序列生成模型

学习问题和 Seq2Seq 模型

肖睿

College of Mathematics
Sichuan University

March 12

本章主要介绍在序列数据上经常使用的一种模型: 自回归生成模型 (AutoRegressive Generative Model). 给定一个包含 N 个序列的数据集 $\mathcal{D} = \{\boldsymbol{x}_{1:T_n}^{(n)}\}_{n=1}^{N}$, 模型学习 $p_\theta(x|\boldsymbol{x}_{1:(t-1)})$ 来最大化整个数据集的对数似然函数

$$\max_\theta \sum_{n=1}^{N} \log p_\theta(\boldsymbol{x}_{1:T_n}^{(n)}) = \max_\theta \sum_{n=1}^{N} \sum_{t=1}^{T_n} \log p_\theta(x_t^{(n)}|\boldsymbol{x}_{1:(t-1)}^{(n)}).$$

曝光偏差（Exposure Bisa）问题：生成序列的过程中存在错误会导致错误传播，后续生成的序列也会偏离真实分布

计划采样：控制替换率 ϵ 作为输入数据中真实数据的比例，在训练前期使用较大的 ϵ，之后逐步减少

减少的方式：令 ε_i 为在第 i 次迭代时的替换率，
(1) 线性衰减：$\varepsilon_i = \max(\varepsilon, k - c_i)$，其中 ε 为最小的替换率，k 和 c 分别为初始值和衰减率.
(2) 指数衰减：$\varepsilon_i = k^i$，其中 $k < 1$ 为初始替换率.
(3) 逆 Sigmoid 衰减：$\varepsilon_i = k/(k + \exp(i/k))$，其中 $k \geq 1$ 来控制衰减速度.

使用最大似然估计来优化模型，这导致训练目标和评价方法不一致. 并且这些评价指标一般都是不可微的，无法直接使用基于梯度的方法来进行优化. 对此我们可以将自回归序列生成看作一种马尔可夫决策过程，并使用强化学习的方法来进行训练.

在第 t 步，动作 a_t 可以看作从词表中选择一个词，策略为 $\pi_\theta(a|s_t)$，其中状态 s_t 为之前步骤中生成的前缀序列 $x_{1:(t-1)}$. 我们可以把一个序列 $x_{1:T}$ 看作马尔可夫决策过程的一个轨迹 (trajectory):

$$\tau = \{s_1, a_1, s_2, a_2, ..., s_T, a_T\}.$$

轨迹 τ 的概率为

$$p_\theta(\tau) = \prod_{t=1}^{T} \pi_\theta(a_t = x_t | s_t = x_{1:(t-1)}),$$

其中状态转移概率 $p(s_t = x_{1:t-1} | s_{t-1} = x_{1:(t-2)}, a_{t-1} = x_{t-1}) = 1$ 是确定性的，可以被忽略.

强化学习的目标是学习一个策略 $\pi_\theta(a|s_t)$，使得期望回报最大，其中

$$\mathcal{J}(\theta) = \mathbb{E}_{\tau \sim p_\theta(\tau)}[G(\tau)] \tag{1}$$

$$= \mathbb{E}_{\mathbf{x}_{1:T} \sim \mathbf{p}_\theta(\mathbf{x}_{1:T})}[G(\mathbf{x}_{1:T})], \tag{2}$$

其中 $G(\boldsymbol{x}_{1:T})$ 表示序列 $x_{1:T}$ 的总回报，可以是 BLEU、ROUGE 或其他评价指标。

在第 t 步时，前缀序列为 $\tilde{h}_t = x_{1:(t-1)}$，词 x_t 的条件概率为

$$p_\theta(x_t|\tilde{h}_t) = \text{softmax}\Big(s(x_t, \tilde{h}_t; \theta)\Big)$$

其中 $s(x_t, \tilde{h}_t; \theta)$ 为未经过 Softmax 归一化的得分函数，$z(\tilde{h}_t; \theta)$ 为配分函数（ Partition Function ）：

$$Z(\tilde{h}_t; \theta) = \sum_{v \in \mathcal{V}} \exp\big(s(v, \tilde{h}_t; \theta)\big).$$

介绍三种加速的方法：层次化 softmax、重要性采样、噪声对比估计

层次化 Softmax

将词表中的词分成 κ 组，并且每一个词只能属于一个分组，每组大小为 $\frac{|\nu|}{K}$。假设词 w 所属的组为 $c(w)$，则

$$p(w|\tilde{h}) = p(w, c(w)|\tilde{h}) \tag{3}$$

$$= p(w|c(w), \tilde{h})p(c(w)|\tilde{h}), \tag{4}$$

因此，一个词的概率可以分解为两个概率的乘积，它们可以分别利用神经网络来估计，这样计算 Softmax 函数时分别只需要做 $\frac{|\nu|}{K}$ 和 K 次求和，从而大大提高了计算速度。
为了进一步降低 Softmax 函数的计算复杂度，我们可以使用更深层的树结构来组织词汇表：二叉树和霍夫曼编码。

二叉树

假设词 v 在二叉树上从根节点到其所在叶子节点的路径长度为 M，其编码可以表示为一个位向量 (bit vector)：$[b_1, \cdots, b_M]^{\mathrm{r}}$。词 v 的条件概率为

$$P(v|\tilde{h}) = p(b_1, \cdots, b_M|\tilde{h}) \tag{5}$$

$$= \prod_{m=1}^{M} p(b_m|b_1, \cdots, b_{m-1}, \tilde{h}) \tag{6}$$

$$= \prod_{m=1}^{M} p(b_m|b_{m-1}, \tilde{h}). \tag{7}$$

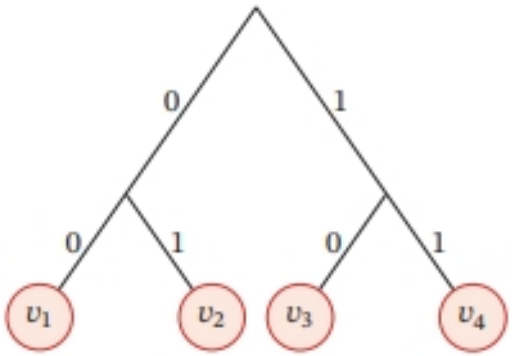
由于 $b_m \in \{0, 1\}$ 为二值变量，我们可以将 $p(b_m|b_{m-1}, \tilde{h})$ 视为二分类问题，使用 Logistic 回归来进行预测。

二叉树

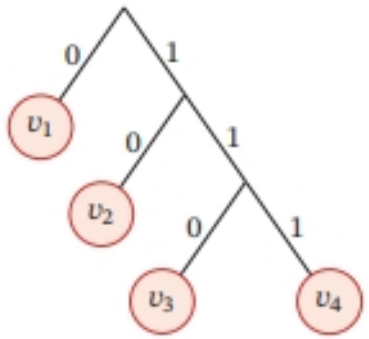
$$p(b_m = 1 | b_{m-1}, \hat{h}) = \sigma(\boldsymbol{w}^{\tau}_{n(b_{m-1})} \boldsymbol{h} + \boldsymbol{b}_{n(b_{m-1})}),$$

其中 $n(b_{m-1})$ 为词 υ 在树 T 上的路径上的第 $m-1$ 个节点，若使用平衡二叉树来进行分组，则条件概率估计可以转换为 $\log_2 |\nu|$ 个二分类问题. 这时原始预测模型中的 Softmax 函数可以用 Logistic 函数代替，计算效率可以加速 $\frac{|V|}{\log_2 |V|}$ 倍.

霍夫曼编码



(a) 平衡树



(b) 霍夫曼编码树

重要性采样

用随机梯度上升来更新参数 θ 时，第 t 个样本 (h_t, x_t) 的目标函数关于 θ 的梯度为

$$\frac{\partial \log p_\theta(x_t|\tilde{h}_t)}{\partial \theta} = \frac{\partial s(x_t, \tilde{h}_t; \theta)}{\partial \theta} - \frac{\partial \log(\sum_\upsilon \exp(s(\upsilon, \tilde{h}_t; \theta)))}{\partial \theta} \quad (8)$$

$$= \frac{\partial s(x_t, \bar{h}_t; \theta)}{\partial \theta} - \mathbb{E}_{p_\theta(\upsilon|\tilde{h}_t)}\left[\frac{\partial s(\upsilon, \tilde{h}_t; \theta)}{\partial \theta}\right]. \quad (9)$$

提议分布

重要性采样是用一个容易采样的提议分布 q 来近似估计分布 p。

$$\mathbb{E}_{p_{\hat{\theta}}(v|\tilde{h}_t)}\left[\frac{\partial s(v,\tilde{h}_t;\theta)}{\partial\theta}\right] = \sum_{v\in\mathcal{V}} p_\theta(v|\tilde{h}_t)\frac{\partial s(v,\tilde{h}_t;\theta)}{\partial\theta} \tag{10}$$

$$= \sum_{v\in V} q(v|\tilde{h}_t)\frac{p_\theta(v|\tilde{h}_t)}{q(v|\tilde{h}_t)}\frac{\partial s(v,\tilde{h}_t;\theta)}{\partial\theta} \tag{11}$$

$$= \mathbb{E}_{q(v|\hat{h}_t)}\left[\frac{p_\theta(v|\hat{h}_t)}{q(v|\hat{h}_t)}\frac{\partial s(v,\hat{h}_t;\theta)}{\partial\theta}\right] \tag{12}$$

$$\approx \frac{1}{K}\sum_{k=1}^{K}\frac{p_\theta(v_k|\hat{h}_t)}{q(v_k|\bar{h}_t)}\frac{\partial s(v_k,\hat{h}_t;\theta)}{\partial\theta}. \tag{13}$$

提议分布

同样适用重要性采样来计算分配函数，使用相同的提议分布函数，并复用上一步中抽取的样本：

$$Z(\tilde{h}_t) = \sum_w \exp\Big(s(w, \tilde{h}_t; \theta)\Big) \tag{14}$$

$$= \sum_w q(w|\tilde{h}_t) \frac{1}{q(w|\tilde{h}_t)} \exp\Big(s(w, \tilde{h}_t; \theta)\Big) \tag{15}$$

$$= \mathbb{E}_{q(w|\tilde{h}_t)} \left[\frac{1}{q(w|\tilde{h}_t)} \exp\Big(s(w, \tilde{h}_t; \theta)\Big)\right] \tag{16}$$

$$\approx \frac{1}{K} \sum_{k=1}^{K} \frac{1}{q(v_k|\hat{h}_t)} \exp\Big(s(v_k, \tilde{h}_t; \theta)\Big), \tag{17}$$

其中 $r(v_k) = \frac{\exp\left(s(v_k, \hat{h}_i; \theta)\right)}{q(v_k|\bar{h}_i)}$。

提议分布

每个样本目标函数关于 θ 的梯度可以近似为

$$\frac{\partial \log p_\theta(x_t|\tilde{h}_t)}{\partial \theta} = \frac{\partial s(x_t, \tilde{h}_t; \theta)}{\partial \theta} - \frac{1}{\sum_{k=1}^{K} r(v_k)} \sum_{k=1}^{K} r(v_k) \frac{\partial s(v_k, \tilde{h}_t; \theta)}{\partial \theta},$$

噪声对比估计

将密度估计问题转化成二分类问题，降低计算复杂度。
噪声对比估计是通过调整模型 $p_\theta(x)$ 使得判别函数 D 很容易分辨出样本 x 来自哪个分布. 令 $y \in \{1, 0\}$ 表示一个样本 x 是真实样本或噪声样本，其条件概率为

$$p(x|y = 1) = p_\theta(x),$$
$$p(x|y = 0) = q(x).$$

一般噪声样本的数量要比真实样本大很多. 为了提高近似效率，我们近似假设噪声样本的数量是具实样本的 K 倍，即 y 的先验分布满足

$$p(y = 0) = Kp(y = 1).$$

噪声对比估计

从真实分布 $p_r(x)$ 中抽取 N 个样本 $x_1, \cdots, x_N$，将其类别设为 $y = 1$。然后从噪声分布中抽取 KN 个样本 $x'_1, \cdots, x'_{\mathrm{KN}}$，将其类别设为 $y = 0$。噪声对比估计的序列生成模型中的学习问题目标是将真实样本和噪声样本区别开来，可以看作一个二分类问题。噪声对比估计的损失函数为

$$\mathcal{L}(\theta) = -\frac{1}{N(K+1)}\left(\sum_{n=1}^{N}\log p(y=1|x_n) + \sum_{n=1}^{KN}\log p(y=0|x'_n)\right). \tag{18}$$

生成对抗模型

价值函数：

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{dan}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))]$$

其中，$E_{x\sim p_{\text{data}}(x)}[\log D(x)]$ 表示当 x 来自 p_{data} 时，$D(x)$ 越接近 1，则鉴别效果越好，$\log D(x)$ 越接近 0。而 $E_{z\sim p_z(z)}[\log(1-D(G(z))]$ 表示鉴别生成器生成的数据是否为真实数据。$D(G(z))$ 越接近 0，效果越好，$\log(1-D(G(z))$ 越接近 0。

基于噪声对比估计的序列模型

给定历史信息 $\tilde{h}$，我们需要判断词表中每一个词 v 是来自于真实分布还是噪声分布，令

$$p(y=1|v,\tilde{h}) = \frac{p_\theta(v|\tilde{h})}{p_\theta(v|\tilde{h}) + Kq(v)}. \tag{19}$$

对于一个训练序列 $x_{1:T}$，将 $\{(\tilde{h}_t, x_t)\}_{t=1}^T$ 作为真实样本。对于每一个 x_t，从噪声分布中抽取 K 个噪声样本 $\{x'_{t,1}, \cdots, x'_{t,k}\}$。噪声对比估计的目标函数是

$$\mathcal{L}(\theta) = -\sum_{t=1}^{T}\left(\log p(y=1|x_t,\hat{h}_t) + \sum_{k=1}^{K}\log\left(1-p(y=1|x'_{t,k},\hat{h}_t)\right)\right). \tag{20}$$

基于噪声对比估计的序列模型

将 $|\nu|$ 类的分类问题转换为了一个二分类问题。我们将负对数配分函数 $-\log Z(\tilde{h};\theta)$ 作为一个可学习的参数 $z_{\tilde{h}}$（即每一个 $\tilde{h}$ 对应一个参数）。这样，条件概率 $p_\Theta(\upsilon|\tilde{h})$ 重新定义为

$$p_\theta(\upsilon|\tilde{h}) = \exp\Big(s(\upsilon,\tilde{h};\theta)\Big)\exp(z_{\tilde{h}}). \tag{21}$$

基于噪声对比估计的序列模型

噪声对比估计方法的一个特点是会促使未归一化分布 $\exp\left(s(\upsilon, \tilde{h}{:}\theta)\right)$ 学习到一个近似归一化的分布，并接近真实的数据分布 $p_r(\upsilon|\hat{h})$：

$$p(y=1|v,\tilde{h}) = \frac{\exp\left(s(\upsilon,\bar{h};\theta)\right)}{\exp\left(s(\upsilon,\tilde{h};\theta)\right)+Kq(\upsilon)} = \sigma(\Delta s(v,\tilde{h};\theta)), \quad (22)$$

其中 $\Delta s(\upsilon,\tilde{h};\theta) = s(\upsilon,\tilde{h};\theta) - log(Kq(\upsilon))$。

小结

基于采样的方法并不改变模型的结构，只是近似计算参数梯度. 在训练时可以显著提高模型的训练速度，但是在测试阶段依然需要计算配分函数。而基于层次化 Softmax 的方法改变了模型的结构，在训练和测试时都可以加快计算速度.

序列到序列 (Sequence-to-Sequence, Seq2Seq): 给定一个序列 $x_{1:S}$, 生成另一个序列 $y_{1:T}$.
序列到序列模型的目标是估计条件概率

$$p_\theta(\mathbf{y}_{1:T}|\mathbf{x}_{1:S}) = \prod_{t=1}^{T} p_\theta(y_t|\mathbf{y}_{1:(t-1)}, \mathbf{x}_{1:S}),$$

其中 $y_t \in \nu$ 为词表 ν 中的某个词. 给定一组训练数据 $\{(x_{S_n}, y_{T_n})\}_{n=1}^{N}$, 我们可以使用最大似然估计来训练模型参数

$$\hat{\theta} = \arg\max \sum_{n=1}^{N} \log p_\theta(\mathbf{y}_{1:T_n}|\mathbf{x}_{1:S_n}).$$

一旦训练完成, 模型就可以根据一个输入序列 x 来生成最可能的目标序列.

$$\hat{\mathbf{y}} = \arg\max_{y} p_{\hat{\theta}}(\mathbf{y}|\mathbf{x}),$$

实现序列到序列的最直接方法是使用两个循环神经网络来分别进行编码和解码，也称为编码器-解码器（Encoder-Decoder）模型。
编码器：首先使用一个循环神经网络 f_{enc} 来编码输入序列 $x_{1:S}$ 得到一个固定维度的向量 u，通常为编码循环神经网络最后时刻的隐状态：

$$\boldsymbol{h}_t^{\text{enc}} = f_{\text{enc}}(\boldsymbol{h}_{t-1}^{\text{enc}}, \boldsymbol{e}_{x_{t-1}}, \theta_{\text{enc}}), \quad \forall t \in [1:S], \tag{23}$$

$$\boldsymbol{u} = \boldsymbol{h}_S^{\text{enc}}, \tag{24}$$

解码器：在生成目标序列时，使用另外一个循环神经网络 f_{dec} 来进行解码。在解码过程的第 t 步时，已生成前缀序列为 $y_{1:(t-1)}$。令 h_t^{dec} 表示在网络 f_{dec} 的隐状态，$\boldsymbol{o}_t \in (0,1)^{|\nu|}$ 为词表中所有词的后验概率，则：

$$\boldsymbol{h}_0^{\text{dec}} = \boldsymbol{u}, \tag{25}$$

$$\boldsymbol{h}_t^{\text{dec}} = f_{\text{dec}}(\boldsymbol{h}_{t-1}^{\text{dec}}, \boldsymbol{e}_{y_{t-1}}, \theta_{\text{dec}}), \tag{26}$$

$$\mathbf{0}_t = g(\boldsymbol{h}_t^{\text{dec}}, \theta_o). \tag{27}$$

在解码过程的第 t 步时，我们首先使用上一步的隐状态 $\boldsymbol{h}_{t-1}^{\text{dec}}$ 作为查询向量，利用注意力机制从所有输入序列的隐状态 $\boldsymbol{H}^{\text{enc}} = [\boldsymbol{h}_1^{\text{enc}}, \cdots, \boldsymbol{h}_S^{\text{enc}}]$ 中选择相关信息：

$$\boldsymbol{c}_t = \text{att}(\boldsymbol{H}^{\text{enc}}, h_{t-1}^{\text{dec}}) \tag{28}$$

$$= \sum_{i=1}^{S} \alpha_i \boldsymbol{h}_i^{\text{enc}} \tag{29}$$

$$= \sum_{i=1}^{S} \text{softmax}\left(s(h_i^{\text{enc}}, h_{t-1}^{\text{dec}})\right) h_i^{\text{enc}} \tag{30}$$

然后，将从输入序列中选择的信息 c_t 也作为解码器 $f_{\text{dec}}(\cdot)$ 在第 t 步时的输入，得到第 t 步的隐状态：

$$\boldsymbol{h}_t^{\text{dec}} = f_{\text{dec}}(\boldsymbol{h}_{t-1}^{\text{dec}}, [\boldsymbol{e}_{y_{t-1}}; \boldsymbol{c}_t], \theta_{\text{dec}}) \tag{31}$$

最后，将 h_t^{dec} 输入到分类器 $g(\cdot)$ 中来预测词表中每个词出现的概率。

建立全连接的网络结构，提高并行计算效率及长程依赖问题。

$$\mathrm{MultiHead}(H) = \mathbf{W}_o[\mathrm{head}_1; \cdots ; \mathrm{head}_M],$$

$$\mathrm{head}_m = \mathrm{self} - \mathrm{att}(\mathbf{Q}_m, \mathbf{K}_m, \mathbf{V}_m) = \mathbf{V}_m \mathrm{softmax}\Big(\frac{\mathbf{K}_m^{\mathsf{T}}\mathbf{Q}_m}{\sqrt{D_k}}\Big)$$

$$\forall m \in \{1, \cdots, M\}, \quad \boldsymbol{Q}_m = \boldsymbol{W}_q^m \boldsymbol{H}, \boldsymbol{K} = \boldsymbol{W}_k^m \boldsymbol{H}, V = \boldsymbol{W}_\upsilon^m \boldsymbol{H},$$

其中 $W_o \in \mathbb{R}^{D_h \times M d_v}$ 为输出投影矩阵，
$W_q^m \in \mathbb{R}^{D_k \times D_h}, W_k^m \in \mathbb{R}^{D_k \times D_h}, W_v^m \in \mathbb{R}^{D_\upsilon \times D_h}$ 为投影矩阵，
$m \in \{1, \cdots, M\}$.

Transformer 模型

(1) 编码器只包含多层的多头自注意力 (Multi-Head Self-Attention) 模块，每一层都接受前一层的输出作为输入. 编码器的输入为序列 $x_{1:S}$，输出为一个向量序列 $H^{\text{enc}} = [h_1^{\text{enc}}, \cdots, h_S^{\text{enc}}]$. 然后，用两个矩阵将 H^{enc} 映射到 K^{enc} 和 V^{enc} 作为键值对供解码器使用，即

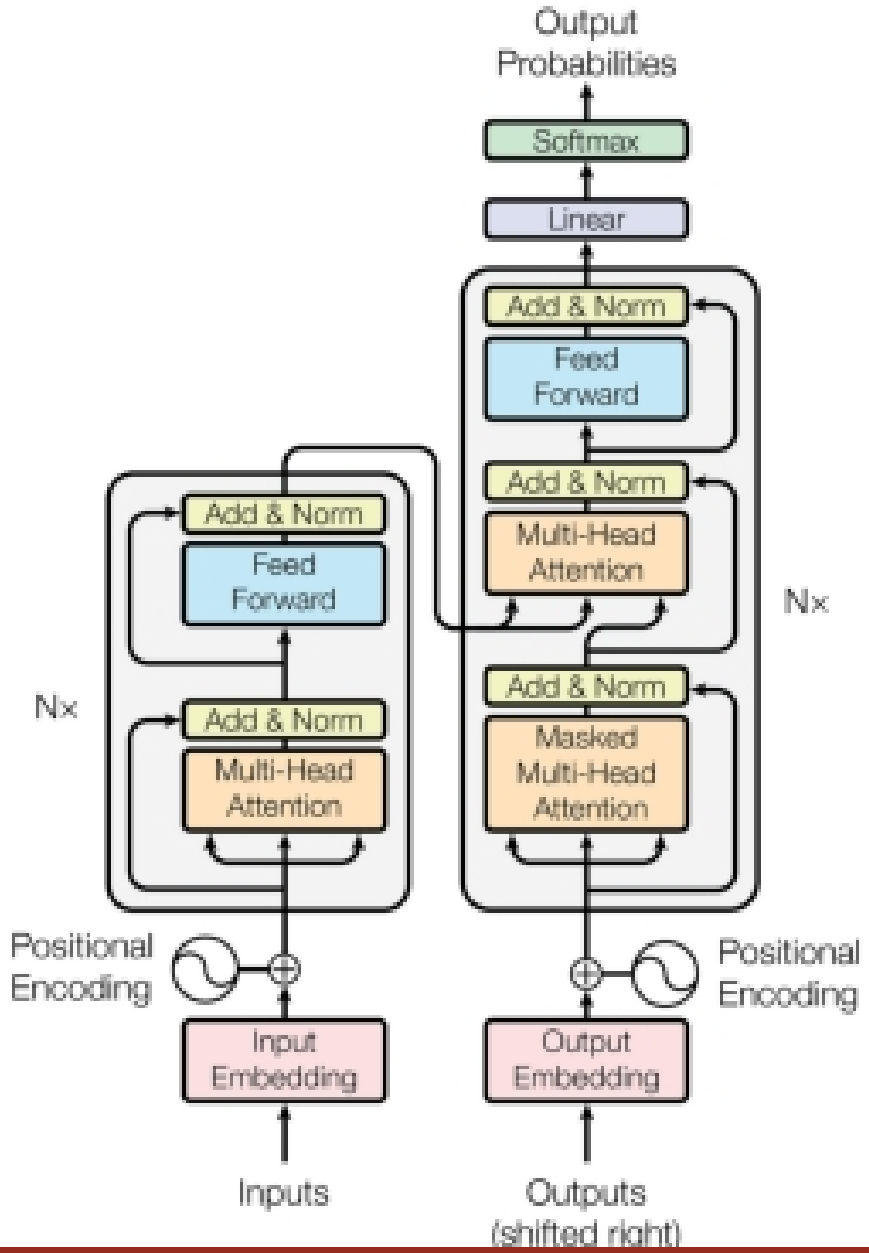
$$K^{\text{enc}} = W_k' H^{\text{enc}},$$
$$V^{\text{enc}} = W_v' H^{\text{enc}},$$

Transformer 模型

a) 掩蔽自注意力模块：第 t 步时，先使用自注意力模型对已生成的前缀序列 $y_{0:(t-1)}$ 进行编码得到 $H^{\text{dec}} = [h_1^{\text{dec}}, \cdots, h_t^{\text{dec}}]$.
b) 解码器到编码器注意力模块：将 h_t^{dec} 进行线性映射得到 q_t^{dec}. 将 q_t^{dec} 的信息.
c) 逐位置的前馈神经网络：使用一个前馈神经网络来综合得到所有信息:

$$\text{FFN}(\mathbf{z}) = \mathbf{W}_2 \text{ReLu}(\mathbf{W}_1 \mathbf{z} + \mathbf{b}_1) + \mathbf{b}_2,$$

其中 W_1, W_2, b_1, b_2 为网络参数.


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Positional Encoding
Positional Encoding
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)

- 为了解决曝光偏差问题，我们在训练时混合使用真实数据和模型生成的数据，控制使用两种数据的比例，将序列生成看作强化学习问题，并使用最大似然估计来预训练模型，并逐步将训练目标由最大似然估计切换为最大期望回报. 进一步，还可以利用生成对抗网络的思想来进行文本生成.
- 由于深度序列模型在输出层使用 softmax 进行归一化，为了提高效率，利用重要性采样来加速 softmax 的计算，或者噪声对比估计来计算非归一化的条件概率.
- 常见的序列到序列生成模型有：基于循环神经网络的序列到序列模型来进行机器翻译，使用注意力模型来改进循环神经网络的长程依赖问题，基于卷积神经网络的序列到序列模型. 目前最成功的序列到序列模型是全连接的自注意力模型，比如 Transformer.