# Comp305

# Biocomputation

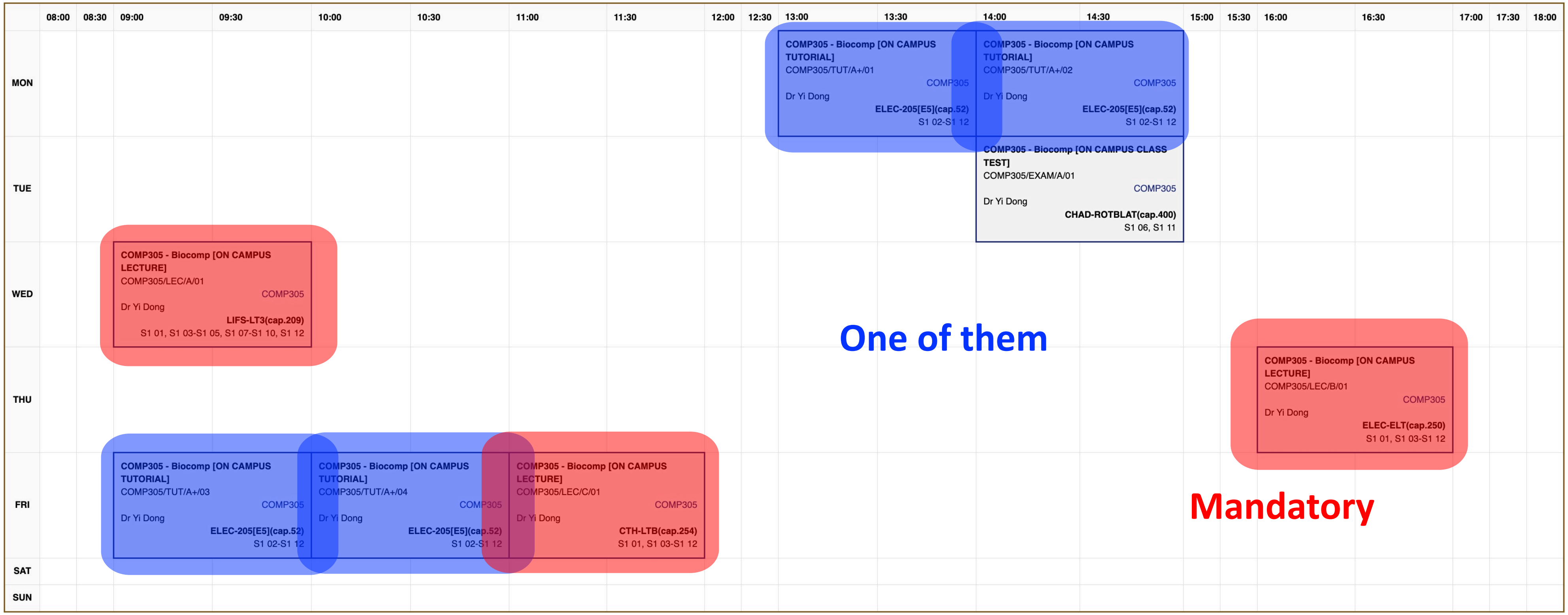
Lecturer: Yi Dong

# Comp305 Part I.

# Artificial Neural Networks

# Comp305 Module Timetable

**Semester 1 View - My 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;

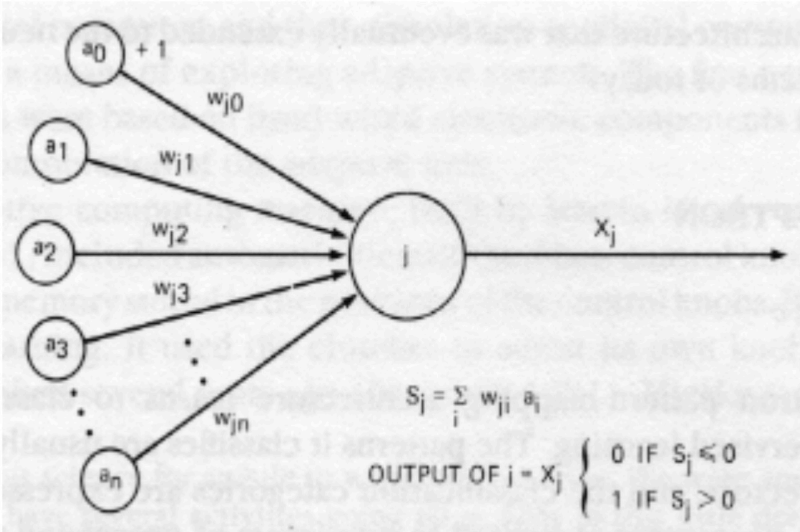
2. Questions help your understanding;

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

# Recap: Neuron Signal Processing

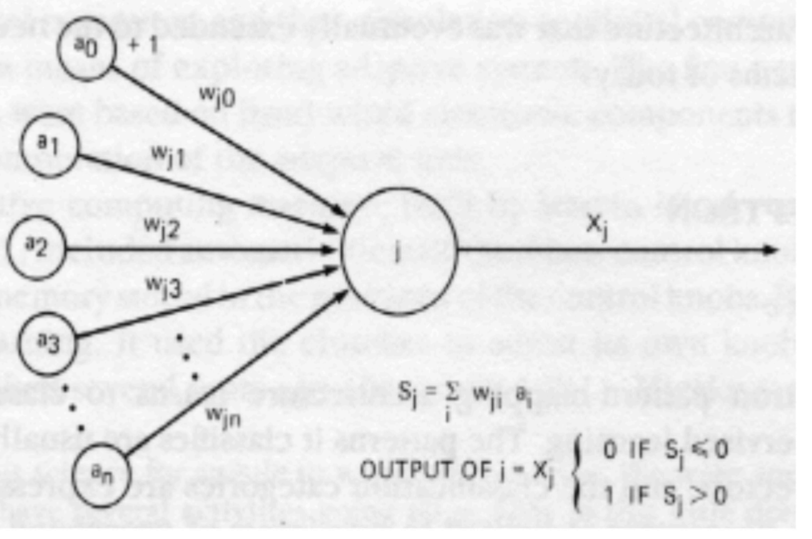
| | | |
|---|---|---|
| **Neuron** | Input | Dendrites: From huge number of neurons, sometimes very distant ones |
| | Excitation | From a repetition of impulses in time at the same synapse (***temporal summation***) or from the simultaneous arrival of impulses at a sufficient number of adjacent synapses (***spatial summation***) to make the "density" of signal high enough at some region of the neuron to overcome ***the excitation threshold***. There are ***refractory periods.*** |
| | Output | Axon: To huge number of neurons, sometimes very distance ones |
| **Neuron-to-Neuron** | Propagation | **spikes, but not subthreshold potentials, propagate regeneratively down the axons.** |

# Recap: Abstract Model of a Neuron




- Shown is an abstract neuron ***j*** with ***n+1*** inputs.
- Each input ***i*** transmits a real value $a_i$ .
- Each connection is assigned with the weight $w_{ji}$.

# Recap: Abstract Model of a Neuron

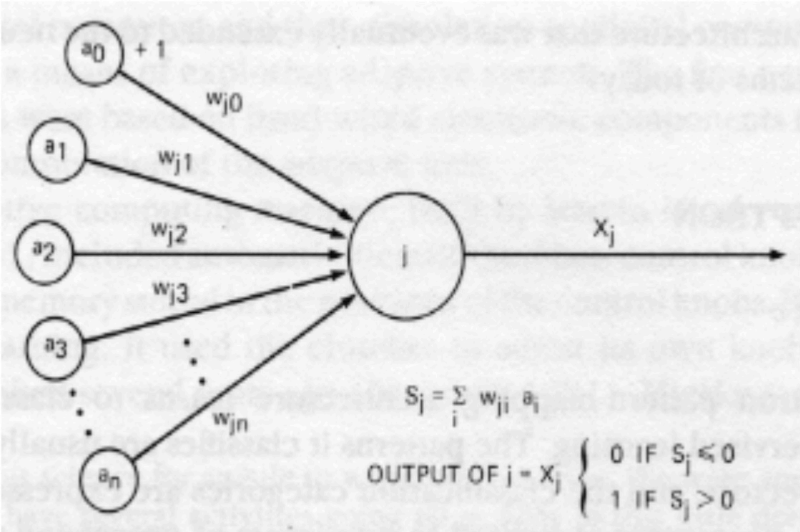



- The ***total input S***, *i.e.* the sum of the products of the inputs with the corresponding weights, ***is compared with the threshold*** (equal to 0 in this case), and the ***outcome $X_j$ is produced consequently.***

# Recap: Abstract Model of a Neuron

| | | | |
|---|---|---|---|
| **Neuron** | Input | Dendrites: From huge number of neurons, sometimes very distant ones | Multiple Inputs |
| | Excitation | From a repetition of impulses in time at the same synapse (***temporal summation***) or from the simultaneous arrival of impulses at a sufficient number of adjacent synapses (***spatial summation***) to make the "density" of signal high enough at some region of the neuron to overcome ***the excitation threshold***. There are ***refractory periods.*** | The abstract neuron is excited (output is equal to 1) when **weighted sum** is above the threshold 0. |
| | Output | Axon: To huge number of neurons, sometimes very distance ones | Single Output. |
| **Neuron-to-Neuron** | Propagation | **spikes, but not subthreshold potentials, propagate regeneratively down the axons.** | The output is binary. |

# Abstract Model of a Neuron




The abstract model can indeed describe the behaviour of a biological neuron. However, it is too general and can be hardly used to solve any practical problems…

Topic 2.

# The McCulloch-Pitts Neuron (1943)

# Topic of Today's Lecture

What is a McCulloch-Pitts Neuron?

# The McCulloch-Pitts Neuron (1943)

McCulloch and Pitts demonstrated that

***"…because of the all-or-none character of nervous activity, neural events and the relations among them can be treated by means of the propositional logic".***

# The McCulloch-Pitts Neuron (1943)

The authors modelled the neuron as

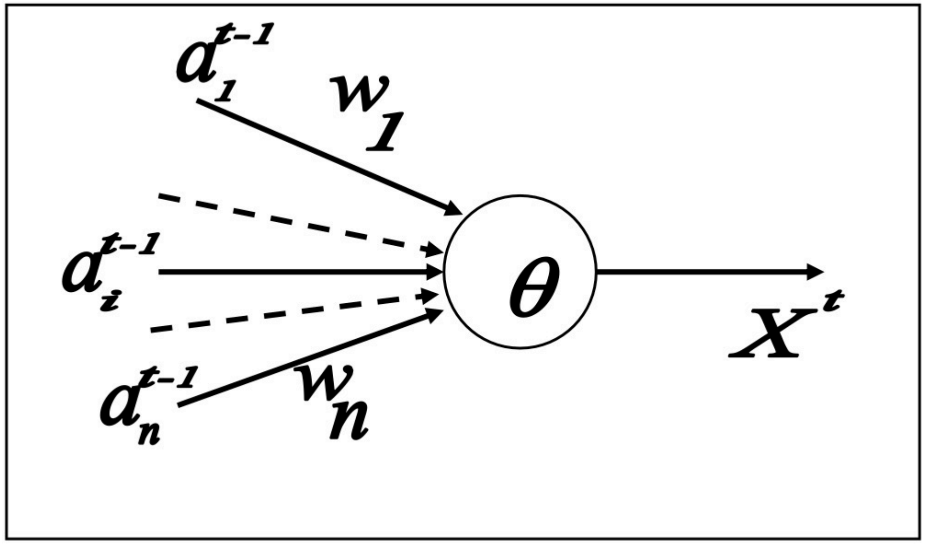
- a ***binary, discrete-time*** input
- with ***excitatory and inhibitory connections and an excitation threshold***.

**The network of such elements was the first model to tie the study of neural networks to the idea of computation in its modern sense.**

Other models (structures): Fully-connected neural networks, Convolutional neural networks, Residual neural network…

# The McCulloch-Pitts Neuron (1943)

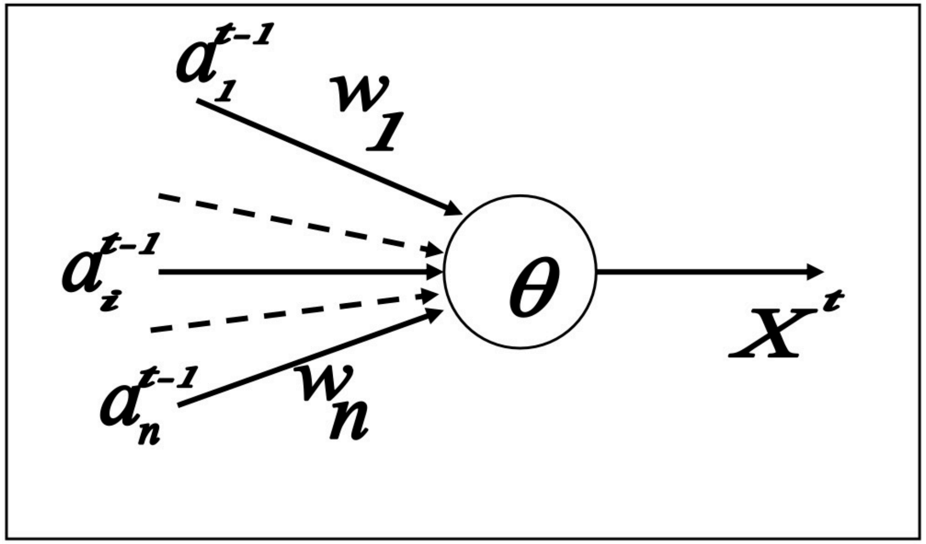
**Discrete Time**




The basic idea was to **divide time into units**, i.e., **steps**, and in each time period at most one spike can be initiated in the axon of a given neuron. **Why?**

# The McCulloch-Pitts Neuron (1943)

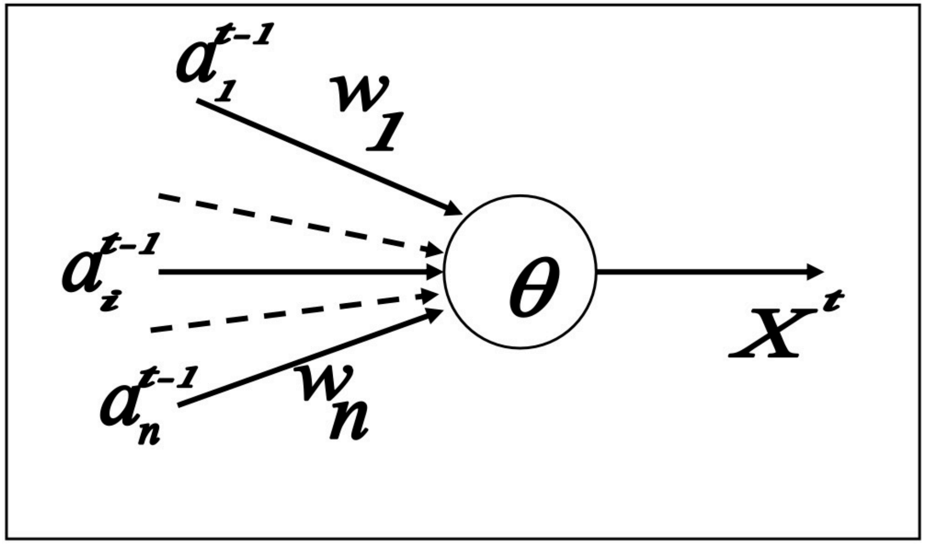
**Discrete Time**




- Discretization**:** Comparable to a refractory period (assumed to be the same for each neuron). **No Zeno executions**!
- Fixed time stepsize: The impulse travels with a nearly **uniform velocity** in a biological neural system.

# The McCulloch-Pitts Neuron (1943)

**Discrete Time**



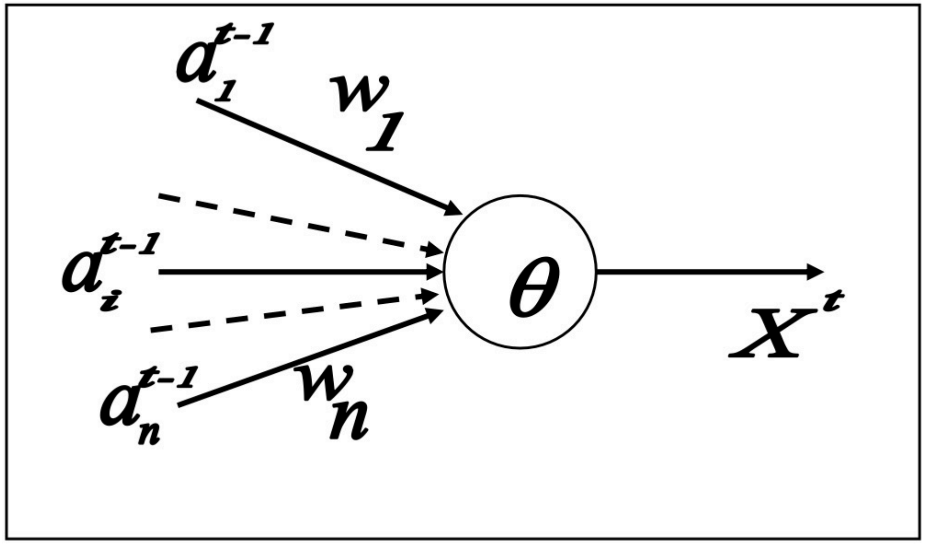

- Thus, the McCulloch-Pitts neuron operates on a discrete time scale,

$$t = 0,1,2,3,\ldots$$

# The McCulloch-Pitts Neuron (1943)

**Binary**

input



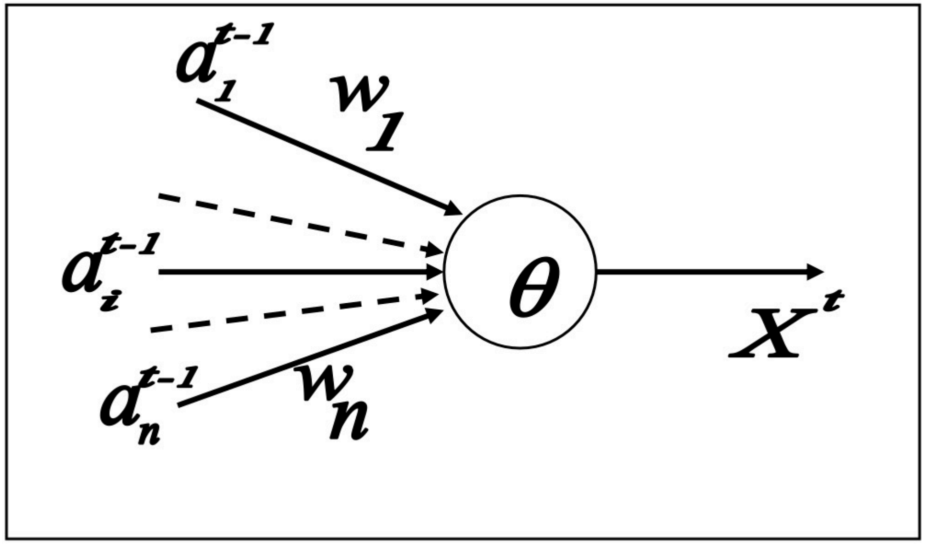

- The input values $a_i^t$ from the $i$-th presynaptic neuron at any instant $t$ may be

equal either to 0 or 1 only. **Why?**

# The McCulloch-Pitts Neuron (1943)

**Binary**

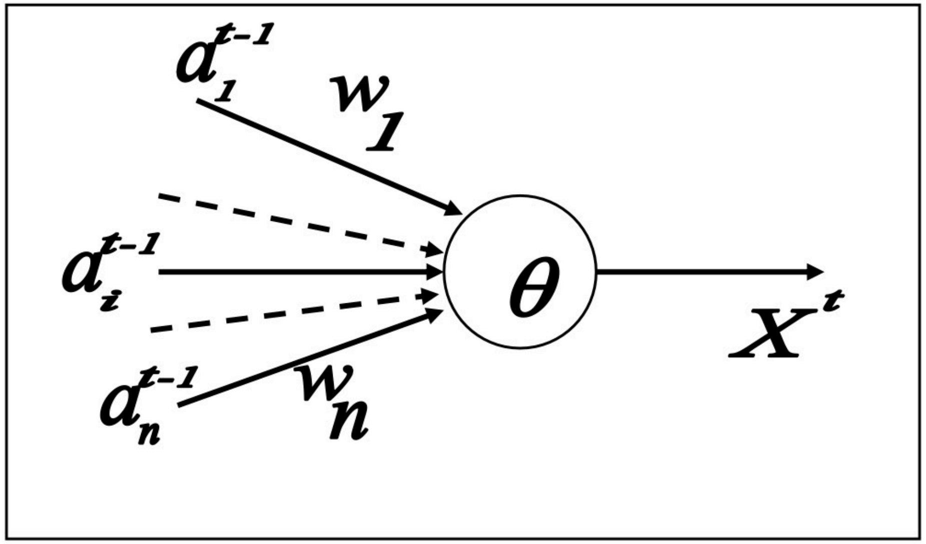
input




- Comparable to the fact that **only spikes are propagated** in biological neural networks;
- The types of the input and the output of a MP neuron are thus **unified**.

# The McCulloch-Pitts Neuron (1943)

***excitatory* and *inhibitory* connections**

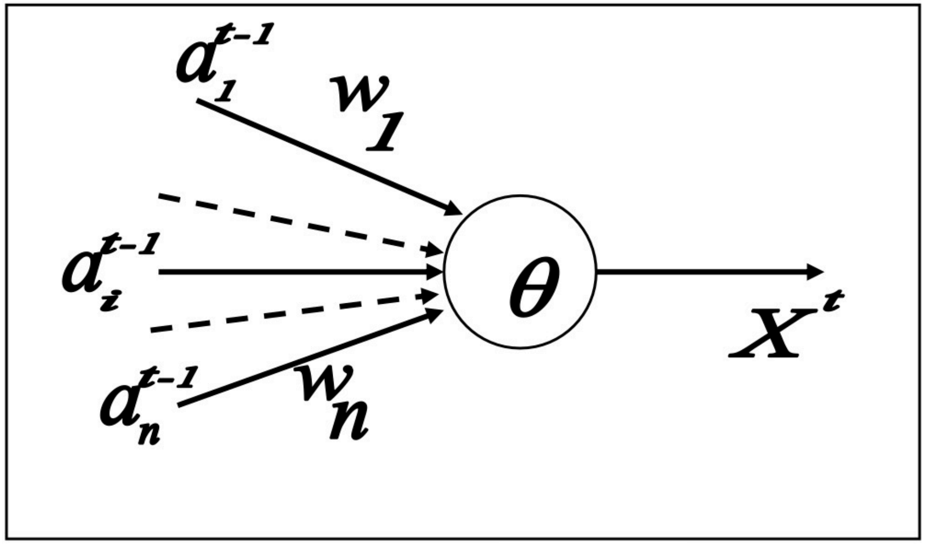


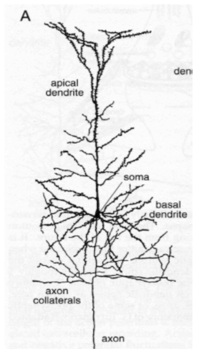

- The weight of connection $w_i$ are
  - +1 for excitatory type connection and
  - -1 for inhibitory type connection.

**Why?**

# The McCulloch-Pitts Neuron (1943)

***excitatory** and **inhibitory***
***connections***



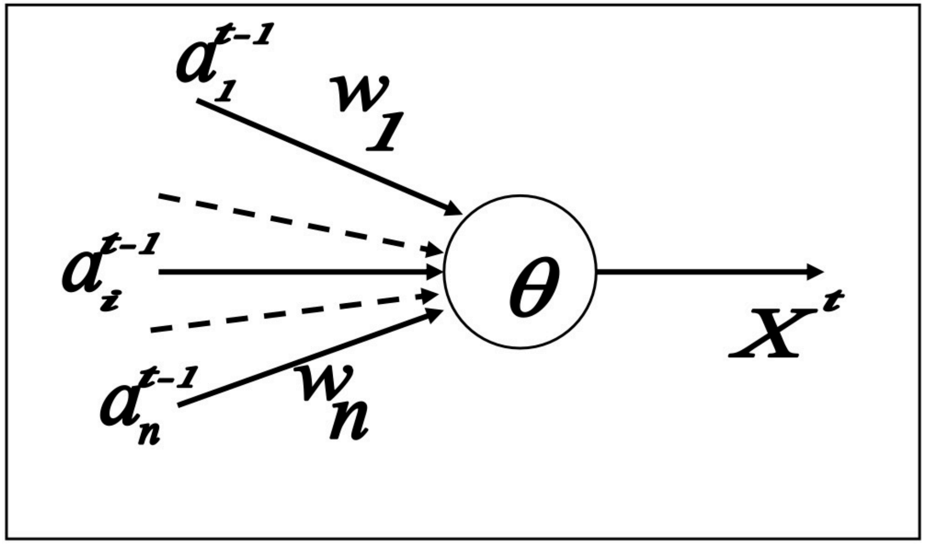




**Cerebral pyramidal cell**

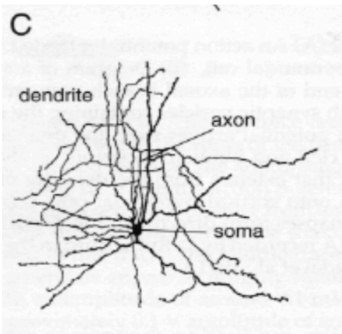
- The weight of connection $w_i$ are
  - +1 for excitatory type connection and
  - -1 for inhibitory type connection.

# The McCulloch-Pitts Neuron (1943)

***excitatory* and *inhibitory***
***connections***



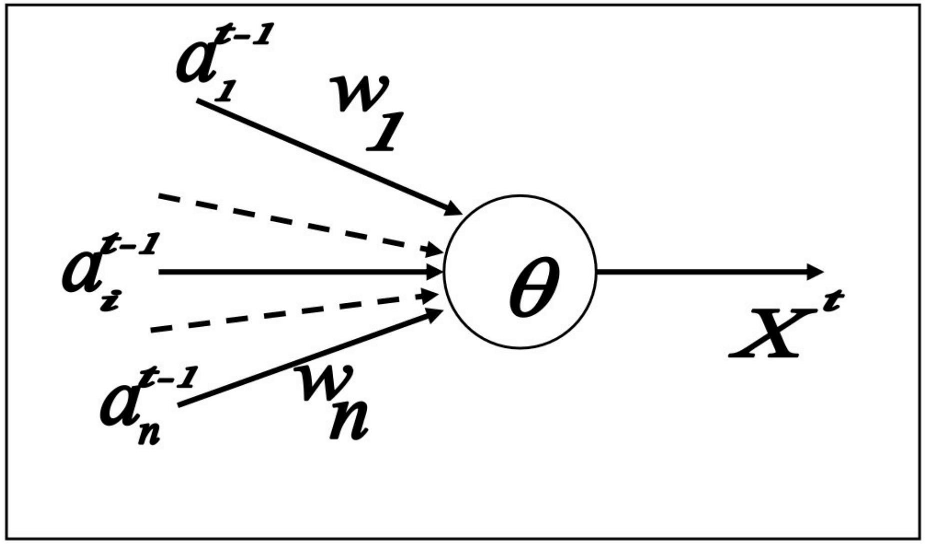
- The weight of connection $w_i$ are
  - +1 for excitatory type connection and
  - -1 for inhibitory type connection.




**Stellate cell**

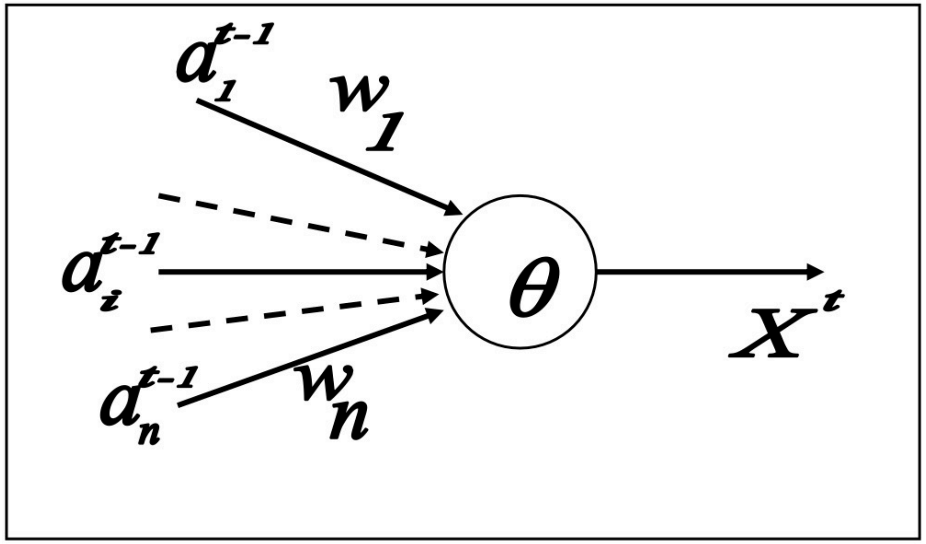
# The McCulloch-Pitts Neuron (1943)

**Threshold**




- There is an excitation threshold $\theta$ associated with the neuron.
- Similar to the basic abstract neuron, $\theta$ is comparable to the potential threshold in a biological neuron.

# MP Neuron: Computation



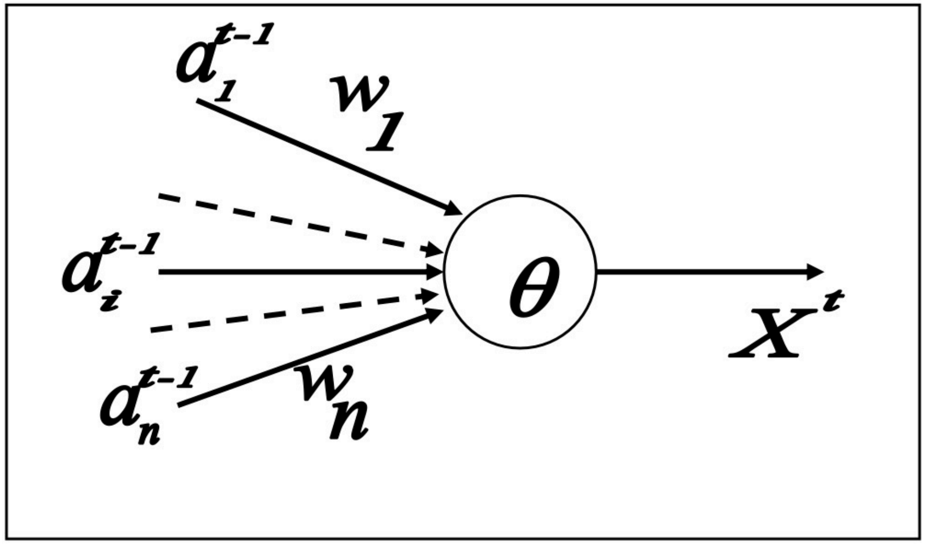

Output $X^t$ of the neuron at the following instant $t$ is defined according to the rule:

$X^t = 1$ if and only if $S^{t-1} = \sum_i w_i a_i^{t-1} \geq \theta$, and $w_i > 0, \forall a_i^{t-1} > 0$.

This is what we are familiar with.

**Why?**

# MP Neuron: Computation



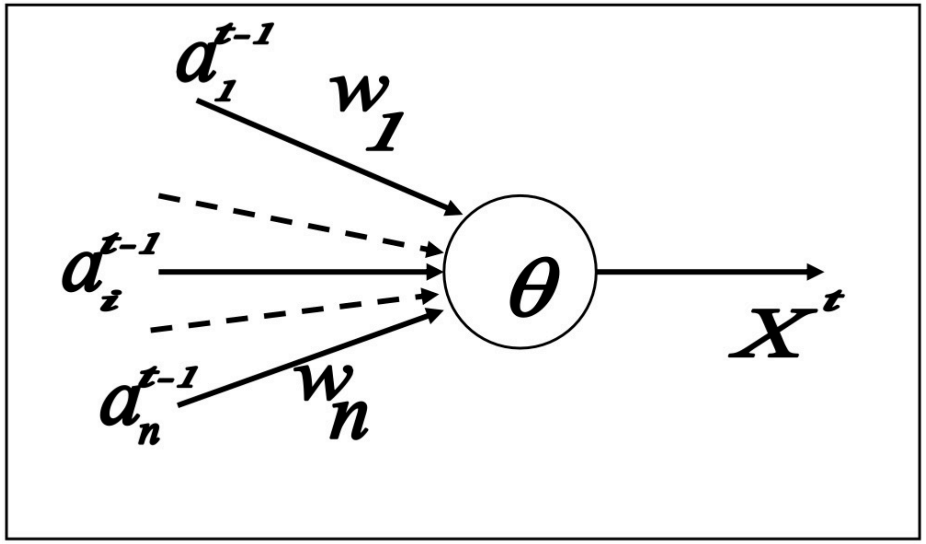

$X^t = 1$ if and only if

$$S^{t-1} = \sum_i w_i a_i^{t-1} \geq \theta,$$

$$w_i > 0, \forall a_i^{t-1} > 0.$$

- The statement $w_i > 0, \forall a_i^{t-1} > 0$ means that activity of a single inhibitory input, i.e., the input via a connection with negative weight -1, prevents excitation of the neuron at that instant.

# MP Neuron: Computation




$X^t = 1$ if and only if

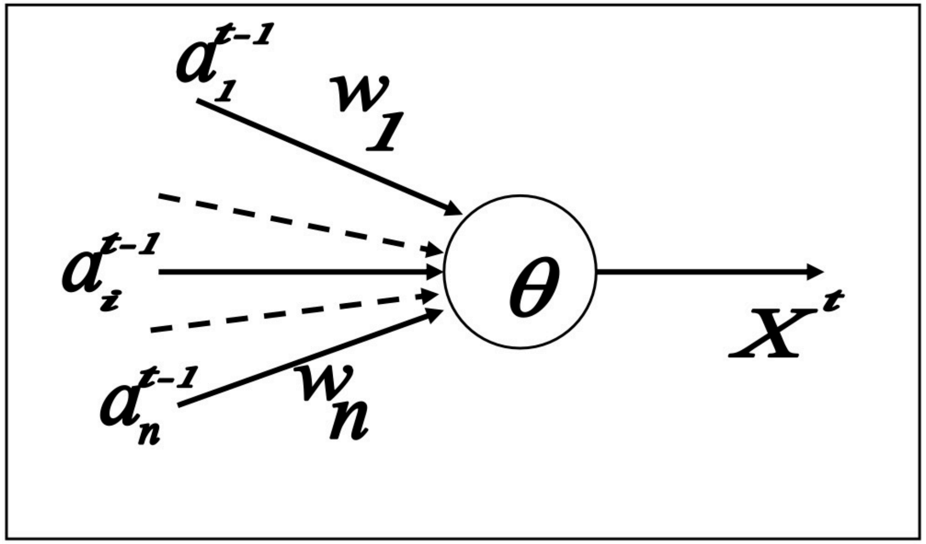
$$S^{t-1} = \sum_i w_i a_i^{t-1} \geq \theta,$$

$$w_i > 0, \forall a_i^{t-1} > 0.$$

- In the MP neuron, we call the instant total input

  $S^{t-1}$: instant state of the neuron

# MP Neuron: Computation




$X^t = 1$ if and only if

$$S^{t-1} = \sum_i w_i a_i^{t-1} \geq \theta,$$

$$w_i > 0, \forall a_i^{t-1} > 0.$$

- The state $S^{t-1}$ of the MP neuron does not depend on the previous state of the neuron itself, but is simply computed by

$$S^{t-1} = \sum_i w_i a_i^{t-1} = f(t-1)$$

# Activation Function

- The neuron output $X^t$ is a function of its state $S^{t-1}$, therefore the output can be also written as a discrete-time function:

$$X^t = X(t) = g(S^{t-1}) = g\big(f(t-1)\big)$$

where

$g$ is the threshold activation function

$$g(S^{t-1}) = H(S^{t-1} - \theta) = \begin{cases} 1, & S^{t-1} \geq \theta; \\ 0, & S^{t-1} < \theta. \end{cases}$$

Other activations (in other neural networks): Rectified Linear Unit (ReLU), sigmoid…

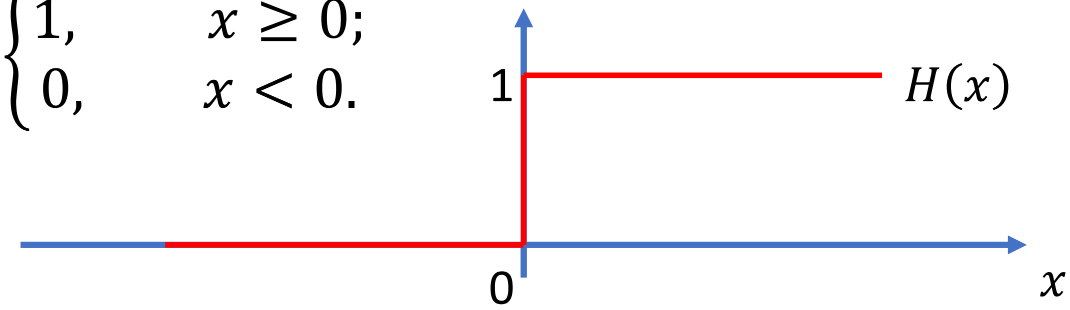
# Activation Function

- $g$ is the threshold activation function

$$g(S^{t-1}) = H(S^{t-1} - \theta) = \begin{cases} 1, & S^{t-1} \geq \theta; \\ 0, & S^{t-1} < \theta. \end{cases}$$

- Here $H$ is the Hesviside (unit step) function:

$$H(x) = \begin{cases} 1, & x \geq 0; \\ 0, & x < 0. \end{cases}$$




- We would like to use the concept of activation function to describe the excitation mechanism in the future.

# Abstract Model of a Neuron

| | Biological neuron | Basic abstract model |
|---|---|---|
| Input | Dendrites: From huge number of neurons, sometimes very distant ones | Multiple Inputs |
| Excitation | From a repetition of impulses in time at the same synapse (***temporal summation***) or from the simultaneous arrival of impulses at a sufficient number of adjacent synapses (***spatial summation***) to make the “density” of signal high enough at some region of the neuron to overcome ***the excitation threshold***. | The abstract neuron is excited (output is equal to 1) when weighted sum is above the threshold 0. |
| Output | Axon: To huge number of neurons, sometimes very distance ones | Single Output. |

# Comparison with Different Models

| | Biological neuron | Basic abstract model | **McCulloch-Pitts Neuron** |
|---|---|---|---|
| Input | Dendrites: From huge number of neurons, sometimes very distant ones | Multiple Inputs | Multiple ***binary***, ***discrete-time*** inputs |
| Excitation | From a repetition of impulses in time at the same synapse (***temporal summation***) or from the simultaneous arrival of impulses at a sufficient number of adjacent synapses (***spatial summation***) to make the "density" of signal high enough at some region of the neuron to overcome ***the excitation threshold***. | The abstract neuron is excited (output is equal to 1) when weighted sum is above the threshold 0. | The output is obtained by computing a threshold **activation function**, if there is no inhibitory inputs.<br><br>Single Output. |
| Output | Axon: To huge number of neurons, sometimes very distance ones | Single Output. | |

# But, …

What can we do with a MP neuron?

# MP-Neuron as a Binary Unit

- ***Simple logical functions*** can be implemented directly with a single McCulloch-Pitts unit.

- The output value **1** can be associated with the logical value ***true*** and **0** with the logical value ***false***.

- In the next lecture, I will demonstrate how weights and thresholds can be set to yield neurons which realise the logical functions **AND**, **OR** and **NOT**.

# MP-Neuron as a Binary Unit

- ***Simple logical functions*** can be implemented directly with **a single McCulloch-Pitts unit**. **Q1: How simple?**

  **Q2: How about multiple neurons, i.e., a MP neural network?**

- The output value **1** can be associated with the logical value ***true*** and **0** with the logical value ***false***.

- In the next lecture, I will demonstrate how weights and thresholds can be set to yield neurons which realise the logical functions **AND**, **OR** and **NOT**.