14 注意力与自注意力

机器学习 · 深度学习基础

学习目标

- 解释注意力如何随查询改变聚合结果，区分查询、键和值的职责
- 跟踪一次缩放点积注意力计算，核对评分、归一化、输出及矩阵形状
- 根据表示来源区分自注意力与交叉注意力，说明Q / K / V如何形成
- 跟踪多头注意力的投影、并行聚合、拼接及输出映射

1 查询相关的信息聚合

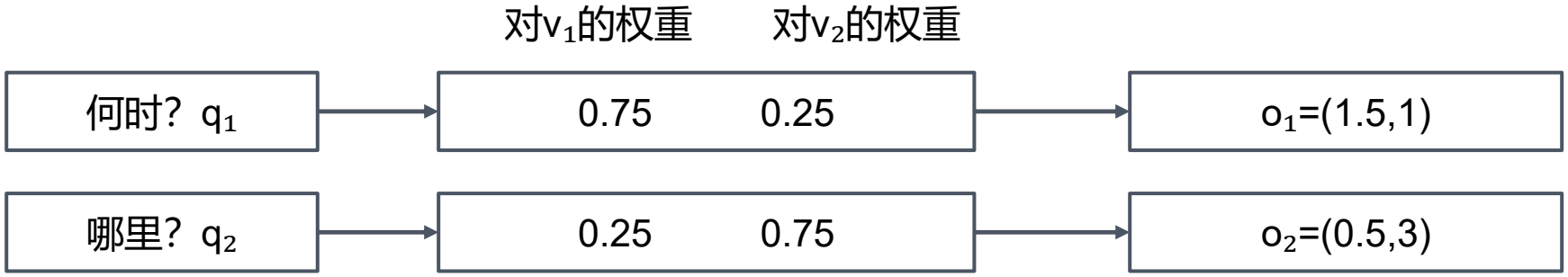
查询、键和值如何形成聚合结果；先算一次，再解释缩放。

当前需要不同，聚合权重也可以不同

同一份会议资料包含时间安排和会场信息。**回答“何时开始”和“在哪里举行”时，需要读取的内容不同**；若始终平均组合两条记录，就无法随问题改变信息的贡献。

把问题表示为查询，保留两条记录的表示，就可以按当前需要改变聚合权重：

同一组记录：v_1=(2,0) 时间信息；v_2=(0,4) 会场信息



问题不同 → 给定分配不同 → 同一组值形成不同组合

固定平均的权重为(0.5,0.5)，输出为(1,2)。不同查询的权重和输出可能不同；图中权重为教学给定，下一步建立可计算的匹配依据。

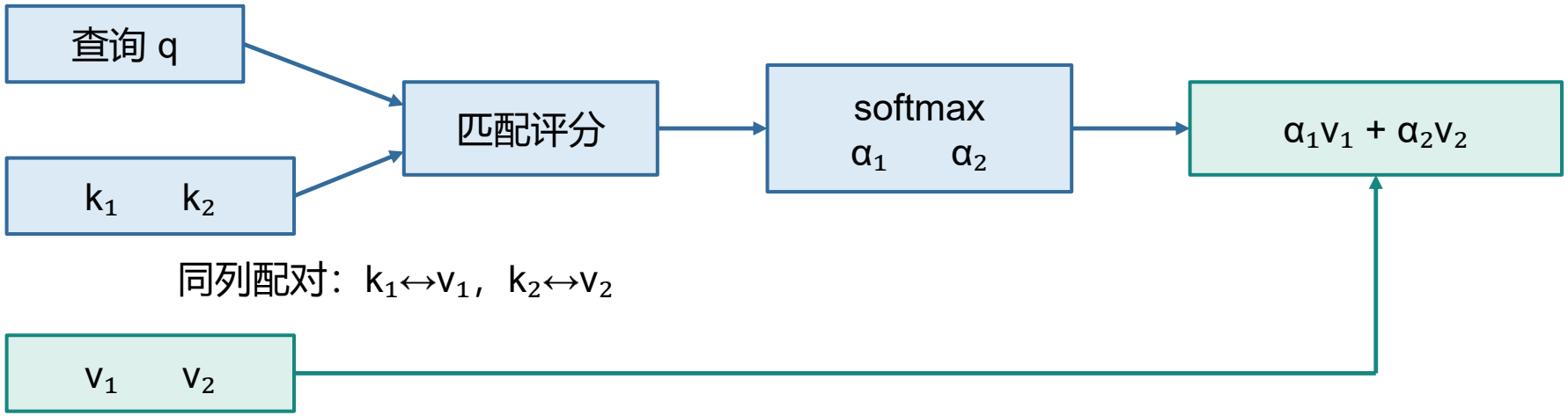
注意力输出是针对当前查询的组合

注意力机制根据当前查询与各条目描述的匹配程度确定权重，再对条目内容加权求和，形成当前查询的输出。

- **怎样把读取需要变成可计算的权重?**
  - 把当前读取条件表示为向量，称为**查询**；它与各条内容的描述形成匹配评分，再转为组合权重。查询不必是一句自然语言问题。
- **为什么先加权，再相加?**
  - 一次读取可能需要综合多条信息。权重先缩放各条内容向量，再在共同的特征坐标中逐分量相加，得到同维度的输出表示。**乘权重控制份额，相加合成表示。**
  - 输出是聚合后的表示，不是最终答案；相加也不保证分量增大，正负分量可能抵消。
  - 核心关系是：**查询相关的匹配 → 动态权重 → 信息聚合。**是否有助于任务仍需学习与实际评价。

查询与键负责匹配，值提供被组合的内容

- 查询（query）：当前读取条件的向量$\boldsymbol{q}$
- **键**（key）：第j条内容用于匹配的向量$\boldsymbol{k}_j$
- **值**（value）：第j条参与加权组合的向量$\boldsymbol{v}_j$，与该键配对



键说明“怎样匹配”，值提供“组合什么”。二者条目一一对应，但向量内容与维度可以不同；定义由计算职责区分。

评分、权重和输出是三个不同对象

对一个查询$\boldsymbol{q}$和m个配对的键值：

$$s_j = score(\boldsymbol{q}, \boldsymbol{k}_j), \alpha_j = \frac{\exp(s_j)}{\sum_{r=1}^{m} \exp(s_r)}, \boldsymbol{o} = \sum_{j=1}^{m} \alpha_j \boldsymbol{v}_j$$

- 评分s_j是匹配函数score产生的实数；比较的是同一查询对各键的匹配
- softmax把这一组评分转为非负权重，且$\sum_j \alpha_j = 1$
- 输出$\boldsymbol{o}$按权重组合值；输出维度与值相同，未必等于任何一条值

归一化在当前查询对应的可用键之间进行。权重具有概率形式，但不自动等于外部事件的真实概率。

一次查询完整经过评分、归一化与聚合

给定$\boldsymbol{q}_1 = (\sqrt{2}, 0)^\top$，两键为$(1,0)^\top, (0,1)^\top$，两值沿用$(2,0)^\top, (0,4)^\top$，$d_k = 2$。

本例采用**缩放点积评分**$s_j = \boldsymbol{q}_1^\top \boldsymbol{k}_j / \sqrt{d_k}$，其中$d_k$是查询与键的维度。先跟踪完整计算，再解释为什么需要这一缩放。

条目j	点积$q_1^T k_j$	除以√2后的s_j	exp(s_j)	权重α_j
1	√2	1	e	a=e/(e+1)≈0.7311
2	0	0	1	b=1/(e+1)≈0.2689

两项指数之和为$e + 1$，每项都除以同一分母；这里$e \approx 2.7183$。

$\boldsymbol{o}_1 = a(2,0)^\top + b(0,4)^\top = (2a, 4b)^\top \approx (1.4621, 1.0758)^\top$

较大的第一项权重没有让输出只剩第一条值；第二条仍按其权重参与组合。

缩放控制点积进入softmax时的幅度

缩放点积评分使用同维查询与键的点积，并除以$\sqrt{d_k}$：

$$s_j = \frac{\boldsymbol{q}^\top \boldsymbol{k}_j}{\sqrt{d_k}}$$

假定键维度$d_k = 64$，给定两个点积为8和0：

进入softmax的评分	第一项权重	第二项权重
未缩放：(8, 0)	0.9997	0.0003
除以√64：(1, 0)	0.7311	0.2689

固定第二评分时，第一权重对自身评分的导数为$\alpha_1(1-\alpha_1)$，两行分别约为0.0003、0.1966。**较大的评分差使权重接近单一选择，对评分变化也更不敏感。**缩放控制这种幅度，不保证权重均匀或消除所有训练困难。

为什么除以键维度的平方根

核心解释：维度增加，点积的典型波动可能增大；缩放用于控制进入softmax的数值尺度。

在所有查询与键分量相互独立、均值为0、方差为1的假设下：

$$Var(\boldsymbol{q}^\top \boldsymbol{k}) = d_k, Std(\boldsymbol{q}^\top \boldsymbol{k}) = \sqrt{d_k}, Var\left(\frac{\boldsymbol{q}^\top \boldsymbol{k}}{\sqrt{d_k}}\right) = 1$$

标准差Std刻画典型波动幅度，方差Var是其平方。因此按上述假设，除以$\sqrt{d_k}$把标准差调回1。

可选推导：先算一项，再对各项求和。 令$z_i = q_i k_i$，E表示均值。独立性给出$E[z_i] = 0$、$E[z_i^2] = E[q_i^2]E[k_i^2] = 1$，故$Var(z_i) = 1$。

$$Var\left(\sum_{i=1}^{d_k} z_i\right) = \sum_{i=1}^{d_k} Var(z_i) = d_k$$

训练后的向量不必满足这些假设；缩放不保证softmax不饱和。

评分函数决定怎样比较查询与键

缩放点积只是评分函数的一种实现；换评分规则后，仍可沿用softmax与值聚合。

评分方式	匹配怎样形成	当前需要理解的条件
点积	相同维度的查询与键逐分量乘积后求和	向量长度和方向都会影响结果
缩放点积	点积再除以键维度的平方根	控制评分幅度；后续仍需softmax
加性评分	查询和键分别映射，再经非线性网络产生实数分数	映射后的维度需可组合，参数由任务学习

本讲用缩放点积展开完整计算。它便于把多个查询与键的匹配写为矩阵乘法；注意力的“匹配后加权组合”并不限定只能使用这一评分函数。

2 完整计算与表示来源

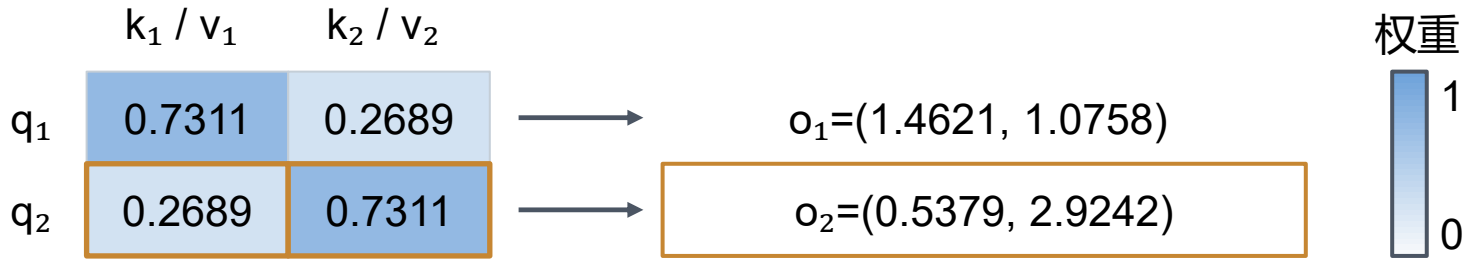
从单查询扩展为多查询矩阵，区分自 / 交叉注意力及归一化轴。

多个查询按行计算各自的权重

增加查询$\boldsymbol{q}_2 = (0, \sqrt{2})^\top$，键和值不变；每行存一条向量的转置，省略批量维：

$\boldsymbol{I}_2$是二阶单位矩阵；$diag(2,4)$以2、4为对角元素，其余元素为0。

$$\boldsymbol{Q} = \sqrt{2}\boldsymbol{I}_2, \boldsymbol{K} = \boldsymbol{I}_2, \boldsymbol{V} = diag(2,4), \boldsymbol{S} = \frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{2}} = \boldsymbol{I}_2$$



A：每行一份分配；颜色越深，权重越大

$$\boldsymbol{O} = \boldsymbol{A}\boldsymbol{V} = \begin{bmatrix} 2a & 4b \\ 2b & 4a \end{bmatrix}$$

每个查询有自己的一行权重；每行沿键归一化。输出行对应查询，输出列对应值的特征。

改变键与改变值，影响不同环节

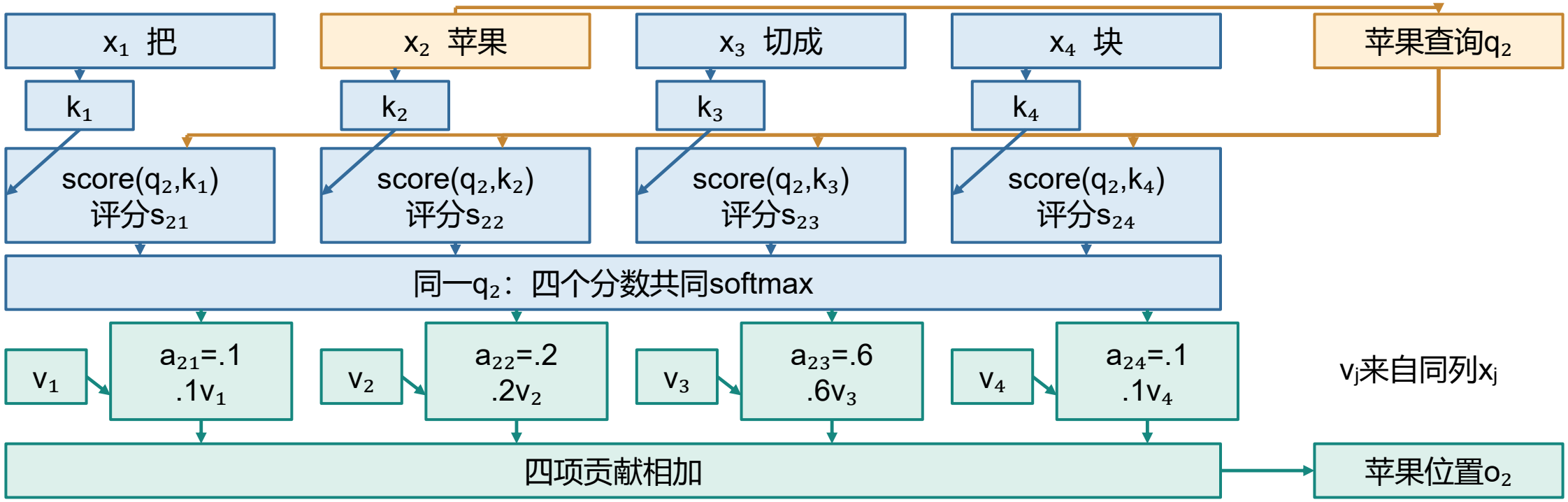
固定第一查询 q_1，沿用两键、两值及 $a \approx 0.7311, b \approx 0.2689$：

对原例的改变	对当前两行值的权重	输出
不改变	(a, b)	(2a, 4b)
只交换两行键，值不动	(b, a)	(2b, 4a)
键和值配对交换行	(b, a)	(2a, 4b)
只把第二值改为(0, 8)	(a, b)	(2a, 8b)

- **键改变评分与权重；值改变被组合的内容**，本例中不参与评分
- 配对重排只改变记录顺序；固定查询且键值配对关系不变时，加权和不变

自注意力让每个位置从同一序列聚合信息

“把 苹果 切成 块”与“看 苹果 发布 新机”：**上下文不同，聚合结果可能不同。**下图取第一句。



$$\boldsymbol{o}_2 = .1\boldsymbol{v}_1 + .2\boldsymbol{v}_2 + .6\boldsymbol{v}_3 + .1\boldsymbol{v}_4$$

Q / K / V同源于X，各位置重复此机制；同源不表示Q=K=V或只读自己。**权重为教学给定。**

三个投影把同一输入接到已完成的计算

上页说明“当前位置怎样读取上下文”；本页说明“Q / K / V怎样产生”。回到前面的**两位置数值例**：将各位置的输入表示按行排列为矩阵$\boldsymbol{X}$。对同一个$\boldsymbol{X}$分别作三次线性映射，得到用于提出查询的$\boldsymbol{Q}$、用于匹配的$\boldsymbol{K}$和用于聚合的$\boldsymbol{V}$：

$$\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}^{\mathrm{Q}}, \boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}^{\mathrm{K}}, \boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}^{\mathrm{V}}$$

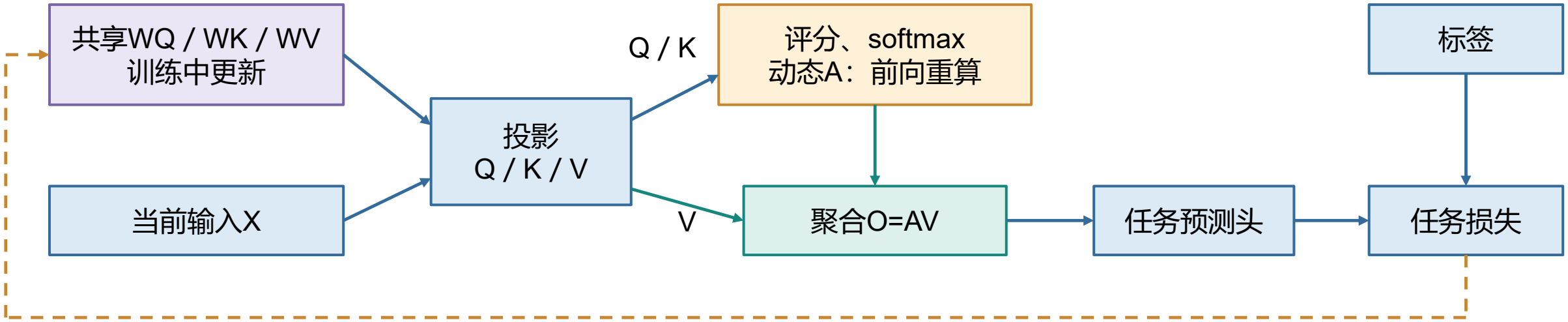
取$\boldsymbol{X} = \boldsymbol{I}_2$，给定$\boldsymbol{W}^{\mathrm{Q}} = \sqrt{2}\boldsymbol{I}_2$、$\boldsymbol{W}^{\mathrm{K}} = \boldsymbol{I}_2$、$\boldsymbol{W}^{\mathrm{V}} = diag(2,4)$，就得到前面完整单查询与多查询算例使用的Q、K、V。

输入位置	自己的权重行	输出表示
位置1	(a, b)	(2a, 4b)
位置2	(b, a)	(2b, 4a)

投影说明这些向量如何产生；注意力说明位置间如何组合信息。此处参数为手工给定，实际任务中由损失学习，不再重复评分与softmax计算。

学习投影参数，按输入计算权重

同一层共享投影参数W^{Q}, W^{K}, W^{V}。**训练更新W；每次前向按当前输入计算A。**



梯度回传与参数更新示意：经中间计算图回传，再更新共享W

同一W可因输入变化得到不同A。任务损失沿评分 / 权重与值两路回传。**A不另设监督标签，也不逐输入保存。**

矩阵形状与代码共同标明归一化方向

```
scores = Q @ K.T / math.sqrt(Q.shape[-1])
weights = torch.softmax(scores, dim=-1)
output = weights @ V
```

形状与对象

上页解释参数怎样学习；这里核对同一算子的前向计算。n为查询数，m为键值对数；d_k为查询 / 键维度，d_v为值维度。

Q：$n \times d_k$

K：$m \times d_k$；V：$m \times d_v$

评分及权重A：$n \times m$；输出O：$n \times d_v$

Q、K、V按行存条目。A的行对应查询、列对应键；O的列对应值的特征。

轴与条件

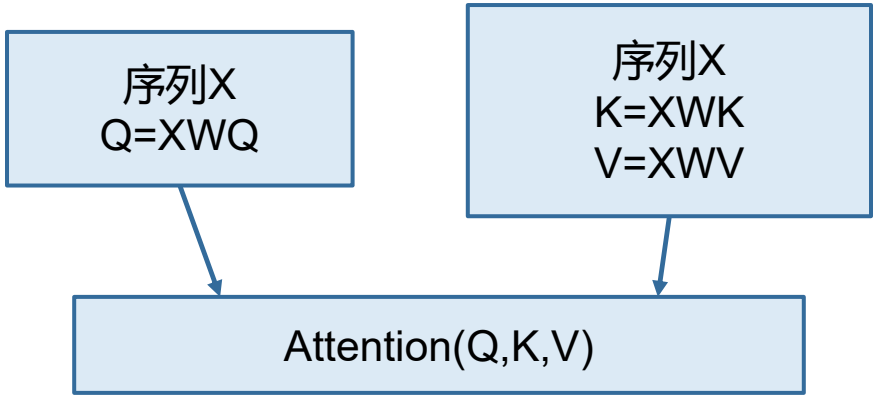
dim=-1沿键位置归一化；K.T使每个查询与每个键点积。省略批量维和掩码；键值行保持配对，查询数可与键数不同。

交叉注意力改变查询与键值的来源

翻译“把苹果切成块”时，目标端已生成“Cut the”，需要预测下一个词。**目标端需要按当前生成进度读取源端信息。**

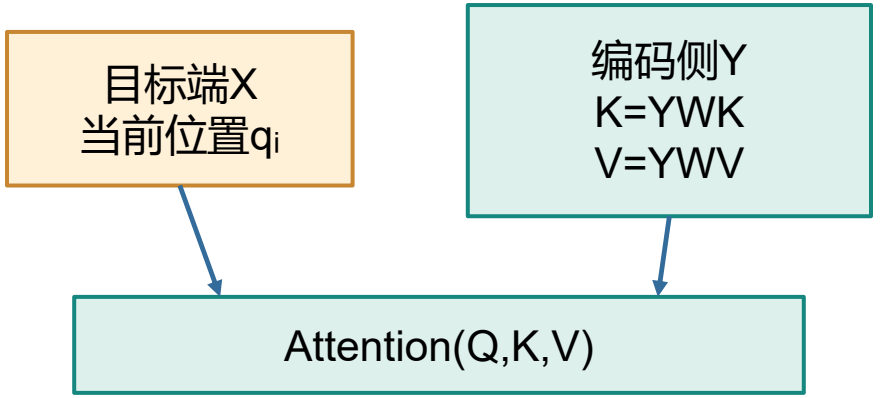
沿用前页的算子：**自注意力同源，交叉注意力跨来源。**

自注意力：同一序列



输出位置仍对应X的查询

交叉注意力：编码—解码示例



聚合源端V，输出仍属于目标位置i

目标端当前表示生成Q，源端各位置的表示生成K和V；匹配后聚合源端信息，所得表示供后续词语预测使用。

下一页另用2查询 / 3键值的独立数值例：权重A为2×3，输出O为$2\times d_v$。

两个查询怎样读取三条内容

沿用 $\boldsymbol{Q} = \sqrt{2}\boldsymbol{I}_2$，给定另一来源已投影的键和值，$d_k = 2$：

条目	K的对应行	V的对应行
1	(1, 0)	(2, 0)
2	(0, 1)	(0, 4)
3	(1, 1)	(1, 1)

$$\boldsymbol{S} = \frac{\boldsymbol{Q}\boldsymbol{K}^{\top}}{\sqrt{2}} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \boldsymbol{A} = \begin{bmatrix} c & d & c \\ d & c & c \end{bmatrix}$$

这里 $c = e/(2e+1) \approx 0.4223$，$d = 1/(2e+1) \approx 0.1554$，每行满足 $2c + d = 1$。

$$\underbrace{\boldsymbol{A}}_{2\times 3} \times \underbrace{\boldsymbol{V}}_{3\times 2} = \underbrace{\boldsymbol{O}}_{2\times 2}$$

两行输出仍对应两个查询；三个键值对提供被读取的内容，不会让输出变成三行。

每个查询在自己的键之间归一化

沿用上页评分矩阵，比较按行与按列归一化的结果：

$$S = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

归一化方式	第一行权重	每行权重之和
沿键位置：每行softmax	(c, d, c)	1
沿查询位置：每列softmax	(a, b, 1/2)	1.5

- 正确问题是：对这个查询，各条键值怎样分配贡献；因此固定查询行，沿键所在的列归一化
- 按列归一化会让不同查询争用同一键的权重，计算的是另一种关系
- 前面2×2例的评分恰好对称，按错轴也可能得到相同数字；**必须根据行列身份选择dim**，而非只看一次输出

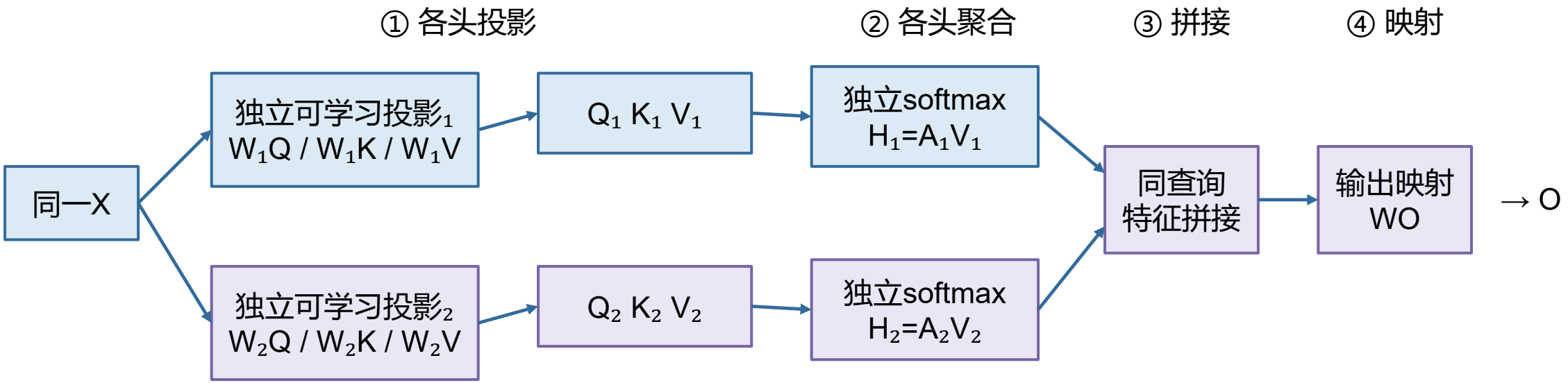
3 多头聚合与结果合并

跟踪逐头评分、聚合、特征拼接与输出映射。

多个头先分别聚合，再合并结果

单头在一组投影下，为每个查询形成一份聚合结果。**若希望组合前先保留多份结果，可以使用多组可学习投影**，分别形成匹配与值表示。

多头注意力让各头独立投影、评分、归一化与聚合；同一查询的各头结果随后才合并。

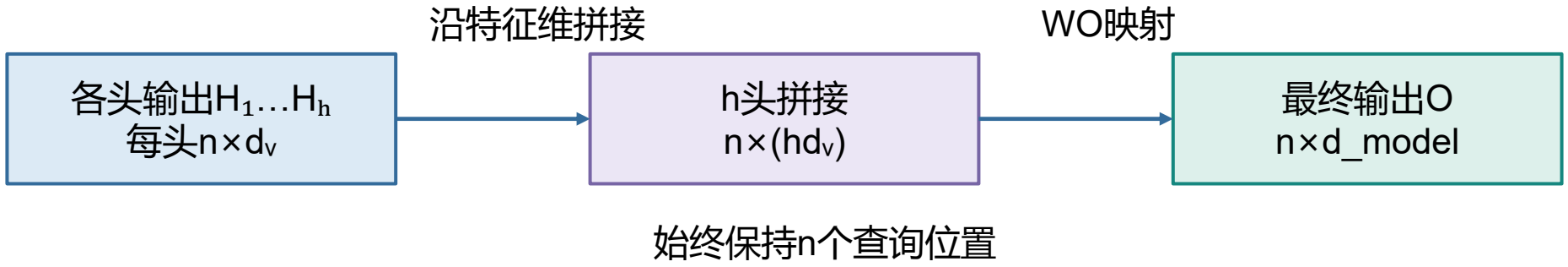


单头：一组投影、一份结果　　多头：多组投影、多份结果后组合

多组投影提供可学习的多份表示，**不保证各头必然互补或具有固定语义**，也不意味着单头不能融合多种信息。

各头沿特征维拼接，查询位置保持对应

先看各头输出怎样合并：**汇集h个头的输出 → 沿特征维拼接 → 输出映射。**



输入X为$n \times d_{\text{model}}$；下表只核对各线性映射的参数形状：

参数	形状
每头WQ / WK	$d_{\text{model}} \times d_k$
每头WV	$d_{\text{model}} \times d_v$
输出映射WO	$(hd_v) \times d_{\text{model}}$

$$\boldsymbol{H}_r = Attention(\boldsymbol{X}\boldsymbol{W}_r^{\mathrm{Q}}, \boldsymbol{X}\boldsymbol{W}_r^{\mathrm{K}}, \boldsymbol{X}\boldsymbol{W}_r^{\mathrm{V}}), \boldsymbol{O} = Concat(\boldsymbol{H}_1, \ldots, \boldsymbol{H}_h)\boldsymbol{W}^{\mathrm{O}}$$

每头查询 / 键维度为d_k，值维度为d_v；常见配置取$d_k = d_v = d_{\text{model}}/h$，具体维度由模型配置决定。

两个头如何形成不同评分

沿用 $\boldsymbol{X} = \boldsymbol{I}_2$，两个头各取 $d_k = d_v = 1$，所以每头缩放因子为1。给定投影：

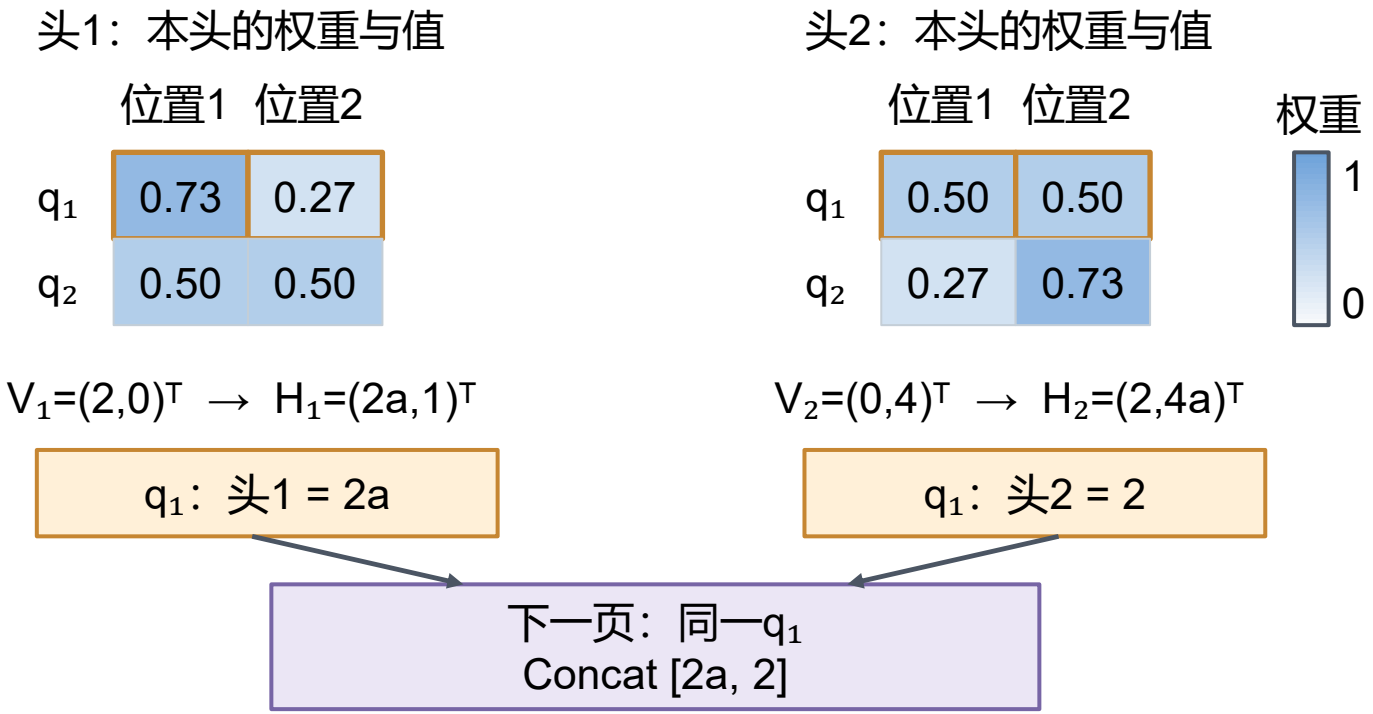
头	WQ=WK	WV
1	$(1, 0)^T$	$(2, 0)^T$
2	$(0, 1)^T$	$(0, 4)^T$

$$\boldsymbol{S}_1 = \boldsymbol{Q}_1\boldsymbol{K}_1^\top = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \boldsymbol{S}_2 = \boldsymbol{Q}_2\boldsymbol{K}_2^\top = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

因X为单位矩阵，Q / K / V直接等于对应投影矩阵。同一输入经不同投影形成不同评分；每个头下一步都沿自己的键位置归一化。

各头分别归一化并组合自己的值

各头独立softmax，沿用$a = e/(e+1), b = 1/(e+1)$。**固定q_1比较两头，再接Concat。**

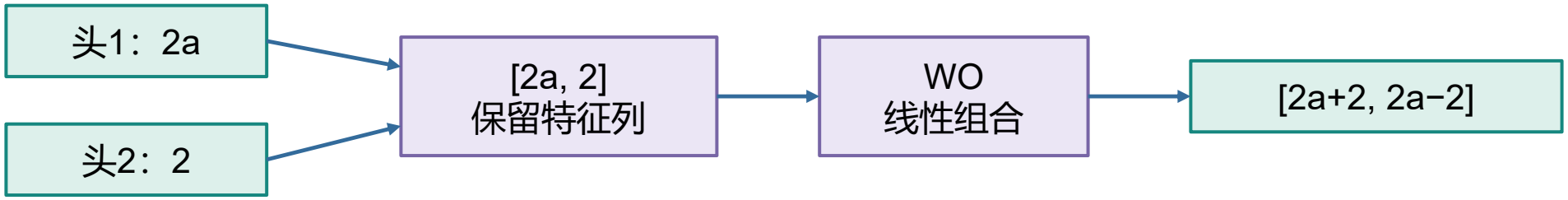


a≈0.7311，b≈0.2689；颜色越深，权重越大

不同投影形成各自的权重和值表示，q_1得到两份结果。即使权重同为均匀分配，也可能因本头V不同而得到不同输出。

拼接保留每个查询的各头结果，再由WO组合

上页各头已独立聚合；现在固定第一查询，**先保留两份结果，再通过输出映射组合。**



对两个查询逐行拼接，给定输出映射：

$$\boldsymbol{C} = Concat(\boldsymbol{H}_1, \boldsymbol{H}_2) = \begin{bmatrix} 2a & 2 \\ 1 & 4a \end{bmatrix}, \boldsymbol{W}^{\mathrm{O}} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$\boldsymbol{O} = \boldsymbol{C}\boldsymbol{W}^{\mathrm{O}} = \begin{bmatrix} 2a+2 & 2a-2 \\ 1+4a & 1-4a \end{bmatrix}$$

本例两列为各头结果的和与差，输出仍对应两个查询。WO为给定的可学习参数；**无dropout时各头先加权平均，WO后可超出各头值范围**，如本例第二列为负。

注意力权重的解释与位置边界

权重与模型解释

- 大权重表示当前聚合的系数较大。值的大小、方向及后续计算也参与结果，**权重大小不能直接确定因果影响。**

位置与可读范围

- 若输入不含位置 / 顺序信息，也无位置相关偏置与掩码，标准自注意力在输入位置置换后，输出也相应置换；它本身不赋予内容先后顺序。位置信息与可读范围需要另外约定，下一讲再展开。

本讲总结

- **为什么需要注意力？** 不同查询可能需要不同信息；查询与键形成匹配权重，对配对值进行动态聚合
- **自注意力实现什么？** 同一输入的每个位置生成查询，读取可访问位置的信息，形成新的表示；交叉注意力则从另一组表示读取键值
- **为什么使用多头？** 多组投影并行形成多份聚合结果，同一查询的结果沿特征拼接，再由输出映射组合；不保证各头必有不同语义

计算主线：$\boldsymbol{A} = softmax_{\mathrm{row}}(\boldsymbol{Q}\boldsymbol{K}^{\top}/\sqrt{d_k})$，$\boldsymbol{O} = \boldsymbol{A}\boldsymbol{V}$。每行对应一个查询，每行在键的位置间归一化。