FEED FORWARD NN

Asst.Prof.Dr.Supakit Nootyaskool
IT-KMITL

Learning Outcome

- Understand how to implement MLP in N:N and 1:N model
- Know under and over-fitting problems
- Have experience using iris data set
- Write code to train Iris data to nuralnet()

Topic

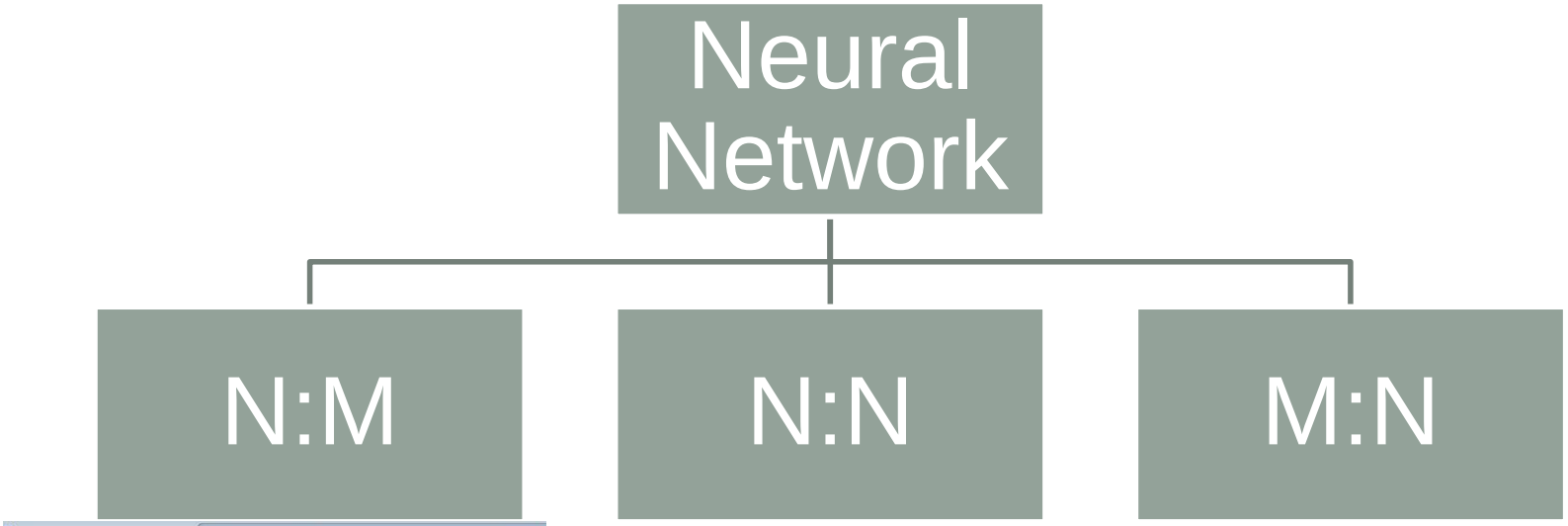
- Perceptron N:N model
- Perceptron 1:N model
- Input data adjustment
- Under and over-fitting
- Iris data set
- Train Iris data to nuralnet()

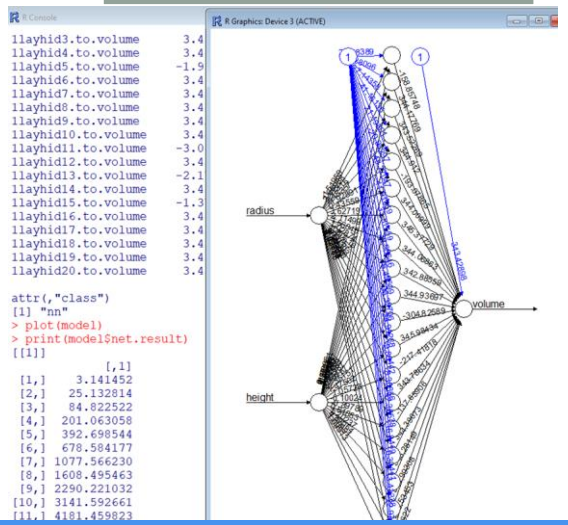
PERCEPTRON N:N, M:N, N:M

เพื่อให้เข้าใจการออกแบบ NN ในลักษณะที่แตกต่างออกไป

"คิดแตกต่างเป็นหนทางในการหาทางออก"

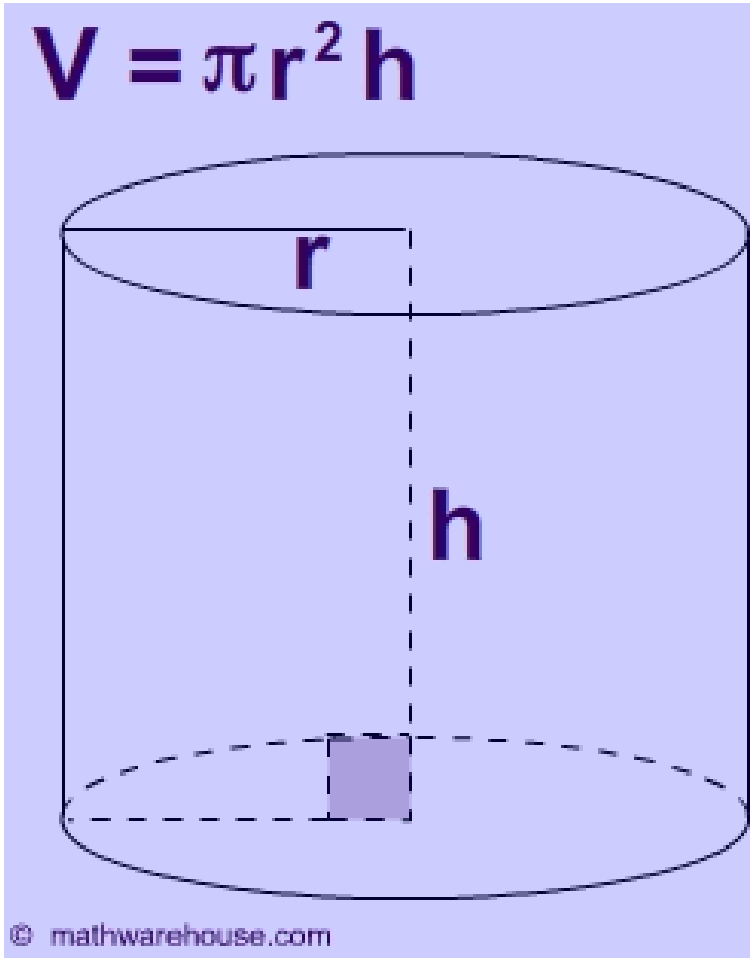
Models Neural Networks



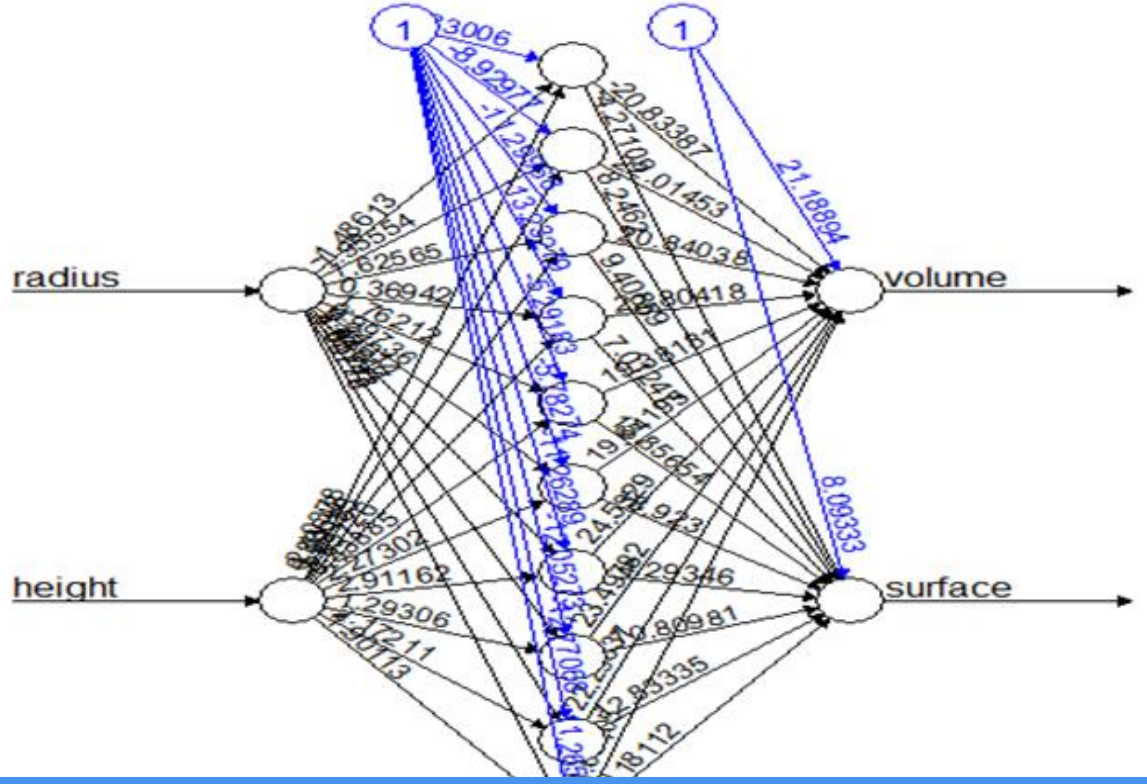


5.1 Perceptron with N:N

input = output



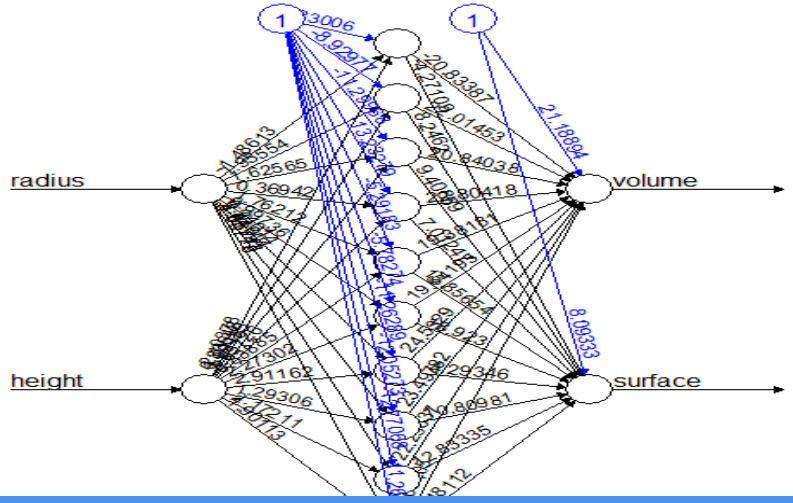
- Volume = pi * r * r * h
- Surface = 2 * pi *r *h



Activity 5.1 N:N Perceptron

```
########################################
#ACTIVITY 5-1 N:N PERCEPTRON
#VOLUME AND SURFACE OF CYLINDER CALCUATING
#WITH PERCEPTRON
########################################

library("neuralnet")

radius = 1:4
height = 1:4
volume = pi*radius*radius*height
surface = 2*pi*radius*height
datatrain = data.frame(radius,height,
        volume,
        surface)
datatrain
```

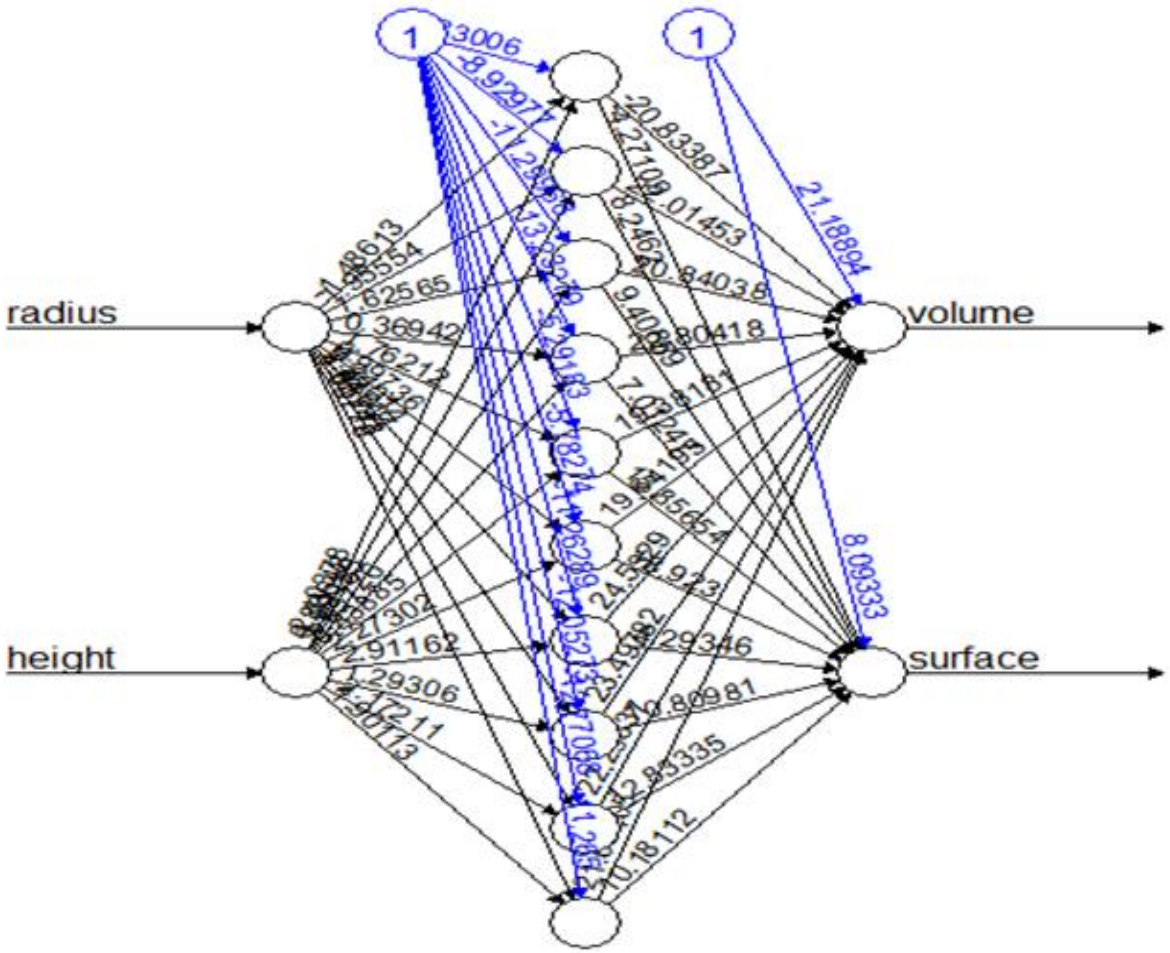
```
model <- neuralnet(
        volume+surface~radius+height,
        datatrain,
        hidden=10, ##<--Change here
        rep = 1,
        linear.output = TRUE)
print(model)
plot(model)
print(model$net.result)
```



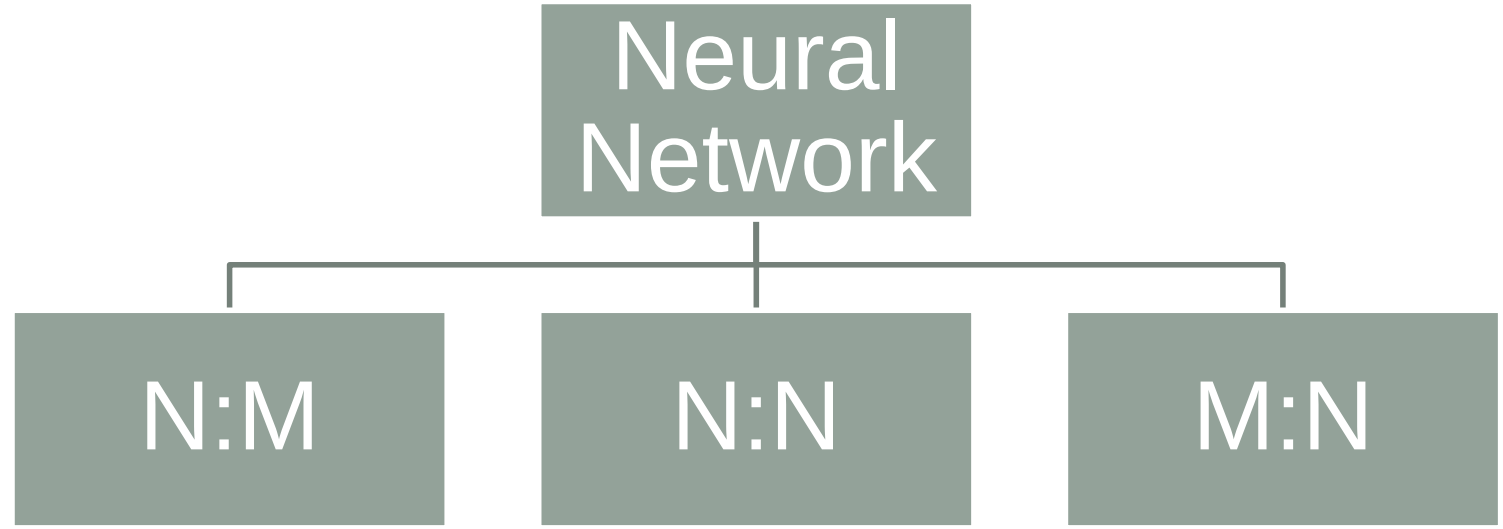
Activity 5.1 N:N Perceptron

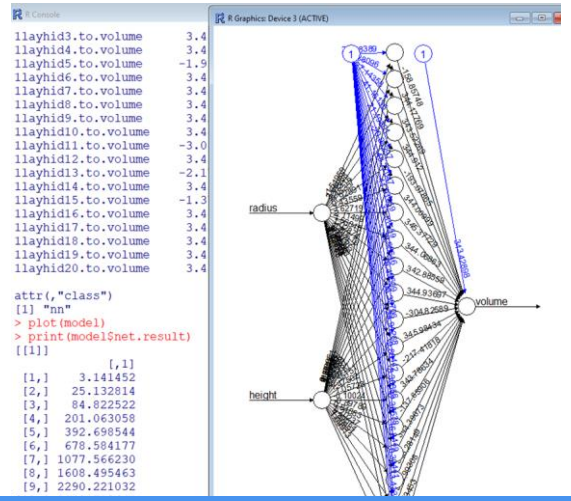
```
$data
  radius height      volume     surface
1      1      1    3.141593    6.283185
2      2      2   25.132741   25.132741
3      3      3   84.823002   56.548668
4      4      4  201.061930  100.530965
```

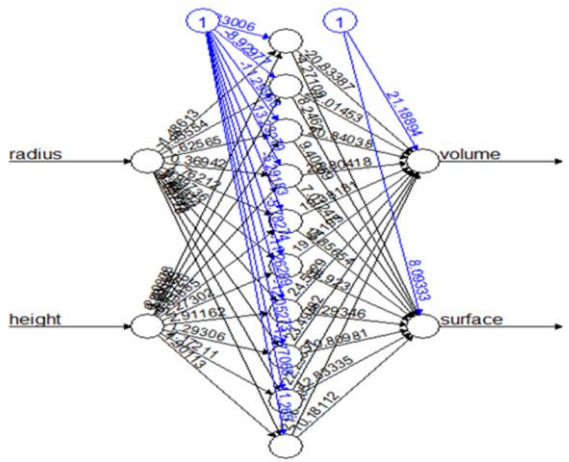
```
$net.result[[1]]
           [,1]        [,2]
[1,]    3.13088    6.293871
[2,]   25.14182   25.122653
[3,]   84.82135   56.550207
[4,]  201.06319  100.531784
```



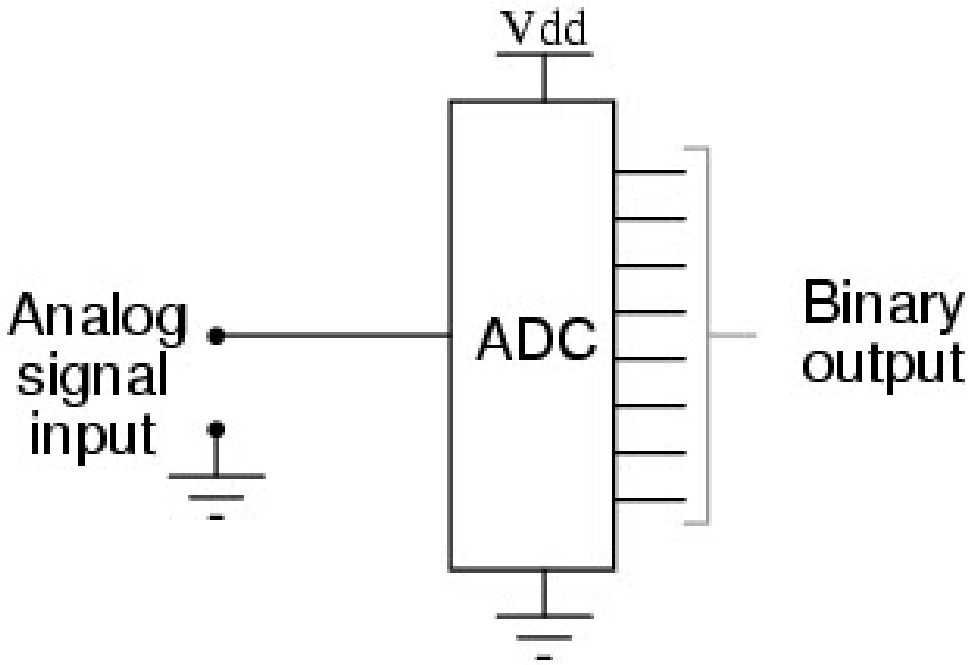
Models Neural Networks





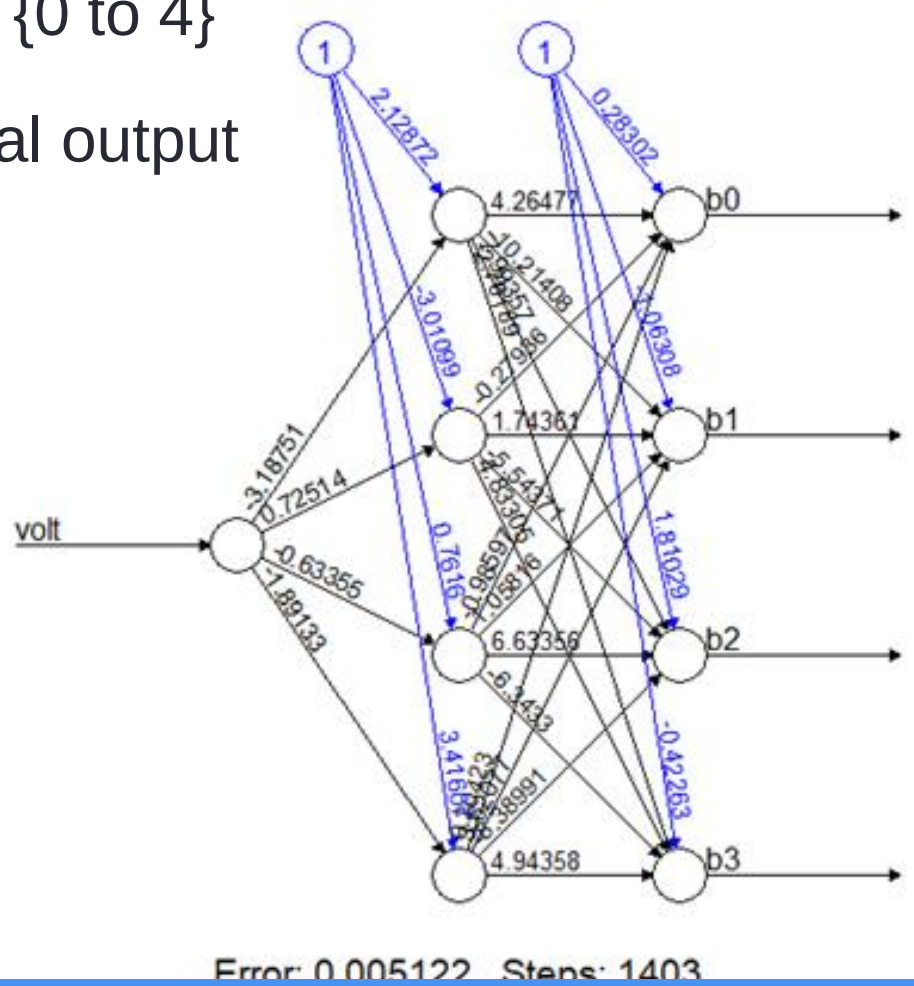


5.2 Perceptron 1:N model



แปลงสัญญาณ Analog → Digital

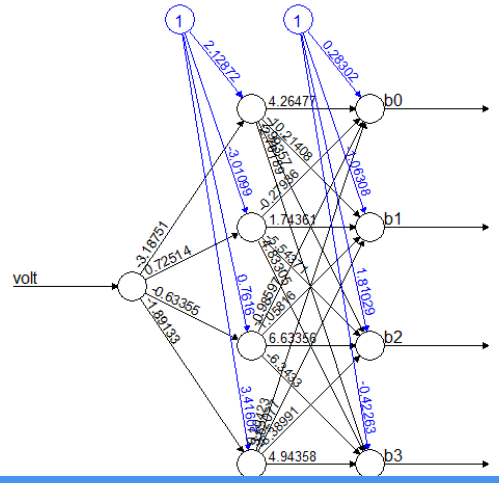
- voltage = {0 to 4}
- 4-bit digital output



Activity 5.2 1:N, Perceptron

```
##########################################
#ACTIVITY 5-2 1:N PERCEPTRON
#ANALOG TO DIGITAL CONVERTER WITH
#PERCEPTRON
##########################################

library("neuralnet")

volt = c(1,2,3,4)
b0 = c(1,0,0,0)
b1 = c(0,1,0,0)
b2 = c(0,0,1,0)
b3 = c(0,0,0,1)
datatrain = data.frame(volt,b0,b1,b2,b3)
datatrain
```

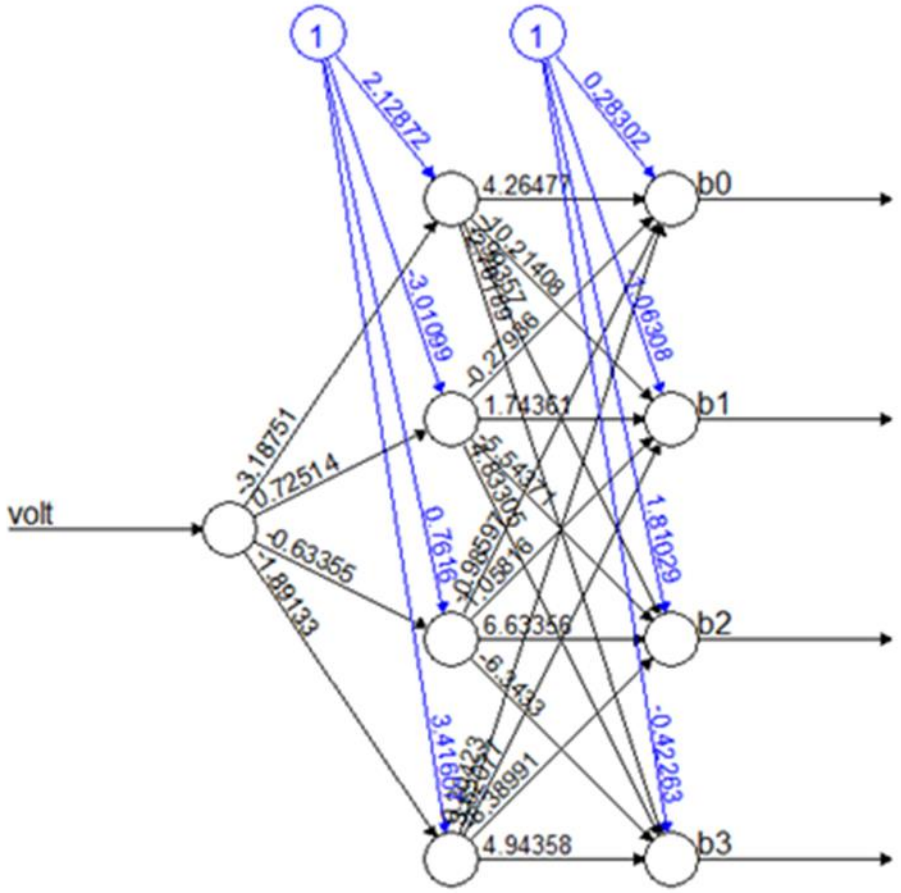
```
model <- neuralnet(
        b0+b1+b2+b3~volt,
        datatrain,
        hidden=4, ##<--Change here
        rep = 1,
        linear.output = TRUE)
print(model)
plot(model)
print(model$net.result)
```



Activity 5.2 1:N, Perceptron

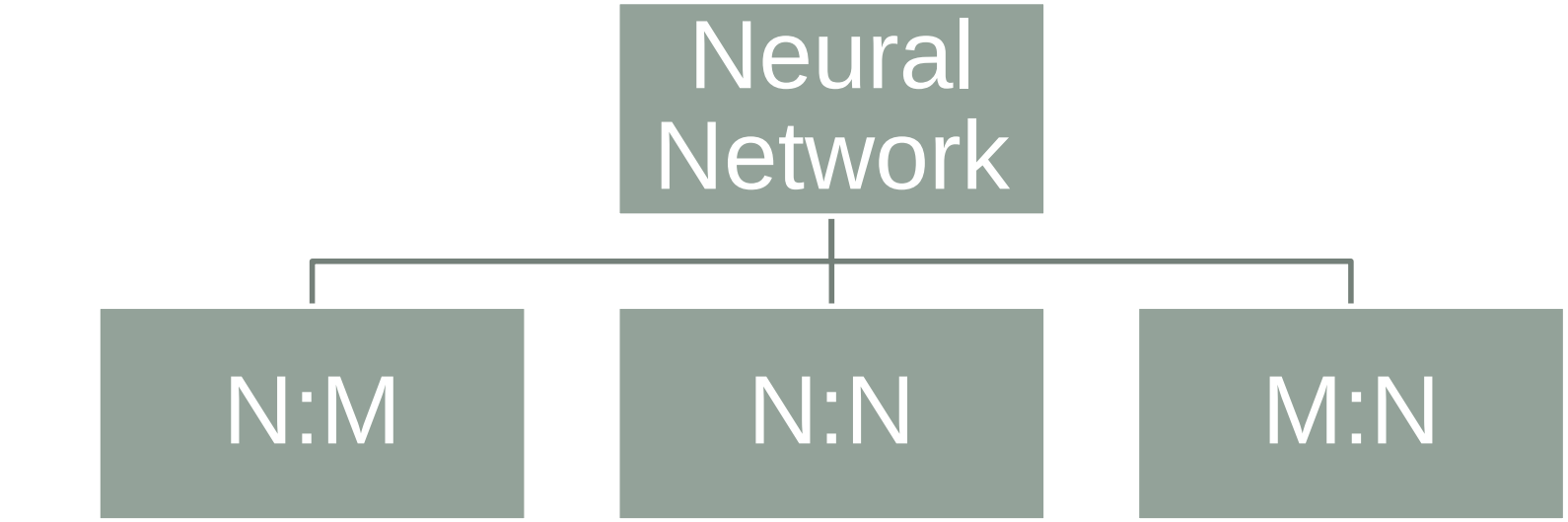
```
$data
  volt b0 b1 b2 b3
1    1  1  0  0  0
2    2  0  1  0  0
3    3  0  0  1  0
4    4  0  0  0  1
```

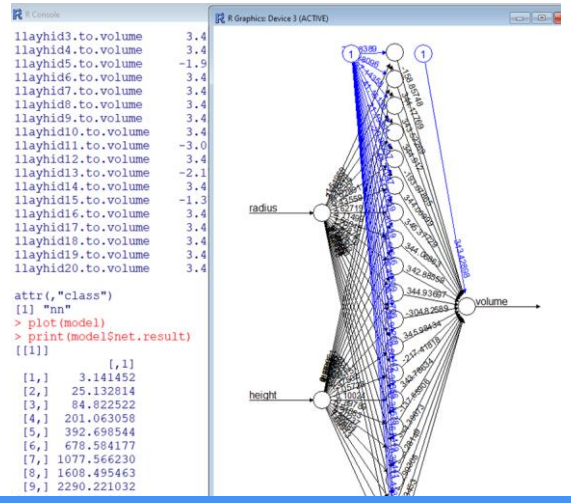
```
$net.result
$net.result[[1]]
             [,1]          [,2]          [,3]          [,4]
[1,]  1.00064948  0.001159281 -0.004748923 -0.02828780
[2,]  0.01309618  0.971998321  0.015997339  0.03412774
[3,] -0.02652515  0.080825292  0.958267428 -0.01105141
[4,]  0.01457693 -0.060783183  0.024395649  0.99534282
```

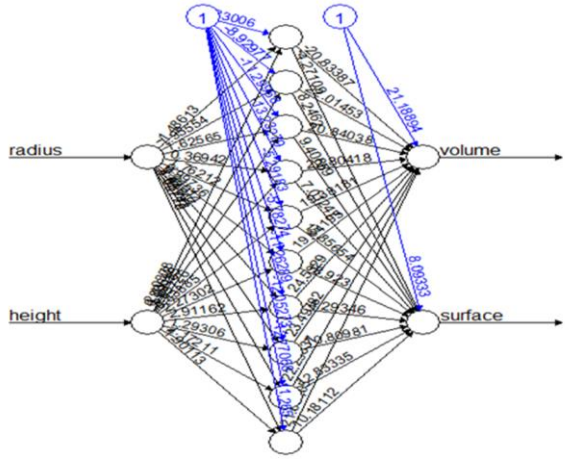


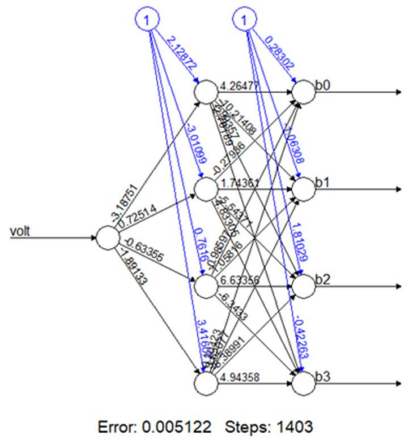
Error: 0.005122 Steps: 1403

Summary: Models Neural Networks









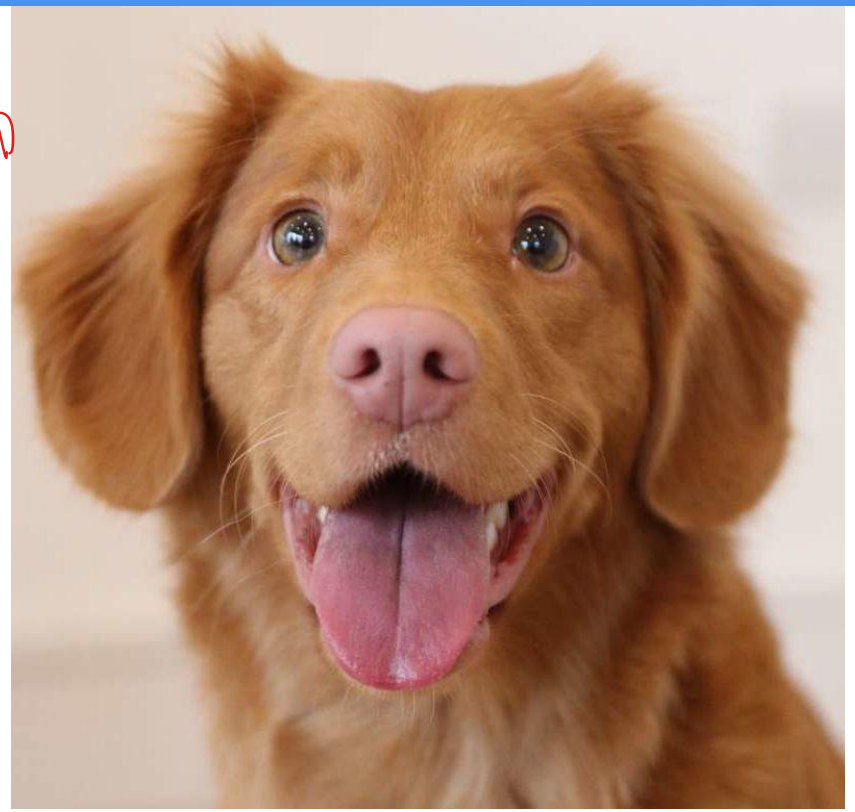
DATA SCALING

เพื่อให้สามารถเขียนโปรแกรมในการปรับขนาดข้อมูลได้

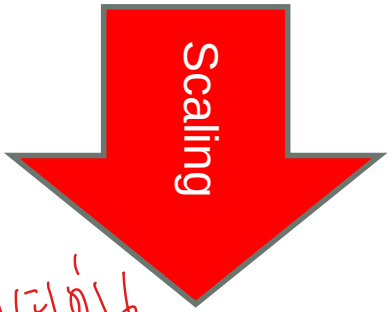
To be able to write a computer code for the adjustment size of the data.

Data scaling

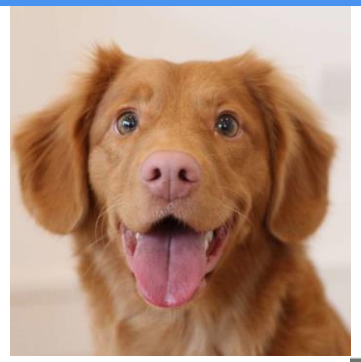
- Data scaling มีอีกชื่อ Normalization (ทรงกขนาด)
- ปรับขนาดโดยยังคง ระยะห่าง ข้อมูลเหมือนเดิม
- ช่วยให้ ลดระยะทางในการค้นหาของ ML
- แก้ปัญหา สอนช้าใน ML
- ต้อง scale down ไม่ใช่ scale up

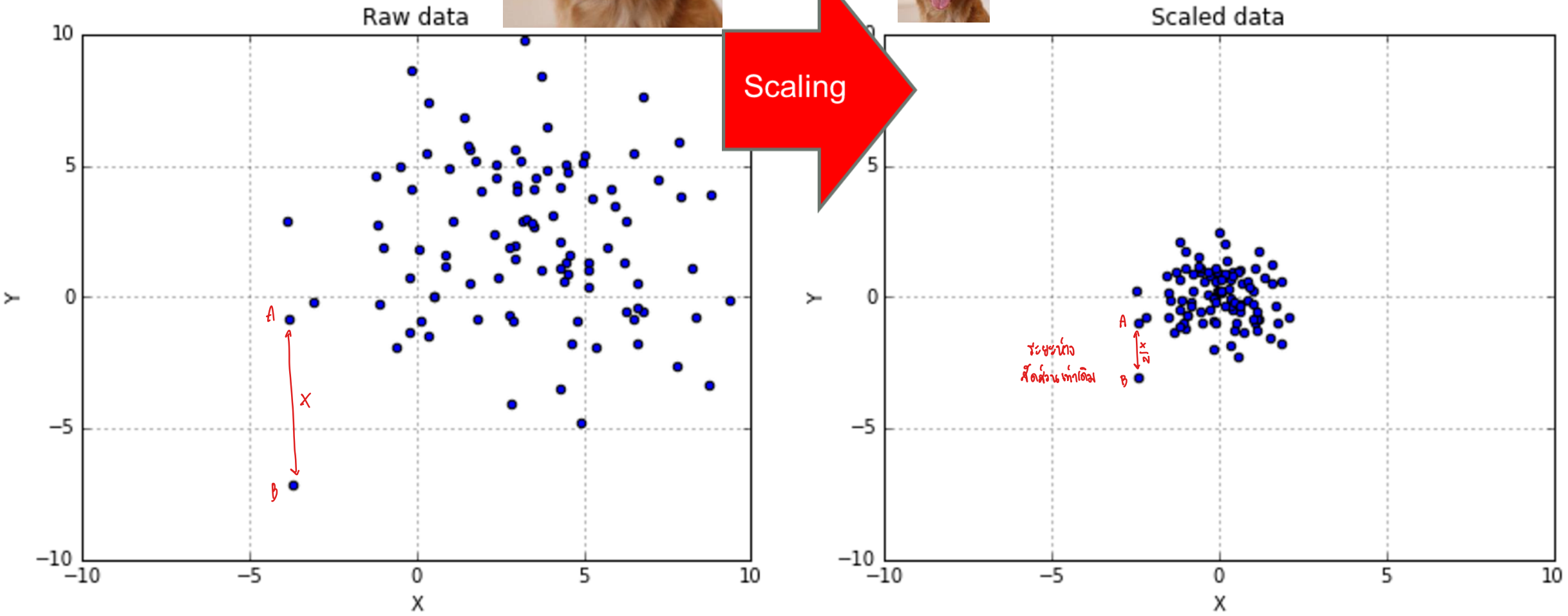


-100 to 100 → Scaling → -1 to 1



Scaling data





Data scaling method

Scaling methods

1. Change Position
จับบวกลบเลื่อน

± ↑ ↓

2. Division
จับข้อมูลหาร

3. Random pickup
มีเยอะสุ่มหยิบตัวแทน

ข้อมูลเยอะมาก

สุ่มหยิบบางตัว

4. Based on data characteristics
พิจารณาตามลักษณะ

มองว่าเป็นสมการอะไร

5.3.1 Z-score normalization

- ให้ค่าเฉลี่ยข้อมูลใหม่เป็น 0 $Z = \frac{X - \bar{X}}{SD}$
- ความห่างระหว่างสมาชิกเหมือนเดิม ต่างที่สัดส่วน
- ใช้สูตรนี้

```
> x = c(1,2,3,4)
> sd(x)
[1] 1.290994
> x = c(11,22,33,44)
> sd(x)
[1] 14.20094
```

$$z = (X - \mu) / \sigma$$

Standard Deviation
ค่าเบี่ยงเบนมาตราฐาน

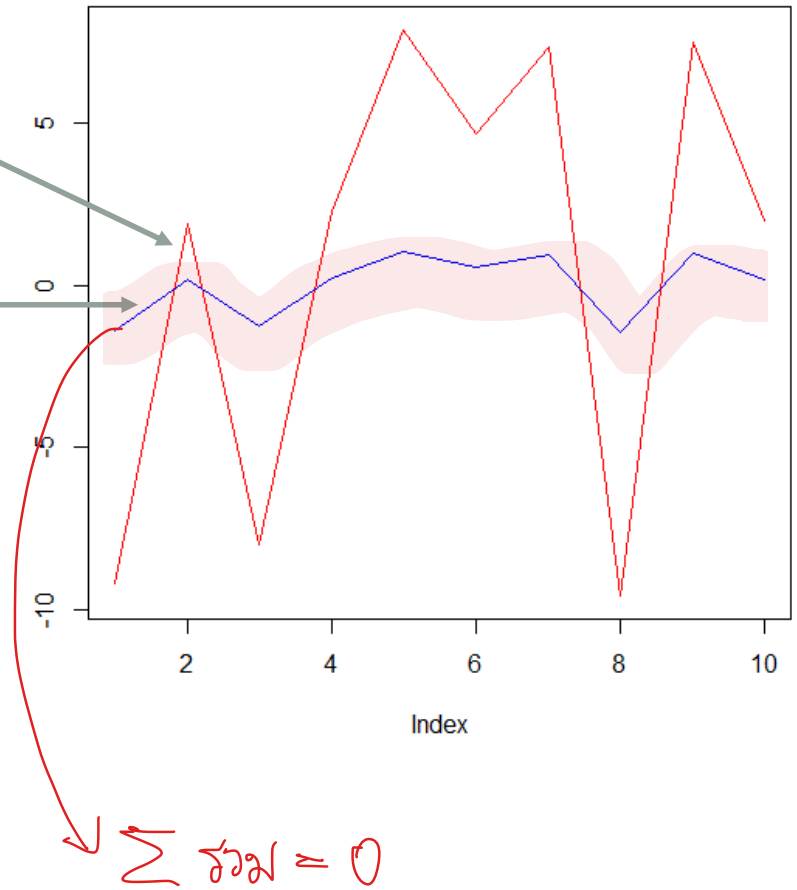
mean
ค่าเฉลี่ย

มีซีกบวกและลบอยู่

5.3.1 Test Z-score function

```
funcZscore = function(x)
{
        avg = mean(x)
        sd = sd(x)
        z = (x - avg) / sd
        return(z)
}
```

```
x1 = runif(10,-10,10)
print(x1)
x2 = funcZscore (x1)
print(x2)
plot(x1,type='l', col='red')
lines(x2,col='blue')
```



5.3.2 Min-max normalization

อยู่ ซึ่งบางทีมันหมด ไม่มีค่าลบ

Sigmoid

- ลดขนาดให้อยู่ในกรอบ 0 ถึง 1
- ความห่างระหว่างสมาชิกเหมือนเดิม ต่างที่สัดส่วน
- ใช้สูตรนี้

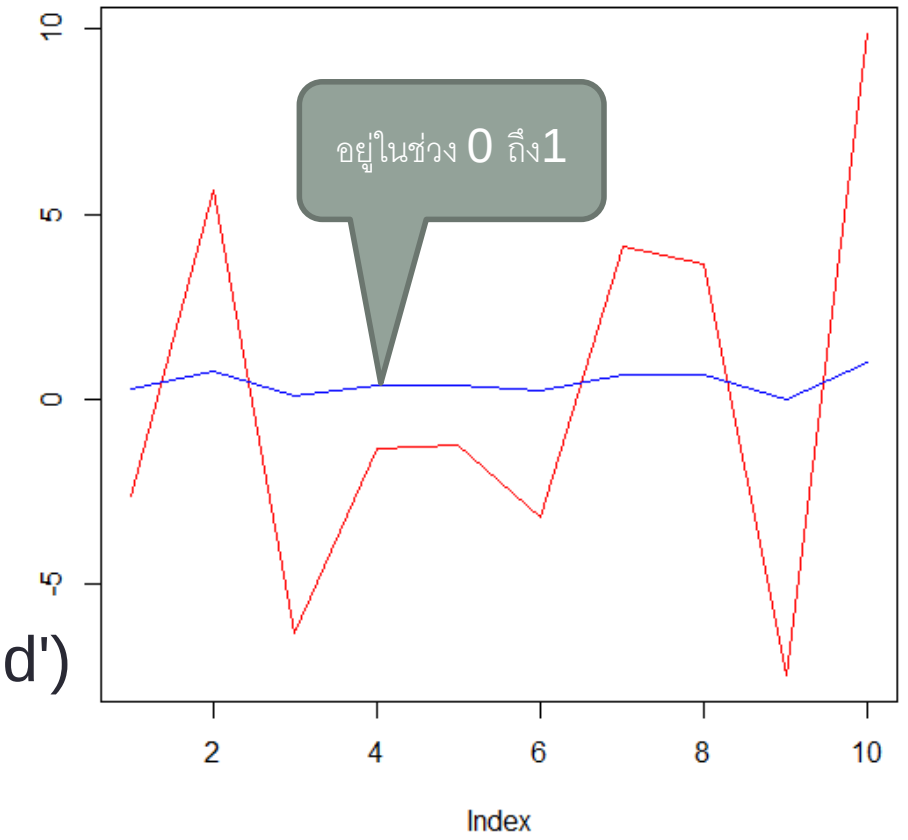
$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

5.3.2 Test Min-Max function

อยู่ ซีก บวก ทั้งหมด

```
funcMaxMin = function(x)
{
        minx = min(x)
        maxx = max(x)
        range = maxx-minx
        z = (x - minx) / range
        return(z)
}
```

```
x1 = runif(10,-10,10)
print(x1)
x2 = funcMaxMin(x1)
#x2 = scale(x1)
print(x2)
plot(x1,type='l', col='red')
lines(x2,col='blue')
```



5.3.3 Shift position

- ใช้การบวกลบเลขเพื่อเลื่อนตำแหน่งข้อมูล
- ความห่างระหว่างสมาชิกเหมือนเดิม ต่างที่สัดส่วน
- ใช้สูตรนี้

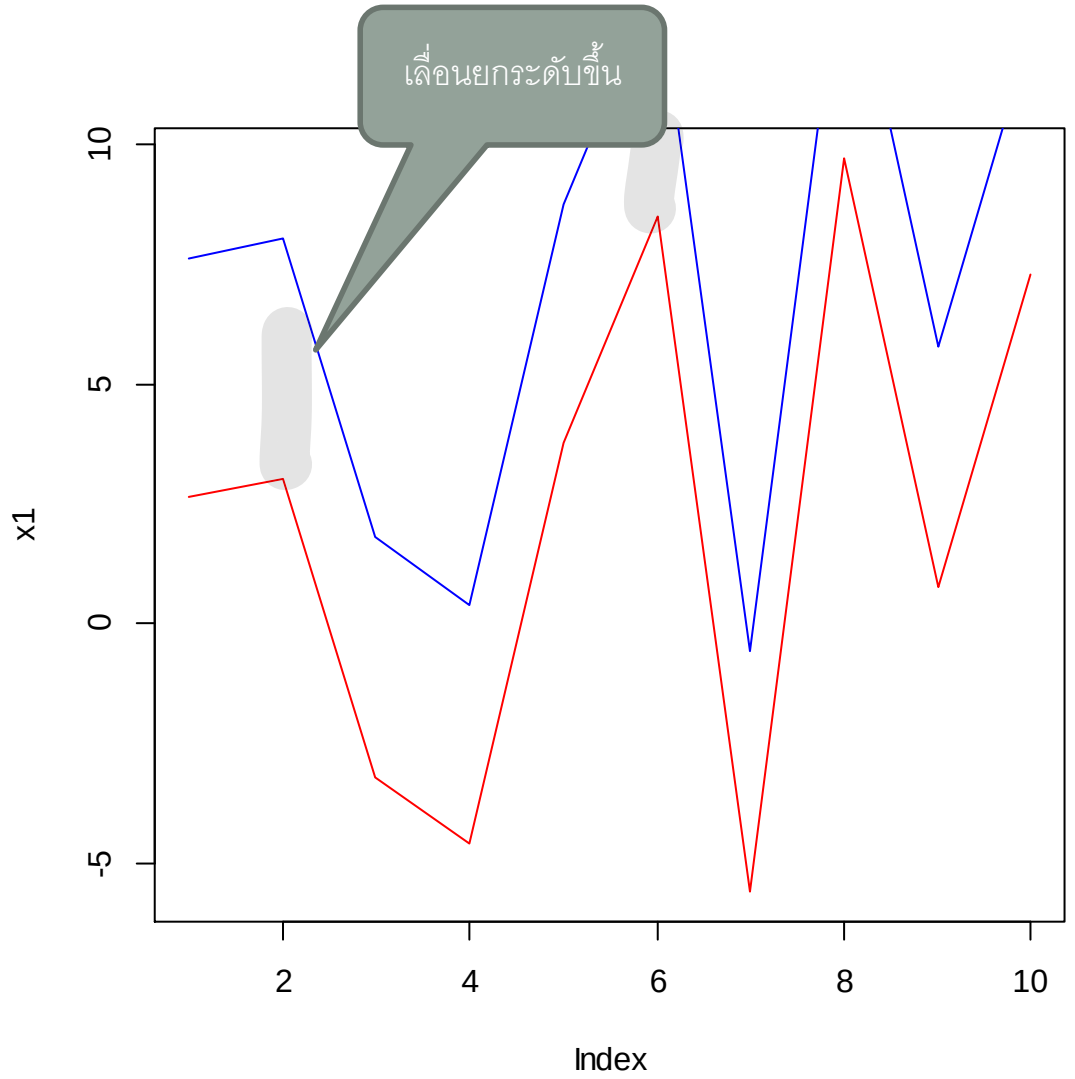
$$z = X + c$$

constant value
ค่าคงที่

ค่าข้อมูล

5.3.3 Shift position

```
x1 = runif(10,-10,10)

print(x1)

x2 = x1+5

print(x2)

plot(x1,type='l', col='red')

lines(x2,col='blue')
```



5.3.4 Ratio adjustment

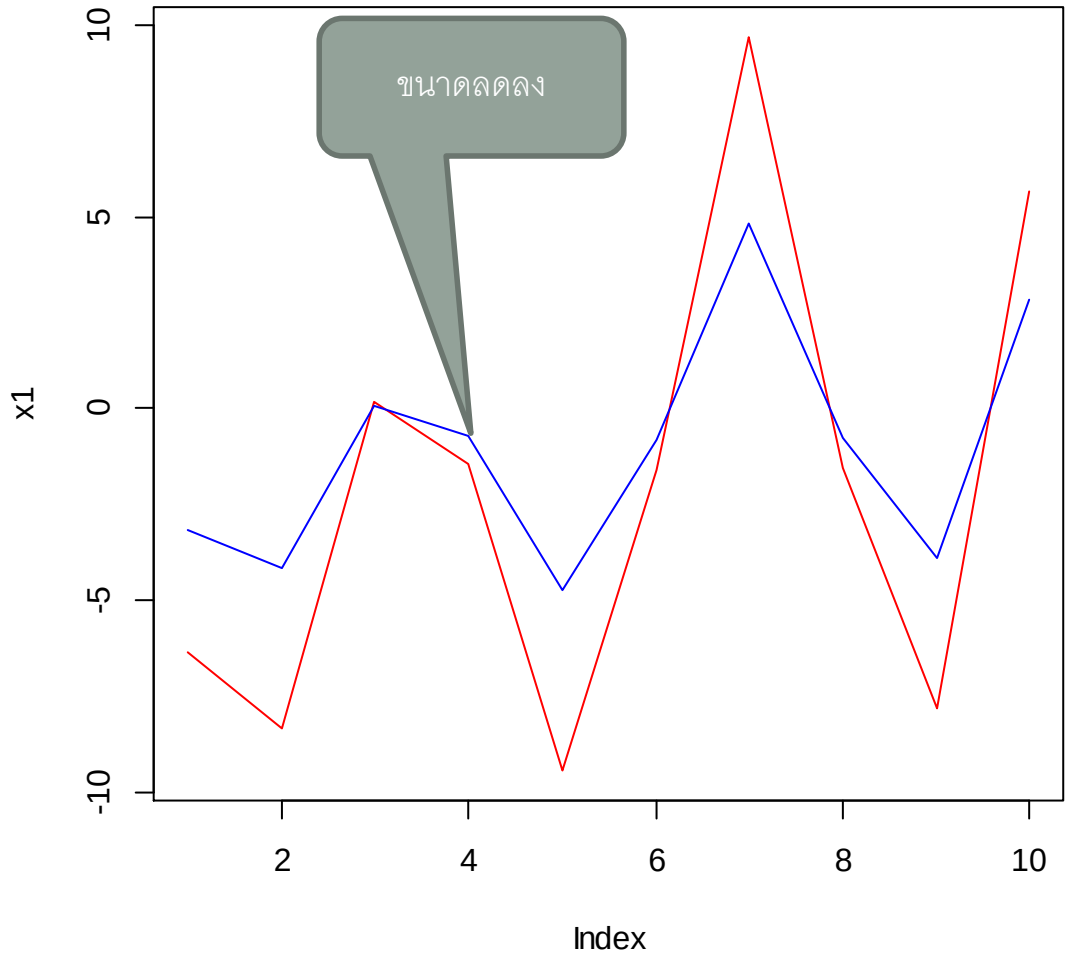
- ใช้การคูณหรือหารเพื่อปรับขนาดข้อมูล
- ความห่างระหว่างสมาชิกเหมือนเดิม ต่างที่สัดส่วน
- ใช้สูตรนี้

$$z = X * c$$

constant value
ค่าคงที่

5.3.4 Ratio adjustment

```
x1 = runif(10,-10,10)

print(x1)

x2 = x1*0.5

print(x2)

plot(x1,type='l', col='red')

lines(x2,col='blue')
```



5.3.5 Zero-Sum Normalization

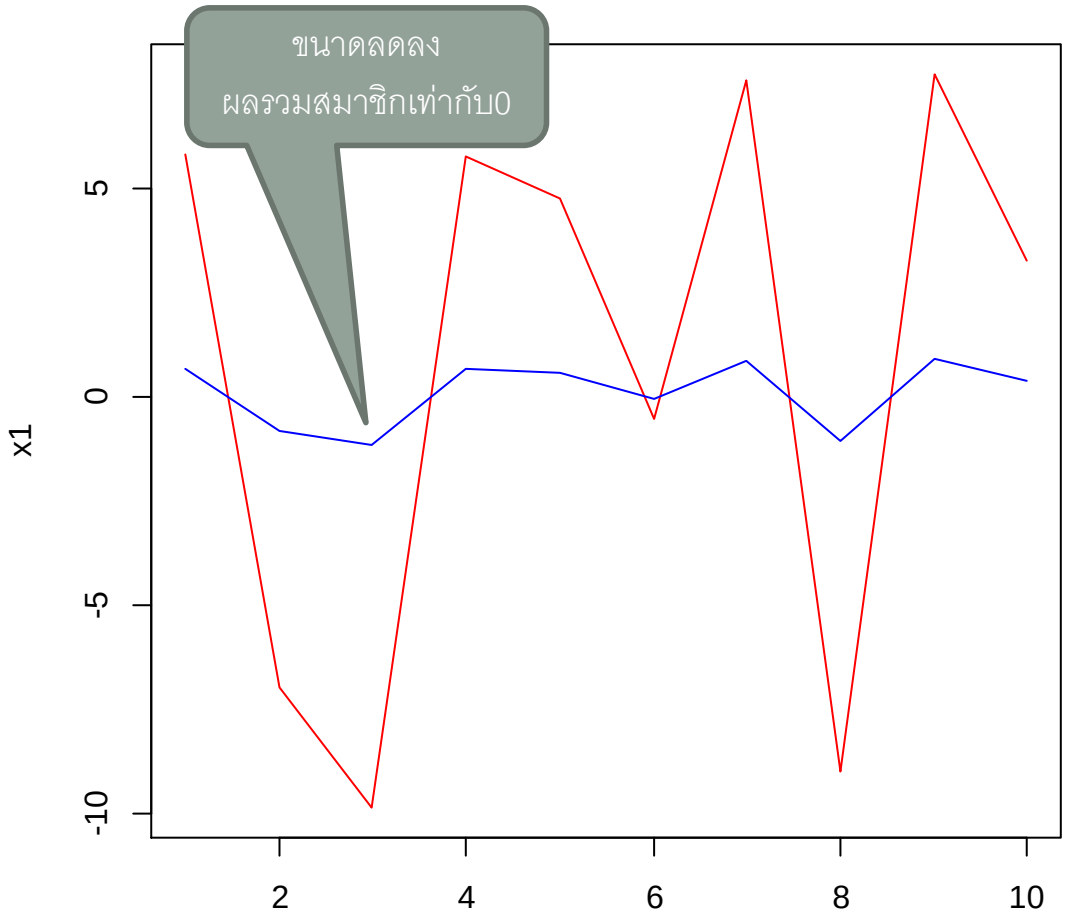
- ผลรวมของสมาชิกมีค่าเป็น 0 (บางที่ใช้ 1)
- ความห่างระหว่างสมาชิกต่างไปไม่เป็นสัดส่วนเดิม
- ใช้สูตรนี้

$$z = \frac{X}{\sum X} + c$$

constant value
ค่าคงที่

5.3.5 Zero-sum normalization

```
x1 = runif(10,-10,10)
print(x1)
funcNormlization = function(x)
{
        sumx = sum(x)
        z = x/sumx
        return(z)
}
x2 = funcNormlization(x1)
print(x2)
plot(x1,type='l', col='red')
lines(x2,col='blue')
```



Excellent data scaling

A = A เหมือนกันไหม?

วิธีการปรับขนาดที่ดีคือ

- ลดหรือเพิ่มแต่ขนาด
- คงความสัมพันธ์ของสมาชิกไว้ ใช้ Pearson's correlation พิสูจน์

Pearson's Correlation

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

Scale of correlation coefficient	Value
$0 < r \leq 0.19$	Very Low Correlation
$0.2 \leq r \leq 0.39$	Low Correlation
$0.4 \leq r \leq 0.59$	Moderate Correlation
$0.6 \leq r \leq 0.79$	High Correlation
$0.8 \leq r \leq 1.0$	Very High Correlation

1 เหมือน
0 ไม่เหมือน

Example code:

```
amount = 1000
lowerbound = -1000
upperbound = 1000
data =runif(amount,lowerbound,upperbound )
plot(data)

mean(data)
sd(data)

#########REDUCE SIZE#####################
data_reducedsize = data/1000
mean(data_reducedsize)
sd(data_reducedsize)

##########Z-SCORE####################
funcZscore = function(x)
{
        avg = mean(x)
        sd = sd(x)
        z = (x - avg) / sd
        return(z)
}
data_applied_zscore = funcZscore(data)
mean(data_applied_zscore)
sd(data_applied_zscore)
{
        minx = min(x)
        maxx = max(x)
        range = maxx-minx
        z = (x - minx) / range
        return(z)
}
data_applied_minmax = funcMaxMin (data)
mean(data_applied_minmax )
sd(data_applied_minmax )

##########NORMALIZATION FUNC####################
funcNormlization = function(x)
{
        sumx = sum(x)
        z = x/sumx
        return(z)
}
data_applied_norm = funcNormlization (data)
mean(data_applied_norm )
sd(data_applied_norm )

##############################
install.packages("ggpubr")
library("ggpubr")
cor(data,data)
cor(data,data_reducedsize)
cor(data,data_applied_zscore)
cor(data,data_applied_minmax)
cor(data,data_applied_norm)

summary(data)
summary(data_reducedsize)
summary(data_applied_zscore)
summary(data_applied_minmax)
summary(data_applied_norm)
```

library สำหรับ Pearson's correlation

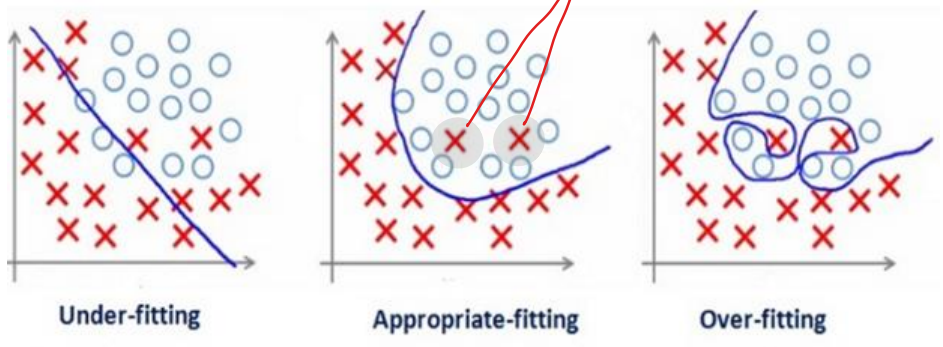
```
> summary(data)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-997.433 -513.745   13.417    8.569  523.796  994.770
> summary(data_reducedsize)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-0.997433 -0.513745  0.013417  0.008569  0.523796  0.994771
> summary(data_applied_zscore)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-1.739737 -0.903267  0.008385  0.000000  0.891012  1.705494
> summary(data_applied_minmax)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.2428  0.5074  0.5050  0.7636  1.0000
> summary(data_applied_norm)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-0.116401 -0.059954  0.001566  0.001000  0.061127  0.116090
```

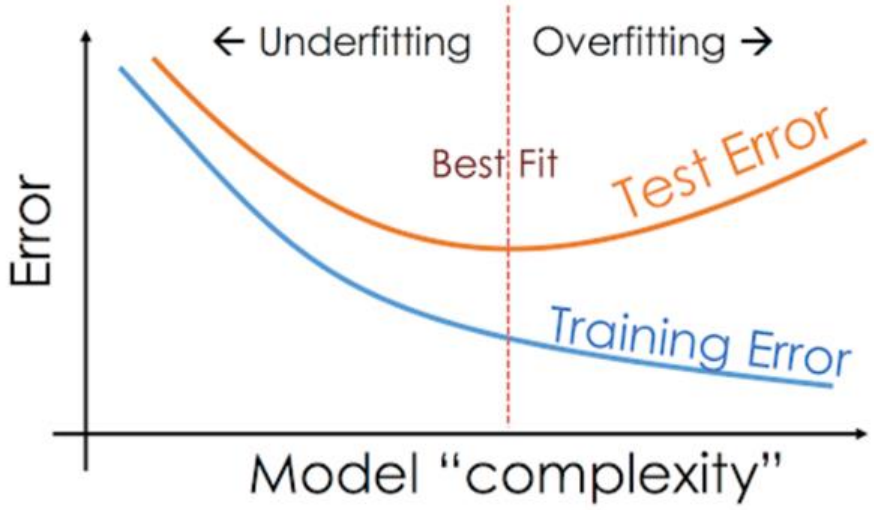
UNDERFITTING – OVERFITTING

เพื่อให้รู้จักปัญหาการสอน ML ดีเกินไปและสอนไม่ดีเกินไป

To recognize a problem of ML training by overfitting and underfitting

5.4.1 Under and Over-fitting





- การสอนที่มุ่งให้ Low error มักจะเกิดปัญหา Overfitting
- กลับกัน สอนที่ไม่ได้ High error ก็เกิดปัญหา Underfitting
- ปัจจัยให้เกิด
  - ลักษณะ model ML
  - ลักษณะข้อมูล กระจายไป
  - ลักษณะข้อมูล มีส่วนผิดที่ครุมเครือมากผสม
  - โปรแกรมเมอร์/นักวิจัย ทำผิดเอง

5.4.2 The method reduce overfitting

- NN เกิด overfitting ได้
- แก้โดยหยุดสอนก่อนได้ค่า error ต่ำมากๆ

- นำ data ที่แตกต่างมาใส่ให้รู้จัก

- ไม่ใช้ target เดียว ต้องสุ่มเลือก target เพื่อสร้างความหลากหลาย

- ใช้หลาย NN ช่วยตัดสินใจโดยการเฉลี่ย ไม่ใช้ตัวเดียว
  - Example AI in MAR spaceship project

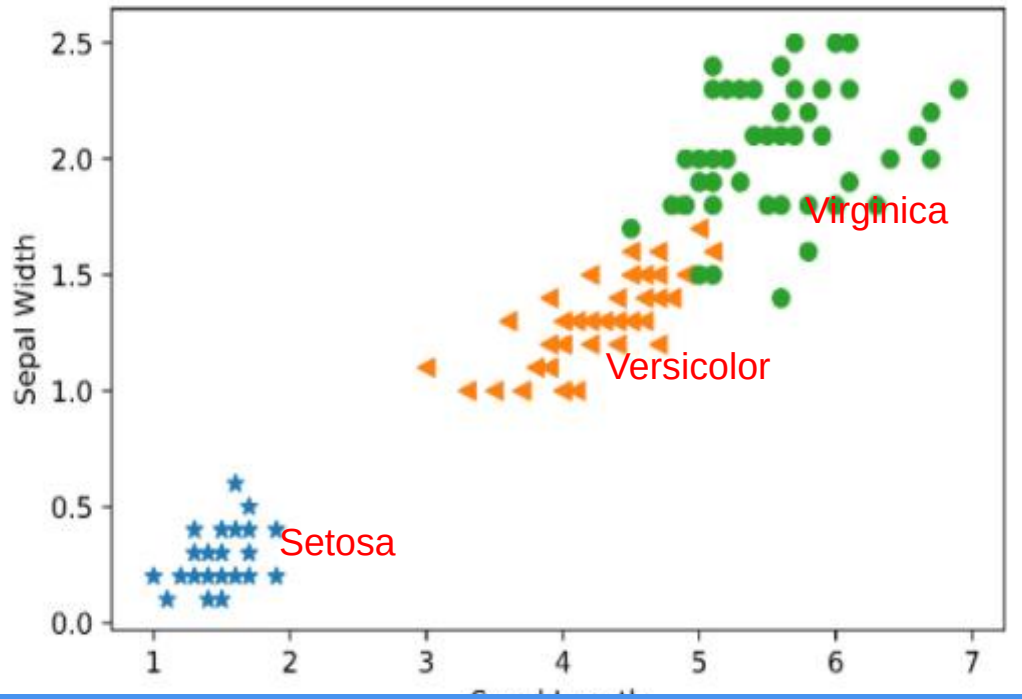
IRIS DATASET

เพื่อมีประสบการใช้ NN แยกชนิดดอกไม้

To experience the usage NN recognize the flower

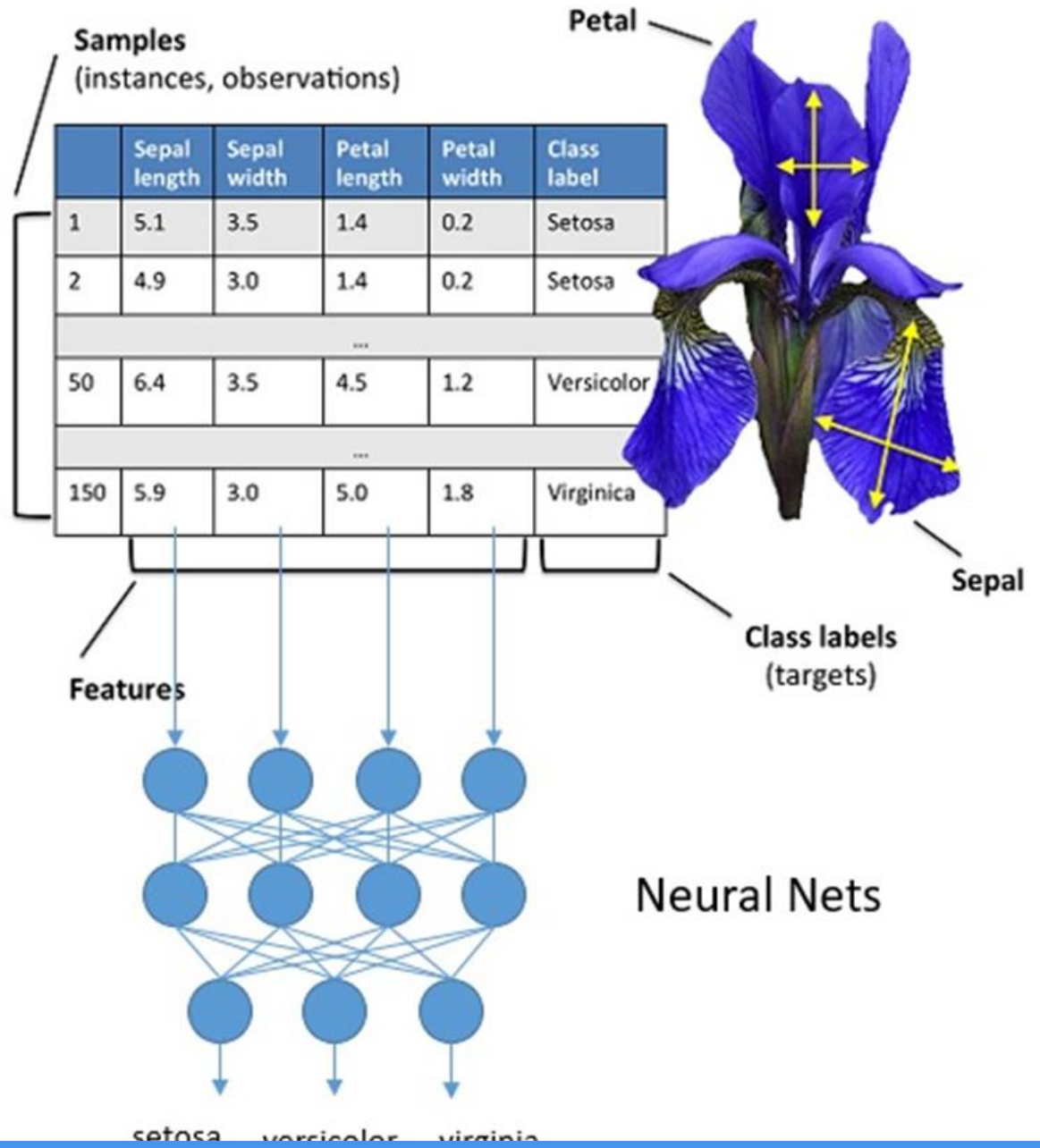
5.5 Iris dataset

- IRIS dataset ไอรีสด้าต้า นิยมใช้กันมาก
- ข้อดีทำให้มี comparator วัดงานวิจัย
- หา download ได้ฟรี
- คนสร้างเขาวัดขนาดของใบ Width, Length และระบุชนิดไว้เป็น target
- ข้างในประกอบด้วย
  - Setosa ซีโทซา
  - Virginica วีจินิคา
  - Versicolor เวอสิคัลเลอร์



5.5.1 Iris dataset

- 5-column in data set
  - Sepal length
  - Sepal width
  - Petal length
  - Petal width
  - Class label

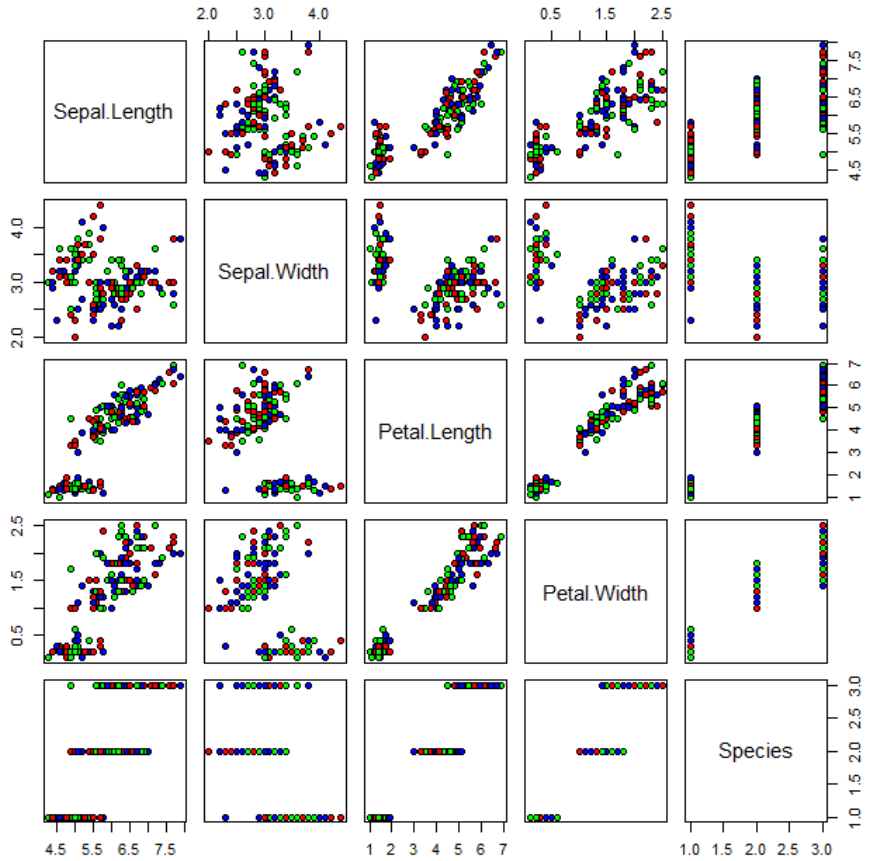


	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Activity 5.5 Load Iris data set

```
#LOAD DATA
data(iris)

#PRINT SUMMARY
summary(iris)

#PLOT
plot(iris,
   bg = c("red","green","blue"),pch=21)
```

```
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
Median :5.800   Median :3.000   Median :4.350   Median :1.300
Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
       Species
setosa     :50
versicolor:50
virginica :50
```



Data specification.

Width:	0.1 to 12.5
Length:	1.0 to 7.9
Unit:	Centimeter
setosa:	50 items
versicolor:	50 items
virginica	50 items

Activity 5.5 Load Iris data set

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
23	4.6	3.6	1.0	0.2	setosa
132	7.9	3.8	6.4	2.0	virginica
38	4.9	3.6	1.4	0.1	setosa
44	5.0	3.5	1.6	0.6	setosa
94	5.0	2.3	3.3	1.0	versicolor
141	6.7	3.1	5.6	2.4	virginica
129	6.4	2.8	5.6	2.1	virginica
65	5.6	2.9	3.6	1.3	versicolor
138	6.4	3.1	5.5	1.8	virginica
53	6.9	3.1	4.9	1.5	versicolor
89	5.6	3.0	4.1	1.3	versicolor
71	5.9	3.2	4.8	1.8	versicolor
18	5.1	3.5	1.4	0.3	setosa
35	4.9	3.1	1.5	0.2	setosa
30	4.7	3.2	1.6	0.2	setosa
111	6.5	3.2	5.1	2.0	virginica
134	6.3	2.8	5.1	1.5	virginica
125	6.7	3.3	5.7	2.1	virginica
16	5.7	4.4	1.5	0.4	setosa
10	4.9	3.1	1.5	0.1	setosa

```
s = sample(1:150,20)
siris = iris[s,]
siris
```

- 20 sample data from 150 data to check data details.

5.6 Iris dataset on neuralnet()

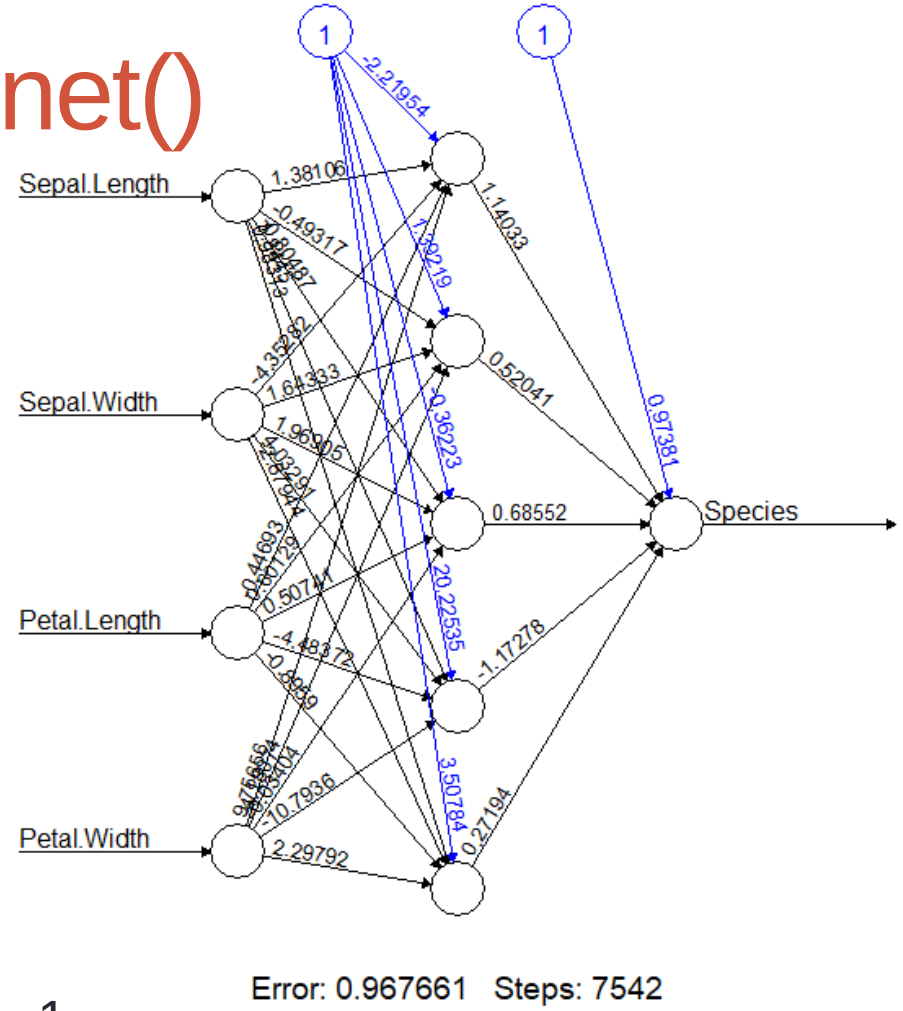
```
##########################################
#ACTIVITY 5-6 IRIS DATA SET ON NEURALNET
##########################################

#PREPARING DATA
data(iris)
head(iris)        #CHECK COLUMN NAME
#Sepal.Length Sepal.Width Petal.Length
#Petal.Width Species

#CHEANGING setosa=1, versicolor=2, virginica=3
datatrain = data.frame(iris)

datatrain$Species =
       c(rep(1,50),rep(2,50),rep(3,50))
datatrain
```

```
#TRAINING
library("neuralnet")
model <- neuralnet(
          Species~Sepal.Length+
          Sepal.Width+Petal.Length+Petal.Width,
          datatrain,
          hidden=5, ##<--Change here
          rep = 1,
          linear.output = TRUE)
print(model)
plot(model)
print(model$net.result)
```

- Before training, Iris at Specie column need to change from text to integer value.

Activity 5.6 Iris dataset on neuralnet()

```
 [1,] 0.9969690    [50,] 0.9980830    [100,] 2.0139561
 [2,] 1.0029164    [51,] 2.0070355    [101,] 2.9829099
 [3,] 1.0000290    [52,] 1.9818641    [102,] 3.0250643
 [4,] 0.9999288    [53,] 1.9961415    [103,] 3.0182555
 [5,] 0.9974936    [54,] 1.9471998    [104,] 3.0470866
 [6,] 0.9993556    [55,] 1.9719388    [105,] 3.0133043
 [7,] 1.0007880    [56,] 2.0322993    [106,] 3.0183982
 [8,] 0.9971131    [57,] 1.9834482    [107,] 2.8782836
 [9,] 1.0044037    [58,] 1.9884696    [108,] 3.0609974
[10,] 0.9938142    [59,] 2.0204959    [109,] 3.0246080
[11,] 0.9962128    [60,] 1.9840318    [110,] 2.9798797
[12,] 0.9974919    [61,] 1.9744310    [111,] 2.9442357
[13,] 0.9954311    [62,] 1.9694776    [112,] 3.0172917
[14,] 0.9988545    [63,] 1.9758656    [113,] 3.0164156
[15,] 0.9962150    [64,] 2.0254515    [114,] 3.0161623
[16,] 1.0008009    [65,] 2.0003704    [115,] 3.0198175
[17,] 0.9997327    [66,] 1.9942003    [116,] 3.0076843
[18,] 0.9999851    [67,] 2.0108983    [117,] 3.0110507
[19,] 0.9971749    [68,] 1.9706303    [118,] 2.9820601
[20,] 0.9991712    [69,] 2.1106838    [119,] 2.9852588
[21,] 0.9951246    [70,] 2.0278231    [120,] 2.8225728
[22,] 1.0002536    [71,] 2.4424848    [121,] 3.0101465
[23,] 0.9991591    [72,] 1.9969952    [122,] 3.0158055
[24,] 1.0114278    [73,] 2.3219683    [123,] 3.0178016
[25,] 0.9972057    [74,] 2.0468231    [124,] 2.8725434
[26,] 1.0008031    [75,] 2.0120565    [125,] 3.0049854
[27,] 1.0033269    [76,] 1.9879417    [126,] 3.0413179
[28,] 0.9963930    [77,] 1.9879632    [127,] 2.7529642
[29,] 0.9969794    [78,] 2.3297510    [128,] 2.7141460
                   [79,] 1.9941549    [129,] 3.0196588
```



- Target
  - setosa=1
  - versicolor=2,
  - virginica=3

Summary

ทำ Quiz วันพุธสอน 20.00น.

หาข้อมูลง่ายๆ 3-4 Variable มา Train เป็น NN และลอง Recognize

→ สรุปใส่ PDF A4 1 หน้า Upload ส่งก่อนครึ่งหน้า