ONECLICK AI

PINN

연준모

목차

1. PINN

PINN

PINN 이전에, FEM 방식에 대해 생각해야 한다.

물리 공간을 아주 작은 격자(Mesh)로 쪼갠다.

그런데, 100m 크기의 안테나를 18GHz(파장 1.6cm)로 해석하려면?

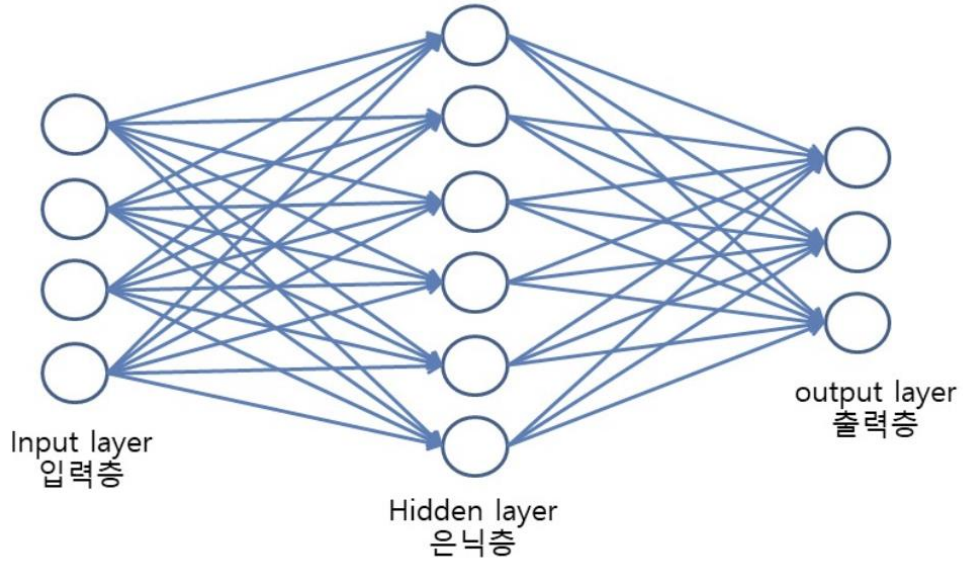
격자 간격이 파장의 1/10이어야 함 → 수십억 개의 격자 필요.

행렬의 크기가 메모리 한계를 초과
따라서 해석이 불가능 하다.

1. PINN

PINN

PINN은 MLP가 미분방정식(PDE)을 풀도록 강제하는 구조이다.



비지도 학습이다.

여기서, 은닉층이 많아진다고 생각하면 편하다

1. PINN

PINN

PINN은 크게 두 가지 경로(Path)로 나뉜다.

신경망 (Approximator) : 입력(x, t)을 받아 해(u)를 예측한다. (일반 MLP와 동일)

물리 정보 (Physics Constraints) : 예측된 u를 미분하여 물리 방정식(PDE)에 대입했을 때 0(잔차, Residual)이 되는지 확인한다.

신경망은 이미 알 고 있으므로, 넘어가자.

만약 기억나지 않는다면?

https://www.youtube.com/watch?v=ZLOM2Ft-HEs

1. PINN 1. 지배방정식

PINN

물리 정보

우리는 보고싶은 물리정보가 어떤 건지에 따라 지배방정식을 정해야 한다

방사 노이즈(Radiated Emission)를 볼 때: **헬름홀츠 방정식**
과도 응답(Transient)이나 ESD를 볼 때: **시간 영역 파동 방정식**
케이블/전도 노이즈(Conducted Emission)를 볼 때: **텔레그래퍼 방정식**

이번에는, 헬름홀츠 방정식을 기준으로 진행 해 보자

PINN

물리 정보

$$\nabla^2 u(\mathbf{x}) + k^2 u(\mathbf{x}) = f(\mathbf{x})$$

$\boldsymbol{u(\mathbf{x})}$ **(u):** 우리가 구하고 싶은 전자기파의 높이(세기)이다. 진폭 말하는 거다.
$\boldsymbol{\nabla^2}$ **(라플라시안,** $\boldsymbol{\nabla}$ **제곱):** 얼마나 꼬불꼬불한가를 뜻한다. 파동이 급격하게 변하는지, 완만하게 변하는지를 나타낸다.
k (파수): 얼마나 자주 진동하는가 이다. 높으면 고주파다.
$\boldsymbol{f(\mathbf{x})}$ **(소스 항, Source):** 파동을 만들어내는 원인이다. 호수에 돌을 던졌을 때, 돌이 떨어진 그 충격을 의미한다.

한 줄 요약: 파동의 꼬불거림($\nabla^2 u$)과 진동수($k^2 u$)를 더하면, 외부에서 가해진 힘(f)과 같아야 한다. 라는 물리 법칙 인 것 같다.

PINN

물리 정보

$f(\mathbf{x})$를 옆으로 넘기면, $\nabla^2 u(\mathbf{x}) + k^2 u(\mathbf{x}) - f(\mathbf{x}) = 0$이 된다.

이 식을 만족시키게끔 강제하기 위해 나중에 손실함수에 넣어버린다.

2. SIREN

SIREN

PINN 에서 사용하는 활성화 함수 이다.

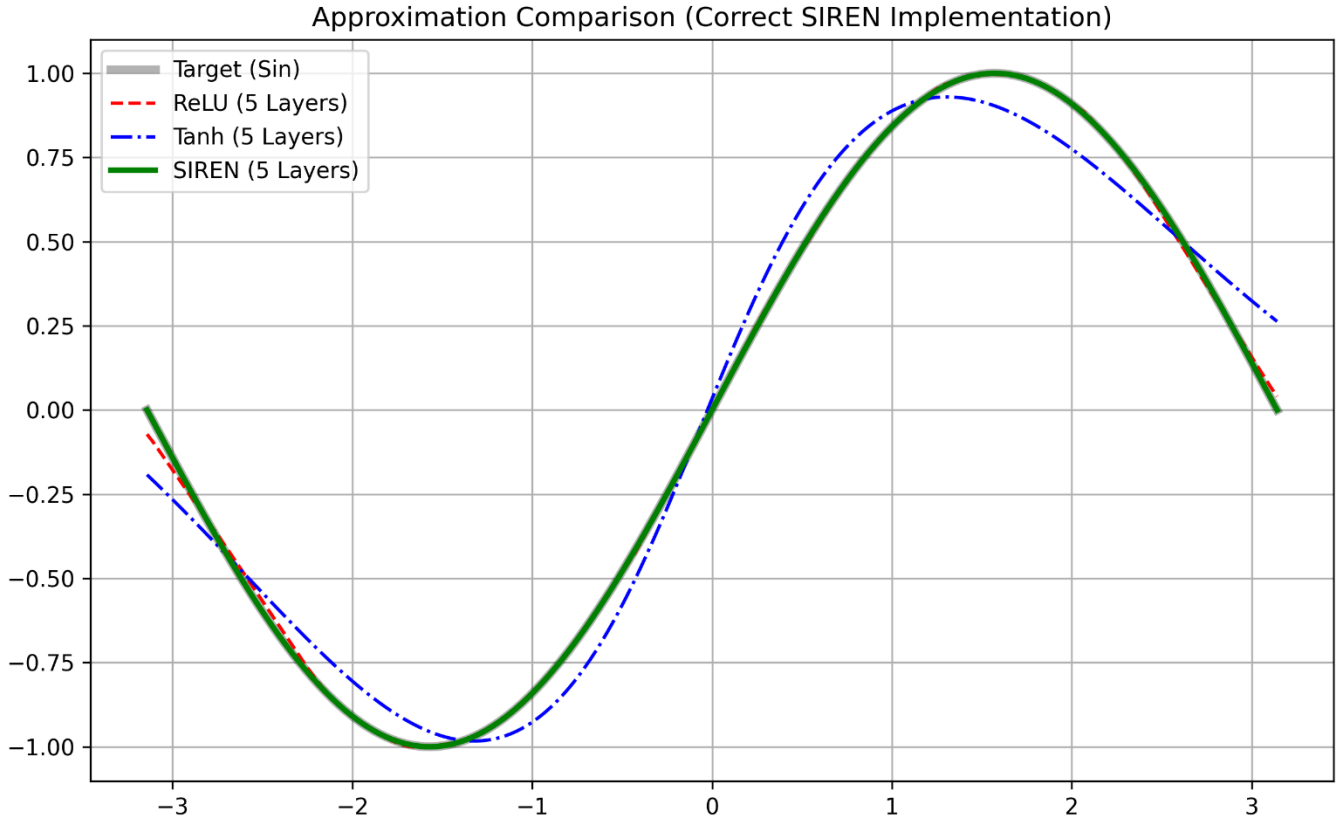
$$\phi(x) = \sin(\omega_0 x)$$

ω_0 (오메가 제로): 증폭 계수. 보톤 30 사용

AI가 정답을 그릴 때, 처음부터 꼬불꼬불한 사인 곡선을 가지고 그리기 시작하게 만드는 거다.
그러면 고주파 파동을 훨씬 빨리 배울 수 있다.

2. SIREN

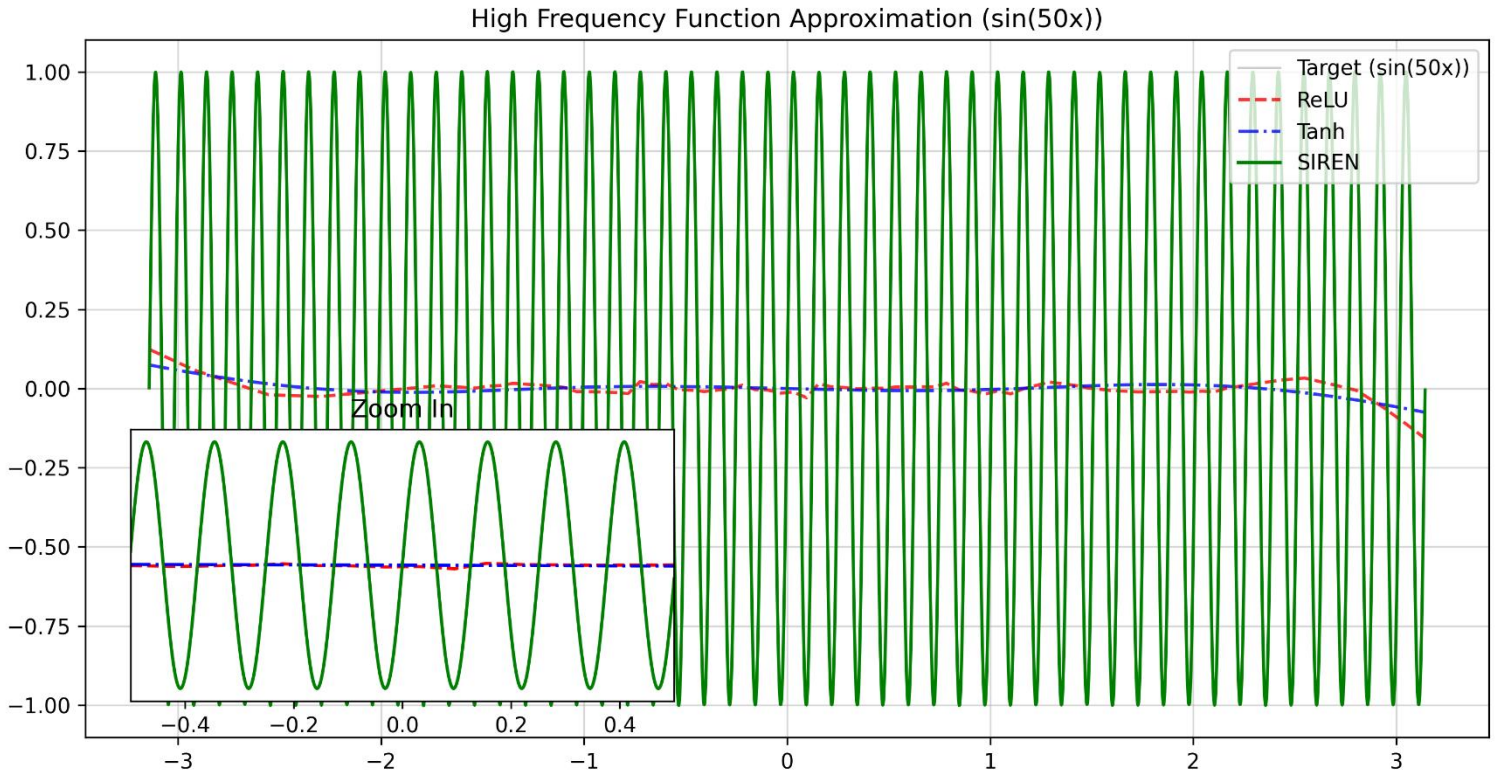
SIREN



한 주기인 저주파. Epoch 500

2. SIREN

SIREN



SIREN 말곤 전부 무너진다

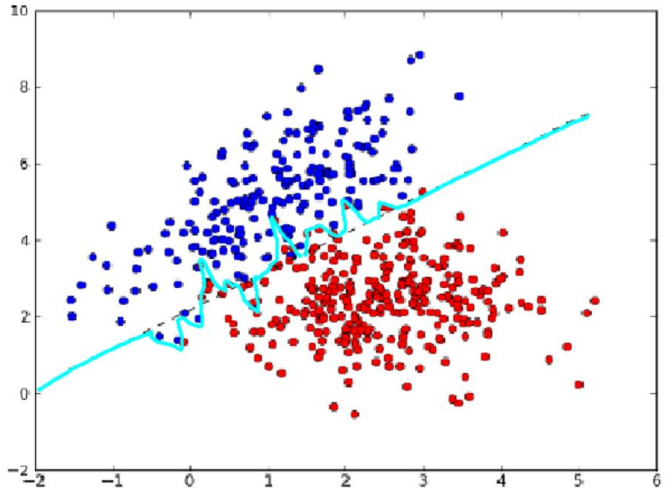
2. SIREN

SIREN

신경망은 **완만하고 부드러운 변화만 배우려고 하고, 급격하게 변하는 복잡한 패턴은 잘 안 배우려고 하는 성질이 있다.**

왜? 급격한 변화는 이상치라서 무시하고 가는게
일반적인 딥러닝(Tr, conv)에서 결과가 더 잘 나오니까

하지만, 고주파는 항상 급격하게 변화한다. 이거에 대응하기 위해, 강제로 30을 곱해서 넣는다.

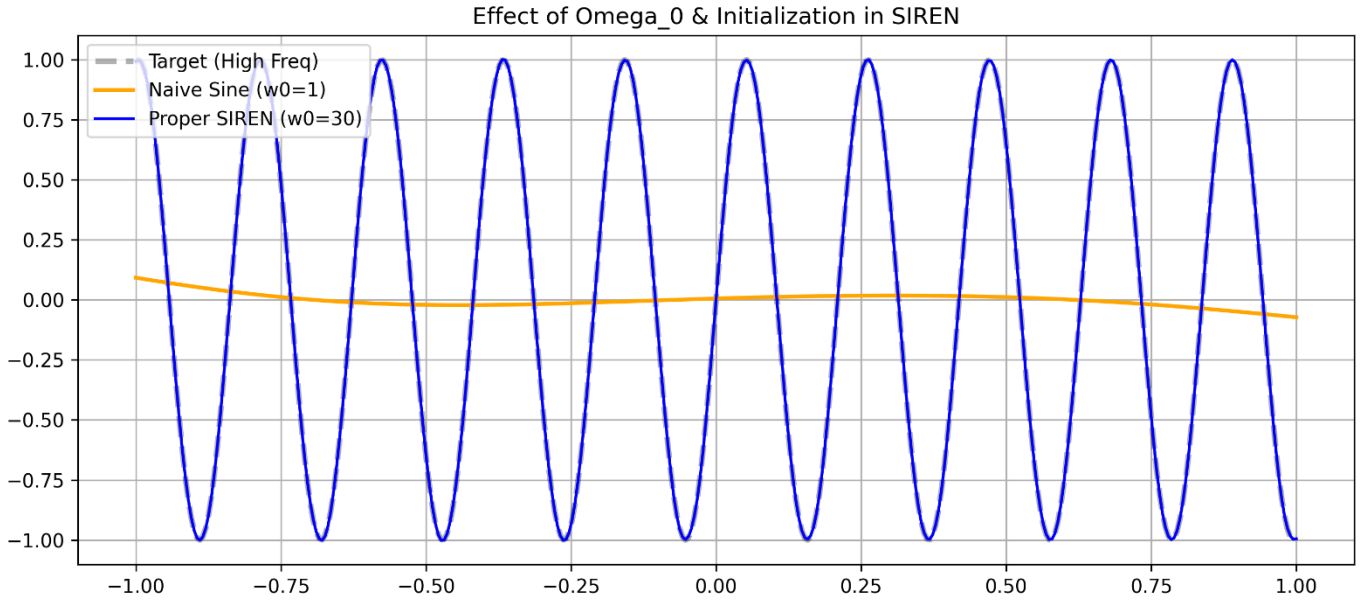


2. SIREN

SIREN

신경망은 **완만하고 부드러운 변화만 배우려고 하고, 급격하게 변하는 복잡한 패턴은 잘 안 배우려고 하는 성질이 있다.**

30이 곱해지면서 더 고주파에 대한 대응이 가능해 졌다.



2. SIREN

SIREN

가중치 초기화

SIREN은 모델 학습 시작 전 처음 가중치 초기화를 할 때, 다음식에 근거하여 초기화 하지 않으면 안된다.

그래야 고르게 분포되어 이쁜 결과가 나온다(Norm)

$$w \sim \mathcal{U}\left(-\frac{\sqrt{6/n}}{\omega_0}, \frac{\sqrt{6/n}}{\omega_0}\right)$$

2. SIREN

SIREN

가중치 초기화

$$W \sim \mathcal{U}\left(-\frac{\sqrt{6/n}}{\omega_0}, \frac{\sqrt{6/n}}{\omega_0}\right)$$

분자 ($\sqrt{6/n}$):

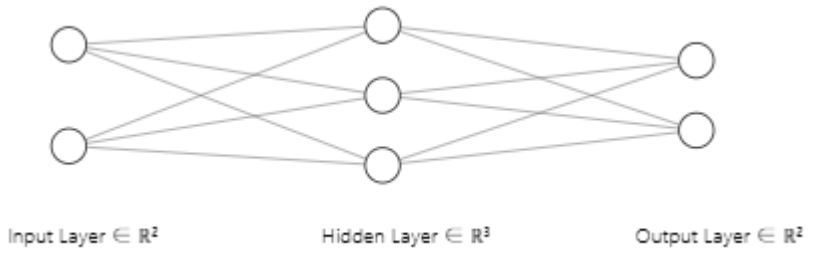
Xavier(Glorot) 초기화 기법이다.

n: 입력 노드의 개수.
입력 노드가 많으면(n이 크면), 그만큼 더해지는 값이 많아져서 결과값이 너무 커질 위험이 있다.
그래서 n이 클수록 가중치(W)를 작게 만들어야 한다.
그래야 입력이 통과한 후에도 신호의 분산(Variance)이 일정하게 유지된다.

2. SIREN

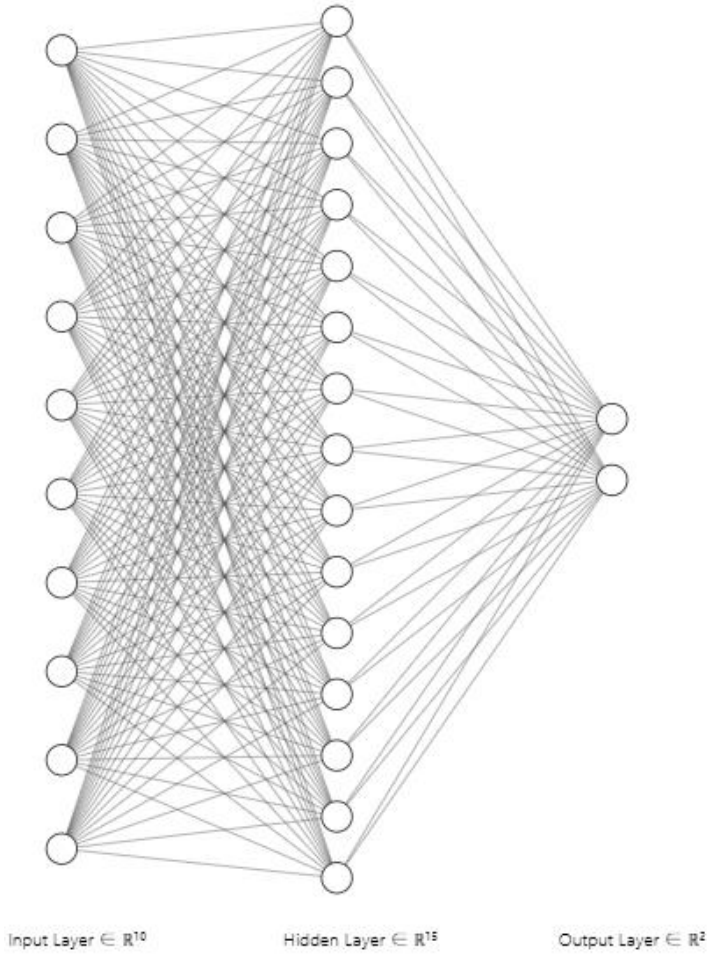
SIREN

왜 더해지는 값이 많아질까?



하나에 더해지는 값만 10개, 값이 너무 커져버린다.
Ttansformer 같은 데에선 이걸 add&Norm 을 사용했지만, 여기선 물리학적 이유로 못쓴다.

https://gihak111.github.io/ai/2025/11/29/SIREN_cant_use_Norm_upload.html



2. SIREN

SIREN

가중치 초기화

$$W \sim \mathcal{U}\left(-\frac{\sqrt{6/n}}{\omega_0}, \frac{\sqrt{6/n}}{\omega_0}\right)$$

분모 (ω_0):

SIREN의 활성화 함수는 $\sin(\omega_0 \cdot x)$이다.

여기서 ω_0(오메가 제로, 보통 30)는 **주파수 증폭 계수**이다

만약 가중치 W를 일반적인 크기로 두면, ω_0가 곱해지면서 값이 **30배 뻥튀기**가 된다

이렇게 되면 sin 함수 내부의 값이 너무 커져서, 순전파 때는 값이 미친 듯이 진동하고,
역전파 때는 기울기가 30배씩 계속 곱해져서 기울기 폭발(Gradient Exploding)이 일어난다

=> 미분할 때 30이 튀어나올 테니, **애초에 가중치를 1/30로 줄여놓자**
그래서 분모에 ω_0를 넣어 **미리 값을 나눠주는 것이**다

3. PINN 손실함수

PINN 손실함수

우리는, 오차가 몇 인지 알아야 한다

입력 : 좌표 (x, y)가 들어감.
예측 : 신경망이 전기장 u를 뱉어냄.
1차 미분 : u가 x에 따라 얼마나 변하는지 계산 ($\frac{\partial u}{\partial x}$) → 자동 미분기능 사용
2차 미분 : 변화율이 또 얼마나 변하는지 계산 ($\frac{\partial^2 u}{\partial x^2}$) → **이 값을 사용**
검사 (Physics Loss) : 2차 미분값 + k^2 * u - 소스 = 0 인지 확인.
학습 : 0이 아니면 그 오차(Loss)를 줄이기 위해 가중치 수정.

3. PINN 손실함수

PINN 손실함수

$$\mathcal{L} = \mathcal{L}_{\mathcal{PDE}} + \mathcal{L}_{\mathcal{BC}}$$

전체 에러($\mathcal{L}$)는 물리 법칙을 어긴 에러($\mathcal{L}_{\mathcal{PDE}}$)와 경계 조건을 어긴 에러($\mathcal{L}_{\mathcal{BC}}$)의 합 이다.

$$\mathcal{L}_{\mathcal{PDE}} = \frac{1}{N_f}\sum_{i=1}^{N_f} |\nabla^2 \hat{u} + k^2 \hat{u} - f|^2$$

$\hat{\boldsymbol{u}}$**:** AI가 추측한 답안지.

$\boldsymbol{\nabla^2 \hat{u} + k^2 \hat{u} - f}$**:** 앞서 본 **지배 방정식.** $\nabla^2 u(\mathbf{x}) + k^2 u(\mathbf{x}) = f(\mathbf{x})$

이항하면 0 되어야 한다

3. PINN 손실함수

PINN 손실함수

∇^2 → 라플라스 연산자. 미분 두 번 이다.

$$\mathcal{L}_{\mathcal{PDE}} = \frac{1}{N_f}\sum_{i=1}^{N_f}|\nabla^2\hat{u} + k^2\hat{u} - f|^2$$

지배 방정식이 헬름홀츠 이기 때문에 2차 미분이 되어 있다.

근데 ReLU? 이차 미분 하면 전부 0 된다 -> 이래서 SIREN 쓰는거다

3. PINN 손실함수

PINN 손실함수

$$\mathcal{L}_{\mathcal{BC}} = \frac{1}{N_b}\sum_{i=1}^{N_b}\left|\hat{u}(\mathbf{x_{edge}}) - g(\mathbf{x_{edge}})\right|^2$$

$\hat{u}(\mathbf{x_{edge}})$: AI가 출력한 가장자리 좌표의 값
$g(\mathbf{x_{edge}})$: 원래의 가장자리 좌표의 값
이거 두 개로 MSE

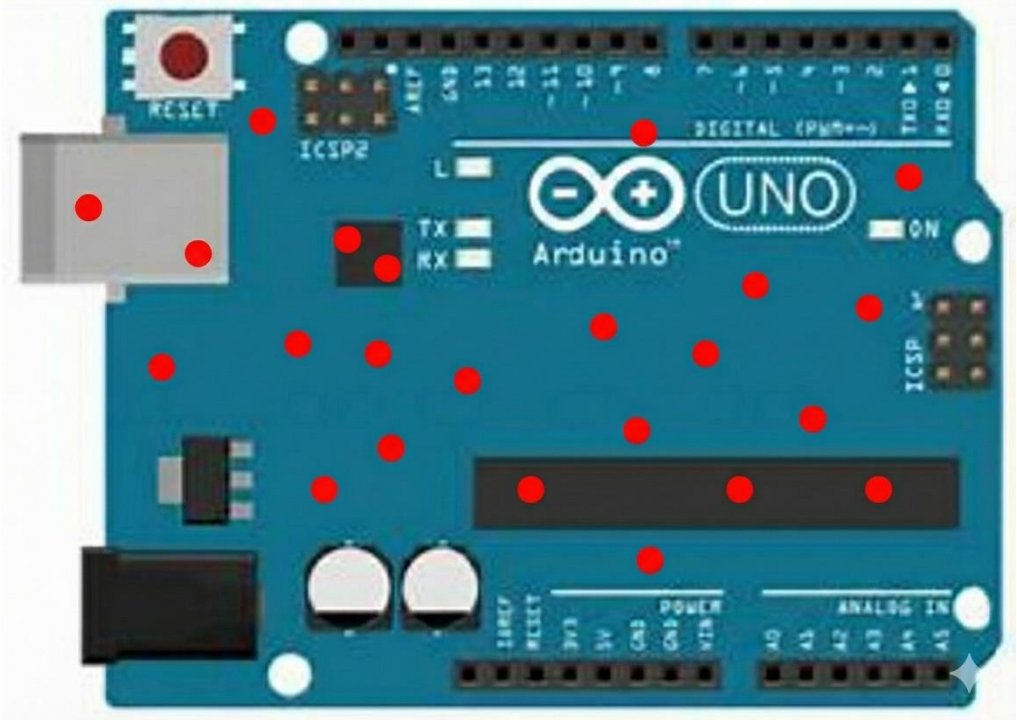
비지도학습 이기 때문에, 가장자리 값을 무조건 0으로 해라 아니면
내부의 물리법칙과 같게 해라 이런 식으로 조건 설정

3. PINN 손실함수

PINN 손실함수

근데, PCB 기판을 넣는다고 했을 때, 이미지로 못넣지 않나요? 그러면 Flatten 해서 넣나요?

아니요 그거 아닙니다.

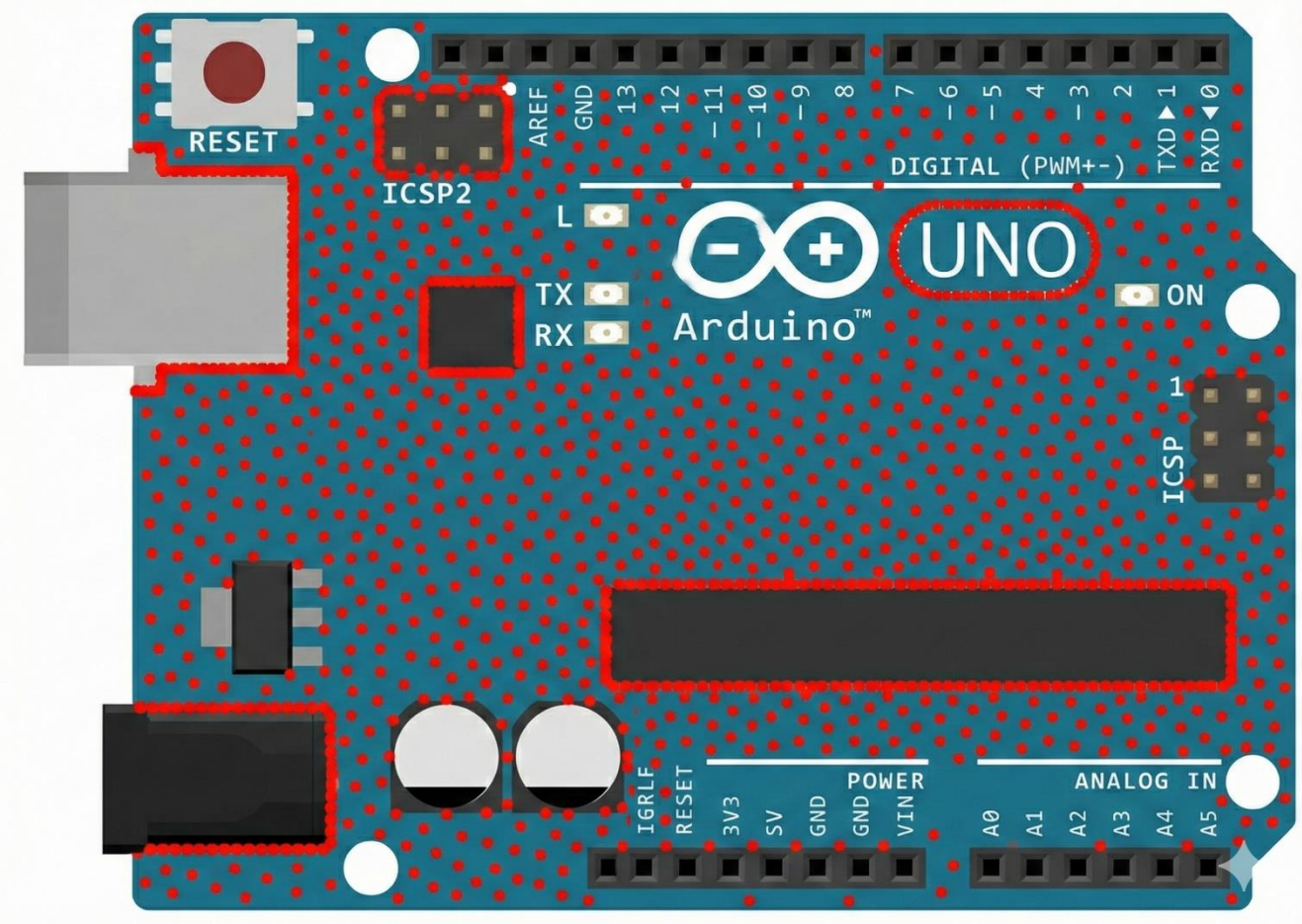


이런식으로 점 찍고, 그 점의 좌표를 넣어서 진행합니다
이때의 점 수가 N_f 입니다

? 근데 저래 점 적으면 무조건 오버피팅 아닌가요?

3. PINN 손실함수

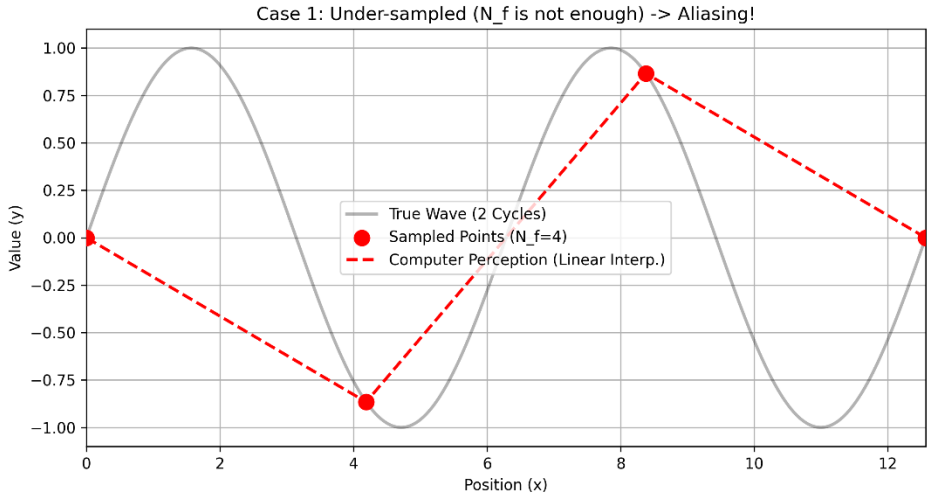
PINN 손실함수

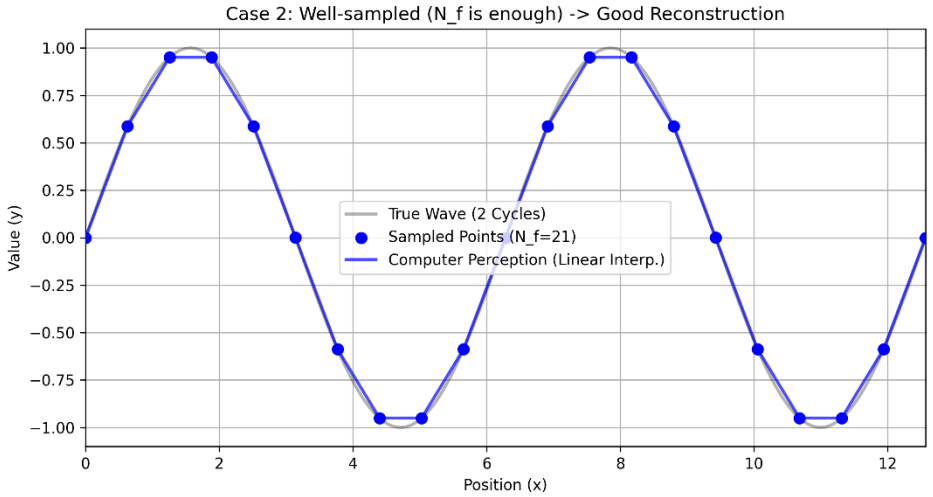


이렇게 떡칠해서 넣습니다. 이거도 사실 적은편임

3. PINN 손실함수

PINN 또한, PINN에선 Epoch 가 Iteration이고, 매 Iteration마다 N_f의 위치가 바뀐다.





기판 크기도 중요하지만, 그 기판에 흐르는 주파수가 더 중요

주파수가 고주판지 저주판지에 따라 N_f 값 정하면 된다.

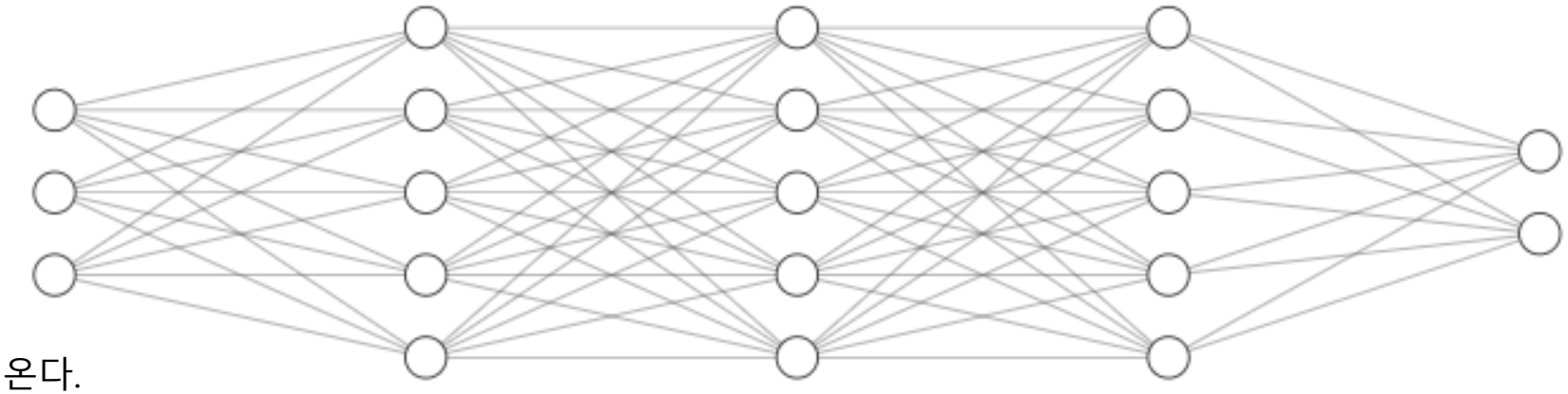
이쁘게 나오는 N_f는 국룰이 파장당 10 ~ 20개 이다. PPW

3. PINN 손실함수

PINN

노드 도착 전 가중치 다 곱해준다

두번 더 반복



좌표값이 들어온다.

노드 도착하면 값 다 더하고,

이후 편향 더해 준다

그 값을 활성화 함수 SIREN 에 넣는다

출력층에 나온 값들 가지고,
손실함수 앞에서 한 그거 계산

유전율 등등의 조건은 다 손실함수에 정의. 좌표 (x, y)를 던져주면, 거기가 금속인지 공기인지 판단해서 유전율 값을 뱉어주는 파이썬 함수를 하나 짜두면 된다

3. PINN 손실함수

PINN

왜 고주파에서 FEM 보다 더 좋을까?

1. **SIREN**을 썼을 때 고주파 학습을 잘 한다

2. 주파수 상관 없이, 앞서 보여준 그 NLP 반복하는 것일 뿐이다.
   하지만, 해석해야 하는 물리적 크기가 커질수록, N_f가 늘어나고,
   주파수가 높아질수록 Iteration이 높아져야 예쁜 결과가 나온다.

FEM: 주파수가 높음 → 격자 개수 폭발 → **행렬 연산 비용 폭발**

PINN: 주파수가 높음 → 가중치 숫자 값만 커질 뿐, **곱셈 연산 횟수는 동일함**.

단, 발산할 확률도 증가. 이건 딥러닝 잘 하면 좋은 결과 나온다.

3. PINN 손실함수

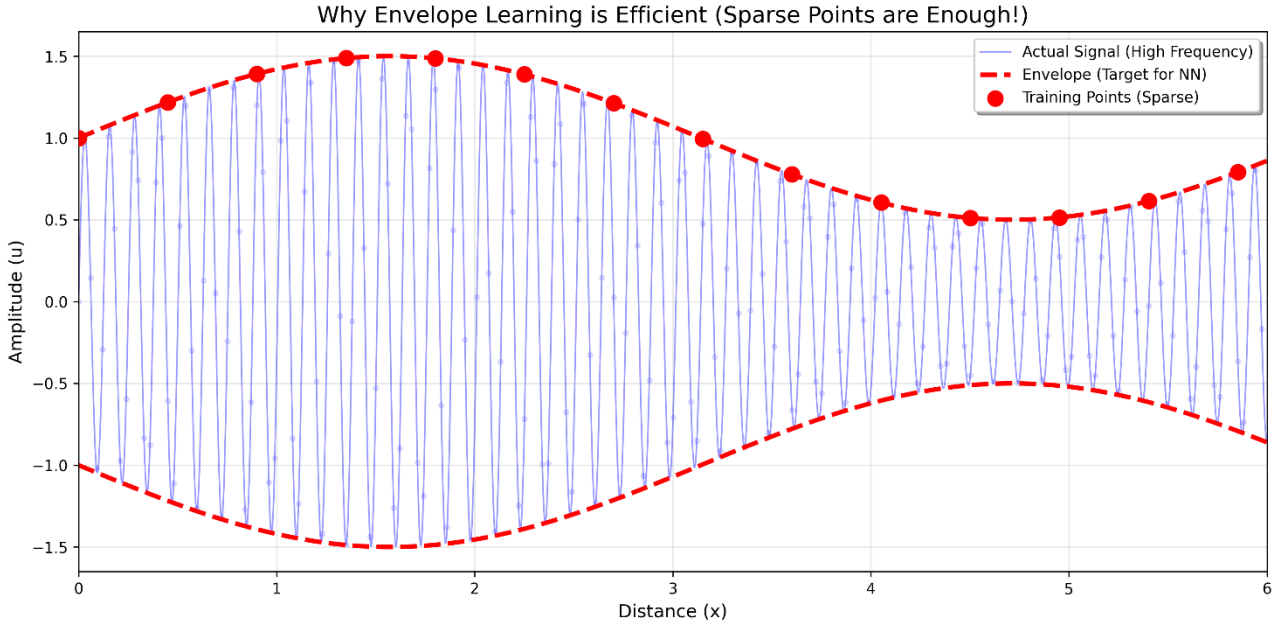
PINN

그러면, 크고 아름다운 안테나 해석을 위해 중요한 지표는 뭐가 있을까?

N_f **:** 파장 하나를 그릴 만큼 점이 촘촘한가? (최소 4개 이상/파장)
Iteration : 충분히 오래 돌렸는가? (SIREN은 수렴이 빠르지만 고주파는 오래 걸림)
Model Size : 신경망이 이 복잡한 파동을 다 기억할 만큼 뚱뚱한가? (노드 개수 늘리기)
Batch Size : 한 번에 충분히 넓은 영역을 보고 있는가

4. Envelope Method (포락선 학습법)

Envelope Method



자 들어가는 N_f의 수가
붉은 색이 많을까,
파란색이 많을까?

빠르게 진동하는 $\sin(kx)$ 부분은 우리가 이미 수식으로 알고 있다.
굳이 모델에 학습시키지 말고, **수학으로 그냥 곱해주자.**
대신 딥러닝은 **천천히 변하는 껍데기(빨간선)만 배우게 하자**

4. Envelope Method (포락선 학습법)

Envelope Method

A. 일반 PINN (힘든 길)

$u(x) = \text{Network}(x)$

신경망이 $u(x)$ 전체를 다 예측해야 한다.
고주파 진동까지 다 그려내야 하므로 **파장당 10개의 점**이 필수이다.

B. 포락선 PINN (똑똑한 길)

$u(x) = \text{Network}(x) \times e^{jkx}$

$\boldsymbol{e^{jkx}}$**:** 진동하는 고주파 성분(Carrier). 이건 **학습하는 게 아니라 우리가 그냥 곱해주는 식** 이다.
$\textbf{Network}(\boldsymbol{x})$**:** 여기가 바로 포락선(Envelope)이다. 신경망은 진동을 뺀 나머지
(진폭과 위상의 느린 변화)만 학습한다.

4. Envelope Method (포락선 학습법)

Envelope Method

	일반 PINN (SIREN)	포락선 PINN (Envelope)
타겟 파형	미친 듯이 진동하는 파동	완만한 곡선 (껍데기)
필요한 점 (N_f)	수백만 개 (파장당 10개)	수천 개 (변화율에 비례)
학습 난이도	극상 (Loss가 잘 안 줄어듦)	하 (Loss가 쭉쭉 떨어짐)
최종 결과물	뭉개지거나, 0으로 수렴 (실패)	**선명한 파동 패턴 (성공)**

수학적으로는 동일한 해를 가리키지만, 돌려보면 포락선 PINN만이 정답에 도달

5. FFT

Near-to-Far Field Transformation (FFT)

근거리장의 분포를 푸리에 변환(Fourier Transform)하면 원거리장 패턴이 된다

100m짜리 안테나를 해석해서, 수 km, 혹은 수백 km 밖(위성 궤도)으로 전파가 어떻게 날아가는지 알고 싶다면?

PINN의 해석 영역(Z_{max})을 100km까지 잡는다.
→ 점(N_f)이 수천억 개 필요함 → **컴퓨터 폭발**.

때문에, 근거리장만 풀고, NTFF 써서 계산한다

5. FFT

Near-to-Far Field Transformation (FFT)

$$E_{far}(\theta) \propto \int_{-D/2}^{D/2} E_{near}(x) \cdot e^{jkx\sin(\theta)} dx$$

$E_{near}(x)$: PINN이 계산한 안테나 바로 앞의 전자기장.
$e^{jkx\sin(\theta)}$: 각도 θ로 날아갈 때 발생하는 위상 지연(Phase Delay).
$\int$: 모든 위치의 파동을 합침.

파이썬 코드 내부에 직접 정의해서 사용. 즉, 다른 것들도 식 정의해서 사용하면 그만이다
예를들어 36000km 상공 정지궤도 위성에 쏠수 있다 없다 이런거도 계산 가능하단 이야기 이다.

6. PINN를 통한 위성안테나 해석

Envelope Method

뭔가 정확도가 낮을것 같은 느낌이 들지만,
실험을 통해 아니라는 것을 증명해 보겠다

앞선 개념을 전부 이해 했으면, 이제 이어지는 코드를 이해할 수 있다.

6. PINN를 통한 위성안테나 해석

Envelope Method

100m급 구조물에서 발생하는 1mm의 미세 오차를 AI가 찾아낼 수 있는가?

타겟 구조물 : 지름 **100m**, 초점거리 40m의 초대형 파라볼라 안테나.
주파수 환경 : 18 GHz (파장 $\lambda \approx 1.67$ cm). 전기적 크기가 6000λ에 달하는 초고주파 환경.
비교군 설정:

1. **Smooth Model (Ideal) :** 오차가 전혀 없는 이상적인 표면.
2. **Defected Model (Test) :** 안테나 표면 지점 중, 무작위로 선정된 **단 10곳**에 **1mm ($\lambda/16$)의 미세 가공 오차**를 주입. 안테나 표면이 1mm 찌그러져 있으면, 전파가 그만큼 더 갔다가 돌아와야 하므로 **왕복 2mm의 경로 차이**가 생긴다. 지름 2cm, 깊이 1mm의 구멍 있다고 생각하자

목적 : PINN이 이 미세한 국소 결함(Local Defect)의 위치를 역추적(Localization)하고, 이것이 전체 통신 성능에 미치는 영향을 정량화하는 것.

6. PINN를 통한 위성안테나 해석

Envelope Method

```
PHYSICAL_BATCH = 12000
ACCUM_STEPS = 10
ITERATIONS = 5000
HIDDEN_DIM = 256
LAYERS = 8
LR = 1e-4
```

PHYSICAL_BATCH : 배치 사이즈 12000개. 적어 보인다.

ACCUM_STEPS(10) : 한번에 가중치 업데이트 하지 않고, Iteration 10번 기다렸다가 업데이트 한다.
이러면, 배치 사이즈를 10 배 키울 수 있다.

ITERATIONS : 5000번 반복학습. N_f도 5000번 바뀐다.

HIDDEN_DIM : 은닉층의 노드 수 엄청 넉넉하다

LAYERS : MLP의 레이어가 8개라는 의미 보통 4 ~ 5층이므로, 엄청나게 큰 것이다

LR = 학습율

6. PINN를 통한 위성안테나 해석

Envelope Method

배치가 120000 이면 충분한 이유

주파수 (f): 18 GHz
파장 (λ): 약 **0.0167 m (1.67 cm)**
해석 영역 (Area):
코드 상 tmp[:,0]*D*1.5 (가로 약 150m) × tmp[:,1]*Z_MAX (세로 120m)
전체 면적 A ≈ 150 × 120 = $\mathbf{18,000\ m^2}$$

일반 PINN 이였다면?

$$N_{req} = \frac{\text{전체 면적}}{(\text{해상도})^2} = \frac{18{,}000}{(0.00167)^2} \approx \frac{18{,}000}{0.00000278}$$

$N_{req} \approx \mathbf{6,470,000,000}$ (약 $\mathbf{65}$억 개)

6. PINN를 통한 위성안테나 해석

Envelope Method

포락선 PINN이라면?

물리적으로 포락선의 변화는 **수십 cm ~ 수 미터** 단위로 완만하게 일어난다.

아주 보수적으로 잡아서 0.5m (50cm)마다 점을 찍어도 포락선의 곡선을 그리는 데 충분하다고 가정하자

$$N_{env} = \frac{\text{전체 면적}}{(\text{해상도})^2} = \frac{18{,}000}{(0.5)^2} = \frac{18{,}000}{0.25}$$

$$N_{env} = \mathbf{72{,}000}\ \text{개}$$

$$\mathbf{120{,}000}\ (\text{공급}) > \mathbf{72{,}000}\ (\text{수요})$$

6. PINN를 통한 위성안테나 해석

Envelope Method

```
FREQ = 18e9;
C = 3e8; LAMBDA = C / FREQ;
K = 2 * np.pi / LAMBDA
D = 100.0;
F_LEN = 40.0;
Z_MAX = 120.0
```

FREQ : 18GHz
C : 빛의 속도
LAMBDA : 파장
K : 파수
D :안테나 지름 100m
F_LEN : 초점거리 40m
Z_MAX : 해석영역 길이 120m

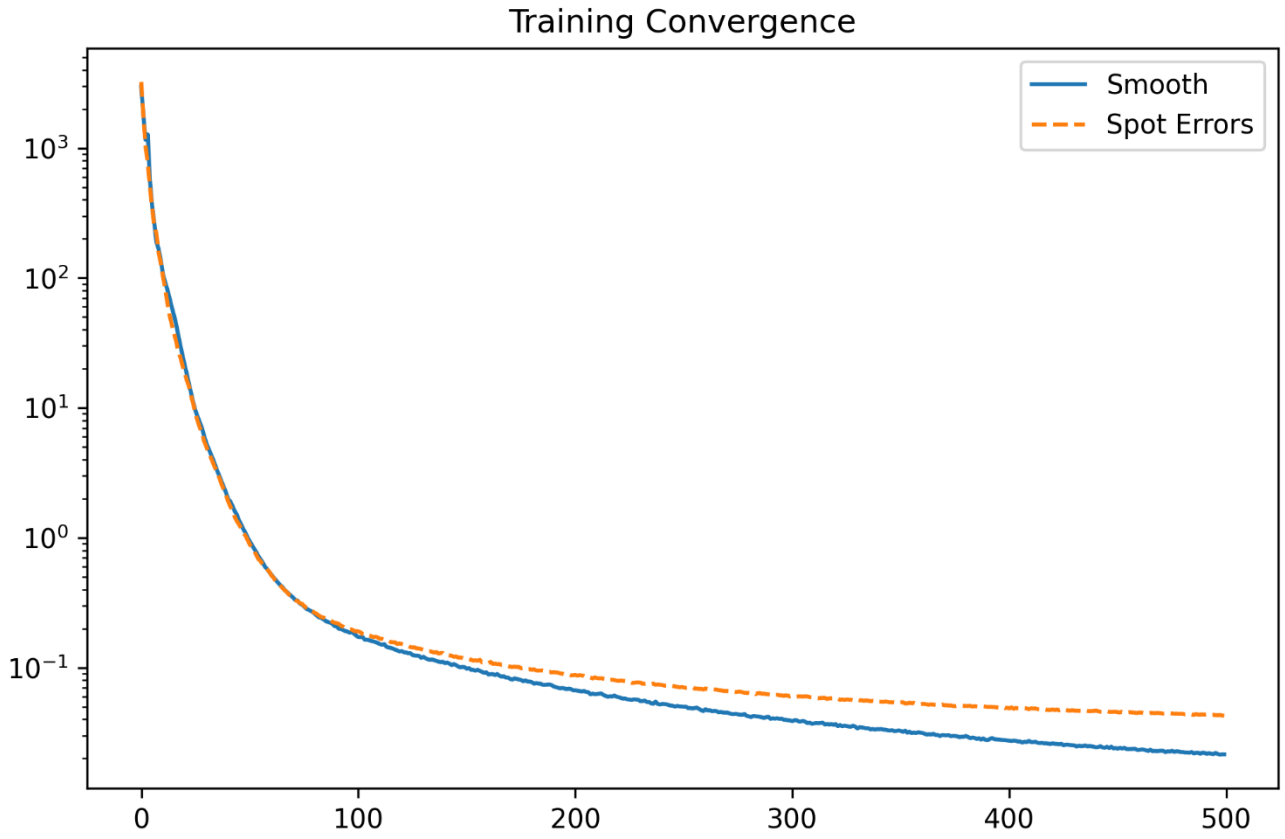
6. PINN를 통한 위성안테나 해석

Envelope Method

앞선 조건에, 다음을 사용
- PINN
- SIREN
- Envelope Method
- Gradientt Accumulation
- Near-to-far Fiealf Transformation

6. PINN를 통한 위성안테나 해석

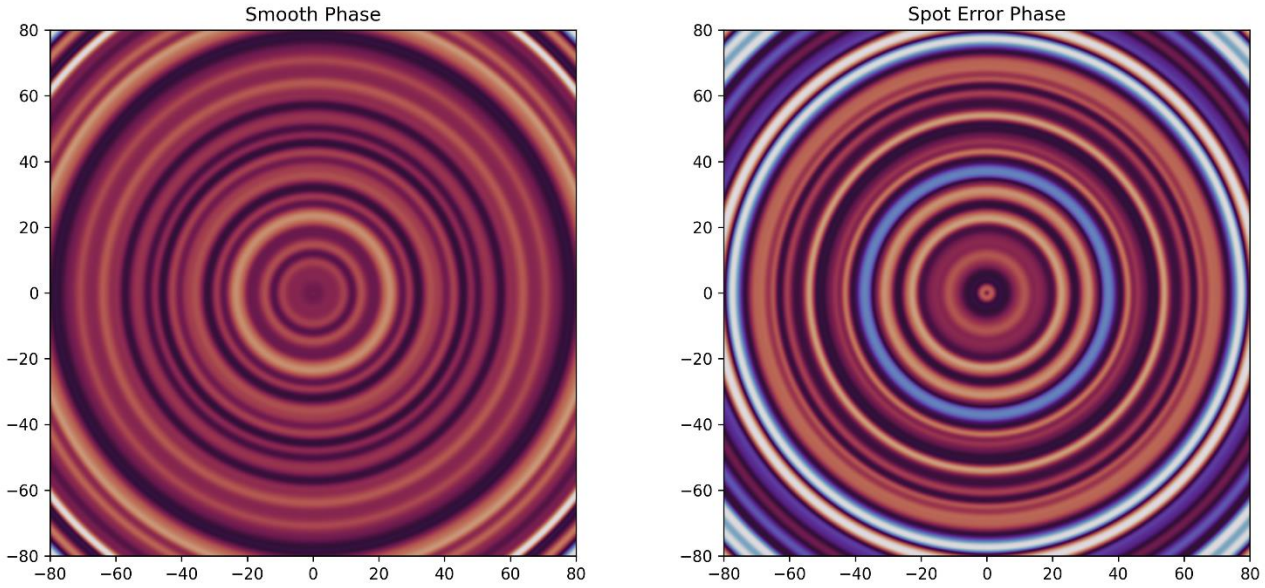
실험 결과



1. Loss curve
오차 포함 모델이 미포함 모델보다 Loss가 약간 더 높게 유지된다.
이는 1mm의 미세한 오차를 잘 인식하고 딥러닝 했다는 증거이다

6. PINN를 통한 위성안테나 해석

실험 결과

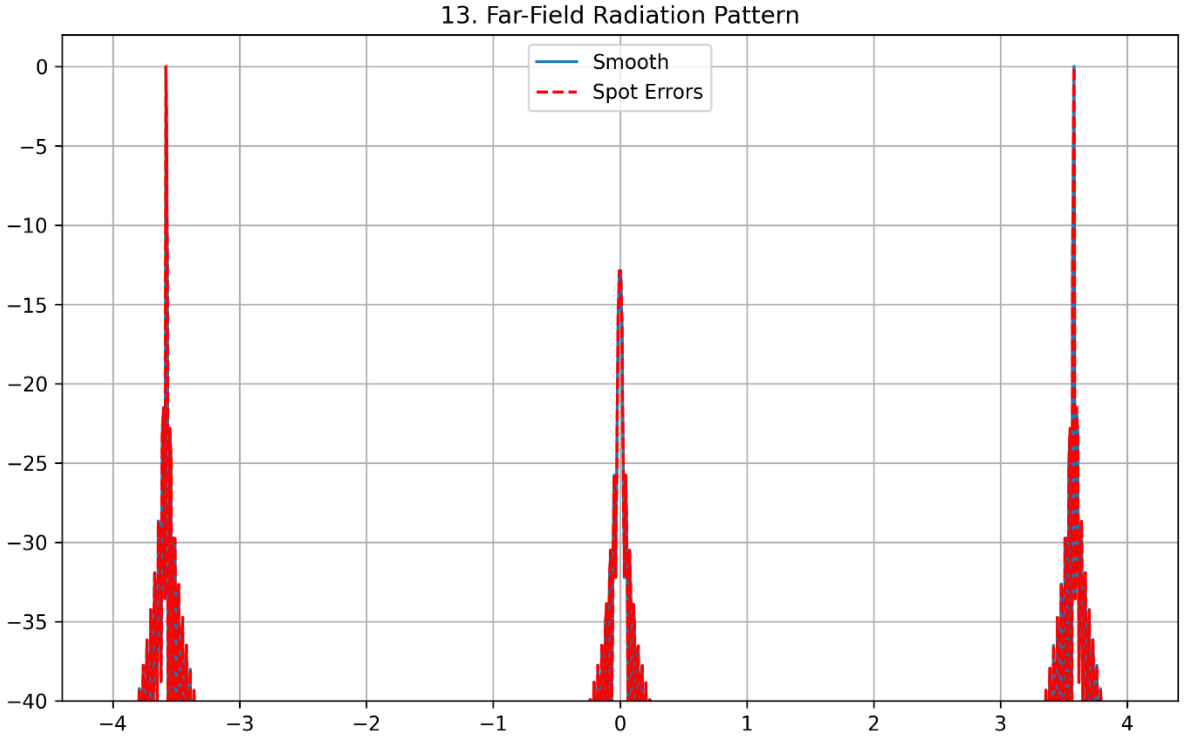


2. 위상 분석
포락선 학습법을 썼지만, 내부적으로 e^{jkx}를 곱해서 복원했기 때문에 위상 정보가 손실되지 않고 살아있음을 보여준다

AI 실험 결과로 그려진, top view의 방사패턴. 아주 이쁘다

6. PINN를 통한 위성안테나 해석

실험 결과



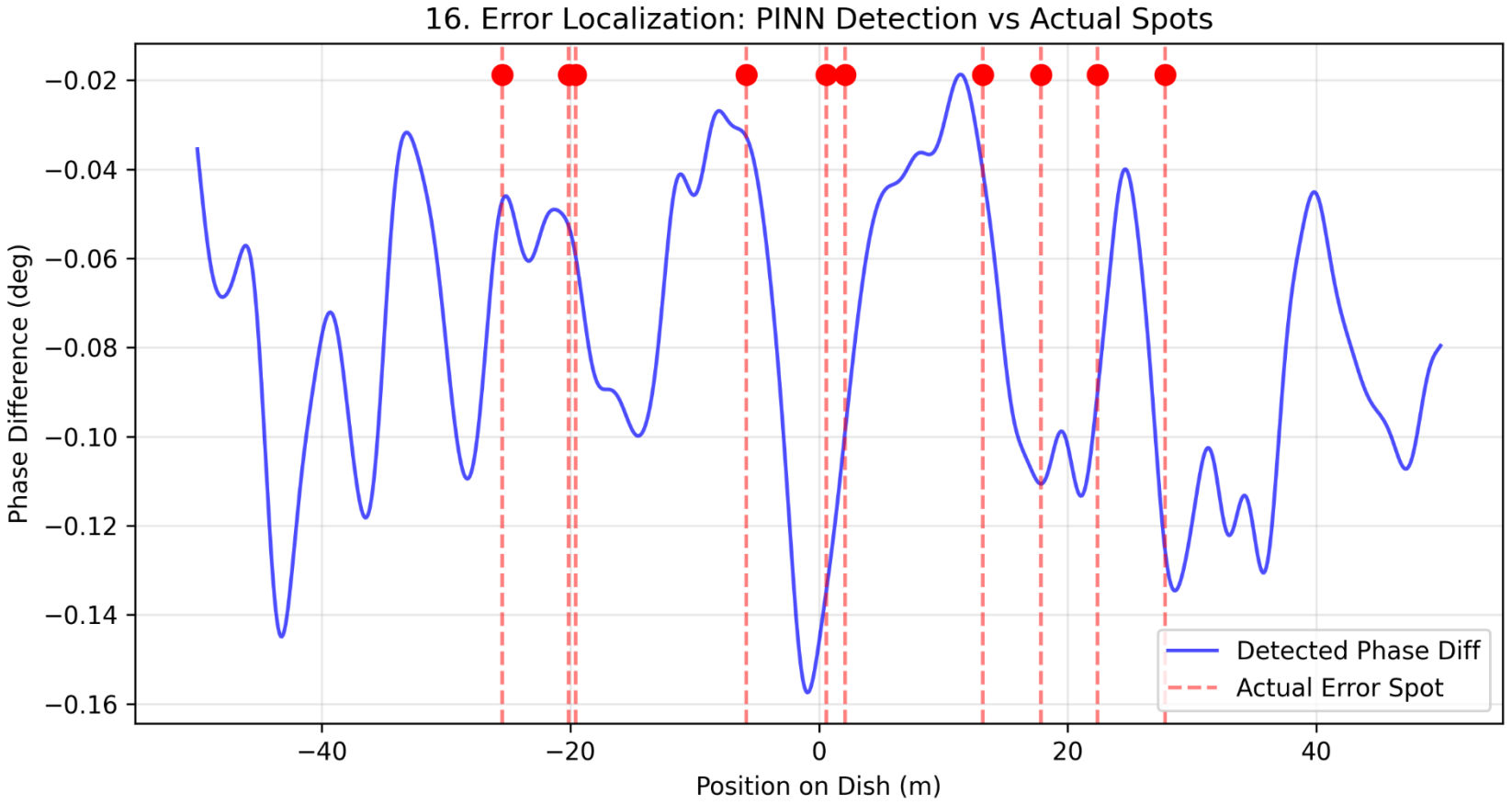
원격장 포인팅 벡터이다

3. 성능평가
1mm 오차 10개 정도는 안테나 성능에 **거의 영향을 주지 않는다.**

6. PINN를 통한 위성안테나 해석

ONECLICK AI

실험 결과



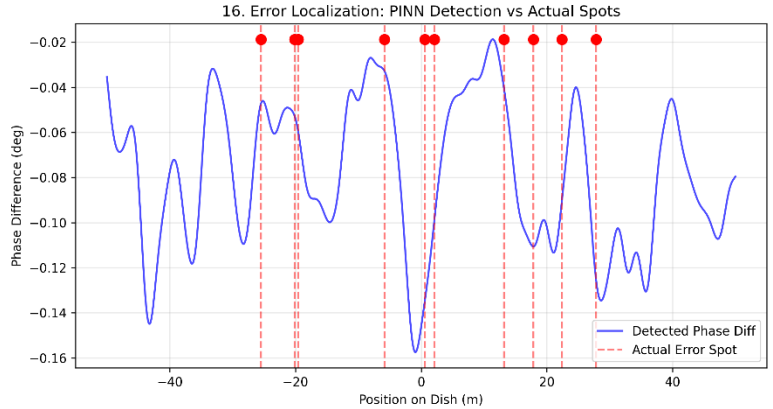
붉은 점 : 실제 오차 위치
파란 선 : 매끈한 안테나의 파동과 오차 있는 안테나의 파동의 위상차이

6. PINN를 통한 위성안테나 해석

ONECLICK AI

실험 결과

즉, 지름 2cm, 깊이 1mm 인 안테나의 오차와 매끈한 안테나의 오차를 구별하지 못한다? 그러면 파란 선이 0도 근처로 쭉 나왔을 것이라는 거다.



붉은 선과 파란 선이 겹치는 곳에서 급격한 변화가 일어나는 것이 실험이 잘 되었다는 증거이다.

파동은 퍼지는 것 이기에, 오차가 없는 다른 부분에서도 오차의 영향을 받아 심하게 흔들리는 모습을 보여준다.

오차 때문에 망가진 전체 파동의 흐름(물결)을 보여주고 있는 것.
PINN이 그 미세한 1mm가 만든 물결을 놓치지 않고 다 잡아냈다는 뜻

6. PINN를 통한 위성안테나 해석

실험 결과

안테나 정밀 진단 표 (AI가 판단한 거임 진짜임)

Metric	Value
Detected Peaks	10 Spots
Gain Loss (dB)	-0.0058
Sidelobe Level (dB)	-0.01

숨겨둔 10개의 오차 출력한 거임 이건 직접 찾은거 아님
이거 밑은 다 직접 계산한 것

10개의 오차 때문에 -0.0058dB의 이득손실 발생
부엽 레벨 변화 -0.01dB로 거의 없다

6. PINN를 통한 위성안테나 해석

실험 결과

이득 손실 계산:

오차 있는 안테나(E_f)와 정상 안테나(E_s)의 전파 세기를 비교해서 계산

```
# 1. 원거리장(Far-Field) 패턴을 데시벨(dB)로 변환
# 핵심: 정상 안테나의 최대값(np.max(FF_s))으로 나누어 정규화 한다.
# 즉, 정상 안테나의 피크 = 0dB가 된다.
FF_f_db = 20*np.log10(FF_f/np.max(FF_s)+1e-12)
# 2. 오차 있는 안테나(FF_f_db)의 최대값을 찾는다.
gain_loss = np.max(FF_f_db)
```

6. PINN를 통한 위성안테나 해석

실험 결과

부엽 레벨 변화 계산:

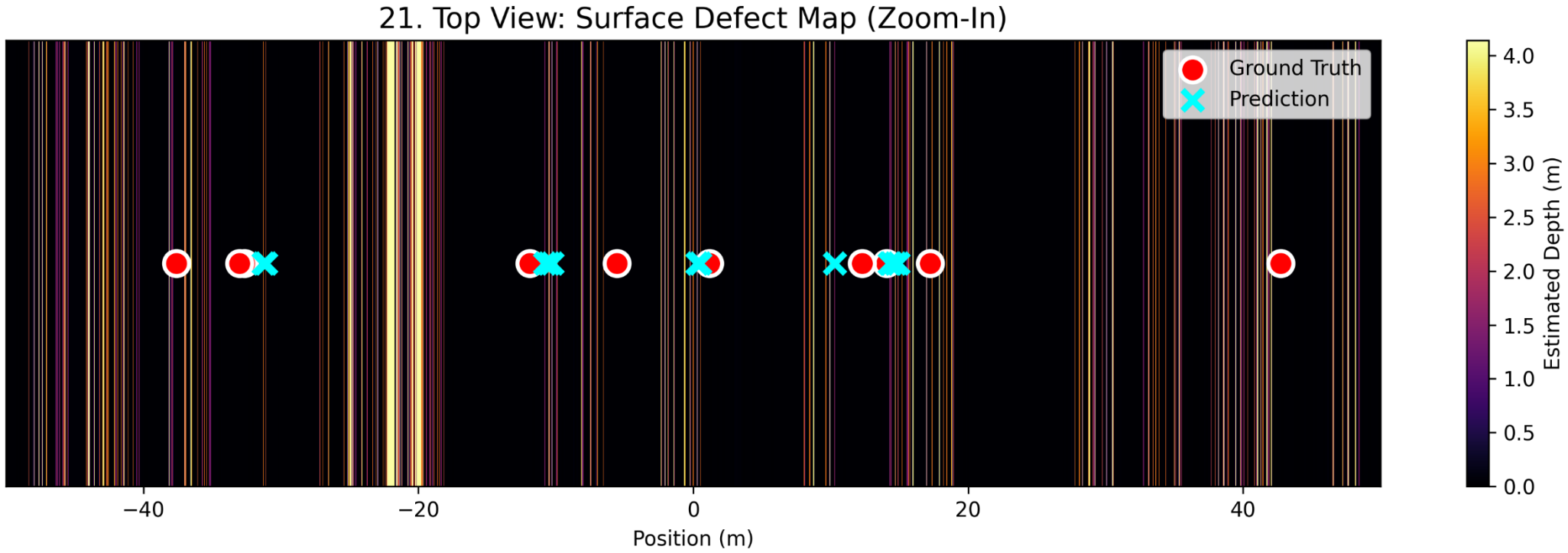
메인 빔(가운데 솟은 것) 말고, **옆으로 새어나가는 전파** 중 가장 센 놈을 찾은 것

```
# 1. 각도 범위 정의: angles는 -4도 ~ +4도
# 2. 조건 필터링: np.abs(angles) > 1
# -> "각도가 -1도보다 작거나, +1도보다 큰 구역(사이드)만 보겠다"는 뜻.
# -> 메인 빔(0도 근처)을 제외하는 코드.
# 3. 그 사이드 구역 중에서 가장 큰 값(max)을 뽑는다.
sidelobe = np.max(FF_f_db[np.abs(angles)>1])
```

6. PINN를 통한 위성안테나 해석

ONECLICK AI

역문제 해결에서 강한 PINN



정확도는 높지 않다 더 실험해 봐야 한다

6. PINN를 통한 위성안테나 해석

결론

지름 100m의 초대형 파라볼라 안테나에 있는 지름 2cm, 깊이 1mm의 오차 10를 통한 위상 변화 등을 완벽하게 파악할 수 있다.

어디에 오류 있는지도 역문제 해결로 찾아낼 수 있다. 전부 다 찾지는 못한다.

이것은 극적인 변화를 주기 위해 구멍 판 것이다. 다른 조건으로도 실험 해 볼 수 있다.

6. PINN를 통한 위성안테나 해석

결론

이거 찾아보면서 본 논문들 정리

Implicit Neural Representations with Periodic Activation Functions
https://arxiv.org/abs/2006.09661

Phase-Shift Deep Neural Network for High-Frequency Approximation and Wave Problems
https://epubs.siam.org/doi/10.1137/19M1310050 [읽어보고 싶은데 비쌈]

Physics-Informed Neural Networks for Electromagnetic Analysis
https://ieeexplore.ieee.org/abstract/document/9740189[이거도,, 비쌈]

Physics-informed neural networks: A deep learning framework for solving forward and inverse problems...
https://www.sciencedirect.com/science/article/abs/pii/S0021999118307125

이거 말고도 조금 더 읽었지만, 읽은거도 이해한게 절반 뿐인 것 같다.

감사합니다