

<http://dx.doi.org/10.17703/JCCT.2023.9.4.563>

JCCT 2023-7-68

디지털 FIR 필터와 Deep Learning을 이용한 ECG 신호 예측 및 경과시간

Predicton and Elapsed time of ECG Signal Using Digital FIR Filter and Deep Learning

윤의중*

Uei-Joong Yoon*

요약 심전도(electrocardiogram, ECG)는 심박동의 속도와 규칙성, 심실의 크기와 위치, 심장 손상 여부를 측정하는데 사용되며, 모든 심장질환의 원인을 찾아낼 수 있다. ECG-KIT를 이용하여 획득한 ECG 신호는 ECG 신호에 잡음을 포함하기 때문에 딥러닝에 적용하기 위해서는 ECG 신호에서 잡음을 제거해야만 한다. 본 논문에서는, ECG 신호에 포함된 잡음은 Digital FIR 해밍 창함수를 이용한 저역통과 필터를 사용하여 제거하였다. LSTM의 딥러닝 모델을 사용하여 3가지 활성화 함수인 sigmoid(), ReLU(), tanh()에 대한 성능 평가를 비교했을 때, 오차가 가장 작은 활성화 함수는 tanh() 함수임을 확인하였고, batch size가 작은 경우가 큰 경우보다 시간이 많이 소요되었다. 또한 GRU 모델의 성능 평가의 결과가 LSTM 모델보다 우수한 것을 확인하였다.

주요어 : 심전도, 디지털 FIR 저역 통과 필터, 해밍 윈도우, 딥러닝, LSTM, GRU

Abstract ECG(electrocardiogram) is used to measure the rate and regularity of heartbeats, as well as the size and position of the chambers, the presence of any damage to the heart, and the cause of all heart diseases can be found. Because the ECG signal obtained using the ECG-KIT includes noise in the ECG signal, noise must be removed from the ECG signal to apply to the deep learning. In this paper, Noise included in the ECG signal was removed by using a lowpass filter of the Digital FIR Hamming window function. When the performance evaluation of the three activation functions, sigmoid(), ReLU(), and tanh() functions, which was confirmed that the activation function with the smallest error was the tanh() function, the elapsed time was longer when the batch size was small than large. Also, it was confirmed that result of the performance evaluation for the GRU model was superior to that of the LSTM model.

Key words : ECG, Digital FIR lowpass filter, Hamming window, Deep learning, LSTM, GRU

1. 서론

심전도(Electrocardiography: ECG)란 심장근육의 활동이 일어날 때 발생하는 전기적 흥분이 신체의 표면(1mV)에 도달하는 것을 증폭 후에 파형으로 기록한 것

을 말한다. ECG는 일반적으로 심근의 이상이나 심실 비대, 심장 내의 전기적 흥분 부위의 이동, 부정맥 등을 임상적으로 검사하는데 중요한 수단이며, ECG에서 측정된 P-QRS-T의 주기와 진폭을 통해 심장과 관련된 질병에 대한 정보를 얻을 수 있다[1, 2].

*정회원, 가천대학교 의공학과 교수 (단독저자)
접수일: 2023년 5월 27일, 수정완료일: 2023년 6월 16일
게재확정일: 2023년 7월 3일

Received: May 27, 2023 / Revised: June 16, 2023
Accepted: July 3, 2023

*Corresponding Author: ujyoon@gachon.ac.kr
Dept. of Biomedical Engineering, Gachon University, Korea

본 논문에서는 ECG 신호를 획득하기 위하여 Physiolab사의 ECG-KIT를 사용하였으며 [3], ECG-KIT는 기본적으로 우측회로, 고역통과 필터, 노치필터, 저역통과 필터, 증폭단 회로로 구성된다. 획득된 ECG 신호들이 잡음을 포함하고 있는데, 딥러닝 프로그램을 사용하기 전에 디지털 FIR 저역통과 필터를 이용하여 ECG 잡음을 제거하고자 한다 [4]. 물론, 잡음이 없는 신호를 획득하였다면 저역통과 필터를 설계하는 과정은 생략된다.

ECG 잡음을 제거한 ECG 데이터는 딥러닝인 LSTM (Long Short Term Memory)과 GRU(Gated Recurrent Unit) 모델에 적용한다. LSTM은 장단기 메모리 셀이라고 하며, RNN에 입력 게이트, 망각 게이트 그리고 출력 게이트를 도입하여 RNN의 장기 의존성의 문제를 해결한 것으로 알려져 있다 [5, 6]. GRU는 2014년에 조경현 등의 논문에서 제안되었는데, 세 개였던 게이트의 수를 두 개의 게이트로 줄이면서 연산 속도가 향상되었다고 알려져 있다 [7]. 최근에 레이더의 고장방지를 위한 공기전조기의 이상탐지 모델 구축 사례에 대한 딥러닝 LSTM 모델에 대해 연구하였고 [8], LSTM과 GRU 모델을 이용하여 달리기 운동을 수행하는 사용자의 심박수를 예측하였으며 [9], 성능평가 지표로 설정한 RMSE가 최저인 모델인 LSTM 모델을 슬라이딩 윈도우 알고리즘의 학습 방법을 제안하였다 [10].

본 논문에서는 획득한 ECG 신호가 잡음을 포함하고 있을 때, 딥러닝 프로그램을 사용하기 전에 전처리 단계로, 잡음이 제거된 ECG 데이터를 생성하기 위해 디지털 FIR 저역통과 필터를 사용하고, 잡음이 제거된 ECG 데이터를 이용하여 딥러닝 LSTM 및 GRU 모델에 적용하여 ECG 데이터에 대한 성능평가(Performance evaluation) 및 경과시간(Elapsed time)에 대해 논하고자 한다.

II. ECG 신호 획득 및 잡음 제거

1. 사지유도법과 ECG 신호 측정 [1]

심전도 ECG 측정 방법으로 사지 유도법을 주로 사용한다 [1]. 사지 유도법이란 인체표면에 전극을 부착하여 심전도를 측정하는 방법으로, 두 개 이상의 전극을 사용하며 필요 기준이 되는 전극을 기준전극(왼팔), 측정하고자 하는 전극을 활동 전극(오른팔), 공통 접지로

왼쪽 다리를 이용한다. 그림 1에 나타난 전형적인 ECG 신호 파형은 P-QRS-T 파형으로 QRS파 중에서 R-peak과는 좌심실에서 대동맥으로 피를 순환시키기 위해 심실이 수축할 때 발생하는 파형으로 R-파의 간격을 이용하여 분당 맥박수를 측정한다 [1, 2, 3].

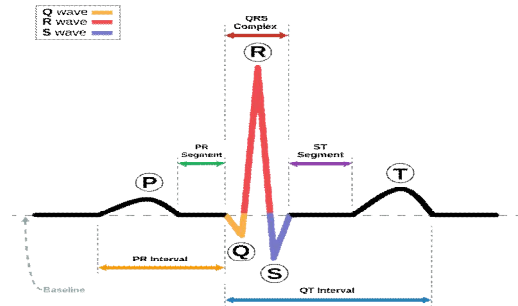


그림 1. ECG에 대한 P-QRS-T 심장 파형
Figure 1. Heart wave of P-QRS-T for ECG

2. 디지털 FIR(Finite impulse response)필터 [4]

딥러닝 프로그램을 사용하기 전에 전처리 단계로, ECG의 잡음을 제거하기 위한 디지털 FIR 저역통과 필터를 이용하여 설계한다.

FIR 필터는 식 (1)과 같이 차분방정식으로 차분방정식에서 비순환 부분만 있는 형태이며, z-변환 후의 시스템 함수는 식 (2)와 같다 [4].

$$y(n) = \sum_{k=0}^{M-1} b_k x(n-(M-1)) \quad (1)$$

$$= b_0 + b_1 x(n-1) + \dots + b_{M-1} (n-(M-1))$$

$$y(n) = \sum_{n=0}^{M-1} b_n z^{-k} = b_0 + \dots + b_{M-1} z^{-(M-1)} \quad (2)$$

FIR 필터의 계수 b_k , FIR 필터의 차수는 M 이다. FIR 필터는 순환 부분이 없어서 안정적이며, IIR(Infinite impulse response)에 비해서 상대적으로 단순하게 설계할 수 있으며, 차분방정식의 계수 b_n 은 임펄스 응답 $h[n]$ 과 창 함수 $w[n]$ 의 곱으로 표현된다.

$$b_n = h[n] \times w[n], 0 \leq n \leq M-1 \quad (3)$$

식 (3)에서 $h[n]$ 은 임펄스 응답계수, $w[n]$ 은 창 함수(Window function)로서 다양한 창 함수를 이용하여 필터를 설계하며, 디지털 차단주파수가 Ω_c 인 저역통과

필터의 임펄스 응답은 식 (4)로 sinc() 함수로 표현된다.

$$h[n] = \Omega_c \text{sinc}(2n\Omega_c), 0 \leq n \leq M-1 \quad (4)$$

식 (4)에서 Ω_c 는 디지털 주파수, M은 필터의 길이를 의미하며, 식 (5)는 본 논문에서 사용한 해밍 창 (Hamming window) 함수이다 [4].

$$w[n] = 0.54 + 0.46 \cos\left(\frac{2n\pi}{M-1}\right), 0 \leq n \leq M-1 \quad (5)$$

결국, 식 (4)에 식 (5)를 곱하여 식 (3)의 계수를 구하면, 식 (1)을 z-변환하여 잡음을 제거하기 위한 식 (2)가 완성된다. 또한 여러 가지 창 함수에 대한 수식들이 존재하는데 참고 문헌을 참고하면 된다 [4].

III. 딥러닝 알고리즘

1. LSTM(Long short term memory) 모델 [5, 6]

딥러닝 LSTM(Long short term memory) 모델을 장단기 메모리 셀이라고도 하는데, LSTM의 구조는 그림 2에 나타냈으며, RNN(Recurrent Neural Network)의 단점을 보완해서 나온 모델로서 [5], 기존의 RNN이 출력과 먼 위치에 있는 정보를 기억할 수 없다는 단점이 있었기 때문에 은닉층의 메모리 셀에 입력 게이트 (Input gate, i_t), 망각 게이트 (Forget gate, f_t), 출력 게이트 (Output gate, o_t)를 추가하여 불필요한 데이터를 지우고 중요한 데이터를 결정하는 역할을 한다 [6].

LSTM 셀의 장기상태와 단기 상태, 그리고 하나의 샘플에 대해 단계별로 출력을 계산하는 순서를 보여주는 식으로 다음과 같이 표현할 수 있다.

$$i_{(t)} = \sigma(W_{xi}^T x_{(t)} + W_{hi}^T h_{(t-1)} + b_i) \quad (6)$$

$$f_{(t)} = \sigma(W_{xf}^T x_{(t)} + W_{hf}^T h_{(t-1)} + b_f) \quad (7)$$

$$o_{(t)} = \sigma(W_{xo}^T x_{(t)} + W_{ho}^T h_{(t-1)} + b_o) \quad (8)$$

$$g_{(t)} = \tanh(W_{xg}^T x_{(t)} + W_{hg}^T h_{(t-1)} + b_g) \quad (9)$$

$$c_{(t)} = f_{(t)} \otimes c_{t-1} + i_{(t)} \otimes g_{(t)} \quad (10)$$

$$h_{(t)} = h_t = o_{(t)} \otimes \tanh(c_{(t)}) \quad (11)$$

식 (6)부터 (11)까지에서 \otimes 는 원소별 곱셈을 의미하며, $W_{xi}, W_{xf}, W_{xo}, W_{xg}$ 는 입력 벡터 $x_{(t)}$ 에 연결된 4개 층의 가중치 행렬, $W_{hi}, W_{hf}, W_{ho}, W_{hg}$ 는 이전 상태의 $h_{(t-1)}$ 에 연결된 4개 층의 가중치 행렬이며, b_i, b_f, b_o, b_g 는 4개 층의 편향을 의미한다.

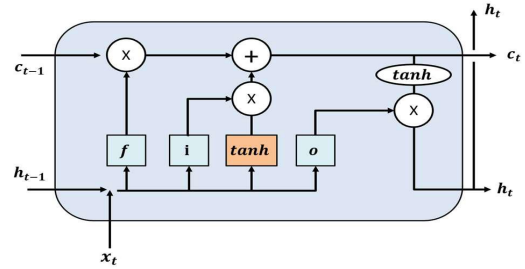


그림 2. LSTM의 구조
 Figure 2. Structure of LSTM

2. GRU(Gated Recurrent Unit) 모델 [7, 9]

그림 3은 GRU의 구조를 나타낸 것으로, GRU는 2014년에 조경현 등의 논문에서 제안되었다 [7, 9]. 이 논문에서 입력 게이트와 망각 게이트를 업데이트 게이트 하나로 융합하였고, 출력 게이트는 리셋 게이트로 대체했다. 즉, 세 개였던 게이트의 수가 두 개로 변화되면서 연산에 필요한 매개 변수의 수가 줄어들고 연산 속도가 향상되었으며 다음 그림 3과 같이 표현할 수 있다.

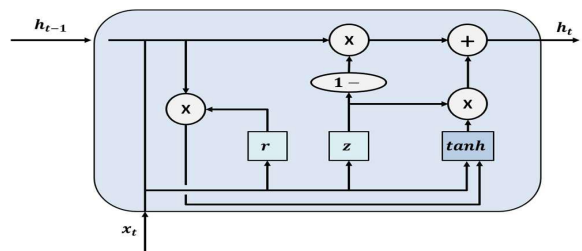


그림 3. GRU의 구조
 Figure 3. Structure of GRU

$$z_{(t)} = \sigma(W_{xz}^T x_{(t)} + W_{hz}^T h_{(t-1)} + b_z) \quad (12)$$

$$r_{(t)} = \sigma(W_{xr}^T x_{(t)} + W_{hr}^T h_{(t-1)} + b_r) \quad (13)$$

$$g_{(t)} = \tanh(W_{xg}^T x_{(t)} + W_{hg}^T (r_{(t)} \otimes h_{(t-1)}) + b_g) \quad (14)$$

$$h_{(t)} = z_{(t)} \otimes h_{(t-1)} + (1 - z_{(t)}) \otimes g_{(t)} \quad (15)$$

GRU는 LSTM보다 구조적으로 간단하고 연산속도에서 성능이 더 뛰어나다는 점에서 최근에 집중을 받고 있는 신경망으로 알려져 있다.

IV. 활성화 및 성능평가 함수

1. 활성화 함수(Activation function) [5]

활성화 함수는 입력된 데이터의 가중 합을 출력 신호로 변환하는 함수로서 세 가지의 활성화 함수들을 사용하여 LSTM 모델에 적용하여 성능평가를 비교하였다 [5]. 식 (16)의 sigmoid() 함수는 0과 1사이의 값만 가질 수 있도록 하는 비선형 함수이며, 식 (17)의 ReLU() 함수는 음수를 입력하면 0을 출력하고, 양수를 입력하면 입력 값을 그대로 반환하며, 식 (18)의 tanh() 함수는 -1과 1 사이의 값을 가지는 비선형 함수이다.

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (16)$$

$$\text{ReLU}(x) = \max(0, x) \quad (17)$$

$$\text{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (18)$$

2. 성능 평가 함수 및 최적화 함수 [5]

딥러닝 LSTM과 GRU 모델을 이용한 성능평가는 평균 제곱오차(Mean Squared Error, MSE)의 대해서 성능평가를 진행하였고, MSE의 값이 0이면 오차가 없는 것으로 평가한다. 본 논문에서는 수치 값이 작으므로 백분율인 %로 표시하였다 [5].

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \times 100(\%) \quad (19)$$

식 (19)에서 y_i 는 실제 값, \hat{y}_i 는 예측 값, n 은 전체 테스트 데이터의 개수이며, 첨자 i 는 데이터의 번호이다. 최적화 함수는 학습 데이터 셋을 이용하여 모델을 학습할 때 실제 결과와 예측한 결과의 차이를 줄여 최적화하는 역할을 하며, 가장 많이 사용하고 있는 adam() 함수를 사용하였다 [5, 9].

V. 디지털 필터설계 및 성능평가 실험

1. 디지털 IIR 저역통과 필터 설계

디지털 FIR 저역통과 필터는 표 1을 만족해야 한다 [4]. 본 논문의 설계사양은 샘플링 주파수 $S = 500$ Hz, 통과대역 주파수 $f_p = 15$ Hz, 저지대역 주파수 $f_s = 45$ Hz, 통과대역 감쇠 $A_p = 2$ dB, 저지대역 감쇠 $A_s = 50$ dB를 만족하는 조건들을 이용하여 차단 주파수 $f_c = 30$ Hz 인 디지털 저역통과 필터를 설계한다. 표 1에서 천이 대역폭 Ω_T 는 저지대역과 통과대역 디지털 주파수의 차이 즉, $\Omega_T = \Omega_s - \Omega_p$ 이며, 표 1에서 해밍 창의 길이 M 은 설계 기준에 따라 해밍 창의 길이는 55 이나, 성능이 우수한 필터를 만들기 위하여 65를 사용하였다.

표 1. 윈도우 함수들의 스펙트럼 특성

Table 1. Spectral characteristics of window functions

Window functions	천이 대역폭 Ω_T	최소 저지대역폭 감쇠 A_{uss} (dB)
Boxcar	$1.8 \pi/M$	21.7
Hann	$6.2 \pi/M$	44
Hamming	$6.6 \pi/M$	53
Blackman	$11 \pi/M$	75.3

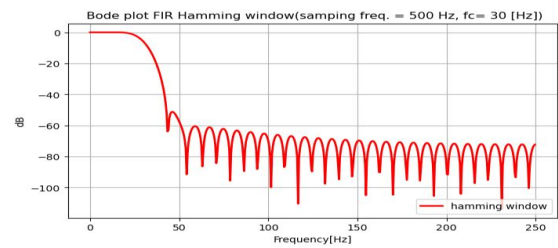


그림 4. 설계사양에 대한 해밍 창의 보드 선도

Figure 4. Bode plot of Hamming window for design specification

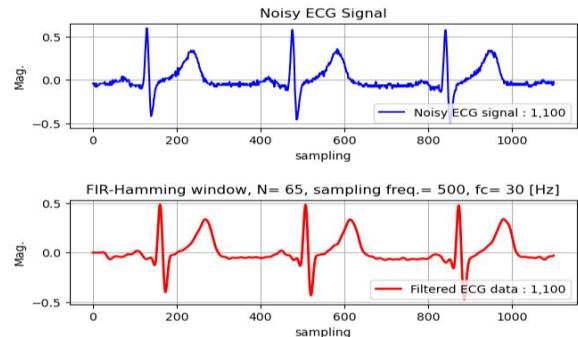


그림 5. 잡음 포함 ECG 및 잡음 제거 ECG 신호의 비교

Figure 5. Comparison of noisy ECG and filtered ECG signal

그림 4는 해밍 창을 이용한 저역통과 필터의 보드 선도(Bode plot)를 나타낸 것으로 표 1에서 나타난 바와 같이 약 53 dB의 감쇠를 나타낸다. 그림 5에서 그림(상)은 잡음을 가진 ECG 신호(Noisy ECG signal), 그림(하)는 창의 길이 N= 65를 사용하여 디지털 FIR 저역통과 필터를 통과시켜 잡음을 제거한 ECG 데이터이며, 생성된 ECG 데이터를 이용하여 딥러닝 모델의 소스 데이터로 사용된다 [4].

2. LSTM 및 GRU 모델의 성능 평가 실험

성능 평가를 위한 컴퓨터 사양은 64 비트, 16 Giga RAM의 성능을 가진 컴퓨터를 사용하였고, ECG 데이터 1,100 개를 7:3으로 training data : test data 의 비율로 나눈 후, 정규화 과정 단계를 실행한다.

LSTM 모델에 적용하기 위해 keras의 모델 도구 중 Sequential(), LSTM의 입력층에서는 은닉층의 노드 개수 unit= 64, 최종 출력층은 1개의 데이터가 출력되어야 하므로 Dense(1) 모델을 import하였고, 최적화 함수는 많이 사용하는 adam()을 사용한다.

표 2는 은닉층의 노드 개수 unit= 64, epoch= 30, batch size= 5일 때 활성화 함수들에 대해 MSE의 성능 평가를 비교하였으며, 성능평가 MSE를 %로 나타내면, sigmoid() 함수인 경우는 0.1169782 %, ReLU() 함수는 0.0720089 %, tanh() 함수는 0.0081644 %의 손실 값으로 정확도 측면에서는 tanh() 함수가 가장 우수하였다.

표 2. LSTM의 3가지 활성화 함수에 대한 성능 평가 비교
 Table 2. Comparison of performance evaluation for three activation functions of LSTM

activation functions	MSE (%)	elapsed time [sec]
sigmoid()	0.1169782	16.0077
ReLU()	0.0720089	15.8303
tanh()	0.0081644	17.8422

그림 6, 7, 8은 은닉층의 노드 개수 unit= 64, epoch= 30, batch size= 5일 때 활성화 함수 tanh() 를 적용한 실행결과를 보였으며, 그림 6, 7, 8에서 범례는 그림 1인 ECG 신호의 QRS과 중에서 급변하는 R-peak 부분의 100개의 실제 ECG 데이터를 확대하여 나타낸 실선, 예측 데이터는 '▲' 로 구분하여 표시하였으며, batch size= 1인 그림 8의 경우가 오차가 거의 없는 것을 확인할 수 있으며, 표 2에서도 언급하였다. 이는 batch

size가 작은 경우가 큰 경우보다 업데이트할 때 사용할 데이터의 개수가 많아지고 시간이 많이 소요되지만, 정확도는 우수하였다.

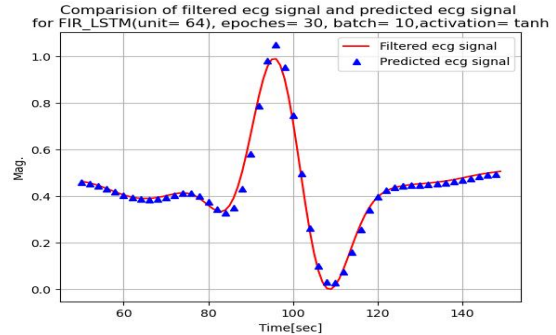


그림 6. batch size= 10 에 대한 예측 결과
 Figure 6. Prediction result for batch size= 10

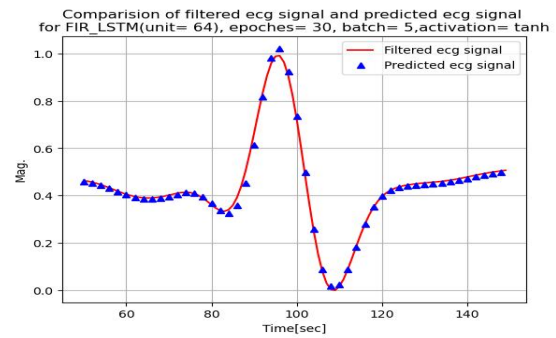


그림 7. batch size= 5 에 대한 예측 결과
 Figure 7. Prediction result for batch size= 5

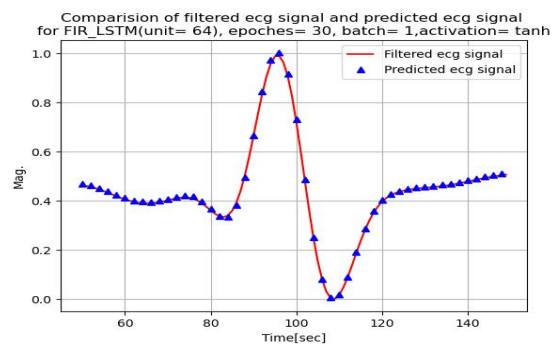


그림 8. batch size= 1 에 대한 예측 결과
 Figure 8. Prediction result for batch size= 1

표 3은 은닉층의 노드 개수 unit= 64, epoch= 30, 활성화 함수 tanh() 를 사용했을 때, 성능평가 MSE를 %로 나타내면, batch size= 10일 때 0.0236639 %, 5일 때

0.0081644 %, 1일 때 0.0035089 %의 손실율을 보여주었다. batch size가 1인 경우가 프로그램 경과시간은 많이 소요되지만, 정확도 측면에서 가장 우수하였다.

표 3. LSTM의 활성화 함수 tanh() 에 대한 성능 평가 비교
Table 3. Comparison of performance evaluation for activation tanh() functions of LSTM

batch size	MSE (%)	elapsed time [sec]
10	0.0236639	13.1500
5	0.0081644	17.8422
1	0.0035089	61.7588

표 4는 은닉층의 노드 개수 unit= 64, epoch= 30, batch size= 1, 활성화 함수 tanh() 를 이용한 LSTM과 GRU 모델에 대한 성능평가 결과로서, 경과시간은 별 차이가 없었으나, 성능평가 MSE를 %로 나타낸 결과 LSTM은 0.0035089 %, GRU는 0.0006022 % 로서, GRU가 LSTM 보다 오차가 작으므로 우수한 딥러닝 모델임을 확인할 수 있었다.

표 4. LSTM과 GRU 모델에 대한 성능 평가 비교
Table 4. Comparison of performance evaluation for LSTM and GRU model

Depp learning	LSTM	GRU
MSE (%)	0.0035089	0.0006022
elapsed time [sec]	61.7588	60.4350

VI. 결 론

본 논문에서는 획득한 ECG 신호에 잡음이 포함되어 있으므로, 샘플링 주파수 500 Hz, 차단 주파수 30 Hz 인 해밍 창 함수를 이용한 디지털 FIR 저역통과 필터를 설계하여 ECG의 잡음을 제거한 후, 딥러닝 LSTM, GRU 모델의 소스 데이터로 사용하였다.

사용된 LSTM 모델은 은닉층의 노드 개수 unit= 64, epoch=30, batch size=5를 사용했을 때, 3가지 활성화 함수 sigmoid(), ReLU(), tanh()에 따른 MSE의 성능평가 결과는 tanh() 함수가 가장 우수하였고, 동일한 조건에서 활성화 함수 tanh()를 사용하고 batch size= 10, 5, 1인 경우를 비교했을 때, batch size= 1인 경우가 10인 경우보다 경과시간은 증가하였지만 손실율(%)은 가장 작았다. 또한, 동일한 조건에서 LSTM과 GRU 모델에 대해 비교한 결과, 경과시간은 별 차이가 없지만, GRU 모델이 LSTM 보다 MSE의 평가 손실율(%)이 적어 성

능이 매우 우수하여 앞으로 기대되는 딥러닝 모델임을 확인할 수 있었다.

향후, 다른 생체 데이터에 포함된 잡음을 디지털 신호처리 방법으로 제거한 데이터들을 LSTM과 GRU 모델에 적용하기 위해 좀 더 많은 연구가 필요하다.

References

- [1] 최윤식, 임상심전도학(제5판), 교보문고, 2009
- [2] <https://a-fib.com/treatments-for-atrial-fibrillation/diagnostic-tests-2/the-ekg-signal/>
- [3] https://physiolab.en.ec2l.com/ECG-KIT Assembly-ECG-sensor-9570299_9570369.html
- [4] 하석운, 파이썬으로 배우는 디지털 신호처리, 생능출판사, 2020.
- [5] Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow(2nd ed.), O'Reilly Media, Inc, 2021.
- [6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, Nov. 1997. DOI: <https://doi.org/10.1162/neco.1997.9.8.1735>
- [7] Kyunghyun Cho et al., "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (2014): pp. 1724-1734, Sep. 2014. <https://doi.org/10.48550/arXiv.1406.1078>
- [8] Kun-Woo Kim, et al, "Anomaly Detection of Air Dryer for Radar based on Machine Learning Algorithms," Journal of the Korea Academia-Industrial cooperation Society(JKAIS), Vol. 24, No. 3 pp. 1-12, 2023.
- [9] Jin-Young Kim et al, "Prediction System of Running Heart Rate based on FitRec," The Journal of the Institute of Internet, Broadcasting and Communication(JIIBC), Vol. 22, No. 6, pp. 165-171, Dec. 31, 2022. <https://doi.org/10.7236/JIIBC.2022.22.6.165>
- [10] J. H. Kim et al, "Prediction of Battery Remaining Life based on Sliding Window-LSTM for Camouflage Therapeutic System," The Journal of Journal of KIIT(JKIIT), Vol. 21, No. 1, pp. 29-38, Jan. 31, 2023. <http://dx.doi.org/10.14801/jkiit.2023.21.1.29>