Understanding Deep Neural Networks

Chapter Two

Network Structure

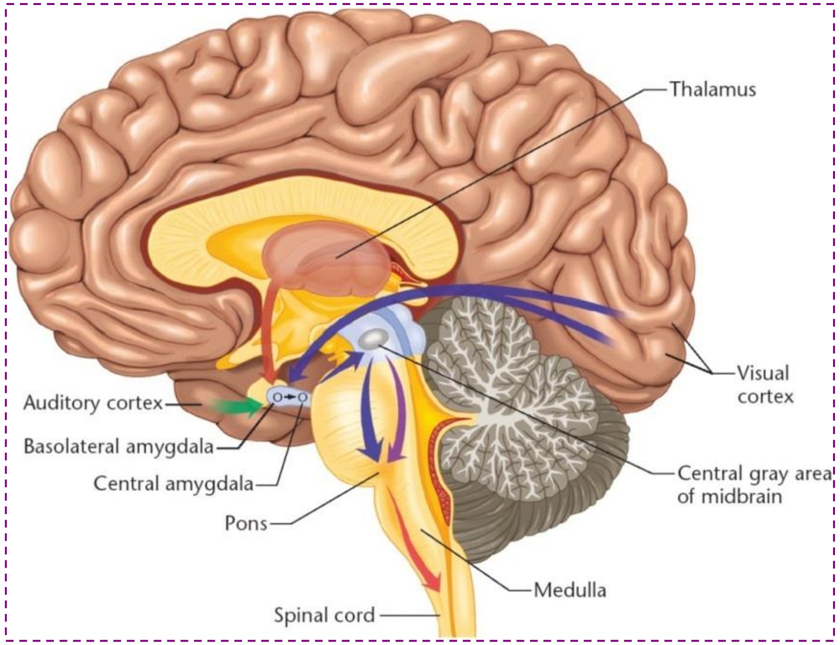
Zhang Yi, *IEEE Fellow*
Autumn 2020

Outline

- Brief Review of Brain Structure
- Computational Model of Neurons
- Computational Model of Neural Networks
- Various Models of Neural Networks
- Discussions
- Assignments

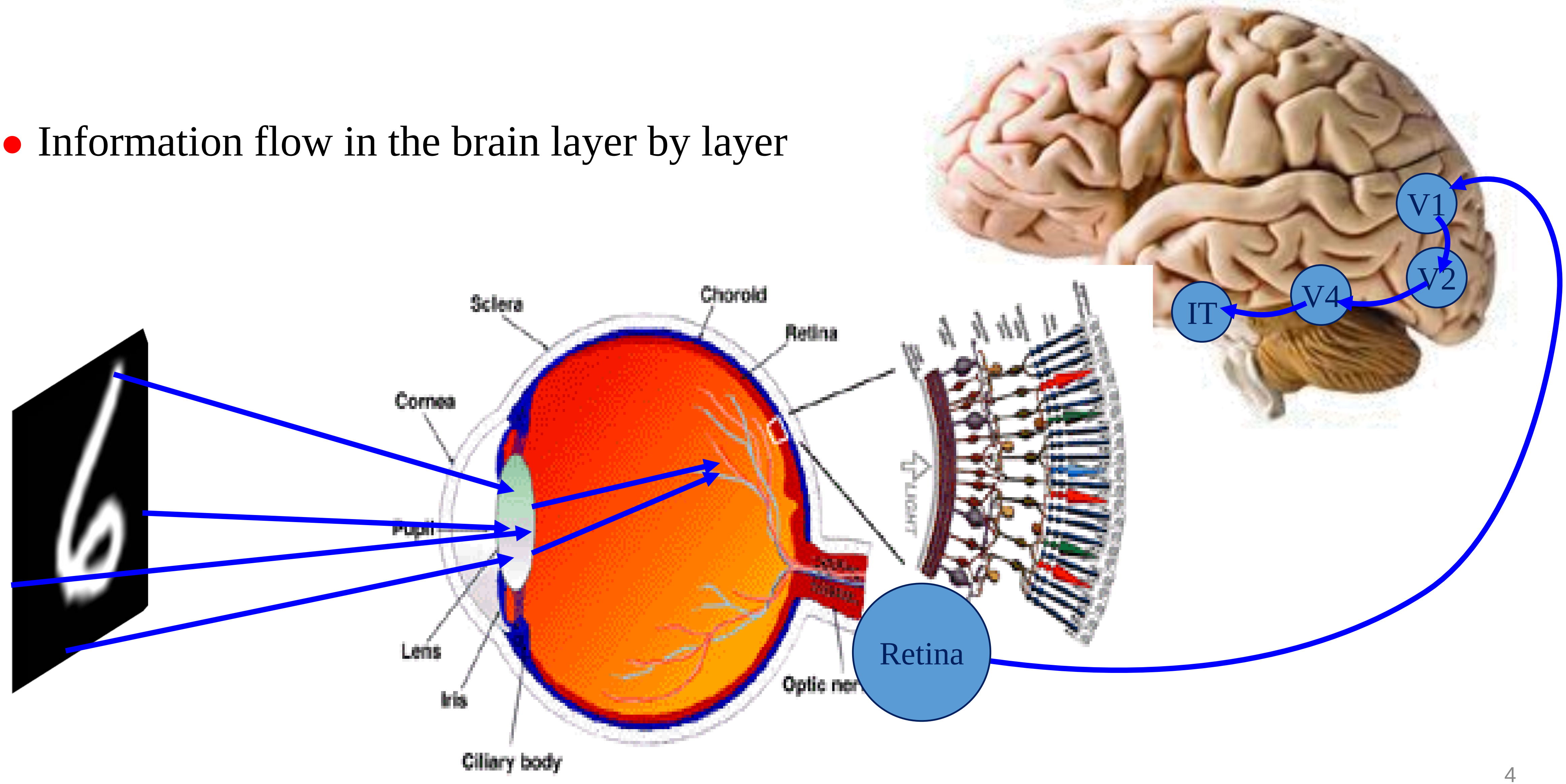
Brief Review of Brain Structure

- A brain contains about 10^{11} neurons
- Each neuron has about 10^4 connections

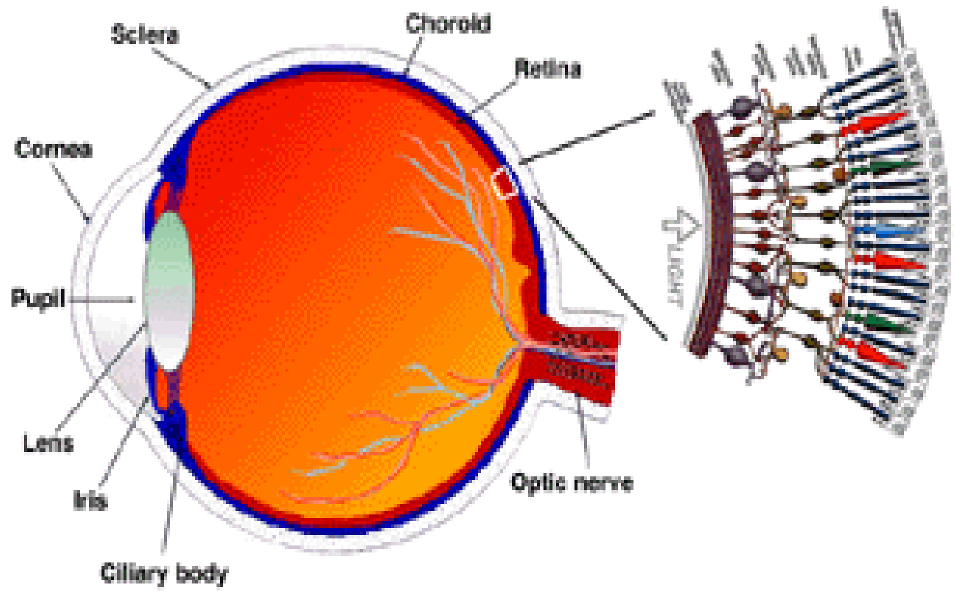


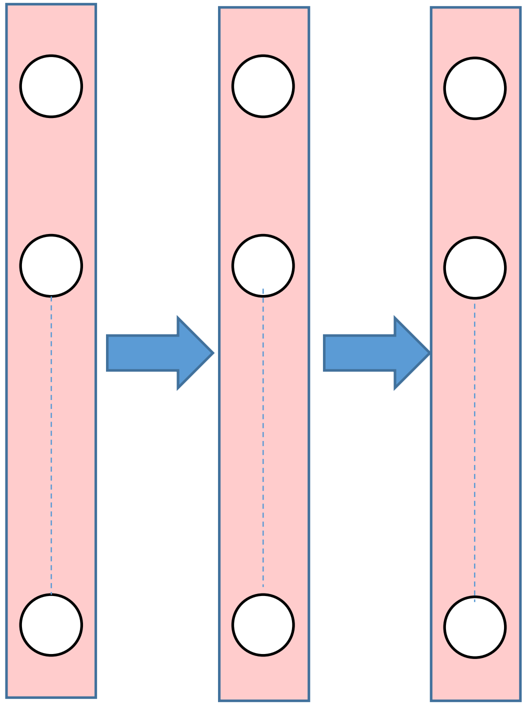
Brief Review of Brain Structure

- Information flow in the brain layer by layer



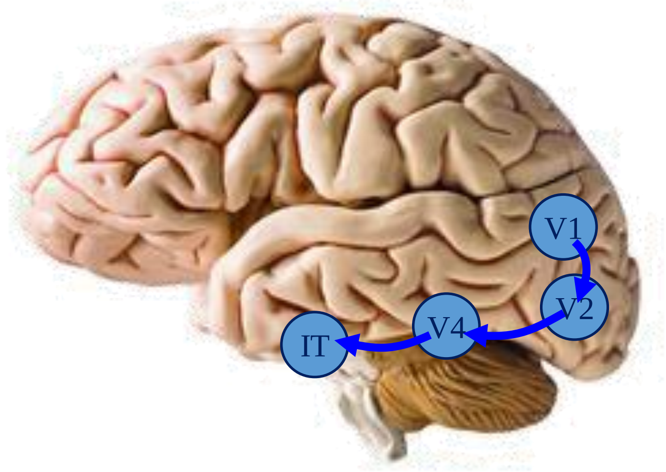
Brief Review of Brain Structure

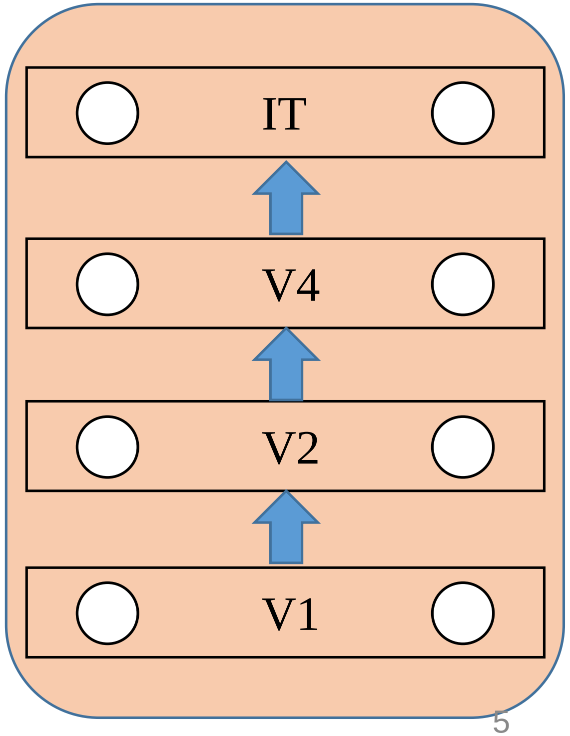




Third layer Mid layer First layer

Concept of Layers



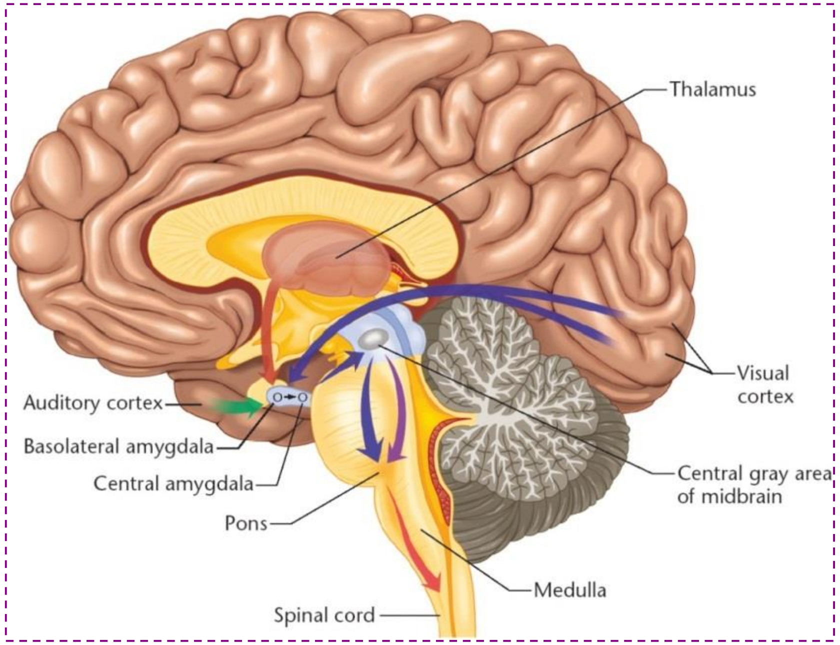


Concept of Layer

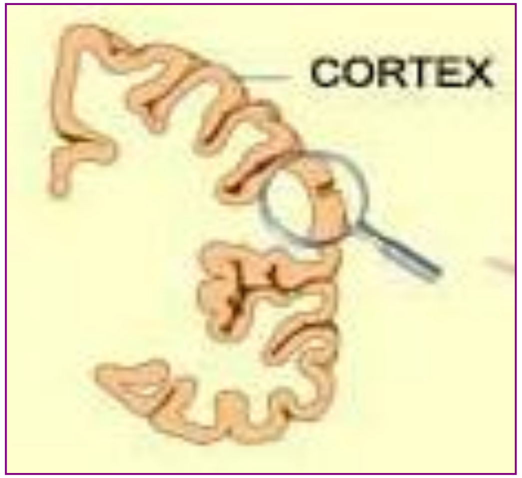
1. Neural network with layers
2. Neurons receive the outputs of neurons at previous layer as inputs.

Brief Review of Brain Structure

- The typical human neocortex:
  - $1000cm^2$
  - Stretched flat, the human neocortical sheet is roughly the size of a large dinner napkin.
  - 2mm thick
  - 30 billion neurons
  - A tiny square millimeter contains an estimated 100,000 neurons.
  - 100 trillion synapses.



Almost everything related to intelligence such as: perception, language, imagination, mathematics, art, music, and planning— occurs on the neocortex.

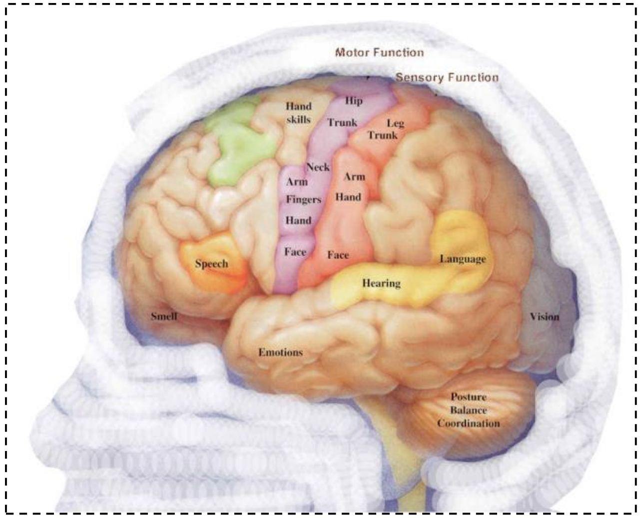


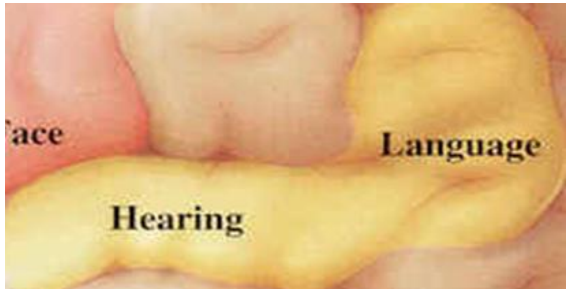
Brief Review of Brain Structure

- A neocortex is divided into several functional regions, such as visual area, auditory area, touch area, associate area, etc..
  - The functional regions are arranged in an irregular patchwork quilt physically.
  - Nearly identical architecture.

Cortex map

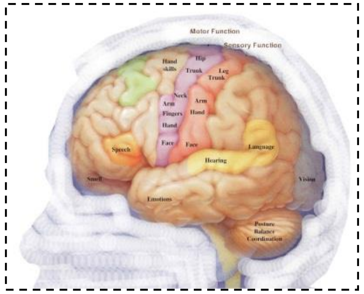
Concept of regions

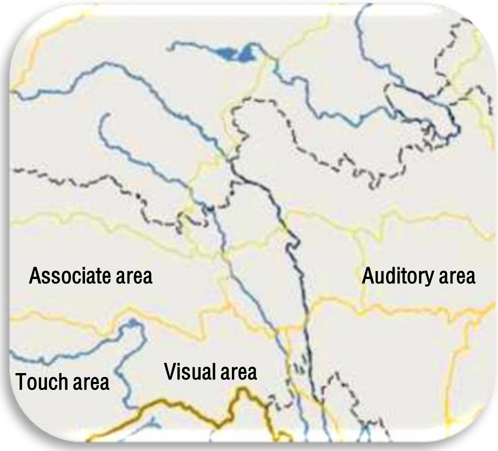


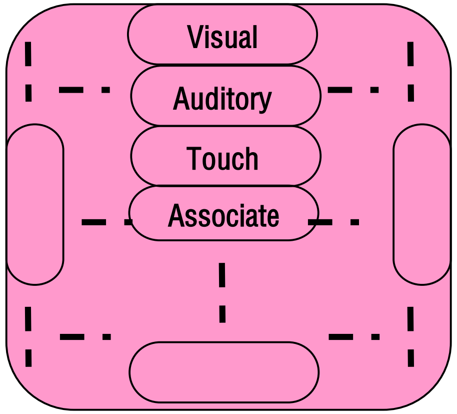


Brief Review of Brain Structure

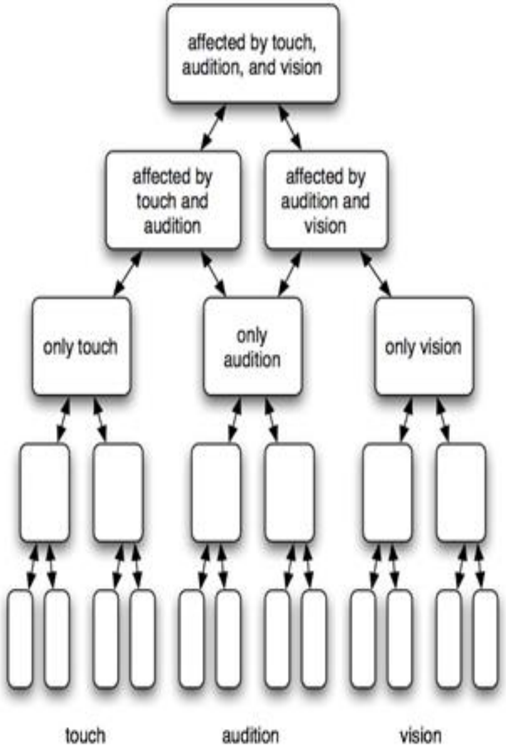
- How are the regions connected?
  - Functionally the regions are arranged in a branching hierarchy.
  - Lower regions feed information up to higher regions.
  - Higher regions send feedback down to lower regions.







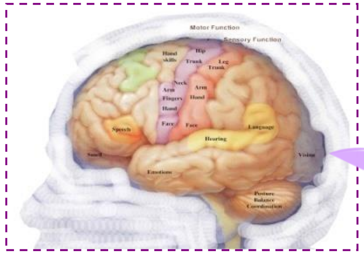
Concept of Hierarchy Connection

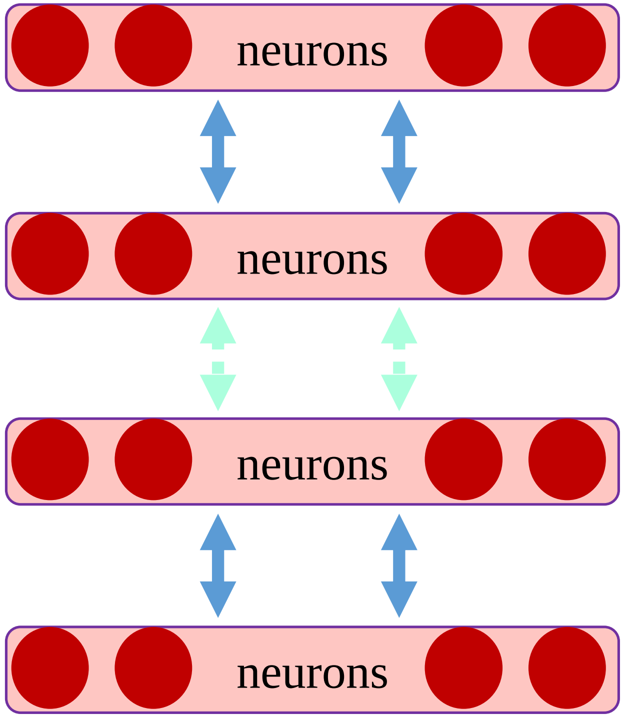


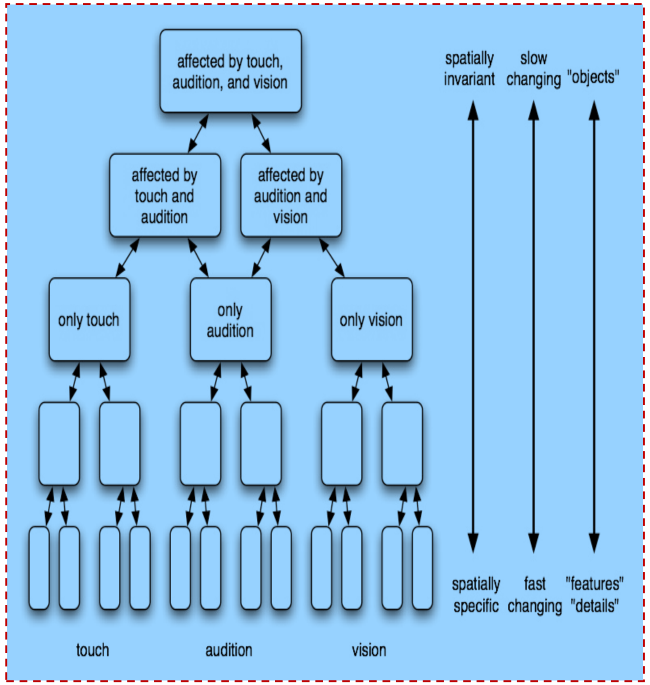
Brief Review of Brain Structure

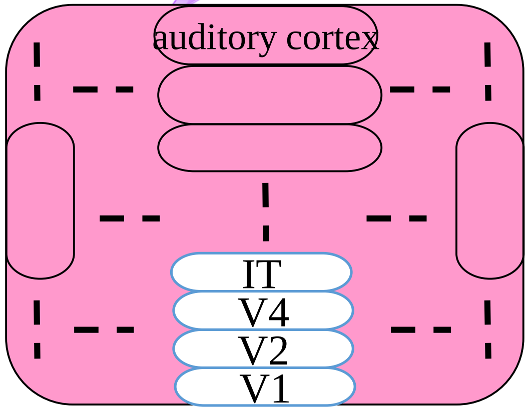
Region

- Physically: irregular quilt.
- Functionally: hierarchy.

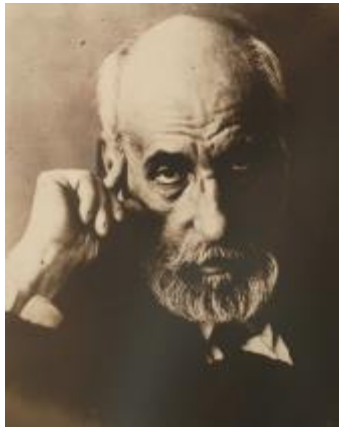


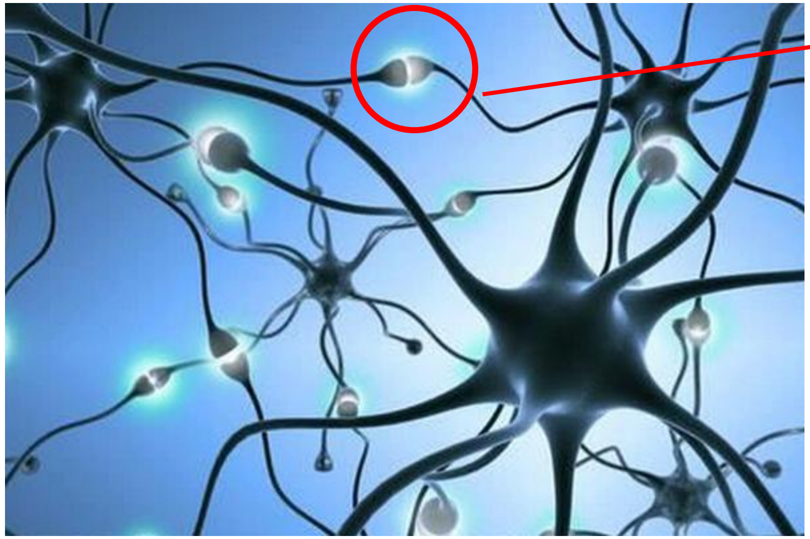




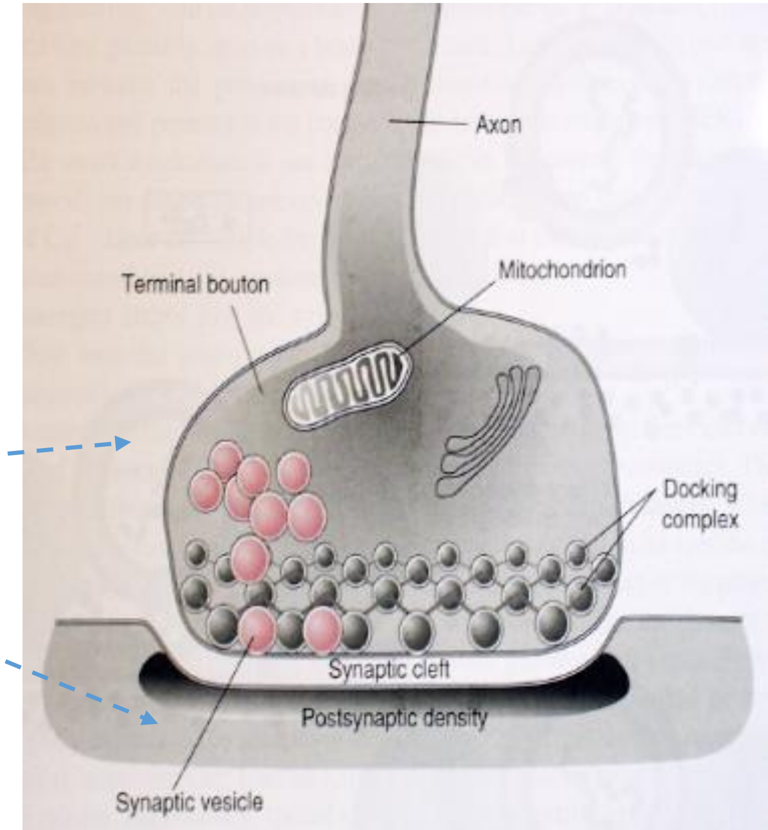


Brief Review of Brain Structure

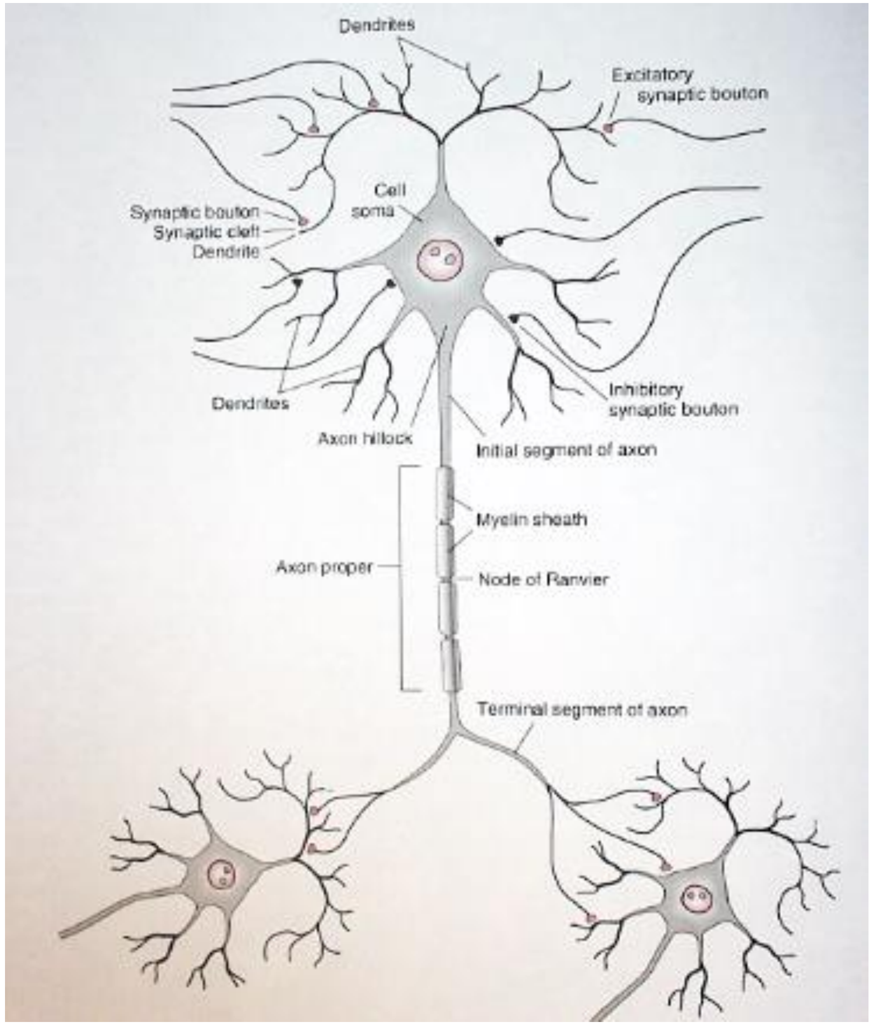




Synapse



Brief Review of Brain Structure



Neural Network = Neurons + Connections

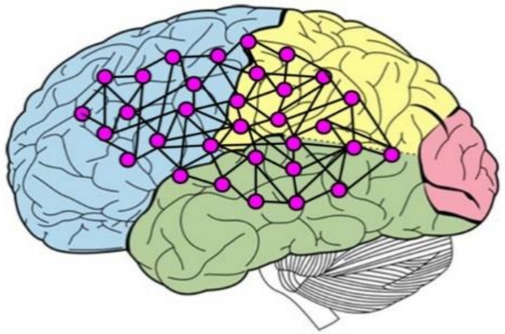
The information flow in the network by some kind of electricity.

Problem: Can we develop computational models for the neural network?

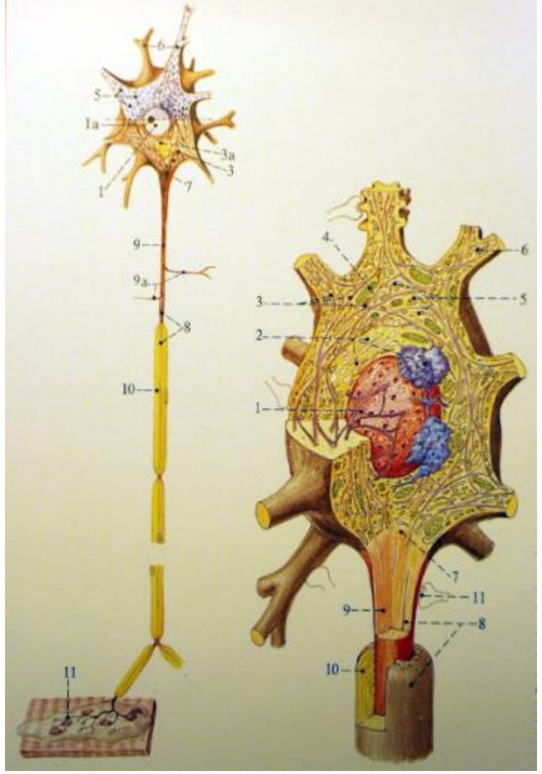
Outline

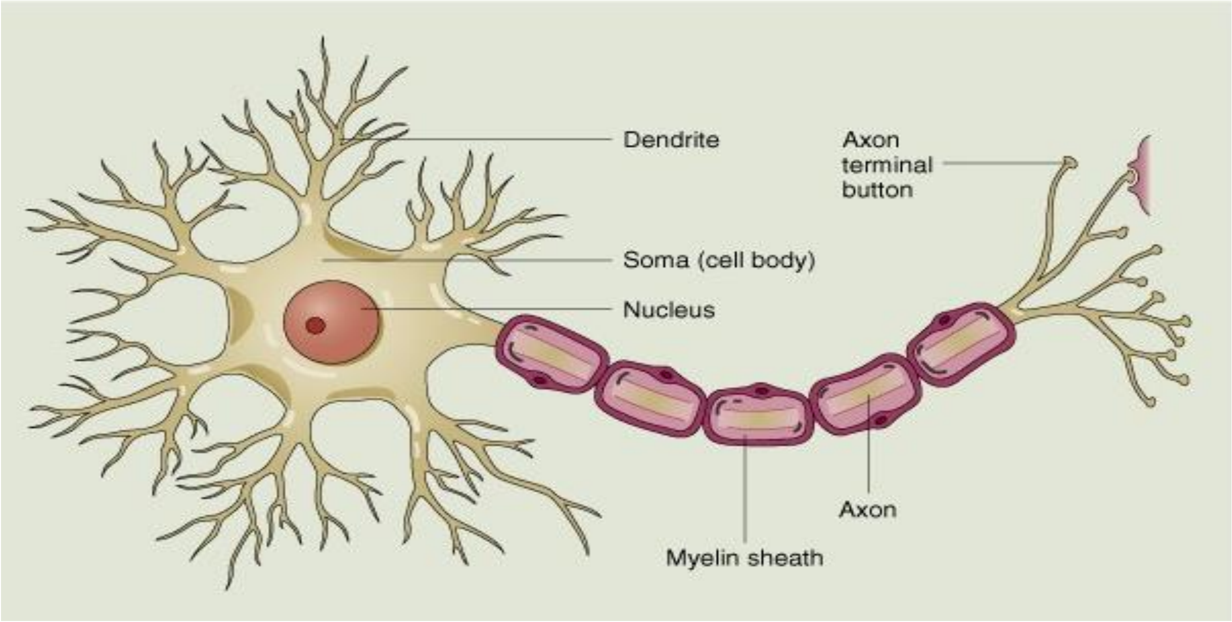
- Brief Review of Brain Structure
- Computational Model of Neurons
- Computational Model of Neural Networks
- Various Models of Neural Networks
- Discussions
- Assignments

Computational Model of Neurons



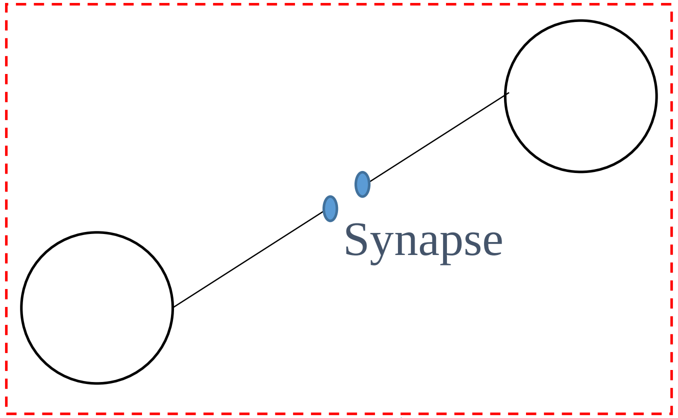
- Basic Components
  - Soma (cell body)
  - Dendrite
  - Axon
- Basic Functions
  - Collecting
  - Functioning
  - Transferring
- Characters
  - Multi-inputs
  - Mon-output





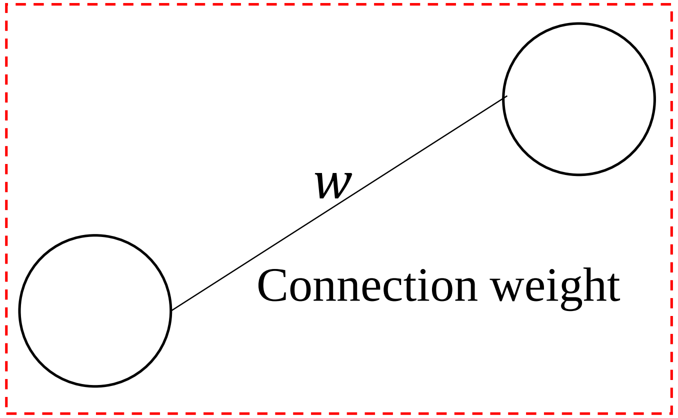
Computational Model of Neurons

Topological abstracting

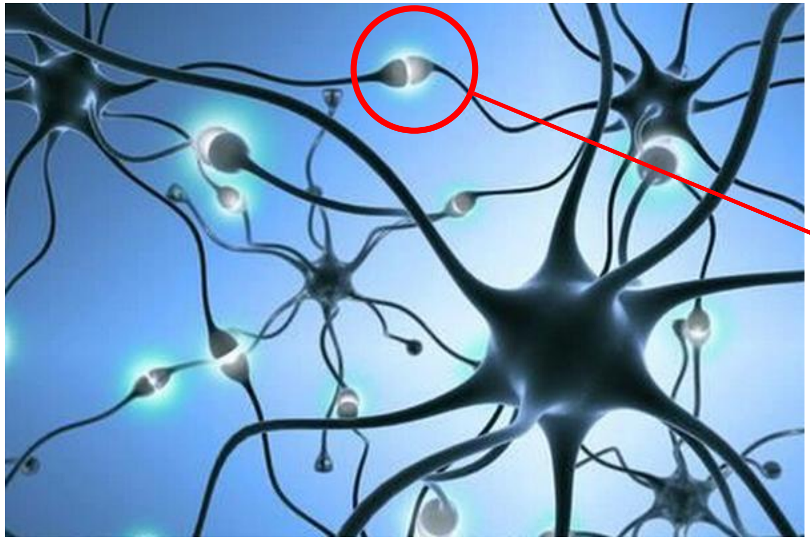


Step 2

Computational model



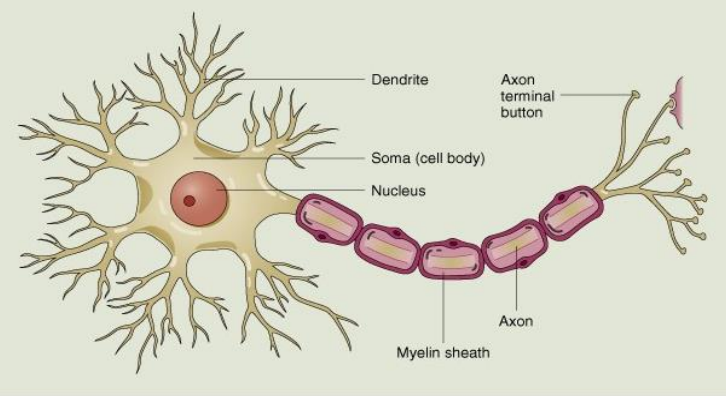
Step 1



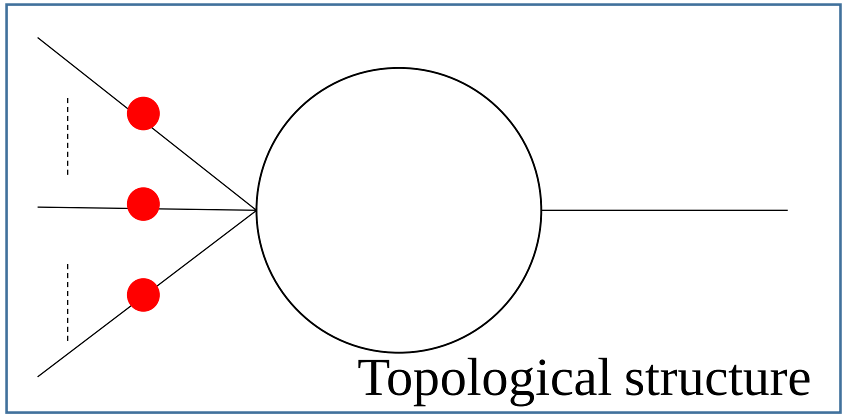
$w > 0$, exciting connection
$w = 0$, no connection
$w < 0$, inhibition connection

Synapse

Computational Model of Neurons

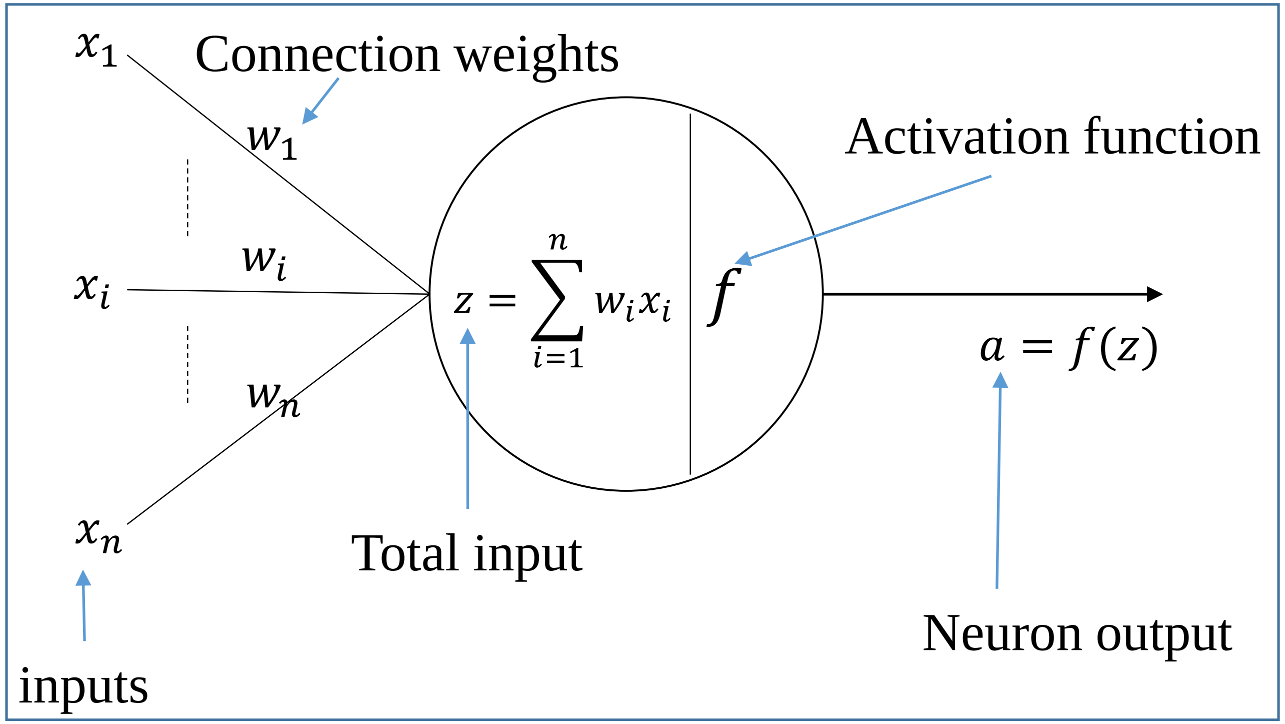


Neuron structure



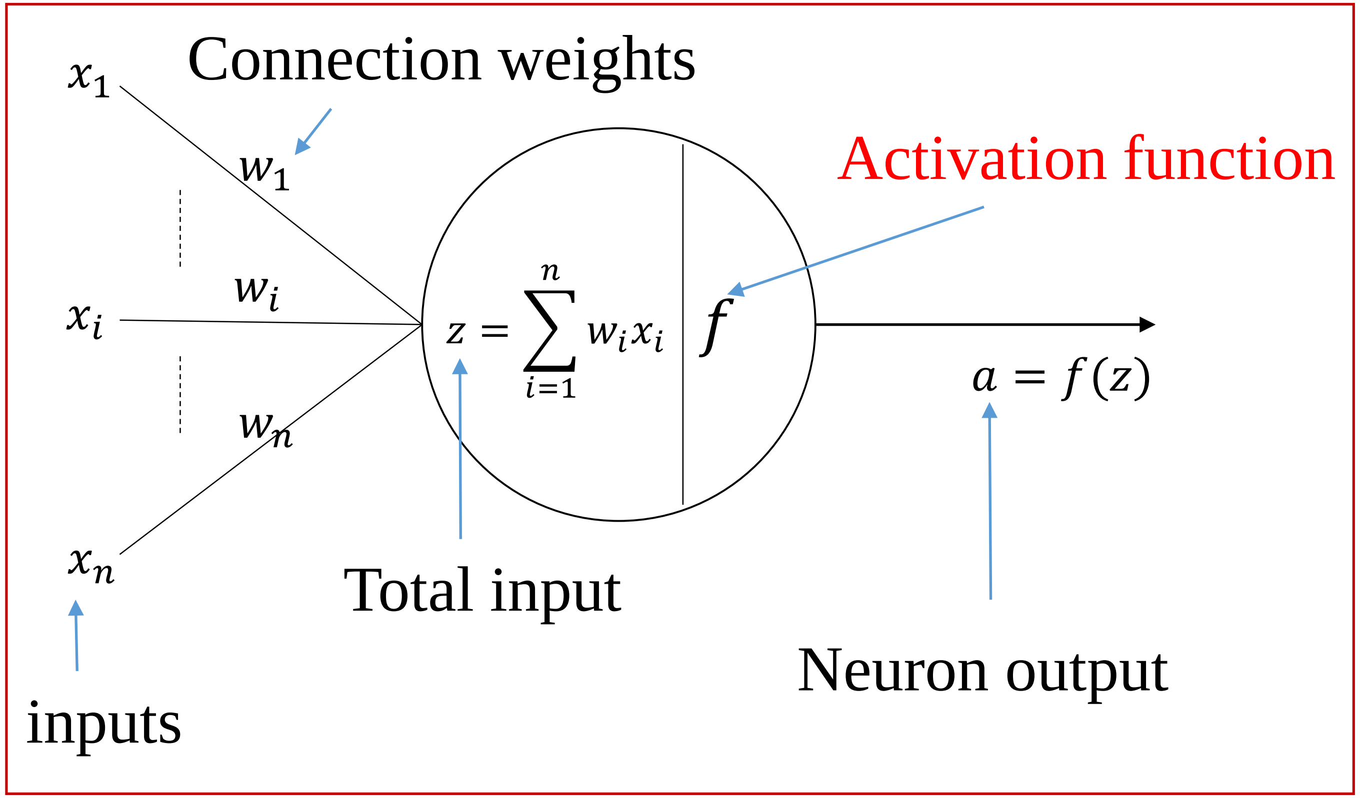
$$a = f\left(\sum_{i=1}^{n} w_i x_i\right)$$

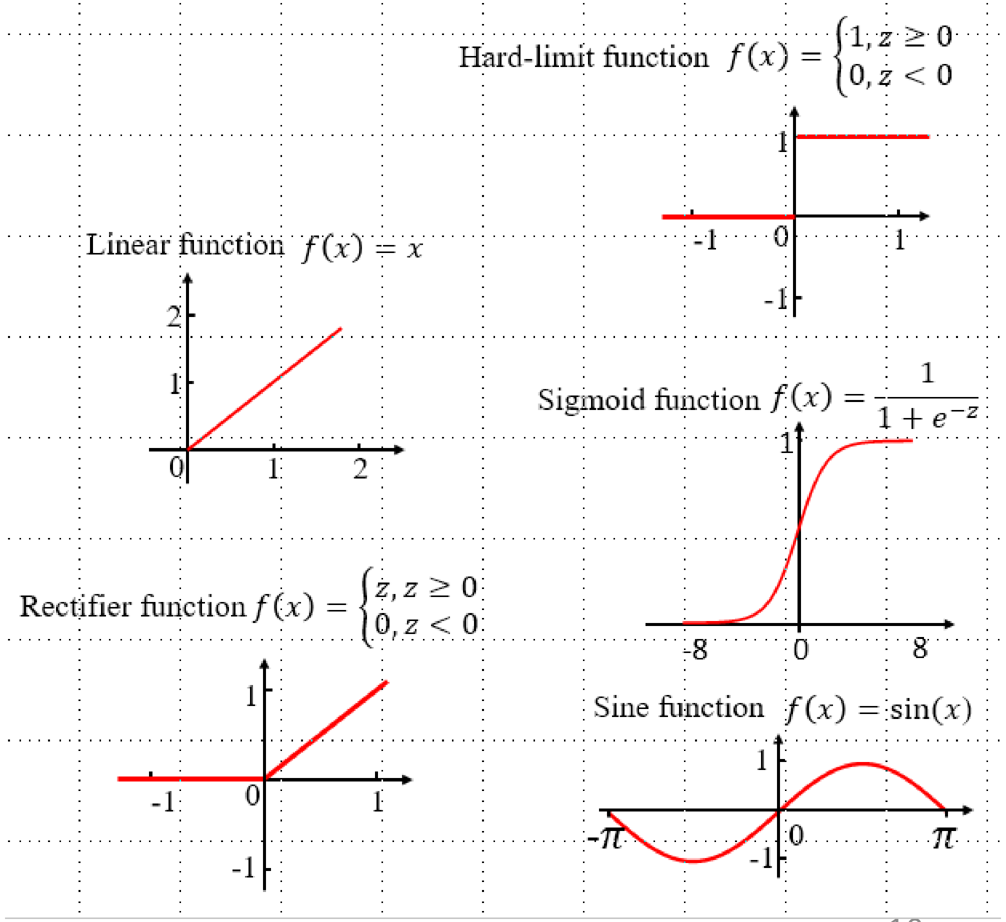
Computational model



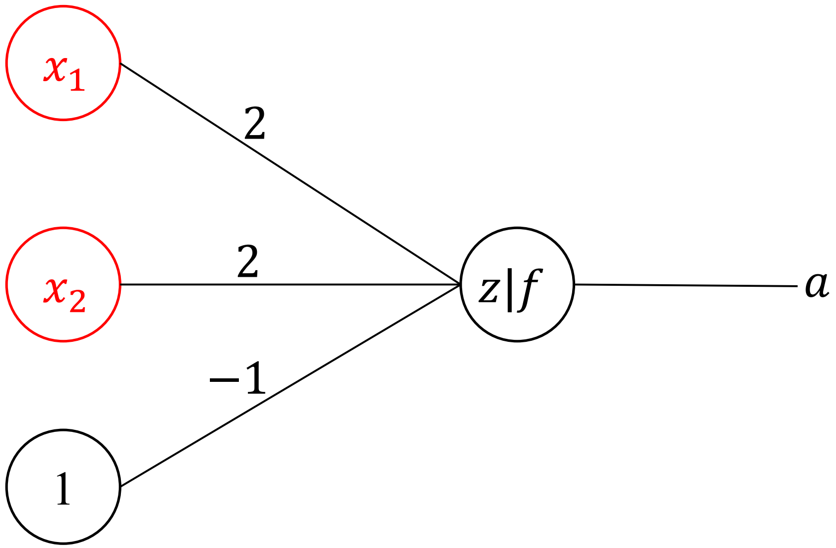
Computational Model of Neurons

$$a = f\left(\sum_{i=1}^{n} w_i x_i\right)$$





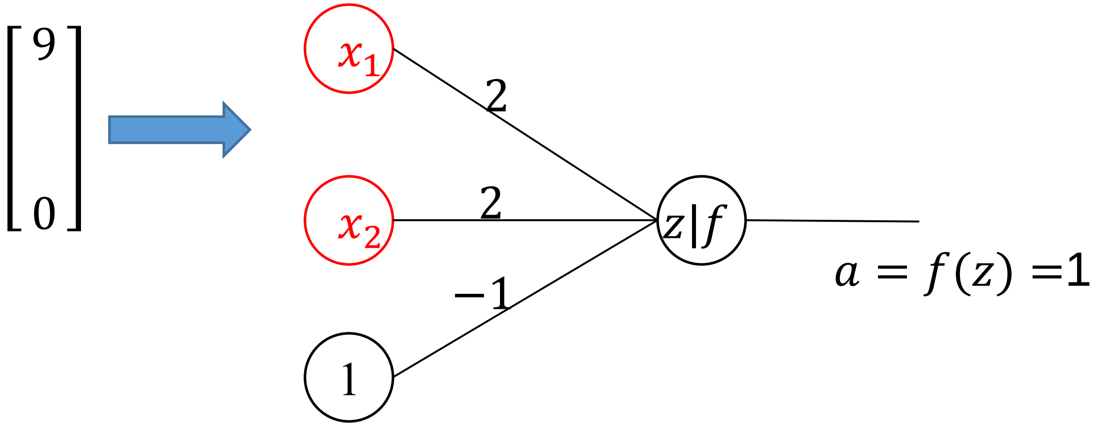
A Simple Example



$$a = f(2x_1 + 2x_2 - 1)$$

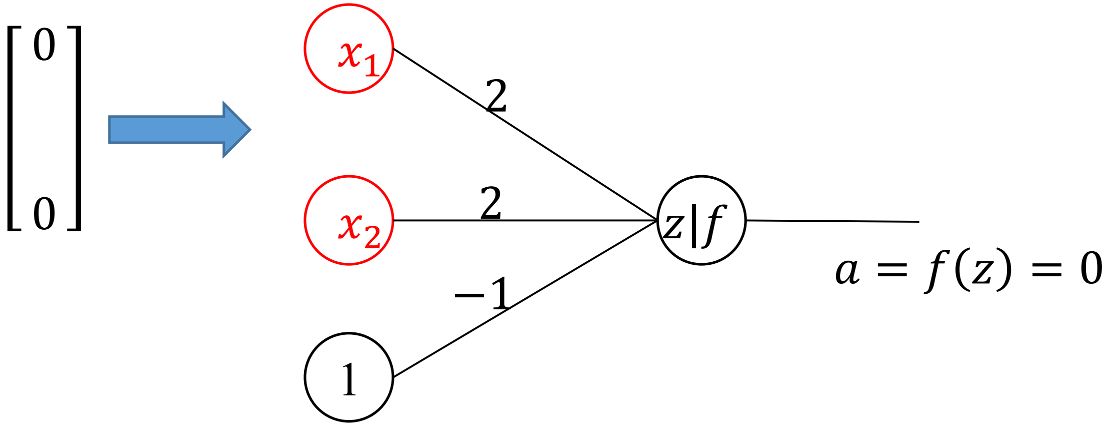
$$z = 2x_1 + 2x_2 - 1$$

$$f(s) = \begin{cases} 1, & s \geq 0 \\ 0, & otherwise \end{cases}$$



$$z = 2x_1 + 2x_2 - 1 = 2 \cdot 9 + 2 \cdot 0 + 1 \cdot (-1) = 17$$

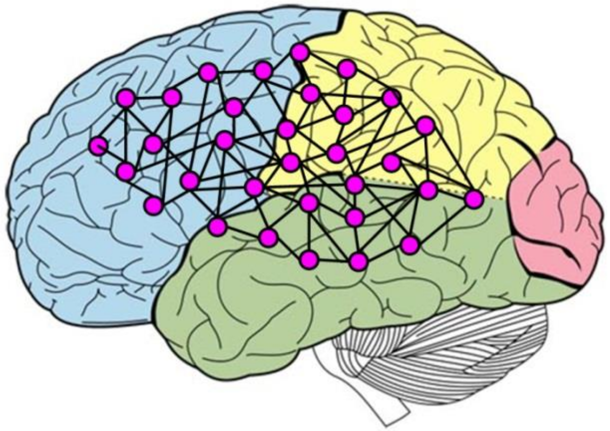
$$a = f(z) = 1$$

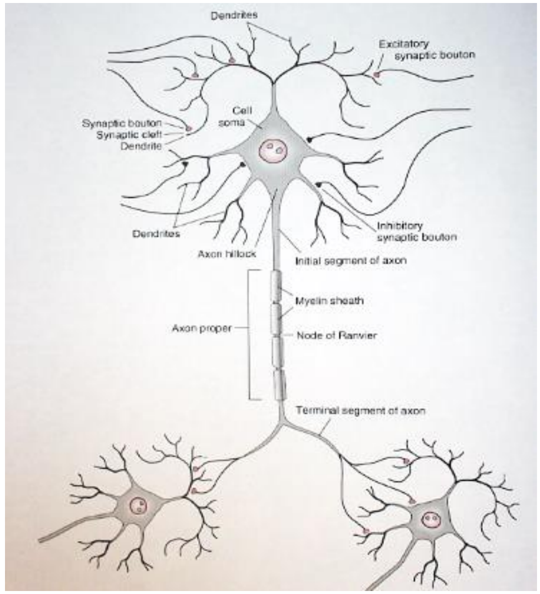


Outline

- Brief Review of Brain Structure
- Computational Model of Neurons
- Computational Model of Neural Networks
- Various Models of Neural Networks
- Discussions
- Assignments

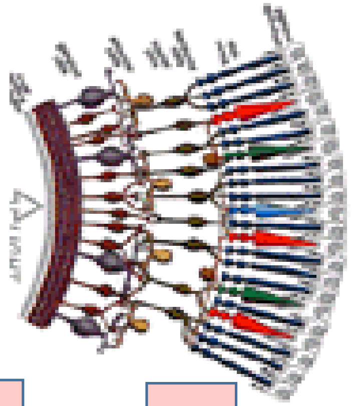
Computational Model of Neural Networks

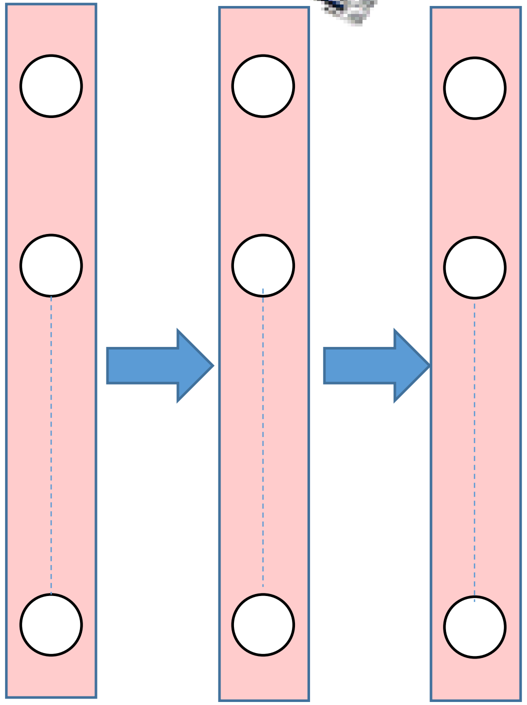




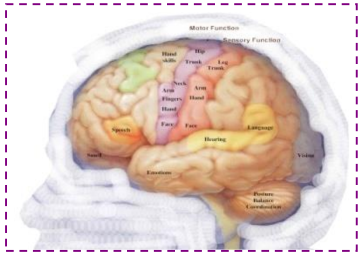
Neural Network = Neurons + Connections

Computational Model of Neural Networks

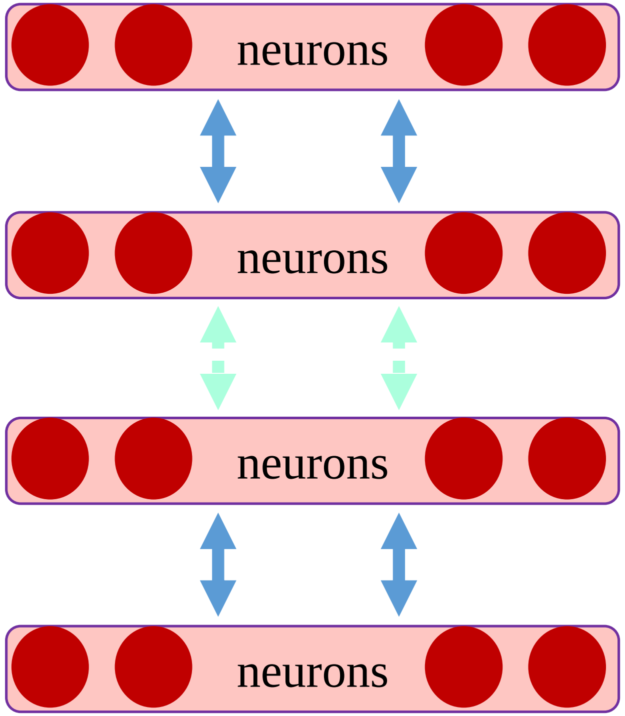




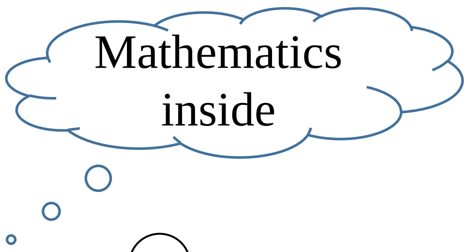
There exist layer structures for neurons.

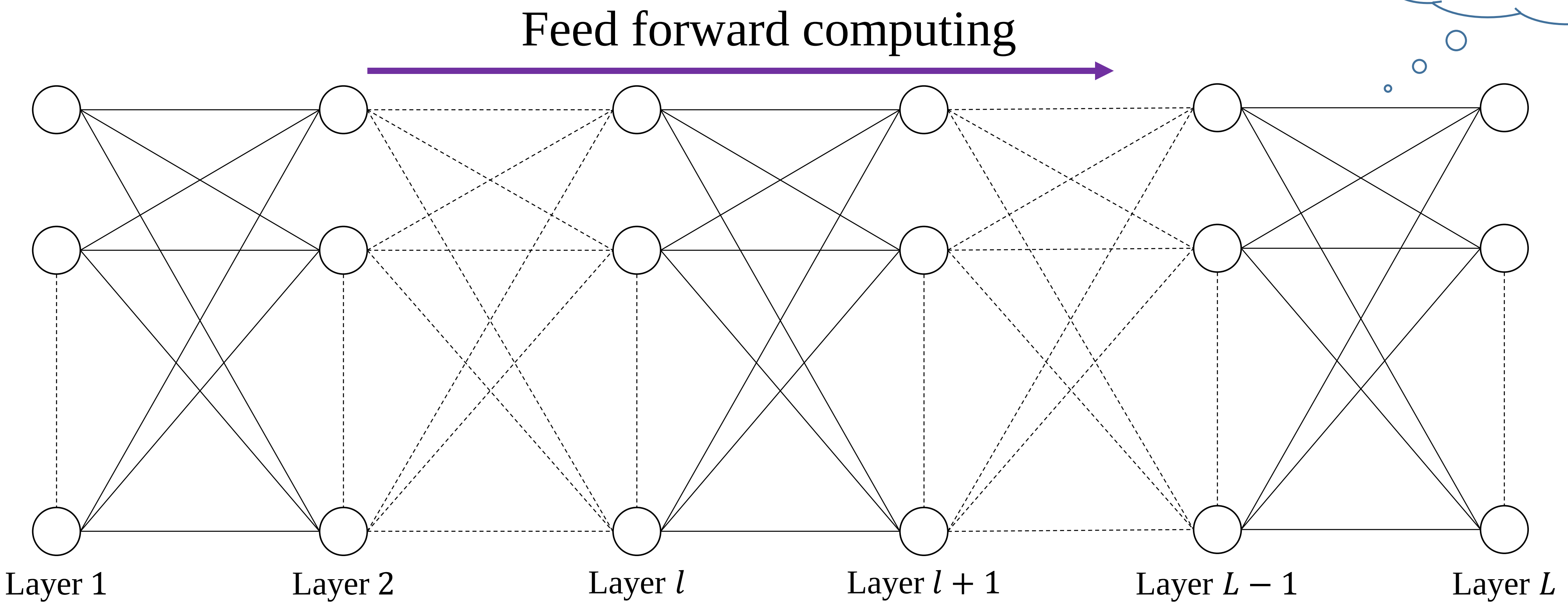


Neurons are
structured in
Layer way.



Computational Model of Neural Networks



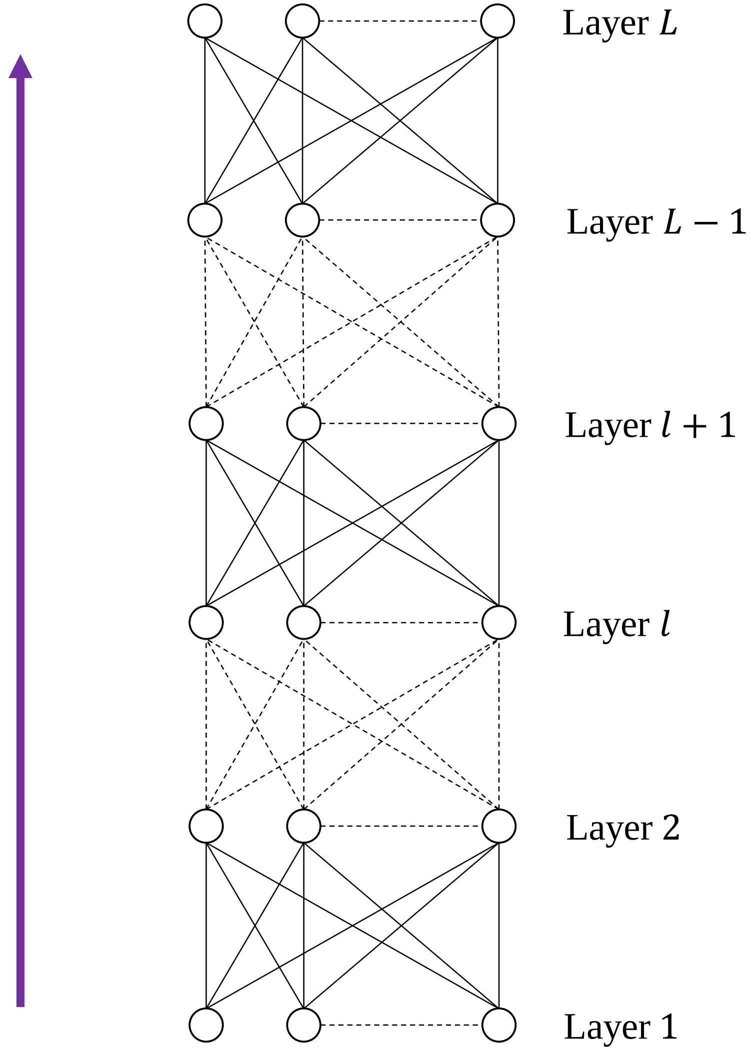


Topological structure of neural networks

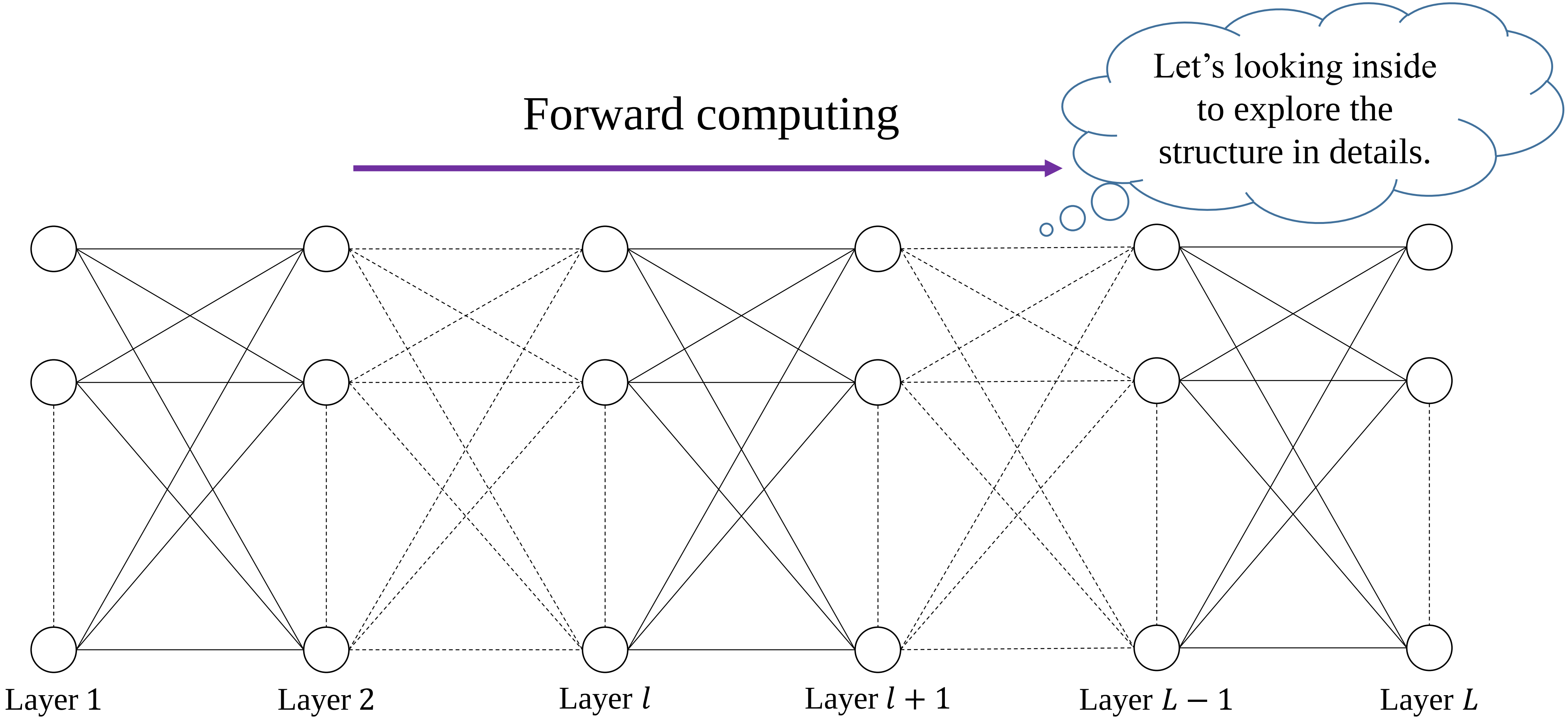
- Two important characters:
  - No any connection in any layer
  - No any connection across any layer

Computational Model of Neural Networks

Another view to NN structure

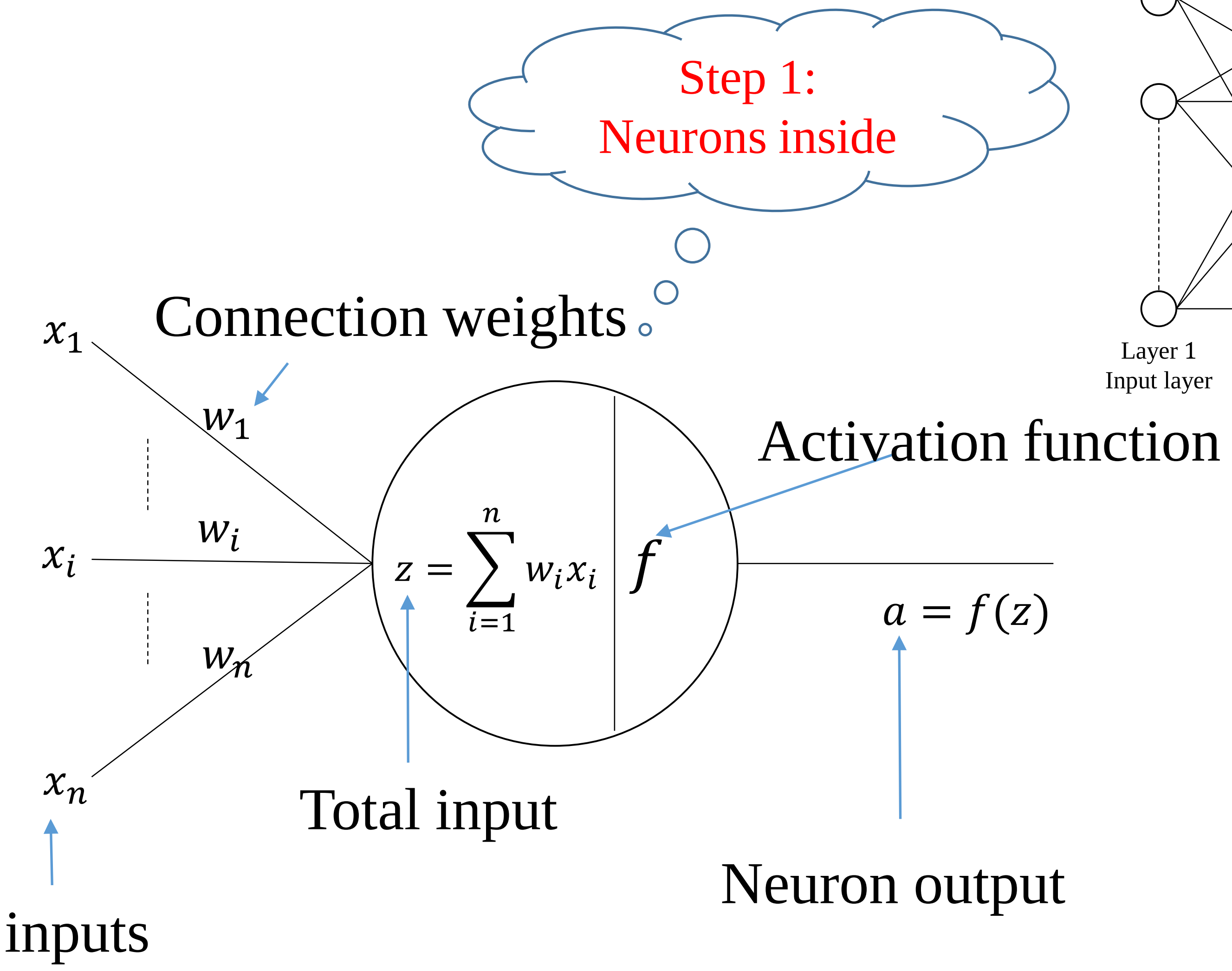


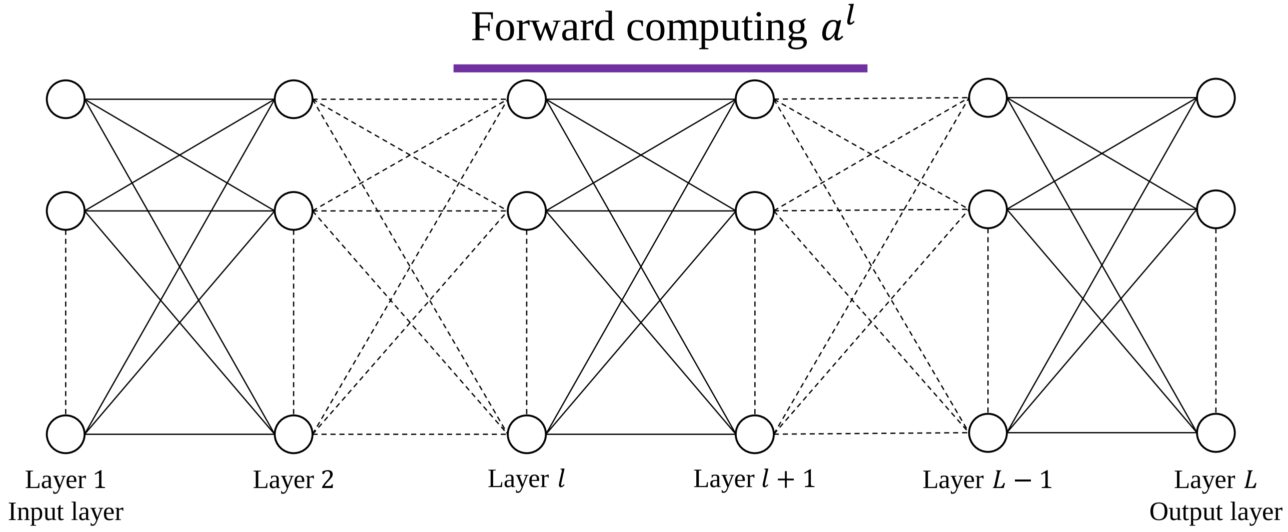
Computational Model of Neural Networks

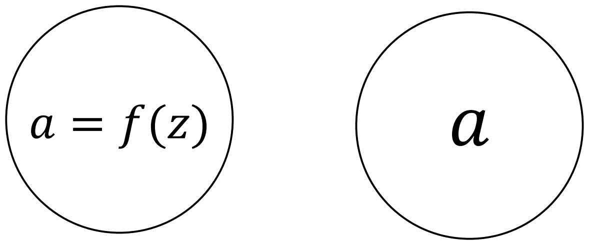


Topological structure of neural networks

Computational Model of Neural Networks



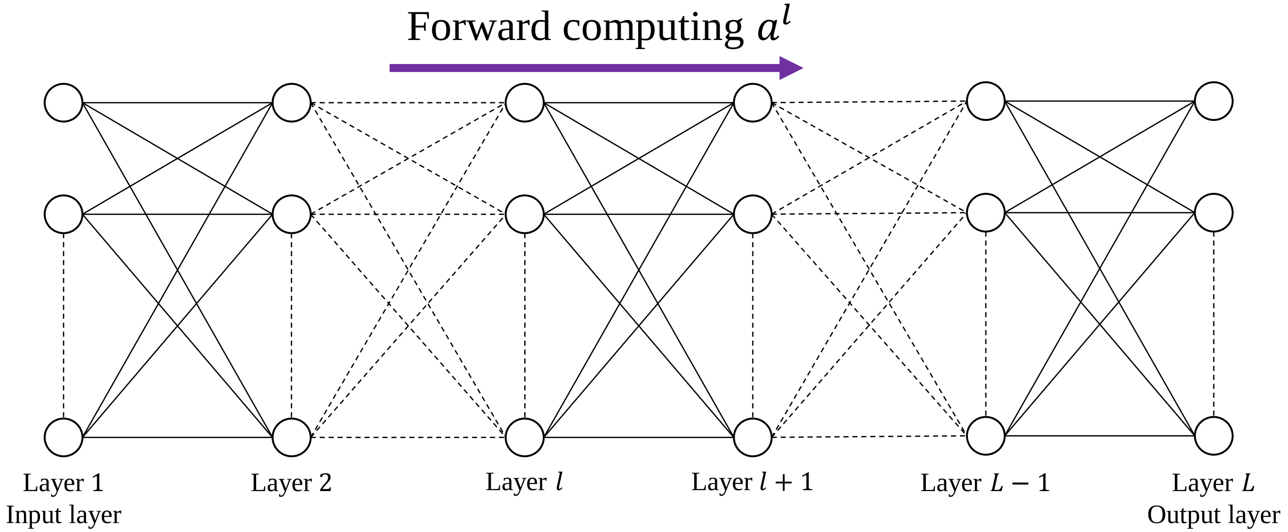


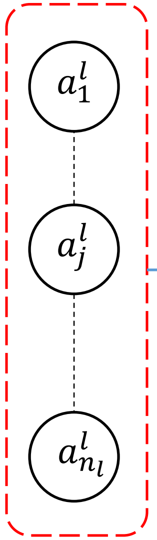


Computational Model of Neural Networks

Step 2：
Layers inside

Layer l contains n_l neurons.





Layer l

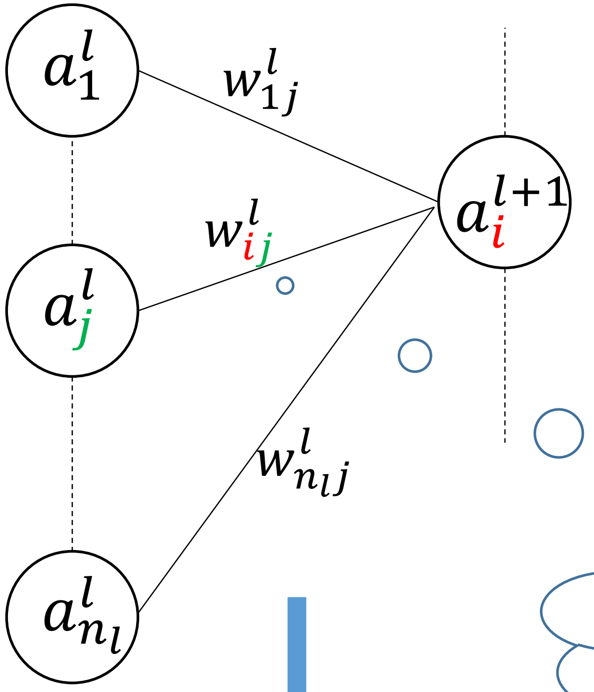
$$a_j^l = f(z_j^l)$$

The neuron located in l layer j^{th} place, a_j^l denotes the output value of the neuron and is called the neuron prediction of the neuron in l layer at j^{th} place.

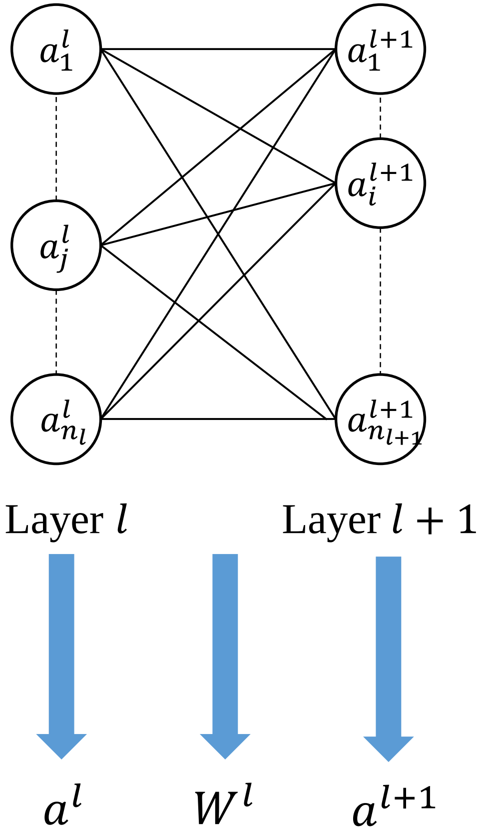
$$\text{Layer output } a^l = \begin{bmatrix} a_1^l \\ \vdots \\ a_j^l \\ \vdots \\ a_{n_l}^l \end{bmatrix}$$

a^l is called the network prediction at l layer

Computational Model of Neural Networks



Step 3:
Connections
between layers

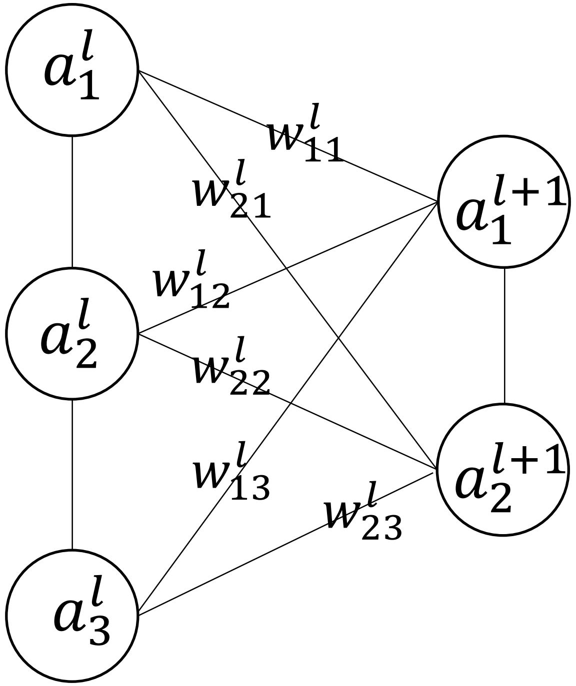


$$W^l = \begin{bmatrix} w_{11}^l & \cdots & w_{1n_l}^l \\ \vdots & w_{ij}^l & \vdots \\ w_{n_{l+1}1}^l & \cdots & w_{n_{l+1}n_l}^l \end{bmatrix}_{n_{l+1}\times n_l}$$

a^l is the input of $l+1$ layer. a^{l+1} can be calculated out from a^l. a^l is called the network prediction at l layer. a^L is called the network prediction.

Computational Model of Neural Networks

Step 3:
Connections between layers:
An Example



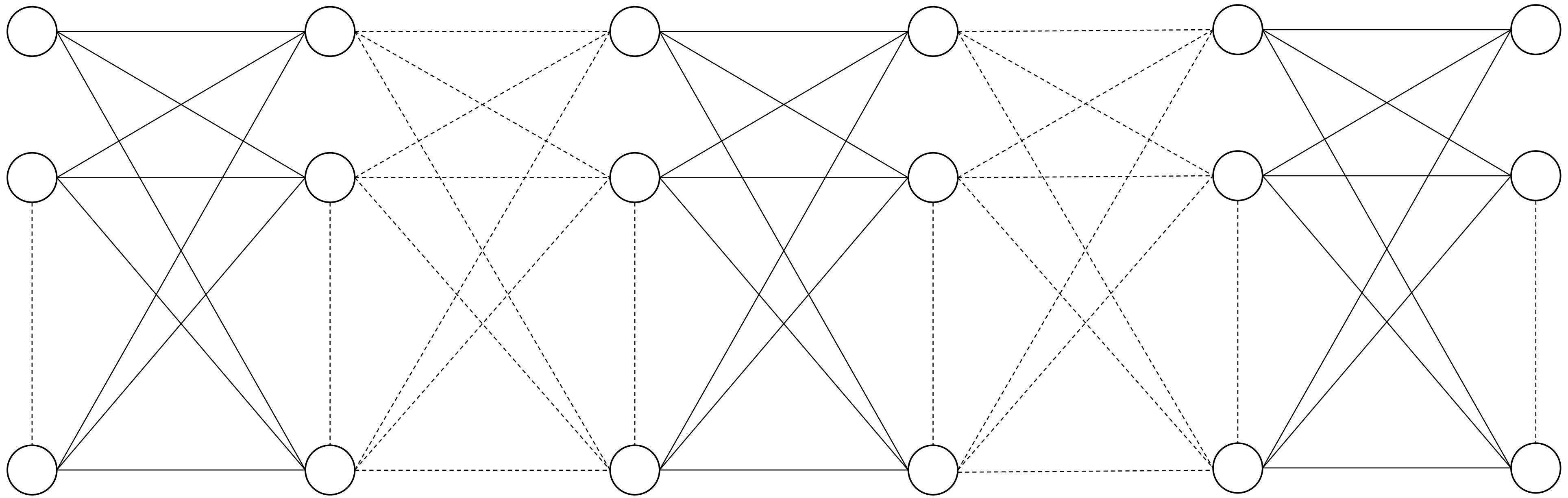
$$W^l = \begin{bmatrix} w_{11}^l & w_{12}^l & w_{13}^l \\ w_{21}^l & w_{22}^l & w_{23}^l \end{bmatrix}_{2\times 3}$$

Standard Structure of Neural Networks

Forward computing

Everything here

	Layer 1		Layer 2		Layer l		Layer $l+1$		Layer $L-1$		Layer L

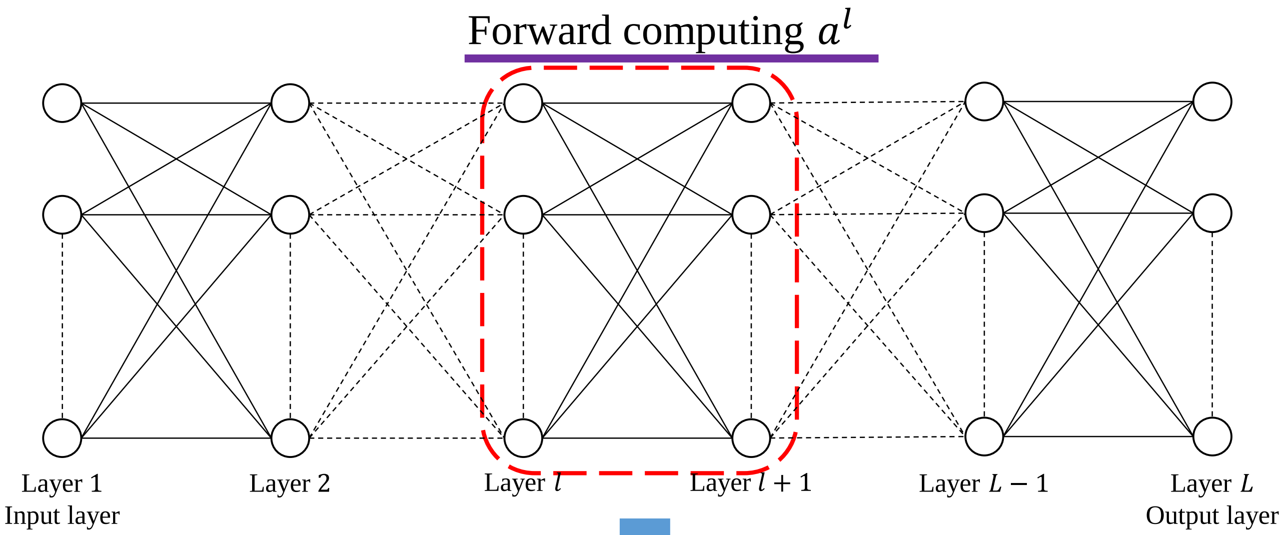


neurons	n_1		n_2		n_l		n_{l+1}		n_{L-1}		n_L
weights		W^1				W^l				W^{L-1}	
predictions	a^1		a^2		a^l		a^{l+1}		a^{L-1}		a^L

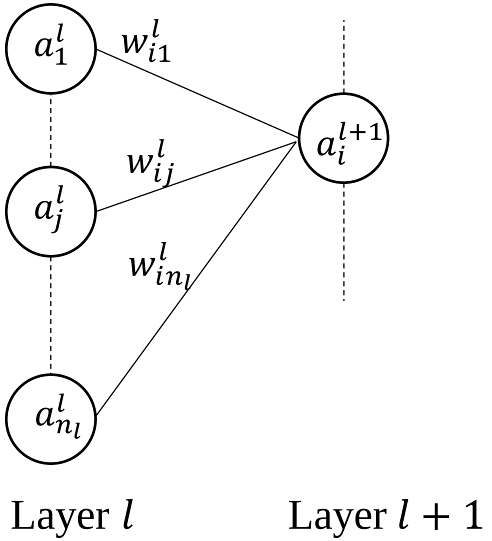
Layer l prediction

Network prediction

Computational Model of Neural Networks

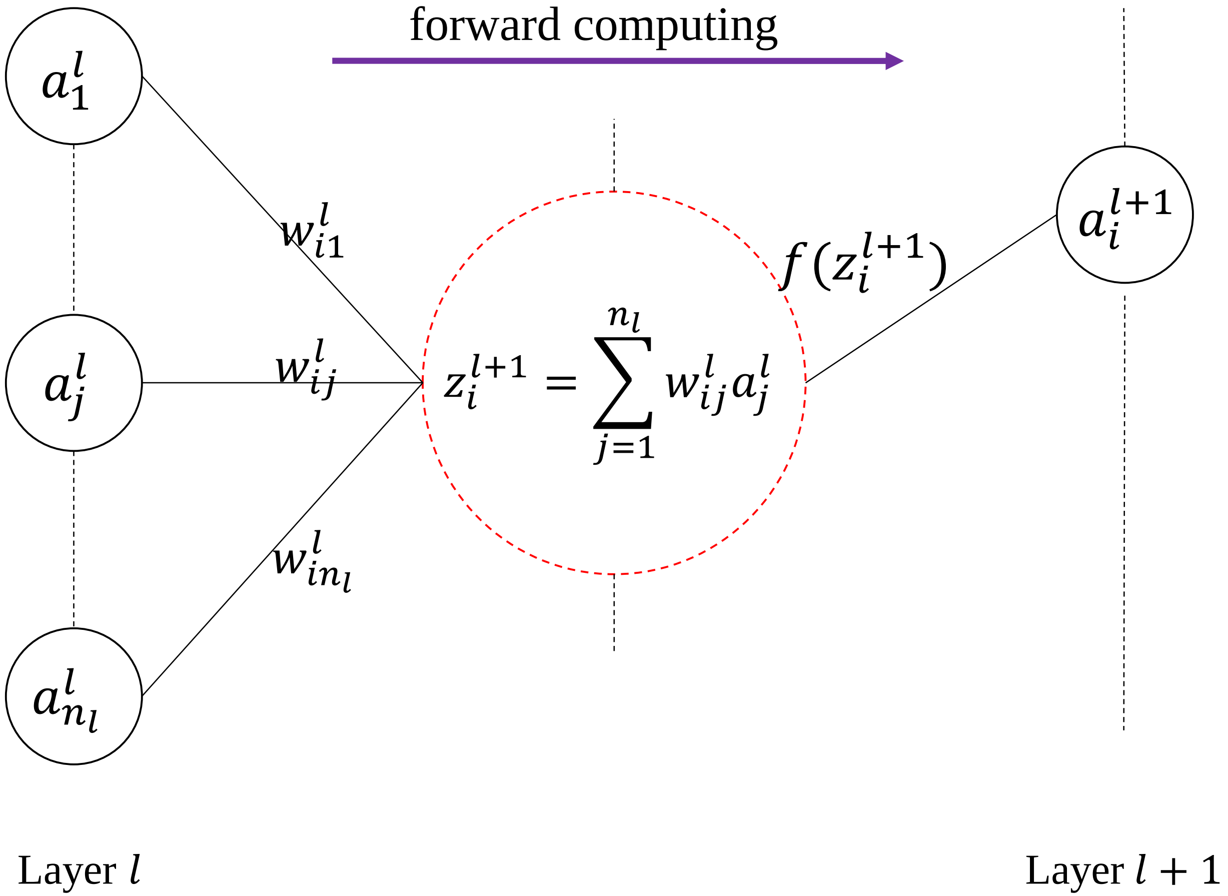


Problem:
How to do the forward computing?



How to use $a_1^l, \cdots, a_{n_l}^l$ and $w_{i1}^l, \cdots, w_{in_l}^l$ to compute a_i^{l+1}?

Computational Model of Neural Networks

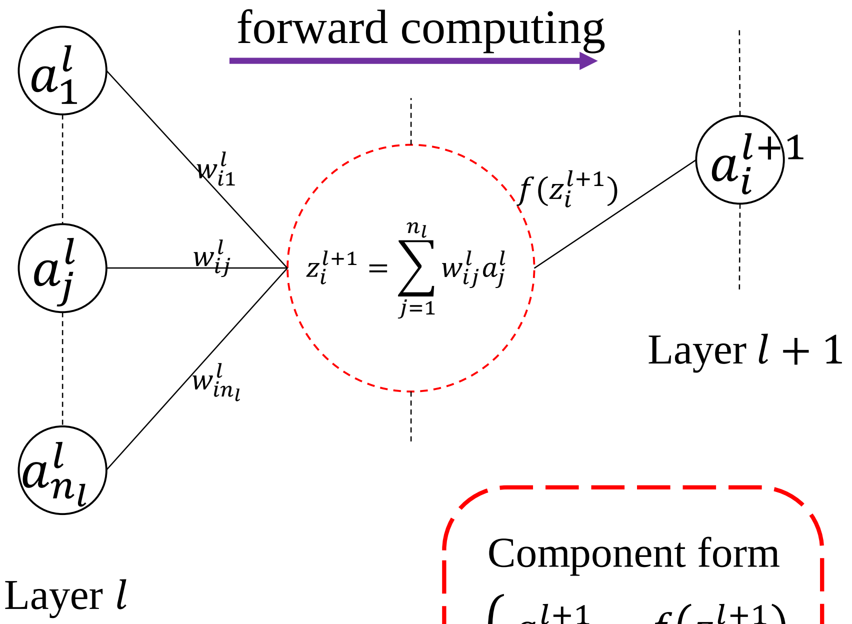


$$
\begin{cases}
a_i^{l+1} = f\left(z_i^{l+1}\right) \\
z_i^{l+1} = \sum_{j=1}^{n_l} w_{ij}^l a_j^l
\end{cases}
$$

$\Updownarrow$

$$
a_i^{l+1} = f\left(\sum_{j=1}^{n_l} w_{ij}^l a_j^l\right)
$$

Computational Model of Neural Networks

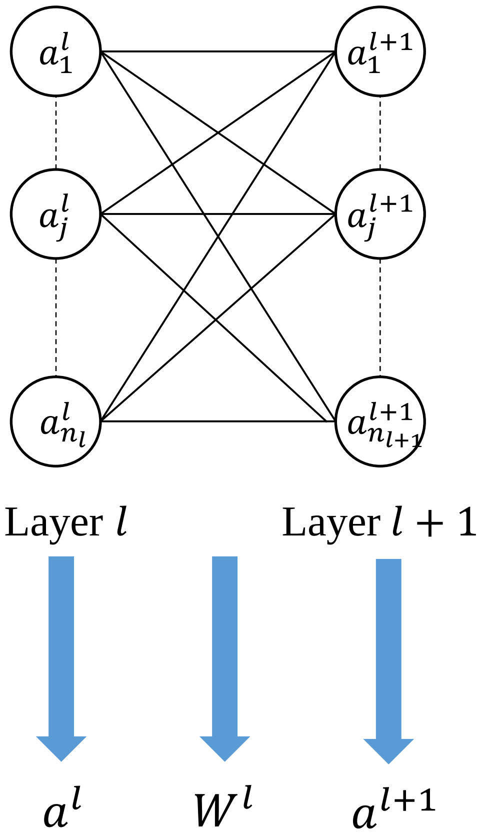


Component form

$$\begin{cases} a_i^{l+1} = f\left(z_i^{l+1}\right) \\ z_i^{l+1} = \sum_{j=1}^{n_l} w_{ij}^l a_j^l \end{cases}$$

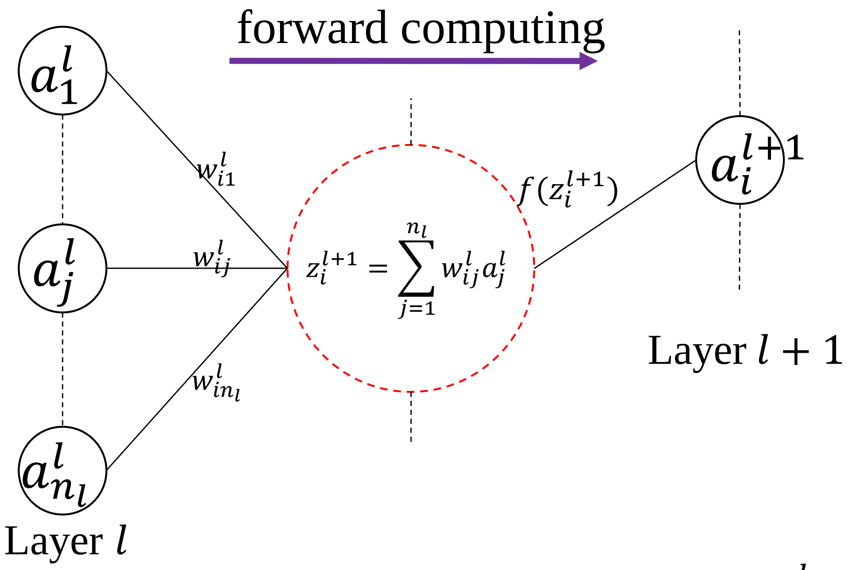
Vector form

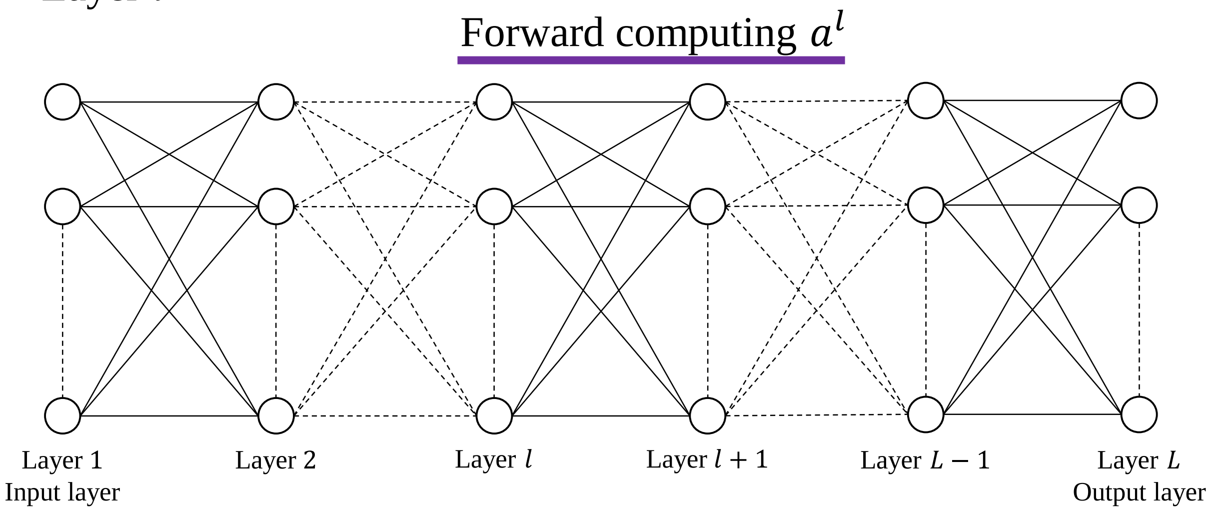
$$\begin{cases} a^{l+1} = f\left(z^{l+1}\right) \\ z^{l+1} = w^l a^l \end{cases}$$



a^l is the input of $l+1$ layer.
a^{l+1} is the representation of a^l.

One page to understand forward computing





Algorithm:

$Input\ W^l, a^1$

$for\ l = 1{:}L$

$$a^{l+1} = fc\left(W^l, a^l\right)$$

$return$

Function $fc\left(W^l, a^l\right)$

$for\ i = 1{:}n_{l+1}$

$$z_i^{l+1} = \sum_{j=1}^{n_l} w_{ij}^l a_j^l$$

$$a_i^{l+1} = f\left(z_i^{l+1}\right)$$

end

Component form

$$\begin{cases} a_i^{l+1} = f\left(z_i^{l+1}\right) \\ z_i^{l+1} = \sum_{j=1}^{n_l} w_{ij}^l a_j^l \end{cases}$$

Vector form

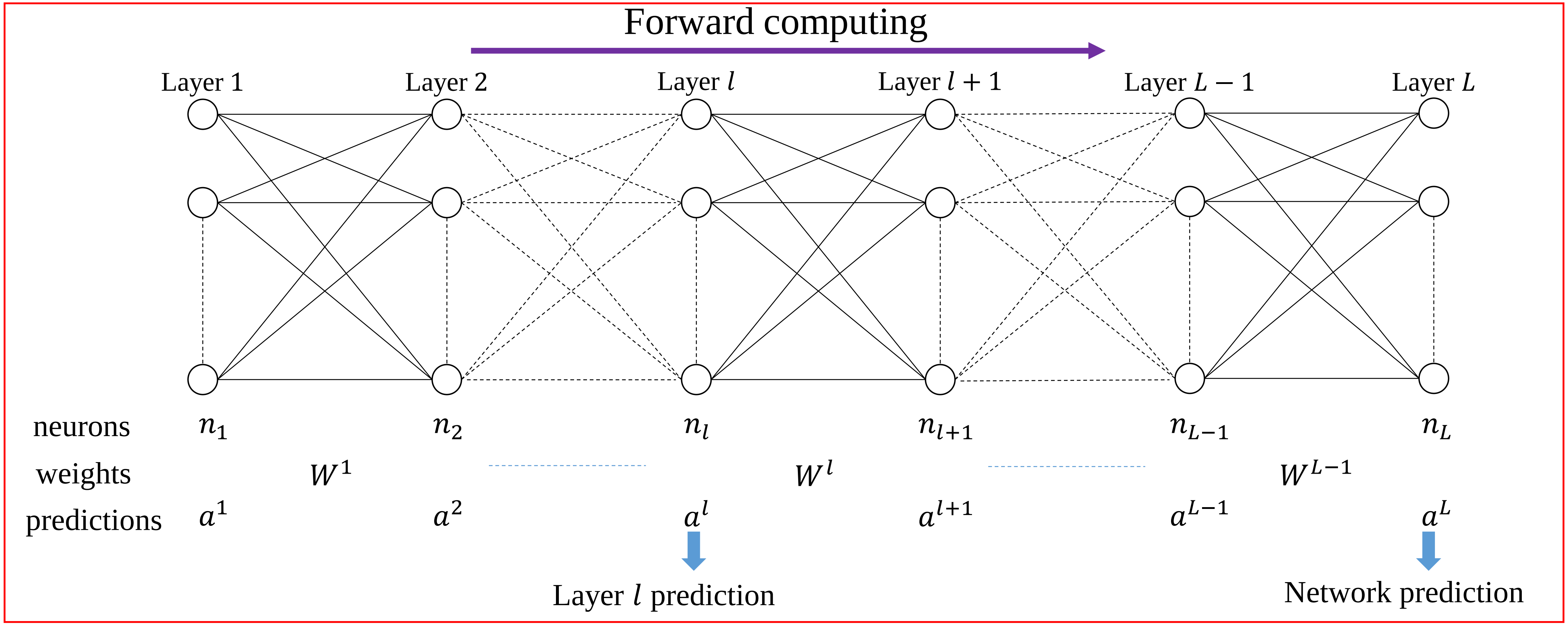
$$\begin{cases} a^{l+1} = f\left(z^{l+1}\right) \\ z^{l+1} = w^l a^l \end{cases}$$

$$a_i^{l+1} = f\left(\sum_{j=1}^{n_l} w_{ij}^l a_j^l\right)$$

Outline

- Brief Review of Brain Structure
- Computational Model of Neurons
- Computational Model of Neural Networks
- Various Models of Neural Networks
- Discussions
- Assignments

Standard Model of Neural Networks

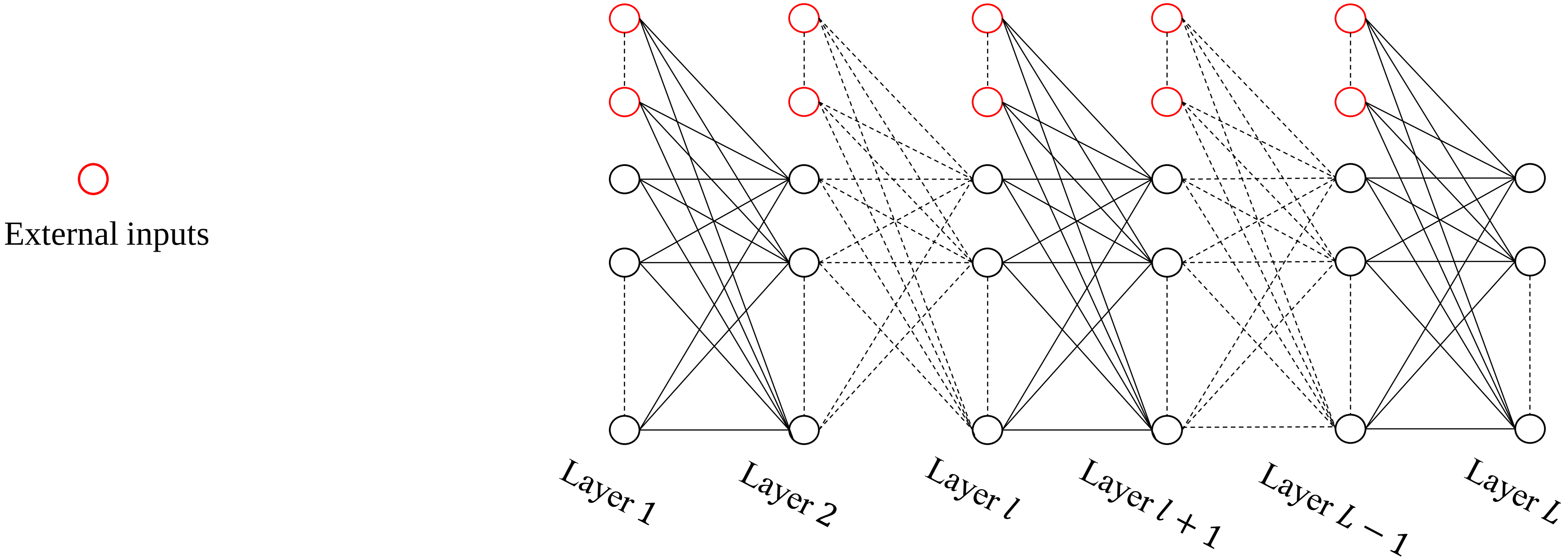


- Two important characters:
  - No any connection in any layer
  - No any connection across any layer

NN Model with External Inputs

External inputs:

If neurons in l layer are not connected to any neurons in previous layer, these neurons are called external inputs of $l + 1$ layer.
External inputs can exist in any layer except the last one.



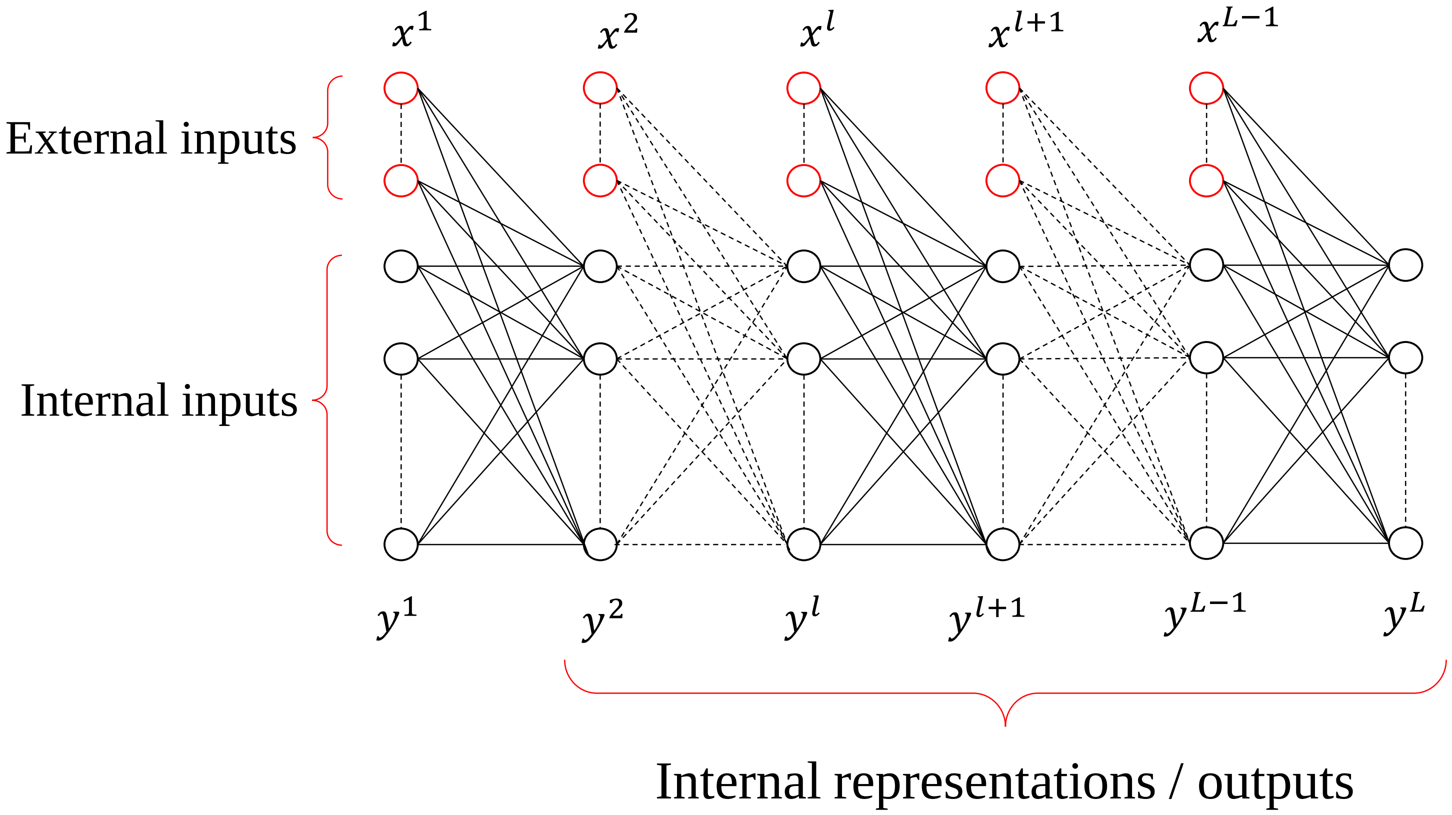
NN Model with External Inputs

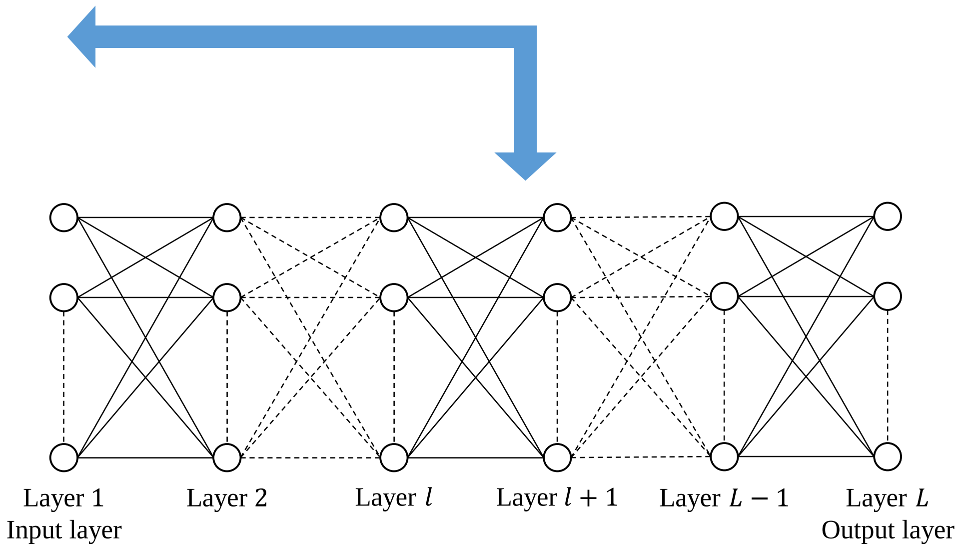
$$a^l = \begin{bmatrix} x^l \\ y^l \end{bmatrix}$$

$$z^{l+1} = W^l a^l$$

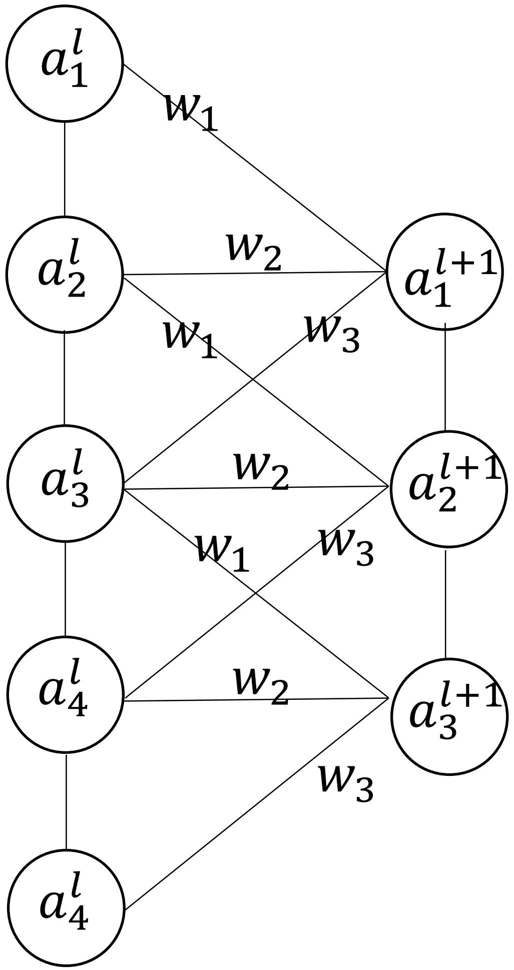
$$y^{l+1} = f\left(z^{l+1}\right)$$

$$a^{l+1} = \begin{bmatrix} x^{l+1} \\ y^{l+1} \end{bmatrix}$$



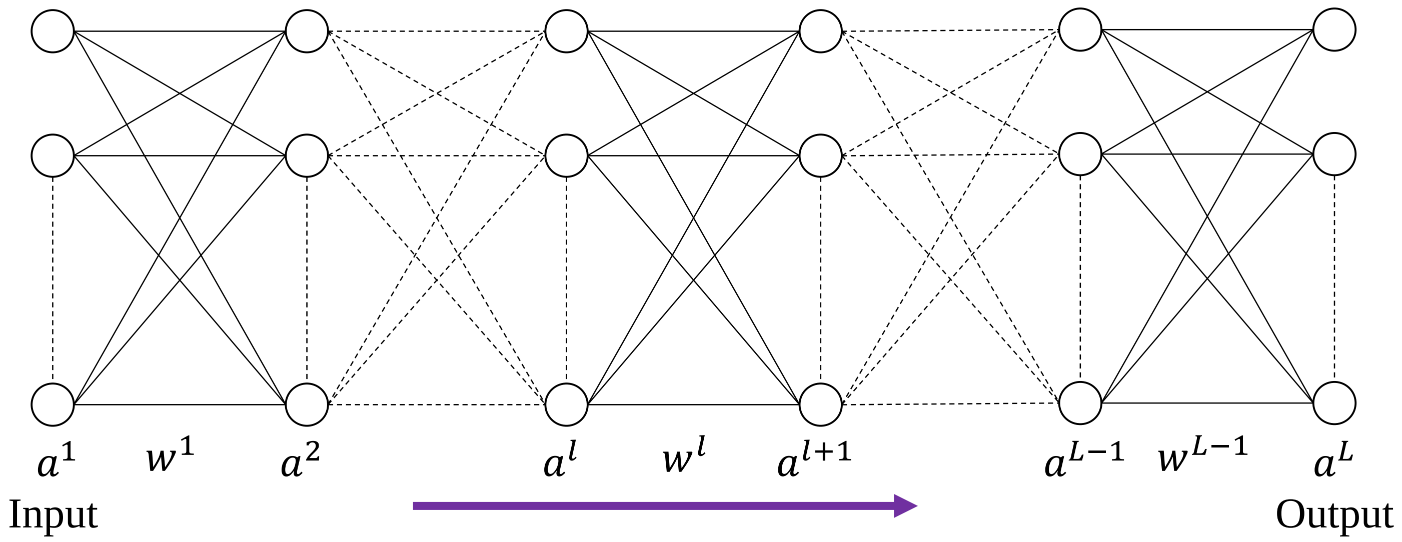


CNNs: Share Connections Weights Between Layers

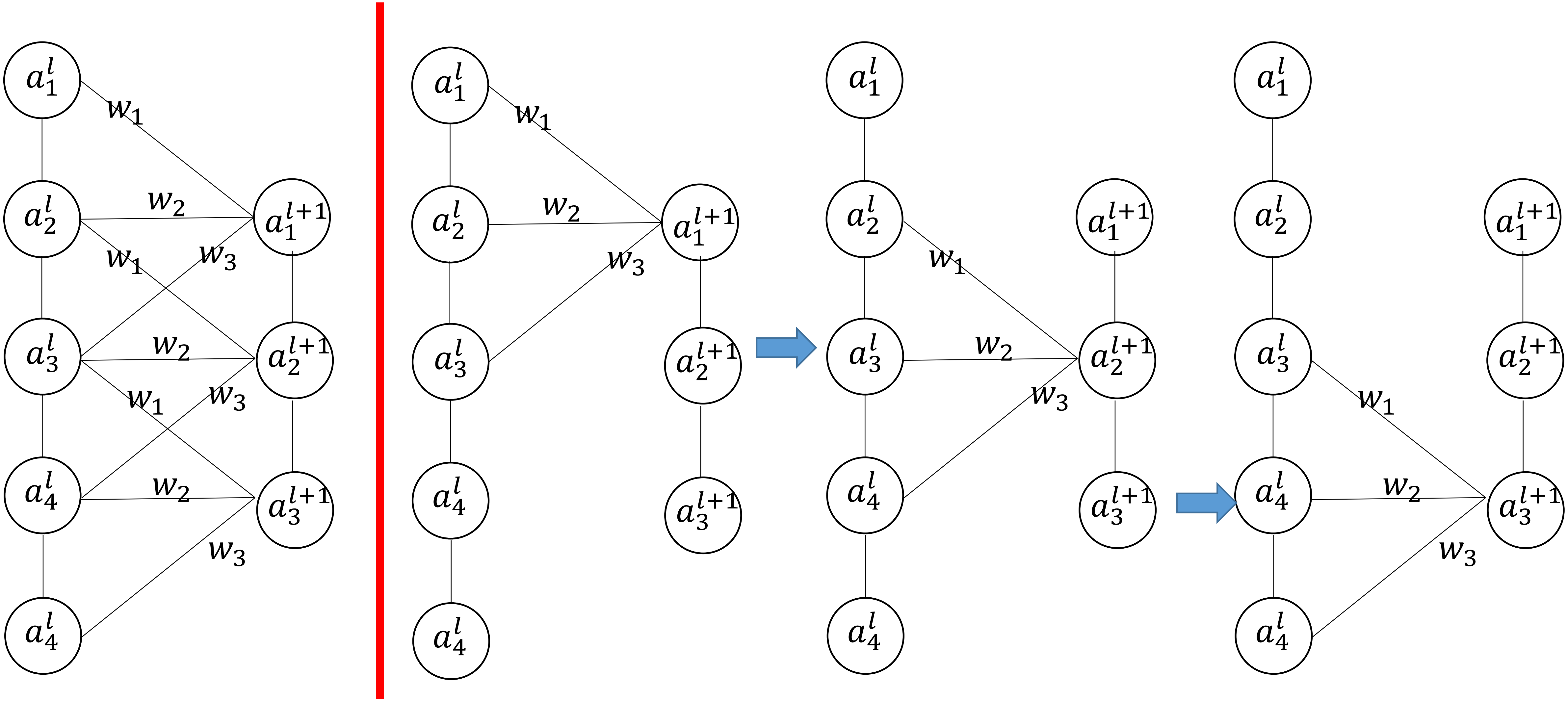


Convolutional Neural Networks

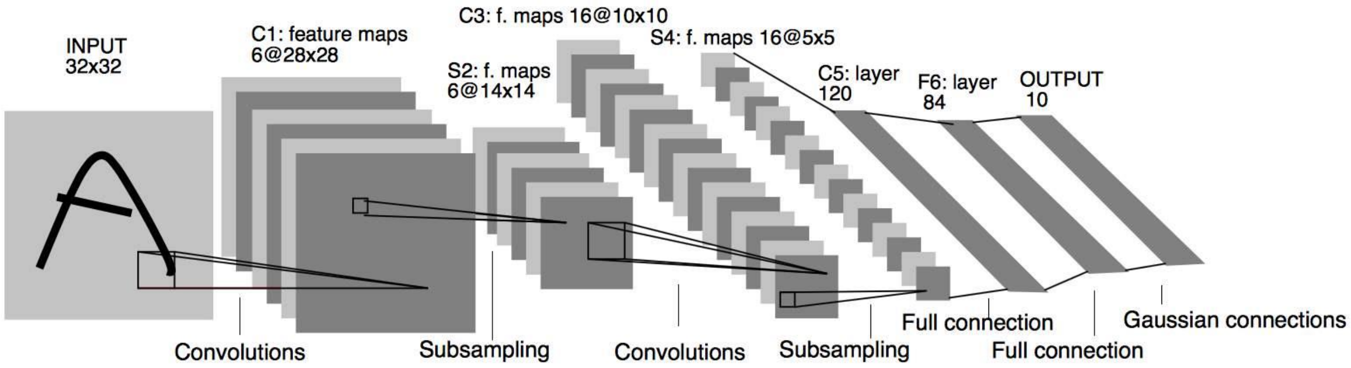
- Share Connection Weights Between Two Layers



CNNs: Share Connections Weights Between Layers



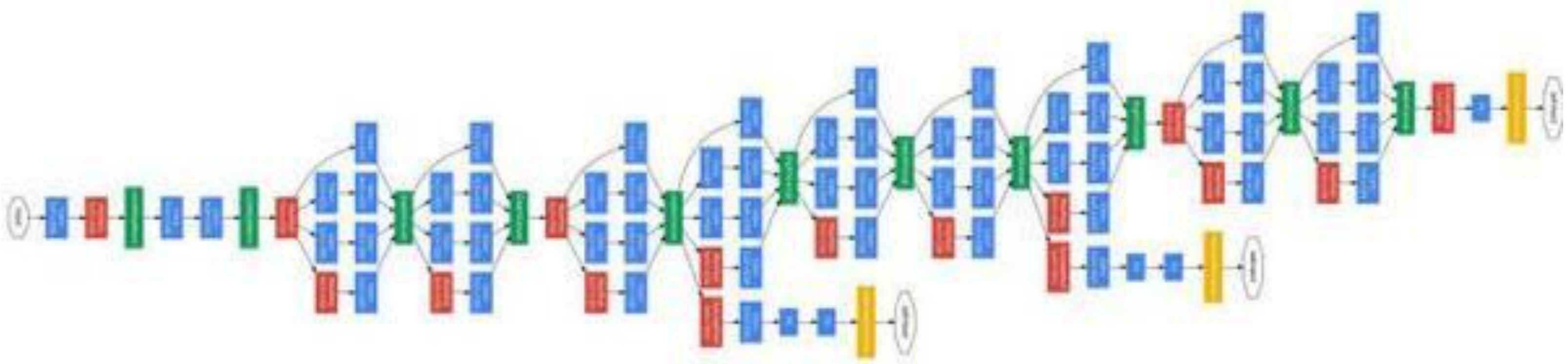
CNNs: Share Connections Weights Between Layers



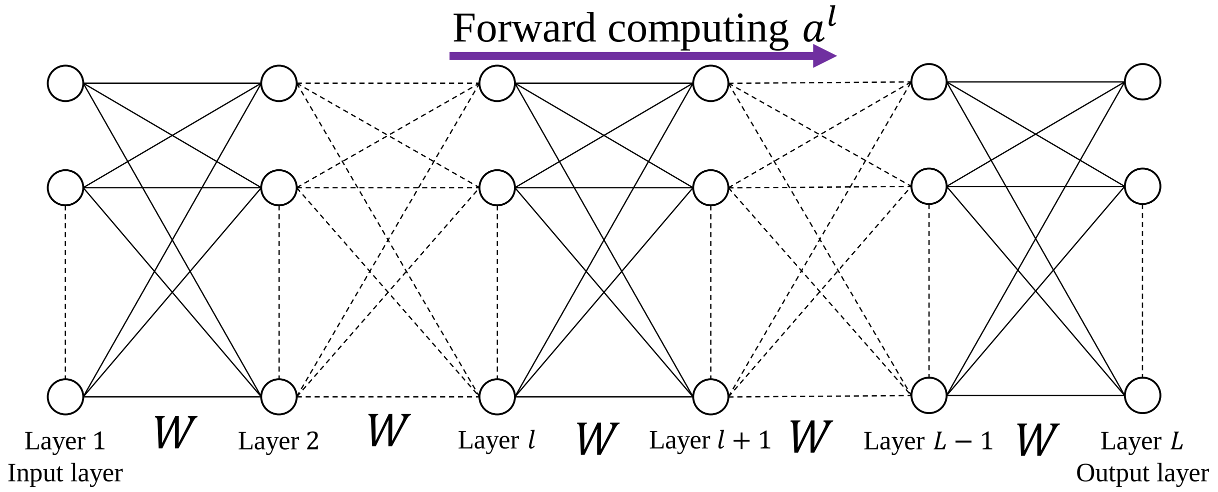
LeNet

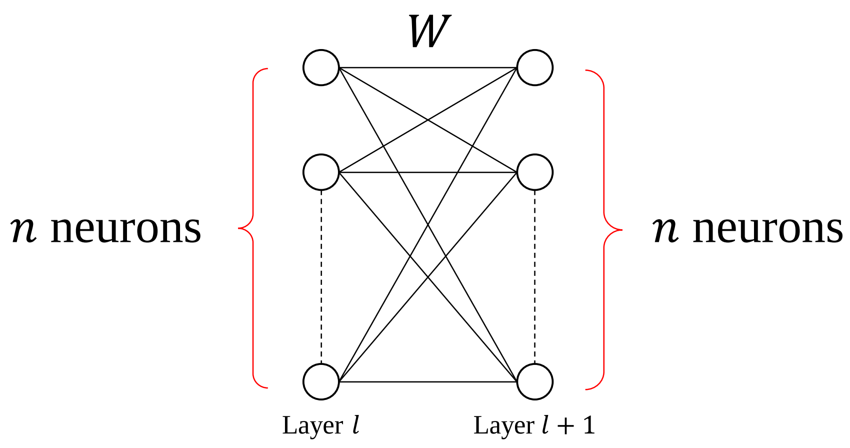
Yann LeCun

The Inception Architecture (GoogLeNet, 2014)



RNNs: Share connection weights in all layers



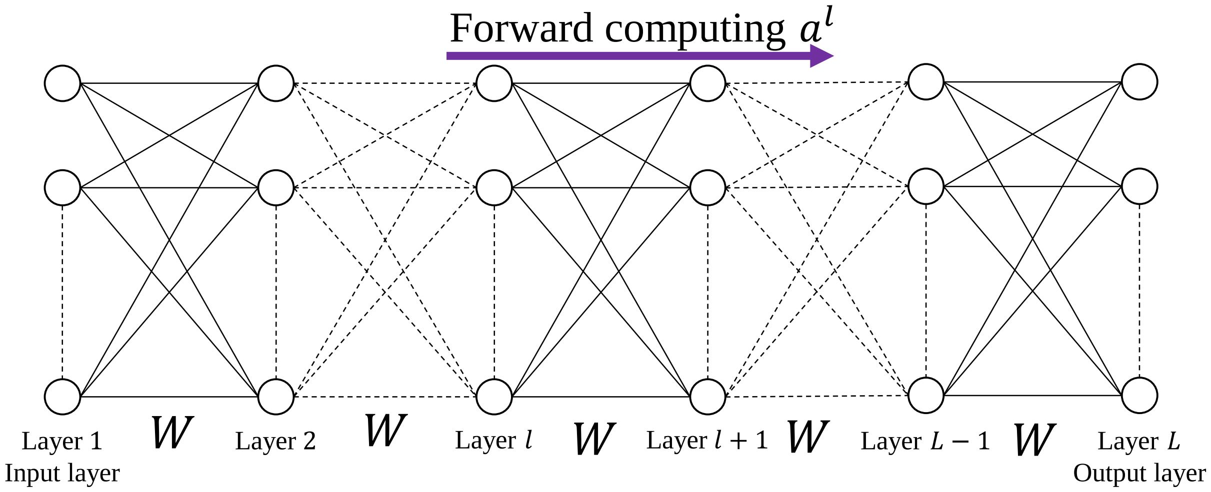


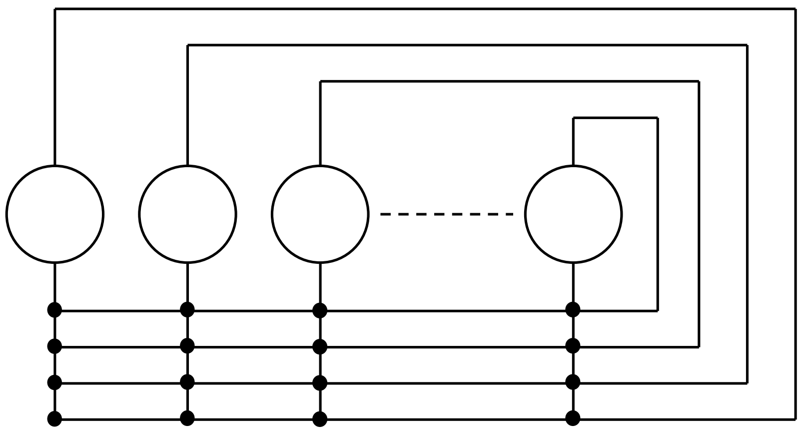
Recurrent Neural Networks

- $n_1 = n_2 = \cdots = n_L = n$
- $W^1 = W^2 = \cdots = W^L = W$

$$a_i^{l+1} = f\left(\sum_{j=1}^{n} w_{ij} a_j^l\right) \quad a^{l+1} = f(W a^l)$$

RNNs: Share connection weights in all layers





Recurrent Neural Networks

- $n_1 = n_2 = \cdots = n_L = n$
- $W^1 = W^2 = \cdots = W^L = W$

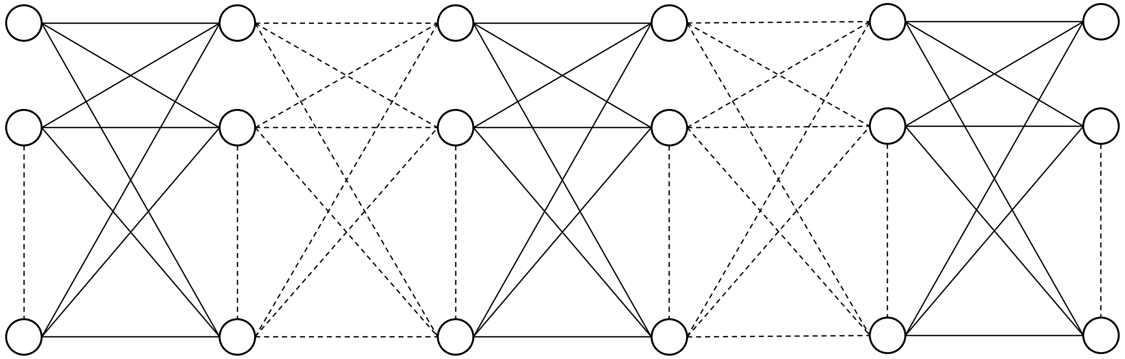
$$a^{l+1} = f\left(Wa^l\right) \xrightarrow{l \to t} a(t+1) = f\left(Wa(t)\right)$$

Discrete Time Neural Networks

CRNNs: Continuous recurrent neural networks

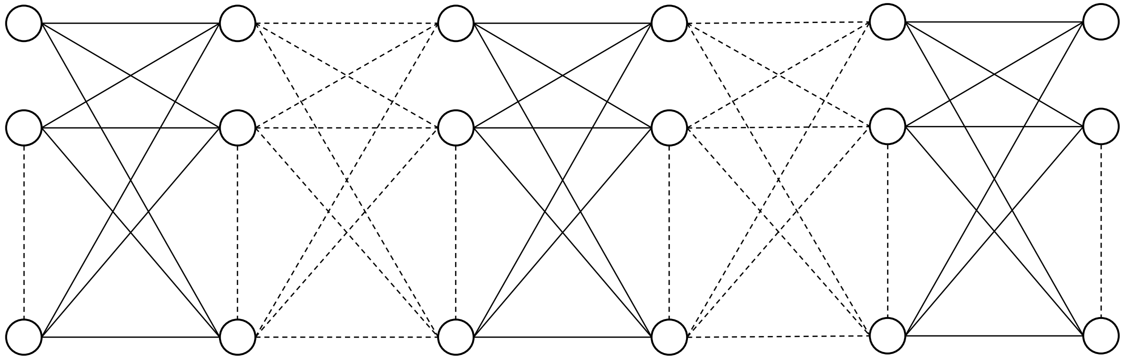
Problem:
How to develop model for
continuous time neural networks?

CRNNs: Continuous recurrent neural networks



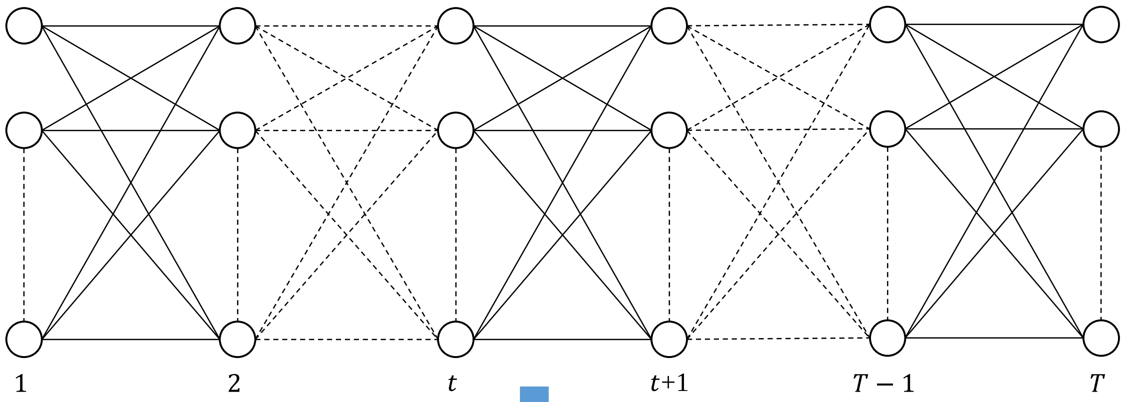
$$a_i^{l+1} = f\left(\sum_{j=1}^{n} w_{ij} a_j^l\right)$$

replace l by t replace a_i^l by $a_i(t)$

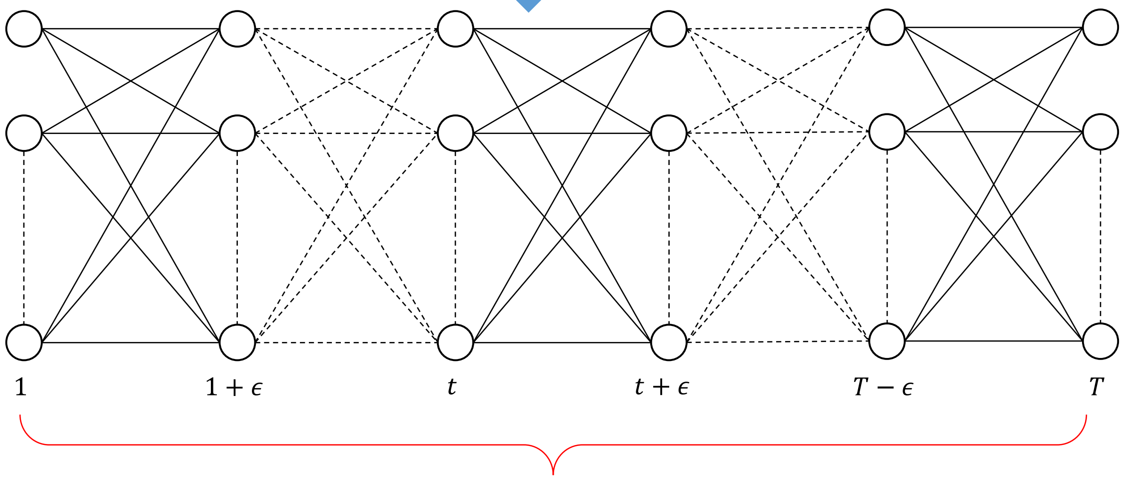


$$a_i(t+1) = f\left(\sum_{j=1}^{n} w_{ij} a_j(t)\right)$$

CRNNs: Continuous recurrent neural networks



Replace time step from 1 to ϵ



ϵ is an infinitesimal variable, thus, there are infinite layers

$$a_i(t+1) = f\left(\sum_{j=1}^{n} w_{ij} a_j(t)\right)$$

Starting from here: $a_i(t+1) = f\left(\sum_{j=1}^{n} w_{ij} a_j(t)\right)$

$$a_i(t+1) - a_i(t) = -a_i(t) + f\left(\sum_{j=1}^{n} w_{ij} a_j(t)\right)$$

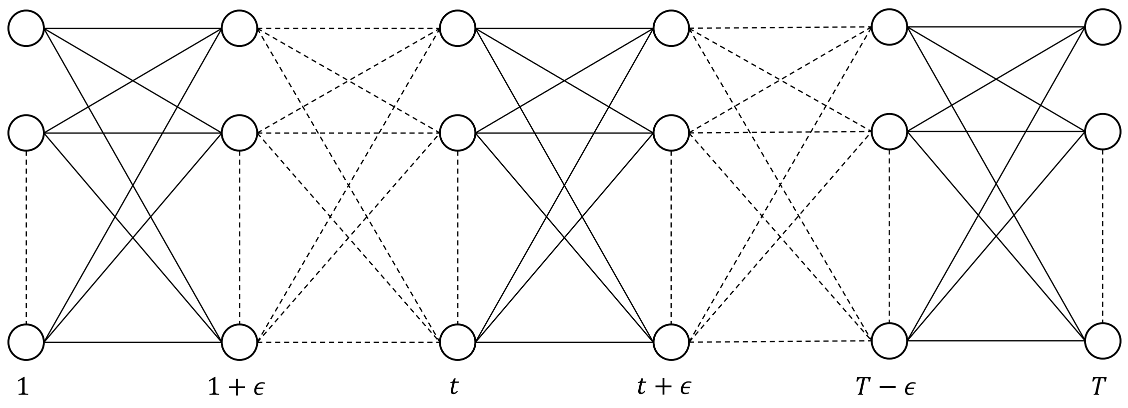
$$a_i(t+1) - a_i(t) = 1 \cdot \left[-a_i(t) + f\left(\sum_{j=1}^{n} w_{ij} a_j(t)\right)\right]$$

$$a_i(t+\epsilon) - a_i(t) = \epsilon \cdot \left[-a_i(t) + f\left(\sum_{j=1}^{n} w_{ij} a_j(t)\right)\right]$$

$$\frac{a_i(t+\epsilon) - a_i(t)}{\epsilon} = \left[-a_i(t) + f\left(\sum_{j=1}^{n} w_{ij} a_j(t)\right)\right]$$

$\epsilon \to 0$

$$\frac{da_i(t)}{dt} = -a_i(t) + f\left(\sum_{j=1}^{n} w_{ij} a_j(t)\right)$$

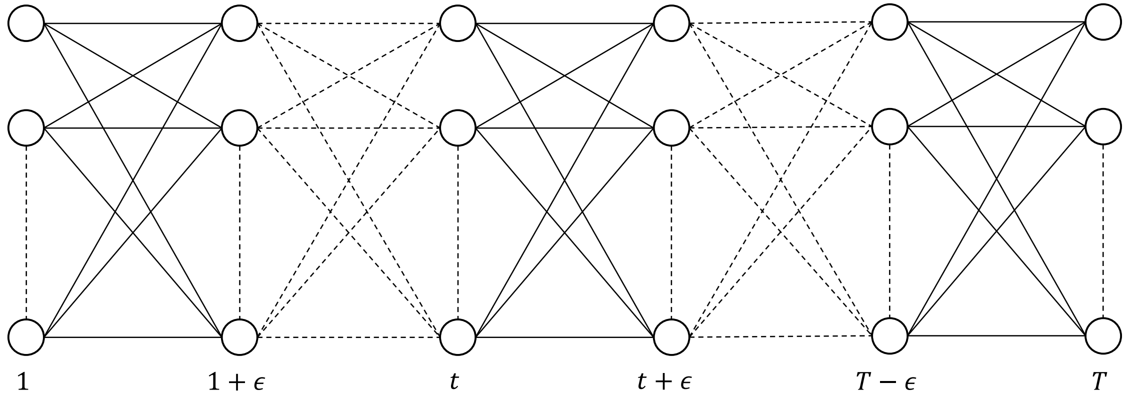


ϵ is an infinitesimal variable, thus, there are infinite layers

$\epsilon \to 0$

$$\frac{da_i(t)}{dt} = -a_i(t) + f\left(\sum_{j=1}^{n} w_{ij} a_j(t)\right)$$

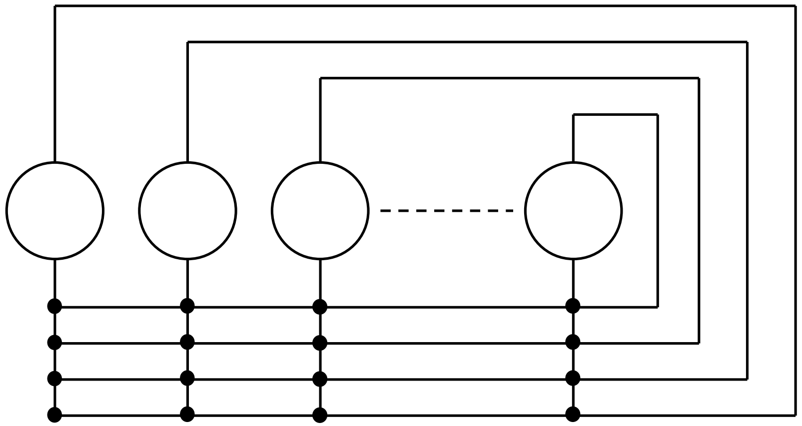
CRNNs: Continuous recurrent neural networks



ϵ is an infinitesimal variable, thus, there are infinite layers

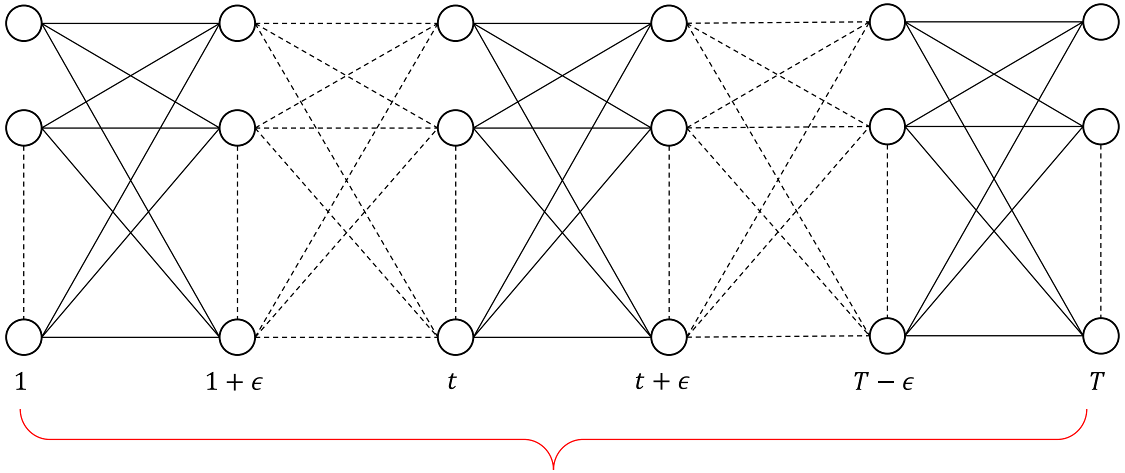
$\epsilon \rightarrow 0$

$$\frac{da_i(t)}{dt} = -a_i(t) + f\left(\sum_{j=1}^{n} w_{ij} a_j(t)\right)$$



CRNNs: Continuous recurrent neural networks

$$y = f\left(\sum_{i=1}^{n} w_i x_i\right)$$



$\epsilon = 1, t = l$

$$a_i^{l+1} = f\left(\sum_{j=1}^{n_l} w_{ij}^l a_j^l\right)$$

$\epsilon \to 0$

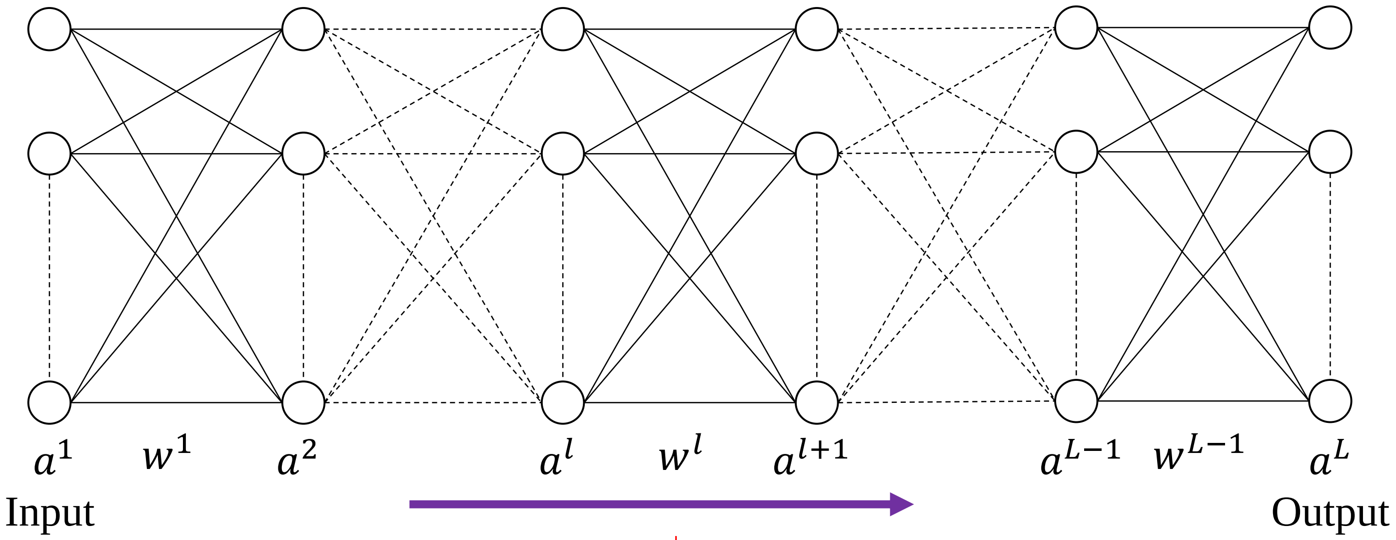
1. $n_1 = n_2 = \cdots = n_L = n$
2. $W^1 = W^2 = \cdots = W^L = W$

$$\frac{da_i(t)}{dt} = -a_i(t) + f\left(\sum_{j=1}^{n} w_{ij} a_j(t)\right)$$

Outline

- Brief Review of Brain Structure
- Computational Model of Neurons
- Computational Model of Neural Networks
- Various Models of Neural Networks
- Discussions
- Assignments

Nonlinear Mapping / Dynamical Systems



A neural network can be looked as a nonlinear mapping or a dynamical system.

$$a^L = f\left(W^{L-1} f\left(W^{L-2} f\left(W^{L-3} \cdots f\left(W^1 a^1\right)\right)\right)\right)$$

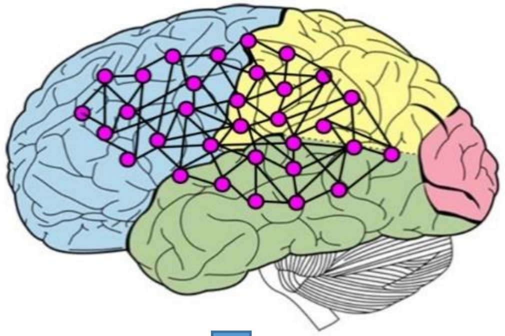
$$R^{n_1} \longrightarrow R^{n_L}$$

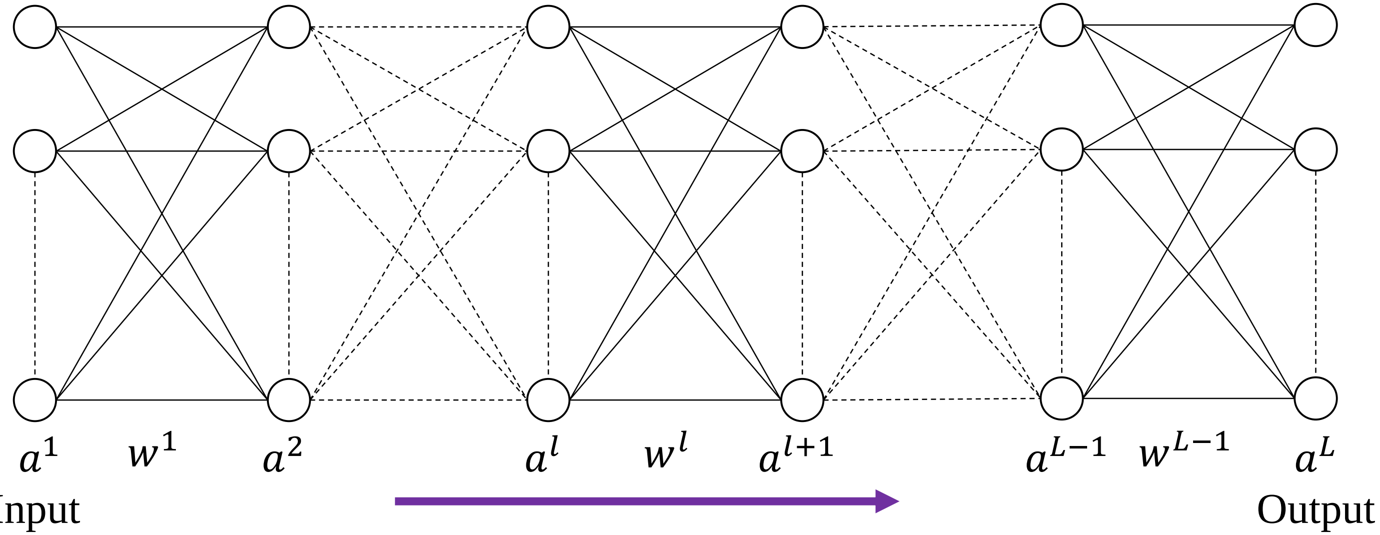
Nonlinear mapping

$$a_i^{l+1} = f\left(\sum_{j=1}^{n_l} w_{ij}^l a_j^l\right) \Rightarrow a_i(l+1) = f\left(\sum_{j=1}^{n_l} w_{ij}(l) a_j(l)\right)$$

Dynamical system

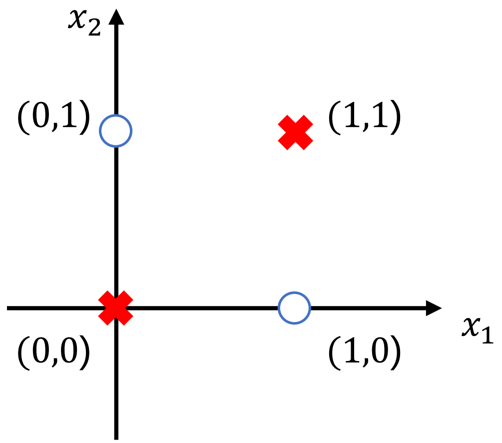
Nonlinear Mapping / Dynamical Systems





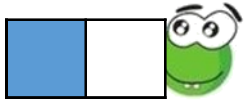
$$a_i^{l+1} = f\left(\sum_{j=1}^{n_l} w_{ij}^l a_j^l\right)$$

An example: XOR-worms problem



Doted worms

$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$

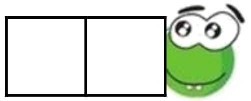


$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$

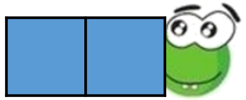
1

0

Smooth worms



$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$



$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$

An example: XOR-worms problem

$$\begin{bmatrix}0\\1\end{bmatrix} \quad \begin{bmatrix}1\\0\end{bmatrix}$$

$$\begin{bmatrix}0\\0\end{bmatrix} \quad \begin{bmatrix}1\\1\end{bmatrix}$$

$$F\left(\begin{bmatrix}x_1\\x_2\end{bmatrix}\right) = f[f(2x_1 + 2x_2 - 1) + f(-x_1 - x_2 + 1.5) - 1.5]$$

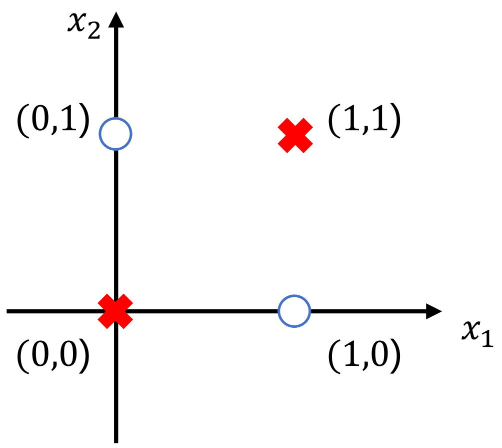
$$f(s) = \begin{cases}1, & s \geq 0\\ 0, & otherwise\end{cases}$$

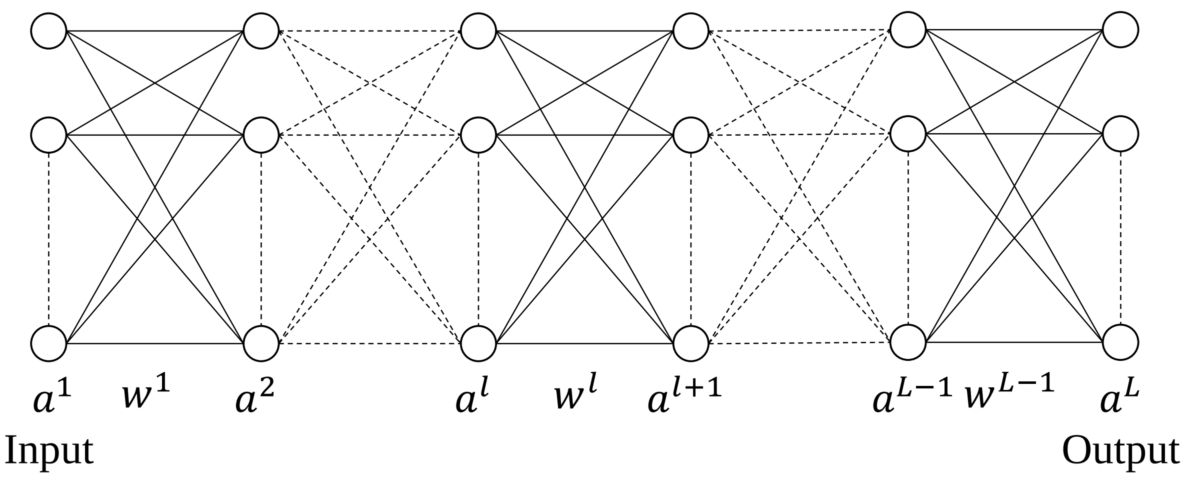
$\rightarrow$ $\boxed{1}$

$\rightarrow$ $\boxed{0}$

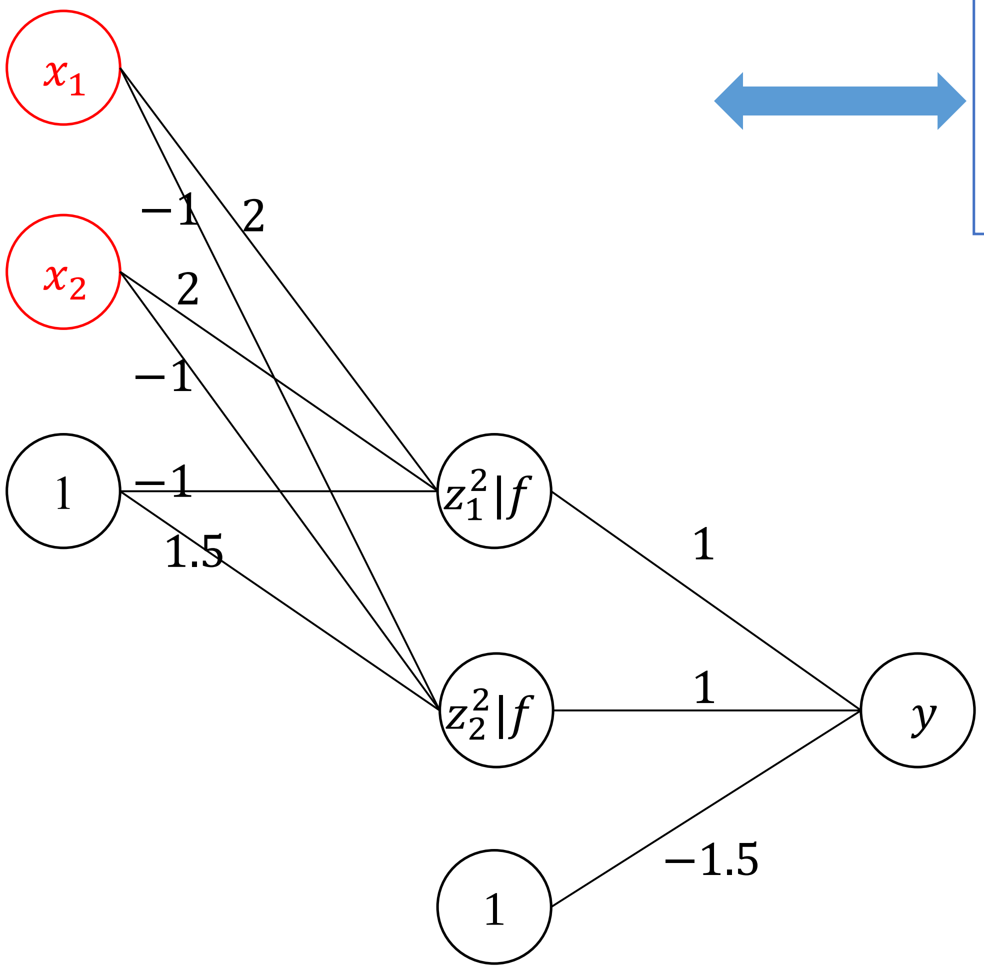
An FNN is a nonlinear mapping.

Problem: Can we construct an FNN to replace *F*?



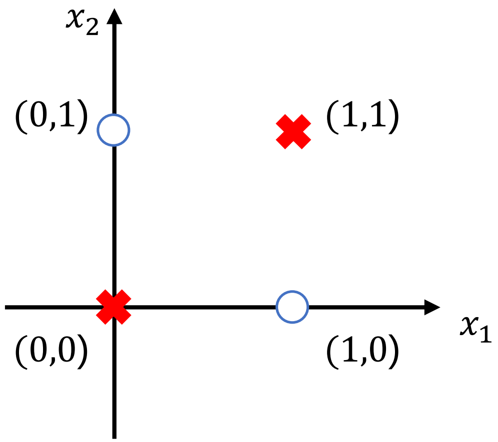


An example: XOR-worms problem



$$F\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = f[f(2x_1 + 2x_2 - 1) + f(-x_1 - x_2 + 1.5) - 1.5]$$

$$f(s) = \begin{cases} 1, & s \geq 0 \\ 0, & otherwise \end{cases}$$



$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

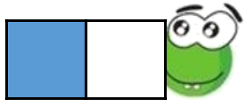
$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Can this network really replace F?
Check it!

An example: XOR-worms problem

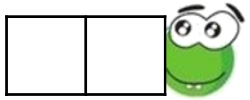
Doted worms

$\begin{bmatrix}0\\1\end{bmatrix}$

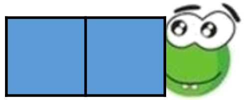


$\begin{bmatrix}1\\0\end{bmatrix}$

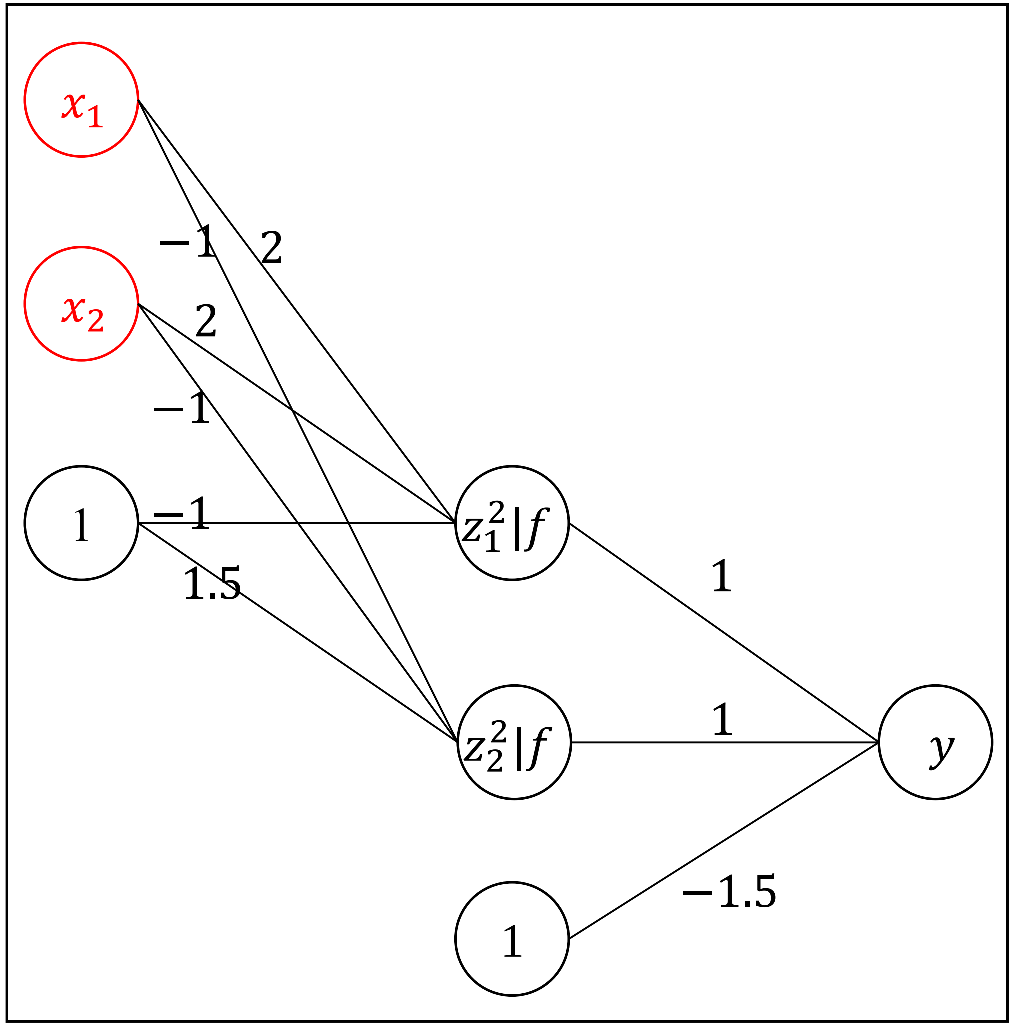
Smooth worms



$\begin{bmatrix}0\\0\end{bmatrix}$



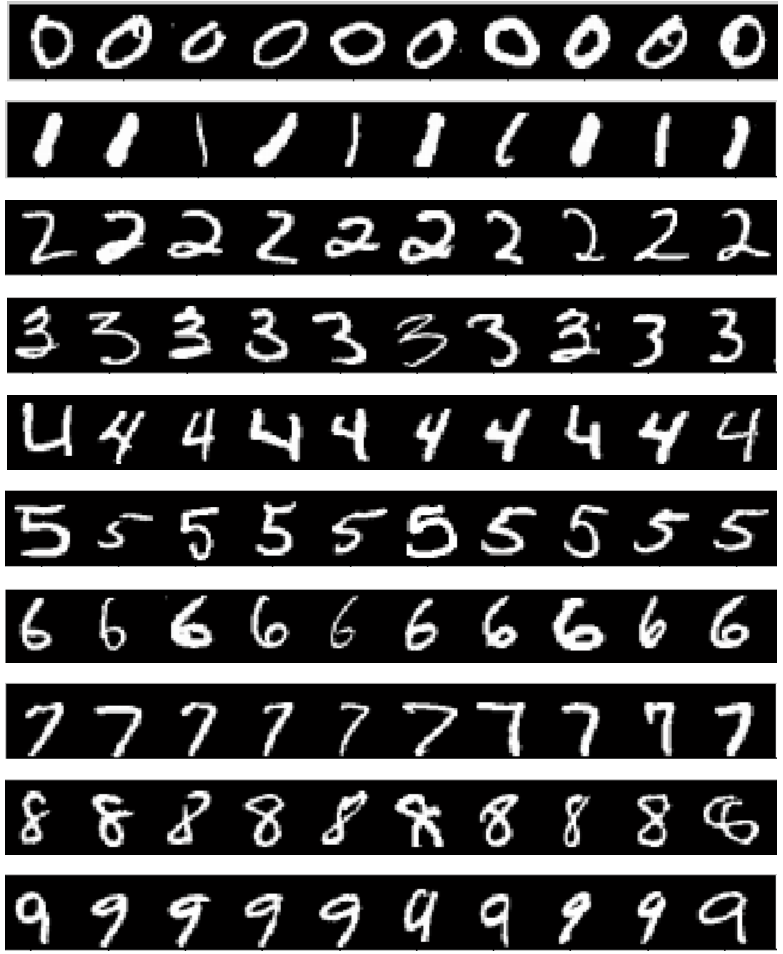
$\begin{bmatrix}1\\1\end{bmatrix}$



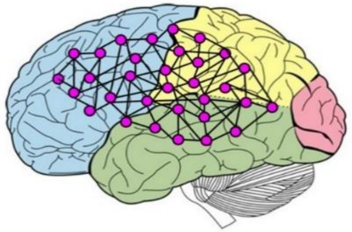
1

0

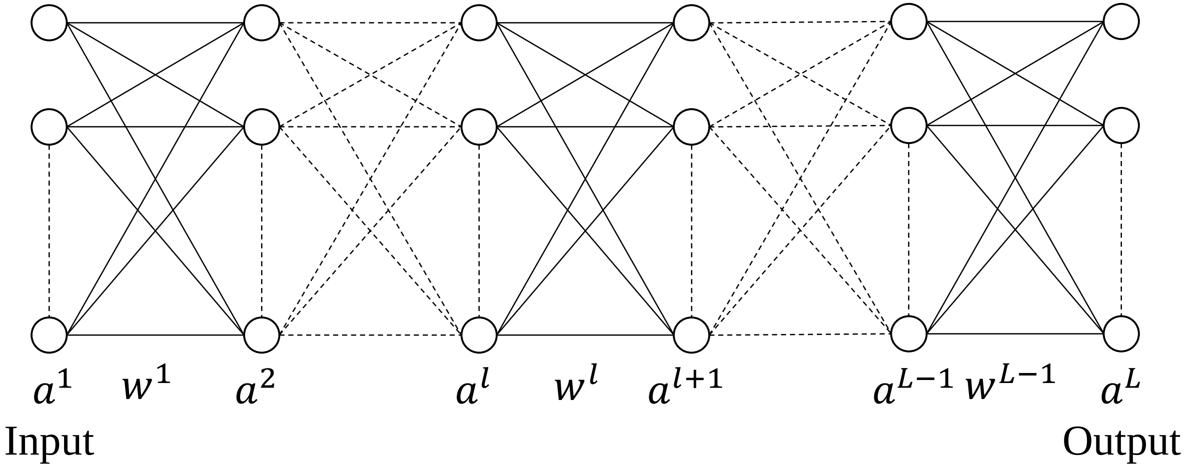
Handwritten Digits Recognition



digitizing → $F: x \to y$ → representation → 0 1 2 3 4 5 6 7 8 9



Problem: Can we construct an FNN to replace *F*?

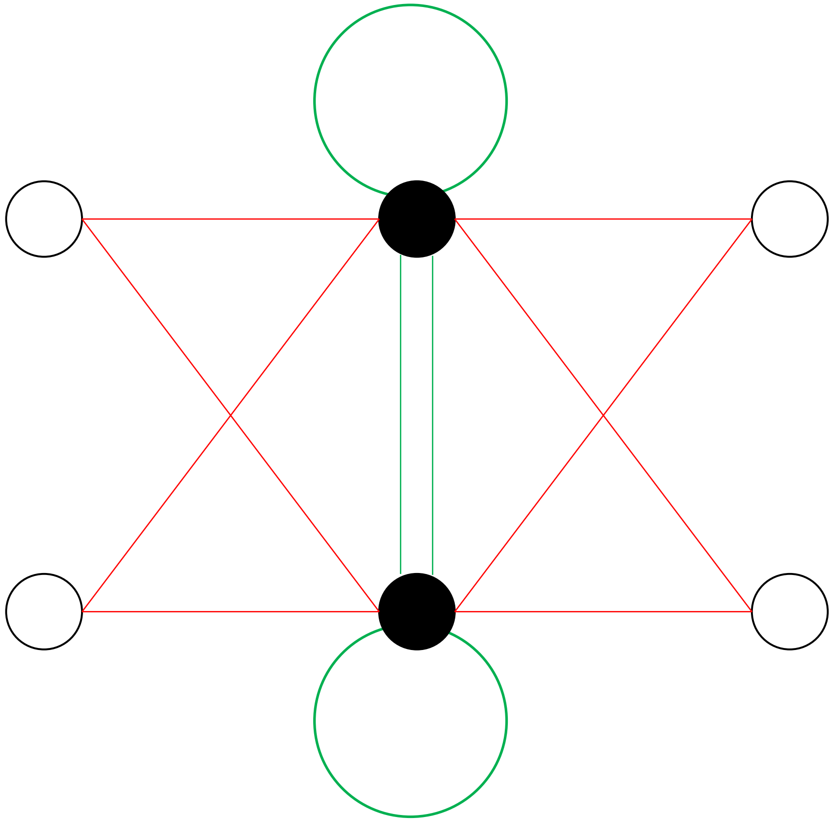


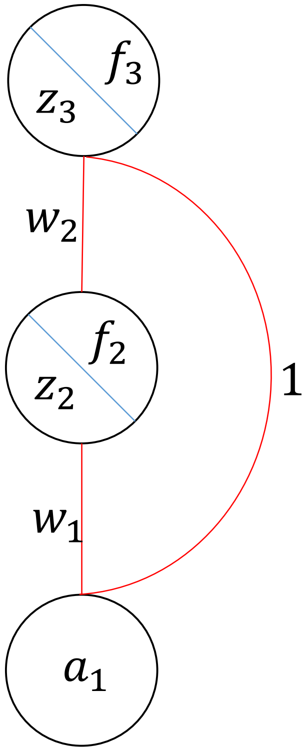
Outline

- Brief Review of Brain Structure
- Computational Model of Neurons
- Computational Model of Neural Networks
- Various Models of Neural Networks
- Discussions
- Assignments

Assignment

Redraw the following two networks to be in standard form, i.e., no any connection in any layer, no connection across any layer.





Assignment

- Implement the forward computing of this NN:
  - in component form
  - in vector form

Algorithm in Component form:

$Input\ W^l, a^1$
$for\ l = 1{:}\,L$
$\qquad a^{l+1} = fc\_c(W^l, a^l)$
$return$

Function $fc\_c(W^l, a^l)$
$for\ i = 1{:}\,n_{l+1}$

$$z_i^{l+1} = \sum_{j=1}^{n_l} w_{ij}^l a_j^l$$

$$a_i^{l+1} = f(z_i^{l+1})$$

end

Algorithm in Vector form:

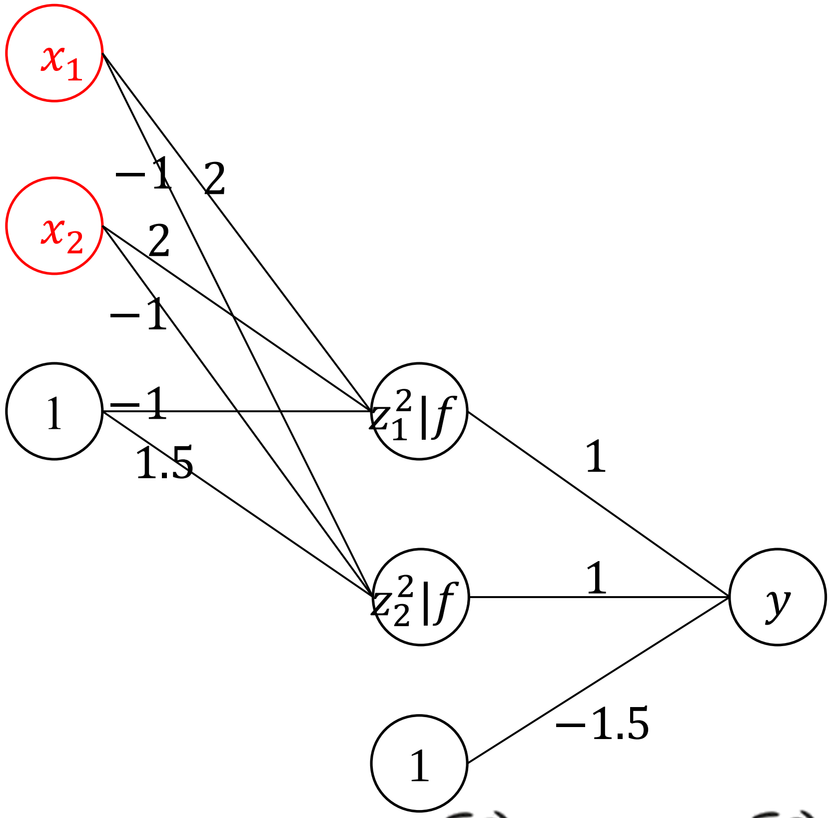
$Input\ W^l, a^1$
$for\ l = 1{:}\,L$
$\qquad a^{l+1} = fc\_v(W^l, a^l)$
$return$

Function $fc\_v(W^l, a^l)$

$$z^{l+1} = W^l a^l$$

$$a^{l+1} = f(z^{l+1})$$

end



Doted worms

$\begin{bmatrix}0\\1\end{bmatrix}$ $\begin{bmatrix}1\\0\end{bmatrix}$

Smooth worms

$\begin{bmatrix}0\\0\end{bmatrix}$ $\begin{bmatrix}1\\1\end{bmatrix}$

Thanks