MULTI-LAYER PERCEPTRON

Asst.Prof.Dr.Supakit Nootyaskool
IT-KMITL

Study map

- 1.Basic programming
  - R-programming
- 2. Perceptron
  - Activity function
- 3. Feed Forward NN
  - Logistic function
- 4. Feed Forward NN
  - XOR gate
  - Multi-layer perceptron
- 5. Example & Library Feed Forward NN
  - N:N, 1:N model
  - iris dataset

- 6. Writing NN Code
  - Data scaling, Confusion matrix
  - Writing NN code
- 7. Recurrent Neural Network
- 8. Apply RNN & Library
- 9. GRU LSTM
- 10. CNN
- 11 Apply GA to NN

Learning Outcome

- Understand number of hidden relate to data complexity.
- Have an experience using neuralnet() to train and predict small data.

Overview NN Pros and Cons

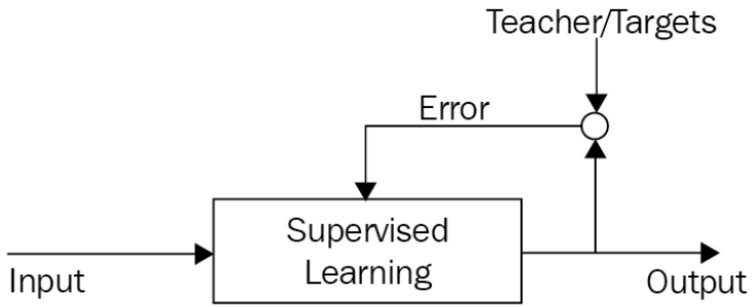
Pros

- NN are goods for nonlinear data with large number of input such as images.
- After training, the predictions are fast
- NN can be trained with any number of inputs and layers
- NN works best with more data points

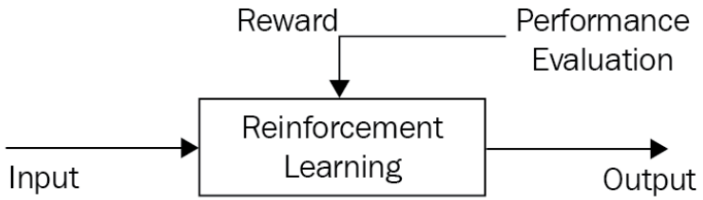
Cons

- NN are black boxes
- NN uses time consuming to training on the CPUs-> solved by using GPUs
- NN may leads to over-fitting.

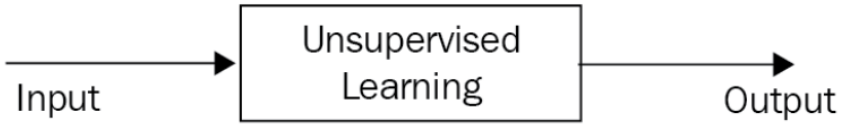
Overview ML



- Supervised learning is the training data as a teacher to the model
- Machine learn from the target data



- Reinforcement learning is ML where is constant feedback given model to adapt to



- Unsupervised is self organization
- The output is trained without a target variable
- Techniques related to unsupervised
  - K-means, hierarchical
  - Dimension reduction
  - Self organization map (SOM)

Analysis train and predict result

Train
NN cat

TP

20

TN

50

FP

12

FN

18

Actual value	Predicted value TRUE	Predicted Value FALSE
TRUE	True Positive (TP)	False Negative (FN) err2
FALSE	False Positive (FP) err1	True Negative (TN)

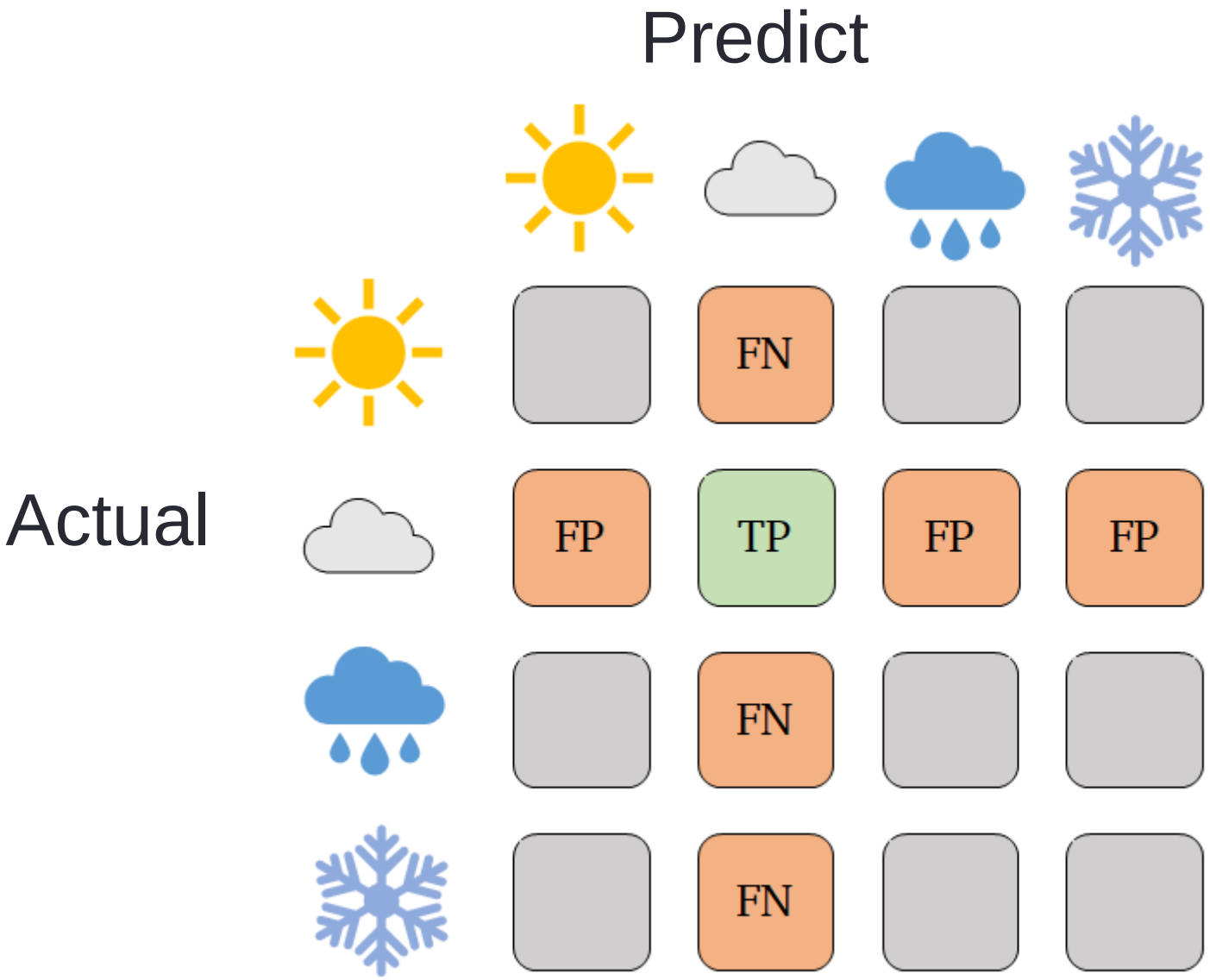
Example: ATK results

ผลตรวจ covid

Actual value	Predicted value TRUE ผลตรวจว่าติด covid	Predicted Value FALSE ผลตรวจว่าไม่ติด
TRUE ติด covid	True Positive (TP)	False Negative (FN) err2
FALSE ไม่ติด covid	False Positive (FP) err1	True Negative (TN)

ความจริง covid

Example: Weather forecasting



Overview Error Matrix

Actual value	Predicted value TRUE	Predicted Value FALSE
TRUE	True Positive (TP)	False Negative (FN) err2
FALSE	False Positive (FP) err1	True Negative (TN)

ความแม่นยำ

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

ความเที่ยงตรง

$$Precision = TP / (TP + FN)$$

Calculate accuracy (แม่นยำ) and precision (เที่ยงตรง)

Train
NN cat

TP  20

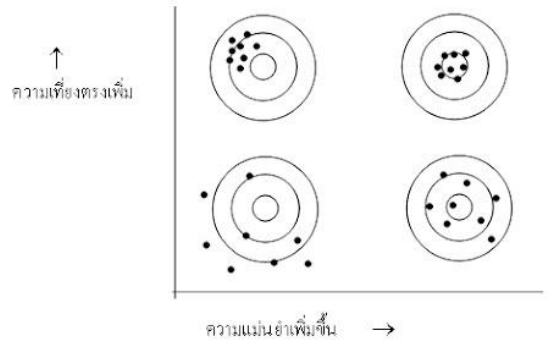
TN  50

FP  12

FN  18

Accuracy = (TP + TN) / (TP + TN + FP + FN)
= (20 + 50) / (20+50+12+18)
= 70 / 100
= 0.7

Precision = TP / (TP + FN)
= 20 / (20 + 50)
0.28



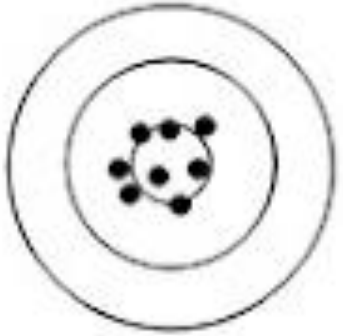
Calculate accuracy (แม่นยำ) and precision (เที่ยงตรง)

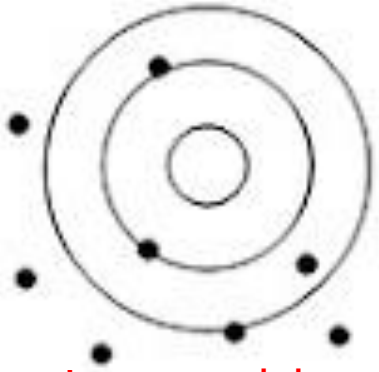
Accuracy = (TP + TN) / (TP + TN + FP + FN)
= (20 + 50) / (20+50+12+18)
= 70 / 100
= 0.7

Precision = TP / (TP + FN)
= 20 / (20 + 50)
= 0.28

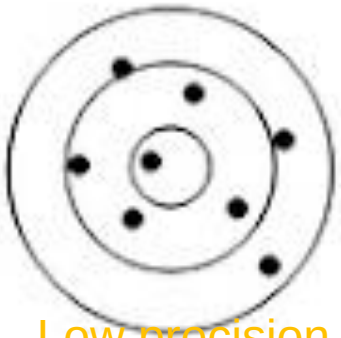
High precision
Low accuracy
เที่ยงตรงแต่ไม่แม่นยำ

High precision
High accuracy
เที่ยงตรง และแม่นยำ



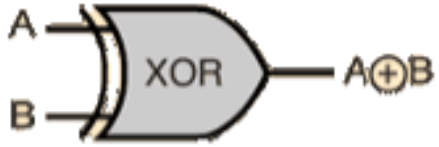


Low precision
High accuracy
ไม่เที่ยงตรง ไม่แม่นยำ

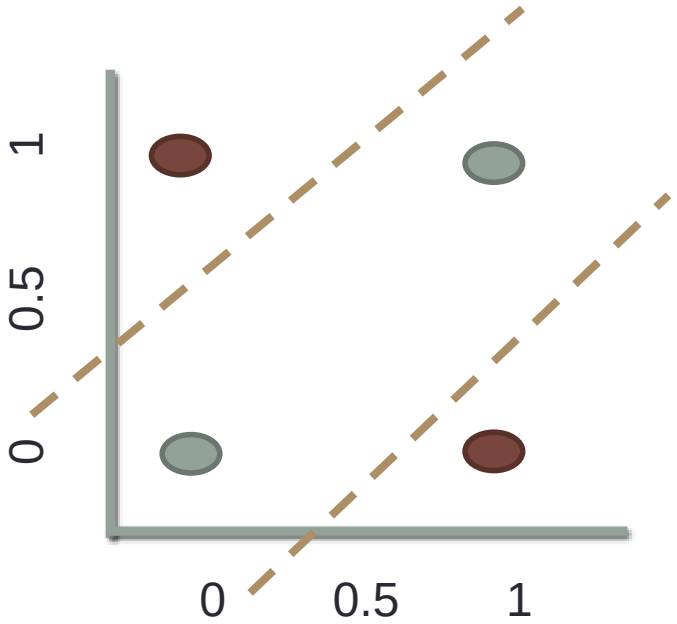


Low precision
High accuracy
ไม่เที่ยงตรงแต่แม่นแม่นยำ

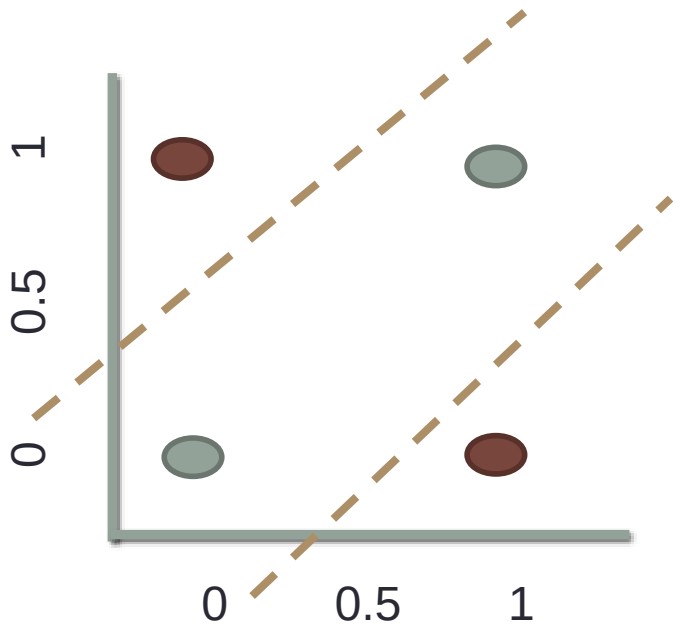
Activity 4.1 Create XOR on Perceptron

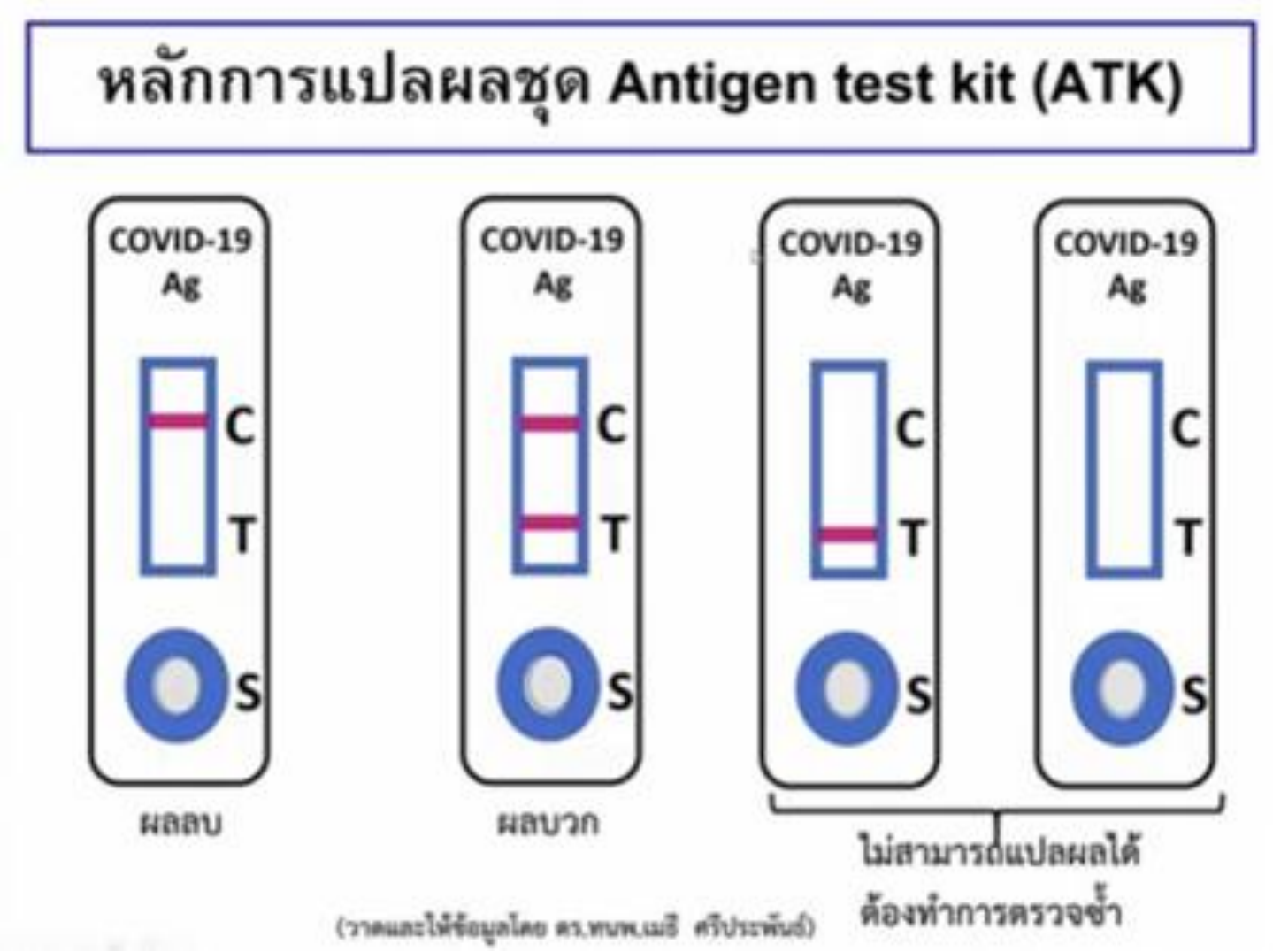


A	B	Out
0	0	0
0	1	1
1	0	1
1	1	0



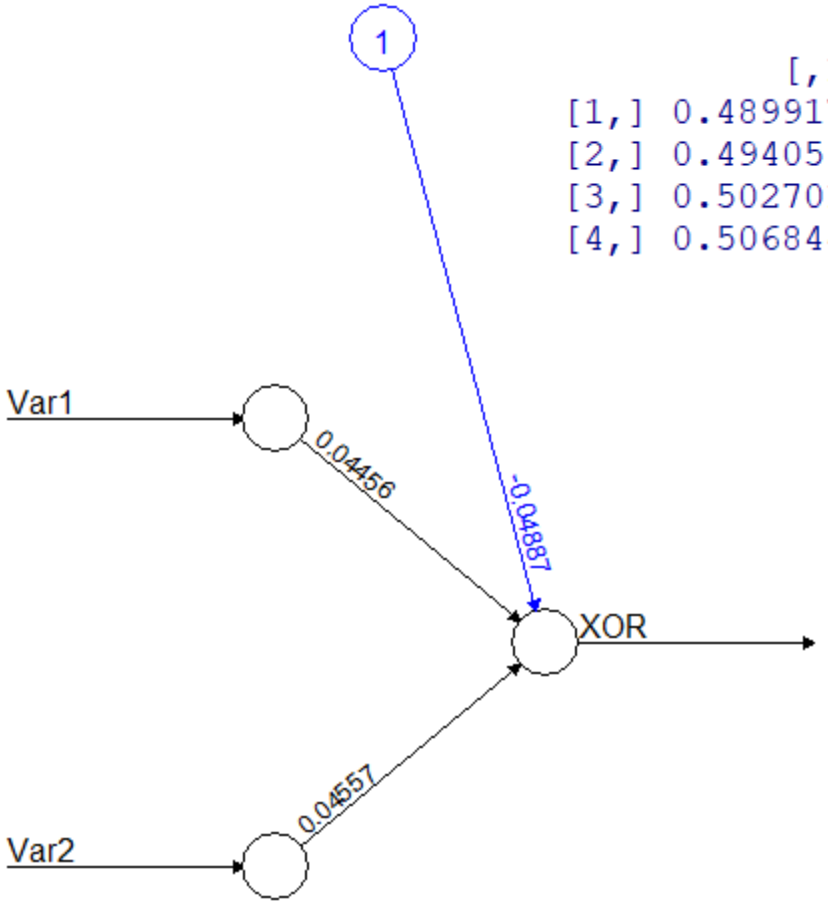

```
library("neuralnet")

XOR = c(0,1,1,0)
truthtable = expand.grid(c(0,1), c(0,1))
XOR.data <- data.frame(truthtable, XOR)
print(XOR.data)

model <- neuralnet( XOR~Var1+Var2,
            XOR.data,
            hidden=0, ##<--Change here
            rep = 5,
            linear.output = FALSE,
            err.fct = "ce")

print(model)
plot(model)
print(model$net.result)
```


1
0.5
0
0
0.5
1


หลักการแปลผลชุด Antigen test kit (ATK)
COVID-19 Ag
C
T
S
ผลลบ
COVID-19 Ag
C
T
S
ผลบวก
COVID-19 Ag
C
T
S
COVID-19 Ag
C
T
S
ไม่สามารถแปลผลได้
ต้องทำการตรวจซ้ำ
(วาดและให้ข้อมูลโดย ดร.ทนพ.เมธี ศรีประพันธ์)

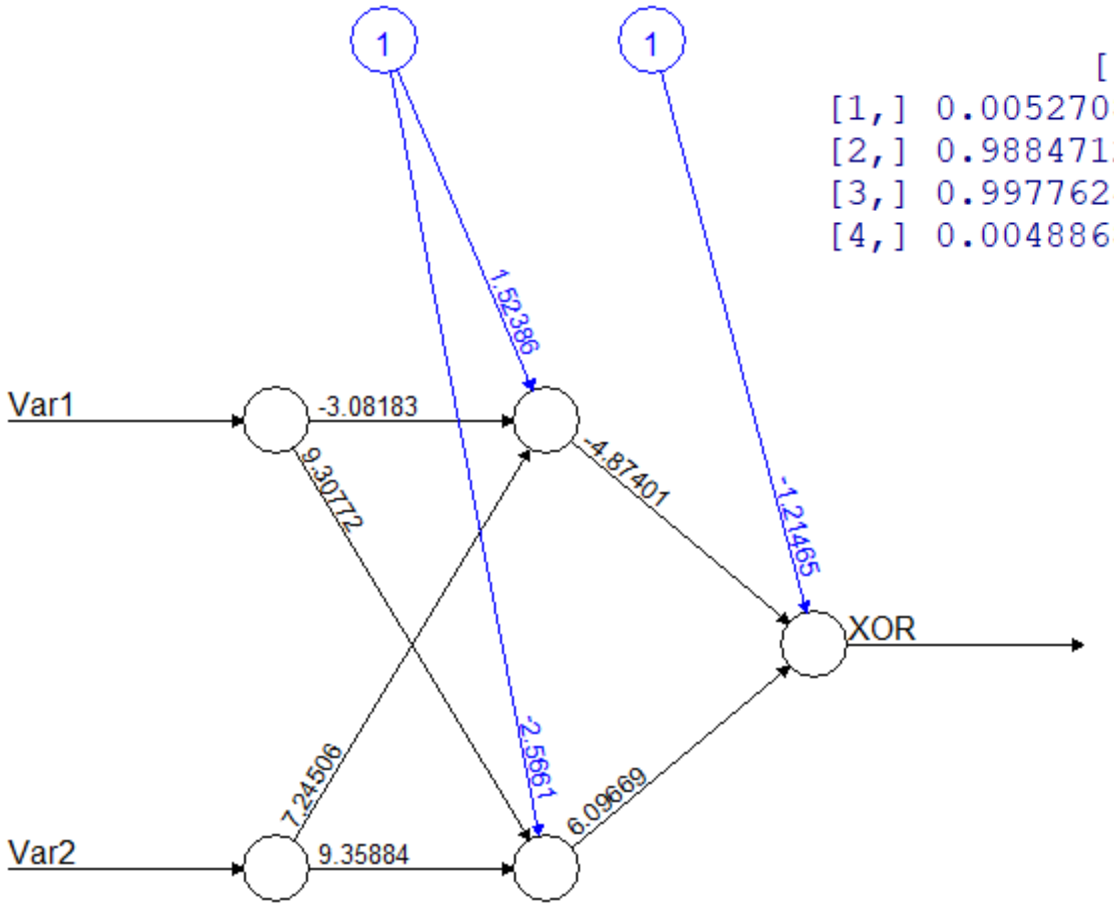
Activity 4.1 Create XOR on Perceptron

Hidden 0




```
          [,1]
[1,] 0.4899179
[2,] 0.4940599
[3,] 0.5027019
[4,] 0.5068445
```

Hidden 2



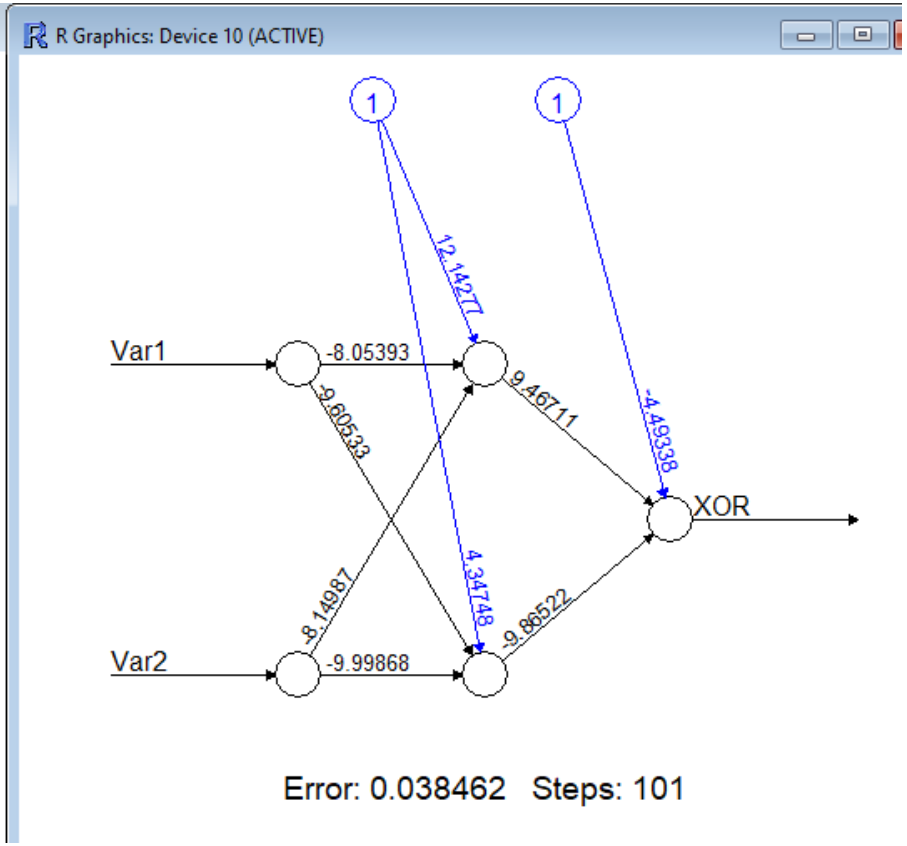

```
          [,1]
[1,] 0.005270833
[2,] 0.988471291
[3,] 0.997762402
[4,] 0.004886802
```

Activity 4.2 Predict the model XOR gate

```
var1 = c(0,1,0,1)
var2 = c(0,0,1,1)
datatest = data.frame(var1,var2)
pred <- predict(model, datatest)
pred
```

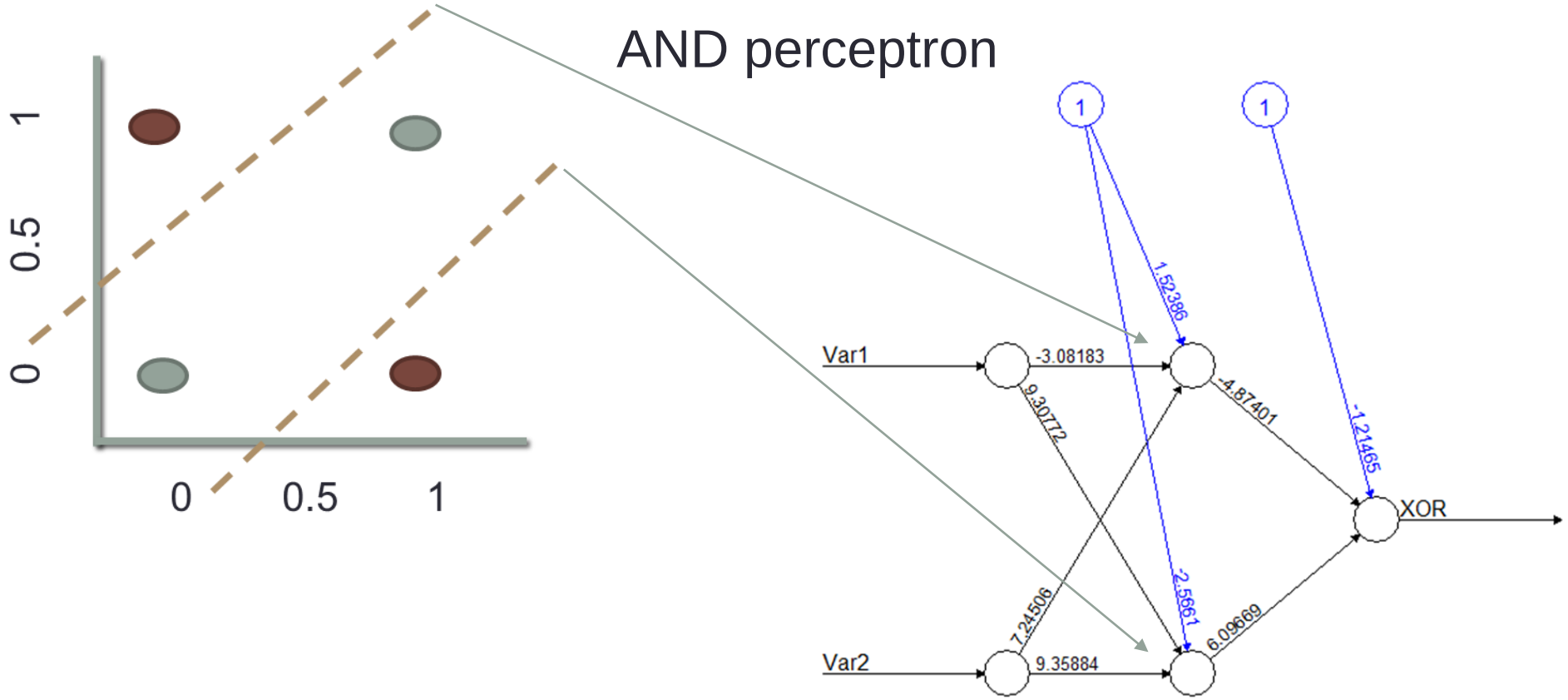
R Console

```
> var1 = c(0,1,0,1)
> var2 = c(0,0,1,1)
> datatest = data.frame(var1,var2)
> pred <- predict(model, datatest)
> pred
             [,1]
[1,] 0.008446495
[2,] 0.991562451
[3,] 0.991571910
[4,] 0.012957772
>
```



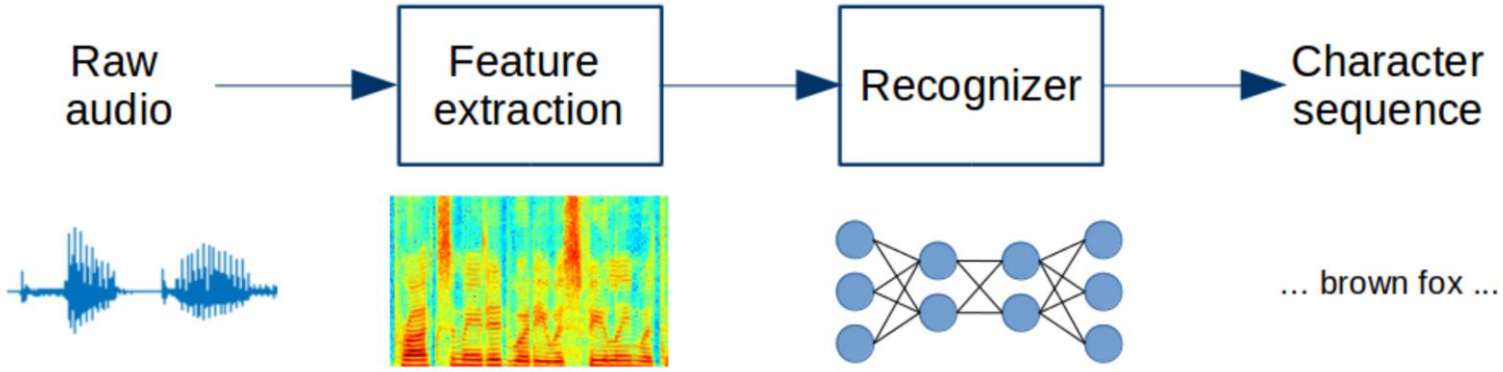
Why does the XOR gate need two hidden nodes?

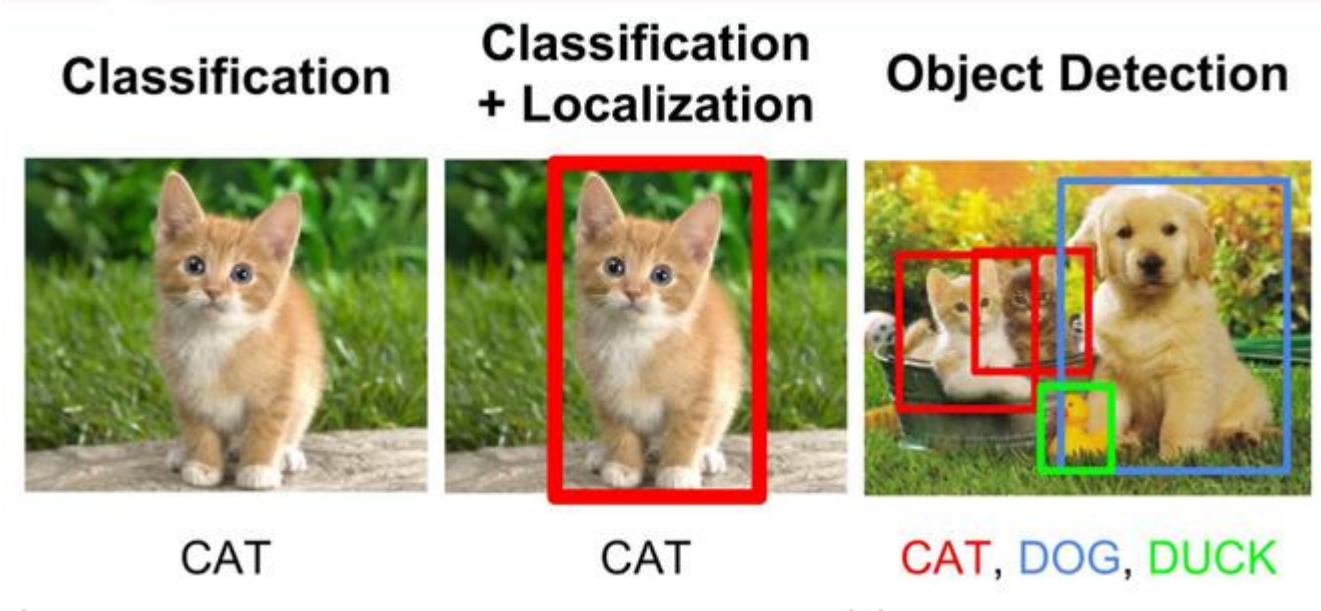
- Uses two classifier
- XOR is a three-layer network combining OR and AND perceptron

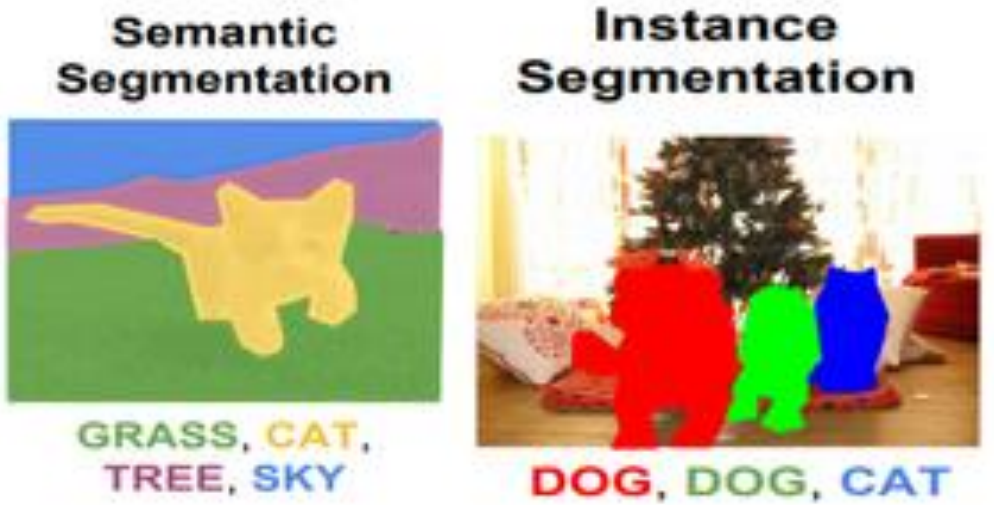


Multi-Layer Perceptron (MLP)

- MLPs are extremely useful for complex problems in the research.
- MLP are used in diverse fields, such as speech recognition,, object recognition, image classification, object localization, object detection, image segmentation, and language translation.







https://www.geeksforgeeks.org/object-detection-vs-object-recognition-vs-image-segmentation/

Activity 4.3 Calculate volume of cylinder with NN

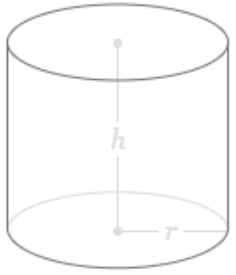
Right cylinder

Solve for volume

$$V = \pi r^2 h$$

r Radius Enter value

h Height Enter value




```
> datatrain
   radius height      volume
1       1      1    3.141593
2       2      2   25.132741
3       3      3   84.823002
4       4      4  201.061930
5       5      5  392.699082
6       6      6  678.584013
7       7      7 1077.566280
8       8      8 1608.495439
9       9      9 2290.221044
10     10     10 3141.592654
11     11     11 4181.459822
12     12     12 5428.672105
```

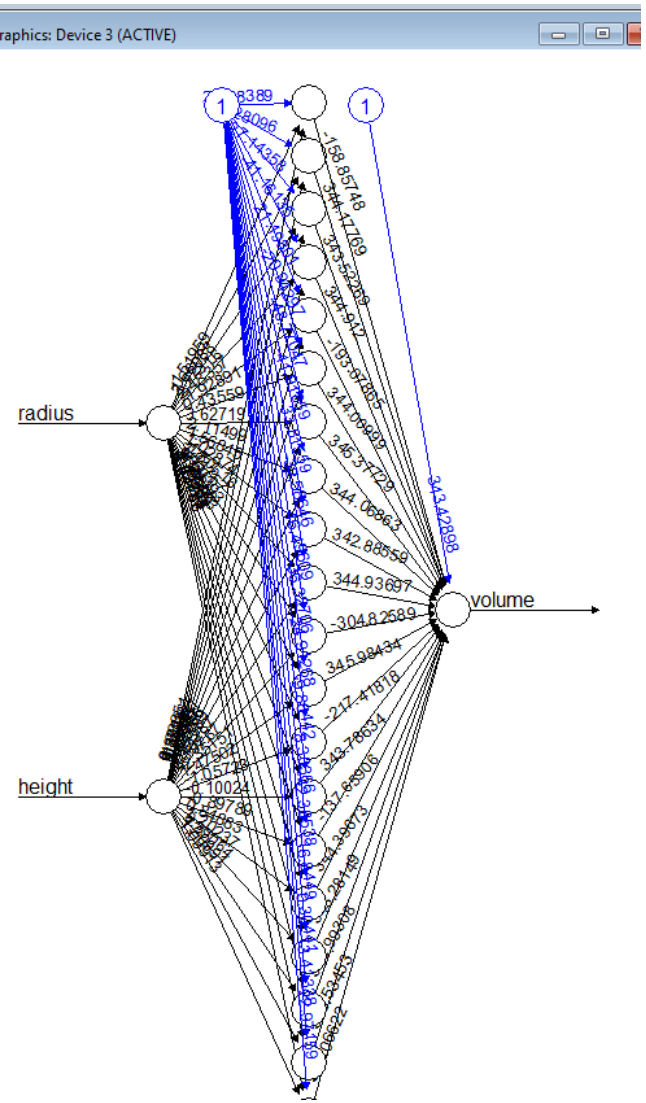
#CREATE DATA TRAIN

```
library("neuralnet")

radius = 1:12

height = 1:12

volume = pi*radius*radius*height

datatrain = data.frame(radius,height,volume)

datatrain
```

Activity 4.3 Calculate volume of cylinder with NN

```
# TRAIN THE DATA

model <- neuralnet( volume~radius+height,
            datatrain,
            hidden=20, ##<--Change here
            rep = 1,
            linear.output = TRUE)

print(model)

plot(model)

print(model$net.result)
```

```
1layhid3.to.volume       3.4
1layhid4.to.volume       3.4
1layhid5.to.volume      -1.9
1layhid6.to.volume       3.4
1layhid7.to.volume       3.4
1layhid8.to.volume       3.4
1layhid9.to.volume       3.4
1layhid10.to.volume      3.4
1layhid11.to.volume     -3.0
1layhid12.to.volume      3.4
1layhid13.to.volume     -2.1
1layhid14.to.volume      3.4
1layhid15.to.volume     -1.3
1layhid16.to.volume      3.4
1layhid17.to.volume      3.4
1layhid18.to.volume      3.4
1layhid19.to.volume      3.4
1layhid20.to.volume      3.4

attr(,"class")
[1] "nn"
> plot(model)
> print(model$net.result)
[[1]]
             [,1]
 [1,]    3.141452
 [2,]   25.132814
 [3,]   84.822522
 [4,]  201.063058
 [5,]  392.698544
 [6,]  678.584177
 [7,] 1077.566230
 [8,] 1608.495463
 [9,] 2290.221032
[10,] 3141.592661
[11,] 4181.459823
[12,] 5428.672105
```



Activity 4.3 Calculate volume of cylinder with NN

```
# SAVE THE MODEL TO A BINARY FILE
save(model,file = "nnmodel.dat")

# PREPARING TEST DATA
radius = runif(12,1,12)
height = radius
datatest = data.frame(radius,height)
# LOAD THE MODEL
model_test = load("nnmodel.dat")
pred <- predict(model, datatest)
pred
datatest[,3] = pred
datatest
```

```
$data
   radius height      volume
1       1      1    3.141593
2       2      2   25.132741
3       3      3   84.823002
4       4      4  201.061930
5       5      5  392.699082
6       6      6  678.584013
7       7      7 1077.566280
8       8      8 1608.495439
9       9      9 2290.221044
10     10     10 3141.592654
11     11     11 4181.459822
12     12     12 5428.672105
```

```
> datatest[,3] = pred
> datatest
      radius   height          V3
1  2.026547 2.026547   25.73332
2  2.390563 2.390563   37.84731
3  7.679099 7.679099 1446.11547
4  1.172730 1.172730    9.28779
5  3.102445 3.102445   96.04251
6  9.432652 9.432652 2675.39493
7  2.831166 2.831166   68.02305
8  9.264796 9.264796 2524.03196
9  1.724214 1.724214   20.05491
10 1.990937 1.990937   24.93392
11 1.752011 1.752011   20.50487
12 4.847668 4.847668  355.61002
```

compare

Activity 4.4

Summary