

딥러닝 알고리즘과 2D Lidar 센서를 이용한 이미지 분류

Image Classification using Deep Learning Algorithm and 2D Lidar Sensor

이 준 호*, 장 혁 준**★

Junho Lee*, Hyuk-Jun Chang**★

Abstract

This paper presents an approach for classifying image made by acquired position data from a 2D Lidar sensor with a convolutional neural network (CNN). Lidar sensor has been widely used for unmanned devices owing to advantages in term of data accuracy, robustness against geometry distortion and light variations. A CNN algorithm consists of one or more convolutional and pooling layers and has shown a satisfactory performance for image classification. In this paper, different types of CNN architectures based on training methods, Gradient Descent(GD) and Levenberg-arquardt(LM), are implemented. The LM method has two types based on the frequency of approximating Hessian matrix, one of the factors to update training parameters. Simulation results of the LM algorithms show better classification performance of the image data than that of the GD algorithm. In addition, the LM algorithm with more frequent Hessian matrix approximation shows a smaller error than the other type of LM algorithm.

요 약

본 논문은 CNN (Convolutional Neural Network)와 2D Lidar 센서에서 획득한 위치 데이터를 이용하여 이미지를 분류하는 방법을 제시한다. Lidar 센서는 데이터 정확도, 형상 왜곡 및 광 변화에 대한 강인성 측면에서의 이점으로 인해 무인 장치에 널리 사용되어 왔다. CNN 알고리즘은 하나 이상의 컨볼루션 및 풀링 레이어로 구성되며 이미지 분류에 만족스러운 성능을 보여 왔다. 본 논문에서는 학습 방법에 따라 다른 유형의 CNN 아키텍처들인 Gradient Descent (GD) 및 Levenberg-arquardt (LM)를 구현하였다. LM 방법에는 학습 파라미터를 업데이트하는 요소 중 하나인 Hessian 행렬 근사 빈도에 따라 두 가지 유형이 있다. LM 알고리즘의 시뮬레이션 결과는 GD 알고리즘보다 이미지 데이터의 분류 성능이 우수하였다. 또한 Hessian 행렬 근사가 더 빈번한 LM 알고리즘은 다른 유형의 LM 알고리즘보다 작은 오류를 보여주었다.

Key words : Deep learning, deep learning neural network, convolutional neural network, object detection, image classification

* Mando Corporation

** School of Electrical Engineering, Kookmin University

★ Corresponding author

E-mail : hchang@kookmin.ac.kr, Tel : +82-2-910-5544

※ Acknowledgment

This work was supported by the National Research Foundation of Korea-grant funded by the Korean Government (NRF-2018R1D1A1B07043462).

Manuscript received, Dec. 6, 2019; revised Dec. 24, 2019; accepted Dec. 27, 2019.

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.

I. 서론

딥 러닝(deep learning)은 인간의 이해능력을 컴퓨터로 실현하는 기술인 인공지능(Artificial Intelligence, AI)을 구성하기 위해 인공 신경망(Artificial Neural Networks, ANN)에 기반하여 컴퓨터를 학습시키는 방식을 말한다. 딥 러닝은 크게 입력 레이어(input layer)와 출력 레이어(output layer) 사이에 여러 개의 은닉 레이어(hidden layer)로 이루어진 심층 신경망(deep neural network, DNN), 컨볼루션 연산을 통해 학습을 진행하는 레이어가 존재하는 컨볼루션 신경망(Convolutional Neural Network, CNN), 과거의 학습 정보를 기억하여 학습에 반영하는 형태인 순환 신경망(Recurrent Neural Network, RNN)으로 나눌 수 있다.

이 중 CNN은 1989년 LeCun에 의해 처음 제안됐으며[1], 이후에 LeNet-5라는 최초의 CNN이 구현되었다[2]. CNN 구조의 컨볼루션 레이어(convolution layer)는 컨볼루션 연산을 통해 3차원의 공간적 정보를 유지한 채 학습을 진행할 수 있다는 장점으로 인해 영상 및 음성 분야에서 활발하게 적용되고 있다[3-6].

LeNet-5가 소개된 이후 컴퓨터 기술의 발전과 계산 속도의 가속화로 인해 다양한 형식의 CNN 기법이 제시되었다: AlexNet은 ILSVRC (ImageNet Large-Scale Visual Recognition Challenge)의 2012년 대회에서 학습된 모델이 예측한 최상위 5개 범주의 오류율을 나타내는 top 5 test error 기준 1위를 차지한 CNN 기반 알고리즘으로서, LeNet-5 구조를 기본으로 2개의 GPU를 기반으로 한 병렬 구조의 네트워크이다[7]. 해당 알고리즘은 LeNet-5와 같이 컨볼루션 연산을 통해 학습이 이루어진다는 점 이외에, 학습 레이어가 많을 때 발생하는 과적합(overfitting) 문제를 발생하기 위해 적용되는 드롭아웃(dropout) 기법을 처음 소개했다. VGGNet은 2014년 ILSVRC 대회에서 높은 성능을 보인 알고리즘으로서 모든 레이어에서 가장 작은 단위의 필터를 사용함으로써 학습량의 감소를 통해 성능을 향상시킨다[8]. 이외에도 하나의 레이어에서 다양한 종류의 필터를 사용하는 GoogleNet[9], 네트워크 학습 과정에서 학습 정보가 소실되는 문제를 개선시킨 ResNet[10], 이미지 내에서 학습할 영역을 미리 제안함으로써 성능을 향상시키는 R-CNN (Regions with CNN) 등이 있다[11, 12].

본 논문에서는 이미지 데이터의 분류를 위해 LeNet-5 구조를 기반으로 딥러닝 알고리즘 종류 중 하나인 CNN 학습이 이루어진다. 이때 학습을 위한 이미지 데이터는 Lidar 센서로부터 측정된 주변 공간 정보이며 해당 공간 안에 존재하는 근접 물체의 위치에 따라 이미지 분류가 이루어진다.

Lidar 센서는 레이저를 이용하여 거리를 측정하는 센서로서 자율주행 자동차 및 무인 로봇 설계에 필요한 핵심 부품으로 주목 받고 있다[13]. 본 실험을 위해 라이다 센서를 중심으로 반경 360도에 대한 주변 위치 정보를 받아와 이미지 데이터로 변환한 후, CNN 학습을 위한 적합한 형태로 가공한다. 출력 데이터는 10개의 영역으로 나누어진 주변 영역 중 하나이다.

본 논문의 구성은 다음과 같다. 2장에서는 CNN의 기본 구조와 실험에 적용될 두 가지 종류의 학습 알고리즘을 소개한다. 3장에서는 CNN 학습을 위한 데이터 가공 방법과 CNN 구조의 초기 설정에 대해 설명하며, 실험 환경 및 결과에 대해서는 4장에서 확인할 수 있다. 그리고 5장은 본 논문에서 도출된 결론과 향후 연구 방향을 나타낸다.

II. 본론

1. CNN 구조

LeNet-5은 크게 특징 추출(feature extraction)과 분류(classification)로 나뉜다(그림 1). 특징 추출 단계는 복수의 컨볼루션 레이어(convolution layer)와 풀링 레이어(pooling layer)로 이루어져 있다. 컨볼루션 레이어는 주어진 이미지의 특징 맵(feature map)을 형성하기 위해 작은 크기(5x5 혹은 3x3)의 컨볼루션 필터를 적용한다. 이 때, 아래와 같이 하나의 필터가 이미지 데이터를 순회하며 반복적으로 컨볼루션 연산하여 특징 맵을 얻어낸다.

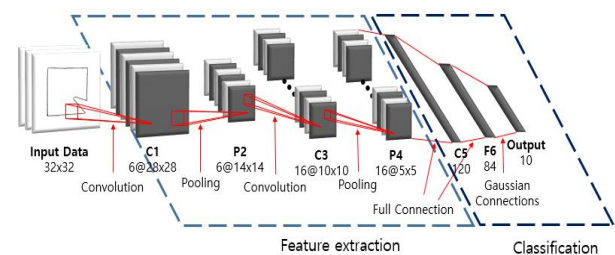


Fig. 1. Architecture of LeNet-5,
그림 1. LeNet-5의 구조

$$C(x,y) = \sum_{a=0}^{k-1} \sum_{b=0}^{k-1} I(x-a,y-b)F(a,b) \quad (1)$$

식 (1)에서 F 와 k 는 필터와 그 크기를 의미하며, I 는 2차원 입력 데이터 배열을 나타내며, x 와 y 는 이미지 데이터 행렬의 인덱스이다. 이웃한 레이어의 모든 노드가 연결되어 있는 전연결 레이어(fully connected layer) 방식을 적용한 기존 신 경망과 비교하여 필터의 파라미터 가중치를 공유(weight sharing)함으로써 학습 파라미터 수를 크게 줄일 수 있다[2].

컨볼루션 레이어 l 의 (m,n) 인덱스에서의 가중치 합 z_{mn}^l 은 아래와 같이 구해질 수 있다.

$$z_{mn}^l = \sum_{a=0}^{A-1} \sum_{b=0}^{B-1} \omega_{ab}^l a_{(m+a)(n+b)}^{l-1} + b^l \quad (2)$$

식 (2)에서 ω_{ab}^l 과 b^l 은 각각 노드 $a_{(m+a)(n+b)}^{l-1}$ 에 대한 가중치(weight)와 바이어스(bias)를 의미하며, A 와 B 는 필터의 가로와 세로의 길이를 의미한다.

또한 컨볼루션 레이어의 출력 데이터의 높이 H_o 와 폭 W_o 은 아래와 같이 계산된다.

$$H_o = \frac{H_i + 2P - H_k}{S} + 1 \quad (3)$$

$$W_o = \frac{W_i + 2P - W_k}{S} + 1$$

식 (3)에서 H_i, W_i, H_k, W_k 는 각각 입력 데이터의 높이와 폭, 필터의 높이와 폭을 의미한다. P 는 특징 맵이 입력 데이터 이미지와 동일한 크기를 유지하기 위해 사용하는 제로 패딩(zero padding)의 크기를 의미한다. S 는 몇 픽셀씩 필터가 컨볼루션 연산을 진행할지 결정하는 Stride의 크기이다. 풀링 레이어는 컨볼루션 레이어에서 도출된 특징 맵의 공간적 해상도를 줄임으로써 문제의 복잡도를 줄이고 왜곡과 이동에 대해 강인해지도록 한다[14]. 특징 맵을 일정 영역으로 나누어 그 영역의 최대값(max pooling) 혹은 평균값(average pooling)을 대표 값으로 선정한다. 풀링 레이어의 출력 데이터 크기는 아래와 같이 결정된다.

복수의 컨볼루션 레이어와 풀링 레이어를 거친 후 출력 데이터는 최종적으로 분류 단계의 전연결 레이어를 거치게 된다. 전연결 레이어는 이미지 분류(image classification)를 수행하며 현재 레이어와

다음 레이어의 노드들이 모두 연결되어 있다. 전연결 레이어에서 L 의 r 인덱스에서의 가중 합 z_r^L 은 식 (4)를 통해 구해진다.

$$z_r^L = \sum_q \omega_{rq}^{L+2} a_q^{L+1} + b_r^{L+2} \quad (4)$$

식 (2)와 마찬가지로 식 (4)에서 ω_{rq}^{L+2} 와 b_r^{L+2} 는 각각 노드 a_q^{L+1} 의 가중치와 바이어스를 나타낸다. CNN 학습은 역전파(backpropagation) 학습 알고리즘을 사용하여 수행할 수 있다[15]. 대표적으로 입력과 출력의 오차의 손실 함수(loss function)로부터 얻어지는 그라디언트(gradient)를 최소화하는 방향으로 파라미터를 학습시키는 GD(Gradient Descent) 알고리즘[16]과 비선형 시스템의 최소 제곱 문제 (least square problem)를 풀기 위해 고안된 LM Levenberg-Marquardt) 기법을 적용한 알고리즘이 있다[17]. 본 논문에서는 두 가지 알고리즘의 학습 소요 시간 및 성능에 대한 비교 연구를 수행한다.

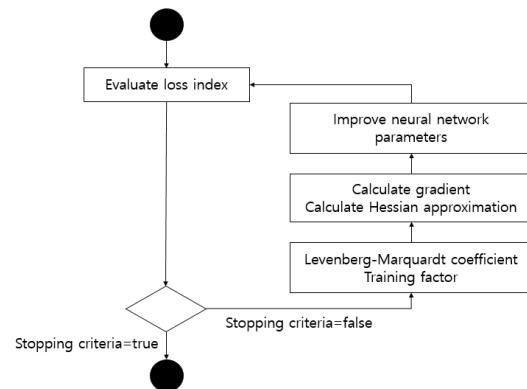


Fig. 2. Flow chart of LM algorithm.

그림 2. LM 알고리즘의 순서도

GD 알고리즘은 컨볼루션 연산을 통해 나온 결과 값들은 활성화 함수인 ReLu(Rectified Linear Unit) 함수를 이용하여 일정 임계치 이상의 값들을 출력한다. ReLu 함수는 이전에 사용되어 왔던 sigmoid 함수에 비해 역전파(backpropagation) 알고리즘 기반의 학습 과정에서 학습 파라미터의 그라디언트가 소실되는 문제(vanishing gradient problem)를 해결할 수 있기 때문에 최근 활성화 함수로 많이 사용되고 있다[18].

그림 2는 LM 알고리즘의 순서도(flow chart)를 나타낸다. 입력과 출력의 오차에 대한 손실 함수를

구하고 일정 기준치보다 높을 경우 LM 알고리즘의 안정성을 위해 추가되는 LM 계수[19]와 학습 계수, 가중치의 변화량 계산을 위한 헤시안 행렬의 예상값(Hessian approximation), 그리고 손실 함수에 대한 그라디언트를 바탕으로 가중치를 다시 갱신하고 이에 대한 새로운 손실 함수를 계산한다[20].

2. CNN 학습

가. CNN 학습을 위한 데이터 가공

라이다 센서를 기준으로 360도 반경에 대한 1032개의 png 이미지 데이터를 얻어냈으며, 이 중 1010개를 학습 데이터(training data)로, 22개를 테스트 데이터(test data)로 사용한다. 360도 반경을 10개의 영역으로 나누고 CNN 학습을 통해 근접 물체가 어느 영역에 위치해 있는지를 계산한다.

그림 1에서 CNN 학습을 위해 이용되는 입력 이미지는 32x32의 크기를 가지기 때문에 Python을 이용해 원 데이터(raw data) 크기(102x102)를 28x8로 변환시키고 MNIST 이미지 데이터 셋으로 가공한다. 본 실험에서는 학습량의 감소를 위해 입력 이미지를 RGB타입이 아닌 gray타입의 이미지로 변환하여 사용한다. 이후 MATLAB 명령어를 통해 32x32 크기의 학습 가능한 데이터로 변환한다.

나. CNN 구조 설계

시뮬레이션을 위한 MATLAB 코드는 [2]에서 제시된 기본적인 LeNet-5 구조를 구현한다. 해당 구조는 그림 1에서 보이는 바와 같이 3개의 컨볼루션 레이어(C1, C3, C5), 2개의 풀링 레이어(P2, P4)와 전연결 레이어(F6, Output)로 구성되어 있으며, 1개의 입력 이미지에 대해 10개의 출력(10개의 영역) 중 하나를 선택한다.

모든 컨볼루션 레이어의 필터 크기는 5x5로 동일하며, C1, C3, C5의 레이어에 대해 각각 6개, 16개, 120개의 필터가 1의 stride로 레이어로 인가되는 입력 이미지와의 컨볼루션 연산을 수행한다. P2와 P4의 풀링 레이트(pooling rate)는 2로 설정하여 풀링 레이어를 통과할 때마다 픽셀의 크기는 1/4로 감소한다. 전체 데이터에 대해 몇번의 학습(forward & backward)을 진행하는지를 의미하는 에폭(epoch) 값은 3으로 설정한다. LM 알고리즘에서의 파라미터 가중치 W 의 업데이트는 식 (5)에 의해 이루어진다.

$$dW = (H + \lambda I)^{-1} (t_c \nabla_W L) \tag{5}$$

식 (5)에서 $W = [\omega_1, \omega_2, \dots, \omega_i]$ 이며 i 는 총 가중치의 개수이며 본 시뮬레이션에서는 14,750개의 가중치에 대한 업데이트가 이루어진다. $H, \lambda, t_c, \nabla_W L$ 은 각각 그림 2에서의 헤시안 행렬의 예상치, LM 계수, 학습 계수, 그리고 오차의 손실 함수에 대한 그라디언트를 의미한다. 본 시뮬레이션에서 계수 λ 와 t_c 는 각각 0.001과 0.0005이다. 또한 한 번의 학습 시 사용되는 데이터의 개수인 배치 사이즈(batch size)는 1로 선정한다.

3. 시뮬레이션 결과

본 실험을 위해 SLAMTEC사의 RPLIDAR A2 360 Degree Laser를 이용했으며, 그림 3과 같이 사각형 모양의 범위를 설정하여 해당 범위 안에서 센서를 기준으로 총 10개의 영역으로 분할한 뒤, 근접 물체가 어떤 영역에 위치해 있는지를 계산한다.

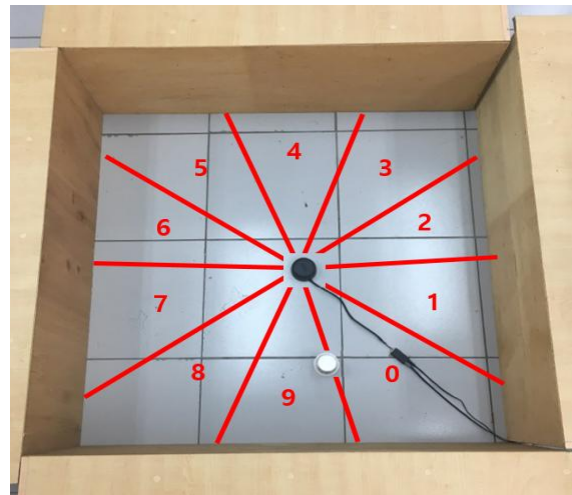


Fig. 3. Experiment setup.

그림 3. 실험 환경

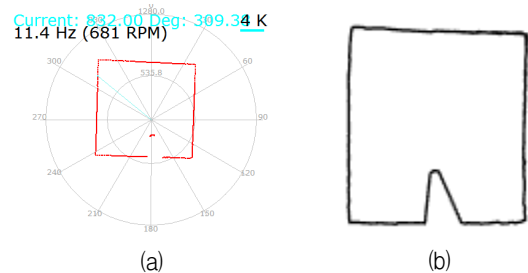


Fig. 4. (a) Position data from Lidar sensor (b) Converted image data from position data.

그림 4. (a) 라이다 센서를 이용한 위치 정보 (b) 변환된 이미지 데이터

그림 4는 라이다 센서로부터 받아오는 실시간 주변 위치 정보에 대한 모습과 MATLAB을 이용한 해당 정보들의 이미지 파일을 나타낸다. 라이다 센서는 약 680 RPM의 속도로 회전한다. 0.25ms의 샘플링 주기로 획득된 데이터는 III. 1절에서의 MNIST 이미지 데이터 셋으로의 가공을 거친 후 III. 2 절에서 설계된 GD 알고리즘과 LM 알고리즘을 이용해 각각 학습을 진행한다. LM 알고리즘의 경우 헤시안 행렬의 예상치를 매 학습(every iteration) 때마다 갱신하는 경우(LM type1)와 300번의 학습(300 iterations)마다 갱신하는 경우(LM type2)로 나누어 성능을 비교했다.

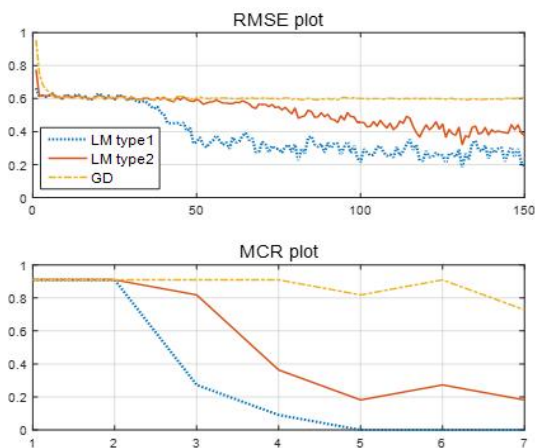


Fig. 5. Training result of CNN based on GD and LM methods.

그림 5. GD 기법과 LM 기법 기반의 CNN 구조에 대한 학습 결과

Table 1. Comparison of learning performance according to different CNN training methods.

표 1. CNN 학습 방식에 따른 성능 비교

	LM type1	LM type2	GD
RMSE	0.2092	0.4574	0.5984
MCR	0	0.1818	0.7272
Time	739.0	512.3	376.4

한 번의 에폭을 수행하기 위해 1의 배치 사이즈에 대해 1010번의 반복 학습이 수행되었으며, 그림 5와 표 1은 GD 알고리즘과 LM 알고리즘의 학습에러에 대한 평균 제곱근 편차(Rood Mean Square Error, RMSE)와 오류 검출율(Miss Classification Rate, MCR)을 나타낸다. LM type1과 LM type2는 각각 3030번과 15번의 헤시안 행렬의 예상값에 대

한 갱신이 이루어졌다. GD 알고리즘의 경우 학습이 진행되는 동안 눈에 띄는 RMSE의 감소를 보이지 못하고 0.5984의 최종 값을 나타냈으며, LM 알고리즘의 경우 두 경우 모두 감소 추세를 보였지만 LM type1과 LM type2 알고리즘은 각각 0.1867과 0.3772로 성능의 차이를 보였다. MCR도 RMSE와 동일한 성능의 차이가 나타났으며 이에 반해 학습에 소요된 시간은 LM type1 알고리즘이 739.0초로 가장 길었고 GD 알고리즘이 376.4초로 가장 짧게 나타났다.

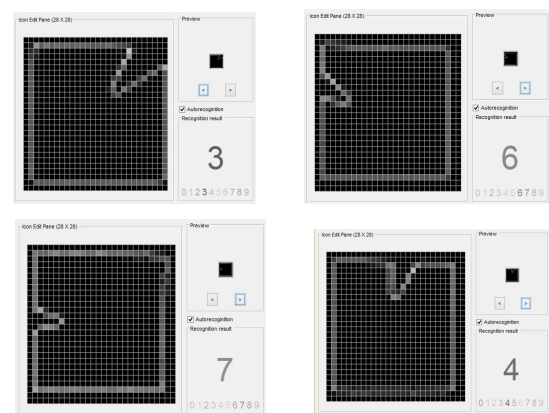


Fig. 6. Test for image classification of CNN with LM type1 algorithm.

그림 6. LM type1 알고리즘 기반 CNN의 이미지 분류 테스트

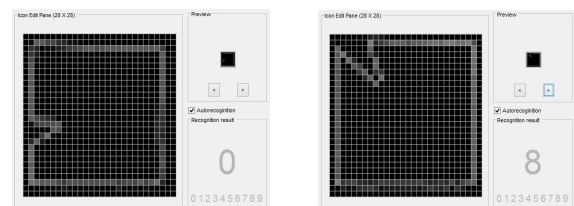


Fig. 7. Failure of image classification of CNN with GM algorithm.

그림 7. GM 알고리즘 기반 CNN의 이미지 분류 실패 모습

그림 6은 LM type1의 실제 이미지 분류에 대한 결과 중 일부를 보인다. 입력으로 들어온 28x28 사이즈의 gray 이미지 데이터에 대해 10개의 영역 중 근접 물체가 해당하는 영역을 오른쪽 하단의 Recognition result 창을 통해 나타낸다. 22개의 테스트 이미지에 대한 실험 결과 모든 이미지 데이터에 대한 분류가 성공적으로 이루어졌다. 이에 반해 그림 7은 학습 결과 세가지 학습 방식 중 가장 큰 RMSE 값을 보인 GM 알고리즘이 적용된 CNN 구조에 대해 이미지 분류가 제대로 이루어지지 않은 경우를 나타낸다.

III. 결론

본 논문에서는 딥러닝 알고리즘 종류 중 하나인 CNN 알고리즘을 기반으로 이미지 분류 작업을 진행한다. 2D Lidar 센서를 이용하여 근접 물체의 위치에 대한 데이터를 획득하고 이를 학습에 적절한 MINST 이미지 데이터 셋으로 가공한다. CNN 알고리즘은 이미지의 공간적 정보를 유지할 수 때문에 영상 분류 작업에 많이 이용되고 있다. 센서를 중심으로 360도 반경을 10개의 영역으로 나누고 센서와 근접 물체에 대한 이미지 데이터에 대해 분류 작업을 수행했다. 이미지 분류에 대한 학습을 위해 적용되는 역전파 알고리즘에 대해 GD 기법과 LM 기법을 각각 적용시켰으며, 학습 시간은 LM 기반의 CNN 알고리즘이 더 오래 소요되었지만 학습 성능 면에서 GD 기법보다 우수하게 나타났다. 특히 LM 기법에서 사용되는 헤시안 행렬의 예상값에 대한 갱신이 자주 이루어질수록 학습 성능이 향상되었으며 학습 시간은 오래 소요되었다.

Lidar 센서는 고출력 펄스의 레이저를 사용하여 정밀한 거리 감지가 가능하기 때문에 운전자 보조 시스템을 위한 위치 감지 센서로 활발하게 이용되고 있다. 따라서 향후 본 논문의 결과를 기반으로, 차량 주행 시 운전자에게 접근 차량의 위치에 대한 정보를 제공하는 연구가 이루어질 계획이다.

References

- [1] LeCun, Yann, et al., "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol.1, no.4, pp.541-551, 1989. DOI: 10.1162/neco.1989.1.4.541
- [2] LeCun, Yann, et al., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol.86, no.11, pp.2278-2324, 1998. DOI: 10.1109/5.726791
- [3] E. S. Park, C. Y. Jang, B. H. Im, H. L. Kim, "Real-time Traffic Signs Recognition using MSERs and CNN," *Journal of Institute of Control, Robotics and Systems*, Vol.23, No.6, pp.489-496, 2017. DOI: 10.5302/J.ICROS.2017.17.0049
- [4] Y. Y. Lee, S. M. Park, I. H. Jang, T. H. Kim, G. B. Sim, "CNN-based Shoe-upper Pattern Recognition and Generation of Adhesive Point," *Journal of Institute of Control, Robotics and Systems*, Vol.23, No.9, pp.725-731, 2017. DOI: 10.5302/J.ICROS.2017.17.0109
- [5] Dae-Hun Kim, Jong-Eun Ha, "Multi-Lane Detection using Convolutional Neural Networks and Transfer Learning," *Journal of Institute of Control, Robotics and Systems*, Vol.23, No.9, pp.718-724, 2017. DOI: 10.5302/J.ICROS.2017.17.0107
- [6] Jun Hoo Cho, Chang Ho Kang, Chan Gook Park, "Convolutional Neural Network-based Automatic Target Recognition Algorithm in SAR Image," *Journal of Institute of Control, Robotics and Systems*, vol.23, No.8, pp.644-649, 2017. DOI: 10.5302/J.ICROS.2017.17.0100
- [7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, Vol.25, No.2 pp.1097-1105, 2012. DOI: 10.1145/3065386
- [8] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [9] Christian Szegedy et al., "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp.1-9, 2015. DOI: 10.1109/CVPR.2015.7298594
- [10] Kaiming He et al., "Deep residual learning for image recognition," In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp.770-778, 2016. DOI: 10.1109/CVPR.2016.90
- [11] Ross Girshick et al., "Rich feature hierarchies for accurate object detection and semantic segmentation," In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp.580-587, 2014. DOI: 10.1109/CVPR.2014.81
- [12] Ross Girshick, "Fast r-cnn," In *Proceedings of the IEEE international conference on computer vision*, pp.1440-1448, 2015.

DOI: 10.1109/ICCV.2015.169

[13] Brent Schwarz, "LIDAR: Mapping the world in 3D," *Nature Photonics*, vol.4, no.7, pp.429–430, 2010. DOI: 10.1038/nphoton.2010.148

[14] Laura Leal-Taixe, Cristian Canton-Ferrer, and Konrad Schindler, "Learning by Tracking: Siamese CNN for Robust Target Association," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp.33–40, 2016.

[15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *nature*, vol.521, no.7553, pp.436–444, 2015. DOI: 10.1038/nature14539

[16] Léon Bottou, "Large-scale machine learning with stochastic gradient descent," In *Proceedings of COMPSTAT'2010*. Physica-Verlag HD, pp.177–186, 2010. DOI: 10.1007/978-3-7908-2604-3_16

[17] Ananth Ranganathan, "The levenberg-marquardt algorithm," *Tutorial on LM algorithm*, vol.11, no.1, pp.101–110, 2004.

DOI: 10.1007/978-3-7908-2604-3_16

[18] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289* (2015).

[19] Fatma Ouled Saad et al., "Quadrupole method: A new approach for solving the direct problem of electrical resistance tomography," *Journal of King Saud University-Science*, vol.29, no.3, pp. 360–370, 2017. DOI: 10.1016/j.jksus.2016.08.007

[20] Martin T Hagan and Mohammad B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE transactions on Neural Networks*, vol.5, no.6, pp.989–993, 1994.

DOI: 10.1109/72.329697

BIOGRAPHY

Junho Lee (Member)



2017 : BS degree in Electrical Engineering, Kookmin University.
2019 : MS degree in Secured Smart Electric Vehicle, Kookmin University.
2019~present : Research Engineer, Mando Corporation.

Hyuk-Jun Chang (Member)



1998 : BS degree in Electrical Engineering, Seoul Nat'l University.
2004 : MS degree in Electrical and Computer Engineering, Seoul Nat'l University.
2009 : PhD degree in Electrical and Electronic Engineering, Imperial College London.

2013~present : Associate Professor, Kookmin University.