

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

THz Photonic Device Design by Machine Learning

by

Jonggeon Lee

Department of Electrical Engineering

Pohang University of Science and Technology

A thesis submitted to the faculty of the Pohang University of Science
and Technology in partial fulfillment of the requirements for the degree
of Master of Science in the Electrical Engineering

Pohang, Korea

12. 30. 2019

Approved by

Haewook Han (Signature)

Academic Advisor

THz Photonic Device Design by Machine Learning

Jonggeon Lee

The undersigned have examined this thesis and hereby
certify that it is worthy of acceptance for a master's degree
from POSTECH

12/30/2019

Committee Chair 한 해 욱 (seal)

Member 공 병 돈 (seal)

Member 박 기 수 (seal)

MEE

20182737

이종건 Jonggeon Lee,

THz Photonic Device Design by Machine Learning.

머신 러닝을 활용한 테라헤르츠 광학소자 설계,

Department of Electrical Engineering, 2020, 57P,

Advisor: Haewook Han

ABSTRACT

Terahertz (THz) technology has attracted due to potentials on basic science and applied engineering. Also, Photonic Integrated Circuits (PICs) based on Silicon-on-insulator (SOI) platform have become a new paradigm for an on-chip subsystem due to a large bandwidth, resistance to interference and nonexistence of Joule effect. We designed the two basic THz PIC devices, Distributed Bragg Reflector (DBR) and 1×2 Optical Power Splitter (OPS), 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
1.1 테라헤르츠 전자파.....	1
1.2 머신 러닝.....	3
1.3 머신 러닝 도구 및 전산 자원.....	6
II. 분산 브레그 반사경	7
2.1 다층 구조에서의 전달 행렬(Transfer Matrix).....	7
2.2 분산 브레그 반사경 (Distributed Bragg Reflector)의 구조와 특성	10
2.3 DBR 전산 모사 도구.....	11
III. 1×2 광전력 분배기.....	12
3.1 1×2 광전력 분배기 (Optical Power Splitter).....	12
3.2 1×2 OPS 전산 모사 도구.....	14
IV. 가법 강화 학습.....	15
4.1 퍼셉트론(Perceptron).....	15
4.2 강화 학습(Reinforcement Learning)	16
4.3 가법 강화 학습(Additive Reinforcement Learning)을 이용한 설계	17
4.3.1 가법 강화 학습.....	17
4.3.2 ARL 을 이용한 DBR 설계 방법.....	21

4.3.3 ARL 을 이용한 DBR 설계 결과.....	23
4.3.4 ARL 을 이용한 1×2 OPS 설계 방법.....	24
4.3.5 ARL 을 이용한 1×2 OPS 설계 결과.....	27
V. 인공 신경망.....	28
5.1 인공 신경망(Artificial Neural Network).....	28
5.1.1 인공 신경망의 구조와 학습.....	28
5.1.2 오버피팅(Overfitting).....	35
5.1.3 합성곱 신경망 (Convolutional Neural Network).....	35
5.1.4 ResNet.....	37
5.2 Inverse Design.....	38
5.3 인공 신경망을 이용한 Inverse Design.....	40
5.3.1 인공 신경망을 이용한 Inverse Design 방법.....	40
5.3.2 인공 신경망을 이용한 DBR Inverse Design.....	40
5.3.3 1×2 OPS Prediction 인공 신경망.....	43
VI. 결론.....	47
참고 문헌.....	49

그림 목차

그림 1.1 전자기파의 주파수 대역 스펙트럼	2
그림 1.2 광학 소자 설계 방법에 따른 실현 가능성과 스펙트럼 복잡성의 관계	3
그림 1.3 머신 러닝 분류	5
그림 1.4 FDTD 전산 모사를 위한 HPC 와 사양	6
그림 2.1 다층 구조에서의 빛의 전파	7
그림 2.2 $N+1/2$ pairs 의 주기적인 DBR 구조	10
그림 3.1 Y-branch 형태의 OPS	12
그림 3.2 머신 러닝 설계를 위해 변경된 OPS 구조	13
그림 4.1 퍼셉트론	15
그림 4.2 행렬 크기 (n) 변화에 따른 ARL 알고리즘 정확도	20
그림 4.3 ARL 을 이용한 DBR 설계 플랫폼	21
그림 4.4 ARL 을 이용한 DBR 설계 보상 정의 방법	22
그림 4.5 ARL 을 이용한 DBR 설계 학습 데이터 분포도	22

그림 4. 6 ARL 으로 설계한 DBR 과 이론 DBR 의 (a) 반사율 (b) 투과율 비교, 점선은 transmission 이 99%인 지점은 의미한다.....	23
그림 4. 7 ARL 으로 설계한 DBR 과 이론 DBR 의 두께 비교.....	24
그림 4. 8 ARL 을 이용한 1×2 OPS 설계 플랫폼	25
그림 4. 9 ARL 을 이용한 1×2 OPS 설계 학습 데이터 분포도.....	26
그림 4. 10 ARL 으로 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).....	27
그림 5. 1 인공 신경망 구조.....	28
그림 5. 2 Gradient descent method 개념도	29
그림 5. 3 부적합한 학습을 사용으로 인해 나타나는 문제점	30
그림 5. 4 Optimizer 성능 비교 [31].....	34
그림 5. 5 활성화 함수 (a) sigmoid (b) tanh (c) Relu	34
그림 5. 6 Regression 결과 (a) 학습이 잘 된 경우 (b) 언더피팅 (c) 오버피팅	35
그림 5. 7 CNN 개념도.....	36
그림 5. 8 Convolution layer 계산 과정	36
그림 5. 9 입력 데이터에 따른 pooling 결과 비교 [35]	37

그림 5. 10 인공 신경망 layer 의 연결 방법 (a) 일반 layer 연결 (b) residual block.....	38
그림 5. 11 DBR prediction 인공 신경망 학습 결과.....	41
그림 5. 12 ANN 으로 설계한 DBR 과 이론 DBR 의 (a) 반사율 (b) 투과율 비교, 점선은 transmission 이 99%인 지점은 의미한다.....	42
그림 5. 13 ANN 으로 설계한 DBR 과 이론 DBR 의 두께 비교	42
그림 5. 14 CNN 을 이용한 OPS prediction 인공 신경망 학습 테스트 [15]..	43
그림 5. 15 OPS prediction 인공 신경망 학습 데이터 분포도.....	44
그림 5. 16 OPS prediction 인공 신경망 학습 결과 (a) FCDNN (b) ResNet	45
그림 5. 17 OPS prediction 인공 신경망 학습 테스트 (a) FCDNN (b) ResNet	45

표 목차

표 4.1 ARL 으로 설계한 DBR 과 이론 DBR 의 반사율 비교.....	23
표 5.1 ANN 으로 설계한 DBR 과 이론 DBR 의 반사율 비교.....	42

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 meV 의 물리량을 가진다. 테라헤르츠파는 전자공학과 광공학의 경계에 위치하기 때문에 융합형 과학기술로의 발전 가능성이 크다.

테라헤르츠파의 가장 큰 장점은 금속을 제외한 유전체 물질등에 대한 투과율이 높으면서도 X-선과 같은 높은 에너지를 가져 이온화 작용을 하는 복사선이 아니므로 인체를 비롯한 생체 샘플에 해를 주지 않는다는 점이다. 또한, 테라헤르츠파의 에너지는 분자들의 비틀림, 회전, 진동 에너지와 상호작용하여 공명을 일으킴으로 인해 물질 고유한 흡수 스펙트럼이 나타나게 한다. 이를 이용하면 미지의 물질 성분에 대한 분자 식별도 가능한 것으로 알려져 있다 [1-3]. 두번째로 테라헤르츠파는 최근 ‘beyond 5G’ 시대에 적합한 주파수 대역으로 주목받고 있다. Edholm’s law 에서 “필요한 데이터 전송량은 18 개월마다 2 배로 증가한다”라고 예측하고 있으며 [4], 실제 무선 통신의 데이터 전송량은 기하급수적으로 증가하고 있다 [5]. 이에 테라헤르츠 대역이 실현 가능성과 적합성이 가장 높다고 평가되고 있다 [6,7]. 이러한 장점을 바탕으로 소재물성, 분자광학, 생명과학 및 생화학, 약품과학, 의료영상 등의 기초과학에서부터 대기관측, 가스감지, 무선통신, 식품검사 및 암세포 진단 등 다양한 분야에서 그 중요성이 나날이 증대되고 있다.

이러한 전자공학과 광공학의 경계에 위치해 있는 테라헤르츠파를 제어하기 위해서는 Photonic Integrated Circuit (PIC)가 필수적이다. PIC 는 Electronic integrated circuit 과 유사한 기능을 수행하면서 넓은 대역을 가지고, 전파 방해에 영향을 받지 않고, 줄 효과 (Joule effect)가 존재하지 않는다는 장점이 있다 [8]. 이러한 PIC 소자 중 광전력 분배기는 빛을 분배하는 기능을 하기 때문에 가장 기본적인면서도 필수적인 소자이다. 또한 여러 유전체 층이 쌓인 구조의 DBR 은 빛의 경로 변경을 위한 거울 역할 뿐만 아니라 레이저 공진기에도 사용될 수 있는 중요하고, 간단한 PIC 소자라고 할 수 있다.

본 논문에서는 머신 러닝을 이용하여 1 테라헤르츠 (1 THz)에서 동작하는 DBR 과 1×2 광전력 분배기를 설계하였다. 먼저, DBR 과 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 DBR 과 1×2 광전력 분배기를 설계하고, 딥 러닝에 속하는 인공 신경망 방법을 이용하여 DBR 설계와 1×2 광전력 분배기의 투과율 예측 인공 신경망의 정확도를 증가시켰다.

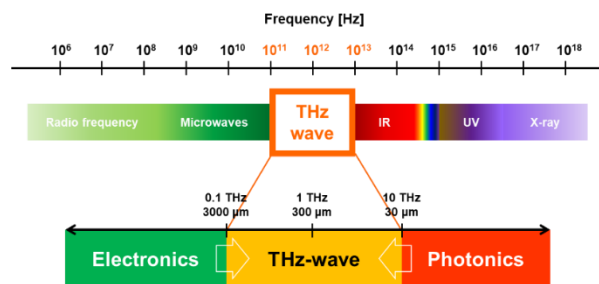


그림 1.1 전자기파의 주파수 대역 스펙트럼

1.2 머신 러닝

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) [9,10] 방법이나, 기본 구조에서 길이나 광학 상수들을 변수로 하여 각각의 변수에 해당하는 모든 소자의 성능을 계산하고 비교하여 최적의 구조를 찾는 방법 [11,12], 일정 주기로 반복되는 패턴을 이용한 자기 이미징 (Self-Imaging) 방법 [13,14] 등으로 설계를 하였다. 그러나 이 방법들은 요구하는 계산 장비 성능 높고, 계산 시간도 오래 걸린다. 또한 이러한 방식으로 설계된 구조들은 실현 가능성 (Design feasibility)과 스펙트럼 복잡성 (Spectral complexity)이 반비례하기 때문에 원하는 스펙트럼을 가지면서도 실현 가능한 구조를 찾기 어렵다 [15]. 머신 러닝 (Machine Learning)으로 광학 소자를 설계한다면 이러한 한계들을 극복할 수 있다.

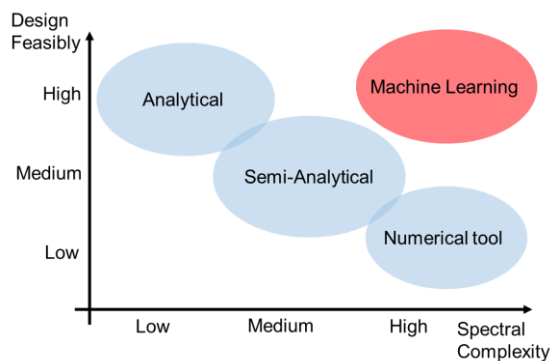


그림 1.2 광학 소자 설계 방법에 따른 실현 가능성과 스펙트럼 복잡성의 관계

머신 러닝은 인공 지능 (Artificial Intelligence)의 한 분야로 어떠한 문제 (tasks)와 결과 (performance measure)로부터 데이터 (experience)를 얻어 학습하는 프로그램을 말한다 [16]. 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 classification 이 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 reduction 이 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.

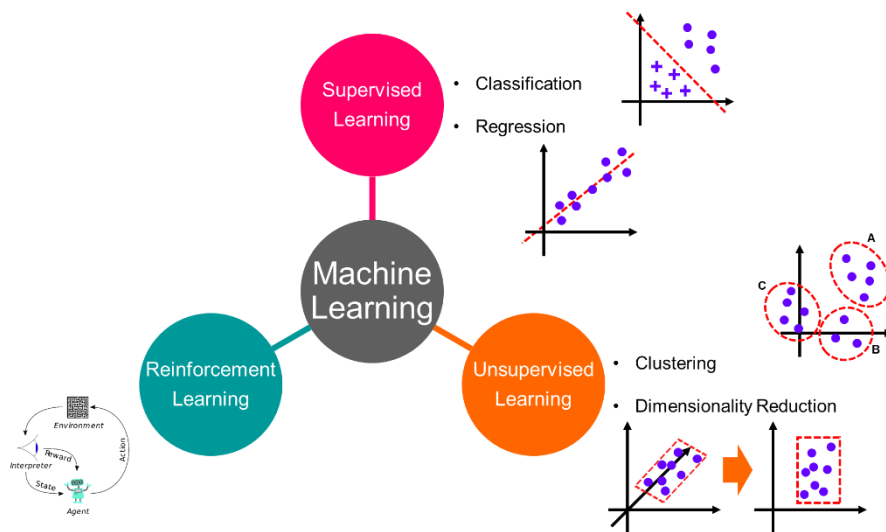


그림 1.3 머신 러닝 분류

1.3 머신 러닝 도구 및 전산 자원

본 연구에서 Anaconda[17] 가상 환경에서 프로그래밍 언어 파이썬 (Python)으로 머신 러닝을 사용하였다. 인공 신경망을 위해서 구글의 오픈 소스 (open source)인 Tensorflow[18]를 사용하였다. 전산 자원은 다음과 같다.



Name	Workstation-NBTP
CPU	Intel Xeon Scalable Silver 4114 (2.2 GHz)
CPU # (core # / thread #)	2 (20 / 40)
RAM	32 GB (DDR4)
RAM (Total)	384 GB
FDTD Software	FDTD Solutions (Lumerical Inc.)

그림 1.4 FDTD 전산 모사를 위한 HPC 와 사양

II. 분산 브레그 반사경

2.1 다층 구조에서의 전달 행렬(Transfer Matrix)

DBR 과 같은 다층 구조에서 빛의 전파 특성은 전달 행렬을 사용하여 쉽게 계산할 수 있다 [19]. 그림 2.1 과 같이 매질이 x 방향에 대해 균일하지 않다면, 유전율(permittivity, ϵ)과 투자율(permeability, μ)을 구간에 대해 나누어 생각할 수 있다.

$$\begin{cases} \mu_l = \mu(x_l) \\ \varepsilon_l = \varepsilon(x_l) \end{cases} \text{ for } -d_{l-1} \geq x \geq -d_l$$

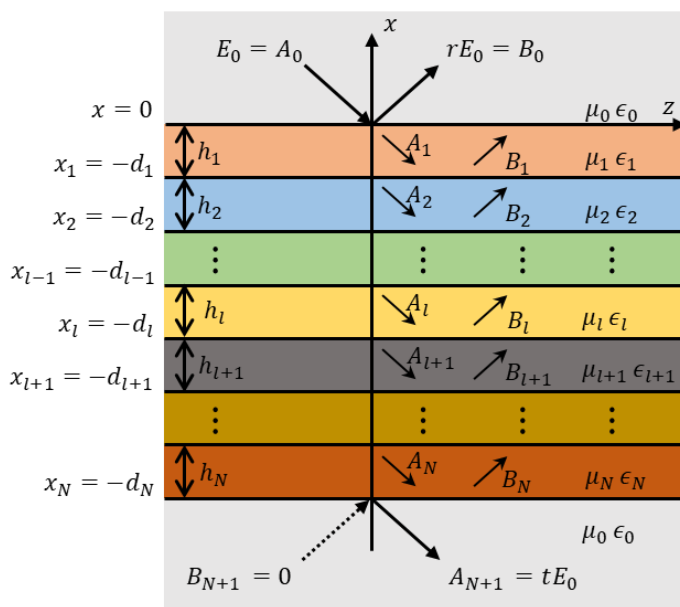


그림 2.1 다층 구조에서의 빛의 전파

입사하는 빛과 반사하는 빛이 모두 TE mode (Transverse Electric mode)라고 생각하면 각각의 전기장($\mathbf{E}_i, \mathbf{E}_r$)은 다음과 같이 나타낼 수 있다.

$$\mathbf{E}_i = \hat{y}E_0 e^{-ik_0x + ik_{0z}z}$$

$$\mathbf{E}_r = \hat{y}rE_0 e^{ik_0x + ik_{0z}z}$$

l 번째 매질 ($x_{l-1} \geq x \geq x_l$)에서의 전기장은 $\mathbf{E}_l = \hat{y}E_y^l$ 이며, 자세히 나타내면 다음과 같다.

$$E_y^l = (A_l e^{-ik_{lx}(x+d_l)} + B_l e^{+ik_{lx}(x+d_l)}) e^{ik_{lz}z}$$

$$H_z^l = \frac{1}{i\omega\mu_l} \frac{\partial}{\partial x} E_y^l = \frac{-k_{lx}}{\omega\mu_l} (A_l e^{-ik_{lx}(x+d_l)} - B_l e^{+ik_{lx}(x+d_l)}) e^{ik_{lz}z}$$

이때 $k_{lx} = \sqrt{\omega^2\mu_l\epsilon_l - k_{0z}^2}$, $k_{lz} = k_{0z}$ 이다. 전기장의 탄젠트(tangent) 성분 E_y 와 자기장의 탄젠트 성분 H_z 가 매질의 경계인 $x = -d_l$ 에서 연속이라는 경계 조건을 사용하면 다음 식을 얻을 수 있다.

$$A_l + B_l = A_{l+1} e^{-ik_{(l+1)x}(-d_l+d_{l+1})} + B_{l+1} e^{+ik_{(l+1)x}(-d_l+d_{l+1})}$$

$$A_l - B_l = \frac{\mu_l k_{(l+1)x}}{\mu_{l+1} k_{lx}} (A_{l+1} e^{-ik_{(l+1)x}(-d_l+d_{l+1})} - B_{l+1} e^{+ik_{(l+1)x}(-d_l+d_{l+1})})$$

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

$$P_{l(l+1)} = \frac{\mu_l k_{(l+1)x}}{\mu_{l+1} k_{lx}}$$

또한, $l+1$ 번째 매질의 두께를 다음과 같이 나타낼 수 있다.

$$h_{l+1} = d_{l+1} - d_l$$

위의 두 식을 정리하면 다음과 같이 전달 행렬을 나타낼 수 있다.

$$\begin{bmatrix} A_l \\ B_l \end{bmatrix} = \bar{\bar{\mathbf{B}}}_{l(l+1)} \begin{bmatrix} A_{l+1} \\ B_{l+1} \end{bmatrix}$$

$$\bar{\bar{\mathbf{B}}}_{l(l+1)} = \frac{1}{2} \begin{bmatrix} (1 + P_{l(l+1)})e^{-ik_{(l+1)}x h_{l+1}} & (1 - P_{l(l+1)})e^{ik_{(l+1)}x h_{l+1}} \\ (1 - P_{l(l+1)})e^{-ik_{(l+1)}x h_{l+1}} & (1 + P_{l(l+1)})e^{ik_{(l+1)}x h_{l+1}} \end{bmatrix}$$

여기서, $\bar{\bar{\mathbf{B}}}_{l(l+1)}$ 은 $l+1$ 번째 매질에서 l 번째 매질로 진행하는 빛을 계산하기 때문에 역방향 전파 행렬(backward-propagation matrix)라고 한다. 정방향 전파 행렬(forward-propagation matrix) $\bar{\bar{\mathbf{F}}}_{l(l+1)}$ 도 다음과 같이 구할 수 있다.

$$\begin{bmatrix} A_{l+1} \\ B_{l+1} \end{bmatrix} = \bar{\bar{\mathbf{F}}}_{(l+1)l} \begin{bmatrix} A_l \\ B_l \end{bmatrix}$$

$$\bar{\bar{\mathbf{F}}}_{(l+1)l} = \frac{1}{2} \begin{bmatrix} (1 + P_{(l+1)l})e^{ik_{(l+1)}x h_{l+1}} & (1 - P_{(l+1)l})e^{ik_{(l+1)}x h_{l+1}} \\ (1 - P_{(l+1)l})e^{-ik_{(l+1)}x h_{l+1}} & (1 + P_{(l+1)l})e^{-ik_{(l+1)}x h_{l+1}} \end{bmatrix}$$

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

$$\begin{bmatrix} E_0 \\ rE_0 \end{bmatrix} = \bar{\bar{\mathbf{B}}}_{01} \bar{\bar{\mathbf{B}}}_{12} \bar{\bar{\mathbf{B}}}_{23} \cdots \bar{\bar{\mathbf{B}}}_{N(N+1)} \begin{bmatrix} A_{N+1} \\ B_{N+1} \end{bmatrix}$$

$$= \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} tE_0 \\ 0 \end{bmatrix}$$

이로부터 다층 구조에 대한 투과 계수(transmission coefficient) t 와 반사 계수(reflection coefficient) r 를 다음과 같이 구할 수 있다.

$$t = \frac{1}{b_{11}}, \quad r = \frac{b_{21}}{b_{11}}$$

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 $1/4$ 이 되어야 한다. 전달 행렬을 $N+1/2$ pairs DBR 구조에 적용하면 다음과 같다.

$$\bar{\mathbf{B}}_{DBR} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \bar{\mathbf{B}}_{0h} (\bar{\mathbf{B}}_{hl} \bar{\mathbf{B}}_{lh})^N \bar{\mathbf{B}}_{hs}$$

여기서 아래 첨자 0 은 빛이 입사하는 매질, s 는 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.

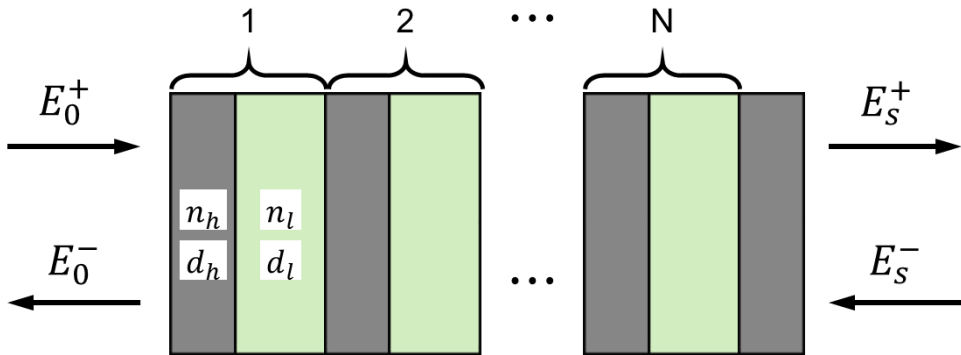


그림 2.2 $N+1/2$ pairs 의 주기적인 DBR 구조

2.3 DBR 전산 모사 도구

머신 러닝을 이용하여 DBR 을 설계하기 위해서는 무작위 다층 구조로 이루어진 학습 데이터가 필요하다. 이는 2.1 에서 서술한 전달 행렬을 계산하여 만들 수 있다. 전산 모사 소프트웨어로는 Python 에서 오픈 소스인 TMM(Transfer Matrix Method) package [20]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 나뉘어가지처럼 (Y-branch) 연결한 형태로, 하나의 입력된 광을 두 개의 똑같은 전력을 가진 광으로 분배하는 소자이다. 광통신 시스템에서 효율적인 광전송을 위해서는 광섬유 사이의 효율적인 결합이 필요하기 때문에 OPS 에 대한 연구가 많이 진행되고 있다 [21, 22].

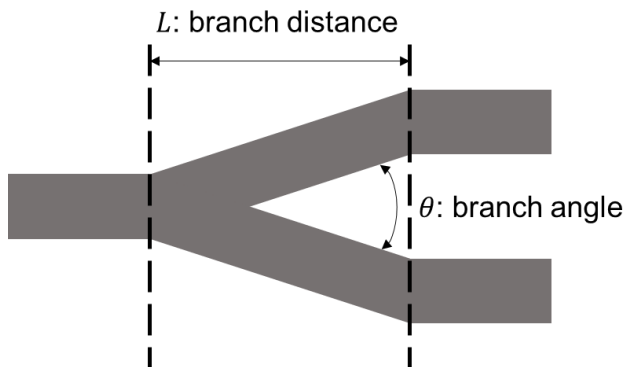


그림 3.1 Y-branch 형태의 OPS

기본 OPS 형태인 Y-branch 는 branch angle 이 작을수록 branch 입력에서 발생하는 분산 손실 (scattering loss)과 출력에서 발생하는 전환 손실 (conversion loss)을 줄일 수 있다 [23]. 따라서 branch distance 가 도파관 두께에 비해 길게 설계하게 되는데, 이 때문에 소자의 크기가 불필요하게 증가하게 된다. 머신 러닝을 이용하여 OPS 를 설계할

경우에는 그림 3.2 와 같은 구조를 이용하여 길이가 길어지는 문제를 해결할 수 있다.

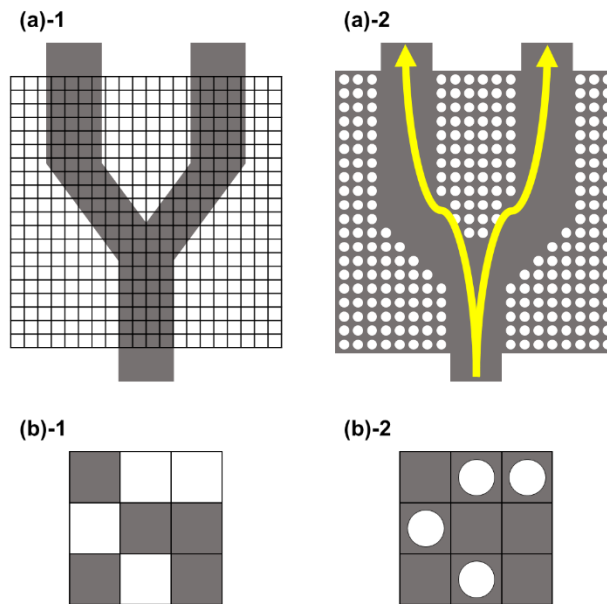


그림 3.2 머신 러닝 설계를 위해 변경된 OPS 구조

그림 3.2(a)-1 과 같이 기존 Y-branch 구조에서 설계 영역을 20×20 pixels 로 나누어 준다. 이때 설계 목표에 맞도록 pixels 의 매질을 결정하게 된다. 그림 3.2 (b)-1 처럼 정사각형의 pixel 을 그대로 사용하게 되면, 실제 공정에서 직각을 나타내기가 어려워 공정 효율이 떨어지게 된다. 따라서 그림 3.2 (b)-2 와 같이 정사각형 pixel 을 원형 pixel 로 바꾸어, 같은 효과를 보이면서도 공정 효율을 획기적으로 증가시킬 수 있다 [9].

1×2 OPS 의 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 OPS 의 성능 검사 지표는 두 출력의 전력 유사도가 아니라, 입력 대비 투과율과 전력 균일도 (power uniformity)가 된다. 이 때 투과율은 입력 대비이기 때문에, 1×2 OPS 의 이상적인 투과율은 3dB 이다.

3.2 1×2 OPS 전산 모사 도구

OPS 의 branch 를 계단 형태로 근사하여 이론적으로 해석할 수도 있지만, 본 논문에서 설계하는 OPS 는 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 cell 로 나누어 Maxwell 방정식으로 근사없이 정확하게 계산하는 도구이다. 따라서 복잡한 구조에서도 높은 정확도를 보인다. 전산 모사 소프트웨어는 Lumerical 사의 상용 소프트웨어인 FDTD Solutions [24]를 사용하였다.

IV. 가법 강화 학습

4.1 퍼셉트론(Perceptron)

퍼셉트론은 1957 년 Frank Rosenblatt 에 의해 제안된 초기 형태의 인공 신경망이다 [25]. 다수의 입력으로부터 하나의 결과를 내보내는 알고리즘으로 실제 뇌를 구성하는 뉴런 (neuron)과 유사한 동작을 한다.

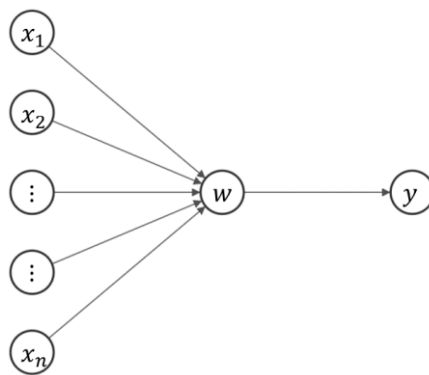


그림 4.1 퍼셉트론

그림 4.1 에서 x 는 입력값, w 는 가중치(weight), y 는 출력값을 의미한다. x 에 입력이 보내지면 w 와 곱해진다 ($x \cdot w$). 곱해진 값이 바로 y 가 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 1 을 출력하고, 미만이면 0 을 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 [26]. 강화 학습 방법으로 문제를 해결하기 위해서는, 그 문제를 Markov Decision Process (MDP)에 맞추어 정의해야한다. MDP 는 다음과 같이 정의된다.

Definition 4.1 MDP 는 5 개의 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 s_0 에서 초기 행동 a_0 를 수행하면 주어진 $T_{s_0a_0}$ 에 따라 다음 상태 s_1 이 확률적으로 결정된다. 그 결과로 보상 $R(s_0, a_0)$ 를 얻게 된다. 이 과정을 s_1 에도 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.

$$R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \dots$$

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

$$E[R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \dots]$$

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 [27]. 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.

ARL 은 학습 단계 (Training phase)와 추론 단계 (Inference phase) 로 나뉘어 진다 [28]. 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, \mathcal{E})에 누적 시킨다. 보상 총합 행렬은 0 으로 초기화 되어있는 상태이다.

$$\mathcal{E}_{n \times n} = \begin{pmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{pmatrix}$$

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

Algorithm 4.1 ARL: Training Phase

```

1: for all  $i \in \{1, \dots, n * N\}$  do
2:     Randomly create  $n \times n$  binary guess matrix  $B$ 
3:     Get  $r = f(A, B)$ 
4:      $\mathcal{E} \leftarrow \mathcal{E} + r * B$ 
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, DBR 을 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, \mathcal{E} 로부터 최종 결과물을 추론한다. \mathcal{E} 는 구조들의 보상들을 모두 누적시킨 상태이기 때문에 원소들의 값이 매우 크기 때문에, 원소들의 평균을

구하는 연산과정에서 자료형의 범위를 넘어갈 수 있다. 따라서 ε 의 최소값을 구하여 모든 원소에서 빼주는 정규화 과정을 거친다..

$$m = \min(\varepsilon)$$

$$\varepsilon := \varepsilon - m$$

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

$$F_{n \times n} = \begin{pmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{pmatrix}$$

그 다음 $\varepsilon_{i,j}$ 가 avg 보다 크거나 같으면 $F_{i,j}$ 에 1 을 작으면 0 을 대입해준다. 다음과 같이 의사 코드로 나타낼 수 있다.

Algorithm 4.2 ARL: Inference Phase

```

1: for all  $i \in \{1, \dots, n\}$  do
2:   for all  $j \in \{1, \dots, n\}$  do
3:     if  $\varepsilon_{i,j} \geq avg$  then
4:        $F_{i,j} = 1$ 
5:     else
6:        $F_{i,j} = 0$ 
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 [27]에서 $n \times n$ 이진 행렬을 찾는 문제로 테스트하였을 때, 10,000 개 데이터로 n 이 10 이하일 때는 90% 정확도로 답을 찾는 것이 가능했다. 하지만 11 부터는 정확도가 90% 미만으로 떨어지며 13 일 때는 정확도가 50%로 이하로 떨어졌다.

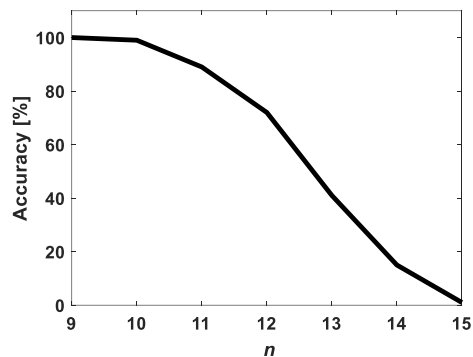


그림 4.2 행렬 크기 (n) 변화에 따른 ARL 알고리즘 정확도

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

4.3.2 ARL 을 이용한 DBR 설계 방법

DBR 의 설계 목표 주파수는 1 THz ($= 300 \mu\text{m}$) 이며, 실제 제작을 고려하여 공기와 fused quartz 를 사용하였다. 설계에서 고려한 파장 영역은 $5 \mu\text{m}$ 단위로 나누어진 $150 \sim 3000 \mu\text{m}$ 이다. 물질들의 굴절률은 테라헤르츠 대역에서 변화가 거의 없기 때문에 설계 목표 주파수인 1 THz 에서의 굴절률로 고정시켜, 공기의 굴절률은 1 ($n_l = 1$), fused quartz (SiO_2)의 굴절률은 2.092 [29]이다 ($n_h = 2.092$).

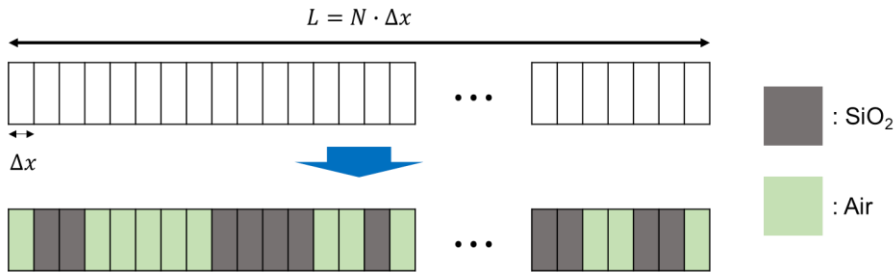


그림 4.3 ARL 을 이용한 DBR 설계 플랫폼

DBR 구조를 이진 행렬 형태로 표현하기 위하여 일정 두께 Δx 를 가지는 단위 층 (unit layer)들이 쌓여있는 구조를 생각한다. 이 때 이진 행렬의 값이 1 이면 quartz 의 굴절률을, 0 이면 공기의 굴절률을 대입한다. 이진 행렬의 길이가 N 이면 실제 DBR 전체 길이는 $N \cdot \Delta x$ 가 된다. 공진 조건을 고려하여 DBR 각층의 두께를 구하면 $d_h = 35.85 \mu\text{m}$, $d_l = 75 \mu\text{m}$ 이다. 한 pair 의 두께가 약 $100 \mu\text{m}$ 이고, $3.5 \sim 4.5$ pairs 정도의 두께를 고려하여 $L = 400 \mu\text{m}$ 로 설정하였다. Δx 가 너무 작으면 N 이 증가하기 때문에 ARL 에 필요한 데이터의 개수가 증가하고, 너무 크게 되면 정확한 층 두께를 구할 수 없게 된다. 따라서 공진 조건을 고려하여 구한 두께를 참고하여 $\Delta x = 5 \mu\text{m}$, $N = 80$ 으로 설정하였다. 설계에서 고려한 빛의 진행 mode 는 TE

mode 이다. 설계 주파수에서 대역폭을 정의하였을 때, 그림 4.4 에서 보이듯이 대역폭에 해당하는 반사율을 R_{in} , 해당하지 않는 반사율을 R_{out} 이라 하여 다음과 같이 보상 r 을 정의하였다.

$$r \equiv \text{mean}(R_{in}) \cdot (1 - \text{mean}(R_{out}))$$

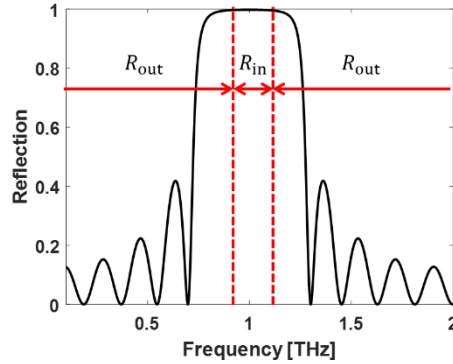


그림 4.4 ARL 을 이용한 DBR 설계 보상 정의 방법

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

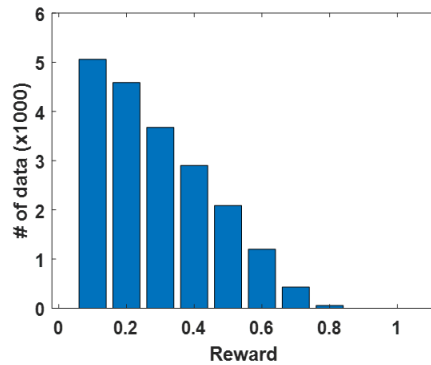


그림 4.5 ARL 을 이용한 DBR 설계 학습 데이터 분포도

4.3.3 ARL 을 이용한 DBR 설계 결과

그림 4.6 은 ARL 을 이용하여 설계한 DBR 과 공진 구조를 고려하여 설계한 DBR 의 반사율을 비교한 그래프이다. ARL 을 이용한 설계에서 전산 모사를 파장 단위로 계산하여 설계 파장을 300 μm , 대역폭을 100 μm 로 하여, 주파수 단위로 변환한 그래프에서는 양끝이 1.2 THz, 0.85 THz 가 되어 대역폭이 치우쳐져 있는 형태로 나타난다.

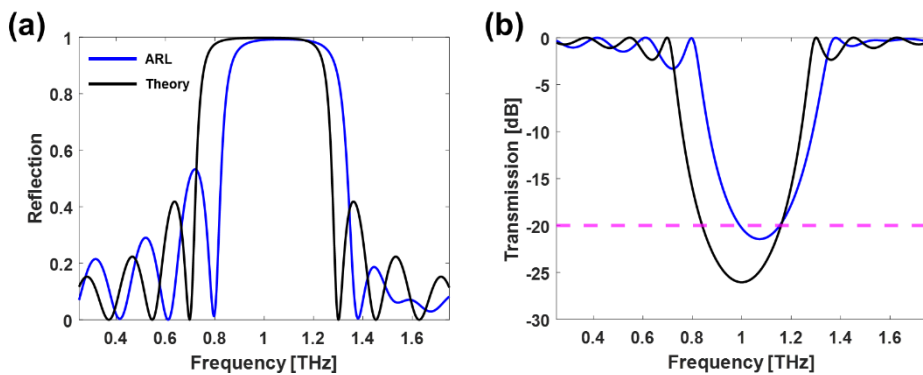


그림 4.6 ARL 으로 설계한 DBR 과 이론 DBR 의 (a) 반사율 (b) 투과율 비교, 점선은 transmission 이 99%인 지점은 의미한다.

	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)

표 4.1 ARL 으로 설계한 DBR 과 이론 DBR 의 반사율 비교

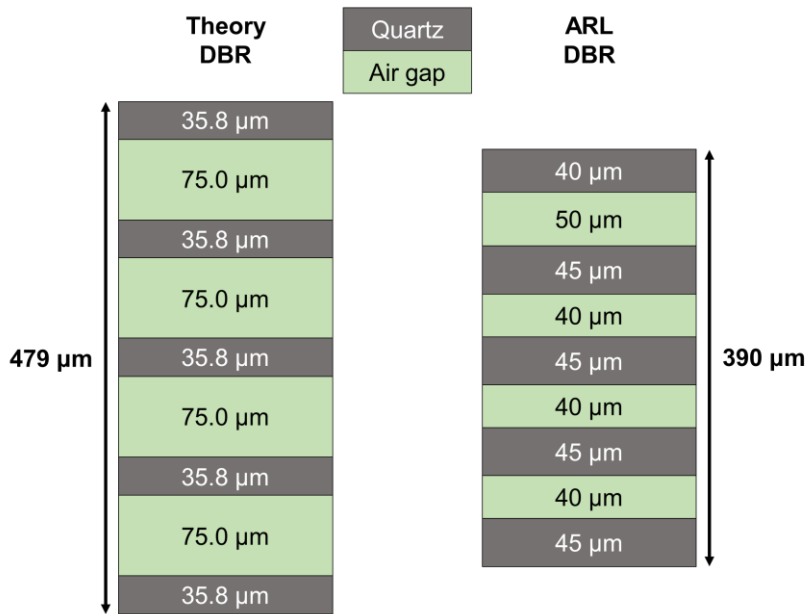


그림 4.7 ARL 으로 설계한 DBR 과 이론 DBR 의 두께 비교

ARL 을 이용하여 설계한 DBR 이 이론으로 설계한 DBR 보다 수치적으로 부족한 성능을 보인다. 하지만 설계 주파수 1 THz 에서의 반사율은 모두 99%를 넘으며, 같은 pair 이면서도 전체 길이가 공진 조건을 고려하여 설계한 DBR 보다 약 100 μm 짧기 때문에 공간 제약이 있는 환경에서 유용하게 사용될 수 있다.

4.3.4 ARL 을 이용한 1×2 OPS 설계 방법

1×2 OPS 의 설계 목표는 일정 대역폭의 투과율을 최대화시키는 것이 아니라, 원하는 광대역의 투과율을 최대화 시키는 것이다. 목표 주파수가 1 THz 이기 때문에 설계 대역폭을 275~325 μm 로 한다. 이를 2 μm 단위로 나누어 총 26 개 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 2 차원으로 나타내어진다. 물질들의 굴절률은 앞에서 말한 바와 같이, 테라헤르츠 대역에서 변화가 거의 없기 때문에 설계 목표 주파수인 1 THz 에서의 굴절률로 고정시켰다. 따라서 공기의 굴절률은 1 ($n_l = 1$)이며, 실리콘의 굴절률은 3.415 [30]이다 ($n_h = 3.415$).

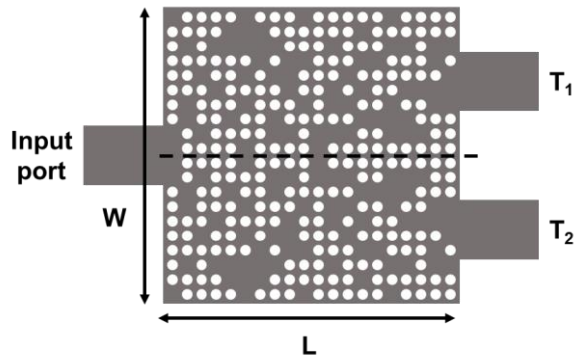


그림 4.8 ARL 을 이용한 1×2 OPS 설계 플랫폼

그림 4.7 은 ARL 을 이용한 1×2 OPS 설계 플랫폼을 나타낸 것이다. $500\ \mu\text{m} \times 500\ \mu\text{m}$ 정사각 실리콘 위에 $9\ \mu\text{m}$ 반지름의 원형 구멍을 식각하여 만든다. 입력과 출력 포트 (port)들은 테이퍼 (taper)가 없으며, 너비가 $100\ \mu\text{m}$ 인 무한한 길이의 직사각형으로 가정한다. 입력 포트는 정사각 실리콘 한 변의 중앙에 위치하며, 출력 포트는 입력 포트 반대편 변에 중앙으로부터 각각 $125\ \mu\text{m}$ 거리에 위치해 있다. OPS 는 대칭 구조이기 때문에 T_1 과 T_2 의 수치가 같다. 따라서 T 로 통일시켜 사용한다. 입력으로는 FDTD 에서 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, T_{\min} 은 최소 투과율 전체 중 최소, T_{\max} 는 최소 투과율 전체 중 최대를 의미한다.

$$r \equiv \min(T) - \frac{T_{\min} + T_{\max}}{2}$$

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

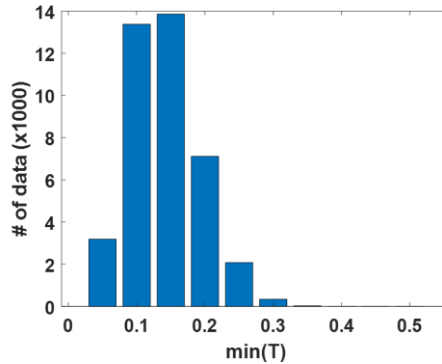


그림 4.9 ARL 을 이용한 1×2 OPS 설계 학습 데이터 분포도

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

4.3.5 ARL 을 이용한 1×2 OPS 설계 결과

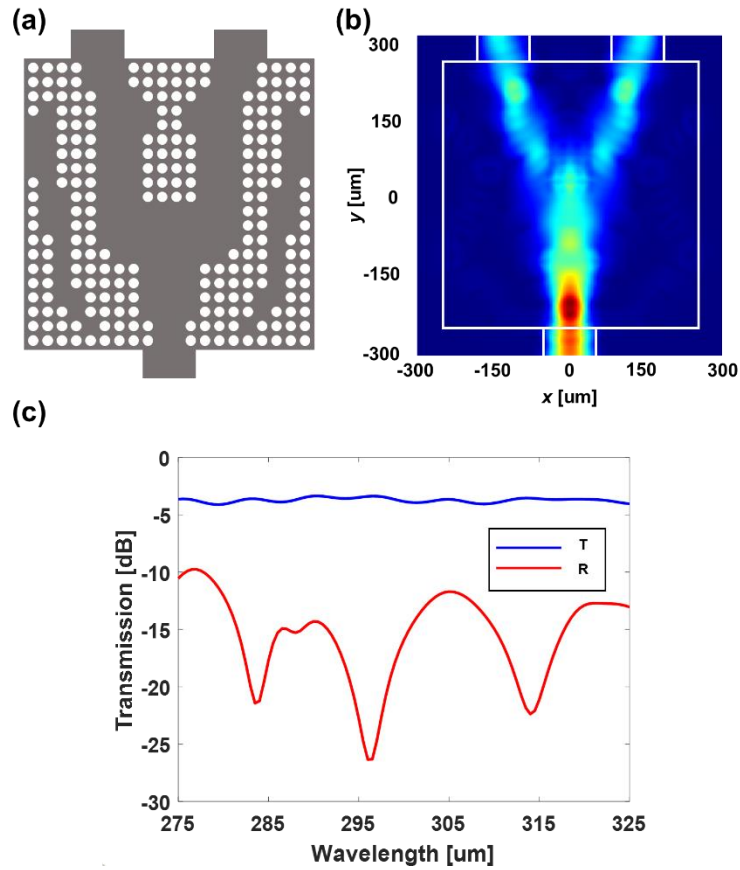


그림 4.10 ARL 으로 설계한 1×2 OPS (a) 식각 구조 (b) 파장이 $299 \mu\text{m}$ 일 때 power flow (c) 입력 포트 반사율 (R)과 출력 포트 투과율 (T)

ARL 을 이용하여 설계한 OPS 의 투과율은 평균 3.71 dB 로 4 dB 보다 적고, 균일도는 0.75 dB 이다. 다른 파장대역에서 설계된 기존 OPS 들보다 branch distance 가 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 regression 을 사용할 것이다. Regression 은 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.

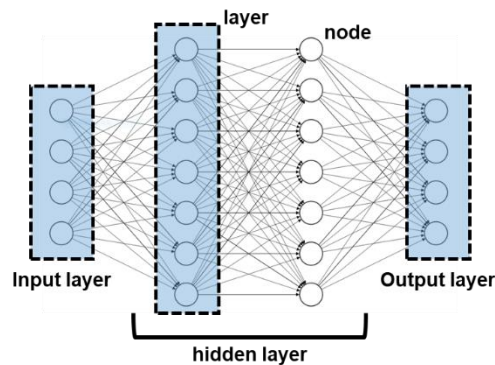


그림 5.1 인공 신경망 구조

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

$$s(x; \theta) = \sum_{j=1}^M w_j x_j + b$$

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 \hat{y} 가 출력된다. 이때 y 와 \hat{y} 의 차이, loss function 을 다음과 같이 정의한다.

$$\text{loss} \equiv \frac{1}{2} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

이 때 N 은 출력의 크기이다. 학습은 loss 를 최소화하는 가중치와 바이어스를 찾는 것을 의미한다. 이 과정은 gradient descent method [31]를 통해 이루어진다. Gradient descent method 는 함수의 기울기가 낮은 쪽으로 계속 이동하며 최솟값 (global minimum value)을 찾는 방법이다.

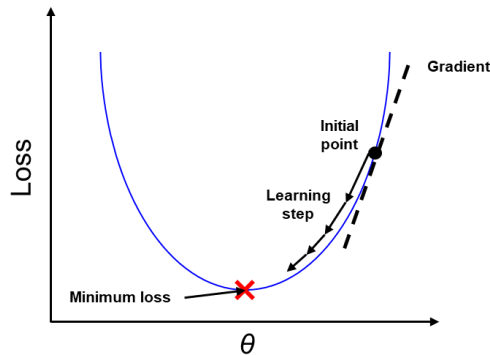


그림 5.2 Gradient descent method 개념도

인공 신경망에서 입력과 출력은 정해져 있기 때문에 loss 와 연관된 변수는 θ 뿐이다. 따라서, loss 와 θ 의 미분 관계를 구하고 이를

사용하여 θ 를 갱신하는 과정을 반복한다. 실제 loss 와 θ 의 관계를 직접적으로 구할 수 없으므로 연쇄 법칙 (chain rule)을 사용한다. 또한, θ 를 갱신하기 위해 반복하는 단계를 learning step 이라 한다.

$$\theta = \theta - r \frac{\partial L}{\partial \theta}$$

$$\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial y} \frac{\partial y}{\partial s} \frac{\partial s}{\partial \theta}$$

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 $1e-2 \sim 1e-3$ 사이의 값에서 시작하여 학습 속도와 가능 여부를 확인하며 조절한다.

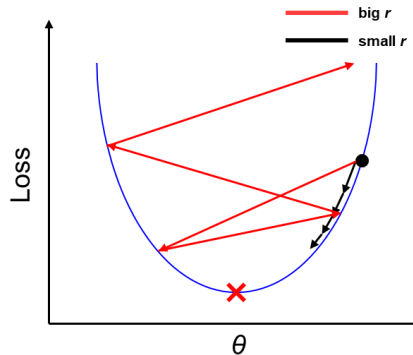


그림 5.3 부적합한 학습율 사용으로 인해 나타나는 문제점

Loss 가 0 이 되는 것은 거의 불가능 하기 때문에 허용 오차 (tolerance)를 사용하여 반복을 멈추거나, 최대 learning step 을 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: $w_j = w_j - r \frac{\partial L}{\partial w_j}$ // L is the loss function

5: **end for**

6: $b = b - r \frac{\partial L}{\partial b}$

7: **if** $L(X; w, b) < \tau$ **then**

8: **break**

9: **end if**

10: **end for**

11: **return** w, b

Algorithm 5.1 의 의사 코드는 labeled 데이터를 사용하여 loss function 을 계산할 수 있는 single layer 에서만 적용된다. 실제 hidden layer 가 존재하게

되면 hidden layer 사이에서는 labeled 데이터가 존재하지 않기 때문에 loss function 을 정의할 수 없는 문제가 발생한다. 이를 해결하기 위하여 고안된 것이 역전파 알고리즘 (backpropagation algorithm)이다. 역전파 알고리즘은 출력에서 계산되는 loss 를 hidden layer 의 매개 변수와 연결시키주는 방법이다. Loss 를 앞에서 뒤로 보낸다는 의미로 ‘역 (back)’을 사용한다. 먼저 N 을 인공 신경망을 구성하는 layer 의 수, $w_{ij}^{(n)}$, $b_i^{(n)}$, $y_i^{(n)}$ 은 n 번째 layer를 구성하는 i 번째 노드의 j 번째 가중치, 바이어스, 출력이다. 또한, n 번째 layer 의 입력은 $n - 1$ 번째 layer 의 출력과 같다.

$$x_j^{(n)} = y_j^{(n-1)}$$

가중치 갱신에 대한 gradient descent method 관계식은 다음과 같다.

$$w_{ij}^{(N)} = w_{ij}^{(N)} - r \frac{\partial L}{\partial w_{ij}^{(N)}}$$

그리고 loss 와 노드의 출력과의 관계를 나타내는 변수 $\delta_k^{(n)}$ 다음과 같이 정의한다.

$$\delta_k^{(n)} \equiv \frac{\partial L}{\partial s_k^{(n)}}$$

n 번째 layer 를 구성하는 노드의 개수를 K_n , 입력되는 데이터의 크기를 M_n 이라하면, hidden layer 중 output layer 와 가장 가까운 $N - 1$ layer 를 구성하는 i 번째 노드의 j 번째 가중치에 대한 갱신 관계식은 다음과 같다.

$$\begin{aligned}
\frac{\partial L}{\partial w_{ij}^{(N-1)}} &= \frac{\partial L}{\partial y_i^{(N-1)}} \frac{\partial y_i^{(N-1)}}{\partial s_i^{(N-1)}} \frac{\partial s_i^{(N-1)}}{\partial w_{ij}^{(L-1)}} \\
&= \left[\sum_{k=1}^{K_N} \frac{\partial L}{\partial s_k^{(N)}} \frac{\partial s_k^{(N)}}{\partial y_i^{(N-1)}} \right] \frac{\partial y_i^{(N-1)}}{\partial s_i^{(N-1)}} \frac{\partial s_i^{(N-1)}}{\partial w_{ij}^{(L-1)}} \\
&= \left[\sum_{k=1}^{K_N} \delta_k^{(N)} w_{ki}^{(N)} \right] \frac{\partial y_i^{(N-1)}}{\partial s_i^{(N-1)}} \frac{\partial s_i^{(N-1)}}{\partial w_{ij}^{(L-1)}} \\
&= \left[\sum_{k=1}^{K_N} \delta_k^{(N)} w_{ki}^{(N)} \right] \frac{\partial y_i^{(N-1)}}{\partial s_i^{(N-1)}} y_j^{(N-2)}
\end{aligned}$$

이 식으로부터 $\delta_k^{(n)}$ 을 저장하여 학습할 때 사용한다면, $n - 1$ 번째 layer 와 loss 사이의 관계를 구할 수 있는 것을 알 수 있다. 따라서, output layer 에서 계산되는 loss 를 hidden layer 까지 전달하여 hidden layer 의 매개 변수와 loss 사이의 관계식을 구할 수 있게 된다.

인공 신경망 학습을 위해 사용하는 역전파 알고리즘 수행 함수를 optimizer 라고 한다. optimizer 들은 gradient descent method 을 시작으로 하여 여러 문제점들을 개선하며 개발되었다. 그 중에서 대표적으로 사용하는 optimizer 는 Adam optimizer [32]이다. 그림 5.4 와 같이 optimizer 의 성능을 이미지 인식 인공 신경망 학습으로 비교해보면 Adam 이 가장 빠르게 학습한다.

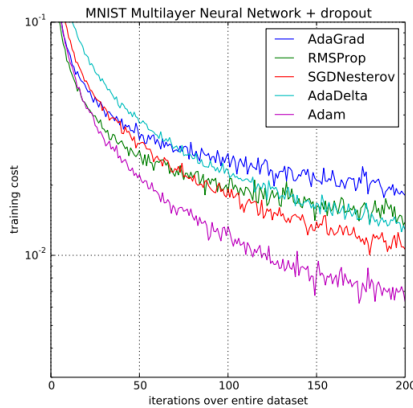


그림 5.4 Optimizer 성능 비교 [31]

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

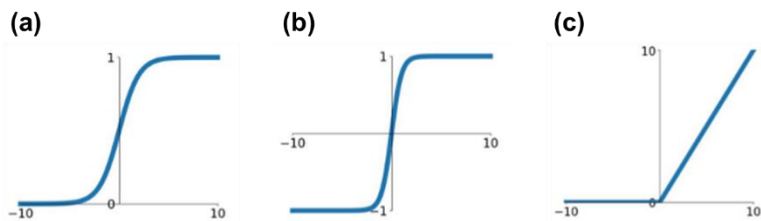


그림 5.5 활성화 함수 (a) sigmoid (b) tanh (c) Relu

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 loss 만 봤을 때는 학습이 제대로 이루어진 것으로 보이나, 테스트 과정에서 test loss 가 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.

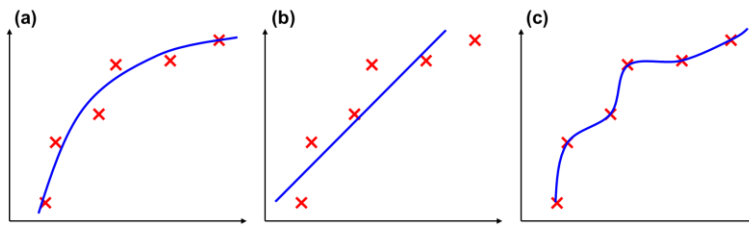


그림 5.6 Regression 결과 (a) 학습이 잘 된 경우 (b) 언더피팅 (c) 오버피팅

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) [33]를 사용한다. CNN 은 고양이의 시각 피질 (visual cortex) 반응에 대한 연구에서 출발하였다 [34]. 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 edge 들의 합성을 통해 인식한다는 것이다. 인공 신경망에서는 convolution layer 가 이런 역할을 수행한다.

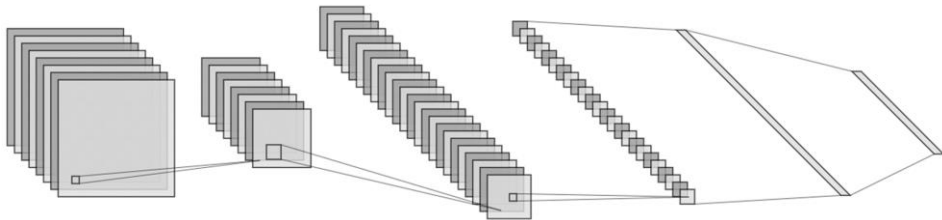


그림 5.7 CNN 개념도

Convolution layer 는 그림 5.8 과 같이 입력의 일정 부분을 filter 와 합성곱 (convolution) 연산을 통해 하나의 값으로 만들어 낸다. Filter 는 일정 간격으로 데이터위에서 움직이는데 이 간격을 stride 라 한다. 연산 결과인 feature map 은 input 보다 크기가 줄어들게 된다. 이를 방지 하기 위하여 feature map 의 외각을 0 으로 padding 을 한다.

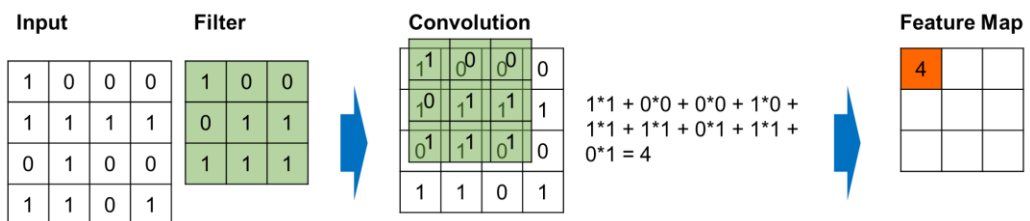


그림 5.8 Convolution layer 계산 과정

Feature map 을 그대로 사용하지 않고 크기를 줄이거나 특정 데이터를 강조하기 위한 과정을 거치는데, 이를 pooling 이라 한다. Pooling 방법에는 대표적으로 average pooling 과 max pooling 이 있다. Average

pooling 은 일정 영역의 해당하는 값들의 평균값을, max pooling 은 최대값을 사용하는 방법이다. 그림과 같이 값이 큰 데이터들 사이에서 작은 값의 데이터를 찾아야 하는 경우 average pooling 가 적합하고, 반대로 값이 작은 데이터들 사이에서 큰 값의 데이터를 찾아야 하는 경우 max pooling 가 적합하다 [35]. 일반적으로 max pooling 을 많이 사용한다.

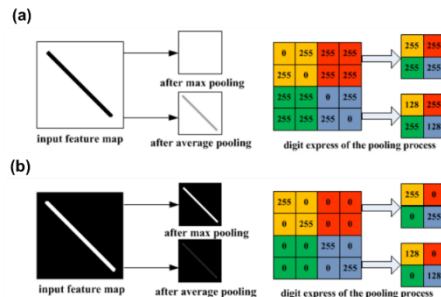


그림 5.9 입력 데이터에 따른 pooling 결과 비교 [35]

CNN 의 마지막 단에는 일반 flat 한 인공 신경망을 사용하였다. 이때는 hidden layer 를 1~2 개 정도 사용하여 output layer 와 연결한다. CNN에서 learning 이란, convolution layer 의 filter 와 마지막 단 hidden layer 의 가중치와 바이어스를 찾는 과정을 의미한다.

5.1.4 ResNet

일반적으로 layer 의 개수가 증가하면 연산하는 노드의 수가 증가하여 연산 자유도 또한 증가하기 때문에 해결할 수 있는 문제의 복잡도 (complexity)가 증가한다. Layer 의 개수가 많은 인공 신경망을 hidden layer 가 깊어진다고 (deep)는 의미로 deep neural network 라고 부른다. 하지만 layer 의 개수가 증가하면 loss 를 계산하는 output layer 에서 멀어질수록 loss 의 영향이 줄어들기 때문에 역전파 알고리즘을 이용한

학습이 어려워진다. 이미지 인식 문제를 해결하기 위해 개발된 CNN 에서도 학습율을 높이기 위해 layer 의 개수를 증가시키면서 비슷한 문제가 발생하였다. 이러한 문제를 해결하고 학습율을 높이기 위해 개발된 인공 신경망이 ResNet 이다 [36]. ResNet 이 일반 인공 신경망과 다른 점은 그림 5.10 과 같은 residual block 이 있다는 것이다. Residual block 에는 입력과 두 layer 이후를 연결해주는 skip connection 이 있어 loss 가 역전과 될 때 output layer 에서 멀어질수록 그 영향이 적어지는 정도를 줄일 수 있다.

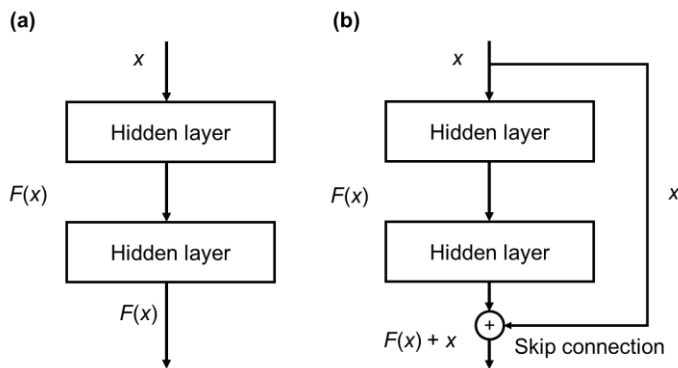


그림 5.10 인공 신경망 layer 의 연결 방법 (a) 일반 layer 연결 (b) residual block

5.2 Inverse Design

Inverse design 은 설계하고자 하는 소자의 목표 성능을 지정하고, 이를 만족시키도록 최적화된 소자의 모양을 찾아가는 수치해석법이다. 이전에는 inverse design 과 반대되는 개념인 forward design 으로 소자를 설계하였다. Forward design 은 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 design 은 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 [37].

Inverse design 는 크게 evolutionary 알고리즘과 gradient-based 알고리즘으로 나뉜다 [38]. 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 [39]. Particle swarming optimization 은 개체가 자기 방어나 음식물을 찾기 위해 군집으로 행동하는 사회적 행동 양식을 아이디어로 한다 [40]. 군집은 서로 상호작용하고 협동하기 때문에 목표 달성에 효과적인 점을 이용하였다. Gradient-based 알고리즘은 수치적으로 구조와 성능의 미분 관계를 구하여 최솟점을 찾는 방법으로 topology optimization [41]이 이에 속한다.

5.3 인공 신경망을 이용한 Inverse Design

5.3.1 인공 신경망을 이용한 Inverse Design 방법

인공 신경망을 이미 컴퓨터 비전 [33], 음성 인식 [42]등의 분야에서 좋은 성과들을 보이며 사용되고 있다. 최근 들어 광학 소자 설계에서도 Inverse design 의 문제점을 보완하고자 인공 신경망을 이용하여 설계하는 시도들이 증가하고 있다 [43-45].

기존 inverse design 인 evolutionary 알고리즘은 매우 많은 반복을 필요로 하기 때문에 시간이 오래 걸리는 문제점이 있으며, gradient-based 알고리즘은 gradient 를 numerical method 로 구하기 때문에 요구하는 연산 장비 성능이 높고, 시간이 오래 걸리는 문제점이 있다. 간단한 연산으로 이루어진 인공 신경망으로 구조의 optical response 계산을 대체하면 gradient 를 analytical method 로 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 4.3 의 ARL 을 이용한 DBR 설계와 같다. 인공 신경망은 100 개의 노드를 가지는 hidden layer 가 4 개로 구성되어 있는 구조이다. 총 20,000 개의 데이터에서 80%인 16,000 개의 데이터는 10 번 복제하여 160,000 개로 만들어 학습에 사용하였다. 20,000 개의 데이터에서 나머지 20%는 학습된 인공 신경망 테스트에 사용하였다. 한 step 당 한 개의 데이터를 학습하는 것이 아닌, 100 개의

데이터를 학습시켰으며, 이러한 양을 batch size 라 한다. Optimizer 는 학습을 1e-3 을 적용한 Adam 을 사용하였다. 인공 신경망 학습 결과는 그림 5.11 과 같다.

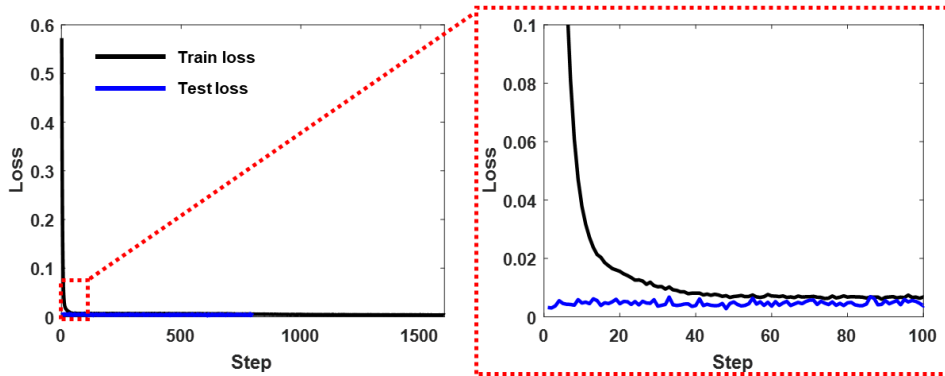


그림 5. 11 DBR prediction 인공 신경망 학습 결과

노드와 layer 의 개수를 변경하면 loss 가 변하기는 하나 그 크기가 매우 작기 때문에 비교하지 않았다. 학습은 총 1600 step 수행하였다. Train loss 는 100 step 이후 0.005 로 수렴하였으며, test loss 도 그와 비슷한 수준이었다. 인공 신경망이 DBR 의 optical response 계산이 가능하도록 학습되었다. Inverse design 에서 최대화하고자 하는 보상 r 을 다음과 같이 정의하였다.

$$r \equiv \text{mean}(R_{\text{in}}) \cdot (1 - \text{mean}(R_{\text{out}}))$$

Inverse design 에서 사용하는 optimizer 로도 학습을 1e-3 을 적용한 Adam 을 사용하였다. 그 결과는 다음과 같다.

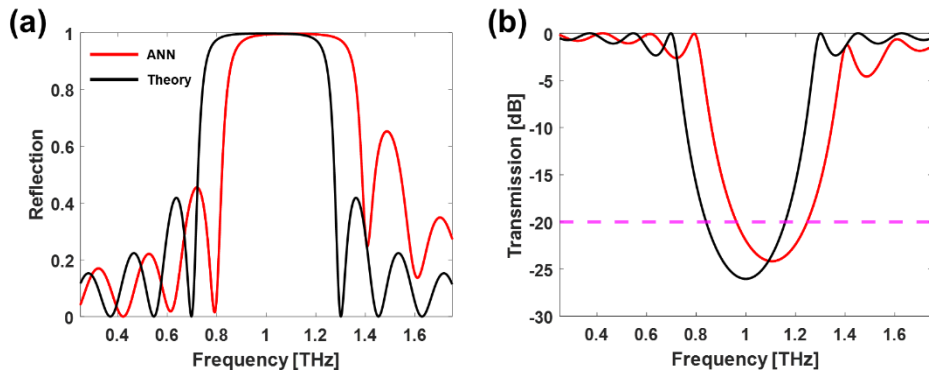


그림 5.12 ANN 으로 설계한 DBR 과 이론 DBR 의 (a) 반사율 (b) 투과율 비교,
점선은 transmission 이 99%인 지점은 의미한다.

	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 ANN 으로 설계한 DBR 과 이론 DBR 의 반사율 비교

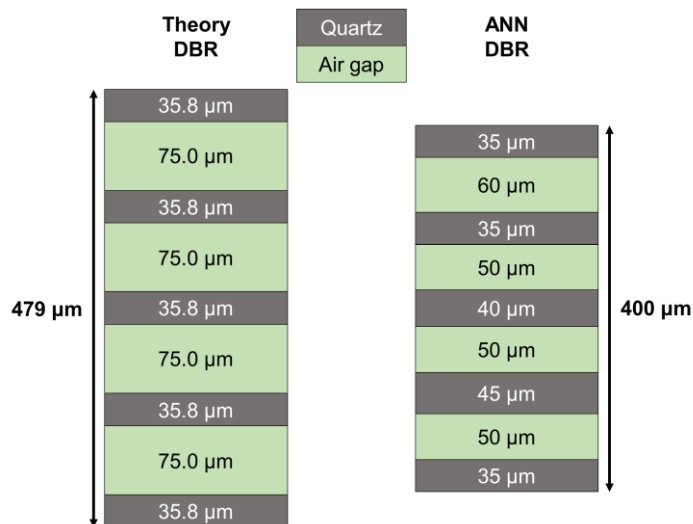


그림 5.13 ANN 으로 설계한 DBR 과 이론 DBR 의 두께 비교

ANN 으로 설계한 DBR 이 이론으로 설계한 DBR 보다 수치적으로 부족한 성능을 보인다. 하지만 설계 주파수 1 THz 에서의 반사율은 모두 99%를 넘으며, 같은 pair 이면서도 전체 길이가 공진 조건을 고려하여 설계한 DBR 보다 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 4.8 의 ARL 을 이용한 1×2 OPS 설계와 같다. 인공 신경망을 이용한 OPS 설계에서 현재 문제점은 OPS prediction 인공 신경망을 학습시키는 것이다. OPS 는 DBR 보다 구조와 해석이 복잡하여 필요로 하는 인공 신경망 학습 정확도가 높다. 기존에 있던 연구에서는 layer 간의 노드들을 모두 연결한 인공 신경망인 FCDNN (Full Connected Deep Neural Network)의 부족한 학습 정확도를 CNN 구조를 사용함으로써 보완하였다 [15].

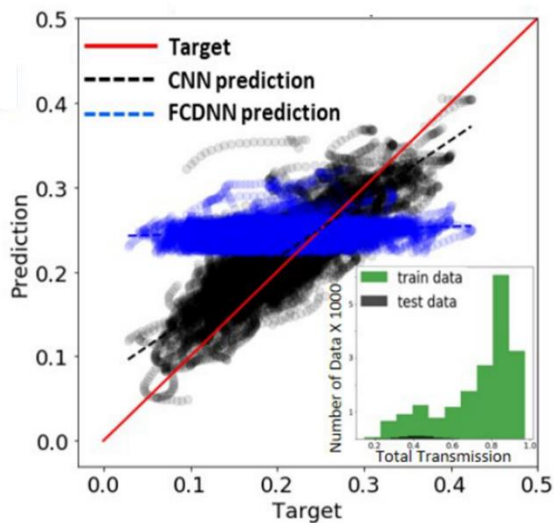


그림 5.14 CNN 을 이용한 OPS prediction 인공 신경망 학습 테스트 [15]

인공 신경망 구조를 바꾸어 상관계수를 0.16 에서 0.85 까지 증가시켰지만 inverse design 에 사용하기에는 부족한 정확도이다. 본 논문에서는 데이터 분포 개선과 인공 신경망 구조 변경을 병행하여 상관 계수를 상승시켰다. 학습 데이터를 평균 투과율에 대해 분포도를 나타내면 다음과 같다.

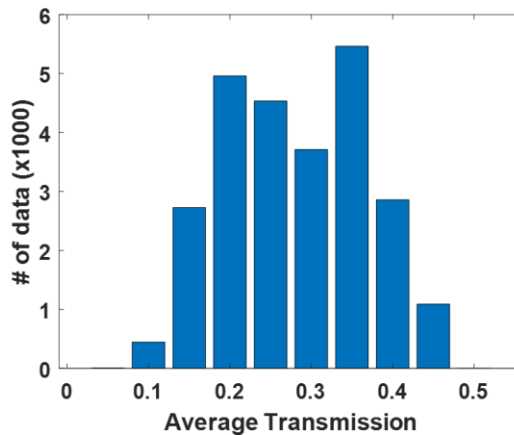


그림 5. 15 OPS prediction 인공 신경망 학습 데이터 분포도

인공 신경망은 FCDNN 과 ResNet 두 구조를 비교하며 사용하였다. FCDNN 은 100 개의 노드를 가지는 hidden layer 가 4 개가 있는 구조이고, ResNet 는 16 개의 residual block 으로 이루어져 있다. 무작위로 생성된 7,000 개의 데이터와 PSO 로 생성한 18,800 개의 데이터를 합하여 총 25,800 개의 데이터를 만들었다. 전체 데이터에서 90%인 23,220 개의 데이터는 5 번 복제하여 116,100 개로 만들어 학습에 사용하였다. 나머지 10%는 학습된 인공 신경망 테스트에 사용하였다. Batch size 는 100 으로 설정하고, Optimizer 는 학습율 $1e-3$ 을 적용한 Adam 을 사용하였다. 인공 신경망 학습 결과는 그림 5.16 과 같다.

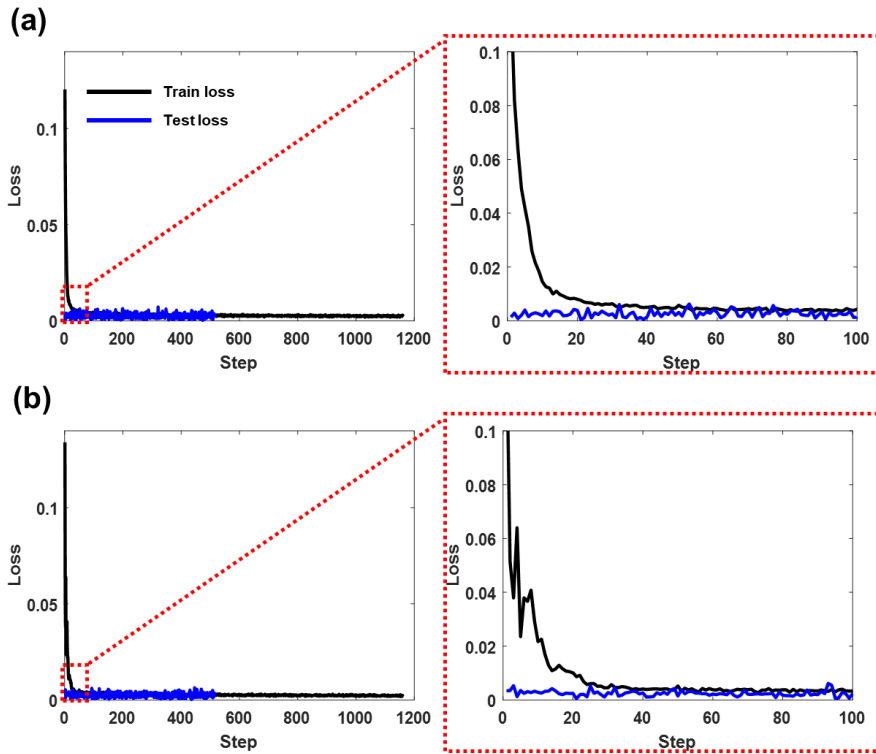


그림 5. 16 OPS prediction 인공 신경망 학습 결과 (a) FCDNN (b) ResNet

학습은 총 1161 step 수행하였다. Train loss 는 100 step 이후 FCDNN 의 경우 0.0026, ResNet 의 경우 0.0025 로 수렴하였으며, test loss 도 그와 비슷한 수준이었다. 구조 변경으로 인한 정확도 차이는 매우 작은 수준이다.

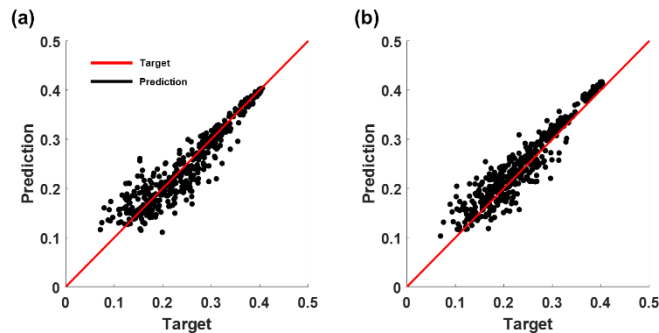


그림 5. 17 OPS prediction 인공 신경망 학습 테스트 (a) FCDNN (b) ResNet

두 구조는 상관 계수에서 차이를 보인다. FCDNN 의 상관 계수는 0.947, ResNet 의 상관 계수는 0.960 이다. 두 값 모두 앞에서 서술한 CNN 을 이용한 결과인 0.85 보다 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 loss 가 0.0025 수준으로 수렴하였다. 이 때 loss 를 mean square error 로 정의하였기 때문에 실제값과 prediction 의 차이는 약 0.05 이다. 이는 OPS 의 투과율 범위인 0 ~ 0.5 에서 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 DBR 과 1×2 OPS 를 설계하였다.

먼저, ARL 알고리즘을 이용하여 DBR 과 1×2 OPS 를 설계하였다. 20,000 개의 학습 데이터와 보상을 $\text{mean}(R_{\text{in}}) \cdot (1 - \text{mean}(R_{\text{out}}))$ 로 정의하여 DBR 을 설계하였다. 설계한 DBR 은 공진 조건을 고려한 DBR 과 비슷한 성능을 보이면서도 약 $100 \mu\text{m}$ 더 작은 (compact) 크기에서 동작하였다. 40,000 개의 학습 데이터와 보상을 $\min(T) - \frac{T_{\text{min}} + T_{\text{max}}}{2}$ 로 정의하여 1×2 OPS 를 설계하였다. 설계한 1×2 OPS 는 출력의 입력 대비 투과율이 3.71 dB, 균일도는 0.75dB 였다. 또한 기존 OPS 의 변환 손실을 줄이기 위해 branch distance 가 길어야 하는 문제점을 $500 \mu\text{m} \times 500 \mu\text{m}$ 에서 OPS 를 설계함으로써 해결할 수 있었다.

다음으로, ANN 과 inverse design 을 이용하여 DBR 과 1×2 OPS 를 설계하였다. ARL 에서 사용한 것과 같은 학습 데이터와 보상, 100 개의 노드로 이루어진 4 개의 hidden layer 로 구성되어 있는 FCDNN 을 이용하여 DBR 을 설계하였다. 설계한 DBR 은 공진 조건을 고려한 DBR 과

비슷한 성능을 보이면서도 약 100 μm 더 작은 크기에서 동작하였다. 1×2 OPS 는 인공 신경망의 prediction 정확도를 향상 시키는 것이 쟁점이다. 선행 연구에서는 인공 신경망의 구조를 FCDNN 에서 CNN 으로 바꾸어 상관 계수를 0.15 에서 0.85 로 상승시켰다. 본 논문에서는 우선적으로 무작위로 생성한 데이터와 PSO 로 생성한 데이터를 합하여 학습 데이터의 분포를 개선함으로써 FCDNN 구조에서 상관 계수를 0.94 로 상승시켰다. 또한 인공 신경망 구조를 skip connection 을 사용하는 ResNet 으로 바꾸어 상관 계수를 0.96 까지 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 DBR 과 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.

참고 문헌

1. P. C. Upadhyaya, Y. C. Shen, A. G. Davies, and E. H. Linfield, "Terahertz Time-Domain Spectroscopy of Glucose and Uric Acid." J. Bio. Phys, vol. 29, no.2-3, pp. 117-121, 2003.
2. P. F. Taday, I. V. Bradley, and D. D. Arnone, "Terahertz Pulse Spectroscopy of Biological Materials: L-Glutamic Acid." J. Bio. Phys, vol. 29, no. 2-3, pp.109-115, 2003.
3. T. R. Globus, D. L. Woodlard, T. Khromova, and A. C. Samuels, "THz-Spectroscopy of Biological Molecules." J. Bio. Phys., vol. 29, no. 2-3, pp.89-100, 2003.
4. S. Cherry, "Edholms law of bandwidth," IEEE Spectrum, vol. 41, no. 7, 2004, pp. 58-60.
5. F. Akyildiz, J. M. Jornet, and C. Han, "Terahertz band: Next frontier for wireless communications," Physical Communication, vol. 12, 2014, pp. 16-32.

6. J. Federici and L. Moeller, "Review of terahertz and subterahertz wireless communications," *Journal of Applied Physics*, vol. 107, no. 11, 2010, p. 111101.
7. 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.
8. E. Bor, O. Alparslan, M. Turduev, Y. S. Hanay, H. Kurt, S. I. Arakawa, and M. Murata, "Integrated silicon photonic device design by attractor selection mechanism based on artificial neural networks: optical coupler and asymmetric light transmitter," *Optics Express*, vol. 26, no. 22, 2018, p. 29032.
9. 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.
10. 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.

11. 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.
12. 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.
13. M. Bachmann, P. A. Besse, and H. Melchior, "General self-imaging properties in $N \times N$ multimode interference couplers including phase relations," *Applied Optics*, vol. 33, no. 18, 1994, p. 3905.
14. 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.
15. 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.
16. T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.

17. Anaconda Software Distribution. Computer software. Vers. 2-2.4.0. Anaconda, Nov. 2016. Web. <<https://anaconda.com>>.
18. 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.
19. S. L. Chuang, Physics of optoelectronic devices. New York, NY: Wiley, 1995.
20. Steven J. Byrnes, “Multilayer optical calculations,” arXiv preprint arXiv:1603.02720v3 [physics.comp-ph]

21. 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.
22. 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.
23. W. S. C. Chang, *Fundamentals of guided-wave optoelectronic devices*. Cambridge: Cambridge University Press, 2010.
24. Lumerical Inc. <https://www.lumerical.com/products/>
25. 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.
26. 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.

27. Çağrı Latifoğlu, “Binary Matrix Guessing Problem,” arXiv preprint arXiv:1701.06167v2, 2017.
28. 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.
29. 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.
30. E. D. Palik, *Handbook of optical constants of solids*. London: Academic Press, 1998.
31. D. P. Bertsekas, *Nonlinear programming*, 2nd ed. Athena Scientific, 1999.

32. Diederik P. Kingma, Jimmy Ba, “Adam: A Method for Stochastic Optimization,” arXiv:1412.6980v9 [cs.LG]
33. 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.
34. 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.
35. 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.
36. 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.

37. 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.
38. 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.
39. M. Mitchell, *An introduction to genetic algorithms*. Cambridge, MA: MIT, 1998.
40. Y. Shi and R. Eberhart, "Empirical study of particle swarm optimization," *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99* (Cat. No. 99TH8406).
41. J. Jensen and O. Sigmund, "Topology optimization for nano-photonics," *Laser & Photonics Reviews*, vol. 5, no. 2, 2010, pp. 308–321.
42. 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.

43. 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.
44. 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.
45. 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.

Acknowledgements

감사의 글

2 년이라는 짧은 시간이었지만 저에게는 정말 길게 느껴졌던 석사과정이 이렇게 마무리 되었습니다. 대학원에서 얻은 소중한 경험과 가르침들이 사회에 첫발을 딛는 저에게 정말 큰 힘이 되리라고 생각합니다. 제가 석사과정을 마칠 수 있도록 도움을 주신 많은 분들에게 감사의 마음을 전합니다.

먼저 학부생 시절부터 저를 대학원까지 지도해주신 한해욱 교수님께 감사의 말씀을 드립니다. 학부 졸업 시즌, 미래에 대해 방황할 때 진심으로 상담해주시고 대학원에서는 연구자로서의 자세를 지도해주신 덕분에 무사히 석사과정을 마무리 지을 수 있었습니다. 교수님께서 저에게 기대하신바에 미치지 못한 점 죄송하고 안타깝게 생각하고 있습니다. 사회에 나가서도 교수님의 가르침을 잊지않고 행동하겠습니다. 그리고 같이 과제 진행 중이신 공병돈 교수님과 멀리서 와주신 박기수 교수님께 바쁘신 와중에도 저의 석사논문 심사에 참석해주셔서 정말 감사드립니다.

적은 인원으로 적지 않은 일들을 처리하는 와중에도 항상 부족한 저를 감싸주셨던 연구실 멤버 모두에게도 정말 감사합니다. 정신적으로 흔들리고 방황할 때 가장 먼저 챙겨 주셨던 규석이 형. 가정에 항상 행복만 가득하고 아무 일 없이 내년에 무사히 졸업하시기를 바랍니다. 졸업하시기 전까지 연구실에 적응하지 못하는 저를 신경써주셨던 진하 형. 몸 건강하시고 하시는 일마다 잘 이루어지길 바랍니다. 그리고 한창

연구실 일 배우고 계신 한얼이 형. 항상 말하지만 기죽지 마시고
당당하게 생각하는 대로 행동하시고 좋은 결과 얻으시길 바랍니다.
그리고 같이 승리빌라에서 동고동락한 은식이 형과 지훈이 형. 모두
낮에는 연구실에서 일하고 퇴근하면 지쳐서 누워있느라 같이 많은 것을
하지는 못했지만, 주말 아침에 좀비같이 일어나는 모습들을 보면
어두컴컴한 기숙사에서 살지 않고 같이 자취를 같이 했기 때문에 제가
석사과정을 탈 없이 마무리 지을 수 있었다고 생각합니다. 또, 행운빌에서
살고 있는 통수쟁이들 기서형, 준희형, 진표. 가끔씩 같이 밥 먹거나
게임하면서 기분전환을 할 수 있어 고맙다는 말을 전하고 싶습니다.
끝으로 모두들 정말 고맙습니다.

제가 이렇게 아무 걱정 없이 공부할 수 있도록 도와주신
부모님께도 감사의 말씀드립니다. 학부 졸업식에서 석박사 졸업생들을
보며 우리 아들은 언제 저렇게 졸업하나 말씀하셨었는데 벌써 이렇게
졸업을 하게 되었습니다. 무소식이 희소식이라며 연락 자주 안하던
아들을 믿고 기다려 주신 덕분에 이렇게 졸업을 할 수 있었습니다.
어쩌면 저의 독단적일 수도 있는 결정임에도 불구하고 믿어주신 덕분에
제가 마음 편히 졸업 준비를 할 수 있었습니다. 이제는 행복한 모습만
보여드리면서 효도하는 아들로 살겠습니다.

모두 순간 순간이 행복하기를 바라며 글을 마치도록 하겠습니다.

Curriculum Vitae

이력서

이름: 이 종 건

학력

2014.03~2018.02: 포항공과대학교 전자전기공학과 (B.S.)

2018.03~2020.02: 포항공과대학교 대학원 전자전기공학과 (M.S.)

경력

2018.03~2020.02: 포항공과대학교 전자전기공학과 연구조교

국제 학회

1. 이종건, 정은아, 최혁재, 민병현, 임진하, 한해욱, “Study of Human Articular Cartilage for THz Biomedical Sensing,” 2018 IEEE-NANOMED, Hawaii
2. 이종건, 정은아, 최혁재, 민병현, 한해욱, “Bio-Sensing of Human Articular Cartilage using Terahertz Time Domain Spectroscopy,” 2019 MTSA, Busan
3. 이종건, 정은아, 이규석, 박기수, 한해욱, “Optical Properties of Crystalline α -Lactose Monohydrate in Terahertz Range,” 2019 MTSA, Busan
4. 이종건, 임미현, 정은아, 최혁재, 한해욱, “Study of Human Lymph Nodes using Terahertz Imaging,” 2019 IEEE-NANOMED, Gwangju