Focal Transformer 模型自验报告

> 林丽森 finder4alex@163.com

1. 模型简介

1.1 网络模型结构简介

借助自注意力捕获短程和长程视觉依赖的能力，使得 `Vision Transformer` 及其变体在各种计算机视觉任务中显示出巨大的潜力，但由于二次计算开销问题，此类网络仍然面临较大挑战，尤其是对于高分辨率视觉任务（例如，目标检测）。

在`Focal Transformer`中，作者提出了`focal self-attention`，这是一种结合细粒度局部和粗粒度全局交互的新机制。使用这种新机制，每个标记以细粒度关注最近的周围标记，但以粗粒度关注远离的标记，因此可以高效且有效地捕获短程和长程视觉依赖。具体描述参见下图。

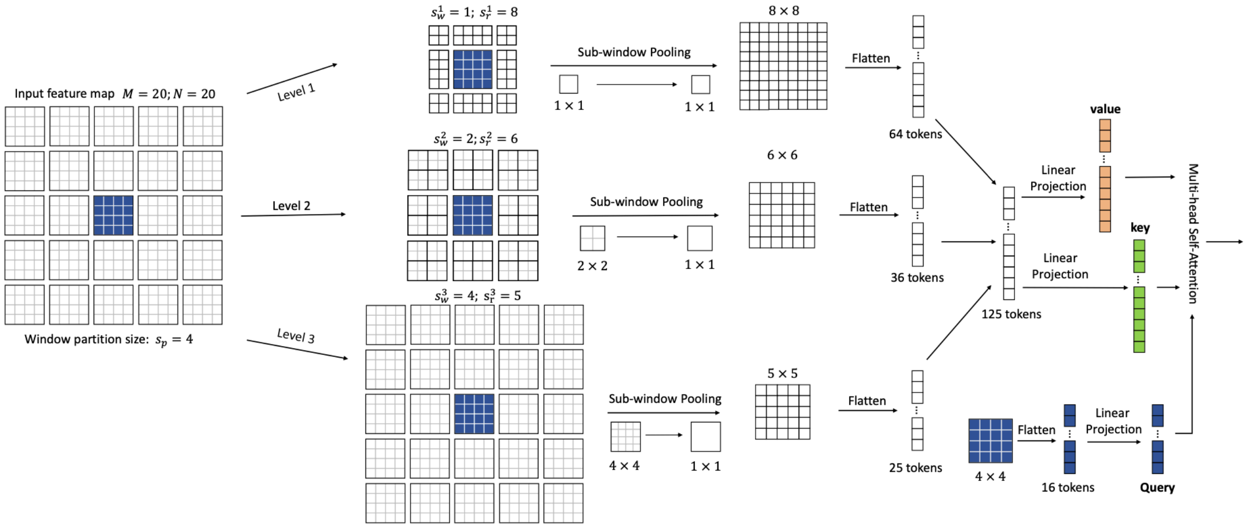


Figure 4: An illustration of our focal self-attention at window level. Each of the finest square cell represents a visual token either from the original feature map or the squeezed ones. Suppose we have an input feature map of size 20 × 20. We first partition it into 5 × 5 windows of size 4 × 4. Take the 4 × 4 blue window in the middle as the query, we extract its surroundings tokens at multiple granularity levels as its keys and values. For the first level, we extract the 8 × 8 tokens which are closest to the blue window at the finest grain. Then at the second level, we expand the attention region and pool the surrounding 2 × 2 sub-windows, which results in 6 × 6 pooled tokens. At the third level, we attend even larger region covering the whole feature map and pool 4 × 4 sub-windows. Finally, these three levels of tokens are concatenated to compute the keys and values for the 4 × 4 = 16 tokens (queries) in the blue window.

通过采用`focal self-attention`结构，文中提出了一种新的`Vision Transformer`模型变体，称为`Focal Transformer`，它在一系列公共图像分类和目标检测基准上实现了优于最先进的`Vision Transformer`的性能。详细网络结构参加下图。

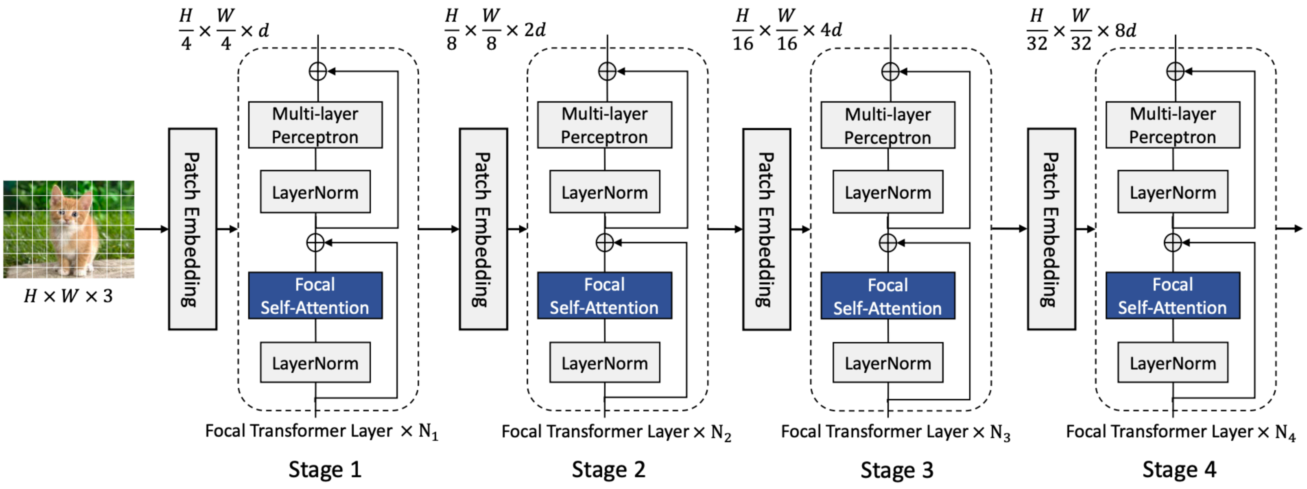

$\frac{H}{4}\times\frac{W}{4}\times d$
$\frac{H}{8}\times\frac{W}{8}\times 2d$
$\frac{H}{16}\times\frac{W}{16}\times 4d$
$\frac{H}{32}\times\frac{W}{32}\times 8d$
$H\times W\times 3$
Patch Embedding
Multi-layer Perceptron
LayerNorm
Focal Self-Attention
LayerNorm
Focal Transformer Layer × N_1
Focal Transformer Layer × N_2
Focal Transformer Layer × N_3
Focal Transformer Layer × N_4
Stage 1
Stage 2
Stage 3
Stage 4

Figure 2: Model architecture for our Focal Transformers. As highlighted in light blue boxes, our main innovation is the proposed focal self-attention mechanism in each Transformer layer.

1.2 数据集

> 所用数据集地址：https://git.openi.org.cn/kaierlong/imagenet2012_whole/datasets

使用训练及测试数据集如下：

```
使用的数据集：ImageNet2012

数据集大小：共1000个类、224*224彩色图像
训练集：共1,281,167张图像
测试集：共50,000张图像
数据格式：JPEG
注：数据在dataset.py中处理。
下载数据集，目录结构如下：
└─dataset
    ├─train                 # 训练数据集
    └─val                   # 评估数据集
```

1.3 代码提交地址

暂时提交在启智中，私有未开源。

仓库地址如下：https://git.openi.org.cn/finder4alex/Focal-Transformer.git

2. 代码目录结构说明

代码目录结构及说明如下：

```
.
├── LICENSE
├── README.md      // 说明文档
├── README_CN.md   //
├── eval.py        //
├── image          //
├── src
│   ├── args.py
│   ├── configs    // 超参数配置目录
│   │   ├── focalv2_small_useconv_patch4_window7_224.yaml
│   │   ├── focalv2_tiny_useconv_patch4_window7_224.yaml
│   │   └── parser.py
│   ├── data       // 数据加载及处理目录
│   │   ├── __init__.py
│   │   ├── augment
│   │   │   ├── __init__.py
│   │   │   ├── auto_augment.py
│   │   │   ├── custom_transforms.py
│   │   │   ├── mixup.py
│   │   │   └── random_erasing.py
│   │   ├── data_utils
│   │   │   ├── __init__.py
│   │   │   └── moxing_adapter.py
│   │   └── imagenet.py
│   ├── models     // 网络结构目录
│   │   ├── __init__.py
│   │   └── focal_transformer
│   │       ├── __init__.py
│   │       ├── focal_transformer_v2.py       // 1.5.1版本支持
│   │       ├── focal_transformer_v2_high.py  // 1.6.0及以上版本支持
│   │       ├── get_focal_transformer.py
│   │       └── misc.py
│   ├── tools      // 相关工具目录
│   │   ├── __init__.py
│   │   ├── callback.py
│   │   ├── cell.py
│   │   ├── criterion.py
│   │   ├── get_misc.py
│   │   ├── optimizer.py
│   │   └── schedulers.py
│   └── trainers   // 训练优化目录
│       ├── __init__.py
│       ├── model_ema.py
│       ├── train_one_step_with_ema.py
│       └── train_one_step_with_scale_and_clip_global_norm.py
└── train.py       // 训练文件
```

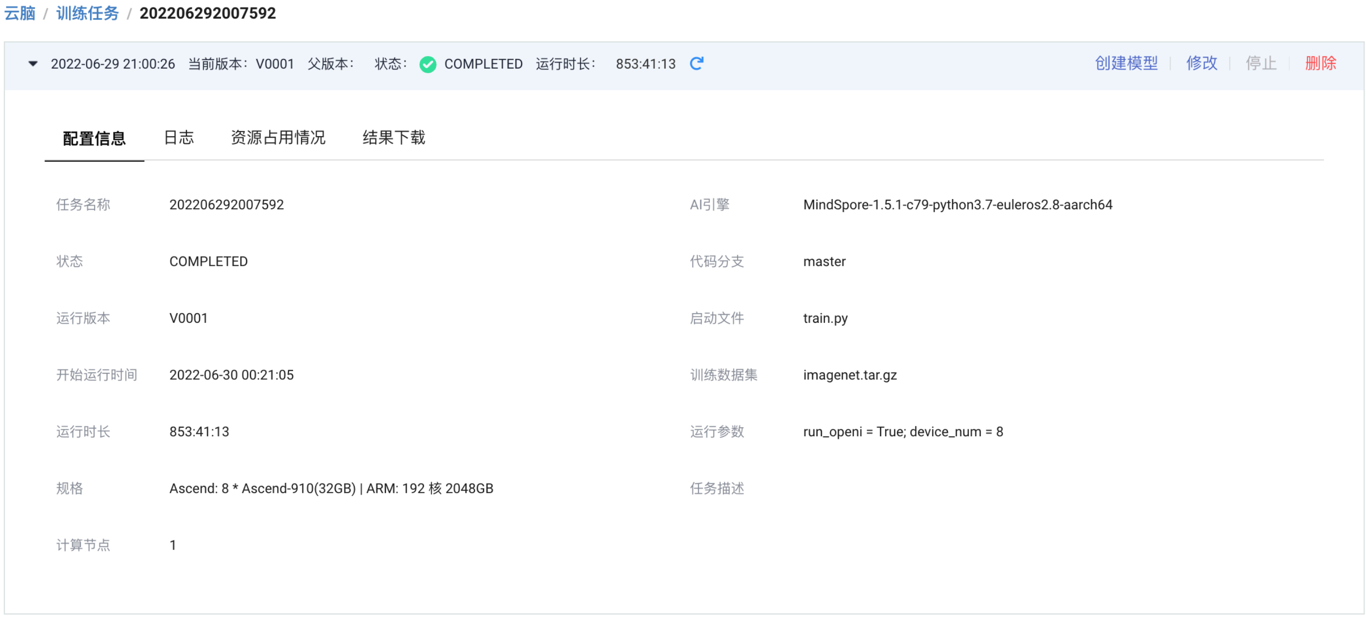
3. 自验结果（交付精度规格时需要补齐）

3.1 自验环境

软硬件环境如下：

- 启智AI引擎：MindSpore-1.5.1-c79-python3.7-euleros2.8-aarch64
- Ascend: 8 * Ascend-910(32GB) | ARM: 192 核 2048GB

详细环境配置参见下图：



云脑 / 训练任务 / 202206292007592

2022-06-29 21:00:26 当前版本：V0001 父版本： 状态： COMPLETED 运行时长： 853:41:13

任务名称	202206292007592	AI引擎	MindSpore-1.5.1-c79-python3.7-euleros2.8-aarch64
状态	COMPLETED	代码分支	master
运行版本	V0001	启动文件	train.py
开始运行时间	2022-06-30 00:21:05	训练数据集	imagenet.tar.gz
运行时长	853:41:13	运行参数	run_openi = True; device_num = 8
规格	Ascend: 8 * Ascend-910(32GB) | ARM: 192 核 2048GB	任务描述	
计算节点	1		

3.2 训练超参数

超参数配置如下：

> 其中data_url由启智平台实际数据地址替换，训练时替换。

```
# Architecture
arch: focalv2_small_useconv_patch4_window7_224

# ===== Dataset ===== #
data_url: ./data/imagenet
set: ImageNet
num_classes: 1000
mix_up: 0.8
cutmix: 1.0
auto_augment: rand-m9-mstd0.5-inc1
interpolation: bicubic
```

```
re_prob: 0.25
re_mode: pixel
re_count: 1
mixup_prob: 1.
switch_prob: 0.5
mixup_mode: batch
crop_ratio: 0.875


# ===== Learning Rate Policy ======== #
optimizer: adamw
lr_scheduler: cosine_lr
base_lr: 0.0005
min_lr: 0.000005
warmup_length: 20
warmup_lr: 0.00001
cool_length: 20
cool_lr: 0.000005


# ===== Network training config ===== #
amp_level: O1
keep_bn_fp32: True
beta: [ 0.9, 0.999 ]
is_dynamic_loss_scale: True
use_global_norm: True
clip_global_norm_value: 5.
enable_ema: True
ema_decay: 0.99992
loss_scale: 1024
weight_decay: 0.05
momentum: 0.9
label_smoothing: 0.1
epochs: 320
batch_size: 64


# ===== Hardware setup ===== #
num_parallel_workers: 16
device_target: Ascend

# ===== Model config ===== #
image_size: 224
patch_size: 4
drop_path_rate: 0.3
embed_dim: 96
depths: [ 2, 2, 18, 2 ]
num_heads: [ 3, 6, 12, 24 ]
window_size: 7
```

```
focal_stages: [ 0, 1, 2, 3 ]
focal_levels: [ 2, 2, 2, 2 ]
focal_windows: [ 7, 5, 3, 1 ]
expand_sizes: [ 3, 3, 3, 3 ]
focal_topK: 128
focal_pool: "fc"
```

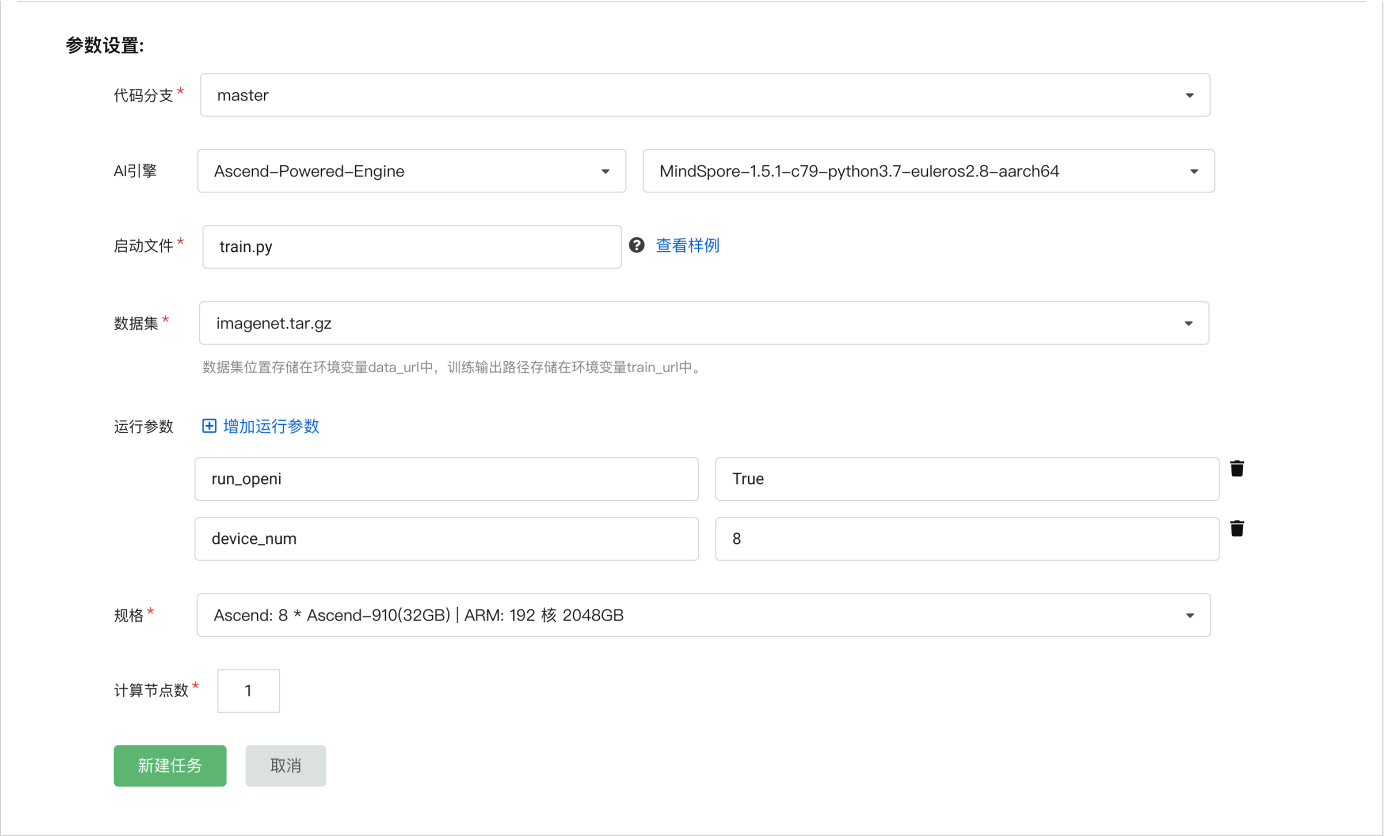
3.3 训练

3.3.1 如何启动训练脚本

训练如何启动：

- 启智平台

模型训练在启智平台完成，完整训练配置如下图所示：



参数设置:

代码分支 *: master

AI引擎: Ascend-Powered-Engine | MindSpore-1.5.1-c79-python3.7-euleros2.8-aarch64

启动文件 *: train.py

数据集 *: imagenet.tar.gz

数据集位置存储在环境变量data_url中，训练输出路径存储在环境变量train_url中。

运行参数:

run_openi | True

device_num | 8

规格 *: Ascend: 8 * Ascend-910(32GB) | ARM: 192 核 2048GB

计算节点数 *: 1

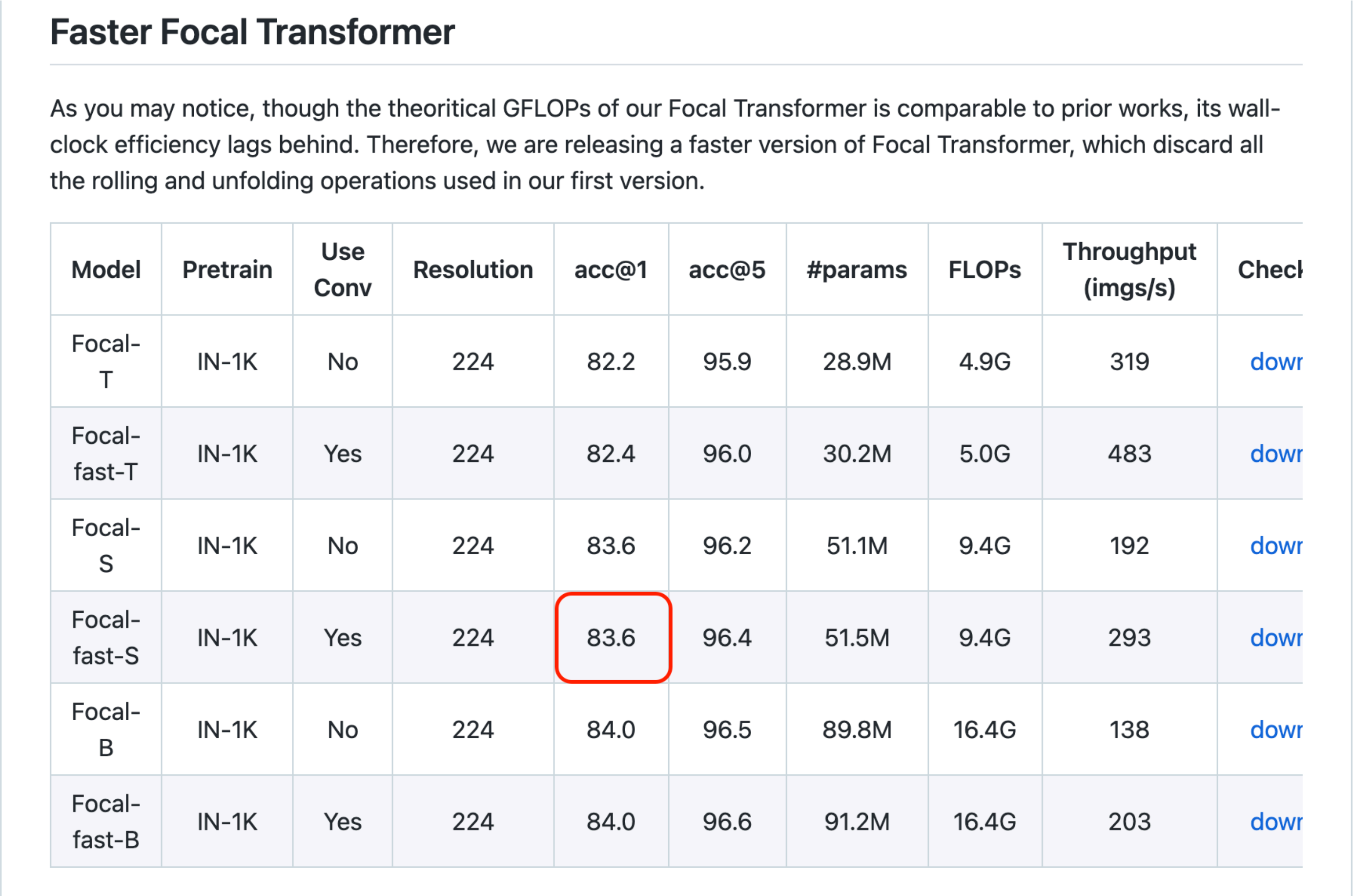
- 本地命令

如果需要本地训练，可以使用如下命令：

```
python3 train.py --run_openi=True --arch=focalv2_small_useconv_patch4_window7_224 --device_num=8
```

3.3.2 训练精度结果

- 论文精度如下:

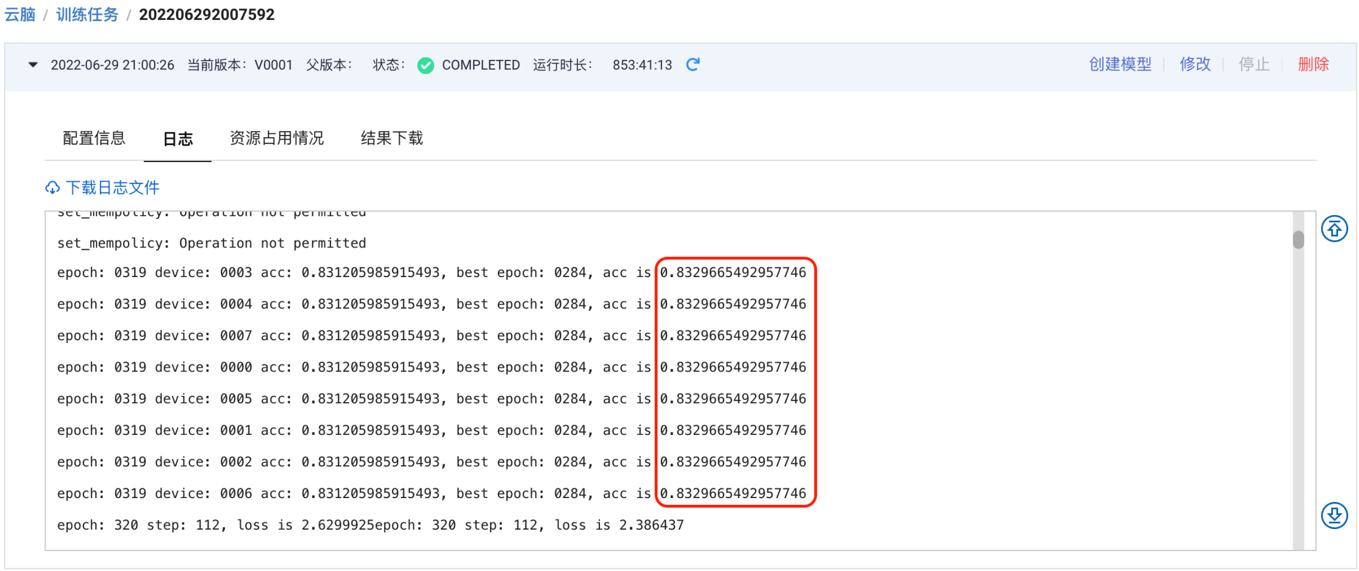


Faster Focal Transformer

As you may notice, though the theoritical GFLOPs of our Focal Transformer is comparable to prior works, its wall-clock efficiency lags behind. Therefore, we are releasing a faster version of Focal Transformer, which discard all the rolling and unfolding operations used in our first version.

Model	Pretrain	Use Conv	Resolution	acc@1	acc@5	#params	FLOPs	Throughput (imgs/s)	Chec
Focal-T	IN-1K	No	224	82.2	95.9	28.9M	4.9G	319	dow
Focal-fast-T	IN-1K	Yes	224	82.4	96.0	30.2M	5.0G	483	dow
Focal-S	IN-1K	No	224	83.6	96.2	51.1M	9.4G	192	dow
Focal-fast-S	IN-1K	Yes	224	83.6	96.4	51.5M	9.4G	293	dow
Focal-B	IN-1K	No	224	84.0	96.5	89.8M	16.4G	138	dow
Focal-fast-B	IN-1K	Yes	224	84.0	96.6	91.2M	16.4G	203	dow

- 复现精度如下:



云脑 / 训练任务 / 202206292007592
2022-06-29 21:00:26 当前版本: V0001 父版本: 状态: COMPLETED 运行时长: 853:41:13
创建模型 | 修改 | 停止 | 删除
配置信息
日志
资源占用情况
结果下载
下载日志文件

```
set_mempolicy: operation not permitted
set_mempolicy: Operation not permitted
epoch: 0319 device: 0003 acc: 0.831205985915493, best epoch: 0284, acc is 0.8329665492957746
epoch: 0319 device: 0004 acc: 0.831205985915493, best epoch: 0284, acc is 0.8329665492957746
epoch: 0319 device: 0007 acc: 0.831205985915493, best epoch: 0284, acc is 0.8329665492957746
epoch: 0319 device: 0000 acc: 0.831205985915493, best epoch: 0284, acc is 0.8329665492957746
epoch: 0319 device: 0005 acc: 0.831205985915493, best epoch: 0284, acc is 0.8329665492957746
epoch: 0319 device: 0001 acc: 0.831205985915493, best epoch: 0284, acc is 0.8329665492957746
epoch: 0319 device: 0002 acc: 0.831205985915493, best epoch: 0284, acc is 0.8329665492957746
epoch: 0319 device: 0006 acc: 0.831205985915493, best epoch: 0284, acc is 0.8329665492957746
epoch: 320 step: 112, loss is 2.6299925epoch: 320 step: 112, loss is 2.386437
```

- 精度结果对比
    - 论文精度为：83.6
- 复现精度为：83.296（最优值）
- 复现精度误差为：(83.6 - 83.296) / 83.6 ≈ 0.36%

3.4 模型推理

推理命令如下：

```
python3 eval.py --config=src/configs/focalv2_small_useconv_patch4_window7_224.yaml --
pretrained={ckpt_path} --device_id={device_id} --device_target={device_target} --
data_url={data_url}
```

4. 参考资料

4.1 参考论文

- Focal Self-attention for Local-Global Interactions in Vision Transformers

4.2 参考git项目

- microsoft/Focal-Transformer

4.3 参考文献

- 新注意力！Focal Transformer：ViT中局部-全局交互的Focal自注意力