Master's Thesis

THz Photonic Device Design by

Machine Learning

Jonggeon Lee **(이 종 건)**

Department of Electrical Engineering

Pohang University of Science and Technology

2020

머신 러닝을 활용한

테라헤르츠 광학소자 설계

THz Photonic Device Design by

Machine Learning

MEE 이종건 Jonggeon Lee,

20182737 THz Photonic Device Design by Machine Learning.

머신 러닝을 활용한 테라헤르츠 광학소자 설계,

Department of Electrical Engineering, 2020, 45P,

Advisor: Haewook Han

ABSTRACT

The data traffic has exponentially increased over the last few years. Therefore, communication system need Terahertz (THz)-band for beyond 5G. Because THz has too large attenuation in the atmosphere to long-distance wireless communication, THz optical fiber communication should be studied. We designed 1 THz Distributed Bragg Reflector (DBR) and 1×2 Optical Power Splitter (OPS) for THz optical fiber communication by machine learning. First, we used Additive Reinforcement Learning (ARL) algorithm which is conceptually simple, fully parallelizable and not depend on initial solutions. Second, we used Artificial Neural Network (ANN) algorithm and inverse design method. ANN make that inverse design can calculate gradient analytically. Using these two methods, we can make the DBR which is shorter than analytically designed DBR. Also, we can make low insertion loss and broadband OPS with ARL and increase accuracy of OPS prediction ANN by training data which has uniform distribution. Our results and approaches are expected to make a significant contribution to the developments of THz photonic devices.

목차

Ⅰ. 서론

1.1 테라헤르츠 전자파………………………………………………

1.2 머신 러닝

1.3 머신 러닝 도구 및 전산 자원

Ⅱ. 분산 브레그 반사경

2.1 다층 구조에서의 전달 행렬

2.2 분산 브레그 반사경 (Distributed Bragg Reflector)의 구조와 특성

2.3 DBR 전산 모사 도구

Ⅲ. 1×2 광전력 분배기

3.1 1×2 광전력 분배기 (Optical Power Splitter)

3.2 1×2 OPS 전산 모사 도구

Ⅳ. 가법 강화 학습

4.1 퍼셉트론 (Perceptron)

4.2 강화 학습 (Reinforcement Learning)

4.3 가법 강화 학습 (Additive Reinforcement Learning)을 이용한 설계

4.3.1 가법 강화 학습

4.3.2 ARL을 이용한 DBR 설계 방법

4.3.3. ARL을 이용한 DBR 설계 결과

4.3.4 ARL을 이용한 1×2 OPS 설계 방법

4.3.5 ARL을 이용한 1×2 OPS 설계 결과

Ⅴ. 인공 신경망

5.1 인공 신경망 (Artificial Neural Network)

5.1.1 인공 신경망의 구조와 학습

5.1.2 오버피팅 (Overfitting)

5.1.3 합성곱 신경망 (Convolutional Neural Network)

5.1.4 ResNet

5.2 Inverse Design

5.3 인공 신경망을 이용한 Inverse Design

5.3.1 인공 신경망을 이용한 Inverse Design 방법

5.3.2 인공 신경망을 이용한 DBR Inverse Design

5.3.3 1×2 OPS Prediction 인공 신경망

Ⅵ. 결론

References

그림 목차

**그림 1.1** 전자기파의 주파수 대역 스펙트럼

**그림 1.2** 광학 소자 설계 방법에 따른 실현 가능성과 스펙트럽 복잡성의 관계

**그림 1.3** FDTD 전산 모사를 위한 HPC와 사양

**그림 2.1** 다층 구조에서의 빛의 전파

**그림 2.2** N+1/2 pairs의 주기적인 DBR 구조

**그림 3.1** Y-branch 형태의 OPS

**그림 3.2** 머신 러닝 설계를 위해 변경된 OPS 구조

**그림 4.1** 퍼셉트론

**그림 4.2** 행렬 크기 (n) 변화에 따른 ARL 알고리즘 정확도

**그림 4.3** ARL을 이용한 DBR 설계 플랫폼

**그림 4.4** ARL을 이용한 DBR 설계 보상 정의 방법

**그림 4.5** ARL을 이용한 DBR 설계 학습 데이터 분포도

**그림 4.6** ARL으로 설계한 DBR과 이론으로 설계한 DBR의 반사율 비교

**그림 4.7** ARL으로 설계한 DBR과 이론으로 설계한 DBR의 두께 비교

**그림 4.8** ARL을 이용한 1x2 OPS 설계 플랫폼

**그림 4.9** ARL을 이용한 1x2 OPS 설계 학습 데이터 분포도

**그림 4.10** ARL으로 설계 한 1x2 OPS (a) 식각 구조 (b) 파장이 299 um 일때 power flow (c) 입력 포트 반사율 (R)과 출력 포트 투과율 (T)

**그림 5.1** 인공 신경망의 구조

**그림 5.2** Gradient descent method 개념도

**그림 5.3** 부적합한 학습율 사용으로 인한 문제점

**그림 5.4** Optimizer 성능 비교

**그림 5.5** 활성화 함수 (a) sigmoid (b) tanh (c) Relu

**그림 5.6** Regression 결과 (a) 학습이 잘 된 경우 (b) 언더피팅 (c) 오버피팅

**그림 5.7** CNN 개념도

**그림 5.8** Convolution layer 계산 과정

**그림 5.9** 입력 데이터에 따른 pooling 결과 비교

**그림 5.10** 인공 신경망 layer의 연결 방법 (a) 일반 layer 연결 (b) residual block

**그림 5.11** DBR prediction 인공 신경망 학습 결과

**그림 5.12** ANN으로 설계한 DBR과 이론으로 설계한 DBR의 반사율 비교

**그림 5.13** ANN으로 설계한 DBR과 이론으로 설계한 DBR의 두께 비교

**그림 5.14** CNN을 이용한 OPS prediction 인공 신경망 학습 테스트

**그림 5.15** OPS prediction 인공 신경망 학습 데이터 분포도

**그림 5.16** OPS prediction 인공 신경망 학습결과 (a) FCDNN (b) ResNet

**그림 5.17** OPS prediction 인공 신경망 학습 테스트 (a) FCDNN (b) ResNet

표 목차

**표4.1** ARL으로 설계한 DBR과 이론 DBR의 반사율 비교

**표 5.1** ANN으로 설계한 DBR과 이론 DBR의 반사율 비교

Ⅰ. 서론

1.1 테라헤르츠 전자파

테라헤르츠 (Terahertz, THz)파는 마이크로파와 적외선 사이에 존재하는 0.1 ~ 10 THz 전자파를 의미한다. 파장으로는 30 ~ 3000 μm, 광자 에너지로는 0.41 ~ 41 meV의 물리량을 가진다. 테라헤르츠파는 전자공학과 광공학의 경계에 위치하기 때문에 융합형 과학기술로의 발전 가능성이 크다.

최근 몇 년간 무선 통신의 데이터 전송량이 기하급수적으로 증가하고 있다 [1]. 이는 “필요한 데이터 전송량은 18개월마다 2배로 증가한다”는 Edholm’s law에 의해서도 예측된 상황이기도 하다 [2]. 이와 같이 데이터 전송량이 늘어나면서 ‘beyond 5G’ 시대에서는 더 넓은 대역의 주파수가 필요해졌다. 이에 테라헤르츠 대역이 실현 가능성과 적합성이 가장 높은 주파수 대역으로 평가되고 있다 [3,4]. 하지만 테라헤르츠는 무선 환경에서 대기중에 흡수되어 손실되는 신호가 많기 때문에, 아주 먼 거리에서의 무선 통신은 어렵다 [5]. 따라서 테라헤르츠 광통신 (fiber optic communication) 연구가 우선적으로 진행되어야 한다.



그림 1. 1 전자기파의 주파수 대역 스펙트럼

1.2 머신 러닝

테라헤르츠 광통신을 위해서는 테라헤르츠 브레그 분산 반사경 (Distributed Bragg Reflector, DBR)과 광전력 분배기 (Optical Power Splitter)가 필수적이다. DBR 같은 경우 이론적인 모델이 정립되어 있어, 높은 효율의 소자를 만드는데 어려움이 없다. 그러나 광전력 분배기의 경우에는 이론적으로 해석하는데 어려움이 있어, 높은 효율을 가지도록 설계하는데 한계가 있다. 기존에는 직접 이진 검색 (Direct Binary Search, DBS) [6,7] 방법이나, 기본 구조에서 길이나 광학 상수들을 변수로 하여 각각의 변수에 해당하는 모든 소자의 성능을 계산하고 비교하여 최적의 구조를 찾는 방법 [8,9], 일정 주기로 반복되는 패턴을 이용한 자기 이미징 (Self-Imaging) 방법 [10,11] 등으로 설계를 하였다. 그러나 이 방법들은 요구하는 계산 장비 성능 높고, 계산 시간도 오래 걸린다. 또한 이러한 방식으로 설계된 구조들은 실현 가능성 (Design feasibly)과 스펙트럼 복잡성 (Spectral complexity)이 반비례하기 때문에 원하는 스펙트럼을 가지면서도 실현 가능한 구조를 찾기 어렵다 [12]. 머신 러닝 (Machine Learning)으로 광학 소자를 설계한다면 이러한 한계들을 극복할 수 있다.



그림 1. 2 광학 소자 설계 방법에 따른 실현 가능성과 스펙트럼 복잡성의 관계

머신 러닝은 인공 지능 (Artificial Intelligence)의 한 분야로 어떠한 문제 (tasks)와 결과 (performance measure)로부터 데이터 (experience)를 얻어 학습하는 프로그램을 말한다 [13]. 실제 데이터들을 학습 데이터 (training data)라고 한다. 딥 러닝은 머신 러닝의 한 분야로 학습이라는 기본 방식은 같지만, 기본 개념 (simple concepts)으로부터 추가적인 개념 (nested hierarchy of concepts)을 스스로 생성해내어 더 쉽게 문제를 해결한다. 예를 들어 사진을 보고 개와 고양이를 구분하는 문제를 머신 러닝과 딥 러닝으로 해결한다고 생각한다면, 머신 러닝은 문제를 해결하기 위해서 개와 고양이의 특징들 (features)을 프로그램에 입력해주어야 한다. 이러한 경우, 개와 고양이 각각의 종마다 생김새가 다르기 때문에 많은 특징들을 입력해야 한다는 어려움이 있다. 하지만 딥 러닝은 학습 데이터들로부터 스스로 특징들을 생성하기 때문에 좀 더 편리하게 문제를 해결 할 수 있다.

머신 러닝 알고리즘은 크게 세 분류로 나눌 수 있다: 지도 학습 (Supervised Learning), 비지도 학습 (Unsupervised Learning), 강화 학습 (Reinforcement Learning). 지도 학습은 데이터가 어떠한 것인지 명시가 돼있는 상태 (labeled)에서 학습하는 방법이다. 데이터들의 경향성을 학습하여 다음 데이터를 예측하는 regression, 분류된 데이터를 학습하여 새로운 데이터를 분류하는 classification이 지도 학습으로 해결하는 대표적인 문제들이다. 비지도 학습은 데이터가 어떠한 것인지 명시 되지 않은 상태 (unlabeled)에서 학습하는 방법이다. 비슷한 데이터들끼리 묶어주는 clustering, 데이터의 차원을 낮추어 분석이 용이해지도록하는 dimensionality reduction이 비지도 학습으로 해결하는 대표적인 문제들이다. 데이터를 명시하기 어려운 빅데이터 문제에서 사용된다. 강화 학습은 문제 상태에 따라 프로그램이 어떠한 행동을 하도록 하는데, 그에 해당하는 보상(reward)를 주면서 보상을 최대화하는 행동을 하도록 학습하는 방법이다. 강화 학습은 보상을 최대로 하는 개념을 이용하여 여러 최적화 문제에서 사용된다.

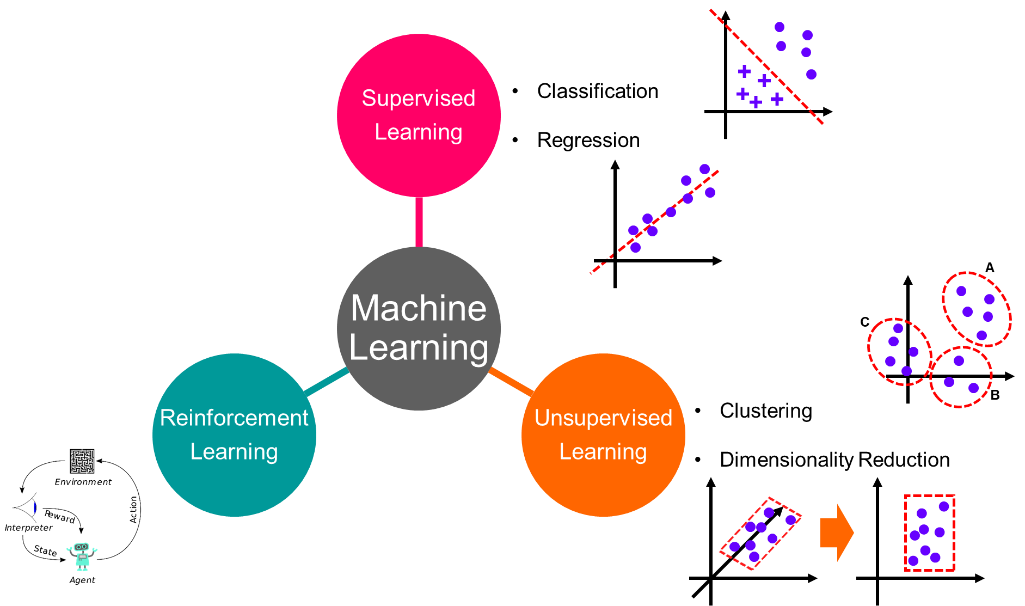


그림 1. 3 머신 러닝 분류

본 논문에서는 테라헤르츠 광통신 연구를 위한 기초연구로 머신 러닝을 이용하여1 테라헤르츠 (1 THz)에서 동작하는 DBR과 1×2 광전력 분배기를 설계하였다. 먼저, DBR과 1×2 광전력 분배기에 대해 알아보고, 머신 러닝으로 문제를 해결하기 적합하도록 기본 광학 소자 구조를 설계하였다. 그 후 머신 러닝에 속하는 가산 강화 학습 방법을 이용하여 DBR과 1×2 광전력 분배기를 설계하고, 딥 러닝에 속하는 인공 신경망 방법을 이용하여 DBR설계와 1×2 광전력 분배기의 투과율 예측 인공 신경망의 정확도를 증가시켰다.

1.3 머신 러닝 도구 및 전산 자원

본 연구에서 Anaconda[14] 가상 환경에서 프로그래밍 언어 파이썬 (Python)으로 머신 러닝을 사용하였다. 인공 신경망을 위해서 구글의 오픈 소스 (open source)인 Tensorflow[15]를 사용하였다. 전산 자원은 다음과 같다.



그림 1. 4 FDTD 전산 모사를 위한 HPC와 사양

Ⅱ. 분산 브레그 반사경

2.1 다층 구조에서의 전달 행렬(Transfer Matrix)]

DBR과 같은 다층 구조에서 빛의 전파 특성은 전달 행렬을 사용하여 쉽게 계산할 수 있다 [16]. 그림 2.1과 같이 매질이 x 방향에 대해 균일하지 않다면, 유전율(permittivity, ε)과 투자율(permeability, μ)을 구간에 대해 나누어 생각할 수 있다.

for



그림 2. 1 다층 구조에서의 빛의 전파

입사하는 빛과 반사하는 빛이 모두 TE mode (Transverse Electric mode)라고 생각하면 각각의 전기장(**E**i, **E**r)은 다음과 같이 나타낼 수 있다.

*l* 번째 매질 ()에서의 전기장은 이며, 자세히 나타내면 다음과 같다.

이때 , 이다. 전기장의 탄젠트(tangent) 성분 *Ey*와 자기장의 탄젠트 성분 *Hz*가 매질의 경계인 에서 연속이라는 경계 조건을 사용하면 다음 식을 얻을 수 있다.

위 두 식을 연립하여 정리하면 두 매질에서 빛의 전달 관계식을 행렬 형태로 나타낼 수 있게 된다. 이를 좀더 간단히 나타내기 위해서 다음과 같이 *Pl(l+1)*와 *l*+1번째 매질의 두께를 정의한다.

또한, *l*+1번째 매질의 두께를 다음과 같이 나타낼 수 있다.

위의 두 식을 정리하면 다음과 같이 전달 행렬을 나타낼 수 있다.

여기서, 은 *l*+1번째 매질에서 *l*번째 매질로 진행하는 빛을 계산하기 때문에 역방향 전파 행렬(backward-propagation matrix)라고 한다. 정방향 전파 행렬(forward-propagation matrix) 도 다음과 같이 구할 수 있다.

앞에서 구한 전달 행렬을 N+1 개의 층을 가진 다층 구조에 적용하면 전체 구조에 대한 빛의 전파 특성을 다음과 같이 구할 수 있다.

이로부터 다층 구조에 대한 투과 계수(transmission coefficient) t와 반사 계수(reflection coefficient) r를 다음과 같이 구할 수 있다.

,

2.2 분산 브레그 반사경 (Distributed Bragg Reflector)의 구조와 특성

Distributed Bragg Reflector (DBR)은 서로 다른 굴절률(index, n)을 가지는 두 물질을 번갈아 배치한 다층 구조의 반사 장치이다. 일반적으로 DBR 양쪽 끝에 같은 물질을 사용하기 때문에 N+1/2 pairs 구조를 가진다. DBR 각 매질의 광학적 두께 (optical thickness)는 반사를 위한 공진 조건을 만족하려면 설계 파장 (design wavelength)의 1/4이 되어야 한다. 전달 행렬을 N+1/2 pairs DBR 구조에 적용하면 다음과 같다.

여기서 아래 첨자 0은 빛이 입사하는 매질, s는 빛이 전체 구조를 투과한 이후의 매질을 나타내며, 보통 입사 매질과 투과 매질은 같다. *h*와 *l*은 각각 상대적으로 높고 낮은 굴절률을 가지는 매질을 의미한다.



그림 2. 2 N+1/2 pairs의 주기적인 DBR 구조

2.3 DBR 전산 모사 도구

머신 러닝을 이용하여 DBR을 설계하기 위해서는 무작위 다층 구조로 이루어진 학습 데이터가 필요하다. 이는 2.1에서 서술한 전달 행렬을 계산하여 만들 수 있다. 전산 모사 소프트웨어로는 Python에서 오픈 소스인 TMM(Transfer Matrix Method) package [17]를 사용하였다.

Ⅲ. 1×2 광전력 분배기

3.1 1×2 광전력 분배기 (Optical Power Splitter)

1×2 Optical Power Splitter (OPS)는 하나의 도파관 (waveguide)에 두 개의 도파관을 Y 자로 나뭇가지처럼 (Y-branch) 연결한 형태로, 하나의 입력된 광을 두 개의 똑같은 전력을 가진 광으로 분배하는 소자이다. 광통신 시스템에서 효율적인 광전송을 위해서는 광섬유 사이간의 효율적인 결합이 필요하기 때문에 OPS에 대한 연구가 많이 진행되고 있다 [18, 19].



그림 3. 1 Y-branch 형태의 OPS

기본 OPS 형태인 Y-branch는 branch angle이 작을수록 branch 입력에서 발생하는 분산 손실 (scattering loss)과 출력에서 발생하는 전환 손실 (conversion loss)을 줄일 수 있다 [20]. 따라서 branch distance가 도파관 두께에 비해 길게 설계하게 되는데, 이 때문에 소자의 크기가 불필요하게 증가하게 된다. 머신 러닝을 이용하여 OPS를 설계할 경우에는 그림 3.2와 같은 구조를 이용하여 길이가 길어지는 문제를 해결할 수 있다.



그림 3. 2 머신 러닝 설계를 위해 변경된 OPS 구조

그림 3.2 (a)-1과 같이 기존 Y-branch 구조에서 설계 영역을 20 × 20 pixels로 나누어 준다. 이때 설계 목표에 맞도록 pixels의 매질을 결정하게 된다. 그림 3.2 (b)-1처럼 정사각형의 pixel을 그대로 사용하게 되면, 실제 공정에서 직각을 나타내기가 어려워 공정 효율이 떨어지게 된다. 따라서 그림 3.2 (b)-2와 같이 정사각형 pixel을 원형 pixel로 바꾸어, 같은 효과를 보이면서도 공정 효율을 획기적으로 증가시킬 수 있다 [6].

1×2 OPS의 목표는 하나의 입력된 광을 두 개의 똑같은 전력을 가진 광으로 분배하는 것이기 때문에 대칭 성질을 사용하여 설계한다. 또한, 대칭 성질을 사용하면 전산 모사 소요시간이 감소하며, 설계 난이도도 감소하게 된다. 1×2 OPS의 성능 검사 지표는 두 출력의 전력 유사도가 아니라, 내부 손실 (insertion loss)과 전력 균일도 (power uniformity)가 된다. 이 때 내부 손실은 입력 대비이기 때문에, 1×2 OPS의 이상적인 내부 손실값은 3dB이다.

3.2 1×2 OPS 전산 모사 도구

OPS의 branch를 계단 형태로 근사하여 이론적으로 해석할 수도 있지만, 본 논문에서 설계하는 OPS는 QR code-like 구조이기 때문에 전산 모사를 통해 해석한다. 전산 모사는 Finite-Difference Time-Domain (FDTD) 방법을 사용하였다. FDTD 전산 모사는 대상을 아주 작은 사각형 cell로 나누어 Maxwell 방정식으로 근사없이 정확하게 계산하는 도구이다. 따라서 복잡한 구조에서도 높은 정확도를 보인다. 전산 모사 소프트웨어는 Lumerical 사의 상용 소프트웨어인 FDTD Solutions [21]를 사용하였다.

Ⅳ. 가법 강화 학습

4.1 퍼셉트론(Perceptron)

퍼셉트론은 1957년 Frank Rosenblatt에 의해 제안된 초기 형태의 인공 신경망이다 [22]. 다수의 입력으로부터 하나의 결과를 내보내는 알고리즘으로 실제 뇌를 구성하는 뉴런 (neuron)과 유사한 동작을 한다.



그림 4. 1 퍼셉트론

그림 4.1에서 x는 입력값, w는 가중치(weight), y는 출력값을 의미한다. x에 입력이 보내지면 w와 곱해진다 (x∙w). 곱해진 값이 바로 y가 되는 것이 아니라, 활성화 함수 (activation function)라는 것을 거치게 된다. 뉴런은 신호가 들어왔을 때 반응을 일으키기 위해 필요한 최소한의 작은 크기인 역치 (threshold)가 존재한다. 퍼셉트론에서도 이와 같은 개념으로 활성화 함수가 존재한다. 그래서 퍼셉트론에서는 주로 일정 값 이상이면 1을 출력하고, 미만이면 0을 출력하는 계단 함수 (Step function)를 활성화 함수로 사용한다. 결과적으로, 퍼셉트론은 데이터들의 선형 분류기로 동작을 한다.

4.2 강화 학습(Reinforcement Learning)

강화 학습은 머신 러닝의 한 분야로 어떠한 문제를 해결할 때 컴퓨터가 스스로 최적의 결정을 내릴 수 있도록 학습시키는 방법이다 [23]. 강화 학습 방법으로 문제를 해결하기 위해서는, 그 문제를 Markov Decision Process (MDP)에 맞추어 정의해야한다. MDP는 다음과 같이 정의된다.

**Definition 4.1** MDP는 5개의 요소(*S*, *A*, *T*, *R*, γ)로 이루어진 집합이다. 각각의 요소는 다음과 같다:

* *S*: 상태 (state)들의 집합
* *A*: 행동 (action)들의 집합
* *T*: 전이 함수 (transition function), 특정 상태에서 특정 행동을 했을 때 다음 상태는 어떤 상태가 될지에 관한 확률
* *R*: 보상 함수 (reward function), 주어진 상태에서 행동을 했을 때 얻게 되는 보상 (reward)
* γ: 감쇠 상수 (discount factor), 처음 받은 보상의 영향을 증가시키고, 시간이 지난 보상의 영향을 줄이는 역할

초기 상태 s0에서 초기 행동 a0를 수행하면 주어진 Ts0a0에 따라 다음 상태 s1이 확률적으로 결정된다. 그 결과로 보상 R(s0, a0)를 얻게 된다. 이 과정을 s1에도 적용시키며 목표 상태에 도달할 때까지 반복한다. 강화 학습의 목표는 시간이 흘렀을 때 얻게 되는 모든 보상들의 합을 최대화 시키는 정책 (policy, π)을 학습하는 것이다. 정책은 특정 상태에서 어떠한 행동을 해야 하는지 결정하는 함수이다. 이때 최대화 하고자 하는 보상들의 합은 감쇠 상수와 함께 다음과 같이 정의 된다.

T에 의해 상태에 따른 행동이 정해지기 때문에 실제 최대화 시키는 값은 보상들의 합의 기대값이다.

4.3 가법 강화 학습(Additive Reinforcement Learning)을 이용한 설계

4.3.1 가법 강화 학습

Additive Reinforcement Learning (ARL)은 퍼셉트론의 가법 성질 (additive)과 강화 학습 (Reinforcement)의 보상 시스템을 합한 알고리즘이다 [24]. 따라서 다른 머신 러닝 알고리즘들 보다 개념적으로 단순하여 여러 문제에서 사용될 수 있다. 또한 초기 조건에서 점점 진화하는 (evolution) 형태가 아닌, 독립적인 데이터들을 사용하기 때문에 모든 과정을 병렬화하여 시간을 단축 시킬 수 있고, 초기 조건에 따라 결과값이 달라지지 않는다.

ARL은 학습 단계 (Training phase)와 추론 단계 (Inference phase) 로 나뉘어 진다 [25]. 학습 단계에서는 데이터로 주어진 구조들의 보상을 계산하여 각 구조들의 설계 목표에 부합하는 정도를 계산한다. *A* 를 우리가 원하는 구조라 가정했을 때, 임의의 구조 *B*에 대해 *A*와 비슷한 정도를 보상으로 나타낼 수 있으며, 그 함수를 *f*(*A*, *B*)라 하자. 보상 *r*을 이진 행렬로 표현한 *B*에 곱하여 보상 총합 행렬 (reward summation matrix, Ɛ)에 누적 시킨다. 보상 총합 행렬은 0으로 초기화 되어있는 상태이다.

이때 구조의 길이를 n, 구조의 개수 (학습 데이터의 개수)를 N이라 하면 다음과 같이 의사 코드 (pseudo code)로 나타낼 수 있다.

**Algorithm 4.1** ARL: Training Phase

1: **for all** *i* ∈ {1, …, *n* \* N} **do**

2: Randomly create *n* × *n* binary guess matrix *B*

3: Get *r* = *f*(*A*, *B*)

4: Ɛ ← Ɛ + *r* \* *B*

5: **end for**

실제 광학 소자를 설계 할 때는 원하는 구조 *A*를 모르는 상태이기 때문에 설계자가 적절한 보상 함수 *f*(*B*)를 정의하여야 한다. 예를 들어, DBR을 설계하고자 할 때는 구조 *B*의 설계 목표 주파수에 해당하는 반사율 (reflectance)을 보상 함수로 정의하여 반사율이 최대가 되는 구조를 찾는다.

추론 단계에서는 학습 단계에서 계산한 reward summation matrix, Ɛ로부터 최종 결과물을 추론한다. Ɛ는 구조들의 보상들을 모두 누적시킨 상태이기 때문에 원소들의 값이 매우 크기 때문에, 원소들의 평균을 구하는 연산과정에서 자료형의 범위를 넘어갈 수 있다. 따라서 Ɛ의 최소값을 구하여 모든 원소에서 빼주는 정규화 과정을 거친다..

그리고 Ɛ 모든 원소의 평균을 *avg*라 하고, 모든 원소가 0으로 초기화된 상태의 최종 결과를 저장할 행렬을 다음과 같이 *F*라 한다.

그 다음 Ɛ*i*, *j*가 *avg*보다 크거나 같으면 *Fi*, *j*에 1을 작으면 0을 대입해준다. 다음과 같이 의사 코드로 나타낼 수 있다.

**Algorithm 4.2** ARL: Inference Phase

1: **for all** *i* ∈ {1, …, *n* } **do**

2: **for all** *j* ∈ {1, …, *n* } **do**

3: **if** Ɛ*i, j* ≥ *avg* **then**

4: *Fi, j* = 1

5: **else**

6: *Fi, j* = 0

7: **end if**

8: **end for**

9: **end for**

의사 코드에서도 알 수 있듯이 다른 머신 러닝 알고리즘들 보다 개념적으로 단순하여 여러 문제에서 사용될 수 있다. 하지만, 학습 데이터가 결과를 얻기에 충분하려면 다른 알고리즘들 보다 많은 양이 필요하다. 알고리즘 제안 논문 [24]에서 *n*×*n* 이진 행렬을 찾는 문제로 테스트하였을 때, 10,000개 데이터로 *n*이 10이하일때는 90% 정확도로 답을 찾는 것이 가능했다. 하지만 11부터는 정확도가 90% 미만으로 떨어지며 13일 때는 정확도가 50%로 이하로 떨어졌다.



그림 4. 2 행렬 크기 (*n*) 변화에 따른 ARL 알고리즘 정확도

하지만 데이터의 개수가 100,000개 되면 정확도가 모두 100%이 된다. 따라서 *n*이 11, 구조의 변수가 100 이상이 되면 10,000에서 100,000개 사이의 학습 데이터가 필요하다.

4.3.2 ARL을 이용한 DBR 설계 방법

DBR의 설계 목표 주파수는 1 THz (= 300 μm) 이며, 실제 제작을 고려하여 공기와 fused quartz를 사용하였다. 설계에서 고려한 파장 영역은 5 μm 단위로 나누어진 150 ~ 3000 μm 이다. 물질들의 굴절률은 테라헤르츠 대역에서 변화가 거의 없기 때문에 설계 목표 주파수인 1 THz에서의 굴절률로 고정시켜, 공기의 굴절률은 1 (*nl* = 1), fused quartz (SiO2)의 굴절률은 2.092 [26]이다 (*nh* = 2.092).



그림 4. 3 ARL을 이용한 DBR 설계 플랫폼

DBR 구조를 이진 행렬 형태로 표현하기 위하여 일정 두께 ∆*x*를 가지는 단위 층 (unit layer)들이 쌓여있는 구조를 생각한다. 이 때 이진 행렬의 값이 1이면 quartz의 굴절률을, 0이면 공기의 굴절률을 대입한다. 이진 행렬의 길이가 *N*이면 실제 DBR 전체 길이는 *N*∙∆*x*가 된다. 공진 조건을 고려하여 DBR 각층의 두께를 구하면 *dh* = 35.85 μm, *dl* = 75 μm 이다. 한 pair의 두께가 약 100 μm 이고, 3.5 ~ 4.5 pairs 정도의 두께를 고려하여 L = 400 μm로 설정하였다. ∆*x*가 너무 작으면 N이 증가하기 때문에 ARL에 필요한 데이터의 개수가 증가하고, 너무 크게 되면 정확한 층 두께를 구할 수 없게 된다. 따라서 공진 조건을 고려하여 구한 두께를 참고하여 ∆*x* = 5 μm, *N* = 80으로 설정하였다. 설계에서 고려한 빛의 진행 mode는 TE mode이다. 설계 주파수에서 대역폭을 정의하였을 때, 그림 4.4에서 보이듯이 대역폭에 해당하는 반사율을 *R*in, 해당하지 않는 반사율을 *R*out이라 하여 다음과 같이 보상 *r*을 정의하였다.

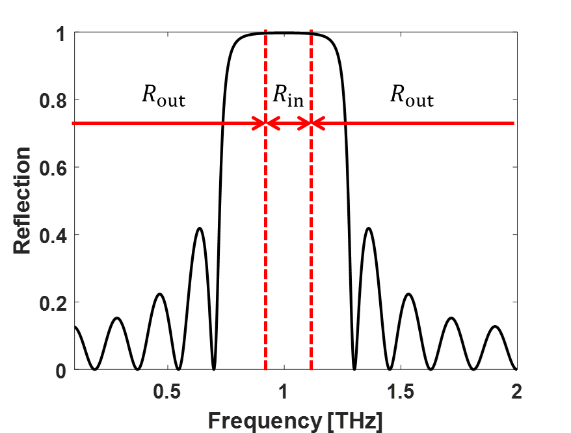


그림 4. 4 ARL을 이용한 DBR 설계 보상 정의 방법

학습 데이터로 길이 *N*의 이진 행렬을 무작위로 생성하여 총 20,000개의 구조와 반사율을 구하였다. 데이터 분포를 보상에 대해 정리하면 그림 4.5로 나타낼 수 있다.



그림 4. 5 ARL을 이용한 DBR 설계 학습 데이터 분포도

4.3.3 ARL을 이용한 DBR 설계 결과

그림 4.6은 ARL을 이용하여 설계한 DBR과 공진 구조를 고려하여 설계한 DBR의 반사율을 비교한 그래프이다. ARL을 이용한 설계에서 전산 모사를 파장 단위로 계산하여 설계 파장을 300 μm, 대역폭을 100 μm로 하여, 주파수 단위로 변환한 그래프에서는 양끝이 1.2 THz, 0.85 THz가 되어 대역폭이 치우쳐져 있는 형태로 나타난다.

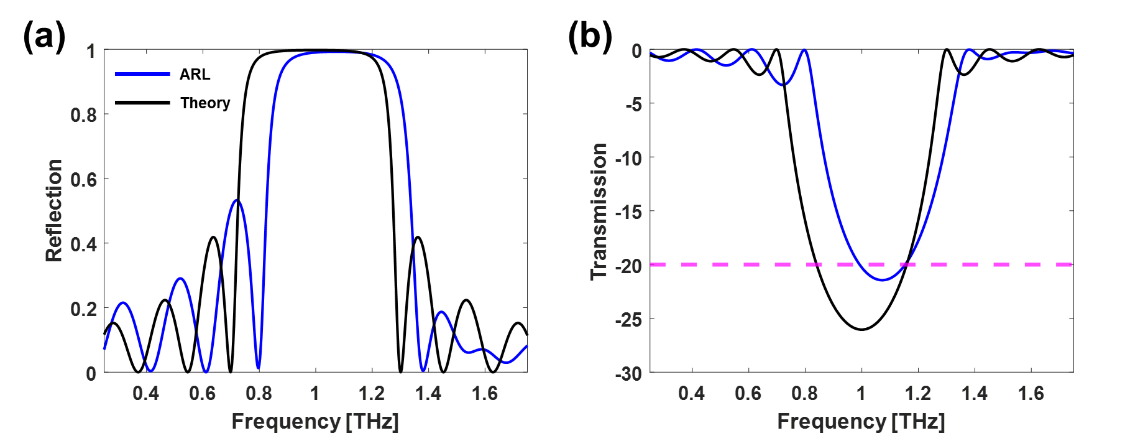


그림 4. 6 ARL으로 설계한 DBR과 이론 DBR의 (a) 반사율 (b) 투과율 비교, 점선은 transmission이 99%인 지점은 의미한다.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Maximum Reflectance | FWHM | 99% bandwidth |
| ARL | 0.9928 at  1.07 THz (280 μm) | 0.541 THz  (145 μm) | 0.287 THz  (75 μm) |
| Theory | 0.9978 at  0.99 THz (300 μm) | 0.553 THz  (180 μm) | 0.354 THz  (110 μm) |

표4. 1 ARL으로 설계한 DBR과 이론 DBR의 반사율 비교



그림 4. 7 ARL으로 설계한 DBR과 이론 DBR의 두께 비교

ARL을 이용하여 설계한 DBR이 이론으로 설계한 DBR보다 수치적으로 부족한 성능을 보인다. 하지만 설계 주파수 1 THz에서의 반사율은 모두 99%를 넘으며, 같은 pair이면서도 전체 길이가 공진 조건을 고려하여 설계한 DBR보다 약 100 μm 짧기 때문에 공간 제약이 있는 환경에서 유용하게 사용될 수 있다.

4.3.4 ARL을 이용한 1×2 OPS 설계 방법

1×2 OPS의 설계 목표는 일정 대역폭의 투과율을 최대화시키는 것이 아니라, 원하는 광대역의 투과율을 최대화 시키는 것이다. 목표 주파수가 1 THz이기 때문에 설계 대역폭을 275~325 μm 로 한다. 이를 2 μm 단위로 나누어 총 26개 지점에서 투과율을 측정한다. OPS 구조에서는 식각을 한다는 의미로 ‘1’이 굴절률이 낮은 공기, 식각을 하지 않는다는 의미로 ‘0’이 굴절률이 높은 실리콘 (Silicon, Si)을 의미한다. 이를 이진 행렬로 표현하면 2차원으로 나타내어진다. 물질들의 굴절률은 앞에서 말한 바와 같이, 테라헤르츠 대역에서 변화가 거의 없기 때문에 설계 목표 주파수인 1 THz에서의 굴절률로 고정시켰다. 따라서 공기의 굴절률은 1 (*nl* = 1)이며, 실리콘의 굴절률은 3.415 [27]이다 (*nh* = 3.415).



그림 4. 8 ARL을 이용한 1×2 OPS 설계 플랫폼

그림 4.7 은 ARL을 이용한 1×2 OPS 설계 플랫폼을 나타낸 것이다. 500 μm×500 μm 정사각 실리콘 위에 9 μm 반지름의 원형 구멍을 식각하여 만든다. 입력과 출력 포트 (port)들은 테이퍼 (taper)가 없으며, 너비가 100 μm인 무한한 길이의 직사각형으로 가정한다. 입력 포트는 정사각 실리콘 한 변의 중앙에 위치하며, 출력 포트는 입력 포트 반대편 변에 중앙으로부터 각각 125 μm 거리에 위치해 있다. OPS는 대칭 구조이기 때문에 T1과 T2의 수치가 같다. 따라서 T로 통일시켜 사용한다. 입력으로는 FDTD에서 기본 모드 (fundamental mode)로 계산된 TM mode (Transverse Magnetic mode)를 사용하였다.

설계 목표가 광대역의 투과율을 최대화시키는 것이기 때문에 보상을 다음과 같이 정의한다. 이때, min(T)는 한 구조의 최소 투과율이며, Tmin은 최소 투과율 전체 중 최소, Tmax는 최소 투과율 전체 중 최대를 의미한다.

학습 데이터로 20×10 이진 행렬을 무작위로 생성하여 총 40,000개의 구조와 투과율을 구하였다. 데이터 분포를 보상에 대하여 나타내면 그림과 같다.

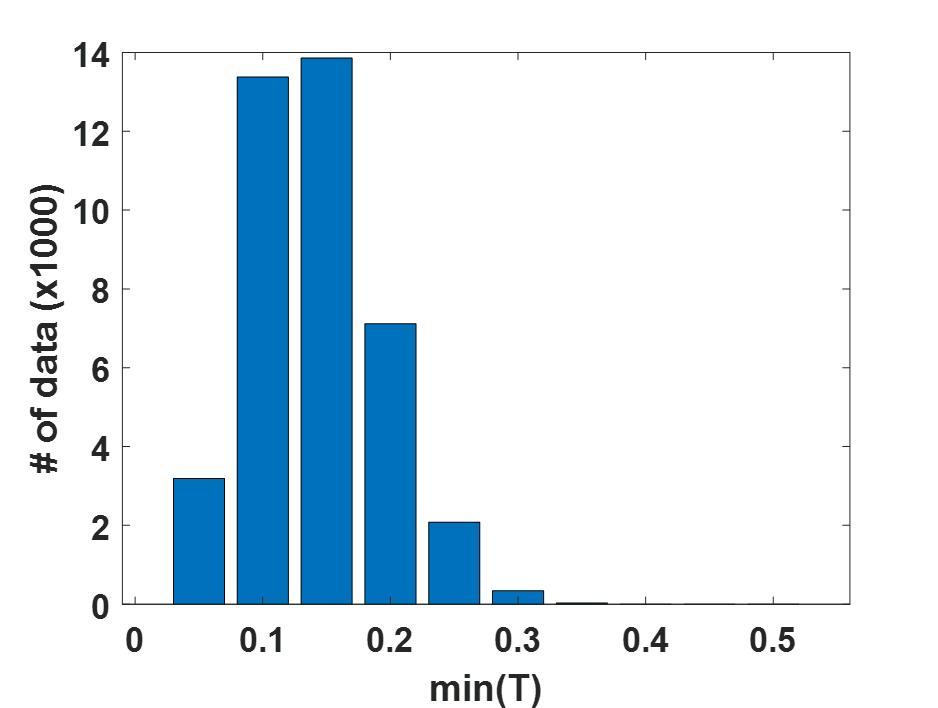


그림 4. 9 ARL을 이용한 1×2 OPS 설계 학습 데이터 분포도

보상을 투과율의 최솟값이 아닌 평균값으로 설정하게 되면, 투과율이 일정하지 않고 peak이 발생할 수 있기 때문에 최솟값을 사용하였으며, 전체의 최솟값과 최댓값으로 정규화를 하였다. 또한 입력 포트 반사율을 최소화 시키는 인자를 사용하지 않더라도 투과율을 최대화 시킴으로써, ARL이 고려해야하는 특징의 개수를 줄여 문제의 복잡도를 감소시킬 수 있다.

4.3.5 ARL을 이용한 1×2 OPS 설계 결과

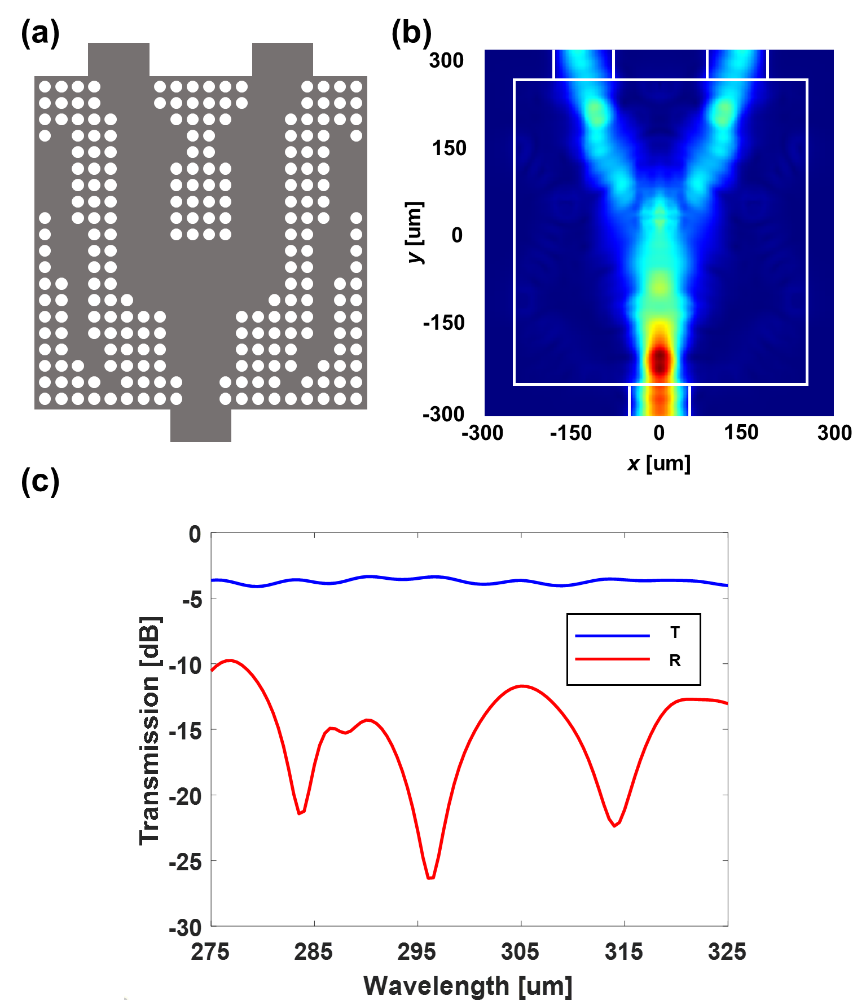


그림 4. 10 ARL으로 설계한 1×2 OPS (a) 식각 구조 (b) 파장이 299 μm 일 때 power flow (c) 입력 포트 반사율 (R)과 출력 포트 투과율 (T)

ARL을 이용하여 설계한 OPS의 내부 손실은 평균 3.71 dB로 4 dB보다 적고, 균일도는 0.75 dB이다. 다른 파장대역에서 설계된 기존 OPS들보다 branch distance가 파장 대비 짧다. 따라서 공간 제약이 있는 환경에서 유용하게 사용될 수 있다.

Ⅴ. 인공 신경망

5.1 인공 신경망(Artificial Neural Network)

5.1.1 인공 신경망의 구조와 학습

인공 신경망은 여러방면으로 사용할 수 있다. 본 논문에서는 인공 신경망의 지도 학습 방법 중 하나인 regression을 사용할 것이다. Regression은 학습 데이터로 학습이 완료되면 새로운 입력이 들어왔을 때 그에 대응하는 출력을 예측 (predict)하는 분야이다.

인공 신경망은 여러 개의 퍼셉트론 노드 (node)로 이루어진 층 (layer)과 이런 층들이 이어져 다층 구조 (multi-layer)로 이루어진 형태이다.



그림 5. 1 인공 신경망 구조

데이터를 입력 받는 layer를 input layer라 하고, 연산 결과가 출력되는 layer를 output layer라 한다. 그 사이에 있는 layer들의 연산과정은 관찰자에게 보이지 않기 때문에 hidden layer라 한다. 각 노드들에는 가중치 (weight, *w*)와 바이어스 (bias, *b*)가 있으며, 출력은 다음과 같다.

여기서 *θ*는 노드를 구성하는 매개 변수들인 가중치와 바이어스를 의미한다. 학습 데이터의 입력을 *x*, 출력을 *y*라 하면, 초기 학습되지 않은 상태의 인공 신경망에 *x*를 입력하면 가 출력된다. 이때 *y*와 의 차이, loss function을 다음과 같이 정의한다.

이 때 *N*은 출력의 크기이다. 학습은 loss를 최소화하는 가중치와 바이어스를 찾는 것을 의미한다. 이 과정은 gradient descent method [28]를 통해 이루어진다. Gradient descent method는 함수의 기울기가 낮은 쪽으로 계속 이동하며 최솟값 (global minimum value)을 찾는 방법이다.



그림 5. 2 Gradient descent method 개념도

인공 신경망에서 입력과 출력은 정해져 있기 때문에 loss와 연관된 변수는 *θ*뿐이다. 따라서, loss와 *θ*의 미분 관계를 구하고 이를 사용하여 *θ*를 갱신하는 과정을 반복한다. 실제 loss와 *θ*의 관계를 직접적으로 구할 수 없으므로 연쇄 법칙 (chain rule)을 사용한다. 또한, *θ*를 갱신하기 위해 반복하는 단계를 learning step이라 한다.

이때, 미분 관계를 갱신에 바로 사용하지 않고 학습율 (learning rate, *r*)을 곱해주어 사용한다. 이는 learning step에서 gradient가 *θ*를 갱신하는 효과를 조절해준다. 학습율이 너무 작다면 학습이 느리게 진행 되고, 너무 크다면 최솟값을 지나칠 수 있다. 따라서 적당한 학습율을 사용하는 것이 중요하다. 보통 1e-2 ~ 1e-3 사이의 값에서 시작하여 학습 속도와 가능 여부를 확인하며 조절한다.



그림 5. 3 부적합한 학습율 사용으로 인해 나타나는 문제점

Loss가 0이 되는 것은 거의 불가능 하기 때문에 허용 오차 (tolerance)를 사용하여 반복을 멈추거나, 최대 learning step을 정해준다. 이를 의사 코드로 나타내면 다음과 같다.

**Algorithm 5.1** Gradient descent method for artificial neuron

**Input**: training dataset *X*,

learning rate *r*

maximum number of iteration *I*,

tolerance τ

**Output**: weights ***w***,

bias *b*

1: Randomly initialize **w** and b

2: **for** *c* = 1 : *I* **do**

3: **for** *j* = 1 : *M* **do**

4: // *L* is the loss function

5: **end for**

6:

7: **if** *L*(*X*; ***w***, *b*) < τ **then**

8: **break**

9: **end if**

10: **end for**

11: **return** **w**, b

Algorithm 5.1의 의사 코드는 labeled 데이터를 사용하여 loss function을 계산할 수 있는 single layer에서만 적용된다. 실제 hidden layer가 존재하게 되면 hidden layer 사이에서는 labeled 데이터가 존재하지 않기 때문에 loss function을 정의할 수 없는 문제가 발생한다. 이를 해결하기 위하여 고안된 것이 역전파 알고리즘 (backpropagation algorithm)이다. 역전파 알고리즘은 출력에서 계산되는 loss 를 hidden layer의 매개 변수와 연결시키주는 방법이다. Loss를 앞에서 뒤로 보낸다는 의미로 ‘역 (back)’을 사용한다. 먼저 *N*을 인공 신경망을 구성하는 layer의 수, , , 은 *n*번째 layer를 구성하는 *i*번째 노드의 *j*번째 가중치, 바이어스, 출력이다. 또한, *n*번째 layer의 입력은 *n* - 1 번째 layer의 출력과 같다.

가중치 갱신에 대한 gradient descent method 관계식은 다음과 같다.

그리고 loss와 노드의 출력과의 관계를 나타내는 변수 다음과 같이 정의한다.

*n*번째 layer를 구성하는 노드의 개수를 *Kn*, 입력되는 데이터의 크기를 *Mn*이라하면, hidden layer 중 output layer와 가장 가까운 *N* - 1 layer를 구성하는 *i*번째 노드의 *j*번째 가중치에 대한 갱신 관계식은 다음과 같다.

이 식으로부터 을 저장하여 학습할 때 사용한다면, *n* - 1 번째 layer와 loss 사이의 관계를 구할 수 있는 것을 알 수 있다. 따라서, output layer에서 계산되는 loss를 hidden layer까지 전달하여 hidden layer의 매개 변수와 loss 사이의 관계식을 구할 수 있게 된다.

인공 신경망 학습을 위해 사용하는 역전파 알고리즘 수행 함수를 optimizer라고 한다. optimizer들은 gradient descent method을 시작으로 하여 여러 문제점들을 개선하며 개발되었다. 그 중에서 대표적으로 사용하는 optimizer는 Adam optimizer [29]이다. 그림 5.4와 같이 optimizer의 성능을 이미지 인식 인공 신경망 학습으로 비교해보면 Adam이 가장 빠르게 학습한다.



**그림 5. 4** Optimizer 성능 비교 [29]

단일 퍼셉트론에서는 계단 함수만을 활성화 함수로 사용하였지만, 활성화 함수는 더 다양하게 sigmoid, tanh, Relu (Rectified Linear Unit)등이 있다. Sigmoid는 사람의 뉴런과 가장 유사하여 초기에 가장 많이 사용하였다. 그러나 0 또는 1에 가까워질수록 gradient가 매우 작아져 역전파과정에서 loss가 전달되지 못하는 vanishing gradient 문제점이 발견되었다. tanh는 sigmoid에서 중심값이 0이 아니라는 문제점을 해결하였지만, 여전히 vanishing gradient 문제가 있다. Relu는 선형 그래프를 꺾은 형태로 vanishing gradient 문제를 해결하여 현재 가장 많이 사용되고 있다.



그림 5. 5 활성화 함수 (a) sigmoid (b) tanh (c) Relu

5.1.2 오버피팅(Overfitting)

인공 신경망을 학습시킬 때 단순히 많은 양의 학습 데이터를 제공한다 하여도 정확한 결과가 나타나지 않을 수 있다. 또는 학습 과정에서 train loss만 봤을 때는 학습이 제대로 이루어 진 것으로 보이나, 테스트 과정에서 test loss가 커서 실제 사용을 못하는 경우도 있다. 또한 같은 데이터를 반복 학습하여 정확한 학습이 되도록 할 수도 있지만, 학습 데이터에 한정된 행동만 할 수 있다. 이러한 방식으로 학습이 제대로 이루어지지 않는 현상을 오버피팅이라고 한다.



그림 5. 6 Regression 결과 (a) 학습이 잘 된 경우 (b) 언더피팅 (c) 오버피팅

다양한 분포를 가지는 데이터 세트를 사용하거나, 인공 신경망이 학습해야하는 특징들을 줄이는 것이 일반적인 해결법이다.

5.1.3 합성곱 신경망 (Convolutional Neural Network)

인공 신경망이 학습해야하는 특징들을 줄이는 방법으로 convolutional neural network (CNN) [30]를 사용한다. CNN은 고양이의 시각 피질 (visual cortex) 반응에 대한 연구에서 출발하였다 [30]. 고양이 뇌의 시각 피질에 전극을 꽂은 상태에서 기울어진 흰색 막대를 보여주면서 시각 피질에 자극을 주어 전극에서 나오는 신호를 증폭하여 오디오로 듣는 실험이다. 결론적으로 이미지를 볼 때 전체를 인식하는 것이 아니라, 이미지를 구성하는 edge들의 합성을 통해 인식한다는 것이다. 인공 신경망에서는 convolution layer가 이런 역할을 수행한다.

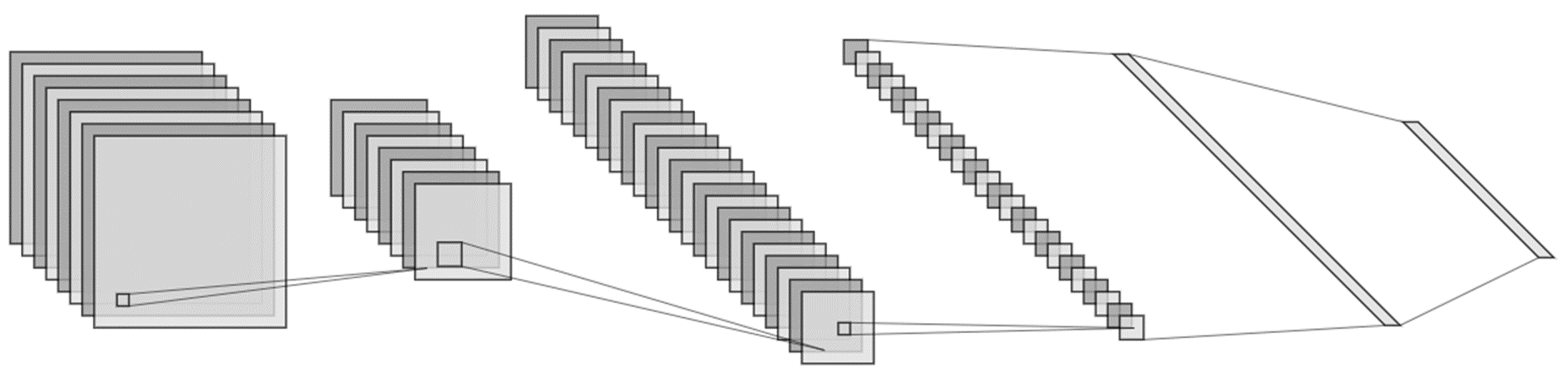


그림 5. 7 CNN 개념도

Convolution layer는 그림 5.8과 같이 입력의 일정 부분을 filter와 합성곱 (convolution) 연산을 통해 하나의 값으로 만들어 낸다. Filter는 일정 간격으로 데이터위에서 움직이는데 이 간격을 stride라 한다. 연산 결과인 feature map은 input보다 크기가 줄어들게 된다. 이를 방지 하기 위하여 feature map의 외각을 0으로 padding을 한다.

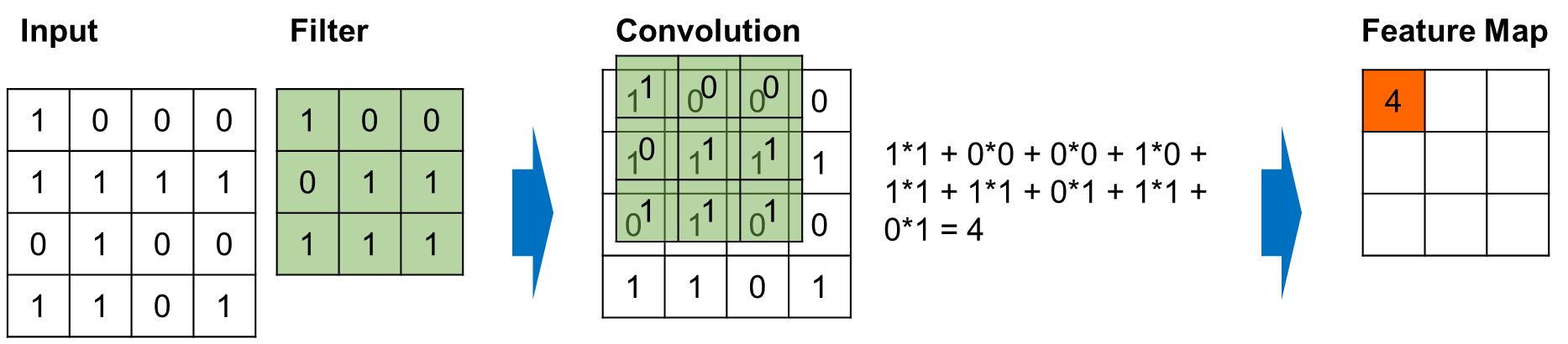


그림 5. 8 Convolution layer 계산 과정

Feature map을 그대로 사용하지 않고 크기를 줄이거나 특정 데이터를 강조하기 위한 과정을 거치는데, 이를 pooling이라 한다. Pooling 방법에는 대표적으로 average pooling과 max pooling이 있다. Average pooling은 일정 영역의 해당하는 값들의 평균값을, max pooling은 최대값을 사용하는 방법이다. 그림과 같이 값이 큰 데이터들 사이에서 작은 값의 데이터를 찾아야 하는 경우 average pooling가 적합하고, 반대로 값이 작은 데이터들 사이에서 큰 값의 데이터를 찾아야 하는 경우 max pooling가 적합하다 [32]. 일반적으로 max pooling을 많이 사용한다.

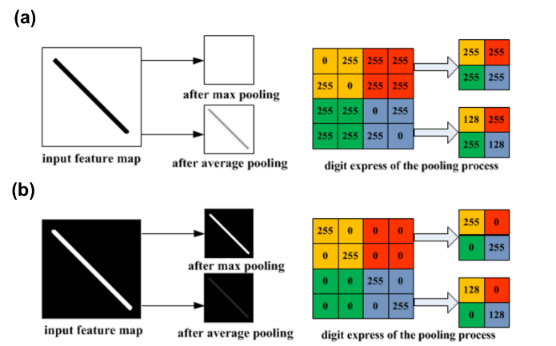


그림 5. 9 입력 데이터에 따른 pooling 결과 비교 [32]

CNN의 마지막 단에는 일반 flat한 인공 신경망을 사용하였다. 이때는 hidden layer를 1~2개 정도 사용하여 output layer와 연결한다. CNN에서 learning이란, convolution layer의 filter와 마지막 단 hidden layer의 가중치와 바이어스를 찾는 과정을 의미한다.

5.1.4 ResNet

일반적으로 layer의 개수가 증가하면 연산하는 노드의 수가 증가하여 연산 자유도 또한 증가하기 때문에 해결할 수 있는 문제의 복잡도 (complexity)가 증가한다. Layer의 개수가 많은 인공 신경망을 hidden layer가 깊어진다 (deep)는 의미로 deep neural network라고 부른다. 하지만 layer의 개수가 증가하면 loss를 계산하는 output layer에서 멀어질수록 loss의 영향이 줄어들기 때문에 역전파 알고리즘을 이용한 학습이 어려워진다. 이미지 인식 문제를 해결하기 위해 개발된 CNN에서도 학습율을 높이기 위해 layer의 개수를 증가시키면서 비슷한 문제가 발생하였다. 이러한 문제를 해결하고 학습율을 높이기 위해 개발된 인공 신경망이 ResNet이다 [33]. ResNet이 일반 인공 신경망과 다른 점은 그림 5.10과 같은 residual block이 있다는 것이다. Residual block에는 입력과 두 layer이후를 연결해주는 skip connection이 있어 loss가 역전파 될 때 output layer에서 멀어질수록 그 영향이 적어지는 정도를 줄일 수 있다.



그림 5. 10 인공 신경망 layer의 연결 방법 (a) 일반 layer 연결 (b) residual block

5.2 Inverse Design

Inverse design은 설계하고자 하는 소자의 목표 성능을 지정하고, 이를 만족시키도록 최적화된 소자의 모양을 찾아가는 수치해석법이다. 이전에는 inverse design과 반대되는 개념인 forward design으로 소자를 설계하였다. Forward design은 소자의 구조를 확정하고 결과를 분석하는 방법으로 원하는 소자를 얻기 위해 구조를 임의로 조금씩 변경해야한다. 따라서, 설계자의 경험이나 직관에 따라 소요되는 시간과 결과물의 성능이 일정하지 않다는 단점이 있다. 이와 다르게 inverse design은 설계자의 간섭없이 정해진 알고리즘 대로 설계가 진행되기 때문에 결과물의 성능이 일정하다. 초기에는 도파관과 같은 간단한 구조에서만 사용되었지만, 연산 장비 성능과 프로그램의 발전으로 좀 더 복잡한 구조에서도 사용할 수 있게 되었다 [34].

Inverse design는 크게 evolutionary 알고리즘과 gradient-based 알고리즘으로 나뉜다 [35]. Evolutionary 알고리즘은 생물의 진화에서 아이디어를 얻어 만들어진 알고리즘으로 genetic 알고리즘과 particle swarming optimization 등이 있다. Genetic 알고리즘은 생물이 세대 (generation)에 걸쳐 진화하면서 일어나는 유전, 돌연변이등의 현상을 활용한 알고리즘이다 [36]. Particle swarming optimization은 개체가 자기 방어나 음식물을 찾기 위해 군집으로 행동하는 사회적 행동 양식을 아이디어로 한다 [37]. 군집은 서로 상호작용하고 협동하기 때문에 목표 달성에 효과적인 점을 이용하였다. Gradient-based 알고리즘은 수치적으로 구조와 성능의 미분 관계를 구하여 최솟점을 찾는 방법으로 topology optimization [38]이 이에 속한다.

5.3 인공 신경망을 이용한 Inverse Design

5.3.1 인공 신경망을 이용한 Inverse Design 방법

인공 신경망을 이미 컴퓨터 비전 [30], 음성 인식 [39]등의 분야에서 좋은 성과들을 보이며 사용되고 있다. 최근 들어 광학 소자 설계에서도 Inverse design의 문제점을 보완하고자 인공 신경망을 이용하여 설계하는 시도들이 증가하고 있다 [40-42].

기존 inverse design인 evolutionary 알고리즘은 매우 많은 반복을 필요로 하기 때문에 시간이 오래 걸리는 문제점이 있으며, gradient-based 알고리즘은 gradient를 numerical method로 구하기 때문에 요구하는 연산 장비 성능이 높고, 시간이 오래 걸리는 문제점이 있다. 간단한 연산으로 이루어진 인공 신경망으로 구조의 optical response계산을 대체하면 gradient를 analytical method로 구할 수 있다. 따라서 evolutionary 알고리즘보다 적은 반복 횟수와 gradient-based 알고리즘보다 적은 시간으로 설계를 진행할 수 있다.

5.3.2 인공 신경망을 이용한 DBR Inverse Design

설계 목표와 플랫폼은 그림 4.3의 ARL을 이용한 DBR 설계와 같다. 인공 신경망은 100개의 노드를 가지는 hidden layer가 4개로 구성되어 있는 구조이다. 총 20,000개의 데이터에서 80%인 16,000개의 데이터는 10번 복제하여 160,000개로 만들어 학습에 사용하였다. 20,000개의 데이터에서 나머지 20%는 학습된 인공 신경망 테스트에 사용하였다. 한 step당 한 개의 데이터를 학습하는 것이 아닌, 100개의 데이터를 학습시켰으며, 이러한 양을 batch size라 한다. Optimizer는 학습율 1e-3을 적용한 Adam을 사용하였다. 인공 신경망 학습 결과는 그림 5.11과 같다.

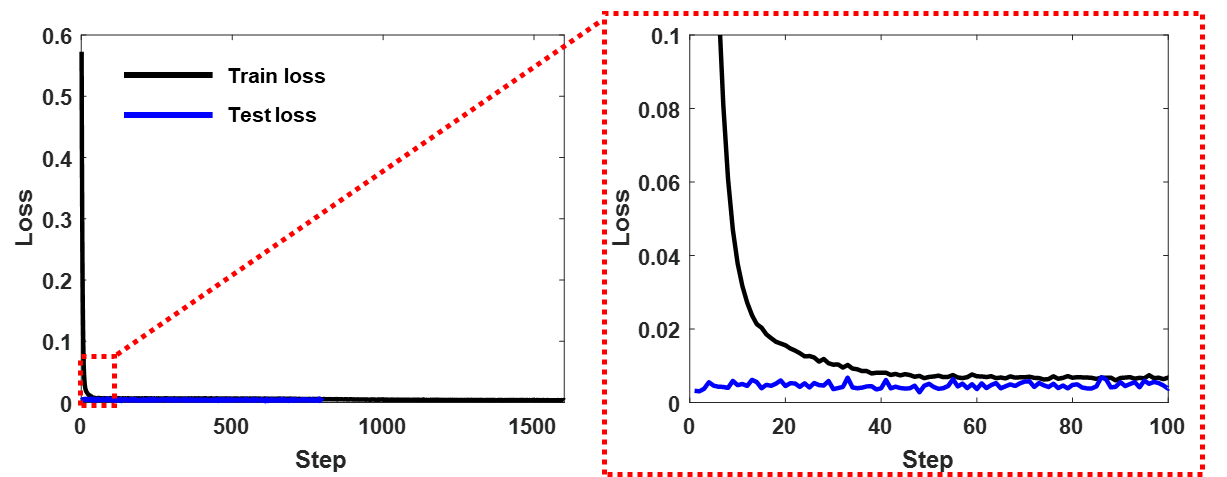


그림 5. 11 DBR prediction 인공 신경망 학습 결과

노드와 layer의 개수를 변경하면 loss가 변하기는 하나 그 크기가 매우 작기 때문에 비교하지 않았다. 학습은 총 1600 step 수행하였다. Train loss는 100 step이후 0.005로 수렴하였으며, test loss도 그와 비슷한 수준이었다. 인공 신경망이 DBR의 optical response 계산이 가능하도록 학습되었다. Inverse design에서 최대화하고자 하는 보상 *r*을 다음과 같이 정의하였다.

Inverse design에서 사용하는 optimizer로도 학습율 1e-3을 적용한 Adam을 사용하였다. 그 결과는 다음과 같다.

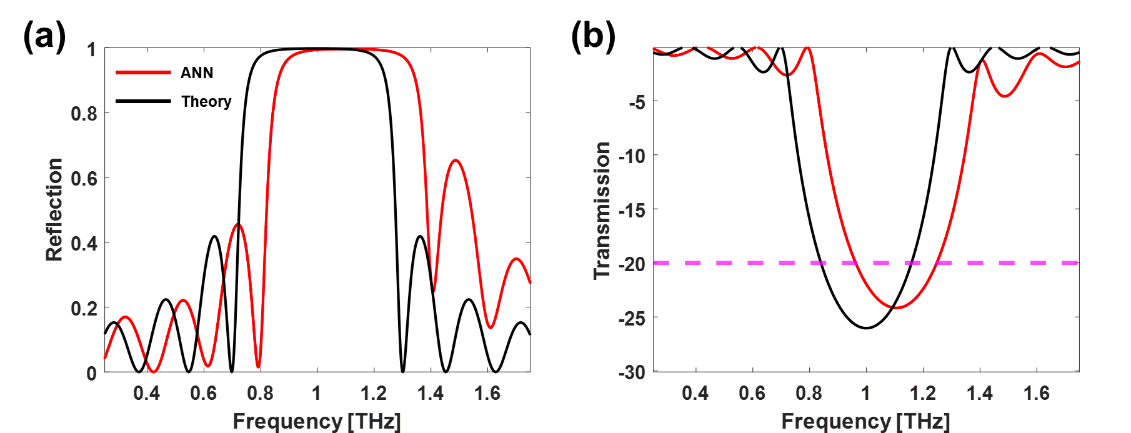


그림 5. 12 ANN으로 설계한 DBR과 이론 DBR의 (a) 반사율 (b) 투과율 비교, 점선은 transmission이 99%인 지점은 의미한다.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Maximum Reflectance | FWHM | 99% bandwidth |
| ANN | 0.9962 at  1.11 THz (270 μm) | 0.584 THz  (155 μm) | 0.339 THz  (85 μm) |
| Theory | 0.9978 at  0.99 THz (300 μm) | 0.553 THz  (180 μm) | 0.354 THz  (110 μm) |

표 5. 1 ANN으로 설계한 DBR과 이론 DBR의 반사율 비교

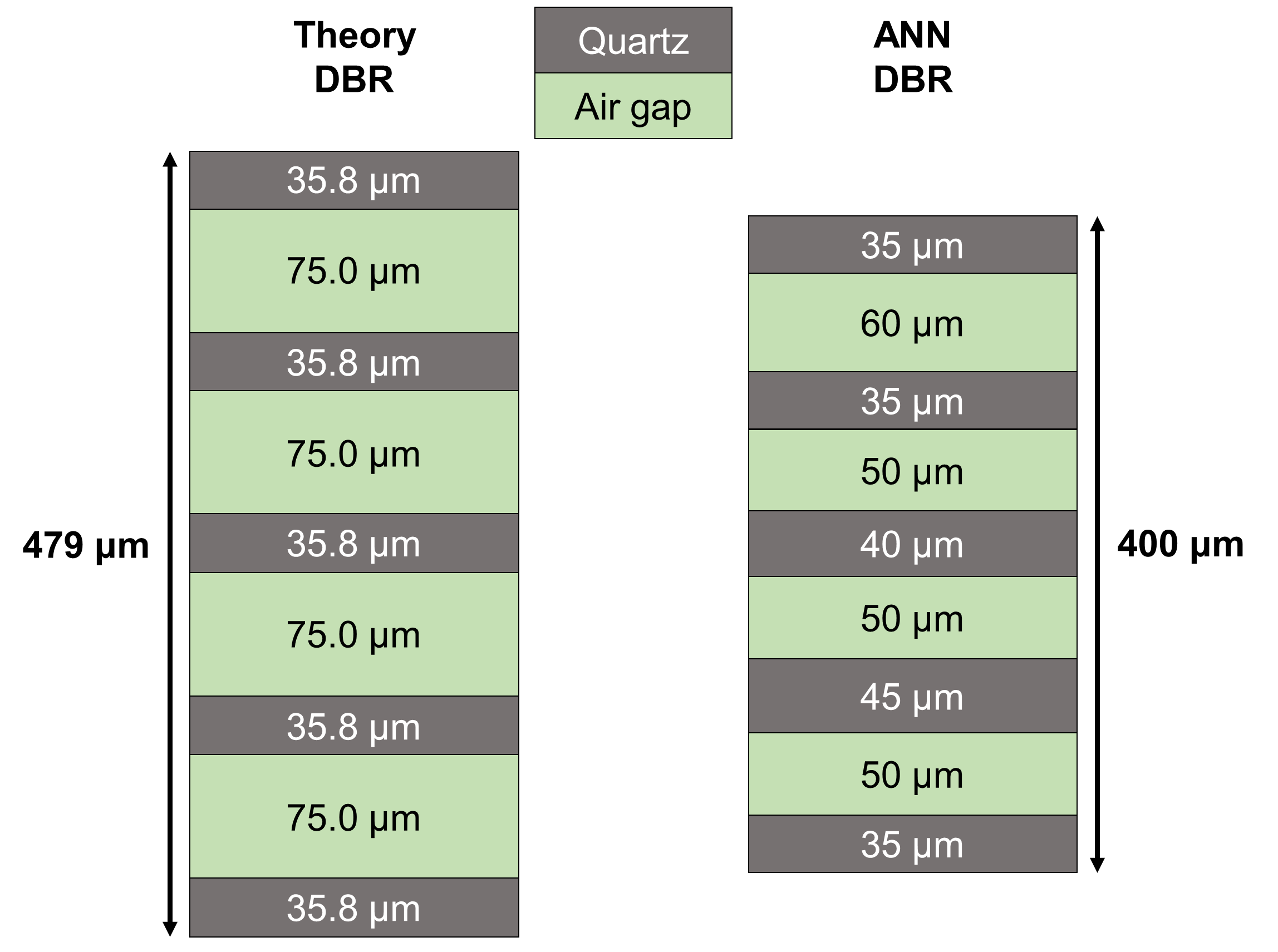


그림 5. 13 ANN으로 설계한 DBR과 이론 DBR의 두께 비교

ANN으로 설계한 DBR이 이론으로 설계한 DBR보다 수치적으로 부족한 성능을 보인다. 하지만 설계 주파수 1 THz에서의 반사율은 모두 99%를 넘으며, 같은 pair이면서도 전체 길이가 공진 조건을 고려하여 설계한 DBR보다 약 100 μm 짧기 때문에 공간 제한 문제가 있는 환경에서 유용하게 사용될 수 있다.

5.3.3 1×2 OPS Prediction 인공 신경망

설계 목표와 플랫폼은 그림 4.8의 ARL을 이용한 1×2 OPS 설계와 같다. 인공 신경망을 이용한 OPS 설계에서 현재 문제점은 OPS prediction 인공 신경망을 학습시키는 것이다. OPS는 DBR보다 구조와 해석이 복잡하여 필요로 하는 인공 신경망 학습 정확도가 높다. 기존에 있던 연구에서는 layer간의 노드들을 모두 연결한 인공 신경망인 FCDNN (Full Connected Deep Neural Network)의 부족한 학습 정확도를 CNN 구조를 사용함으로써 보완하였다 [12].

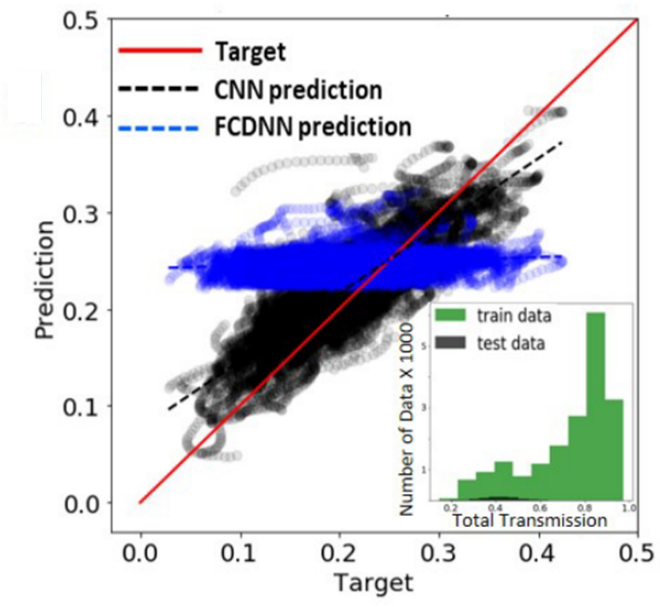


그림 5. 14 CNN을 이용한 OPS prediction 인공 신경망 학습 테스트 [12]

인공 신경망 구조를 바꾸어 상관계수를 0.16에서 0.85까지 증가시켰지만 inverse design에 사용하기에는 부족한 정확도이다. 본 논문에서는 데이터 분포 개선과 인공 신경망 구조 변경을 병행하여 상관 계수를 상승시켰다. 학습 데이터를 보상 min(T)에 대해 분포도를 나타내면 다음과 같다.



그림 5. 15 OPS prediction 인공 신경망 학습 데이터 분포도

인공 신경망은 FCDNN과 ResNet 두 구조를 비교하며 사용하였다. FCDNN은 100개의 노드를 가지는 hidden layer가 4개가 있는 구조이고, ResNet는 16개의 residual block으로 이루어져 있다. 무작위로 생성된 7,000개의 데이터와 PSO로 생성한 18,800 개의 데이터를 합하여 총 25,800개의 데이터를 만들었다. 전체 데이터에서 90%인 23,220개의 데이터는 5번 복제하여 116,100개로 만들어 학습에 사용하였다. 나머지 10%는 학습된 인공 신경망 테스트에 사용하였다. Batch size는 100으로 설정하고, Optimizer는 학습율 1e-3을 적용한 Adam을 사용하였다. 인공 신경망 학습 결과는 그림 5.16과 같다.

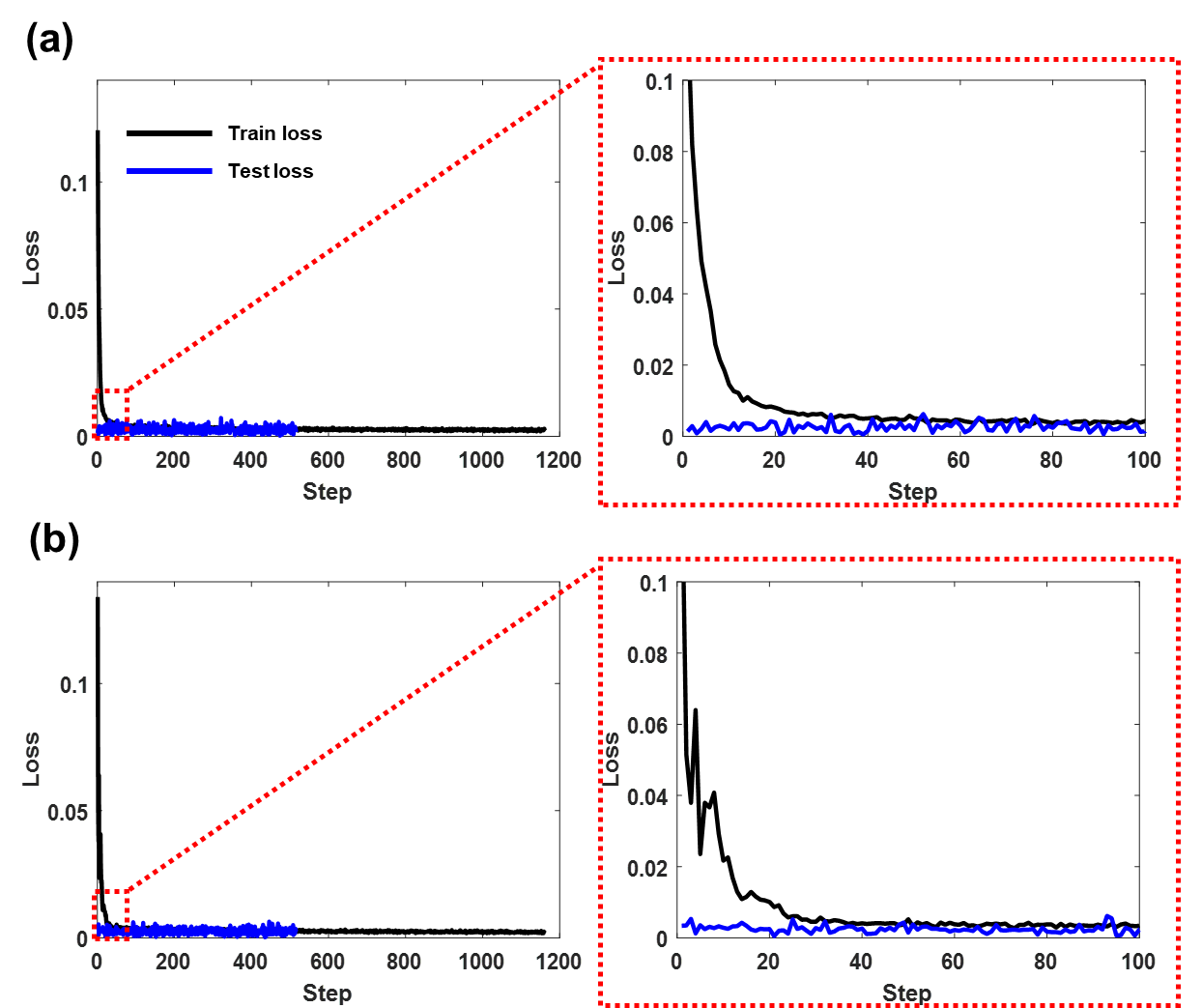


그림 5. 16 OPS prediction 인공 신경망 학습 결과 (a) FCDNN (b) ResNet

학습은 총 1161 step 수행하였다. Train loss는 100 step이후 FCDNN의 경우 0.0026, ResNet의 경우 0.0025로 수렴하였으며, test loss도 그와 비슷한 수준이었다. 구조 변경으로 인한 정확도 차이는 매우 작은 수준이다.

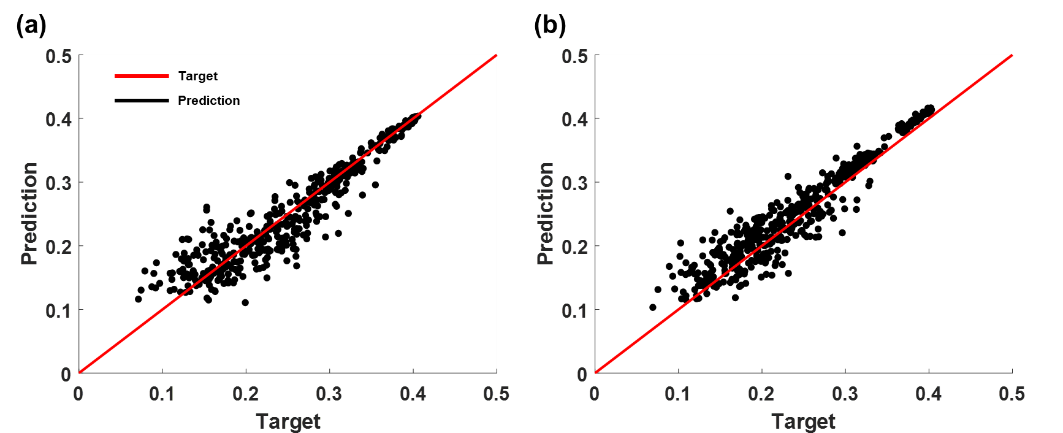


그림 5. 17 OPS prediction 인공 신경망 학습 테스트 (a) FCDNN (b) ResNet

두 구조는 상관 계수에서 차이를 보인다. FCDNN의 상관 계수는 0.947, ResNet의 상관 계수는 0.960이다. 두 값 모두 앞에서 서술한 CNN을 이용한 결과인 0.85보다 높았다. 이를 보았을 때 인공 신경망 구조 변경으로 prediction 정확도 향상은 가능하지만 그 정도가 작으며, 균일한 분포의 학습 데이터를 사용하는 것이 정확도 향상에 많은 도움이 되는 것을 알 수 있다.

두 구조 모두 loss가 0.0025 수준으로 수렴하였다. 이 때 loss를 mean square error로 정의하였기 때문에 실제값과 prediction의 차이는 약 0.05이다. 이는 OPS의 투과율 범위인 0 ~ 0.5에서 상대적으로 무시할 수 없는 크기이며, 식각 구조가 조금만 다른 두 구조의 투과율 차이보다 크다. 따라서 학습 데이터의 양을 증가 시키거나, 새로운 구조의 인공 신경망을 사용하여 더 높은 정확도의 OPS prediction 인공 신경망이 개발되어야 한다.

Ⅵ. 결론

최근 통신 분야에서 필요로 하는 데이터 전송량이 늘어나면서 테라헤르츠 통신 연구가 활발해지고 있다. 테라헤르츠는 대기중에 흡수되는 양이 많기 때문에 아주 먼 거리에서의 무선 통신은 어렵다. 따라서 테라헤르츠 통신을 넓은 분야에서 사용하기 위해서는 광케이블을 이용한 테라헤르츠 광통신 연구가 진행되어야 한다. 본 논문은 테라헤르츠 광통신 연구를 위한 기초연구로 머신 러닝을 활용하여 테라헤르츠 대역에서 동작하는 DBR과 1×2 OPS를 설계하였다.

먼저, ARL 알고리즘을 이용하여 DBR과 1×2 OPS를 설계하였다. 20,000개의 학습 데이터와 보상을 로 정의하여 DBR을 설계하였다. 설계한 DBR은 공진 조건을 고려한 DBR과 비슷한 성능을 보이면서도 약 100 μm 더 작은 (compact) 크기에서 동작하였다. 40,000개의 학습 데이터와 보상을 로 정의하여 1×2 OPS를 설계하였다. 설계한 1×2 OPS 는 입력 대비 출력의 내부 손실이 3.71 dB, 균일도는 0.75dB 였다. 또한 기존 OPS의 변환 손실을 줄이기 위해 branch distance가 길어야 하는 문제점을 500 μm × 500 μm 에서 OPS를 설계함으로써 해결할 수 있었다.

다음으로, ANN과 inverse design을 이용하여 DBR과 1×2 OPS를 설계하였다. ARL에서 사용한 것과 같은 학습 데이터와 보상, 100개의 노드로 이루어진 4개의 hidden layer 로 구성되어 있는 FCDNN을 이용하여 DBR을 설계하였다. 설계한 DBR은 공진 조건을 고려한 DBR과 비슷한 성능을 보이면서도 약 100 μm 더 작은 크기에서 동작하였다. 1×2 OPS는 인공 신경망의 prediction 정확도를 향상 시키는 것이 쟁점이다. 선행 연구에서는 인공 신경망의 구조를 FCDNN에서 CNN으로 바꾸어 상관 계수를 0.15에서 0.85로 상승시켰다. 본 논문에서는 우선적으로 무작위로 생성한 데이터와 PSO로 생성한 데이터를 합하여 학습 데이터의 분포를 개선함으로써 FCDNN 구조에서 상관 계수를 0.94로 상승시켰다. 또한 인공 신경망 구조를 skip connection을 사용하는 ResNet으로 바꾸어 상관 계수를 0.96까지 상승시켰다. 하지만 완전한 구조를 설계하기에는 아직 정확도가 부족하기 때문에 더 많은 학습 데이터를 사용하거나, 인공 신경망의 구조를 개선하는 등의 노력이 필요하다.

이처럼 머신 러닝을 활용하여 테라헤르츠에서 동작하는 DBR과 1×2 OPS 설계 특성을 살펴봄으로써 광학 소자 설계에 필요한 머신 러닝의 조건과 각 알고리즘의 장단점을 알 수 있었다. 현재 개발되어 있는 머신 러닝 알고리즘들과 본 연구결과를 활용한다면 복잡한 기능을 수행하는 광학 소자를 적은 연산 자원과 시간만으로도 효율적으로 개발할 수 있을 것으로 기대된다.

REFERENCES

1. F. Akyildiz, J. M. Jornet, and C. Han, “Terahertz band: Next frontier for wireless communications,” Physical Communication, vol. 12, 2014, pp. 16–32.
2. S. Cherry, “Edholms law of bandwidth,” IEEE Spectrum, vol. 41, no. 7, 2004, pp. 58–60.
3. J. Federici and L. Moeller, “Review of terahertz and subterahertz wireless communications,” Journal of Applied Physics, vol. 107, no. 11, 2010, p. 111101.
4. T. Kleine-Ostmann and T. Nagatsuma, “A Review on Terahertz Communications Research,” Journal of Infrared, Millimeter, and Terahertz Waves, vol. 32, no. 2, 2011, pp. 143–171.
5. H.-J. Song and T. Nagatsuma, “Present and Future of Terahertz Communications,” IEEE Transactions on Terahertz Science and Technology, vol. 1, no. 1, 2011, pp. 256–263.
6. L. Lu, M. Zhang, F. Zhou, and D. Liu, “An Ultra-compact Colorless 50:50 Coupler Based on PhC-like Metamaterial Structure,” Optical Fiber Communication Conference, 2016.
7. K. Chandu, M. Stanich, C. W. Wu, and B. Trager, “Direct binary search (DBS) algorithm with constraints,” Color Imaging XVIII: Displaying, Processing, Hardcopy, and Applications, 2013.
8. P. Besse, E. Gini, M. Bachmann, and H. Melchior, “New 2×2 and 1×3 multimode interference couplers with free selection of power splitting ratios,” Journal of Lightwave Technology, vol. 14, no. 10, 1996, pp. 2286–2293.
9. Y. Tian, J. Qiu, M. Yu, Z. Huang, Y. Qiao, Z. Dong, and J. Wu, “Broadband 1 × 3 Couplers With Variable Splitting Ratio Using Cascaded Step-Size MMI,” IEEE Photonics Journal, vol. 10, no. 3, 2018, pp. 1–8.
10. M. Bachmann, P. A. Besse, and H. Melchior, “General self-imaging properties in N × N multimode interference couplers including phase relations,” Applied Optics, vol. 33, no. 18, 1994, p. 3905.
11. L. Soldano and E. Pennings, “Optical multi-mode interference devices based on self-imaging: principles and applications,” Journal of Lightwave Technology, vol. 13, no. 4, 1995, pp. 615–627.
12. I. Malkiel, M. Mrejen, A. Nagler, U. Arieli, L. Wolf, and H. Suchowski, “Plasmonic nanostructure design and characterization via Deep Learning,” Light: Science & Applications, vol. 7, 2018, no. 1.
13. T. M. Mitchell, Machine Learning. New York: McGraw-Hill, 1997.
14. Anaconda Software Distribution. Computer software. Vers. 2-2.4.0. Anaconda, Nov. 2016. Web. <https://anaconda.com>.
15. Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, Xiaoqiang Zheng, “Tensorflow: Large-scale machine learning on heterogeneous distributed systems,” arXiv preprint arXiv:1603.04467, 2016.
16. S. L. Chuang, Physics of optoelectronic devices. New York, NY: Wiley, 1995.
17. Steven J. Byrnes, “Multilayer optical calculations,” arXiv preprint arXiv:1603.02720v3 [physics.comp-ph]
18. L. Han, B. P.-P. Kuo, N. Alic, and S. Radic, “Ultra-broadband multimode 3dB optical power splitter using an adiabatic coupler and a Y-branch,” Optics Express, vol. 26, no. 11, 2018, p. 14800.
19. F. Costache, H. Hartwig, K. Bornhorst, and M. Blasl, “Variable Optical Power Splitter with Field-Induced Waveguides in Liquid Crystals in Paranematic Phase,” Optical Fiber Communication Conference, 2014.
20. W. S. C. Chang, Fundamentals of guided-wave optoelectronic devices. Cambridge: Cambridge University Press, 2010.
21. Lumerical Inc. https://www.lumerical.com/products/
22. F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain.,” Psychological Review, vol. 65, no. 6, 1958, pp. 386–408.
23. V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, “An Introduction to Deep Reinforcement Learning,” Foundations and Trends® in Machine Learning, vol. 11, no. 3-4, 2018, pp. 219–354.
24. Çağrı Latifoğlu, “Binary Matrix Guessing Problem,” arXiv preprint arXiv:1701.06167v2, 2017.
25. M. Turduev, E. Bor, C. Latifoglu, I. H. Giden, Y. S. Hanay, and H. Kurt, “Ultracompact Photonic Structure Design for Strong Light Confinement and Coupling Into Nanowaveguide,” Journal of Lightwave Technology, vol. 36, no. 14, 2018, pp. 2812–2819.
26. C. L. Davies, J. B. Patel, C. Q. Xia, L. M. Herz, and M. B. Johnston, “Temperature-Dependent Refractive Index of Quartz at Terahertz Frequencies,” Journal of Infrared, Millimeter, and Terahertz Waves, vol. 39, no. 12, 2018, pp. 1236–1248.
27. E. D. Palik, Handbook of optical constants of solids. London: Academic Press, 1998.
28. D. P. Bertsekas, Nonlinear programming, 2nd ed. Athena Scientific, 1999.
29. Diederik P. Kingma, Jimmy Ba, “Adam: A Method for Stochastic Optimization,” arXiv:1412.6980v9 [cs.LG]
30. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” Communications of the ACM, vol. 60, no. 6, 2017, pp. 84–90.
31. D. H. Hubel and T. N. Wiesel, “Receptive fields, binocular interaction and functional architecture in the cats visual cortex,” The Journal of Physiology, vol. 160, no. 1, 1962, pp. 106–154.
32. D. Yu, H. Wang, P. Chen, and Z. Wei, “Mixed Pooling for Convolutional Neural Networks,” Rough Sets and Knowledge Technology Lecture Notes in Computer Science, 2014, pp. 364–375.
33. K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
34. S. Molesky, Z. Lin, A. Y. Piggott, W. Jin, J. Vucković, and A. W. Rodriguez, “Inverse design in nanophotonics,” Nature Photonics, vol. 12, no. 11, 2018, pp. 659–670.
35. K. Yao, R. Unni, and Y. Zheng, “Intelligent nanophotonics: merging photonics and artificial intelligence at the nanoscale,” Nanophotonics, vol. 8, no. 3, 2019, pp. 339–366.
36. M. Mitchell, An introduction to genetic algorithms. Cambridge, MA: MIT, 1998.
37. Y. Shi and R. Eberhart, “Empirical study of particle swarm optimization,” Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406).
38. J. Jensen and O. Sigmund, “Topology optimization for nano-photonics,” Laser & Photonics Reviews, vol. 5, no. 2, 2010, pp. 308–321.
39. G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, “Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups,” IEEE Signal Processing Magazine, vol. 29, no. 6, 2012, pp. 82–97.
40. D. Liu, Y. Tan, E. Khoram, and Z. Yu, “Training deep neural networks for the inverse design of nanophotonic structures,” Conference on Lasers and Electro-Optics, 2019.
41. J. Peurifoy, Y. Shen, L. Jing, Y. Yang, F. Cano-Renteria, B. G. Delacy, J. D. Joannopoulos, M. Tegmark, and M. Soljačić, “Nanophotonic particle simulation and inverse design using artificial neural networks,” Science Advances, vol. 4, no. 6, 2018.
42. X. Lin, Y. Rivenson, N. T. Yardimci, M. Veli, Y. Luo, M. Jarrahi, and A. Ozcan, “All-optical machine learning using diffractive deep neural networks,” Science, vol. 361, no. 6406, 2018, pp. 1004–1008.