

Digital
Image
Process
Report #11



Deep Learning 실습

로봇기계공학과
21721160
최병희

Deep Learning 실습

- 문제 - 주어진 CNN 학습 및 R-CNN 코드 실습

이번 과제의 목적은 주어진 코드를 통해 Matlab으로 CNN 학습 과정과 Object Detection의 일종인 R-CNN의 detection 결과를 확인하는 것입니다. 코드는 1) CNN 학습 결과 확인, 2) R-CNN을 사용한 Object Detection 결과 확인의 순서로 되어 있습니다.

- 원리 설명

CNN은 학습 데이터에 대해서 필터로 컨벌루션 연산을 진행합니다. 레이어를 거치며 출력된 feature map은 사용한 필터에 대해서 강조된 결과를 갖게 되며, 여러 레이어를 거칠수록 pooling으로 더욱 필터가 강조된 결과를 가지며, relu 함수를 통해 비선형성을 추가합니다. 이렇게 산출된 결과는 필터 형태의 가중치인데, 이 가중치는 역전파와 경사 하강법을 통해 보정되어 다시 순전파 연산을 재개합니다. 이러한 과정을 여러 번 거치고 나면 물체를 인식하는 필터 형태의 가중치가 학습 결과로서 나오게 됩니다.

CNN을 활용한 Object Detection의 성공적인 초기 모델인 R-CNN은 region-proposal을 사용하여 먼저 물체끼리의 경계를 분류한 후 그에 대해 CNN으로 학습을 진행하는 방식입니다. 물체의 구분은 강의에 나왔던 blob과 같은 원리로 구분됩니다. 이 R-CNN은 이후 Faster-R-CNN, Masked-R-CNN 등 많은 형태의 CNN 형태로 발전합니다.

- Source code

결과만을 보는 실습이기에 주어진 코드를 그대로 실행하였습니다.

- 결과 비교 및 배운 점

CNN 학습은 주어진 cifar10 데이터셋을 사용하여 10개 class의 분류를 학습합니다. 코드에서 첫 번째로 출력한 이미지는 학습 데이터 중 100개를 추출한 결과입니다. 이와 같은 이미지를 50000장 학습합니다. 하지만 이번 실습에는 학습하기에 오랜 시간이 걸리므로, 학습 결과가 저장된 파일을 제공합니다. 학습 결과 파일에는 각 레이어마다의 결과가 저장되어 있기 때문에, 첫 번째 컨벌루션 레이어에서 사용한 필터의 형태를 불러왔습니다. 이 코드에서 사용되는 필터는 5x5 필터로, 이 필터를 backpropagation과 경사 하강법을 통해 수정함으로써 가중치를 얻습니다. 코드에서 출력한 이미지는 이 필터들의 원형을 나타낸 것입니다. cifar10에서 주어진 test data를 통해 평균 정확도를 측정합니다. 나온 accuracy 결과는 다음과 같습니다.



두 번째로 R-CNN 결과를 확인합니다. 코드에는 학습 레이어의 형태는 나와있지 않지만, R-CNN은 유명한 레이어이기 때문에 Deep Learning Toolbox에서 레이어를 제공하여 `trainRCNNObjectDetector` 함수를 통해 사용할 수 있습니다. R-CNN은 먼저 학습된 가중치를 통해 1차적으로 물체가 있을 만한 구역을 구분한 후 그에 대한 bounding box 학습을 진행합니다. 물체를 인식하여 분류하는 것만이 아닌 물체의 위치까지 찾아내야 하므로 단순히 분류만 진행하는 CNN보다 조금 더 복잡한 구조를 갖고 있습니다. 실습 코드의 세 번째, 네 번째 사진은 R-CNN의 검출 결과입니다.





초기 모델인 R-CNN을 사용한 결과가 생각보다 잘 나와서 놀랐습니다. yolov3, yolov5를 사용해 보았지만 학습시킨 것에 비해 성능이 조금 못 미친다는 느낌이 있었는데, test data에 대해서 confidence값이 0.99정도 나와서 R-CNN도 사용해볼 만하다는 생각이 들었습니다. 하지만 속도 측면에서 빠르지 느린지는 사진 몇 장으로 알 수 없어, 어느 정도 정량적인 비교가 필요할 듯 합니다.