在6月7日前尽力完成以下任务，暂定于6月7日进行汇报，汇报时长10分钟，QA 5分钟，具体安排届时通知。任务难度较大，请不要以彻底完成为目标，完成多少部分说多少部分即可。

1. 阅读Megatron-LM[1]论文和源代码[2]，重点关注其中的流水线并行（Pipeline Model Parallelism）。
2. **如果有高性能GPU（显存>20G）**：配置Megatron-LM运行环境，部署Megatron-LM。实现GPT2-small模型（12层decoder，hidden size=768）的预训练（建议使用WikiText2数据集），参考运行脚本（pretrain_gpt_distributed.sh），如果有多张GPU请开启流水线并行，参考（pretrain_gpt_distributed_with_mp.sh）。保存运行的log。
3. **如果没有高性能GPU**：后续实验均采用朴素CNN、VGG（参考[6]中的模型结构）
4. 阅读冻结训练相关论文[3-5]了解冻结训练技术，在Megatron-LM或者朴素CNN、VGG上实现基础的冻结训练（冻结固定层数）。给出实验结果（loss随iteration变化曲线、每个iteration实际耗时曲线），并与基线（正常训练）进行对比。

   Tips：冻结一个层的实现可以将其所有参数的requires_grad标志设置为false，以从梯度计算中排除子图[7]。
5. 在任务2的基础上复现AutoFreeze算法[4]，给出实验结果（冻层层数随iteration变化曲线，loss随iteration变化曲线、每个iteration实际耗时曲线）。

   Tips：每层梯度的抓取可以使用torch中的钩子函数torch.nn.Module.register_full_backward_pre_hook或者参考AutoFreeze的官方实现。
6. 汇报内容：阐述实验环境及数据集的准备过程；讲解对Megatron-LM代码框架的理解、冻结功能的实现；实验结果图片；整个任务期间遇到的问题与解决方案等等。

前置知识/技能：pytorch，linux bash，docker，git，数据可视化

参考资料：

[1]Narayanan D, Shoeybi M, Casper J, et al. Efficient large-scale language model training on gpu clusters using megatron-lm[C]//Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. 2021: 1-15.

[2]https://github.com/NVIDIA/Megatron-LM

[3]Brock A, Lim T, Ritchie J M, et al. Freezeout: Accelerate training by progressively freezing layers[J]. arXiv preprint arXiv:1706.04983, 2017.

[4] Liu Y, Agarwal S, Venkataraman S. Autofreeze: Automatically freezing model blocks to accelerate fine-tuning[J]. arXiv preprint arXiv:2102.01386, 2021.

[5] Wang Y, Sun D, Chen K, et al. Egeria: Efficient dnn training with knowledge-guided layer freezing[C]//Proceedings of the Eighteenth European Conference on Computer Systems. 2023: 851-866.

[6]https://github.com/tntnnlrw/federated-learning/blob/master/neural_nets.py

[7] PyTorch. PyTorch autograd mechanics. https://pytorch.org/docs/stable/notes/autograd.html