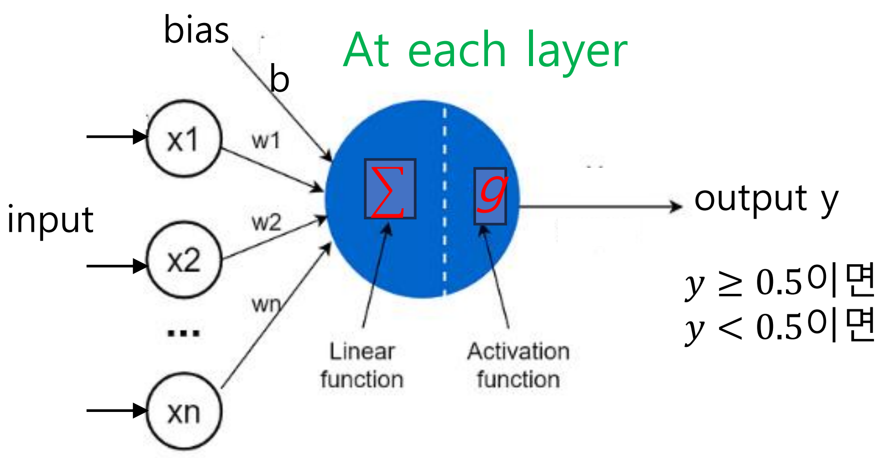
컴퓨터 응용통계

Neural Network

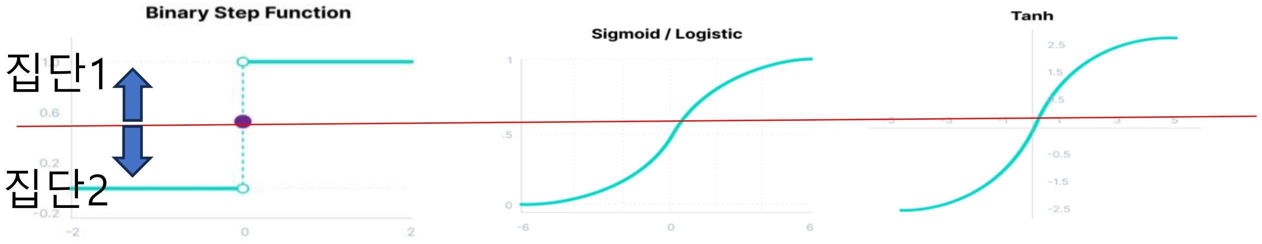
최경미

ANN

$$f(x_i) = g\left(\beta_0 + \sum_{k=1}^{K} \beta_k\, g\left(w_{k0} + \sum_{j=0}^{p} w_{kj}\ x_{ij}\right)\right) \in (-\infty, \infty)$$



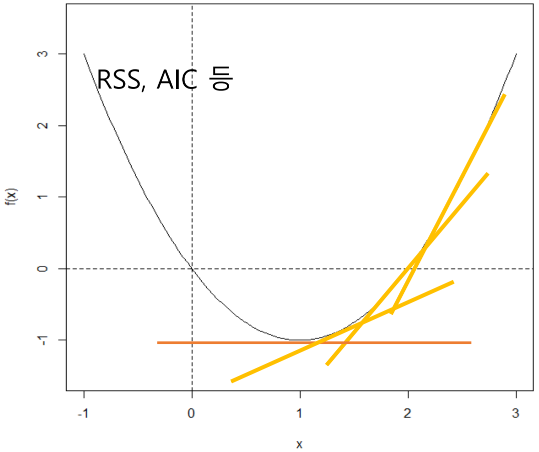
$y \geq 0.5$이면, 그룹 1
$y < 0.5$이면, 그룹 2



$$g(u) = \begin{cases} 0, u \leq 0 \\ 1, u > 0 \end{cases} \qquad g(u) = \frac{1}{1+e^{-u}} \qquad g(u) = \tanh u \in (0,1)$$

STEPS

1. Data = Training set + Test set
2. Fit the model with the training set
3. Predict with the test set
4. Evaluation
5. Fix the model
6. Repeat 2-5



혼동행렬	A로 분류	B로 분류
원래 A		오류
원래 B	오류	

```
##
# 회귀분석 복습
##
# (1) state.x77
mydata <- data.frame(state.x77)
mydata$Density <- mydata$Population/mydata$Area
mydata$Population <- NULL
mydata$Area <- NULL
fit <- lm(Life.Exp~., data=mydata)

#install.packages("car")
library(car)
del <- c("Alaska","Hawaii","Nevada", "North Dakota")
mydata <- mydata[ !rownames(mydata) %in% del, ]
fit <- lm(Life.Exp~., data=mydata)
influencePlot(fit,  id.method="identity")

library(MASS)
fit<-stepAIC(fit)
summary(fit)  #MSE = 0.6857^2 = 0.4701845
```

```
##
# ANN
##
install.packages("neuralnet")
library(neuralnet)

mydata <- data.frame(state.x77)

# Normalize the data
maxs <- apply(mydata, 2, max)
mins <- apply(mydata, 2, min)
scaled <- as.data.frame(scale(mydata, center = mins,  scale = maxs - mins))

index <- sample(1:nrow(mydata), round(0.75 * nrow(mydata)))
train <- scaled[index,]
test <- scaled[-index,]

nn <- neuralnet(Life.Exp~., data=train, hidden=c(5,4),linear.output = TRUE)


# Predict on test data
pred <- compute(nn, test)

# 환산 후 MSE 계산
pred.res <- pred$net.result * (max(mydata$Life.Exp) - min(mydata$Life.Exp))+
min(mydata$Life.Exp)
test.res <- (test$Life.Exp) * (max(mydata$Life.Exp) - min(mydata$Life.Exp)) +
min(mydata$Life.Exp)

MSE.nn <- sum((test.res - pred.res)^2) / length(test.res)
MSE.nn

# Plot the neural network
plot(nn)
```