

LSTM을 이용한 재밍 기법 예측

이경훈¹⁾ · 조제일²⁾ · 박정희^{*,1)}

¹⁾ 충남대학교 컴퓨터공학과

²⁾ 국방과학연구소 제2기술연구본부

Prediction of Jamming Techniques by Using LSTM

Gyeong-Hoon Lee¹⁾ · Jeil Jo²⁾ · Cheong Hee Park^{*,1)}

¹⁾ Department of Computer Science and Engineering, Chungnam National University, Korea

²⁾ The 2nd Research and Development Institute, Agency for Defense Development, Korea

(Received 26 November 2018 / Revised 24 January 2019 / Accepted 25 February 2019)

ABSTRACT

Conventional methods for selecting jamming techniques in electronic warfare are based on libraries in which a list of jamming techniques for radar signals is recorded. However, the choice of jamming techniques by the library is limited when modified signals are received. In this paper, we propose a method to predict the jamming technique for radar signals by using deep learning methods. Long short-term memory(LSTM) is a deep running method which is effective for learning the time dependent relationship in sequential data. In order to determine the optimal LSTM model structure for jamming technique prediction, we test the learning parameter values that should be selected, such as the number of LSTM layers, the number of fully-connected layers, optimization methods, the size of the mini batch, and dropout ratio. Experimental results demonstrate the competent performance of the LSTM model in predicting the jamming technique for radar signals.

Key Words : Jamming(재밍), Radar Signal(레이다 신호), Deep Learning(딥러닝), LSTM(장단기 기억 구조)

1. 서론

재밍이란 특정 주파수나 전파의 사용을 교란시키는 것으로 전자전(electronic warfare)에서는 적군의 통신, 레이더 등을 교란시키는 전자공격(electronic attack)을

의미한다. 함정 또는 항공기에 탑재되는 자체 보호용 재머에서 재밍의 목적은 적 레이더 또는 RF(Radio Frequency) 탐색기로부터 발생한 신호를 적절한 방법으로 교란시킴으로써 아군을 보호하는 것이다. 재밍 기법은 크게 잡음 재밍, 거리 기만 재밍, 속도 기만 재밍, 각도 기만 재밍으로 구분된다^[1]. 수신된 위협 신호의 특성에 따라 재밍 기법의 효과가 다르기 때문에, 전자전에서 발생하는 다양한 위협 신호에 대하여 적

* Corresponding author, E-mail: cheonghee@cnu.ac.kr
Copyright © The Korea Institute of Military Science and Technology

절한 재밍 기법을 적용하는 것이 필요하다.

기존의 재밍 방법은 장기간에 걸쳐 레이더 신호를 수집, 분석하여 해당 신호 타입에 대한 적절한 재밍 기법을 라이브러리에 저장해두고, 위협 신호가 수신되었을 때 라이브러리 정보를 기반으로 재밍 기법을 선택하는 방법이다. 위협 신호가 라이브러리에 존재하는 경우에는 쉽고 빠르게 재밍 기법을 적용할 수 있지만, 기존과 다르게 변형된 위협 신호가 수신되는 경우에 대하여 라이브러리를 사용하는 방법은 한계가 있다. 그러나, 기계학습 알고리즘을 사용하여 위협신호에 대한 재밍기법 예측 모델을 학습하여 주파수나 펄스반복주기가 달라져도 학습한 모델을 이용하여 가장 유사한 위협신호 패턴에 대한 재밍기법을 예측함으로써 기존의 방식보다는 큰 재밍효과를 기대할 수 있다.

레이더 신호는 펄스 도착 시간에 따라 기록된 PDW(Pulse Description Word) 리스트로 저장된다. 시간에 따라 RF(Radar Frequency)와 PRI(Pulse Repetition Interval)가 변조될 수 있기 때문에 시간에 따른 순차적인 특성을 파악하는 것이 필요하다. 기계학습 딥러닝 알고리즘 중 순환 신경망(Recurrent Neural Network, RNN)은 순차데이터의 시간 의존성을 기반으로 특징을 추출할 수 있는 모델이다. 특히 LSTM(Long Short-Term Memory)^[9]은 순차데이터의 장단기 의존성을 파악하는데 뛰어난 성능을 보인다. 시퀀스 길이가 길어져도 과거의 정보들을 유지하면서 현재의 입력 값과 결합할 수 있기 때문에, 데이터 간의 시간 의존적 관계를 학습하는데 효과적이다.

본 논문에서는 LSTM 모델을 학습하여 위협 신호에 대응하는 효과적인 재밍 기법을 예측하는 방법을 제안한다. LSTM은 데이터로부터 자동적으로 특징을 추출하여 시퀀스를 분류하는 문제에 뛰어난 성능을 보임에도 불구하고, 우리가 아는 범위 내에서는 위협 신호에 대한 재밍 기법 예측을 위하여 LSTM을 사용한 연구 사례는 없다. LSTM 모델의 구조를 다양하게 시도하면서 성능을 검증하기 위하여, LSTM 층의 수, 전결합 층의 수, 최적화 방법, 미니 배치의 크기, 드롭아웃 적용 비율 등 필요한 학습 파라미터에 대한 비교 실험을 통하여 안정적으로 학습하면서 모델의 성능을 높일 수 있는 방법을 제안한다.

2장에서 관련 연구에 대하여 소개하고 3장에서는 딥러닝에 대한 기본 이론을 설명한다. 4장에서는 실험에서 사용한 데이터와 기본 모델에 대하여 설명하고 5장에서는 학습 파라미터와 모델 구조에 따른 비교

실험의 결과를 보여준다. 6장에서는 최종 LSTM 모델에 의한 실험 결과를 나타내고 결론을 맺는다.

2. 관련 연구

레이더 신호 분야에서는 신호의 특징을 파악하기 위하여 시간 영역과 주파수 영역에서 다양한 추출 방법을 사용한다. L. Jian-xun et al.^[2]은 전자전에서 항재밍(anti-jamming)을 목적으로 적군을 탐지하기 위한 레이더 신호와 아군을 기만하기 위하여 적군이 생성한 재밍 신호를 구분하기 위하여 진폭의 평균과 분산 등을 이용한 통계값과 푸리에 변환에 의해 변환된 신호 데이터의 진폭(amplitude)과 위상(phase)의 값을 이용하여 만든 이변량 스펙트럼(bispectrum)을 특징값으로 추출하여 일반 레이더 신호와 재밍 레이더 신호의 차이를 비교하여 보여주었다.

A. Mendoza^[3]는 주파수(frequency) 변조 형태에 따른 레이더 신호를 분류하기 위하여 시간 영역에서 자기 상관 계수와 주파수 영역에서 멱스펙트럼(power spectrum)값을 레이더 신호의 특징값으로 추출하였고 인공신경망(Neural Network, NN) 모델을 이용하여 신호를 분류하였다. Y. J. Ryoo^[4]는 레이더 신호에 적용된 PRI 변조 방법에 따라 신호를 분류하기 위하여 자기 상관 계수를 이용하였다. 각 변조 방법에 해당하는 신호의 자기 상관 계수를 구하여 비교 분석하고 변조 방법에 따라 신호를 구분하기 위한 구분자를 정의하였다. 레이더 신호가 입력되었을 때 정의된 구분자를 이용하여 신호에 적용된 PRI의 변조 방법을 예측하여 신호를 분류하였다.

G. Noone^[5]은 PRI 변조에 따른 레이더 신호를 분류하기 위하여 펄스 도착 시간(Time Of Arrival, TOA) 사이의 미분값을 이용한다. 잡음을 고려하여 임계값 ϵ 을 정하고 미분값이 ϵ 보다 크면 1(증가), 미분값이 $(-\epsilon, \epsilon)$ 에 있으면 0(일정), $-\epsilon$ 보다 작으면 -1(감소)로 부분열을 구한 후 평균을 계산하여 특징값으로 사용하였다. 추출한 특징값을 이용하여 3층의 다층 퍼셉트론(Multilayer Perceptron, MLP) 모델을 학습하였다. 이 방법은 특징 추출과 모델 학습을 자동적으로 빠르고 쉽게 구현할 수 있지만, 적절한 임계값을 정해주어야 하는 문제가 있다. J. Kauppi et al.^[6]은 레이더 신호의 TOA 사이의 미분값을 사용하되 임계값이 TOA의 값에 따라 변화도록 설정하여 PRI 변조 방법에 따라 레

이다 신호를 분류하였다. J. Han et al.^[7]는 G. Noone^[5]와 동일한 방법으로 TOA 사이의 미분값을 이용하여 -1, 0, 1을 성분으로 가지는 부분열을 구한다. 레이더 신호의 부분열에서 -1, 0, 1이 존재하는 비율을 각각 계산하여 지니 인덱스, 엔트로피 함수 등으로 특징값을 추출하고 커널 SVM(Support Vector Machine)으로 PRI의 변조 방법에 따라 레이더 신호를 분류하였다.

특징값을 레이더 신호로부터 직접 추출하지 않고, 딥러닝 기술을 이용하여 자동적으로 특징값을 추출하고 분류하는 연구도 있다. Z. Wu et al.^[8]는 5개의 재밍 모드 타입, audio jamming, narrowband jamming, pulse jamming, sweep jamming, spread spectrum jamming을 분류하기 위하여 CNN(Convolutional Neural Network) 3층, 풀링(pooling)층 2층, 전결합층(fully connected layer) 4층, 배치 정규화층 4층으로 구성된 13층의 인공신경망을 사용하였다. CNN을 사용한 모델은 자동으로 특징을 추출하고 신호의 순차적 정보를 지역적으로 이용할 수 있다는 장점이 있지만, 지역적 정보만 얻을 수 있기 때문에 신호의 장기 시간 의존성을 파악하기 어렵다는 단점이 있다.

3. 딥러닝

3.1 인공신경망(Artificial Neural Network, ANN)

인공신경망은 인간이나 생물의 뇌신경 시스템으로부터 힌트를 얻은 학습 알고리즘으로 Fig. 1과 같이 인공 뉴런(노드)의 가중치 결합으로 형성한 네트워크에 대해 학습을 통하여 뉴런들의 결합 가중치 크기를 결정하여 문제를 해결하는 모델을 의미한다.

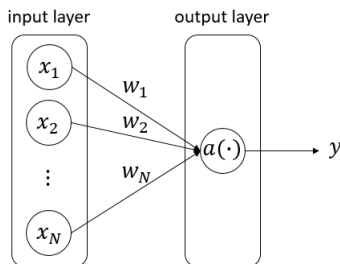


Fig. 1. The basic structure of ANN

인공 신경망은 다차원 입력 $X=(x_1, \dots, x_N)$ 을 받아서 식 (1)에 의해 계산되는 y 를 출력하는 구조이다.

$$y = a\left(\sum_{i=1}^N w_i x_i\right) = a(W^T X) \quad x_0 = 1, i = 1, \dots, N \quad (1)$$

$W=(w_0, \dots, w_n)$ 에서 w_i 는 입력의 i 번째 요소 x_i 에 대한 결합 가중치(weight)로써 실수값을 가진다. 함수 $a(\cdot)$ 는 일반적으로 비선형 함수이며, 활성화함수(activation function) 혹은 출력함수(output function)이라고 한다. 심층 신경망(Deep Neural Network, DNN)은 입력층과 출력층 사이에 여러 개의 은닉층으로 이루어진 인공 신경망이다. 입력 데이터 $X_i=(x_{i1}, x_{i2}, \dots, x_{iJ})$ 와 이에 대한 타겟값 $y_i=(y_{i1}, \dots, y_{iK})$ 의 쌍들의 집합이 주어졌을 때, 입력 데이터에 대해 신경망의 출력값이 타겟값에 가까워지도록 결합 가중치 w 를 반복적으로 수정하며 학습을 진행한다. 신경망 모델에서는 출력값 \hat{y}_i 와 타겟값 y_i 간 오차 함수로 주로 식 (2)에 의한 교차 엔트로피 오차(Cross-Entropy Error, CEE)함수와 평균 제곱 오차(Mean Squared Error, MSE)함수를 사용한다.

$$CEE = -\sum_{i=1}^N \sum_{k=1}^K (y_{ik}) \ln(\hat{y}_{ik}) \quad (2)$$

$$MSE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|^2$$

딥러닝 신경망은 기존의 신경망보다 더 많은 은닉층을 가진 깊은 신경망을 말한다. 모델이 깊어짐에 따라 발생하였던 기술기 소실문제와 과적합 문제는 신경망 모델의 가중치 초기화 방법, 활성화 함수 ReLU(Rectified Linear Unit)의 사용, 정규화 방법 등의 연구가 진행되면서 어느 정도 해결이 되었고, 메모리와 그래픽 카드 등의 하드웨어가 발달하면서 심층신경망의 학습이 가능해졌다. 딥러닝 신경망의 깊은 구조는 입력과 출력 사이의 더욱 복잡한 비선형 관계도 모델링을 할 수 있게 만들었다. 최근 딥러닝은 대량의 데이터 획득과 강력한 분산병렬 계산을 기반으로 음성 인식분야와 컴퓨터비전 분야에서 최고 수준의 성능을 보여주고 있다.

3.2 LSTM(Long Short-Term Memory)

RNN은 인공 신경망의 한 종류로 유닛간의 연결이 Fig. 2와 같이 순환적 구조를 가지는 특성을 가지고 있다. 이러한 구조는 은닉층에 저장되어 있는 과거의

정보들을 현재의 입력 값과 결합하여 사용할 수 있게 함으로써 입력되는 데이터 간의 시간 의존적 관계를 학습할 수 있게 한다.

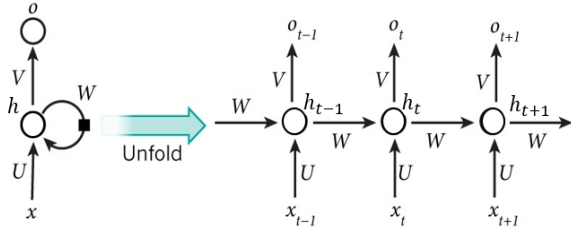


Fig. 2. The basic structure of RNN

그러나, RNN은 긴 길이의 시퀀스에 대해 기울기 소실이나 폭발과 같은 문제가 발생할 수 있다. 이에 대한 대안으로 LSTM^[9]이 사용된다. LSTM유닛은 forget 게이트, input 게이트, output 게이트 등으로 구성된 셀(cell)로 이루어져 있다. forget 게이트는 이전의 셀 상태를 얼마나 잊어버릴지 결정하는 게이트로, 이전 셀의 정보(h_{t-1})와 현재 입력된 데이터(x_t), 바이어스의 가중합(weighted sum)에 시그모이드 함수를 취한 형태이며 과거의 정보를 기억하는 정도를 나타내는 0과 1 사이의 값을 반환한다. LSTM은 기존 RNN에서 발생하였던 기울기 소실 문제를 해결하고자 하였다.

4. 실험 환경

4.1 실험 데이터

레이다의 정보는 펄스 도착 시간에 따라 PDW 리스트로 저장된다. RF(Radio Frequency)는 Fig. 3과 같이 고정(fix), 빠른 변화(agile), 도약(hopping), 정현파(sine), -톱니파(-sawtooth), +톱니파(+sawtooth) 등 6가지의 유형을 고려하였다. PRI는 Fig. 4와 같이 안정(stable), 지터(jitter), 스테거(stagger), 체류 후 변경(dwells&switch), 정현파, -톱니파, +톱니파 등 7가지의 유형을 고려하였다. 현재 운용되고 있는 레이더들의 제원을 참조하여 RF와 PRI 유형과 값의 범위를 다르게 설정하여 640개 위협신호 타입을 구성하였다. 각 타입마다 최소 500개에서 최대 4500개 사이의 PDW 파일을 생성하여 총 1,157,500개 PDW 위협 신호 파일을 생성하였다. 각 위협 신호 타입에 대하여 라이브러리 정보를 이용

하여 4개의 재밍 기법 중에서 하나의 재밍 기법을 할당하였다.

각 위협 신호는 AOA(Angle Of Attack), AMP(Amplitude), RF(Radio Frequency), PRI(Pulse Repetition Interval), PW(Pulse Width)의 5개 PDW 특성값의 시퀀스 $[X_1 = (p_1, i_1, f_1, a_1, m_1), \dots, X_n = (p_n, i_n, f_n, a_n, m_n)]$ 로 표현하였다. 따라서, RF값들의 시퀀스 $[f_1, \dots, f_n]$ 와 PRI값들의 시퀀스 $[a_1, \dots, a_n]$ 의 시변특성을 학습하여 유형을 파악하는 것이 재밍 기법 예측 성능 향상의 요인이 될 수 있다.

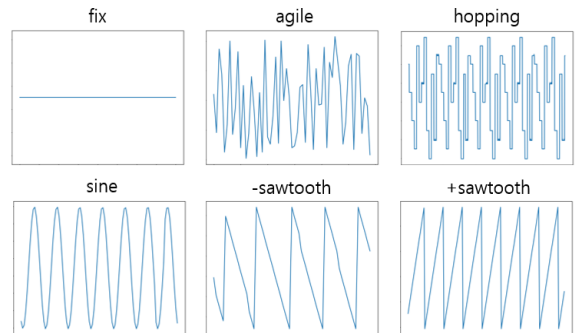


Fig. 3. Types of RF

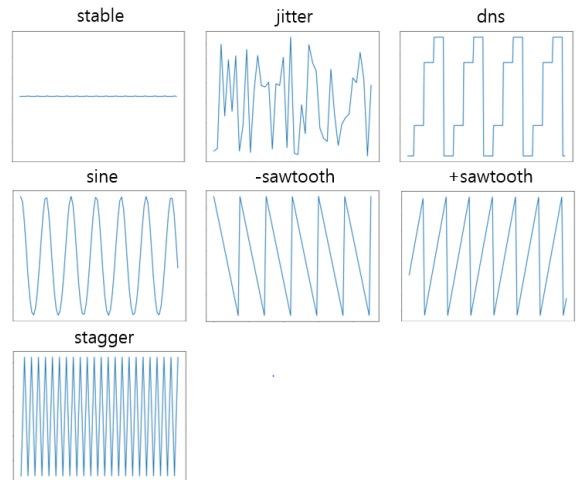


Fig. 4. Types of PRI

4.2 실험 방법

각 위협 신호 타입마다 데이터를 7:1:2의 비율로 나누어 전체 위협 신호데이터를 훈련(PDW 파일 810,250개), 평가(PDW 파일 115,750개), 시험(PDW 파일

231,500개) 데이터 셋으로 분할하였다. 훈련 데이터를 이용하여 모델을 학습하고 평가 데이터를 이용하여 가장 성능이 높은 모델을 선택한다. 선택한 모델을 이용하여 시험 데이터로 최종 성능을 측정할 수 있다. 모든 실험은 텐서플로우(tensorflow)를 이용하여 구현하였으며 NVIDIA GPU 1080Ti를 활용하여 학습하였다.

5. LSTM 모델 학습에 의한 재밍기법 예측

5.1 LSTM 기본 모델

4절에서 설명한 데이터를 이용하여 입력 레이다 신호에 대해 재밍 기법을 예측하는 LSTM 모델을 학습한다. 우리가 학습에 이용한 LSTM 기본 모델은 Fig. 5와 같다. 입력층으로부터 레이다 신호 데이터를 입력받아 LSTM 레이어를 통과시킨 후 출력층에서는 각 재밍 기법에 속한 확률을 소프트맥스 함수를 이용하여 계산한다. LSTM 모델에 입력되는 위협 신호는 AOA, AMP, RF, PRI, PW의 5개 특성값의 시퀀스로 표현되므로, Fig. 5의 입력층(input layer)은 5개의 유닛으로 구성된다. LSTM 레이어에는 200개 유닛을 설정하였고, 출력층은 4개의 재밍 기법 각각에 해당하는 4개의 유닛으로 구성된다.

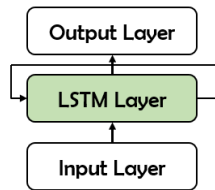


Fig. 5. The basic structure for LSTM model

최적의 재밍 기법 예측 성능을 가지는 LSTM 모델을 구성하기 위해 최적화 방법, 미니 배치의 크기, 드롭 아웃 기법, LSTM 층과 전결합층의 수, 입력 데이터의 속성 선택, 학습률 감소 방법을 차례대로 결정하면서 기본 모델을 수정해 가는 과정을 거치게 된다. Table 1은 결정해야 하는 파라미터와 파라미터값을 보여주고 있다. 7가지 종류의 파라미터에 해당하는 값을 차례대로 선택하는 과정을 진행하는 동안 아직 결정되지 않은 파라미터에 대해 기본적으로 사용한 값을 밑줄로 표시하였다.

Table 1. Parameters

No	매개 변수	변수값
1	최적화 방법	RMSProp, <u>Adam</u> , Adadelata
2	배치 사이즈 (개)	50, 100, <u>200</u>
3	드롭 아웃 비율 (%)	<u>0</u> , 10, 30, 50
4	LSTM 레이어 수 (개)	<u>1</u> , 2
5	전결합층	O, <u>X</u>
6	속성의 수 (개)	3, <u>5</u>
7	학습률 감소	O, <u>X</u>

5.2 최적화 방법(Optimization method) 선택

LSTM 모델 학습에서의 최적화 방법이란, 출력과 타겟 사이의 오차가 작아지도록 모든 가중치를 조정하는 방법을 의미한다. 경사하강법(gradient descent)은 식 (3)과 같이 가중치에 대한 비용 함수의 편미분 값을 학습률과 곱하여 가중치에 빼줌으로써 조정한다. λ 는 학습률을 나타내며 일반적으로 작은 실수값을 가진다.

$$w_i \leftarrow w_i - \lambda \frac{\partial L(W)}{\partial w_i} \quad (3)$$

Fig. 5의 기본 모델에 대하여 확률적 기울기 하강법(Stochastic Gradient Descent, SGD)을 기반으로 하는 방법으로 RMSProp, Adam, Adadelata를 적용하여 성능을 비교한다. 각 방법들은 학습률을 적응적으로 감소시킴으로써 안정적인 학습을 유도하는 방법들이다. Adam과 RMSProp는 0.003, Adadelata는 기존 논문에서 제안한 1.0을 초기 학습률로 설정한다. 총 학습 주기를 10으로 설정하고 각 주기마다 평가 데이터에 대한 정확도(validation accuracy)를 측정한다. Fig. 6은 3개의 비교 방법들의 평가 데이터에 대한 주기마다의 정확도를 보여준다. 가장 높은 평가 정확도를 가지는 주기에서의 정확도를 Table 2에 나타내었다. RMSProp 방법은 주기 2에서 오차 함수의 값에 nan이 발생하여 학습을 더 이상 진행할 수 없었다. Table 2는 Adam 최적화 방법을 사용하였을 때가 성능이 높다는 것을 보여준다. 이후에 파라미터 결정을 위한 실험에서는 최적화 방법을 Adam으로 설정하여 진행하였다.

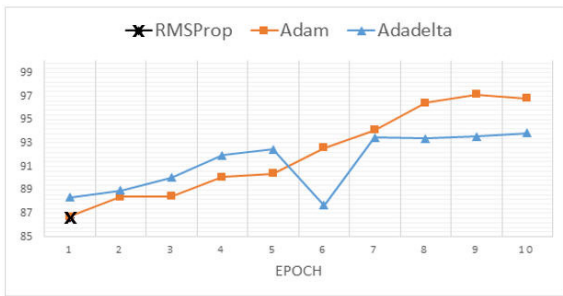


Fig. 6. Validation accuracy by optimization methods

Table 2. The best accuracy by optimization methods

최적화 방법	학습률	학습 정확도	평가 정확도	주기
RMSProp	0.003	81.21	86.72	1
Adam		96.11	97.15	9
Adadelta	1.0	93.18	93.86	10

5.3 미니 배치 크기 선택

미니 배치를 이용한 확률적 기울기 하강법(SGD)는 데이터를 일정 개수만큼 배치로 구성하여 한꺼번에 학습하는 방법으로 데이터를 하나씩 학습하는 것보다 실험적으로 좋은 성능을 보이지만, 미니 배치의 크기에 따라 성능이 달라지기 때문에 적절한 크기를 선택해야 하는 문제가 있다. 미니 배치 사이즈를 50, 100, 200으로 바꾸어가면서 모델을 학습하고 그에 따른 정확도를 비교하였다. Fig. 7은 배치사이즈에 따른 주기 마다의 평가 데이터 정확도를 보여준다. 배치 사이즈가 50과 100인 경우 각각 주기 4와 8에서 nan이 발생하였다. 학습이 진행될수록 200인 경우 성능이 높아지는 것을 알 수 있다.

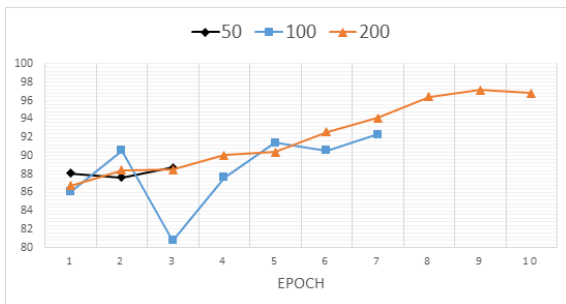


Fig. 7. The validation accuracy by batch sizes

Table 3에서는 가장 높은 평가 정확도를 보인 주기에서의 학습 및 평가 정확도를 보여준다. 배치 사이즈를 200으로 설정하여 이후의 모델 구성을 위한 실험을 진행하였다.

Table 3. The best accuracy by batch sizes

배치 사이즈	학습 정확도	평가 정확도	주기
50	87.98	88.70	3
100	90.73	92.30	7
200	96.11	97.15	9

5.4 드롭아웃 비율 선택

신경망의 깊이가 깊어질수록 데이터 간의 복잡한 구조를 학습할 수 있지만, 학습 데이터에 대하여 과적합이 발생할 수 있다. 드롭아웃(dropout)^[10]은 학습 시 확률적으로 은닉층의 유닛을 탈락시킴으로써 과적합을 막아주는 방법이다. 이 실험에서는 LSTM의 출력층에 드롭아웃을 적용하였고 드롭아웃 비율을 0, 10, 30, 50 %로 늘려가면서 실험을 한다. Fig. 8에서는 드롭아웃을 사용하지 않는 경우에 학습 주기 초기 부분

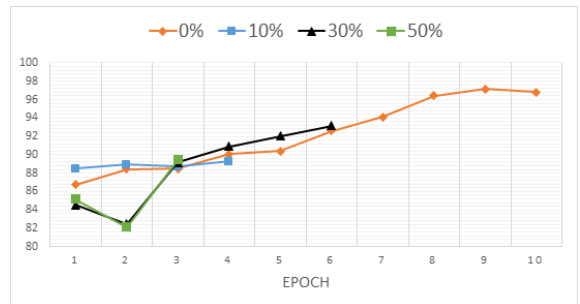


Fig. 8. The validation accuracy by dropout rates

Table 4. The best accuracy by dropout rates

Dropout rate(%)	학습 정확도	평가 정확도	주기
0	96.11	97.15	9
10	86.81	88.91	2
30	91.66	93.15	6
50	87.03	89.43	3

을 제외하고 모든 주기에서 평가 데이터에 대한 정확도가 대체로 더 좋은 것을 보여준다. Table 4는 평가 정확도가 가장 높았을 때의 주기에서의 정확도를 비교한다. 이후의 실험에서 드롭아웃을 적용하지 않는 모델 사용을 결정하였다.

5.5 LSTM 레이어 수 선택

LSTM 층의 수를 늘리면 데이터 간의 더욱 복잡한 관계를 학습할 수 있기 때문에 성능이 더 좋을 수 있다. Fig. 9의 왼쪽 그림과 같이 LSTM 1층을 사용하는 경우와 오른쪽 그림과 같이 LSTM 2층을 사용하는 경우를 비교하였다. LSTM layer의 유닛 수는 모든 층에서 200으로 설정하였고, 초기화 방법도 동일하게 설정하였다. Fig. 10에서 보여주듯이 주기 1에서는 1층을 사용한 모델의 평가 정확도가 높았지만 학습이 진행될수록 2층의 성능이 더 향상되었다. Table 5는 가장 높은 평가 정확도를 보인 주기에서의 학습 및 평가 정확도를 보여준다. LSTM 2층을 사용하였을 경우에 1층을 사용하였을 때보다 학습 정확도는 2.01 %, 평가 정확도는 1.25 % 높았다. 따라서 이후의 실험에서는 LSTM 2층을 사용하는 모델을 이용한다.

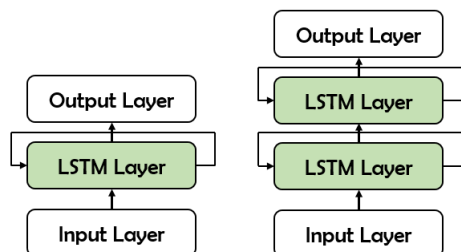


Fig. 9. Model structures with 1 or 2 LSTM layers

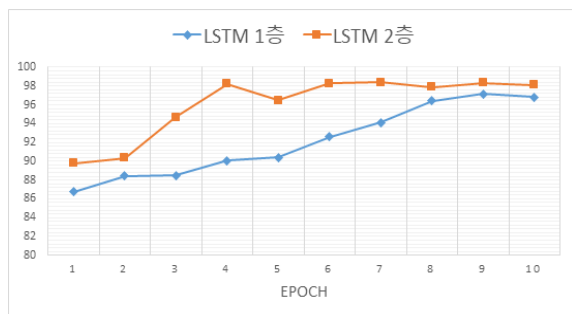


Fig. 10. The validation accuracy by LSTM layers

Table 5. The best accuracy by LSTM layers

LSTM layer의 수	학습 정확도	평가 정확도	주기
1층	96.11	97.15	9
2층	98.12	98.40	16

5.6 전결합층 선택

LSTM 층에 결합하는 전결합층(Fully Connected Layer, FCL)의 유무에 의한 성능 비교 실험을 위해 20주기를 학습하고 Fig. 11과 같이 전결합층을 추가한 경우와 추가하지 않은 경우의 모델에 대한 결과를 비교하였다. Fig. 12에서는 전결합층이 있는 경우 모델의 학습이 안정적이지 않지만 모델의 성능이 향상 될 수 있음을 보여준다. 성능을 향상시키면서 안정적으로 학습하는 방법은 5.8절에서 설명한다. Table 6은 전결합층을 사용한 경우에 사용하지 않은 경우보다 학습 정확도는 0.07 %, 평가 정확도는 0.30 % 높은 것을 보여준다. 따라서 우리는 전결합층을 추가한 모델을 사용하기로 결정하였다.

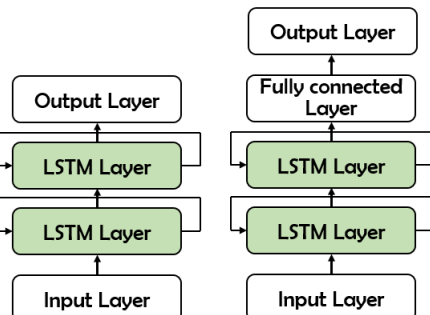


Fig. 11. Model structure with a fully connected layer

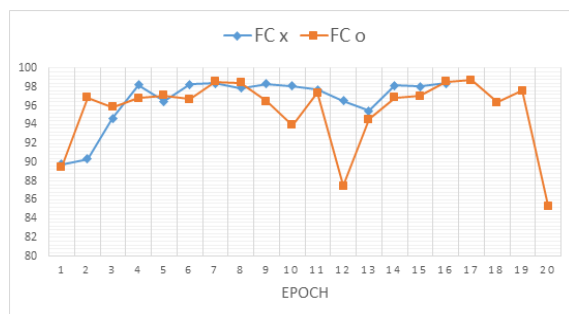


Fig. 12. The validation accuracy by FCL

Table 6. The best accuracy by FCL

전결합층	학습 정확도	평가 정확도	주기
O	98.19	98.74	17
X	98.12	98.4	16

5.7 속성 선택

속성 AOA, AMP, RF, PRI, PW 중 AOA와 AMP는 수신 안테나의 설정에 따라 변화하는 값으로 레이다 신호의 고유 특징이 아니다. 이 절에서는 신호의 특징을 나타내는 RF, PRI, PW만을 사용하여 학습하는 경우의 성능을 실험한다. Fig. 13은 3개의 특징을 사용한 경우 학습이 안정적으로 수행되고 성능이 높은 것을 나타낸다. Table 7은 3개의 특징을 사용한 경우가 5개 특성 모두 사용한 경우보다 학습 정확도는 0.82 %, 평가 정확도는 0.32 % 높은 것을 보여준다. 실험 결과에 따라 속성 AOA와 AMP를 제거하고 RF, PRI, PW를 사용하는 것으로 결정하였다. 따라서 입력층의 유닛 수는 3으로 설정하게 된다.

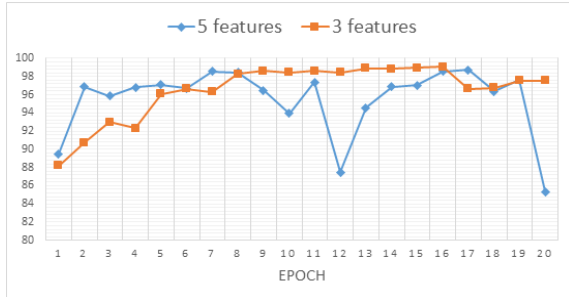


Fig. 13. The validation accuracy by feature sets

Table 7. The best accuracy by feature sets

특징의 수	학습 정확도	평가 정확도	주기
5	98.19	98.74	17
3	98.88	99.06	16

5.8 학습률 감쇠 방법 선택

학습률 감쇠 방법은 모델을 학습할 때 주기가 진행됨에 따라 가중치 업데이트에서 사용되는 학습률을 이전 주기보다 작게 설정하는 것을 의미한다. 학습률 감

쇠를 이용하면 가중치(weight) 및 편차(bias)의 변화하는 정도를 줄여줌으로써 모델을 안정적으로 학습할 수 있다. 학습률 감쇠를 위하여 학습 주기가 10이 지난 후 이전 학습률에 0.9를 곱하여 점차 줄여나가는 방법을 이용하였다. Fig. 14에서 학습률 감쇠를 사용하지 않았을 경우 주기에 따라 정확도의 변동이 크지만 학습률 감쇠를 사용하면 안정적으로 학습이 된다는 것을 보여준다. Table 8에서 학습률 감쇠 방법을 적용한 경우가 그렇지 않은 경우보다 학습 정확도는 0.11 %, 평가 정확도는 0.38 % 높아지는 것을 알 수 있다.

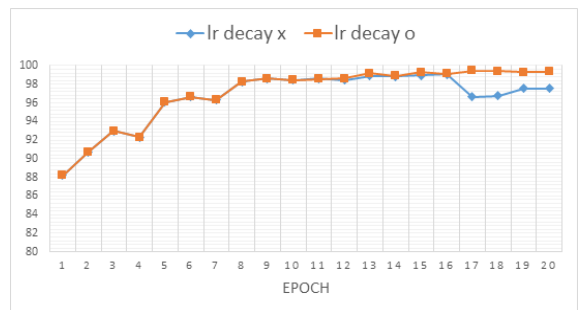


Fig. 14. Validation accuracy by learning rate decay

Table 8. The best accuracy by learning rate decay

감쇠	학습 정확도	평가 정확도	주기
O	99.12	99.44	17
X	99.01	99.06	16

6. 결론

5절에서는 실험을 통해 LSTM 모델의 파라미터 값을 차례대로 결정하는 과정을 보여주었다. 7:1:2의 비율로 나누어 구성된 훈련, 평가, 시험 데이터 셋에 대해, 훈련 데이터로 모델을 학습하고 평가 데이터의 정확도가 가장 높은 주기의 모델을 선택하는 과정을 통해 각 파라미터 값을 결정하였다. 훈련, 평가, 시험 데이터 셋으로 나누어 5절에서 결정한 설정 값으로 30 주기까지 학습한 뒤 최종적으로 모델을 선택하여 시험 데이터의 성능을 평가하는 과정을 10번 반복하여 수행하였다. Table 9는 10번 반복하여 얻은 최종 선택 모델의 정확도에 대한 평균과 표준편차를 보여준다.

Table 9. Accuracy(%) of train, validation, test data

	학습 정확도	평가 정확도	시험 정확도
평균	99.53	99.55	99.54
표준편차	0.077	0.078	0.073

본 논문에서 수행한 640개 타입의 위협 신호 데이터를 이용한 실험에서는 최적화 방법으로는 Adam 최적화 방법이 높은 성능을 보여주었고, 배치 사이즈 실험에서는 200이 적절한 값이었다. 드롭 아웃을 적용한 경우 과적합을 막으면서 모델의 성능을 높일 것으로 기대하였지만, 사용하지 않는 것이 오히려 좋은 성능을 보였다. 또한, 모델을 깊게 쌓아 복잡한 구조를 가질수록 좋은 성능을 보인다는 것을 실험적으로 확인하였다. 또한, 학습률 감소를 사용할 경우, 안정적으로 학습이 되었다. 향후 연구에서는, 새로운 미상 신호를 생성하여 본 논문에서 제안한 LSTM 모델을 기반으로 미상 신호에 대한 재밍 기법 예측 연구를 할 것이다.

References

- [1] R. N. Lothes, M. B. Szymanski, and R. G. Wiley, "Radar Vulnerability to Jamming," Artech House, 685 Canton Street, Norwood, MA, pp. 29-134, 1990.
- [2] L. Jian-xun, S. Qi. and Y. Hai, "Signal Feature Analysis and Experimental Verification of Radar Deception Jamming," IEEE CIE International Conference on Radar, Vol. 1, pp. 230-233, 2011.
- [3] A. Mendoza, "Classification of Radar Jammer FM Signals Using a Neural Network Approach," PhD Thesis, The University of Texas at El Paso, 2017.
- [4] Y. J. Ryoo and W. W. Kim, "Recognition of PRI Modulation Types of Radar Signals Using the Autocorrelation," The Institute of Electronics Engineers of Korea, Vol. 43, No. 3, pp. 61-68, 2006.
- [5] G. P. Noone, "A Neural Approach to Automatic Pulse Repetition Interval Modulation Recognition," Information, Decision and Control, pp. 213-218, 1999.
- [6] J. P. Kauppi and K. S. Martikainen, "An Efficient Set of Features for Pulse Repetition Interval Modulation Recognition," 2007 IET International Conference on Radar Systems, pp. 1-5, 2007.
- [7] J. W. Han and J. H. Jin, "A Novel Method of Recognizing PRI Modulation Type of Radar Signal based on Support Vector Machine," The Institute of Electronics and Information Engineers, Vol. 2017, No. 6, pp. 679-682, 2017.
- [8] Z. Wu, Y. Zhao, Z. Yin and H. Luo, "Jamming Signals Classification Using Convolutional Neural Network," IEEE International Symposium on Signal Processing and Information Technology(ISSPIT), pp. 062-067, 2017.
- [9] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computatnm, Vol. 9, No. 8, pp. 1735-1780, 1997.
- [10] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Over-Fitting," The Journal of Machine Learning Research, Vol. 15, No. 1, pp. 1929-1958, 2014.