Artificial Intelligence and Machine Learning

Midterm Report

Yang, Hao Zhong, Wu, Po Hsun

Contents

Chapter 1

Electronic Components

1.1 Resistor

Under normal circumstances, in order to prevent the LED from being damaged due to an excessively high voltage applied to the LED, a control resistor is used to limit the flow of current to make the circuit operate smoothly.

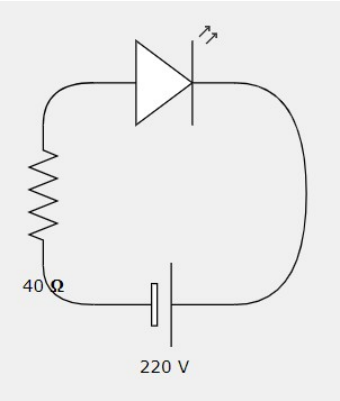


Figure 1.1

1.1.1 Variable resistor(Potentiometer)

Different from general resistors, it has three terminals. When in use, by changing the resistance value between the sliding end and the two fixed ends, different voltage division ratios are formed, so as to change the potential of the sliding point.

1.2 Operational Amplifier

Basically, an operational amplifier has two inputs (inverting and non-inverting) and one output.And the operational amplifier will generate two different outputs according to whether the voltage passed in by Vp is greater than Vn like (2.7).

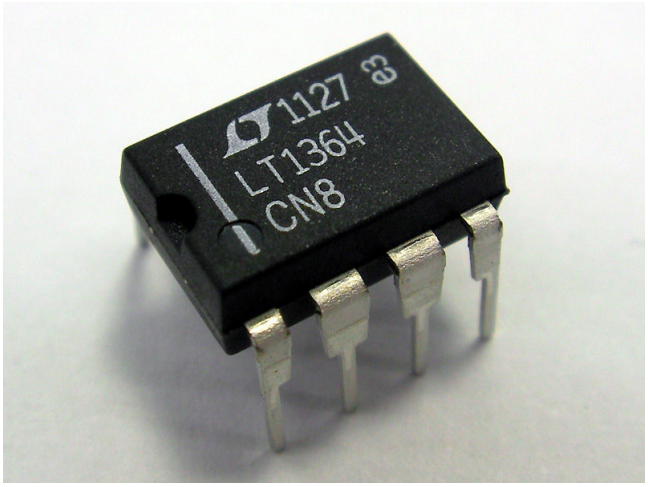


Figure 1.2

1.3 Power supply

In order to supply the electronic power, we at least use two 9V batteries in this project because we need both positive (+) and negative (-) voltages like Fig.1.3.

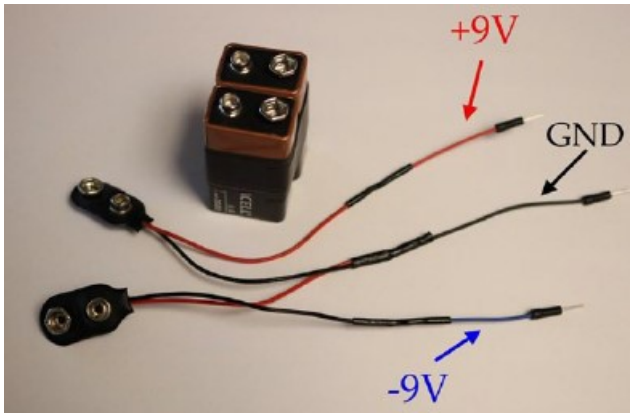


Figure 1.3

Need to pay attention when implementing

- how the battery clips are wired
- Ensuring the supply voltages are stable even under varying load conditions, you can use a +5V regulator and a -5V regulator.

1.3.1 Voltage regulator

The purpose of a voltage regulator is to provide a stable voltage that does not change under different load conditions, and when wiring up these regulators, remember their ”pin out.” which the function of each pin is different, as shown in Fig.1.4.

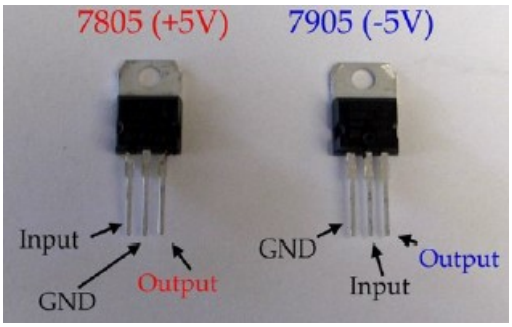


Figure 1.4

1.4 LED

LEDs are often used to identify whether current is looping in a circuit, with shorter legs of cathode and longer legs of anode. In circuit diagrams, LEDs are sometimes represented as diodes.

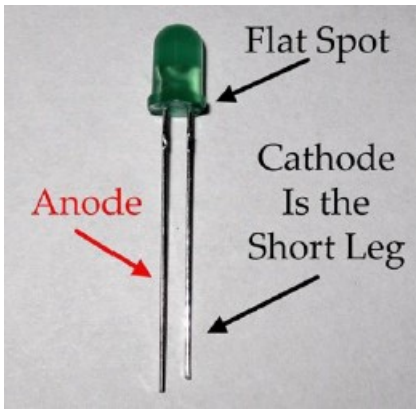


Figure 1.5

1.5 SPDT switch

In order to have two dissimilar outputs to the network, a SPDT could switch two different paths of circuits by using a switch which as shown Fig.1.6, and can be done manually or included through the electromagnetic coil.

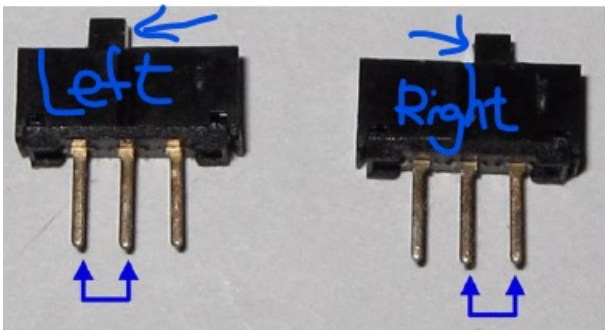


Figure 1.6

Chapter 2

Basic Circuit

2.1 Voltages Summing

The Fig.2.1 is the summation circuit. By KCL(Kirchhoff's Current Law), the sum of current entering the output node must always be 0.

$$I_1 + I_2 = 0 \tag{2.1}$$

We can use Ohm's law to replace $I_1,\ I_2$, then become

$$\frac{V_{\text{out}} - V_1}{R_1} + \frac{V_{\text{out}} - V_2}{R_2} = 0 \tag{2.2}$$

So we can get the voltage of output node V_{out} is

$$V_{\text{out}} = \frac{R_2 V_1 + R_1 V_2}{R_1 + R_2} \tag{2.3}$$

For the case in the textbook, the resistor R_1 is equal to the resistor R_2. So, we can get the output voltage was the average of the input voltages.

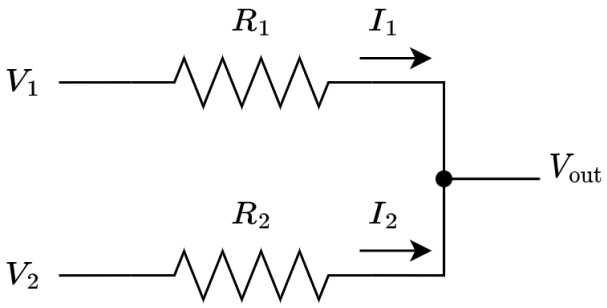


Figure 2.1: Summation circuit

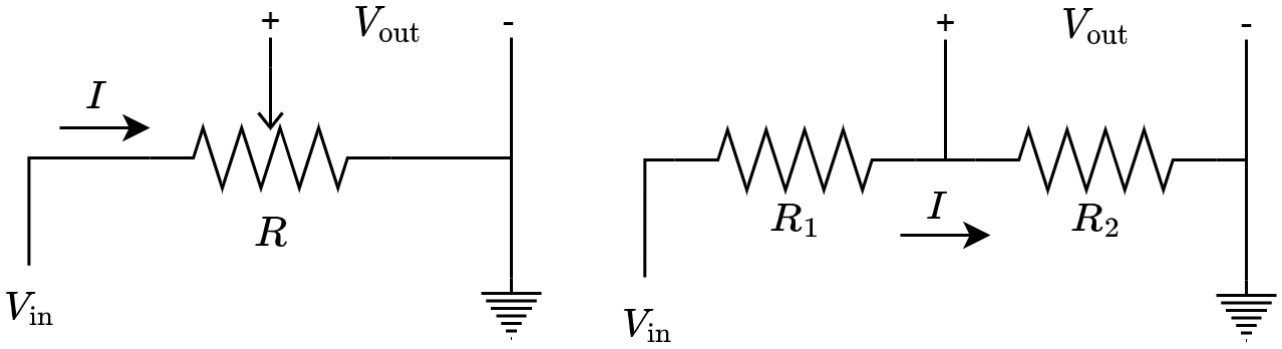
2.2 Voltage Dividing

The Fig.2.2 is the circuit of dividing voltage by Potentiometer. Because the R_1 and R_2 are in the same loop. The current passed through both resistors was equal. From the Ohm's law and resistor series rule, we can get the current I.

$$I = \frac{V_{\text{in}}}{R} = \frac{V_{\text{in}}}{R_1 + R_2} \tag{2.4}$$

Then we can get the output voltage V_{out}.

$$V_{\text{out}} = IR_2 = \frac{R_2}{R_1 + R_2} V_{\text{in}} \tag{2.5}$$



(a) Potentiometer (b) Equivalent

Figure 2.2: Divide circuit

2.3 Voltage Comparing

Fig.2.3 is the open loop operational amplifier(Op-Amp). And the relation between output voltage and input voltages is

$$V_{\text{out}} = A_{\text{OL}} \left(V_{\text{p}} - V_{\text{n}} \right) \tag{2.6}$$

the A_{OL} is the open loop gain of op-amp. In general, the gain value is huge but the power supply can't provide that high voltage. We can rewrite the (2.6) become

$$V_{\text{out}} = \begin{cases} +V, & \text{if } V_{\text{p}} > V_{\text{n}} \\ -V, & \text{if } V_{\text{p}} < V_{\text{n}} \\ 0, & \text{if } V_{\text{p}} = V_{\text{n}} \end{cases} \tag{2.7}$$

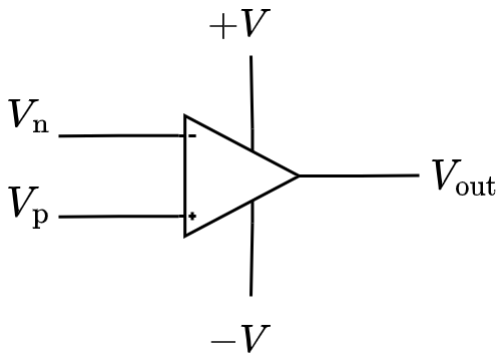


Figure 2.3: Operational amplifier

Chapter 3

Build Up Neural Network

3.1 Introduction

The target of the neural network is to function like an XOR gate. So we define the neural network like Fig.3.1. For the input layer, it will input True(+5V) or False(-5V). After the input layer, the signal will fully connect to the hidden layer. Each hidden node will sum up two signals after weighting. Then go through the activation function and output to the output layer. There has only one node in the output layer. Same as the hidden layer, the node in the output layer will sum up two signals after weighting and pass through the activation function. After the activation function will output the result(True or False) of the neural network.

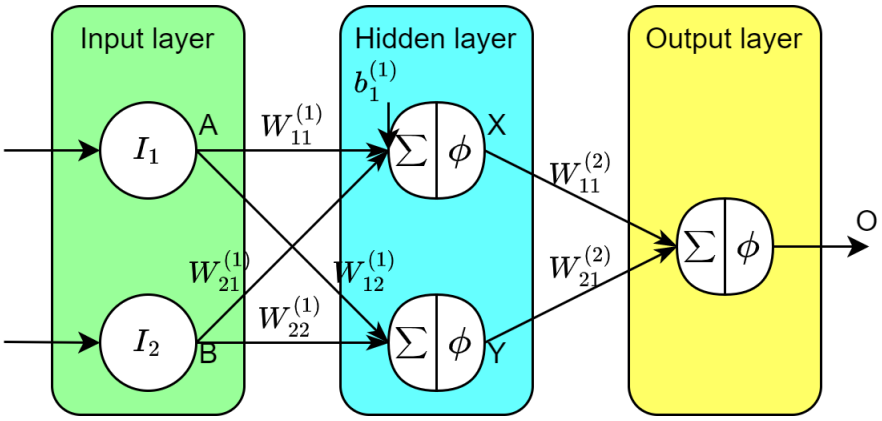


Figure 3.1: Neural network architecture

The weighting method is using a potentiometer to divide the voltage in Ch.2.2. So the weight value will between 0 and 1. And the summation method is summed up by circuit summing. From the result of Ch.2.1, the circuit summation actually is finding the average over all the input. And the activation function is using Op-Amp comparator in Ch.2.3. For Fig.3.2, the comparator will compare the input and the reference voltage. If the voltage is bigger then the reference voltage, then will output the positive supply. Otherwise, will output the negative supply.

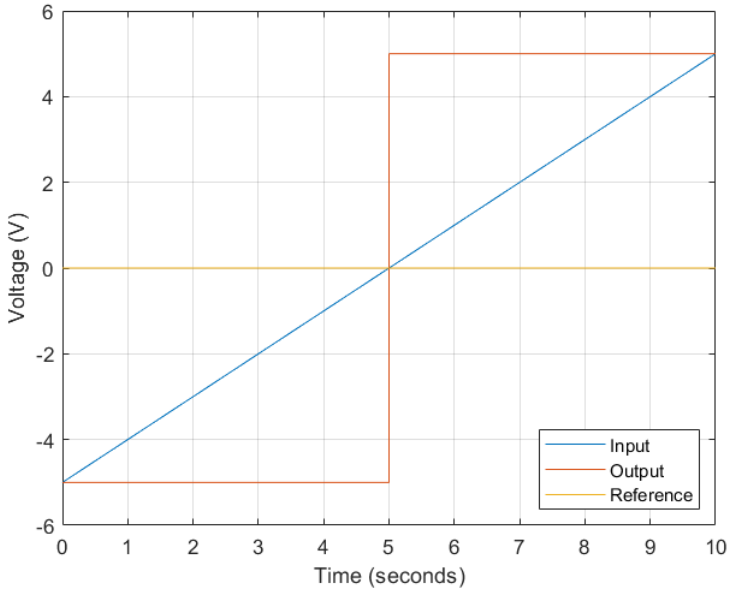


Figure 3.2: Op-Amp comparator activation function

For the beginning to build up the neural network, we need to build up the power system. The power system is like Fig.3.3. There have two different voltage for the electronic components. One is $+5V$ comes from 7805 regulator for positive power supply. Another is $-5V$ comes from 7905 regulator for negative power supply. This two power will use at create the signals and amplifier's power.

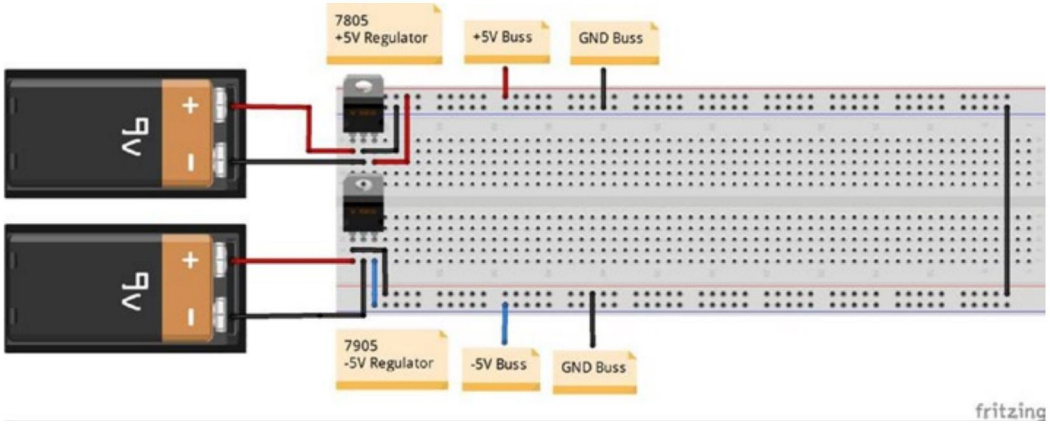


Figure 3.3: Protoboard power system

3.2 Input layer

The input layer node is present by Fig.3.4. The True and False signal is controlled by the SPDT switch. If the switch connects to the positive power supply, the input layer will output a True signal. Otherwise, if the switch connects to the negative power supply, the input layer will output a False signal. And we can see that there has a branch after the SPDT switch. We will connect a LED to display the state of input signal. If the LED is on, the output is True. Otherwise, the output will be False.

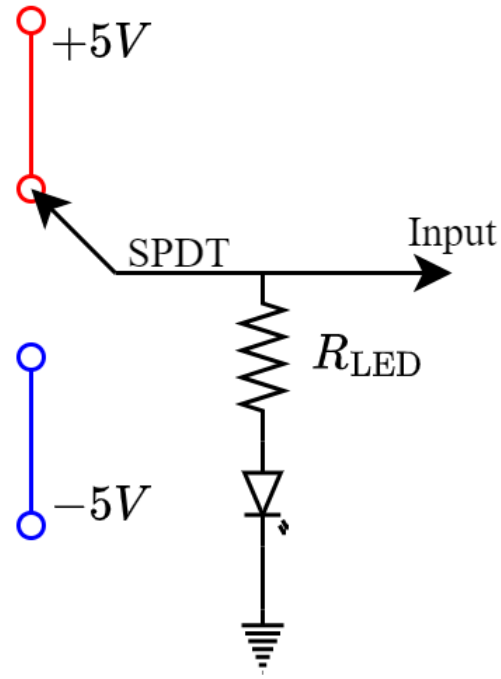


Figure 3.4: Input node circuit

After adding the input layer on the protoboard will be like Fig.3.5.

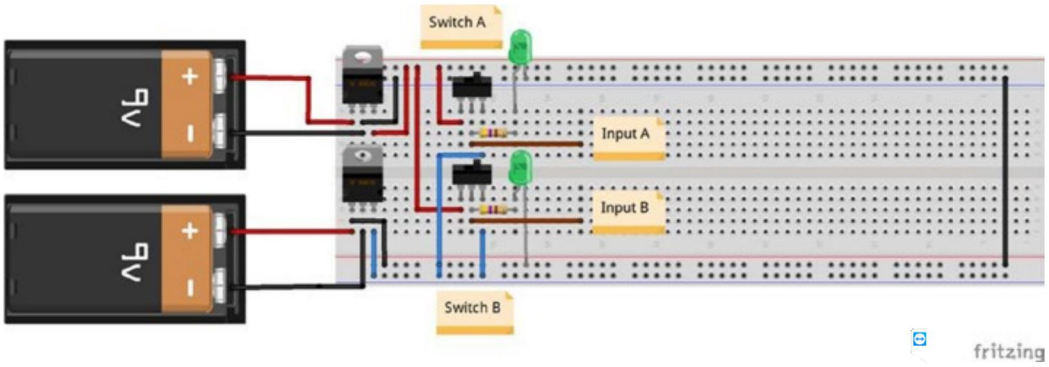


Figure 3.5: Input layer on the protoboard

3.3 Hidden layer

The hidden layer node is present by Fig.3.6 and Fig.3.7. For the first node in hidden layer will connect to two signal from input layer and a bias signal. And the secend node will connect to two signals from input layer. First, those signals will pass through a potentiometer to divide voltage. Next, connect two signals to two 100KΩ resistor and connect it together to get the average signals. Last after sum up, we will connect to the comparator as activation function then output.

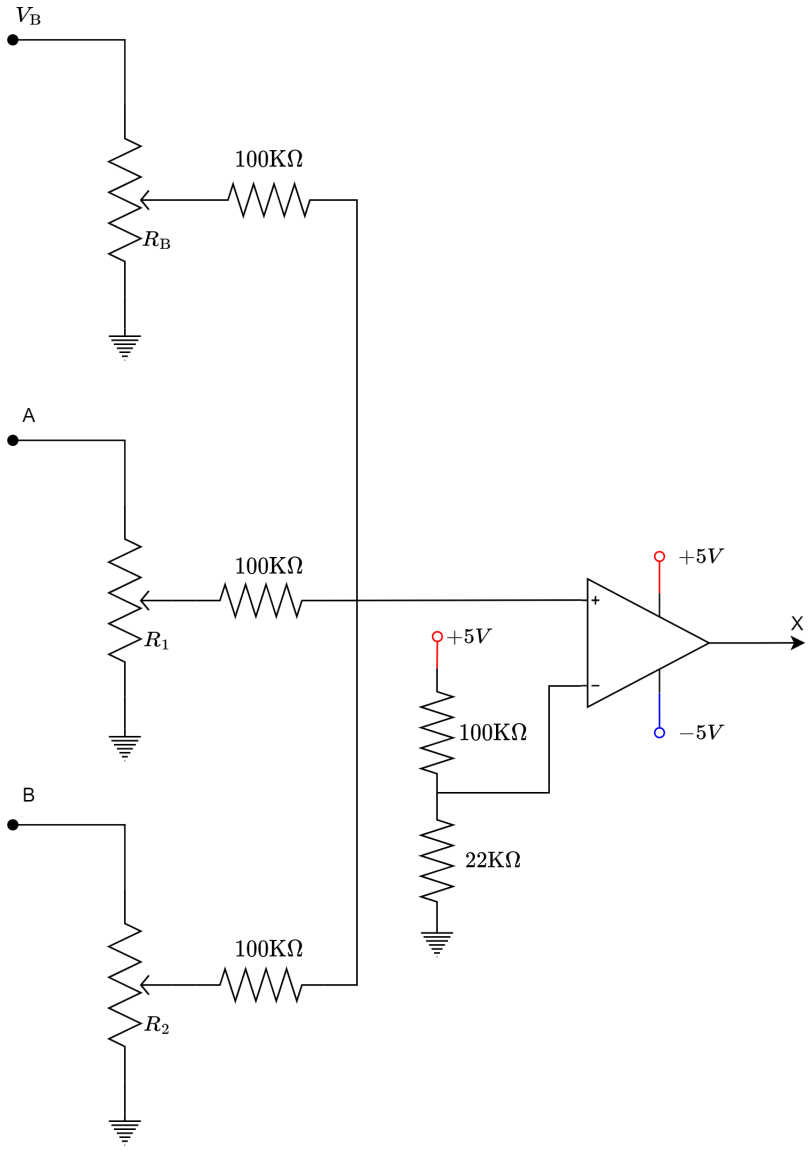


Figure 3.6: First hidden node circuit

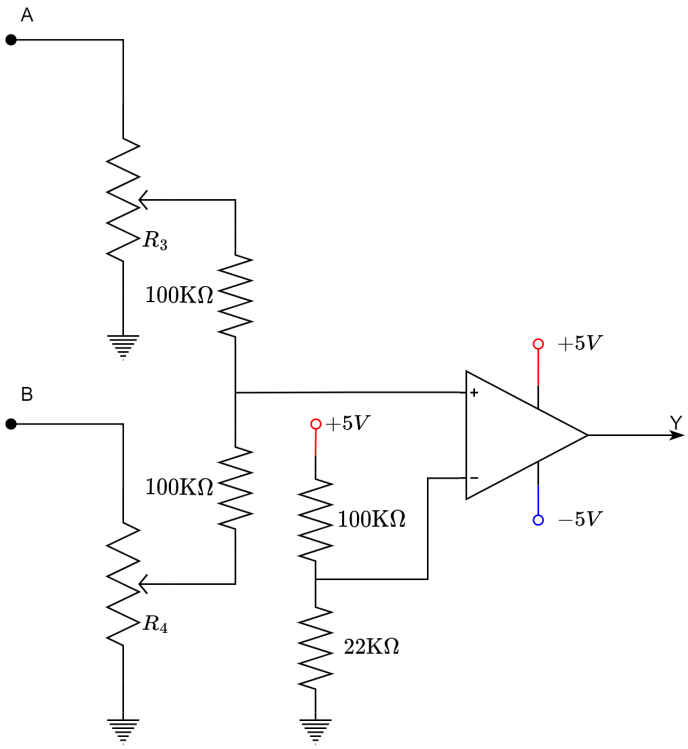


Figure 3.7: Secend hidden node circuit

After adding the hidden layer on the protoboard will be like Fig.3.8.

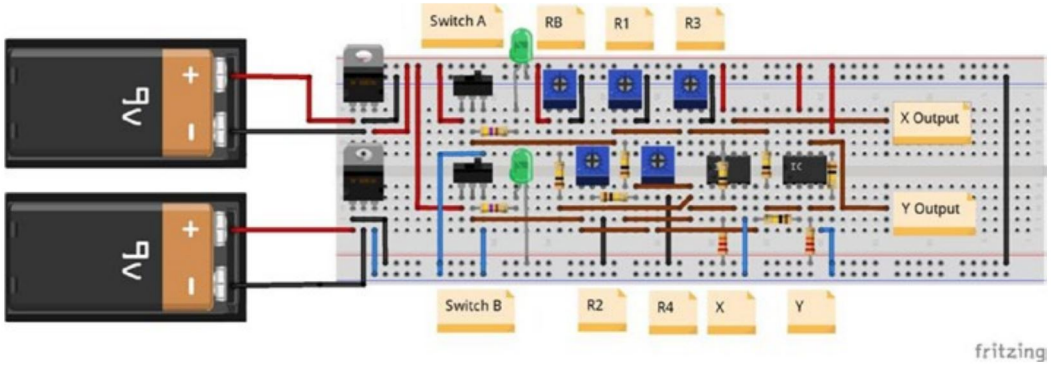


Figure 3.8: Hidden layer on the protoboard

3.4 Output layer

The output layer is present by Fig.3.9. For the output layer is similar to the hidden node. The different is the input signal X and Y is from the output of hidden layer. And the output will connect to a LED system to display the result.

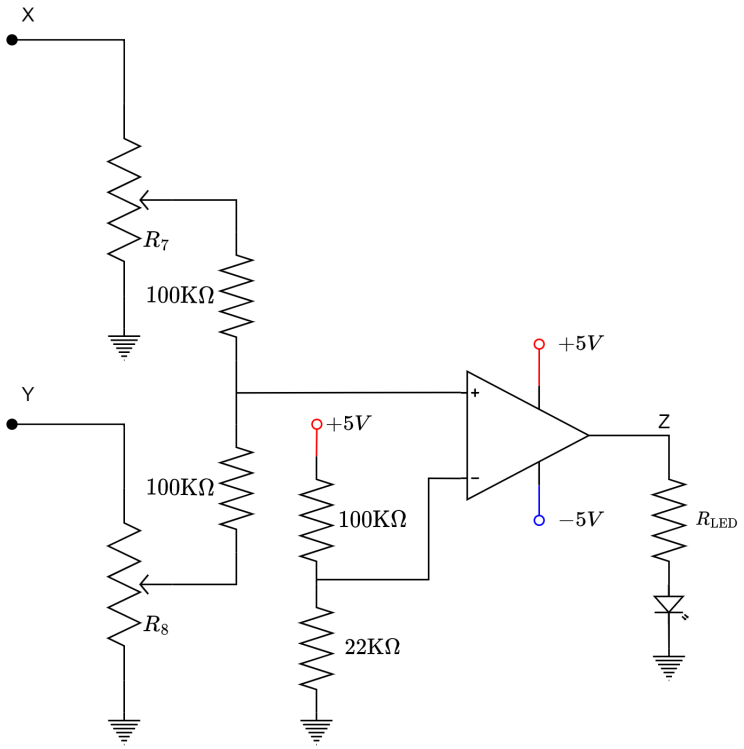


Figure 3.9: Output layer circuit

After adding the output layer on the protoboard will be like Fig.3.10.

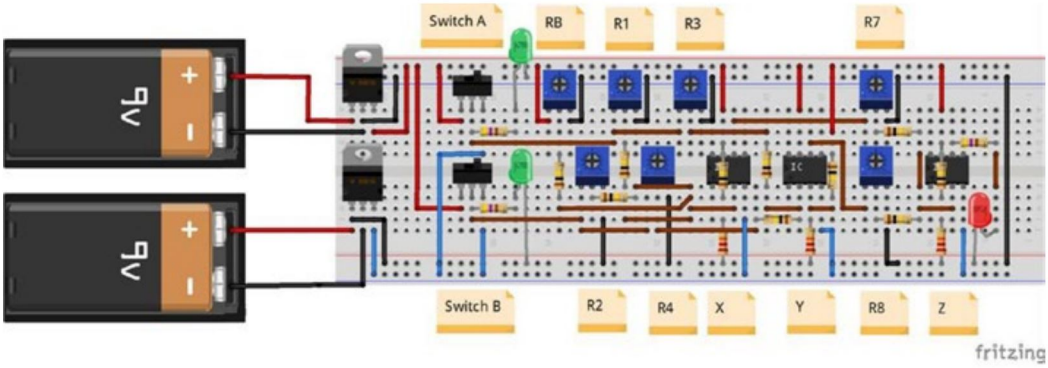


Figure 3.10: Output layer on the protoboard

3.5 Summary

From Fig.3.1 we can know that the neural network model is

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \phi\left(\begin{bmatrix} W_{11}^{(1)} & W_{12}^{(1)} \\ W_{21}^{(1)} & W_{22}^{(1)} \end{bmatrix}\begin{bmatrix} A \\ B \end{bmatrix} + \begin{bmatrix} b_1^{(1)} \\ 0 \end{bmatrix}\right) = \phi\left(v^{(1)}\right) \tag{3.1}$$

$$Z = \phi\left(\begin{bmatrix} W_{11}^{(2)} & W_{12}^{(2)} \end{bmatrix}\begin{bmatrix} X \\ Y \end{bmatrix}\right) = \phi\left(v^{(2)}\right) \tag{3.2}$$

A and B are the input signal, X and Y are the output signal of hidden layer. The weight and the bias of hidden layer is

$$W^{(1)} = \begin{bmatrix} \frac{R_1}{3\times100} & \frac{R_2}{3\times100} \\ \frac{R_3}{2\times100} & \frac{R_4}{2\times100} \end{bmatrix} \tag{3.3}$$

$$b^{(1)} = \begin{bmatrix} \frac{R_\mathrm{b}}{3\times100} \\ 0 \end{bmatrix} \tag{3.4}$$

the term of $\frac{1}{2}$ or $\frac{1}{3}$ is taking average of each node, and the term of $\frac{1}{100}$ is from voltage divide. The weight of output layer is

$$W^{(2)} = \frac{1}{2 \times 100} \begin{bmatrix} R_7 & R_8 \end{bmatrix} \tag{3.5}$$

For the value of resistors, $R_\mathrm{n} \in [0, 100]$. And the activation function is

$$\phi(x) = \begin{cases} +5, & \text{if } x > \frac{5\times22}{122} \\ -5, & \text{if } x < \frac{5\times22}{122} \\ 0, & \text{if } x = \frac{5\times22}{122} \end{cases} \tag{3.6}$$

Chapter 4

Training Neural Network

4.1 Weight Tuning

4.1.1 Back prapagation

For easier to analyse the neural network, we use the hyperbolic tangent function to approximate the comparator. The approximate function of (3.6) is

$$\bar{\phi}(x) = 5\tanh\left(10\left(x - \frac{5\times 22}{122}\right)\right) \tag{4.1}$$

The derivative of $\bar{\phi}$ is

$$\bar{\phi}'(x) = 50\left(1 - \tanh^2\left(10\left(x - \frac{5\times 22}{122}\right)\right)\right) \tag{4.2}$$

After that we can rewrite the neural network model become

$$y^{(1)} = \bar{\phi}\left(W^{(1)}x + b^{(1)}\right) = \bar{\phi}\left(v^{(1)}\right) \tag{4.3}$$

$$y^{(2)} = \bar{\phi}\left(W^{(2)}y^{(1)}\right) = \bar{\phi}\left(v^{(2)}\right) \tag{4.4}$$

where x is the input of neural network, $y^{(1)}$ is the output of hidden layer, and $y^{(2)}$ is the output of neural network.

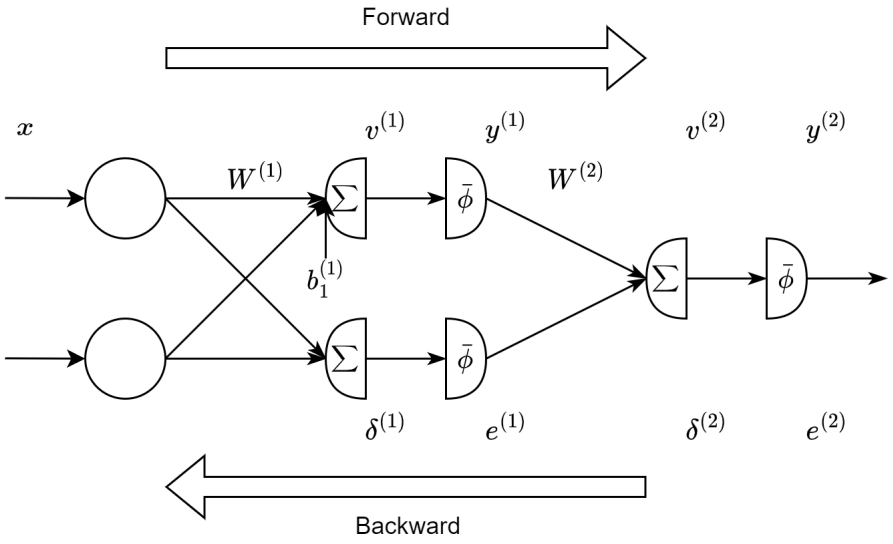


Figure 4.1: Neural network analysis

Define the loss function is square function

$$J = \frac{1}{2}(d - y^{(2)})^2 = \frac{1}{2}e^2 \tag{4.5}$$

d is the currect value in the dataset. For the output layer,

$$\frac{\partial J}{\partial W^{(2)}} = -\delta^{(2)} y^{(1)^T} \tag{4.6}$$

$$= -\bar{\phi}'(v^{(2)}) e^{(2)} y^{(1)^T} \tag{4.7}$$

The update rule of output layer will be

$$W^{(2)} \leftarrow W^{(2)} + \alpha \delta^{(2)} y^{(1)^T} \tag{4.8}$$

For the hidden layer,

$$\begin{aligned}
\frac{\partial J}{\partial W^{(1)}} &= -\delta^{(1)} x^T \\
&= -\begin{bmatrix} \bar{\phi}'(v_1^{(1)}) & 0 \\ 0 & \bar{\phi}'(v_2^{(1)}) \end{bmatrix} e^{(1)} x^T \\
&= -\begin{bmatrix} \bar{\phi}'(v_1^{(1)}) & 0 \\ 0 & \bar{\phi}'(v_2^{(1)}) \end{bmatrix} W^{(2)^T} \delta^{(2)} x^T \\
&= -\begin{bmatrix} \bar{\phi}'(v_1^{(1)}) & 0 \\ 0 & \bar{\phi}'(v_2^{(1)}) \end{bmatrix} W^{(2)^T} \bar{\phi}'(v^{(2)}) e^{(2)} x^T
\end{aligned} \tag{4.9}$$

$$\begin{aligned}
\frac{\partial J}{\partial b^{(1)}} &= -\delta^{(1)} \\
&= -\begin{bmatrix} \bar{\phi}'(v_1^{(1)}) & 0 \\ 0 & \bar{\phi}'(v_2^{(1)}) \end{bmatrix} e^{(1)} \\
&= -\begin{bmatrix} \bar{\phi}'(v_1^{(1)}) & 0 \\ 0 & \bar{\phi}'(v_2^{(1)}) \end{bmatrix} W^{(2)^T} \delta^{(2)} \\
&= -\begin{bmatrix} \bar{\phi}'(v_1^{(1)}) & 0 \\ 0 & \bar{\phi}'(v_2^{(1)}) \end{bmatrix} W^{(2)^T} \bar{\phi}'(v^{(2)}) e^{(2)}
\end{aligned} \tag{4.10}$$

In our case, $\bar{\phi}'(v_2^{(1)}) = 0$ because we don't have secend bias. Finally, the update rule of hidden layer will be

$$W^{(1)} \leftarrow W^{(1)} + \alpha \delta^{(1)} x^T \tag{4.11}$$

$$b^{(1)} \leftarrow b^{(1)} + \alpha \delta^{(1)} \tag{4.12}$$

Appendix A

Multi-input summation circuit

For multi-input case of Fig.2.1,

$$\sum_{n=1}^{N} I_n = 0 \tag{A.1}$$

replace I_n by Ohm's law.

$$\begin{aligned} &\sum_{n=1}^{N} \frac{V_{\text{out}} - V_n}{R_n} = 0 \\ \Rightarrow\ & V_{\text{out}} \sum_{n=1}^{N} \frac{1}{R_n} = \sum_{n=1}^{N} \frac{V_n}{R_n} \end{aligned} \tag{A.2}$$

For the summation of left-hand side,

$$\begin{aligned} \sum_{n=1}^{N} \frac{1}{R_n} &= \frac{1}{R_1} + \frac{1}{R_2} + \cdots + \frac{1}{R_n} \\ &= \frac{R_1 + R_2 + \cdots + R_n}{R_1 R_2 \cdots R_n} = \frac{\sum_{n=1}^{N} R_n}{\prod_{n=1}^{N} R_n} \end{aligned} \tag{A.3}$$

subtitude (A.3) into (A.2)

$$\begin{aligned} V_{\text{out}} &= \frac{\prod_{n=1}^{N} R_n \times \sum_{n=1}^{N} \frac{V_n}{R_n}}{\sum_{n=1}^{N} R_n} \\ &= \frac{\sum_{n=1}^{N} (\frac{1}{R_n} \prod_{m=1}^{N} R_m) V_n}{\sum_{n=1}^{N} R_n} \end{aligned} \tag{A.4}$$

Then the result is the formula of multi-input summation.

Appendix B

Back propagation for bias

From the update rule,

$$b_k^{(1)} \leftarrow b_k^{(1)} - \alpha \frac{\partial J}{\partial b_k^{(1)}} \tag{B.1}$$

The gradent of loss function w.r.t. the bias can be split by Chain rule.

$$\frac{\partial J}{\partial b_k^{(1)}} = \frac{\partial J}{\partial v_k} \underbrace{\frac{\partial v_k^{(1)}}{\partial b_k^{(1)}}}_{=1} = \frac{\partial J}{\partial v_k^{(1)}} \tag{B.2}$$

The gradent of loss function w.r.t. the output of hidden layer summation is

$$\begin{aligned} \frac{\partial J}{\partial v_k^{(1)}} &= \frac{\partial J}{\partial y_k^{(1)}} \frac{\partial y_k^{(1)}}{\partial v_k^{(1)}} \\ &= -e_k^{(1)} \bar{\phi}'(v_k^{(1)}) \triangleq -\delta_k^{(1)} \end{aligned} \tag{B.3}$$

The gradent of loss function w.r.t. the output of hidden layer is

$$\begin{aligned} \frac{\partial J}{\partial y_k^{(1)}} &= \frac{\partial J}{\partial v^{(2)}} \frac{\partial v^{(2)}}{\partial y_k^{(1)}} \\ &= -(\delta^{(2)} w_{1k}^{(2)}) \triangleq -e_k^{(1)} \end{aligned} \tag{B.4}$$

Substitude (B.2)~(B.4) into (B.1)

$$b_k^{(1)} \leftarrow b_k^{(1)} + \alpha \delta_k^{(1)} \tag{B.5}$$

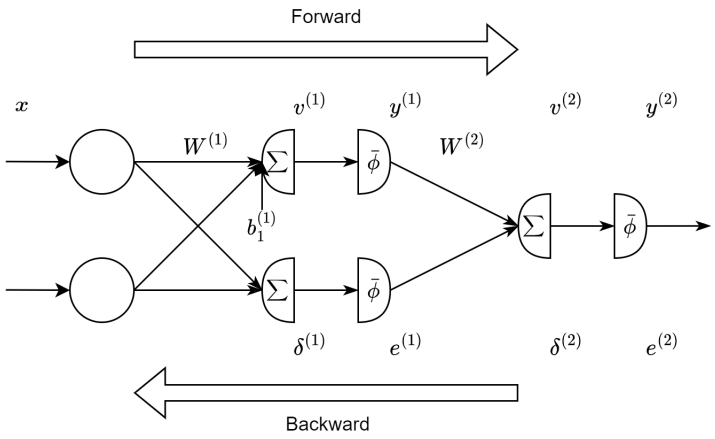


Figure B.1: Neural network analysis