SAMSUNG

Samsung Innovation Campus

Artificial Intelligence Course

Together for Tomorrow!
Enabling People
Education for Future Generations

Module 8 – Neural Network and Deep Learning

Artificial Intelligence Course

Module Description

Module objectives

- ✓ Build and train the deep neural networks.
- ✓ Optimize the deep learning neural networks preventing the overfitting and vanishing gradient problems.
- ✓ Gain proficiency with the deep learning libraries such as TensorFlow and Keras.

Module contents

- ✓ Unit 8a. Basics of Neural Network

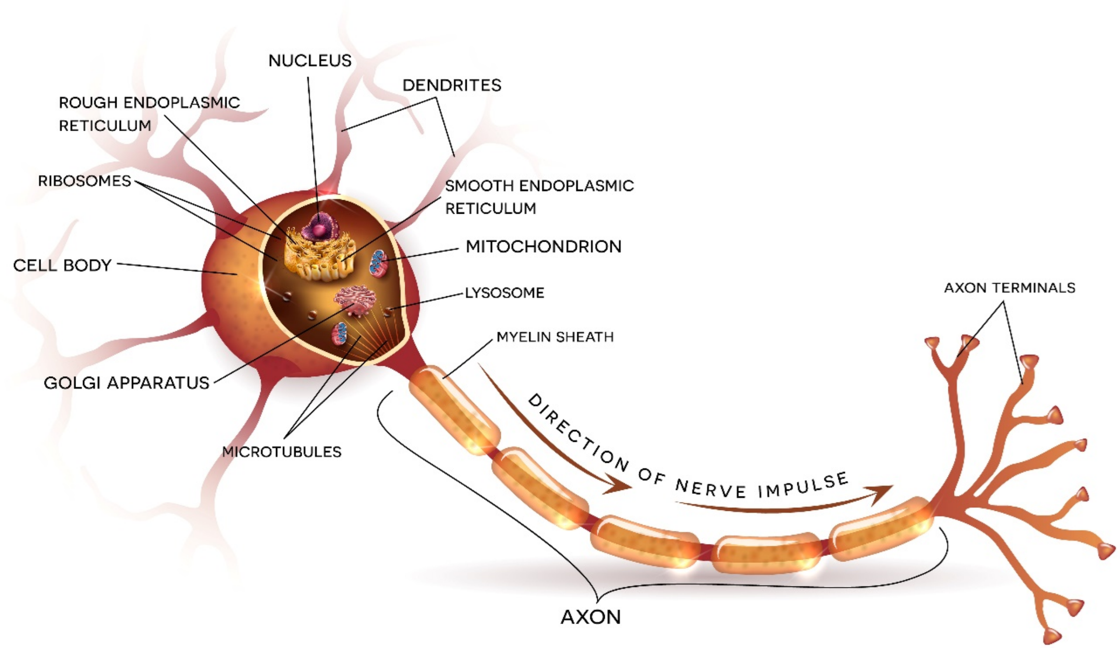
Unit 8a.

Basics of Neural Network

- **8a.1. Understanding Perceptron with Excel**
- 8a.2. Understanding Multilayer Perceptron with Excel
- 8a.3. From Multilayer Perceptron to Deep Learning
- 8a.4. Error Backpropagation and Gradient Descent to Reduce Errors
- 8a.5. Activation Function
- 8a.6. Deep Learning
- 8a.7. Problems with Deep Learning and the Solutions

Overview

- Biological origin:
  - Human's neuron is the smallest information processing unit that composes the neural system. It consists of a cell body, dendrites, and an axon. Below is a more simplified version of the above figure.

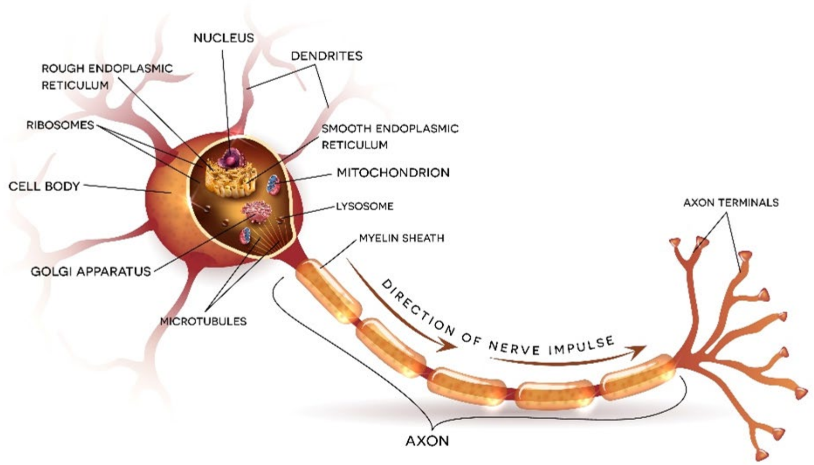


  - A cell body transmits simple operations, a dendrite transmits received signals, and an axon transmits the executed results.

Overview

Biological origin:

- They receive and store various information by exchanging chemical signals with adjacent neurons through a structure named synapse.
- The number of neurons in the brain of human being is about 1011 neurons. And the average neuron has around 1,000 synapses. So in the human brain 100trillion (1014) synapses are interconnected.
- In the 1940's, several researchers who studied biological neural network began to investigate how to emulate the mechanisms of the neural network. Perceptron is one of the early artificial neural network models.
- From here on, the term 'neural network' refers to 'artificial' neural network.



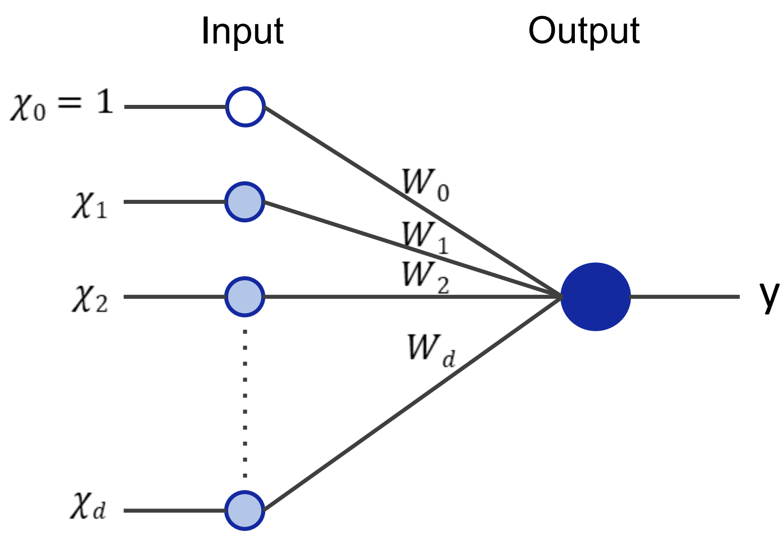
History of Neural Network

Frank Rosenblatt
(1928~1971)

- In 1943, McCulloch and Pitts first delineated the working mechanisms of neurons.
- Rosenblatt's perceptron is based on the McCulloch Pitts' neuron model.
- In 1958, Rosenblatt first proposed perceptron.
- In 1969, Minsky and Papert mathematically proved the limitations of the perceptron in their book 『Perceptrons』. It was found that a perceptron could not solve the XOR problem because it was only a linear classifier.

Perceptron

The Structure of a Perceptron



Structure of a perceptron

- Comprehend the structure of a perceptron by solving the OR function of the truth table.

Perceptron

- First, you need to understand the OR operation of the truth table. The results of the OR operation are below.

X1	X2	OR operation
False	False	F
True	False	T
False	True	T
True	True	T

- What are the weight values to solve the OR operation problem?

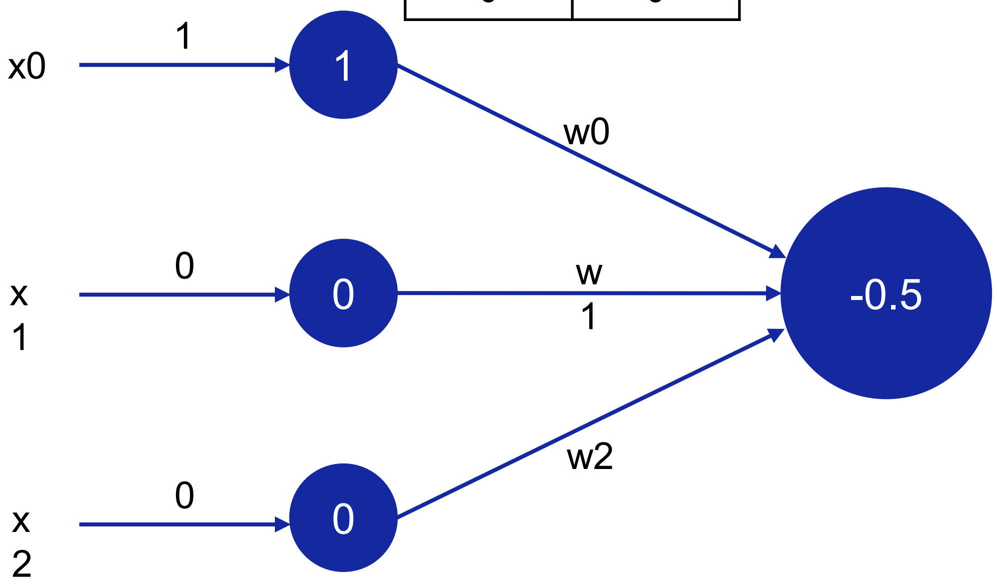
8a.1. Understanding Perceptron with Excel

Perceptron

- Suppose that we know the 'w' values to solve a problem. w0 is -0,5, w1 is 1, w2 is 1.
  (w: a weight vector)
- Execute Excel function after substituting x1 with 0, and x2 with 0.

x1	X2
0	0

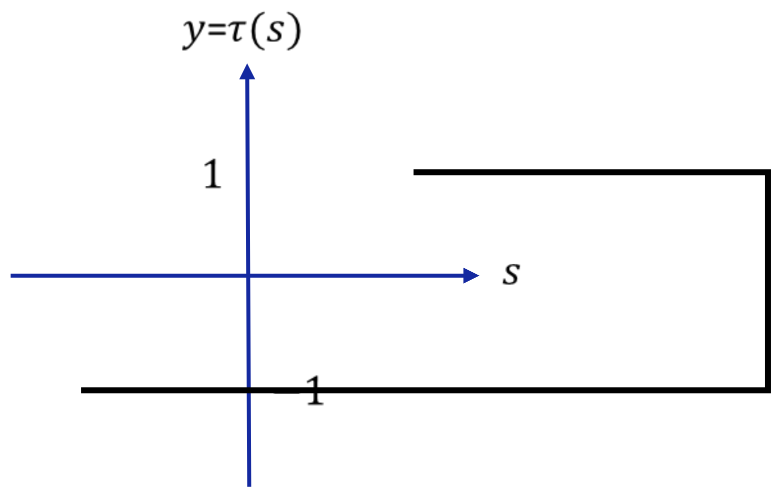
w0	w1	w2
-0.5	1	1



x0 * w0	-0.5
	+
x0 * w1	0
	+
x0 * w2	0
Sum	-0.5

Perceptron

- Put the value of the sum into a function.
- Here, that function is the step function.



Use threshold function as an activation function $\tau(s)$

Perceptron

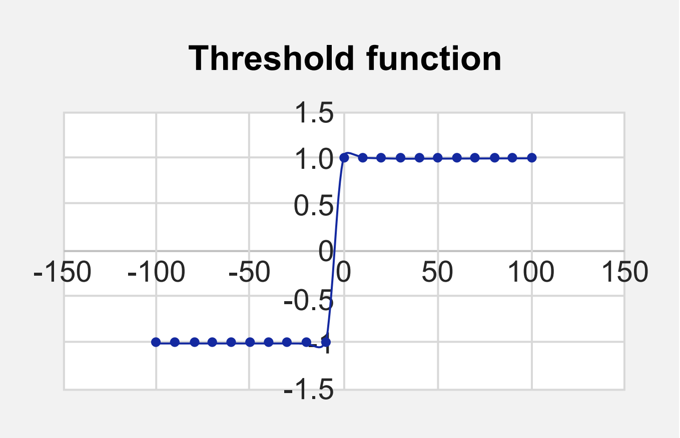
- A threshold function can be defined as follows, and we can create a graph by putting random values in the Excel table.

A threshold can be defined as follows,

$$y = \tau(s)$$

$$\text{here } s = w_0 + \sum_{i=1}^{d} w_i x_i, \quad \tau(s) = \begin{cases} 1 & s \geq 0 \\ -1 & s < 0 \end{cases}$$

-100	-1
-90	-1
-80	-1
-70	-1
-60	-1
-50	-1
-40	-1
-30	-1
-20	-1
-10	-1
0	1
10	1
20	1
30	1
40	1
50	1
60	1
70	1
80	1
90	1
100	1



Perceptron

- Put the result form the previous slide, -0.5 as s, and run it through the threshold function, the return is -1.
- Let's learn about the truth table.

A truth table displays truth or false for all results of the propositions or the combination of their Boolean functions.

For example, in case of the conjunction of two statements P and Q, $P \land Q$, the truth table can be constructed as below.

In addition, true·false is also notated as T·F or 1·0.

Proposition P	Proposition Q	
True	True	True
True	False	False
False	True	False
False	False	False

Perceptron

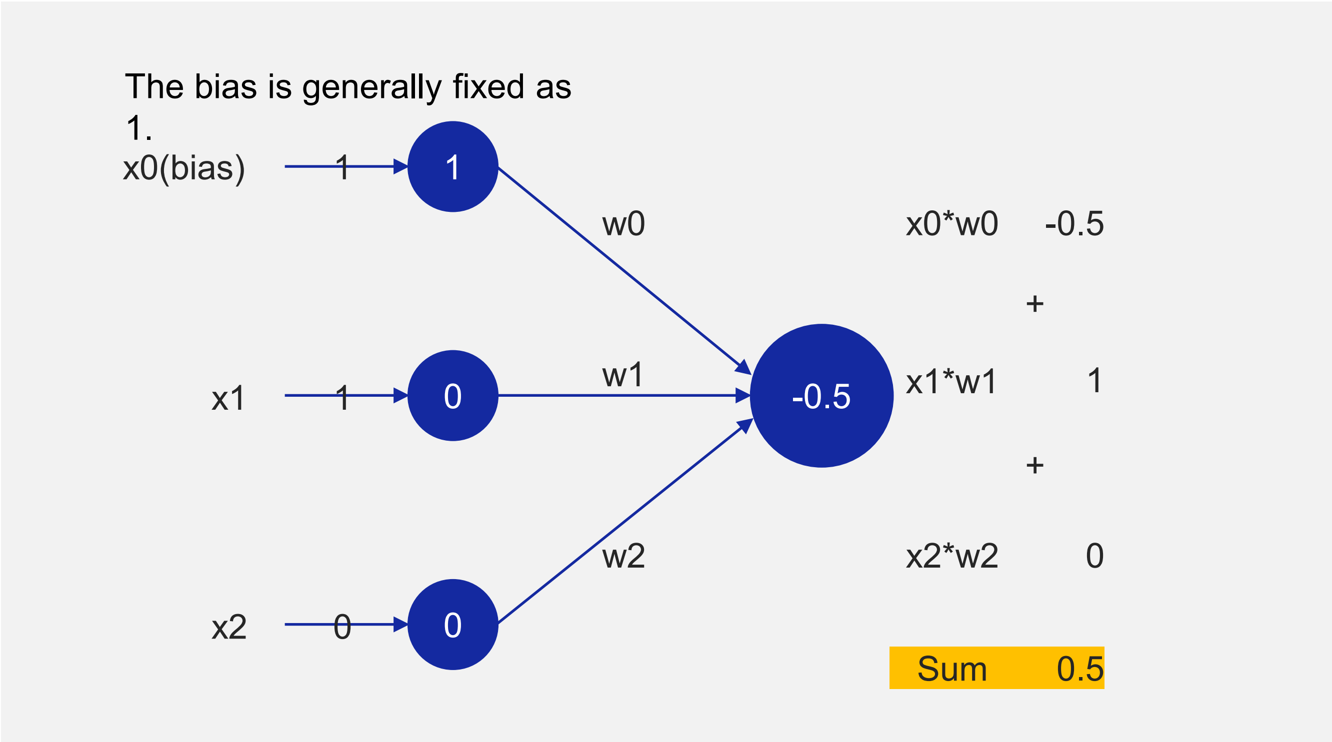
- A truth table can be expressed with 1 and 0 as follows.

x1	x2
0	0
1	0
0	1
1	1

- The figure from Slide 9 put 0 for the first variable x1, and 0 for x2 as well.
- x0 is called a bias, and it initializes as 1.

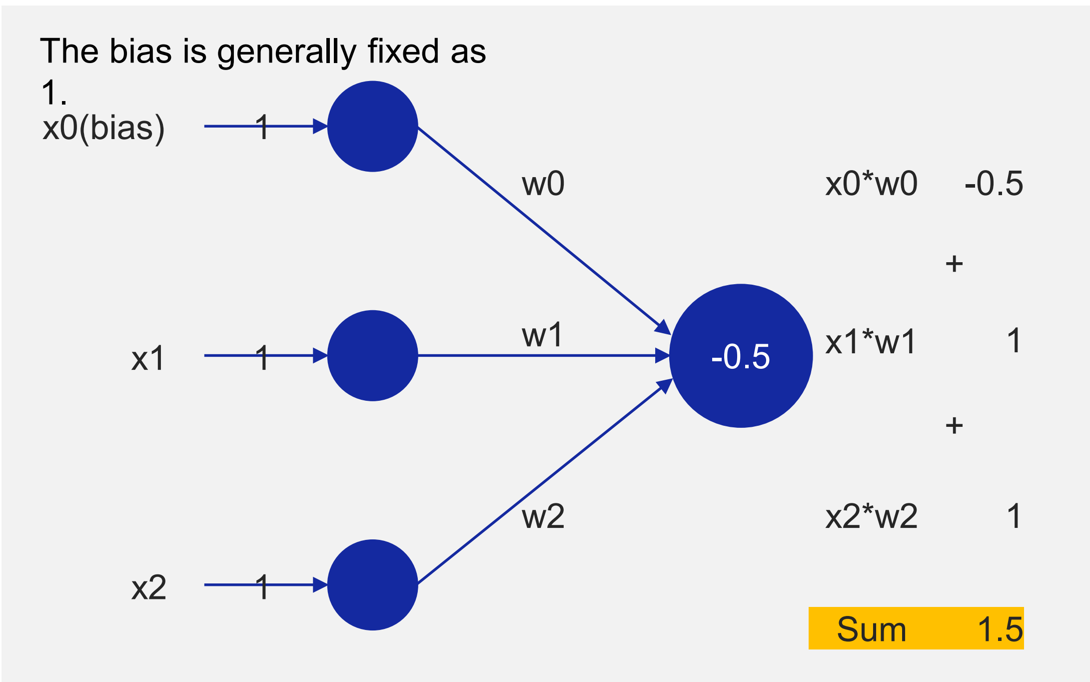
Perceptron

- Substitute x1 with 1, and x2 with 0.



Perceptron

- The sum is 0.5. Run this through the threshold function, and the result is 1.
- Likewise, if we process all values in the truth table through the perceptron, the results are as follows.

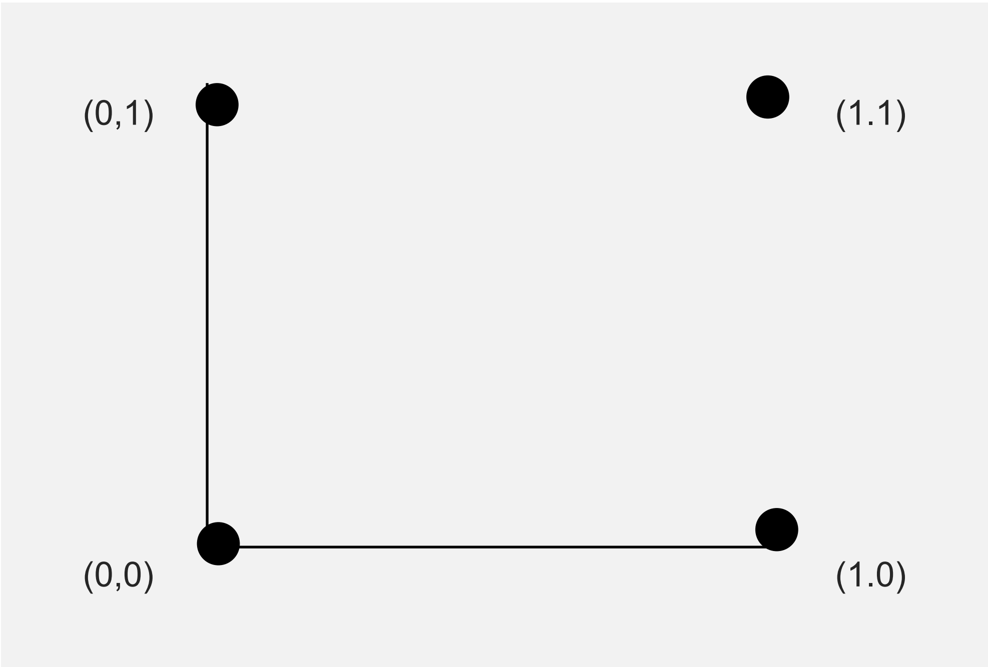


- 4 results can be displayed in a table.

x1	x2		
0	0	-0.5	-1
1	0	0.5	1
0	1	0.5	1
1	1	1.5	1

Perceptron

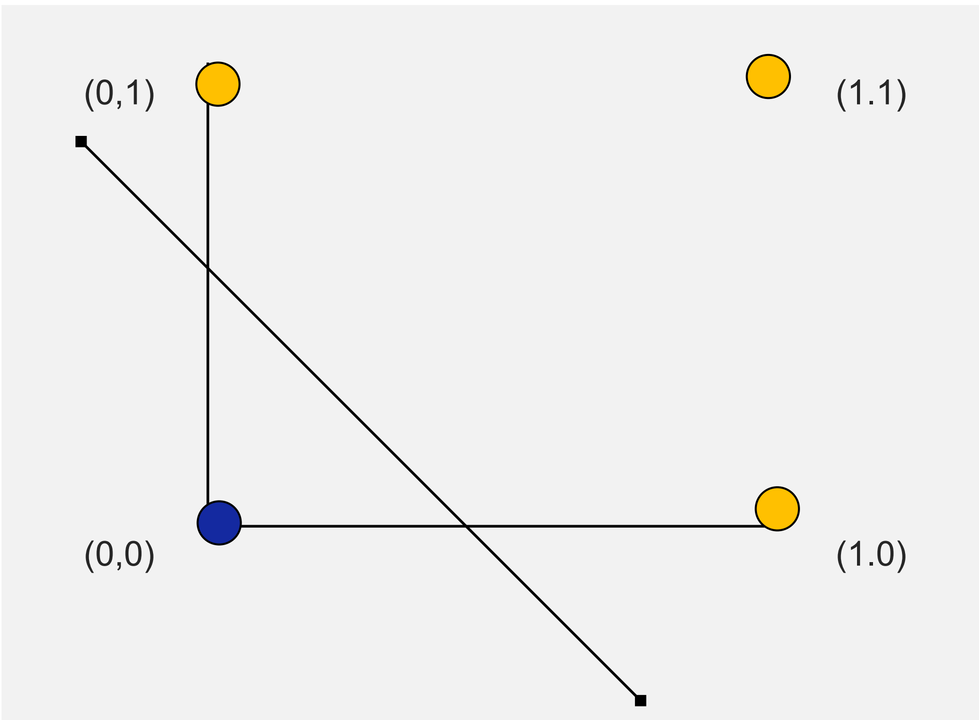
- These results can be shown in the coordinate plane as follows.
- The results can be grouped into that of point (0,0) or the rest.



x1	x2		
0	0	-0.5	-1
1	0	0.5	1
0	1	0.5	1
1	1	1.5	1

Perceptron

- Such classification can be presented as a geometrical figure as follows.
- If you recall the basic notion of machine learning through linear regression, machine learning is a process that builds regression equation with the predicted values of the slope and y-intercept. The figure below shows how linear regression forms a line that separate the two groups.



x1	x2		
0	0	-0.5	-1
1	0	0.5	1
0	1	0.5	1
1	1	1.5	1

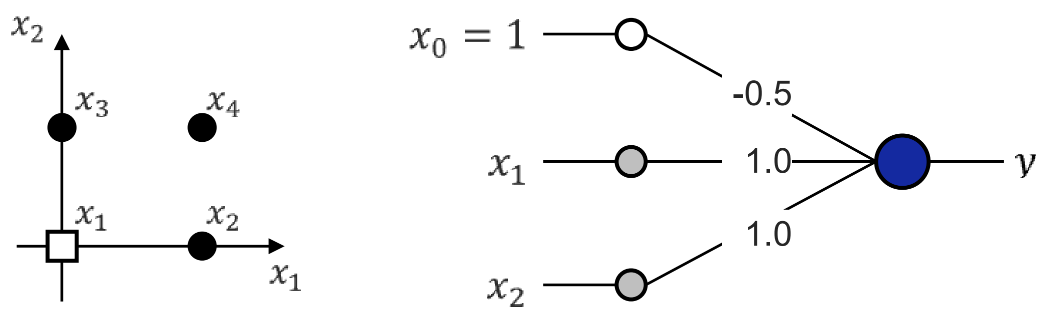
- This table displays the OR operation of a truth table.

OR operation

x1	x1	x2	Value before running threshold function	Value after running threshold function
0	0	0	=$A$2*$C$7+B2*$C$8+C2*$C$9	=IF(D2>=0,1,-1)
	0	1	=$A$2*$C$7+B3*$C$8+C3*$C$9	=IF(D3>=0,1,-1)
	1	0	=$A$2*$C$7+B4*$C$8+C4*$C$9	=IF(D4>=0,1,-1)
	1	1	=$A$2*$C$7+B8*$C$8+C5*$C$9	=IF(D5>=0,1,-1)

w1	-0.5
w2	1
w3	1

$$x_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, y_1 = -1, x_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, y_2 = 1, x_3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, y_3 = 1, x_4 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, y_4 = 1$$



(a) Train set (b) Perceptron

Example of a perceptron's operation using the OR logic gate

OR operation

Let's input four samples to the perceptron and check the results.

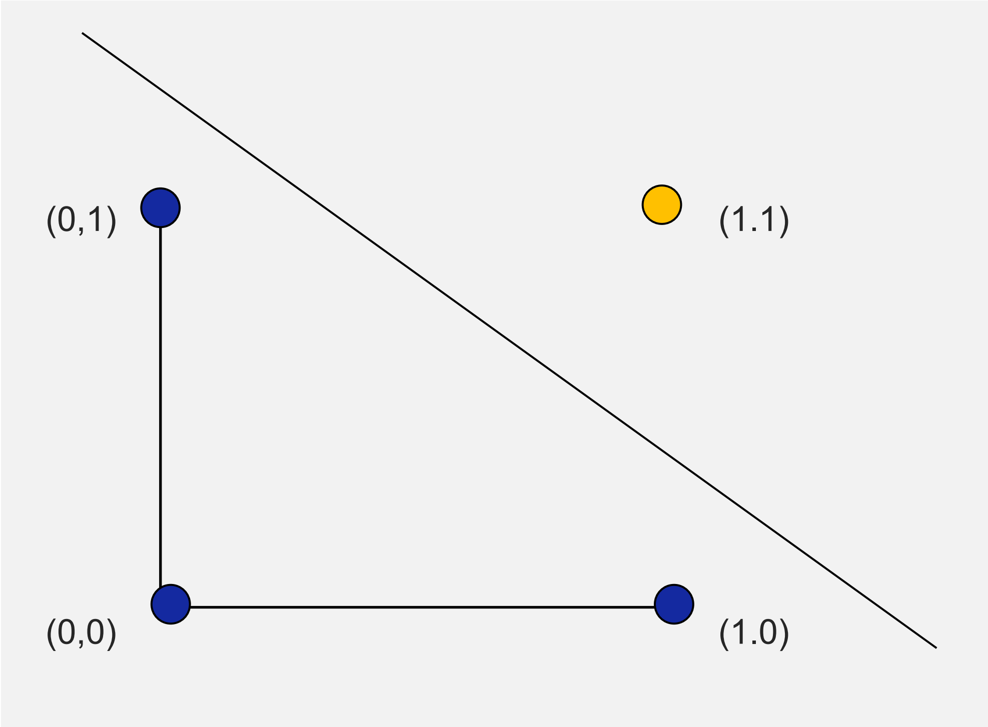
$$x_1 : s = -0.5 + 0 * 1.0 + 0 * 1.0 = -0.5, \qquad \tau(-0.5) = -1$$
$$x_2 : s = -0.5 + 1 * 1.0 + 0 * 1.0 = 0.5, \qquad \tau(0.5) = 1$$
$$x_3 : s = -0.5 + 0 * 1.0 + 1 * 1.0 = 0.5, \qquad \tau(0.5) = 1$$
$$x_4 : s = -0.5 + 1 * 1.0 + 1 * 1.0 = 1.5, \qquad \tau(1.5) = 1$$

- As you've seen in the previous slide, the perceptron delivered correct results for all four samples.
- It can be said that this perceptron classifies the train set with 100 % performance.

AND operation

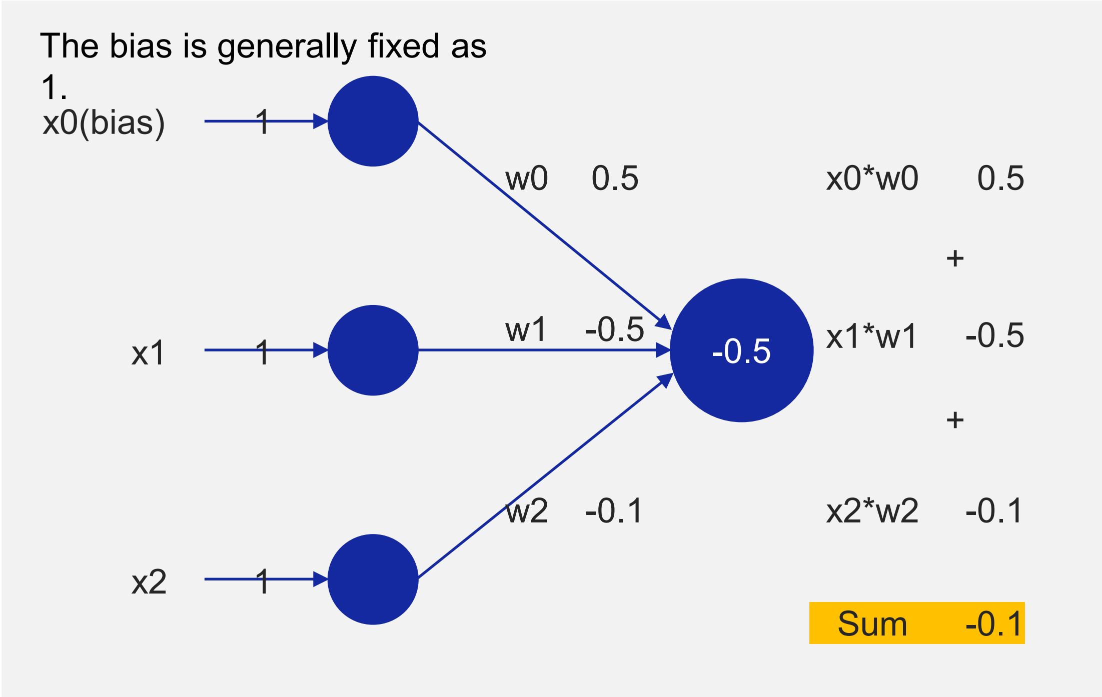
- To recap, you can solve the OR problem by perceptron with the appropriate w values.
- Now, the AND operation. The truth table is on the left, and the geometrical solution is the line that separates the units into two groups.

x1	x2	AND operation
0	0	F
1	0	F
0	1	F
1	1	T



AND operation

- Find the values of w0, w1, w2 that return the result from the previous slide.
- First, let's apply a random value, and then gradually change the values of w0, w1, and w2 to find the solution.

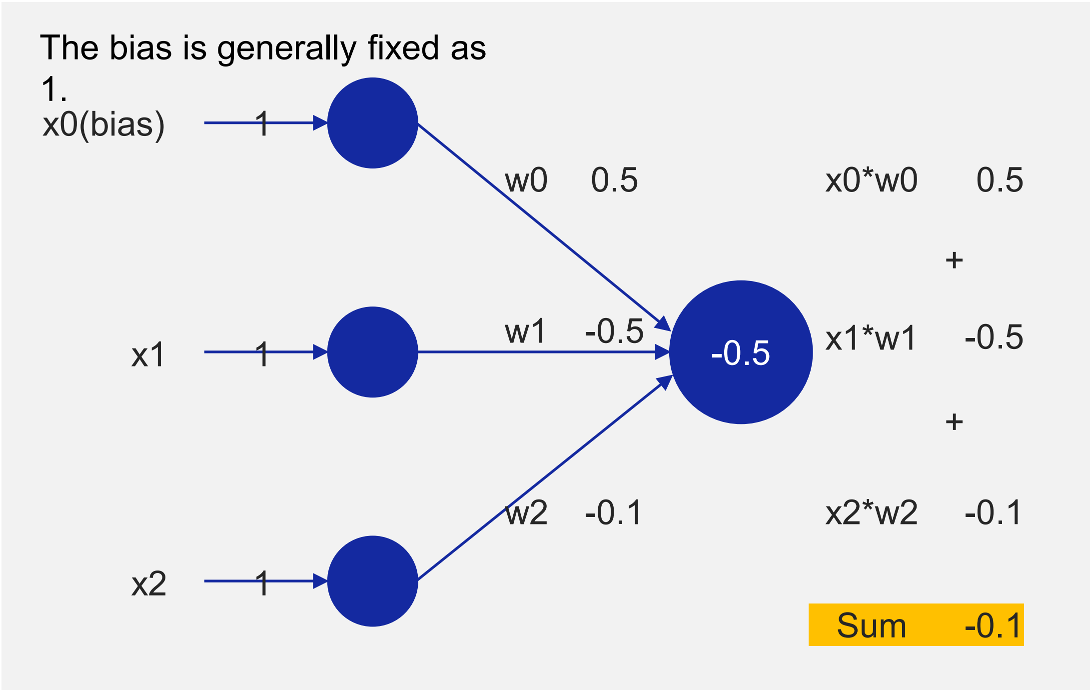


- 4 results can be displayed in a table.

x1	x2		
0	0	0.5	1
1	0	0	1
0	1	0.4	1
1	1	-0.1	-1

- AND operation of a truth table.

AND operation



- 4 results can be displayed in a table.

x1	x2		
0	0	0.5	1
1	0	0	1
0	1	0.4	1
1	1	-0.1	-1

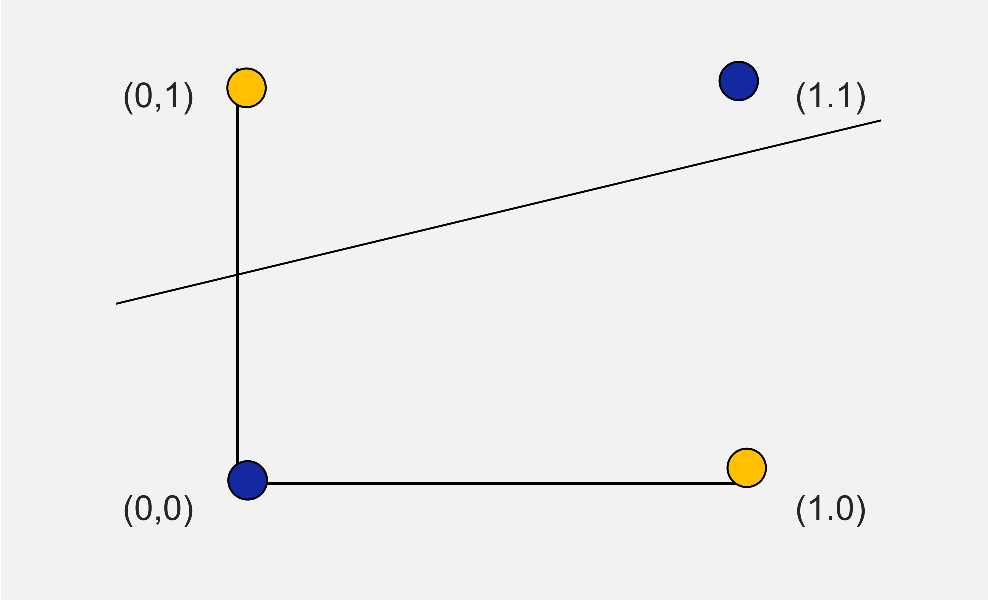
- AND operation of a truth table.

- By substituting w0 with 0.5, w1 with -0.5, w2 with -0.1, the equation solves the AND operation.
- You found a structure that with appropriate values for the w vector, solves a certain problem.
- Although the perceptron solved the OR and AND problem, it could not solve the XOR. Let's find out why.

XOR Operation

- XOR operation in a truth table.
- In XOR, the result is true if two propositions are the opposite.

x1	x2	XOR Operation
0	0	F
1	0	T
0	1	T
1	1	F

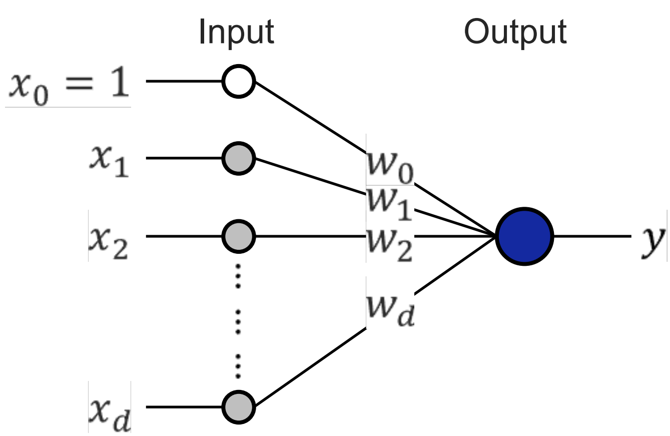


- Since the perceptron is a linear classifier, you cannot find a line that separate the blue dots and the red dots.
- A multilayer perceptron solved the problem.

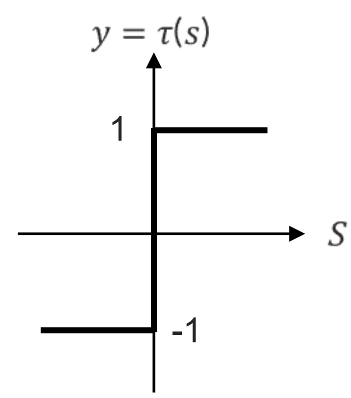
8a.1. Understanding Perceptron with Excel

Summary of Perceptron's Structure

- It has an input layer and an output layer.
- The input layer does not operate, so the perceptron is considered a single layer structure.
- The i th node of the input layer takes x_i from the feature vector $x = (x_1, x_2, \cdots, x_d)^{\mathrm{T}}$.
- The bias node always takes 1 as input.
- The output layer has a single node.
- The connection of the i th node of the input later and the output layer has weight w_i.



(a) Structure of a perceptron



(b) Use threshold function as an activation function $\tau(s)$

Mechanism of a perceptron

Unit 8a.

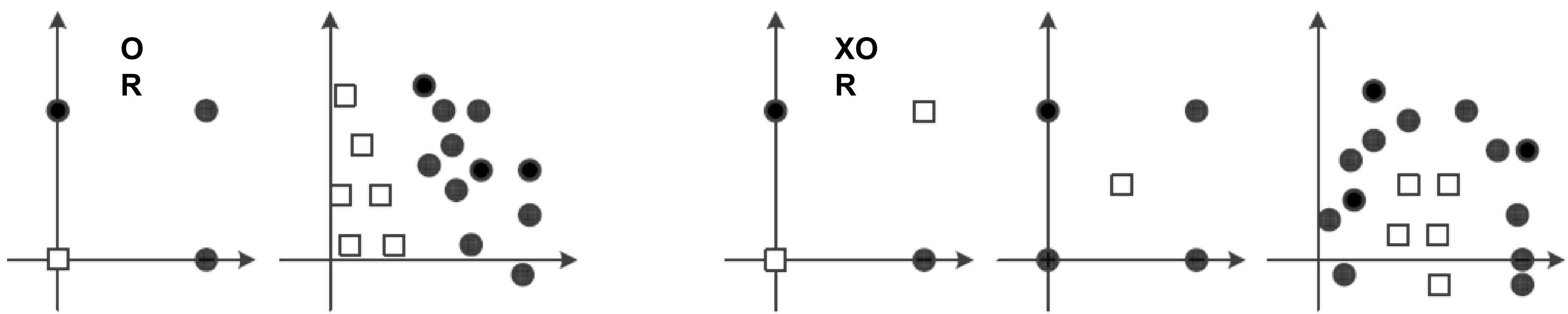
Basics of Neural Network

Understand Multilayer Perceptron with Excel

Perceptron cannot compute the XOR function.

- Linear classification is impossible, and certain amount of errors occur.
- 75% accuracy is the limit in XOR function.

Linearly separable case and linearly inseparable case



(a) Linearly separable

(b) Linearly inseparable

Multilayer Perceptron

- In XOR, the inputs are not linearly separable.
- It could not solve a simple problem like XOR.

- This was claimed by Marvin Minsky, an MIT professor who was the pioneer of the AI field, in his book <Perceptrons>, published in 1969.

  □ People realized that emulating the diagram of ‘neuron □ neural network □ intelligence’ through the diagram of ‘perceptron □ artificial neural network □ artificial intelligence’ was not an easy task.

  After this publication, research on AI took a downturn.

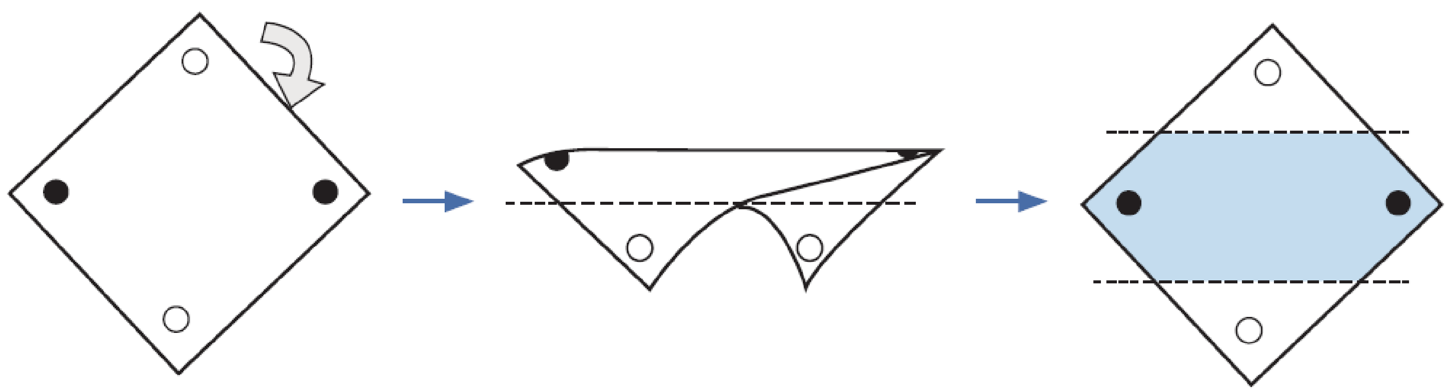
  After 10 years, this problem was solved by ‘multilayer’ perceptron.

Multilayer Perceptron

- What are the ways you can separate the inputs that are not linearly separable?
- First, the inputs can be separated in a non-linear form.
- Second, a two-dimensional figure on a plane, could be modified into a three-dimensional figure to enable separation.

Solution for the XOR problem

- Artificial Intelligence scientists had to overcome the XOR problem to develop an artificial neural network.
- An innovative idea provided hints for the solution.

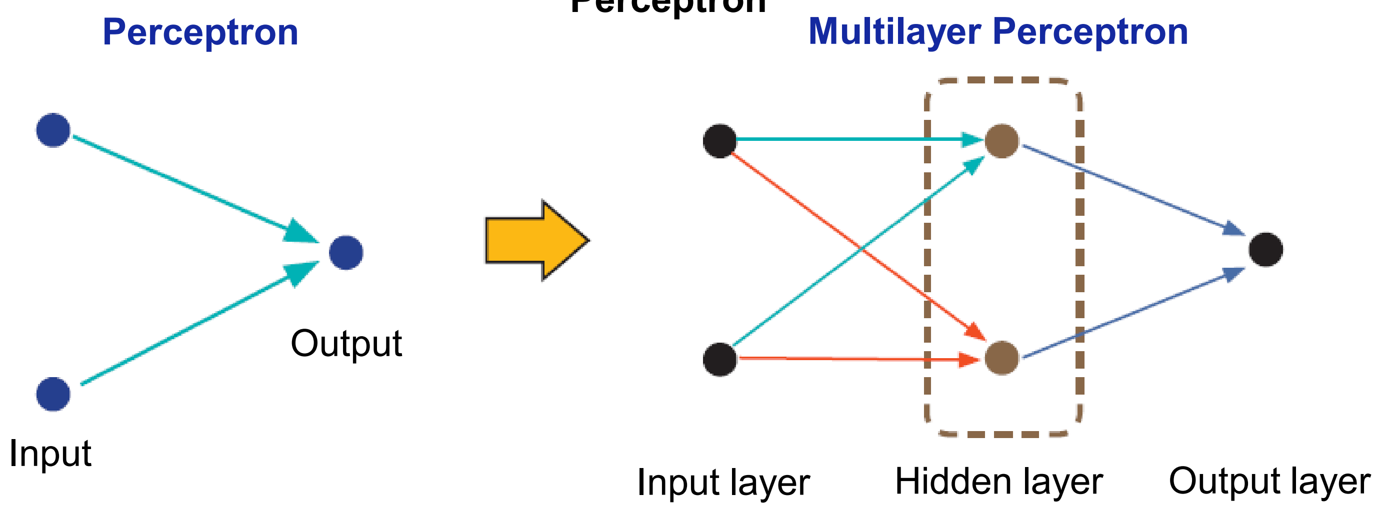


- You had to bend the plane!

Solution for the XOR problem

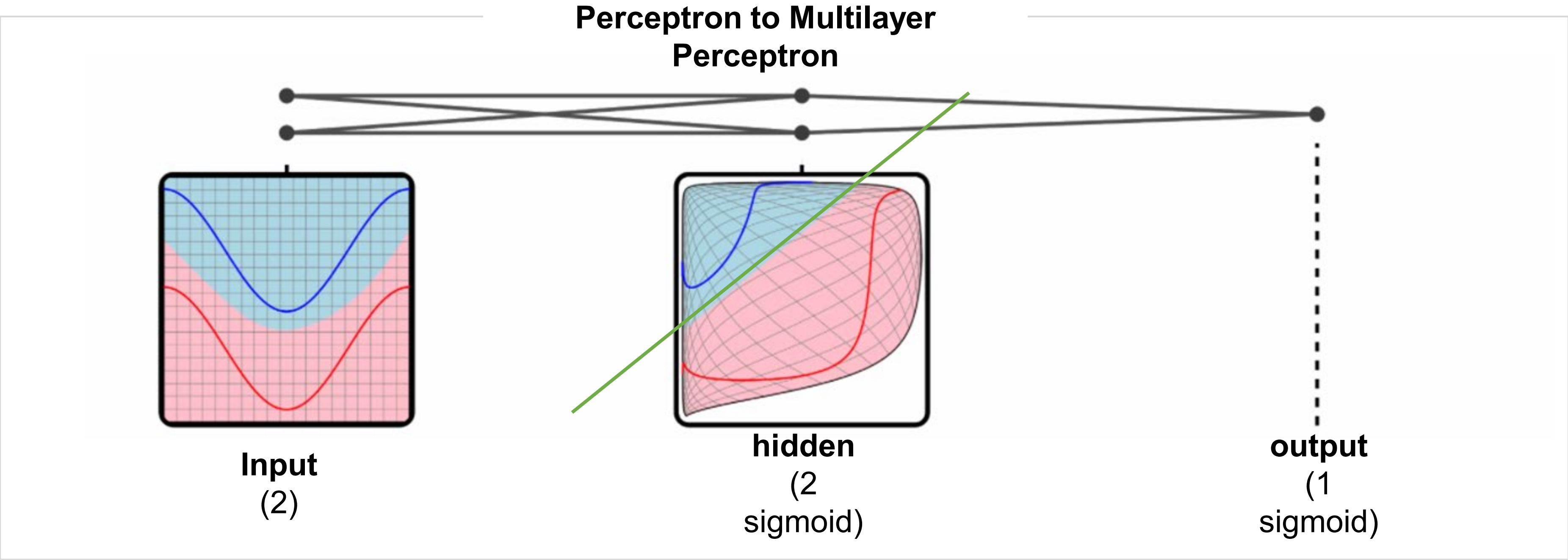
- Modify the coordinate plane itself.
- To solve the XOR, you need to run two perceptrons simultaneously.
- To make this possible, a hidden layer is necessary.

Perceptron to Multilayer Perceptron



Solution for the XOR problem

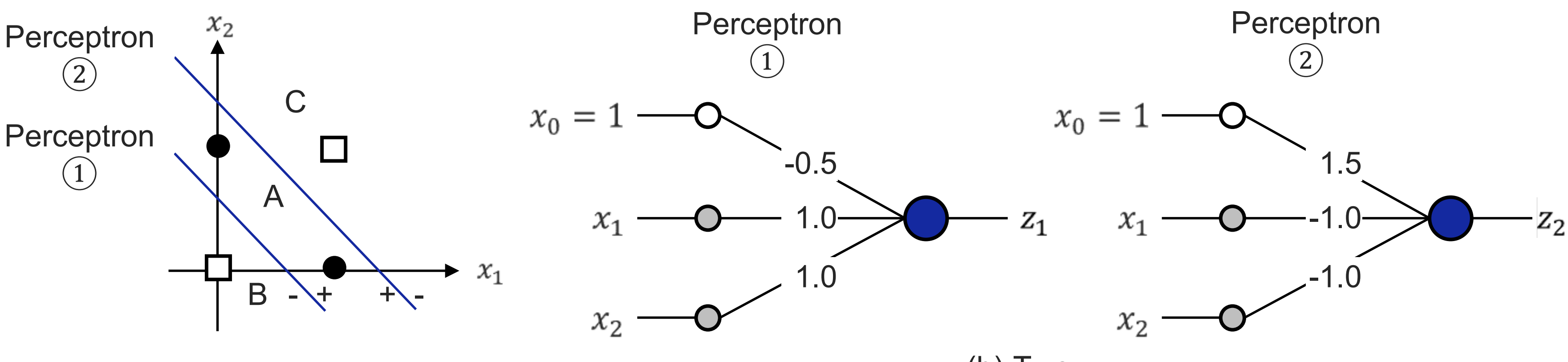
- If you divide the plane into sections of blue and red with regards to the inputs, no line can separate the two. However, if a hidden layer warps the plane, the curve line separating the two sections becomes a straight line.



Solution for the XOR problem

Solving the XOR with 2 perceptrons

- If perceptron ① and perceptron ② is both +1, the unit is ● . If not, □.



(a) Separating the plane by two perceptrons

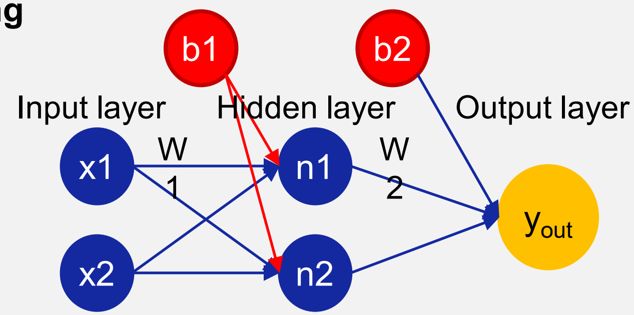
(b) Two perceptrons

Solution for the XOR problem

Solution for the XOR problem

- Solve the XOR problem by applying multilayer perceptron in Excel.

Solving XOR



Sigmoid function

=1/(1+exp(-x))

x1	x2
0	0
0	1
1	0
1	1

W1
-2
2
-2
2

W2
1
1

n1 After running sigmoid function		
-1	0.268941	0
1	0.731059	1
1	0.731059	1
3	0.952574	1

n2		
-1	0.268941	0
1	0.731059	1
1	0.731059	1
3	0.952574	1

output
0
1
1
0

Desired value
0
1
1
0

b1
3
-1

b2
-1

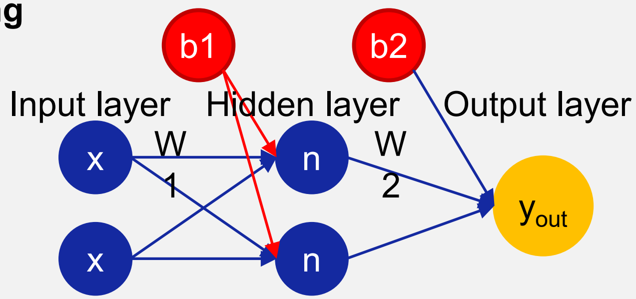
x1	x2	or	and	nand
0	0	0	0	1
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

or	nand	
0	1	0
1	1	1
1	1	1
1	0	0

Solution for the XOR problem

- Compute the XOR by applying multilayer perceptron in Excel.

Solving XOR



x1	x2
0	0
0	1
1	0
1	1

W1
-2
2
-2
2

b1
3
-1

W2
1
1

b2
-1

n1	After running sigmoid function	
=C12*$F$12+D12*$F$14+$F$18	=1/(1+EXP(-K12))	1
=C13*$F$12+D13*$F$14+$F$18	=1/(1+EXP(-K13))	1
=C14*$F$12+D14*$F$14+$F$18	=1/(1+EXP(-K14))	1

n2		
=C12*$F$13+D12*$F$15+$F$19	=1/(1+EXP(-O12))	0
=C13*$F$13+D13*$F$15+$F$19	=1/(1+EXP(-O13))	1
=C14*$F$13+D14*$F$15+$F$19	=1/(1+EXP(-O14))	1
=C15*$F$13+D14*$F$15+$F$19	=1/(1+EXP(-O15))	1

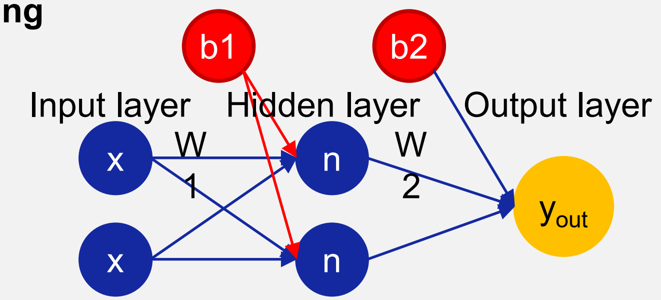
x1	x2	or	and	nand
0	0	0	0	1
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

or	d	
0	1	0
1	1	1
1	1	1
1	0	0

Solution for the XOR problem

- Compute the XOR by applying multilayer perceptron in Excel.

Solving XOR



x1	x2
0	0
0	1
1	0
1	1

W1
-2
2
-2
2

W2
1
1

b1
3
-1

b2
-1

n1 After running sigmoid function		
=C12*$F$12+D12*$F$14+$F$18	=1/(1+EXP(-K12))	1
=C13*$F$12+D13*$F$14+$F$18	=1/(1+EXP(-K13))	1
=C14*$F$12+D14*$F$14+$F$18	=1/(1+EXP(-K14))	1

n2		
=C12*$F$13+D12*$F$15+$F$19	=1/(1+EXP(-O12))	0
=C13*$F$13+D13*$F$15+$F$19	=1/(1+EXP(-O13))	1
=C14*$F$13+D14*$F$15+$F$19	=1/(1+EXP(-O14))	1
=C15*$F$13+D14*$F$15+$F$19	=1/(1+EXP(-O15))	1

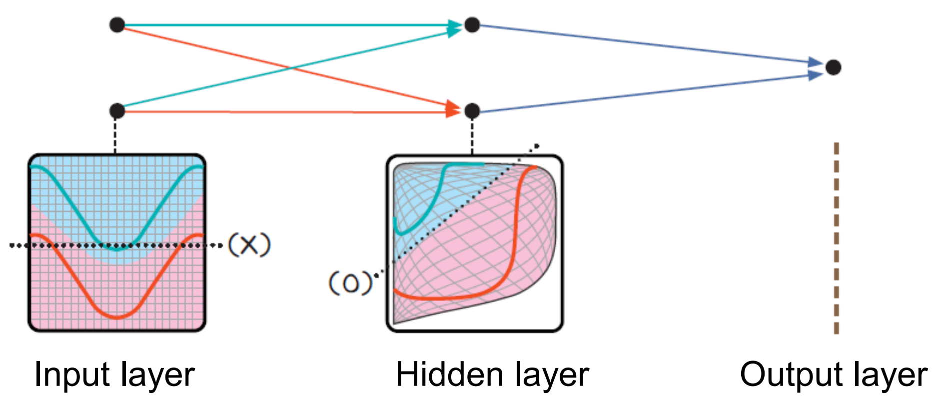
x1	x2	or	and	nand
0	0	0	0	1
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

or	d	
0	1	0
1	1	1
1	1	1
1	0	0

Solution for the XOR problem

- The hidden layer warps the coordinate plane.

x1	x2	or	and	nand	or	nand	xor
0	0	0	0	1	0	1	0
0	1	1	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	1	1	0	1	0	0



- Hidden layer warps the plane. (https://goo.gl/8qEGHD for more reference)

『Perceptrons』 by Minsky

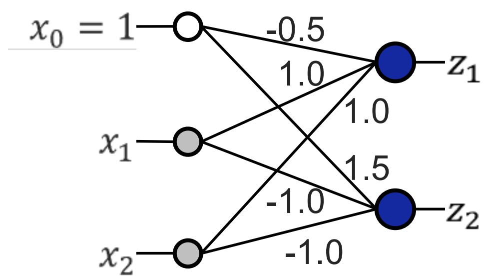
- Minsky pointed out the limitations of the perceptron and suggested a multilayer structure as a solution. The level of technology at the time was not advanced enough to materialize his idea.
- In 1974, Paul Werbos proposed backpropagation algorithm in his doctoral thesis.
- In 1986, Rumelhart established the multilayer theory in his book 『Parallel Distributed Processing』 and revived interest on the neural network.

Central Idea to the Multilayer Perceptron

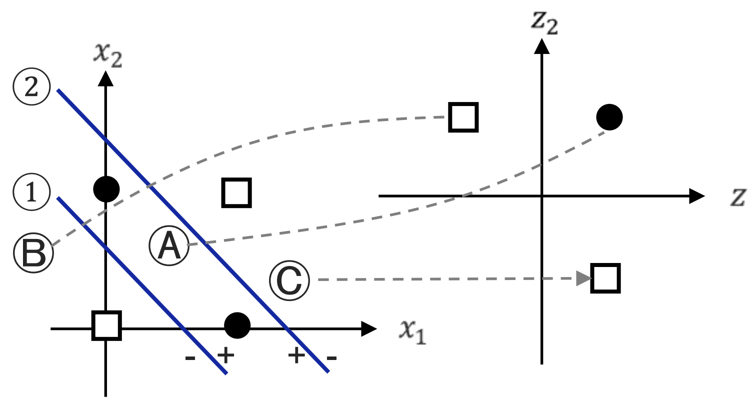
- Insert a hidden layer. It transforms a given feature space into a new one much favorable for classification.
- Use a sigmoid activation function. Perceptron used a step function as an activation function.
  This function is categorized as a hard decision decoder. However, a multilayer perceptron utilizes sigmoid function, which is a soft decision decoder, as an activation function. In soft-decision decoding, the output is continuous, and indicates confidence. This enables a more flexible decision making.
- Use backpropogation algorithm. Multilayer perceptron consists of many layers in consecutive order. Thus, backpropogation algorithm, which progresses in reverse order, and computes gradients and renews weights in each layer, is effective.

Central Idea to the Multilayer Perceptron

- Parallel combine two perceptrons
  - Transform a space $x = (x_1, x_2)^T$ into a new feature space $z = (z_1, z_2)^T$
  - The new feature space z is linear separable.



(a) Parallel combine two perceptrons



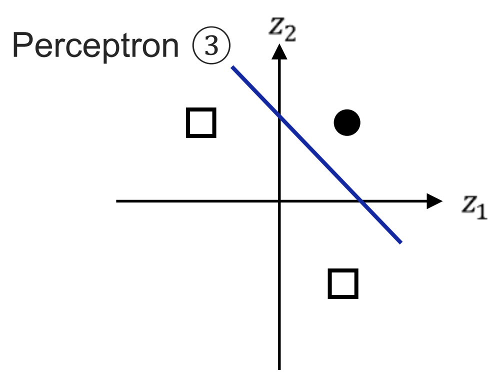
(b) Transform a feature space x into a new feature space z

Transformation of a feature space

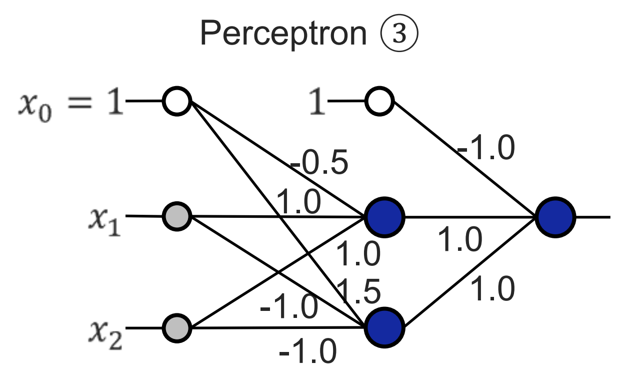
- Equivalent to a human-being manually executing feature learning

Central Idea to the Multilayer Perceptron

- If you serial combine a single perceptron
  - In feature space z, if you serial combine perceptron 3 which performs linear classification, it becomes a **multilayer perceptron**.



(a) Separation in a new feature space

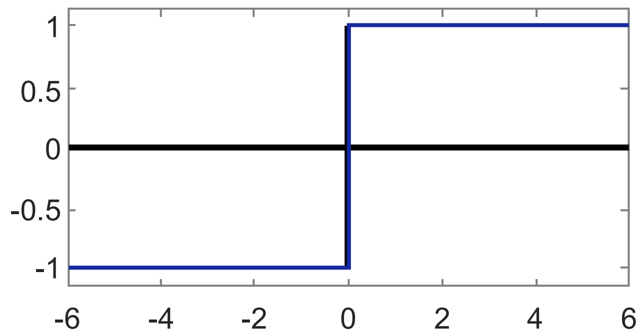


(b) Multilayer perceptron that combines three perceptrons

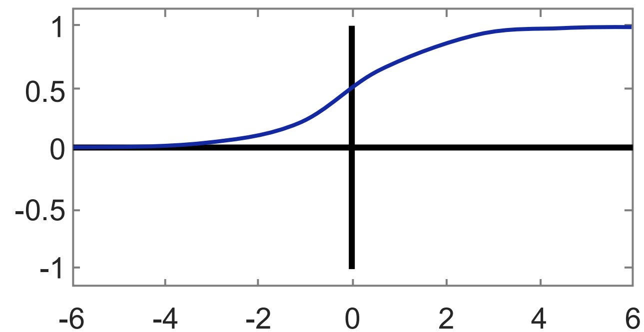
Multilayer Perceptron

Central Idea to the Multilayer Perceptron

- A threshold function is a hard-decision decoder (transforms an area into a dot). Other activation functions are soft-decision decoders (transforms an area into an area).



(a) Threshold function



(b) Logistic sigmoid

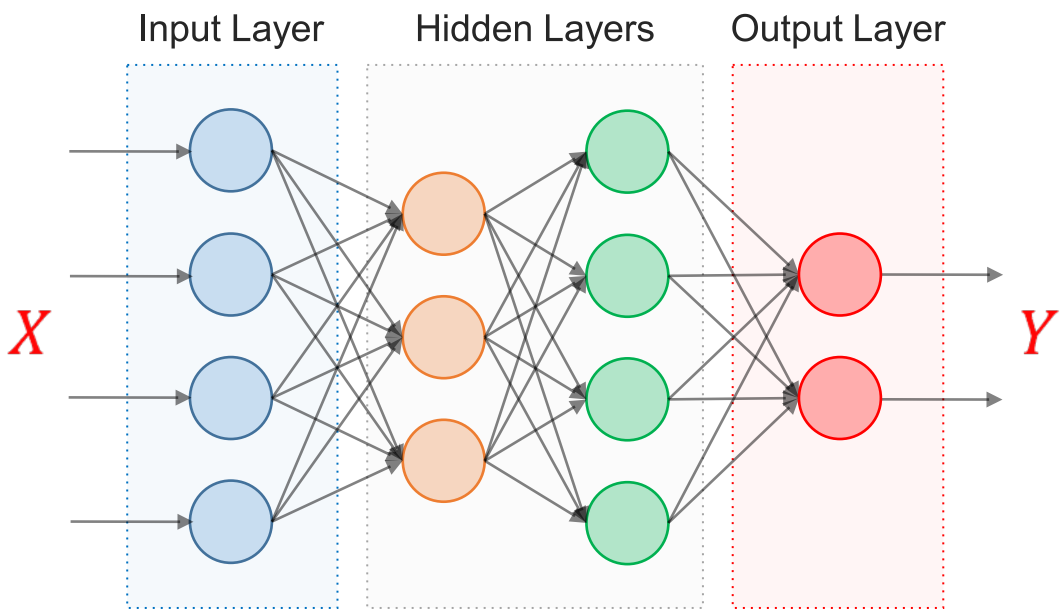
Activation function used in neural network

Unit 8a.

Basics of Neural Network

Artificial Neural Network (ANN)

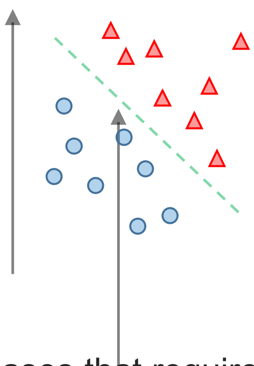
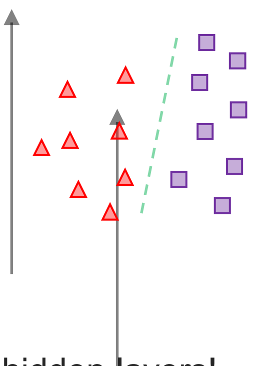
About the artificial neural network (ANN):



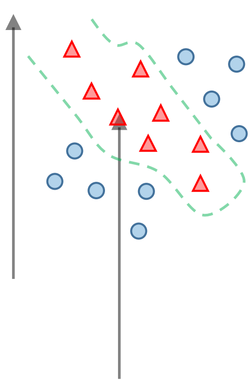
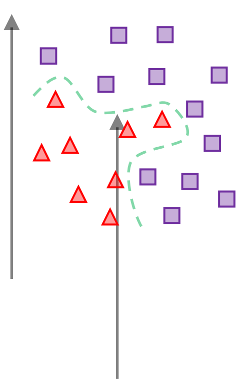
- Mimics the neural connections of a biological brain.
- There can be several hidden layers.

Why Artificial Neural Network (ANN)?

- For a logistic regression (even for the multi-class variant), the decision boundaries are linear.

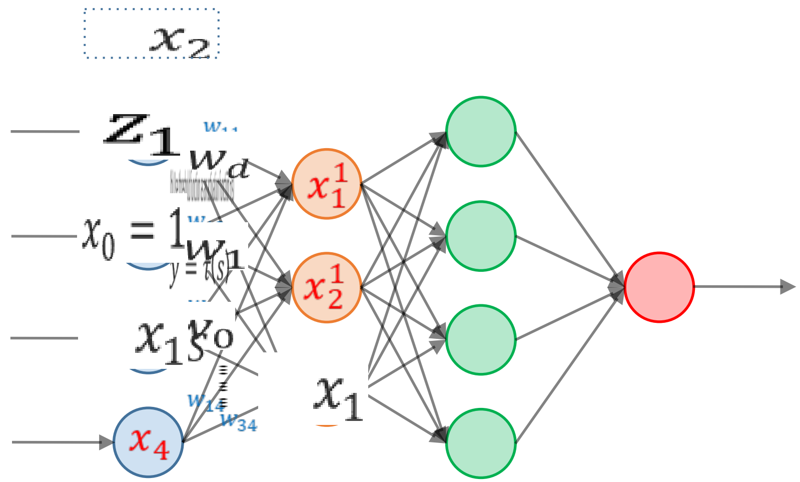
- How about the cases that require non-linear decision boundaries? ⇒ ANN with hidden layers!

ANN training: forward propagation.

1) When the values of X_i are given, apply the weights and propagate the signal forward to the next layer.

Example

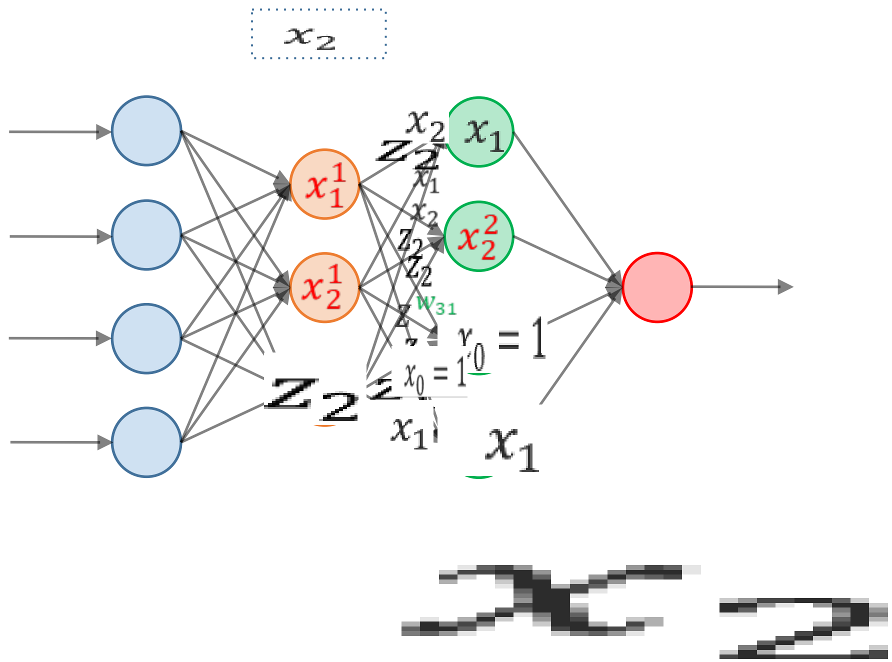


$$\begin{bmatrix} x_1^1 \\ x_2^1 \\ x_3^1 \end{bmatrix} = Activation\left(\begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \right)$$

- ANN training: forward propagation.

2) Propagate forward to the next layer.

Example



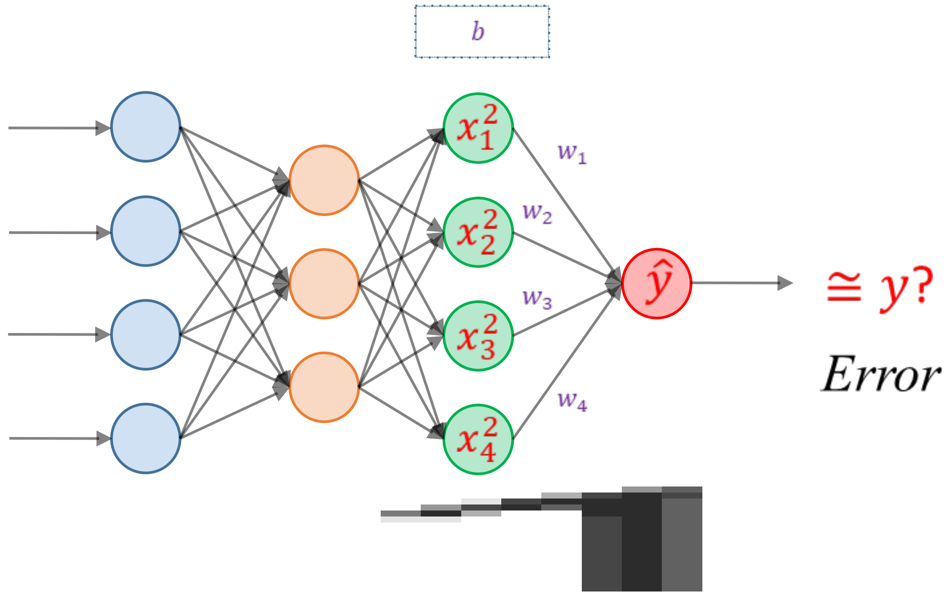
Unit 8a.

Basics of Neural Network

ANN training: forward propagation

- Propagate all the way; the difference between the estimated value $\hat{y}$ and the true value y is the error.

Example

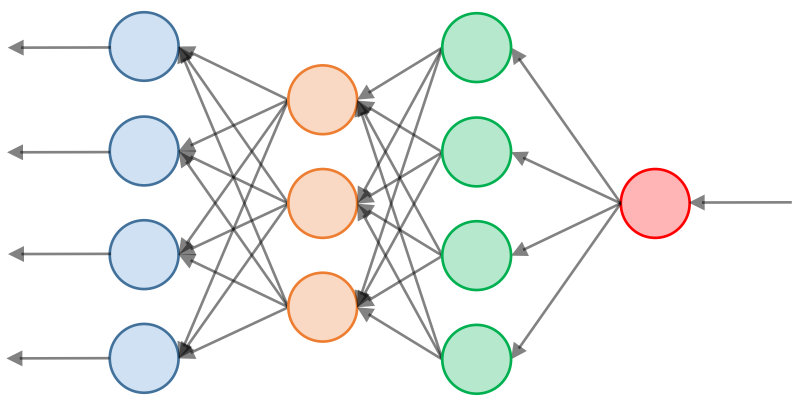


$$Error = y - \hat{y}$$

ANN training: backward propagation

- Propagate the error backward and update the parameters by gradient descent algorithm. Repeat from 1).

Example



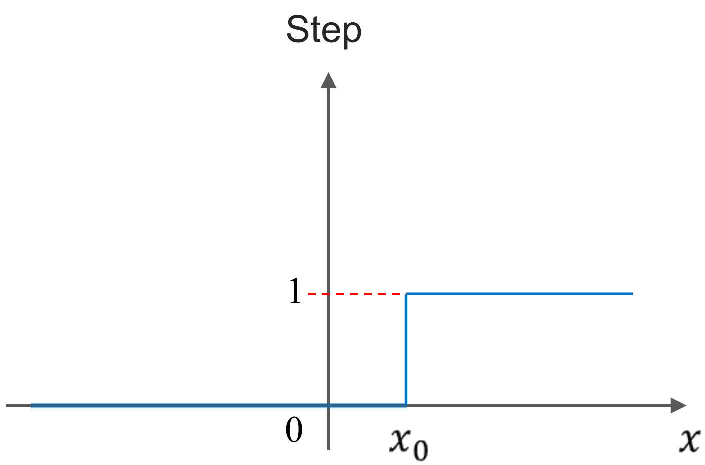
$$W \leftarrow W - \eta \, \nabla Loss$$
$$b \leftarrow b - \eta \, \nabla Loss$$

Unit 8a.

Basics of Neural Network

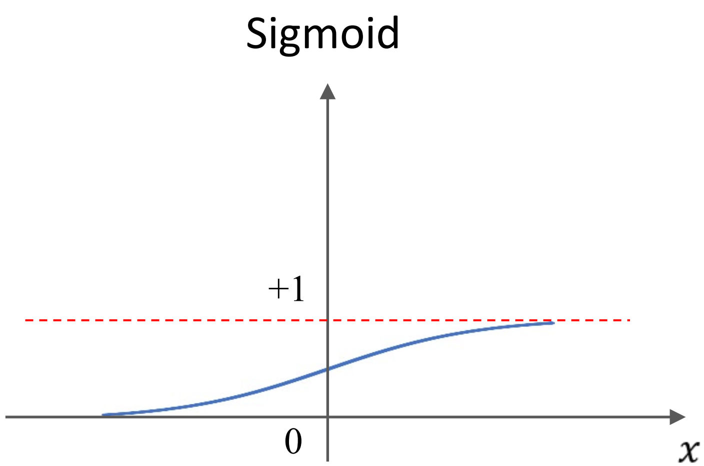
Activation Function

- Activation function: Linear threshold or Step.
  - $Step(x) = \theta(x - x_0))$
  - An activation function that was frequently used in the early days of neural network is the step function.



Activation function: Sigmoid $\sigma(x)$.

- $\sigma(x) = \frac{1}{1+e^{-x}}$
  - The Sigmoid is another widely used activation function. Note that the output value ranges between 0 and 1.

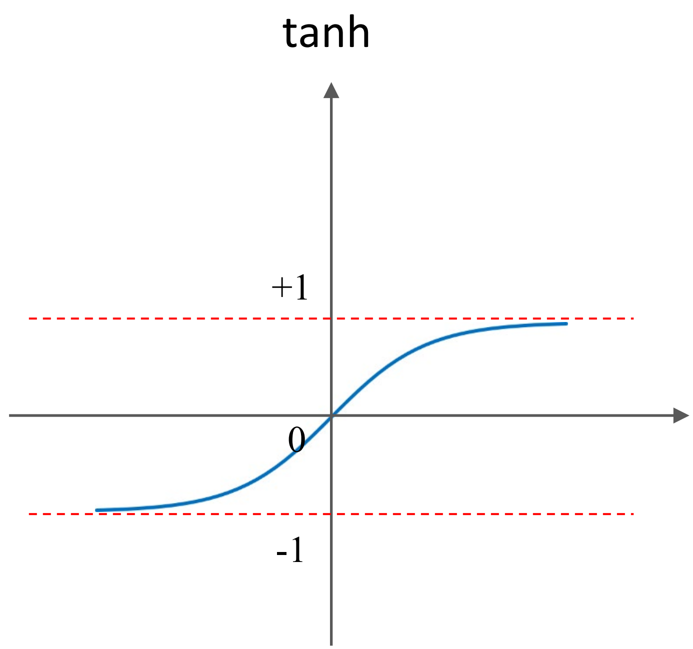


- When we have more than two classes, it can be generalized to the "Softmax".

8a.5. Activation Function

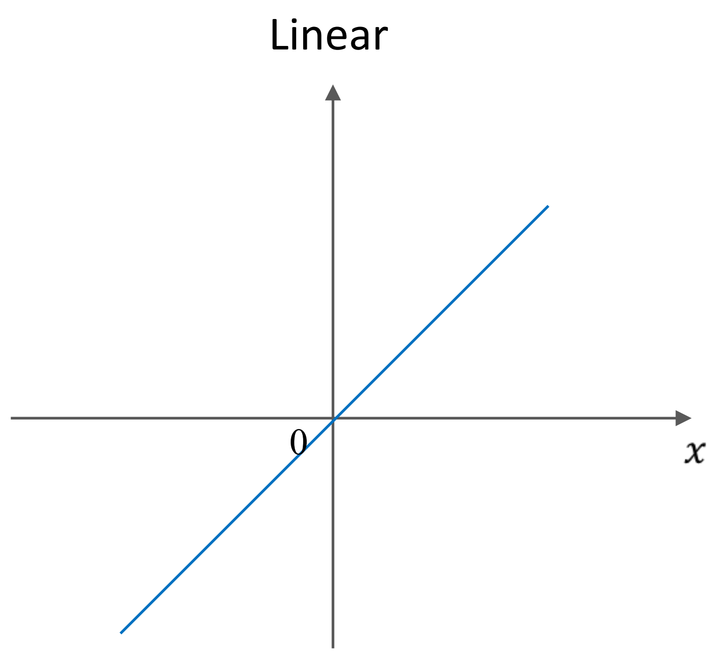
Activation function: Hyperbolic tangent or tanh.

- $tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
- The hyperbolic tangent (tanh) looks similar to the Sigmoid. However, the output value ranges between -1 and +1.



Activation function: Linear.

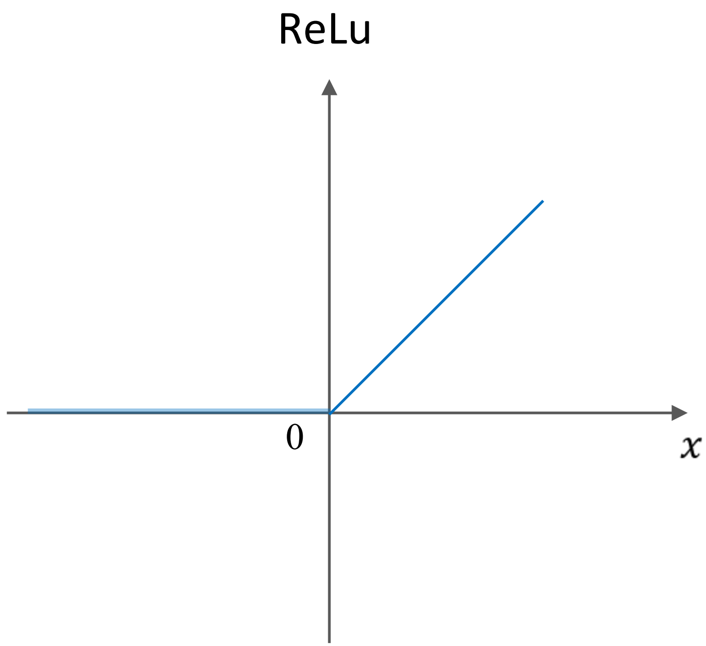
- $Linear(x) = x$
- The linear activation's output is the same as the input. The linear activation function is used in the regression.



8a.5. Activation Function

Activation function: ReLu (Rectifier Linear Unit).

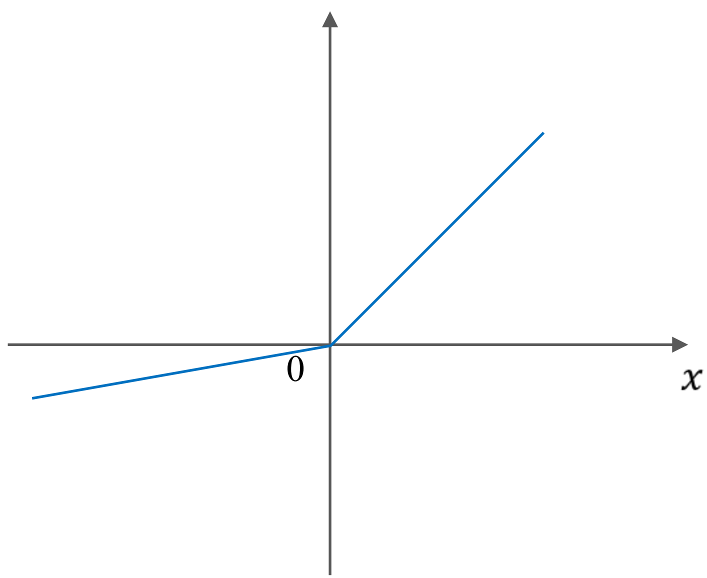
- $ReLu(x) = max(0, x)$
- ReLu is another commonly used activation function. It can be expressed in a simple functional form.



Activation function: Leaky ReLu.

- $Leaky\ ReLu(x) = max(\alpha x, x)\ \ where\ \alpha \in (0,1)$
- Leaky ReLu is a slightly "tweaked" version of ReLu.

Leaky ReLu



8a.5. Activation Function

Activation function:

- This is a summary of some of the most commonly used activation functions.
- Softmax is often used at the output layer of neural networks that do classification.

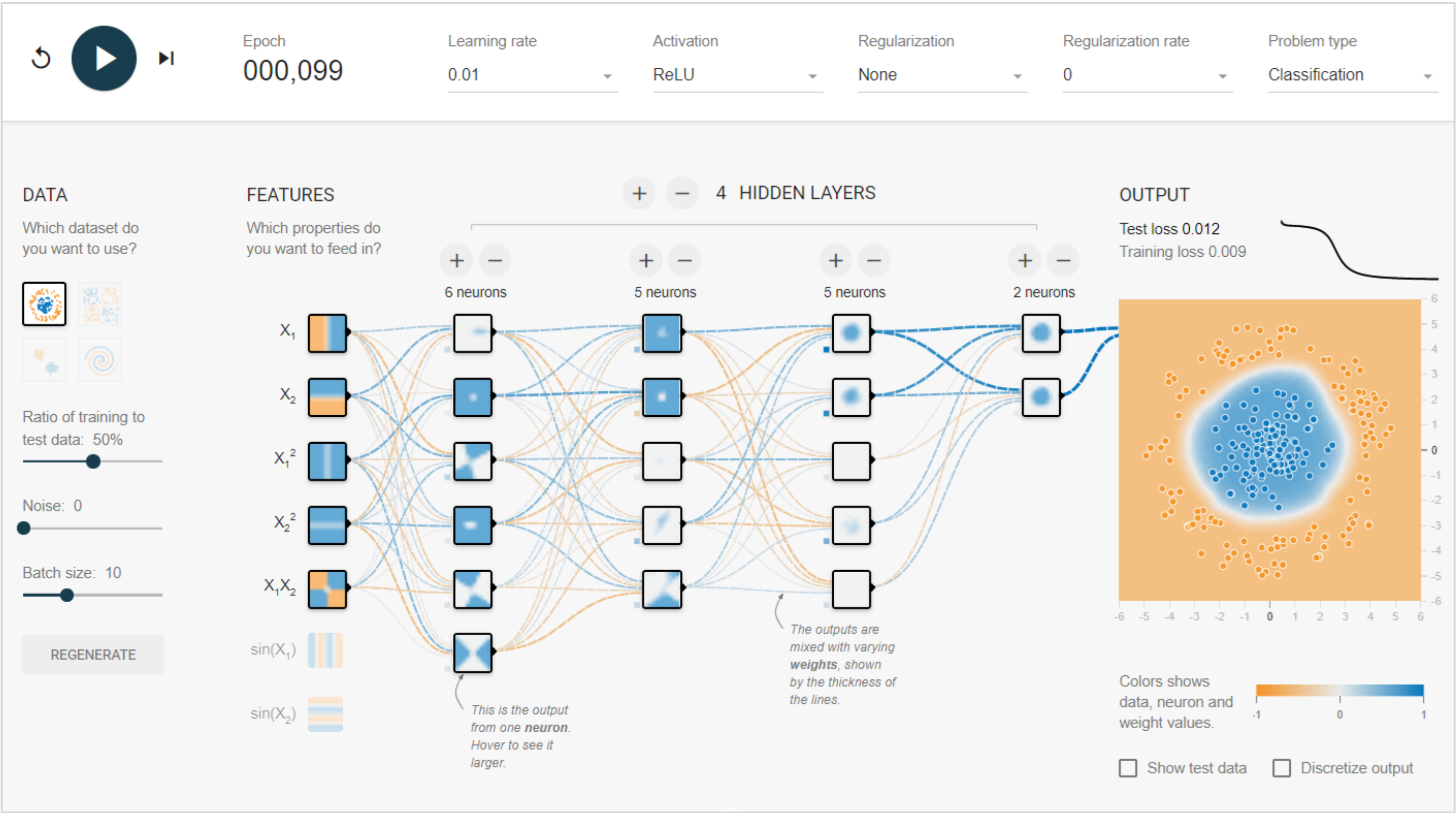
Name	Formula	TensorFlow
Step		
Sigmoid		tf.math.sigmoid(x)
Tanh		tf.math.tanh(x)
ReLu		tf.nn.relu(x)
Softmax		tf.nn.softmax(x)

Unit 8a.

Basics of Neural Network

Artificial neural network: a recommended site

- Access the website below to check artificial neural network.



Website: https://playground.tensorflow.org

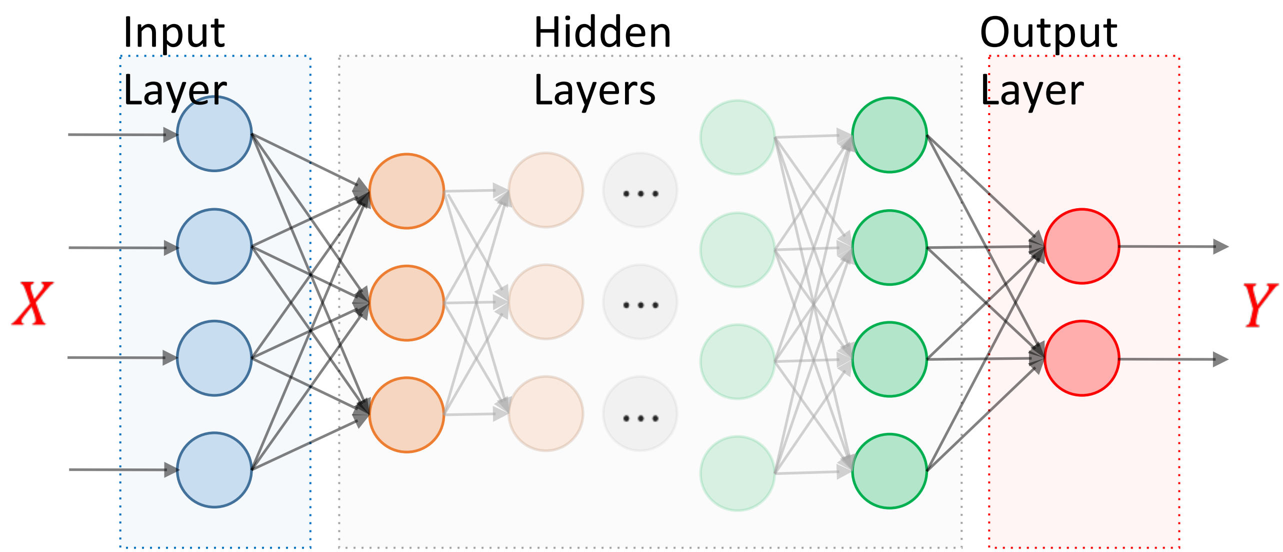
Artificial neural network: a recommended site

- The GUI of the website (https://playground.tensorflow.org) is intuitive.
- Select data from the left side panel in the website.
- Adjust the features, layers and nodes.
- From the top drop down menus, customize the learning rate, activation function, regularization, etc.
- Press the Play button to start training ⇒ training loss decreases as the training progresses.
- The thickness of an edge line represents the weight that can also be changed manually.

8a.6. Deep Learning

About the Deep Neural Network (DNN):

- There may be several hidden layers.
- With more hidden layers, we have a higher degree of abstraction.



Unit 8a.

Basics of Neural Network

- 8a.1. Understanding Perceptron with Excel
- 8a.2. Understanding Multilayer Perceptron with Excel
- 8a.3. From Multilayer Perceptron to Deep Learning
- 8a.4. Error Backpropagation and Gradient Descent to Reduce Errors
- 8a.5. Activation Function
- 8a.6. Deep Learning
- **8a.7. Problems with Deep Learning and the Solutions**

8a.7. Problems of Deep Learning and the Solution

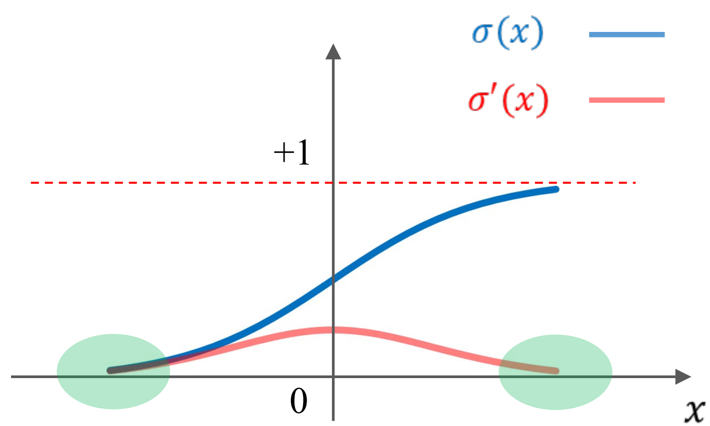
Common problems with the DNN:

- As the 'signal' propagates through the layers, it can die out (vanish) or grow uncontrollably.

  a) We should avoid using the Sigmoid $\sigma(x)$ as the activation function and use the ReLu instead.

  → The derivative $\sigma'(x)$ becomes very small as the absolute value of x or $|x|$ increases.

  → Small $\sigma'(x)$ may cause the so-called vanishing gradient problem.



As the ‘signal’ propagates through the layers, it can die out (vanish) or grow uncontrollably.

- As the ‘signal’ propagates through the layers, it can die out (vanish) or grow uncontrollably.

  b) We should initialize the weights such that the variance of the internal nodes is roughly constant.

  → Initializing as a constant such as 1 is not optimal.

  → Better to randomly initialize the weights with center at 0 and standard deviation $\sim\frac{1}{\sqrt{N_{nodes}}}$.(*)

  Here, N_{nodes} = average number of nodes in the neighboring layers (before and after).

- (*) Refer to the research paper at http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf

8a.7. Problems of Deep Learning and the Solution

Common problems with the DNN:

- Also, DNNs are prone to the overfitting. To alleviate this problem we can do:

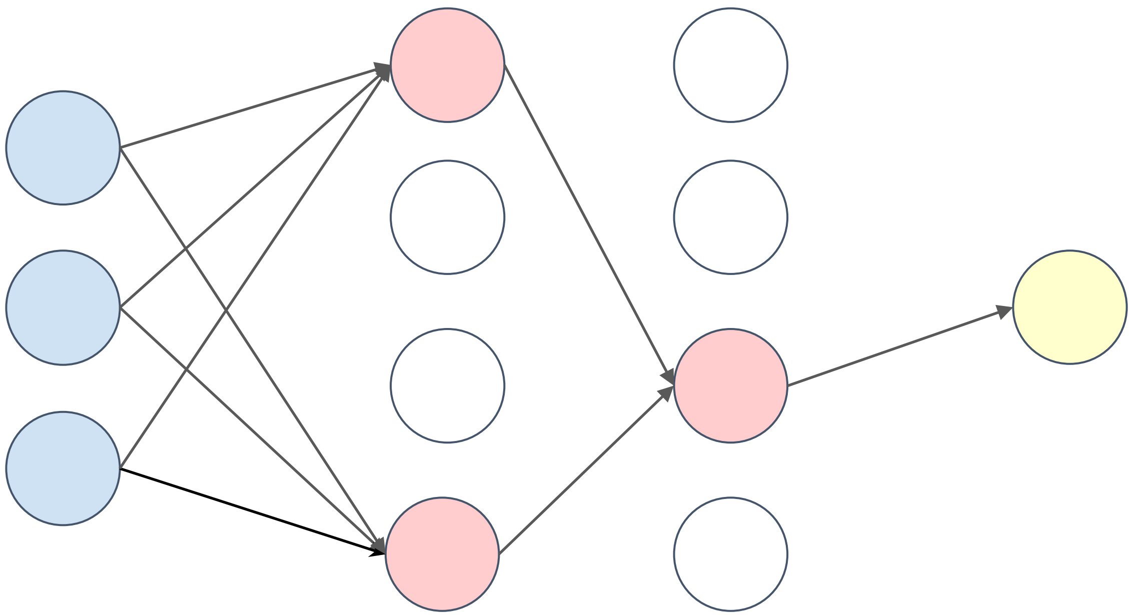
  a) Regularization of the weights: L1 or L2 regularization similar to the Lasso or Ridge.

  b) Dropout: randomly exclude certain nodes during the training step to avoid over dependency.

  c) Data augmentation: add noise, create new data by transformation, etc.

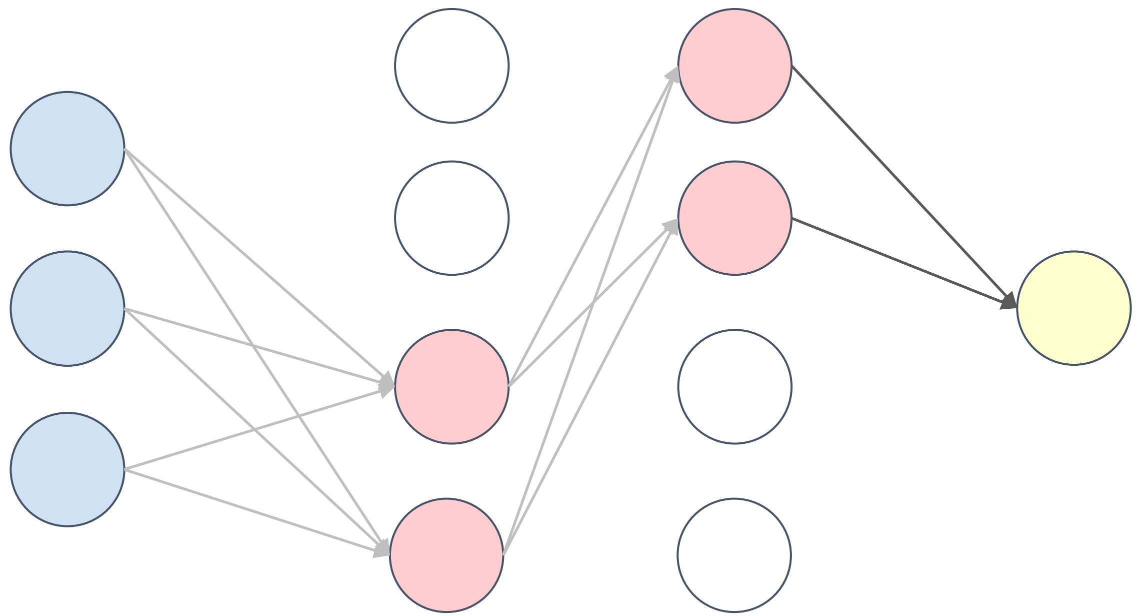
Dropout:

- Randomly exclude certain nodes during the training step to avoid over dependency.



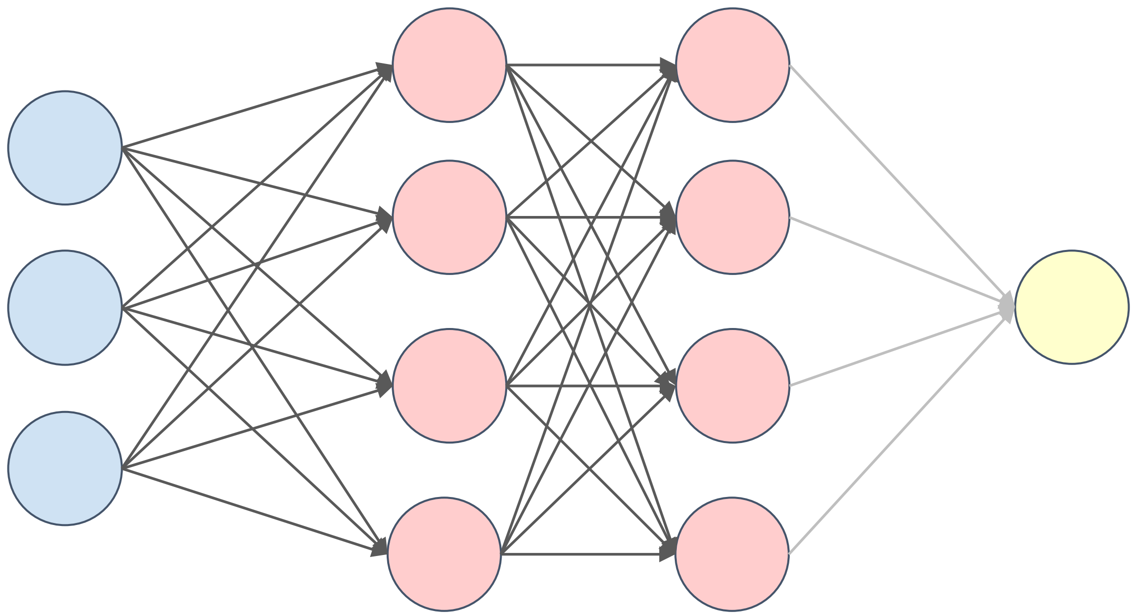
Dropout:

- Randomly exclude certain nodes during the training step to avoid over dependency.



Dropout:

- When predicting, use all the nodes and edges without excluding.



Coding Exercise #0514

Follow practice steps on 'ex_0514.ipynb' file

Coding Exercise #0515

Follow practice steps on 'ex_0515.ipynb' file

Coding Exercise #0516

Follow practice steps on 'ex_0516.ipynb' file

End of Document

Together for Tomorrow!
Enabling People

Education for Future Generations