박사학위논문

딥 러닝 모델 최적화 기반 순차 데이터 예측 시스템

Deep Learning Model Optimization based Sequential Data Prediction System

2018년 12월

배재대학교 대학원

컴퓨터공학과 컴퓨터공학전공

이 종 원

딥 러닝 모델 최적화 기반 순차 데이터 예측 시스템

지도교수 정 회 경

배재대학교 대학원

컴퓨터공학과 컴퓨터공학전공

이 종 원

2018년 12월

이종원의 박사학위 논문을 제출함.

이종원의 박사학위 논문을 인준함.

심사위원장 김창수 (인)

심사위원 이정희 (인)

심사위원 홍진우 (인)

심사위원 유정주 (인)

심사위원 정회경 (인)

2018년 12월

배재대학교 대학원

딥 러닝 모델 최적화 기반 순차 데이터 예측 시스템

이 종 원

지도교수 정 회 경

배재대학교 대학원 컴퓨터공학과

데이터 예측 시스템들은 데이터를 예측하기 위해 특정 분야의 데이터를 컴퓨터가 분석하여 규칙을 찾아내고 데이터를 예측하였다. 이러한 방법은 과거 데이터를 분석한 결과로 사람이 규칙을 도출할 수 있어야 데이터를 예측하는 것이 가능하였다. 이에 반해 규칙을 도출할 수 없는 데이터들의 데이터를 예측하는 것은 사람의 능력으로는 한계가 있어 정확도가 낮아지는 문제점이 발생할 수 있다.

이를 해결하기 위해 컴퓨터를 활용하여 방대한 데이터를 데이터 예측 프로그램에 학습 데이터로 입력하고 결과로 데이터를 예측하였다. 이러한 방법론을 활용하기 위해서 고성능 컴퓨터로 딥 러닝(Deep Learning) 기술을 적용하여 데이터를 예측하고 있다. 해당 방법론이 활용되고 있는 분야로는 기상 데이터를 분석하여 날씨를 예측하는 날씨 분석과 스포츠 경기의 데이터를 예측하는 것이 대표적이다.

딥 러닝 기술은 프로그램이 데이터를 기반으로 학습을 진행하고 진행된 학습을 기반으로 데이터를 처리하는 것이다. 이는 과거에 사람이 직접 데이터를 분석하는 것보다 대규모 데이터를 분석하기에 적합하고 이로 인해

정확도가 올라가는 이점이 있다. 또한 목적에 따라 적합한 딥 러닝 모델을 적용하여 데이터를 예측할 경우 정확도의 기댓값이 높아지는 이점이 있다.

현재 딥 러닝 모델 중에서 데이터를 예측하기 위해 사용되는 모델은 신경망 구조를 기반으로 하는 DNN(Deep Neural Network) 모델과 RNN(Recurrent Neural Network) 모델이다. DNN 모델은 학습 데이터 내에서 규칙을 찾아내지 못하더라도 반복 학습을 통해 데이터 예측에 대한 정확도를 올릴 수 있고, RNN은 학습 과정 중에서 은닉층에서 적용될 가중치가 학습을 진행할 수록 변화하여 데이터를 예측하고 이로 인해 정확도를 올릴 수 있다. 이에 반해 DNN은 반복 학습의 횟수가 많아야 정확도가 높아지고 RNN은 가중치 변화의 횟수가 많아져야 정확도가 높아지기 때문에 결국 두 모델들은 학습의 반복이 많아져야 하는 문제점이 있다.

본 논문에서는 데이터 예측을 위해 딥 러닝 모델 기반 순차 데이터 예측 시스템을 제안한다. 제안하는 시스템에서 비정형 데이터를 순차 데이터로 정제하기 위해 전처리기를 구현하였다. 전처리기는 딥 러닝 모델에 학습 데이터를 입력하기 전에 데이터들을 정제하는 기능을 수행한다. 데이터는 '데이터 : 인덱스' 구조로 이루어진 데이터 쌍이 되고 이러한 데이터 쌍들의 집합을 딥 러닝 모델에 입력하여 학습을 진행한다.

딥 러닝 모델은 DNN 모델, 기본 LSTM 모델, 상태유지 LSTM 모델을 활용하여 시스템을 각각 구축한다. 그리고 각 모델들의 설정 값을 변경하면서 정확도의 변화량을 분석한다. 또한 시퀀스의 길이를 변경해가며 실험을 진행하여 가장 정확도가 높은 데이터 셋과 시퀀스 길이의 비율을 제시한다.

딥 러닝 모듈 기반 시스템의 실험을 바탕으로 순차 데이터 예측에 가장 정확도가 높고 효율적인 딥 러닝 모듈을 선정하고 기존 시스템들과 비교 분석을 진행하여 제안하는 시스템의 우수성을 검증한다.

제안하는 시스템을 활용할 경우 학습 데이터가 적어도 높은 정확도를 요구하는 분야에서 기존 시스템들에 비해 효율성이 높을 것으로 사료된다.

주요어 : 데이터 정제, 딥 러닝, 머신 러닝, 전처리기, DNN, LSTM, RNN

목 차

그 림 목 차

도 표 목 차

약 어 표

ANN	Artificial Neural Network
CNN	Convolutional Neural Network
CUDA	Compute Unified Device Architecture
DNN	Deep Neural Network
GPU	Graphics Processing Unit
HDFS	Hadoop Distributed File System
HPSS	Harmonic Percussive Source Separatation
MLP	Multilayer Perceptron
LSTM	Long Short Term Memory
NN	Neural Network
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network

Ⅰ. 서 론

1.1 연구배경 및 목적

특정 분야의 데이터들을 규합하고 이를 분석하여 데이터를 예측하는 방법론이 다양한 분야에서 활용되고 있다. 이는 과거를 분석하여 미래를 예측하고 이에 대한 대비를 진행할 수 있는 장점이 있다. 예를 들어, 날씨를 분석하여 내일 날씨를 국민들에게 알려주는 기상청은 슈퍼 컴퓨터를 활용하여 과거의 날씨 정보를 분석하고 이에 대한 예측을 도출한다. 해당 방법론을 위해 지속적으로 연구된 것이 딥 러닝이다.

딥 러닝 모델은 목적에 따라 사용되는 모델이 다르다. 이는 각 모델의 구조적인 특징으로 인해 특정 목적에 적합한 모델이 있기 때문이다. 영상 처리와 이미지 처리는 CNN(Convolutional Neural Network) 모델을 주로 사용하고 있다. 영상이나 이미지를 분할하여 처리하는 CNN 모델의 구조적인 특징이 해당 목적을 정확하게 달성할 수 있다.

방대한 데이터를 학습하여 실제 상황에 적용시키는 경우에는 DNN 모델을 사용한다. DNN 모델은 프로그램이 학습한 데이터들의 규칙을 찾아 이를 상황에 맞게 적용하여 특정 상황에 적합한 결과를 도출한다. 이러한 구조적인 특징으로 인해 데이터를 예측하는 목적을 위해 기존 시스템들은 DNN 모델을 기반으로 개발되었다[1].

DNN 모델은 학습 데이터의 양이 많고 학습의 반복 횟수가 많을수록 정확도가 높아지는데 이는 프로그램의 정확도를 올리기 위해 필요한 자원이 큰 것을 의미한다. 학습 데이터의 양이 적거나 학습의 반복 횟수가 적다면 DNN 모델의 정확도는 떨어지고 이는 효율성이 낮아짐을 의미한다. 또한 Vanishing Gradient 현상이 발생할 수 있는 문제점이 있다[2].

이러한 단점을 해결하기 위해 데이터 예측 시스템에 사용되고 있는 다른 모델은 RNN 모델이다. RNN 모델은 프로그램이 학습을 진행할수록 은닉층에서 사용되는 가중치의 값이 학습 상황에 따라 변화하여 학습의 반복 횟수가 적더라도 DNN 모델에 비해 높은 정확도를 기대할 수 있다. 이에 반해 RNN 모델은 반복 횟수가 늘어남에 따라 가중치의 변화 횟수가 증가하고 오차 범위가 늘어날 수 있다.

데이터를 예측하는 프로그램은 어떤 상황에서든지 높은 정확도를 보장받을 수 있어야 한다. DNN 모델과 같이 학습 데이터의 양이 많고 학습의 반복 횟수가 많아야 하는 전제조건은 데이터를 예측하는 프로그램에 적합하다고 보기 어렵다. 또한 RNN 모델처럼 정확도의 오차 범위가 큰 경우도 적합한 모델이라고 볼 수 없다.

이러한 문제점들을 해결하기 위해 데이터를 예측하는 프로그램의 기반이 되는 딥 러닝 모델이 갖춰야할 필수조건들은 다음과 같다.

- 학습 데이터의 양이 적어도 정확도를 보장
- 반복 학습의 횟수가 적어도 정확도를 보장
- 입력 데이터의 종류를 제한하지 않음

본 논문에서 제안하는 시스템은 비정형 데이터 셋을 전처리기에서 순차 데이터로 정제하고 학습하여 특정 데이터 후에 나올 데이터를 예측하는 시스템이다.

딥 러닝 모델의 구조적인 특징으로 인해 배치사이즈, 데이터 셋과 시퀀스 길이에 따라 정확도가 다르게 도출될 수 있다. 이에 대한 연구를 진행하여 최적화된 배치사이즈, 데이터 셋과 시퀀스 길이의 비율을 제시한다. 이를 통해 기존의 딥 러닝 모델 기반 데이터 예측 시스템에 비해 높은 정확도를 기대할 수 있다.

제안하는 시스템을 활용할 경우 기존의 데이터를 예측하는 시스템들의 문제점이었던 학습 데이터의 양과 반복 학습의 횟수에 대한 의존 문제를 해결할 수 있고 데이터의 유동적인 변화에 보다 높은 정확도를 보장받을 수 있다. 이러한 장점들을 기반으로 제안하는 시스템은 데이터를 예측하기 위해 개발된 기존 시스

템들의 연구 방향을 제시할 수 있을 것으로 사료된다.

1.2 연구내용 및 범위

데이터 예측 시스템은 사용자가 적은 리소스를 투입하여도 높은 정확도로 데이터를 예측할 수 있어야 한다. 이러한 목적에 적합한 딥 러닝 모델 기반 시스템을 연구하기 위해 기존의 데이터 예측 시스템을 분석하고 제안하는 시스템과 비교 실험을 통해 시스템의 효율성을 검증한다.

- 기존의 딥 러닝 모델 기반의 데이터 예측 시스템을 분석하고 데이터 예측 시스템으로써 갖춰야할 요구사항 분석
- 딥 러닝 모델이 효율적인 학습을 진행할 수 있도록 데이터 셋을 '데이터 : 인덱스' 구조로 정제하는 전처리기를 개발
- 학습 및 데이터 예측의 정확도를 상승시키기 위해 데이터 셋과 시퀀스 길이의 비율, 배치사이즈, 가중치 초기화 사이즈에 대한 연구를 진행하고 각 모델 기반 시스템들의 최적화된 설정 값 도출
- 도출한 설정 값을 기반으로 데이터 예측 시스템 설계 및 구현
- 28개 데이터 셋과 52개 데이터 셋으로 실험을 진행하여 DNN 모델 기반 시스템과 기본 LSTM 모델 기반 시스템, 상태유지 LSTM 모델 기반 시스템을 비교 분석하고 정확도와 효율성 2가지를 기준으로 적합한 시스템을 선정
- 기존의 데이터 예측시스템과 비교 분석을 진행하고 제안하는 시스템의 우수성을 검증

1.3 논문의 구성

본 논문에서는 데이터 예측을 위해 기존에 사용되고 있는 DNN 모델과 RNN 모델의 문제점을 분석하고 이를 해결하기 위해 전처리기와 딥 러닝 모델이 결합한 데이터 예측 시스템을 제안한다..

2장에서는 기존의 데이터 예측 시스템들에 대한 분석을 진행하고 데이터 예측 시스템이 갖춰야할 요구사항을 분석한다.

3장에서는 전처리기와 딥 러닝 모델로 구성된 시스템을 설계한다. 전처리기는 데이터 예측을 위해서 비정형 데이터를 '데이터 : 인덱스' 구조의 순차 데이터로 변환하는 기능을 수행한다. 그리고 DNN 모델과 기본 LSTM 모델, 상태유지 LSTM 모델 기반 시스템을 설계하고 각 모델 별로 배치사이즈, 가중치 초기화 사이즈 등 정확도 향상을 위해 설정 값 최적화에 대한 연구를 진행하고 정확도와 효율성을 증대시킬 수 있도록 한다.

4장에서는 3장에서 설계한 시스템들을 구현하고 3가지 시스템들을 비교 분석한다. 그리고 가장 우수한 성능의 시스템과 기존 시스템을 비교하여 정확도와 효율성을 기준으로 제안하는 시스템을 평가한다.

5장에서는 해당 연구의 결론을 서술한다.

Ⅱ. 기존 데이터 예측 시스템 분석

본 장에서는 딥 러닝 모델을 활용하여 구축한 데이터 예측 시스템들을 분석하고 데이터 예측 시스템이 필요로 하는 요구사항을 정리한다.

2.1 DNN 모델 기반 시스템

DNN 모델은 ANN(Artifitial Neural Network) 모델을 기초로 하고 있다. ANN 모델은 사람의 신경망 원리 및 구조를 모방하여 만든 기계학습 알고리즘이다. 이런 ANN 모델은 여러 문제점을 가지고 있었는데, 학습 과정에서 파라미터의 최적값을 찾기 어려운 문제가 있었다. Sigmoid 함수의 사용은 기울기 값에 의해 가중치가 결정되는데 이런 기울기 값이 뒤로 갈수록 작아져 0에 수렴하는 오류가 있었고 로컬 미니마(Local Minima)를 최저 에러로 인식하여 더 이상 학습을 진행하지 않는 경우도 있다. 또한 학습 데이터에 따른 과적합 문제가 있었다. 데이터가 많지 않은 경우 학습 데이터에만 특화되어 이전의 형태와 다른 데이터를 받았을 때 처리를 하지 못하는 단점이 있었다.

마지막으로 ANN 모델은 학습 시간이 너무 느린 단점이 있었다. 은닉층이 증가될수록 연산량이 급격히 증가하여 하드웨어에 많은 부담이 되었다.

이러한 ANN 모델의 문제점은 후에 많은 방법으로 개선되었다. 과적합과 로컬 미니마는 사전훈련을 통해 과적합을 방지할 수 있는 Initialize Point 알고리즘이 개선되었고 ReLU(Rectified Linear Unit)나 Drop-out 기법 등으로 로컬 미니마 현상도 해결할 수 있었다. 느린 학습시간은 그래픽 카드의 발전으로(GPU, CUDA) 학습시간이 빠르게 개선되었고 부족한 데이터 예측은 딥 러닝 모델을 활용하여 해결되었다.

이 ANN 모델의 여러 문제가 해결되면서 모델의 은닉층을 많이 늘릴 수 있었는데 여기서 새로 나온 방법이 DNN 모델이다. DNN 모델은 은닉층을 2개 이상 지닌 학습 방법으로, 컴퓨터가 스스로 분류 레이블을 만들어 내고 공간을 왜곡하고 데이터를 구분 짓는 과정을 반복하여 최적의 구분선을 도출해낸다. 많은 데이터와 반복 학습을 필요로 하며 Pre-training 기법과 Back-propagation을 통해 현재 최적의 학습방법으로 사용되고 있다. 여러 고차원에서 데이터 분류가 가능해짐에 따라 DNN 모델은 다양한 분야에 활용되기 시작했는데 영상처리, 음성인식, 자연어 처리까지 DNN 모델을 통해 학습할 수 있게 되었다. 대표적인 DNN 모델을 응용한 모델로는 CNN, RNN, LSTM 등이 있다.

그림 1은 DNN 모델의 구조를 나타내고 그림 2는 딥 러닝 모델이 활성함수 Sigmoid 함수와 Back-propagation을 활용하였을 때 발생할 수 있는 Vanishing Gradient 문제를 나타낸 그림이다[1, 2].

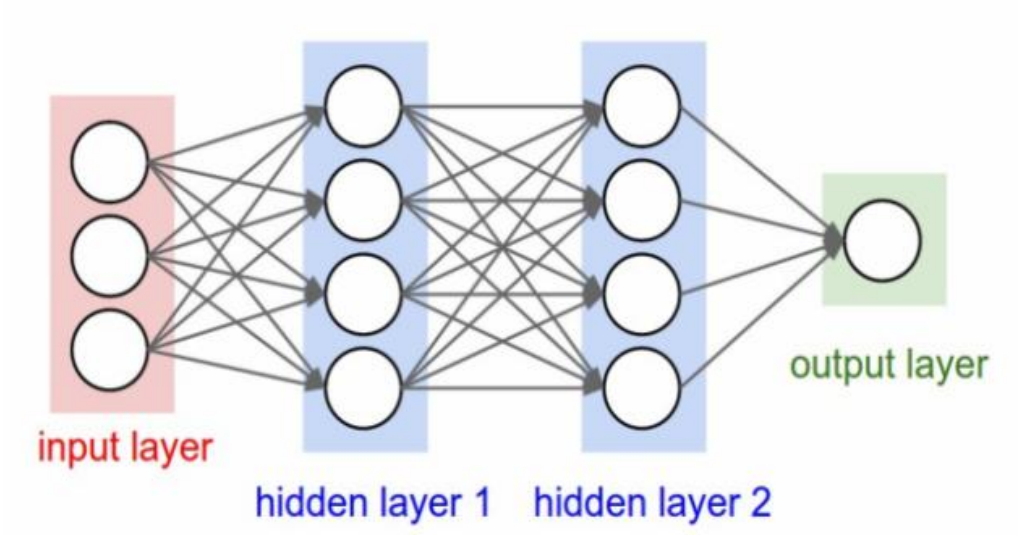


그림 1. DNN 모델의 구조

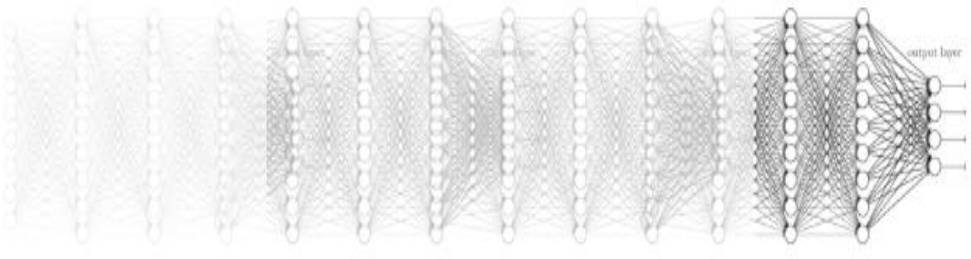


그림 2. Vanishing Gradient

Vanishing Gradient는 Sigmoid 함수의 문제점으로써 발생한다. 0과 1 사이의 값을 전달하게 되는데 이로 인해 전달되는 값이 처음 입력한 값에 비해 현저하게 작아지는 현상이 발생하여 정확도가 떨어지는 것이다. 이를 해결하기 위해서는 Sigmoid 함수을 대체할 함수가 필요했고 이를 해결한 함수가 ReLU이다. 그림 3은 두 함수를 그림으로 나타낸 것이다[3].

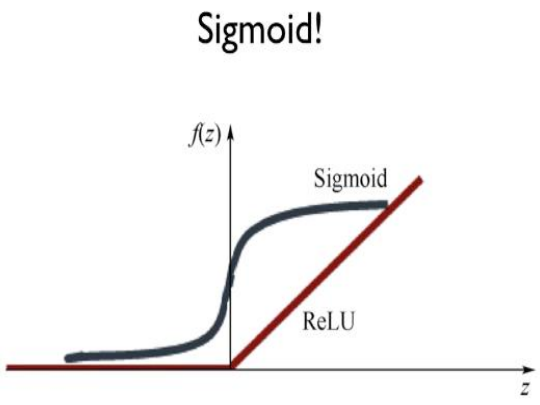


그림 3. Sigmoid 함수와 ReLU 함수

ReLU 함수는 전달하려는 값이 0보다 작을 때는 0을 전달하고 0보다 큰 값은 해당 값을 그대로 전달하는 방식이다. 이로 인해 Sigmoid 함수를 활용했을 때보다 오차범위가 줄고 정확도가 향상되었다. 그림 4는 DNN 모델을 활용한 미디어 오디오를 분석하여 음성을 검출하는 시스템이다.

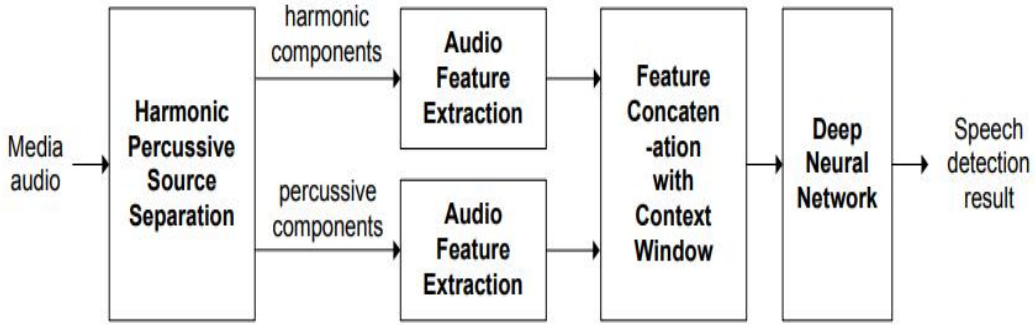


그림 4. 오디오 분석 기반 음성 검출 시스템

그림 4의 오디오 분석 기반 음성 검출 시스템은 입력된 미디어의 오디오 신호를 HPSS(Harmonic Percussive Source Separatation) 알고리즘을 이용하여 고조파 및 퍼커시브 성분으로 분해하고 오디오 특징을 추출하고 오디오의 문맥 정보를 포함하도록 입력 벡터를 구성한다. 문맥 정보들을 DNN 모델이 학습하고 이를 기반으로 음성으로 예측하고 검출을 수행한다. 그림 5는 DNN 은닉층 개수에 따른 손실 값을 그래프로 정리한 것이다.

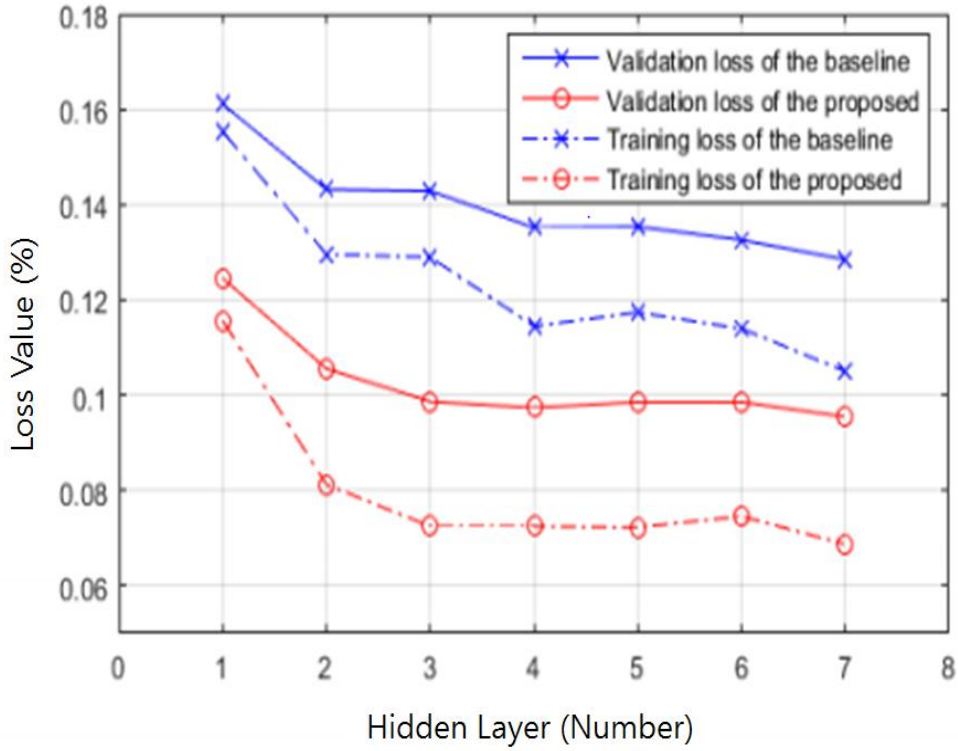


그림 5. 은닉층 개수에 따른 손실 값 그래프

그림 5의 결과에 따라 DNN 모델의 은닉층 개수가 4개 이상일 때 손실 값의 차이가 거의 없는 것으로 나타났고 실험을 위해 4개의 은닉층을 활용하였다고 기술되어있다[4]. 해당 시스템은 음성 인식 분야에 DNN 모델을 적용하여 기존에 사용되고 있는 시스템들에 비해 성능 향상을 검증하였다[5-7].

그림 6은 비디오 영상 내에 이벤트들을 분석하고 비디오 영상들을 이벤트를 기준으로 분류하는 시스템을 CNN 모델 기반으로 구성했을 때와 DNN 모델 기반으로 구성했을 때의 정확도 차이를 표로 정리한 것이다.

Type	CNN		MLP	
	Hit @1	Hit @5	Hit @1	Hit @5
Crop & Color	18.6 %	28.1 %	55.9 %	74.1 %

그림 6. CNN 모델 기반과 DNN 모델 기반 정확도 차이

그림 6의 비디오 분류 시스템은 비디오를 이미지와 오디오 두가지 성분으로 구성되어 있다고 가정하고 성분들의 시간을 동기화하고 시간에 맞게 재배열한 뒤 이미지들을 분석하여 이벤트들을 추출한다. 그리고 CNN 모델과 DNN 모델을 활용하여 비디오 내에 추출한 이벤트들을 분류한다. 해당 논문에서 제안하는 시스템은 기존 시스템에 비해 성능이 낮았으나 비슷한 방법을 활용하여 정확도가 높은 시스템을 분석하고 DNN 모델을 LSTM 모델로 변경하여 시스템을 구축할 경우 성능이 향상될 것이라고 예측하고 있다[8-11]. 해당 시스템은 비디오 및 이미지 처리 분야에 DNN 모델을 적용하여 기존에 사용되고 있는 시스템들과 다른 접근 방식을 제안하였다.

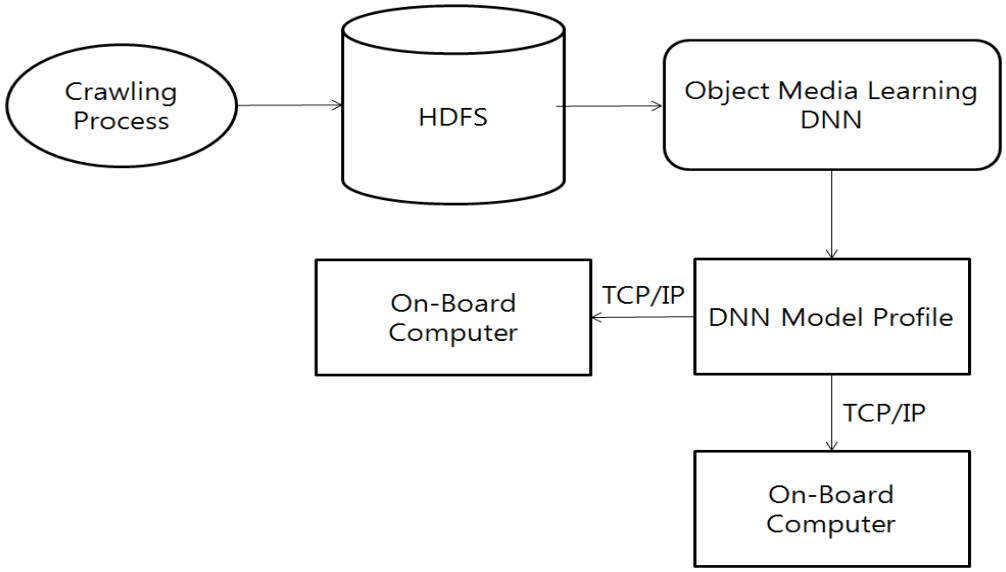


그림 7. 객체 학습 및 인식 시스템

그림 7의 시스템은 온톨로지(Ontolohy)를 구축하여 카테고리 단위로 데이터를 수집하고 수집된 데이터들을 DNN 모델을 활용하여 학습하고 객체를 인식하도록 하는 시스템이다. 그림 8은 해당 시스템의 온톨로지 구성과 크롤링(Crawling) 과정을 나타낸다.

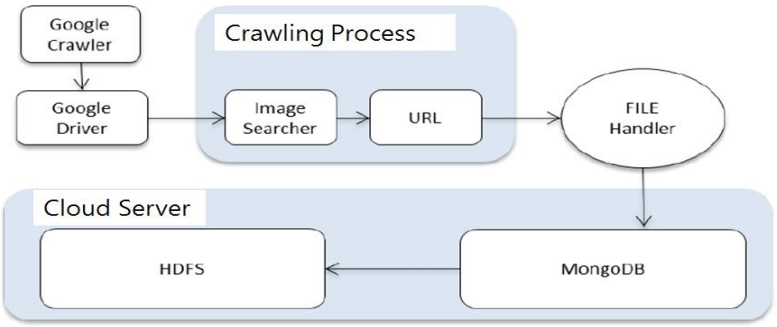


그림 8. 온톨로지 구성 및 크롤링 과정

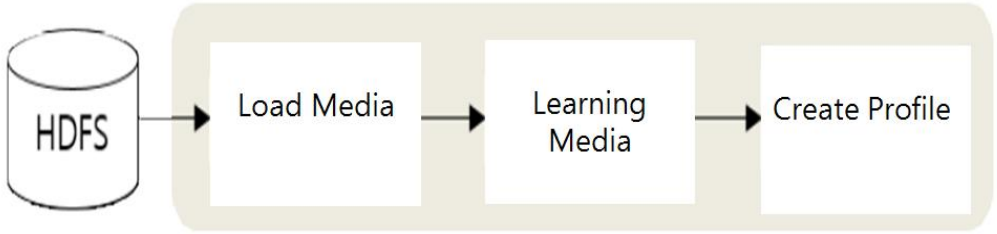


그림 9. 객체 영상 학습 서버 시스템 구성도

그림 9는 객체 영상을 학습하여 프로파일을 생성하는 시스템의 구성도를 나타낸다. 그리고 그림 10과 같은 프로파일이 생성된다.

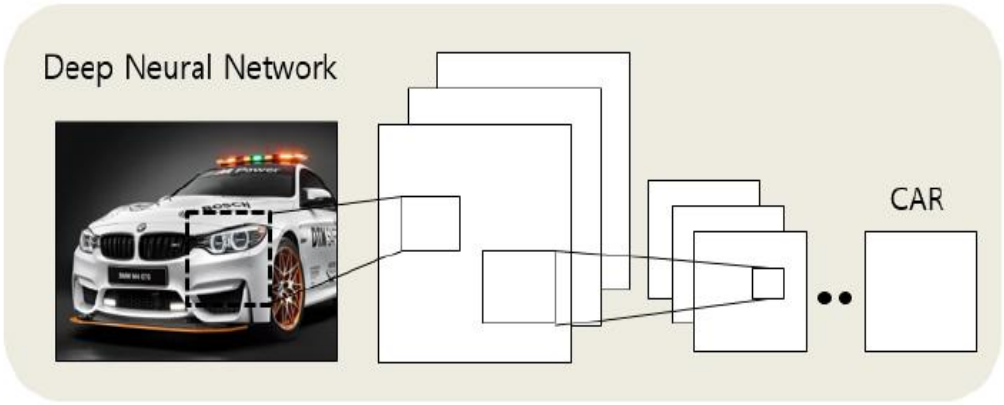


그림 10. DNN 모델의 구조

그림 10과 같이 각 이미지에 대한 프로파일 생성이 그림 11과 같은 구조로 구성된 객체 인식 시스템에 객체를 인식하도록 한다.

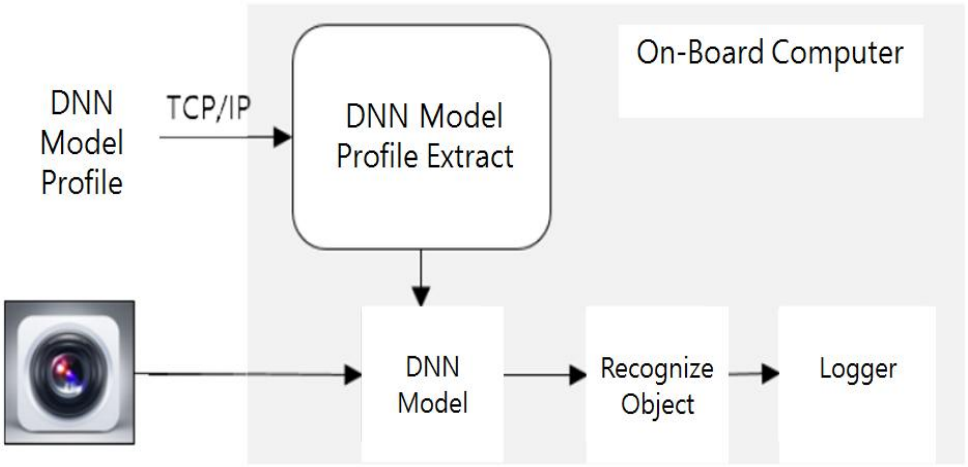


그림 11. 모델 동기화 구조

해당 시스템은 영상이나 이미지를 DNN 모델로 학습하고 이를 기반으로 객체를 인식하는 시스템이다[12-14]. 기존에는 이러한 목적을 달성하기 위해 CNN 모델을 활용하였으나 본 논문에서는 DNN 모델 기반으로 시스템을 구축하여도 해당 기능을 수행할 수 있다는 점을 시사하였다[15-18].

2.2 RNN 모델 기반 시스템

RNN 모델은 인공신경망 모델의 한 종류로써 데이터 셋을 모델에 입력하면 입력층과 은닉층, 출력층까지 노드를 한 번씩 거쳐서 진행되는 일반 인공신경망 모델과는 다른 구조로 작동된다. 입력에 대한 은닉층의 값을 모델 내부의 메모리에서 기억하고 다음 입력 값을 출력할 때 적용되는 가중치에 영향을 준다. 이로 인해 시계열적인 정보를 효과적으로 분석하는 특징이 있다.

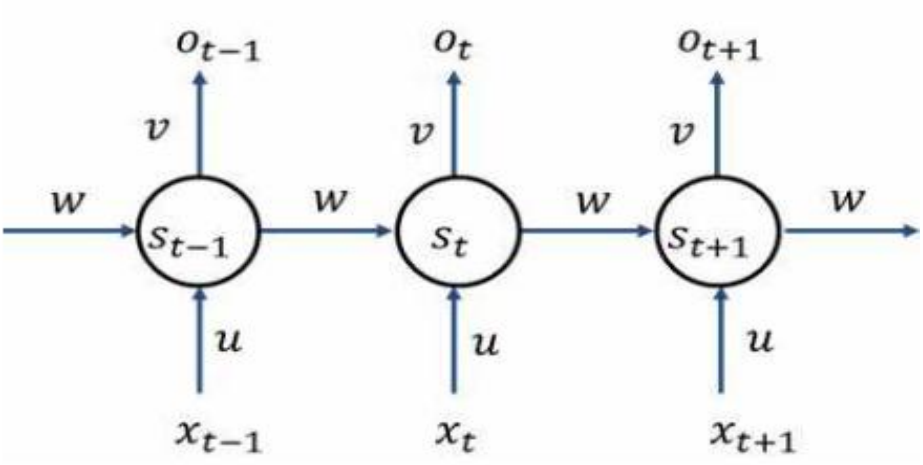


그림 12. RNN 모델의 구조

그림 13은 모바일 RPG 게임 고객의 시계열적인 행동 패턴을 분석하고 이탈을 예측하는 시스템의 구조도이고 그림 14는 일반 NN(Neural Network) 모델과 비교 실험을 진행한 결과이다[19-23].

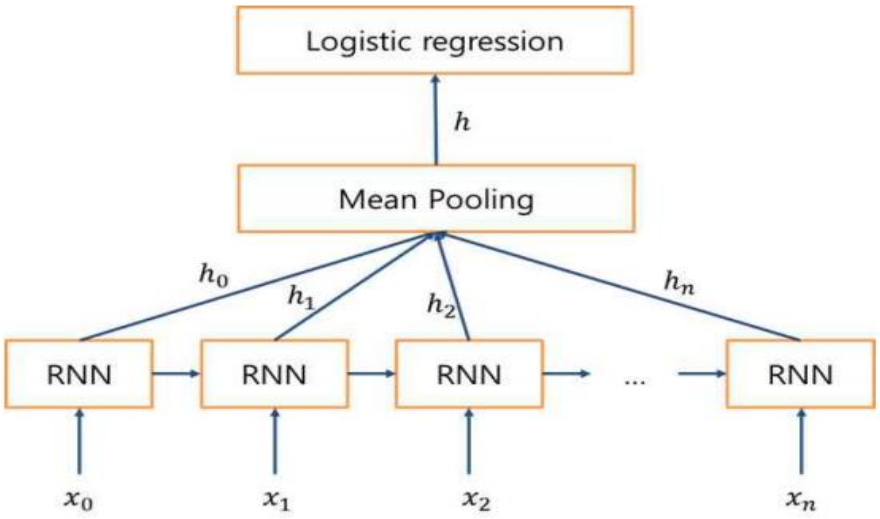


그림 13. 고객 이탈 예측 시스템 구조도

Model	7 days play Data	14 days play Data	20 days play Data
RNN	74.5	76.0	75.7
NN	60.6	62.3	60.7

그림 14. 고객 이탈 예측 시스템 비교 실험 결과

제안하는 시스템은 일반 NN 모델을 활용하는 시스템에 비해 정확도가 높다고 기술되어있다. 이는 시간의 흐름에 따라 데이터를 정리한 뒤 이를 데이터 셋으로 정제하여 RNN 모델을 활용했기 때문으로 분석된다. 일반 NN 모델은 다양하고 광범위한 데이터를 분석하기에 적합하지만 데이터 셋을 정제하여 심층 분석을 진행할 경우 NN 모델을 개선한 DNN 모델과 RNN 모델을 활용하는 것이 보다 높은 정확도를 보이는 것으로 확인되었다[24-26].

2.3 LSTM 모델 기반 시스템

LSTM 모델은 RNN 모델이 긴 시퀀스 데이터를 분석하면서 발생할 수 있는 장기의존성 문제를 해결하기 위해 개발된 모델이다. LSTM 모델은 3종류의 게이트와 재귀적 구조를 가진 메모리 셀로 이루어져있다. 3종류의 게이트에서 정보의 흐름을 조절하면서 은닉변수를 거치고 최종 출력값을 계산하게 된다. 그림 15는 LSTM 모델의 구조도이고 그림 16은 LSTM 모델 기반 한국어 문장 생성 시스템의 구조도이다.

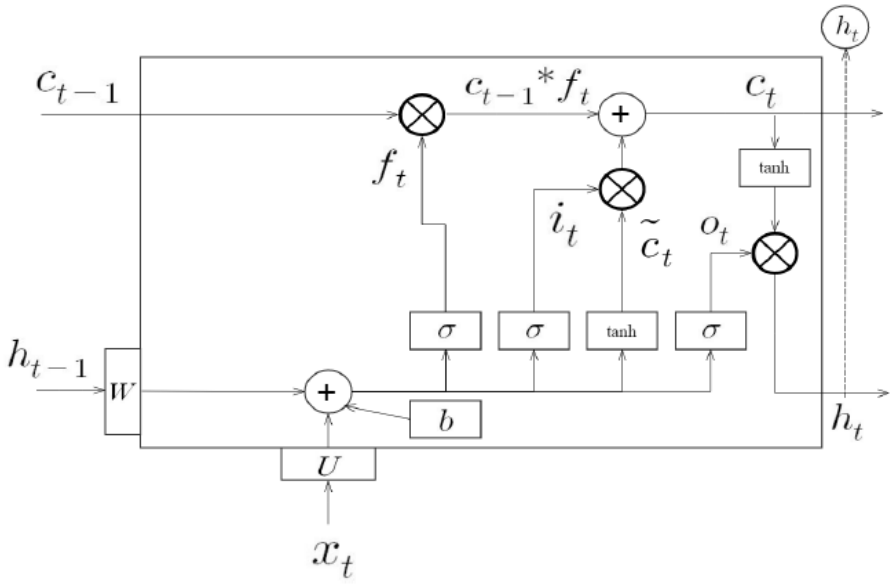


그림 15. LSTM 모델의 구조도

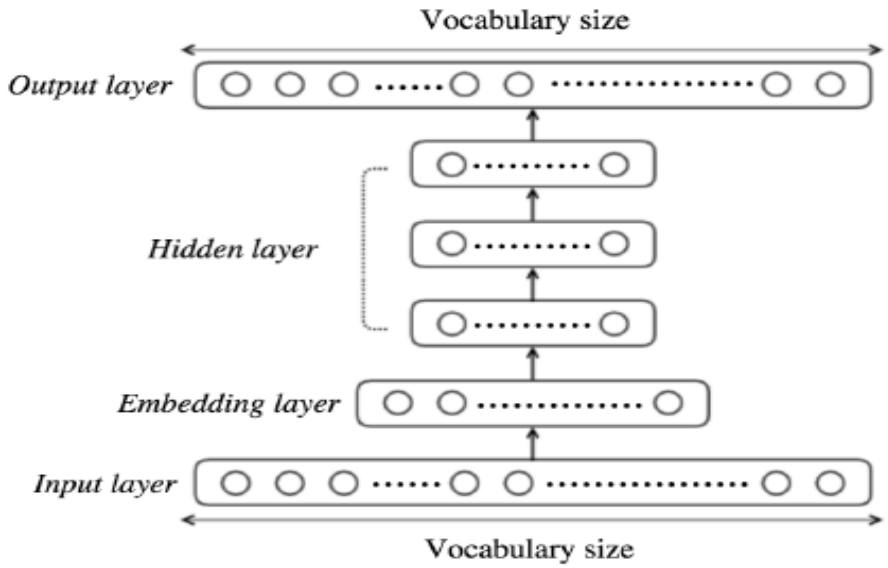


그림 16. LSTM 모델 기반 한국어 문장 생성 시스템 구조도

그림 16의 시스템은 LSTM 모델을 기반으로 문장의 일부분이 주어졌을 때 나머지 부분을 생성한 뒤 완성시키는 시스템이다[27-30]. 해당 논문에서는 데이터 셋과 시퀀스 길이에 대한 연구를 통해 기존 시스템들과의 차별성을 언급하였고 성능에 중요한 영향을 미친 것으로 판단하였다[31-33].

2.4 순차 데이터 예측 시스템 개발을 위한 요구사항 분석

데이터 예측 시스템은 특정 데이터 셋을 학습하여 다음에 나올 데이터를 예측하는 기능을 사용자에게 제공해야 한다[34-36]. DNN 모델 기반 시스템들은 DNN 모델의 은닉층의 개수를 변경해가며 실험을 진행하였고 이에 따라 정확도를 상승시켰다. RNN 모델 기반 시스템들과 LSTM 모델 기반 시스템들은 데이터 셋을 순차 데이터로 정제하고 이를 학습하여 데이터를 처리하였고 이에 따라 정확도를 상승시킨 연구 결과를 확인하였다[37-39].

이러한 연구들을 바탕으로 본 논문에서는 전처리기와 딥 러닝 모델을 기반 데이터 예측 시스템을 제안한다. 전처리기는 비정형 데이터를 순차 데이터로 변환한 뒤 딥 러닝 모델이 학습할 수 있도록 한다[40, 41]. 그리고 딥 러닝 모델의 설정 값들에 대한 연구를 진행하여 최적화된 수치를 도출한 뒤 이를 시스템에 적용한다. 또한 데이터 셋과 시퀀스 길이의 변화에 따라 도출되는 정확도를 분석하여 모델 별로 가장 높은 정확도를 도출한 데이터 셋과 시퀀스 길이의 비율을 제시한다[42-44].

이러한 설정 값 최적화를 기반으로 DNN 모델 기반 시스템과 기본 LSTM 모델 기반 시스템, 상태유지 LSTM 모델 기반 시스템을 구현하고 비교 실험을 진행한다. 실험 결과를 정확도와 효율성을 기준으로 분석하고 3가지 시스템 중 적합한 시스템을 선정한다. 그리고 기존의 데이터 예측 시스템과의 비교 분석을 통해 제안하는 시스템의 성능을 검증한다.

Ⅲ. 데이터 예측 시스템 설계

본 장에서는 DNN 모델 기반 시스템과 기본 LSTM 모델 기반 시스템, 상태유지 LSTM 모델 기반 시스템을 설계한다. 또한 각 모델 별로 정확도에 영향을 미치는 설정 값들에 대한 연구를 진행한다.

DNN 모델 기반 시스템은 은닉층의 개수와 배치사이즈에 대한 연구를 진행하여 최적화된 수치를 도출하고 구현을 위한 설계를 진행한다.

기본 LSTM 모델 기반 시스템은 배치사이즈에 대한 연구를 진행하고 상태유지 LSTM 모델 기반 시스템은 가중치 초기화 사이즈에 대한 연구를 진행한 뒤 설계를 진행한다.

3.1 제안하는 시스템의 개요

그림 17은 기존에 사용되고 있는 데이터 예측 시스템의 구성도이다.

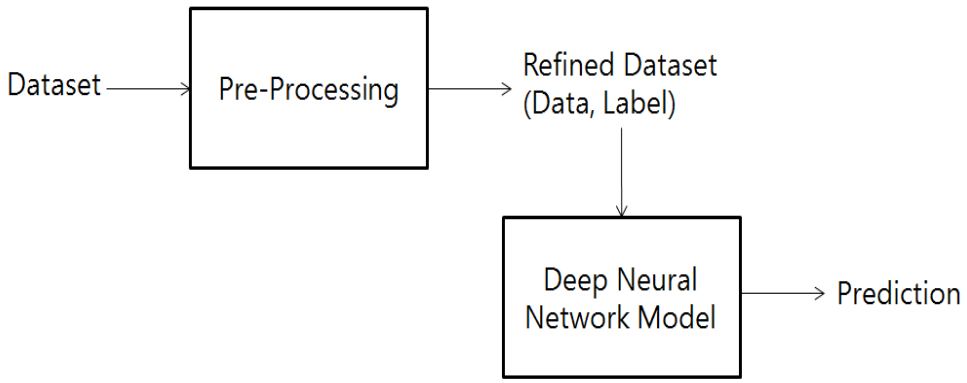


그림 17. 기존 데이터 예측 시스템의 구성도

기존의 데이터 예측 시스템은 사용자가 데이터 셋을 정제하고 이를 딥 러닝 모델에 학습 데이터로 사용한다. 그리고 딥 러닝 모델이 학습을 완료하고 데이터 예측을 수행한다. 그림 18은 제안하는 시스템의 구성도이다.

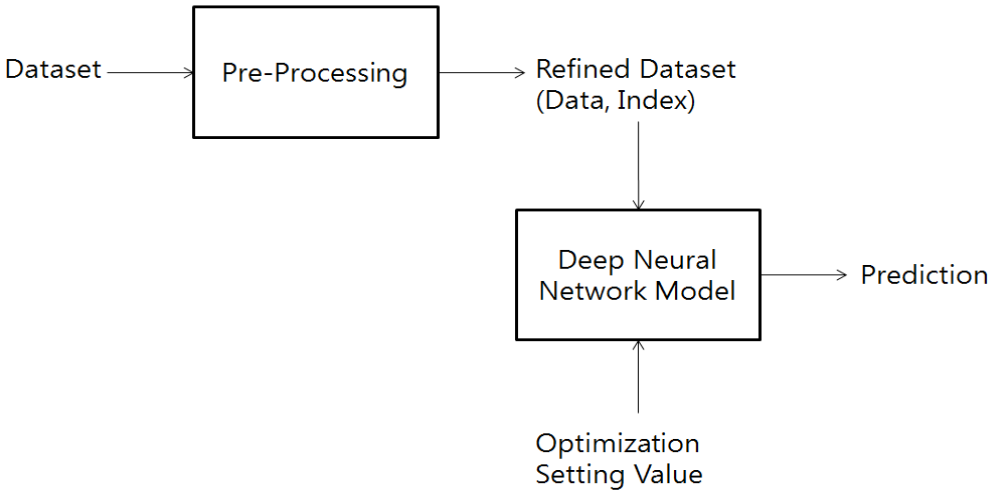


그림 18. 제안하는 시스템의 구성도

제안하는 시스템은 사용자가 데이터 셋을 '데이터 : 라벨값' 구조로 정제하는 작업을 전처리기가 수행한다. 기존의 데이터 정제 작업과의 차이점은 다음과 같다.

- 기존 시스템의 데이터 정제 작업은 '데이터 : 라벨값' 구조로 데이터 정제 작업을 수행한다.
- 제안하는 시스템의 데이터 정제 작업은 '데이터 : 인덱스' 구조와 '인덱스 : 데이터' 구조로 데이터 정제 작업을 수행한다.

두 가지 구조로 데이터를 정제하는 이유는 다음과 같다.

- 기존 시스템은 설정된 배치사이즈 만큼 가중치를 갱신하며 '데이터 : 라벨

값' 구조의 데이터 셋으로 학습을 진행한다.

- 제안하는 시스템은 최적화된 배치사이즈 만큼의 횟수만큼 '데이터 : 인덱스' 구조와 '인덱스 : 데이터' 구조의 순차 데이터 셋으로 학습을 진행한다. 이는 RNN 모델과 LSTM 모델은 데이터 셋을 순차 데이터로 정제한 뒤 이를 학습하면 모델의 구조적인 특징으로 인해 정확도가 상승되기 때문이다.

기존 시스템은 데이터 셋과 시퀀스 길이의 비율에 대해 연구를 진행한 경우가 미흡하며 본 논문에서는 이를 진행한다. 이러한 연구를 진행하는 이유는 다음과 같다.

- 기존 시스템은 은닉층의 개수에 대한 연구를 진행한 경우가 대부분이며 데이터 셋과 시퀀스 길이의 비율에 대한 연구는 미흡한 실정이다. 시퀀스 길이를 변경할 경우 같은 데이터 셋으로 학습 및 데이터 예측을 진행했을 때 정확도가 다르게 도출되는 경우가 많으며 이는 정확도 향상을 위해 연구를 진행하여 최적화된 수치를 도출해낼 필요성이 있다.
- 제안하는 시스템은 DNN 모델 기반 시스템과 기본 LSTM 모델 기반 시스템, 상태유지 LSTM 모델 기반 시스템의 데이터 셋과 시퀀스 길이의 비율을 변경해가며 실험을 진행하고 최적화된 수치를 도출하여 제시한다.

본 논문에서 제안하는 시스템은 딥 러닝 모델 중 DNN 모델 기반 시스템과 기본 LSTM 모델 기반 시스템, 상태유지 LSTM 모델 기반 시스템을 설계 및 구현한다. 이는 각 모델의 구조적인 특징으로 인해 지니고 있는 장단점을 분석하기 위함이다. 또한 각 모델 별 설정에 대한 실험을 통해 최적화된 수치를 도출하고 이를 바탕으로 시스템을 설계 및 구현할 경우 데이터 예측에 가장 적합한 시스템을 선정할 수 있다.

3.2 전체 시스템 설계

제안하는 시스템은 전처리기와 딥 러닝 모델로 설계하였다. 그림 19는 시스템의 구조도이다.

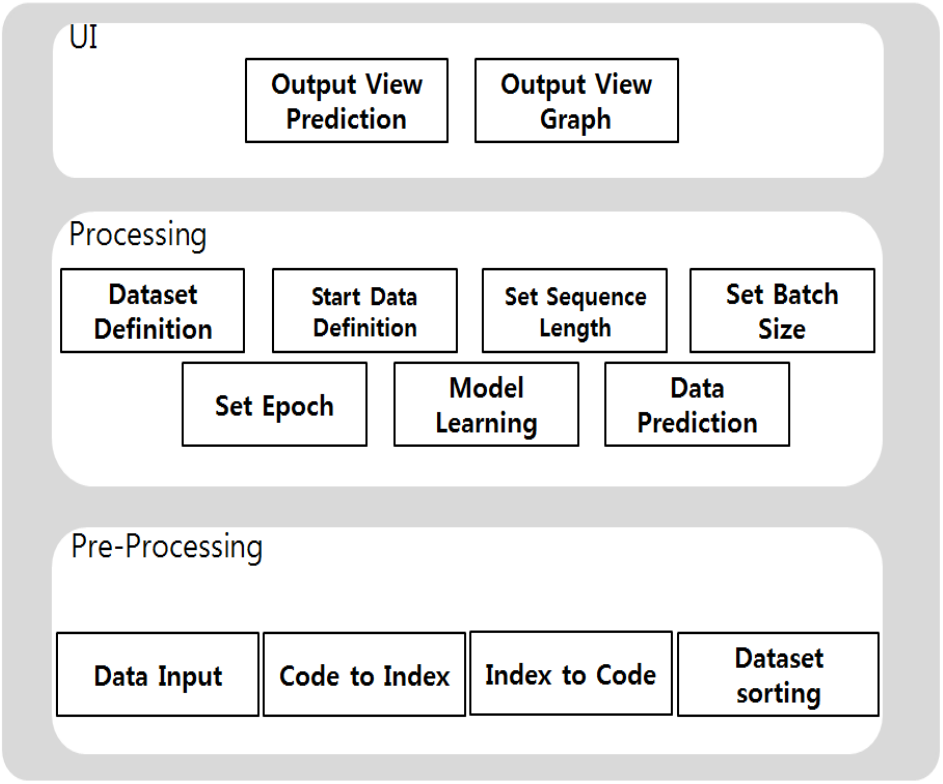


그림 19. 시스템 구조도

전처리기에서는 사용자가 입력한 데이터를 '데이터 : 인덱스', '인덱스 : 데이터' 구조로 데이터를 변환하고 이를 딥 러닝 모델에 적용한다. 그리고 데이터 예측에 적용할 시작 데이터를 설정하고 시퀀스 길이, 배치사이즈, 반복 학습의 횟수 등을 정의한 뒤 학습을 시작한다. 학습이 완료되면 시스템은 데이터를 예측하고 이에 대한 결과들을 그래프로 정리하여 사용자에게 보여주도록 한다. 이는 제안하

는 시스템이 제공하는 기능들의 정확도를 사용자가 알 수 있게 한다. 그림 20은 시스템의 흐름도이다.

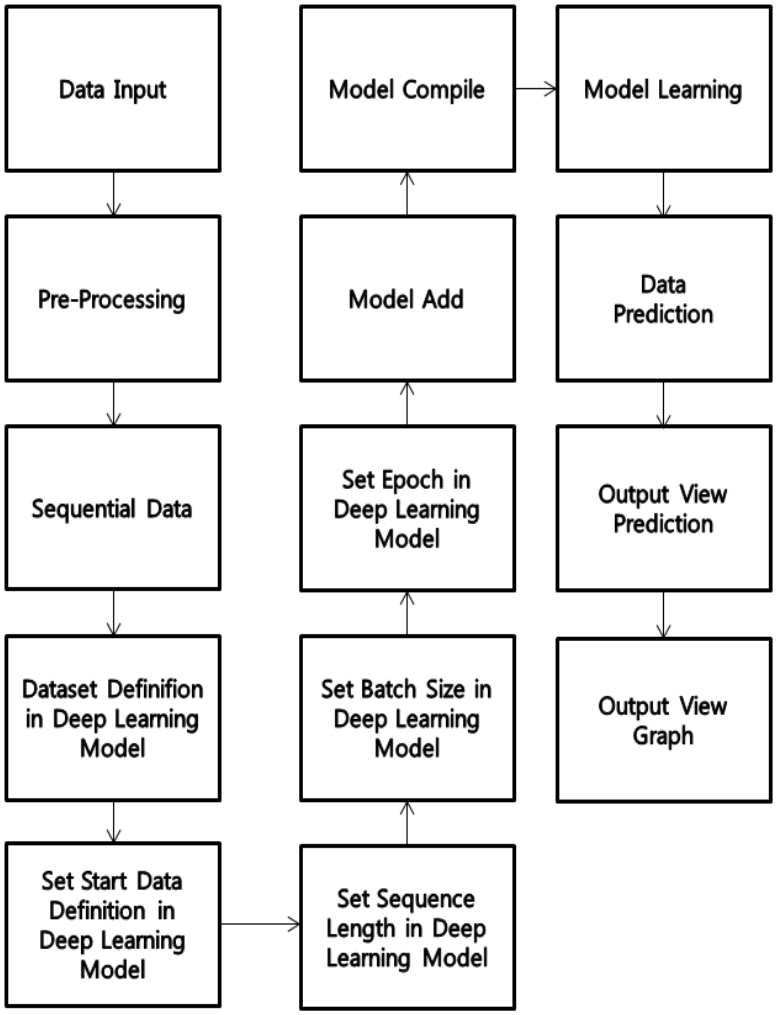


그림 20. 시스템 흐름도

시스템이 시작되면 전처리기는 사용자가 입력한 데이터를 두 가지 패턴으로 데이터를 변환한다. 데이터는 '데이터 : 라벨값' 형태의 구조로 변환되고 실제 데이터는 '데이터 : 인덱스'와 '인덱스 : 데이터' 구조이다. 전처리기에서의 작업이 완료되면 전처리기가 정제한 데이터를 입력 데이터로 삽입하면 된다.

제안하는 시스템은 전처리기가 정제한 순차 데이터를 삽입하고 최적화된 배치 사이즈 만큼의 가중치 갱신을 바탕으로 학습을 진행한다. 목표가 되는 데이터 셋을 정제하는 방법과 최적화된 배치사이즈, 데이터 셋과 시퀀스 길이의 비율 설정을 수행한다.

전처리기와 순차 데이터 예측을 위한 딥 러닝 모델로 구성된 제안하는 시스템은 DNN 모델과 2 가지 LSTM 모델로 성능 비교 실험을 진행하여 효율성을 검증하여야 한다. LSTM 모델은 기본 LSTM 모델과 상태유지 LSTM 모델의 차이점을 명시한다.

3.2.1 전처리기 설계

딥 러닝 모델이 순차 데이터를 활용하여 학습을 진행하고 데이터를 처리할 수 있도록 데이터 정제 작업을 수행해줄 전처리기가 필요하다. 전처리기는 딥 러닝 모델들이 정제된 순차 데이터들을 기반으로 학습을 수행할 경우 정확도가 상승되는 점에서 고안된 방법이다. 그림 21은 전처리기의 슈도코드이고 그림 22는 전처리기의 데이터 흐름도이다.

```
Procedure Pre-Processing()
                Input(Dataset)
                Code to Index(Dataset, Dataset(Data, Index))
                Index to Code(Dataset, Dataset(Index, Code))
                Output(Refined Dataset)
end Procedure
```

그림 21. 전처리기 슈도코드

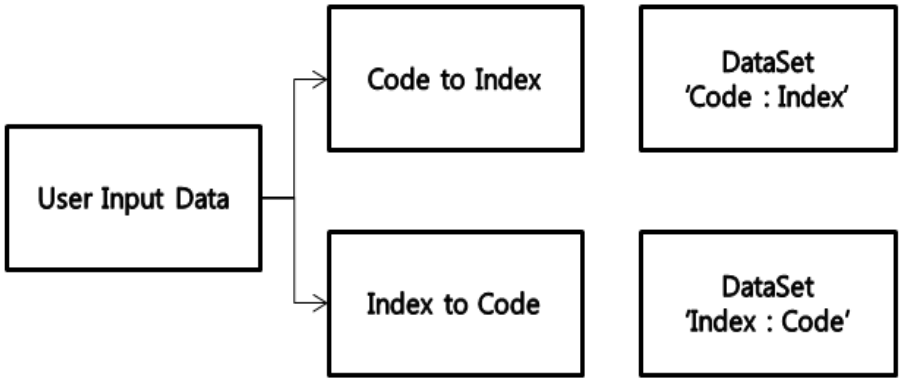


그림 22. 전처리기 데이터 흐름도

전처리기에서는 사용자가 입력한 데이터 셋을 두 가지 형태로 정제한다. 기존의 데이터 예측 시스템에서는 딥 러닝 모델의 소스 코드에 사용자가 직접 라벨 값을 부여하는 형식으로 정제 작업을 진행하였다. 이러한 상황을 방지하고자 제안하는 시스템에서는 전처리기가 수행한다.

Java로 구현된 전처리기에서는 Scanner 클래스를 기반으로 구현된 기능으로 사용자로부터 데이터를 입력받는다. 입력 받은 데이터는 두 가지 패턴으로 정제되는데 두 가지 패턴 모두 '데이터 : 라벨값' 구조로 도출된다. 이는 LSTM 모델이 데이터와 인덱스 두 가지 값을 가지고 학습을 수행하기 때문에 이와 같은 형

식으로 두 종류의 값을 도출한다.

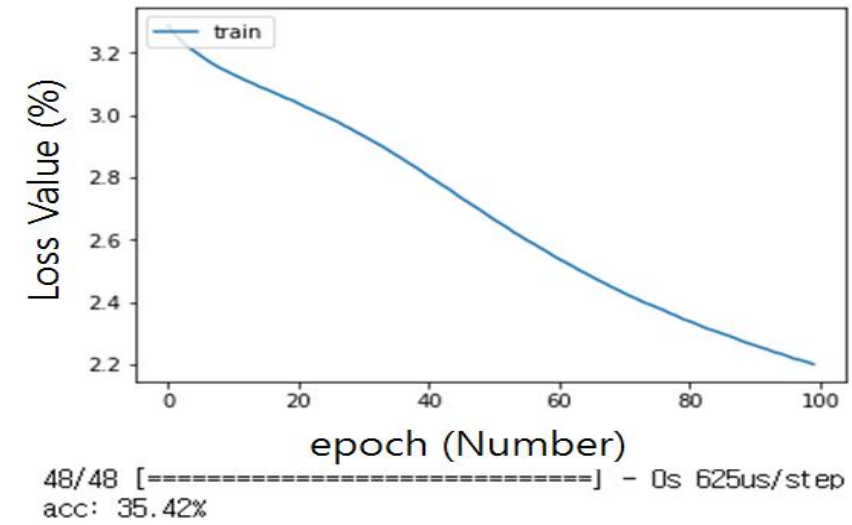
3.2.2 각 모델 별 설정 값 최적화 연구

데이터 정제 작업을 마친 데이터 셋은 '데이터 : 인덱스' 구조이며 비정형 데이터에서 순차 데이터로 정제된다. 딥 러닝 모델들의 배치사이즈를 변경해가며 실험을 진행하여 최적화된 수치를 도출한다. 그리고 정제된 데이터들을 모델이 학습하고 데이터를 예측하는 작업을 수행할 때 데이터 셋과 시퀀스 길이의 비율을 분석하여 기대 정확도를 높일 수 있다.

배치사이즈 최적화를 위해서는 각 모델 별로 같은 데이터 셋으로 학습하여 실험을 진행하여야 하고 도출된 결과로 분석을 진행한다. 이를 위해 제안하는 시스템은 DNN 모델과 LSTM 모델, 상태유지 LSTM 모델 기반으로 데이터 예측 시스템을 설계하였다. 그리고 시스템 구현 후 데이터 셋과 시퀀스 길이의 비율에 따라 정확도 변경 실험을 진행하여 최적화된 데이터 셋과 시퀀스 길이의 비율을 제시한다.

3.2.2.1 DNN 모델 기반 시스템 설계

DNN 모델 기반 시스템을 구축하기 위해서는 DNN 모델의 은닉층 개수를 설정하기 위해 은닉층의 개수를 2개에서 5개로 변경해가며 각 은닉층의 개수마다 2회씩 총 8회의 실험을 진행하였고 실험의 결과는 다음과 같다. 그림 23은 은닉층을 2개로 설정한 DNN 모델의 성능 평가이다.



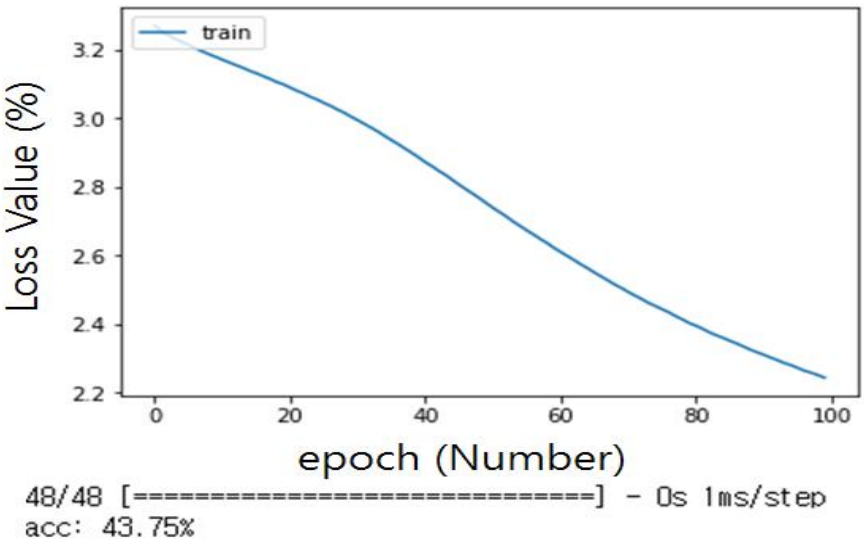
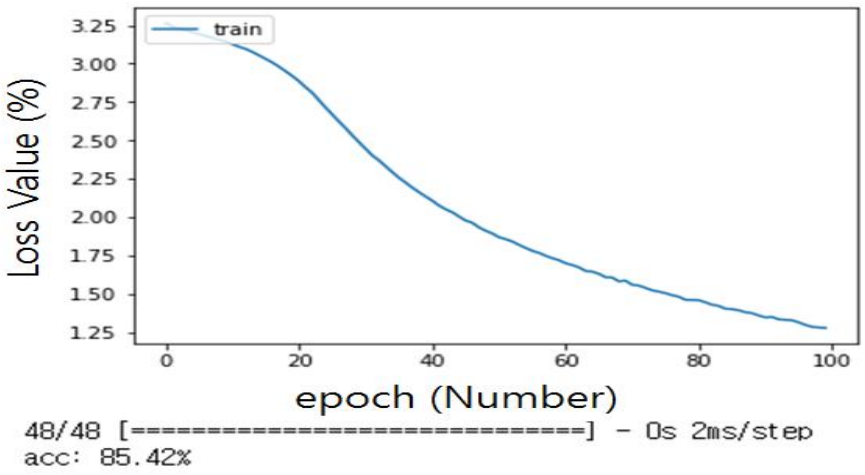


그림 23. DNN 모델 평가 - 은닉층 2개

DNN 모델의 은닉층을 2개로 설정하였고 반복 학습의 횟수는 100회, 데이터 셋은 52개의 비정형 데이터, 시퀀스의 길이는 4로 설정하였다. 해당 실험의 결과

는 35.42 퍼센트와 43.75 퍼센트의 정확도를 보였다. 그림 24는 은닉층을 3개로 설정한 DNN 모델의 성능 평가이다.



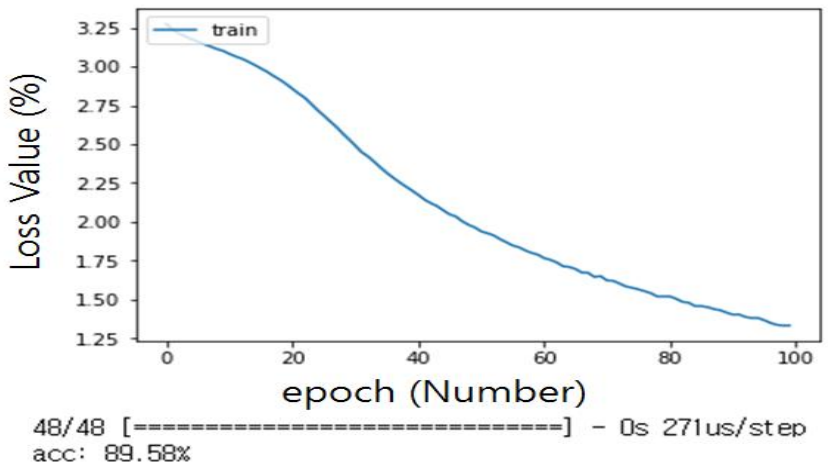


그림 24. DNN 모델 평가 - 은닉층 3개

DNN 모델의 은닉층을 3개로 설정하였고 반복 학습의 횟수는 100회, 데이터 셋은 52개의 비정형 데이터, 시퀀스의 길이는 4로 설정하였다. 해당 실험의 결과

는 85.42 퍼센트와 89.58 퍼센트의 정확도를 보였다. 그림 25는 은닉층을 4개로 설정한 DNN 모델의 성능 평가이다.

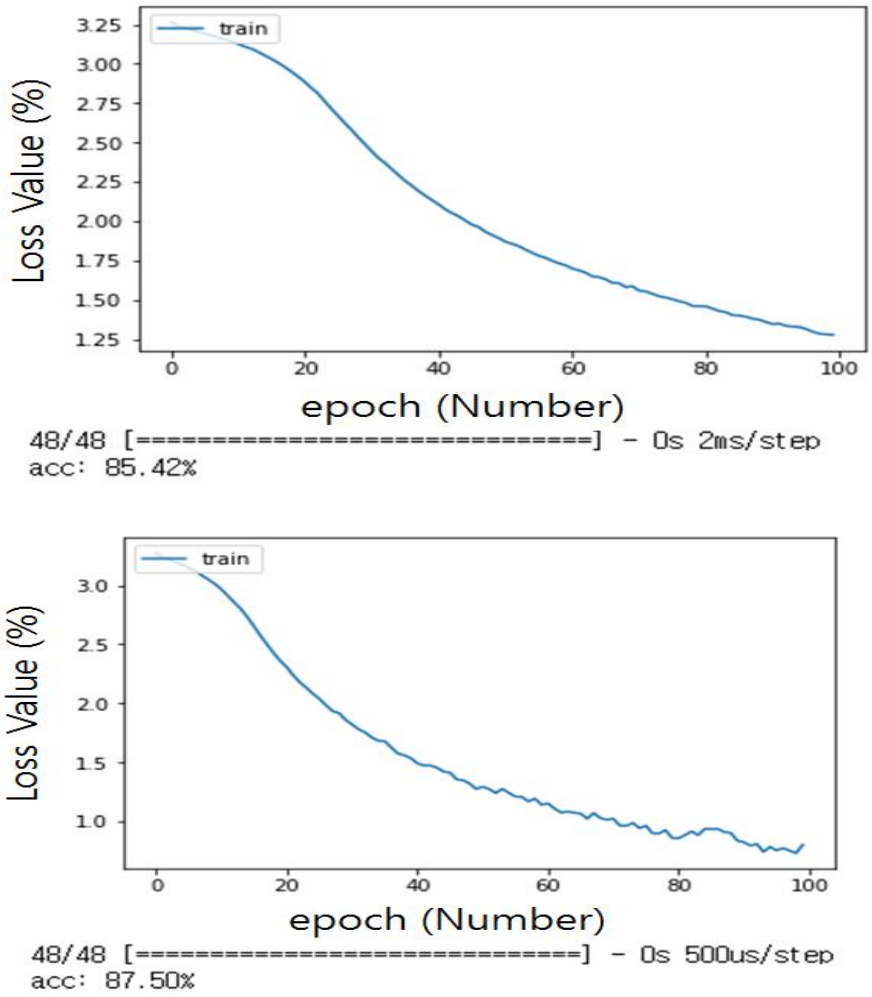


그림 25. DNN 모델 평가 - 은닉층 4개

DNN 모델의 은닉층을 4개로 설정하였고 반복 학습의 횟수는 100회, 데이터 셋은 52개의 비정형 데이터, 시퀀스의 길이는 4로 설정하였다. 해당 실험의 결과는 85.42 퍼센트와 87.50 퍼센트의 정확도를 보였다. 그림 26은 은닉층을 5개로

설정한 DNN 모델의 성능 평가이다.

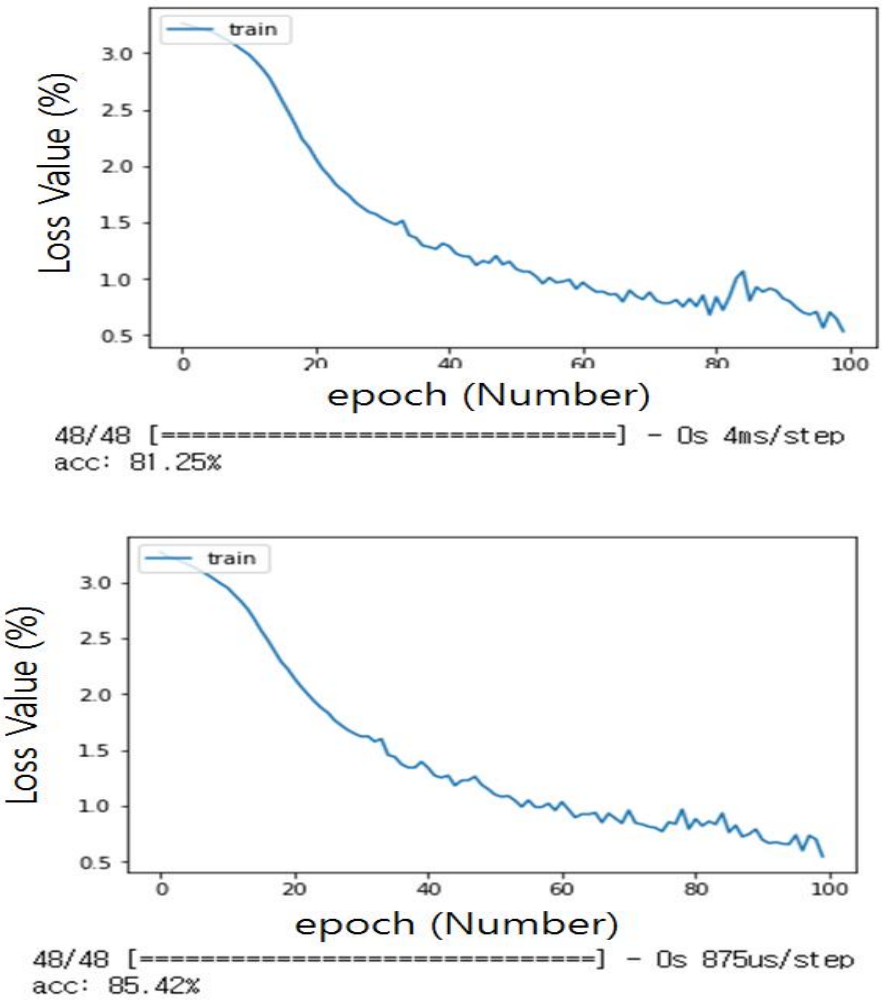


그림 26. DNN 모델 평가 - 은닉층 5개

DNN 모델의 은닉층을 5개로 설정하였고 반복 학습의 횟수는 100회, 데이터 셋은 52개의 비정형 데이터, 시퀀스의 길이는 4로 설정하였다. 해당 실험의 결과는 81.25 퍼센트와 85.42 퍼센트의 정확도를 보였다. 총 8회의 은닉층 변경 실험을 진행하였고 3개의 은닉층으로 학습을 진행하였을 때 정확도가 가장 높은 것

을 확인하였다. 이로 인해 본 논문에서 실험을 진행하기 위해 DNN 모델의 은닉층을 3개로 설정하였다.

DNN 모델 기반 데이터 예측 시스템의 은닉층 개수 설정 실험을 통해 은닉층을 3개로 설정하였을 때 가장 정확도가 높은 것으로 실험 결과가 도출되었고 배치사이즈 변경 실험을 진행하여 최적화된 배치사이즈를 도출하기 위해 실험을 진행한다. 표 1은 위 실험에서 설계한 DNN 모델 기반 데이터 예측 시스템이 배치사이즈를 변경해가며 실험을 진행하였고 이에 대한 결과를 표로 정리한 것이다. 그림 27은 배치사이즈 별 두 차례 도출된 정확도의 평균을 그래프로 나타낸 것이다. 실험을 위해 사용한 데이터 셋은 52개의 순차 데이터이며 시퀀스 길이 4, 속성 수 1, 반복 학습의 횟수는 100회로 설정하였다.

표 1. DNN 모델 기반 데이터 예측 시스템 배치사이즈 변경 실험

Batch Size	First Experiment Accuracy (%)	Second Experiment Accuracy (%)
1	79.17	83.33
2	87.50	83.33
3	79.17	79.17
4	91.67	95.83
5	93.75	97.92
6	81.25	85.42
7	89.58	93.75
8	89.58	85.42
9	68.75	72.92
10	89.58	85.42
11	60.42	64.58

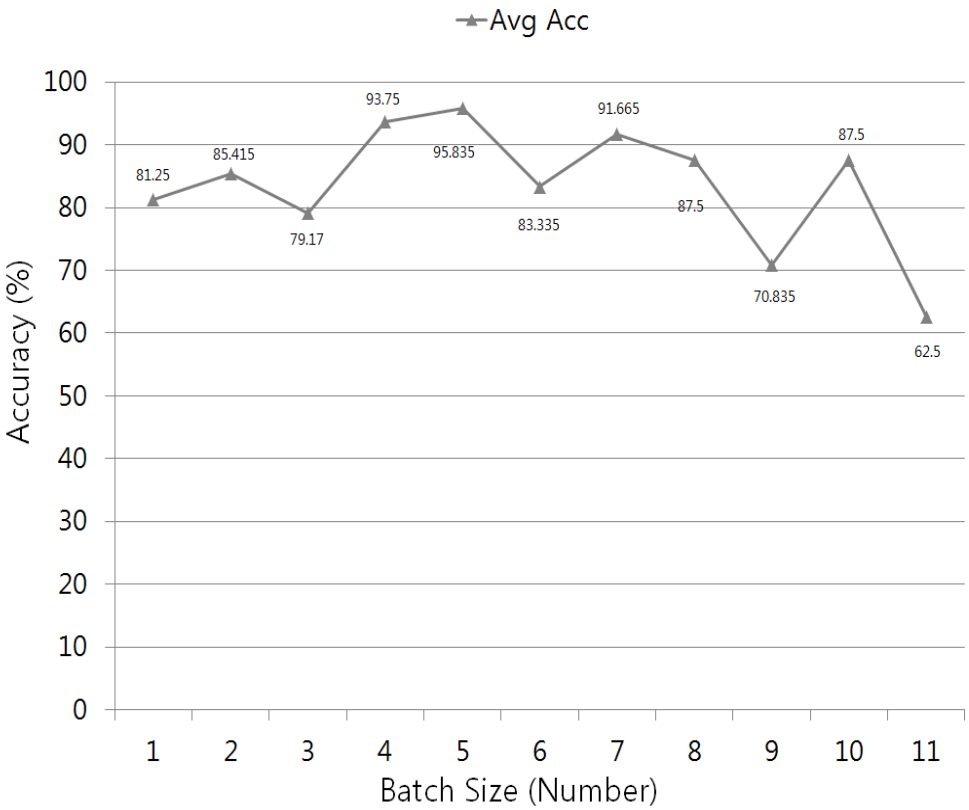


그림 27. 배치사이즈 별 평균 정확도 - DNN 모델 기반 시스템

배치사이즈를 5로 설정하였을 때 정확도가 가장 높았다. 또한 배치사이즈의 숫자가 작아질수록 학습의 시간이 길어지는데 반해 정확도의 상승이 일어나는 구간이 배치사이즈 4까지 인것을 확인하였다. 그리고 5부터 배치사이즈가 커질수록 정확도가 대부분의 구간에서 낮아지는 점을 확인하였다.

은닉층 개수 설정 실험과 배치사이즈 실험을 통해 얻은 결과를 기반으로 DNN 모델 기반 데이터 예측 시스템은 3개의 은닉층과 배치사이즈를 5로 설계한다. 그리고 반복 학습의 횟수는 100회로 설정하는데 이는 과적합 현상을 방지하며 반복 학습의 횟수가 많아질수록 시스템의 정확도가 100퍼센트에 도달하기 때문에 제안하는 시스템의 목적인 적은 데이터와 반복 학습의 횟수에도 높은 정확도가

도출되는 점을 확인하기 위함이다.

그림 28은 실험을 위해 설계한 은닉층 3개와 배치사이즈 5, 반복 학습의 횟수 100으로 설정된 DNN 모델 기반 데이터 예측 시스템의 구조도이다. 길이가 4 단위로 구성된 순차 데이터들을 입력받고 다음 순차의 데이터를 라벨값으로 부여하고 이러한 구조의 데이터들을 기반으로 학습을 진행하는 방식이다. 그리고 그림 29는 DNN 모델 기반 데이터 예측 시스템의 흐름도이다.

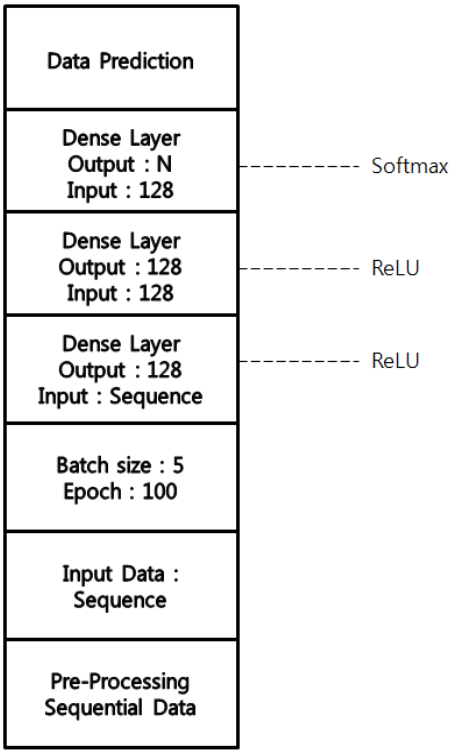


그림 28. DNN 모델 기반 데이터 예측 시스템 구조도

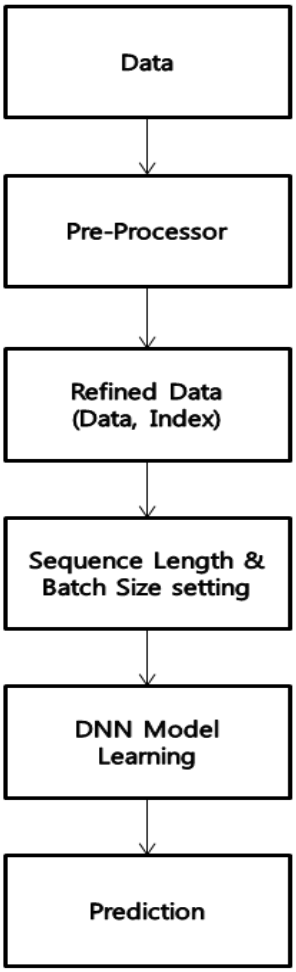


그림 29. DNN 모델 기반 데이터 예측 시스템 흐름도

DNN 모델 기반의 데이터 예측 시스템은 사용자가 데이터를 준비한 뒤 정제를 수행하는 것으로 시작한다. 그리고 정제가 완료된 순차 데이터로 구성된 데이터 셋을 DNN 모델에 적용하여 학습을 수행한다. 학습을 수행하는 첫 번째 은닉층과 두 번째 은닉층은 활성 함수로 ReLU를 활용한다. 그리고 세 번째 은닉층은 활성 함수로 Softmax를 활용한다. 해당 학습이 완료되면 사용자가 입력한 데이터들 다음에 나올 데이터를 예측하는 작업을 수행한다.

DNN 모델의 슈도코드는 다음과 같다. 사용자는 'code2index' 부분과 'index2code' 부분에 학습 데이터를 정의한다. 그리고 'total_data' 부분에 학습 데이터로 사용할 데이터를 입력한다. 'sequence_number'는 사용자가 학습을 위해 설정하게 될 시퀀스의 길이를 설정하는 부분이다. '# sequence length'로 주석 처리 한 부분들은 일관된 숫자로 표현되어야 한다. 그리고 '# start data'로 주석 처리 한 부분은 학습을 완료한 모델이 실험을 시작할 때 초기값을 설정하는 부분이다. 모델의 학습 방법을 변경하기 위해서는 주석 처리한 6 곳의 데이터를 수정해야하고 학습 데이터를 변경하기 위해서는 'code2index' 부분과 'index2code', 'seq' 부분을 수정해야한다.

```
# Dataset Definition
code2idx = {} idx2code = {} total_data = []

# Trainning setting
window_size = sequence_number # sequence length
trainning_data = dataset[:,0:sequence_number] # sequence length
result_data = dataset[:,sequence_number]   # sequence length
reshape(total_sequence-sequence_number, sequence_number, 1)
# sequence length

#Model Definition
model = Sequential()
model.add(Dense(128, input_dim= sequence_number, activation='relu'))
# sequence length
model.add(Dense(128, activation='relu'))
model.add(Dense(one_hot_vec_size, activation='softmax'))

# Data Prediction setting
Prediction_start = [] # start data
reshape(total_data, (1, sequence_number, 1)) # sequence length
```

3.2.2.2 기본 LSTM 모델 기반 시스템 설계

기본 LSTM 모델 기반 데이터 예측 시스템의 배치사이즈를 변경해가며 실험을 진행하였다. 이에 대한 결과는 표 2와 같다. 그리고 그림 30은 배치사이즈 별 두 차례 도출된 정확도의 평균을 그래프로 나타낸 것이다. 실험을 위해 사용한 데이터 셋은 52개의 순차 데이터이며 시퀀스 길이 4, 속성 수 1, 반복 학습의 횟수는 100회로 설정하였다.

표 2. 기본 LSTM 모델 기반 데이터 예측 시스템 배치사이즈 변경 실험

Batch Size	First Experiment Accuracy (%)	Second Experiment Accuracy (%)
1	100	100
2	100	100
3	100	100
4	100	100
5	100	100
6	85.42	85.42
7	97.92	97.92
8	97.92	97.92
9	93.75	97.92
10	97.92	97.92
11	97.92	97.92

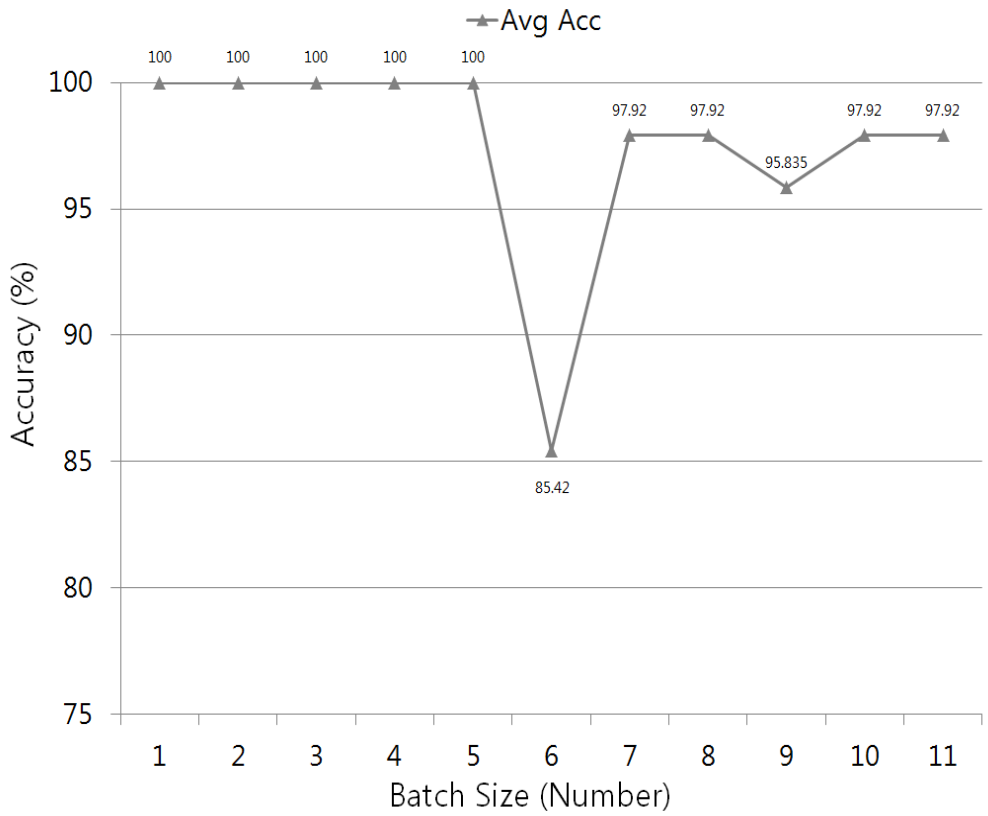


그림 30. 배치사이즈 별 평균 정확도 - 기본 LSTM 모델 기반 시스템

기본 LSTM 모델 기반 데이터 예측 시스템은 배치사이즈 1부터 5까지 정확도가 100 퍼센트로 가장 높았고 배치사이즈 6부터 11까지는 정확도가 약간 낮아졌다. 그리고 배치사이즈 1부터 3까지는 배치사이즈 4와 5로 설정한 것에 비해 시스템이 학습에 소요하는 시간이 상대적으로 길었다. 이로 인해 배치사이즈를 4와 5로 설정하는 것이 효율적이다.

그림 31은 데이터 셋 52, 시퀀스의 길이 4, 속성 수 1, 배치사이즈 5, 반복 학습의 횟수 100회로 설정된 기본 LSTM 모델 기반 데이터 예측 시스템의 구조도이다. 길이가 4 단위로 구성된 순차 데이터들을 입력받고 다음 순차의 데이터를 라벨값으로 부여하고 이러한 구조의 데이터들을 기반으로 학습을 진행하는 방식이다. 그림 32는 기본 LSTM 모델 기반 데이터 예측 시스템의 흐름도이다.

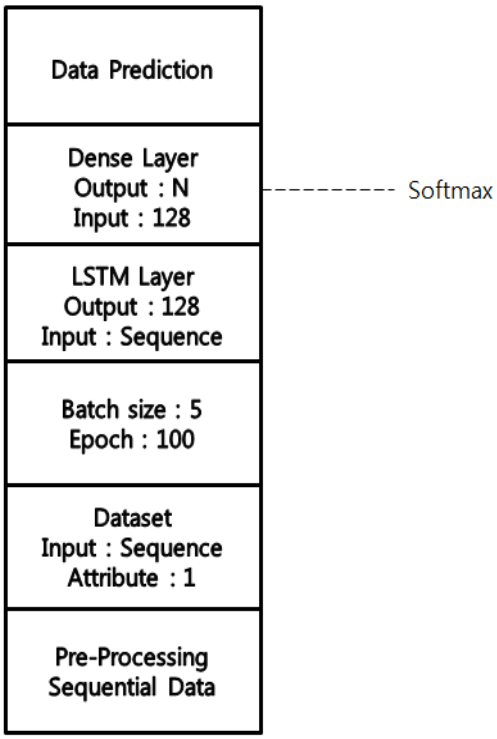


그림 31. 기본 LSTM 모델 기반 데이터 예측 시스템의 구조도

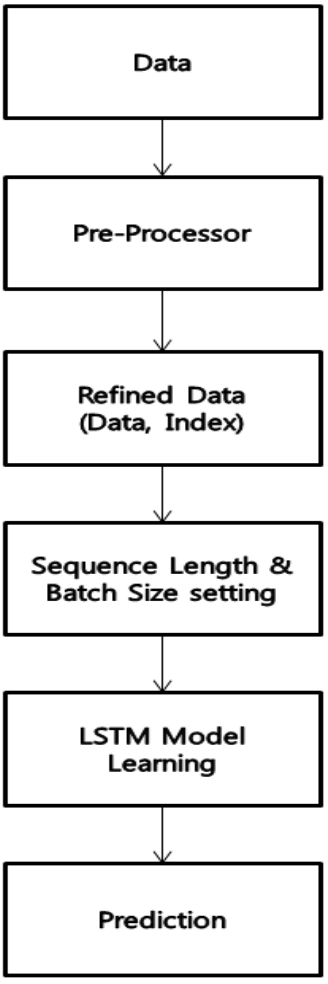


그림 32. 기본 LSTM 모델 기반 데이터 예측 시스템의 흐름도

기본 LSTM 모델 기반 데이터 예측 시스템의 흐름은 사용자가 데이터를 준비한 뒤 정제를 수행하는 것으로 시작한다. 그리고 정제가 완료된 순차 데이터로 구성된 데이터 셋을 딥 러닝 모델에 적용하여 학습을 수행한다. 학습을 수행할 때에는 DNN 모델의 세 번째 은닉층에서 활용한 활성 함수 Softmax를 활용한다. 해당 학습이 완료되면 사용자가 입력한 데이터들 다음에 나올 데이터를 예측하는 작업을 수행한다.

기본 LSTM 모델의 슈도코드는 다음과 같다. 사용자는 'code2index' 부분과 'index2code' 부분에 학습 데이터를 정의한다. 그리고 'seq' 부분에 학습 데이터로 사용할 데이터를 입력한다. 'sequence_number'는 사용자가 학습을 위해 설정하게 될 시퀀스의 길이를 설정하는 부분이다. '# sequence length'로 주석처리 한 부분들은 일관된 숫자로 표현되어야 한다. 그리고 '# start data'로 주석처리 한 부분은 학습을 완료한 모델이 실험을 시작할 때 초기값을 설정하는 부분이다. 모델의 학습 방법을 변경하기 위해서는 주석 처리한 6 곳의 데이터를 수정해야하고 학습 데이터를 변경하기 위해서는 'code2index' 부분과 'index2code', 'seq' 부분을 수정해야한다. DNN 모델과 다른 점으로는 데이터 셋으로 학습을 진행할 때 사용자가 입력한 시퀀스 값의 개수가 'total_sequence'이고 해당 값으로 설정을 변경해줘야 한다.

```
# Dataset Definition
code2idx = {} idx2code = {} total_data = []

# Trainning setting
window_size = sequence_number # sequence length
trainning_data = dataset[:,0:sequence_number] # sequence length
result_data = dataset[:,sequence_number]    # sequence length
reshape(total_sequence-sequence_number, sequence_number, 1)
# sequence length

#Model Definition
model = Sequential()
model.add(LSTM(128, input_shape = (sequence_number, 1)))
# sequence length
model.add(Dense(one_hot_vec_size, activation='softmax'))

# Data Prediction setting
Prediction_start = [] # start data
reshape(total_data, (1, sequence_number, 1)) # sequence length
```

3.2.2.3 상태유지 LSTM 모델 기반 시스템 설계

상태유지 LSTM 모델 기반 데이터 예측 시스템은 상태유지 기능으로 인해 가중치가 업데이트 되고 초기화가 되기 때문에 배치사이즈 자체는 변경할 수 없다. 이로 인해 배치사이즈처럼 가중치를 초기화하는 사이즈를 변경해가며 실험을 진행하였고 이에 대한 결과는 표 3과 같다. 그리고 그림 33은 가중치 초기화 사이즈 별 두 차례 도출된 정확도의 평균을 그래프로 나타낸 것이다.실험을 위해 사용한 데이터셋은 52개의 순차 데이터이며 시퀀스 길이 4, 속성 수 1, 반복 학습의 횟수는 100회, 상태유지 기능 On으로 설정하였다.

표 3. 상태유지 LSTM 모델 기반 데이터 예측 시스템 가중치 초기화 수치 실험

Initialize Size	First Experiment Accuracy (%)	Second Experiment Accuracy (%)
1	58.33	35.42
2	62.50	43.75
3	83.33	72.92
4	91.67	100
5	100	87.50
6	100	100
7	100	100
8	100	100
9	100	100
10	100	100
11	100	100

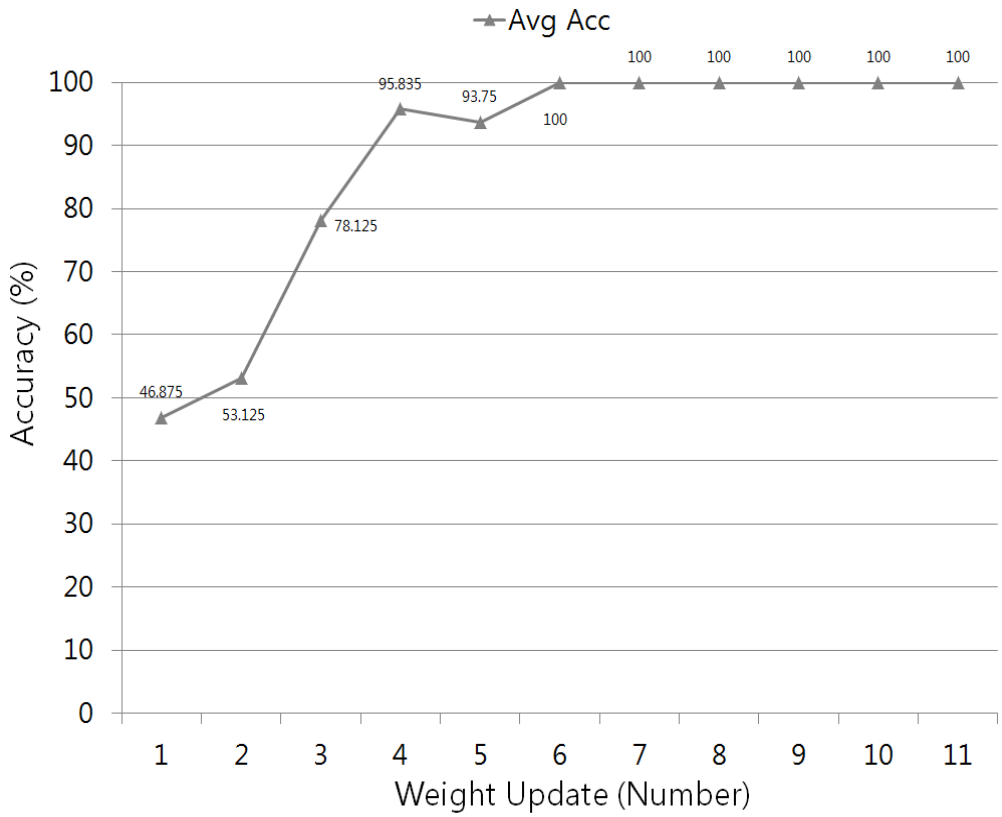


그림 33. 가중치 초기화 사이즈 별 평균 정확도 - 상태유지 LSTM 모델 기반 시스템

상태유지 LSTM 모델 기반 데이터 예측 시스템은 가중치 초기화 사이즈 1부터 5까지 정확도가 지속적으로 상승하였고 6부터는 정확도가 100 퍼센트였다. 가중치 초기화 수치가 늘어날수록 상태유지 LSTM 모델 기반 데이터 예측 시스템은 학습의 횟수가 느려졌다. 이에 따라 시스템의 속도가 급격히 느려지게 된다. 이러한 결과를 바탕으로 상태유지 LSTM 모델 기반 데이터 예측 시스템은 가중치 초기화 사이즈를 6으로 설정하는 것이 효율적이다.

그림 34는 데이터 셋 52, 시퀀스의 길이 4, 속성 수 1, 가중치 초기화 사이즈 6, 반복 학습의 횟수 100회로 설정된 상태유지 LSTM 모델 기반 데이터 예측 시스템의 구조도이다. 길이가 4 단위로 구성된 순차 데이터들을 입력받고 다음 순차의 데이터를 라벨값으로 부여하고 학습 상태가 기억되며 다음 학습 때 영향을 미친다. 상태유지의 여부에 따라 결과가 달라지는데 해당 실험에서의 상태유지는 새로운 학습을 시작할 때 가중치를 초기화하여 진행하였다. 그림 35는 상태유지 LSTM 모델 기반 데이터 예측 시스템의 흐름도이다.

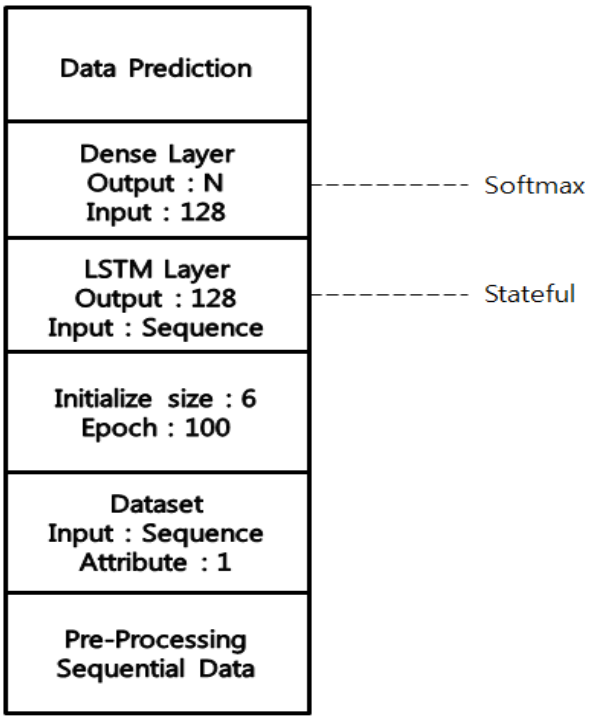


그림 34. 상태유지 LSTM 모델 기반 데이터 예측 시스템 구조도

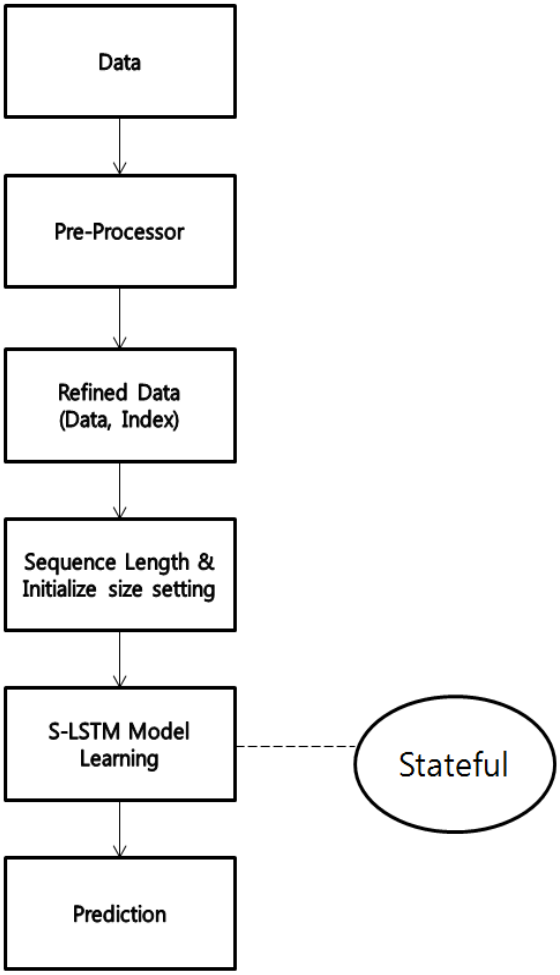


그림 35. 상태유지 LSTM 모델 기반 데이터 예측 시스템 흐름도

상태유지 LSTM 모델 기반 데이터 예측 시스템의 흐름은 사용자가 데이터를 준비한 뒤 정제를 수행하는 것으로 시작한다. 그리고 정제가 완료된 데이터 셋을 딥 러닝 모델에 적용하여 학습을 수행한다. 학습을 수행할 때에는 DNN 모델 세 번째 은닉층과 기본 LSTM 모델에서 활용한 활성 함수로 Softmax를 활용한다. 또한 학습이 진행될 때 Stateful 기능을 통해 이전 학습이 다음 학습에 영향을 준다. 해당 학습이 완료되면 사용자가 입력한 데이터들 다음에 나올 데이터를 예측하는 작업을 수행한다.

상태유지 LSTM 모델의 알고리즘 슈도코드는 다음과 같다. 기본 설정은 기본 LSTM 모델과 같고 기본 LSTM 모델과 다른 점은 모델을 정의할 때 상태유지 기능을 위해 'stateful=True' 설정을 진행한다.

```
# Dataset Definition
code2idx = {}  idx2code = {} total_data = []

# Trainning setting
window_size = sequence_number # sequence length
trainning_data = dataset[:,0:sequence_number] # sequence length
result_data = dataset[:,sequence_number]   # sequence length
reshape(total_sequence-sequence_number, sequence_number, 1)
# sequence length

#Model Definition
model = Sequential()
model.add(LSTM(128, batch_input_shape = (1, sequence_number, 1),
stateful=True)) # sequence length
model.add(Dense(one_hot_vec_size, activation='softmax'))

# Data Prediction setting
Prediction_start = [] # start data
reshape(total_data, (1, sequence_number, 1)) # sequence length
```

DNN 모델 기반 시스템과 기본 LSTM 모델 기반 시스템을 구축할 때 배치사이즈가 미치는 영향과 상태유지 LSTM 모델 기반 시스템을 구축할 때 가중치 초기화 사이즈가 미치는 영향을 정리한 표 1, 2, 3을 통해 최적화된 수치를 도출하였다.

DNN 모델 기반 시스템과 기본 LSTM 모델 기반 시스템은 상태유지 LSTM 모델 기반 시스템에 비해 학습 속도가 빨랐다. 정확도는 평균 정확도가 97.53 퍼센트인 기본 LSTM 모델 기반 시스템과 평균 정확도가 87.97 퍼센트의 상태유지 LSTM 모델 기반 시스템, 평균 정확도가 83.52 퍼센트의 DNN 모델 기반 시스템 순으로 확인하였다.

이러한 모델들의 특징을 분석하였을 때 데이터 예측 시스템에 적합한 모델은 LSTM 모델이다. 데이터 예측 시스템이 사용자에게 제공해야할 가장 중요한 기능은 정확도와 속도이다.

기본 LSTM 모델 기반 시스템은 다른 모델 기반 시스템에 비해 배치사이즈를 변경하여도 정확도가 가장 높았으며 학습의 속도도 DNN 모델 기반 시스템과 차이가 근소하였다.

이러한 실험 결과를 바탕으로 4장에서는 기본 LSTM 모델 기반 시스템을 구현하고 기존의 DNN 모델 기반 시스템과 제안하는 시스템의 차이를 분석하기 위해 실험을 진행한다. 그리고 실험 결과를 바탕으로 제안하는 시스템의 효율성을 명시한다.

Ⅳ. 데이터 예측 시스템 구현 및 실험

이 장에서는 DNN 모델 기반 시스템과 기본 LSTM 모델 기반 시스템, 상태유지 LSTM 모델 기반 시스템을 구현하고 가장 효율이 좋은 시스템을 선정한다. 그리고 실험을 통해 기존 시스템과 제안하는 시스템의 비교 분석을 진행하여 제안하는 시스템의 효율성을 검증한다.

4.1 시스템 구현

이 절에서는 시스템 구현에 대해 서술한다. 전처리기 구현에 앞서 전처리기가 수행해야할 기능으로는 사용자가 입력한 데이터를 정렬하여 데이터 셋 형태로 가공한 뒤 이를 출력한다. 그림 36은 전처리기가 시작됐을 때 초기화면이고 그림 37은 사용자가 입력한 데이터를 전처리기가 데이터 셋을 가공하여 출력한 화면이다.

그림 36. 전처리기 시작 화면

분석을 원하는 데이터를 입력하시오
a b c d e f g h i j k m l n o p q r s t u v w x y z

code to index
'a':0,'b':1,'c':2,'d':3,'e':4,'f':5,'g':6,'h':7,'i':8,'j':9,'k':10,'m':11,'l':12,
'n':13,'o':14,'p':15,'q':16,'r':17,'s':18,'t':19,'u':20,'v':21,'w':22,'x':23,'y':24,'z':25

index to code
0:'a',1:'b',2:'c',3:'d',4:'e',5:'f',6:'g',7:'h',8:'i',9:'j',10:'k',11:'m',12:'l',
13:'n',14:'o',15:'p',16:'q',17:'r',18:'s',19:'t',20:'u',21:'v',22:'w',23:'x',24:'y',25:'z'

그림 37. 전처리기 출력 화면 1

그림 38은 영어로 작성된 시를 분석한 화면이다.

```
분석을 원하는 데이터를 입력하시오
My Heart is like a singing bird Whose nest is in a watered shoot

code to index
'My':0,'Heart':1,'is':2,'like':3,'a':4,'singing':5,'bird':6,
'Whose':7,'nest':8,'is':9,'in':10,'a':11,'watered':12,'shoot':13

index to code
0:'My',1:'Heart',2:'is',3:'like',4:'a',5:'singing',6:'bird',
7:'Whose',8:'nest',9:'is',10:'in',11:'a',12:'watered',13:'shoot'
```

그림 38. 전처리기 출력 화면 2

그림 39는 한글로 작성된 시를 분석한 화면이다.

```
분석을 원하는 데이터를 입력하시오
내 마음은 물가의 가지에 둥지를 튼 한 마리 노래하는 새입니다

code to index
'내':0,'마음은':1,'물가의':2,'가지에':3,'둥지를':4,
'튼':5,'한':6,'마리':7,'노래하는':8,'새입니다':9

index to code
0:'내',1:'마음은',2:'물가의',3:'가지에',4:'둥지를',
5:'튼',6:'한',7:'마리',8:'노래하는',9:'새입니다'
```

그림 39. 전처리기 출력 화면 3

전처리기는 형태소 분석기와 같은 기능을 수행한다. ‘ ’나 ‘ ’와 같이 공백이 있을 경우 다음 데이터로 인식한다. 또한 줄바꾸기 역시 다음 데이터로 인식하게끔 구현하였다. 이와 같이 데이터에 인덱스를 부여하여 ‘데이터 : 인덱스’ 구조와 ‘인덱스 : 데이터’ 구조로 전처리를 수행한다. 그리고 전처리기에서 정제된 데이터는 그림 40과 같은 형태로 딥 러닝 모델의 학습에 활용된다.

```
(39, 14)
[[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13]
 [ 1  2  3  4  5  6  7  8  9 10 11 12 13 14]
 [ 2  3  4  5  6  7  8  9 10 11 12 13 14 15]
 [ 3  4  5  6  7  8  9 10 11 12 13 14 15 16]
 [ 4  5  6  7  8  9 10 11 12 13 14 15 16 17]
 [ 5  6  7  8  9 10 11 12 13 14 15 16 17 18]
 [ 6  7  8  9 10 11 12 13 14 15 16 17 18 19]
 [ 7  8  9 10 11 12 13 14 15 16 17 18 19 20]
 [ 8  9 10 11 12 13 14 15 16 17 18 19 20 21]
 [ 9 10 11 12 13 14 15 16 17 18 19 20 21 22]
 [10 11 12 13 14 15 16 17 18 19 20 21 22 23]
 [11 12 13 14 15 16 17 18 19 20 21 22 23 24]
 [12 13 14 15 16 17 18 19 20 21 22 23 24 25]
 [13 14 15 16 17 18 19 20 21 22 23 24 25  0]
 [14 15 16 17 18 19 20 21 22 23 24 25  0  1]
 [15 16 17 18 19 20 21 22 23 24 25  0  1  2]
 [16 17 18 19 20 21 22 23 24 25  0  1  2  3]
 [17 18 19 20 21 22 23 24 25  0  1  2  3  4]
 [18 19 20 21 22 23 24 25  0  1  2  3  4  5]
 [19 20 21 22 23 24 25  0  1  2  3  4  5  6]
 [20 21 22 23 24 25  0  1  2  3  4  5  6  7]
 [21 22 23 24 25  0  1  2  3  4  5  6  7  8]
 [22 23 24 25  0  1  2  3  4  5  6  7  8  9]
 [23 24 25  0  1  2  3  4  5  6  7  8  9 10]
 [24 25  0  1  2  3  4  5  6  7  8  9 10 11]
 [25  0  1  2  3  4  5  6  7  8  9 10 11 12]
 [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13]
 [ 1  2  3  4  5  6  7  8  9 10 11 12 13 14]
 [ 2  3  4  5  6  7  8  9 10 11 12 13 14 15]
 [ 3  4  5  6  7  8  9 10 11 12 13 14 15 16]
 [ 4  5  6  7  8  9 10 11 12 13 14 15 16 17]
 [ 5  6  7  8  9 10 11 12 13 14 15 16 17 18]
 [ 6  7  8  9 10 11 12 13 14 15 16 17 18 19]
 [ 7  8  9 10 11 12 13 14 15 16 17 18 19 20]
 [ 8  9 10 11 12 13 14 15 16 17 18 19 20 21]
 [ 9 10 11 12 13 14 15 16 17 18 19 20 21 22]
 [10 11 12 13 14 15 16 17 18 19 20 21 22 23]
 [11 12 13 14 15 16 17 18 19 20 21 22 23 24]
 [12 13 14 15 16 17 18 19 20 21 22 23 24 25]]
```

그림 40. 학습 데이터의 구조

학습은 순차적으로 데이터를 입력받고 이에 대한 라벨값을 부여하는 방식으로 이루어진다. 그림 41은 학습이 진행되는 동안 손실값과 정확도를 도출하는 과정이다.

```
Epoch 80/100
 - 0s - loss: 0.6949 - acc: 0.8095
Epoch 81/100
 - 0s - loss: 0.6803 - acc: 0.7619
Epoch 82/100
 - 0s - loss: 0.7671 - acc: 0.5476
Epoch 83/100
 - 0s - loss: 1.0452 - acc: 0.5238
Epoch 84/100
 - 0s - loss: 0.8223 - acc: 0.6429
Epoch 85/100
 - 0s - loss: 0.8392 - acc: 0.6190
Epoch 86/100
 - 0s - loss: 0.6328 - acc: 0.8571
Epoch 87/100
 - 0s - loss: 0.5462 - acc: 0.8810
Epoch 88/100
 - 0s - loss: 0.5523 - acc: 0.9286
Epoch 89/100
 - 0s - loss: 0.5370 - acc: 0.8810
Epoch 90/100
 - 0s - loss: 0.5153 - acc: 0.9524
Epoch 91/100
 - 0s - loss: 0.4608 - acc: 0.9762
Epoch 92/100
 - 0s - loss: 0.4567 - acc: 0.9048
Epoch 93/100
 - 0s - loss: 0.4442 - acc: 1.0000
Epoch 94/100
 - 0s - loss: 0.4246 - acc: 1.0000
Epoch 95/100
 - 0s - loss: 0.4214 - acc: 0.9762
Epoch 96/100
 - 0s - loss: 0.4141 - acc: 0.9286
Epoch 97/100
 - 0s - loss: 0.4095 - acc: 0.9762
Epoch 98/100
 - 0s - loss: 0.4119 - acc: 0.9762
Epoch 99/100
 - 0s - loss: 0.3986 - acc: 0.9524
Epoch 100/100
 - 0s - loss: 0.3790 - acc: 0.9762
```

그림 41. 학습 과정

학습이 완료되면 그림 42와 같이 학습 과정을 그래프로 나타낸다. 또한 그래프 아래에는 사용자가 입력한 데이터 뒤에 나올 데이터를 예측하고 이에 대한 평가를 진행한 것을 나타낸다.

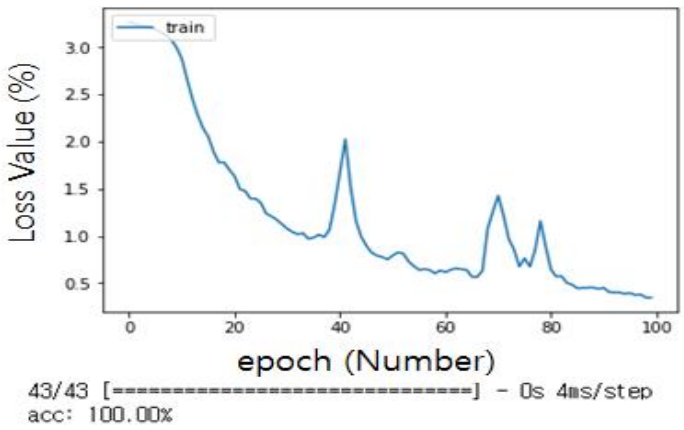


그림 42. 학습 결과 및 예측 수행

4.2 정확도 비교 실험

이 절에서는 DNN 모델 기반 시스템과 기본 LSTM 모델 기반 시스템, 상태유지 LSTM 모델 기반 시스템을 구현하고 데이터 셋과 시퀀스 길이의 비율을 변경해가며 실험을 진행한다. 그리고 구현한 시스템들을 평가하고 데이터 예측 시스템에 가장 적합한 시스템을 선정한다. 구현 및 실험을 진행할 PC는 CPU Core (TM) i5-4690 3.50GHz, RAM 8G, OS Windows 7 64bit를 사용하였다. 사용한 툴은 전처리기를 개발하기 위해 사용한 Eclipse와 딥 러닝 모델을 구현하기 위해 Anaconda3, Jupyter notebook을 활용하였다. 사용한 프로그래밍 언어로는 전처리기를 개발하기 위해 Java 1.8.0_181, 딥 러닝 모델을 구현하기 위해 Python 3.54 버전을 사용하였다.

4.2.1 평가 기준

각 모델 기반으로 구현한 시스템들의 데이터 셋과 시퀀스 길이를 변경해가며 실험을 진행하고 결과로 나온 정확도를 분석하여 3개 모델의 평균 정확도와 학습 및 데이터 예측에 소요되는 시간을 비교한다. 그리고 해당 비교 분석을 통해 가장 적합한 시스템을 선정한다. 실험은 28개와 52개 데이터로 이루어진 두 개의 데이터 셋을 삽입하여 학습을 진행하고 이에 대한 데이터 예측을 수행한다. 데이터 셋 28을 통해 학습을 진행할 때는 시퀀스의 길이를 2에서부터 14까지 늘려가며 실험을 진행한다. 데이터 셋 52를 통해 학습을 진행할 때는 시퀀스의 길이를 2에서부터 26까지 늘려가며 실험을 진행한다. 각 모델 기반 시스템 별로 38회의 실험을 진행한 뒤 이에 대한 결과를 바탕으로 데이터 셋과 시퀀스 길이의 최적화된 비율을 제시한다.

4.2.2 DNN 모델 기반 시스템 실험

DNN 모델은 은닉계층을 3개로 설정하고 배치사이즈는 5로 설정하였다. 그리고 반복 학습의 횟수는 100회로 설정하였고 데이터 셋이 28개일 때 실험 13회, 데이터 셋이 52개일 때 실험 25회를 진행하였다. 표 4은 DNN 모델 기반 데이터 예측 시스템이 데이터 셋 28로 실험을 진행한 결과이고 표 5는 데이터 셋 52로 실험을 진행한 결과이다.

표 4. 시퀀스 길이 변경 실험 정리 1 (DNN 모델)

Dataset	Sequence Length (number)	First Experiment Accuracy (%)	Second Experiment Accuracy (%)
28개	2	73.08	73.08
28개	3	96.00	96.00
28개	4	95.83	95.83
28개	5	95.65	95.65
28개	6	77.27	77.27
28개	7	85.71	85.71
28개	8	95.00	95.00
28개	9	94.74	94.74
28개	10	94.44	94.44
28개	11	82.35	82.35
28개	12	100	100
28개	13	93.33	93.33
28개	14	100	100

데이터 셋 28로 학습과 실험을 진행하였을 경우 평균 정확도는 91.03 퍼센트이다.

표 5. 시퀀스 길이 변경 실험 정리 2 (DNN 모델)

Dataset	Sequence Length (number)	First Experiment Accuracy (%)	Second Experiment Accuracy (%)
52개	2	78.00	82.00
52개	3	81.63	73.47
52개	4	93.75	89.58
52개	5	80.85	80.85
52개	6	82.61	78.26
52개	7	84.44	80.00
52개	8	86.36	81.82
52개	9	83.72	88.37
52개	10	73.81	78.57
52개	11	87.80	82.93
52개	12	87.50	87.50
52개	13	84.62	82.05
52개	14	78.95	84.21
52개	15	75.68	81.08
52개	16	69.44	69.44
52개	17	97.14	91.43
52개	18	97.06	97.06
52개	19	90.91	90.91
52개	20	90.62	90.62
52개	21	90.32	90.32
52개	22	83.33	45.83
52개	23	89.66	89.66
52개	24	96.43	96.43
52개	25	96.30	96.30
52개	26	100	100

데이터 셋 52로 학습과 실험을 진행하였을 경우 평균 정확도는 84.64 퍼센트이다. 총 38회 도출된 정확도의 평균은 87.83 퍼센트이다.

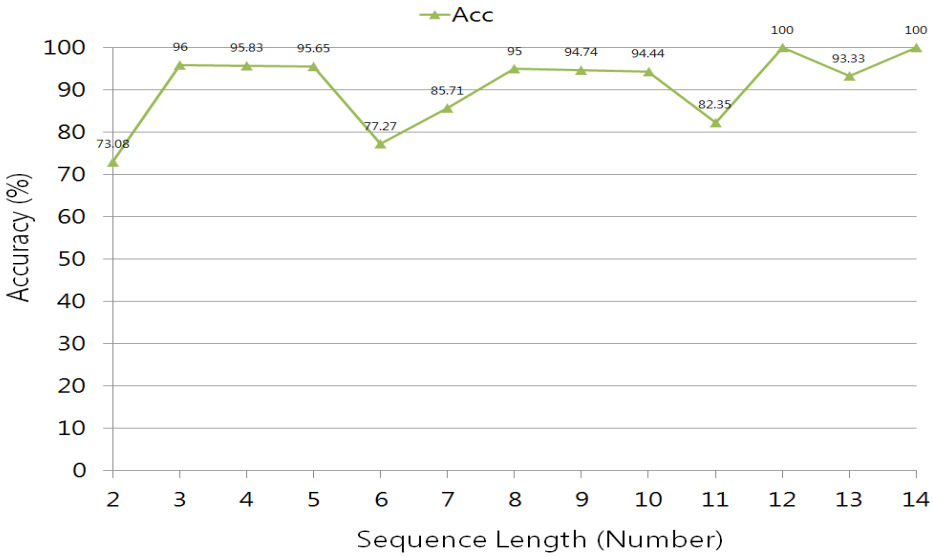


그림 43. 시퀀스 길이 변경 실험 정리 1 (DNN 모델)

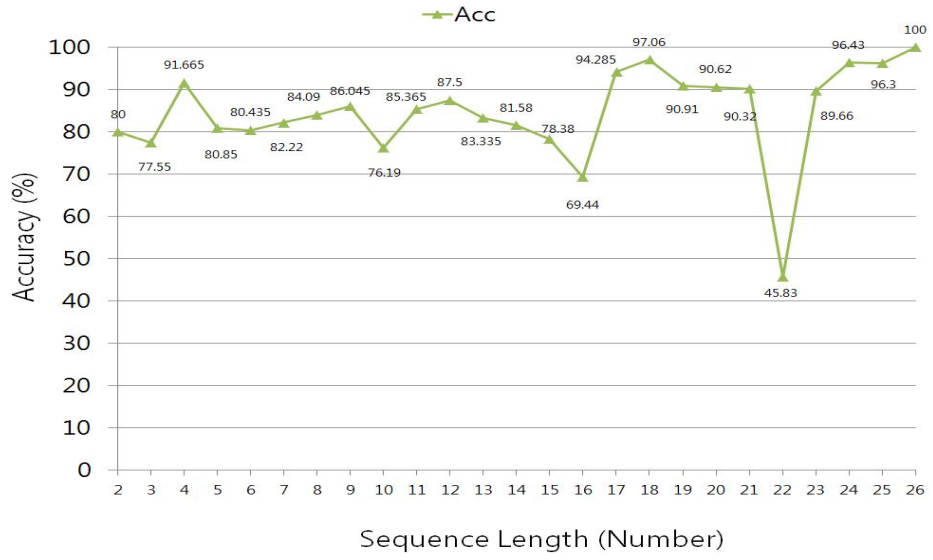


그림 44. 시퀀스 길이 변경 실험 정리 2 (DNN 모델)

4.2.3 기본 LSTM 모델 기반 시스템 실험

기본 LSTM 모델 기반 데이터 예측 시스템은 데이터 셋이 28개일 때 실험 13회, 데이터 셋이 52개일 때 실험 25회를 진행하였다. 표 6은 기본 LSTM 모델 기반 데이터 예측 시스템이 데이터 셋 28로 실험을 진행한 결과이고 표 7은 데이터 셋 52로 실험을 진행한 결과이다.

표 6. 시퀀스 길이 변경 실험 정리 3 (기본 LSTM 모델)

Dataset	Sequence Length (number)	First Experiment Accuracy (%)	Second Experiment Accuracy (%)
28개	2	88.46	88.46
28개	3	100	100
28개	4	100	100
28개	5	100	100
28개	6	100	100
28개	7	100	100
28개	8	100	100
28개	9	100	100
28개	10	100	100
28개	11	100	100
28개	12	100	100
28개	13	100	100
28개	14	100	100

데이터 셋 28로 학습과 실험을 진행하였을 경우 평균 정확도는 99.11 퍼센트이다.

표 7. 시퀀스 길이 변경 실험 정리 4 (기본 LSTM 모델)

Dataset	Sequence Length (number)	First Experiment Accuracy (%)	Second Experiment Accuracy (%)
52개	2	90.00	82.00
52개	3	97.96	97.96
52개	4	97.92	97.92
52개	5	97.87	97.87
52개	6	97.83	97.83
52개	7	97.78	97.78
52개	8	100	100
52개	9	100	100
52개	10	97.62	97.62
52개	11	100	100
52개	12	100	100
52개	13	100	100
52개	14	94.74	89.47
52개	15	97.30	100
52개	16	88.89	91.67
52개	17	100	100
52개	18	91.18	97.06
52개	19	93.94	93.94
52개	20	90.62	96.88
52개	21	100	96.77
52개	22	96.67	100
52개	23	96.55	100
52개	24	100	96.43
52개	25	100	92.59
52개	26	92.31	96.15

데이터 셋 52로 학습과 실험을 진행하였을 경우 평균 정확도는 84.64 퍼센트이다. 총 38회 도출된 정확도의 평균은 91.87 퍼센트이다.

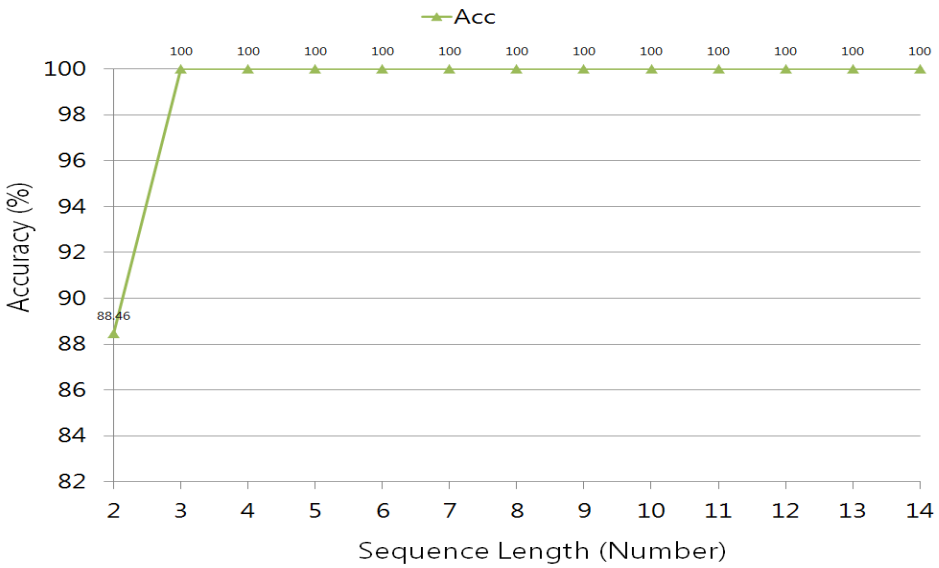


그림 45. 시퀀스 길이 변경 실험 정리 3 (기본 LSTM 모델)

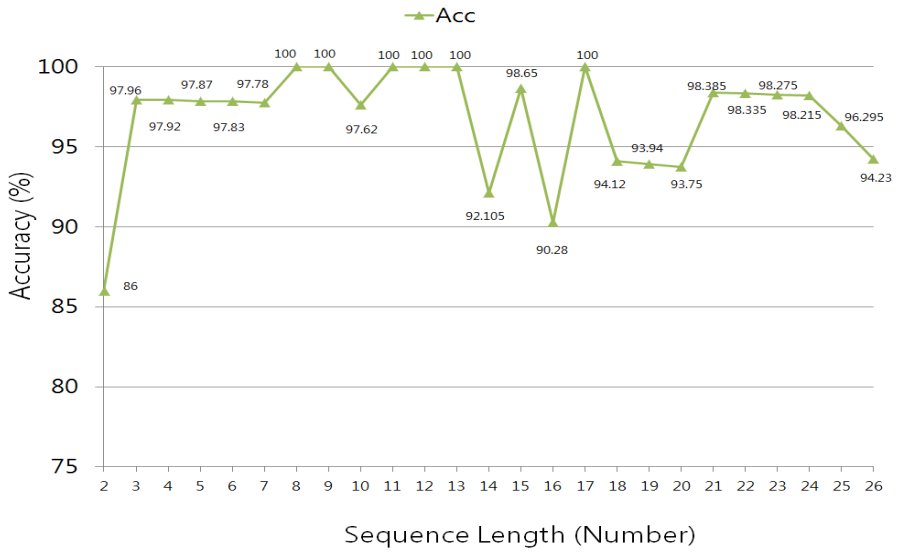


그림 46. 시퀀스 길이 변경 실험 정리 4 (기본 LSTM 모델)

4.2.4 상태유지 LSTM 모델 실험

상태유지 LSTM 모델 기반 데이터 예측 시스템은 데이터 셋이 28개일 때 실험 13회, 데이터 셋이 52개일 때 실험 25회를 진행하였다. 표 8은 상태유지 LSTM 모델 기반 데이터 예측 시스템이 데이터 셋 28로 실험을 진행한 결과이고 표 9는 데이터 셋 52로 실험을 진행한 결과이다.

표 8. 시퀀스 길이 변경 실험 정리 5 (상태유지 LSTM 모델)

Dataset	Sequence Length (number)	First Experiment Accuracy (%)	Second Experiment Accuracy (%)
28개	2	100	100
28개	3	80	44
28개	4	54.17	100
28개	5	100	100
28개	6	100	100
28개	7	100	100
28개	8	100	100
28개	9	100	100
28개	10	100	100
28개	11	100	100
28개	12	100	100
28개	13	100	100
28개	14	100	100

데이터 셋 28로 학습과 실험을 진행하였을 경우 평균 정확도는 95.31 퍼센트이다.

표 9. 시퀀스 길이 변경 실험 정리 6 (상태유지 LSTM 모델)

Dataset	Sequence Length (number)	First Experiment Accuracy (%)	Second Experiment Accuracy (%)
52개	2	96.00	78.00
52개	3	69.39	73.47
52개	4	100	100
52개	5	97.87	100
52개	6	73.91	86.96
52개	7	100	88.89
52개	8	100	88.64
52개	9	100	67.44
52개	10	100	100
52개	11	100	100
52개	12	100	100
52개	13	100	100
52개	14	100	100
52개	15	100	100
52개	16	100	100
52개	17	100	100
52개	18	100	100
52개	19	100	100
52개	20	100	100
52개	21	100	100
52개	22	100	100
52개	23	100	100
52개	24	100	100
52개	25	100	100
52개	26	100	100

데이터 셋 52로 학습과 실험을 진행하였을 경우 평균 정확도는 96.41 퍼센트이다. 총 38회 도출된 정확도의 평균은 95.86 퍼센트이다.

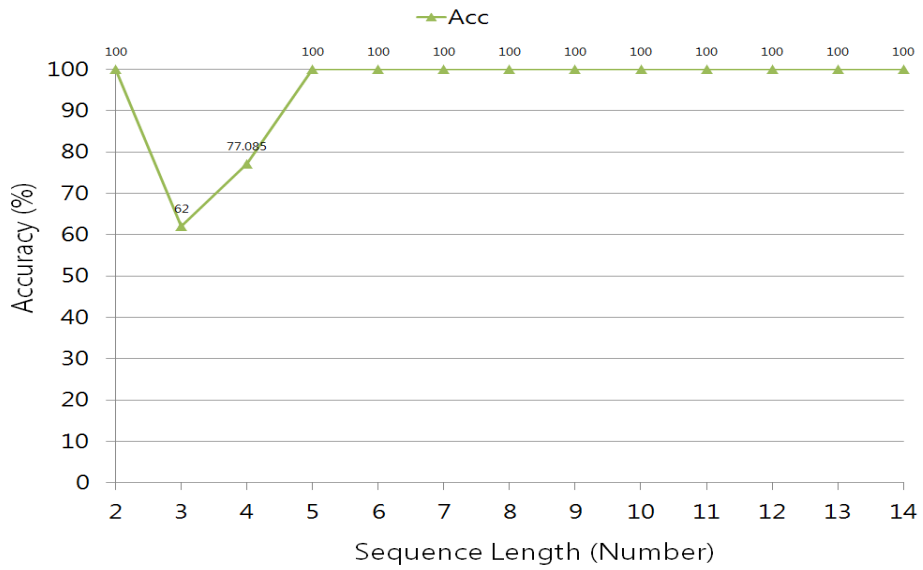


그림 47. 시퀀스 길이 변경 실험 정리 5 (상태유지 LSTM 모델)

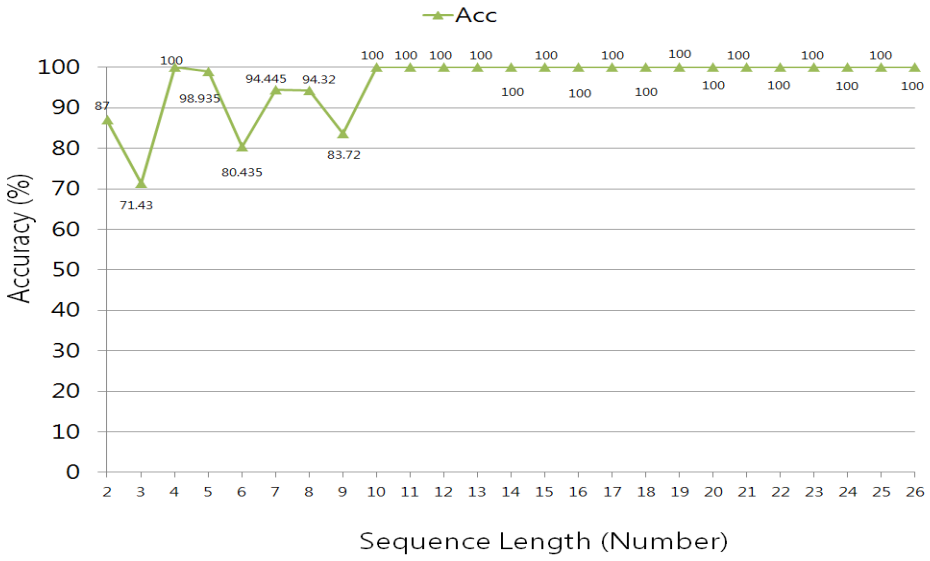


그림 48. 시퀀스 길이 변경 실험 정리 6 (상태유지 LSTM 모델)

4.2.5 실험 결과

실험에 사용한 데이터 셋 28과 데이터 셋 52는 비정형 데이터로써 알파벳과 숫자가 비순차로 이루어진 데이터 셋이다. 데이터 셋은 알파벳 a에서 z, 숫자 1부터 9까지 랜덤으로 생성한 값이다. 표 10은 각 모델 기반 시스템의 실험 결과를 정리한 것이다.

표 10. 성능 평가

비교 항목	DNN 모델 기반 시스템	기본 LSTM 모델 기반 시스템	상태유지 LSTM 모델 기반 시스템
평균 정확도	Rank 3	Rank 2	Rank 1
학습 속도	Rank 1	Rank 2	Rank 3
효율성	Rank 2	Rank 1	Rank 3
배치사이즈	5	5	1
가중치 초기화 사이즈	x	x	6
비율	2 : 1	6 : 1	5 : 1

DNN 모델 기반 데이터 예측 시스템으로 진행한 실험을 통해 도출된 결론은 다음과 같다. 데이터 셋의 데이터 개수와 시퀀스 길이의 비율이 2 : 1 일 때 정확도가 100 퍼센트에 도달했다. 이는 반복 학습의 횟수가 100회인데도 불구하고 높은 정확도가 결과로 도출되었다. 이러한 결과를 바탕으로 DNN 모델 기반 데이터 예측 시스템의 이상적인 설정 값은 배치사이즈 5, 데이터 셋과 시퀀스 길이의 비율을 2 : 1이다.

기본 LSTM 모델 기반 데이터 예측 시스템으로 진행한 실험을 통해 도출된 결론은 다음과 같다. 데이터 셋이 적은 경우 정확도가 100 퍼센트에 도달한다. 이에 반해 데이터 셋의 양이 커질 경우 정확도가 약간 낮아지는데 데이터 셋과 시퀀스 길이의 비율이 6 : 1에 가까울수록 정확도가 높게 도출되었다. 이러한 결과를 바탕으로 기본 LSTM 모델 기반 데이터 예측 시스템의 이상적인 설정 값은 배치사이즈 5, 데이터 셋과 시퀀스 길이의 비율을 6 : 1이다.

상태유지 LSTM 모델 기반 데이터 예측 시스템으로 진행한 실험을 통해 도출된 결론은 다음과 같다. 데이터 셋 28로 진행한 실험에서는 기본 LSTM 모델 기반 데이터 예측 시스템에 비해 평균 정확도가 낮았으나 데이터 셋 52로 진행한 실험에서는 높은 것으로 나타났다. 이는 상태유지 LSTM 모델의 가중치 초기화 방식이 다른 모델들과 다르기 때문에 데이터 셋이 많아질수록 효율성이 높아지게 된다. 또한 데이터 셋과 시퀀스 길이의 비율이 5 : 1 부터는 시퀀스의 길이 변화에 상관없이 정확도가 100 퍼센트로 도출되었다. 이러한 결과를 바탕으로 상태유지 LSTM 모델 기반 데이터 예측 시스템의 이상적인 설정 값은 가중치 초기화 사이즈 6, 데이터 셋과 시퀀스 길이의 비율을 5 : 1이다. 그림 49는 평균 정확도를 비교한 그래프이고 그림 50, 51, 52는 각 시스템의 최저 정확도와 최고 정확도를 표현한 그래프이다. 그리고 표 11은 각 시스템이 학습 및 데이터 예측에 소요한 시간을 정리한 것이다.

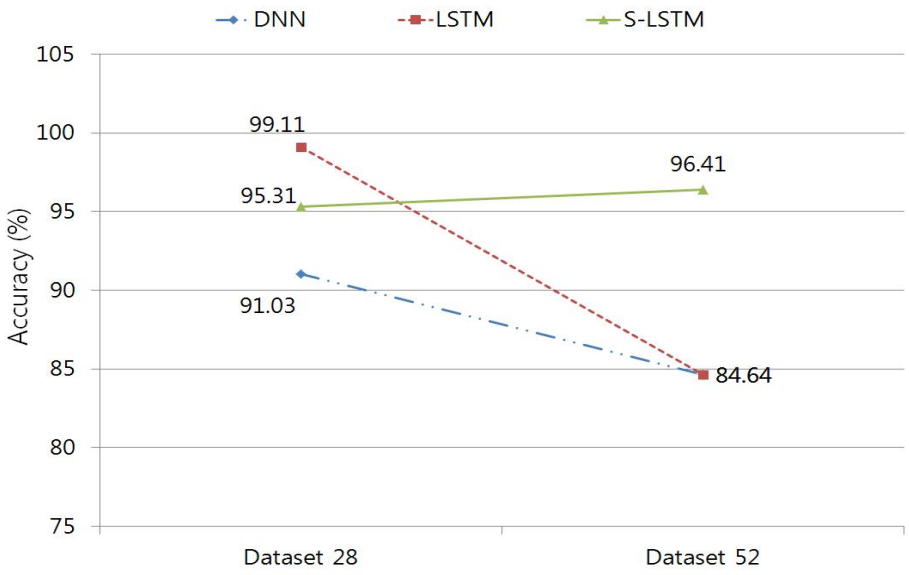


그림 49. 평균 정확도 비교 그래프

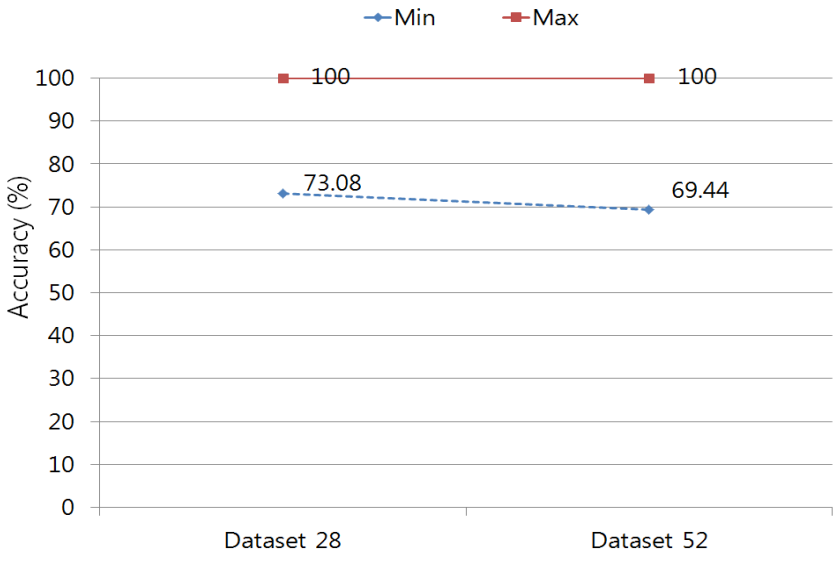


그림 50. 최저 정확도와 최고 정확도 1 (DNN 모델)

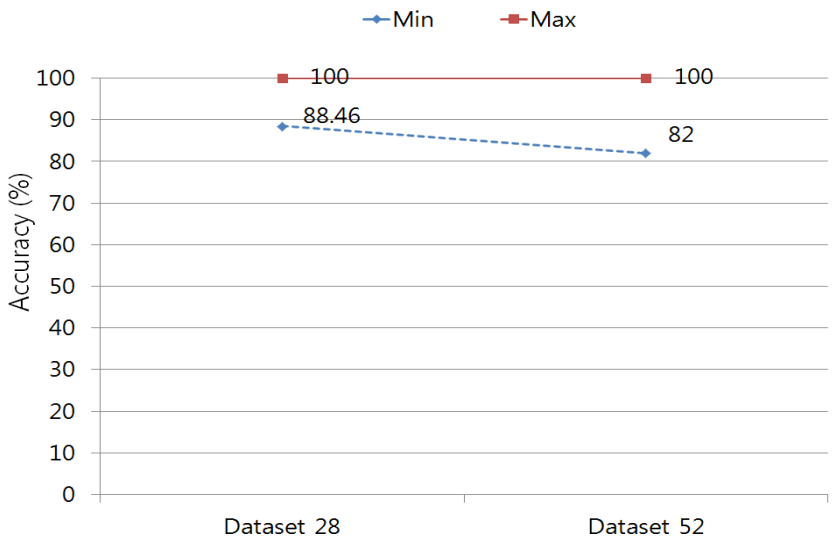


그림 51. 최저 정확도와 최고 정확도 2 (기본 LSTM 모델)

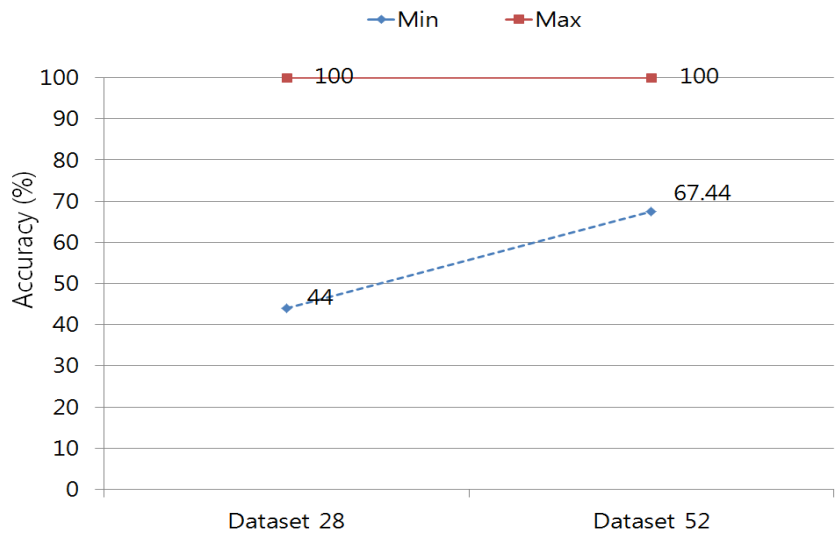


그림 52. 최저 정확도와 최고 정확도 3 (상태유지 LSTM 모델)

표 11. 학습 및 데이터 예측에 소요되는 시간

Dataset	DNN 모델 기반 시스템	기본 LSTM 모델 기반 시스템	상태유지 LSTM 모델 기반 시스템
Dataset 28	1ms ~ 2ms	3ms ~ 4ms	2s ~ 3s
Dataset 52	2ms ~ 3ms	3ms ~ 4ms	3s ~ 4s

DNN 모델 기반 데이터 예측 시스템은 정확도가 가장 낮았으나 학습 및 데이터 예측에 소요되는 시간이 가장 짧았다.

기본 LSTM 모델 기반 데이터 예측 시스템은 데이터 셋 28에서 정확도가 가장 높았으며 데이터 셋 52에서는 정확도가 많이 낮아졌다. 그럼에도 오차범위가 가장 좁았다.

상태유지 LSTM 모델 기반 데이터 예측 시스템은 평균 정확도가 가장 높았으나 학습 및 데이터 예측에 소요되는 시간이 길었다. 또한 데이터 셋이 길어질수록 소요되는 시간이 기하급수적으로 늘어남에 따라 고성능 컴퓨터로 시스템을 구축하지 않을 경우 효율성이 떨어지게 된다.

4.3 고찰

정확도와 데이터 예측에 소요되는 시간을 고려하여 각 시스템을 평가한다면 기본 LSTM 모델 기반 시스템이 가장 효율적이다. 상태유지 LSTM 모델 기반 데이터 예측 시스템이 정확도가 높더라도 필요한 시간이 길기 때문에 비슷한 시간을 기본 LSTM 모델 기반 시스템에 적용한다면 반복 학습의 횟수를 늘릴 수 있고 이로 인해 정확도가 상승하여 상태유지 LSTM 모델보다 평균 정확도가 높게 도출된다. 상태유지 LSTM 모델 기반 데이터 예측 시스템이 학습과 데이터 예측에 소요한 시간은 기본 LSTM 모델 기반 데이터 예측 시스템에 비해 10배 이상 길었다. 이는 반복 학습의 횟수를 1000회 이상으로 설정한 기본 LSTM 모델 기반 데이터 예측 시스템이 필요로 하는 시간보다 길다. 또한 기본 LSTM 모델 기반 데이터 예측 시스템은 반복 학습의 횟수가 200회 이상일 때부터 평균 정확도가 100 퍼센트에 도달하였다. 이러한 결과들을 바탕으로 데이터 예측에 가장 적합한 모델은 기본 LSTM 모델이다.

제안하는 시스템은 기존의 데이터 예측 시스템들을 분석하였고 정확도 향상을 위해 전처리기를 개발하였다. 전처리기는 비정형 데이터를 순차 데이터로 정제하여 딥 러닝 모델이 학습할 수 있도록 한다. 그리고 딥 러닝 모델은 3 가지 모델을 기반으로 시스템을 구축하였다.

DNN 모델 기반 시스템은 은닉층의 개수 3, 배치사이즈 5, 데이터 셋과 시퀀스 길이의 비율이 2 : 1 일 때 가장 높은 평균 정확도를 보였다. 이를 바탕으로 다음과 같은 결론을 도출하였다.

- 은닉층의 개수 3
- 배치사이즈 5
- 데이터 셋과 시퀀스 길이의 비율 2 : 1

기본 LSTM 모델 기반 시스템은 배치사이즈 5, 데이터 셋과 시퀀스 길이의 비율이 6 : 1 일 때 가장 높은 평균 정확도를 보였다. 이를 바탕으로 다음과 같은 결론을 도출하였다.

- 배치사이즈 5
- 데이터 셋과 시퀀스 길이의 비율 6 : 1

상태유지 LSTM 모델 기반 시스템은 가중치 초기화 사이즈를 6으로 설정하고 데이터 셋과 시퀀스 길이의 비율이 5 : 1 일 때 가장 높은 평균 정확도를 보였다. 이를 바탕으로 다음과 같은 결론을 도출하였다.

- 가중치 초기화 사이즈 6
- 데이터 셋과 시퀀스 길이의 비율 5 : 1

상태유지 LSTM 모델은 데이터 예측에 필요한 시간이 너무 긴 문제점이 있었다. 또한 학습일 진행될 때 오차범위가 커서 실제 데이터를 처리할 때 예측에 오류가 발생할 확률이 높았다.

이러한 실험 결과를 분석한 결과 기본 LSTM 모델이 보이는 데이터 예측의 정확도와 데이터 예측에 필요한 시간 등을 고려하였을 때 가장 적합한 시스템은 기본 LSTM 모델 기반 시스템으로 확인하였다.

실험 결과를 바탕으로 선정한 기본 LSTM 모델 기반 데이터 예측 시스템과 기존에 활용되고 있는 데이터 예측 시스템을 비교 분석한 결과는 표 12와 같다.

표 12. 기존 시스템과 제안하는 시스템의 비교 분석 결과

비교 항목	기존 시스템	제안하는 시스템
학습 데이터	비순차 데이터	순차 데이터
사용 모델	DNN, RNN, LSTM	LSTM
설정 값 최적화	X	O
데이터 셋과 시퀀스 비율	X	O
정확도	약 70 ~ 90 퍼센트	90 퍼센트 이상
반복 학습의 횟수	2000회 이상	100회

기존 시스템들은 비순차 데이터를 주로 활용하여 학습을 진행하였고 제안하는 시스템은 순차 데이터를 활용하여 학습을 진행하였다. 이는 LSTM 모델의 구조적인 특징으로 인해 순차 데이터를 기반으로 학습을 진행할 경우 정확도가 향상되기 때문에 전처리기를 통해 비정형 데이터를 순차 데이터로 정제하는 작업을 수행하였다.

기존 시스템들의 사용 모델로는 DNN 모델, RNN 모델, LSTM 모델 등 다양한 딥 러닝 모델을 사용하였다. 본 논문에서는 DNN 모델과 기본 LSTM 모델, 상태유지 LSTM 모델을 기반으로 시스템을 구축하였고 이들의 성능 비교 실험을 통해 기본 LSTM 모델 기반 시스템이 가장 효율적인 것으로 확인하였다.

기존 시스템들은 설정 값 최적화를 은닉층의 개수나 데이터 셋의 양으로 초점을 두었다. 이에 반해 본 논문에서는 배치사이즈, 데이터 셋과 시퀀스 길이의 비율 등을 분석하였고 같은 데이터 셋과 모델을 기반으로 학습 및 데이터 예측을 진행하여도 설정 값 최적화를 통해 정확도를 향상시킬 수 있는 것으로 확인하였다.

기존 시스템들의 정확도는 일반적으로 70 퍼센트에서 90 퍼센트 사이였다. 이에 반해 제안하는 시스템은 평균 91.87 퍼센트의 정확도를 보였다. 이는 기존 시스템에 비해 정확도가 향상되었다.

기존 시스템들의 반복 학습의 횟수는 일반적으로 2000회 이상이다. 이에 반해 제안하는 시스템은 반복 학습의 횟수를 100회로 설정하고 학습 및 데이터 예측을 진행하였음에도 기존 시스템에 비해 높은 정확도를 보였으며 데이터 예측에 소요되는 시간이 짧았다.

이러한 실험 결과 분석을 통해 다음과 같은 결론을 도출하였다.

- 학습에 사용할 데이터는 순차 데이터로 정제할 경우 정확도가 향상됨
- 기본 LSTM 모델이 데이터 예측 시스템에 적합함
- 배치사이즈, 데이터 셋과 시퀀스 길이의 비율 설정 등을 통해 정확도 향상과 학습 및 데이터 예측에 소요되는 시간을 단축시킬 수 있음

본 논문에서는 학습에 사용할 데이터를 순차 데이터로 정제하고 기본 LSTM 모델의 설정 값 최적화를 통해 기존 시스템들에 비해 예측의 정확도를 향상시켰으며 소요되는 시간을 줄였다. 이는 제안하는 시스템이 기존 시스템들에 비해 정확도와 효율성이 우수한 점을 입증하였다.

데이터 예측을 위해 시스템을 구축할 때 본 논문에서 제안하는 설정 값과 전처리기를 활용한다면 기존 시스템들에 비해 향상된 성능을 기대할 수 있을 것으로 사료된다.

Ⅴ. 결 론

데이터를 예측하는 시스템들이 다양한 방법론들을 활용하여 개발되고 있다. 그 중에서도 특정 데이터 뒤에 나올 데이터를 예측하기 위해서 딥 러닝 모델을 활용하는 시스템들은 대부분이 DNN 모델을 기반으로 개발되었다. DNN 모델은 데이터 셋과 학습을 진행할 때의 시퀀스 길이, 반복 학습의 횟수를 설정하고 학습을 진행한 뒤 데이터 예측을 진행한다.

데이터를 예측하기 위해 사용된 다른 딥 러닝 모델로는 RNN 모델이 있다. RNN 모델은 DNN 모델과 달리 학습이 진행된 시간의 흐름에 따라 가중치를 변경해가며 상황에 맞게 변화시키는 알고리즘으로 구성되어있다. DNN 모델에 비해 수치의 변화에 적응할 수 있는 장점이 있으나 시간의 흐름을 어떻게 설정하느냐에 따라 데이터를 예측하는 모델의 정확도가 달라지기 때문에 DNN 모델에 비해 오차범위가 큰 단점이 있었다. 이로 인해 RNN 모델을 활용하기 위해서는 이러한 단점을 해결하기 위한 알고리즘이나 전처리기 등을 추가로 구성해야하고 이는 RNN 모델을 활용하기 위해서 필요한 리소스가 DNN 모델에 비해 큰 것을 의미한다.

본 논문에서는 전처리기와 딥 러닝 모델을 기반으로 데이터 예측 시스템을 개발하였다. 전처리기에서는 각 모델이 학습을 진행하기 전에 비정형 데이터를 “데이터 : 인덱스” 구조의 순차 데이터로 정제한다. 이는 RNN 모델과 LSTM 모델이 순차 데이터를 분석하기에 적합한 구조를 지니고 있기 때문이다. 이러한 이유로 인해 전처리기는 데이터를 순차 데이터로 정제한다.

그리고 각 모델의 설정 값들에 대한 연구를 진행하였다. DNN 모델과 기본 LSTM 모델 기반 데이터 예측 시스템은 배치사이즈를 변경해가며 실험을 진행하였고 최적화된 배치사이즈를 도출하였다. 그리고 상태유지 LSTM 모델 기반 시스템은 가중치 초기화 사이즈 연구를 진행하였다. 또한 데이터 셋과 시퀀스 길이의 비율에 따라 정확도가 달라지는 것을 확인하였고 이를 분석하여 각 모델 기반 시스템들이 가장 높은 정확도를 도출하는 비율을 제시하였다.

DNN 모델 기반 데이터 예측 시스템은 은닉층의 개수 3, 배치사이즈 5, 데이터 셋과 시퀀스 길이의 비율이 2 : 1 일 때 가장 높은 정확도를 보였다. 정확도는 타 모델들에 비해 낮았으나 학습 및 데이터 예측에 필요한 시간이 가장 짧았다.

기본 LSTM 모델 기반 데이터 예측 시스템은 배치사이즈 5, 데이터 셋과 시퀀스 길이의 비율이 6 : 1 일 때 가장 높은 정확도를 보였다. 데이터 셋 28에서 진행한 실험에서는 타 모델 기반 시스템들에 비해 가장 높은 정확도를 보였으며 데이터 셋 52에서는 정확도가 가장 많이 낮아졌다. 학습 및 데이터 예측에 필요한 시간은 DNN 모델 기반 데이터 예측 시스템에 비해 상대적으로 길었다.

상태유지 LSTM 모델 기반 데이터 예측 시스템은 가중치 초기화 사이즈 6, 데이터 셋과 시퀀스 길이의 비율이 5 : 1 일 때 가장 높은 정확도를 보였다. 데이터 셋 28에서 진행한 실험에서는 기본 LSTM 모델 기반 시스템에 비해 정확도가 낮았으나 데이터 셋 52에서 진행한 실험에서는 타 모델 기반 시스템들에 비해 가장 높은 정확도를 보였다. 이에 반해 학습 및 데이터 예측에 필요한 시간이 가장 길었다.

3 가지 모델을 기반으로 구축한 시스템들을 분석한 결과 가장 적합한 것은 기본 LSTM 모델 기반 시스템이다. 정확도를 단일 기준으로 설정한다면 상태유지 LSTM 모델 기반 시스템이 적합하지만 데이터 예측에 소요되는 시간이 가장 긴 문제점이 있다. 또한 상태유지 LSTM 모델 기반 시스템이 100회의 반복 학습을 진행할 때 기본 LSTM 모델 기반 시스템은 1000회 이상의 반복 학습을 진행할 수 있다. 1000회 이상의 반복 학습을 진행할 경우 기본 LSTM 모델 기반 시스템의 평균 정확도는 100 퍼센트에 가까워지고 이러한 실험 결과를 바탕으로 데이터 예측을 목적으로 시스템을 구축할 때는 기본 LSTM 모델이 가장 적합하다는 결론을 도출하였다.

따라서 데이터 예측을 목적으로 시스템을 구축한다면 순차 데이터로 데이터를 정제하기 위한 전처리기와 기본 LSTM 모델을 기반으로 시스템을 구축하는 것이 효율적이다. 그리고 모델의 설정 값을 변경하여 높은 정확도를 도출하였다.

본 논문은 데이터 예측을 위해 개발되는 시스템들에 대해 높은 정확도와 효율성을 기대할 수 있는 설정 값에 대한 연구를 진행하였다. 이러한 점들로 인해 본 논문에서 제안하는 방법론이 다양한 분야에서도 높은 효율성을 기대할 수 있을 것으로 사료된다.

참고문헌

[1] https://steemit.com/ai/@martinmusiol/artificial-intelligence-blog-post-1

[2] http://physics2.mju.ac.kr/juhapruwp/?p=1517

[3] http://pythonkim.tistory.com/40

[4] I. S. Jang, C. H. Ahn, J. G. Seo, Y. S. Jang, “DNN based Speech Detection for the Media Audio,” Journal of Broadcast Engineering, Vol. 22, No. 5, pp. 632-642, 2017.

[5] I. S. Jang, C. H. Ahn, J. G. Seo, Y. S. Jang, “Enhanced feature extraction for speech detection in media audio,” Proceeding of 18th Annual Conference of the International Speech Communication Association (INTERSPEECH 2017), pp. 479-483, 2017.

[6] B. Lehner, G. Widmer and R. Sonnleitner, “Improving voice activity detection in movies,” Proceeding of 16th Annual Conference of the International Speech Communication Association (INTERSPEECH 2015), pp. 2942-2946, 2015.

[7] R. Füg, A. Niedermeier, J. Driedger, S. Disch, M. Müller "Harmonicpercussive-residual sound separation using the structure tensor on spectrograms," Proceeding of Acoustics, Speech and Signal Processing (ICASSP), 2016.

[8] Y. J. Lee, J. H. Nang, "A Personal Video Event Classification Method based on Multi-Modalities by DNN-Learning," Journal of Korea Information Science Society, Vol. 43, No. 11, pp. 1281-1297, 2016.

[9] J. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga G. Toderici, "Beyond Short Snippets: Deep Networks for Video Classification," Proc. of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, pp. 4694-4702, 2015.

[10] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, "Large-Scale Video Classification with Convolutional Neural Networks," Proc. of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Vol. 37, pp. 448-456, 2014.

[11] Z. Wu, Y. G. Jiang, J. Wang, J. Pu X. Xue, "Exploring Inter-feature and Inter-class Relationships with Deep Neural Networks for Video Classification," Proc. of the 22nd ACM International Conference on Multimedia, pp. 167-176, 2014.

[12] M. H. Jang, M. J. Lee, Y. G. Ha, "DNN Based Object Learning and Recognition System," Korea Information Science Society, pp. 1692-1694, 2016.

[13] S. W. Kim, J. W. Kim, "Customer Behavior Prediction of Binary Classification Model Using Unstructured Information and Convolution Neural Network - The Case of Online Storefront," Journal of Intelligence nd Information Systems, Vol. 24, No. 2, pp. 221-241, 2018.

[14] S. M. Ahn, "Deep learning architectures and applications," Journal of Intelligence and Information Systems, Vol. 22, No. 2, pp. 127-142, 2016.

[15] Yiğit, İ. O., A. F. Ateş, M. Güvercin, H. Ferhatosmanoğlu, and B. Gedik, "Call center text mining approach," Paper presented at the

Signal Processing and Communications Applications Conference, pp. 1-4, 2017.

[16] Gridach, M., H. Haddad, H. Mulki, “Churn identification in microblogs using convolutional neural networks with structured logical knowledge,” Paper presented at the Proceedings of the 3rd Workshop on Noisy User-Generated Text, pp. 21-30, 2017.

[17] K. T. Kim, B. M. Lee, J. W. Kim, “Feasibility of Deep Learning Algorithms for Binary Classification Problems,” Journal of Intelligence and Information Systems, Vol. 23, No. 1, pp. 95-108, 2017.

[18] LeCun, Y., Y. Bengio, and G. Hinton, “Deep learning,” Nature, pp. 436-444, 2015.

[19] S. H. Lee, J. H. Lee, “Customer Churn Prediction Using RNN,” Proceedings of the Korean Society of Computer Information Conference, Vol. 24, No. 2, pp. 45-48, 2016.

[20] M. A. H. Farquad, Vadlamani Ravi, and S. BapiRaju, "Churn prediction using comprehensible support vector machine: An analytical CRM application," Applied Soft Computing, Vol. 19, pp. 31-40, 2014.

[21] Runge J, Gao P, Garcin F, and Faltings B., "Churn prediction for high-value players in casual social game," 2014 IEEE Conference on Computational Intelligence and Games, pp. 1-8, 2014.

[22] Liao H. Y, Chen K. Y, Liu D. R, Chiu Y. L, "Customer Churn Prediction in Virtual Worlds," In Advanced Applied Informatics 2015 IIAI 4th International Congress on IEEE, pp. 115-120, 2015.

[23] Sharma A, Panigrahi D, Kumar, P, "A neural network based approach for predicting customer churn in cellular network services," arXiv preprint arXiv:1309.3945, 2013.

[24] Huang B, Kechadi M. T, and Buckley B, "Customer churn prediction in telecommunications," Expert Systems with Applications, Vol. 39 No. 1, pp. 1414-1425, 2012.

[25] J. Y. Park, "Estimation of Electrical Loads Patterns by Usage in the Urban Railway Station by RNN," The Transactions of the Korean Institute of Electrical Engineers, Vol. 67, No. 11, pp. 1536-1541, 2018.

[26] Zhichao Shi, Hao Liang, and Venkata Dinavahi, "Direct interval forecast of uncertain wind power based on recurrent neural networks," IEEE Trans. Sustain. Energy, Vol. 9, No. 3, pp. 1177-1187, 2018.

[27] Y. H. Kim, Y. K. Hwang, T. G. kang, K. M. Jung, "LSTM Language Model Based Korean Sentence Generation," The Journal of Korean Institute of Communications and Information Sciences, Vol. 41 No. 5, pp. 592-601, 2016.

[28] Van Quan Nguyen, Linh Van Ma, Jinsul Kim, "LSTM-based Anomaly Detection on Big Data for Smart Factory Monitoring," Journal of Digital Contents Society, Vol. 19, No. 4, pp. 789-799, 2018.

[29] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, "Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges," IEEE Access, Vol. 6, pp. 6505-6519, 2017.

[30] M. Shafiee and M. Finkelstein, “An optimal age-based group maintenance policy for multi-unit degrading systems,” Reliability Engineering & System Safety, Vol. 134, pp. 230-238, 2015.

[31] Kv, R. Satish, and N. P. Kavya, “Trend Analysis of E-Commerce Data using Hadoop Ecosystem,” International Journal of Computer Applications, Vol. 147, No. 6, pp. 1-5, 2016.

[32] L. Zhou, S. Pan, J. Wang, and A. V. Vasilakos, “Machine learning on big data: Opportunities and challenges,” Neurocomputing, Vol. 237, pp. 350-361, 2017.

[33] Jordan, I. Michael and M. M. Tom, “Machine learning: Trends, perspectives, and prospects,” Science, Vol. 349, No. 6245, pp. 255-260, 2015.

[34] J. W. Kim, H. A. Pyo, J. W. Ha, C. K. Lee, J. H. Kim, “Deep learning algorithms and applications,” Korea information science society, Vol. 33, No. 8, pp. 25-31, 2015.

[35] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, “Long short term memory networks for anomaly detection in time series,” Proceedings, pp. 89, 2015.

[36] T. Olsson and A. Holst, “A Probabilistic Approach to Aggregating Anomalies for Unsupervised Anomaly Detection with Industrial Applications,” in FLAIRS Conference, pp. 434-439, 2015.

[37] W. K. Lee, S. Y. Ahn, M. S. Yim, S. T. Chun, "Forecasting Energy Data Using LSTM," Korea Information and Science Society, pp. 693-695, 2016.

[38] D. H. Seo, J. S. Lyu, E. J. Choi, S. H. Cho, D. K. Kim, "Web

based Customer Power Demand Variation Estimation System using LSTM," Journal of the Korea Institute of Information and Communication Engineering, Vol. 22, No. 4, pp. 587-594, 2018.

[39] D. H. Shin, K. H. Choi, C. B. Kim, "Deep Learning Model for Prediction Rate Improvement of Stock Price Using RNN and LSTM," The Journal of Korean Institute of Information Technology Vol. 15, No. 10, pp. 9-16, 2017.

[40] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market prediction," FAU Discussion Papers in Economics, No. 11, pp. 310-342, 2017.

[41] J. W. Lee, "A Stock Trading System based on Supervised Learning of Highly Volatile Stock Price Patterns," Journal of Korean Institute of Information Scientists and Engineers, Vol 19, No. 1, pp. 23-29, 2013.

[42] K. J. Jeong, J. S. Choi, "Deep Recurrent Neural Networks," Communications of the Korean Institute of Information Scientists and Engineers, Vol. 33, No. 8, pp. 39-43, 2015.

[43] H. S. Kim, H. C. Song, J. W. Shin, S. K. Ko, B. T. Lee, "RNN-LSTM based Short-Term Electricity Demand Forecasting using Holiday Information," Electronics and Telecommunications Research Institute, Vol. 33, No. 8, pp. 39-43, 2015.

[44] W. He, “Deep neural network based load forecast,” Computer modeling and New technologies, vol. 18, No. 3, pp.258-262, 2014.

Deep Learning Model Optimization based Sequential Data Prediction System

Jong Won Lee

Department of Computer Engineering
Graduate School of Paichai University
Daejeon, Korea
(Supervised by Professor Hoekyung Jung)

Currently, data prediction systems analyze the data of a specific field by computer to predict the data and predict the data. In this method, it is possible to predict the data by analyzing the past data so that the person can derive the rule. On the other hand, predicting the data of the data which can not derive a rule has a limitation due to the ability of the person, and the accuracy may be lowered.

To solve this problem, computer was used to input vast amount of data into the data prediction program as learning data and to predict the data as a result. In order to utilize this methodology, data is predicted by applying a deep learning technique to a high performance computer. One of the areas where the methodology is used is to analyze the weather

data to predict the weather and to predict the data of the sporting events. Deep learning technology is a program that conducts learning based on learning data and analyzes the experimental data with progressed learning results. This is advantageous for analyzing large-scale data in the past than for human-to-human data analysis, which increases accuracy. Also, when the data is predicted by applying a suitable deep learning model according to the purpose, there is an advantage that the expected value of the accuracy is increased.

Currently, models used for predicting data among deep-learning models are Deep Neural Network (DNN) model and Recurrent Neural Network (RNN) model based on neural network structure. Although the DNN model can not find the rules in the learning data, it can increase the accuracy of data prediction through iterative learning. The RNN predicts the data as the weight to be applied in the hidden layer changes as learning proceeds, . On the other hand, the accuracy of DNN is increased if the number of iterative learning is high, and the accuracy of RNN is increased if the number of weight change is increased.

In this paper, we propose a data prediction system based on a deep learning model for data prediction. The proposed system also developed a preprocessor to refine unstructured data into sequential data. The preprocessor performs the function of refining the data before inputting the learning data to the deep learning model. Data is a data pair consisting of 'data: index' structure, and learning is performed by inputting a set of these data pairs into a deep learning model.

The deep learning model uses DNN model, basic Long short-term memory (LSTM) model, and stateful LSTM model to build each system. Then, the variation of the accuracy is analyzed while changing the set values of the respective models. We also experimented with varying the length of the sequence and present the ratio of the most accurate dataset and sequence length.

Based on these studies, we could expect higher accuracy than deep run

model based systems developed for predicting existing data.
If the proposed system is used in the field which requires at least high accuracy, it is considered that the learning data is more efficient than the existing systems.

Key words : Data Refined, Deep Learning, DNN, Machine Learning, LSTM, Pre-Processing, RNN

감사의 글(Acknowledgement)

5년의 대학원생으로써의 삶을 큰 사고 없이 지낼 수 있게 도와주신 지도교수 정회경 교수님께 머리 숙여 감사의 말씀을 전합니다. 많은 일들이 있었고 이에 따라 감정의 변화가 있었으나 부족한 제가 연구를 진행할 수 있게뜸 지도해주신 은혜는 죽어도 잊지 못할 것입니다. 또한 상중이랑 도연이, 도안이의 도움이 있었기에 논문에 집중할 수 있었고 이에 대한 고마움을 표현합니다. 그리고 연구실을 통해 알게 된 모든 이들에게 감사의 인사를 전합니다.

대학교에서부터 대학원 생활을 무난하게 할 수 있게끔 평생을 저의 뒷바라지 해주신 부모님께 감사하단 말씀을 전합니다. 고단한 삶을 사시지만 아들을 위해 희생해주신 부모님이 있었기에 마지막까지 연구를 진행할 수 있었다고 생각됩니다. 이제 제가 보답할 차례이니 사랑한다는 말을 전하고 싶습니다.

마지막으로 제 심신의 버팀목이 되어준 민경이에게 사랑한다고 전하고 싶습니다. 이종원이라는 사람에게 도움과 격려를 주신 모든 분들께 머리 숙여 감사드립니다.

2018년 12월

이 종 원 드림