Motivation/Rationale

Artificial Neural Networks are developed from mathematical models of biological neurons. Various models of biological neuron exist. A model generally describes the structure of biological neuron, function, and learning mechanism. To make a model more usable, primitive features of biological neuron are modeled and reused to make complex ANNs.

Syllabus

Unit 2 – Neural Network Architecture **[_ hrs.]**

2.1. Biological Neural Networks (structure, activation, lateral inhibition)
2.2. Learning mechanism

NEURAL NETWORK ARCHITECTURE

Biological Neural Networks

- Network of neurons
- Connected by dendrites and synapses

Structure of the neurons

A single neuron consists of four elements;

- Dendrites
- Synapses
- Cell body
- Axon

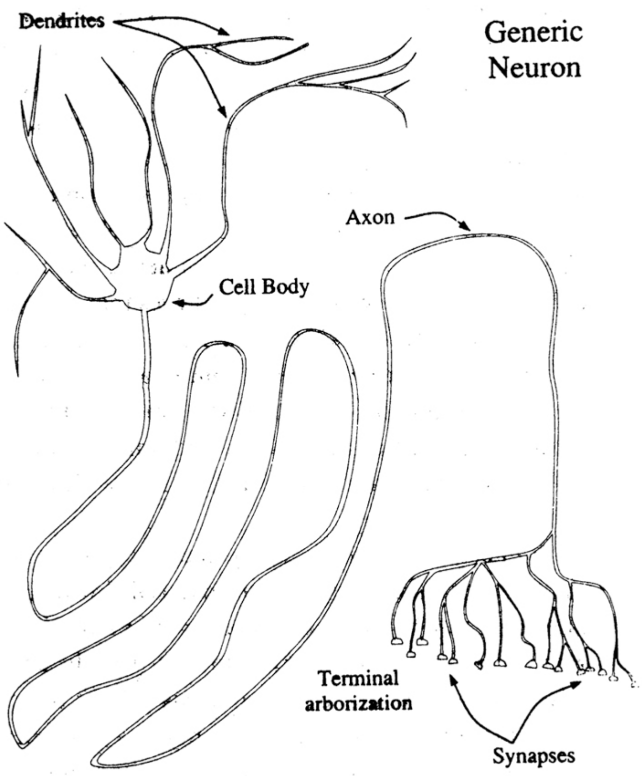

Fig. 1: A typical biological neuron

- Neurons receive signals through dendrites and produce a response through axon.
- Dendrites receive the signals at the contact regions with other cells, the synapses.

Synapse
- Two types: electrical and chemical
- Electrical synapse – 200 m/sec
- Chemical synapse – movement of Na^+, K^+, 20 m/sec
- Considered to be holding 8 – 10 bits of information
- 10 billion neurons in human body, every neuron is connected to other 5000-10000 neurons

Two types of operation by neuron
- activation – fires some output (spikes)
- lateral inhibition – copy signal to the next neuron with some resistance

Transmission
The fundamental problem of any information processing system is the transmission of information, as data storage can be transformed into a recurrent transmission of information between two points.

Some salts, present in our body, dissolve in the intra-cellular and extra cellular fluid and dissociate into negative and positive ions. Sodium-chloride, for example, dissociates into Positive sodium ions ($Na+$) and negative chlorine ions ($Cl-$). Other positive ions present in the interior or exterior of the cells are potassium ($K+$) and calcium($Ca2+$). The membranes of the cells exhibit different degrees of permeability for each one of these ions. The permeability is determined by the number and size of pores in the membrane, the so-called ionic channels. These are macromolecules with forms and charges which allow only certain ions to go from one side of the cell membrane to the other. Channels are selectively permeable to sodium, potassium or calcium ions. The specific permeability of the membrane leads to different distributions of ions in the interior and the exterior of the cells and this, in turn, to the interior of neurons being negatively charged with respect to the extracellular fluid.

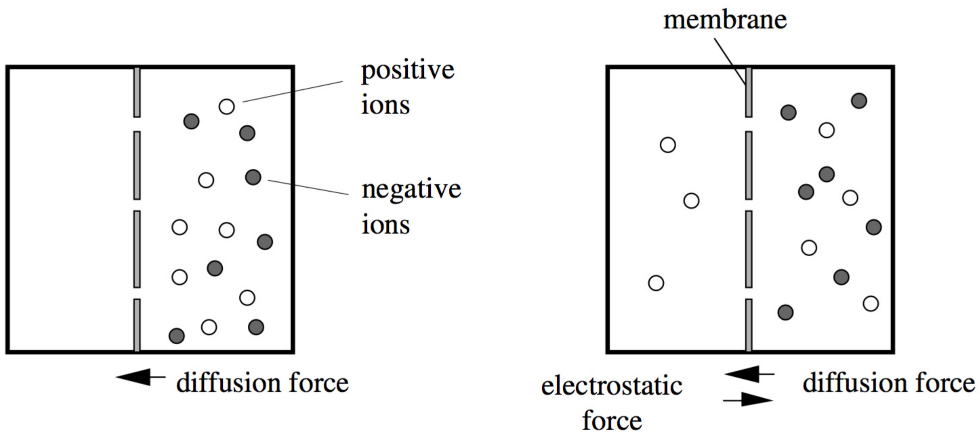

Fig. 2: Diffusion of ions through a membrane

The membrane is permeable only to positive ions. Initially the same number of positive and negative ions is located in the right side of the box. Later, some positive ions move from the right to the left through the pores in the membrane. This occurs because atoms and molecules have a thermodynamical tendency to distribute homogeneously in space by the process called diffusion. The process continues until the electrostatic repulsion from the positive ions on the left side balances the diffusion potential. A potential difference, called the ***reversal potential***, is established and the system behaves like a small electric battery. In a cell, if the initial concentration of potassium ions in its interior is greater than in its exterior, positive potassium ions will diffuse through the open potassium-selective channels. If these are the only ionic channels, negative ions cannot disperse through the membrane. The interior of the cell becomes negatively charged with respect to the exterior, creating a potential difference between both sides of the membrane.

Because there are several different concentrations of ions inside and outside of the cell, the question is, what is the potential difference which is finally reached. The exact potential in the interior of the cell depends on the mixture of concentrations. A typical cell's potential is −70mV, which is produced mainly by the ion concentrations shown in figure below. (A^- designates negatively charged biomolecules). The two main ions in the cell are sodium and potassium.

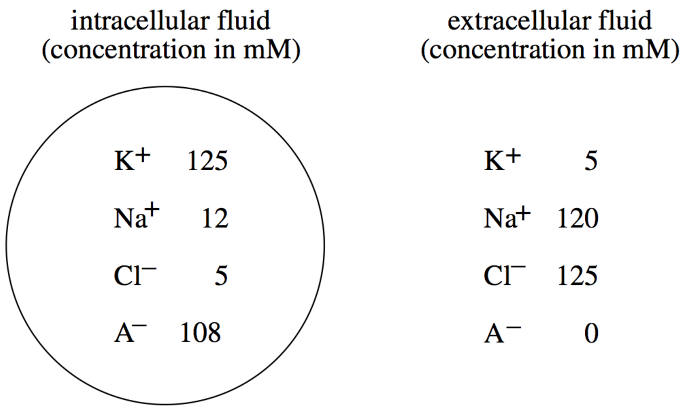

Fig. 3 : Ion concentrations inside and outside a cell

Electronic model of cell membrane

The British scientists Alan Hodgkin and Andrew Huxley were able to show that it is possible to build an electric model of the cell membrane based on very simple assumptions. The membrane behaves as a capacitor made of two isolated layers of lipids. It can be charged with positive or negative ions. The different concentrations of several classes of ions in the interior and exterior of the cell provide an energy source capable of negatively polarizing the interior of the cell.

The specific permeability of the membrane for each class of ion can be modeled like a conductance (the reciprocal of resistance).

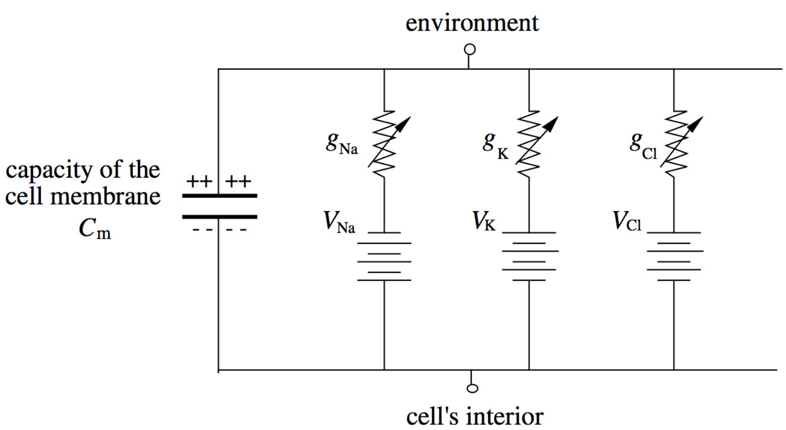

Fig. 4: The Hodgkin-Huxley model of a cell membrane

- The conductances gNa, gK, andgL reflect the permeability of the membrane to sodium, potassium, and leakages, i.e., the number of open channels of each class.
- A signal can be produced by modifying the polarity of the cell through changes in the conductance's gNa and gK.
- By making gNa larger and the mobility of sodium ions greater than the mobility of potassium ions, the polarity of the cell changes from −70mV to a positive value, nearer to

the 58mV at which sodium ions reach equilibrium. If the conductance gK then becomes larger and gNa falls back to its original value, the interior of the cell becomes negative again, over shooting in fact by going below −70mV.

- To generate a signal, a mechanism for depolarizing and polarizing the cell in a **controlled way** is necessary. Electrically controlled ionic channels perform this.

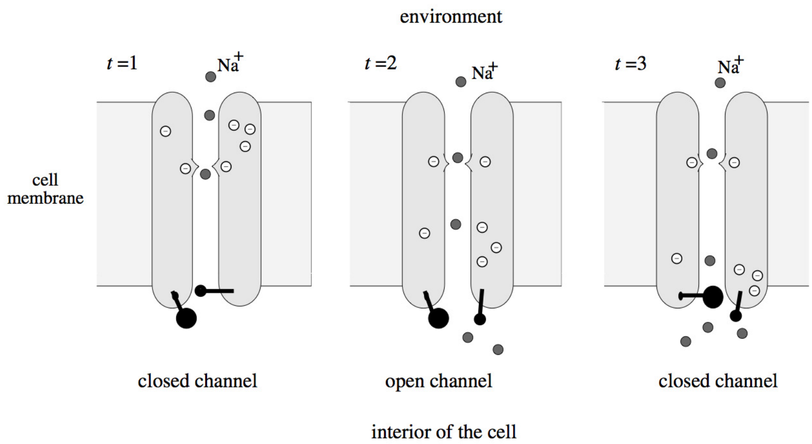

Fig. 5: Electrically controlled ionic channels

Action potential

Neural signals are produced and transmitted at the cell membrane. The signals are represented by depolarization waves traveling through the axons in a self-regenerating manner. Following figure shows the form of such a depolarization wave, called an action potential. The x-dimension is shown horizontally and the diagram shows the instantaneous potential in each segment of the axon.

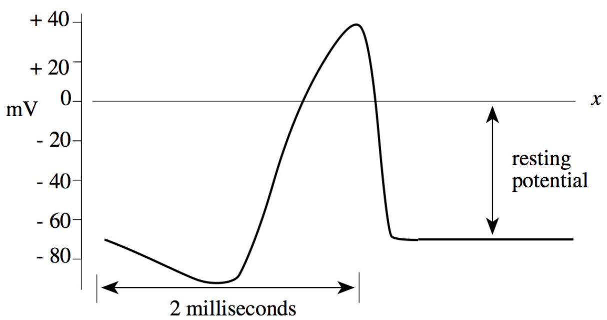

Fig. 6 : Action potential

The action potential traverses through the axon.

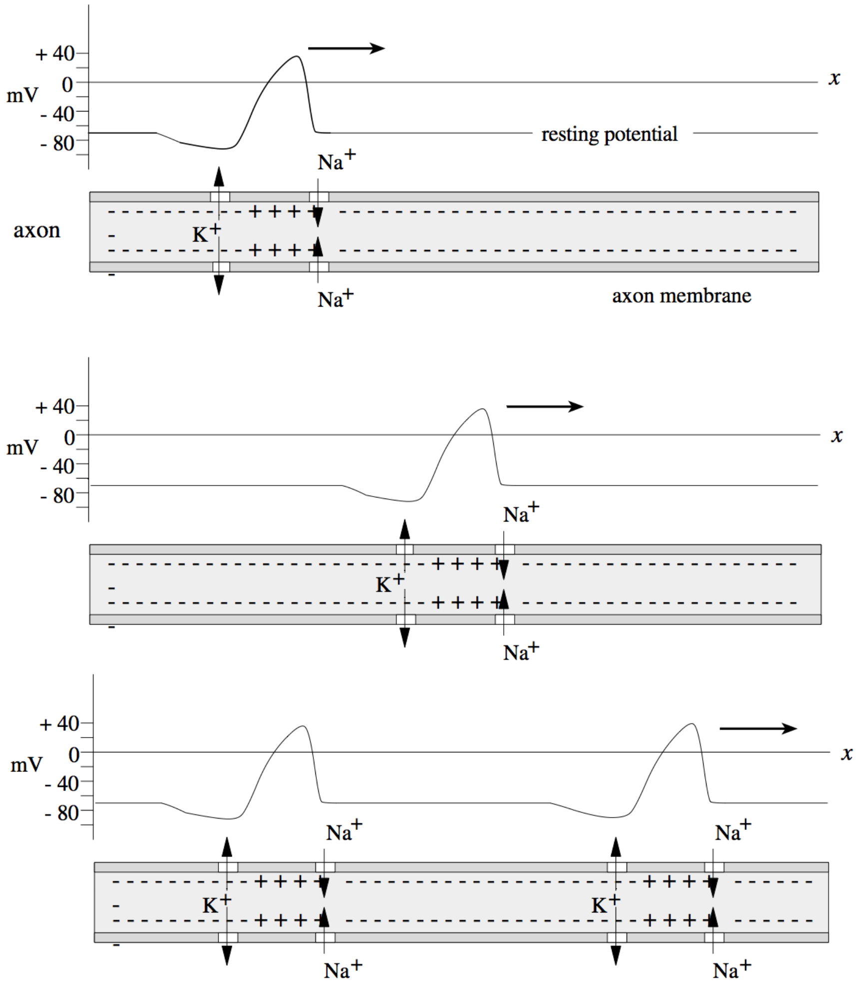

Fig. 7: Transmission of action potentials

The instantaneous variation of the cell's potential V can be described as a function of the conductance's of sodium, potassium and leakages (gNa, gK, gL) and of the equilibrium potentials for all three groups of ions called VNa, VK and VL with respect to the current potential:

$$\frac{dV}{dt} = \frac{1}{C_{\mathrm{m}}}(I - g_{\mathrm{Na}}(V - V_{\mathrm{Na}}) - g_{\mathrm{K}}(V - V_{\mathrm{K}}) - g_{\mathrm{L}}(V - V_{\mathrm{L}}))$$

In this equation Cm is the capacitance of the cell membrane. The terms (V−VNa), (V−VK), (V−VL) are the electromotive forces acting on the ions. Any variation of the conductance's translates into a corresponding variation of the cell's potential V. The variations of gNa and gK are given by differential equations which describe their oscillations. The conductance of the leakages, gL, can be taken as a constant.

The cable equation (why neurons are slow?)

Refer to the section 'Why are neurons so slow?' in Anderson, A. J., *An Introduction to Neural Networks*, page 25.

Information processing (activity)

A neuron codes its level of activity by adjusting the frequency of the generated impulses. This frequency is greater for a greater stimulus. The information is transmitted from cell to cell using frequency modulation. This form of transmission helps to increase the accuracy of the signal and to minimize the energy consumption of the cells.

Processing of neuron - activation

- The neuron acts like an **integrator** in accumulating the effect of input spikes.
- The accumulation cannot go on forever:
    - o If the accumulation is sufficiently high in a short time period, the neuron spikes - **Activation**
    - o Over a longer time period without spiking, the neuron gradually **leaks** the built-up potential.

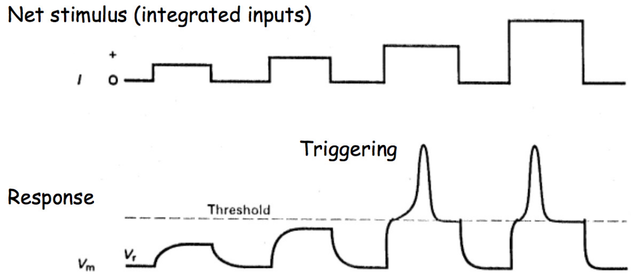

Fig. 8 : Action potential triggering phenomenon

- Because of the triggering behavior, the neuron indicates intensity of stimulation by the **frequency** of spikes, not amplitude.
- Because of the necessary refractory period, there is a maximum **saturation** frequency at which the neuron can operate.

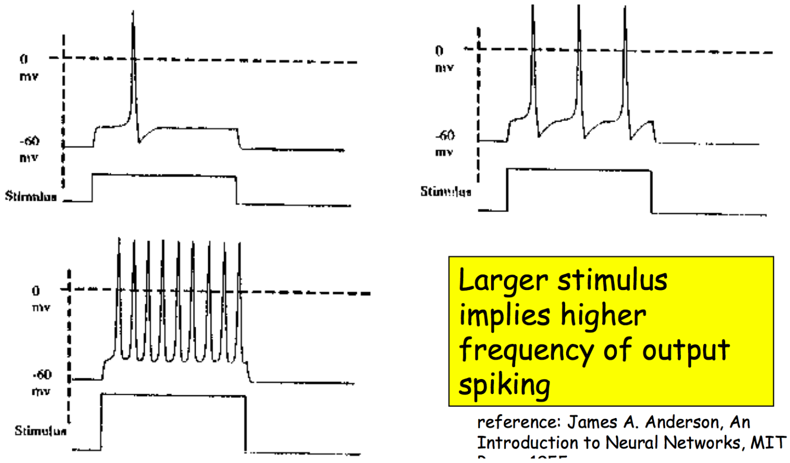

Fig. 9: Spiking frequency

- The flow of information from one neuron to another is typically based on the **rate of spiking**. The higher the rate, the more active the neuron.

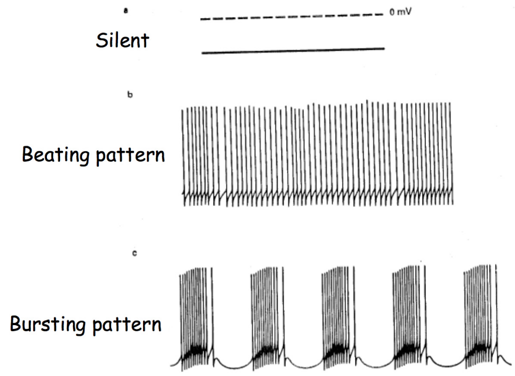

Fig. 10 : Different patterns of spiking

- We abstract this rate into a **single number**, as if transmitted in a single instant.

Signal abstractions
- We've already mentioned that signals can be abstracted into a single real number.
- We sometimes further abstract into a two-valued set, such as {-1, +1} or {0, 1}, depending on the intended application.

Lateral inhibition
- The sensory neurons sense the signal and transmit over other neurons.
- Due to massive connectivity of neurons, some of the signal returns back to the same neuron.
- The weight of inhibition from a neuron some distance away from a target neuron is given by;

$$inhibitory\_weight = \max\_strength * e^{-(Distance / length\_constant)}$$

where, max_strength : maximum amount of inhibition
distance: the distance
length_constant: approximation

Storage of Information - Learning
Neural networks learn by controlling the flow of ions through synapse and changing the threshold in the neuron.

The ionic channels permeable for different kinds of molecules, like sodium, calcium, or potassium ions. These channels are blocked by a magnesium ion in such away that the permeability for sodium and calcium is low. If the cell is brought up to a certain excitation level, the ionic channels lose the magnesium ion and become unblocked. The permeability for Ca^{2+} ions increases immediately. Through the flow of calcium ions a chain of reactions is started which produces a durable change of the threshold level of the cell. Following fiture shows the process.

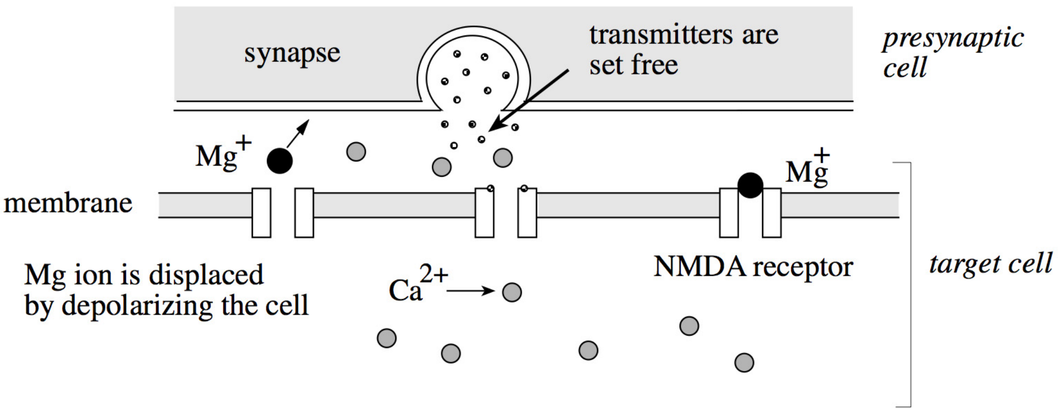

Fig. 11 : Synaptic Learning

The mechanism explained above increase their **plasticity** i.e., their **adaptability** to changing circumstances. Through the modification of the membrane's permeability, a cell can be trained to fire more often by setting a lower firing threshold. The stored information must be refreshed periodically in order to maintain the optimal permeability of the cell membrane.

This explanation is the foundation for artificial neural networks. The storage of information results into two functions;

- Learning
    - Error corrective learning

- Memory-based learning
- Hebbian learning
- Competitive learning
- Boltzman learning
- Supervised learning
- Unsupervised learning
- Reward-based learning (re-enforcement)

These learning mechanisms can be used to solve following tasks;

- Pattern association
- Pattern recognition
- Functional approximation
- Control
- Filtering
- Beamforming

- Memory – distributed approach
  - Associative memories
  - Correlation matrix memories
  - Recall

References

- Rojas, R., *Neural Networks – A Systematic Introduction*
- Anderson, J.A., *An Introduction to Neural Networks*
- Haykin, S., *Neural Networks – A Comprehensive Foundation*
- Keller, B., *Neural Networks, CS-152*

PRACTICAL # __

(May 31, 2009)

We'll verify your computer program for error corrective learning in computer lab.

Here's my program.

```
/* --------------------------------------------------------------------
 Program name: error corrective learning
 Description : This program demonstrates adaptation feature of a single
             perceptron using error corrective learning.
 By           : Pramod Parajuli
 Date         : May 24, 2009
------------------------------------------------------------------*/

#include <stdio.h>
#include <stdlib.h>

/* Let's write a function for uniform random number (0 - 1) generator. */
#define RANDOM_NUM (double) rand()/RAND_MAX

int main (int argc, char * argv[]){

          double x[2];          // the inputs
          double w[2];          // the weights
          double sum, target, error; //sum = i * w, target, error = target-sum

          long epoch;           // defines no. of iterations
          long counter;

          double learning_rate; // learning rate of the perceptron

          srand(time(NULL));    // seed the random number generator to
                                // current time which is purely random

          x[0]= 2.0;            // setting up the inputs
          x[1] = 1.5;
          w[0]= RANDOM_NUM;     // assign the random weights
          w[1] = RANDOM_NUM;

          epoch = 10;           // let's train the perceptron for 10 times
          learning_rate = 0.1;  // learning rate
          target = 5.0;         // let's train the perceptron to sum up to 5.0

          printf("First input x[0]: %lf, second input x[1]: %lf\n",
                         x[0], x[1]);
          printf("First random weight: %lf, second random weight: %lf\n",
                         w[0], w[1]);

          sum = w[0] * x[0] + w[1] * x[1];
          error = target - sum;       // first error

          printf("Epoch: %ld, first sum: %lf, target: %lf, first error:
                   %lf\n\n", epoch, sum, target, error);

          for(counter = 0; counter < epoch; counter++){
             printf("Iteration no. %ld, w[0]: %lf, w[1]: %lf, sum: %lf, error:
                   %lf\n", counter, w[0], w[1], sum, error);

             // wi+1 = wi + x * e * alpha
```

```
            w[0] = w[0] + x[0] * error * learning_rate;
            w[1] = w[1] + x[1] * error * learning_rate;

            //  finding the new sum
            sum = w[0] * x[0] + w[1] * x[1];
            // calculating error for next iteration of learning
            error = target - sum;

        }

        printf("\npress any key to exit...");
        getc(stdin);

        return 0;
}
```

The output produced

```
bash-3.2$ ./mytest
First input x[0]: 2.000000, second input x[1]: 1.500000
First random weight: 0.853871, second random weight: 0.013394
Epoch: 10, first sum: 1.727834, target: 5.000000, first error: 3.272166

Iteration no. 0, w[0]: 0.853871, w[1]: 0.013394, sum: 1.727834, error: 3.272166
Iteration no. 1, w[0]: 1.508304, w[1]: 0.504219, sum: 3.772938, error: 1.227062
Iteration no. 2, w[0]: 1.753717, w[1]: 0.688279, sum: 4.539852, error: 0.460148
Iteration no. 3, w[0]: 1.845747, w[1]: 0.757301, sum: 4.827444, error: 0.172556
Iteration no. 4, w[0]: 1.880258, w[1]: 0.783184, sum: 4.935292, error: 0.064708
Iteration no. 5, w[0]: 1.893199, w[1]: 0.792890, sum: 4.975734, error: 0.024266
Iteration no. 6, w[0]: 1.898053, w[1]: 0.796530, sum: 4.990900, error: 0.009100
Iteration no. 7, w[0]: 1.899872, w[1]: 0.797895, sum: 4.996588, error: 0.003412
Iteration no. 8, w[0]: 1.900555, w[1]: 0.798407, sum: 4.998720, error: 0.001280
Iteration no. 9, w[0]: 1.900811, w[1]: 0.798599, sum: 4.999520, error: 0.000480

press any key to exit...
```