

<http://dx.doi.org/10.17703/JCCT.2023.9.4.611>

JCCT 2023-7-74

# 디지털 IIR Filter와 Deep Learning을 이용한 ECG 신호 예측을 위한 성능 평가

## Performance Evaluation for ECG Signal Prediction Using Digital IIR 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 신호에서 잡음은 Digital IIR Butterworth의 저역 통과 필터를 이용하여 제거하였다. LSTM의 딥러닝 모델을 사용하여 3가지 활성화 함수인 sigmoid(), ReLU(), tanh() 함수에 대한 성능 평가를 비교했을 때, 오차가 가장 작은 활성화 함수는 tanh() 함수임을 확인하였으며, 또한 LSTM과 GRU 모델에 대한 성능 평가와 경과 시간을 비교한 결과 GRU 모델이 LSTM 모델보다 우수한 것을 확인하였다.

**주요어** : 심전도, 디지털 IIR 저역통과 필터, 딥러닝, LSTM, GRU

**Abstract** ECG(electrocardiogram) is a test 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, the noise of the ECG signal was removed using the digital IIR Butterworth low-pass filter. When the performance evaluation of the three activation functions, sigmoid(), ReLU(), and tanh() functions, was compared using the deep learning model of LSTM, it was confirmed that the activation function with the smallest error was the tanh() function. Also, When the performance evaluation and elapsed time were compared for LSTM and GRU models, it was confirmed that the GRU model was superior to the LSTM model.

**Key words** : Electrocardiogram, Digital IIR lowpass filter, 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 것을 말한다. ECG는 일반적으로 심근의 이상이나 심실 비대, 심장 내의 전기적 흥분 부위의 이동, 부정맥, 흥분전도의 이상 등을 임상적으로 검사하는데 중요한 수

\*정희원, 가천대학교 의공학과 교수 (단독저자)  
접수일: 2023년 5월 15일, 수정완료일: 2023년 6월 10일  
게재확정일: 2023년 7월 5일

Received: May 15, 2023 / Revised: June 10, 2023

Accepted: July 5, 2023

\*Corresponding Author: [ujyoon@gachon.ac.kr](mailto:ujyoon@gachon.ac.kr)

Dept. of Biomedical Engineering, Gachon University, Korea

단이며, ECG에서 측정된 P-QRS-T의 주기와 진폭을 통해 심장과 관련된 질병에 대한 정보를 얻을 수 있다 [1, 2].

본 논문에서는 ECG 신호를 획득하기 위하여 Physio lab사의 ECG-KIT를 사용하였으며 [3], ECG-KIT는 우축회로, 고역통과 필터, 노치필터, 저역통과 필터 및 증폭회로로 구성된 회로를 이용하여 ECG 신호를 획득한다 [3].

본 논문에서는 딥러닝 적용 모델에 적용하기 전인 전단계에서 획득된 ECG 신호들은 잡음을 포함하고 있으므로, 잡음을 제거하기 위해서 디지털 저역통과 필터를 사용하여 ECG 잡음을 제거하기로 한다. 디지털 저역통과 필터의 설계하는 방법은 유한 임펄스 응답인 FIR(Finite Impulse Response)과 무한 임펄스 응답인 IIR(Infinite Impulse Response)방법이 있으며, 본 논문에서는 샘플링 주파수 600 Hz, 차단 주파수 60 Hz를 가지는 디지털 IIR 방식의 버터워스(Butterworth) 저역통과 필터를 설계하여 ECG 신호의 잡음을 제거하도록 설계하였다 [4].

ECG 잡음을 제거한 ECG 데이터는 딥러닝인 LSTM(Long Short Term Memory)과 GRU(Gated Recurrent Unit)모델에 적용한다. LSTM을 장단기 메모리 셀이라고 하며, RNN에 입력 게이트, 망각 게이트 그리고 출력 게이트를 도입하여 RNN 장기 의존성 문제를 해결한 것으로 알려져 있다 [5, 6]. 또한 GRU는 2014년에 조경현 등의 논문에서 제안되었는데 [7], 세 개였던 게이트의 수가 두 개로 변화하게 되면서 연산 속도가 향상된 것으로 알려져 있다. 최근에 레이더의 고장방지를 위한 공기건조기의 이상탐지 모델 구축 사례를 딥러닝 LSTM 모델에 대해 연구한 바 있으며 [8], LSTM과 GRU 모델을 이용하여 달리기 운동을 수행하는 사용자의 심박수를 예측하였고 [9], LSTM 모델을 슬라이딩 윈도우 알고리즘 기반으로 학습 데이터 수를 조절하는 방법을 제안한 바 있다 [10].

본 논문에서는 잡음을 포함하고 있는 ECG 신호에 대해 잡음이 제거된 ECG 데이터를 생성하기 위해 디지털 IIR 저역통과 필터를 설계하였으며, 잡음이 제거된 ECG 데이터를 이용하여 딥러닝인 LSTM과 GRU 모델에 적용하여 3가지 활성화 함수들에 대한 ECG 데이터의 신호를 예측하고 성능 평가(Performance evaluation) 및 경과시간(Elapsed time)에 대해 논하고자 한다.

## II. ECG 신호 획득 및 잡음 제거

### 1. 사지유도법과 ECG 신호 측정

심전도 ECG 측정 방법으로 사지 유도법을 주로 사용한다[1]. 사지 유도법이란 체표면에 전극을 부착하여 심전도를 측정하는 방법이다. 두 개 이상의 전극을 사용하며 필요기준이 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]-[3].

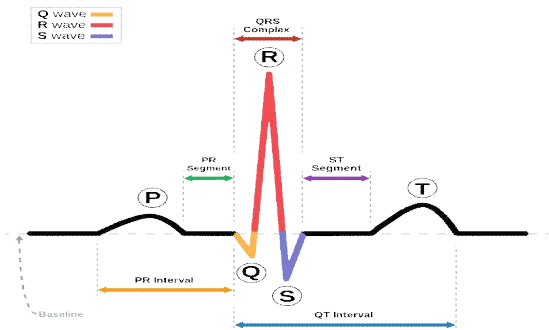


그림 1. ECG에 대한 P-QRS-T 심장 파형  
Figure 1. Heart wave of P-QRS-T for ECG

### 2. 디지털 IIR(Infinite impulse response) 필터 [4]

IIR 필터는 임펄스가 무한한 특징이 있는 디지털 필터의 한 종류이며, 재귀적으로(feedback) 적용되어 필터링 연산이 수행된다 [4]. IIR 필터의 차분방정식은 식 (1)과 같으며, 여기서 N을 필터의 차수(order)라고 한다.

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

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^M b_k z^{-k}}{1 + \sum_{l=1}^N a_l z^{-l}} \quad (2)$$

$$= \frac{b_0 + b_1 z^{-1} + \dots + b_M z^{-M}}{1 + a_1 z^{-1} + \dots + a_N z^{-N}}, \quad a_0 = 1$$

그리고 식 (1)의 차분 방정식을  $z$ -변환하여 시스템 함수  $H(z)$ 로 나타내면 식 (2)와 같다. 식 (2)에서  $b_k$ 와

$a_i$ 은 필터의 계수이며,  $a_N \neq 0$  일 때 차수(order) N 이다. 버터워스 필터, 체비세프 필터, 베셀 필터 등이 있으며, 본 논문에서는 디지털 IIR 버터워스 저역통과 필터를 설계하기로 하며, 자세한 설계 방법은 참고문헌을 참조하면 된다 [4].

### III. 딥러닝 알고리즘

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

그림 2는 LSTM 모델을 장단기 메모리 셀이라고도 하는데, RNN에 입력 게이트, 망각 게이트 그리고 출력 게이트를 도입하여 RNN 장기 의존성 문제를 해결했다. LSTM의 구조는 그림 2에 나타냈으며, RNN(Recurrent Neural Network)의 단점을 보완해서 나온 모델이며, 기존의 RNN이 출력과 먼 위치에 있는 정보를 기억할 수 없다는 단점이 있으므로, 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 중요 데이터를 결정하는 역할을 한다 [6].

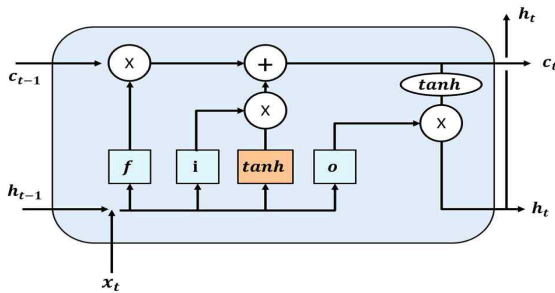


그림 2. LSTM의 구조  
 Figure 2. Structure of LSTM

LSTM 셀의 장기상태와 단기 상태, 그리고 하나의 샘플에 대해 단계별로 출력을 계산하는 순서를 보여주는 식으로 다음과 같이 표현할 수 있다 [5].

$$i_{(t)} = \sigma(W_{xi}^T x_{(t)} + W_{hi}^T h_{(t-1)} + b_i) \quad (3)$$

$$f_{(t)} = \sigma(W_{xf}^T x_{(t)} + W_{hf}^T h_{(t-1)} + b_f) \quad (4)$$

$$o_{(t)} = \sigma(W_{xo}^T x_{(t)} + W_{ho}^T h_{(t-1)} + b_o) \quad (5)$$

$$g_{(t)} = \tanh(W_{xg}^T x_{(t)} + W_{hg}^T h_{(t-1)} + b_g) \quad (6)$$

$$c_{(t)} = f_t \otimes c_{t-1} + i_{(t)} \otimes g_{(t)} \quad (7)$$

$$h_{(t)} = o_{(t)} \otimes \tanh(c_{(t)}) \quad (8)$$

식 (3)부터 (8)까지에서  $\otimes$ 는 원소별 곱셈을 의미하며,  $W_{xi}, W_{xf}, W_{xo}, W_{xg}$ 는 입력 벡터  $x_{(i)}$ 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.

#### 2. GRU(Gated Recurrent Unit) 모델 [5, 7]

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

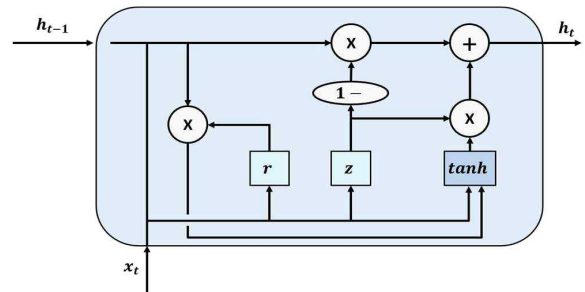


그림 3. GRU의 구조  
 Figure 3. Structure of GRU

$$z_{(t)} = \sigma(W_{xz}^T x_{(t)} + W_{hz}^T h_{(t-1)} + b_z) \quad (9)$$

$$r_{(t)} = \sigma(W_{xr}^T x_{(t)} + W_{hr}^T h_{(t-1)} + b_r) \quad (10)$$

$$g_{(t)} = \tanh(W_{xg}^T x_{(t)} + W_{hg}^T (r_{(t)} \otimes h_{(t-1)}) + b_g) \quad (11)$$

$$h_{(t)} = z_{(t)} \otimes h_{(t-1)} + (1 - z_{(t)}) \otimes g_{(t)} \quad (12)$$

GRU는 LSTM과 비슷하거나 더 좋은 성능을 낸다고 알려져 있으며, 구조적으로 LSTM보다 간단하고 연산 속도에서 성능이 더 뛰어나다는 점에서 최근에 집중 받고 있는 신경망이다.

### IV. 활성화 및 성능평가 함수

#### 1. 활성화 함수 (Activation function)

활성화 함수는 입력된 데이터의 가중 합을 출력 신호로 변환하는 함수이다. 인공 신경망에서 이전 레이어에

대한 가중 합의 크기에 따라 활성화 여부가 결정되며, 본 논문에서는 주로 쓰이는 세 가지의 활성화 함수(sigmoid, tanh, ReLU)를 사용하여 LSTM 모델에 적용하여 성능 평가 및 경과시간을 비교하였다 [5].

식 (13)의 sigmoid 함수는 0과 1사이의 값만 가질 수 있도록 하는 비선형 함수이며, 식 (14)의 ReLU 함수는  $x$ 가 양수이면 미분 값이 1이고,  $x$ 가 음수이면 미분 값이 0인 함수이다. 식 (15)의 tanh 함수는 수학적으로 sigmoid 함수를 평행 이동한 함수이며, -1과 1 사이의 값을 가질 수 있도록 하는 비선형 함수로 sigmoid 함수보다 더 우수한 성능을 가진다 .

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (13)$$

$$\text{ReLU}(x) = \max(0, x) \quad (14)$$

$$\text{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (15)$$

## 2. 성능 평가 함수 및 최적화 함수

딥러닝 LSTM과 GRU 모델을 이용한 성능평가는 평균 절대오차(Mean Absolute Error, MAE), 평균 제곱오차(Mean Squared Error, MSE)의 2가지에 대해서만 성능 평가를 진행하였고, 본 논문에서는 수치 값이 작으므로 백분율인 %로 표시하였다 [5, 10].

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \times 100(\%) \quad (16)$$

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \times 100(\%) \quad (17)$$

식 (16), (17)을 살펴보면,  $y_i$ 는 실제 값,  $\hat{y}_i$ 는 예측 값,  $n$ 은 전체 테스트 데이터의 개수이며, 아래 첨자  $i$ 는 데이터의 번호이다. 최적화 함수는 학습 데이터 셋을 이용하여 모델을 학습할 때 실제 결과와 예측한 결과의 차이를 줄여 최적화하는 역할을 한다. 본 논문에서는 연구 과정에서 가장 많이 사용하고 있는 RMSProp 방식과 Momentum 방식을 합쳐 각각의 단점을 보완하고 장점을 극대화시킨 adam() 함수를 사용하기로 한다.

## V. 필터 설계 및 딥러닝 성능평가

### 1. 디지털 IIR 버터워즈 저역통과 필터 설계 [4]

본 논문에서는 ECG의 잡음을 제거하기 위해 II-2의 디지털 IIR 버터워즈 저역통과 필터의 설계는 샘플링 주파수 600 Hz, 차단 주파수 30 Hz, 차수(order) 5를 가지며, 계산된 식 (2)의 계수들인  $a$ ,  $b$ 는 다음과 같다.

$$\begin{aligned} a_0 &= 1, a_1 = -3.98454312, a_2 = 6.43486709, \\ a_3 &= -5.25361517, a_4 = 2.16513291, a_5 = -0.35992825, \\ b_0 &= 5.97957804e-05, b_1 = 2.98978902e-04, \\ b_2 &= 5.97957804e-04, b_3 = 5.97957804e-04, \\ b_4 &= 2.98978902e-04, b_5 = 5.97957804e-05 \end{aligned}$$

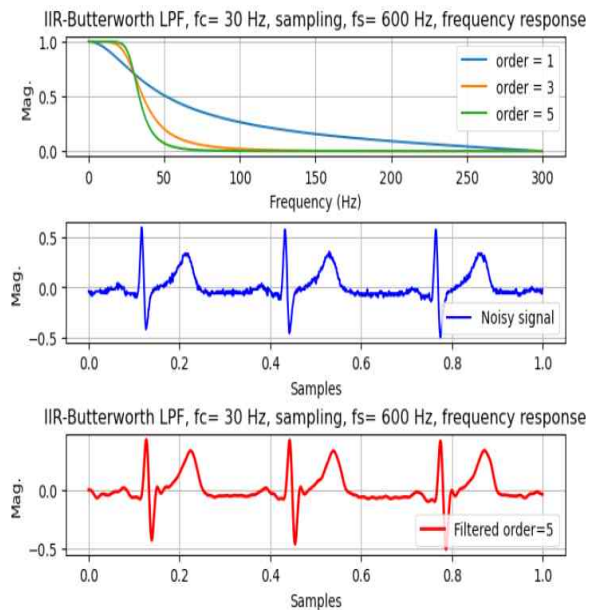


그림 4. 보드 선도 및 디지털 IIR Butterworth lowpass filter를 적용한 ECG 잡음 제거 신호

Figure 4. Bode plot and ECG signal of noise removed applying digital IIR Butterworth lowpass filter

디지털 IIR 버터워즈 저역통과 필터를 적용한 ECG 신호에 대한 해석을 종합적으로 그림 4에 나타내었다. 그림 4의 위 그림은 차수 1, 3, 5로 설계된 입출력 신호의 크기에 따른 감쇠 정도를 나타내는 저역통과 필터의 보드 선도(Bode plot)를 나타내었으며, 중간 그림은 ECG 장비를 이용하여 실험에서 획득한 잡음을 포함한 ECG 신호, 아래 그림은 디지털 IIR 버터워즈 저역 통과 필터를 적용하여 얻은 2.2절의 식 (2)의 계수  $a$ 와  $b$ 를 적용하여 잡음을 제거한 1,100개의 ECG 데이터이며,

데이터의 갯수는 ECG 신호가 주기적으로 매번 반복되는 신호이므로 저자가 임의로 설정하였음을 밝힌다.

## 2. LSTM 모델 설정 및 성능 평가 [5, 9]

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 나누어서 사용하며, 정규화 과정을 거쳐 실행시킨다.

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 import하여 신경망 모델을 구현했다. 최적화 함수는 adam(), epoch는 100회, batch size는 5로 설정하여 3가지 활성화 함수들에 대해 MAE, MSE의 2가지 함수들의 성능 평가를 비교하였다.

표 1. LSTM의 3가지 활성화 함수에 대한 성능 평가 비교  
 Table 1. Comparison of performance evaluation for three activation functions of LSTM

Activation function	MAE (%)	MSE (%)	elapsed time[sec]
sigmoid	0.8971221	0.0148704	50.8554
ReLU	0.5899207	0.0063005	51.2822
tanh	0.1945010	0.0008036	55.5531

표 1은 성능평가 MAE를 기준으로 %로 나타내면, sigmoid() 함수인 경우는 0.8971221 %, ReLU() 함수의 경우는 0.5899207 %, tanh() 함수의 경우는 0.3101949 %의 손실율을 보여주었으며, 성능평가 MSE를 기준으로 tanh() 함수인 경우, 0.0008036 %의 손실율을 보여 가장 우수한 활성화 함수인 것으로 판명되었다.

그리고 각 활성화 함수를 적용했을 때, ECG 신호에서 급격한 변화를 보이는 P-QRS-T 파형 중에서 R-peak 부분의 100개의 실제 및 예측 데이터의 결과를 확대하여 비교하였다. 활성화 함수 sigmoid(), ReLU(), tanh() 함수를 적용했을 때 각각 그림 5, 6, 7에 나타난 것으로, 범례에서 ECG 신호의 R-peak 부분의 실제 데이터는 실선, 예측 데이터는 '▲' 로 구분하여 표시하였으며, tanh() 함수를 적용한 그림 7의 경우가 오차가 가장 작았다.

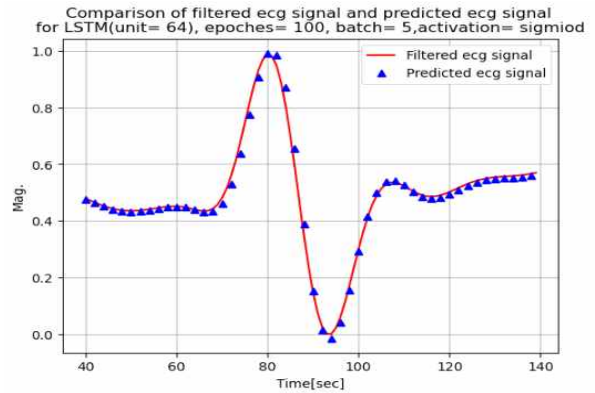


그림 5. 활성화 sigmoid() 함수를 이용한 예측 결과  
 Figure 5. Prediction result using activation sigmoid() function

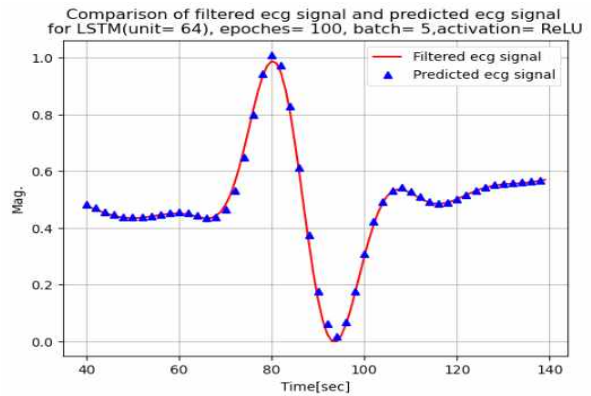


그림 6. 활성화 ReLU() 함수를 이용한 예측 결과  
 Figure 6. Prediction result using activation ReLU() function

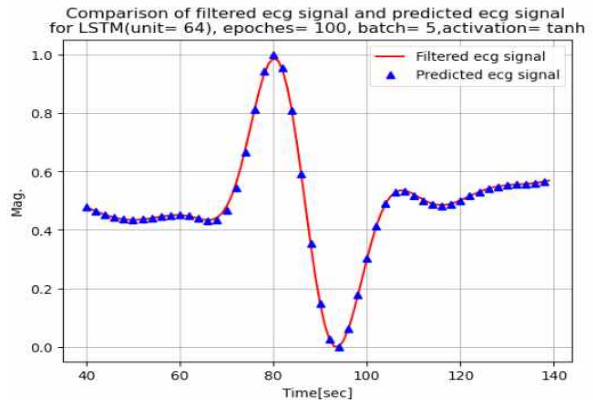


그림 7. 활성화 tanh() 함수를 이용한 예측 결과  
 Figure 7. Prediction result using activation tanh() function

## 3. LSTM과 GRU 모델의 성능 평가 비교

표 2는 표 1과 동일한 조건을 적용한 경우로, LSTM과 GRU의 은닉층의 노드 개수 unit= 64, epoch는 100회, batch size는 5, 최적화 함수는 adam()을 사용하였을 때, LSTM과 GRU 모델에서 LSTM 모델에서 가장

우수한 활성화 함수인  $\tanh()$ 를 사용하였다. 프로그램 경과시간을 보면 GRU 모델이 52.2984 초, LSTM 모델이 55.5531 초로 3.2547 초 정도 우수하였고, 성능 평가 MAE를 기준으로 GRU는 0.1641324 %, LSTM은 0.1945010 % 으로 GRU 모델이 0.0303686 % 만큼 어느 정도 우수한 것으로 판명되었다. 또한 성능평가 MSE 기준으로 GRU는 0.0006343 %, LSTM은 0.0008036 % 으로 GRU 모델이 0.0001693 % 만큼 MSE도 어느 정도 우수한 것으로 판명되었다.

표 2. LSTM과 GRU에 대한 성능 평가와 경과시간  
Table 2. Performance evaluation and elapsed time of LSTM and GRU

Performance evaluation	LSTM	GRU
MAE (%)	0.1945010	0.1641324
MSE (%)	0.0008036	0.0006343
Elapsed time [sec]	55.5531	52.2984

## VI. 결 론

본 논문에서는 획득한 ECG 신호가 잡음을 포함하고 있는 신호일 때, 샘플링 주파수 600 Hz, 차단 주파수 30Hz, 차수 5를 갖는 디지털 IIR 버터워즈 저역통과 필터를 설계하여 잡음을 제거한 ECG 데이터를 생성한 후, 딥러닝인 LSTM 및 GRU 모델의 소스 데이터로 사용하였다.

사용된 LSTM 모델은 은닉층의 노드 개수  $unit=64$ ,  $epoch=100$ ,  $batch\ size=5$ 를 사용했을 때 ECG 데이터를 이용하여 3가지 활성화 함수  $\text{sigmoid}()$ ,  $\text{ReLU}()$ ,  $\text{tanh}()$ 에 따른 MAE, MSE에 대한 성능 평가 결과를 비교했을 때, 3가지 활성화 함수 중에서  $\text{tanh}()$  함수가 오차 값(%)이 제일 작은 값으로 가장 우수하였다.

동일한 조건을 적용했을 때, GRU 모델과 LSTM 모델에 대한 성능 평가 및 경과 시간을 비교하였을 때 2가지 모두 GRU 모델이 LSTM 모델보다 우수한 것을 확인하였다. 따라서 GRU 모델은 성능 평가나 경과시간 측면에서 많은 데이터 양들을 처리할 때 LSTM 모델보다 유리할 것으로 앞으로 기대되는 딥러닝 모델임을 확인하였다.

향 후, 생체 데이터에 포함된 잡음을 다른 디지털 신호처리 방법을 이용한 LSTM과 GRU 모델에 적용하기 위해서는 좀 더 많은 연구가 필요할 것으로 본다.

## 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.ec21.com/ECG-KITAssembl-y-ECG-sensor-9570299\\_9570369.html](https://physiolab.en.ec21.com/ECG-KITAssembl-y-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>