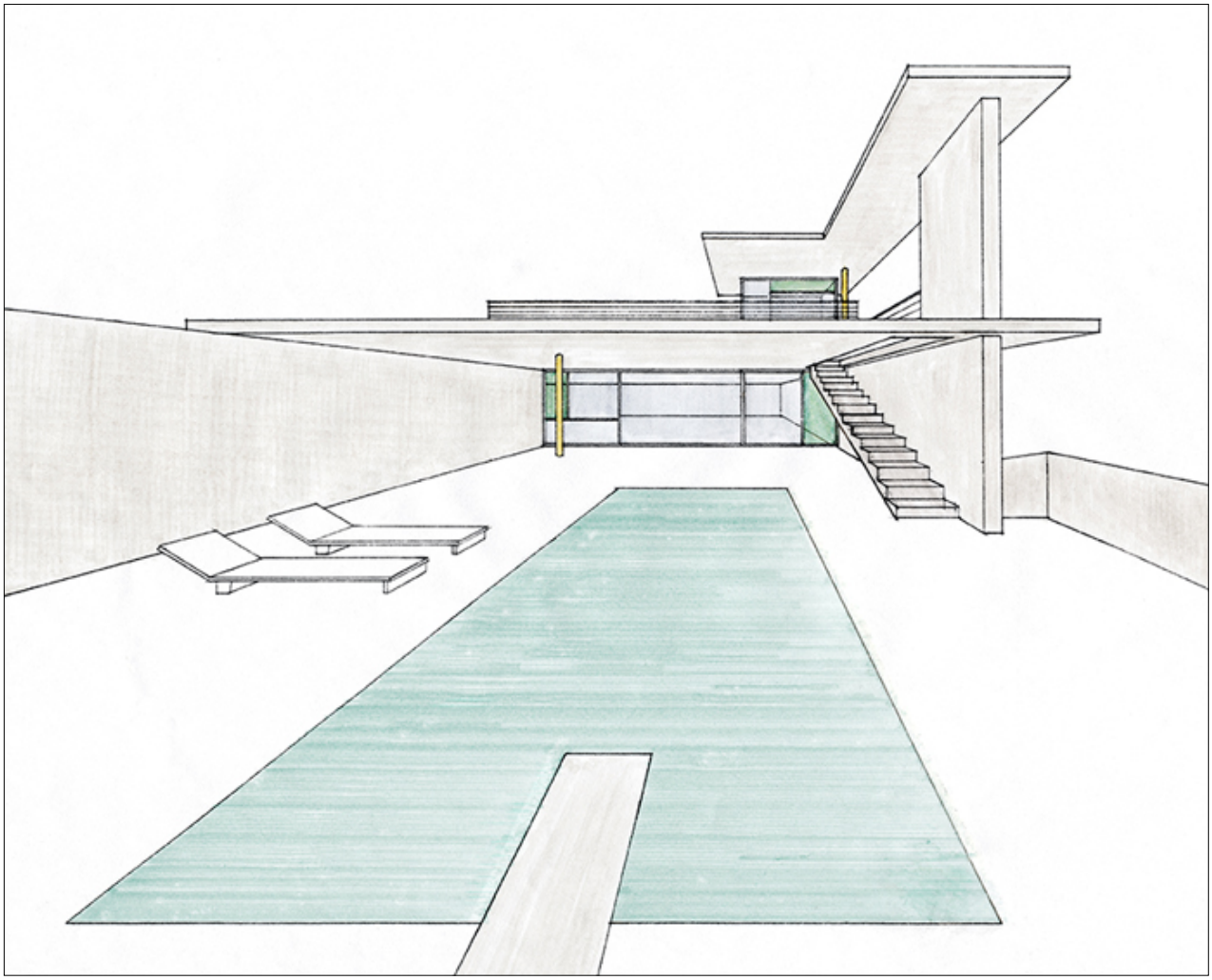
Multiple Layer Perceptron (MLP)



Assignment 5

Multiple Layer Perceptron (MLP)

Artificial Intelligence for Robotics
Dr. Min-Fan Ricky Lee

Multiple Layer Perceptron (MLP)

Submission

1. Matlab Code
2. Report (IEEE Paper Format)

1 MLP Step by Step from Scratch

Task 1: Modify Matlab program ***NNBPL_Complete_Cycle.mlx*** according to lecture example[1, pp. 37-69] as shown in Figure 1 by modifying weight updating algorithm (adding momentum)[1, pp. 80-83]

$$\Delta w^{(l)}(t+1) = -\eta \frac{\partial E_c(t)}{\partial w^{(l)}} + \gamma \Delta w^{(l)}(t) \tag{1}$$

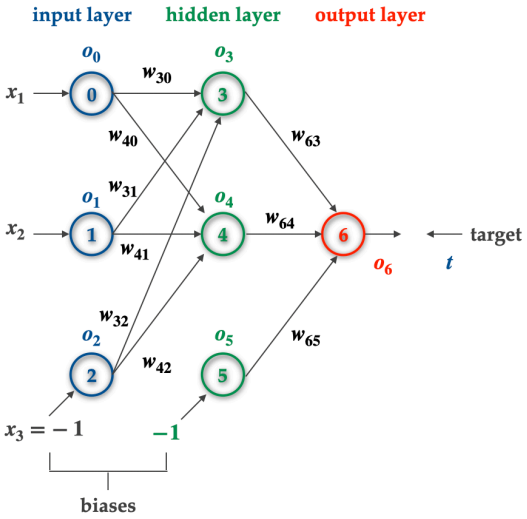


Figure 1. Lecture example: one hidden layer

Task 2: Modify the Matlab program ***NNBPL_Complete_Cycle.mlx*** according to the lecture example[1, pp. 37-69] as shown in Figure 1 by adding one more hidden layer with two nodes and one biases as shown in Figure 2.

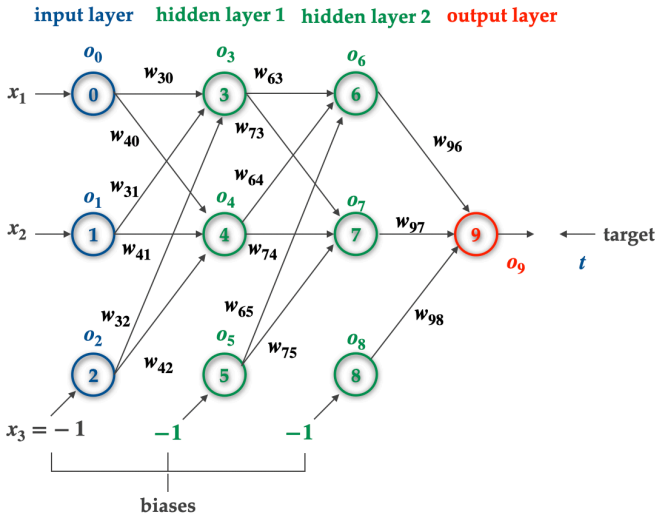


Figure 2. Two hidden layers

Multiple Layer Perceptron (MLP)

Take 3: Modify the Task 2 by modifying weight updating algorithm (adding momentum)[1, pp. 80-83] as in Equation (1).
Take 4: Compare the result among the configurations from the original, Task 1, Task 2 and Task 3.

2 Function Approximation

Modified ***Fcn_Approx_MLP.mlx*** to approximate ***f(x) = x·sin(x)*** from range [0:10] of variable ***x*** as illustrated in Lecture[1, pp. 70-83].

Task 1 : **Fixed Six** input/output **samples** were selected from range [0:10] of variable ***x*** by changing 3, 5 and 20 hidden nodes respectively as illustrated [1, pp. 71-73] in Lecture. Plot the actual vs. approximation and calculate Measure Root Mean Square Error (RMSE) as

$$\text{RMSE} = \sqrt{\sum_{i=1}^{N} \frac{(\hat{y}_i - y_i)^2}{N}} \tag{2}$$

$\hat{y}, y$ | approximated and actual value respectively

N=200 | sample size

Task 2: **Fixed** number of **neuron** (**Five**) by changing 3, 10 and 20 samples respectively randomly selected as illustrated in Lecture [1, pp. 74-76]. Make the plot and calculate RMSE as in Task 1.

3 Environment/Situation Classifier

Task 1: Modify either above program to implement the MLP [1, pp. 117-127], [2] as shown in Figure 3 to classify the environment as shown in Figure 4. Network has one input layer (18 neurons), three hidden layers (10 neurons, 5 neurons and 12 neurons) one output layer (7 neuron). This network is partially connected between the input layer and the second hidden layer and its remainder is fully connected.

Example Input-Out Pattern: Input layer (18 neurons) [2 2 2 1 1 1 1 2 2 2 2 0 0 0 0 2 2 2] as shown in Figure 5(b) corresponding to output layer(7 neuron) [0 1 0 0 0 0 0] as the Class 2 as shown in Figure 4.

Task 2: Connect to V-rep (simulation) or actual robot (Turtle bot 2) with sonar or laser range finder to test it.

Multiple Layer Perceptron (MLP)

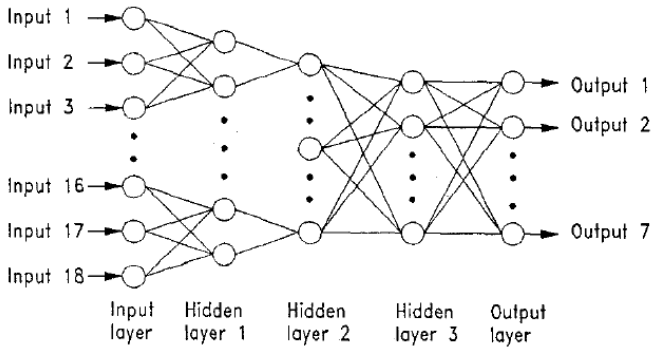


Figure 3. The structure of the neural network used as situation classifier.

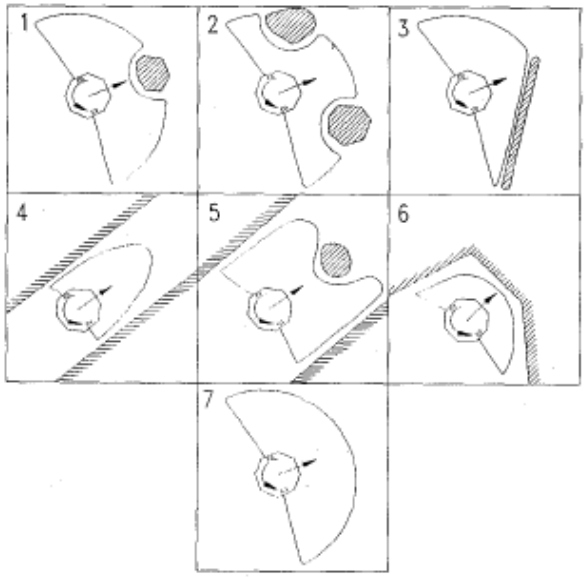


Figure 4. The typical classes of environmental situations.

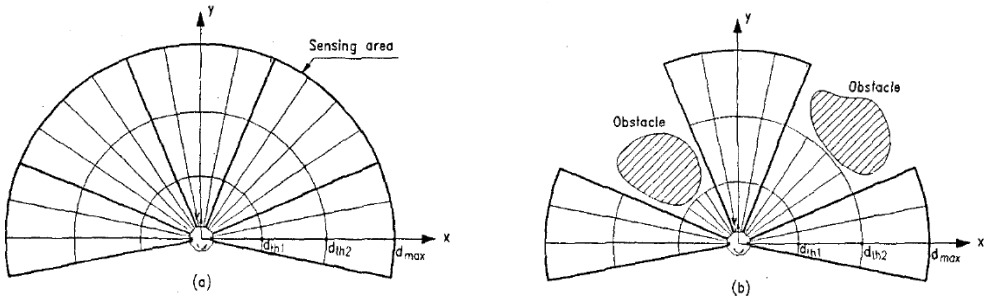


Figure 5. An example of training sample pattern. (a) The divided sensing area. (b) An example of class 2.

References

1. Min-Fan Ricky Lee,"*Artificial Intelligence*, " Lecture 2 Slide, 2020
2. H. R. Beom and H. S. Cho, "A Sensor-based Obstacle Avoidance Controller For A Mobile Robot Using Fuzzy Logic And Neural Network," Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Raleigh, NC, USA, 1992, pp. 1470-1475.