IT2024 Lecture Diary October 25, 2024

October 28, 2024

1 Outlines

(a) Contributions of John J. Hopfield and Geoffrey Hinton in *artificial neural networks*.

- Considered a brain as a giant network of bi-state neurons. The output of each neuron can only have value -1 and $+1$.
- Modelled a network of interconnected bi-state neurons as a physical system.
- From the properties of the network, Hopfield and Hinton provided clues on memory and learning.

(b) On survey.

(c) In the followings, the notation T has been used under two contexts. For a matrix $\mathbf{W}$ (resp. a vector $\mathbf{s}$), $\mathbf{W}^T$ (resp. $\mathbf{s}^T$) refers to the transpose of $\mathbf{W}$ (resp. $\mathbf{s}$). In a neuron model, either in the Hopfield network or Baltzmann machine, T refers to a positive constant called *temperature*.

2 On Hopfield Network (Single-Layer)

(a) Let $u_1(t), u_2(t), \cdots, u_n(t)$ be the inputs to the neurons at time t. $s_1(t), s_2(t), \cdots, s_n(t)$ be the outputs. The output of the i^{th} neuron in a Hopfield network is governed by the threshold logic function[1] $h(\cdot)$.

$$s_i(t) = h(u_i(t)) = \begin{cases} 1 & \text{if } u_i(t) \geq \tau_i, \\ -1 & \text{if } u_i(t) < \tau_i. \end{cases} \tag{2}$$

for $i = 1, \cdots, n$. The input $u_i(t)$ depends on the outputs at time $(t-1)$ and the synaptic weights associated with.

$$\begin{aligned} u_1(t) &= w_{11}s_1(t-1) + w_{12}s_2(t-1) + \cdots + w_{1n}s_n(t-1) \\ u_2(t) &= w_{21}s_1(t-1) + w_{22}s_2(t-1) + \cdots + w_{2n}s_n(t-1) \\ \vdots \quad & \quad \vdots \\ u_n(t) &= w_{n1}s_1(t-1) + w_{n2}s_2(t-1) + \cdots + w_{nn}s_n(t-1). \end{aligned}$$

[1] It should be noted that the neurons could be defined as a tri-state model.

$$s_i(t) = h(u_i(t)) = \begin{cases} 1 & \text{if } u_i(t) > \tau_i, \\ 0 & \text{if } u_i(t) = \tau_i, \\ -1 & \text{if } u_i(t) < \tau_i. \end{cases} \tag{1}$$

for $i = 1, \cdots, n$. In this regard, the total number of states is 3^n (instead of 2^n) for a Hopfield network with n neurons.

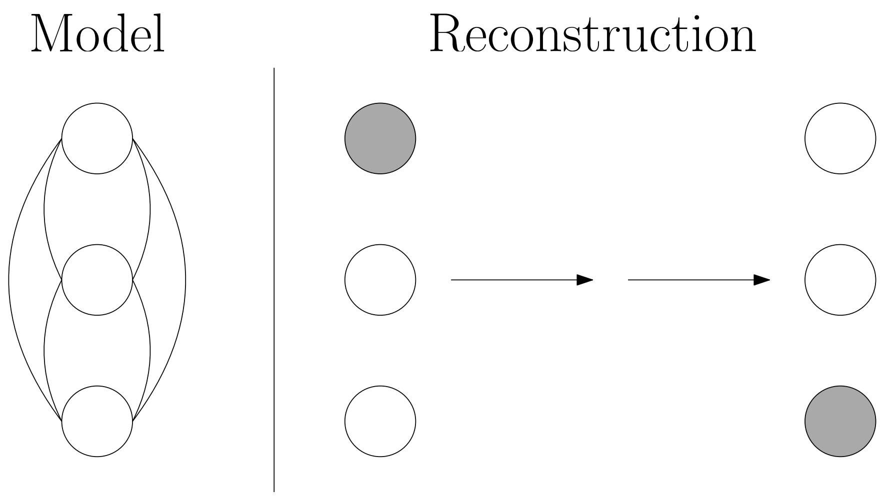


Figure 1: Pattern reconstruction (equivalently, memory recall) of a three-neuron Hopfield network.

In compact form, $\mathbf{u}(t) = \mathbf{W}\mathbf{s}(t-1) - \boldsymbol{\tau}$ and hence

$$\mathbf{s}(t) = \mathbf{h}\left(\mathbf{W}\mathbf{s}(t-1) - \boldsymbol{\tau}\right). \tag{3}$$

(b) Consider a Hopfield network with three neurons, as shown in Figure 1. The dynamics of the network is defined as follows :

$$\begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \end{bmatrix} = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} s_1(t-1) \\ s_2(t-1) \\ s_3(t-1) \end{bmatrix}, \text{ and } \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.5 \\ -0.5 \end{bmatrix}. \tag{4}$$

The outputs of the network at time t is again governed by (2).

Question 1: In accordance with the model (4), if $s_1(0) = 1$, $s_2(0) = s_3(0) = -1$, what will be the outputs of the network at times $t = 1, 2, 3$?

(c) Similarly, the dynamics of the network is changed to the following.

$$\begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \end{bmatrix} = \begin{bmatrix} 0 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} s_1(t-1) \\ s_2(t-1) \\ s_3(t-1) \end{bmatrix}, \text{ and } \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{5}$$

The outputs of the network at time t is again governed by (2).

Question 2: In accordance with the model (5), if $s_1(0) = 1$, $s_2(0) = s_3(0) = -1$, what will be the outputs of the network at times $t = 1, 2, 3, 4, 5$?

(d) As each neuron output can have three possible states, the total number of states of a three-neuron Hopfield network is 8, as shown in Figure 2.

(e) Now, the dynamics of the network is changed to the following.

$$\begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} s_1(t-1) \\ s_2(t-1) \\ s_3(t-1) \end{bmatrix}, \text{ and } \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{6}$$

The outputs of the network at time t is again governed by (2).

Question 3: In accordance with the model (6), if $s_1(0) = 1$, $s_2(0) = s_3(0) = -1$, what will be the outputs of the network at times $t = 1, 2, 3$?

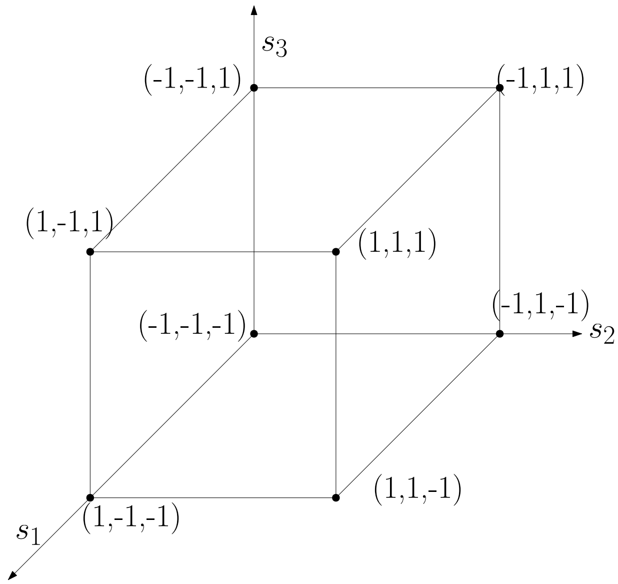


Figure 2: Eight states of a three-neuron Hopfield network.

(f) Next, the dynamics of the network is changed to the following. The weights are determined by the Hebbian learning rule which learns a pattern $(1, -1, -1)$.

$$\begin{bmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} s_1(t-1) \\ s_2(t-1) \\ s_3(t-1) \end{bmatrix}, \text{ and } \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{7}$$

The outputs of the network at time t is again governed by (2).

(g) Questions regarding $\mathbf{s}(0)$ close to the state $(1, -1, -1)^T$.

Question 4: In accordance with the model (7), if $s_1(0) = 1$, $s_2(0) = -1$ and $s_3(0) = -1$, what will be the outputs of the network at times $t = 1, 2$ and so on? It is going to see if the model (7) is able to memorize the pattern.

Question 5: In accordance with the model (7), if $s_1(0) = -1$, $s_2(0) = -1$ and $s_3(0) = -1$, what will be the outputs of the network at times $t = 1, 2$ and so on? It is going to see if the model (7) is able to reconstruct the memorized pattern from a noisy initial inputs.

Question 6: In accordance with the model (7), if $s_1(0) = 1$, $s_2(0) = 1$ and $s_3(0) = -1$, what will be the outputs of the network at times $t = 1, 2$ and so on? It is going to see if the model (7) is able to reconstruct the memorized pattern from a noisy initial inputs.

Question 7: In accordance with the model (7), if $s_1(0) = 1$, $s_2(0) = -1$ and $s_3(0) = 1$, what will be the outputs of the network at times $t = 1, 2$ and so on? It is going to see if the model (7) is able to reconstruct the memorized pattern from a noisy initial inputs.

(h) Questions regarding $\mathbf{s}(0)$ close to the state $(-1, 1, 1)^T$.

Question 8: In accordance with the model (7), if $s_1(0) = -1$, $s_2(0) = 1$ and $s_3(0) = 1$, what will be the outputs of the network at times $t = 1, 2$ and so on? It is going to see if the model (7) is able to reconstruct the memorized pattern from a noisy initial inputs.

Answer: Given $\mathbf{s}(0) = (-1, 1, 1)^T$ and $\boldsymbol{\tau} = (0, 0, 0)^T$, $(\mathbf{u}(1) - \boldsymbol{\tau})$ is given as follows :

$$\begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -3 \\ 3 \\ 3 \end{bmatrix}.$$

By (2), $\mathbf{s}(1) = (-1, 1, 1)^T$.

Question 9: In accordance with the model (7), if $s_1(0) = 1$, $s_2(0) = 1$ and $s_3(0) = 1$, what will be the outputs of the network at times $t = 1, 2$ and so on? It is going to see if the model (7) is able to reconstruct the memorized pattern from a noisy initial inputs.

Answer: Given $\mathbf{s}(0) = (1, 1, 1)^T$ and $\boldsymbol{\tau} = (0, 0, 0)^T$, $(\mathbf{u}(1) - \boldsymbol{\tau})$ is given as follows :

$$\begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}.$$

By (2), $\mathbf{s}(1) = (-1, 1, 1)^T$.

Question 10: In accordance with the model (7), if $s_1(0) = -1$, $s_2(0) = -1$ and $s_3(0) = 1$, what will be the outputs of the network at times $t = 1, 2$ and so on? It is going to see if the model (7) is able to reconstruct the memorized pattern from a noisy initial inputs.

Answer: Given $\mathbf{s}(0) = (-1, -1, 1)^T$ and $\boldsymbol{\tau} = (0, 0, 0)^T$, $(\mathbf{u}(1) - \boldsymbol{\tau})$ is given as follows :

$$\begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}.$$

By (2), $\mathbf{s}(1) = (-1, 1, 1)^T$.

Question 11: In accordance with the model (7), if $s_1(0) = -1$, $s_2(0) = 1$ and $s_3(0) = -1$, what will be the outputs of the network at times $t = 1, 2$ and so on? It is going to see if the model (7) is able to reconstruct the memorized pattern from a noisy initial inputs.

Answer: Given $\mathbf{s}(0) = (-1, 1, -1)^T$ and $\boldsymbol{\tau} = (0, 0, 0)^T$, $(\mathbf{u}(1) - \boldsymbol{\tau})$ is given as follows :

$$\begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}.$$

By (2), $\mathbf{s}(1) = (-1, 1, 1)^T$.

(i) In accordance with the state update equation (3), the network state $\mathbf{s}(t)$ might be cycling at a few states or get stuck at a state.

(j) Consider an energy function $E(s_1, \cdots, s_n)$ defined as follows for (3) :

$$E(\mathbf{s}|\mathbf{W}, \boldsymbol{\tau}) = -\frac{1}{2}\mathbf{s}^T\mathbf{W}\mathbf{s} - \boldsymbol{\tau}^T\mathbf{s} = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}s_i s_j - \sum_{i=1}^{n}\tau_i s_i. \tag{8}$$

By (8), we can get the energies for the states of the three-neuron Hopfield network (7).

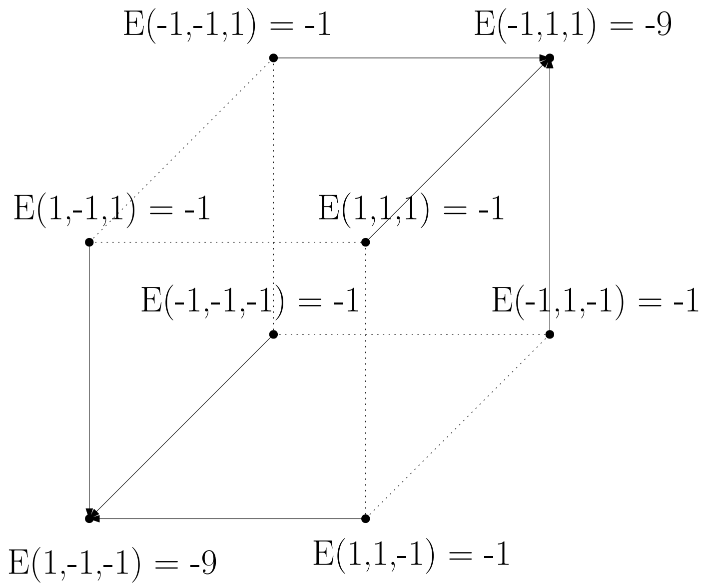


Figure 3: State update is equivalent to searching for the state with minimum energy. In accordance with the model 3, the energies E of both states $(1,-1,-1)$ and $(-1,1,1)$ are the minimum.

$E(1,-1,-1)$	-9	$E(-1,1,1)$	-9
$E(-1,-1,-1)$	-1	$E(1,1,1)$	-1
$E(1,1,-1)$	-1	$E(-1,-1,1)$	-1
$E(1,-1,1)$	-1	$E(-1,1,-1)$	-1

Thus, the state update is equivalent to the process of searching for the state $\mathbf{s}$ its energy $E(\mathbf{s})$ is a minima, as shown in Figure 3. Moreover, we can state the following theorem for the energies of states $\mathbf{s}$ and $-\mathbf{s}$.

Theorem 1 *If* $\boldsymbol{\tau} = \mathbf{0}$, $E(s_1, \cdots, s_n) = E_{(} - s_1, \cdots, -s_n)$.

(Proof.) Obvious from (8).

That is to say, Hopfield network will have the patterns $\mathbf{p}$ and $-\mathbf{p}$ in its memory once it has been trained to memorize $\mathbf{p}$.

(k) In accordance with the above property being analyzed by John J. Hopfield, Hopfield network could be designed for solving an optimization problem like production cost minimization. In which, a decision maker needs to decide which firms have to be selected so that the production cost is the minimum. If the cost function can be expressed in the form of (8), it is able to design a Hopfield network to solve the corresponding optimization problem which involves binary decision variables.

(l) It should be noted that optimization problems involving binary decision variables are well-known difficult problems. The worst-case time spent on solving this type of problems could be as long as $\alpha \times 2^n$, where α is the time spent on calculating the energy of a state $E(\mathbf{s})$. Figure 5 shows the worst-case computational time against the number of binary decision variables.

If $n = 10$ and $\alpha = 0.0001$, the worst-case computational time is about 0.1 second. If $n = 20$ and $\alpha = 0.0001$, the worst-case computational time is about 105 seconds. If $n = 50$ and $\alpha = 0.0001$, the worst-case computational time is about 3.57×10^3 years.

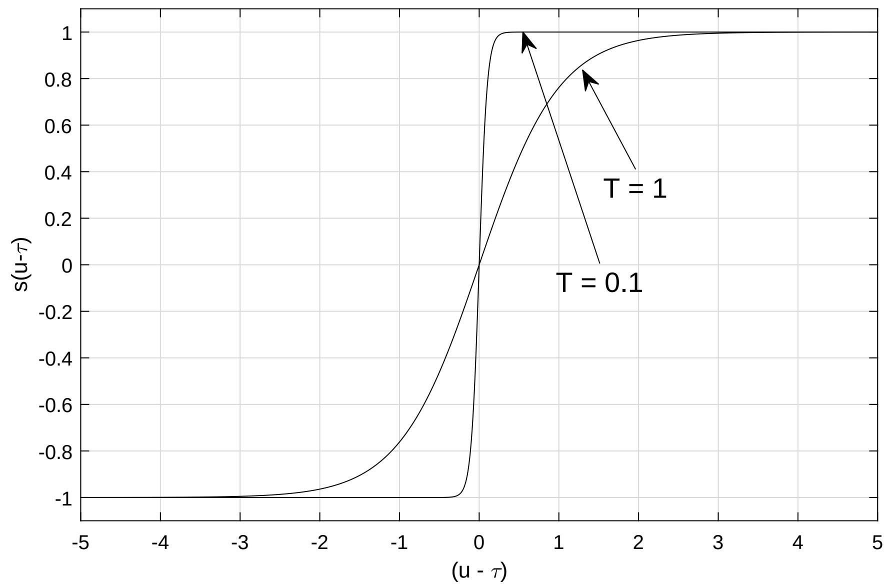


Figure 4: Hyperbolic tangent function for a neuron. Clearly, the function is very close to the deterministic neuron model as defined in (2) if $T \to 0$.

If $n = 100$ and $\alpha = 0.0001$, the worst-case computational time is 10^{26} seconds. It is equivalent to 4×10^{18} years. If $n = 100$ and $\alpha = 10^{-9}$, the worst-case computational time is still 10^{20} seconds. It is equivalent to 4×10^{12} years. It is thus hoped that mapping the optimization problem to a Hopfield network can largely reduce the time for getting the solution.

(m) Therefore, designing a Hopfield network for solving a quadratic optimization problem with binary decision variables could be a practical method. However, determining the parameters for a Hopfield network for solving an optimization problem, the discontinuous neuron model as defined (2) leads to a mathematical problem. $E(\mathbf{s})$ is not differentiable. Accordingly, John J. Hopfield later proposed another Hopfield network model in which the neuron model is defined by the smooth hyperbolic tangent function.

$$h(u_i - \tau_i) = \frac{\exp((u_i - \tau_i)/T) - \exp(-(u_i - \tau_i)/T)}{\exp((u_i - \tau_i)/T) + \exp(-(u_i - \tau_i)/T)}, \tag{9}$$

where T is set to be a small positive number, say $T = 0.01$. Figure 4 shows the shapes of (9) for $T = 1$ and $T = 0.1$.

Given a cost function of a *zero-one quadratic optimization problem*, it is thus able to derive the conditions for setting (resp. design) the connection matrix $\mathbf{W}$, the threshold vector $\boldsymbol{\tau}$ and the temperature T. Once the design has been obtained, an analog hardware Hopfield can be made to solve the problem.

(n) With the hyperbolic tangent model for a neuron, an n neurons Hopfield network could be implemented by an electronic circuit with n operational amplifier (OP-AMP) circuits. This electronic circuit performs analog computation in contrast to digital computation as in a digital computer. The electronic circuit could be applied in simulating the dynamic behavior of the differential equation of the form $d\mathbf{x}(t)/dt = \mathbf{f}(\mathbf{x}(t), t)$. This special electronic circuit with a number of Op-Amp is called the analog computer.

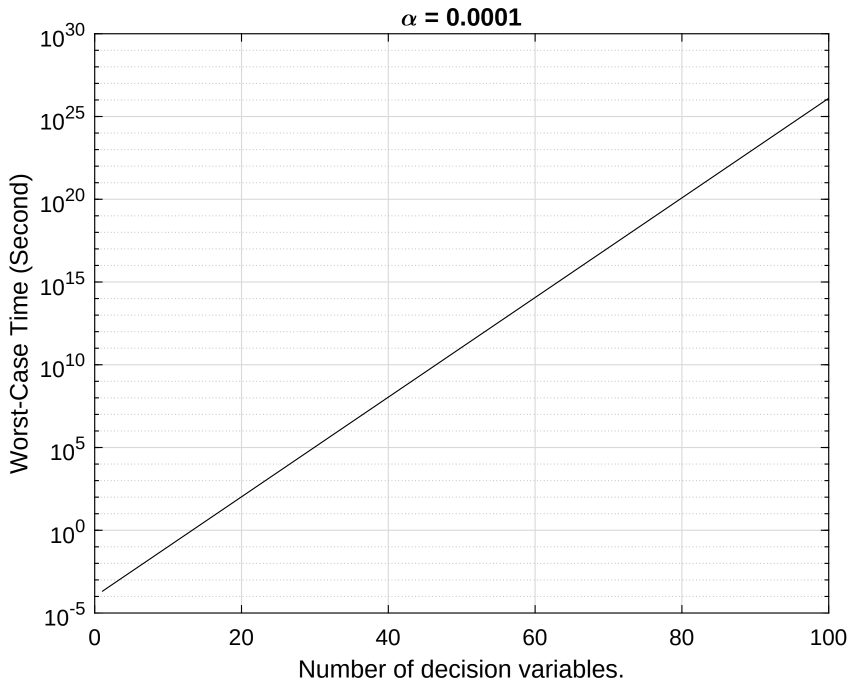


Figure 5: Worst-Case computational time of an optimization problem with binary decision variable. Here, $\alpha = 0.0001$. For $n \geq 50$, the worst-case computational time is thousands of years.

3 On Boltzmann Machine (Visible-Hidden-Layer)

(a) The dynamics of Geoffrey Hinton's Boltzmann machine is in principle the same as a Hopfield network. The input to the i^{th} neuron is the weighted sum of the neuron outputs in the previous time step. However, the output of a neuron is not defined as a deterministic equation as in (2). Instead, the output of a neuron is stochastic.

$$P(s_i(t) = 1) = \frac{1}{1 + \exp(-(u_i(t) - \tau_i)/T)} \text{ and } P(s_i(t) = -1) = \frac{\exp(-(u_i(t) - \tau_i)/T)}{1 + \exp(-(u_i(t) - \tau_i)/T)}, \tag{10}$$

where T is a positive control factor called *temperature*. That is to say, the output of the i^{th} neuron can be -1 (resp. $+1$) even if $u_i(t) > \tau_i$ (resp. $u_i(t) < \tau$).

Figure 6 shows the shapes of (10) for $T = 1, 0.5, 0.1$. From the tendency, one can see that the shape of $P(s_i = 1)$ tends to be a step function if $T \to 0$. That is to say, $P(s_i = 1) \to 1$ if $(u_i - \tau_i) > 0$. $P(s_i = -1) \to 1$ if $(u_i - \tau_i) < 0$. In such case, the response of this *probabilistic neuron* behaves almost the same as the *deterministic neuron* as defined in (1).

(b) For a given $\mathbf{W}$ and $\boldsymbol{\tau}$, the probability mass function of $\mathbf{s}$ follows a *Boltzmann distribution* as follows :

$$P(\mathbf{s}) = \frac{\exp\left(-E(\mathbf{s}|\mathbf{W}, \boldsymbol{\tau})/T\right)}{\sum_{\mathbf{s}'} \exp\left(-E(\mathbf{s}'|\mathbf{W}, \boldsymbol{\tau})/T\right)}, \tag{11}$$

where $E(\mathbf{s}|\mathbf{W}, \boldsymbol{\tau})$ is given by (8). $P(\mathbf{x})$ is a function dependent on $E(\mathbf{s}|\mathbf{W}, \boldsymbol{\tau})$ and has the following property.

$$P(\mathbf{s}) > P(\mathbf{s}') \text{ if } E(\mathbf{s}|\mathbf{W}, \boldsymbol{\tau}) < E(\mathbf{s}'|\mathbf{W}, \boldsymbol{\tau})$$

for all $T > 0$.

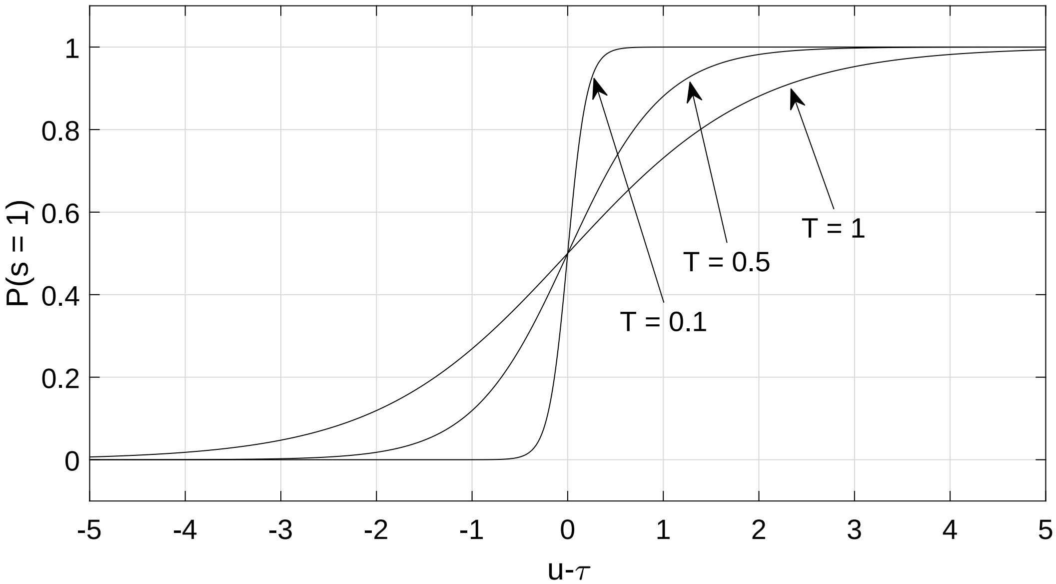


Figure 6: The probability of a neuron its output is one. If the temperature $T \to 0$, $P(s_i = 1) \to 1$ if $(u_i - \tau_i) > 0$. $P(s_i = -1) \to 1$ if $(u_i - \tau_i) < 0$. The neurons behave as the deterministic neurons stated in (1).

Let $Q(\mathbf{s})$ be the probability mass function of the state $\mathbf{s}$ in true population. The goal of Boltzmann learning is to find the $\mathbf{W}$ and $\boldsymbol{\tau}$ such that $P(\mathbf{s}) \to Q(\mathbf{s})$.

(c) There is a major different between Hopfield networks and Boltzmann machines. Hopfield network consists of a single layer of so-called visible neurons, while Boltzmann machines consists of two layers of neurons namely the visible and hidden layers. For both the Hopfield network and Boltzmann machine, memory is determined by the patterns presented to the visible layer. Boltzmann machine memorizes those patterns by using more neurons in the hidden layer. These hidden neurons could lead the Boltzmann machine to memorize more patterns with the same number of visible neurons.

(d) From another viewpoint, a Boltzmann machine attempts to model a neural structure which consists of a layer of sensory neurons and another layer of neurons in the brain.

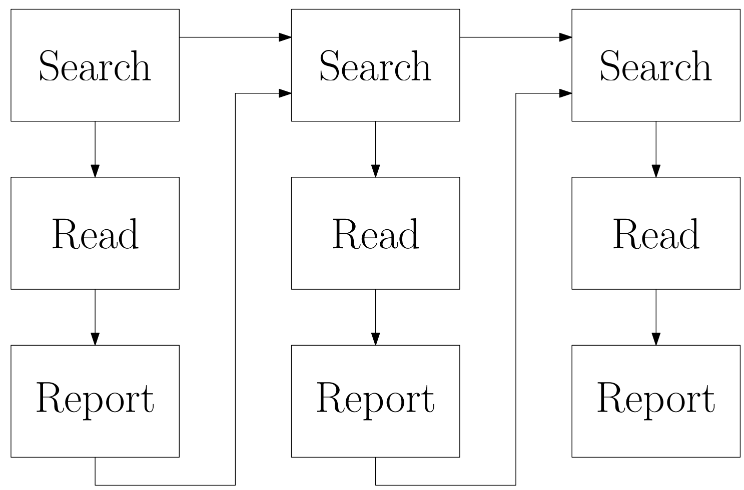
(e) Let $\mathbf{v}$ and $\mathbf{h}$ be the vectors of the visible states and the hidden states. The goal of Boltzmann learning is again to find the $\mathbf{W}$ and $\boldsymbol{\tau}$ such that $P(\mathbf{v}) \to Q(\mathbf{s})$. With hidden neurons, the probability mass function $P(\mathbf{v}|\mathbf{W}, \boldsymbol{\tau})$ is given by

$$P(\mathbf{v}|\mathbf{W}, \boldsymbol{\tau}) = \frac{\sum_{\mathbf{h}'} \exp\left(-E\left(\mathbf{v}, \mathbf{h}'|\mathbf{W}, \boldsymbol{\tau}\right)/T\right)}{\sum_{\mathbf{v}'} \sum_{\mathbf{h}'} \exp\left(-E(\mathbf{v}', \mathbf{h}'|\mathbf{W}, \boldsymbol{\tau})/T\right)} \tag{12}$$

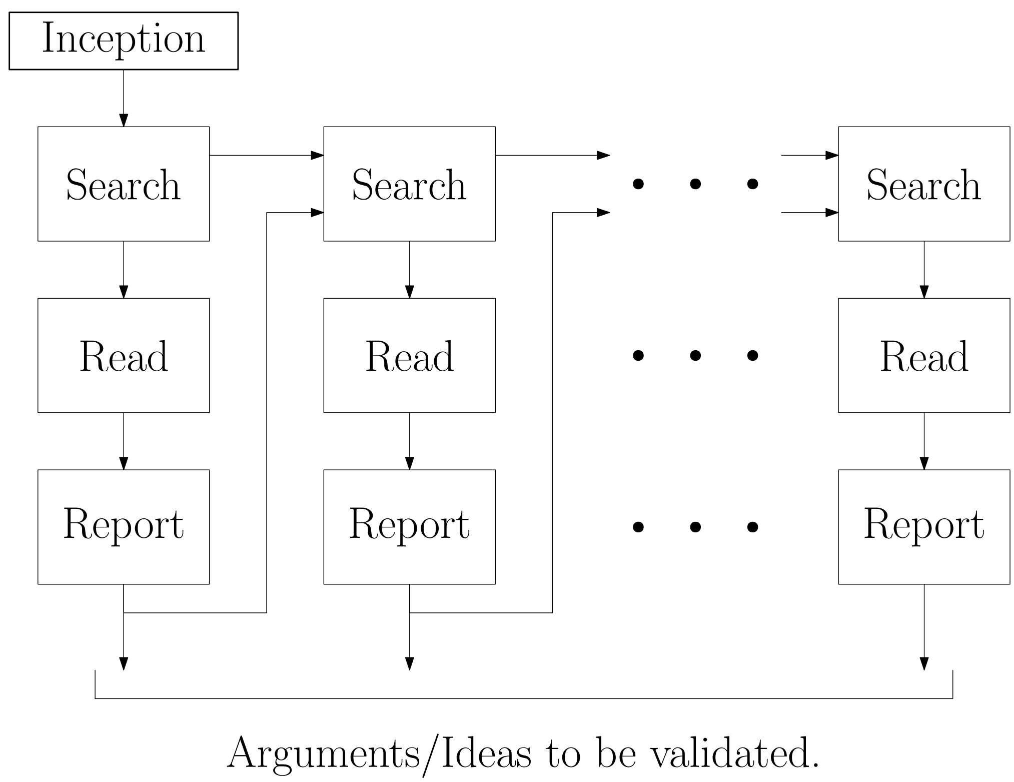
4 On Survey

(a) A note on survey, see Figure 7. Note that the diagram only shows three rounds of search-read-report. In reality, the number of rounds are definitely more than three.

Question: In each round of search-and-read, how many papers to be collected and read? Once you have read a paper, you need to ensure that you understand everything in it. You are able to give a seminar talking about the paper.



(a) General perception.



(b) Infinite rounds.

Figure 7: Survey could be a lengthy process. It is a multiple-round process. AI tools could help at the inception stage. They might help at the search and report steps. The report compiled in each round is basically a progress report.

(b) Every survey could be a life-long process, as shown in Figure 7b, if you really want to survey everything about a topic. There is no method to estimate how much time you can complete a survey. It is all depended on how much you want to know about the topic.

(c) To validate the arugments/ideas bringing out from the surveys are yet life-long projects. Getting partial result for these arguments/ideas is clearly difficult and could be a valuable work. But, it has to say that they might not be contributable. If you cannot ensure the novelty of the problem, your contribution is none. You could simply repeat (equivalently copy) someone else work.