

# 인공지능 기계학습 방법 비교와 학습을 통한 디지털 신호변화

이덕균, 박지은\*  
대구대학교 인문교양대학 교수

Digital signal change through artificial intelligence machine  
learning method comparison and learning

Dokkyun Yi, Jieun Park\*  
Professor, Seongsan Liberal Arts College, Daegu University

요 약 앞으로의 시대는 인공지능을 이용한 다양한 분야에 다양한 제품이 생성될 것이다. 이러한 시대에 인공지능의 학습 방법의 동작 원리를 알고 이를 정확하게 활용하는 것은 상당히 중요한 문제이다. 이 논문은 지금까지 알려진 인공지능 학습 방법을 소개한다. 인공지능의 학습은 수학의 고정점 반복 방법(fixed point iteration method)을 기반으로 하고 있다. 이 방법을 기반으로 수렴 속도를 조절한 GD(Gradient Descent) 방법, 그리고 쌓여가는 양을 누적하는 Momentum 방법, 마지막으로 이러한 방법을 적절히 혼합한 Adam(Adaptive Moment Estimation) 방법 등이 있다. 이 논문에서는 각 방법의 장단점을 설명한다. 특히, Adam 방법은 조정 능력을 포함하고 있어 기계학습의 강도를 조정할 수 있다. 그리고 이러한 방법들이 디지털 신호에 어떠한 영향을 미치는 지에 대하여 분석한다. 이러한 디지털 신호의 학습과정에서의 변화는 앞으로 인공지능을 이용한 작업 및 연구를 수행함에 있어 정확한 활용과 정확한 판단의 기준이 될 것이다.

주제어 : 기계학습, 인공지능, 디지털 신호, 최적화, 방법

Abstract In the future, various products are created in various fields using artificial intelligence. In this age, it is a very important problem to know the operation principle of artificial intelligence learning method and to use it correctly. This paper introduces artificial intelligence learning methods that have been known so far. Learning of artificial intelligence is based on the fixed point iteration method of mathematics. The GD(Gradient Descent) method, which adjusts the convergence speed based on the fixed point iteration method, the Momentum method to summate the amount of gradient, and finally, the Adam method that mixed these methods. This paper describes the advantages and disadvantages of each method. In particularly, the Adam method having adaptivity controls learning ability of machine learning. And we analyze how these methods affect digital signals. The changes in the learning process of digital signals are the basis of accurate application and accurate judgment in the future work and research using artificial intelligence.

Key Words : Machine Learning, Artificial Intelligence, Digital Signal, Optimization, Method

\*This work was supported by the National Research Foundation of Korea (NRF-2017R1E1A1A03070311)

\*Corresponding Author : Jieun Park(writer2yah@daegu.ac.kr)

Received July 31, 2019

Revised September 10, 2019

Accepted October 20, 2019

Published October 28, 2019

## 1. 서론

4차 산업혁명의 시대라 이야기하던 것도 이제는 식상한 말이 되었을 정도로 많은 변화가 이루어지고 있다. 포드, 볼보 및 대형 자동차 회사들은 '2020년에 무인차를 상용화 하겠다' 선언하였다. 벤츠, 아우디 등의 해외 대형 자동차 업체는 구글 등의 소프트웨어 회사에 자동차라는 생산품을 납품하는 회사로 전락하는 것을 방지하고자 각자 자신만의 무인자동차를 만들고 있다. 특히 포드는 앞으로의 회사는 자동차를 생산하는 것뿐만 아니라 자동차를 공유하여 서비스 하는 회사로 탈바꿈하려 하고 있다. 이러한 대형 자동차 회사부터 소규모 가전에 이르기까지 인공지능의 활용 범위는 나날이 커지고 있는 실정이다. 전자 가전제품에서의 인공지능을 활용한 무인 청소기, 대화하는 TV 등 앞으로의 시대에는 인공지능을 활용하지 않으면 안 되는 시기에 우리는 도달했다. 이러한 사실은 신문 기사 등의 매스컴을 통하여 무수히 접하고 있으며 간단한 조사만으로도 확인 가능하다. 뿐만 아니라 기계학습은 다양한 분야에 적용하려는 연구가 이루어지고 있다. 특히, 기계학습을 이용한 유류 유출, 기상 예측[1], 제품 불량 예측[2], 대화시스템[3], 의학 분야[4] 등 다양한 응용 사례가 있다.

이 논문에서는 인공지능의 학습 방법인 비용함수의 최적화방법에 관한 설명과 정수화 되어 들어오는 디지털 신호의 학습과정에서의 변화를 설명한다. 우선은 디지털화(정수화) 된 Data값의 입력으로부터(학습하고자 하는 Data) 학습의 결과로 얻고자 하는 결과를 우리는 결정할 수 있다. 이를 통하여 우리는 간단한 비용함수를 정의할 수 있다[5-11]. 좀 더 자세한 사항은 본문에서 설명하겠다. 이 비용함수를 이용하여 비용함수를 최소가 되게 만드는 변수들을 구한다. 이러한 과정을 기계학습이라 한다. 이렇게 기계학습을 통한 비용함수를 최소로 만드는 변수를 결정했으면, 학습 이후에 들어오는 디지털 신호는 비용함수를 최소화 하는 변수들에 의하여 변화하게 되고 이를 통하여 결과 값을 도출하게 된다. 즉 기계학습 이후의 동작을 할 수 있게 되는 것이다.

비용함수의 최솟값을 계산하기 위해서는 기본적으로 비용함수의 일차 미분이 0이 되는 값을 찾아가는 방법을 사용하게 된다. 이 방법이 가장 기본적인 방법이며 비용함수의 일차미분 함수가 0이 되도록 만족하는 점을 찾으므로 비용함수의 최솟값을 구하게 된다. 이때 결정되는 변수들을 이용하여 기계는 학습을 했다고 우리는 이야기 한다. 비용함수의 일차미분을 0으로 만드는 방법 중

가장 기본이 되는 방법이 GD(Gradient Descent) 방법이다[12-18]. 이 방법이 가장 기본이 되며, 이를 기반으로 지금까지의 기계학습이 발전해 왔다. 이후에 널리 사용되는 방법 중의 하나인 Momentum방법에 대하여 설명할 것이다. 그러나 이 방법은 여러 가지 이로운 점이 있으나, 변수의 변화가 심하고 해의 수렴성면에 있어 안정적이지 못한 부분이 있다. 이러한 문제점을 극복하고자 개발된 방법이 Adam(Adaptive Moment Estimation) 방법이다[18]. 기존 기계학습 방법의 문제점은 학습의 어려움에 따라 학습의 강도 조절을 하지 못하는 것이었다. 그러나 Adam 방법은 학습의 어려움에 따라 학습 강도를 조절할 수 있는 능력을 포함하고 있다. 이 방법이 가장 널리 사용되고 있으며, 상당부분 효과적인 방법으로 잘 알려진 방법이다. 또한 학습하고자하는 Data의 복잡도가 올라갈수록 인공지능계의 복잡도가 올라갈 수 밖에 없는데, Adam방법이 상당히 유용하게 사용될 수 있다. 이를 수치적 실험을 통하여 확인한다. 이처럼 학습 어려움의 변화에 따른 학습 강도를 조정하는 연구가 이루어지고 있다.

### 1.1 인공지능 학습방법

인공지능의 학습방법은 많은 data로부터 이 data를 이용하여 비용함수(Cost function)을 만들고 이를 최적화하는 방법으로 학습이 이루어진다. 여기서 인공지능 학습에 있어 해결해야 하는 문제가 발생한다. 우선은 비용함수를 정의하는 것이다. 다양한 문제에 따른 다양한 비용함수가 정의 되는데 여기서는 가장 기본이 되는 Convex 함수의 기본을 가지고 논의하고자 한다. 이는 data로부터 참값의 차이로 만들어진다. 다음 수식 방법으로 확인해 보자. 우선 학습 시켜야할 data를 다음의 값으로 정해보자. Data set  $Ds = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ 로 정의한다. 이 data set은  $n$ 개의 쌍으로 이루어져 있다. 즉 각  $x$ 의 값에 따라  $y$ 의 값이 나오기를 기대하는 것이다. 우리는  $y$ 를 참 값이라 생각한다. 우리가 인공신경망의 구조로부터 함수  $N_f$ 를 만들었다고 할 때 비용함수는 다음과 같은 모습으로 구성 된다. 비용함수

$$C = \frac{1}{n} \sum_{i=1}^n (y_i - N_f(x_i))^2$$

이다. 여기서 제곱을 한 것은 convex성질을 주기 위함이며, 이 효과로 최솟값을 찾아갈 수 있게 된다. 다음으로 인공 신경망으로부터의 함수  $N_f$ 를 정의하자.  $N_f$ 는 신경망의 구조에 따라 다양하게 구성할 수 있는데 우선은 가장 기본이 되는 1-layer, 1-row 구조를 설명하겠다. 가장 기본이 되

는 변수 값  $x$ 로부터 가장 기본이 되는 연산은 곱셈과 덧셈이다. 이를 계산하면  $w x + b$ 로 쓸 수 있다( $w$ 는 weight라 하고,  $b$ 는 bias라고 한다). 여기에 활성화 함수(activation function)을 추가하여  $N_f$  함수를  $\sigma(w x + b)$ 로 최종 정의하게 된다. 그러므로 비용함수는 최종적으로 다음과 같이 쓰일 수 있다.

$$C(w, b) = \frac{1}{n} \sum_{i=1}^n (y_i - N_f(w x_i + b))^2.$$

보통 활성화 함수는 0과 1사이의 값을 가지는 sigmoid function을 사용하게 되며,  $y$  값의 범위는 크기조절을 통하여 0과 1사이의 값을 가지도록 만든다. 가장 기본이 되는 1-layer, 1-row 구조를 보았고 다차(Multi) layer와 다차(Multi) row의 구조를 가지는 것이 보편적인  $N_f$  함수의 구조이다. 여기서 다양한 Data와 다양한 문제를 풀기 위해서는 구조가 복잡해 지는데 우리는 여기서 합성 함수를 사용하게 된다. 우리는 깊은 구조의 인공지능 신경망을 다음과 같이 기술 할 수 있다.  $N_f(x) = N_f(w_1 N_{f_1}(w_2 N_{f_2} \dots + b) \dots)$ . 즉, 합성함수를 이용하여 구성하게 된다.

### 1.2 Deep 구조의 문제

Data의 양이 많아지고 다루고자 하는 문제의 복잡도가 올라갈 때, 인공신경계의 신경망의 수 또한 이에 부합하게 복잡도가 올라갈 수 밖에 없다. 즉 다차(Multi) layer의 수가 많아진다. 그러므로 이러한 Fig. 1 (Deep 구조)에서는 신경망 Fig. 2의 변이( $w, b$ ) 값의 변화가 잘 이루어지지 못한다고 경험상으로 잘 알려진 사실이다. 또한 학습하고자 하는 Data의 양이 많아질수록 기계학습이 요구하는 인공신경망은 복잡해 질 수 밖에 없다. 복잡한 인공신경망은 여러 번의 합성함수를 통하여 복잡하고 깊은 구조의 인공 신경망을 구성하게 된다. 이러한 구조에서 비용함수의 미분을 통하여 계산하게 되면 합성함수의 미분에 따라 계산된다. 따라서 깊은 구조의 값일수록 여러 번의 미분 값의 곱으로 이루어진다. 그러므로 깊은 구조의 값들이 변화 없이 소멸하는 문제가 발생할 수 있다. 이러한 문제를 gradient vanishing problem[19]이라고 말한다. 이를 해결하고자 CNN이라는 방법이 도입되었다. CNN의 방법은 고양이의 뇌를 분석하여 얻은 결과로, 동물이 물체를 보는 동안에 뇌의 활성화 부분을 조사해 본 결과, 영상을 인식함에 있어 뇌의 특정 부분만이 활성화 되는 것을 알아냈다. 이러한 점을 착안하여 영상을 기

계학습화 할 때, 인공신경망의 특정 부분만이 활성화 되도록 도입된 방법이다. 이 방법을 활용하기 위하여, 일정 부분의 값들을 적당한 가중치 곱과 그들의 합을 통하여(convolution) 다음 layer에 전달하는 방법이다.

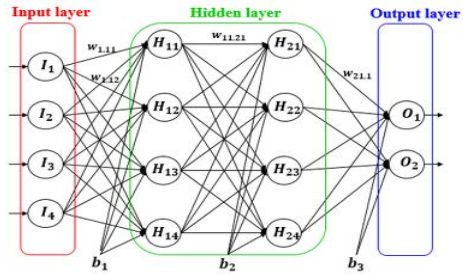


Fig. 1. Deep Neural Network

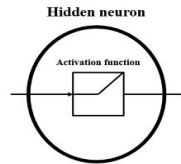


Fig. 2. Single Neural Network

이러한 방법은 복잡한 학습을 가능하게 했으며, 우리가 전에 경험한 인간과 기계의 대결인 알파고의 알고리즘 부분에도 쓰인 방법이다.

## 2. 최적화방법 비교

1.1장에서 확인 했듯이 인공지능의 학습은 비용함수의 최적화를 통하여 하게 된다. 그럼 기존의 잘 알려진 학습 방법인 GD, Momentum, 그리고 Adam에 관하여 설명을 하도록 하겠다. 이 방법들은 비용함수의 최적화를 목표로 함으로 비용함수의 일차 미분의 값을 0으로 만드는 방향으로 계산하게 된다. 즉  $\frac{\partial C}{\partial w} = 0$ 을 만족하는 변수  $w$ 의 값을 찾는 것이다. 동일한 방법으로  $b$ 에 관하여도 수행한다.  $\frac{\partial C}{\partial b} = 0$ 을 만족하는  $b$ 를 구하면 된다. 여기서 수치적 방법은 fixed point iteration 방법이라는 것이 있다. 이 방법은  $w$ 의 변화를 다음과 같이 해 나가는 방법이다.  $i$ 번째  $w$ 의 값을 알고 있다고 한다

면, 그 다음 시간의 변화는  $w_{i+1} = w_i - \frac{\partial C}{\partial w}$ 의 방법으로 계산하는 방법이다. 이 방법은  $w$ 의 값이 수렴하면 자동적으로  $w = w - \frac{\partial C(w)}{\partial w}$ 을 만족하고, 그러므로 수렴 값  $w$ 가  $\frac{\partial C}{\partial w} = 0$ 을 만족시키는 방법이다.

### 2.1 GD(Gradient Descent) 방법

이 방법은 fixed point iteration방법에 학습율(Learning rate)를 추가한 방법으로 식은 다음과 같다.  $w_{i+1} = w_i - \eta \frac{\partial C}{\partial w}$ .  $\eta$  값에 따라 학습의 강도가 변화하는 방법으로 경험적 값으로  $\eta$ 를 결정짓는다.

### 2.2 Momentum 방법

이 방법은 처음부터 계산되는  $\frac{\partial C}{\partial w}$ 의 값을 더하는 방법으로 식은 다음과 같다.

$m_i = \alpha m_{i-1} + \beta \frac{\partial C}{\partial w}$ . 여기서  $\alpha$ 와  $\beta$ 는 적당한 상수이다. 이렇게 계산된  $m$  값을 가지고 다음과 같은 식을 얻는다.  $w_{i+1} = w_i - \eta m_i$ .  $m$ 의 값의 효과로  $\frac{\partial C}{\partial w}$ 의 값이 계속 누적되어  $w$ 에 영향을 미치는 방법이다.

### 2.3 Adam 방법

Adam 방법은 앞의 Momentum방법에 Adaptive(조정가능)의 변수를 두는 방법으로 다음과 같다. Momentum방법에 계산되던 가중치는  $m_i = \beta_1 m_{i-1} + (1 - \beta_1) \frac{\partial C}{\partial w}$ 로 동일하게 계산하고, 여기에  $v_i = \beta_2 v_{i-1} + (1 - \beta_2) \left(\frac{\partial C}{\partial w}\right)^2$ 를 계산하여 최종적으로  $w$ 의 변화를 다음과 같이 한다.  $w_{i+1} = w_i - \eta \frac{\widehat{m}_i}{(\widehat{v}_i + \epsilon)^{1/2}}$ . 여기서  $\widehat{m}_i = \frac{m_i}{(1 - \beta_1^i)}$ 이며,  $\widehat{v}_i = \frac{v_i}{(1 - \beta_2^i)}$ 이다. Momentum의 방법에  $v$ 값의 계산을 통하여, Momentum의 효과도 나오며, 동시에 일정 비율의 값으로 학습되는 효과를 얻는다.

다양한 방법의 실험을 통하여 각 학습방법이 얼마나 영향을 미치는 지에 대하여 실험을 수행 한다.

### 3.1 layer의 수에 따른 이진 분류 결과 비교

앞에서 설명했듯이 인공신경망이 복잡한 일을 수행하기 위해서는 다차(Multi) layer와 다차(Multi) row의 구조를 가져야만 한다고 이야기 하였다. row의 변화는 인공신경망의 깊이와 관계가 없으므로, 이 실험에서는 다차 layer의 수에 따른 학습의 경과를 비교하고자 한다. 여기서 참은 1로 거짓은 0으로 선정한다. 즉 참과 거짓을 구분하는 경우에 대한 실험을 하고자 한다. 즉 이진 분류를 통하여 각 방법들의 성능을 비교하고자 한다. 어떤 대상을 두 종류로 분류하는 것은 기계 학습의 큰 분야 중 하나인 분류 문제의 가장 기본적인 경우이다. 이 실험에서는 layer의 개수가 3개, 5개일 때 각 방법들이 비용함수를 얼마나 최소화 시키는지와 각 파라미터들을 얼마나 변화시키는지 비교하였다. 일반적으로 파라미터들의 초기 값은 랜덤 값으로 준다. (bias의 경우는 초기 값을 0으로 주기도 한다.) 여기에서는 weight의 초기 값들은 랜덤 함수를 이용하여 얻은 후 그 값을 고정하여 실험하였다.

$x_i$ 는  $i$ 번째 학습 데이터를,  $y_i$ 는  $x_i$ 의 결과를,  $y_i^j$ 는  $y_i$ 를 계산하는데 쓰인 layer이다. 그리고 학습 데이터 ( $x_i, y_i$ )들의 집합을  $LS$  라고 표기하겠다. 이번 실험에서 사용한 학습 데이터 집합은  $LS = \{(5, 1), (-2, 0), (50, 1), (-0.2, 0)\}$ 으로 양수를 입력하면 1로, 음수를 입력하면 0으로 분류해주는 데이터이다. 이 학습 데이터 집합은 실험의 편의를 위하여 설정된 값이다.

#### 3.1.1 3-layer실험

layer의 수에 따른 변화를 보기 위함으로 3-layer의 경우를 가정했으며 row의 개수는 1로 한 경우에 한하여 실험한다. row의 개수도 학습에 영향을 미치나 이것을 layer의 변화에 비하여 적으므로 무시하기로 한다. 3-layer에서의 인공신경망의 함수 값은 다음과 같이 계산된다.

$$N_f(w_1, w_2, w_3, x) = \sigma(w_1 \sigma(w_2 \sigma(w_3 x + b_3) + b_2) + b_1) \quad (1)$$

식(1)을 그림으로 표현하면 다음과 같다.

## 3. 수치실험 결과

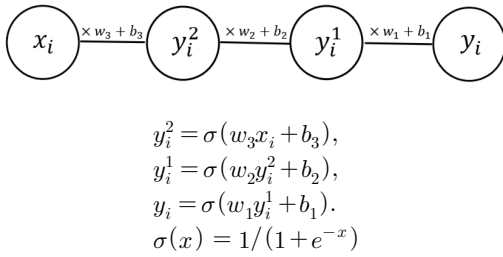


Fig. 3. 3-layer Neural Network

식 (1)을 다음의 조건하에서 계산하였다. 파라미터들의 초기 값은  $w_1 = 0.19771976$ ,  $w_2 = 0.15725612$ ,  $w_3 = -0.28434913$ ,  $b_1 = b_2 = b_3 = 0$ 이며 학습율 (learning rate)는  $\eta = 0.1$ 이고 반복횟수는 GD는 1000번, Momentum과 Adam은 200번이다. GD는 학습 속도가 느리기 때문에 GD만 반복 횟수를 1000번으로 하였다.

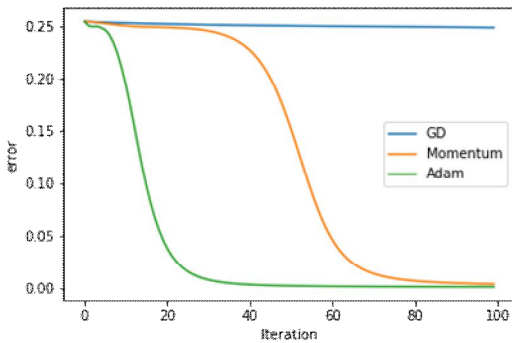


Fig. 4. Numerical Errors

Fig. 4는 주어진 학습 데이터 집합에 각 방법들을 적용하여 파라미터들을 최적화 시켰을 때의 비용함수의 함수 값을 반복횟수별로 나타낸 결과이다. 이 그림은 Adam 방법이 가장 빠르게 파라미터들을 최적화 시키는 것을 보여준다.

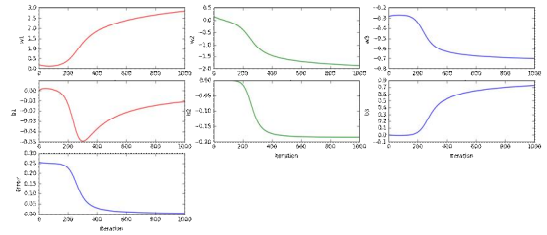


Fig. 5. Results of GD Method

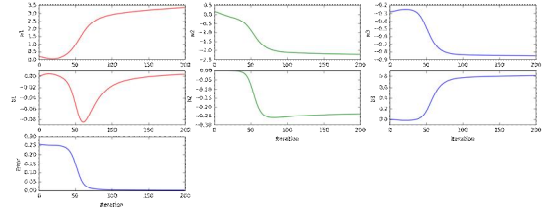


Fig. 6. Results of Momentum Method

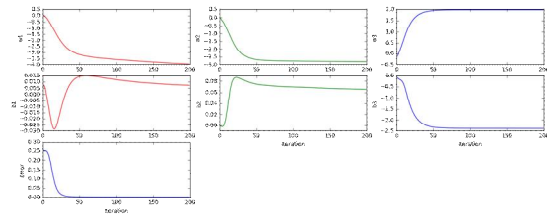


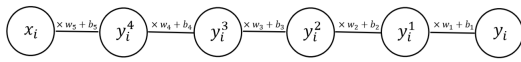
Fig. 7. Results of Adam Method

Fig. 5는 GD 방법, Fig. 6은 Momentum 방법, Fig. 7은 Adam 방법의 수치적 파라미터들의 변화를 보여 준다. Fig. 5를 보면 GD 방법은 다른 방법들에 비해서 파라미터의 최적화 속도가 가장 느린 것을 알 수 있다. 또한 chain rule을 이용한 backpropagation에서 활성화 함수의 미분의 함수값이 1보다 항상 작기 때문에 초반에 이용되는 파라미터들은 변화량이 적어짐을 알 수 있다. (이것이 Vanishing Gradient의 원인이 된다.) Fig. 5-7를 보면 Adam 방법이 초반에 이용되는 파라미터인  $w_3$ 과  $b_3$ 를 가장 많이 변화시키는 것을 확인할 수 있다.

### 3.1.2 5-layer 실험

좀 더 학습이 어려운 경우인 5-layer의 경우에 대하여 실험한다. 5-layer의 경우에서의 인공신경망의 계산 값은 다음과 같다.

$$N_f(w_1, w_2, w_3, w_4, w_5, x) = \sigma(w_1\sigma(w_2\sigma(w_3\sigma(w_4\sigma(w_5x + b_5) + b_4) + b_3) + b_2) + b_1) \tag{2}$$



$$\begin{aligned}
 y_i^4 &= \sigma(w_5 x_i + b_5), \\
 y_i^3 &= \sigma(w_4 y_i^4 + b_4), \\
 y_i^2 &= \sigma(w_3 y_i^3 + b_3), \\
 y_i^1 &= \sigma(w_2 y_i^2 + b_2), \\
 y_i &= \sigma(w_1 y_i^1 + b_1). \\
 \sigma(x) &= 1/(1 + e^{-x})
 \end{aligned}$$

Fig. 8. 5-layer Neural Network  
 파라미터들의 초기 값은  $w_1 = 0.34004834$ ,  
 $w_2 = 1.09908653$ ,  $w_3 = -0.38793867$ ,  $w_4 = -1.48041442$ ,  
 $w_5 = -0.74076256$ ,  $b_1 = b_2 = b_3 = b_4 = b_5 = 0$ 이며  
 3-layer의 실험 결과와 마찬가지로 학습율(learning rate)는  $\eta = 0.1$ 이고 반복횟수는 GD는 1000번, Momentum과 Adam은 200번 수행했다.

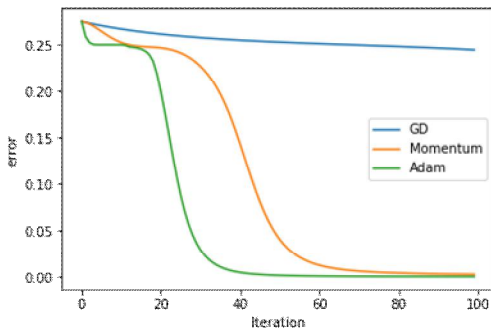


Fig. 9. Numerical Errors

3-layer의 실험 결과와 마찬가지로 Adam 방법이 가장 빠르게 파라미터들을 최적화 시키는 것을 보여준다.  
 다음 Fig. 10-12는 Adam 방법이 초반에 이용된 파라미터를 가장 많이 변화시키는 것을 보여준다.

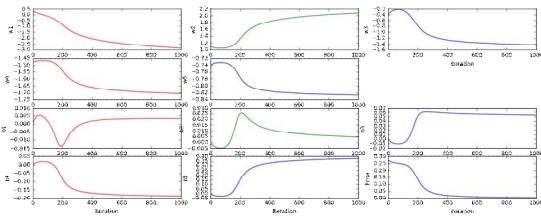


Fig. 10. Results of GD Method

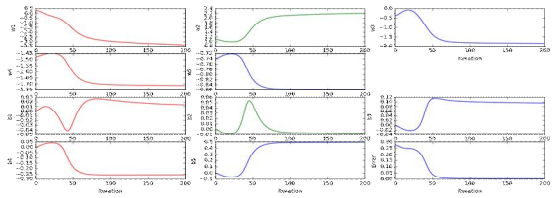


Fig. 11. Results of Momentum Method

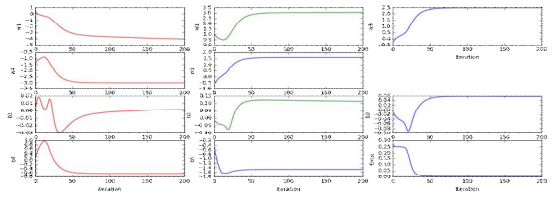


Fig. 12. Results of Adam Method

### 3.2 일차 함수에 관한 실험

이 실험은 비용함수에 관한 것으로 비용함수는 convex의 성질을 가지고 있다고 가정했으나, data의 복잡성과 layer의 수 등에 따라 non-convex의 성질이 나타난다. 이를 학습 방법이 해결할 수 있는 지에 대한 실험을 하고자 한다. 기계학습에서 이용되는 GD 방법은 convex 함수의 최적화 용도로 만들어졌으며 Momentum과 Adam은 GD를 기반으로 한 방법들이다. 하지만 기계학습에서 최적화 하는 비용함수는 일반적으로 convex 함수가 아니다. 따라서 이번 실험에서는 non-convex 함수에서 각 방법들의 성능을 비교하고자 한다. 이번 실험에서는 2개의 극솟값을 갖는 1변수 함수를 이용하여 각 방법들의 성능을 비교한다. 성능 비교에 이용된 비용함수는  $C(w) = (w+5)(w+3)(w-1)(w-10)/800 + 3$ 이며 이 함수는  $w \approx 7.1047$ 에서 최솟값을 갖는다. 파라미터  $w$ 의 초기값은 -9로 설정하였으며 반복횟수는 100번이다. 이에 따른 결과는 Fig. 13-15이다.

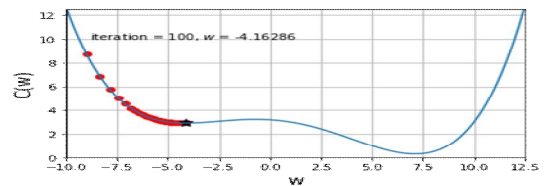


Fig. 13. Results of GD Method

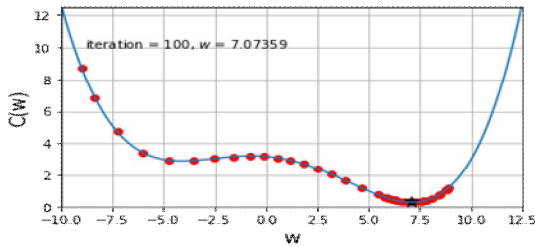


Fig. 14. Results of Momentum Method

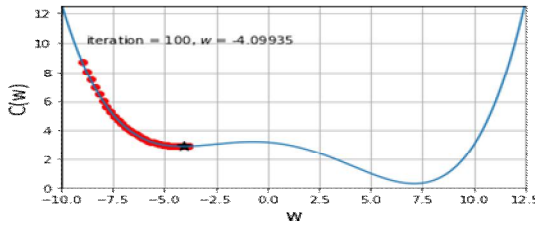


Fig. 15. Results of Adam Method

Fig. 13-15는 각 방법들이 주어진 함수의 최솟값을 찾아가는 과정을 보여 준다. GD 방법의 경우는 정의에 의하여 파라미터가 극값을 만나면 더 이상 값이 변하지 않는다. Momentum 방법의 경우는 파라미터가 극값을 만났다고 하더라도 극값을 만나기 전까지의 값들이 더해져 있어서 값이 변하지 않는 것을 막아준다. Adam 방법도 Momentum과 같은 기능이 있기는 하지만 학습이 반복될수록 파라미터들의 변화를 천천히 하도록 한다. Fig. 13-15를 비교하면 Adam 방법은  $w$ 의 변화가 다른 방법들에 비하여 더 촘촘한 것을 알 수 있다.

### 3.3 다차원 상황에서의 실험

일차원 공간 상황 이외의 다차원 상황에서의 학습방법에 관한 비교 실험을 수행한다. 이 실험에서는 2차원 함수를 이용하여 각 방법들의 성능을 비교한다. 여기서 이용된 함수는 Stybliski-Tang 함수

$C(w_1, w_2) = \frac{1}{2} \sum_{i=1}^2 (w_i^4 - 16w_i^2 + 5w_i)$ 이다. 하지만 비용함수는 함수 값이 항상 0보다 크거나 같아야 함으로 위에 거론한 함수에 상수 80을 더하여  $C(w_1, w_2) = \frac{1}{2} \sum_{i=1}^2 (w_i^4 - 16w_i^2 + 5w_i) + 80$  을 이용한다. 이 함수는 4개의 극솟값을 갖으며 최솟값은  $(w_1, w_2) = (-2.903534, -2.903534)$ 에서 갖는다. 이 실험에서 초기 값은  $(w_1, w_2) = (0, 6)$ 이고 반복횟수는 300번이다. 이에 따른 결과는 Fig. 16에서 보인다.

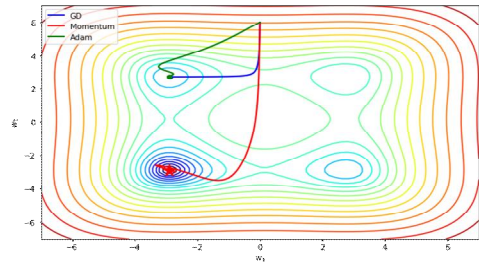


Fig. 16. Results of methods in 2-dimension

## 4. 결론

수치적인 실험의 결과에서는 인공지능계의 복잡도가 올라갈수록 Adam의 방법이 효과적임을 알 수 있다. 그러나 극소 미분이 0이 되는 값이 있는 경우 즉, 부분 최솟값이 존재하는 경우에 있어서는 Adam 방법의 한계가 드러난다. 이를 극복하는 연구가 필요하다고 생각한다.

입력되는 디지털 신호는 학습의 결과로 만들어지는 변수들의 값에 의하여 디지털 값을 벗어나게 된다. 이를 극복하고자 활성화(Activation function, Sigmoid) 함수를 이용하게 되는데 이 함수의 값이 0과 1의 값을 정해 주게 된다. 그러나 이 값도 연속 함수의 값을 가지므로 최종 결과에 있어서 우리가 원하는 디지털화는 부족하다 하겠다. 따라서 다양한 방법을 이용하여 디지털화를 할 수 있는 방법을 고안해야 한다. 그 중 하나의 방법이 확률을 이용하는 방법이 있다. 이러한 방법들의 변화는 비용함수의 변화를 유도하며 이를 극복하기 위한 또 다른 비용함수의 최적화 방법을 필요로 하게 된다. 이 논문에서는 기존 학습의 방법이 디지털 신호에 미치는 변화에 대하여 논의하였다. 비용함수의 디지털화를 통한 적용을 연구하면 그에 변화하는 비용함수의 최적화에 적합한 비용함수 계산방법이 필요하 하겠다. 이 논문에서 설명한 기초적인 방법을 기반으로 변화하는 비용함수의 최적화를 발전시키고 연구가 이루어 질 것이다. 이 연구의 후속으로 변화하는 비용함수의 최적화에 관한 연구도 수행 중에 있다.

## REFERENCES

[1] G. D. Kim & Y. H. Kim. (2017). A Survey on Oil Spill and Weather Forecast Using Machine Learning Based on Neural Networks and Statistical Methods. *Journal of the Korea Convergence Society*, 8(10), 1-8.  
 [2] J. Ku. (2017). A Study on the Machine Learning Model



for Product Faulty Prediction in Internet of Things Environment. *Journal of Convergence for Information Technology*, 7(1), 5-60.

[3] Y. Jeong. (2018). Machine Learning Based Domain Classification for Korean Dialog System. *Journal of Convergence for Information Technology*, 9(8), 1-8.

[4] Y. Namgoong, C. O. Kim & C. J. Lee. (2019). A machine learning model for the derivation of major molecular descriptor using candidate drug information of diabetes treatment. *Journal of the Korea Convergence Society*, 10(3), 23-30.

[5] L. Deng, J. Li, J. Huang, K. Yao, D. Yu, F. Seide, M. L. Seltzer, G. Zweig, X. He, J. Williams, Y. Gong & A. Acero. (2013). *Recent advances in deep learning for speech research at microsoft*. ICASSP.

[6] A. Graves. (2013). *Generating sequences with recurrent neural networks*. arXiv preprint arXiv:1308.0850.

[7] A. Graves, A. Mohamed & G. Hinton. (2013). Speech recognition with deep recurrent neural networks. *In Acoustics, Speech and Signal Processing(ICASSP), 2013 International Conference*, 6645-6649

[8] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever & R. R. Salakhutdinov. (2012). *Improving neural networks by preventing co-adaptation of feature detectors*. arXiv preprint arXiv:1207.0580

[9] T. Tieleman & G. E. Hinton. (2013). *Lecture 6.5 - RMSProp, COURSERA: Neural Networks for Machine Learning*. Technical report.

[10] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. Le, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang & A. Y. Ng. (2012). Large scale distributed deep networks. *in NIPS*.

[11] N. Jaitly, P. Nguyen, A. Senior & V. Vanhoucke. (2012). Application of pretrained deep neural networks to large vocabulary speech recognition. *in Interspeech*.

[12] S. Amari. (1998). Natural gradient works efficiently in learning. *Neural computation*, 10(2), 251-276.

[13] J. Duchi, E. Hazan & Y. Singer. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12, 2121-2159.

[14] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen & T. N. Sainath. (2016). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine*, 29(6), 82-97.

[15] A. Krizhevsky, I. Sutskever & G. E. Hinton. (2012). Imagenet classification with deep convolutional neural networks. *In Advances in neural information processing systems*, 1097-1105.

[16] R. Pascanu & Y. Bengio. (2013). *Revisiting natural gradient for deep networks*. arXiv preprint arXiv:1301.3584.

[17] I. Sutskever, J. Martens, G. Dahl & G. E. Hinton. (2013). On the importance of initialization and momentum in deep learning. *In Proceedings of the 30<sup>th</sup> International Conference on Machine Learning(ICML-13)*, 1139-1147.



3rd International Conference for Learning Representations, San Diego.

[18] E. Moulines & F. R. Bach. (2015). ADAM: A method for stochastic optimization. *The 3rd International Conference for Learning Representations, San Diego*.

[19] Y. Bengio, P. Simard & P. Frasconi. (1994). Learning long-term dependencies with gradient is difficult. *IEEE Transaction on neural networks*, 5(2), 157-166.

이 덕 균(Dokkyun Yi)

[장학원]

- 2000년 2월 : 한국과학기술원 수학과 (석사)
- 2005년 2월 : 한국과학기술원 수학과 (박사)
- 2015년 3월 ~ 현재 : 대구대학교 인문교양대학 조교수
- 관심분야 : 수치해석, 이미지처리, 인공지능
- E-Mail : dkyi@daegu.ac.kr

박 지 은(Jieun Park)

[장학원]

- 
- 1998년 2월 : 이화여자대학교 과학고 육과(화학전공) (학사)
  - 2013년 2월 : 이화여자대학교 과학고 육학과 (박사)
  - 2015년 3월 ~ 현재 : 대구대학교 인문교양대학 조교수
  - 관심분야 : 문제해결력, 표상, 인공지능
  - E-Mail : writer2yah@daegu.ac.kr