Journal of the Korea Academia-Industrial cooperation Society
Vol. 21, No. 2 pp. 189-194, 2020

https://doi.org/10.5762/KAIS.2020.21.2.189
ISSN 1975-4701 / eISSN 2288-4688

딥러닝을 위한 경사하강법 비교

강민제
제주대학교 전자공학과

Comparison of Gradient Descent for Deep Learning

Min-Jae Kang
Department of Electronic Engineering, Juju National University

요 약 본 논문에서는 신경망을 학습하는 데 가장 많이 사용되고 있는 경사하강법에 대해 분석하였다. 학습이란 손실함수가 최소값이 되도록 매개변수를 갱신하는 것이다. 손실함수는 실제값과 예측값의 차이를 수치화 해주는 함수이다. 경사하강법은 오차가 최소화되도록 매개변수를 갱신하는데 손실함수의 기울기를 사용하는 것으로 현재 최고의 딥러닝 학습알고리즘을 제공하는 라이브러리에서 사용되고 있다. 그러나 이 알고리즘들은 블랙박스형태로 제공되고 있어서 다양한 경사하강법들의 장단점을 파악하는 것이 쉽지 않다. 경사하강법에서 현재 대표적으로 사용되고 있는 확률적 경사하강법(Stochastic Gradient Descent method), 모멘텀법(Momentum method), AdaGrad법 그리고 Adadelta법의 특성에 대하여 분석하였다. 실험 데이터는 신경망을 검증하는 데 널리 사용되는 MNIST 데이터 셋을 사용하였다. 은닉층은 2개의 층으로 첫 번째 층은 500개 그리고 두 번째 층은 300개의 뉴런으로 구성하였다. 출력 층의 활성화함수는 소프트맥스함수이고 나머지 입력 층과 은닉 층의 활성화함수는 ReLu함수를 사용하였다. 그리고 손실함수는 교차 엔트로피 오차를 사용하였다.

Abstract This paper analyzes the gradient descent method, which is the one most used for learning neural networks. Learning means updating a parameter so the loss function is at its minimum. The loss function quantifies the difference between actual and predicted values. The gradient descent method uses the slope of the loss function to update the parameter to minimize error, and is currently used in libraries that provide the best deep learning algorithms. However, these algorithms are provided in the form of a black box, making it difficult to identify the advantages and disadvantages of various gradient descent methods. This paper analyzes the characteristics of the stochastic gradient descent method, the momentum method, the AdaGrad method, and the Adadelta method, which are currently used gradient descent methods. The experimental data used a modified National Institute of Standards and Technology (MNIST) data set that is widely used to verify neural networks. The hidden layer consists of two layers: the first with 500 neurons, and the second with 300. The activation function of the output layer is the softmax function, and the rectified linear unit function is used for the remaining input and hidden layers. The loss function uses cross-entropy error.

Keywords : Gradient Descent Method, Deep Learning, Neural Networks, MNIST, Softmax, Cross-Entropy

1. 서론

신경망은 2000년대에 들어오면서 딥러닝(deep learning)이란 새로운 이름으로 개명되었고 구글의 딥러닝 바둑 알고리즘이 세계적인 프로기사들을 이김으로써 세상의 조명을 받게 되었다. 신경망이 새롭게 태어날 수

*Corresponding Author : Min-Jae Kang(Jeju National Univ.)
email: minjk@jejunu.ac.kr
Received October 24, 2019 Revised December 24, 2019
Accepted February 7, 2020 Published February 29, 2020

있었던 것은 빅데이터와 프로세서의 발전 덕분이다. 빅데이터 확보로 많은 경우의 수를 테스트하고, 프로세서의 성능개선 또는 클라우드 컴퓨팅 기술로 엄청난 매개변수들을 학습할 수 있게 되었다. 물론 힌톤교수의 불굴의 노력도 중요한 역할을 하였다. 1990년대까지 신경망을 연구하던 많은 학자들이 신경망의 한계를 느끼면서 새로운 방향으로 거의 전환하였다. 그러나 힌톤교수는 신경망을 계속 연구할 수 있는 여건을 찾아서 대학교를 옮기기도 하였고 결국은 신경망을 딥러닝으로 재탄생시켰다[1].

딥러닝은 방대한 량의 데이터를 다루며 또 이를 위해서 구조적으로 은닉층의 수가 상당히 늘어난다. 이것은 신경세포들을 연결하는 매개변수들이 늘어나고 이를 학습하는 계산량이 방대해졌음을 의미한다. 프로세서와 클라우드 컴퓨팅 기술의 발전으로 방대한 량의 계산을 할 수 있게 되었지만 아직도 효율적으로 학습할 수 있는 방법이 필요하다[2].

경사하강법(gradient descent method)은 신경망을 학습하는 데 현재 가장 많이 쓰이는 알고리즘이다. 학습이란 손실함수가 최소값이 되도록 매개변수를 갱신하는 것이다. 손실함수는 실제값과 예측값의 차이를 수치화해주는 함수이다. 경사하강법은 두 값의 차이 즉 오차가 최소화되도록 매개변수를 갱신하는데 손실함수의 기울기를 사용하는 것으로 현재 최고의 딥러닝 학습알고리즘을 제공하는 라이브러리에서 사용되고 있다. 그러나 이 알고리즘들은 블랙박스형태로 제공되고 있어서 다양한 경사하강법들의 장단점을 파악하는 것이 쉽지 않다. 따라서 효율적인 학습을 위하여 데이터의 특성에 맞는 학습알고리즘을 선택하는 것은 그리 간단하지 않다.

본 논문에서는 딥러닝학습에 사용되는 경사하강법들의 특성을 분석하려고 한다. 2장에서는 다층신경망의 학습에 대하여 간단히 설명하고 3장에서는 확률적 경사하강법(Stochastic Gradient Descent method), 모멘텀법(Momentum method), AdaGrad법 그리고 Adadelta법의 특성에 대하여 분석하고 4장에서는 MNIST 데이터셋을 이용하여 실험한 결과들을 비교분석하려고 한다.

2. 다층신경망 학습

2.1 다층신경망 구조

다층신경망은 입력층, 은닉층 그리고 출력층으로 이루어진다. Fig. 1에서는 은닉층은 2층으로 구성되었지만 필요에 따라 더 많은 층을 첨가할 수 있다. 그리고 뉴런과 뉴런을 연결하는 것은 가중치(w)이다.

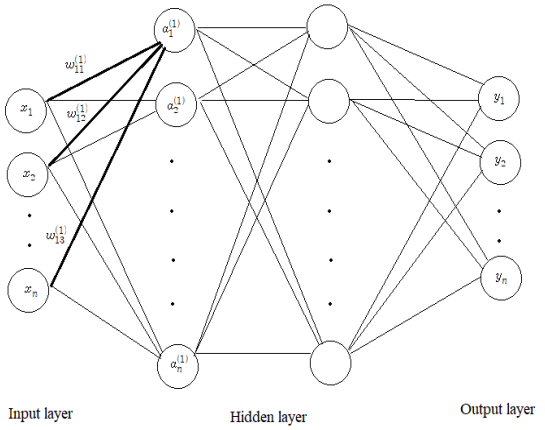

Fig. 1. Structure of Neaural Networks

첫 번째 은닉층에서 i번째 뉴런에 전달되는 신호 a_i^1는 이 뉴런의 입력에 연결된 입력층의 모든 값에 가중치를 곱하여 다음과 같이 구한다.

$$a_i^1 = \sum_{j=1}^{n} w_{ij}^{(1)} x_j \tag{1}$$

가중치 $w_{ij}^{(1)}$는 첫 번째 은닉층에서 i번째 뉴런과 입력층의 j번째 뉴런을 연결한다. 뉴런의 활성화 함수로는 주로 시그모이드(sigmoid) 함수와 ReLu(rectifired linear unit) 함수가 사용되며, 시그모이드 함수는 다음과 같고

$$f(a) = \frac{1}{1+e^{-a}} \tag{2}$$

ReLu함수는 다음과 같이 표현된다.

$$f(a) = \begin{cases} a & (a > 0) \\ 0 & (a \le 0) \end{cases} \tag{3}$$

2.2 신경망 학습

신경망을 학습하는 과정은 손실함수를 감소하는 방향으로 가중치를 업데이트한다. 가장 많이 쓰이는 손실함수는 평균제곱오차와 교차엔트로피오차이다. 평균제곱오차는 다음과 같고,

$$J = \frac{1}{2}\sum_k (y_k - t_k)^2 \tag{4}$$

교차엔트로피오차는 다음과 같다.

$$J = -\sum_k t_k \log y_k \tag{5}$$

여기서 y_k는 신경망의 출력, t_k는 참값이고 k는 데이터의 차원수를 나타낸다. 신경망의 학습은 신경망의 출력(y_k)이 참값(t_k)에 가깝게 되도록 즉 손실함수가 최소가 되도록 최적의 가중치(w)를 찾아가는 과정이다[4].

3. 신경망 학습 알고리즘

3.1 미니배치 확률적 경사하강법

경사하강법은 매개변수 w에 대한 손실함수의 기울기를 이용하여 손실함수가 감소되도록 새로운 매개변수를 갱신하는 방법으로 다음과 같이 표현된다[5].

$$w = w - \eta \cdot \nabla_w J(w) \tag{6}$$

여기서 η는 학습률이다.

경사하강법에는 세 종류가 있다 즉, 전체경사하강법, 확률적 경사하강법 그리고 미니배치 확률적 경사하강법이다.

전체 경사하강법은 매개변수를 한 번 갱신하는 데 전체 데이터 세트를 사용한다. 하지만 빅데이터의 규모는 수십억이 넘는 경우가 많다. 이런 경우는 계산하는 데 너무 많은 시간이 걸려서 이용하는 것이 현실적이지 않다.

확률적 경사하강법은 매개변수를 갱신하기 위하여 무작위로 샘플링된 한 개의 x(레이블 t)에 대하여 다음과 같이 구한다.

$$w = w - \eta \cdot \nabla_w J(w; x^{(i)}; t^{(i)}) \tag{7}$$

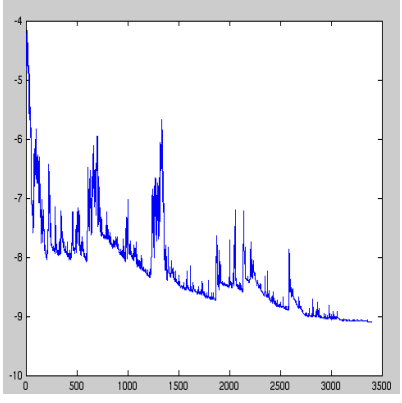

Fig. 2. SGD fluctuation(Source: Wikipedia)

이 방법은 훨씬 적은 계산으로 적절한 기울기를 얻을 수 있지만 노이즈가 매우 심하다. 따라서 Fig. 2처럼 너무 자주 매개변수를 갱신하면서 손실함수가 심하게 파동을 일으킬 수 도 있다. 이로 인해 확률적 경사하강법은 심한 파동에 의하여 부분극소점(local minimum)으로 수렴할 가능성도 있다[4].

미니배치 확률적 경사하강법(미니 배치 SGD)는 전체 배치 경사하강법과 확률적경사하강법간의 절충안이다. 배치크기를 n으로 하여 다음과 같이 매개변수를 갱신한다.

$$w = w - \eta \cdot \nabla_w J(w; x^{(i:i+n)}; t^{(i:i+n)}) \tag{8}$$

미니 배치는 일반적으로 무작위로 선택한 배치크기를 주로 50개에서 256개 사이로 구성한다. 미니 배치 확률적 경사하강법은 확률적 경사하강법보다는 노이즈를 줄이고 안정적으로 수렴하며 전체배치 경사하강법에 비해서는 계산량을 많이 줄일 수 있다.

3.2 모멘텀법

확률적 경사하강법의 단점은 비등방성함수에서 즉, 방향에 따라 자주 기울기가 달라지는 함수에서 탐색경로가 지그재그로 되어 비효율적이다. 모멘텀 알고리즘은 이런 단점을 보안하기 위하여 제안된 알고리즘이다. 즉 기존 탐색방향을 고려하여 매개변수를 갱신함으로써 진동하는 것을 방지할 수 있다[5].

$$\begin{aligned} v_{t+1} &= \gamma v_t + \eta \cdot \nabla_w J(w) \\ w &= w - v_{t+1} \end{aligned} \tag{9}$$

여기서 γ는 모멘트 효과에 대한 가중치이다.

v_t는 0 으로 초기화되어 있고 반복이 될 때마다 현재의 그래디언트 $\eta \cdot \nabla_w J(w)$가 다음번 모멘트 v_{t+1}에 누적된다. 경사하강법에서 모멘트 항이 추가된 것이다.

3.3 AdaGrad법

경사하강법에서 데이터의 특성에 따라 효과적인 학습률(η)을 선택하는 것은 중요하다. 이런 점에 아이디어를 둔 것이 AdaGrad(Adaptive Gradient)이다. AdaGrad는 변수들을 갱신할 때 각각의 변수마다 학습률을 다르게 조정한다[6].

학습률의 조정은 지금까지 많이 변화한 변수들은 최저치에 근접했을 확률이 높을 것으로 보고 학습률을 작게 하여 세밀하게 최저치에 도달하도록 하고, 적게 변화한 변수들은 최저치 값에 도달하기 위해서는 많이 이동해야 할 확률이 높기 때문에 학습률을 크게한다. AdaGrad의 갱신 방법은 다음과 같다.

$$G = G + \nabla_w J(w) \odot \nabla_w J(w) \tag{10}$$
$$w = w - \frac{\eta}{\sqrt{G+\varepsilon}} \odot \nabla_w J(w)$$

여기서 $\odot$의 기호는 행렬의 원소별 제곱을 의미한다. 이때 ε는 0으로 나누는 것을 방지하기 위한 작은 값이다.

3.4 Adadelta법

Adadelta법은 AdaGrad법의 문제점을 보안하여 제안된 알고리즘이다. AdaGrad법에서 학습을 계속 진행하면 G는 계속 증가하기 때문에 변화율이 너무 작아져서 결국 거의 움직이지 않게 된다. 이를 보완하기 위하여 지수이동평균개념과 이차근사법 개념을 도입한다. 매개변수 w에 대하여 손실함수의 기울기를 다음과 같이 정의하고

$$g_t = \nabla_w J(w_t) \tag{11}$$

g_t에 대하여 지수이동평균수식을 도입하면 다음과 같다.

$$E[g^2]_t = (1-\gamma) E[g^2]_{t-1} + \gamma g_t^2 \tag{12}$$

여기서 가중치 γ는 다음과 같고, n은 윈도우 사이즈이다.

$$\gamma = \frac{2}{n+1} \tag{13}$$

만약 $n = 19$이면 $\gamma = 0.1$이 되고, 최근 19개 정도의 데이터가 가중평균되는 결과가 된다. AdaGrad법에서 G 대신 $E[g^2]$를 사용하면 다음과 같다.

$$w = w - \frac{\eta}{\sqrt{E[g^2]+\varepsilon}} \odot \nabla_w J(w) \tag{14}$$

이 식은 RMSprp법이라 하는 또 다른 AdaGrad법을 보완한 방법이다. 이런 명칭은 식(14)의 분모항 ($\sqrt{E[g^2]+\varepsilon}$)이 RMS(root mean squared) 표준에 해당하기 때문이다. Adadelt법은 식(14)에서 고정학습률 η대신에 매개변수증가률($\triangle w$)을 지수이동평균값으로 다음과 같이 대체한 것이다.

$$w = w - \frac{\sqrt{E[\triangle w^2]+\varepsilon}}{\sqrt{E[g^2]+\varepsilon}} \odot \nabla_w J(w) \tag{15}$$

또한 Adadelt법은 다른 경사하강법들과 달리 이차근사법을 이용한 보다 발전된 식으로 해석할 수 있다[7]. 이는 식(15)에서 다음과 같이 RMS 값들을 대체하면

$$\frac{dJ}{dw} \simeq \sqrt{E[g^2]}\ ,\ dw \simeq \sqrt{E[\triangle w^2]} \tag{16}$$

식(15)는 다음과 같이 간주할 수 있다.

$$w \simeq w - \frac{dw}{\frac{dJ}{dw}} J'(w) = w - \frac{1}{\frac{d^2 J}{dw^2}} J'(w) \tag{17}$$

따라서 다음과 같은 뉴턴의 이차근사법과 같은 형태가 된다[8].

$$w = w - \frac{J'(w)}{J''(w)} \tag{18}$$

4. 실험 및 결과

실험은 확률적경사하강법(SGD), 모멘텀법, AdaGrad법 그리고 Adadelta법의 특성 및 성능을 비교분석하기 위하여 다층 신경망을 이용하였다. 실험은 손으로 쓴 숫자 이미지를 판단하는 신경망을 학습시키는 것이다. MNIST 데이터 셋은 인공지능 연구의 권위자 LeCun교수가 만든 것이고 신경망을 검증하는 데 널리 사용된다. MNIST는 손으로 쓴 숫자 이미지이며 숫자는 0에서 9까지의 값을 갖는 고정 크기 이미지 (28x28 픽셀)로 크기 표준화되어 있다. MNIST는 60,000개의 트레이닝 셋과 10,000개의 테스트 셋으로 이루어져 있고 이중 트레이닝 셋을 학습데이터로 사용하고 테스트 셋은 신경망을 검증하는 데에 사용한다. 신경망은 입력층, 은닉층 그리고 출력층으로 구성되었다. 입력층은 (28x28)크기 이미지를 위하여 784개의 뉴런으로 구성되었으며, 은닉층은 2개의 층으로 첫 번째층은 500개 그리고 두 번째층은 300개의 뉴런으로 구성하였다. 출력층은 숫자 0에서 9까지의 값을 판정하기 위하여 10개의 뉴런으로 구성되었다. 출력층의 활성화함수는 소프트맥스함수이고 나머지 입력층과 은닉층의 활성화함수는 ReLu함수를 사용하였다. 그리고 손실함수는 교차 엔트로피 오차를 사용하였다.

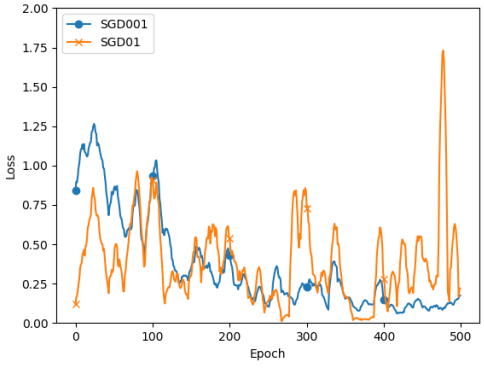

Fig. 3. Fluctuation of SGD for η=0.01 and η=0.1

Fig. 3은 확률적 경사하강법에서 파동이 일어나는 것을 알 수 있다. 학습률 η=0.01에서는 파동은 일어나지만 점차적으로 수렴하는 것을 보여준다.

Fig. 4는 미니 배치 확률적 경사하강법을 적용한 그림으로 배치크기는 256이다. 파동이 거의 없어 졌고 두 학습률에서 모두 수렴하였고 학습률 η=0.1인 경우가 수렴을 더 빠르다.

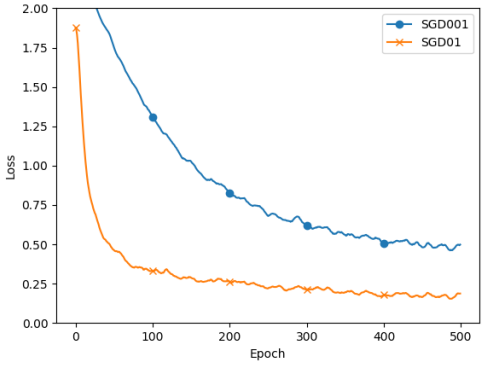

Fig. 4. Test of mini batch SGD: batch size=256, learning rate η=0.1 and 0.01

Fig. 5는 동일한 학습률 η=0.01에서 다양한 모멘텀에 따른 모멘텀법이 수렴과정을 보여준다.

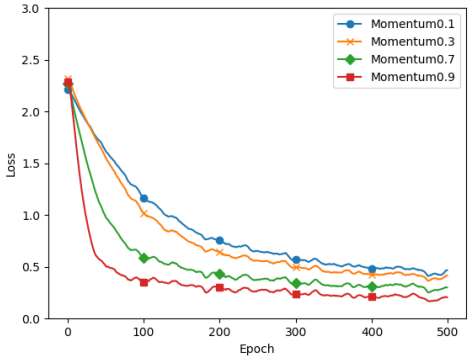

Fig. 5. Test of Momentum for case momentum =0.1, 03, 0.5, 0.7

Fig. 6은 가중치 γ에 따른 Adadelta법을 적용하여 테스트한 그림이다. 즉 윈도우 크기에 따른 차이를 보려고 하였으나 별 차이는 없고, 윈도우 크기가 크면 미세하지만 수렴속도가 더 빨랐다.

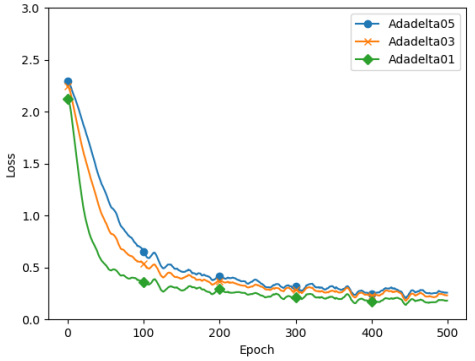

Fig. 6. Test of Adadelta for weight γ=0.5, 03, 0.1

Fig. 7은 학습률 η=0.01에서 4가지의 경사하강법을 비교한 그림이다. SGD는 미니배치 확률 경사하강법으로 배치크기는 256이다. 모멘텀법에서는 모멘텀이 0.9이다. Adadelta에서 가중치 γ=0.1에서 윈도우 크기는 19이다. Fig. 7에서 알 수 있듯이 AdaGrad와 Adadelta는 크게 차이는 없으나 수렴이 진행되면서 미세하나마 Adadelta가 우위를 보였다.

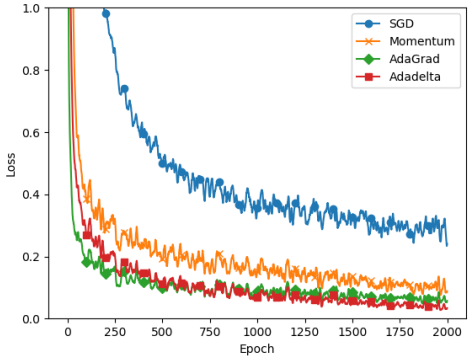

Fig. 7. Test of SGD, Momentum, AdaGrad and Adadelta

5. 결론

본 논문에서는 신경망 학습 알고리즘들을 MNIST 데이터 세트에 적용하여 실험하고 분석하였다. 확률적 경사하강법에서 배치크기가 1인 경우는 파동이 심하게 발생하고, 학습률(η)이 η=0.01에서는 수렴하지만 0.1인 경우는 수렴하지 않았다. 배치크기가 256인 경우에는 모든 학습률에서 수렴함을 보였다.

모멘텀법은 물리의 관성성질을 이용하여 확률적 경사하강법의 파동현상을 보완하기 위해 사용할 수 있다. 그

러나 실험에서는 큰 차이를 나타내지 않았다. 다만 모멘텀이 클수록 수렴속도가 빠름을 보였다.

AdaGrad법과 Adadelta법은 학습을 진행하면서 효과적으로 학습률을 조절하기 위하여 사용된다. Adadelta법은 갱신된 매개변수들의 윈도우 크기를 조절하여 AdaGrad법이 최종적으로 학습률이 너무 작게 되는 것을 방지하기 위하여 제안되었고 2차 근사법특성이 가미된 알고리즘으로 기대가 많았으나 실험에서는 두 방법이 크게 차이는 없었다. 이는 향후 다른 데이터세트에서 실험하여 재확인 할 필요가 있다고 판단된다.

References

[1] Smith, Craig S, "The Man Who Helped Turn Toronto into a High-Tech Hotbed". The New York Times. Retrieved 27 June 2017.

[2] J. Liang, E. Meyerson, and R. Miikkulainen. Evolutionary architecture search for deep multitask networks. arXiv preprint arXiv:1803.03745, 2018.

[3] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. Le, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Ng, “Large scale distributed deep net works,” in NIPS, 2012.

[4] T. Schaul, S. Zhang, and Y. LeCun, “No more pesky learning rates,” arXiv:1206.1106, 2012.

[5] N. Jaitly, P. Nguyen, A. Senior, and V. Vanhoucke, “Application of pretrained deep neural networks to large vo cabulary speech recognition,” in Interspeech, 2012.

[6] G. Morse and K. O. Stanley. Simple evolutionary optimization can rival stochastic gradient descent in neural networks. In The Genetic and Evolutionary Computation Conference (GECCO), pages 477-484, 2016.

[7] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” in COLT, 2010.

[8] S. Becker and Y. LeCun, “Improving the convergence of back-propagation learning with second order methods,” Tech. Rep., Department of Computer Science, University of Toronto, Toronto, ON, Canada, 1988.

강 민 제(Min-Jae Kang) [정회원]

- 1982년 2월 : 서울대학교 전기공학과(공학사)
- 1991년 2월 : 미국 루이빌대 전기공학과(공학박사)
- 2003년 2월 ~ 2004년 2월 : 미국 일리노이주립대학 방문교수
- 1992년 3월 ~ 현재 : 제주대학교 전자공학과 교수

〈관심분야〉
신경회로망, ERT영상복원, 접지시스템 설계