PreDet: Large-scale weakly supervised pre-training for detection

Vignesh Ramanathan
Facebook AI
vigneshr@fb.com

Rui Wang
Facebook AI
ruiw@fb.com

Dhruv Mahajan
Facebook AI
dhruvm@fb.com

1. Learning rate parameters

ResNext-101-32x8d + FPN: For all models, including PreDet , InfoMin, SEER, from-scratch and classification pre-trained, models we conducted a grid-search to identify the best learning-rate and gamma values for each model. Additionally, we also did a grid-search to identify the best scaling factor by which learning rate for the mask-head parameters should be scaled for each of the models, when training a mask-RCNN model. The learning rate, γ and scaling factor (lr_{mf}) were searched in the following sets respectively: $(0.005, 0.01, 0.02, 0.03, 0.04)$, $(0.1, 0.15, 0.2)$ and $(1.0, 4.0)$. The best learning rate parameters for different models are shown in Tab. 1:

pre-training	dataset	model	lr	γ	lr_{mf}
from-scratch	COCO	mask-RCNN	0.02	0.1	4.0
cls-Imagenet			0.02	0.1	1.0
cls-IG50M			0.01	0.15	1.0
SEER-1B			0.02	0.1	4.0
Infomin			0.02	0.1	4.0
PreDet -ImageNet			0.02	0.1	4.0
PreDet -IG50M			0.01	0.1	4.0
from-scratch	LVIS-v1	mask-RCNN	0.02	0.1	4.0
cls-Imagenet			0.02	0.1	1.0
cls-IG50M			0.01	0.15	1.0
SEER-1B			0.02	0.1	4.0
Infomin			0.02	0.1	4.0
PreDet -ImageNet			0.02	0.1	4.0
PreDet -IG50M			0.005	0.2	4.0
from-scratch	COCO	Retinanet	0.02	0.1	-
cls-Imagenet			0.01	0.1	-
cls-IG50M			0.01	0.1	-
SEER-1B			0.01	0.1	-
Infomin			0.01	0.1	-
PreDet -ImageNet			0.01	0.1	-
PreDet -IG50M			0.01	0.1	-

Table 1. Learning rate parameters used for fine-tuning Resnext-101-32x8d+FPN pre-trained with different methods, for different models and datasets.

R50 + FPN for mask-RCNN : For PreDet -IG50M model, we used lr, γ values of $0.01, 0.15$ and did not scale the learning rate for the mask-head parameters.

ResNext-101-32x8d + FPN for mask-RCNN on smaller target datasets: When training on smaller subsets of COCO datasets, we did a grid-search to identify best learning rate and γ from the following sets respectively: $(0.04, 0.03, 0.02)$ and $(0.02, 0.05, 0.1)$. We also tried different overall fine-tuning iterations such as $60k, 70k, 130k$ iterations in addition to standard $1\times, 2\times$ schedules. We dropped the learning rate by γ twice at the following iterations for these 3 new schedules: for $60k$ iterations, dropped lr at $(40k, 55k)$ iterations, for $70k$ iterations, dropped lr at $(50k, 65k)$ iteration and for $130k$ iterations, dropped lr at $(100k, 120k)$ iterations. The best learning rate schedules for the different target dataset-sizes are shown below in Tab. 2.

pre-training	dataset-size	train schedule	lr	γ
from-scratch	35k	540k	0.04	0.02
cls-Imagenet		90k	0.04	0.02
PreDet		90k	0.04	0.02
from-scratch	10k	90k	0.04	0.02
cls-Imagenet		60k	0.03	0.02
PreDet		60k	0.02	0.1
from-scratch	5k	80k	0.02	0.1
cls-Imagenet		60k	0.02	0.1
PreDet		60k	0.02	0.1
from-scratch	1k	60k	0.02	0.1
cls-Imagenet		60k	0.02	0.1
PreDet		60k	0.02	0.1

Table 2. Learning rate parameters and fine-tuning used for target datasets of different sizes.

Regnet64 + FPN for mask-RCNN : We use lr of 0.01 and γ of 0.1 for PreDet . We use lr of 0.02 and γ of 0.1 for Imagnet pre-trained and SEER-IG1B models. As per SEER , we use a weight decay of $5e^{-5}$ for both pre-training and fine-tuning.

2. Detailed ResNext-101-32x8d + FPN results

We show detailed results including AP^{box}_{75} ,AP^{box}_{50} ,AP^{mask}_{75} ,AP^{mask}_{50} for ResNext-101-32x8d+FPN model for COCO and LVIS-v1 datasets below (corresponding to Fig.5, Fig.6 in main draft) in Tab. 3 and Tab. 4 respectively.

3. Details of CRPN

The query score generation and bounding box regression parts of the CRPN are run in parallel for all Q input queries. We visualize these components in detail in Fig. 1. Note

pre-training	sched.	mask-RCNN						RetinaNet		
		AP^{bbox}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}	AP^{bbox}	AP^{box}_{50}	AP^{box}_{75}
scratch	3×	43.3	63.0	47.7	38.8	60.4	41.6	38.9	57.8	41.8
	6×	45.6	65.6	49.6	40.4	62.8	43.9	40.7	60.0	43.3
	9×	45.8	65.6	50.2	40.7	63.0	44.2	40.7	59.7	43.4
cls-Imagenet	1×	43.8	65.3	47.9	39.0	61.6	42.3	41.4	61.8	44.1
	3×	44.3	64.5	48.6	39.5	61.7	42.6	-	-	-
	6×	44.9	64.9	49.0	39.9	62.2	43.0	-	-	-
SEER-IG1B	1×	44.3	65.7	48.6	39.9	62.4	43.1	40.3	59.9	43.2
	3×	45.1	65.6	49.4	40.1	62.7	43.2	41.7	60.9	44.7
Infomin	1×	44.8	65.5	49.0	40.2	62.5	43.5	43.0	62.8	46.4
	3×	45.6	65.9	49.9	40.5	63.2	43.6	-	-	-
cls-IG50M	1×	44.4	65.8	48.5	39.4	62.2	42.1	41.8	62.1	44.3
	3×	44.6	65.1	48.7	39.5	61.9	42.6	-	-	-
PreDet -Imagenet	1×	45.8	66.9	50.2	40.8	63.6	43.9	43.1	63.3	46.4
PreDet -IG50M	1×	**47.1**	**68.2**	**51.6**	**41.7**	**65.2**	**45.1**	**45.1**	**67.8**	**50.6**

Table 3. Results on MS-COCO for mask-RCNN and RetinaNet, with ResNeXt-101-32x8d + FPN backbone when pre-trained with different approaches. We report results upto the minimum fine-tuning schedule at which the model's performance converges.

pre-training	sched.	mask-RCNN					
		AP^{bbox}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
scratch	2×	22.6	35.9	24.1	21.7	33.6	22.9
	3×	26.1	40.1	28.4	25.1	38.1	27.0
	6×	28.0	42.4	30.4	26.9	40.3	28.9
cls-Imagenet	1×	24.5	39.2	26.3	24.3	37.2	25.9
	2×	26.1	40.5	28.3	25.1	38.2	27.1
	3×	25.6	39.6	27.6	24.8	37.5	26.2
SEER-IG1B	1×	28.2	44.3	30.0	27.7	41.8	29.5
	2×	28.6	43.8	30.8	27.8	41.4	29.8
Infomin	1×	25.3	39.6	27.1	24.7	37.6	26.5
	2×	27.3	41.6	29.6	26.3	39.4	28.3
PreDet -ImageNet	1×	26.1	40.7	27.7	25.6	38.6	27.3
PreDet -IG50M	1×	**30.1**	**46.7**	**32.1**	**29.2**	**44.2**	**31.1**

Table 4. Results on LVIS-v1 for mask-RCNN with ResNeXt-101-32x8d + FPN backbone when pre-trained with different approaches. We report results upto the minimum fine-tuning schedule at which the model's performance converges. Results are averaged over 3 runs for every model.

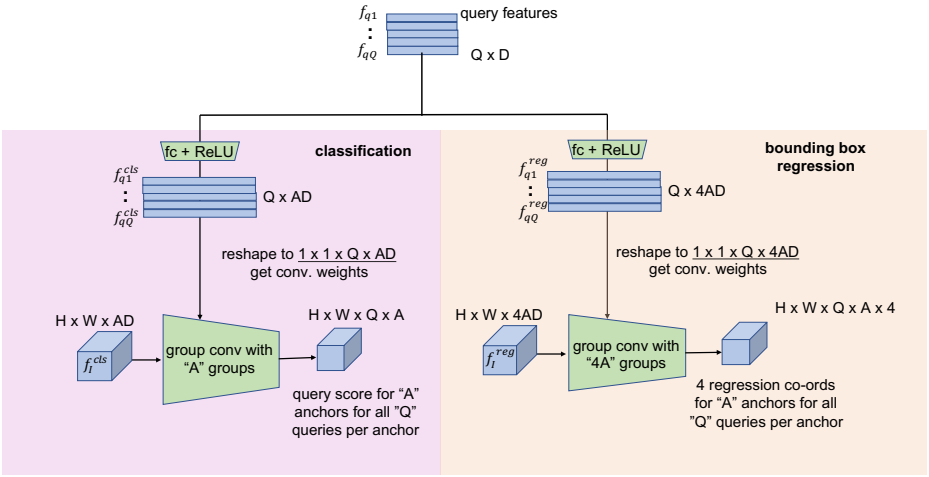

Figure 1. The classification and regression parts of the CRPN are visualized in more detail. This shows how query scores for A anchor boxes and 4 regression co-ordinates to regress these A anchor boxes to Q queries are computed for all anchors in parallel.

that the group convolutions are 1×1 convolutions whose weights are dynamically set by the output of the FC+ReLU layers in the classification and box regression parts.