



이상치에 따른 불확실성에 관한 연구

Study on Uncertainty according to Outlier

심재원*^{ID} · 남정찬**^{ID} · 민희준*^{ID} · 김가연***^{ID} · 이효진**^{ID} · 정혜영****^{ID}

Jae-Won Shim, Jeong-Chan Nam, Hee-Jun Min, Ga-Yeon Kim, Hyo-Jin Lee and Hye-Young Jeong[†]

*한양대학교 일반대학원 응용수학과 석사과정, **한양대학교 ERICA 응용수학과 학사과정,

한양대학교 일반대학원 인공지능융합학과 석사과정, *한양대학교 ERICA 응용수학과 조교수

*MS Course, Graduate School of Applied Mathematics, Hanyang University

**BS Course, Applied Mathematics, Hanyang University ERICA

***MS Course, Graduate School of Applied Artificial Intelligence, Hanyang University

****Asistent Professor, Applied Mathematics, Hanyang University ERICA

Received: Jun. 23, 2021
Revised: Jul. 14, 2021
Accepted: Jul. 14, 2021
[†]Corresponding author
(hyjunglove@hanyang.ac.kr)

요 약

딥러닝 모델의 성능을 평가하는 지표는 정확도를 평가할 뿐, 해당 모델의 불확실성을 평가할 수 없다. 불확실성은 정확도와는 별개로, 안정성 또는 신뢰도의 의미를 가질 수 있다. 불확실성을 측정할 수 있다면 기존 모델의 성능을 평가하는 지표 외에 새로운 지표가 될 가능성을 가지고 있다. 불확실성을 측정하기 위한 딥러닝 모델로 베이지안 신경망을 사용한다. 본 논문은 불확실성이 딥러닝 모델에서 성능을 평가하는 지표가 될 수 있음을 확인해보려 하였다. 불확실성을 측정하기 위한 베이지안 신경망 모델로 베이지안 LSTM (Bayesian Long Short Term Memory)을 사용하였고, 모델의 학습에 실제 데이터를 적용하여 불확실성을 측정하였으며, 이를 기존 모델의 성능 지표와 비교분석 하였다. 실제 데이터를 이용한 실험에서 이상치가 추가됨에 따라 기존 모델의 성능지표는 증가하거나 감소하는 여러 가지 양상을 보이는 반면, 불확실성은 이상치에 따라 일관적으로 증가하는 결과를 보였다.

키워드 : 불확실성, 베이지안 신경망, 드랍아웃, 베이지안 LSTM

Abstract

Uncertainty is a separate measure of accuracy, and may have implications for stability or reliability. If uncertainty can be regulated, it has the potential to be a new metric for evaluating a performance of Deep learning models. Deep learning models for measuring uncertainty are Bayesian neural networks, which can be implemented by applying a drop-out scheme to each layer of existing deep learning models. This paper confirms that uncertainty can be an indicator of performance evaluation in deep learning models. We used Bayesian LSTM as a Bayesian neural network model to measure uncertainty, and measured uncertainty, which is compared to performance metrics from existing models.

Key Words : Uncertainty, Bayesian neural networks, Drop-out, Bayesian LSTM

본 논문은 한국연구재단 연구자지원사업(NRF-2019R111A1A01046810)과 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임. (No.2020-0-01343, 인공지능융합연구센터지원(한양대학교)).



This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted noncommercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. 서 론

딥러닝 (Deep learning) 모델은 입출력 데이터 간의 관계를 통계적으로 학습하는 모델로서, 여러 분야에서 분석 및 예측하기 위한 목적으로 다양하게 활용되고 있다. 현대의 딥러닝 모델은 지도학습을 통한 예측에서 좋은 성능을 보였다. 하지만 딥러닝 모델은 입출력 변수 간의 인과관계를 설명할 수 없으며 학습에 따른 가중치의 결과가 점추정으로 출력되어 예측하므로, 완전한 예측이라고 할 수 없다. 점추정을 통한 예측은 결과값 평균들을 출력한 것과 같으므로, 딥러닝 모델의 결과를 대표하고 있을 뿐, 모델

의 모든 출력값을 설명할 수 없다.

이는 딥러닝 모델의 예측 정확도와는 다른 의미를 가지며, 신뢰도 또는 안정성 파악의 문제를 해결할 수 없다. 딥러닝의 정확도는 높을 수 있으나 불확실성이 존재할 수 있고, 이러한 불확실성을 정량적으로 파악하지 않은 채 모델을 사용하는 것은 위험이 따를 수 있다. 딥러닝 모델의 불확실성은 딥러닝이 발전함에 따라 논의되기 시작했으며, 모델을 실제 산업 현장에 투입하려는 움직임에 따라 모델의 불확실성의 파악과 정량화가 최근 대두되기 시작하였다[1][2][3][4]. 딥러닝 모델의 산업화를 앞두고 모델이 가진 불확실성에 대한 연구가 이제 태동하기 시작하여 국내외에서 실제 데이터를 이용한 불확실성 파악의 적용 사례는 아직 많지 않다.

A Der Kiureghian(2009)은 불확실성을 발생 원인에 따라 내재적(Aleatory) 또는 인식론적(Epistemic)으로 분류했으며, 각각의 특성을 정의하였다[5]. 데이터 또는 지식의 부족에서 발생하는 불확실성은 인식론적 불확실성(Epistemic)으로, 딥러닝 모델에 대한 불확실성을 말한다. 인식론적 불확실성은 모델의 가중치를 결정하는 과정에서 해당 파라미터를 구하기 어려운 경우, 테스트 데이터가 기존 학습의 범위를 벗어난 경우 발생할 수 있다. 인식론적 불확실성은 모델 가중치를 변경하거나 훈련 데이터를 더 수집하는 방법으로 줄일 수 있다. 내재적 불확실성은 데이터의 불확실성을 말한다. 데이터 자체의 불확실성이므로, 데이터를 더 수집하여 감소시킬 수는 없다. 대신 내재적 불확실성은 데이터가 가진 결측치 또는 이상치에 의해 발생할 수 있는 불확실성으로, 데이터 전처리를 통해 불확실성을 줄일 수 있다.

모델의 불확실성에 대한 연구에서 Yarin Gal(2016)은 물리, 생물학, 제조 등의 광범위한 분야에서 모델의 불확실성을 표현하는 것의 중요성을 강조하며 BNNs 모델을 활용하여 딥러닝에서 실질적인 불확실성 추정치를 얻을 수 있는 도구를 개발하였다[6]. Y Ovadia(2019)은 이전의 불확실성을 정량화하는 방법에서는 데이터셋 이동(dataset shift)의 영향을 고려하지 않았음을 언급하며, 실험을 통해 out-of-distribution example에 대해 불확실성을 효율적으로 표현하는 방법을 논의했다[7]. Y Kwon(2020)은 점 추정(point estimation)을 기반으로 하여 불확실성을 정량화하는 자연스러운 방법을 제안했다[8].

BNNs(Bayesian Neural Networks, 베이지안 신경망)는 학습을 통해 가중치가 상수로 고정되는 것이 아닌 가중치들이 확률 분포를 따르는 확률 변수로 표현된다[9]. BNNs는 신경망의 가중치에 사전 분포를 배치하여 매개변수 함수 집합에 대한

분포를 유도하고, 모델의 가중치에 대한 분포를 추론하여 딥러닝 모델의 확률적 해석을 제공한다. BNNs는 공식화할 수 있으나 매개변수의 개수에 따라 많은 계산량을 요구하여 효율성의 문제를 가지며, 구현이 어렵다고 알려져 있다. 최근 계산 기술의 발달로 인해 BNNs의 구현에 대한 다양한 방법이 제시되고 있다. Yarin Gal(2016)이 제안한 드랍아웃(몬테카를로 드랍아웃, MonteCarlo Drop-out) 기법은 신경망의 과적합(overfitting)을 막는 정규화 방식의 하나로 일부 노드를 탈락시켜 학습하는 방법이다. 일반 신경망 모델의 모든 층에 드랍아웃 기법을 적용하면 BNNs의 학습과 같은 결과를 얻을 수 있다는 것이 증명되었다[10].

본 연구는 BNNs를 구현하기 위하여 기존 신경망에 드랍아웃 방법을 적용하여 BNNs와 같은 결과값을 출력할 수 있다는 점을 이용하였다. BNNs에 실제 데이터를 적용하여 불확실성이 기존 예측 성능 측도와는 다른 형식의 딥러닝 모델의 신뢰도를 나타낼 수 있는 가능성을 확인해보고자 하였다. 현재 구현된 BNNs 모델 중 시계열 데이터를 예측하는 LSTM 모델[11]의 신경망의 모든 층에 드랍아웃 방식을 적용한 베이지안 LSTM(Bayesian LSTM, B-LSTM) 모델을 사용하여 불확실성을 정량화하였다[12]. 모델이 학습할 데이터에 비정상적인 데이터가 포함되면 모델의 신뢰도가 감소할 수 있다는 가정하에 데이터에 비정상 값을 추가하여 모델을 학습하였다. 해당 모델은 BNNs의 특징과 같이 가중치를 분포로 출력하여 반복 연산을 통하여 결과값을 추출하고, 불확실성의 정량화를 통하여 기존 모델의 성능을 평가하는 지표와 비교분석 하였다.

본 연구의 실험에서 데이터는 광동제약과 대웅제약을 사용하였으며, 모델의 학습과 테스트, 비교에 사용된 기간은 2016년 3월 11에서 2021년 2월 25일로 1212일을 사용하였다. 전체 데이터의 70%를 학습 데이터(training data), 30%를 테스트 데이터(test data)로 사용하였다. 입력 데이터를 통한 불확실성의 차이를 확인하기 위하여 각 데이터에 이상치를 추가하여 불확실성의 변화를 비교 및 분석하고, 또한 이상치를 추가한 여부에 따라 모델의 성능을 함께 확인하기 위하여 모델 적합도와 예측 오차를 비교하였다.

본 논문은 다음과 같이 구성되어 있다. 2장은 본 연구에서 사용된 BNNs와 B-LSTM 모델의 기본 이론 및 구동 방식에 관해 설명하였다. 3장은 불확실성을 정량화하는 수식적인 이론에 관해 설명하였다. 4장은 실제 데이터를 적용한 실험을 통해 모델 적합도와 예측오차, 정량화한 불확실성을 비교하였다. 5장은 결론 및 향후 연구 방향에 대해 제시하였다.

2. Byesian LSTM

2-1. BNNs

BNNs는 모델의 가중치에 대한 사전분포(prior distribution)를 가정하고, 베이저안 추론 과정을 거쳐 가중치에 대한 사후분포(posterior)를 계산함으로써 신경망 모델의 예측치의 확률적 출력을 가능하게 한 딥러닝 알고리즘이다[9]. BNNs 가중치의 사후분포 $p(w|D)$ 는 식 (1)에서 볼 수 있듯 파라미터 w 에 대한 사전분포 $p(w)$ 와 우도 $p(D|w)$ 에 의해 결정된다. 가중치가 확률로 표현되므로, 결과값 또한 분포로 표현된다.

$$p(w | D) = \frac{p(D | w)p(w)}{p(D)} = \frac{\prod_{i=1}^N p(y_i | x_i, w)}{p(D)} \quad (1)$$

Kendall & Gal[13]은 데이터를 통해 불확실성을 정량화하는 인공신경망 모델을 제작하였다. BNNs는 위에서 언급한 바와 같이 분포를 추정하여 가중치를 출력한다. BNNs의 추정 분포를 추정하기 위하여 변분추론을 사용하였고, 이를 통해 사후예측 분포는 식 (2)와 같이 정의된다.

$$\hat{q}(\hat{y}|\hat{x}) = \int p(\hat{y}|f^w(\hat{x}))\hat{q}_\theta(w)dw \quad (2)$$

식 (2)에서 $f^w(\hat{x})$ 은 신경망의 출력값을 의미하며 $\hat{q}_\theta(w)$ 는 변분추론을 통해 얻은 추정된 분포를 의미한다. Kendall과 Gal[13]이 제시한 BNNs는 은닉층이 분리된 모델로 예측치의 평균($\hat{y}$)과 분산($\hat{\sigma}$)을 출력한다. 평균($\hat{y}$)과 분산($\hat{\sigma}$)은 손실함수(loss function)에 포함되어 분산($\hat{\sigma}$)을 출력하기 위한 별도의 단계 없이 학습할 수 있으며, 식(3)과 같이 표현된다.

$$Loss = \sum_i \frac{1}{2\sigma^2} \|y_i - \hat{y}_i\|^2 + \frac{1}{2} \log \hat{\sigma}_i^2 \quad (3)$$

서론에서 언급한 바와 같이, 해당 신경망을 구현하기에 파라미터 개수에 따라 많은 계산량의 한계를 가지고 있다. 이를 위한 대안으로 기존 신경망의 모든 층에 드롭아웃 방식을 적용한 학습을 통하여 BNNs와 같은 결과값을 출력하게 한다[10].

2-2. Bayesian LSTM

RNN(Recurrent Nueral Network)은 시계열 분석을 위한 신경망으로, 시간적 상관관계가 있는 데이터에서 사용되는 인공신경망을 말한다. RNN은 학습 데이터를 통해 시간에 따른 주기적인 패턴을 익히고 이를 이용하여 예측한다. 하지만 데이터의 기간이 길어지면 역전파 시 그라디언트 소실 문제(vanishing gradient)를 일으키는 것으로 알려져있다. 이를 극복하기 위한 구조가 LSTM(Long Short Term Memory)이다. LSTM은 RNN의 구조에 셀 상태(Cell State)를 추가하여 그라디언트 소실 문제를 해결하였다[11].

LSTM의 셀 상태 안에는 입력 게이트(Input Gate), 망각 게이트(Forget Gate), 출력 게이트(Output Gate) 3종류의 게이트가 있다. 망각 게이트에는 입력값이 시그모이드 함수를 거쳐 0과 1 사이의 값을 가지게 된다. 이 값에 따라 이전 단계의 값을 이용할지, 버릴지를 결정하게 된다. 결정된 결과에 따라 입력 게이트에서 출력 게이트와 동일하게 입력값을 받게 된다. 이전 단계와 마찬가지로 입력값은 시그모이드 함수를 거쳐 0과 1 사이의 값을 가지게 되는데 값에 따라 이전 셀 상태에 얼마나 새로운 정보를 반영할지를 결정한다. 이때, 1의 출력값이 나오면 해당 값은 tanh 함수를 거친 출력값이 나오게 되어 셀 상태에 더해진다. 이후 출력 게이트에서는 입력값이 들어가면 현재 단계의 출력을 얼마나 반영할지 시그모이드 함수를 거쳐 0과 1 사이의 값을 통해 결정되며 tanh 함수에 의해 C_t 에서 나온 값을 -1과 1 사이의 값으로 변환한다. 이러한 게이트를 거쳐 결과값 h_t 는 출력값으로 출력되거나 다음 상태(state)의 입력값으로 들어간다. 위의 단계를 식 (4)와 같이 정의한다.

$$\begin{pmatrix} f_t \\ i_t \\ \tilde{C}_t \\ C_t \\ o_t \\ h_t \end{pmatrix} = \begin{pmatrix} \sigma(W_f[h_{t-1}, x_t] + b_f) \\ \sigma(W_i[h_{t-1}, x_t] + b_i) \\ \tanh(W_C[h_{t-1}, x_t] + b_C) \\ f_t * C_{t-1} + i_t * \tilde{C}_t \\ \sigma(W_o[h_{t-1}, x_t] + b_o) \\ o_t * \tanh(C_t) \end{pmatrix} \quad (4)$$

LSTM 알고리즘에 드롭아웃 기법을 적용하여 BNNs의 출력 방식과 같게 기존에 고정되어 있던 가중치를 분포로 출력하게 재현한 B-LSTM 모델(Bayesian LSTM)은 이미 구현되어있다[12]. 식 (5)는 이 B-LSTM의 수식을 표현한 것이다. 기존 LSTM의 모든 수식에 드롭아웃 연산자를 적용하였으며, 모든 정보가 없어지거나 전 단계에서의 정보를 기억하지 못하는 현상은 생기지 않으며

LSTM의 기본 원리인 장기기억은 유지한다.

$$\begin{pmatrix} f_t \\ i_t \\ \tilde{C}_t \\ C_t \\ o_t \\ h_t \end{pmatrix} = \begin{pmatrix} \sigma(W_f[D(h_{t-1}), x_t] + b_f) \\ \sigma(W_i[D(h_{t-1}), x_t] + b_i) \\ \tanh(W_C[D(h_{t-1}), x_t] + b_C) \\ f_t * C_{t-1} + i_t * \tilde{C}_t \\ \sigma(W_o[D(h_{t-1}), x_t] + b_o) \\ o_t * \tanh(C_t) \end{pmatrix} \quad (5)$$

3. 불확실성

출력한 결과값을 이용하여 모델의 불확실성을 비교하기 위하여 Kendall & Gal[13]이 제안한 불확실성의 정량화 방법을 사용하였다. 정량화된 불확실성은 불확실성 발생의 원인을 통하여 내재적 불확실성(Aleatoric Uncertainty)과 인식적 불확실성(Epistemic Uncertainty)으로 분류할 수 있다.

Kendall & Gal[13]은 두 불확실성을 구분하여 추정하는 방법을 다음과 같이 제시하였다. BNNs 모델에서 불확실성은 모델이 추정한 예측값 $\hat{y}$ 의 분산을 의미하며, 분산식은 (6) 과 같다.

$$Var[\hat{y}] = E[(\hat{y})^T(\hat{y})] - E[\hat{y}]^T E[\hat{y}] \quad (6)$$

위의 식을 이용하여 불확실성을 정량화하기 위한 식 (7), (8), (9)로 정의할 수 있다. 다음 불확실성을 구하는 식에서 T 는 모델이 반복연산 거쳐 출력하는 샘플링(sampling) 횟수를 뜻하며, y_t 는 샘플링을 통해 출력된 예측값의 평균값, σ_t 는 샘플링별 예측값들의 표준편차 값을 의미한다.

$$\widehat{Var}[\hat{y}] = \frac{1}{T} \sum_{t=1}^T \hat{y}_t^2 - \left(\frac{1}{T} \sum_{t=1}^T \hat{y}_t \right)^2 + \frac{1}{T} \sum_{t=1}^T \sigma_t^2 \quad (7)$$

$$U_{epistemic} : \frac{1}{T} \sum_{t=1}^T \hat{y}_t^2 - \left(\frac{1}{T} \sum_{t=1}^T \hat{y}_t \right)^2 \quad (8)$$

$$U_{aleatoric} : \frac{1}{T} \sum_{t=1}^T \sigma_t^2 \quad (9)$$

이는 드랍아웃을 통한 확률적 처리 과정을 이용한 BNNs 의 추론 과정을 모사한 것이라고 할 수 있다. 신경망 가중치의 몬테카를로 근사를 통해 신

경망 모델의 불확실성을 분리하여 정량화할 수 있게 하였다. 위 불확실성은 예측값 $\hat{y}$ 을 통해 얻을 수 있으며, 이는 기존 통계적 개념을 통하여 아래의 식 (10), (11)로 나타낼 수 있다.

$$U_{epistemic} : E[\hat{\sigma}^2] \quad (10)$$

$$U_{aleatoric} : Var[\hat{\mu}] \quad (11)$$

위 식에서 확인할 수 있듯, 출력된 데이터의 분산과 평균을 이용하여 불확실성을 정량화할 수 있는 것을 알 수 있다.

본 연구에서 사용하는 B-LSTM 모델은 기존 LSTM 신경망에 드랍아웃을 적용하여 BNNs와 같은 결과를 재현한 모델로 가중치를 분포로 출력하므로 불확실성을 정량화할 수 있다. 본 연구의 불확실성 또한 출력된 예측값을 위 식을 이용하여 모델들의 불확실성을 정량적으로 파악하였다.

4. 실험 및 결과

4.1 B-LSTM 모델 제작 및 성능 비교

본 절에서는 B-LSTM 모델에 실제 데이터를 적용해 보기 위하여 주식 데이터를 대상으로 실험을 진행하였다. 표1은 사용한 주식 데이터와 불확실성의 차이를 보기 위하여 이상치(outlier)를 추가한 데이터에 대한 설명이다. 광동제약과 대웅제약 데이터의 변수 중 입력 변수는 종가(Close)를 사용하였다. 모델은 주식의 종가와 시간에 따라 이후 종속 변수인 주식의 종가(Close)를 예측하도록 제작되었다.

본 연구의 목적은 모델의 불확실성을 정량적으로 측정하고, 학습 데이터의 이상치가 모델의 불확실성에 미치는 영향을 확인하는 것이다. 데이터에 이상치를 추가함에 따라 모델들의 신뢰도 또는 정확도에 영향을 미친다고 가정하였다. 데이터 내에서 학습 데이터와 테스트 데이터를 나누어 사용하였으며, 학습데이터의 비율은 전체 데이터의 70%로 설정하였다. 모델들의 차이를 두기 위하여 학습 데이터 중 하나의 데이터에만 이상치를 추가하여 불확실성을 출력하였다(표1). #1은 광동제약 데이터, #2는 #1에 이상치를 추가한 데이터, #3은 대웅제약 데이터, #4는 #3에 이상치를 추가한 데이터이다. 이상치는 실제 주식의 상한가를 이용하여 이틀 동안 연속으로 최고 상한가를 달성한 상황을 가정하였다. 이상치는 훈련 데이터에만 적용하였으며,

훈련 데이터 이외의 신경망 구조, 최적화 알고리즘 등의 학습 조건은 4개 모델 모두 동일하게 구성하였다. 해당 모델의 초기 파라미터 설정은 표2와 같이 설정하였다. 표2에서 Number of experiment는 3절에서 설명한 샘플링 횟수를 의미한다.

표 1. B-LSTM 모델 훈련 및 테스트 데이터
Table 1. Dataset information

Model	Outliers	# of train data	Train data period	# of test data	Test data period
#1	X	855	2016.03.11 ~ 2019.09.11	357	2019.09.16 ~ 2021.2.25
#2	O	855	2016.03.11 ~ 2019.09.11	357	2019.09.16 ~ 2021.2.25
#3	X	855	2016.03.11 ~ 2019.09.11	357	2019.09.16 ~ 2021.2.25
#4	O	855	2016.03.11 ~ 2019.09.11	357	2019.09.16 ~ 2021.2.25

표 2. 초기 B-LSTM 모델 파라미터
Table 2. First B-LSTM model parameter

parameter	setting
epoch	150
activation function	tanh
number of hidden layers(nodes)	1 (128 - 32)
optimization algorithm	Adam
batch-size	128
dropout rate	0.5
learning rate	0.01
Number of experiment	100

모델 학습 이후 테스트 데이터를 이용하여 B-LSTM 모델의 예측 성능을 비교하였다. 그림 1은 두 가지 경우(원 데이터, 이상치를 포함한 데이터)의 네 가지 모델을 학습 데이터 예측값(파란색 실선), 테스트 데이터 예측값(빨간색 실선), 실제값(녹색 실선)으로 표현한 그래프이다. 드랍아웃을 적용하여 구현하였기 때문에 가중치가 무작위로 추출되며 매번 다른 결과를 도출한다. 즉 그림 1은 해당 예측의 100번의 반복 연산을 통해 도출한 결과값의 평균을 이용하여 예측 성능을 구한 결과이다. 그래프상에서 4개 모델의 예측 성능을 확인하기 위해 예측오차 MSE (Mean Squared Error)와 결정계수(coefficient of determination, R^2)을 사용하였다. 예측오차의 확인 편의성을 위하여 로그를 적용하여 수치를 비교하였다. 네 모델 중 첫 번째 데이터를 이용한 두 모델의 결정계수는 0.80, 0.45로 큰 차이를 보였고, 예측오차 또한 0.007, 0.016으

로 차이를 보였다. 오히려 두 번째 데이터를 이용한 모델이 결정계수가 0.71, 0.95, 예측오차가 0.026, 0.008로 앞의 첫 번째 모델과 상반된 결과를 출력하였다.

표 3. B-LSTM 모델 성능 비교
Table 3. Comparing of models' performance

Model	Data (2016.03.11~2021.2.25)	
	MSLE	R^2
#1	0.007	0.08
#2	0.016	0.45
#3	0.026	0.71
#4	0.008	0.95

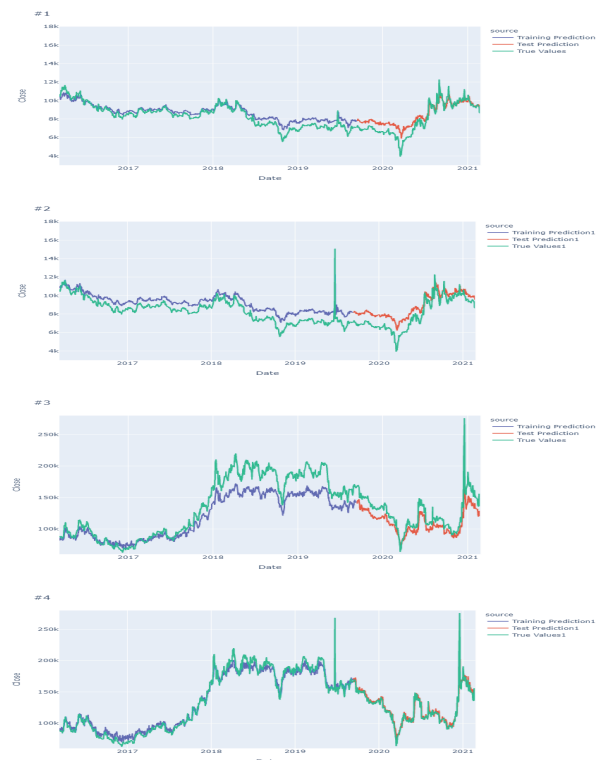


그림 1. 모델 예측 결과
Fig 1. Result of Prediction

4.2 모델의 불확실성 분석

4.1절에서는 주식 데이터를 이용하여 B-LSTM 모델을 제작하고, 테스트 데이터를 이용하여 성능을 비교하였다. 실험 결과에 따르면 데이터가 이상치를 보유함에 따라 #2 모델에 비해 #1 모델이 더 준수한 성능을 보였고, #3 모델과 #4 모델은 이와 반대로 이상치를 보유한 모델이 더 나은 성능을 보였다. 해당 성능 지표는 반복 연산을 통해 얻은 결과값의 평균을 이용하여 구한 것으로, 예측 성능의 척도를 완전히 확신할 수 없다는 것을 보여주고

있다. 또한 데이터 또는 모델의 높은 불확실성을 가지고 있다면 기존 예측 성능 척도는 성능 비교의 역할을 제대로 하지 못할 수 있다.

본 절에서는 모델이 내재하고 있는 불확실성을 정량화하여 분석하였다. 불확실성은 데이터에 따라 매우 큰 값이 출력될 수 있어 심의 거듭제곱을 통해 소수점 이하 둘째 자리까지 표현하였다.

불확실성은 두 가지 종류로 나눌 수 있고, 이는 각각의 특징을 가지고 있으므로 두 가지 종류의 불확실성으로 분리하여 정량화할 필요가 있다. 내재적 불확실성은 데이터에서 얻을 수 있는 변화하는 불확실성이므로 각각의 모델에서도 증가함을 표 4에서 확인할 수 있다. 내재적 불확실성은 각 반복 연산하여 출력된 결과값의 분산의 평균이고, 이는 최댓값과 최솟값에 영향을 받을 수 있다. 하지만 표 4에서 볼 수 있듯이 최댓값과 최솟값의 차이가 크지 않고, 이는 어느 한 값에 의해 내재적 불확실성이 변화한 것이 아니라 전체적인 내재적 불확실성이 변화하였음을 의미한다.

표 4. 내재적 불확실성 정량화
Table 4. Quantified Aleatoric Uncertainty

Model	Aleatoric		
	Amount (mean)	Min.	Max.
#1	1.07×10^{12}	1.03×10^{12}	1.12×10^{12}
#2	2.25×10^{12}	2.12×10^{12}	2.39×10^{12}
#3	1.35×10^{17}	1.26×10^{17}	1.41×10^{17}
#4	4.74×10^{17}	4.45×10^{17}	5.02×10^{17}

인식론적 불확실성은 모델의 가중치가 가지는 불확실성이다. 가중치 또한 데이터를 통하여 얻게 되므로 이상치를 포함한 모델의 인식론적 불확실성이 증가함을 예측하였다. 표 5에서 볼 수 있듯이 인식론적 불확실성 또한 이상치를 가진 모델이 증가함을 확인할 수 있다. 인식론적 불확실성은 3장에서 설명한 바와 같이 반복 연산 평균값의 분산이므로 최댓값과 최솟값의 영향을 받지 않는다.

표 5. 인식론적 불확실성 정량화
Table 5. Quantified Epistemic Uncertainty

Model	Epistemic
	Amount (variance)
#1	2.42×10^1
#2	2.70×10^1
#3	6.10×10^3
#4	1.14×10^4

두 불확실성의 확실히 분리하고 비교하기 위하여 모델의 드랍아웃율(dropout rate)을 변화하여 비교하여 보았다. 드랍아웃율은 각 층에서 탈락시키는 노드의 비율을 의미한다. 비율은 25%(0.25),

50%(0.5), 70%(0.7)로 차이를 두었으며, 다른 조건은 이전과 모두 같게 하였다.

표 6은 각 모델의 드랍아웃율에 따른 인식론적 불확실성의 차이를 보여주고 있다. 이는 베이지안 신경망을 구현할 때, 모델의 모든 층에 드랍아웃을 적용하며 얼마나 많은 노드를 탈락시키느냐에 따라 인식론적 불확실성에서 큰 변화를 줄 수 있음을 확인할 수 있다.

표 6. 드랍아웃율에 따른 인식론적 불확실성 정량화
Table 6. Quantified Epistemic according to Dropout rate

Model	Epistemic		
	0.25	0.5	0.7
#1	9.47	24.18	29.55
#2	11.99	27.02	56.22
#3	2.53×10^3	6.10×10^3	1.20×10^4
#4	4.10×10^3	1.14×10^4	2.11×10^4

내재적 불확실성 또한 변화를 확인하기 위하여 비교하였다. 내재적 불확실성은 인식론적 불확실성보다 작은 폭의 변화를 보이며, 전체적으로 비슷한 양상을 가짐을 표 7을 통해 알 수 있다. 이를 통해 내재적 불확실성과 인식론적 불확실성의 구분 또한 가능할 것으로 판단된다.

표 7. 드랍아웃율에 따른 내재적 불확실성 정량화
Table 7. Quantified Aleatoric according to Dropout rate

Model	Aleatoric		
	0.25	0.5	0.7
#1	1.60×10^{12}	1.07×10^{12}	2.12×10^{12}
#2	2.45×10^{12}	2.25×10^{12}	3.39×10^{12}
#3	1.47×10^{17}	1.35×10^{17}	1.75×10^{17}
#4	3.80×10^{17}	4.74×10^{17}	4.05×10^{17}

드랍아웃율을 변화시킨 모델들의 예측 성능과 모델 적합도 또한 #1 모델이 0.008로 모두 같거나 과 같이 큰 차이를 보이지는 않았다. #2 모델의 경우 0.013, 0.014, 0.015로 작은 차이를 보였다. #3 모델은 0.026, 0.033, 0.020으로 차이를 보였으나, #4 모델은 0.003으로 모두 같은 값을 나타내기도 하였다. 모델 적합도는 #1 모델에서 0.79, 0.76, 0.79로 큰 차이를 보이지 않았으며, #2가 0.57, 0.61, 0.50, #3은 0.70, 0.74, 0.67, #4는 0.97로 모두 같은 값을 보이는 등, 일관되지 않지만, 전체적으로 준수한 성능을 보였다.

이에 비해 수치로 알 수 있듯이 불확실성의 양이 이상치 여부에 따라 일관적인 차이가 있음을 확인할 수 있다. 해당 결과는 4.1절에서 구한 모델

의 성능과 다른 결과를 보이고 있으며, 이는 모델이 내재한 불확실성이 예측 성능을 나타내는 지표와 다른 평가의 지표로 사용할 수 있다는 것을 확인할 수 있다.

5. 결론 및 향후 연구

본 논문은 불확실성이 인공 신경망 모델에서 기존 예측 성능 지표 이외에 모델의 신뢰도를 확인하는 새로운 지표가 될 수 있음을 확인하고자 하였다. 또한 실제 데이터 적용 사례가 많지 않음을 인지하고, 실제 데이터를 적용하여 불확실성을 비교분석해보았다. 해당 불확실성의 정량화를 위해 BNNs를 구현하려 하였다. 이를 위해 기존 LSTM 모델에 드롭아웃을 적용하여 BNNs와 같은 방식의 결과값을 출력하도록 구현된 B-LSTM을 사용하고, 데이터에 이상치를 추가하여 비정상적인 데이터를 조성하였다. 학습된 B-LSTM 모델의 예측값을 반복 연산을 통해 추출하였으며, 반복 연산을 통해 얻은 예측값을 이용하여 불확실성을 구하였다. 불확실성이 기존 예측성능을 나타내는 지표와 다른 평가를 내릴 수 있는 가능성을 확인하기 위하여 기존 예측 성능 척도와 모델의 예측성능이 보이는 결과와 불확실성 사이에서 차이를 확인하였으며, 두 가지로 분리한 불확실성을 함께 제시하였다.

실험을 통해 본 논문에서 불확실성이 데이터의 작은 변화에도 차이가 있음을 확인하였다. 또한 모델의 예측성능의 척도가 보여주는 값과 불확실성이 보여주는 수치의 차이가 있음을 통해 불확실성이 다른 의미를 내포하고 해석할 수 있음을 볼 수 있었다. 이상치 여부에 따라 내재적 불확실성과 인식론적 불확실성의 변화는 이상치의 개수에 따른 차이와 이상치 제거를 통한 불확실성의 양의 차이를 확인할 필요성을 시사한다. 향후 불확실성의 정량화가 단순 통계적 의미를 지니는 것이 아니라 수학적 의미 또한 가질 수 있는 방법에 대해 연구하며 또한, 실제 주가 조작과 같은 다수 데이터의 변화 또는 이상치 탐지와 같은 방법을 통해 데이터에서 이상치를 제거하였을 때 불확실성에 미치는 영향에 관한 연구를 진행하고자 한다.

References

- [1] Niculescu-Mizil, Alexandru, and Rich Caruana. "Predicting good probabilities with supervised learning." *Proceedings of the 22nd international conference on Machine learning*. pp. 625-632, 2005.
- [2] Nuzzo, Regina. "Scientific method: statistical errors." *Nature News* 506.7487 pp. 150-152, 2014.

- [3] Bojarski, Mariusz, et al. "End to end learning for self-driving cars", arXiv, 2016.
- [4] Ren, Jie, et al. "Likelihood ratios for out-of-distribution detection", arXiv, 2019.
- [5] Der Kiureghian, Armen, and Ove Ditlevsen. "Aleatory or epistemic? Does it matter?." *Structural safety*, vol. 31, no. 2, pp. 105-112, 2009.
- [6] Gal, Yarin. "Uncertainty in deep learning.", doctoral degree, University of Cambridge 1.3, April. 2016.
- [7] Ovidia, Yaniv, et al. "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift.", arXiv, 2019.
- [8] Kwon, Yongchan, et al. "Uncertainty quantification using Bayesian neural networks in classification: Application to biomedical image segmentation.", *Computational Statistics & Data Analysis*, vol. 142, 2020.
- [9] Neal, R. M. Bayesian learning for neural networks, Springer Science & Business Media, 2012.
- [10] Gal, Yarin, and Zoubin Ghahramani. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning.", *Proceedings of The 33rd International Conference on Machine Learning*, New York, USA, pp. 1050-1059, June, 2016.
- [11] Hochreiter, S., & Schmidhuber, J., "Long short-term memory", *Neural computation*, vol. 9, no. 8, pp. 1735-1780, 1997.
- [12] Fortunato, Meire, Charles Blundell, and Oriol Vinyals. "Bayesian recurrent neural networks." arXiv, 2017.
- [13] Kendall, Alex, and Yarin Gal. "What uncertainties do we need in bayesian deep learning for computer vision?." arXiv, 2017.

저 자 소 개



정혜영(Hye-Young Jung)

2014년: 연세대학교 이학박사

2014년~2016년: 서울대학교 통계학과 박사후 연구원

2016년~2020년: 서울대학교 기초교육원 강의교수

2020년~현재: 한양대학교 ERICA 응용수학과 조교수

관심분야 : Fuzzy modeling, Topological data analysis, Machine/deep learning,
ORCID Number : 0000-0003-0569-370X
E-mail : hyjunglove@hanyang.ac.kr



심재원(Jae-Won Shim)

2014년~2021년 : 한양대학교 ERICA
응용수학과 이학사
2021년~현재 : 한양대학교 일반대학원
응용수학과 석사과정

관심분야 : Fuzzy modeling, Machine/deep learning, Soft Computing
ORCID Number : 0000-0002-1018-2771
E-mail : anald@hanyang.ac.kr



이호진(Hyo-Jin Lee)

2017년~현재 : 한양대학교 ERICA
응용수학과 학부생

관심분야 : Fuzzy, Machine/deep learning, Soft Computing
ORCID Number : 0000-0002-2422-0572
E-mail : wlsdl8888@naver.com



남정찬(Jeong-Chan Nam)

2016년~현재 : 한양대학교 ERICA
응용수학과 학부생

관심분야 : Fuzzy, Machine/deep learning, Soft Computing
ORCID Number : 0000-0002-3944-7550
E-mail : jcnam1022@gmail.com



민희준(Hee-Jun Min)

2015년~2021년 : 한양대학교 ERICA
응용수학과 이학사
2021년~현재 : 한양대학교 일반대학원
응용수학과 석사과정

관심분야 : Fuzzy modeling, Machine/deep learning, Soft Computing
ORCID Number : 0000-0003-4897-9812
E-mail : minsun1129@hanyang.ac.kr



김가연(Ga-Yeon Kim)

2016년~2021년 : 한양대학교 ERICA
응용수학과 이학사
2021년~현재 : 한양대학교 일반대학원
인공지능융합학과
석사과정

관심분야 : Machine/deep learning, Soft Computing, Data mining
ORCID Number : 0000-0003-0117-6019
E-mail : rkds5485@naver.com