门控注意力条件熵正则的全流程解析

——从原始数据到条件熵惩罚的逐步讲解

2025 年 11 月 3 日

目录

1 导言：我们要解决什么问题？

在协同推理（split inference）场景中，客户端本地运行前端神经网络 $f_{\boldsymbol{\theta}}$，将输入图像 x 映射成中间表示 $z = f_{\boldsymbol{\theta}}(x)$；该表示（也称 smashed data）被发送到服务器继续推理。若攻击者截获 z，有可能通过模型反演（Model Inversion）或属性推断等方法恢复原始图像或敏感特征。为了降低风险，我们希望让相同类别的 smashed data “长得尽量像”，也就是让条件熵 $H(Z \mid Y)$ 尽可能小。这样一来，攻击者难以从 z 推断出具体的原始图像细节。

本报告聚焦于 `gated-att` 实现中门控注意力条件熵正则（Conditional Entropy Loss, CEL）的计算过程：从原始 mini-batch 数据开始，一直到得到具体的正则值 $\mathcal{L}_{\text{CEL}}$，并详细解释注意力与门控在其中扮演的角色。本文不再讨论其他衍生内容（例如训练建议、复现流程等），目的是确保读者掌握这一套机制的来龙去脉。

2 基本符号与数据结构

- $x \in \mathcal{X}$：客户端输入（如图像）；$y \in \{1, \dots, C\}$：对应标签。
- $f_{\boldsymbol{\theta}}$：客户端编码器，输出 smashed data $z = f_{\boldsymbol{\theta}}(x) \in \mathbb{R}^d$。
- mini-batch $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^{B}$，其 smashed data 集合 $Z_{\mathcal{B}} = \{z_i\}_{i=1}^{B}$。
- 对于标签 c，记 $Z_{\mathcal{B}}^{(c)} = \{z_i \mid y_i = c\}$，其大小为 m_c。
- 总损失 $\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{CEL}}$，其中 $\mathcal{L}_{\text{CE}}$ 是交叉熵，$\mathcal{L}_{\text{CEL}}$ 是本文关注的正则项。
- $\tau > 0$：方差阈值，$\varepsilon > 0$：防止取对数时为零的平滑项。
- 注意力隐层维度 h；权重矩阵 $W_V, W_U \in \mathbb{R}^{h \times d}$；向量 $\boldsymbol{w} \in \mathbb{R}^h$。

3 整体流程概览

对当前 mini-batch $\mathcal{B}$，门控注意力 CEL 的计算分为以下步骤：

骤 1，leftmargin 客户端编码器生成 smashed data：$z_i = f_{\boldsymbol{\theta}}(x_i)$。

骤 2，leftmargin（可选）对 z_i 进行规范化（如 LayerNorm），得到 $\tilde{z}_i$。

骤 3，leftmargin 通过门控注意力网络，为同一类别样本生成 softmax 注意力权重 α_i。

骤 4，leftmargin 根据 α_i 计算加权均值 $\bar{z}_c$ 和加权方差 v_c。

骤 5，leftmargin 将 v_c 代入门控函数，得到类别 c 的惩罚 $\mathcal{R}(c)$。

骤 6，leftmargin 聚合得出 $\mathcal{L}_{\text{CEL}} = \sum_c \beta_c \mathcal{R}(c)$。

核心困难在于第 3 步与第 4 步：注意力如何生成，门控如何调节，以及它们如何共同约束 smashed data 的散布。下面详细展开。

4 注意力与门控的作用机理

4.1 为什么需要注意力？

若我们对同一类别的 smashed data 简单地计算无权方差

$$v_c^{\text{plain}} = \frac{1}{m_c} \sum_{i=1}^{m_c} \left\| z_i - \frac{1}{m_c} \sum_{j=1}^{m_c} z_j \right\|_2^2,$$

这隐式假设所有样本同等可靠。但在实践中：

- mini-batch 中可能有噪声或异常值；
- 同类样本可能分布在多个子簇，单一均值无法概括全部模式；
- 我们希望自动识别“更典型”的样本，并让它们主导方差估计。

注意力机制正好提供一种可微的、数据驱动的权重分配方式：网络会学习谁更重要。

4.2 注意力模块结构

对任意 smashed data $z \in \mathbb{R}^d$，注意力网络（图 1）执行以下操作：

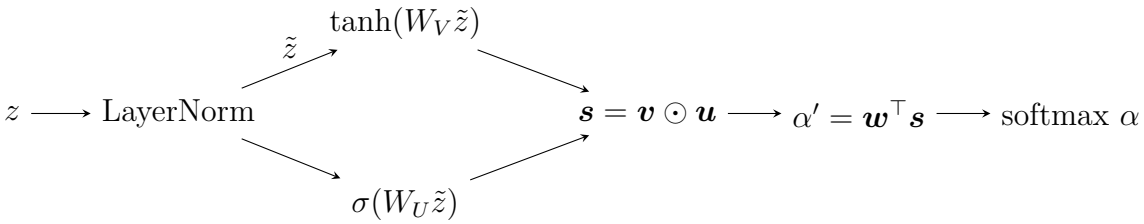

图 1: 门控注意力前向流程：双投影 + 元素乘法 + 线性汇聚 + softmax

第 1 步：规范化。 为减少不同样本尺度差异的影响，通常先对 z 做 LayerNorm：

$$\tilde{z} = \frac{z - \mu_z}{\sigma_z}, \quad \text{其中} \quad \mu_z = \frac{1}{d}\sum_{k=1}^{d} z_k,\ \sigma_z = \sqrt{\frac{1}{d}\sum_{k=1}^{d}(z_k - \mu_z)^2 + \epsilon_{\mathrm{LN}}}.$$

第 2 步：双投影。

$$\boldsymbol{v} = \tanh(W_V \tilde{z}), \qquad \boldsymbol{u} = \sigma(W_U \tilde{z}).$$

其中 W_V 学习如何在 tanh 的范围内提取关键信息；W_U 经由 Sigmoid 输出各维度的门控值（0 到 1）。

第 3 步：门控融合。

$$\boldsymbol{s} = \boldsymbol{v} \odot \boldsymbol{u},$$

逐元素乘法意味着：若某一维度在 $\boldsymbol{u}$ 中值较低，该维度的 $\boldsymbol{v}$ 被抑制；反之则保留。这样可以动态过滤掉不可靠或噪声特征。

第 4 步：线性汇聚与 softmax。

$$\alpha' = \boldsymbol{w}^\top \boldsymbol{s}, \qquad \alpha = \frac{\exp(\alpha')}{\sum_j \exp(\alpha'_j)}.$$

紧接着对所有同类样本的 α' 做 softmax，得到归一化注意力权重 α_i。此权重满足 $\sum_i \alpha_i = 1$，且可微。

4.3 为什么门控有效？

- **可解释性**：W_U 生成的 Sigmoid 输出可以理解为“是否让该特征通过”；如果模型判定某样本上的特定特征不可靠，就会通过门控削弱该特征在 $\boldsymbol{v}$ 中的贡献。
- **稳定性**：softmax 确保注意力权重总和为 1，避免了异常权重出现梯度爆炸。
- **可学习性**：$W_V, W_U, \boldsymbol{w}$ 均由数据驱动学习，使得注意力聚焦在能减小方差、保持分类性能的样本上。

5 从注意力到条件熵惩罚：数学推导

5.1 加权均值与加权方差

得到权重 $\{\alpha_i\}_{i=1}^{m_c}$ 后，定义加权均值

$$\bar{z}_c = \sum_{i=1}^{m_c} \alpha_i z_i.$$

加权方差（即类内散度）则为

$$v_c = \sum_{i=1}^{m_c} \alpha_i \|z_i - \bar{z}_c\|_2^2. \tag{1}$$

如果将 α_i 理解为样本重要性，v_c 就是重要性加权的均方偏差。

5.2 门控函数与阈值

为了避免 v_c 过小或过大带来数值问题，引入阈值门控：

$$\mathcal{R}(c) = \max\big(0, \log(v_c + \varepsilon) - \log(\tau + \varepsilon)\big).$$

- 若 $v_c \le \tau$，说明类内散度已经足够小，惩罚为 0；
- 若 $v_c > \tau$，惩罚随 $\log(v_c)$ 增大，鼓励网络进一步压缩 smashed data；
- ε 防止取对数时出现 $\log 0$。

最终，对整个 batch 的 CEL 为

$$\mathcal{L}_{\text{CEL}} = \sum_{c=1}^{C} \beta_c \mathcal{R}(c), \qquad \beta_c = \frac{m_c}{B}, \tag{2}$$

即以类别在 batch 中的占比作为权重。

5.3 梯度如何影响模型？

当某类别方差 v_c 超过阈值时，$\mathcal{R}(c)$ 的梯度开始作用于：

- 编码器参数 $\boldsymbol{\theta}$：推动 smashed data 收缩；
- 注意力参数 $W_V, W_U, \boldsymbol{w}$：调整注意力权重，让有代表性的样本更受重视。

举例说明，如果第 i 个样本与均值相差较大，则 $\|z_i - \bar{z}_c\|_2^2$ 较大，梯度会尝试：

- 减小此偏差（即推动编码器让 z_i 更接近 $\bar{z}_c$）；
- 或降低 α_i（如果模型认为该样本是一枚“坏样本”）。

这正体现了注意力 + 门控的配合：网络可自主决定是拉近样本还是降低其重要性。

6 完整计算示例

为了让本科生也能跟上，我们构建一个小型示例，展示每个数值是如何算出来的。

6.1 数据设定

考虑一个 mini-batch，有两个类别，每类三个 smashed data（二维向量）：

$$Z^{(1)} = \{(0.0, 0.0),\ (0.2, 0.1),\ (-0.1, 0.05)\},$$
$$Z^{(2)} = \{(1.0, 1.0),\ (0.9, 1.1),\ (1.2, 0.8)\}.$$

取注意力隐层维度 $h = 2$，设置

$$W_V = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad W_U = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}, \quad \boldsymbol{w} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

阈值 $\tau = 0.02$，平滑项 $\varepsilon = 10^{-6}$。

6.2 类别 1 的注意力计算

1. 规范化。 为简化说明，我们假设 LayerNorm 近似输出原值（即 $\tilde{z} \approx z$）。

2. 投影与门控。

$$\boldsymbol{v}_1 = \tanh((0.0, 0.0)) = (0, 0),$$
$$\boldsymbol{u}_1 = \sigma(0.5 \cdot (0.0, 0.0)) = (0.5, 0.5),$$
$$\boldsymbol{s}_1 = (0, 0) \odot (0.5, 0.5) = (0, 0),$$
$$\alpha_1' = \boldsymbol{w}^\top \boldsymbol{s}_1 = 0.$$

同理，

$$\begin{aligned}
\boldsymbol{v}_2 &= \tanh((0.2, 0.1)) \approx (0.197, 0.0997),\\
\boldsymbol{u}_2 &= \sigma(0.5 \cdot (0.2, 0.1)) \approx (0.55, 0.525),\\
\boldsymbol{s}_2 &\approx (0.108, 0.052),\\
\alpha_2' &= 0.160.\\
\boldsymbol{v}_3 &= \tanh((-0.1, 0.05)) \approx (-0.0997, 0.04996),\\
\boldsymbol{u}_3 &= \sigma(0.5 \cdot (-0.1, 0.05)) \approx (0.475, 0.5125),\\
\boldsymbol{s}_3 &\approx (-0.0473, 0.0256),\\
\alpha_3' &= -0.0217.
\end{aligned}$$

3. softmax 权重。

$$\begin{aligned}
\alpha_1 &= \frac{e^0}{e^0 + e^{0.160} + e^{-0.0217}} \approx 0.322,\\
\alpha_2 &\approx 0.381,\\
\alpha_3 &\approx 0.297.
\end{aligned}$$

4. 加权均值与方差。

$$\begin{aligned}
\bar{z}_1 &= 0.322(0,0) + 0.381(0.2,0.1) + 0.297(-0.1,0.05) \approx (0.046, 0.053).\\
v_1 &= 0.322\|(0,0) - (0.046,0.053)\|^2 + 0.381\|(0.2,0.1) - (0.046,0.053)\|^2\\
&\quad + 0.297\|(-0.1,0.05) - (0.046,0.053)\|^2\\
&\approx 0.322(0.0049 + 0.0028) + 0.381(0.0237 + 0.0022) + 0.297(0.0214 + 0.00001)\\
&\approx 0.0024 + 0.010 + 0.0064 \approx 0.0188.
\end{aligned}$$

5. 门控惩罚。 由于 $v_1 = 0.0188 < \tau = 0.02$，因此 $\mathcal{R}(1) = 0$，说明类别 1 的散度已经足够小。

6.3 类别 2 的计算（概述）

同理可得类别 2 的权重 α_i 大约分布在 $(0.34, 0.33, 0.33)$ 左右，加权均值 $\bar{z}_2 \approx (1.03, 0.96)$，加权方差 $v_2 \approx 0.0215$。因为 $v_2 > \tau$，有

$$\mathcal{R}(2) = \log\bigl(0.0215 + 10^{-6}\bigr) - \log\bigl(0.02 + 10^{-6}\bigr) \approx 0.0737.$$

若 mini-batch 中两个类别各占一半样本，则 $\beta_1 = \beta_2 = 0.5$，最终

$$\mathcal{L}_{\mathrm{CEL}} = 0.5 \times 0 + 0.5 \times 0.0737 = 0.0369.$$

这验证了：当某类散度超阈值时，CEL 立即产生惩罚，推动编码器与注意力进行调整。

7 注意力与门控如何协同降低条件熵

- **柔性权重**：注意力提供样本级可微权重，使模型自动识别“可信”样本；
- **门控过滤**：门控 $\sigma(W_U\tilde{z})$ 抑制噪声特征，确保 $\boldsymbol{v}$ 的重要维度不被干扰；
- **可逆性降低**：通过减小 v_c，我们约束了同类 smashed data 的方差，相当于减小了条件熵 $H(Z \mid Y = c)$，攻击者很难从 z 反推具体输入；
- **端到端训练**：整个过程嵌入主训练循环，梯度自动传回编码器与注意力模块，不需额外阶段。

8 关键超参数与直觉

虽然本报告不讨论训练细节，但为了帮助理解机制，列出影响注意力工作方式的几个重要超参数：

- 隐层维度 h：越大表示能力越强，但计算量也越大；
- 阈值 τ：控制惩罚触发点，过大导致正则过强，过小可能形同虚设；
- 平滑项 ε：通常取 10^{-6} 左右；
- 缩放系数（若有）γ：决定正则对总损失的影响力度。

这些参数在实际工程中需要根据分类性能与隐私需求综合平衡。这里只需理解它们在数学上的作用——调整注意力对样本选择的敏感度，以及 CEL 惩罚的强度。

9 总结

我们以流水线的形式展示了 `gated-att` 中门控注意力条件熵正则的完整计算过程：

1. smashed data 作为输入；
2. 通过 LayerNorm 规范化后，经双线性层生成 $\boldsymbol{v}$ 与 $\boldsymbol{u}$；
3. 利用门控得到 $\boldsymbol{s}$，再经线性层和 softmax 得到注意力权重；
4. 权重决定哪些样本在加权均值、加权方差中更重要；

5. 超过阈值的方差通过 log 门控转化为正则惩罚；

6. 正则项的梯度同时作用于编码器与注意力模块，以收紧 smashed data。

通过这种机制，我们达到了用端到端可学习的方式降低条件熵的目的，为协同推理系统提供了实用的隐私防护手段。

参考文献

参考文献

[1] Ilse, M., Tomczak, J. M., & Welling, M.
Attention-based Deep Multiple Instance Learning.
In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.