AI fundamentals & curriculums:
AI 인재양성을 위한 필수교육

Changho Suh
School of EE
Graduate School of AI (by courtesy)
KAIST

September 9, 2020
Edtech Korea Forum

Three keywords in AI

1. Machine learning

2. Deep learning

3. Tensorflow

Machine learning

collection of **instructions** given to computers

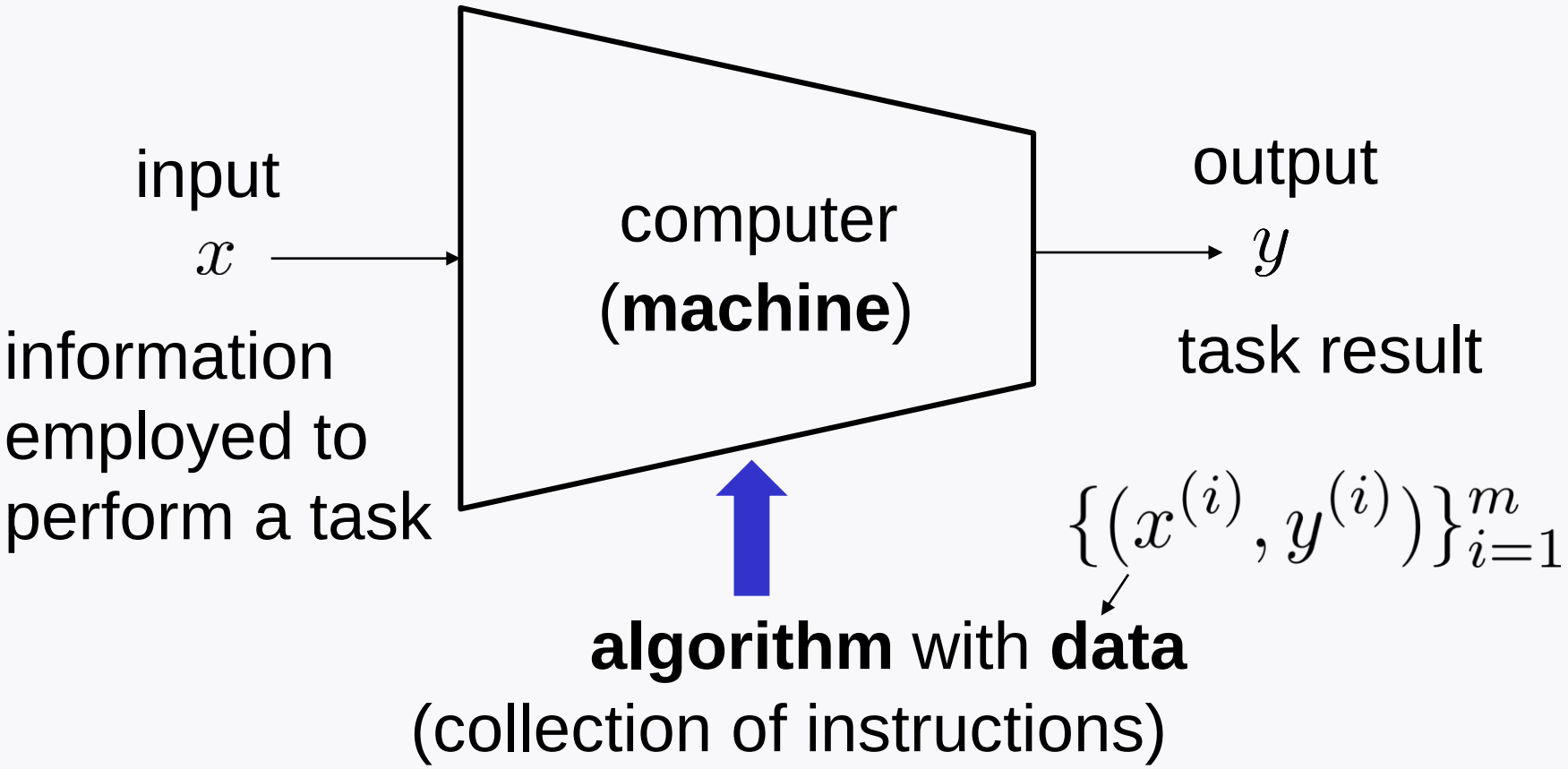
↓

A field about *algorithm*

Why give instructions to computers?

So that computers can do a specific task of interest like human beings.

Machine learning in picture

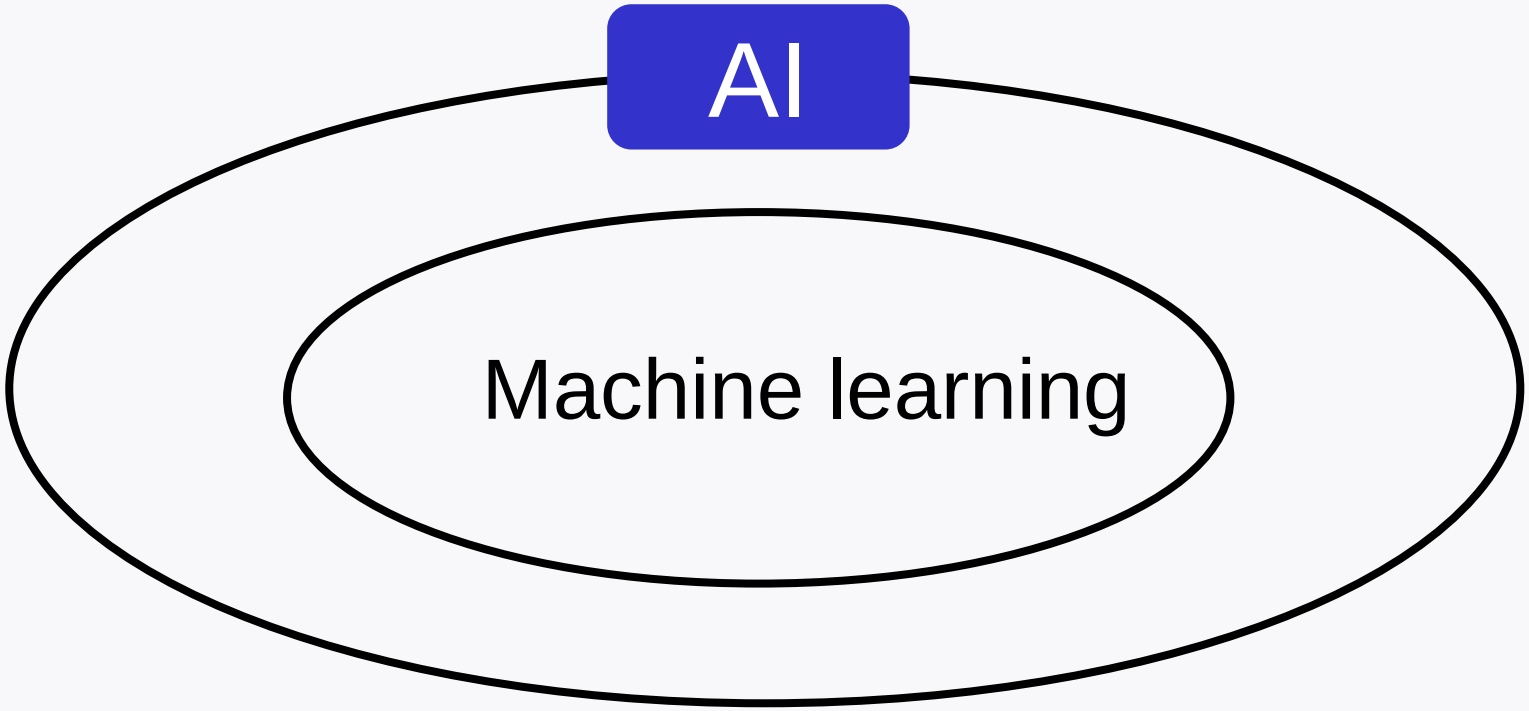


Note: Trained machine should be like *human beings*.

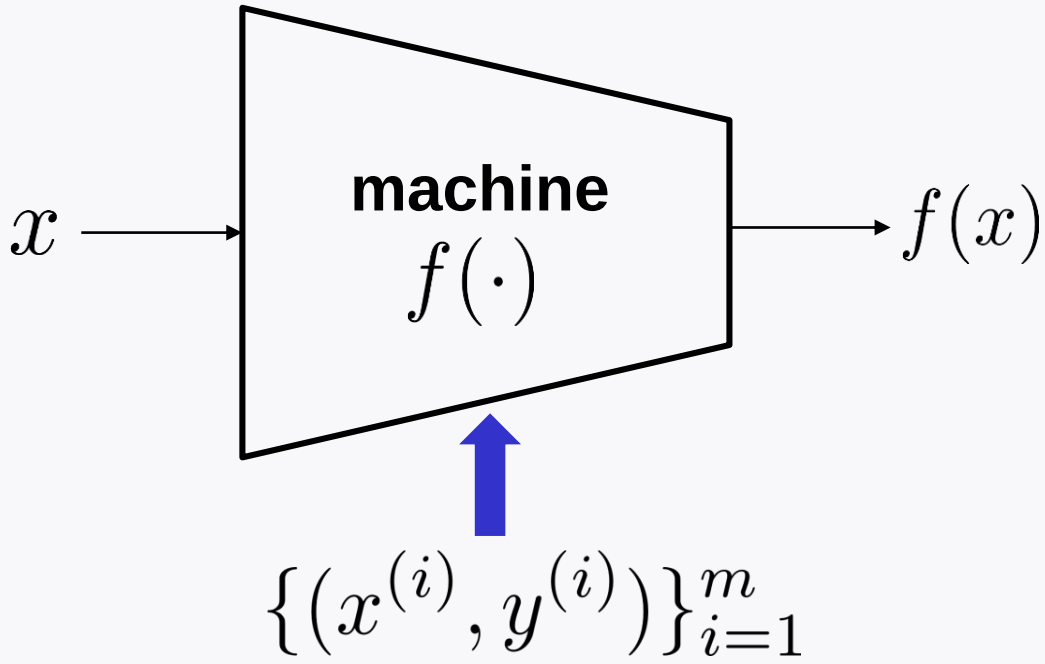
Machine learning vs AI

Mission of machine learning:

Creating “**Artificial Intelligence**”!



How to train machine?



Through **optimization**!

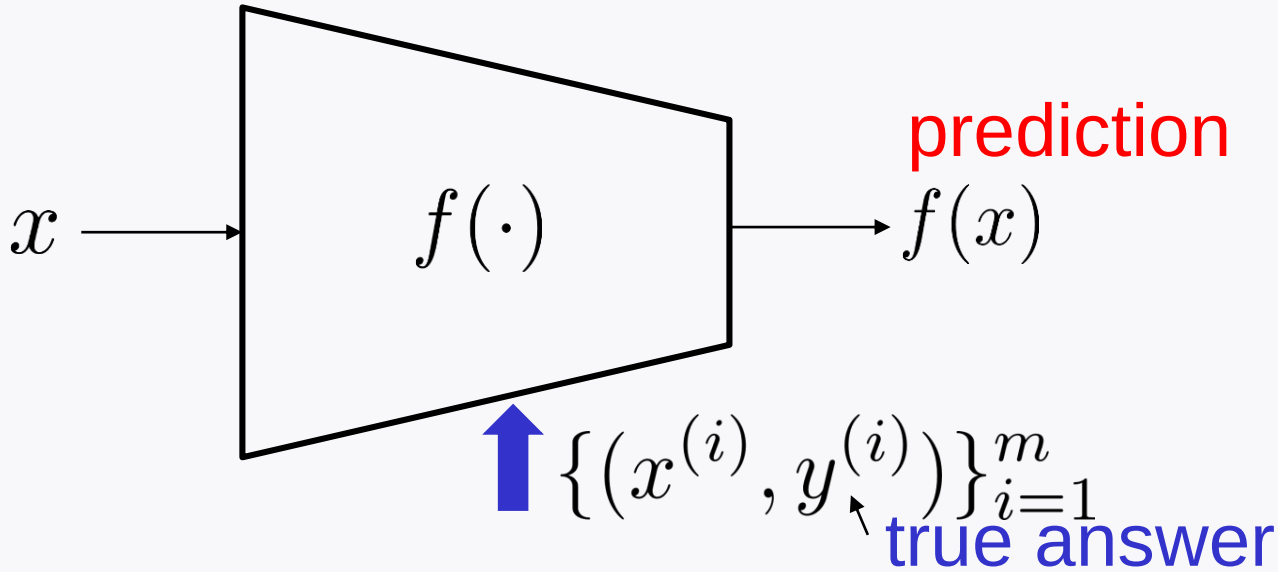
Optimization

Two concepts required to define optimization

1. A certain quantity of interest: **objective function**
   (a real number)
   scalar
2. **Optimization variable** (collection of real numbers)
   vector

Definition: Finding an **optimization variable** that minimizes (or maximizes) the **objective function**.

Objective function?



What we want: $f(x^{(i)}) \approx y^{(i)}$ for all i

prediction true answer

How to **quantify closeness**?

One way is to employ a loss function: $\ell(y^{(i)}, f(x^{(i)}))$

Example: $\ell(y, \hat{y}) = (y - \hat{y})^2$

Optimization variable?

$$\min_{f(\cdot)} \sum_{i=1}^{m} \ell(y^{(i)}, f(x^{(i)}))$$

What affects the objective function is:

Prediction function $f(\cdot)$

Challenge: There are *so many* choices for function.

How to deal with function optimization?

$$\min_{f_w(\cdot)} \sum_{i=1}^{m} \ell(y^{(i)}, f_w(x^{(i)}))$$

A common way:

Specify a **function class** (e.g., linear, quadratic …)

Represent $f(\cdot)$ with parameters w

Consider the parameters as optimization variable.

How to choose function class?

$$\min_{w} \sum_{i=1}^{m} \ell(y^{(i)}, f_w(x^{(i)}))$$

One architecture was suggested:

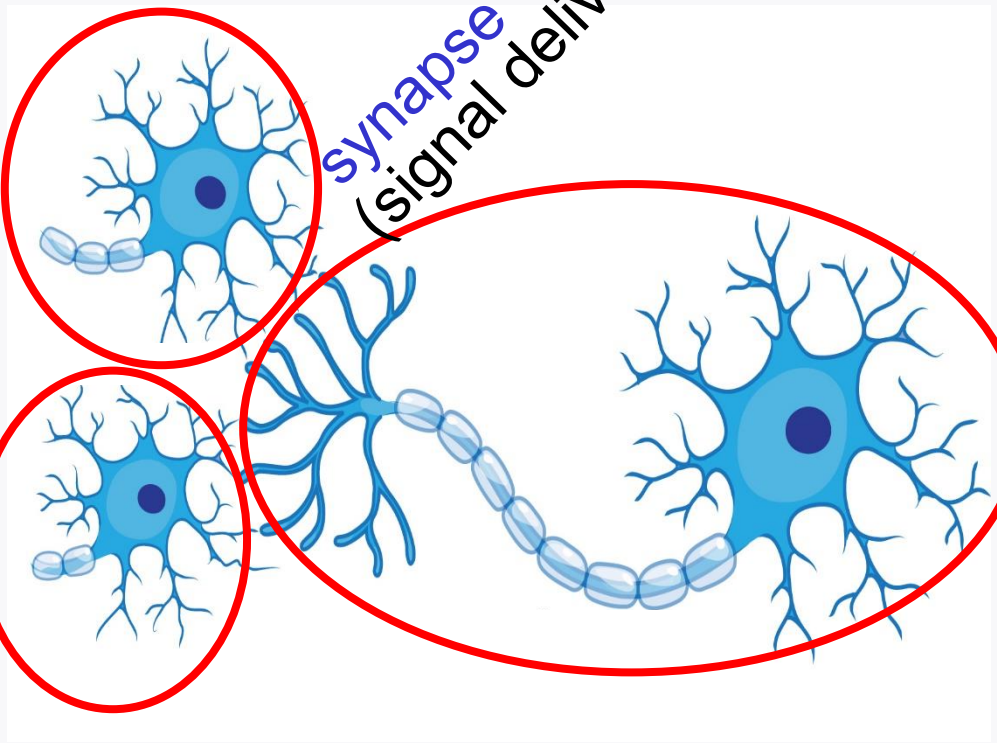
Perceptron

Frank Rosenblatt ‘57
(psychologist)

Perceptron

Inspired by: Brains of intelligent beings.

Inside brains: There are many electrically excitable cells, called neurons.

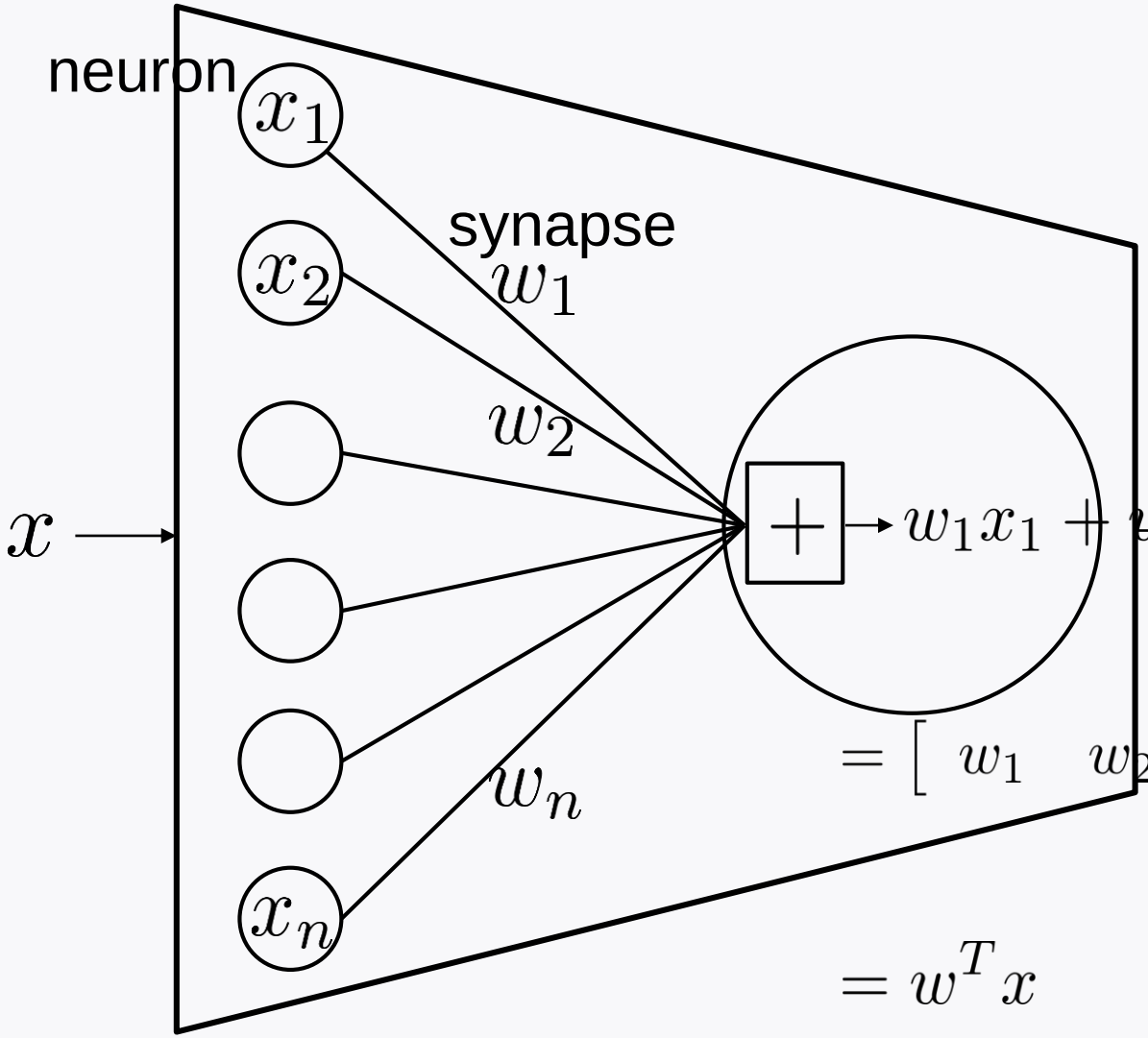


activation

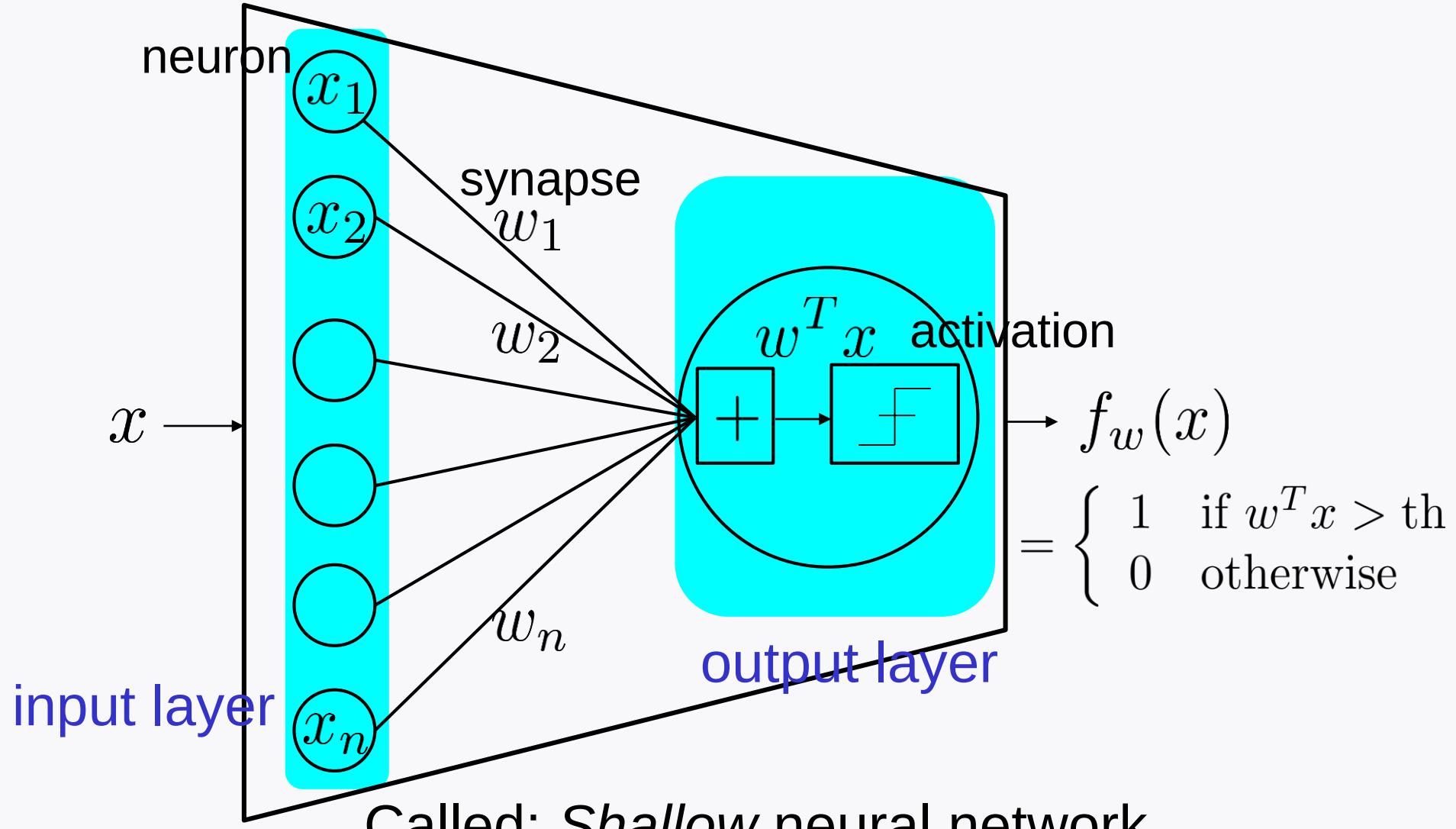
voltage $\uparrow \implies 1$

$\downarrow \implies 0$

Perceptron


neuron
x_1
x_2
x_n
synapse
w_1
w_2
w_n
x
+
$w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$
$= \begin{bmatrix} w_1 & w_2 & \cdots & w_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$
$= w^T x$

Perceptron



Called: *Shallow* neural network

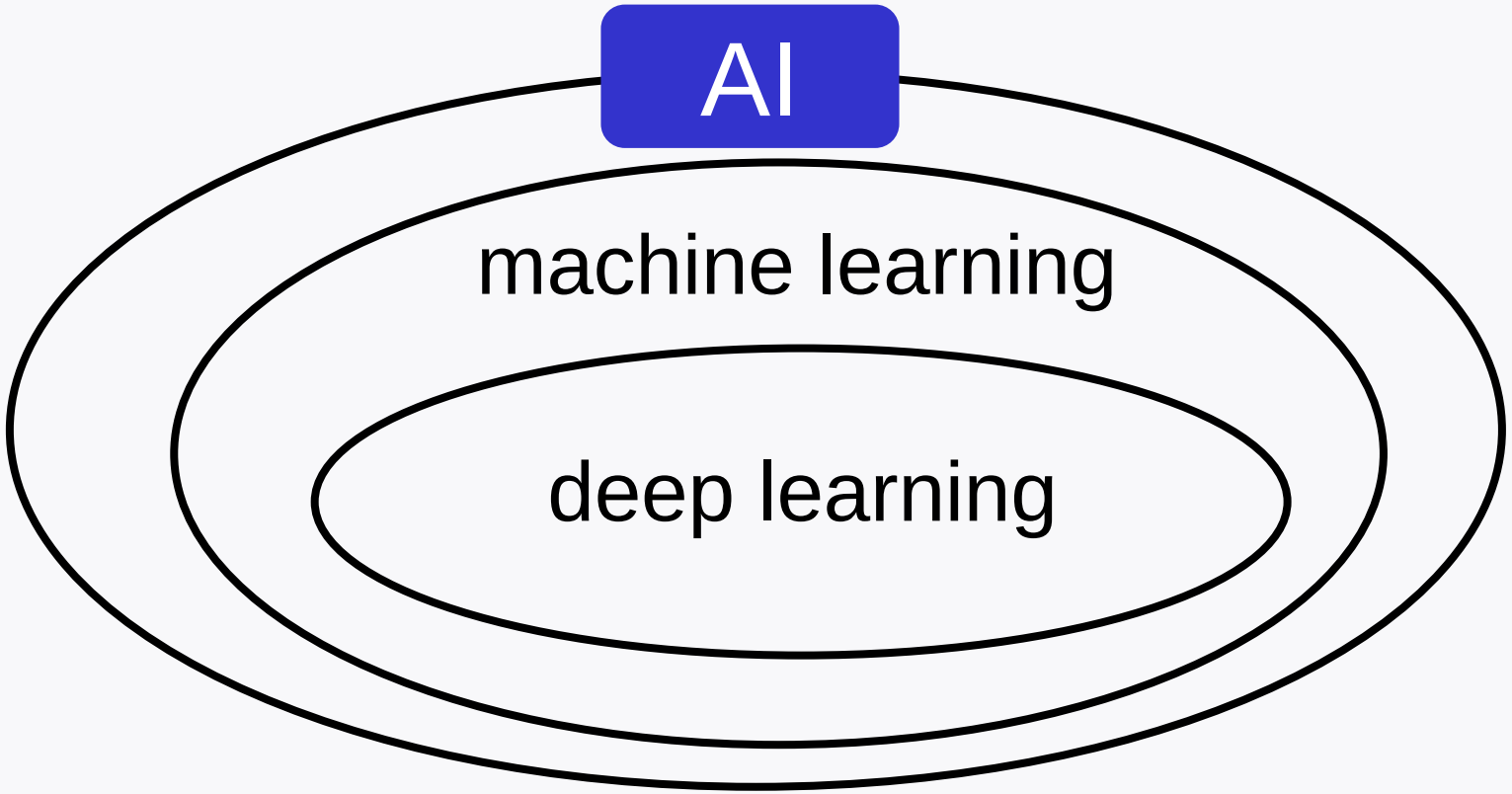
Multi-layer neural networks

A neural network may contain layers between input/output layers.

Such network is called: **Deep neural networks (DNNs)**

Deep learning: DNN-based machine learning

AI vs machine learning vs deep learning



Turns out: Machines trained via deep learning can achieve *human-level performances!*

How to solve deep learning-based opt?

$$\min_{w} \sum_{i=1}^{m} \ell(y^{(i)}, f_w(x^{(i)}))$$

Function class $f_w(x)$: Deep neural network

Rely on **computer programming.**

One popular programming tool for deep learning:

Tensorflow

Three fundamentals for AI

	Current curriculums
1. Optimization	from Middle school 3rd grade
2. Matrix	from High school 2nd grade (yet not entirely)
3. Programming	from Middle school 1st grade (yet since 2018)

Propose new AI curriculums

1. Optimization

   from Middle school 1st grade

   Can teach in the context of *quadratic* functions

2. Matrix

   from Middle school 1st grade

   Can teach *independently from* other contents

3. Programming

   from Middle school 1st grade

   Focus on easy language like **Python**

   Example-driven teaching