

특징 맵 가중치 변경에 따른 MNIST 숫자인식 학습속도 개선에 관한 연구

김종식*, 강대성**

A Study on the Improvement of MNIST Numbers Learning Speed by Changing the Weight of the Feature Map

Jong-Sik Kim*, Dae-Seong Kang**

이 논문은 2017 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.2017R1D1A1B04030870)

요 약

본 논문에서는 CNN Convolution 필터 사용 이후 특징 맵들의 중복적 영향을 분석하고, MNIST 필기체 숫자 학습속도 개선을 위해 다음 3가지 방법으로 연구하였다. 첫째는 특징점 중복이 적은 외부 사이트에 가중치를 크게 주었을 때 인식을 변화, 둘째는 외부 사이트에 가중치를 크게 준 상태에서 필터 수 감소 시 학습속도개선 정도를, 셋째는 중복적 영향이 적은 외부 Line을 연산 제외 시 학습속도 개선에 관해 연구하였다. 그 결과 특징점 중복이 적은 외부 Line에 가중치를 주었을 때 숫자 인식을 향상은 미미하였으며, 동일한 가중치 상태에서 필터 수를 반으로 감소시키고 인식률은 변화가 없을 때 처리속도는 약 12% 정도 향상되었다. 그리고 특징점 중복이 적은 외부를 연산 처리에서 제외했을 때 인식을 변화 없이 학습속도는 약 25% 정도 개선됨을 증명하였다.

Abstract

In this paper, the redundant effects of feature maps after the use of CNN Consolidation filters were analyzed, and the following three methods were studied to improve the learning speed of MNIST cursive numbers. First, we studied changes in recognition rate when the external side with less overlap of features was heavily weighted, secondly the degree of improvement in learning speed when reducing the number of filters with a large weighting on the external side, and thirdly, the external line with less redundant impact was studied for improvement in learning speed when excluding computations. As a result, the numerical recognition rate improvement was minimal when the external line with less feature-point redundancy was weighted, and the processing speed was improved by about 12% when the filter count was halved under the same weight and there was no change in the recognition rate. And when the external with less feature-point redundancy are excluded from the computational processing, the learning speed has proven to be improved by about 25% without a decrease in recognition rate.

Keywords

CNN, convolution, deep learning, feature map, weight

* 동아대학교 전자공학과 석사과정
- ORCID: <https://orcid.org/0000-0002-1459-1943>
** 동아대학교 전자공학과 교수(교신저자)
- ORCID: <https://orcid.org/0000-0003-0186-2430>

• Received: Dec. 24, 2019, Revised: Mar. 17, 2020, Accepted: Mar. 20, 2020
• Corresponding Author: Dae-Seong Kang
Dept. of Dong-A University, 37 NaKdong-Daero 550, beon-gil saha-gu,
Busan, Korea,
Tel.: +82-51-200-7710, Email: dskang@dau.ac.kr

1. 서 론

최근 딥러닝 기법의 하나인 CNN은 영상인식 분야에서 탁월한 성능을 보인다. CNN은 그림 1과 같이 이미지 특징점을 추출하는 컨볼루션 레이어(Convolution layer), 풀링 레이어(Pooling layer)와 추출된 특징점을 이용하여 분류목적으로 사용되는 Fully-connected layer로 구성된다[1]-[3]. 한 이미지에서 전체 가중치를 얻고 단일(소수) 필터만 사용하여 특징점을 추출하는 전통적 알고리즘보다[4], CNN은 컨볼루션 계층에 영상 데이터를 직접 입력하고 다층으로 구성된 여러 종류의 필터에서 자동적 필터 생성 방식으로 특징이 자동 추출되는 기능으로 구성되어 있다.[5]

CNN은 다층 및 여러 종류의 필터를 이용한 학습에 의해서 이미지 인식 등에서 뛰어난 성능을 인정받고 있으며, 층이 깊어질수록 정확도는 더욱 높아진다. GPU 등 연산 처리 H/W 환경의 발달로 층과 필터 수를 크게 늘린 모델이 성능 향상을 주도하고 있으며[6][7], 층의 규모가 커질수록 CNN 시스템의 최적화는 어렵고, 상대적으로 큰 연산량을 요구하게 된다. 지금까지 대부분의 연구는 GPU의 성능 향상에 따른 GPGPU의 병렬처리나, CNN 층에서 필터 수를 줄이므로 네트워크의 용량을 소형화하는 방식으로 성능을 개선하고 있다[8].

제안된 방법은 컨볼루션 필터 사용 후 발생하는 특징 맵들이 필터에 따른 많은 중복적 영향을 분석하고, 중복적 영향을 적게 받는 특징점(외부 라인)에는 가중치를 많이 주었을 때 필터 수와 가중치 값에 따라 MNIST 필기체 숫자 인식 학습속도 및

정확도가 어떻게 차이 나는지에 대해 연구하였다. 그리고 특징점에 영향을 적게 주는 외부를 제외하고 연산을 진행하였을 때 인식률 감소 없이 학습처리 시간은 어떻게 변하는지에 대해 연구하였다.

II. 컨볼루션 관련 이론

2.1 컨볼루션 시 특징 맵 가중치

컨볼루션은 필터를 일정 간격으로 이동하며 특징 맵을 추출한다. 만일 이미지 크기가(7×7), 필터 크기가 (3×3), 스트라이드(Stride)가 1이라 가정했을 때 그림 2에는 컨볼루션 이후 특징 맵(Feature map) 출력 결과를 보인다.

컨볼루션은 연산 특성상 중앙 라인 부분이 특징 맵에 영향을 많이 주고 외부 라인은 특징 맵에 영향을 적게 줄 수밖에 없는 구조적 한계를 가지고 있다. 이것을 극복하기 위해 확장된(Dilated) 컨볼루션에서는 중복요소를 최소화하는 기법을 적용하고 있다[9][10]. 하지만 대부분의 컨볼루션은 중복적 요소를 활용하여 인식률을 향상하는 데 활용하고 있다.

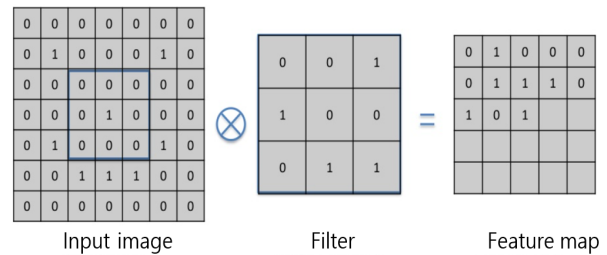


그림 2. 합성곱 연산

Fig. 2. Convolution operation

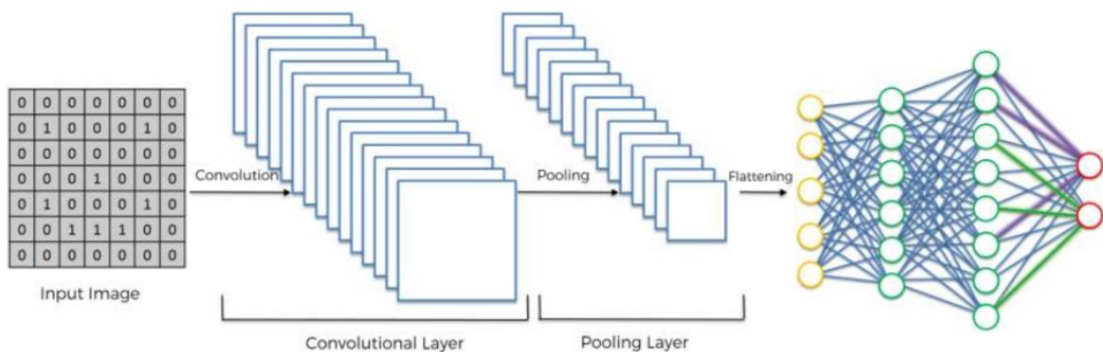


그림 1. CNN의 구조

Fig. 1. CNN model structure

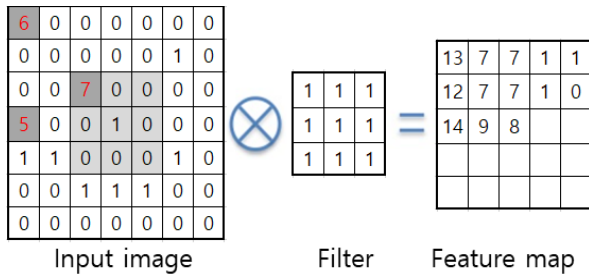


그림 3. 특징점 가중치
Fig. 3. Weight of feature points

이로 인하여 학습 시간이 길어지는 원인이 된다. 그림 3은 입력 이미지에서 중간 부근의 입력 이미지 데이터 7과 외부 이미지 데이터 5, 6이 특징점에 미치는 영향에 대해서 도식화하였다. 구성 조건은 이미지(7×7), 필터(3×3)에 데이터는 모두 1, 스트라이드가 1이라 가정을 하면, 그림 3과 같이 특징 맵이 출력된다. 외부에 있는 이미지 입력 데이터 5, 6의 경우는 특징 맵 추출 시 특징점 출력 데이터 14와 13에 각각 한번 영향을 주지만 이미지 입력 데이터 7은 9번 특징점에 영향을 주는 것을 알 수 있다. 즉 대다수 외부 입력 데이터는 특징점에 별다른 영향을 주지 못하고 중앙 부분에 있는 값만 특징점에 절대적 영향을 주는 것을 알 수 있다.

그림 4에서는 이미지 사이즈가 (9×9), 필터 크기가 (3×3), 스트라이드가 1일 때 필터가 입력 이미지 데이터 각 라인 별로 특징 맵에 어떠한 영향을 주는지를 계산하였다. 그림 4의 수평 라인은 H1 ~ H9으로, 수직은 W1 ~ W9으로 정하였다. H1라인, H9라인, W1 라인, W9 라인인은 (3×3) 필터에 스트라이드가 1일 경우 1회 ~ 3회만 특징 맵에 영향을 미치고, H2 라인, H8 라인, W2 라인, W8 라인은 4회 ~ 6회 특징 맵에 영향을 끼친다. 하지만 H3 ~ H7 라인, W3 ~ W7 라인은 특징 맵에 9회에 걸쳐 영향을 주어 내부 라인은 외부에 비해 특징 맵에 최대 영향을 주는 것을 알 수 있다.

즉, 외부(H1, H9, W1, W9) 라인은 72회 정도 특징 맵에 영향을 미치며, 내부(H2, H8, W2, W8) 라인은 136회 특징 맵에 가중치를 가지며, 중앙(H3 ~ H7, W3 ~ W7) 라인은 225회 특징 맵에 영향을 주어 중앙 부분일수록 특징 맵에 미치는 가중치가 크다는 것을 알 수 있다.

	W1	W2	W3	W4	W5	W6	W7	W8	W9
H 1	1	2	3	3	3	3	2	1	
H 2	2	4	6	6	6	6	4	2	
H 3	3	6	9	9	9	9	6	3	
H 4	3	6	9	9	9	9	6	3	
H 5	3	6	9	9	9	9	6	3	
H 6	3	6	9	9	9	9	6	3	
H 7	3	6	9	9	9	9	6	3	
H 8	2	4	6	6	6	6	4	2	
H 9	1	2	3	3	3	3	2	1	

Image size(9X9), Filter size(3X3)
 ■ Outside Line affects feature map 72 times
 ■ Internal Line affects feature map 136 times
 ■ Center Line affects feature map 225 times

그림 4. 라인 별 특징점 가중치
Fig. 4. Effects of line on feature points

2.2 필터 크기에 따른 특징 맵 가중치

표 1, 2는 컨볼루션 필터의 크기는 (3×3)과 (5×5)로 크기를 각각 정하고 이미지는 MNIST에서 제공하는 (28×28)사이즈로 하였을 때 필터가 특징 맵에 영향을 미치는 가중치를 분석하였다. 그 결과 필터 사이즈에 따라 외부의 영향이 5%(필터 3×3), 3%(필터 5×5) 정도 영향을 준다는 것을 알 수 있었다. 컨볼루션은 연산 구조상 중앙 라인 부분이 특징 맵에 가장 많은 영향을 주고 외부 라인은 특징 맵에 영향을 적게 줄 수밖에 없는 구조적 한계를 가지고 있다. 대부분의 이미지는 중요 사물이나 글자가 중심을 차지하고 있어 외부의 중요성은 크게 없어 보인다. 하지만 인식률을 올리기 위해서는 외부 라인의 특징점이 중요하고, 학습속도 향상을 위해서는 외부 라인이 그렇게 중요해 보이지는 않는다.

표 1. 필터 크기가(3×3)일 때 특징 맵 가중치
Table 1. Feature map weight with filter size (3×3)

MNIST(28×28)	Outside	Inside	Center	Total
Filter Weight	312	592	5184	6088
(3×3) (%)	5%	10%	85%	100%

표 2. 필터 크기가(5×5)일 때 특징 맵 가중치
Table 2. Feature map weight with filter size (5×5)

MNIST (28×28)	Outside	Inside (1)	Inside (2)	Inside (3)	Center	Total
Filter Weight	440	856	1284	1664	10,000	12948
(5×5) (%)	3%	7%	10%	13%	77%	100%

표 1에는 필터 크기가 3×3일 때 각 라인은 특징 맵 얼마나 영향을 주는지를 표시하였다. 그 결과 외부 라인은 특징 맵 가중치에 5%의 영향을 주고, 내부 라인은 특징 맵 가중치에 10%의 영향을 주고, 중앙 부분은 85%가 특징 맵 가중치에 영향을 주는 것을 알 수 있었다. 그림 7에는 각 셀 단위로 중복적 영향을 도식화했다.

표 2는 필터 크기가 5×5일 때 각 라인은 특징 맵 얼마나 영향을 주는지를 표시하였다. 필터가 커지면 그와 비례하게 각 라인이 세분화 되게 영향을 주는 것을 알 수 있으며 학습 개수도 필터(3×3)에서 6088 번인 것이 필터(5×5)에서는 12948로 약 50% 정도 학습 시간이 더 필요함을 알 수 있다. 그리고 외부 라인은 특징 맵 가중치에 3%의 영향을 주고, 내부(1) 라인은 특징 맵 가중치에 7%의 영향을 주고, 내부(2) 라인은 특징 맵 가중치에 10%의 영향을 주고, 중앙 부분은 77%가 특징 맵 가중치에 영향을 주는 것을 알 수 있다.

III. 제안하는 방법

대부분 이미지 구조상 중앙에 사물이 있고 (MNIST 경우는 숫자) 컨볼루션 연산 특성도 중앙에만 약 77% ~ 85% 정도 중복적 영향을 주어 다음과 같이 3가지 방안으로 연구를 진행하였다. 첫 번째 안은 외부 라인에 가중치 값을 중앙과 비례하게 가중치를 주었을 때 컨볼루션 인식 정확도의 변화를, 그리고 두 번째는 외각 라인에 가중치를 첫 번째처럼 유지한 상태에서 필터 숫자를 줄였을 때 연산 처리 속도의 개선을 연구하였다. 그리고 세 번째 안은 특징점에 영향을 많이 주는 중앙 라인만 연산 처리를 하였을 때 인식 정확도 및 학습 시간을 기존과 비교 연구하였다.

3.1 외부 라인 가중치 변경

필터 크기가 (3×3)일 때 내부와 외부 라인은 15%, 필터 크기가 (5×5)일 때 내부와 외부 라인은 33% 정도 특징 맵에 영향을 주어 이번 연구에서는 내부와 외부 라인에 영향과 반비례한 임의의 가중

치를 특징 맵에 값을 주었을 때 인식률과 학습 시간의 개선 정도를 분석하였다. 표 3은 특징 맵에 임의의 가중치를 외부 라인에 10 또는 20, 내부 라인에는 5를 곱했을 때 특징점 변화에 대한 예시를 나타내고 있다. 중앙은 가중치를 추가하지 않았다.

표 3. 필터 크기(3×3)일 때 가중치 설정값(예시)
Table 3. Weight setting value at filter size (3×3)

MNIST(28×28)	Outside	Inside	Center	Total
(3X3) (%)	5%	10%	85%	100%
Weight(1)	10	5	0	15
Weight(2)	20	5	0	25

그림 5에서 상단 그림은 가중치를 주지 않았을 경우 일반적 연산 예를 표시한 것이며, 하단 그림은 특징 맵에 가중치를 외부와 내부로 구분하여 진행하였을 때 입력 이미지에 가중치가 더해진 값과 그 결과 연산이 완료된 특징 맵을 보인다. 연산 방식은 현재 데이터 값이 0보다 크면서 이미지가 외부에 있으면 10을 곱하고 내부에 있으면 5를 곱하는 방식으로 알고리즘을 구현하였다.

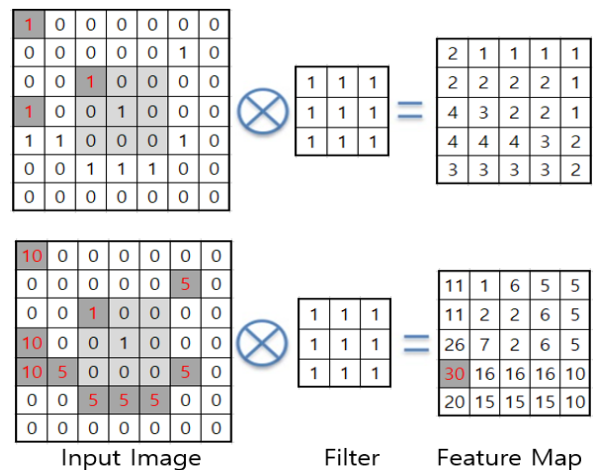


그림 5. 가중치(10,5) 추가 시 특징점
Fig. 5. Feature points when adding weights(10,5)

3.2 외부 라인 가중치에 따른 필터 수

3.1절의 연속적 검토 건으로 외부 라인 가중치 변경 시 사용되는 필터의 수량을 컨볼루션 2에서 16개, 10개, 8개로 변경 시 필터 수량 감소에 따른 MNIST 번호 인식 정확도와 학습 시간 변화를 검토

하였다. 즉 컨볼루션 2의 특징 맵에 외부와 내부에 가중치(10, 5)를 준 상태에서 인식을 변화가 없는 상태까지 필터 수를 줄였을 때 학습 시간이 어느 정도 단축이 가능한지를 연구하였다.

그림 6은 컨볼루션 필터 수를 이미지화 한 것이다. 특징 맵은 필터 수량에 비례하여 출력된다.

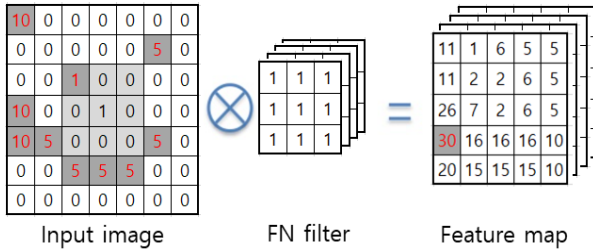


그림 6. 컨볼루션 필터 수
Fig. 6. Convolution filter number

3.3 중앙 부분만 연산 처리

특징 맵에 영향이 적은 외부 라인을 연산에서 제외하고 중앙 부분만 컨볼루션을 수행했을 때 정확도 및 학습 시간을 기존과 비교하였다. 그림 7은 필터(3x3) 사용 시 MNIST(28x28)에서 외부라인이 특징점에 영향을 주는 빈도를 숫자로 표시하였다.

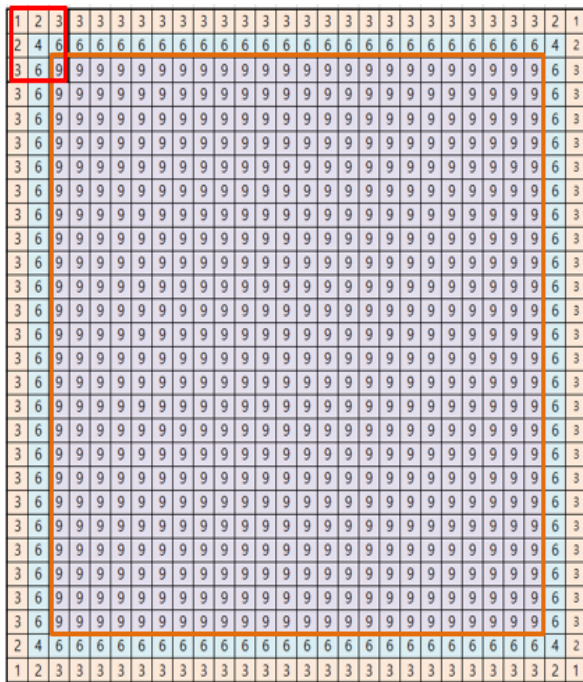


그림 7. MNIST(28x28)에서 필터(3x3) 사용 시 특징점 영향
Fig. 7. Feature point influence on filter(3x3) in MNIST(28x28)

외부의 경우는 최대 3회만 특징점에 영향을 주고, 내부 라인의 경우는 특징점에 최대 6회의 영향을 주지만 중앙은 모두 동일하게 9회에 걸쳐 특징점에 영향을 주는 것으로 나타났다. 즉 28 라인과 27 라인을 제외하고 특징점에 영향을 많이(85%) 주는 중앙 라인만 특징점 연산이나 학습 진행 시 상관관계를 분석하였다.

IV. 실험 방법 및 결과

CNN 특징 맵 가중치 실험은 Apple사 Intel Core i5 2.9GHz, RAM 8GB 컴퓨터 환경에서 실험을 수행하였다. 이미지는 MNIST 데이터 셋을 이용하였으며 데이터 개수는 5,000개, 1만 개, 3만 개로 나누어 진행했다. 그리고 입력 이미지 크기는 MNIST에서 제공하는 (28x28)을 이용하였다.

실험에 사용된 파라미터는 다음과 같다.

1. 필터 개수 : Conv1(16개), Conv2(16개/10개)
2. 필터 크기 : Conv1(5x5), Conv2(3x3)
3. Wight 값 : Conv1(0), Conv2(10, 5)
4. Test 수량 : 1,000개(무작위 선정)
5. 네트워크 구성 : Conv1 → Relu → Conv2 → Relu → Pool → Affine → Relu → Dropout → Affine → Dropout → Softmax

표 4에는 컨볼루션 2의 특징 맵 외부와 내부에 가중치 값을 주었을 때(외부에 10, 내부에 5) 인식을 및 학습 시간 변화를 기록했다. 구현 알고리즘은 현재 데이터 값이 0보다 클 경우 이미지가 외부에 있으면 10을 곱하고 내부에 있으면 5를 곱하는 방식으로 알고리즘을 구현하였다. 학습 데이터 수가 5천 개 일 경우 인식을 큰 차이를 보이지 않으나 학습 데이터 수를 증가할수록 데이터 인식이 증가 하였다.

표 4. Conv2 가중치 변경에 따른 결과
Table 4. Result of weight change of Conv2 filter

Conv2 weight	Data	5,000	10,000	30,000
Previous (0, 0)	Accuracy	0.956	0.975	0.982
	Time(s)	560	1050	2936
Change (10, 5)	Accuracy	0.964	0.98	0.99
	Time(s)	570	1069	2940
Change (20, 5)	Accuracy	0.96	0.979	0.989
	Time(s)	568	1072	2936

그림 8은 데이터 수가 3만일 경우 인식률은 98%에서 99%로 약 1%로 인식률 향상에 미미한 결과를 얻었고, 그 결과를 그래프로 표시하였다.

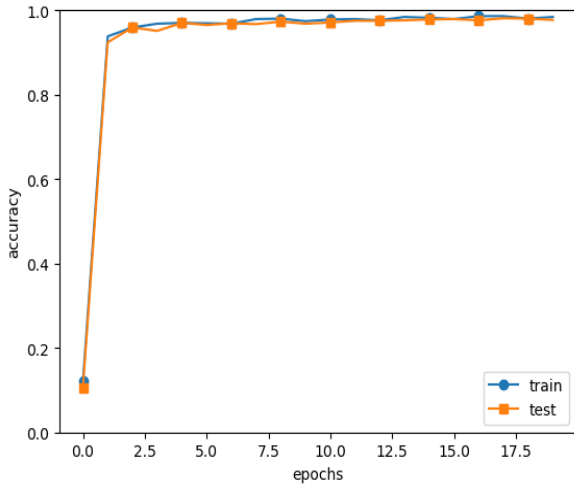


그림 8. 데이터 개수 3만 시 Conv2 weight(10,5) 그래프
Fig. 8. Conv2 weight(10,5) graph in 30,000 data

표 5. Conv2 필터 수 변경에 따른 결과
Table 5. Results of changes in the number of Conv2 filter

Conv2 filter	Data	5,000	10,000	30,000
Previous (16ea)	Accuracy	0.958	0.975	0.98
	Time(s)	660	1169	3218
Change (10ea)	Accuracy	0.964	0.98	0.99
	Time(s)	570	1069	2940
Change (8ea)	Accuracy	0.96	0.97	0.98
	Time(s)	550	1052	2829

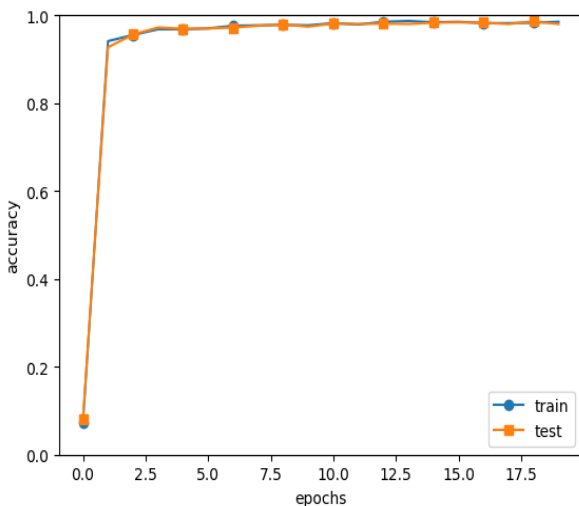


그림 9. Conv2 filter(10개) 그래프(Data 개수 3만)
Fig. 9. Conv2 filter(10) graph in 30,000 data

그리고 가중치 값을 20으로 높인다고 인식률은 크게 개선되지 않았고 외부와 내부를 동일하게 가중치를 주어도 인식률과 학습 시간은 개선되지 않았다. 현재 값은 몇 번의 비교 실험을 통하여 얻은 최적의 값을 결과로 표시하였다.

표 5와 그림 9는 표 4처럼 컨볼루션 2의 특징 맵에 외부와 내부에 가중치(10, 5)를 준 상태에서 인식률이 동일한 상태까지 필터 수를 줄였을 때 학습 시간이 어느 정도 단축이 되는지를 연구하였다. 표 5의 측정 결과처럼 기존의 16개 필터를 사용했을 때와 동일한 98% 인식률이 유지되는 지점까지 필터 수를 줄이면서(10개, 8개 등) 실험한 결과, 필터 수가 반으로 줄었을 때 인식률은 기존과 동일한 98% 수준이며 학습 시간은 데이터 3만 개 기준 약 389초(약 12%) 정도 개선되는 결과를 얻었다. 필터 개수가 반 이하로 떨어지면 학습 시간은 대폭 감소하나 인식 정확도도 같이 저하하는 결과가 나타나 특징 맵 가중치(10, 5)를 준 상태에서 최적의 필터 수 감소는 필터 수를 반(8개)으로 줄였을 때임을 알 수 있었다. Conv1은 영상 입력 부분이라 필터 수를 변동하지 않고 그대로(16개) 사용했다.

컨볼루션은 연산 구조상 중앙 라인이 특징 맵에 가장 많은 영향을 주고 외부 라인은 특징 맵에 영향을 적게 줄 수밖에 없는 구조적 한계를 가지고 있다. 대부분의 이미지나 숫자 또는 중요 사물이나 글자도 이미지의 중심을 차지하고 있다.

표 6에는 컨볼루션 입력 이미지 사이즈가 28×28에서 특징점에 영향이 적은 외부 라인을 연산 처리를 하지 않고 이미지 중앙 부분만 학습 데이터로 활용할 경우 인식률의 정확도와 학습 시간 감소 정도를 연구하였다.

표 6. 중앙 부분 연산처리 결과
Table 6. Center part calculation result

MNIST size	Data	5,000	10,000	30,000
Previous (28×28)	Accuracy	0.956	0.975	0.982
	Time(s)	560	1050	2936
Change (26×26)	Accuracy	0.966	0.974	0.98
	Time(s)	501.8	940.8	2630.7
Change (24×24)	Accuracy	0.959	0.971	0.98
	Time(s)	415.2	774.8	2176.8

필터사이즈가 5×5이고, MIST 입력 이미지 사이즈(28×28)에서 외부 라인(28×28)과 내부 라인(26×26)을 연산 처리 하지 않고 특징점을 추출할 경우 인식을 정확도는 98%를 유지하면서 학습 연산 처리 시간은 3만 개 기준으로 기존대비 약 25% 정도 학습시간이 단축되는 결과를 얻었다.

V. 결 론

실험 결과 CNN의 성능을 유지하면서, 특징 맵의 외부에 가중치를 추가했을 때는 정확도는 기존 과별 차이가 없으나, 외부에 가중치를 추가하고 인식을 98%로 동일하게 유지한 상태에서 필터수를 16개에서 8개로 반으로 줄였을 때 학습시간은 기존 대비 약 12% 정도 학습 시간이 단축되는 결과를 얻었다. 그리고 특징점에 영향을 적게 주는 외부를 제외하고(28×28, 26×26 라인) 연산을 진행하였을 때는 인식률은 기존 98%로 동일하나 학습시간은 기존 대비 약 25% 정도 학습속도가 단축되는 연구결과를 얻었다.

향후는 지금 사용하는 필기체인식 MNIST 데이터 대신 이미지 객체인식 ImageNet 데이터를 활용하여 컨볼루션 특징 맵의 가중치 설정 방법 시 인식을 개선과 특징점에 영향이 적은 외부 Line을 연산 처리에서 제외하였을 때 동일한 인식률에서 학습속도가 개선되는지 추가 연구가 필요하다.

References

[1] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview", *Neural Networks*, Vol. 61, pp. 85-117, Jan. 2015.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning", *Nature*, Vol. 521, pp. 436-444, May 2015.

[3] Yi Sun, Xiaogang Wang, and Xiaoou Tang, "Deep Learning Face Representation from Predicting 10,000 Classes", *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1891-1898, Jun. 2014.

[4] LeCun, Yann, et al., "Gradient based learning applied to document recognition", *Proceedings of the IEEE*, pp. 2278-2324, Nov. 1998.

[5] Lee, Kyung Min and Lin, and Chi Ho, "Video Stabilization Algorithm of Shaking image using Deep Learning", *Journal of IIBC*, Vol. 19, No. 1, pp. 145-152, Feb. 2019.

[6] Se-Jin Choi, Min-Cheol Kim, Jun-Mo Jeong, and Kwang-Yeob Lee, "Accelerating Convolutional Neural Network's training algorithm using CUDA", *Annual Conference of IEIE*, pp. 1265-1267, Jun. 2017.

[7] Yunjoo Park, Donghee Shin, Kyungwoon Cho, and Hyokyung Bahn, "Analyzing Fine-Grained Resource Utilization for Efficient GPU Workload Allocation", *Journal of IIBC*, Vol. 19, No. 1, pp. 111-116, Feb. 2019.

[8] Kisung Seo, "Evolutionary Computation Based CNN Filter Reduction", *The transactions of The Korean Institute of Electrical Engineers*, Vol. 67, No. 12, pp. 1665-1670. Dec. 2018.

[9] Fisher Yu and Vladlen Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions", *Published as a conference paper at ICLR 2016*, arXiv:1511.07122v3, Apr. 2016.

[10] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 40, pp. 834-848, Apr. 2018.

저자소개

김 종 식 (Jong-Sik Kim)



1991년 2월 : 부경대학교
전자공학과(공학사)
2018년 9월 ~ 현재 : 동아대학교
전자공학과 석사과정
관심분야 : 영상처리, AI

강 대 성 (Dae-Seong Kang)



1994년 5월 : Texas A&M 대학교
전자공학과(공학박사)
1995년 ~ 현재 : 동아대학교
전자공학과 교수
관심분야 : 영상처리, 패턴인식