

针对官方 YoloX-nano 网络主要进行如下修改

整体修改描述:

模型描述:yolox-nano 网络(第一个 concat 的两路输入通道数改为 32), focus 层替换为等价卷积, RELU 激活函数。模型 fp32 大小 3.43MB, int8 模型 950kb; 网络输入 size:416*416*3 前处理 RGB mean= [123.675, 116.28, 103.53], norm= [0.017124753831663668, 0.01750700280112045, 0.017429193899782133] !!!

1.第一个 concat 的两路输入通道修改为 32, 修改位置如下:

YOLOX/ models / backbone / csp_darknet.py

```
330 class CSPDarknet(nn.Module):
331
332     def __init__(self, dep_mul=1., wid_mul=1., out_indices=(3, 4, 5), depthwise=False, act="silu"
333         super().__init__()
334         self.out_features = out_indices
335         Conv = DWConv if depthwise else BaseConv
336
337         base_channels = int(wid_mul * 64) # 64
338         base_depth = max(round(dep_mul * 3), 1) # 3
339
340         # stem
341         self.stem = Focus(3, base_channels, ksize=3, act=act)
342
343         # dark2
344         self.dark2 = nn.Sequential(
345             Conv(base_channels, base_channels * 2, 3, 2, act=act),
346             CSPLayer(
347                 base_channels * 2, base_channels * 2,
348                 n=base_depth, depthwise=depthwise, act=act,
349                 expansion=1.0, #concat输入通道数32倍数, for relu_yolox porting
350             ),
351         )
352
353         # dark3
354         self.dark3 = nn.Sequential(
355             Conv(base_channels * 2, base_channels * 4, 3, 2, act=act),
356             CSPLayer(
357                 base_channels * 4, base_channels * 4,
358                 n=base_depth * 3, depthwise=depthwise, act=act,
359             ),
360         )
361
362         # dark4
```

2.focus 层替换为等价卷积:

YOLOX/ models / backbone / csp_darknet.py

```

44  ...
45  #原focus
46  class Focus(nn.Module):
47      """Focus width and height information into channel space."""
48
49      def __init__(self, in_channels, out_channels, ksize=1, stride=1, act="silu"):
50          super().__init__()
51          self.conv = BaseConv(in_channels * 4, out_channels, ksize, stride, act=act)
52
53      def forward(self, x):
54          # shape of x (b,c,w,h) -> y(b,4c,w/2,h/2)
55          patch_top_left = x[..., ::2, ::2]
56          patch_top_right = x[..., ::2, 1::2]
57          patch_bot_left = x[..., 1::2, ::2]
58          patch_bot_right = x[..., 1::2, 1::2]
59          x = torch.cat(
60              (patch_top_left, patch_bot_left, patch_top_right, patch_bot_right,), dim=1,
61          )
62          return self.conv(x)
63  ...
64  class Focus(nn.Module):
65      def __init__(self, in_channels, out_channels, ksize=1, stride=1, act="silu"):
66          super().__init__()
67          self.focus_conv = nn.Conv2d(3, 12, 2, 2, bias = False)
68          self._init_focus_conv()
69          self.conv = BaseConv(in_channels * 4, out_channels, ksize, stride, act = act)
70
71      def _init_focus_conv(self):
72          conv_1_weight = np.zeros((3, 2, 2))
73          conv_1_weight[0, 0, 0] = 1
74          conv_1_weight = np.expand_dims(conv_1_weight, 0)
75
76          conv_2_weight = np.zeros((3, 2, 2))
77          conv_2_weight[1, 0, 0] = 1

```

3.将激活函数替换为 ReLU:

YOLOX/ config.py

```

69  # base params
70  opt.warmup_lr = 0 # start lr when warmup
71  opt.basic_lr_per_img = 0.01 / 64.0
72  opt.scheduler = "yoloxwarmcos"
73  opt.no_aug_epochs = 30 # close mixup and mosaic augments in the last 15 epochs
74  opt.min_lr_ratio = 0.05
75  opt.weight_decay = 5e-4
76  opt.warmup_epochs = 5
77  opt.depth_wise = False # depth_wise conv is used in 'CSPDarknet-nano'
78  opt.stride = [8, 16, 32] # YOLOX down sample ratio: 8, 16, 32
79  opt.activation = 'relu' #'silu'#
80  # train augments
81  opt.degrees = 10.0 # rotate angle
82  opt.translate = 0.1
83  opt.scale = (0.1, 2)
84  opt.shear = 2.0
85  opt.perspective = 0.0

```