arXiv:2602.03394v1 [stat.ML] 3 Feb 2026

通过二次近似改进线性化拉普拉斯近似

Pedro Jim´enez[1], Luis A. Ortega[1,2],Pablo
Morales-´Alvarez[3,4], Daniel Hern´andez-Lobato[1,5]*

1- 机器学习组 - 计算机科学系 马德里自治大学高等理工学院，西班牙 2- 奥尔堡大学计算机科学系，哥本哈根 3- 视觉信息处理组 统计与运筹学系，格拉纳达大学，西班牙 4- 信息与通信技术研究中心（CITIC），格拉纳达大学，西班牙 5- 马德里自治大学高级基础物理研究中心，西班牙

摘要。深度神经网络（DNN）常常会对分布外预测表现出过度自信，因此有必要进行贝叶斯不确定性量化。线性化拉普拉斯近似（LLA）通过对深度神经网络进行线性化，并对所得模型应用拉普拉斯推断来实现这一目标。值得注意的是，预测时也使用了线性模型。我们认为，对后验分布进行这种线性化可能会降低其对真实拉普拉斯近似的拟合程度。为缓解这一问题，同时避免显著增加计算成本，我们提出了二次拉普拉斯近似（QLA）。QLA通过高效的幂迭代得到秩一因子，并用它近似近似拉普拉斯对数后验中的每个二阶因子。QLA有望在不构造通常难以计算的完整海森矩阵的情况下，使后验精度更接近完整拉普拉斯近似的后验精度。预测时，QLA同样使用线性化模型。在五个回归数据集上的实证结果表明，与LLA相比，QLA带来的不确定性估计改进幅度虽小，却始终如一。

1 引言

深度神经网络（DNNs）是许多监督学习任务中的最先进模型，在不同领域均取得了卓越的性能[1]。尽管这类模型具有强大的预测能力，但对于远离训练分布的输入，它们通常会产生过度自信的输出，从而削弱了其在安全关键应用中的可靠性 [2]。一种有原则的解决方案是采用

*DHL 和 LAO 感谢由 MCIN 资助的项目 PID2022-139856NB-I00，以及由马德里自治区资助的 IDEA-CM 项目（TEC-2024/COM-89）和 ELLIS 马德里分部提供的经费支持。我们感谢马德里自治大学科学计算中心（CCC-UAM）提供的支持。PMA 感谢由 MCIN/AEI/10.13039/501100011033（西班牙科学部）资助的项目 PID2022-140189OB-C22，以及由大学、研究与创新部和欧盟（EU）安达卢西亚 ERDF 计划 2021–2027 资助的 C-EXP-153-UGR23 项目。

对模型参数进行贝叶斯处理，由此得到参数的后验分布，并通过边缘化得到能够在给出点预测的同时量化不确定性的预测分布 [3]。

由于涉及高维积分且参数数量庞大，对现代深度神经网络进行精确贝叶斯推断是难以处理的。因此，近似推断方法被广泛使用。拉普拉斯近似（LA） [4] 经典且概念简单，但在深度模型中计算要求很高，因为它需要处理非常大的海森矩阵。线性化拉普拉斯近似（LLA） [5] 在参数附近对神经网络进行线性化，并在所得线性模型中应用LA。LLA大幅降低了计算成本，并在实践中给出有用的预测分布。重要的是，使用线性模型进行预测还能缓解LA的部分偏差 [5]。

然而，后验计算中的线性化可能会降低表达能力，并损害不确定性校准。为了更好地平衡效率与保真度，我们提出二次拉普拉斯近似 （QLA）。QLA围绕最大后验估计（MAP）对神经网络构建二阶展开。不过，它用通过高效海森向量积进行幂迭代得到的秩一因子，来近似占主导的逐数据点精度项。这样无需形成完整海森矩阵，便能得到更精细的雅可比矩阵 。对于预测，我们采用线性化预测模型；如 [5],所示，这有助于减轻一种倾向的影响，即向数据不支持的参数空间区域分配过多概率质量。从经验上看，在五个回归数据集上，QLA相对于LLA在不确定性估计方面取得了虽小但一致的改进。

2 线性化拉普拉斯近似

我们考虑一个回归问题，其数据为 $\mathcal{D} = \{(\mathbf{x}_1, y_1) \ldots, (\mathbf{x}_N, y_N)\}$，其中每个输入为 $\mathbf{x}_n \in \mathbb{R}^D$ ，目标为 $y_n \in \mathbb{R}$。神经网络是一个参数化函数 $f\colon \mathbb{R}^D \times \mathbb{R}^P \to \mathbb{R}$ ，具有 P 个可学习参数，将输入 $\mathbf{x}_n$ 和参数向量 $\theta \in \mathbb{R}^P$ 映射为预测输出 $f(\mathbf{x}_n, \theta)$。

目标变量被建模为网络输出加上随机噪声，从而导出条件观测模型 $p(y \mid f(\mathbf{x}\ \theta))$。假设各样本之间条件独立，似然可分解为

$$p(\mathcal{D} \mid \theta) = \prod_{n=1}^{N} p(y_n \mid f(\mathbf{x}_n, \theta)) \ . \tag{1}$$

给定先验分布 $p(\theta)$，贝叶斯处理旨在推断以观测数据为条件的参数后验分布。然而，由于 θ 的高维性以及 f 的非线性，现代深度神经网络中的精确推断难以处理。因此，我们采用拉普拉斯近似，它用一个以最大后验估计 θ^* 为中心的高斯分布 $q(\theta)$ 来近似后验分布：

$$q(\theta) = \mathcal{N}(\theta \mid \theta^*, \Sigma), \qquad \theta^* = \arg\max_\theta \ell(\theta, \mathcal{D}) \ , \tag{2}$$

其中 $\ell(\theta,\ \mathcal{D}) = \sum_{n=1}^{N} \log\ p(y_n \mid f(\mathbf{x}_n,\ \theta)) + \log\ p(\theta)$ 是对数联合密度。精度矩阵由 θ^* 处的负海森矩阵给出：$\Sigma^{-1} = -\nabla_\theta^2 \ell(\theta,\ \mathcal{D})\Big|_{\theta=\theta^*}$。

为计算这个海森矩阵，我们使用其逐数据分解，

$$\nabla_\theta^2 \ell\left(\theta, \mathcal{D}\right) = \sum_{n=1}^N \nabla_\theta^2 \log p\left(y_n \mid f\left(\mathbf{x}_n, \theta\right)\right) + \nabla_\theta^2 \log p(\theta)\,. \tag{3}$$

先验通常易于处理，因此我们重点关注数据相关求和项：

$$\nabla_\theta \log p\left(y \mid f\left(\mathbf{x}, \theta\right)\right) = \mathbf{r}\left(y; f\right) \mathcal{J}_\theta(\mathbf{x})\,, \tag{4}$$

$$\nabla_\theta^2 \log p\left(y \mid f\left(\mathbf{x}, \theta\right)\right) = \mathbf{r}\left(y; f\right) \mathcal{H}_\theta(\mathbf{x}) - \mathcal{J}_\theta(\mathbf{x}) \Lambda(y; f) \mathcal{J}_\theta^\intercal(\mathbf{x})\,, \tag{5}$$

其中，$\mathcal{J}_\theta(\mathbf{x})$ 和 $\mathcal{H}_\theta(\mathbf{x})$ 分别是神经网络的参数雅可比矩阵和参数海森矩阵，$\mathbf{r}\left(y;\ f\right)$ 表示残差，而 $\Lambda(y;\ f)$ 表示逐输入噪声。广义高斯—牛顿（GGN）近似 [6] 舍弃了项 $\mathcal{H}_\theta(\mathbf{x})^\intercal \mathbf{r}\left(y;\ f\right)$，从而得到

$$\nabla_\theta^2 \log p\left(y \mid f\left(\mathbf{x}, \theta\right)\right) \approx -\mathcal{J}_\theta(\mathbf{x}) \Lambda(y; f) \mathcal{J}_\theta^\intercal(\mathbf{x})\,. \tag{6}$$

这一近似可视为在 θ^* [5] 处对深度神经网络进行线性化，

$$f\left(\mathbf{x}, \theta\right) \approx f_{\text{lin}}^{\theta^*}(\mathbf{x}, \theta) = f(\mathbf{x}, \theta^*) + \mathcal{J}_{\theta^*}^\intercal(\mathbf{x})(\theta - \theta^*)\,. \tag{7}$$

对线性化模型应用拉普拉斯近似后，后验近似变为

$$q(\theta) = \mathcal{N}\left(\theta \mid \theta^*, \Sigma_{\mathbf{GGN}}\right)\,, \quad \Sigma_{\mathbf{GGN}}^{-1} = \sum_{n=1}^N \mathcal{J}_{\theta^*}(\mathbf{x}_n) \Lambda(y_n; f_n) \mathcal{J}_{\theta^*}^\intercal(\mathbf{x}_n) + \mathbf{S}_0^{-1}\,,$$

其中，$\mathbf{S}_0$ 是先验协方差，通常为对角矩阵。在 $q(\theta)$ 下进行边缘化，可得到标准拉普拉斯预测，

$$p\left(y \mid \mathbf{x}, \mathcal{D}\right) = \mathbb{E}_{q(\theta)}\left[p\left(y \mid f\left(\mathbf{x}, \theta\right)\right)\right]\,. \tag{8}$$

按照 [5], ，也可以改用线性化模型进行预测，以获得更好的结果，从而得到广义线性模型（GLM）预测

$$p_{\text{GLM}}\left(y \mid \mathbf{x}, \mathcal{D}\right) = \mathbb{E}_{q(\theta)}\left[p\left(y \mid f_{\text{lin}}^{\theta^*}\left(\mathbf{x}, \theta\right)\right)\right]\,. \tag{9}$$

对于高斯观测，这会简化为一个高斯预测分布，其均值为 $f(\mathbf{x}\ \theta^*)$，方差为 $\mathcal{J}_{\theta^*}^\intercal(\mathbf{x}) \Sigma_{\mathbf{GGN}} \mathcal{J}_{\theta^*}(\mathbf{x})$。

3 二次拉普拉斯近似

LLA 提供了一个高效的框架，大幅简化了原始拉普拉斯近似的计算。通过在局部用线性模型近似深度神经网络，LLA 无需显式计算海森矩阵；在使用线性化模型进行预测推断时，还能得到更稳定的不确定性估计 [5]。

然而，后验分布中的线性假设可能会使后验方差估计产生偏差，从而限制预测不确定性估计的准确性。为解决这一局限，我们提出了二次拉普拉斯近似 (QLA)，它通过在后验计算中采用深度神经网络的二次局部近似来扩展 LLA：

$$f\left(\mathbf{x}, \theta\right) \approx f_{\text{quad}}^{\theta^*}\left(\mathbf{x}, \theta\right) = f_{\text{lin}}^{\theta^*}\left(\mathbf{x}, \theta\right) + 0.5\left(\theta - \theta^*\right)^\intercal \mathcal{H}_{\theta^*}\left(\mathbf{x}\right)\left(\theta - \theta^*\right)\,, \tag{10}$$

其中，$\mathcal{J}_{\theta^*}(\mathbf{x})$ 和 $\mathcal{H}_{\theta^*}(\mathbf{x})$ 分别表示在最大后验估计 θ^*处计算的神经网络雅可比矩阵和海森矩阵。这对应于围绕 θ^* 的二次泰勒展开(QTE)。这正是拉普拉斯近似所采用的方法。

与经典拉普拉斯近似一样，我们将后验分布近似为一个以最大后验解为中心的高斯分布：

$$q(\theta) = \mathcal{N}\left(\theta \mid \theta^*, \Sigma_{\mathbf{QTE}}\right), \tag{11}$$

其精度为

$$\Sigma_{\mathbf{QTE}}^{-1} = \sum_{n=1}^{N} \nabla_\theta^2 \log p\left(y_n \mid f_{\text{quad}}^{\theta^*}(\mathbf{x}_n, \theta)\right) + \mathbf{S}_0^{-1}. \tag{12}$$

聚焦于依赖数据的项，我们得到

$$\nabla_\theta^2 \log p\Big(y \mid f_{\text{quad}}^{\theta^*}(\mathbf{x}, \theta)\Big)\Big|_{\theta=\theta^*} = \mathbf{r}(y; f_{\text{quad}}^{\theta^*})\mathcal{H}_{\theta^*}(\mathbf{x}) - \mathcal{J}_{\theta^*}(\mathbf{x})\Lambda(y; f_{\text{quad}}^{\theta^*})\mathcal{J}_{\theta^*}^\intercal(\mathbf{x}).$$

此时，我们面临着显式求得神经网络海森矩阵的挑战。然而，在$PyTorch$等框架中，无需构造完整海森矩阵即可高效计算海森向量积。利用这一特性，我们采用幂迭代方法来估计

$$\mathbf{A} := \nabla_\theta^2 \log p\Big(y \mid f_{\text{quad}}^{\theta^*}(\mathbf{x}, \theta)\Big)\Big|_{\theta=\theta^*}. \tag{13}$$

我们以参考点处的神经网络雅可比矩阵$\mathbf{z}^{(0)} = \mathcal{J}_{\theta^*}(\mathbf{x})$初始化该过程，并递归更新

$$\mathbf{z}^{(k+1)} = \frac{\mathbf{A}\mathbf{z}^{(k)}}{||\mathbf{A}\mathbf{z}^{(k)}||}. \tag{14}$$

根据分配律，每次与$\mathbf{A}$相乘时，$\mathbf{A}\mathbf{z}^{(k)}$ 可分解为

$$\mathbf{A}\mathbf{z}^{(k)} = \mathbf{r}(y; f_{\text{quad}}^{\theta^*})\mathcal{H}_{\theta^*}(\mathbf{x})\mathbf{z}^{(k)} - \mathcal{J}_{\theta^*}(\mathbf{x})\Lambda(y; f_{\text{quad}}^{\theta^*})\mathcal{J}_{\theta^*}^\intercal(\mathbf{x})\mathbf{z}^{(k)}. \tag{15}$$

因此，无需构造完整的海森矩阵，只需计算海森-向量积 $\mathcal{H}_{\theta^*}(\mathbf{x})\mathbf{z}^{(k)}$ ，而这可以通过自动微分高效获得。

经过若干次迭代后，该算法收敛到主特征向量，记为 $\hat{\mathbf{z}}$，它可以解释为一个精细化的雅可比矩阵，同时捕获一阶信息以及最显著的二次贡献。在这一视角下，我们用秩一分解 $\mathbf{A} \approx \hat{\mathbf{z}}\hat{\mathbf{z}}^\intercal$ 来近似 $\mathbf{A}$ ，从而得到高斯精度矩阵的如下表达式：

$$\Sigma_{\mathbf{QTE}}^{-1} \approx \sum_{n=1}^{N} \hat{\mathbf{z}}_n \hat{\mathbf{z}}_n^\intercal + \mathbf{S}_0^{-1}. \tag{16}$$

在 $q(\theta)$ 下进行边缘化，并使用神经网络二次近似，可得到近似预测分布

$$p_{\text{QTE}}(y \mid \mathbf{x}, \mathcal{D}) = \mathbb{E}_{q(\theta)}\left[p\left(y \mid f_{\text{quad}}^{\theta^*}(\mathbf{x}, \theta)\right)\right]. \tag{17}$$

对于高斯观测，预测均值为

$$f(\mathbf{x}, \theta^*) + \frac{1}{2}\mathrm{tr}\left(\Sigma_{\mathbf{QTE}}\mathcal{H}_{\theta^*}(\mathbf{x})\right) , \tag{18}$$

且预测方差为

$$\mathcal{J}_{\theta^*}^{\intercal}(\mathbf{x})\Sigma_{\mathbf{QTE}}\mathcal{J}_{\theta^*}(\mathbf{x}) + \frac{1}{2}\mathrm{tr}\left[\left(\mathcal{H}_{\theta^*}(\mathbf{x}_i)\Sigma_{\mathbf{QTE}}\right)^2\right] . \tag{19}$$

在此阶段会出现两个缺点。首先，预测均值不再与在最大后验参数处计算得到的网络输出一致。其次，所得方差在形式上更接近标准拉普拉斯近似，未能充分缓解该近似倾向于将过多概率质量分配给缺乏依据的参数区域这一问题。

为了解决这一问题，我们改用线性化模型进行预测，与 LLA 中的做法相同；线性化模型是对我们的二次展开的最佳一阶近似。由此得到的预测分布为

$$p_{\mathrm{QTE}}\left(y \mid \mathbf{x}, \mathcal{D}\right) = \mathcal{N}\left(y \mid f\left(\mathbf{x}, \theta^*\right), \mathcal{J}_{\theta^*}^{\intercal}(\mathbf{x})\Sigma_{\mathbf{QTE}}\mathcal{J}_{\theta^*}(\mathbf{x})\right) . \tag{20}$$

4 实验

我们在标准 UCI 回归数据集上比较 LLA 和 QLA：波士顿房价、能源效率、游艇水动力学、混凝土抗压强度和葡萄酒质量。训练前，我们对输入和目标值进行了标准化。为评估分布外样本的不确定性估计，我们采用 [7], 中提出的中间划分，另见 [8]。对于每个输入维度，我们先对数据排序，再将中间三分之一留作测试集，外侧三分之二作为训练集，从而为每个数据集得到 D 个间隙划分。

对于每个数据集和每个间隔划分，我们使用反向传播训练一个深度神经网络（因此每个数据集有 D 个神经网络）。深度神经网络超参数（权重衰减 $\{0.0, 10^{-4}, 10^{-3}\}$、单元数量 $\{20, 30, 50\}$以及层数 $\{1, 2, 3\}$）通过在训练集上使用内部交叉验证方法确定。然后，我们同时使用LLA和QLA获得预测分布，并在相应的测试划分上计算所选指标。在QLA中，我们使用由LLA估计得到的先验方差。该估计通过最大化边际似然来完成，如 [5]所述。我们在QLA中使用幂迭代法进行10次迭代。最后，我们报告跨数据划分取平均后的指标。

预测分布的质量使用负对数似然和连续排序概率评分[9]来衡量。这些是涉及LLA与不确定性估计的文献中常用的指标 [2]。由于LLA和QLA具有相同的预测均值，这些指标上的差异反映了预测方差的差异，因此可直接评估每种方法产生真实不确定性估计的能力。

完整结果见表 1。该表展示了每个数据集上 LLA 和 QLA 在中间区间划分上的平均测试得分（NLL 和连续排序概率评分，数值越低越好）。最佳结果以

加粗。总体而言，QLA 的性能与 LLA 相当，并且在大多数数据集上都能在不确定性指标方面带来幅度较小但一致的改进，尽管差异通常很小。这些结果表明，QLA 能够优化后验精度，并适度改善不确定性估计，同时保持初始深度神经网络的预测性能。

表 1：LLA 和 QLA 在各数据集上的平均指标结果。

数据集	负对数似然 (↓)		连续排序概率评分 (↓)	
	LLA	QLA	LLA	QLA
波士顿	2.7290 ± 0.0402	$\mathbf{2.7279 \pm 0.0401}$	2.0581 ± 0.0941	$\mathbf{2.0579 \pm 0.0941}$
能源	2.8704 ± 0.4524	$\mathbf{2.8387 \pm 0.4455}$	2.5091 ± 0.8006	$\mathbf{2.5009 \pm 0.7975}$
游艇	2.5153 ± 0.0899	$\mathbf{2.5143 \pm 0.0887}$	$\mathbf{1.6166 \pm 0.2008}$	1.6169 ± 0.1999
混凝土	3.3332 ± 0.0227	$\mathbf{3.3331 \pm 0.0228}$	3.8750 ± 0.1003	$\mathbf{3.8742 \pm 0.1003}$
Wine	1.0022 ± 0.0089	$\mathbf{1.0015 \pm 0.0089}$	0.3663 ± 0.0040	$\mathbf{0.3661 \pm 0.0040}$

5 结论

本文介绍了二次拉普拉斯近似 （QLA），这是对线性化拉普拉斯近似的扩展，通过幂迭代法获得的高效秩一更新，在对数后验中纳入二阶项。这种方法无需显式计算海森矩阵，即可改进后验精度。在预测方面，它通过线性化预测模型保留了 LLA 稳定的预测行为，从而缓解了将过多概率质量分配给缺乏支持的参数区域的倾向 [5]。在五个回归数据集上的实验表明，与 LLA 相比，QLA 几乎不增加额外计算成本，却能在不确定性指标上带来小幅但稳定的提升。未来工作将致力于把 QLA 扩展到分类和多元回归，并通过低秩策略或基于诱导点的策略提升其在大型神经网络上的可扩展性 [2]。

参考文献

[1] Y. LeCun、Y. Bengio 和 G. Hinton。深度学习。 *Nature*，521:436–444，2015。[2] L. A. Ortega、S. Rodr´ıguez-Santana 和 D. Hern´andez-Lobato。用于贝叶斯深度学习的变分线性化拉普拉斯近似。载于 *ICML*，2024。[3] C. M. Bishop。 *Pattern Recognition and Machine Learning*。施普林格纽约，纽约州，2006。[4] D. J. C. MacKay。贝叶斯模型比较与反向传播网络。载于 *Proceedings of the 5th International Conference on Neural Information Processing Systems* 1991 ，。

[5] A. Immer、M. Korzepa 和 M. Bauer。通过局部线性化改进贝叶斯神经网络的预测。载于 *AISTATS*，2021。[6] J. Martens 和 R. Grosse。使用 Kronecker 因子近似曲率优化神经网络。载于 *ICML*，2015。[7] A. Y. K. Foong、Y. Li、J. M. Hern´andez-Lobato 和 R.E. Turner。贝叶斯神经网络中的“中间”不确定性。 *Uncertainty and Robustness in Deep Learning*，2019。[8] P. Morales-Alvarez、D. Hern´andez-Lobato、R. Molina 和 J. M. Hern´andez-Lobato。深度神经网络中的激活层级不确定性。载于 *ICLR*，2021。[9] T. Gneiting 和 A. E. Raftery。严格适当评分规则、预测与估计。*Journal of the American Statistical Association*，102:359–378，2007。