Sequence-discriminative training of DNNs 笔记

杨超
385526069@qq.com
placebokkk.github.io

December 25, 2019

记录《Sequence-discriminative training of deep neural networks》[1] 论文中 MMI 的推导。

1 神经网络输出和观测的似然值关系

NN 使用 softmax 作为输出

$$y_{ut}(s_{ut}) = P(s|\mathbf{o}_{ut}) = \frac{\exp\{a_{ut}(s)\}}{\sum_{s'} \exp\{a_{ut}(s')\}} \tag{1}$$

MMI 的目标是最大化完全似然和边缘似然的比值，神经网络输出的不是观察的似然，利用贝叶斯条件公式，可得到似然函数

$$\log p(\mathbf{o}_{ut}|s) = \log y_{ut}(s) + \log p(\mathbf{o}_{ut}) - \log P(s) \tag{2}$$

其中 $P(s)$ 是全局统计得到的状态 s 出现的概率，$p(\mathbf{o}_{ut})$ 是数据 $\mathbf{o}_{ut}$ 的本身分布，均和 $a_{ut}(s)$ 无关。

2 帧级别 Cross-Entroy 的梯度推导

帧级别 Cross-Entroy 的损失函数如下，目标是最小化该函数：

$$\mathcal{F}_{CE} = -\sum_{u=1}^{U}\sum_{t=1}^{T_u} \log y_{ut}(s_{ut}) \tag{3}$$

对 softmax 前的 activation 求导，利用该值可以继续反向传播计算任意 NN 的参数的梯

度:

$$
\begin{aligned}
\frac{\partial \mathcal{F}_{CE}}{\partial a_{ut}(s)} &= -\frac{\partial \log y_{ut}(s_{ut})}{\partial a_{ut}(s)} \\
&= -\frac{\partial \log \frac{\exp\{a_{ut}(s_{ut})\}}{\sum_{s'} \exp\{a_{ut}(s')\}}}{\partial a_{ut}(s)} \\
&= -\frac{\partial \left\{ \log \exp\{a_{ut}(s_{ut})\} - \log \sum_{s'} \exp\{a_{ut}(s')\} \right\}}{\partial a_{ut}(s)} \\
&= -\frac{\partial \log \exp\{a_{ut}(s_{ut})\}}{\partial a_{ut}(s)} + \frac{\partial \log \sum_{s'} \exp\{a_{ut}(s')\}}{\partial a_{ut}(s)} \\
&= -\frac{\delta_{s;s_{ut}} \cdot \exp\{a_{ut}(s_{ut})\}}{\exp\{a_{ut}(s_{ut})\}} + \frac{\exp\{a_{ut}(s\}}{\sum_{s'} \exp\{a_{ut}(s')\}} \\
&= -\delta_{s;s_{ut}} + y_{ut}(s)
\end{aligned}
\tag{4}
$$

3 序列级别 MMI 的梯度推导

注意原论文公式（5）的目标是使分子尽量等于分母，通过优化整个目标函数，使得正确标注文本 Wu 对应的似然概率尽量大，而其他文本序列 W 的似然概率尽量小。

3.1 原文中的公式 (6) 推导

根据原文公式 (5)

$$
\mathcal{F}_{MMI} = \sum_{u'} \left\{ \log p(\mathbf{O}_{u'}|S_{u'})^{\kappa} P(W_{u'}) - \log \sum_{W} p(\mathbf{O}_{u'}|S)^{\kappa} P(W) \right\} \tag{5}
$$

类似 CE，我们的目标也是要计算该式对 softmax 前的 activation 的导数，但是该导数不易直接求得，因此先对 $\log p(\mathbf{o}_{ut}|r)$ 求导，再利用链式求导法则求得目标。

对 $\log p(\mathbf{o}_{ut}|r)$ 求导时，只需考虑 $\mathcal{F}_{MMI}$ 求和中关于 u 的项即可，其他项和 $\log p(\mathbf{o}_{ut}|r)$ 无关，因此对 $\log p(\mathbf{o}_{ut}|r)$ 求导为 0，所以只需要计算下式的导数

$$
\log p(\mathbf{O}_u|S_u)^{\kappa} P(W_u) - \log \sum_{W} p(\mathbf{O}_u|S)^{\kappa} P(W) \tag{6}
$$

其中

$$
p(\mathbf{O}_u|S_u) = \prod_{t'} p(o_{ut'}|s_{ut'}) \tag{7}
$$

$\log p(\mathbf{O}_u|S_u)^{\kappa} P(W_u)$ 称为分子项，$\log \sum_W p(\mathbf{O}_u|S)^{\kappa} P(W)$ 称为分母项。下面分别对这两部分进行求导。

3.1.1 分子项

分子项 $\log p(\mathbf{O}_u|S_u)^{\kappa} P(W_u)$ 根据 (7) 展开

$$
\begin{aligned}
\log p(\mathbf{O}_u|S_u)^{\kappa} P(W_u) &= \kappa \cdot \sum_{t'} \log p(\mathbf{o}_{ut'}|s_{ut'}) + P(W) \\
&= \kappa \cdot \sum_{t' \neq t} \log p(\mathbf{o}_{ut'}|s_{ut'}) + \kappa \cdot \log p(\mathbf{o}_{ut}|s_{ut}) + P(W)
\end{aligned}
\tag{8}
$$

$t' \neq t$ 对应的项以及 P(W) 都和 $\log p(\mathbf{o}_{ut}|r)$ 无关，因此只需计算 $\kappa \cdot \log p(\mathbf{o}_{ut}|s_{ut})$ 对 $\log p(\mathbf{o}_{ut}|r)$ 的导数。若 $s_{ut} = r$, 则导数为 κ，否则导数为 0. 所以分子项的导数为

$$\kappa \cdot \delta_{r;s_{ut}} \tag{9}$$

3.1.2 分母项

分母项 $\log \sum_W p(\mathbf{O}_u|S)^\kappa P(W)$ 对 $\log p(\mathbf{o}_{ut}|r)$ 求导，得

$$\frac{1}{\sum_W p(\mathbf{O}_u|S)^\kappa P(W)} \cdot \frac{\partial \sum_W p(\mathbf{O}_u|S)^\kappa P(W)}{\partial \log p(\mathbf{o}_{ut}|r)} \tag{10}$$

因为 $\sum_W p(\mathbf{O}_u|S)^\kappa P(W)$ 是一个关于不同 W 求和式，分开考虑其中各项

- 若 W 对应的 S，其 s_t（t 时刻的所处状态）不等于 r，则该项与 $\log p(\mathbf{o}_{ut}|r)$ 无关，求导结果为 0。
- 若 W 对应的 S，其 s_t 等于 r，则该项与 $\log p(\mathbf{o}_{ut}|r)$ 有关。我们任选一个这种 W 来分析，假设其为 $\hat{W}$, 该项可写为

$$p(\mathbf{O}_u|S)^\kappa P(\hat{W}) = \prod_{t' \neq t} p(\mathbf{o}_{ut'}|s_{ut'})^\kappa \cdot p(\mathbf{o}_{ut}|s_{ut})^\kappa \cdot P(\hat{W}) \tag{11}$$

注意其中 $P(\hat{W})$ 以及 $t' \neq t$ 的项均和 $\log p(\mathbf{o}_{ut}|r)$ 无关。因为 $s_{ut} = r$, 所以 $p(\mathbf{o}_{ut}|s_{ut})^\kappa$ 和 $\log p(\mathbf{o}_{ut}|r)$ 有关，该式对 $\log p(\mathbf{o}_{ut}|r)$ 求导结果如下

$$\frac{\partial p(\mathbf{O}_u|S)^\kappa P(\hat{W})}{\partial \log p(\mathbf{o}_{ut}|r)} = \frac{\partial p(\mathbf{o}_{ut}|s_{ut} = r)^\kappa}{\partial \log p(\mathbf{o}_{ut}|r)} \cdot \prod_{t' \neq t} p(\mathbf{o}_{ut'}|s_{ut'})^\kappa \cdot P(\hat{W}) \tag{12}$$

令 $\mathbf{a} = \log p(\mathbf{o}_{ut}|s_{ut} = r)$, 则

$$\begin{aligned} \frac{\partial p(\mathbf{o}_{ut}|s_{ut} = r)^\kappa}{\partial \log p(\mathbf{o}_{ut}|r)} &= \frac{\partial (exp^{\mathbf{a}})^\kappa}{\partial \mathbf{a}} = \frac{\partial exp^{\mathbf{a}\kappa}}{\partial \mathbf{a}} \\ &= \kappa exp^{\mathbf{a}\kappa} = \kappa (exp^{\mathbf{a}})^\kappa \\ &= \kappa p(\mathbf{o}_{ut}|s_{ut} = r)^\kappa \end{aligned} \tag{13}$$

将 (13) 其带回 (12)

$$\frac{\partial p(\mathbf{O}_u|S)^\kappa P(\hat{W})}{\partial \log p(\mathbf{o}_{ut}|r)} = \kappa p(\mathbf{o}_{ut}|s_{ut} = r)^\kappa \cdot \prod_{t' \neq t} p(\mathbf{o}_{ut'}|s_{ut'})^\kappa \cdot P(\hat{W}) = \kappa \prod_{t'} p(\mathbf{o}_{ut'}|s_{ut'})^\kappa \cdot P(\hat{W}) \tag{14}$$

以上只是其中一个 $\hat{W}$ 的导数，(10) 中其他满足 s_t 等于 r 的 W 也要参与求导。将所有的这些 $\hat{W}$ 求和，(10) 变为

$$\frac{1}{\sum_W p(\mathbf{O}_u|S)^\kappa P(W)} \cdot \sum_{\hat{W}: s_t = r} \kappa \prod_{t'} p(\mathbf{o}_{ut'}|s_{ut'})^\kappa \cdot P(\hat{W}) \tag{15}$$

即

$$\frac{\kappa \sum_{\hat{W}: s_t = r} \prod_{t'} p(\mathbf{o}_{ut'}|s_{ut'})^\kappa \cdot P(\hat{W})}{\sum_W p(\mathbf{O}_u|S)^\kappa P(W)} \tag{16}$$

3.1.3 结论

根据公式 (6),(9), (16)

$$\begin{aligned}\frac{\partial \mathcal{F}_{MMI}}{\partial \log p(\mathbf{o}_{ut}|r)} &= \kappa \cdot \delta_{r;s_{ut}} - \frac{\kappa \sum_{\hat{W}:s_t=r} \prod_{t'} p(\mathbf{o}_{ut'}|s_{ut'})^{\kappa} \cdot P(\hat{W})}{\sum_W p(\mathbf{O}_u|S)^{\kappa} P(W)} \\ &= \kappa(\delta_{r;s_{ut}} - \gamma_{ut}^{DEN}(r))\end{aligned} \tag{17}$$

其中

$$\gamma_{ut}^{DEN}(r) = \frac{\sum_{\hat{W}:s_t=r} \prod_{t'} p(\mathbf{o}_{ut'}|s_{ut'})^{\kappa} \cdot P(\hat{W})}{\sum_W p(\mathbf{O}_u|S)^{\kappa} P(W)} \tag{18}$$

$\gamma_{ut}^{DEN}(r)$ 表示音频数据 $\mathbf{O}_u$ 在 t 时刻处在 r 状态的概率。该值可以在完整的解码图中用前向后项算法求的。为了减少计算量也可以在一个更小的 Lattice 上计算。该 Lattice 需要使用一个已有的识别系统生成，Lattice 中的路径除了 Wu，其他的路径正是模型容易混淆的路径。

3.2 原论文中的公式 (7) 推导

本文公式 (17)（原论文公式 (6)）得到的是 $\mathcal{F}_{MMI}$ 对于似然的导数，利用链式法则计算 $\mathcal{F}_{MMI}$ 对 softmax 前的 activation 值的导数

$$\frac{\partial \mathcal{F}_{MMI}}{\partial a_{ut}(s)} = \sum_r \frac{\partial \mathcal{F}_{MMI}}{\partial \log p(\mathbf{o}_{ut}|r)} \frac{\partial \log p(\mathbf{o}_{ut}|r)}{\partial a_{ut}(s)} \tag{19}$$

根据 (2)

$$\frac{\partial \log p(\mathbf{o}_{ut}|r)}{\partial a_{ut}(s)} = \frac{\partial \log y_{ut}(r)}{\partial a_{ut}(s)} \tag{20}$$

根据 (4)

$$\begin{aligned}\frac{\partial \log p(\mathbf{o}_{ut}|r)}{\partial a_{ut}(s)} &= \frac{\partial \log y_{ut}(r)}{\partial a_{ut}(s)} \\ &= \delta_{s;r} - y_{ut}(s)\end{aligned} \tag{21}$$

将 (17),(21) 带入 (19)

$$\begin{aligned}\frac{\partial \mathcal{F}_{MMI}}{\partial a_{ut}(s)} &= \sum_r \left\{\kappa(\delta_{r;s_{ut}} - \gamma_{ut}^{DEN}(r))(\delta_{s;r} - y_{ut}(s))\right\} \\ &= \sum_r \left\{\kappa(\delta_{r;s_{ut}} - \gamma_{ut}^{DEN}(r)) \cdot \delta_{s;r}\right\} - \sum_r \left\{\kappa(\delta_{r;s_{ut}} - \gamma_{ut}^{DEN}(r)) \cdot y_{ut}(s)\right\} \\ &= \kappa(\delta_{s;s_{ut}} - \gamma_{ut}^{DEN}(s)) - \kappa\left\{\sum_r \delta_{r;s_{ut}} - \sum_r \gamma_{ut}^{DEN}(r)\right\} y_{ut}(s) \\ &= \kappa(\delta_{s;s_{ut}} - \gamma_{ut}^{DEN}(s)) - \kappa\left\{1 - 1\right\} y_{ut}(s) \\ &= \kappa(\delta_{s;s_{ut}} - \gamma_{ut}^{DEN}(s))\end{aligned} \tag{22}$$

4 解释

- $y_{ut}(s)$: 给定神经网络声学模型，$\mathbf{o}_{ut}$ 的输出状态是 s 的概率

- $\gamma_{ut}^{DEN}(s)$：给定神经网络声学模型和解码网络，$\mathbf{O}$, 其 t 时刻输出状态是 s 的概率，因此 $\gamma_{ut}^{DEN}(s)$ 不仅依赖于 t 时刻的 $\mathbf{o}_{ut}$，还依赖其他时刻的 $\mathbf{o}$，并且依赖解码网络（语言学模型等信息）.
- $\gamma_{ut}^{DEN}(s)$ 可以看作是 $y_{ut}(s)$ 的一种拓展。

为什么 (4) 和 (22) 的最终形式 (原论文中 (4) 和 (6)) 之间相差了一个负号?

这是因为公式 (3) 需要**最小化** Cross Entropy 的目标损失函数，而公式 (5) 需要**最大化** MMI 的目标函数，差了一个负号。

不仅是帧级别的 CE 和 MMI，其他目标函数（如 CTC，CRF）用梯度法计算得到的公式，也都能找到比较明确的物理意义，即模型预测的值和标注的值之差。

References

[1] Karel Veselý1, Arnab Ghoshal, Lukáš Burget, Daniel Povey `Sequence discriminative training of deep neural networks`. Interspeech. Vol. 2013. 2013