CECS 551
Assignment 3
Total: 70 Points

General Instruction

- Use `Python 3`, any other programming language is not acceptable.
- You can import modules in the Python Standard Library (please check the full list *here*). If you want to use any other library, please consult with the instructor.
- Submit uncompressed file(s) in the Dropbox folder via BeachBoard (Not email).

1. Implement multi-layer neural network **WITHOUT** using external deep learning libraries such as `Keras, Caffe, Theano, TensorFlow, PyTorch, ...`
   (a) Find `Assignment_3_scratch.ipynb`.
   (b) Consider a neural network as shown in Figure 1.
      - The width of the layer 1 is 2, and the width of the layer 2 is 1.
      - The activation functions of the layer 1 are the hyperbolic tangent.
      - The activation function of the layer 2 is the sigmoid.
      - The loss function is the mean squared error.

   (c) (26 points) Write formulas of $\frac{\partial L}{\partial \vec{W}^{(1)}}$, $\frac{\partial L}{\partial \vec{w}^{(2)}}$, $\frac{\partial L}{\partial \vec{b}^{(1)}}$, and $\frac{\partial L}{\partial b^{(2)}}$ in the Jupyter notebook file. **Please don't** write down steps of the derivation, final formulas would be sufficient. Please use LaTeX equations in the Jupyter notebook.
   (d) (44 points) Implement the model **without** using any deep learning libraries. You need to optimize the parameters $\vec{W}^{(1)}, \vec{w}^{(2)}, \vec{b}^{(1)},$ and $b^{(2)}$ using the gradient descent method. For example, $b^{(2)} \leftarrow b^{(2)} - \eta \frac{\partial L}{\partial b^{(2)}}$ where η is a small positive number. Your program should predict $\hat{y}$ for a given input $\vec{x}$ and report the training losses and the validation losses.
   (e) Submit your `ipynb` file.

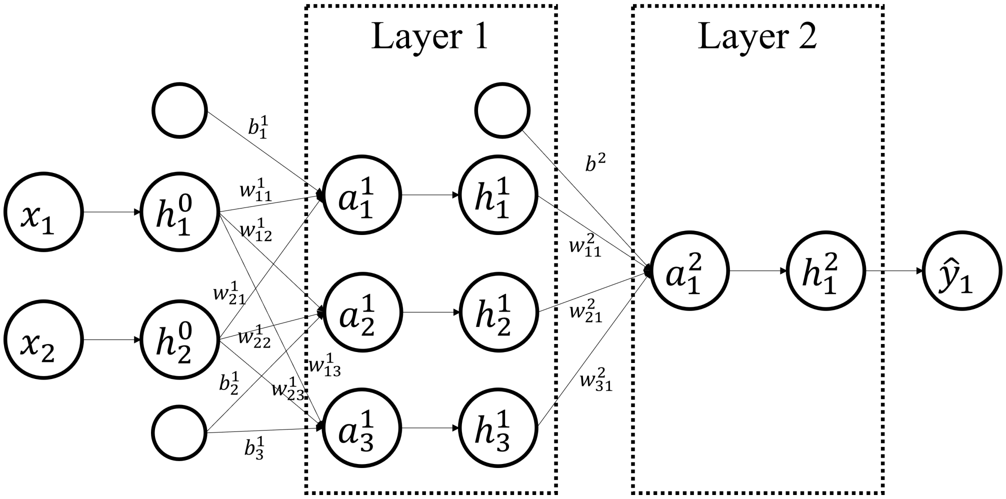

Figure 1: network design