Experiments

cheng yang

November 15 2022

Model	Fine-tune?	FLOPs(G)	Parameters(M)	SPL(%)	Reward
ResNet50-Baseline	N	1.23	5.59	0.897	7.2
ResNet50-Pruned	N	0.6 ↓51.2%	2.43 ↓56.5%	**0.901**	7.1
ResNet18	N	0.54	2.66	0.885	7.23

Table 1: Based on Habitat-lab, comparison of pruning and original ResNet on Gibson. In "Fine-tune?" column, "Y" and "N" indicate whether to use the pre-trained model as initialization or not(from scratch), respectively. The larger number of "SPL" and "Reward" is better, but "SPL" is more convincing.

Environments (BatchSize)	Latency(ms) Baseline	Latency(ms) Pruned	Realistic Speed-up(%)	Theoretical Speed-up(%)
bs=1	8.5	8.5	0	51.2
bs=8	18.9	12.8	34.4	51.2
bs=16	35.5	21.7	38.9	51.2
bs=64	101.9	76.4	25.02	51.2

Table 2: Comparison on the theoretical and realistic speedup on a GTX3060 GPU. I only count the time consumption of the forward procedure.

Environments (BatchSize)	Latency(ms) Baseline	Latency(ms) Pruned	Realistic Speed-up(%)	Theoretical Speed-up(%)
bs=1	35.7	25.5	28.6	51.2
bs=8	139.0	115.2	17.1	51.2

Table 3: Comparison on the theoretical and realistic speedup on the "Intel(R) Core(TM) i7-10870H CPU @ 2.20GHz". I only count the time consumption of the forward procedure.

To sum up, We can learn the following information from the above table:

1. Based on previous pruning method, The ResNet50-Pruned could achieve a better performance than the ResNet50-Baseline, but with significantly fewer FLOPs and Parameters(from Table1).

2. Due to the cuda feature, it makes almost no acceleration at bs=1. As bs increases, the speedup ratio is limited by IO delay and some base libraries to the point that there is gap between theoretical and realistic model(from Table2).

3. For cpu feature, with the increase of bs, the speedup ratio decreases(from Table3).

Interestingly, Almost all papers on structured pruning do not talk about the actual acceleration ratio on GPU(Unstructured pruning does not target gpu, so it is not discussed), especially bs=1. But I don't think this is very practical, because in practice, bs is almost always 1. So, I potentially came up with an idea that further optimizes the actual performance of pruning, especially for bs=1. Maybe it should be called "Probe-wise Pruning".

Personal feelings:

Through understanding the reinforcement learning, I feel more and more that it is the future trend, I expect it to explode, just as deep learning did in 2016 (but I only started to understand deep learning and computer vision in 2020).