

인공신경망(Artificial Neural Network)

<http://contents.kocw.net/KOCW/document/2016/yeungnam/leejeayoung/09.pdf>

<http://ocw.ulsan.ac.kr/CourseLectures.aspx?CollCd=11161&DeptCd=11178&CourseNo=20101G0298501>

10주차 신경망 모형

인간의 두뇌 신경(뉴런)들이 상호작용하여 경험과 학습을 통해서 패턴을 발견하고 이를 통해서 특정 사건을 일반화하거나 데이터를 분류하는데 이용되는 기계학습방법.

인간의 개입 없이 컴퓨터가 스스로 인지하고 추론하고, 판단하여 사물을 구분하거나 특정 상황의 미래를 예측하는데 이용될 수 있는 기계학습 방법

문자, 음성, 이미지 인식, 증권시장 예측, 날씨 예보 등 다양한 분야에서 활용.

(1) 생물학적 신경망 구조

인간의 생물학적 신경망의 구조

수상돌기로부터 외부 신호를 입력받고 시냅스에 의해서 신호의 세기를 결정한 후 이를 세포핵으로 전달하면 입력신호와 세기를 토대로 신경자극을 판정하여 축색돌기를 통해서 다른 신경으로 전달

(2) 인공신경망과 생물학적 신경망의 비교

[그림 15.7] 생물학적 신경망과 인공신경망

생물학적 신경망	인공신경망	역할
수상돌기	입력신호(x)	외부 신호 받음
시냅스	은닉층	신호의 세기(weight)결정
세포핵	활성함수	신경자극에 대한 판정, 전달여부 결정
축색돌기	출력신호(y)	출력신호를 보냄

(3) 가중치 적용

[그림 15.8] 외부 신호 입력에 대한 가중치 적용

(4) 활성화 함수

활성 함수는 망의 총합과 경계값(bias)를 계산하여 출력신호(y)를 결정

일반적으로 활성 함수는 0과 1사이의 확률분포를 갖는 시그모이드 함수(Sigmoid function)를 이용
현재 인공신경망에서는 시그모이드 함수를 이용한다.

[그림 15.9] 스텝 함수와 시그모이드 함수

(5) 퍼셉트론(Perceptron)

퍼셉트론: 생물학적인 신경망처럼 신경과 신경이 하나의 망 형태로 나타내기 위해서 여러 개의 계층으로 다층화하여 만들어진 인공신경망

[그림 15.10] 다층화한 퍼셉트론 모형

퍼셉트론의 계층(layer)별 구성요소

입력(input): x_1, x_2, x_3

입력층(input layer): 입력의 가중치(w)와 경계값(b)

은닉층(hidden layer): 입력의 가중치(w)와 경계값(b)

출력층(output layer): 입력의 가중치(w)와 경계값(b)

출력(output): o_1, o_2, o_3

인공신경망은 은닉층에서의 연산 과정이 공개되지 않기 때문에 블랙박스 모형으로 분류
따라서 어떤 원인으로 결과가 도출되었는지에 대한 설명을 할 수 없다.

(6) 인공신경망 기계학습과 역전파 알고리즘

출력값(o_1)과 실제 관측값(y_1)을 비교하여 오차(E)를 계산하고, 이러한 오차(E)를 줄이기 위해서
가중치(w)와 경계값(b)를 조정한다.

$$\text{오차}(E) = \text{관측값}(y_1) - \text{출력값}(o_1)$$

인공신경망(퍼셉트론)은 기본적으로 단방향 망(Feed Forward Network)으로 구성된다. 즉 입력층
→ 은닉층 → 출력층의 한 방향으로만 전파되는데 이런 전파 방식을 개선하여 역방향으로 오차를

전파하여 은닉층의 가중치와 경계값을 조정하여 분류정확도를 높이는 역전파(Backpropagation) 알고리즘을 도입.

역전파 알고리즘은 출력에서 생긴 오차를 신경망의 역방향(입력층)으로 전파하여 순차적으로 편미분을 수행하면서 가중치(w)와 경계값(b)등을 수정한다.

실습 (간단한 인공신경망 모델 생성)

nnet패키지에서 제공하는 nnet()함수
형식: nnet(formula, data, weights, size)

where

formula: $y \sim x$ 형식으로 반응변수와 설명변수 식

data: 모델 생성에 사용될 데이터 셋

weights: 각 case에 적용할 가중치(기본값: 1)

size: 은닉층(hidden layer)의 수 지정

<https://www.rdocumentation.org/packages/nnet/versions/7.3-15/topics/nnet>

1단계: 패키지 설치

```
install.packages("nnet")  
library(nnet)
```

nnet 패키지 설치

2단계: 데이터 셋 생성

```
df = data.frame(  # 데이터프레임 생성 - 입력 변수(x)와 출력변수(y)  
  x2 = c(1:6),  
  x1 = c(6:1),  
  y = factor(c('no', 'no', 'no', 'yes', 'yes', 'yes'))  
)
```

```
str(df)
```

3단계: 인공신경망 모델 생성

```
model_net = nnet(y ~ ., df, size = 1)
```

nnet()함수

where size: 은닉층의 수

4단계: 모델 결과 변수 보기

```
model_net
```

5단계: 가중치(weights)보기

```
summary(model_net)
```

6단계: 분류모델의 적합값 보기

```
model_net$fitted.values
```

7단계: 분류모델의 예측치 생성과 분류 정확도

```
p <- predict(model_net, df, type = "class")  
table(p, df$y)
```

실습 (iris 데이터 셋을 이용한 인공신경망 모델 생성)

1단계: 데이터 생성

```
data(iris)  
idx = sample(1:nrow(iris), 0.7 * nrow(iris))  
training = iris[idx, ]  
testing = iris[-idx, ]  
nrow(training)  
nrow(testing)
```

2단계: 인공신경망 모델(은닉층 1개와 은닉층 3개) 생성

```
model_net_iris1 = nnet(Species ~ ., training, size = 1)  
model_net_iris1
```

```
model_net_iris3 = nnet(Species ~ ., training, size = 3)
model_net_iris3
```

* 입력 변수의 값들이 일정하지 않거나 값이 큰 경우에는 신경망 모델이 정상적으로 만들어지지 않기 때문에 입력 변수를 대상으로 정규화 과정이 필요하다.

3단계: 가중치 네트워크 보기 – 은닉층 1개 신경망 모델

```
summary(model_net_iris1)
```

4단계: 가중치 네트워크 보기 – 은닉층 3개 신경망 모델

```
summary(model_net_iris3)
```

5단계: 분류모델 평가

```
table(predict(model_net_iris1, testing, type = "class"), testing$Species)
```

```
table(predict(model_net_iris3, testing, type = "class"), testing$Species)
```

classificationMetrics() 함수: 분류 metrics 생성

<https://www.rdocumentation.org/packages/performanceEstimation/versions/1.1.0/topics/classificationMetrics>

실습 (neuralnet패키지를 이용한 인공신경망 모델 생성)

neuralnet패키지는 역전파(Backpropagation)알고리즘을 적용할 수 있다. 또한 가중치 망을 시각화하는 기능도 제공한다.

neuralnet()함수

형식: neuralnet(formula, data, hidden = 1, threshold = 0.01, stepmax = 1e+05, rep = 1, startweights = NULL, learningrate.limit = NULL, algorithm = "rprop+")

where

formula: y ~ x형식으로 반응변수와 설명변수 식

data: 모델 생성에 사용될 데이터 셋

hidden = 1: 은닉층(hidden layer)의 수 지정

threshold = 0.01: 경계값 지정

stepmax = 1e+05: 인공신경망 학습을 위한 최대 스텝 지정

rep = 1: 인공신경망의 학습을 위한 반복 수 지정
startweights = NULL: 랜덤으로 초기화된 가중치를 직접 지정
learningrate.limit = NULL: backpropagation 알고리즘에서 사용될 학습비율을 지정
algorithm = "rprop+": backpropagation과 같은 알고리즘 적용을 위한 속성

<https://www.rdocumentation.org/packages/neuralnet/versions/1.44.2/topics/neuralnet>

1단계: 패키지 설치

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

neuralnet패키지 설치

2단계: 데이터 셋 생성

```
data("iris")  
idx = sample(1:nrow(iris), 0.7 * nrow(iris))  
training_iris = iris[idx, ]  
testing_iris = iris[-idx, ]  
dim(training_iris)  
dim(testing_iris)
```

iris데이터

3단계: 수치형으로 컬럼 생성

```
training_iris$Species2[training_iris$Species == 'setosa'] <- 1  
training_iris$Species2[training_iris$Species == 'versicolor'] <- 2  
training_iris$Species2[training_iris$Species == 'virginica'] <- 3
```

```
training_iris$Species <- NULL  
head(training_iris)
```

```
testing_iris$Species2[testing_iris$Species == 'setosa'] <- 1  
testing_iris$Species2[testing_iris$Species == 'versicolor'] <- 2  
testing_iris$Species2[testing_iris$Species == 'virginica'] <- 3
```

```
testing_iris$Species <- NULL  
head(testing_iris)
```

4단계: 데이터 정규화

4-1단계: 정규화 함수 정의

```
normal <- function(x) {  
  return((x - min(x)) / (max(x) - min(x)))  
}
```

4-2단계: 정규화 함수를 이용하여 학습데이터/검정데이터 정규화

```
training_nor <- as.data.frame(lapply(training_iris, normal))  
summary(training_nor)
```

```
testing_nor <- as.data.frame(lapply(testing_iris, normal))  
summary(testing_nor)
```

더 알아보기 (정규화 vs. 표준화)

정규화(Normalization): 데이터의 분포가 특정 범위 안에 들어가도록 조정하는 방법(예, 모든 값을 0과 1 사이의 값으로 재표현, 확률값)

$$(X - \text{Min}(X)) / (\text{Max}(X) - \text{Min}(X))$$

표준화(Standardization): 동일한 평균을 중심으로 관측값들이 얼마나 떨어져 있는지를 나타내는 방법 (예, 표준화 변수 Z를 이용하여 $N(0,1)$ 로 표현)

$$(X - \bar{X}) / \text{표준편차}$$

5단계: 인공신경망 모델 생성 - 은닉 노드 1개

```
model_net = neuralnet(Species2 ~ Sepal.Length + Sepal.Width +  
  Petal.Length + Petal.Width,  
  data = training_nor, hidden = 1)  
  
model_net  
plot(model_net)
```

시각화 포함

6단계: 분류모델 성능 평가

6-1단계: 모델의 예측치 생성 – compute()함수 이용

```
model_result <- compute(model_net, testing_nor[c(1:4)])  
model_result$net.result
```

compute()함수

<https://www.rdocumentation.org/packages/neuralnet/versions/1.44.2/topics/compute>
<https://www.rdocumentation.org/packages/neuralnet/versions/1.44.2/topics/predict.nn>

6-2단계: 상관관계 분석 – 상관계수로 두 변수 간 선형관계의 강도 측정

```
cor(model_result$net.result, testing_nor$Species2)
```

7단계: 분류모델 성능 향상 – 은닉층 노드 2개 지정, backprop속성 적용

7-1단계: 인공신경망 모델 생성

```
model_net2 = neuralnet(Species2 ~ Sepal.Length + Sepal.Width +  
                        Petal.Length + Petal.Width,  
                        data = training_nor, hidden = 2,  
                        algorithm = "backprop", learningrate = 0.01)
```

7-2단계: 분류모델 예측치 생성과 평가

```
model_result <- compute(model_net, testing_nor[c(1:4)])  
cor(model_result$net.result, testing_nor$Species2)
```

Neuralnet()함수 내 learningrate 속성은 역전파 알고리즘을 적용할 경우 학습비율을 지정하는 속성

Ch16 연습문제 (분류분석) 풀기 → 분류분석 종료 후