深度学习最优化项目任务书

0. 作业要求

1. 本作业为小组作业，原则上每组 3 人（以组队文档为准）。
2. 本次作业包含一次课堂展示、一份实验代码、一份实验报告。
3. 课堂展示要求：
    a. 请各组在 2023 年 12 月 26 日上午 11:59 前将展示 ppt/pdf 发送至邮箱 pkudlopt@163.com，邮件主题命名为“【课堂展示】第 XX 组-姓名 1-姓名 2-姓名 3”，附件命名为“第 XX 组展示-姓名 1-姓名 2-姓名 3.pptx/pdf”。
    b. 课堂展示时间为 2023 年 12 月 26 日上课时间，每个小组将有 7 分钟时间上台演示自己的项目作业并回答问题，请各小组严格控制时间。
    c. 课堂展示时需记录教师或同学们的提问与反馈，并根据提问与反馈进一步完善项目实验和实验报告
    d. 课堂展示时，可以不必完成大作业中的所有任务。当有些任务点没有完成时，需要给出当前的进展，以及遇到的困难。未完成的任务必须要在提交实验报告前完成。
4. 代码与实验报告要求：
    a. 1 份实验代码，1 份实验报告。请各小组在 2024 年 1 月 14 日 23:59 前将实验代码压缩包、实验报告 pdf 随附件发送至邮箱 pkudlopt@163.com，邮件主题命名为“【上机作业】第 XX 组-姓名 1-姓名 2-姓名 3”，附件命名为“第 XX 组代码-姓名 1-姓名 2-姓名 3.zip”、“第 XX 组报告-姓名 1-姓名 2-姓名 3.pdf”。
    b. 代码要求：1）除主要代码文件外，请在代码文件夹内建立文件“readme.txt”并简单阐述代码的运行逻辑，建议使用 Jupyter Notebook（.ipynb）组织代码。2）代码组织要规范，注释要清晰。 3）请不要在上传的代码文件夹内存放较大（大小超过 50M）的数据集，若为自己选择的数据集或与作业指定的官方数据集有差异，请先上传至北大网盘并将共享链接随邮件发出。
    c. 实验报告要求：1）实验报告中需要包含项目背景、项目目标、实验设计方式、实验参数、实验结果等部分。请如实汇报实验结果(实验结果有可能会与理论相矛盾)，不得捏造数据。2）请在实验报告中描述每名成员的分工与贡献。3）报告部分应简明扼要、逻辑清晰，请不要在报告正文中大量粘贴程序代码。4）建议使用 latex 或者 markdown 编辑实验报告，中英文均可。

d. 优化器：建议大家自行编写优化器(如 SGD, Momentum SGD, Adam 等)，但也可以调用 Pytorch 或 TensorFlow 中自带的优化器。如果是后者，必须要在实验报告中说明。如果项目任务中注明需要自行编写优化算法，则需要严格遵守任务书规定。

5. 大作业成绩由项目展示情况、代码规范与注释合理性、实验报告质量三方面综合评定。由于项目难度有细微区别，评定成绩时也会考虑项目难度的影响。
6. 当对任务理解不清楚时，请及时联系教师或者助教。
7. 每项任务最多支持两队同时独立参与。当某项目有两支队伍参与时，评定成绩时难免可能会对两队进行比较。所以建议同学们尽量分散选择项目。

1. SGD and Momentum SGD

作业描述：

在课程中，我们讨论了 momentum SGD 等价于学习率增大后的标准 SGD 算法，参见 Lecture 8 的 Slides. 请在如下任务中对比学习率为 gamma 的 momentum SGD 以及学习率为 gamma/(1-beta) 的 standard SGD 的 training loss 曲线与 test accuracy 曲线，其中 beta 为动量系数。以上两种优化算法均需要自行编写。

- 在实际数据集(可在 LIBSVM [R1-2]中自行挑选)中做线性回归任务，对两种算法的结果进行对比与分析
- 在 MNIST 数据集中训练 LeNet 神经网络，对两种算法的结果进行对比分析
- 在 CIFAR10 的数据集中训练 Resnet-18 神经网络，对两种算法结果进行对比
- 在 MNIST 的数据集中测试不同的 constant learning rate 对 SGD 的收敛曲线及 test accuracy 的影响
- 在 MNIST 的数据集中测试不同的 batch-size 对 SGD 的收敛曲线及 test accuracy 的影响

参考文献：

[R1-1] Kun Yuan et.al., *On the Influence of Momentum Acceleration on Online Learning*, JMLR 2016.
[R1-2] Chih-Chung Chang and Chih-Jen Lin, *LIBSVM : a library for support vector machines*. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

2. Adam

作业描述：

Adam 是当前主流的自适应梯度算法。请在如下任务中比较 Momentum SGD, Adam, AdamW 三种算法的 training loss 与 test accuracy 曲线。

- 在 MNIST 数据集中训练 LeNet 神经网络，测试以上三种算法的性能
- 在 CIFAR10 数据集训练 Resnet-18 神经网络，测试以上三种算法的性能
- 在某种自然语言处理任务的训练或微调中，测试以上三种算法的性能。例如，对 BERT 模型进行微调[R2-1, R2-2]

- 在以上任务中至少选择一种来测试 Adam 参数 β_1, β_2 以及学习率 γ的影响

参考文献：

[R2-1] Dorottya Demszky et.al., *GoEmotions: A Dataset of Fine-Grained Emotions,* ACL 2020
[R2-2] 参考代码: https://github.com/monologg/GoEmotions-pytorch

3. Anderson Acceleration

作业描述：

在课程中，我们介绍了 Anderson 加速算法，其在一些任务上有着不俗的表现。请在如下任务中比较 GD with Polyak momentum, GD with Nesterov momentum, Aderson acceleration 三种算法在达到 1e-7 求解精度时的收敛曲线以及 CPU 计算时间。以上三种优化算法均需要自行编写。

- 在实际数据集(可在 LIBSVM [R1-2]中自行挑选)中做线性回归任务
- 在实际数据集(可在 LIBSVM [R1-2]中自行挑选)中做逻辑回归任务
- 探索历史点数 m 对 Anderson 算法的影响，如对算法收敛速度，内存开销，CPU 计算时间的影响

参考文献:

[R3-1] V. Mai and M. Johansson, *Anderson acceleration of proximal gradient methods*, in International Conference on Machine Learning. PMLR, 2020, pp. 6620–6629.

4. Adaptive SGD

作业描述：

在课程中, 我们介绍了一系列梯度自适应算法。请在如下任务中比较 AdaGrad, RMSProp, Adam, Momentum SGD 以及 Standard SGD 这 5 种方法

- 构造一个 2 维或 3 维的非凸问题, 利用以上 5 种算法进行求解, 并参考[R4-1]中的展示效果, visualize 所有算法的整个收敛过程。注意，该任务中以上所有算法均需要手动编写。
- 在实际数据集(可在 LIBSVM [R1-2]中自行挑选)中做逻辑回归任务，测试以上 5 种算法的 traning loss 和 test accuracy 曲线
- 在 CIFAR10 数据集训练 Resnet-18 神经网络, 测试以上 5 种算法的 traning loss 和 test accuracy 曲线

参考文献:

[R4-1] https://imgur.com/a/Hqolp (需要科学上网)

5. Gauss smoothness and sphere smoothness

作业描述：

在课程中，我们介绍了多种零阶优化算法。请手动实现基于 Finite-difference, Gauss smoothing, Sphere smoothing 的零阶优化算法，并完成如下任务：

- 在实际数据集(可在 LIBSVM [R1-2]中自行挑选)中做线性回归任务，比较以上三种算法的收敛精度、收敛速度、以及到达相同精度时对函数值的采样数
- 针对 Gauss smoothness，探究不同光滑半径的选取对算法性能的影响
- 针对 Sphere smoothness，使用在同一点采多个随机方向的近似梯度并取平均的方法（minibatch），测试是否能对算法性能有显著提升

6. Fine-Tuning DNN Models with Just Forward Passes

作业描述：

在最近的一篇文献[R6-1]中，作者声称使用零阶随机优化算法 MeZO(也即只使用前向传播)可以显著节省内存，减少单次算法迭代的时间，并且不会对算法性能造成显著影响。本任务将理解 MeZO 算法并对其进行复现。

- 理解 MeZO 算法，并复现 MeZO 对 RoBERTa-large 模型在 SST-2 数据集中进行 finetune 的实验结果。具体而言，复现文献[R6-1]里 Table 18 中第二列 k=16 时 MeZO 与 FT 的实验结果。注意：可以直接使用文献[R6-1]中提供的代码。
- 在上一点任务中，对比 MeZO 与 FT 的单次迭代的时间，占用内存的大小，以及训练至 Table 18 所列写的结果时需要的总 epoch 数量与训练的总时间。
- 手动实现 MeZO 算法，并测试其基于 Cifar10 数据集对 ResNet 网络训练的效果 。

参考文献:

[R6-1] Sadhika Malladi et. al., *Fine-Tuning Language Models with Just Forward Passes,* NeurIPS 2023. https://arxiv.org/pdf/2305.17333.pdf

7. Automatic Mixed Precision

作业描述：

混合精度训练是节省模型训练内存，提高训练速度的关键技术。本项任务将探索混合精度训练的相关特性：

- 学习 AMP 代码库的使用方法,参考[R7-1]。在 CIFAR10 数据集训练 Resnet-18 神经网络时，引入 AMP 进行混合精度训练。比较使用全精度与混合精度的算法占用内存、训练时间、training loss 与 test accuracy.
- 为了更好体现混合精度训练的效果，测试不同 batch-size 下全精度训练与混合精度训练的对比效果。(batch-size 越大，可能混合精度的速度与内存优势越明显)
- 测试 loss scaling 对混合精度训练的影响。探索如果没有 loss scaling，混合精度训练是否还会工作。

参考文献:

[R7-1] Automatic mixed precision.
https://pytorch.org/tutorials/recipes/recipes/amp_recipe.html

8. 8-bit Adam optimizer

作业描述：

8 bit Adam optimizer 是一种更高效的混合精度训练技巧。其作者声称可以节省 75%的内存开销。本项任务将探索 8-bit Adam optimizer 的相关特性:

- 阅读文献[R8-1]，学习使用代码[R8-2]。在某种自然语言处理任务的训练或微调中，对比 standard Adam 与 8 bit Adam 的效果，如内存开销、训练时间、training loss 与 test accuracy 等。
- 测试 block-wise quantization 对 8 bit Adam 训练的影响。探索如果没有 block-wise quantization, 8 bit Adam 是否还会工作。

参考文献:

[R8-1] Tim Dettmers et. al., *8-BIT OPTIMIZERS VIA BLOCK-WISE QUANTIZATION,* ICLR 2022

[R8-2] https://github.com/TimDettmers/bitsandbytes

9. Variance Reduction

作业描述：

方差缩减是随机算法的一种重要思想。它可以在选取固定学习率时，让随机梯度的方差自动衰减为 0，进而起到加速算法的效果。请自行编写 Standard SGD, SAGA，与 SVRG 三种算法，并在以下任务中做性能测试。

- 在实际数据集(可在 LIBSVM [R1-2]中自行挑选)中做线性回归任务，在使用相同学习率下，比较以上三种算法的收敛性质的表现。此外，画出三种算法随机梯度方差随算法步数增大时的变化情况。改进 SAGA 算法，使得其额外增加的内存不依赖于问题变量的维度。
- 在实际数据集(可在 LIBSVM [R1-2]中自行挑选)中做逻辑回归任务，在使用相同学习率下，比较以上三种算法的 training loss 和 test accuracy 曲线。此外，画出三种算法随机梯度方差随算法步数增大时的变化情况。改进 SAGA 算法，使得其额外增加的内存不依赖于问题变量的维度。
- 在 FashinMNIST 数据集中选择合适的网络使用以上三种算法训练 Resnet-18 神经网络，比较 training loss 和 test accuracy 曲线。

参考文献:

[R9-1] Rie Johnson and Tong Zhang, *Accelerating Stochastic Gradient Descent using Predictive Variance Reduction*, NIPS 2013

[R9-2] Aaron Defazio, et. al., *SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives*, NIPS 2014

10. Adversarial Learning

作业描述：

对抗学习是增强深度神经网络的重要方法。本项任务将探索攻击与防御算法的性能效果。

- 针对 MNIST 数据集的 LeNet 神经网络，自行编写基于 FGSM 与 PGD 的攻击算法，测试

LeNet 对这两种算法的攻击效果；测试扰动误差对攻击算法的性能影响。

- 针对 MNIST 数据集的 LeNet 神经网络, 自行编写基于 FGSM 与 PGD 的防御算法(Lecture 10-1 中的 Algorithm 1, 2)，测试防御算法对 LeNet 的稳定性是否带来提升。
- 测试"Free"对抗学习(Algorithm 3)和"Fast"对抗学习算法(Algorithm 4)针对 MNIST+LeNet 的性能。

参考文献:

[R10-1] A. Shafahi et. al., *Adversarial training for free*, NIPS 2019

[R10-2] E. Wong, et. al., *Fast is better than free: Revisiting adversarial training*, NIPS 2020

11.Gradient Clipping

作业描述：

Gradient clipping 被广泛用于解决深度神经网络中的梯度爆炸问题。并且其在(L0, L1)-smooth 假设下被证明比标准梯度下降有更好的性能表现。请按照以下要求探索 Gradient clipping 的性质：

- 验证(L0,L1)-smooth 假设。复现文献[R11-1]中的 Figure 1(a)和 Figure 1(b)
- 在 LSTM 任务中验证(L0,L1)-smooth 假设，也即复现文献[R11-2]中的 Figure 1.
- 针对上述两种任务(toy 问题和 LSTM 问题)，自行编写 Gradient clipped SGD 算法，并与标准 SGD 算法的 training loss 或 test accuracy 进行比较。

参考文献:

[R11-1] B. Zhang et. al., *Improved Analysis of Clipping Algorithms for Non-convex Optimization*, NeurIPS 2020

[R11-2] J. Zhang, et. al., *Why gradient clipping accelerates training: A theoretical justification for adaptivity*, ICLR 2020

12.Sampling strategy

作业描述：

采样顺序对 SGD 算法的收敛性质有很大影响。我们考虑如下 5 种采样策略:

1. 有放回的均匀采样
2. importance sampling (可使用[R12-1]或[R12-2]的算法)
3. reshuffling-once，即只在采样前对数据 shuffle 一次，之后一直按照该顺序进行无放回采样
4. random-reshuffling，即每个 epoch 都需要对数据 shuffle 一次
5. 按类别采样。即每个 epoch 中，先无放回采第一类数据，接着采第二类数据，一直到采最后一类的数据。每个 epoch 都重复该过程，类间数据可以 random-reshuffle.

请在以下任务中对比以上 5 中采样策略的 training loss 和 test accuracy

- 在某实际数据集(可在 LIBSVM [R1-2]中自行挑选)中做逻辑回归任务

- 在 MNIST 中训练 LeNet
- 在 Cifar10 中训练 ResNet

参考文献:

[R12-1] Kun Yuan et. al., *Stochastic gradient descent with finite samples sizes*, IEEE MLSP, 2016

[R12-2] Peilin Zhao and Tong Zhang, *Stochastic Optimization with Importance Sampling for Regularized Loss Minimization*, ICML 2015

13.Learning to Optimize

作业描述：

Proximal gradient descent 是求解非光滑问题的重要方法。利用神经网络来加速优化算法也是当前学术界的前沿热点。本项目将复现利用深度神经网络来加速 proximal gradient descent 的算法。

- 学习文献[R13-1]中的 FISTA 算法。FISTA 算法是 proximal gradient descent (也被称为 ISTA) 算法的 Nesterov 加速版本
- 基于文献[R13-2]中第 4.1 节的实验设定, 自行编写 ISTA,FISTA, LISTA, LISTA-CP 四种方法, 复现图 3 的实验结果。
- 基于 ISTA,FISTA,LISTA-CP,复现类似图 4 的结果，其中对于 FISTA 算法可以不考虑自适应罚参数的情形。

参考文献:

[R13-1] A. Beck and M. Teboulle, *A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems*, SIAM J. Imaging Science, 2009

[R13-2] X. Chen, et. al., *Theoretical Linear Convergence of Unfolded ISTA and its Practical Weights and Thresholds,* NeurIPS 2018.

14.SGD Generalization and Flat Minimum

作业描述：

在课程中，我们讨论了 SGD 对损失景观更为“平坦”的最小值的偏好，并以此解释了 SGD 相比 GD 在泛化性上的优越性。请参考[R14-1]或其它参考资料，完成下列探究任务：

- A toy model：参考[R14-1]的 Section 5.1，在二维情形下构造一组损失函数，并针对这一损失函数，选取不同的学习率和迭代起始点，分别尝试 GD 与 SGD 训练。绘制损失函数的等高线图，并进一步绘制不同情形的迭代轨迹图，以观察不同学习率下 GD 和 SGD 是否能从更为“尖锐”的最小值逃逸到相对”平坦“的最小值，并以此探究 SGD 与 GD 迭代过程中学习率与所收敛到的稳定点损失景观的关系。最好能够以动图的方式来 visualize 这个逃脱过程，visualization 可以参考[R14-3]
- 对 FashionMNIST 选取合适的神经网络，分别进行 GD 与 SGD 训练，通过比较 test accuracy 体现 SGD 相比 GD 在泛化性能上的优越性。可以参考[R14-1]中图 3 的实验设定。

- 对于上一问的模型，尝试在使用 GD 进行充分训练再切换为 SGD 训练，观察是否能得到更高的 test accuracy，可以参考 Lecture7-2 中第 6 页幻灯片中的图和[R14-2]中的图 1 与 Section 4.3。

参考文献:

[R14-1] Z. Zhu, J. Wu, B. Yu, L. Wu, and J. Ma, *The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects*, in *International Conference on Machine Learning*. PMLR, 2019, pp. 7654–7663.

[R14-2] L. Wu, C. Ma *et al.*, "How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[R14-3] https://imgur.com/a/Hqolp (需要科学上网)

15.Sharpness-aware Minimization

作业描述：

在课程中，我们介绍了 Sharpness-aware Minimization（SAM）方法及其在提升泛化能力上的优越性。请参考[R15-1]或其它参考资料，完成下列探究任务：

- A toy model：在二维情形下构造一组损失函数，并针对这一损失函数，选取合适的学习率和迭代初始点，分别进行标准 SGD 与带有 SAM 的 SGD 训练。绘制损失函数的等高线图，并进一步绘制不同情形的迭代轨迹图，并以此说明 SAM 在迭代过程中会更青睐于周边景观更为"平坦"的最小值。最好能够以动图的方式来 visualize 整个收敛过程，visualization 可以参考[R14-3]
- 请你继续研究，SAM 中这种对平坦解的倾向性与扰动项ϵ大小的关系
- 对 CIFAR10 选取合适的神经网络，分别利用标准 SGD 与带有 SAM 的 SGD 进行求解，通过比较 test accuracy 体现 SGD 相比 GD 在泛化性能上的优越性。

参考文献:

[R15-1] *P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware minimization for efficiently improving generalization," in International Conference on Learning Representations, 2020.*