특집 최신 미디어와 인공지능

Edge Device를 위한 시각 인식 모델

□ 최종현 / 광주과학기술원

요약

AI 시스템은 우리 생활 전반에서 다양한 예측을 도와주는 장치로써 그 중요성이 크다. AI 시스템의 활용도는 AI 장치가 얼마나 우리 생활 전반에 다각도로 이용되어야 하는지에 달려있다. 현재 AI 시스템은 높은 정확도를 위해 학습과 추론에 고성능 컴퓨팅 장비를 필요로 한다. 고성능 장치를 우리 생활 저변에서 쉽게 설치하고 사용할 수 없기 때문에, AI 시스템을 우리 생활에 사용하기 위해서 크게 두 가지의 접근법을 사용하고 있다. 첫째, 고성능 네트워크와 고성능 컴퓨팅 서버를 사용하여 end-user 장치의 계산 복잡도를 최소화하는 시스템을 설계할 수 있다. 둘째, AI 시스템의 학습 및 추론 효율성을 높여, 서버와 네트워크 없이도 end-user 장치에서 최선의 성능을 내는 시스템을 설계할 수 있다. 첫번째 접근법은 고성능 네트워크의 발전을 수반하고, 네트워크의 항상성을 전제로 하기 때문에, 실현하는데 많은 시간과 자원이 요구된다. 두번째 접근법은 비용-효율적이긴 하나 첫번째 접근법에 비해 AI 시스템의 성능이 다소 떨어질 수 있다. 이 글에서는 두번째 접근법의 AI 시스템, 특히 시각 인식 시스템을 응용으로 하는 기술들을 살펴보도록 하겠다.

I. 서 론

AI 시스템의 진가는 우리의 일상 생활에서 다양한 형태의 예측을 정확하고 빠르게 할 수 있게 도와줄 때에 발휘된다. 다만, AI 시스템이 높은 정확도로 예측하기 위해서 학습과 추론 과정에 고성능 컴퓨팅 자원이 요구된다는 점이 AI 시스템을 다양한 일상 생활 속에서 적용하는 것에 큰 걸림돌이 되고 있다. AI 시스템을 다양한 형태로 적용(deploy)할 때 크게 두 가지 접근법을 사용한다고 요약할 수 있다. 첫째, 고성능 네트워크와 고성능 컴퓨팅 서버를 사용하여 end-user 장치의 계산 복잡도를 최소화하는 시스템을 설계할 수 있다. 둘째, AI 시스템의 학습 및 추론 효율성을 높여, 서버와 네트워크 없이도 end-user 장치에서 최선의 성능을 내는 시스템을 설계할 수 있다.

첫번째 접근법은 유/무선의 네트워크가 늘 end-user 장치에 연결되어야 하고, 전력소모가 많다. 그러나 높은 성능의 학습/추론 성능을 기대할 수 있다. 이러한 적용법은 end-user 장치의 크기가 크고 성능이 중요한 경우에 주로 사용한다. 장치의 크기와 네트워크 availability는 첫번째 접근법을 사용한 AI 시스템의 활용도를 크게 저하시킨다. 두번째 접근법은 네트워크가 필요 없고 네트워크로 인한 전력 소모가 없기 때문에, 거의 항상 AI 시스템의 추론 결과를 이용할 수 있는 큰 장점이 있으나, AI 학습/추론 성능이 end-user 장치의 연산 성능과 비례하기 때문에, 일반적으로 첫번째 접근법에 비해 정확도가 낮다는 치명적인 단점이 있다. 최근에는 이 두가지 접근법을 결합하여, end-user 장치의 성능을 높이고 네트워크의 부담을 낮추는 기법 모두를 사용하는 접근법도 널리 연구되고 있다.

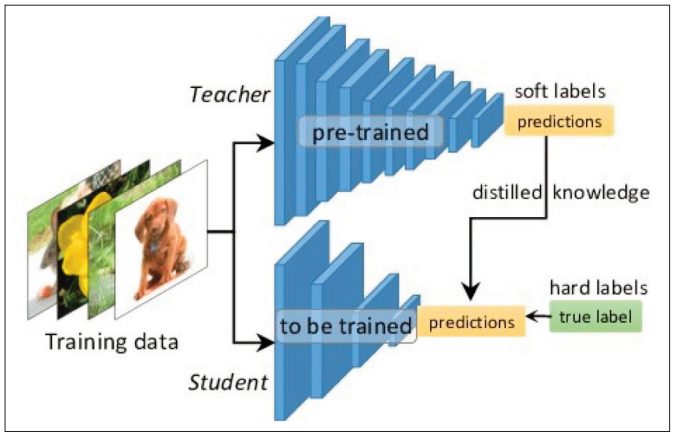
이 글에서는 두번째 접근법, 즉, end-user 장치에서 AI 시스템의 성능을 높일 수 있는 방법들에 대해 다루고자 한다. 이 방법들을 크게 세가지의 카테고리로 분류하였으며, 이 방법들은 기계 학습을 사용하는 다양한 AI 어플리케이션에 이용이 가능하나, 주로 시각 인식 알고리즘을 중심으로 리뷰하고자 한다.

II. 지식 증류 기법 (Knowledge Distillation)

지식 증류 기법은 기존에 학습된 네트워크(이하 teacher)가 학습한 지식을 전달받아서 새로운 네트워크(이하 student)를 학습하는 방법론이다. 보통 student 네트워크는 teacher에 비해 네트워크 크기(number of parameter, number of layers 등)가 작은 경우가 대부분이며, 크기가 작은 네트워크가 높은 성능의 정확도를 갖게 학습하는 방법이다.

1. 최초의 지식 증류 기법 (Vanilla Knowledge Distillation)

최초의 지식 증류 기법은 2015년에 발표된 이후



<그림 1> 지식증류기법

에 이미지 분류 등의 분야에서 매우 활발하게 응용되고 있다(Hinton et al., 2015). 〈그림 1〉은 지식 증류 지법을 도식화한 그림이다. 기본 아이디어는 softmax operation을 거치기 전의 soft label (또는 logit)이 학습된 네트워크의 정보를 풍부하게 담고 있다고 보고, 이 soft label과 학습용 데이터 (training set or training split)를 함께 사용하여 student를 학습하는 방식이다. Student 네트워크는 보통 teacher보다 용량이 작은 네트워크를 사용함으로써, 네트워크 압축 효과를 얻을 수 있다.

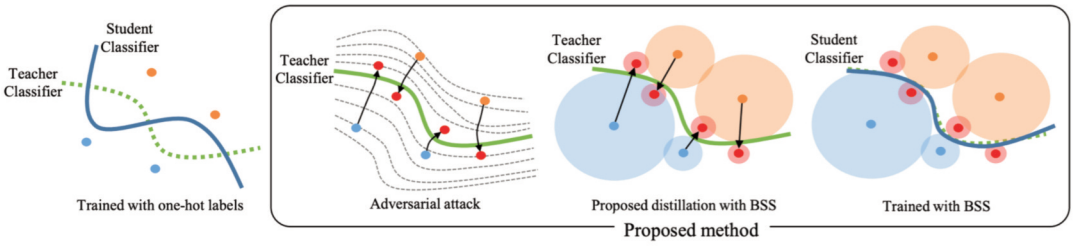
보통의 경우 지식 증류를 통해 학습한 student는 teacher보다 정확도가 낮게 마련이나, 최근에는 이를 개선하여 teacher의 정확도를 상회하는 student를 학습하는 방법이 몇 가지 제안되었다.

2. Teacher 네트워크 성능을 상회하는 지식 증류 기법

A Trivial Solution: Teacher 네트워크의 성능을 높이는 가장 간단하지만 지식 증류의 효과가 필요 없는 방법은 student 네트워크를 용량이 작으면서 학습 효율이 좋은 것을 사용하면 된다. 예를 들어 teacher 네트워크에 ResNet-18이 사용되었다면, student 네트워크에 MobileNet V2를 사용하는 경우, MobileNet V2에서 지식 증류 없이도 ResNet-18보다 좋은 학습 성능을 보이면서 용량은 작기 때문에, 지식 증류 덕분이 아니라 그 자체의 학습 효과로 더 좋은 성능을 내는 네트워크를 학습할 수 있다.

Student 네트워크의 학습 성능이 teacher보다 명백히 나쁜 경우라면, teacher의 학습 성능을 상회하는 student, 즉 '청출어람'하는 네트워크를 학습하기가 쉽지 않다. 이를 위해 최근에 제안된 방법 중 한 가지를 소개하고자 한다.

Teacher 네트워크의 decision boundary는 training set에 있는 sample들과 그들이 있을 법한 feature space상의 지역들로 결정된다. Student 네트워크를 지식 증류 기법을 활용하여 학습하는 시점에, teacher 네트워크의 decision boundary를 더 잘 보존할 수 있는 sample들을 적대 학습(adversarial learning) 기법을 사용하여 생성하고, 생성된 sample들을 'boundary supporting sample'이라 부른다. 이 boundary supporting sample들과 기존의 학습 sample들과 합쳐서 학습할 경우 학습 정확도가 높아지는 방법이다(Heo et al., 2019).



The concept of knowledge distillation using samples close to the decision boundary. The dots in the figure represent the training sample and the circle around a dot represents the distance to the nearest decision boundary. The samples close to the decision boundary enable more accurate knowledge transfer.

<그림 2> 적대적 샘플 생성을 통한 지식 증류 기법

〈그림 2〉는 서술된 방법을 모사한 그림이다.

Ⅲ. 네트워크 가지치기(Pruning)

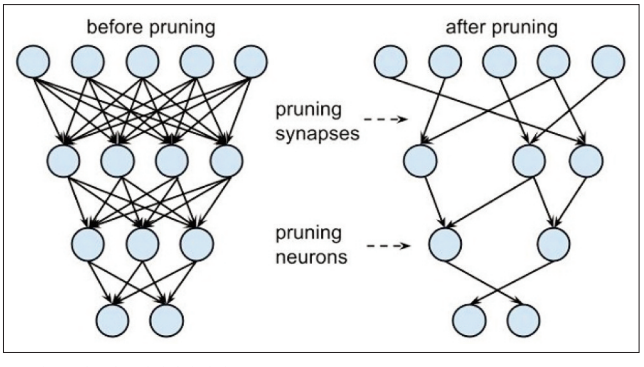
네트워크 가지치기는 현재 사용중인 deep network들이 과-파라메터화(overparameterized) 되어 있다고 가정한다. 즉, 필요한 인식 성능을 내기 위해 모든 parameter가 필요하지 않다는 사실로부터, 인식에 꼭 필요하지 않은 parameter들을 가지치기 하여, 학습과 추론의 정확도와 효율을 높이는 것을 목표로 한다. 〈그림 3〉은 가지치기 방법을 도식화 한 그림이다. 그 중에서 고전적인 방법론 중 하나인 Optimal Brain Damage(OBD) 방법과 최근에 발표된 방법들을 살펴보고자 한다.

1. Optimal Brain Damage(OBD)

Optimal Brain Damage(OBD) 방법은 네트워크를 일반적으로 학습한 뒤에, 학습된 parameter 또는 weight 값의 '중요도'를 각각의 weight값이 약간의 움직임(perturbation)으로 변화할 경우 loss 함수 값의 변하는 양(loss가 적게 변하는 parameter가 중요하지 않은 것)을 바탕으로 중요하지 않은 parameter 또는 weight들을 없애고, 다시 학습하는 iteration을 반복하는 방식으로 네트워크 가지치기(weight를 0으로 만드는)를 수행한다(LeCun et al., 1990). 이 연구에서 중점적으로 다룬 부분은 loss의 변화량을 계산하는 복잡한 수식을 2차 Taylor 전개로 근사하여 계산 복잡도를 현저히 줄인 것에 있다.

2. Deep Neural Network을 위한 가지치기 방법들

OBD 방법이 발표된지 오랜 시간이 지난 이후에 deep neural network에서 네트워크 가지치기를 하는 방법들이 발표되기 시작하였다. OBD 방법과 근본적으로 같은 방식의 가지치기를 사용하지만, 각 weight의 perturbation에 의한 loss의 변화량을 계산하지 않고, 단순히 weight의 magnitude를 중요도로 사용하여 가지치기를 하되, 가지치기 연산을 할 때마다 학습을 잘 시켜준다면 가지치기 후에도 높은 성능을 얻는 네트워크를 얻을 수 있다는 방법



<그림 3> 네트워크 가지치기 방법

<표 1> AlexNet에서의 Han et al.의 가지치기 방법(Network Pruning)

Network	Top-1 Error	Top-5 Error	Parameters	Compression Rate
Baseline Caffemodel [26]	42.78%	19.73%	61.0M	1×
Data-free pruning [28]	44.40%	-	39.6M	1.5×
Fastfood-32-AD [29]	41.93%	-	32.8M	2×
Fastfood-16-AD [29]	42.90%	-	16.4M	3.7×
Collins & Kohli [30]	44.40%	-	15.2M	4×
Naive Cut	47.18%	23.23%	13.8M	4.4×
SVD [12]	44.02%	20.56%	11.9M	5×
Network Pruning	**42.77%**	**19.67%**	**6.7M**	**9×**

이 발표되었다(Han et al., 2015). (AlexNet에서 비교한 〈표 1〉 참조)

최근 어떤 중요도 추정 방식을 사용하든, 가지치기 후 재-학습이 잘 된다면 가지치기 후의 네트워크의 정확도가 크게 차이나지 않는다는 연구 결과도 발표되었다(Mittal et al., 2018). 〈표 2〉는 VGG-16 네트워크 구조에서 가지치기 할 weight를 고르는 방법에 따른 가지치기 후 네트워크의 이미지 분류 정확도를 보여준다.

<표 2> Mittal et al.의 다양한 weight 중요도에 따른 가지치기 정확도 비교

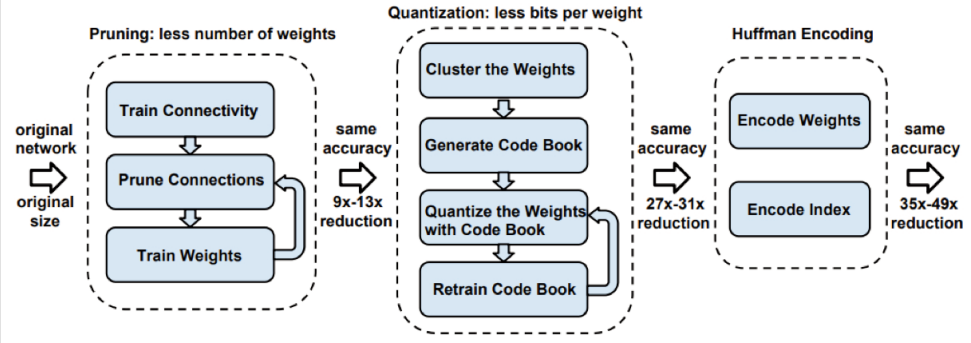
Heuristic	25 %	50%	75%
Random	0.650	0.569	0.415
Mean Activation	0.652	0.570	0.409
Entropy	0.641	0.549	0.405
Scaled Entropy	0.637	0.550	0.401
l_1-norm	**0.667**	**0.593**	**0.436**
APoZ	0.647	0.564	0.422
Sensitivity	0.636	0.543	0.379

Comparison of different filter pruning strategies on VGG-16

Ⅳ. 파라미터 양자화 (Parameter Quantization)

네트워크 가지치기 방법에서는 원하는 정확도를 갖는 네트워크를 학습하기 위해 모든 parameter가 필요하지 않을 수 있다는 동기에서 출발하는 연구라면, 양자화는 원하는 정확도를 갖는 네트워크를 학습하기 위해 각 parameter의 precision (예, 32 bit)이 모두 필요하지 않을 수 있다는 동기에서 출발한 연구이다. Double precision (64 bit)의 parameter가 더 높은 정확도를 갖는 수 표현법을 이용하므로, 미분 등의 연산에서 더욱 정확한 값을 저장할 수 있게 해주어 학습의 정확도도 높아질 것이라고 예측할 수 있다. 그러나 single precision (32 bit)의 parameter가 double precision에 비해 정확도 손실은 무시할 수준이면서 학습 효율은 크게 높으므로 일반적으로 많이 사용되고 있다.

Double precision에서 single precision으로의 변화처럼 비교적 무시할 만한 granularity의 표현법 차이가 아니라 32 bit에서 8 bit, 4 bit 또는 극단적으로 1 bit으로 바꾸는 양자화 레벨의 변화에도 정확도를 보존하는 네트워크를 학습하기 위한 네트워크의 구조나 학습방법이 효율적인 네트워크를 만들기 위해 연구되고 있다. 또한 고정된 양자화 레벨을 사용하는 것이 아니라 가변적인 양자화 레벨을 통해 정확도와 네트워크 크기의 trade-off를 더 잘 조절하는 네트워크를 학습하는 방법도 제안되고 있다.



The three stage compression pipeline: pruning, quantization and Huffman coding. Pruning reduces the number of weights by 10×, while quantization further improves the compression rate: between 27× and 31×. Huffman coding gives more compression: between 35× and 49×. The compression rate already included the meta-data for sparse representation. The compression scheme doesn't incur any accuracy loss.

<그림 4> Han et al.의 Deep Compression

1. Deep Compression 방법

네트워크의 parameter 또는 weight들을 비슷한 값을 갖는 그룹으로 클러스터링 하는 방법을 통해 양자화하는 방법으로, 각 network를 weight 클러스터의 무게중심 값으로 표현하여 네트워크의 크기를 획기적으로 줄이는 방식이다. 이미 가지치기가 된 네트워크에 클러스터리 기반의 양자화 방법과 허프만 코딩을 사용하여 추가적으로 네트워크 크기를 줄이게 되면, 정확도는 무시할만한 손실을 보이지만 네트워크 크기를 큰 폭으로 압축할 수 있게 된다.

가지치기를 통해 얻은 압축율(13배)에서 양자화와 허프만 코딩을 추가적으로 사용하면 최대 총 49배까지 압축율을 얻을 수 있다. 즉, 36배 정도의 추가적인 압축율을 얻을 수 있게 된다(Han et al., 2015). 〈그림 4〉는 가지치기, 양자화, 허프만 코딩을 모두 사용한 deep compression방식의 모식도이다.

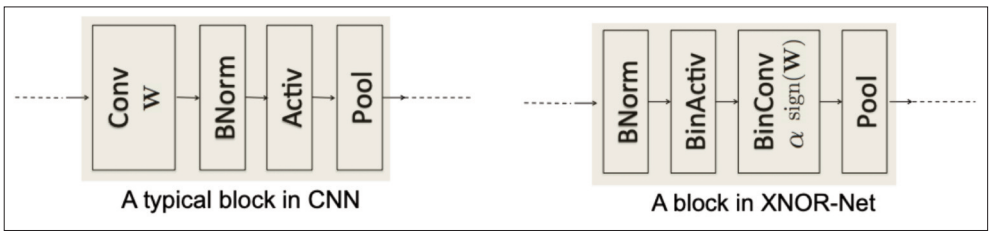
2. 극단적인 양자화 네트워크 – Binary Neural Networks

가장 극단적인 형태의 양자화는 32 bit single precision의 parameter를 1 bit로 변환하는 것이다. Parameter가 1 bit가 되면 곱셈과 합 연산으로 이루어진 합성곱(convolutional) 연산은 bit 간의 XNOR 연산과 bit-counting 연산으로 대체할 수 있게 되어 엄청난 계산 효율을 얻을 수 있다. 그러나 이런 binary parameter는 정확도에서 큰 손실을 보게 되어, 실제 사용하기에 많은 제약이 있다.

최근 binary parameter를 가지고도 좋은 성능을 내는 네트워크 구조를 만들려는 연구가 많이 있다. 그 중 매우 초창기 연구로 gradient의 값을 특정 threshold로부터 이진화(binarize)하여 back-propagation하여 parameter와 activation map 모두를 이진화 하는 Binarized Neural Network (BNN)이라는 연구가 있다(Hubara et al., 2016). BNN은 이론적으로 32배의 적은 메모리와 23배 빠른 처리속도를 낼 수 있다. 〈표 3〉은 BNN의 정확도 성능으로써, MNIST, SVHN, CIFAR-10 등의 작은 데이터셋에서 floating point 버전에 비해 정확도 손실이 거의 없는 것을 확인할 수 있다.

<표 3> Classification test error rates of DNNs trained on MNIST(fully connected architecture), CIFAR-10 and SVHN (convnet). No unsupervised pre-tranining or data augmentation was used.

Data set	MNIST	SVHN	CIFAR-10
Binarized activations+weights, during training and test			
BNN (Torch7)	1.40%	2.53%	10.15%
BNN (Theano)	0.96%	2.80%	11.40%
Committee Machines' Array (Baldassi et al., 2015)	1.35%	-	-
Binarized weights, during training and test			
BinaryConnect (Courbariaux et al., 2015)	1.29± 0.08%	2.30%	9.90%
Binarized activations+weights, during test			
EBP (Cheng et al., 2015)	2.2± 0.1%	-	-
Bitwise DNNs (Kim & Smaragdis, 2016)	1.33%	-	-
Ternary weights, binary activations, during test			
(Hwang & Sung, 2014)	1.45%	-	-
No binarization (standard results)			
No regularization	1.3± 0.2%	2.44%	10.94%
Gated pooling (Lee et al., 2015)	-	1.69%	7.62%



<그림 5> ResNet CNN block과 XNOR-Net CNN block의 비교

비슷한 시기에 발표된 floating point weight를 갖는 네트워크 중에서 매우 성공적인 ResNet(He et al., 2016)이라는 네트워크 구조로부터 이진화 네트워크로 사용하기 위한 중요한 아키텍처적인 아이디어가 발표되었다. 이 새로운 이진 네트워크 구조는 XNOR-Net이라는 이름으로 발표되었으며, 〈그림 5〉와 같이 ResNet block에서 convolutional layer와 batch normalization 및 activation layer의 순서를 바꾸므로써 ResNet 구조에서 이진화 네트워크의 학습을 가능하게 하였다(Rastegari et al., 2016).

XNOR-Net은 large scale 데이터 셋인 Image-

<표 4> This table compares the final accuracies (Top1 - Top5) of the full precision network with our binary precision networks; Binary-Weight-Networks(BWN) and XNOR-Networks(XNOR-Net) and the competitor methods; BinaryConnect(BC) and BinaryNet(BNN).

Classification Accuracy(%)									
Binary-Weight				Binary-Input-Binary-Weight				Full-Precision	
BWN		BC[11]		XNOR-Net		BNN[11]		AlexNet[1]	
Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
56.8	**79.4**	35.4	61.0	**44.2**	**69.2**	27.9	50.42	56.6	80.2

Net에서 기존의 BNN대비 큰 폭의 성능 향상을 보이며 주목을 받았다. XNOR-Net 저자들은 이 아이디어로 xnor.ai 라는 스타트업을 창업하였으며, 2020년 1월 Apple에 2천억원 이상의 금액에 합병되어 exit 하였다.

V. 결 론

AI 시스템을 우리 생활에서 밀접하게 사용하기 위한 계산-효율적인 시각 인식 모델들에 대해 살펴보았다. 구체적으로 지식 증류 기법, 네트워크 가지치기, 네트워크 양자화 등의 관점에서 계산 효율적인 시각 인식 모델들을 소개하였다. 현재 발표되고 있는 최신 성능의 AI 알고리즘이 우리 생활에서 유용하게 사용될 수 있도록, 계산 효율적인 AI 모델들을 다루는 이 분야는 연구할 부분이 많이 남아 있으며 그 활용도가 매우 높다는 점에서 연구자들이 주목할 필요가 있다.

참 고 문 헌

[1] G. Hinton et al., “Distilling the Knowledge in a Neural Network”, arXiv 2015
[2] B. Heo et al., “Knowledge Distillation with Adversarial Samples Supporting Decision Boundary”, AAAI 2019
[3] Y. LeCun et al., “Optimal Brain Damage”, NeurIPS 1990
[4] S. Han et al., “Learning both Weights and Connections for Efficient Neural Networks”, NeurIPS 2015
[5] I. Hubara et al., “Binarized Neural Networks: Training Neural Networks with Weights and Activations Constrained to +1 or -1,” NeurIPS 2016
[6] K. He et al., Deep Residual Learning for Image Recognition, CVPR 2016
[7] M. Rastegari et al., “XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks,” ECCV 2016

필 자 소 개

최 종 현

– 2003년 : 서울대학교 전기공학 학사
– 2008년 : 서울대학교 전기컴퓨터공학 석사
– 2013년 : 어도비 연구소 연구인턴
– 2014년 : 디즈니 연구소 연구인턴
– 2014년 : 마이크로소프트 연구소 연구인턴
– 2015년 : 메릴랜드 주립대학교 전기컴퓨터공학 박사
– 2015년 : 컴캐스트 랩스(워싱턴, DC) 선임연구원
– 2016년 ~ 2018년 : 앨런 인공지능 연구소(시애틀, WA) 연구원
– 2018년 ~ 현재 : 광주과학기술원(GIST) 인공지능대학원/전기전자컴퓨터공학부 조교수