CECS 451
Assignment 7
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 of libraries (please check the full list *here*). If you want to use any other library, please consult with the instructor.

1. Implement a multi-layer neural network from scratch. Please avoid using any deep learning libraries, such as PyTorch, Keras, etc.
   (a) Locate the file `Assignment_7_scratch.ipynb`. This file includes a cell to generate a toy dataset and divide it into three segments: training, validation, and testing.
   (b) Consider the neural network illustrated in Figure 1.
   - Layer 1 has a width of 3, and Layer 2 has a width of 1.
   - The activation functions used are sigmoid functions.
   - The loss function is the mean squared error, defined as $L(y_i, \hat{y}_i) = \frac{1}{2}(y_i - \hat{y}_i)^2$.
   - The program should predict $\hat{y}_i$ for a given input $\vec{x}_i$.

   (c) (10 points) Optimize the parameters w's and biases b's using the gradient descent method. For instance, update $b^{(2)}$ as follows: $b^{(2)} \leftarrow b^{(2)} - \eta \frac{\partial L}{\partial b^{(2)}}$, where η is the learning rate, a small positive number. Use small random numbers between $(-1, 1)$ for the initial parameters and zeros for the initial biases.
   (d) (10 points) Update the parameters for each sample in the training dataset. Completing one epoch involves updating the parameters for all samples.
   (e) (10 points) After every epoch, compute the training and validation costs using the following formula:
   $$J_D = \frac{1}{n} \sum_{i \in D} L(y_i, \hat{y}_i)^2$$
   where $D \in \{\text{train}, \text{valid}\}$. Plot the cost values as illustrated in Figure 2a.
   (f) (10 points) Adjust the learning rate η by monitoring the training and validation costs to determine the optimal learning rate. Once finalized, train your network with the combined training and validation dataset (`train_valid`) using this learning rate.
   (g) (10 points) Predict $\hat{y}_i$ from `x_test` and compare these predictions with `y_test`. Create a scatter plot to illustrate the relationship between the true values (y's) and the predicted values ($\hat{y}$'s), as shown in Figure 2b.
   (h) Submit your `ipynb` file.

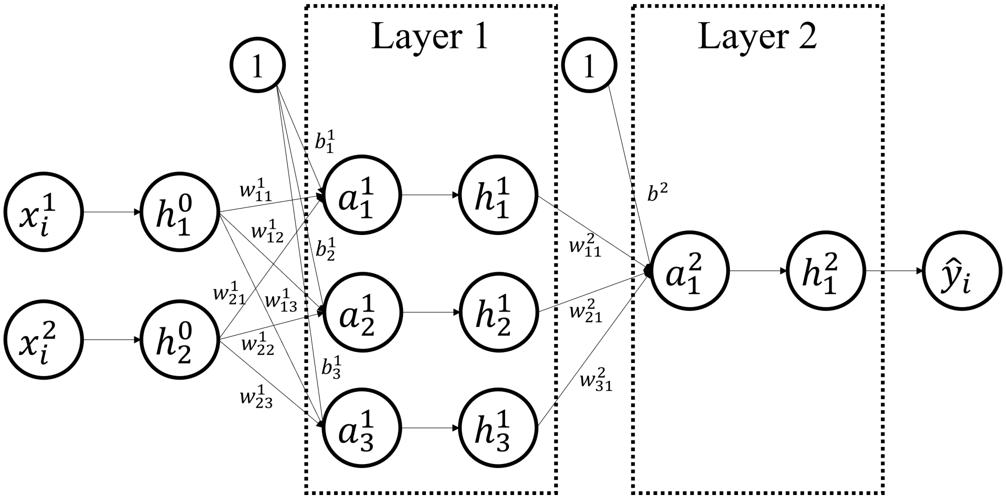
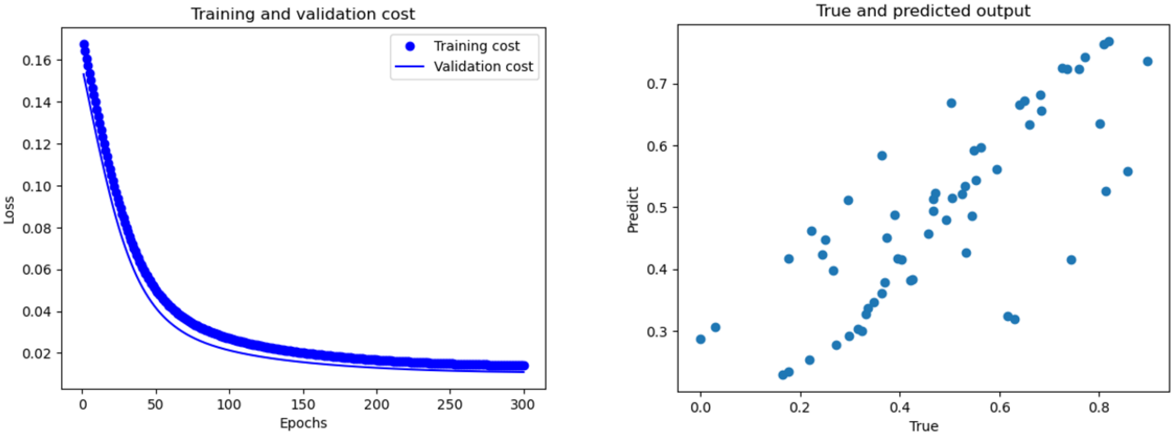


Figure 1: network design



(a) Training and validation cost

(b) True and predicted output

Figure 2: Output plots