

WFSO 알고리즘을 이용한 인공 신경망과 합성곱 신경망의 학습

장현우¹ · 정성훈^{2*}¹한성대학교 전자정보공학과²한성대학교 기계전자공학부

Training Artificial Neural Networks and Convolutional Neural Networks using WFSO Algorithm

Hyun-Woo Jang¹ · Sung Hoon Jung^{2*}¹Department of Electronics and Information Engineering, Hansung University, Seoul 02876, Korea^{2*}School of Mechanical and Electronic Engineering, Hansung University, Seoul 02876, Korea

[요 약]

본 논문에서는 최적화 알고리즘으로 개발된 WFSO(Water Flowing and Shaking Optimization) 알고리즘을 사용한 인공신경망과 합성곱 신경망의 학습 방법을 제안한다. 최적화 알고리즘은 다수의 후보 해를 기반으로 탐색해 나가기 때문에 일반적으로 속도가 느린 단점이 있으나 지역 최소값에 거의 빠지지 않고 병렬화가 용이하며 미분 불가능한 활성화함수를 갖는 인공신경망 학습도 가능하고 구조와 가중치를 동시에 최적화할 수 있는 장점이 있다. 본 논문에서는 WFSO 알고리즘을 인공신경망 학습에 적용하는 방법을 설명하고 다층 인공신경망과 합성곱 신경망에서 오류역전파 알고리즘과 성능을 비교한다.

[Abstract]

This paper proposes the learning method of an artificial neural network and a convolutional neural network using the WFSO algorithm developed as an optimization algorithm. Since the optimization algorithm searches based on a number of candidate solutions, it has a drawback in that it is generally slow, but it rarely falls into the local optimal solution and it is easy to parallelize. In addition, the artificial neural networks with non-differentiable activation functions can be trained and the structure and weights can be optimized at the same time. In this paper, we describe how to apply WFSO algorithm to artificial neural network learning and compare its performances with error back-propagation algorithm in multilayer artificial neural networks and convolutional neural networks.

색인어 : WFSO, 인공신경망, 지역 최소값, 최적화, 합성곱 신경망**Key word** : WFSO, Artificial Neural Network, Local Minimum, Optimization, CNN<http://dx.doi.org/10.9728/dcs.2017.18.5.969>

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.

Received 19 August 2017; Revised 29 August 2017

Accepted 31 August 2017

***Corresponding Author; Sung Hoon Jung**

Tel: +82-2-760-4344

E-mail: shjung@hansung.ac.kr

I. 서론

최근 들어 딥러닝이 여러 분야에서 성공적인 결과를 보이면서 인공지능망이 새롭게 조명을 받고 있다. 더불어 인공지능망을 학습하는 방법에 대한 관심도 증가하고 있다. 현재까지 인공지능망 학습에 가장 널리 사용되는 알고리즘은 오류역전파 알고리즘(Error Back Propagation Algorithm)이다 [1]. 이 방식은 학습데이터의 입력별로 출력 층에서 원하는 출력과 현재출력 사이의 오류를 계산하고 이를 이용하여 입력 층 쪽으로 가중치를 갱신하는 방법이다. 오류 역전파 알고리즘이 대부분의 경우 잘 동작하지만 세 가지 해결해야 할 문제가 있다. 첫 번째로 지역 최소값으로 수렴할 가능성이 있다. 기본적으로 오류 역전파 알고리즘은 경사도를 기반으로 가중치를 갱신하기 때문에 지역 최소값에 빠지면 벗어날 수 없다. 두 번째로 다층 인공지능망을 학습시킬 경우 출력층에서 발생한 오류를 이용한 가중치 갱신 값이 역전파 과정에서 소실되어 입력층 쪽의 가중치가 학습되지 않는 문제가 있다. 그러므로 특히 다층을 갖는 심층 인공지능망(Deep Artificial Neural Network)의 경우 학습이 제대로 수행되지 않는다. 세 번째로 미분 불가능한 함수를 활성화함수로 갖는 인공지능망에는 적용할 수 없다 [1]. 첫 번째와 두 번째 문제를 해결하는 것은 여러 가지 방법이 개발되어 있으나 세 번째 문제는 근본적인 문제로서 해결할 방법이 없다.

인공지능망을 학습시키는 다른 방법으로는 최적화 알고리즘을 이용한 방법들이 있다 [2-5]. 최적화 알고리즘으로 개발된 유전알고리즘(Genetic Algorithms) [2], 진화전략에서 파생된 차분진화 알고리즘(Differential Evolution) [3], 개체와 군집의 이동이력에 근거한 입자군집최적화(Particle Swarm Optimization) 알고리즘 [4], 그리고 비교적 최근에 개발된 인공 벌 군집최적화(Artificial Bee Colony Optimization) [5] 알고리즘으로 인공지능망을 학습시키는 연구들이 진행되어왔다. 최적화알고리즘을 이용하여 인공지능망을 학습하게 되면 지역 최소값에 빠지는 문제가 상당부분 완화되고 심층인공지능망에서 오류가 사라지는 문제가 발생하지 않아 은닉 층이 많은 심층 인공지능망도 선행학습 없이 학습시킬 수 있으며 가중치와 함께 구조도 동시에 최적화 할 수 있다. 또한 활성화함수가 미분 불가능한 함수로 되어 있는 인공지능망의 경우도 학습 가능하다. 단점으로는 다수의 개체를 이용하여 가중치를 최적화하기 때문에 시간이 오래 걸린다는 것이다. 하지만 병렬화가 용이하기 때문에 대규모 병렬화를 통하여 수행 시간을 크게 단축시킬 수 있다. 최적화 알고리즘으로 인공지능망을 학습시킬 때 가장 문제가 될 만한 것은 매우 많은 차원의 가중치 공간에서 적절한 가중치를 찾는 방식이기 때문에 심층인공지능망처럼 은닉 층이 많은 경우 탐색공간이 매우 커져서 좋은 해로 수렴하기 어렵다는 것에 있다. 그렇기 때문에 현재 심층인공지능망을 학습하는 최적화알고리즘 개발은 아직 초기 단계에 머물러 있다.

본 논문에서는 최적화 기법으로 개발한 WFSO 알고리즘을 이용한 인공지능망 학습 방법을 제안한다. WFSO 알고리즘은

다변수 함수에서 지역 최소값에 빠지지 않고 전역 최소값을 빠르게 찾아가는 최적화 알고리즘이다 [6]. 우리는 WFSO 알고리즘을 인공지능망에 적용하기 위해 적절한 평가 함수를 구현하고 탐색 성능이 개선되도록 몇몇 부분을 수정하였다. 그리고 성능을 평가하기 위해 XOR 학습 문제와 MNIST 손 글씨 학습 문제를 오류 역전파 알고리즘과 WFSO 알고리즘으로 각각 학습시키고 그 결과를 비교하였다. 비교 결과 WFSO 알고리즘은 오류역전파 알고리즘에 비해 탐색에 시간이 오래 걸리지만 오류 역전파 알고리즘 보다 학습이 더 잘 되고 MNIST 손글씨 학습과 같은 문제에서도 오류역전파 알고리즘을 사용하는 텐서플로우 보다 좋은 성능을 내는 것을 알 수 있었다. 다만 최근 컴퓨터비전 문제에서 많이 응용되고 있는 심층신경망인 CNN(Convolutional Neural Networks) 구조에서는 텐서플로우와 비교하였을 때 약간 성능이 좋지 않았다. 이는 심층신경망과 같이 파라미터가 많은 문제에서는 보다 더 정교한 최적화 파라미터를 찾거나 알고리즘을 보완해야 할 필요가 있음을 보여준다. 그러나 텐서플로우처럼 상당부분 최적화된 프로그램과 유사한 성능을 보이는 것으로 보았을 때 WFSO 가 인공지능망 학습에 상당히 효율적임을 알 수 있다. 최적화 방법으로 인공지능망을 학습하게 되면 좀 더 다양한 분야에 인공지능망을 응용할 수 있다 [7,8].

본 논문은 다음과 같이 구성되어 있다. 2절에서는 기존의 WFSO 알고리즘을 설명한다. 3절에서 WFSO를 이용한 ANN의 학습방법에 대하여 다루며 4절에서 실험 결과에 대해 분석한다. 마지막으로 5절의 결론으로 끝을 맺는다.

II. 기존의 WFSO 알고리즘

WFSO 알고리즘은 다변수 함수에서 다수의 해를 지역 최소값에 빠지지 않고 전역 최적해를 찾아내는 알고리즘이다 [6]. Algorithm 1에서 보듯이 이 알고리즘은 세 단계의 반복으로 구성된다. 주변을 무작위 탐색하여 더 나은 값으로 위치를 이동하는 **Flowing**, 순위가 나쁜 물방울의 위치를 재설정하는 **Pouring**, 가장 좋은 물방울들이 일정기간 동안 움직임이 정체된 경우 이들을 강제로 흔들어 정체된 지역을 빠져나오게 하는 **Shaking**이 각각의 단계이다. 그림 1은 **Flowing** 단계를 보여준다. **Flowing** 단계에서 WFSO 알고리즘은 각각의 물방울을 진동시킨다. 이때 물방울은 각각 고유한 분산을 가지고 가우시안 랜덤 함수에 따라 진동한다. 만약 진동하여 도달한 곳이 더 적합한 위치이면 해당 위치로 갱신한다.

Pouring 단계에서는 물방울을 적합도 순으로 정렬하고 적합도가 나쁜 물방울들의 위치를 재설정한다. **Shaking** 단계는 물방울의 움직임이 지역 최소값에서 정체된 경우 적합도에 상관 없이 강제로 이동하여 지역 최소값을 탈출하도록 한다. 그림 2는 **Shaking** 단계를 보여주는데 **Shaking** 단계에서는 적합도가 나쁜 곳으로도 일정한 조건으로 이동할 수 있도록 한다.

Algorithm1. Water Flowing and Shaking Optimization (WFSO)

```

// t: time //
// n, m: the number of water drops and the maximum number of flowing trial //
// p, W: the pouring percent of bad water drops and the collection of water drops //
// hwi, pwi: the height and the position of water drop wi //
// rwi, fm: the rank of water drop wi and the number of water drops failed to move //
// σwi2: the variance of Gaussian noise of water drop wi //
// Ts: the threshold value for deciding the shaking mode //
t ← 0
initializing W(t)
evaluating W(t)
while(not termination-condition) do
  t ← t + 1
  flowing W(t)
  for i = 1 to n // for all water drops
    for j = 1 to m // flowing trial
      try to move to a new position, pwi(t+1) = pwi(t) + N(0, σwi2)
      if hwi(t+1) < hwi(t) then
        move to the position pwi(t+1)
        set moving success
        exit for loop
      endif
    endfor
  if not moving success then
    set pwi(t+1) = pwi(t)
    fm = fm + 1
  endif
  endfor
  pouring W(t)
  the p percent of bad water drops pour to the best position
  shaking W(t)
  if fm > Ts then
    for i = 1 to n // for all water drops
      set σwi = σwi + |N(0, rwi/n)|
      set σbest = 0 for elitism
      randomly select the + or - direction of each axis and
      set the direction to d
      set pwi*(t) = pwi(t)
      for j = 1 to m // flowing trial
        try to move to a new position, pwi(t+1) = pwi(t) + d * |N(0, σwi2)|
        if hwi(t+1) < hwi(t) then
          move to the position pwi(t+1)
          set moving success
          exit for loop
        endif
      endfor
    else
      move temporarily to a new position pwi(t) = pwi(t+1)
    endif
  endfor
  if not moving success then
    set pwi(t+1) = pwi*(t)
  endif
  endfor
  else
    decrease the σwi2(t+1) = 0.9 * σwi2(t), subject to σwi2(t+1) ≥ σmin2
  endif
  evaluating W(t)
end

```

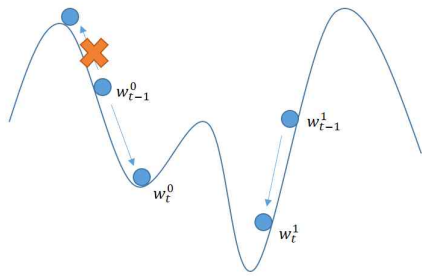


그림 1. Flowing 단계
Fig. 1. Flowing Step

예를 들어 만약 최대이동시도 값 m 이 5인 경우 적합도가 나빠지더라도 5번까지는 무조건 임시로 이동해본다. 만약 5번 임시 이동에서 계속 적합도가 나빠진 경우에는 지역 최소값을 벗어나지 못한 것으로 판단하여 초기 위치로 되돌아온다. 그러나

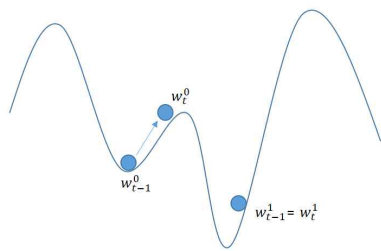


그림 2. Shaking 단계
Fig. 2. Shaking Step

5번 시도하던 중에 이전 임시 이동위치보다 더 좋아지는 경우가 발생하면 그 위치로 이동한다. 이렇게 하면 지역 최소값을 벗어날 수 있다. 이는 마치 밀바닥이 여러 곳에서 오목한 그릇에 담긴 물방울을 흔들었을 때 오목한 곳에서 물방울이 튀어나

와 다른 오목한 곳으로 들어갈 수 있도록 하는 것과 유사하다. 이를 위해 n 차원 공간에서 특정 방향으로 방향을 잡고 무조건 m 번 이동해 보는 것이다.

오류역전과 알고리즘은 오류를 역전과하여 가중치를 갱신하므로 다층 인공신경망 구조에서 입력에 가까운 층의 가중치가 거의 갱신되지 않는 문제가 있다. 그러나 WFSO 알고리즘은 가우시안 무작위 이동에 기반하여 탐색하므로 각각의 가중치 값이 유사한 비율로 갱신되어 다층 구조에서도 학습이 잘 수행된다. 또한 지역 최소값에 빠진 경우 이를 빠져나오는 Shaking 단계가 있어서 목표로 하는 오류율까지 계속하여 탐색이 가능하다. 다만 파라미터를 잘 못 설정하면 오류율이 올라갔다 내려갔다 진동하거나 일정 이하로 떨어지지 않는 문제가 발생할 수 있다. 이는 WFSO 알고리즘의 각종 파라미터를 적절히 조절함으로써 일정부분 극복 가능하다.

III. WFSO를 이용한 ANN의 학습

그림 3은 WFSO를 이용한 ANN의 학습과정을 보여준다. WFSO 알고리즘은 진동하는 물방울이 더 적합한 위치로 이동하며 탐색한다. 인공신경망을 학습시키기 위해 물방울의 위치를 인공신경망의 가중치로 사용하고 학습용 데이터로부터 오류율을 계산하였다. 계산된 오류율을 현재 물방울이 갖는 위치의 높이로 판단하여 물방울이 지금보다 위치가 더 낮은 곳으로 움직이게 한다. 인공신경망이 사용하는 가중치가 N 개인 경우 WFSO 알고리즘의 각 물방울은 N 차원의 평면 위에서 높이가 가장 낮은 곳으로 움직이면서 전역 최소값을 갖는 위치를 탐색한다. Flowing 단계에서 물방울이 어떤 위치로 이동을 시도하면 해당 위치 값으로 인공신경망의 가중치를 재설정하고 학습용 데이터를 활용하여 그 위치의 오류율을 계산한다. 만약 새로운 위치가 오류율이 더 낮다면 물방울을 그 위치로 갱신한다. 본 논문의 실험에서는 모든 학습용 데이터의 오류 제곱합 평균 (Mean Squared Error, MSE)을 오류율로 사용하였다. 또한 총 10

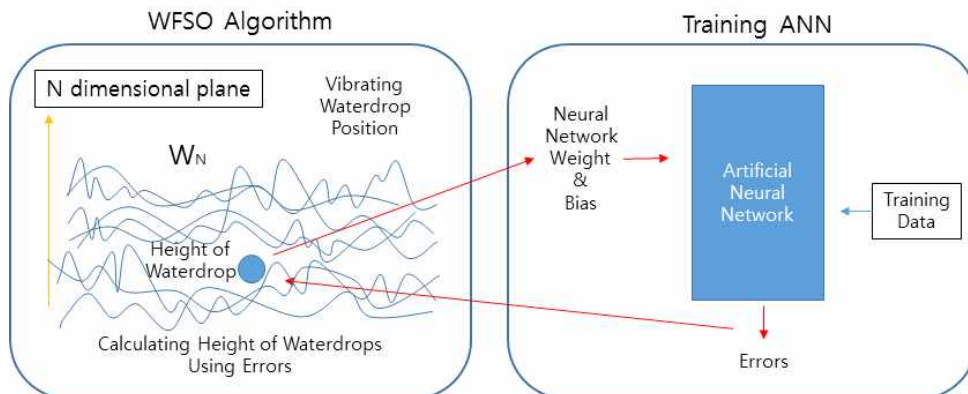


그림 3. WFSO를 이용한 ANN의 학습
Fig. 3. Learning ANN using WFSO

개의 물방울을 사용하였고 각각의 물방울은 이동을 5회 시도 하여 더 낮은 오류율을 찾도록 하였고 실패한 경우 그 횟수를 기록하도록 하였다. 이동에 사용하는 가우시안의 분산(σ^2)은 초기값을 0.1로 설정하였다.

기존 Pouring 알고리즘은 오류율이 높은 물방울을 오류율이 가장 낮은 물방울의 위치로 옮기는 연산을 수행했다. 하지만 실험 결과 가장 낮은 물방울이 지역 최소값에 멈춰있는 경우 모든 물방울이 해당 위치에 정제될 가능성이 있었다. 이런 결과를 반영하여 Pouring 단계는 물방울을 오류율로 정렬한 후 오류율이 높은 상위 3분의 1에 해당하는 물방울의 위치를 무작위로 초기화하는 방식으로 구성하였다. 이는 위치가 좋지 않은 물방울들을 제거하고 그 양만큼의 물방울을 새로 넣는 것으로 볼 수 있다. 매 단계마다 Pouring이 수행하는 경우 한 번 초기화된 물방울이 계속해서 초기화되는 현상이 발생하였다. 이를 해결하기 위해 Flowing 100회 당 1회와 같은 주기로 Pouring을 수행하였다.

Shaking 단계는 어느 지역에서 물방울의 움직임이 정제된 경우 그릇을 흔들어 그 물방울들이 지역 최소값에서 빠져나오도록 만드는 것이다. 실험을 반복한 결과 물방울이 가진 분산의 값이 클수록 탐색할 영역이 넓어져 좋은 해를 찾을 가능성이 있지만 제한 횟수 내에 탐색을 실패할 가능성도 높다는 것을 알 수 있었다. 또한 분산이 작을수록 탐색에 성공할 가능성은 높지만 좋은 해를 찾을 가능성이 낮아서 학습이 매우 느리게 되었다. 이를 반영하여 Soft Shaking을 추가하였다. Soft Shaking에서 물방울은 Flowing 단계에 성공하여 이동한 경우 더 좋은 해를 찾기 위하여 분산이 1.01배 되며 Flowing 단계를 연속하여 3회씩 실패할 때마다 실패할 확률을 낮추기 위해 0.8배 된다. 만약 이러한 분산 조정에도 불구하고 연속하여 이동에 10회 실패하였다면 기존 알고리즘의 Shaking(Hard Shaking으로 부름)을 시도하도록 구성했다. 그러므로 Hard Shaking 하기 전에 최대 로 수행할 수 있는 Soft Shaking은 3회, 6회, 9회에서 연속 실패하는 경우로 총 3회이다. Hard Shaking은 2절에서 설명한 것처럼 지역 최소값을 강제로 벗어날 수 있도록 하는 방법이기 때문에 대부분의 지역 최소값에 빠진 후보 해들이 지역 최소값을 벗어날 수 있다. 그러므로 전역 최소값을 효과적으로 찾을 수 있다.

IV. 실험 및 결과

4-1 XOR 학습

첫 번째 실험 문제로 간단하면서도 인공신경망 역사에서 의미 있는 XOR 문제를 선택하였다. 인공신경망의 구성은 입력 노드 2개, 은닉 노드 2개, 출력 노드 1개를 갖는다. 이 신경망의 가중치는 바이어스를 포함하여 9개로 구성된다. 따라서 WFSO 알고리즘에서 사용하는 각각의 물방울은 9차원의 최적화 파라미터를 갖게 된다.

WFSO 알고리즘과 오류역전과 알고리즘으로 각각 XOR 회로 인공신경망을 학습시켰다. 오류역전과 알고리즘으로 학습시킬 인공신경망의 활성화 함수는 $x < 0$ 에서 $f(x) = 0.1x$ 를 갖는 Leaky ReLU로 설정하였으며 WFSO 알고리즘으로 학습시킬 인공신경망의 활성화 함수는 Sigmoid로 설정하였다. 이렇게 설정한 이유는 오류역전과 알고리즘에서는 Sigmoid 보다 ReLU가 더 성능이 좋았으며 WFSO에서는 Sigmoid가 ReLU보다 더 성능이 좋았기 때문이다. 오류역전과 알고리즘의 학습률(Learning Rate)은 0.1로 설정하였다. 각각 MSE가 0.0001에 도달할 때까지 학습하였고 10초 이상 목표하는 MSE에 도달하지 못하면 학습에 실패한 것으로 하였다. 초기 가중치의 범위는 $[-1, 1]$ 로 설정하였다.

표 1. WFSO 알고리즘의 학습 성능 비교 (2-2-1 구조)

Table 1. Performance comparison of WFSO algorithm (2-2-1 structure)

Algorithm	Backpropagation	WFSO
Trial	1000	1000
Success	834	1000
Failure	166	0
Average Success Time(ms)	6.087	35.312

표 1은 은닉층 노드가 2개인 경우 오류역전과 알고리즘과 WFSO 알고리즘의 학습 성능을 비교한 것이다. 실험은 병렬화와 GPU의 사용 없이 3.40GHz의 4세대 Intel CPU 코어 1개로 시행하였다. XOR 학습을 1000회씩 실행하였을 때 오류역전과 알고리즘의 경우 1000회 중 166회 제한 시간 안에 목표 MSE까지 도달하지 못하였다. 학습에 실패한 경우 평균 epoch는 6,359,697회이다. 학습률을 0.01로 설정하였을 경우 실패 횟수는 1000회 중 137회로 감소했으나 성공 시 평균 소요 시간은 327ms로 증가했다. 학습률을 0.3으로 설정한 경우 실패 횟수는 1000회 중 228회로 증가했고 성공 시 평균 소요 시간은 1.3ms로 감소했다. 학습률을 변동하여도 시간 내에 목표 MSE까지 도달하지 못하는 경우가 전체 시도의 10% 이상 존재했다. XOR 문제가 단순한 문제이지만 네트워크 구조가 간단한 경우 가중치와 바이어스의 초기값에 따라서 지역 최소값에 빠질 가능성이 높다고 알려져 있다. 그에 비해 WFSO 알고리즘으로 실시한 학습의 경우 1000회 모두 제한 시간 내에 목표 SSE에 도달하였다. 탐색 성공 시의 소요 시간만을 고려하면 학습률이 0.1인 경우 오류역전과 알고리즘은 평균 6ms, WFSO 알고리즘은 평균 35ms로 약 6배의 차이를 보였다.

표 2는 은닉층 노드가 5개인 경우 오류역전과 알고리즘과 WFSO 알고리즘의 학습 성능을 비교한 것이다. 은닉층 노드를 5개로 늘린 경우 학습률 0.1의 오류역전과 알고리즘과 WFSO 알고리즘 모두 10초 이내에 목표 MSE에 도달하였다. 이때 오류역전과의 성공 시 평균 소요 시간은 1.412ms로 감소하였고

WFSO 알고리즘의 소요 시간은 54.312ms로 증가하였다. 은닉층 노드 개수가 증가하면 XOR 회로를 구현하는 방식이 증가해서 지역 최소값에 빠질 가능성이 줄어들고 이 때문에 오류역전파 알고리즘의 학습이 실패할 가능성이 줄어든 것으로 판단된다.

표 2. WFSO 알고리즘의 학습 성능 비교 (2-5-1 구조)
Table 2. Performance comparison of WFSO algorithm (2-5-1 structure)

Algorithm	Backpropagation	WFSO
Trial	1000	1000
Success	1000	1000
Failure	0	0
Average Success Time(ms)	1.412	54.312

그에 비해 WFSO 알고리즘은 은닉층 노드의 증가로 전체 가중치 개수가 증가하여 탐색해야할 공간이 넓어졌기 때문에 소요 시간이 늘어난 것으로 판단된다. 결론적으로 WFSO 알고리즘을 통한 학습은 다수의 해를 기반으로 탐색하기 때문에 시간이 더 오래 걸리지만 인공신경망의 크기에 상관없이 안정적으로 최적화를 수행하는 것을 알 수 있다.

또한 활성화 함수를 계단 함수로 설정하여 XOR 학습을 시도하였다. 이때 WFSO 알고리즘은 학습이 가능했지만 오류역전파 알고리즘은 불가능했다. 오류역전파 알고리즘은 활성화 함수의 미분이 정의되어야 하는데 계단 함수는 x=0에서 미분 불가능이기 때문이다. 이러한 WFSO 알고리즘의 특성은 결과가 명확히 1 혹은 0으로 나타나는 논리 회로 문제와 참/거짓을 다루는 문제에 대한 인공 신경망 학습에서 큰 장점으로 작용할 것이다.

4-2 MNIST 학습 (ANN 과 비교)

MNIST는 손으로 쓴 아라비아 숫자에 대한 데이터가 저장되어 있다. 해당 데이터세트에는 28*28 픽셀 사이즈의 55,000개의 훈련용 데이터와 10,000개의 테스트 데이터가 포함되어 있다. 따라서 학습용 데이터로 학습한 인공신경망이 학습용 데이터에 과적합 되었는지의 여부를 알아볼 수 있다.

이 실험에서는 구글이 개발한 인공신경망 학습 오픈소스 라이브러리인 텐서플로우와 성능을 비교한다. 은닉층이 하나인 인공신경망을 이용하여 MNIST 데이터를 오류역전파 알고리즘으로 학습시킨다. MNIST 데이터 세트는 28*28 픽셀 크기의 데이터를 제공하므로 인공신경망의 입력 노드는 784개로 설정하였다. 또한 10개의 은닉층 노드, 10개의 출력층 노드를 설정했다. 출력의 결과는 각각 0에서부터 9까지 각각 그 숫자일 가능성을 나타내며 가장 높은 출력으로 최종적으로 해당 그림이 어떤 숫자를 나타내는지를 결정한다.

WFSO 알고리즘으로 학습시키기 위해 적합도 평가 함수를 구현하였다. 하나의 학습 데이터에 대한 적합도를 10개 출력의 MSE로 설정하고 전체 학습 데이터에 대한 MSE를 최종적인 평가용 적합도로 설정하였다. 또한 Flowing 400회당 1번 씩 Pouring 단계가 실행되도록 수정하여 적용했다. 실험을 반복한 결과 모든 물방울이 Flowing 400회까지는 지역적인 값에 빠지는 경우가 적었기 때문이다. WFSO 학습의 경우 Flowing부터 Shaking까지의 동작을 2,500회 반복하였다.

사용하는 학습 데이터의 개수는 1,000개, 2,000개, 3,000개 등으로 구성하여 실험하였으며 과적합 여부를 평가하기 위한 테스트 데이터는 10,000개를 사용했다. 표 3은 실험결과를 보여 준다. 모든 데이터 크기에서 텐서플로우 보다 WFSO가 더 학습이 잘되었음을 볼 수 있다.

표 3. WFSO 알고리즘의 학습 성능 비교 (784-10-10 구조)
Table 3. Performance comparison of WFSO algorithm (784-10-10 structure)

Data Size	Accuracy (%)	
	Backpropagation	WFSO
1,000	80.12	86.03
2,000	81.46	88.58
3,000	82.01	89.60

4-3 MNIST 학습 (CNN 과 비교)

이 실험에서는 MNIST 학습을 위한 신경망을 다층 CNN 구조로 구성하고 이를 WFSO 알고리즘과 텐서플로우로 학습시킨 후 비교한다. WFSO 알고리즘은 아직 연산이 최적화되어 있지 않기 때문에 MNIST 데이터를 그대로 적용할 경우 상당히 많은 시간이 소요된다. 실험을 빠르게 진행하기 위해 28x28 픽셀로 구성된 MNIST 이미지를 Bicubic 보간법 알고리즘을 활용하여 10*10 픽셀 사이즈 이미지로 조정하여 사용했다.

그림 4는 학습에 사용한 CNN 인공신경망의 구조이다. 10*10 이미지를 활용해 두 차례의 Convolution과 Pooling으로 1*1의 Feature Map을 6개 생성하였다. 이후 10개의 은닉층을 가진 Fully Connected ANN을 통과하여 10개의 출력을 생성했다. 1,000개의 학습용 데이터와 10,000개의 평가용 데이터를 활용했다. WFSO 알고리즘의 경우 Flowing부터 Shaking까지의 단계를 10,000회 반복하였다. 텐서플로우는 1,000개의 데이터에 대해 0.0002의 학습률로 30 epoch 수행하였다. 그 결과 WFSO 알고리즘을 사용한 경우 65.92%의 정확도를 보였고 텐서플로우를 사용한 경우 73.14%의 정확도를 보였다. 이러한 결과는 WFSO 알고리즘이 다층 신경망에 대해서도 학습할 수 있으며 그 차이가 텐서플로우와 크게 나지 않는다는 것을 보여주었다.

WFSO 탐색 알고리즘이 하나의 CPU에서 동작되어 비교적

간단한 CNN 구조에서 비교 실험하였으나 보다 복잡한 CNN에

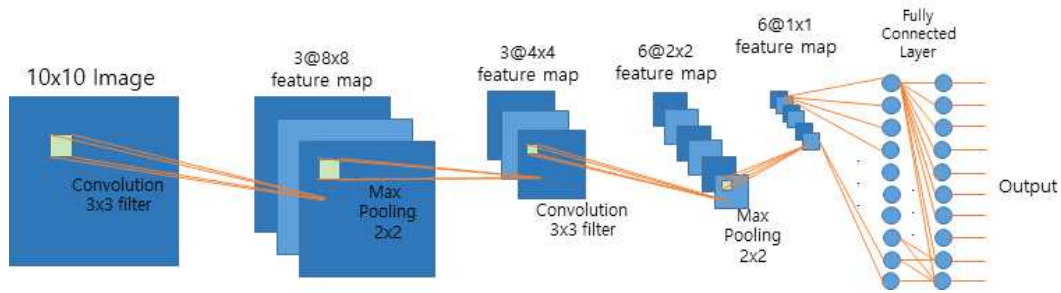


그림 4. MNIST 학습을 위한 CNN 구조
Fig. 4. Structure of CNN for MNIST Learning

서도 유사한 성능을 낼 수 있을 것으로 예상된다. 또한 WFSO 알고리즘은 각 물방울이 독립적으로 동작하기 때문에 GPU에서 수행되게 수정하면 매우 빠른 동작이 가능할 것으로 예상된다.

V. 결론

본 논문에서는 인공신경망 학습을 위한 새로운 방법으로 WFSO 알고리즘을 제안하였다. WFSO 알고리즘은 다변수 함수에 대해 지역적인 값에 머물지 않고 최대값이나 최소값을 찾아내는 알고리즘으로 이러한 성질은 인공신경망에서 강점을 가진다. WFSO 알고리즘을 이용해 XOR 학습, 단층 신경망 MNIST 학습, 다층 신경망 MNIST 학습의 세 가지 실험을 진행했다. XOR 학습에서 오류역전과 알고리즘은 인공신경망의 크기가 작을 경우 지역 최소값에 머무르는 경우가 있었지만 WFSO 알고리즘은 인공신경망의 크기가 작아도 학습이 잘 수행되는 것을 보였다. 또한 WFSO 알고리즘은 활성화 함수가 미분 불가능한 경우에도 학습이 가능하다는 것을 알 수 있었다. 단층 신경망의 MNIST 학습에서 WFSO 알고리즘은 시간은 오래 걸렸지만 텐서플로우보다 더 좋은 성능을 보였다. 다층 CNN 신경망에 대한 MNIST 학습은 WFSO 알고리즘이 다층적인 학습이 가능하며 그 성능이 텐서플로우와 유사함을 보였다. 현재 개발된 WFSO 알고리즘은 초기 알고리즘으로 각종 파라미터가 최적화되지 못한 상황이다. 앞으로 알고리즘과 파라미터를 보다 더 최적화하면 텐서플로우와 유사하거나 높은 성능을 낼 수 있을 것으로 기대한다. 또한 WFSO 알고리즘은 여러 개의 물방울이 독립적으로 움직이기 때문에 병렬화 하기가 매우 용이하다. 향후 GPU 상에 병렬화를 하게 되면 속도도 매우 빨라질 것으로 예상된다. 알고리즘 최적화와 병렬화를 완성하면 심층신경망을 과적합을 피하여 빠르게 학습할 수 있는 새로운 학습방식으로 자리 잡을 수 있을 것으로 기대한다. 또한 학습 속도를 더 향상시키거나 정확도를 향상시키기 위한 더 좋은 파라미터와 학습방법에 대한 추가적인 연구가 필요하다 하겠다.

감사의 글

본 연구는 한성대학교 교내학술연구비 지원과제로서 2017년 한국디지털콘텐츠학회 하계공동학술발표대회에서 초기 결과를 발표하였습니다.

참고문헌.

- [1] V. Vemuri, "Artificial Neural Networks: Theoretical Concepts," *IEEE Computer Society Press*, 1990
- [2] D. J. Montana and L. Davis, "Training Feedforward Neural Networks Using Genetic Algorithms," *In IJCAI*, vol. 89, pp. 762-767, Aug 1989.
- [3] J. Ilonen, J. K. Kamarainen, and J. Lampinen, "Differential evolution training algorithm for feed-forward neural networks," *Neural Processing Letters*, vol. 17, no. 1, pp. 93-105. 2003.
- [4] Michael Meissner, Michael Schmuker, and Gisbert Schneider, "Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training," *BMC bioinformatics*, vol. 7, no. 1, pp. 125-135, 2006
- [5] D. Karaboga, B. Akay, and C. Ozturk, "Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks," *MDAI*, vol. 7, pp. 318-319, 2007.
- [6] Sung Hoon Jung, "Water Flowing and Shaking Optimization," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 12, no. 2, pp. 173-180, Jun 2012.
- [7] K.-T. Bae and C.-J. Kim, "An Agricultural Estimate Price Model of Artificial Neural Network by Optimizing Hidden Layer," *Journal of Advanced Information Technology and Convergence*, vol. 14, no. 12, pp. 161-169, 2016.
- [8] K.-H. Kim and S. H. Jung, "Automatic Generation of a Configured Song with Hierarchical Artificial Neural

Networks," *Journal of DCS*, vol. 18, no. 4, pp. 641-647, 2017.



장 현 우 (Hyun Woo Jang)

2012년 3월 ~ 현재: 한성대학교 전자정보공학과 (학부과정)

관심분야 : IT, 인공지능



정 성 훈 (Sung Hoon Jung)

1988년 2월: 한양대학교 전자공학과 (공학사)

1991년 2월: KAIST 전기및전자공학과 (공학석사)

1995년 2월: KAIST 전기및전자공학과 (공학박사)

1996년 3월 ~ 현재: 한성대학교 기계전자공학부 교수

관심분야 : 인공지능, 시스템생물학, 융합공학