Week 7: Designing Model Structure for Optimization | Worked Companion

Mathematical scope of the heating-load decision

The cost used to choose u in [0, 1] is a dimensionless calculation example. It is not a cost law derived from the thermodynamics of a real process.

Consider structural restrictions when there is evidence that the fixed-context cost is convex. Do not transfer this assumption automatically to concentration, yield, or recovery.

Check network convexity, formulation equivalence, and decision quality on the reference function separately.

The induction step in the convexity proof

Assume each component of h is convex and W ≥ 0. Multiplying the componentwise convexity inequality by W preserves its direction for $0 \le \lambda \le 1$.

$$Wh(\lambda u + (1-\lambda)v) \le \lambda Wh(u) + (1-\lambda)Wh(v)$$

The affine term Au + b satisfies equality. Add it, then compose with convex, nondecreasing ReLU to obtain a convex next layer. Apply the same reasoning to the nonnegative output sum.

Two inclusions for the extended epigraph

Let h^* denote actual forward activations. A feasible lifted h satisfies $h \geq h^*$ at the first layer. Substituting this relation and $W \geq 0$ into the next-layer constraints propagates the inequality.

Thus every lifted feasible point has $t \geq f(u)$. Conversely, for any (u, t) with $t \geq f(u)$, choosing $h = h^*$ satisfies the extended constraints.

Projection onto (u, t) is exactly the epigraph. This proof does not require every optimal auxiliary variable to be tight.

Restoring two auxiliary layers to forward values

Consider $h_1=\mathrm{ReLU}(u-0.4)$, $h_2=\mathrm{ReLU}(2h_1-0.1)$, and $f=0.2-u+3h_2$. Hidden weight 2 and output weight 3 are positive. Fix $u=0.6$.

State	h_1	h_2	Output expression
Actual forward	0.2	0.3	0.5
Loose epigraph point	0.4	0.7	1.7
Restore first layer only	0.2	0.7	1.7
Restore second layer	0.2	0.3	0.5

Lowering h_1 lowers $2h_1-0.1$, so the existing h_2 still satisfies its lower bound. Lowering h_2 then decreases the objective because its output coefficient is positive. This is the layer-by-layer restoration argument.

With output coefficient -3, raising h_2 would improve the objective and the argument would fail. An external requirement $h_2\geq 0.7$ would also prevent restoration to 0.3. Both weight signs and external constraints matter.

Objective monotonicity and auxiliary recovery

If the objective is nondecreasing in t and all other constraints permit reducing t and h to forward values, recovery preserves the optimal value.

This condition holds directly for min t. An upper bound $t \leq b$ also permits reduction. Lower bounds $h \geq b$, output rewards, and state constraints coupled through output equalities require separate inspection.

An activation with zero weight along all output paths can remain slack without affecting the objective. Interpret predictions by reevaluating the network at u.

Piecewise calculation of the two-hinge model

u interval	f(u)	Slope
[0, 0.4]	$0.4 - 0.6u$	-0.6
[0.4, 0.7]	$0.4u$	0.4
[0.7, 1]	$0.9u - 0.35$	0.9

Continuity gives $f(0.4) = 0.16$ and $f(0.7) = 0.28$. The slope switches from negative to positive at the unique minimum 0.4.

Solving $f \le 0.25$ on each interval leaves $u \ge 0.25$ and $u \le 0.625$. The third interval contributes no feasible points.

Cost and feasibility between two knots

Keep only active ReLUs in each interval to simplify the cost.

u interval	Active terms	Cost	Condition for $f(u) \le 0.25$
0–0.4	None	$0.4-0.6u$	$u \ge 0.25$
0.4–0.7	$u-0.4$	$0.4u$	$u \le 0.625$
0.7–1	Both	$0.9u-0.35$	$u \le 2/3$, outside this interval

The feasible interval is $[0.25, 0.625]$. At minimum $u=0.4$, both ReLUs are zero and $f=0.4-0.6\times0.4=0.16$.

For example, $u=0.6$ gives $h_1=0.2, h_2=0$ and $f=0.4-0.36+0.2=0.24$. This matches $0.4u=0.24$ in the middle interval, checking the piecewise expression against the original network.

LP matrices passed to SciPy

The variable order is [u, h_1, h_2]. Add the objective constant 0.4 after solving.

```
result = linprog(
    [-0.6, 1.0, 0.5],
    A_ub=[[1, -1, 0], [1, 0, -1]],
    b_ub=[0.4, 0.7],
    bounds=[(0, 1), (0, None), (0, None)],
    method='highs')
value = result.fun + 0.4
```

The first matrix row is $u - h_1 \leq 0.4$. Check the returned status before using result.x.

LP matrix columns and inequalities

Fix variable order $[u,h_1,h_2]$. Each row of linprog's A_ub is dotted with the variable vector and bounded above by the matching b_ub entry.

Code value	Expression	Role
Objective $[-0.6,1,0.5]$	$-0.6u+h_1+0.5h_2$	Cost without constant 0.4
Row $[1,-1,0]$, rhs 0.4	$u-h_1\le 0.4$	$h_1\ge u-0.4$
Row $[1,0,-1]$, rhs 0.7	$u-h_2\le 0.7$	$h_2\ge u-0.7$
h bounds (0,None)	$h_1,h_2\ge 0$	Other ReLU lower bounds

At solution $[0.4,0,0]$, row values $[0.4,0.4]$ satisfy their upper bounds. Add constant 0.4 to solver objective -0.24 to report 0.16. The original forward cost must also be 0.16.

Positive h coefficients make this objective minimize h. Changing their signs requires rechecking the recovery argument.

Complete solution of the inequality-direction counterexample

The true problem is min u with $0 \leq u \leq 1$ and $\mathrm{ReLU}(u) \geq 0.8$. $\mathrm{ReLU}(u) = u$ on this interval, so the optimum is 0.8.

The relaxation uses $h \geq u$, $h \geq 0$, $h \geq 0.8$. Every $u \in [0, 1]$ can be paired with $h = \max(u, 0.8)$, expanding the projected feasible set to the full interval.

The optimum is 0. The solver correctly solves the revised problem but does not recover the original answer.

The fixed-knot ICNN fitting matrix

$$\Phi_i = \left[1, u_i, \mathrm{ReLU}\left(u_i - \kappa_1\right), \ \ldots, \mathrm{ReLU}\left(u_i - \kappa_9\right)\right]$$

$$\beta = \left[b, a, w_1, \ \ldots, w_9\right]^T$$

$$\min_\beta \left\| \Phi\beta - q \right\|_2^{\,2}, w \geq 0$$

b and a are free; only w is nonnegative. At fixed Φ, the Hessian $2\Phi^T\Phi$ is positive semidefinite, so this is a convex quadratic fit.

If knots or deep-layer weights are also learned, Φ depends on the parameters and this argument no longer applies directly.

At u=0.25, the feature row is [1,0.25,0.15,0.05,0,0,0,0,0,0,0]. Only knots 0.1 and 0.2 have been crossed. Fit least squares with b and a unrestricted and lower bounds of zero on the remaining nine weights.

Reference minimizer and decision loss

$$q(u) = (u - .6)^2 + .1u + .2$$
$$q'(u) = 2(u - .6) + .1 = 0 \Rightarrow u^* = .55$$
$$q(.5) - q(.55) = .2600 - .2575 = .0025$$

Near a flat minimum, a small prediction error can shift the selected input. Input error 0.05 and cost loss 0.0025 are distinct metrics.

The fitted slope on [0.5, 0.6] is approximately 0.0000233, nearly flat. Check solver tolerances alongside the selected point.

Prediction error and loss from the selected input

Complete the square in $q(u)=(u-0.6)^2+0.1u+0.2$ to obtain $q(u)=(u-0.55)^2+0.2575$. Squared distance from the minimizer equals the cost increase.

Input	Reference calculation	Cost
Reference minimum $u=0.55$	$0^2+0.2575$	0.2575
Model choice $u=0.50$	$(-0.05)^2+0.2575$	0.2600
Difference	$0.2600-0.2575$	0.0025

The difference 0.001563 between predicted cost 0.258437 and reference cost 0.260000 is prediction error at the selected point. Difference 0.0025 from the reference minimum is cost lost because of the selection.

These comparisons use different baselines. State the quantities subtracted in both accuracy reports and decision evaluations.

Freezing PICNN coefficients and context gates

Suppose a context path r(c) produces a gate s(c) ≥ 0. At fixed c, nonnegative W diag(s(c)) preserves a nonnegative weighted sum of previous convex activations.

A gate multiplying u is a fixed number and can be absorbed into an affine input connection with unrestricted sign.

The lab uses $|u - c| - 0.1c^2$ to check the fixed-context LP and a counterexample to joint convexity. It does not train the paper's full PICNN architecture.

For fixed c=0.2, represent $|u - 0.2|$ with $t \geq u - 0.2$ and $t \geq 0.2 - u$, and minimize $t - 0.004$. The minimum is -0.004 at $u=0.2, t=0$. If c becomes a variable, $-0.1c^2$ is no longer a constant, so this LP no longer applies unchanged.

Scaling and physical constraints

Composing convex f with a fixed affine input map u = Sx + r preserves convexity in x. Bounds, units, and context must still agree between the formulation and forward code.

A positive output scale preserves convexity. A negative scale makes a convex function concave, requiring corresponding changes in objective and inequality interpretation.

A concentration-balance projection can mix components with negative coefficients. The Week 4 equality guarantee alone establishes neither componentwise convexity nor this LP representation.

Exercises

1. Determine whether $f(u) = \mathrm{ReLU}(-2u + 1)$ is input-convex and justify it.
2. Find the feasible interval for $f \leq 0.20$ in the two-hinge model.
3. Does adding $h \leq 1$ to the linear inequalities for $h \geq \mathrm{ReLU}(u)$ impose equality? Find a counterexample.
4. Test joint convexity of $f(u;c) = |u - c| - 0.1c^2$ using a midpoint.
5. Does a lower independent RMSE necessarily reduce the error in the selected input? Explain.

Short answer checks

1. Convex: it is the maximum of the affine function $-2u + 1$ and zero.
2. $[1/3, 1/2]$. Solve $0.4 - 0.6u \leq 0.2$ on the left and $0.4u \leq 0.2$ in the middle.
3. $u = 0$, $h = 0.8$ is a counterexample. An upper bound alone does not recover the graph.
4. Endpoint values at $(u, c) = (0, 0)$ and $(1, 1)$ average -0.05; the midpoint value is -0.025.
5. No guarantee: local shape, gradients, and constraint-boundary errors influence the decision.

Reproduction record and review criteria

The lab was run with NumPy 2.5.4 and SciPy 1.18.1. Timing depends on the execution environment.

Include fitted coefficients, data splits, sign checks, LP status and forward error, and reference decision loss. With multiple minimizers, distinguish objective-value equivalence from input nonuniqueness.

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