

LSTM을 활용한 고속도로 교통정보 예측 모델 개발 방법론

Methodology for Developing a Predictive Model for Highway Traffic Information Using LSTM

이요섭* · 진형석** · 김예진*** · 박성호**** · 윤일수*****

* 주저자 : 아주대학교 교통공학과 석박사 통합과정
 ** 교신저자 : (주) 이젠시스 해외컨설팅팀 이사
 *** 공저자 : 아주대학교 교통공학과 석박사 통합과정
 **** 공저자 : 아주대학교 혁신융합단 연구조교수
 ***** 공저자 : 아주대학교 교통시스템공학과 교수

Yoseph Lee* · Hyoung-suk Jin** · Yejin Kim* · Sung-ho Park*** · Ilsoo Yun*

* Dept. of Transportation Eng., Ajou Univ.
 ** Overseas Consulting Team, EZENSYS Co.,Ltd
 *** Dept. of Center for Convergence and Open Sharing, Ajou Univ.

† Corresponding author : Hyoung-suk Jin, hsjin@ezensys.co.kr

Vol. 22 No.5(2023)
 October, 2023
 pp.01~18

pISSN 1738-0774
 eISSN 2384-1729
<https://doi.org/10.12815/kits.2023.22.5.1>

Received 20 July 2023
 Revised 16 August 2023
 Accepted 21 August 2023

© 2023. The Korea Institute of
 Intelligent Transport Systems. All
 rights reserved.

요약

최근 빅데이터 및 딥러닝 기술의 발전으로 다양한 교통정보가 널리 수집 및 활용되고 있다. 특히 시계열 특성을 갖는 교통정보 예측 분야에서는 장단기 메모리(long short term memory, LSTM)가 널리 사용되고 있다. LSTM에 입력되는 시계열 데이터의 추세, 계절성, 주기 등이 상이하기 때문에 시계열 데이터를 기반으로 한 예측 모델에서도 데이터의 특성에 따라 하이퍼 파라미터의 적합한 값을 찾는 시행착오법이 필수적이다. 이에 적합한 하이퍼 파라미터를 찾는 방법론이 정립된다면, 정확도가 높은 모델 구성에 소요되는 시간을 줄일 수 있다. 따라서, 본 연구에서는 국내 고속도로 차량검지기 데이터와 LSTM을 기반으로 교통정보 예측 모델을 개발하였으며, LSTM의 하이퍼 파라미터별 평가지표 변화를 통해 예측 결과에 미치는 영향평가를 수행하였다. 또한, 이를 기반으로 교통분야에서 고속도로 교통정보 예측에 적합한 하이퍼 파라미터를 찾는 방법론을 제시하였다.

핵심어 : 교통정보, 예측, 장단기 메모리, 하이퍼 파라미터, 최적화

ABSTRACT

With the recent developments in big data and deep learning, a variety of traffic information is collected widely and used for traffic operations. In particular, long short-term memory (LSTM) is used in the field of traffic information prediction with time series characteristics. Since trends, seasons, and cycles differ due to the nature of time series data input for an LSTM, a trial-and-error method based on characteristics of the data is essential for prediction models based on time series data in order to find hyperparameters. If a methodology is established to find suitable hyperparameters, it is possible to reduce the time spent in constructing high-accuracy models. Therefore, in this study, a traffic information prediction model is developed based on highway vehicle detection system (VDS) data and LSTM, and an impact assessment is conducted through changes in the LSTM evaluation indicators for each hyperparameter. In addition, a methodology for finding hyperparameters suitable for predicting highway traffic information in the transportation field is presented.

Key words : Traffic information, Prediction, Long short term memory, Hyper-parameters, Optimization

I. 서론

1. 연구의 배경 및 목적

교통정보 예측은 교통시설 이용자와 운영자 모두에게 효율적인 대안 선택 및 운영을 위한 중요한 과정으로 활용된다(Yu and Kim, 2010). 최근에는 IT 기술의 발달로 교통 분야에서 많은 양의 데이터를 다양한 형태로 수집·저장할 수 있게 되었으며, 이를 교통정보 예측, 교통시설 운영, 자율주행 등에서 다양하게 활용하고 있다(Kim et al., 2016).

교통정보 예측의 기반이 되는 데이터는 많은 양이 차량 검지기(vehicle detection system, VDS)를 활용하여 수집된다. VDS를 통해 수집되는 데이터는 시계열 데이터이다. 시계열 데이터는 데이터의 장기적인 증감 형태인 추세, 계절적 요인이 시계열에 영향을 주는 특성인 계절성, 고정된 빈도가 아닌 형태로 증가나 감소하는 모습을 보이는 주기 등의 특성을 가지고 있다(Hyndman and Athanasopoulos, 2018).

이렇게 수집된 시계열 데이터를 이용하는 예측 모형은 전통적인 시계열 회귀모형, 자동회귀누적이동평균(autoregressive integrated moving average, ARIMA)모형부터 순환신경망(recurrent neural network, RNN)의 일종인 장단기메모리(long short term memory, LSTM), Conv-LSTM(convolutional long short term memory) 등이 있다. 일반적인 시계열 예측 모델은 데이터의 종류별로 상이한 추세, 계절성, 주기 등을 적합하게 사용할 수 있도록 해당 특성들을 분리·제거하는 방법을 사용하고 있다(Ryu and Kim, 2018). 하지만, LSTM 기반 시계열 예측 모델의 경우 충분한 기간으로 데이터가 입력되는 경우 추세, 계절성, 주기 등을 학습할 수 있어 예측 분야에서 널리 활용되고 있다(Muzaffar abd Afshari, 2019).

LSTM 모델은 시계열 데이터의 특징을 학습할 수 있는 장점이 있으나, 입력되는 데이터와 모형에 따라 최적의 성능을 낼 수 있도록 모델 내부의 하이퍼 파라미터(hyper-parameter)를 시행착오법(trial-error method)을 통해 조절해야 한다(Sohn and Kim, 2021). 일반적으로 LSTM의 최적화에서 수정되는 하이퍼 파라미터는 window size, hidden layer, unit, batch size, dropout rate, epoch 등이 있다. 시행착오법은 딥러닝 모델의 정확도를 높이고, 안정적인 결과를 산출하기 위해 필수적인 과정이지만, 하이퍼 파라미터를 수정할 때마다 모델링을 수행해야 하기 때문에 많은 시간이 소요된다.

따라서, 본 연구에서는 LSTM을 활용하는 교통정보 예측 모델 구성 시, 시행착오법의 효율적인 적용을 위한 방법론을 제시하고자 한다. 이를 위해 한국도로공사 공공데이터 포털(<http://data.ex.co.kr/>)에서 제공하는 고속도로 교통량·속도 데이터를 기반으로 LSTM을 활용한 고속도로 교통정보 예측 모델을 개발하였으며, 하이퍼 파라미터 및 활성화 함수의 설정 순서 및 모델 예측 정확도에 대한 영향을 판단하였다. 최종적으로 해당 모델 개발 과정 및 하이퍼 파라미터별 예측 정확도에 대한 영향 판단을 기반으로 고속도로 교통정보의 예측 모델 개발의 방법론을 제시하였다.

2. 연구의 범위 및 절차

본 연구의 공간적 범위는 VDS 데이터가 수집되는 국내 고속도로 본선부로 설정하며, 시간적 범위로는 고속도로 공공데이터 포털에서 제공하고 있는 고속도로 교통정보 중 최근 3개년에 해당하는 2020 ~ 2022년으로 설정한다.

연구의 절차는 LSTM 모델의 구조, 하이퍼 파라미터, 활성화 함수에 대한 고찰을 우선하여 수행한다. 이후, 분석 대상구간을 선정하여 고속도로 교통정보 예측 모델을 개발·평가한다. 가장 최적화된 예측 모델에

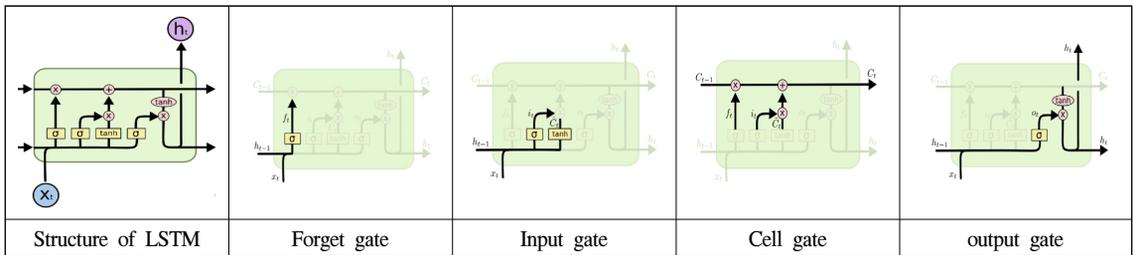
서 하이퍼 파라미터에 변동을 주어 모델의 예측 정확도에 미치는 영향을 판단한다. 최종적으로, 이론적 근거 및 모델 구성 결과, 하이퍼 파라미터의 예측값에 대한 영향을 기반으로 고속도로 교통정보 예측 모델의 개발 방법론을 제시한다.

II. 관련 이론 및 연구 고찰

1. 관련 이론 고찰

1) LSTM

LSTM은 기존 RNN이 가진 과거 학습 결과를 전달하지 못하는 기울기 소실(vanishing gradient) 문제를 극복하기 위해 특정 게이트를 통해 과거의 정보와 새로운 정보 전달 정도를 결정하는 게이트들의 조합을 사용하는 모델이다(Malhotra et al., 2016). LSTM의 구조는 <Fig. 1>과 같이, forget gate, input gate, cell gate, output gate를 중심으로 서로 정보를 주고받는 구조로 구성되어 있다. 네 개의 게이트 중 LSTM에서 가장 핵심적인 역할을 하는 게이트는 cell gate이다. Cell gate는 이전 단계의 셀 상태에서 어떤 정보를 잊고 어떤 정보를 추가할 것인지를 결정 후, 이전 시간 단계의 셀 상태와 새로운 셀 상태의 후보 값을 계산하여 최종 셀 상태를 다음 시간 단계로 전달하는 역할을 수행한다(Olah, 2015; Lee and Han, 2021). Forget gate에서는 입력된 정보 중 어떤 정보를 버릴지 결정하며, input gate에서는 어떤 값을 업데이트할 것인지, output gate에서는 출력값을 결정한다(Lee and Han, 2021).



Source : Understanding LSTM Networks, Olah(2015)

<Fig. 1> Structure of LSTM and individual gate spheres

LSTM 모델의 전체적인 연산 흐름은 다음과 같다. 먼저, forget gate에서는 현재 입력과 이전 상태의 기억을 입력으로 받아, sigmoid 함수를 통과시킨 값의 정도에 따라 정보의 보존 정도를 결정한다. 이후 input gate에서는 현재 입력의 어떤 값을 cell gate에 저장할 것인지를 sigmoid 함수 및 tanh 함수를 통해 결정한다. Cell gate에서는 forget gate에서 계산된 망각 정도를 기반으로 이전 상태를 업데이트하며, input gate에서 입력된 값을 더해 새로운 cell state를 설정한다. 마지막으로 output gate에서는 cell gate에서 업데이트된 cell state에 tanh 함수를 적용하여 현재 시점의 은닉 값을 설정한다. 이에 출력값을 계산하기 위해 현재 시점의 입력값을 기억 할지를 결정하는 sigmoid 함수값을 곱하여 최종 예측값을 결정한다(Olah, 2015).

2) 하이퍼 파라미터

하이퍼 파라미터는 머신러닝·딥러닝의 모델링 과정에서 사용자가 직접 세팅해주는 값을 뜻한다 (Brownlee, 2018). 일반적으로 LSTM 모델에서 설정하는 하이퍼 파라미터들은 입력 데이터의 길이를 결정하는 window size(Shi et al., 2022), 손실 함수를 최소화하기 위한 방법인 optimizer(Ruder, 2017), 은닉층 수를 결정하는 hidden layer(dense), 뉴런 수를 결정하는 unit, 반복 학습 횟수를 결정하는 epoch(Goodfellow et al., 2016), 학습되는 정도를 의미하는 learning rate, 일부 뉴런을 무작위로 제거하여 과적합을 방지하는 dropout rate (Srivastava et al., 2014), 모델이 한 번에 처리하는 데이터의 수인 batch size, 예측값과 실제 값 사이의 차이를 측정하는 함수인 loss function 등이 있다(Goodfellow et al., 2016).

본 연구에서는 LSTM 모델의 최적화 및 개발 방법론을 제시하기 위해 하이퍼 파라미터들을 크게 네 가지로 분류하였다. 입력 데이터부는 모델에 입력하는 데이터의 설정과 관련된 하이퍼 파라미터, 모델 구조부는 모델 은닉층과 내부 연산과 관련된 하이퍼 파라미터로 구성하였다. 또한, 모델 학습부는 학습 빈도, 입력값의 처리, 학습 정도, 오버피팅과 관련된 하이퍼 파라미터로 구성하였으며, 학습 결과부는 모델 예측값을 평가하는 지표로 구성하였다. 해당 내용은 <Table 1>과 같다.

<Table 1> Categorization of hyper-parameters in a LSTM Model

Categories	Hyper-parameters	Description
Input data section	Window size	Hyper-parameters that set the size of the data input to the model
Model structure section	Optimizer, hidden layer, unit	Hyper-parameters that set the hidden unit and calculations inside the model
Model learning section	Epoch(patience), batch size, learning rate, dropout rate	Hyper-parameters involved in the frequency of training, processing of inputs, degree of learning, overfitting within the model
Learning outcome section	Loss function	Hyper-parameter related to setting metrics to evaluate the predictive value of the model

2. 관련 연구 고찰

1) 교통정보 예측

Park et al.(2021)은 LSTM 및 CNN-LSTM 신경망을 활용한 도시부 간선도로 속도 예측 연구에서 도시부 간선도로 내의 여러 링크에 대한 정량적 속도 예측을 수행하였다. LSTM 신경망에는 도시부 간선도로의 속도, 시간, 휴일, 요일, 강수량 데이터를 입력받았으며, CNN-LSTM 신경망에서는 속도 데이터를 입력받아 유고 상황에 대응하는 속도 예측을 수행하였다. 개별 모델은 예측 시점에서 이전 120분까지의 5분 단위 데이터를 입력하여 5분 단위로 60분까지의 속도를 예측할 수 있도록 구성하였다. 모델 구성 결과 LSTM 모델의 경우 서울시 내부순환로에서 정상 교통류에 대하여 평균 7.43km/h의 오차율로, CNN-LSTM 모델의 경우 7.66km/h의 오차율로 예측하였다.

Park et al.(2020)은 교통 돌발상황을 고려한 도로 속도 예측 기법 연구를 수행하였다. LSTM 기반 예측 모델에서 예측 대상 도로의 속도 데이터뿐만 아니라 연결된 도로의 속도 데이터도 활용하였으며, 이에 이벤트 가중치를 적용해 예측 및 모델 평가를 수행하였다. 이벤트 가중치는 돌발상황 발생 시의 속도 변화 추이를 고려한 가중치를 활용하였다. LSTM 모델에서 하이퍼 파라미터는 dropout, hidden layer(dense), loss function을 설정하였다. 모델 구성 결과 돌발상황을 고려한 예측 결과가 이를 고려하지 않은 예측 결과보다 현저히 낮은 RMSE(root mean squared error, 평균 제곱근 오차) 값을 보였다.

2) LSTM 최적화

Hossain et al.(2020)은 LSTM을 기반으로 한 네트워크 공격 탐지 모델을 구성하였다. 해당 연구에서는 LSTM 기반 모델의 하이퍼 파라미터의 값을 미세하게 조정하는 방법으로 모델 성능을 향상시킬 수 있는 방법을 제시하였다. 조정된 하이퍼 파라미터들은 optimizer, learning rate, loss function, activation function, hidden layer이다. 다양한 네트워크 공격을 가장 높은 정확도로 감지하는 모델은 learning rate는 0.0001, optimizer는 RMSprop(root mean square propagation)로 설정하였을 때의 모델로 나타났다. 연구 결과, 하이퍼 파라미터값이 탐지 정확도에 어떤 영향을 미치는지 확인할 수 있었으며, 네트워크 공격 탐지에 가장 적합한 하이퍼 파라미터값 조합을 발견할 수 있었다.

Yadav et al.(2020)은 LSTM을 기반으로 인도 주식 시장의 시계열 예측을 수행하였다. 해당 연구에서는 LSTM 모델의 batch 사이의 정보 소거 여부를 통해 저장상태 및 비저장 상태의 예측력 비교와 hidden layer의 수를 조절을 통한 예측력 비교를 수행하였다. 개별 모델의 optimizer는 ADAM, batch size는 64, loss function은 RMSE로 통일하여 비교를 수행하였다. LSTM의 저장상태 및 비저장 상태의 예측력 비교는 통계적으로 유의하지 않다고 나타났으나, 안전성은 비저장 LSTM 모델이 더욱 높은 것으로 나타났다. 또한, hidden layer의 수를 조절한 결과, 1개의 hidden layer를 가진 모델이 가장 높은 정확도로 예측하는 것으로 나타났다. <Table 2>는 LSTM 최적화 관련 논문에서 제시한 최적화 모델의 목적 및 최적 하이퍼 파라미터를 표기하였다.

<Table 2> Model objectives and optimized hyper-parameters by LSTM optimization study

Authors	Purpose of model	Hyper-parameters					
		Epoch	Optimizer	Learning rate	Batch size	Loss function	Hidden layer
Hossain et al.(2020)	Network attack detection	300	RMSprop	0.0001	512	categorical_crossentropy	5
Yadav et al.(2020)	Indian stock market prediction	30	ADAM	-	64	RMSE	1

3. 연구의 차별성

기존 선행연구 논문을 살펴보면 LSTM, CNN-LSTM 등 다양한 모델을 활용한 교통정보 예측 모델이 개발되어 왔으며 개별 연구들 모두 하이퍼 파라미터의 설정은 시행착오법을 통해 수행되었다. 다만, 이를 설정하는 체계적인 방법 및 예측 결과에 대한 하이퍼 파라미터의 영향 정도를 확인하기는 어려운 것으로 보인다. 또한, LSTM의 최적화와 관련된 연구 역시 다양한 방면에서 수행되었으나 교통정보 예측을 목적으로 한 모델에서의 최적화와 관련된 연구는 부족한 것으로 보인다.

따라서, 본 연구에서는 LSTM을 기반으로 한 고속도로 교통정보 예측 모델을 개발하고, 해당 모델에서 하이퍼 파라미터의 조절을 통한 최적화를 수행하였다. 최적화된 모델의 하이퍼 파라미터 수정을 통해 개별 하이퍼 파라미터의 예측 결과에 대한 영향 정도를 확인하였으며, 이를 통해 시행착오법을 수행할 때에 하이퍼 파라미터의 중요도를 비교하고자 한다. 또한, 이를 기반으로 고속도로 교통정보 예측 모델 개발의 방법론을 제시하고자 한다.

Ⅲ. 고속도로 교통정보 예측 모델 개발

1. 자료 수집 및 대상 구간 선정

1) 고속도로 교통정보 수집 및 대상 구간 후보군 선정

고속도로 교통정보 수집의 경우, 한국도로공사 고속도로 공공데이터 포털에서 제공하고 있는 국내 고속도로 전 구간을 대상으로 5분 단위 VDS 데이터를 수집하였다. 본 연구의 목적은 고속도로 교통정보 예측 모델 개발 방법론을 제시함에 있어, 이상적이고 보편적인 교통정보를 포함하는 구간을 대상구간으로 선정하도록 하였다. 따라서, 교통정보 예측 모델을 구성하기에 적합한 데이터 셋을 선정하기 위해 세가지 기준을 기반으로 예측 모델 구성 대상 구간 후보군 선정을 수행하였다.

기준은 첫 번째로 버스전용차로, 갓길차로의 경우 VDS가 설치되지 않은 경우가 많아, 교통정보에서 오차를 일으킬 수 있으므로, 이를 수행하지 않는 구간일 것. 두 번째로 고속도로 진출입부의 경우 연속된 교통류의 흐름이 확인되지 않으므로, 진출입부를 포함하여 연속된 VDS가 3개 이상 존재할 것. 세 번째로 2023년 기준 인근 3년 데이터 중, 교통정보가 수집되지 않아 VDS 데이터상 ‘-1’로 표기되는 오류 데이터가 10만 건/년(약 3개월) 미만으로 발생하는 연속된 2개년 데이터가 존재할 것으로 설정하였다. 해당 기준을 만족하는 국내 고속도로 본선 구간은 총 7개 구간으로 <Table 3>과 같다.

<Table 3> Candidates for the target section

Section Name	Conzone ID	Installation of LCS/Bus only lane	No. of VDS	No. of Error Data		
				2020	2021	2022
Gonjiam JC → Gonjiam IC	0352CZE384	X	5	43,214	83,145	132,483
Gonjiam IC → Gonjiam JC	0352CZS383	X	5	1,437	45,311	139,836
Gyeonggi Gwangju JC → Sinyewol Bridge	0370CZS030	X	4	684	31,336	105,785
Milyang IC → Milyang JC	0552CZE085	X	4	1,893	2,750	1,343
Ulsan JC → Bumseo IC	0652CZS010	X	6	13,403	63,632	59,597
Donggyeongju IC → Nampo Port IC	0652CZS040	X	10	1,348	1,492	127,284
Gumdan Yangchon IC → Daegot IC	4001CZE130	X	4	4,939	5,400	0

2) 대상 구간 선정 및 데이터 가공

교통량이 지나치게 적은 구간은 교통류의 주행 평균속도가 100km/h를 웃돌고 거의 변화가 없어 예측 모델 구성에 적절하지 않을 것으로 판단하였다. 따라서, 대상 구간 선정은 두 가지 기준을 기반으로 수행하였다. 첫 번째로 하루 중 교통류 평균 주행속도가 1km/h 이상 80km/h 이하인 시간대가 2시간 이상일 것으로 설정하였다. 두 번째 기준도 동일한 이유로, 하루 중 5분 기준 교통량이 100대 이상인 시간대가 2시간 이상 있을 것으로 설정하였다.

해당 기준을 기반으로 7개 후보 구간의 교통량 및 속도 데이터를 검토하였다. 곤지암 JC → 곤지암 IC, 곤지암 IC → 곤지암 JC 구간의 경우 평균속도가 1km/h 이상 80km/h 이하에 속하는 데이터가 약 3%에 해당하여 예측 모델의 구성에 부적합하다고 판단하였으며, 동일한 이유로 밀양 IC → 밀양 JC 구간 역시 평균속도가 1km/h 이상 80km/h 이하에 속하는 데이터는 약 1.3%에 해당하여 제외하였다. 이후, 동경주 IC → 남포항 IC, 검단양촌 IC → 대곶 IC 구간은 5분 기준 교통량이 100대 이상에 해당하는 데이터가 0.93%, 0.20%로 교

통량이 현저히 낮게 나타났기 때문에 대상 구간에서 제외하였다. 최종적으로 선정된 경기광주 JC → 신월천교 구간의 교통정보 수집 상태는 <Table 4>와 같다.

<Table 4> Traffic information collection status of Gyeonggi Gwangju JC → Sinyewol Bridge

Year	Number of error data	Length (m)	Average speed (km/h)	Average traffic volume (veh/5min)	Percentage of average speed between 1km/h and 80km/h	Percentage of traffic volume over 100veh/5min
2020	684	3,200	93.473	117	24,150/316,800(7.623%)	185,022/316,800(58.403%)
2021	31,336		84.402	109	46,020/420,480(10.945%)	222,813/420,480(52.990%)
2022	105,785		55.761	71	36,941/420,480(8.785%)	142,916/420,480(33.989%)

선정된 경기광주 JC → 신월천교 구간의 교통량 및 속도 데이터는 모델 입력 및 학습에 용이하도록 5분 간격으로 3년간의 데이터를 수집하였으며, VDS별로 분류하여 데이터를 병합하였다. 데이터를 검토하였을 때, 2022년의 데이터는 오류 데이터가 10만 건(약 3개월) 이상이며, 교통량 및 평균 속도의 양상이 2020년~2021년 데이터와 상이하다. 따라서, 이를 제외하고 2020년~2021년의 데이터를 취합하여 대상구간으로 최종 선정하였다.

본 절에서는 고속도로 교통정보 예측 모델 개발 방법론을 제시하기 위해 모델 구성 과정을 예시로 보이고 검증하는 것을 목적으로 하므로, 선정된 경기광주 JC → 신월천교의 진출입로 VDS를 제외하고 교통량 및 속도 데이터의 오류가 가장 적은 VDS(0370VDS01650)을 대상구간으로 선정하여 모델을 구성하였다.

2. 모델 구성 및 시행착오법 적용

본 절에서는 모델 구성 대상구간의 시계열 데이터를 기반으로 구성된 LSTM 모델 최적화를 위해 다양한 하이퍼 파라미터들을 조정하고, 최적값을 산정하는 과정을 예시로 보인다. 하이퍼 파라미터 중, epoch(patience)는 개별 모델의 loss function이 감소하지 않는 상태가 지속되는 횟수, MSE(mean squared error)로 설정된 loss function은 모델의 평가지표인 점을 감안하여 시행착오법을 수행하는 대상에서 제외하였다. 또한, 본 연구에서 제시된 시행착오법 과정 및 결과는 분석의 용이성을 위해 교통량 데이터를 대상으로 하였다.

1) 모델 입력 데이터, 예측 기간 및 모델 기본 설정

예측 모델의 구성에 앞서, 모델에 입력하여 학습시킬 데이터의 기간 및 모델을 통해 예측할 기간을 먼저 설정한다. 본 연구에서는 학습시킬 데이터는 수집된 2020년~2021년 총 2개년 데이터를 입력하여 학습을 수행하며, 예측 시간은 1시간을 설정하였다. 또한, 시행착오법을 수행하기 위해 모델의 초깃값을 설정하도록 한다. 따라서, 시행착오법을 수행하기 이전의 기본 설정은 다음 <Table 5>와 같이 임의로 설정하였으며 이후 하이퍼 파라미터를 수정하며 모델의 성능을 평가하도록 한다.

<Table 5> Base setting of the prediction model

Categories	Hyper-parameters	Setting values
Input data section	Window size	8
Model structure section	Hidden layer	1
	Unit	16

Categories	Hyper-parameters	Setting values
	Optimizer	Adam
Model learning section	Epoch(patience)	5
	Batch size	16
	Learning rate	0.001
	dropout Rate	0.2
Learning outcome section	Loss function	MSE
Evaluation value(volume)	-	21.524
Evaluation value(speed)		20.135

2) 입력 데이터부 하이퍼 파라미터 설정

입력 데이터부는 구성된 모델에 입력되는 데이터들의 크기 및 시간 단위를 조절할 수 있는 하이퍼 파라미터로 구성하였다. LSTM에서 처리되는 데이터들의 단위를 해당하는 window size는 LSTM에서 예측을 수행할 때에 시간적인 연속성을 가지는 시퀀스 기준이다. 예를 들어, window size가 10인 경우, 10개 데이터를 하나의 묶음으로 예측을 수행하며, 예측값 역시 10개의 데이터 묶음에 상응하는 데이터로 출력된다. 따라서, 예측 기간을 넘어서는 window size로 학습을 수행할 수 없기 때문에, 이를 최우선으로 설정해 최적값을 찾는 것이 모델의 구성에 있어 효율적인 방법으로 판단하였다.

본 연구에서는 5분 단위 데이터를 기반으로 LSTM 모델을 학습시켰으며, 1시간을 예측하도록 설정하였기 때문에 window size를 12 이하로 설정하도록 하였다. Window size는 1(5분), 2(10분), 3(15분), 4(20분), 5(25분), 6(30분), 8(40분), 10(50분)을 순차적으로 설정하여 시행착오법을 통해 성능을 비교하였으며, window size 별 MSE는 다음 <Table 6>과 같다. 평가를 수행한 결과, window size 4(20분)를 기준으로 설정하였을 때 예측값이 실제 값과 가장 유사한 것으로 나타났다.

<Table 6> Window size selection by evaluation score

Window sizes	Train scores	Test scores
1	34.375	15.796
2	31.696	12.592
3	32.264	12.078
4	32.812	11.502
5	31.449	13.384
6	30.982	14.499
8	32.885	17.668
10	31.658	20.434

3) 모델 구조부 하이퍼 파라미터 설정

모델 구조부는 입력된 데이터들이 처리되는 과정 및 방법에 관여하는 하이퍼 파라미터들로 구성하였다. 모델 구조부에는 입력된 데이터를 처리하는 hidden layer와 hidden layer 내부에서 개별 입력 데이터들의 연산을 담당하는 unit, 학습의 방향 및 편향 등을 설정하는 optimizer가 해당한다. 해당 하이퍼 파라미터들은 은닉층 내부 변수들을 결정하는 과정에 핵심적인 파라미터이다. 따라서, 모델 학습부 이전에 설정되는 것이 합당하다고 판단하였다.

Optimizer는 loss function을 감소시키는 방향으로 학습이 진행될 수 있도록 하는 경사하강법(gradient descent, GD)을 기본으로 한다. 이후, 학습시간을 감소시키며 학습률을 높이는 방향으로 발전하여 확률적 경사하강법(stochastic gradient descent, SGD), RMSprop, Adam(adaptive moment estimation), Adadelata, Adamax, Nadam, Ftrl 등으로 다양화되었다. 개별 optimizer의 개략적인 설명은 다음 <Table 7>과 같다.

모델 구조부에서는 학습의 수행 방향을 설정하는 optimizer를 가장 우선적으로 설정한다. 이후, 학습을 수행하는 hidden layer의 수를 먼저 설정하고, 개별 layer의 unit 수를 설정한다. 다만, 개별 hidden layer에서 unit의 수를 설정할 수 있기 때문에, 직관성 있는 하이퍼 파라미터의 영향도를 판단하기 위해 hidden layer 별로 unit의 수는 통일하여 성능을 비교하였다.

<Table 7> Descriptions of Individual optimizers

Optimizer	Description
SGD	Improves on Gradient Descent by selecting random values
Adagrad	Optimization techniques based on learning rates specific to certain parameters The learning rate is adjusted based on how often that parameter is updated while learning
RMSprop	Maintaining a dynamic moving average of the squares of the gradient Standardize the gradient by dividing it by the root of the average
Adam	Based on dynamic measurements of first and second order moments Computationally efficient, has low memory usage, and is immune to diagonal rescaling of the gradient
Adadelata	Improves Stochastic Gradient Descent method by an adaptive learning rate per dimension Advantage of being able to learn stably even if the gradient is updated very often
Adamax	As a type of Adam technique, it is based on the infinity norm Perform better than Adam when utilizing models that use vectorization such as embedding
Nadam	Optimization technique applied to Adam using Nesterov momentum instead of regular momentum
Ftrl	Improve Adam to account for tilt in future positions Address the shortcomings of other optimization in that different dimensions have different learning rates

Source : <https://towardsdatascience.com/understanding-rmsprop-faster-neural-network-learning-62e116fcf29a>

본 연구에서는 Keras 라이브러리에서 제공하는 <Table 7>의 optimizer들을 대상으로 MSE를 비교해 최적 optimizer를 선정하였다. Optimizer 별 모델 평가를 수행한 결과, adam을 활용한 모델의 예측값이 실제 값과 가장 유사한 것으로 나타났다. 이후, hidden layer는 최소 설정값인 1부터 5까지 점차 증가시켰으며 최적의 hidden layer 수로 고정한 후, unit의 수를 조정하였다. Unit은 최소 1부터 128까지 실험하였다. 모델 구조부의 최적화 결과 <Table 8>과 같이 optimizer는 Adam, hidden layer는 4, unit은 8일 때에 MSE는 11.252로 가장 낮은 값을 보였다.

<Table 8> The process of optimizing model structure section with hyper parameter tuning

Optimizer			Hidden layer(optimizer : Adam)			Unit(hidden layer : 4)		
Type	Train score	Test score	Number	Train score	Test score	Number	Train score	Test score
SGD	33.856	12.350	1	31.994	11.437	1	31.837	11.465
Adagrad	34.358	12.540	2	31.930	11.551	2	31.769	11.766
RMSprop	34.004	12.472	3	32.438	13.207	3	31.039	12.259

Optimizer			Hidden layer(optimizer : Adam)			Unit(hidden layer : 4)		
Adam	30.614	11.827	4	31.929	10.719	4	32.106	11.466
Adadelta	34.470	12.661	5	31.902	12.568	8	30.319	11.252
Adamax	33.276	12.277				16	30.565	12.588
Nadam	32.417	13.028				32	32.469	13.505
Ftrl	34.304	12.408				64	30.565	12.588
-	-	-				128	31.141	11.768
-	-	-				-	-	-

4) 모델 학습부 하이퍼 파라미터 설정

모델 학습부는 전반적인 학습에 관여하는 하이퍼 파라미터로 구성하였다. Learning rate는 학습 정도, dropout rate는 오버피팅의 방지를 위한 뉴런 비활성화 비율, batch size는 학습의 다중 처리하는 역할을 수행한다. 모델의 학습은 모델의 기본적인 구조 및 학습 방향의 설정 이후에 수행되어야 하므로 마지막 순서로 설정하였다.

모델 학습부에서는 모델 학습의 내부에서 가중치 업데이트의 크기를 결정하는데 가장 큰 영향을 주는 파라미터인 learning rate를 가장 먼저 설정하도록 한다. 이후, 모델 내부의 일부 뉴런을 비활성화하는 dropout rate, 마지막으로 batch size를 설정하도록 하였다.

Learning rate는 0.001부터 시작하여 점진적으로 0.1까지 증가시킨 후, dropout rate는 0.1부터 시작하여 점진적으로 0.8까지, batch size는 최솟값인 1부터 64까지 증가시켜 실험하였다. 모델 학습부의 최적화 결과 <Table 9>와 같이 learning rate는 0.01, dropout rate는 0.2, batch size는 32일 때에 MSE는 10.872로 가장 낮은 값을 보였다.

<Table 9> The process of optimizing model learning section with hyper parameter tuning

Learning rate			Dropout rate			Batch size		
Ratio	Train score	Test score	Ratio	Train score	Test score	Size	Train score	Test score
0.001	32.302	13.073	0.1	22.114	11.578	1	28.813	11.926
0.01	31.389	10.525	0.2	29.882	10.810	2	31.617	11.845
0.1	27.332	17.332	0.4	56.488	12.974	4	29.622	10.982
			0.6	79.772	22.623	8	30.049	12.123
			0.8	109.021	33.019	16	30.026	11.720
			-	-	-	32	30.766	10.872
			-	-	-	64	29.251	12.767
			-	-	-	-	-	-

3. 예측 모델 평가

본 연구에서는 경기광주 JC → 신월천교 구간의 2020년~2021년 교통량 및 속도 데이터를 대상으로 교통 정보 예측 모델을 개발하였다. LSTM 모델의 내부 하이퍼 파라미터는 시행착오법을 통해 입력 데이터부, 모델 구조부, 모델 학습부 순서로 loss function을 기반으로 최적값을 선정, 최종 구성된 모델은 <Table 10>과 같다. 임의로 구성한 모델의 test score는 교통량이 21.524, 평균 속도가 20.135인 것에 비해, 최적화 이후의 test score는 교통량이 10.955, 속도가 15.952로 MSE가 큰 폭으로 감소하였다.

또한, loss function 외의 평가지표를 통해 개선 전·후 모델의 평가를 수행하였다. 평가지표는 실제 값과

예측값의 차를 계산한 평균절대오차(mean absolute error, MAE), 평가 오류를 퍼센트로 반환한 평균절대비오차(mean absolute percentage error, MAPE), MAPE가 0 근처에서 발산하는 문제를 해결한 대칭평균절대비오차(symmetric mean absolute percentage error, SMAPE)를 사용하였으며(Moon et al., 2017), <Table 11>과 같다. 개발된 교통정보 예측 모델은 대상 구간의 5분 기준 평균 교통량이 117대, 평균 속도가 88인 것을 고려하였을 때, 교통량, 속도 예측 모델의 경우 MAE가 7.616, 15.324, MAPE가 16.582, 16.859 SMPAE가 8.994, 9.266으로 평가되었으므로 오차율이 약 15% 내외인 모델이 구성되었다.

<Table 10> Setting of the optimized prediction model

Categories	Hyper-parameters	Setting values
Input data section	Window size	4
Model structure section	Optimizer	Adam
	Hidden layer	4
	Unit	8
Model learning section	Epoch patience	5
	Learning rate	0.01
	Dropout Rate	0.2
	Batch size	32
Learning outcome section	Loss function	MSE
Evaluation value(volume)	-	10.955
Evaluation value(speed)	-	15.952

<Table 11> Model evaluation results before and after optimization

Categories	Before optimization (volume)	After optimization (volume)	Before optimization (speed)	After optimization (speed)
MAE	17.581	7.616	19.965	15.324
MAPE	33.482	16.582	21.462	16.859
SMAPE	21.920	8.994	12.038	9.266

IV. 하이퍼 파라미터 영향 평가 및 개발 방법론 제시

1. 하이퍼 파라미터 영향 평가

본 절에서는 최적화된 교통량 예측 모델의 하이퍼 파라미터를 일부 수정한 후, 평가지표를 분석하여 하이퍼 파라미터별로 모델의 예측 정확도에 미치는 영향을 확인하고자 한다. 하이퍼 파라미터는 모델의 최적화 과정과 동일하게 분류별로 수행하도록 하며, 최적값일 때를 중심으로 4사분위수까지 범위를 설정해 개별 하이퍼 파라미터의 변동에 따른 평가지표 변화를 비교하도록 한다.

다만, 하이퍼 파라미터들의 단위가 상이할 경우 min-max scaler를 통해 단위를 통일해 비교하였으며, optimizer의 경우 loss function에 기반해 학습 방향을 설정하는 하이퍼 파라미터이기에 영향 평가에서 제외하였다.

1) 입력 데이터부 영향 평가

입력 데이터부의 하이퍼 파라미터에 해당하는 window size는 최적값 4를 기준으로 감소·증가시켜 평가지표의 차이를 확인하였다. 확인 결과, <Table 12>와 같이 최적값을 중심으로 멀어질수록 평가지표는 증가하는 추세를 확인할 수 있다.

<Table 12> Input data section impact assessment

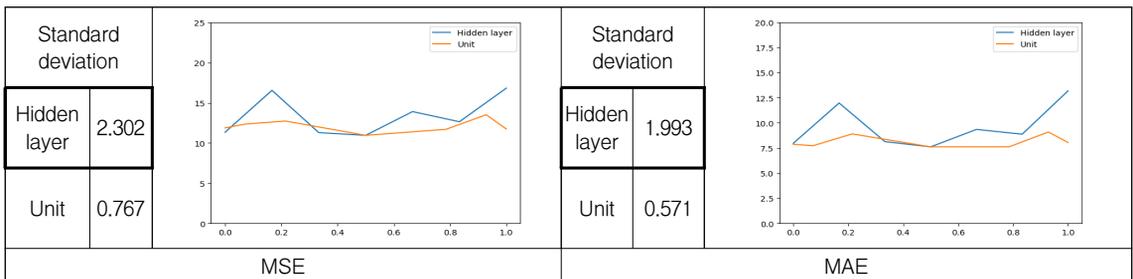
Window size	MSE	MAE	MAPE	SMAPE
1	17.545	12.591	25.823	15.240
2	12.510	9.485	18.500	10.513
3	13.103	9.515	18.698	11.016
4	10.955	7.616	16.582	8.994
5	12.773	9.053	18.397	10.733
6	16.533	11.875	23.103	14.550
7	13.539	11.264	25.636	13.253

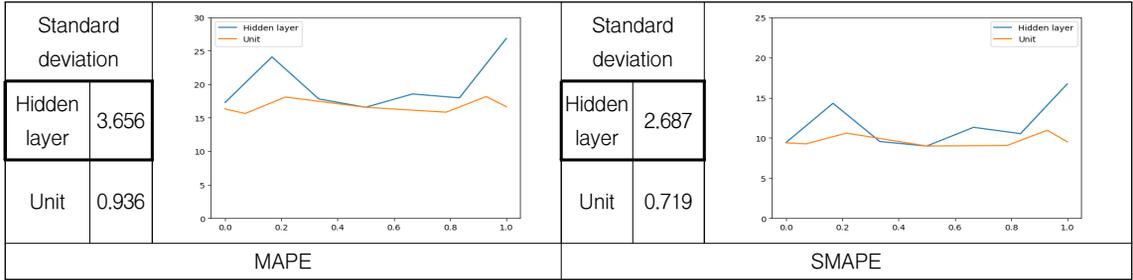
2) 모델 구조부 영향 평가

모델 구조부의 하이퍼 파라미터에 해당하는 hidden layer, unit은 개별 최적값 4, 8을 기준으로 감소·증가시켜 평가지표의 차이를 확인하였다. 확인 결과, <Table 13>과 같이 최적값을 기준으로 하였을 때, 평가지표가 증가함을 확인할 수 있다. 또한, hidden layer와 unit의 변화에 따른 예측 결과에 따른 영향 평가를 수행한 결과, <Fig. 2>와 같이 hidden layer의 변화가 unit의 변화보다 평가지표에 큰 폭으로 영향을 주는 것으로 나타났다.

<Table 13> Model structure section impact assessment

Hidden layer					Unit				
Number	MSE	MAE	MAPE	SMAPE	Number	MSE	MAE	MAPE	SMAPE
1	11.342	7.976	17.294	9.449	1	11.881	7.874	16.327	9.390
2	16.548	11.981	24.105	14.312	2	12.353	7.727	15.649	9.266
3	11.290	8.136	17.817	9.552	4	12.734	8.904	18.106	10.602
4	10.955	7.616	16.582	8.994	8	10.955	7.616	16.582	8.994
5	13.907	9.352	18.566	11.325	12	11.701	7.621	15.845	9.068
6	12.647	8.876	17.988	10.535	14	13.518	9.096	18.183	10.959
7	16.822	13.184	26.851	16.737	15	11.744	8.053	16.673	9.508





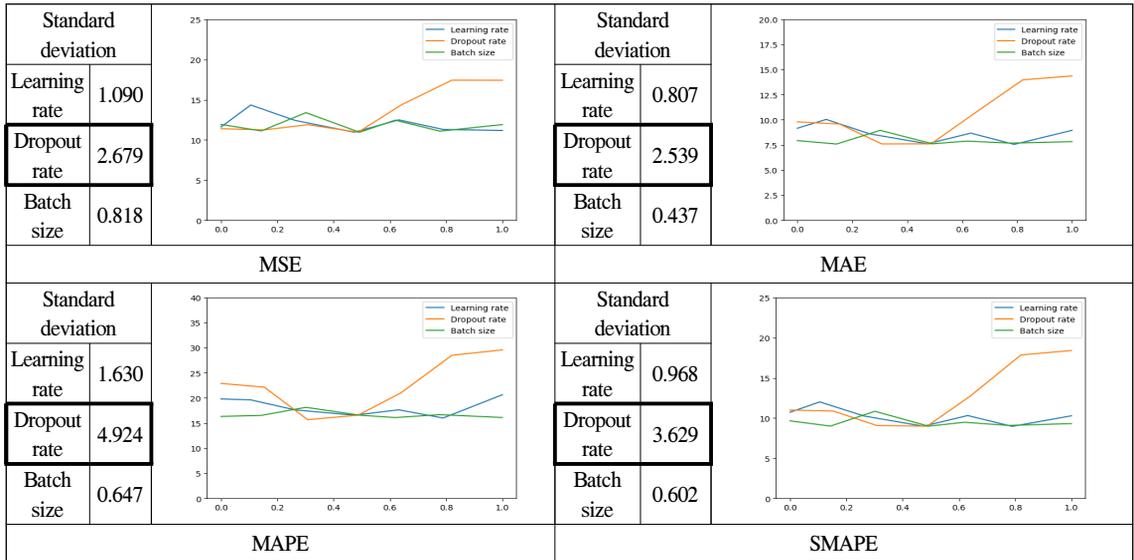
<Fig. 2> Impact assessment changes by model structure section hyper-parameters

3) 모델 학습부 영향 평가

모델 구조부의 하이퍼 파라미터에 해당하는 learning rate, dropout rate, batch size는 개별 최적값 0.01, 0.2, 32를 기준으로 감소·증가시켜 평가지표의 차이를 확인하였다. 확인 결과, <Table 14>와 같이 최적값을 중심으로 하였을 때, 평가지표가 증가하였다. 또한, learning rate, dropout rate, batch size의 변화에 따른 예측 결과에 따른 영향 평가를 수행한 결과, <Fig. 3>과 같이 dropout rate의 변화가 평가지표에 가장 큰 폭으로 영향을 주는 것으로 나타났으며, 다음으로 learning rate, batch size 순으로 영향을 미치는 것으로 확인되었다.

<Table 14> Model learning section impact assessment

Learning rate					Dropout rate				
Ratio	MSE	MAE	MAPE	SMAPE	Ratio	MSE	MAE	MAPE	SMAPE
0.001	11.623	9.164	19.812	10.733	0.01	11.395	9.799	22.901	11.000
0.003	14.349	10.042	19.631	12.016	0.07	11.251	9.583	22.154	10.872
0.006	12.426	8.607	17.633	10.259	0.13	11.895	7.601	15.685	9.082
0.01	10.955	7.616	16.582	8.994	0.2	10.955	7.616	16.582	8.994
0.013	12.497	8.679	17.648	10.317	0.26	14.369	10.573	21.082	12.753
0.016	11.305	7.556	16.017	8.972	0.33	17.438	13.978	28.506	17.851
0.02	11.179	8.955	20.647	10.285	0.4	17.429	14.370	29.573	18.409
Batch size									
Size	MSE	MAE	MAPE	SMAPE					
1	11.926	7.935	19.280	9.658					
10	11.120	7.594	16.536	9.006					
20	13.396	8.955	18.148	10.849					
32	10.955	7.616	16.582	8.994					
40	12.461	7.887	16.111	9.483					
50	11.089	7.680	16.682	9.090					
64	11.911	7.832	16.123	9.318					



<Fig. 3> Impact assessment changes by model learning section hyper-parameters

2. 모델 개발 방법론 제시

본 절에서는 앞서 수행한 고속도로 교통정보 예측 모델 예시 및 이론적 근거, 하이퍼 파라미터들의 영향도 평가 등을 기반으로 고속도로 교통정보 예측 모델 개발 방법론을 제시하고자 한다. 본 연구에서 제시하는 개발 방법론은 예측 모델의 구성을 위한 방법론으로, 대상구간 수집 및 선정은 제외한다. 모델 개발 방법론은 기초 모델 구조부 설정 이후에 입력 데이터부, 모델 구조부, 모델 학습부 수정 순서로 구성되었다.

가장 먼저 모델의 목적성에 부합하는 기초 모델 구조부를 설정하며, 해당 단계에서 epoch(patience), loss function을 설정한다. 이후 모델 구조부, 모델 학습부의 하이퍼 파라미터를 수정하는 순서로 모델 개발 방법론을 구성하였다.

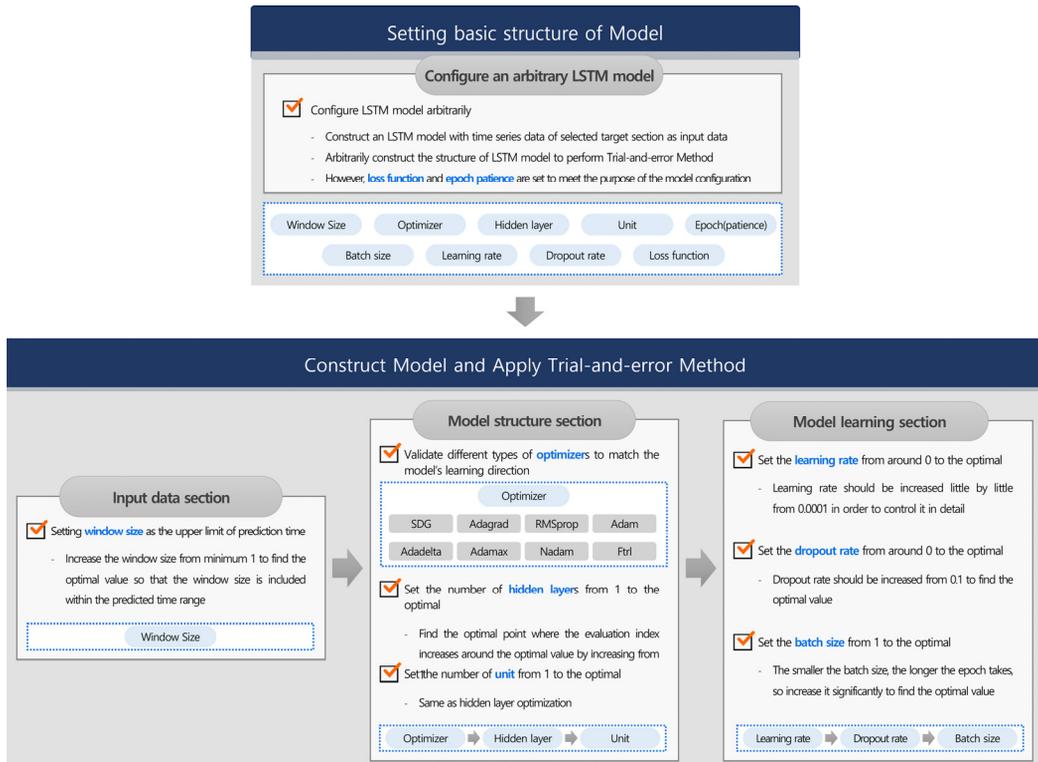
입력 데이터부의 window size의 경우, 모델의 목적 및 예측 기간에 따라 변화할 수 있어 가장 먼저 시행착오법을 수행할 수 있도록 하였다. 본 연구에서 개발한 모델의 경우, 1시간 예측을 목적으로 하므로 window size의 상한값은 12로 설정된다. 이를 기반으로 최솟값인 1에서 12까지의 조정을 통해 최적값을 설정하였다.

이후 모델 구조부에서는 optimizer, hidden layer, unit 중 optimizer를 가장 먼저 조절할 수 있도록 하였다. Optimizer는 모델의 학습 방향을 설정하는 하이퍼 파라미터이며, 향후 hidden layer와 unit 내부의 연산 방향을 수행하기 때문에 먼저 조절할 수 있도록 설정하였다. 이후, LSTM 모델의 구조상 hidden layer의 내부에 unit이 존재하도록 구성되며, 하이퍼 파라미터의 영향 평가에서도 unit과 비교하여 표준편차의 변위에 3배 이상 영향을 미치므로 hidden layer를 먼저 조정하도록 한다. 마지막으로 unit의 수를 조절하여 모델 구조부의 최적값을 찾을 수 있도록 한다. Hidden layer와 unit의 경우, 최솟값을 1부터 시작하여 점차 증가시키며 시행착오법을 적용한다. 이는 최적값을 찾기 위함이며, 최적값에서 더 증가시켜 모델링을 수행하는 경우 평가지표는 최적값을 중심으로 점차 증가하는 양상을 보인다.

모델 학습부에서는 가장 먼저 learning rate를 조절할 수 있도록 한다. 이는, 모델의 가중치 업데이트의 크기를 결정하는데 가장 큰 영향을 주는 파라미터이기 때문이다. Learning rate도 유사하게, 최솟값인 0에 가까운 0.0001부터 점차 증가시키며 시행착오법을 적용하며, 평가지표는 최적값을 중심으로 가장 낮고 이후 증가

하는 양상을 보인다. 이후에 dropout rate를 조절할 수 있도록 하였다. Dropout rate는 hidden layer 내부의 unit을 무작위로 비활성화시키는 비율이기 때문에 영향 평가에서 압도적으로 높은 변위를 보였다고 판단하였다. 따라서, learning rate의 조절 이후에 이를 조절할 수 있도록 설정하였다. Batch size는 모델 내부에서 학습 다중 처리의 비율에 해당하는 하이퍼 파라미터이기 때문에 시행착오법을 수행하여도 평가지표에 큰 영향을 주지 않는 것으로 나타났다. 다만, batch size의 크기에 따라 모델의 epoch 소요 시간이 크게 차이가 났으므로, 이를 고려하여 큰 폭으로 조절하며 최적값을 찾도록 한다.

최종적으로 제시하는 고속도로 교통정보 예측 모델 개발 방법론은 <Fig. 4>와 같다.



<Fig. 4> Methodology for developing a predictive model for highway traffic information using LSTM

V. 결론 및 향후 연구 과제

최근 빅데이터 및 딥러닝의 발전으로 다양한 교통정보가 수집·활용되고 있으며, 국내에서는 교통 운영 측면에서 널리 활용되고 있다. 교통정보 운영 및 예측 분야에서는 LSTM이 널리 활용되고 있으며, LSTM의 최적화에 관련된 연구도 수행되고 있다. 다만, 교통 분야에서 LSTM을 활용한 교통정보 예측 모델의 최적화, 하이퍼 파라미터의 영향 평가 등은 부족한 실정이다. 따라서, 본 연구에서는 LSTM을 기반으로 한 고속도로 교통정보 예측 모델을 개발 및 평가하였으며, 이론적 근거 및 하이퍼 파라미터의 예측 결과에 대한 영향 정도 확인을 통해 고속도로 교통정보 예측 모델 개발의 방법론을 제시하였다.

본 연구에서 고속도로 교통정보 예측 모델의 구성은 다음과 같은 순서로 수행되었다. 먼저, 고속도로 교통

정보 수집의 경우, 국내 도로공사에서 제공하는 고속도로 본선 구간의 5분 단위 VDS 데이터를 활용하였다. 대상구간 선정에서는 교통정보 예측 모델을 구성하기에 적합한 데이터 셋을 선정하기 위해 세가지 기준을 기반으로 고속도로 교통정보 예측 모델 구성의 대상구간 후보군을 선정하였다.

선정된 모델 구성 대상구간 후보군 중, 첨두시간 및 비첨두시간과 같은 교통류의 다양한 흐름을 반영할 수 있도록 대상구간을 선정하였다. 선정된 대상구간은 경기광주 JC → 신월천교 구간이었으며 4개의 VDS 구간으로 구성되었다. 4개의 VDS 구간 중, 진출입부에 해당하는 처음과 마지막 VDS를 제외하고 오류 데이터가 비교적 적은 2번째 VDS 구간인 0370VDS01650의 2020년~2021년 시점의 데이터로 교통정보 예측 모델을 개발할 수 있도록 설정하였다.

모델 구성 및 시행착오법 적용에서는, 우선적으로 최적화를 적용하기 위해 임의로 LSTM 모델을 구성하였으며, 이를 시행착오법을 기반으로 하이퍼 파라미터를 수정·개선하여 최적의 모델을 구성하였다. 하이퍼 파라미터의 수정은 <Table 1>에서 제시한 하이퍼 파라미터 분류에 따라 입력 데이터부, 모델 구조부, 모델 학습부, 학습 결과부의 순서로 수정하였다. 가장 먼저 입력 데이터부의 window size를, 다음으로 모델 구조부의 optimizer, hidden layer, unit을, 마지막으로 모델 학습부의 learning rate, dropout rate, batch size를 조정하였다. 마지막으로 예측 모델의 최적화 이후에 모델을 MSE, MAE, MAPE, SMAPE를 기반으로 평가하였다.

경기광주 JC → 신월천교 구간의 2020년~2021년 교통량 데이터를 대상으로 교통량 예측 모델을 개발·평가한 결과는 다음과 같다. 경기광주 JC → 신월천교 구간의 5분 기준 평균 교통량이 117대, 평균 속도가 88인 것을 고려하였을 때, 교통량, 속도 예측 모델의 경우 MAE가 7.616, 15.324, MAPE가 16.582, 16.859 SMPAE가 8.994, 9.266으로 평가되었으므로 오차율이 약 15% 내외인 모델이 구성되었다.

또한, 최적화된 모델의 하이퍼 파라미터 수정을 통해 하이퍼 파라미터의 분류 별로 개별 하이퍼 파라미터들의 영향을 비교하였다. 전체적인 비교 결과 본 연구에서 수행한 하이퍼 파라미터 수정의 우선순위와 하이퍼 파라미터의 영향은 유사하게 나타났다. 다만, 모델 학습부의 dropout rate와 learning rate의 영향은 하이퍼 파라미터 수정의 우선순위와 차이가 있었다. 이는, dropout rate와 learning rate의 영향 평가 범위에 있어 차이가 있었기 때문이라고 판단된다.

최종적으로, 고속도로 예측 모델의 개발·평가 결과 및 하이퍼 파라미터의 영향 평가 결과를 기반으로 고속도로 교통정보 예측 모델 개발 방법론을 제시하였다. 해당 방법론은 모델 개발을 목적으로 하기에, 고속도로 교통정보 수집 및 가공을 제외하고 제시하였다. 가장 먼저 모델의 최적화를 수행하기 위해 임의의 모델을 구성한다. 구성 시에 loss function 및 epoch(patience)는 모델의 목적에 부합하도록 가장 먼저 설정하도록 한다. 이후 시행착오법을 활용한 하이퍼 파라미터의 조정은 입력 데이터부의 window size, 모델 구조부의 optimizer, hidden layer, unit, 모델 학습부의 learning rate, dropout rate, batch size를 순서로 하도록 한다. 마지막으로 시계열 예측 모델에서 높은 빈도로 선택되는 평가지표인 MSE, MAE, MAPE, SMAPE를 기반으로 평가할 수 있도록 한다.

본 연구는 이론적 근거 및 고속도로 교통량 예측 모델 개발·평가 결과를 기반으로 교통정보 예측 모델을 개발하였다. 또한, 교통정보 예측모델의 하이퍼 파라미터 영향 평가를 통해 하이퍼 파라미터 별 중요도를 판단하였으며 이를 기반으로 교통정보 예측모델의 개발 방법론을 제시하였다. 다만, 고속도로의 보편적이고 이상적인 시계열 데이터를 선정하여 경기광주 JC → 신월천교 구간만을 대상으로 모델을 개발하고 검증하였다. 이는 일반적인 고속도로의 교통정보의 예측을 위해서이나, 고속도로의 물리적 구조와 같은 환경적 요소들을 포함하지 않았다는 한계를 가지고 있다. 또한, 딥러닝 모델의 특성 상, 하이퍼 파라미터의 영향 정도는 데이터의 종류에 따라 달라질 수 있어 향후 연구에서는 이를 고려하여 교통류 흐름의 다양한 요소들을 고려함이 합당하다. 따라서, 향후 다양한 환경적 요소들을 포함하여 대상 구간을 확대한 연구를 수행할 경우, 실

시간 교통정보 제공을 위한 교통정보 예측 모델 개발에 용이하게 활용될 수 있을 것이라 사료된다.

ACKNOWLEDGMENTS

본 논문은 국토교통부 자율주행 기술개발 혁신사업 ‘주행 및 충돌상황 대응 안전성 평가기술개발 (RS-2021-KA160637)’ 과제 지원에 의해 수행되었습니다.

REFERENCES

- Brownlee, J.(2018), “Better Deep Learning: Train Faster, Reduce Overfitting, and Make better Predictions”, *Machine Learning Mastery*, p.540.
- Bushaev, V.(2018), *Understanding RMSprop-faster neural network learning*, June, 13, 2023. <https://towardsdatascience.com/understanding-rmsprop-faster-neural-network-learning-62e116fcf29a>, 2023.07.19.
- Goodfellow, I., Bengio, Y. and Courville, A.(2016), *Deep learning*, MIT Press.
- Hossain, M. D., Ochiai, H., Fall, D. and Kadobayashi, Y.(2020), “LSTM-based network attack detection: Performance comparison by hyper-parameter values tuning”, *2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, pp.62-69.
- Hyndman, R. K. and Athanasopoulos, G.(2018), *Forecasting: Principles and Practice* (3rd ed.), OTexts: Melbourne, Australia. OTexts.com/fpp3, 2023.07.19.
- Kim, H. J., Park, S. H. and Jang, K. T.(2016), “Short-term Traffic States Prediction Using k-Nearest Neighbor Algorithm: Focused on Urban Expressway in Seoul”, *Journal of Korean Society of Transportation*, vol. 34, no. 2, pp.158-167.
- Lee, J. and Han, J.(2021), “Layer-wise Relevance Propagation (LRP) Based Technical and Macroeconomic Indicator Impact Analysis for an Explainable Deep Learning Model to Predict an Increase and Decrease in KOSPI”, *Journal of Korean Institute of Information Scientists and Engineers*, vol. 48, no. 12, pp.1289-1297.
- Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P. and Shroff, G.(2016), *LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection*, arXiv preprint arXiv:1607.00148.
- Moon, J., Park, J., Han, S. and Hwang, E.(2017), “Power Consumption Forecasting Scheme for Educational Institutions Based on Analysis of Similar Time Series Data”, *Journal of Korean Institute of Information Scientists and Engineers*, vol. 44, no. 9, pp.954-965.
- Muzaffar, S. and Afshari, A.(2019), “Short-Term Load Forecasts Using LSTM Networks”, *Energy Procedia*, vol. 158, pp.2922-2927.
- Olah, C.(2015), *Understanding LSTM Networks(2015)*, Oct, 10, 2022, <https://colah.github.io/posts/2015-08-Understanding-LSTMs>, 2023.07.19.
- Park, B., Bae, S. and Jung, B.(2021), “Speed Prediction of Urban Freeway Using LSTM and

- CNN-LSTM Neural Network”, *The Journal of the Korea Institute of Intelligent Transport System*, vol. 20, no. 1, pp.86-99.
- Park, S., Choi, D., Bok, K. and Yoo, J.(2020), “Road Speed Prediction Scheme Considering Traffic Incidents”, *The Journal of the Korea Contents Association*, vol. 20, no. 4, pp.25-37.
- Ruder, S.(2016), *An overview of gradient descent optimization algorithms*, arXiv preprint arXiv:1609.04747.
- Ryu, K. D. and Kim, W. J.(2018), “Comparative Analysis of Time Series Method for Forecasting the Call Arrival of Call Center”, *The Journal of Korean Institute of Information Technology*, vol. 16, no. 8, pp.83-96.
- Shi, J., Jain, M. and Narasimhan, G.(2022), *Time series forecasting (tsf) using various deep learning models*, arXiv preprint arXiv:2204.11115.
- Sohn, E. S. and Kim, J. K.(2021), “FlappyBird Competition System: A Competition-Based Assessment System for AI Course”, *Journal of Korea Multimedia Society*, vol. 24, no. 4, pp.593-600.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.(2014), “Dropout: A simple way to prevent neural networks from overfitting”, *The Journal of Machine Learning Research*, vol. 15, no. 1, pp.1929-1958.
- Yadav, A., Jha, C. K. and Sharan, A.(2020), “Optimizing LSTM for time series prediction in India stock market”, *Procedia Computer Science*, vol. 167, pp.2901-2100.
- Yu, J. H. and Kim, J. H.(2010), “Development of an incident impact analysis system using short-term traffic forecasts”, *Journal of the Korean Society of Road Engineers*, vol. 12, no. 4, pp.1-9.