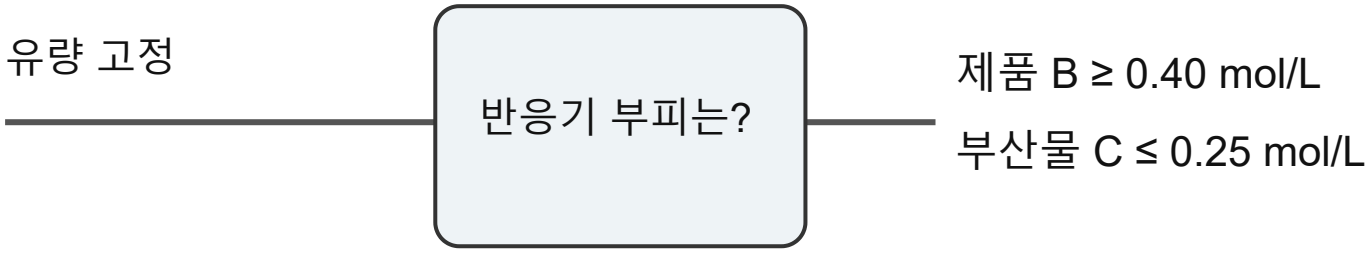
5주차: 예측 모델을 최적화 문제에 넣기

제품 규격을 맞추는 최소 반응기

제품 B는 0.40 mol/L 이상, 부산물 C는 0.25 mol/L 이하가 필요합니다. 유량이 일정할 때 반응기 부피를 얼마나 줄일 수 있을까요?



체류시간을 늘리면 B와 C가 함께 변합니다.

제품 규격과 반응기 부피 최소화

체류시간을 주면 신경망이 출구 농도를 계산합니다. 설계에서는 반대로, 농도 조건을 만족하는 체류시간을 찾아야 합니다.

체류시간을 늘리면 A가 B로 전환되지만 B도 C로 반응합니다. B만 크게 만드는 선택과 부피를 줄이는 선택은 같지 않습니다.

이번 주에는 이 선택을 최적화 문제로 적고, 고정된 ReLU 모델의 계산을 선형식과 이진변수로 표현해봅니다.

최적화 정식화의 학습 목표

1. 조작변수·고정 조건·예측 출력을 구분하고 목적함수와 제약을 적습니다.
2. ReLU 한 개의 Big-M 식을 두 분기로 증명합니다.
3. 입력 범위에서 유효한 pre-activation bounds를 계산
4. 직접 작성한 MILP와 forward 계산을 같은 입력에서 대조합니다.

실습 제출물: 정식화, 검증 결과와 해의 해석

결정변수와 고정 조건

역할	실습의 값 또는 범위
결정변수	체류시간 $\tau \in [2, 10]$ min
고정 조건	T = 330 K, C_{Af} = 1.2 mol/L
고정 유량	Q = 1 L/min
예측 출력	A·B·C 농도 y [mol/L]
목적	부피 $V = Q\tau$ [L] 최소화

여기서 τ는 고정 유량에서 부피를 선택하는 설계변수. 온도까지 선택하려면 온도를 입력으로 받는 모델과 그 영역의 bounds가 필요합니다.

체류시간을 선택하면 고정 유량 Q에서 필요한 부피 $V=Q\tau$도 정해집니다. 이미 부피가 정해진 장치에서 체류시간을 바꾸려면 유량 등을 바꾸어야 하므로 이 설계 문제와 변수·비용이 달라집니다.

최적화 문제의 기본 구성

$$\min_{u,y} J(u, y; d)$$
$$y = f_\theta(u; d),\ g(u, y; d) \le 0$$

u는 설계·운전 변수, d는 고정 조건, y는 모델 출력입니다. 학습을 마친 가중치 θ도 고정합니다.

가능해: 신경망 등식과 공정 제약을 모두 만족하는 점. 최적해: 가능해 중 목적함수가 가장 작은 점

예측에서는 u를 주고 y를 계산했습니다. 최적화에서는 u와 y를 함께 미지수로 놓고, y=fθ(u)가 두 값을 묶습니다. solver가 y를 마음대로 낮추거나 높여 규격을 맞추지 못하도록 예측 관계 자체를 제약으로 넣습니다.

CSTR 설계 최적화 문제

$$\min_{\tau, y} V = Q\tau$$
$$2 \le \tau \le 10, y = f_\theta(\tau)$$
$$y_B \ge .40, y_C \le .25, y \ge 0$$

Q = 1 L/min이므로 τ를 줄이면 V도 줄어듭니다. y의 농도 단위는 mol/L이며, 모델의 출력은 이 단위로 복원한 뒤 규격과 비교합니다.

농도 규격: B ≥ 0.40. 순도나 생산량을 요구한다면 해당 정의로 제약을 다시 적어야 합니다.

τ=2는 부피가 작아 목적값은 좋지만 B가 부족할 수 있습니다. τ를 늘려 B 하한에 도달하면 더 작은 부피는 그 제약을 위반합니다. 다만 C 상한도 같은 지점에서 만족하는지 확인해야 최소 가능 설계가 됩니다.

실습의 기준 농도

완전 혼합·일정 부피·동일한 유입 및 유출 유량을 가정하고, 1:1 연속반응 A → B → C를 사용합니다. 330 K에서 $k_1 = 0.2\ \text{min}^{-1}$, $k_2 = 0.1\ \text{min}^{-1}$입니다.

$$C_A = \frac{1.2}{1 + .2\tau}$$

$$C_B = \frac{.2\tau C_A}{1 + .1\tau}, C_C = .1\tau C_B$$

다섯 체류시간에서 이 식을 계산해 작은 ReLU 모델을 구성합니다.

가중치를 고정한 계산 그래프

$$h^0 = x$$
$$a^\ell = W^\ell h^{\ell-1} + b^\ell$$
$$h^\ell = \max\left(0, a^\ell\right)$$
$$y = W_{\text{out}} h^K + b_{\text{out}}$$

ReLU: 각 은닉층에서 성분별 적용. 최적화 변수: 입력·중간값·출력. 수치 상수: W, b

가중치까지 변수로 두면 W와 h의 곱이 생기므로 여기서 유도하는 MILP가 성립하지 않습니다.

신경망 내부 숫자도 최적화 변수가 되지만 자유롭게 선택되지는 않습니다. 각 층의 가중합 등식과 ReLU 제약이 앞 층 값으로 뒷 층 값을 결정합니다. 원래 순전파를 동시에 만족해야 하는 연립 관계로 옮기는 것입니다.

입력 scaling의 등식 표현

$$x = \frac{\tau - 2}{8}, 0 \le x \le 1$$
$$\tau = 2 + 8x$$

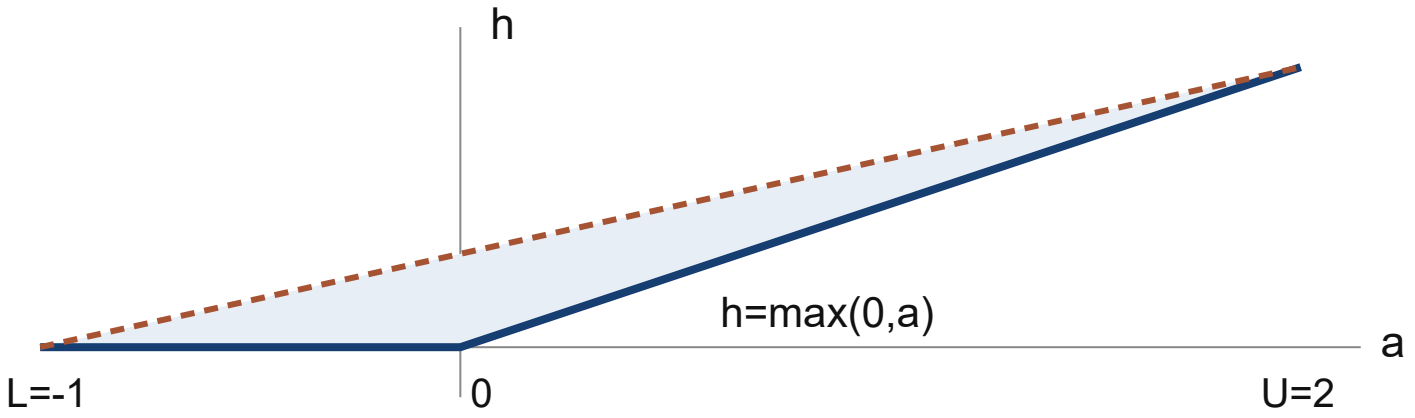
실습은 τ를 무차원 입력 x로 바꿉니다. 학습 때 사용한 offset과 scale을 최적화에서도 그대로 사용합니다.

출력까지 표준화했다면 y = μ + Ds를 추가합니다. μ와 D가 고정이면 이 복원식은 선형입니다.

예를 들어 τ=6이면 x=(6−2)/8=0.5입니다. 신경망에 τ=6을 그대로 넣으면 학습 때의 0~1 입력 범위를 벗어납니다. solver 변수의 이름보다 실제로 어떤 좌표가 각 식에 들어가는지 확인합니다.

ReLU 함수의 구간별 선형 구조

$$h = \max(0, a)$$



파란 선이 정확한 그래프입니다. 색칠한 영역은 이진변수를 연속으로 풀었을 때 허용되는 점을 포함합니다.

a<0 구간은 수평선 h=0, a>0 구간은 직선 h=a입니다. 두 선분 중 어느 쪽을 사용할지 이진변수 z가 선택합니다. z를 연속값으로 풀면 두 선분 사이의 점도 허용되어 출력이 실제 ReLU보다 커질 수 있습니다.

ReLU 그래프와 하한 제약

$$h \geq 0, h \geq a$$

a = −0.2일 때 h = 0.4도 이 두 식을 만족하지만 ReLU 출력은 0입니다.

목적함수나 후속 layer가 큰 h를 선호할 수 있으므로, 임의의 신경망에서는 이 두 하한만 넣어서는 안 됩니다. 정확한 그래프에는 두 분기에 맞는 상한도 필요합니다.

한 개 ReLU의 Big-M 정식화

유효한 유한 bounds $L \le a \le U$를 알고 있고 $L < 0 < U$라고 놓습니다.

$$h \ge 0, h \ge a$$
$$h \le a - L(1 - z)$$
$$h \le Uz, z \in \{0, 1\}$$

분기: z = 0은 비활성, z = 1은 활성. L·U: 해당 뉴런의 pre-activation 하한·상한

$h \le Uz$는 $z=0$일 때 h를 0에 고정합니다. $h \le a - L(1-z)$는 $z=1$일 때 h를 a에 고정합니다. 선택되지 않은 상한은 유효한 L,U 덕분에 해당 가지의 올바른 점을 잘라내지 않습니다.

비활성 분기(z = 0)의 동치성

$$h \le U0 = 0, h \ge 0$$
$$h = 0, a \le h = 0$$

나머지 상한 $0 \le a - L$은 $a \ge L$이므로 만족합니다. 따라서 이 분기는 $L \le a \le 0, h = 0$을 정확히 나타냅니다.

활성 분기(z = 1)의 동치성

$$h \le a - L0 = a, h \ge a$$
$$h = a \ge 0$$

나머지 상한 h ≤ U는 a ≤ U와 같습니다. 따라서 0 ≤ a ≤ U, h = a를 얻습니다.

a = 0에서는 z가 0 또는 1이어도 출력 h는 0으로 정해집니다.

하나의 ReLU 제약에 수치 대입

$a=2x-1$, $x\in[0,1]$이면 $L=-1, U=1$입니다. 네 식은 $h\ge 0$, $h\ge a$, $h\le a+1-z$, $h\le z$입니다.

입력과 가지	하한	상한	결정된 출력
$x=0.2$, $a=-0.6$, $z=0$	$h\ge 0$	$h\le 0$; $h\le 0.4$	$h=0$
$x=0.8$, $a=0.6$, $z=1$	$h\ge 0.6$	$h\le 0.6$; $h\le 1$	$h=0.6$
$x=0.8$에서 잘못된 $z=0$	$h\ge 0.6$	$h\le 0$	불가능

```
a = 2 * x - 1
h = max(0., a)
z = int(a >= 0.)
assert h >= 0 and h >= a
assert h <= a + 1 - z and h <= z
```

이진변수가 임의의 출력을 만드는 것이 아니라, 주어진 a와 양립하는 가지를 선택합니다. $a=0$에서는 두 z가 가능하지만 h는 모두 0입니다.

활성·비활성 뉴런의 선형 표현

유효한 범위	대체 식	이진변수
$U \le 0$	$h = 0$	불필요
$L \ge 0$	$h = a$	불필요
$L < 0 < U$	Big-M 네 부등식	1개

범위를 벗어나는 입력까지 허용하면 이 단순화의 근거가 사라집니다. 입력 영역을 바꾸면 bounds와 분류도 다시 확인합니다.

범위가 $[-2,-0.1]$이면 어떤 입력에도 켜지지 않아 $h=0$으로 대체합니다. $[0.2,1.3]$이면 항상 $h=a$입니다.
부호가 바뀔 가능성이 있는 뉴런에만 가지 선택이 필요하므로 bounds가 이진변수 수를 결정합니다.

가중치 부호를 반영한 구간 bounds

$$a = \sum_i w_i h_i + b$$
$$L = b + \sum_i \min\left(w_i l_i, w_i u_i\right)$$
$$U = b + \sum_i \max\left(w_i l_i, w_i u_i\right)$$

각 입력이 $l_i \le h_i \le u_i$를 만족한다고 가정합니다. 음의 가중치는 입력의 위아래 끝점을 뒤집습니다.

이 구간 계산은 유효한 범위를 주지만, 입력 간 상관관계를 무시하므로 가능한 실제 범위보다 넓을 수 있습니다.

Bounds 손계산 예제

$$a = 2x_1 - 3x_2 + .5$$
$$0 \le x_1 \le 1,\ -1 \le x_2 \le 2$$
$$L = -5.5, U = 5.5$$

최솟값은 $(x_1, x_2) = (0, 2)$, 최댓값은 $(1, -1)$에서 나옵니다. 임의의 큰 숫자를 고르기 전에 이 계산부터 합니다.

하한은 양수 계수 2에 x_1의 하한 0을, 음수 계수 −3에 x_2의 상한 2를 넣어 $0-6+0.5=-5.5$입니다. 상한은 반대로 $2-3\times(-1)+0.5=5.5$입니다. 음수 계수가 곱해질 때 극값 방향이 뒤집힙니다.

Bounds의 유효성과 정식화의 정확성

실제 pre-activation이 $a = 0.7$인 점에 $U = 0.5$를 사용하면 $h = 0.7$을 표현할 수 없습니다. $z = 1$에서도 $h \leq 0.5$이기 때문입니다.

반대로 충분히 넓은 유효 bounds는 정수해의 그래프를 유지하지만, 연속 완화를 느슨하게 만들 수 있습니다.

학습 샘플에서 관측한 최솟값·최댓값은 전체 입력 영역의 bounds를 자동으로 보장하지 않습니다.

ReLU 보간 모델의 고정 계수

τ = 2, 4, 6, 8, 10 min의 기준 농도를 선형 보간합니다. 꺾이는 위치를 hidden layer에 고정하고, 보간 기울기에서 출력 가중치를 계산

$$a_j = x - t_j, h_j = \max\left(0, a_j\right)$$

$$t = \left[\begin{array}{cccc} 0 & .25 & .50 & .75 \end{array}\right]$$

$$y_k = b_k + \sum_{j=0}^{3} W_{\mathrm{jk}} h_j$$

j = 0, 1, 2, 3이며 출력 k는 A, B, C입니다. 이 실습의 모델 구성은 보간이고, 경사하강 학습은 사용하지 않습니다.

첫 구간의 기울기는 처음부터 적용합니다. 이후 절점에서 ReLU가 켜질 때마다 새 기울기와 이전 기울기의 차이를 더합니다. 그러면 현재 켜진 기울기 변화량의 합이 그 구간의 기울기가 됩니다.

보간 기울기에서 ReLU 가중치로

B의 첫 세 절점 값은 τ=2,4,6에서 0.285714286,0.380952381,0.409090909입니다. x=(τ−2)/8이므로 절점의 x 간격은 2/8=0.25입니다.

계산	결과
첫 구간의 x 기울기	(0.380952381−0.285714286)/0.25=0.380952381
두 번째 구간의 x 기울기	(0.409090909−0.380952381)/0.25=0.112554113
첫 ReLU 가중치 W_0B	첫 기울기 0.380952381
절점 0.25의 가중치 W_1B	새 기울기−이전 기울기=−0.268398268

0.25<x<0.5에서는 첫 두 ReLU만 미분이 1이므로 전체 기울기는 0.380952381 − 0.268398268 = 0.112554113입니다. 새 항은 절점에서 0이므로 값의 연속성도 유지됩니다.

τ 기준 기울기를 사용했다면 x 기준 가중치로 옮길 때 8을 곱해야 합니다. 같은 농도 변화가 x에서는 8배 짧은 간격에 걸쳐 일어나기 때문입니다. A와 C 열도 같은 방법으로 구합니다.

실습 모델의 뉴런별 bounds

뉴런	a_j	L_j	U_j	처리
0	x	0	1	$h_0 = x$
1	x − 0.25	−0.25	0.75	이진변수
2	x − 0.50	−0.50	0.50	이진변수
3	x − 0.75	−0.75	0.25	이진변수

총 이진변수는 3개입니다. a를 식에 대입해 없애면 연속변수는 x 1개, h 4개, y 3개로 8개입니다.

τ=5 min에서 은닉값과 B 농도

x=(5−2)/8=0.375를 네 절점 [0,0.25,0.50,0.75]에서 뺍니다.

j	$a_j=x-t_j$	h_j	B에 더해지는 값 $W_jB\cdot h_j$
0	0.375	0.375	0.380952381×0.375=0.142857143
1	0.125	0.125	−0.268398268×0.125=−0.033549784
2	−0.125	0	0
3	−0.375	0	0

편향 0.285714286을 더하면 B≈0.395021645 mol/L입니다. 4분과 6분의 농도를 반씩 섞어도 (0.380952381+0.409090909)/2=0.395021645가 됩니다.

ReLU 가중합과 구간 보간이 같은 계산을 다른 표현으로 수행합니다. 기준 공정의 5분 B=0.4와는 차이가 있어, 이 예측은 B≥0.40 규격에 조금 못 미칩니다.

Big-M 제약의 행렬 표현

뉴런 1에서 $a_1 = x - 0.25$, $L_1 = -0.25$, $U_1 = 0.75$

$$h_1 - x \geq -.25$$
$$h_1 - x + .25z_1 \leq 0$$
$$h_1 - .75z_1 \leq 0$$

$h_1 \geq 0$과 $0 \leq z_1 \leq 1$은 변수 bounds에 적고, z_1을 정수로 지정합니다. 실습은 같은 방식으로 전체 행렬을 직접 조립합니다.

두 번째 행은 $h_1 \leq x - 0.25 + 0.25(1 - z_1) = x - 0.25z_1$을 왼쪽으로 모은 식입니다. $x=0.5, z_1=1$이면 $h_1 \leq 0.25$이고, 첫 행이 $h_1 \geq 0.25$를 주므로 $h_1 = 0.25$로 정해집니다.

제약식 한 줄을 solver 행으로 저장하기

전체 변수 순서는 $[x, h_0, h_1, h_2, h_3, y_A, y_B, y_C, z_1, z_2, z_3]$입니다. $h_1 - x + 0.25 z_1 \le 0$의 계수는 해당 세 열에만 들어갑니다.

```
import numpy as np
row = [-1, 0, 1, 0, 0, 0, 0, 0, 0.25, 0, 0]
row_lower = -np.inf
row_upper = 0.0
```

대입	행 내적	판정
$x=0.5, h_1=0.25, z_1=1$	$-0.5+0.25+0.25=0$	상한 0 만족
$x=0.5, h_1=0.30, z_1=1$	$-0.5+0.30+0.25=0.05$	상한 위반

출력 등식 $y_B - \sum W_j B\ h_j = b_B$는 해당 행의 하한과 상한을 모두 b_B로 둡니다. z_1 열은 정수 표시와 $[0,1]$ bounds를 함께 설정합니다. 계수 배열만 만들고 이 두 설정을 빼면 이진변수가 되지 않습니다.

SciPy LinearConstraint는 각 행을 `lb <= A @ v <= ub`로 해석합니다. 따라서 변수 순서와 계수의 부호를 같은 행에서 대조합니다.

MILP 표현의 성립 조건

고정 계수의 affine layer, ReLU, affine scaling을 위 식으로 연결하면 신경망 그래프를 MILP로 표현할 수 있습니다. 전체 문제의 목적함수와 나머지 제약도 선형이어야 합니다.

가변 유량 Q와 가변 농도의 곱 Qy, 비선형 속도식, sigmoid·tanh를 그대로 추가하면 현재 MILP 구조를 벗어납니다.

4주차의 투영도 계수 행렬이 고정이고 입력에 affine이면 마지막 선형 등식으로 연결할 수 있습니다.

고정 입력의 MILP·forward 일치 검사

체류시간 하나를 고정하고 제품 규격 제약을 뺀 뒤 신경망 등식만 풉니다. 계산된 모든 h와 y를 forward 결과와 비교

확인 입력 41개: 끝점, 모든 꺾임점, 꺾임점 양쪽, 난수 입력. 최대 차이: 1.25e-08

이 대조는 정식화 구현을 검사합니다. 공정 예측 오차는 별도의 기준값과 비교해야 합니다.

입력을 고정하면 원래 신경망의 출력은 이미 정해져 있습니다. 이때 solver가 다른 출력을 주면 가중치 배치, scaling, 부호 또는 bounds 구현을 의심할 수 있습니다. 제품 규격은 이 점의 신경망 계산과 별개의 제약이라 검사 중에는 제거합니다.

CSTR 설계 최적화 결과

항목	이번 실행
최소 τ [min]	5.353846
최소 V [L]	5.353846
예측 [A, B, C] [mol/L]	[0.584615, 0.400000, 0.215385]
이진변수 [z_1, z_2, z_3]	[1, 0, 0]
종료 상태	optimal, status = 0

B의 하한이 활성이고 C의 상한에는 여유가 있습니다. 독립적인 구간별 선형 계산에서도 같은 체류시간을 얻었습니다.

B의 여유는 $0.400000-0.40=0$이고 C의 여유는 $0.25-0.215385=0.034615$ mol/L입니다. τ를 조금 줄이면 B가 먼저 부족해집니다. 이 활성 제약이 부피를 더 줄이지 못하게 합니다.

제품 하한에 도달하는 보간 비율

4~6분 구간에서 B는 0.380952381에서 0.409090909로 증가합니다. 필요한 증가량을 구간 전체 증가량으로 나눕니다.

계산	값
필요한 B 증가량	$0.40-0.380952381=0.019047619$
구간 전체 증가량	$0.409090909-0.380952381=0.028138528$
구간 비율 α	$0.019047619/0.028138528\approx0.676923077$
체류시간	$4+2\alpha\approx5.353846154$ min
같은 α의 C	$0.152380952+\alpha(0.245454545-0.152380952)\approx0.215384615$

이전 구간의 B는 모두 0.40보다 작습니다. 이 첫 교점에서 $C\le0.25$와 비음수 조건도 만족하므로 최소 부피를 얻습니다. B만 따로 보간한 뒤 다른 τ의 C와 비교하면 같은 설계의 규격 검사가 되지 않습니다.

정식화의 동치성과 공정 예측 오차

MILP는 고정한 보간 모델을 정확히 표현합니다. 이 성질이 보간 모델의 공정 오차를 없애지는 않습니다.

선택된 $\tau = 5.353846$ min에서 기준 농도는 [0.579495, 0.404137, 0.216369] mol/L이고, 모델 예측과 차이가 있습니다.

해의 범위: 주어진 surrogate 문제의 최적해. 실제 공정에 대한 설계 판단에는 기준 공정의 규격 확인과 목적값 비교가 더 필요합니다.

LP 완화와 목적값 하한

$z \in [0, 1]$로 완화하면 최소화 문제의 목적값은 MILP 최적값의 하한입니다. 그러나 중간값과 출력이 원래 신경망 그래프를 벗어날 수 있습니다.

$L = -1$, $U = 2$, $a = 0$에서 $z = 1/3$, $h = 2/3$은 완화식을 만족하지만 실제 ReLU 출력은 0입니다.

이번 CSTR 예제에서는 LP와 MILP의 목적값이 같았습니다. 한 사례의 목적값 일치로 일반적인 그래프 동치가 성립하지는 않습니다.

최소화 문제에서 허용 집합을 넓히면 기존 최적점도 여전히 허용되므로 최적값이 더 커질 수 없습니다. 그래서 LP 값은 하한입니다. 다만 LP의 출력은 신경망 그래프 위에 없을 수 있어 그 입력을 그대로 설계안으로 채택하지 않습니다.

실습 실행과 제출

```
python -m pip install -r requirements.txt
python week05_lab.py --output-dir my-results
```

1. 한 뉴런의 bounds와 행렬 부호를 손으로 확인
2. `results.json`의 forward 오차와 종료 상태를 확인
3. `milp_matrix.csv`의 행 하나를 수식으로 되돌려 적습니다.
4. B 하한을 0.40에서 0.41로 바꾸고 가능해가 남는지 설명

읽기자료 · 실습 묶음

출처와 지정 읽기

Boyd & Vandenberghe, Convex Optimization, 4.1절: 변수·목적함수·가능집합.

Anderson et al., Strong mixed-integer programming formulations for trained neural networks, 2.1절: ReLU Big-M 표현.

SciPy milp 공식 문서: bounds, integrality, LinearConstraint와 반환 상태.