Week 5 Reading: ReLU Graphs and MILP Derivations

Optimization formulation for reactor design

At a fixed flow of 1 L/min, find the minimum volume satisfying the product specifications. Since $V = Q\tau$, residence time τ expresses the design choice.

The fixed context is 330 K and an A feed concentration of 1.2 mol/L. Outputs are A, B, and C concentrations; require $B \geq 0.40$ and $C \leq 0.25$ mol/L.

We first define the process problem, translate the predictor into constraints, and then verify the computed solution.

Feasible set of the CSTR design problem

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

The τ range defines both the model domain and the starting point for bound calculation. Concentration inequalities encode product requirements, while y = fθ(τ) encodes the predictor.

The objective V is in L; y is in mol/L. τ and V have the same numerical value only because Q = 1 L/min.

Interpolation data and model construction

τ [min]	A [mol/L]	B [mol/L]	C [mol/L]
2	0.857143	0.285714	0.057143
4	0.666667	0.380952	0.152381
6	0.545455	0.409091	0.245455
8	0.461538	0.410256	0.328205
10	0.400000	0.400000	0.400000

Interpolate linearly between adjacent rows. Each segment is a convex combination of its endpoint concentrations, preserving nonnegativity and the total 1.2 within the domain.

The table displays six decimals; computations use full-precision coefficients.

Derivation of a ReLU representation of linear interpolation

Let c_i be the concentration vector at knot i and compute each segment slope s_i. Start with the first slope and add its change at each breakpoint.

$$s_i = \frac{c_{i+1} - c_i}{\tau_{i+1} - \tau_i}$$

$$y(\tau) = c_0 + s_0(\tau - \tau_0) + \sum_{i=1}^{3} (s_i - s_{i-1}) \max(0, \tau - \tau_i)$$

Within segment i, active hinge slopes telescope to $s_0 + (s_1 - s_0) + \cdots + (s_i - s_{i-1}) = s_i$. Continuity at the knots and the correct initial value identify the desired interpolant.

All weights after input scaling

Use $x = (\tau - 2)/8$, $h_j = \max(0, x - t_j)$, and $t = [0, 0.25, 0.50, 0.75]$. In $y = b + W^T h$, rows of W index neurons and columns index components.

j	W_jA	W_jB	W_jC
0	-0.761904762	0.380952381	0.380952381
1	0.277056277	-0.268398268	-0.008658009
2	0.149184149	-0.107892108	-0.041292041
3	0.089510490	-0.045687646	-0.043822844

b = [0.857142857, 0.285714286, 0.057142857] mol/L. W is also in mol/L, while h is dimensionless.

Full-precision coefficients are stored in `frozen_network.json`. Reconstructing them from the rounded table can introduce small output differences.

From interpolation slopes to ReLU weights

The first three B knot values at $\tau=2,4,6$ are $0.285714286, 0.380952381, 0.409090909$. Since $x=(\tau-2)/8$, adjacent knots are $2/8=0.25$ apart in x.

Calculation	Result
First slope in x	$(0.380952381-0.285714286)/0.25=0.380952381$
Second slope in x	$(0.409090909-0.380952381)/0.25=0.112554113$
First ReLU weight W_0B	first slope: 0.380952381
Weight W_1B at knot 0.25	new minus previous slope: -0.268398268

For $0.25<x<0.5$, only the first two ReLUs have derivative 1, so the total slope is $0.380952381 - 0.268398268 = 0.112554113$. The new hinge is zero at its knot, preserving continuity.

Slopes computed with respect to τ must be multiplied by 8 to obtain weights for x: the same concentration change occurs over an x interval eight times shorter. Construct the A and C columns in the same way.

Hidden values and B concentration at τ=5 min

Subtract thresholds [0,0.25,0.50,0.75] from x=(5−2)/8=0.375.

j	$a_j=x-t_j$	h_j	Contribution $W_jB\cdot h_j$
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

Adding bias 0.285714286 gives B≈0.395021645 mol/L. Taking the midpoint of the 4-min and 6-min values also gives (0.380952381+0.409090909)/2=0.395021645.

The ReLU sum and segment interpolation express the same calculation. They differ from reference-process B=0.4 at 5 min, so this prediction falls slightly below the B≥0.40 specification.

Both directions of Big-M equivalence

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

A feasible formulation point with z = 0 has h = 0 and a ≤ 0; with z = 1 it has h = a ≥ 0. Either branch gives h = max(0, a).

Conversely, choose any point on the ReLU graph. Set z = 0 if a ≤ 0 and z = 1 if a ≥ 0; all four inequalities hold. Valid bounds L ≤ a ≤ U are needed in this direction too.

Substituting numbers into one ReLU formulation

For $a=2x-1$ and $x\in[0,1]$, $L=-1, U=1$. The inequalities are $h\geq 0$, $h\geq a$, $h\leq a+1-z$, and $h\leq z$.

Input and branch	Lower bound	Upper bound	Output
$x=0.2$, $a=-0.6$, $z=0$	$h\geq 0$	$h\leq 0$; $h\leq 0.4$	$h=0$
$x=0.8$, $a=0.6$, $z=1$	$h\geq 0.6$	$h\leq 0.6$; $h\leq 1$	$h=0.6$
Wrong $z=0$ at $x=0.8$	$h\geq 0.6$	$h\leq 0$	Impossible

```
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
```

The binary selects a branch compatible with a; it does not create an arbitrary output. At $a=0$, either z is possible but both give $h=0$.

Equivalence of multilayer ReLU formulations

Fix input x. The first affine layer uniquely determines a^1, and its ReLU constraints determine h^1 exactly.

Once h^1 is fixed, the next affine layer determines a^2. Repeating this argument through the final layer shows that the MILP output equals the forward output.

Conversely, insert all forward intermediate values and choose each z from its neuron's sign to obtain an MILP-feasible point. Bounds must be valid at every layer; nonunique z values at zero do not change the output.

Layerwise propagation of interval bounds

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

After ReLU, the interval is [max(0, L), max(0, U)]. Pass it to the next affine layer as its input bounds.

For example, if $h_1 = x$ and $h_2 = x$, then $h_1 - h_2$ is always zero. Using only the separate intervals [0, 1] gives [−1, 1]. This remains valid but loses their correlation.

Variable ordering in the objective and constraint matrix

The variable order is [x, h_0, h_1, h_2, h_3, y_A, y_B, y_C, z_1, z_2, z_3]. Each matrix row is stored as row_lower ≤ Av ≤ row_upper.

For V = 2 + 8x, the constant 2 does not change the optimizer, so the solver minimizes 8x. Add 2 back when reporting V and its dual bound.

An integer variable becomes binary only when its bounds are [0, 1]. If pre-activations are explicit variables, assign lower bounds that permit negative values.

Storing one constraint as a solver row

Use variable order $[x, h_0, h_1, h_2, h_3, y_A, y_B, y_C, z_1, z_2, z_3]$. Only three columns are nonzero for $h_1 - x + 0.25 z_1 \le 0$.

```
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
```

Substitution	Row dot product	Check
$x=0.5, h_1=0.25, z_1=1$	$-0.5+0.25+0.25=0$	satisfies upper bound 0
$x=0.5, h_1=0.30, z_1=1$	$-0.5+0.30+0.25=0.05$	violates upper bound

For output equality $y_B - \sum W_{jB} h_j = b_B$, set both row bounds to b_B. Mark the z_1 column as integer and also give it bounds $[0,1]$. A coefficient row alone does not make a variable binary.

SciPy LinearConstraint interprets each row as `lb <= A @ v <= ub`. Check variable order and coefficient signs against the same equation.

Compute the optimal residence time by hand

B stays below 0.40 throughout 2-4 min. It increases over 4-6 min, so compute the first point reaching 0.40.

$$\tau = 4 + 2\frac{.40 - .380952380952}{.409090909091 - .380952380952}$$
$$\tau \approx 5.353846\text{min}$$

The same interpolation fraction gives [A, B, C] = [0.584615, 0.400000, 0.215385] mol/L. This satisfies C ≤ 0.25 and nonnegativity, so it is the minimum feasible residence time.

Interpolation fraction at the product lower bound

Across 4–6 min, B increases from 0.380952381 to 0.409090909. Divide the required increase by the full segment increase.

Calculation	Value
Required B increase	$0.40-0.380952381=0.019047619$
Full segment increase	$0.409090909-0.380952381=0.028138528$
Segment fraction α	$0.019047619/0.028138528\approx0.676923077$
Residence time	$4+2\alpha\approx5.353846154$ min
C at the same α	$0.152380952+\alpha(0.245454545-0.152380952)\approx0.215384615$

B is below 0.40 on all earlier segments. At this first crossing, $C\leq0.25$ and nonnegativity also hold, giving the minimum volume. Comparing interpolated B with C at a different τ would not check specifications for the same design.

The continuous relaxation of a scalar ReLU

For L < 0 < U, relax z to [0, 1] and eliminate it to obtain a triangle.

$$h \geq 0, h \geq a$$
$$h \leq \frac{U}{U-L}(a - L)$$

The upper inequalities give h/U ≤ z ≤ (h − a + L)/L. Combining them, taking care with the negative sign of L, yields the last inequality.

This is the convex hull of the scalar (a, h) graph. When a is an affine function of several inputs, it need not be the strongest relaxation in the original input space.

MILP representation of affine equality projection

$$y = Py_{\text{raw}} + Ru + q$$

If the Week 4 projection has this affine form with fixed P, R, and q, append it directly as output equalities. No additional binaries are required.

If the constraint matrices depend on the decisions, inverses or products of variables may appear, so the same conclusion does not automatically apply.

When adding a separate conservation equality, check compatibility with existing outputs. A new equality imposed on an approximate predictor can shrink or empty its feasible set.

Exercise 1: signs and branches

For $a = 2x - 1$ and $x \in [0, 1]$, find L and U and write the Big-M inequalities. Calculate (a, h, z) at $x = 0.2$ and 0.8.

Check: $L = -1$, $U = 1$, with $h \geq 0$, $h \geq 2x - 1$, $h \leq 2x - z$, and $h \leq z$.

At $x = 0.2$ the tuple is $(-0.6, 0, 0)$; at $x = 0.8$ it is $(0.6, 0.6, 1)$. At $x = 0.5$, $h = 0$ and either binary value is possible.

Exercise 2: product specifications and feasibility

Does feasibility remain if $B \geq 0.41$ replaces $B \geq 0.40$ while $C \leq 0.25$ stays unchanged? Use the knot table and segmentwise interpolation.

Check: C is 0.245455 at 6 min and 0.328205 at 8 min. The C bound limits residence time to about 6.109859 min.

B there is about 0.409155 mol/L, and its values in earlier segments are smaller. The modified surrogate problem is therefore infeasible. This calculation helps distinguish an infeasible solver result from an implementation error.

Exercise 3: inputs for formulation verification

Random checks can miss errors at breakpoints or domain endpoints. Include $\tau = 2, 4, 6, 8, 10$ and points on both sides of interior knots.

At each fixed input, measure the maximum absolute difference in h and y. At $a = 0$, require output agreement rather than identical binary values.

Check: the supplied run had a maximum difference of about 1.25×10^{-8} across 41 inputs. Solver tolerances can leave small differences very close to a breakpoint. Distinguish mathematical equivalence of a valid formulation from finite-precision computation.

Lab files and sources

`week05_lab.py` includes coefficient construction, MILP matrix assembly, fixed-input checks, and independent segment calculations. Use `frozen_network.json` and `fit_knots.csv` to inspect the model; use `results.json` and `milp_matrix.csv` to inspect the solve.

Boyd & Vandenberghe, Convex Optimization, §4.1: variables, objectives, and feasible sets.

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

SciPy milp documentation: bounds, integrality, LinearConstraint, and return status.

Lecture · Lab bundle