ISSN 2765-5482

데이터과학연구

Journal of Data Science

제10권

2021년 9월

데 이 터 과 학 연 구 소

The Research Center for Data Science

데이터과학연구소 임원 및 연구부

- 소장 : 임예지 교수 (응용통계학과, yaeji@cau.ac.kr)
- 간사 : 곽일엽 교수 (응용통계학과, ikwak2@cau.ac.kr)

- 연구부

박상규 교수 (응용통계학과)
김삼용 교수 (응용통계학과)
김영화 교수 (응용통계학과)
이재헌 교수 (응용통계학과)
성병찬 교수 (응용통계학과)
박재현 교수 (컴퓨터공학과)
임창원 교수 (응용통계학과)
김원국 교수 (응용통계학과)
황범석 교수 (응용통계학과)
이재우 교수 (산업보완학과)
이주영 교수 (응용통계학과)

데이터과학연구 편집위원회

편집위원장 : 임예지 교수 (응용통계학과)
편집 위원 : 곽일엽 교수 (응용통계학과)

Journal of Data Science

2021. 9 Vol.1

Contents

The Research Center for Data Science

데이터과학연구 제10권
2021년 pp.1-18

Deep Learning Based Flood Detection Using SAR Image Data

김시현[1], 원혜진[1], 김창우[1], 문정현[1]

Abstract

A flood occurs when there is an overflow of water that inundates a portion of land that is normally dry, and it can happen in several ways. For example, a flood can be caused by an excess of rainwater in saturated ground, overflow of water bodies such as rivers and lakes, rapid snow or ice melting, storm surge or tsunami. Such events are exacerbated by climate change effects. Flooding is a natural hazard that causes a lot of deaths every year and the number of flood events is increasing worldwide because of climate change effects. Annually, floods cause more than $40 billion in damage worldwide. Detecting and monitoring floods is of paramount importance in order to reduce their impacts both in terms of affected people and economic losses. Our goal is to detect flood events from satellite imagery. In this work we compare the accuracy and the prediction performances of recent Deep Learning algorithms for the pixel-wise water segmentation task.

1) All authors have equal contribution, Department of Applied Statistics, Chung-Ang University, Seoul 06974, Korea

1. 서론

1.1. 연구배경 및 목적

홍수는 매년 많은 사망자를 발생시키는 자연재해이며, 기후변화 영향으로 인해 전 세계적으로 홍수 사건의 수가 증가하고 있다. 홍수는 일반적으로 건조한 토지의 일부를 범람시키는 물이 넘쳐흐를 때 발생하며, 여러 가지 방법으로 발생할 수 있다. 예를 들어, 홍수는 포화 지면의 과도한 빗물, 강이나 호수 같은 수역의 범람, 빠른 눈이나 얼음 용해, 폭풍해일 또는 쓰나미에 의해 발생할 수 있다. 이러한 사건들은 기후 변화 영향에 의해 악화될 수 있다. 대부분의 홍수는 극심한 강우에 의해 유발되며, 따라서 어느 정도까지는 미리 예측 할 수 있다. 매년, 홍수는 전세계적으로 400억 달러 이상의 피해를 야기한다. 2007년부터 2016년까지 수해로 5553명이 숨졌고 2017년 한 해에만 3331명의 사망자가 발생했으며 점점 더 홍수에 대한 피해량은 증가하고 있는 추세이다 (OECD 2016). 홍수를 감지하고 모니터링하는 것은 홍수의 영향을 받는 사람들과 경제적 손실 측면에서 영향을 줄일 수 있는 측면에서 중요하다. 그래서 현재 전 세계적으로 홍수에 대한 영향을 줄이기 위한 새로운 접근법과 도구의 필요성을 지적하면서 홍수의 심각성이 증가하고 있다 (Below와 Wallemacq, 2018). 이는 우리가 참가하게 된 ETCI 2021 Competition on Flood Detection 대회가 나오게 된 배경이다. 본 대회의 목표는 개방 water flood area를 묘사하는 접근 방식을 개발하고자 한다. 또한, Syntetic aperture rader(SAR) 이미지 데이터셋에 대해 알고리즘을 훈련 한 후 flood label 픽셀을 식별하는 알고리즘을 개발하고자 한다. 이를 위해 우리는 픽셀 단위 물 segmentation 작업에 관한 최신 Deep Learning 알고리즘의 예측 정확도를 비교할 것이다. 물 흐름에 대한 모니터링은 효과적인 조기 경고를 구현하기 위한 핵심이며, 현장 데이터의 분석은 조기 사건 감지와 홍수 영향의 실시간 이해에 기여할 수 있다. 이를 위해 인공지능(AI) 기반 알고리즘을 통한 영상 및 영상 자동분석을 통해 영상 및 영상 자동분석을 할 수 있다. 알고리즘의 예측 정확도는 Intersection Over Union(IOU) 점수로 평가된다.

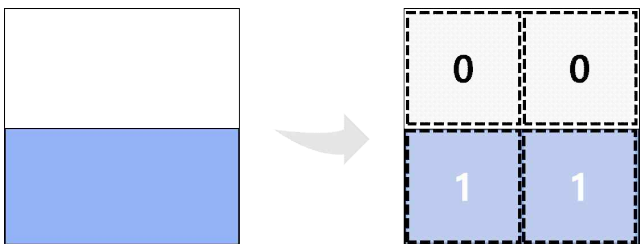
2. 기존 방법론

Computer Vision 분야에서 최근 자주 사용되는 모델들의 기반은 Semantic Segmentation이다. 먼저 Semantic Segmentation에 대해 이해하기 위해 Semantic Segmentation의 기초 모델을 먼저 소개하고 이를 바탕으로 우리 팀이 대회에 벤치마킹한 모델과 우리가 적용시킨 모델에 대해서 소개하도록 한다.

2.1. Semantic Segmentation Task

Segmentation은 모든 픽셀의 레이블을 예측하는 것으로 대표적으로 FCN, SegNet, DeepLab 등의 모델들이 있다 (Chen 등, 2018, Badrinarayanan 등, 2015, Chen 등, 2017). Semantic Image Segmentation의 목적은 사진에 있는 모든 픽셀을 해당하는 class로 분류하는 것이다. 이미지에 있는 모든 픽셀에 대한 예측을 하는 것이기 때문에 Dense Prediction이라고도 불린다. Input으로 RGB Color 이미지(Height X Width X 3) 또는 흑백 이미지(Height X Width X 1)이며, Output으로는 각 픽셀들이 어느 Class에 속하는지 나타내는 레이블을 나타낸 Segmentation Map이다.

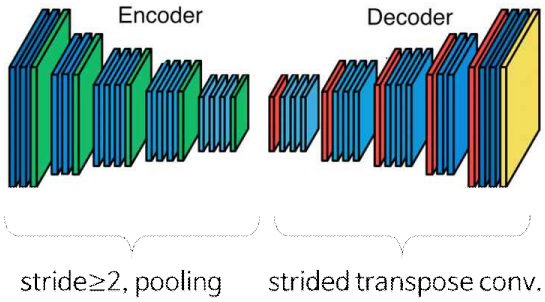
[그림 1] 2x2 Semantic Segmentation 예시



Semantic Segmentation 모델들은 보통 Downsampling & Upsampling의 형태를 가지고 있다. Downsampling이라 불리는 Encoder의 주목적은 차원을 줄여서 적은 메모리로 깊은 Convolution을 할 수 있게 하는 것이다. 보통 Stride를 2이상으로 하는 Convolution을 사용하거나 Pooling을 사용한다. 이 과정을 통해서 Feature의 정보가 손실된다. Upsampling이라 불리는 Decoder는 Downsampling을 통해서 받은 결과의 차원을 늘려 Input과 같은 차원으로 만들어주는 과정이다. 주로 Strided Transpose Convolution을 사용한다 (Dumoulin과 Visin, 2016).

GAN의 Encoder와 Decoder 구조와 비슷하다.

[그림 2] Downsampling, Upsampling의 모형

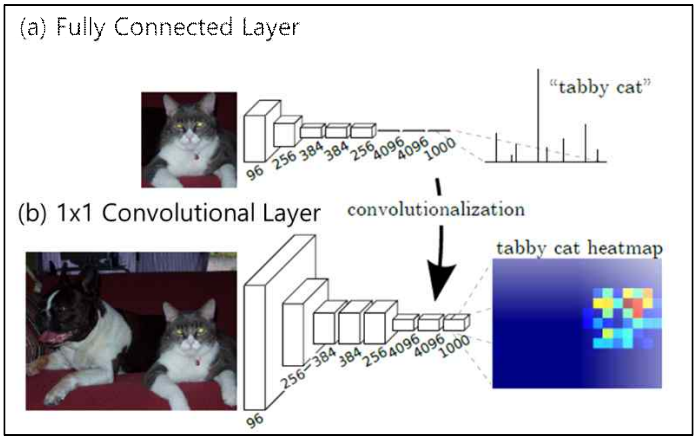


2.2. Fully Convolutional Network for Semantic Segmentation (FCN)

Semantic Segmentation 형태의 모델을 상용화하는데 큰 몫을 한 것은 Fully Convolutional Network for Semantic Segmentation(FCN)이다 (Long 등, 2015). 이전에는 Fully Connected layer로 모델을 만들었다. FCN이 나온 후 어떤 크기의 이미지로도 Segmentation Map을 만들 수 있게 되었고 그 때 SoTA였던 Patch Classification 보다 훨씬 빠르게 결과를 낼 수 있었다. 이후에 나오는 Semantic Segmentation 방법론은 거의 대부분 FCN을 기반으로 했다고 할 정도로 큰 임팩트를 주었다. FCN 구조가 탄생하게 된 배경에는 이미지 분류에 상용화 되어 있는 CNN 구조를 사용하여 이미지 내의 물체들을 Segmentation 하려는 아이디어에서 비롯되었다 (Simonyan과 Zisserman, 2015). 하지만 CNN의 구조를 차용하는 데 2가지 문제가 있었다. 첫 번째 문제는 입력 이미지의 크기가 고정되어야 한다는 점이다. Convolution 연산은 입력 이미지의 크기가 달라도 진행할 수 있지만 Fully-Connected 계층은 크기가 고정되어 있기 때문에 입력 이미지의 크기가 달라지면 입력을 받을 수 없는 문제가 생기게 된다. 두 번째 문제는 물체의 위치 정보가 사라지게 된다. Fully-Connected 계층으로 Feature Map이 넘어가는 순간, 데이터가 1차원으로 Flatten 되기 때문에 가지고 있던 위치적 정보는 사라지게 된다. 이러한 문제들로 Fully-Connected Layer([그림 3] (a)) 대신 1x1

Convolution layer([그림 3] (b))로 대체하자는 발상을 가지게 되었다.

[그림 3] Fully-Connected Layer와 1x1 Convolutional Layer

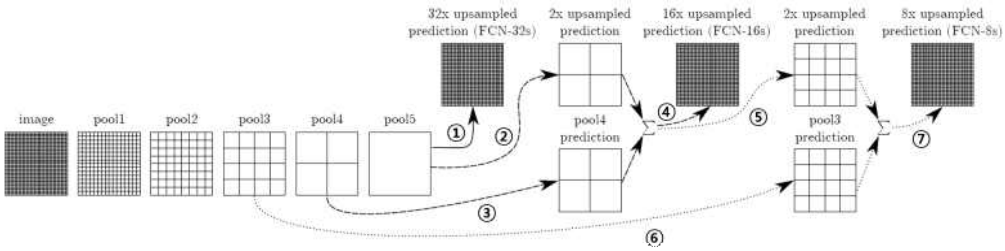


[그림 3] (b)에서 볼 수 있듯이 사진 내의 물체가 고양이 임을 알아냄과 동시에 그 물체의 위치까지 대략적으로 파악할 수 있게 되었다.

2.3. Skip Connection

Pooling을 반복할수록 지역적인 위치 정보가 사라지므로, 현재 레이어에서 Upsample한 결과와 이전 Pooling 레이어에서 예측한 결과를 같이 이용하면 더 정확한 Prediction이 가능할 것이다. Skip Connection 기법이 이 아이디어를 구현하고 있다.

[그림 4] Skip Connection 구조



[그림 4]의 번호를 보면서 Skip Connection을 소개하면,

① Pool 5 레이어에서 픽셀 단위로 예측한 Prediction Map을 32의 Stride로 Deconvolution해서, FCN-32s라는 결과를 출력한다. FCN-32s는 Upsample을 32배의 해상도로 진행했다는 얘기로, Segmentation이 비교적 자세하지 못한다. 출력된 결과는 원본 이미지를 기반으로 물체의 위치, 모양, 클래스를 예측한 Segmentation Map이 된다.
② Pool 5 레이어를 거친 Prediction Map에 쌍선형 보간(Bilinear Interpolation)을 적용하여 해상도를 두배로 늘리면서 초기화한 2x Upsample레이어를 생성한다. Prediction Map의 픽셀들을 사용하여 해상도를 임의로 두배로 올린 레이어를 만드는 것이다. 이 레이어를 학습 후, Prediction map을 생성한다.
③ Pool 4에서 픽셀 단위로 예측한 Prediction map을 구한다.
④ ②에서 만든 2x Upsampled Map의 Prediction Map과 ③에서 구한 pool4 레이어의 Prediction Map을 더해서 단일 Prediction Map을 생성한다. 논문에서는 이 더하는 작업을 단순한 원소간 합으로 정의한다. 이 Map은 16 Stride로 Upsample되어, FCN-16s Segmentation Map을 생성한다. 이 맵은 FCN-32s보다 더 높은 해상도를 가진 Segmentation을 수행한 결과를 보여준다.
⑤ ④에서 만든 Prediction map을 기반으로 다시 쌍선형 보간을 이용하여 초기화한 2x Upsample 레이어를 생성하고 학습한다. 그리고 그 레이어에서 Prediction Map을 생성한다.
⑥ Pool 3 레이어에서 픽셀 단위로 예측한 Prediction Map을 구한다.
⑦ ⑤에서 만든 2x Upsampled Map의 Prediction과 ⑥에서 구한 Pool3 레이어의 Prediction Map을 더해서 단일 Prediction Map을 생성한다. 이 Map은 8 Stride로 Upsample되어, FCN-8 Segmentation map을 생성하게 된다. 이 맵 또한 FCN-16s보다 높은 해상력을 가지는 Segmentation의 결과를 보여주게 된다.

3. 연구 방법론

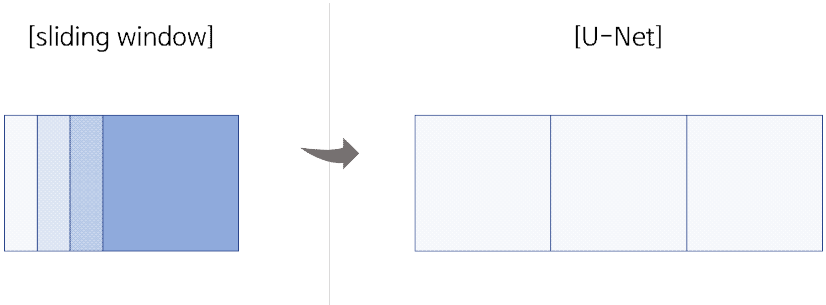
Segmentation Map의 크기는 입력 이미지와 동일하게 모두 같으므로, 2배로 Upsample된 Prediction을 16 Stride로 Devolution한 결과와 4배로 Upsample된 prediction을 8 stride로 Deconvolution한 결과는 Segmentation의 정확도에서 차이가 있을 것이다. Prediction의 해상도가 높을수록 Segmentation의 정밀도가 높아질 것이기 때문이다. 따라서 FCN을 기반으로 한 U-Net 구조의 모델과 이를 보완하기 위한 대체 방안으로 'PAN'구조의 모델을 통해 홍수 탐지를 진행한다.

3.1. U-Net-Architecture

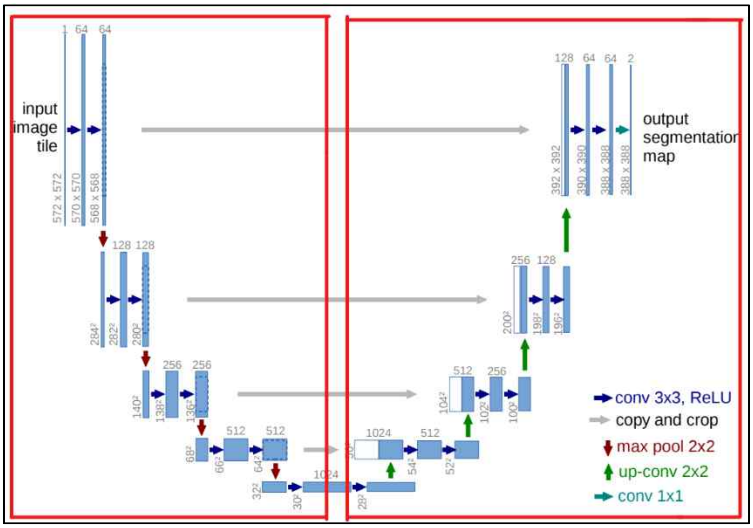
U-Net은 Fully Convolution Network(FCN)을 기반으로 하며, 적은 데이터를 가지고 정확한 Segmentation을 하기 위해 FCN 구조를 변경하였다 (Ronneberger 등, 2015). 여기서 Path는 이미지 단위로 인식하는 것이며, Context는 이웃한 픽셀들 간의 관계, 즉, 글을 읽고 문장을 이해하듯이 이미지도 이미지의 일부를 보고 다음 이미지를 파악하는 것이다. Patch/Tile은 이미지를 인식하는 단위이다.

U-Net은 2가지 장점을 가지고 있다. 첫 번째 장점은 기존 네트워크들의 단점인 느린 속도를 개선하여 속도가 빨라졌다. 속도 향상이 가능했던 이유는 Overlap 비율이 감소하였기 때문이다. [그림 5]와 같이 기존 Sliding Window 방식을 사용하게 되면, 이전 Path에서 검증이 끝난 부분을 다음 Path에서 다시 검증하기 때문에 낭비가 심했었다. 하지만 U-Net은 검증이 끝난 곳은 건너뛰고, 다음 Patch부터 새로운 검증을 하므로 속도가 빨라지게 된다. 두 번째 장점은 Context 인식과 Localization의 Trade-Off에 빠지지 않는다. 만약 Patch Size가 커지면 큰 범위의 이미지를 한 번에 인식을 하기 때문에 Context를 인식하는데 효과가 좋다. 하지만 Localization에서 안 좋은 영향을 끼치게 된다. 왜냐하면 너무 큰 범위를 한 번에 인식하기 때문에 Localization에서는 수행을 제대로 하지 못하게 된다. 반대로 범위를 작게 조정하면 세밀한 Localization이 가능하지만 Context의 인식률이 낮아지게 된다.

[그림 5] U-Net의 장점

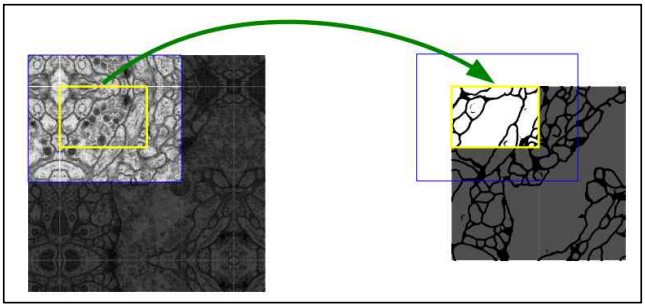


[그림 6] U-Net Architecture(Lowest resolution에서 32x32 픽셀의 예시



[그림 6]에서 각 파란색 Box는 다중 채널의 Feature Map에 해당된다. 채널의 수는 Box위에 보여 진다. X-Y-Size는 Box의 왼쪽아래 가장자리에 제공이 된다. 흰색 Box는 Copied Feature Maps를 표현하였다. 화살표들은 다른 Operations를 나타낸다. 회색 화살표는 Contracting Path(Encoding)과 Expansive Path(Decoding)에 의해 대응이 된다는 것을 알 수 있다.

[그림 7] Overlap-Tile Strategy

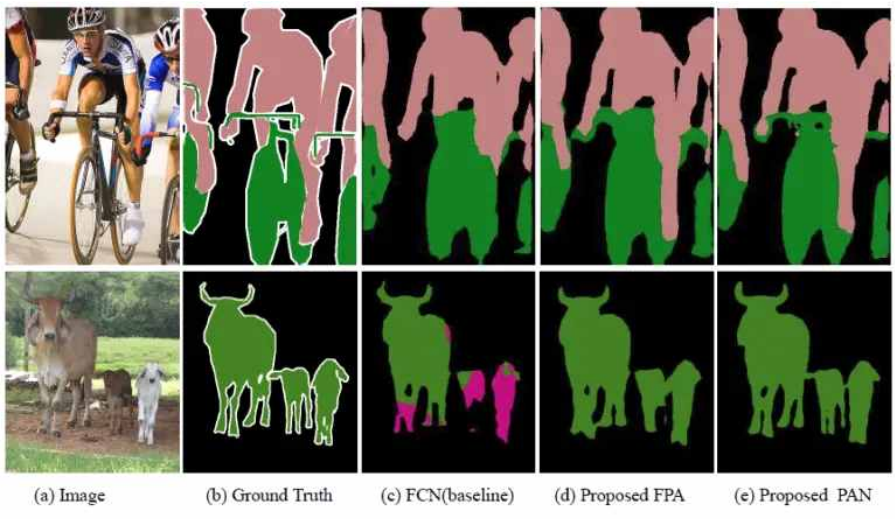


[그림 7]의 Overlay-Tile Strategy 그림에서 보이듯이 파란색 Tile과 노란색 Tile이 있는 것을 볼 수 있다. 즉, 파란색 Area를 가지고 노란색 Area의 Segmentation을 예측했다는 것을 알 수 있다. 그리고 노란색 Area 바깥의 Missing Data는 Mirroring을 가지고 보정을 하였다. Mirroring이란 [그림 7]의 Overlay-Tile Strategy의 오른쪽 그림의 파란색 사각형 안의 빈 부분을 노란색 Area에 인접한 부분을 거울에 반사된 것처럼 그 부분을 복사를 하여 채워졌다는 것이다. 이러한 Overlay-Tile Strategy를 통해서 좋은 이미지 해상도를 가지게 되며, GPU의 메모리를 효율적으로 사용한다는 것이 U-Net의 특징이다.

U-Net은 주로 3x3 Convolution을 사용하고 있으며, 각 Convolution Block은 3x3 Convolution이 2개씩 이루어져 있는 것을 볼 수 있다. 그리고 그 사이에는 Dropout이 있다. [그림 6]의 빨간 왼쪽 부분인 Contracting Path에서의 Block은 3x3 Convolution Block 2개가 이루어진 것이 총 4개의 형태이다. 그리고 각 Block은 빨강색 화살표인 Maxpool 2x2를 사용하여 사이즈를 줄여주면서 다음 Block으로 넘어가게 된다. [그림 6]의 오른쪽 부분 Expanding Path에서는 3x3 Convolution Block에 초록색 화살표인 Up-Conv 2x2 라고 불리는 것이 앞에 붙어있다. 즉, Contracting path에서 줄어든 사이즈를 다시 키워가면서 Convolution Block을 이용하는 형태이다. 그리고 아래쪽의 단계에서 얻어진 Feature들과 Concatenate하여 사용하였다.

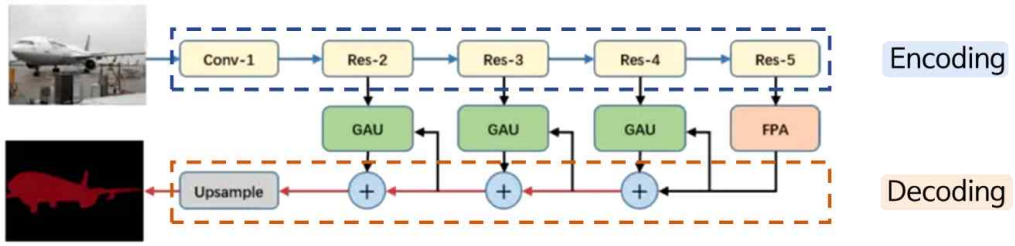
3.2. Pyramid Attention Network(PAN)

[그림 8] 여러 방법을 통한 mapping 결과



[그림 8]의 (c)를 보면 FCN의 경우 자전거의 핸들이 사라진 것을 확인할 수 있다. 이러한 손실 문제는 Unet 구조를 사용하거나 Kernel을 크게 잡으면 어느 정도 해결된다고 알려져 있다. 다만 연산 량이 많아지는 문제가 있다. 이러한 점을 보완하기 위해 Upsampling 단계에서 Attention 모듈을 사용하는 기법인 Global Attention Upsample(GAU)을 사용한 Pyramid Attention Network(PAN)이 제안되었다 (Li 등 2018).

[그림 9] Overview of Pyramid Attention Network

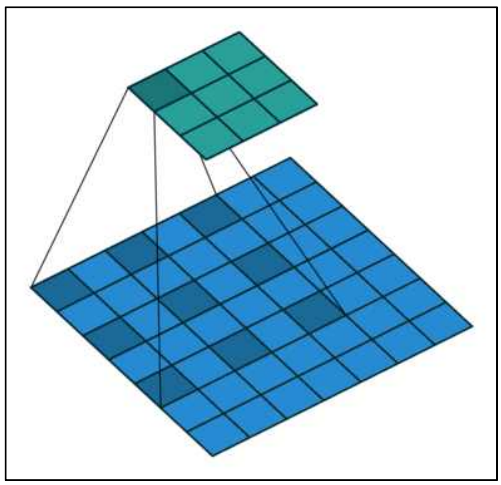


[그림 9]를 보면 네트워크는 Encoder-Decoder 아키텍처로 구성되어 있다. Res-5까지 거쳐서 인코딩 된 결과가 Feature Pyramid를 통과해 Attention을 받아, 원본

크기로 다시 커지게 된다. 파란색 선이 Downsampling, 빨간색 선이 Upsampling 이다.

PAN의 구조에서는 인코더 부분에서 Dilated Convolution을 사용할 수 있다는 것이 특징이다. Dilated Convolution은 필터 내부에 Zero Padding을 추가해 강제로 Receptive Field를 늘리는 방법이다.

[그림 10] 2D Conv. using a 3-kernel (with a dilation rate of 2 and no padding)



[그림 10]을 보면 진한 파란 부분만 Weight가 있고 나머지 부분은 0으로 채워진다. Receptive Field는 필터가 한 번 보는 영역으로 사진의 feature를 추출하기 위해선 receptive Field가 높을수록 좋다. Pooling을 수행하지 않고도 Receptive Field를 크게 가져갈 수 있기 때문에 Spatial Dimension 손실이 적고 대부분의 Weight가 0이기 때문에 연산의 효율이 좋다. 공간적 특징을 유지하기 때문에 Segmentation에서 많이 사용한다.

4. 데이터 구성 및 실험 설명

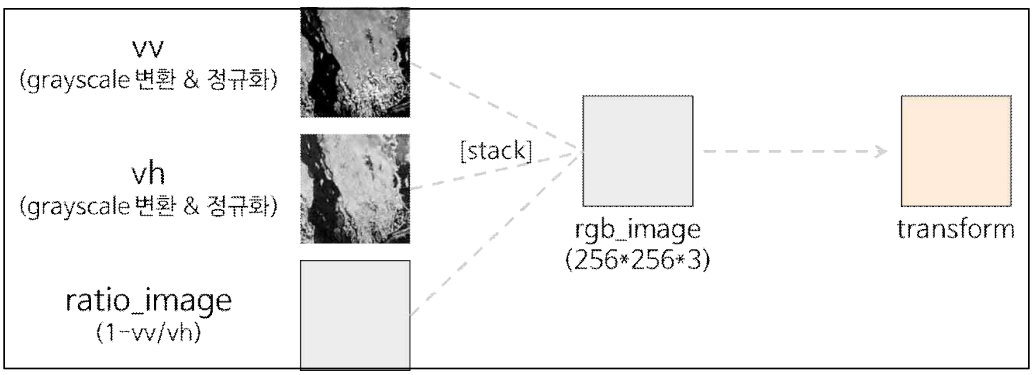
4.1. 데이터 구성 및 전처리

[표 1] Train, Validation Set

bangladesh		nebraska		northal		florence	
date	image	date	image	date	image	date	image
	vv		vv		vv		vv
	vh		vh		vh		vh
17.03.14	water	17.01.08	water	19.04.07	water	18.05.10	water
	body		body		body		body
	label		label		label		label
	vv		vv		vv		vv
	vh		vh		vh		vh
17.03.12	water	17.11.16	water	19.06.06	water	18.05.22	water
	body		body		body		body
	label		label		label		label
...	...	...	...	...	...	...	...

[표 1]은 데이터 구성 요약표이다. 4개의 지역마다 여러 시점에 촬영한 위성 사진으로 구성되어 있으며, 총 43,805장으로 Train Set과 Validation Set을 9:1로 Random Split했다.

[그림 11] 데이터 전처리 방법



[그림 11]과 같이 VV, VH Tile을 변환 및 정규화 이후 RGB Image 파일로 합친 후 Transform을 진행하였다.

Transformation을 적용하는 것은 Input Image와 Mask 모두에 동일한 변환을 적용해야 하기 때문이다. Pytorch를 통해 데이터 셋을 만들고 전처리 과정을 진행하였다. Albumentations 라이브러리를 통해 Trained Models의 질을 높이는 Augmentation을 진행하였다. 또한 주최측에서도 말하기를 데이터 세트를 탐색할 때 Noise가 있는 불규칙한 이미지를 발견할 수 있다고 했다. Noise Data는 SAR 이미지의 가장자리에 있는 일부 타일에는 정보가 없으며, 이러한 Train, Validation Data를 Filtering도 진행해 보았다. Filtering을 진행했을 때 약 20%의 이미지 데이터가 Noise Data로 분류되었다.

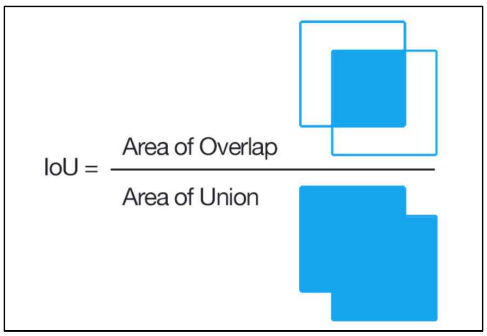
4.2. 실험 설명

U-Net과 PAN 모델을 사용하였으며 각자의 Encoder는 VGG, ResNet, Mobilenet 등을 사용하였으며, 데이터 전처리 방법들도 달리 하였다. 또한 Colab의 GPU, GPU TITAN XP를 사용하여 실험하였으며, Pretrained 가중치를 사용하지 않는 방법과 사용하는 방법을 사용하였다. Transform 또한 사용한 방법과 사용하지 않은 방법 등을 사용하였으며 데이터 처리를 Random Split으로 나누고 Noise Data를 제거한 Data Set과 제거하지 않은 Data Set으로 나눠서 실험을 진행하였다. Epoch과 Batch Size 또한 여러 방법으로 실험하였다. Learning Rate는 0.001 , Optimizer는 Adam Optimizer, Loss는 Cross Entropy Loss를 사용하였다.

5. 결론

5.1. 실험 결과

[그림 12] IOU(Interaction Over Union



실험 결과 성능은 [그림 12] Intersection over Union(IOU)로 두 박스의 교집합의 영역을 합집합 영역 넓이로 나눠서 얼마나 일치하는지 여부를 판단하는 평가 지표이다 (Rahman과 Wang, 2016).

[표 2]에서 PAN보다 U-Net의 성능이 더 좋은 것을 확인할 수 있으며, Noise data를 제거하지 않은 data set으로 실험하는 것이 더 좋은 성능임을 볼 수 있다. 또한, Pretrained 가중치를 불러와서 학습시키는 것이 성능이 더 높은 것을 확인할 수 있으며, epoch이 높으면 오버피팅으로 성능이 좋지 않게 나오는 것을 확인할 수 있다. Vgg16, Vgg19 보다는 encoder에서 resnet34를 사용하는 것이 좋은 성능을 기록했다. 대회 마감일 6월 30일 기준 IOU score 0.537 로 5등을 기록했다. (ETCI 2021 Competition on Flood Detection, 2021).

[표 2] 모델 결과 및 IOU Score

Model	Data	Hyper-parameter	Pretrain	DATA Transform	Encoder name	Score (IOU)
U-net	train/val/test	epoch=20 batch size=8	-	yes	resnet34	0.3
U-net	train/val/test	epoch=5 batch size=64	-	yes	resnet34	0.3719
U-net	train/val/test	epoch=10 batch size=32	imagenet	no	resnet34	0.5370
U-net	noisy data filtering train & val/test	epoch=50 batch size=64	-	yes	vgg16	0.072
U-net	noisy data filtering train & val/test	epoch=50 batch size=64	imagenet	yes	vgg16	0.083
U-net	noisy data filtering train & val/test	epoch=50 batch size=32	imagenet	no	vgg16	0.024

U-net	train & val/test	epoch=30 batch size=64	imagenet	yes	vgg19	0.0408755773
U-net	train & val/test	epoch=15 batch size=64	imagenet	yes	resnet34	0.2659540006
U-net	train/val/test	epoch=50 batch size=64	imagenet	yes	vgg16	0.066
PAN	train/val/test	epoch=10batch size=16	imagenet	yes	mobilenet_v2	0.01
PAN	random		imagenet	yes		

5.2. 한계점 및 시사점

여러 한계점 및 시사점이 남는데 첫 번째로는, input data를 충분히 활용하지 못했다는 점이다. VV, VH의 비율을 정의하여 새로운 이미지 데이터로 학습을 진행하였으나 water body 이미지는 활용하지 못했다. 만약 water body 이미지를 포함하여 학습을 진행하였다면 더 좋은 홍수 탐지 성능을 기대할 수 있었다고 생각한다. 두 번째, PAN 모델의 overfitting이 발생했다. U-Net을 보완하기 위해 PAN 모델을 훈련시켰으나, 오히려 과적합이 발생하는 것을 확인하였고, 시간이 촉박해 과적합을 해결하지 못했다. 세 번째, computing resource의 한계가 있었다. Colab GPU를 이용했음에도 불구하고 제한이 생기고, 학습에 오랜 시간 소요 및 오류가 발생해 보다 좋은 환경에서 훈련했을 때, 다양한 모델 실험의 가능성과 성능 향상을 기대할 수 있었을 것이라는 아쉬움이 남는다.

참고문헌

[1] OECD (2016). Financial Management of Flood Risk, p. 136.

[2] Below, R. and Wallemacq P. (2018). Annual Disaster Statistical Review 2017, *Centre for Research on the Epidemiology of Disasters (CRED)*

[3] Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F. and Adam, H. (2018) Encoder-decoder with atrous separable convolution for se-mantic image segmentation, *European Conference on Computer Vision (ECCV),* 833-851.

[4] Badrinarayanan, V., Kendall, A. and Cipolla, R. (2015) Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12), 2481 - 2495.

[5] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A. L. (2017) Deeplab: Semantic image segmentation with deepconvolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 40(4), 834-848.

[6] Dumoulin, V. and Visin, F. (2016) A guide to convolution arithmeticfor deep learning, *ArXiv e-prints*.

[7] Simonyan, K. and Zisserman, A. (2015) Very deep convolutional networksfor large-scale image recognition, *International Conference on Learning Representations (ICLR)*

[8] Long, J., Shelhamer, E. and Darrell, T. (2015) Fully convolutional networks for semantic segmentation. *In Proceedings of the IEEE*

conference on computer vision and pattern recognition (CVPR), 3431-3440.

[9] Ronneberger, O., Fischer, P. and Brox, T. U-net: Convolutionalnetworks for biomedical image segmentation. (2015) *Medical Image Computing and Computer-Assisted Intervention (MICCAI),* Springer, LNCS, Vol.9351: 234—241.

[10] Li, H., Xiong, P., An, J. and Wang, L. (2018) Pyramid attentionnetwork for semantic segmentation. *ArXiv e-prints*.

[11] Rahman, M. A. and Wang, Y. (2016) Optimizing intersection-over-unionin deep neural networks for image segmentation. *International Symposium on Visual Computing (ISVC)*

[12] ETCI 2021 Competition on Flood Detection. (2021). URL: https://nasa-impact.github.io/etci2021/

데이터과학연구 제10권
2021년 pp.19-30

A Study on Counting Faces from a Given Image

김나린[1], 김수빈[1], 김수현[1], 양성원[2], 이영[1]

Abstract

This paper explores 1-stage detector and 2-stage detector to solve face counting issue among objection detection problems. We use the fifth version of You Only Looks Onces YOLO model and Faster Region Proposal Convolutional Neural Network (faster RCNN) model as 1-stage model and 2-stage model, respectively. As a result, Yolo 5 performs better than faster RCNN in the perspectives of both time and accuracy. To train YOLO 5 model, we preprocess the image data to the format of the YOLO 5 input image which is the size of 640×640 and modify the information of the bounding box. After the image set is resized by three methods, it is trained by four kinds of YOLO 5 model. The x-large model with white background has the lowest RMSE among the 12 performances. However we think that the inconsistency of the bounding box of the training data reduces the overall performance of the model.

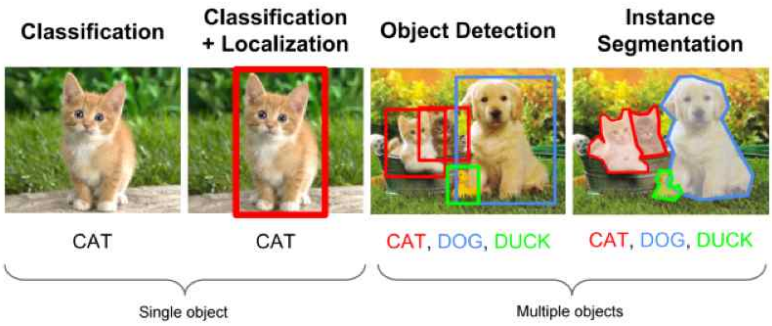
1) All authors have equal contribution, Department of Applied Statistics, Chung-Ang University, Seoul 06974, Korea

2) All authors have equal contribution, School of Economics, Chung-Ang University, Seoul 06974, Korea

1. 서론

우리는 Focus Face 팀명으로 Analyticsvidhya 의 Face Counting Challenge 에 참가했다. 이미지에 존재하는 사람의 얼굴 수를 count 하는 것이 본 대회의 주제이며 평가 지표는 test data에서 예측한 얼굴 수에 대한 RMSE 이다. face counting 문제는 컴퓨터 비전의 object detection 의 한 분야다. Liu 등 (2020)에 따르면 Object detection은 물체를 구분하는 문제; classification 와 물체의 위치 탐지; localization 를 합한 문제이다. 최근에는 개체의 정확한 경계까지 파악하는 Segmentation 문제도 제시되고 있다. Classification 과 localization 을 동시에 해결하는지 여부에 따라 1-Stage Detector와 2-Stage Detector로 구분할 수 있다. 일반적으로 2-Stage Detection이 1-Stage Detection에 비해 정확도가 높다는 장점이 있고, 비교적 시간이 오래 걸려 자율주행과 같이 실시간 탐지가 필요한 분야에 적용하기 어려운 단점이 있다. 우리 팀은 두 가지 방법을 모두 적용하여 face counting 시도했다. 이론과 달리 2-Stage Detection에서 loss 가 더 높게 나왔다.

[그림 1] The kinds of object detecting problem.

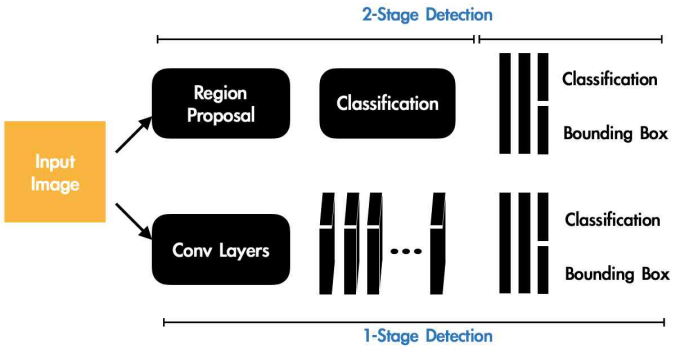


2. 방법론

2.1. 데이터

5,733개 image와 그에 대응하는 bounding box 정보를 training data, 2,465개 image를 test data로 제공받았다. image의 width는 345에서 612까지 height는 261에서 612까지 다양한 크기로 구성되어 있다.

[그림 2] Structures of 1-stage and 2-stage detector.

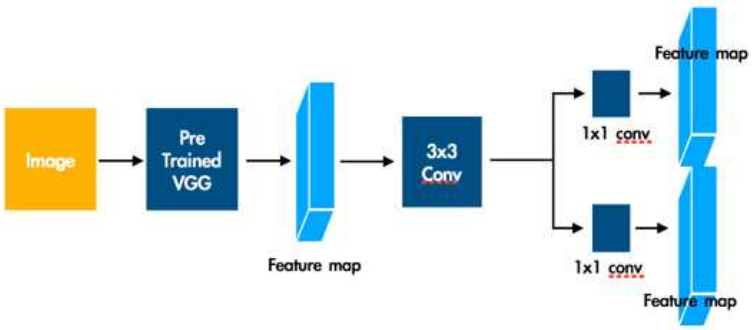


2.2. 2-Stage Detector : Faster Region Proposal Convolutional Neural Network

먼저 Ren 등 (2015)을 통해 Fast RCNN과 가장 큰 차이점인 Region Proposal Network (RPN) 을 살펴본다. Faster RCNN 은 원본 이미지를 통해 Feature map을 얻는 것으로 시작한다. 800×800×3 의 input image를 VGG-16에 입력하여 50×50×512 크기의 Feature map을 생성한다고 가정한다. 첫째로 50×50 의 각 픽셀마다 가로와 세로 크기 3개와 비율 3개를 설정하여 총 9개를 만든다. 즉, 총 22,500개의 anchor box를 설정한다. 둘째로 RPN을 통해 region proposal을 형성해야 한다. 3×3 conv 연산을 통해 동일한 크기 (50×50×512) 의 Feature map을 생성하고, 두 개의 1×1conv 연산을 통해 classification (

$50\times50\times2\times9$) 과 bounding box regression ($50\times50\times4\times9$)을 계산한다. 22,500개의 anchor box와 classification 및 bounding box regression을 입력 값으로 하는 Proposal layer를 통해 상위 N개의 region proposals을 남기는 작업을 한다. 이때, Non-Maximum Suppression을 사용한다. 이후 Ground truth label을 통해 각각의 anchor box가 물체를 포함하고 있는지 분류하는 과정을 따른다. Ground truth label은 anchor box와 ground truth box의 IoU (IoU = Area of Overlap / Area of Union)를 계산하여 IoU가 0.7이면 1(Positive), 0.3 ~ 0.7이면 0 그리고 0.3이하이면 -1(Negative) 으로 설정한다. 다음으로 Max pooling과정을 따르는데, 이후의 과정은 Fast RCNN과 동일하다.

[그림 3] The result of baseline model of faster RCNN



참여한 Challenge에서 2등을 기록하고 있는 깃허브[3]를 참조하여 Baseline모델을 실험해본 결과 [그림 4] 를 얻었다. 정확도가 높지만 시간이 오래 걸린다는 단점이 있는 2-Stage Model의 일반적인 모습과 상반되게 loss가 높게 나와 개선이 필요했으며, 이어지는 1-Stage Model 이 시간과 정확도에서 더 나은 성능을 보였기 때문에 더 자세한 실험을 진행하기로 한다.

[그림 4] The result of baseline model of faster RCNN

```
8 [==============================>.] - ETA: 0s - loss: 1.6680 - rpn_reg_loss:
0.7055 - rpn_cls_loss: 0.1629 - frcnn_reg_loss: 0.5327 - frcnn_cls_loss: 0.2
4138/4138 [==============================] - 3158s 763ms/step - loss: 1.6681
 - rpn_reg_loss: 0.7055 - rpn_cls_loss: 0.1629 - frcnn_reg_loss: 0.5328 - fr
cnn_cls_loss: 0.2669 - val_loss: 2.4211 - val_rpn_reg_loss: 0.9442 - val_rpn
 cls_loss: 0.1831 - val_frcnn_reg_loss: 0.9505 - val_frcnn_cls_loss: 0.3434
```

3) https://github.com/AbhinayReddyYarva/FaceCountingChallenge-AnalyticsVidhya

2.3. 1-Stage Detector : You Only Look Once 5th. version

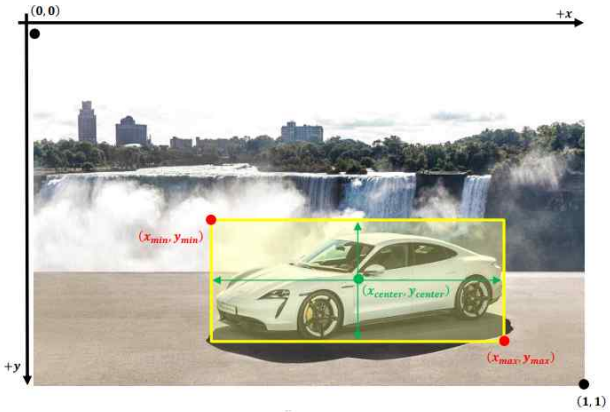
기존에 제안된 방법들 중에서 Single Shot Multi Box Detector (SSD) 를 활용한 open API 를 적용하여 우리 데이터를 training하지 않고 예측만 수행한 결과 성능이 좋지 않았다. 챌린지에서 제공한 image가 노이즈가 많다고 추측할 수 있었다. 다음으로 1-Stage object detection의 대표적으로 알려진 YOLO 계열 모델을 적용했다. Divvala 등 (2016)의 논문에 따르면 기존 모델들과 비교했을 때 크게 3가지 장점이 있다. 첫 번째로 localization과 classification 문제를 하나의 회귀문제로 보고 해결 할 수 있기 때문에 매우 빠르다. 또한 예측을 할 때 이미지의 전체를 고려하기 때문에 sliding window나 region proposal과 다르게 물체에 대한 정보뿐만 아니라 주변 정보까지 학습하여 문제를 해결한다. 마지막으로 YOLO 는 기존 모델들에 비해서 학습하지 않은 새로운 이미지에 훨씬 robust 하다. 따라서, 새로운 도메인에 적용하여도 비교적 성능이 뛰어나다.

우리 팀은 가장 최근에 개발된 YOLO fifth version (YOLO v5) 모델로 훈련 및 예측을 했다. Thuan (2021)의 document에 따르면 YOLO는 $S \times S$ 그리드를 생성하고 그리드 안 bounding box를 예측한다. Bounding box 레이블은 $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$ 형태 [그림 5]이며 YOLO label format에 맞게 bounding box의 중심 값과 가로, 세로 길이가 담긴 vector로 변환한다. 또한, YOLO 의 input format은 그리드 셀 내 bounding box의 상대적인 위치 값인 normalization 된 값

$$
\begin{aligned}
x_{center} &= \frac{x_{\min} + x_{\max}}{2} \\
y_{center} &= \frac{y_{\min} + y_{\max}}{2} \\
width &= x_{\max} - x_{\min} \\
height &= y_{\max} - y_{\min}
\end{aligned}
$$

으로 변환한다. 객체의 중심이 그리드 셀 안에 위치한다면, 그 그리드 셀은 해당 객체를 검출한다. 셀 내부에 B개 의 bounding box와 그 bounding box에 대한 confidence score를 예측하고, bounding box에 특정 개체가 나타날 확률과 예측된 bounding box가 객체에 얼마나 잘 맞는지를 계산한다.

[그림 5] Bounding box of YOLO model.



3. 실험

3.1. Model Settings

YOLO 의 Model Architecture 은 Backbone(CSPNet), Neck(PANet), Head(YOLOv4) 으로 설정하며, Activation Function 은 Leaky ReLU (middle/hidden layers), Sigmoid (final detection layer) 이다. Optimization Function 와 Loss Function 으로는 SGD 와 Binary Cross-Entropy with Logits 을 각각 사용했다.

3.2. Model Settings of YOLO 5th version model for our data

YOLO 의 input image는 width와 height 비율이 1 : 1 이고 32배수의 크기만 허용 한다. 따라서 image resize를 모델 학습 전에 선행했다. 전체 5, 773개

image의 width와 height의 최댓값은 612이었고 이것으로 width × height = 640 × 640 기준을 설정했다. 첫 번째로는 [그림 6] 의 원본 (a) 사진이 (b) 처럼 늘어나도록 image의 비율을 무시하고 늘리는 방식을 시도했다. 두 번째로는 640 × 640 크기의 흰색 배경을 생성하는 아이디어에 착안하여 [그림 6] (c) 와 같이 resize 했다. 기존 train image 5, 773개에 5개 이하의 얼굴이 포함된 이미지가 4, 553개였고 20명이 넘는 군집 이미지도 있었다. 따라서 우리는 5개 이하의 face counting 성능을 올리기 위해서 흰 배경처리 후, 5개 이하의 얼굴이 포함된 이미지를 반으로 줄인 후 추가한 10, 326개의 데이터셋 까지 총 3가지 training set을 준비했다.

YOLO model 학습에서 bounding box 정보는 image의 좌측상단을 (0,0) 으로, 우측하단을 (1,1) 로 정의한다. 따라서 bounding box의 중심과 width, height 상대적인 좌표 값을 요구한다. 따라서 앞 서 준비한 세 가지 training set에 대응하여 각각 bounding box 정보를 수정하여 학습에 활용하였다.

Pre-trained YOLO v5 모델의 4개 버전 (s, m, l, x) 을 비교하기 위해 batch size를 2, 4, 8로 조정하고 3개의 training image set에 적용해 보았다. training dataset을 training 80%, validation 20%로 나누어 학습했다. 이후 업데이트된 파라미터와 test data 2,463장으로 얼굴 수를 예측했다.

[그림 6] Samples of resized image.

(a) origin (612 x 261) (b) aspect ratio change (c) white background (d) less than 5 faces

4. 결론

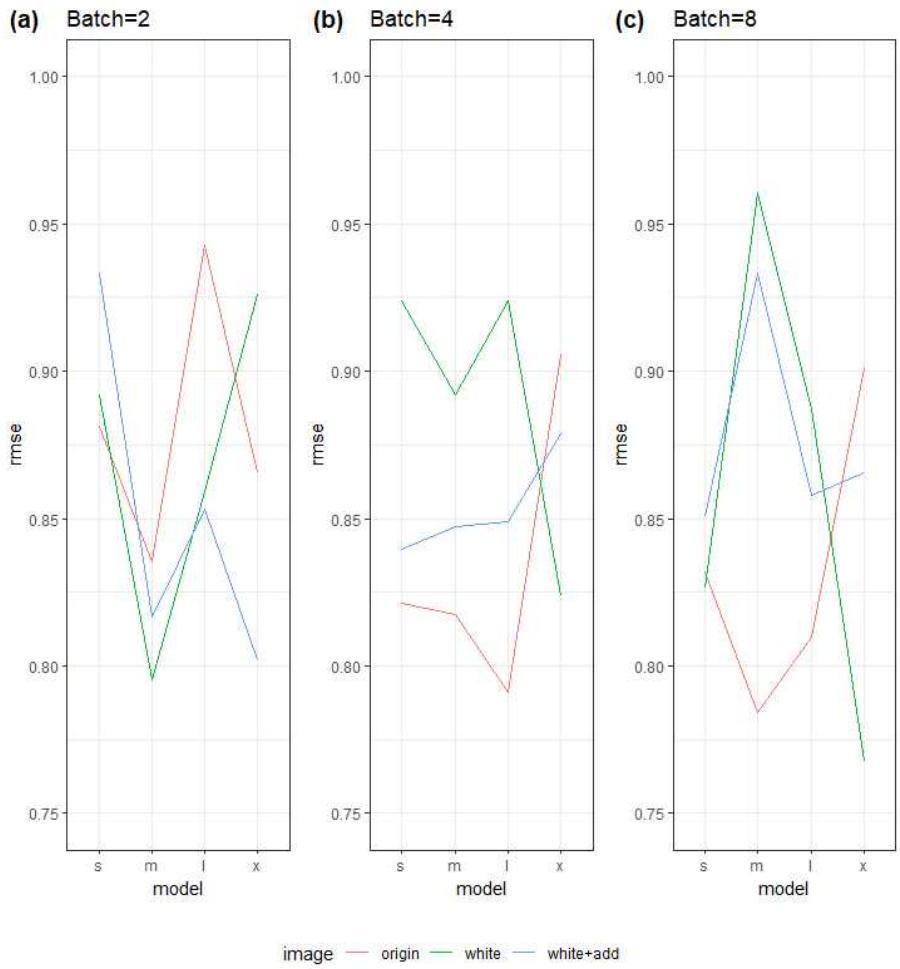
우리 데이터 분석 결과는 1-Stage Detection 방식인 YOLO 의 성능이 faster RCNN 보다 좋았으며, 이는 여러 개의 object 이 아닌 face 한 종류를 detection

하는 문제이기 때문으로 추정한다. YOLO v5 모델을 사용하여 3 가지 image set을 통해 학습 후 평가한 결과, [그림 7] 에서 확인 할 수 있듯이 test RMSE가 0.7677에서 0.9606까지 분포하였고 모형의 복잡도와 batch size에 따라 특정 패턴이 있다고 보기는 어렵다. [표 1] 과 [그림 7] (c) 의 초록색 line 가장 우측 결과에서 볼 수 있듯이, 흰 배경을 더한 image set으로 x model 을 batch size = 8로 학습시킨 후 테스트 한 경우가 성적이 가장 좋았다. 이 결과를 대회에 제출하였을 때 7월 13일 기준으로 1위를 기록하였다. 추후 연구로 Optimize 할 때 SGD 뿐만 아니라 Adam으로 바꾸어 학습이 가능하다. 또한, Anchor box를 도메인에 맡게 새로 구성하여 실험할 수 있다. (Glenn Jocher이 정한 가장 best한 5개의 anchor box가 디폴트)

Faster RCNN 의 경우는 Feature Map의 특성을 최대한 이용하는 방법을 구상해보았다. Featurized Image Pyramid (FPN) 을 이용하면 ResNet101에서 각 단계의 Feature Map을 모두 이용할 수 있는데, 저수준과 고수준의 특성 및 화질을 모두 이용할 수 있는 장점이 생긴다. 또한 RoI Pooling과 FC layer는 데이터 손실이 일어날 수밖에 없는 구조인데, Pooling 과정에서 데이터 손실을 줄이는 방법으로 각각을 RoI Align과 Global Average Pooling (GAP) 으로 사용하면 데이터 손실이 적을 것이다. 위 두 방법을 실제 적용하지는 못했지만, 정확성을 높이는 것이 Face Counting 의 목표인 만큼 추후에 시도를 해볼 예정이다.

아쉬운 점은 챌린지에서 제공한 training data의 bounding box의 정확한 기준이 공지되어 있지 않았다는 사실이다. [그림 8] 과 같이 (a) 이미지는 뒷 편의 작은 얼굴들은 잡아내지 않았다. 또한 [그림 8] 의 (b)처럼 어떤 image는 뒤통수에 bounding box가 있는 반면에 다른 image (c) 에는 그렇지 않은 경우가 존재했다. 위 부분이 조금 더 명확했다면 더 좋은 성능이 나왔을 것으로 보인다.

[그림 7] Test RMSE for 3 batch sizes (2, 4, 8) and 4 YOLO v5 models (s, m, l, x) with 3 resized image sets respectively.

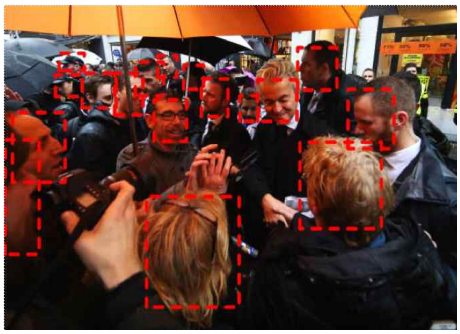


[표 1] Test RMSE for 3 batch sizes and 4 YOLO v5 models with 3 resized image sets respectively.

	batch size	s	m	l	x
setting 1	2	0.88	0.84	0.94	0.87
	4	0.82	0.82	0.79	0.91
	8	0.83	0.78	0.81	0.90
setting 2	2	0.89	0.80	0.86	0.93
	4	0.92	0.89	0.92	0.82
	8	0.83	0.96	0.89	0.77
setting 3	2	0.93	0.82	0.85	0.80
	4	0.84	0.85	0.85	0.88
	8	0.85	0.93	0.86	0.87

[그림 8] Uncertainty in the bounding box criteria

(a)



(b)

(c)

참고문헌

[1] AnalyticsVidhya (2021). Face counting challenge. *https://datahack.analyticsvidhya.com/contest/vista-codefest-computer-vision-1*.

[2] Jocher, G. (2015). You only looks onece (yolo) v5. *Github https://github.com/ultralytics/yolov5*.

[3] Thuan, D. (2021). Evolution of YOLO Algorithm and YOLOv5. The State-of-the-art Object Detection Algorithm (Bachelor's Degree in Information Technology). Oulun ammattikorkeakoulu.

[4] Reddy (2020). Face count challenge 2nd positioned. *Github https://github.com/AbhinayReddyYarva/FaceCountingChallenge-AnalyticsVidhya*.

[5] Ren, S., He, K., Girshick, R. and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems (NIPS)*, 28, 91-99.

[6] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016). You only look once: Unified, real-time object detection. *In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* pp. 779-788.

[7] Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X. and Pietikäinen, M. (2020) Deep Learning for Generic Object Detection: A Survey. *Int J Comput Vis* 128, 261-318

데이터과학연구 제10권
2021년 pp.31-43

Cryptocurrency Price Forecasting Using CNN

이은희[1], 서기태[1], 김백승,[1] 김수연[1]

Abstract

The price of cryptocurrency is very volatile, so predicting the accurate price is an important issue. We built a CNN (Convolutional Neural Network) model on cryptocurrency price data based on studies that showed CNN model is usefull for extracting features of time series data not only for image data. The Hyper-parameters of the CNN were determined by Random-Search Methods, and its performance was compared with ARIMA, LSTM and GRU. Finally, we propose pre-processing method that can improve the performance of CNN model on time series data. This study was submitted using data from Season 3 of the AI Bit Trader Contest hosted by DACON.

1) All authors have equal contribution, Department of Applied Statistics, Chung-Ang University, Seoul 06974 , Korea

1. 서론

한국은행의 국제경제리뷰(2020.04)에 따르면 COVID-19의 세계적 대유행이 장기화 되면서 각국의 확산 억제조치와 외부활동 자제로 인해 세계 실물경제 전반에 상당한 부정적 영향이 초래되었다. 이에 각국의 정부는 경제 침체를 극복하기 위해 통화 정책으로 금리인하를 시행하였고, 그 결과 많은 투자자금이 주식시장과 암호화폐 시장으로 유입되었다. 2021년은 암호화폐 시장이 활성화되었지만 비트코인을 포함한 암호화폐 가격은 주가보다 변동성이 커 정확한 가격을 예측하는 데 어려움이 있다.

본 연구에서는 데이콘(DACON)에서 주최하는 '인공지능 비트 트레이더 경진대회에 참여하여 암호화폐의 가격 변동을 예측하는 딥러닝 모형을 구현한다. 주된 관심이 되는 모형은 CNN 모형으로 CNN(Convolutional Neural Network)은 일반적으로 이미지 데이터 분석에서 뛰어난 성능을 보여왔으며, 주가 예측과 같은 시계열 데이터 분석에도 많이 활용되고 있다. 또한, 시계열 데이터 분석을 위해 많이 사용되는 LSTM보다 Training Time, Computational Intensity 측면에서 더 효과적이라는 분석 결과도 존재한다 (Cao 등, 2019). 이 같은 이유로 CNN을 핵심 모형으로 설정하였고, 비교 모형으로는 시계열 데이터에 사용되는 전통적인 통계 분석 기법인 ARIMA, 시계열 데이터에 가장 일반적으로 사용되는 딥러닝 모형인 RNN(LSTM, GRU)을 사용한다.

본 논문의 구성은 다음과 같다. 2장에서는 연구에서 사용한 각 모형에 대해 알아본다. 3장에서는 대회에서 제공하는 데이터에 대해 알아보고 더 나아가 수익률을 최대화하기 위한 전략을 설명한다. 4장에서는 모형 별 성능을 비교하고 5장에서는 본 분석의 한계점과 개선 방안에 대해 논의한다.

2. 모형 설명

2.1. ARIMA (Auto-Regressive Integrated Moving Average)

시계열 자료에 사용되는 대표적인 통계 모형으로는 시계열 데이터 Y_t를 이전 시점의 관측값 $Y_{t-1}, Y_{t-2}, \cdots, Y_1$의 선형 모형으로 적합시키는 AR(p) 모형과 Y_t를 현재와 과거의 백색 잡음(White Noise) 의 선형 모형으로 적합시키는 MA(q) 모형, 그리고 두 가지를 모두 고려한 ARMA(p,q)가 있다. 하지만, 위 모형들은 정상적인

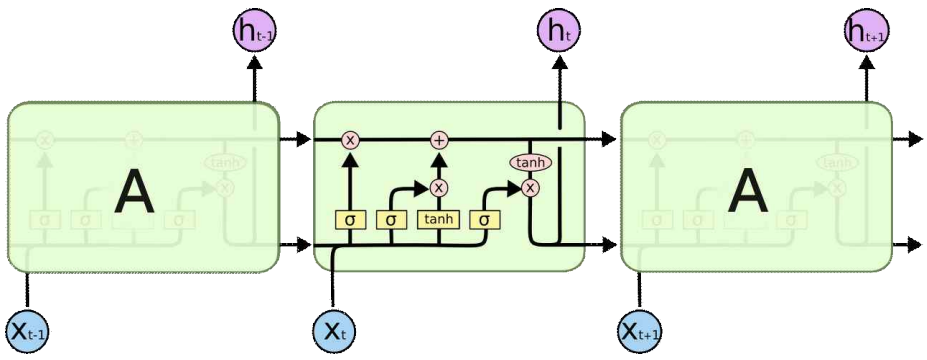
시계열(Stationary Time Series)에 대한 모형식으로 대부분의 시계열 자료가 비정상적(Non-Stationary)인 현실 자료에서 적절하지 않다. 따라서, 시계열 자료가 정상적이도록 차분(Differencing)을 한 뒤에 ARMA(p,q)를 적용하는 ARIMA(p,d,q)이 등장하였고, 일반적인 모형식은 아래와 같다 (Zhang 등, 2003).

$$ARIMA(p,d,q) \Rightarrow (1-\phi_1 B-\phi_2 B^2-\cdots-\phi_p B^p)(1-B)^d Y_t = c+(1+\theta_1 B+\cdots+\theta_q B^q)$$

2.2. LSTM (Long Short-Term Memory)

LSTM은 RNN의 BPTT(Backpropagation Through Time) 과정에서 발생하는 문제를 해결하고자 제안된 모형으로 기존의 RNN 모형의 Hidden state에 Cell state가 추가된 모형이다. 기존의 RNN은 Input 노드가 방향성을 가지고 연결되어 있기 때문에 t시점의 시계열 반응 변수 Y_t를 X_t뿐만 아니라 이전 시점의 X_{t-k}까지 사용하여 예측하게 된다. 하지만 이전 시점의 정보 X_{t-k}와 사용하는 시점 t와 거리가 멀 수록 학습 능력이 떨어지는 기울기 소실 (Vanishing Gradient), 기울기 폭발 (Exploding Gradient) 문제를 가지고 있다. 따라서 LSTM은 이전 시점의 정보 X_{t-k}를 기억 혹은 망각하도록 하는 Cell State가 존재하여 State가 오래 지속되더라도 정보의 손실이 비교적 적다. (Hochreiter 등, 1997) LSTM의 Cell State를 도식화한 것은 [그림 1]과 같다.

[그림 1] LSTM의 Cell State 구조



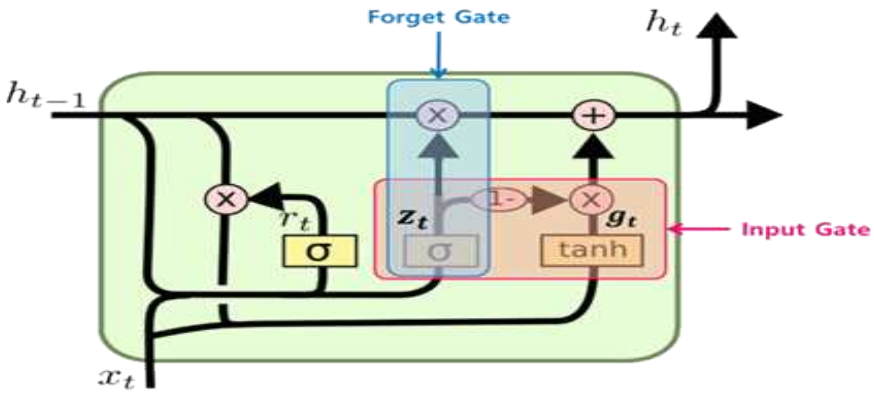
2.3. GRU (Gated Recurrent Unit)

GRU는 LSTM의 Cell State 구조가 간소화된 모델로 Cho 등 (2014) 에서 소개되었다. LSTM과 기본적으로 동일한 구조를 가지고 있지만 LSTM보다 학습효율이 좋고 LSTM과 비슷한 성능을 가지고 있어 시계열 데이터 분석에 자주 사용된다. GRU의 Cell State 구조를 도식화한 것은 [그림 2]와 같다.

2.4. CNN (Convolutional Neural Network)

일반적인 다층 인공신경망(DNN) 계층은 입력 데이터를 일렬로 배치한 후 (Flatten) 이전 은닉층과 다음 은닉층이 완전이 연결되도록 (Fully Connected) 연결한 이전 은닉층과 다음 은닉층이 완전이 연결되도록 (Fully Connected) 연결한 형태이지만, CNN은 해당 계층 이전에 컨볼루셔널 층(Convolutional Layer)을

[그림 2] GRU의 Cell State 구조



도입하여 계층별 뉴런들이 컨볼루셔널 필터(Convolutional filtter)를 통해 다음 뉴런을 위한 값으로 반복해서 계산되고, 풀링(Polling), 활성함수(Activation function), 드롭아웃(Drop out) 등을 통해 데이터에 특징을 추출한다. 일반적으로 CNN은 이미지 분석에 뛰어난 성능을 보여왔지만 시계열 데이터의 특징을 인식하는 것에서도 뛰어난 성능을 보인다 (Jin 등, 2020).

3. 데이터 설명과 사용 전략

3.1. 대회 소개 및 데이터 설명

본 대회의 목표는 비트 코인이 포함된 10가지 종류의 암호화폐 가격을 예측하여 가장 높은 수익률을 갖는 모델을 만드는 것이다. 학습에 사용되는 Train Data는 train_x와 train_y로 이루어져 있으며 train_x는 암호화폐의 23시간동안의 분 단위 정보(시점 t=1, ... ,1380), train_y는 train_x에 포함된 시점 이후 2시간에 대한 분 단위 정보 (시점 t=1, ... ,120)를 담고 있다. Test Data인 test_x는 train_x는 암호화폐의 23시간동안의 분 단위 정보(시점 t=1, ... ,1380)를 담고 있다. 데이터의 크기는 train_x, train_y가 각각 (10,942,020, 11), (951,480, 11)이며 test_x는 (1,048,800, 11)이다. 각 Data Set에 포함된 암호화폐의 정보(Columns)는 [표 1]과 같다.

본 대회는 '매수 비율'과 '매도 시점'을 결정하기 위해 아래와 같은 내용을 가정한다.

1) train_x와 test_x가 종료되는 시점, 즉, $t = 1380$에서 반드시 매수가 이루어진다.
2) train_x와 test_x의 종료 직후, 시점 $1 \leq t \leq 1380$에 매수한 코인을 반드시 매도해야 한다.
3) 이전 Sample_id에서의 투자 후 수익 (또는 손실)이 다음 Sample_id의 초기 자본금이 된다.

위의 3가지 가정을 만족하면서, Test Data에서의 모든 투자가 종료되었을 때 최종 수익률이 높아지도록 '매수 비율'과 '매도 시점'을 결정한다. 각 Sample_id에 따른 매수 비율과 매도 시점이 저장된 파일을 제출하면 최종 수익률이 계산된다. 위 내용을 그림으로 도식화하면 [그림 3]와 같다.

[표 1] 데이터 변수 설명

변수 명	변수 설명
Sample_id	개별 샘플의 인덱스
time	동일한 샘플 내 시간 정보
coin_index	10가지 종류의 코인에 대한 인덱스 (0~9)
open	open price
high	high price
close	close price
voluume	거래 량
quote_av	quote asset volume
trades	거래 건수
tb_base_av	taker buy base asset volume
tb_quate_av	taker buy quote asset volume

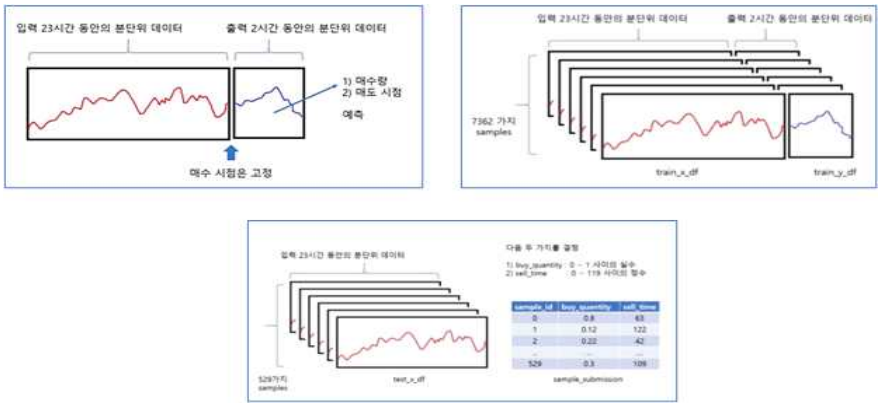
3.2. Scheme

우리가 예측해야하는 Time Series는 'open'이며, 23시간 이후, 2시간의 'open' Series를 예측하여 가장 높은 수익을 낼 수 있는 '매수 비율'과 '매도 시점'을 정하게 된다. 본 팀은 최종 수익률을 높이기 위해 'High risk, High return' 전략을 선택하였다. 이 전략에는 두 가지 상황이 있는데 각 상황에 따른 전략은 다음과 같다.

1) 예측한 120분 중 암호화폐의 가격이 1을 넘는 구간이 있을 때 고정된 매수 시점에서 매수 비율을 1로 하고(전량 매수) 예측한 암호화폐의 가격에 대해 120분 중 가장 높은 지점을 매도 시점으로 정한다.
2) 예측한 120분 중 암호화폐의 가격이 1을 넘는 구간이 없을 때 고정된 매수 시점에서 매수 비율을 0으로 하고 매도를 하지 않는다.

우리는 총 120분에 대해서 정확한 예측을 수행하는 모델을 적합하기만 한다면 이러한 전략이 최고의 수익률을 높일 수 있을 것이라고 생각하였다.

[그림 3] ‘매수 비율’과 ‘매도 시점’ 결정 과정 도식화



4. 실험 결과

4.1. 모형 적합

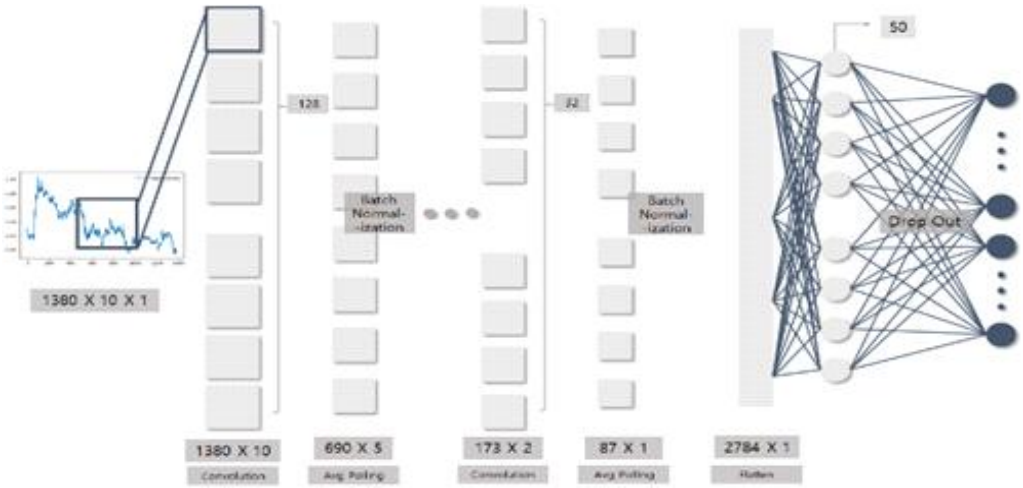
ARIMA는 시계열(Series)만을 이용해서 모형을 적합하기 때문에, Open Price를 Target Time Series로 선택하여 Sample_id 별로 Keras Library의 ‘auto_arima’ Function을 사용하여 모형을 적합했다. 단, auto_arima가 수렴하지 않을 경우에는 임의로 ARIMA(1,1,1)를 사용하였고, Test time points인 120분을 한번에 예측하였다.

RNN 모델의 경우 Open Price를 사용하여 LSTM과 GRU를 활용해 Layer를 구성했다. 하이퍼파라미터 튜닝 시에는 별도의 tuning method 없이 실험자가 층을 깊게 쌓아보고, 정규화를 해보는 등의 다양한 방식으로 실험을 병행하였다. RNN 모형은 초반 60분을 먼저 예측한 후, 이것을 기반으로 나머지 60분을 예측하는 Autoregressive(자기회귀) 방식을 선택하였다 (Guo 등, 2018).

CNN 모델은 Convolution Layer와 Flatten Layer로 구성하고 Overfitting을 방지하기 위해서 Dropout Layer를 추가하며 Layer를 다양하게 구성했고, Hyperparameter Optimization을 통해 최적의 하이퍼 파라미터의 값을 탐색했다. 불필요한 수행 횟수를 대폭 줄이며 정해진 간격(Grid) 사이의 값에 대해서 확률적으로 탐색이 가능한 Random Search 방법을 선택하였다. Random Search

사용을 위해 Keras에서 제공하는 Keras Tuner 라이브러리를 사용하였다. CNN 모델에 적용한 Tuning Range와 Search Result는 [표 2]를 통해 확인할 수 있다.

[그림 4] CNN Model Architecture with Random Search



4.2. 실험 결과

주어진 test_x Data의 각 Sample_id에 따라 예측한 매수 비율과 매도 시점 결과를 주최측에 제출하면 초기 자본금 10,000달러에서 최종 수익금(Score)을 계산해준다. 분석 결과 RNN 기반 모델들은 결과값이 평균에 몰리는 경향을 보였고, 정규화가 이러한 문제를 해결해 줄 것이라고 판단하여 MinMax Normalization을 시도하였지만 결과는 오히려 더 낮아졌다. 층을 깊게 구성했을 때보다 층을 단순하게 구성했을 때 더 높은 최종 수익률을 기록했다. CNN 모델 또한 RNN 기반 모델과 동일하게 층을 깊게 구성했을 때보다 층을 단순하게 구성했을 때 더 높은 최종 수익률을 기록했다. 모든 모형에 대한 Score 결과는 [표 3]와 같다.

[표 2] CNN의 Hyperparameter Tune Range, Result of Random Search

Parameter	Range	Best Value
Number of Convolutional Layers	[3, 5]	4
Size of filter of Convolutional Layers	[32 , 512]	layer1 = 128 layer2 = 352 layer3 = 256 layer4 = 32
Kernel size of Convolutional Layers	No tune	3
Activation function	No tune	Relu
Method of Polling	['Avg' or ' Max']	Avg (for all layers)
Padding	No tune	Same
Number of Flatten layer	No tune	1
Number of nodes of flatten layer	[30 , 100]	50
Dropout rate	[0 , 0.5]	0.06
Learning rate	[1e-04 , 1e-02]	0.00047782
Loss	No tune	MSE
Validation Loss = 0.000469		

[표 3] 적합된 모델의 최종 수익률

Model	Information	Score
ARIMA	Auto_Arima	9630.1477
LSTM	LSTM layer =1 / Dense layer = 1 LSTM Units = 16 / Epochs = 30	13533.8037
	LSTM layer = 2 / Dense layer = 2 LSTM Units = 32 / Epochs = 30	11911.3113
	LSTM layer = 2 / Dense layer = 2 LSTM Units = 32 / Epochs = 30 Applied MinMax Normalization	6176.9907
GRU	GRU Layer = 1 / Dense Layer = 1 GRU Units = 16 / Epochs = 30	4201.1918
	GRU Layer = 2 / Dense Layer = 2 GRU Units = 32 / Epochs = 30	6565.8799
	GRU Layer = 2 / Dense Layer = 2 GRU Units = 32 / Epochs = 30 Applied MinMax Normalization	6176.9907
CNN	Convolutional Layer = 2 Flatten Layer = 1 Size of filter of convolutional layer = 300 The number of nodes of flatten layer = 100 Dropout rate = 0.4 Learning rate = 0.001 Kernel size = 2 Polling size = 2 Activation function = Relu Method of polling = Max Epoch = 10	16320.6726
	Result of Random Serach	10062.6977

5. 결론 및 개선방안

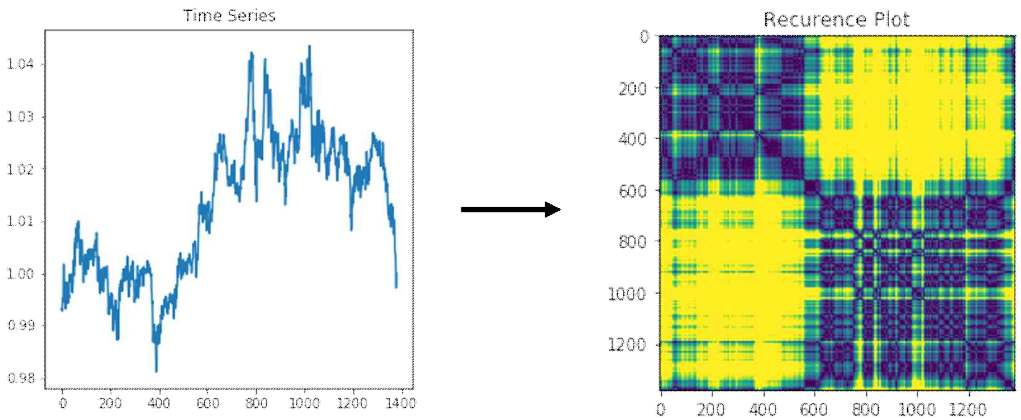
5.1. 결론

우리의 기대와 달리 CNN 모형은 다른 모형들에 비해 좋은 성능을 보이지 않았다. 이는 CNN 모형이 주어진 시계열 데이터가 갖고 있는 패턴을 충분히 잡아내지 못한 것으로 보인다. 우리는 그 원인이 CNN 모형의 구조적인 특징에 있다고 판단했다. CNN은 이미지처럼 격자 형태의 데이터의 특징을 추출하는 것에 특화된 Convolutional Layer를 가지고 있기 때문에, 방향성을 가지고 있는 시계열 데이터의 특징을 추출하기위해서는 Recurrrence Plot 알고리즘과 같은 전처리 과정이 필요하다.

5.2. Recurrence Plot 알고리즘

CNN은 기본적으로 이미지와 같은 격자 형태 데이터에 최적화된 모형이다. 따라서 시계열 데이터를 이미지로 표현할 수 있다면 CNN모델의 성능을 올릴 수 있을 것으로 예상된다. 우리는 시계열 데이터를 이미지로 표현하는 RP(Recurrence Plot) 알고리즘을 제안하고 비트 트레이더 데이터에 적용하는 방법을 모색하고자 한다. 시계열 데이터를 RP 알고리즘을 이용하여 이미지로 변환하는 과정은 두 단계로 구성된다 (Hatami 등, 2012).

[그림 5] 암호화폐 데이터에 RP 알고리즘을 적용한 결과



1) 시계열 데이터의 m=2차원 공간 궤적을 구성

2) m=2차원 공간 궤적을 RP 행렬로 표현

이를 우리의 데이터에 적용하면, x축을 Time, y축을 Price로 생각하여 시계열 데이터를 2차원의 공간 궤적으로 구성할 수 있다. 이 궤적을 이용하여 구한 거리 행렬은 x축, y축, 거리의 속성을 가지는 이미지 데이터로 볼 수 있고 CNN의 입력값으로 사용할 수 있다. 이를 적용한 결과는 [그림 5]과 같다. 또한, Zhang 등(2017)에 제안된 Mix-up Agumentation 기법을 사용한다면 CNN 성능 향상에 도움이 될 것으로 기대한다.

참고문헌

[1] “코로나19 글로벌 확산이 세계 경제에 미치는 영향.”「한국은행」2020년 04월 12일.

[2] Cao, J. and Jinghan W. (2019) Stock price forecasting model based on modified convolution neural network and financial time series analysis, *International Journal of Communication Systems* 32.12 e3987.

[3] Zhang, G. P. (2003) Time series forecasting using a hybrid ARIMA and neural network model, *Neurocomputing* 50, 159-175.

[4] Hochreiter, S. and Schmidhuber, J. (1997) Long short-term memory, *Neural Computation*, 9(8), 1735-1780.

[5] Cho, K., Merrienboer, B., Gulcehre, C., Bougares, F., Schwenk, H., Bengio Y. (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation, *Conference on Empirical Methods in Natural Language Processing (EMNLP2014)*, 1724-1734.

[6] Jin, X., Yu, X., Wang, X., Bai, Y., Su, T. and Kong, J. (2020) Prediction for Time Series with CNN and LSTM. *Proceedings of the 11th International Conference on Modelling, Identification and Control (ICMIC2019)*. 631-641.

[7] Guo, T., Lin, T., and Lu, Y. (2018) An interpretable LSTM neural network for autoregressive exogenous model, *6th International Conference on Learning Representations (ICLR2018).*

[8] Bergstra, J. and Bengio, Y. (2012) Random search for hyper-parameter optimization, *Journal of machine learning research* , 13, 281-305.

[9] Hatami, N., Gavet, Y. and Debayle, J. (2018) Classification of time-series images using deep convolutional neural networks, *10th international conference on machine vision (ICMV 2018).*

[10] Zhang, H., Cisse, M., Dauphin, Y. N. and Lopez-Paz, D. (2018) mixup: Beyond empirical risk minimization, *6th International Conference on Learning Representations (ICLR2018).*

데이터과학연구 제10권
2021년 pp.44-60

Classification and Text Mining of Identical Product Images for Price Match Guarantee

김원석[1], 정상욱[1], 권채원[1], 김다운[1], 박관열[1]

Abstract

In this study, we tried to make the price match guarantee efficient and fast using the images and labels of the products provided in the competition. We have tried analyzing and classifying several models using deep learning and machine learning techniques. Our chosen models were four image classification models (NFNet, EfficientNet, ResNet, EfficientNet+Arcface) and one natural language processing model (BERT). We applied two methods (Sobel Edge, Rotation) as augmentation methods for image data, and we trained classification models on two examples of training data from the original and extended data from the original to see how much the extended data affected the results.

1) All authors have equal contribution, Department of Applied Statistics, Chung-Ang University, Seoul 06974, Korea

1. 서론

1.1. 연구배경 및 목적

인터넷이 점점 더 발달함에 따라 점점 더 많은 업체가 자신들의 비즈니스를 전자 상거래 업체에 등록하고 있다. 그로 인해 판매 항목들이 날이 가면 갈수록 기하급수적으로 증가하고 있으며, 만족스러운 검색 및 구매 경험을 제공하기 어려워지고 있다.

전자 상거래 포털의 목적 중 하나는 구매자와 판매자 모두에게 제품 기반 경험을 도입하는 것이다. 구매자의 관점에서 이것은 동일한 실제 제품을 참조하는 오퍼를 그룹화하여 다른 판매자가 판매함으로써 검색 프로세스를 용이하게 함을 의미한다.

반면에 판매자는 고품질 제품 카탈로그에 접근하여 리스트를 만드는 과정을 가속화하고 구매자에게보다 보다 완전한 제품 설명을 제공할 수 있다.

전자 상거래 포털에서 제품 기반 경험을 달성하는 것은 동일한 제품을 자동으로 찾을 수 있어야만 가능하다. 이 과정을 제품 매칭이라고 한다. 전자 상거래에서 제품 매칭은 대부분 많은 제품, 제품간의 높은 이질성, 누락된 제품 및 다양한 수준의 데이터 품질로 인해 쉬운 작업이 아니다.

제품 매칭은 전자 상거래 포털에서 제품을 판매하는 많은 판매자에서 동일한 제품의 제안을 식별하고 해당 정보를 제품 카탈로그의 단일 항목에 통합하는 것을 목표로 한다. 오퍼는 공급 업체가 제공한 정보로 설명된 특정 상품의 사례이다. 이 정보에는 제목, 텍스트 설명, 속성, 카테고리 및 사진이 포함될 수 있다. 일반적인 전자 상거래 시장에서는 동일한 제품에 대한 많은 사례를 찾을 수 있다.(Janusz Tracz 등, 2020)

이전에는 제품 매칭은 종종 규칙 기반 방법과 문자열 유사성 측정과 같은 수작업 기능에 의존했었다. 하지만 딥러닝과 머신러닝이 발달함에 따라 자연어 처리 분야에서는 end-to-end 방법, 이미지 처리 분야에서는 ResNet, EfficientNet, NFNet

등 다양한 방식이 등장했고, 이러한 방식으로 좀 더 빠르고 정확하게 제품 매칭을 해 볼 수 있게 되었다.

「Shopee - Price Match Guarantee」 대회는 '최저가 보장'을 알고리즘으로 해결하고자 개최된 대회이다. 이 대회는 플랫폼을 이용하는 판매자들이 올린 수천 개 상품들의 판매글 제목과 이미지를 분석해 어떤 판매글의 이미지가 동일한 상품인지 분류하는 것을 목적으로 한다.

우리는 이 연구를 통해 딥러닝과 머신러닝 기반 방법으로 Price Match Guarantee를 위해 여러 가지 모형을 시도해 보고, 누구보다 높은 정확도를 가진 최적화된 모형을 만들고자 한다.

2. 데이터 소개

2.1. 데이터 수집

Shopee에서 제공한 34,250개의 Train Images를 제공해 이를 활용해 분류모형을 만드는데 활용하였다[2]. Test Images 70,000개에 대해서는 공개되지 않아 활용할 수 없었다.

Data set은 크게 사진에 대한 정보를 담고 있는 메타정보를 의미하는 train.csv 파일과 traing image 파일로 구성되어 있다. train.csv 파일은 총 5개의 컬럼으로 구성되어 있는데, Posting_id, Image, Image_phash, Title, Label_group으로 구성되어 있다.

2) https://www.kaggle.com/c/shopee-product-matching/data

[표 1] 사용한 변수 정리

	변수명	설명
이미지 정보	posting_id	이미지의 이름
	image	이미지 파일 이름
	image_phash	이미지 고유 해쉬키
	title	이미지 제목
	label_group	이미지 카테고리
이미지	image	이미지 파일

[표 1]은 분석에서 사용한 변수들을 정의한 표로써 image_phash를 제외한 나머지 변수 및 데이터를 이용하였다. 이를 이용해 데이터 전체를 training set으로 이용과 동시에 test set으로 이용하였다.

여기서 분석에 실제로 활용하는 값은 해당 row의 이미지가 무엇인지 알려주는 image 컬럼과, 그 이미지를 업로드할 때 판매자가 작성한 title 컬럼, 그리고 해당 이미지가 무엇을 의미하는 것인지 주최측에서 분류해 놓은 label_group 컬럼이다.

총 34,250개의 row로 이루어져 있는데, 이 중 중복되는 이미지도 존재해 유니크한 이미지의 값은 32,414개 이다. 이미지 파일은 image 컬럼에 있는 파일명으로 저장되어 있다.

두 범주의 데이터 모두 이 곳에서 제공하는 데이터를 이용해 두 가지 방법으로 데이터를 확장시켰는데 이 방법에 대해서는 2.2절에서 설명하고자 한다.

2.2. 데이터 확장

본 논문은 주최 측에서 제공된 중복 제거된 32,414개 이미지 외에도 학습데이터를 확장(Augmentation)하기 위해서 각도변환과 sobel-edge 두 가지의 데이터 확장방법을 사용했다. 두 가지의 방법을 적용 후 총 96,423개(32,414 * 3)의 학습데이터를 구성하였다.

[그림 1] 원본 이미지

2.2.1 Sobel edge Detection

첫 번째 과정인 Sobel Edge Detection은 Sobel Operator의 아이디어인 “Isotropic 3×3 Image Gradient Operator”를 활용했다. 우리는 파이썬을 이용하여 원본 이미지파일에 sobel-edge를 적용하였다. Sobel-edge가 진행되는 알고리즘은 다음과 같다.

1) 2차원의 이미지를 input한다.

2) 각 축 별로 Gaussian Blur 연산을 적용한다.

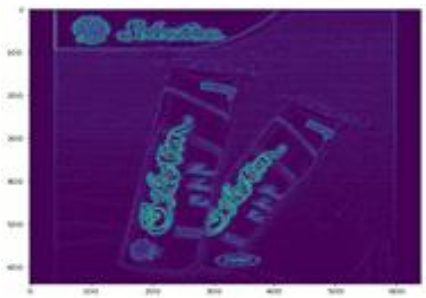
3) 색상을 RGB에서 Grayscale로 변환한다.

4) Sobel Kernel을 이용해 회선처리한다.

5) 회선 처리해 계산된 결과를 8-bit unsigned integer로 변환한다.

6) 이미지 결과를 확인한다.

[그림 2] 원본 이미지에 Sobel-edge 방법을 적용한 이미지



[그림 1]의 원본 그림에 Sobel-edge 필터를 적용하면 [그림 2]와 같은 데이터가 생성된다.

2.2.2 각도 변환

두 번째 과정인 각도 변환은 이미지를 각각 90° 회전시킨 이미지를 생성하는 것이다. 이미지의 각도를 변환시킴으로써 같은 라벨에 대해 더 다양한 데이터를 생성해 낼 수 있고, 학습과정에서 모형의 과적합을 방지할 수 있다. 따라서 이 각도의 변화로 인한 이미지 인식을 학습시켜 이미지를 올바른 레이블에 분류 하도록 데이터를 확장시킴으로써 정확도 증가를 기대했다.

[그림 3] 원본 이미지를 90° 회전시킨 이미지

예를 들어 [그림 1]의 원본 그림을 90° 회전시키면 [그림 3]과 같은 데이터가 생성된다.

실제로 이미지 분류 모형을 학습시킬 때에는 ①원본 학습데이터 32,414개 이미지를 활용한 방법과 ② 확장데이터를 포함한 96,423개 이미지를 활용한 방법에 대해서 각각 적용해 보았다.

3. 딥러닝 모형

3.1. 텍스트 모형

3.1.1 BERT

BERT란, 구글에서 개발한 자연어처리 사전 훈련 기술이며, 특정 분야에 국한된 기술이 아니라 모든 자연어 처리 분야에서 좋은 성능을 내는 범용 Language Model이다. BERT의 핵심 구조는 Transformer 모형에서 Encoder 부분만 사용하고, pre-training(사전학습)과 fine-tuning 시 구조를 조금 다르게 하여 Transfer Learning을 용이하게 만드는 것이다. BERT등장 이전에는 데이터의 전처리 임베딩을 위해 Word2Vec, GloVe, Fasttext 방식을 많이 사용했지만, 최근 고성능을 내

는 대부분의 모델에서 BERT를 많이 사용한다.

기존의 모형들은 문장의 길이가 길어질수록 첫 단어의 의미가 끝 단어의 의미까지 반영되기 어렵다는 한계점이 존재해서 Attention을 추가한 RNN 구조를 사용했지만, 연산 속도가 매우 느리다는 단점이 존재한다. 그 단점을 보완하고자 Attention만을 사용하는 신경망을 구성하는 방법이 제안되었고, 그 방법이 바로 Self-Attention을 사용하는 Transformer 모델이다. BERT는 Transformer 블록으로 이루어져 있고, 모델 속성이 양방향을 지향함으로써 성공적인 Performance를 이끌어 내고 있다 (Devlin 등, 2018).

BERT와 같은 Transformer 기반 모델은 자연어 처리의 광범위한 작업을 위해 최첨단 기술을 도입했다. 기업에 대한 사전 교육을 통해 Transformer는 작업별 미세 조정(fine tuning)을 위한 소량의 교육 데이터로도 우수한 성능을 제공할 수 있다.

제품 일치를 위해 BERT를 미세 조정하면 최신 프레임워크인 Deepmatcher보다 더 좋은 성능을 보여줄 수 있다 (Tracz 등, 2020). 또한, Peeters 등 (2020)의 연구에 따르면 BERT가 다른 baseline 모형에 비해 데이터의 양에 상관없이 좋은 성능을 낼 수 있다. 이런 식으로, BERT가 여러 논문에서 자연어 처리 방법 중에 가장 효율적인 방법 중 하나라는 것을 보았기에, 우리는 BERT를 사용하기로 하였다.

BERT는 이 데이터에서 Title 데이터로 모형을 학습시키고, 그 Title과 비슷한 상위 다섯 개를 찾아서 비교하여 1,2위의 문장의 이미지 아이디를 저장해놓는다. 그리고 그 두 개의 문장 유사도가 0.95 이상이면 count를 +1하고 posting_id라c고 저장하고, matching을 해보는 식으로 결과를 낸다.

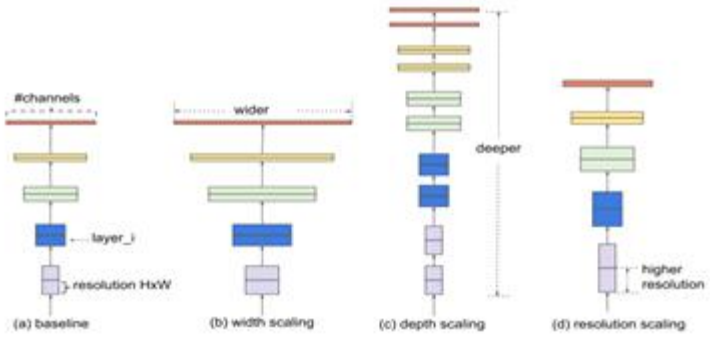
3.2. 이미지 모형

3.2.1 EfficientNet

Image 데이터 분류 모형에는 정확도(Accuracy)를 초점으로 한 모델과 효율성

(Efficient)을 초점으로 한 모델과 같이 크게 두가지 방향으로 제안되고 있는데, 이러한 모델들의 성능을 크게 상회하는 모델인 EfficientNet이 제안된 바 있다 (Tan과 Le, 2019).

[그림 4] 주요 Scaling factor



[그림 4]는 이미 존재하는 모델의 size를 키워주는 여러 방법들을 보여주고 있는데, 대표적으로 (b)Filter(Channel)의 개수를 늘리는 Width scaling과 (c)Layer의 개수를 늘리는 Depth scaling과 (d)Input image의 해상도를 높이는 Resolution scaling이 자주 사용된다. 기존의 연구에서는 3가지 Scaling의 Coefficient를 임의로 증가해도 정확도가 무조건 향상되지 않아 수동적인 튜닝 과정이 필요했다. 튜닝을 진행하더라도 산출되는 모형의 정확도와 효율성이 상충될 수 있다. 하지만 EfficientNet은 적절한 Depth, Width, Resolution을 찾기 위해 Grid search를 사용해 Alpha, Beta, Gamma를 구한다. Grid search를 통해 얻은 Network Width, Depth, Resolution을 균형적으로 확장하는 compound scale coefficient에 따라 depth, width, resolution의 크기가 증가하고, accuracy가 향상된다 (Tan과 Le, 2019).

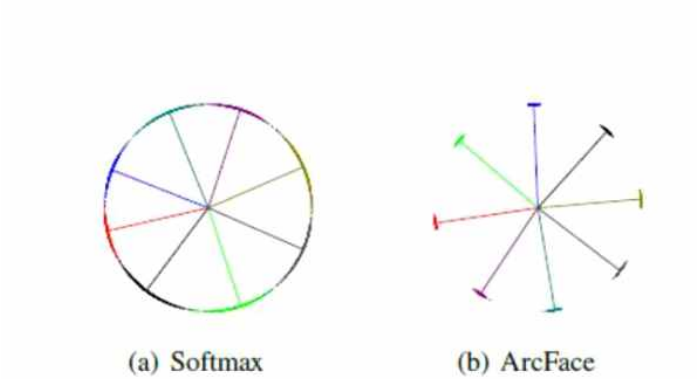
EfficientNet은 이렇게 효율적이면서도 정확도가 높은 모형으로, 2019년에 논문이 발표된 이후로 많은 사람들이 이미지를 분류하기 위해 사용해 왔다. 우리가 이 데이터를 얻기 위해 참가한 대회에서도 많은 상위권 팀들이 이미지 분류 과정에서 EfficientNet을 사용하였다. 또한 우리는 Kaggle notebook을 이용하여 대회를 진

행하였기 때문에 비교적 가벼운 모형으로 시험해 보는 것이 중요하였기에, 모형의 효율성이 높다는 점도 이 모형을 선택하는데 영향을 주었다. 이 모형으로 실험해본 결과, 원본 데이터는 0.659, 확장된 데이터는 0.553의 F1-score를 얻을 수 있었다.

3.2.2 EfficientNet + ArcFace

Discriminative power를 강화하는 적절한 loss function로 최근 제안된 내용이 Arcface이다. Deng(2018)등은 Softmax와 ArcFace loss의 차이를 보여주기 위해서, 1개의 class당 1500개의 이미지가 있고 8개의 class로 구성되어 있는 Face image에 대해서 2D Feature embedding network를 Softmax loss와 ArcFace loss를 각각 학습시켰다.

[그림 5] Softmax와 Arcface의 Feature embedding



[그림 5]에서 각 점들은 샘플을 나타내고, 선은 각 Identity의 중심 방향을 나타낸다. Feature normalization을 기반으로, 모든 Face feature들은 고정된 반지름을

가지는 호 공간에 존재한다.

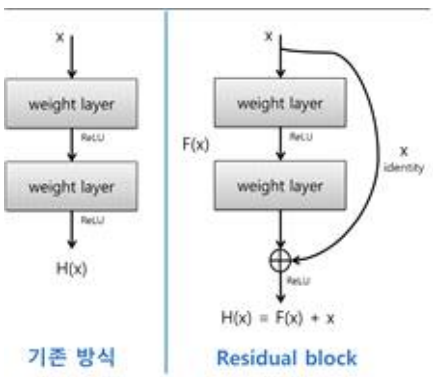
이 때, (a) Softmax loss는 대략 분리될 수 있는 Feature embedding을 만들지만, Decision boundary에서는 명확하게 구분되지 않는 반면, 제안된 (b) ArcFace loss에서는 가장 가까운 클래스 간의 더욱 명백한 Gap이 생기는 것을 확인할 수 있다.

우리가 시도하고자 하는 이미지 학습에서 Loss function의 성능을 높일 수 있다면 예측 정확도가 높아질 것이라 예상했고, 우리가 선택한 모형 중 EfficientNet을 선택하여 ArcFace를 적용하여 성능을 확인해 보았고, F1 Score 기준으로 0.055에서 0.078까지 성능이 높아지는 것을 확인할 수 있었다.

3.2.3 ResNet

일반적으로 이미지 분류에 쓰이는 CNN 모델의 경우 Layer가 깊어질수록 성능이 좋아지는 것으로 알려져 있다. 그런데 Layer가 깊어짐에 따라 성능이 떨어지는 현상이 발생하기도 하는데, 이는 역전파(Back-propagation) 과정에서 Gradient가 중간에서 점점 0에 수렴되어 학습이 되지 않는 VGP(Vanishing Gradient Problem) 문제 때문이다.

[그림 6] 기존 방식과 Residual block

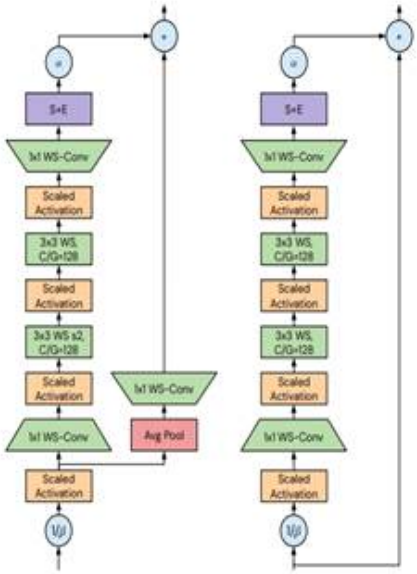


이를 개선하고자 기존 방식과 달리 Layer가 깊어져도 Gradient가 잘 이어질 수 있도록 <그림 6>과 같이 지름길(Shortcut)을 만들어주는 아이디어가 제안되었다. 이때, 이 지름길을 Residual Block이라 부르며, 이로 인해 모형의 이름이 ResNet이라고 불리게 되었다(He 등, 2015). ResNet 모형을 통해 모형을 깊게 쌓으면 모형이 잘 수렴하지 않는 문제를 극복할 수 있게 되었고, ImageNet 데이터 대회 등을 통해 모형을 깊게 쌓은 ResNet 모형들의 성능이 우수하여 이미지 분류 문제에 ResNet을 사용해 보았다.

3.2.4 Normalizer-Free Net

일반적인 이미지 분류 모델은 배치 정규화를 사용한다. 하지만 이는 기울기(Gradient) 평가 시간이 오래 걸리고, 배치 사이즈에 영향을 크게 받고, 분산 학습에서 구현 에러의 주된 원인이 된다는 단점이 있다. 그래서 배치 정규화를 진행하지 않는 대신 pre-activation을 진행하는 방식을 사용하는 Normalizer Free Network 모델을 사용하였다.

[그림 7] NfNet transition, non-transition block 세부 프로세스



이 모형의 구조는 [그림 7] 과 같다. 이는 마치 기존의 ResNet 또는 EfficientNet 모형에서 각 Batch 정규화를 제거한 모형과 같은 형태와 유사하다. 이 알고리즘 그대로 사용해 모형을 구현하게 되면 기존의 Batch Normalization의 장점인 Residual Branch의 스케일감소, Gradient exploding problem 감소 등 이러한 기능이 없어지면서 성능이 떨어지게 되는데, 이를 보완하기 위한 핵심 아이디어로써 Adaptive Gradient Clipping(AGC)를 Input data 이전의 처리에 사용한다.

기존 모형 학습을 안정화 하는 방법으로 Gradient Clipping방법이 많이 사용되고 있는데, 이는 기울기가 특정 임계값을 초과하지 않도록 해 모형 학습을 안정화 하는 방법이다. 이것을 이용하면 더 커다란 Learning Rate에서도 안정적으로 학습이 가능해 Gradient exploding problem 감소 역할을 할 수 있게 된다(Brock 등, 2021).

이 모형은 이미지 Classification을 위해 제안된 모형이므로 현재 발표된 이미지를 분류하는 방법 중 가장 최신 모형인 만큼 유용하다고 여겨졌다. 이 연구에서는 epoch를 15회 진행하며 Loss가 가장 낮았던 epoch=15의 학습된 Normalizer-Free Net을 이용해 예측을 진행했다.

4. 실험결과

우리 팀에서 적용해 본 BERT, EfficientNet, EfficientNet+ ArcFace, ResNet, NfNet 등 5가지 모형에 대한 F1 score를 산출해 보았다. Original Data만을 활용했을 때와 Original Data에 Augmentation Data를 추가하여 학습시켜 결과를 각각 산출하였고, 결과는 다음과 같다.

[표 2] 모형 별 F1 score 결과

Model	Original Data	Original Data+ Augmented data
BERT	0.466	-
EfficientNet	0.659	0.553
EfficientNet+ ArcFace	0.735	0.608
ResNet	0.712	0.621
NfNet	0.876	0.577

우리가 고려한 모형 전체적으로 데이터를 확장시켜 학습시킨 모형보다 원본 데이터만을 활용한 모형의 F1 score가 더 높은 다소 의외의 결과를 보였다. 이에 따라 원본 데이터를 통해 학습한 NFNet 모형의 F1 score가 0.876으로 가장 높게 나타나 본 논문에서 고려한 모형 중 베스트로 선정하며 마무리하게 되었다.

5. 결론

딥러닝에 대한 여러 연구들을 접해보며 모형을 직접 학습시키고 검증해보는 등의 시도를 해볼 수 있었다. 본 시도에서 NfNet이 우리가 적용한 모델 중 가장 성능이 좋다는 것을 확인할 수 있었다.

이번 연구를 계기로 딥러닝에 대한 여러 연구들을 접해보며 모형을 직접 학습시키고 검증해보는 등 다양한 시도를 해볼 수 있었다. 이러한 도전의 결과로 대회의 최종 목적인 더 싸고 다양한 제품들을 탐색할 수 있도록 조금이나마 도움이 될 수 있었고, 더 나아가 우리가 학습시킨 모형에 대해 유사한 목적을 가진 이미지 분류 시 참고할 수 있는 Transfer learning의 기반이 될 수 있는 소기의 성과를 달성하게 되었다.

더 많은 데이터를 학습시키고자 확장된 데이터를 사용하였음에도 모든 모형에서 성능이 나아지지 않은 점에 대해서 정확한 원인 파악이 필요하다. Data Augmentation을 적용한 목적을 달성하지 못했던 점이 한계로 남았다.

모형 학습 과정에서 Ensemble, Boost 기법 등을 활용한 결합 모형에 대한 시도나 ArcFace를 EfficientNet 외 모형의 Loss Function으로 적용하여 학습을 시켰을 때의 결과를 시도한다면 더 좋은 분류기가 형성돼 앞으로의 이미지 분류에서 더욱 정확한 결과를 도출해 낼 것임을 의심치 않는다.

참고문헌

[1] Tracz, J., Wojcik, P., Jasinska-Kobus, K., Belluzzo, R., Mroczkowski, R. and Gawlik, I. (2020). BERT-based similarity learning for product matching, *Proceedings of the Workshop on Natural Language Processing in E-Commerce (EComNLP)*, pages 66-75, Barcelona, Spain.

[2] Devlin, J., Chang, M. W., Lee, K. and Toutanova K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171-4186

[3] Peeters, R., Bizer, C. and Glavaš, G. (2020). Intermediate Training of BERT for Product Matching, *Proceedings of the 2nd International Workshop on Challenges and Experiences from Data Integration to Knowledge Graphs co-located with 46th International Conference on Very Large Data Bases (DI2KG 2020)*

[4] Tan, M. and Le, Q. V. (2019), EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, *International Conference on Machine Learning (ICML)*

[5] He, K., Zhang, X., Ren, S. and Sun, J. (2015), Deep Residual Learning for Image Recognition, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*

[6] Deng, J., Guo, J., Xue, N. and Zafeiriou, S. (2018), ArcFace: Additive Angular Margin Loss for Deep Face Recognition, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*

[7] Brock, A., De, S., Smith, S. L. and Simonyan, K. (2021). High-performance Large-Scale Image Recognition Without Normalization. *9th International Conference on Learning Representations (ICLR)*

논문투고 안내

"데이터과학연구"에 게재할 논문을 연중 접수하고 있습니다. 게재를 원하시는 분은 다음 사항들을 참조하기 바랍니다.

1. 논문분야: 경영, 사회, 교육, 공학, 의·약학 등의 학문 분야에서 데이터 조사 및 분석을 활용한 논문을 폭넓게 게재한다.
2. 논문부수 및 접수방법: 20매 내외의 분량을 연구소 E-mail을 통하여 접수한다. (E-mail : data@cau.ac.kr)
3. 심사 및 수정: 본 논문집에 투고한 논문은 편집위원회의 심사를 거쳐 수정 및 보완과 게재 여부를 결정한다.
4. 발간: 본 논문집은 연간 2회(6월, 12월) 발행을 원칙으로 한다.
5. 논문심사 진행사항 문의: 연구소 조교 (전화 02-820-6352)

데이터과학연구 제10권 2021

2021년 9월 29일 인쇄
2021년 9월 30일 발행

발행인 **임 예 지**
편집인 **곽 일 엽**

발행처 **중앙대학교 데이터과학연구소**
서울특별시 동작구 흑석동 221
중앙대학교 경영경제대학 응용통계학과
전화 (02) 820-5499 팩스 (02) 814-5498
E-mail : data@cau.ac.kr
Homepage : http://rcds.cau.ac.kr
인 쇄 주식회사 다컴애드

15
9 772765 548004
ISSN 2765-5482