Data Structure Homework #1
XOR Operator Training Program
Using Neural Network

Ping-Chen,Chung(鍾秉辰)
Department of Communication Engineering
National Central University

September 26, 2022

Contents

1 Theoretical Background of Neural Network **2**
1.1 Neurons 2
1.2 Network of Neurons 3

2 Methods and Resources **3**
2.1 Dynamic Memory Allocation in C 3
2.1.1 The Difference of Static and Dynamic Memory Allocation 3
2.1.2 Usable Functions to Dynamically Allocate Memory . . 4
2.2 Function Libraries Used in This Program 4

3 Compile Results **5**
3.1 Compile Parameters 5

4 Execution Results **5**

5 Analysis **6**
5.1 Loss Function 6
5.2 Graph Display 6
5.3 Further Improvement 6

Abstract

This report introduces the basic structure of neural network and function in C language, used dynamic memory allocation to complete the neural network training system of XOR operator, and implemented it into the binary checksum program of a user-input string. The author then analyzed the graph of loss function(MSE loss) and proposed the further improvement of this program.

1 Theoretical Background of Neural Network

In this section, we're going to introduce the definition of neural networks, and its implementations in C as a library.

1.1 Neurons

Neurons are the fundamental part of the neural network, which simulates the real neuron from organisms. To turn it into math model, Scientists has given the specific definition and functions of neurons.

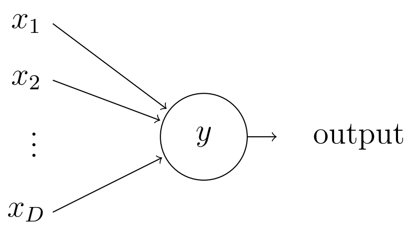

Figure 1: Model of Neuron [1]

Since the actual neuron in real live sends eletric signals only if the inputs has reached a specific conditions, we modeled it as the *threshold* of the neuron to activate and send signal to neurons in the next layer. *weight* is then applied to every inputs to adjust the importance of each input, and the neuron only activates if the total sum of every input multiplied by its weight is greater than the threshold.The mathematical function of a neuron is given below:

$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{ threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{ threshold} \end{cases} \tag{1}$$

Equation 1 shows one kind of the neurons: the perceptrons. However, in the modern model of neural network, we use Sigmoid functions or Softmax functions with other approximating techniques to evaluate neuron outputs in the aim of improving the accuracy of the trained model.

1.2 Network of Neurons

As the figure 2 shows, each neuron of the same layer is connected to every node of the next layer, therefore the output of this model can be transfomed into matrix form:

Matrixlization of Neuron Networks [2]
Let x inputs be numbered $i_1, i_2, \cdots i_x$, then matrix $\mathbf{I}$ is the matrix storing $1 \times x$ elements of inputs in the same layer. Each input neuron has its y weights respectively points towards y neurons of the next layer, then the total of input weight is represented by an $x \times y$ matrix $\mathbf{W}$. Multiply $\mathbf{I}$ and $\mathbf{W}$ gets the inputs of the y neuron in the next layer, stored in $y \times 1$ matrix $\mathbf{H}$.
The same method can be done repetitively until the output layer is reached.

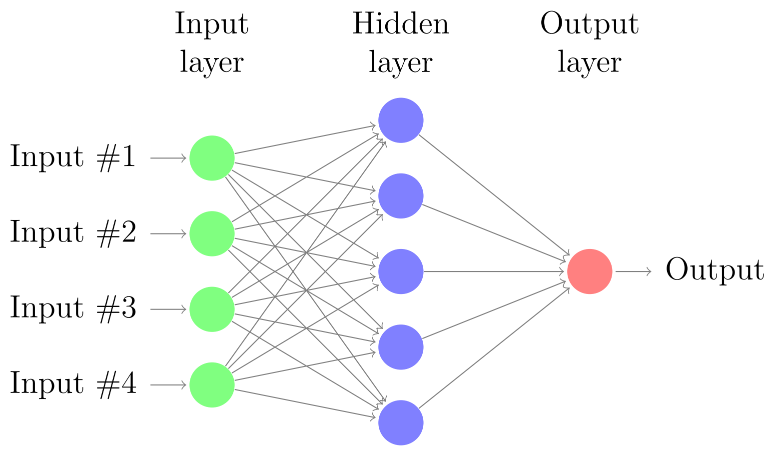

Figure 2: An Example of Neural Network

2 Methods and Resources

2.1 Dynamic Memory Allocation in C

2.1.1 The Difference of Static and Dynamic Memory Allocation

According to The reference [3], the author has concluded the advantages and disadvantages of static memory allocation and dynamic memory allocation, shown below:

Static Memory Allocation

- Allocation of memory is done automatically by the compiler. This will reduce the execution time of the program.
- Uses stack data structures.
- Allocated memory (including variables and arrays) cannot be freed until the program terminates.
- Memory is allocated before the script executes.
- Simple usage and easy for new learners to use.

Dynamic Memory Allocation

- Memory is allocated in runtime. Hence, the execution time will be increased.
- Uses heap data structures.
- Memory can be resized / freed by scripts, therefore lessen wasted memory in program.
- Memory should be freed manually after use, or it could cause unknown problems to debug.
- Needs better knowing of memory allocation to prevent unknown errors.

2.1.2 Usable Functions to Dynamically Allocate Memory

- `malloc()` allocates the dynamic memory without the initialization.
- `calloc()` allocates the dynamic memory with initialization to prevent accidentally access to corrupted memory.
- `realloc()` reallocates the existing memory to expand / shrink its memory size.
- `free()` frees the memory let it be recycled by operating system.

2.2 Function Libraries Used in This Program

In this program, we used an open-sourced neural network library Genann [4], with a small number of library file to include and clear documentation for users to embed into their codes.

3 Compile Results

3.1 Compile Parameters

Please open the project directory in command line and execute the following command:

```
$ make main
$ ./main
```

Listing 1: Compile Parameters

```
cc -Wall -Wshadow -O3 -g -march=native   -c -o main.o main.c
cc   main.o genann.o commonFunctions.o  -lm -o main
```

Listing 2: Compile Results

Note that the warnings from the compiler has be omitted for concise.

4 Execution Results

Below is one sample I/O of the program. Note that the input with >> is user input.

```
Welcome to this program. Please choose the mode you want to
   execute this program.
Enter 1 to show quadratic loss of the training process.
Enter 2 to enter custom string and show the XOR checksum
   based on NN learning output.
>>2
Please enter the training iteration. (integer)
The training result would be better if trained for 5000 times
     for 1 layer with 2 nodes per layer.
>>5000
Training Result: 0 1 1 0
If the result is not "0 1 1 0 ", please consider increase the
     iteration count or try rerun this program.
Please enter a string. This Program will output the XOR
   checksum of this string.
>>test_string

Checksum by XOR operator: 0
Checksum generated by neural network: 0
```

Listing 3: Main Program Execution Result

5 Analysis

5.1 Loss Function

From the reference [5], we can know that there exists various ways to evaluate the accuracy of a neural network. For simplicity, this program uses MSE (Mean-Square-Error) to be our loss function to evaluate the accuracy of the neural network. The equation is shown below:

$$\text{MSE} = \left(\frac{1}{n}\right)\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2}$$

where y_i is the actual training output and $\hat{y}_i$ is the desired output.

5.2 Graph Display

The graph below shows the loss function with sample training iterations (30000 times).

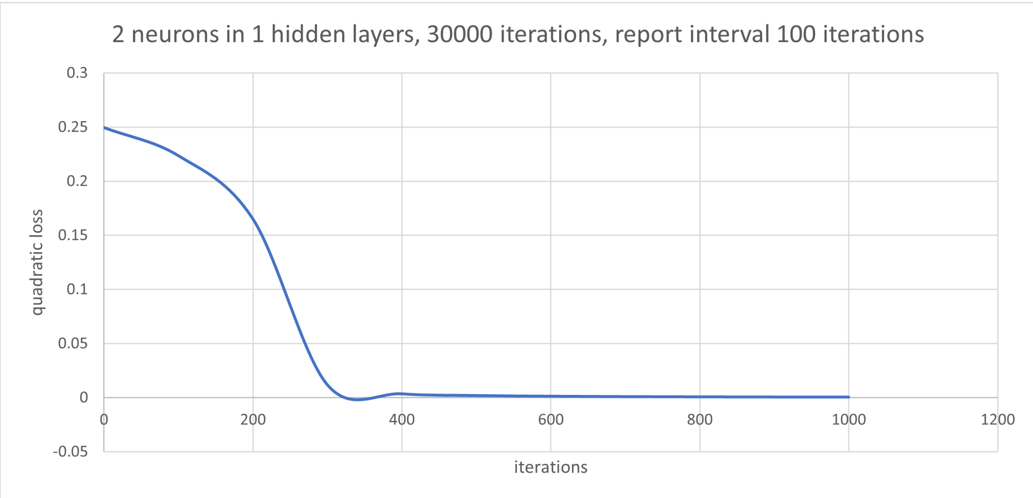

Figure 3: Graphs of Loss Function: MSE v.s. iterations

Note that this graph only shows up to 1000 iterations since the value after 1000 iterations approaches zero and is trivial.

5.3 Further Improvement

Although the program has successfully executed, there is still some issues to be fixed, listed below:

- an CSV output of loss function could be added when training completes.
- Data storage of bits can be reduced to `bool` array instead of `char` array to reduce memory.

References

[1] http://neuralnetworksanddeeplearning.com/chap1.html

[2] https://ml-cheatsheet.readthedocs.io/en/latest/forwardpropagation.html

[3] https://www.geeksforgeeks.org/static-and-dynamic-memory-allocation-in-c/

[4] https://github.com/codeplea/genann

[5] https://www.statlect.com/glossary/loss-function/