JANT 한국항행학회논문지
Journal of Advanced Navigation Technolc

응용 및 융합 기술

J. Adv. Navig. Technol. 25(2): 150-155, Apr. 2021

LSTM 신경망을 이용한 1차원 객체추적

One-dimensional Object Tracking using LSTM Neural Network

박 선 배 · 유 도 식*
홍익대학교 대학원 전자전기공학과

Sun-Bae Park · Do-Sik Yoo*
Department of Electronic and Electrical Engineering, Graduate School, Hongik University, Seoul, 04066, Korea

[요 약]

객체추적은 객체의 위치변화를 찾는 것으로, 이전시간의 객체의 위치와 주어진 관측 데이터를 바탕으로 객체의 위치를 추적하는 신호처리의 한 분야이다. 객체추적 기법에는 대표적으로 칼만필터와 파티클필터가 있는데, 두 필터 모두 시스템 모델을 알고 있어야 좋은 성능을 낼 수 있다. 퍼셉트론 신경망에 피드백 루프를 추가한 재귀신경망은 데이터의 시계열적 상관관계를 활용할 수 있어 객체 추적에도 사용되고 있으며, 장기의존성 문제를 해결한 LSTM으로 발전하여 다양한 분야에 활용되고 있다. 본 논문에서는 이러한 LSTM의 추적 성능을 검증하기 위하여 1차원 객체 추적이라는 공통의 문제를 설정하고, 칼만필터, 파티클필터와의 추적 성능을 비교한다. 보다 다양한 관측 환경에서의 추적 성능 비교검증을 위하여 가우시안 잡음 외에도 라플라스, 지수, 균등 분포의 잡음이 있는 경우도 상정하였다. 그 결과 LSTM 신경망은 시스템 모델이 주어지지 않고, 학습데이터만으로 학습을 하여 안정적인 성능을 낼 수 있다는 것을 확인하였다.

[Abstract]

Object tracking is a technique of signal processing that estimates objects locations based on past locations and present time observed data. While, Kalman filter and particle filter are among the most notable object tracking schemes, these filters need to know the system model to achieve optimal performance. The recursive neural network (RNN) with a feedback loop added to the perceptron neural network can be used for object tracking. Also, RNN evolved into long-short term memory (LSTM) that solved the long-term dependence problem and is being used in various fields. In this paper, in order to study the tracking performance of LSTM, we consider a simple problem of one-dimensional object tracking, and compare the tracking performance with Kalman and particle filters. In order to test the tracking performance in diverse observation environments, various noise models such as Gaussian, Laplace, exponential, and uniformly distributed noises are considered. Under the various circumstances, we observe that LSTM neural network achieves fairly stable performance without knowing the system model.

Key word : Artificial neural network, Kalman filter, Long-short term memory, Object tracking, Particle filter.

https://doi.org/10.12673/jant.2021.25.2.150

Received 16 March 2021; **Revised** 12 April 2021
Accepted (Publication) 28 April 2021(30 April 2021)

***Corresponding Author : Do-Sik Yoo**

Tel: : +82-2-320-3048

E-mail: yoodosik@hongik.ac.kr

Ⅰ. 서 론

본 논문에서는 시계열 데이터를 처리할 수 있는LSTM (long-short term memory) 신경망을 이용하여 1차원 수직선상에서 움직이는 객체를 추적하는 성능을 칼만필터, 파티클필터와 비교한다. 칼만필터, 파티클필터와 같은 기법들은 객체추적에 최적화된 기법이지만 시스템 모델이 정확히 주어져야 좋은 성능을 낼 수 있다. 반면 LSTM 신경망의 경우 학습 데이터만 주어지면 안정적인 성능을 낼 수 있다. 우리는 시뮬레이션을 통해 LSTM 신경망 추적기법과 시스템 모델을 정확히 알고 있을 때와 잘못 알고 있을 때의 칼만필터, 파티클필터의 추적성능과 비교한다. 그 결과 LSTM 신경망의 경우 시스템 모델이 주어지지 않아도 학습 데이터만 주어지면 칼만필터, 파티클필터와 유사한 성능을 낼 수 있다는 것을 보인다.

객체추적은 어떠한 관측환경 하에서 객체의 위치 변화를 찾는 일련의 신호처리 과정이다. 객체추적은 현재시간의 정보만으로 관측하여 객체의 위치를 추정하는 것 외에도 데이터의 시계열적 상관관계를 이용하여 이전시간의 추적 결과를 이용할 수 있다. 따라서 객체추적기법 없이 시시각각 객체의 위치를 추정하는 것에 비해 추적기법을 사용하는 것이 사용할 수 있는 정보량이 많아 객체의 위치 정확도면에서 성능이 좋다.

객체추적 기법은 CCTV 영상을 필두로 한 컴퓨터비전 분야에서 활발하게 진행되어 자동으로 행인을 추적하는 등 스마트 CCTV의 기반기술이 되었다[1]. 컴퓨터비전 분야에서는 더 나아가 최근 화두가 되고 있는 자율주행 자동차에서 주변 상황을 관측 및 예측 할때에도 사용이 되고 있다[2]. 또한 추적기법은 컴퓨터비전 분야 외에도 위상배열 레이더 기반의 객체 추적 기법 등 다양한 분야에 활용되고 있다[3].

객체 추적 기법으로는 대표적으로 칼만필터가 있다[4]. 칼만필터는 객체의 위치를 선형으로 예측한 후, 현재시간의 관측값을 이용해 보정하는 대표적인 재귀필터이다. 칼만필터는 상태천이와 관측을 선형으로 모델링할 수 있고, 잡음이 정규분포인 경우에는 최소자승해를 도출할 수 있지만, 객체추적 시 시스템 모델을 알고 있어야 하며 비선형 시스템이나 잡음이 정규분포가 아닌 경우에는 객체추적성능이 떨어진다는 단점이 있다. 파티클필터 또한 추적기법에서 대표적으로 쓰이는 기법이다. 파티클필터 기법은 다수의 파티클들을 이용해 객체의 위치를 예측하고, 각각의 위치에서 확률밀도함수값을 계산하여 객체의 위치를 보정하는 시계열 몬테카를로 기법의 일종이다[5]. 파티클필터는 비선형시스템에도 적용이 가능하며, 잡음이 정규분포가 아니어도 객체추적이 가능하여 다양한 곳에 활용이 가능하다[6]. 하지만 시스템모델을 알고 있어야하며 정확한 추적을 위해 다수의 파티클들을 사용할 경우 연산량이 많아진다는 단점이 있다.

인공신경망은 생물의 신경망을 모방한 것으로 데이터 분류 기법으로 좋은 성능을 보이며, 최근에는 복잡한 구조의 딥러닝으로 발전하여 영상, 음성 및 텍스트 처리 등의 분야에 활용되고 있으며[7] 위상배열 레이더의 객체의 위치를 추정하는데 활용[3]하는 등으로 다양한 분야로 확장되고 있다. 특히 텐서플로우를 비롯한[8] 딥러닝 라이브러리로 시뮬레이션에 접근이 매우 쉬워졌으며 GPU를 활용한 병렬연산이 가능하여 빠른 속도로 학습이 가능하다. 또한 인공신경망은 피드백 루프를 이용해 시계열성 데이터를 처리할 수 있는 재귀신경망(recurrent neural network)을 이용하여 객체추적에도 활용되고 있는 상황이다[9]. 퍼셉트론 신경망에 피드백 루프만을 추가한 재귀신경망은 그 자체로도 시계열성 데이터를 처리할 수 있지만, 현재 관측한 시간과 이전 시간의 처리결과는 잘 반영되는 반면, 그 이전시간대들의 처리결과는 잘 반영되지 않는 장기의존성 문제(the problem of long-term dependencies)가 발생한다. 때문에 재귀신경망의 구조를 변경하여 이러한 문제를 해결할 수 있는 LSTM 신경망이 발표되어 사용되고 있다[10]. 신경망 기반의 객체추적은 학습데이터만 주어지면 학습이 가능하고, 시스템 모델이 주어지지 않아도 추적이 가능하다는 점에서 실제 환경에서 사용하기 좋다. 이러한 신경망 기반의 추적기법들은 학습 데이터만 충분히 주어진다면 좋은 성능을 낼 수 있기 때문에 기존의 추적 기법으로 추적하기 어려웠던 영상기반의 추적 기법에 많이 쓰이고 있다.

본 논문에서는 이러한 신경망 기반 추적기법 중에서도 LSTM의 성능을 중점적으로 살펴보기 위하여 1차원 추적문제를 설정해두고, 칼만필터, 파티클필터와의 성능을 비교해보려 한다. 더 나아가 칼만필터, 파티클필터를 이용해 추적할 때 시스템 모델을 정확히 모를 경우에 성능이 나빠지는 것도 살펴보아서 시스템모델이 주어지지 않아도 추적이 가능한 LSTM의 성능과 비교하여 실제 적용 시 얼마나 효과적인지를 보려한다. 이러한 결과들을 바탕으로 2차원 추적문제, 비선형추적문제 등 다양한 문제에 확장할 수 있는 기반을 마련하고 일반적인 추적기법으로써의 LSTM 기반 기법의 가능성을 살펴보려 한다.

본 논문의 구성은 다음과 같다. 2장에서는 시스템 모델에 대해 서술하고 3장에서는 본 논문에서 중점적으로 살펴볼 LSTM에 대해 자세히 기술하여 장기의존성 문제를 어떻게 해결할 수 있는지를 살펴본다. 4장에서는 시뮬레이션을 통해 칼만필터, 파티클필터와 성능을 비교하여 LSTM 기반 기법의 성능을 검증하고, 5장에서는 본 논문의 결론을 맺는다.

Ⅱ. 시스템 모델

본 논문에서는 1차원 수직선상에서 움직이는 객체의 위치추적을 다룬다. 해당 객체는 0에서 출발하여 평균속도 +1로 움직이며 이때 초기위치와 시간 t의 상태천이식은 식 (1)과 같다.

$$x_0 = 0, \quad x_{t+1} = x_t + 1 + e_t \tag{1}$$

이를 행렬을 이용하여 식 (2), (3)과 같이 나타낼 수 있다.

$$\begin{pmatrix} x_0 \\ \vdots \\ x_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix},\ \begin{pmatrix} x_{t+1} \\ \vdots \\ x_{t+1} \end{pmatrix} = \begin{pmatrix} 1\,1 \\ 0\,1 \end{pmatrix} \begin{pmatrix} x_t \\ \vdots \\ x_t \end{pmatrix} + \begin{pmatrix} e_t \\ 0 \end{pmatrix} \tag{2}$$

$$X_{t+1} = AX_t + N_t^x,\ X_t = \begin{pmatrix} x_t \\ \vdots \\ x_t \end{pmatrix}, A = \begin{pmatrix} 1\,1 \\ 0\,1 \end{pmatrix}, N_t^x = \begin{pmatrix} e_t \\ 0 \end{pmatrix} \tag{3}$$

이러한 객체들을 관측한 관측식은 식 (4)와 같으며, 이를 행렬로 나타내면 식 (5)와 같다.

$$y_t = x_t + \eta_t \tag{4}$$

$$Y_t = HX_t + N_t^y, Y_t = (y_t), H = (1\,0), N_t^y = (\eta_t) \tag{5}$$

여기서, e_t와 η_t는 각각 상태천이잡음, 관측잡음이며, 각각의 잡음들이 정규분포(normal distribution)인 경우 식 (6)-(8)을 만족하는 평균이 0인 랜덤프로세스이다.

$$E[e_t e_\tau] = \sigma_x^2 \delta_{t\tau} \tag{6}$$

$$E[\eta_t \eta_\tau] = \sigma_y^2 \delta_{t\tau} \tag{7}$$

$$E[e_t \eta_\tau] = 0 \tag{8}$$

여기서 $\delta_{t\tau}$는 디렉-델타 함수이며 상태천이잡음 e_t와 관측잡음 η_t는 시간 t에서 각각 평균이 0이고, 분산이 σ^2인 정규분포, 그 외에 라플라스분포(laplace distribution), 지수분포(exponential distribution), 균등분포(uniform distribution)인 랜덤변수로, 다양한 분포의 잡음이 있는 기본적인 추적 문제를 다룬다. 또한 시간 t는 초기값 0을 제외한 1-50까지 설정하였으며, 이때 객체의 위치 x_t의 평균값은 $1,2,3,\cdots,49,50$이 된다.

Ⅲ. LSTM 신경망

재귀신경망은 시간 t의 데이터를 처리하는데 t-1때의 처리결과를 이용하는 인공신경망이다. 시계열적 데이터를 처리할 수 있어 음성, 영상, 문자열 처리 등 다양한 분야에 사용되고 있다. 그러나 일반적인 재귀신경망의 경우 시간 t의 데이터 처리를 위해 t-1때의 처리결과는 많이 반영되지만, t-2, t-3 때와 같이 그 이전의 처리결과는 희석되어 반영되는 장기 의존성 문제가 발생한다. 이러한 문제로 인해 일반적인 재귀신경망은 긴 시간의 데이터를 처리하기 어렵다.

LSTM는 이러한 장기 의존성 문제를 해결하기 위하여, 은닉상태(hidden state)와 별도로 셀 상태(cell state)를 신경망 내부에 두었다. 해당 셀 상태는 입력받은 데이터에서 장기적으로 사용할 부분과 단기적으로 사용할 부분을 저장해두고 필요없는 부분은 삭제한다. LSTM 신경망의 구조는 그림 1과 같다.

LSTM 신경망에서는 망각게이트(forget gate), 입력게이트(input gate), 출력게이트(output gate)의 세가지 게이트가 있으며 이를 이용해 셀 상태에 필요한 정보를 저장하고 필요없는 정보를 제거하는 갱신과정을 거치며, 셀 상태의 저장된 정보들과 이전 상태, 입력값을 바탕으로 출력값을 도출한다.

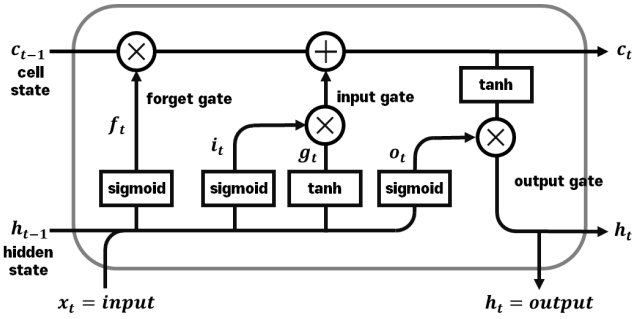

그림 1. LSTM 신경망의 구조
Fig. 1. The structure of LSTM neural network.

망각게이트에서는 이전상태 h_{t-1}와 입력 x_t를 결합하여 셀 상태에서 삭제할 부분을 결정한다. 예를들어 망각게이트의 출력벡터가 2, 5번째 값이 0에 가깝다면 셀 상태의 2, 5번째 값을 0으로 만들어 정보를 삭제한다. 망각게이트의 출력 f_t 수식은 식 (9)와 같다.

$$f_t = \text{sigmoid}(W_x^f x_t + W_h^f h_{t-1} + b_f) \tag{9}$$

여기서 W_x^f, W_h^f는 망각게이트의 가중치 행렬이며, b_f는 바이어스 벡터이다. 또한 sigmoid(•)는 시그모이드(sigmoid) 활성함수를 의미한다.

입력게이트에서는 이전상태 h_{t-1}와 입력 x_t를 결합하여 셀 상태에 저장할 정보를 생성하는 게이트이다. 입력게이트에서는 두가지 벡터를 이용하는데, 우선 g_t 벡터로 식 (10)과 같이 셀 상태에 저장할 정보를 생성한다. g_t벡터는 쌍곡탄젠트(hyperbolic tangent) 활성함수를 이용하기 때문에 각 성분들이 −1에서 1사이의 값을 가진다. 이후 i_t 벡터로 식 (11)과 같이 이를 얼마나 반영할지를 결정한다. i_t는 시그모이드 활성함수를 이용하여 각 성분이 0에서 1사이이기 때문에 g_t의 정보를 셀 상태에 얼마나 적용할지를 결정할 수 있다.

$$g_t = \tanh(W_x^g x_t + W_h^g h_{t-1} + b_g) \tag{10}$$

$$i_t = \text{sigmoid}(W_x^i x_t + W_h^i h_{t-1} + b_i) \tag{11}$$

여기서 W_x^i, W_h^i, W_x^g, W_h^g 는 각각 가중치 행렬이며, b_i, b_g는 바이어스 벡터이다. 또한 tanh(•)는 쌍곡탄젠트 활성함수이다.

셀 상태 갱신 과정에서는 망각게이트, 입력게이트의 출력벡터들을 바탕으로 셀 상태에서 필요없는 정보를 삭제하고, 새로 입력할 정보들을 저장하는 역할을 한다. 셀 상태 갱신과정의 수식은 식 (12)와 같다.

$$c_t = \text{sigmoid}(f_t \circ c_{t-1} + i_t \circ g_t) \quad (12)$$

여기서 $\circ$는 벡터들 각각의 원소들을 곱하는 아다마르 곱(hadamard product)이다.

마지막 출력게이트에서는 이전상태 h_{t-1}와 입력 x_t, 셀 상태를 이용하여 출력값을 결정하는 단계이며, 이 출력값이 바로 다음 시간 t+1에서 이전 상태가 된다. 출력게이트에서는 우선 식 (13)과 같이 h_{t-1}, x_t를 결합하여 o_t 벡터를 만든 후, 이 o_t 벡터와 셀 상태 c_t로 식 (14)와 같이 최종 출력값을 도출한다.

$$o_t = \text{sigmoid}(W_x^o x_t + W_h^o h_{t-1} + b_o) \quad (10)$$

$$h_t = \tanh(c_t) \circ o_t \quad (11)$$

2장에서 언급한 시스템 모델에 LSTM을 적용한다면, 입력벡터는 관측값인 1차원 벡터 Y_t 이고, 출력벡터는 객체의 위치와 속도의 추정값인 2차원 벡터 $\hat{X}_t$이 되어야 한다. 만일 LSTM 내부 셀의 크기를 n이라 하면 가중치 행렬 $W_x^f, W_x^i, W_x^g, W_x^o$의 크기는 각각 n×1, $W_h^f, W_h^i, W_h^g, W_h^o$의 크기는 n×n이 된다. 즉, 출력벡터의 차원이 n차원 벡터가 되므로 LSTM의 출력벡터를 n×2 크기의 가중치 행렬을 갖는 퍼셉트론 층에 통과시켜 최종 출력값 $\hat{X}_t$을 얻는다.

Ⅳ. 성능검증

LSTM 신경망 기반 추적 기법의 성능 검증을 위해 상태천이 잡음 e_t와 관측잡음 η_t가 각각 정규분포, 라플라스분포, 지수분포, 균등분포의 네 가지인 상황을 가정하였으며, (상태천이 분산, 관측분산)은 (1, 1), (1, 4)의 두가지 경우로 설정하였다. 또한 시뮬레이션은 객체가 t=0일 때 0에서 출발하여 t=50까지 이동하는 것으로 진행하였고 객체의 위치 추정은 t=1부터 50까지 진행하였다. 추적 결과는 객체의 실제 궤적과 추적한 궤적 차이의 rmse(root mean square error)로 설정하였다.

LSTM의 경우 퍼셉트론의 수를 128개로 설정해 두었으며 이 때 학습이 필요한 변수는 67,330개이다. 각 신경망들은 50만개의 학습 데이터를 이용해 AdamOptimizer로 적절한 성능이 될 때까지 epoch를 반복하는 학습인 Early stop 학습을 진행하였으며, 10만개의 데이터를 따로 생성하여 추적 성능을 검증하였다. 또한 LSTM의 성능을 객관적으로 검증하기 위하여, 칼만필터, 5000개의 파티클을 사용하는 파티클필터로도 추적을 진행해 비교하였으며 LSTM과 동일한 10만개의 데이터로 검증하였다.

아래의 표 1과 표 2에서는 (천이잡음분산, 관측잡음분산)은 각각 (1, 1), (1, 4)인 상황에서 칼만필터와 파티클필터가 시스템 모델을 정확히 알고있다고 가정하고 진행하였다. 칼만필터의 경우 천이잡음분산과 관측잡음분산, 파티클필터의 경우 추가로 잡음들의 확률밀도함수까지 알고 있다고 가정하였다. LSTM의 경우 각각의 환경의 학습데이터로 학습을 완료한 후 추적을 진행하였다.

표 1. (천이잡음분산, 관측잡음분산)이 (1, 1)인 경우의 각 알고리듬 별 rmse 성능

Table 1. The rmse result of each algorithms when (transition noise variance, observer noise variance) is (1, 1).

Noise distribution	Kalman filter	Particle filter	LSTM
Normal	0.7848	0.7847	0.7863
Laplace	0.7848	0.7803	0.7822
Exponential	0.7851	0.7620	0.7621
Uniform	0.7848	0.7818	0.7843

표 2. (천이잡음분산, 관측잡음분산)이 (1, 4)인 경우의 각 알고리듬 별 rmse 성능

Table 2. The rmse result of each algorithms when (transition noise variance, observer noise variance) is (1, 4).

Noise distribution	Kalman filter	Particle filter	LSTM
Normal	1.2424	1.2401	1.2422
Laplace	1.2421	1.1942	1.1967
Exponential	1.2414	1.0906	1.0935
Uniform	1.2422	1.1902	1.1931

각 필터들이 시스템 모델을 잘 알고있을 경우 정규분포 잡음에서는 칼만필터와 파티클필터의 성능이 각각 0.7848, 0.7847로 가장 좋고, 라플라스, 지수, 균등분포에서는 파티클필터의 성능이 0.7803, 0.7620, 0.7818로 가장 좋았다. 파티클필터의 경우 정규분포가 아닌 관측환경에서도 각각의 환경에 맞는 분포로 추적을 할 수 있기 때문에 정규분포 이외에서는 좋은 성능을 보인다. LSTM은 정규분포에서는 칼만필터, 파티클필터에 비해 rmse가 0.7863으로 그리 좋지 않았지만 그 외의 분포에서는 칼만필터와 파티클필터의 중간정도의 성능을 보였다.

다음 시뮬레이션에서는 칼만필터, 파티클필터가 시스템 모델을 잘못 알고 있을 경우의 추적성능을 LSTM의 추적성능과 비교하였다. 표 3에서는 천이잡음분산, 관측잡음분산이 각각 1, 1인데, 각 필터들이 천이잡음분산, 관측잡음분산을 1, 4로 잘못 알고있을 때의 결과이며, 표 4에서는 천이잡음분산, 관측잡음분산이 각가 1, 4인데, 각 필터들이 천이잡음분산, 관측잡음분산을 1, 1로 잘못 알고있을 때의 결과이다. 비교를 위해 시스템 모델을 몰라도 학습 데이터만으로 학습 후 추적이 가능한 LSTM의 추적 결과를 표 3과 표 4에도 나타내었다.

칼만필터와 파티클필터의 경우, 시스템 모델에 오차가 있는

표 3. (천이잡음분산, 관측잡음분산)이 (1, 1)인 것을 (1, 4)로 알고있는 경우의 각 알고리듬 별 rmse 성능
Table 3. The rmse result of each algorithms when (transient noise variance, observed noise variance) is (1, 1), but misunderstood as (1, 4).

Noise distribution	Kalman filter	Particle filter	LSTM
Normal	0.9079	0.9123	0.7863
Laplace	0.9076	0.8715	0.7822
Exponential	0.9087	0.9658	0.7621
Uniform	0.9077	0.9799	0.7843

표 4. (천이잡음분산, 관측잡음분산)이 (1, 4)인 것을 (1, 1)로 알고있는 경우의 각 알고리듬 별rmse 성능
Table 4. The rmse result of each algorithms when (transient noise variance, observed noise variance) is (1, 4), but misunderstood as (1, 1).

Noise distribution	Kalman filter	Particle filter	LSTM
Normal	1.5257	1.3913	1.2422
Laplace	1.5252	1.3502	1.1967
Exponential	1.5243	1.3425	1.0935
Uniform	1.5259	1.6186	1.1931

경우 추적 성능이 크게 악화되는 것을 볼 수 있다. 특히 표 3과 같이 시스템 모델의 관측잡음분산이 작은 것을 크다고 아는 상태에서 추적을 진행한 것 보다, 표 4와 같이 관측잡음분산이 큰 것을 작다고 아는 상태에서 성능이 더 악화된 것을 볼 수있다. 특히 파티클필터에서 지수분포일때와 균등분포일 때 성능 저하가 더 심한데, 이는 시스템 모델의 관측잡음분산 4를 1로 알고 추적을 진행할 때, 파티클들의 가중치를 업데이트하는 과정에서 확률분포함수의 0인 영역의 차이가 크기 때문이다. 이 때문에 실제 확률분포의 모델링 차이가 커서 큰 오차가 나게 된다. 반면 LSTM은 시스템 모델을 모르는 상태에서 학습만으로 시스템 모델을 아는 경우의 칼만필터, 파티클필터와 유사한 성능을 낼 수 있다. 실제 환경에서 시스템 모델이 급변하거나, 측정하기 어려운 경우에는 LSTM을 이용한 추적 알고리듬이 안정적인 성능을 낼 수 있는 것을 볼 수 있다.

Ⅴ. 결 론

본 논문에서 우리는 LSTM 신경망 기반의 추적 기법과 칼만필터, 파티클필터의 성능을 비교하여 그 성능을 검증하였다. 그 결과 시스템 모델을 잘 알고 있는 경우에는 전반적으로 파티클필터의 성능이 우수하였고 LSTM 신경망 기반의 추적 기법의 성능은 칼만필터의 성능보다는 우수했으나 파티클필터의 성능보다는 떨어졌다. 그러나 시스템 모델을 정확하게 알지 못하는 경우에는 칼만필터, 파티클필터의 성능이 매우 감소하였다. LSTM 신경망 기반의 추적 기법의 경우 시스템 모델이 주어지지 않아도 학습 데이터만으로 신경망을 학습한 후 추적이 가능하기 때문에 학습 데이터만 충분히 주어진다면 안정적인 성능을 낼 수 있다.

추후 우리는 이러한 결과를 바탕으로 2차원 혹은 비선형 시스템에서의 추적에 LSTM 신경망 기반의 추적 기법의 성능을 비교하는 것으로 연구를 확장할 계획이며, 보다 다양한 환경에 적용가능한 추적 기법으로 발전시킬 계획이다.

Acknowledgments

이 연구는 정부(교육부)의 재원으로 한국연구재단 기본연구지원사업(과제번호: NRF-2018R1D1A1B07045719)의 지원을 받아 수행되었습니다.

References

[1] A. Gautam, and S. Singh, (2019, December). “Trends in video object tracking in surveillance: a survey,” in *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, Palladam: India, pp. 729-733, Dec. 2019.

[2] A. Kuramoto, M. A. Aldibaja, R. Yanase, J. Kameyama, K. Yoneda, and N. Suganuma, “Mono-camera based 3d object tracking strategy for autonomous vehicles,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Changshu: China, pp. 459-464, Jun. 2018.

[3] S. B. Park, and D. S. Yoo, “Three stage neural networks for direction of arrival estimation,” *The Journal of Korea Navigation Institute*, Vol. 24, No. 1, pp. 47-52. Feb. 2020.

[4] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Basic Engineering*. Vol. 82, No. 1, pp. 35–45, 1960.

[5] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian bayesian tracking,” *IEEE Transactions on signal processing*, Vol. 50, No. 2, pp. 174-188, 2002.

[6] M. Speekenbrink, “A tutorial on particle filters,” *Journal of Mathematical Psychology*, Vol. 73, pp. 140-152, 2016.

[7] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, “A survey of deep neural network architectures and their applications,” *Neurocomputing*, Vol. 234, pp. 11-26, 2017.

[8] M. Abadi, A. Agarwal, B. Paul, et al. (2016, Mar). Tensorflow: large-scale machine learning on heterogeneous

distributed systems. Available: https://arxiv.org/abs/1603.04467.

[9] M. Fiaz, A. Mahmood, S. Javed, and S. K. Jung, "Handcrafted and deep trackers: recent visual object tracking approaches and trends," *ACM Computing Surveys (CSUR)*, Vol. 52, No. 2, pp.1-44, 2019.

[10] S. Hochreiter, and J. Schmidhuber, "Long short-term memory," *Neural Computation*, Vol. 9, No. 8, pp. 1735-1780, 1997.

박 선 배 (Sun-Bae Park)

2015년 2월 : 홍익대학교 전자·전기공학부 (공학사)
2015년 3월 ~ 2017년 2월 : 홍익대학교 대학원 전자·정보·통신공학과 (공학석사)
2017년 3월 ~ 현재 : 홍익대학교 대학원 전자전기공학과 박사과정
※관심분야 : 머신러닝, 신호처리, 패턴인식, 레이더신호처리, 영상처리 및 필터링 이론

유 도 식 (Do-Sik Yoo)

2002년 2월 : 미시간대학교 전자컴퓨터공학과 (공학박사)
2006년 9월 ~ 2011년 3월 : 홍익대학교 전자전기공학부 조교수
2011년 4월 ~ 2016년 3월 : 홍익대학교 전자전기공학부 부교수
2016년 4월 ~ 현재 : 홍익대학교 전자전기공학부 교수
※관심분야 : 머신러닝, 통신 및 신호처리, 정보이론, 어레이신호처리 등