1. Giới thiệu

Object detection là một bài toán quan trọng trong Computer Vision. Trong post này tôi sẽ giới thiệu YOLOv2, một trong những phương pháp tốt nhất và nhanh nhất (real-time) hiện nay. Tôi viết 2 bài về YOLO, bài thứ nhất (bài này) sẽ giúp các bạn hiểu về nguyên lý hoạt động của YOLO, bài thứ 2 sẽ giới thiệu cách huấn luyện mô hình YOLO cho dữ liệu riêng

Bài báo gốc về YOLO có thể tìm ở đây version 1, version 2.

Code source, hướng dẫn chạy test và huấn luyện với dữ liệu ảnh Pascal Voc ở đây here.

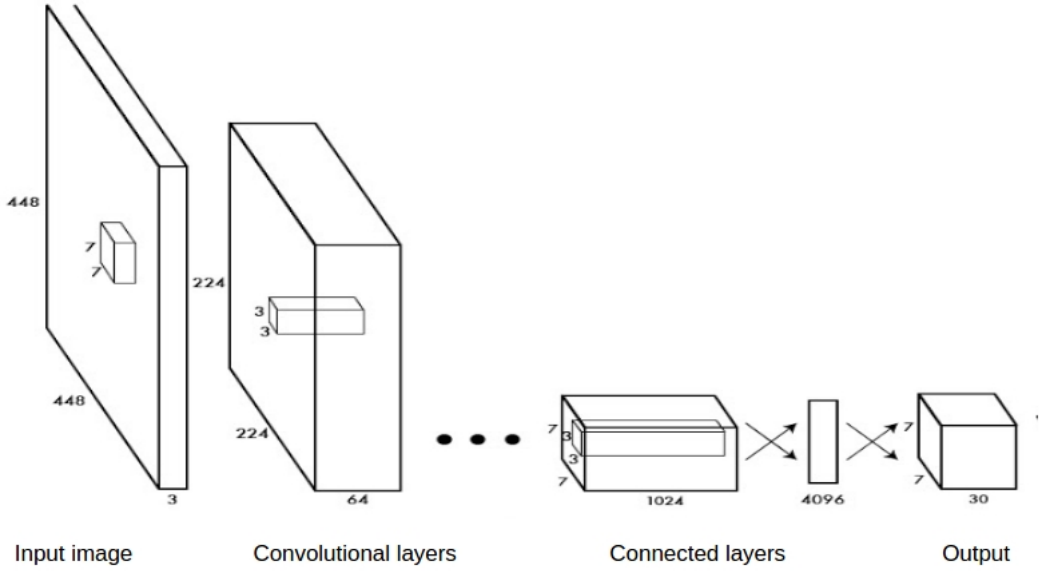
Before you continue, make sure to watch the awesome YOLOv2 trailer. https://www.youtube.com/watch?v=VOC3huqHrss

2. Dependencies

Để đọc bài này, cần kiến thức cơ bản về neural network

3. Nhận dạng

YOLO là một deep net kết hợp giữa convolutional layers và connected layers :



architecture

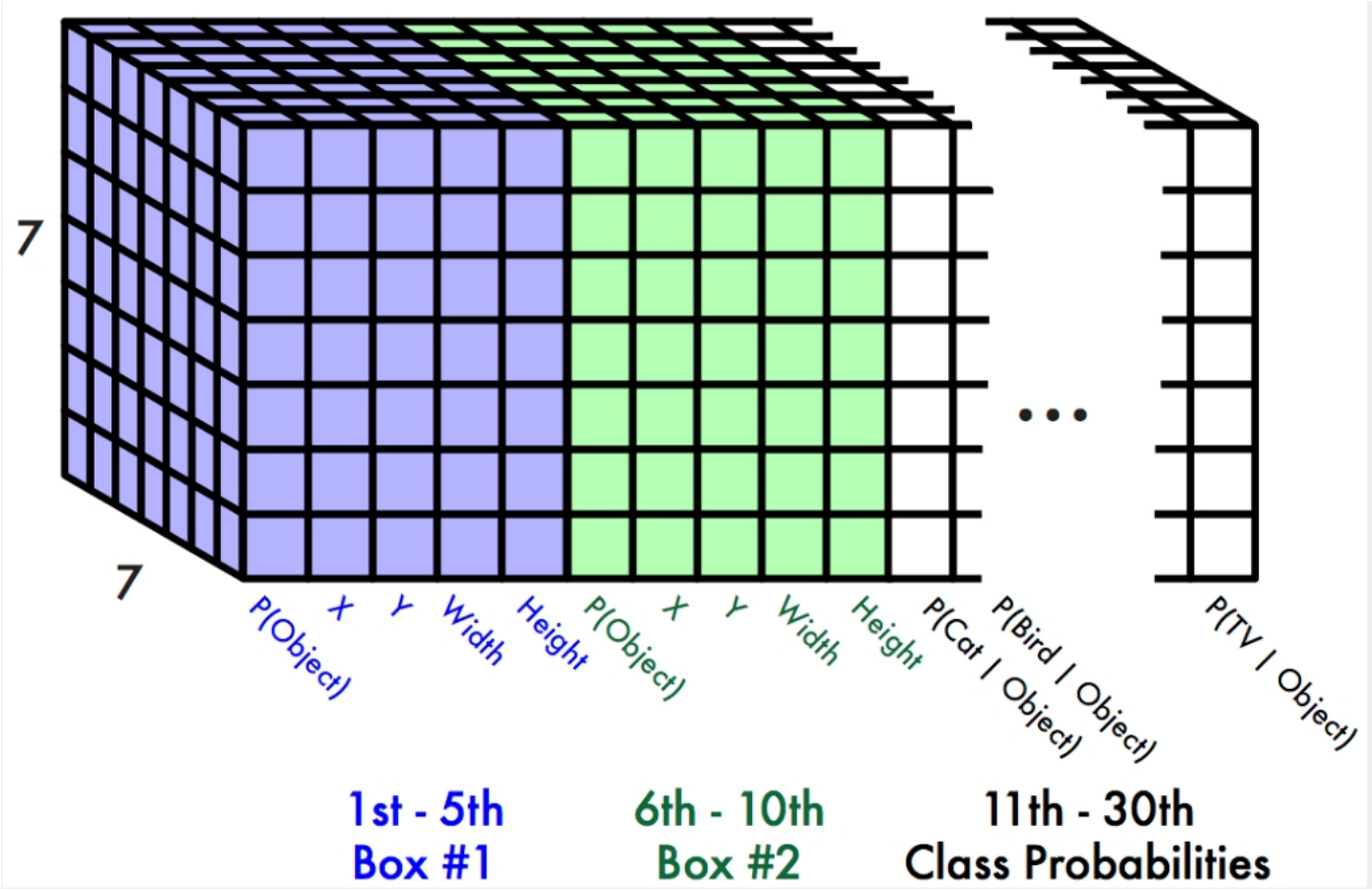
YOLO phân chia hình ảnh thành một mạng lưới 7x7 ô (grid_size=7x7):

grid

YOLO sẽ dự đoán xem trong mỗi ô xem liệu có object mà điểm trung tâm rơi vào ô đó. Và dự đoán điểm trung tâm, kích thước của object đó và xác xuất object đó là object nào trong các objects cần xác định.

Mỗi ô này có trách nhiệm dự đoán 2 hộp (boxes_number=2) bao quanh. Mỗi 1 hộp mô tả hình chữ nhật bao quanh một object.

Giả sử ta đang huấn luyên YOLO nhận dạng 20_objects khác nhau. Sau khi qua các layers, image input sẽ được biến đổi thành 1 tensor kích thước 7x7x30.



yolo_output

Ta có thể hiểu là mỗi ô sẽ có 30 tham số , tham số thứ nhất là xác xuất ô có chứa 1 object, tham số 2,3,4,5 lần lượt là x_center, y_center, width, height của box 1. Tương tự tham số 6,7,8,9,10 là của box 2. Tham số thứ 11 là xác xuất object trong ô là object1 (trong 20_objects cần nhận dạng). Tương tự tham số 12 là xác xuất object trong ô là object2 ... cho đến tham số 30 là xác xuất object trong ô là object20.

Ta có thể hiểu đơn giản như sau: 1. Image input được resize thành 1 image 448x448x3 (image_dimension = 448x448 với số channels = 3, YOLO sử dụng hệ màu HSV) 2. Qua các layers, biến đổi image 448x448x3 thành 1 grid có kích thước 7x7 với số tham số cho mỗi ô trong grid là 30 (30 = 5xboxes_number + number_of_objects) 3. Neural net có nhiệm vụ huấn luyện các **trọng số của các layers** để có được mô hình tốt cuối cùng. 4. Để nhận dạng một image mới, các bước 1,2 sẽ được thực hiện. Sau đó dựa vào các **tham số trong grid 7x7x30** ta sẽ xác định được các box chứa object với xác xuất cao. (các box đè lên nhau sẽ được loại bằng phương pháp NMS, chỉ giữ lại box có xác xuất cao nhất)

Có thể xem video sau của các bạn gấu Nga để hiểu rõ hơn (chỉ có phần nhận dạng, ko có phần huấn luyện) https://youtu.be/L0tzmv--CGY&usg=ALkJrhgvWHttKiRTLXiQDrI_a3f0tpJ0oA

4. Huấn luyện

Để hiểu cách neural net huấn luyện các trọng số thì điều quan trọng nhất là phải hiểu hàm mất mát (loss function)

4.1 Các khái niệm

- $1_{i\,j}^{obj} = 1$ nếu box thứ j của ô thứ i có chứa object. Vì huấn luyện cần các image với ground-truth (vị trí của các objects) nên YOLO biết điểm trung tâm của từng object rơi vào ô nào trong grid 7x7.
- $1_{i\,j}^{noobj} = 1$ nếu box thứ j của ô thứ i không chứa object.
- $1_{i}^{obj} = 1$ nếu ô thứ i có chứa object
- $S^2 = 7 \times 7$, B = boxes_number = số box mỗi ô sẽ dự đoán, được cố định = 2
- $\lambda_{coord} = 5.0$, $\lambda_{noobj} = 0.5$
- $classes$: các lớp đối tượng cần được nhận dạng, ví dụ chó, mèo, oto...

$$\lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right.$$

$$+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 \right.$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C} \right.$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i \right.$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in}$$

loss function

4.2 Hàm mất mát

- 1. Tính toán loss (tổn thất) của điểm trung tâm (x, y) cho hộp j của ô i nơi object tồn tại. Chú ý là $\hat{x}_i$ là tham số của tensor output của neural net còn x_i là của ground-truth. Tương tự cho tất cả các biến khác.
- 2. Tính toán tổn thất width và height của hộp j của ô i nơi object tồn tại.
- 3. Đối với các hộp j của ô i nơi object tồn tại, tính tổn thất của xác xuất object tồn tại. Chú ý C_i luôn = 1.
- 4. Đối với hộp j của ô i và nơi không có object, tính tổn thất của xác xuất này. Chú ý C_i luôn = 0.
- 5. Tính tổn thất của xác suất có điều kiện cho ô i nơi object tồn tại. Chú ý $p_i(c)$ luôn = 1 nếu đúng lớp c với ground-truth, ngược lại thì $p_i(c)$ luôn = 0.

- $\lambda_{coord} = 5.0$ thông số cân bằng để cân bằng tổn thất tọa độ (x, y, w, h) với các tổn thất khác.
- $\lambda_{noobj} = 0.5$ thông số cân bằng để cân bằng giữa hộp có và không có object. (Nói chung, đa số các ô trong image không có object, rất ít ô có object)

4.3 Huấn luyện

Đây là kiến thức cơ bản của neural net. Neural net sẽ tính toán từng ảnh (có thể lặp lại 1 ảnh) để tối ưu hàm mất mát. Việc tối ưu này sẽ giúp neural net tìm ra 1 bộ trọng số tốt nhất để biểu diễn dữ liệu của bạn và giúp nhận dạng các ảnh mới.

Hiểu hàm mất mát giúp bạn hiểu cơ chế hoạt động của từng neural net và cách huấn luyện chúng và giúp bạn dễ dàng hiểu code source, và trải nghiệm với nó.

5. YOLO version 2 (YOLOv2)

5.1 Thay đổi quan trọng : sử dụng Anchor boxes

Anchor boxes là các box được định nghĩa trước về hình dạng (width, height). Kĩ thuật này được giới thiệu trong Faster RCNN (paper)
YOLO dự đoán trực tiếp các thông số của hộp chứa object (hình chữ nhật, bounding box) dựa vào connected layers. YOLOv2 loại bỏ connected layers và các convolutional layers sẽ dự đoán các tham số của hộp chứa object dựa vao anchor boxes rồi tinh chỉnh x,y,width,height cũng như các xác xuất $\hat{C}_i$ và $\hat{p}_i(c)$)
Để có anchor boxes, YOLOv2 sử dụng k-means clustering trên các ground-truth boxes (thông số các objects trong các ảnh dùng để huấn luyện).

5.2. Các thay đổi khác cần lưu ý

- Sử dụng Batch normalization
- grid_size=13x13
- box_number=5

- image_dimension = 416x416

Có thể so sánh sự thay đổi dựa vào file config của YOLO vs YOLOv2:
cfg YOLO vs. cfg YOLOv2