

# 회귀분석과 딥러닝의 예측 정확성에 대한 비교 그리고 딥러닝 모델 최적화를 위한 기법들의 중요성에 대한 실증적 분석

조민호\*

Comparison of Prediction Accuracy Between Regression Analysis and Deep Learning, and Empirical Analysis of The Importance of Techniques for Optimizing Deep Learning Models

Min-Ho Cho\*

## 요 약

인공지능 기법 중에서 딥러닝은 많은 곳에서 사용되어 효과가 입증된 모델이다. 하지만, 딥러닝 모델이 모든 곳에서 효과적으로 사용되는 것은 아니다. 이번 논문에서는 회귀분석과 딥러닝 모델의 비교를 통하여 딥러닝 모델이 가지는 한계점을 보여주고, 딥러닝 모델의 효과적인 사용을 위한 가이드를 제시하고자 한다. 추가로 딥러닝 모델의 최적화를 위해 사용되는 다양한 기법 중, 많이 사용되는 데이터 정규화와 데이터 셔플링 기법을 실제 데이터를 기반으로 비교 평가하여 딥러닝 모델의 정확성과 가치를 높이기 위한 기준을 제시하고자 한다.

## ABSTRACT

Among artificial intelligence techniques, deep learning is a model that has been used in many places and has proven its effectiveness. However, deep learning models are not used effectively in everywhere. In this paper, we will show the limitations of deep learning models through comparison of regression analysis and deep learning models, and present a guide for effective use of deep learning models. In addition, among various techniques used for optimization of deep learning models, data normalization and data shuffling techniques, which are widely used, are compared and evaluated based on actual data to provide guidelines for increasing the accuracy and value of deep learning models.

## 키워드

Artificial Intelligence, Machine Learning, Neural Network, Deep Learning, Model Optimization

인공 지능, 기계 학습, 신경망, 딥러닝, 모델 최적화

## I. 회귀분석과 딥러닝의 비교

인공지능에 관한 관심이 커지면서, 인공지능 기법이 번역이나 자율 주행과 같은 전통적인 영역 외에

도 기업의 의사결정이나 마케팅 분야에 적용되어 좋은 효과를 거두고 있다[1-4][10][11]. 하지만, 인공지능 기법을 사용하여 좋은 결과를 얻기 위해서는 많은 사항을 고려해야 한다.

\* 교신저자 : 중원대학교 컴퓨터공학과

• 접수일 : 2023. 02. 02

• 수정완료일 : 2023. 03. 08

• 게재확정일 : 2023. 04. 17

• Received : Feb. 02, 2022, Revised : Mar. 08, 2023, Accepted : Apr. 17, 2023

• Corresponding Author : Min-Ho Cho

Dept. Computer System Engineering, JungWon University,

Email : chominhokr@jwu.ac.kr

이번 연구에서는 인공지능 기법의 효과적 시행을 위하여 인공지능을 대표하는 통계적 분석기법과 신경망 기반의 딥러닝 기법을 가치와 용도 및 고려 요소 실증을 통해 비교한다. 구체적으로, 통계적 분석기법 중 회귀분석을 선정하였고, 딥러닝 모델 중 다층신경망을 사용하여 모델을 구성하였다. 그리고, 기업에서 사용하는 데이터를 사용해서 예측을 수행한 다음 결과를 비교, 분석하여 두 모델의 활용 방안에 대해 파악할 계획이다[5-8]. 딥러닝의 대표적인 사용처가 예측과 식별임을 감안하여 예측 분야에 중점을 두어 평가한다. 분석을 위하여 회귀분석은 R을 사용하였고, 다층신경망은 텐서플로와 테라스를 사용해서 환경을 구성하였다[7][9][11].

이번 연구의 목표는 다양한 인공지능 기법의 적용 방향성이다. 딥러닝 모델을 적용하여 좋은 결과를 얻기 위해서는 레이블 된 데이터가 수만 건 이상 필요하고, 많은 학습 시간이 필요하다. 이런 점에서 데이터의 준비에 걸리는 시간과 돈이 크다는 점을 강조하고자 한다. 현실적으로 수만 건의 레이블 된 데이터 준비가 어려운 환경에서 억지로 딥러닝을 적용하기보다는 통계적 기법을 사용하는 것이 좀 더 좋은 결과를 얻을 수 있다는 점을 보여주고자 한다.

회귀분석 모델과 딥러닝 모델을 평가하기 위하여 적용한 데이터는 데이터분석 분야에서 널리 사용되는 BostonHousing을 사용하였다.

BostonHousing 데이터는 미국 보스턴의 집값을 결정하는 14개의 요인과 507개의 데이터로 구성되어 있다. 데이터의 모양이나 크기가 현실에서 가능한 데이터의 형태를 갖추고 있다. 이번 연구를 위하여 BostonHousing 데이터 중에서 특별한 기여를 보이지 못하는 indus와 medv, age 항목을 제외한 11개의 요인을 대상으로 분석을 수행하였다.

연구 과정은 BostonHousing 데이터를 기반으로 회귀 모델과 다층신경망 모델을 만들고, 이것을 이용하여 예측을 수행한 후에 예측의 정확성을 MAE를 이용하여 측정 비교하였다. 모델의 정확성은 MSE(평균제곱오차)와 MAE(평균절대오차)로 측정 비교할 수 있는데, 이들의 의미를 정리하면 그림 1과 같다.

MSE(Mean Squared Error)

$$MSE = \frac{1}{n} \sum_{i=1}^n (f_i - y_i)^2$$

(n : data number,  
 $f_i$  : forecasting value of i th,  
 $y_i$  : actual value of i th)

MAE(Mean Absolute Error)

$$MAE = \frac{1}{n} \sum_{i=1}^n |f_i - y_i|$$

그림 1. MSE와 MAE의 의미  
 Fig. 1 The mean of MSE and MAE

이번 연구에서는 예측값과 실제값의 정확성을 좀 더 정확하게 보여주는 MAE를 사용하였다.

회귀분석 기법 중에서 다중회귀를 사용하여 BostonHousing 데이터에 대한 모델을 개발하고, 상세 내용은 그림 2와 같다. 그림 2에서 보여주는 회귀분석의 예측 모델은 수식으로 표현하면 그림 3과 같다. 그림 2에서 개발된 모델은 p-value가 0.05 보다 적으므로 통계적으로 유의한 모델임을 확인할 수 있다. 그리고 Adjust R-square가 0.7348이므로 73%의 예측 정확성을 가지는 모델이다.

```

R Console
> summary(reduced)

Call:
lm(formula = MEDV ~ LSTAT + B + PTRATIO + TAX + RM + DIS + RAD +
    TAX + PTRATIO + B + LSTAT, data = Regdata)

Residuals:
    Min       1Q   Median       3Q      Max
-15.5984  -2.7386  -0.5046   1.7273  26.2373

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  36.341145   5.067492   7.171 2.73e-12 ***
LSTAT        -0.108413   0.032779  -3.307 0.001010 **
B            0.045845   0.013523   3.390 0.000754 ***
PTRATIO      0.045845   0.013523   3.390 0.000754 ***
TAX          -17.376023  3.535243  -4.915 1.21e-06 ***
RM           3.801579   0.406316   9.356 < 2e-16 ***
DIS          -1.492711   0.185731  -8.037 6.84e-15 ***
RAD           0.299608   0.063402   4.726 3.00e-06 ***
TAX          -0.011778   0.003372  -3.493 0.000521 ***
PTRATIO     -0.946525   0.1129066  -7.334 9.24e-13 ***
B            0.009291   0.002674   3.475 0.000557 ***
LSTAT       -0.522553   0.047424  -11.019 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.736 on 494 degrees of freedom
Multiple R-squared:  0.7406,    Adjusted R-squared:  0.7348
F-statistic: 128.2 on 11 and 494 DF,  p-value: < 2.2e-16
    
```

그림 2. R에서 다중회귀 예측 모델 생성 화면  
 Fig. 2 Multiple regression prediction model creation screen in R

$$\begin{aligned}
 y = & 36.341145 \\
 & - 0.108413 * CRIM \\
 & + 0.045845 * ZN \\
 & + 2.718716 * CHAS \\
 & - 17.376023 * NOX \\
 & + 3.801579 * RM \\
 & - 1.492711 * DIS \\
 & + 0.299608 * RAD \\
 & - 0.011778 * TAX \\
 & - 0.946525 * PTRATIO \\
 & + 0.009291 * B \\
 & - 0.522553 * LSTAT
 \end{aligned}$$

그림 3. 다중회귀 예측 모델

Fig. 3 Multiple regression analysis model

회귀분석과 예측 정확성을 비교하기 위하여 딥러닝 모델 중 다층신경망을 사용하고 BostonHousing 데이터를 이용하여 모델을 만들었다. 제작된 모델은 텐서플로 2.0과 케라스를 사용으로 작성하였다. 모델의 개발 과정 전체를 표현하는 것은 큰 의미가 없고, 이해에 도움이 되지 않는다. 그래서 모델에 대한 부분과 학습에 대한 부분을 따로 정리하여 그림 4에 제시하였다.

```

# Model generation
model = Sequential()
model.add(Dense(64, activation='relu', input_shape=(13,)))
model.add(Dense(64, activation='relu'))
model.add(Dense(1))

[53] # Model Compile
model.compile(loss='mse', optimizer=Adam(lr=0.001), metrics=['mae'])

/usr/local/lib/python3.7/dist-packages/keras/optimizer_v2/adam.py:105:
super(Adam, self).__init__(name, **kwargs)

[54] # EarlyStopping Prepare
early_stop = EarlyStopping(monitor='val_loss', patience=30)

[55] # Learning
history = model.fit(train_data, train_labels, epochs=500,
                    validation_split=0.2, callbacks=[early_stop])
    
```

그림 4. 딥러닝(다층신경망) 예측 모델

Fig. 4 Deep learning prediction model

이제, 회귀분석의 모델과 딥러닝의 모델이 개발되었으므로 이들의 정확성에 대한 검증은 수행할 시점이다. 검증의 수행을 위하여, 모델의 개발에 사용하였던 BostonHousing 데이터 중에서 임의 추출된 10개의 데이터를 대상으로 예측을 수행하여 모델의 예측 정확성에 대해 평가하였다. 이에 대한 자료는 엑셀의 형태로 그림 5에 제공하였다.

Actual Value	Forecasting by regression	Forecasting by Neural Network
7.2	9.602556157	11
18.8	20.84918806	18
19	21.29365481	23
27	32.59713747	32
22.2	25.9468762	24
24.5	20.72410252	23
31.2	28.60739964	30
22.9	25.07379006	25
20.5	20.23145155	22
23.2	22.40086556	16
	Difference between actual and regression values	Difference between actual and neural network vallues
	2.402556157	3.8
	2.04918806	0.8
	2.293654813	4
	5.597137474	5
	3.746876205	1.8
	3.775897482	1.5
	2.592600358	1.2
	2.173790062	2.1
	0.268548455	1.5
	0.799134443	7.2
	Regression's MAE	Neural Network's MAE
	2.569938351	2.89

그림 5. 두 개 예측 모델의 정확성 비교

Fig. 5 Comparison of accuracy of 2 prediction models

그림 5의 내용은 임의 추출된 10개의 값에 대하여 회귀 모델에서 예측한 값과 딥러닝(=다층신경망)에서 예측한 값을 정리하고, 각 값이 실제 값과의 차이를 계산한 다음, 각 모델의 MAE를 계산하였다. 그림 5에서 확인할 수 있는 바와 같이 회귀 모델의 MAE는 2.5699이고, 딥러닝 모델에 의한 MAE는 2.89로 나타났다.

위의 결과를 기반으로 확인할 수 있는 점은 BostonHousing과 같이 엄청나게 크지 않은 데이터 또는 일반적인 환경에서 얻을 수 있는 데이터를 대상으로 하는 예측이나 식별의 경우에는 회귀를 포함한 통계 기반의 기법들이 딥러닝 기법에 비하여 정확도가 부족하지 않다는 것이다. 즉, 특정 분야에서 엄청나게 많은 레이블 데이터를 얻는 경우 딥러닝 모델이 좀 더 나은 정확도를 보일 것으로 기대가 된다. 하지만, 일반적인 환경에서 우리가 얻을 수 있는 데이터를 기반으로 하는 경우 통계 기반의 모델을 활용하면 빠르고, 간단하고, 정확한 결과를 얻어서 실무에 활용할 수 있다. 지면 관계상 회귀분석의 결과만을 제시하였지만, 랜덤포레스트, 서포트 벡터머신과 같은 기법들을 적용하여도 간단하지만, 높은 예측정확도를 얻을 수 있었다.

## II. 딥러닝 모델의 최적화를 위한 기법의 효과 비교

두 번째 주제는 딥러닝 모델의 최적화를 위해 사용하는 다양한 기법 중에서 가장 많이 사용되는 학습 데이터의 최적화 기법인 서플링과 정규화의 효과에 대한 실증적 검증이다.

연구는 데이터를 기반으로 딥러닝 모델을 만들어서 예측 정확성을 확인한다. 그리고, 개발된 딥러닝 모델에 서플링과 정규화를 모두 적용하여 모델을 최적화한 다음 예측 정확성을 확인한다. 마지막으로 처음에 개발된 딥러닝 모델에 정규화만 적용한 다음 예측 정확성을 확인하고 비교하는 방식으로 진행한다. 이러한 절차를 통하여 기본 모델의 정확도를 얻고, 이와 비교할 수 있는 최적화된 모델의 정확도를 얻어 비교할 수 있다. 마지막으로 최적화 방법의 비교로 두 개를 모두 적용한 경우와 한가지만 적용한 경우를 비교하게 된다.

앞에서 언급한 3개 딥러닝 모델의 개발 및 평가를 위한 데이터로는 BostonHousing 데이터를 사용한다. 사용하는 데이터는 14개 요인과 507개의 데이터로 구성되며, indus, medv, age 항목을 제외한 11개 항목만을 사용하여 분석을 수행하였다. 모델의 예측 정확성의 평가를 위하여 MAE를 사용하여 예측값과 실제값의 정확성을 확인하였다.

처음에 개발한 모델은 입력되는 데이터에 대하여 서플링과 정규화를 수행하지 않은 딥러닝 모델이다. 텐서플로와 케라스를 이용하여 개발된 모델의 모습이 그림 6에 제시되어 있다

그림 6을 통하여 BostonHousing 데이터를 기반으로 딥러닝(=다층신경망) 모델을 개발하고, 학습한 다음, 모델의 예측 정확성을 평가하는 과정을 확인할 수 있다.

그림 6에서 주의 깊게 보아야 하는 것은 입력 데이터에 대한 최적화를 하지 않은 상태에서의 딥러닝 모델의 예측 정확성은 MAE가 4.407이라는 점이다.

이제, 그림 6에 제시된 모델에 데이터의 서플링과 정규화 전처리를 수행하는 과정을 추가하여 모델을 최적화해 보자. 이에 대한 과정이 그림 7에 제시되어 있다.

```
[12] # Package import
from tensorflow.keras.datasets import boston_housing
from tensorflow.keras.layers import Activation, Dense, Dropout
from tensorflow.keras.models import Sequential
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.optimizers import Adam
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

[13] # Prepare data set
(train_data, train_labels), (test_data, test_labels) = boston_housing.load_data()

[14] # Check data sets shape
print(train_data.shape)
print(train_labels.shape)
print(test_data.shape)
print(test_labels.shape)

(404, 13)
(404,)
(102, 13)
(102,)

# Check data set
columns = ['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX', 'PTRATIO', 'B', 'LSTAT']
df = pd.DataFrame(train_data, columns=columns)
df.head()

CRIM  1.2247  0.0  8.14  0.0  0.538  6.142  91.7  3.9769  4.0  307.0  21.0  396.90  18.72
1  0.02177  82.5  2.03  0.0  0.415  7.610  15.7  6.2700  2.0  348.0  14.7  395.38  3.11
2  4.89822  0.0  18.10  0.0  0.631  4.970  100.0  1.3225  24.0  666.0  20.2  375.52  3.26
3  0.03961  0.0  5.19  0.0  0.515  6.037  34.5  5.9853  5.0  224.0  20.2  396.90  8.01
4  3.69311  0.0  18.10  0.0  0.713  6.376  88.4  2.5671  24.0  666.0  20.2  391.43  14.65

[16] # Check data sets label
print(train_labels[0:10])

[15.2  42.3  50.  21.1  17.7  18.5  11.3  15.6  15.6  14.4]

[17] # Model generation
model = Sequential()
model.add(Dense(64, activation='relu', input_shape=(13,)))
model.add(Dense(64, activation='relu'))
model.add(Dense(1))

[18] # compile of model
model.compile(loss='mse', optimizer=Adam(lr=0.001), metrics=['mae'])

/usr/local/lib/python3.7/dist-packages/keras/optimizers_v2/adam.py:105: UserWarning: The 'lr' argument is deprecated
super(Adam, self)._init__(**kwargs)

[21] # prepare of earlystopping
early_stop = EarlyStopping(monitor='val_loss', patience=30)

[22] # Learning of model
history = model.fit(train_data, train_labels, epochs=100,
                    validation_split=0.2, callbacks=[early_stop])

# Evaluation
test_loss, test_mae = model.evaluate(test_data, test_labels)
print("loss: {:.3f}#mae: {:.3f}".format(test_loss, test_mae))

4/4 [=====] - 0s 5ms/step - loss: 34.7837 - mae: 4.4066
loss: 34.784
mae: 4.407
```

그림 6. 딥러닝 기본 모델과 정확도

Fig. 6 Deep learning basic model and accuracy

딥러닝 모델에서 데이터의 최적화를 위한 방법은 여러 가지가 있다. 구체적으로, 입력 데이터의 조작 과정 또는 구성된 모델의 일부를 변경, 생략하는 과정 등이 있다. 이번 연구에서는 입력 데이터에 대한 최적화에 초점을 맞추어 진행한다. 다른 경우는 모델에 따라서 각기 다른 상황이 발생하므로 일관된 검증이 어렵다.

그림 7을 통하여 모델에 투입되는 데이터의 최적화가 어떻게, 어떤 단계에서 수행되는지 확인할 수 있고, 실제 수정된 데이터의 모습도 확인할 수 있다.

이제 개선된 모델의 예측 정확성에 대하여 알아

볼 시간이다. 이에 대한 자료는 그림 7에 제시되어 있다.

```

31 # Shuffling of data
order = np.argsort(np.random.random(train_labels.shape))
train_data = train_data[order]
train_labels = train_labels[order]

32 # Normalization of data
mean = train_data.mean(axis=0)
std = train_data.std(axis=0)
train_data = (train_data - mean) / std
test_data = (test_data - mean) / std

33 # data check after shuffling and normalization
column_names = ['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX', 'PTRATIO', 'B', 'LSTAT']
df = pd.DataFrame(train_data, columns=column_names)
df.head()
    
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO
0	-0.404234	3.728990	-1.432729	-0.256833	-1.249327	0.774315	-1.021665	2.260950	-0.511142	-0.902087	-1.536177
1	-0.334637	0.348906	-1.048738	-0.256833	0.763232	0.810991	1.110488	-0.852972	-0.511142	-0.853943	-2.491740
2	-0.371619	-0.483615	-0.720935	-0.256833	-0.455458	2.619712	0.332879	-0.417158	-0.165822	-0.595170	-0.489608
3	-0.396698	-0.483615	0.253663	-0.256833	-1.027395	-0.554489	-1.161419	0.375997	-0.511142	-0.047533	0.101931
4	-0.401703	2.886469	-1.408880	-0.256833	-1.309080	1.420378	-1.100501	1.759988	-0.856463	-0.462774	-2.673752

그림 7. 셔플링과 정규화가 추가된 딥러닝 모델과 데이터의 모습

Fig. 7 Deep learning model with shuffling and normalization, and what the data looks like

그림 7에서 개발된 모델을 이용하여 확인한 예측 정확성은 2.929이다. 상세 내용은 그림 8에 정리되어 있다.

```

34 # Evaluation
test_loss, test_mae = model.evaluate(test_data, test_labels)
print('loss: {:.3f} | mae: {:.3f}'.format(test_loss, test_mae))

4/4 [=====] - 0s 3ms/step - loss: 21.8017 - mae: 2.9287
loss: 21.802
mae: 2.929
    
```

그림 8. 개선된 딥러닝 모델과 정확도

Fig. 8 Revised deep learning model and accuracy

그림 8에서 확인할 수 있는 바와 같이 정규화와 셔플링을 통해 최적화된 딥러닝 모델은 일반 딥러닝 모델보다 확연히 개선된 예측 정확성을 보인다. 그림 6의 4.407이 그림 8에서 2.929로 개선된 것을 확인하기를 바란다.

이제 마지막으로 기본 딥러닝 모델에 데이터의 정규화만 적용하여 개선된 모델을 만들어 보자. 이 과정은 그림 9에 제시되어 있으며, 그림 7의 모델에서 셔플링을 주석처리 함으로써 얻을 수 있다.

그림 9의 평가 부분을 보면, 개발된 모델의 예측 정확도가 2.982인 것을 확인할 수 있다. 그림 7의 모델에서 단지 입력되는 데이터의 정규화 부분만을 적용하여 학습을 진행한 결과이다.

최종 결과는 셔플링을 함께 적용한 경우, 모델 예측정확도가 2.929이고, 정규화만 적용한 경우가 2.982이다.

```

39 # Shuffling of data
#order = np.argsort(np.random.random(train_labels.shape))
#train_data = train_data[order]
#train_labels = train_labels[order]

40 # Normalization of data
mean = train_data.mean(axis=0)
std = train_data.std(axis=0)
train_data = (train_data - mean) / std
test_data = (test_data - mean) / std

41 # data check after shuffling and normalization
column_names = ['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD', 'TAX', 'PTRATIO', 'B', 'LSTAT']
df = pd.DataFrame(train_data, columns=column_names)
df.head()

42 # Evaluation
test_loss, test_mae = model.evaluate(test_data, test_labels)
print('loss: {:.3f} | mae: {:.3f}'.format(test_loss, test_mae))

4/4 [=====] - 0s 4ms/step - loss: 24.4202 - mae: 2.9820
loss: 24.420
mae: 2.982
    
```

그림 9. 정규화만 적용된 딥러닝 모델과 정확도

Fig. 9 Deep learning model and accuracy with only normalization

이 두 모델의 차이는 상대적으로 크지 않아서 딥러닝 모델의 입력 데이터 최적화를 위한 기법 중 에서 정규화 부분이 큰 영향을 미치고, 데이터의 셔플링은 상대적으로 적은 영향을 미친다는 점을 확인할 수 있다.

딥러닝 모델에서 최적화 과정을 거치지 않은 모델의 예측정확도가 4.407이었던 점을 고려한다면, 딥러닝 모델에서 입력 데이터의 최적화 과정을 거치는 것이 중요함을 확인할 수 있다.

결론적으로 많은 최적화 기법 중에서 입력되는 데이터의 최적화 과정은 모델의 예측정확도를 높이는 데 중요한 기법임을 확인하였다. 입력 데이터의 최적화 기법 중 정규화가 중요하고, 셔플링도 정규화보다는 적기는 하지만, 확실한 효과를 보여준다는 점을 확인할 수 있다.

### III. 결론

이번 연구는 인공지능의 다양한 기법이 번역, 이미지 인식, 챗봇 분야와 같은 전통적 분야 외에 기업의 경영에 관련된 분야에서 많이 사용되는 최근의 흐름에 맞추어, 인공지능 기법이 기업에 적용되는 경우, 어떠한 가이드와 방향성이 필요한지 설명하고자 하였다.

우선, 인공지능은 신경망 기반의 딥러닝 외에도 통계에 기반한 기법과 강화학습 관련 분야를 포함하고 있다. 이번 연구에서는 기업에 적용할 수 있는 딥러닝과 통계 기반 기법을 비교하고, 이들을 기업



에 적용하기 위한 기준을 실험을 통해 보여주고 있다. 결론적으로 기업에서 인공지능 기법을 사용하는 경우, 투자 대비 효과의 관점에서 통계 관련 기법의 활용을 검토하는 것이 좋은 방안을 실무 데이터를 통하여 보여주었다. 이번 연구에서 사용한 다중 회귀 외에도 랜덤포레스트, 서포트벡터머신, 베이즈 안추론과 같은 간단하면서도 높은 정확도를 제공하는 통계적 기법이 있음을 고려해 주기 바란다.

다음으로 신경망 기반의 모델의 최적화를 위해서 사용하는 많은 기법 중에서 데이터에 관련하여 사용하는 최적화 기법인 서플링과 정규화를 비교 평가하였다. 결론적으로 신경망 기반의 모델은 학습하기 전에 학습 데이터에 대한 최적화 기법인 서플링과 정규화를 반드시 적용해야 하고, 이중, 정규화가 서플링보다 더 중요한 기법임을 실무 데이터를 이용하여 실증했다. 이번 연구의 결과가 인공지능 기법을 사용하고자 하는 기업의 담당자에게 유용하게 활용될 것으로 기대한다.

### References

[1] C. Kim, S. Choi and K. Kwahk, "Investigation of Research Trends in Information Systems Domain Using Topic Modeling and Time Series Regression Analysis," *Journal of Digital Contents Society*, vol. 18, no. 6, Oct. 2017, pp. 1143-1150.

[2] J. Liu, X. Kong, F. Xia, X. Bai, L. Wang, Q. Qing and I. lee, "Artificial Intelligence in the 21<sup>st</sup> century," *IEEE Access*, vol. 6, Mar. 2018, pp. 34403-34421.

[3] Y. Guo, Y. Liu, A. oerlemans, S. Lao, S. Wu and M. S. Lew, "Deep Learning for visual understanding : A review," *Neurocomputing*, vol. 187, Apr. 2016, pp. 27-48.

[4] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senoir, V. Vanhoucke, P. Nguyen, T. N. Sainath and B. Kingbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, issue. 6, Nov. 2012, pp. 82-97.

[5] S. Dreiseitl and L. O. Machado, "Logistic regression and artificial neural network

classification models : a methodology review," *Journal of Bio medical Informatics*, vol. 32, issue. 5-6, Oct. 2002, pp. 352-359.

[6] M. Cho, "A study on the history, classification and development direction of artificial intelligence," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 16, no. 2, April. 2021, pp. 307-312.

[7] J. Moon and Y. Lee, "Artificial Intelligence Computing Platform Design for Underwater Localization" *J. of the Korea Institute of Electronic Communication Sciences*, vol. 17, no. 1, Feb, 2022, pp. 119-124.

[8] C. Lee, H. Park, "A Comparative Study on the Accuracy of Important Statistical Prediction Technique of Marketing Data," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 16, no. 3, June. 2021, pp. 519-524.

[9] S. Jung and Y. Chung, "Comparison of audio event detection performance using DNN," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 13, no. 3, June 2018, pp. 571-578.

[10] M. Dahiya, "A Tool of Conversation : Chatbot," *International Journal of Computer Sciences and Engineering*, vol. 5, issue. 5, May 2017, pp. 158-161.

[11] J. Jhang and C. Zong, "Deep Neural Networks in Machine Translation: An Overview," *IEEE Intelligent Systems*, vol. 30, issue. 5, Sept.-Oct. 2015, pp. 16-25.

### 저자 소개



#### 조민호(Min-Ho Cho)

1989년 인하대학교 졸업(공학사)  
1989년 ~ 2012년  
HP Korea, Openwave, SK C&C  
에서 개발자, 컨설팅, PM의 역할을  
을 수행

2003년 숭실대학교 대학원 컴퓨터공학과 졸업  
(공학박사)

2013년~ 중원대학교 컴퓨터공학과 교수

※ 관심분야 : 인공지능 및 기계학습, 데이터분석, 소프트웨어공학, 데이터 마이닝 및 빅데이터