高维情形下线性模型的泛化误差研究

2024 春季学期四月工作

Yukun Dong

目录

任务: **1**

对于真实数据集，利用随机梯度下降训练核学习机处理分类任务 . . 2

方法一 2

方法二 4

复现 fig4，在 MNIST 数据集上，不同样本个数对应的 RKHS norm 的变化（失败） 6

参考文献 **7**

任务:

- 阅读：Diving into the shallows: a computational perspective on large-scale shallow learning。该文章从计算的角度研究了核学习机的优化手段。
- 复现：Belkin, Ma, and Mandal (2018) ，To Understand Deep Learning We Need to Understand Kernel Learning。该文章给出了一些实验的详细细节，涉及到 kernel 的编程，值得复现。具体来说，要结合真实数据集（首先使用 FashionMNIST 数据集）

对于真实数据集，利用随机梯度下降训练核学习机处理分类任务

方法一

这个方法来自文献 Ma and Belkin (2019)，其目的是最大程度地利用 GPU 的存储资源和并行计算资源。当 batch size 在一定范围内增加时，训练耗时会线性减小，只到一个阈值。该方法的目标是找到这个阈值 m^*。具体来讲，m^* 的大小和我们的 Gram matrix 的最大特征值是直接相关的。近似地有如下解析表达式：

We can calculate $m^*(k)$ explicitly using kernel matrix K (depending on the data),

$$m^*(k) = \frac{\beta(K)}{\lambda_1(K)} \text{ where } \beta(K) \triangleq \max_{i=1,\ldots,n} k(\boldsymbol{x}_i, \boldsymbol{x}_i)$$

For any shift invariant kernel k, after normalization, we have $\beta(K) = \max_{i=1}^{n} k(\boldsymbol{x}_i, \boldsymbol{x}_i) \equiv 1$.

因此，可以针对已有计算资源 G 和计算出来的 m^* 的大小构造一个新的 kernel K_G，使得 K_G 和原有的 Gram matrix K 具有相同的插值解。

此外，为了估计 K 的最大的 q 个特征值，我们采用子抽样的方法进行高效的估计：

Input: Kernel function $k(\boldsymbol{x}, \boldsymbol{z})$, EigenPro parameter q, mini-batch size m, step size η, size of fixed coordinate block s

initialize model parameter $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^T \leftarrow 0$
subsample s coordinate indices $r_1, \ldots, r_s \in \{1, \ldots, n\}$ for constructing $\mathscr{P}_q$, which form fixed coordinate block $\boldsymbol{\alpha}_r \triangleq (\alpha_{r_1}, \ldots, \alpha_{r_s})^T$
compute top-q eigenvalues $\Sigma \triangleq \text{diag}(\sigma_1, \ldots, \sigma_q)$ and corresponding eigenvectors $V \triangleq (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_q)$ of subsample kernel matrix $K_s = [k(\boldsymbol{x}_{r_i}, \boldsymbol{x}_{r_j})]_{i,j=1}^{s}$

在此子取样估计的结果之上再进行参数的更新：

for $t = 1, \ldots$ **do**

1. sample a mini-batch $(\boldsymbol{x}_{t_1}, y_{t_1}), \ldots, (\boldsymbol{x}_{t_m}, y_{t_m})$
2. calculate predictions on the mini-batch

$$f(\boldsymbol{x}_{t_j}) = \sum_{i=1}^{n} \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}_{t_j}) \quad \text{for} \quad j = 1, \ldots, m$$

3. update **sampled coordinate block** corresponding to the mini-batch $\boldsymbol{\alpha}_t \triangleq (\alpha_{t_1}, \ldots, \alpha_{t_m})$,

$$\boldsymbol{\alpha}_t \leftarrow \boldsymbol{\alpha}_t - \eta \cdot \frac{2}{m}(f(\boldsymbol{x}_{t_1}) - y_{t_1}, \ldots, f(\boldsymbol{x}_{t_m}) - y_{t_m})^T$$

4. evaluate the following feature map $\phi(\cdot)$ on the mini-batch features $\boldsymbol{x}_{t_1}, \ldots, \boldsymbol{x}_{t_m}$:

$$\phi(\boldsymbol{x}) \triangleq (k(\boldsymbol{x}_{r_1}, \boldsymbol{x}), \ldots, k(\boldsymbol{x}_{r_s}, \boldsymbol{x}))^T$$

5. update **fixed coordinate block** $\boldsymbol{\alpha}_r$ to apply $\mathscr{P}_q$,

$$\boldsymbol{\alpha}_r \leftarrow \boldsymbol{\alpha}_r + \eta \cdot \frac{2}{m} \sum_{i=1}^{m} (f(\boldsymbol{x}_{t_i}) - y_{t_i}) \cdot V D V^T \phi(\boldsymbol{x}_{t_i})$$

where $D \triangleq (1 - \sigma_q \cdot \Sigma^{-1})\Sigma^{-1}$

end for

将此算法应用在 CIFAR-10 和 FashionMNIST 数据集上，采用不同的 kernel 得到的结果如下：

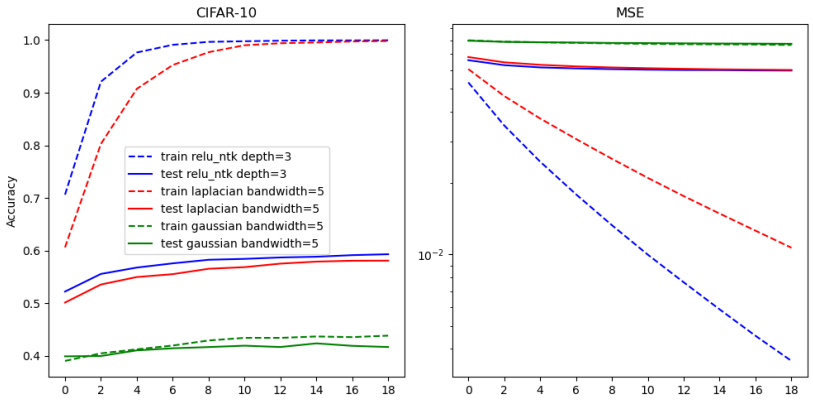


图 1: CIFAR-10

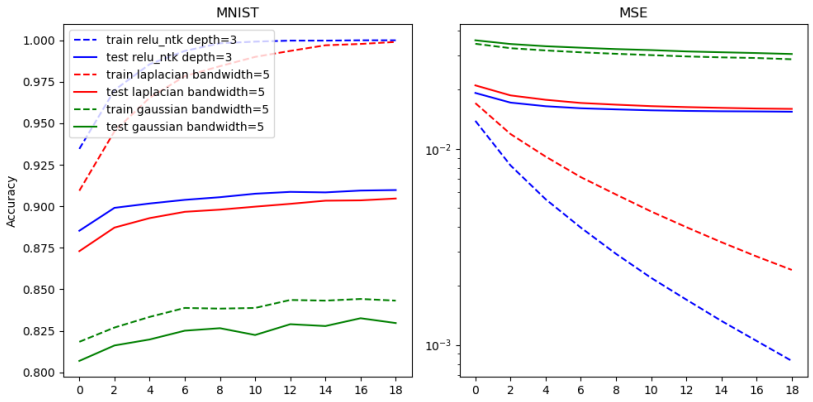


图 2: FashionMNIST

可以看出，relu_ntk 和 Laplacian 核表现很好，然而 Gaussian 核往往意味着更差的效果和更久的训练时间。

方法二

方法二主要参考 Abedsoltan, Belkin, and Pandit (2023) 的工作，是 Ma and Belkin (2019) 的续集。Ma and Belkin (2019) 的工作的局限性在于核学习机只能是

$$f(x) = \sum_{i=1}^{n} \alpha_i K(x, x_i)$$

的形式，即 kernel 的个数等于样本个数：相当于针对 interpolation 情形设计的算法，而没有给出 kernel 个数为 $p \neq n$ 时的迭代算法。Abedsoltan, Belkin, and Pandit (2023) 的工作则是设计算法解下列问题：

$$\text{argmin}_{f \in \mathcal{H}} \sum_{i=1}^{n} (f(x_i) - y_i)^2 \quad \text{subject to} \quad f(x) = \sum_{i=1}^{p} \alpha_i K(x, z_i) \quad \forall x$$

在方法一的基础之上，这里使用了两处预处理器 (preconditioner)，采用了基于投影的预处理器：

$$f^{t+1} = \text{proj}_{\mathcal{C}} \left(f^t - \eta \mathcal{P}(\nabla L(f^t)) \right),$$

其中 ∇L 是 Fréchet 导数，$\mathcal{P}$ 是基于 K 的前 q 个特征值和特征向量共同构建的。迭代步骤如下，其中 “EigenPro2” 对应方法一中的算法。

Require: Data $(X, \boldsymbol{y})$, centers Z, batch size m, Nyström size s,, preconditioner level q.

Fetch subsample $X_s \subseteq X$ of size s
$(\Lambda, \boldsymbol{E}, \lambda_{q+1}) \leftarrow$ top-q eigensystem of $K(X_s, X_s)$
$\boldsymbol{C} = K(Z, X_s)\boldsymbol{E}(\Lambda^{-1} - \lambda_{q+1}\Lambda^{-2})\boldsymbol{E}^\top \in \mathbb{R}^{p\times s}$
while Stopping criterion is not reached **do**
    Fetch minibatch $(X_m, \boldsymbol{y}_m)$
    $\boldsymbol{g}_m \leftarrow K(X_m, Z)\boldsymbol{\alpha} - \boldsymbol{y}_m$
    $\boldsymbol{h} \leftarrow K(Z, X_m)\boldsymbol{g}_m - \boldsymbol{C}K(X_s, X_m)\boldsymbol{g}_m$
    $\boldsymbol{\theta} \leftarrow$ EigenPro 2.0$(Z, \boldsymbol{h})$
    $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} - \frac{n}{m}\eta\,\boldsymbol{\theta}$

在 CIFAR-10 和 FashionMNIST 数据集上训练结果如下：

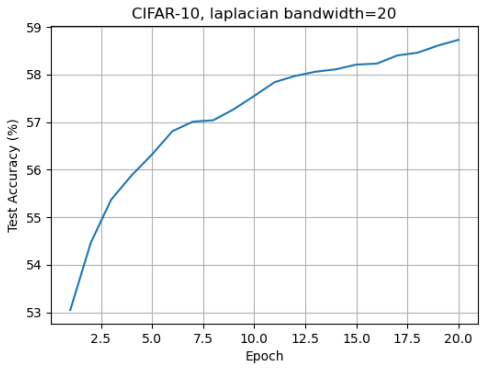


图 3: CIFAR-10

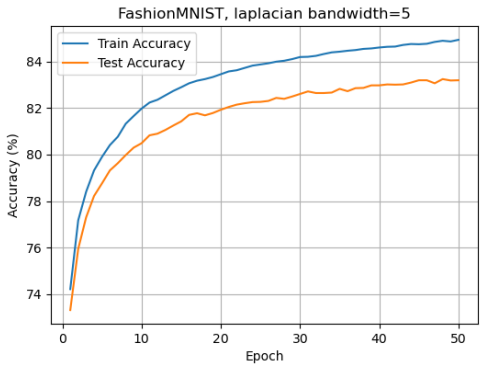
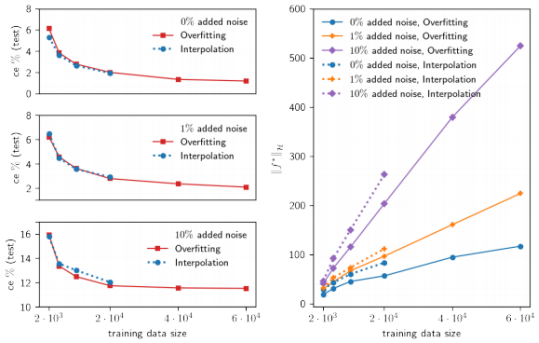


图 4: FashionMNIST

复现 fig4，在 MNIST 数据集上，不同样本个数对应的 RKHS norm 的变化（失败）

论文 Belkin, Ma, and Mandal (2018) 给 label 加上比率为 ε 的噪声，可以观测到噪声的程度越大 RKHS 的 norm 越大，然而在我的实验中并没有观测到这一点。此外，文章对 interpolation 和 overfit 情况进行了区分并且强调 interpolation 的 norm 会有指数级别增长，仍然无法给出多项式级别的 bounds。在计算 overfit 情形的 RKHS norm 的时候失败，代码有待进一步修复。下面给出待复现的原图：



(a) MNIST

遇到问题的主要原因：在 y 是一元的情况下，RKHS norm 易于用公式计算；但是在多分类问题中采用的是独热编码，这就导致了问题；此外，似乎计算

复现 *fig4*,在 *MNIST* 数据集上,不同样本个数对应的 *RKHS norm* 的变化(失败)7

RKHS norm 需要算很久。

参考文献

Abedsoltan, Amirhesam, Mikhail Belkin, and Parthe Pandit. 2023. "Toward Large Kernel Models." In *Proceedings of the 40th International Conference on Machine Learning*, edited by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, 202:61–78. Proceedings of Machine Learning Research. PMLR. https://proceedings.mlr.press/v202/abedsoltan23a.html.

Belkin, Mikhail, Siyuan Ma, and Soumik Mandal. 2018. "To Understand Deep Learning We Need to Understand Kernel Learning." In *Proceedings of the 35th International Conference on Machine Learning*, edited by Jennifer Dy and Andreas Krause, 80:541–49. Proceedings of Machine Learning Research. PMLR. https://proceedings.mlr.press/v80/belkin18a.html.

Ma, Siyuan, and Mikhail Belkin. 2019. "Kernel Machines That Adapt to GPUs for Effective Large Batch Training." *Proceedings of Machine Learning and Systems* 1: 360–73.