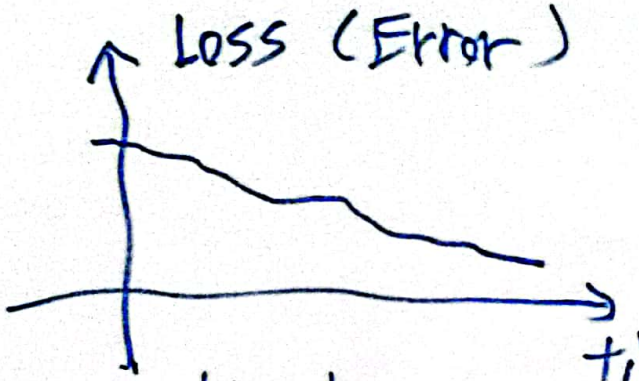
My name is Itsuo Kumazawa from Tokyo Tech.
Email adress is
kumazawa@is.l.titech.ac.jp

Ask questions and submit report to this email address.

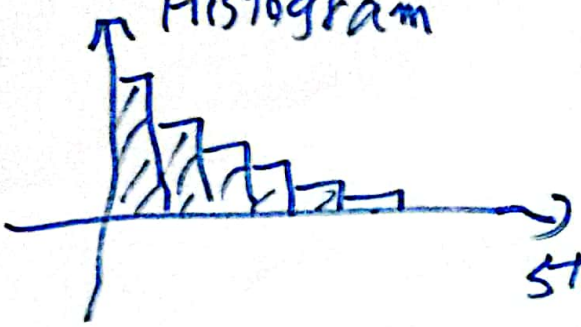
How to take credits for this course:
There is no examination.
But you have to submit reports on Five Tasks of programming.

What you should submits:
1. One file to explain about the five tasks.
   - Word file
   - PDF file
   - Jupiter Note book.

   ① The problem
   ② Methods and equations you used for the programing.
   ③ Computation results. Use graphical representations.



Graphical presentation.



Please appeal your results by using graphs and figures. Just numerics is not good.

To work for a company and to present your reseach results presentation skill is very important. It is more important than the research skill.

④ Please give discussions on the results. ↓ Most important.

2. Another file containing the source codes of your program.
python, C, C++, Matlab

Deep learning is so popular, there many functions for programming.

Just calling these functions is not good.
Please use just elementaly (numerical) functions. Tensor flow

Please submit 1, and 2. for the five tasks.

One file for 1. including ① ② ③ ④ for the five tasks

One file for 2. including source codes for the five tasks.

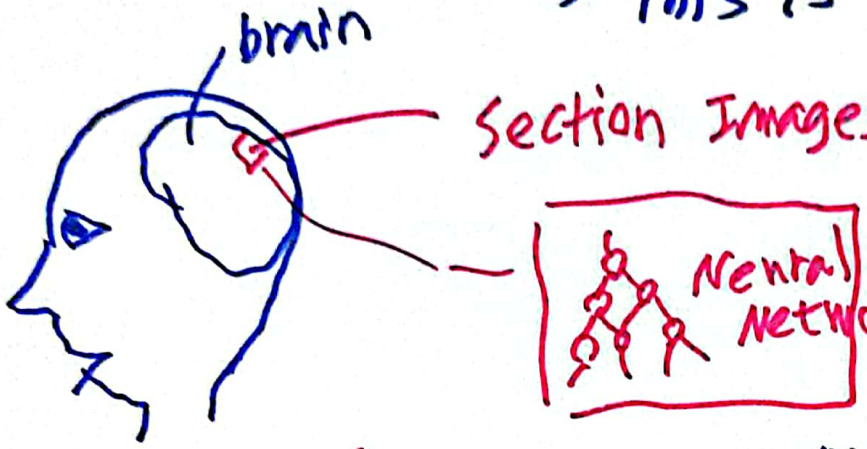
Submit two files by the end of February, Feb. 29th.

You can use the source codes found in the Net, But modify them for your own results.

Jan 8th.

① Biological Neural Network

↳ This is the base of AI.



simplified.

There are many kinds of neurons.
Each neuron receives many inputs
but computes just one output.
The neurons are connected by neural fibers.
There are feed back connections.

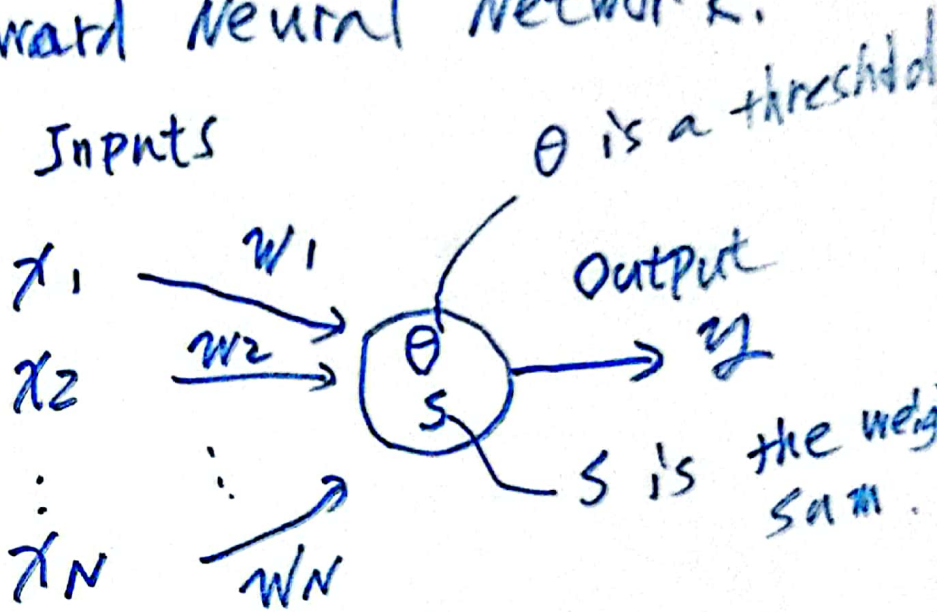
② Artificial Neural Network

Only one type of neuron.
Each neuron computs just one output
from many inputs.
simplified connections.

③ Structures of Artificial Neural Networks.

(1) Feed Forward Neural Network.

Neuron



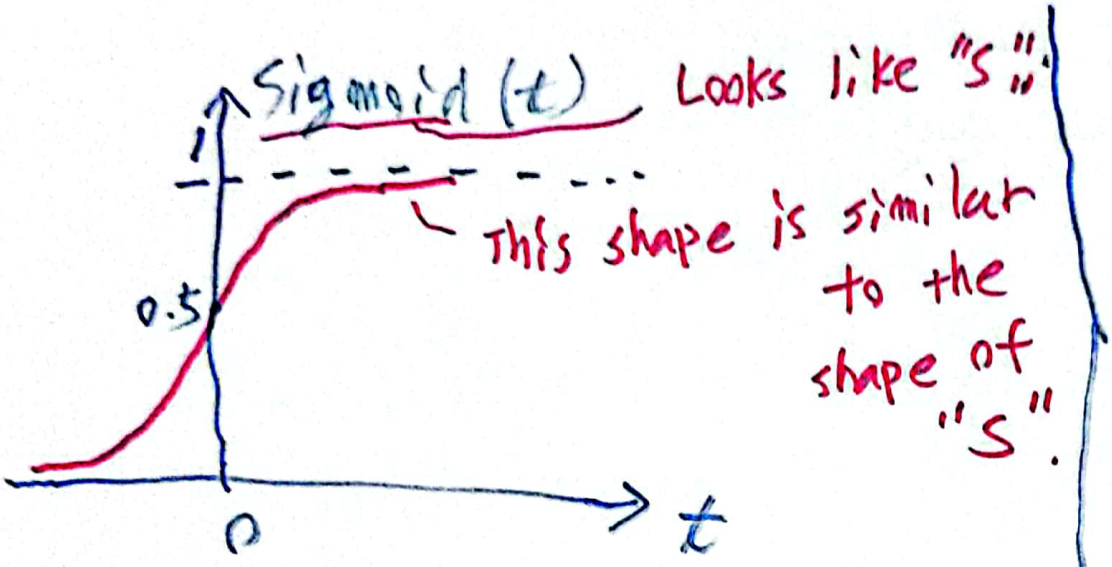
Weighted Sum

$$S = \sum_{i=1}^{N} w_i x_i$$

x_i Inputs.

w_i Connection weights

These connection weights are adjusted by machine learning.

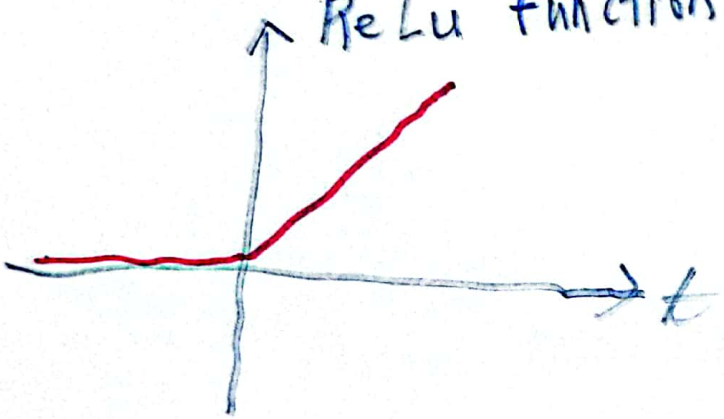


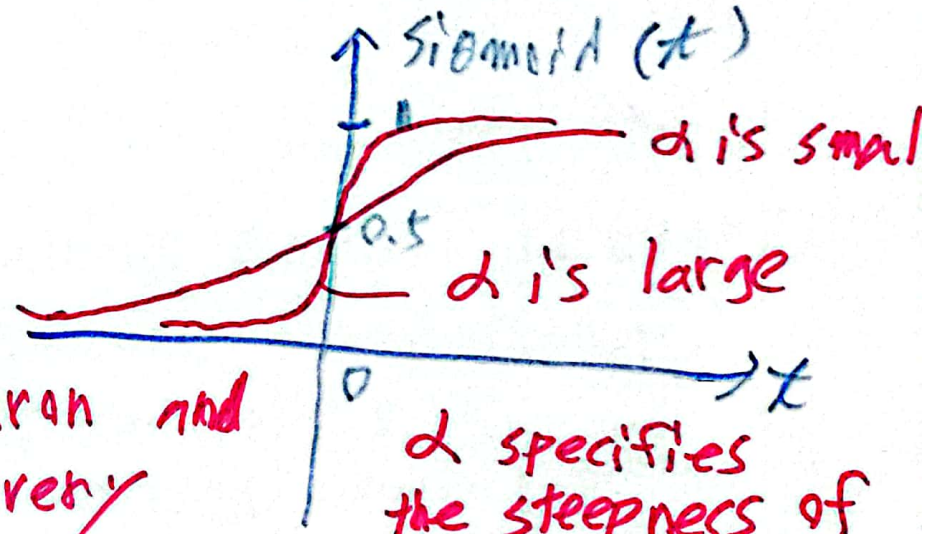
$$y = \text{Sigmoid}\left(\underbrace{s}_{\text{weighted sum}} - \underbrace{\theta}_{\text{Threshold}}\right)$$

$$\text{Sigmoid}(t) \Rightarrow \frac{1}{1+e^{-\alpha t}}$$

e: Exponential, α: Gain

↓ simplified

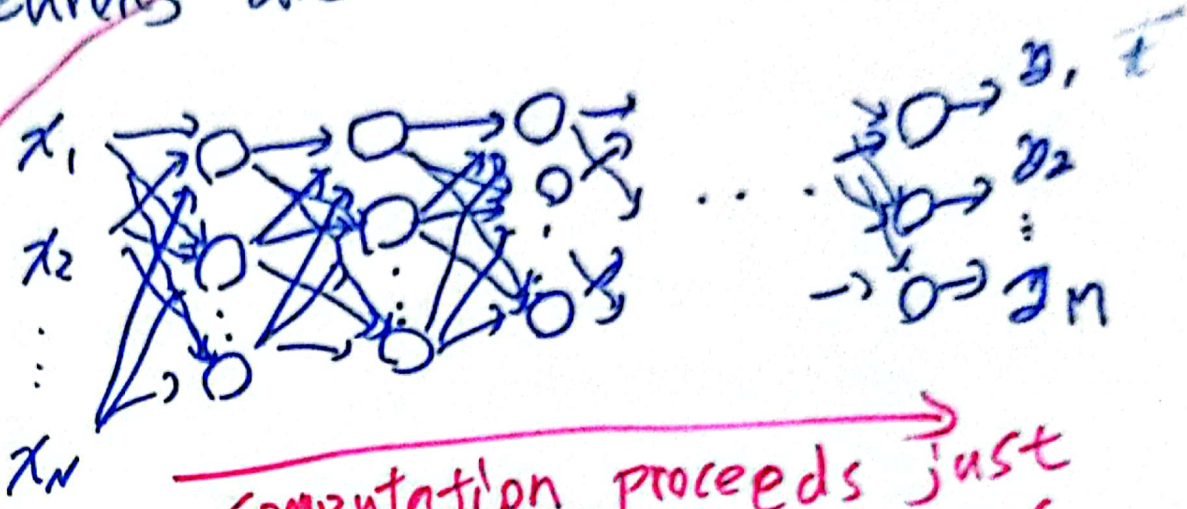




α specifies the steepness of Sigmoid Function.

The function of Neuron and Neural Network is very sensitive to α.

In a Feed Forward Neural Network, neurons are connected like this:



Computation proceeds just in a way.

The signals flow from the inputs to the outputs.

There is no feed back.

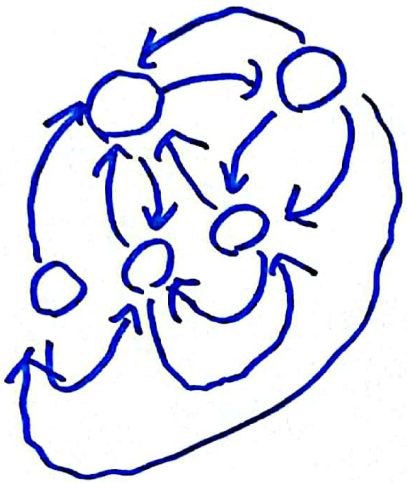
Many popular deep learning neural networks are based on Feed Forward Neural Networks.

The most popular deed learning neural network is CNN (Convolutional Neural Network).

CNN is also a kind of Feed Forward Neural Network.

(2) Recurrent Neural Network.

We use the same neuron as Feed ~~For~~ Forward Neural Network.
But they are interconnected with Feed Backs.



The neurons are connected with feed backs.

It is more complex than Feed Forward Neural Network.

In this course, we study about Recurrent Neural Netwo

The first task for your report.

Computation of the Neuron Models.

There are three models of neurons.

① Deterministic Binary Model.

- Deterministic → The output is always same if the inputs are same
- Binary → The output takes 0 or 1.

② Deterministic Continuous Model.

- Continuous → The output takes an analog value.

③ Probabilistic Binary Model.
(Stochastic)

① Deterministic Binary Model

$$s = \sum_{i=1}^{N} w_i x_i$$

weighted sum

Threshold

$$y = \begin{cases} 0 & \text{if } s < \theta \\ 1 & \text{if } s \geq \theta \end{cases}$$

② Deterministic Continuous Model.

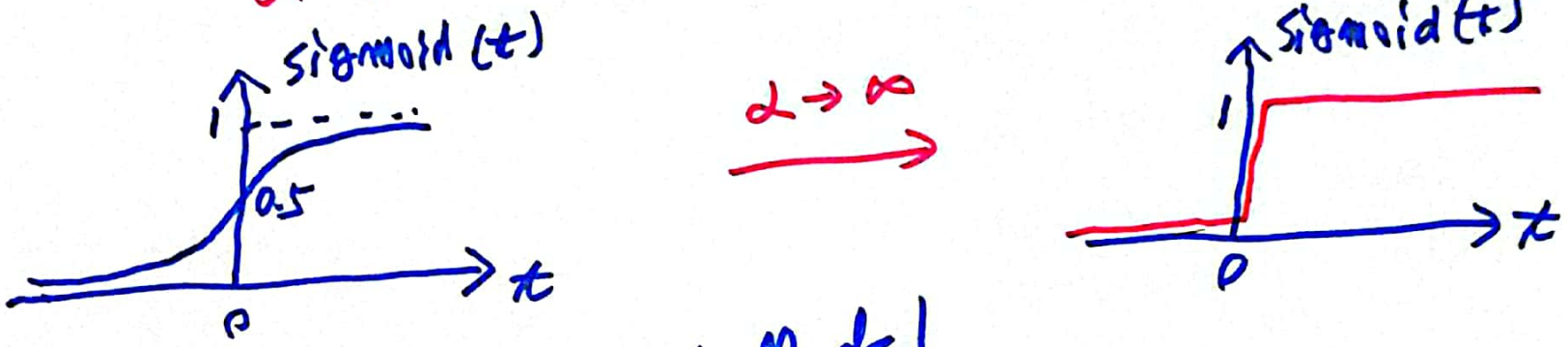
$$S = \sum_{i=1}^{N} w_i x_i$$

$$\text{sigmoid}(S) = \frac{1}{1+e^{-\alpha S}}$$

α is a gain

$$y = \text{sigmoid}(S-\theta)$$

$\alpha \to \infty$ then it becomes the deterministic model.



③ Probabilistic Binary Model.

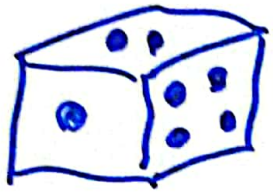
↓

Even for the same inputs, the output can take a different value.

A random number generator is used for the computation.

└ For example, in the C language ran() function is a random number generator

A die (dice) is a random number generator.



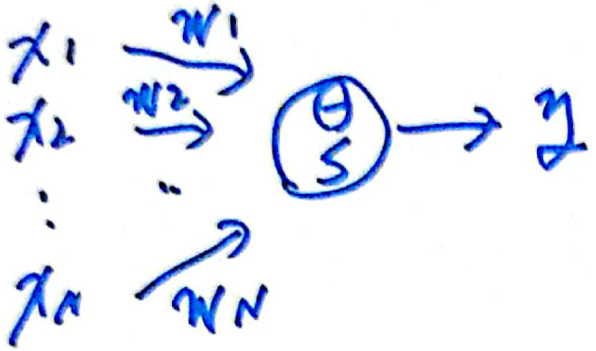
It generates a number between 1 and 6 randomly.

ran() is a die that generates the number between 0 and $2^{16} - 1$

└ This is different depending on the system. The actual number is defined in stdio.h in case of C language. The macro "RAND_MAX" has the defined number.

In case of probabilistic model, use a die to compute its output.

For example.



① Compute S by

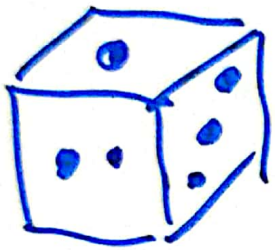
$$S = \sum_{i=1}^{N} w_i x_i$$

② Compute Probability.

$P = \text{sigmoid}(S - \theta)$

③ Use a die to compute the output.

$$y = \begin{cases} 1 & \text{if the number generated by the die is or equal less than } P * 6 \\ 0 & \text{if the die} \geq P * 6 \end{cases}$$



6 faces

It generates one of $1, 2, \cdots, 6$

$P = 0.5 \rightarrow$ 0 and 1 should happen equally with probability 0.5.

Just check it!

$p = 0.5 \rightarrow p * 6 = 3$

$$y = \begin{cases} 1 & \text{if the die} \leq 3 \\ 0 & \text{if the die} > 3 \end{cases}$$

This happens with the probability 0.5

This also happens with probability 0.5.

$p = 0.66.. \rightarrow p * 6 = 4$

Output y should be 1 with probability 0.66..

y should be 0 with probability 0.33..

$$y = \begin{cases} 1 & \text{if the die} \leq 4 \\ 0 & \text{if the die} > 4 \end{cases}$$

We understand "the Probabilistic Binary Model" can be computed using the die.

In the computer we cannot use the real die, so we use the artificial die (random number generator)

The artificial die has many faces.

The number of faces is

RAND_MAX ($= 2^{16} - 1$)

↳ stdio.h defines this number.

↙ change 6 to RAND_MAX

The output y of the Probabilistic Binary Model can be computed by

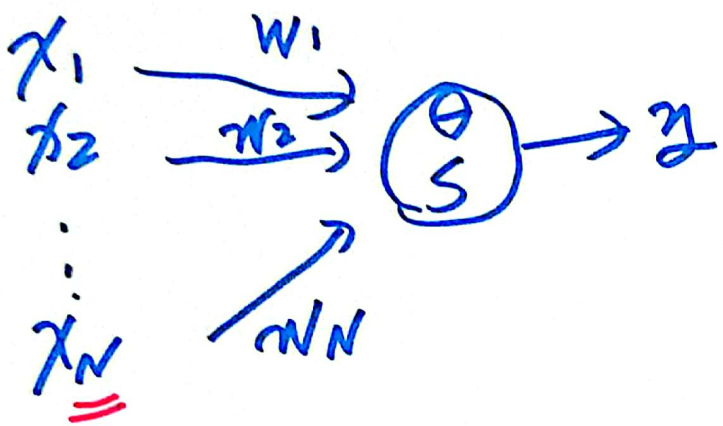
Task 1. ←

$$y = \begin{cases} 1 & \text{if } ran() \leq P * RAND\_MAX \\ 0 & \text{else } (ran() > P * RAND\_MAX) \end{cases}$$

$$P = sigmoid(S - \theta)$$

$$S = \sum_{i=1}^{N} w_i x_i .$$

Task 1 : Make the program for the Probabilistic Binary Model of the neuron.



$N = 4$

$x_1 = 0, x_2 = 1, x_3 = 1, x_4 = 0$

$w_1 = 2, w_2 = 0.5, w_3 = -$

$w_4 = -2$

$\theta = 0.5$

Compute y 100 times
1000 times
10000 times.

Check if $y = 1$ happens with probability

$P = \text{sigmoid}(s - \theta)$

α (Gain) is important. Please compare the results for different α, $\alpha = 0.1, 1, 5, 10$