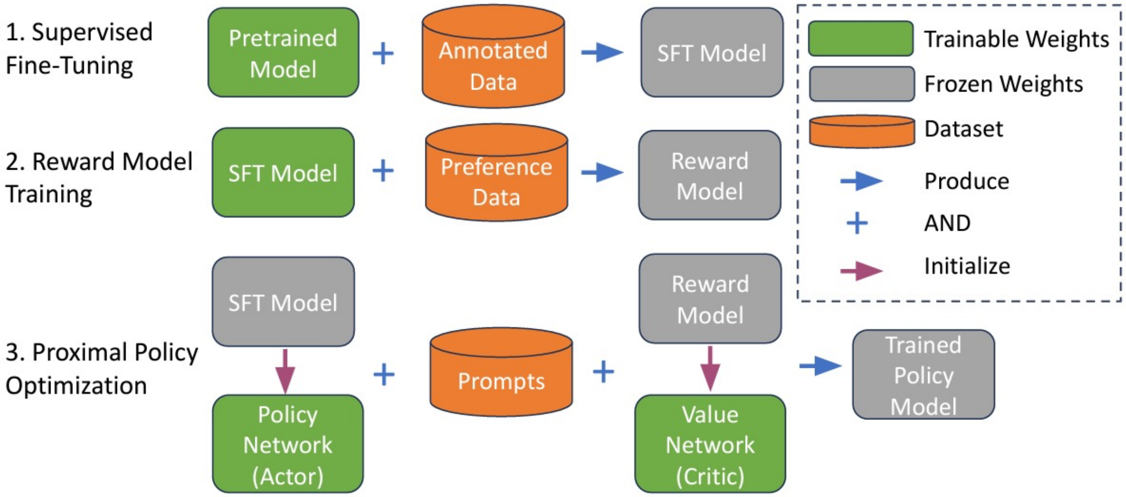
SGL

Update Weights From Distributed

Chenyang Zhao

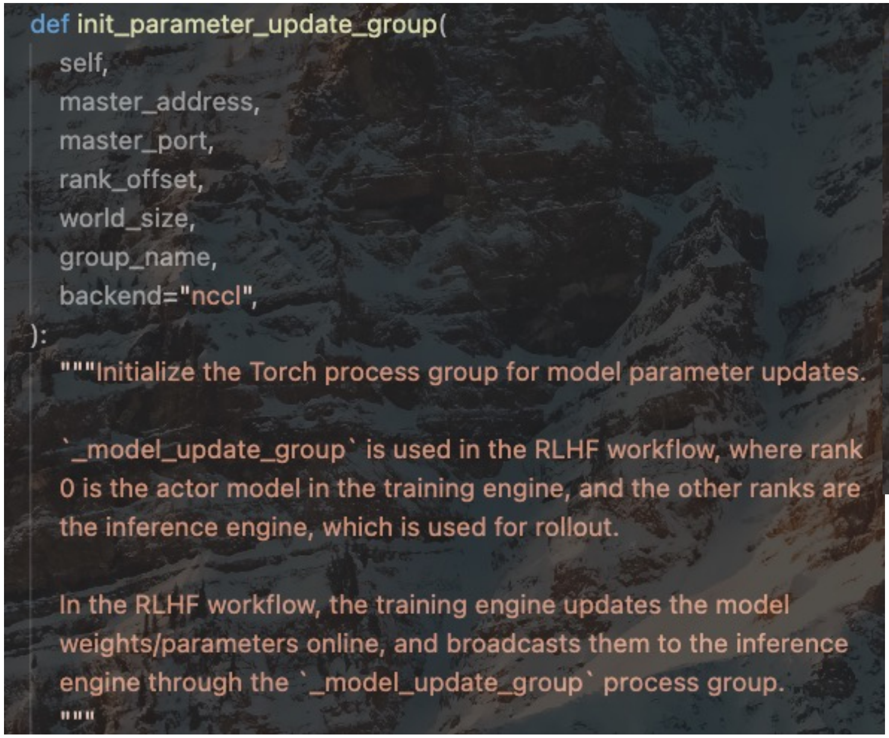
Update Weights In RLHF



RLHF needs serval thousand rounds of weights update and rollout / inference.

1. Relaunch the engine thousands of time?
2. Update weights from disk thousands of time?
3. Update weights directly from torch.distributed.

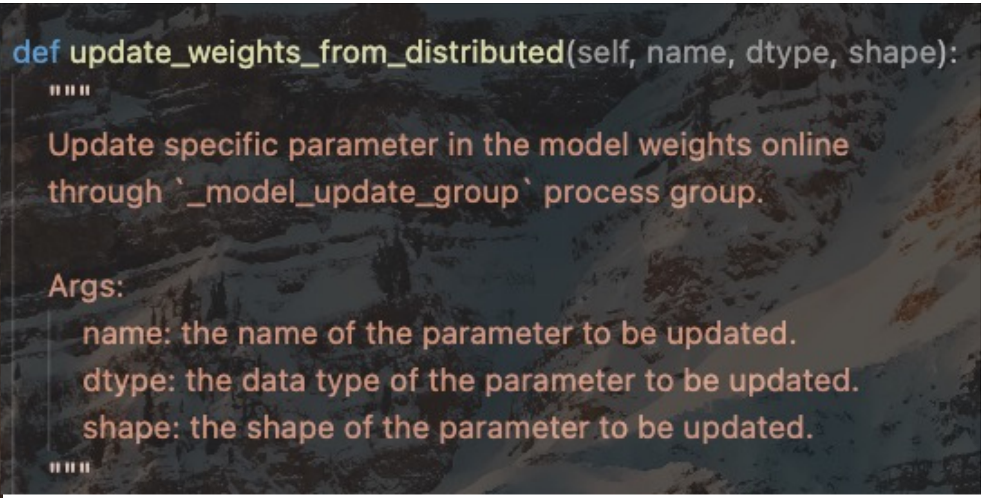
Update weights from torch.distributed



```
def init_parameter_update_group(
    self,
    master_address,
    master_port,
    rank_offset,
    world_size,
    group_name,
    backend="nccl",
):
    """Initialize the Torch process group for model parameter updates.

    `_model_update_group` is used in the RLHF workflow, where rank
    0 is the actor model in the training engine, and the other ranks are
    the inference engine, which is used for rollout.

    In the RLHF workflow, the training engine updates the model
    weights/parameters online, and broadcasts them to the inference
    engine through the `_model_update_group` process group.
    """
```



```
def update_weights_from_distributed(self, name, dtype, shape):
    """
    Update specific parameter in the model weights online
    through `_model_update_group` process group.

    Args:
        name: the name of the parameter to be updated.
        dtype: the data type of the parameter to be updated.
        shape: the shape of the parameter to be updated.
    """
```

Current Usage And Performance

See: test/srt/test_update_weights_from_distributed.py

1. Launch HF model on Rank 0;
2. Launch SGLang Engine/Runtime on Rank 1; DP is supported.
3. Broadcast and load;

On our H100 cluster, 1B/8B llama takes less than 0.5s.