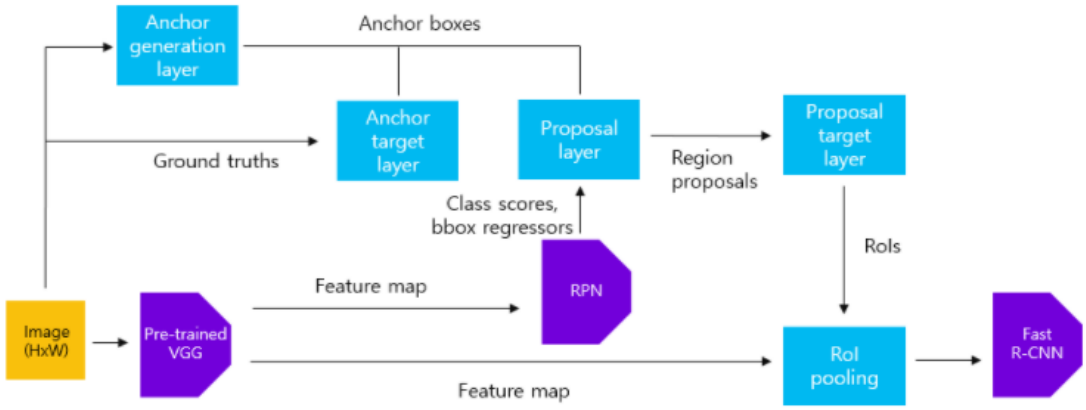
Faseter R-CNN

All about RPN : Train



Training Faster R-CNN

Anchor target layer, Proposal target layer → 학습을 위해 필요한 ground truth를 만들어주는 layer들

1) feature extraction by pre-trained VGG16

pre-trained된 VGG16 모델에 800×800×3 크기의 원본 이미지를 입력하여 50×50×512 크기의 feature map을 얻습니다. 여기서 sub-sampling ratio는 1/16입니다.

- **Input** : 800×800×3 sized image
- **Process** : feature extraction by pre-trained VGG16
- **Output** : 50×50×512 sized feature map

2) Generate Anchors by Anchor generation layer

region proposals를 추출하기에 앞서 **원본 이미지에 대하여 anchor box를 생성**하는 과정이 필요합니다. 원본 이미지의 크기에 sub-sampling ratio를 곱한만큼의 grid cell이 생성되며, 이를 기준으로 각 grid cell마다 9개의 anchor box를 생성합니다. 즉, 원본 이미지에

50×50(=800×1/16 x 800×1/16)개의 grid cell이 생성되고, 각 grid cell마다 9개의 anchor box를 생성하므로 총 22500(=50×50×9)개의 anchor box가 생성됩니다.

- **Input** : 800×800×3 sized image
- **Process** : generate anchors
- **Output** : 22500(=50×50×9) anchor boxes

3) Class scores and Bounding box regressor by RPN

RPN은 VGG16으로부터 feature map을 입력 받아 anchor에 대한 **class score, bounding box regressor를 반환**하는 역할을 합니다.

- **Input** : 50×50×512 sized feature map
- **Process** : Region proposal by RPN
- **Output** : class scores(50×50×2×9 sized feature map) and bounding box regressors(50×50×4×9 sized feature map)

4) Region proposal by Proposal layer

Proposal layer에서는 2)번 과정에서 생성된 anchor boxes와 RPN에서 반환한 class scores와 bounding box regressor를 사용하여 **region proposals를 추출**하는 작업을 수행합니다. 먼저 Non maximum suppression을 적용하여 부적절한 객체를 제거한 후, class score 상위 N개의 anchor box를 추출합니다. 이후 regression coefficients를 anchor box에 적용하여 anchor box가 객체의 위치를 더 잘 detect하도록 조정합니다.

- **Input**
  - 22500(=50×50×9) anchor boxes
  - class scores(50×50×2×9 sized feature map) and bounding box regressors(50×50×4×9 sized feature map)
- **Process** : region proposal by proposal layer
- **Output** : top-N ranked region proposals

5) Select anchors for training RPN by Anchor target layer

Anchor target layer의 목표는 **RPN이 학습하는데 사용할 수 있는 anchor를 선택**하는 것입니다. 먼저 2)번 과정에서 생성한 anchor box 중에서 원본 이미지의 경계를 벗어나지 않는 anchor box를 선택합니다. 그 다음 positive/negative 데이터를 sampling해줍니다.

여기 positive sample은 객체가 존재하는 foreground, negative sample은 객체가 존재하지 않는 background를 의미합니다.

전체 anchor box 중에서 1) ground truth box와 가장 큰 IoU 값을 가지는 경우 2) ground truth box와의 IoU 값이 0.7 이상인 경우에 해당하는 box를 positive sample로 선정합니다. 반면 ground truth box와의 IoU 값이 0.3 이하인 경우에는 negative sample로 선정합니다. IoU 값이 0.3~0.7인 anchor box는 무시합니다. 이러한 과정을 통해 RPN을 학습시키는데 사용할 데이터셋을 구성하게 됩니다.

- **Input** : anchor boxes, ground truth boxes
- **Process** : select anchors for training RPN
- **Output** : positive/negative samples with target regression coefficients

6) Select anchors for training Fast R-CNN by Proposal Target layer

Proposal target layer의 목표는 proposal layer에서 나온 region proposals 중에서 **Fast R-CNN 모델을 학습시키기 위한 유용한 sample을 선택**하는 것입니다. 여기서 선택된 region proposals는 1)번 과정을 통해 출력된 feature map에 RoI pooling을 수행하게 됩니다. 먼저 region proposals와 ground truth box와의 IoU를 계산하여 0.5 이상일 경우 positive, 0.1~0.5 사이일 경우 negative sample로 label됩니다.

- **Input** : top-N ranked region proposals, ground truth boxes
- **Process** : select region proposals for training Fast R-CNN
- **Output** : positive/negative samples with target regression coefficients

7) Max pooling by RoI pooling

원본 이미지를 VGG16 모델에 입력하여 얻은 feature map과 6) 과정을 통해 얻은 sample을 사용하여 RoI pooling을 수행합니다. 이를 통해 고정된 크기의 feature map이 출력 됩니다.

- **Input**
  - 50×50×512 sized feature map
  - positive/negative samples with target regression coefficients
- **Process** : RoI pooling
- **Output** : 7×7×512 sized feature map

8) Train Fast R-CNN by Multi-task loss

나머지 과정은 Fast R-CNN 모델의 동작 순서와 동일합니다. 입력 받은 feature map을 fc layer에 입력하여 4096 크기의 feature vector를 얻습니다. 이후 feature vector를 Classifier와 Bounding box regressor에 입력하여 (class의 수가 K 라고 할 때)각각 (K+1), (K+1) x 4 크기의 feature vector를 출력합니다. 출력 된 결과를 사용하여 Multi-task loss를 통해 Fast R-CNN 모델을 학습 시킵니다.

- **Input** : 7×7×512 sized feature map
- **Process**
  - feature extraction by fc layer
  - classification by Classifier
  - bounding box regression by Bounding box regressor
  - Train Fast R-CNN by Multi-task loss
- **Output** : loss(Loss loss + Smooth L1 loss)

How RPN and Detector share feature maps: 4-step alternating training

1. alternating training

RPN을 학습진행 → proposal을 사용하여 Fast R-CNN을 학습 → Fast R-CNN학습이 끝난 후 Bouding Box Regression을 통해 Fast R-CNN에 의해 조절된 Network를 RPN을 초기화 할 때 사용.

이 과정을 계속 반복

2. Approximate joint training

RPN과 Fast R-CNN을 트레이닝 할 때 하나의 네트워크로 합친다.

3. non-approximate joint training

RPN으로 예측된 bounding boxes를 input으로 함 이때, box의 좌표와 다른 RoI polling layer가 필요.

Implementation details: scales, hyper-parameters

Multi-scale feature extraction은 정확도를 향상 시키지만, 정확도 대비 속도가 그렇게 좋지 못함. 다양한 크기의 regions를 예측하는데 이미지 피라미드,필터 피라미드가 없기에 running time을 절약할 수 있음. 이미지는 single **scale**(짧은 쪽 :600 픽셀) scale 3개와 3개의 aspect **ratio**를 가지고 9개의 **anchor** 생성

이미지 경계에 존재하는 **anchor** 는 학습할 때 무시함. 일반적인 1000 * 600 이미지의 경우 대략 20000 (= 60 × 40 × 9) **anchor** 가 나옴. 그 중 이미지 경계에 존재하는 앵커를 무시하면, 이미지 당 6000개의 앵커가 있음. 하지만 test에서는 이미지 경계에 존재하는 이상치를 무시하지 않음.

NMS를 사용해서 전체 6000개의 앵커중 IOU > 0.7이상인 2000개의 proposals만 남김. 학습 시에는 2000개의 proposals을 학습하지만, test시에는 속도 및 정확도 테스트를 위해 top-N proposals를 사용하여 실험 진행.