End-to-End Object Detection with Transformers

Nicolas Carion , Francisco Massa , Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko (Facebook AI Team), 2020

2025-03-05

지능형솔루션개발팀 안시은

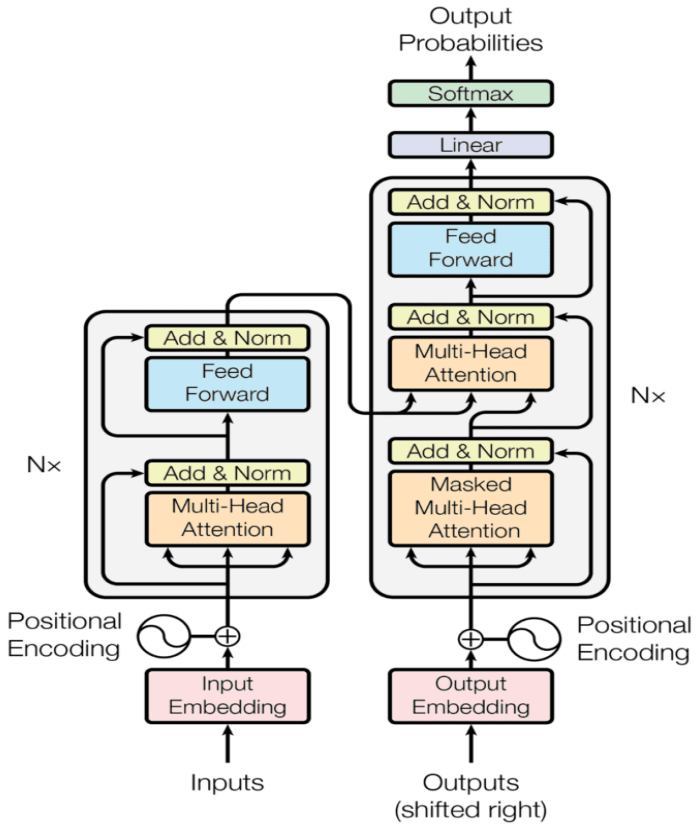
목차

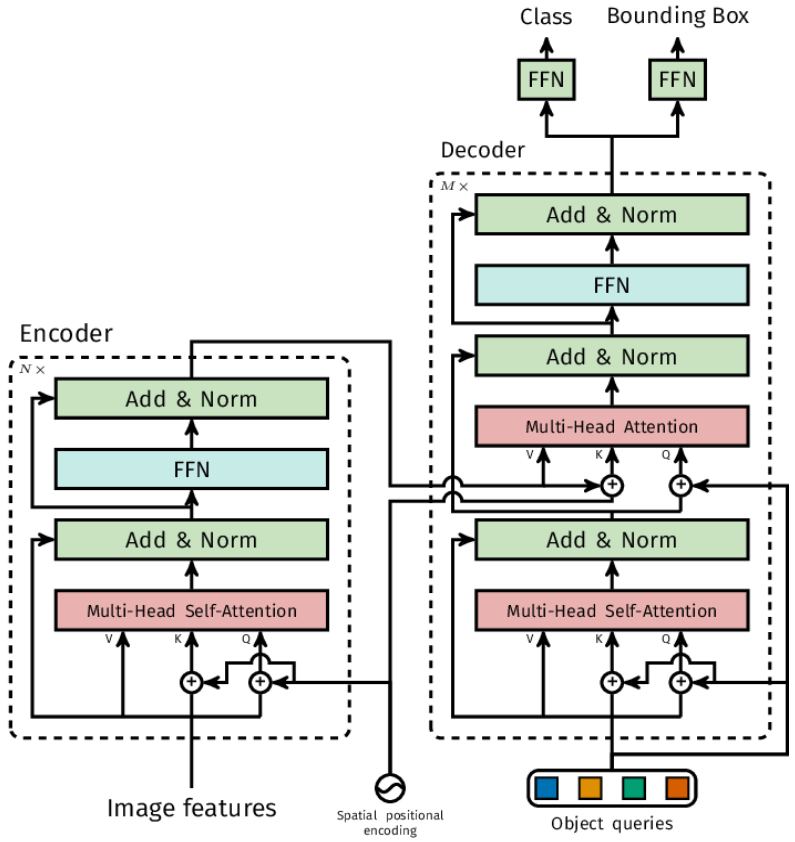
개요

- 기존 object detection 모델에서는 다수의 anchor 생성, NMS와 같은 후처리 과정이 무조건적으로 진행되고 있음.
- 위와 같은 후처리 과정을 사용할 필요 없이, 이진 매칭을 통해 중복 예측을 방지하는 transformer 기반의 end to end 모델 DETR을 제시하고자 함.

Transformer for NLP task vs DETR Transformer

- 주요 차이점:
  - Self attention 사용을 통한 객체 병렬적 처리
  - Object detection task를 위한 object query





Object Query란,

- == object query features + object query positional embedding
  - → 모두 learnable
  - Object query features(주체)
    - decoder 초기 입력 값으로 들어감(initially zero)
    - 고정된 N개 만큼 생성되고 학습 전에 초기화됨.(하이퍼 파라미터로서 설정 가능, 논문에서는 100개로 설정.)
    - bbox 정보를 담고 있음
    - 각 디코더 레이어를 지나면서 업데이트됨. (+ spatial positional encoding과 positional embedding의 위치 정보 도움을 받음)
    - 매 레이어마다 업데이트
  - Object query positional embedding
    - 학습 전에 초기화됨
    - 모델이 어떤 쿼리가 어떤 객체를 예측하는지 학습할 수 있도록 도움을 주는 위치 인코딩
    - 순전파에서는 업데이트X, 역전파가 끝나고 업데이트

Set Prediction Loss

$$\mathcal{L}_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \left[-\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\hat{\sigma}(i)}) \right]$$

모델이 예측한 클래스 확률,
객체가 없을 때에 대하여
1/10으로 probability 감소

bbox loss

$$\hat{\sigma} = \arg\min_{\sigma \in \mathfrak{S}_N} \sum_{i}^{N} \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)})$$

$$-\mathbb{1}_{\{c_i \neq \varnothing\}} \hat{p}_{\sigma(i)}(c_i) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\sigma(i)})$$

- N == object query의 지정된 예측 개수(논문에서는 100개로 set)
- c_i == ground truth class
- b_i == ground truth bbox

Set Prediction Loss

$$\mathcal{L}_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \left[-\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\hat{\sigma}}(i)) \right]$$

$$\lambda_{\text{iou}} \mathcal{L}_{\text{iou}}(b_i, \hat{b}_{\sigma(i)}) + \lambda_{\text{L1}} ||b_i - \hat{b}_{\sigma(i)}||_1$$

→ GIoU + L1 loss:
가장 일반적으로 사용되는 L1 loss는, 작은 박스와 큰 박스의 상대 오차가 비슷하더라도 서로 다른 크기의 값을 가짐.
이러한 문제를 완화하기 위해 L1 loss와 GIoU(generalized IoU loss)의 조합을 사용

사용된 데이터셋

- coco 2017 object detection dataset

학습 및 추론 비용

- 학습
  - 베이스 모델 학습 300 epochs on 16 V100 GPUs 3일간 진행(batch size 64, gpu당 4장씩 학습)
- 추론

Model	GFLOPS/FPS	#params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster RCNN-DC5	320/16	166M	39.0	60.5	42.3	21.4	43.5	52.5
Faster RCNN-FPN	180/26	42M	40.2	61.0	43.8	24.2	43.5	52.0
Faster RCNN-R101-FPN	246/20	60M	42.0	62.5	45.9	25.2	45.6	54.6
Faster RCNN-DC5+	320/16	166M	41.1	61.4	44.3	22.9	45.9	55.0
Faster RCNN-FPN+	180/26	42M	42.0	62.1	45.5	26.6	45.4	53.4
Faster RCNN-R101-FPN+	246/20	60M	44.0	63.9	**47.8**	**27.2**	48.1	56.0
DETR	86/28	41M	42.0	62.4	44.2	20.5	45.8	61.1
DETR-DC5	187/12	41M	43.3	63.1	45.9	22.5	47.3	61.1
DETR-R101	152/20	60M	43.5	63.8	46.4	21.9	48.0	61.8
DETR-DC5-R101	253/10	60M	**44.9**	**64.7**	47.7	23.7	**49.5**	**62.3**

학습 방법

- 메인 코드 및 모델 다운로드:
  - https://github.com/facebookresearch/detr

	name	backbone	schedule	inf_time	box AP	url	size
0	DETR	R50	500	0.036	42.0	model | logs	159Mb
1	DETR-DC5	R50	500	0.083	43.3	model | logs	159Mb
2	DETR	R101	500	0.050	43.5	model | logs	232Mb
3	DETR-DC5	R101	500	0.097	44.9	model | logs	232Mb

감사합니다.