

2017학년도 창의융합 과제연구 보고서

퍼셉트론 인공 신경망을 활용한 약 처방 자동화 시스템

출품 부문	정보
-------	----

2017. . .

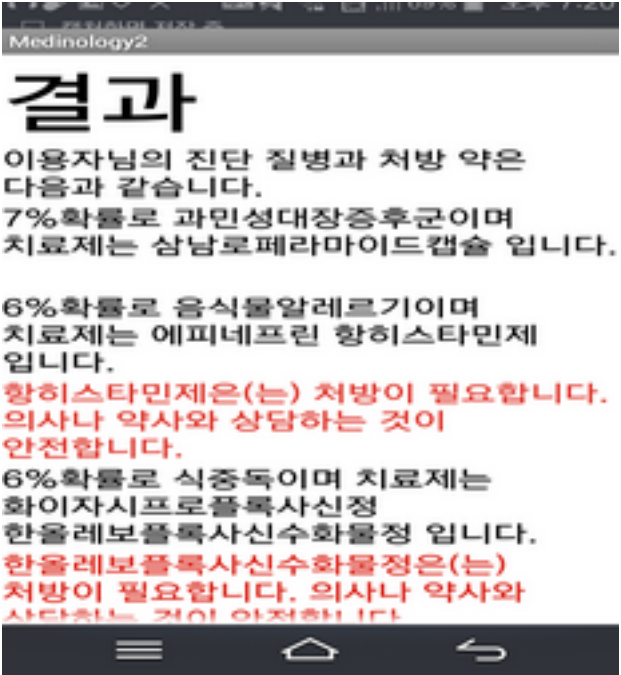
팀원	(2)학년(14)반 ()번 이름(양현서)
	(2)학년(12)반 ()번 이름(이현빈)
	(2)학년(12)반 ()번 이름(이승현)
지도 교사	(인)

※ 비고

1. 작 품 명 : 간단명료하게 작품의 내용과 방법을 잘 반영해야 함
2. 출품 부문 : 물리, 화학, 동물, 식물, 지구과학, 농림수산, 산업 및 에너지, 환경, 융합, 사회 등
3. 최종 보고서는 A4 용지에 표지서식 포함 15~20쪽 분량으로 작성
4. 본문은 작품연구의 동기 및 목적, 선행연구 고찰, 연구 절차 및 방법, 작품의 내용, 결론, 전망 및 활용성 등을 체계적으로 기술.

(신명조(영문 : Times New Roman) 11pt, 줄간격 160, 여백 좌우 20, 상하 25를 기준으로 함)

5. 파일명 : 보고서(보고서제목_학생명).hwp

2017학년도 창의융합 과제연구 작품요약서	
출품부문	작 품 명
정보	퍼셉트론 인공 신경망을 활용한 약처방자동화 시스템
<div> <div> <h3>1. 탐구(연구) 동기</h3> <p>인공지능이 각광 받고 있는 이 시대에 인공지능을 공부하다 우리가 공부한 것들을 바탕으로 누군가에게 도움이 되는 무언가를 만들 수 있지 않을까 하는 고민에서 이 연구를 시작하게 되었다. 늦은 밤 혹은 약국이 없는 외진 지역에서 어떤 약을 구매해야할지 몰라 약을 구매하지 못하거나 잘못된 정보로 약을 구매하는 것을 보았다. 잘못된 약 복용으로 부작용이 생기거나 약을 구매하지 못해서 생기는 사고를 방지하기 위해 시스템을 개발 하고 싶다는 생각을 하게 되었다. 증상별로 약을 추천해줄 수 있는 방법을 고민하다가 퍼셉트론을 이용한 인공신경망을 떠올리게 되었다.</p> </div> <div>  <p>결과 이용자들의 진단 질병과 처방 약은 다음과 같습니다. 7%확률로 과민성대장증후군이며 치료제는 삼남로페라마이드캡슐 입니다. 6%확률로 음식물알레르기이며 치료제는 에피네프린 항히스타민제 입니다. 항히스타민제는(는) 처방이 필요합니다. 의사나 약사와 상담하는 것이 안전합니다. 6%확률로 식중독이며 치료제는 화이자시프로플록사신정 한올레보플록사신수화물정 입니다. 한올레보플록사신수화물정은(는) 처방이 필요합니다. 의사나 약사와 상담하는 것이 안전합니다.</p> </div> </div> <div> <h3>2. 탐구(연구) 내용</h3> <h4>가. 선행 연구 고찰 및 탐구의 독창성</h4> <p>dbpia, krpia, 구글 검색등을 통해 약처방 자동화 시스템과 관련된 여러 선행연구를 찾으려 했으나 찾을 수 없었다. 비슷한 것이라고 해봐야 처방전을 넣으면 약을 주거나 약을 잘라서 봉투에 담아주는 기계 정도였다. 그래서 우리의 탐구가 더 독창적이고 필요한 것이라 생각된다.</p> <h4>나. 탐구 절차 및 방법</h4> <p>퍼셉트론 인공신경망을 제작하는 데에 있어 원래는 퍼셉트론이라는 단위 클래스를 기반으로 전체 신경망을 구성하려 하였으나 참고 문헌을 보고 나서 이 방법이 너무 비효율적임을 깨닫고 행렬을 이용한 방법으로 방향을 바꾸었다. 기본적인 베이스 프로그램은 Visual Basic을 이용해 코딩하였고, 사용자가 데이터를 입력하면 입력된 데이터를 C++로 코딩된 퍼셉트론 클래스로 전달 후 퍼셉트론 클래스에서 출력된 데이터를 변환해 사용자에게 총 3순위의 질병 및 확률, 처방될 약과 주의할 점을 나타내는 순서로 시스템을 구성하였다.</p> <h4>다. 작품의 주요 내용</h4> <p>우리 탐구의 가장 핵심내용은 C++을 이용한 퍼셉트론이라고 할 수 있다.medinology_cpp.exe 즉 인공신경망은 다음과 같은 부분으로 나누어진다.</p> <ul style="list-style-type: none"> -VB에서 전달된 환자의 건강 정보 자료를 파싱하는 부분 -weight.txt 파일의 내용을 신경망의 가중치, 편향에 로드하는 부분 -신경망을 생성하여 초기화하고 Predict함수를 호출하여 결과를 계산하는 부분 -계산 결과를 분석하여 질병 명을 확률 순서대로 정렬하여 파일에 출력하는 부분 <h3>3. 탐구(연구) 결과</h3> <p>실제로 앱을 완성하고 구동시켜본 결과 증상에 맞는 질병과 약이 나오지만 확률이 낮은 부분을 확인 할 수 있었다. 이는 짧은 기간 수작업으로 데이터를 생성한 만큼 절대적인 데이터 량의 부족으로 인한 것으로 생각된다.</p> </div>	

목차

서론

-탐구동기 및 목적

-이론적 배경

본론

-비주얼 베이직 언어의 역할과 원리

-학습데이터 제작

-어플리케이션의 제작

-c++언어의 역할과 전체적인 구동의 흐름

-어플리케이션 구동

결론

-기대효과

-보완할 점

-참고문헌

서론

탐구의 목적 및 동기

인공지능이 각광 받고 있는 이 시대에 인공지능을 공부하다 우리가 공부한 것들을 바탕으로 누군가에게

도움이 되는 무언가를 만들 수 있지 않을까 하는 고민에서 이 연구를 시작하게 되었다. 늦은 밤 혹은 약국이 없는 외진 지역에서 약을 구매해야하는데 어떤 약을 구매해야할지 몰라 약을 구매하지 못하거나 잘못된 정보로 약을 구매하는 것을 보았다. 잘못된 약 복용으로 부작용이 생기거나 약을 구매하지 못해서 생기는 사고를 방지하기 위해 시스템을 개발 하고 싶다는 생각을 하게 되었다.

증상별로 약을 추천해줄 수 있는 방법을 고민하다가 퍼셉트론을 이용한 인공신경망을 떠올리게 되었다. 퍼셉트론 인공신경망은 학습을 통해 판단이 가능하므로 여러 임상사례들을 학습시켜 약을 처방해줄 수 있을 것이라 판단하였다. 휴리스틱(사람스러운) 기능을 가지고 있는 약 처방 프로그램을 개발하여 어떤 약을 선택해야할지 모르겠는 상황 또는 병원을 좋아하지 않으시는 어르신들이 병원을 가도록 경고를 해주거나 하는 인공지능 시스템을 개발하여 사람들을 편리하게 하고 건강을 유지하는데 도움을 줌을 목적으로 한다.

이론적 배경

퍼셉트론 인공신경망

퍼셉트론이란 입력단자가 여러 개이고 출력 단자가 한 개인 객체이다. 퍼셉트론은 각 입력마다 가중치가 있어서 여러 개의 입력 값을 주면 각 단자의 입력값과 가중치를 곱해 모두 더한 값을 구해 가중 합을 정하고 고정된 임계치보다 가중 합이 작으면 0, 그렇지 않으면 1과 같은 방식으로 출력값을 내보낸다. 이러한 퍼셉트론은 단순한 판정만이 가능할 뿐인데 다수의 퍼셉트론을 여러 계층으로 배열하여 인공신경망을 만들 수 있다. 이렇게 만든 인공신경망을 학습시키기 위해서는 먼저 학습을 위한 입력값들 즉 학습데이터를 만들어야 한다. 학습 단계에서는 학습데이터와 정답값을 함께 인공 신경망에 제공하고 출력값을 구한 후 정답에 해당하는 값과 차이가 줄어들도록 가중치를 갱신한다. 어떤 학습데이터가 주어지면 이 때의 출력값을 구하고 학습데이터와 함께 제공된 정답에 해당하는 값에서 출력값을 뺀 값 즉 오차 값을 구한다. 이 오차 값의 일부가 출력단자에서 입력 단자로 되돌아가면서 각 계층의 퍼셉트론 별로 출력 신호를 만드는데 관여한 모든 가중치들에 더해지는 방식으로 가중치들이 갱신된다. 이러한 과정을 다양한 학습 데이터에 대하여 반복하면 출력값들이 각각의 정답값에 수렴하게 되어 판정 성능이 좋아지게 되고 본연구에서는 이러한 퍼셉트론을 여러 계층으로 배열한 인공신경망을 제작하고 각 증상, 병원체별 학습데이터를 제작, 학습을 시켜 각 상황에 알맞은 약을 처방할 수 있는 시스템을 만든다.

딥러닝

컴퓨터가 여러 데이터를 이용해 마치 사람처럼 스스로 학습할 수 있게 하기 위해 인공 신경망(ANN: artificial neural network)을 기반으로 구축한 한 기계 학습 기술이다.

딥 러닝은 인간의 두뇌가 수많은 데이터 속에서 패턴을 발견한 뒤 사물을 구분하는 정보처리 방식을 모방해 컴퓨터가 사물을 분별하도록 기계를 학습시킨다.

딥 러닝 기술을 적용하면 사람이 모든 판단 기준을 정해주지 않아도 컴퓨터가 스스로 인지·추론·판단할 수 있게 된다. 음성·이미지 인식과 사진 분석 등에 광범위하게 활용된다.

본 연구에서는 퍼셉트론 인공신경망에 여러 가지 병에관한 학습데이터를 제공해 딥러닝을 하고 약을 처방한다.

시그모이드 함수

시그모이드 함수(Sigmoid function)는 S와 같은 형태(Sigmoid curve)를 가진 함수를 말한다. . 주로 학습 곡선등을 나타내는 함수로, 0에 가까운 작은 값에서 일정한 유한값에 접근하는 함수이다. 1과 0사이를 부드럽게 이어준다. 특징은 x가 어떤값이어도 바로 1혹은 0으로 값을 얻어낼수 있다는 것이다. 주로 어떤

현상을 단순화하여 1과 0으로 놓고, 그 사이값으로 확률을 추론하는데에 많이 사용된다. 함수식은 다음과 같다.

< 그림1> 시그모이드 함수 구현

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  #####
5  ## 시그모이드
6
7  def sigmoid(x):
8      return 1 / (1 + np.exp(-x)) # numpy의 브로드캐스트
9
10 x = np.array([-1.0, 1.0, 2.0])
11 #print(np.exp(-x))
12 y = sigmoid(x)
13 print(y)
14
15 x = np.arange(-10, 10, 0.1)
16 y = sigmoid(x)
17 plt.plot(x,y)
18 plt.ylim(-0.1, 1.1)
19 plt.show()
```

본론

visual basic언어의 역할과 원리

Visual Basic으로 코딩된 Medinology는 기본적인 베이스 프로그램으로 사용자가 질병과 약 처방을 결정할 수 있는 정보들을 이 프로그램에 입력하면 입력된 데이터를 C++로 코딩된 퍼셉트론 클래스로 전달 후 퍼셉트론 클래스에서 출력된 데이터를 변환해 사용자에게 총 3순위의 질병 및 확률, 처방될 약과 주의할 점을 나타낸다.

Medinology.exe를 실행하면 사용자가 성별, 나이, 임신 여부, 증상, 과거 병력, 현재 투약중인 약과 같은 정보를 입력할 수 있는 폼이 나타난다. 입력이 완료되면 System.IO.StreamWriter 클래스를 사용해 Input.WriteLine으로 사용자에게 입력받은 정보를 Input.txt에 저장한다. 이 후 Shell("Medinology_cpp.exe")로 퍼셉트론 클래스인 Medinology_cpp.exe를 실행시켜 출력을 Output.txt로 받게 된다. 세 가지의 병 코드와 확률이 입력되어 있는 Output.txt를 System.IO.StreamReader 클래스를 사용해 Output.txt 내용을 읽어 들인다. 읽어 들인 내용을 결과 폼에서 If문을 사용하여 변환된 질병 및 확률과 그에 따라 처방될 약이 출력되고 만약 임신부, 어린이, 어르신에게 위험한 약이라면 의사 및 약사에게 상담 받아야 한다는 권고 또한 출력된다. 실제로는 매우 방대한 양이지만 가장 핵심부분의 일부 소스코드는 다음과 같다.

'//Form_Input.vb에서 Input.txt를 쓰는 과정

```
Dim Input As New System.IO.StreamWriter("Input.txt", False)
Input.WriteLine(Gender)
```

```

Input.WriteLine(Age)
Input.WriteLine(Preg)
Input.WriteLine(Symp01 & Symp02 & Symp03 & Symp04 & Symp05 & Symp06 & Symp07
& Symp08 & Symp09 & Symp10 & Symp11 & Symp12 &
    Symp13 & Symp14 & Symp15 & Symp16 & Symp17 & Symp18 & Symp19 &
    중략
    Symp49 & Symp50 & Symp51)
Input.WriteLine(His01 & His02 & His03 & His04 & His05 & His06 & His07 & His08 & His09
& His10 & His11)
Input.WriteLine(Medi01 & Medi02 & Medi03 & Medi04 & Medi05 & Medi06 & Medi07 &
Medi08 & Medi09 & Medi10 & Medi11)
Input.Close()

Me.Hide()
Form_Process.Show()

```

```

'//Form_Process.vb에서 Medinology_cpp.exe를 실행하는 과정
Private Sub Form_Process_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Shell("Medinology_cpp.exe")
End Sub

```

```

'//Form_Output.vb에서 Output.txt를 읽는 과정
Dim Output As New System.IO.StreamReader("Output.txt")
Dim Per1 As Integer = Output.ReadLine()
Dim Di1 As Integer = Output.ReadLine()
Dim Per2 As Integer = Output.ReadLine()
Dim Di2 As Integer = Output.ReadLine()
Dim Per3 As Integer = Output.ReadLine()
Dim Di3 As Integer = Output.ReadLine()

```

```

'//Form_Output.vb에서 1순위 질병과 처방 약과 주의해야할 점들을 출력하는 과정
If Di1 = 0 And Preg = "F" And Age >= 13 Then
    Label_1_Di.Text = "비브리오 불니피쿠스 감염증"
    Label_1_Medi.Text = "크라포란주, 독시사이클린"
    Label_1_Warn1.Text = "병원 처방이 필요합니다."
Elseif Di1 = 0 And Age < 13 Then
    Label_1_Di.Text = "비브리오 불니피쿠스 감염증"
    Label_1_Medi.Text = "크라포란주, 독시사이클린"
    Label_1_Warn1.Text = "크라포란주 = 병원 처방이 필요합니다."
    Label_1_Warn2.Text = "독시사이클린 = 병원 처방이 필요합니다. 어린이에게 위험하므로 의사
    및 약사와 상담이 필요합니다."
Elseif Di1 = 0 And Preg = "T" Then
    Label_1_Di.Text = "비브리오 불니피쿠스 감염증"
    Label_1_Medi.Text = "크라포란주, 독시사이클린"
    Label_1_Warn1.Text = "병원 처방이 필요합니다. 임산부에게 위험하므로 의사 및 약사와 상담
    이 필요합니다."

```

```

Elseif Di1 = 1 And Preg = "F" And Age >= 13 Then
    Label_1_Di.Text = "렙토스피라병"
    Label_1_Medi.Text = "독시사이클린, 앰실린캡슐"
    Label_1_Warn1.Text = "병원 처방이 필요합니다."
Elseif Di1 = 1 And Age < 13 Then
    Label_1_Di.Text = "렙토스피라병"
    Label_1_Medi.Text = "독시사이클린, 앰실린캡슐"
    Label_1_Warn1.Text = "독시사이클린 = 병원 처방이 필요합니다. 어린이에게 위험하므로 의사
    및 약사와 상담이 필요합니다."
    Label_1_Warn2.Text = "앰실린캡슐 = 병원 처방이 필요합니다.

```

학습데이터

퍼셉트론 인공신경망을 학습시켜주기 위해서는 학습데이터가 필요하다. 학습데이터는 증상-질병, 질병-약, 약-특징의 3가지 행렬데이터를 엑셀을 통해 제작하였다. 1은 해당항목에 해당됨을 0은 해당되지않음을 의미한다.

학습데이터중 증상-질병의 원핫 인코딩데이터의 일부는 다음과 같다

〈표1〉

	비브리오 블니피쿠스 감염증	렙토스피라 병	대상포진	식중독
가려움증	0	0	0	0
객담	0	0	0	0
객혈	0	0	0	0
관절통	0	0	0	0
구취'	0	0	0	0
구토	0	1	0	1
권태감	0	0	0	0
근육통	1	1	0	0
기침	0	0	0	0
긴박뇨	0	0	0	0
난청	0	0	0	0
눈부심	0	0	0	0
두드러기	0	0	0	0
두통	0	1	0	0
딱지	0	0	0	0
만성피로	0	0	0	0
무좀	0	0	0	0
반점	0	0	0	0

다음은 질병-약 학습데이터의 일부이다.

〈표2〉

	크라포란주	독시사이클린	엠실린캡슐	대유아씨클로버겔
비브리오 불니피쿠스 감염증	1	1	0	0
렙토스피라병	0	1	1	0
대상포진	0	0	0	1
식중독	0	0	0	0
장티푸스	0	0	0	0
췌장염	0	1	0	0
헤르페스 구순염	0	0	0	1
부신피질 기능부전증	0	0	0	0
감염성 장염	0	0	0	0
과민성 대장 증후군	0	0	0	0
간염	0	0	0	0
급성 담낭염	0	0	0	0
급성췌장염	0	0	0	0

아래는 약-특징의 학습데이터의 일부이다.

〈표3〉

	처방필요 여부	임산부 위험	어린이 위험	노약자 위험
크라포란주	1	1	0	0
독시사이클린	1	1	1	0
엠실린 캡슐	1	1	0	0
대유아씨클로버겔	0	1	1	0
대한팜씨클로버정	1	1	1	0
화이자시프로플록사신정	1	1	1	0
한올레보플록사신수화물정	1	1	1	0
세트리악손주	1	1	0	0
미노신킨캡슐	1	1	1	0
제이알히드로코르티손정	1	1	0	0
씨제이메트로니다졸	1	1	0	0
삼남로페라마이드 캡슐	0	1	0	1
우루사정	0	1	1	0
이미페넴	1	1	1	0

어플리케이션 제작

이 프로젝트는 편리성을 중점으로 두고 개발하는 것이므로 모바일 버전도 있으면 좋을것 같다고 판단했다. 그래서 제작해놓은 퍼셉트론 클래스로 앱을 제작하기로 결정하였다.

안드로이드 개발환경은 PC의 Android Studio와 모바일의 AIDE가 있다. 어플리케이션에 대해 간단히 설명하자면

- 1.첫 번째 화면에서 환자의 기초 정보를 수집한다.
- 2.두 번째 화면에서는 환자의 증상을 고르게 한다.

3. 세 번째 화면에서는 환자의 질병명과 그 확률, 그리고 그에 맞는 약제들의 이름을 순서대로 출력하고, 노인같은 저시력자들을 위해 음성으로 결과를 읽어준다.

이때 면책 조항도 나오는데 이는 이 프로그램이 아직 초기 버전이므로 예측이 틀릴 수 있고, 이것이 중대한 결과를 초래할 수 있기 때문에 주의하고 병원에 가라는 내용이 담겨 있다.

기술을 상세히 설명해보겠다. 이 애플리케이션은 C++로 구현되는 PC 버전을 포팅하는 것이므로 안드로이드 NDK와 JAVA의 JNI(Java Native Interface)를 활용하였다.

안드로이드의 프로그램 시작점은 MainActivity 클래스의 onCreate() 메소드이다. 이 메소드에서는 주로 앱의 화면과 데이터를 초기화한다 그리고 .res/ 디렉토리의 layout xml 파일 리소스를 화면에 전개하는 일을 한다. 이것이 무슨 말이나 하면, 안드로이드에서는 화면 모양을 정의하는 파일이 데이터로 따로 존재하는데 이 화면 모양을 레이아웃이라 하고 이것을 담은 파일 내용대로 안드로이드 운영체제에게 표시해 달라고 한다는 의미이다. 그런데 본 애플리케이션의 MainActivity에서는 레이아웃을 담은 파일조차 없다. 왜냐하면 초기 Medinology에서는 xml을 이용하여 화면 레이아웃을 정의하였었다. 이후 화면에 뿌려진 View(버튼, 텍스트 등)들의 id를 얻어오고 그들에게 여러 가지 초기화를 실행했다. 하지만 실행시 매번 onCreate에서 크래시가 발생하는 상황이 발생했다. 그래서 하나의 해결방안으로 동적으로 뷰를 생성하는 방식을 선택하게 되었다.

c++언어의 역할과 구동의 흐름

c++은 VB에서 input을 받아 Shell 함수로 medinology_cpp.exe 파일을 호출한다.

/> > VB에서 input을 받아 파싱하여 인공지능망 predict 함수의 입력값을 생성한다.

main 함수에서 input.txt를 분리한 후 parse 함수, do 함수, save 함수를 차례로 호출한다.

parse 함수는 main에서 잘라놓은 input.txt의 라인들을 파싱하여 증상, 나이, 성별 등의 데이터를 추출한다.

do 함수는 LoadWeights 함수에서 퍼셉트론의 가중치를 담고 있는 weight.txt 파일을 로드하여 신경망의 가중치를 로드하고, predict 함수로 질병을 예측한 후, 그것을 분석하여 확률과 병 인덱스를 구한다.

save 함수는 계산된 결과를 VB를 위한 형식에 맞게 output.txt에 출력한다.

Trainer 흐름

trainer는 실제로 가중치를 이용해 퍼셉트론을 학습시키는 일을 한다, 이 과정을 흔히 딥러닝이라고 부른다. main 함수에서는 데이터베이스를 읽는 LoadDataSet(), 실제 신경망을 학습시키는 Train(), 학습된 가중치를 파일에 기록하는 SaveWeights 함수를 차례로 호출한다.

LoadDataSet 함수는 인터넷에서 csv 파일을 파싱하는 소스를 구해 왔다. 이를 이용하여 symdis.csv를 파싱하여 이 결과를 dataset 2차원 배열에 저장한다.

Train 함수는 신경망의 그라디언트 구하는 함수를 호출하여 신경망의 인풋 값에 따른 기울기를 구하게 한 후 getXXX 함수들로 그 값을 구해오고, 그 값을 이용하여 신경망의 가중치, 편향을 갱신한다. Train_sub 함수가 실제 일을 하고 train 함수는 이 과정을 n번 반복시킨다.

Save 함수는 Train 함수에서 업데이트된 신경망의 가중치를 파일에 기록한다. 파일 형식은 행렬의 차원을 첫 줄에 적고 가중치를 개행으로 구별하여 출력하는 것이다.

안드로이드 애플리케이션 실행 흐름

기본 구조는 java side에서는 사용자 입력과 입출력 처리, 그리고 결과 보여주는 일을 한다.

C++ side에서는 실제 예측과 결과 리턴을 한다.

전체적인 구동의 흐름

안드로이드 앱이 처음 실행되면 호출되는 메소드는 MainActivity class의 onCreate라고 해도 무방하다. 이 메소드에서는 화면을 초기화하고 데이터베이스를 로드하는 일을 한다. 보통 안드로이드 앱들의 OnCr

ate에서는 XML 레이아웃 id를 사용하여 안드로이드 운영체제에게 화면을 구성하도록 지시하는데, medinology에서는 아래쪽에 설명될 버그들로 인해 버그를 고치는 과정에서 동적으로 모든 뷰를 생성하는 방식으로 바꾸었다.

OnCreate메소드는 뷰를 생성하고 초기화하며, 데이터베이스를 로드하는 메소드들을 호출한다.

소스코드의 일부는 다음과 같다. (94~101행)

```
scrollView.addView(inlinearLayout);
    firstFrame.addView(scrollView);
    mainLayout.addView(firstFrame);
    setContentView(mainLayout);
    LoadSymptoms();
    LoadDiseaseAndMedinames();
    ReadDisease_Drug(diseaseNames.size(), mediNames.size());
    ReadDrug_Detail(mediNames.size,4);
```

LoadSymptoms 메소드에서는 증상명들을 로드하고 이에 따라 체크박스 증상선택 화면을 구성한다. 이 메소드의 일부(151~190행 - 파일 파싱)

```
try
{
    BufferedReader br = new BufferedReader(new
InputStreamReader(getResources().openRawResource(R.raw.symptoms), "euc-kr"));

    String line;
    String kind,sym;
    ArrayList<String> buf=null;
    while ((line = br.readLine()) != null)
    {
        buf = new ArrayList<String>();
        if (i % 2 == 0)
        {
            kind = line;
            kinds.add(kind);
            if (i != 0)
            {
                symptomNames.add(new
ArrayList<String>(buf));
                //buf.clear();
            }
        }
        else
        {
            line.replaceAll(" ", " ");
            StringTokenizer tokenizer=new StringTokenizer(line, " ");
            while (tokenizer.hasMoreTokens())
            {
                sym = tokenizer.nextToken(" ");
                buf.add(sym);
            }
        }
        ++i;
    }
    if (buf != null)
    {
```

```
symptomNames.add(new ArrayList<String>());
```

200~ 253행 - 실제로 뷰 생성후 숨기기
중략

```
for (int j=0;j < sz;++j)
{
    String symnam=kinds.get(j);
    TextView tv=new TextView(this);
    tv.setText(symnam);
    tv.setTextSize(20);
    FrameLayout frame=new FrameLayout(this);
    ScrollView scr=new ScrollView(this);
    GridLayout grid=new GridLayout(this);
    grid.setColumnCount(3);
    grid.setOrientation(grid.HORIZONTAL);
    GridLayout.LayoutParams par=new GridLayout.LayoutParams();
    GridLayout.Spec gspec=GridLayout.spec(0, 3);
    par.columnSpec = gspec;
    grid.addView(tv, par);
    int ssz=symptomNames.get(j).size();
    Toast.makeText(this, new Integer(ssz).toString(), 1);//.show();
    cboxes=new ArrayList<CheckBox>();
    for (int k=0;k < ssz;++k)
    {
        cboxes.add(new CheckBox(this));
        CheckBox cb=cboxes.get(k);
        cb.setText(symptomNames.get(j).get(k));
        cb.setChecked(false);
        grid.addView(cb);
    }
}
```

중략 }

```
button.setOnClickListener(this);
nextButtons.add(button);
grid.addView(button);
scr.addView(grid);
frame.addView(scr);
frame.setVisibility(frame.GONE);
mainLayout.addView(frame);
symptomTabFrames.add(frame);
```

320~ 330 가중치 리소스를 C++에서 읽을 수 있도록 export

```
inStream = getResources().openRawResource(R.raw.weights);
outStream = new FileOutputStream(file);
byte[] buffer = new byte[ 1024];
int length; //copy the file content in bytes
while ((length = inStream.read(buffer)) > 0)
{
    outStream.write(buffer, 0, length);
}
inStream.close();
outStream.close();
Toast.makeText(this, "DB copy success", 1).show();
```

346~388행 Invoke 함수 내부. C++부분을 호출한다.
//자료를 가공하여 Native로 보낸다.

```
        byte [] syms= toPrimitives(symptombytes);
        symptombytes.clear();
        initData(male, preg, age, 50, syms, 31);
        LoadWeights();
        calcData();
        중략
        disid1 = getDisID(0);
        중략
        prob2 = getProb(1);
        mediid1 = getDrugID(disid1);
        중략
        String medi1=mediIDsToString(mediid1);
        중략
        String DiseaseThree=diseaseNames.get(disid3);
        Intent intent=new Intent(this, ShowActivity.class);
        intent.putExtra("com.kyunggi.medinology.diseaseone.MESSAGE",
new Integer(prob1).toString() + "%확률로 " + DiseaseOne + "이며 치료제는 " + medi1 + "입니
다.");
        중략
        intent.putExtra("com.kyunggi.medinology.diseaseone.DRUGS",
mediid1);
        중략
        intent.putExtra("com.kyunggi.medinology.basic.preg",preg);
        startActivity(intent);
        예외 처리후략
```

410~438 질병에 따른 약 id들 얻어오기

private int[] getDrugID(int n)

```
{
    int siz=disdrugTable.get(n).size();// One hot encoded
    ArrayList<Integer> resa=new ArrayList<Integer>();
    for (int i=0;i < siz;++i)
    {
        if (disdrugTable.get(n).get(i) == 1)
        {
            resa.add(new Integer(i));
        }
    }
    resa Aligning 생략
    return res;
}
```

private String mediIDsToString(int[] ids)

```

{
    int len=ids.length;
    String ret=new String();
    for (int i=0;i < len;++i)
    {
        ret += mediNames.get(ids[i]) + " ";
    }
    return ret;
}

```

515~590행 질병-약 테이블 로드

ArrayList<ArrayList<Integer>> disdrugTable;

```

void ReadDisease_Drug(int dis, int drugs)
{
    disdrugTable = new ArrayList<ArrayList<Integer>>();
    try
    {
        BufferedReader br = new BufferedReader(new
InputStreamReader(getResources().openRawResource(R.raw.disdru)));
        String line = "";
        int row =0 ,i=0;

        while ((line = br.readLine()) != null)
        {
            String tok;
            disdrugTable.add(new ArrayList<Integer>());
            StringTokenizer tokenizer=new
StringTokenizer(line, ",");
            while (tokenizer.hasMoreTokens())
            {
                tok = tokenizer.nextToken(",");
                disdrugTable.get(row).add(tok.charAt(0)
== '1' ?1: 0);
            }
            row++;
        }
    }
}

```

```

        }
        br.close();
    }
    예외 처리, ReadDrug_detail 생략
592~621행 임신여부 체크 모순방지처리 생략
622~690행 증상 받기 버튼 눌렀을때처리
//버튼이 클릭되었을 때 호출됨
@Override
public void onClick(View p)
{
    //Toast.makeText(this, "clk", 1).show();
    Button p1=(Button)p;
    if (nextButtons == null)return;
    int sz=nextButtons.size();// 2
    try
    {
        for (int i=0;i < sz;++i)// 0,1
        {
            Button button=nextButtons.get(i);
            if (button == p1)
            {
                if (i == 1)
                {
                    ScrollView sc= (ScrollView)
symptomTabFrames.get(1).getChildAt(0);
                    GridLayout gv=(GridLayout)
sc.getChildAt(0);
                    for (int j = 0; j < gv.getChildCount();
j++)
                    {
                        View v = gv.getChildAt(j);
                        if (v instanceof CheckBox)
                        {
                            symptombytes.add(new
Byte((byte)((((CheckBox)v).isChecked() ?1: 0)));
                        }
                    }
                }
            }
            else
            {
                age = NP_Age.getValue();
            }
        }
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}

```

```

        preg = CB_Preg.isChecked();
        male = RB_Male.isChecked();
        if(!male&&!RB_Female.isChecked())
        {
            Toast.makeText(this,"성별을 체크
크해 주세요.",3).show();
            return;
        }
        firstFrame.setVisibility(View.GONE);
    }
    symptomTabFrames.get(1).setVisibility(View.VISIBLE);

    if (i == sz - 1)
    {
        Invoke();
    }
    mainLayout.invalidate();
후략

```

ShowActivity 22~47행 TTS 처리 부분

```

public void onInit(int p1)
{
    Locale enUs = new Locale("korea"); //Locale("en_US");
    if (myTTS.isLanguageAvailable(enUs) ==
TextToSpeech.LANG_AVAILABLE)
    {
        myTTS.setLanguage(enUs);
    }
    else
    {
        myTTS.setLanguage(Locale.KOREA);
    }
    //myTTS.setLanguage(Locale.US); // 언어 설정 , 단말기에 언어 없는 버
전에선 안되는듯
    myTTS.setPitch((float) 0.1); // 높낮이 설정 1이 보통, 6.0미만 버전에선
높낮이도 설정이 안됨
    myTTS.setSpeechRate(1); // 빠르기 설정 1이 보통
    //myTTS.setVoice();
    myTTS.speak("결 과. 이용자들의 진단 질병과 해당 약은 다음과 같

```



```

    같습니다.", TextToSpeech.QUEUE_FLUSH, null); // tts 변환되어 나오는 음성
    myTTS.speak(disease1, TextToSpeech.QUEUE_ADD, null);
//QUEUE_FLUSH 다음에 나오는 QUEUE_ADD

```

후략

93~131행 주의 사항 만드는 부분

```

ret=MainActivity.mediNames.get(ids[i]);
    inisize=ret.length();
    ret+="은(는)";
    ArrayList<Integer> arr=drugditTable.get(ids[i]);
    if(arr.get(0)==1)
    {
        ret+=" 처방이 필요합니다.";
    }
    if((arr.get(1)==1)&&(preg))
    {
        ret+=" 임신부에게 위험합니다.";
    }
}

```

49~86행 전달받은 정보를 화면에 출력하는 부분

```

public void onCreate(Bundle savedInstanceState)
{

```

```

    ReadDisease_Drug(31,36);
    setContentView(R.layout.showresult);
    // Get the message from the intent

```

```

    Intent intent = getIntent();

```

```

    disease1 =
    intent.getStringExtra("com.kyunggi.medinology.diseaseone.MESSAGE");

```

중략

TextView

```

disThreeTextView=(TextView)findViewById(R.id.diseasethreeTextView);
    disThreeTextView.setText(disease3);

```

중략

int[]

```

mediid3=intent.getIntArrayExtra("com.kyunggi.medinology.diseasethree.
DRUGS");

```

```

    age=intent.getIntExtra("com.kyunggi.medinology.basic.age",
20);

```

```

        male=intent.getBooleanExtra("com.kyunggi.medinology.basic.
gender",false);
        preg=intent.getBooleanExtra("com.kyunggi.medinology.basic.
preg",true);
        comment1=BuildComment(mediid1);
        중략
        TextView commenttv3=(TextView)
findViewById(R.id.commentThree);
        commenttv3.setText(comment3);
        myTTS = new TextToSpeech(this, this); //곧 OnInit이 호출된다.
    }

```

아래쪽에 설명될 것이라고 한 버그는 다음과 같다.

findViewById 메소드 위치가 잘못되어 이 메소드가 아직 setContentView 메소드에 의해 생성되지 않은 뷰를 찾으려 했다.

실제 구동 사진

결론(추가 작성 필요 어떤 내용적어야할지 감이 안잡힘)

기대효과

병원 등 의료시설에 접근하기 어려운 도시 노인이나 빈민, 촌락의 주민들이 기초적 의료 서비스를 받을 수 있도록 할 수 있다. 자신의 증상에 대한 병명을 알려주고 경고함으로써 병원에 가지 않으려는 사람을 설득할 수 있다. 그리고 사람의 실수를 막아주는 컴퓨터의 막강한 장점을 적극 활용하여 의료사고를 줄이는데 기여할 수 있다. 또한 간단한 질병이나 증상은 꼭 병원을 가지 않더라도 편리하게 약을 선택 할 수 있으며 만약 이 시스템을 활용하여 자판기와 같은 것을 만든다면 약국이 문을 닫았을 때 혹은 약국이 없는 지역에서 유용하고 편리하게 사용할 수 있다.

의의와 보완점

약을 처방해주는 것은 환자의 안전과 건강이 직결된 문제인 만큼 굉장히 중요한 사안이다. 그만큼 의학과 약학에서 전문적인 지식이 필요하다. 하지만 우리는 전문가가 아니고 학습데이터를 제작함에 있어서 전문 서적을 여러 권 찾아보는 등의 노력을 하였지만 의학적으로 완전하고 신뢰할 수 있는 정보라고 할 수 없다. 그리고 빅 데이터가 아닌 짧은 기간 동안 사람이 엑셀작업으로 직접 만든 데이터로 데이터의 양이 충분히 많지 않다. 따라서 어플리케이션에서 처방해주는 약과 병은 정확하지 않을 수 있다. 하지만 기존 시장에 존재하지 않던 어플리케이션을 제작함으로써 다른 개발자들의 참여를 유도 할 수 있고 우리가 공부한 내용을 가지고 사회에 기여할 수 있는 장치를 만들어냈다는 것에 의의가 있다고 생각한다.

참고문헌

알찬 진료의를 위한 약처방 가이드 개정2판 대한내과학회 편(한국의학원)
밑바닥부터 시작하는 딥러닝 사이토 고키 저 (한빛미디어)