Faster-RCNN

Tan Wentao

2018 年 3 月 23 日

1 train-frcnn

1.1 introdution

This file is training the faster RCNN. It will load the config params firstly, then initialize the network params used by the command line params.You must choose pascal_voc or simple, then you will get all images.Using get_anchor_gt() to get the anchors.Then it defines the network architecture.The network includes base_networks(Resnet, VGG, Inception), rpn and classifier. Train the network and calculate the loss in the end.

1.2 functions

- sys.setrecursionlimit
  It can change the maximum recurion depth.

- OptionParse
  You can create an instance of OptionParser, populate it with options, and parser the command line.
  (1)path: Path to training data.
  (2)parser: Parser to use one of simple or pascal_voc.
  (3)num_rois: Number of ROIs per iteration which be randomly selected.
  (4)horizontal_flips: Augment with horizontal flips in training.
  (5)vertical_filps: Augment with vertical flips in testing.
  (6)rot_90: Augment with 90 degree rotations in training.
  (7)num_epoch: Number of epochs.
  (8)config_filename: Location to store all the metadata related to the training to be used when testing.
  (9)output_weight_path: Output path for weights.
  (10)input_weight_path: Input path for weights. If not specified, will try to load default weights provided by keras.

2 test-frcnn

3 measure-map

4 keras-frcnn

4.1 config

- verbose

- use_horizontal_flips
  Augment with horizontal flips in training.
- use_vertical_flips
  Augment with vertical flips in training.
- anchor_box_scales
  Set a list with multi-scales of anchor.
- anchor_box_ratios
  Set a list with multi-ratios(height : width) of a anchor.
- im_size
  Set the image size.
- img_channel_mean

- img_scaling_factor

- num_rois

- rpn_stride

- balanced_classes

- std_scaling

- classifier_regr_std

- rpn_min_overlap
- rpn_max_overlap
- classifier_min_overlap
- classifier_max_overlap
- class_mapping
- image_dim_ordering
- model_path

4.2 data-augment

- copy
  copy(x) will return a shallow copy of x. deepcopy(x) will return a deep copy of x. In case of shallow copy, a reference of object is copied in other object. It means that any changes made to a copy of object do reflect in the original object. In case of deep copy, a copy of object is copied in other object. It means that any changes made to a copy of object do not reflect in the original object.
- augment
  Using three methods to enhance the image data.There are 1.use_horizontal_flips, 2.use_vertical_filps and 3. rot_90.The function will return the augment images and original images.

4.3 data-generators

- get_img_output_length
  It calculate the size of the image after four convlutions.The kernel size is [7,3,1,1], padding size is 6 and stride is 2.Then can calculate the output size.

$$output\_size = \frac{input\_size - kernel\_size + stride}{stride} \tag{1}$$

- area
  Input is a box A, then will return the area of A.

- union

  Inputs are two boxs A and B. A(or B) is [x_A_min, y_A_min, x_A_max, y_A_max]

$$\begin{cases} x\_min = min(x\_A\_min, x\_B\_min) \\ y\_min = min(y\_A\_min, y\_B\_min) \\ x\_max = max(x\_A\_max, x\_B\_max) \\ y\_max = max(y\_A\_max, y\_B\_max) \end{cases} \tag{2}$$

  According to the above formula, then

$$\begin{cases} w = x\_max - x\_min \\ h = y\_max - y\_min \end{cases} \tag{3}$$

  So it will return quadruples [x, y, w, h]. It represents the largest region after the merge.

- intersection

  Input are two boxs A and B. A(or B) is [x_A_min, y_A_max, y_A_max]

$$\begin{cases} x\_min = max(x\_A\_min, x\_B\_min) \\ y\_min = max(y\_A\_min, y\_B\_min) \\ x\_max = min(x\_A\_max, x\_B\_max) \\ y\_max = min(y\_A\_max, y\_B\_max) \end{cases} \tag{4}$$

  According to the above formula. Then

$$\begin{cases} w = x\_max - x\_min \\ h = y\_max - y\_min \end{cases} \tag{5}$$

  So it will return quadruples [x, y, w, h]. It represents the intersection region.

- get_new_img_size

  The size of original image is adjusted according to the length of the minimum side of the input.

$$\begin{cases} \frac{width}{height} = \frac{new\_width}{new\_height} \\ new\_width = img\_min\_size \quad if \quad width < height \\ new\_heighte = img\_min\_size \quad if \quad width > height \end{cases}$$

- SimpleSelector
- SimpleSelector::skip_sample_for_balanced_class
- calc_rpn
- threadsafe_iter

- threadsafe_generator
- get_anchor_gt
-

4.4 FixBatchNormalization

4.5 losses

- rpn_loss_regr
- rpn_loss_cls
- class_loss_regr
- class_loss_cls

4.6 pascal-voc-parse

4.7 resnet

This is the model of resnet50. Reference: [Deep Residual Learning for Image Recognition](https://arxiv.org/abs/1512.03385)

- identity_block
- identity_block_td
- conv_block
- conv_block_td
- nn_base
- classifier_layers
- rpn
- classifier

4.8 roi-helpers

4.9 RoiPoolingConv

4.10 simple-parser

5 requirements

- h5py

- Keras==2.0.3
- numpy
- opencv-python
- sklearn