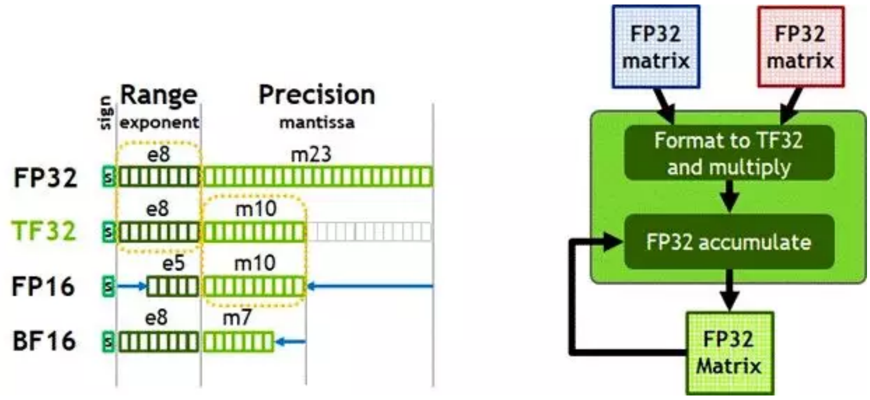
2022 年 4 月 18 日

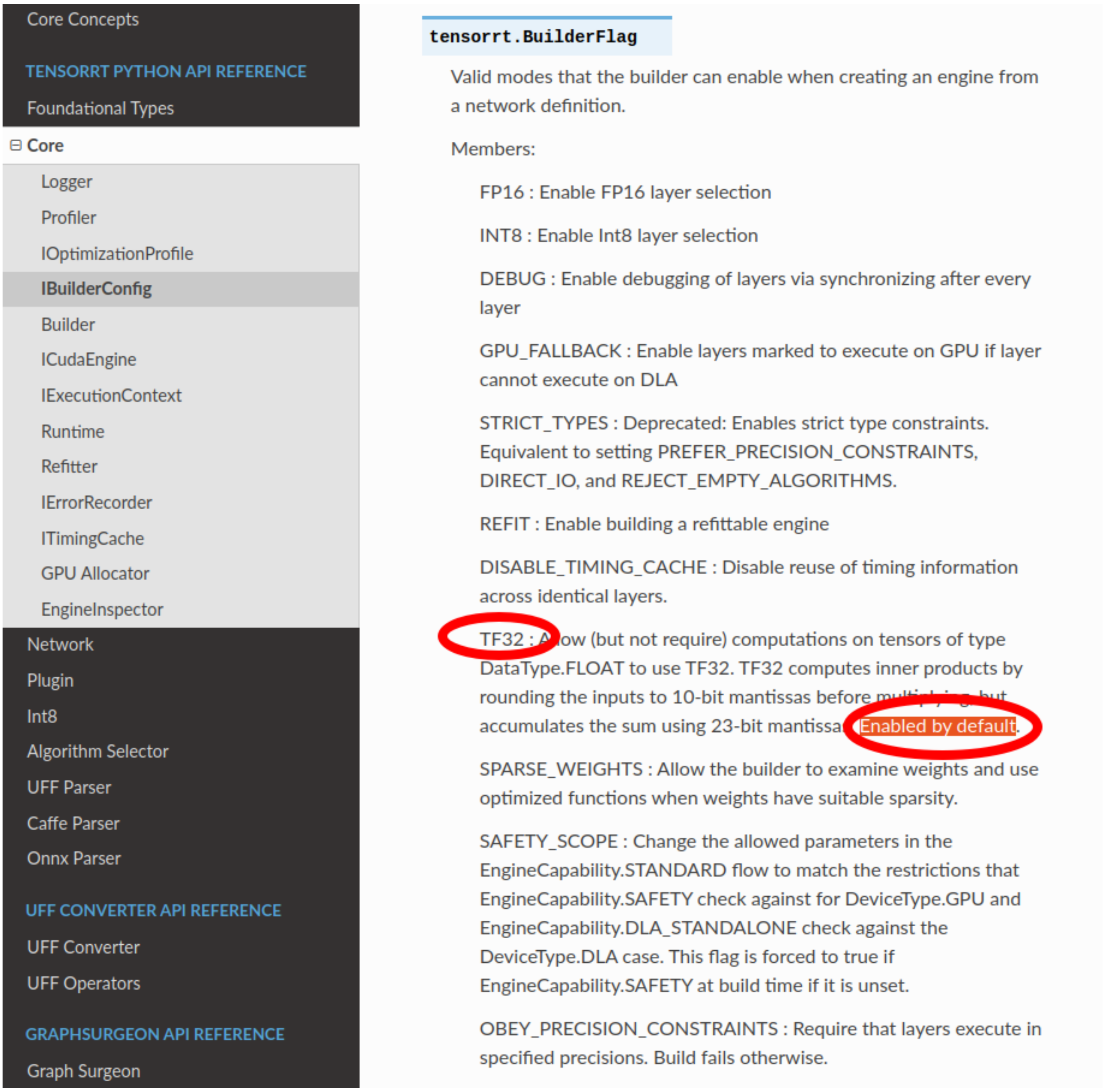
- 目前很多同学们完成了模型构建，但是苦于误差问题而不能得到较好的分数，也限制了进一步优化，所以今天我们给出一些提示
- TF32
  - TF32 采用了与 FP16 相同的 10 位尾数和与 FP32 相同的 8 位指数
  - 位分布和计算过程:



- **Ampere 及以上的GPU具有原生 TF32 计算能力**
- **TensorRT 默认开启 TF32**
  - 验证：在 docker 中用 python 运行下面这段代码，观察默认的 config.flags 是啥

```
import tensorrt as trt
logger = trt.Logger(trt.Logger.ERROR)
builder = trt.Builder(logger)
network = builder.create_network(1 << int(trt.NetworkDefinitionCreationFlag.EXPLICIT_BATCH))
profile = builder.create_optimization_profile()
config = builder.create_builder_config()
print("default Flag:",config.flags)
print("TF32 Flag:",1<<int(trt.BuilderFlag.TF32))
```

- TensorRT 与 TF32 的文档说明 link
  - 截图:



Core Concepts

TENSORRT PYTHON API REFERENCE

Foundational Types

⊟ Core

- Logger
- Profiler
- IOptimizationProfile
- IBuilderConfig
- Builder
- ICudaEngine
- IExecutionContext
- Runtime
- Refitter
- IErrorRecorder
- ITimingCache
- GPU Allocator
- EngineInspector

Network

Plugin

Int8

Algorithm Selector

UFF Parser

Caffe Parser

Onnx Parser

UFF CONVERTER API REFERENCE

UFF Converter

UFF Operators

GRAPHSURGEON API REFERENCE

Graph Surgeon

`tensorrt.BuilderFlag`

Valid modes that the builder can enable when creating an engine from a network definition.

Members:

FP16 : Enable FP16 layer selection

INT8 : Enable Int8 layer selection

DEBUG : Enable debugging of layers via synchronizing after every layer

GPU_FALLBACK : Enable layers marked to execute on GPU if layer cannot execute on DLA

STRICT_TYPES : Deprecated: Enables strict type constraints. Equivalent to setting PREFER_PRECISION_CONSTRAINTS, DIRECT_IO, and REJECT_EMPTY_ALGORITHMS.

REFIT : Enable building a refittable engine

DISABLE_TIMING_CACHE : Disable reuse of timing information across identical layers.

TF32 : Allow (but not require) computations on tensors of type DataType.FLOAT to use TF32. TF32 computes inner products by rounding the inputs to 10-bit mantissas before multiplying, but accumulates the sum using 23-bit mantissas. Enabled by default.

SPARSE_WEIGHTS : Allow the builder to examine weights and use optimized functions when weights have suitable sparsity.

SAFETY_SCOPE : Change the allowed parameters in the EngineCapability.STANDARD flow to match the restrictions that EngineCapability.SAFETY check against for DeviceType.GPU and EngineCapability.DLA_STANDALONE check against the DeviceType.DLA case. This flag is forced to true if EngineCapability.SAFETY at build time if it is unset.

OBEY_PRECISION_CONSTRAINTS : Require that layers execute in specified precisions. Build fails otherwise.

- 说人话版：
  - 测评服务器 A30 默认开启 TF32，如果要用 FP32，需要手动 `config.flags = config.flags & ~(1 << int(trt.BuilderFlag.TF32))`，或用其他 flag 覆盖它
  - 大家本地显卡（比如我的GTX1070）如果不支持 TF32，则会忽略掉该 flag，默认使用 FP32
  - 速度 FP16 > TF32 > FP32，精度 FP16 < TF32 < FP32
  - 模型的优化瓶颈和误差控制瓶颈并不是在 TF32 / FP32，但如果大家不放心，可以先用 FP32 保证结果精度然后进行其他优化。其他优化完成后可能 TF32 或者 FP16 的精度也能满足要求了