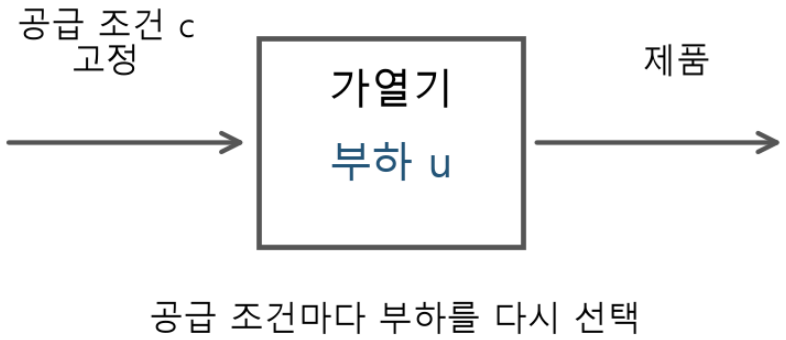
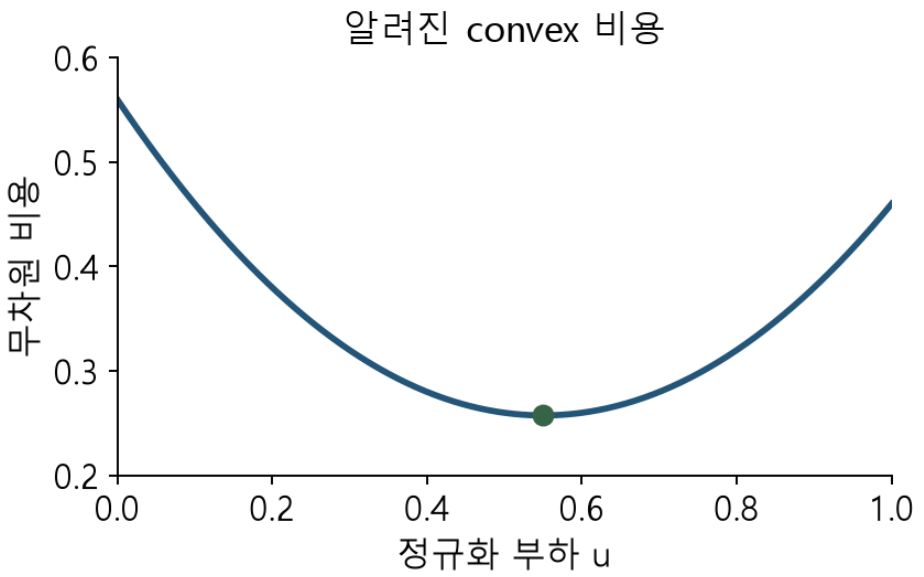
7주차: 최적화 활용을 위한 모델 구조 설계

유틸리티 비용 모델과 반복 운전 선택

공급 조건이 바뀔 때마다 가열 부하를 다시 정해야 합니다. 비용 곡선을 신경망으로 학습하면 운전 조건을 찾는 계산도 모델 구조에 따라 달라집니다.



공급 조건마다 부하를 다시 선택



비용의 convex 구조를 유지하면서 부하를 선택하는 방법을 다뤄봅니다.

모델 선택과 정식화의 연결

5·6주차에서는 학습된 일반 ReLU 네트워크를 MILP로 정확히 표현했습니다. 이번 주에는 학습 단계에서 입력에 대한 convexity를 부여합니다.

알려진 convex 비용을 먼저 다룹니다. 반응기 농도나 수율이 convex라는 가정은 하지 않습니다.

구조 제약을 추가하면 표현할 수 있는 함수가 줄어들므로, 예측 오차와 의사결정 결과를 함께 확인합니다.

학습 목표와 제출 항목

1. 결정변수와 고정 context를 구분하고 ICNN의 부호 조건을 확인합니다.
2. ReLU epigraph LP의 목적값 동치성과 적용 조건을 설명합니다.
3. 부등식 방향·음의 가중치·가변 context에 대한 반례를 계산
4. 학습한 작은 ICNN의 LP 해를 forward 계산과 기준 함수로 검증

결정변수와 고정 context

정규화한 가열 부하 u를 선택하고, 공급 조성·외기 조건 c는 측정한 값으로 고정합니다. 모델 매개변수 θ도 최적화 전에 고정합니다.

$$\min_{u \in U} f_\theta(u; c)$$
$$U = \{u : Gu \le r\}, c = c_{obs}$$

여기서 convexity는 u에 관한 성질입니다. c나 θ에 대해서도 convex라고 결론 내릴 수는 없습니다.

c는 현재 관측한 공급 조건이고 u는 지금 선택할 부하입니다. 공급 조건이 바뀌면 같은 u의 비용 곡선도 바뀔 수 있어 c를 다시 고정하고 풉니다. 학습에서 바꾸는 θ와 운전에서 선택하는 u를 구분합니다.

Convex 함수와 sublevel set

두 입력 사이의 함수값이 두 함수값을 잇는 선분보다 높지 않으면 convex입니다.

$$f(\lambda u + (1-\lambda)v) \leq \lambda f(u) + (1-\lambda) f(v)$$
$$0 \leq \lambda \leq 1$$

Convex f에 대한 f(u) ≤ b는 convex 영역을 만듭니다. f(u) ≥ b는 일반적으로 그렇지 않습니다. 이 방향 차이가 LP 사용 조건에 반영됩니다.

Boyd and Vandenberghe, Convex Optimization, §§3.1-3.2 and 4.1-4.3

두 입력이 각각 비용 상한 b를 만족하면 그 사이 혼합 입력의 비용은 두 비용의 가중평균 이하이므로 b 이하입니다. 이 계산이 sublevel set의 convexity를 설명합니다. 비용 하한에는 같은 부등식 논리를 적용할 수 없습니다.

ICNN의 입력 연결과 부호 조건

$$h_0 = 0$$
$$h_k = \mathrm{ReLU}\left(W_k h_{k-1} + A_k u + b_k\right)$$
$$f(u) = a^T u + d + w^T h_K$$

W와 w의 모든 원소를 0 이상으로 제한합니다. 입력에서 직접 들어오는 A와 a, bias의 부호는 제한하지 않습니다.

ReLU는 convex이고 비감소합니다. 이 조건으로 각 층과 출력의 입력 convexity를 유지합니다.

Amos, Xu and Kolter, Input Convex Neural Networks, ICML 2017, pp. 146-155

이전 층의 convex 곡선을 양수 가중치로 더하면 아래로 굽는 부분이 새로 생기지 않습니다. 반면 그 곡선에 음수 가중치를 곱하면 위아래가 뒤집힙니다. 직접 입력의 affine 항은 기울기가 음수여도 굽음이 없어 별도 취급합니다.

층별 convexity 유도

첫 층은 affine 함수와 0의 최댓값이므로 convex입니다. 이전 층의 성분들이 convex라고 가정합니다.

$W \geq 0$이면 그 성분의 가중합도 convex이고, affine 입력 연결을 더해도 유지됩니다. 여기에 convex·비감소 ReLU를 적용합니다.

출력의 $w \geq 0$도 같은 역할을 합니다. 입력 연결의 음의 기울기는 affine 항이므로 이 논증을 깨지 않습니다.

귀납 단계에서 필요한 것은 이전 층의 각 성분이 convex라는 조건입니다. W의 각 원소가 비음수라면 성분별 convexity 부등식에 곱해 더해도 방향이 유지됩니다. 그 다음 비감소 함수 ReLU를 적용해도 대소관계를 뒤집지 않습니다.

음의 가중치와 입력 convexity

은닉 출력에 음의 가중치를 허용하면 위의 충분조건이 사라집니다.

$$f(u) = -\mathrm{ReLU}(u),\ -1 \le u \le 1$$
$$f(0) = 0 > \left(f(-1) + f(1)\right)/2 = -.5$$

음의 입력 연결을 금지할 필요는 없습니다. 예를 들어 ReLU(−u + 0.4)는 여전히 convex입니다. 제한하는 가중치가 어느 연결에 있는지 확인합니다.

학습 중 부호 제약

Gradient 갱신 뒤 W를 max(W, 0)으로 투영하거나, 자유 매개변수 V에서 W = softplus(V)를 계산할 수 있습니다. 출력 가중치 w에도 적용합니다.

여기서 softplus는 가중치를 양수로 만드는 매개화입니다. 추론할 때 은닉 활성화는 ReLU로 유지합니다.

학습 후 확인: 가중치의 부호. 일반적인 깊은 ICNN의 매개변수 학습: 비선형·비볼록 문제

가중치 하나가 갱신 뒤 −0.03이면 clipping은 0으로 바꿉니다. 다른 가중치가 0.2이면 그대로 둡니다. 이는 함수가 입력에 대해 convex하도록 제약하는 것이며, 가중치 전체를 찾는 학습 문제의 convexity를 보장하지는 않습니다.

ReLU 그래프와 epigraph

$$h = \mathrm{ReLU}(s) = \max(0, s)$$
$$h \geq 0, h \geq s$$

두 부등식은 ReLU 위쪽 영역을 표현합니다. s = 0에서 h = 10도 허용하므로 그래프 등식 자체와 같지는 않습니다.

이 하한 제약을 LP에 사용하려면 보조변수를 실제 출력으로 줄일 수 있어야 하며, 목적함수와 나머지 제약도 그 방향을 허용해야 합니다.

고정 입력에서 h를 실제 ReLU값보다 크게 두어도 두 하한은 만족합니다. 최소화 목적이 h를 줄이는 방향이고 다른 제약도 그 감소를 허용할 때만 실제 순전파 값으로 내려도 손해가 없습니다. 이 조건이 등식 그래프 대신 epigraph를 사용하는 근거입니다.

ReLU ICNN의 확장 LP

제약 적용 범위: 유한한 ReLU 네트워크, 고정된 가중치

$$h_k \geq 0$$
$$h_k \geq W_k h_{k-1} + A_k u + b_k$$
$$t \geq a^T u + d + w^T h_K$$

입력 제약 $Gu \leq r$를 추가하고 t를 최소화합니다. $W, w \geq 0$이고 나머지 식도 선형이면 이진변수 없이 LP가 됩니다.

활성화를 softplus로 바꾸면 convexity는 유지할 수 있어도 이 유한한 선형 제약식으로 정확히 표현되지는 않습니다.

t는 비용의 상한을 나타내는 보조 변수입니다. 모든 허용된 점에서 $t \geq f(u)$이고 실제 순전파 값을 넣으면 $t = f(u)$가 가능하므로, t 최소화는 원래 비용 최소화와 같은 최적값을 얻습니다.

Forward 복원과 목적값 동치성

u를 고정하면 첫 층의 허용 h는 실제 ReLU 출력 이상입니다. $W \geq 0$을 사용해 층별로 반복하면 모든 h가 실제 출력 이상이 됩니다. 따라서 $t \geq f(u)$입니다.

반대로 실제 forward 값을 h에 넣고 $t = f(u)$로 두면 모든 제약을 만족합니다. 이 두 포함관계가 epigraph의 정확성을 보입니다.

min t의 최적값과 선택한 u의 목적값이 일치합니다. 출력에 영향을 주지 않는 보조변수는 최적해에서도 slack을 가질 수 있습니다.

두 층의 보조변수를 forward로 복원하기

$h_1=\mathrm{ReLU}(u-0.4)$, $h_2=\mathrm{ReLU}(2h_1-0.1)$, $f=0.2-u+3h_2$인 ICNN을 봅니다. 은닉 연결 2와 출력 연결 3이 양수입니다. 입력 $u=0.6$을 고정합니다.

상태	h_1	h_2	출력식의 값
실제 forward	0.2	0.3	0.5
느슨한 epigraph 가능점	0.4	0.7	1.7
첫 층만 실제값으로 내림	0.2	0.7	1.7
둘째 층도 실제값으로 내림	0.2	0.3	0.5

h_1을 낮추면 $2h_1-0.1$도 낮아져 기존 h_2의 하한 제약을 깨뜨리지 않습니다. 이어 h_2를 낮추면 양의 출력 계수 때문에 목적값도 내려갑니다. 이 순서가 층별 복원 논증입니다.

출력 계수가 -3이면 h_2를 높일수록 목적값이 낮아져 이 논리가 성립하지 않습니다. 외부에서 $h_2\ge 0.7$을 요구해도 실제값 0.3으로 복원할 수 없으므로, 부호와 외부 제약을 함께 확인합니다.

출력 사용 방식에 따른 적용 범위

사용 방식	판단
min f(u)	epigraph 목적함수로 표현
$f(u) \le b$	$t \le b$를 추가하면 정확한 입력 feasible set
$f(u) \ge b$	$t \ge b$만으로는 실제 출력 하한을 강제하지 못함
y = f(u)	출력 그래프 등식은 일반적으로 비convex
max f(u)	convex 함수 최대화에 같은 LP 논증을 적용할 수 없음

다른 제약이 h나 t를 큰 쪽으로 밀어 올리는 경우도 다시 확인합니다.

비용 상한은 $f(u) \le t \le b$를 만들기 때문에 원래 상한도 만족합니다. 비용 하한 $t \ge b$는 f(u)가 작아도 t만 키워 만족할 수 있습니다. 동일한 모델에서도 출력이 목적식인지 상한인지 하한인지에 따라 정식화가 달라집니다.

손 계산용 두 hinge 비용 모델

$$f(u) = .4 - .6u + h_1 + .5h_2$$
$$h_1 = \mathrm{ReLU}(u - .4), h_2 = \mathrm{ReLU}(u - .7)$$
$$0 \le u \le 1$$

직접 수치 검증을 위한 무차원 비용 예제. 구간별 기울기: −0.6 → 0.4 → 0.9

최소점: 첫 번째 기울기 전환점 u = 0.4. 최솟값: f(0.4) = 0.16

u=0.4 전에는 두 ReLU가 모두 0이라 기울기는 −0.6입니다. 첫 ReLU가 켜지면 기울기 1이 더해져 0.4가 되고, 두 번째가 켜지면 0.5가 더해져 0.9가 됩니다. 부하를 늘릴 때 비용이 내려가다가 올라가는 첫 절점이 최소점입니다.

손 계산 모델의 LP와 비용 상한

$$\min_{u,h} .4 - .6u + h_1 + .5h_2$$
$$h_1 \geq 0, h_1 \geq u - .4$$
$$h_2 \geq 0, h_2 \geq u - .7, 0 \leq u \leq 1$$

비용 상한 0.25를 추가하면 feasible 입력은 [0.25, 0.625]입니다. 같은 LP의 출력 affine 식에 상한을 걸어 이 구간을 얻습니다.

실습 검증: 최소 목적값 0.16, forward 일치

두 절점 사이의 비용과 가능 구간

각 구간에서 켜진 ReLU만 남겨 비용식을 정리합니다.

u 구간	켜진 항	비용식	f(u)≤0.25의 조건
0~0.4	없음	0.4−0.6u	u≥0.25
0.4~0.7	u−0.4	0.4u	u≤0.625
0.7~1	두 항 모두	0.9u−0.35	u≤2/3, 해당 구간에 없음

가능 구간은 [0.25,0.625]입니다. 최소점 u=0.4에서 두 ReLU가 0이므로 f=0.4−0.6×0.4=0.16입니다.

예를 들어 u=0.6이면 $h_1=0.2, h_2=0$이고 f=0.4−0.36+0.2=0.24입니다. 중간 구간의 0.4u=0.24와 일치하므로 구간식과 원래 신경망을 서로 확인할 수 있습니다.

LP 행렬의 열과 부등식

변수 순서를 $[u,h_1,h_2]$로 고정합니다. linprog의 A_ub는 각 행과 변수 벡터를 내적한 값이 b_ub 이하가 되도록 해석합니다.

코드의 값	해당 수식	역할
목적 계수 $[-0.6,1,0.5]$	$-0.6u+h_1+0.5h_2$	상수 0.4를 뺀 비용
행 $[1,-1,0]$, 우변 0.4	$u-h_1\le 0.4$	$h_1\ge u-0.4$
행 $[1,0,-1]$, 우변 0.7	$u-h_2\le 0.7$	$h_2\ge u-0.7$
h의 bounds (0,None)	$h_1,h_2\ge 0$	ReLU의 다른 하한

해 $[0.4,0,0]$을 대입하면 행 값은 $[0.4,0.4]$로 우변 이하입니다. solver 목적값 -0.24에 상수 0.4를 더해 0.16을 보고합니다. 원래 forward에서 계산한 비용도 0.16이어야 합니다.

양의 h 계수 때문에 이 목적식은 h를 가능한 작게 만듭니다. 목적 계수의 부호를 바꾸면 이 복원 논리를 다시 검사해야 합니다.

출력 하한 제약의 반례

$0 \le u \le 1$에서 $f(u) = \mathrm{ReLU}(u)$라 두고 $f(u) \ge 0.8$을 만족하며 u를 최소화. 실제 최적점은 $u = 0.8$

$$h \ge u, h \ge 0, h \ge .8$$

이 완화에서는 $u = 0$, $h = 0.8$이 허용됩니다. 비용을 줄이려고 보조변수를 부풀린 결과입니다. 작은 MIP gap이나 LP optimal 종료 상태로 이 모델링 오류를 발견할 수는 없습니다.

$u=0, h=0.8$을 두면 세 선형 부등식은 모두 만족하지만 실제 $f(0)=0$입니다. 출력 하한을 잘못 표현한 모델은 정확히 풀어도 잘못된 입력을 반환합니다. solver 상태 확인 뒤 순전파 검사가 필요한 이유입니다.

출력 보상과 그래프 등식의 반례

같은 ReLU에서 $-h$를 최소화하면 h의 상한이 없는 epigraph LP는 unbounded입니다. 실제 $f(u)$는 $[0, 1]$에 있어 $\min -f(u) = -1$입니다.

또한 $f(u) = |u|$의 그래프 위에 있는 $(-1, 1)$, $(1, 1)$의 중점 $(0, 1)$은 그래프에 없습니다.

비용 epigraph의 동치성을 농도 등식이나 출력 보상에 그대로 옮기면 문제가 달라집니다.

PICNN과 고정 context

PICNN은 일부 입력에만 convexity를 부여합니다. 공급 조건 c를 처리하는 경로는 비선형이어도, c를 고정한 뒤 결정변수 u의 경로가 필요한 부호 조건을 만족해야 합니다.

다음 페이지는 context가 층의 계수를 결정하는 충분조건 형태를 사용합니다. 원 논문의 gate 구조도 고정 context에서 이 관점으로 해석할 수 있습니다.

Amos, Xu and Kolter, Input Convex Neural Networks, ICML 2017, pp. 146-155

조건부 계수와 LP 구성

$$h_k = \mathrm{ReLU}\left(W_k(c)\, h_{k-1} + A_k(c)\, u + b_k(c) \right)$$
$$W_k(c) \geq 0, w(c) \geq 0$$

관측 c를 context 경로에 한 번 넣어 계수를 계산한 뒤 고정합니다. 나머지 결정변수와 제약이 선형이면 ReLU epigraph를 LP로 구성합니다.

c도 최적화하면 W(c)h나 A(c)u가 비선형 결합이 될 수 있습니다. 고정 context의 결론이 공동 최적화로 이어지지는 않습니다.

관측 c에서 계산된 W(c), A(c), b(c)는 숫자입니다. 이를 고정하면 최적화 변수 u와 h에 대해 모든 식이 affine입니다. c도 변수로 남기면 “숫자 계수×변수”가 “변수의 함수×변수”가 되어 같은 LP가 아닙니다.

조건부 convexity와 공동 convexity

$$f(u;c) = |u-c| - .1c^2$$
$$0 \le u \le 1, 0 \le c \le 1$$

고정 c에서 최적점은 u = c. 실습은 c = 0.2, 0.5, 0.8에 대한 LP를 각각 풉니다.

u = c인 선 위에서는 f = −0.1c²입니다. 끝점 함수값의 평균은 −0.05인데 중점 값은 −0.025이므로 공동 convexity 조건을 위반합니다.

고정 c=0.2에서는 $|u-0.2|$를 보조변수 t의 두 하한 t≥u−0.2, t≥0.2−u로 표현하고 t−0.004를 최소화합니다. u=0.2,t=0에서 최솟값 −0.004입니다. c가 변수가 되면 목적식의 −0.1c²가 상수가 아니므로 이 LP를 그대로 사용할 수 없습니다.

알려진 convex 함수를 이용한 학습

$$q(u) = (u - .6)^2 + .1u + .2$$
$$f(u) = b + au + \sum_{j=1}^{9} w_j \mathrm{ReLU}(u - \kappa_j)$$
$$\kappa_j = .1j, w_j \geq 0$$

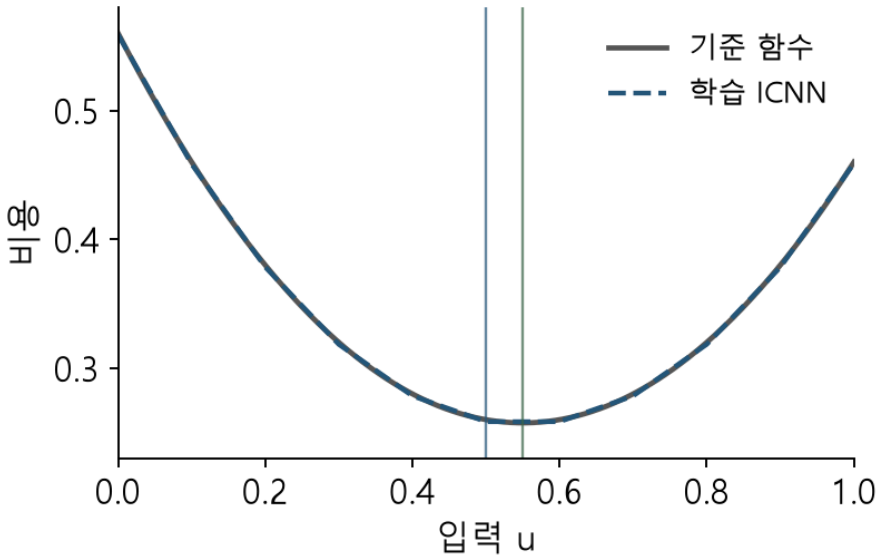
41개 균등 입력에서 계수를 학습하고 별도의 200개 중간 지점에서 평가합니다. Knot을 고정했으므로 이 실습의 학습은 bound가 있는 선형 least squares입니다.

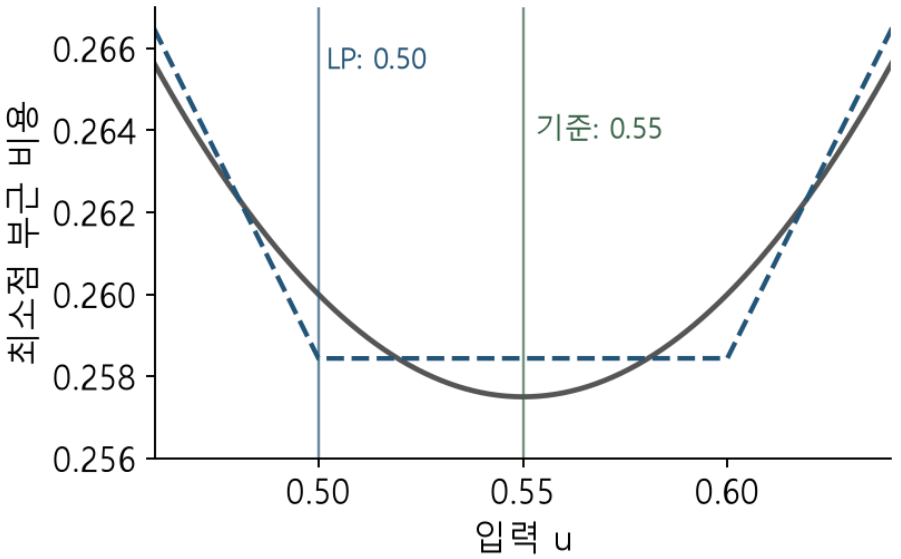
고정 knot을 사용하는 이 얕은 ICNN은 convex 학습이 가능한 특수한 경우입니다. 깊은 ICNN의 일반적인 학습에는 이 결론을 적용할 수 없습니다.

절점 κ가 고정이면 각 입력의 ReLU 특성을 미리 계산할 수 있습니다. 계수 b,a,w만 미지수이므로 제한된 선형 최소제곱입니다. w≥0은 절점에서 기울기가 감소하지 않게 해 학습된 곡선의 convexity를 유지합니다.

예를 들어 u=0.25의 특성 행은 [1,0.25,0.15,0.05,0,0,0,0,0,0,0]입니다. 절점 0.1과 0.2만 지났으므로 두 hinge만 양수입니다. b,a 두 계수는 음수도 허용하고 나머지 9개에는 하한 0을 주어 최소제곱을 풉니다.

학습 곡선과 선택한 부하





독립 평가 RMSE: 0.000760. LP 선택: u = 0.50. 기준 함수의 최소점: u = 0.55

두 점 사이의 기준 비용 차이는 0.0025입니다. 예측 곡선이 가까워도 최소점 위치는 달라질 수 있습니다.

최소점 근처에서 기준 곡선이 평평하면 입력이 조금 이동해도 비용 차이는 작습니다. 이 예의 입력 차이 0.05와 비용 차이 0.0025는 서로 다른 평가량입니다. 그래프가 겹쳐 보인다는 이유만으로 최적 입력까지 같다고 판단하지 않습니다.

예측 오차와 선택한 입력의 손실

기준 비용 $q(u)=(u-0.6)^2+0.1u+0.2$를 완전제곱으로 정리하면 $q(u)=(u-0.55)^2+0.2575$입니다. 최소점에서 벗어난 거리의 제곱이 비용 증가량입니다.

입력	기준 비용 계산	비용
기준 최소 $u=0.55$	$0^2+0.2575$	0.2575
모델 선택 $u=0.50$	$(-0.05)^2+0.2575$	0.2600
차이	$0.2600-0.2575$	0.0025

모델이 선택한 점의 예측 비용 0.258437과 기준 비용 0.260000의 차이 0.001563은 그 점의 예측 오차입니다. 기준 최소값과의 차이 0.0025는 선택 때문에 생긴 비용 손실입니다.

같은 숫자를 비교하는 기준이 다릅니다. 정확도 표와 의사결정 평가 표에서 각각 무엇을 빼는지 명시합니다.

LP 결과의 검증 항목

1. 입력 domain, context, scaling, 모델 가중치를 저장
2. W와 w의 부호, 목적함수 방향, 모든 출력 사용 위치를 확인
3. 선택한 u를 원래 네트워크에 넣어 LP 목적값과 비교
4. 기준 함수의 목적값·제약을 따로 계산

학습 모델의 LP 목적값: 0.258437. 기준 비용: 0.260000. 두 값의 차이: 서로게이트 오차

모델 구조의 선택 기준

대상과 조건	검토할 접근
알려진 convex 비용의 최소화	ICNN 또는 고정-context PICNN
ReLU·선형 출력·선형 외부 제약	조건을 확인한 epigraph LP
농도 등식·복잡한 비convex 응답	일반 서로게이트와 적절한 NLP/MILP
Convexity 근거가 약한 데이터	구조 제약의 예측 편향과 결정 오차 비교

실습 실행과 확인 결과

```
python -m pip install -r requirements.txt
python week07_lab.py --output-dir my-results
```

코드 묶음을 풀어 실행합니다. results.json, 학습 계수, training.csv와 heldout.csv를 저장

손 계산 최적값 0.16, 학습 모델의 부호 조건, LP·forward 일치, 출력 하한 반례와 unbounded 반례를 자동 검사합니다. Notebook에는 같은 함수를 단계별로 실행하는 셀이 있습니다.

과제: 동치성 조건과 의사결정 오차

1. 두 hinge 모델에 비용 상한 0.25를 추가하고 feasible 구간의 양 끝을 계산
2. 출력 하한 반례를 실행하고 잘못 허용된 u와 실제 출력을 기록
3. 학습 모델의 knot 간격을 0.1에서 0.05로 바꿔 예측 오차와 기준 비용 손실을 비교
4. c를 고정해야 하는 이유와 현재 모델에 허용된 출력 사용 방식을 설명합니다.

해설 자료와 원문

7주차 해설과 연습에서 epigraph 동치성 증명, LP 행렬, 부호 제약 학습과 반례를 계산

Amos, Xu and Kolter, Input Convex Neural Networks, ICML 2017, pp. 146-155

Boyd and Vandenberghe, Convex Optimization, §§3.1-3.2 and 4.1-4.3

SciPy bounded least squares · SciPy HiGHS LP interface