# Homework 2

CS 521: Fall 2025

Due on Friday, October 3, 2025 11:59 PM Central

**Homework Policy:** You are allowed to collaborate with your classmates, but report this in your submission. Even if you work on problems together, each student must write up their solution individually in their own words in their submission. This CS 473 course page on academic integrity is a handy reference: `https://courses.engr.illinois.edu/cs473/sp2023/integrity.html`. We have ZERO tolerance for breaches of academic integrity, such as, but not restricted to, plagiarism, use of LLMs except for when stated explicitly, etc.
You can take upto 3 days extra in total across the semester without explanation. Once you exhaust the three-day late submission allowance, any further late homework submissions will result in a 25% penalty per each extra day. 5 minutes late in submitting the homework will be counted as 1 day late. If you have exceptional personal circumstances that will prevent you from submitting the assignment on time, write to the course staff as soon as possible. Unless explicitly stated, use of LLMs for solving homework or cheating from others is not allowed – we will detect it! There will be an oral **exam** if we detect violations, during which you will be asked to answer a randomly selected question from the homework. If you are unable to show sufficient knowledge, then you will get zero for the full homework.
Please format your submission as a PDF (file naming convention: *CS521_⟨netid⟩_hw2.pdf*) and upload it on Gradescope by the deadline. Try to start every problem on a new page, to accurately map pages to questions on Gradescope.
**Programming Problems:** Read these guidelines carefully! Describe your solution at a high-level (if applicable). Write only small snippets of code, if at all. DO NOT paste tall walls of code into your submission! Upload your solution files to GitHub publicly and provide a link to the relevant files as part of your solution.

**Points: Problem 1: 5 + 10 = 15 points, Problem 2: 10 + 15 = 25 points, Problem 3: 10 + 20 = 30 points, Problem 4: 30 points; Total: 100 points**

**Problem 1** (Operations on Complex Intervals)**.** A complex interval is defined as the set of all complex numbers whose real and imaginary parts lie within given intervals:

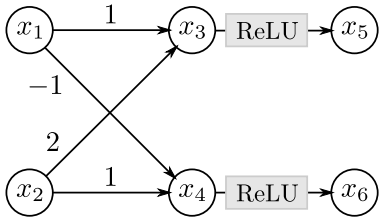
$$I = [a, b] + i[c, d] = \{\, x + iy \;\mid\; x \in [a, b],\; y \in [c, d]\}.$$

Consider two complex intervals: $I_1 = [a_1, b_1] + i[c_1, d_1], \quad I_2 = [a_2, b_2] + i[c_2, d_2]$.

(a) Compute the most precise complex interval soundly capturing the addition $I_1 + I_2$.

(b) Repeat the above for $I_1 \cdot I_2$.

**Problem 2** (Certification using Zonotopes)**.** Consider the following small neural network with two input neurons $x_1, x_2$ and two output neurons $x_5, x_6$. The network consists of an affine layer followed by a ReLU layer.




$$x_5 = \text{ReLU}(x_1 + 2x_2)$$
$$x_6 = \text{ReLU}(x_2 - x_1)$$

You are given the following zonotope $\phi$ over the input neurons:

$$\phi: \quad \begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \cdot \varepsilon_1 + \begin{pmatrix} 1 \\ 2 \end{pmatrix} \cdot \varepsilon_2 + \begin{pmatrix} 4 \\ 3 \end{pmatrix}$$

Your goal is to prove that $x_5 \geq x_6$ for all inputs $x_1, x_2$ in the zonotope $\phi$.

(a) Draw the 2D shape of $\phi$.

(b) Is it possible to design Zonotope transformers for the affine and ReLU layers to prove the property? If yes, then show them. If not, then why?

**Problem 3** (MILP encoding for Maxpool)**.** In the lecture, we learned the Mixed Integer Linear Programming (MILP) based encoding of the ReLU operation. In this exercise, your goal is to design an encoding of the Maxpool operation using MILP.

(a) Design a MILP encoding for the Maxpool operation $y := max(x_1, x_2)$ where the input bounds for $x_1$ and $x_2$ are $[a_1, b_1]$ and $[a_2, b_2]$ with $a_1, a_2, b_1, b_2 \in \mathbb{R}$.

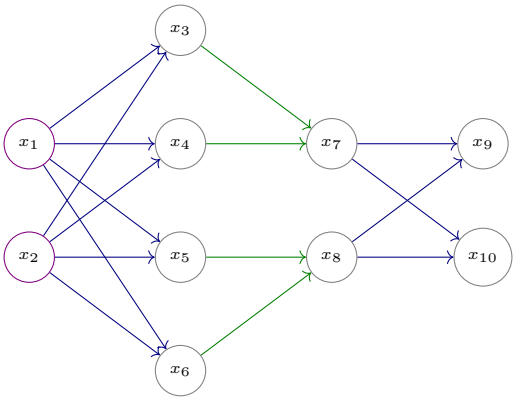



Figure 1: Fully-connected network with Affine and ReLU operations.

(b) Now, consider the neural network shown in Fig. 1. The neural network has two input $(x_1, x_2)$ and two output $(x_9, x_{10})$ neurons and consist of two layers with affine transformations (edges colored blue) and one layer with maxpool operation (edges colored green). The transformations in the network are given as:

$$
\begin{aligned}
x_3 &:= x_1 + x_2 \\
x_4 &:= x_1 - 2 \\
x_5 &:= x_1 - x_2 \\
x_6 &:= x_2 \\
x_7 &:= max(x_3, x_4) \\
x_8 &:= max(x_5, x_6) \\
x_9 &:= x_7 \\
x_{10} &:= -x_7 + x_8 - 0.5
\end{aligned}
$$

Use the MILP encoding for ReLU and Maxpool for verifying the property that for all values of $x_1, x_2 \in [0, 1]$, the output at $x_9 > x_{10}$. Can the MILP analysis prove this property? Show your work.

**Problem 4** (Programming)**.** In this problem, you will implement interval analysis for a simple neural network. The starter code of this problem is at `https://github.com/enyijiang/CS521FA25HW/blob/main/hw2/mnist_interval_analysis.ipynb`.

**Network Training.** We provide a fully connected neural network. You need to train your network on MNIST with 15 epochs and report the clean accuracy.

**Implement interval analysis** for your network. Use this to measure the robustness for the whole test set of your network for 10 $l_\infty$ neighborhoods, sized evenly between

0.01 and 0.1 (inclusive). Report the verified accuracy (the percentage of the images that are successfully verified) for those different values chosen. As usual, present your observations clearly along with experimental evidence (numbers, images, etc.) in your submission.

**Solution Requirements.** You should present your solution for this in the form of a Jupyter notebook. We recommend using Google Colab since we can interact with your solution easily, but you can also just upload the notebook to your GitHub repo.

**Hint.** You may refer to this repo `https://github.com/Zinoex/bound_propagation` for the implementation of interval analysis.

**Bonus Research Question** (25 points). In this problem, we ask you to compute bounds on the worst-case error function for a VAE model with two inputs $x_1$ and $x_2$, when the same noise $v$ with $||v||_\infty \leq \epsilon, \epsilon \in \mathbb{R}$ is added to them. The bound should hold with at least 95% probability. The worst-case error with respect to a given noise $v$ added to $x_i$ refers to the reconstruction loss (mean-squared error) between the original input $x_i + v$ to the encoder and the point on the corresponding output distribution at the decoder maximizing the error. You can consider a ReLU-based encoder and ReLU-based decoder, such as the one available at `https://github.com/uiuc-focal-lab/civet/tree/main`. You can also refer to the paper here [XBVS25] and [LXS+23], which is a real-world example of studying UAP on a VAE model. To adapt to the VAE model, you need to find the support set for the same noise added to those two inputs and bound the worst-case error of the decoder using relational verifications by designing a difference tracker.

## References


[LXS+23] Zikun Liu, Changming Xu, Emerson Sie, Gagandeep Singh, and Deepak Vasisht. Exploring practical vulnerabilities of machine learning-based wireless systems. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, pages 1801–1817, Boston, MA, April 2023. USENIX Association.

[XBVS25] Changming Xu, Debangshu Banerjee, Deepak Vasisht, and Gagandeep Singh. Support is all you need for certified VAE training. In *The Thirteenth International Conference on Learning Representations*, 2025.