JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 29 No. 9, pp. 1-7, September 2024
https://doi.org/10.9708/jksci.2024.29.09.001

Improving Test Accuracy on the MNIST Dataset using a Simple CNN with Batch Normalization

Seungbin Lee*, Jungsoo Rhee**

*Graduate student, Department of Smart Convergence Security, Busan University of Foreign Studies, Busan, Korea
**Professor, Department of Smart Convergence Security, Busan University of Foreign Studies, Busan, Korea

[Abstract]

In this paper, we proposes a Convolutional Neural Networks(CNN) equipped with Batch Normalization(BN) for handwritten digit recognition training the MNIST dataset. Aiming to surpass the performance of LeNet-5 by LeCun et al., a 6-layer neural network was designed. The proposed model processes 28x28 pixel images through convolution, Max Pooling, and Fully connected layers, with the batch normalization to improve learning stability and performance. The experiment utilized 60,000 training images and 10,000 test images, applying the Momentum optimization algorithm. The model configuration used 30 filters with a 5x5 filter size, padding 0, stride 1, and ReLU as activation function. The training process was set with a mini-batch size of 100, 20 epochs in total, and a learning rate of 0.1. As a result, the proposed model achieved a test accuracy of 99.22%, surpassing LeNet-5's 99.05%, and recorded an F1-score of 0.9919, demonstrating the model's performance. Moreover, the 6-layer model proposed in this paper emphasizes model efficiency with a simpler structure compared to LeCun et al.'s LeNet-5 (7-layer model) and the model proposed by Ji, Chun and Kim (10-layer model). The results of this study show potential for application in real industrial applications such as AI vision inspection systems. It is expected to be effectively applied in smart factories, particularly in determining the defective status of parts.

▸ **Key words**: MNIST dataset, Batch Normalization, Max Pooling, Fully connected layers, CNN, LeNet-5

[요 약]

본 논문은 MNIST 데이터셋을 활용한 손글씨 숫자 인식에서 합성곱 신경망(CNN)과 배치정규화(BN)를 결합한 모델을 제안한다. LeCun et al.의 LeNet-5 모델의 성과를 뛰어넘는 것을 목표로 6계층 신경망 구조를 설계하였다. 제안된 모델은 28x28 픽셀 이미지를 입력으로 받아 합성곱, 맥스 풀링, 완전연결계층을 거쳐 처리하며, 특히 배치정규화계층을 도입하여 학습 안정성과 성능을 향상시켰다. 실험에서는 60,000개의 훈련 이미지와 10,000개의 테스트 이미지를 사용하였으며, Momentum 최적화 알고리즘을 적용하였다. 모델 구성에서는 30개의 필터, 필터 사이즈 5x5, 패딩 0, 스트라이드 1을 사용하였고, ReLU 활성화 함수를 채택하였다. 훈련 과정에서는 미니배치 사이즈 100, 총 20 에포크, 학습률 0.1로 설정하였다. 결과적으로 제안된 모델은 99.22%의 테스트 정확도를 달성하여 LeNet-5의 99.05%를 상회하였으며, F1-score 0.9919를 기록하여 모델의 성능을 입증하였다. 또한, 본 논문에서 제안한 6계층 모델은 LeCun et al.의 LeNet-5(7계층 모델)와 Ji, Chun and Kim(10계층 모델)이 제안한 모델보다 더 단순한 구조로 모델의 효율성을 강조하였다. 본 연구의 결과는 AI 비전 검사기 등 실제 산업 응용에서 활용 가능성을 보여주며, 특히 스마트팩토리에서 부품의 불량 상태를 판별하는 데 효과적으로 적용될 수 있을 것으로 기대된다.

▸ **주제어**: MNIST 데이터셋, 배치정규화, 맥스 풀링, 완전연결계층, CNN, LeNet-5

• First Author: Seungbin Lee, Corresponding Author: Jungsoo Rhee
*Seungbin Lee (dltmdqls98@naver.com), Department of Smart Convergence Security, Busan University of Foreign Studies
**Jungsoo Rhee (rhee@bufs.ac.kr), Department of Smart Convergence Security, Busan University of Foreign Studies
• Received: 2024. 07. 04, Revised: 2024. 08. 23, Accepted: 2024. 08. 26.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

MNIST 데이터셋은 0~9의 28*28 픽셀 손글씨 이미지로 구성되어 있으며, 그 특성으로 인해 손글씨 분류 작업의 훈련에 사용되고 있다[1].

손글씨 분류 작업의 정확도와 관련된 선행 연구들을 살펴보면 다음과 같다. Han et al.는 MNIST 데이터셋과 잡음이 포함된 Kaggle 데이터셋에 CNN을 적용하여 테스트 정확도를 비교 분석하였다. 실제로 잡음을 입힌 Kaggle 데이터를 활용해 저자들은 93%의 테스트 정확도를 얻었다[2]. LeCun et al.는 MNIST 데이터셋을 CNN인 LeNet-5를 적용하여 학습시켜 테스트 정확도를 99.05%로 향상시킴으로서 1998년 당시 세상을 놀라게 했다[3]. Batch Normalization(BN)을 적용한 최초의 선행 연구로서 Ioffe and Szegedy는 BN계층을 포함한 은닉5계층 완전연결신경망(Fully connected ANN)을 소개하였고, 이를 최적화 알고리즘 SGD로 훈련시켜 손글씨의 테스트 정확도를 95.2%로 향상됨을 보였다. 또한, Ioffe and Szegedy는 ANN이외의 다양한 신경망 구조와 데이터셋에서 BN의 적용 가능함을 주장하였다[4]. Lee and Rhee는 Ioffe and Szegedy의 신경망에서 최적화 알고리즘 Adam을 사용하여 훈련하여 테스트 정확도를 96.47%까지 향상시켰고, BN계층을 포함하지 않은 경우는 95.3%로 나타나 BN의 효과를 재검증하였다[1]. Ioffe and Szegedy의 실험은 Adam이 소개되기 전에 수행하였음으로 Adam을 적용한 경우와는 Table 1에서와 같이 테스트 정확도에서 차이를 보인다.

Table 1. Test accuracy in previous studies

	CNN (kaggle)	LeNet-5	Ioffe's ANN with BN	ANN without BN	ANN with BN	CNN with BN (2018)
Author /Year	Han et al. /2019	LeCun et al. /1998	Ioffe & Szegedy /2015	Lee & Rhee /2023	Lee & Rhee /2023	Ji, Chun &Kim /2018
optimizer	Adam	SGD	SGD	Adam	Adam	-
test accuracy	93%	99.05%	95.2%	95.3%	96.47%	99%

BN의 효과를 완전연결신경망에서 확인하였듯이, BN을 CNN에 결합하여 MNIST 데이터셋의 테스트 정확도를 검증하려는 시도는 자연스러운 귀결일 것이다. 그러므로 본 논문은 BN을 CNN과 결합하여 LeCun et al.의 논문의 결과보다 향상된 테스트 정확도를 보여주는 것을 목표로 한다. 논문의 목표를 달성하기 위해, 제안된 CNN에 네 가지 최적화 알고리즘 -SGD, Adam, AdaGrad, Momentum-을 모두 적용하여 훈련하고 테스트 정확도를 최고로 올려주는 최적화 알고리즘을 선택한다. 참고로 Ji, Chun and Kim도 CNN과 BN을 결합하여 MNIST데이터셋에서 학습하였으나 테스트 정확도가 99%로 LeNet-5에서 학습한 테스트 정확도 99.05%를 능가하지 못하였다[5]. 이러한 결과는 CNN에서 BN의 효과가 예상만큼 뚜렷하게 나타나지 않았음을 시사한다.

III. The Proposed Scheme의 Fig. 3, Fig. 4에서 보여진 것과 같이, BN이 완전연결신경망과 결합한 경우와 마찬가지로, BN이 CNN과 결합한 경우에도 테스트 정확도를 향상시키는 결과를 보여준다.

II. Preliminaries

2.1 Fully Connected ANN

완전연결신경망은 Universal Approximation Theorem에 따라 어떠한 연속 함수라도 충분히 정확하게 근사할 수 있는 능력이 있어 복잡한 비선형 함수를 모델링할 수 있다. 또한 구현이 비교적 간단하며, 딥러닝 프레임워크에서 지원하는 표준적인 신경망 구조이고, 다양한 입력 형태를 처리할 수 있어 여러 분야에서 활용될 수 있다.

그러나 완전연결신경망의 한계점은 아래와 같이 지적된다.

공간적 구조의 무시: 완전연결신경망은 입력 데이터의 공간적 구조를 무시하고 모든 입력과 출력 사이에 완전히 연결된 구조를 가지기 때문에 이미지, 영상 등과 같은 공간적 데이터에서는 부적절할 수 있다. 이는 특히 이미지에서 픽셀 간의 관계를 고려하지 않고 처리하는 한계가 있음을 의미한다.

고차원 데이터 처리의 어려움: 완전연결신경망은 입력 데이터가 고차원일 때 처리하기 어려울 수 있다.

계산 비용: 완전연결신경망은 많은 가중치와 많은 연산이 필요하기 때문에 계산 비용이 매우 크다는 단점이 있다. 특히 대규모 데이터셋에서는 학습 및 추론 과정에서 많은 시간과 자원을 소모할 수 있다. 위와 같은 한계점 때문에 이미지의 경우에는 CNN을 사용한다.

2.2 CNN

CNN은 영상을 학습하기 위해 주로 사용되는 딥 러닝의 한 종류이다. 일반적인 딥 러닝은 한 층의 노드가 그 다음 층의 노드와 전부 연결되는 완전연결신경망인데, 이는 과도한 파라미터를 만드는 원인이 되어 학습 속도를 느려지

게 하고, 과적합의 원인이 된다. 이러한 문제를 해결하기 위해 합성곱 연산을 도입한 CNN이 개발되었다[6]. 입력받은 이미지가 합성곱계층들을 거치며 특징을 추출하고 특징 맵을 구성한다. pooling을 통해 입력 이미지의 크기 축소 문제와 과적합을 방지할 수 있다. 과적합은 dropout을 통해서도 방지하는 데 도움을 줄 수 있다. 입력 이미지가 이 두 계층을 지나며 높은 수준의 특징이 추출된다. 추출한 특징은 완전연결계층에서 분류하고 식별한다. 따라서 CNN은 이미지 데이터를 가지고 높은 정확도의 결과를 도출하는 데 효율적이기 때문에 영상처리 분야에 많이 사용된다[7].

아래는 CNN의 합성곱계층 구성요소에 대한 서술이다.

가중치 공유(Shared Weights): 합성곱 층의 가중치는 모든 공간적 위치에서 공유되어, 파라미터 수를 크게 줄이고 이동 불변성을 촉진한다.

풀링(Pooling): 풀링 층은 컨볼루션 층에서 얻은 특징 맵을 압축하는 계층으로, 이를 통하여 특성 맵의 크기가 작아져 모델의 연산량과 학습 시간이 줄어들고 사소한 특징을 제거하는 효과를 얻는다. 컨볼루션 신경망에서는 주로 최대 풀링(max pooling)을 사용한다[8].

스트라이드(Stride): 스트라이드는 필터가 입력을 따라 이동하는 간격을 의미한다. 스트라이드 값이 클수록 출력의 크기가 작아지며, 특징 맵을 더 적은 연산으로 얻을 수 있다.

패딩(Padding): 패딩은 입력의 가장자리에 추가된 픽셀을 의미한다. 주로 입력의 공간적 차원을 유지하거나 경계 정보 손실을 방지하기 위해 사용한다. 패딩에는 제로 패딩이 일반적으로 사용되며, 이는 모든 추가 픽셀이 0인 경우를 말한다[9, 10].

CNN은 이미지 분류, 객체 탐지, 의미론적 분할 등 다양한 컴퓨터 비전 작업에서 뛰어난 성과를 보여주었다. LeCun et al.의 선구적인 연구는 손글씨 숫자 인식을 위한 LeNet-5를 소개했다.

2.3 BN

배치 정규화(Batch Normalization, BN)는 Ioffe and Szegedy에 의해 도입된 기법으로, 심층 신경망 훈련을 개선하여 내부 공변량 변화를 줄이는 역할을 한다. 내부 공변량 변화란 훈련 중에 파라미터가 업데이트됨에 따라 신경망 활성화의 분포가 변하는 것을 의미한다. BN은 각 층의 입력을 미니 배치 단위로 정규화함으로써 이 문제를 해결한다.

정규화(Normalization): 각 미니 배치에 대해, BN은 활성화 값을 미니 배치 평균을 빼고 미니 배치 표준 편차로 나누어 정규화한다.

스케일 및 이동(Scale and Shift): 정규화 후, BN은 학습 가능한 스케일 및 이동 파라미터를 적용하여 신경망의 표현력을 복원한다.

BN은 학습 과정을 안정화시켜 더 높은 학습률을 가능하게 하고, 초기화에 대한 신중한 설정을 줄이며, 정규화 기법으로 작용하여 드롭아웃의 필요성을 줄이는 등의 효과가 있다. 이 기법은 많은 최첨단 신경망 아키텍처에 널리 채택되어 성능 향상에 기여하고 있다.

III. The Proposed Scheme

기울기 소실 및 폭발과 공변량 이동과 같은 문제로 인해 심층 신경망을 훈련시키는 것은 상당히 복잡하고 어려워졌다. 특히 공변량 이동은 훈련 데이터에 대한 입력 분포의 변화를 반영한다[1, 4].

딥 러닝의 맥락에서 이것은 각 층이 각 훈련 단계에서 새로운 분포에 적응해야 함을 의미하며, 이는 상당한 속도 저하라는 부작용을 가져온다. 이러한 문제들을 피하기 위해, BN이 유망한 기술로서 도입되었다[1, 4]. BN을 사용하였을 때의 공변량 이동 완화를 포함한 알려진 장점들은 참고문헌에서 다룬다[1, 4, 11].

BN은 입력을 표준화하여 평균을 0으로, 분산을 1로 정규화 하는 과정이다. 실제로 각 미니 배치에 대한 평균과 분산을 계산한 후, 이러한 값에 기반하여 정규화를 수행하며, 이는 수식 1에서 볼 수 있다.

2015년 Ioffe and Szegedy는 MNIST 데이터셋에 역전파 코드를 포함한 BN을 최초로 적용하였다. 각 은닉층은 시그모이드 비선형 함수를 활성화 함수로 사용하며, 가중치는 작은 무작위 가우시안 값으로 초기화하였다[4].

아래의 수식 1의 오른쪽 항들은 BN의 수리적 표현이다. 방정식의 왼쪽 항들은 각각 미니 배치의 평균, 분산, 정규화된 데이터를 정의한다.

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i$$
$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \qquad (1)$$
$$\hat{x_i} = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}$$

그리고, BN이 포함된 초기화 코드와 역전파 코드는 참고 문헌에서 자세히 설명하고 있다.[1, 4, 11].

BN계층을 포함한 CNN을 소개하기 전, LeCun et al.가 제안한 LeNet-5는 아래와 같이 구성된다[3].

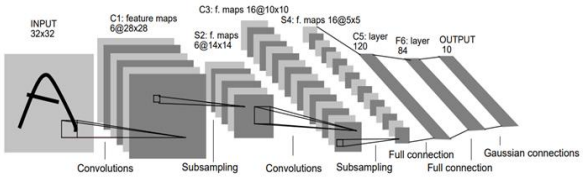

Fig. 1. Lenet-5 Model

Fig. 1에서 보여지는 것과 같이 입력 데이터는 32*32 이미지 데이터로서 합성곱계층1로 유입된다. 유입된 데이터는 pooling계층과 유사한 subsampling계층2을 통과하여 합성곱계층3, subsampling계층4을 연속적으로 통과하고, 다시 최종 합성곱계층5로 유입된다. 마지막으로 처리된 데이터는 affine계층6을 통과하여 Euclidean Radial Basis Function 및 SoftMax계층7로 출력된다. 여기서, 활성화 함수로는 tanh를 사용한다. 요약하면, LeNet-5는 정확히 7계층 신경망으로 구성 되었다.

저자들이 논문에서 제안하는 신경망은 BN을 결합한 CNN으로 아래의 Fig. 2로 구성된다. 자세한 계층구성은 아래의 2) 모델 구성 부분에서 상세히 다룬다.

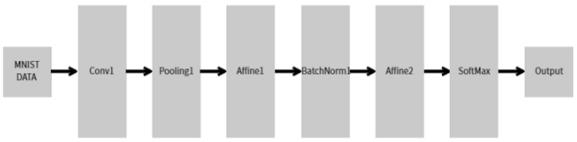

Fig. 2. A proposed CNN Model equipped with BN

본 논문에서 제안된 신경망에서의 실험을 원활하게 진행하기 위해, 아래의 절차로 진행하고 신경망에 사용한 중요 코드의 일부를 명기한다.

1) 데이터 준비: MNIST 데이터셋은 손으로 쓴 숫자들의 이미지로 구성된 데이터셋으로, 60,000개의 훈련 이미지와 10,000개의 테스트 이미지를 포함하고 있다. 이미지들은 28*28 픽셀 크기의 합성곱계층으로 데이터를 입력하기 위해 flatten=False 옵션을 사용하여 이미지들을 평탄화하지 않고 원본 2D 형태로 유지한다.

```
(x_훈련, t_훈련), (x_테스트, t_테스트) =load_mnist(flatten=False)
```

code. 1

2) 모델 구성: BN계층을 적용한 SimpleConvNet 클래스를 사용하여 CNN을 정의한다[1, 11]. max epochs는 20이고 필터의 개수는 30개, 필터 사이즈는 5, 패딩은 0, 스트라이드는 1로 구성되어있다. 이 신경망에서는 28*28 이미지 데이터가 합성곱계층1에 유입된다. 유입된 데이터는 Pooling계층2(max pooling)를 지나 Affine계층3을 통과한다. 그리고 BN계층4에서 배치정규화를 진행하고 Affine계층5을 통과하여 SoftMax계층6으로 출력된다. 활성화 함수로는 Relu를 사용한다. 실제로 논문의 신경망은 6계층 신경망으로 구성된다.

```
network = SimpleConvNet(input_dim=(1,28,28),
                        conv_param = {'filter_num': 30,
                    'filter_size': 5, 'pad': 0, 'stride': 1},
                        hidden_size=100, output_size=10,
                    weight_init_std=0.01)

self.layers = OrderedDict()
        self.layers['Conv1'] = Convolution(self.params['W1'],
                                self.params['b1'],

          conv_param['스트라이드'], conv_param['패드'])
        self.layers['렐루1'] = Relu()
        self.layers['풀링2'] = Pooling(pool_h=2, pool_w=2,
                                       stride=2)
        self.layers['아핀3'] = Affine(self.params['W2'],
                                      self.params['b2'])
        self.layers['배치정규화4'] =
                BatchNormalization(gamma=np.ones(100),
                                   beta=np.zeros(100))
        self.layers['렐루2'] = Relu()
        self.layers['아핀5'] = Affine(self.params['W3'],
                                      self.params['b3'])

        self.last_layer6 = SoftmaxWithLoss()
```

code. 2

3) 훈련 설정: Trainer 클래스는 신경망의 훈련을 담당한다. 훈련 과정에서는 100개의 미니배치를 사용하며, 총 에포크 수는 20으로 설정한다. 훈련설정에서는 Momentum 최적화 알고리즘을 사용하며, learning rate은 0.1로 설정한다. epoch마다 1,000개의 테스트 샘플을 사용하여 모델을 평가한다.

```
max_epochs = 20
Trainer(network, x_train, t_train, x_test, t_test,
                    epochs=max_epochs, mini_batch_size=100,
            optimizer='Momentum', optimizer_param={'lr': 0.1},
evaluate_sample_num_per_epoch=1000)
```

code. 3

4) 훈련 실행 및 결과 저장: trainer.train() 메소드를 호출하여 훈련을 시작한다. 훈련이 완료되면, 모델의 매개변수(가중치와 편향)를 "params.pkl" 파일에 저장한다. 이렇게 저장된 매개변수는 모델을 재사용하거나 분석할 때 활용될 수 있다.

```
trainer.train()
network.save_params("params.pkl")
print("Saved Network Parameters!")
```

code. 4

5) 결과 시각화: 마지막으로, Fig. 3, Fig. 4로 훈련과 테스트 정확도를 수치데이터로 확인하고, 이를 통해 모델의 학습 진행 상황을 epoch 별로 그래프로 표시한다. 그래프는 "train"과 "test" 라벨을 사용하여 훈련 정확도와 테스트 정확도를 각각 나타낸다. Fig. 3, Fig. 4에서와 같이, BN과 CNN이 결합된 제안된 신경망에서의 테스트 정확도는 99.22%이다.

```
train loss:0.0001983281751535189
train loss:0.0002880048602750l183
train loss:0.000581073381310562
train loss:0.0001455770924940003
train loss:0.0001160016175889032 8
train loss:0.0001169350404166830 7
train loss:0.0006261986099540035
train loss:0.0009741439282814275
train loss:0.0003936188987058515
train loss:0.0001078930892909164 5
train loss:0.0017537801845362833
train loss:0.0001067296524690692 4
train loss:0.000864874223131342
train loss:0.0004531861675833094
train loss:0.0006370124297017961
train loss:0.001236684964875127
train loss:0.0004768038880488937 3
=============== Final Test Accuracy ===============
test acc:0.9922
```

Fig. 3. A final test accuracy

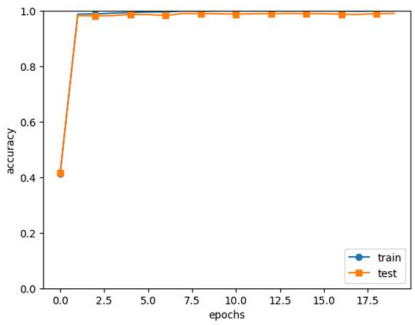

Fig. 4. CNN accuracy graph with BN applied

이 실험을 실행하기 위해 GPU를 사용하지 않은 13세대 인텔(R) 코어(TM) i7-13700K가 장착된 데스크탑을 사용했다. Python 버전 3.9.13과 Jupyter를 사용하여 Python으로 실행코드를 작성하였다.

IV. Conclusions

본 논문에서는 CNN과 BN의 결합 효과와 중요성을 강조하고, MNIST 데이터셋에서 다양한 최적화 기술을 사용하는 신경망의 학습 과정에서 BN의 역할을 연구하였다. 실제로 Fig. 3에서와 같이 BN을 적용한 1개의 합성곱계층을 가진 CNN에서 테스트정확도는 99.22%로 LeCun et al.가 제안한 LeNet-5에서의 테스트 정확도를 상회하는 것으로 나타난다.

Table 2. Detailed Comparison of LeNet-5 and the proposed CNN with BN

model	input	hidden 1	hidden 2	hidden 3	hidden 4	hidden 5	hidden 6	output
LeNet-5	mnist	conv	Sub sampling	conv	Sub sampling	conv	affine	RBF & soft max
CNN with BN	mnist	conv	pooling	affine	BN	affine	-	soft max

더 나아가 위의 Table 2에서 보여지듯이 제안한 CNN의 모델 구성이 더 간단하다는 것도 이 논문의 결과로 알 수 있다. 2)모델구성에서 언급한 것과 같이, Table 2는 논문에서 제안한 CNN과 LeNet-5의 계층 모델을 상세히 비교한 것이다.

참고로 Table 3는 서론에서 언급한 Ji, Chun and Kim이 제안한 모델과 비교한 표이다. 제안한 모델은 10계층으로 이루어져있어, 본 논문에서 제안한 모델과의 차이를 보인다.

본 연구에서는 배치 정규화를 적용한 단순 합성곱 신경망 모델을 제안하고, 이를 LeNet-5와 비교하여 테스트 정확도의 향상을 확인하고, LeNet-5보다 제안 모델이 좀 더 단순할 수 있음을 알 수 있다. 이는 제안 모델의 학술적 가치를 강조한다. 또한, 스마트팩토리 등에서 CNN을 활용한 AI비전검사기가 사람의 육안 업무를 대신하여 부품의 불량상태를 판별하고 있으므로 본 논문에서 제안한 정확도가 향상된 신경망 모델을 사용한다면 효율적인 AI비전검사기 제작에 기여할 수 있을 것으로 기대된다.

일반적으로 테스트 정확도가 낮은 경우, 사용한 모델이 적합한지를 Confusion Matrix, F1-score 등의 통계적 지표를 이용하여 점검한다. 실제 신경망 모델 훈련에서 데이터셋이 불균형한 경우 통계적 모델 적합도 검증을 한다. Simple CNN과 같은 모델 구성에서 MNIST 데이터셋을

Table 3. Detailed comparison of CNN with BN (2018) and the proposed CNN with BN

model	input	hidden1	hidden2	hidden3	hidden4	hidden5	hidden6	hidden7	hidden8	hidden9	output
CNN with BN(2018)	minst	conv	BN	maxpooling	conv	BN	maxpooling	fc625	BN	maxpooling	fc10
CNN with BN	mnist	conv	pooling	affine	BN	affine	-	-	-	-	softmax

사용할 경우에는 전통적으로 모델 적합도 검증을 생략한다. 참고로 본 논문에서 제안한 F1-score는 0.9919이다. 그럼에도 불구하고 본 연구에서는 모델의 성능을 테스트 정확도로만 판단하였고, precision, recall, F1-score, AUROC 등의 추가적인 지표를 사용하지 않은 점은 한계로 지적될 수 있다. 또한, LeNet-5가 발표된 1998년과 현재의 기술적 환경이 상당히 다르기 때문에, 본 논문에서는 훈련 연산 속도를 직접 비교하지 않았으며, 실제 연산 시간이 더 적게 소요되는지에 대한 실험도 진행하지 않았다. 이는 본 논문의 한계 중 하나이다. 그러나 퍼스널 컴퓨터(13세대 인텔(R) 코어(TM) i7-13700K가 장착된 데스크탑)를 사용하여 훈련을 진행한 결과, 훈련 시간이 약 20분정도로 측정되었다. 참고로, Kim과 Kang의 연구에서는 MNIST 데이터셋을 사용해 학습 속도 개선을 연구했으며, 이미지 중앙 부분만을 학습 데이터로 활용하여 외부 라인의 연산을 생략했을 때 학습 시간 감소와 인식률에 미치는 영향을 분석하였다. 이 연구에서는 훈련 시간이 약 49분 소요되었고, 인식률은 99%를 달성하였다[12]. 다만, 사용된 장비의 성능 차이로 인해 두 결과를 정확히 비교하기는 어렵다.

테스트 정확도를 99.5%이상을 목표로 신경망 모델을 구성한 것이었으나, 제안된 신경망 모델에서는 달성하지 못하였다. 향후 연구에서는 더 높은 테스트 정확도 달성을 위해 가중치의 초기값 설정 방법에 대한 추가 연구를 진행하여 모델을 구축함으로써 이 목표를 달성하고자 한다. 여유가 된다면 MNIST 외의 다양한 데이터셋에 대한 모델의 성능을 검증하여 일반화 가능성을 연구할 계획이다.

ACKNOWLEDGEMENT

"This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ICAN(ICT Challenge and Advanced Network of HRD) program(IITP-2024-2020-0-01825) supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation)"

REFERENCES

[1] S. B. Lee and J. S. Rhee, "Applying Batch Normalization to the MNIST Dataset", Quantitative Bio-Science, vol. 42, no. 2, pp. 133-137, Nov. 2023, DOI: http://doi.org/10.22283/qbs.2023.42.2.133

[2] M. R. Han, S. H. Kim, S. I. Lee, K. H. Kim, and J. B. Kim, "A Study on the AI Based MNIST Handwriting Recognition", Jounal of The Korea Society of Information Technology Policy & Management, vol. 11, no. 6, pp. 1509-1517, Dec. 2019.

[3] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[4] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", Proceedings of the International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 37, pp. 448-456, Feb. 2015, DOI: https://doi.org/10.48550/arXiv.1502.03167

[5] M. G. Ji, J. C. Chun, and N. G. Kim, "An Improved Image Classification Using Batch Normalization and CNN", Journal of Internet Computing and Services, vol.19, no.3, pp. 35-42, June. 2018, DOI : 10.7472/jksii.2018.19.3.35

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Communications of the ACM, vol. 60, no. 6, pp. 84-90, May 2017, DOI:http://dx.doi.org/10.1145/3065386

[7] S. H. Park, Y. S. Choi, and H. J. Lee, "CNN-Based Linux Malware Detection Using XAI", Journal of Korea Multimedia Society, vol. 27, no. 4, pp. 514-523, April. 2024, DOI: http://doi.org/10.9717/kmms.2024.27.4.514

[8] K. S. Jang, "Rotation-Invariant Handwritten Numeral Recognition Using Convolution Neural Network", Journal of the Korea Institute of Information and Communication Engineering, vol. 27, no. 12, pp. 1501-1507, Dec. 2023, DOI: http://doi.org/10.6109/jkiice.2023.27.12.1501

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[10] S. Kim and S. Choi, "Development and Application of an Artificial Intelligence Education Program for Elementary School Students: Focusing on CNN", The Korean Association Of Computer Education, vol. 27, no. 3, pp. 43-54, May. 2024. DOI:https://doi.org/10.32431/kace.2024.27.3.004

[11] K. Saitoh, Deep Learning from Scratch, 1st ed., 12th printing. Seoul: Hanbit Media, Inc., Jan. 2017.

[12] J. S. Kim and D. S. Kang, "A Study on the Improvement of MNIST Numbers Learning Speed by Changing the Weight of the Feature Map", Journal of Korean Institute of Information Technology, vol.18, no.3, pp. 109-116, Mar. 2020. DOI:http://dx.doi.org/10.14801/jkiit.2020.18.3.109

Authors

Seungbin Lee received the B.S. degree in Smart Convergence Security from Busan University of Foreign Studies, Korea, in 2023. Seungbin Lee entered the graduate school at Busan University of Foreign Studies in 2023 as a Master's student. His research interests include artificial intelligence and information security.

Jungsoo Rhee received the B.S. degree in Mathematics Education from Kyungpook National University, Korea, in February 1982, the M.S. degree in Mathematics from Kyungpook National University, Korea, in February 1984, and the Ph.D. degree in Mathematics from Florida State University, USA, in August 1993. Dr. Rhee joined the faculty of the Department of Smart Convergence Security at Busan University of Foreign Studies, Korea, in 1994. He is currently a Professor in the Department of Smart Convergence Security at Busan University of Foreign Studies. His research interests include AI/Cybersecurity, Quantum information science, Cryptography, and Fourier Analysis.