

이학학사학위 논문

딥러닝을 활용한 분리수거의 디지털 트랜스포메이션 연구

Digital Transformation of Separate Collection Using Deep Learning

2020. 12.

심호길

인하대학교 물리학과



이학학사학위 논문

**딥러닝을 활용한 분리수거의 디지털 트랜스포메이션
연구**

**Digital Transformation of Separate Collection Using
Deep Learning**

2020. 12.

심호길

인하대학교 물리학과

지도교수 : 이재우 (인)

논문을 이학학사 학위 논문으로 제출함.

이 논문을 심 호 길의 이학학사 학위논문으로 인정함.

년 월 일

논문사정위원장	(인)
---------	-----

논문사정위원	(인)
--------	-----

논문사정위원	(인)
--------	-----

논문사정위원	(인)
--------	-----

초 록

전 세계적으로 문제가 되고 있는 폐기물 처리에 대한 문제 중, 분리수거에 대한 재사용율을 높일 수 있는 방안으로 AI를 제시하고 실제로 이 것이 산업에 적용될 수 있는지 확인해보는 연구를 하였다.

AI가 분리수거에 적용되기 위해 이미지를 분류할 수 있는 능력을 가지고 있는 신경망을 설계하고자 했다. 이미지 분류에서 많이 사용되고 있는 컨볼루션 신경망을 딥러닝을 통해 구현하였고, 높은 정확도를 가지고 있는 사전 학습된 모델을 사용하여 정확도를 높이하고자 하였다. 개발환경은 구글의 코랩을 사용하여 디바이스의 의존성을 줄이고자 하였다. 6가지의 분리수거 항목으로 데이터를 분류하여 신경망을 학습하였다.

결과적으로 정확도 90%이상을 가지고 있는 신경망을 구현하였고, 이 이미지 분류 신경망을 더욱 발전시킨다면 산업과 일상에서 디바이스와 결합되어 분리수거에서 좋은 결과를 이룰 수 있다고 생각된다.

목 차

제 1 장 서론

제 2 장 배경 이론

1. 디지털 트랜스포메이션과 딥러닝
2. 퍼셉트론
3. 인공신경망과 딥러닝
4. 신경망의 학습원리
5. 컨볼루션 신경망

제 3 장 연구 방법

1. 연구 환경
2. 신경망 모델

제 4 장 결과 및 분석

제 5 장 결론

참고문헌

제 1 장 서론

4차 산업의 시대가 도래하면서 가장 화두로 떠오른 것은 AI분야라고 할 수 있다. 인간이 풀 수 없는 문제를 풀어주거나 인력이 부족한 부분을 채워주는 등 현재 여러 산업 분야에서 급속도로 스며들고 있다. 이러한 AI가 적용되어야 할 분야 중 하나를 보자면 폐기물 처리를 이야기할 수 있다. 우리나라의 연간(2017년 기준) 폐기물 처리비용은 23조원에 육박하며, 그 발생량은 일일 43만톤 그 중에서도 플라스틱 폐기물 발생량은 8162톤이며 재사용율은 62%로 일본의 재사용율 83%에 21%p 차이가 난다.

AI를 통해 재사용율을 높이고 처리비용을 줄이기위한 방법 중 하나는 딥러닝을 통해 이미지 분별 신경망을 만드는 것이다. 딥러닝은 간단한 사전지식과 적절한 데이터만으로도 쉽게 또한 저렴한 비용으로 시작할 수 있는 첨단분야이다. 전문적인 지식이 없더라도 이미지를 분별할 수 있으며 환자의 데이터를 보고 어떤 병이 있는지 예측할 수 있다.

하지만 그러함에도 접근성에 대한 문제점은 존재한다. 그 문제점은 바로 디바이스의 높은 사양을 요구한다는 것이다. 개발환경의 경우는 클라우드 서비스를 통해 쉽게 구성할 수 있지만 실제로 딥러닝을 통해 만들어진 모델을 적용한 어플리케이션을 구동할 환경을 개인이 가지고 있기는 어려운 상황이다.

이러한 문제점을 해결하기 위한 방법으로 서비스를 클라우드 컴퓨팅을 통해 제공하여 이용자의 디바이스 의존성을 줄여주는 디지털 트랜스포메이션(DX) 기술이 각광을 받고 있다. 실제 기업들은 자신들의 서비스들을 모두 DX로 변환을 하기 시작하였으며, 소프트웨어의 한계를 벗어나기 위해 웹을 통해 제공하기도 한다.

이번 연구는 딥러닝을 통해 폐기물 처리에 한 부분인 분리수거를 기계가 인간을 대체할 수 있는지에 대한 가능성을 확인하고, 실제로 적용해볼 수 있는지에 대해 알아보려고 한다.

제 2 장 배경 이론

2.1 디지털 트랜스포메이션과 딥러닝

디지털 트랜스포메이션(Digital Transformation, DX 또는 DT)이란 클라우드 컴퓨팅을 활용하여 서비스 사용자의 디바이스 사양에 대한 의존성 줄이고 서비스 자체에 대한 의존성을 높이는 기술방식을 말한다. 첨단 산업이 점점 발전됨에 따라 제공하는 서비스의 하드웨어 요구사양이 점점 높아지면서 서비스를 제공받는다 하더라도 이용자의 디바이스 사양이 낮아 의도한 기능들을 제대로 사용할 수 없는 상황이 많아지고 있다. 디지털 트랜스포메이션은 서비스 제공자가 고성능을 필요로 하는 기능 혹은 서비스의 전체적인 부분까지도 클라우드 컴퓨팅을 이용하여 앞서 말한 문제점들을 해결해준다.

딥러닝을 기반으로 한 인공지능 서비스를 제공하기 위해서는 이용자 또한 높은 사양의 디바이스가 없을 경우 이용자가 원하는 서비스의 기능을 이용하지 못할 경우가 있다. 이용자의 디바이스로 학습을 진행하지 않는다고 해도 입력데이터를 계산하고 결과를 도출하기는 저사양의 디바이스로 어려운 상황이다. 즉 서비스를 제공하기 위해서는 디지털 트랜스포메이션은 필수적인 요소이다.

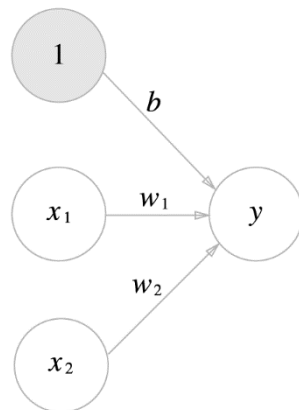
2.2 퍼셉트론(Perceptron)

퍼셉트론이란 1957년 프랑크 로젠블라트가 고안한 알고리즘으로 인공 신경망과 딥러닝의 근간이 된다. 퍼셉트론은 인간의 뉴런을 모방하여 수학적으로 표현한 계산 알고리즘이며, 그 동작은 뉴런과 같이 다수의 신호를 입력으로 받아 하나의

신호를 출력한다. x_1 과 x_2 가 각각 입력신호, y 를 출력신호, w_1 과 w_2 를 가중치, b 를 편향이라 하면 단층 퍼셉트론의 수식적인 표현은 다음과 같다.

$$y = \begin{cases} 0, & w_1x_1 + w_2x_2 + b \leq 0 \\ 1, & w_1x_1 + w_2x_2 + b > 0 \end{cases}$$

<그림 1>에서 전체그림을 뉴런, 각 입력신호와 출력신호를 가지고 있는 원을 노드라 하자. 입력 노드에서 나오는 신호들(x_1, x_2)이 가중치(w_1, w_2)에 의해 출력 노드에 대한 영향력이 정해지고, 각 신호와 가중치 그리고 편향(b) 세개의 신호가 뉴런의 끝 출력 노드(y)에 전달되어 그 크기에 따라 활성화될지 안될지가 결정이 된다. 여기서 편향은 뉴런이 얼마나 쉽게 활성화하는지에 대한 척도로 볼 수 있다.



<그림 1> 도식화된 퍼셉트론

더 나아가 입력신호와 가중치의 곱 그리고 편향의 합을 a , 출력신호 y 를 $h(a)$ 라 하면 수식적인 표현은 다음과 같다.

$$a = b + w_1x_1 + w_2x_2$$

$$y = h(a)$$

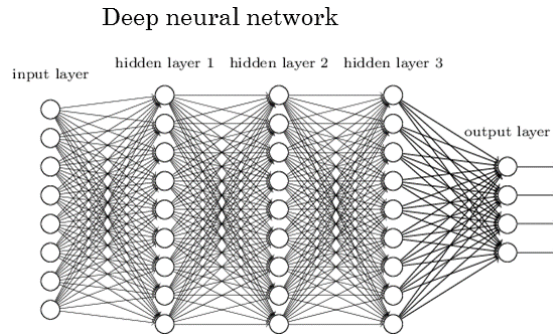
여기서 $h(a)$ 를 활성화 함수라 한다. 활성화 함수는 앞서 보인 것처럼 임계 값을 기점으로 0과 1을 출력하는 계단 함수 뿐 만 아니라 시고모이드 함수, ReLU 함수, 소프트맥스 함수 등 상황에 따라 다양한 함수들이 쓰인다.

기본적인 퍼셉트론은 <그림 1>과 같이 단층의 신경망을 말한다. 이 퍼셉트론을 통해 인공신경망의 기본적인 동작원리를 살펴볼 수 있다.

2.3 인공신경망과 딥러닝

인공신경망(Artificial Neural Network)은 인간의 신경망에서 아이디어를 얻어 그 작동을 모방하는 알고리즘이다. 인간의 신경망의 뉴런들은 망 내에 반복적인 신호가 발생할 때 그 신호를 학습하여 기억하는 행동을 한다. 이러한 뉴런처럼 발생한 신호를 학습하기위해 인공신경망은 각 인공 뉴런에 가중치라는 것을 적용시켜 신호에 대해 같이 행동을 하는 뉴런들을 더 단단히 연결시키는 학습 개념을 사용한다. 다양한 입력 신호 및 데이터에 의해 학습이 된 인공신경망은 그 학습을 기반으로 새로운 입력에 대해 결과물을 산출한다.

딥러닝은 이러한 인공신경망 알고리즘을 이용하여 기계학습을 시키는 한 분야이다. 딥러닝의 딥(Deep)이 의미하는 것은 인공신경망(이하 신경망)의 층이 깊고, 그에 따라 고려해야할 가중치 및 변수 많다는 것이다. 층에 대한 기준을 보자면 보통 3개 이상의 층을 가지는 신경망을 심층망(Deep Network)이라고 한다. 이러한 기준으로 딥러닝은 3개이상의 심층을 가지고 있는 신경망 즉 심층신경망으로 정의되기도 한다.



<그림 2> 심층 신경망

기본적으로 딥러닝 이전의 신경망은 2~3개의 층을 두고 층마다의 노드들의 가중치들을 조절하여 학습하는 방식을 사용하였다. 딥러닝은 여기서 더 나아가 층수를 많이 늘려 학습하는 방식을 사용한다. 하지만 이것은 곧 복잡한 연산과 높은 수준의 하드웨어를 필요로 한다. 이러함에도 불구하고 계층을 계속 늘리는 이유는 이전의 신경망의 경우 복잡한 입출력 관계를 가지는 문제에 대해 노드를 늘리는 방법을 사용하여 복잡도를 높였지만 효과적이지 못했고, 층을 올리는 딥러닝의 경우 이전의 신경망보다 획기적으로 경우의 수가 많아져 복잡도를 효과적으로 높일 수 있다. 또한 인간수준의 지능을 인공지능망을 통해 얻기 위해서는 인간이 가지고 있는 신경망만큼 혹은 더욱 많은 층을 쌓는 설계가 필수적이라고 볼 수 있다.

2.4 신경망의 학습원리

신경망의 학습원리는 기본적으로 입력층과 출력층 사이에 임의의 노드들과 가중치들을 설정 후 데이터를 입력하여 출력 값을 데이터의 출력과 비교 후 피드백하여 가중치들을 조정하거나 노드들을 수정한다.

인공신경망은 일반적으로 입력층과 출력층 사이에 여러 개의 은닉층을 갖는 다층 신경망을 의미한다. 이 다층 신경망의 은닉층은 각 층마다 앞 층에서 전달받

은 신호를 각각의 노드마다의 가중치를 확인 후 합산하여 활성화 함수에 적용시켜 다음 층에 전달한다. 데이터의 수를 m , 입력 값을 x , 일련의 과정들을 거친 후 출력층에서 나온 예측값을 $H(x)$ (Hypothesis), 실제 데이터의 결과값을 y 라 하면, 손실함수(Cost 혹은 Loss)는

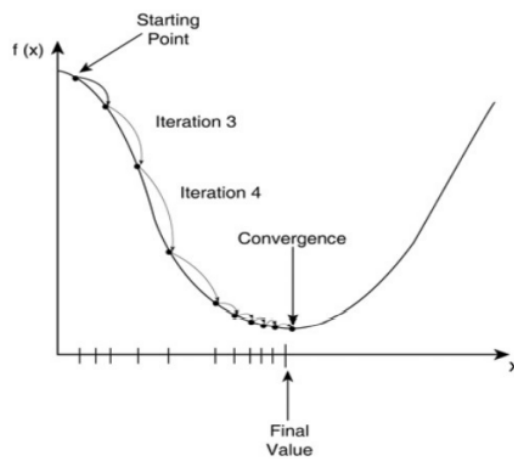
$$cost = \frac{1}{m} \sum_{i=1}^m (H(x_{(i)}) - y_{(i)})^2$$

이다. 현재 cost에 쓰인 함수는 평균 제곱 오차가 쓰였지만 Cross-entropy 등 상황에 따라 다른 손실함수를 사용할 수 있다.

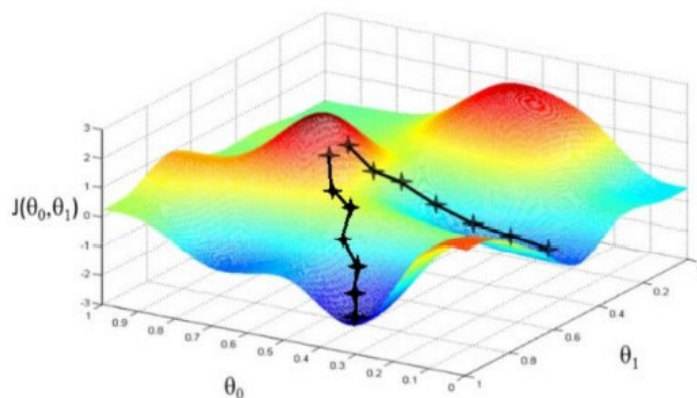
Cost 함수는 예측값과 결과값의 차이로 표현되고 있고 결과값은 데이터에 의해 정해진 값이며 결국 cost 함수는 예측값에 대한 함수가 된다. 예측값 $H(x)$ 를 간단한 수식으로 표현하면 2.2의 퍼셉트론과 같이 $H(x) = wx + b$ 로 표현할 수 있으며 여기서 x 는 데이터에 의해 정해진 값, w (가중치), b (편향)는 변수이다. 즉 cost 함수는 w 와 b 의 함수 $cost(w, b)$ 로 표현이 가능하다.

신경망 학습은 cost 함수를 최솟값으로 만드는 데에 목표를 두고 있다. Cost 함수를 최소화하는 방법으로는 경사하강법(Gradient Descent)을 기반으로 한 오차 역전파 알고리즘을 사용한다. 먼저 경사하강법은 cost 함수를 미분 후 그 값

이 0에 가까워지는 즉 기울기가 낮아지는 방향으로 가중치를 단계적으로 조정하여 cost를 최소로 만드는 방법이다.



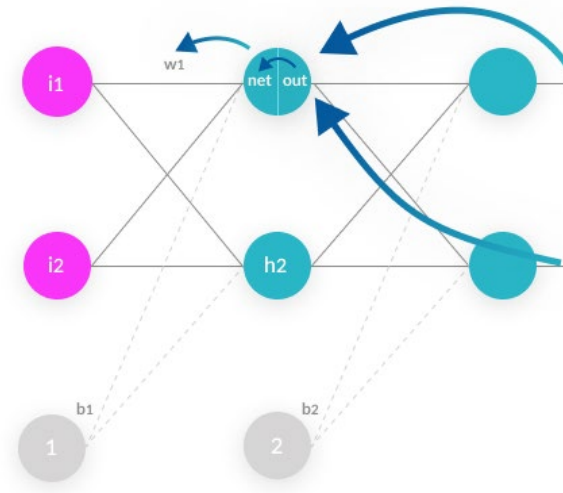
<그림 3> 2차원에서의 경사하강법



<그림 4> 3차원에서 경사하강법

<그림 3>와 <그림4>에서 2차원과 3차원에서의 경사하강법을 그래프를 통해 보여주고 있다. <그림 3>에서도 마찬가지지만 <그림 4>에서 보면 경사도가 0인 지점은 안장점이나 최대값을 갖는 점을 나타낼 수도 있고, 시작점에 따라 최소값을 갖는 점이 아닌 극솟값을 갖는 점일 수도 있다. 결국은 cost 함수를 알맞게 설정하고, 시작점과 기울기 정보를 통해 어느정도 실험적인 방법이 필요할 수 있다.

경사하강법을 기반한 오차 역전파는 다층 신경망에서의 경사하강법을 사용하는 방법을 말한다. 입력층과 출력층만 보는 것이 아닌 출력층에서부터 바로 전 층 또 그 층에서 전 층과 같이 단계별로 경사하강법을 적용하여 한꺼번에 계산했을 때의 복잡성을 줄여준다.



<그림 5> 오차 역전파의 진행

즉 오차를 경사하강법을 매개로 하여 출력층에서 입력층으로 역으로 전파하여 가중치 및 편향을 조정한다.

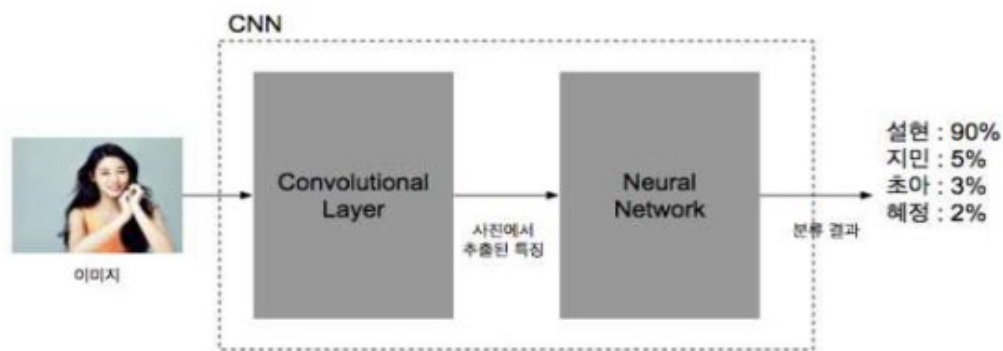
2.5 컨볼루션 신경망(합성곱 신경망, CNN, Convolution Neural Network)

컨볼루션 신경망은 이미지 인식 분야에 큰 발전을 가져온 강력한 성능을 보이는 알고리즘이다. 컨볼루션 신경망은 사람의 시각인지 과정을 모방한 방법이며 입력된 이미지의 특징을 커널(혹은 필터)를 도입하여 추출하는 기법이다.

인간이 이미지의 전체적인 데이터를 분석하기 보다 몇몇의 여러 특징들의 조합으로 사물을 구별하는 것에 착안이 되었다. 구별점이 없는 이미지의 RGB데이터만으로는 인공지능망이 적절한 가중치들을 조정하지 못하였고 수동적으로 조절한

혹은 딥러닝에 의해 스스로 학습한 컨볼루션 레이어를 통해 특징을 강조하여 추출하여 문제점을 해결하였다.

<그림 6>에서 보는 바와 같이 전체적인 컨볼루션 신경망의 작동원리는 기존에 있던 신경망에 좀더 효과적으로 이미지의 특징을 강조하여 가중치조정을 하기 위해 특징 추출기인 컨볼루션 레이어가 추가된다. 컨볼루션 레이어는 이미지 뿐 아니라 좀더 다차원의 데이터 혹은 단순한 1차원의 데이터까지 사용이 가능하다.



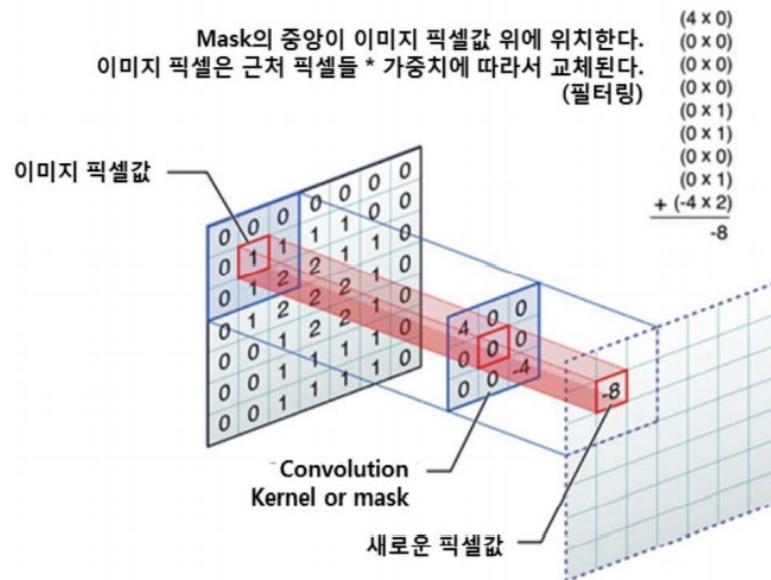
<그림 6> 컨볼루션 신경망의 작동원리

컨볼루션 신경망에서 컨볼루션이란 합성곱을 의미하며 본래 수학에서 사용되는 용어로, 신호에서 원하는 특징을 얻어내기 위한 신호처리 연산을 의미한다. 즉 데이터의 추출해내고 싶은 특징에 맞춰 커널을 설계하고 데이터와 커널의 합성곱을 통해 그 특징을 추출하여 그것을 기반으로 출력할 값을 결정한다.

여기서 특징들이 추출의 예는 이미지에서 수평, 수직 성분만을 추출 혹은 곡선만을 추출, 그리고 이미지 편집기에서 볼 수 있는 날카롭게(sharpen), 흐림(blur) 등을 들 수 있다.

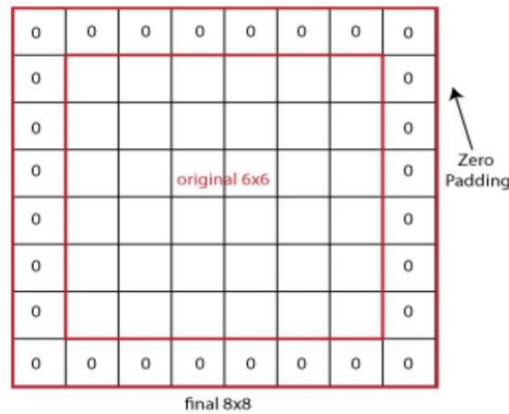
커널은 입력 이미지 데이터에서 왼쪽 상단부터 정해진 칸 수만큼 오른쪽으로 이동하면서 합성곱을 계산하게 된다. 이때 커널의 이동 간격을 보폭(stride)라 하며, 커널이 적용되어 새로 생성된 이미지 데이터를 특성지도(feature map)이라

한다. 커널을 통해 만들어진 특성지도는 활성화 함수가 적용되어 다음 커널 혹은 다음 층으로 넘어가게 된다.



<그림 7> 컨볼루션 신경망에서의 합성곱

합성곱이 적용되면 2D의 경우 상대적으로 가운데 부분은 중첩이 되어 정보의 유실이 적지만 끝부분의 경우 중첩이 적어 유실될 가능성이 높아진다. 이러한 정보 유실을 방지하기 위해 패딩(padding) 기법을 사용되게 된다. 패딩은 <그림 8> 처럼 끝부분 주위에 추가적으로 0값을 넣어 인위적으로 이미지의 전체적인 크기를 늘린다.



<그림 8> 6X6 이미지에 폭 1의 패딩 적용

컨볼루션 레이어를 거친 데이터는 필터링이 되어 특징들이 추출된 상태로 나온다 하더라도 데이터의 양은 변함이 없거나 레이어에 따라 크게 부풀려질 수 있다. 여기서 이미지의 경우 구성하는 픽셀들은 인접한 픽셀들끼리 비슷한 데이터를 갖고 있는 경우가 많다. 이러한 경우 연산을 줄이기 위해 풀링 레이어 (Pooling Layer)를 사용하는 경우가 있다. 설정한 범위내에서 최대값 혹은 평균값을 대푯값으로 설정하여 전체적으로 데이터의 양을 줄인다. 꼭 필요한 레이어는 아니지만 학습 데이터에 의존성이 높아 생기는 과적합을 줄여주는 효과를 얻을 수 있다.

복잡하고 깊은 신경망은 과적합을 유발하기 쉽고 그러한 것을 해결하기 위한 방법이 드랍아웃(Dropout)이다. 드랍아웃은 무작위로 뉴런의 부분집합을 제거하는 방법이다. 이 방법은 결과값이 노드들의 한방향으로 치우치게 되는 현상을 인위적으로 노드들을 결손 시켜 과적합을 막는 방법이다.

최종적으로 풀링 레이어까지 적용된 데이터는 다음 인공신경망에 입력되어 출력층에 도달하고 오차역전파를 통해 학습이 진행된다.

제 3 장 연구 방법

컨볼루션 신경망을 설계하고 분리수거 이미지 데이터를 통해 신경망을 학습시키는 연구를 실시하였다.

3.1 연구 환경

딥러닝을 학습하기 위한 환경은 높은 GPU 가속 사양을 요구로 하기 때문에 로컬환경에서는 환경을 구성하기 어려웠다. GPU 가속을 지원하는 클라우드 서비스 중 무료로 제공하는 구글의 코랩(Colaboratory)을 사용하였고 딥러닝에 필요한 패키지들은 기본적으로 내장 되어있다. 딥러닝에 사용한 프레임워크로는 텐서플로(TensorFlow)를 사용하였고 버전은 2.3 버전, 내장된 파이썬(Python)의 버전은 3.6.9버전이다. 또한 주피터 노트북을 통해 코드를 작성하였다. 사용된 데이터셋은 데이터과학자 커뮤니티인 캐글(Kaggle)에서 공개 되어있는 “Garbage Classification” 을 사용하였다.

3.2 신경망 모델

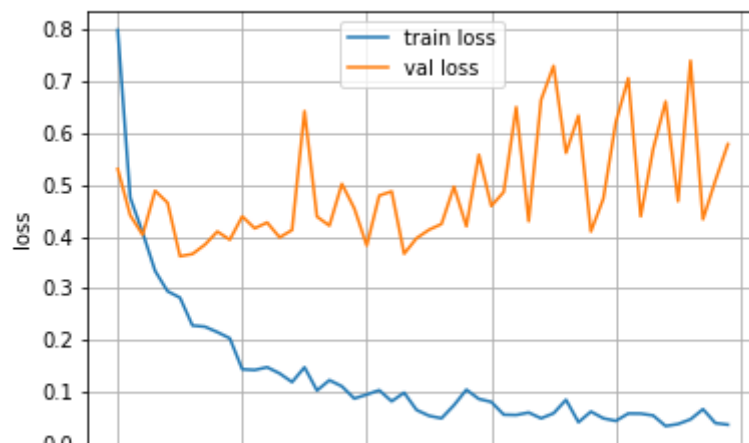
신경망 모델의 학습효율을 높이고 학습하는데 걸리는 시간을 줄이기 위해 사전 훈련된 모델(pre-trained model)로 Inception V3를 사용하고 이것을 특징 추출기로 사용하였다. 특징 추출기는 학습 시 자체의 가중치를 조정하지 않고 입력 데이터의 특징만을 뽑는 역할을 한다. 실제로 특징을 받아 학습을 할 신경망 층은 순서대로 GlobalAveragePooling2D, Dropout, ReLU를 활성화 함수를 사용한 Dense,

소프트맥스를 활성화 함수로 사용한 Dense로 구성되어 있다. 훈련 데이터를 순회할 에포크(epoch)는 50회로 설정하였고 실행단에 데이터를 한 번에 넘겨주는 단위인 배치 사이즈는 16으로 하였다. 타겟 사이즈는 300 X 300으로 하였다. 학습의 상용된 데이터 카테고리는 cardboard, glass, metal, paper, plastic, trash 총 6가지이며 학습에 사용된 데이터는 이미지 2199장, 확인작업에 사용된 검증 데이터는 328장이 있다.

신경망의 학습은 앞에서 설계한 모델에 카테고리별로 라벨링이 된 데이터를 입력하여 진행하였고, 학습 데이터를 한번 순회하면 학습에 참여하지 않는 검증 데이터를 순회하며 학습된 모델의 성능을 확인하였다.

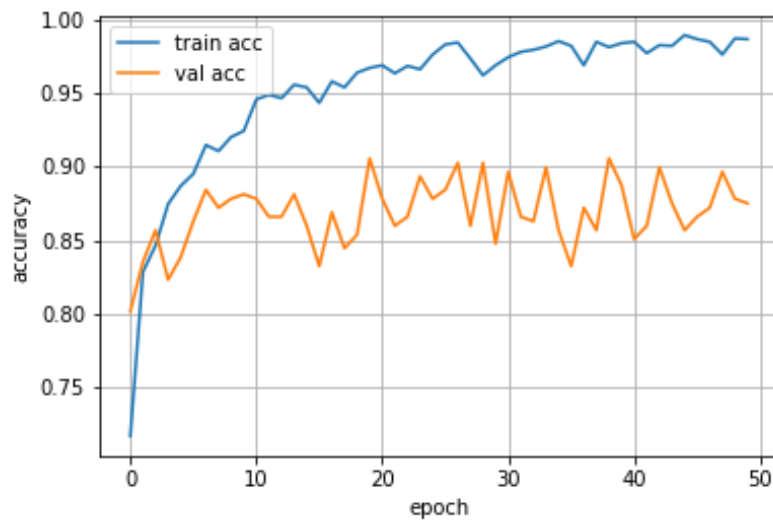
제 4 장 결과 및 분석

실험의 결과로는 학습오차(train loss 혹은 cost)가 시작값은 0.7990, 최종값은 0.0371, 최대값 0.7990 최소값 0.0344이고 확인작업에서의 검증오차(valid loss)는 시작값이 0.5306, 최종값은 0.5789, 최대값은 0.7398, 최소값은 0.3623 또한 학습 정확도(train accuracy)는 시작값이 0.7171, 최종값이 0.9864, 최대값이



<그림 9> 에포크 당 오차

0.9891, 최소값이 0.7171 이고 검증 정확도(valid accuracy)는 시작값이 0.8018, 최종값이 0.8750, 최대값이 0.9055, 최소값이 0.8018이다.



<그림 10> 에포크 당 정확도 그래프

전체적으로 보았을 때 실제로 분리수거를 위해 사용될 모델은 검증 정확도가 가장 높고 오차가 가장 적은 모델이다. 검증 정확도의 최대값 90.55%를 갖는 모델은 검증 오차를 0.41, 학습 정확도가 98.09%, 학습 오차가 0.0623이다.

설계한 모델은 학습 정확도의 최대값은 98.91% 검증 정확도의 최대값이 90.55%로 나왔다. 학습 정확도가 높은 것은 좋지만 최종적으로 사용해야할 지표는 검증 정확도이다. 검증 정확도는 학습에 사용되지 않은 새로운 데이터에 대한 정확도를 의미하기 때문이다. 검증 정확도가 90%로 낮은 수치를 갖는 것은 아니지만, 학습 정확도와 8%정도 차이가 발생하고 있다. 이러한 차이는 과적합 문제를 생각할 수 있다. 학습이 진행될수록 변수들이 학습데이터에 과도하게 최적화되어 새로운 데이터에 대한 정확도가 차이나는 것으로 보인다.

또한 데이터의 다양성을 생각해볼 수 있다. 다양성이 부족한 데이터는 학습의 효율을 떨어뜨리고 과적합을 유발한다. 데이터를 단순히 늘리는 방법을 사용할 수 있겠지만 그러한 방법 보다는 직접 선별하여 다양성을 늘리는 것이 좋은 방법이다.

정확도가 아무리 높다고 하더라도 분별하지 못하는 데이터는 존재한다. 인간과 같이 인공 신경망 또한 확실하지 않은 카테고리의 교집합에 있는 이미지는 분별하기 어렵다. 사용한 데이터를 보았을 때 학습 데이터와 검증 데이터에서 이러한 데이터가 다소 있는 것을 확인하였고 이 것이 좀 더 높은 정확도를 얻지 못한 원인 중 하나로 생각된다.

딥러닝의 의미는 층 수가 많아 학습할 수 있는 경우의 수가 많아지는 것에 있다. 특징 추출기는 학습을 하지 못하게 설정되어 설계를 하였고, 실제 학습한 층은 4개의 층으로 되어있다. 특징 추출기를 학습을 할 수 있게 설계를 하거나 학습하는 층 수를 늘려 정확도를 높이는 방법을 생각해볼 수 있다.

사용된 코드와 결과는

<https://github.com/simkoon/DeepLerningForGarbageClassification/blob/main/GarbageClassification.ipynb>

에서 확인할 수 있다.

제 5 장 결론

폐기물 처리에서 분리수거는 발생량을 줄이고 재사용율을 늘릴 수 있는 가장 간단한 방법 중에 하나이다. 하지만 작업 기피 현상과 부족한 인력, 사회적 인식으로 인해 재사용율은 매년 제자리 걸음 중이다. 이러한 문제를 개선하기 위해서는 AI의 도입이 필요하다고 생각되고 그 중 가장 중요한 작업은 이미지를 통한 물체의 구별이 필수적이다. 딥러닝은 이미지 분류에 있어서 많은 발전을 해왔고 그 결과 또한 높은 수준까지 올라와 있다.

이번 연구에서는 이러한 문제점을 딥러닝을 통해 해결방안을 확인하고자 하였다. 디바이스에 의존하지 않기 위해 클라우드 서비스인 코랩을 사용하였고 높은 수준의 정확도를 얻기 위해 사전 학습된 모델을 사용하여 신경망을 설계하였다. 이러한 방법을 통해 만들어진 신경망은 최대 90%이상의 검증 정확도를 보여주었고 어느정도 실제 산업에서 쓰일 수 있는 수준의 정확도라고 생각된다.

초창기에 설계한 신경망의 경우 80%정도의 낮은 정확도를 가지고 있었다. 90%이상의 결과를 볼 수 있었던 것은 이미 높은 수준의 정확도를 갖는 모델을 사용하고, 연구에서의 원하는 목표에 알맞은 데이터를 선정하여 학습을 시킨 것에 있다고 생각한다.

이 결과물을 통해 분리수거에서의 AI도입은 충분히 실현이 가능하다고 보여지며, 실제 디바이스와의 연결을 통해 산업 혹은 일상 생활에 재사용율을 늘릴 수 있을 것이다.

참고문헌

- [1] 박원기. "딥러닝 소개와 금융업 적용 사례." 국내석사학위논문 고려대학교 정책대학원, 2018. 서울
- [2] 김연규. "CNN과 특징 추출 알고리즘을 통한 한글 인식 성능 개선에 관한 연구." 국내석사학위논문 부산대학교 대학원, 2017. 부산
- [3] 이현호. "CNN의 일반화 오류 평가 방법." 국내석사학위논문 부산대학교 대학원, 2020. 부산
- [4] 사이토 고키, *밑바닥부터 시작하는 딥러닝*, 한빛미디어 (2019.01)
- [5] 조태호, *모두의 딥러닝*, 길벗 (2020.09)
- [6] 김환희, *시작하세요! 텐서플로 2.0 프로그래밍*, 위키북스 (2020.05)