Models for Computer vision project

Wednesday, 18 December 2024 15:25

Topics that we need to dig deep into:

DETR (Detection Transformer)

Object detection model - by FB AI

replaces traditional object detection methods like R-CNN, Yolo with transformer based approach.

R CNN:

Conditional DETR:

Improved version of DETR:

① faster convergence
② better detection accuracy.

Deformable DETR:

Used to handle high resolution images.

Uses deformable attention mechanism focusing only on the relevant part of the image

Especially for small or sparse object

DETA:

Detection transformer with Anchor

Deta modified DETR by re-introducing Anchor based queries into transformer Framework

Anchor based initialization of queries improves performance Faster convergence & higher accuracy

TATR:

A table structure recognition model that extends DETR-like transformers for tabular data extraction.

Pdf's & scanned forms → Document analysis.

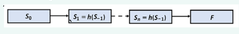
Model	Key Feature	Use Case	Strengths
DETR	End-to-end object detection with Transformers	Object detection and segmentation	Simplicity, no handcrafted components
Conditional DETR	Conditional queries for faster training	Faster training with DETR architecture	Speed and improved accuracy
Deformable DETR	Deformable attention for efficiency	High-resolution object detection	Handles small objects efficiently
DETA	Anchor-based query initialization	Faster convergence with Transformer	Best of both anchor-based and Transformer
Table Transformer	Extract table structures from images	Table recognition in scanned documents	Tailored for tabular data extraction
YOLOS	Transformer-only architecture	Purely Transformer-based object detection	Simplified architecture, experimental

Origin of Transformers:

1982, John Hopfield → RNN

RNN evolved to form LSTM

Each state S_n - captures the info of S_{n-1}



In 1980's Yann LeCun designed CNN (Convolutional Neural Network)

Late 2017, Transformers with attention head sublayers & more.

Attention layer manages the relationships between words in a sequence by performing pair-wise analysis.

Transformers have a quadratic time complexity O(n²) because they analyze all relationships between words at once, leveraging parallel processing, aiming to understand the entire "book"—or data sequence—more thoroughly and quickly.

A generative model can be summed up as $t = f(n)$

transformer works at token level (a piece of word)

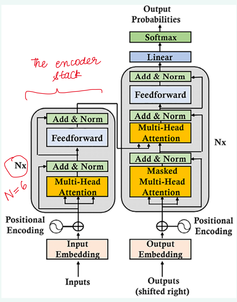
This makes the o/p dynamic based on the inputs

Features of RNN:

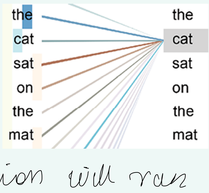
① process input in sequence one at a time
② The same weight across the network
③ Retains the memory of previous

right):

LSTM:



There is no recurrence network used here.



Attention will run dot product between the word & all other words, including itself.

Encoder mainly consists of

① Multi-headed Attention Mechanism
② Fully connected Feed forward network.