딥러닝을 활용한 지문자 인식 및 음성화 (TTS)

1118 이현수, 1314 이건희
지도 교사 : 송석리

(초록) 초록은 15줄을 넘지 않도록 한다. 논문은 12장 이내로 작성을 권장한다. 국문 서체는 바탕체, 영문 서체는 Times New Roman, 글자크기 10 point로 작성한다.

1. 서론

1.1. 연구 동기

예전부터 장애인들의 언어인 '수어'에 관심을 많이 가지고 있었는데, 이를 배울 기회를 가지게 되어 수어를 배운 적이 있다. 그 중에서도 지문자는 다른 수어들보다 보다 간단하였고, 한글의 모든 자음(쌍자음 포함)과 모음을 표현할 수 있기 때문에 매우 인상적이었다.

지문자란, 손으로 말을 시각적으로 나타내는 문자이다. (2. 이론적 배경 2.4. 절 참고) 수화로 표현할 수 없는 말에 사용되며, 주로 고유명사나 외래어를 표현할 때 사용된다.

대부분의 농인 (청각 장애인) 이 수어를 즐겨 쓰고, 특히 지문자를 애용한다고 한다. 그 이유는 청각 장애인들의 경우 일반 문자보다, 수어가 훨씬 익숙하게 느끼기 때문이다.

따라서, 청각 장애인이 수어 (지문자) 로 표현하는 말을 비장애인들이 쓰는 말로 번역해 농인들의 의사소통을 보다 원활하게 하고 싶었고, 마침 딥러닝을 접할 기회가 생겨 CNN이라는 이미지 인식 알고리즘을 학습하며 이가 실현 가능하다는 것을 깨닫게 되었다. 기존 선행 논문들의 경우. 한글 지문자 인식은 거의 구현되어 있지 않았기에, 이런 동기까지 더해져, 연구를 시작하게 되었다.

1.2. 연구의 필요성

이 연구는, 청각 장애인들의 의사소통을 조금 더 원활하게 도와줄 수 있다. 연구를 통해 개발한 프로그램은 농인들이 수어 (지문자) 로 자신이 하고 싶은 말을 표현하면, 이를 자동으로 인식하여서 문장으로 변환하고, 비장애인들이 사용하는 언어(음성) 으로 자동 변환해주는 과정을 모두 포함하고 있다. 결론적으로, 언어 의사소통의 장벽을 없앨 수 있으며, 장애인과 비장애인과의 거리감 역시 줄어들 수 있다.

2. 이론적 배경

2.1. 딥러닝 기본이론

가. Deeplearning

딥러닝이란 주어진 데이터들을 스스로 학습하고, 그 학습 내용을 바탕으로 컴퓨터 스스로 생각 및 사고 (판단)을 하여서 알맞은 결론을 만들어 내는 것이다. 대표적으로 딥러닝을 구현한 예로는 구글의 '알파고' 가 있으며, 바둑 기수를 지속적으로 학습하여서 자신이 이기기 위해서 바둑을 어디에 두어야 할지 판단한다.

즉, 딥러닝은 머신러닝을 실현하기 위한 하나의 방법이다. 딥러닝을 활용한 연구들로는 사진 인식, 음성 인식, 글자 인식, 자동 번역 및 단어완성 등이 있다.

나. 신경망 설계

딥러닝 알고리즘의 구조는 기본적으로 '인공 신경망 (NeuralNetwork)'로 이루어져 있다. 신경망은 입력 노드 (계층)로부터 데이터들이 들어온 후, 히든 노드 (계층)에서 학습을 진행하고 판단을 세운 후, 출력 노드 (계층)에서 최종적인 값을 출력한다.

신경망 내에서 딥러닝이 이루어지는 과정은 다음과 같다. 최종적으로 학습을 통해 신경망이 만들어낸 추축값과 정답의 일치율을 높여야 하므로, 그 차이 (오차)를 최소화 시키는 과정을 따라 딥러닝이 진행된다. 오차 최소화를 위해 사용하는 방법으로는 '최소 제곱법' 이 가장 널리 쓰이는데, 이는 주로 비교적 적은 양의, 단순한 데이터들을 이용해서 그 관계를 찾을 때 사용된다. 예시로 주어져 있는 아래

데이터에서 x와 y는 선형 관계를 가지고 있다고 가정하고, 그 관계식을 $y = ax + b$라고 해 보자.

표1. 선형 관계를 가지는 데이터 예시

x	y
1	2
2	5
3	7
4	9
5	13
6	15
7	17
8	20

먼저 아래와 같은 수식을 통해서 기울기와 절편 a, b의 값을 정할 수 있다.

$$a = \frac{\sum_{k=1}^{n}(x_k - x_{평균})(y_k - y_{평균})}{\sum_{k=1}^{n}(x_k - x_{평균})} \cdots (1)$$

$$b = y_{평균} - a \times x_{평균} \cdots (2)$$

이때 기울기 a의 값을 '가중치(W)' 라고 하고, 절편의 값 b를 '편향' 이라고 한다. 가중치와 편향이 변수별로 다르다면 가중치 행렬과 편향 행렬을 생각할 수 있는데, 이때의 선형 관계식은 아래와 같이 나타낼 수 있다.

$$y = x \cdot W + b \cdots (3)$$

즉, 행렬곱으로 표현할 수 있다.

다음으로는 오차를 구해서 그 오차를 최소화 시키는 과정에 대한 설명이다. 데이터 양이 많거나 그 관계가 복잡할 경우 (3) 만으로는 가중치와 편향을 결정할 수 없다. 따라서 오차를 계산해야 하는데, 오차로는 '평균 제곱근 오차(RMSE)' 를 가장 많이 사용한다.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - Y_i)^2} \cdots (4)$$

이때 X_i는 예측 값을, Y_i는 실제 값 (정답)을 의미한다.

오차를 줄여가는 방법으로는 '경사 하강법' 을 이용한다.

다. 경사 하강법

경사 하강법이란, 매우 조금씩 그 값 (W, b)을 바꿔 가면서 오차가 최소가 되는 최적의 점을 찾는 방법이다. 그래프에서 보자면, x좌표를 조금씩 바꿔 가면서 최솟값을 가지는 점을 찾는 과정이다. 해당 점에서 미분 계수를 구해서, 그 기울기 부호의 반대 방향으로 이동한다. 예를 들어, 해당 점에서의 미분계수가 양수라면, 음의 방향으로 ($-x$ 방향) 조금 더 이동하고, 이동한 점에서 다시 미분을 해서 이 과정을 반복하는 것이다.

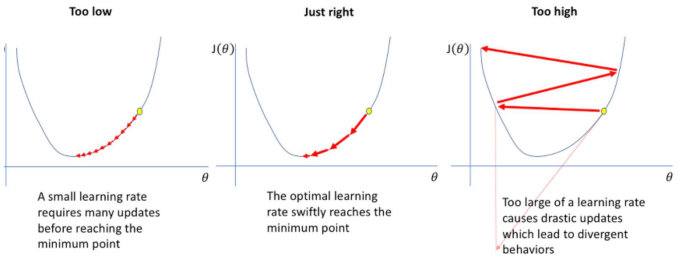

그림. 1. 경사 하강법의 원리
출처 머신러닝단기집중과정

이때 그래프 상에서 '얼마만큼' 움직일지를 정하는 변수가 바로 '학습률(learning rate)' 이다. 학습률이 크다는 것은 이동 폭이 크다는 것이고, 학습률이 작다는 것은 그 반대이다. 적절한 값의 학습률을 설정해야 오차를 최소화 할 수 있으며, 너무 크거나 작은 경우 최솟값에 도달하지 못하거나 시간이 너무 오래 걸릴 수 있다.

2.2. CNN

CNN이란, Convolution Neural Network의 약자로, 이미지 인식 등을 보다 효율적으로 도와주는 딥러닝의 한 방법 중 하나이다. 기존 신경망 알고리즘을 이용하면 사진 데이터 (3차원)을 1차원 평면배열로 변환시키는 과정에서 데이터 손실이 일어나게 되는데, 이를 보완해서 그 정확도를 높이는 것이 바로 CNN의 핵심 원리이다. CNN 알고리즘의 크게 컨볼루션 계층, 풀링 계층, 그리고 판단 및 출력계층으로 구성된다.

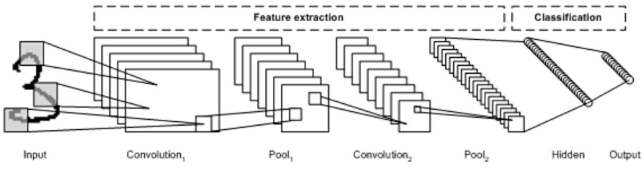

그림. 2. CNN의 원리
출처 https://goo.gl/fkphjL

먼저 하나의 이미지가 입력되면, '컨볼루션 계층' 에서 이 이미지의 특징적 정보들을 추출해 내어서 압축시킨다. 이후 한번 더 압축 과정을 거치는데, 이것이 바로 '풀링' 이다. 주로 '맥스 풀링' 기법을 이용해서 풀링을 진행하며, 이 연구 역시 '맥스 풀링' 기법을 사용하였다. 이 과정을 설정한 노드 개수만큼 반복하여서 최종적으로 판단 (결론) 을 내린다.

2.3. Dropout

적은 데이터로 학습을 시도하면 그 정확도가 매우 낮다. 데이터 양이 부족하기 때문에 신경망이 충분한 학습을 진행하지 못하였기 때문이다.

하지만, 그렇다고 해서 너무 많은 데이터가 있어도 '과적합 (Overshooting)' 이 발생하여서 정확도가 낮아진다. 학습 횟수 역시 마찬가지다. 너무 적은 학습은 '학습량 부족' 이라는 결론을 야기하지만, 너무 많은 학습 역시 과적합을 발생시킨다. 즉, 적절한 양의 데이터와 학습 횟수를 설정해야 최적의 정답률을 얻을 수 있다.

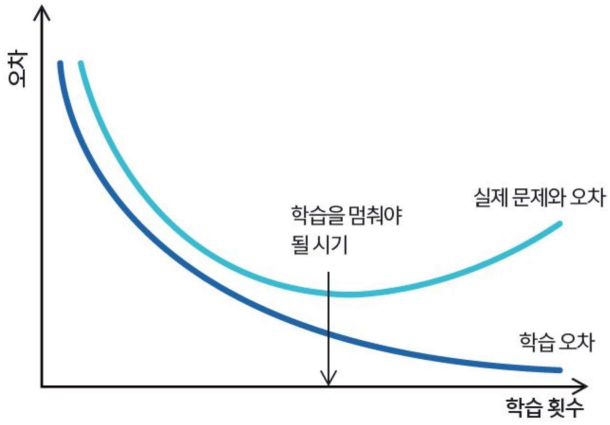

그림. 3. 학습 횟수가 너무 많아지면 오히려 오차가 증가하는 과적합 현상
출처 http://dongascience.donga.com/news.php?idx=11225

하지만 이 '적절한' 횟수를 알아내는 것은 힘들다. 따라서 '드롭아웃' 기법이 사용되는데, 이 드롭아웃 기법은 통해서 데이터나 학습 횟수가 과적합을 불러올 정도로 많아도 자동적으로 줄여 준다. 이 연구에서도 드롭아웃 기법을 이용해서 과적합을 방지하였다.

2.4. 지문자

지문자는 수화보다 비교적 간단하고 단순한, 손을 이용한 문자로, 한글 자음과 모음을 표현할 수 있다. 아래는 지문자로 한글을 표현하는 방법이다.

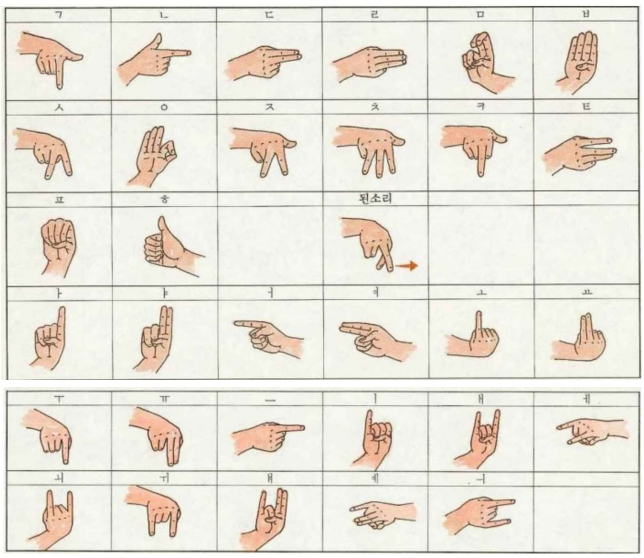

그림. 4. 한글 지문자 자음, 모음
출처 http://www.urimal.org/1222

이 연구에서는 딥러닝(CNN)을 통해 위 지문자들을 인식하는 알고리즘을 개발하였다.

3. 연구의 목적

- 기존 논문과의 차별성 역시 언급할 것

4. 연구 방법

4.1. 개발 환경

이 연구에서 최종적으로 목표하는 것은 사진 인식과 TTS 기능이다. CNN알고리즘이 가장 적합하다 판단하였기에, Python을 이용하여서 개발하기로 하였다. 아래는 정확한 개발 환경이다.

Windows-64bit Anaconda, Tensorflow (텐서플로) 기반 Python 3.7 Pycharm, Jupyter Notebook 동시 사용 [GitHub에 코드 수시 업로드]

아래는 본 연구 시 설치하거나 불러온 library 목록이다.

```
import tensorflow
import os
import cv2
import numpy
import re
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import OneHotEncoder
from numpy import array
from gtts import gTTS
```

4.2. 연구 절차

우리는 최종적으로 원하는 결과물을 얻기 위해 다음 절차에 따라서 연구(개발)을 진행하였고 검토 역시 병행하였다.

① 데이터 생성하기

: 한글 지문자 데이터는 빅데이터를 얻을 수 있는 사이트 (Kaggle) 에서도 찾을 수 없었다. 따라서 직접 찍기로 하였으며, 검정색 바탕을 배경으로 모든 지문자에 대해서 촬영 하였다.

또한, 사람마다 손 모양이나 색깔 등이 약간씩 다르므로 연구자 2명 외에도 주변인의 손 사진 역시 찍어서 학습 데이터에 추가함으로써 그 정확도를 높였다. 데이터 생성은 수시로 이루어졌다.

② 이미지 읽어오기

: 기존 CNN 알고리즘에서 손글씨를 인식할 때는, Tensorflow 에서 기본적으로 MNIST 데이터를 제공하였기 때문에 이미지를 변환할 필요가 없었다. 하지만 현재 인위적으로 데이터를 생성하였기 때문에, 아래와 같은 사진 가공 과정이 필요하였다.

㉠ 촬영한 사진을 28pixel × 28pixel 로 변환한다.(Resize) 이때는 사진 편집 프로그램 (PhotoScape) 을 이용하여서 일괄적으로 변환하였다.

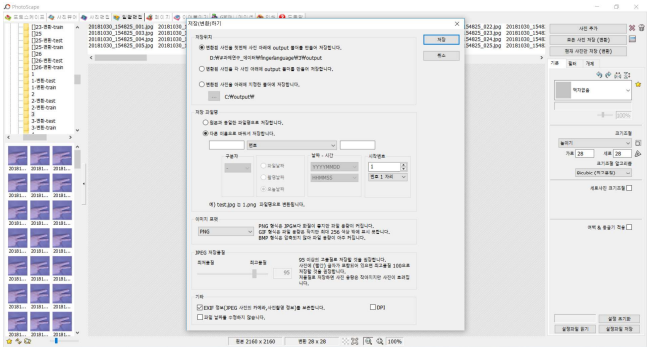

그림. 5. 이미지를 28 × 28 사이즈로 변환하는 과정

㉡ Resize 시킨 이미지를 숫자 배열로 변환하고, 이를 CNN 인식 데이터에 첨부시킨다.

㉢ 첨부된 사진들을 정답과 함께 배열에 최종적으로 저장하고, 인식을 시작한다.