



# 연구논문/작품 최종보고서

2018 학년도 제1 학기

제목 : Super Sensor

차호근(2012310053)

이수인(2013313073)

2018 년 5 월 8 일

지도교수: 신 동 군

서명 

계획(10)	주제(20)	개념(20)	상세(30)	보고서(20)	총점(100)

\* 지도교수가 평가결과 기재

## ■ 요약

스마트 응용 기술이 확장됨에 따라 실생활에 인터넷 기술을 접목하는 IoT(Internet of Things)가 등장했다. 이는 다양한 방식으로 사용자의 요구사항에 맞는 센서를 설치해 원하는 정보를 얻고 확인을 가능케 하였다.

본 과제는 General-purpose sensing 방식으로 스마트 홈 구현에 높은 효율성과 정확성 달성하기 위함이다. 기존 스마트 홈서비스에 많이 사용되는 카메라 센서는 사생활 문제로 과감하게 제외시켰다. 이로 인해 발생하는 정보 수집량의 감소를 보완하기 위해 보드에 부착된 다양한 센서를 통해 상호 데이터 값의 정형화, 패턴 분석 및 학습으로 감지되는 행동을 보다 정확하게 포착하고 인지하게 하였다. 또한, 머신러닝 기법을 적용하여 일차원적으로 물체의 작동여부 뿐 아니라 물체의 상태를 체크하고 상황에 따라 사용자에게 알릴 수 있는 더욱 유의미하고 고차원적인 활용이 가능하다.

본 작품은 아두이노와 온·습도센서, 사운드센서, 조도센서, 가속도센서, 컬러센서, PIR센서와 Grideye에 기반해 진행되었다. 수집한 데이터의 구조는 각 센서 값과, 이전 값과의 차, 열상감지센서의 경우는 각 행과 열의 평균으로 구성된다. 데이터 수집을 진행하며 이벤트, 즉 라벨별 유의미하게 작용하는 센서를 분석할 수 있었다. 본 작품의 데이터셋은 31개의 X feature와 1개의 Y feature로 구성된다. Y feature의 경우 9가지의 single-event와 7가지의 double-event로 구성된다. 지도학습으로 수집된 데이터는 분류기법의 정확도를 높이기 위한 데이터 정형화를 거쳐 30개와 25개의 unit을 가진 두 개의 은닉층을 통과하여 학습되며 96.22%의 높은 정확도를 보인다.

마지막으로 본 작품을 진행하면서 느낀 점과 배운 것, 아쉬운 점을 제시하며 본 논문을 마무리한다.

## ■ 서론

### 가) 제안배경 및 필요성

인터넷 기술과 정보산업(IT) 기술이 발달함에 따라 다양한 사물들이 인간과 상호 연결을 이루었다. 사물과의 적용 범위와 환경은 갈수록 확장되며 인간의 요구를 충족시키고 있다. 이는 IoT(Internet of Things)의 등장으로 이어졌으며, 현재 4차 산업혁명의 중심에 있다. IoT는 인터넷을 기반으로 모든 사물을 연결하여 정보를 상호 소통하는 지능형 기술 및 서비스로 사람들의 생활 속에 편의와 위험요소에 대한 예측 등에 긍정적인 변화를 불러일으키고 있다. 가장 쉽게 접근할 수 있는 적용 환경으로 주거공간인 집에서 가장 방대한 연구가 진행되고 있다. 그 외에도 회사, 건물, 공장, 도시, 나라까지 영향을 미치고 있다. 환경에 따라 사용자들의 요구사항은 다양하다. 집에서만 해도 자동화, 엔터테인먼트, 교육, 보안, 에너지 관리, 스마트 가전, 스마트카, 헬스케어 등 일상생활과 관련된 방대한 분야에 걸쳐 요구사항

이 존재한다[1]. 이에 따라 세계 홈 스마트 시장은 지속적인 상승세를 보일 것으로 예상되며 현재 통신사업자, 가전사업자, 플랫폼사업자 중심으로 시장이 개화되고 있다. 대표적인 국내 스마트홈 시장은 현재 IoT@home 이라는 브랜드의 홈 IoT 서비스를 출시한 LG U+와 기가IoT 홈매니저를 출시할 예정인 KT 그리고 다양한 제조사들과 협력해 홈 IoT 생태계 구축에 주력하고 있는 SKT 와 같은 통신사업자들이 제공하는 통신 연계 중심의 스마트 홈서비스가 있다[2].

사용자의 관심분야, 환경에 따라 요구사항은 달라지기 때문에 매우 다양한 측면에서의 데이터 수집과 분석이 필요하다. 즉, 요구사항 하나를 만족하기 위해 적게는 수개의 많게는 수십 가지의 센서가 필요하다. 이는 소비자들에게는 비용과 관리 등의 문제로 다가온다. 이러한 문제는 본 연구 “슈퍼 센서”로 극복될 수 있다. 슈퍼 센서는 여러 다양한 센서가 복합적으로 부착되어 있으며 단 하나의 개체의 설치로도 비교적 넓은 지역의 센싱이 가능하다. 사람의 눈에 띄지 않아 외관을 해치지 않으며 벽의 콘센트를 이용한 전원공급이 가능하다. 또한, 슈퍼 센서 시스템은 가공되지 않은 데이터를 행동 가능한 자료로 변화시켜 사용자의 요구사항에 대한 답변이 가능하다.

## 나) 연구논문/작품의 목표

현재 빈번하게 사용되는 센싱 방법들은 주로 센서와 facet간 일 대 일, 다 대 일, 다 대 다의 특징을 가진다. 즉, 감지되는 측면은 센서의 개수, 혹은 그 이하와 같다. 본 과제는 General-purpose Sensing 방식을 이용하여 센서와 facet 간 일 대 다, 다 대 다의 특징을 가진 이상적인 센싱 방법을 설계 및 구현하는 것이다.

## 다) 연구논문/작품 전체 overview

### 1. 서론

최근 사물들이 자동화되고 정보산업 기술과 융합되어 인간과 상호 연결되어가는 추세와 앞으로의 발전 방향을 언급하고 어떤 배경으로 인해 본 작품 연구의 시발점이 되었는지 서술한다.

또한 선행된 연구에서 제안된 방법과 결과를 간단하게 서술하고 개선하려는 연구 방향 및 목표를 제안한다.

### 2. 관련연구

먼저 하드웨어 측면에서 앞서 연구되어 온 센서의 다양한 센싱 방식을 서술한다. 각 방법이 어떤 상황에서 쓰이고 그에 따른 장·단점과 제한사항을 언급한다. 다음은 머신러닝에 관련한 소프트웨어 측면에서의 관련 연구를 서술한다. 본 작품은 분류학습에 기반을 두기 때문에 SVM(Support Vector Machine) 알고리즘과 인공신경

망(Neural Network; NN)의 연구에 대해 설명한다.

### 3. 제안 작품 소개

먼저 하드웨어 측면에서 본 작품에 사용하는 디바이스인 Arduino UNO, 각 센서의 명칭과 용도, 그리고 하드웨어의 전체적인 구성을 설명한다.

본 작품에서 사용하는 머신러닝 기법은 Neural Network로, Supervised Learning을 기초로 한다. 측정된 Data set이 Training set에 과도하게 overfit 되지 않게 lambda의 계수를 조정하였다.

가속도 센서의 경우 샘플링 속도에 따라 관측할 수 있는 행동의 범위가 광범위하다. 따라서 샘플링을 8kHz의 속도로 설정하여 미세한 진동 및 떨림까지 감지할 수 있게 하였다.

관측되는 데이터 값의 노이즈를 최소화하고, 분류기법시 더욱 유의미한 값을 갖게 하기 위해 데이터의 정형화를 이루었다. 조도, 온도, 습도, 음성 데이터의 경우 Event 발생 시 변화 값이 크기 때문에, 15번의 루프 전에 측정된 값과의 차(Error)를 Class에 포함시켰다. 음성 데이터의 경우 소리의 진폭을 측정하기 때문에 일정 시간동안 측정된 최대값과 최소값의 차인 Range를 반환한다. 색깔 센서는 각 R, G, B의 값을 10번씩 측정한 후 이의 평균값을 반환한다. 적외선 열상 감지 센서는 각 행과 열의 평균값을 반환하여 머신러닝의 정확도와 연관성을 증가시킨다.

센서에서 측정하는 데이터 값은 사물과 보드 사이의 거리에 민감하기 때문에 다양한 거리에 따라 측정을 실시하였다.

Event 센싱은 머신러닝의 정확도를 최대화하기 위해 Point Measurement 방법을 사용한다.

### 4. 구현 및 결과분석

본 작품의 데이터셋의 구조와 라벨별 유의미하게 작용하는 센서를 제시하고 분석한다. 학습에 사용한 인공신경망 모델은 30개와 25개로 구성된 두 개의 은닉층을 갖는다. 모델의 overfit을 방지하면서 cost를 최소화하는 최적의  $\theta^1$ 과  $\theta^2$ 는 Cost Gradient가 2000번 반복하며 학습된다. 데이터의 Regularization  $\lambda$  계수는 10으로 설정했다. Single Event와 Double-event를 학습하였다.

지금까지 수집해온 데이터를 통합하여 Feature Scaling을 적용시켜 학습한 결과의 정확도와 세부 사항들을 표로 나타낸다.

### 5. 결론

본 연구를 진행하며 느낀 점과 배운 점, 아쉬운 점을 제시한다.

## ■ 관련연구

### 가) Special-purpose sensor

Special-purpose sensor는 하나의 센서로 한 가지의 환경 측면을 관찰하는 센서다. 예를 들어 UpStream이나 WaterBot은 수도꼭지에 마이크를 부착해 물 사용 여부를 감지한다. 유사하게, HVAC는 방 안의 온도와 사람의 존재 여부를 확인한다. 이와 같이 특정 목적 센서는 문의 열림과 닫힘 감지나 공간에 사람이 있는지 같은 명확하고 저차원적인 센싱 문제들을 다루는 데 많이 사용된다[3].

### 나) Distributed Sensing System

Distributed Sensing System은 여러 Special-purpose sensor를 배치한 뒤 네트워크를 통해 연결해 환경을 감지한다. 동질성의 센서를 같이 배치함으로써 감지 범위를 확장할 수 있으며, 이질성의 센서를 배치함으로써 상호보완적인 데이터를 읽어 감지의 정확도를 높일 수도 있다[3].

### 다) Infrastructure-mediated Sensor

Infrastructure-mediated sensor는 배치 비용과 공간 침해를 줄이기 위해 고안된 방법이다. 센서를 시설의 전략적 조사 지점에 배치함으로써 변화를 감지한다. 예를 들어 집의 배수관에 설치된 센서는 거주자가 샤워를 하는지, 설거지를 하는지, 즉 수도의 사용을 감지할 수 있다. 이처럼 배관에만 설치해도 수도의 사용여부를 파악할 수 있기 때문에 위의 방법처럼 각 수도꼭지마다 센서를 설치할 필요가 없다. 따라서 많은 비용의 절감을 이루어 낼 수 있다[8].

### 라) Direct and Indirect Sensing

대부분의 센서는 Direct sensing 방식을 사용한다. 감지하고자 하는 사물에 물리적으로 접하여 변화를 감지한다. 이 방법은 높은 수준의 신호를 산출해 낼 수 있기 때문에 많이 채택되어 사용되고 있다. 하지만 전력을 공급받을만한 수단이 마땅치 않아 배터리를 사용해 주기적으로 충전을 해주어야 하는 문제점이 있다[9].

전력 문제가 해결된, 사물의 근처에만 설치하더라도 감지할 수 있는 방식도 있다. Indirect sensing은 전자기기의 모드가 바뀔 때 발생하는 노이즈를 인지해 사물의 변화를 감지한다. 하지만 전력과 공간적 문제가 해결되어도 결과의 정확도가 비교적 떨어진다는 단점이 있다.

### 마) General-purpose Sensing

General-purpose sensing은 다양한 용도의 융통성 있는 센서들을 센서 보드에 부착해 사용하는 것이다. 이 방식을 이용한 센싱은 사물 적용 범위가 넓으며, 사물이나 환경의 특징에 맞게 별다른 조정 없이 많은 측면을 감지할 수 있다.

## 바) Neural Network Model – Error Back Propagation Algorithm

다변수 비선형 문제에 가장 적합한 대표적인 인공신경망 알고리즘은 오류 역전파 학습 알고리즘(Error back-propagation training algorithm)이다. 오류 역전파 학습 알고리즘은 다층 퍼셉트론의 구조를 갖고 있으며 학습(Training) 단계와 회상(Recall) 단계로 동작한다. 학습 단계에서는 입력 패턴과 목적 패턴이 다수의 입출력 쌍으로 주어지고, 각 입력 패턴에 대해 출력을 먼저 계산한 후, 기대값과 출력값의 차이를 줄이는 방향으로 연결 강도를 조정한다. 조정 과정이 끝나면 같은 과정의 학습을 반복함으로써 최적의 연결 강도를 구한다. 회상 단계에서는 입력만 주어지면 반복 계산 없이 연결 강도와 계산에 의해 적절한 출력이 즉시 계산된다[9].

## 사) SVM(Support Vector Machine)

Support Vector Machine 알고리즘은 주로 Classification과 Regression에 사용되는 머신러닝 기법이다. SVM은 다른 두 집단에 속한 Data set이 주어졌을 때 이 데이터를 분리하는 최적의 Decision Boundary를 찾는다. 이 기법은 학습 데이터의 양이 적을 때 신속한 학습을 수행할 수 있으며, 불균형 데이터 집합에 대해서 우수한 성능을 보인다.

## ■ 제안 작품 소개

### 가) Hardware

센서 연결 보드는 필요한 센서의 종류와 개수를 고려하였을 때 적합하다고 판단한 Arduino UNO R3를 사용한다.

센서는 아날로그와 디지털, 두 종류의 센서로 구분된다. 아날로그 센서는 조도센서(Flying-Fish)와 음성센서(ADMP401), 총 2가지를 사용한다. 조도센서는 빛의 밝기를, 음성센서는 소리의 진폭을 측정한다.

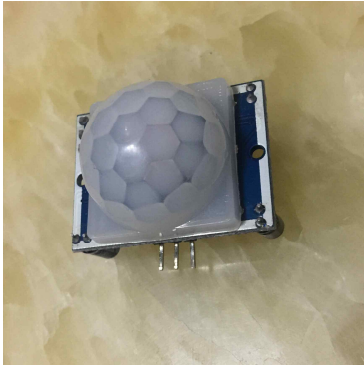


<조도센서>

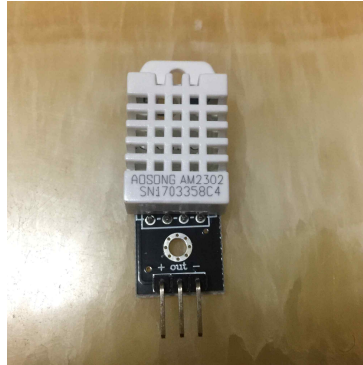


<음성센서>

디지털 센서는 온·습도센서(AM2302), 가속도센서(MPU6050), PIR센서(HC-SR501), 컬러센서(TCS3200), 적외선 열 감지 센서 Grideye(AMG8833), 총 5가지를 사용한다. 온·습도 센서는 환경의 온도와 습도를, 가속도 센서는 X, Y, Z 3축의 가속도를 측정한다. PIR센서는 적외선으로 인체의 움직임을 감지한다. 컬러센서는 Red, Green, Blue의 값을 측정하고, 적외선 열 감지 센서는 8X8개의 픽셀로 구성되어 열상카메라의 기능을 한다.



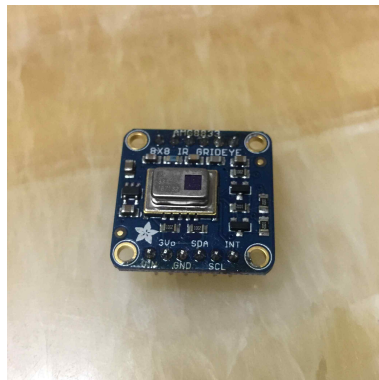
<PIR센서>



<온·습도센서>



<가속도센서>



<Grideye>



<컬러센서>

본 작품의 기본 구조는 아래의 그림과 같이 7개의 센서가 연결된 아두이노에서 PC로 데이터를 전송해 학습하는 방식을 사용한다.



#### 나) Machine Learning

기본적인 기계학습 방식은 지도학습(supervised learning)하에 진행된 Multi-Classification이다. 통제된 상황 아래(예를 들어, light on-off 또는 전화가 오는 상황 등) 수집된 데이터에 임의로 지정한 특정한 라벨(y값)을 붙인다. 이를 바탕으로 supervised learning을 진행하며 딥러닝 모델이 training set에 과도하게 overfit 되지 않으면서 동시에 cost를 최소한으로 줄일 수 있는, 최적화된 모델을 찾는다.

#### 다) Sample Rate

가속도 센서의 경우 아두이노 보드와 I2C(Inter Integrated Circuit) 방식으로 통신한다. 이 센서의 경우 샘플링 속도에 따라 관측되는 값이 다르고 매우 민감하기 때문에, Sample rate를 8kHz로 설정하였다. 그 외의 센서는 아두이노 보드와 시리얼 통신을 한다. Matlab에서의 통신 속도를 고려하여 딜레이를 최소화 시키기 위해 아두이노 보드의 Serial 통신 보드레이트는 19200bps로 설정하였다.

#### 라) 데이터 정형화

센서에서 측정되는 데이터 노이즈를 최소화하기 위해 여러 방법을 통해 데이터를 정형화하였다.

온도와 습도, 조도와 음성의 경우 앞서 측정한 데이터와의 차(Error)가 유의한 데이터로 적용되기 때문에 각 센서의 측정값 외에 이전 측정값과의 차를 반환하여 데이터로 출력한다. 하지만 온도와 습도의 경우 환경의 갑작스러운 변화가 있지는 않은 이상 변화의 폭이 작기 때문에 15번 전에 측정한 값과의 차를 반환한다.

음성센서의 경우 보드가 작동하는 Start time을 Millisecond 단위로 측정하여



50ms까지 데이터를 반복적으로 수집한다. 해당 시간동안 수집한 데이터는 최댓값과 최솟값의 차(Range)를 반환한다. 되는 진폭의 Range(Maximum-Minimum)를 반환한다.

컬러센서의 경우 보드 한 번의 작동 당 10번의 측정을 한다. 측정된 값은 각 R, G, B의 측정값으로 나뉘어 평균값을 반환한다.

Grideye의 경우 총 측정한 64개의 픽셀 값을 각 행과 열의 평균값을 내어 총 16개의 데이터를 반환하여 머신러닝의 정확도와 연관성을 증가시킨다.

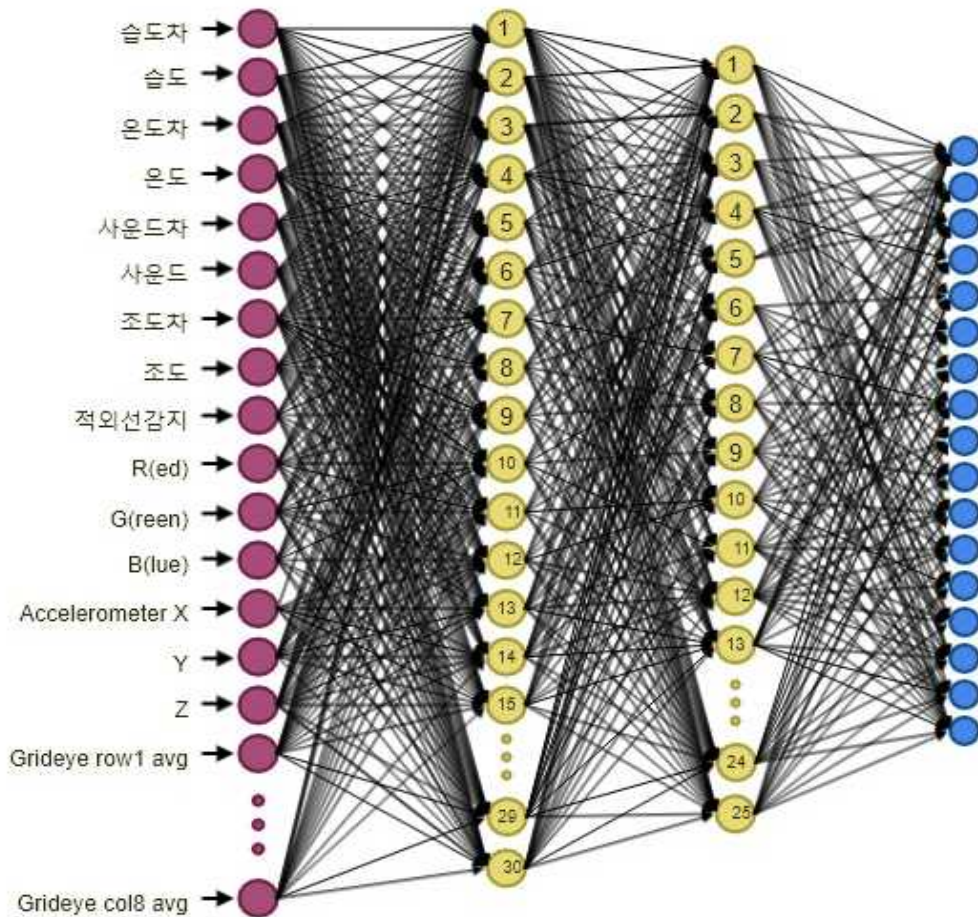
#### **마) Point Measurement**

전등의 켜고 끄, 문의 여닫음은 점으로, 전화벨소리는 선으로 카운터를 사용해 점과 선의 측정 클래스를 나누어 데이터를 수집하였으나, 벨소리의 종류는 다양하고 소리의 크기도 천차만별이기 때문에 선으로 측정할 경우 음성센서의 측정값이 낮은데도 전화벨이 울리는 것으로 인식되어 머신러닝의 정확도를 낮추는 영향을 끼친다. 따라서 데이터 수집은 모두 점으로 진행하였다.

## ■ 구현 및 결과분석

### 가) Neural Network

뉴럴 네트워크 알고리즘은 Octave에서 구현했다. 데이터는 Matlab형식을 그대로 사용하였고, Octave에서 로드가 가능하다.



완성된 인공신경망은 위의 그림과 같다. 31개의 feature들을 가진 Input data X는 2개의 Hidden layer를 거쳐 설정한 라벨들 중 하나로 output된다. 1개의 Hidden layer 1은 30개의 neural unit을 가지고 있으며, Hidden layer 2는 25개의 unit을 가진다. cost gradient를 2000번 반복하며 모델이 training set에 지나치게 overfit되지 않으면서 cost를 최소화하는 optimal  $\theta^1$ 과  $\theta^2$ 를 학습한다. Regularization을 위한  $\lambda$  계수는 5로 설정했다. Output은 라벨 1-9는 실내에서 자주 관찰될 수 있는 사용자의 행동패턴 중 single event를, 라벨 10부터는 1-9의 single 이벤트가 동시에 발생할 경우이다.

다음은 N.N를 구현한 알고리즘이다.

- 1) 데이터 로드 ( $X = m \times \text{feature 개수}$ ,  $y = m \times 1$ )
- 2) Scale features
- 3) 데이터를 Training set(70%)와 Test set(30%)으로 랜덤하게 나눈다.
- 4) Hidden layer size\_1 = 30
- 5) Hidden layer size\_2 = 25
- 6)  $\text{initial\_}\theta^1 = \text{random values in } [-\epsilon, \epsilon] (\text{input\_layer\_size} \times \text{hidden\_layer\_size\_1})$
- 7)  $\text{initial\_}\theta^2 =$   
 $\text{random values in } [-\epsilon, \epsilon] (\text{hidden\_layer\_size\_1} \times \text{hidden\_layer\_size\_2})$
- 8)  $\text{initial\_}\theta^3 = \text{random values in } [-\epsilon, \epsilon] (\text{hidden\_layer\_size\_2} \times \text{labels number})$
- 9)  $\text{initial\_}\theta = [\text{initial\_}\theta^1 (:); \text{initial\_}\theta^2 (:); \text{initial\_}\theta^3];$
- 10)  $\lambda = 5$
- 11)  $\text{cost } J = \text{nnCostFunction}(\text{initial\_}\theta, \text{input\_layer\_size}, \text{hidden\_layer\_size},$   
 $\dots \text{num\_labels}, \text{trainX}, \text{trainY}, \text{lambda});$
- 12)  $\text{costFunction} = @(p) \text{nnCostFunction}(p, \text{input\_layer\_size}, \dots$   
 $\text{hidden\_layer\_size}, \text{num\_labels}, \text{trainX}, \text{trainY}, \lambda);$
- 13)  $\text{options} = \text{optimset('MaxIter', 2000)};$
- 14)  $[\theta, \text{cost}] = \text{fmincg}(\text{costFunction}, \text{initial\_}\theta, \text{options});$
- 15)  $\theta^1, \theta^2, \theta^3 =$   
 $\theta (1 \times \text{total theta}) \rightarrow \theta^1 (\text{input} \times \text{hidden1}), \theta^2 (\text{hidden1} \times \text{hidden2}),$   
 $\theta^3 (\text{hidden2} \times \text{label}) \text{로 reshape.}$
- 16)  $\text{pred} = \text{predict}(\theta^1, \theta^2, \text{testX})$
- 17)  $\text{accuracy} = (\text{predict} == \text{test\_y}) \text{ 개수} / \text{test\_y의 개수}$
- 18)  $\text{confusion matrix} = \text{confusionmat}(\text{test\_y}, \text{predict})$

7)임의로 선택한  $\theta^1, \theta^2, \theta^3$ 를 initial\_θ로 벡터화시키며(열의 개수가 1) 통합한다.  
 8)λ는 model이 overfit되지 않도록 방지하기 위한 regularization의 계수이다.  
 9)nnCostFunction을 이용해 임의로 선택한 θ에서의 cost J값을 구하고 확인해본다.  
 10)nnCostFunction의 주소값을 저장하고, 11)gradient cost computation을 몇 번 반복할 것인가, 2000번으로 option 지정한다. 12)nncostFunction의 주소값과 임의로 선택했던 θ값, option값을 parameter로 받고 gradient cost computation을 2000번 반복한 후 학습된 θ와 cost를 리턴한다. 13)벡터화되어 있는 θ를 다시  $\theta^1 (\text{input\_layer\_size} \times \text{hidden\_layer\_size1})$ ,  $\theta^2 (\text{hidden\_layer\_size1} \times \text{hidden\_layer\_size2})$ ,  $\theta^3 (\text{hidden\_layer\_size2} \times \text{labels number})$ 로 reshape시킨다. 14)학습한 θ를 바탕으로 test\_X의 y를 predict한다. 15,16) test set의 accuracy와 confusion matrix를 확인한다. confusion matrix의 행은 실제 y값을 의미하고, 열은 예측된 y값, 즉 h(x)를 의미한다.

## 다) 결과분석

### Data set

Index	1	2	3	4	5	6	7	8	9	10-12	13-15	16-23	24-31	Total
Feat-ure	습도차	습도	온도차	온도	사운드차	사운드	조도차	조도	PIR	RGB	가속도 x,y,z	Grideye row means (1-8)	Grideye column means (1-8)	31
Label	1	2	3	4	5	6	7	8	9					
event	default	phone ring	tissue	door close	vacuum	faucet	light on	light off	stove					
size	5535	1843	348	187	2788	662	449	442	2421					
Label	10	11	12	13	14	15	16	17	Total					
event	2+7	2+8	2+9	7+9	8+9	5+7	5+8	5+9						
size	290	296	797	261	259	351	350	199	17478					

### 라벨 별 주된 Feature 분석

- 2)phone ring : 5)사운드차, 6)사운드  
전화벨이 울림에 따라 소리 센서가 민감하게 반응
- 3)tissue : 5)사운드차, 6)사운드, 9)PIR  
티슈 뽑을 때 나는 소리에 소리 센서와, 사람의 움직임에 PIR센서가 민감하게 반응
- 4)door close: 5)사운드차, 6)사운드, 13-15)가속도x,y,z  
문 닫힘에 발생하는 소리와 진동에 따라 소리 센서와 가속도 센서가 민감하게 반응
- 5)vacuum: 5)사운드차, 6)사운드, 13-15)가속도x,y,z  
진공청소기 작동 소리와 진동에 따라 소리 센서와 가속도 센서가 민감하게 반응
- 6)faucet: 1)습도차, 2)습도, 5)사운드차, 6)사운드, 13-15)가속도x,y,z  
수도를 틀 때 물이 떨어지는 소리와 진동에 소리 센서와 가속도 센서가 민감하게 반응, 일정기간 이상 수도를 사용할 시 습도의 변화를 보임
- 7,8) light : 7)조도차, 8)조도, 10-12)RGB  
전등을 켜고 끄에 따라 발생하는 빛의 세기와 R,G,B의 파장 변화에 조도 센서와 색깔 센서가 민감하게 반응
- 9) stove : 3)온도차, 4)온도, 16-23)Grideye row means, 24-31)Grideye column means  
가스레인지 불의 세기, 작동기간에 따라 온도 센서와 적외선 열상감지센서가 민감하게 반응

## 테스트 결과

- Accuracy = 96.22%
- Confusion matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1656	19	1	0	1	6	0	1	8	0	0	0	0	0	0	0	0
2	9	546	0	4	24	2	0	0	0	0	0	0	0	0	0	0	0
3	21	3	93	1	0	0	0	1	0	0	0	0	0	0	0	0	0
4	18	8	0	27	1	0	0	0	0	0	0	0	0	0	0	0	0
5	0	18	0	0	938	0	0	0	0	0	0	0	0	0	0	1	1
6	2	3	0	0	0	193	0	0	0	0	0	0	0	0	0	0	0
7	2	1	0	0	0	0	124	0	0	3	0	0	0	0	1	0	0
8	4	0	0	0	0	0	0	111	0	0	2	0	0	0	0	3	0
9	8	0	0	0	0	0	0	0	750	0	0	0	0	1	0	0	0
10	0	3	0	0	0	0	3	0	0	79	0	0	0	0	1	0	0
11	0	1	0	0	0	0	0	1	0	0	88	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	3	0	0	211	0	0	0	0	0
13	0	0	0	0	0	0	0	0	2	0	0	0	82	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	74	0	0	0
15	0	0	0	0	0	1	0	0	0	3	0	0	0	0	95	0	0
16	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	109	0
17	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	68

## Demonstration

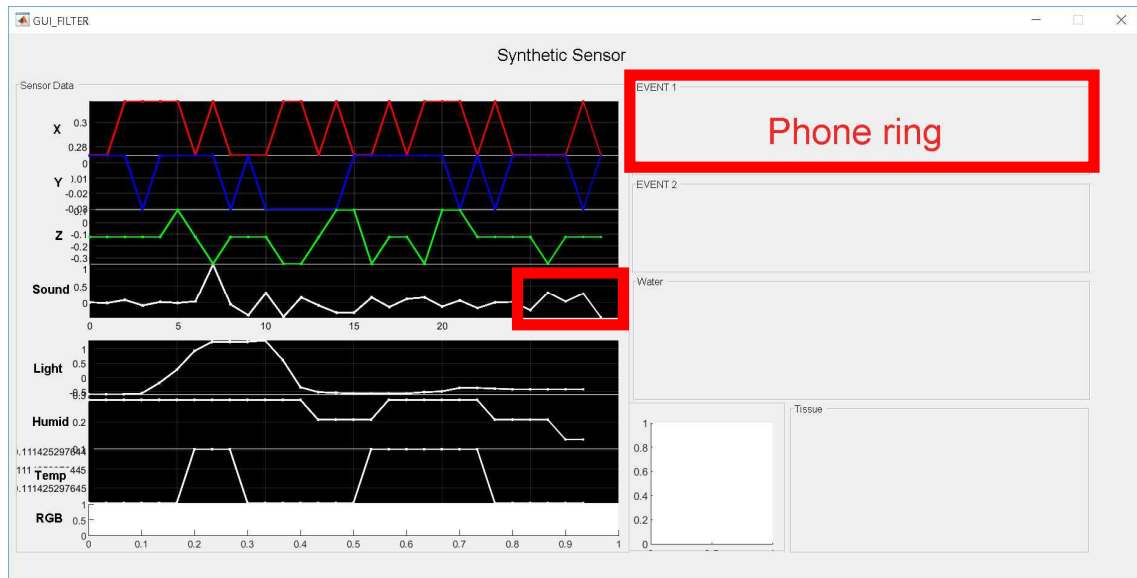
### Case 1. No event

- Label 1. No event

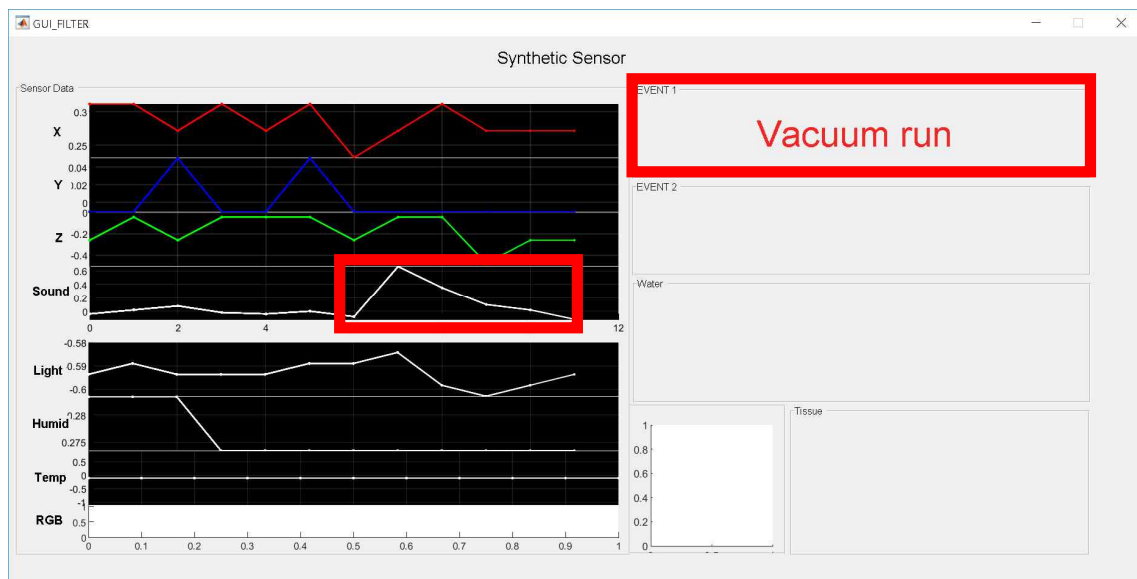


## Case 2. Single event

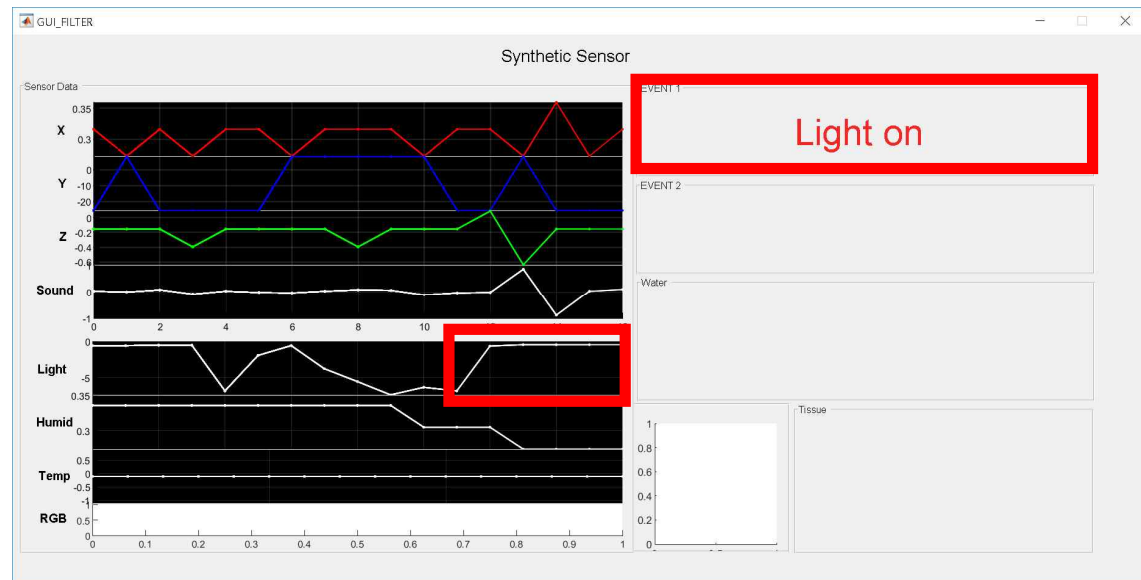
- Label 2. Phone ring



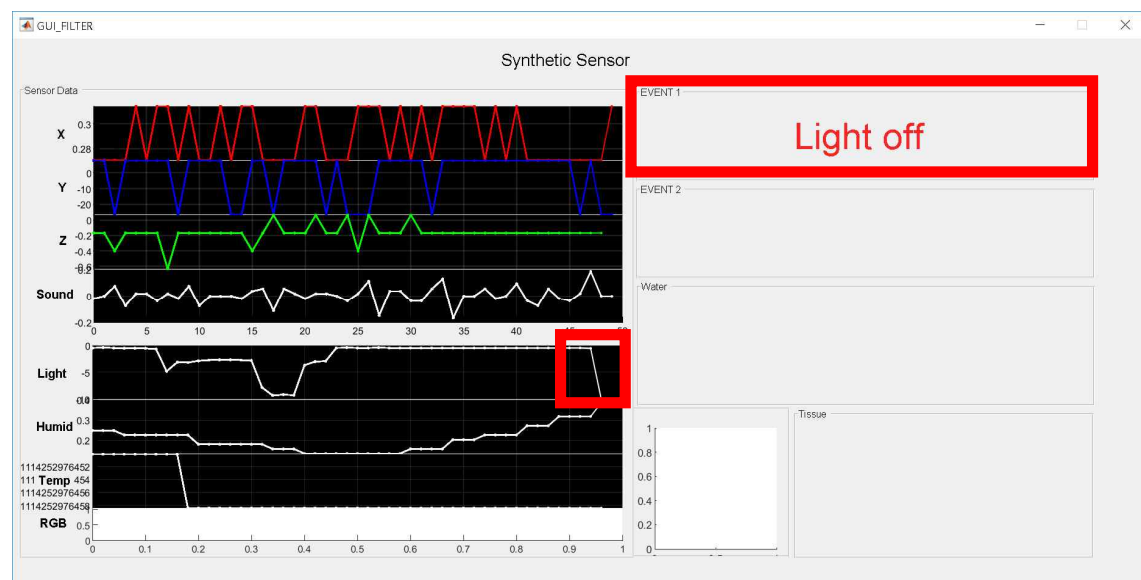
- Label 5. Vacuum run



- Label 7. Light on



- Label 8. Light off

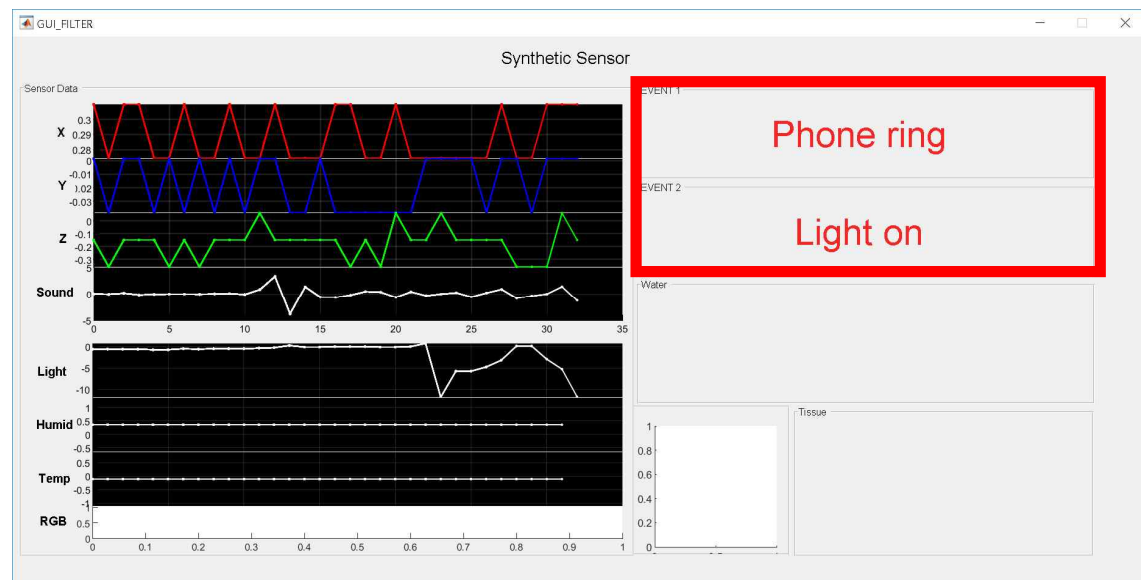


- Label 9. Stove



### Case 3. Multi-Events

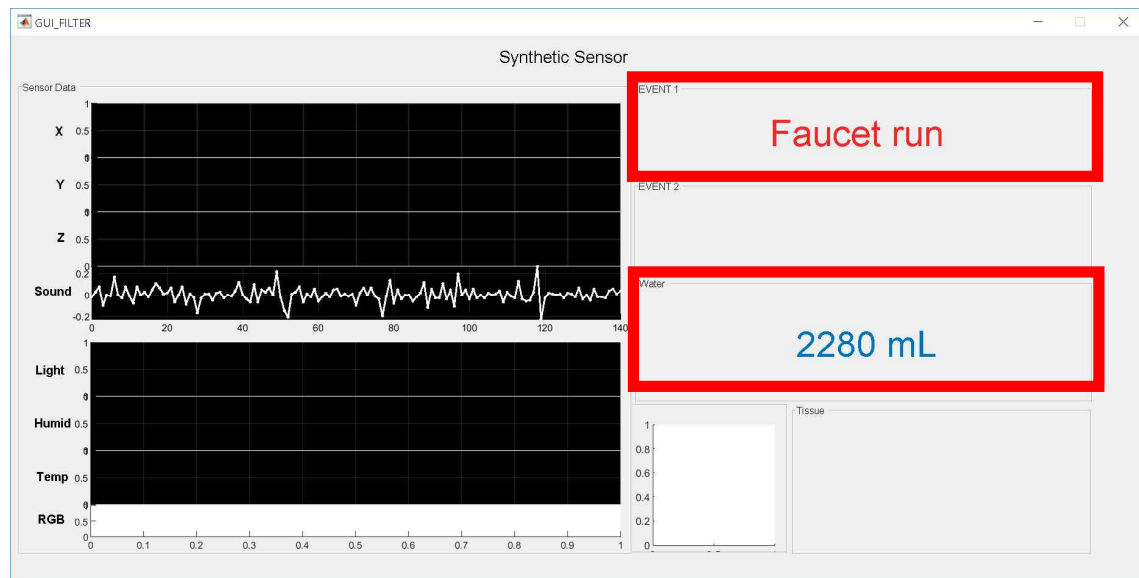
- label 10. Phone ring + Light on





#### Case 4. Application

- Label 6. Faucet run (사용한 물의 양 측정)



#### ■ 결론 및 소감

AI가 새로운 제 4차 산업혁명의 중심 기술로 급부상하면서 머신러닝을 배우고 실제로 구현해보기 위해 이 Synthetic sensor를 졸업 작품의 주제로 선정하게 되었다. 학교에서 딥러닝을 배울 기회가 없었기 때문에 작품 구현을 시작하기에 앞서 머신러닝 MOOC 강의를 들으며 스터디 하고 기본지식을 쌓는데 오랜 시간을 들였다. 머신러닝은 인간의 실제 생활과 밀접한 연관이 있으므로 예상치 못했던 것 하나하나가 큰 변수로 작용하며 벽에 자주 부딪혔다. 현실은 매우 복잡하고 쉽게 명료화시킬 수 없는 수많은 상호작용의 집합체였다. 같은 event이더라도 집에서, 카페에서, 학교에서 장소에 따라 데이터는 천차만별로 달라졌다. 또한, 대부분의 인간 행동은 single-event가 아니다. 대부분의 event는 동시에 다발적으로 일어난다. 따라서 머신러닝 모델이 실효성을 획득하기 위해선 multi-event가 일어났을 때 기계가 혼동 없이 모두 잡아낼 수 있어야 한다. 본 작품은 single-event와 double-event의 데이터를 정형화 과정을 거쳐 인공신경망을 통해 감독학습하였다. 감독된 event의 발생을 인지할 수 있으며, 측정 환경에 따라 약간의 오차를 보인다.

본 과제를 진행하며 많은 것을 배웠다. 우선, 앞서 작성한 것처럼 머신러닝 강의를 수강하며 다양한 머신러닝 이론과 기법을 배웠고, 아두이노라는 하드웨어와 여러 가지 센서를 다루는 법도 배웠다. 흥미로운 주제에 대한 새로운 학습을 하며 작

품 진행을 재미있게 할 수 있었다. 하지만 아쉬운 점도 많이 남는다. 프로세스를 진행함에 따라 현 단계에서 새로 깨닫거나 배운 내용을 이전 단계에서 미리 알았다면 시간을 더욱 효율적으로 사용하고 효과도 높일 수 있었을 거라는 생각을 자주 하게 되었다. 또한, 졸업 작품 제출이라는 제한적인 시간 안에 최고의 성과를 내지 못한 것 같아 더욱 아쉬움이 따른다.

## ■ 참고문헌

- [1] 김영관, "스마트홈, (홈IoT) 성공의 전제조건", 디지에코 보고서, 2014년 7월
- [2] 김영관, "스마트홈(홈IoT) 생태계 6개 구성요소", 디지에코 보고서, 2014년 11월.
- [3] Gierad Laput, Yang Zhang, Chris Harrison(2017), Synthetic Sensors: Towards General-Purpose Sensing, Carnegie Mellon University Human-Computer Interaction Institute
- [4] Stacey Kuznetsov and Eric Paulos, "UpStream: motivating water conservation with low-cost water flow sensing and persuasive displays", Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2010.
- [5] Ernesto Arroyo, Leonardo Bonanni, and Ted Selker, "Waterbot: exploring feedback and persuasive techniques at the sink", Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2005.
- [6] Neil Klingensmith, Joseph Bomber, and Suman Banerjee, "Hot, cold and in between: enabling fine-grained environmental control in homes for efficiency and comfort", Proceedings of the 5th international conference on Future energy systems, 2014.
- [7] James Scott, A.J. Bernheim Brush, John Krumm, Brian Meyers, Michael Hazas, Stephen Hodges, and Nicolas Villar, "PreHeat: controlling home heating using occupancy prediction", Proceedings of the 13th international conference on Ubiquitous computing, 2011.
- [8] Jason Hill, Mike Horton, Ralph Kling, and Lakshman Krishnamurthy, "The platforms enabling wireless sensor networks" Communications of the ACM 47, 2004년 6월
- [9] 신동철 "사무실의 피난소요시간 산정을 위한 인공지능경망 모델 개발", 한국방재학회 논문집 제14권 6호, 2014년 12월