LANKA NIPPON BIZTECH INSTITUTE

2024/3/4-2024/3/5

AI Basics

Kazuaki Tanaka

Kyushu Institute of Technology

About Me

- Kazuaki TANAKA（たなか かずあき）
- Kyushu Institute of Technology, JAPAN
  Associate Professor
- Ph. D in Information Science

- Research topics: embedded systems, IoT, mruby, wireless communication

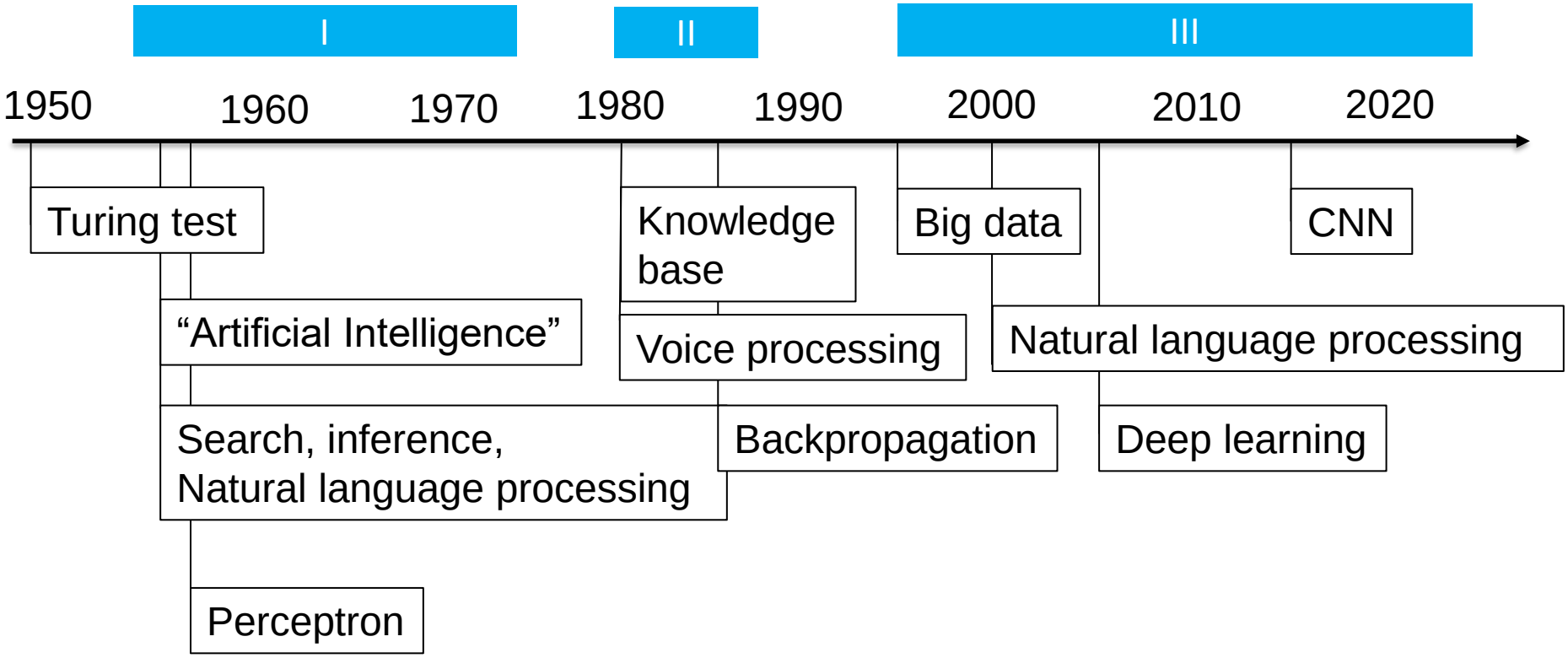

Kyushu Island
Kazakhstan
Mongolia
Kyrgyzstan
Turkmenistan
China
Sea of Okhotsk
Sea of Japan
South Korea
Japan
Syria
Iraq
Iran
Afghanistan
Pakistan
Nepal
India
Myanmar (Burma)
Thailand
Vietnam
Philippines
Philippine Sea
East China Sea
Saudi Arabia
Oman
Yemen
Gulf of Aden
Arabian Sea
Bay of Bengal
Gulf of Thailand
Laccadive Sea
Ethiopia
Somalia
Kenya
Malaysia
Indonesia
Banda Sea
Arafura Sea
Papua New Guinea
Mozambique

Our Campus
Kyushu Institute of Technology

Contents

- What does AI do ?
- Neural Networks
  - Perceptron
  - Basic AI Behavior
  - Learning
  - Some Issues
- Application Fields of AI
- Demo

History of AI



What does AI do?

What AI can do

- Learning and Decision Making
  - Learn data (this data means XXX)
  - Determine data based on the learned information (this data is XXX)

- Model
  - Decide (or more correctly, classification) from input and get output result

Model

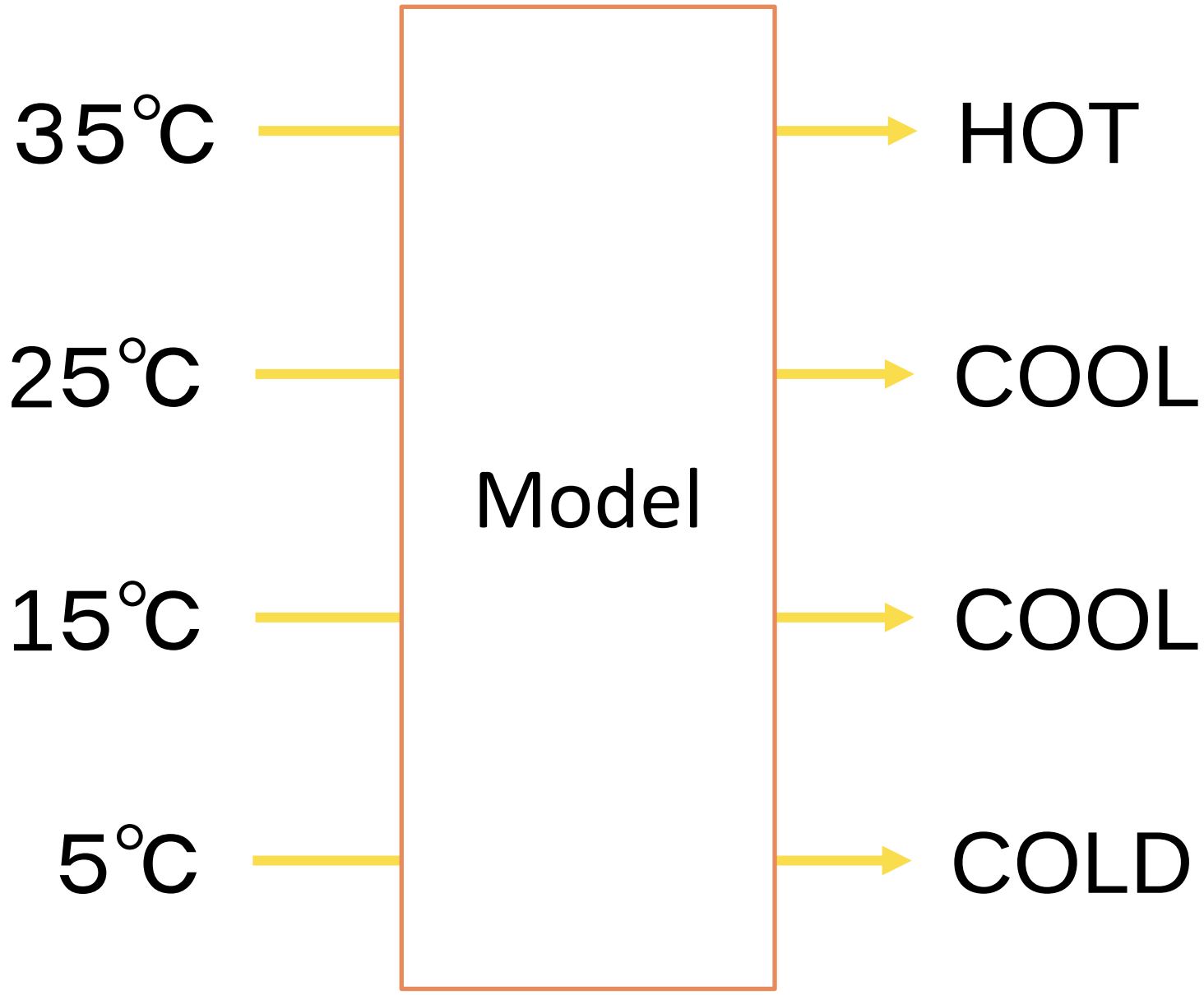
CAT

CAT

FLOWER

Simple case

- Temperature and our feeling
  - HOT / COOL / COLD


35℃
25℃
15℃
5℃
Model
HOT
COOL
COOL
COLD

Training data (teaching data)

Temp.	Feeling
35	HOT
30	HOT
29	HOT
27	HOT
24	COOL
20	COOL
10	COOL
8	COLD
0	COLD
-4	COLD

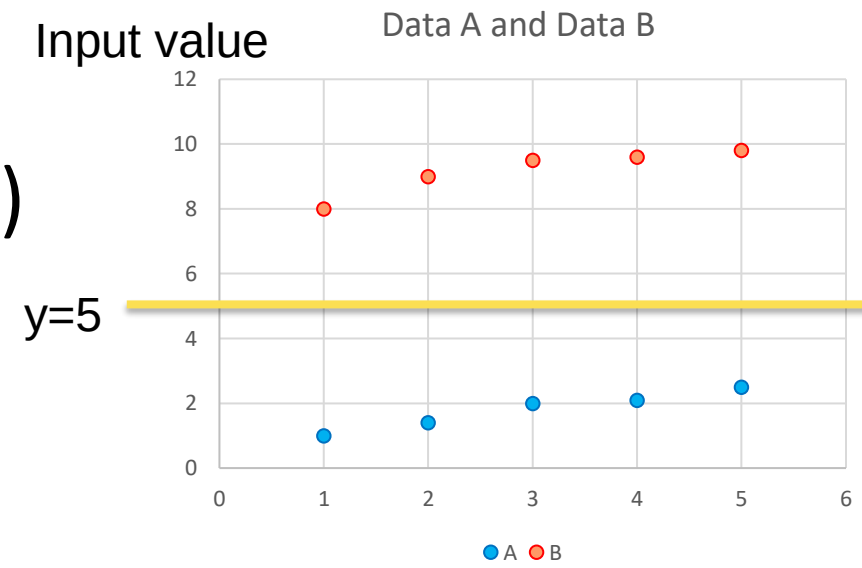
How to generate?

Model

Example

- I want to classify data A and data B
  - Want to draw a line: y=5

- Learning
  - Finding this line
- Determine (Classification)
  - Classify the data to Data A or Data B



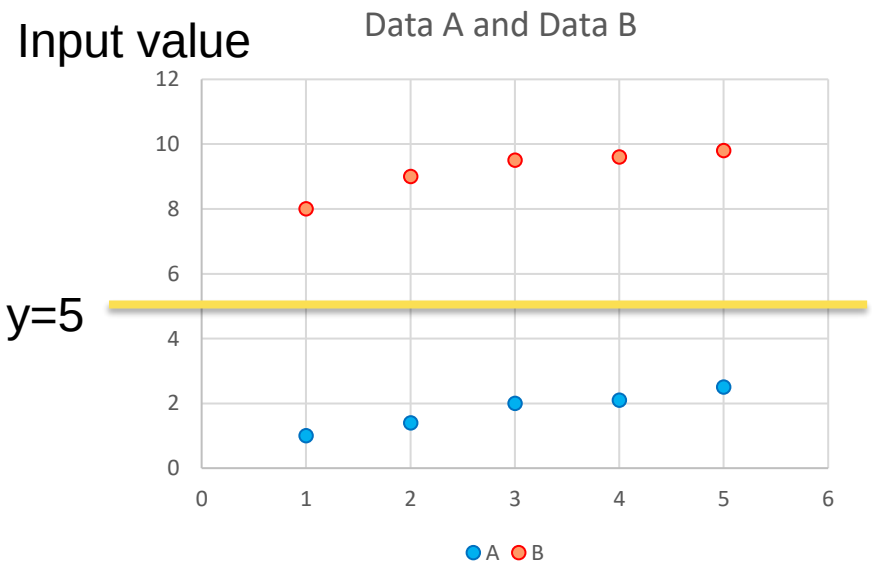
Example

- Classification
- Once getting line y=5, we can classify new data

New data 7.5 → Data B

New data 2.2 → Data A

New data -3.1 → Data A



How to find this line?

- Easy if there is only one input (one dimension)
  - Considering a straight-line $y = \alpha$
  - Try various α to find a line that looks good

- If the distance from data A and data B are equal, it is the appropriate line.
- A line is drawn in the middle of data A and data B

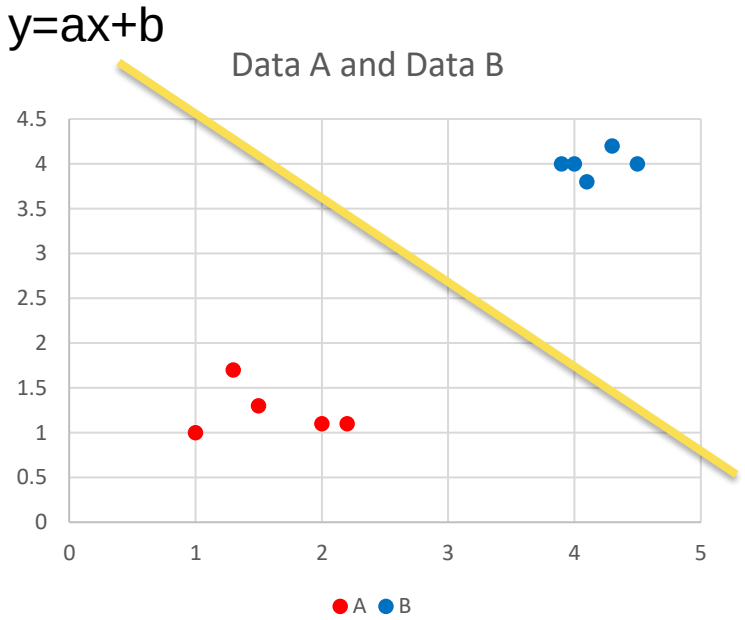
Dimension

- This example: Data A and Data B
  - one dimension (only one input)
- Expand dimension,
  - Temperature and Humidity => weather forecast
  - 1,000 dimension (sound intensity in time series)
    => Speech recognition
  - 10,000 dimension (100x100 image)
    => OCR
  - 1,000,000 dimensions (1,000x1,000 image)
    => Image recognition

Basic AI Processing

Example of a complicated case

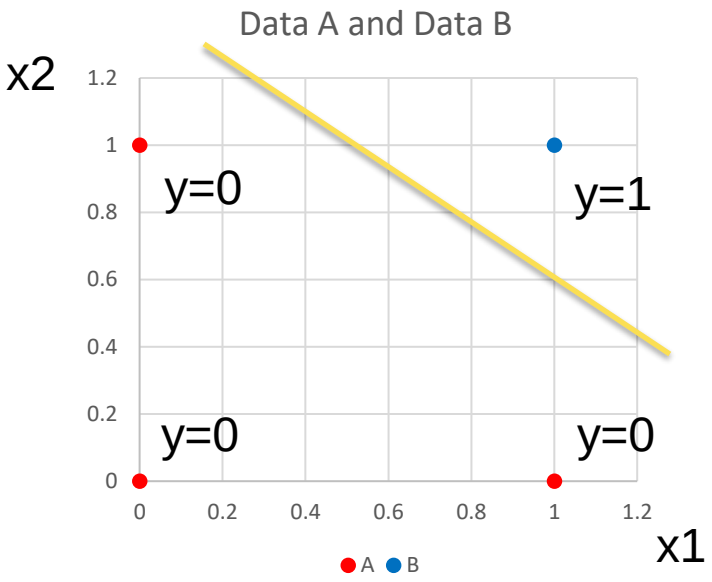
- Two inputs (x,y)
- Learning
  - Find $y = ax + b$
- Classification
  - y-(ax+b) is positive or negative
  - Above or below the yellow line
  - Negative: A, Positive: B



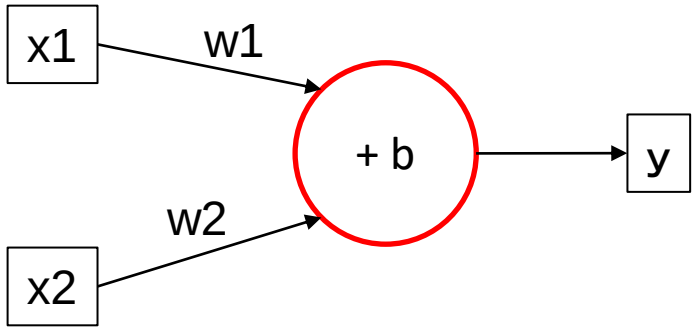
Perceptron

- Mathematical approach
- Input： (x1, x2)
- Output： y

- (0,0) → 0
- (0,1) → 0
- (1,0) → 0
- (1,1) → 1



Weight (w) and Bias (b)



$$y = \begin{cases} 0 & (w1x1 + w2x2 + b \leq 0) \\ 1 & (w1x2 + w2x2 + b > 0) \end{cases}$$

Find values for *w1,w2,b*

One answer: *w1*=1, *w2*=1, *b*=-1

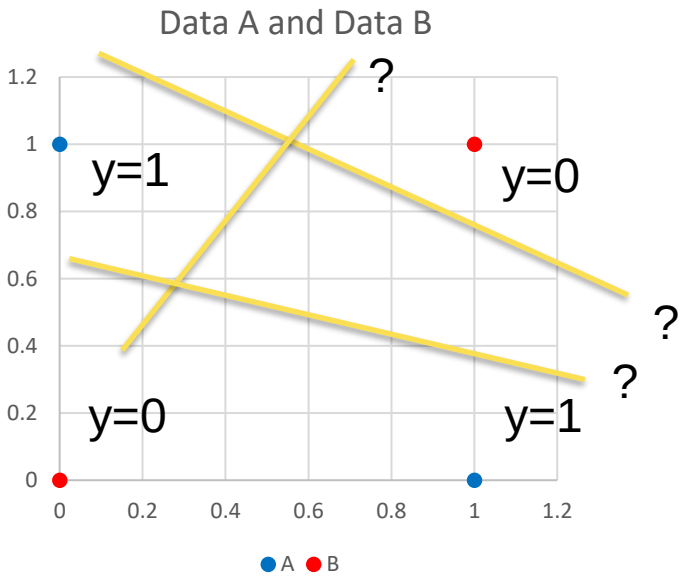
Answer

- Let *w1*=1, *w2*=1, *b*=-1 for $w1x1 + w2x2 + b$

(x1,x2)	$x1 + x2 - 1$		
(0,0)	-1	≤ 0	A (y=0)
(0,1)	0	≤ 0	
(1,0)	0	≤ 0	
(1,1)	1	> 0	B (y=1)

Another case:

- Cannot be solved
  - (0,0) → 0
  - (0,1) → 1
  - (1,0) → 1
  - (1,1) → 0



- We cannot draw a line for separation

FYI: AND, OR, NOT, XOR

- (0,0) → 0
- (0,1) → 0
- (1,0) → 0
- (1,1) → 1

AND

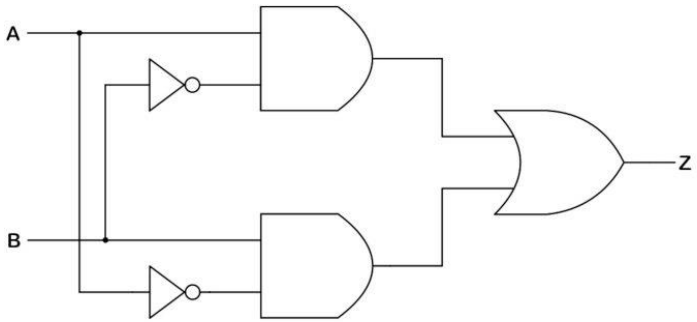
AND is possible
And also, OR and NOT are possible

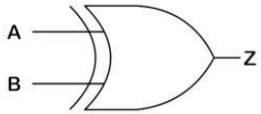
XOR is impossible

- (0,0) → 0
- (0,1) → 1
- (1,0) → 1
- (1,1) → 0

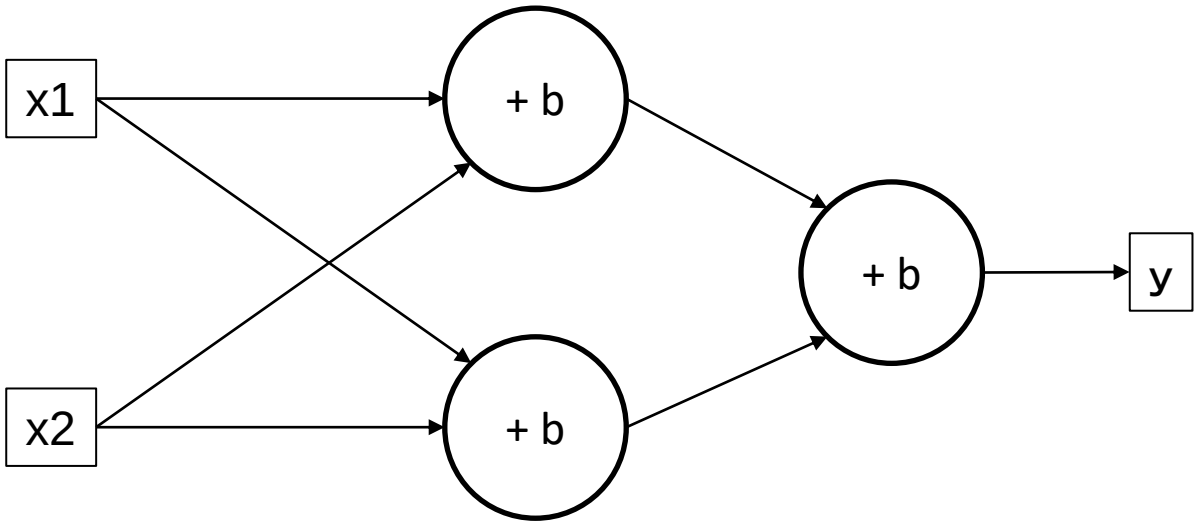
XOR

However, XOR can be realized by combination with AND, OR, NOT



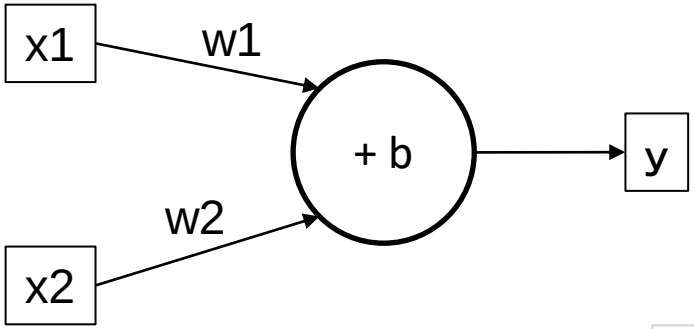


Multilayer perceptron



Neural Networks (NN)

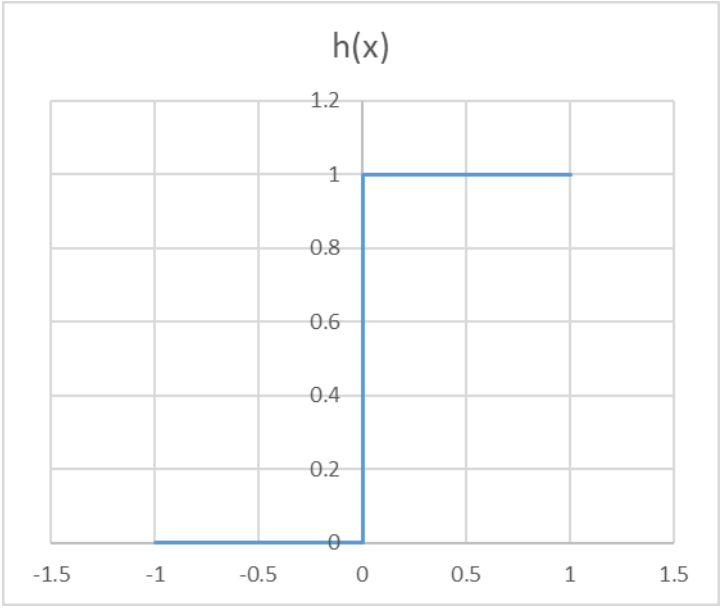
Organizing the Perceptron Methodology



$$y = h(w1x1 + w2x2 + b)$$

$$h(x) = \begin{cases} 0 & (x \leq 0) \\ 1 & (x > 0) \end{cases}$$

h(x) is called the "activation function"



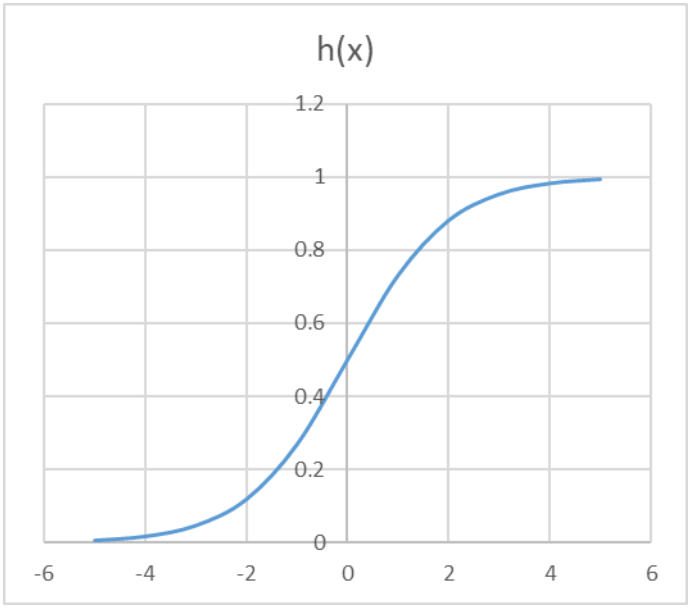
Activation Function

- The activation function can be an appropriate function if the following conditions are satisfied.
  - Monotonically increasing
  - Non-linear function
  - Can find derivatives (need not be continuous)

Activation Function conditions

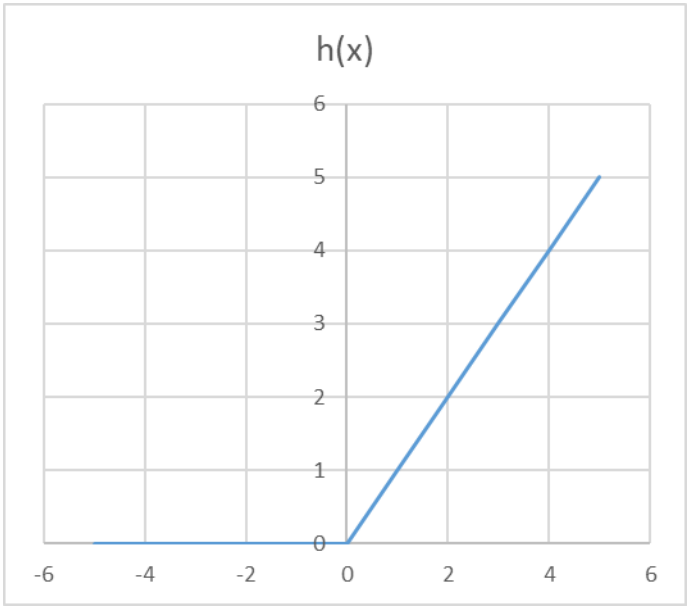
- It is monotonically increasing
  - Since the magnitude of the output is related to the magnitude of the input
- Non-linear function
  - Linear function (e.g., h(x) = ax is NOT good)
- Differentiable
  - Needed for learning

Frequently used activation functions



Sigmoid functions

$$\frac{1}{1+e^{-x}} = \frac{\tanh(x/2)+1}{2}$$

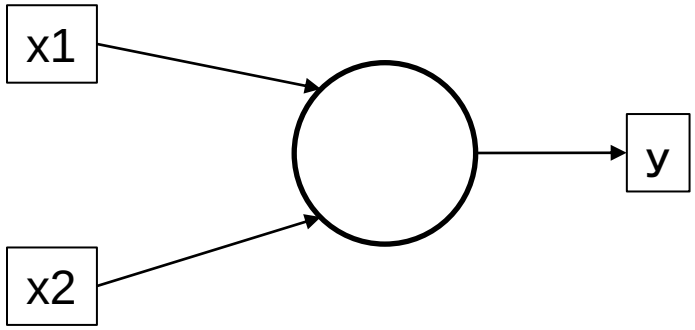


ReLU functions

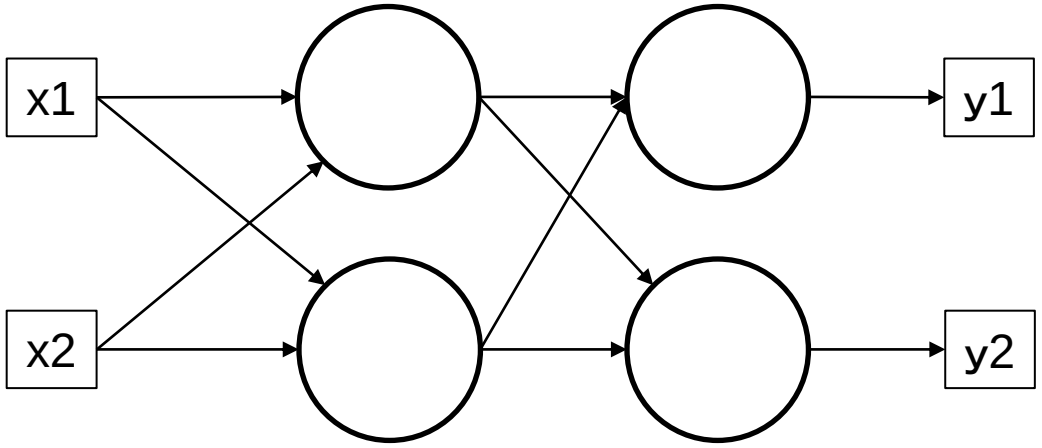
$$x^+ = \max(0, x)$$

Output of NN

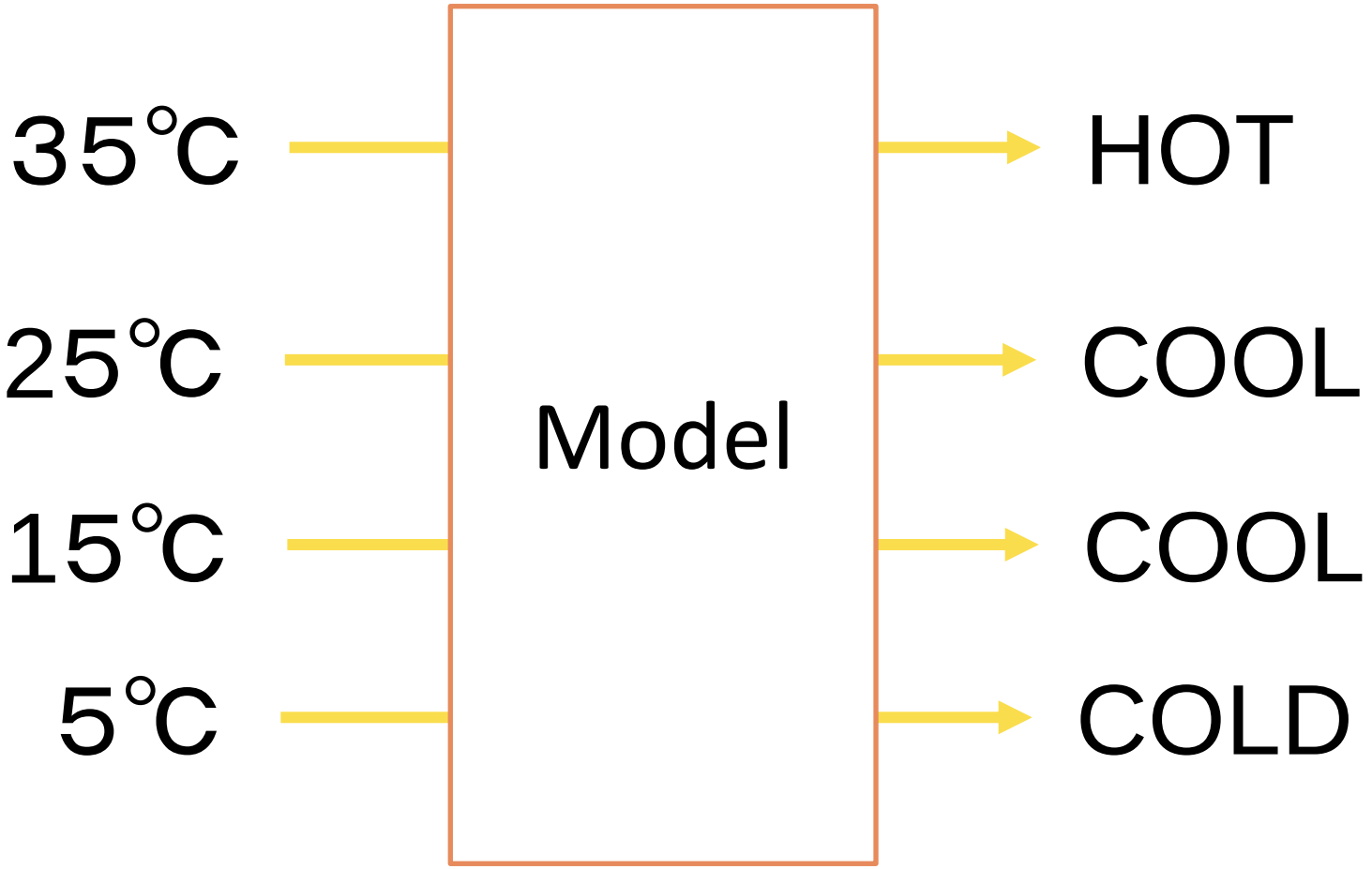
(0,0) → 0
(0,1) → 1
(1,0) → 1
(1,1) → 0



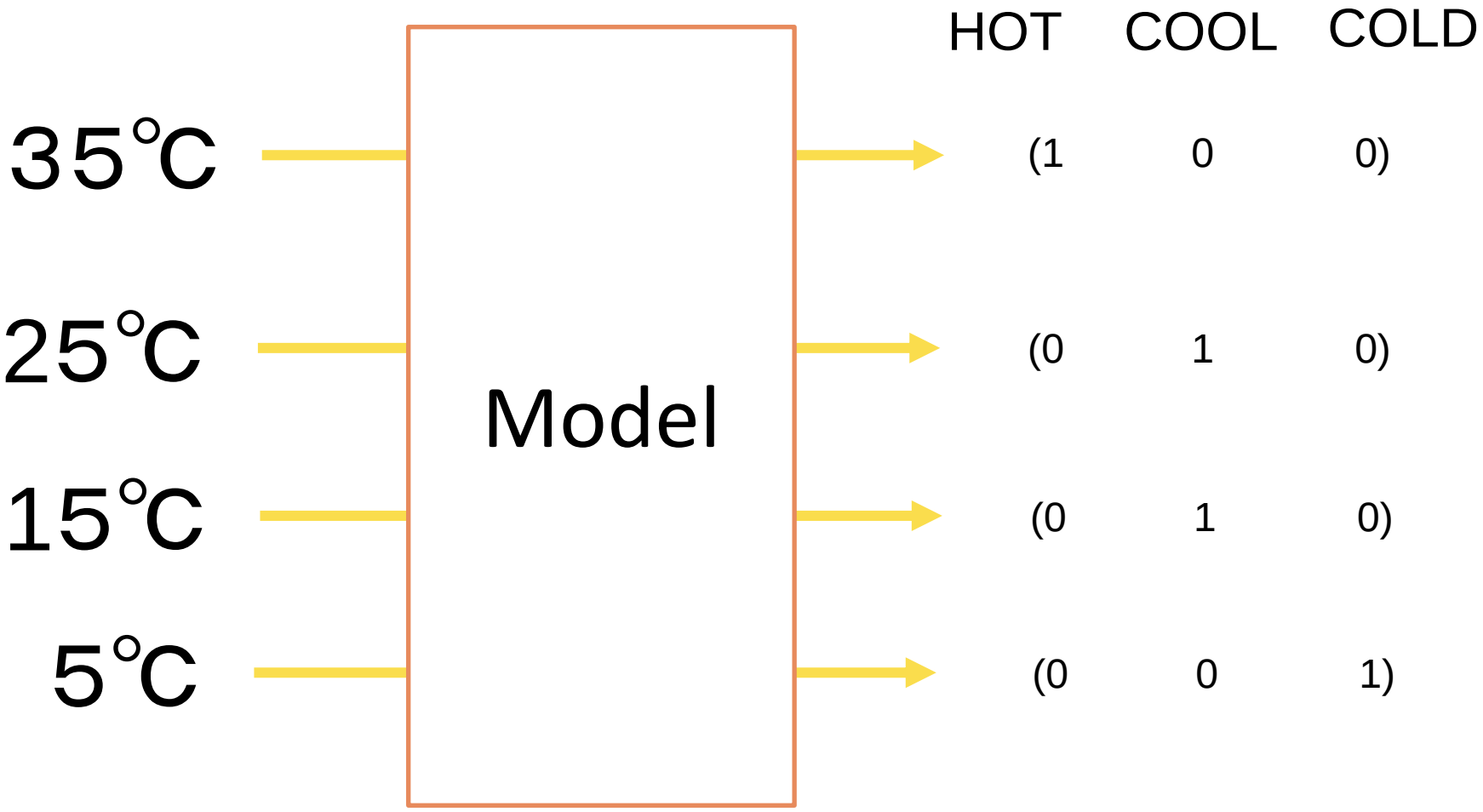
(0,0) → (1,0)
(0,1) → (0,1)
(1,0) → (0,1)
(1,1) → (1,0)



Previous output

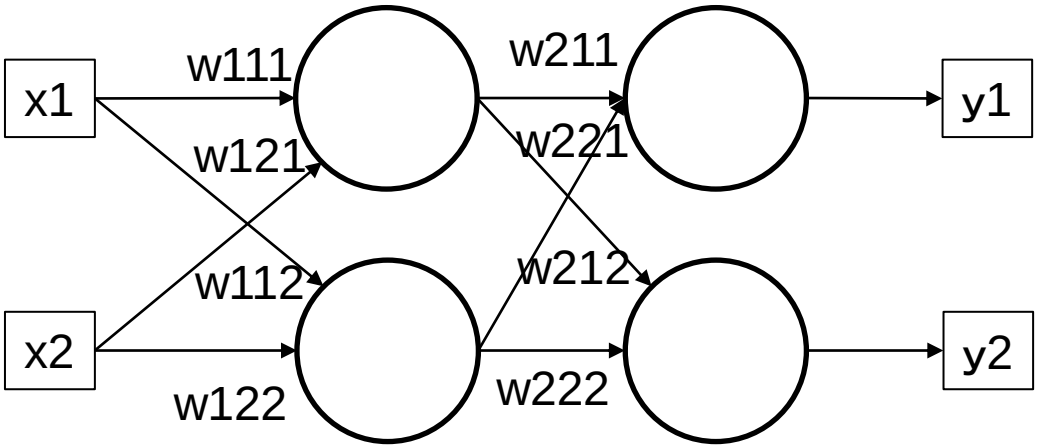


NN output



ok, find w

- How to find w?



w<layer><input><output>

In this neural network, there are 8 w's.
In addition, there are total 12 unknowns because of 4 bias b's

Finding unknowns

- Cannot be solved by algebraic equations.
  - Unknowns: 12
    8 w's and 4 b's
  - Equations: 8
    Set of Input and Output

- Solve analytically
  - Find appropriate w and b
    by testing inputs and outputs

Learning

- Use some w and b as an initial value
- Input data and get NN output
- Calculate the error from the correct answer and adjust w and b according to the error (back propagation method)

- Adjust w and b to minimize the error from the correct answer

Error with Correct Answer

- It's called the “loss function”.
- Example of Loss functions:

$$E = \frac{1}{2}\sum_{k=1}^{N}(y_k - t_k)^2$$

Sum of squares error

$$E = -\sum_{k=1}^{N} t_k \log y_k$$

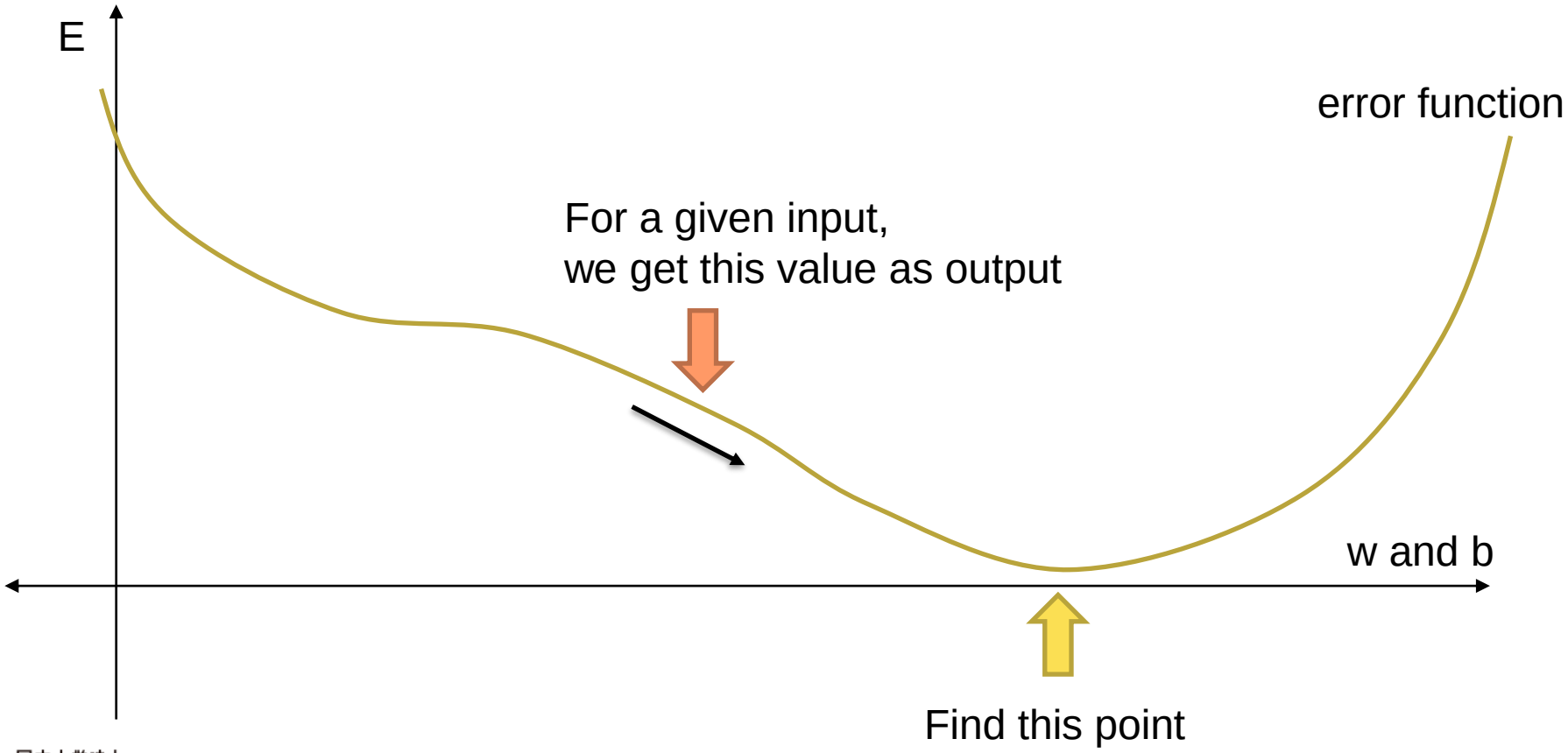
cross-entropy

t： expected output （Teaching data）

y：output of NN

DO learn

- Minimize error E

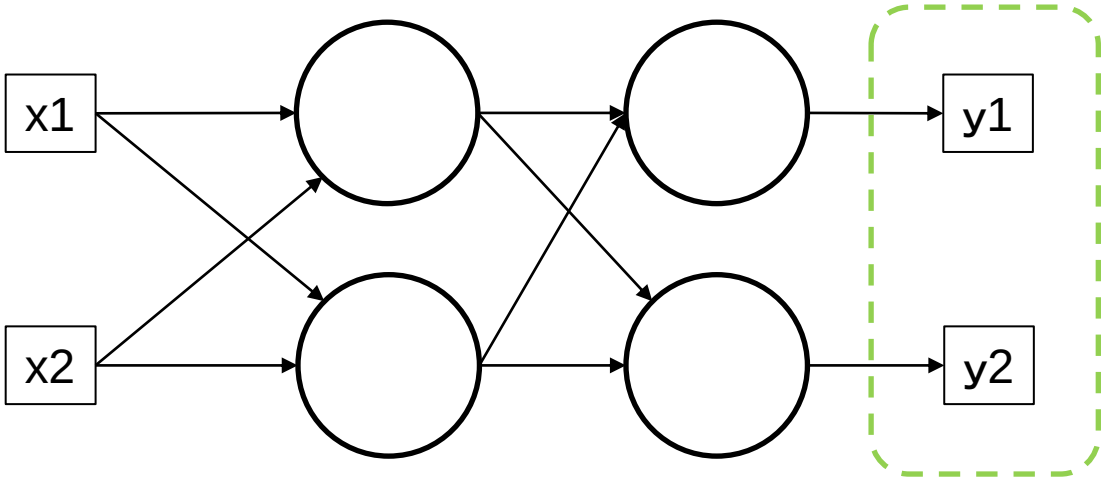


In practice

- Finding the partial derivative of the loss function
  - Loss function includes activation function, so derivative of activation function is also needed
  - Since there are multiple inputs, derivatives are needed for each of x1 and x2. (partial derivative)

Partial derivative

- Find a value to adjust w so that the error is minimized (this is the partial derivative)



Error between output and teaching data

Back Propagation Method

- Steps of learning

(1) Calculate the error E between the output of NN and the teaching data. If the error is small enough, the training is completed.

(2) Put the error E into the partial differential equation of the loss function, let A

(3) Put value A in the partial differential equation of the activation function, let B

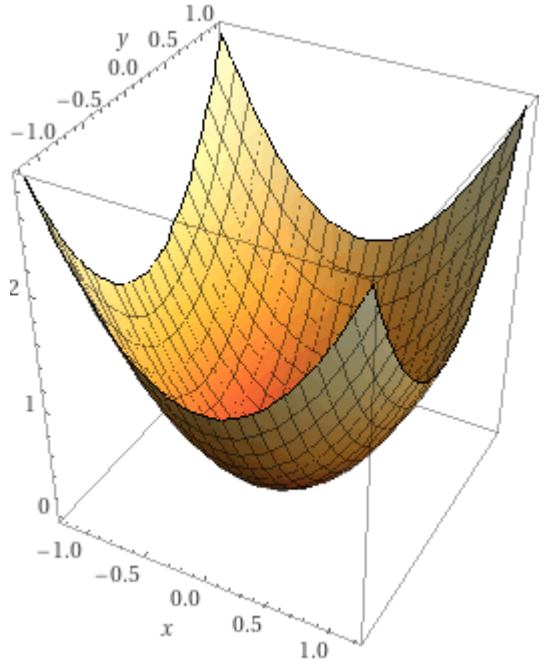
(4) Correct w according to the value B (in the direction of b)

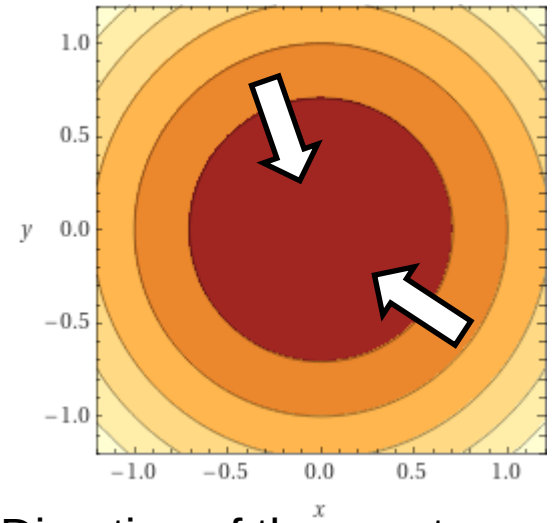
(5) Return to (1)

Some Issues

Learning = Adjusting w's

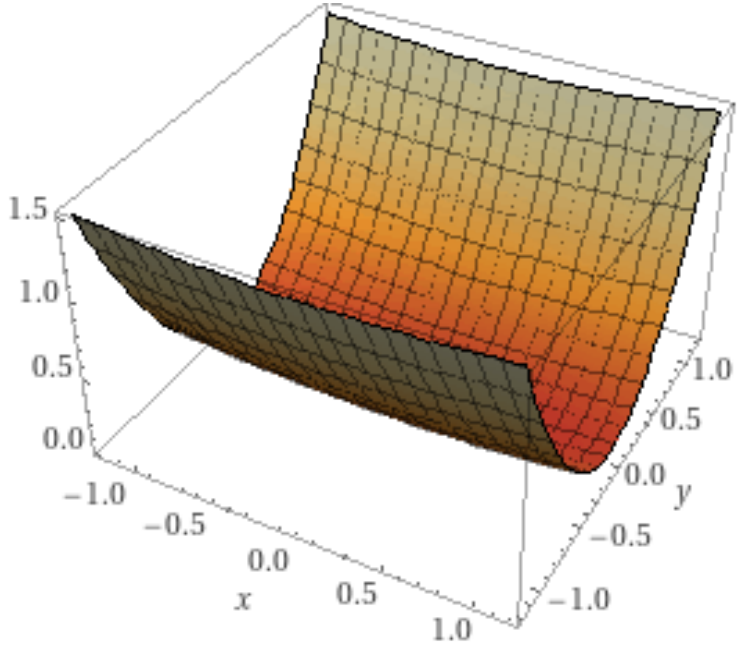
- How to adjust w
  - Gradient method: Adjustment proportional to the magnitude of the derivative

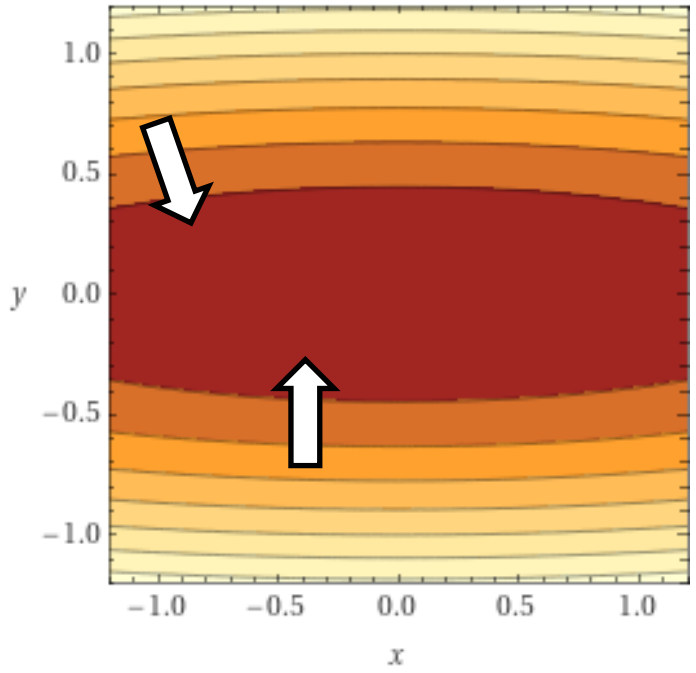




Direction of the greater gradient
(To the center of the circle)

Gradient method problem





Does not move toward the center
(moves slowly toward the center)

Learning Algorithms

- Algorithm for adjusting w's
  - SGD (Stochastic Gradient Descent)
  - Momentum
  - AdaGrad (Adamar Gradient)
  - RMSProp (AdaGrad + moving average)
  - etc.

Overfitting

- Learns the input data in detail.
  - Like a perfect output (1.000, 0.000, 0.000, 0.000, 0.000) for the training data
  - However, wrong output if input data is a little different from teacher data

Overfitting

- Balancing Data Scale and Model Scale

- Data size > Model size
  - Cannot learn
- Scale of data ≈ Scale of model
  - Can be trained
- Data size ＜ Model size
  - Too much learning (overfitting)

scaling

- Matching a range of values
  - Scaling of input data is especially important

- Examples
  - Input: Height, weight, BMI, average body temperature
- Output:
  - Health status: very good/Good/ok/bad/serious

- Adjustment of w does not work well
  because the range of values (magnitude of values)
  between inputs are different.

Scaling Examples

- Normalization
  - Set values in the range 0 to 1
- Standardization
  - Set the mean of the values to 0 and the standard deviation (SD) to 1

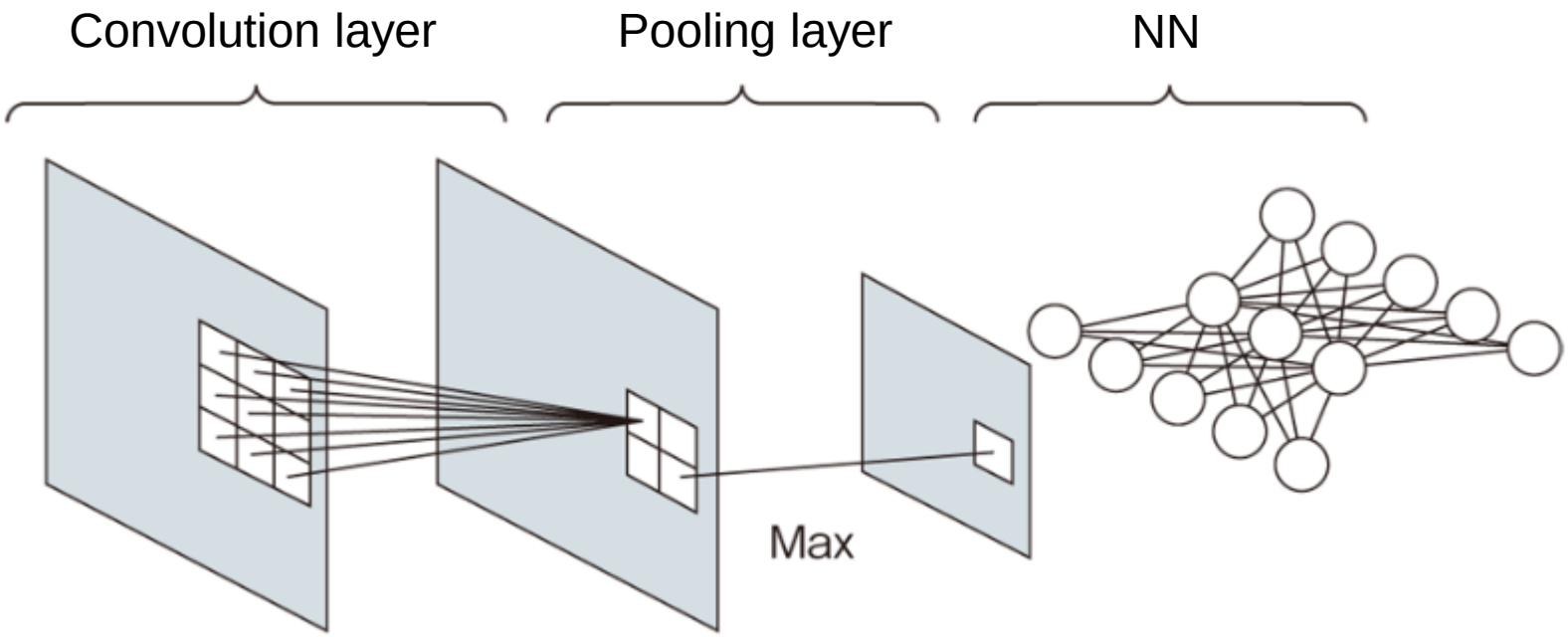
AI Applications

Speech and image recognition

- Data is spatially spread out
- CNN
  - Convolutional Neural Network

- Filtering for data with spread: convolution
- Extract the maximum value of the data: pooling (to discover features)

CNN

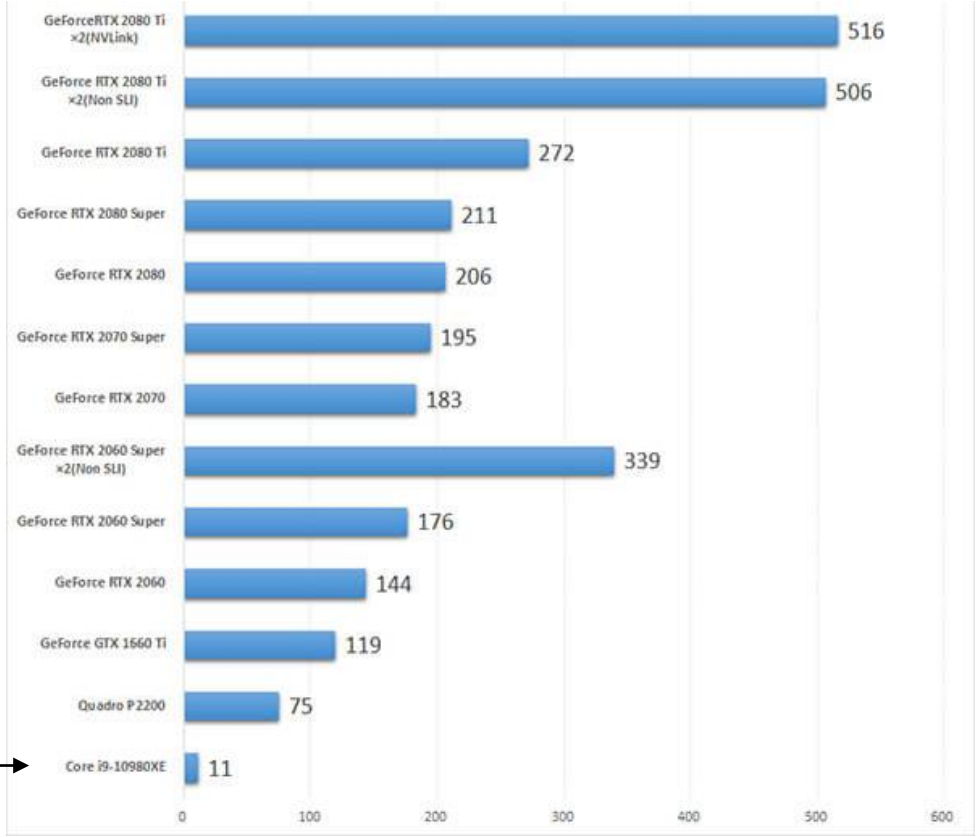

Convolution layer
Pooling layer
NN
Max
Lower resolution image

Example

- MNIST (database of handwritten numbers)
  - Recognize numbers from image data
  - Current error rate is 0.09

GPU

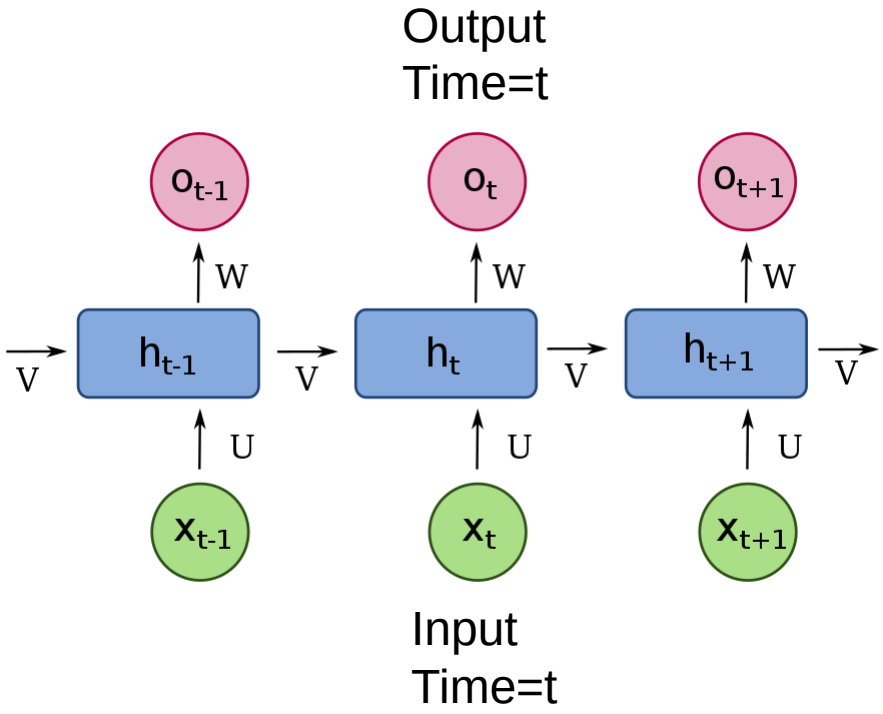
- Speed of matrix operations
  - Convolution
  - Back propagation (Learning)



CPU, Intel Core i9 →

Dealing with time series data

- RNN (Recurrent NN)
  - Treating past (most recent) output as input



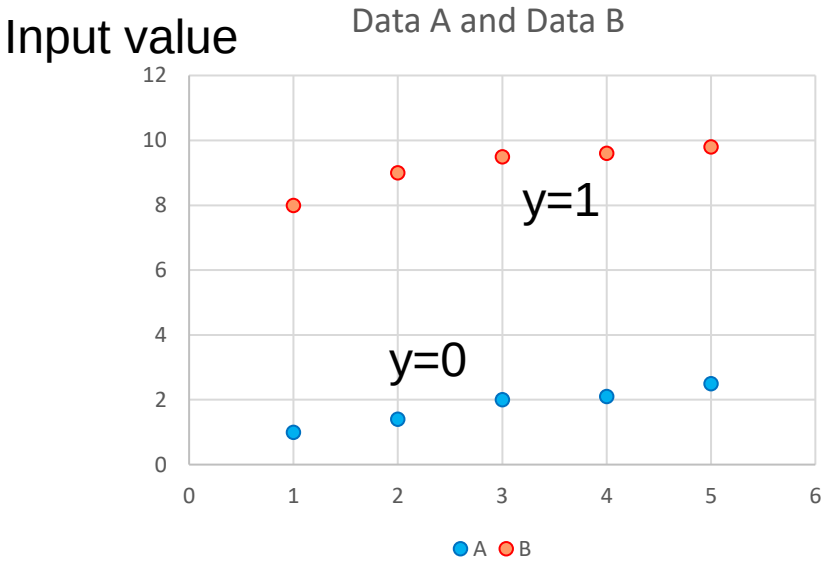
About Development Environment

- Python+Tensorflow(+keras)
  - Library of datasets (database)
  - Adopt for GPU.

- Google Collaboratory
  - https://colab.research.google.com/

DEMO

- Jupiter Notebook



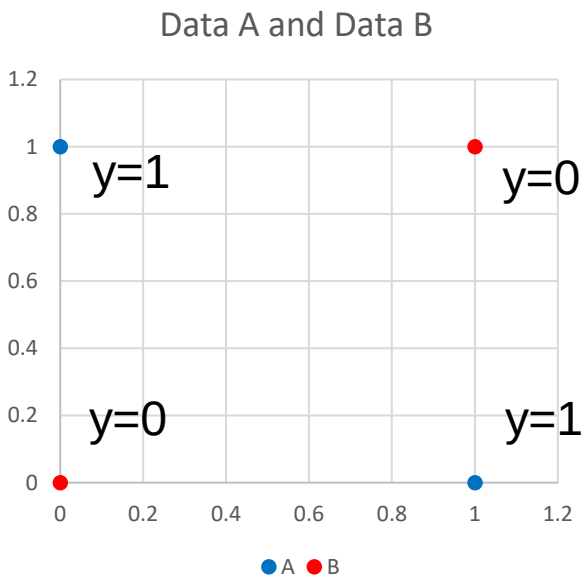
Inputs and Outputs

- Input/Output

```
x=np.array([[  1,1.5,  2,2.1,2.3,    8,  9,9.8,9.9, 10]])
y=np.array([[  0,  0,  0,  0,  0,    1,  1,  1,  1,  1]])
```

DEMO

- Jupiter Notebook



Inputs and Outputs

- Input/Output

```
x=np.array([[0,0,1,1],
            [0,1,0,1]])
y=np.array([[0,1,1,0]
```

Conclusion

Key Points of AI Application

- Have an input to output relationship
- Quality of input data
- Learning accuracy and speed
- Algorithm selection
  - Trial and error

- Many of features are provided in libraries, so the actual work is tuning

- AI is NOT silver bullet.

- You should know about:
  - Math
  - Clean Teaching data
  - Appropriate algorithms, functions
  - Evaluating outputs
  - Applying to real fields