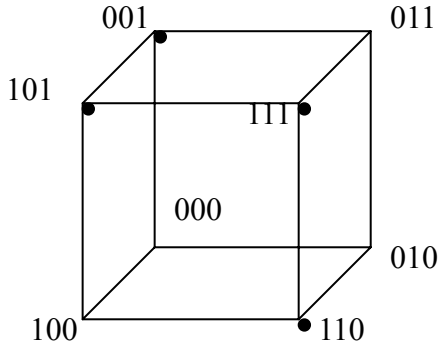
Chapter 1

Reification of Boolean Logic

Exercises

1.1 (a) Design a feedforward network to divide the black dots from other corners with fewest neurons and layers. Please specify the values of weights and thresholds.



(b) Is it possible to do (a) with a single neuron? Why?

1.2 Consider the neural network in Fig. P1. The value of each neuron can be 0 or 1 and the activation function used is f (net) = {1, net>0; 0, net<0}. Each neuron decides its own value according to the values of neighbors. We can adjust the weights of neurons so that every neuron will have distinct set of action rules (e.g. {000|0, 001|1, …, 111|0} is a set of action rules). If two neurons that have different sets of action rules are considered to be different kinds, then, how many kinds of neurons can we have in the network?

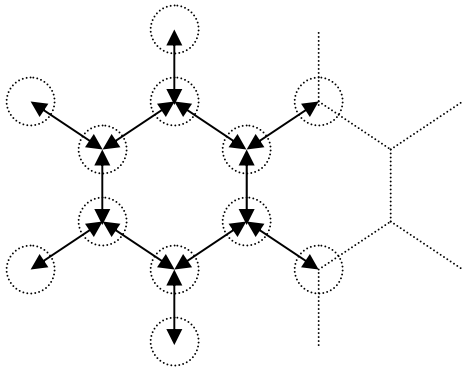


Fig. P1.

1.3 Design a feedforward network which provides the classification of the following pattern vectors:

Class 1:

$X = [0\ 0\ 0\ 0]^t, [0\ 0\ 1\ 0]^t, [0\ 0\ 1\ 1]^t, [0\ 1\ 0\ 0]^t, [0\ 1\ 0\ 1]^t, [0\ 1\ 1\ 0]^t, [0\ 1\ 1\ 1]^t, [1\ 0\ 1\ 0]^t, [1\ 1\ 0\ 0]^t, [1\ 1\ 1\ 0]^t, [1\ 1\ 1\ 1]^t$

Class 2:

$X = [0\ 0\ 0\ 1]^t, [1\ 0\ 0\ 0]^t, [1\ 0\ 0\ 1]^t, [1\ 0\ 1\ 1]^t, [1\ 1\ 0\ 1]^t$

Please specify the value of weights and thresholds and use as few neurons and layers as possible.

1.4 Please refer to p.217 in "Introduction to Artificial Neural Systems" by Jacek M. Zurada.

$$M(J, n) = 2^J,\ n \geqq J.$$

Is it enough to use J neurons as hidden layer to represent all 2^{2^n} Boolean functions when n = J?

1.5 To compare neural model equation $y = \sigma(\Sigma\ w_i x_i + w_0)$ with Boolean equations which use only operators $\cap$, $\cup$ and ~.

(1) List all different Boolean equations in the forms as follows.

(a) $Y = a_1X_1 \cup a_2X_2 \cup a_3X_3$.

(b) $Y = a_1X_1 \cap a_2X_2 \cap a_3X_3$.

(c) $Y = a_1X_1 \cup a_2X_2 \cap a_3X_3$.

(d) $Y = a_1X_1 \cap a_2X_2 \cup a_3X_3$.

(a_1, a_2, a_3 can be ~ or empty string.)

And specify which equations appear in neural model's 104 functions, which didn't. (Hint: You can use the 3-cube to represent Boolean equations)

(2) We have five methods to cut the cube to get the 104 functions. For each cutting method, write a corresponding Boolean equation in the forms given above.

1.6 We have these training patterns:

0: (0, 0, 0), (0, 1, 1), (1, 1, 0)

1: (0, 1, 0), (1, 0, 0), (1, 1, 1).

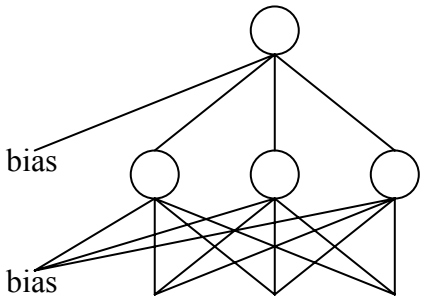
The training result:

$$W_1 = \begin{bmatrix} -0.9 & 1.5 & -1.0 \\ 2.5 & -3.7 & 0.37 \\ 2.5 & -2.0 & 4.0 \end{bmatrix}$$

$$W_2 = [5\quad 6\quad 6]$$

$$Bias_1 = [-0.95\quad -2.0\quad -3.1]$$

$$Bias_2 = -3$$



(1) Write the Boolean function of training patterns. (Make simplification)

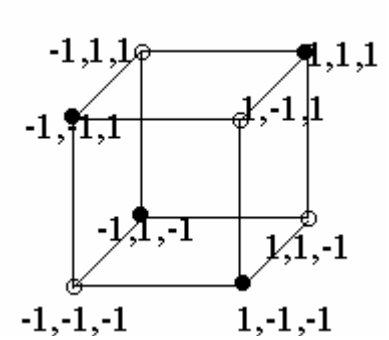
(2) Write the nested Boolean function of the network architecture for each

neuron. Notice that the Boolean function you wrote must be in the forms given in problem 1.

(3) Prove that function (1) and (2) are equivalent.

(4) (0, 0, 1) and (1, 0, 1) didn't appear in training patterns. So, we can get four different Boolean functions. Please choose the simplest Boolean function and compare it with the output of the above neural network of these two patterns.

1.7 Training patterns:

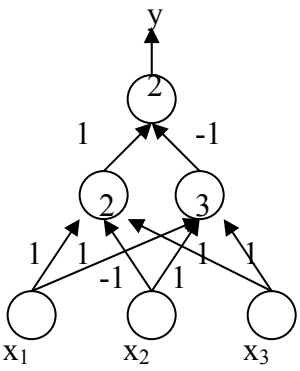


(1) Write the Boolean function. (You must make simplification)

(2) Design a 3-2-1 neural network to classify these patterns. If it cannot classify these patterns correctly, construct your own neural network to classify these patterns.

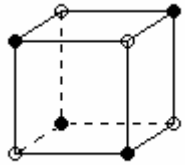
1.8 Boolean functions and neural network have a one to one map in architecture. Please find the map and make a description of it.

Hint: $y = (x_1 \wedge \neg x_2 \wedge x_3) \vee \neg(x_1 \wedge x_2 \wedge x_3)$ can map to this neural network



1.9 Find the coloring graph solution for $x_1 \oplus x_2 \oplus x_3 = Y$.

Remember $x_1 \oplus x_2 \oplus x_3 = Y$ is,



coloring graph is similar to that in page 79 "Turing Machines" by J. E. Hopcroft, 70-80.

1.10 Proposition: The Boolean function is defined as: there are six Boolean variables as input (x1, x2, …, x6) of this function, the output of this function is 1 only when any two variables are 1, (more than two variables or less than two variables are 1, the output of this function is 0.)

(a) Use Boolean algebra or Boolean function to express this proposition.

(b) Write a program (Turing machine, Lisp, C, or other programs) to simulate this expression, the input of the program is these six Boolean variables, the output of the program is according to the proposition.

(c) Discuss the trained weights of the multilayer network in homework for the above proposition. Can you figure out the meaning of those weights?

(d) Construct a multilayer network for this proposition. Use six neurons in input layer and one neuron in output layer.

1.11 In class we discussed a neuron with two Boolean variables as input. This neuron can simulate 14 Boolean functions (14 Boolean states) excluding the XOR. Assume the neuron is in state Si, which is one of the 14 Boolean functions. When we slightly tune the weights w1 w2 w3 of this neuron the current state Si will change to Sj first. Discuss all possible such first Sj when this neuron is in Si for all 14 states, i = 1~14.

1.12 Write an algorithm to train the neural network in problem 2 by its training

patterns. Notice that the algorithm must be constructed by the following subroutine:

train_one_neuron(x, d, w, y, Δw)

input: x, d, w. output: y, Δw.

x: an array of the input value.

d: a desire output.

w: the weights.

y: the output.

Δw: an array of the value of the weights should be added.

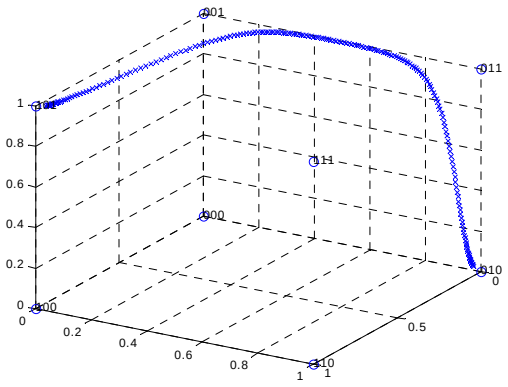
1.13 In appendix A we showed a modular design method for training ANN on data flow machine. Each neuron's desire response

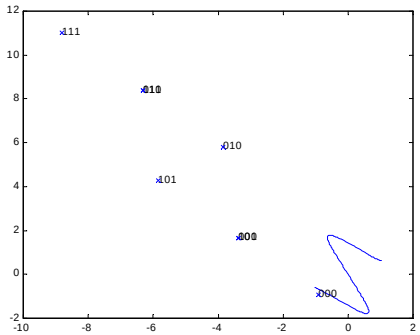
$$d_j^l = y_j^l + \sum_{i=1}^{m_{l+1}} \frac{\Delta w_{ij}^{l+1}}{\eta y_i^l} w_{ij}^{l+1}, \qquad \{l+1 = 2....L\}.$$

is inducted from choosing bipolar sigmoid function $\sigma(u) = \frac{2}{1+e^{-u}} - 1$. Please try to formalize the desire response when we choose unipolar sigmoid function

$$\sigma(u) = \frac{1}{1+e^{-u}}$$

1.14 Discuss and analyze the results obtained in training a 1-3-2 ANN. You can use the following results or your own results. The input is a continues number between 0 to 1 and output is a 'S' curve in the 2D plane.





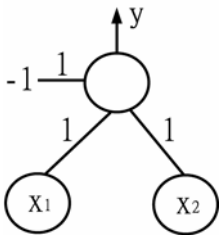
1.15 (a) A dataflow information processing architecture is a MIMD architecture without global or shared memory in which each processing element only operates when all of the necessary information that it needs to function has arrive. Show that neural networks are dataflow architectures.

(b) Invent a concrete, detailed example of a dataflow architecture that is not a neural network.

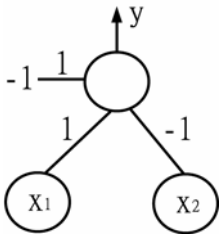
1.16 We have learned basic type of neural network:

$$y = f(\sum_{i=1}^{n} w_i x_i + \theta)$$

We can directly use it to simulate Boolean logic. For example: OR Boolean function: $y=x_1$ OR x_2, where x_1, x_2, y are Boolean variables having values {0,1}.



When above network use 'hardlim' active function, its output $y=x_1$ OR x_2.
Another example is AND Boolean function: $y=x_1$ AND $\neg x_2$



Please check the truth table for it by yourself and answer questions below:
(a) draw the neural network that perform:

$$y = \left(x_1 \vee \neg x_2 \vee x_3 \vee \neg x_4\right)$$

Note: the weights and bias should be {-1,1} for simplicity.

(b) draw the neural network that peform:

$$y = \left(x_1 \wedge \neg x_2 \wedge x_3 \wedge x_4\right)$$

Note: the weights and bias should be {-1,1} for simplicity.

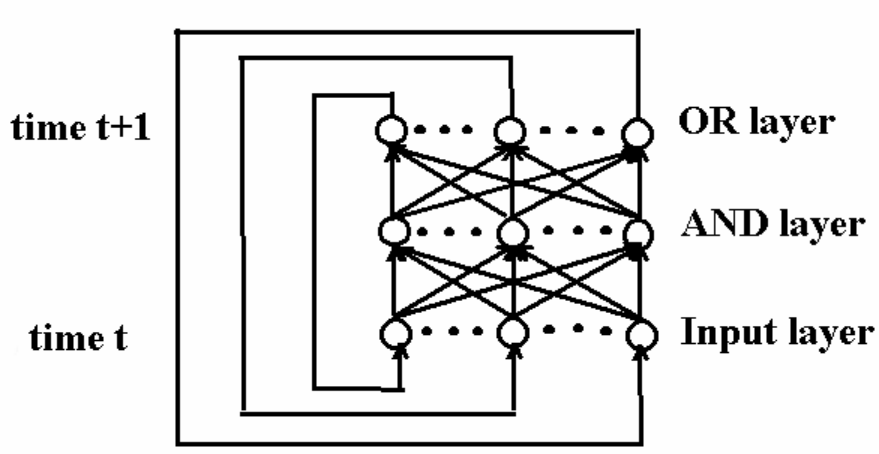
(a) Can you formalize how to set up the weights and biases for a given OR expression?

(b) Can you formalize how to set up the weights and biases for a given AND expression?

(c) Try to use a 3-layers neural network for simulating the DNF equation:

$$y = \left(x_1 \wedge \neg x_2 \wedge x_3\right) \vee \neg\left(x_3 \wedge x_4 \wedge x_5\right)$$

(f) The neural network's outputs can be feedback as inputs like:



It can simulate many kinds of dynamic process like gene regulation and etc. If n Boolean variables $x_1, x_2, \ldots, x_n$ are changed by time and its value are known for a period of time. We can build ANN model like above for these n-variables. Please try to build an ANN model for the given 4-variables.

Time	x_1	x_2	x_3	x_4
1	1	0	1	1
2	0	1	0	0
3	0	0	0	1
4	1	1	1	1
5	0	1	1	0
6	1	1	0	0
7	1	0	0	0
8	0	1	0	1

(Hint: first rewrite $x_1 = F(x_1, x_2, x_3, x_4)$ and get its DNF)