arXiv:2303.01540v1 [stat.ML] 2 Mar 2023

采用概率反向传播的贝叶斯神经网络变分 EP

Kehinde Olobatuyi*

统计学与数理金融，米兰-比可卡大学，
意大利

摘要

我提出了一种用于非线性逻辑回归的新方法，该方法采用具有两层神经网络（NN）模型结构，并在网络权重上设置层次先验。我提出了一种称为变分期望传播（VEP）方法的期望传播混合方法，用于对权重、先验的层次尺度参数以及 zeta 的后验分布进行近似积分。通过采用因子化后验近似，我推导出了一种计算效率高的算法，其复杂度与独立稀疏逻辑回归模型集成的复杂度相近。该方法可以扩展到标准激活函数和神经网络模型结构之外，从多个稀疏线性模型构建灵活的非线性二元预测器。我考虑了一个具有逻辑回归似然的层次贝叶斯模型，并对称为权重的参数和超参数设定高斯先验分布。我从 E 步和 M 步的角度开展工作，分别利用计算得到的后验分布计算近似后验并更新参数。

1 引言

贝叶斯方法有诸多优势，已被用于解决许多复杂模型。尤其是，它们以统一的方式处理各种代价下的不确定性，因此独具特色。贝叶斯方法的一项突破性进展是将其应用于神经网络。神经网络（NN）在监督学习问题领域重新兴起。在典型用法中，神经网络是表达能力很强的模型，能够从输入/输出示例中学习复杂函数的近似表示（[Hornik 等人， 1989]）。神经网络取得的一些成就，归功于利用随机优化在海量数据集上训练它们的能力（[Bottou， 1986]），以及通过反向传播更新称为权重的参数（[Rumelhart 等人， 1986]）。更快的计算设备、更大的数据集，以及正则化技术（随机失活）（[Srivastava 等人， 2014]）和用于非线性变换的整流线性单元（[Nair 和 Hinton， 2010]）等创新，共同推动了神经网络在语音识别（[Hinton 等人， 2012]； [Hannun 等人， 2014]）、自然语言处理（[Collobert 和 Weston， 2008]； [Sutskever 等人， 2014]）以及计算机视觉（[Krizhevsky 等人， 2012]）等任务上的成功。然而，神经网络学习仍面临一些挑战，原因如下：首先，处理大型数据集需要针对许多超参数进行专门调优，例如学习率、动量和权重衰减，而这些参数都依赖于层。

*电子邮箱： k.olobatuyi@campus.unimib.it；通讯作者

其次，神经网络只输出网络中权重的点估计。基于点估计的网络无法反映参数的不确定性。因此，在许多情况下，这些权重的表现可能不佳。理想情况下，应在给出预测的同时提供不确定性估计。此外，还常常存在过拟合和欠拟合（模型过于复杂或过于简单）的问题。文献中已采用神经网络的贝叶斯方法，以避免随机优化带来的这些缺陷（[MacKay， 1992]）。在实践中，贝叶斯技术可以通过对参数进行边缘化，从后验分布中自动推断超参数的取值。此外，贝叶斯技术还会将估计的不确定性纳入预测。贝叶斯方法通常通过对参数取值进行平均，而非采用单一的点估计，从而能够有效应对过拟合。已有多种用于神经网络贝叶斯学习的方法，例如拉普拉斯近似（[MacKay， 1992]）、哈密顿蒙特卡罗（[Neal， 1995]）、期望传播（[Jylanki 等人， 2014]）和变分推断（[Hinton 和 Camp， 1993]）。不过，期望传播尤其未能得到广泛应用，因为当似然与先验分布不属于同一个二次指数族时，无法通过对参数积分来简单地计算归一化因子，计算过程需要严格的解析推导。一个值得注意的例外是 [Graves， 2011]提出的可扩展变分推断方法。然而，该方法在实践中表现不佳，因为随机梯度计算中的蒙特卡罗近似会带来噪声。此外，基于期望传播的可扩展方法主要针对二值权重网络；将其扩展到连续权重时效果并不理想，因为它无法捕捉后验方差中的不确定性。对于前馈神经网络的输入层权重而言，高斯先验看起来可能并不合适，因为先验上允许权重取较大值 w_j ，而该权重可能对应无关变量 x_j ，从而降低模型的预测能力。这种行为与线性模型类似：多层感知机（MLP）网络中，与每个隐藏单元相关联的输入层权重可视为一个独立的线性模型，其输出会在下一层进行非线性变换。对未知输入权重的后验不确定性进行积分，可以减轻无关变量可能造成的有害影响。然而，如果输入变量或特征的总数相对于观测值数量按比例大幅增长，这种做法可能仍不够充分。抑制无关特征影响的另一种策略是，将线性模型与促进稀疏性的先验相结合，并使用一般激活函数以及隐藏单元之间的交互项（[Jylanki 等人， 2014]）。一种类似的方法是采用分层自动相关性确定（ARD）先验（[MacKay， 1992]; [Neal， 1996]）：为每个权重分别指定高斯先验 $w_j = \mathcal{N}(0, \tau_{l_j})$，并通过独立的方差超参数$\tau_{l_j}$ 对与每个特征相关联的权重组进行正则化。[MacKay，995a] 介绍了一种用于神经网络的 ARD 方法，其中使用拉普拉斯方法近似对相关性参数 τ_j 及其他模型超参数（如噪声参数）的点估计进行积分，以计算边际似然估计。此外， [Neal， 1996] 从随机近似的角度处理同一问题，例如采用马尔可夫链蒙特卡罗（MCMC），在这种方法中，近似积分针对所有模型参数（包括 w_j 和τ_{l_j}）的后验不确定性进行。已有多种计算效率较高的算法被提出，用于确定边际似然的相关性参数点估计（[Tipping， 2001]； [Qi 等人， 2004]； [Wipf 和 Nagarajan， 2008]）。然而，这些方法与线性模型有关，基于点估计的方法可能会发生过拟合，因为相关特征的最大后验（MAP）估计τ_{l_j} 也可能接近于零，正如 [Qi 等人， 2004]所示。将每个输入的非线性程度分别作为超参数进行优化时，这种方法也可用于基于高斯过程（GP）先验的无限神经网络（Williams，1998； [Rasmussen 和 Williams， 2006]）。基于促进稀疏性的先验（如拉普拉斯先验（[Seeger， 2008]）和尖峰与平板先验（[Hern´andez-Lobato 等人， 2008], [Hernandez-Lobato， 2010]）） 构造的 ARD 先验替代方案已被广泛

已在线性模型中提出。期望传播已被用于这些方法，以高效地对解析上难以处理的后验分布进行积分。这些方法相较于 ARD 先验的优势在于，它们不会遭受过拟合。这是由于对后验不确定性进行了近似积分，而该不确定性源于权重上的稀疏先验。由于其在许多实际应用中的计算效率和准确性，期望传播为近似这种积分提供了一种有用的马尔可夫链蒙特卡罗替代方案（Nickish 和 Rasmussen，2008年；[Hernandez-Lobato， 2010]）。在非线性回归中，已针对神经网络考虑了有利于稀疏性的拉普拉斯先验（Williams，1995），其中通过拉普拉斯近似进行推断。然而，拉普拉斯近似存在不连续性问题，即在后验众数处，对数后验密度的曲率对于所有类型的先验分布可能并未得到良好定义，其中包括在原点处导数不连续的拉普拉斯分布（Williams，1995；[Seeger， 2008]）。要成功实施该算法，需要进行一些额外的近似（Williams，1995）。相比之下，由于相对于高斯测度传播先验项的矩，期望传播可直接实施。拉普拉斯近似的另一个特性是，它用最大后验（MAP）估计近似未知量的后验均值，并用后验分布在众数处的负海森矩阵近似其后验协方差。这种设置可能会低估未知变量的整体不确定性，从而导致偏斜后验分布（Nickish 和 Rasmussen，2008年）或多峰后验分布（Jylanki 等人，2011）的预测性能不佳。此外，在典型的实际神经网络应用中，当未知量数量相对于观测值数量较多时，MAP 可能与后验均值存在显著差异。

1.1 主要贡献

本研究旨在从交织的视角和不同的散度角度，探讨期望传播与变分推断方法。我们对 EP 和 VB 算法进行了新的统一，使 EP 在层次贝叶斯框架中的解析处理不再那么严格。变分 EP（VEP）在 VB 更新阶段融入了 EP 的传播算法。VEP 着眼于期望传播算法的核心局限，同时推广了变分推断算法。EP 和 VB 算法最关键的部分是最小化 Kullback-Leibler 散度。这种同时适用于 EP 和 VB 方法的新方法称为变分期望传播（VEP）算法。VEP 具有多项优势，例如将 EP 的精化策略融入 VB，从而推广变分推断算法。此外，通过数据精化先验，而非像 VB 那样将先验对近似后验的贡献固定不变，预计能够提高 VB 的准确性，但这会牺牲全局更新规则。此外，VEP 在 EP 和 VB 算法之间建立了一条路径，或起到调和作用。VEP 通过在变分推断算法中加入扩展，避开了 EP 严格解析求解所面临的问题。此外，它还采用 VB 方法求解 EP 中难以处理的倾斜后验分布。这使得我们能够从混合 EP 与 VB 的角度出发，近似层次贝叶斯模型。我们提出了一些理论框架，以建立 EP 与 VB 之间的联系。此外，我们遵循 [Jylanki 等人提出的策略，将稀疏线性模型应用于非线性回归；该策略将偏好稀疏性的先验与双层回归模型相结合。其目标是采用 Jose 和 Ryan [Jose 和 Ryan， 2015]提出的概率反向传播方法，解决由神经网络观测模型产生的、解析上难以处理的似然函数所带来的挑战，从而构建可靠的高斯期望传播近似。

最后，我们针对以逻辑回归作为似然、以高斯先验分布作为先验的层次贝叶斯模型，推导其变分期望传播。在深度神经网络的背景下开展研究。

通过采用逻辑函数的近似下界（Jaakkola and Jordan，2000年）处理逻辑回归，本文扩展了 Jose and Ryan 的工作， [Jose and Ryan， 2015]。不过，本文使用边际似然计算超先验分布的参数。

2 EP算法的基本描述

本节简要回顾我们模型所依赖的期望传播（EP）和变分推断（VI）算法。设有一个数据集，其中包含来自参数化概率模型 $p(\vec{x}|\vec{\theta})$ 的 N 独立同分布样本 $\mathcal{D} = \{\vec{x}_i\}_{i-1}^{N}$，该模型由一个未知的 D 维向量 $\vec{\theta}$ 参数化，其分布为 $p_0(\vec{\theta})$。精确贝叶斯推断需要计算参数的后验分布，而这通常难以处理。给定数据后，后验分布可表示如下；

$$p(\vec{\theta}|\mathcal{D}) \propto \prod_{i=1}^{N} t(\vec{\theta}) p_0(\vec{\theta}) \approx q(\vec{\theta}) \propto p_0(\vec{\theta}) \prod_{i=1}^{N} \tilde{t}_i(\vec{\theta}) \tag{1}$$

其中， $p(\vec{x}_i|\vec{\theta}) = t_i(\vec{\theta})$ 是精确项， $\tilde{t}(\vec{\theta})$ 是近似项，而 $q(\vec{\theta})$ 是将由 EP 进一步优化的近似后验分布。EP 的目标是通过优化近似项，捕捉数据点经由似然项对后验分布的贡献，即 $t_i(\vec{\theta}) \propto p(\vec{x}_i|\vec{\theta})$。同理，一种方法是寻找使精确后验与以下分布之间的 Kullback–Leibler（KL）散度最小的近似后验：该分布由一个近似项替换其中一个似然项而形成， $t_i(\vec{\theta})$，即 $\mathrm{KL}[p(\vec{\theta}|\mathcal{D})||p(\vec{\theta}|\mathcal{D})\tilde{t}_i(\vec{\theta})/p(\vec{x}_i|\vec{\theta})]$。然而，由于涉及精确后验分布，这种更新仍然难以计算。EP 则通过从后验分布中移除一个观测值来近似这一过程，即 $p_{-i}(\vec{\theta}) \propto p(\vec{\theta}|\vec{x}_i)/t_i(\vec{\theta})$，称为精确留一法后验。然后用近似留一法后验 $q_{-i}(\vec{\theta}) \propto q(\vec{\theta})/\tilde{t}(\vec{\theta})$ 替代，称为腔分布。由于这会使各近似因子的更新相互耦合，因此必须迭代更新。从更一般的角度来看，EP 首先选择一个待更新因子，并将其从近似后验分布中移除，以生成腔分布。随后纳入 $\vec{\ }\ \vec{\ }\ \vec{\ }$ 对应的似然项，得到倾斜后验 $\tilde{p}_i(\theta) \propto q_{-i}(\theta)t_i(\theta)$。EP 通过最小化 $\mathrm{KL}[\tilde{p}(\vec{\theta})||q(\vec{\theta})]$ 来更新近似后验分布。这确保了精确项对精确后验和倾斜后验的贡献以相同的度量进行衡量。当近似分布属于指数族时，KL 最小化通常可转化为矩匹配。最后，通过用新的近似后验分布除以旧的近似后验分布来更新近似项。实际应用中，EP 通常因局部更新而表现良好。不过，EP 更新有一个显著的局限性：需要通过对参数边缘化来计算归一化因子。当需要对许多参数进行边缘化，因而涉及大量积分时，这一计算通常会变得难以处理。

在介绍了带有局部更新的 EP 后，另一种观点不同的确定性近似方法是变分贝叶斯（VB）。现在，让我们更详细地探讨变分优化的概念如何应用于推断问题。假设我们有一个完整的贝叶斯模型，其中所有参数都被视为随机变量。我们的概率模型指定了联合分布 $p(\vec{x}, \vec{\theta})$，目标是找到后验分布 $\vec{\ }\ \vec{\ }\ \vec{\ }p(\theta||\vec{x})$以及模型证据 $p(\vec{x})$的近似。我们最小化 $\mathrm{KL}[q(\theta)||p(\theta||\vec{x})]$ （即 EP 中 KL 散度的反向形式）。请注意，与我们先前介绍的 EP 不同，VI 是对近似后验分布取平均。

3 变分期望传播

根据 Minka [Minka, 2005], EP 最小化前向 Kullback-Leibler 散度 KL($p||q$)；只有当两个分布属于同一个指数族——尤其是高斯族——时，才使用矩匹配。然而，许多研究凸显了当倾斜分布与近似分布完全来自不同分布族时，EP 可能多么严格且难以处理。Zoeta 和 Heskes（2005年）使用高斯求积来解决 EP 中不同分布族（如贝塔分布）之间的矩不匹配问题。此外，Roberts 和 Olobatuyi（2020年，即将发表）使用随机搜索算法来解决高斯分布与指数分布之间的矩不匹配问题。与此相反，VB 最小化反向 Kullback-Leibler 散度 KL($q||p$)。VB 已用于处理许多不相容分布的组合，方法是通过优化分布 q，间接最大化下界 $\mathcal{L}(q)$。理解这两种 Kullback-Leibler 散度之间的差异，可以注意到：在潜在空间中，当 p 接近零而q 不接近零的区域，会对 Kullback-Leibler 散度 KL($q||p$)产生很大的正向贡献。因此，最小化这种形式的 KL 散度会得到避开 p 较小区域的分布 q 。类似地，Magnus 等人（2009年）将 VB 与 EP 结合，用于贝叶斯稀疏因子分析。Kim 和 Wand（2016年）开展了一项比较研究，考察 VB 和 EP 算法得出的均值估计与方差估计的准确性。结果表明，VB 得出的均值估计往往比 EP 算法得出的均值估计更准确。与此相反，VB 得出的方差估计会低估实际方差，或不如 EP 算法得出的方差估计准确。这也证实了 [Bishop, 2006] 的研究：VB 得出的近似后验分布的均值能够很好地近似精确后验分布的均值，但其方差估计会偏低。

这里，我们证明，Kullback-Leibler 散度 KL($p||q$)小于或等于 KL($q||p$)加上任意常数，并通过局部优化体现期望传播中的留一法（LOO）方法。即:

$$\mathrm{KL}(p||q) \leq \mathrm{KL}(q||p) + \text{constant} \tag{2}$$

通过增大反向 KL 散度，将潜变量视为相互因子化的变量，从而得到纯平均场解的精度。最小化方程 2 中的 KL($p||q$)后，按照 Minka [Minka, 2001], 的矩匹配，可以将方程 30 中的归一化常数 Z 以及 p 与 q^{new} 之间的 Kullback-Leibler 散度表示为 m、 v的函数，并求得 $\log Z$ 相对于这些量的梯度，即

$$m^{new} = m_{-i} + v_{-i}\nabla_m \log Z$$

$$v^{new} = v_{-i} - v_{-i}^2\left[\left(\nabla_m \log Z\right)^2 - 2\nabla_v \log Z\right] \tag{3}$$

我们展示 Bishop（2009年）中的贝叶斯线性回归模型示例。该示例已通过变分贝叶斯方法求解。w 的似然函数和先验为

$$p(\mathbf{t}|\mathbf{w}) = \prod_{i=1}^{N}\mathcal{N}(t_i|\mathbf{w}^T\phi_i, \beta^{-1}) \text{ and } p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}) \tag{4}$$

$\vec{\ }$ $\vec{\ }$其中 $\phi_i = \phi(\mathbf{x}_i)$。 α 的先验如下

$$p(\alpha) = Gam(\alpha|a_0, b_0) \tag{5}$$

首先，我们使用 EP 算法，通过 KL(ˆ$p_i\|q_i$) 和 KL($q_i||\hat{}p_i$) 来逼近 EP。在这里，按照 EP 的方法，腔分布、近似后验和倾斜后验如下：

$$q_{-i}(\mathbf{w}) = \mathcal{N}(\mathbf{w}|m_{-i}, v_{-i}),\ q(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\vec{m}_w, \vec{v}_w),\ \text{and}\ \hat{p}_i(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\hat{m}_i, \hat{v}_i) \tag{6}$$

我们证明了 KL(ˆ$p_i\|q_i$) ≡ KL($q_i\|\hat{}p_i$)。然而，根据 Kullback-Leibler 散度的对称性可知，KL($p\|q$) ≠ KL($q\|p$)。按照传统的变分贝叶斯方法，我们首先从反向 KL 散度入手。

$$\ln q_i(\mathbf{w}) = E_\alpha\Big[\ln q_{-i}(\mathbf{w})t_i(\mathbf{w})\Big] = E_\alpha\Big[\ln \mathcal{N}(\mathbf{w}|m_{-i}, v_{-i})\mathcal{N}(t_i|\mathbf{w}^T\phi_i, \beta^{-1})\Big] \tag{7}$$

这里需要注意的是，相对于 $q(\alpha)$ 的期望为常数，因此与后续内容无关，将在下文中移除。现在，$q($w) 的均值和方差如下

$$m_w = \left(m_{-i} + v_{-i}\vec{\phi}_i t_i \beta\right)\left(1 + v_{-i}\vec{\phi}^T\vec{\phi}\beta\right)^{-1} \text{ and } v_w = v_{-i}\left(1 + v_{-i}\vec{\phi}^T\vec{\phi}\beta\right)^{-1} \tag{8}$$

现在从 正向 Kullback-Leibler 散度的方向出发，我们首先计算归一化常数

$$Z_t = \int \mathcal{N}(\mathbf{w}|m_{-i}, v_{-i})\mathcal{N}(t_i|\mathbf{w}^T\phi_i, \beta^{-1})\, dw = \mathcal{N}(m_t, v_t) \tag{9}$$

其中，t 的均值 m_t 和方差 v_t 如下

$$m_t = m_{-i}\vec{\phi}_i \text{ and } v_t = \left(\beta^{-1} + v_{-1}\vec{\phi}^T\vec{\phi}\right)^{-1} \tag{10}$$

计算 $\nabla_m \log Z_t$ 和 $\nabla_v \log Z_t$，并使用公式 9 得到如下结果

$$m_w = \left(m_{-i} + v_{-i}\vec{\phi}_i t_i \beta\right)\left(1 + v_{-i}\vec{\phi}^T\vec{\phi}\beta\right)^{-1} \text{ and } v_w = v_{-i}\left(1 + v_{-i}\vec{\phi}^T\vec{\phi}\beta\right)^{-1} \tag{11}$$

这在局部近似下确立了 KL(ˆ$p_i\|q_i$) ≡ KL($q_i\|\hat{}p_i$) 之间的等价性。

4 模型

本节重点讨论多层感知机神经网络，其中，未知函数值 $f_i = f(\mathbf{x}_i)$ 与一个 d 维输入向量 $\mathbf{x}_i$ 相关，其建模方式如下：

$$\hat{f}(\mathbf{x}_i) = \sum_{k=1}^{K} \mathbf{W}_{kL}^T g(\mathbf{W}_{kl}^T \vec{z}_{l-1}), \qquad l = 1, ..., L \tag{12}$$

其中，$g(x)$ 是非线性激活函数，K 表示隐藏单元的数量，而 $\mathcal{W} = \{\mathbf{W}_l\}_{l=1}^L$ 是网络所有权重的集合或数组，其维度为 $K_l \times (K_{l-1}+1)$，对应全连接层之间的权重。我们用向量 $\{\vec{z}_l\}_{l=0}^L$ 表示各层的输出，其中 $\vec{z}_0$ 是输入层。$\{\vec{z}_l\}_{l=1}^{L-1}$ 表示隐藏层的输出，而 $\vec{z}_L = \sigma($ˆ$f_i)$ 是输出层的输出。每个隐藏层的激活函数均为修正线性单元（RELUs），即 $a(x) = \max(x, 0)$，（[Nair 和 Hinton, 2010]）。下一小节将解释模型的似然函数。

4.1 似然定义

这里，我们说明局部变分方法在贝叶斯逻辑回归模型中的应用。重点是基于 Jaakkola and J ordan 2000年方法的变分处理。这种变分处理得到类似于拉普拉斯方法的高斯近似。然而，与拉普拉斯方法相比，变分近似具有更大的灵活性，因此能够提高准确性。此外，在模型证据具有严格界的情况下，变分方法可以被表述为对定义明确的目标函数进行优化。

二分类

从蒙特卡罗采样技术的角度研究了逻辑回归（Dybowski 和 Roberts，2005年）。对于二元输出，最后一层的输出会通过 Sigmoid 函数进行变换；对于多类别输出，则会通过 Softmax 进行变换。基于下界的变分近似使得逻辑回归的似然函数（由 Sigmoid 函数或 Softmax 控制）可以用二次型的指数函数来近似。

首先注意到， y 的条件分布可以写成

$$p(y_i|a_L,\vec{\Theta}) = \sigma(a_L)^{y_i}(1-\sigma(a_L))^{1-y_i}$$

$$= e^{-y_i a_L}\frac{e^{-a_L}}{1+e^{-a_L}} = e^{-y_i a_L}\sigma(-a_L) \tag{13}$$

↝ ^，其中 $a_L = f_i$，而 Θ 是所有超参数的集合， f 来自方程 12。式 13 中逻辑 Sigmoid 函数的变分下界为

$$\sigma(u) \geq \sigma(\zeta)\exp\{(u-\zeta)/2 - \lambda(\zeta)(u^2-\zeta^2)\} \tag{14}$$

其中 $\lambda(\zeta) = \frac{1}{2\zeta}\left[\sigma(\zeta) - \frac{1}{2}\right]$。因此，似然函数可写为

$$p(y_i|a_L,\vec{\Theta}) = e^{y_i a_L}\sigma(-a_L) \geq e^{y_i a_L}\sigma(\zeta)\exp\{-(a_L+\zeta)/2 - \lambda(\zeta)((a_L)^2-\zeta^2)\} \tag{15}$$

此外，该下界分别应用于似然函数中的每一项，因此每个训练集 ($\mathbf{x}_i, y_i$) 都对应一个变分参数 ζ_i。最后，似然函数的下界记为

$$h(\vec{\Theta},\vec{\zeta}) = \prod_{i=1}^{N}\sigma(\zeta_i)\exp\left\{y_i a_L - (a_L+\zeta_i)/2 - \lambda(\zeta_i)(a_L^2-\zeta_i^2)\right\} \tag{16}$$

似　　然函数采用的是用于二分类的 Sigmoid 函数下界，其形式为
给出　　在公式 16 中，这使得后验分布在解析上无法处理。

4.2 先验定义

我们采用促进稀疏性的先验 $p(w_{kjl}|\tau_{kjl})$，并使用层次尺度参数 τ_{kjl}^{-1}，其中权重 w_{kjl} 是 $\mathbf{W}_l$ 的第 k 行、第 j 列， τ_{kjl}^{-1} 控制所有权重 w_{kjl} 的先验方差。我们对权重设置如下高斯先验

$$p(w_{kjl}|\tau_{kjl}) = \mathcal{N}(w_{kjl}|0,\tau_{kjl}^{-1}) \tag{17}$$

其中，方差为公式 17 中的 τ_{kjl}^{-1} 。权重的分组方式可以自由选择，也可以使用其他权重先验分布来代替高斯分布。对方差参数 $\vec{\tau}_l^{-1} > 0$ 的近似推断采用非负支撑先验分布，以约束方差为非负值。为此，计算上最方便的非负支撑先验分布方案是在精度控制参数上使用截断高斯先验分布，如下所示；

$$p(\tau_{kjl}) = \frac{2}{\operatorname{erfc}(-m_0/\sqrt{v_0})}\mathcal{N}(\tau_{kjl}|m_0, v_0)U(\tau_{kjl}) \tag{18}$$

其中 $k = 1$、...、K、 $j = 1$、...、$K_{l-1} + 1$，以及 $l = 1$、...、L。公式 18 对应于相关层先验精度 $\vec{\tau}_l$ 的截断高斯先验，其中 $U(.)$ 是阶跃函数， m_0 和 v_0 分别是该精度的位置参数和尺度参数。显然，截断高斯先验与高斯似然共轭，并且由于 $\operatorname{erfc}(0) = 1$，其后验分布可以按照标准高斯分布的方式计算，因此公式 18 中有一个 2 的常数。

然而，要解决 EP 算法，截断高斯先验只有在位置参数 m_0 固定为零时才具有计算可行性，从而使互补误差函数消失。此外，我们注意到，偏置已经包含在权重矩阵的设置中。

4.3 后验分布

给定前文所述的先验定义以及一组 N 个观测值 $\mathcal{D} = \{\mathbf{X},\mathbf{y}\}$，其中 $\mathbf{y} = [y_i, ..., y_N]^T$，且特征为 $\mathbf{X} = [\mathbf{x}_1、...、\mathbf{x}_N]^T$，先验与超参数的联合后验分布如下：

$$p(\mathbf{w}, \vec{\tau}|\mathcal{D}, \vec{\zeta}, \vec{\gamma}) = Z^{-1}\prod_{i=1}^{N} h(y_i|\hat{f}_i, \zeta_i)\prod_{k=1}^{K_l}\prod_{j=1}^{K_{l-1}+1}\prod_{l=1}^{L} p(w_{kjl}|\tau_{kjl})\prod_{l=1}^{L} p(\vec{\tau}_l|\vec{\gamma}) \tag{19}$$

其中， $\vec{\gamma} = \{\zeta, m_0, v_0\}$ 包含在 EP、VB 和 VEP 算法的期望最大化版本中 E步需要计算的所有超参数，而 Z_{EP} 是边际似然Z 的近似，即观测值在给定 $\vec{\gamma}$ 条件下的边际似然，如下所示

$$Z = p(\mathbf{y}|\mathbf{X}, \vec{\zeta}, \vec{\gamma}) = \int h(\mathbf{y}|\mathbf{X}, \mathbf{w}, \vec{\zeta}, \vec{\gamma})p(\mathcal{W}|\vec{\tau})p(\vec{\tau}|\vec{\gamma})d\mathbf{w}d\vec{\tau} \tag{20}$$

5 近似推断

本节介绍如何利用 EP 变体，高效地对未知模型参数 $\vec{\mathbf{w}}$ $\vec{\tau}$ 和 ζ进行近似贝叶斯推断。首先，在第 5.1 节中，我们介绍如何使用近似项构造后验近似；随后在第 6 节中，我们讨论适用于确定参数的 VB 与 EP 算法的混合方法。

5.1 近似后验

我们为精确后验分布构造了一个解析可处理近似。在适当情况下，我们用未归一化高斯分布近似所有似然项和先验项。由于具有矩匹配的特性，高斯分布已成为近似神经网络

权重的常用分布族。(Seeger，2008；Freitas，1999)。不过，我们对神经网络权重精度的先验分布采用截断高斯分布。这一点很重要，因为它能对先验分布施加非零约束。相反，也可以考虑其他指数族分布，例如用于权重精度参数的伽马分布 (Jose and Ryan， [Jose and Ryan， 2015])。我们对公式 ?? 中的精确后验分布作如下近似：

$$p(\mathbf{w}, \vec{\tau}|\mathcal{D}, \vec{\zeta}, \vec{\gamma}) = Z^{-1}\prod_{i=1}^{N} h(y_i|\hat{f}_i, \zeta_i)\prod_{k=1}^{K_l}\prod_{j=1}^{K_{l-1}+1}\prod_{l=1}^{L} p(w_{kjl}|\tau_{kjl})p(\tau_{kjl}|\gamma_{jkl}) \tag{21}$$

$$\approx Z_{EP}^{-1}\prod_{i=1}^{N}\tilde{Z}_{y,i}\tilde{t}(\hat{f}_i)\prod_{k=1}^{K_l}\prod_{j=1}^{K_{l-1}+1}\prod_{l=1}^{L}\tilde{Z}_{kjl}^{w}\tilde{t}(w_{kjl})\tilde{t}(\tau_{kjl}) \tag{22}$$

似然项近似

˜根据公式 22，依赖于权重 w 的精确似然项通过 f_i

$$h(y_i|a_L, \zeta_i) \approx \tilde{Z}_{y,i}\tilde{t}_i(a_L|\tilde{m}_i, \tilde{v}_i) = \tilde{Z}_{y,i}\mathcal{N}(a_L|\tilde{m}_i, \tilde{v}_i) \tag{23}$$

其中 $\tilde{Z}_{y,\ n} = \int \tilde{t}(a_L|\tilde{m}_i, \tilde{v}_i)\ da_L$ 是一个标量缩放参数或归一化常数。这里，我们假设隐藏层和输出层的所有权重都包含在 w 中。请注意，符号 $\mathcal{N}$ 用于表示归一化高斯分布。请注意，我们将 S 形函数的下界近似为似然函数；该似然函数是一个在二元目标 y_i上归一化的概率分布，我们用潜变量 f_i 上的一个非归一化高斯 ˆ 分布对其进行近似。这一点很重要，因为我们关注的是 ˆ 似然如何随潜变量 f_i变化。相反，这与回归情形略有不同，回归情形使用高斯分布作为似然函数，并使用线性模型作为输出 y_i ，从而使其成为高斯分布。我们计算后验分布，以研究ˆ似然函数如何随 f_i变化。

先验项近似

所有权重 w_{ijl} for $i = 1,..., K$, $j = 1,..., K_{l-1} + 1$ 和 $l = 1,..., L$ 的先验项通常用高斯分布近似。考虑到权重精度上的先验分布结构，我们特别采用了因子分解分布，

$$p(w_{kjl}|\tau_{kjl}) \approx \tilde{Z}_{kjl}^{w}\tilde{t}(w_{kjl})\tilde{t}_{kjl}^{\tau}(\tau_{kjl}) \propto \mathcal{N}(w_{kjl}|\tilde{m}_{kjl}^{w}, \tilde{v}_{kjl}^{w})\mathcal{N}(\tau_{kjl}|\tilde{m}_{kjl}^{\tau}, \tilde{v}_{kjl}^{\tau}) \tag{24}$$

其中，与网络权重和精度分别关联的因子分解位点近似具有位置和尺度参数 $\tilde{m}_{kjl}^{w}$ 和 $\tilde{m}_{kjl}^{\tau}$, $\tilde{v}_{kjl}^{w}$, 和 $\tilde{v}_{kjl}^{\tau}$ 。此外，τ_{kjl} 的近似项也假设为截断高斯分布，其他任何指数分布也可能适用。

联合后验近似

˜独立局部似然 t_i 的乘积为

$$q(\vec{\hat{f}}) = \prod_{i=1}^{N}\tilde{Z}_{y,i}\mathcal{N}(\hat{f}_i|m_{\hat{f}}, v_{\hat{f}}) = \mathcal{N}(\vec{\hat{f}}|\vec{m}_{\hat{f}}, \vec{v}_{\hat{f}})\prod_{i=1}^{N}\tilde{Z}_{y,i} \tag{25}$$

需要使用期望传播进行多次处理的先验和超先验，是公式 22 中的因子，如下所示

$$q(\mathbf{w},\vec{\tau}) = \prod_{k=1}^{K_l}\prod_{j=1}^{K_{l-1}+1}\prod_{l=1}^{L}\tilde{Z}_{kjl}^{w}\mathcal{N}(w_{kjl}|m_{kjl}^{w},v_{kjl}^{w})\mathcal{N}(\tau_{kjl}|m_{kjl}^{\tau},v_{kjl}^{\tau}) \tag{26}$$

在公式 26 中，我们假设近似后验分布之间相互独立，并采用如下因子分解分布方法

$$q(\mathbf{w},\vec{\tau}) = q(\mathbf{w})q(\vec{\tau}) \tag{27}$$

其中，网络权重的近似后验分布如下

$$q(\mathbf{w}) = \prod_{k=1}^{K_l}\prod_{j=1}^{K_{l-1}+1}\prod_{l=1}^{L}\mathcal{N}(w_{kjl}|m_{kjl}^{w},v_{kjl}^{w}) \tag{28}$$

从概念上讲，可以通过两种方式理解超先验 τ 的近似后验分布：一种是将近似项与超先验分布结合起来，得到如下结果

$$q(\vec{\tau}) = \prod_{k=1}^{K_l}\prod_{j=1}^{K_{l-1}+1}\prod_{l=1}^{L}\mathcal{N}(\tau_{kjl}|m_{kjl}^{\tau},v_{kjl}^{\tau}) \tag{29}$$

Now, 将 w 的参数近似与 $\vec{\tau}$ 一起乘以公式中的先验 28 以及 2 9 得到近似后验

$$q(\mathbf{w}) \propto \mathcal{N}(\vec{M}_w,\vec{V}_w) \quad \text{and} \quad q(\vec{\tau}) \propto \mathcal{N}(\vec{M}_\tau,\vec{V}_\tau)U(\tau) \tag{30}$$

其中，采用高斯乘法策略可得

$$\vec{M}_\tau = \vec{V}_\tau\tilde{\vec{V}}_\tau^{-1}\tilde{\vec{M}}_\tau \quad \text{and} \quad \vec{V}_\tau = \left(\vec{V}_0^{-1}+\tilde{\vec{V}}_\tau^{-1}\right)^{-1} \tag{31}$$

其中，精度 τ_{kjl} 的边际后验分布为

$$q(w_{kjl}) \propto \mathcal{N}(m_{kjl}^{w},v_{kjl}^{w}) \quad \text{and} \quad q(\tau_{kjl}) \propto \mathcal{N}(m_{kjl}^{\tau},v_{kjl}^{\tau})U(\tau) \tag{32}$$

其中， m_{kjl}^{w} 和 v_{kjl}^{w} 是网络权重 $q(w_{kjl})$的近似分布的均值和方差参数，而 m_{kjl}^{τ} 和 v_{kjl}^{τ} 是精度 $q(\tau_{kjl})$的近似分布的均值和方差参数。近似后验$\vec{\ }$的均值向量 $\vec{M}_\tau$ 是由 m_{kjl}^{τ} 组成的向量，而近似后验 V_τ 的协方差为对角矩阵，其 v_{kjl}^{τ} 适用于近似后验。

6 变分贝叶斯与期望传播的混合

定义近似后验分布的局部站点近似参数，通过变分贝叶斯与期望传播的混合来确定。下面我们将概述似然和权重先验项的期望传播更新。这里，我们考虑按顺序更新的期望传播。

6.1 超先验项的 EP 更新

如上文公式 24 所述，每个精确先验因子都由相应的近似先验近似表示，其形式为

$$t(w_{kjl}, \tau_{kjl}) = \mathcal{N}(w_{kjl}|0, \tau_{kjl}^{-1}) \quad \text{and} \quad \tilde{t}(\tau_{kjl}) = \mathcal{N}(\tau_{kjl}|\tilde{m}_{kjl}^{\tau}, \tilde{v}_{kjl}^{\tau}) \tag{33}$$

首先，我们将所有 $\tilde{t}(\tau_{kjl})$ 初始化为均匀分布，即 $\tilde{m}_{kjl}^{\tau} = 0$ 和 $\tilde{v}_{kjl}^{\tau} = \infty$。EP 开始将所有精确的先验因子 $t(w_{kjl}, \tau_{kjl})$ 纳入 q，在 $K \times (K_{l-1}+1) \times L$ 次迭代中进行。这里，我们关注每个权重对应的单独精度参数。这一点很重要，因为它能够反映每层每个单元更新时权重估计的准确程度。这种方法唯一的缺点是内存效率低下。不过，我们不会将每次更新存储在内存中。首次将 $t(w_{kjl}, \tau_{kjl})$ 纳入 q 时，我们会按如下方式更新 $\tilde{t}(\tau_{kjl})$ 和 q：

$$\tilde{m}_{kjl}^{\tau} = 0 \quad \text{and} \quad \tilde{v}_{kjl}^{\tau} = v_0, \quad m_{kjl}^{\tau} = 0 \quad \text{and} \quad v_{kjl}^{\tau} = v_0 \tag{34}$$

其中，v_0 是 τ 上截断高斯超先验的参数。在后续迭代中，我们首先从 τ 的近似后验中移除近似因子，以得到腔分布，从而对 $\tilde{t}(\tau_{kjl})$ 进行细化。该腔分布计算为 q 与 t 的比值。因此，τ_{kjl} 上的腔边缘分布为

$$q_{-kjl}(\tau_{kjl}) = q(\tau_{kjl})\tilde{t}(\tau_{kjl})^{-1} = \mathcal{N}(\tau_{kjl}|m_{-kjl}^{\tau}, v_{-kjl}^{\tau}) \tag{35}$$

其中，m_{-kjl}^{τ} 和 v_{-kjl}^{τ} 如下：

$$(v_{-kjl}^{\tau})^{-1} = (v_{kjl}^{\tau})^{-1} - (\tilde{v}_{kjl}^{\tau})^{-1}$$

$$m_{-kjl}^{\tau} = m_{kjl}^{\tau} + (\tilde{v}_{kjl}^{\tau})^{-1} v_{-kjl}^{\tau}(m_{kjl}^{\tau} - \tilde{m}_{kjl}^{\tau}) \tag{36}$$

w_{kjl} 的边缘分布的腔分布也为

$$q_{-kjl}(w_{kjl}) = q(w_{kjl})\tilde{t}(w_{kjl})^{-1} = \mathcal{N}(w_{kjl}|m_{-kjl}^{w}, v_{-kjl}^{w}) \tag{37}$$

其中，m_{-kjl}^{w} 和 v_{-kjl}^{w} 如下：

$$(v_{-kjl}^{w})^{-1} = (v_{kjl}^{w})^{-1} - (\tilde{v}_{kjl}^{w})^{-1}$$

$$m_{-kjl}^{w} = m_{kjl}^{w} + (\tilde{v}_{kjl}^{w})^{-1} v_{-kjl}^{w}(m_{kjl}^{w} - \tilde{m}_{kjl}^{w}) \tag{38}$$

6.2 计算 τ_{kjl}和 w_{kjl}的倾斜分布

纳入所有先验因子后，我们计算倾斜后验分布 $\hat{p}(\tau_{kjl})$。倾斜分布由腔分布与精确先验项 $t(\tau_{kjl})$ 组合而成：

$$\hat{p}(\tau_{kjl}) = \hat{Z}_w^{-1} q_{-kjl} t(\tau_{kjl}) p(\tau_{kjl}) = \mathcal{N}(\tau_{kjl}|\hat{m}_{kjl}^{\tau}, \hat{v}_{kjl}^{\tau}) \tag{39}$$

其中，归一化因子 Z_w 如下所示

$$Z_w = \int t(\tau_{kjl}) p(\tau_{kjl}) q_{-kjl} \, d\tau_{kjl} = \int \mathcal{N}(\tau_{kjl}|0, v_0^{\tau}) \mathcal{N}(\tau_{kjl}|m_{-kjl}^{\tau}, v_{-kjl}^{\tau}) U(\tau_{kjl}) \, d\tau_{kjl} \tag{40}$$

我们通过变分贝叶斯的 KL散度($q||p$)计算 $\log Z_w$ ，而不是直接通过 KL散度($p||q$)计算 Z 。我们根据 KL散度($q||p$)计算 $\log Z_w$，其中 $\theta_{kjl} = (w_{kjl}, \tau_{kjl})$ 如下所示:

$$-\mathrm{KL}(q_{kjl}||\hat{p}_{kjl}) = \int q_{kjl}(\theta_{kjl}) \log \frac{\hat{p}_{kjl}(\theta_{kjl})}{q_{kjl}(\theta_{kjl})} d\theta_{kjl} \tag{41}$$

$$= \int q_{kjl}(\theta_{kjl}) \left[\log \left(\frac{t_{kjl}(\theta_{kjl}) p(\tau_{kjl}) q_{-kjl}(\theta_{kjl})}{Z_w} \right) - \log q_{kjl}(\theta_{kjl}) \right] d\theta_{kjl} \tag{42}$$

通过重新排列公式 42，我们得到

$$-\mathrm{KL}(q_{kjl}||\hat{p}_{kjl}) = \int q_{kjl}(\theta_{kjl}) \log \left(\frac{t_{kjl}(\theta_{kjl}) p(\tau_{kjl}) q_{-kjl}(\theta_{kjl})}{q_{kjl}(\theta_{kjl})} \right) d\theta_{kjl} - \log Z_w \tag{43}$$

将公式变形以求出 $\log Z_w$ ，然后重新排列，我们得到

$$\log Z_w = \int q_{kjl}(\theta_{kjl}) \log \left(\frac{t_{kjl}(\theta_{kjl}) p(\tau_{kjl}) q_{-kjl}(\theta_{kjl})}{q_{kjl}(\theta_{kjl})} \right) d\theta_{kjl} + \mathrm{KL}(q_{kjl}||\hat{p}_{kjl}) \tag{44}$$

其中

$$\mathcal{L}(\tau_{kjl}) = \int q_{kjl}(\theta_{kjl}) \log \left(\frac{t_{kjl}(\theta_{kjl}) p(\tau_{kjl}) q_{-kjl}(\theta_{kjl})}{q_{kjl}(\theta_{kjl})} \right) d\theta_{kjl} \tag{45}$$

然后使用因子化方法 $q_{kjl}(\theta_{kjl}) = q_{kjl}(w_{kjl}, \tau_{kjl}) = q_{kjl}(w_{kjl}) q_{kjl}(\tau_{kjl})$

$$\log Z_w = \mathcal{L}(\theta_{kjl}) + \mathrm{KL}(q_{kjl}||\hat{p}_{kjl}) \tag{46}$$

6.3 超先验参数 τ_{kjl}

这里，与变分贝叶斯一样，我们最大化公式 46 中的下界，并使用以下因子化方法对 $q(w_{jkl})$ 求期望。因此，最小化 Kullback-Leibler 散度等同于最大化下界；我们选取所有仅依赖于 τ_{kjl} 的精确分布，并得到最优解 $q(\tau_{kjl})$ 的一般表达式如下

$$\ln q^*(\tau_{kjl}) = \mathrm{E}_w \left[t(w_{kj}, \tau_{kjl}) p(\tau_{kjl}) q_{-kjl}(\tau) \right] + \mathrm{const} \tag{47}$$

$$m^\tau_{kjl} = v^\tau_{kjl} (v^\tau_{-kjl})^{-1} m^\tau_{-kjl} - \frac{1}{2} v^\tau_{kjl} (v^w_{kjl} + [m^w_{kjl}]^2) \tag{48}$$

其中，我们使用了相对于 q_w 和 $\mathrm{E}(w^2) = v_w + m_w^2$ 的期望，于是公式 48 变为

$$m^\tau_{kjl} = \left[m^\tau_{-kjl} - \frac{1}{2} v^\tau_{-kjl} (v^w_{kjl} + [m^w_{kjl}]^2) \right] \frac{v_0}{v^\tau_{-kjl} + v_0} \quad \text{and} \quad v^\tau_{kjl} = \left(v_0^{-1} + (v^\tau_{-kjl})^{-1} \right)^{-1} \tag{49}$$

6.4 先验权重 w_{kjl}

我们根据变分期望传播的设置计算近似后验均值 m_w 和方差 v_w 。我们使用相对于近似后验$q(\tau_{kjl})$ 的期望，并将所有仅取决于权重 w_{kjl} 的项分解如下

$$\ln q^*(w_{kjl}) = \mathrm{E}_\tau \left[t(w_{kjl}, \tau_{kjl}) q_{-kjl}(w_{kjl}) \right] + \mathrm{const} \tag{50}$$

然后，近似后验 $q(w_{kjl})$ 的均值和方差计算如下

$$m^w_{kjl} = m^w_{-kjl}(v^w_{-kjl})^{-1}v^w_{kjl} \quad \text{and} \quad v^w_{kjl} = \left[(v^w_{-kjl})^{-1} + m^\tau_{kjl}\right]^{-1} \tag{51}$$

在公式 51 中，$\mathrm{E}[\tau_{kjl}] = m^\tau_{kjl}$。最后，我们更新近似因子 $\tilde{t}(\tau_{kjl})$ 和 $\tilde{t}(w_{kjl})$ 的参数

$$\tilde{v}^\tau_{kjl} = \left[(v^\tau_{kjl})^{-1} - (v^\tau_{-kjl})^{-1}\right]^{-1} \quad \text{and} \quad \tilde{m}^\tau_{kjl} = \tilde{v}^\tau_{kjl}\left[m^\tau_{kjl}(v^\tau_{kjl})^{-1} - m^\tau_{-kjl}(v^\tau_{-kjl})^{-1}\right] \tag{52}$$

$$\tilde{v}^w_{kjl} = \left[(v^w_{kjl})^{-1} - (v^w_{-kjl})^{-1}\right]^{-1} \quad \text{and} \quad \tilde{m}^w_{kjl} = \tilde{v}^w_{kjl}\left[m^w_{kjl}(v^w_{kjl})^{-1} - m^w_{-kjl}(v^w_{-kjl})^{-1}\right] \tag{53}$$

分别。

6.5 似然项的 EP 更新

这里，我们考虑更新公式 23 中定义的似然站点 $\tilde{t}(w_{kjl})$ 和近似后验$q(w_{kjl})$ 的过程。精确似然项 $p(y_i|f_i)$ 是逻辑回归模型，并由泰勒级数的下界 $h(\vec{\Theta}, \vec{\zeta})$ 近似，该下界不依赖于权重精度 τ_{kjl}。后验近似可分解为 $q(z_{kjl}, a_{kjl}, w_{kjl})$，而精确似然为 $p(y_i|a_L) = t(a_L)$，其中 a_L 是最后一层的矩阵乘法。现在，我们按如下方式计算 Z_y

$$Z_y = \int h(y_i|a_L)\mathcal{N}(a_L|m^{a_L}_{kj}, v^{a_L}_{kj})\, da_L \tag{54}$$

$$= \exp\left\{y_i a_L - (a_L - \zeta_i)/2 - \lambda(\zeta_i)(a_L^2 - \zeta_i^2) - \frac{1}{2v^{a_L}_{kj}}(a_L^2 - m^{a_L}_{kj})^2\right\} \tag{55}$$

通过对矩阵乘法 a_L 积分消去，我们得到

$$Z_y = \mathcal{N}(y_i|m_y, v_y) \tag{56}$$

其中，均值 m_y 和 v_y 如下

$$m_y = \frac{1}{2} - \frac{m^{a_L}_{kj}}{v^{a_L}_{kj}} \quad \text{and} \quad v_y = \frac{1}{\zeta_i}\left[\frac{1}{2} - \sigma(\zeta_i)\right] - \frac{1}{v^{a_L}_{kj}} \tag{57}$$

注意，我们使用了

$$\lambda(\zeta_i) = \frac{1}{2\zeta_i}\left[\sigma(\zeta_i) - \frac{1}{2}\right]$$

公式 25 中 $q(f_i) = q(a_L)$ 的近似后验均值和方差的更新规则如下

$$m^{a_L}_{new} = m^{a_L}_{\text{old}} + [m_y - y_i]v_y^{-1} \text{ and } v^{a_L}_{new} = v^{a_L}_{old} + v_y^{-1}[2m^{a_L}_{old}(y_i - m_y) - 1] \tag{58}$$

7 概率反向传播

本节将介绍适用于该模型的概率反向传播算法。PBP 不会像 Jose and Ryan [Jose and Ryan, 2015] 那样，对网络中的突触权重使用点估计。相反，它使用一组一维高斯分布，每个分布都近似一个不同权重的边缘后验分布。PBP 还有两个阶段，与 BP 的阶段相对应。在第一阶段，输入数据在网络中向前传播。然而，由于权重现在是随机的，每一层产生的激活值也具有随机性，因此会形成（难以处理的）分布。PBP 依次用一组一维高斯分布近似这些分布中的每一个，使其匹配相应的边缘均值和方差。在这一阶段结束时，PBP 计算的不是预测误差，而是目标变量边缘概率的对数。在第二阶段，使用与经典反向传播相同的反向模式微分，将该量相对于近似高斯后验的均值和方差的梯度反向传播。最后，利用这些导数更新后验近似的均值和方差。

7.1 梯度的推导

我们推导边际似然对数的梯度关于高斯近似后验 q 中网络权重的均值和方差的梯度，即公式 54 中给出的 $\log Z_y$。在 PBP 中，相应的算法对每个神经元包含两个变量，即均值和方差。每一层使用的激活函数都是 ReLU激活，由于权重现在是随机的，激活值也会变为随机变量。每一层的输出表示为z_l，矩阵乘法表示为 a_l，其中 $l = 1$、...、L。最后一层使用的激活函数是 S形函数 $\sigma(.)$。我们从输入层开始，沿网络向前传播至最后一层，即输出层。在介绍 PBP 的一般概念之前，暂且假设我们有 $L = 3$ 。

7.2 前向传播

考虑一类由以下函数形式定义的神经网络

$$z_L = \frac{1}{1 + \exp[-a_L]} \tag{59}$$

其中 a_L 表示最后一层的矩阵乘法，明确写作如下

$$a_L = \vec{w}_L^T g_L \left[\sum_{l=1}^{L-1} \sum_{k=1}^{K} g_l \left(\sum_{j=1}^{J} w_{kjl} \vec{z}_{l-1} \right) \right] \tag{60}$$

$$\text{For Layer } L = 1, \ \mathrm{E}\left[\mathbf{w}_1^T z_0\right] \text{ and } m_{kj}^w m^{z_0} \tag{61}$$

7.3 反向传播

对于最后一层,

$$\frac{\partial Z_y}{\partial m_{kj}^{w_L}} = \frac{\partial \log Z_y}{\partial m_{kj}^{a_L}} \frac{\partial m_{kj}^{a_L}}{\partial m_{kj}^{w_L}} + \frac{\partial \log Z_y}{\partial v_{kj}^{a_L}} \frac{\partial v_{kj}^{a_L}}{\partial m_{kj}^{w_L}}, \ \text{ and } \ \frac{\partial Z_y}{\partial v_{kj}^{w_L}} = \frac{\partial \log Z_y}{\partial m_{kj}^{a_L}} \frac{\partial m_{kj}^{a_L}}{\partial v_{kj}^{w_L}} + \frac{\partial \log Z_y}{\partial v_{kj}^{a_L}} \frac{\partial v_{kj}^{a_L}}{\partial v_{kj}^{w_L}} \tag{62}$$

对于隐藏层

$$\frac{\partial Z_y}{\partial m_{kj}^{w_l}} = \frac{\partial \log Z_y}{\partial m_{kj}^{a_l}} \frac{\partial m_{kj}^{a_l}}{\partial m_{kj}^{w_l}} + \frac{\partial \log Z_y}{\partial v_{kj}^{a_l}} \frac{\partial v_{kj}^{a_l}}{\partial m_{kj}^{w_l}}, \text{ and } \frac{\partial Z_y}{\partial v_{kj}^{w_l}} = \frac{\partial \log Z_y}{\partial m_{kj}^{a_l}} \frac{\partial m_{kj}^{a_l}}{\partial v_{kj}^{w_l}} + \frac{\partial \log Z_y}{\partial v_{kj}^{a_l}} \frac{\partial v_{kj}^{a_l}}{\partial v_{kj}^{w_l}} \tag{63}$$

其中

$$\frac{\partial \log Z_y}{\partial m_{kj}^{a_l}} = \frac{\partial \log Z_y}{\partial m_{kj}^{a_{l+1}}} \frac{\partial m_{kj}^{a_{l+1}}}{\partial m_{kj}^{a_l}} + \frac{\partial \log Z_y}{\partial v_{kj}^{a_{l+1}}} \frac{\partial v_{kj}^{a_{l+1}}}{\partial m_{kj}^{a_l}}, \tag{64}$$

$$\frac{\partial \log Z_y}{\partial v_{kj}^{a_l}} = \frac{\partial \log Z_y}{\partial m_{kj}^{a_{l+1}}} \frac{\partial m_{kj}^{a_{l+1}}}{\partial v_{kj}^{a_l}} + \frac{\partial \log Z_y}{\partial v_{kj}^{a_{l+1}}} \frac{\partial v_{kj}^{a_{l+1}}}{\partial v_{kj}^{a_l}} \tag{65}$$

$$\frac{\partial m_{kj}^{a_{l+1}}}{\partial m_{kj}^{a_l}} = \frac{\partial m_{kj}^{a_{l+1}}}{\partial m_{kj}^{z_l}} \frac{\partial m_{kj}^{z_l}}{\partial m_{kj}^{a_l}} + \frac{\partial m_{kj}^{a_{l+1}}}{\partial v_{kj}^{z_l}} \frac{\partial v_{kj}^{z_l}}{\partial m_{kj}^{a_l}}, \text{ and } \frac{\partial m_{kj}^{a_{l+1}}}{\partial v_{kj}^{a_l}} = \frac{\partial m_{kj}^{a_{l+1}}}{\partial m_{kj}^{z_l}} \frac{\partial m_{kj}^{z_l}}{\partial v_{kj}^{a_l}} + \frac{\partial m_{kj}^{a_{l+1}}}{\partial v_{kj}^{z_l}} \frac{\partial v_{kj}^{z_l}}{\partial v_{kj}^{a_l}} \tag{66}$$

$$\frac{\partial v_{kj}^{a_{l+1}}}{\partial m_{kj}^{a_l}} = \frac{\partial v_{kj}^{a_{l+1}}}{\partial m_{kj}^{z_l}} \frac{\partial m_{kj}^{z_l}}{\partial m_{kj}^{a_l}} + \frac{\partial v_{kj}^{a_{l+1}}}{\partial v_{kj}^{z_l}} \frac{\partial v_{kj}^{z_l}}{\partial m_{kj}^{a_l}}, \text{ and } \frac{\partial v_{kj}^{a_{l+1}}}{\partial v_{kj}^{a_l}} = \frac{\partial v_{kj}^{a_{l+1}}}{\partial m_{kj}^{z_l}} \frac{\partial m_{kj}^{z_l}}{\partial v_{kj}^{a_l}} + \frac{\partial v_{kj}^{a_{l+!}}}{\partial v_{kj}^{z_l}} \frac{\partial v_{kj}^{z_l}}{\partial v_{kj}^{a_l}} \tag{67}$$

每一层矩阵乘法输出的均值和方差分别定义为 $m_{kj}^{a_l}$ 和 $v_{kj}^{a_l}$。此外，作为下一层输入的激活函数的均值和方差分别定义为 $m_{kj}^{z_l}$ 和 $v_{kj}^{a_l}$。首先，根据公式 61，通过计算一阶矩和二阶矩对矩阵乘法进行随机化，具体如下

$$E[a_2] = E[w_{kj}^2]E[z^2] = (\text{Var}(w) + (E[w])^2)(\text{Var}(z) + (E[z])^2) \tag{68}$$

一阶矩和二阶矩如下所示

$$m_{kj}^{a_l} = m_{kj}^{z_{l-1}} m_{kj}^{w_l} \tag{69}$$

$$v_{kj}^{a_l} = (m_{kj}^{z_{l-1}})^2 v_{kj}^{w_l} + v_{kl}^{z_{l-1}} (m_{kj}^{w_l})^2 + v_{kj}^{z_{l-1}} v_{kj}^{w_l} \tag{70}$$

随机化 ReLU 激活函数如下

$$m_{kj}^{z_l} = \vec{\Phi}(\alpha_{kj}) \left[m_{kj}^{a_l} + \sqrt{v_{kj}^{a_l}} \gamma_{kj} \right] \tag{71}$$

$$v_{kj}^{z_l} = m_{kj}^{z_l} \left[m_{kj}^{a_l} + \sqrt{v_{kj}^{a_l}} \gamma_{kj} \right] \vec{\Phi}(-\alpha_{kj}) + \vec{\Phi}(\alpha_{kj}) v_{kj}^{a_l} (1 - \gamma_{kj}^2 - \gamma_{kj}\alpha_{kj}) \tag{72}$$

其中，$\gamma_{kj} = \phi\left(\alpha_{kj}\right) / \vec{\Phi}\left(\alpha_{kj}\right)$、$\alpha_{kj} = m_{kj}^{a_l} / \sqrt{v_{kj}^{a_l}}$（其中 ϕ）和 $\vec{\Phi}$分别表示标准高斯概率密度函数和累积分布函数。从公式 54 中的归一化常数 Z_y 开始，梯度如下

$$\frac{\partial \log Z_y}{\partial m_{kj}^{a_L}} = [m_y - y_i] v_y^{-1} (v_{kj}^{a_L})^{-1} \tag{73}$$

$$\frac{\partial \log Z_y}{\partial v_{kj}^{a_L}} = \frac{1}{(v_{kj}^{a_L})^2} \left[\frac{1}{2v_y^2} (y_i - m_y)^2 + \frac{m_{kj}^{a_L}}{v_y} (y_i - m_y) - \frac{1}{2v_y} \right] \tag{74}$$

为方便起见，这里提前给出公式 54

$$m_y = \frac{1}{2} - m_{kj}^{a_L} (v_{kj}^{a_L})^{-1} \text{ and } v_y = \frac{1}{\zeta_i} \left[\frac{1}{2} - \sigma(\zeta_i) \right] - \frac{1}{v_{kj}^{a_L}} \tag{75}$$

这是感兴趣的上层相对于下层的梯度 a_l 和 z_{l-1}

$$\frac{\partial m_{kj}^{a_l}}{\partial m_{kj}^{z_{l-1}}} = m_{kj}^{w_l},\ \ \frac{\partial m_{kj}^{a_l}}{\partial v_{kj}^{z_{l-1}}} = 0,\ \ \frac{\partial v_{kj}^{a_l}}{\partial m_{kj}^{z_{l-1}}} = 2m_{kj}^{z_{l-1}} v_{kj}^{w_l},\ \text{ and }\ \frac{\partial v_{kj}^{a_l}}{\partial v_{kj}^{z_{l-1}}} = (m_{kj}^{w_l})^2 + v_{kj}^{w_l}. \tag{76}$$

这是相对于同一关注层的梯度 a_l 和 w_l

$$\frac{\partial m_{kj}^{a_l}}{\partial m_{kj}^{w_l}} = m_{kj}^{z_{l-1}},\ \ \frac{\partial m_{kj}^{a_l}}{\partial v_{kj}^{w_l}} = 0,\ \ \frac{\partial v_{kj}^{a_l}}{\partial m_{kj}^{w_l}} = 2v_{kj}^{z_{l-1}} m_{kj}^{w_l},\ \text{ and }\ \frac{\partial v_{kj}^{a_l}}{\partial v_{kj}^{w_l}} = (m_{kj}^{z_{l-1}})^2 + v_{kj}^{z_{l-1}}. \tag{77}$$

这是相对于同一关注层的梯度 z_l 和 a_l

$$\frac{\partial m^{z_l}}{\partial m^{a_l}} = \vec{\Phi}(\alpha_{kj})\left[1 + \sqrt{v_{kj}^{a_l}}\frac{\partial \gamma_{kj}}{\partial m_{kj}^{a_l}}\right] + \frac{\partial \alpha_{kj}}{\partial m_{kj}^{a_l}}\left[m_{kj}^{a_l} + \sqrt{v_{kj}^{a_l}}\gamma_{kj}\right]\phi(\alpha_{kj}) \tag{78}$$

$$\frac{\partial m_{kj}^{z_l}}{\partial v_{kj}^{a_l}} = \frac{\partial \alpha_{kj}}{\partial v_{kj}^{a_l}}\left[m_{kj}^{a_l} + \sqrt{v_{kj}^{a_l}}\gamma_{kj}\right]\phi(\alpha_{kj}) + \vec{\Phi}(\alpha_{kj})\left[\sqrt{v_{kj}^{a_l}}\frac{\partial \gamma_{kj}}{\partial v_{kj}^{a_l}} + \frac{\gamma_{kj}}{2\sqrt{v_{kj}^{a_l}}}\right] \tag{79}$$

$$\begin{aligned}\frac{\partial v_{kj}^{z_l}}{\partial m_{kj}^{a_l}} = {} & m_{kj}^{z_l}\left[1 + \sqrt{v_{kj}^{a_l}}\frac{\partial \gamma_{kj}}{\partial m_{kj}^{a_l}}\right]\vec{\Phi}(-\alpha_{kj}) + \frac{\partial \alpha_{kj}}{\partial m_{kj}^{a_l}}\phi(\alpha_{kj}) v_{kj}^{a_l}(1 - \gamma_{kj}^2 - \alpha_{kj}\gamma_{kj}) \\ & - \left[m_{kj}^{a_l} + \sqrt{v_{kj}^{a_l}}\gamma_{kj}\right]\left[m_{kj}^{z_l}\phi(\alpha_{kj})\frac{\partial \alpha_{kj}}{\partial m_{kj}^{a_l}} - \vec{\Phi}(-\alpha_{kj})\frac{\partial m_{kj}^{z_l}}{\partial m_{kj}^{a_l}}\right] \\ & - \vec{\Phi}(\alpha_{kj}) v_{kj}^{a_l}\left[2\gamma_{kj}\frac{\partial \gamma_{kl}}{\partial m_{kj}^{a_l}} + \alpha_{kj}\frac{\partial \gamma_{kj}}{\partial m_{kj}^{a_l}} + \gamma_{kj}\frac{\partial \alpha_{kj}}{\partial m_{kj}^{a_l}}\right]\end{aligned} \tag{80}$$

$$\begin{aligned}\frac{\partial v_{kj}^{z_l}}{\partial v_{kj}^{a_l}} = {} & \vec{\Phi}(\alpha_{kj})\left\{\left[1 - \gamma_{kj}^2 - \alpha_{kj}\gamma_{kj}\right]\left[1 + v_{kj}^{a_l}\gamma_{kj}\frac{\partial \alpha_{kj}}{\partial v_{kj}^{a_l}}\right] - v_{kj}^{a_l}\left[2\gamma_{kj}\frac{\partial \gamma_{kl}}{\partial v_{kj}^{a_l}} + \alpha_{kj}\frac{\partial \gamma_{kj}}{\partial v_{kj}^{a_l}} + \gamma_{kj}\frac{\partial \alpha_{kj}}{\partial v_{kj}^{a_l}}\right]\right\} \\ & + m_{kj}^{z_l}\left\{\left[\sqrt{v_{kj}^{a_l}}\frac{\partial \gamma_{kj}}{\partial v_{kj}^{a_l}} + \frac{\gamma_{kj}}{2\sqrt{v_{kj}^{a_l}}}\right]\vec{\Phi}(-\alpha_{kj}) - \left[m_{kj}^{a_l} + \sqrt{v_{kj}^{a_l}}\gamma_{kj}\right]\left[\phi(\alpha_{kj})\frac{\partial \alpha_{kj}}{\partial v_{kj}^{a_l}} - \frac{\vec{\Phi}(-\alpha_{kj})}{m_{kj}^{z_l}}\frac{\partial m_{kj}^{z_l}}{\partial v_{kj}^{a_l}}\right]\right\}\end{aligned} \tag{81}$$

现在我们计算相对于 $m_{kj}^{a_l}$ 和 $v_{kj}^{a_l}$的 γ 和 α

$$\frac{\partial \gamma_{kj}}{\partial m_{kj}^{a_l}} = -\frac{\partial \alpha_{kj}}{\partial m_{kj}^{a_l}}\left[\alpha_{kj}\gamma_{kj}(\alpha_{kj}) + \gamma_{kj}^2(\alpha_{kj})\right] \quad \text{and} \quad \frac{\partial \alpha_{kj}}{\partial m_{kj}^{a_l}} = \frac{1}{\sqrt{v_{kj}^{a_l}}} \tag{82}$$

$$\frac{\partial \gamma_{kj}}{\partial v_{kj}^{a_l}} = -\frac{\partial \alpha_{kj}}{\partial v_{kj}^{a_l}}\left[\alpha_{kj}\gamma_{kj}(\alpha_{kj}) + \gamma_{kj}^2(\alpha_{kj})\right] \quad \text{and} \quad \frac{\partial \alpha_{kj}}{\partial v_{kj}^{a_l}} = -\frac{m_{kj}^{a_l}}{2v_{kj}^{a_l}\sqrt{v_{kj}^{a_l}}} \tag{83}$$

参考文献

[Bishop, 2006] Bishop, M. C. (2006).《模式识别与机器学习》（信息科学与统计学）。Springer-Verlag New York, Inc.，美国新泽西州锡考克斯。

[Bottou, 1986] Bottou, L. (1986). 利用随机梯度下降进行大规模机器学习。国际计算统计会议，巴黎：https://doi.org/10.1007/978-3-7908- 2604-3-16，323:177–187。[Collobert and Weston, 2008] Collobert, R. 和 Weston, J. (2008). 一种用于自然语言处理的统一架构：采用多任务学习的深度神经网络。载于第25届国际机器学习会议论文集，25:160–167。[Graves, 2011] Graves, A. (2011). 神经网络的实用变分推断。载于神经信息处理系统进展，24:2348–2356。[Hannun et al., 2014] Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A. 和 Ng, A. Y. (2014). 深度语音：扩展端到端语音识别。百度研究院 – 硅谷人工智能实验室：arXiv:1412.5567v2[cs.CL] *19* 2014年12月，2:1–12。[Hernandez-Lobato, 2010] Hernandez-Lobato, J. M. (2010). 平衡机器学习中的灵活性与鲁棒性：半参数方法和稀疏线性模型。博士论文：Universidad Aut ´onoma de Madrid，第1–172页。[Hern ´andez-Lobato et al., 2008] Hern ´andez-Lobato, J. M., Dijkstra, T. 和 Heskes, T. (2008). 从疟原虫的基因表达时间序列中发现调控因子：一种层次化方法。神经信息处理系统进展，20:649–656。[Hinton and Camp, 1993] Hinton, G. 和 Camp, D. V. (1993). 通过最小化权重的描述长度来保持神经网络简单。载于第六届计算学习理论年会论文集，第5–13页。[Hinton et al., 2012] Hinton, G., Li, D., Dong, Y., G.-D., Abdel-rahman, M., Navdeep, J., Andrew, S., Vincent, V., Nguyen, P., Brian, K. 和 Tara, S. (2012). 用于语音识别中声学建模的深度神经网络。IEEE信号处理杂志，29:82–97。[Hornik et al., 1989] Hornik, K., Stinchcombe, M. 和 White, H. (1989). 多层前馈网络是通用逼近器。神经网络：http://dx.doi.org/10.1016/0893-6080(89)90020-8，2:359–366。[Jose and Ryan, 2015] Jose, M. H. 和 Ryan, A. P. (2015). 用于贝叶斯神经网络可扩展学习的概率反向传播。arXiv:1502.05336v*2* [stat.ML]。第1–15页。[Jylanki et al., 2014] Jylanki, P., Mummenmaa, A. 和 Vehtari, A. (2014). 针对具有促进稀疏性的先验的神经网络的期望传播。机器学习期刊，15(1):1849– 1901。[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I. 和 Hinton, G. E. (2012). 使用深度卷积神经网络进行Imagenet分类。神经信息处理系统，141:1097– 1105。[MacKay, 1992] MacKay, D. (1992). 应用于分类网络的证据框架。神经计算：https://doi.org/10.1162/neco.1992.4.5.720，4(5):720–736。[MacKay, 995a] MacKay, D. (1995a). 贝叶斯神经网络与密度网络。《核仪器与物理研究方法》，354(1):73–80。

[Minka, 2001] Minka，T.（2001年）。用于近似贝叶斯推断的期望传播。载于《第十七届人工智能不确定性会议论文集》，J. Breese 和 D Koller 编，第20卷，第362–369页。XML。[Minka, 2005] Minka，T.（2005年）。散度度量与消息传递。技术报告，Micorsoft Research，剑桥。1：1–7。[Nair 和 Hinton, 2010] Nair，V. 和 Hinton，G. E.（2010）。修正线性单元改进了受限玻尔兹曼机。载于《第27届国际机器学习会议论文集》，27：807–814。[Neal, 1995] Neal，R. M.（1995）。神经网络的贝叶斯学习。博士论文，多伦多大学，1。[Neal, 1996] Neal，R. M.（1996）。《神经网络的贝叶斯学习》。Springer Science+Business Media New York。Springer New York，NY，第1版。[Qi 等, 2004] Qi，Y. A.、Minka，T. P.、Picard，R. 和 Ghahramani，Z.（2004）。通过期望传播进行预测性自动相关性确定。载于《第二十一届国际机器学习会议论文集》，第85–92页。[Rasmussen 和 Williams, 2006] Rasmussen，C. E. 和 Williams，C. K. I.（2006）。《机器学习中的高斯过程》。自适应计算与机器学习。麻省理工学院出版社，第1版。[Rumelhart 等, 1986] Rumelhart，D. E.、Hinton，G. E. 和 Williams，R. J.（1986）。通过误差反向传播学习表征。《自然》，323：533–536。[Seeger, 2008] Seeger，M.（2008）。稀疏线性模型的贝叶斯推断与最优设计。《机器学习研究杂志》，（9）：759–813。[Srivastava 等, 2014] Srivastava，N.、Hinton，G.、Sutskever，I. 和 Salakhtdinov，R.（2014）。随机失活：防止神经网络过拟合的一种简单方法。《机器学习研究杂志》，15：1929–1958。[Sutskever 等, 2014] Sutskever，I.、V.、O. 和 Quoc，V. L.（2014）。使用神经网络进行序列到序列学习。arXiv：1409.3215v*3* [cs.CL], 1：1–9。[Tipping, 2001] Tipping，M.（2001年）。稀疏贝叶斯学习与相关向量机。Micorsoft Research，1：211–244。[Wipf 和 Nagarajan, 2008] Wipf，D. 和 Nagarajan，S.（2008）。自动相关性确定的新视角。生物磁成像实验室，加州大学旧金山分校，第1–8页。