

A5 YOLO report

B094020007 陳文薇

a. a brief discussion about your implement

In the beginning, I tried to find out if there's some reference. However, i found out that the way to write this assignment is different from the others I can search. For the very first try, i wrote the functions from top to bottom, but chi cheng told me that i can begin with the forward function. So I follow the instructions in the forward function and coding... here's how i implemented in every function.

find_best_iou_boxes(),

First, identify the number of grid cells associated with the object. Next, calculate the Intersection over Union (IoU) between the predicted bounding box center (`box_p_center`) and the target bounding box center (`box_t_center`). Determine the highest IoU value and its corresponding index. Finally, mark the position of the object by assigning 1 to the grid cell associated with the bounding box that achieved the highest IoU (position labeling), and assign the highest IoU value to the corresponding index (IoU labeling for the same position).

get_class_prediction_loss() & get_contain_conf_loss(),
just simply calculate the mse loss.

get_no_object_loss(),

First, identify the cells in both `pred_tensor` and `target_tensor` that do not correspond to the object. Create a mask with the same dimensions as the `no_object_prediction`. Set the values in the confidence column of the mask to 1. Determine the locations where the `no_object_prediction_mask` is present in both the prediction and target. Finally, calculate the Mean Squared Error (MSE) based on these two sets of values.

get_regression_loss(),
apply the formula in yolo v1 paper.

forward(),

1. Generate a target tensor with the same dimensions as the prediction tensor. Identify cells with and without objects in the target tensor and store them separately. Expand both sets to match the size of the target tensor.
2. For cells with objects in the prediction tensor, determine their predicted values, extract bounding box information, and retrieve class information based on the predicted values.
3. For cells with objects in the target tensor, determine their target values, extract bounding box information, and retrieve class information based on the target values.
4. Calculate the losses for class predictions (`get_class_prediction_loss`) and no-object predictions (`get_no_object_loss`).
5. Identify corresponding indices from `bounding_box_pred` and `bounding_box_target` and determine the value of the best Intersection over Union (IoU).
6. Calculate the containment confidence loss (`get_contain_conf_loss`) by considering only the confidence values.
7. Finally, compute the total loss by combining all the calculated losses.

- b. Report the best validation mAP in all of your experiments and discuss any strategies

or tricks you've employed.

first record: epoch=35, mAP=0.43

second:epoch=50, mAP=0.40

third:epoch=50, iou threshold=0.66, mAP=0.53

so train more and modify the threshold might help.(modify iou threshold might help is reference to a student in our class that he wrote a public report on kaggle.)

- c. Report the results for extra credits and also provide a discussion, if any

predict_video是參考predict.py中的predict_image()下去改動的。