Pruning YOlov7

Created on : 2024-07-16
tags: []
References: https://github.com/amitpant7/YOLOv7_Prune

Results of Pruning the First Trained Model with mAP = 0.6

After Retraining on Pascal VOC

After retraining on the Pascal VOC dataset for 10 epochs, the mean Average Precision (mAP) results are as follows:

```
val: Scanning 'Pascal-Voc-2007-1/valid/labels.cache' images and labels... 1001 f
               Class      Images      Labels           P           R      mAP@.5
                 all        1001        3021         0.8       0.766       0.822       0.603
           aeroplane        1001          68       0.915       0.897       0.929       0.682
             bicycle        1001          81       0.753       0.889       0.897       0.639
                bird        1001         156         0.9       0.699       0.812       0.571
                boat        1001          96       0.783       0.688       0.797       0.488
              bottle        1001          75       0.784       0.628       0.697       0.502
                 bus        1001          49       0.764       0.939       0.929       0.766
                 car        1001         291       0.857       0.863       0.923       0.706
                 cat        1001          89       0.862        0.84       0.888       0.732
               chair        1001         266       0.723       0.617       0.704       0.501
                 cow        1001         111       0.721       0.847       0.843       0.581
         diningtable        1001          55       0.756       0.691       0.716       0.556
                 dog        1001         117       0.954       0.714       0.881       0.698
               horse        1001          63       0.906       0.921       0.932       0.743
           motorbike        1001          73       0.696       0.753       0.776       0.531
              person        1001        1031       0.818       0.798       0.833       0.544
         pottedplant        1001         137        0.74       0.582       0.662       0.358
               sheep        1001          53       0.651        0.66       0.706       0.504
                sofa        1001          80         0.7       0.699       0.759       0.608
               train        1001          60       0.903         0.8       0.878       0.698
           tvmonitor        1001          70       0.821         0.8       0.868       0.643
Speed: 7.0/1.3/8.2 ms inference/NMS/total per 416x416 image at batch-size 16
```

Quantization mAP:

```
VAIQ_NOTE]: =>Successfully convert 'Model_0' to xmodel.(nndct/Model_0_int.xmodel)
                 all        258         769       0.789       0.721       0.774       0.545
           aeroplane        258          19       0.916       0.947       0.946       0.721
             bicycle        258          19       0.616       0.846       0.827        0.61
                bird        258          43       0.864       0.443       0.703       0.426
                boat        258          17        0.88       0.765       0.781       0.511
              bottle        258          15       0.542       0.333       0.389       0.254
                 bus        258          13       0.851       0.879       0.952         0.7
                 car        258          62       0.904       0.919       0.954       0.702
                 cat        258          20       0.771         0.9       0.864       0.711
               chair        258          66        0.66       0.591       0.646       0.421
                 cow        258          42       0.731       0.762       0.832       0.571
         diningtable        258          11       0.877       0.545        0.63       0.331
                 dog        258          29       0.844       0.552       0.784       0.587
               horse        258          21       0.984       0.952       0.971       0.686
           motorbike        258          10       0.976         0.9       0.913       0.641
              person        258         277       0.814       0.774        0.81       0.496
         pottedplant        258          32       0.573         0.5         0.5       0.268
               sheep        258          12       0.574       0.417       0.419       0.321
                sofa        258          30       0.758       0.733       0.814       0.599
               train        258          14       0.802       0.714       0.804       0.681
           tvmonitor        258          17       0.841       0.941       0.936       0.664
Speed: 2190.6/1.8/2192.5 ms inference/NMS/total per 416x416 image at batch-size 1
Results saved to runs/test/yolov7_640_val51
```

Inference Time:

- **CPU:** 3 FPS
- **GPU P100:** 66.31 FPS

Without post-processing:

- **GPU:** 68.23 FPS

Model Pruning:

All of the pruning experiments are based on lamp pruning by towards structured pruning, torch-pruning.

First Experiment

Pruning 20% of Channels:

- **Iteration:** 1/1
- **Parameters:** 37.30 M => 22.64 M
- **MACs:** 22.21 G => 18.20 G

After Quantization

```
[VAIQ_NOTE]: =>Successfully convert 'Model_0' to xmodel.(nndct/Model_0_int.xmodel)
               all         258         769       0.732       0.632       0.663       0.425
         aeroplane         258          19           1        0.87       0.916       0.583
           bicycle         258          19       0.737       0.737       0.802       0.535
              bird         258          43       0.686       0.279       0.385       0.228
              boat         258          17       0.705       0.765       0.678        0.36
            bottle         258          15       0.994         0.4       0.531       0.338
               bus         258          13       0.935       0.769       0.776       0.482
               car         258          62       0.846       0.797       0.895       0.595
               cat         258          20       0.778         0.7       0.742       0.476
             chair         258          66       0.573         0.5       0.517       0.309
               cow         258          42       0.582       0.571       0.621       0.394
       diningtable         258          11       0.592       0.455       0.451        0.26
               dog         258          29       0.672        0.69       0.698       0.436
             horse         258          21       0.892        0.79       0.813       0.561
         motorbike         258          10       0.882       0.749       0.875        0.61
            person         258         277       0.701       0.644       0.651       0.348
       pottedplant         258          32        0.69       0.375       0.451       0.243
             sheep         258          12       0.235        0.25       0.105      0.0772
              sofa         258          30       0.722       0.667       0.734       0.481
             train         258          14       0.754       0.786       0.729       0.535
         tvmonitor         258          17       0.673        0.85       0.896       0.642
Speed: 1348.7/1.7/1350.4 ms inference/NMS/total per 416x416 image at batch-size 1
```

Further Experiments

Experiment	Pruning Ratio	Retrain Epochs	Prune Percentage	mAP_0.5	mAP_0.5:0.95	CPU Speed (FPS)	GPU P100 Speed (FPS)	GPU Speed without Post-processing (FPS)
Second	0.25	40 (one-shot)	45%	0.70878	0.47282	4.80	64.50	-
Third	0.35	40	60%	0.6584	0.43099	5.00	71.00	-
Fourth	0.15	40	37.30 M => 26.88 M	0.749	0.517	-	-	-

Let me know if you need any adjustments or additional details.

Iterative vs One-shot Pruning

I tried pruning YOLOv7 with a channel pruning ratio of 0.35 in two ways:

1. **One-shot Pruning:** Pruned the model directly by 0.35 and retrained for 40 epochs.
2. **Iterative Pruning:** Pruned 25% of channels and retrained for 20 epochs. After improving the mAP, pruned this 25% channel-pruned network by an additional 14% (totaling 35%) and retrained for 20 epochs. The model was not able to surpass the initial mAP.

Summary Tables

All pruned models were trained for 40 epochs:

Pruning Results Summary

Experiment	Pruning Ratio (Channels)	Prune Percentage	Retrain Epochs	mAP (0.5)	mAP (0.5:0.95)	Speed (CPU)	Speed (GPU P100)
Initial (Retrain) (37.30M)	-	-	10	0.81	0.6	3 FPS	66.31 FPS
Pruned	0.15	-	40	0.51	0.74	-	-
Pruned	0.25	45%	40	0.70878	0.47282	4.80 FPS	64.50 FPS
Pruned	0.35	60%	40	0.6584	0.43099	5 FPS	71 FPS

Video Inference FPS

The FPS includes post-processing time too.

Models	Total Prune (In terms of parameters)	CPU (FPS)	GPU (P100) (FPS)	Model Size (MB)
No prune	37.3 M (100%)	4.98	77	75
10% prune (L2 Norm, M)	20%	5.00	58	60
20% prune (L2 Norm, M)	36%	5.4	66	48
25% prune (LAMP, M)	45%	4.65	76	41
35% prune (L2 Norm, M)	60%	8.00	87	31
50% prune (GroupNorm, Group Norm)	75%	13	124	19

Deploying on FPGA

Model	Post-training Quantization mAP	FPS FPGA Board (Kria KV260 (DPU))	Comments
No_Prune_map_60	0.72 / 0.54	7 FPS	These results are without video buffer
Pruned_C10_T20_map54_magnitude	0.736 / 0.498	14	on video buffer the
Pruned_C20_T36_map_51_magnitude	0.71 / 0.47	15	fps improved to 25.
Pruned_C35_T60_map43_LAMP	0.61 / 0.4	16	

Model	Post-training Quantization mAP	FPS FPGA Board (Kria KV260 (DPU))	Comments
Pruned_C50_T75_map35_GroupNorm	0.5 / 0.29	17	

Observations

1. **Why Model Inference Increases with Pruning (e.g., C=0.25 for GPU):**
   - The matrix multiplication operations in GPUs are most efficient when the channels are in multiples of 8.
2. **Why Pruning Didn't Improve mAP in Channel Pruning on GPU:**

1 month later

spolisetty Moderator — Jan 04 '22

Hi,

If you channel prune models in the right way (and then compress them), you won't get any increase in speed in TensorRT. GPUs are simply very good at dense math, so unless sparsity is appropriately structured or weights are very sparse, sparse computations are unlikely to improve performance.

Thanks