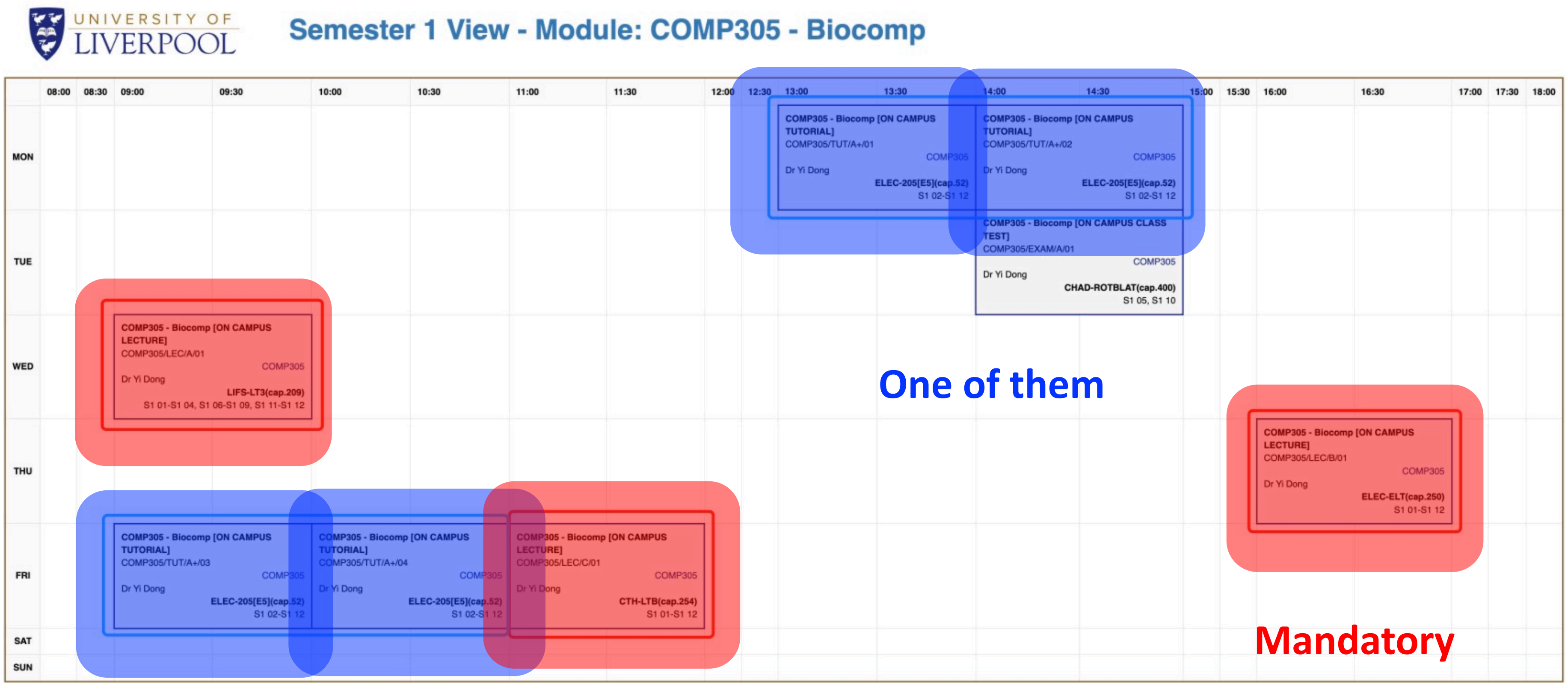
Comp305

Biocomputation

Lecturer: Yi Dong

Comp305 Module Timetable



There will be **26-30** lectures, thee per week. The lecture slides will appear on Canvas. Please use Canvas to access the lecture information. There will be **9** tutorials, one per week.

Lecture/Tutorial Rules

Questions are welcome as soon as they arise, because

1. Questions give feedback to the lecturer;

2. Questions help your understanding;

3. Your questions help your classmates, who might experience difficulties with formulating the same problems/doubts in the form of a question.

Sample and Practice Questions

- One selected past class test will be uploaded to Canvas as sample class test.

- Homework questions are practice questions for class tests and the exam.

- No solutions will be provided based on the department policy.

Academic Integrity

- The Computer Science Student Canvas page now has a guidance document that details information relating to academic integrity, particularly in relation to Computer Science.

- https://canvas.liverpool.ac.uk/courses/80362/pages/academic-integrity-information

MP Neuron

- Motivated by **All-Or-None** property of information processing by a biological neuron;

- Proposed to represent any **propositional logic formula**.

Representation Power of a single MP Neuron

- A single MP neuron can be used to represent some Boolean functions which are linearly separable.

- Linear separability (for Boolean functions): There exists a line (plane) such that all inputs which produce a 1 for the function lie on one side of the line (plane) and all inputs which produce a 0 lie on other side of the line (plane).

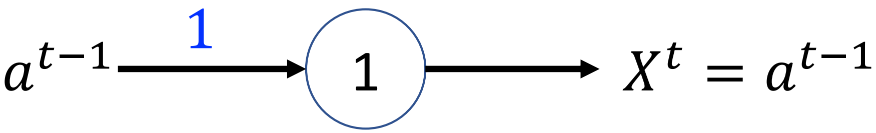
- Completeness: Can each linear separable function be represented by a single MP neuron? **No**!

- A single MP neuron describes a specific linear boundary that is only determined by the threshold in the hyper-cube state space, removing the faces corresponding to inhibitory connections.

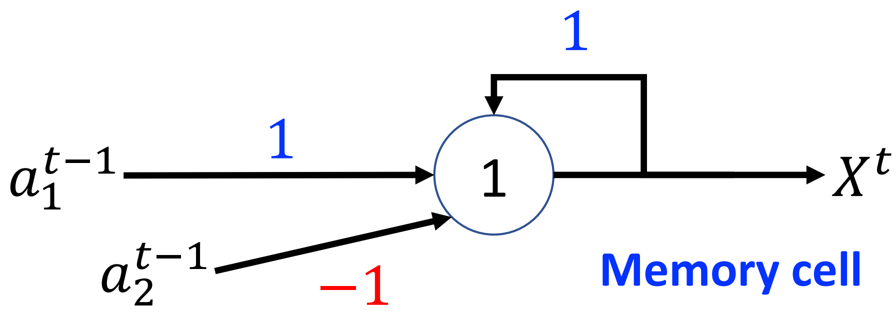
Presentation Power of a MP neural network

- Although the McCulloch-Pitts neuron model was very simplistic, it can ***perform*** the ***basic logic operations*** **AND, OR** and **NOT**

- An MP neural network ***can implement*** any ***multivariate propositional logic function***, with the thresholds and weights being appropriately selected.
- Furthermore, the discrete time, *or* unity delay property of the model makes it even ***possible to build*** ***a sequential digital circuitry***.



Register cell



Memory cell

Drawbacks of MP Neuron Model

- The main "ideological" problems of the McCulloch- Pitts model were that.
    - The network must be **completely specified before its using**
    - There were **no free parameters** to suit different problems.

- ***There are actually no well-known learning algorithms for standard MP neuron models***.

Comp305 Part I.

Artificial Neural Networks

Topic 3.

Hebb’s Rules

“… Cells that fire together, wire together…”

Topic of today's lecture

Hebb's Rules and the historical background.

MP neuron vs. Brain Function

- The **McColloch-Pitts** neuron made a base for a machine (network of units) capable of
  - ***storing information and***
  - ***producing logical and arithmetical operations on it***

- These correspond to the main functions of the brain as
  - **to store knowledge and**
  - **to apply the knowledge stored to solve problems**.

ANN Learning Rules

- The **McCulloch-Pitts** neuron made a base for a machine (network of units) capable of
  - ***storing information and***
  - ***producing logical and arithmetical operations on it***

- The next step
  - must be to realise another important function of the brain, which is

***to acquire new knowledge through experience*, i.e. *learning*.**

ANN Learning Rules

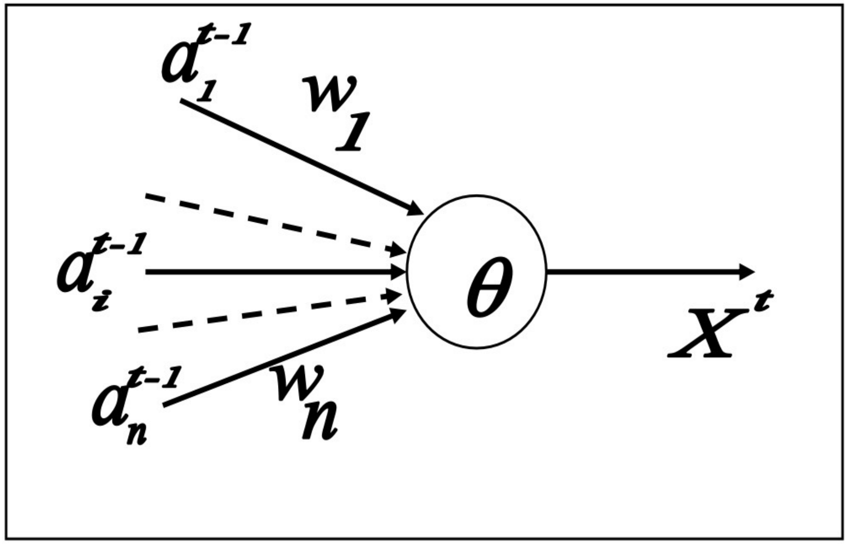
- Learning means

  to change in response to experience

- In an MP neural network, no free parameters.

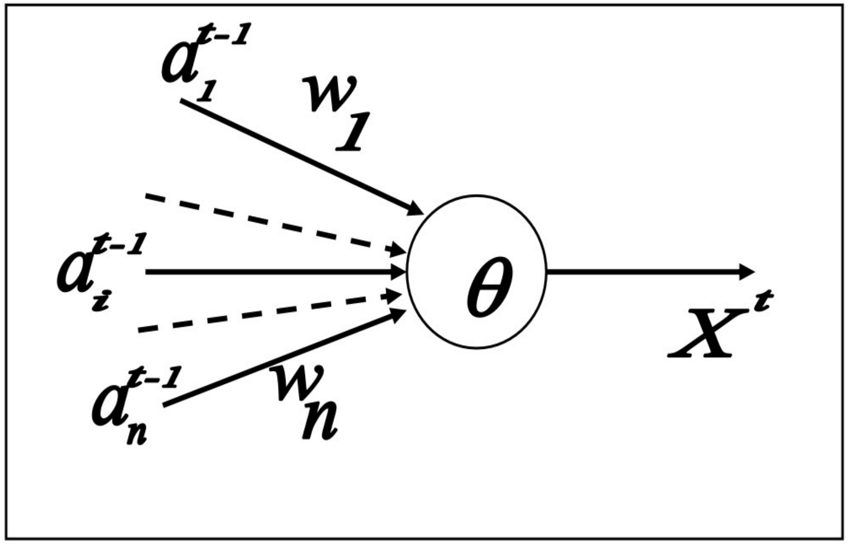
- We need a new model, of which the parameters are easily changeable (to be learnt).

ANN Learning Rules



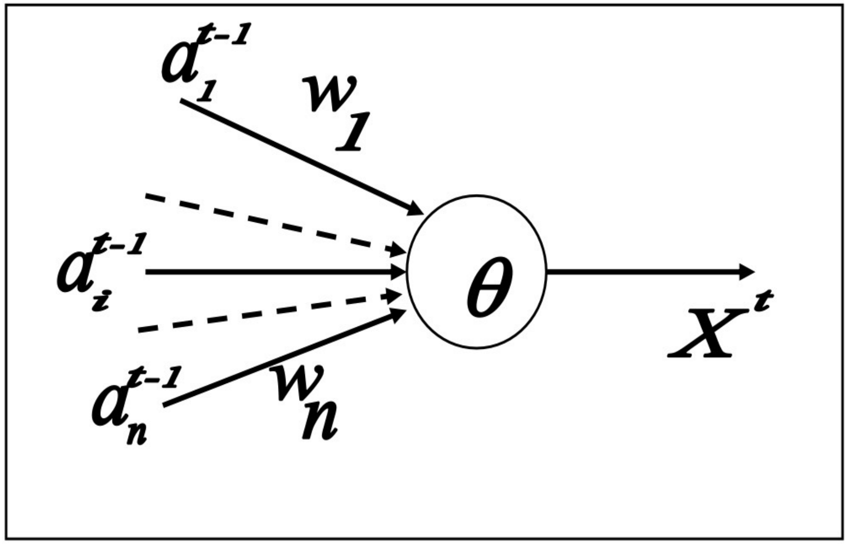
Q: What parameters do you think can be free parameters?

ANN Learning Rules



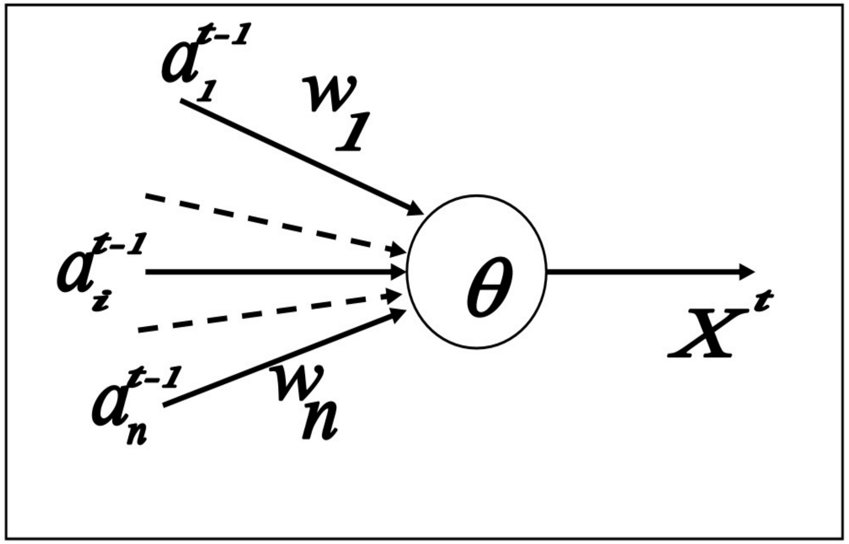
- The ideal **free parameters to adjust**, and so to resolve learning, **are the weights of connections w**.

Beyond Standard MP Neuron



- From now, we do **NOT** have the restriction on the weight. That is, the weight can be any real value.
- We do **NOT** check the inhibitory input.

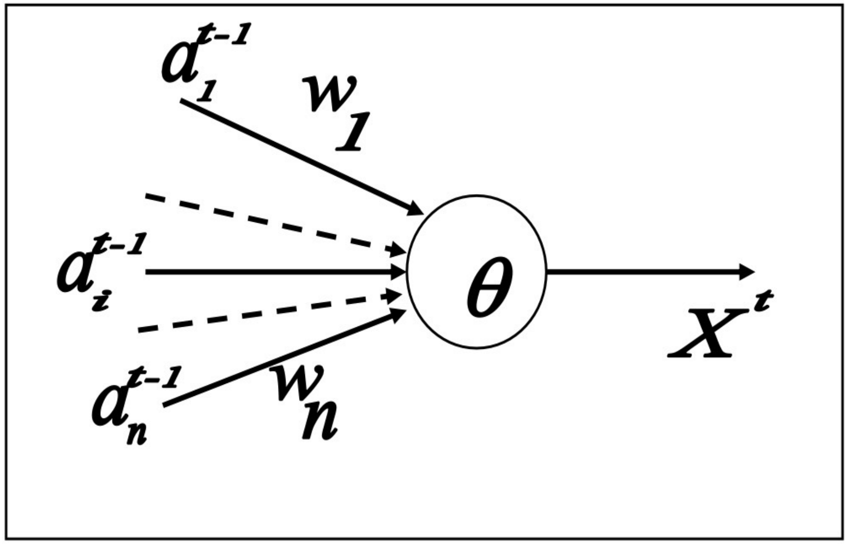
Beyond Standard MP Neuron



- From now, we do **NOT** have the restriction on the weight. That is, the weight can be any real value.
- We do **NOT** check the inhibitory input.

Why? The motivation is **different**!
We do not consider the propositional logic here. Instead, we think about **learning**!

Beyond Standard MP Neuron

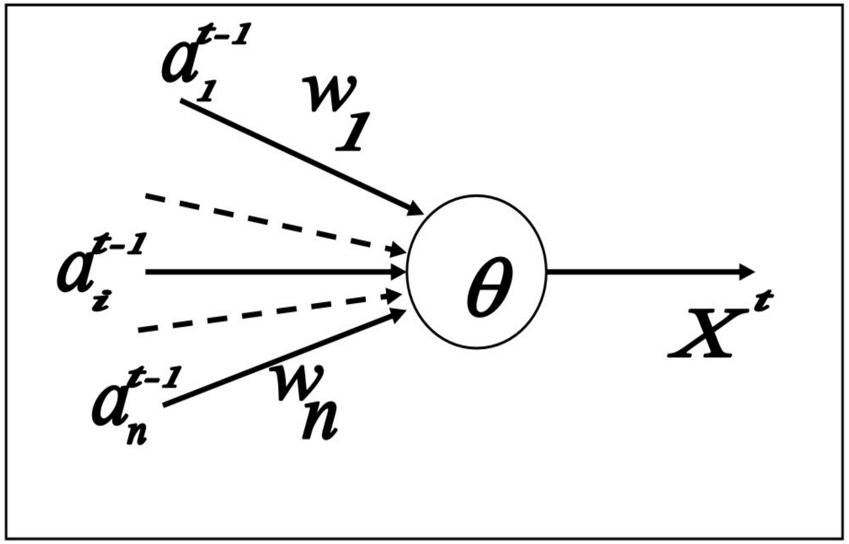


- From now, we do **NOT** have the restriction on the weight. That is, the weight can be any real value.
- We do **NOT** check the inhibitory input.

Motivations are different. Thus, the models are different.

Why? The motivation is **different**!
We do not consider the propositional logic here. Instead, we think about **learning**!

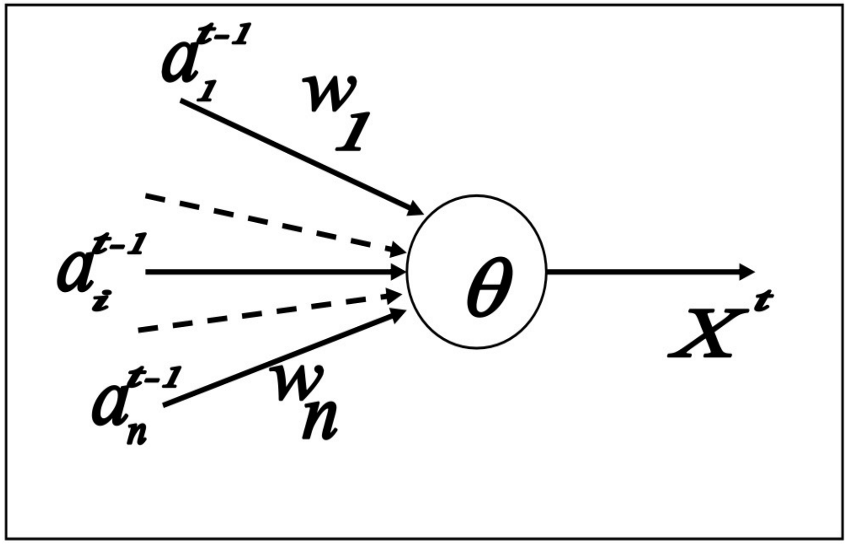
ANN Learning Rules



Definition:

ANN learning rule is
the rule how to adjust the weights of connections to get desirable output.

ANN Learning Rules



Much work in Artificial Neural Networks focuses on the learning rules that define

how to <u>change the weights of connections</u> between neurons to better adapt a network to serve some overall function.

ANN Learning Rules

- As for the first time the problem was formulated in **1940s**, when experimental neuroscience was limited, the classic definitions of these learning rules came not from biology, but

from psychological studies of
Donald Hebb and **Frank Rosenblatt**.

Hebb’s Rule (1949)

Hebb proposed that

a particular type of

use-dependent modification

of the connection strength of synapses

might underlie

learning in the nervous system.

Hebb’s Rule (1949)

Hebb introduced a neurophysiological ***postulate*** :

“... When an axon of cell A

- **is near enough to excite a cell B and**
- **repeatedly and persistently takes part in firing it,**

some growth process or metabolic change

takes place in one or both cells,

such that A’s efficiency as one of the cells firing B, is increased. ”

Hebb's Rule (1949)

In another word:

"… Cells that fire together, wire together…"

Hebb’s Rule (1949)

An experimental confirmation to Hebb’s hypothesis came much later: in **1970**s, physiologists **Tim Bliss** and **Terje Lomo** discovered

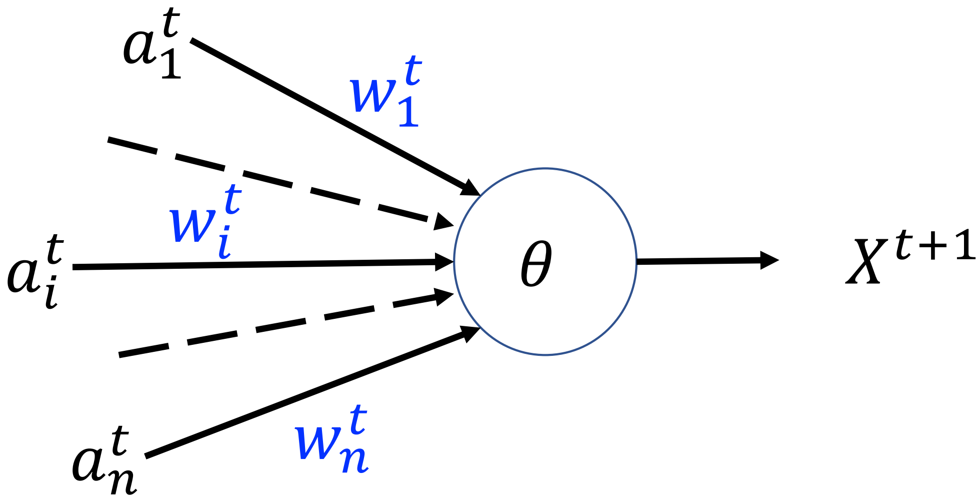
the long-term potentiation, *a sustained state of increased synaptic efficacy consequent to intense synaptic activity*.

Hebb's Rule (1949)

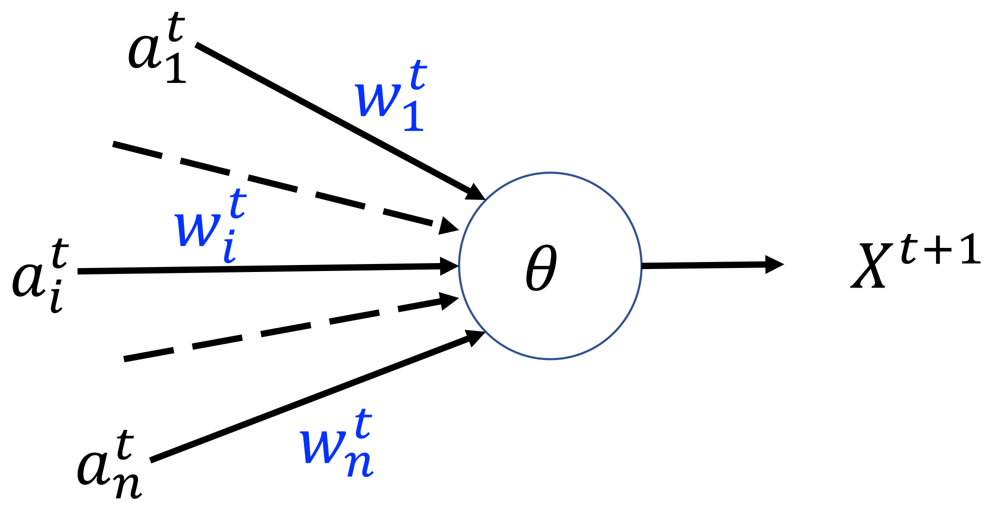
- The conditions that Hebb predicted would lead to changes in synaptic strength

have now been found to cause the **long-term potentiation** in some neurons of ***hippocampus*** *and other brain areas.*

Hebb's Rule (1949)



Hebb's Rule (1949)



Consider the above neuron, the weight w_i^t is what we want to learn.

Again, here we allow w_i^t could be other than -1 or 1, and we do not check inhibitory inputs. It is NOT an MP neuron.

Hebb's Rule (1949)

- The simplest formulation of Hebb's rule is to increase weight of connection at every next instant in the way:

$$w_i^{t+1} = w_i^t + \Delta w_i^t,$$

Where

$$\Delta w_i^t = C a_i^t X^{t+1}$$

Hebb's Rule (1949)

$$w_i^{t+1} = w_i^t + \Delta w_i^t,$$

Where

$$\Delta w_i^t = C a_i^t X^{t+1}$$

w_i^t is the weight of connection at instant t,

w_i^{t+1} is the weight of connection at the next instant $t+1$,

Δw_i^t is the increment by which the weight of connection is enlarged,

C is positive coefficient which determines learning rate.

a_i^t is input value from the presynaptic neuron at instant t,

X^{t+1} is output of the postsynaptic neuron at the instant $t+1$.

Hebb's Rule (1949)

$$w_i^{t+1} = w_i^t + \Delta w_i^t,$$

Where

$$\Delta w_i^t = C a_i^t X^{t+1}$$

Thus, the weight of connection changes at the next instant only if both preceding input via this connection and the resulting output simultaneously are not equal to 0.

Input is not equal to 0. $\Longrightarrow$ Excitatory input.

Output is not equal to 0. $\Longrightarrow$ Neuron is fired.

Hebb's Rule (1949)

$$w_i^{t+1} = w_i^t + \Delta w_i^t,$$

Where

$$\Delta w_i^t = C a_i^t X^{t+1}$$

- The second equation emphasizes the correlation nature of a Hebbian synapse.
- Sometimes the Hebb's rule is referred to as activity product rule.
- Hebb's original learning rule referred exclusively to excitatory synapses

Algorithm of Hebb's Rule for a Single Neuron (NOT MP)

1. Set the neuron threshold value θ and the learning rate C.
2. Set *random initial values* for the weights of connections w_i^t.
3. Give instant input values a_i^t by the input units.
4. Compute the instant state of the neuron $S^t = \sum_i w_i^t a_i^t$
5. Compute the instant output of the neuron X^{t+1}
$$X^{t+1} = g(S^t) = H(S^t - \theta) = \begin{cases} 1, & S^t \geq \theta; \\ 0, & S^t < \theta. \end{cases}$$
6. Compute the instant corrections to the weights of connections $\Delta w_i^t = C a_i^t X^{t+1}$
7. Update the weights of connections $w_i^{t+1} = w_i^t + \Delta w_i^t$
8. **Go to the step 3**.