



東南大學

本科毕业设计（论文）报告

题 目： 基于物理启发的 Transformer 人工
神经网络超参数优化及性能分析

学 号： 10Q21108

姓 名： 蒋海东

学 院： 物理学院

专 业： 物理学类（强基）

指导教师： 张家钧

起止日期： 2023 年 12 月至 2024 年 5 月

东南大学毕业（设计）论文独创性声明

本人声明所呈交的毕业（设计）论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得东南大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

论文作者签名：_____ 日期：____年____月____日

东南大学毕业（设计）论文使用授权声明

东南大学有权保留本人所送交毕业（设计）论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括刊登）论文的全部或部分内容。论文的公布（包括刊登）授权东南大学教务处办理。

论文作者签名：_____ 导师签名：_____
日期：____年____月____日 日期：____年____月____日

摘要

此 L^AT_EX 模板基于东南大学教务处 2024 年 1 月最新 MS Word 版本制作。

摘要内容独立于正文而存在，是论文内容高度概括的简要陈述，应准确、具体、完整地概括论文的主要信息，内容包括研究目的、方法、过程、成果、结论及主要创新之处等，不含图表，不加注释，具有独立性和完整性，一般为 400 字左右。

“摘要”用三号黑体加粗居中，“摘”与“要”之间空 4 个半角空格。摘要正文内容用小四号宋体，固定 1.5 倍行距。

论文的关键词是反映毕业设计（论文）主题内容的名词，一般为 3-5 个，排在摘要正文部分下方。关键词与摘要之间空一行。关键词之间用逗号分开，最后一个关键词后不加标点符号。

关键词：关键词 1，关键词 2，关键词 3，关键词 4

ABSTRACT

English abstract should correspond to the contents in the Chinese abstract. It should describe the thesis contents as a third-person perspective. The basic tense used in the abstract is the present tense. Other tenses like the past tense or the completion tense should only be used when necessary.

ABSTRACT should be centered and bolded with Times New Roman \zihao{3}.

The English abstract contents format is the same as the Chinese abstract. The English “KEY WORDS” should be in all uppercase, and all keywords have the first letter in capital. Keywords are separated by an English comma.

KEY WORDS: Keywords 1, Keywords 2, Keywords 3, Keywords 4

目 录

摘 要	I
ABSTRACT	III
目 录	V
第一章 相关理论技术基础	1
1.1 引言	1
1.2 从物理启发的神经网络	1
1.2.1 人工神经网络的起步	1
1.2.2 霍普菲尔德网络	2
1.2.3 玻尔兹曼机	4
1.2.4 与 Transformer 的联系	5
1.3 Transformer 模型	6
1.3.1 编码器-解码器模块	6
1.3.2 位置嵌入	7
1.3.3 注意力机制	7
1.3.4 多头注意力	8
1.3.5 全连接前馈神经网络	9
1.3.6 残差函数与层归一化	9
1.4 本章小结	9
第二章 参数定义与评估方法	11
2.1 引言	11
2.2 模型训练相关超参数定义	11
2.2.1 训练步数计算	11
2.2.2 预热策略	12
2.2.3 学习率	12
2.3 超参数对模型性能的影响	13
2.3.1 解码器-编码器堆叠层	13
2.3.2 注意力头	13
2.4 翻译评价指标	14
2.4.1 准确率	14
2.4.2 BLEU 分数计算	14

第一章 相关理论技术基础

1.1 引言

本论文研究基于物理启发的 Transformer 人工神经网络。因此本章主要介绍基础理论知识：与物理思想相关的霍普菲尔德网络、玻尔兹曼机，及其它们对神经网络的深远影响；Transformer 模型结构及各个模块具体细节。

1.2 从物理启发的神经网络

人工神经网络诞生虽不足百年，但其发展之迅速，技术之成熟，运用之广泛，是如今科学研究、生产工作、日常生活重要的组成部分。其中有两个重要的物理模型，在人工神经网络停滞不前时，为研究者提供思维模型，突破技术瓶颈。霍菲尔德网络启发于伊辛模型（Ising Model），玻尔兹曼机启发于玻尔兹曼分布（Boltzmann Contribution）。

1.2.1 人工神经网络的起步

人工神经网络的思想最早可追溯至 1943 年，Warren McCulloch 与 Walter Pitts 提出形式化神经元模型（McCulloch-Pitts 模型）^[1]，属于联结主义思想的产物。

用布尔函数 1/0 状态模拟神经元的“兴奋-抑制”行为，因此可以使用数字电路中的逻辑门实现，它接受多个输入，然后产生单一的输出。通过改变神经元的激发阈值，就可完成“与（AND）”、“或（OR）”及“非（NOT）”等三个状态转换功能。

1958 年，Frank Rosenblatt 出一种具有单层计算单元的神经网络，称作感知机（Perceptron）模型，引入基于误差的权值更新机制，实现了从数据中学习。这一模型在图像识别与语音处理等早期任务中取得初步成功，成为早期神经网络研究的重要里程碑^[2]。

然而，Marvin Minsky 和 Seymour Papert 在其 1969 年出版的作品《Perceptrons》中指出，感知机无法表示异或（XOR）等非线性可分问题，且缺乏构建深层网络的理论基础^[3]。这一观点在当时极具影响力，导致神经网络研究在 1970 年代一度停滞。

1982 年，Hopfield 提出 Hopfield 网络（Hopfield Network），将神经网络建模为一个具有物理能量函数的动力学系统，引入哈密顿量的概念，使网络状态朝向能量极小值演化，从而实现稳定的联想记忆。这一模型首次将统计物理中的能量极小化思想引入神经计算，开创了神经网络研究的新方向^[4]。

随后，Hinton 与 Sejnowski 提出了玻尔兹曼机（Boltzmann Machine），在 Hopfield 网络的基础上引入概率采样机制与玻尔兹曼分布，使模型能够学习复杂的数据分布，成为生成模型研究的重要起点^[5]。

1986 年，Rumelhart、Hinton 与 Williams 提出反向传播（Backpropagation）算法，使多

层感知机的训练成为可能，彻底改变了神经网络的可训练性，促进了深度学习的崛起^[6]。

此后发展出的卷积神经网络（CNN）、循环神经网络（RNN）等架构，在设计思想与优化机制上，都受到了 Hopfield 网络的深刻影响。

1.2.2 霍普菲尔德网络

设 Λ 是一个 d 维晶格上的格点集合，每个格点 $k \in \Lambda$ 上定义一个离散自旋变量 $\sigma_k \in \{-1, +1\}$ ，构成整个系统的自旋构型 $\sigma = \{\sigma_k\}_{k \in \Lambda}$ 。

对于每对最近邻格点 i 和 j ，存在相互作用常数 J_{ij} ，不考虑外部磁场的作用。表示系统能量的哈密顿量为：

$$H(\sigma) = - \sum_{\langle ij \rangle} J_{ij} \sigma_i \sigma_j - \mu \sum_j h_j \sigma_j \quad (1.1)$$

其中， μ 表示磁矩， $\langle ij \rangle$ 表示对所有最近邻格点对求和。 σ_i 和 σ_j 表示单个原子磁矩参数，只能为 +1 或 -1，分别指自旋向上和自旋向下。

在一维模型中，并不能证明相变的发生，但在二维、三维系统中已被证明可以发生自发磁化。系统的相变过程趋向于使 $H(\sigma)$ 最小，从而达到热力学稳定态。

对于一个拥有 L 个晶格点的模型，自旋配置有 2^L 种可能，这使得伊辛模型在数值计算中非常困难。

Metropolis-Hastings 算法是统计物理中一种蒙特卡洛算法，用于计算伊辛模型的数值解。该算法的特点是，系统处于状态 V ，并且会有一定的概率跳到 μ 。如果不发生状态变化，则保持当前状态，直到达到某些条件或者热平衡，例如伊辛模型被完全磁化。

伊辛模型及其研究为 Hopfield 网络的诞生提供了灵感。John Hopfield 在 1982 年提出的 Hopfield 网络，将神经网络抽象为一个离散动力学系统^[4]。其能量函数定义为：

$$E_{\text{Hopfield}} = -\frac{1}{2} \sum_{i \neq j} T_{ij} V_i V_j \quad (1.2)$$

其中 V_i 表示第 i 个神经元的状态，为 0 表示“抑制”，为 1 表示“兴奋” T_{ij} 为第 i 与第 j 个神经元之间的连接权重，其表达式如下。

在 Hopfield 网络中，连接权重 T_{ij} 的定义为：

$$T_{ij} = \sum_s (2V_i - 1)(2V_j - 1) \quad (1.3)$$

当 $V_i = V_j$ 时， $T_{ij} = 1$ ；当 $V_i \neq V_j$ 时， $T_{ij} = -1$ 。这意味着：若两个神经元状态相反，则连接权重为负；若两个神经元状态相同，则连接权重为正。满足 $T_{ij} = T_{ji}$ 、 $T_{ii} = 0$ ，表示连接的方向不改变连接权重，自己不与自己连接。

假设网络中存在 N 个神经元，且权重矩阵已知。现在将某一个神经元状态 V_s 翻转，其余 $N - 1$ 个神经元状态保持不变。根据如下规则对神经元状态进行更新：

$$V_i = \begin{cases} 1, & \text{if } \sum_{j \neq i} T_{ij} V_j > 0 \\ 0, & \text{if } \sum_{j \neq i} T_{ij} V_j < 0 \end{cases} \quad (1.4)$$

可以证明，经过该规则计算后， V_i 会恢复到原来的状态，而其他神经元的状态不会受到该翻转的干扰。若将每个神经元视为图像中的一个像素点，那么这一过程就类似于将一张残缺的图像输入网络后，输出为恢复完整的图像。这种特性表明 Hopfield 网络具备记忆功能。在后续的研究中，发现网络的记忆能力的数量级为 $K^{\max} \approx 0.14N$ ， N 为神经元数量^[7]。

定义该网络的能量函数为：

$$E = -\frac{1}{2} \sum_i \sum_j w_{ij} V_i V_j = -\frac{1}{2} \mathbf{V}^T \mathbf{W} \mathbf{V} \quad (1.5)$$

该能量函数在形式上与量子力学中的能量计算公式具有相似性：

$$\langle E \rangle = \langle \psi | \hat{H} | \psi \rangle \quad (1.6)$$

其中 \hat{H} 是系统的哈密顿量， $|\psi\rangle$ 是系统的波函数，描述系统所处的状态。 \mathbf{V} 包含所有神经元的状态，而 $|\psi\rangle$ 则描述系统中粒子的全部信息。两者在结构上的相似性进一步说明了 Hopfield 网络与物理思想（尤其是统计物理与量子力学）之间的深刻联系。

根据上述能量函数的定义，每当根据更新规则发生状态变化时，能量的变化量为：

$$\Delta E = -\Delta V_i \sum_{j \neq i} T_{ij} V_j \quad (1.7)$$

可以证明 ΔE 始终为负值（见附录证明），即每当 V_i 更新时，系统总能量都会减少。因此，状态的演化过程会使能量下降，直到达到某一局部最低值^[4]。

这种机制与物理中的伊辛模型（Ising Model）是同构的。权重 T_{ij} 类似于交换耦合系数。当 T_{ij} 是对称但具备随机特征（例如自旋玻璃）时，系统中会出现多个局部稳定态^[8]。

通过构造一个与 Ising 模型形式等价的能量函数，Hopfield 网络将神经元状态的更新过程视为能量递减的演化过程，使网络最终收敛到某个局部极小值。这种迁移不仅在数学结构上体现出一致性，更在功能上赋予了神经网络以“记忆存储”的能力：通过设计权重矩阵使多个已知模式成为网络的稳定状态，Hopfield 网络就可以将信息“编码”为能量地形中的极小值，实现联想记忆和容错恢复功能。简言之，Hopfield 网络正是借助 Ising 模型

中“能量极小值对应稳定态”的思想，将复杂的神经信息处理问题转化为一个能量优化过程，从而构建了早期神经网络理论的坚实基础。

Hopfield 网络将网络行为统一到一个“能量函数”框架下，通过设计合适的能量景观来实现特定功能。这一思想直接催生了一大类能量模型（Energy-Based Models），包括玻尔兹曼机（Boltzmann Machine）、受限玻尔兹曼机（RBM）、深度置信网络（DBN）以及后来的生成对抗网络（GAN）和扩散模型（Diffusion Models），都可以看作是在不同层面上对“用能量函数刻画数据分布并通过最小化采样来进行学习”这一思路的延续。

1.2.3 玻尔兹曼机

玻尔兹曼机（Boltzmann Machine）也是一种基于统计物理思想构建的神经网络模型，Geoffrey Hinton、Terry Sejnowski 和 David Ackley 等人在 1985 年提出，其建模核心来自热力学体系中的玻尔兹曼分布。在热力学中，系统各个微观状态的概率与其能量成负指数关系，温度作为控制参数调节系统对高能量状态的接受程度^[9]。这一思想被引入神经网络的构建中，使得神经元的激活状态可以类比为物理粒子的状态，整个网络的状态组合对应一个能量值，网络在演化过程中趋向于能量较低的状态，从而自然地编码出输入数据中的约束与结构。

玻尔兹曼机由一组二值随机单元组成，按功能分为可见单元和隐藏单元。单元之间通过带权重的双向对称连接相互作用，网络整体构成一个无向图结构。在任意时刻，网络状态由所有单元的激活状态确定，对应一个总能量，记为 $E(\mathbf{v}, \mathbf{h})$ ，其中 \mathbf{v} 表示可见层状态， \mathbf{h} 表示隐藏层状态。能量函数形式如下：

$$E(\mathbf{v}, \mathbf{h}) = - \sum_{i < j} w_{ij} s_i s_j - \sum_i b_i s_i \quad (1.8)$$

其中 $s_i \in \{0, 1\}$ 为第 i 个单元的状态， w_{ij} 为连接权重， b_i 为偏置项。系统状态的概率服从玻尔兹曼分布：

$$P(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \exp(-E(\mathbf{v}, \mathbf{h})) \quad (1.9)$$

其中 $Z = \sum_{\mathbf{v}, \mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}))$ 是配分函数，用于对所有可能状态的概率进行归一化。该模型由此成为一个定义明确的概率生成模型，可以用来模拟训练数据的分布结构。

网络的学习过程即为对权重 w_{ij} 和偏置 b_i 的调整，使得模型生成的分布尽可能接近训练数据的真实分布。训练目标是最小化模型分布与数据分布之间的 Kullback-Leibler 散度（KL 散度），这一目标函数可通过梯度下降方法进行优化。具体的权重更新规则为：

$$\Delta w_{ij} = \eta (\langle s_i s_j \rangle_{\text{data}} - \langle s_i s_j \rangle_{\text{model}}) \quad (1.10)$$

其中第一项表示在训练数据输入时（即可见单元被“钳制”）时，单位 i 与 j 同时激活的期望值，第二项为网络自由运行下该期望值的估计。这种更新机制只依赖于局部信息，即连接的两个单元的行为，但优化目标是整个网络生成分布的全局特性^[10]。

为了避免陷入局部最小能量态，玻尔兹曼机引入温度作为控制变量，使用模拟退火（simulated annealing）方式进行采样。在高温下，系统状态变换允许较大能量跃迁，搜索空间广；随着温度下降，系统逐渐收敛至较优解^[11]。这一机制与物理系统退火过程完全对应，为网络提供了一种有效的全局优化策略。

玻尔兹曼机可以用于表示和学习复杂概率结构，尤其适合处理具有多个弱约束的任务场景。例如在视觉感知、模式完成和缺失数据补全等任务中，网络能够在输入部分已知的条件下推断出最有可能的整体结构。这种推断能力来自于网络在训练过程中构建出的内部生成模型，能够自动编码高阶特征和潜在变量。

实际训练中，为提高训练效率和稳定性，常采用一种变种形式——限制玻尔兹曼机（Restricted Boltzmann Machine, RBM），其结构中仅允许可见层与隐藏层之间存在连接，而层内不允许连接。RBM 在结构上简化了训练过程，使得条件概率可以独立计算，从而显著提升了采样与学习效率。这一结构随后被用于构建深度信念网络（Deep Belief Network, DBN），成为无监督预训练策略的关键组成部分，对后续深度学习技术的发展产生了重要推动作用^[12]。

玻尔兹曼机的提出标志着能量函数、统计物理与神经网络的结合，也为生成模型的发展奠定了理论基础。其将复杂的数据分布建模问题转化为可解释的能量优化过程，并通过局部更新规则实现全局学习目标，为多层神经网络的理论发展和实践应用提供了重要模型范式。^[9]

1.2.4 与 Transformer 的联系

经典的霍普菲尔德网络被后续研究者改进，引入一个多项式相互作用函数^[7]，并断言该模型具备更强的记忆储存能力。随后有研究者数学证明了这一断言，并得出该模型的存储容量随神经元数量呈指数级增长的结论^[13]。

基于改进版网络，形成的现代霍普菲尔德网络。研究表明，Transformer 模型中的注意力机制就是现代 Hopfield 网络的更新规则，意味着 Transformer 的注意力头可等价为现代 Hopfield 网络的状态更新过程。

现代霍普菲尔德网络（Modern Hopfield Networks）可集成到深度学习架构，其神经元

状态具有连续性，参数具有可微性，因此能通过反向传播进行端到端训练；网络通过单次更新（one update）完成模式检索，与深度学习层的单次激活特性兼容。因为，现代霍普菲尔德网络可作为深度神经网络中的专用记忆层，赋予网络动态存储和调用信息的能力。

1.3 Transformer 模型

由 Vaswani et al. 等人提出的 Transformer 模型完全基于注意力机制，没有任何卷积层或循环神经网络层，是一个自回归模型。

同众多优秀的序列到序列（sequence-to-sequence, seq2seq）模型一样，Transformer 使用编码器-解码器（Encoder-Decoder）结构，如图 1-1 所示。

在模型设计之初，就是应用于语言翻译任务，这也是本文的机器学习任务。输入输出的过程，就是将输入序列嵌入至预训练的向量空间，找到输入向量与上下文向量的关联度向量，两向量之和指向新的向量就是输出序列^[14]。下面将讲解一个标准的 Transformer 模型，其各个参数，并非本次实验所用参数，具体见 ??。

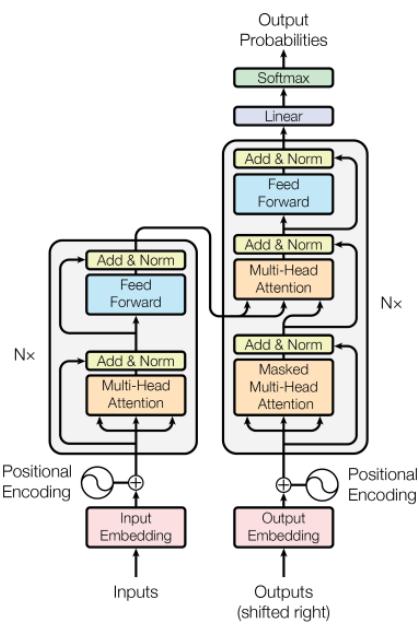


图 1-1 Transformer 模型结构图

1.3.1 编码器-解码器模块

编码器（Encoder）由 $N = 6$ 个相同的层构成，每个层包含两个子层。第一个子层是一个多头注意力（Multi-Head Attention）机制，第二个子层是一个前馈神经网络（Feed Forward Network, FFN）。在每个子层之后，添加了残差连接（Residual Connection）并进行层正则化（Layer Normalization）。整个模型中的输入、输出以及各子层的中间输出都是维度为 $d_{model} = 512$ 的向量。

解码器 (Decoder) 同样由 $N = 6$ 个相同的层堆叠组成。解码器的输入为模型在上一个时间步的输出。每个解码器层除了包含与编码器相同的两个子层外，还增加了一个掩码自注意力机制 (Masked Self-Attention)。为了保证自回归特性，Transformer 在解码器中使用了因果掩码，使得对于输出序列 h_t ，每一步的预测仅依赖于先前的 h_{t-1} 及更早的状态。

1.3.2 位置嵌入

当一个序列 $\mathbf{Z} \in \mathbb{R}^{n \times d}$ 输入 Transformer 模型时，模型首先通过一个线性变换将该序列嵌入到固定维度的向量空间中。具体而言，输入序列与一个共享的词嵌入权重矩阵相乘得到词元嵌入向量。该嵌入权重矩阵在编码器和解码器中是共享的，即输入和输出共用同一组权重，这种做法被称为**权重绑定 (weight tying)**。权重绑定可以有效减少模型参数数量，通常可以将参数数量至少减少一半，同时不损害模型性能。

在获得词嵌入向量后，Transformer 模型通过加入**位置编码 (Position Encoding)** 将序列中各词元的位置信息注入模型表示中，以弥补模型缺乏显式顺序建模能力的缺陷^[14]。位置编码向量与词元嵌入向量逐元素相加，其每个维度由一组不同频率的正余弦函数计算而成，具体公式如下：

$$\text{PE}_{\text{pos}, 2i} = \sin\left(\frac{\text{pos}}{10000^{\frac{2i}{d_{\text{model}}}}}\right), \quad \text{PE}_{\text{pos}, 2i+1} = \cos\left(\frac{\text{pos}}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \quad (1.11)$$

其中， pos 表示位置索引， i 表示嵌入维度索引 ($i \in [0, d_{\text{model}}/2 - 1]$ ， $d_{\text{model}} = 512$)， $\text{PE}_{\text{pos}, k}$ 表示位置编码在第 k 维的分量。

这种设计使得位置编码在任意位置偏移 k 情况下满足如下线性可变性： $\text{PE}_{\text{pos}+k}$ 可表示为 PE_{pos} 的线性函数，从而便于模型捕捉相对位置信息。正弦曲线的波长组成一个几何序列，从 2π 到 $10000 \cdot 2\pi$ ，确保不同位置具有独特且连续变化的编码方式^[15]。

在传统的 RNN 模型中，输入序列（比如一个句子）里的各个 token（对应一个词）按它们在序列中的顺序一次处理，每个时间步 RNN 处理一个词元^[16]。同时，上一个时间步会输出一个向量，与当前词元一并输入。可以认为，状态向量隐含了输入序列中的位置信息。

当序列输入注意力层时，序列中所有的词元是并行输入的，如不提供位置信息，序列里的相同的词元对注意力层来说就不会有语法和语义上的差别，它们会产生相同的输出。

1.3.3 注意力机制

注意力机制 (Attention Mechanism) 最早由 Bahdanau 等人于 2014 年提出^[17]，Transformer 模型中的注意力机制为“缩放版的点积注意力”，具体公式如 (1.15) 所示。序列的词元向量组成矩阵，分别由三个可训练的权重矩阵，线性变换得到矩阵 \mathbf{Q} 、 \mathbf{K} 、 \mathbf{V} 。设一层有 h

个头，每个注意力头的维度为：

$$d_h = \frac{d_{\text{model}}}{h}. \quad (1.12)$$

投影矩阵为：

$$W_j^Q, W_j^K, W_j^V \in \mathbb{R}^{d_{\text{model}} \times d_k} \quad (1.13)$$

对输入 $Z \in \mathbb{R}^{n \times d_{\text{model}}}$ ，计算：

$$Q_j = ZW_j^Q, \quad K_j = ZW_j^K, \quad V_j = ZW_j^V \quad (1.14)$$

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \quad (1.15)$$

其中 \mathbf{Q} 为查询矩阵 (Query Matrix)， \mathbf{K} 为键矩阵 (Key Matrix)，维度均为 d_k 。 $\mathbf{Q}\mathbf{K}^\top$ 用于计算相似度。 \mathbf{V} 为值矩阵 (Value Matrix)，维度 d_v 。实际上，权重矩阵的维度都由注意力的维度决定，存在关系 $d_h=d_k=d_V$ 。 $\frac{1}{\sqrt{d_k}}$ 为缩放因子，作用是保证注意力层输出不会过大。 $\mathbf{Q}\mathbf{K}^\top$ 用于计算相似度，矩阵元素为一个词元相对于另一个词元关联度。 softmax 函数作用于矩阵的每一行，将其元素映射到 $[0, 1]$ 之间的概率分布。这样更加突出单词之间的关联性，并且使得模型更加关注重要的语义信息。

1.3.4 多头注意力

多头注意力是将 $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ 投影到低维空间，每一个低维空间，就是一个注意力头。对于输入 $Z \in \mathbb{R}^{n \times d}$ ，使用不同的线性变换得到：

$$\text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \quad \text{for } i = 1, \dots, h \quad (1.16)$$

其中 $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{n \times d_k}$ 是每个头独立的参数， $d_k = \frac{d_{\text{model}}}{h}$ 。每个头执行一次注意力计算 (1.15)，然后对多头的输出进行拼接并做一次线性变换：

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (1.17)$$

多头注意力使模型在不同子空间中并行地关注输入的不同方面或关系模式，从而增强理解能力和表达能力。多头注意可以有效地提高模型能力：具有 8 个头的模型，与相同尺寸的模型相比，BLEU 分数提升一分。对 Transformer 模型各层中的注意力头数进行修剪，编码器-解码器注意力层的注意力头数最不容易被修剪的，即它们在被修剪的过程中最晚被移除^{2019VoitaPrune}，侧面说明了多头注意力的重要性。

1.3.5 全连接前馈神经网络

前馈神经网络是一个两层的全连接层，第一层的激活函数为 Relu，第二层不使用激活函数，其表达式如下：

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (1.18)$$

虽然线性变换在不同位置上是相同的，但它们在每一层之间使用不同的参数。该网络的输入输出都为 $d_{\text{model}} = 512$ ，中间层 $d_{\text{ff}} = 2048^{[14]}$ 。FFN 在模型中增加了非线性变换计算，增强模型的表达能力。

1.3.6 残差函数与层归一化

每个子层输出如下：

$$\text{layernorm}(h + \text{sublayer}(h)) \quad (1.19)$$

其中，函数符 $\text{layernorm}(\bullet)$ 表示层归一化运算， $\text{sublayer}(\bullet)$ 表示前馈子层计算结果， $h \in \mathbb{R}^d$ 表示上层输入的隐藏状态向量

残差连接在 Transformer 架构中不仅有助于缓解梯度消失问题，还在信息流动、模型稳定性和性能提升方面发挥着重要作用。通过对残差连接的结构和机制进行改进，可以进一步增强 Transformer 模型的表达能力和训练效率^[18]。层归一化对于稳定递归网络的隐态动态是非常有效的，实验表明层归一化可以大大减少训练时间^[19]。

1.4 本章小结

本章主要介绍人工神经网络如何从物理思想启发，然后介绍 Transformer 模型的具体结构。首先介绍伊森模型和霍普菲尔德网络的关系，两者对系统粒子状态的描述方式、能量函数形式以及遵循最低能量原则都体现相似性。；然后介绍玻尔兹曼机，通过模拟玻尔兹曼分布形式，实现神经元的激活，学习数据包含信息概率分布，以能量最小化的形式输出结果；最后详细介绍了 Transoformer 结构及各个模块。

第二章 参数定义与评估方法

2.1 引言

人工神经网络经过训练获得的权重（Weights）、偏置（Bias）等数据被称为参数（Parameter），无法提前预知。而决定模型结构、训练进程、梯度更新幅度等指标被称为超参数（Hyperparameter），这决定了模型的训练效率和训练结果。准确率（Accuracy）常作为标签分类任务的指标，而评价神经机器翻译的指标是 BLUE 分数。因此，本章首先严格定义了一系列超参数：训练步数、批量大小、训练轮次、预热步数、层数、注意力头数，为后续实验论述做铺垫。然后介绍训练步数、学习率计算方法，以及几个超参数对模型性能的影响。最后介绍 BLUE 分数的计算方法。

2.2 模型训练相关超参数定义

本节对常用的参数进行申明与定义，在后续研究中会继续使用这些名称。将在后续小节对重点参数的作用做进一步阐释。

训练步数（Training Steps） 表示模型的迭代次数，即优化器（Optimizer）更新参数的总次数。本文使用 `train step` 作为符号表示。

批量大小（Batch Size） 定义为在单张 GPU 上，一个训练步骤中同时处理的样本数量。在序列到序列任务（如中英翻译任务）中，一个样本通常是一个中英文句子对。本文符号表示为 `batch size`。

训练轮次（Train Epoch） 表示模型完整遍历一次训练集的次数。本文使用 `epoch` 作为符号表示。

学习率（Learning Rate） 表示在每个训练步中，模型参数沿梯度方向更新的幅度，是损失函数梯度乘积的缩放因子。本文符号表示为 `lr`。

预热步数（Warmup Steps） 指训练初期学习率随训练步数逐渐线性增长的最大步数。本文符号表示为 `warmup`。

层数（Layers） 表示 Transformer 模型中编码器和解码器的堆叠层数，详见 1.3.1。本文使用 `N` 作为参数表示。

注意力头数（Multi-Head Attention） 表示每层中并行计算的注意力头的数量。本文符号表示为 `h`。

2.2.1 训练步数计算

根据上面的定义，可以得出训练步数计算公式如下：

$$N_{\text{step}} = \left\lceil \frac{N_{\text{pairs}}}{B} \right\rceil \times E \quad (2.1)$$

其中 N 为训练集句子对总数, B 为批大小, E 为训练轮次数, $\lceil \cdot \rceil$ 为向上取整运算符。

训练步数与训练轮数作为等效评估指标, 均可有效监测模型收敛状态。基于训练步数计算的“步每小时”(steps per hour)可精准量化模型运算效率, 该指标在预实验阶段具有关键指导价值, 尤其在时间预算受限或计算资源不足的场景下。

2.2.2 预热策略

预热学习调度策略是与 Transformer 模型一同提出的^[14], 又称为 Noam-style 的预热加衰减调度。该策略在训练初期将学习率从零线性上升至某一峰值, 再按照步数反比规律衰减。这种做法有效缓解了模型训练初期的不稳定现象。

Transformer 模型中残差结构和层归一化的位置决定了预热步数的必要性。具体而言, 标准结构采用 Post-LN (层归一化在残差之后) 设计, 网络中靠近输出层的参数初始期望偏大, 导致在训练初期使用较大学习率时出现训练不稳定、梯度爆炸等问题^[20]。为此, 合理设置 warmup 步数成为保证训练稳定的重要手段。

Xiong 等人^[20]进一步提出将 LayerNorm 前移 (即 Pre-LN 架构), 能够有效避免上述问题。在 Pre-LN 框架下, 即使预热步数设置为 0 (warmup=0), 也能实现更快的收敛速度和更好的收敛效果。Kaplan 等人在研究大规模语言模型缩放规律时指出, 预热步数应与模型规模、batch size、总训练步数等参数协同设定^[21]。一般经验为设置为总训练步数的 2% ~ 6%, 以确保学习率平滑过渡。

因此, warmup 步数不仅仅是训练稳定性优化的一部分, 也与最终模型性能密切相关。本文将在后续章节深入研究 warmup 步数对模型收敛速度与翻译质量的影响。

2.2.3 学习率

学习率是推动模型前进的关键因子, 决定模型能否收敛, 合适的学习率能让训练事半功倍。

在标准梯度下降 (Gradient Descent) 中, 学习率 η 作为缩放因子, 控制参数沿梯度方向更新的幅度。其基本更新公式为:

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_{\theta} \mathcal{L}(\theta_t) \quad (2.2)$$

其中 θ_t 为第 t 步的模型参数, η 为学习率 (即 lr); 为 $\nabla_{\theta} \mathcal{L}(\theta_t)$ 为损失函数 \mathcal{L} 对参数的梯度。该公式表示每一次参数更新, 模型将沿负梯度方向前进, 步长由学习率 η 决定。如果 η 太大, 可能导致震荡或发散; 若太小, 则收敛缓慢。指出学习率是影响网络训练速度和稳定性的最关键参数之一。

讨论了学习率过大导致发散、过小导致收敛慢的风险

在标准的 Transformer 模型中，学习率遵循预热策略，不再是固定值，而在训练中动态调整，其定义如下：

$$\text{lr}(\text{step}) = \text{factor} \cdot d_{\text{model}}^{-0.5} \cdot \min(\text{step}^{-0.5}, \text{step} \cdot \text{warmup}^{-1.5}) \quad (2.3)$$

其中 $\text{factor} = 1$, $\text{warmup} = 2000$, d_{model} 为模型的嵌入维度。学习率预热阶段线性增长，之后随 $1/\sqrt{\text{step}}$ 衰减。

2.3 超参数对模型性能的影响

2.3.1 解码器-编码器堆叠层

编码器-解码器堆叠层数是 Transformer 模型最核心的结构参数之一，决定了模型的表示能力和学习复杂模式的能力。理论上，更多的层可以使模型具备更强的特征抽象与长距离依赖建模能力，因此在大型语言建模、机器翻译等任务中表现出显著优势^[14,21]。

然而，实际研究表明 Transformer 的层数存在大量冗余。在一定任务场景下，模型并不需要特别深的结构即可达到较高性能。例如，Michel 等人^[22] 在注意力裁剪实验中发现，即便裁剪掉 20% ~ 40% 的注意力头，模型性能也几乎不受影响，说明深层注意力存在冗余。

进一步研究表明，通过知识蒸馏（knowledge distillation）得到的轻量级模型在层数更少的条件下依然可以逼近大型模型的性能。例如，MobileBERT^[23] 将 BERT-large 蒸馏至只有 1/4 层数的结构，依然在多个 NLP 任务上表现良好。这说明并非层数越多越好，过多的堆叠反而可能引入训练难度、过拟合风险以及资源消耗。

此外，Transformer 层数对性能的影响往往是非线性的。在某一层数之前，增加层数能带来性能提升；但超过临界点后，性能反而可能下降甚至急剧退化^[20,24]。因此，探索合适的层数配置，是提升模型性能与效率的关键路径之一。

2.3.2 注意力头

注意力头数是 Transformer 架构核心创新之一，其设置直接影响模型的表达能力、计算效率以及泛化性能。合理选择注意力头数不仅关系到是否能够充分建模输入之间的复杂依赖关系，还涉及到训练稳定性与资源利用效率的权衡。因此，头数是挖掘 Transformer 模型潜力的关键超参数之一。

然而，注意力头数并非越多越好。在模型尺寸较小时，例如当嵌入向量维度仅为 $d_{\text{model}} = 128$ 时，若设置头数为 $h = 8$ ，根据公式 (1.12) 每个注意力头的维度仅为 $d_h = 16$ ，表达能力受到限制，训练过程可能难以收敛。

此外，在计算机视觉任务中，已有研究表明，与其使用大量注意力头，不如减少头数使用并堆叠更多的编码器层来获得更强的建模能力。在特定设置下，两者甚至能够实现相近的训练效果^[25]。

进一步地，已有研究发现，并非所有注意力头都是必要的。一些注意力头在训练后未能有效参与特征建模。通过逐层剪枝（Layer-wise Pruning），在保持性能稳定的情况下，移除冗余的注意力头，从而显著减少计算开销^[26]。

2.4 翻译评价指标

2.4.1 准确率

准确率（Accuracy）的计算方法非常见，公式如下：

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \quad (2.4)$$

这个计算方式强调结果的完全匹配，适用于标签分类，因为标签的正确与否具有二值性。而神经机器翻译中，输出序列长度可变，存在多种译法都是有效的，因此准确率会严重低估模型性能。

2.4.2 BLEU 分数计算

用人工评价机器翻译虽然准确，但需要大量人力成本，所需评估时间可以长达数月，而且无法重复使用。Kishore Papineni 等人在 2002 年提出 BLEU（Bilingual Evaluation Understudy）指标^[27]，这是一种自动评估机器翻译质量的指标，用于衡量机器翻译输出与一个或多个参考翻译之间的相似度。

BLEU 计算中，一个重要参数是 n -gram。 n -gram 是指在一个文段中，连续的 n 个字组成的块。例如，1-gram 是一个字，2-gram 是连续的两个字，3-gram 是连续的三个字，依此类推。将候选翻译（Candidate，即机器翻译结果）中的 n -gram 出现次数，与参考翻译（Reference）对应 n -gram 进行比较，可以得到 BLEU 分数。首先计算的是改进版精度分数（Modified Precision Score），计算公式如下：

$$p_n = \frac{\sum_{C \in \{\text{Candidates}\}} \sum_{\text{n-gram} \in C} \text{Count}_{\text{clip}}(\text{n-gram})}{\sum_{C' \in \{\text{Candidates}\}} \sum_{\text{n-gram}' \in C'} \text{Count}_{\text{clip}}(\text{n-gram}') \quad (2.5)}$$

公式理解为将与参考翻译 n -gram 一致的片段数量求和，再除以候选翻译 n -gram 总数。

示例 1. Reference: 这是本很精彩的书。

Candidate: 这本书非常精彩。

此时 1-gram 分数为 $p_1 = \frac{5}{7}$ ，显然该实例的准确率为 0。但机器翻译有时会过度生成合理的词语，比如模型出现了过拟合、失函数停在鞍点、神经网络层数较浅等问题。此时需要加上限制规则， $\text{Count}_{\text{clip}}(\text{n-gram}) = \min(\text{Count}, \text{Max_Ref_Count})$ 对于一特定 n -gram，

计算参考翻译中的最大值，将该最大值与候选翻译中的 n -gram 数量进行比较，取两者最小值参与计算。

示例 2. Reference: 你 们 有 小 孩 吗?

Candidate: 你 孩 孩 孩 孩 孩?

此时的 1-gram 分数为 $p_1 = \frac{2}{6}$ 。

机器翻译会出现另一个问题，生成的句子长短与参考翻译长度不匹配。这个超长的候选翻译会包含多个参考翻译的词语，在 1-gram 时会表现更佳。这就像在猜测答案，将所有可能的结果放进来，但语义并不通顺。但是，不同 n 值的 n -gram 很好地惩罚了这一现象，超出的词语会使更长的 n -gram 表现不佳，BLEU 分数出现下降对于过短的翻译句子，引入简短惩罚（Brevity Penalty, BP）因子，计算公式如下：

$$BP = \begin{cases} 1 & \text{if } c > r \\ \exp\left(1 - \frac{r}{c}\right) & \text{if } c \leq r \end{cases} \quad (2.6)$$

其中 c 是候选翻译的总长度（总词语数）， r 是参考翻译的（总词语数）。(2.6) 保证超长的句子不会受到惩罚，而越短的句子，得到的惩罚越重。如果时取翻译句子的平均总长度作为最佳匹配长度，会过度惩罚长度较差的句子。

至此，所有规则都已具备，BLEU 分数计算如下：

$$\text{BLEU} = BP \times \exp\left(\sum_{n=1}^N w_n \cdot \log(\text{precision}_n)\right) \quad (2.7)$$

其中 N 是最大 n -gram 长度（通常为 4）； w_n 是每个 n -gram 精度的权重，通常取 $w_n = \frac{1}{N}$ ； precision_n 是第 n -gram 的精度； BP 是简短惩罚因子。

取对数能够平滑不同精度值之间的差异，当高阶与低阶 n -gram 分布差异较大时。

一方面，一个高阶的 n -gram 的精度为零，取对数能缓冲这影响，避免这些高阶 n -gram 直接将整体 BLEU 分数拉低到零。

另一方面，低阶 n -gram 强调翻译用词的匹配性，而高阶 n -gram 更能够捕捉翻译的语法和流畅性。通过对精度取对数，可以避免低阶 n -gram 过度贡献 BLEU 分数的，并增加高阶 n -gram 对评分的贡献。

计算 p_n 的值，可以检测结果用词准确；计算不同 n 值的 n -gram 能反应候选翻译生成的顺序，检测用词语序；使用 BP 因子，保证了翻译长度的匹配性；对数计算与权重分分配，综合考虑各项影响。多维度的评价、合理的量化和对算力低要求，使得 BLEU 分数更

接近人工判断。BLEU 自 2002 年提出以来，成为神经机器翻译的主流评价方法。时至今日，几乎所有的机器翻译任务，一定会计算 BLEU 分数，衡量模型性能。这对中文的语义准确性评估相当重要。