CECS 551
Assignment 2
Total: 50 Points

General Instruction

- Submit uncompressed file(s) in the Dropbox folder via Canvas (Not email).
- Use `Python 3`, any other programming language is not acceptable.
- You can import modules in the following list (please check the full list Announcements - List of allowed libraries for the assignments.). If you want to use any other library, please consult with the instructor.

---

1. Implement a multi-layer neural network **without** using external deep learning libraries like PyTorch, Keras, Caffe, Theano, etc. Follow the steps below:
   (a) Find the `Assignment_2.ipynb` file, which trains a model to approximate the Exclusive OR (XOR) function using PyTorch.
   (b) Consider a neural network as shown in Figure 1:
      - The first layer has a width of 2, and the second layer has a width of 1.
      - The activation function for the layers is the sigmoid.
      - The loss function is binary cross-entropy.
   (c) (18 points) In the Jupyter notebook file, write the formulas for the partial derivatives $\frac{\partial L}{\partial \vec{W}^{(1)}}$, $\frac{\partial L}{\partial \vec{w}^{(2)}}$, $\frac{\partial L}{\partial \vec{b}^{(1)}}$, and $\frac{\partial L}{\partial b^{(2)}}$. Do not include the derivation steps; only the final formulas are needed. Use LaTeX equations in the notebook.
   (d) Implement the model without using any deep learning libraries.
   (e) (20 points) Optimize the parameters $\vec{W}^{(1)}, \vec{w}^{(2)}, \vec{b}^{(1)}$, and $b^{(2)}$ using the gradient descent method. For example, update $b^{(2)}$ as $b^{(2)} \leftarrow b^{(2)} - \eta \frac{\partial L}{\partial b^{(2)}}$, where η is a small positive number.
   (f) (6 points) Plot a graph showing the relationship between the epoch number and the loss.
   (g) (6 points) Predict $\hat{y}$ for a given input $\vec{x}$.
   (h) Submit your `ipynb` file.

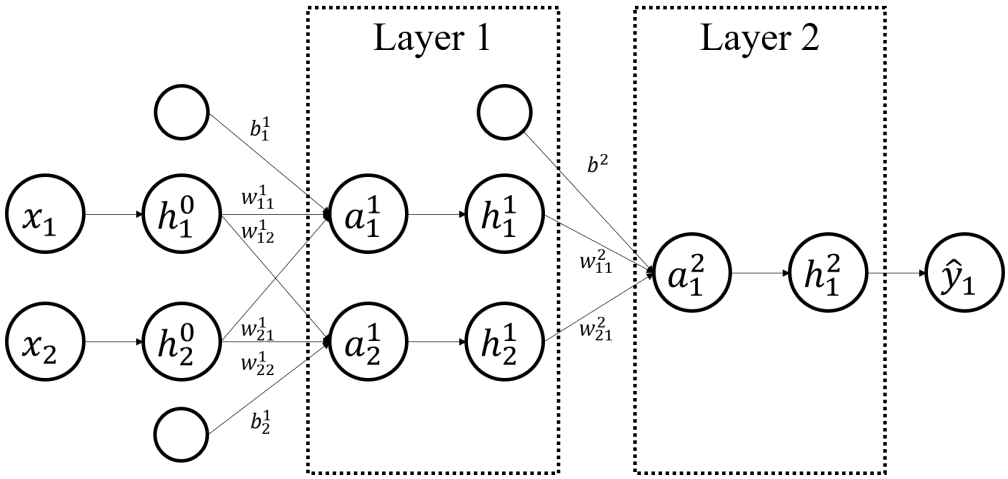

Figure 1: Network architecture