1. Implementation Details

Pretraining is implemented using PyTorch [3] on 8 p3dn.24xlarge Amazon EC2 instances with a total of 64 Tesla V100 GPUs, while the fine-tuning uses PyTorch on a single p3dn.24xlarge Amazon EC2 instance with 8 Tesla V100 GPUs. We use DistributedDataParallel for multi-GPU training. We use automatic mixed precision for all of our experiments, and channels-last memory format for the ResNeXt experiments, in order to improve the training throughput. All of our model training runs and performance benchmarks use PyTorch 1.7.1, CUDA 11.0, and cuDNN 8.

Vision Transformer pretraining uses a warmup phase of 10k steps, total batch size of 8192, base learning rate (LR) of 8e-4, and linear decay LR schedule of 2 epochs in length, such that around 2.6B images are processed during the main phase of training. We train using the AdamW [2] optimizer with a weight decay value of 0.05. Vision Transformer Unified Visual Embedding fine-tuning uses a warmup phase of 5k steps, base LR of 0.24, and cosine decay LR schedule of 20 epochs in length. We fine-tune using the SGD optimizer with a base LR of 0.24 and weight decay of 1e-4 for the non-sparse parameters. Vision Transformer ImageNet fine-tuning uses a warmup phase of 5k steps, base LR of 0.03, cosine decay LR schedule of 50k steps, SGD optimizer, and zero weight decay.

ResNeXt-101 pretraining uses a warmup phase of 15k steps, total batch size of 12288, base learning rate of 0.03, and step LR schedule of 20 steps and $\gamma = 0.5$. We train using the LARS [6] optimizer with a weight decay value of 1e-4. The hyperparameters of the ResNeXt-101 Unified Visual Embedding fine-tuning are largely the same as [4], except the base learning rate is 0.03.

For pretraining, we use the Inception [5] random crop strategy, whereas for Unified Visual Embedding fine-tuning we apply horizontal mirroring, random crops, and color jitter to the resized images. For ImageNet fine-tuning we directly apply the data augmentation strategy that is specified in the original work on Vision Transformers [1].

For ablations on the sample count of the pretraining dataset, we linearly interpolate the training schedule length between the minimum and maximum value, i.e., 100 epochs on the 13M dataset and 2 epochs on the 1.3B dataset.

References


