第 十六 章 注意力机制与多智能体强化学习

注意力机制 (Attention) 是一种重要的深度学习方法，它最主要的用途是自然语言处理，比如机器翻译、情感分析。本章的目的不是详细解释注意力机制的原理，而是它在多智能体强化学习 (MARL) 中的应用。第 16.1 简单介绍自注意力机制 (Self-Attention)，它是一种特殊的注意力机制。第 16.2 将自注意力机制应用在 MARL，改进中心化训练或中心化决策。当智能体数量 m 较大时，自注意力机制对 MARL 有明显的效果提升。

16.1 自注意力机制

注意力机制 (Attention) 最初用于改进循环神经网络 (RNN)，提高 Sequence-to-Sequence (Seq2Seq) 模型的表现。自注意力机制 (Self-Attention) 是注意力机制的一种扩展，不局限于 Seq2Seq 模型，可以用于任意的 RNN。后来 Transformer 模型将 RNN 剥离，只保留注意力机制。与 RNN + 注意力机制相比，只用注意力机制居然用更好的表现，在机器翻译等任务上的效果有大幅提升。本节不深入讨论注意力机制与 RNN、Seq2Seq 之间的关系，而只介绍本章所需的一些知识点。

考虑这样一个问题：输入是长度为 m 的序列 $(\boldsymbol{x}^1, \cdots, \boldsymbol{x}^m)$，序列中的元素都是向量，要求输出长度同样为 m 的序列 $(\boldsymbol{c}^1, \cdots, \boldsymbol{c}^m)$；如图 16.1 所示。问题还有两个要求：

- 第一，序列的长度 m 是不确定的，可以变化。
- 第二，输出的向量 $\boldsymbol{c}^i$ 不是仅仅依赖于向量 $\boldsymbol{x}^i$，而是依赖于所有的输入向量 $(\boldsymbol{x}^1, \cdots, \boldsymbol{x}^m)$。

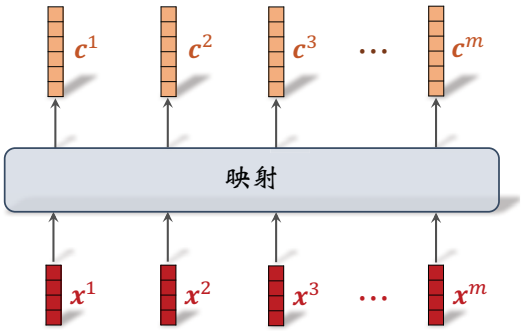

图 16.1: 将一个长度为 m 的向量序列映射到另一个同等长度的向量序列。

可以用简单的全连接网络逐个把向量 $\boldsymbol{x}^i$ 映射到 $\boldsymbol{c}^i$，但是这样得到的 $\boldsymbol{c}^i$ 仅依赖于 $\boldsymbol{x}^i$ 一个向量而已，不满足第二个要求。第 12 章介绍的 RNN 也不满足第二个要求；RNN 输出的向量 $\boldsymbol{c}^i$ 只依赖于 $(\boldsymbol{x}^1, \cdots, \boldsymbol{x}^i)$，而不依赖于 $(\boldsymbol{x}^{i+1}, \cdots, \boldsymbol{x}^m)$。

自注意力层 (Self-Attention Layer) 可以解决上述问题。如图 16.2 所示，自注意力层的输入是序列 $(\boldsymbol{x}^1, \cdots, \boldsymbol{x}^m)$，其中的向量的大小都是 $d_{\text{in}} \times 1$。自注意力层有三个参数矩阵：

$$\boldsymbol{W}_q \in \mathbb{R}^{d_q \times d_{\text{in}}}, \qquad \boldsymbol{W}_k \in \mathbb{R}^{d_q \times d_{\text{in}}}, \qquad \boldsymbol{W}_v \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}.$$

序列长度 m 不会影响参数的数量。不论序列有多长，参数矩阵只有 $\boldsymbol{W}_q, \boldsymbol{W}_k, \boldsymbol{W}_v$。这三个参数矩阵需要从训练数据中学习。自注意力层通过以下步骤，把输入序列 $(\boldsymbol{x}^1, \cdots, \boldsymbol{x}^m)$ 映射到输出序列 $(\boldsymbol{c}^1, \cdots, \boldsymbol{c}^m)$，输出向量的大小都是 $d_{\text{out}} \times 1$。

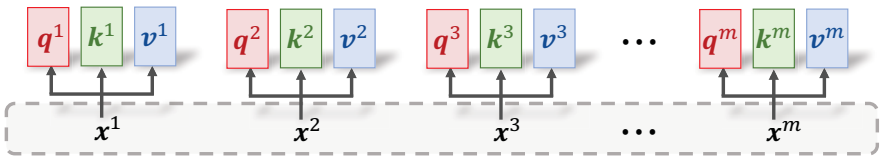

图 16.2: 首先把 $\boldsymbol{x}^i$ 映射到三元组 $(\boldsymbol{q}^i, \boldsymbol{k}^i, \boldsymbol{v}^i)$, $\forall i = 1, \cdots, m$。

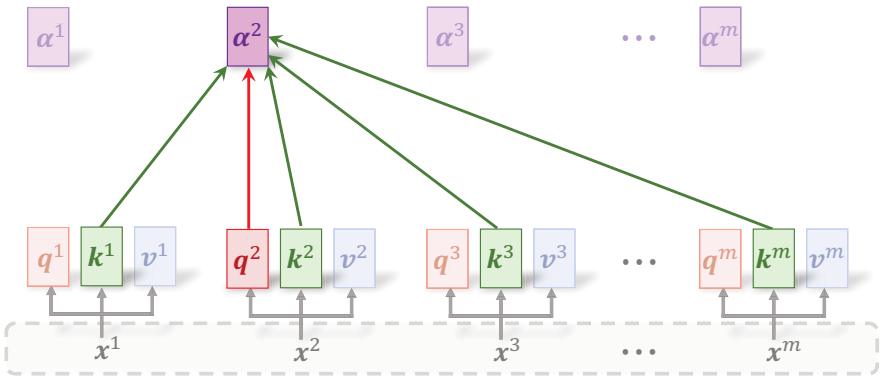

图 16.3: 然后用 $\boldsymbol{q}^i$ 和 $(\boldsymbol{k}^1, \cdots, \boldsymbol{k}^m)$ 计算权重向量 $\boldsymbol{\alpha}^i \in \mathbb{R}^m$, $\forall i = 1, \cdots, m$。

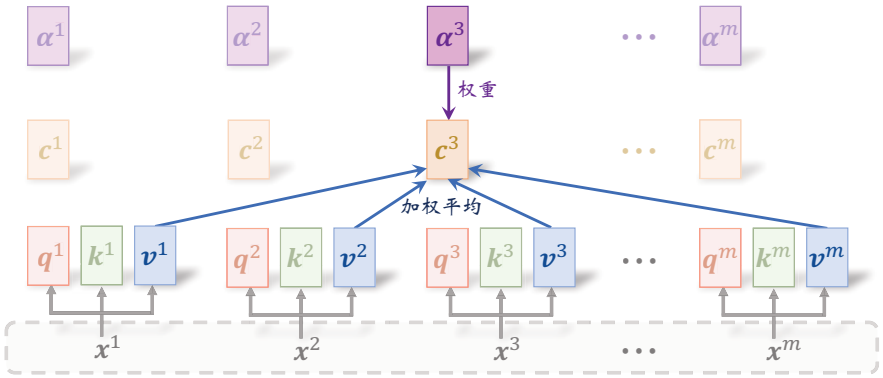

图 16.4: 最后用 $\boldsymbol{\alpha}^i$ 和 $(\boldsymbol{v}^1, \cdots, \boldsymbol{v}^m)$ 计算输出向量 $\boldsymbol{c}^i \in \mathbb{R}^{d_{\text{out}}}$, $\forall i = 1, \cdots, m$。

1. 如图 16.2 所示，对于所有的 $i = 1, \cdots, m$，把输入的 $\boldsymbol{x}^i$ 映射到三元组 $(\boldsymbol{q}^i, \boldsymbol{k}^i, \boldsymbol{v}^i)$：

$$\begin{aligned} \boldsymbol{q}^i &= \boldsymbol{W}_q \boldsymbol{x}^i \in \mathbb{R}^{d_q}, \\ \boldsymbol{k}^i &= \boldsymbol{W}_k \boldsymbol{x}^i \in \mathbb{R}^{d_q}, \\ \boldsymbol{v}^i &= \boldsymbol{W}_v \boldsymbol{x}^i \in \mathbb{R}^{d_{\text{out}}}. \end{aligned}$$

2. 如图 16.3 所示，计算权重向量 $(\boldsymbol{\alpha}^1, \cdots, \boldsymbol{\alpha}^m)$，每个权重向量的大小都是 $m \times 1$。第 i 个权重向量 $\boldsymbol{\alpha}^i$ 依赖于 $\boldsymbol{q}^i$ 和 $(\boldsymbol{k}^1, \cdots, \boldsymbol{k}^m)$：

$$\boldsymbol{\alpha}^i = \text{softmax}\Big(\langle \boldsymbol{q}^i, \boldsymbol{k}^1 \rangle, \langle \boldsymbol{q}^i, \boldsymbol{k}^2 \rangle, \cdots, \langle \boldsymbol{q}^i, \boldsymbol{k}^m \rangle\Big), \qquad \forall\, i = 1, \cdots, m.$$

公式中的 $\langle \cdot, \cdot \rangle$ 是向量内积。由于向量 $\boldsymbol{\alpha}^i$ 是 Softmax 函数的输出，它的元素都是正实数，而且相加等于 1。向量 $\boldsymbol{\alpha}^i$ 的第 j 个元素（记作 α^i_j）表示 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$ 的相关性；$\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$ 越相关，那么元素 α^i_j 就越大。

3. 如图 16.4 所示，计算输出向量 $(\boldsymbol{c}^1, \cdots, \boldsymbol{c}^m)$，每个输出向量的维度都是 d_{out}。第 i 个输出向量 $\boldsymbol{c}^i$ 依赖于 $\boldsymbol{\alpha}^i$ 和 $(\boldsymbol{v}^1, \cdots, \boldsymbol{v}^m)$：

$$\boldsymbol{c}^i = \big[\boldsymbol{v}^1, \boldsymbol{v}^2, \cdots, \boldsymbol{v}^m\big] \cdot \boldsymbol{\alpha}^i = \sum_{j=1}^{m} \alpha^i_j \boldsymbol{v}^j, \qquad \forall\, i = 1, \cdots, m.$$

$\boldsymbol{c}^i$ 是向量 $\boldsymbol{v}^1, \cdots, \boldsymbol{v}^m$ 的加权平均，权重是 $\boldsymbol{\alpha}^i = [\alpha^i_1, \cdots, \alpha^i_m]$。

为什么这种神经网络结构叫做注意力 (Attention) 呢？如图 16.5 所示，向量 $\boldsymbol{x}^i$ 位置上的输出是 $\boldsymbol{c}^i$，它是做加权平均计算出来的：

$$\boldsymbol{c}^i = \alpha^i_1 \boldsymbol{v}^1 + \alpha^i_2 \boldsymbol{v}^2 + \cdots + \alpha^i_m \boldsymbol{v}^m.$$

权重 $\boldsymbol{\alpha}^i = [\alpha^i_1, \cdots, \alpha^i_m]$ 反映出 $\boldsymbol{c}^i$ 最“关注”哪些输入的 $\boldsymbol{v}^j = \boldsymbol{W}_v \boldsymbol{x}^j$。如果权重 α^i_j 大，说明 $\boldsymbol{x}^j$ 对 $\boldsymbol{c}^i$ 的影响较大。$\boldsymbol{c}^i$ 应当重点关注对其影响较大的 $\boldsymbol{x}^j$。

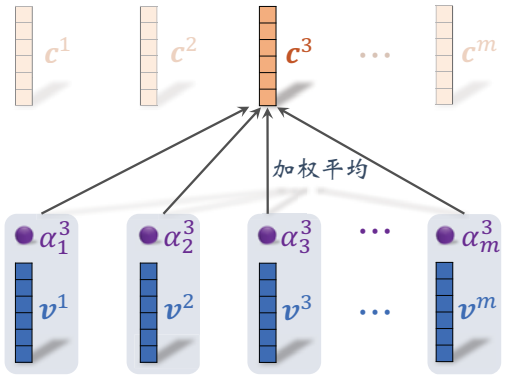

图 16.5: 第 i 个输出向量 $\boldsymbol{c}^i$ 由权重 $\boldsymbol{\alpha}^i = [\alpha^i_1, \cdots, \alpha^i_m]$ 和向量 $(\boldsymbol{v}^1, \cdots, \boldsymbol{v}^m)$ 决定。

上述自注意力层叫做**单头自注意力层** (Single-Head Self-Attention Layer)，简称“单头”。实践中更常用的是**多头自注意力层** (Multi-Head Self-Attention Layer)，简称“多头”，它是多个单头的组合，见图 16.6。设多头由 l 个单头组成。每个单头有自己的 3 个参数矩阵，所以多头一共有 $3l$ 个参数矩阵。它们的输入都是序列 $(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m)$，它们的输出都是长度为 m 的向量序列。

$$\begin{aligned} &\text{第 1 个自注意力层输出：} && (\boldsymbol{c}^1_1, \boldsymbol{c}^2_1, \boldsymbol{c}^3_1, \cdots, \boldsymbol{c}^m_1), \\ &\text{第 2 个自注意力层输出：} && (\boldsymbol{c}^1_2, \boldsymbol{c}^2_2, \boldsymbol{c}^3_2, \cdots, \boldsymbol{c}^m_2), \\ &\qquad\vdots && \qquad\vdots \\ &\text{第 } l \text{ 个自注意力层输出：} && (\boldsymbol{c}^1_l, \boldsymbol{c}^2_l, \boldsymbol{c}^3_l, \cdots, \boldsymbol{c}^m_l). \end{aligned}$$

其中每个向量 $\boldsymbol{c}_j^i$ 的大小都是 $d_{\text{out}} \times 1$。多头的输出记作序列 $(\boldsymbol{c}^1, \cdots, \boldsymbol{c}^m)$，其中每个 $\boldsymbol{c}^i$ 都是做连接 (Concatenation) 得到的：

$$\boldsymbol{c}^i = \left[\boldsymbol{c}_1^i; \boldsymbol{c}_2^i; \cdots; \boldsymbol{c}_l^i\right] \in \mathbb{R}^{ld_{\text{out}}}, \qquad \forall\, i = 1, \cdots, m.$$

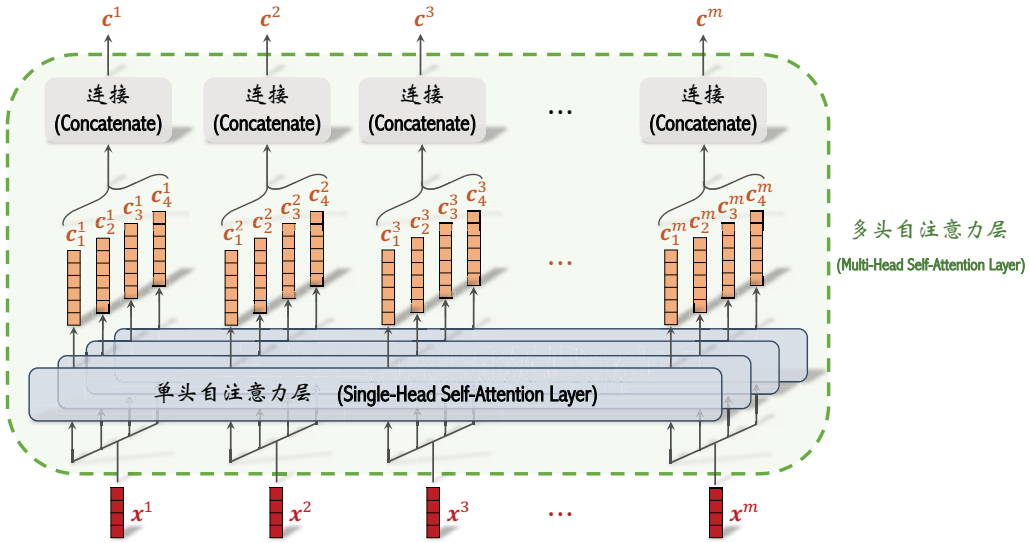

图 16.6: 这个例子中，多头自注意力层由 $l = 4$ 个单头自注意力层组成。

总结一下，多头自注意力层把长度为 m 的向量序列映射到同等长度的向量序列；长度 m 可以任意变化，神经网络结构无需改变。实现一个多头自注意力层需要指定三个超参数：单头的数量 l、每个单头输出的大小 d_{out}、向量 q^i 和 $\boldsymbol{k}^i$ 的大小 d_q。多头的输出是长度为 m 的向量序列，每个向量的大小是 $ld_{\text{out}} \times 1$。超参数 d_q 不影响输出的大小，它只在计算权重向量 $\boldsymbol{\alpha}^1, \cdots, \boldsymbol{\alpha}^m$ 的时候使用。

16.2 自注意力在中心化训练中的应用

自注意力机制 (Self-Attention) 是改进多智能体强化学习 (MARL) 的一种有效技巧，可以应用在中心化训练或中心化决策当中。多智能体系统中有 m 个智能体，每个智能体有自己的观测（记作 $o^1, \cdots, o^m$）和动作（记作 $a^1, \cdots, a^m$）。我们考虑非合作关系的 MARL。如果做中心化训练，需要用到 m 个状态价值网络

$$v\Big(\big[o^1, \cdots, o^m\big]; \boldsymbol{w}^1\Big), \quad \cdots, \quad v\Big(\big[o^1, \cdots, o^m\big]; \boldsymbol{w}^m\Big),$$

或 m 个动作价值网络

$$q\Big(\big[o^1, \cdots, o^m\big], \big[a^1, \cdots, a^m\big]; \boldsymbol{w}^1\Big), \quad \cdots, \quad q\Big(\big[o^1, \cdots, o^m\big], \big[a^1, \cdots, a^m\big]; \boldsymbol{w}^m\Big).$$

由于是非合作关系，m 个价值网络有各自的参数，而且它们的输出各不相同。我们首先以状态价值网络 v 为例讲解神经网络的结构。

不使用自注意力的状态价值网络：

图 16.7 是状态价值网络 v 最简单的实现。每个价值网络是一个独立的神经网络，有自己的参数。底层提取特征的卷积网络可以在 m 个价值网络中共享（即复用），而上层的全连接网络不能共享。神经网络的输入是所有智能体的观测的连接 (Concatenation)，输出是实数

$$\widehat{v}^i = v\Big(\big[o^1, \cdots, o^m\big]; \boldsymbol{w}^i\Big).$$

这种简单的神经网络结构有几个不足之处。

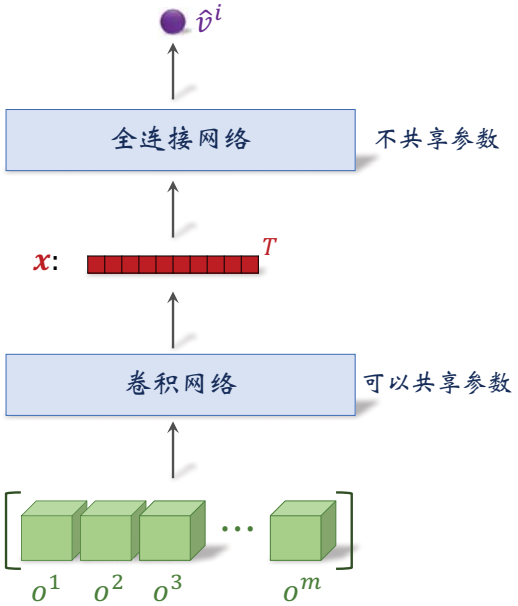

图 16.7: 第 i 号状态价值网络最简单的实现。

- 智能体数量 m 越大，神经网络的参数越多。神经网络的输入是 m 个观测的连接，它们被映射到特征向量 $\boldsymbol{x}$。m 越大，我们就必须把向量 $\boldsymbol{x}$ 维度设置得越大，否则 $\boldsymbol{x}$ 无法很好地概括 $[o^1, \cdots, o^m]$ 的完整信息。$\boldsymbol{x}$ 维度越大，全连接网络的参数就越多，神经网络就越难训练（即得收集更多的经验才能训练好神经网络）。
- 当 m 很大的时候，并非所有智能体的观测 $o^1, \cdots, o^m$ 都与第 i 号智能体密切相关。第 i 号智能体应当学会判断哪些智能体最相关，并重点关注密切相关的智能体，避免决策受无关的智能体干扰。
- 价值网络的输入是 $[o^1, \cdots, o^m]$，即所有观测的连接。如果交换其中 o^j 和 o^k 的位置，那么价值网络输出的 $\widehat{v}^i$ 会发生变化。理想情况下，只要 $j \neq i$、$k \neq i$，那么交换 o^j 和 o^k 的位置就不该改变第 i 号价值网络的输出值 $\widehat{v}^i$。

使用自注意力的状态价值网络： 图 16.8 是对状态价值网络更好的实现方式，避免了上面讨论的三种不足之处。神经网络的结构是这样的：

- 输入仍然是所有智能体的观测 $o^1, \cdots, o^m$。对于所有的 i，用一个卷积网络把 o^i 映射到特征向量 $\boldsymbol{x}^i$。这些卷积网络的参数都是相同的。
- 自注意力层的输入是向量序列 $(\boldsymbol{x}^1, \cdots, \boldsymbol{x}^m)$，输出序列 $(\boldsymbol{c}^1, \cdots, \boldsymbol{c}^m)$。向量 $\boldsymbol{c}^i$ 依赖于所有的观测 $o^1, \cdots, o^m$，但是 $\boldsymbol{c}^i$ 主要取决于最密切相关的一个或几个观测。
- 第 i 号全连接网络把向量 $\boldsymbol{c}^i$ 作为输入，输出一个实数 $\widehat{v}^i$，作为第 i 号价值网络 $v([o^1, \cdots, o^m]; \boldsymbol{w}^i)$ 的输出。在非合作关系的设定下，m 个状态价值网络是不同的，因此 m 个全连接网络不共享参数。

图 16.8 中只用了一个自注意力层。其实可以重复自注意力层，比如：

$$\cdots \longrightarrow \text{自注意力层} \longrightarrow \text{全连接层} \longrightarrow \text{自注意力层} \longrightarrow \text{全连接层} \longrightarrow \cdots$$

自注意力的层数是一个超参数，需要用户自己调。

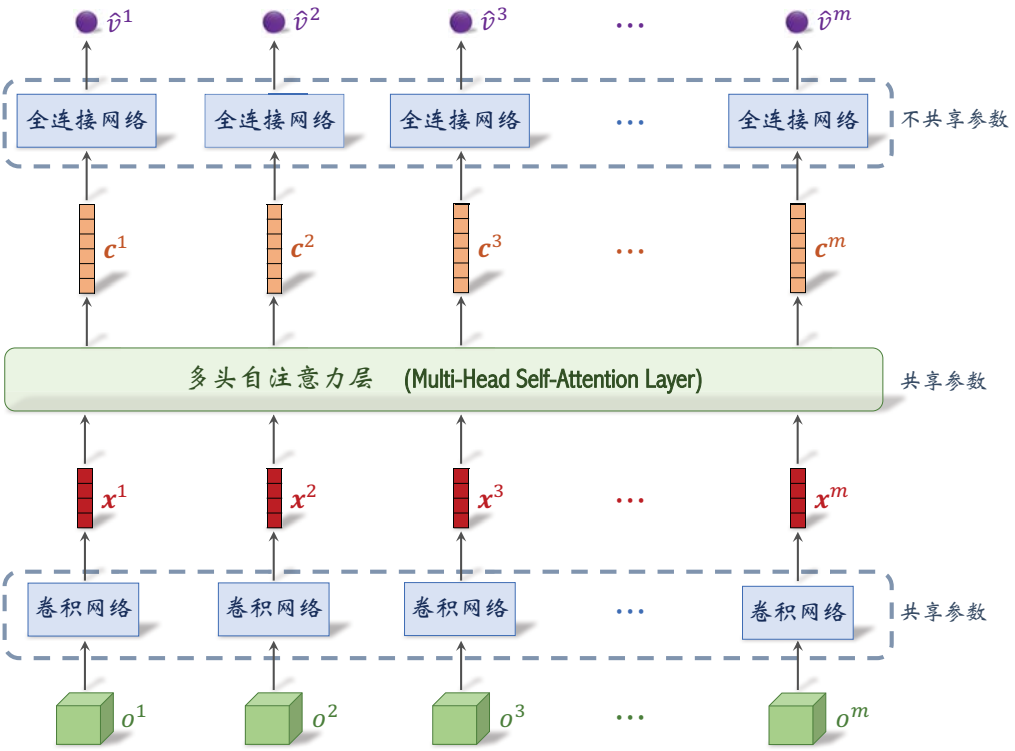

图 16.8: 带有自注意力的状态价值网络。图中的 $\widehat{v}^i = v\big([o^1, \cdots, o^m]; \boldsymbol{w}^i\big)$ 是第 i 个动作价值网络的输出。

使用自注意力的动作价值网络： 上一章介绍了 MADDPG，它是一种连续控制方法，用于非合作关系的设定。它的架构是“中心化训练 + 去中心化决策”，在中央控制器上部署 m 个动作价值网络，把第 i 个记作：

$$\widehat{q}^i \;=\; q\Big(\,[o^1, \cdots, o^m]\,,\,[\boldsymbol{a}^1, \cdots, \boldsymbol{a}^m]\,;\,\boldsymbol{w}^i\Big).$$

它的输入是所有智能体的观测和动作，输出是实数 $\widehat{q}^i$，表示动作价值。可以按照图 16.9

实现动作价值网络。在 MADDPG 中使用这样的神经网络结构可以提高 MADDPG 的表现，尤其是当 m 较大的时候，效果的提升较大。

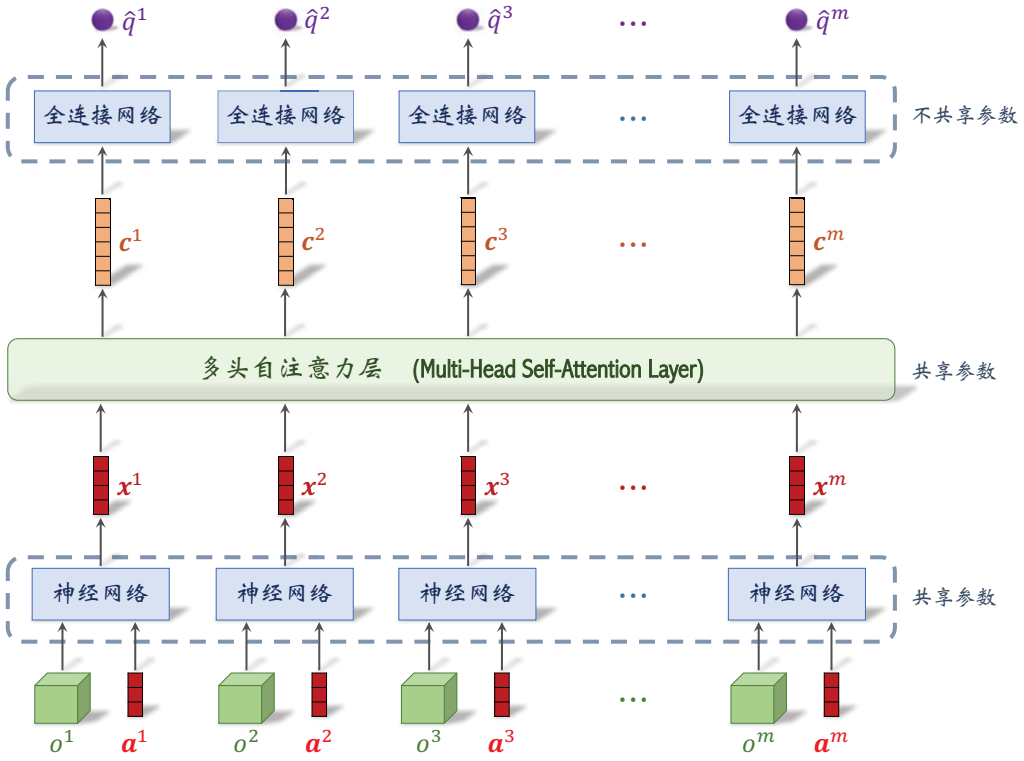

图 16.9: 带有自注意力的动作价值网络。图中的 $\widehat{q}^i = q\big([o^1,\cdots,o^m],[a^1,\cdots,a^m];\boldsymbol{w}^i\big)$ 是第 i 个动作价值网络的输出。

使用自注意力的中心化策略网络： 对于“中心化训练 + 中心化决策”的系统架构，需要在中央控制器上部署 m 个策略网络，每个策略网络都需要知道所有 m 个智能体的观测 $o^1,\cdots,o^m$。

- 对于离散控制，第 i 个策略网络记作：

$$\widehat{\boldsymbol{f}}^i \;=\; \pi\Big(\;\cdot\;\Big|\;[o^1,\cdots,o^m]\,;\,\boldsymbol{\theta}^i\Big).$$

  策略网络的输出是向量 $\widehat{\boldsymbol{f}}^i$，它的维度是第 i 个动作空间的大小 $|\mathcal{A}^i|$，$\widehat{\boldsymbol{f}}^i$ 的元素表示每种动作的概率。根据 $\widehat{\boldsymbol{f}}^i$ 做随机抽样，得到动作 a^i，第 i 号智能体执行这个动作。

- 对于连续控制，第 i 个策略网络记作：

$$\boldsymbol{a}^i \;=\; \boldsymbol{\mu}\Big(\;[o^1,\cdots,o^m]\,;\,\boldsymbol{\theta}^i\Big).$$

  它的输出是动作 $\boldsymbol{a}^i$，它是 d 维向量，d 是连续控制问题的自由度。第 i 号智能体执行动作 $\boldsymbol{a}^i$。

不管是离散控制还是连续控制，上述两种策略网络中都可以使用自注意力层，神经网络的结构与图 16.8 中的 $v(s;\boldsymbol{w}^i)$ 几乎一样，唯一区别是神经网络的输出由实数 $\widehat{v}^1,\cdots,\widehat{v}^m$

变成向量 $\widehat{\boldsymbol{f}}^1, \cdots, \widehat{\boldsymbol{f}}^m$ 或者 $\boldsymbol{a}^1, \cdots, \boldsymbol{a}^m$。

总结： 自注意力机制在**非合作关系**的 MARL 中普遍适用。如果系统架构使用**中心化训练**，那么 m 个价值网络可以用一个神经网络实现，其中使用自注意力层。如果系统架构使用**中心化决策**，那么 m 个策略网络也可以实现成一个神经网络，其中使用自注意力层。在 m 较大的情况下，使用自注意力层对效果有较大的提升。

第 十六 章 相关文献

注意力机制 (Attention) 由 2015 年的论文 [6] 提出，将注意力机制与 RNN 结合，可以大幅提升 RNN 在机器翻译任务上的表现。2017 年的论文 [111] 提出 Transformer 模型，去掉 RNN，只保留注意力，在机器翻译任务上取得了远优于 RNN+ 注意力的表现。2019 年的论文 [53] 将注意力层用到多智能体的 Actor-Critic 中。