



HS 알고리즘을 이용한 CNN의 Hyperparameter 결정 기법

Method that determining the Hyperparameter of CNN using HS algorithm

이우영* · 고광은* · 김종우** · 심귀보*†

Woo-Young Lee, Kwang-Eun Ko, Zong-Woo Geem and Kwee-Bo Sim†

*중앙대학교 전자전기공학부, **가천대학교 에너지IT학과

*Department of Electrical and Electronics Engineering, Chung-Ang University

**Department of Energy IT, Gachon University

요약

Convolutional Neural Network(CNN)는 특징 추출과 분류의 두 단계로 나눌 수 있다. 그 중 특징 추출 단계의 커널의 크기, 채널의 수, stride 등의 hyperparameter는 CNN의 구조를 결정할 뿐만 아니라 특징을 추출하는 데에도 영향을 주기 때문에 CNN의 전체적인 성능에도 영향을 준다. 본 논문에서는 Parameter-Setting-Free Harmony Search(PSF-HS) 알고리즘을 이용하여 CNN의 특징 추출 단계에서의 hyperparameter를 최적화 하는 방법을 제안하였다. CNN의 전체 구조를 설정한 뒤 hyperparameter를 변수로 설정하였고 PSF-HS 알고리즘을 적용하여 hyperparameter를 최적화 하였다. 시뮬레이션은 MATLAB을 이용하여 진행하였고 CNN은 mnist 데이터를 이용하여 학습과 테스트를 했다. 총 500번 동안 변수를 업데이트했고 제안하는 방법을 이용하여 구한 CNN 구조 중 가장 높은 정확도를 가지는 구조는 99.28%의 정확도로 mnist 데이터를 분류하는 것을 확인할 수 있었다.

키워드 : Convolutional Neural Network, Parameter-Setting-Free Harmony Search Algorithm, 메타 휴리스틱 알고리즘, Hyperparameter 최적화.

Abstract

The Convolutional Neural Network(CNN) can be divided into two stages: feature extraction and classification. The hyperparameters such as kernel size, number of channels, and stride in the feature extraction step affect the overall performance of CNN as well as determining the structure of CNN. In this paper, we propose a method to optimize the hyperparameter in CNN feature extraction stage using Parameter-Setting-Free Harmony Search (PSF-HS) algorithm. After setting the overall structure of CNN, hyperparameter was set as a variable and the hyperparameter was optimized by applying PSF-HS algorithm. The simulation was conducted using MATLAB, and CNN learned and tested using mnist data. We update the parameters for a total of 500 times, and it is confirmed that the structure with the highest accuracy among the CNN structures obtained by the proposed method classifies the mnist data with an accuracy of 99.28%.

Key Words : Convolutional Neural Network, Parameter-Setting-Free Harmony Search Algorithm, Metaheuristic Algorithm, Hyperparameter Optimization.

Received: Jan. 4, 2017
Revised: Feb. 17, 2017
Accepted: Feb. 17, 2017
†Corresponding authors
kbsim@cau.ac.kr

1. 서론

다층신경망(Multi-layer neural network)은 사람이 학습하는 방법을 모방하여 구현한 기계학습 알고리즘인 인공신경망 구조의 일종으로 함수 추론, 패턴 인식, 클러스터링 등의 다양한 분야에서 사용된다. 다층신경망은 입력 층과 출력 층 사이에 하나 이상의 은닉 층을 가지는 구조로 이루어져 있고 입력 데이터와 목표 출력으로 이루어진 학습 데이터를 이용하여 학습하는 지도학습 기반의 학습 모델이다. 이러한 다층신경망은 미리 정해진 규칙을 이용하여 입력 데이터에 대한 신경망의 출력 값과 목표 출력 사이의 오차가 최소가 되도록 가중치를 조정하는 방법으로 학습을 진행한다[1].

다층신경망은 비선형적인 문제를 해결할 수 있고 학습 데이터로부터 원하는 근사 함수를 만들 수 있다는 장점이 있다. 하지만 데이터의 특징을 고려하지 않고 원본 데이터에 대하여 직접적으로 연산하기 때문에 많은 학습데이터를 필요로 하고, 그에 따라 학습 시간이 많이 소요되는 점과 과적합 문제라는 단점이 존재한다. 이러한 다층신경망의 단점을 보완하기 위하여 Convolutional Neural Network(CNN)라는 새로운 신경망 구조가 제안되었다.

본 논문은 본 학회 2016 추계학술대회에서 선정된 우수논문입니다.
본 연구는 한국연구재단 중견연구지원사업(No.2012-0008726)에서 지원하여 연구하였습니다.
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.

CNN은 1990년대에 제안된 신경망 구조로 지도학습 기법을 기반으로 하는 기계학습 모델이다. CNN은 convolution 층, pooling 층, fully-connected 층의 여러 단계로 구성되어 있다[2]. 그 중 convolution 층과 pooling 층은 입력 데이터의 특징을 추출하는 역할을 하고 fully-connected 층은 앞에서 추출된 특징을 분류하는 역할을 한다. CNN의 처리과정에는 다른 알고리즘과 달리 특징을 추출하는 단계가 내부에 포함되어있기 때문에 입력 데이터의 특징을 추출하는 별도의 전처리 과정 없이 직접 연산이 가능하다는 장점이 있다. 또한 데이터의 특징을 이용하여 학습하기 때문에 국부적 특징추출 및 분류에 좋은 성능을 보이고 dropout 기법을 이용하면 과적합 문제도 방지할 수 있기 때문에 이미지 인식 분야에서 널리 사용되고 있다[3, 4].

CNN의 구조를 결정할 때에는 커널의 크기와 stride, 채널의 수 등의 hyperparameter를 정해야한다. 이러한 파라미터들은 CNN의 전체적인 구조를 결정할 뿐만 아니라 학습 시간, 정확도 등의 성능에도 직접적인 영향을 주기 때문에 과적합 문제에 빠지지 않고 기대했던 결과를 얻기 위해서는 hyperparameter에 대한 최적화 작업이 선행되어야 한다. 하지만 아직까지는 hyperparameter를 최적화하는 규칙이 없다는 것이 일반적이고 hyperparameter를 결정할 때 많은 부분을 경험이나 설계자의 직관에 따라 결정해야 하는 상황이다. 실제로 좋은 성능을 보이는 CNN 구조를 살펴보면 Alexnet은 커널의 크기를 11, 5, 3으로 설정하였고 ZFNet은 커널의 크기를 7, 5, 3으로 설계자의 직관에 따라 hyperparameter를 설정하였음을 알 수 있다[2, 5, 6].

본 논문에서는 Parameter-Setting-Free Harmony Search(PSF-HS) 알고리즘을 이용하여 CNN의 특징 추출 단계의 hyperparameter를 결정하는 방법을 제안한다. 실험에서는 CNN 각 층의 구조를 고정한 뒤 커널 필터의 크기와 채널의 수를 변수로 설정하였고, PSF-HS 알고리즘을 이용하여 생성된 변수로 CNN을 학습시킨 뒤 정확도를 기준으로 변수를 업데이트하는 방법으로 시뮬레이션을 진행하였다.

2. 배경 및 관련 연구

2.1 Convolutional Neural Network

CNN은 다층신경망의 한 종류로 패턴 인식 분야에서 좋은 성능을 보이고 있어 이미지, 텍스트 등의 데이터를 분류, 인식하는데 주로 사용되고 있다. 일반적인 다층신경망은 입력 층, 은닉 층, 출력 층으로 이루어진 구조를 가지고 있어 원본 데이터에 대하여 바로 연산하여 출력을 계산하는 반면, CNN은 원본 데이터의 특징을 추출하는 단계와 추출한 특징을 분류하는 단계로 이루어져 있어

원본 데이터에 대하여 바로 연산하지 않고 원본 데이터의 특징을 추출한 뒤 연산한다는 특징을 가지고 있다[7, 8].

일반적으로 CNN은 입력 층, convolution 층, pooling 층, fully-connected 층, 출력 층으로 구성되어 있다. convolution 층과 pooling 층은 원본 데이터의 특징을 추출하는 역할을 하고 fully-connected 층은 특징을 분류하는 분류기의 역할을 한다. 일반적인 다층신경망은 모든 노드에 대하여 계산을 수행하기 때문에 학습시킬 파라미터의 개수가 많지만 CNN은 특징을 추출할 때 convolution 연산과 pooling 연산을 진행하기 때문에 추출되는 특징의 크기가 원본 데이터에 비하여 작고 가중치를 공유하기 때문에 학습시킬 파라미터의 개수가 적다는 장점이 있다. 그림 1은 CNN의 일종인 LeNet-5 구조이다[2].

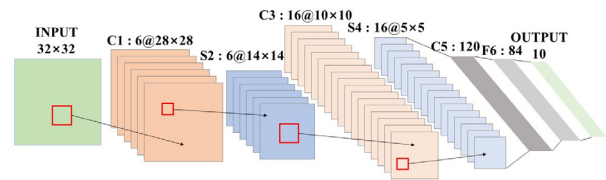


그림 1. CNN 구조의 예시
Fig. 1. Example of CNN structure

그림 1의 C1, C3, C5는 convolution 층으로 입력으로 들어오는 데이터에 5x5크기의 커널 필터를 사용하여 convolution 연산을 하고 입력으로 들어오는 데이터의 특징을 추출하는 역할을 한다. S2, S4층은 pooling 층으로 일반적으로 max pooling 방법을 이용하여 차원을 축소시킨다. convolution과 pooling을 여러 번 반복하여 데이터의 특징을 추출하며 데이터 추출이 끝나는 층 뒤에 분류기를 위치하여 CNN 구조를 완성한다. F6층은 일반적인 다층신경망과 같은 구조로 구성되어 있는 fully-connected 층으로 특징 추출 단계에서 추출된 특징을 분류하는 역할을 한다.

2.2 Harmony Search Algorithm

Harmony search(HS) 알고리즘은 음악의 즉흥 연주에서 연주자들이 최적의 화음을 찾아내기 위하여 더 좋은 하모니를 찾는 방법을 모방한 메타 휴리스틱 알고리즘이다. HS 알고리즘은 무작위하게 생성된 변수가 저장되는 공간인 하모니 메모리에 변수를 생성하고 업데이트 하는 방법으로 최적의 해를 찾는다. HS 알고리즘에서 변수를 생성하는 방법에는 무작위로 변수를 생성하는 랜덤선택, 하모니 메모리의 변수를 사용하는 기억 회상, 하모니 메모리의 변수를 가져와 변형하는 피치 조정이 있다. HS알고리즘은 초기 하모니 메모리를 생성하고 새로운 해를 생성하여 기존의 해와 성능 비교를 통하여 하모니 메모리를 업데이트하는 방법으로 최적화를 진행한다. HS 알고리즘은 초기 수렴 안정성, 지역 해 문제

등에 우수하다고 알려져있다. HS 알고리즘의 순서도는 그림 2와 같다[9, 10, 11, 12].

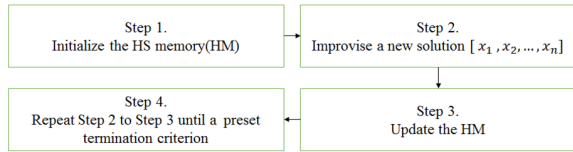


그림 2. HS 알고리즘의 순서도
Fig. 2. Flow chart of HS algorithm

기억 회상을 사용할 확률(Hammony Memory Considering Rate, HMCR)과 피치 조정을 사용할 확률(Pitch Adjusting Rate, PAR)은 변수를 생성하는데 영향을 주기 때문에 최적의 해를 찾는 성능에 영향을 준다. 따라서 HS 알고리즘을 사용할 때 HMCR과 PAR을 정하는 문제는 매우 중요하다. PSF-HS 알고리즘은 HMCR과 PAR을 자동으로 결정하기 위하여 제안된 개념이다. 일반적인 HS 알고리즘은 HMCR과 PAR을 고정된 뒤 최적화를 진행하지만 PSF-HS 알고리즘은 초기 HMCR과 PAR을 바탕으로 구성된 하모니 메모리를 분석한 다음 기억 회상과 피치 조정을 사용한 횟수를 하모니 메모리의 크기(Hammony Memory Size, HMS)로 나눠 HMCR과 PAR을 업데이트하며 최적화를 진행한다. 식 (1)과 식 (2)는 각 PSF-HS 알고리즘에서 HMCR과 PAR을 업데이트하는 식을 나타낸다[13].

$$HMCR_i = \frac{n(y_i^j = Memory)}{HMS} \quad (1)$$

$$PAR_i = \frac{n(y_i^j = Pitch)}{HMS} \quad (2)$$

3. PSF-HS 알고리즘을 이용한 CNN의 hyperparameter 결정 방법

CNN의 convolution 연산을 할 때 커널의 크기와 stride는 입력 데이터의 특징을 추출하는 연산을 하는 픽셀의 범위를 결정하기 때문에 데이터의 특징 계산에 영향을 주고 convolution 연산을 하고 난 바로 다음 층에 생성되는 feature map의 크기에 영향을 준다. 식 (3)은 feature map의 크기를 계산하는 공식이다.

$$f = \frac{i - k}{s} + 1 \quad (3)$$

식 (3)에서 f는 feature map의 크기를 나타내고 i는 입력 데이터의 크기이다. 그리고 k는 커널의 크기이고 s는 stride를 나타낸다.

채널의 수는 feature map의 깊이이므로 커널의 크기, stride, 채널의 수등의 hyperparameter는 각 층에서의 학습시킬 파라미터의 개수를 결정하고 특징을 추출하는 데에 영향을 준다는 것을 알 수 있다. CNN은 특징 추출 단계에서 특징을 추출한 뒤 특징을 분류하는 분류기이기 때문에 hyperparameter를 결정하는 것은 매우 중요하다.

우리는 CNN의 특징 추출 단계에서의 hyperparameter들을 최적화하기 위하여 PSF-HS 알고리즘을 사용하였다. 제안하는 방법의 순서도는 그림 3과 같다.

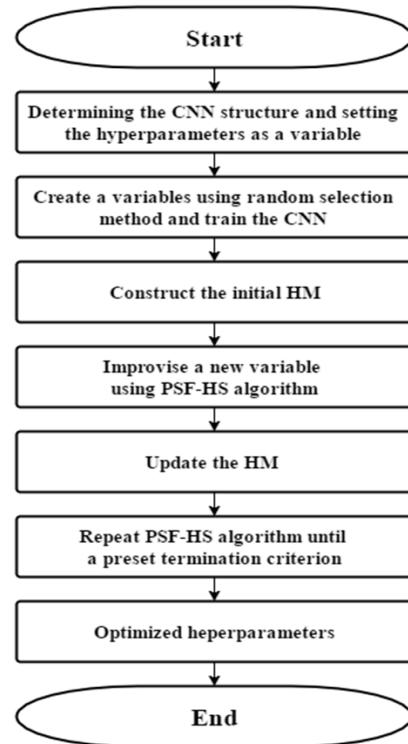


그림 3. 제안하는 방법의 순서도
Fig. 3. Flow chart of proposed method

CNN의 특징 추출 단계에서의 적절한 hyperparameter를 구하기 위하여 제안하는 방법의 첫 번째 단계는 CNN의 각 층에서 어떤 연산을 할 것인지 미리 정하고 hyperparameter를 변수로 설정하는 것이다. 다음 단계는 랜덤 선택 방법으로 변수를 생성하여 CNN을 학습시키고 초기 하모니 메모리를 구성하는 것이다. 이 때, 하모니 메모리는 각 변수들과 변수들을 이용하여 학습한 CNN의 정확도로 구성한다. 초기 하모니 메모리를 구성한 뒤에는 일정 반복 동안 설정해준 초기 HMCR과 PAR에 따라 랜덤 선택, 기억 회상, 피치 조정을 이용하여 변수를 생성한 뒤 CNN을 학습시킨다. 다음으로 새로운 변수로 생성한 CNN 구조를 하모니 메모리에 CNN 구조 중 가장 정확도가 낮은 구조와 비교하여 정확도가 더 높은 CNN 구조를 하모니 메모리에 저장하는 방식으로 하모니 메모리를 업데이트 한다. 일정 반복이 지나면 종료조건을 만족할 때 까지 식 (1)과 식

(2)를 이용하여 HMCR과 PAR을 업데이트 하며 새로운 변수를 생성하고 하모니 메모리에 있는 CNN 구조 중 가장 낮은 정확도가 낮은 구조와 새로운 변수를 이용하여 만든 CNN 구조의 정확도를 비교하여 하모니 메모리를 업데이트하는 방법으로 hyperparameter를 최적화 한다.

4. 시물레이션 방법과 결과

실험에서는 PSF-HS 알고리즘을 이용하여 CNN의 특징 추출 단계의 커널의 크기와 채널의 수가 CNN의 성능에 어떠한 영향을 주는지 알아보기 위하여 제안하는 방법을 이용하여 실험을 진행하였다. 먼저 LeNet-5구조를 참고하여 전체적인 CNN의 층 구조를 고정하였다. 다음으로 convolution 연산을 위한 커널의 크기를 변수 k_1, k_2, k_3 으로 설정하였고 각 층의 차원인 채널의 수를 변수 c_1, c_2 로 설정하였다. 실험에서 사용한 CNN 구조는 그림 4와 같다.

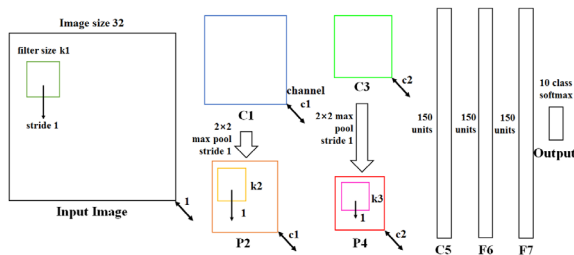


그림 4. 실험에서 사용한 CNN의 구조
Fig. 4. CNN structure in experiment

실험에서 사용한 CNN의 입력 데이터의 크기는 $32 \times 32 \times 1$ 로 설정하였다. 입력 데이터에 $k_1 \times k_1$ 크기의 커널을 이용한 convolution 연산을 통하여 C1 층을 생성하였고 C1층에 2×2 max pooling을 이용하여 P2 층을 생성하였다. C1 층과 P2 층의 채널의 수는 변수 c_1 로 설정하였다. 다음으로 P2층에 $k_2 \times k_2$ 크기의 커널을 이용한 convolution 연산을 통하여 C3 층을 생성하였고 C3 층에 2×2 max pooling을 이용하여 P4 층을 생성하였다. P4에서 C5로 넘어갈 때에는 C5 층이 $1 \times 1 \times 150$ 크기의 벡터로 생성되도록 k_3 를 생성하였다. fully-connected 층인 F6는 $1 \times 1 \times 150$ 크기로 고정하였고 10개의 클래스로 분류되도록 출력 층을 설정하였다. 활성화함수로는 softmax를 사용하였다. 식 (4)는 softmax 함수를 나타낸다.

$$p_j = \frac{e^{x_j}}{\sum_{k=1}^K e^{x_k}} \quad \text{for } j = 1, 2, \dots, K \quad (4)$$

식 (5)에서 p_j 는 j번째 노드의 출력이고 이는 확률로 나타낸다. x_j 는 j번째 노드의 입력을 나타낸다.

하모니 메모리의 크기는 20으로 설정했고 랜덤선택 방법으로 하모니 메모리의 초기 변수 k_1, k_2, k_3, c_1, c_2 를 생성하였다. 다음으로 초기 변수와 mnist 학습 데이터 50,000장을 이용하여 CNN을 학습시켰고 mnist 테스트 데이터 10,000장을 이용하여 정확도를 구하였다[14]. 정확도는 정확하게 분류된 테스트 데이터의 개수를 전체 테스트 데이터의 개수로 나누어서 구하였다. k_1, k_2, k_3, c_1, c_2 , 정확도로 초기 하모니 메모리를 구성하였다.

PSF-HS 알고리즘을 이용하여 변수를 생성할 때 k_1 의 크기는 32보다 작은 홀수로 생성되도록 설정하였다. k_2 의 범위는 입력 이미지에 P2의 크기보다 작아야하기 때문에 식 (5)범위의 홀수로 설정하였다.

$$k_2 < \frac{32 - k_1 + 1}{2} \quad (5)$$

그리고 C5 layer가 $1 \times 1 \times 150$ 크기의 벡터로 생성되도록 k_3 를 생성하였다. 식 (6)은 C5 층을 1×1 크기로 생성하기 위한 k_3 의 값을 나타낸다.

$$k_3 = \frac{\left(\frac{32 - k_1 + 1}{2} - k_2 + 1 \right)}{2} \quad (6)$$

채널의 수 c_1, c_2 는 1보다 크고 100보다 작은 범위에서 생성되도록 설정하였다. 초기 하모니 메모리를 생성한 뒤에는 PSF-HS를 이용하여 하모니 메모리를 업데이트하는 과정을 500번 반복하였다. 시물레이션을 위하여 Intel Core i7-4790K CPU @ 4.0 GHz, 8 GB RAM memory, NVIDIA GeForce GTX 970 그래픽카드를 장착하고 Kubuntu 14.04.4 LTS 운영체제를 사용하는 PC를 사용하였다. vcnn double-bladed 라이브러리를 이용하여 코드를 구현하였고 MATLAB R2015b을 이용하여 시물레이션을 진행하였다[15]. 표 1은 초기에 생성된 하모니 메모리이다.

초기 하모니 메모리의 각 변수를 이용하여 CNN의 구조를 생성한 뒤 정확도를 확인해본 결과 20개의 CNN은 97.47%~99.00%의 정확도로 mnist 테스트 데이터를 분류하는 것을 확인할 수 있었다. 또한 초기 하모니 메모리를 구성할 때 생성된 변수들은 일정 값에 수렴하지 않고 넓은 범위에서 생성되는 것을 확인할 수 있었다.

초기 하모니 메모리를 생성하고 난 뒤에 하모니 메모리의 크기의 3배인 60번 반복할 동안은 HMCR=0.5, PAR=0.5로 설정하여 기본적인 HS 알고리즘을 진행하였고 그 이후에는 종료조건을 만족할 때 까지 PSF-HS 알고리즘을 진행하였다. 종료조건은 변수를 500번 업데이트하는 것 또는 정확도가 99.5% 이상인 구조를 찾는

표 1. 초기 하모니 메모리
Table 1, Initial HM

	Kernel size of CNN			Number of channels		ACC
	k1	k2	k3	c1	c2	
1	13	3	4	12	25	0.9833
2	5	7	4	47	62	0.9900
3	9	9	2	28	29	0.9855
4	5	3	6	37	62	0.9909
5	13	7	2	4	54	0.9747
6	13	3	4	37	60	0.9872
7	9	9	2	25	43	0.9876
8	17	5	2	18	49	0.9769
9	13	7	2	16	69	0.9881
10	5	3	6	37	65	0.9899
11	13	7	2	42	52	0.9858
12	9	5	4	20	48	0.9900
13	13	3	4	48	69	0.9888
14	5	3	6	1	60	0.9820
15	5	3	6	28	56	0.9896
16	9	5	4	10	36	0.9866
17	9	9	2	47	62	0.9869
18	5	7	4	18	33	0.9883
19	9	5	4	30	53	0.9891
20	13	3	4	5	55	0.9803

것으로 설정하였다. 시뮬레이션 결과 변수를 500번 업데이트하고 시뮬레이션이 종료되었다. 표 2는 업데이트된 하모니 메모리를 나타낸다.

표 2. 업데이트된 하모니 메모리
Table 2, Updated HM

	Kernel size of CNN			Number of channels		ACC
	k1	k2	k3	c1	c2	
1	5	3	6	38	68	0.9915
2	5	3	6	37	66	0.9912
3	9	5	4	35	63	0.9914
4	9	5	4	42	68	0.9912
5	5	3	6	36	66	0.9915
6	5	3	6	35	59	0.9922
7	5	7	4	36	60	0.9919
8	9	5	4	44	65	0.9917
9	5	3	6	42	55	0.9912
10	5	3	6	43	55	0.9911
11	9	5	4	41	63	0.9911
12	9	5	4	42	66	0.9911
13	9	5	4	29	59	0.9911
14	9	5	4	43	62	0.9912
15	9	5	4	25	60	0.9928
16	9	5	4	16	63	0.9915
17	9	5	4	39	55	0.9914
18	9	5	4	37	68	0.9911
19	9	5	4	43	61	0.9918
20	9	5	4	42	62	0.9912

PSF-HS를 이용하여 HM을 500번 업데이트한 결과 99.11%-99.28%의 정확도로 모든 CNN구조가 초기에 생성된 20개의 CNN 구조보다 더 정확하게 mnist 테스트 데이터를 분류하는 것을 확인할 수 있었다. 표 2에 빨간색으로 표시한 CNN 구조에서 커널의 크기가 k1=9, k2=5, k3=4의 값으로 수렴하고 있는 것을 확인하였다. 또한 채널의 수도 c1은 대부분 40 근처의 범위에서 생성되었고 c2

는 60 근처의 범위에서 생성되는 것을 확인하였다. 업데이트 된 하모니 메모리에서 가장 높은 정확도를 가지는 구조는 k1=9, k2=5, k3=4, k1=25, k2=60일 경우였고 99.28%의 정확도로 mnist 테스트 데이터를 분류하는 결과를 확인할 수 있었다. 그림 5는 제안하는 방법으로 구한 CNN 구조 중 가장 높은 정확도를 가지는 CNN 구조이다.

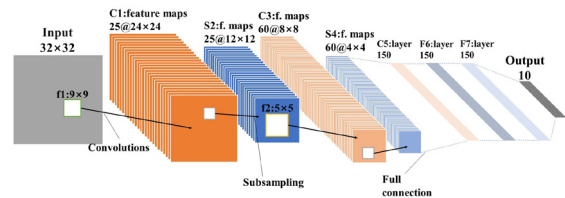


그림 5. 제안하는 방법으로 구한 CNN 구조 중 가장 높은 정확도를 가지는 CNN 구조
Fig. 5. The one which has the highest accuracy among the structure of CNN obtained by the proposed method

5. 결론

본 논문에서는 PSF-HS를 이용하여 CNN의 특징 추출 단계에서의 적절한 커널의 크기와 채널의 수를 구하는 방법을 제안하였다. 최적화하고자 하는 hyperparameter를 변수로 설정한 뒤 제안하는 방법을 이용하여 업데이트 과정을 거치면 변수들이 일정 범위 내로 수렴하려는 현상을 확인할 수 있었다.

향후 연구에서는 mnist 데이터 이외에도 상대적으로 정확도가 낮게 분류되는 데이터를 이용하여 추가 실험을 진행할 것이다. 또한 stride까지 포함한 hyperparameter가 CNN의 분류 정확도에 어떠한 영향을 주는지 확인하기 위하여 입력 데이터의 크기가 더 큰 CNN 구조에 stride까지 변수로 설정한 뒤 CNN의 특징 추출 단계에서의 hyperparameter를 결정하는 방법과 CNN 층 구조를 결정하는 방법에 대하여 연구를 진행할 예정이다.

References

[1] G. S. Choi, C. Yu, R. M. Jin, S. K. Yu and M. G. Chun, "Short-term water demand forecasting algorithm using AR model and MLP," *Journal of Korean Institute of Intelligent Systems*, vol. 19, no. 5, pp.713-719, 2009.

[2] 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, 1998.

[3] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929-1958, 2014.

[4] J. H. Yu and K. B. Sim, "Face Classification Using Cascade Facial Detection and Convolutional Neural Network," *Journal of Korean Institute of Intelligent Systems*, vol. 26, no. 1, pp. 70-75, 2016.

[5] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks." In *Advances in neural information processing systems*, pp. 1097-1105, 2012.

[6] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," In *European conference on computer vision*, pp. 818-833, 2014.

[7] W. Wang, J. Yang, J. Xiao, S. Li and D. Zhou, "Face Recognition Based on Deep Learning," *Human Centered Computing*, pp. 812-820, 2014.

[8] S. Ahn, "Deep Learning Architectures and Applications," *Journal of Intelligence and Information Systems*, vol. 22, no. 2, pp. 127-142, 2016.

[9] X. S. Yang and Z. W. Geem, *Music-inspired harmony search algorithm: theory and applications*, 2009.

[10] K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice," *Computer methods in applied mechanics and engineering*, vol. 194, no. 36, pp. 3902-3933, 2005.

[11] T. J. Lee, S. M. Park, K. E. Ko, W. K. Sung and K. B. Sim, "Implementation of unsupervised nonlinear classifier with binary harmony search algorithm," *Journal of Korean Institute of Intelligent Systems*, Vol. 23, no. 4, pp. 354-359, 2013.

[12] S. K. Lee, K. E. Ko and K. B. Sim, "Study on Improvement of Convergence in Harmony Search Algorithms," *Journal of Korean Institute of Intelligent Systems*, Vol. 21, no. 3, pp. 401-406, 2011.

[13] Z. W. Geem and K. B. Sim, "Parameter-setting-free harmony search algorithm," *Applied Mathematics and Computation*, Vol. 217, no. 8, pp. 3881-3889, 2010.

[14] Y. LeCun, C. Cortes and C. J. Burges, MNIST handwritten digit database. AT&T Labs [Online]. Available: <http://yann.lecun.com/exdb/mnist/>, 2010, [Accessed: December 25, 2016]

[15] J. S. Ren and L. Xu, "On vectorization of deep convolutional neural networks for vision tasks," *arXiv preprint arXiv:1501.07338*, 2015.

저자 소개



이우영(Woo-Young Lee)

2016년 : 중앙대학교 전자전기공학부공학사
 2016년~현재 : 중앙대학교 대학원
 전자전기공학과 석사과정

관심분야 : 의도인식, 감성인식, 지능로봇, 지능시스템, 사물인터넷 (IoT), 빅데이터 등.

Phone : +82-2-820-5319

E-mail : lwy0611@cau.ac.kr



고광은(Kwang-Eun Ko)

2007년 : 중앙대학교 전자전기공학부공학사
 2007년~2009년 : 중앙대학교 대학원
 전자전기공학과 공학석사
 2009년 ~ 현재 : 중앙대학교 대학원
 전자전기공학과 박사과정

관심분야 : 지능형 로봇, 인간-컴퓨터 인터페이스, 감정 인식, 뇌 컴퓨터 인터페이스, 뉴로로보틱스, 의도 인식 등.

Phone : +82-2-820-5319

E-mail : kkeun@cau.ac.kr



김종우(Zong Woo Geem)

1991년 : 중앙대학교 토목공학과 공학사
 2000년 : 고려대학교 토목공학과 공학박사
 2009년 : Johns Hopkins Univ. 환경관리
 이학석사

1998년~2000년 : Virginia Tech 방문연구

2000년~2001년 : Univ. of Maryland, College Park 박사후 연구

2012년~현재 : 가천대학교 에너지IT학과 조교수

관심분야 : Harmony Search 알고리즘, 인간 경험의 도함수,
지속가능성, 물-환경-에너지-식량-인프라 연계

Phone : +82-31-750-5586

E-mail : geem@gachon.ac.kr

Homepage URL : <http://harmonysearch.info>



심귀보(Kwee-Bo Sim)

1990년 : The University of Tokyo 전자공학과
공학박사

1991년~현재 : 중앙대학교 전자전기공학부
교수

2006년~2007년 : 한국지능시스템학회회장

관심분야 : 인공생명, 뇌-컴퓨터 인터페이스, 감정인식, 의도인식,
유비쿼터스 지능형로봇, 지능시스템, 컴퓨터이셔널
인텔리전스, 지능형 홈 및 홈 네트워크, 유비쿼터스
컴퓨팅 및 센서 네트워크, 소프트 컴퓨팅(신경망,
퍼지, 진화연산), 다개체 및 자율분산로봇시스템,
인공면역시스템, 지능형 감시시스템, 사물인터넷(IoT),
빅데이터 등.

Phone : +82-2-820-5319

E-mail : kbsim@cau.ac.kr

Homepage URL : <http://alife.cau.ac.kr>