

学校代码: 10284

分类号: TP183

密级: 公开

UDC: 004.8

学号: DZ1833001



南京大學

博士学位论文

论文题目 面向人工神经网络的

新型神经元模型研究

作者姓名 安俊逸

专业名称 计算机科学与技术

研究方向 人工智能

导师姓名 申富饶 教授

2024年5月28日

答辩委员会主席	杨明教授
评阅人	胡鹤轩 教授
	魏秀参 教授
	宋方敏 教授
	武港山 教授
	路通 教授

论文答辩日期 2024年5月23日

研究生签名:

导师签名:

Research on New Neuron Model for Artificial Neural Network

by
Junyi An

Supervised by
Professor Furao Shen

A dissertation submitted to
the graduate school of Nanjing University
in partial fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Computer Science and Technology



Department of Computer Science and Technology
Nanjing University

May 28, 2024

南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目： 面向人工神经网络的新型神经元模型研究

计算机科学与技术 专业 2018 级博士生姓名： 安俊逸

指导教师（姓名、职称）： 申富饶 教授

摘 要

神经网络是机器学习中的重要技术。目前，神经网络在很多重大挑战中表现出卓越的成绩，其应用也覆盖了人类社会的方方面面。从模型设计的角度出发，神经网络的研究可以分为两类：1. 针对网络架构的研究；2. 针对神经元模型的研究。在过去的几十年中，各种各样的网络架构相继被提出，但针对神经元模型的研究相对较少。目前，大部分的神经网络依然沿用上个世纪40年代提出的MP (McCulloch-Pitts) 神经元模型。尽管在神经网络发展史上也出现过一些具有重大意义的神经元模型，但大部分模型都无法广泛应用于当下主流的网络架构。

本文对新型神经元模型进行探索和设计，致力于构造通用，易扩展，易训练，表达能力强的神经元模型。此外，本文也考虑了主流网络架构与新型神经元模型的结合，针对不同网络架构以及对应下游任务的特性，进一步探索神经元模型的高效使用方法。总结来说，本文所涉及的主要工作包括：

1. 一种基于物理弹性碰撞模型的通用神经元模型：IC 神经元 (Inter-layer Collision Neuron)。在物理一维弹性碰撞模型中，速率，动量等物理量在碰撞前后会发生确定形式的变换，本文受启迪于碰撞模型，设计出一种通用模型(IC 神经元)。不同于传统的生物启发神经元，IC 神经元具有较强的物理合理性和数理解释性。本文展示了IC 神经元可以拟合比MP 神经元更复杂的非线性分布。此外，IC 神经元保留了MP 神经元易集成，易训练，轻量性的特点，能够用于各种网络架构中。实验结果表明，IC 神经元可以应用于各种神经网络，并在多个任务中取得了优于MP 神经网络的泛化性能。此外，IC 神经网络保持了和MP 神经网络一致的计算消耗，具有重要的应用价值。

2. 一种基于IC 神经元的高性能卷积神经网络算法：IC-CNN+。卷积神经网络 (Convolutional Neural Network, CNN) 通常用于处理高维图像数据，本文将IC

神经元适配到深度卷积模型中。在高维输入中，IC神经元的表达能力会受到一定程度地限制。为解决该问题，本文将动态卷积 (Dynamic Convolution) 技术与 IC 神经元相结合，构造了一种新型卷积结构 (IC-CNN+)。在 IC-CNN+ 中，动态卷积可以进一步扩展 IC 神经元的表达能力，使其能更准确地拟合高维数据中的数据分布。实验结果表明，IC-CNN+ 的实验效果显著超过了 IC 神经元与卷积结构的简单组合。此外，IC-CNN+ 可以广泛提升深度卷积神经网络的性能，并几乎不引入新的计算消耗。

3. 一种基于 IC 神经元的近似等变图神经网络算法：IC-GNN+。图神经网络 (Graph Neural Network, GNN) 常用于提取 3D 数据的特征。本文将 IC 神经元适配到图神经网络中来提取 3D 特征，然而，IC 神经元与图神经网络架构的简单结合不满足空间 $SO(3)$ 变换的等变性或不变性约束。本文将群表示 (Group Representation) 和相应等变技术引入 IC 神经元，设计了一种基于 IC 神经元的近似等变结构 (IC-GNN+)。通过放松严格的等变性，该结构能够使图神经网络达到更高的表达能力，突破传统等变模型的限制，同时该结构能从训练数据中快速学习近似的等变性。实验结果表明，IC-GNN+ 在分子属性预测和点云分类中都取得了优秀的结果，展现了其在 3D 领域的重要应用价值。

4. 一种基于 IC 神经元的记忆型递归神经网络算法：IC-RNN+。递归神经网络 (Recurrent Neural Network, RNN) 通常用于处理序列型数据。本文将 IC 神经元适配到递归神经网络中来解决各种序列相关任务，然而，IC 神经元与递归结构的简单结合无法有效学习序列中的长期依赖关系。针对该问题，本文提出一种作用于神经元层面的记忆/遗忘结构，并将它与 IC 神经元相结合，构建了能捕获长期依赖关系的新型递归结构 (IC-RNN+)。实验结果表明，IC-RNN+ 可以有效解决包含长期依赖关系的自然语言或者时序任务，其效果超过了同为递归结构的 LSTM 和 GRU 算法。在一些任务上，IC-RNN+ 的结果也可以超过目前主流的 Transformer 模型的结果。

关键词：机器学习，神经网络，MP 神经元，卷积神经网络，图神经网络，递归神经网络

南京大学研究生毕业论文英文摘要首页用纸

THESIS: Research on New Neuron Model for Artificial Neural Network

SPECIALIZATION: Computer Science and Technology

POSTGRADUATE: Junyi An

MENTOR: Professor Furao Shen

ABSTRACT

Neural networks are a critical technology in machine learning. Currently, neural networks demonstrate exceptional performance in many significant challenges and their applications span various aspects of human society. From the perspective of model design, neural network research can be categorized into two main areas: 1. research on network architecture, and 2. research on neuron models. Over the past few decades, various network architectures have been proposed, but research on neuron models has been relatively sparse. Most neural networks still use the MP (McCulloch-Pitts) neuron model proposed in the 1940s. Although there have been some significant neuron models in the history of neural network development, most of them have not been widely adopted in mainstream neural network architectures.

This paper aims to explore and design new neuron models, aiming to develop universal, easily extensible, easily trainable, and highly expressive neuron models. Additionally, this paper considers the integration of new neuron models with mainstream network architectures, exploring efficient methods of using neuron models based on the characteristics of different network architectures and corresponding downstream tasks. In summary, the main contributions of this paper include:

1. A general neuron model based on the physical elastic collision model: the Inter-layer Collision Neuron (IC Neuron). Inspired by the physical one-dimensional elastic collision model, where quantities such as velocity and momentum undergo deterministic transformations before and after collisions, a general model (IC Neuron) is designed in this paper. Unlike traditional biologically-inspired neurons, IC Neurons have strong

physical rationality and mathematical interpretability. This paper demonstrates that IC Neurons can fit more complex nonlinear distributions than MP Neurons. Moreover, IC Neurons retain the easy integration, easy training, and lightweight characteristics of MP Neurons, making them suitable for various network architectures. Experimental results show that IC Neurons can be applied to various neural networks, achieving better generalization performance than MP Neurons across multiple tasks. Furthermore, IC Neurons maintain the same computational cost as MP Neurons, making them highly valuable in practical applications.

2. A high-performance convolutional neural network algorithm based on IC Neurons: IC-CNN+. Convolutional Neural Networks (CNNs) are typically used for processing high-dimensional image data. IC Neurons are applied to deep convolutional models in this paper. In high-dimensional inputs, the expressive capability of IC Neurons can be somewhat limited. To address this issue, Dynamic Convolution technology is combined with IC Neurons to construct a new convolutional structure (IC-CNN+) in this paper. In IC-CNN+, Dynamic Convolution further enhances the expressive capability of IC Neurons, allowing them to more accurately fit the data distribution in high-dimensional data. Experimental results indicate that IC-CNN+ significantly outperforms the simple combination of IC Neurons and convolutional structures. Additionally, IC-CNN+ can widely improve the performance of deep convolutional neural networks with almost no additional computational cost.

3. An approximate equivariant graph neural network algorithm based on IC Neurons: IC-GNN+. Graph Neural Networks (GNNs) are commonly used for extracting features from 3D data. IC Neurons are applied to GNNs for 3D feature extraction in this paper. However, the simple combination of IC Neurons and GNNs does not satisfy the equivariance or invariance constraints of spatial $SO(3)$ transformations. In this paper, Group Representation and corresponding equivariant techniques are introduced into IC Neurons, designing an approximate equivariant structure (IC-GNN+). By relaxing strict equivariance, this structure allows GNNs to achieve higher expressive capabilities, overcoming the limitations of traditional equivariant models, while quickly learning approximate equivariance from training data. Experimental results show that

IC-GNN+ achieves excellent results in molecular property prediction and point cloud classification, demonstrating its significant application value in the 3D domain.

4. A memory-based recurrent neural network algorithm based on IC Neurons: IC-RNN+. Recurrent Neural Networks (RNNs) are typically used for processing sequential data in this paper. IC Neurons are applied to RNNs to address various sequence-related tasks. However, the simple combination of IC Neurons and recurrent structures cannot effectively learn long-term dependencies in sequences. To solve this problem, the author proposes a memory/forgetting structure at the neuron level and combines it with IC Neurons to construct a new recurrent structure (IC-RNN+) capable of capturing long-term dependencies. Experimental results show that IC-RNN+ effectively solves natural language or time-series tasks involving long-term dependencies, outperforming other recurrent structures such as LSTM and GRU algorithms. In some tasks, IC-RNN+ also surpasses the results of the current mainstream Transformer models.

KEYWORDS: Machine Learning; Neural Networks; MP Neurons; Convolutional Neural Networks; Graph Neural Networks; Recurrent Neural Networks

目 录

第一章 绪论	1
1.1 引言	1
1.2 神经元建模的研究现状及面临的问题	4
1.3 本文的主要工作	6
第二章 理论基础和相关研究工作	9
2.1 MP 神经元	9
2.2 全连接神经网络结构	11
2.3 卷积神经网络结构	12
2.3.1 基础卷积模型	12
2.3.2 动态卷积方法	13
2.4 递归神经网络结构	14
2.4.1 基础递归模型	14
2.4.2 长期依赖关系模型	15
2.5 图神经网络结构	17
2.5.1 消息传递模型	17
2.5.2 图神经网络中的等变性和不变性	18
2.5.3 球谐函数	19
2.5.4 等变神经元模型	21
2.6 神经元模型和神经网络技术研究进展	22
2.7 本章小结	25
第三章 一种基于物理碰撞模型的通用神经元研究	27
3.1 背景介绍	27
3.2 基于弹性碰撞原理的数学模型	28

3.3	基于物理碰撞模型建模的 IC 神经元	30
3.3.1	IC 神经元的数学模型	30
3.3.2	信号复用机制	30
3.3.3	基于 IC 神经元的非线性表达能力分析	31
3.3.4	基于可调节超平面的 IC 神经元模型	33
3.4	IC 神经元结构在多种神经网络架构中的应用	35
3.4.1	IC 神经元与主流网络架构的结合	35
3.4.2	IC 神经网络复杂度分析	37
3.5	实验结果与讨论	38
3.5.1	数据集和基线配置	38
3.5.2	IC 全连接神经网络实验结果分析	40
3.5.3	IC 卷积神经网络实验及讨论	41
3.5.4	IC 递归神经网络实验及讨论	46
3.5.5	IC 神经元在 MP 衍生神经元体系中的应用	47
3.5.6	消融实验	50
3.6	本章小结	51
第四章 一种基于 IC 神经元的高性能卷积算法		53
4.1	背景介绍	53
4.2	动态卷积与 IC 卷积结构的结合	54
4.3	IC-CNN+ 中的动态卷积算法研究	57
4.3.1	全局结构	58
4.3.2	压缩模块	59
4.3.3	自注意力模块	60
4.3.4	动态卷积核生成	60
4.4	基于 IC 卷积结构的快速训练算法	62
4.5	实验结果与分析	64
4.5.1	数据集和基线配置	64
4.5.2	ImageNet 实验结果	65
4.5.3	CIFAR 实验结果	66

4.5.4	训练过程优化的实验结果	66
4.5.5	LADConv 模型的实验结果	67
4.5.6	可视化展示	69
4.5.7	消融实验	69
4.6	本章小结	71
第五章	一种基于 IC 神经元的近似等变图神经网络算法	73
5.1	背景介绍	73
5.2	神经元的等变性分析	74
5.3	基于 IC 神经元的近似等变模块	77
5.3.1	节点嵌入向量	77
5.3.2	近似等变的消息计算模块	78
5.3.3	更新模块	80
5.4	多层 IC 图神经网络	80
5.5	实验结果与分析	82
5.5.1	数据集与基线配置	82
5.5.2	分子数据集实验结果	84
5.5.3	点云数据集实验结果	86
5.5.4	消融实验	87
5.6	本章小结	88
第六章	一种基于 IC 神经元的记忆性递归神经网络算法	89
6.1	背景介绍	89
6.2	长期依赖与 IC 神经元之间的关系	90
6.3	神经元层面的记忆/遗忘结构	92
6.4	基于记忆/遗忘结构的 IC 递归神经网络	94
6.5	多层 IC 递归神经网络	95
6.6	实验结果与讨论	96
6.6.1	数据集和基线配置	96
6.6.2	文本分类数据集实验结果	99
6.6.3	时间序列数据集实验结果	100

6.6.4 消融实验	102
6.7 本章小结	103
第七章 总结与展望	105
7.1 全文总结	105
7.2 研究展望	106
参考文献	109
致 谢	121
A: 攻读博士学位期间的学术成果和获奖情况	123

插图目录

1-1	ResNet-18 神经网络的层级结构，虚线框表示网络架构的层级。大部分研究针对虚线框内的结构进行优化，神经元层面的相关研究相对较少。	2
1-2	本文各个章节之间的逻辑结构图。	7
2-1	MP 神经元结构。	10
2-2	全连接神经网络结构。图中相同颜色的节点表示同一隐藏层神经元。	11
2-3	卷积神经网络结构。卷积操作将滑动窗口内的特征映射到特定维度的向量中，所有滑动窗口共享一组映射参数。	12
2-4	递归神经网络结构。神经网络在每一步计算时都会利用上一步输出的隐藏变量。	14
2-5	图神经网络结构。在计算中，图中每个节点的嵌入向量都由与之相邻的节点进行更新，最终图神经网络输出每个节点更新多次后的嵌入向量。	17
2-6	等变和非等变下 3D 特征的变化 ^[34] 。 $F(\cdot T_g(\mathbf{X}))$ 表示一种等变的转换过程。等变性可以保证变化前特征细节的可追溯性，但是非等变性会打破这种可追溯的特性。	18
2-7	球谐函数在球面上的可视化展示。	20
2-8	将 3 维空间中的旋转投影到高维空间。左侧表示一个三维空间的方向向量和旋转后的向量；中间表示球谐函数表示下的高阶等变向量；右侧表示球谐函数投影到球坐标系的函数形式。	21
3-1	一维弹性碰撞的物理模型。两物体碰撞后，它们的速度发生相应变化。	28

3-2	神经元结构对比。(a) MP 神经元; (b) IC 神经元。	30
3-3	神经元在二维空间的非线性表示。(a) MP 神经元; (b) IC 神经元; (c) 基于 IC 神经元的一种 XOR 问题解决方案。	32
3-4	(a): 3-D 输入空间中的分割超平面 ($H = 0$); (b): 超平面根据调节权重 w' 进行旋转的过程。	33
3-5	全连接、卷积和递归结构中的 IC 神经元。红色区域代表 IC 神经元的计算结构。	37
3-6	各个任务上 IC 全连接神经网络的训练曲线。	42
3-7	IC-CNN 和传统 CNN 在图像分类任务中的训练曲线。	43
3-8	IC-RNN 和传统 RNN 在时间序列任务中的训练曲线。	46
3-9	MG 和 AUST 时间序列数据集上的预测结果。	49
4-1	IC-CNN+ 示意图。 $g(\mathbf{X})$ 为生成动态卷积的过程。虚线框内表示 IC 神经元的计算流程。	56
4-2	不同视觉动态化机制 (Vision Attention) 的示意图。符号 F_{NN} 表示用于抽象特征的各种神经网络。(a): 通道动态化机制; (b) 空间动态化机制; (c) 通道和空间混合动态化机制; (d) 卷积核动态化机制 (本文提出方法)。	57
4-3	LADConv 动态化方法的全局结构。 \mathbf{w}_θ 表示静态卷积核, \mathbf{w}_{dy} 表示动态卷积核, Position attention 表示动态卷积核在空间位置上的位置编码。LADConv 最终的输出用于替换 IC 卷积结构中的 $(\mathbf{I} * \mathbf{X})$ 项。	58
4-4	基于梯度的类激活图 (Grad-CAM) ^[81] 的可视化特征。所使用的基础模型是 ResNet-18。	70
5-1	旋转对称模型 ^[86] 。从左上到右下依次为 2 重对称, 3 重对称, 5 重对称, 10 重对称模型。	75
5-2	IC-GNN+ 模型的示意图。通过旋转到局部坐标系可以简化 CG 张量积的形式。简化后, IC-GNN+ 利用 IC 神经元近似表示 CG 张量积中的核心模块。	77

5-3	(a) 基于 IC 神经元的近似等变结构。其中红圈表示同一个 IC 神经元模型。(b) 多层消息模块。每一层计算的消息向量与边特征被送入一个 IC 近似等变模块。	81
6-1	IC 神经元与递归神经元的简单结合。IC 神经元只能用于提取当前时刻下输入信号的特征，不能考虑其它时刻的信号。	91
6-2	记忆/遗忘结构。	93
6-3	基于记忆/遗忘结构的 IC-RNN+ 结构。	95
6-4	构建多层 IC-RNN+ 的两种策略。(a) 在多层中重复利用历史状态 \mathbf{e}_t 。(b) 每一层重新计算历史状态 \mathbf{e}_t 。	96

表格目录

3-1	三种神经元在全连接神经网络中的分类效果。	40
3-2	不同神经网络模型的 FLOPs/Params 对比。FLOPs (KMac) 表示前 向传播的计算量。Params (K) 反映了可学习参数的数量。	41
3-3	图像任务中 IC-CNN 和传统 CNN 的性能对比。	43
3-4	CIFAR 数据集上的实验结果。表中结果均为 Top-1 准确率 (%)。	44
3-5	ImageNet 数据集的实验结果。	45
3-6	IC 卷积神经网络在 PASCAL VOC 2007+2012 测试集上的检测性 能。	45
3-7	时序任务中 IC-RNN 和传统 RNN 的性能对比。	47
3-8	PSNN、SMN 及其相应的 IC 模型在 MG 和 AUST 时间序列数据 集上性能的比较。	49
3-9	分类任务中 IC-FC, FC-A(更深的模型), FC-B(更宽的模型) 模型 的对比。	50
3-10	分类任务中 IC-FC, FC-A, FC-B 模型的计算消耗 (FLOPs/Params) 对比。	51
3-11	IC 神经网络中使用不同激活函数的消融实验。	51
4-1	深度模型中, IC 神经元退化的比例。	55
4-2	ImageNet 下各种模型的精度和计算消耗对比。准确率 (%) 使用 single-crop 测试。IC+ 表示 IC-CNN+ 模型。	65
4-3	CIFAR 数据集下的 IC-CNN+ 以及基础模型的 Top-1 分类结果。	66
4-4	IC 卷积神经网络在快速训练策略下的 ImageNet 分类结果。	67
4-6	基础模型为 MobileNetV2 时的 ImageNet 分类结果。	69
4-7	IC-CNN+ 中动态卷积方法的消融实验。	70
4-8	不同压缩模块的消融实验。	71

4-9	LADConv 超参数的消融实验。	71
5-1	等变神经元, MP 神经元, IC 神经元的 n 重对称实验。	76
5-2	几种图神经元性能的对比。	76
5-3	OC20 实验的超参数配置	83
5-4	QM9 实验的超参数设置。	84
5-5	OC20 IS2RE 数据集上的能量预测效果.	85
5-6	在 QM9 数据集上的各种化学属性结果。 † 表示使用不同的数据分区。粗体和下划线分别表示最佳结果和第二佳结果。	85
5-7	ModelNet40 数据集中的分类效果。	86
5-8	近似等变性的消融实验。	87
5-9	随机旋转测试集下各模型的表现效果。	88
5-10	IC-GNN+ 组成模块的消融实验。	88
6-1	斐波那契数据实验。表中数值表示预测值与真实值之间的 MAE。	91
6-2	数据集信息统计。	97
6-3	时间序列数据集的基本信息。	98
6-4	文本分类数据集中的对比结果。	99
6-5	时间序列数据上各基础模型的预测结果。	100
6-6	时间序列数据上多种时序方法的预测结果。	101
6-7	记忆/遗忘模块的消融实验。	102
6-8	记忆/遗忘模块的消融实验。	103

第一章 绪论

1.1 引言

人工智能 (Artificial Intelligence, AI) 是一门致力于使计算机系统具备类似人类智能的科学和技术领域，它的概念最早由 John McCarthy 在 1956 年的达特茅斯会议提出^[1]。人工智能技术经过半个多世纪的发展，已经覆盖了人类社会的方方面面，它涉及的领域包括信息，医疗，教育，交通，工业等。在互联网高速发展的现在，人工智能技术能推动着各项技术前行。在国内，国家更是将其列为了重要发展目标之一。2019 年 8 月，科技部印发《国家新一代人工智能开放创新平台建设工作指引》和《国家新一代人工智能创新发展试验区建设工作指引》，对现有人工智能平台实行有效支撑。2022 年 7 月，国家出台《关于加快场景创新以人工智能高水平应用促进经济高质量发展的指导意见》，旨在积极拓展人工智能的各类应用场景。

在人工智能领域内，机器学习是一条重要分支，它致力于将历史数据输入各类计算机系统，并使其能够从数据中发现潜在规律或者特征，学习到相关知识信息。最终，机器学习技术还会利用这些学习到的知识来做出预测和决策^[2]。从在上世纪 80 年代，“从样例中学习”的思想广泛进入到机器学习算法的研究，诞生了大量的经典算法，例如支持向量机^[3]，决策树^[4]，多层感知器^[5]等。“从样例中学习”的思想也更贴近人类学习未知知识的过程。在过去的十多年中，随着硬件技术的突飞猛进，机器学习技术也得到迅猛发展，相关算法被应用到现实世界的各个任务中，代表性的有自然语言处理^[6]，计算机视觉^[7]，语音识别^[8]等。

在机器学习的众多技术中，神经网络较为特别，这是一种受启发于生物神经网络，通过连接若干基础计算单元组成的模型。早在上个世纪 20 年代年代，Wilhelm Lenz 和 Ernst Ising 就对类似神经元阈值元素组成的非学习网络结构进行了研究。1958 年，心理学家 Frank Rosenblatt 发明了感知器模型，真正意义实现了第一个人工神经网络。在神经网络发展历程中，虽然由于硬件系统的缺陷，

其研究一度陷入停滞，但随着 21 世纪硬件技术的进步，神经网络逐渐表现出强大的学习性能。2006 年，Hinton 等人成功训练了一个深层次的复杂神经网络模型，拉开了深度神经网络的研究热潮。深度模型揭露了：通过增加神经网络深度和宽度，可以有效拓展模型的模型容量。这一性质也是其它机器学习技术所不具备的。目前，深度神经网络已经在各种挑战中取得了重大突破，尤其在一些复杂任务中表现出超越人类的水平。伴随着实际应用，深度神经网络方面的研究也逐渐成为各个应用神经网络领域中的焦点。在 2019 年，图灵奖授予 Geoffrey Hinton, Yann LeCun 和 Yoshua Bengio 三位科研人员，以表彰他们在深度学习领域做出的突出贡献。

神经网络的研究在于两个重点：1. 神经网络建模；2. 训练算法。这两部分组成完整的神经网络应用框架。目前训练算法主要以梯度反向传播^[9]及其衍生算法为主，而神经网络建模的相关研究集中在神经元模型和网络架构两个方面。网络架构涉及宏观层次的结构建模，包括神经元的连接方式，模块的组合等。随着神经网络研究热度的上升，各种天马行空的网络架构或者宏观模型被设计出来，然而，大部分神经网络模型的研究更关注网络架构层面，对最基础的神经元模型的考虑相对较少。以图1-1所示 ResNet-18 模型为例，近几年有很多工作关注整体模型^[7]，残差模块^[10-11]或卷积操作^[12-13]，它们都属于高层次的网络架构，神经元层面的结构却鲜为关注。

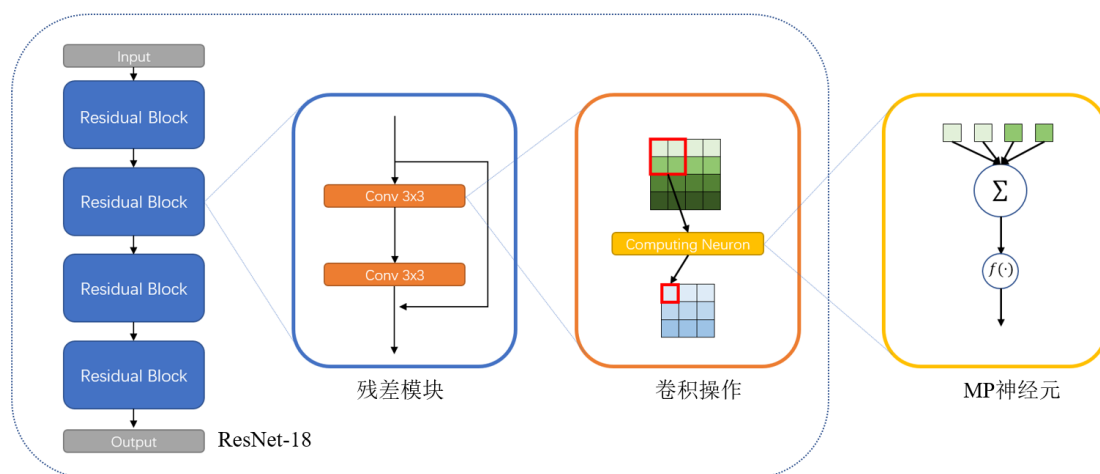


图 1-1 ResNet-18 神经网络的层级结构，虚线框表示网络架构的层级。大部分研究针对虚线框内的结构进行优化，神经元层面的相关研究相对较少。

本文从神经元建模的层面出发探索高性能的神经元模型。目前的大部分神经网络依赖生物系统启发的神经元，这类神经元模型借鉴生物神经细胞的信息

交互行为，试图模拟真实的生物神经网络^[14]。其中 MP (McCulloch-Pitts) 神经元模型是其中最为著名和泛用的一种^[15]。MP 神经元模型诞生于上个世纪 40 年代，由 Warren McCulloch 和 Walter Pitts 提出。在随后的演化中，MP 神经元逐渐被确定为 $y = f(\mathbf{W}\mathbf{x})$ 的数学形式，其中 \mathbf{x} 和 \mathbf{y} 是神经元输入和输出， $f(\cdot)$ 是非线性激活函数， \mathbf{W} 是输入的权重。在随后的近 80 年，也陆续有其它类型的神经元模型被提出，例如 Hodgkin-Huxley 模型^[16]和脉冲神经元模型^[17]等。然而 MP 目前依然是主流的神经网络的基础计算单元。它具有以下几个优点：1. 结构灵活，易堆叠成大型网络；2. 模型轻量；3. MP 神经元组成的模型能够逼近任意连续函数。然而，MP 神经元的非线性函数形式十分简单，具有较大的拓展空间。本文总结 MP 神经元的优点和缺点，研究神经元工作机理，致力于设计能够代替 MP 神经元的新型神经元模型。

在设计新型神经元模型之前，本文回顾神经元研究历史，发现很多神经元模型都从生物学的角度出发，模仿生物神经网络的行为。然而，生物神经元是一个非常复杂的模型，人工神经网络中的神经元只能模仿其最简单的行为特征。其次，脑神经网络为生物赋予了理解，推理，记忆等诸多功能，然而这些功能的行为逻辑与人工神经网络解决现实问题的行为逻辑相差甚大。此外，很多脑神经机制至今依然还处于研究阶段。因此，从生物神经元中仿真依然面临很严峻的挑战。一些工作尝试从数学或者物理学层面去设计新型神经元或神经网络，例如 Hopfield 模型^[18]和玻尔兹曼机 (Boltzmann Machine)^[19]。这些模型往往更具有物理合理性和数理解释性。然而，这一类研究同样也面临着诸多挑战，尚未在现实应用中展现出计算优势。总结来说，在深度学习占据主导地位的当下，新型神经元的设计面临的挑战可以总结为以下几点：

- 在解决复杂现实任务时，需要构建深度神经网络以保证足够的模型容量 (Model Capacity)。新型神经元结构需要具备易集成，拓展性强的特点，便于构建深度神经网络或应用到已经存在的深度神经网络中；
- 神经元的非线性拟合能力是神经网络能够近似复杂连续函数的重要前提。新型神经元需要具备一定的非线性表达能力。然而，过分复杂的非线性结构会增加拟合未知分布的难度，表现为更高的过拟合风险；
- 目前神经网络网络架构已经发展出多个主流分支，例如卷积神经网络，图神经网络，递归神经网络等。新型神经元在应用时需要考虑不同网络架构

的特性；

- 深度神经网络会将神经元的计算消耗放大成千上万倍，因此，新型神经元需要时轻量的，计算友好的。

虽然新型神经元的研究面临诸多挑战，但它具有重大意义。一般来说，网络架构的研究会偏向特定的任务或领域，例如卷积神经网络之于图像等高维数据，图神经网络之于非结构化数据，递归神经网络之于时序数据等。而神经元的研究既可以针对专一任务，也可以泛用于大部分神经网络架构及任务，这对于模型种类繁多的深度学习领域尤为重要。总结来说，神经元模型研究的意义可以总结为以下几个方面：

- 神经元与大部分网络架构研究互补，新型神经元模型可以直接嵌入到大部分已经设计好的网络架构中，省去繁琐的超参数调试过程和部分训练成本，具有重要应用意义。
- 神经元的研究不受下游任务的约束，新型神经元模型应用各种各样的机器学习任务中。
- 神经网络的结构由宏观到微观可以分为多个层次，其中神经元是最底层的计算单元，新型神经元的研究能够促进更高层次模块的设计和研究。

目前深度神经网络大多仍基于简单的 MP 神经元，它所展示出来的学习能力还比较初步，具有很大的优化空间。探索和设计学习能力更强大的新型神经元，对深度学习，乃至整个神经网络领域，都有重大意义。

本章第一节讲述了神经网络技术在人工智能领域中的地位以及新型神经元研究的重要性，后续章节安排如下：第二节介绍神经网络技术和神经元建模的研究现状；第三节介绍本文对新型神经元研究的相关工作。

1.2 神经元建模的研究现状及面临的问题

神经网络技术已经被应用到机器学习任务的各个方面。一般来说，神经网络是由若干神经元单元组成的数学模型。在应用时，利用学习算法 (例如 BP 反向传播算法) 优化神经网络中的参数可以使整个模型拟合未知的分布。针对神经网络的研究主要集中在两个方面：神经网络学习算法，神经网络模型建模，这两部

分研究相互独立。神经网络模型建模又细分为神经元模型和网络架构两个层面，经过半个多世纪的研究，各种神经网络架构相继被提出，常见的网络架构有前馈全连接网络^[5]、卷积神经网络^[20]、图神经网络^[21]、递归神经网络^[22]等。然而，针对神经元建模的相关相对较少。在 1943 年，Warren McCulloch 和 Walter Pitts 提出 MP 神经元^[15]，经过七八十年的发展，MP 神经元已成为当前最为成功且流行的神经元模型。期间还诞生了其他类型的神经元模型，例如第二代神经元模型：脉冲神经网络^[14]，但是由于 MP 神经元在构建深度模型上的巨大优势，其它类型神经元都无法撼动它的主导地位。然而，MP 神经元结构简单，在生物学中它仿照的也是最简单的神经细胞行为，其结构存在较大的优化空间。

大部分人工神经元模型是根据生物神经系统的结构进行数学建模的。根据建模尺度的不同，生物启发的人工神经元模型可分为三大类：(1) 根据宏观尺度的生物神经细胞行为建模，这一类研究以 MP 模型为代表，将神经元模型表达为静态的实值函数，可以用于拟合确切的分布函数。以这一类神经元为基础的神经网络模型在多个领域都取得了重大的突破；(2) 根据微观尺度神经细胞行为建模，这一类研究以脉冲神经网络模型为代表，描述了生物神经系统在毫秒/微秒级水平上的通信行为，考虑更底层的生物神经元行为，这一类模型的优势是具有一定的时空表达能力^[23]。脉冲神经网络模型在高性能计算^[24]、神经形态计算^[25]等领域受到越来越多的关注和应用；(3) 目前一部分研究^[26]考虑从中观尺度生物神经元行为中建模，描述了生物神经系统在分钟/小时/日级水平上的通信行为，这一类模型可以兼顾时空表达能力和强拟合能力的特性，在时间序列等任务上表现出一定的优越性。尽管这三种尺度的神经元模型都旨在探索更贴近生物神经元行为的数学模型，然而，真实的生物神经元结构十分复杂，甚至有些生物学机理仍然在探索中，目前的人工神经元研究只能模拟其中的部分的行为。

除了生物神经元体系外，一些物理模型或生物物理模型也极大的促进了神经元研究的发展。John Hopfield 在 1982 年提出了 Hopfield 网络^[18]，它基于一种能量函数来描述神经元之间的相互作用。玻尔兹曼机 (Boltzmann Machine)^[19]是一种基于能量的随机生成式神经网络模型，它受到了统计物理学中玻尔兹曼分布和马尔可夫链的启发，可以用于模拟大量随机变量之间的相互作用。Hopfield 网络和玻尔兹曼机在图像识别，优化控制等任务中都表现出良好应用价值。Christopher Morris 在 1981 提出的 Morris-Lecar 神经元^[27]和 Eugene M. Izhikevich 在 2003

提出的 Izhikevich 神经元^[28]关注神经元中膜电位变化的物理过程，他们有效地拓展了脉冲神经网络的应用。基于物理或生物物理的模型往往具有更强的物理合理性和数理解释性。然而，相比于传统的 MP 神经元，这一类研究尚未在现实应用中展现出计算优势，还处于探索阶段，但它们的阶段性成功也激励着后来的研究跳出生物体系，从更广泛的交叉学科中构建新型神经元模型。

上述新型神经元的研究已经展现了在小体系中超过了 MP 神经元的性能，但在复杂体系中，它们构成的神经网络往往受到其他因素的约束，包括：1. 训练算法；2. 计算资源；3. 模型规模等。到目前为止，非 MP 的神经元依然难以应用到深度神经网络模型以及复杂现实世界任务。回顾机器学习的发展史，神经元模型的数次革新都极大推动了机器学习技术的发展。尽管当下的深度学习技术的重心依旧建立在 MP 神经元模型上，但 MP 神经元本身所刻画的数学模型非常简单，以它为核心构建的神经网络模型所展现出的建模能力依然比较初步。针对当下神经网络发展现状，从更广泛的交叉学科中探索和设计综合性能更强的神经元模型，对机器学习乃至人工智能的发展具有重要意义。

1.3 本文的主要工作

如前文所述，神经网络建模包括架构和神经元模型两部分。在过去的二十年中，出现了各种基于不同网络架构的学习模型与方法，但基础神经元模型的研究相对较少。本文专注于新型神经元建模，分析传统 MP 神经元的优点和缺点，结合多种网络架构的特性，设计更高性能的计算单元。图1-2列举出了本文各章之间的联系，具体内容如下：

在第三章中，本文探索通用神经元的设计。不同于传统神经元模型借鉴生物神经系统的思路，本文从物理中的弹性碰撞模型启发信息前馈模型，并提出了一种新型神经元，被称为层间碰撞神经元 (IC 神经元)。IC 神经元保留了 MP 神经元易集成，轻量级等优点，同时，它通过切分输入空间，增加了神经元输出的非线性模式，提升了对目标函数的非线性拟合能力。通过异或 (XOR) 问题，本文展示了 IC 神经元解决线性不可分问题的可能。由于 IC 神经元是一种最基础的计算单元，它可以被应用到各种网络架构中。实验结果表明，在相同网络架构和超参数配置下，多种架构下的 IC 神经网络均取得优于传统神经网络的性能。该

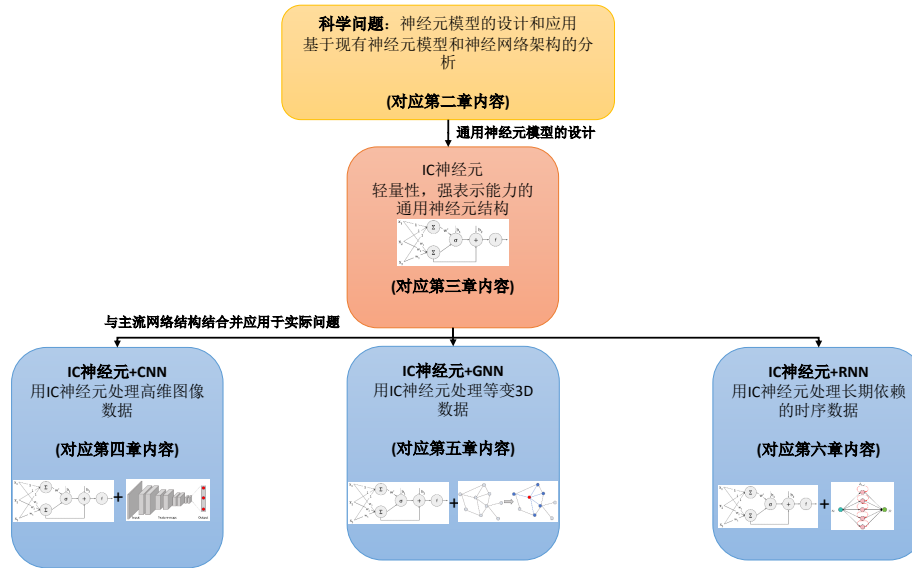


图 1-2 本文各个章节之间的逻辑结构图。

章的实验覆盖多项任务，包括图像，文本，时序等，较为全面的反映 IC 神经元通用能力和应用价值。

在第四章、第五章和第六章中，本文将主流的网络架构与第三章提出的 IC 神经元相结合，进一步设计高性能的计算单元来解决一些实际问题。

在第四章中，本文探索 IC 神经元与卷积神经网络的高效结合。卷积神经网络通常用于处理高维的图像数据。而 IC 神经元非线性拟合能力会受到高维数据的限制。针对该问题，本文利用动态卷积 (Dynamic Convolution) 技术将 IC 卷积结构中切分输入空间的线性超平面替换为一个可学习的超曲面。更新后的 IC 卷积结构被称为 IC-CNN+，它能更准确地拟合高维数据中的数据分布。此外，动态卷积的类型也对 IC 卷积结构的性能有明显影响。目前已有的动态卷积方法都面临一个重要问题：缺乏动态卷积核与局部特征之间的关联。针对该问题，本文提出一种新的动态卷积算法 (LADConv)，它能够完美地适配 IC-CNN+ 结构。实验结果表明，IC-CNN+ 能显著提升一些深度模型在视觉任务中的性能，并且实验也证明新型动态卷积 LADConv 的性能超过了目前已有的动态卷积算法。

在第五章中，本文探索 IC 神经元在图神经网络中的深层应用。图神经网络常用于处理三维空间中的无结构数据，例如点云或者分子数据，而这些数据对应的下游任务除了要求提取几何特征外，还要求特征满足 $SO(3)$ 变换等变性或者 $SO(3)$ 变换不变性。IC 神经元与图神经网络的简单结合无法满足该需求，因

为 IC 神经元本身不具备等变性约束。本文针对上述问题，将等变群表示技术引入 IC 神经元，设计了一种基于 IC 神经元的近似等变结构 IC-GNN+。IC-GNN+ 能够突破传统等变模型表达能力的理论上限，同时，IC-GNN+ 也可以通过训练逼近严格的等变。实验结果表明，IC 图神经网络在分子属性预测和点云分类任务中都取得了优异的性能。

在第六章中，本文探索了 IC 神经元与递归神经网络的高效结合。递归神经网络通常用于处理序列型数据，许多现实世界中的序列任务都需要提取长期依赖关系。然而，IC 神经元本身不具备提取长期依赖关系的能力，其与递归结构的简单结合仍然会受到限制。针对这一问题，本文提出了一种作用于神经元层面的记忆/遗忘机制，并将其与 IC 神经元相结合，构建了一种新型递归结构 IC-RNN+。实验结果表明，IC-RNN+ 能够有效学习长期依赖关系，并在多个序列型任务中超过同为递归模型的 LSTM 以及 GRU 模型。此外，在一些任务上，IC-RNN+ 也能取得和主流的 Transformer 类模型媲美的结果。

在第七章，本文对提出的算法进行了总结，重新梳理了各种 IC 神经元结构，以便更好地理解本文的工作。另一方面，由于神经元建模是一个极具挑战性的任务，随着研究的深入，很多尚未解决的学术问题逐渐浮出水面，这些问题将激励着未来的研究方向，需要研究人员做进一步的思考与探索。这一章也对未来可能的研究方向与应用前景进行展望与探讨。

第二章 理论基础和相关研究工作

本章中对神经元以及神经网络架构的基础知识进行介绍，以便更好地理解全文的研究工作。网络架构包括卷积神经网络架构，递归神经网络架构，图神经网络架构。之后，本章对后文图神经网络章节用到的一些数学工具进行介绍，包括球谐函数 (Spherical Harmonics)，CG 张量积 (Clebsch-Gordan Tensor Product) 等。最后，本章对现有的神经元以及神经网络的研究进行回顾，包括基础神经元的和网络架构层面的研究。

2.1 MP 神经元

神经元是神经网络的基础构件，目前，大部分神经网络都以 MP 神经元为基础。MP 神经元最早由心理学家 W · S · McCulloch 和数学逻辑学家 W · Pitts 在 1943 年构建^[15]，它们模仿生物系统中的神经元结构，并建立了能够描述神经元相互作用的数学模型。最初的 MP 神经元也被称为线性阈值单元 (LTU)，在传递过程中应用了一种阈值函数，根据输入和权重的加权和是否超过了设定的阈值来输出不同的结果。其数学表达式可以表示为：

$$y = \begin{cases} 1 & \text{if } \sum_{i=1}^n w_i \cdot x_i + b > \text{threshold} \\ 0 & \text{otherwise.} \end{cases} \quad (2-1)$$

最初提出的 MP 模型只支持二进制输入和输出。后续研究表明，任何布尔函数都可以通过 MP 神经元模型来实现。通过适当的权重设置，许多布尔决策问题可以用 MP 神经元模型来解决。例如，可以利用 MP 神经元模型实现与函数 (AND) 和或函数 (OR)。

随着计算机技术的飞速发展，模型的输入输出逐渐变得丰富起来，相关任务也更加复杂，这为 MP 神经元的应用带来了几个问题：

- 当输入是非布尔的实数时，MP 神经元应当如何运算？
- MP 神经元能够表示与函数 (AND) 和或函数 (OR)，那是否能表示异或函数 (XOR) 或者更复杂的函数形式？
- 阈值函数是一种不连续的函数，当输出期望是连续函数时，MP 神经元应当怎么样调整？

为了解决上述问题，后续的研究人员对 MP 神经元进行了改进，允许其接受整数或浮点数输入和输出。另外，他们还用连续函数替代了阈值函数，如 Sigmoid 和 Tanh 函数等，这些函数通常被称为非线性激活函数。因此，MP 神经元的结构发生了变化，如图2-1所示。其公式表示为：

$$y = f\left(\sum_i^n w_i x_i + b\right). \quad (2-2)$$

转换为矩阵表示为：

$$y = f(\mathbf{w}\mathbf{x} + b), \quad (2-3)$$

其中， $f(\cdot)$ 代表激活函数， \mathbf{w} 和 w_i 表示输入的权重， b 表示偏置。在这个模型中，神经元接收来自 n 个其他神经元传递过来的输入信号，这些输入信号通过带权重的连接进行传递，神经元接收到的总输入值经过激活函数处理，生成神经元的输出。激活函数通常是单调递增、连续、可微和有界的。然而，在输入为实数的情况下，激活函数也可以是分段线性函数或阶跃函数。最近，研究人员还探索了具有多个零点的非单调、无界和振荡激活函数，这些激活函数在一些任务中能够表现出更好的性能。

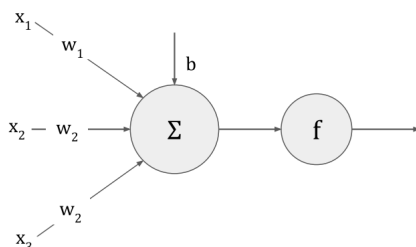


图 2-1 MP 神经元结构。

经过数十年的发展，MP 神经元模型已经被应用在神经网络领域的方方面面。此外，随着深度学习的发展，MP 神经元被应用于更深层次的神经网络结构

中，如深度卷积神经网络（Deep Convolutional Neural Networks）^[20]。通过增加网络的深度和复杂性，MP 神经元在图像识别^[7]、自然语言处理^[6]等领域取得了显著的成果。

总的来说，MP 神经元作为神经网络的基本组成单元，经过多年的发展和演化，在人工智能领域发挥着重要作用，为各种复杂任务的解决提供了有效的工具和方法。

2.2 全连接神经网络结构

全连接神经网络（Fully-Connected Neural Network, FC Neural Network），也称为多层感知器（Multilayer Perceptron, MLP），是一种经典的前馈神经网络结构，其由多个全连接层组成。从 MP 神经元的角度来看，全连接神经网络的核心思想是每个神经元与上一层的所有神经元都有连接，因此全连接层中的每个神经元都能够接收到上一层所有神经元的输出，并通过权重进行加权求和以及激活函数处理，从而得到下一层神经元的输出。

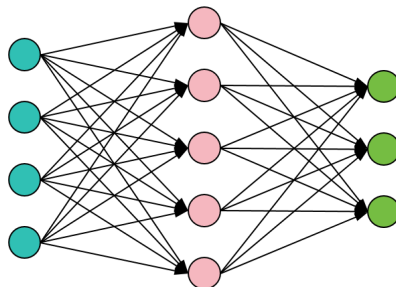


图 2-2 全连接神经网络结构。图中相同颜色的节点表示同一隐藏层神经元。

全连接神经网络如图 2-2 所示。整个全连接神经网络由多个全连接层堆叠而成，每一层都通过反向传播算法进行训练，优化网络参数，使得网络能够更好地适应给定的任务。

目前，全连接神经网络是神经网络领域中最常用的结构之一，适用于各种机器学习任务，尤其在图像识别、自然语言处理等领域取得了重要的成就。然而，全连接神经网络的参数量较大，且无法处理特殊的数据类型，因此在实际应用中往往需要结合其他技术进行优化和改进。

2.3 卷积神经网络结构

2.3.1 基础卷积模型

卷积神经网络 (Convolutional Neural Network, CNN) 是一种受 MP 神经元概念启发的神经网络架构, 它在计算机视觉领域取得了巨大成功。CNN 主要用于处理图像数据等高维数据, 其核心思想是通过卷积操作来提取图像中的特征, 并通过这些特征来进行分类、检测或其他任务。

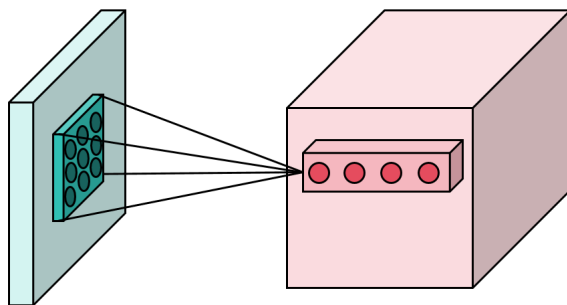


图 2-3 卷积神经网络结构。卷积操作将滑动窗口内的特征映射到特定维度的向量中, 所有滑动窗口共享一组映射参数。

在 CNN 中, 卷积操作是一种核心的操作, 如图 2-3 所示, 它类似于滑动窗口与 MP 神经元的组合。具体来说, 卷积操作是将一个滤波器 (也称为卷积核) 与输入数据的局部区域进行逐元素相乘, 并将结果相加得到输出。这个过程可以用以下数学公式表示:

$$Y(i, j) = \sum_m \sum_n X(i + m, j + n) \cdot W(m, n) + b, \quad (2-4)$$

其中, $Y(i, j)$ 表示输出特征图的第 i, j 个像素值, $X(i + m, j + n)$ 表示输入图像的第 $i + m, j + n$ 个像素值, $W(m, n)$ 表示卷积核的第 m, n 个权重值, b 表示偏置项。这个公式表达了输入图像上每个位置的输出是通过卷积核与该位置及其周围像素值的加权求和计算得到的, 即 MP 神经元的计算方式。需要注意的是, 方程 (2-4) 中省略了激活函数。下文用 $*$ 表示卷积过程, 传统卷积层可以被整理为:

$$\mathbf{Y} = \mathbf{w} * \mathbf{X}. \quad (2-5)$$

基于 MP 神经元的卷积层是卷积神经网络模型的核心操作, 它可以用来提取局

部特征。卷积神经网络通常由多个卷积层、池化层和全连接层组成。在卷积层中，通过使用不同大小的卷积核和不同数量的滤波器，可以提取不同层次的特征。池化层用于降低特征图的尺寸，并增强模型对平移不变性的学习能力。全连接层用于将卷积层提取的特征映射到最终的输出类别。

总的来说，卷积神经网络通过卷积操作和层级结构的设计，实现了对图像等高维数据的高效处理和特征提取，成为了图像分类、目标检测等计算机视觉任务中的重要工具。

2.3.2 动态卷积方法

在第四章中，本文提出了一种卷积结构中的特殊计算模型，这一技术使用了卷积中的动态化方法，即视觉注意力机制 (Vision Attention)，在过去十年中，这类方法被广泛应用于卷积神经网络的各个方面。根据动态化的目标，这些方法可以分为几类：1. 针对通道的动态化方法 (Channel Attention)，它们生成动态系数以表示特征图中不同通道的重要性；2. 针对空间像素的动态化方法 (Spatial Attention)，它们生成动态系数以表示特征图中空间像素的重要性。此外，自注意力机制 (Self-attention) 也是一种动态化的计算模块，被广泛应用于视觉模型中。这些方法大多针对的目标是卷积网络中的特征图 (Feature Maps) 进行优化。最近几年，研究人员开始关注将动态化技术应用到卷积核，提出了一种被称为动态卷积的方法，这也是本文在第四章中重点研究的技术。

传统的卷积中，卷积核的权重在训练后是固定的。相比之下，动态卷积方法将注意力应用于静态卷积核，使最终的卷积核的数值可以动态调整。其一般形式可以表示为

$$\begin{aligned} \mathbf{w}_{dy} &= g(a(\mathbf{X}), \mathbf{w}_\theta) \\ \mathbf{Y} &= \mathbf{w}_{dy} * \mathbf{X}, \end{aligned} \tag{2-6}$$

其中 \mathbf{w}_θ 和 \mathbf{w}_{dy} 表示静态卷积核 (参数化核) 和动态卷积核。 $a(\cdot)$ 用于产生注意力系数， $g(\cdot)$ 用于构建动态卷积核。大多数工作都集中在设计操作 $g(\cdot)$ 上。与传统的卷积依赖于固定权重的滤波器不同，动态卷积允许根据当前输入样本动态调整滤波器的权重，从而捕捉更精细的特征。在这个领域的开拓者中，Brandon Yang 等人^[12]和 Yinpeng Chen 等人^[13]定义了动态卷积的原型，他们将动态滤波

器定义为由依赖于输入特征的注意力加权的 n 个静态卷积核的聚合，表示为：

$$\mathbf{w}_{dy} = a(\mathbf{X})_1 \mathbf{w}_{\theta_1} + a(\mathbf{X})_2 \mathbf{w}_{\theta_2} + \cdots + a(\mathbf{X})_n \mathbf{w}_{\theta_n}, \quad (2-7)$$

其中每个 $a(\cdot)_i$ 都表示一个轻量级的 SE 模块^[10]。这些工作在轻量级架构中展示了准确性的提高，为后续研究铺平了道路。Li Yunsheng 等人^[29]和 He 等人^[30]分别利用矩阵分解和稀疏核来代替 n 个平行卷积核，增强了动态卷积的参数和计算效率。为了探索动态卷积的潜力，Chao Li 等人^[31]遵循了 n 个平行卷积核，并在核空间的多个维度上引入了额外的注意力，取得了在流行的视觉任务中优异结果，包括 ImageNet^[32]和 COCO^[33]数据集。由于动态卷积灵活的性质，本文在第四章将其应用到 IC 卷积模块的研究中，此外，本文也在第四章对上述动态卷积方法进行了深入研究，分析他们中存在的问题，探索更高效的结构。

2.4 递归神经网络结构

2.4.1 基础递归模型

递归神经网络 (Recurrent Neural Network, RNN) 是一种基于 MP 神经元的神经网络架构，可以用于处理序列数据，例如文本、语音和时间序列数据。RNN 可以被看作是一种具有内部反馈机制的全连接神经网络，其中隐藏层的输出被送回到网络的输入端。这种内部反馈机制使得 RNN 能够捕捉序列中的时间依赖关系，并在处理序列数据时具有记忆功能。

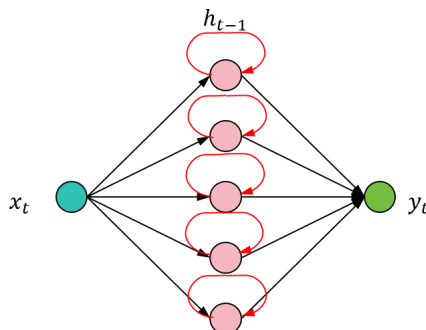


图 2-4 递归神经网络结构。神经网络在每一步计算时都会利用上一步输出的隐藏变量。

RNN 的结构如图2-4所示，隐藏状态更新公式可以表示为：

$$h_t = f(\mathbf{W}x_t + \mathbf{U}h_{t-1} + \mathbf{b}), \quad (2-8)$$

其中， h_t 是时间步 t 的隐藏状态， x_t 是时间步 t 的输入， \mathbf{W} 和 \mathbf{U} 是权重矩阵， \mathbf{b} 是偏置向量， $f(\cdot)$ 是激活函数。

RNN 的输出可以通过隐藏状态进一步计算得到：

$$y_t = g(\mathbf{W}'h_t + \mathbf{b}_y), \quad (2-9)$$

其中， y_t 是时间步 t 的输出， \mathbf{W}' 是从隐藏状态到输出的权重矩阵， \mathbf{b}_y 是偏置向量， $g(\cdot)$ 是激活函数。

RNN 的一个关键特点是它的隐藏状态不仅依赖于当前时间步的输入，还依赖于前一个时间步的隐藏状态。这种内部反馈机制使得 RNN 能够在处理序列数据时保留一定的记忆，从而更好地捕捉序列中的不同单元之间的依赖关系。

总的来说，递归神经网络通过内部反馈机制和隐藏状态来处理序列数据，并且具有一定的记忆功能，能够捕捉序列中的时间依赖关系。

2.4.2 长期依赖关系模型

RNN 提出后，从 RNN 结构又衍生出两种递归神经网络结构：LSTM（长短期记忆网络）和 GRU（门控循环单元）。

LSTM（长短期记忆网络）：LSTM 是一种特殊的递归神经网络结构，旨在解决传统 RNN 无法解决的长期依赖问题。它通过引入三个门控机制来控制信息的流动：遗忘门（forget gate）、输入门（input gate）和输出门（output gate）。与传统 RNN 不同，LSTM 具有一个额外的记忆单元（cell state），可以根据门控机制选择性地保留或丢弃信息。

在 LSTM 中，隐藏状态更新公式由以下部分组成：

$$\begin{aligned}
f_t &= \lambda(W_{hf}h_{t-1} + W_{xf}x_t + b_f) \\
i_t &= \lambda(W_{hi}h_{t-1} + W_{xi}x_t + b_i) \\
\tilde{C}_t &= \tanh(W_{h\tilde{C}}h_{t-1} + W_{x\tilde{C}}x_t + b_{\tilde{C}}) \\
C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \\
o_t &= \lambda(W_{ho}h_{t-1} + W_{xo}x_t + b_o) \\
h_t &= o_t \odot \tanh(C_t),
\end{aligned} \tag{2-10}$$

其中， $\lambda(\cdot)$ 表示 Sigmoid 激活函数。遗忘门的作用是控制了记忆单元中的信息是否被遗忘，输入门的作用是控制了新信息如何被整合到记忆单元中；在 LSTM 中，记忆单元的更新由遗忘门和输入门共同决定；输出门控制了隐藏状态的产生。

GRU（门控循环单元）：GRU 是一种更轻量级的门控循环神经网络，相比于 LSTM，它具有类似的功能，但参数更少，且计算效率更高。GRU 包含了更新门（update gate）和重置门（reset gate），以控制隐藏状态的更新。

在 GRU 中，隐藏状态更新公式由以下部分组成：

$$\begin{aligned}
r_t &= \lambda(W_{hr}h_{t-1} + W_{xr}x_t + b_r) \\
z_t &= \lambda(W_{hz}h_{t-1} + W_{xz}x_t + b_z) \\
\tilde{h}_t &= \tanh(W_{hh}(r_t \odot h_{t-1}) + W_{xh}x_t + b_h) \\
h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t,
\end{aligned} \tag{2-11}$$

其中，重置门的作用是控制了历史隐藏状态有多少信息被遗忘；更新门的作用是控制了新的隐藏状态和历史隐藏状态的权重比例；最终，隐藏状态由更新门控制产生。

LSTM 和 GRU 可以帮助模型捕捉长期依赖关系，从而提高神经网络在处理序列数据时的性能。这是两种主流的递归神经网络架构，在第六章中，本文用它们来评估提出来的新模型。

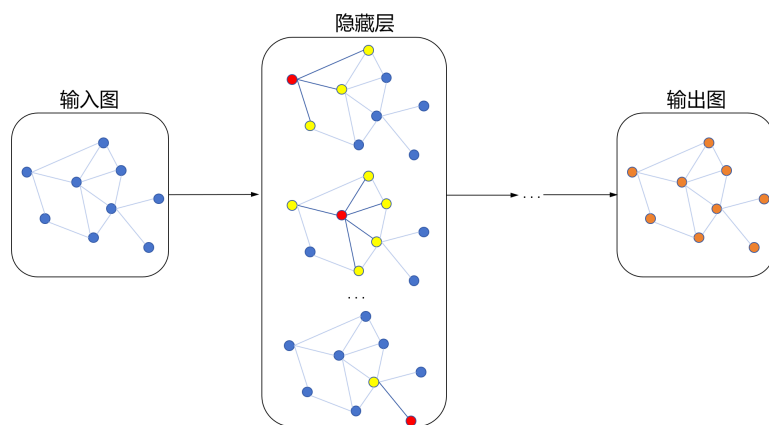


图 2-5 图神经网络结构。在计算中，图中每个节点的嵌入向量都由与之相邻的节点进行更新，最终图神经网络输出每个节点更新多次后的嵌入向量。

2.5 图神经网络结构

2.5.1 消息传递模型

图神经网络（Graph Neural Networks, GNNs）是一类用于处理图结构数据的深度学习模型。它们可以有效地捕捉图中节点之间的关系和拓扑结构，并用于各种任务，如节点分类、图分类、3D 任务。本文重点关注 3D 任务中基础图神经网络的建设，如图2-5所示，这一类网络也被称为消息传递神经网络（Message Passing Neural Networks, MPNN），它的核心思想是通过消息传递机制在图上进行信息的计算和聚合，以实现节点和图的表示学习。具体来说，图神经网络分为几个组成部分：

- 消息传递过程：图神经网络的核心操作是消息传递过程。在消息传递过程中，每个节点通过与其相邻节点的特征提取，收集来自邻居节点的消息，并用于更新自身的表示。

消息传递过程通常包括两个步骤：消息计算和消息聚合。在消息计算步骤中，图中每个节点将其自身的信息与邻居节点的信息结合，并生成一个消息向量。生成消息向量的函数通常采用神经网络中的全连接层。在消息聚合步骤中，节点会聚合所有收到的消息，生成一个汇总的消息向量，用于更新节点的表示。

- 节点更新：在收集到消息后，每个节点会使用汇总的消息向量来更新自身的表示。这个更新过程可以通过神经网络中的全连接层、卷积层等操作来

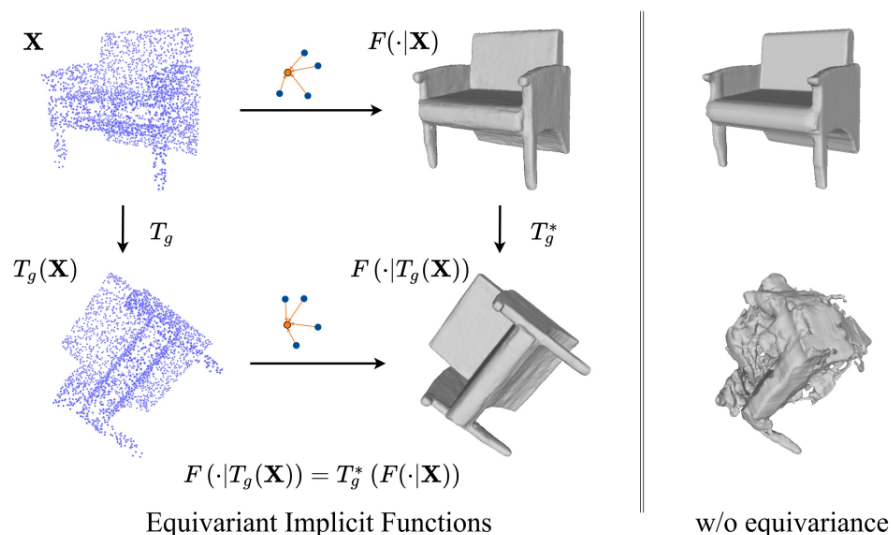


图 2-6 等变和非等变下 3D 特征的变化^[34]。 $F(\cdot|T_g(\mathbf{X}))$ 表示一种等变的转换过程。等变性可以保证变化前特征细节的可追溯性，但是非等变性会打破这种可追溯的特性。

完成，以实现节点表示的非线性转换和更新。

- 图级别操作：除了节点级别的操作外，图神经网络还可以执行图级别的操作。例如，对于图分类任务，可以在消息传递过程之后，对所有节点表示进行汇总，生成整个图的表示。这个图级别的表示可以通过全局池化操作或者其他聚合操作来实现。
- 可变图结构：图神经网络能够处理具有可变图结构的数据，即图中的节点和边的数量和连接关系可以随着输入数据的变化而变化。这种灵活性使得图神经网络可以处理各种图结构数据。

总的来说，图神经网络通过消息传递机制在图上进行信息的计算和聚合，以实现节点或者图的表示学习。其灵活性和适用性使得它成为图领域中的重要模型之一。

2.5.2 图神经网络中的等变性和不变性

大部分 3D 任务都需要模型提取的特征具有 $SO(3)$ 变换等变性质或者不变性质。广义的等变性定义如下：

给定一个群 \mathbf{G} 和变换参数 $g \in \mathbf{G}$ ，若函数 $\phi : \mathcal{X} \rightarrow \mathcal{Y}$ 满足以下等变性条件，

则称其对于 g 是等变的:

$$T'(g)[\phi(x)] = \phi(T(g)[x]), \quad (2-12)$$

其中 $T'(g) : \mathcal{Y} \rightarrow \mathcal{Y}$ 和 $T(g) : \mathcal{X} \rightarrow \mathcal{X}$ 分别表示 \mathcal{Y} 和 \mathcal{X} 上对应的变换。不变性是等变性的一种特殊情况。这种情况下, $T'(g)$ 是一个恒等变换, 表示 ϕ 的输出不受施加到输入上的变换的影响。本文主要关注 $SO(3)$ 变换下的等变性和不变性, 它是 3D 任务的重要思想。举个例子, 在点云分类任务中, 目标类别不会根据输入点云在空间中的旋转而变换, 表现出一种严格的 $SO(3)$ 变换不变性; 在分子任务中, 图神经网络需要计算每个几点所受到的力, 即一个三维矢量。该矢量会随着输入的旋转变换而发生相同的旋转变换, 表现出一种严格的 $SO(3)$ 变换等变性。一般来说, 等变性可以包含不变性。图2-6展示了等变和非等变下三维特征的变化, 可以看出, 当模型所代表的隐藏函数不等变时, 模型的输出会受到空间 $SO(3)$ 变换的干扰。

2.5.3 球谐函数

最近几年, 球谐函数逐渐成为构建等变模型的重要工具。本文第四章也用到了球谐函数相应知识, 因此, 本章对球谐函数进行简单介绍。

为了方便表示。本章用 S^2 表示单位球, 其中球坐标 (θ, φ) 分别表示极角和方位角。符号 \mathbb{R} 表示实数集, 而 \mathbf{R} 表示三维向量的旋转矩阵。 $SO(3)$ 表示特殊正交群, 即三维旋转群。

球谐函数 (Spherical Harmonics) 是一类定义在球面 S^2 上的函数, 它们构成一组正交基, 并且具有一些特殊的代数性质, 在等变模型中被广泛使用^[35-36]。本文使用实值球谐函数, 表示为 $Y_m^l : S^2 \rightarrow \mathbb{R}$, 其中 l 和 m 分别表示主量子数 (后文表示为阶数, degree) 和磁量子数 (后文表示为次数, order)。已知, 任何定义在 S^2 上的平方可积函数都可以通过球谐基表示为

$$f(\theta, \varphi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l f_m^l Y_m^l(\theta, \varphi), \quad (2-13)$$

其中 f_m^l 是 Fourier 系数。对于任意带有方向 (θ, φ) 的向量 \mathbf{r} , 可以定义一个球谐函

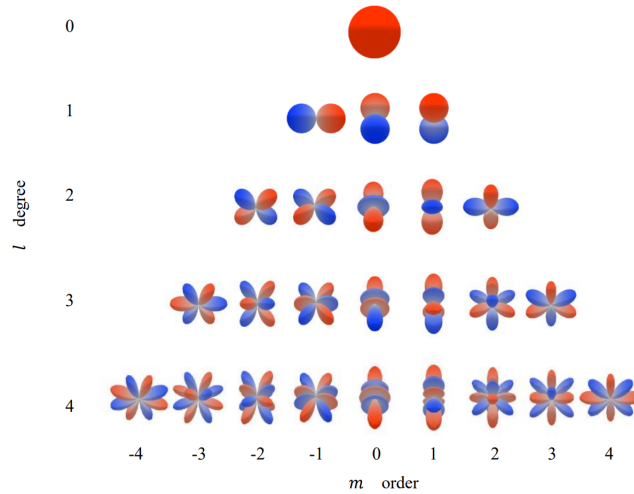


图 2-7 球谐函数在球面上的可视化展示。

数基底向量 $\mathbf{Y}^l(\theta, \varphi) = [Y_{-l}^l(\theta, \varphi); Y_{-l+1}^l(\theta, \varphi); \dots; Y_l^l(\theta, \varphi)]^T$ ，它是一个具有 $2l + 1$ 个元素的向量。

为了进一步理解球谐函数，本章罗列出一些球谐函数基底的具体数学形式：

$$\begin{aligned}
 Y_0^0(\theta, \varphi) &= \frac{1}{2} \sqrt{\frac{1}{\pi}} \\
 Y_{-1}^1(\theta, \varphi) &= \frac{1}{2} \sqrt{\frac{3}{2\pi}} \sin \theta e^{-i\varphi} \\
 Y_0^1(\theta, \varphi) &= \frac{1}{2} \sqrt{\frac{3}{\pi}} \cos \theta \\
 Y_1^1(\theta, \varphi) &= \frac{-1}{2} \sqrt{\frac{3}{2\pi}} \sin \theta e^{i\varphi} \\
 Y_{-2}^2(\theta, \varphi) &= \frac{1}{4} \sqrt{\frac{15}{2\pi}} \sin^2 \theta e^{-2i\varphi} \\
 Y_{-1}^2(\theta, \varphi) &= \frac{1}{2} \sqrt{\frac{15}{2\pi}} \sin \theta \cos \theta e^{-i\varphi} \\
 Y_0^2(\theta, \varphi) &= \frac{1}{4} \sqrt{\frac{5}{\pi}} (3 \cos^2 \theta - 1) \\
 Y_1^2(\theta, \varphi) &= \frac{-1}{2} \sqrt{\frac{15}{2\pi}} \sin \theta \cos \theta e^{i\varphi} \\
 Y_2^2(\theta, \varphi) &= \frac{1}{4} \sqrt{\frac{15}{2\pi}} \sin^2 \theta e^{2i\varphi},
 \end{aligned} \tag{2-14}$$

这些球谐函数在球面上的表示如图2-7所示。球谐函数可用于编码方向^[37-38]，并将频域中的表示映射到空间域上的信号^[36,39]。

球谐函数的一个常用属性是，对于任意的 $\mathbf{R} \in SO(3)$ ，满足以下方程：

$$\mathbf{Y}^l(\mathbf{R}\mathbf{r}) = \mathbf{D}^l(\mathbf{R})\mathbf{Y}^l(\mathbf{r}), \tag{2-15}$$

其中 $\mathbf{D}^l(\mathbf{R})$ 是一个 $(2l + 1) \times (2l + 1)$ 的矩阵，称为 Wigner-D 矩阵，其阶数为 l 。因此， \mathbf{R} 和 $\mathbf{D}^l(\mathbf{R})$ 对应于 (2-12) 中的 $T(g)$ 和 $T'(g)$ 。为了方便理解，本章将 Wigner-D 矩阵和球谐函数之间的等变性关系可视化到图2-8中， $\mathbf{Y}^l(\mathbf{r})$ 可由相同阶数 l 的 Wigner-D 矩阵进行旋转。之前的一些工作^[40-41] 约定：可由 Wigner-D 矩阵的在 $(2l + 1)$ -维向量空间旋转到向量称为 Type- l 向量，对应的空间称为 Type- l 空间。所有的 Type- l 向量以及 Wigner-D 都是一种不可约表示。后文中沿用这种表述方式。

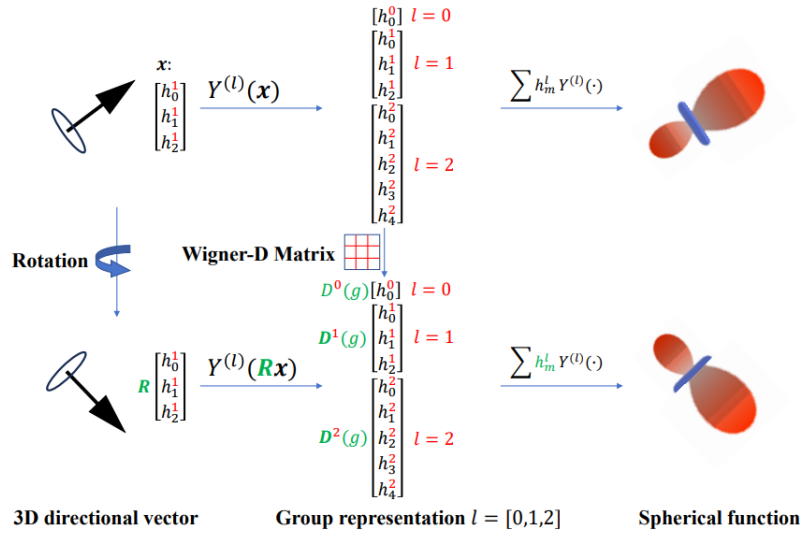


图 2-8 将 3 维空间中的旋转投影到高维空间。左侧表示一个三维空间的方向向量和旋转后的向量；中间表示球谐函数表示下的高阶等变向量；右侧表示球谐函数投影到球坐标系的函数形式。

2.5.4 等变神经元模型

构建等变图神经网络需要保证模型中所有操作都是严格等变的。然而，MP 神经元是不等变的。如图2-6所示，不等变的结构所提取的特征是无法在高维空间中旋转的，从而影响 3D 任务中特征的应用。因此，在 2018 年，Thomas 等人提出了一种等变的基础计算单元^[42]，称为 TFN。由于该结构和神经元处于同一层次的建模，本文在后文中称其为等变神经元。TFN 基于球谐函数的表示，它

的输入和输出都是不可约等变表示。传统神经元的线性层在等变神经元中变为了等变线性层^[43]，这种结构只在阶数相同的 Type- l 向量之间进行线性加和，不破坏 Type- l 向量内部的信息。传统神经网络中的非线性部分变为了 CG 张量积 (Clebsch - Gordan Tensor Product, CG 张量积)^[44]，该操作是等变神经元的核心，能够从等变表示中学习到丰富的抽象特征。CG 张量积最初用于描述量子力学中的角动量耦合，目前成为设计等变网络的重要工具^[35,41-42]。

CG 张量积 \otimes 的定义如下，

$$(\mathbf{u} \otimes \mathbf{v})_m^l = \sum_{m_1=-l_1}^{l_1} \sum_{m_2=-l_2}^{l_2} C_{(l_1, m_1)(l_2, m_2)}^{(l, m)} \mathbf{u}_{m_1}^{l_1} \mathbf{v}_{m_2}^{l_2}, \quad (2-16)$$

其中 \mathbf{u} 和 \mathbf{v} 分别表示具有 l_1 和 l_2 阶的可转动表示。在 (2-16) 的右侧， C 表示 CG 系数，是一个稀疏张量，在满足以下条件时产生非零项，

$$|l_1 - l_2| \leq l \leq (l_1 + l_2). \quad (2-17)$$

基于 CG 张量积，当 \mathbf{u} 和 \mathbf{v} 分别为 Type- l_1 和 Type- l_2 的向量时，输出是一个 Type- l 的向量。此外，目前的等变神经元可能还包括其它一些等变操作，包括等变门操作^[45-46] 和等变归一化操作^[47]等。

2.6 神经元模型和神经网络技术研究进展

关于神经网络模型的研究主要分为两类：1. 网络架构的研究；2. 新型神经元模型的研究。网络架构的研究建立在神经元模型的基础上。神经网络模型已被研究超过半个世纪，研究人员开发出了多种多样的网络架构，前文提到的卷积，递归，图神经网络都属于网络架构层面的研究。然而，神经元模型的相关研究较少。目前新型神经元模型的研究主要被分为两类：以 MP 神经元为核心的第一代神经元和以脉冲神经元为核心的第二代神经元。

第一代神经元的相关研究。继 1943 年 Warren McCulloch 和 Walter Pitts 提出了著名的 MP 神经元模型之后，后世的科研人员完善它的细节并将 MP 神经元发扬光大。20 世纪 50 年代，Frank Rosenblatt 提出了感知器模型，这是第一个具有学习能力的神经元模型。感知器能够学习简单的线性决策边界，但受到了其局

限性的限制，例如无法解决线性不可分问题。在 20 世纪 70 年代末和 80 年代初，感知器模型被扩展为多层感知器，加上反向传播算法的发展，复杂的神经网络逐渐被提出。进入 21 世纪，MP 类的神经元被应用到各种复杂模型中，使神经网络从机器学习的众多技术中脱颖而出。

MP 神经元的结构简单，但却有易集成，拟合能力强大的优点。目前关于很多关于 MP 神经元的研究集中在它的数理性质上，其中近似和优化通常被认为是该研究中最重要的两个问题。

神经元的近似理论关注于研究神经元或者神经网络模型是否是满足万能逼近理论 (Universal Approximation)。该理论分析神经元模型具有足够的灵活性和表达能力，使其可以逼近某个函数群。目前最为主流的理论是：只要神经网络足够大并且具备足够的层数和参数，就能够以任意精度逼近任何连续函数^[48-50]。本文在后面的章节中也会用到类似的理论来分析神经元性质。

此外，研究人员还关注神经网络逼近特定函数所需的网络结构和参数数量，即逼近复杂度。例如，一些研究讨论了深度和宽度对神经网络逼近能力的影响^[51]，以及神经网络在维度灾难下的性能^[52]。

优化理论重点关注如何训练神经网络中的参数。到目前为之，研究人员已经从几个方向入手试图简化优化问题。首先，对目标函数施加较强的假设，例如假设函数是凸的或者多项式的。其次，通过减弱神经网络的激活函数来简化问题。最后，设计特殊的算法来进行优化，如基于张量分解、半空间、稀疏性以及核函数等方法。近年来，一些学者尝试从优化复杂度的角度来研究基于梯度下降或者随机梯度下降算法的神经网络优化理论，其中包括隐式正则化理论^[53]和间隔最大化理论^[54]等代表性成果。

第二代神经元的相关研究。MP 神经元模型是第一代神经元模型。随后提出的脉冲神经元模型 (Spiking Neural Model, SNN) 是第二代神经元模型^[17]，它是一种基于神经脉冲传递的模型，模拟了生物神经系统中神经元之间的信息传递方式。与传统的人工神经网络不同，脉冲神经网络的核心是模拟神经元在接收到足够的输入信号后发放脉冲的行为。这种传输模式使得 SNN 更贴近生物神经网络的工作原理。在脉冲神经网络中，神经元不再简单地输出连续值，而是在特定时刻生成脉冲信号，这种脉冲信号的频率和时序编码了信息。SNN 的学习算法通常涉及时间相关的处理，即考虑脉冲的时序性，而不仅仅是传统的输入输出对

之间关系。

脉冲神经网络有着广泛的应用前景，涵盖了感知处理以及时间序列分析等领域。同时，由于其与生物神经系统的相似性，SNN 也被广泛运用于神经科学和认知科学研究中，有助于更深入地理解生物系统的运作方式。然而，脉冲神经元模型在训练算法方面存在明显不足，使大体系模型无法有效训练。此外，脉冲神经网络受限于脉冲信号的形式，在真实应用中需要通过预处理过程将其转化为脉冲序列，这也导致目前 SNN 主要应用在静态数据上，如何更好地利用 SNN 在时序数据和连续输入流上的优势，也是当前研究的重点之一^[55]。

非生物启发的神经元或神经网络模型。由于神经元模型与生物神经网络系统相似，第一代和第二代神经元的构造都尝试用数理过程去建模生物神经元中的一些机制。然而，真实世界的生物神经网络较为复杂，很多机理仍然处于研究之中，研究人员很难用数学模型去模仿神经元行为。随着神经网络的发展，研究人员逐渐发现生物神经元以外的一些机制也对神经网络的发展有促进作用，注意力机制 (Attention mechanism)^[56]就是其中的典型。注意力机制模仿了人类在观察事物时会有关注重点的行为。目前，注意力机制被广泛应用在视觉和自然语言处理任务中。此外，一些研究还将物理模型应用到神经网络中。Hopfield 网络^[18]就是一种受到物理系统中能量最小化原理启发的神经网络模型，它能够存储和恢复稳态，并被用于解决优化问题。玻尔兹曼机 (Boltzmann Machine)^[19]是一种基于能量的随机生成式神经网络模型，它受到了统计物理学中玻尔兹曼分布和马尔可夫链的启发，可以用于模拟大量随机变量之间的相互作用。最近几年，量子计算发展迅猛，一些研究人员也考虑将量子计算系统与神经网络相结合。在基于量子启发的神经元模型中，神经元的处理单元通常被设计成类似于量子比特 (qubits) 的结构。这些“量子神经元”可以利用量子叠加和量子纠缠等特性来处理信息，从而提高神经网络的计算能力和效率。此外，该模型还可能利用量子计算中的其他概念，如量子门和量子算法，来实现更复杂的信息处理和学习任务。目前量子神经元主要受到量子计算瓶颈的限制。基于物理模型的神经元往往具有更强的物理合理性和数理解释性。本文从物理模型出发，探索更高效的神经元计算模型。此外，本文对神经元的研究集中在应用价值方面，期望设计能够替换传统 MP 神经元的模型，并应用在广泛的任务和模型中。

2.7 本章小结

本章从神经元角度出发，介绍了经典的神经网络架构的基础知识，包括全连接神经网络，卷积神经网络，递归神经网络，图神经网络，这些网络架构是后续章节研究重点。同时，本章也对一些衍生结构进行介绍，包括动态卷积，长期依赖递归结构，等变图神经网络等，他们中采用到的技术也会在后续章节频繁出现。最后，本章回顾了神经元模型的发展历程，对他们的起源和应用做了简单概括。本章的介绍会对后续章节的阅读提供相应帮助。

第三章 一种基于物理碰撞模型的通用神经元研究

3.1 背景介绍

人工神经网络 (Artificial Neural Network, ANN) 目前已被广泛应用于各种机器学习任务当中。研究者通过任务的的特性设计各式各样的神经网络架构, 例如卷积神经网络之于图像, 递归神经网络之于文本等。一些工作也从更高层次的网络架构出发设计神经网络, 例如跃层连接^[7], 长期依赖记忆网络^[22]等。这些研究多数是针对神经元的连接方法以及宏观的网络形态, 针对神经元层面的相关研究较少。目前大部分神经网络结构依然沿用传统的 McCulloch-Pitts (MP) 神经元 (第一代神经元) 结构。MP 神经元由简单的线性层和非线性变换组成, 表示为:

$$\mathbf{y} = F(\mathbf{w}\mathbf{x}), \quad (3-1)$$

其中 \mathbf{w} 是可学习权重, \mathbf{x} 和 \mathbf{y} 是输入和输出, $F(\cdot)$ 是激活函数。MP 简单灵活的结构使其容易集成为复杂的神经网络, 集成后神经网络可以表示复杂的数据分布。

尽管 MP 神经元的结构较为简单, 但是改造以 MP 神经元为核心的神经网络体系并不容易。Chiyuan Zhang 等人^[57]在 2017 年探讨了模型复杂性与泛化性能之间的关系, 并指出: “神经网络的泛化能力并不像传统机器学习理论所预测的那样随着模型复杂性的增加而线性下降”。一般来说, 当基础单元的非线性拟合能力更强大时, 集成后的神经网络可能能够拟合更复杂的函数或分布。但在实际情况下, 过分复杂的非线性形式可能会带来严重的过拟合风险: 模型记住了训练数据中的噪声和细微特征, 而不是学习到数据的真正潜在分布, 导致模型不能在未见数据上泛化。

基于上述挑战, 本章将对新型神经元模型进行探索。通过研究经典物理模型与信息传递模型中的关联, 本章提出了一种通用的神经元: IC (Inter-layer Colli-



图 3-1 一维弹性碰撞的物理模型。两物体碰撞后，它们的速度发生相应变化。

sion) 神经元。它具有比 MP 神经元更强的非线性表示能力，同时保留了 MP 神经元易集成，轻量性的特点。IC 神经元可以被广泛应用于各种网络框架，本章采用常见的全连接网络，卷积网络，循环网络来设计 IC 神经网络，并通过实验验证了它们的有效性。本章的贡献总结如下：

- 提出一种非 MP 的新型神经元 (IC 神经元)，可以应用到多数以 MP 神经元为核心的神经网络架构中；
- 分析了 IC 神经元的非线性形式，证明其非线性拟合能力优于 MP 神经元；
- 实验验证了 IC 神经元在多种神经网络架构都能取得优于传统 MP 神经网络的效果。

3.2 基于弹性碰撞原理的数学模型

本章提出的 IC 神经元受到物理中弹性碰撞模型的启发。在本章中，首先描述了一个符合动量和能量守恒定律的物理碰撞系统，然后基于该模型建模一个信息前馈的数学模型。

物理模型常常揭示了物理量的更新或转移过程，例如，弹性碰撞模型可以清晰地解释物体间动量，能量等物理量的转移。考虑一个不存在任何阻力的理想物理场景，两个物体在一维空间中移动，其中左方向 (-) 表示负方向，右方向 (+) 表示正方向。它们的质量分别为 m_1 和 m_2 。最初， m_1 位于 m_2 的左侧，它们的速度均为零。当 m_1 获得正向速度 v_1 朝向 m_2 运动时，根据能量守恒和动量守

恒定律，碰撞后 m_1 和 m_2 的速度分别为：

$$v'_1 = \left(\frac{2m_1}{m_1 + m_2} - 1 \right) v_1, v'_2 = \frac{2m_1}{m_1 + m_2} v_1. \quad (3-2)$$

整个过程如图3-1所示。碰撞过程发生后，物体 m_2 将始终向右移动 (+)，而物体 m_1 的运动方向取决于 m_1 和 m_2 之间的数值关系。在物理模型中，设计者通常需要通过一些方法需要观测物理行为后的结果。为了观察碰撞模型的结果，观测者可以在一维空间最右侧放置一个观测平台来测量速率。因此，碰撞模型可能会观测到两种可能的结果：1. 物体 m_2 以速率 $|v'_2|$ 通过观测平台，未检测到物体 m_1 ；2. 物体 m_1 以速率 $|v'_1|$ 通过观测平台，物体 m_2 以速率 $|v'_2|$ 通过观测平台。

本章观察到，上述碰撞模型与信息前馈模型有共同之处。碰撞模型记录的是物体速率或动量的变化，而这些变化依赖于物体的质量 m_1 和 m_2 ，属于模型中内置的参数。在传统神经网络模型中，信号通过固定数学模型实现变换，这些模型也可以通过一些内置的参数来调整数学模型，类似于质量 m_1 和 m_2 的作用。因此，上述物理模型可以被视为一个信息前馈系统。为了简化该系统的输出，不同物体的输出速度合并成一项

$$v'' = \sigma(v'_1) + v'_2, \quad (3-3)$$

其中， $\sigma(\cdot)$ 表示 ReLU 函数，用于判断 m_1 是否通过观测平台。在物理中，不同物体的速率累加没有具体物理意义，但在信息学中，信号之间可以通过累加来简化输出信号的形式^[58]。将符号转化为神经网络常用符号后，方程(3-3)转换为

$$y = \sigma((w - 1)x) + wx. \quad (3-4)$$

x 和 y 表示神经元的输入和输出，其中输入对应物理碰撞模型的初始速度 v_1 。权重 $w = \frac{2m_1}{m_1 + m_2}$ 与物体的质量相对应。方程(3-4)的特点是选择性保留了碰撞物体 m_1 在碰撞后所携带的信息量。如果不考虑碰撞后的物体 m_1 ，方程(3-4)会直接退化为 MP 神经元的形式。此外，方程(3-4)是一种初步的单输入单输出神经元，不能直接用于构建神经网络。下一节将介绍如何将方程(3-4)拓展为通用的神经元模型。

3.3 基于物理碰撞模型建模的 IC 神经元

3.3.1 IC 神经元的数学模型

神经网络模型是由若干神经元模型连接组成，机器学习中绝大多数的网络架构都要求神经元具有接受多维输入的性质。本章将方程(3-4)的结构应用到输入中的每一维，并在计算输出时累加所有结果。其次，本章依照传统 MP 神经元的设计技巧，引入偏置项 (Bias) 和激活函数 (Activation Function)。基础 IC 神经元的数学形式表示为

$$\begin{aligned}
 y &= f\left(\sum_{i=1}^n w_i x_i + \sigma\left(\sum_{i=1}^n (w_i - 1)x_i + b_1\right) + b_2\right) \\
 &= f\left(\sum_{i=1}^n w_i x_i + \sigma\left(\sum_{i=1}^n w_i x_i - x_{sum} + b_1\right) + b_2\right).
 \end{aligned}
 \tag{3-5}$$

上式中 $f(\cdot)$ 用于表示激活函数，包含 Sigmoid 或 ReLU 等函数。 b_1 和 b_2 是两个独立的偏置，用于调整模块分布的中心。 $x_{sum} = \sum_{i=1}^n x_i$ 项表示所有输入特征的总和。在碰撞模型向神经元演化的过程中，两物体质量 m_1 和 m_2 都被归纳到 w_i 一项中。因此，在应用中只用将 w_i 定义为可学习的参数。相比于 MP 神经元 $f(\sum_{i=1}^n w_i x_i + b)$ ，IC 神经元只增加了一项偏置参数。图 3-2 展示了 IC 神经元和 MP 神经元结构上的对比。

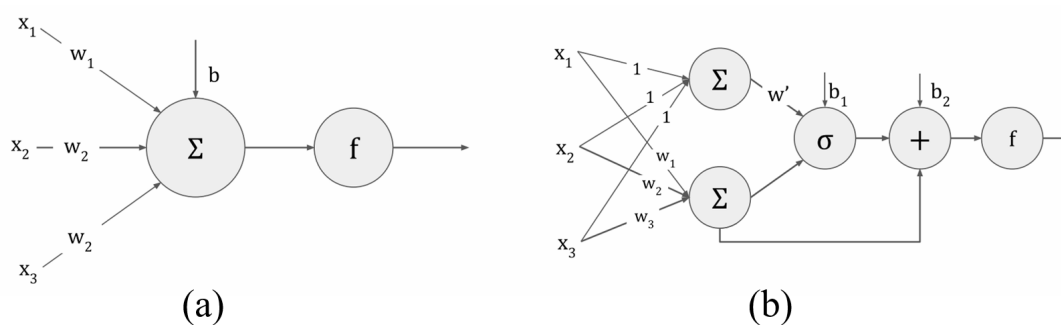


图 3-2 神经元结构对比。(a) MP 神经元；(b) IC 神经元。

3.3.2 信号复用机制

前文提到在 IC 神经元的-一个特点是选择性对物体 m_1 碰撞后的信号进行复用。一些神经网络的研究表明将浅层的信号引用到深层模块能有效提升性能。以

著名的 ResNet^[7]为例，它将浅层信号直接恒等映射到深层模块，这种方法可以为深层模块提供一些浅层信号的原始细节，从而提升模型对关键特征的学习能力。然而，IC 神经元的信号复用原理与跃层连接并不相同，具体来说，一些工作^[59]指出跃层连接必须跨越多层，单层的跃层连接并不能提升图像分类的准确率。换言之，简单地将 MP 神经元的输入合并到输出信号上并不会提升模型的性能。

IC 神经元的工作原理更类似于信号筛选。第3.2章构建的物理场景通过控制 m_1 和 m_2 之间的数值关系来决定物体 m_1 是否继续向观测者感兴趣的方向移动，这可以看作是一个筛选过程，神经网络设计者总是希望神经元的输出携带关键信息，并抑制无关信息^[10]。回到 IC 神经元， $\sigma(\sum_{i=1}^n (w_i - 1)x_i + b_1)$ 可以被认为是发生碰撞后物体 m_1 剩余的信息量。通过 $\sigma(\cdot)$ 可以选择是否将这部分信息传到后续模块中。

综上所述，IC 神经元本质上不同于跃层链接，它复用信号的目的是希望引入一些额外的信息量，而这些信息有可能有助于神经元学习更复杂的输入模式。下一节会从非线性表达能力的数学角度进一步介绍 IC 神经元带来的优势。

3.3.3 基于 IC 神经元的非线性表达能力分析

神经元中的非线性使得神经网络可以表示复杂的连续函数。本章对 MP 神经元和 IC 神经元的非线性表达能力进行分析。为了简化表述，下文中省略了偏置项。MP 神经元的非线性取决于激活函数 f ，而 IC 神经元的非线性还来自复用的信号。假设 $H = \sum_{i=1}^n (w_i - 1)x_i$ 表示一个切分 n 维输入空间的超平面，即 $H = 0$ ，则方程 (3-5)可以根据超平面 $H = 0$ 划分为两部分：

$$y = \begin{cases} f(2 \sum_{i=1}^n w_i x_i - \sum_{i=1}^n x_i) & \text{if } H \geq 0 \\ f(\sum_{i=1}^n w_i x_i) & \text{if } H < 0. \end{cases} \quad (3-6)$$

表达能力是反映模型拟合复杂分布或函数的能力。直觉上来讲，IC 神经元比 MP 神经元具有更强的表达能力，因为它在激活操作 $f(\cdot)$ 之前可以生成两种不同的线性表示，而 MP 神经元只包含一种。非线性过后，IC 神经元能够拟合一些 MP 神经元不能拟合的函数类型。

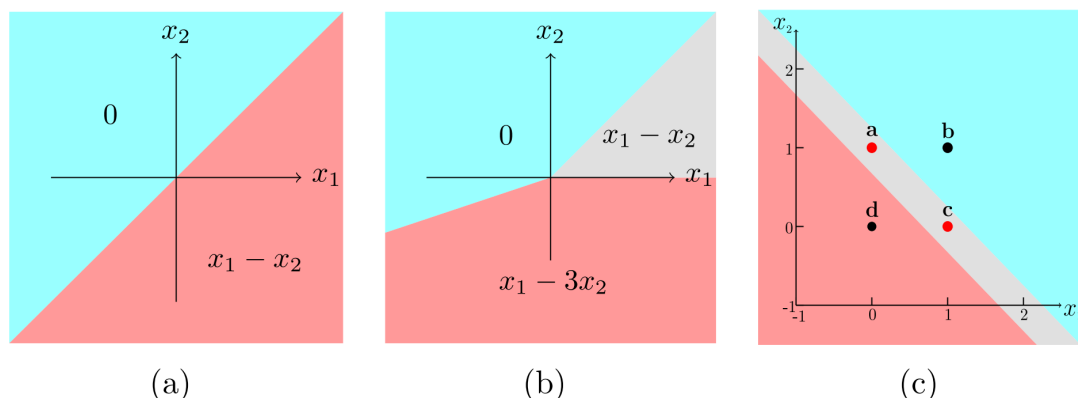


图 3-3 神经元在二维空间的非线性表示。(a)MP 神经元；(b)IC 神经元；(c) 基于 IC 神经元的一种 XOR 问题解决方案。

为了直观地区分两种神经元之间的差异，本章使用 2-D 数据 $([x_1, x_2])$ 作为输入， ReLU 作为激活函数 $f(\cdot)$ 来展示两种神经元表示的非线性分布。图 3-3(a, b) 显示二维输入下两种神经元的非线性表示。这种非线性将输入空间划分为多个子空间，其中每个子空间代表一个线性函数或分布。通过图 3-3(b) 可以发现，单个 IC 神经元引入额外非线性模式，这是一种更具表达能力的形式，能够解决一些更复杂的非线性任务。以异或问题 (XOR) 为例，XOR 是典型的线性不可分问题，将其映射到 2 维平面上，XOR 问题转化为区分 $(1, 0)$, $(0, 1)$ 和 $(1, 1)$, $(0, 0)$ 。为了方便理解，本章用 ReLU 函数作为激活函数 $f(\cdot)$ 。MP 神经元的非线性表示如图 3-3(a) 所示 (直线分割的线性空间)，XOR 问题中的四个点需要被分成至少三个子空间，显然 MP 神经元无法将四个点区分开，解决异或问题至少需要两个以上的 MP 神经元。然而，IC 神经元为用单个神经元解决 XOR 问题提供了可能性。图 3-3(c) 给出了使用单个 IC 神经元的解决方案，将平面划分为三个子空间，其中 $(1, 0)$, $(0, 1)$ 所在的空间可以由一种线性函数表示。 $(0, 0)$ 和 $(1, 1)$ 所在的空间可以用两种线性函数表示，通过调节权重可以使两个点的函数表示相似。

本章也给出解决 XOR 问题中 IC 神经元的权重。二维输入的 IC 神经元表示为 $f(x_1, x_2) = \sigma(w_1x_1 + w_2x_2 + b_1 + \sigma((w_1 - 1)x_1 + (w_2 - 1)x_2 + b_2))$ ，其中的参数为 $w_1 = w_2 = 0.2805$, $b_1 = -0.3506$, $b_2 = 0.6463$ 。该权重由 Adam 优化器训练 10 轮得到。在权重随机初始化的情况下，10 次重复实验全部可以成功解决 XOR 问题。综上所述，单个 IC 神经元比单个 MP 神经元具有更强的表达能力。当多个神经元组合成一个神经网络整体时，表达能力优势会被进一步放大。然而，非线性能力的分析也反映出了 IC 神经元的弊端：非线性函数缺乏灵活

性。在图3-3(a)表示的 MP 神经元中，非线性边界可以根据权重灵活变换。然而，在 IC 神经元中，三条非线性边界由同一组权重 $[w_1, w_2]$ 决定。他们之间存在约束，不能灵活调节，这可能会导致 IC 神经网络表现出过拟合现象。为了解决该问题，下一节将对基础 IC 神经元的结构进行进一步优化。

3.3.4 基于可调节超平面的 IC 神经元模型

为了解决 IC 神经元中非线性函数不灵活的问题，本章提出一种方法放松三条非线性边界之间的约束。前文提到基础 IC 神经元由超平面 $H = 0$ 将输入空间划分为两部分，然后在不同的空间内执行不同线性变换，从而增加最终神经元能够表达的模式数量。其中，超平面是由 $H = \sigma \left(\sum_{i=1}^n (w_i - 1)x_i + b_1 \right)$ 决定。本章将常数项 1 改为一可调节的参数 w' ，则整个 IC 神经元变为：

$$\begin{aligned} y &= f \left(\sum_{i=1}^n w_i x_i + \sigma \left(\sum_{i=1}^n (w_i - w') x_i + b_1 \right) + b_2 \right) \\ &= f \left(\sum_{i=1}^n w_i x_i + \sigma \left(\sum_{i=1}^n w_i x_i - w' x_{sum} + b_1 \right) + b_2 \right), \end{aligned} \quad (3-7)$$

w' 被称为调节权重，它可以被视为一个神经元的内置权重，与输入无关。具体来说，对于单个的神经元，无论输入是多少维，调节权重只有一维。这种新结构称为可调节 IC 神经元，它不仅保留了信号复用的优点，而且在非线性表示方面具有更强的灵活性。

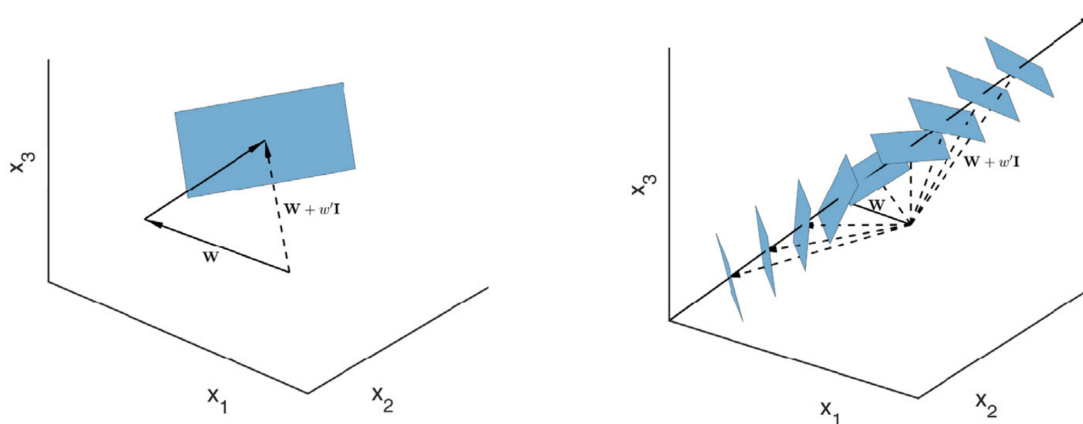


图 3-4 (a): 3-D 输入空间中的分割超平面 ($H = 0$); (b): 超平面根据调节权重 w' 进行旋转的过程。

图3-4(a)展示了三维空间中基础 IC 神经元超平面的计算。当输入权重 w 固

定时，超平面 $H = 0$ 与坐标系之间的角度也固定。在这种情况下，无法进一步优化超平面划分的子空间。在可调节 IC 神经元中，超平面的计算中有两个独立于输入权重 w 的参数： w' 可以用于改变超平面与坐标系的角度的，偏置 b_2 用于使超平面在整个空间平移。图3-4(a)展示了添加调节权重后超平面可以在三维空间中自由旋转。对于更高维度，本章给出 w' 的可调节范围的理论说明。

定理 3.3.1 通过调节调节权重 w' ，超平面 $\sum_{i=1}^n (w_i - w')x_i = 0$ 可以围绕向量 \mathbf{W} 和 \mathbf{I} 的叉积方向任意旋转。其中， $\mathbf{W} = (w_1, w_2, \dots, w_n)^T$ 表示所有输入权重组成的向量， $\mathbf{I} = \underbrace{(1, 1, \dots, 1)}_n^T$ 表示一个全一向量。 \mathbf{I} 的维度与 \mathbf{W} 相匹配。

证明 3.3.1 $\sum_{i=1}^n (w_i - w')x_i = 0$ 的法向量由 $\mathbf{H} = (w_1 - w', w_2 - w', \dots, w_n - w')^T$ 决定。 \mathbf{H} 和 \mathbf{I} 之间的角度可以由下式计算得到：

$$\cos(\theta) = \frac{\mathbf{H}^T \cdot \mathbf{I}}{|\mathbf{H}||\mathbf{I}|} = \frac{\mathbf{W}^T \cdot \mathbf{I} - Nw'}{\sqrt{N|\mathbf{W}|^2 - 2Nw'\mathbf{W}^T \cdot \mathbf{I} + N^2w'^2}}. \quad (3-8)$$

当 $\mathbf{W}^T \cdot \mathbf{I} \geq Nw'$ ：

$$\cos(\theta) = \sqrt{1 + \frac{(\mathbf{W}^T \cdot \mathbf{I})^2 - N|\mathbf{W}|^2}{N|\mathbf{W}|^2 - 2Nw'\mathbf{W}^T \cdot \mathbf{I} + N^2w'^2}}. \quad (3-9)$$

方程(3-9)表示连续递减函数，当 $w' \in (-\infty, \frac{\mathbf{W}^T \cdot \mathbf{I}}{n}]$ 时， $\cos(\theta) \in [0, 1)$ 。同理，当 $w' \in [\frac{\mathbf{W}^T \cdot \mathbf{I}}{n}, \infty)$ 时， $\cos(\theta) \in (-1, 0]$ 。因此，可以推论出：当 $w' \in (-\infty, \infty)$ 时， $\theta \in (0, \pi)$ 。显然，旋转轴的方向与 \mathbf{W} 和 \mathbf{I} 的叉积相同。通过调节 w' ，超平面 $H = 0$ 沿旋转轴方向旋转至任意角度。

定理 3.3.1 表明超平面 $\sum_{i=1}^n (w_i - w')x_i = 0$ 可以覆盖所有平行于 \mathbf{W} 和 \mathbf{I} 叉积方向的超平面。基础 IC 神经元不具有该性质。因此，在一些特殊任务中，MP 神经元可能会表现出比基础 IC 神经元更好的效果。然而，在充分训练的情况下，可调节 IC 神经网络总能取得不逊于 MP 神经网络的结果。因为 MP 神经元是可调节 IC 神经元的一种特殊情况。

命题 3.3.1 在封闭的输入空间内，对于任何 MP 神经元所表示的分布，都可以找到一种可调节 IC 神经元来完美表示该分布。

命题 3.3.1 可以被简单证明。在封闭空间内，无论输入权重 \mathbf{W} 取任何实数值，总能找到一组 w' 和 b_1 使得分割超平面完全与封闭空间相离，此时，IC 神经元退化为 MP 神经元。

综上所述，通过调整 \mathbf{W} 和 w' 之间的关系，IC 神经元既可以保留 MP 神经元的表达能力，又可以灵活拟合一些更复杂的非线性函数或分布。

3.4 IC 神经元结构在多种神经网络架构中的应用

3.4.1 IC 神经元与主流网络架构的结合

本章展示 IC 神经元在各种神经网络架构中的简单应用。需要注意的是，在实际应用中，需要考虑不同的网络架构或者下游任务的特性才能出发挥 IC 神经元最大的潜能，相应工作会在第四，五，六章节中进行介绍，本章只介绍最基础的 IC 神经元应用方法。

由于 IC 神经元具有将任意尺寸的输入转换为 1-D 输出的特点，本章将 IC 神经元集成到了最流行的神经网络结构中，例如全连接、卷积和循环结构。

全连接结构。首先，本章使用 IC 神经元构建一个全连接层。规定输入维度为 n ，神经元数量为 m ，第 j 个神经元的输出可以表示为：

$$y_j = f\left(\sum_{i=1}^n w_{ij}x_i + \sigma\left(\sum_{i=1}^n w_{ij}x_i - w'_j x_{sum}\right)\right), j \in [1, m]. \quad (3-10)$$

为了简化符号，这里和下面省略了偏置项。方程(3-10)可以通过矩阵和向量的形式重新定义为：

$$\mathbf{y} = f(\mathbf{W}\mathbf{x} + \sigma(\mathbf{W}\mathbf{x} - \mathbf{w}'\mathbf{I}\mathbf{x})), \quad (3-11)$$

其中 \mathbf{I} 被定义为一个 n 维的全 1 向量，用于计算 x_{sum} 。 \mathbf{W} 表示连接权重， \mathbf{w}' 是表示调节权重的向量。方程 (3-11) 表示基于 IC 神经元单个隐藏层。通过逐层连接 IC 神经元，IC 全连接 (IC-FC) 网络可以表示更复杂的非线性分布。

卷积结构。卷积结构使用核和滑动窗口来捕获局部区域的特征，将输入特征图 $\mathbf{X} \in \mathbb{R}^{H' \times W' \times C'}$ 映射到输出特征图 $\mathbf{U} \in \mathbb{R}^{H \times W \times C}$ 。为了简化符号，激活算子

被省略了。输出特征图可以根据通道切分为 $\mathbf{U} = [\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_C]$ ，其中

$$\mathbf{U}_i = \mathbf{W}_i * \mathbf{X}. \quad (3-12)$$

其中， $\mathbf{W}_i \in \mathbb{R}^{k \times k \times C'}$ 是一个滤波器核， $*$ 表示卷积算子。本章保留滑动窗口的特性，利用共享权重的 IC 神经元计算每个窗口中的特征。新的卷积核由 $[\mathbf{W}_i, w'_i]$ 表示，IC 卷积表示为：

$$\begin{aligned} \mathbf{U}_i &= [\mathbf{W}_i, w'_i] * \mathbf{X} \\ &= \mathbf{W}_i * \mathbf{X} + \sigma(\mathbf{W}_i * \mathbf{X} - w'_i \times (\mathbf{I} * \mathbf{X})), \end{aligned} \quad (3-13)$$

其中用于求和局部区域的 \mathbf{I} 是与 \mathbf{W}_i 大小相同的全一张量。由于由方程 (3-13) 定义的 IC 卷积层相比于传统卷积具有更强的非线性表达能力，它可以捕获更准确的局部特征，并且不改变输入和输出特征图的大小。通过堆叠这种结构可以构建 IC 卷积神经网络 (IC-CNN)。

递归结构。经典的递归神经网络使用前一个隐藏状态来计算当前单元的隐藏状态。给定一个输入序列 $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_T)$ ，递归神经网络通过迭代以下方程计算 h -维隐藏向量序列 $\mathbf{h} = (\mathbf{h}_1, \dots, \mathbf{h}_T)$ 和输出向量序列 $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_T)$ ：

$$\mathbf{h}_t = f(\mathbf{W}_1 \mathbf{x}_t + \mathbf{W}_2 \mathbf{h}_{t-1}), \mathbf{y}_t = \mathbf{W}_3 \mathbf{h}_t \quad (3-14)$$

其中 \mathbf{W} 表示权重矩阵， $f(\cdot)$ 表示激活函数，下标 $t \in [1, T]$ 表示第 t 次迭代步骤。本章使用 IC 结构来拟合当前输入信号上 \mathbf{x}_t ，隐藏向量 \mathbf{h}_t 的计算方式表示为：

$$\mathbf{h}_t = f(\mathbf{W}_1 \mathbf{x}_t + \sigma(\mathbf{W}_1 \mathbf{x}_t - w' \mathbf{I} \mathbf{x}_t) + \mathbf{W}_2 \mathbf{h}_{t-1}). \quad (3-15)$$

方程 (3-15) 集中于用 IC 神经元学习输入状态的特征，保留了传统递归神经网络对隐藏状态的处理方法。方程 (3-15) 构建的递归模型被称为 IC 递归神经网络 (IC-RNN)。

上述三种网络架构的应用被整理到图 3-5 中。总结来说，由于 IC 神经元具有和 MP 神经元相同连接方式，因此它可以被用到各种基于 MP 神经元的网络架

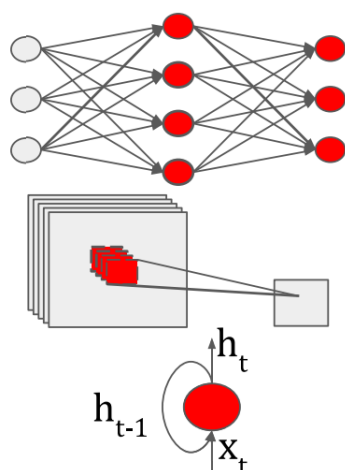


图 3-5 全连接、卷积和递归结构中的 IC 神经元。红色区域代表 IC 神经元的计算结构。

构。因此，除了上述三种网络架构，IC 神经元还可已被用到一些更高层次的网络架构，例如深度卷积模型和图神经网络等。在后面的大章节中，本文会进行深入的探索。

3.4.2 IC 神经网络复杂度分析

空间复杂度。本章比较了标准的 IC 神经元与 MP 神经元在三种网络结构中的性能。对于 IC 全连接和递归结构，参数的增加完全取决于调整权重 w' 和偏置 b_1 。假设有 m 个隐藏神经元和 n 维输入，IC 网络将增加参数从 $(n+1) \times m$ 到 $(n+3) \times m$ 。增加的比例为 $\frac{2}{n+1}$ ，可以看作是在隐藏层中增加了 2 个单元。卷积结构的结果类似，增加比例与卷积核的数量相关。实际应用中通常需要成百数千的神经元或卷积核，使 IC 神经元的增加的空间复杂度可以忽略不计。

计算复杂度。神经网络的计算复杂度主要来自于矩阵运算。在 IC 全连接和递归结构中，包含 $n \times m$ 乘法的 $\mathbf{W}\mathbf{x}$ 项被使用了两次，但只计算了一次。 $\mathbf{w}'\mathbf{I}\mathbf{x} = \mathbf{w}'(\mathbf{I}\mathbf{x})$ 中 $\mathbf{I}\mathbf{x}$ 项在整个隐藏层只需被计算一次， \mathbf{w}' 项需要 m 个乘法计算。IC 神经元隐藏层的计算复杂度为 $O(nm + m + n)$ 。与 MP 神经元隐藏层的计算复杂度 $O(nm)$ 相比，IC 神经元隐藏层增加了 $\frac{1}{n} + \frac{1}{m}$ 。对于 IC 卷积结构， $\mathbf{W}_i * \mathbf{X}$ 在每个卷积核的计算内只计算了一次。 $w'_i \times \mathbf{I}$ 在所有卷积核中只需要被计算一次。因此，额外的复杂度大约是传统卷积结构的 $\frac{1}{c}$ 。在构建 IC 神经网络时，本文不改变神经元模型外的操作，包括批量归一化 (BN)、池化等。当隐藏层数量或者卷积核数量

相对较大时，IC 神经网络的计算消耗与 MP 神经网络相似。

3.5 实验结果与讨论

本章将 IC 全连接神经网络，IC 卷积神经网络和 IC 递归神经网络应用到多种下游任务来综合测试 IC 神经元的有效性和泛用性。本章的实验既包含浅层的神经网络结构，也包含工业界广为使用的深度学习模型结构。除此之外，本章还评估了基础 IC 神经元的性能以研究调节权重 w' 的有效性。在下文中，基础 IC 神经元构成的模型被简写为 IC-FC-B、IC-RNN-B 和 IC-CNN-B。本章还在最后用一系列消融实验来分析 IC 神经元。

3.5.1 数据集和基线配置

本章实验的目标是比较 IC 神经网络和传统 MP 神经网络在各种机器学习任务中的性能。为了公平比较，在构建网络时使用完全相同的超参数来构建两种网络。具体来说，本章使用 ReLU 作为所有神经元的激活函数 $f(\cdot)$ ，Adadelata^[60] 作为优化器。调节权重 w' 也通过优化器来训练。在分类任务中，损失函数是交叉熵函数，即：

$$\mathcal{L}_{\text{CE}}(y, \hat{y}) = - \sum_i y_i \log(\hat{y}_i), \quad (3-16)$$

其中， y 是真实的标签向量， \hat{y} 是模型预测的标签概率向量。分类任务中的评估指标是 top-1 精度。在回归任务中，损失函数是平均绝对误差 (MAE)：

$$\mathcal{L}_{\text{MAE}}(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|. \quad (3-17)$$

回归任务中评估指标包括平均绝对误差和平均平方误差 (MSE)。同时，本章记录不同模型的 FLOPs 和参数量 (Params) 以分析 IC 神经网络占用的计算资源。训练中，每轮的结果都被记录下来以分析模型收敛的趋势。在一些任务中，验证集准确率波动较大。因此，本章对验证曲线进行了平滑处理，以更好地可视化测试准确率的趋势。每个实验重复三次，并记录最佳准确率，以减少偶然误差的影响。

在 IC 全连接神经网络的实验中。本章使用了多种常见分类任务来验证模型，对于每个任务，根据输入数据维度设置了神经网络层数和隐藏神经元的数量，具

体细节如下。

音乐分类。该任务使用 GTZAN 数据集，该数据集包含 1000 个音乐片段，均匀分为 10 个类别。每个片段存储在一个 30 秒长的文件中。数据集被分为 710 个片段用于训练和 290 个片段用于测试。此外，梅尔频率截谱系数 (MFCC) 特征、色度和其他一些特征被用来表示每个音乐片段，将原始声波转换为一个 253 维的特征向量。实验中构建了一个“输入-256-128-输出”结构的 IC-FC 网络。

手部动作识别。该任务使用 UCI sEMG 数据集，该数据集包含 1800 条记录，分为六种手部动作，即球形、尖端、掌部、侧面、圆柱形和钩状。sEMG 数据集是一个时间序列数据集，其中 EMG 传感器每秒捕获 500 个特征，作为时间序列中的一个单元。每条序列共包含关联 3000 个特征。实验中构建了一个“输入-1024-512-输出”结构的 IC 全连接网络来处理 sEMG 数据。此外，由于该数据集中包含显式的上下文关系，它也被用于后文中 IC 递归神经网络的评估。

情感分类。该任务使用 IMDB 数据集，该数据集包含 25000 条电影评论用于训练，另外 25000 条用于测试。评论数据经过预处理，提取了 tf-idf 特征，并裁剪为长度相等的 5000 维特征。实验中构建了一个“输入-1024-512-256-输出”结构的 IC 全连接神经网络。同 sEMG 数据集一样，IMDB 数据集包含上下文关系，它也被用于后文中 IC 递归神经网络的评估。

图像分类。该任务使用 MNIST 数据集。MNIST 包含 60K 个大小为 28×28 的训练图像和 10K 个测试图像。实验中构建了结构为“输入-256-128-输出”的 IC 全连接神经网络。在进入神经网络前，每个 MNIST 中的图像数据被拉展为一条 784 维的向量。MNIST 是标准的图像数据集，它也被用于后文中 IC 卷积神经模型的评估。

此外，本章还评估了 IC 网络在一些低维任务中的表现。这些任务使用 UCI YEAST、UCI ADULT 和 UCI LETTER 数据集上，它们的输入分别为 8/14/16 维。YEAST 有 1038/446 个训练/测试样本。LETTER 有 16000/4000 个训练/测试样本。ADULT 有 32561/16281 个训练/测试样本。实验中为本章 YEAST、ADULT 和 LETTER 数据集分别构建了“输入-32-16-输出”、“输入-32-16-输出”、“输入-64-32-输出”的 IC 全连接神经网络结构。需要注意的是，ADULT 数据的方差相对较大，因此本章实验在神经网络网络输入端引入了一个批归一化 (BN) 操作。

表 3-1 三种神经元在全连接神经网络中的分类效果。

Model	MNIST	IMDB	GTZAN	sEMG	LETTER	YEAST	ADULT
FC	98.02	87.59	54.14	40.00	93.25	59.19	84.99
IC-FC	98.42	88.28	56.21	41.67	94.00	61.88	85.20
IC-FC-B	98.37	88.13	55.17	41.11	93.50	60.54	85.14

3.5.2 IC 全连接神经网络实验结果分析

表 3-1 记录了 IC 全连接神经网络和相应的 MP 全连接神经网络在上述任务中的分类效果。从表中可以得到几个结论。首先，IC 网络在所有任务中都取得了优于 MP 神经网络的分类准确率。值得注意的是，在一些 MP 神经网络准确率不饱和的任务中，IC 网络可以取得明显的提升，例如 GTZAN（提升为 2.07%）、sEMG（提升为 1.67%）和 YEAST（提升为 2.69%）等。

在计算资源消耗方面，从表 3-2 中可以观察到，相比于 MP 神经网络，IC 网络只增加了很少的浮点运算数和参数量。FLOPs/Params 的最明显增加是在 YEAST（8.16%/9.09%）和 ADULT（8.65%/9.62%）这两个低维数据集上。它们的隐藏层单元较少，放大了调节权重和残差项对计算消耗的影响，这也与第 3.4.2 章中的分析一致。然而，在实际应用中，这种小型神经网络并不会造成难以承受的计算消耗。对于一些较大的模型，IC 神经网络产生的计算消耗和 MP 神经网络基本是相同的，这点在 IMDB 和 sEMG 数据集上的实验中得到了证明，FLOPs/Params 的增长率为 0.14%/0.06% 和 0.15%/0.08%。总而言之，IC 全连接神经网络在多个任务上表现出更好地泛化性能，且几乎不增加额外的计算负担，有望成为一种替换 MP 神经元的新型神经元。

在 IC 全连接神经网络的实验中，本章也对比了基础 IC 神经元 ((3-5)) 的性能。从表 3-2 中观察到基础 IC 神经元也能够所有任务中的取得优于 MP 神经网络的结果，这反映了第 3.3.1 章构造的信息复用结构是有效的。相比于可调节 IC 神经元，虽然基础 IC 神经元在计算量和参数量上有微小的下降，但他的分类结果不及可调节 IC 神经元的结果。在多数应用中，可调节 IC 神经元更适合作为神经网络模型的基础单元。

除了最终准确率之外，训练过程中的准确率也可以反映模型对未知分布的学习能力。IC 全连接神经网络和 MP 全连接神经网络的训练过程如图 3-6 所示，可以看出，IC 神经网络在较短的时间内达到收敛，尽管 IC 神经网络和 FC 神经

表 3-2 不同神经网络模型的 FLOPs/Params 对比。FLOPs (KMac) 表示前向传播的计算量。Params (K) 反映了可学习参数的数量。

Model	MNIST	IMDB	GTZAN	sEMG
MP-FC	235.14/235.15	5777.66/5777.67	99.20/99.21	3600.90/3600.92
IC-FC	236.56/235.91	5785.99/5781.25	100.09/99.98	3606.46/3603.97
IC-FC-B	236.18/235.53	5784.20/5779.46	99.71/99.59	3604.92/3602.44
Model	LETTER	YEAST	ADULT	-
MP-FC	4.00/4.03	0.98/0.99	1.04/1.04	-
IC-FC	4.18/4.22	1.06/1.08	1.13/1.14	-
IC-FC-B	4.08/4.12	1.02/1.03	1.09/1.09	-

网络都使用相同的初始化和优化器，但在最初的几轮中，IC 神经网络的精度上升速度明显高于 MP 神经网络。这个现象在 MNIST 和 IMDB 数据集中最为明显。IC 神经网络在这两个数据集上都表现出较快的收敛速度。此外，IC 神经网络和基础 IC 神经元组成的模型 (IC-FC-B) 收敛速度相似，反映了 IC 神经元快速收敛的能力可能来源于信息复用的操作。此外，可调节 IC 神经元的网络能够在收敛时达到更高的准确率。

3.5.3 IC 卷积神经网络实验及讨论

本章中对比了 IC 卷积神经网络和 MP 卷积神经网络的性能。卷积神经网络通常用于处理图像数据，本章用四个数据集来验证模型：MNIST，CIFAR10/100 和 ImageNet。其中 MNIST 和 CIFAR10 的数据集较简单，可以通过浅层卷积网络进行学习。ImageNet 是深度学习最流行的数据集之一，本章将 IC 神经元应用到深度模型以评估 ImageNet 上的表现。下文会对数据集和模型配置的细节进行介绍。

MNIST 数据集。 实验中用三个 3×3 的 IC 卷积层组成特征提取模块，它们的输出通道数分别为 32, 64, 64，步幅为 1, 2, 2。在 IC 卷积层捕获特征后，实验中使用一个具有“卷积特征维度-256-输出”的全连接网络结构作为分类器。

CIFAR 数据集。 CIFAR10 数据集包含 60K 个 $32 \times 32 \times 3$ 的 RGB 图像，分为 10 类，其中包括 50K 个训练图像和 10K 个测试图像。CIFAR100 数据集包含 60000 张 32×32 的彩色图像，分布在 100 个类别中，其中 50000 张图像用于训练，10000 张用于测试。两个数据集都被用于深度学习模型的实验中。这些实验

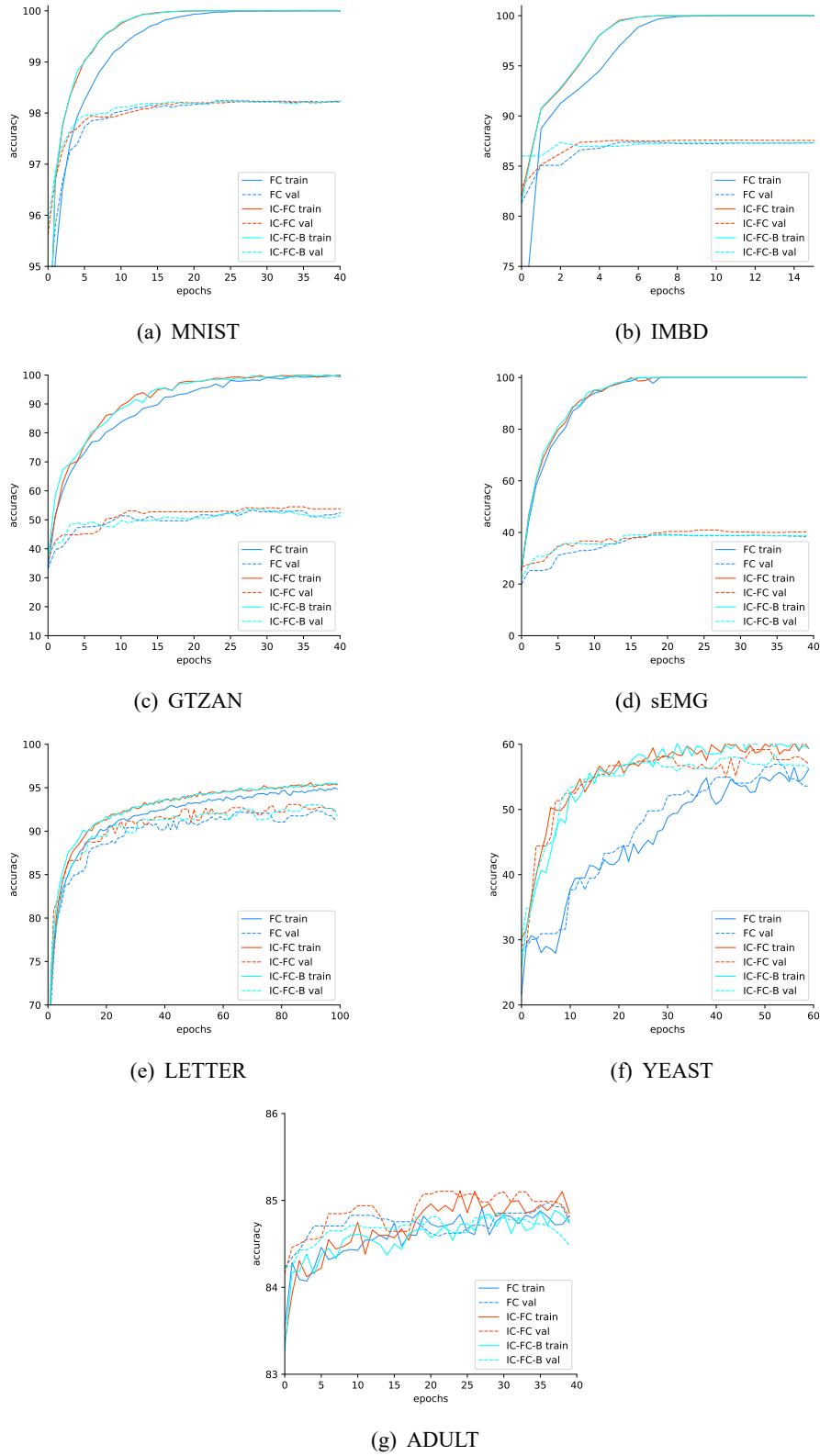


图 3-6 各个任务上 IC 全连接神经网络的训练曲线。

表 3-3 图像任务中 IC-CNN 和传统 CNN 的性能对比。

Model	MNIST (Acc.(%)/FLOPs/Params(k))	CIFAR10 (Acc.(%)/FLOPs/Params(k))
CNN	98.84/6491.20/861.39	81.25/9067.26/1107.72
IC-CNN	98.97 /6623.70/861.71	83.02 /9258.75/1108.05
IC-CNN-B	98.95/6582.93/861.55	82.82/9205.50/1107.88

中用到的 IC 卷积网络都是基于市面上流行的深度模型，包括 VGG^[20]，ResNet^[7]，DenseNet^[61]，MobileNet^[62]，SENet^[10]和 ResNeXt^[11]。除此之外，由于 CIFAR10 的图像较为简单，类别较少，本章在 CIFAR10 上增加一组浅层 IC 卷积神经网络的实验。浅层网络的结构与 MNIST 中的结构相同。

ImageNet 数据集。ILSVRC ImageNet 2012 分类数据集^[32]是深度学习中最常用的图像数据集之一。它包含超过 100 万张彩色图像，分为 1000 个类别，包括 128 万张训练图像和 5 万张验证图像。本章实验用它来验证 IC 深度卷积神经网络的性能。用到的深度模型包含 ResNet^[7]和 SENet^[10]。

浅层模型实验结果分析。为了评估 IC 神经元在卷积结构中的表现，浅层模型使用了传统的全连接层作为分类器，而不是 IC-FC 结构。表 3-3 中记录的 IC 卷积神经网络达到了更高的准确性。实验结果表明，IC 神经元能够促进卷积核捕获到更细粒度的局部特征。同时，表 3-3 也记录了不同模型的 FLOPs/Params。与 IC-FC 的实验相似，IC 卷积神经网络增加的计算消耗也基本可以忽略不计。

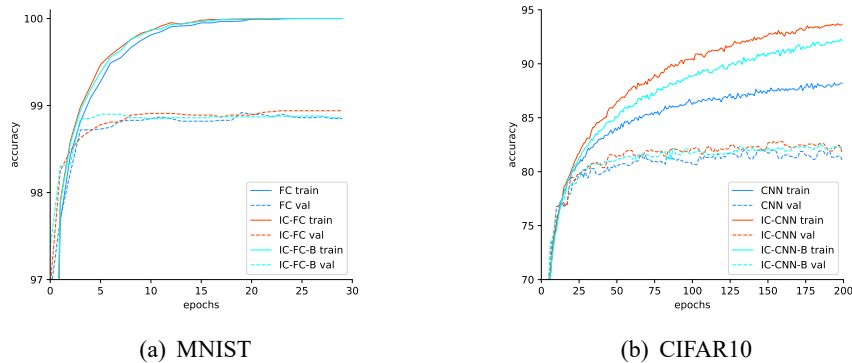


图 3-7 IC-CNN 和传统 CNN 在图像分类任务中的训练曲线。

图 3-7 中展示了浅层模型中的训练曲线，可以观察到 IC-CNN 具有比传统卷积网络更快的收敛速度。在 CIFAR10 的训练曲线上，验证集精度与训练集有明显差异，且 IC-CNN 进一步扩大了这种差距。这是由于浅层模型在 CIFAR10 中的过拟合现象导致的，而 IC 神经网络进一步放大了这种现象。

深度模型实验结果分析。浅层 IC-CNN 的实验结果表明 IC 神经元与卷积结构的结合能够提升传统卷积的性能。这为 IC 神经元在图像任务中的广泛应用提供了可能。为了进一步验证 IC 卷积神经网络的性能，本章实验将 IC 神经元应用到一些已经存在的深度卷积模型中，并用更复杂的图像任务验证其性能。新构造深度模型只替换卷积层为 IC 卷积，保留原模型的规范化，池化等操作。

实验首先在 CIFAR10/CIFAR100 上广泛验证不同的深度模型。在这些实验中，优化器采用随机梯度下降（SGD）方法^[63]，其中权重衰减率为 10^{-4} ，动量为 0.9。每个模型在批量大小为 128 的情况下进行 200 个周期的训练。学习率初始化为 0.1，在第 60 和第 120 个周期时降低 10 倍。每组实验重复三次取平均以避免偶然误差。

表 3-4 CIFAR 数据集上的实验结果。表中结果均为 Top-1 准确率 (%)。

Model	CIFAR10	CIFAR100	Model	CIFAR10	CIFAR100
VGG-16	93.64	72.93	ResNet-18	95.02	75.61
IC-VGG-16	93.90	73.35	IC-ResNet-18	95.20	75.95
DenseNet-121	95.41	77.01	ResNet-50	94.94	77.74
IC-DenseNet-121	95.52	79.15	IC-ResNet-50	95.12	78.59
MobileNet	90.08	65.98	SE-ResNet-18	94.92	76.44
IC-MobileNet	90.90	67.32	IC-SE-ResNet-18	95.06	76.60
ResNeXt	95.38	77.77	SE-ResNet-50	94.90	78.58
IC-ResNeXt	95.46	78.09	IC-SE-ResNet-50	95.02	79.18

对比表 3-4 中的 top-1 准确率，可以观察到 IC 网络的平均准确率超过了基线网络 0.24%/0.77%。尤其在 MoblieNet 基线上。MoblieNet 基线的准确率并没有像其他模型那样饱和，IC-MoblieNet 明显改善了 MoblieNet 的性能（从 90.08%/65.98% 提高到 90.90%/67.32%）。实验结果证明：即使这些深层架构已经具有较强的泛化能力，IC 神经元仍然能够提高性能。同时，实验也展示了 IC 结构在深度模型体系中的有效性和普适性。

本章实验在 ImageNet 上进一步验证 IC 神经元用于一些复杂视觉任务的可能性。优化器采用了带有权重衰减率为 10^{-4} 和动量为 0.9 的 SGD 方法。训练过程设置为 120 个周期，批量大小为 256。学习率初始设定为 0.1，并且每 30 个周期将减小 10 倍。每个实验只运行一次。基线模型的报告结果是在相同的设置下复现的，部分结果甚至超过了原始论文中报告的结果。

从表 3-5 中可以看出，IC 神经元可以提高具有不同深度的网络的性能。更

表 3-5 ImageNet 数据集的实验结果。

Model	Top-1 acc./Top-5 acc.	GFLOPs/Params
ResNet-50	76.15/92.87	4.12/25.56M
IC-ResNet-50	76.73/93.31	4.14/25.60M
ResNet-101	77.37/93.54	7.85/44.55M
IC-ResNet-101	77.90/93.88	7.88/44.65M
ResNet-152	78.43/94.11	11.58/60.19M
IC-ResNet-152	78.58/94.23	11.62/60.34M
SE-ResNet-50	77.63/93.64	4.13/28.09M
IC-SE-ResNet-50	78.02/93.88	4.15/28.13M
SE-ResNet-101	78.35/94.09	7.86/49.33M
IC-SE-ResNet-101	78.52/94.20	7.89/49.42M
ResNeXt-101	79.30/94.54	16.51/88.79M
IC-ResNeXt-101	79.52/94.78	16.56/88.99M

表 3-6 IC 卷积神经网络在 PASCAL VOC 2007+2012 测试集上的检测性能。

framework	backbone	mAP (%)
Faster R-CNN	ResNet-50	79.5
	IC-ResNet-50	80.1
Retinanet	ResNet-50	77.3
	IC-ResNet-50	78.0

重要的是，深度 IC 模型的额外 FLOPs/Params 可以忽略不计 ($< 0.5\%/0.3\%$)。通常在深度学习应用中，增加深度或宽度是一种常用的提升性能的方法。现在，IC 神经元提供了一种更有效的改善深度模型性能的方法。此外，众所周知，深度学习的一个缺点是其复杂的超参数设计^[11]。增加深度或宽度提升性能的方法面临多次试错，调参的困窘，而基于现有模型替换 IC 神经元可以绕过这一障碍。

为了进一步验证 IC 神经元在相关下游任务中的表现，本章将深度 IC 神经网络应用到目标检测中。数据集使用 PASCAL VOC 2007+2012 目标检测数据集^[64]，该数据集包含约 5K 个训练/验证图像和 5K 个测试图像，涵盖了 20 个对象类别。实验中使用了两种常见检测框架：Faster R-CNN^[65]和 Retinanet^[66]。在应用中，本章实验将骨干网络替换为同等规模的 IC 卷积神经网络，通过一次预训练和一次联合训练直接得到优化后的结果。所有配置与开源社区^[67]中描述一致。评估指标是均值平均精度 (mAP)。如表 3-6 所示，IC-ResNet-50 在 Faster R-CNN 和 Retinanet 框架中分别超过了 ResNet-50 1.0% 和 0.9%。这部分实验进一步验证了 IC 神经元的实际应用能力。

总而言之，深度学习实验表明，IC 可以直接应用于多种现存模型，它可以为模型设计者提供了一种简单，低成本的方法来构建合适的深度模型。

3.5.4 IC 递归神经网络实验及讨论

递归神经网络通常用于处理含有上下文信息的数据。本章实验构建了一些简单的 IC 递归模型来处理时间序列数据，包括前文提到的 sEMG 和 IMDB 数据集。使用前文提到的处理方法，sEMG 数据被处理成多个 EMG 数值特征，每个 EMG 特征都是一个 500 维向量，表示时间序列中的一个单元。实验中使用“输入-80-输出”的结构来构造 IC 递归神经网络的隐藏层。此外，在 IMDB 实验中，本章实验建立字典并用词嵌入操作将每个单词转换为一个 128 维向量。实验中使用“输入-128-输出”的结构来构造 IC 递归神经网络的隐藏层。此外，所有实验都基于相同的结构建立 MP 神经元的递归神经网络并进行训练。

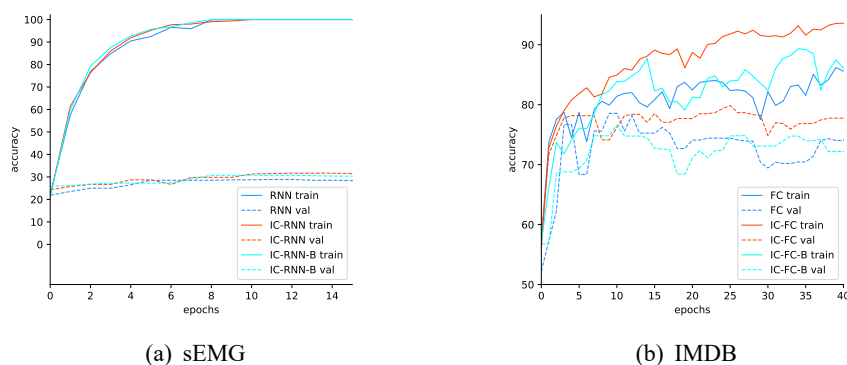


图 3-8 IC-RNN 和传统 RNN 在时间序列任务中的训练曲线。

IC 递归神经网络的实验结果如表 3-7 所示，IC-RNN 取得了更高的准确性，并在计算消耗上与基线模型基本保持一致。此外，图 3-8 也给出递归神经网络中的训练曲线。IC 神经元能在递归神经网络结构中取得更好的收敛能力和泛化性能。本章的实验只针对最基础的递归神经网络结构。在一些应用更广泛的循环结构，例如 LSTM 和 GRU 中，底层的构造单元也是 MP 神经元。本章介绍的 IC 神经元同样可以用于这些模型的性能提升。

表 3-7 时序任务中 IC-RNN 和传统 RNN 的性能对比。

Model	sEMG (Acc.(/FLOPs/Params(k))	IMDB (Acc.(/FLOPs/Params(k))
RNN	29.07/240.48/240.57	80.64/662.46/662.56
IC-RNN	31.48 /243.64/240.73	82.86 /672.91/662.69
IC-RNN-B	30.93/243.56/240.65	79.12/672.85/662.62

3.5.5 IC 神经元在 MP 衍生神经元体系中的应用

MP 神经元中的“线性 + 非线性”体系被广泛应用到多种后来的神经元体系中。本文推测其他 MP 神经元衍生神经元模型可能也会受益于 IC 神经元的思想，本章实验测试了 Pi-Sigma 神经网络 (PSNN)^[68] 和单一乘法神经元 (SMN)^[69] 模型。单树突神经元模型 (DNM) 和脉冲神经元 (SNU) 虽然也提出了广为流传的神经元结构，但他们不是 MP 神经元衍生的模型。

PSNN。 PSNN 是一个高阶神经网络，它使用隐藏神经元输出的乘积。其输出由以下公式给出：

$$y = f\left(\prod_{j=1}^H h_j\right), h_j = \sum_{i=1}^N (w_{ij}x_i + b_j), \quad (3-18)$$

其中， x_i 是输入， w_{ij} 和 b_j 是 PSNN 的权重和偏置。 h_j 表示隐藏层中的神经元， $f(\cdot)$ 表示适当的激活函数。 H 项是表示隐藏神经元数量的超参数。尽管有一些研究尝试优化隐藏神经元乘积的阶数和组合，或者优化训练方法^[70-71]，但对线性变换部分的改进较少被考虑。受 IC 神经元的启发，本章将信号复用的操作和调整权重 w' 整合到 h_j 的表示中：

$$h_j = \sum_{i=1}^n w_{ij}x_i + \sigma\left(\sum_{i=1}^n w_{ij}x_i - w'x_{sum} + b_{1j}\right) + b_{2j}, \quad (3-19)$$

其中， b_{1j} 和 b_{2j} 是两个独立的偏置。新的结构表示为 IC-PSNN。

SMN。 SMN 基于多项式结构，输出由以下方程得到：

$$y = f(u) \quad (3-20)$$

$$u = \prod_{i=1}^n (w_i x_i + b_i),$$

其中 x_i 是 SMN 的输入, w_i 和 b_i 是 SMN 的权重和偏置。f 表示激活函数。本章在 $(w_i x_i + b_i)$ 上应用了 σ 操作和调整权重 w' , 则 u 的公式改变为:

$$u = \prod_{i=1}^n (w_i x_i + \sigma((w_i - w')x_i + b_{1i}) + b_{2i}). \quad (3-21)$$

新的结构表示为 IC-SMN。通过引入 IC 神经元的思想, PSNN 和 SMN 可以利用多组线性变换提升最终非线性的模式数量。当多余的非线性对任务无用时, 新模型也可以通过调整权重 w' 和 b_1 项来返回传统的 SMN 和 PSNN, 类似与本文在第 3.3.3 章的论证。

实验结果分析。 本章实验使用了时间序列预测任务来评估应用了 IC 思想的 PSNN 和 SMN。任务建立在 Mackey-Glass (MG) 和澳大利亚啤酒消费 (AUST) 数据集上。MG 时间序列数据用于描述白血病患者非线性振荡产生的白细胞生成情况。MG 的延迟差分函数给定为:

$$y(t+1) = (1-b)y(t) + a \frac{y(t-\tau)}{1+y^{10}(t-\tau)}, \quad (3-22)$$

其中 $a = 0.2, b = 0.1, \tau = 17$ 。该实验中使用 $y(t)$ 、 $y(t-6)$ 、 $y(t-12)$ 、 $y(t-18)$ 来预测 $y(t+1)$ 。通过 MG 的延迟差分函数可以生成了 1000 个样本, 并将其均匀分为训练集和测试集。为了方便权重的训练, 数据集在 0.1 和 0.9 之间进行了归一化处理。PSNN 的隐藏神经元数量设置为 5。实验中使用 Adadelta 优化器在 500 个 epochs 中训练模型。图 3-9(a, b) 展示了 IC 模型和基本模型的预测结果。从图中可以观察到, PSNN、SMN 及其对应的 IC 模型可以近似预测 MG 的延迟差分函数。此外, IC 模型倾向于关注 MG 数据在峰值处的特征。

AUST 数据集是一个真实世界的时间序列数据集, 记录了从 1956 年到 1994 年澳大利亚的啤酒消费情况, 共有 154 连续时间段的样本。实验中使用前 100 个样本进行训练, 最后 54 个样本进行验证。实验配置与 MG 实验相同, 用 $y(t)$ 、 $y(t-1)$ 、 \dots 、 $y(t-6)$ 来预测 $y(t+1)$ 。图 3-9(c, d) 展示了 AUST 数据集上的预测结果, 可以观察到 IC 模型在 AUST 的低谷处, 如横坐标中 130 到 140 的区域, 展现出更好的预测结果。表 3-8 反映了两个数据集中 MSE 和 MAE 的结果。综合表 3-8 和图 3-9 中的结果, 可以得出结论: 引入 IC 神经元的思想可以在一些场景

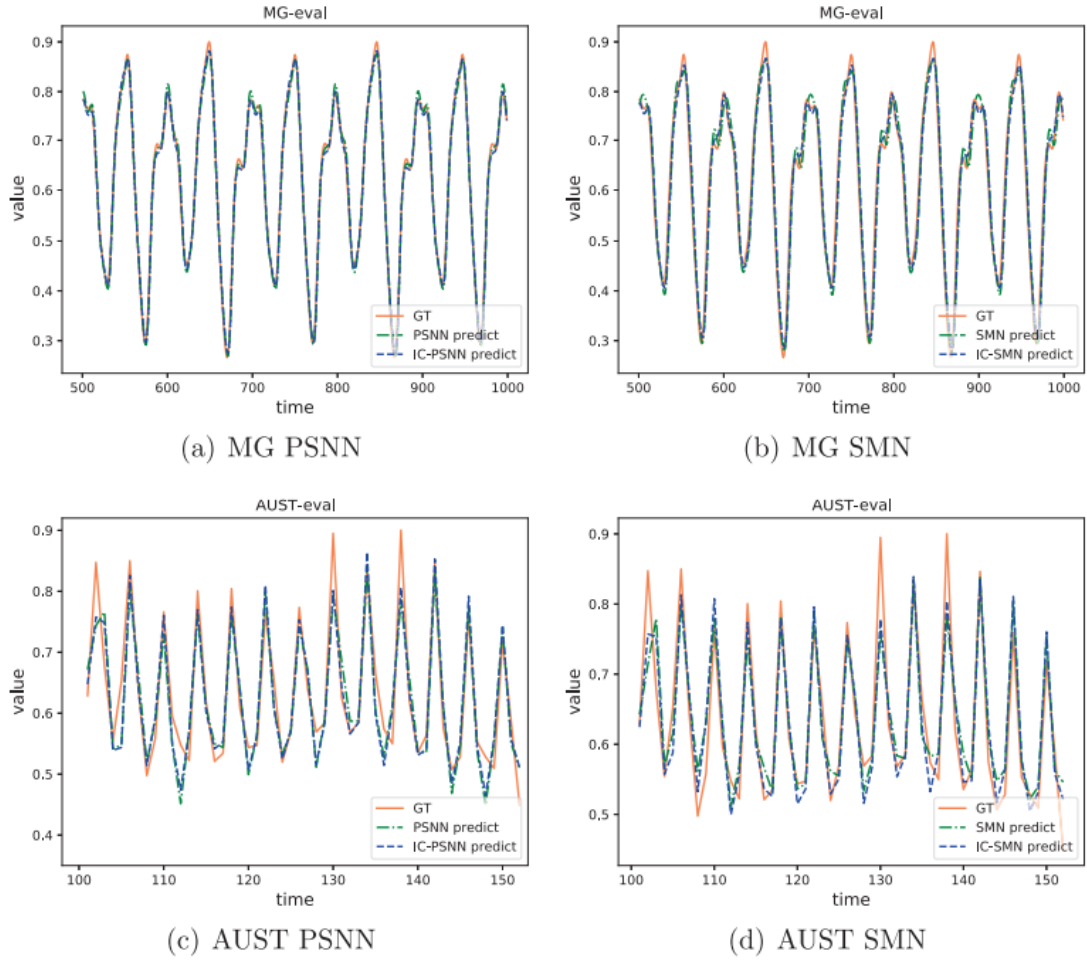


图 3-9 MG 和 AUST 时间序列数据集上的预测结果。

下增强 PSNN 和 SMN 对时序数据的表示能力。

本章实验进一步验证了 IC 神经元的泛用性。不止是 MP 神经元构建的神经网络，基于 MP 的神经元模型也能够通过 IC 神经元提升其性能。众多实验都表明，IC 神经元在大部分场景中都可以取代 MP 神经元以使模型达到更高的性能。

表 3-8 PSNN、SMN 及其相应的 IC 模型在 MG 和 AUST 时间序列数据集上性能的比较。

Dataset	Model	Testing MSE	Testing MAE	Model	Testing MSE	Testing MAE
MG	PSNN	0.0001	0.0096	SMN	0.0008	0.0234
	IC-PSNN	0.0001	0.0073	IC-SMN	0.0006	0.0205
AUST	PSNN	0.0024	0.0399	SMN	0.0023	0.0363
	IC-PSNN	0.0019	0.0339	IC-SMN	0.0017	0.0334

表 3-9 分类任务中 IC-FC, FC-A(更深的模型), FC-B(更宽的模型) 模型的对比。

Model	MNIST	IMDB	GTZAN	sEMG	LETTER	YEAST	ADULT
FC-A	98.55	87.98	55.52	40.37	93.56	60.54	85.14
FC-B	98.53	87.74	55.17	40.56	93.70	60.31	84.99
IC-FC	98.42	88.28	56.21	41.67	94.00	61.88	85.20

3.5.6 消融实验

模型容量消融实验。在实际应用中,提升神经网络模型容量可以从增加网络宽度(隐藏层数量)或者深度出发。本章中提出的 IC 神经元也能够提升模型容量。为了对比不同策略的优劣,本章用更宽或更深的 MP 神经网络来和表3-1中 IC-FC 的结果对比。MP 神经网络的结构如下:

- MNIST: 输入-512-1256-输出, 输入-256-256-128-输出
- IMDB: 输入-1500-1000-500-输出, 输入-1024-1024-512-256-输出
- GTZAN: 输入-512-256-输出, 输入-256-256-128-输出
- sEMG: 输入-1500-1000-输出, 输入-1024-1024-512-输出
- LETTER: 输入-100-50-输出, 输入-32-32-16-输出
- YEAST: 输入-100-50-输出, 输入-32-32-16-输出
- ADULT: 输入-100-50-输出, 输入-32-32-16-输出

为了方便表示,下文用 FC-A 表示深度增加的模型,用 FC-B 表示宽度增加的模型。结果记录在表 3-9 和表 3-10 中。更深更宽的 FC 网络使用了更多的参数和计算量,但它们的结果仍然比 IC 网络差。值得注意的是, YEAST 的 FC-B 网络的计算资源消耗是 IC-FC 网络的六倍(6.45KMacs/6.46KB vs. 1.06KMacs/1.08KB),但其准确率仍然低于 IC-FC 网络(60.76% vs. 61.04%)。相比于扩展网络规模,利用 IC 神经元替换 MP 神经元是一种更经济和有效的提升模型性能的方法。

激活函数对 IC 神经网络的影响。上述实验中,激活函数都是 ReLU 函数。前文提到,IC 神经元的激活函数 $f(\cdot)$ 可以是 MP 神经元常用的激活函数。为了进一步探索 IC 神经元的泛用能力,本章用 Tanh 和 ELU^[72]代替 ReLU。这两种激活函数都可以取到负值区域。表 3-11 中的实验结果表明,替换激活函数,IC 神经网络依然能在大部分任务上取得超过 MP 神经网络的效果。在唯一没超过的 MNIST 数据集(Tanh 激活函数)上,IC-FC 和 FC 的差距只有 0.09%,基本达到了

表 3-10 分类任务中 IC-FC, FC-A, FC-B 模型的计算消耗 (FLOPs/Params) 对比。

Model	MNIST	IMDB	GTZAN	sEMG
FC-A	798.46/798.47	6811.89/6811.91	526.59/526.60	4650.5/4650.50
FC-B	535.81/535.82	9504.01/9504.04	263.94/263.95	6008.5/6008.51
IC-FC	236.56/235.91	5785.99/5781.25	100.09/99.98	3606.46/3603.97
Model	LETTER	YEAST	ADULT	-
FC-A	8.19/8.22	2.03/2.04	2.1/2.1	-
FC-B	8.08/8.11	6.45/6.46	6.65/6.65	-
IC-FC	4.18/4.22	1.06/1.08	1.13/1.14	-

表 3-11 IC 神经网络中使用不同激活函数的消融实验。

Function	Model	MNIST	IMDB	GTZAN	sEMG	LETTER	YEAST	ADULT
Tanh	FC	98.23	87.84	53.79	31.72	93.20	60.54	83.96
	IC-FC	98.31	88.52	54.48	33.52	93.45	61.88	84.19
ELU	FC	98.55	88.18	54.48	34.44	92.93	60.31	83.88
	IC-FC	98.52	88.24	55.52	36.48	93.65	61.66	84.24

持平的水平。

3.6 本章小结

本章受弹性碰撞模型启发, 提出了 IC 神经元模型, 与 MP 神经元相比, 它可以用于拟合更复杂的非线性分布。通过 XOR 问题的实验, 本章展示了单个的 IC 神经元具备区分线性不可分问题的能力。同时, IC 神经元保持了 MP 神经元易集成和轻量级的特点。本章将 IC 神经元集成到主流的网络架构中, 包括全连接、卷积和递归结构, 构建了多种 IC 网络。实验表明, 与传统网络相比, IC 网络能够达到更高的泛化性能和更快的收敛速度。此外, IC 网络带来的额外计算负担很小, 当模型规模较大时, IC 神经网络产生的计算负担与 MP 神经网络几乎一致。通过深度学习实验, 本章进一步表现了 IC 神经元在当下主流深度神经网络中的适用性。总结来说, IC 神经元作为一种新型神经元, 为构建更高效的神经网络提供了一条路径。

第四章 一种基于 IC 神经元的高性能卷积算法

4.1 背景介绍

第三章介绍了一种可以代替传统 MP 神经元的新型神经元结构 (IC 神经元), 并将它与传统卷积层相结合。实验结果表明 IC 神经元可以有效提升很多现存卷积模型的性能。然而, 上一章构建的 IC 卷积神经网络只是一种 IC 神经元和卷积结构的简单结合, 这一结构面临一个问题: 当接受到的输入维度较大时, IC 神经元的表达能力会受到明显限制。

第三章中指出, IC 神经元的工作原理是用一超平面将输入空间切分成两部分, 再在子空间执行不同的线性变换, 从而增强表达能力。为了在给定任务上取得良好的性能, 超平面通常需要满足或近似满足潜在的数据分布。例如在 XOR 问题解决方案 (见图 3-3) 中, 超平面 (二维空间中为一直线) 必须要在二维空间中将四个点切分成数量为 1 和数量为 3 的两组。在一些低维任务中, 数据分布较为简单, 此时可以找到一个超平面来完美或近似满足数据分布。然而, 超平面是一个线性结构。在高维任务中, 数据分布更复杂, 难以用一个线性超平面来满足所有数据分布。例如, 卷积模型经常会涉及一些高维的输入, 例如在 ResNet-18 模型的最后一个卷积层中, 滑动窗口内神经元接受到的输入是 $4608(3 \times 3 \times 512)$ 维的输入。对于一些更大型的模型, 例如 ConvNeXt-T^[73], 卷积层内神经元接受到的输入维数甚至可以过万。在这种输入空间中, IC 神经元引入的额外非线性模式很可能会无效化。

为了优化 IC 卷积层在高维数据中的表现, 本章提出一种方法将动态卷积引入 IC 神经元。在此之前的 IC 神经元的超平面由参数 \mathbf{w} 和 w' 确定, 当模型训练结束后, 超平面也随之固定。在改进的方法中, 超平面的参数由神经元的输入动态计算得到。这种情况下, 超平面本质上变为一个可学习的超曲面。新构建的卷积模型被称为 IC-CNN+。除了将超平面转化为动态超曲面的思想外, IC-CNN+ 中用到的动态卷积技术也尤为关键。在此之前, 一些动态卷积技术已被提出, 然而

它们普遍存在一些尚未解决的问题，导致动态化参数无法准确引导超曲面的学习。本章针对相关问题，并提出了一种更合理的动态化方法，称为局部感知动态卷积 (LADConv)。本章通过实验验证了 LADConv 与 IC-CNN+ 具有非常好的契合度，组合他们可以显著提升卷积模型的性能。

针对超平面的优化方法是 IC-CNN+ 中最重要的贡献。除此之外，本章还基于 IC 卷积结构的特性提出了一种快速训练策略，在一些情景下可以大幅缩短训练时间。本章主要贡献总结如下：

- 提出了一种基于动态卷积的 IC 卷积结构 (IC-CNN+)，可以有效解决 IC 神经元在高维数据中受限的问题；
- 针对现有动态卷积方法的缺陷，提出一种新型动态卷积算法 LADConv，它可以很好地适配 IC 神经元，发挥其潜能；
- 实验验证了 IC-CNN+ 模型可以作为一种高效的即插即用模块提升各种深度卷积神经网络的性能。

4.2 动态卷积与 IC 卷积结构的结合

首先，本章简单回顾 IC 神经元和 IC 卷积结构。单个 IC 神经元的输出表示为：

$$y = f(\mathbf{W}\mathbf{x} + \sigma(\mathbf{W}\mathbf{x} - w'\mathbf{I}\mathbf{x} + b_1) + b_2), \quad (4-1)$$

其中 $\sigma(\cdot)$ 是整形线性单元 (ReLU)。 b_1 和 b_2 是两个独立的偏置，用于调整模型分布的中心。 w' 是与输入维度无关的调整权重。 f 和 \mathbf{W} 表示激活函数和输入权重，与 MP 神经元中所做的操作类似。 \mathbf{I} 是与 \mathbf{W} 相同大小的全 1 张量。通过引入 $\sigma(\cdot)$ 操作，IC 神经元可以增加输出非线性模式的数量。具体而言， $H = (\mathbf{W} - w'\mathbf{I})\mathbf{x}$ 可以表示 n 维输入空间中的一个超平面 ($H = 0$)，它将 IC 神经元分为以下两种情况：

$$y = \begin{cases} f(2 \sum_{i=1}^n w_i x_i - \sum_{i=1}^n x_i) & \text{if } H \geq 0 \\ f(\sum_{i=1}^n w_i x_i) & \text{if } H < 0. \end{cases} \quad (4-2)$$

为了简化符号，这里省略了偏置项。超平面 $H = 0$ 将输入空间划分为两个子空

间，在这两个子空间中输入信号经历不同的变换。

在 IC 卷积中，输入特征图用 $\mathbf{X} \in \mathbb{R}^{C' \times H' \times W'}$ 表示，IC 卷积层的输出特征图 $\mathbf{U} \in \mathbb{R}^{C \times H \times W}$ 可以根据通道拆分为 $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_C]$ 。每个通道特征图 $\mathbf{u}_i \in \mathbb{R}^{H \times W}$ 由以下公式给出：

$$\mathbf{u}_i = \mathbf{w}_i * \mathbf{X} + \sigma(\mathbf{w}_i * \mathbf{X} - w'_i \times (\mathbf{I} * \mathbf{X})), \quad (4-3)$$

其中 $*$ 表示卷积操作。 $\mathbf{w}_i \in \mathbb{R}^{k \times k \times C'}$ 表示卷积核， \mathbf{I} 是与 \mathbf{w}_i 形状相同的全 1 张量。

尽管方程(4-4)在每个滑动窗口内都保留了 IC 神经元切分输入空间的特性，但是分割用的超平面的空间位置只由权重 \mathbf{w} , w' 和偏置决定。当训练结束时，超平面被完全被固定住。此外，超平面是线性的。在高维任务中，数据分布往往无法用一个线性平面直接划分开来。因此，IC 卷积的结构可能无法表示高维空间中的数据分布。

为进一步论证上述观点，也即 IC-CNN+ 提出的动机，本章进行了如下实验：训练 IC 深度卷积神经网络，然后随机选择 500 张测试数据输入模型，当某个神经元(卷积核)在所有测试数据上都表现出如下性质时，本章定义该 IC 神经元退化为 MP 神经元。

$$\mathbf{w}_i * \mathbf{X} + \sigma(\mathbf{w}_i * \mathbf{X} - w'_i \times (\mathbf{I} * \mathbf{X})) \equiv \mathbf{w}_i * \mathbf{X}. \quad (4-4)$$

实验采用了 Cifar10/100 数据集和 ImageNet 数据集，实验结果如表4-1所示，在深度模型中，大量的 IC 神经元没有发挥用处，数据只会落在超平面的一侧。此外，随着模型加深，IC 神经元的退化比例还在上升。该实验反映了深度模型中的 IC 卷积结构的表达能力会收到高维数据的限制。

表 4-1 深度模型中，IC 神经元退化的比例。

Model	ImageNet	Cifar10	Cifar100
ResNet-18	45.32%	37.64%	35.24%
ResNet-34	56.31%	54.75%	48.26%

本章提出一种方法来改进 IC 卷积结构。首先，定义一组动态化权重 \mathbf{w}^{dy} 。不

同于参数化的权重 \mathbf{w} , \mathbf{w}^{dy} 由当前输入特征图计算得到:

$$\mathbf{w}^{dy} = g(\mathbf{X}), \tag{4-5}$$

其中, $g(\cdot)$ 是一个轻量级的神经网络。本章可以将之前的动态卷积技术^[13,31]直接用于生成 \mathbf{w}^{dy} , 在这些方法中, $g(\cdot)$ 依赖于 Squeeze-and-Excitation (SE) 模块^[10]表示。该模块首先用全局池化操作 (GAP) 将输入 \mathbf{X} 压缩成一个一维向量, 再通过全连接层将该向量投影到 $k \times k \times C'$ 维卷积核对应的空间。动态权重 \mathbf{w}^{dy} 由 SE 模块的结果与参数化的静态卷积核组合生成。在生成动态卷积核后, 本章用 \mathbf{w}^{dy} 代替全 1 卷积核 \mathbf{I} 。新的卷积结构表示为:

$$\mathbf{u}_i = \mathbf{w}_i * \mathbf{X} + \sigma(\mathbf{w}_i * \mathbf{X} - \mathbf{w}'_i \times (\mathbf{w}^{dy} * \mathbf{X})). \tag{4-6}$$

超平面变为 $(\mathbf{w}_i - \mathbf{w}'_i g(\mathbf{X}))\mathbf{x} = 0$ 形式的超曲面, 其中 \mathbf{x} 表示滑动窗口内的输入, \mathbf{X} 表示整个输入特征图。方程 (4-6) 的卷积结构被称为 IC-CNN+。这是一种基于 IC 神经元的优化结构, 其宏观结构如图4-1所示。

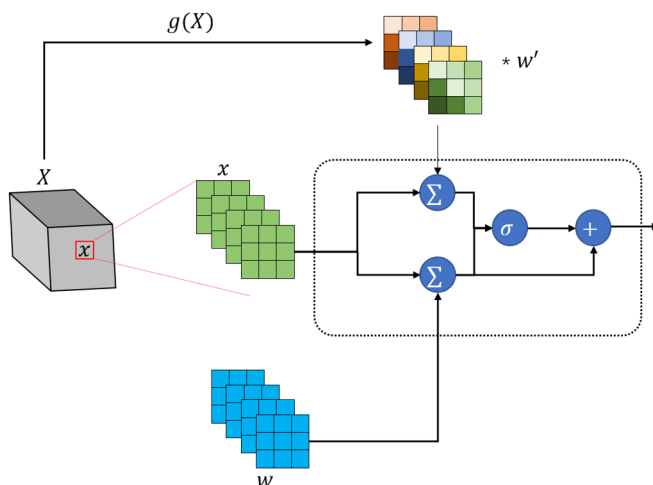


图 4-1 IC-CNN+ 示意图。 $g(\mathbf{X})$ 为生成动态卷积的过程。虚线框内表示 IC 神经元的计算流程。

实际应用中, $g(\cdot)$ 通常用一神经网络表示, 因此, 超曲面的具体形式可以根据给定任务学习得到。通过训练 $g(\cdot)$ 模块, 该超曲面能够拟合复杂的高维数据分布。此外, 方程 (4-6) 所表示的超曲面还具有以下几个优点:

- 超曲面中, $g(\mathbf{X})$ 由输入特征图直接计算得到, 对于所有卷积核 $g(\mathbf{X})$ 项只

用计算一次，时间复杂度与原始 IC 卷积层中的全一卷积核 \mathbf{I} 相似，保证了 IC-CNN+ 具有和基础 IC 卷积结构 (第三章所示结构) 相似的计算效率。

- $\mathbf{w}^{dy} = g(\mathbf{X})$ 将输入特征图上的全局信息融入到动态参数中。因此，在每个滑动窗口内，神经计算单元能将局部特征与全局特征联系起来，有利于捕获更准确的特征。

4.3 IC-CNN+ 中的动态卷积算法研究

尽管上一节中通过动态卷积核优化了 IC 卷积层的结构，但生成动态卷积核的过程依然存在一系列问题。本文通过分析发现现存的动态卷积普遍存在一个问题：忽略了输入特征中局部信息的特异性，而这点对于动态卷积核 \mathbf{w}^{dy} 在 IC 神经元中的性能至关重要。

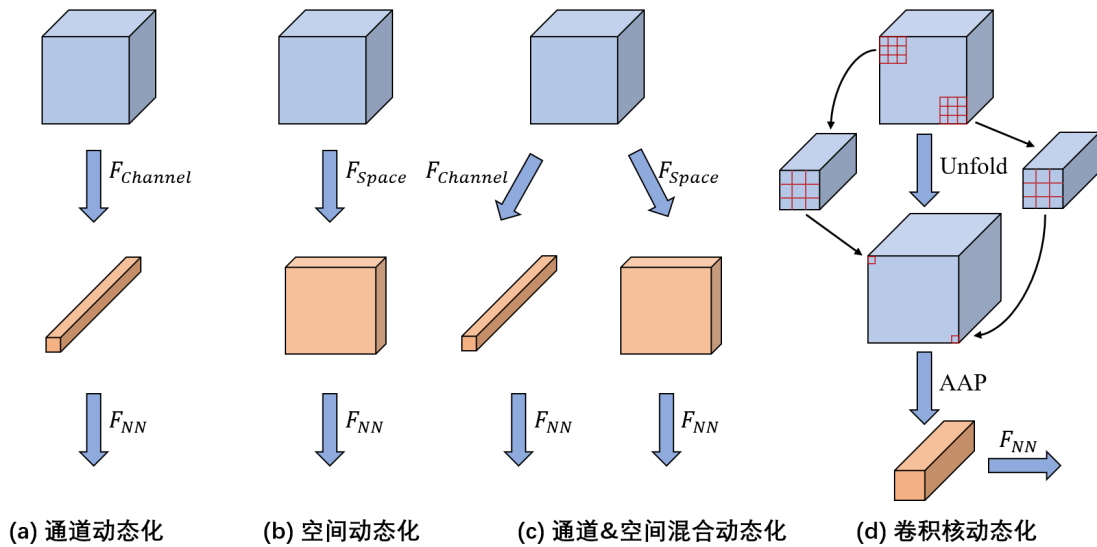


图 4-2 不同视觉动态化机制 (Vision Attention) 的示意图。符号 F_{NN} 表示用于抽象特征的各种神经网络。(a): 通道动态化机制; (b) 空间动态化机制; (c) 通道和空间混合动态化机制; (d) 卷积核动态化机制 (本文提出方法)。

动态卷积核的目的与传统静态卷积核的目的是一样的，即准确学习局部区域内的特征。IC-CNN+ 中用到的动态卷积核也遵循这一目的：由动态卷积核表示的超曲面需要学习滑动窗口内的数据分布。之前的动态卷积工作^[12-13,31]大部分采用 SE 模块来生成动态核。这种结构的特点是足够轻量级，但是缺点也很明显：SE 模块过分压缩空间信息，在前馈过程中丢失了大量局部信息，难以指导动态卷积核的行为。具体来说，SE 模块首先用全局池化操作将所有像素聚合成单个像素，不可逆地压缩了空间信息，只保留了通道上的分布信息。这种操作生

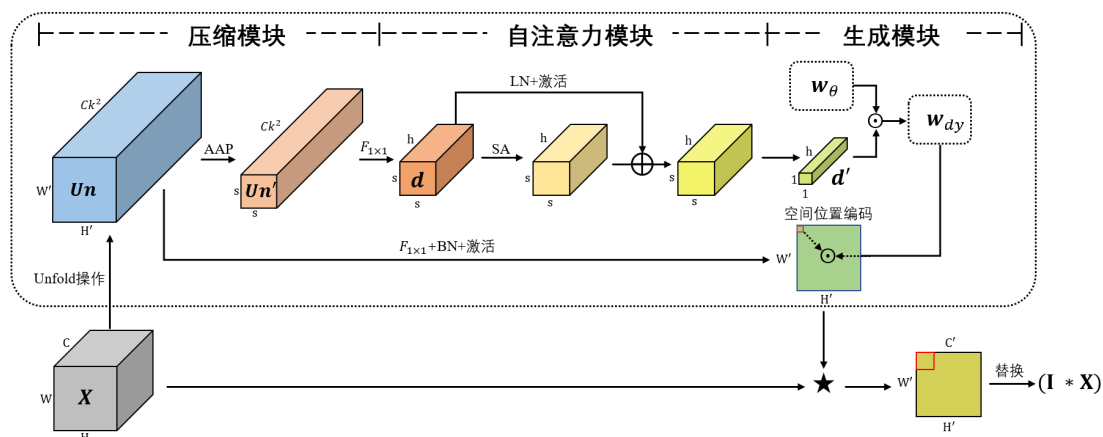


图 4-3 LADConv 动态化方法的全局结构。 w_θ 表示静态卷积核， w_{dy} 表示动态卷积核，Position attention 表示动态卷积核在空间位置上的位置编码。LADConv 最终的输出用于替换 IC 卷积结构中的 $(I * X)$ 项。

成的卷积核难以理解到哪些局部特征是关键。为了更好地理解这一过程，本章将一些动态化原理绘制到图 4-2 中，其中，(a) 表示将动态化技术作用到通道重校准上，SE 模块提出的初衷也是为了解决这一问题。通道重校准要求动态化系数只反映通道的重要性，不考虑空间信息，这与 SE 模块的构造相契合。(b) 表示将动态化技术用到空间像素上。该方法中空间动态化技术首先用通道上的全局池化将通道压缩成 1 维。同时保留空间所有像素分布的完整性，再用大核卷积等方法去生成动态化系数，其设计理念刚好与通道动态化相反。

上述两种技术的共通点是它们都将动态系数的生成过程与任务目标联系在一起，尽可能减少与任务目标有关的信息损失。同理，动态卷积核技术的任务目标是构建一个能够学习局部特征的卷积核，因此，在生成阶段需要保证局部信息不会过度丢失。本文期望的动态卷积核生成方式如图 4-2(d) 所示，该结构能够考虑局部区域中的特异性，可以定义哪些类型的局部特征是关键。基于这种思想，本章构造了一种新型动态卷积核，它被称为局部感知动态卷积 (LADConv)，它可以和 IC 卷积结合并发挥 IC 神经元在高维数据中的潜能。具体结构的介绍如下。

4.3.1 全局结构

LADConv 的全局结构如图 4-3 所示。LADConv 由压缩、自注意力和生成模块组成，其中，压缩和自注意力模块用于生成局部特征描述符，而生成模块将特征描述符转化为动态卷积核。最终，动态卷积核被用于替换 IC 卷积中的 $I * X$ 操

作。

4.3.2 压缩模块

压缩模块 (Squeeze Module) 是所有动态化方法的前置操作。他需要满足一个基本需求：保证模块的轻量性，不会影响整体模型的复杂度。在 LADConv 的压缩模块中，首先定义输入中的局部特征区域。即普通卷积核遍历的每一个子区域。这些局部区域的大小被定义为 $k \times k \times C$ 。为了便于后续计算，本章将输入中每个局部区域转换为一个 Ck^2 维向量，然后将所有向量连接起来构造一个新的特征图，表示为 $\mathbf{Un} \in \mathbb{R}^{H \times W \times Ck^2}$ 。这个操作被称为特征图的“展开 (unfolding)”，可以形式化地表示为

$$\begin{aligned} \mathbf{Un} &= \text{Unfold}(\mathbf{X}) \\ \mathbf{Un}_{ij} &= [\mathbf{X}_{i-[k/2],j-[k/2]}; \dots; \mathbf{X}_{i+[k/2],j+[k/2]}], \end{aligned} \quad (4-7)$$

其中下标 i 和 j 表示像素的顺序， \mathbf{Un}_{ij} 表示输入 \mathbf{X} 位置 (i, j) 处的局部区域内的元素集合。展开操作重新排列输入元素而不引入任何数学变换。接下来，本章对 \mathbf{Un} 应用自适应平均池化 (AAP) 来提取代表性的局部特征：

$$\mathbf{Un}' = \text{AAP}(\mathbf{Un}), \quad (4-8)$$

其中 $\mathbf{Un}' \in \mathbb{R}^{s \times s \times Ck^2}$ 。超参数 s 表示自适应平均池化沿输入特征图高度和宽度维度采样的数量。为了在每个局部分布中捕获深层特征，本文在 \mathbf{Un}' 上使用了一个维度缩减的 1×1 卷积：

$$\mathbf{d} = \sigma(\text{BN}(F_{1 \times 1}(\mathbf{Un}'))), \quad (4-9)$$

其中， 1×1 卷积的输出尺寸表示为 h 。应用中 h 一般被设置为 $\text{Max}(\lambda, C/r)$ 。 λ 和 r 是超参数。另外，BN 和 σ 代表批标准化操作和 ReLU 激活操作。

通过上述过程，压缩模块将输入特征图中的局部信息存储在向量 \mathbf{d} 中。这个模块即满足了压缩模块轻量级的需求，也服从了动态卷积核依赖局部特征的归纳偏置。

4.3.3 自注意力模块

压缩模块的结果 \mathbf{d} 包含了 s^2 个局部特征，这些局部特征是平级的，反映了原始输入特征图中不同区域中的局部特征。由于他们是平级特征且不存在顺序关系，直接利用这些特征生成动态卷积核会受到一些阻碍。一种直接的方法是对 s^2 个局部特征进行平均，然而，这种方法忽视了局部特征的多样性，削弱了关键局部细节的表示。为了克服这一挑战，本章利用自注意力 (Self-attention, SA) 来学习这些局部特征之间的相互依赖关系，从而保留信息丰富的局部特征。具体来说，首先将 \mathbf{d} 重组 (Reshape) 为一个 $s^2 \times h$ 的张量，并应用层标准化 (LayerNorm, LN) 操作。然后，自注意力模块将 s^2 代表的维度视为序列维度来应用自注意力。公式表达如下：

$$\begin{aligned} \mathbf{d}_{Norm} &= \text{LN}(\mathbf{d}) \\ \text{SA}(\mathbf{d}) &= \frac{FQ(\mathbf{d}_{Norm})FK(\mathbf{d}_{Norm})^T}{\sqrt{h}FV(\mathbf{d}_{Norm})} \\ \mathbf{d} &:= \text{SA}(\mathbf{d}) + \mathbf{d}, \end{aligned} \quad (4-10)$$

其中 $FQ(\cdot)$ 、 $FK(\cdot)$ 和 $FV(\cdot)$ 对应于三个输出维度为 h 的全连接层。方程 (4-10) 省略了维度重组的操作。最后的特征由 s^2 维度的全局平均池化 (GAP) 得到，其公式表达为

$$\mathbf{d}' = \text{GAP}(\mathbf{d}), \quad (4-11)$$

其中， \mathbf{d}' 是一个 h 维向量，它也被成为局部特征的描述符，用于生成动态卷积核。

输入中的局部分布本身就具有多样性的特点。本章所应用的自注意力模块可以有效地捕获多样局部特征之间的依赖关系。然而，方程 (4-8) 中的池化操作会导致了多样性的丢失。通过增大超参数 s 来减轻多样性的损失，为自注意力模块提供更多的局部细节，但也增大了计算消耗。第4.5.7章的消融实验探索了 s 的取值范围，并给出了平衡准确性和计算效率的经验配置。

4.3.4 动态卷积核生成

生成局部特征描述符 \mathbf{d}' 后，它被用于生成最终的动态卷积核。本章从核内空间和核在整个卷积的位置两个角度来生成动态核。对于核内空间，本章利用

ODConv^[31]的方法。首先，用全连接层将 \mathbf{d}' 投影成动态向量，再将这些向量与 n 个大小为 $k \times k \times C$ 的直接参数化的静态参数核相乘，最后加权求和得到一个大小为 $k \times k \times C$ 的动态卷积核。具体操作如下。

构造的动态向量包括 \mathbf{A}_{in} 、 \mathbf{A}_s 和 \mathbf{A}_p 。他们分别表示输入通道、卷积核空间大小，由全连接层生成：

$$\mathbf{A} = \text{Act}(F_{1 \times 1}(\mathbf{d}')), \quad (4-12)$$

其中 $\text{Act}(\cdot)$ 表示激活函数。 \mathbf{A}_{in} 和 \mathbf{A}_s 使用 Sigmoid 激活函数， \mathbf{A}_p 使用 SoftMax 激活函数。本章用 $\mathbf{w}_\theta \in \mathbb{R}^{n \times k \times k \times C}$ 统一表示 n 个静态卷积核 (参数化核)。然后，上述生成的动态向量与 \mathbf{w}_θ 的不同维度一一相乘，例如， $\mathbf{A}_{in} \in \mathbb{R}^C$ 与 \mathbf{w}_θ 的输入通道维度 (C 表示维度) 相乘。 $\mathbf{A}_p \in \mathbb{R}^n$ 表示 n 维度上的动态系数，也即 n 个静态卷积核的权重。最后，将加权后的 n 个卷积核相加得到动态卷积核。整个过程表示为以下方程。

$$\mathbf{w}_{dy} = \mathbf{A}_p \odot \mathbf{A}_s \odot \mathbf{A}_{in} \odot \mathbf{w}_\theta \quad (4-13)$$

$$\mathbf{w}_{dy} := \text{Sum}_p(\mathbf{w}_{dy}),$$

其中 $\text{Sum}_p(\cdot)$ 表示对 n 维度进行聚合。需要注意的是，上述过程没有考虑输出通道的动态化系数，因为在 IC-CNN+ 中，被代替的全 1 卷积核 \mathbf{I} 只包含两个维度：1. 空间维度 $k \times k$ ；2. 输入通道维度 k 。不包含输出通道维度。

方程 (4-13) 中的 \mathbf{w}_{dy} 已经可以用于 IC-CNN+ 超曲面方程的构建。然而，如果在动态卷积核中加入位置信息可以进一步提升超曲面方程对空间位置的感知能力，有望进一步提升 IC-CNN+ 的性能。因此，本章在方程 (4-13) 的动态卷积核上应用简单的位置信息编码。由于压缩模块中的 \mathbf{U}_n 的每个像素都表示原始特征中的一块局部区域，因此，可以直接使用 1×1 卷积将 \mathbf{U}_n 转化为位置编码。

$$P_{ij} = \text{Sigmoid}(F_{1 \times 1}(\mathbf{U}_{ij})). \quad (4-14)$$

动态卷积核的卷积结果 \mathbf{U}_{ij} 的 (i, j) -th 像素由以下公式给出

$$\mathbf{U}_{ij} = \text{Sum}((P_{ij} \cdot \mathbf{w}_{dy}) \odot \mathbf{U}_{ij}). \quad (4-15)$$

在实现时，空间位置编码相当于将矩阵 \mathbf{P} 与卷积结果 $\mathbf{w}_{dy} * \mathbf{X}$ 沿着空间维度进行

乘法，这是一种很轻量级的形式，不会明显增加的前馈的计算消耗。

LADConv 在 IC-CNN+ 中的应用。LADConv 最大的优点是将局部特征融入到动态卷积核的计算中，而 IC 卷积中切分输入空间的超平面本质上也是拟合局部区域内的数据分布，两种结构的目的相互契合。在应用中，可以将两者组合使用。

LADConv 算法也可以单独作为一种新型卷积，但在这种情况下，需要大量扩展生成的卷积核数量，增加了参数量和计算量，同时也增加了模型运行的并行难度。本文会在实验中对独立的 LADConv 进行分析。

4.4 基于 IC 卷积结构的快速训练算法

在深度模型中，IC-CNN+ 可以基于一个已有的网络框架进行设计。例如，ResNet-50 IC-CNN+ 是将 ResNet-50 中所有卷积层都替换为方程(4-6)中的结构。IC-CNN+ 的训练方法与基础模型的训练无异。例如开源社区内通常使用随机梯度下降 (Stochastic Gradient Descent, SGD) 和每 30 轮学习率下降策略来训练 ResNet-50, 同样的配置也可以应用到 ResNet-50 IC-CNN+ 上。除此之外, IC-CNN+ 主要面对的是深度视觉模型。它们的共同点是：1. 基础模型普遍存在开源预训练模型。2. 重头训练需要花费较长的时间和计算资源。基于上述特点，本章根据 IC-CNN+ 的特性设计了一种基于预训练模型的快速训练算法。这种方法在深度模型应用中会显著降低训练成本。

通过观察 IC-CNN+ 中卷积层的数学形式 (方程 (4-6)), 可以发现 IC 卷积可以拆分成一个普通卷积 $\mathbf{w}_i * \mathbf{X}$ 和一个残差项 $\sigma(\mathbf{w}_i * \mathbf{X} - w'_i \times (\mathbf{w}^{dy} * \mathbf{X}))$ 。受启发于这点，本章将预训练的基础网络用于 IC-CNN+ 模型中。EWC 模型^[74]反映了同一神经网络模型可以通过多种权重配置达到最优泛化性能，也就是说让一个神经网络达到最好效果的权重不唯一，这些权重在高维空间内组成了一个连续的集合。通过将 IC 卷积中残差项的值设为 0，IC 卷积网络将会退回基础模型。因此，可以假设基础网络的最优参数集合和 IC 卷积网络的最优参数集合在交集。在这种情况下，基础网络的预训练模型可以直接用于 IC 卷积网络的训练。首先，本章在每个 IC 卷积层中添加一个缩放因子 α 来控制残差项影响，然后通过

基础模型的预训练权重来初始化 IC 卷积神经网络：

$$\mathbf{u}_i = \mathbf{w}_i^* * \mathbf{X} + \alpha \times \sigma(\mathbf{w}_i^* * \mathbf{X} - w'_i \times (\mathbf{w}_{dy} * \mathbf{X})), \quad (4-16)$$

其中 \mathbf{w}_i^* 是从预训练模型中加载的权重，而对于权重 w'_i 和动态卷积核中的权重，可以使用随机初始化。由于 $\mathbf{w}_i^* \otimes \mathbf{X}$ 项可以独立捕获特征，因此本章在初始化时将 α 设置为零，使 IC 神经网络的性能等同于预训练基础模型的性能。初始化后，缓慢调控 α 的值并用小学习率微调所有参数，IC-CNN+ 可以逐渐突破预训练模型的性能。在实际应用中， α 也可以被视为可学习参数与其它参数一同训练。

目前大部分深度学习模型都存在开源的预训练模型，例如 ResNet, ConvNeXt 等。基于预训练模型，IC 神经网络可以在很少的轮次中达到收敛，大大减少了训练时间。然而，上述提出的训练过程基于一个假设：IC 神经网络和基础网络的最优参数集合存在交集。实际应用中存在不满足这个假设的情况，此时微调会让 IC 神经网络朝着一个精度变差的方向训练，并且难以达到优于预训练模型的性能。为了解决这一问题，本章利用自蒸馏技术来定义图像分类任务中的损失函数。引入的损失函数由三部分组成：第一部分是建立在 IC-CNN+ 最终输出上的蒸馏损失，它使用 Kullback-Leibler (KL) 散度。首先需要计算带有温度缩放的 SoftMax 分布^[75]：

$$p^i(x; \tau) = \frac{e^{s_i(x)/\tau}}{\sum_k e^{s_k(x)/\tau}}, \quad (4-17)$$

其中 x 是数据样本， $s(x)$ 是最后 softmax 层的输出， i 是类别索引， $s_i(x)$ 是 x 在类别 i 上得到的得分对数， τ 是温度。KL 散度损失 L_{kl} 定义如下：

$$L_{kl} = -\tau^2 \sum_{x \in \mathcal{D}_x} \sum_{i=1}^C p_t^i(x; \tau) \log(p_s^i(x; \tau)), \quad (4-18)$$

其中 t 和 s 分别表示教师模型（基础模型的预训练模型）和学生模型（IC-CNN+ 模型）。 C 是类别总数， \mathcal{D}_x 表示数据集。第二部分损失用于特征蒸馏。本章使用 $F_t(x)$ 和 $F_s(x)$ 分别表示基础模型和 IC-CNN+ 的最终特征图。特征损失 L_{fe} 由 L2 损失测量：

$$L_{fe} = \sum_{x \in \mathcal{D}_x} \|F_t(x) - F_s(x)\|_2^2. \quad (4-19)$$

在图像分类任务中，训练 IC-CNN+ 的完整损失函数 L 是标准交叉熵损失 L_{ce} 和蒸馏损失的组合：

$$L = \lambda_1 L_{ce} + \lambda_2 L_{fe} + \lambda_3 \max(L_{kl} - e, 0), \quad (4-20)$$

其中 λ_i 是平衡权重。 e 是一个常数，用于提升基础模型和 IC-CNN+ 模型 SoftMax 分布差距的容忍度。

引入信息蒸馏可以让 IC-CNN+ 的训练不会偏离基础模型所代表的参数集合，避免了精度大幅下降。同时，本章希望 IC-CNN+ 能够取得优于基础模型的结果，因此在损失(4-20)中引入了 margin 损失的思想，允许 IC-CNN+ 和基础模型的结果存在差异。平衡权重 λ_2 通常被设置为一个小于 0.1 的值，方便 IC-CNN+ 突破基础模型的限制。总结来说，损失(4-20)为训练 IC-CNN+ 引入了一些额外分类信息。结合加载预训练参数和微调，IC-CNN+ 可以在很短的时间内收敛。这种训练方法称为“弱信息蒸馏”(WLD)，它为 IC-CNN+ 的训练提供了额外的知识，同时保证了它学习新知识的能力。

4.5 实验结果与分析

本章通过一系列比较实验来验证 IC-CNN+ 有效性。此外，本章还设计了多项消融实验评估动态卷积在 IC-CNN+ 中的作用。

4.5.1 数据集和基线配置

本章的实验使用了包含超过 1000 万张彩色图像的 ILSVRC ImageNet 2012 分类数据集^[32]。ImageNet 共有 1000 个类别，包括 1280 万张训练图像和 5 万张验证图像。实验中使用两个版本的 ResNets (ResNet-18、ResNet-50) 构建相应的 IC-CNN+ 网络。为了公平比较，本章在 ImageNet 上的所有实验都使用相同的环境设置。优化器采用随机梯度下降^[63]方法，权重衰减为 10^{-4} ，动量为 0.9。训练过程设置为 120 个 epochs，批大小为 256。学习率初始值设为 0.1，并且每 30 个 epochs 减小 10 倍。

4.5.2 ImageNet 实验结果

表 4-2 ImageNet 下各种模型的精度和计算消耗对比。准确率 (%) 使用 single-crop 测试。IC+ 表示 IC-CNN+ 模型。

Backbone	Model	Top-1 Acc (%)	Top-5 Acc (%)	Flops (GMac)	Params
ResNet-18	Baseline	70.25	89.38	1.81	11.69M
	+SE	70.98 (↑ 0.73)	90.03 (↑ 0.65)	1.82	11.78M
	+CBAM	71.01 (↑ 0.76)	89.85 (↑ 0.47)	1.82	11.78M
	+ECA	70.60 (↑ 0.35)	89.68 (↑ 0.30)	1.82	11.69M
	+CGC	71.60 (↑ 1.35)	90.35 (↑ 0.97)	1.83	11.69M
	+WE	71.00 (↑ 0.75)	90.00 (↑ 0.62)	1.82	11.90M
	+IC	70.77 (↑ 0.52)	89.98 (↑ 0.60)	1.82	11.74M
	+(IC+)	72.38 (↑ 2.13)	90.68 (↑ 1.30)	1.84	11.91M
ResNet-50	Baseline	76.23	93.01	3.86	25.56M
	SE	77.31 (↑ 1.08)	93.63 (↑ 0.58)	3.87	28.07M
	CBAM	77.46 (↑ 1.23)	93.59 (↑ 0.58)	3.89	28.07M
	ECA	77.34 (↑ 1.11)	93.64 (↑ 0.63)	3.87	25.56M
	CGC	76.79 (↑ 0.56)	93.37 (↑ 0.36)	3.88	25.59M
	WE	77.10 (↑ 0.87)	93.50 (↑ 0.59)	3.86	28.10M
	+IC	76.73 (↑ 0.50)	93.31 (↑ 0.30)	3.87	25.60M
	+(IC+)	77.49 (↑ 1.26)	93.70 (↑ 0.69)	3.88	25.80M

本章通过将块中的整个 3×3 卷积层替换为 3×3 的 IC 卷积层来构建 IC-ResNet-18 和 IC-ResNet-50。训练后将 IC 卷积神经网络的计算结果与基础模型结果进行对比。除此之外，本章还将 IC-CNN+ 与一些前沿动态化方法或者特殊卷积结构进行对比，包括 SE^[10]，CBAM^[76]，ECA^[77]，CGC^[78]和 WE^[79]。这些方法与 IC-CNN+ 的共通点是都是即插即用模块，训练和应用过程相似，因此可以作为比较方法。

实验结果如表4-2所示，表中结果由之前工作^[31]提供。IC-CNN+ 在 ResNet-18 和 ResNet-50 上分别将 top-1 准确率提升了 2.13% 和 1.26%。与基础 IC 卷积结构相比，IC-CNN+ 在 ResNet-18 和 ResNet-50 上都取得了显著提升。验证了 IC-CNN+ 的有效性。此外，无论 IC-CNN 还是 IC-CNN+，它们在 Flops 和 Params 上的增加都在 1% 左右。这个结果也表明了 IC-CNN+ 并不会破坏 IC 卷积结构轻量性的特点。

此外，IC-CNN+ 在与表4-2中其它方法的对比中也表现出优异的性能。除此之外 SE，CBAM 和 ECA 是在计算出特征图进行特征重校准。他们与 IC-CNN+ 作用的位置不同，相互独立。在实际应用中，可以将 IC-CNN+ 与上述方法进行结合，以取得更高的准确率。

4.5.3 CIFAR 实验结果

本章实验通过 CIFAR 数据集验证 IC-CNN+ 在广泛的深度卷积模型中的泛用能力。在这些实验中，优化器采用随机梯度下降（SGD）方法^[63]，其中权重衰减率为 10^{-4} ，动量为 0.9。每个模型在批量大小为 128 的情况下进行 200 个周期的训练。学习率初始化为 0.1，在第 60 和第 120 个周期时降低 10 倍。每组实验重复三次取平均以避免偶然误差。

表 4-3 CIFAR 数据集下的 IC-CNN+ 以及基础模型的 Top-1 分类结果。

Model	CIFAR10	CIFAR100	Model	CIFAR10	CIFAR100
VGG-16	93.64	72.93	ResNet-18	95.02	75.61
+(IC)	93.90	73.35	+(IC)	95.20	75.95
+(IC+)	93.96	74.15	+(IC+)	95.48	77.14
DenseNet-121	95.41	77.01	ResNet-50	94.94	77.74
+(IC)	95.52	79.15	+(IC)	95.12	78.59
+(IC+)	95.70	79.68	+(IC+)	95.21	79.63
MobileNet	90.08	65.98	SE-ResNet-18	94.92	76.44
+(IC)	90.90	67.32	+(IC)	95.06	76.60
+(IC+)	91.25	68.44	+(IC+)	95.28	77.98
ResNeXt	95.38	77.77	SE-ResNet-50	94.90	78.58
+(IC)	95.46	78.09	+(IC)	95.02	79.18
+(IC+)	95.56	78.82	+(IC+)	95.47	79.87

CIFAR 数据集的实验结果如表4-3所示，相比于基础的 IC 卷积结构，IC-CNN+ 在八个模型上都取得了明显提升，尤其在 CIFAR100 数据集上。实验结果验证了本章对 IC 卷积结构的改进是有效的，并且适用于广泛的卷积模型。此外，表4-3中 SENet 的结果也展示了 IC-CNN+ 可以与一些即插即用模块相互结合，达到更高的准确率。

4.5.4 训练过程优化的实验结果

前两节从模型的性能上验证了 IC-CNN+ 的有效性。为了让 IC-CNN+ 更好地部署到实际应用，本章还引入了一种基于 IC 卷积结构的快速训练的方法 (WLD)，可以降低深度模型的训练时间。通过结合模型和训练方法，可以全面地评估 IC-CNN+ 的应用价值。

本章实验建立在 ResNet 和 EfficientNet^[80]模型上。EfficientNet 是一种基于网络搜索技术的模型，它的训练超参数较为复杂，且训练周期较长，且精度接近饱

表 4-4 IC 卷积神经网络在快速训练策略下的 ImageNet 分类结果。

Model	Top-1 err.	Top-5 err.	Model	Top-1 err.	Top-5 err.
ResNet-18	70.25	89.38	EfficientNet-B0	76.30	93.20
IC-ResNet-18	71.43	90.32	IC-EfficientNet-B0	76.41	93.25
ResNet-50	76.23	93.01	EfficientNet-B1	78.80	94.40
IC-ResNet-50	77.05	93.44	IC-EfficientNet-B1	78.95	94.51

和。IC-CNN+ 可以利用快速训练方法大幅降低 IC 卷积结构在 EfficientNet 模型上的重新训练。快速训练策略的超参数如下所示：训练过程设置为 30 个 epochs，学习率初始值为 0.001，每 15 个 epochs 减小 10 倍（EfficientNets 的实验为 10 个 epochs）。其他超参数设置为： e 为 0.005； λ 初始设置为 0.9，逐渐减小至 0.1。实验结果如表 4-4 显示，快速训练策略在较少的训练轮次下取得了超过基础模型的性能。在这个实验中需要注意的是，EfficientNets 是一种通过网络搜索得到的神经网络结构，搜索后的模型逼近最大的模型容量和饱和的精度，因此，IC 卷积结构在这些模型上的提升没有 ResNets 模型上的提升显著。

总而言之，受益于 IC 卷积结构的特性，WLD 确实能够极大的缩短训练时间，在一些计算资源受限或 backbone 超参数难以复现的应用场景下，WLD 可以有效提升 IC-CNN+ 的应用能力。但同时本文也观察到快速训练的结果和重头训练的结果之间依然还有差距，当计算资源充足时且对准确率要求很高时，重头训练的意义要大于使用 WLD。

4.5.5 LADConv 模型的实验结果

本章将动态卷积方法引入基础 IC 卷积结构的超平面建模中，前文提到，动态卷积的性能会影响 IC-CNN+ 的性能。为了深入评估动态卷积的影响，本章实验将提出的 LADConv 方法与其它一些卷积核动态化方法进行对比。

目前主流的卷积核动态化方法包括了 CondConv^[12]，DyConv^[13]，DCD^[29]，SD-Conv^[30]，还有 ODCConv^[31]，其中 ODCConv 在动态卷积领域上取得了最领先的效果。

这部分实验用到的基础模型包括 ResNet-18, ResNet-50, ResNet-101, MobileNetv2。为了确保公平比较，本章实验保持了与其他对比方法相同的训练设置。在 ResNet 实验中，模型使用随机梯度下降（SGD）进行了 100 个 epoch 的训练，其中权重衰减为 10^{-4} ，动量为 0.9。初始学习率设置为 0.1，并且每 30 个 epoch 减小 10

倍。对于 MobileNetV2 实验，训练持续时间延长到了 150 个 epoch，权重衰减为 4×10^{-5} 。初始学习率为 0.05，并遵循一种余弦周期，在一个周期内衰减为零。与以前的工作一致^[13,31]，本章实验在前 10 个 epoch 中采用了温度策略，并且对于 ResNet-18 采用了 0.1 的 dropout 率，对于其他模型采用了 0.2 的 dropout 率。ConvNeXt 实验中的训练配置与原论文中的配置保持一致^[73]。

LADConv 模型引入了几个超参数。在传统的卷积结构中， λ 和 r 都设置为 16。对于深度可分离卷积 (DWConv) 结构，隐藏数 h 分别直接配置为 4 和 8，用于 MobileNetV2 和 ConvNeXt-T。实验中将 s 设置为 3 用于 ResNets 和 MobileNetV2 的最后建筑块中，而在其他建筑块中， s 设置为 5。在 ConvNeXt-T 中， 7×7 卷积被替换为具有 $s = 2$ 的 LADConv 结构。此外，本章用 (1x) 和 (4x) 表示使用了 1 个或 4 个并列卷积核。

ResNets 性能比较。表 4-2 展示了基于 ResNet 的模型的验证准确度和 FLOPs/Params。结论如下：首先，CondConv 和 DyConv 通过引入额外的核参数来增强基线模型，导致总参数数量显著增加（对于 ResNet-18 分别为 7.0/3.9 倍，对于 ResNet-50 分别为 5.0/3.9 倍）。LADConv (1x) 展示了与基线模型相似水平的参数和 FLOPs，胜过所有动态卷积方法，除了 ODCConv (4x)。当多个核应用于 LADConv 时，LADConv (4x) 取得了最佳结果，在最小的额外计算成本下，超过了 ODCConv (4x) 在 ResNet-18/ResNet-50 上的 0.41%/0.34%。在 ResNet-101 比较中，LADConv (1x) 表现最佳，比第二好的方法 (ODConv (2x)) 少了 42.32% 的参数。

MobileNetV2 上的比较。在 MobileNetV2 上的实验探究了 LADConv 在轻量级架构上的有效性。比较结果见表 4-6。可以观察到，CondConv、DyConv 和 ODCConv (4x) 相对于基线模型都取得了显著的改进；然而，它们大幅增加了参数数量，损害了模型的轻量级特性。相比之下，LADConv (1x) 保持了与基线相当的参数效率，并且在 MobileNetV2 (1.0x) 和 MobileNetV2 (0.75x) 上的结果接近 ODCConv (4x) (差距分别为 0.09% 和 0.11%)。此外，LADConv (4x) 取得了最佳结果，在 MobileNetV2 (1.0x) 和 MobileNetV2 (0.75x) 上分别比 ODCConv (4x) 高出 0.35% 和 0.27%。

通过本章的实验，可以得出结论，LADConv 本身就能表现出优于其它动态卷积方法的性能。他可以刻画与局部特征联系更密切的局部动态卷积核，有利于 IC-CNN+ 发挥出 IC 神经元的潜能。

表 4-6 基础模型为 MobileNetV2 时的 ImageNet 分类结果。

Backbone	Model	Top-1 Acc (%)	Top-5 Acc (%)	Flops (MMac)	Params
1.0×	Baseline	71.65	90.22	300.8	3.50M
	+ CondConv (8×)	74.13 (↑ 2.48)	91.67 (↑ 1.45)	318.1	22.88M
	+ DyConv (4×)	74.94 (↑ 3.29)	91.83 (↑ 1.61)	317.1	12.40M
	+ DCD	74.18 (↑ 2.53)	91.72 (↑ 1.50)	318.4	5.72M
	+ ODConv (1×)	74.84 (↑ 3.19)	92.13 (↑ 1.91)	311.8	4.94M
	+ ODConv (4×)	75.42 (↑ 3.77)	92.18 (↑ 1.96)	327.1	11.52M
	+ SD-Conv	75.30 (↑ 3.65)	-	261.9	7.7M
	+ LADConv (1×)	75.33 (↑ 3.68)	92.16 (↑ 1.94)	353.1	4.76M
	+ LADConv (4×)	75.77 (↑ 4.12)	92.25 (↑ 2.03)	367.8	11.33M
0.75×	Baseline	69.18	88.82	209.1	2.64M
	+ CondConv (8×)	71.79 (↑ 2.61)	90.17 (↑ 1.35)	223.9	17.51M
	+ DyConv (4×)	72.75 (↑ 3.57)	90.93 (↑ 2.11)	220.1	7.95M
	+ DCD	71.92 (↑ 2.74)	90.20 (↑ 1.38)	222.9	4.08M
	+ ODConv (1×)	72.43 (↑ 3.25)	90.82 (↑ 2.00)	217.1	3.51M
	+ ODConv (4×)	73.81 (↑ 4.63)	91.33 (↑ 2.51)	226.3	7.50M
	+ SD-Conv	73.20 (↑ 4.02)	-	171.8	5.0M
	+ LADConv (1×)	73.70 (↑ 4.52)	91.03 (↑ 2.21)	252.5	3.48M
	+ LADConv (4×)	74.08 (↑ 4.90)	91.39 (↑ 2.57)	261.3	7.47M

4.5.6 可视化展示

前文中介绍了 LADConv 的结构更匹配卷积核的归纳偏置，因为它将局部信息引入到动态核的计算中。这种特点也能在 IC-CNN+ 的基础模块中拟合更准确的分割超曲面。为了更深入地了解 LADConv 在局部特征表示上的优越性，本章进行了一项可视化分析，如图 4-4 所示，不同模型的激活图^[81]被可视化出来。本章选择分类效果次好的 ODConv 进行对比。图 4-4 从 ImageNet 验证集中随机选择了一些图像进行可视化。通过对比可以发现，LADConv 在捕获精确的物体细节方面表现出色，同时最大程度地减少了非物体区域的干扰。可视化结果进一步验证了将 LADConv 应用于 IC-CNN+ 中的科学性。

4.5.7 消融实验

本章的消融实验分为几个方面：1. 是否 IC-CNN+ 也可以收益于其它的动态卷积方法？2. LADConv 方法中的性能提升是否真的源自他可以提取局部特征？3. LADConv 中的超参数对模型性能有什么影响？

IC-CNN+ 中的动态卷积方法。首先，将不同的动态卷积方法应用到 IC 卷积结构中，包括 DyConv，ODConv 和 LADConv。其中 DyConv 使用 (4×) 结构，ODConv 和 LADConv 均使用 (1×) 结构。实验结果如表 4-7 所示，可以观察到相

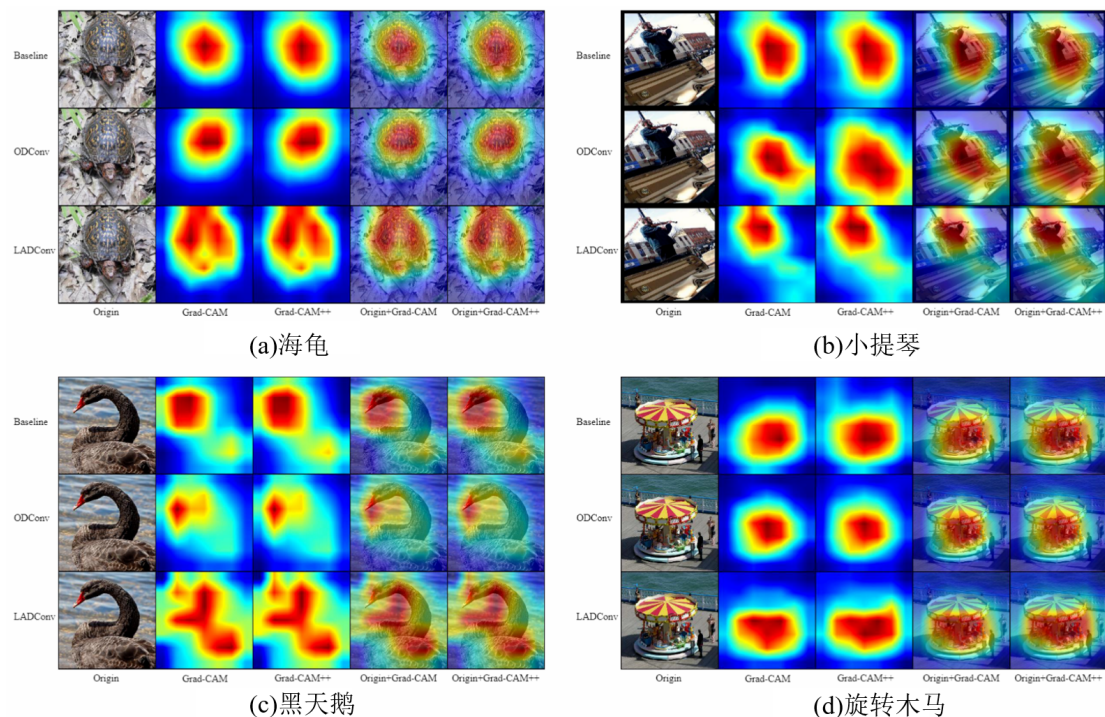


图 4-4 基于梯度的类激活图 (Grad-CAM)^[81]的可视化特征。所使用的基础模型是 ResNet-18。

比于基础 IC 卷积结构，动态化方法均能有效提升神经网络的性能，间接表明了“超曲面更适合拟合高维数据分布”这一论证。此外，所有动态化方法中，LADConv 对 IC 卷积提升的幅度最大，在 ResNet-18/ResNet-50 上超过 ODConv 方法 0.20%/0.28%。

表 4-7 IC-CNN+ 中动态卷积方法的消融实验。

Backbone	Model	Top-1 Acc (%)	Backbone	Model	Top-1 Acc (%)
ResNet-18	Baseline	70.25	ResNet-50	Baseline	76.23
	+IC	70.77		+IC	76.73
	+(IC+) (DyConv)	71.54		+(IC+) (DyConv)	76.94
	+(IC+) (ODConv)	72.18		+(IC+) (ODConv)	77.21
	+(IC+) (LADConv)	72.38		+(IC+) (LADConv)	77.49

LADConv 中局部特征的作用。本章设置消融实验来判断 LADConv 性能的提升是否来自它对局部特征的捕捉。由于 LADConv 模型是在压缩模块提取了局部特征，本章将压缩模块替换为其它一些方法进行实验，包括 SE^[10]、ECA^[77]和 Fca^[82]中的压缩模块。为了公平比较，AAP 中的 s 被设为 1，并移除了自注意力操作和位置注意力。所有压缩模块都用于生成具有相同维度的特征描述符。该消融实验基本模型使用的基础网络是 ResNet-18。

实验结果如表 4-8 所示，LADConv 自身的压缩模块在相似计算消耗的情况

表 4-8 不同压缩模块的消融实验。

Squeeze Module	Top-1 Acc (%)	Flops (GMac)	Params
Baseline	70.25	1.81	11.69M
SE	73.10	1.84	11.94M
ECA	72.94	1.84	11.69M
Fca	73.07	1.84	11.94M
(Ours)	73.49	1.84	12.58M

下，优于 SE 模块（第二好的结果）0.39%。对比的方法都使用全局池化将像素压缩至一维，他们可以被视为提取了所有 1×1 区域的局部特征。然而， 1×1 的局部特征不足以全面描述 $k \times k$ 的局部区域的特性，导致他们在指导 $k \times k$ 动态卷积核行为的时候效果不如 LADConv 自身的压缩模块。结果表明，LADConv 性能的提升与它捕获局部特征有关。

超参数配置。 LADConv 模型只引入了两个额外的超参数：AAP 在空间维度的采样数 s 和缩减系数 r 。本章采用 ResNet-18 基础模型进行消融实验。首先评估了 r 的影响。如表 4-9 所示，配置 $r = 8$ 和 $r = 16$ 的结果类似，优于 $r > 16$ 的配置。因此，在基本的 LADConv 结构（用于 ResNets）中，默认采用配置 $r = 16$ 。

超参数 s 表示局部特征的多样性。本章实验基于 ResNet-18 基础网络来寻找最优超参数 s 的经验值。从表 4-9 中的结果可以看出，配置 $s \geq 5$ 比 $s < 5$ 带来了更显著的改进。然而，较大的 s 值会增加计算负担。最终，本文选择 $s = 5$ 作为默认设置，在模型准确性和计算效率之间取得平衡，这种配置可以应用在大部分卷积核大小为 3×3 的模型中。

表 4-9 LADConv 超参数的消融实验。

r	s	Top-1 Acc (%)	Flops (GMac)	Params
8	5	73.52	1.84	12.64M
16	5	73.55	1.84	12.85M
32	5	73.41	1.84	12.45M
16	2	73.35	1.84	12.64M
16	3	73.43	1.84	12.64M
16	8	73.52	1.85	12.64M
16	16	73.58	1.87	12.64M

4.6 本章小结

本章中发现了高维表示约束 IC 卷积结构的问题。通过深入探索，定位到问题在于 IC 神经元中的线性超平面难以拟合高维空间中的复杂分布。针对此问题，

本章应用动态卷积技术将 IC 卷积结构中的超平面替换为一可学习的超曲面，构造了一种高效结构 IC-CNN+。此外，本章也发现现有的动态卷积缺乏对局部特征的刻画，导致动态卷积核捕捉局部特征的能力较差，针对此问题，本章提出了一种新型动态卷积称为 LADConv。在实验中，与 LADConv 结合的 IC-CNN+ 在实验中表现出良好的泛化性能，LADConv 也可以单独作为一种卷积结构提升基础网络的性能。

第五章 一种基于 IC 神经元的近似等变图神经网络算法

5.1 背景介绍

第三章介绍了 IC 神经元及其简单应用。本章致力于将其与图神经网络相结合，用于提取一些 3D 数据中的特征并解决下游任务。随着自动驾驶和药物研发领域的快速发展，3D 数据的特征提取目前收到广泛的关注，例如点云或分子数据。图神经网络是处理 3D 数据的主流神经网络，然而，图神经网络在提取这些数据的特征时需要考虑空间中的 $SO(3)$ 变换不变性 (Invariance) 或等变性 (Equivariance) 约束。传统的图神经网络和 IC 神经元都不具有学习等变性以及不变性的能力，导致模型提取的 3D 特征不能应用到下游任务中。

本章集中于设计能处理 3D 数据 IC 图神经网络。大部分 3D 任务对等变性和不变性有硬性要求，关于等变性和不变性的介绍可以回顾第 2.5.3 章，旋转不变性本质上是旋转等变性的一种特殊情况。因此，在后文中主要考虑等变性的实现。目前研究人员主要使用两种方法让图神经网络学习到 3D 数据的等变特征：

- 手工提取不变特征，并将这些特征输入到普通神经网络中进行学习，等变特征由等变的方向向量与不变特征相乘得到^[83-84]。
- 在图神经网络中用一种等变神经元^[42]代替 MP 神经元去学习等变特征，该神经元依赖群表示 (Group Representation) 技术，内部的操作都满足严格的等变性。

第一种方法对手工特征有很高的要求，两点之间的距离，夹角，二面角都属于不变手工特征，当手工特征类型越丰富时，神经网络学习到的特征也更细致。然而，手工特征存在上限，对于一些复杂任务，网络设计者很难将所有特征都考虑到。这一类方法最有代表性的是 2020 年提出的 GemNet^[84]，它最高考虑到二面角的手工特征，无法捕获更高阶的特征。第二种方法直接将原始坐标输入到神经网络，通过等变神经元自动地去学习一些高阶特征，包括节点之间的相互

作用以及局部集合特征。首先，等变神经元的输入数据必须满足等变群表示，一般用多组球谐函数基底组合表示 (详情见第 2.5.3 章)。然后，等变神经元用等变线性层代替 MP 神经元的线性变换，而激活函数部分用 Gate 操作和 CG 张量积 (Clebsch-Gordan Tensor Product) 代替^[43]。等变神经元能够保证输出的严格等变性，然而，该神经元中的严格等变操作会削弱图神经网络的表达能力^[85-86]，使其无法拟合一些复杂的特征信息。

本章通过对比了 MP 神经元，IC 神经元和等变神经元在图神经网络中的优劣，提出了一种基于 IC 神经元的近似等变结构。使用新型 IC 图神经元构建的模型被称为 IC-GNN+，它可以有效应用于多种 3D 数据，提取有效的特征，本章贡献总结如下：

- 分析了几种神经元在等变性要求下的优点和缺点，明确 IC-GNN+ 的设计动机；
- 结合 IC 神经元与等变群表示技术，设计了一种近似等变结构 IC-GNN+，可以用来提取 3D 数据中的等变特征。
- 实验验证了 IC-GNN+ 的有效性，在分子以及点云 3D 数据上，IC-GNN+ 都能提取有效的特征，表现出强大的泛化能力。

5.2 神经元的等变性分析

大部分 3D 任务中需要神经网络输出的特征保证等变性 (包含不变性)，同时又要求神经网络有能力拟合潜在的数据分布。对于等变性约束，第 2.5.3 章介绍了三维空间中的旋转等变性，简单来说，神经网络模型往往由一系列操作串联组成，当所有串联结构都是等变时，整个模型是等变的。对于拟合数据分布的能力，图神经网络学界通常用 GWL 图同构测试^[86]来定量的评估模型。GWL 图同构测试与模型的表达能力有直接关系，后文都用表达能力 (Expressive power) 这一术语来描述图神经网络拟合数据分布的能力。在图神经网络领域，表达能力可以用来 GWL 同构图测试^[86]评估。Nadav Dym 和 Haggai Maron 在 2020 年提出了相关理论^[85]来反映等变神经元中表达能力的上限。

定理 5.2.1 ^[85] 每个连续的 G -等变函数都可以被 $\cup_{l \in \mathbb{N}_+} F_{C(l),l}^{TFN}$ 中的函数逼近。

该理论中, l 表示群表示中的主量子数 (阶数), 即等变表示的解数 (详情请阅读第 2.5.3 章), 当阶数越大时, 输入维度越高, 等变操作的计算消耗更高。 F^{TFN} 表示 TFN 模型^[42]中的计算单元, 即前文提到的等变神经元结构。该理论中所展示的逼近描述了一种过程: 给定目标等变函数 $f(\cdot)$, 对于任意小值 ϵ , 总能用 TFN 表示出一个函数与 $f(\cdot)$ 的差值小于 ϵ 。

以上定理表明等变神经元有能力拟合任何等变函数, 然而, 它基于一个重要的前提: 等变表示的阶数 l 可以取到无穷大。在实际应用中, 无穷阶数显然是不可行的。许多工作都只考虑 $l < 4$ 的情况, 因为高阶表示会造成显著的计算消耗提升。Nadav Dym 和 Haggai Maron 在他们的工作^[85]中也证明了基于群表示的等变模型可以通过增加最大阶数来增强它们的表达能力。这一点也在另一项理论工作中得到体现^[86], 作者利用了 GWL 图同构测试分析等变模型是否能识别 n 重对称结构。 n 重对称结构如图 5-1 所示。实验结果如表 5-1 中的 TFN 模型 (等变神经元) 所示。当对称结构的 n 值超过模型的最大阶数 l 时, 模型无法区分对称结构。

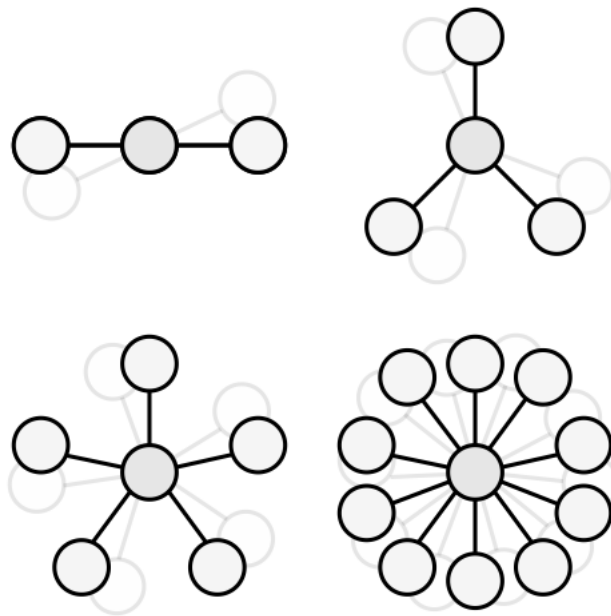


图 5-1 旋转对称模型^[86]。从左上到右下依次为 2 重对称, 3 重对称, 5 重对称, 10 重对称模型。

为了评估非等变操作对表达能力的影响, 本章用 n 重对称实验评估 MP 神经元和 IC 神经元。在等变神经元的实验中, 输入输出必须被编码为不可约表示。公平起见, 本章在 MP 神经元和 IC 神经元实验中也应用相同的输入。实验结果

如表 5-1 所示。MP 神经元和 IC 神经元可以通过低阶不可约表示区分高阶对称结构，突破等变神经元的限制。但这两种神经元也无法覆盖所有的高阶对称结构，例如，当 $l = 1$ 时，MP 神经元和 IC 神经元无法区分 10 重对称结构。此外，由于 IC 神经元对未知函数更强的拟合能力，它在区分对称结构的实验中略好于 MP 神经元。

表 5-1 等变神经元，MP 神经元，IC 神经元的 n 重对称实验。

Model	2-fold	3-fold	5-fold	10-fold
$TFN_{l=1}$	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0
$TFN_{l=2}$	100.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0
$TFN_{l=3}$	100.0 ± 0.0	100.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0
$TFN_{l=5}$	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	50.0 ± 0.0
$TFN_{l=10}$	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
MP_0	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0
$MP_{l=1}$	97.5 ± 2.5	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0
$MP_{l=2}$	100.0 ± 0.0	100.0 ± 0.0	87.5 ± 12.5	82.5 ± 17.5
$MP_{l=3}$	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
$MP_{l=5}$	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
$MP_{l=10}$	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
IC_0	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0
$IC_{l=1}$	100.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0	50.0 ± 0.0
$IC_{l=2}$	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	87.5 ± 12.5
$IC_{l=3}$	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
$IC_{l=5}$	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
$IC_{l=10}$	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0

尽管 MP 神经元和 IC 神经元能够突破等变神经元表达能力的限制，但是直接应用他们会过度损害模型整体的等变性，在未知数据上表现出较差的泛化性能。例如，在表 5-1 中，MP 和 IC 神经元只能学习到特定形状的对称性质，当该形状发生旋转时，MP 和 IC 神经元会失效。表 5-2 记录了几种神经元的对比。表中复杂度只考虑除线性操作外的最大复杂度，由于用到了张量积，等变神经元的复杂度也远高于 MP 神经元和 IC 神经元。本章的工作旨在用 IC 神经元设计一种兼顾等变性和表达能力的神经元模型。

表 5-2 几种图神经元性能的对比。

模型	表达能力	等变性	时间复杂度
等变神经元 (TFN)	受阶数 l 限制	严格等变	$O(l^3)$
MP 神经元	不受阶数 l 限制	不等变	$O(l)$
IC 神经元	不受阶数 l 限制	不等变	$O(l)$

5.3 基于 IC 神经元的近似等变模块

基于第 5.2 章的分析，可以发现 IC 神经元虽然具有较强的表达能力，但会严重破坏等变性。本章基于 IC 神经元提出一种近似等变变体 IC-GNN+。

IC-GNN+ 核心的思想是将 IC 神经元结构与传统等变群表示结合到一起，利用 IC 神经元灵活的特性代替等变神经元中一些严格的等变操作。如图 5-2 所示，传统的等变模块依赖于 CG 张量积来提取图神经网络中的消息变量，IC-GNN+ 首先用局部坐标系旋转将 CG 张量积简化，然后用 IC 神经元代替其中的一些操作。下文将对模型的细节进行介绍。

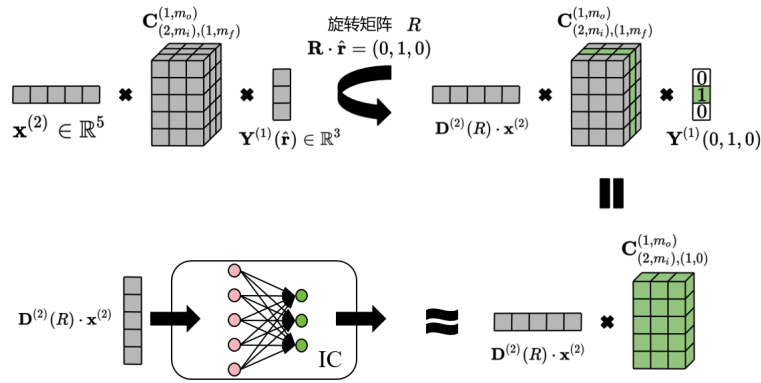


图 5-2 IC-GNN+ 模型的示意图。通过旋转到局部坐标系可以简化 CG 张量积的形式。简化后，IC-GNN+ 利用 IC 神经元近似表示 CG 张量积中的核心模块。

5.3.1 节点嵌入向量

首先，图神经网络中每一个节点都用一组嵌入向量 \mathbf{x} 表示。为了让其匹配等变的群表示，本章用球谐函数基底来刻画 \mathbf{x} 。具体来说，本章定义了球谐函数的最大阶数 L ，嵌入向量需要包含 $[0, L]$ 内的所有球谐函数。对于每个阶数 l ，一共有 $2l + 1$ 个球谐函数。因此， $[0, 1, \dots, L]$ 总共有 $(L + 1)^2$ 个基函数（详情请参考第 2.5.3 章）。此外，本章将嵌入向量 \mathbf{x} 扩张为 C 通道的张量，它的大小表示为 $(L + 1)^2 \times C$ 。图神经网络中，每一层通过计算节点之间的相互作用来更新节点嵌入向量。为了方便描述，下文中，第 k 层第 i 个节点的嵌入向量表示为 \mathbf{x}_i^k 。在模型的最开始，本章对嵌入向量 \mathbf{x}^0 进行初始化，其中 $l = 0$ 维度上的值设置为 1，或由节点内置属性编码， $l > 0$ 的值设置为 0。由于 Type-0 向量（Type- l 向量

的定义参考第 2.5.3 章) 属于不变特征, 因此, \mathbf{x}^0 是对于空间 $\text{SO}(3)$ 变换严格等变的。

5.3.2 近似等变的消息计算模块

在提取 3D 特征时, 图神经网络将提取节点周围每个邻居的特征, 然后汇总到一起更新每个节点:

$$\begin{aligned} \mathbf{m}_{ij}^k &= \mathbf{M}^k(\mathbf{x}_i^k, \mathbf{x}_j^k, z_i, z_j, \mathbf{r}_{ij}) \\ \mathbf{x}_i^{k+1} &= \mathbf{U}^k(\mathbf{x}_i^k, \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij}^k), \end{aligned} \quad (5-1)$$

其中 $\mathcal{N}(i)$ 表示第 i 节点的邻居集合。 \mathbf{m}_{ij}^k 表示第 i 节点为中心, 邻居第 j 节点的特征, \mathbf{M}^k 和 \mathbf{U}^k 是消息计算模块和消息更新模块。 \mathbf{M}^k 用来提取节点之间的消息特征, 并整合到一起, 是整个图神经网络的关键。 \mathbf{U}^k 用于将整合后的消息更新到节点的嵌入向量中。

本文重点关注 \mathbf{M}^k 的构建。首先本章给出用等变神经元构建 \mathbf{M}^k 模块的传统做法:

$$\mathbf{m}_{ij} = (\mathbf{x}_i + \mathbf{x}_j) \otimes \mathbf{S}^L(\mathbf{r}_{ij}), \quad (5-2)$$

其中 \otimes 表示 CG 张量积。上式在节点的嵌入向量与相对坐标向量 \mathbf{r}_{ij} 的球谐表示间进行张量积, 这种方法可以有效地提取节点间的等变特征。之前的一些工作^[42,87]的等变图神经网络都将方程 (5-2) 中的结构作为提取节点特征的核心操作。

接下来, 本章介绍如何用 IC 神经元改造方程 (5-2) 的结构。为了方便描述, 在下文中使用 $\mathbf{D}(\mathbf{R}_{ij}) = \mathbf{D}^0(\mathbf{R}_{ij}) \oplus \mathbf{D}^1(\mathbf{R}_{ij}), \dots, \oplus \mathbf{D}^L(\mathbf{R}_{ij})$ 来表示一个从阶数 0 到 L 的 Wigner-D 矩阵的块对角矩阵。 \mathbf{R}_{ij} 表示将 \mathbf{r}_{ij} 转换为固定方向 $[0, 0, 1]$ 的三维旋转矩阵。需要注意的是, 有多个旋转矩阵 \mathbf{R}_{ij} 可以将 \mathbf{r}_{ij} 旋转到 $[0, 0, 1]$ 。本章在应用时随机选择其中一种, 这种随机性会在后文讨论。

首先, 将 (5-2) 展开为 $\mathbf{x}_i \otimes \mathbf{S}^L(\mathbf{r}_{ij}) + \mathbf{x}_j \otimes \mathbf{S}^L(\mathbf{r}_{ij})$, 并且每一项可以转化为

$$\mathbf{x}_i \otimes \mathbf{S}^L(\mathbf{r}_{ij}) = \mathbf{D}^{-1}(\mathbf{R}_{ij})(\mathbf{x}'_i \otimes \mathbf{S}(\mathbf{C})), \quad (5-3)$$

其中 $\mathbf{x}'_i = \mathbf{D}(\mathbf{R}_{ij})\mathbf{x}_i$, $\mathbf{C} = [0, 0, 1]$ 是一常数向量。方程(5-3)将张量积简化为一个

稀疏矩阵乘法^[88]，因此这种方法能够减少计算消耗，并为神经网络学习 CG 张量积的等变性提供了可能性。这种做法类似于将全局坐标系沿某一规则转换到局部坐标系。之前的工作^[39,89]已经证明，坐标框架的有效旋转可以促进图神经网络中滤波器 (神经网络) 的学习。为了让神经网络学习 (5-3) 的等变性，一个直观的方法是将 $(\mathbf{x}'_i \otimes \mathbf{S}(\mathbf{C}))$ 项替换为一个 IC 神经网络，表示为 $NN(\mathbf{x}'_i)$ 。整个消息计算模块的公式变为

$$\mathbf{x}_i \otimes \mathbf{S}^L(\mathbf{r}_{ij}) \approx \mathbf{D}^{-1}(\mathbf{R}_{ij})(NN(\mathbf{x}'_i)). \quad (5-4)$$

方程(5-4)基于一个假设：即 IC 神经元可以完全学习到 CG 张量积中的运算逻辑。由前文可知，IC 神经网络和 MP 神经网络逼近的函数集是相似的，即 IC 神经网络可以用于拟合任意的连续函数，包含所有等变函数。此外，表5-1中的结果也显示，IC 神经元有能力近似高阶等变神经元的函数形式。因此，本章所基于的假设是合理的。

然而，方程(5-4)中旋转矩阵 \mathbf{R}_{ij} 引入随机性，并且 IC 神经元的一般形式是非等变的，使其无法补偿引入的随机性。因此，用 IC 神经元将会破坏等变性。为了减轻等变性的损失，本章提出了一种应用 IC 神经元的新方法，首先将 (5-3) 转换为

$$\mathbf{x}_i \otimes \mathbf{S}^L(\mathbf{r}_{ij}) = \mathbf{D}^{-1}(\mathbf{R}_{ij})(\mathbf{x}''_i \otimes \mathbf{S}(\mathbf{C}) + \mathbf{x}_i \otimes \mathbf{S}(\mathbf{C})), \quad (5-5)$$

其中 $\mathbf{x}''_i = (\mathbf{D}(\mathbf{R}_{ij}) - \mathbf{I})\mathbf{x}_i$ ， \mathbf{I} 表示单位矩阵。然后，本章使用一个 IC 神经网络 $NN(\mathbf{x})$ 来近似等变函数 $(\mathbf{x} \otimes \mathbf{S}(\mathbf{C}_{ij}))$ 。因此，(5-5) 近似为

$$\mathbf{x}_i \otimes \mathbf{S}^L(\mathbf{r}_{ij}) \approx \mathbf{D}^{-1}(\mathbf{R}_{ij})(NN(\mathbf{x}''_i) + NN(\mathbf{x}_i)). \quad (5-6)$$

方程(5-6)中只用到一个 IC 神经网络，也就是说 $NN(\mathbf{x}''_i)$ 和 $NN(\mathbf{x}_i)$ 的权重是共享的。假设神经网络 $NN(\cdot)$ 能够通过训练拟合高阶 CG 张量积所表示的数学过程，则(5-6)的右侧是等变的。相比于直接应用 $NN(\mathbf{x}')$ 的方法，方程(5-6)解决了两个问题：

- $NN(\mathbf{x}')$ 拆成 $NN(\mathbf{x}''_i)$ 和 $NN(\mathbf{x}_i)$ 两项后， $NN(\mathbf{x}_i)$ 项消除了旋转矩阵 \mathbf{R}_{ij} 中的随机性，提升了整体结构对等变性的学习能力。

- 在方程5-3构建的等变结构中，边(方向)的特征被嵌入在 Wigner-D 矩阵中， $\mathbf{D}^{-1}(\mathbf{R}_{ij})$ 和 $\mathbf{D}(\mathbf{R}_{ij})$ ，顶点的特征被嵌入在 \mathbf{x}_i 中。当输入为 $\mathbf{x}' = \mathbf{D}(\mathbf{R}_{ij})\mathbf{x}$ 一项时，两部分特征被融合到一起，神经网络不易隐式地单独学习其中一种特征 (\mathbf{x}' 无法反推出唯一的 $\mathbf{D}(\mathbf{R}_{ij})$ 和 \mathbf{x}_i)。通过拆分成 \mathbf{x}_i 和 \mathbf{x}_i'' 两项后，神经网络能够区分边嵌入向量和顶点嵌入向量，更有利于深层特征的学习。

方程(5-6)提出了一种基于 IC 神经元的近似等变的结构，该计算单元可以被可视化为图5-3(a)中的结构。在 3D 任务中，需要计算每两个相邻顶点之间的相互作用或代表的几何特征。假设两个顶点的嵌入向量是 \mathbf{x}_i 和 \mathbf{x}_j ，边上的特征向量即为两个点之间的空间向量 \mathbf{r}_{ij} (被编码为 \mathbf{R}_{ij} 和 $\mathbf{D}(\mathbf{R}_{ij})$)。两个顶点之间的特征(消息)可以表示为

$$\begin{aligned} \mathbf{m}_{ij} &= \mathbf{D}^{-1}(\mathbf{R}_{ij})(NN(\mathbf{x}_i'') + NN(\mathbf{x}_i)) + \mathbf{D}^{-1}(\mathbf{R}_{ij})(NN(\mathbf{x}_j'') + NN(\mathbf{x}_j)) \\ &= \mathbf{D}^{-1}(\mathbf{R}_{ij})(NN(\mathbf{x}_i'') + NN(\mathbf{x}_i) + NN(\mathbf{x}_j'') + NN(\mathbf{x}_j)). \end{aligned} \quad (5-7)$$

得到两两顶点之间的特征后，图神经网络可以更新每个顶点的嵌入向量。

5.3.3 更新模块

上述近似等变操作构成了图神经网络的消息模块。通过聚合消息，可以得到每个顶点的消息总和：

$$\mathbf{m}_i = \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij}. \quad (5-8)$$

更新模块将消息总和与原本的点嵌入进行结合，生成新的点嵌入。本章用到一些已有结构，例如“等变线性层-Gate-等变线性层”的严格等变结构^[43]，或者基于傅里叶变换的近似等变结构^[39]作为更新模块。

5.4 多层 IC 图神经网络

IC-GNN+ 的核心是用一近似等变结构构建图神经网络的消息计算模块。对于一些复杂 3D 任务，需要增加图神经网络的总深度以及每一层中消息计算模块的深度来提升模型容量。总深度表示“消息计算-消息更新”结构的迭代次数，可

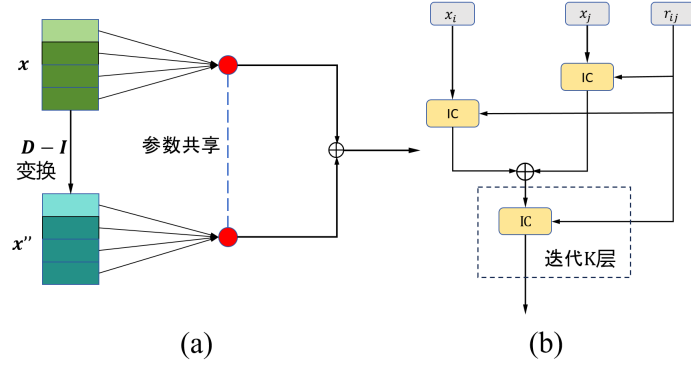


图 5-3 (a) 基于 IC 神经元的近似等变结构。其中红圈表示同一个 IC 神经元模型。(b) 多层消息模块。每一层计算的消息向量与边特征被送入一个 IC 近似等变模块。

以被轻易增加，本章利用方程(5-7)和方程(5-6)在每一个消息计算模块中构建多层神经网络结构，如图 5-3(b) 所示。

消息在进去每个 IC 隐藏层的时候，需要在高维空间旋转到局部坐标系，在输出时，需要旋转矩阵的逆矩阵旋转回正常坐标系。两个操作相互抵消。因此，本章可以省去一些 IC 隐藏层之间的旋转和逆旋转操作，进一步减少计算量。此外，在一些 3D 任务中，图节点本身表示一些特有属性。例如在分子数据中，每一个节点都具有固定的原子类型；在 RGB 点云中，每一个节点都具有固定的颜色属性。在处理这种数据时，需要在图神经网络的每一层输入中都加入这些信息。假设节点 i 和节点 j 的特有属性的编码为 \mathbf{e}_i 和 \mathbf{e}_j ，本章通过以下方式将特有属性引入图神经网络每层的计算中：

$$\begin{aligned}
 \mathbf{c}_{ij} &= NN'(\mathbf{e}_i + \mathbf{e}_j) \\
 \mathbf{m}_{ij} &= NN(\mathbf{x}_i'') + NN(\mathbf{x}_i) + NN(\mathbf{x}_j'') + NN(\mathbf{x}_j) \\
 \mathbf{m}_{ij} &:= \mathbf{m}_{ij} \circ \mathbf{c}_{ij} \\
 \mathbf{m}_{ij} &:= \mathbf{D}^{-1}(\mathbf{R}_{ij})\mathbf{m}_{ij}.
 \end{aligned} \tag{5-9}$$

上式中， $NN'(\cdot)$ 表示另一组 IC 神经网络，具有不同的权重。 \mathbf{c}_{ij} 是由节点特有属性编码得到的特征，一般满足旋转不变特征。当 $NN(\cdot)$ 拟合的函数具有等变性或近似等变性时，因此， $NN(\cdot)$ 生成的消息在旋转到局部坐标系后会保持不变性或近似不变性。不变特征彼此之间的乘积依然是不变的，并且相比于等变特征，神经网络学习不变特征内的深度信息更为容易。因此，方程中(5-9)中将 \mathbf{c}_{ij} 与旋转后的等变特征进行相乘，再进行逆旋转。注意 \mathbf{c}_{ij} 属于不变向量，使用时可将

其看作一种 Type-0 向量并与消息的通道维度进行相乘。举个例子， \mathbf{c}_{ij} 的大小一般为 $C \times 1$ ，消息特征的大小为 $C \times (L + 1)^2$ ，他们在通道维度上进行乘法，这种做法不会破坏消息的等变性。

基于上述思想，本章可以构建多种类型 IC 图神经网络。由于 IC-GNN+ 只关注模型本身的建设，在实际任务中，它可以与多种已有技术进行结合，例如，点云的分类和分子能量预测都需要将所有顶点特征转换为全局特征，传统方法利用所有顶点上的全局平均池化来提取全局特征，在实验中，本章将一些 3D 数据的技巧结合到 IC-GNN+ 上。

5.5 实验结果与分析

本章实验重点验证 IC-GNN+ 在等变 3D 任务中的表现性能。此外，本章通过一系列消融实验验证近似等变性是否有助于提取更适合 3D 任务的特征。

5.5.1 数据集与基线配置

本章实验采用了两种 3D 数据集：分子数据集和点云数据集。

OC20 分子数据集：OC20 分子数据集是一个用于训练模型以预测结构松弛期间的力和能量的大型分子数据集。该数据集包含超过 130 百万个分子结构，并使用 CC Attribution 4.0 许可证。这些分子结构的数据用于开发分子动力学模拟和量子化学计算等领域的机器学习模型。OC20 数据集的建立旨在促进材料科学、化学和计算机科学等领域的研究和应用。本章将实验设定在 OC20 的 IS2RE 任务上，该任务以能量预测为核心，使用的评价指标为 MAE 和 EWT(%), 其中，EWT(%) 表示预测后的能量与真实能量差小于阈值的比例，反映了预测能量的可用性。

QM9 分子数据集：该数据集提供了 134k 个稳定的小有机分子的量子化学属性，这些分子包含多达 29 个原子，构成了一个相关、一致且全面的化学空间。每个原子都用 3D 位置坐标和其原子类型的嵌入描述 (H、C、N、O、F)。具体来说，QM9 数据集包含了包括几何结构、原子核坐标、原子类型、原子能量、振动频率、电子能量和热力学性质在内的大量信息。该数据集对应的任务是通过容易获取的原子坐标和原子类型预测一些难以获得的分子属性，包括原子能量，

表 5-3 OC20 实验的超参数配置

超参数	数值或配置
学习率调整策略	在第 10、14、16、18 个 epoch 时乘以 0.3
warm up 步数	100
最大学习率	4×10^{-4}
批量大小	32
epoch 数	20
权重衰减	0.5×10^{-3}
截断半径 \tilde{A}	8
IC 神经网络的隐藏层大小	512
图神经网络层数 K	16
每一层中隐藏层的层数	2
IC 神经网络的最大阶数	6
CG 张量积的最大阶数	2
最大阶数	6
最大邻居数	40

电子能量等。不同 OC20 数据集，QM9 数据集的规模要小很多，学习等变性的难度更大。

ModelNet40 点云数据集：ModelNet40 是一个用于三维物体识别的点云数据集，其中包含 40 个不同的物体类别。该数据集由康奈尔大学的研究人员创建，旨在为研究人员提供用于训练和评估三维物体识别算法的标准数据集。

模型配置：在 OC20 和 QM9 的实验中，本章将 IC-GNN+ 建立在 2022 年提出图神经网络 SCN^[39] 框架上，SCN 的更新模块是将等变群表示用傅里叶变换转换到时域中，再在时域上应用 1×1 卷积提取特征，这也是一种近似等变操作，基于傅里叶变换中的卷积理论。本章沿用它的更新模块，将它的消息模块中的计算单元替换为方程(5-9)的形式。此外，本章还用在计算消息时引入了严格等变的 CG 张量积操作，该操作作用在点嵌入向量与边嵌入向量之间，它的输出与 IC 神经网络 (即方程(5-9)) 的输出进行求和得到最终的消息。该操作可以增强整体模型的等变性，当 IC 神经网络没有学习到准确的等变性时，这个操作可以对等变性进行补偿。此外，严格等变操作的分支用到的最大阶数为 L' ，本章一般将此值设置较小以防止 CG 张量积占用过多计算资源。在求和时，本章只需将 IC 神经网络输出和 CG 张量积输出的对应阶数相加。其它相关配置记录在表5-3和5-4中。

在 ModelNet40 的实验中，本章基于 DGCNN 模型^[90]的宏观结构设计图神经网络。在 DGCNN 中，点云经过 4 个 EdgeConv 层得到点集特征。本章将 EdgeConv 层替换消息传递结构。其中消息模块由两层近似等变的 IC 神经网络层组成。而

表 5-4 QM9 实验的超参数设置。

超参数	值或描述
学习率调整策略	余弦学习率
warm up 轮数	5
最大学习率	1.5×10^{-4}
批量大小	128
epoch 数	400
权重衰减	0.5×10^{-3}
截断半径 \tilde{A}	5
IC 神经网络的隐藏层大小	512
图神经网络层数	12
IC 神经网络的最大阶数	4
CG 张量积的最大阶数	3

更新模块采用简单的“等变线性层-Gate-等变线性层”组成^[43]。此外，本章保持了 DGCNN 的超参数设置，对于点特征，将其扩充为 $l = 2$ 的表示。举个例子，原始 DGCNN 的第一层将 $n \times 3$ 的点集转换为 $n \times 64$ 的点特征。对应的 IC-GNN+ 将第一层改为 $n \times 9 \times 1$ 的张量，并用第 5.3.1 章的方法初始化该张量中的值。第一层的输出变为 $n \times 9 \times 64$ 的张量，其中，9 所在的维度表示 $l = 2$ 的 Type-2 向量，64 所在的维度表示通道数，与原始 DGCNN 相对应。在最后将特征用于分类时，本章将用点特征的 Type-0 向量部分用于分类，因为 Type-0 对应不变特征。优化器和训练策略的配置遵循之前的工作^[90]。

5.5.2 分子数据集实验结果

在分子数据集中，本章实验对比的基线方法包括、SEGNN^[41] 和 TFN^[42]，Dimenet++^[91]，SchNet^[83]，PaiNN^[92]。在 OC20 任务中，本章实验还包括了额外的基线 GemNet^[37-38]、SphereNet^[93] 和 SCN^[39]。在 QM9 任务中，本章实验还将 L1Net^[94]、Cormorant^[95]、LieConv^[96]、TorchMD-NET^[97] 和 EQGAT^[98] 与 IC-GNN+ 进行比较。

OC20 IS2RE 数据集。 IS2RE 问题可以通过两种方法解决：1. 直接从初始原子结构预测放松能量；2. 使用由预测的力计算得到的放松结构来预测能量。第二种方法往往更准确，但需要从其他更大的数据集中训练得到高效的力模型。本章专注于第一种方法，并且不使用预训练模型。测试集被分为四个子集，包括从与训练相同的分布中采样（ID）、未见过的吸附剂（OOD Ads）、未见过的催化剂（OOD Cat）以及未见吸附剂和催化剂（OOD Both）。它们的大小相似。如

表 5-5 OC20 IS2RE 数据集上的能量预测效果.

Model	Energy MAE (meV) ↓					EwT (%) ↑			
	ID	OOD Ads	OOD Cat	OOD Both	Average	ID	OOD Ads	OOD Cat	OOD Both
Median baseline	1750	1879	1709	1664	1750	0.71	0.72	0.89	0.74
CGCNN	615	916	622	851	751	3.40	1.93	3.10	2.00
SchNet	639	734	662	704	685	2.96	2.33	2.94	2.21
PaiNN	575	783	604	743	676	3.46	1.97	3.46	2.28
TFN (SElin)	584	766	636	700	672	4.32	2.51	4.55	2.66
GemNet-dT	527	758	549	702	634	4.59	2.09	4.47	2.28
DimeNet++	562	725	576	661	631	4.25	2.07	4.10	2.41
GemNet-OC	560	711	576	671	630	4.15	2.29	3.85	2.28
SphereNet	563	703	571	638	619	4.47	2.29	4.09	2.41
SEGNN	533	692	537	679	610	5.37	2.46	4.91	2.63
SCN	516	643	530	604	573	4.92	2.71	4.42	2.76
IC-GNN+	526	633	528	568	564	4.85	2.74	4.44	2.79

表 5-5 所示, IC-GNN+ 在平均能量 MAE 方面优于所有先前的方法。此外, 值得注意的是, IC-GNN+ 在未见分布上明显优于其他方法, 这可以通过 OOD 任务上明显较低的平均 MAE (IC-GNN+: 576 vs. SCN: 592) 来证明。该实验可以推论出 IC-GNN+ 在未见分布上的提升性能可能源于学习到与实际物理属性密切相关的特征。此外, IC-GNN+ 在 OOD Ads 和 OOD Both 中的 EwT 指标方面取得了最佳结果。

QM9 数据集。 QM9 数据集中多个化学属性的预测结果如表 5-6 所示, IC-GNN+ 在四个任务上取得了最佳结果, 在一个任务上取得了第二好的结果。本章对所有任务使用相同的架构, 而 TorchMD-NET 和 PaiNN 则针对 μ 和 R^2 任务采用了不同的架构, 这可能导致了更准确的归纳偏置。此外, HDGNN 在能量变量 (G 、 H 、 U 、 U_0) 任务上落后于 SOTA 方法。这可能归因于这些目标可能受益于更多的架构, 包括注意力机制、邻居-邻居交互和问题定制架构, 这些架构在对比方法中被使用^[99]。

表 5-6 在 QM9 数据集上的各种化学属性结果。† 表示使用不同的数据分区。粗体和下划线分别表示最佳结果和第二佳结果。

Task Units	α bohr ³	Δ meV	ϵ_{HOMO} meV	ϵ_{LUMO} meV	μ D	C_v cal/(mol K)	G meV	H meV	R^2 bohr ³	U meV	U_0 meV	ZPVE meV
SchNet	.235	63	41	34	.033	.033	14	14	.073	19	14	1.70
Cormorant†	.085	61	34	38	.038	.026	20	21	.961	21	22	2.02
L1Net	.088	68	46	35	.043	.031	14	14	.354	14	13	1.56
LieConv†	.084	49	30	25	.032	.038	22	24	.800	19	19	2.28
TFN†	.223	58	40	38	.064	.101	-	-	-	-	-	-
DimeNet++	.044	<u>33</u>	25	20	.030	.023	8	7	.331	<u>6</u>	<u>6</u>	1.21
PaiNN	<u>.045</u>	46	28	20	<u>.012</u>	<u>.024</u>	7.35	5.98	<u>.066</u>	5.83	5.85	1.28
TorchMD-NET	.059	36	20	18	.011	.026	<u>7.62</u>	<u>6.16</u>	.033	<u>6.38</u>	<u>6.15</u>	1.84
SEGNN†	.060	42	<u>24</u>	<u>21</u>	.023	.031	15	16	.660	13	15	1.62
EQGAT	.053	32	<u>20</u>	16	.011	<u>.024</u>	23	24	.382	25	25	2.00
IC-GNN+	.046	32	18	16	.017	.023	15	17	.396	9.67	10.21	<u>1.23</u>

QM9 数据集比 OC20 数据集要小得多,这使得 IC-GNN+ 这种近似等变模型在学习等变性时容易出现过拟合的情况。在 QM9 实验中,本章使用的对比方法都是严格等变的,在这种情况下,IC-GNN+ 依然能在部分任务上取得最优结果。

综上所述,IC-GNN+ 在分子数据集上可以取得优异的成果,对于一些较小的数据集,IC-GNN+ 也能有效学习到等变性,使模型表现出优秀的泛化性能,与最先进的严格等变模型相比也不落下风。

5.5.3 点云数据集实验结果

本章基于 DGCNN 的网络框架构建点云上的 IC-GNN+。为了全面对比,本章实验不止与基础模型 DGCNN 对比,还与一些流行的点云骨干模型对比,包括 PointNet^[100], PointNet++^[101], Kd-net^[102], DensePoint^[103], LDGCNN^[104], KPConv^[105], SPH3D-GCN^[106], FKACnv^[107]和 ConvPoint^[108]。

表5-7显示了 ModelNet40 上的分类任务的结果,IC-GNN+ 在这个数据集上取得了最好的结果。表格中,标注 (2048 pts) 的模型是建立在 2048 个点的点云模型上,其它的模型都是基于 1024 个点的点云。相比于基础模型 DGCNN, IC-GNN+ 在 1024 点上提升了 0.3%/0.2%。在点集更多的 2048 点模型中,IC-GNN+ 表现出更明显的提升,准确率上涨了 0.3%/0.5%。

表 5-7 ModelNet40 数据集中的分类效果。

Method	m-acc(%)	ov-acc(%)
PointNet	86.0	89.2
PointNet++	-	90.7
Kd-net	-	90.6
DensePoint	-	93.2
DGCNN	90.2	92.9
DGCNN (2048 pts)	90.7	93.5
LDGCNN	90.3	92.9
KPConv-rigid	-	92.9
KPConv-deform	-	92.7
SPH3D-GCN (2048 pts)	88.5	91.4
FKACnv	89.9	92.5
FKACnv (2048 pts)	89.7	92.5
ConvPoint	88.5	91.8
ConvPoint (2048 pts)	89.6	92.5
IC-GNN+	90.5	93.1
IC-GNN+ (2048 pts)	91.0	94.0

不同于分子数据,点云数据有时并不满足严格的等变性:当用不同的激光雷达从同一角度对同一物体进行扫描时,噪声会导致点的位置并不完全相同,此

时，人们会更关注局部区域的等变性。IC-GNN+ 中的非严格等变性可能会契合点云数据的特性。

5.5.4 消融实验

由于 IC-GNN+ 模型利用放缩的等变性来提升模型表达能力，在消融实验中，本章探索几个重要问题：1. IC-GNN+ 的结构是否能学习到近似等变性？2. 近似等变结构学习到的特征是否比严格等变模块以及无约束学习模块学到的特征更好？此外，本章的消融实验还会探索模型中影响 IC-GNN+ 的因素。

表 5-8 近似等变性的消融实验。

Model	Energy MAE (meV) ↓				Average	ID MAD
	ID	OOD Ads	OOD Cat	OOD Both		
近似等变模型 (IC-GNN+)	560	735	572	651	616	19.9
严格等变模型	591	752	646	704	673	0
无约束模型	883	1052	962	996	973	78.5

近似等变性的意义。首先，本章实验比较了严格等变性和近似等变性。尽管在某些分子任务中，IC-GNN+ 表现出竞争性结果，但重要的是要确定这些改进是否归因于近似等变的设计。为了解决这个问题，本章构建了与 IC-GNN+ 结构类似的严格等变模型。该模型只使用张量积计算消息，此外还构建了一个传统图神经网络，使用 $NN([\mathbf{x}_i; \mathbf{x}_j; \mathbf{S}(\mathbf{r}_{ij})])$ 来计算消息，该模型不受任何等变性约束。这个实验中 IC-GNN+ 以及对比模型的层数设置为 $K = 8$ ，通道数 $C = 64$ ，其它配置与表5-3中一致。严格等变模型的最大阶数设为 $L = 4$ 。从表 5-8 中可以得出几个结论：1. 近似等变模型可以比等变模型获得更好的泛化能力；2. 设计近似等变模型需要仔细考虑，因为无约束的学习模块可能会显著削弱模型的泛化能力。

除了分子数据集外，本章也在点云数据上对近似等变性进行探索，表5-9记录了不同模型在随机旋转测试集后的表现效果，其中 PointNet 和 DGCNN 都是完全不考虑等变性的模型，当测试集发生随机旋转后，它们的性能发生了严重的坍缩，这使得模型输出特征不能满足 3D 任务的需求。相比之下，IC-GNN+ 引入了等变性约束，它的准确率还可以维持在一个较高的水平。该实验进一步验证了近似等变性对 3D 任务的意义。

IC-GNN+ 中的近似等变性的评估。此外，本章实验在表 5-8中测试了 IC-GNN+ 模型的等变性，使用平均绝对差 (Mean Absolute Difference, MAD) 来定

表 5-9 随机旋转测试集下各模型的表现效果。

Method	m-acc(%)	ov-acc(%)
PointNet	52.8	56.4
DGCNN	69.2	72.3
IC-GNN+	80.4	83.1

量的表示等变性。MAD 的数值等于 $|E - E'|$ ，其中 E 是预测能量， E' 是基于随机 $SO(3)$ 变换的预测能量。当模型完全等变时，MAD 的结果应该为 0。表格中的 MAD 数值基于 ID 任务的预测结果。实验结果显示 IC-GNN+ 不能保证严格的等变性，但相比于无约束学习的图神经网络，IC-GNN+ 能达到一种近似的等变性。

影响 IC-GNN+ 的因素。本章实验还对 IC-GNN+ 的组成模块进行分析。实验在 IS2RE 分子数据集上进行。本章构建了三种变体 1. 舍弃旋转到局部坐标系的的操作，直接将 IC 神经元应用到原始群表示上；2. 使用不同的 IC 神经网络处理 \mathbf{x}'' 和 \mathbf{x} ；3. 将 $NN(\mathbf{x}) + NN(\mathbf{x}'')$ 简化为 $NN(\mathbf{x}')$ 。表5-10表明本章应用在 IC-GNN+ 上的操作都有利于提升其泛化性能，尤其是旋转到局部坐标系的的操作。它可以有效降低 IC 神经网络学习到等变性的难度。

表 5-10 IC-GNN+ 组成模块的消融实验。

Model	局部坐标系	共享权重	$NN(\mathbf{x}')$	MAE (meV) ↓
IC-GNN+	+	+	+	560
	-	+	+	728
	+	-	+	594
	+	+	-	572

5.6 本章小结

本章将 IC 神经元应用于图神经网络并尝试处理各类 3D 数据，这面临一个挑战：大部分的 3D 任务都要求特征是等变的，例如分子属性预测和点云分类。本章首先分析了 MP 神经元，IC 神经元和等变神经元的等变性和表达能力。第三章中介绍了 IC 神经元能够引入一种高表达能力的结构，本章在图神经网络中引入 IC 神经元和等变技术，使其在具备高表达能力的同时，能够学习到近似等变特征。通过分子和点云的实验证明 IC 图神经网络在 3D 数据上具有良好的泛化能力。此外，消融实验证明通过充分训练，IC 图神经网络能够逼近严格等变性，满足大部分 3D 任务的需求。

第六章 一种基于 IC 神经元的记忆性递归神经网络算法

6.1 背景介绍

本章第三章提出了 IC 神经元及其简单应用。其中一种应用是将 IC 神经元应用到递归神经网络 (Recurrent Neural Network, RNN) 的计算中。然而，第三章提出的 IC 递归神经网络只是简单将 IC 神经元用于拟合当前输入状态的特征，并未考虑隐藏变量中的历史信息，且 IC 神经元本身也不具备时序信号表示的能力。这在实际应用时会产生一系列问题。

递归神经网络的主要特点是具有递归连接，可以处理序列数据，并在时间步之间传递信息。它常被应用于自然语言处理、时间序列分析、语音识别等任务中。然而，传统的递归结构在处理这些任务时面临一系列挑战，主要包括以下几点：

- 在反向传播过程中，由于循环连接的存在，梯度会反复相乘或相加，导致梯度在时间步长的传播过程中呈指数级衰减或爆炸，使得模型难以训练；
- 传统递归神经网络的隐藏状态只能捕捉到当前时刻之前的信息，而对于较长的序列，模型往往会遗忘之前的信息。在训练过程中，模型可能无法有效地学习长期依赖关系，造成性能衰退。

将 IC 神经元与递归结构简单结合后，模型依然会存在上述问题，尤其是长期依赖关系的低效学习会限制递归神经网络的应用。长期依赖关系指的是文本序列中跨越较长时间跨度的词语或短语之间的关联，这种关系非常常见，具体来说，它包含了两种重要特征：

- 语义连贯性：文本中的词语通常在意义上相互关联，而这种关联不仅仅局限于相邻的词语，还可能延伸到文本的较远部分。例如，在理解一篇文章或对话时，前文中提到的主题可能会在后文中得到进一步解释或延伸。

- 上下文理解：长期依赖关系有助于更好地理解文本中的上下文信息。对于某个词语或短语的理解可能需要考虑文本中前面和后面的内容，以便更准确地把握其含义和语境。

理论上，传统的递归神经网络是能够学习到这些长期依赖关系，但在实际应用中，模型却很难真正捕获准确的关系。目前的研究人员通过两种方法来解决这个问题：1. 在递归神经网络中引入记忆/遗忘机制，使模型可以记忆序列中改的一些关键特征，从而捕获长期依赖关系，例如 LSTM 和 GRU 模型；2. 通过自注意力机制 (Self-attention) 计算序列中任意两个元素之间的关系，从而学习到长期依赖关系。相比而言，两种方法各有优劣，自注意力机制可以捕获序列上任意长的相互关系，而 LSTM 和 GRU 在处理一些距离过长的关系时效果较差。但另一方面，LSTM 和 GRU 更适合学习时序数据中的顺序信息。自注意力机制不能直接用于学习顺序信息，需要和位置编码等技术相结合。

为了让 IC 神经元与递归神经网络的结合具备更广泛的应用能力。本章考虑一个重要问题，是否能将上述提取长期依赖关系的技术与 IC 神经元结合？由此，本章进行了一系列探索，并提出了一种高效递归计算结构 IC-RNN+，本章主要贡献总结如下：

- 分析 IC 神经元在长期依赖关系表示中的局限性；
- 提出一种新型的记忆/遗忘模块，该模块具有更小的尺寸，可以直接应用于 IC 神经元内部。IC-RNN+ 基于这种模块定义，能够学习长期依赖关系；
- 实验证明 IC-RNN+ 在多个时序任务中都表现出优于基础 IC-RNN 的效果。在与 LSTM，GRU，自注意力模型等长期依赖模型的比较中也不落下风。

6.2 长期依赖与 IC 神经元之间的关系

本章构造实验来分析 IC 神经元对长期依赖关系捕获的影响。为了方便理解，本章用斐波那契数列构造时间序列数据，并通过神经网络预测最后时刻的值。具体如下。

斐波那契数组满足如下性质：

$$x_t = x_{t-1} + x_{t-2}. \quad (6-1)$$

这是一个典型的时间序列数据，为了增强其中长期依赖关系的数量，本章只保留前 K 数组元素并使其周期性出现。例如 $K = 4$ 时，数据表示 $[1, 2, 3, 5, 1, 2, 3, 5, \dots]$ 。本章在不同的 K 下生成长度为 100 的时间序列数据，并用前 70% 训练模型，后 30% 验证模型。损失函数使用 MAE。优化器采用 Adam 优化器，学习率为 0.01，训练轮次为 100 轮。

表 6-1 斐波那契数据实验。表中数值表示预测值与真实值之间的 MAE。

Model	$K = 2$	$K = 5$	$K = 10$	$K = 20$
RNN	0.0	0.025	0.032	0.083
IC-RNN	0.0	0.021	0.037	0.076
LSTM	0.0	0.003	0.008	0.014
GRU	0.0	0.005	0.006	0.022

几种递归模型的效果如表6-1所示。可以发现，当周期变长时，RNN 的结果明显变差，并且当周期继续变大时，RNN 的结果会发生严重的坍塌。对于 IC 神经元，它本身不具备提取时序特征的能力，也不会产生对历史状态的记忆，如图6-1所示，IC 神经元本身是无法学习到时序中的历史信息的。因此，IC 神经元 RNN 的简单结合并不能解决核心问题，效果依然较差。

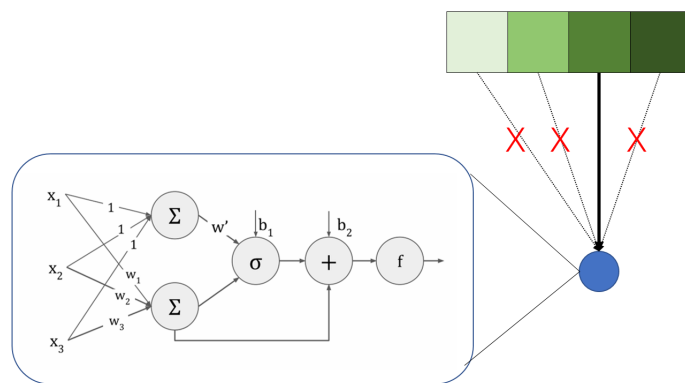


图 6-1 IC 神经元与递归神经元的简单结合。IC 神经元只能用于提取当前时刻下输入信号的特征，不能考虑其它时刻的信号。

LSTM 和 GRU 是两种改进后的递归神经网络结构，他们通过在网络中引入记忆/遗忘机制使递归网络能够保留序列中出现的关键信息。后面的序列单元在计算时都能够感知到这些关键信息，从而捕获长期依赖关系。表6-1中的结果显示，这样的设计能够显著提升模型处理长周期数据的能力。值得注意的是，当依赖关系的距离过长时，LSTM 和 GRU 的性能也会下降，他们并不能处理任意长的依赖关系。

在下文中，本章提出一种神经元层面的记忆/遗忘单元，并将其与 IC 神经元结合，构建一种记忆性递归神经网络 IC-RNN+。

6.3 神经元层面的记忆/遗忘结构

首先，本章回顾一下 LSTM 和 GRU 中的传统记忆/遗忘模块，并分析他们的优缺点，以引出本章的核心模块。

LSTM 通过引入三个门（输入门、遗忘门和输出门）和一个细胞状态来实现对长期依赖的建模。其中细胞状态是需要一个维护的变量，输入门允许模型选择性地更新细胞状态，遗忘门允许模型选择性地遗忘细胞状态中的信息，输出门允许模型选择性地输出细胞状态中的信息。通过这种方式将重要的历史信息保留在细胞状态中，以便未来构建长期依赖关系。

在 GRU 中，不存在像 LSTM 细胞状态这种显式变量来记录关键历史信息，它只有两个门：更新门和重置门。更新门控制着模型决定从前一个状态中保留多少信息，而重置门控制着模型决定忽略多少先前的状态。因此，在 GRU 中，隐藏变量就是关键历史信息的载体，GRU 中的门结构控制关键信息向远处传递下去。

综上所述，LSTM 和 GRU 是通过维护一个有关历史状态的变量实现长期依赖关系的捕捉。而这两个模型都是通过对旧状态的压缩和新信号的学习来维护这个变量，本章将这两个过程称为“遗忘”和“记忆”。

然而，LSTM 和 GRU 中的记忆/遗忘结构存在两个缺陷：

- LSTM 和 GRU 中的遗忘结构是对旧有历史的整体遗忘，这导致一些较长的序列中，开头的关键信息依然会被遗忘，无法构建长期依赖关系。
- LSTM 和 GRU 将捕获的关键历史信息存储到一个向量中，容易造成信息的冗余。对于一些复杂时序数据，可能存在多个关键信息单元以及多个长期依赖关系，LSTM 和 GRU 的细胞状态或隐藏变量无法全面表示它们。

由此，本章提出一种新的记忆/遗忘结构来解决上述问题。如图6-2所示，首先对于每一个神经元，为其赋予两个状态参数 e 和 k ，它们的意义分别为：

- e 表示该神经元存储的信息量；

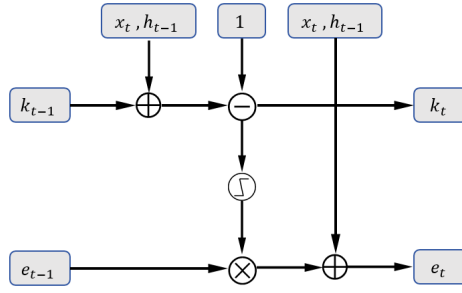


图 6-2 记忆/遗忘结构。

- k 表示该神经元存储信息量的保持时间。

上述两个状态参数均为标量，他们也是神经元层面的参数。当递归神经网络从时刻 $t-1$ 计算到时刻 t 时，首先根据 k 判断当前神经元中的信息是否还需要保留：

$$e_t = \begin{cases} e_{t-1} & \text{if } k-1 > 0 \\ 0 & \text{otherwise.} \end{cases} \quad (6-2)$$

方程(6-2)中假设了模型从时刻 $t-1$ 计算到时刻 t 后，神经元的时间状态过了一个单位。因此，模型判断 $k-1$ 是否还是正值， $k-1$ 不是正值时，表明该神经元需要将它包含的历史状态遗忘。

为了方便表示，方程(6-2)被简化为

$$e_t = I g(e_{t-1}, k_{t-1}). \quad (6-3)$$

构建完遗忘模块，本章将当前时刻的输入 \mathbf{x}_t 与上一时刻的输出 \mathbf{h}_{t-1} 引入，构建记忆模块。全局的记忆/遗忘结构表示为：

$$\begin{aligned} k_t &= \sigma(\mathbf{w}_1 \mathbf{x}_t + \mathbf{u}_1 \mathbf{h}_{t-1}) + k_{t-1} - 1 \\ e_t &= I g(e_{t-1}, k_t) \\ e_t &:= e_t + \lambda(\mathbf{w}_2 \mathbf{x}_t + \mathbf{u}_2 \mathbf{h}_{t-1}) \end{aligned} \quad (6-4)$$

为方便表示和理解，本章省略了偏置项。 $\sigma(\cdot)$ 指 ReLU 函数， $\lambda(\cdot)$ 指 Sigmoid 函数。在方程(6-4)中，首先根据现有输入状态和历史状态更新存储时间参数，来决

定该神经元中的信息是否需要增大保持时间。然后，根据保持时间判断携带信息是否需要被遗忘，最终，对现有输入状态和历史状态进行记忆。

方程(6-4)构建的记忆/遗忘单元是一种神经元层面结构，它维护一个标量 e_t ，在递归神经网络的计算中， e_t 可以将重要的历史信息传递至后续计算单元。通过堆叠多个神经元，可以将多项重要的历史信息传递至后序计算单元。相比于 LSTM 和 GRU 中的计算单元，本章提出的结构具备以下优势：

- 在递归神经网络的计算中， e_t 可以将重要的历史信息传递至后序计算单元。通过堆叠多个神经元，可以将多项重要的历史信息或长期依赖关系传递至后序计算单元。
- 本章提出的记忆/遗忘结构采用了按虚拟时间流逝的方法遗忘信息，在没有达到神经元中信息保持时间时，来自远距离的信息可以被一直保留，有利于捕获更长距离的依赖关系。

6.4 基于记忆/遗忘结构的 IC 递归神经网络

由于方程(6-4)建立在神经元层面，本章将其与 IC 神经元相结合构建高性能的 IC 递归神经网络。为方便后续表述，本章首先简单回顾第三章中的 IC 递归结构。给定一个输入序列 $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_T)$ ，IC 递归神经网络通过迭代以下方程计算 h -维隐藏向量序列 $\mathbf{h} = (\mathbf{h}_1, \dots, \mathbf{h}_T)$ 和输出向量序列 $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_T)$ ：

$$\mathbf{h}_t = \mathbf{f}(\mathbf{W}\mathbf{x}_t + \sigma(\mathbf{W}\mathbf{x}_t - \mathbf{w}'\mathbf{I}\mathbf{x}_t) + \mathbf{U}\mathbf{h}_{t-1}) \quad (6-5)$$

其中 \mathbf{W} ， \mathbf{U} 和 \mathbf{w}' 表示输入和隐藏状态的权重以及调节权重。 $\mathbf{f}(\cdot)$ 表示激活函数， $\sigma(\cdot)$ 指 ReLU 函数。下标 $t \in [1, T]$ 表示第 t 次迭代步骤。

方程(6-5)只是将 IC 神经元切分输入空间的优势应用到输入信号的拟合上。本章对其进行优化，将方程(6-4)中的记忆/遗忘单元引入 IC 神经元中，单个神经元的输出变为

$$h_t = \mathbf{f}(\mathbf{w}\mathbf{x}_t + \mathbf{u}\mathbf{h}_{t-1} + q_1 e_t + \sigma(\mathbf{w}\mathbf{x}_t - \mathbf{w}'\mathbf{I}\mathbf{x}_t + q_2 e_t)) \quad (6-6)$$

在方程(6-6)中，IC 神经元在融入了历史状态 e_t 的信息。用此结构构建的递

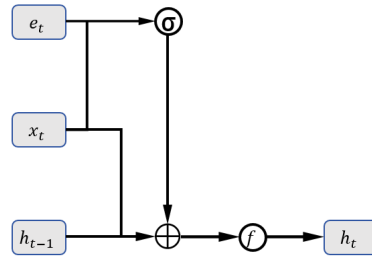


图 6-3 基于记忆/遗忘结构的 IC-RNN+ 结构。

归神经网络被称为 IC-RNN+, 它的结构如图6-3所示, 可表示为如下公式。

$$\begin{aligned}
 \mathbf{k}_t &= \sigma(\mathbf{w}_1 \mathbf{x}_t + \mathbf{u}_1 \mathbf{h}_{t-1}) + \mathbf{k}_{t-1} - \mathbf{I} \\
 \mathbf{e}_t &= Ig(\mathbf{e}_{t-1}, \mathbf{k}_t) \\
 \mathbf{e}_t &:= \mathbf{e}_t + \lambda(\mathbf{w}_2 \mathbf{x}_t + \mathbf{u}_2 \mathbf{h}_{t-1}) \\
 \mathbf{h}_t &= f(\mathbf{W} \mathbf{x}_t + \mathbf{U} \mathbf{h}_{t-1} + \mathbf{Q}_1 \mathbf{e}_t + \sigma(\mathbf{W} \mathbf{x}_t - \mathbf{W}' \mathbf{I} \mathbf{x}_t + \mathbf{Q}_2 \mathbf{e}_t)),
 \end{aligned} \tag{6-7}$$

上式中, \mathbf{e}_t 表示对一层内的所有神经元状态进行拼接组成的向量, \mathbf{k} 同理。由此, IC-RNN+ 完全构造完毕。它解决了基础 IC 递归神经网络难以捕获长期依赖关系的问题。

6.5 多层 IC 递归神经网络

方程(6-7)表示一种单层的计算结构, 在实际应用中, 可能需要模型具备多层拓展的能力以获得更强的模型容量。为此, 本章给出两种策略将 IC-RNN+ 扩展为多层。

第一种策略在多层中复用 \mathbf{e}_t 变量, 这种方法假设在当前时刻, 所有层的计算都是瞬间完成的, 不会对信息保持时间产生影响。其结构如图6-4(a)所示, 公式表示为

$$\mathbf{h}_t^l = f(\mathbf{W}^l \mathbf{h}_t^{l-1} + \mathbf{U}^l \mathbf{h}_{t-1}^l + \mathbf{Q}_1 \mathbf{e}_t + \sigma(\mathbf{W}^l \mathbf{h}_t^{l-1} - \mathbf{W}^{l'} \mathbf{I} \mathbf{h}_t^{l-1} + \mathbf{Q}_2 \mathbf{e}_t)), \tag{6-8}$$

其中 \mathbf{h}_t^0 就是 \mathbf{x}_t 。所有层共用一组 \mathbf{e}_t 以及相关参数, 降低了整体模型参数量和计算量, 且 \mathbf{e}_t 中包含的历史信息不会因为层数增加而削弱, 将捕获的长期依赖关

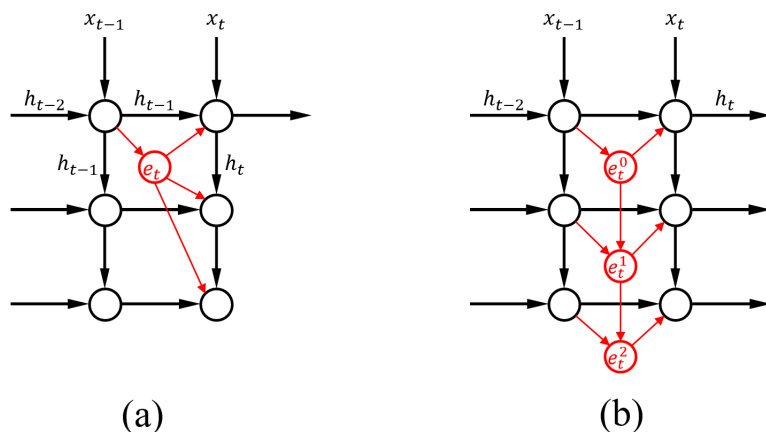


图 6-4 构建多层 IC-RNN+ 的两种策略。(a) 在多层中重复利用历史状态 e_t 。(b) 每一层重新计算历史状态 e_t 。

系应用在模型的全部层面。

第二种策略是在多层中应用不同的历史状态，上一层的历史状态会影响下一层的历史状态。其结构如图6-4(b)所示，公式表示为

$$\begin{aligned} \mathbf{e}_t^l &:= \mathbf{e}_t^l + \lambda(\mathbf{w}\mathbf{e}_t^{l-1} + \mathbf{u}\mathbf{h}_{t-1}) \\ \mathbf{h}_t^l &= f(\mathbf{W}^l\mathbf{h}_t^{l-1} + \mathbf{U}^l\mathbf{h}_{t-1}^l + \mathbf{Q}_1\mathbf{e}_t^l + \sigma(\mathbf{W}'^l\mathbf{h}_t^{l-1} - \mathbf{W}''^l\mathbf{h}_t^{l-1} + \mathbf{Q}_2\mathbf{e}_t^l)). \end{aligned} \quad (6-9)$$

第一层中的 \mathbf{e}_t 以及通过方程(6-7)得到。以后每一层中的历史状态都需要考虑它上一层的状态。这种策略每一层都要重新计算 \mathbf{e}_t ，会带来更多的计算量，但它在每一层中引入不同程度的历史状态，有可能捕获更多类型的长期依赖关系，提升模型的性能。一般来说，应用哪种策略取决于具体任务。在下一章的实验中，本章将展示实际任务中 IC-RNN+ 的表现。

6.6 实验结果与讨论

本章通过一系列实验来验证 IC-RNN+ 在序列任务中的有效性。此外，本章还进行消融实验，对 IC-RNN+ 中的记忆/遗忘模块进行全面分析。

6.6.1 数据集和基线配置

本章实验采用了两种序列数据评估 IC-RNN+ 的性能：文本类数据和周期性时间序列数据。

文本数据集：本章使用五种文本分类数据集，他们的介绍如下。

- **ACL-IMDB**，用于电影评论情感分析的数据集。ACL-IMDB 数据集包含来自互联网电影数据库 (IMDB) 的 50,000 条电影评论，其中 25,000 条用于训练，25,000 条用于测试。每条评论都被标记为正面或负面，用于二元情感分类任务。
- **Elec**，用于商品评论情感分析的数据集。Elec 数据集是亚马逊电子产品评论数据集的子集，包含约 1.8 百万条评论，评论分为正面和负面两类，用于情感分析和推荐系统研究。
- **AG-News**，用于新闻分类的数据集。AG-News 数据集包含从 2004 年到 2015 年收集的新闻文章，分为四类：世界、体育、商业和科技。每个类别有 30,000 条训练样本和 1,900 条测试样本。
- **DBpedia**，用于实体分类的数据集。DBpedia 数据集是从维基百科中提取的结构化信息，用于分类不同类别的实体。该数据集包含 14 类，每类有 40,000 个训练样本和 5,000 个测试样本。
- **RCV1**，用于新闻分类的数据集。RCV1 (Reuters Corpus Volume 1) 是路透社发布的一个大型新闻语料库，包含超过 80 万个新闻文档。该数据集用于多类文本分类任务。本章采用 JZ15a 的方法，基于二级主题进行多类主题分类，并构建其训练、测试集。

表 6-2 数据集信息统计。

数据集	训练集大小	测试集大小
ACL-IMDB	25,000	25,000
Elec	25,000	25,000
AG-News	120,000	7,600
DBpedia	560,000	70,000
RCV1	15,564	49,821

上述数据集的规模信息被总结到表6-2中。接下来本文对实验配置进行描述。本章实验遵循了前人的实验配置^[109]，使用 300 维的预训练向量来初始化嵌入层。ACL-IMDB、RCV1 数据集中的预训练向量使用 word2vec^[110]。其他数据集使用 fastText 预训练的嵌入^[111]。在模型层面，本章构建使用隐藏单元数为 512 的单层 IC-RNN+ 和单层双向 IC-RNN+。为了正则化，本章对词嵌入和 IC-RNN+ 的隐藏状态应用了 dropout (pdrop = 0.5)。本章还使用了词语 dropout 策略^[109]。具体来

说, 本章以概率 $p = 0.1$ 随机将一个词设置为“UNK”, 即未知词。在训练阶段, 本章使用 Adam 优化器进行优化, 初始学习率设置为 10^{-3} 。Adam 中的 $\beta_1 = 0$, $\beta_2 = 0.98$, $\epsilon = 10^{-8}$ 。训练过程采用指数学习率衰减方案。此外, 本章也对梯度裁剪, 将最大 L2 范数设置为 1, 本章实验通过整个序列进行时间反向传播, 即没有截断序列。损失函数使用交叉熵损失函数。

由于 IC-RNN+ 是一种只在模型层面创新的结构, 本章实验只将其与几种基础模型结构相比较, 包括 RNN, 基础 IC-RNN, LSTM, GRU, 自注意力机制。所有对比算法的训练方法与 IC-RNN+ 完全一样。

时间序列预测数据集: 时间序列数据不同于文本数据, 他们不需要将文本转换为抽象特征, 而是由直接采集到的数据得到, 本章使用了八个常用的时间序列预测数据, 他们每个都包含丰富的长期依赖关系, 他们的所表示的任务如下所示。

- **ETTM1, ETTM2:** 用于预测电力负荷, 每 15 分钟记录一次。
- **ETTh1, ETTh2:** 用于预测电力负荷, 每 15 分钟记录一次。
- **Electricity:** 用于长期预测电力负荷, 每小时记录一次。
- **Traffic:** 用于交通流量预测, 每小时记录一次。
- **Weather:** 用于天气预报, 每 10 分钟记录一次。
- **Exchange:** 用于汇率预测, 每天记录一次。

表6-3记录了所有数据集的详细信息, 包括维度、序列长度和数据集大小。针对数据集的预处理, 本章遵循之前的工作^[112]。在训练时, 本章使用 Adam 优化器进行优化, 初始学习率设置为 10^{-4} 。Adam 中的 $\beta_1 = 0$, $\beta_2 = 0.98$, $\epsilon = 10^{-8}$ 。Batch 的大小为 32, 训练轮数为 20 轮, 本章在训练中应用了早停策略, 即根据验证集损失的收敛情况选择终止训练时刻。损失函数采用均方误差 (MSE)。针对验证集和测试集的评估是用 MSE 和平均绝对误差 (MAE)。

表 6-3 时间序列数据集的基本信息。

数据集	序列单元维度	序列长度	数据集大小 (训练集, 验证集, 测试集)
ETTM1,ETTM2	7	{96, 192, 336, 720}	(34,465, 11,521, 11,521)
ETTh1,ETTh2	7	{96, 192, 336, 720}	(8,545, 2,881, 2,881)
Electricity	321	{96, 192, 336, 720}	(18,317, 2,633, 5,261)
Traffic	862	{96, 192, 336, 720}	(12,185, 1,757, 3,509)
Weather	21	{96, 192, 336, 720}	(36,792, 5,271, 10,540)
Exchange	8	{96, 192, 336, 720}	(5,120, 665, 1,422)

在时间序列预测的实验中，本章除了将 IC-RNN+ 与基础计算结构 (RNN, 基础 IC-RNN, LSTM, GRU, Self-attention) 进行比较，还将其应用到先进的时间序列框架中，与一些较新工作比较，进一步展示 IC-RNN+ 的泛用能力，此外，在这部分的实验中，本章也对构建多层 IC-RNN+ 的两种策略进行评估。

6.6.2 文本分类数据集实验结果

表 6-4 文本分类数据集中的对比结果。

Model	ACL-IMDB	Elec	AG-News	DBpedia	RCV1
RNN	78.42	77.48	80.32	78.22	74.26
IC-RNN	80.21	79.54	81.77	78.37	76.52
LSTM	84.22	83.63	87.25	84.54	82.34
GRU	85.13	85.24	87.21	83.65	82.48
Bi-LSTM	86.42	84.32	86.48	85.14	84.56
Bi-GRU	86.28	86.10	87.54	93.20	83.38
Transformer	88.57	88.42	90.28	91.46	83.56
IC-RNN+	89.45	89.17	89.74	88.38	83.25
Bi-IC-RNN+	89.74	88.06	90.45	89.28	84.34

本章对比了几种不同的模型在 ACL-IMDB、Elec、AG-News、DBpedia 和 RCV1 数据集上的性能，Bi-IC-RNN+ 是 IC-RNN+ 的双向形式，该实验中所有模型均为一层。实验结果如表6-4所示。

IC-RNN 模型是本章优化的基础方法，其在所有数据集上的性能均优于传统的 RNN 模型。具体来说，IC-RNN 在 ACL-IMDB、Elec、AG-News、DBpedia 和 RCV1 数据集上的准确率分别提升了 1.79%、2.06%、1.45%、0.15%、2.26%。这些改进表明 IC-RNN 在还是能发挥 IC 神经元的优势，然而，IC-RNN 算法难以有效捕获长期依赖关系，这对它的性能产生了较大限制。

LSTM 和 GRU 模型在所有数据集上均表现出色，尤其是在 AG-News 和 RCV1 数据集上，准确率分别达到了 87.25% 和 82.48%。双向 LSTM (Bi-LSTM) 和双向 GRU (Bi-GRU) 在大多数数据集上的表现优于单向模型，这表明双向结构有助于捕捉序列数据中的前后文信息。Transformer 模型在多个数据集上表现最佳，特别是在 DBpedia 数据集上，准确率分别达到了 91.46%。这几种模型都对长期依赖关系进行有效学习，表现出优于传统 RNN 的性能。

本章提出的 IC-RNN+ 和 Bi-IC-RNN+ 模型在所有数据集上均取得了显著的性能提升。IC-RNN+ 在 Elec 数据集上取得了最高的 89.17% 的准确率，比现有

的最优模型（Transformer）的 88.42% 高出 0.75%。Bi-IC-RNN+ 在 ACL-IMDB、AG-News 和 RCV1 数据集上分别达到了 89.74%、90.45% 和 84.34% 的准确率，相较于其他模型有显著提升。

IC-RNN+ 和 Bi-IC-RNN+ 模型在处理长期依赖关系上表现出色，这是 IC-RNN 模型无法有效实现的。特别是在处理长文本数据集时，IC-RNN+ 和 Bi-IC-RNN+ 模型展示了极强的泛化能力和稳定性，这使其在文本分类任务中具有显著优势。

总体而言，本章提出的改进模型在各个数据集上的表现均优于传统模型，表现了所提方法的有效性和优越性。

6.6.3 时间序列数据集实验结果

本章实验对验证 IC-RNN+ 在时间序列任务上的性能。过去序列长度设置为 96，预测长度为 (96, 192, 336, 720)，结果取平均。

表 6-5 时间序列数据上各基础模型的预测结果。

Models	RNN		LSTM		GRU		Transformer		IC-RNN		IC-RNN+	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	1.542	0.985	1.324	0.858	1.378	0.869	0.978	0.745	1.548	1.066	0.986	0.762
ETTm2	2.896	1.362	2.395	1.177	2.264	1.098	1.352	0.863	2.764	1.288	1.064	0.804
ETTh1	1.548	0.994	1.198	0.821	1.326	0.830	0.842	0.694	1.438	1.045	0.960	0.776
ETTh2	4.872	1.644	3.095	1.352	3.432	1.487	2.354	1.301	4.298	1.657	2.265	1.241
Electricity	0.641	0.603	0.559	0.549	0.543	0.540	0.374	0.424	0.587	0.564	0.382	0.435
Traffic	1.462	0.853	1.011	0.541	1.032	0.544	0.904	0.528	1.301	0.798	0.882	0.516
Weather	0.806	0.612	0.444	0.454	0.694	0.586	0.346	0.387	0.617	0.528	0.566	0.502
Exchange	2.853	1.472	2.105	1.221	2.086	1.230	1.350	0.928	2.842	1.485	1.261	0.869

实验结果如表6-5所示。首先可以观察到 IC-RNN 算法在一些数据集上的表现较差，特别是在 ETTm2、ETTh2 和 Exchange 数据集上，其均方误差（MSE）和平均绝对误差（MAE）较高。这表明 IC-RNN 算法在处理这些数据集时可能无法有效学习长周期的特性，导致预测性能较差。而本章提出的 IC-RNN+ 模型在大多数数据集上表现出色，相比于 IC-RNN，其在所有数据集上的 MSE 和 MAE 都有所降低。特别是在 ETTm2 和 ETTh2 数据集上，IC-RNN+ 的性能显著优于 IC-RNN，验证了本章引入记忆/遗忘模块的有效性。

与传统的递归神经网络模型（如 LSTM 和 GRU）相比，IC-RNN+ 在大多数数据集上表现更好。这三类模型都使用记忆/遗忘机制来捕获长期依赖。不同的是 IC-RNN+ 是通过一个历史状态和历史状态保持时间来控制记忆和遗忘，LSTM

和 GRU 都是通过简单门结构来控制记忆和遗忘。实验结果表明表明 IC-RNN+ 中的记忆遗忘机制更有可能学习到长周期中的序列性质，从而改善了预测性能。

与 Transformer 模型相比，IC-RNN+ 在大多数数据集上的性能接近或略有优势。特别是在 ETTm1、ETTh1 和 Electricity 数据集上，IC-RNN+ 的性能与 Transformer 相当。然而，在 Traffic 和 Exchange 数据集上，Transformer 模型表现更好。

综上所述，本章提出的 IC-RNN+ 模型在多个数据集上取得了较好的性能。然而，在某些数据集上，仍有改进的空间，需要进一步优化模型以提高预测精度。

除了与基础模型进行对比外，本章也评估 IC-RNN+ 与一些先进的时间序列方法的对比，对比方法包括 LSSL^[113]、Pyraformer^[114]、Informer^[115]、Reformer^[116]。在本实验中，本章基于两种不同的策略构建两层 IC-RNN+ 模型，隐藏层神经元数量均为 128，实验基于之前工作的配置^[112]。

表 6-6 时间序列数据上多种时序方法的预测结果。

Models	LSSL		Pyraformer		Informer		Reformer		IC-RNN+ St1		IC-RNN+ St2	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTM1	0.533	0.532	0.691	0.607	0.961	0.734	0.799	0.671	0.542	0.502	0.562	0.531
ETTM2	0.735	0.598	1.410	0.810	1.535	0.900	1.479	0.915	0.788	0.608	0.728	0.604
ETTh1	0.777	0.664	1.040	0.795	1.072	0.837	1.029	0.805	0.720	0.653	0.654	0.608
ETTh2	4.872	1.644	4.431	1.729	2.686	1.494	6.736	2.191	2.264	1.452	2.082	1.430
Electricity	0.313	0.401	0.311	0.397	0.272	0.370	0.338	0.422	0.285	0.376	0.283	0.351
Weather	0.271	0.334	0.634	0.548	0.696	0.602	0.803	0.656	0.617	0.528	0.374	0.485

实验结果如表6-6所示，St1 表示共享历史状态的策略，St2 表示不共享的策略。IC-RNN+ 算法在多个数据集上表现出色，特别是在 ETTm1、ETTM2、ETTh1 和 ETTh2 数据集上，IC-RNN+ St2 在 MSE 和 MAE 指标上都取得了显著的优势。

与基于 Transformer 的模型（如 LSSL、Pyraformer、Informer 和 Reformer）相比，IC-RNN+ St1 和 St2 在多个数据集上取得了更低 MSE 和 MAE 值，特别是在 ETTm2 和 ETTh2 数据集上，显示了其在处理长时间序列预测任务上的优势。虽然 LSSL 在 ETTm1 和 Weather 数据集上取得了最佳的 MSE，但 IC-RNN+ St1 和 St2 在大多数其他数据集上表现更为稳定。

IC-RNN+ St1 是一种更轻量的结构，相比于 IC-RNN+ St2，其计算复杂度更低，但在大多数数据集上的预测性能略低于 St2。然而，IC-RNN+ St1 在一些数据集上（如 ETTm1 和 ETTh1）仍然表现出色，表明了其在保持较低计算复杂度的同时依然具有较强的预测能力。

总体而言，本章提出的 IC-RNN+ 及其变体在多个时间序列预测任务中表现优异，特别是在捕获长期依赖关系方面。IC-RNN+ 结合了 RNN 类模型的优势，并通过改进的记忆遗忘机制显著提升了预测精度。相比于基于 Transformer 的模型，IC-RNN+ 已在多个数据集上实现了更低的预测误差，显示出其在长时间序列预测任务中的潜力，在未来具有重要应用价值。

6.6.4 消融实验

在对比实验中，IC-RNN+ 已经表现出超越 IC-RNN 以及其它一些基础模型的性能，可以用来构建新型的递归神经网络模型，本章通过消融实验进一步验证 IC-RNN+ 中各组成部分的性能。

记忆/遗忘模块。首先验证记忆遗忘模块，本章构建了几种变体来验证细节模块:1. 去除遗忘模块，历史状态 \mathbf{e}_t 只负责记忆新的信号，不负责遗忘；2. 遗忘模块依赖于历史状态保持时间参数 k ，去出该参数，设置一定长的保持时间，到达时间后，自动忘记历史状态 \mathbf{e}_t ；3. 在计算历史状态 \mathbf{e}_t 时，只考虑当前时刻输入 \mathbf{x}_t ，不考虑隐藏变量 \mathbf{h}_t 的影响。

表 6-7 记忆/遗忘模块的消融实验。

Model	ACL-IMDB	Elec	AG-News	DBpedia	RCV1
RNN	78.42	77.48	80.32	78.22	74.26
IC-RNN	80.21	79.54	81.77	78.37	76.52
IC-RNN+	89.45	89.17	89.74	88.38	83.25
IC-RNN+ w/o Ig	80.13	78.24	78.53	74.52	76.60
IC-RNN+ w/o k	83.97	84.20	83.31	80.28	77.78
IC-RNN+ w/o h_{t-1}	83.97	84.20	83.31	80.28	77.78

实验结果如表6-7所示。首先，IC-RNN+ 相比于基础的 RNN 和 IC-RNN 模型，在所有数据集上的表现均有显著提升。例如，在 ACL-IMDB 数据集上，IC-RNN+ 的准确率达到 89.45%，相比于 IC-RNN 的 80.21% 有显著提升。这表明引入的记忆遗忘机制对模型性能有重要影响。其次，消融实验结果显示，去除遗忘模块的 IC-RNN+ (IC-RNN+ w/o Ig) 在所有数据集上的表现均显著下降。例如，在 Elec 数据集上，准确率从 89.17% 降至 78.24%。这说明遗忘模块在处理长时间依赖关系时起到了关键作用。此外，当遗忘模块不依赖于历史状态保持时间参数 k ，而是设置固定的保持时间时 (IC-RNN+ w/o k)，模型性能有所下降。例如，在 AG-News 数据集上，准确率从 89.74% 降至 83.31%。这表明动态调整的遗忘时间

比固定的保持时间更有效。最后，若在计算历史状态 \mathbf{e}_t 时不考虑隐藏变量 \mathbf{h}_{t-1} 的影响 (IC-RNN+ w/o \mathbf{h}_{t-1})，模型性能同样有所下降。例如，在 DBpedia 数据集上，准确率从 88.38% 降至 80.28%。这表明在历史状态计算中引入隐藏状态信息对于捕获复杂时间依赖关系至关重要。

记忆/遗忘模块与 IC 神经元的结合。评估完记忆遗忘模块后，本章对记忆/遗忘模块与 IC 神经元的结合方式进行评估。构造了以下几种变体：1. 去除 $\mathbf{Q}_1\mathbf{e}_t$ 项。2. 去除超平面中的历史状态附加项 $\mathbf{Q}_2\mathbf{e}_t$ 。

表 6-8 记忆/遗忘模块的消融实验。

Model	ACL-IMDB	Elec	AG-News	DBpedia	RCV1
IC-RNN+	89.45	89.17	89.74	88.38	83.25
IC-RNN+ w/o $\mathbf{Q}_1\mathbf{e}_t$	83.11	80.48	83.65	84.23	77.71
IC-RNN+ w/o $\mathbf{Q}_2\mathbf{e}_t$	84.87	81.217	85.88	85.68	80.69

实验结果如表6-8所示。可以观察到几个结论：1. $\mathbf{Q}_1\mathbf{e}_t$ 项的作用主要在于计算隐藏状态时能够引入历史信息，建立一些长期依赖关系，如果去掉这一项，IC-RNN+ 的性能将会显著下降；2. $\mathbf{Q}_2\mathbf{e}_t$ 项的作用主要在于在超平面计算时引入历史信息，让 IC 神经元对历史关键信息更加敏感，相比于第一种变体，去掉 $\mathbf{Q}_2\mathbf{e}_t$ 也会使性能下降，但不如第一种消融变体下降明显。 $\mathbf{Q}_1\mathbf{e}_t$ 项与 $\mathbf{W}\mathbf{x}_t$ 项以及 $\mathbf{U}\mathbf{h}_{t-1}$ 项并列，他们具有相同层次的重要度，共同计算时序特征。

综上所述，实验结果验证了 IC-RNN+ 模型中个模块的有效性。IC-RNN+ 通过记忆/遗忘机制有效解决了基础 IC-RNN 类模型在处理长时间序列数据时的不足。未来的工作可以进一步优化该机制，以期在更多实际应用中取得更优表现。

6.7 本章小结

本章致力于研究和探索 IC 神经元与递归模型的深层应用，旨在将 IC 神经元应用于广泛的自然语言任务。IC 神经元与递归神经网络的简单结合难以捕获准确的长期依赖关系，因此在很多自然语言任务表现较差。本章首先设计了一种不同于 LSTM 或 GRU 的记忆/遗忘模块，它具备提取更长距离依赖关系的能力。然后将其应与 IC 神经元结合构建 IC-RNN+。本章也提出了两种策略构建多层的 IC-RNN+ 模型。实验结果表明，记忆遗忘模块在 IC-RNN+ 模型中扮演着关键的角色，并且对模型的性能具有显著的提升作用。本章也将 IC-RNN+ 模型与其他

基线模型进行了比较分析，结果显示 IC-RNN+ 在各个数据集上都取得了明显的性能优势。综上所述，本章的研究结果充分展示了 IC-RNN+ 模型在序列任务中的有效性和优越性，为处理长序列数据提供了一种可行且有效的解决方案。

第七章 总结与展望

7.1 全文总结

本文主要研究深度学习时代中的新型神经元模型。随着神经网络在人类社会中的广泛应用，针对神经网络模型的研究也愈来愈多。很多精巧的网络架构相继被提出，包括卷积神经网络，递归神经网络，图神经网络等。然而，神经网络的基础单元，即神经元模型，依然延续着上个世纪 40 年代提出的 MP 神经元。在过去的八十年，也有其它新型神经元模型诞生，但大部分神经元模型受限于计算设备和学习算法等问题，无法像 MP 神经元集成为深度神经网络。然而，MP 神经元的结构简单，他所建模的神经网络还有很大优化空间。如果新型神经元的研究具有取代 MP 神经元的能力，他将有可能应用到广泛的机器学习任务和神经模型中。综上所述，新型神经元的研究具有重要研究价值和应用价值。

本文集中于神经元层面的结构研究，通过两种思路设计高效神经网络结构：
1. 设计普适的神经元结构广泛提升神经网络的性能；
2. 针对不同网络架构或者下游任务的特性应用专精的神经元或基础计算结构。

1. 在第三章中，针对现有 MP 神经元拟合能力弱等神经元层面的问题，本文提出了一种基于物理弹性碰撞的新型神经元模型：IC 神经元。不同于传统生物启发的神经元结构。物理启发的 IC 神经元具有更强的物理合理性和数理解释性。本文通过 XOR 问题验证了单个 IC 神经元可以解决线性不可分问题。相比于 MP 神经元，IC 神经元在增强非线性拟合能力的同时，保持了模型的易集成，轻量性特点，使其可以应用在诸多神经网络架构中。本文通过大量实验验证了，IC 神经元能在各类任务和模型中取代现在最流行的 MP 神经元，以 IC 神经元为核心的神经网络能在计算消耗相同的情况下取得更好的泛化性能。

2. 在第四章，本文将 IC 神经元适配到典型的卷积神经网络中。针对 IC 神经元的表达能力会受高维数据约束的缺点，本文提出了一种基于动态卷积的新型 IC 卷积结构 IC-CNN+。在引入动态化方法后，IC 神经元中的线性超平面转换为

可学习的超曲面。它可以根据实际数据分布调节自身形状，使 IC 神经元能够拟合高维空间中的复杂特征。另一方面，本文也对 IC-CNN+ 中的动态卷积方法进行了研究，提出了一种新的动态卷积方法 LADConv。它不仅解决了局部特征丢失的问题，还能与 IC-CNN+ 完美适配。本文通过实验证明，结合 LADConv 的 IC-CNN+ 结构能够显著提升深度卷积神经网络的性能。

3. 在第五章，本文将 IC 神经元适配到图神经网络中以提取 3D 数据的特征，并应用到下游任务。针对 IC 神经元不满足 3D 任务等变性约束的问题，本文将等变群表示技术与 IC 神经元结合，提出了一种近似等变的结构 IC-GNN+。IC-GNN+ 用 IC 神经元代替一些严格等变操作，并通过旋转到局部坐标系来降低 IC 神经元学习等变性的难度。同时，IC 神经元的引入可以突破传统等变模型表达能力的理论上限。本文通过实验证明，IC-GNN+ 在点云、分子数据中都可以学习到有效的近似等变特征，在下游任务中表现出优秀的泛化性能。

4. 在第六章，本文将 IC 神经元适配到递归神经网络中以处理自然语言任务。针对 IC 神经元无法有效学习长期依赖关系的问题，本文将记忆/遗忘机制与 IC 神经元结合，提出了一种记忆性递归神经网络结构 IC-RNN+。本文首先分析了 LSTM 和 GRU 模型中记忆/遗忘机制的优点和缺点，然后提出了一种应用于神经元层面的记忆/遗忘结构，再将其与 IC 神经元结合。本文的实验展示了 IC-RNN+ 在文本分类和时间序列预测中都表现出超越 LSTM 和 GRU 模型的性能，并在与当下主流的 Transformer 模型比较中也不落下风。

7.2 研究展望

本文在神经元模型领域做了一定的探索，首先提出了一种通用的神经元模型，再根据具体任务或网络架构设计了专精于 2D 数据，3D 数据和序列数据的三种模型。三项工作均取得了一定成效。但同时也引入了一系列问题可能成为未来的研究方向：

1. 虽然 IC 神经网络能够增加非线性模式，但这种提升是基于相同的一组参数，这有可能加剧 IC 神经网络的过拟合风险。事实上，本文也在部分实验中观察到 IC 神经元表现出强于 MP 神经元的过拟合现象。因此，后续研究可以继续沿着 IC 神经元的结构优化的方式走下去，设计更灵活的非线性结构，减小过拟

合风险。

2. 在 IC 卷积结构中，动态卷积可以有效提升 IC 神经元的表达能力，但也带来一个问题：动态卷积需要根据当前输入来生成动态卷积核，导致并行化运行效率低。这个问题在训练时表现的更为明显。如何提升动态卷积并行化效率是一个值得研究的课题。

3. 在 IC 图神经网络研究中，近似等变性需要通过学习得到，这要求模型能得到充足训练。然而，一些 3D 数据的获取成本较高。在这种情况下，数据增强技术可以用来增强近似等变性。如何设计针对近似等变性的有效数据增强方法也是未来的一个重要研究方向。

4. 在 IC 递归神经网络的研究中，Transformer 是一种不基于记忆/遗忘机制的模型，它可以捕获任意长的依赖关系。未来的研究可以考虑如何将 IC 神经元的思想应用到 Transformer 类模型中，以进一步提升序列任务上的性能。

参考文献

- [1] MCCARTHY J. Artificial intelligence, logic, and formalising common sense [J]. Machine Learning and the City: Applications in Architecture and Urban Design, 2022: 69-90.
- [2] 陆汝钤. 人工智能[J]. 科学出版, 1996.
- [3] BOSER B E, GUYON I M, VAPNIK V N. A training algorithm for optimal margin classifiers[C]//Proceedings of the fifth annual workshop on Computational learning theory. 1992: 144-152.
- [4] QUINLAN J R. Induction of decision trees[J]. Machine learning, 1986, 1: 81-106.
- [5] ROSENBLATT F. The perceptron: a probabilistic model for information storage and organization in the brain.[J]. Psychological review, 1958, 65(6): 386.
- [6] DEVLIN J, CHANG M W, LEE K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.
- [7] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. (CVPR). 2016: 770-778.
- [8] NASSIF A B, SHAHIN I, ATTILII I, et al. Speech recognition using deep neural networks: A systematic review[J]. IEEE access, 2019, 7: 19143-19165.
- [9] RUDER S. An overview of gradient descent optimization algorithms[J]. arXiv preprint arXiv:1609.04747, 2016.
- [10] HU J, SHEN L, SUN G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. (CVPR).

- 2018: 7132-7141.
- [11] XIE S, GIRSHICK R, DOLLÁR P, et al. Aggregated residual transformations for deep neural networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. (CVPR). 2017: 1492-1500.
 - [12] YANG B, BENDER G, LE Q V, et al. Condconv: Conditionally parameterized convolutions for efficient inference[J]. Advances in neural information processing systems. (NeurIPS), 2019, 32.
 - [13] CHEN Y, DAI X, LIU M, et al. Dynamic convolution: Attention over convolution kernels[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. (CVPR). 2020: 11030-11039.
 - [14] MAASS W. Networks of spiking neurons: the third generation of neural network models[J]. Neural networks, 1997, 10(9): 1659-1671.
 - [15] MCCULLOCH W S, PITTS W. A logical calculus of the ideas immanent in nervous activity[J]. The bulletin of mathematical biophysics, 1943, 5: 115-133.
 - [16] HODGKIN A L, HUXLEY A F. A quantitative description of membrane current and its application to conduction and excitation in nerve[J]. The Journal of physiology, 1952, 117(4): 500.
 - [17] LAING C R, CHOW C C. A spiking neuron model for binocular rivalry[J]. Journal of computational neuroscience, 2002, 12: 39-53.
 - [18] HOPFIELD J J. Neural networks and physical systems with emergent collective computational abilities.[J]. Proceedings of the national academy of sciences, 1982, 79(8): 2554-2558.
 - [19] SALAKHUTDINOV R, HINTON G. Deep boltzmann machines[C]//Artificial intelligence and statistics. 2009: 448-455.
 - [20] SIMONYAN K, ZISSERMAN A. Very Deep Convolutional Networks for Large-Scale Image Recognition[C]//International Conference on Learning Representations. (ICLR). 2015.

- [21] GILMER J, SCHOENHOLZ S S, RILEY P F, et al. Neural message passing for quantum chemistry[C]//International Conference on Machine Learning. (ICML). 2017: 1263-1272.
- [22] STAUDEMAYER R C, MORRIS E R. Understanding LSTM—a tutorial into long short-term memory recurrent neural networks[J]. arXiv preprint arXiv:1909.09586, 2019.
- [23] MASSE N Y, YANG G R, SONG H F, et al. Circuit mechanisms for the maintenance and manipulation of information in working memory[J]. *Nature neuroscience*, 2019, 22(7): 1159-1167.
- [24] LEE H, KIM C, LEE S, et al. An accurate and fair evaluation methodology for SNN-based inferencing with full-stack hardware design space explorations[J]. *Neurocomputing*, 2021, 455: 125-138.
- [25] ROY K, JAISWAL A, PANDA P. Towards spike-based machine intelligence with neuromorphic computing[J]. *Nature*, 2019, 575(7784): 607-617.
- [26] ZHANG S Q, ZHOU Z H. Flexible transmitter network[J]. *Neural Computation*, 2021, 33(11): 2951-2970.
- [27] BAO B, YANG Q, ZHU L, et al. Chaotic bursting dynamics and coexisting multistable firing patterns in 3D autonomous Morris–Lecar model and microcontroller-based validations[J]. *International Journal of Bifurcation and Chaos*, 2019, 29(10): 1950134.
- [28] IZHIKEVICH E M. Simple model of spiking neurons[J]. *IEEE Transactions on neural networks*, 2003, 14(6): 1569-1572.
- [29] LI Y, CHEN Y, DAI X, et al. Revisiting Dynamic Convolution via Matrix Decomposition[C]//International Conference on Learning Representations. (ICLR). 2021.
- [30] HE S, JIANG C, DONG D, et al. Sd-conv: Towards the parameter-efficiency of dynamic convolution[C]//Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. (WACV). 2023: 6454-6463.

- [31] LI C, ZHOU A, YAO A. Omni-Dimensional Dynamic Convolution[C]// International Conference on Learning Representations (ICLR). 2022.
- [32] RUSSAKOVSKY O, DENG J, SU H, et al. Imagenet large scale visual recognition challenge[J]. International journal of computer vision, 2015, 115(3): 211-252.
- [33] LIN T Y, MAIRE M, BELONGIE S, et al. Microsoft coco: Common objects in context[C]//Proceedings of the European conference on computer vision. (ECCV). 2014: 740-755.
- [34] CHEN Y, FERNANDO B, BILEN H, et al. 3d equivariant graph implicit functions[C]//Proceedings of the IEEE international european conference on computer vision. (ECCV). 2022: 485-502.
- [35] KONDOR R, LIN Z, TRIVEDI S. Clebsch–gordan nets: a fully fourier space spherical convolutional neural network[J]. Advances in Neural Information Processing Systems. (NeurIPS), 2018, 31.
- [36] COHEN T S, GEIGER M, KÖHLER J, et al. Spherical cnns[J]. arXiv preprint arXiv:1801.10130, 2018.
- [37] GASTEIGER J, BECKER F, GÜNNEMANN S. Gemnet: Universal directional graph neural networks for molecules[J]. Advances in Neural Information Processing Systems. (NeurIPS), 2021, 34: 6790-6802.
- [38] GASTEIGER J, SHUAIBI M, SRIRAM A, et al. GemNet-OC: Developing Graph Neural Networks for Large and Diverse Molecular Simulation Datasets [J]. Transactions on Machine Learning Research, 2022.
- [39] ZITNICK C L, DAS A, KOLLURU A, et al. Spherical Channels for Modeling Atomic Interactions[J]. Advances in neural information processing systems. (NeurIPS), 2022.
- [40] CHAMI I, YING Z, RÉ C, et al. Hyperbolic graph convolutional neural networks[J]. Advances in Neural Information Processing Systems. (NeurIPS), 2019, 32.

- [41] BRANDSTETTER J, HESSELINK R, van der POL E, et al. Geometric and physical quantities improve e (3) equivariant message passing[C]// International Conference on Learning Representations. (ICLR). 2021.
- [42] THOMAS N, SMIDT T, KEARNES S, et al. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds[J]. Advances in neural information processing systems. (NeurIPS), 2018.
- [43] LIAO Y L, SMIDT T. Equiformer: Equivariant Graph Attention Transformer for 3D Atomistic Graphs[C]// . 2023.
- [44] GRIFFITHS D J, SCHROETER D F. Introduction to quantum mechanics[M]. Cambridge university press, 2018.
- [45] WEILER M, GEIGER M, WELLING M, et al. 3d steerable cnns: Learning rotationally equivariant features in volumetric data[J]. Advances in Neural Information Processing Systems. (NeurIPS), 2018, 31.
- [46] FUCHS F, WORRALL D, FISCHER V, et al. Se (3)-transformers: 3d rotation translation equivariant attention networks[J]. Advances in Neural Information Processing Systems. (NeurIPS), 2020, 33: 1970-1981.
- [47] GEIGER M, SMIDT T, M. A, et al. Euclidean neural networks: e3nn[CP]. 2022.
- [48] BARRON A R. Approximation and estimation bounds for artificial neural networks[J]. Machine learning, 1994, 14: 115-133.
- [49] CYBENKO G. Approximation by superpositions of a sigmoidal function[J]. Mathematics of control, signals and systems, 1989, 2(4): 303-314.
- [50] FUNAHASHI K I. On the approximate realization of continuous mappings by neural networks[J]. Neural networks, 1989, 2(3): 183-192.
- [51] SUN S, CHEN W, WANG L, et al. On the depth of deep neural networks: A theoretical view[C]// Proceedings of the AAAI Conference on Artificial Intelligence: vol. 30: 1. 2016.
- [52] POGGIO T, BANBURSKI A, LIAO Q. Theoretical issues in deep networks

- [J]. Proceedings of the National Academy of Sciences, 2020, 117(48): 30039-30045.
- [53] SOUDRY D, HOFFER E, NACSON M S, et al. The implicit bias of gradient descent on separable data[J]. Journal of Machine Learning Research, 2018, 19(70): 1-57.
- [54] LYU K, LI J. Gradient descent maximizes the margin of homogeneous neural networks[J]. arXiv preprint arXiv:1906.05890, 2019.
- [55] RUMELHART D E, HINTON G E, WILLIAMS R J. Learning representations by back-propagating errors[J]. nature, 1986, 323(6088): 533-536.
- [56] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. Advances in Neural Information Processing Systems. (NeurIPS), 2017, 30.
- [57] ZHANG C, BENGIO S, HARDT M, et al. Understanding deep learning (still) requires rethinking generalization[J]. Communications of the ACM, 2021, 64(3): 107-115.
- [58] STEARNS S D. Of aldaptive signal processing[M]. 1985.
- [59] HE K, ZHANG X, REN S, et al. Identity mappings in deep residual networks [C]//Proceedings of the IEEE international european conference on computer vision. (ECCV). 2016: 630-645.
- [60] ZEILER M D. Adadelta: an adaptive learning rate method[J]. arXiv preprint arXiv:1212.5701, 2012.
- [61] IANDOLA F, MOSKEWICZ M, KARAYEV S, et al. Densenet: Implementing efficient convnet descriptor pyramids[J]. arXiv preprint arXiv:1404.1869, 2014.
- [62] HOWARD A G, ZHU M, CHEN B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. (CVPR). 2017.
- [63] LECUN Y, BOSER B, DENKER J S, et al. Backpropagation applied to handwritten zip code recognition[J]. Neural computation, 1989, 1(4): 541-551.

- [64] EVERINGHAM M, GOOL L V, WILLIAMS C K I, et al. The Pascal Visual Object Classes (VOC) Challenge[J]. *International journal of computer vision*, 2010, 88(2): 303-338.
- [65] REN S Q, HE K M, GIRSHICK R B, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[C]// *Advances in Neural Information Processing Systems*. (NeurIPS). 2015: 91-99.
- [66] LIN T Y, GOYAL P, GIRSHICK R, et al. Focal loss for dense object detection [C]// *Proceedings of the IEEE international conference on computer vision*. (ICCV). 2017: 2980-2988.
- [67] CHEN K, WANG J Q, PANG J M, et al. Mmdetection: Open mmlab detection toolbox and benchmark[J]. *arXiv:1906.07155*, 2019.
- [68] SHIN Y, GHOSH J. The pi-sigma network: An efficient higher-order neural network for pattern classification and function approximation[C]// *International Joint Conference on Neural Networks*. (IJCNN): vol. 1. 1991: 13-18.
- [69] HUSSAIN A J, LIATSIS P, TAWFIK H, et al. Physical time series prediction using recurrent pi-sigma neural networks[J]. *International Journal of Artificial Intelligence and Soft Computing*, 2008, 1(1): 130-145.
- [70] LI C K. A sigma-pi-sigma neural network (SPSNN)[J]. *Neural Processing Letters*, 2003, 17: 1-19.
- [71] NAYAK J, NAIK B, BEHERA H. A hybrid PSO-GA based Pi sigma neural network (PSNN) with standard back propagation gradient descent learning for classification[C]// *2014 international conference on control, instrumentation, communication and computational technologies*. (iccicct). 2014: 878-885.
- [72] CLEVERT D A, UNTERTHINER T, HOCHREITER S. Fast and accurate deep network learning by exponential linear units (elus)[J]. *arXiv preprint arXiv:1511.07289*, 2015.
- [73] LIU Z, MAO H, WU C Y, et al. A convnet for the 2020s[C]// *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

- (CVPR). 2022: 11976-11986.
- [74] KIRKPATRICK J, PASCANU R, RABINOWITZ N, et al. Overcoming catastrophic forgetting in neural networks[J]. Proceedings of the National Academy of Sciences, 2017, 114(13): 3521-3526.
- [75] HINTON G E, VINYALS O, DEAN J. Distilling the Knowledge in a Neural Network[J]. Advances in Neural Information Processing Systems. (NeurIPS), 2015.
- [76] WOO S, PARK J, LEE J Y, et al. Cbam: Convolutional block attention module [C]//Proceedings of the European conference on computer vision. (ECCV). 2018: 3-19.
- [77] WANG Q, WU B, ZHU P, et al. ECA-Net: Efficient channel attention for deep convolutional neural networks[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. (CVPR). 2020: 11534-11542.
- [78] LIN X, MA L, LIU W, et al. Context-gated convolution[C]//Proceedings of the European conference on computer vision. (ECCV). 2020: 701-718.
- [79] QUADER N, BHUIYAN M M I, LU J, et al. Weight excitation: Built-in attention mechanisms in convolutional neural networks[C]//Proceedings of the European conference on computer vision. (ECCV). 2020: 87-103.
- [80] TAN M, LE Q. Efficientnet: Rethinking model scaling for convolutional neural networks[C]//International Conference on Machine Learning. (ICML). 2019: 6105-6114.
- [81] SELVARAJU R R, COGSWELL M, DAS A, et al. Grad-cam: Visual explanations from deep networks via gradient-based localization[C]//Proceedings of the IEEE international conference on computer vision. (ICCV). 2017: 618-626.
- [82] QIN Z, ZHANG P, WU F, et al. Fcanet: Frequency channel attention networks [C]//Proceedings of the IEEE/CVF international conference on computer vision. (ICCV). 2021: 783-792.
- [83] SCHÜTT K T, SAUCEDA H E, KINDERMANS P J, et al. Schnet—a deep

- learning architecture for molecules and materials[J]. *The Journal of Chemical Physics*, 2018, 148(24): 241722.
- [84] GASTEIGER J, GROSS J, GÜNNEMANN S. Directional message passing for molecular graphs[C]//International Conference on Learning Representations. (ICLR). 2020.
- [85] DYM N, MARON H. On the Universality of Rotation Equivariant Point Cloud Networks[C]//International Conference on Learning Representations. (ICLR). 2021.
- [86] JOSHI C K, BODNAR C, MATHIS S V, et al. On the expressive power of geometric graph neural networks[J]. *arXiv preprint arXiv:2301.09308*, 2023.
- [87] BATZNER S, MUSAEILIAN A, SUN L, et al. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials[J]. *Nature communications*, 2022, 13(1): 1-11.
- [88] PASSARO S, ZITNICK C L. Reducing SO(3) Convolutions to SO(2) for Efficient Equivariant GNNs[C]//Proceedings of Machine Learning Research. (PMLR): vol. 202. 2023: 27420-27438.
- [89] KOFINAS M, NAGARAJA N, GAVVES E. Roto-translated local coordinate frames for interacting dynamical systems[J]. *Advances in Neural Information Processing Systems. (NeurIPS)*, 2021, 34: 6417-6429.
- [90] WANG Y, SUN Y, LIU Z, et al. Dynamic graph cnn for learning on point clouds [J]. *ACM Transactions on Graphics (tog)*, 2019, 38(5): 1-12.
- [91] KLICPERA J, GIRI S, MARGRAF J T, et al. Fast and uncertainty-aware directional message passing for non-equilibrium molecules[J]. *arXiv preprint arXiv:2011.14115*, 2020.
- [92] SCHÜTT K, UNKE O, GASTEGGER M. Equivariant message passing for the prediction of tensorial properties and molecular spectra[C]//International Conference on Machine Learning. (ICML). 2021: 9377-9388.

- [93] LIU Y, WANG L, LIU M, et al. Spherical message passing for 3d graph networks[C]//International Conference on Learning Representations. (ICLR). 2021.
- [94] MILLER B K, GEIGER M, SMIDT T E, et al. Relevance of rotationally equivariant convolutions for predicting molecular properties[J]. arXiv preprint arXiv:2008.08461, 2020.
- [95] ANDERSON B, HY T S, KONDOR R. Cormorant: Covariant molecular neural networks[J]. Advances in Neural Information Processing Systems. (NeurIPS), 2019, 32.
- [96] FINZI M, STANTON S, IZMAILOV P, et al. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data[C]//International Conference on Machine Learning. (ICML). 2020: 3165-3176.
- [97] THÖLKE P, FABRITIIS G D. Equivariant Transformers for Neural Network based Molecular Potentials[C]//International Conference on Learning Representations. (ICLR). 2022.
- [98] LE T, NOÉ F, CLEVERT D A. Equivariant Graph Attention Networks for Molecular Property Prediction[J]. arXiv preprint arXiv:2202.09891, 2022.
- [99] BRANDSTETTER J, HESSELINK R, van der POL E, et al. Geometric and Physical Quantities improve E(3) Equivariant Message Passing[C]//International Conference on Learning Representations. (ICLR). 2022.
- [100] QI C R, SU H, MO K, et al. Pointnet: Deep learning on point sets for 3d classification and segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. (CVPR). 2017: 652-660.
- [101] QI C R, YI L, SU H, et al. Pointnet++: Deep hierarchical feature learning on point sets in a metric space[J]. Advances in Neural Information Processing Systems. (NeurIPS), 2017, 30.
- [102] KLOKOV R, LEMPITSKY V. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models[C]//Proceedings of the IEEE interna-

- tional conference on computer vision. (ICCV). 2017: 863-872.
- [103] LIU Y, FAN B, MENG G, et al. Densepoint: Learning densely contextual representation for efficient point cloud processing[C]//Proceedings of the IEEE international conference on computer vision. (ICCV). 2019: 5239-5248.
- [104] ZHANG K, HAO M, WANG J, et al. Linked dynamic graph cnn: Learning on point cloud via linking hierarchical features[J]. arXiv preprint arXiv:1904.10014, 2019.
- [105] THOMAS H, QI C R, DESCHAUD J E, et al. Kpconv: Flexible and deformable convolution for point clouds[C]//Proceedings of the IEEE/CVF international conference on computer vision. (ICCV). 2019: 6411-6420.
- [106] LEI H, AKHTAR N, MIAN A. Spherical kernel for efficient graph convolution on 3d point clouds[J]. IEEE transactions on pattern analysis and machine intelligence, 2020, 43(10): 3664-3680.
- [107] BOULCH A, PUY G, MARLET R. FKACConv: Feature-kernel alignment for point cloud convolution[C]//Proceedings of the Asian Conference on Computer Vision. (ACCV). 2020.
- [108] BOULCH A. ConvPoint: Continuous convolutions for point cloud processing [J]. Computers & Graphics, 2020, 88: 24-34.
- [109] SACHAN D S, ZAHEER M, SALAKHUTDINOV R. Revisiting lstm networks for semi-supervised text classification via mixed objective function[C] //Proceedings of the AAAI conference on artificial intelligence: vol. 33: 01. 2019: 6940-6948.
- [110] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.
- [111] JOULIN A, GRAVE E, BOJANOWSKI P, et al. Bag of tricks for efficient text classification[J]. arXiv preprint arXiv:1607.01759, 2016.
- [112] WU H, HU T, LIU Y, et al. Timesnet: Temporal 2d-variation modeling for general time series analysis[C]//International Conference on Learning Representations

- tations. (ICLR). 2022.
- [113] GU A, JOHNSON I, GOEL K, et al. Combining recurrent, convolutional, and continuous-time models with linear state space layers[J]. Advances in Neural Information Processing Systems. (NeurIPS), 2021, 34: 572-585.
- [114] LIU S, YU H, LIAO C, et al. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting[C] // International conference on learning representations. 2021.
- [115] ZHOU H, ZHANG S, PENG J, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting[C] // Proceedings of the AAAI conference on artificial intelligence: vol. 35: 12. 2021: 11106-11115.
- [116] KITAEV N, KAISER Ł, LEVSKAYA A. Reformer: The efficient transformer [J]. arXiv preprint arXiv:2001.04451, 2020.

致 谢

本文是对我攻读博士学位期间所做工作的总结。在读博期间，我得到过很多人的帮助，在此深深地感谢他们。

首先，我要向我的导师，申富饶教授，表达我最深切的感谢。在读博最开始，迷茫始终困惑着我，是您在学术上给予我大量的支持，并给我指定了确切的研究方向，消除了我的迷茫。每周一次的个人讨论让我不断在博士这条路上成长。您还在我遇到困难时给予我鼓励和帮助，让我顺利走过整个博士。您是良师，亦是益友。

在此，我还要感谢我一路同行的同门和朋友们。我和一起入学的硕士同学建立了深厚友谊，从系楼到宿舍楼的路上不再孤单。感谢你们在我遇到挑战时给予的帮助，在我需要放松时提供的欢笑，以及在我需要讨论时给予的智慧。此外，我还要感谢实习中遇到的同事们，从你们身上我学习了很多新鲜知识，激发了对研究的热情。

这里还要感谢我的家人，尤其是我的父母。您们无条件的爱和支持是我坚持下去的最大动力。感谢你们一直以来对我的理解和包容，感谢您们在我追求梦想的道路上始终如一地支持我。

最后，我要感谢在我博士阶段出现的小卓同学。博士是一条漫长的河，谢谢你成为我逆流而上的港湾，让这段旅途充满温暖。

在这段漫长而充满挑战的旅程中，有太多的人给予了我帮助和支持。虽然无法一一列举，但我心中充满感激之情。正是因为有了你们，我才能顺利走完我的博士生涯，谢谢你们！

A: 攻读博士学位期间的学术成果和获奖情况

攻读博士学位期间完成的学术成果

已发表论文

1. **Junyi An**, Chao Qu, Zhipeng Zhou, Fenglei Cao, Yinghui Xu, Yuan Qi, Furao Shen. Hybrid Directional Graph Neural Network for Molecules. International Conference on Learning Representations, 2024. (ICLR 国际学习表征大会, **Spotlight**)
2. **Junyi An**, Fengshan Liu, Furao Shen, Jian Zhao, Ruotong Li, Kepan Gao. IC neuron: An efficient unit to construct neural networks. Neural Networks, 2022.
3. Yuanjie Yan, **Junyi An**, Jian Zhao, Furao Shen. Hybrid optimization with unconstrained variables on partial point cloud registration. Pattern Recognition, 2023.

在投论文

1. **Junyi An**, Qianwei Tang, Chao Qu, Yujin Lu, Hongyuan Mei, Furao Shen, Jian Zhao. Group Equivariant Network for 3D Point Cloud Registration.
2. **Junyi An**, Yujin Lu, Lingming Zhang, Jian Zhao, Hongyuan Mei, Baile Xu, and Furao Shen. Local-Aware Dynamic Convolution.
3. **Junyi An**, Fengshan Liu, Jian Zhao and Furao Shen. IC Networks: Remodeling the Basic Unit for Convolutional Neural Networks.

攻读博士学位期间参与的科研课题

1. 参与科技部创新 2030 重大项目：“基于神经可塑性的脉冲网络高效学习机制与类脑智能系统”。
2. 参与国家自然科学基金面上项目：“基于深度感知增量式联想记忆神经网络的信息融合系统研究”。

3. 参与国家自然科学基金面上项目：“面向增量式无监督学习的新型神经网络研究”。

已授权发明专利

1. 一种基于静态求解和粒子滤波的鲁棒 tdoa 定位方法. Furao Shen, Zhi Zhou, Jun Li, **Junyi An**.
2. 一种复杂环境下的 TDOA 定位追踪方法. Furao Shen, Zhi Zhou, Feiyan Yu, **Junyi An**, Jun Li.
3. 一种复杂环境下的三维定位追踪方法. Furao Shen, Bilan Lai, **Junyi An**, Jian Zhao.

攻读博士学位期间已出版的专著

1. 自组织增量学习神经网络 [M]. 电子工业出版社. 申富饶. 参与该书第四章写作和修改。

攻读博士学位期间获得的主要奖励

1. 博士英才奖学金一等奖，2021 年。
1. 优秀研究生，2021 年。
3. ICCV 会议 MVP 竞赛点云配准赛道一等奖，2021 年。