



스마트 모터 진단 시스템의 구현

송호범*, 정준**

The Implementation of Smart Motor Diagnosis System

Ho-Bum Song*, Jun Jeong**

본 논문은 2018년도 동양미래대학교 학술연구과제 지원에 의하여 연구되었음

요 약

모터의 회전자 손상을 진단하기 위해, 모터를 정지 상태로 유지하고 회전자를 일정한 위치로 이동한 후에 인버터를 통해 측정된 저항 및 인덕터 값을 분석하는 방법이 많이 사용되고 있지만, 이것은 회전자를 일정한 위치에 이동해야 하는 반복적인 작업을 필요로 한다. 이에 본 연구에서는 반복적인 작업을 최소화할 수 있도록 인버터, 딥-러닝 엔진, 모바일 앱 프로그램을 포함하는 스마트 모터 진단 시스템을 구현하여, 인버터를 통해 측정된 소수의 데이터를 입력으로 사용하는 딥-러닝 알고리즘을 실행하여 빅 데이터를 생성하고, 이 데이터를 기반으로 학습을 진행한 후에 모터를 진단할 수 있도록 하였다. 실험을 통해 이 진단 시스템은 진단 장치의 진단 정확도를 검증할 수 있을 뿐만 아니라, 새로운 방식의 딥-러닝 스마트 진단 시스템으로 사용 가능성을 확인 할 수 있었다.

Abstract

In order to diagnose a rotor damage of the motor, the motor is stopped first and the rotor is moved to a predetermined position, thereafter a method of analyzing the resistance and the inductance measured through the inverter is widely used. However it's method requires repetitive tasks to move the rotor to a predetermined position. In this study, a smart motor diagnosis system using inverter, deep-learning engine and mobile app program were implemented to minimize repetitive tasks. Using the deep-learning algorithm, the big data were generated by applying a small number of data measured through the inverter and then the motor could be diagnosed. Through experiments, it was confirmed that this diagnosis system not only can verify the diagnostic accuracy of the motor but also can be used as a new type of deep-learning smart diagnosis system.

Keywords

deep learning, DCGAN, DNN, inverter, cloud DB, mobile app, motor diagnosis

* 동양미래대학교 로봇자동화공학부
- ORCID: <https://orcid.org/0000-0002-3451-5729>
** 동양미래대학교 로봇자동화공학부(교신저자)
- ORCID: <https://orcid.org/0000-0002-9755-3060>

• Received: Nov. 27, 2018, Revised: Dec. 12, 2018, Accepted: Dec. 15, 2018
• Corresponding Author: Jun Jeong
Dept. of Robot & Automation System, Dongyang Mirae University, 445
Gyeongin-ro, Guro-gu, Seoul, 152-714, Korea,
Tel.: +82-2-2610-1775, Email: hbsong@dongyang.ac.kr

1. 서 론

유도전동기는 경제성이 우수하고 구조적인 특성에 기인하는 장점이 있어 산업현장에서 구동장치로서 많이 사용되고 있다. 고정자 권선 단락 또는 개방으로 인한 고장, 회전자 바의 균열 등의 손상에 의한 고장[1][2] 등 여러 가지 종류의 고장이 유도전동기에서 발생할 수 있으며, 이로 인한 경제적인 손실이 발생한다. 이 손실을 최소화하기 위한 다양한 고장 진단 방법이 현장에서 적용되고 있다[3]. 유도전동기의 회전자 손상으로 고장이 발생한 경우에는 유도전동기를 정지 상태로 유지한 후에 전압 및 전류 등을 측정하고, FFT 알고리즘으로 이 데이터를 분석하여 회전자 손상을 진단하는 단상 회전 시험법이 많이 사용되고 있으며, 상용화 된 유도전동기 진단 장치에 이러한 기술이 적용되어 있다[4][5]. 하지만 이 방식을 사용하여 유도전동기를 진단하기 위해서는 정지 상태에서 일정한 위치로 회전자를 이동시킨 후에 전압 및 전류를 측정하는 반복 작업이 필요하다. 산업 설비의 고장 진단에 적용빈도가 증가하는 빅 데이터 및 인공지능 알고리즘을 이용하여 유도전동기 고장을 진단하기에는 이 방식을 적용하는데 많은 불편함이 있다. 이에 본 연구에서는 유도전동기를 단상 회전 시험법으로 진단하기 위한 인버터를 제작하고, 이 인버터에서 측정된 유도전동기의 저항 및 인덕터 값을 모바일 장치를 통해 취득한 후에 클라우드 데이터베이스로 저장하고, 딥-러닝 알고리즘으로 진단하는 스마트 모터 진단 시스템을 구현한다. 유도전동기를 일정한 각도로 이동하는 반복 작업을 최소화하기 위해서, 딥-러닝 서버는 데이터베이스에 저장된 측정 데이터를 DCGAN(Deep Convolutional Generative Adversarial Networks)[9]-[11] 알고리즘으로 구현한 딥-러닝 엔진의 입력으로 사용하여 인버터에서 측정된 데이터와 유사한 특성을 지닌 빅 데이터를 생성토록 한다. FFT 알고리즘을 통해 이 데이터에 유도전동기의 정상 또는 고장 상태를 나타내는 정보를 포함시키고, DNN(Deep Neural Network)[6]-[8] 알고리즘으로 구현한 딥-러닝 엔진을 학습시킨다. 학습이 완료되면 모바일 장치에서 구현된 앱 프로그램은 딥-러닝 엔진

에게 인버터에 접속된 진단 대상 모터(유도전동기)로부터 측정된 데이터를 기반으로 고장 유무를 확인할 수 있도록 한다.

II. 스마트 모터 진단 시스템의 설계 및 구현

그림 1은 본 연구에서 제안한 인버터, 클라우드 데이터베이스, 딥-러닝 서버, 모바일 장치가 무선으로 구동토록 설계한 스마트(딥-러닝) 모터 진단 시스템 구성도를 나타낸다.

모바일 장치에서 실행되는 앱 프로그램은 인버터의 제어부에 내장한 블루투스 모듈에 접속을 한다. 인버터는 전원부에 접속된 모터의 저항 및 인덕터 값을 산출한다. 앱 프로그램은 WIFI 통신을 통해 이 데이터를 클라우드 데이터베이스에 저장하고, 딥-러닝 서버는 인터넷을 통해 데이터베이스에 저장된 측정 데이터를 읽는다. 이 후에 DCGAN 엔진을 실행하여 빅 데이터를 생성하고, DNN 엔진을 학습시키고, DNN 엔진은 모터를 진단하고, 이 결과를 앱 프로그램을 통해 확인한다.

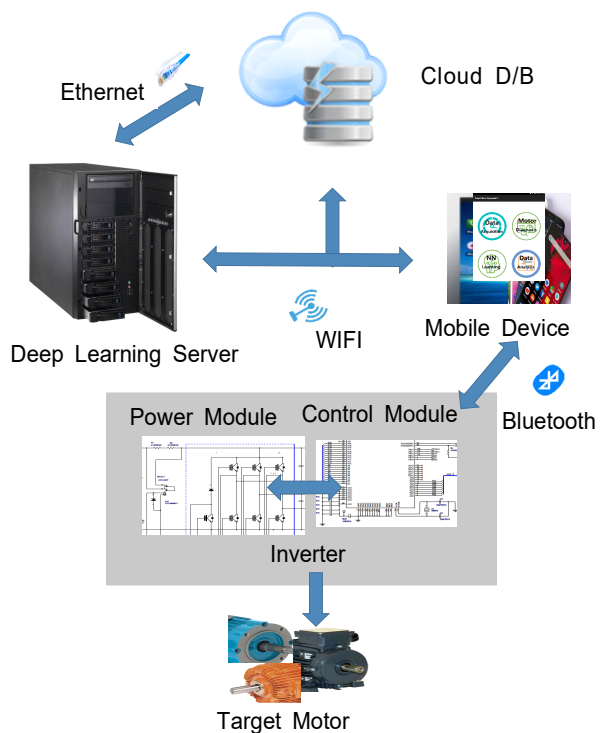


그림 1. 스마트 모터 진단 시스템 구성도
Fig. 1. Block diagram of smart motor diagnosis system

2.1 하드웨어

전단 시스템의 하드웨어인 인버터는 교류 입력전압 220V, 교류 최대 출력전류 20A 및 교류 차단전류 30A에서 구동되며 전원부와 제어부로 구성한다. 전원부의 DC-AC 컨버터는 스위칭 소자로 600V, 30A에서 동작할 수 있는 게이트 드라이버를 내장한 “FSBB30CH60” IPM 모듈을 사용한다. “TPS223Y” FET를 사용한 플라이백 방식의 DC-DC 컨버터는 “SKBPC5016” 브릿지 다이오드로 구성된 정류회로에서 출력된 180V ~ 420V의 직류입력전압을 사용하여 4가지 종류의 직류전원을 발생시킨다. +5V, 0.2V 직류전원은 IPM 로직 및 제어부 로직에서 사용하고, +15V, 0.2A 직류전원은 IPM 제어전원 및 OP-AMP (+)제어전원으로 사용하고, -15V, 0.1A 직류전원은 OP-AMP (-)제어전원으로 사용하고, +24V, 0.1A 직류전원은 릴레이 구동전원으로 사용한다. 자연 냉각 방식을 채택하고 과전류 및 과전압 보호회로를 내장하고 있다. 그림 2는 인버터의 전원부 회

로를 나타낸다.

인버터의 제어부는 32Bit, 20MHz 클럭에서 구동되는 RISK 기반의 “SH7034”를 프로세서로 사용하고, 저전력에서 동작하는 “BS62LV1605” SRAM 2개를 사용하여 4MByte로 구성하여 프로그램 및 데이터를 저장할 수 있도록 한다. 프로세서는 200 μ S 주기로 인터럽트를 발생시켜 인버터의 전압 및 전류를 측정하고 이 값들을 A/D 컨버터를 사용하여 10Bit 데이터로 변환하여 SRAM에 저장한다. 직렬 포트 1은 인버터의 현재 상태, 전압 또는 전류 등의 측정값을 표시하는 LCD 모듈용으로 사용하고, 직렬 포트 2는 블루투스 통신용으로 사용한다. 이 직렬 포트에 SPP 프로파일을 내장된 “FB155BC”를 접속한다. 40핀 커넥터는 전원부와 제어부의 인터페이스를 위해 사용한다. 이 커넥터를 통해 전원부는 제어부 구동에 필요한 TTL 또는 OP-AMP 전원, 센서에서 측정된 전압 및 전류 신호를 전달하고 제어부는 전원부의 IPM 모듈에 필요한 정현파 PWM 신호 등을 전달한다.

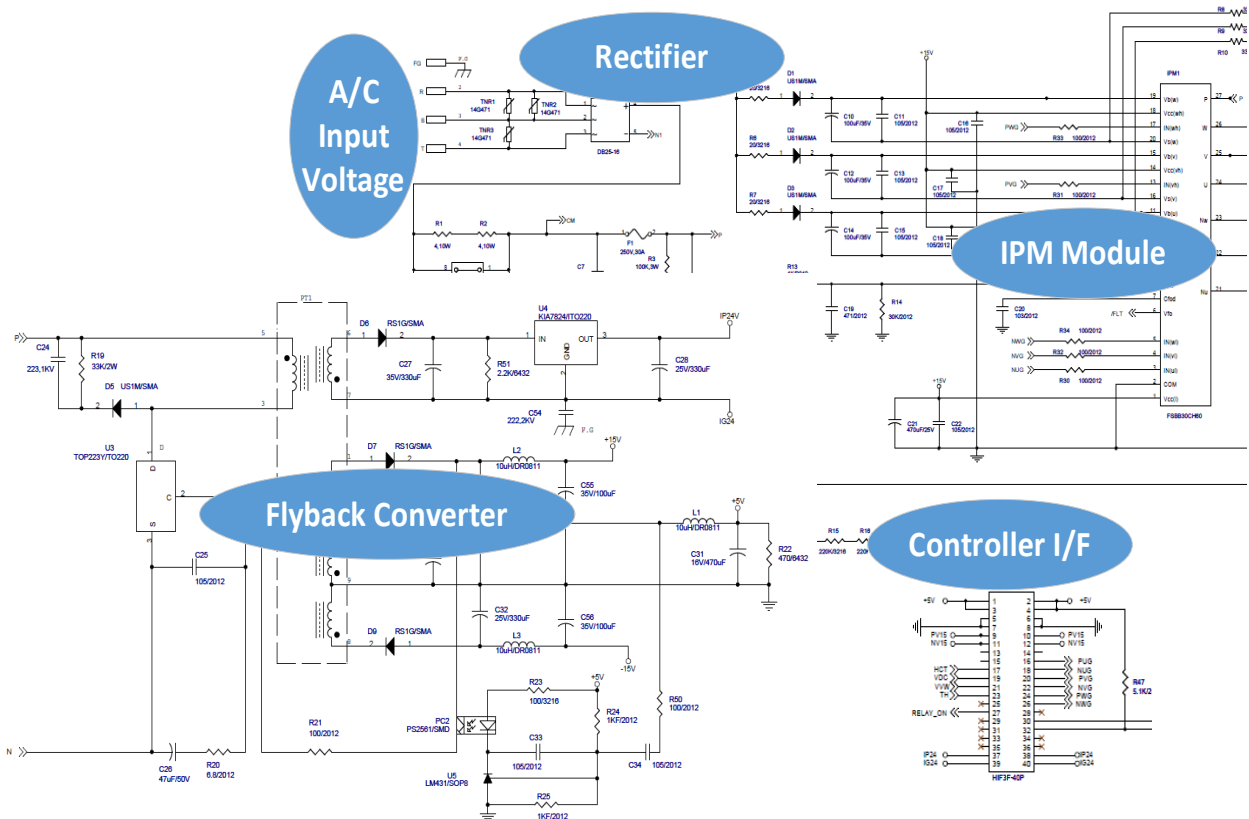


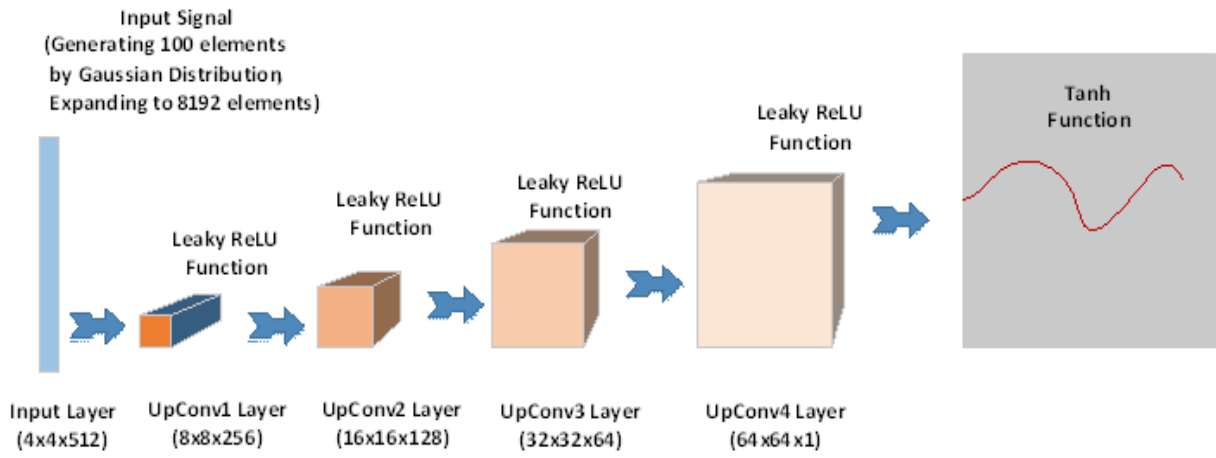
그림 2. 인버터 전원부 회로
Fig. 2. Power circuit of inverter

2.2 소프트웨어

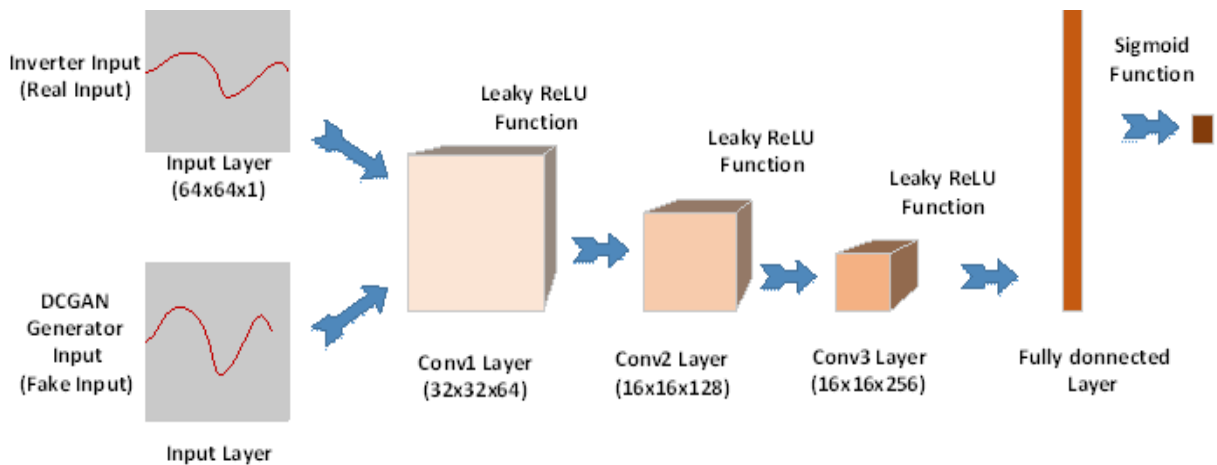
진단 시스템의 소프트웨어는 인버터를 구동시킨 후에, 이 인버터의 전압 및 전류를 측정하여 모터의 저항 및 인덕턴스 값을 산출하고, 이 데이터를 모드 버스 프로토콜(Modbus protocols)로 모바일 장치로 전송하는 인버터의 제어부에서 구현한 시스템 소프트웨어, 딥-러닝 서버에서 파이썬(Python)으로 구현한 학습용 빅 데이터를 생성하는 DCGAN 알고리즘 기반의 딥-러닝 엔진 및 모터의 진단 정확도를 판별하는 DNN 알고리즘 기반의 딥-러닝 엔진, 클라우드 데이터베이스인 “Firebase”와 연동하여 인버터에서 생성된 데이터를 저장하고, 모터 진단을 실행시키고, 진단 결과를 확인 하는 모바일 장치에서 구현

한 액티비티(Activity) 등으로 구성된다.

DCGAN 엔진에 적용하는 생성기(Generator)의 계층 구조 및 선택기(Discriminator)의 계층 구조는 그림 3과 같다. DCGAN 생성기는 입력으로 가우션 분포로 생성한 100개의 랜덤 데이터를 사용한다. 8192개의 입력 뉴런으로 확장한 후에 4번의 Up-Convolution 과정을 실행하여 64×64 크기의 CNN에서 사용하는 입력 뉴런 데이터를 생성한다. 4개의 Up-Convolution 계층의 출력 함수는 Leaky ReLU 함수를 사용하고 마지막 계층의 출력 함수는 Tanh 함수를 사용한다. 반면에 DCGAN 선택기는 입력으로 실제 입력에 해당하는 인버터에서 측정된 입력 데이터(Real Input Data)와 생성기에서 생성된 거짓 입력(Fake Input Data)을 교대로 사용한다.



(a) DCGAN generator



(b) DCGAN discriminator

그림 3. DCGAN의 계층 구조

Fig. 3. Layer architecture of DCGAN

3번의 Convolution 과정을 실행하여 모터의 정상 또는 고장 상태를 판별하는 “0 ~ 1” 사이의 값을 출력토록 학습한다. 3개의 Convolution 계층의 출력 함수는 Leaky ReLU 함수를 사용하고, 마지막 계층의 출력함수는 Sigmoid 함수를 사용한다.

인버터에서 측정된 36개의 저항 또는 인덕턴스 값을 기초 데이터로 사용하여 대량의 모터 진단 데이터 셋을 생성하고, DCGAN 엔진을 실행하여 빅 데이터를 생성하고, DNN 엔진을 학습시키고, 이 엔진을 사용하여 모터를 진단하는 과정은 다음과 같은 절차로 진행된다.

① 진단 모터의 회전자를 일정한 위치에 고정하고 인버터를 실행시킨 후에, 측정된 전압 v , 전류 i , 및 인가한 주파수 f 를 식 (1)에 대입하여 저항 및 인덕턴스 값을 계산한다.

$$\text{전력} : P = \frac{1}{T} \int_0^T v(t) \times i(t) dt = \frac{1}{T} \sum_k v(k) \times i(k)$$

$$\text{전류} : I = \sqrt{\frac{1}{T} \int_0^T i(t)^2 dt} = \sqrt{\frac{1}{T} \sum_k i(k)^2}$$

$$\text{전압} : V = \sqrt{\frac{1}{T} \int_0^T v(t)^2 dt} = \sqrt{\frac{1}{T} \sum_k v(k)^2}$$

$$\text{전압} : V = \sqrt{\frac{1}{T} \int_0^T v(t)^2 dt} = \sqrt{\frac{1}{T} \sum_k v(k)^2}$$

$$\text{역률} : p.f = \frac{P}{I \times V},$$

$$\text{저항} : R = \frac{V}{I} \times \cos(p.f)$$

$$\text{인덕턴스} : L = \frac{V}{I} \times \frac{\sin(p.f)}{2\pi f} \quad (1)$$

② 회전자를 10도 간격으로 이동시킨 후에, 동일한 과정을 반복하여 총 36개의 저항 및 인덕턴스 값을 데이터베이스에 저장한다.

③ 데이터베이스에 저장된 데이터 36개를 사용하여 2개의 값 사이에 평균값을 추가하여 72개의 데이터로 확장하고, 이 데이터를 9개 구간으로 구분한 후에 랜덤 샘플링으로 8개의 값을 제거하여 64개로 축소된 대량의 데이터를 생성한다.

④ FFT 알고리즘을 실행하여 직류성분 대비 고조파 성분을 계산하여, 생성한 모든 데이터를 정상 상태 데이터와 고장 상태 데이터로 분류한다.

⑤ 정상 상태 및 고장 상태 데이터는 DCGAN 선택기의 인버터 입력 데이터로 사용하기 위해 One-Hot Encoding 과정을 실행하여 64×64 크기의 데이터로 변환한다.

⑥ DCGAN 엔진을 실행시켜 생성기에서 생성된 64×64 크기의 입력 데이터 셋을 FFT 알고리즘을 실행하여 정상 상태 데이터와 고장 상태 데이터로 분류 한다.

⑦ 이 데이터는 DNN 엔진의 학습용 또는 진단용 입력 데이터로 사용한다.

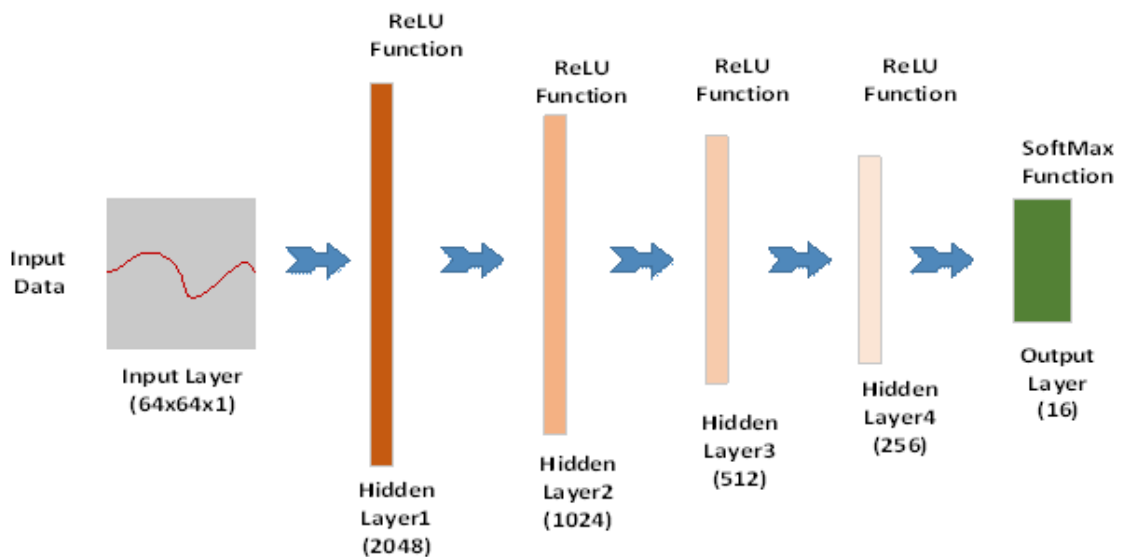


그림 4. DNN의 계층 구조
Fig. 4. Layer architecture of DNN

그림 4는 모터를 진단하는 DNN 엔진의 계층 구조를 나타낸다. DCGAN 선택기에서 생성된 64×64 크기의 입력 데이터를 입력층에 인가한 후에 순차적으로 2048, 1024, 512, 256개 뉴런으로 구성된 4개의 은닉층과 16개 뉴런으로 구성된 출력층을 통해 학습을 진행한다. 은닉층은 출력함수로 ReLU 함수를 사용하고 출력층은 출력함수로 식 (2)와 같은 SoftMax 함수를 사용한다. 16개의 출력뉴런 가운데 8개는 모터의 정상 상태를 나타내는 값으로 설정하고, 나머지 8개는 모터의 고장 상태를 나타내는 값으로 설정한다.

$$S_j = \frac{e^{a_j}}{\sum_{k=1}^N e^{a_k}}, \quad \forall j \in 1 \dots N \quad (2)$$

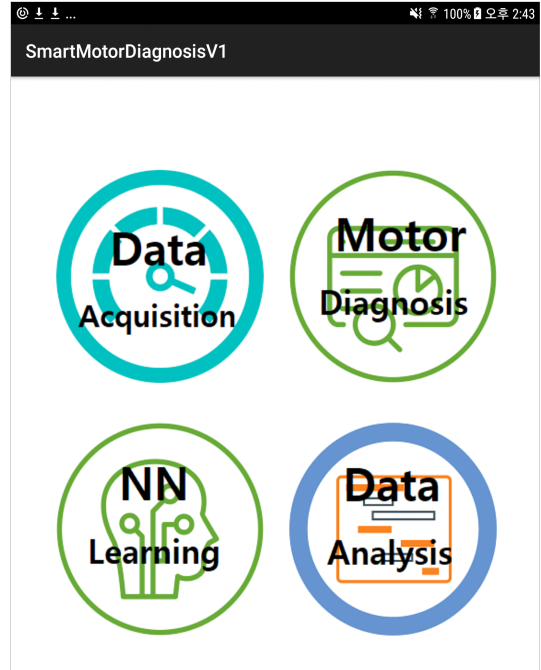
III. 실험 결과

본 연구에서 제작한 스마트 모터 진단 시스템은 그림 5에 나타내었다. 발열에 의한 오동작을 예방하기 위해 전원부의 뒷면에는 방열판과 IPM 모듈을 부착하고, 앞면에는 제어부 및 LCD를 배치하고, 40핀 커넥터를 통해 전원부와 제어부가 데이터를 전송하고, 태블릿은 외부의 지지대에서 별도로 사용할 수 있도록 하였다.

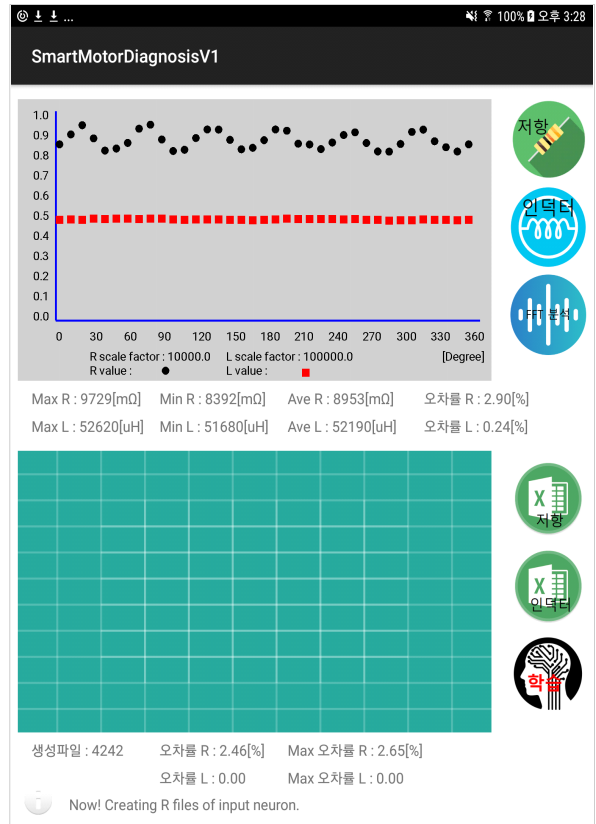


그림 5. 구현한 스마트 모터 진단 시스템
Fig. 5. Implemented smart motor diagnosis system

그림 6은 “Dash Board 액티비티”와 “NN Learning 액티비티”가 태블릿에서 실행한 결과를 나타내었다.



(a) Dash board activity



(b) NN learning activity

그림 6. 앱 프로그램 실행 결과
Fig. 6. Executing result of app. program

표 1. 실험 1 요약

Table 1. Summary of Exp. 1

Data Mode	Inverter Output		DCGAN Input	DCGAN Output	DNN Input		Accuracy		
	Motor Status	R error (%)	File Number	File Number	File Number(Normal)	File Number(Fault)	Number	Result (%)	
Training	Normal (4 Sets)	0.31	70,000	6,254	5,361	893	1	18.35	
		0.22		8,118	339	7,779			
		0.95		6,703	1,887	4,816	2	18.34	
		0.25		6,373	5,680	693			
	Fault (3 Sets)	1.83		7,889	314	7,575	3	18.33	
		1.37		7,240	1,978	5,262			
		4.14		7,179	66	7,113			
	Testing	Fault (1 Sets)		10,000	618	618	0	Ave.	18.34
					6,556	0	6,556		

“Dash Board 액티비티” 실행화면에서 “Data Acquisition” 메뉴를 터치하면 인버터를 통해 측정된 저항 및 인덕터 값을 데이터베이스에 저장하는 “Data Acquisition 액티비티”가 실행되고, “NN Learning” 메뉴를 터치하면 FFT 알고리즘 및 DCGAN 알고리즘을 통해 DNN 엔진의 학습용 데이터를 생성한 후에 이 엔진을 학습시키는 “NN Learning 액티비티”가 실행되고, “Motor Diagnosis” 메뉴를 터치하면 학습된 DNN 엔진을 이용하여 모터를 진단하는 “Motor Diagnosis 액티비티”가 실행된다. “NN Learning 액티비티” 실행화면에서 왼쪽 상단 저항 및 인덕터 버튼을 클릭하면 데이터베이스에서 읽어 들인 저항 및 인덕터의 최소값, 최대값 및 FFT 알고리즘으로 산출한 오차율을 나타내고, 그래프에는 저항 및 인덕터 값의 변화를 나타낸다. 반면에 왼쪽 하단 저항 및 인덕터 버튼을 클릭하면 DCGAN 엔진이 실행되어 DNN 엔진의 학습용 데이터를 생성하고, 학습 버튼을 클릭하면 이 데이터를 이용하여 DNN 엔진을 학습한다.

인버터에 진단 대상 모터를 접속한 후에 테블릿의 앱 프로그램을 사용하여 다음과 같은 절차로 실험을 진행하였다.

① 인버터를 실행시켜 4세트의 고장 상태 데이터와 4세트의 정상 상태 데이터를 취득한 후에, 1세트의 고장 상태 데이터를 제외한 7세트의 데이터에서 DCGAN 엔진에서 사용하는 70,000개의 입력 데이터를 생성한다.

② DCGAN 엔진을 실행하여 49,756개의 DNN 입력 데이터를 생성한 후에, FFT 알고리즘을 실행하여 15,625개의 정상 상태 데이터와 34,131개의 고장 상태 데이터로 분류한다.

③ 학습용 데이터로 사용하는 7세트에서 생성한 49,756개 데이터와 테스트 데이터로 사용하는 8번째 세트 데이터에서 생성된 618개의 정상 상태 데이터만을 사용하여 DNN 엔진을 학습한다.

④ 테스트용 데이터 6,556개를 DNN에 엔진에 인가하여 3번 연속 실험을 실시한 후에 측정된 진단 정확도를 표 1에 나타내었다.(실험 1)

⑤ 학습용 데이터로 사용하는 7세트에서 생성한 49,756개 데이터와 테스트 데이터로 사용하는 8번째 세트 데이터에서 생성된 618개의 정상 상태 데이터 및 6000개의 고장 상태 데이터를 사용하여 새로운 DNN 엔진을 학습한다.

⑥ 사용하지 않은 고장 상태 테스트용 데이터 556개를 DNN 엔진에 인가하여 3번 연속 실험을 실시한 후에 측정된 진단 정확도를 표 2에 나타내었다.(실험 2)

그림 7은 DNN 엔진의 학습률(learning rate)을 “0.005”로 설정한 후에 실험 1을 실시하여 산출한 Epoch 횟수에 대한 Cost 값 및 진단 정확도 값의 변화를 나타낸 그래프이다. Cost 값은 Epoch 횟수가 증가함에 따라 “0.20384 ~ 0.00032”로 감소하고 있지만, 진단 정확도 값은 “10.90% ~ 18.53%”로 크게 변화하지 않음을 알 수 있다.

표 2. 실험 2 요약

Table 2. Summary of Exp. 2

Data Mode	Inverter Output		DCGAN Input	DCGAN Output	DNN Input		Accuracy			
	Motor Status	R error (%)	File Number	File Number	File Number(Normal)	File Number(Fault)	Number	Result (%)		
Training	Normal (4 Sets)	0.31	70,000	6,254	5,361	893	1	90.00		
		0.22		8,118	339	7,779				
		0.95		6,703	1,887	4,816				
		0.25		6,373	5,680	693				
	Fault (3 Sets)	1.83		7,889	314	7,575	3	84.31		
		1.37		7,240	1,978	5,262				
		4.14		7,179	66	7,113				
	Testing	Fault (1 Sets)			10,000	6,618	618	0	Ave.	88.33
						556	0	556		

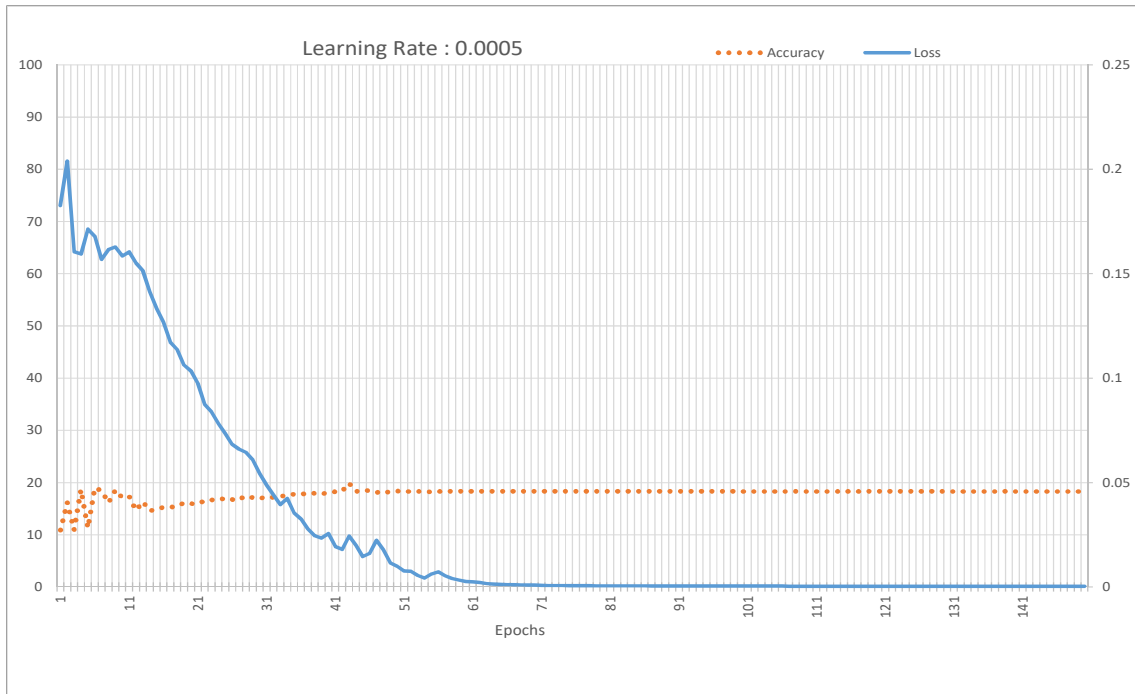


그림 7. DNN 엔진 실행 결과(실험 1)
 Fig. 7. Executing result of DNN engine(Exp. 1)

반면에 그림 8은 DNN 엔진의 학습률을 동일하게 설정한 후에, 실험 2를 실시하여 산출한 Epoch 횟수에 대한 Cost 값 및 진단 정확도 값의 변화를 나타낸 그래프이다. Cost 값은 Epoch 횟수가 증가함에 따라 “0.19083 ~ 0.02528”로 감소하고 있고, 진단 정확도 값은 “11.46% ~ 88.33%”로 증가하고 있다.

따라서 이 실험을 통해, DCGAN 알고리즘을 통해 빅 데이터를 생성하고 이 데이터로 적절하게 학습한 DNN 엔진을 사용하여 구현한 딥-러닝 기반의 모터 진단 장치는 기존의 FFT 알고리즘을 채택한 모터 진단 장치의 진단 정확도와 거의 동일한 결과를 얻을 수 있음을 확인 할 수 있었다.

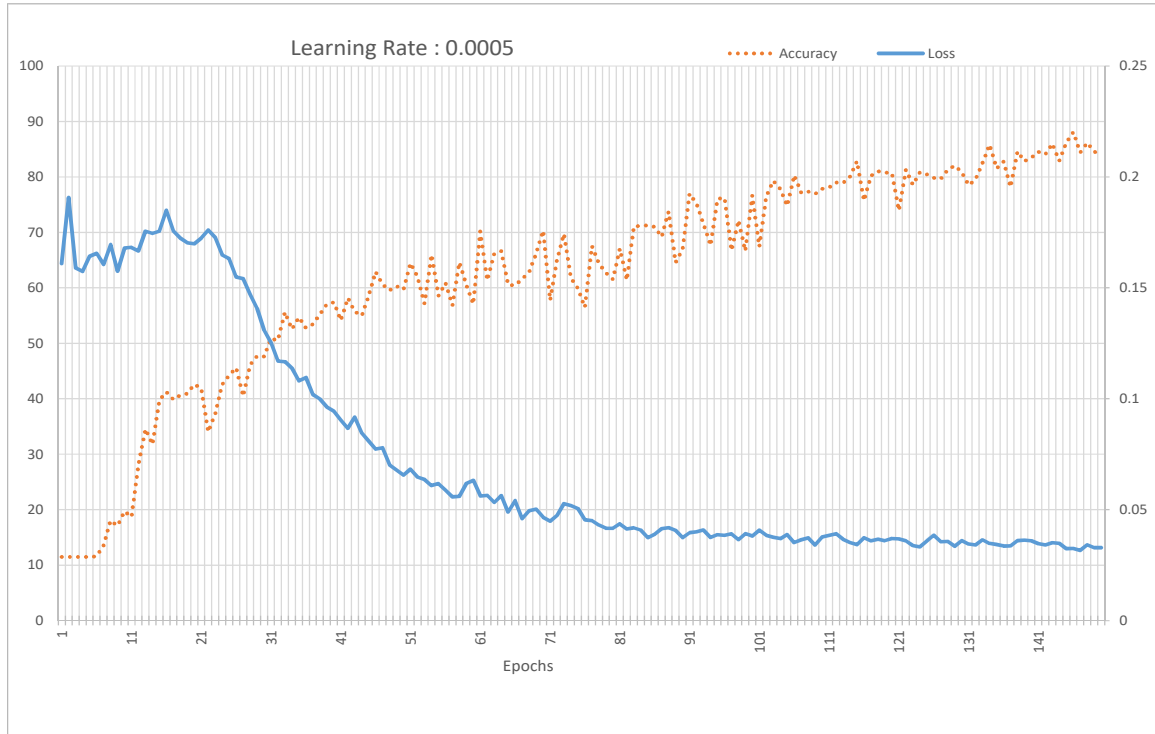


그림 8. DNN 엔진 실행 결과(실험 2)
Fig. 8. Executing result of DNN engine(Exp. 2)

IV. 결 론

본 연구에서는 모터를 진단하기 위해 전원부 및 제어부로 구성되는 인버터를 제작하였고, 이 제어부에 내장한 블루투스 모듈을 통해 모드 버스 프로토콜로 데이터를 전송할 수 있는 모바일 장치에서 실행되는 앱 프로그램, 클라우드 데이터베이스와 연동하여 딥-러닝 서버에서 실행되는 텐서플로우로 기반의 딥-러닝 엔진 등을 포함하는 스마트 모터 진단 시스템을 구현하였다. 실험 결과를 통해 이 진단 시스템은 다음과 같은 특징을 지니고 있음을 알 수 있었다.

첫째로 클라우드 및 로컬 환경에서 테블릿과 같은 모바일 장치로 기동되는 스마트 모터 진단 시스템을 개발하였다. 이 시스템은 상용화된 FFT 기반의 진단 알고리즘을 채택한 모터 진단 장치에 딥-러닝 알고리즘을 추가하여 기능을 확장시킬 수 있는 구조를 지니고 있어, 진단 장치의 진단 성능을 향상시킬 수도 있고, 별도로 휴대가 용이한 딥-러닝 기반의 모터 진단 장치로 사용할 수도 있다.

둘째로 인버터에서 취득한 데이터 셋을 클라우드 데이터베이스에 저장하고 있는 구조로 설계되어 있어 지속적인 사용으로 데이터가 증가되면, 이 데이터로 딥-러닝 엔진의 추가 학습을 진행할 수 있어, 현재 90% 정도의 진단 정확도를 더 향상시킬 수 있는 가능성이 있다.

셋째로 딥-러닝 서버를 사용하지 않고 성능이 우수한 딥-러닝 API를 내장한 모바일 장치에서 실행할 수 있는 구조로 용이하게 변경할 수 있으므로 소형의 모바일 모터 진단 장치로 확장할 수 있는 가능성을 제시한다.

향후에 현장에서 운전 중인 모터의 외부에 별도의 물리적인 결선을 하지 않고 전압, 전류, 진동 등을 측정할 수 있는 제어장치를 부착한 후에, 모바일 장치로 이 데이터를 취득하고, 지속적으로 딥-러닝 엔진의 학습 데이터로 활용할 수 있는 구조를 지닌 모터 진단 장치를 개발하려고 한다. 이런 구조로 개발된 모터 진단 장치는 측정이 간편한 장점을 지니고 있어, 자율주행 전기 자동차의 모터 진단 분야에 사용할 수 있을 것으로 예상된다.

References

- [1] F. Immovilli, C. Bianchini, M. Cocconcelli, A. Bellini, and R. Rubini, "Bearing fault model for induction motor with externally induced vibration", IEEE Trans. Ind. Electron., Vol. 60, No. 8, pp. 3408-3418, Aug. 2013.
- [2] B. Xu, L. Sun, L. Xu, and G. Xu, "Improvement of the Hilbert method via ESPRIT for detecting rotor fault in induction motors at low slip", IEEE Trans. Energy Convers, Vol. 28, No. 1, pp. 225-233, Mar. 2013.
- [3] P. M. De La Barrera, G. R. Bossio, and J. A. Solsona, "High-resistance connection detection in induction motor drives using signal injection", IEEE Trans. Ind. Electron, Vol. 61, No. 7, pp. 3563-3573, Jul. 2014.
- [4] Georgakopoulos, E. Mitronikas, and A. Safacas, "Detection of induction motor faults in inverter drives using inverter input current analysis", IEEE Trans. Ind. Electron, Vol. 58, No. 9, pp. 4365-4373, Sep. 2011.
- [5] B. Kim, K. Lee, J. Yang, S. B. Lee, E. J. Wiedenbrug, and M. R. Shah, "Automated Detection of Rotor Faults for Inverter-Fed Induction Machines Under Standstill Conditions", IEEE Trans. on Ind. App., Vol. 47, No. 1, pp. 2024-2033, Jan. 2013.
- [6] Dong-Ha Shin and Chang-Bok Kim, "A Study on Deep Learning Input Pattern for Summer Power", Journal of KIIT, Vol. 14, No. 11, pp. 127-134, Nov. 2016.
- [7] Ho-Bum Song and Jun Jeong, "The Implementation of smart DNC depaneling router", Journal of KIIT, Vol. 15, No. 3, pp. 11-22, Mar. 2017.
- [8] Ho-Bum Song and Jun Jeong, "The Implementation of smart Plug-in Switching Mode Power Supply", Journal of KIIT, Vol. 16, No. 2, pp. 17-27, Feb. 2018.
- [9] Danchen Zhao, Jingkun Weng, and Yuehu Liu, "Generating traffic scene with deep convolutional generative adversarial networks", Chinese Automation Congress(CAC), pp. 6612-6617, Oct. 2017.
- [10] Kai Zhang and Wei Zhang, "Loop closure detection based on generative adversarial networks for simultaneous localization and mapping systems", Chinese Automation Congress(CAC), pp. 7916-7919, Oct. 2017.
- [11] Yeh, R. A., Chen, C., Lim, T. Y., Schwing, A. G., HasegawaJohnson, M., and Do, M. N., "Semantic image inpainting with deep generative models", In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6882-6890, Nov. 2017.

Authors

송 호 범 (Ho-Bum Song)



1984년 2월 : 서울대학교
전기공학과(공학사)
1986년 2월 : 서울대학교
전기공학과(공학석사)
1986년 ~ 1994년 : 삼성전자
시스템개발실 선임 연구원
2005년 : 성균관대학교 정보통신

공학부 박사 수료

1995년 3월 ~ 현재 : 동양미래대학교 로봇자동화공학부 교수
관심분야 : 딥-러닝, 모바일 프로그램, 임베디드 시스템

정 준 (Jun Jeong)



1994년 2월 : 연세대학교
기계공학과(공학사)
1996년 8월 : 연세대학교
기계공학과(공학석사)
2001년 8월 : 연세대학교
기계공학과(공학박사)
2002년 ~ 2005년 : 삼성전자

스토리지사업부 책임연구원

2006년 3월 ~ 현재 : 동양미래대학교 로봇자동화공학부
부교수

관심분야 : 임베디드 시스템, 자동제어