

CNN 모델 성능 개선을 통한 무인매장 상품 재고 파악 모델 연구 Report

- 희망지도교수 : 박운상 교수님
- 학 과 : 데이터사이언스 전공
- 학 번 : A64056
- 성 명 : 이보성

1. 주제

- CNN 모델 성능 개선을 통한 무인매장 상품 재고 파악 모델 연구

2. 어떤 데이터를 사용할 것인지?

- > Kaggle, 구글, AI HUB 에서 확보한 데이터는 2가지이다.
- > 제품 진열장에 제품이 있는 132장의 이미지, 제품이 없이 비어 있는 42장의 이미지 한 셋
- > AI HUB 에서 제공하는 2만장 정도의 제품 이미지 한 셋

3. 어떻게 이용할 것인지? 그렇게 판단한 근거는?

- > 200장의 이미지 데이터, 해당 데이터로는 overfitting 의 위험성이 높음, Keras에서 제공하는 ImageDataGenerator 및 K-Fold Cross Validation 알고리즘을 이용해 데이터의 수를 늘려주고, deep-learning model layer 를 최대한 단순히 구성하는 별도의 모델 연구
- > 2만장의 이미지 데이터로 CNN기반의 ResNet 사용, 학습시킨 데이터를 기반으로, 매대에 해당 상품이 존재하고 있는 확률을 판단

4. 모델은 어떤 것으로 할지?

- > 베이스가 되는 모델은 ResNet 모델 사용해서 데이터 분석 및 이미지 분류 확률을 계산. 단, AWS 에서 2019년 발표한

‘ Bag of Tricks for Image Classification with Convolutional Neural Networks’ – AWS, 2019

논문을 토대로 ResNet 모델의 이미지 분류 정확도 및 데이터 분석 처리속도 향상을 목표로 함

5. 결과는 어떤 것으로 할지?

- > 학습된 모델의 정확성과 처리속도의 성능이 개선되었음을 확인하고, 모델 layer를 깊게 설계하거나, 구조를 변경하지 않더라도, 학습하고자 하는 모델의 batch size, learning rate schedule 등을 조정하는 방법으로도 좋은 성능개선의 효과를 나타낼 수 있는 것을 결과를 목표로 하고자 함.

6. 그래서 결론은 어떻게 낼지.

사용한 데이터를 통해 기존 CNN 모델보다 성능이 개선 되어진 모델을 제안 하는 것과, 코로나 언택트 시대를 지나옴에 따라 소비자의 비대면 서비스에 대한 인식의 변화, 매장의 smart 화에 따른 편의성 증대로 인한 무인 매장의 증가에 발맞추어 무인 매장의 완전 자동화에 Computer Vision 을 접목시킨 AI 모델을 제안하는 것을 결론으로 제안하고자 함

7. 논문리뷰

1). ' Bag of Tricks for Image Classification with Convolutional Neural Networks' – AWS, 2019 Paper 리뷰

이 논문에서 이를 대신하여 모델 구조를 변경하지 않고도 정확도를 개선 할 수 있는, 이른바 CNN 학습 Trick 에 대해 정보를 제공하고 있습니다. 이는 논문을 준비하는 저에게 가이드라인으로서 많은 도움이 될 것으로 예상됩니다.

해당 논문의 초록은 아래와 같습니다.

'이미지 분류 연구는 data 확장이나 최적화 방법의 변화 등 학습 방법을 개선하는 것으로도 좋은 성능을 얻을 수 있다.

하지만, 대부분의 이러한 개선점들은 구현부에 간략하게 언급되거나, 소스코드에서만 확인 가능한 경우가 있는데, 이 논문에서는 앞서 말한 이러한 개선점들에 대해 조사하고, ablation study (절제술 - 의학계에서는 절제술로 불리며, 딥러닝에서는 모델의 building blocks를 제거했을 때 전체 성능에 미치는 효과를 알기 위한 과학적 실험을 의미한다)를 통해 이들이 모델의 정확도에 미치는 영향을 경험적으로 평가할 것이다.'

그리고 실제로 이러한 개선점들을 결합했을 때 다양한 CNN 모델에서 성능이 개선되는 것을 확인할 수 있었다고 한다. 또한, 이미지 분류에서의 정확도 개선은 다른 도메인인 Object detection 과 Semantic segmentation에서도 좋은 결과로 이어졌다고 합니다.

Model	FLOPs	top-1	top-5
ResNet-50 [9]	3.9 G	75.3	92.2
ResNeXt-50 [27]	4.2 G	77.8	-
SE-ResNet-50 [12]	3.9 G	76.71	93.38
SE-ResNeXt-50 [12]	4.3 G	78.90	94.51
DenseNet-201 [13]	4.3 G	77.42	93.66
ResNet-50 + tricks (ours)	4.3 G	79.29	94.63

[표1.] 각 모델간 trick 적용여부의 의 연산량과 validation 정확도

해당 trick을 사용했을 경우 ResNet-50 모델에서는 성능이 약 4%가량 상승 한 것을 확인 할 수 있습니다.

적용 된 Trick 들의 간략한 내용은 아래와 같습니다.

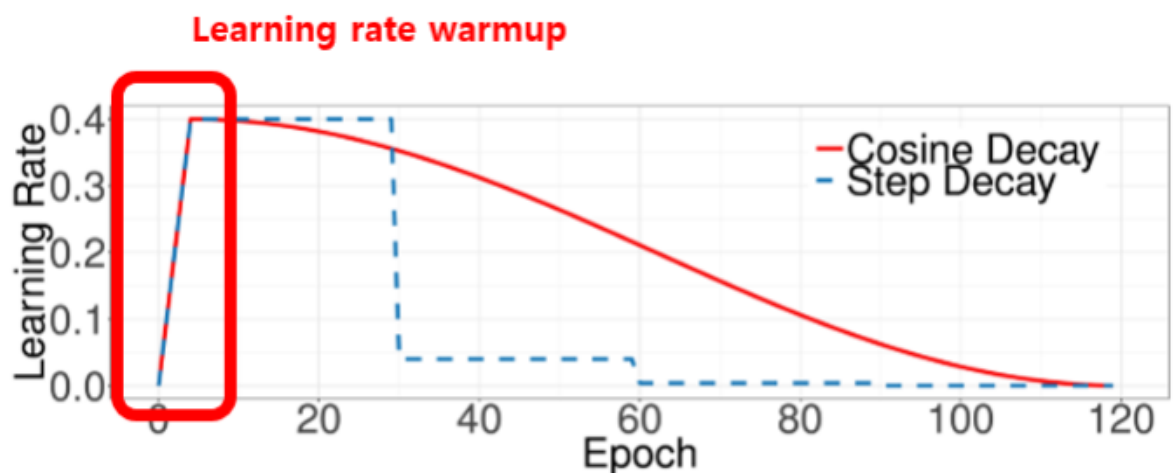
1. Large-batch training

(1) Linear scaling learning rate

논문에서는 batch size가 256일 때, 초기 learning rate를 0.1로 설정하고, batch size를 변경하면 초기 learning rate를 $0.1 \times b/256$ 으로 설정하라고 추천합니다.

(2) Learning rate warm up

학습을 시작할 때, learning rate를 작게 설정하고 학습이 안정되었을 때, 초기의 learning rate로 설정하는 것입니다. learning rate를 초기의 learning rate로 높일 때는 한 번에 높이는 것이 아니라 여러 단계에 걸쳐서 높이는 것을 추천합니다.



(a) Learning Rate Schedule

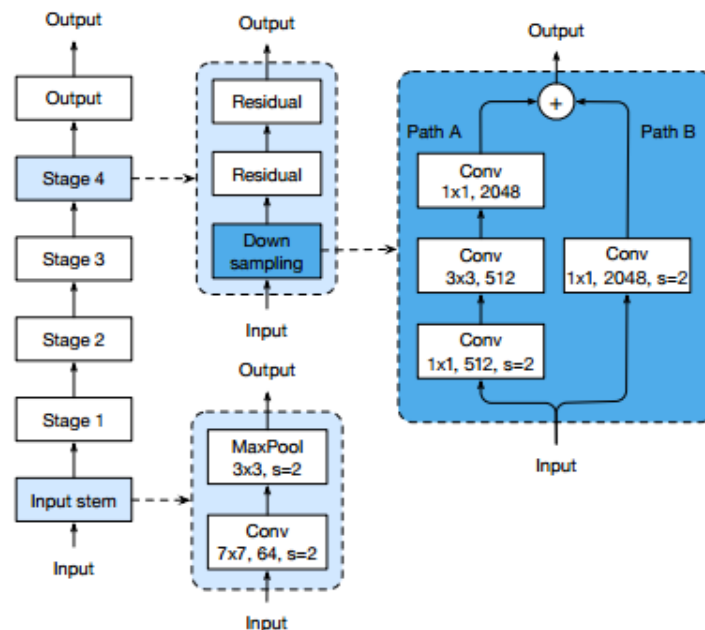
[그림1.] Learning Rate Schedule 조정

(3) Learning rate warm up

residual block의 끝에 위치해있는 모든 Batch Normalization 감마값을 0으로 initialize 합니다. 이렇게 하면 모든 residual block은 입력값을 그대로 반환하게 되어 초기 학습에 도움이 된다고 하네요.

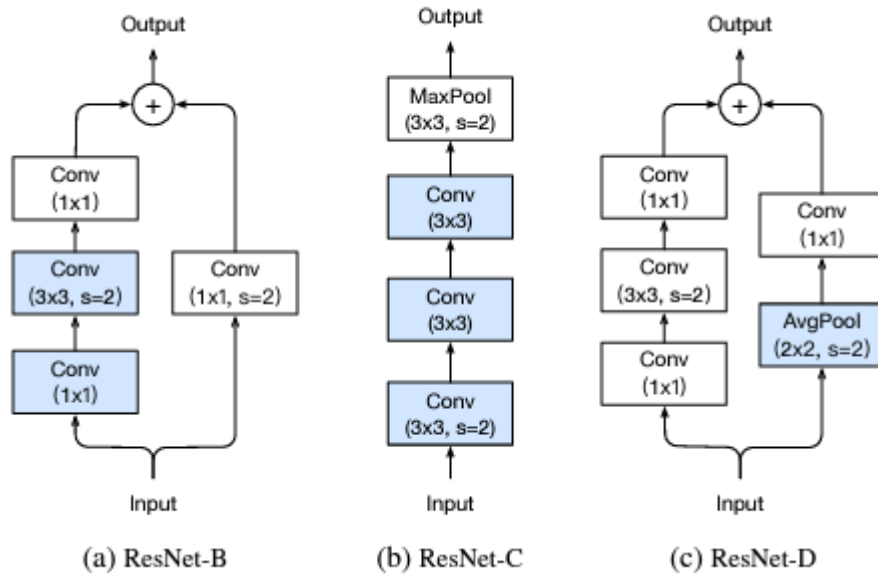
2. Model Tweaks 기법

ResNet의 residual block을 수정하여 성능을 1% 향상시킵니다. 기존 ResNet 구조는 아래와 같습니다.



[그림2.] ResNet 기본 구조

resnet의 초기 레이어에 7x7 conv가 사용하는 것을 확인할 수 있습니다. 이것을 3x3 conv 3개로 대체합니다. 또한, down sample을 진행할 때, residual block과 shortcut connection에서 stride=2인 1x1 conv를 사용합니다. 이것을 residual block에서는 3x3 conv에서 stride=2를 사용하고, shortcut connection에서는 avg pool을 사용하여 다운 샘플링을 진행합니다.



[그림3.] Three ResNet Tweaks 적용

ResNet-B Model 은 down sampling 블록 수정 적용, ResNet-C 모델은 Input stem 추가로 수정, ResNet-D 는 다시 down sampling block 을 수정한 모델입니다.

해당 모델들의 정확도는 아래와 같이 향상 되었음을 확인 할 수 있었습니다.

Model	#params	FLOPs	Top-1	Top-5
ResNet-50	25 M	3.8 G	76.21	92.97
ResNet-50-B	25 M	4.1 G	76.66	93.28
ResNet-50-C	25 M	4.3 G	76.87	93.48
ResNet-50-D	25 M	4.3 G	77.16	93.52

[표2.] ResNet Tweaks 적용 후 정확도

그 외에 몇가지의 정확도 및 처리속도 향상을 위한 기법(Trick) 들이 소개되어 있습니다.

해당 기법들 외에 추가적으로 적용할 수 있는 방향을 고민하고 연구하며 처리속도와 정확도가 향상된 CNN 모델을 연구하는 것을 목표로 하려 합니다.

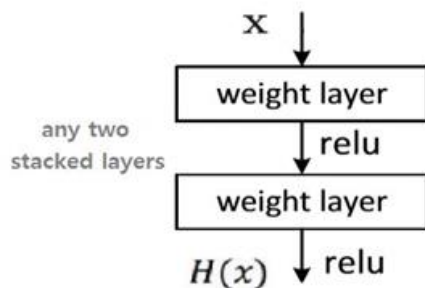
2) 'CNN 딥러닝 모델을 활용한HS품목 이미지 분류' – 이동주 , 한밭대학교 창업경영대학원

본 논문에서는 HS 품목 이미지 분류를 위해 ResNet 모델 외에 VGG-16, VGG-19, Inception V3 모델을 사용하였다. VGG-16 모델이 가장 정확도가 높게 나왔지만, 제가 사용하고자 하는 ResNet 모델의 기본 구조에 대해 리뷰하고자 한다.

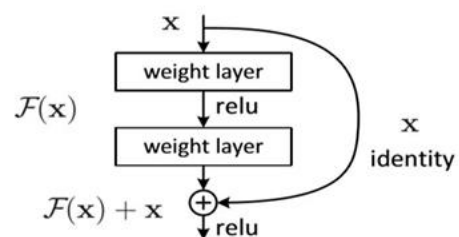
(1) ResNet

ResNet은 2015년 ILSVRC(ImageNet Large-Scale Visual Recognition Challenge) 대회에서 우승하였고 Top5 Error가 3.6%로 사람이 분류하는 수준인 5%를 넘어선 모델이다

레이어가 깊어질수록 Gradient vanishing(기울기 소실) 또는 Exploding(기울기 폭발) 때문에 오히려 성능을 떨어트리는 원인이 되는데 이런 현상을 Degradation 문제라고 한다. 레이어가 깊은 상태에서 학습을 많이 진행한 경우 Weigth(가중치)의 분포가 균등하지 않고 역전파시 기울기가 충분치 않아 안정적인 학습 진행을 할 수 없는 문제가 발생된다. 이러한 Degradation 문제를 해결하기 위해 ResNet 저자들은 Residual block을 제시하였다. 이는 Gradient가 잘 흐를 수 있도록 하기 위한 일종의 Shortcut connection(지름길)을 만들어 주는 것과 같은 역할을 한다. 이는 ResNet 의 기본 구조가 되어진다.



[그림1.] CNN 신경망 구조



[그림2.] ResNet 신경망 구조

[그림1.]은 입력 x 를 받아서 2개의 weight layer를 통과해 출력 $H(x)$ 를 얻는 구조이다. 만약에 $H(x)$ 가 아닌 $H(x)-x$ 가 나올 수 있게 수정하여, 입출력 차이를 얻을 수 있도록 학습시킨다면 2개의 weight layer는 $H(x)-x$ 를 나오도록 학습시켜야 되며, $F(x)=H(x)-x$ 라고 한다면, 출력 $H(x)=F(x)+x$ 가 나오게 된다. 이것이 [그림2.]인 Residual learning: a building block이다. 입력이 들어오고 출력이 나오면 이 입출력을 합하는 기법이며, 이를 위해 입력과 출력 Dimension이 같아야만 한다. 중간 layer는 입출력 사이의 차를 학습하여 깊은 신경망도 보다 쉽게 최적화가 가능하며, 늘어난 만큼의 깊이로 인해 정확도가 개선된다.

본 논문에서 DataSet 은 총 3가지로 구성 후 각각의 CNN 모델에 학습시킨 후 결과를 도출했다.

각 모델은 사전 학습된 모델(Pre-trained Model)의 가중치를 이용하였다. 사전 학습된 모델들은 1,000개 클래스를 분류하는 대회인 ILSVRC(ImageNet Large Scale Visual Recognition Challenge)를 대상으로 만들어져서 출력층 유닛(Unit)이 1,000개가 된다.

이 사전 학습된 모델의 출력층이 1,000개이므로 새로운 HS품목 이미지 분류에 맞는 출력층으로 대체하였다. 사용한 가중치는 이미지넷(ImageNet)을 이용하고 사전 학습된 Convolution Layer 층은 그대로 두고 Fully Connected 부분을 재구성 하였다.

이후 DataSet 을 다른 구성으로 나누어서 테스트를 진행 하였으며, 그중 Train과 Test의 수를 동일하게 하여 각각 3,000개와 1,250개의 학습 데이터를 만들어 학습할 수 있게 하였다. Epochs는 100, 150회 적용하여 수행하였다. 수행결과 Epochs 150, Batch Size 32, Optimizer(LR) RMSprop(lr=1e-4), Dropout 0.5로 하여 수행하였을 때 VGG16 모델에서 73.12%의 정확도가 많이 향상되었다. 또한 전반적으로 모든 모델에서 정확도가 많이 향상되었다.

Epochs	VGG16	VGG19	ResNet50	Inception-V3
100	68.12%	65.00%	71.88%	41.25%
150	73.12%	71.25%	64.38%	45.00%

[표1.] 모델별 실험 정확도 결과

해당 표에서 볼 수 있듯이, Epochs 100회 기준으로는 ResNet50 모델이 가장 좋은 성능을 나타내었으며, 150회 기준에서는 VGG-16 모델이 가장 좋은 성능을 나타내는 것으로 볼 수 있다.

해당 논문에서 사용되어진 데이터셋은 해외수출입시 세액 부과를 위한 HS부류 코드를 데이터셋으로 사용하였다. 해당 데이터셋은 부호로 이루어져 있어 데이터의 양이 많지 않음에도 특정한 모양과, 패턴을 가지고 있기에 부족한 데이터셋으로도 어느 정도 보장된 성능을 나타낼 수 있음을 확인 할 수 있었다.