Architecture des systèmes d'information

Machine-Learning

Bibliographie

You Only Look Once: Unified, Real-Time Object Detection

Prieur Maxime - Quarez Etienne

Clément Chatelain

Janvier - Mai 2020

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES ROUEN

I. Papier

La méthode décrite est intitulée “You Only Look Once : Unified, Real-Time Object Detection” et a été initialement décrite dans un papier du même nom. Le papier original est écrit par 4 co-auteurs : Joseph Redmon, Santosh Divvala de Allen Institute for AI, Ross Girshick de Facebook AI et Ali Farhadi, enseignant à l’université de Washington.

Le document est paru en 2016 dans le cadre de la conférence “vision par ordinateur et la reconnaissance de schéma/modèle” (CVPR).

Suite à la parution, le modèle a pu être amélioré et est aujourd'hui à sa version 3 dont les spécificités ont été décrites dans “YOLOv3 : An Incremental Improvement” paru le 8 avril 2018.

II. Problème de machine-learning

Le problème que ce papier essaye de résoudre est la reconnaissance d’objet, plus spécifiquement, en temps réel.

Les données en entrée sont donc des images couleur de taille variable. Le problème étant la reconnaissance d’objets dans une image, les données de sortie sont multiples pour une seule image. Chaque donnée de sortie correspond à un objet et est constituée d’une liste de coordonnées, d’une valeur de confiance et d’un label. Les coordonnées correspondent à la bounding box estimée de l’objet sur l’image.

III. Solutions déjà présentes

En général, les systèmes de vision par ordinateur tendent vers des structures de plus en plus larges et profondes avec certes, de bonnes précisions mais souvent au détriment de la vitesse de traitement.

En plus de cela, la majorité des réseaux tels que R-CNN classifient des zones proposées avant de déterminer plus précisément les contours de l’objet, d’éliminer les duplicatas et d’adapter la détection en fonction des autres objets. Malheureusement le traitement de ce genre de méthode est long et dur à optimiser puisqu'il faut traiter chaque composant séparément. En plus de cela le focus sur des parties de l’image ne permet pas au model d’étudier le contexte de l’image et induit ainsi des erreurs d’identification sur le second-plan.

III. Solutions proposée

La détection est abordé comme un problème de régression dans le but de séparer spatialement à la fois les bounding boxes et la probabilité d'attribution. La pipeline étant un réseau unique, il est possible de l’optimiser de bout en bout. De plus, la méthode traite l’image dans son ensemble sans utiliser de fenêtre mouvante.

Le système sépare l'image en une grille de SxS cellules. La cellule contenant le centre d'un objet est responsable de sa détection. A la place d'anchors, créant une instabilité, Yolo prédit ses bounding boxes depuis des clusters générées par l'algorithme k-means. Le réseau renvoie 4 coordonnées pour chaque boxe : $t_x,\ t_y,\ t_w,\ t_h$. On obtient alors : $b_x = \sigma(t_x) + c_x,\ b_y = \sigma(t_y) + c_y,\ b_w = p_w exp(t_w)\ et\ b_h = p_h exp(t_h)$. Avec (c_x, c_y) l'offset de la cellule depuis le coin supérieur gauche de l'image et (p_w, p_h) les dimensions de la prior bounding box (une de celles générées par k-means).

A. Modèle - DarkNet-53

Le modèle d'extraction de caractéristiques de la méthode YoloV3 est appelé Darknet-53 (dû aux 53 couches de convolution). Ce modèle prend en entrée des images de dimension variables et se compose de couches de convolution pourvue de filtres 3x3 et 1x1 ainsi que de couches résiduels. Les filtres 1x1 permettent de compresser la représentation des caractéristiques.

Toutes les couches de convolution effectuent une batch normalisation afin de stabiliser l'entraînement, rendre plus rapide la convergence et régulariser le modèle (permet également de supprimer la couche de dropout sans faire d'overfitting).

Suite à ces couches, on retrouve une couche GAP (global average pooling) permettant la classification pour différent format d'input. Enfin, le modèle se termine par une couche entièrement connectée et une couche softmax.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Architecture du modèle YoloV3

B. Loss et métriques

- La métrique de distance utilisée pour l'algorithme K-means utile à la définition des priors pour les bounding boxes est $d(box, centroid) = 1 - IOU(box, centroid)$.

- La loss utilisée pour la détermination de l'emplacement des bounding boxe est la Sum-Squared Error (Loss pour les méthodes de régression) qui calcul la racine de la dimension d'une boxe pour adresser le problème d'échelle entre la taille des boxes.

$$
\begin{aligned}
&\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
&+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2
\end{aligned}
$$

Fonction Loss du modèle YoloV3

1_i^{obj} indique si l'objet apparaît dans la ième cellule et $1_{i,j}^{obj}$ que la jème prédiction de box de la cellule i est responsable de la prédiction.

- La loss utilisée pour la prédiction de classe est la loss binaire cross entropique.

IV. Expérimentation

Network	AP	AP50	AP75
YoloV2	21.6	44.0	19.2
YoloV3	33.0	57.9	34.4
ResNet-101	40.8	61.1	44.1
Faster R-CNN	36.8	57.7	39.2

Performances de différent modèles sur Coco dataset

On observe avec l'évolution de la valeur de la métrique que les performances de Yolo diminuent à mesure que le seuil de l'IOU augmente, indiquant une difficulté du modèle à aligner parfaitement ses boxes avec l'objet réel.

V. Conclusion

Bien que la méthode émette moins de faux positifs sur l'arrière-plan d'une image, celle-ci reste moins précise par rapport aux autres méthodes, quant à l'emplacement des objets. De plus la méthode peine à prédire des objets ayant des ratios non-habituels et souffre de faible recall comparé aux autres méthodes de détection. Pour conclure, on peut dire que la méthode Yolo est une méthode de détection d'objet en temps réel efficace et rapide même si celle-ci commence à se faire un peu vieille en comparaison avec des méthodes de l'état de l'art actuelles tel que CenterNet ou bien ResNet. En remarque, j'aimerais ajouter que les premiers papiers font fit d'une très bonne rigueur scientifique sur la présentation de Yolo, malheureusement l'auteur principal Joseph Redmond a laissé beaucoup de points flous sur sa dernière amélioration.

VI. Références

https://arxiv.org/pdf/1506.02640.pdf, *You Only look Once : Unified, Real-Time Object Detection*. **Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi**. 9 Mai 2016

https://arxiv.org/pdf/1612.08242.pdf, *Yolov9000 : Better Faster Stronger*. **Joseph Redmon, Ali Farhadi**. 25 Décembre 2016

https://arxiv.org/pdf/1804.02767.pdf, *Yolov3 : An Incremental Improvement*. **Joseph Redmon, Ali Farhadi**. 8 Avril 2018

VII. Annexe

A. Terminologie

- **Anchor box** : Ensemble de “pré-bounding box” ayant un ratio taille/largeur déterminé pour permettre au modèle de détecter plusieurs objets de façon spécifique et optimisée dans une image.

- **Squelette / Backbone** : Réseau d’extraction de caractéristiques utilisé au sein de l’architecture profonde. Encode les entrées vers une certaine représentation.

- **IOU (Intersection Over Union) :** Métrique représentant la précision de la zone prédite par rapport à la zone réelle de l’objet. $IoU = \frac{Aire\ d'intersection}{Aire\ d'union}$.

- **Interpolation bilinéaire** : Calcul de la valeur d’une fonction en un point quelconque à partir de ses deux plus proches voisins dans chaque direction.

- **Précision** : Métrique permettant de contrôler le taux de faux positifs.
  $Precision = \frac{True\ positive}{True\ Positive + False\ Positive} = \frac{Nombre\ de\ prédictions\ positives\ correctes}{Nombre\ de\ prédictions\ positives\ totales}$

- **Rappel :** Métrique permettant de contrôler le taux de faux négatifs.
  $Recall = \frac{True\ positive}{True\ Positive + False\ Negative} = \frac{Nombre\ d'objets\ prédits\ positifs}{Nombre\ d'objets\ véritablement\ positifs}$

- **AP** (Average Precision) : Moyenne de la précision en fonction des différentes valeurs du Rappel (recall).
  $AP = \frac{1}{(10+1)} \sum_{Recall=0}^{10} Precision(\frac{Recall}{10})$
  Note : Pour savoir si un objet est correctement prédit en détection d’objets, un seuil arbitraire est introduit sur l’IoU, par exemple AP_{50} signifie qu’un objet est considéré comme correctement prédit si son IoU est supérieur à 0.50.

- **mAP** (mean Average Precision) : Moyenne des AP de chaque classe

- **NMS** : Pour “non-maximum suppression” est une technique permettant de s’assurer que le détecteur ne détecte que une fois un objet.