법은 본 논문과 같이 이미지 자체를 기준으로 수집하는 방법도 있지만 검색 엔진을 사용하는 방법 역시 많이 사용되고 있다 [5]. 검색 엔진을 활용하는 경우 특정 이미지에 대한 키워드를 통해 검색 후 관련 URL을 수집 후 해당 URL에 포함된 이미지를 데이터 셋에 추가하는 방식으로 동작한다. 이미지 인식 이외에 자연어 처리와 같은 경우 주로 신문, 소셜 네트워크에서 관련 이슈를 검색하여 주요 단어를 포함한 문장을 수집하는 방식으로 동작한다. 최근에는 이와 같이 웹을 통해 수집되는 데이터의 정확성을 개선하기 위하여 본 논문에서 제안하는 방법과 같이 기존에 학습을 수행한 기계학습 모델을 사용하여 정확도를 판단하여 일정 정확도를 가지는 데이터만을 데이터셋에 추가하는 방식으로 동작하고 있다.

III. 딥러닝(Deep learning)

딥러닝(Deep learning)은 대표적인 기계학습 알고리즘으로 이미지 분류와 자연어 처리와 같은 다양한 분야에서 높은 성능을 보이고 있다. 다른 기계학습 알고리즘과 마찬가지로 Deep learning 역시 다수의 데이터를 반복적으로 입력으로 사용하여 실제 연산에 사용되는 매개변수를 변경하면서 학습이 이루어진다. 딥러닝 연산

은 크게 두 가지 과정으로 그 연산을 나눌 수가 있다. 첫 번째는 Feed-forward과정으로 기존의 초기화된 매개변수를 사용하여 입력된 데이터를 개발자가 정의한 학습모델을 사용하여 Loss 값을 연산한다. 두 번째는 Feed-forward 과정에서 연산한 Loss 값을 사용하여 Gradient-descent를 수행하여 매개변수를 학습 하는 Back-propagation 과정이다.

대표적인 학습용 데이터 셋으로는 MNIST, CIFAR-10 그리고 ImageNet이 있으며 각각 수 만장의 이미지로 구성된 데이터셋이다. 하지만 실제 Deep learning을 활용하여 특정 기능을 구현하려면 실제 해당 기능에 사용되는 데이터 셋을 구성하여 학습을 수행해야한다.

특정 기능을 위한 데이터셋을 구성하는 방법으로는 개발자가 직접 임의의 데이터를 생성하거나 실제 사용자들이 입력한 데이터를 수집하는 것이 있다. 하지만 직접 임의의 데이터를 생성하는 것은 그 양에 한계가 있으며 과적화 현상으로 인해 정확도가 감소할 수가 있다. 사용자들의 입력을 수집하여 구성할 경우 다수의 데이터를 빠르게 모을 수 있다는 장점이 있지만 실제 학습과 무관한 데이터들이 수집될 수 있다는 위험성이 있다.

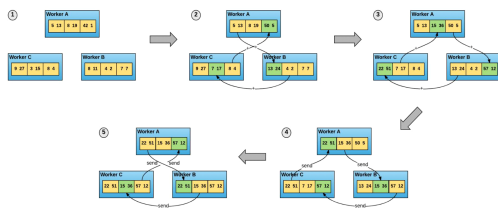


Fig. 2 Structure of All-reduce aggregation method consisting of only Worker

3.1. Parallel-SGD(Stochastic gradient descent)

SGD는 딥러닝에서 매개변수 학습을 위해 사용된다. 딥러닝을 다수의 서버를 사용하여 데이터 병렬화를 수행 할 경우 각 서버는 입력된 데이터를 통해 매개변수를 학습하기 위한 gradient를 연산하고 특정 서버에서 여러 개의 gradient들을 aggregation하여 학습에 반영한다. Aggregation은 모든 Gradient들을 합하여 전체 서버 수로 나누는 Parallel-SGD를 사용하여 연산이 이루어진다.

딥러닝 분산 처리 시 Aggregation을 수행하는 방법은 Parameter Server방식과 All-reduce 방식이 있다. Parameter Server방식은 그림 1과 같이 일반적인 Server-client 구

조로 Parameter Server와 Worker로 구성되어 있다. Worker들은 각자 다른 데이터를 입력으로 SGD를 수행하여 Gradient를 계산한다. Parameter server는 Worker들이 연산한 gradient를 모아 aggregation을 수행 후 이를 매개변수에 반영하고 학습된 매개변수를 다시 Worker들에게 배포한다. All-reduce 방식은 그림 2에서 알 수 있듯이 Parameter server 방식과 달리 Gradient를 계산하는 Worker들만으로 구성되어 있다. Aggregation의 경우 모든 Gradient를 한 곳에 모으지 않고 MPI(Message passing interface)를 사용하여 Worker들이 직접 Peer-to-Peer 방식으로 데이터를 교환하며 aggregation 연산을 수행한다.

IV. 블록체인

기존의 서버 클라이언트 모델은 중앙 서버가 모든 데이터를 저장 관리하는 구조이다. 이는 중앙 서버에 문제가 발생했을 경우 전체 시스템의 기능이 마비되는 단점이 있다. 블록체인은 Peer-to-Peer 모델을 적용한 기술이다 [6]. 이는 중앙 서버에 의존하는 모델과 다르게 블록체인 네트워크를 이루는 각 노드들이 블록체인 데이터를 가지고 있다 [7]. 대표적인 활용으로는 전자 화폐인 비트코인과[8], 이더리움(Ethereum)[9]이 있다.

드론 군집 비행에서 예를 들면, 기존의 서버 클라이언트 모델에서 특정 드론이 어떤 물체를 식별하지 못하고 있는 경우 중앙 서버에 상황을 알린 후에 서버가 독립적으로 판단한다. 하지만, 블록체인 모델에서는 실제 군집 비행을 수행중인 드론들 간에 상황 공유하고 함께 판단하므로 오차를 줄일 수 있다. 또한 블록체인은 각 블록을 가지고 있는 각 객체들이 동일한 블록체인망에 참여 중인 블록들의 데이터만을 공유하고 집단 연산을 수행하기에 보안성이 보장되는 Peer-to-Peer기반의 분산처리 시스템으로 활용이 가능하다.

블록 생성 방법에는 크게 POW(Proof-Of-Work)방식과 POS(Proof-Of-Stack)방식이 있다. POW방법은 컴퓨팅 파워를 사용하여 블록을 생성한다. 이 과정을 채굴이라고 하며, 채굴을 하는 노드를 Miner 라고 한다. 채굴에 성공한 노드는 블록을 생성할 때 사용한 컴퓨팅 파워에 대한 보상을 받는다. 또한, 생성된 트랜잭션과 블록이 유효한지는 각 노드들이 가지고 있던 블록체인 데이터

를 사용하여 검증한다. 따라서, 한 노드가 공격당해도 다른 노드들의 검증을 통해 블록체인 시스템에는 영향을 받지 않는다는 특징이 있다. 대표적으로 POW (Proof-Of-Work)방식을 사용하는 비트코인의 경우, 네트워크를 마비시키기 위해서 전체 네트워크 컴퓨팅 파워의 51%이상을 점유해야 하는데 이는 한화로 약 2600 억원 가치이다. 블록에 등록되어 있는 모든 데이터들은 해시함수로 암호화 되어있기 때문에 위조가 불가능하며 보안성이 높다는 장점이 있다.

그러나, POW방식은 컴퓨팅 파워에 소모되는 전기 에너지 비용 낭비와 전용 채굴기 등장으로 채굴을 독점한다는 문제점이 있다. POW방식의 이러한 문제점을 해결하기 위해 POS방식이 제안되었다. POS방식에서는 채굴 작업이 없다. Miner 대신, 거래내역을 검증하고 다음 블록을 생성할 Validator가 존재한다. 블록 생성에 참여하길 원하는 노드들은 자신이 가진 토큰의 전체 또는 일부를 일종의 보증금처럼 블록체인 네트워크에 묶어 놓고, 그 양에 비례하여 Validator가 될 수 있는 확률을 갖는다. 이 확률에 기반하여 다음 블록을 생성할 Validator가 무작위로 선택된다. 블록 생성 과정 참여에 대한 보상은 묶어 놓은 토큰의 이자 방식으로 받게 된다.

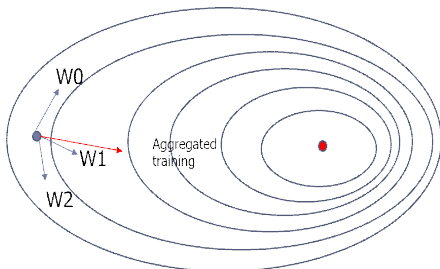


Fig. 3 Shifting of parameter values using Parallel-SGD

V. 제안 설계

본 논문에서는 기계 학습에 사용할 양질의 학습용 데이터 셋을 구성하기 위하여 블록체인 구조로 구성된 분산 처리 시스템을 활용하여 데이터를 수집하는 방안을 제안한다. 블록체인 망을 사용하여 데이터를 수집할 경우 해당 네트워크에 참여한 사용자들이 직접 학습용 데이터를 입력하기에 빠른 시간에 많은 데이터를 수집할 수가 있다. 하지만 사용자들이 입력한 데이터를 학습에

사용할 경우 특정 데이터에 과적화가 이루어질 수 있다. 이를 해결하기 위하여 본 논문에서 제안하는 시스템에서는 All-reduce방식의 Parallel-SGD를 사용하여 각 사용자의 데이터들 마다 우선순위를 부여하여 양질의 데이터만을 추출하고자 한다.

Parallel-SGD의 Aggregation 방식은 다음과 같다. Worker들은 동일한 매개변수를 parameter server로부터 배포받아 학습을 시작하고 각자의 입력 데이터를 통해 Gradient를 계산한다. 그림 3의 W0, W1, W2의 화살표는 각 gradient를 통해 매개변수 값의 변화를 나타낸다. Aggregation의 경우 각 Gradient 벡터의 평균값으로 표현할 수 있다.

Algorithm The process of determining the data quality

```

1: function BlockIndexofHighQuality:
2:    $Data_n$  is Data from  $N_{th}$  Block
3:
4:   with each Block:
5:      $logit_n = model(Data_n)$ 
6:      $loss_n = loss_{function}(logit_n, label)$ 
7:      $gradient_n = GradientOptimizer$ 
8:       .computeGradient( $loss_n$ )
9:      $List_{grad}.append(gradient_n)$ 
10:
11:    $gradSum = sum(List_{grad})$ 
12:
13:   for  $i$  in range( $n$ ):
14:      $List_{inner} = innerproduct($ 
15:        $gradSum, gradient_n)$ 
16:
17:
18:    $I = List_{inner}.sortInDescOrder(List_{inner})$ 
19:   return  $I$ 
20:
21: end function

```

Fig. 4 The process of determining the data quality by calculating the slope of the data input to each block

본 논문에서는 이 Aggregated gradient를 각 Worker에서 학습에 사용한 데이터를 평가하기 위한 지표로서 사용하고자 한다. Worker들이 생성한 gradient 중에서 aggregated gradient에 가까운 gradient 일수록 더욱 빠르게 parameter를 수렴시키는데 공헌 할 수가 있다. 이는 Aggregated gradient와 각 gradient를 내적 한 값을 통해 비교를 할 수 있다. 내적 값이 가장 큰 gradient일 수록 해당 gradient를 연산한 input data를 현재 학습모델에

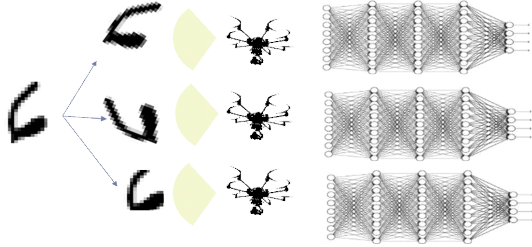


Fig. 5 Perform training using modulated data for each worker

가장 적합한 양질의 데이터라 볼 수 있다. Gradient를 사용하여 양질의 데이터를 판별하는 과정은 그림 4의 순서도와 같으며 아래의 단계로 설명할 수 있다.

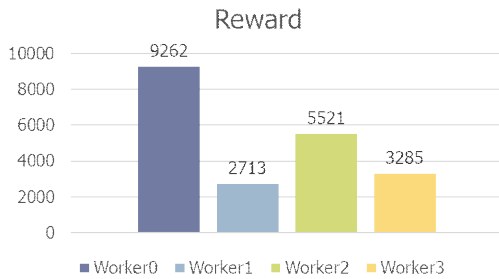


Fig. 6 The number of times that workers using different modulations received rewards

Step 1. 각 블록은 입력받은 데이터를 통해 기존의 학습 모델을 사용하여 Feed-forwarding 과정을 통해 Loss를 연산

Step 2. 연산한 Loss를 통해 Back-propagation을 수행하여 매개변수를 학습 할 Gradient를 연산

Step 3. Gradient를 연산한 블록들은 All-reduce 방식을 사용하여 gradSum을 구하는 Aggregation을 수행

Step 4. 각 블록에서 연산한 Gradient를 gradSum과 벡터 내적을 수행 하여 내적 값들을 비교 후 내림차순으로 정렬 후 큰 내적 값을 가지는 gradient를 연산한 블록일 수록 입력 데이터를 양질의 데이터로 판단

블록체인을 사용할 경우 각 블록들은 입력된 데이터를 가지고 gradient를 계산한다. 계산된 gradient는 All-reduce 방식을 사용하여 다른 블록들의 gradient를

공유하여 aggregated gradient를 계산하며 이에 가장 가까운 gradient를 계산한 블록의 데이터를 양질의 데이터로 판단한다. 그림 3의 경우 붉은 색으로 표시된 aggregated gradient에 가장 가까운 gradient를 계산한 블록은 W1의 gradient라 판단 할 수 있다. 가까운 gradient를 선정할 때는 aggregated gradient와 각각 내적을 수행하여 내적 값이 가장 큰 gradient를 가장 가까운 gradient로 판단한다. 또한 양질의 학습 데이터를 입력한 각 블록에게 이더 형태의 보상을 제공하기 위하여 Aggregated gradient와 내적한 값의 비율에 따라 개발자가 설정한 전체 이더의 수를 분배하는 방식을 제안한다. 실제 이더 리움의 경우 특정 분산 작업에 참여한 블록들에게 작업의 공헌도에 따라 이더를 분배하며 본 논문에서는 해당 데이터에 대한 gradient와 aggregated gradient를 내적한 값의 크기를 데이터 수집 작업의 공헌도로써 사용하고 자 한다.

VI. 실험 및 결과 분석

제안된 양질의 데이터 선별 방법이 올바르게 동작함을 확인하기 위하여 본 논문에서는 기존의 MNIST 데이터셋의 데이터를 활용하였다. MNIST는 0부터 9까지 총 10종류의 숫자 손글씨 이미지로 이루어진 학습용 데이터셋으로 각 7000장씩 총 7만장의 이미지로 이루어져 있다. 실험에서는 7만 장 중 1만 장의 이미지를 사용하였으며 동일한 한 장의 이미지에 대하여 그림 5와 같은 방법으로 하나의 Worker를 제외하고는 각자 변조된 이미지를 입력이미지로 사용하였다. 인식을 위한 학습 모델은 시각 이미지 분류에 주로 사용되는 Convolutional neural network인 LeNet-5를 사용하였다. 본 실험에서는 실제 학습이 아닌 데이터 품질만을 판단하기 위한 실험이기에 사전에 학습을 수행한 매개변수를 사용하였다. 사전 학습의 경우 실험에 사용할 1만장의 이미지를 제외한 6만장의 이미지만을 사용하여 학습을 수행하여 과적화를 방지하였다.

실험에서는 총 4개의 Worker를 사용하였고 연산 속도를 가속화하기 위하여 각 Worker는 하나의 GPGPU를 사용하여 데이터 인식을 수행하였다. 실험에 사용한 장비는 NVIDIA의 딥 러닝 학습용 서버인 DGX-1으로 총 8개의 Tesla P100 GPU에서 4개의 GPU를 활용하였

다. 또한 All-reduce 방식의 분산 딥 러닝을 수행해야 하기에 이를 지원하는 딥 러닝 라이브러리인 horovod를 활용하였다.

실험의 내용은 아래와 같다. 학습에 사용하지 않은 1만장의 이미지를 각 Worker의 Input으로 사용하였으며 양질의 데이터를 입력하는 것으로 가정한 Worker0를 제외한 3개의 Worker는 Input을 변조 후 데이터 품질 판단을 수행하였다. Worker1과 Worker3의 경우 각각 다른 방향으로 45도 회전한 이미지를 변조된 이미지로 사용하였고 Worker2는 전체 이미지에서 부분을 잘라낸 이미지를 변조된 이미지로 사용하였다.

실험 결과는 그림 6를 통해 확인 할 수 있다. 그림 6의 결과를 보면 총 1만장의 이미지를 입력으로 사용하면서 계산한 Gradient를 Aggregated gradient와 내적 한 결과가 가장 작은 값과 두 번째로 작은 값을 가지는 Worker에 보상을 부여하게끔 했을 때 각각의 Worker가 보상을 받은 횟수를 비교하였다. 그래프를 보면 변조된 이미지를 입력으로 사용한 다른 Worker에 비교했을 때 원본 이미지를 사용한 Worker 0가 1만 번 중 9262번으로 가장 많은 보상을 받았음을 확인할 수 가 있다. 또한 Worker 2의 경우 회전한 이미지를 사용한 Worker 1이나 Worker 3에 비해 더 많은 보상을 받았음을 확인할 수 있는데 이는 회전이미지에 비해 부분을 잘라낸 경우가 특정 숫자로 판단하기가 수월하였기에 양질의 이미지로 판단이 가능하였음을 알 수 있다. 실험을 통해 본 논문에서 제안한 양질의 데이터 판단 방법이 변조된 이미지보다 원본의 이미지를 올바르게 구별하였음을 확인할 수 있었다.

VII. 결론 및 향후 연구 계획

본 논문은 기계 학습에 필수요소인 학습용 데이터를 효율적으로 수집하기 위해 블록체인을 사용하는 방법을 제안하였다. 또한 수집된 데이터들 사이에 양질의 데이터만을 선별하기 위해 Parallel-SGD에서 계산한 Aggregated gradient를 지표로 사용하였으며 실험을 통해 이를 증명하였다. 본 논문에서는 특정 기능만을 위한 데이터를 수집한 것이 아닌 구성된 데이터 셋을 사용하여 그 성능을 확인하였지만 추후에는 제안한 시스템을 사설 블록체인망을 통해 특정 기능을 위한 데이터를 수

집하여 실제 활용 가능 여부를 확인하고자 한다.

ACKNOWLEDGEMENT

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2018-2018-0-01423) supervised by the IITP(Institute for Information & communications Technology Promotion)

References

- [1] M. Zeinkevich, M. Weimer, L. Li, A. J. Smola, "Parallelized Stochastic Gradient` Descent," *Advances in Neural Information Processing Systems* 23, pp. 2595-2603, 2010.
- [2] A. Sergeev, M. D. Balso, "Horovod: fast and easy distributed deep learning in TensorFlow," *arXiv e-prints*, Feb. 2018.
- [3] E. C. Ferrer, "The blockchain: a new framework for robotic swarm systems," *arXiv e-prints*, Jun. 2017.
- [4] Y. Lecun, C. Cortes, C. J.C. Burges, "The Mnist database" [Internet]. Available: <http://yann.lecun.com/exdb/mnist/>.
- [5] J. Li, S. Y. Shin, H. C. Lee, "Text Mining and Visualization of Papers Reviews Using R Language," *Journal of information and communication convergence engineering*, 15-3, pp. 170-174, Sep. 2017.
- [6] E. Seo, J. Jang, "Design of Driving Record System using Block Chain," *The Korea Institute of Information and Communication Engineering, Journal of the Korea Institute of Information and Communication Engineering*, vol. 22, no. 6, pp. 916-921, Jun. 2018.
- [7] V. K. Rao, R. Caytiles, "SUBGRAPH WITH SET SIMILARITY IN ADATABASE," *Asia-pacific Journal of Convergent Research Interchange, HSST*, vol. 3, no. 2, pp. 29-37, Jun. 2017.
- [8] S. Nakamoto. (2011, November) Bitcoin: A peer-to-peer electronic cash system [Internet]. Available: <http://bitcoin.org/bitcoin.pdf>.
- [9] G. Wood. (2015) ETHEREUM: A secure decentralised generalised transaction ledger [Internet]. Available: <http://gavwood.com/paper.pdf>.



김영랑(Youngrang Kim)

한국항공대학교 정보통신공학과 공학사

※관심분야: 분산컴퓨팅, 빅데이터 플랫폼, 고성능 컴퓨팅, 인공지능 플랫폼



우정훈(Junghoon Woo)

한국항공대학교 정보통신공학과 공학사

※관심분야: 분산컴퓨팅, 빅데이터 플랫폼, 고성능 컴퓨팅



이재환(Jaehwan Lee)

•서울대학교 전기공학부 공학사

•서울대학교 전기공학부 공학석사

• 메릴랜드 주립대학(University of Maryland at College Park) 컴퓨터 과학과 박사

• 현재: 한국항공대학교 항공전자정보공학부 조교수

※관심분야: 분산컴퓨팅, 빅데이터 플랫폼, 고성능 컴퓨팅



신지선(Ji Sun Shin)

2001년 2월: 서울대학교 컴퓨터공학과 졸업

2009년 5월: 메릴랜드 주립대학(University of Maryland at College Park) 컴퓨터 과학과 박사

2009년 9월~2012년 2월: 삼성 SDS 책임연구원

2012년 3월~현재: 세종대학교 조교수

※관심분야: 정보보호, 암호학, 컴퓨터 보안