# 幅度注意力：别让一组相似的键窃取你的概率质量

Vuk Rosić

2026 年 2 月 15 日

## 注意力对几何视而不见

标准注意力纯粹基于查询（Query）与单个键（Key）之间的成对兼容性来计算权重：

$$\mathrm{Attn}(Q,K,V)=\mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

这种方式隐含地假设所有键提供的信息都是独立的。它对**键集合本身的内在几何结构**视而不见，尤其无法考虑**冗余性**。

在高维语义空间中，键往往聚集成密集的群组——重复的词元（Token）、近义词或反复出现的功能模式。我们将这些称为**分组键（Grouped Keys）**。

### 劫持机制

这些分组键利用了 softmax 归一化中的一个漏洞，即**概率劫持（Probability Hijacking）**。

1. **形成群组**：系统性的冗余意味着许多键 $\{k_1,\dots,k_m\}$ 占据向量空间中的同一区域。因此，它们与查询的点积几乎相同：$q\cdot k_i\approx$ 常数。
2. **劫持**：softmax 函数在分母中对这些分数的指数值求和。一个包含 50 个平庸键的簇——仅仅因为数量众多——就能积累巨大的概率质量，淹没一个高度相关的“独特”键。

例如，考虑一个关键键 $k_{\text{unique}}$，其相关性分数为 $e^{q\cdot k_u}=20$，以及一个包含 50 个冗余键的簇，每个键的相关性分数为 $e^{q\cdot k_i}=1$。注意力权重变为：

$$\mathrm{Weight}(k_{\text{unique}})=\frac{20}{20+\underbrace{(1+\cdots+1)}_{50\text{ 次}}}=\frac{20}{70}\approx\mathbf{0.28}$$

$$\mathrm{Weight}(\text{簇})=\frac{50}{70}\approx\mathbf{0.71}$$

尽管独特键的相关性是任何单个簇成员的 **20 倍**，但冗余群组仅凭数量优势就主导了注意力机制。

模型被迫过度关注信号的频率而非其信息含量。它浪费容量去检索同一冗余特征 50 次，同时淹没了仅出现一次的关键信号。

# 数学解法：幅度（Magnitude）

为了解决这个问题，我们需要一种方法来衡量一个集合中 **“有效点数”**。

直觉上，如果你有 3 个键向量：

- 如果它们彼此远离（正交），它们提供 **3 个信息单元**。
- 如果它们完全相同（克隆），它们仅提供 **1 个信息单元**。
- 如果它们有一定相似性，则提供 **1 到 3 之间**的信息单元。

能够捕捉这种连续 “有效计数” 的数学工具叫做**幅度（Magnitude）**。它为每个键分配一个权重 $\mu_j$，表示其独特贡献。

## 工作原理：权重方程

核心机制是一个线性系统，用于求解每个键的 “独特性权重”。

$$Z\boldsymbol{\mu} = \mathbf{1}$$

以下是每个分量的精确含义：

1. $Z$ **（相似度矩阵）**：这是一个 $N \times N$ 的矩阵，其中元素 $Z_{ij}$ 是一个介于 0 和 1 之间的数，衡量键 $i$ 和键 $j$ 之间的相似度。
   - $Z_{ij} = 1$ 表示两个键完全相同。
   - $Z_{ij} \approx 0$ 表示两个键完全不同（正交）。
2. $\boldsymbol{\mu}$ **（未知权重）**：这是一个列向量，包含序列中**每个**键的权重：$\boldsymbol{\mu} = [\mu_1, \mu_2, \ldots, \mu_N]^\top$。在求和公式中，我们用下标 $i$ 表示 “当前键”（即我们正在检查其约束的键），用下标 $j$ 遍历其所有邻居。
3. $\mathbf{1}$ **（单位约束）**：这是一个全 1 向量，作为每一行的目标值。

### 矩阵系统的可视化

为了理解 $i$ 和 $j$ 之间的关系，我们来看 3 个键（$N = 3$）的完整系统：

$$\begin{bmatrix} Z_{11} & Z_{12} & Z_{13} \\ Z_{21} & Z_{22} & Z_{23} \\ Z_{31} & Z_{32} & Z_{33} \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

- **第 1 行（i=1）**：$Z_{11}\mu_1 + Z_{12}\mu_2 + Z_{13}\mu_3 = 1$。第一个键（$i = 1$）必须平衡自身的权重 $\mu_1$ 与邻居 2 和 3 的加权相似度。
- **第 2 行（i=2）**：$Z_{21}\mu_1 + Z_{22}\mu_2 + Z_{23}\mu_3 = 1$。第二个键（$i = 2$）有自己的约束，对所有邻居 $j \in \{1, 2, 3\}$ 求和。

**方程的作用：**

该方程对集合中的每一个键 $i$ 施加严格的约束：

$$\underbrace{\mu_i \cdot Z_{ii}}_{\text{自身贡献}} + \underbrace{\sum_{j \neq i} \mu_j \cdot Z_{ij}}_{\text{来自邻居的贡献}} = 1$$

用通俗的话说："我自身的权重加上所有邻居的加权相似度之和必须恰好等于 1。"

这迫使产生一个权衡：

- **如果一个键没有邻居（$Z_{ij} \approx 0$）**：（请记住，$Z_{ij}$ 是由距离计算得出的相似度，通常为 $e^{-\text{距离}^2}$，因此距离很远意味着 $Z \approx 0$）。第二项消失，方程简化为 $\mu_i \cdot 1 = 1$，因此 $\mu_i$ 必须为 **1**。该键保留完整权重。
- **如果一个键有许多相同的邻居（$Z_{ij} \approx 1$）**：第二项变得非常大，因为许多邻居为求和贡献了值。为了使总和保持等于 1，权重 $\mu$ 必须严格下降。具体来说，如果有 $N$ 个相同的键，它们的权重必须降至 $1/N$，以使其总和保持等于 1。

通过求解这个系统，我们可以精确地推导出每个键相对于整个群组的冗余程度。这些权重的总和 $|X| = \sum \mu_i$ 就是集合的**幅度（Magnitude）**。它告诉我们真正存在多少信息。

## 相似度核的选择

为了使这个系统在实践中可行，我们必须为相似度 $Z_{ij}$ 选择一个具体的公式。我们不能使用任意函数；我们需要一个能保证线性系统 $Z\boldsymbol{\mu} = \mathbf{1}$ 确实可解的函数。

我们使用**高斯核（Gaussian Kernel）**：

$$Z_{ij} = e^{-t \cdot \|x_i - x_j\|^2}$$

这一特定选择至关重要，因为高斯核是**严格正定的（Strictly Positive Definite, SPD）**。

简单来说，这个性质确保相似度矩阵 $Z$ 始终是**可逆的**。如果没有这个性质，即使两个键只是略微相似，数学计算也可能“崩溃”（导致除以零错误或无穷多解）。高斯核保证只要你的词元不是完美的克隆，权重 $\boldsymbol{\mu}$ 就始终存在唯一且稳定的解。

（如果你想深入了解技术细节，可以研究：**“为什么高斯（RBF）核是严格正定的？”**、**“严格正定性如何保证矩阵可逆性？”** 以及 **“用严格正定矩阵求解线性系统的数值稳定性”**。）

有了这个稳定的基础，我们现在可以构建完整的注意力机制。

### 关键性质

| 性质 | 描述 |
|---|---|
| 冗余抑制 | 密集簇中的点获得更低的 $\mu_j$ |
| 独特性放大 | 孤立/边界点获得更高的 $\mu_j$ |
| 信息可加性 | 总信息量是不相关部分的总和 |
| 多样性上限 | $\lvert X\rvert$ 代表绝对最大多样性 |

幅度权重 $\mu_j$ 回答的问题是：**“在给定所有其他点的情况下，点 $j$ 贡献了多少独特的几何信息？”**

## 幅度注意力：构建方法

### 第一步——键空间几何（高斯核）

给定键 $\{k_1, \ldots, k_n\}$，我们构建相似度矩阵 $Z$。与使用固定点积的标准注意力不同，我们使用带有**可学习尺度参数** $t$ 的高斯核：

$$Z_{jl} = \exp\left(-t \cdot \frac{\|k_j - k_l\|^2}{d_k}\right)$$

参数 $t$ 充当**几何分辨率控制器**。通过使 $t$ 可学习，模型可以自主调节其相似度度量的“锐度”：

- **高 $t$ 值**会创建一个严格的过滤器，只有极其接近的向量才被视为冗余。
- **低 $t$ 值**会创建一个更宽泛的过滤器，允许模型抑制并非完全相同但语义相关的“近义词”或语义簇。

本质上，模型学习的是一个最优半径——在这个半径内，两条信息应被视为“相同”，从而将抑制机制的灵敏度调整到特定数据集的最佳状态。

## 第二步——求解权重（迭代法）

为了求解 $\boldsymbol{\mu}$，我们使用**截断共轭梯度法（Truncated Conjugate Gradient, CG）**来求解系统 $Z\boldsymbol{\mu} = \mathbf{1}$。

相比代价高昂的 $O(N^3)$ 矩阵求逆，共轭梯度法仅需 3–5 次迭代即可找到最优权重。这使得复杂度保持在 $O(N^2)$，与标准注意力的开销一致。

当键几乎完全相同时，系统可能在数值上变得不稳定。我们使用**吉洪诺夫正则化（Tikhonov Regularization）**（$Z + \epsilon I$）来确保求解器始终能找到唯一且稳定的解，避免训练过程中梯度“爆炸”。

## 第三步——幅度门控

一旦我们获得了独特性权重 $\boldsymbol{\mu}$，我们在注意力求和之前对值（$V$）施加一个**幅度门控（Magnitude Gate）**。

$$\text{Gate}_j = \sigma(\beta \cdot \mu_j + \gamma)$$

$$\text{Output} = \text{Attn}(Q, K, (\text{Gate} \cdot V))$$

与仅调整注意力分数（只改变概率）的方法不同，这种方法**物理性地衰减**信号质量。通过在求和之前将每个值向量 $V_j$ 乘以门控值，冗余词元被缩放至接近零。它们实际上“缩小”或从隐藏状态中消失，防止其冗余特征在最终输出中累积并压过独特信息。

考虑一个包含 50 个重复词元的簇。它们的幅度权重将为 $\mu_j = 1/50 = 0.02$。如果幅度门控传递了这个权重，则整个簇的贡献变为：

$$\sum_{\text{50 个词元}} \text{Score} \cdot (0.02 \cdot V) = 50 \cdot (\text{Score} \cdot 0.02 \cdot V) = \mathbf{1.0} \cdot \text{Score} \cdot V$$

50 个词元的庞大数量在数学上被压缩，迫使模型将整个簇视为仅 **1 个信息单元**。

可学习的参数（$\beta, \gamma$）允许模型决定冗余抑制的力度。模型学会只在几何独特性权重确实能提高性能的地方“信任”它们。