

Geometric Deep Learning
Grids, Groups, Graphs,
Geodesics, and Gauges
几何深度学习
格网, 群, 图, 测地线, 规范

Michael M. Bronstein¹, Joan Bruna², Taco Cohen³, Petar Veličković⁴ (著)
子仁⁵(译)

2021 年 5 月 24 日

¹Imperial College London / USI IDSIA / Twitter

²New York University

³Qualcomm AI Research. Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

⁴DeepMind

⁵NUDT, luojunren17@nudt.edu.cn

目录

前言	1
注记	3
1 引言	4
2 高维学习	5
2.1 函数正则化归纳偏置	6
2.2 维数诅咒	7
3 几何先验	10
3.1 对称, 表示与不变性	11
3.2 同构与自同构	16
3.3 变形稳定性	18
3.4 尺度分离	21
3.5 几何深度学习的蓝图	25
4 几何域: 五个世代	29
4.1 图与集合	29

4.2	格网与欧几里得空间	33
4.3	群与齐次空间	38
4.4	测地线与流形	41
4.5	规范与丛	53
4.6	几何图与网格	57
5	几何深度学习模型	64
5.1	卷积神经网络	65
5.2	群等变卷积神经网络	70
5.3	图神经网络	73
5.4	Deep Sets, Transformers, 潜在图推理	75
5.5	等变消息传递网络	78
5.6	内在网格卷积神经网络	80
5.7	递归神经网络	84
5.8	长短期记忆网络	90
6	问题与应用	97
6.1	化学与药物设计	97
6.2	药物重新定位	98
6.3	蛋白质生物学	98
6.4	推荐系统与社交网络	99
6.5	交通预测	100

6.6	物体识别	101
6.7	博弈对抗	102
6.8	文本与语音合成	103
6.9	医疗保健	104
6.10	粒子物理和天体物理	106
6.11	虚拟与增强现实	107
7	历史视角	109
7.1	数学与物理中的对称	109
7.2	机器学习中早期使用对称	110
7.3	图神经网络	111
7.4	计算化学	112
7.5	节点嵌入	112
7.6	概率图模型	113
7.7	Weisfeiler-Lehman 形式化	114
7.8	高阶方法	115
7.9	信号处理与调和分析	116
7.10	图与网格上信号处理	117
7.11	计算机图形学与几何处理	117
7.12	算法推理	119
7.13	几何深度学习	120

前言

至欧几里得《几何原本》以来的近两千年, 单词“几何”与“欧几里得几何”是同义词, 因为没有几何类型的几何存在. 欧几里得几何的垄断在 19 世纪被终结, 其中有由洛巴切夫斯基, 博利亚伊, 高斯和黎曼构建的非欧几里得几何. 到 19 世纪末, 这些研究已分散到各个领域, 数学家和哲学家就这些几何的有效性以及它们之间的关系以及“真实几何”的本质进行了辩论.

年轻的数学家费利克斯·克莱因 (Felix Klein) 指出了解决这一难题的方法, 他于 1872 年被任命为小规模巴伐利亚州埃尔兰根大学的教授在一份作为埃尔兰根纲领进入数学编年史的研究计划书中, 克莱因提出将几何作为不变量的研究, 即在某种变换下不变的性质, 称为几何的对称性. 这种方法做出了一些澄清, 表明当时已知的各种几何形状可以通过适当选择对称变换来定义, 并使用群论的语言来形式化. 例如, 欧几里得几何与长度和角度有关, 因为这些属性由欧几里得变换 (旋转和平移) 保留, 而仿射几何研究平行性, 由仿射变换保留. 当考虑各自的群时, 这些几何之间的关系是显而易见, 因为欧几里得群是仿射群的子群, 而仿射群又是射影变换群的子群.

埃尔兰根纲领对几何学的影响非常深远. 此外, 它蔓延到其他领域, 特别是物理学, 在那里对称原理允许从对称的第一性原理 (一个被称为诺瑟定理的惊人结论) 中导出守恒定律, 甚至允许将基本粒子分类为对称群的不可约表示. 用范畴理论 (*Category theory*) 的创立者萨缪尔·艾伦伯和桑德斯·麦克兰恩的话来说, 现在在纯数学中普遍存在的范畴理论可以被视为克莱因“埃尔兰根纲领”的延续, 在这个意义上, 一个几何空间及其变换群被推广到一个范畴及其映射代数”.

在写作时, 深度学习领域的状态有点让人想起十九世纪的几何领域. 对于各种各样的数据, 有一个名副其实神经网络体系结构动物园, 但是很少有统一的原则. 与过去一样, 这使得很难理解各种方法之间的关系, 不可避免地导致不同应用领域中相同概念的重新发明和品牌化. 对于一个试图学习该领域的新手来说, 去吸收大量多余的想法是一场真正的噩梦.



根据一种普遍的看法, “埃尔兰根纲领 (Erlangen Programme)” 于 1872 年 10 月在克莱因的就职演说中发表. 克莱因确实做了这样的演讲 (尽管是在同年 12 月 7 日), 但这是针对非数学观众的, 并且主要关注他关于数学教育的想法. 如今被称作“埃尔兰根纲领”实际是他为他的教授任命而准备的研究计划书 *em Vergleichende Betrachtungen über neuere geometrische Forschungen* (“几何学最新研究比较综述”). 见 [Tobies \(2019\)](#). 见 [Marquis \(2009\)](#).

在本文中, 我们尝试将“埃尔兰根纲领”的思维模式应用到深度学习领域, 最终目标是实现该领域的系统化和“点与点之间的联系”。我们将这种几何化的尝试称为“几何深度学习”, 并忠实于费利克斯·克莱因的精神, 提出从对称和不变性的第一原则中导出不同的归纳偏置和实现它们的网络架构。特别是, 我们专注于一大类神经网络, 用于分析非结构化集合、格网、图形和流形, 并表明它们可以以统一的方式理解为保留这些域的结构和对称性的方法。

我们相信, 这篇文章将吸引广大深入学习的研究人员、实践者和爱好者。新手可以将它作为几何深度学习的概述和介绍。一个经验丰富的深度学习专家可能会发现从基本原则和一些令人惊讶的联系中获得熟悉的体系结构的新方法。从业者可能会获得如何解决各自领域问题的新见解。

在现代机器学习这样一个快节奏的领域, 写这样一个文本的风险是, 它在天亮之前就变得过时和不相关了。专注于基础, 我们希望我们讨论的关键概念将超越它们的具体实现, 正如 Claude Adrien Helvétius 所说,“对某些原则的认识很容易取代对某些事实的认识”。

“对某些原则的了解很容易弥补对某些事实了解的不足。”
([Helvétius, 1759](#))

注记

Ω, u	域, 域上点
$x(u) \in \mathcal{X}(\Omega, \mathcal{C})$	域上信号 $x : \Omega \rightarrow \mathcal{C}$
$f(x) \in \mathcal{F}(\mathcal{X}(\Omega))$	域上信号的函数 $f : \mathcal{X}(\Omega) \rightarrow \mathcal{Y}$
$\mathfrak{G}, \mathfrak{g}$	群, 群元素
$\mathfrak{g}.u, \rho(\mathfrak{g})$	群作用, 群表示
$\mathbf{X} \in \mathcal{C}^{ \Omega \times s}$	离散域上信号的矩阵表示
$\mathbf{x}_u \in \mathcal{C}^s$	离散信号 \mathbf{X} 在元素 $u \in \Omega$ 处向量表示
$x_{uj} \in \mathcal{C}$	离散信号 \mathbf{X} 在元素 $u \in \Omega$ 处第 j 个分量的标量表示
$\mathbf{F}(\mathbf{X})$	离散信号上的函数, 返回矩阵形式的离散信号
$\tau : \Omega \rightarrow \Omega$	域自同构
$\eta : \Omega \rightarrow \Omega'$	两个不同域之间同构
$\sigma : \mathcal{C} \rightarrow \mathcal{C}'$	激活函数 (逐点非线性)
$G = (\mathcal{V}, \mathcal{E})$	带点 \mathcal{V} 和边 \mathcal{E} 的图
$\mathcal{T} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$	带点 \mathcal{V} , 边 \mathcal{E} , 面 \mathcal{F} 的网格
$x \star \theta$	与滤波器 θ 的卷积
S_v	变换算子
φ_i	基函数
$T_u\Omega, T\Omega$	在 u 处的切空间, 切丛
$X \in T_u\Omega$	切向量
$g_u(X, Y) = \langle X, Y \rangle_u$	黎曼测度
$\ell(\gamma), \ell_{uv}$	曲线 γ 的长度, 边 (u, v) 上的离散测度

1 引言

过去十年见证了数据科学和机器学习的实验革命, 其缩影是深度学习方法. 事实上, 许多以前被认为遥不可及的高维学习任务——如计算机视觉、围棋或蛋白质折叠——在适当的计算规模下实际上是可行的. 值得注意的是, 深度学习的本质是建立在两个简单的算法原则之上的: 第一, 表示或特征学习 (*feature learning*) 的概念, 其中自适应的, 通常是分层的, 特征为每个任务捕获适当的规律性概念, 第二, 通过局部梯度下降的学习, 通常被实现为反向传播 (*backpropagation*).

虽然学习高维度的一般函数是一个讨厌的估计问题, 但大多数感兴趣的任务都不是一般的, 并且带有来自底层物理世界的低维度和结构的基本预定义规则. 本文通过统一的几何原理来揭示这些规律, 这些原理可以应用于广泛的应用领域.

利用一个大系统的已知对称性是对抗维度诅咒的一种强有力的经典疗法, 并且构成了大多数物理理论的基础. 深度学习系统也不例外, 从早期开始, 研究人员就采用神经网络来利用物理测量产生的低维几何, 例如图像中的网格、时间序列中的序列或分子中的位置和动量, 以及它们相关的对称性, 例如平移或旋转. 在我们的整个论述中, 我们将把这些模型, 以及许多其他模型, 描述为相同的基本几何规则的自然实例.

这种以“埃尔兰根纲领”为精神的“几何统一”努力有双重目的: 一方面, 它提供了一个通用的数学框架来研究最成功的神经网络架构, 如 CNNs、RNNs、GNNs 和 Transformers. 另一方面, 它给出了一个建设性的过程, 将先前的物理知识结合到神经体系结构中, 并为构建未来尚未发明的体系结构提供了有原则的方法.

在继续之前, 值得注意的是, 我们的工作涉及表示学习架构 (*representation learning architectures*) 和利用其中的数据对称性. 许多令人兴奋的可以使用这种表示的途径 (*pipelines*)(如自我监督学习、生成模型或强化学习) 并不是我们关注的中心. 因此, 我们不会深入综述有影响的神经网络途径, 如

这同样适用于用于优化 (*optimising*) 或规范 (*regularising*) 我们的架构的技术, 例如 Adam (Kingma and Ba, 2014), dropout (Srivastava et al., 2014) 和批归一化 (batch normalisation) (Ioffe and Szegedy, 2015).

变分自动编码器 (variational autoencoders (Kingma and Welling, 2013))、生成对抗网络 (generative adversarial networks (Goodfellow et al., 2014))、标准化流 (normalising flows (Rezende and Mohamed, 2015))、深度 Q 网络 (deep Q-networks (Mnih et al., 2015))、近端策略优化 (proximal policy optimisation (Schulman et al., 2017)) 或深度互信息最大化 (deep mutual information maximisation (Hjelm et al., 2019)). 尽管如此, 我们认为, 我们将侧重的原则在所有这些领域都非常重要.

此外, 虽然我们试图撒下一张相当宽的网来说明我们几何蓝图的力量, 但我们的工作并没有试图准确总结几何深度学习的全部现有研究成果. 相反, 我们深入研究了几个著名的架构, 以展示这些原则, 并在现有的研究中对它们进行基础研究, 希望我们为读者留下足够的参考资料, 以便有意义地将这些原则应用到他们遇到或设计的任何未来的几何深度架构中.

2 高维学习

监督机器学习, 在其最简单的形式中, 考虑一组来自定义在 $\mathcal{X} \times \mathcal{Y}$ 上的服从数据分布 P 的 N 个独立同分布 (*i.i.d.*) 观察值 $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, 其中 \mathcal{X} 和 \mathcal{Y} 分别是数据和标签域. 这种设置的定义特征是 \mathcal{X} 是一个高维空间: 人们通常假设 $\mathcal{X} = \mathbb{R}^d$ 是一个 d 维的欧几里得空间.

让我们进一步假设标签 y 由未知函数 f 生成, 满足 $y_i = f(x_i)$, 并且学习问题简化为使用参数化函数类 $\mathcal{F} = \{f_{\theta \in \Theta}\}$ 来估计函数 f . 神经网络是这种参数函数类的常见实现, 在这种情况下, $\theta \in \Theta$ 对应于网络权重. 在这种理想的设置中, 标签中没有噪声, 现代深度学习系统通常在所谓的插值机制 (*interpolating regime*) 下运行, 其中对于所有 $i = 1, \dots, N$ 估计的 $\tilde{f} \in \mathcal{F}$ 满足 $\tilde{f}(x_i) = f(x_i)$. 学习算法的性能是根据从 P 中抽取新样本的期望性能 (*expected performance*), 使用一些损失 $L(\cdot, \cdot)$ 来衡量的

$$\mathcal{R}(\tilde{f}) := \mathbb{E}_P L(\tilde{f}(x), f(x)),$$

其中平方损失 $L(y, y') = \frac{1}{2}|y - y'|^2$ 是最常用的.

统计学习理论关注的是基于集中不等式的更精确的广义概念; 我们将在未来的工作中回顾其中的一些内容.

因此, 一个成功的学习方案需要对 f 的正则性 (regularity) 或归纳偏置 (inductive bias) 概念进行编码, 通过函数类 \mathcal{F} 的构造和正则化的使用来施加. 我们将在下一节简要介绍这些概念.

2.1 函数正则化归纳偏置

现代机器学习使用大型、高质量的数据集, 这些数据集与适当的计算资源一起, 激发了丰富的函数类 \mathcal{F} 的设计, 这些函数类 \mathcal{F} 具有插值如此大的数据的能力. 这种思维模式很适合神经网络, 因为即使是最简单的架构选择也会产生稠密的 (dense) 函数类. 逼近几乎任意函数的能力是各种通用逼近定理 (Universal Approximation Theorems) 的主题; 几个这样的结果在 1990 年代被应用数学家和计算机科学家证明和推广 (例如, 见 Cybenko (1989); Hornik (1991); Barron (1993); Leshno et al. (1993); Maierov (1999); Pinkus (1999)).

集合 $\mathcal{A} \subset \mathcal{X}$ 的闭集如果满足

$$\mathcal{A} \cup \left\{ \lim_{i \rightarrow \infty} a_i : a_i \in \mathcal{A} \right\} = \mathcal{X}.$$

则集合被称为稠密的 (dense).

这意味着 \mathcal{X} 中的任何一点都任意靠近 \mathcal{A} 中的一点. 一个典型的通用近似结果表明, 例

如由两层感知器

$f(\mathbf{x}) = \mathbf{c}^\top \text{sign}(\mathbf{A}\mathbf{x} + \mathbf{b})$ 表示的函数类在 \mathbb{R}^d 上的连续函数空间中是稠密的.

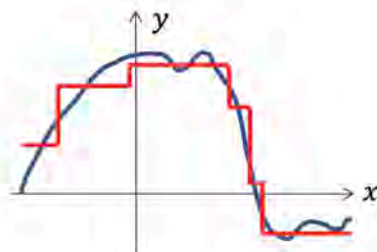
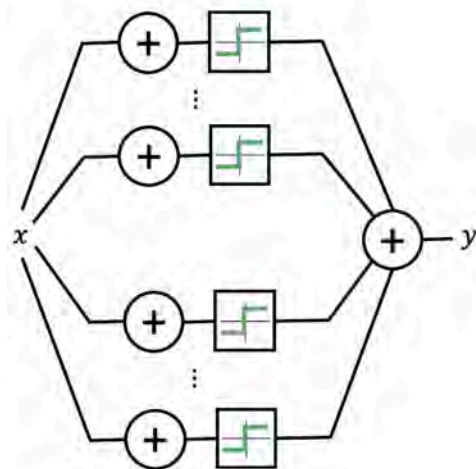


图 1: 多层感知器 (Rosenblatt, 1958) 是最简单的前馈神经网络, 是通用的逼近器: 只有一个隐藏层, 它们可以表示阶跃函数的组合, 允许以任意精度逼近任何连续函数.

然而, 通用近似并不意味着没有归纳偏置. 给定一个具有泛逼近性的假设空间 \mathcal{F} , 我们可以定义一个复杂度度量 $c: \mathcal{F} \rightarrow \mathbb{R}_+$ 并将我们的插值问题重新

定义为

$$\tilde{f} \in \arg \min_{g \in \mathcal{F}} c(g) \quad \text{s.t.} \quad g(x_i) = f(x_i) \quad \text{for } i = 1, \dots, N,$$

即, 我们在我们的假设类中寻找最正则 (regular) 的函数. 对于标准函数空间, 这种复杂性度量可以被定义为一个范数 (norm), 使 \mathcal{F} 成为一个巴拿赫空间 (Banach space), 并允许在泛函分析中利用过多的理论结果. 在低维中, 样条是函数逼近的主要工具. 它们是可以由上面的公式表示, 用一个范数捕捉光滑性的经典概念, 比如三次样条的二阶导数的平方范数 $\int_{-\infty}^{+\infty} |f''(x)|^2 dx$.

非正式地说, 一个范数 $\|x\|$ 可以看作是向量 x 的一个“长度”. 巴拿赫空间是一个配有范数的完备向量空间.

在神经网络的情况下, 复杂度度量 c 可以用网络权重来表示, 即 $c(f_{\theta}) = c(\theta)$. 网络权重的 L_2 范数, 即权重衰减 (weight decay), 或所谓的路径范数 (path-norm, (Neyshabur et al., 2015)) 是深度学习文献中的流行选择. 从贝叶斯的角度来看, 这种复杂性度量也可以被解释为感兴趣函数的先验的负对数. 更一般地说, 这种复杂性可以通过将其纳入经验损失 (导致所谓的结构风险最小化) 来显式计算, 或者通过某种优化方案来隐式计算. 例如, 众所周知, 欠定最小二乘目标上的梯度下降将选择具有最小 L_2 范数的插值解. 这种隐式正则化结果对现代神经网络的扩展是当前研究的主题 (例如, 参见Blanc et al. (2020); Shamir and Vardi (2020); Razin and Cohen (2020); Gunasekar et al. (2017)). 总之, 一个自然的问题出现了: 如何定义有效的先验来捕捉现实世界预测任务的期望正则性 (expected regularities) 和复杂性?

2.2 维数诅咒

虽然低维 ($d = 1, 2$ 或 3) 插值是一项经典的信号处理任务, 使用日益复杂的正则类 (如样条插值、小波、曲波或脊波) 对估计误差进行非常精确的数学控制, 但高维问题的情况完全不同.

为了传达这一思想的本质, 让我们考虑一个经典的正则性 (regularity) 概念, 它可以很容易地扩展到高维: 1-李普希茨函数 $f: \mathcal{X} \rightarrow \mathbb{R}$, 即对于所有 $x, x' \in \mathcal{X}$, 满足 $|f(x) - f(x')| \leq \|x - x'\|$ 的函数. 这一假设只要求目标函数是局部 (locally) 光滑的, 即如果我们稍微扰动输入 x (用范数 $\|x - x'\|$ 衡

量), 输出 $f(x)$ 不允许有太大的变化. 如果我们对目标函数 f 的唯一了解是它是 1-李普希茨, 那么我们期望需要多少次观测才能确保我们的估计 \tilde{f} 接近 f ? 图2显示, 一般的答案在 d 维上必然是指数的, 这表明随着输入维的增加, 李普希茨类的增长“太快”: 在许多即使是适度的 d 维的应用中, 样本的数量将大于宇宙中的原子数量. 如果用一个全局光滑性假设来代替李普希茨类, 比如 Sobolev 类 $\mathcal{H}^s(\Omega_d)$, 情况不会更好. 事实上, 经典的结果 (Tsybakov, 2008) 为阶数为 $\epsilon^{-d/s}$ 的 Sobolev 类建立了逼近和学习的极大极小率, 表明 f 上的额外平滑度假设仅在 $s \propto d$ 时改善统计图像, 这在实践中是不现实的假设.

如果 $f \in L^2(\Omega_d)$ 且广义的 s 阶导数是平方可积的 $\int |\omega|^{2s+1} |\hat{f}(\omega)|^2 d\omega < \infty$, 则函数 f 属于 Sobolev 类 $\mathcal{H}^s(\Omega_d)$, 其中 \hat{f} 是 f 的傅里叶变换; 参见第 4.2 节.

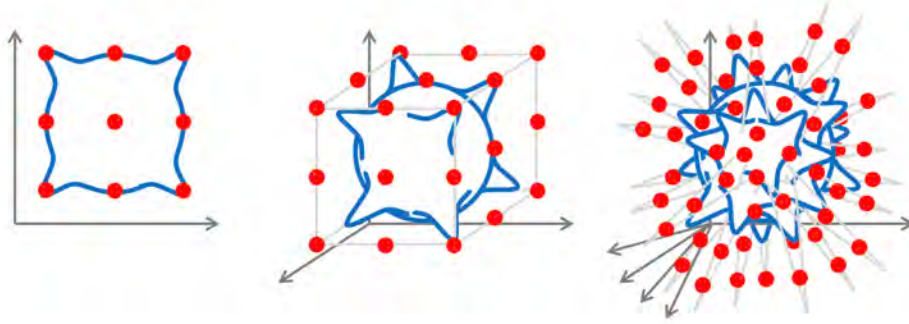


图 2: 我们考虑一个李普希茨函数 $f(x) = \sum_{j=1}^{2^d} z_j \phi(x - x_j)$, 其中 $z_j = \pm 1$, $x_j \in \mathbb{R}^d$ 位于每个象限, ϕ 是一个局部支持的李普希茨“凸块”. 除非我们在第 2^d 个象限的大部分中观察到这个函数, 否则我们在预测它时会不断出错. 这个简单的几何论证可以通过最大差异 (Maximum Discrepancy) 的概念 (von Luxburg and Bousquet, 2004) 来形式化, 为李普希茨类定义为 $\kappa(d) = \mathbb{E}_{x, x'} \sup_{f \in \text{Lip}(1)} \left| \frac{1}{N} \sum_l f(x_l) - \frac{1}{N} \sum_l f(x'_l) \right| \simeq N^{-1/d}$, 衡量两个独立 N 样本预期之间的最大预期差异. 确保 $\kappa(d) \simeq \epsilon$ 需要 $N = \Theta(\epsilon^{-d})$; 相应的示例 $\{x_l\}_l$ 定义了一个域的 ϵ -网络. 对于直径为 1 的 d 维欧氏域, 其大小呈指数增长为 ϵ^{-d} .

全连接神经网络定义了函数空间, 这些空间允许更灵活的正则性 (regularity) 概念, 这是通过考虑复杂性函数 c 的权重而获得的. 特别是, 通过选择稀疏促进正则化 (regularisation), 他们有能力打破这种维数灾难 (Bach, 2017). 然

而, 这是以对目标函数 f 的性质做出强有力的假设为代价的, 例如 f 依赖于输入的低维投影的集合 (见图3). 在大多数现实世界的应用中 (如计算机视觉、语音分析、物理或化学), 感兴趣的函数往往表现出复杂的长期相关性, 无法用低维投影来表达 (图3), 这使得这一假设不现实. 因此, 有必要通过利用物理域的空间结构和 f 的几何先验来定义正则性 (regularity) 的另一个来源, 正如我们在下一节3中所描述的.

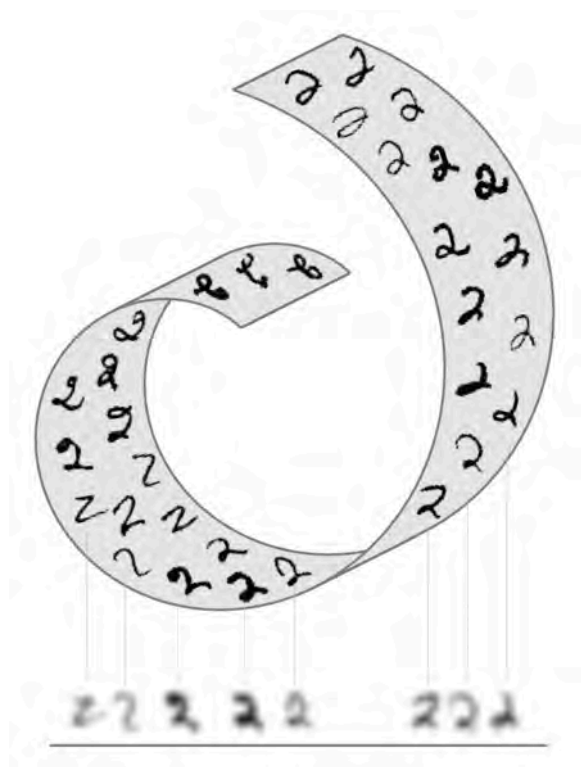


图 3: 如果对于某个未知的 $\mathbf{A} \in \mathbb{R}^{k \times d}, k \ll d$, 假设未知函数 f 很好地近似为 $f(\mathbf{x}) \approx g(\mathbf{A}\mathbf{x})$ 那么浅层神经网络可以捕捉这种归纳偏置, 例如Bach (2017). 在典型的应用中, 这种对低维投影的依赖是不现实的, 如这个例子所示: 低通滤波器将输入图像投影到低维子空间; 虽然它传达了大部分语义, 但大量信息丢失了.

3 几何先验

现代数据分析是高维学习的代名词. 虽然第 2.1 节的简单论证表明, 由于维数的诅咒, 从一般的高维数据中学习是不可能的, 但在物理结构数据中, 我们希望使用两个基本原理: 对称和尺度分离. 在本文中考虑的设置中, 这个额外的结构通常来自于输入信号的基本领域的结构: 我们将假设我们的机器学习系统操作于某些领域 Ω 上的信号 (函数). 而在许多情况下, 领域 Ω 上的点的线性组合是没有明确定义的, 我们可以将矢量空间 Ω 上的信号线性组合, 也就是说, 信号的空间构成了一个矢量空间. 此外, 由于我们可以定义信号之间的内积, 这个空间是一个 *Hilbert* 空间.

Ω 必须是一个矢量空间, 以使表达式 $\alpha u + \beta v$ 有意义.

当 Ω 有一些额外的结构, 我们可能可以进一步限制 $\mathcal{X}(\omega, \mathcal{C})$ 中信号的类型. 例如, 当 Ω 是一个光滑流形, 我们可能要求信号是光滑的. 只要可能, 为简洁起见, 我们将省略 \mathcal{C} 的范围

Ω 上复值 (\mathcal{C} -值) 信号的空间 (对于 Ω 是一个集合, 可能带有附加结构, 而 \mathcal{C} 是一个向量空间, 其维度称为通道)

$$\mathcal{X}(\Omega, \mathcal{C}) = \{x : \Omega \rightarrow \mathcal{C}\} \quad (1)$$

是一个拥有矢量空间结构的泛函空间. 信号的加法和数乘定义如下:

$$(\alpha x + \beta y)(u) = \alpha x(u) + \beta y(u) \quad \text{for all } u \in \Omega,$$

其中 α, β 是实的标量. 给定复空间 \mathcal{C} 上的一个内积 $\langle v, w \rangle_{\mathcal{C}}$ 和集合 Ω 上一个测度 μ (我们可以在该集合上定义积分), 我们可以在 $\mathcal{X}(\Omega, \mathcal{C})$ 上定义内积为

$$\langle x, y \rangle = \int_{\Omega} \langle x(u), y(u) \rangle_{\mathcal{C}} d\mu(u). \quad (2)$$

当 Ω 的域是离散的, μ 可以选择为可数测度, 这个时候积分变成了求和. 为了简洁, 我们将省略测度并且使用 du

例如, 取 $\Omega = \mathbb{Z}_n \times \mathbb{Z}_n$ 为一个二维的 $n \times n$ 格网, x 为一个 RGB 图像 (即一个信号 $x : \Omega \rightarrow \mathbb{R}^3$), f 是一个作用于 $3n^2$ 维输入的函数 (例如单层感知器). 我们将在下面更详细地看到, 域 Ω 通常具有一定的结合结构和对称性. 尺度分离的结果来自于我们将信号转移到域的粗糙版本时保留信号的重要特征的能力 (在我们的例子中, 通过粗化底层网络对图像进行子采样).

我们将证明这两个原理, 我们一般将其称为几何先验, 在大多数现代深度学

习体系结构中都很突出. 在上述图像的情况下, 几何先验以共享权值 (利用平移对称) 和池化 (利用尺度分离) 的卷积滤波器的形式构建到卷积神经网络 (CNN) 中. 将这些思想扩展到其它领域, 如图和流形, 并展示几何先验如何从基本原理中产生, 是几何深度学习的主要目标, 也是我们的主题.

3.1 对称, 表示与不变性

通俗的说, 物体或系统的对称性是一种变换, 使物体或系统的某一特性保持不变或不变. 这种转换可以是平滑的、连续的, 也可以是离散的. 对称性在许多机器学习任务中无处不在. 例如, 在计算机视觉中, 物体的类别在平移作用下是不变的, 所以平移是视觉对象分类问题中的对称性. 在计算化学中, 不依赖分子在空间中的方向来预测其性质的任务需要旋转不变性. 离散对称在描述粒子系统时自然出现, 在这些粒子系统中, 粒子没有正则排序, 因此可以任意排列, 在许多动力系统中也是如此, 通过时间反转对称 (例如详细平衡系统或牛顿第二运动定律). 正如我们将在 4.1 节中看到的, 排列对称也是图结构数据分析的中心.

对称群 一个物体的对称的几何满足若干性质. 首先, 对称可以组合获得新的对称: 如果 g 和 h 是两个对称, 那么它们的组合 $g \circ h$ 和 $h \circ g$ 也是对称. 原因是, 如果两个变换都保持对象不变, 那么转换的复合也保持不变, 因此复合也是一个对称. 此外, 对称总是可逆的, 而其逆也是一个对称. 这表明所有对称的集合构成了一个被称为群的代数对象. 由于这些对象将是几何深度学习的数学模型的核心, 它们值得正式定义和详细讨论:

我们将遵循在群论中使用的并置符号约定, $g \circ h = gh$, 它应该从右到左理解: 我们首先应用 h , 然后是 g . 次序很重要, 因为在很多情况下, 对称是非交换的. 熟悉李群的读者可能会对我们选择用尖角字体表示群元素感到不安, 因为它是李代数的一种常见符号.

一个群是一个集合 \mathcal{G} 附带有一个二元运算 $\circ: \mathcal{G} \times \mathcal{G} \rightarrow \mathcal{G}$ 称为组合 (为了简洁, 通过素并列标记 $g \circ h = gh$) 满足以下定理:

结合律: $(gh)k = g(hk)$ 对于所有 $g, h, k \in \mathcal{G}$.

单位元: 存在一个唯一元素 $e \in \mathcal{G}$ 满足 $eg = ge = g$ 对于所有 $g \in \mathcal{G}$.

可逆性: 对于每个 $g \in \mathcal{G}$ 存在一个唯一可逆的 $g^{-1} \in \mathcal{G}$ 满足 $gg^{-1} = g^{-1}g = e$.

闭合性: 群在二元运算组合作用下是闭合的, 也就是说, 对于每个 $g, h \in \mathcal{G}$, 我们有 $gh \in \mathcal{G}$.

注意 可交换性不是这个定义的一部分, 也就是说, 我们可能有 $gh \neq hg$. 群满足 $gh = hg$ 对于所有 $g, h \in \mathcal{G}$ 称为是可交换的或者 *Abelian*.

以挪威数学家 Niels Henrik Abel (1802–1829) 的名字命名.

虽然群可以很大, 甚至可以是无限维的, 但是它们通常由少数几个元素的组合组成, 少数的几个元素称为群生成元. 正式地, \mathcal{G} 称为是由一个子集 $S \subseteq \mathcal{G}$ (称为群生成元) 生成, 如果每个元素 $g \in \mathcal{G}$ 可以被写成集合 S 中有限个元素和它们的逆有限次组合. 例如, 一个等变三角形的对称群 (二面体群 D_3) 由 60° 旋转和镜像对称生成 (Figure 4). 我们将在下面详细讨论的一维平移群, 是由无穷小的位移产生的, 这是可微对称李群的一个例子.

李群有一个可微的流形结构. 我们将在章节 4.3 学习的一个这样的例子: 特殊的正交群 $SO(3)$, 它是一个 3 维流形

注意, 我们已经将群定义成一个抽象的对象, 没有说明群元素是什么样子的 (例如, 一些领域的变换群), 我们仅仅怎样组合. 因此, 不同种类的物体可能具有相同的对称群. 例如, 前面提到的三角形的旋转和对称镜像组合的群和三元组排列的置换组成的群是相同的 (我们可以使用旋转和反射以任何形式置换三角形的角—见图4).

图4所示的关系图 (其中每个节点与一个群元素相关联, 每个箭头与一个生成元相关联), 称为 *Cayley* 图.

群作用与群表示 我们不是将群看作是抽象实体, 我们更感兴趣的是群如何作用于数据. 由于我们认为我们的数据本质上是一些域 Ω , 我们将研究群如何作用于 Ω (例如平面点的平移), 并从中获得同样的群在信号空间 $\mathcal{X}(\Omega)$ 上的作用 (例如平面图像和特征图的平移).

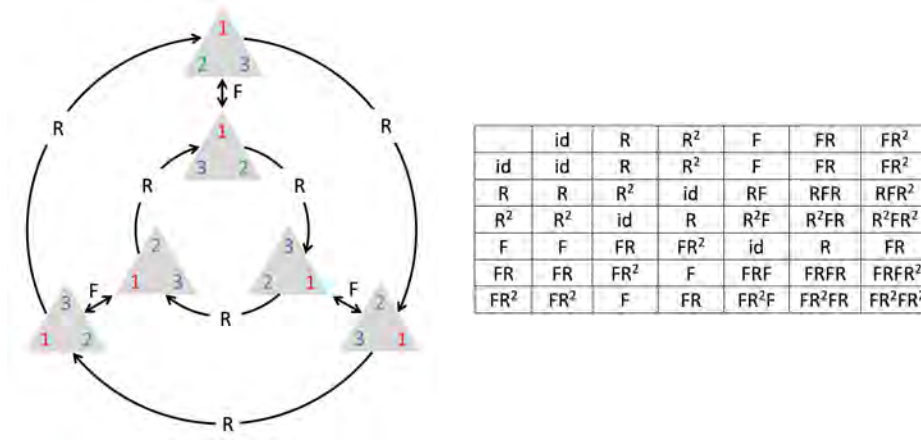


图 4: 左: 角被标为 1, 2, 3 的等边三角形, 以及三角形所有可能的旋转和反射. 三角形的旋转/对称反射组成的群 D_3 仅仅由两个元素生成 (旋转 60° R 和反射 F), 并且和三个元素置换群 Σ_3 是相同的. 右: 群 D_3 的乘法表. g 行和 h 列中的元素对应于元素 gh .

群 \mathfrak{G} 在集合 Ω 上的一个群作用 定义为一个映射 $(g, u) \mapsto g.u$, 该映射以一种与群运算兼容的方式将群元素 $g \in \mathfrak{G}$, 点 $u \in \Omega$ 和集合 Ω 上的其余元素关联, 也就是 $g.(h.u) = (gh).u$ 对于所有 $g, h \in \mathfrak{G}$ 和 $u \in \Omega$. 我们将在后续章节中看到很多群作用的例子.

技术上讲, 我们在此处定义的是左群作用.

例如, 在平面上, 欧式群 $E(2)$ 是 \mathbb{R}^2 的转换组成的群, 该转换能保持欧式距离, 并且该转换可以由平移、旋转和反射组成. 然而, 同样的群也可以作用于平面上的图像空间 (通过平移、旋转和翻转像素的格网), 以及通过神经网络学习的表示空间. 更准确地说, 如果我们有一个群 \mathfrak{G} 作用于 Ω , 我们自动获得群 \mathfrak{G} 作用于空间 $\mathcal{X}(\Omega)$:

保持距离的变换称为等距. 根据 Klein 的纲领计划, 经典欧式几何起源于这个群

$$(g.x)(u) = x(g^{-1}u). \quad (3)$$

由于 g 的逆, 这确实是一个有效的群作用, 我们有 $(g.(h.x))(u) = ((gh).x)(u)$.

最重要群作用, 也是我们整个文章中将反复遇到的, 是 线性群作用, 也称为群表示. 方程 (3) 中对信号确实是线性的, 在这个意义上

$$g.(\alpha x + \beta x') = \alpha(g.x) + \beta(g.x')$$

对于任何标量 α, β 和信号 $x, x' \in \mathcal{X}(\Omega)$. 我们可以将线性作用描述为映射 $(\mathbf{g}, x) \mapsto \mathbf{g}.x$, 映射关于变量 x 是线性的; 或者等价的, 通过局部套用, 可以将线性作用描述为映射 $\rho: \mathfrak{G} \rightarrow \mathbb{R}^{n \times n}$, 映射给每一个群元素 \mathbf{g} 分配一个可逆矩阵 $\rho(\mathbf{g})$. 矩阵的维度 n 一般是任意的, 不一定与群的维数或者 Ω 的维数有关, 但在深度学习的应用中, n 通常是群作用的特征空间的维数. 例如, 我们可以让 2D 平移群作用于 n 个像素的图像空间.

当 Ω 是无限维的, 信号 $\mathcal{X}(\Omega)$ 的空间是无限维的, 这种情况下 $\rho(\mathbf{g})$ 是这个空间上的一个线性算子, 而不是一个有限维矩阵. 但在实践中, 我们必须始终离散到有限的格网中.

与一般的群作用一样, 将矩阵分配给群元素应该与群作用兼容. 更具体地说, 表示复合群元素 \mathbf{gh} 的矩阵应该等于 \mathbf{g} 和 \mathbf{h} 的表示的矩阵乘积:

群 \mathfrak{G} 的一个 n 维实表示是一个映射 $\rho: \mathfrak{G} \rightarrow \mathbb{R}^{n \times n}$, 分配每一个 $\mathbf{g} \in \mathfrak{G}$ 一个可逆矩阵 $\rho(\mathbf{g})$, 并且满足条件 $\rho(\mathbf{gh}) = \rho(\mathbf{g})\rho(\mathbf{h})$, 对于所有 $\mathbf{g}, \mathbf{h} \in \mathfrak{G}$. 一个表示称为是酉的或者正交的如果矩阵 $\rho(\mathbf{g})$ 是酉的或者正交的, 对于所有 $\mathbf{g} \in \mathfrak{G}$.

相似地, 一个复表示是一个映射 $\rho: \mathfrak{G} \rightarrow \mathbb{C}^{n \times n}$ 满足同样的方程.

用群表示的语言来写, 群 \mathfrak{G} 在信号 $x \in \mathcal{X}(\Omega)$ 上的作用定义为 $\rho(\mathbf{g})x(u) = x(\mathbf{g}^{-1}u)$. 我们再次证明

$$(\rho(\mathbf{g})(\rho(\mathbf{h})x))(u) = (\rho(\mathbf{gh})x)(u).$$

不变与等变函数 信号 $\mathcal{X}(\Omega)$ 下的域 Ω 的对称性对定义在此类信号上的函数 f 施加了结构. 它被证明是一个强大的归纳偏差, 通过减少可能的插值空间 $\mathcal{F}(\mathcal{X}(\Omega))$ 来提高学习效率, 使之满足对称性先验. 我们将在本文探讨两种重要的情况是不变函数和等变函数.

一般来说, f 既依赖于信号又依赖于域, 也即 $\mathcal{F}(\mathcal{X}(\Omega), \Omega)$. 为了简洁, 我们通常会省略后一种依赖关系.

一个函数 $f: \mathcal{X}(\Omega) \rightarrow \mathcal{Y}$ 是 \mathfrak{G} -不变的 如果 $f(\rho(\mathbf{g})x) = f(x)$, 对于所有 $\mathbf{g} \in \mathfrak{G}$ 和 $x \in \mathcal{X}(\Omega)$, 也即它的输出不受在输入上群作用的影响.

注意, 信号处理的书籍中通常使用的术语“平移-不变性”指的是平移-等变性, 例如线性平移-不变性系统.

不变性的一个经典例子是平移不变性, 在计算机视觉和图像分类等模式识别应用中产生. 在这种情况下, 函数 f (通常作为卷积神经网络实现) 输入一张图像, 并输入图像包含某个特定类的对象 (例如猫或狗) 的概率. 通常合理的

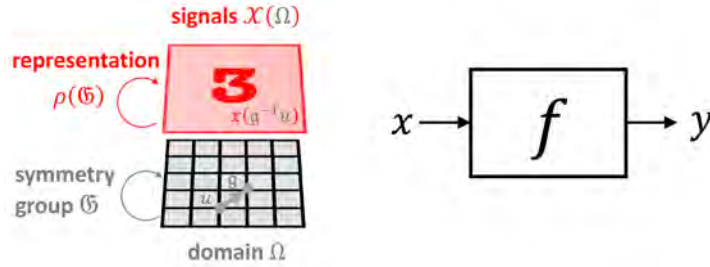


图 5: Three spaces of interest in Geometric Deep Learning: the (physical) domain Ω , the space of signals $\mathcal{X}(\Omega)$, and the hypothesis class $\mathcal{F}(\mathcal{X}(\Omega))$. Symmetries of the domain Ω (captured by the group \mathfrak{G}) act on signals $x \in \mathcal{X}(\Omega)$ through group representations $\rho(\mathfrak{g})$, imposing structure on the functions $f \in \mathcal{F}(\mathcal{X}(\Omega))$ acting on such signals.

假设是分类结果不受图像中物体位置的影响, 即函数 f 必须是平移不变的. 多层感知机可以近似任何平滑函数, 却没有这种特性—这也是上世纪 70 年代将这些架构应用于模式识别问题的早期尝试失败的原因之一. 以卷积神经网络为代表的具有局部权值共享的神经网络体系结构的发展, 除其它原因外, 是由对平移不变对象分类的需求驱动的.

如果我们仔细看看 CNN 的卷积层, 我们会发现他们不是平移-不变的, 却是平移-等变的: 换句话说, 输入到卷积层的移动会在输出特征图中产生相同数量的移动.

一个函数 $f : \mathcal{X}(\Omega) \rightarrow \mathcal{X}(\Omega)$ 是 \mathfrak{G} -等变的 如果 $f(\rho(\mathfrak{g})x) = \rho(\mathfrak{g})f(x)$ 对于所有 $\mathfrak{g} \in \mathfrak{G}$, 也就是说, 对输入的群作用会同样的影响输出.

再次借助于计算机视觉, 一个典型的需要平移-等变性的应用是图像分割, 其中 f 的输出是一个像素级的图像掩码. 显然, 分割掩码必须随着输入图像的平移而平移. 在这个例子中, 输入和输出的域是相同的, 但是因为输入有三个颜色通道, 而输出每个类有一个通道, 所以表示 $(\rho, \mathcal{X}(\Omega, \mathcal{C}))$ 和 $(\rho', \mathcal{X}(\Omega, \mathcal{C}'))$ 有些不同.

更一般地, 我们可能有 $f : \mathcal{X}(\Omega) \rightarrow \mathcal{X}(\Omega')$, 其中输入和输出空间对于相同群 \mathfrak{G} , 分别有不同的域 Ω, Ω' 和不同的表示 ρ, ρ' . 在这种情况下, 等变被定义为

$$f(\rho(\mathfrak{g})x) = \rho'(\mathfrak{g})f(x).$$

然而, 即使是以前的图像分类用例, 也实现为一系列卷积 (平移-等变) 层, 然后是全局池化层 (平移-不变的). 正如我们将在章节 3.5 中看到的, 这是大多数深度学习架构的总体蓝图, 包括 CNN 和图神经网络 (GNNs).

3.2 同构与自同构

不同对象之间的可逆和保持结构的映射通常被称为同构 (希腊语为 ‘equal shape’). 从一个对象到自身的同构称为一个自同构, 或者对称

子群与结构层次 如前所述, 对称 是一种保持某种性质或结构的变换, 对于一个给定的结构, 所有这种变换的集合形成了一个对称群. 经常发生的情况是感兴趣的结构不是一个而是多个, 因此我们可以在我们的域 Ω 上考虑多层结构. 因此, 什么是对称取决于所考虑的结构, 但在所有情况下, 对称都是遵循这个结构的可逆映射.

在最基本的层上, 域 Ω 是一个集合, 它有最少的结构: 我们所能说的是, 集合具有一定的基数. 对于有限几何, 基数是集合元素的数量 (‘大小’), 而对于无限集合, 基数表示不同种类的无穷大, 如自然数的可数无穷大, 或连续 \mathbb{R} 的不可数无穷大. 保持这种结构的自然映射是双射 (可逆映射), 我们可以把它看作集合-级别对称. 我们可以很容易地验证这是一个群通过检查以下群公理: 两个双射的组合也是一个双射 (闭合性), 结合性源于函数组合的结合性, 映射 $\tau(u) = u$ 是恒等元, 对于每个 τ , 根据定义存在逆元, 满足 $(\tau \circ \tau^{-1})(u) = (\tau^{-1} \circ \tau)(u) = u$.

根据应用场景的不同, 可能会有更多层次的结构. 例如, 如果 Ω 是一个拓扑空间, 我们可以考虑保持连续性的映射: 这种映射被称为同胚, 除了集合之间的简单双射之外, 它也是连续的, 并且具有连续的逆. 直觉上, 连续函数是良好定义的 (存在且唯一), 并将点 u 的邻域 (开放集) 中的点映射到 $\tau(u)$ 附近的邻域.

每个可微函数都是连续的. 如果映射连续可微 “足够多次”, 就称它是光滑的.

我们可以进一步要求映射和它的逆是 (连续) 可微的, 也就是说, 映射和它的逆在每一点都有导数 (而且导数也是连续的). 这进一步要求来自微分流形的微分结构, 这样的映射称为微分同胚, 并表示为 $\text{Diff}(\Omega)$. 我们将遇到的其它结构示例包括距离或度量 (度量保持映射称为等距) 或方向 (据我们所知, 方

向保持映射没有一个通用的希腊名称).

一个度量 或距离是一个函数 $d : \Omega \times \Omega \rightarrow [0, \infty)$ 满足对于所有 $u, v, w \in \Omega$:

不可区分之同一性: $d(u, v) = 0$ 当且仅当 $u = v$.

对称性: $d(u, v) = d(v, u)$.

三角不等式: $d(u, v) \leq d(u, w) + d(w, v)$.

一个带有度量的空间 (Ω, d) 称为一个度量空间.

要考虑的正确结构级别取决于问题本身. 例如, 当分割组织病理切片图像时, 我们可能希望将图像的翻转版本视为等效的 (因为在显微镜下可以翻转样本), 但如果我们试图对路标进行分类, 我们只需要将保持方向的变换视为对称性 (因为反射可能会改变路标的含义).

当我们增加需要保留的结构层级时, 对称群将会变小. 的确, 增加结构等价于选择一个子群, 它是更大的群的子集, 它满足群的公理:

记 (\mathfrak{G}, \circ) 是一个群, 并且 $\mathfrak{H} \subseteq \mathfrak{G}$ 是一个子集. 如果说 \mathfrak{H} 是群 \mathfrak{G} 的一个子群 如果 (\mathfrak{H}, \circ) 在同样的群运算下构成一个群.

例如, 欧几里得等距群 $E(2)$ 是平面微分同胚群 $\text{Diff}(2)$, 而保向等距群 $SE(2)$ 又是 $E(2)$ 的子群. 这一层次结构遵循在序言中概述的 Erlangen 纲领哲学: 在卡莱茵的构造中, 投影、仿射和欧几里得具有越来越多的不变量, 并对应着逐渐变小的群.

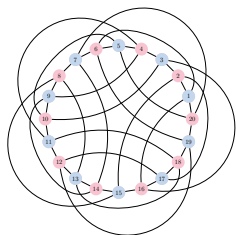
同构与自同构 我们把对称性描述为从物体到自身的结构保持和可逆映射. 这样的映射也被称为自同构, 并描述了一个对象与它本身等价的方式. 然而, 一个同样重要的映射类别称为同构, 它展示了两个不相同的对象之间的等价性. 这些概念经常被混淆在一起, 但是区分它们对于我们接下来的讨论来说

是必要的.

要理解它们之间的区别, 请考虑一个集合 $\Omega = \{0, 1, 2\}$. 集合 $\Omega = \{0, 1, 2\}$ 的一个自同构是一个双射 $\tau : \Omega \rightarrow \Omega$, 满足一个循环平移 $\tau(u) = u + 1 \pmod{3}$. 这样的映射保留了基数属性, 并将 Ω 映射到自身. 如果我们有另一个有相同数量元素的集合 $\Omega' = \{a, b, c\}$, 那么双射 $\eta : \Omega \rightarrow \Omega'$ 满足 $\eta(0) = a, \eta(1) = b, \eta(2) = c$ 是一个集合同构

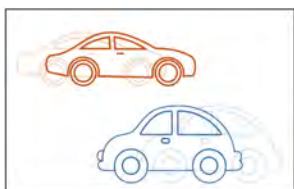
正如我们将在4.1节中看到的图, 结构的概念不仅包括节点的数量, 还包括连通性. 两个图 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ 和 $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ 之间的同构 $\eta : \mathcal{V} \rightarrow \mathcal{V}'$ 是节点之间的双射, 它将连通节点对映射为连通节点对, 同样也将非连通节点对映射为非连通节点对. 因此, 两个同构的图在结构上是相同的, 不同的只是它们的节点排序方式. 另一方面, 图的自同构或对称是映射 $\tau : \mathcal{V} \rightarrow \mathcal{V}$, 它将图的节点映射回自身, 同时保持连通性. 具有非平凡自同构 (也就是说, $\tau \neq \text{id}$) 的图呈现对称性.

也就是说, $(\eta(u), \eta(v)) \in \mathcal{V}'$
当且仅当 $(u, v) \in \mathcal{V}$.



3.3 变形稳定性

Folkman 图 (Folkman, 1967) 就是一个具有 3840 个自同构的图的漂亮例子, 它以许多对称的方式来绘制的.



在视频中以不同速度移动的两个物体定义了一个在平移群之间的变换.

在第3.1–3.2节中介绍的对称形式抓住了一个理想的世界, 在那里我们确切地知道哪些变换可以被视为对称, 并且我们想要确切地考虑这些对称. 例如, 在计算机视觉中, 我们可以假设平面平移是精确对称的. 然而, 现实世界具有噪声的, 这个模型在两个方面存在缺陷. 首先, 虽然这些简单的群提供了一种方式理解域 Ω (以及其上的信号 $\mathcal{X}(\Omega)$) 上的全局对称性的方法, 但它们不能很好地捕获局部对称性. 例如, 考虑一个由几个物体的视频场景, 每个物体都沿着自己不同的方向移动. 在随后的帧中, 产生的场景将包含大致相同的语义信息, 但没有全局平移解释从一个帧到另一个帧的转换. 在其它情况下, 例如被摄像机观察到的一个可变形的 3D 物体, 它只是非常难以描述保持物体一致的一组群变换. 这些例子说明, 在现实中, 我们更感兴趣的是一个更大的变换集合, 其中全局的、精确的不变性被局部的、不精确的不变性所代替. 在我们的讨论中, 我们将区分两种场景: 域 Ω 是固定的, 信号 $x \in \mathcal{X}(\Omega)$ 正在发生变形, 以及域 Ω 本身可能发生变形的设置.

对信号变形的稳定性 在许多应用中, 我们先验地知道信号 x 的一个小变形不应该改变 $f(x)$ 的输出, 因此很容易把这种变形看作是对称性. 例如, 我们可以把小的微分同胚 $\tau \in \text{Diff}(\Omega)$, 甚至小的双射视为对称. 然后, 小变形可以组合成大变形, 所以“小变形”不能形成一个群, 并且我们不能只对小变形要求不变性或等变性. 因为大变形实际上会改变输入的语义内容, 所以使用完整群 $\text{Diff}(\Omega)$ 作为对称群也不是一个好主意.

例如, 两个 ϵ -等距的复合是一个 2ϵ -等距, 这不符合群的闭合性.

一个更好的方法是用一个复杂度测度 $c(\tau)$ 来衡量给定一个 $\tau \in \text{Diff}(\Omega)$ 到一个给定对称子群 $\mathfrak{G} \subset \text{Diff}(\Omega)$ (也就是说, 平移) 之间的距离, 因此无论什么时候 $\tau \in \mathfrak{G}$, 都有 $c(\tau) = 0$. 我们现在可以用变形稳定性 (或近似不变性) 这样“较软”的概念来替换之前关于群作用下的精确不变性和等价性的定义:

$$\|f(\rho(\tau)x) - f(x)\| \leq Cc(\tau)\|x\|, \quad \forall x \in \mathcal{X}(\Omega) \quad (4)$$

其中 $\rho(\tau)x(u) = x(\tau^{-1}u)$ 如前述定义, 并且 C 是一个独立于信号 x 的常数. 满足上述方程的函数 $f \in \mathcal{F}(\mathcal{X}(\Omega))$ 称为几何稳定的. 我们将在下一节3.4中看到这些函数的示例.

因为对于 $\tau \in \mathfrak{G}, c(\tau) = 0$, 这个定义推广了上面定义的 \mathfrak{G} -不变性性质. 它在应用中的效用取决于引入适当的变形成本. 对于定义在连续欧几里得平面上的图像, 一个普遍的选择是 $c^2(\tau) := \int_{\Omega} \|\nabla \tau(u)\|^2 du$, 它测量了 τ 的弹性, 即它与恒定矢量场的位移有多大不同. 这种变形代价实际上是一个常被称为 *Dirichlet* 能量的标准, 可以用来量化 τ 与平移群之间的距离.

对域变形的稳定性 在很多应用中, 被变形的对象不是信号, 而是几何域 Ω 本身. 这方面典型的实例是处理图和流形的: 一个图可以在不同的时间实例中建模一个社会网络, 其中包含略微不同的社会关系 (如图所示), 或者一个流形可以建模一个经历非刚性变形的 3D 对象, 这种变形可以量化如下. 如果 \mathcal{D} 表示所有可能的可变域空间 (如所有图的空间, 或黎曼流形的空间), 则可以为 $\Omega, \tilde{\Omega} \in \mathcal{D}$ 定义一个合适的度量 (“距离”) $d(\Omega, \tilde{\Omega})$, 满足 $d(\Omega, \tilde{\Omega}) = 0$, 如果 Ω 和 $\tilde{\Omega}$ 在某种意义上是等价的; 例如, 当图是同构的时候, 图的编辑距离消失; 当两个流形是等距的时候, 带有测地线距离的黎曼流形之间的

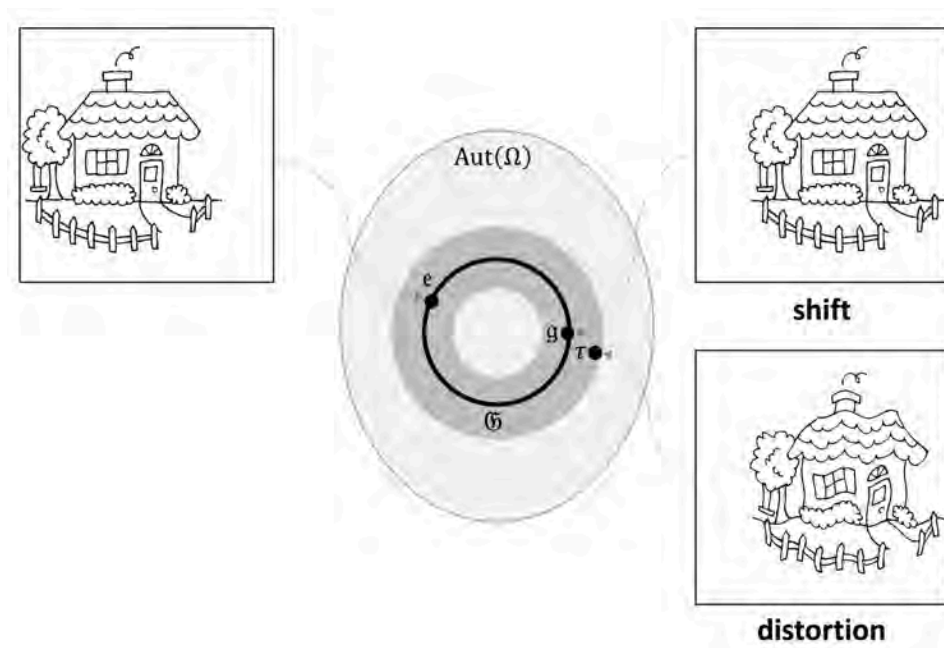


图 6: 从 Ω 到自身的所有双射的集合形成集合自同构群 $\text{Aut}(\Omega)$, 其中一个对称群 \mathfrak{G} (以圆表示) 是它的子群. 几何稳定性将 G -不变性和等变性的概念扩展到 “ G 周围的变换” (如灰色环所示), 并在变换之间的度量意义上予以量化. 在这个例子中, 图像的平滑失真接近平移.

Gromov-Hausdorff 距离消失.

这种域之间距离的常见构造依赖于可逆映射族 $\eta : \Omega \rightarrow \tilde{\Omega}$, 试图以一种方式“对齐”域, 以最好地保存相应的结构. 例如, 在图或黎曼流形 (视为具有测地线距离的度量空间) 的情况下, 这种对齐可以比较成对邻接或距离结构 (分别为 d 和 \tilde{d}),

$$d_{\mathcal{D}}(\Omega, \tilde{\Omega}) = \inf_{\eta \in \mathfrak{G}} \|d - \tilde{d} \circ (\eta \times \eta)\|$$

其中 \mathfrak{G} 是同构群, 如双射或等距, 范数定义在乘积空间 $\Omega \times \Omega$ 上. 换句话说, 元素 $\Omega, \tilde{\Omega}$ 之间的距离为被“提升”到域本身之间的距离, 这是通过考虑保持内部结构的所有可能的对齐. 给定一个信号 $x \in \mathcal{X}(\Omega)$ 和一个变形域 $\tilde{\Omega}$, 就可以考虑变形信号 $\tilde{x} = x \circ \eta^{-1} \in \mathcal{X}(\tilde{\Omega})$.

稍微滥用符号, 我们将 $\mathcal{X}(\mathcal{D}) = \{(\mathcal{X}(\Omega), \Omega) : \Omega \in \mathcal{D}\}$ 定义为在变化的域上定义的可能输入信号的集合. 函数 $f : \mathcal{X}(\mathcal{D}) \rightarrow \mathcal{Y}$ 对于域变形是稳定的如果

$$\|f(x, \Omega) - f(\tilde{x}, \tilde{\Omega})\| \leq C\|x\|d_{\mathcal{D}}(\Omega, \tilde{\Omega}) \quad (5)$$

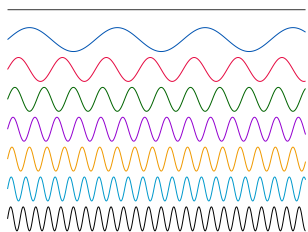
对于所有 $\Omega, \tilde{\Omega} \in \mathcal{D}$, 并且 $x \in \mathcal{X}(\Omega)$. 我们将在4.4–4.6中讨论流形的稳定性概念, 其中等距变形起着至关重要的作用. 此外, 还可以证明, 域变形的稳定性是信号变形的稳定性的自然推广, 通过查看后者的体积形式的变形Gama et al. (2019).

图编辑距离度量通过一些列图编辑操作使两个图同构的最小代价. The Gromov-Hausdorff 距离度量两个度量空间之间对应的最小可能的度量失真, 见 Gromov (1981).

两个图可以通过二次分配问题 (Quadratic Assignment Problem, QAP) 对齐, 该问题考虑了两个图的最简单形式, 两个图 G, \tilde{G} 具有相同大小 n , 并且求解 $\min_{\mathbf{P} \in \Sigma_n} \text{trace}(\mathbf{A}\mathbf{P}\mathbf{A}^{\top})$, 其中 $\mathbf{A}, \tilde{\mathbf{A}}$ 是相对应的邻接矩阵, Σ_n 是 $n \times n$ 的置换矩阵群. 图编辑距离可以与这样的 QAP 相关联 (Bougleux et al., 2015).

3.4 尺度分离

虽然变形稳定性大大加强了全局对称性先验, 但它本身不足以克服维数诅咒, 从这个意义上说, 非正式地说, 随着域的大小的增加, 仍然有“太多”的函数遵循公式 (4). 克服这一诅咒的关键是利用物理任务的多尺度结构. 在描述多尺度表示之前, 我们需要介绍傅里叶变换的主要元素, 它们依赖于频率而不是尺度.



傅里叶基函数具有全局支撑集. 因此, 局部信号产生的能量跨越所有频率.

傅利叶变换与全局不变量 可以说 最著名的信号分解是傅里叶变换, 也是调和分析的基石. 经典的一维傅里叶变换

$$\hat{x}(\xi) = \int_{-\infty}^{+\infty} x(u) e^{-i\xi u} du$$

将函数 $x(u) \in L^2(\Omega)$ 在域 $\Omega = \mathbb{R}$ 表达为一组正交基函数 $\varphi_\xi(u) = e^{i\xi u}$ 的线性组合, 以它们的振荡速率 (或者说 频率) ξ 为指标. 这样的频率组织揭示了关于信号的重要信息, 例如它的平滑性和本地化. 傅里叶基本是具有深厚的几何基础, 可以解释为域的自然振动, 与其几何结构有关 (见例子 [Berger \(2012\)](#)).

接下来, 我们将交替地使用卷积和 (互) 相关,

傅里叶变换 在信号处理中发挥着重要的作用, 因为它提供了卷积的一个对偶公式,

$$(x \star \theta)(u) = \int_{-\infty}^{+\infty} x(v) \theta(u+v) dv$$

$$(x \star \theta)(u) = \int_{-\infty}^{+\infty} x(v) \theta(u-v) dv$$

这在机器学习中很常见: 两者之间的区别在于滤波器是否被反映出来, 而由于滤波器通常是可学习的, 这一区别纯粹是符号上的.

这是线性信号滤波的标准模型 (这里和后面, x 表示信号, θ 表示滤波器. 我们将在下面展示, 卷积算子在傅里叶基中被对角化了, 这使得卷积表示为各自傅里叶变换的乘积称为可能,

$$\widehat{(x \star \theta)}(\xi) = \hat{x}(\xi) \cdot \hat{\theta}(\xi),$$

这就是信号处理中的卷积定理.

实时证明, 许多基本的微分算子, 比如拉普拉斯算子, 都被描述为欧几里得域上的卷积. 由于这种微分算子可以在非常一般的几何上进行本质上的定义, 这提供了一个正式的步骤来扩展傅里叶变换超越欧几里得域, 包括图、群和流形. 我们将在 4.4 节中对此进行详细讨论.

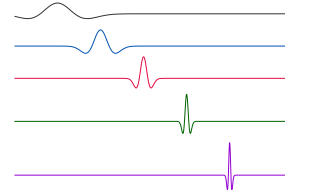
傅里叶变换的一个重要方面是它们揭示了信号和域的全局特性, 例如平滑性和传递性. 这样的全局行为在域的全局对称 (如平移) 存在时是方便的, 但不适于研究更一般的微分同态. 这需要一种权衡空间和频率本地化的表示, 就像我们接下来看到的.

多尺度表示 局部不变性的概念可以通过从基于傅里叶频率的表示转换到基于尺度的表示来表达, 这是小波等多尺度分解方法的基石. 多尺度方法的本质是将定义在域 Ω 上的函数分解为在时域和频域上都具有局部化的初等函数. 在小波的情况下, 这是通过关联一个平移和膨胀的滤波器 (母小波) ϕ 来实现的, 产生一个称为连续小波变换的空域-频域组合表示.

$$(W_\psi x)(u, \xi) = \xi^{-1/2} \int_{-\infty}^{+\infty} \psi\left(\frac{v-u}{\xi}\right) x(v) dv.$$

平移和膨胀的滤波器称为小波原子; 它们的坐标位置和膨胀对应小波变换的坐标 u 和 ξ . 这些坐标通常采用二向采样 ($\xi = 2^{-j}$ 和 $u = 2^{-j}k$), j 称为尺度因子. 多尺度信号表示在捕获全局平滑之外的规律性特性方面带来了重要的好处, 例如分段平滑, 这使其称为 90 年代信号和图像处理以及数值分析中的流形工具

参考 [Mallat \(1999\)](#) 获得全面的介绍



与傅里叶相反, 小波原子是局部的和多尺度的, 允许捕获信号的精细原子有小的空间支撑集和粗的细节原子有大的空间支撑. 术语 原子在这里与傅里叶分析中的“基元素”同义, 但要注意小波是冗余的(过完备).

多尺度表示的变形稳定性 多尺度局部小波分解比傅里叶分解的好处是考虑到小变形“附近”潜在的对称群的影响. 让我们在欧几里得定义域和平移群中说明这个重要的概念. 由于傅里叶表示对角线化了平移算子 (可以认为是卷积, 我们将在 4.2 节中看到更多细节), 因此它是平移变换的有效表示. 然而, 傅里叶分解在高频形变下是不稳定的. 相反, 小波分解在这种情况下提供了稳定的表示.

实时上, 让我们考虑 $\tau \in \text{Aut}(\Omega)$ 及其相关的线性表示 $\rho(\tau)$. 当 $\tau(u) = u - v$ 是一个平移时, 我们将在第 4.2 节中验证, 算子 $\rho(\tau) = S_v$ 是一个可以与卷积交换的平移算子. 由于卷积算子被傅里叶变换对角化了, 频域上的平移作用相当于移动傅里叶变换的复相位,

$$(\widehat{S_v x})(\xi) = e^{-i\xi v} \hat{x}(\xi).$$

因此, 去除了复相位的傅里叶系数 $f(x) = |\hat{x}|$ 是一个简单的平移-不变函数 $f(S_v x) = f(x)$, 然而, 如果我们只有近似的平移, However, if we have only approximate translation, $\tau(u) = u - \tilde{\tau}(u)$, 其中 $\|\nabla \tau\|_\infty = \sup_{u \in \Omega} \|\nabla \tilde{\tau}(u)\| \leq \epsilon$, 情况完全不同: 这是可以证明的

$$\frac{\|f(\rho(\tau)x) - f(x)\|}{\|x\|} = \mathcal{O}(1)$$

不管 ϵ 有多小 (即 τ 离称为一个平移有多近). 因此, 这种傅里叶表示在变形下是不稳定的, 不管它有多小. 这种不稳定性表现在一般域和非刚性转换中; 我们将在4.4节中使用傅里叶变换的自然扩展来分析 3D 形状时看到这种不稳定性的另一个实例.

小波为这个问题提供了一个补救方法, 它也揭示了多尺度表示的威力. 在上面的例子中, 我们可以证明 (Mallat, 2012) 小波分解 $W_\psi x$ 近似等变于变形.

这个符号意味着 $\rho(\tau)$ 作用在 $(W_\psi x)(u, \xi)$ 的空间坐标上.

$$\frac{\|\rho(\tau)(W_\psi x) - W_\psi(\rho(\tau)x)\|}{\|x\|} = \mathcal{O}(\epsilon).$$

换句话说, 使用局部滤波器而不是使用频率将信号分解成尺度, 将一个全局不稳定的表示变成了一组局部稳定的特征. 重要的是, 在不同尺度上的这种测量还不是不变的, 需要逐步处理到低频, 这暗示了现代神经网络的深层构成本质, 并在我们接下来的几何深度学习蓝图中得到表现.

尺度分离先验 我们可以从这个观点出发, 将数据域 Ω 的多尺度粗化考虑为一个层次结构 $\Omega_1, \dots, \Omega_J$. 事实证明, 这种粗化可以在非常一般的域上定义, 包括格网、图和流形. 非正式地说, 一个粗化吸收附近的点 $u, u' \in \Omega$ 在一起, 因此只需要在域内一个适当的度量概念. 如果如果 $\mathcal{X}_j(\Omega_j, \mathcal{C}_j) := \{x_j : \Omega_j \rightarrow \mathcal{C}_j\}$ 表示定义在粗化域 Ω_j 上的信号, 我们非正式地说函数 $f : \mathcal{X}(\Omega) \rightarrow \mathcal{Y}$ 是尺度 j 上是局部稳定的, 如果它承认 $f \approx f_j \circ P_j$ 的因式分解, 其中 $P_j : \mathcal{X}(\Omega) \rightarrow \mathcal{X}_j(\Omega_j)$ 是一个非线性粗化和 $f_j : \mathcal{X}_j(\Omega_j) \rightarrow \mathcal{Y}$. 换句话说, 虽然目标函数 f 可能依赖于整个域上特征之间复杂的长期相互作用, 但在局部稳定函数中, 通过首先关注向粗尺度传播的局部相互作用, 可以分离跨尺度的相互作用.

快速多极法 (FMM) 是一种数值技术, 最初发展用来加快计算 n -体问题中的长距离作用力. FMM 对靠近的源进行分组, 并将它们视为单一源.

这些原理 在物理和数学的许多领域中是非常重要的, 例如在所谓的重归一化群的统计物理中, 或在中啊哟的数值算法, 如快速多极方法中发挥了杠杆作用. 在机器学习中, 多尺度表示和局部不变性是支撑卷积神经网络和图卷积神经网络效率的基本数学原理, 通常以 (局部池化) 的形式实现. 在未来的工作中, 我们将进一步开发计算调和与分析计算, 将这些原理统一到我们的几何领域, 并将阐明尺度分离的统计学习好处.

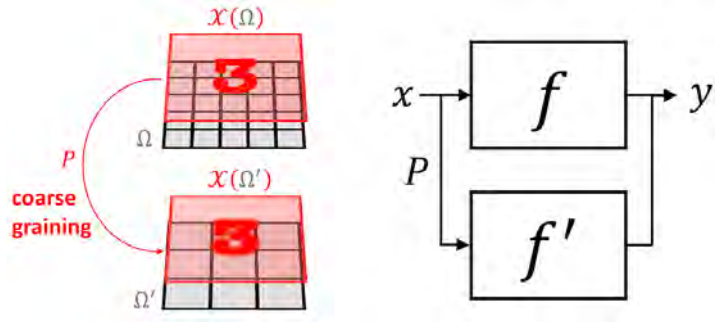


图 7: 头像分类任务的尺度分离说明. 在粗格网 $\mathcal{X}(\Omega')$ 上定义的信号分类器 f' 应该满足: $f \approx f' \circ P$, 其中 $P: \mathcal{X}(\Omega) \rightarrow \mathcal{X}(\Omega')$.

3.5 几何深度学习的蓝图

可以将 3.1-3.4 节中讨论的对称性、几何稳定性和尺度分离的几何原理结合起来, 为学习高维数据的稳定表示提供一个通用蓝图. 这些表示将由函数 f 作用于域 Ω 上的信号 $\mathcal{X}(\Omega, \mathcal{G})$ 产生, 该域 Ω 被赋予了一个对称群 \mathcal{G} .

到目前为止, 我们所描述的几何先验并没有为构建这样的表示规定一个特定的架构, 而是一系列必要的条件. 然而, 它们暗示了一个公理结构, 可证明满足这些几何先验, 同时确保一个高度表达的表示, 可以近似任何满足这些先验的目标函数.

一个简单的初步观察是, 为了获得一个高度表达的表示, 我们需要引入一个非线性元素, 因为如果 f 是线性的且 \mathcal{G} -不变的, 那么对于所有 $x \in \mathcal{X}(\Omega)$,

$$f(x) = \frac{1}{\mu(\mathcal{G})} \int_{\mathcal{G}} f(g.x) d\mu(g) = f\left(\frac{1}{\mu(\mathcal{G})} \int_{\mathcal{G}} (g.x) d\mu(g)\right),$$

此处, $\mu(g)$ 被称为群 \mathcal{G} 上的 Haar 测度, 并对整个群进行积分

这表明 F 只依赖于 x , 而仅仅通过 \mathcal{G} -平均 $Ax = \frac{1}{\mu(\mathcal{G})} \int_{\mathcal{G}} (g.x) d\mu(g)$. 在图像和翻译的情况下, 这将只需要使用输入的平均 RGB 颜色!

虽然这个推理表明线性不变量族并不是一个非常丰富的对象, 但线性等变族提供了一个更强大的工具, 因为它可以通过与适当的非线性映射组合来构建丰富而稳定的特征, 正如我们现在将要解释的那样. 事实上, 如果 B :

$\mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{X}(\Omega, \mathcal{C}')$ 是 \mathfrak{G} -等变的, 满足 $B(\mathfrak{g}.x) = \mathfrak{g}.B(x)$ 对于所有 $x \in \mathcal{X}$ 和 $\mathfrak{g} \in \mathfrak{G}$, 和 $\sigma : \mathcal{C}' \rightarrow \mathcal{C}''$ 是一个任意 (非线性) 映射, 那么我们可以简单验证组合 $U := (\sigma \circ B) : \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{X}(\Omega, \mathcal{C}'')$ 也是 \mathfrak{G} -等变的, 其中 $\sigma : \mathcal{X}(\Omega, \mathcal{C}') \rightarrow \mathcal{X}(\Omega, \mathcal{C}'')$ 是由 $(\sigma(x))(u) := \sigma(x(u))$ 给出的 σ 的实例化.

这个简单的性质允许我们通过将 U 与群平均 $A \circ U : \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{C}''$ 组合来定义一个非常一般的 \mathfrak{G} -不变量族. 因此, 一个自然的问题是, 对于 B 和 σ 的适当选择, 任何 \mathfrak{G} -不变函数是否可以用这种模型在任意精度上近似. 通过适当地将群平均推广到一般的非线性不变量, 不难从非结构化矢量输入中应用标准的通用逼近定理来证明浅层“几何”网络也是通用逼近器. 然而, 正如在傅里叶不变量和小波不变量的情况中已经描述, 在浅层全局不变量和变形稳定性之间有一个基本的张量. 这引导了另一种表示, 它考虑局部等变映射. 进一步的假设 Ω 配备了距离度量 d , 对于某个小半径 r , 我们称一个等变映射 U 是局部的, 如果对于 $\mathcal{N}_u = \{v : d(u, v) \leq r\}$, $(Ux)(u)$ 只依赖于 $x(v)$ 的值; 后一个集合 \mathcal{N}_u 被称为接受野.

有意义的度量可以在格网、图、流形和群上定义. 一个明显的例外是集合, 其中没有预定义的度量概念.

一层局部等变映射 U 不能近似具有长期相互作用的函数, 而是由几个局部等变映射 $U_J \circ U_{J-1} \cdots \circ U_1$ 的组成增加了感受野同时保持了局部等变的稳定性. 通过交错的下采样使域粗化 (再次假设一个度量结构), 感受野进一步增加, 通过多分辨率分析 (MRA, see e.g. [Mallat \(1999\)](#)) 完成并行.

“感受野”一词起源于神经科学文献, 指的是影响给定神经元输出的空间域.

综上所述, 输入域的几何结构, 利用底层对称群的知识, 提供了三个关键的构建模型: (i) 一个局部等变映射; (ii) 全局不变映射, (iii) 粗化算子. 这些构建模型提供了一个丰富的函数逼近空间与规定的不变性和稳定性的性质, 通过将它们结合在一个方案中, 我们称之为 几何深度学习蓝图 (Figure 8).

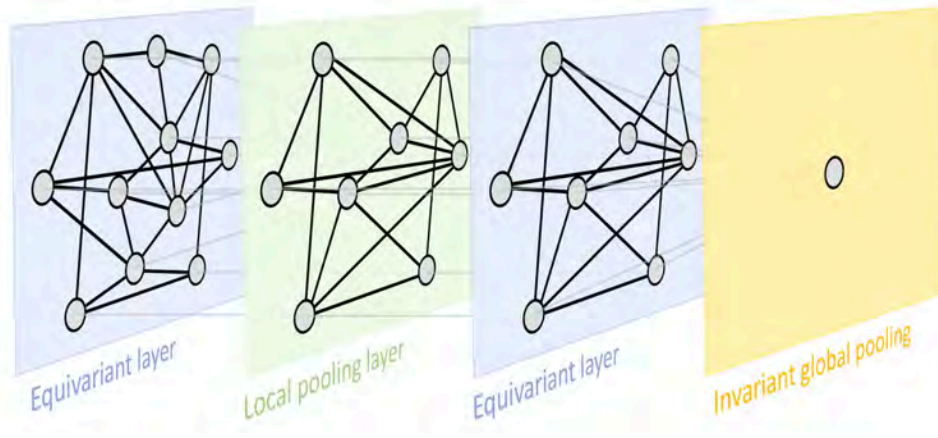


图 8: 几何深度学习蓝图, 以图为例. 一个典型的图神经网络架构可能包含置换等变层 (计算节点级特征)、局部池化层 (图粗化) 和置换不变的全局池化层 (读取层).

几何深度学习的蓝图

记 Ω 和 Ω' 是域, \mathfrak{G} 是域 Ω 上的一个对称群, 如果 Ω' 可以被认为是 Ω 的一个紧版本, 那么可以写 $\Omega' \subseteq \Omega$.

我们定义了以下构建模块:

线性 \mathfrak{G} -等变层 $B: \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{X}(\Omega', \mathcal{C}')$ 满足 $B(\mathfrak{g}.x) = \mathfrak{g}.B(x)$ 对于所有 $\mathfrak{g} \in \mathfrak{G}$ 和 $x \in \mathcal{X}(\Omega, \mathcal{C})$.

非线性性 $\sigma: \mathcal{C} \rightarrow \mathcal{C}'$ 以元素方式适用于 $(\sigma(x))(u) = \sigma(x(u))$.

局部池化 (粗化) $P: \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{X}(\Omega', \mathcal{C})$, 满足 $\Omega' \subseteq \Omega$.

\mathfrak{G} -不变层 (全局池化) $A: \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{Y}$ 满足 $A(\mathfrak{g}.x) = A(x)$ 对于所有 $\mathfrak{g} \in \mathfrak{G}$ 和 $x \in \mathcal{X}(\Omega, \mathcal{C})$.

使用这些模块可以构建 \mathfrak{G} 不变的函数 $f: \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{Y}$, 用以下形式:

$$f = A \circ \sigma_J \circ B_J \circ P_{J-1} \circ \cdots \circ P_1 \circ \sigma_1 \circ B_1$$

其中模块的选择满足每个模块的输出空间与下一个模块的输入空间匹配. 不同的模块可能利用不同的对称群选择.

几何深度学习的不同场景 当假定域 Ω 是固定的, 而只对定义在该域上的不同输入信号感兴趣时, 或者域是输入的一部分, 随着在该域上定义的信号一起变化时, 可以对不同设置进行重要的区分. 在计算机视觉应用遇到了前一种情况的一个经典实例, 其中图像被假定为在一个固定的域 (格网) 上定义. 图分类任务是后一种设定的一个例子, 图的结构和它上定义的信号 (例如节点特征) 都很重要. 在域变化的情况下, 几何稳定性 (即对 Ω 的变形不敏感) 在几何深度学习体系中起着至关重要的作用.

该蓝图具有适当的通用性, 可用于广泛的几何领域. 因此, 不同的几何深度学习方法在选择领域、对称群和上述构建模块的具体实现细节方面有所不同. 我们将在下文中看到, 目前使用的大量深度学习体系结构都属于这种方案, 因此可以从常见的几何原理推导出来.

在以下章节 (4.1–4.6) 中, 我们将描述聚焦于 “5G” 的各种几何领域中, 并在 5.1–5.8 节中介绍几何深度学习在这些领域的具体实现.

结构	域 Ω	对称群 \mathcal{G}
CNN	格网	平移
Spherical CNN	球面 / $\text{SO}(3)$	旋转 $\text{SO}(3)$
Intrinsic / Mesh CNN	流形	等距 $\text{Iso}(\Omega)$ / 规范对称 $\text{SO}(2)$
GNN	图	置换 Σ_n
Deep Sets	集合	置换 Σ_n
Transformer	完备图	置换 Σ_n
LSTM	1 维格网	时间规整

4 几何域：五个世代

我们文本的主要焦点将是图、格网、群、测地线和规范. 在这种情况下, 我们所说的“群”是指齐次空间中的全局对称变换, 流形上的“测地线”度量结构, 以及切丛 (和一般的向量丛) 上定义的“规范”局部参考系. 这些概念将在后面详细解释. 在接下来的章节中, 我们将详细讨论这些结构之间的主要共同点和主要区别特征, 并描述与之相关的对称群. 我们的阐述不是按照普遍性的顺序——事实上, 格网是图的特殊情况——而是采用突出我们的几何深度学习蓝图这一重要概念的方式.

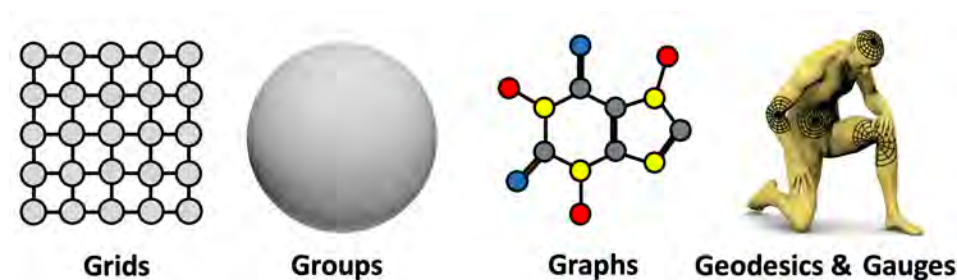


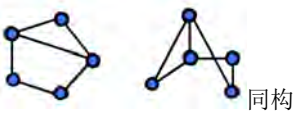
图 9: 几何深度学习的 5 个世代 (5G): 具有全局对称性的网格、群和齐次空间、图、流形上的测地线和度量以及规范 (切丛或特征空间的标架)。

4.1 图与集合

在科学的多个分支中, 从社会学到粒子物理学, 图被用作关系和相互作用系统的模型. 从我们的角度来看, 图产生了一种非常基本的不变性, 这种不变性是由置换群模拟的. 此外, 我们感兴趣的其他对象, 如格网和集合, 可以作为图的特殊例.

图 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ 是节点 \mathcal{V} 和节点间的边 $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ 构成的集合. 出于以下讨论的目的, 我们将进一步假设节点被赋予 s -维节点特征, 对于所有的 $u \in \mathcal{V}$, 表示为 \mathbf{x}_u . 社交网络可能是最常研究的图示例之一, 其中节点代表用户, 边对应于它们之间的友谊关系, 节点特征模拟用户属性, 例如年龄、简档图片

根据应用领域的不同, 节点 (Nodes) 也可以称为顶点 (vertices), 边 (edges) 通常称为链接 (links) 或关系 (relations). 我们将交替使用这些术语.



同构

是两个图之间的保边双射. 这里显示的两个同构图在节点的重新排序上是相同的.

等. 通常也可以赋予边或整个图形特征;但是由于这不会改变本节的主要发现, 我们将在后续工作中讨论.

图的关键结构性质是 \mathcal{V} 中的节点通常不假设以任何特定的顺序出现, 因此对图执行的任何操作都不应该依赖于节点的顺序. 作用于图的函数应该满足的理想性质是置换不变性, 这意味着对于任何两个同构的图, 这些函数的结果是相同的. 我们可以把这看作是我们蓝图的一个特定设置, 其中域 $\Omega = \mathcal{G}$, 空间 $\mathcal{X}(\mathcal{G}, \mathbb{R}^d)$ 是 d -维节点方向信号的空间. 我们考虑的对对称性由置换群 $\mathfrak{S} = \Sigma_n$ 给出, 置换群的元素是节点指数集合 $\{1, \dots, n\}$.

正好有 $n!$ 个这样的排列, 所以 Σ_n , 即使对于适度的 n , 也是一个非常大的群.

让我们首先说明集合上置换不变性的概念, 集合是无边图的特殊情况 (即 $\mathcal{E} = \emptyset$). 通过将节点特征堆叠为 $n \times d$ 矩阵的行 $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^\top$, 我们确实有效地指定了节点的顺序. 节点集合上的置换 $\mathbf{g} \in \Sigma_n$ 的作用相当于对 \mathbf{X} 的行进行重新排序, 这可以表示为 $n \times n$ 置换矩阵 $\rho(\mathbf{g}) = \mathbf{P}$, 其中每一行和每一列恰好包含一个 1, 而所有其他条目为零.

在这个集合上运行的函数 f 被称为置换不变的, 如果对于任何这样的置换矩阵 \mathbf{P} , 它保证了 $f(\mathbf{P}\mathbf{X}) = f(\mathbf{X})$. 一个简单的函数是

$$f(\mathbf{X}) = \phi \left(\sum_{u \in \mathcal{V}} \psi(\mathbf{x}_u) \right), \quad (6)$$

其中, 函数 ψ 独立地应用于每个节点的特征, 而 ϕ 应用于其求和聚合 (sum-aggregated) 输出: 由于求和独立于其输入提供的顺序, 这样的函数相对于节点集的置换是不变的, 因此保证总是返回相同的输出, 无论节点如何置换.

像上面这样的函数提供了一个“全局”的图形输出, 但是通常, 我们会对以节点方式“局部”工作的函数感兴趣. 例如, 我们可能想要应用一些函数来更新每个节点中的特征, 获得潜在节点特征的集合. 如果我们把这些潜在特征堆砌成矩阵 $\mathbf{H} = \mathbf{F}(\mathbf{X})$ 就不再是置换不变的. \mathbf{H} 的行的顺序应该和 \mathbf{X} 的行的顺序捆绑在一起, 这样我们就知道哪个输出节点特征对应哪个输入节点. 相反, 我们需要一个更细粒度的置换等价性概念, 一旦我们“提交”一个输入置换, 它会一致地置换结果对象. 形式上, $\mathbf{F}(\mathbf{X})$ 是置换等变函数, 如果, 对于

我们对函数 $\mathbf{F}(\mathbf{X})$ 使用粗体符号, 以强调它输出节点方向的向量特征, 因此是一个矩阵值函数.

任意置换矩阵 \mathbf{P} , 有 $\mathbf{F}(\mathbf{P}\mathbf{X}) = \mathbf{P}\mathbf{F}(\mathbf{X})$. 由权重矩阵指定的共享节点线性变换

$$\mathbf{F}(\mathbf{X}) = \mathbf{X} \quad (7)$$

是这种置换等变函数的一种可能构造, 在我们的例子中产生 $\mathbf{h}_u = \Theta^\top \mathbf{x}_u$ 形式的潜在特征.

这个构造自然来自我们的几何深度学习蓝图. 我们可以首先尝试描述线性等变函数 ($\mathbf{F}\mathbf{P}\mathbf{X} = \mathbf{P}\mathbf{F}\mathbf{X}$ 形式的函数), 对于这种函数, 很容易验证任何这样的映射都可以写成两个生成元的线性组合, 单位元 $\mathbf{F}_1\mathbf{X} = \mathbf{X}$, 平均 $\mathbf{F}_2\mathbf{X} = \frac{1}{n}\mathbf{1}\mathbf{1}^\top\mathbf{X} = \frac{1}{n}\sum_{u=1}^n \mathbf{x}_u$. 正如将在第 5.4 节中描述的, 流行的 *Deep Sets* (Zaheer et al., 2017) 架构是遵循这一蓝图.

我们现在可以将置换不变性和等变性的概念从集合推广到图. 在一般设置 $\mathcal{E} \neq \emptyset$ 中, 图的连通性可以用 $n \times n$ 邻接矩阵 \mathbf{A} 来表示, 定义为

$$a_{uv} = \begin{cases} 1 & (u, v) \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

请注意, 现在邻接矩阵和特征矩阵 \mathbf{A} 和 \mathbf{X} 是“同步的”, 也就是说, a_{uv} 指定了 \mathbf{X} 由第 u 行和第 v 行描述的节点之间的邻接信息. 因此, 将置换矩阵 \mathbf{P} 应用于节点特征 \mathbf{X} 自动意味着将其应用于 \mathbf{A} 的行和列, $\mathbf{P}\mathbf{A}\mathbf{P}^\top$. 我们说 (图式函数) 是置换不变的, 如果

$$f(\mathbf{P}\mathbf{X}, \mathbf{P}\mathbf{A}\mathbf{P}^\top) = f(\mathbf{X}, \mathbf{A}) \quad (9)$$

同时, 逐点函数是置换等变的, 如果

$$\mathbf{F}(\mathbf{P}\mathbf{X}, \mathbf{P}\mathbf{A}\mathbf{P}^\top) = \mathbf{P}\mathbf{F}(\mathbf{X}, \mathbf{A}) \quad (10)$$

对于任何置换矩阵 \mathbf{P} .

在这里, 我们可以首先描述线性等变函数的特征. 如 Maron et al. (2018) 所观察到的, 满足等式 (10) 的任何线性 \mathbf{F} 可以表示为十五个线性生成器的线性组合; 值得注意的是, 这一系列生成器独立于 u . 在这些生成器中, 我们的

当图是无向图时, 即 $(u, v) \in \mathcal{E}$ iff $(v, u) \in \mathcal{E}$, 邻接矩阵是对称的, $\mathbf{A} = \mathbf{A}^\top$.
 $\mathbf{P}\mathbf{A}\mathbf{P}^\top$ 是 Σ_n 在矩阵上的表示.

为了强调这样一个事实, 即我们在图上操作的函数现在需要考虑邻接信息, 我们使用符号 $f(\mathbf{X}, \mathbf{A})$.

这对应于贝尔数 B_4 , 它计算按 4 个元素划分一组的数量, 在这种情况下, 由索引作用于邻接矩阵的线性映射的 4 个索引 $(u, v), (u', v')$ 给出.

蓝图特别提倡那些也是本地的, 即节点 u 上的输出直接依赖于图中的相邻节点. 我们可以通过定义一个节点与另一个节点相邻意味着什么, 在我们的模型构建中显式地形式化这个约束.

通常, 节点 u 本身包含在它自己的邻居中. 节点 u 的 (无向) 邻域, 有时也称为 1 跳, 定义为

$$\mathcal{N}_u = \{v : (u, v) \in \mathcal{E} \text{ or } (v, u) \in \mathcal{E}\} \quad (11)$$

邻域的特征是多集 (*multiset*)

$$\mathbf{X}_{\mathcal{N}_u} = \{\{\mathbf{x}_v : v \in \mathcal{N}_u\}\}. \quad (12)$$

多集, 表示为 $\{\{\dots\}\}$, 是同一元素可以出现多次的集合. 这里的情况是这样的, 因为不同节点的特征可以相等.

在单跳邻居上运行很好地符合我们蓝图的局部性方面: 即, 使用 \mathcal{E} 中的边将我们的图度量定义为节点之间的最短路径距离.

因此, *GDL* 蓝图给出了一个在图上构造置换等变函数的一般方法, 通过指定一个局部函数 ϕ 来操作一个节点及其邻域 $\phi(\mathbf{x}_u, \mathbf{X}_{\mathcal{N}_u})$ 的特征. 然后, 通过将 ϕ 独立地应用于每个节点的邻域, 可以构造置换等变函数 \mathbf{F} (见图 10):

$$\mathbf{F}(\mathbf{X}, \mathbf{A}) = \begin{bmatrix} \text{---} & \phi(\mathbf{x}_1, \mathbf{X}_{\mathcal{N}_1}) & \text{---} \\ \text{---} & \phi(\mathbf{x}_2, \mathbf{X}_{\mathcal{N}_2}) & \text{---} \\ & \vdots & \\ \text{---} & \phi(\mathbf{x}_n, \mathbf{X}_{\mathcal{N}_n}) & \text{---} \end{bmatrix} \quad (13)$$

由于 \mathbf{F} 是通过将共享函数 ϕ 局部应用于每个节点来构造的, 所以它的置换等变性依赖于 ϕ 的输出, 而 ϕ 的输出与 \mathcal{N}_u 中节点的顺序无关. 因此, 如果 ϕ 被构造为置换不变的, 那么这个性质是满足的. 正如我们将在未来的工作中看到的, ϕ 的选择在这种方案的表达能力中起着至关重要的作用. 当 ϕ 是单射时, 它相当于 WeisFiler-Lehman 图同构测试的一个步骤, 这是图论中的一个经典算法, 它通过迭代颜色细化过程提供了两个图同构的一个必要条件.

同样值得注意的是, 在这个例子中, 定义在集合上的函数和更一般的图之间的区别在于, 在后一种情况下, 我们需要明确地说明域 (*domain*) 的结构. 因

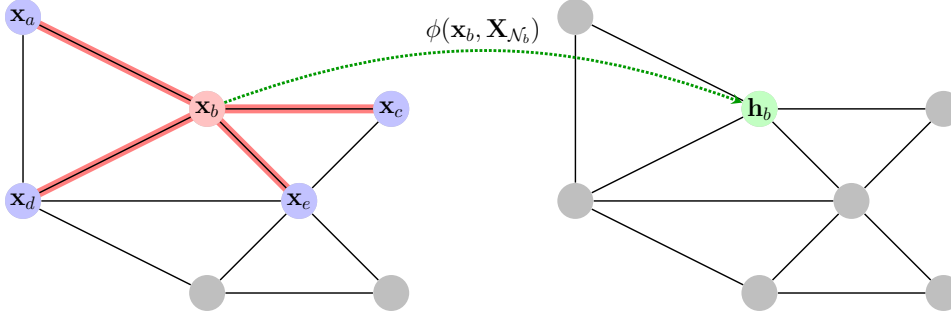


图 10: 通过将置换不变函数 ϕ 应用于每个邻域, 在图上构造置换等变函数的说明. 在这种情况下, ϕ 被应用于节点 b 的特征 \mathbf{x}_b 及其邻域特征的多集 (multiset), $\mathbf{X}_{\mathcal{N}_b} = \{\{\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c, \mathbf{x}_d, \mathbf{x}_e\}\}$. 以这种方式将 ϕ 应用于每个节点的邻域, 恢复出潜在特征矩阵的行 $\mathbf{H} = \mathbf{F}(\mathbf{X}, \mathbf{A})$.

此, 在机器学习问题中, 从域是输入的一部分这个角度来说, 图是独立的, 而当处理集合和格网 (图的两种特殊情况) 时, 我们只能指定特征并假设域是固定的. 这种区别将是我们讨论中反复出现. 因此, 几何稳定性 (对域变形的不变性) 的概念在大多数关于图的学习问题中是至关重要的. 从我们的构造可以直接得出, 置换不变和等变函数在同构 (拓扑等价) 图上产生相同的输出. 这些结果可以推广到近似同构的图, 并且有几个关于图扰动下稳定性的几个结论 (Levie et al., 2018). 在我们关于流形的讨论中, 我们将回到这个重点上, 我们将使用它作为工具来进一步详细研究这种不变性.

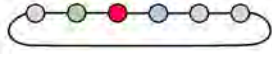
其次, 由于其额外的结构, 图和格网不同于集合, 可以以非平凡的方式进行粗化, 从而生成各种池化操作.

更准确地说, 我们不能仅仅假设集合结构来定义一个非平凡的粗化. 现有从无序集合推断拓扑结构的方法, 并且那些方法可以允许非平凡的粗化.

4.2 格网与欧几里得空间

我们考虑的第二种对象是格网. 公平地说, 深度学习对计算机视觉、自然语言处理和语音识别的影响尤其显著. 这些应用程序都有一个共同的几何特征: 包含格网结构. 如前所述, 格网是特殊邻接图的一种特殊情况. 然而, 由于网格中节点的顺序是固定的, 定义在网格上的信号的机器学习模型不再需要考

虑置换不变性, 并且具有更强的几何先验: 平移不变性.



正如我们将在后面看到的, 这使得网格成为一个齐次空间. *As we will see later, this makes the grid a homogeneous space.*

循环矩阵与卷积 让我们更详细地阐述这一点. 为了简单起见, 假设周期边界条件, 我们可以把一维格网想象成一个节点索引为 $0, 1, \dots, n-1$ 模 n (为简洁起见, 我们将省略它), 邻接矩阵的元素 $a_{u, u+1 \bmod n} = 1$, 其它为零. 与我们之前讨论的一般图相比有两个主要区别. 首先, 每个节点 u 与其邻居 $u-1$ 和 $u+1$ 具有相同的连通性, 因此在结构上无法与其他节点区分开来. 其次, 也是更重要的一点, 由于格网的节点有固定的顺序, 我们也有固定的邻居顺序: 我们可以称 $u-1$ 为“左邻居”, $u+1$ 为“右邻居”. 如果我们使用之前的方法, 使用局部聚合函数 ϕ 来设计等变函数 \mathbf{F} , 那么我们现在在格网的每个节点都有 $\mathbf{f}(\mathbf{x}_u) = \phi(\mathbf{x}_{u-1}, \mathbf{x}_u, \mathbf{x}_{u+1})$: ϕ 不再需要是置换不变的. 对于特定选择的线性变换 $\phi(\mathbf{x}_{u-1}, \mathbf{x}_u, \mathbf{x}_{u+1}) = \theta_{-1}\mathbf{x}_{u-1} + \theta_0\mathbf{x}_u + \theta_1\mathbf{x}_{u+1}$, 我们可以将 $\mathbf{F}(\mathbf{X})$ 写成矩阵乘积,

$$\mathbf{F}(\mathbf{X}) = \begin{bmatrix} \theta_0 & \theta_1 & & & \theta_{-1} \\ \theta_{-1} & \theta_0 & \theta_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \theta_{-1} & \theta_0 & \theta_1 \\ \theta_1 & & & \theta_{-1} & \theta_0 \end{bmatrix} \begin{bmatrix} \text{---} & \mathbf{x}_0 & \text{---} \\ \text{---} & \mathbf{x}_1 & \text{---} \\ & \vdots & \\ \text{---} & \mathbf{x}_{n-2} & \text{---} \\ \text{---} & \mathbf{x}_{n-1} & \text{---} \end{bmatrix}$$

请注意这种非常特殊的多对角线结构, 每个对角线上有一个元素重复, 在机器学习文献中有时称作“权重共享”.

更一般地, 给定向量 $\boldsymbol{\theta} = (\theta_0, \dots, \theta_{n-1})$, 循环矩阵 $\mathbf{C}(\boldsymbol{\theta}) = (\theta_{u-v \bmod n})$ 通过附加向量 $\boldsymbol{\theta}$ 的循环移位版本获得. 循环矩阵是离散卷积的同义词, 由于周期边界条件, 它是一个回环 (circular) 或循环 (cyclic) 卷积. 在信号处理中, $\boldsymbol{\theta}$ 通常被称为“滤波器”, 在 CNNs 中, 它的系数是可以学习的.

$$(\mathbf{x} \star \boldsymbol{\theta})_u = \sum_{v=0}^{n-1} x_{v \bmod n} \theta_{u-v \bmod n}$$

因为 $\mathbf{C}(\boldsymbol{\theta})\mathbf{x} = \mathbf{x} \star \boldsymbol{\theta}$. 一个 $\boldsymbol{\theta} = (0, 1, 0, \dots, 0)^\top$ 的特殊选择对应一个循环矩阵, 它将向量向右移动一个位置. 这个矩阵叫做 (右) 移位或平移算子, 用 \mathbf{S} 表示.

左移位算子表示为 \mathbf{S}^\top . 显然, 先左移再右移 (或者反过来) 表示没有作用, 也就是说 \mathbf{S} 是正交的: $\mathbf{S}^\top \mathbf{S} = \mathbf{S} \mathbf{S}^\top = \mathbf{I}$.

循环矩阵可以用它们的交换性 (commutativity) 来刻画：循环矩阵的乘积是可交换的，即对于任意 θ 和 η , $C(\theta)C(\eta) = C(\eta)C(\theta)$. 由于移位是循环矩阵，我们得到的卷积算子就有了常见的平移或移位等变性，

$$SC(\theta)x = C(\theta)Sx.$$

这种交换性质并不奇怪，因为基本的对称群 (平移群) 是阿贝尔群。此外，相反的方向似乎也是正确的，即一个矩阵是循环的当且仅当它与移位可交换。这反过来允许我们将卷积定义为平移等变线性运算，并且很好地说明了几何先验的力量和几何机器学习的总体哲学：卷积源于平移对称的第一原理。

请注意，与集合和图上的情况不同，线性独立的移位等变函数 (卷积) 的数量随着域的大小而增加 (因为我们在循环矩阵的每个对角线上都有一个自由度)。然而，尺度分离优先保证滤波器可以是局部的，导致每层相同的 $\Theta(1)$ -参数很复杂，正如我们将在第 5.1 节讨论的，如何使用这些原则实现卷积神经网络体系结构。

离散傅利叶变换推导 我们已经提到了傅利叶变换及其与卷积的联系：傅利叶变换将卷积运算对角化，这是信号处理中使用的一个重要特性，以便在频域中作为傅利叶变换的逐元素乘积来执行卷积。然而，教科书通常只陈述这个事实，很少解释傅立叶变换来自哪里，傅立叶基础有什么特别之处。在这里，我们可以展示它，再次证明对称的基本原则是多么的基础。

出于这个目的，回想一下线性代数中的一个事实，如果 (可对角化的) 矩阵相互交换，则它们是可联合对角化的 (jointly diagonalisable)。换句话说，所有的循环矩阵都有一个共同的特别向量，它们的区别仅在于它们的特征值。因此，我们可以选择一个循环矩阵并计算它的特征向量——我们确信这些也将是所有其他循环矩阵的特征向量。选择移位算子很方便，因为它的特征向量恰好是离散傅里叶基

$$\varphi_k = \frac{1}{\sqrt{n}} \left(1, e^{\frac{2\pi i k}{n}}, e^{\frac{4\pi i k}{n}}, \dots, e^{\frac{2\pi i (n-1)k}{n}} \right)^\top, \quad k = 0, 1, \dots, n-1,$$

我们必须另外假设不同的特征值，否则可能有多种可能的对角化。这个假设满足我们选择的 S 。

S 是正交但非对称的，因此，它的特征向量是正交的，但特征值是复数 (单位根)。

我们可以把它排列成一个 $n \times n$ 傅立叶矩阵 $\Phi = (\varphi_0, \dots, \varphi_{n-1})$ 。乘以 Φ^* 得

注意特征向量是复数，所以我们在变换 Φ 时需要取复数共轭。

到离散傅里叶变换, 乘以 Φ 得到离散傅里叶逆变换.

$$\hat{x}_k = \frac{1}{\sqrt{n}} \sum_{u=0}^{n-1} x_u e^{-\frac{2\pi i k u}{n}} \quad x_u = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} \hat{x}_k e^{+\frac{2\pi i k u}{n}}.$$

由于傅里叶变换是一个正交矩阵 ($\Phi^* \Phi = \mathbf{I}$), 它在几何上表现为坐标系的变化, 相当于 n -维旋转. 在这个坐标系 (“傅立叶域”) 中, 循环矩阵的作用变成元素积.

因为所有循环矩阵都是可联合对角化的, 所以它们也可以通过傅里叶变换对角化, 并且仅在特征值上有所不同. 由于循环矩阵 $\mathbf{C}(\theta)$ 的特征值是滤波器的傅里叶变换 (例如参见 [Bamieh \(2018\)](#)), $\hat{\theta} = \Phi^* \theta$, 我们得到卷积定理:

$$\mathbf{C}(\theta) \mathbf{x} = \Phi \begin{bmatrix} \hat{\theta}_0 & & \\ & \ddots & \\ & & \hat{\theta}_{n-1} \end{bmatrix} \Phi^* \mathbf{x} = \Phi (\hat{\theta} \odot \hat{\mathbf{x}})$$

因为傅里叶矩阵 Φ 有特殊的代数结构, 所以 $\Phi^* \mathbf{x}$ 和 $\Phi \mathbf{x}$ 的乘积可以使用快速傅立叶变换算法以 $\mathcal{O}(n \log n)$ 的复杂度计算. 这是频域滤波在信号处理中如此流行的原因之一; 此外, 滤波器通常直接在频域中设计, 因此傅里叶变换 $\hat{\theta}$ 不是显式计算的 (*explicitly computed*).

除了我们在这里所做的傅立叶变换和卷积的推导的教学价值之外, 它还提供了一个将这些概念推广到图的方案. 意识到环图 (*ring graph*) 的邻接矩阵恰好是移位算子, 可以通过计算邻接矩阵的特征向量来设计图傅立叶变换和卷积算子类比 (例如参见 [Sandryhaila and Moura \(2013\)](#)). 早期试图通过类比中枢神经系统开发图神经网络, 有时被称为 “谱图神经网络”, 利用了这一精确的蓝图. 我们将在第 4.4–4.6 节看到这个类比有一些重要的局限性. 第一个限制来自于格网是固定的这一事实, 因此格网上的所有信号都可以用相同的傅里叶基表示. 相比之下, 在一般的图形上, 傅立叶基取决于图结构. 因此, 我们不能直接比较两个不同图上的傅立叶变换——这个问题对于机器学习问题没有泛化性. 其次, 作为一维格网的张量积构造的多维格网保留了底层结构: 傅立叶基和相应的频率 (特征值) 可以在多个维度上组织. 例如, 在图像中, 我们可以自然地谈论水平和垂直频率, 过滤器有方向的概念. 在图上, 傅立叶域的结构是一维的, 因为我们只能通过相应频率的大小来组织傅立叶基函数. 因此, 图滤波器的无视方向的或各向同性的.

在图形信号处理中, 图形拉普拉斯的特征向量经常替代邻接矩阵的来构造图傅立叶变换, 参见 [Shuman et al. \(2013\)](#). 在格网上, 两个矩阵都有联合特征向量, 但在图中, 它们的结构虽然相关, 但有些不同.

连续傅里叶变换推导 为了完整起见，作为下一次讨论的铺垫，我们在连续场景中重复我们的分析。与第3.4节类似，考虑定义在 $\Omega = \mathbb{R}$ 上的函数，平移算子 $(S_v f)(u) = f(u - v)$ 将 f 移动某个位置 v 。将 S_v 应用傅立叶基函数，通过指数的关联性可得 $\varphi_\xi(u) = e^{i\xi u}$ 。

$$S_v e^{i\xi u} = e^{i\xi(u-v)} = e^{-i\xi v} e^{i\xi u},$$

即 $\varphi_{\xi}(u)$ 是 S_v 的复特征向量，复特征值 $e^{-i\xi v}$ ——正好对应了我们在离散设置中的情况。由于 S_v 是酉算子（即，对于任何 p 和 $x \in L_p(\mathbb{R})$, $\|S_v x\|_p = \|x\|_p$ ），任何特征值 λ 必须满足 $|\lambda| = 1$ ，这恰好对应于上述特征值 $e^{-i\xi v}$ 。此外，平移算子的谱很简单（simple），这意味着共享相同特征值的两个函数必须共线。实际上，假设对于某些 ξ_0 , $S_v f = e^{-i\xi_0 v} f$ 。在两边取傅立叶变换，我们得到

$$\forall \xi, e^{-i\xi v} \hat{f}(\xi) = e^{-i\xi_0 v} \hat{f}(\xi),$$

这意味着 $\xi \neq \xi_0$ 时， $\hat{f}(\xi) = 0$ ，因此 $f = \alpha \varphi_{\xi_0}$ 。

对于平移等价 ($S_v C = C S_v$) 的一般线性算子 C ，我们有

$$S_v C e^{i\xi u} = C S_v e^{i\xi u} = e^{-i\xi v} C e^{i\xi u},$$

意味着 $C e^{i\xi u}$ 也是 S_v 的特征值函数，具有特征值 $e^{-i\xi v}$ ，由此从谱的简单性得出 $C e^{i\xi u} = \beta \varphi_\xi(u)$ ；换句话说，傅立叶基是所有平移等变算子的特征基。所以， C 在傅立叶域中是对角的，可以表示为 $C e^{i\xi u} = \hat{p}_C(\xi) e^{i\xi u}$ ，其中 $\hat{p}_C(\xi)$ 是作用于不同频率 ξ 的传递函数。最后，对于任意函数 $x(u)$ ，通过线性

特征值征函数 (Eigenfunction) 与“特征向量”同义，用于指代连续算子的特征向量。

$$\begin{aligned} (Cx)(u) &= C \int_{-\infty}^{+\infty} \hat{x}(\xi) e^{i\xi u} d\xi = \int_{-\infty}^{+\infty} \hat{x}(\xi) \hat{p}_C(\xi) e^{i\xi u} d\xi \\ &= \int_{-\infty}^{+\infty} p_C(v) x(u-v) dv = (x \star p_C)(u), \end{aligned}$$

其中 $p_C(u)$ 是 $\hat{p}_C(\xi)$ 的傅里叶逆变换。因此，每个线性平移等变算子都是卷积。

平移群的谱特征是泛函分析中一个更一般的结果”斯通定理”(Stone’s Theorem) 的特殊情况，它导出了任何单参数酉群的等价特征。

4.3 群与齐次空间

在技术上, 我们需要群是局部紧的 (locally compact), 这样就存在一个左不变的哈尔测度. 关于这个测度积分, 我们可以通过任意群元素“移位”被积函数, 得到同样的结果, 就像我们对于函数 $x: \mathbb{R} \rightarrow \mathbb{R}$

我们对格网的讨论强调了移位和卷积是如何紧密相连的: 卷积是线性移位等变运算, 反之亦然, 任何移位等变线性算子都是卷积. 此外, 移位算子可以通过傅里叶变换联合对角化. 事实证明, 这是一个更大的故事的一部分: 对于任何对称群, 卷积和傅里叶变换都可以定义为求和或积分.

考虑欧几里得域 $\Omega = \mathbb{R}$. 我们可以将卷积理解为模式匹配操作: 我们将滤波器 $\theta(u)$ 的移位副本与输入信号 $x(u)$ 进行匹配. 点 u 处的卷积的值 $(x \star \theta)(u)$ 是信号 x 与经 u 移位过的滤波器的内积,

注意, 我们这里定义的 $\int_{-\infty}^{+\infty} x(u) \theta(u-v) du$ 不是卷积, 而是互相关 (cross-correlation), 在深度学习

中默认使用的名称是“卷积”. 我们这样做是为了与下面的讨论保持一致, 比如在它们的符号中

$$(\rho(\mathbf{g})x)(u) = x(u-v) \text{ and } (\rho(\mathbf{g}^{-1})x)(u) = x(u+v).$$

请注意, 在这种情况下, u 既是域 $\Omega = \mathbb{R}$ 上的一个点, 也是变换群的一个元素, 我们可以用域本身来标识 $\mathfrak{G} = \mathbb{R}$. 我们现在将展示如何推广这种构造, 通过简单地用作用于 Ω 的另一个群 \mathfrak{G} 替换变换群.

群卷积 如第 3 节所述, 群 \mathfrak{G} 在域 Ω 上的作用通过 $\rho(\mathbf{g})x(u) = x(\mathbf{g}^{-1}u)$ 在信号空间 $\mathcal{X}(\Omega)$ 上诱导出 \mathfrak{G} 的表示 ρ . 在上面的例子中, \mathfrak{G} 是平移群, 其元素通过移动坐标 $u+v$ 起作用, 而 $\rho(\mathbf{g})$ 是作为 $(S_v x)(u) = x(u-v)$ 作用于信号的移动算子. 最后, 为了对信号应用滤波器, 我们采用我们的假设, $\mathcal{X}(\Omega)$ 是一个希尔伯特空间, 带有内积

$$\langle x, \theta \rangle = \int_{\Omega} x(u) \theta(u) du,$$

积分是用 Ω 上的不变测度 μ 来完成的. 如果 μ 是离散的, 这意味着是 Ω 上的求和.

其中, 为了简单起见, 我们假设标量值信号为 $\mathcal{X}(\Omega, \mathbb{R})$; 通常内积具有等式 (2) 的形式.

既然已经定义了如何变换信号并将其与滤波器匹配, 我们就可以定义 Ω 上信号的群卷积,

$$(x \star \theta)(\mathbf{g}) = \langle x, \rho(\mathbf{g})\theta \rangle = \int_{\Omega} x(u) \theta(\mathbf{g}^{-1}u) du. \quad (14)$$

注意 $x \star \theta$ 取我们群 \mathfrak{G} 的元素 \mathbf{g} 上的值, 而不是取域 Ω 上的点. 因此, 下一层需要 $x \star \theta$ 作为输入, 应该作用于群 \mathfrak{G} 上定义的信号, 我们稍后将回到这一点.

就像传统的欧氏卷积是如何移位等变的一样, 更一般的群卷积是 \mathfrak{G} -等变的. 主要观察结果是, 用 \mathbf{g} 变换滤波器 $\rho(\mathbf{g})\theta$ 匹配信号 x 与用未变换滤波器 θ 匹配逆变换信号 $\rho(\mathbf{g}^{-1})x$ 是一样的. 从数学上讲, 这可以表示为 $\langle x, \rho(\mathbf{g})\theta \rangle = \langle \rho(\mathbf{g}^{-1})x, \theta \rangle$. 基于这一认识, 群卷积 (14) 的 \mathfrak{G} -等变紧接其定义, 群表示具有性质 $\rho(\mathbf{h}^{-1})\rho(\mathbf{g}) = \rho(\mathbf{h}^{-1}\mathbf{g})$,

$$(\rho(\mathbf{h})x \star \theta)(\mathbf{g}) = \langle \rho(\mathbf{h})x, \rho(\mathbf{g})\theta \rangle = \langle x, \rho(\mathbf{h}^{-1}\mathbf{g})\theta \rangle = \rho(\mathbf{h})(x \star \theta)(\mathbf{g}).$$

让我们看一些例子. 选择 $\Omega = \mathbb{Z}_n = \{0, \dots, n-1\}$ 和循环移位群 $\mathfrak{G} = \mathbb{Z}_n$. 在这种情况下, 群元素是指数的循环移位, 即元素 $\mathbf{g} \in \mathfrak{G}$ 可以用一些 $u = 0, \dots, n-1$ 来标识 $\mathbf{g}.v = v - u \bmod n$, 而逆元素为 $\mathbf{g}^{-1}.v = v + u \bmod n$. 重要的是, 在这个例子中, 群的元素 (移位) 也是域的元素 (索引). 因此, 我们可以通过一些简单 (*abuse*) 的符号来识别这两种结构 (即 $\Omega = \mathfrak{G}$); 在这种情况下, 我们的群卷积表达式

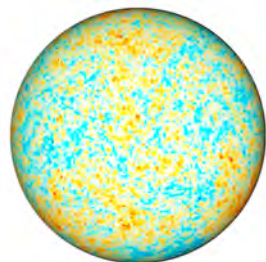
$$(x \star \theta)(\mathbf{g}) = \sum_{v=0}^{n-1} x_v \theta_{\mathbf{g}^{-1}v},$$

$$\text{导出熟悉的卷积 } (x \star \theta)_u = \sum_{v=0}^{n-1} x_v \theta_{v+u \bmod n}.$$

其实这里再说一遍, 这是互相关.

球面卷积 现在考虑带旋转群的二维球面 $\Omega = \mathbb{S}^2$, 特殊正交群 $\mathfrak{G} = \text{SO}(3)$. 虽然选择这个例子是出于教学上的原因, 但它实际上非常实用, 并在许多应用中出现. 比如在天体物理学中, 观测数据往往自然具有球面几何. 此外, 当模拟分子并试图预测其性质时, 例如为了虚拟药物筛选的目的, 球形对称性在化学应用中非常重要.

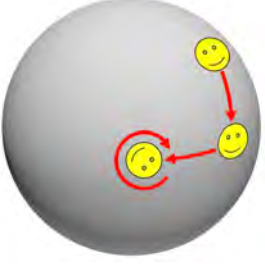
将球面上的一个点表示为三维单位向量 $\mathbf{u} : \|\mathbf{u}\| = 1$, 群的作用可以表示为 3×3 正交矩阵 $\mathbf{R}, \det(\mathbf{R}) = 1$. 球形卷积因此可以写成信号和旋转滤波器之



普朗克空间天文台捕获的宇宙微波背景辐射是 \mathbb{S}^2 上的一个信号.

间的内积,

$$(x \star \theta)(\mathbf{R}) = \int_{\mathbb{S}^2} x(\mathbf{u})\theta(\mathbf{R}^{-1}\mathbf{u})d\mathbf{u}.$$



首先要注意的是, 现在群与定义域不同: 群 $\text{SO}(3)$ 是一个李群, 它实际上是一个三维流形, 而 \mathbb{S}^2 是一个二维流形. 因此, 在这种情况下, 与前面的例子不同, 卷积是 $\text{SO}(3)$ 上的函数, 而不是 Ω 上的函数.

这具有重要的实际后果: 在我们的几何深度学习蓝图中, 我们通过将后续算子应用于前一个算子的输出来连接多个等变映射 (深度学习行话中的“层”). 在平移的情况下, 我们可以按顺序应用多个卷积, 因为它们的输出都定义在同一个域 Ω 上. 一般设定中, $x \star \theta$ 是 \mathfrak{G} 上函数, 而不是 Ω 上的函数, 我们随后不能使用完全相同的运算——这意味着下一个运算必须处理 \mathfrak{G} 上的信号, 即 $x \in \mathcal{X}(\mathfrak{G})$.

\mathfrak{G} 作用在 \mathfrak{G} 本身定义的函数上的表示称为 \mathfrak{G} 的正则表示 (regular representation)

我们对群卷积的定义允许这种情况: 我们把 $\Omega = \mathfrak{G}$ 通过 \mathfrak{G} 的复合运算定义的群作用 $(g, h) \mapsto gh$ 作为 \mathfrak{G} 自身作用的域, 这就生成了 $(\rho(g)x)(h) = x(g^{-1}h)$ 作用在 $x \in \mathcal{X}(\mathfrak{G})$ 上的表示 $\rho(g)$. 就像以前一样, 内积是通过对信号和域上滤波器的点积进行积分来定义的, 域上滤波器现在等于 $\Omega = \mathfrak{G}$. 在我们的球卷积示例中, 第二层卷积将具有以下形式

$$((x \star \theta) \star \phi)(\mathbf{R}) = \int_{\text{SO}(3)} (x \star \theta)(\mathbf{Q})\phi(\mathbf{R}^{-1}\mathbf{Q})d\mathbf{Q}.$$

由于卷积涉及内积, 而内积又需要在域 Ω 上积分, 因此我们只能在小 (离散情况下) 或低维 (连续情况下) 的域 Ω 上使用它. 例如, 我们可以在平面 \mathbb{R}^2 (二维) 或特殊正交群 $\text{SE}(3)$ (三维) 上使用卷积, 或者在图的有限节点集 (n 维) 上使用卷积, 但是实际上我们不能在具有 n 个元素的置换群 Σ_n 上执行卷积. 同样, 在更高维度的群上进行积分, 如仿射群 (包含平移、旋转、剪切和缩放, 总共 6 维), 在实践中是不可行的. 然而, 正如我们在第 5.3 节中所看到的, 我们仍然可以通过处理定义在 \mathfrak{G} 作用的低维空间 Ω 上的信号来为大群 \mathfrak{G} 建立等变卷积. 事实上, 可以证明两个域 Ω, Ω' 之间的任何等变线性映射 $f: \mathcal{X}(\Omega) \rightarrow \mathcal{X}(\Omega')$ 都可以写成类似于这里讨论的群卷积的广义卷积.

其次，我们注意到，我们在前面部分中从卷积的移位等变性质导出的傅立叶变换也可以通过将信号投影到对称群的不可约表示的矩阵元素上而扩展到更一般的情况。我们将在以后的工作中讨论这个问题。在这里研究的 $SO(3)$ 的情况下，对应球谐函数 (spherical harmonics) 和维格纳 D 函数 (Wigner D -functions)，它们在量子力学和化学中有广泛的应用。

最后，我们指出了到目前为止支撑我们在这一部分讨论的假设：无论 Ω 是格网、平面还是球体，我们都可以将每个点变换成任何其他点，直观地意味着域中的所有点“看起来都是相同的”。具有这种性质的域 Ω 称为齐次空间，其中对于任何 $u, v \in \Omega$ ，存在 $g \in \mathfrak{G}$ ，使得 $g.u = v$ 。在下一节中，我们将尝试放宽这一假设。

附加性质， $e.u = u$ 和 $g(h.u) = (gh).u$ 都是默认的。

4.4 测地线与流形

在我们的最后一个例子中，球体 S^2 是一个流形，由于它的均匀结构，它是一个具有全局对称的特殊群。不幸的是，这不是大多数流形的情况，它们通常不具有全局对称性。在这种情况下，我们不能直接定义 Ω 上信号空间上的 \mathfrak{G} 作用，并使用它来“滑动”滤波器，以便将卷积定义为经典构造的直接推广。然而，流形确实有两种不变性，我们将在本节中探讨：保留度量结构和局部参考系更改的变换。

以及旋转群 $so(3)$ ，由于它是一个李群。

虽然对许多机器学习读者来说，流形可能看起来有点像外来的物种，但实际上它们在各种科学领域都很常见。在物理学中，流形作为我们宇宙模型起着核心作用——根据爱因斯坦的广义相对论，重力来自时空的曲率，被建模为伪黎曼流形。在计算机图形学和视觉等更“平淡无奇”的领域，流形是 $3D$ 形状的常见数学模型。这种模型的应用范围很广，从虚拟和增强现实以及通过“运动捕捉”获得的特殊效果，到处理蛋白质相互作用的结构生物学，这些蛋白质相互作用像 $3D$ 拼图的碎片一样粘在一起（“绑定”在化学术语中）。这些应用的共同点是使用流形来表示一些 $3D$ 对象的边界表面。

“ $3D$ ”这个术语有些误导，指的是嵌入空间 (embedding space)。形状本身是 $2D$ 流形 (表面)。

这类模型之所以方便，有几个原因。首先，它们提供了 $3D$ 对象的简洁描述，



人是非刚性物体以近似等轴

消除了基于格网的表示中为“空白空间”分配内存的需要。其次，它们允许忽略对象的内部结构。这是一个便利的性质，例如在结构生物学中，蛋白质分子的内部折叠通常与分子表面发生的相互作用无关。第三，也是最重要的，人们经常需要处理可变形的物体 (deformable objects)，这些物体会发生非刚性变形。我们自己的身体就是这样一个例子，计算机图形学和视觉中的许多应用，比如前面提到的运动捕捉和虚拟化身，都需要变形不变性 (deformation invariance)。这种变形可以很好地模拟为保持 (黎曼) 流形固有结构的变换，即沿流形被测量的点之间的距离，而不考虑流形嵌入环境空间的方式。

我们应该强调的是，在我们的几何深度学习蓝图中，流形属于变形域 (varying domains)，在这个意义上，它类似于图。我们将强调区域变形不变性概念的重要性——我们在第 3.3 节中称之为“几何稳定性”。由于机器学习的读者可能不太熟悉微分几何，我们将介绍我们讨论所需的基本概念，并请读者参考 Penrose (2005) 的详细阐述。

我们所说的“平滑”是指可微分的次数足够，这是为了方便而默认的。这里的“变形”是指不同的同胚

(diffeomorphic)，也就是说，我们可以用一个光滑的可逆映射来映射两个邻域。

形式上，切丛是不相交并

$$\text{union}) T\Omega = \bigsqcup_{u \in \Omega} T_u \Omega.$$

对于任意非零向量 $X \neq 0$ ，如果 $g(X, X) > 0$ ，则双线性函数 g 称为正定函数。如果把 g 表示成矩阵 \mathbf{G} ，那就是 $\mathbf{G} \succ 0$ 。行列式 $|\mathbf{G}|^{1/2}$ 提供了一个局部体积元素，它不依赖于基的选择。

黎曼流形 由于流形的形式定义有些复杂，我们倾向于以牺牲精度为代价来描述。在这种情况下，我们可以把 (可微的或光滑的) 流形看作是局部欧氏的光滑多维曲面，在这种意义下，围绕它的任何一点的任何小邻域都可以变形为 \mathbb{R}^s 的邻域；在这种情况下，流形被认为是 s -维的。这使得我们可以通过正切空间 $T_u \Omega$ 局部逼近点 u 周围的流形。后者可以通过想象一个典型的二维流形，即球体，并在一个点上附加一个平面来可视化：通过足够的缩放，球面将看起来是平面的 (图 11)。所有切空间的集合称为切丛，表示为 $T\Omega$ ；我们将在第 4.5 节中详细阐述丛的概念。

一个切向量 (tangent vector)，我们用 $X \in T_u \Omega$ 表示，可以认为是从点 u 的局部位移。为了测量切向量的长度和它们之间的角度，我们需要给切空间配备附加的结构，表示为正定双线性函数 $g_u : T_u \Omega \times T_u \Omega \rightarrow \mathbb{R}$ 平滑地依赖于 u 。这样的函数被称为黎曼度量，以纪念 1856 年引入这个概念的伯恩哈特黎曼 (Bernhardt Riemann)。并且可以认为是切空间上的内积， $\langle X, Y \rangle_u = g_u(X, Y)$ 是任意两个切向量 $X, Y \in T_u \Omega$ 之间夹角的表达式。该度量还引入了一个允

许局部测量向量长度的范数 $\|X\|_u = g_u^{1/2}(X, X)$.

我们必须强调, 切向量是抽象的几何体, 它们本身就存在, 并且是无坐标的. 如果我们要把切向量 X 用数字表示为一组数字, 我们只能把它表示为一列坐标 $\mathbf{x} = (x_1, \dots, x_s)$, 相对于某个局部基 $\{X_1, \dots, X_s\} \subseteq T_u\Omega$. 类似地, 度量可以表示为一个 $s \times s$ 矩阵 \mathbf{G} , 用此基的元素 $g_{ij} = g_u(X_i, X_j)$. 我们将在第 4.5 节回到这一点.

不幸的是, 向量常常是用它们的坐标来标识的. 为了强调这个重要的区别, 我们用 X 表示切向量, 用 \mathbf{x} 表示它的坐标.

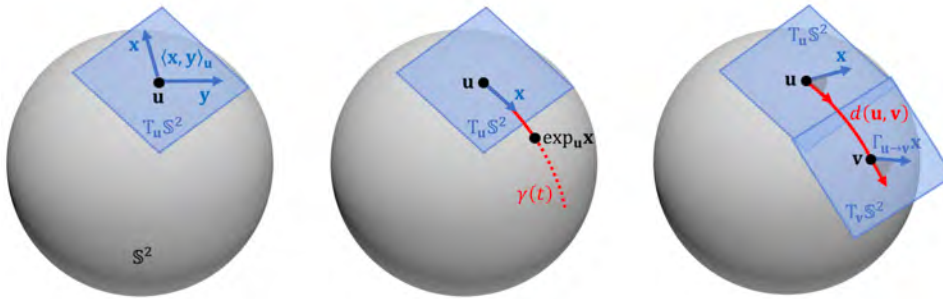


图 11: 以二维球面 $S^2 = \{\mathbf{u} \in \mathbb{R}^3 : \|\mathbf{u}\| = 1\}$, 实现了 \mathbb{R}^3 的子集 (子流形) 为例说明黎曼几何的基本概念. 球面的切线空间为 $T_u S^2 = \{\mathbf{x} \in \mathbb{R}^3 : \mathbf{x}^\top \mathbf{u} = 0\}$, 是一个 2D 平面, 因此这是一个二维流形. 黎曼度量只是对任意 $\mathbf{x}, \mathbf{y} \in T_u S^2$ 限制在切平面 $\langle \mathbf{x}, \mathbf{y} \rangle_u = \mathbf{x}^\top \mathbf{y}$ 上的欧氏内积. 指数映射由 $\exp_u(\mathbf{x}) = \cos(\|\mathbf{x}\|)\mathbf{u} + \frac{\sin(\|\mathbf{x}\|)}{\|\mathbf{x}\|}\mathbf{x}$ 给出, 对于 $\mathbf{x} \in T_u S^2$. 测地线是长度为 $d(\mathbf{u}, \mathbf{v}) = \cos^{-1}(\mathbf{u}^\top \mathbf{v})$ 的大圆弧.

配有度量的流形称为黎曼流形 (Riemannian manifold), 可以完全用度量表示的性质称为内在 (intrinsic) 性质. 这是我们讨论的一个关键概念, 因为根据我们的模板, 我们将寻求构造作用于定义在 Ω 上的信号的函数, 这些函数对于等距 (isometries) 度量保持变换是不变的, 这些变换使流形变形而不影响其局部结构. 如果这样的函数可以用内在量来表示, 它们就自动保证是等距不变的, 因此不受等距变形的影响. 这些结果可以进一步推广到处理近似等距问题; 因此, 这是我们蓝图中讨论的几何稳定性 (域变形) 的一个实例.

正如我们所指出的, 黎曼流形的定义不需要在任何空间中的几何实现, 但它表明, 任何光滑的黎曼流形都可以通过使用欧几里得空间的结构来诱导黎曼

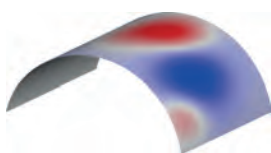


由

于 Nash (1956), 这个结果被称为嵌入定理. 折纸艺术是 \mathbb{R}^3 平面不同等距嵌入的表现 (图: Shutterstock/300 libraries)

度量, 从而实现为足够高维度的欧几里得空间的子集 (在这种情况下, 它被称为“嵌入”在该空间中). 然而, 这种嵌入不一定是唯一的——正如我们将看到的, 黎曼度量的两种不同的等距实现是可能的.

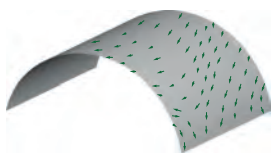
标量与矢量场 由于我们对 Ω 上定义的信号感兴趣, 我们需要提供流形上标量和向量值函数的正确概念. 一个 (光滑的) 标量场是 $x: \Omega \rightarrow \mathbb{R}$ 形式的函数, 标量场形成一个定义了内积的向量空间 $\mathcal{X}(\Omega, \mathbb{R})$



标量场示例.

$$\langle x, y \rangle = \int_{\Omega} x(u)y(u)du, \quad (15)$$

其中 du 是由黎曼度量引入的体积元素. (平滑) 切向量场是下式的函数 $X: \Omega \rightarrow T\Omega$ 在相应的切空间中为每个点指定一个切向量, $u \mapsto X(u) \in T_u\Omega$. 向量场也构成向量空间 $\mathcal{X}(\Omega, T\Omega)$, 内积通过黎曼度量定义,



$$\langle X, Y \rangle = \int_{\Omega} g_u(X(u), Y(u))du. \quad (16)$$

向量场的例子. 这些场通常被假定与流形本身具有相同的正则性类 (光滑性).

内在梯度 另一种思考 (和确切定义) 向量场的方式是作为导数的广义概念. 在经典微积分中, 人们可以通过微分 $dx(u) = x(u + du) - x(u)$ 局部线性化 (平滑) 函数, 得到函数 x 在点 u 处的值的变化, 结果是极小位移 du . 然而, 在我们的例子中, 简单使用这个定义是不可能的, 因为由于缺乏一个全局向量空间结构, “ $u + du$ ” 形式的表达式在流形上是没有意义的.

解决办法是用切向量作为局部无穷小位移的模型. 给定一个光滑标量场 $x \in \mathcal{X}(\Omega, \mathbb{R})$, 我们可以把一个 (光滑) 矢量场想象成一个线性映射 $Y: \mathcal{X}(\Omega, \mathbb{R}) \rightarrow \mathcal{X}(\Omega, \mathbb{R})$ 满足一个求导的性质: 对于任意常数 $c, Y(c) = 0$ (对应于常数函数有消失导数的意思), $Y(x + z) = Y(x) + Y(z)$ (线性), $Y(xz) = Y(x)z + xY(z)$ (积或莱布尼茨规则), 对于任意光滑标量场 $x, z \in \mathcal{X}(\Omega, \mathbb{R})$ 可以证明, 人们可以使用这些性质来定义向量场公理. 微分 $dx(Y) = Y(x)$ 可视为算子 $(u, Y) \mapsto Y(x)$ 并解释如下: 点 u 处位移 $Y \in T_u\Omega$ 导致的 x 的变化由 $d_u x(Y)$ 给出. 因此, 它是经典方向导数概念的延伸.

重要的是, 这种构造不使用黎曼度量, 因此可以扩展到 4.5 节中讨论的更一般丛的构造.

或者, 在每个点 u 处, 微分可以被视为线性泛函 $dx_u : T_u\Omega \rightarrow \mathbb{R}$ 作用于切向量 $X \in T_u\Omega$. 向量空间上的线性泛函称为对偶向量或对偶向量; 如果另外给我们一个内积 (黎曼度量), 对偶向量总是可以表示为

$$dx_u(X) = g_u(\nabla x(u), X).$$

这是里 *Riesz-Fréchet* 表示定理的一个结果, 根据这个定理, 每个对偶向量都可以表示为一个向量的内积.

点 u 处微分的表示是正切向量 $\nabla x(u) \in T_u\Omega$, 称为 x 的 (固有) 梯度; 类似于经典微积分中的梯度, 它可以被认为是 x 的最陡增加的方向. 梯度被认为是算子 $\nabla : \mathcal{X}(\Omega, \mathbb{R}) \rightarrow \mathcal{X}(\Omega, T\Omega)$ 在每个点 $x(u) \mapsto \nabla x(u) \in T_u\Omega$ 上赋值; 因此, 标量场 x 的梯度是矢量场 ∇x .

测地线 现在考虑流形上的一条光滑曲线 $\gamma : [0, T] \rightarrow \Omega$, 端点 $u = \gamma(0), v = \gamma(T)$. 曲线在点 t 处的导数是切线向量 $\gamma'(t) \in T_{\gamma(t)}\Omega$, 称为速度向量 (velocity vector). 在所有连接 u 点和 v 点的曲线中, 我们对最小长度 (minimum length) 的曲线感兴趣, 也就是说, 我们寻求 γ 来最小化长度泛函

默认情况下, 曲线以弧长参数化形式给出, 因此 $\|\gamma'\| = 1$ (恒速).

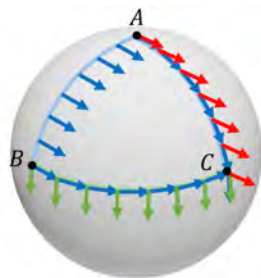
$$\ell(\gamma) = \int_0^T \|\gamma'(t)\|_{\gamma(t)} dt = \int_0^T g_{\gamma(t)}^{1/2}(\gamma'(t), \gamma'(t)) dt.$$

这种曲线被称为测地线 (来自希腊 $\gamma\epsilon\omicron\delta\alpha\iota\sigma\iota\alpha$, 字面意思是“地球的划分”), 它们在微分几何中起着重要的作用. 对我们的讨论至关重要, 我们定义测地线的方式是内在的 (intrinsic), 因为它们只依赖于黎曼度量 (通过长度泛函).

熟悉微分几何的读者可能会记得, 测地线是一个更一般的概念, 它们的定义实际上不一定需要黎曼度量, 而是一个连接 (也称为协变导数, 因为它将导数的概念推广到向量和张量场), 这是公理化定义的, 类似于我们对微分的构造. 给定一个黎曼度量, 存在一个唯一的特殊联络, 叫做 *Levi-Civita* 联络, 它在黎曼几何中经常默认有. 由这种联络产生的测地线是我们上面定义的长度最小化曲线.

Levi-Civita 联络无扭转, 符合度量要求. 黎曼几何的基本定理保证了它的存在唯一性.

接下来我们将展示如何使用测地线来定义在流形上传输切向量的方式 (平行传输), 创建从流形到切空间的局部内在映射 (指数映射), 以及定义距离 (测地线度量). 这将允许我们通过在切线空间局部应用滤波器来构造类似卷积的运算.



向量从 A 到 C 的欧氏传输在球面上没有意义, 因为得到的向量 (红色) 不在切面上. 从 A 到 C (蓝色) 的平行传输沿路径旋转矢量. 它依赖于路径: 沿着 BC 和 ABC 的路径前进会产生不同的结果.

假设流形是可定向的, 否则为 $O(s)$.

平行传输 我们在处理流形时已经遇到的一个问题是, 我们不能直接加减两点 $u, v \in \Omega$. 当试图比较不同点处的切向量时, 会出现相同的问题: 尽管它们具有相同的维数, 但它们属于不同的空间, 例如, $X \in T_u\Omega$ 和 $Y \in T_v\Omega$, 因此不能直接比较. 测地线提供了一种将向量从一个点移动到另一个点的机制, 方式如下: 设 γ 为连接点 $u = \gamma(0)$ 和 $v = \gamma(T)$ 的测地线, 设 $X \in T_u\Omega$. 我们可以沿测地线定义一组新的切向量 $X(t) \in T_{\gamma(t)}\Omega$, 使得 $X(t)$ 的长度和它与曲线的速度向量之间的角度 (通过黎曼度量表示) 是常数,

$$g_{\gamma(t)}(X(t), \gamma'(t)) = g_u(X, \gamma'(0)) = \text{const}, \quad \|X(t)\|_{\gamma(t)} = \|X\|_u = \text{const}.$$

结果, 我们在端点 v 处得到一个唯一的向量 $X(T) \in T_v\Omega$.

用上述符号定义为 $\Gamma_{u \rightarrow v}(X) = X(T)$ 的映射 $\Gamma_{u \rightarrow v}(X) : T_u\Omega \rightarrow T_v\Omega$ 和 $T_v\Omega$ 称为平行传输 (parallel transport) 或联络 (connection); 后一个术语意味着它是切线空间 $T_u\Omega$ 和 $T_v\Omega$ 之间的“连接”机制. 由于角度和长度保持条件, 平行传输只等于向量的旋转, 所以它可以与特殊正交群 $SO(s)$ 的一个元素 (称为切丛的结构群) 相关联, 我们将在第 4.5 节中用 $\mathfrak{g}_{u \rightarrow v}$ 进一步详细讨论.

正如我们前面提到的, 联络可以独立于黎曼度量进行公理化定义, 从而提供了沿着任何平滑曲线的平行传输的抽象概念. 然而, 这种传输的结果取决于所走的道路.

指数映射 局部围绕一个点 u , 在给定的方向 $X \in T_u\Omega$ 上定义唯一的测地线总是可能的, 即满足 $\gamma(0) = u, \gamma'(0) = X$, 当 $\gamma_X(t)$ 定义为所有 $t \geq 0$ 时 (也就是我们可以从一个点 u 拍摄测地线, 只要我们愿意), 流形就说是测地线完备的 (geodesically complete), 指数映射定义在整个切空间上. 由于紧流形是测地线完备的, 我们可以默认这一性质.

测地线的这种定义提供了一个点和一个方向, 给出了从正切空间 $T_u\Omega$ 到 Ω 的自然映射 (一个子集), 称为指数映射 $\exp : B_r(0) \subset T_u\Omega \rightarrow \Omega$, 它是通过沿测地线在 X 方向上走一个单位步长来定义的, 即 $\exp_u(X) = \gamma_X(1)$. 指数

请注意, 测地线的完备性并不一定保证 \exp 是全局微分同胚性—— $\exp_u(B_r(0)) \subseteq T_u\Omega$ 被微分同胚映射的关于 u 的最大半径 r 被称为单射半径 (injectivity radius).

映射 \exp_u 是一种局部微分同胚, 因为它将 $T_u\Omega$ 上原点的邻域 $B_r(0)$ (球或半径 r) 变形为邻域 u . 相反, 也可以将指数映射视为流形到切线空间的内在局部变形 (“展平”).

测地距离 一个被称为 *Hopf-Rinow* 定理的结果保证了测地线完备流形也是完备度量空间, 在该空间中, 人们可以将任意一对点 u, v 之间的距离 (称为测地距离或度量) 实现为它们之间最短路径的长度

$$d_g(u, v) = \min_{\gamma} \ell(\gamma) \quad \text{s.t.} \quad \gamma(0) = u, \gamma(T) = v,$$

这是存在的 (即, 获得了最小值).

等距算子 现在考虑我们的流形 Ω 到另一个具有黎曼度量 h 的流形 $\tilde{\Omega}$ 的变形, 我们假设它是流形之间的微分同胚 $\eta : (\Omega, g) \rightarrow (\tilde{\Omega}, h)$. 它的微分 $d\eta : T\Omega \rightarrow T\tilde{\Omega}$ 定义了各个切丛之间的映射 (称为前推), 这样在一个点 u , 我们有 $d\eta_u : T_u\Omega \rightarrow T_{\eta(u)}\tilde{\Omega}$, 解释如下: 如果我们用切向量 $X \in T_u\Omega$ 从点 u 做一个小位移, 映射 η 将被切向量 $d\eta_u(X) \in T_{\eta(u)}\tilde{\Omega}$ 从点 $\eta(u)$ 处位移.

由于前推 (*pushforward*) 提供了将两个流形上的切向量相关联的机制, 它允许将度量 h 从 $\tilde{\Omega}$ 拉回 (*pullback*) Ω ,

$$(\eta^*h)_u(X, Y) = h_{\eta(u)}(d\eta_u(X), d\eta_u(Y))$$

如果拉回度量在每一点上都与 Ω 一致, 即 $g = \eta^*h$, 则映射 η 称为 (黎曼) 等距 (*isometry*). 对于二维流形 (表面), 等距可以直观地理解为非弹性变形, 使流形变形而不 “拉伸” 或 “撕裂”.

根据它们的定义, 等距保持了固有的结构, 如测地距离, 它完全用黎曼度量来表示. 因此, 我们也可以从度量几何的位置来理解等距, 作为度量空间 $\eta : (\Omega, d_g) \rightarrow (\tilde{\Omega}, d_h)$ 之间的距离保持映射 (‘度量等距’), 在这个意义上

$$d_g(u, v) = d_h(\eta(u), \eta(v))$$

Hopf-Rinow 定理由此建立了测地线完备性和度量完备性之间的等价性, 后者意味着每个柯西序列都收敛于测地距离度量.

请注意, 术语 “度量” 有两种含义: 黎曼度量 g 和距离 d . 为了避免混淆, 我们将使用术语 “距离” 来指代后者. 我们的标记 d_g 使距离依赖于黎曼度量 g , 尽管测地线长度定义为 L .

前推和后推是伴随算子 $\langle \eta^*\alpha, X \rangle = \langle \alpha, \eta_*X \rangle$, 其中 $\alpha \in T^*\Omega$ 是对偶向量场 (dual vector field), 在每一点定义为作用于 $T_u\Omega$ 的线性泛函, 内积分别定义在向量场和对偶向量场.

对于所有的 $u, v \in \Omega$, 或者更简洁地说, $d_g = d_h \circ (\eta \times \eta)$. 换句话说, 黎曼等距也是度量等距. 在连通 (connected) 流形上, 反过来也成立: 每个度量等距也是黎曼等距.

这个结果被称为 *Myers-Steenrod* 定理. 我们默认我们的流形是连通的.

在我们的几何深度学习蓝图中, η 是区域变形的模型. 当 η 为等距时, 任何内禀 (*intrinsic*) 量都不受这种变形的影响. 人们可以通过度量扩张 (metric dilation) 的概念来推广精确 (度量) 等距

$$\text{dil}(\eta) = \sup_{u \neq v \in \Omega} \frac{d_h(\eta(u), \eta(v))}{d_g(u, v)}$$

或者通过度量失真 (metric distortion)

我们在第 3.2 节中提到的度量空间之间的

$$\text{dis}(\eta) = \sup_{u, v \in \Omega} |d_h(\eta(u), \eta(v)) - d_g(u, v)|,$$

Gromov-Hausdorff 距离可以表示为最小可能的度量失真.

它们分别捕捉 η 下测地距离的相对和绝对变化. 函数 $f \in \mathcal{F}(\mathcal{X}(\Omega))$ 在域变形下的稳定性条件 (5) 在这种情况下可以改写为

$$\|f(x, \Omega) - f(x \circ \eta^{-1}, \tilde{\Omega})\| \leq C\|x\|\text{dis}(\eta).$$

内在对称 上述的一个特殊情况是域本身的微分同胚 (我们在 3.2 节中称之为自同构), 我们将用 $\tau \in \text{Diff}(\Omega)$ 表示. 如果拉回度量满足 $\tau^*g = g$, 我们就称之为黎曼 (自) 等距, 或者如果 $d_g = d_g \circ (\tau \times \tau)$, 则称之为度量 (自) 等距. 毫不奇怪, 等距以 $\text{Iso}(\Omega)$ 表示的合成算子组成一个群, 称为等距群 (isometry group); 恒等式元素是映射 $\tau(u) = u$, 逆元素总是存在的 (根据 τ 作为微分同胚的定义). 因此, 自等距是流形的内在对称性.

流形上的连续对称是由称为 *Killing* 场的特殊切向量场无穷小生成的, 以 *Wilhelm Killing* 命名.

流形上的傅立叶分析 我们现在将展示如何在流形上构造内在的类似卷积的运算, 通过构造, 这些运算对于等距变形是不变的. 为此, 我们有两个选择: 一个是使用傅立叶变换的类比, 并将卷积定义为傅立叶域中的乘积. 另一种是通过将滤波器与信号局部相关来定义空间卷积. 让我们先讨论谱方法.

我们提醒, 在欧几里得域中, 傅里叶变换是作为循环矩阵的特征向量获得的, 由于它们的可交换性, 它们是联合可对角化的. 因此, 任何循环矩阵, 特别是

微分算子, 都可以用来定义一般域. 在黎曼几何中, 通常使用拉普拉斯算子的正交特征基, 我们将在这里定义.

为了这个目的, 回想一下我们对特征梯度算子的定义 $\nabla : \mathcal{X}(\Omega, \mathbb{R}) \rightarrow \mathcal{X}(\Omega, T\Omega)$, 产生一个切向量场, 它指示流形上标量场最陡增加的局部方向. 以类似的方式, 我们可以定义散度算子 $\nabla^* : \mathcal{X}(\Omega, T\Omega) \rightarrow \mathcal{X}(\Omega, \mathbb{R})$. 如果我们把切向量场看作流形上的一个流, 散度测量一个场在一个点上的净流量, 从而可以区分场的“源”和“汇”. 我们使用符号 ∇^* (与普通 div 相反) 来强调这两个算子是伴随的,

$$\langle X, \nabla x \rangle = \langle \nabla^* X, x \rangle,$$

其中我们使用标量场和矢量场之间的内积 (15) 和 (16).

Laplacian (在微分几何中也称为 *Laplace-Beltrami* 算子) 是定义为 $\mathcal{X}(\Omega)$ 上的算子 $\Delta = \nabla^* \nabla$, 它可以解释为一个无穷小球面上围绕一个点的函数的平均值与该点上该函数的值之间的差. 它是数学物理中最重要的算子之一, 用于描述热扩散、量子振荡和波传播等各种现象. 重要的是, 在我们的上下文中, 拉普拉斯是内在的, 因此在 Ω 的等距下是不变的.

从这个解释中也很清楚拉普拉斯是各向同性的. 我们将在第 4.6 节中看到, 定义各向异性拉普拉斯 $\nabla^*(A(u)\nabla)$ 是可能的 (参见 (Andreux et al., 2014; Boscaini et al., 2016b)), 其中 $A(u)$ 是决定局部方向的位置相关张量.

很容易看出拉普拉斯算子是自共轭的 (“对称的”),

$$\langle \nabla x, \nabla x \rangle = \langle x, \Delta x \rangle = \langle \Delta x, x \rangle.$$

上面表达式左边的二次型, 其实就是已经熟悉的狄利克雷能量

$$c^2(x) = \|\nabla x\|^2 = \langle \nabla x, \nabla x \rangle = \int_{\Omega} \|\nabla x(u)\|_u^2 du = \int_{\Omega} g_u(\nabla x(u), \nabla x(u)) du$$

度量 x 的光滑度.

拉普拉斯算子包含一个特征值分解

$$\Delta \varphi_k = \lambda_k \varphi_k, \quad k = 0, 1, \dots$$

如果流形是紧的 (这是我们默认的), 并且有正交特征函数, $\langle \varphi_k, \varphi_l \rangle = \delta_{kl}$, 由于 Δ 的自伴性, 则具有可数谱. 拉普拉斯特征基也可以被构造为一组狄利克

雷能量的正交最小值

$$\varphi_{k+1} = \arg \min_{\varphi} \|\nabla \varphi\|^2 \quad s.t. \quad \|\varphi\| = 1 \text{ and } \langle \varphi, \varphi_j \rangle = 0$$

对于 $j = 0, \dots, k$, 允许将其解释为 Ω 上最平滑的正交基. 特征函数 $\varphi_0, \varphi_1, \dots$ 相应的特征值 $0 = \lambda_0 \leq \lambda_1 \leq \dots$ 可以解释为经典傅里叶变换中原子和频率的类比.

事实上, $e^{i\xi u}$ 是欧几里得拉普拉斯 $\frac{d^2}{du^2}$ 的特征函数.

这个正交基允许将 Ω 上的平方可积函数展开成傅立叶级数

$$x(u) = \sum_{k \geq 0} \langle x, \varphi_k \rangle \varphi_k(u)$$

请注意, 该傅立叶变换具有离散指数, 因为由于 Ω 的紧性, 频谱是离散的

其中, $\hat{x}_k = \langle x, \varphi_k \rangle$ 被称为傅立叶系数或 x 的 (广义) 傅立叶变换.. 截断傅立叶级数导致的近似误差是有界的 ([Aflalo and Kimmel, 2013](#))

$$\left\| x - \sum_{k=0}^N \langle x, \varphi_k \rangle \varphi_k \right\|^2 \leq \frac{\|\nabla x\|^2}{\lambda_{N+1}}.$$

[Aflalo et al. \(2015\)](#) 进一步表明, 没有其他的基础获得更好的误差, 使拉普拉斯特征基最适合表示流形上的平滑信号.

流形上的谱卷积 谱卷积 (Spectral convolution) 可以被定义为信号 x 和滤波器 θ 的傅立叶变换的乘积,

$$(x \star \theta)(u) = \sum_{k \geq 0} (\hat{x}_k \cdot \hat{\theta}_k) \varphi_k(u). \quad (17)$$

请注意, 这里我们使用经典傅里叶变换的一个性质 (卷积定理) 作为定义非欧几里得卷积的一种方式. 由于它的构造, 谱卷积是内在的, 因此是等距不变的. 再者, 由于拉普拉斯算子是各向同性的, 所以没有方向感; 从这个意义上说, 由于邻居聚合的置换不变性, 情况类似于我们在第4.1节中对图的描述.

实际上, 由于需要对角化拉普拉斯算子, 直接计算 (17) 显得过于昂贵. 更糟糕的是, 它在几何上变得不稳定: 拉普拉斯算子的高频特征函数可能会由于



图 12: 区域扰动下谱滤波器的不稳定性. 左: 网格 Ω 上的信号 \mathbf{x} . 中间: Ω 上拉普拉斯算子 Δ 特征基的谱滤波结果. 右: 应用于几乎等距扰动域 $\tilde{\Omega}$ 的拉普拉斯 $\tilde{\Delta}$ 特征向量的相同谱滤波器产生了非常不同的结果.

域 Ω 上甚至很小的近等距扰动而发生显著变化 (见图 12). 通过将滤波器实现为形式为 $\hat{p}(\Delta)$ 的谱传递函数, 提供了更稳定的解决方案,

$$(\hat{p}(\Delta)x)(u) = \sum_{k \geq 0} \hat{p}(\lambda_k) \langle x, \varphi_k \rangle \varphi_k(u) \quad (18)$$

$$= \int_{\Omega} x(v) \sum_{k \geq 0} \hat{p}(\lambda_k) \varphi_k(v) \varphi_k(u) dv \quad (19)$$

其可以以两种方式解释: 或者作为谱滤波器 (18), 其中我们识别 $\hat{\theta}_k = \hat{p}(\lambda_k)$, 或者作为空间滤波器 (19), 其具有位置相关核 $\theta(u, v) = \sum_{k \geq 0} \hat{p}(\lambda_k) \varphi_k(v) \varphi_k(u)$.

该公式的优点在于, 可以用少量系数来参数化 $\hat{p}(\lambda)$, 并且选择诸如多项

式 $\hat{p}(\lambda) = \sum_{l=0}^r \alpha_l \lambda^l$ 这样的参数函数可以有效地计算滤波器, 如

$$(\hat{p}(\Delta)x)(u) = \sum_{k \geq 0} \sum_{l=0}^r \alpha_l \lambda_k^l \langle x, \varphi_k \rangle \varphi_k(u) = \sum_{l=0}^r \alpha_l (\Delta^l x)(u),$$

完全避免了谱分解. 我们将在第 4.6 节进一步详细讨论这种结构.

流形上的空间卷积 第二种选择是尝试在流形上定义卷积, 方法是在不同点匹配滤波器, 就像我们在公式 (14) 中所做的那样,

$$(x \star \theta)(u) = \int_{T_u \Omega} x(\exp_u Y) \theta_u(Y) dY, \quad (20)$$

我们现在必须使用指数映射从正切空间中获取标量场 x 的值, 滤波器 θ_u 在每个点定义了正切空间, 因此与位置有关. 如果定义内在的滤波器, 这种卷

基于通过傅里叶变换表达的谱卷积的几何深度学习方法通常被称为“谱”方法, 与我们以前在图设置中看到的“空间”方法相反. 我们在这里看到这两种观点可能是等价的, 所以这种二分法有些人认为, 不完全合适.

积将是等距不变的, 这是我们在许多计算机视觉和图形应用中提到的关键特性.

然而, 我们需要注意与我们之前在第4.2-4.3节中的结构的几个实质性差异. 首先, 因为流形通常不是齐次空间, 所以我们不再有全局群结构, 允许我们在一个点定义共享滤波器 (即, 在每个 u 处相同的 θ , 而不是表达式 (20) 中的 θ_u), 然后移动它. 流形上的类似操作需要平行传输, 允许在其他 $T_v\Omega$ 上应用共享 θ (定义为 $T_u\Omega$ 上的函数). 然而, 正如我们已经看到的, 这通常取决于 u 和 v 之间的路径, 所以我们移动滤波器的方式很重要. 第三, 由于我们只能在局部使用指数映射, 滤波器必须是局部的, 支持度由单射半径限定. 第四, 也是最关键的, 我们不能使用 $\theta(X)$, 因为 X 是一个抽象的几何对象: 为了让它用于计算, 我们必须相对于一些局部基 $\omega_u: \mathbb{R}^s \rightarrow T_u\Omega$ 来表示它, 作为坐标 $\mathbf{x} = \omega_u^{-1}(X)$ 的 s -维数组. 这允许我们将卷积 (20) 重写为

$$(x \star \theta)(u) = \int_{[0,1]^s} x(\exp_u(\omega_u \mathbf{y})) \theta(\mathbf{y}) d\mathbf{y}, \quad (21)$$

在单位立方体上定义过滤器. 因为指数映射是内在的 (通过测地线的定义), 所以得到的卷积是等距不变的.

然而, 这种默认的假设是我们可以将框架 ω_u 带到另一个流形, 即 $\omega'_u = d\eta_u \circ \omega_u$. 然而, 以一致的方式获得只给定一个流形 Ω 的这样一个框架 (或称规范) 是充满困难的. 首先, 光滑的全局规范可能不存在: 这是不可平行的 (parallelisable) 流形上的情况, 在这种情况下, 不能定义光滑的不消失切向量场. 第二, 我们在流形上没有正则规范, 所以这个选择是任意的; 由于我们的卷积依赖于 ω , 如果选择一个不同的规范, 我们会得到不同的结果.

球面 S^2 是一个不可平行流形的例子, 这是 *Poincaré-Hopf* 定理的结果, 通俗地说就是 “一个人要梳理一个毛球, 就必须制造一个毛球.”

我们应该注意的是, 这是一个实践偏离理论的情况: 在实践中, 有可能建立大多数是光滑的框架, 具有有限数量的奇点, 例如, 通过在流形上取一些固有标量场的固有梯度. 此外, 这样的结构是稳定的, 即这样构造的框架在等距流形上是相同的, 在近似等距流形上是相似的. 事实上, 这种方法在流形深度学习的早期工作中使用过 (*Masci et al., 2015; Monti et al., 2017*).



使

用 *Melzi et al. (2019)* 的 *GFrames* 算法, 在近似等距流形 (仅显示一个轴) 上构建稳定规范的示例.

然而, 这种解决方案并不完全令人满意, 因为在奇点附近, 滤波器方向 (以相对于规范的固定方式定义) 会有很大变化, 即使输入信号和滤波器是平滑的, 也会导致非平滑的特征图. 此外, 没有明确的理由为什么在某一点 u 的一个给定方向应该被认为等同于在完全不同的点 v 的另一个方向. 因此, 尽管有实际的替代方案, 我们接下来将寻找一个理论上更有根据的方法, 该方法将完全独立于规范的选择.

4.5 规范与丛

规范的概念, 我们已经定义为切线空间的框架, 在物理学中相当普遍: 它可以指任何向量丛的框架, 而不仅仅是切丛. 非正式地, 向量丛描述了由另一个空间参数化的向量空间族, 并且由一个基空间 (base space) Ω 组成, 该基空间 Ω 具有附着到每个位置 $u \in \Omega$ 的相同向量空间 \mathbb{V} (称为纤维) (对于切线丛, 这些是切线空间 $T_u\Omega$). 粗略来说, 一个丛在 u 处局部看起来像 $\Omega \times \mathbb{V}$ 的乘积, 但在全局可能是“扭曲的”, 并且有一个整体不同的结构. 在几何深度学习中, 可以使用纤维来模拟流形 Ω 中每个点的特征空间, 纤维的尺寸等于特征通道的数量. 在这种情况下, 一种新的迷人的对称, 称为规范对称 (gauge symmetry), 可能会出现.

历史上, 纤维丛最早出现在 *Élie Cartan* 的现代微分几何 (然而他并没有明确定义它们), 然后在 20 世纪 30 年代作为拓扑学领域的一个独立对象得到进一步发展.

让我们再考虑一个 s 维流形 Ω , 它有一个切丛 $T\Omega$, 和一个向量场 $X: \Omega \rightarrow T\Omega$ (在这个术语中称为切丛上的一个截面). 相对于切丛的规范 ω , X 表示为函数 $\mathbf{x}: \Omega \rightarrow \mathbb{R}^s$. 然而, 重要的是要认识到, 我们真正感兴趣的是底层的几何对象 (矢量场), 其表示为函数 $\mathbf{x} \in \mathcal{X}(\Omega, \mathbb{R}^s)$ 取决于规范 ω 的选择. 如果我们改变规范, 我们也需要改变 \mathbf{x} , 以便保留被表示的基础向量场.

切丛与结构群 当我们改变规范时, 我们需要在每个点应用一个可逆矩阵, 将旧规范映射到新规范. 该矩阵对于每个点上的每对仪表都是唯一的, 但在不同点上可能不同. 换句话说, 规范变换 (gauge transformation) 是一种映射 $\mathbf{g}: \Omega \rightarrow \text{GL}(s)$, 其中 $\text{GL}(s)$ 是可逆 $s \times s$ 矩阵的一般线性群. 它作用于规范上 $\omega_u: \mathbb{R}^s \rightarrow T_u\Omega$ 产生新的规范 $\omega'_u = \omega_u \circ \mathbf{g}_u: \mathbb{R}^s \rightarrow T_u\Omega$. 规范变换通

过 $\mathbf{x}'(u) = \mathbf{g}_u^{-1}\mathbf{x}(u)$ 作用于每个点的坐标矢量场, 以产生相对于新规范的 X 坐标表示 \mathbf{x}' . 基础矢量场保持不变:

$$X(u) = \omega'_u(\mathbf{x}'(u)) = \omega_u(\mathbf{g}_u\mathbf{g}_u^{-1}\mathbf{x}(u)) = \omega_u(\mathbf{x}(u)) = X(u),$$

这正是我们想要的. 更一般地说, 我们可能有一个根据 $GL(s)$ 的表示 ρ 变换的几何量的域, 例如 2-张量 (矩阵) 的域 $\mathbf{A}(u) \in \mathbb{R}^{s \times s}$, 它们像 $\mathbf{A}'(u) = \rho_2(\mathbf{g}_u^{-1})\mathbf{A}(u) = \rho_1(\mathbf{g}_u)\mathbf{A}(u)\rho_1(\mathbf{g}_u^{-1})$ 那样变换. 在这种情况下, 规范变换 \mathbf{g}_u 通过 $\rho(\mathbf{g}_u)$ 来衡量.

有时我们可能希望将注意力限制在具有特定属性的帧上, 例如正交帧、右手帧等. 不出所料, 我们对形成一个群的一组保持属性的变换感兴趣. 例如, 保持正交性的群是正交群 $O(s)$ (旋转和反射), 另外保持方向或“手性” (‘handedness’) 的群是正交群 $O(s)$ (纯旋转). 因此, 一般来说, 我们有一个群 \mathfrak{G} , 称为丛的结构群 (structure group), 规范变换是一个映射 $\mathbf{g} : \Omega \rightarrow \mathfrak{G}$. 一个关键的观察是, 在所有具有给定性质的情况下, 对于给定点的任何两个帧, 恰好存在一个与它们相关的规范变换.

我们用 s 表示基础空间 Ω 的尺寸, d 表示纤维的维度. 对于切丛, $d = s$ 是底层流形的维数. 对于 RGB 图像, $s = 2, d = 3$.

如前所述, 规范理论超越了切丛, 一般我们可以考虑一个向量空间的丛, 它的结构和维数不一定与基空间 Ω 的结构和维数相关. 例如, 彩色图像像素在 2D 格网上具有位置 $u \in \Omega = \mathbb{Z}^2$, 并且在 RGB 空间中具有值 $\mathbf{x}(u) \in \mathbb{R}^3$, 因此像素空间可以被视为具有基空间 \mathbb{Z}^2 和附着在每个点上纤维 \mathbb{R}^3 的向量丛. 习惯上, 相对于具有 R 、 G 和 B (按此顺序) 的基本向量的标尺来表示 RGB 图像, 使得图像的坐标表示看起来像 $\mathbf{x}(u) = (r(u), g(u), b(u))^T$. 但是我们同样可以在每个位置独立地置换基向量 (颜色通道), 只要我们记住在每个点使用的帧 (通道的顺序). 作为一种计算操作, 这是相当没有意义的, 但是正如我们将很快看到的, 考虑 RGB 颜色空间的规范变换在概念上是有用的, 因为它允许我们表达规范对称性——在这种情况下, 颜色之间的等价性——并使图像上定义的函数尊重这种对称性 (同等地对待每种颜色).

在这个例子中, 我们选择了 3 个颜色通道的排列作为丛的结构群 $\mathfrak{G} = \Sigma_3$. 其他选择, 如色调旋转 $\mathfrak{G} = \text{SO}(2)$ 也是可能的.

就像在流形上的矢量场的情况一样, RGB 规范变换改变图像的数字表示 (在每个像素独立地置换 RGB 值), 但不改变底层图像. 在机器学习应用中, 我们对在这样的图像上构造函数 $f \in \mathcal{F}(\mathcal{X}(\Omega))$ 感兴趣 (例如, 执行图像分类

或分割), 实现为神经网络的层. 因此, 无论出于什么原因, 如果我们要对我们的图像应用规范变换, 我们也需要改变函数 f (网络层), 以便保留它们的意义. 为简单起见, 考虑 1×1 卷积, 即取 RGB 像素 $\mathbf{x}(u) \in \mathbb{R}^3$ 特征向量 $\mathbf{y}(u) \in \mathbb{R}^C$ 的映射. 根据我们的几何深度学习蓝图, 输出与群表示 ρ_{out} 相关联, 在这种情况下是结构群 $\mathfrak{G} = \Sigma_3(RGB \text{ 通道置换})$ 的 C 维表示, 类似地, 输入与 $\rho_{\text{in}}(\mathbf{g}) = \mathbf{g}$ 相关联. 然后, 如果我们对输入应用规范变换, 我们需要将线性映射 (1×1 卷积) $f: \mathbb{R}^3 \rightarrow \mathbb{R}^C$ 更改为 $f' = \rho_{\text{out}}^{-1}(\mathbf{g}) \circ f \circ \rho_{\text{in}}(\mathbf{g})$, 以便输出特征向量 $\mathbf{y}(u) = f(\mathbf{x}(u))$ 在每个点都像 $\mathbf{y}'(u) = \rho_{\text{out}}(\mathbf{g}_u)\mathbf{y}(u)$ 一样变换. 事实上, 我们证实:

$$\mathbf{y}' = f'(\mathbf{x}') = \rho_{\text{out}}^{-1}(\mathbf{g})f(\rho_{\text{in}}(\mathbf{g})\rho_{\text{in}}^{-1}(\mathbf{g})\mathbf{x}) = \rho_{\text{out}}^{-1}(\mathbf{g})f(\mathbf{x}).$$

这里, 符号 $\rho^{-1}(\mathbf{g})$ 应理解为群表示 (矩阵) $\rho(\mathbf{g})$ 的逆.

规范对称 说我们认为规范变换是对称的, 就是说通过规范变换相关的任何两个规范都被认为是等价的. 例如, 如果我们取 $\mathfrak{G} = \text{SO}(d)$, 任何两个右手正交帧被认为是等价的, 因为我们可以通过旋转将任何这样的帧映射到任何其他这样的帧. 换句话说, 没有“向上”或“向右”等明显的局部方向. 类似地, 如果 $\mathfrak{G} = \text{O}(d)$ (正交群), 那么任何左手和右手正交帧都被认为是等价的. 在这种情况下, 也没有首选方向. 一般来说, 我们可以考虑一个群 \mathfrak{G} 和每个点 u 上的一组帧, 这样对于其中的任何两个点, 都有一个唯一的 $\mathbf{g}(u) \in \mathfrak{G}$ 将一个帧映射到另一个帧上.

在我们的几何深度学习蓝图中, 将规范变换视为对称, 我们感兴趣的是使作用于信号的函数 f 在 Ω 上定义, 并相对于规范表示, 应该与这样的变换等价. 具体来说, 这意味着如果我们对输入应用一个规范变换, 输出应该经历相同的变换 (也许通过 \mathfrak{G} 的一个不同表示). 我们之前注意到, 当我们改变规范时, 函数 f 也应该改变, 但是对于规范等变映射, 情况并非如此: 改变规范会使映射保持不变. 要看到这一点, 请再次考虑 RGB 颜色空间示例. 如果 $f \circ \rho_{\text{in}}(\mathbf{g}) = \rho_{\text{out}}(\mathbf{g}) \circ f$, 则映射 $f: \mathbb{R}^3 \rightarrow \mathbb{R}^C$ 是等变的, 但在这种情况下, 应用于 f 的规范变换没有影响: $\rho_{\text{out}}^{-1}(\mathbf{g}) \circ f \circ \rho_{\text{in}}(\mathbf{g}) = f$. 换句话说, 规范等变映射的坐标表达式独立于规范, 就像在图形的情况下, 无论输入节点如何置换, 我们都应用了相同的函数. 然而, 与图的情况和到目前为止覆盖的其他例子

不同, 规范变换不是作用于 Ω , 而是通过每个 $u \in \Omega$ 的变换 $\mathbf{g}(u) \in \mathfrak{G}$ 分别作用于每个特征向量 $\mathbf{x}(u)$.

当我们在具有更大空间支持的流形上观察滤波器时, 进一步的考虑进入了画面. 让我们首先考虑一个简单的例子, 从 s 维流形 Ω 上的标量场到标量场的映射 $f: \mathcal{X}(\Omega, \mathbb{R}) \rightarrow \mathcal{X}(\Omega, \mathbb{R})$. 与向量和其他几何量不同, 标量没有方向, 因此标量场 $x \in \mathcal{X}(\Omega, \mathbb{R})$ 对于规范变换是不变的 (它根据平凡表示进行变换 $\rho(\mathbf{g}) = 1$). 因此, 从标量场到标量场的任何线性映射都是规范等变的 (或不变的, 在这种情况下是相同的). 例如, 我们可以将 f 写成类似于 (19) 的公式, 作为与位置相关滤波器的卷积运算 $\theta: \Omega \times \Omega \rightarrow \mathbb{R}$,

$$(x \star \theta)(u) = \int_{\Omega} \theta(u, v) x(v) dv. \quad (22)$$

这意味着我们在每个点都有一个潜在的不同的滤波器 $\theta_u = \theta(u, \cdot)$, 即没有空间权重共享——仅仅只有规范对称性是不能提供的.

现在考虑一个更有趣的映射例子: 从向量场到向量场的 $f: \mathcal{X}(\Omega, T\Omega) \rightarrow \mathcal{X}(\Omega, T\Omega)$. 相对于规范, 输入和输出向量场 $X, Y \in \mathcal{X}(\Omega, T\Omega)$ 是向量值函数 $\mathbf{x}, \mathbf{y} \in \mathcal{X}(\Omega, \mathbb{R}^s)$. 这些函数之间的一般线性映射可以使用我们用于标量 (22) 的相同等式来编写, 只是用矩阵值的 $\Theta: \Omega \times \Omega \rightarrow \mathbb{R}^{s \times s}$ 来代替标量核. 矩阵 $\Theta(u, v)$ 应该将 $T_v\Omega$ 中的切向量映射到 $T_u\Omega$ 中的切向量, 但是这些点具有不同的规范 (different gauges), 我们可以任意和独立地改变它们. 也就是说, 滤波器必须满足对于所有 $u, v \in \Omega$ 的 $\Theta(u, v) = \rho^{-1}(\mathbf{g}(u))\Theta(u, v)\rho(\mathbf{g}(v))$, 其中 ρ 表示 \mathfrak{G} 对矢量的作用, 由 $s \times s$ 旋转矩阵给出. 由于 $\mathbf{g}(u)$ 和 $\mathbf{g}(v)$ 可以自由选择, 因此这是对滤波器的一个过于严格的限制.

一种更好的方法是首先通过连接将向量传输到公共切空间, 然后仅在一个点施加规范等变, 也就是, 在每个点仅单个规范变换. 代替 (22), 我们可以定义如下矢量场之间的映射,

$$(\mathbf{x} \star \Theta)(u) = \int_{\Omega} \Theta(u, v) \rho(\mathbf{g}_{v \rightarrow u}) \mathbf{x}(v) dv, \quad (23)$$

其中 $\mathbf{g}_{v \rightarrow u} \in \mathfrak{G}$ 表示沿连接这两点的测地线从 v 到 u 的平行传输; 它的表示 $\rho(\mathbf{g}_{v \rightarrow u})$ 是一个 $s \times s$ 旋转矩阵, 当向量在点之间移动时, 它旋转向量. 请

注意, 假设该测地线是唯一的, 这仅在局部成立, 因此过滤器必须具有局部支持. 在规范变换 g_u 下, 该元素变换为 $g_{u \rightarrow v} \mapsto g_u^{-1} g_{u \rightarrow v} g_v$, 场本身变换为 $x(v) \mapsto \rho(g_v)x(v)$. 如果滤波器与结构群表示 $\Theta(u, v)\rho(g_u) = \rho(g_u)\Theta(u, v)$ 可交换, 则等式 (23) 定义了规范等变卷积, 其变换如下

$$(x' \star \Theta)(u) = \rho^{-1}(g_u)(x \star \Theta)(u).$$

在上述变换下.

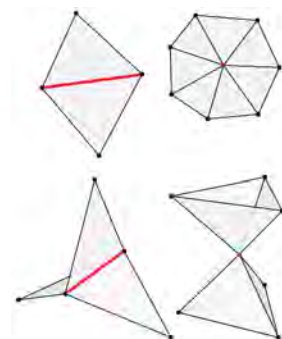
4.6 几何图与网格

我们将用几何图 (geometric graphs)(即可以在某些几何空间中实现的图形) 和网格 (meshes) 来结束我们对不同几何域的讨论. 在我们的“5G”几何领域中, 网格介于图和流形之间: 在许多方面, 它们类似于图, 但它们的附加结构也允许将它们类似于连续对象. 出于这个原因, 我们并不认为网格是我们方案中的一个独立对象, 事实上, 我们要强调的是, 我们在本节中推导的许多网格构造也直接适用于一般的图形.

正如我们在第4.4节中已经提到的, 二维流形 (表面) 是建模 3D 对象 (或者更好地说, 这种对象的边界表面) 的常见方式. 在计算机图形学和视觉应用中, 这样的表面通常被离散为三角形网格 (triangular meshes), 这可以被粗略地认为是通过将三角形沿其边缘粘合在一起而获得的表面的分段平面近似. 因此, 网格是具有附加结构的 (无向) 图: 除了节点和边之外, 网格 $\mathcal{T} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ 还具有形成三角形面 (triangular faces) 的节点的有序三元组 $\mathcal{F} = \{(u, v, q) : u, v, q \in \mathcal{V} \text{ and } (u, v), (u, q), (q, v) \in \mathcal{E}\}$; 节点的顺序定义了面的方向.

进一步假设每条边恰好由两个三角形共享, 并且入射到每个节点上的所有三角形的边界形成单个边环. 这个条件保证了每个节点周围的 1 跳邻域是圆盘状的, 因此网格构成了一个离散流形 (discrete manifold)——这种网格被称为流形网格 (manifold meshes). 类似于黎曼流形, 我们可以在网格上定义一个度量. 在最简单的情况下, 可以从网格节点 x_1, \dots, x_n , 通过边的欧几里得

三角形网格是拓扑结构的例子, 称为简单复形 (simplicial complexes).



流形 (顶部) 和非流形 (底部) 边和节点的示例. 对于有边界的流形, 可以进一步定义恰好属于一个三角形的边界边 (boundary edges).

长度表示, $\ell_{uv} = \|\mathbf{x}_u - \mathbf{x}_v\|$. 以这种方式定义的度量自动满足诸如三角形不等式 (triangle inequality) 之类的性质, 即任何 $(u, v, q) \in \mathcal{F}$ 和任何边的组合的形式满足 $\ell_{uv} \leq \ell_{uq} + \ell_{vq}$. 任何可以单独用 ℓ 表示的是内在的 (intrinsic), 网格的任何变形中保持 ℓ 的都是等距的——这些概念在我们4.4节的讨论中已经为读者所熟悉.

拉普拉斯矩阵 类似于我们对图形的处理, 让我们假设一个具有 n 个节点的 (流形) 网格, 每个节点与一个 d 维特征向量相关联, 我们可以将它排列 (假设一些任意的排序) 成一个 $n \times d$ 矩阵 \mathbf{X} . 这些特征可以表示节点的几何坐标以及其他属性, 例如颜色、法线等, 或者在特定的应用中, 例如化学, 其中几何图形模拟分子, 属性, 例如原子序数.

让我们首先看看网格上的谱卷积 (17), 我们提醒读者, 它来自拉普拉斯算子. 考虑到网格是一个底层连续曲面的离散化, 我们可以将拉普拉斯算子离散化为

$$(\Delta \mathbf{X})_u = \sum_{v \in \mathcal{N}_u} w_{uv} (\mathbf{x}_u - \mathbf{x}_v), \quad (24)$$

或者在矩阵向量表示法中, 表示为 $n \times n$ 对称矩阵 $\Delta = \mathbf{D} - \mathbf{W}$, 其中 $\mathbf{D} = \text{diag}(d_1, \dots, d_n)$ 是度矩阵 (degree matrix), $d_u = \sum_v w_{uv}$ 是节点 u 的度 (degree). 很容易看出, 等式 (24) 执行相邻特征的局部置换不变聚集 $\phi(\mathbf{x}_u, \mathbf{X}_{\mathcal{N}_u}) = d_u \mathbf{x}_u - \sum_{v \in \mathcal{N}_u} w_{uv} \mathbf{x}_v$, 并且 $\mathbf{F}(\mathbf{X}) = \Delta \mathbf{X}$ 实际上是我们在图上构造置换等变函数的一般蓝图 (13) 的实例.

请注意, 在 (24) 中, 我们对拉普拉斯的定义中没有适用于网格的内容; 事实上, 这种构造对于任意图也是有效的, 边权重用邻接矩阵 $\mathbf{W} = \mathbf{A}$ 来标识, 即, 如果 $(u, v) \in \mathcal{E}$, $w_{uv} = 1$, 否则为零. 以这种方式构建的拉普拉斯算子通常被称为组合的, 以反映它们仅仅捕获图形的连通性结构的事实. 对于几何图 (其不一定具有额外的网格结构, 但其节点具有空间坐标, 从而以边长的形式引入度量), 通常使用与度量成反比的权重, 例如 $w_{uv} \propto e^{-\ell_{uv}}$.

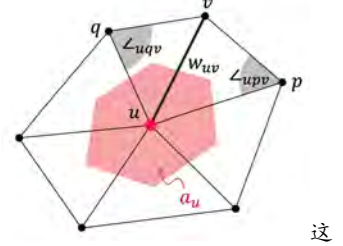
在网格上, 我们可以利用由面提供的附加结构, 并使用余切公式 (cotangent formula) 定义等式 (24) 中的边权重 (Pinkall and Polthier, 1993; Meyer

这种情况下的度数等于邻居的数量
如果图是有向的, 则对应的拉普拉斯算子是非对称的.

et al., 2003)

$$w_{uv} = \frac{\cot \angle_{uqv} + \cot \angle_{upv}}{2a_u} \quad (25)$$

其中 \angle_{uqv} 和 \angle_{upv} 是三角形 (u, q, v) 和 (u, p, v) 中与共享边 (u, v) 相对的两个角, a_u 是局部面积元素, 通常根据在共享节点 u 的三角形 (u, p, q) 的重心上构造的多边形的面积计算所得, 形式如下 $a_u = \frac{1}{3} \sum_{v, q: (u, v, q) \in \mathcal{F}} a_{uvq}$.



余切拉普拉斯算子可以被证明具有多个方便的性质 (例如, 参见 Wardetzky et al. (2007)): 它是一个正半定 (positive-semidefinite) 矩阵, $\Delta \succeq 0$, 因此具有非负特征值 $\lambda_1 \leq \dots \leq \lambda_n$, 可以看作是频率的类比, 它是对称的, 因而具有正交的特征向量, 并且是局部的 (即 $(\Delta \mathbf{X})_u$ 的值只取决于 1 跳邻居, \mathcal{N}_u). 也许最重要的性质是当网格无限细化时余切网格拉普拉斯矩阵 Δ 向连续算子 Δ 的收敛. 等式 (25) 构成了 4.4 节中定义在黎曼流形上的拉普拉斯算子的适当离散化.

虽然人们期望拉普拉斯算子是内在的, 但从等式 (25) 来看, 这并不十分明显, 而且要完全用离散度量 ℓ 来表示余切权重需要付出一些努力

$$w_{uv} = \frac{-\ell_{uv}^2 + \ell_{vq}^2 + \ell_{uq}^2}{8a_{uvq}} + \frac{-\ell_{uv}^2 + \ell_{vp}^2 + \ell_{up}^2}{8a_{uvp}}$$

其中三角形 a_{ijk} 的面积表示为

$$a_{uvq} = \sqrt{s_{uvq}(s_{uvq} - \ell_{uv})(s_{uvq} - \ell_{vq})(s_{uvq} - \ell_{uq})}$$

使用 Heron 的半周公式 $s_{uvq} = \frac{1}{2}(\ell_{uv} + \ell_{uq} + \ell_{vq})$. 这赋予拉普拉斯 (以及与其相关联的任何量, 例如其特征向量和特征值) 等距不变性 (isometry invariance), 这是几何处理和计算机图形学中非常喜欢的属性 (参见 Wang and Solomon (2019) 的优秀评论): 网格的任何不影响度量的变形 (不“拉伸”或“挤压”网格的边缘) 不会改变拉普拉斯.

最后, 正如我们已经注意到的, 拉普拉斯算子 (25) 的定义对于 \mathcal{N}_u 中节点的排列置换是不变的, 因为它涉及求和形式的聚集. 虽然在一般的图中, 由于缺乏邻居的规范排序, 这确实是一个缺点, 但是在网格上, 我们可以根据某些方向 (例如, 时钟方向) 对 1 跳邻居进行排序, 唯一的不确定性是第一个节点

这个公式的最早使用可以追溯到 MacNeal (1949) 的博士论文, 他在加州理工学院电子模拟计算机上开发了这个公式来求解偏微分方程.

为了避免三角形变得病态, 一些技术条件必须施加在精化上. 一个这样的例子是一个奇怪的三角测量圆柱体, 在德语中被称为 Schwarzscher Stiefel (Schwarz's boot), 在英语文学中被称为 “Schwarz lantern”, 由 Cauchy-Schwarz 于 1880 年提出, 他是一位德国数学家, 因 Cauchy-Schwarz 不等式而闻名.



基
于拉普拉斯的滤波器是各向同性的. 在平面上, 这种滤波器具有径向对称性.

的选择. 因此, 代替任何可能的排列置换, 我们需要考虑循环移位 (旋转), 这直观地对应于 4.5 节中讨论的 $SO(2)$ 规范变换产生的模糊性. 对于固定规范, 可以定义对局部方向敏感的各向异性拉普拉斯算子 (anisotropic Laplacian), 这相当于改变度量或权重 w_{uv} . [Andreux et al. \(2014\)](#); [Boscaini et al. \(2016b\)](#) 用这种结构来设计形状描述符; [Boscaini et al. \(2016a\)](#) 在早期网格上的几何深度学习体系结构中也使用这种结构.

谱分析与网格 正交特征向量 $\Phi = (\varphi_1, \dots, \varphi_n)$ 对角化拉普拉斯矩阵 ($\Delta = \Phi \Lambda \Phi^\top$, 其中 $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ 是拉普拉斯特征值的对角矩阵), 被用来与非欧几里得傅立叶基进行类比, 允许在网格上执行频谱卷积作为各个傅立叶变换的乘积,

$$\mathbf{X} \star \theta = \Phi \text{diag}(\Phi^\top \theta)(\Phi^\top \mathbf{X}) = \Phi \text{diag}(\hat{\theta}) \hat{\mathbf{X}},$$

其中滤波器 $\hat{\theta}$ 直接在傅立叶域中设计. 同样, 这个公式中没有什么特定于网格的, 人们可以使用一般 (无向) 图的拉普拉斯矩阵. 人们很容易利用卷积的谱定义将 CNN 泛化到图上, 这实际上是由本文作者之一 [Bruna et al. \(2013\)](#) 完成的. 然而, 非欧几里得傅立叶变换似乎对底层网格或图的微小扰动极其敏感 (见第 4.4 节中的图 12), 因此只能在必须处理固定域上的不同信号时使用, 而不能在不同域之间进行推广. 不幸的是, 许多计算机图形学和视觉问题属于后一类, 人们在一组 3D 形状 (网格) 上训练神经网络, 并在另一组上进行测试, 结论显示基于傅立叶变换的方法不合适.

如第 4.4 节所述, 最好使用形式 (18) 的谱滤波器, 对拉普拉斯矩阵应用一些传递函数 $\hat{p}(\lambda)$,

$$\hat{p}(\Delta) \mathbf{X} = \Phi \hat{p}(\Lambda) \Phi^\top \mathbf{X} = \Phi \text{diag}(\hat{p}(\lambda_1), \dots, \hat{p}(\lambda_n)) \hat{\mathbf{X}}.$$

一般情况下, 特征分解的复杂度为 $\mathcal{O}(n^3)$.

当 \hat{p} 可以用矩阵向量乘积表示时, $n \times n$ 矩阵的特征分解可以完全避免. 例如, [Defferrard et al. \(2016\)](#) 使用 r 次多项式 (polynomials) 作为滤波函数,

$$\hat{p}(\Delta) \mathbf{X} = \sum_{k=0}^r \alpha_k \Delta^k \mathbf{X} = \alpha_0 \mathbf{X} + \alpha_1 \Delta \mathbf{X} + \dots + \alpha_r \Delta^r \mathbf{X},$$

等于 $n \times d$ 特征矩阵 \mathbf{X} 乘以 $n \times n$ 拉普拉斯矩阵 r 倍。由于拉普拉斯算子通常是稀疏的 (具有 $\mathcal{O}(|\mathcal{E}|)$ 个非零元素), 因此该运算具有低复杂度的 $\mathcal{O}(|\mathcal{E}|dr) \sim \mathcal{O}(|\mathcal{E}|)$ 。此外, 由于拉普拉斯算子是局部的, 所以 r 次多项式滤波器被局部化在 r 跳邻域中。

网格是近似正则图, 每个节点有 $\mathcal{O}(1)$ 个邻居, 导致 Δ 中有 $\mathcal{O}(n)$ 个非 0。

然而, 在处理网格时, 这种精确的属性有一个缺点, 因为过滤器的实际支持 (即它覆盖的半径) 取决于网格的分辨率。我们必须记住, 网格是由一些底层连续表面的离散化产生的, 我们可以用两个不同的网格 \mathcal{T} 和 \mathcal{T}' 来表示同一个对象。在更精细的网格中, 可能必须使用比更粗糙的网格更大的邻域 (因此, 大度数 r 滤波器)。



不同分辨率网格上的两跳邻居。

由于这个原因, 在计算机图形应用中, 更常见的是使用有理滤波器 (rational filters), 因为它们与分辨率无关。有许多方法来定义这种滤波器 (例如, 参见 [Patanè \(2020\)](#)), 最常见的是作为一些有理函数的多项式, 例如, $\frac{\lambda-1}{\lambda+1}$ 。更一般地说, 可以使用复函数, 例如将实直线映射到复平面中单位圆的 *Cayley* 变换 $\frac{\lambda-i}{\lambda+i}$ 。 [Levie et al. \(2018\)](#) 使用了表示为 *Cayley* 多项式的谱滤波器, 复系数为 $\alpha_l \in \mathbb{C}$ 的实有理函数,

$$\hat{p}(\lambda) = \operatorname{Re} \left(\sum_{l=0}^r \alpha_l \left(\frac{\lambda - i}{\lambda + i} \right)^l \right).$$

Cayley 变换是 Möbius 变换的一个特例。当应用于拉普拉斯 (正半无限矩阵) 时, 它将其非负特征值映射到复半圆。

当应用于矩阵时, *Cayley* 多项式的计算需要矩阵求逆,

$$\hat{p}(\Delta) = \operatorname{Re} \left(\sum_{l=0}^r \alpha_l (\Delta - i\mathbf{I})^l (\Delta + i\mathbf{I})^{-l} \right),$$

在信号处理中, 多项式滤波器被称为有限脉冲响应 (FIR), 而有理滤波器被称为无限脉冲响应 (IIR)。

这可以近似以线性复杂度来实现。与多项式滤波器不同, 有理滤波器不具有局部支持, 而是具有指数衰减 ([Levie et al., 2018](#))。与傅里叶变换的直接计算相比, 一个关键的区别是多项式和有理滤波器在基础图形或网格的近似等距变形下是稳定的——例如 [Levie et al. \(2018, 2019\)](#); [Gama et al. \(2020\)](#); [Kenlay et al. \(2021\)](#) 展示了各种这类结果。

风格当作算子和函数映射 函数映射的范例建议将网格视为算子。正如我们将展示的, 这允许利用网格的附加结构获得更有趣的不变性类型。出于我们

讨论的目的, 假设网格 \mathcal{T} 是在具有坐标 \mathbf{X} 的嵌入节点上构建的. 如果我们构建像拉普拉斯算子这样的内在算子, 可以表明它完全编码网格的结构, 并且可以恢复网格 (直到其等距嵌入, 如 [Zeng et al. \(2012\)](#) 所示). 其他一些算子也是如此 (例如, 见 [Boscaini et al. \(2015\)](#); [Corman et al. \(2017\)](#); [Chern et al. \(2018\)](#)), 所以我们将假设一个一般的算子, 或 $n \times n$ 矩阵 $\mathbf{Q}(\mathcal{T}, \mathbf{X})$, 作为我们网格的表示.

在这种观点下, 第4.1节关于 $f(\mathbf{X}, \mathcal{T})$ 形式的学习函数的讨论可以重新表述为 $f(\mathbf{Q})$ 形式的学习函数. 类似于图和集合, 网格的节点也没有正则序, 即网格上的函数必须满足置换不变性或等变性条件,

$$\begin{aligned} f(\mathbf{Q}) &= f(\mathbf{PQP}^\top) \\ \mathbf{PF}(\mathbf{Q}) &= \mathbf{F}(\mathbf{PQP}^\top) \end{aligned}$$

对于任何置换矩阵 \mathbf{P} . 然而, 与一般的图相比, 我们现在有了更多的结构: 我们可以假设我们的网格来自于一些底层连续表面 Ω 的离散化. 因此, 可以具有不同的网格 $\mathcal{T}' = (\mathcal{V}', \mathcal{E}', \mathcal{F}')$, 其中 n' 个节点和坐标 \mathbf{X}' 表示与 \mathcal{T} 相同的对象 Ω . 重要的是, 网格 \mathcal{T} 和 \mathcal{T}' 可以具有不同的连接结构, 甚至不同数量的节点 ($n' \neq n$). 因此, 我们不能把这些网格看作仅仅是节点重排的同构图, 而把置换矩阵 \mathbf{P} 看作它们之间的对应关系.

[Ovsjanikov et al. \(2012\)](#) 引入了函数映射, 作为对应于这种设置的概念的概括, 用函数之间的对应关系 (映射 $\mathbf{C}: \mathcal{X}(\Omega) \rightarrow \mathcal{X}(\Omega')$, 见图13) 代替了两个域上的点之间的对应关系 (映射 $\eta: \Omega \rightarrow \Omega'$). 函数映射是线性算子 \mathbf{C} , 表示为矩阵 $n' \times n$, 在各个域上的信号 \mathbf{x}' 和 \mathbf{x} 之间建立对应关系

$$\mathbf{x}' = \mathbf{C}\mathbf{x}.$$

在大多数情况下, 函数映射在谱域中实现, 作为傅立叶系数

之间 $k \times k$ 到 $\hat{\mathbf{C}}$ 的映射 $\mathbf{x}' = \Phi' \hat{\mathbf{C}} \Phi^\top \mathbf{x}$, 其中 Φ 和 Φ' 分别是 (截断的) 拉普拉斯特征基的 $n \times k$ 和 $n' \times k$

[Rustamov et al. \(2013\)](#) 表明, 为了保证等积 (area-preserving) 映射, 函数映射必须是正交的, $\mathbf{C}^\top \mathbf{C} = \mathbf{I}$, 即是正交群 $\mathbf{C} \in \mathbf{O}(n)$ 的一个元素. 在这种情况下, 我们可以使用 $\mathbf{C}^{-1} = \mathbf{C}^\top$ 来反转映射.

矩阵, 满足 $k \ll n, n'$.

函数映射还建立了网格的算子表示之间的关系,

$$\mathbf{Q}' = \mathbf{CQC}^\top, \quad \mathbf{Q} = \mathbf{C}^\top \mathbf{Q}' \mathbf{C},$$

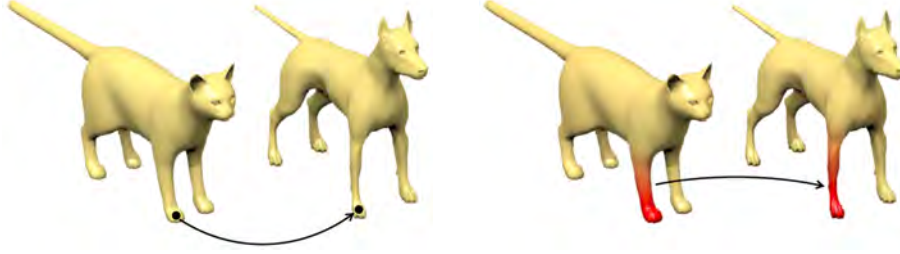


图 13: Pointwise map (left) vs functional map (right).

我们可以解释如下：给定 \mathcal{T} 的算子表示 \mathbf{Q} 和函数映射 \mathbf{C} ，我们可以通过首先将信号从 \mathcal{T}' 映射到 \mathcal{T} (使用 \mathbf{C}^\top)，应用算子 \mathbf{Q} ，然后映射回 \mathcal{T}' (使用 \mathbf{C}) 来构造 \mathcal{T}' 的表示 \mathbf{Q}' 。这导致我们在网格上重新划分不变 (remeshing invariant) (或等变) 函数的更一般的类别，满足

请注意，我们从右向左阅读这些操作。

$$\begin{aligned} f(\mathbf{Q}) &= f(\mathbf{CQC}^\top) = f(\mathbf{Q}') \\ \mathbf{CF}(\mathbf{Q}) &= \mathbf{F}(\mathbf{CQC}^\top) = \mathbf{F}(\mathbf{Q}') \end{aligned}$$

对于任意 $\mathbf{C} \in \mathbf{O}(n)$ 。很容易看出，前面的排列置换不变性和等变性的设置是一个特殊的情况，可以认为是一个微不足道的重新网格划分，其中只有节点的顺序被改变。

这是由置换矩阵的正交性得出的， $\mathbf{P}^\top \mathbf{P} = \mathbf{I}$ 。

[Wang et al. \(2019a\)](#) 证明了给定算子 $\mathbf{Q} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$ 的特征分解，任何重网格不变 (或等变) 函数可以表示为 $f(\mathbf{Q}) = f(\mathbf{\Lambda})$ 和 $\mathbf{F}(\mathbf{Q}) = \mathbf{V}\mathbf{F}(\mathbf{\Lambda})$ ，或者换句话说，重网格不变函数只涉及 \mathbf{Q} 的谱。事实上，拉普拉斯特征值的函数在实践中已被证明对表面离散化和扰动是鲁棒的，这解释了基于拉普拉斯算子的谱构造在计算机图形学中的流行，以及在图的深度学习中的流行 ([Defferrard et al., 2016](#); [Levie et al., 2018](#))。由于这个结果指的是一个一般的算子 \mathbf{Q} ，除了无处不在的拉普拉斯算子之外，还有多种选择——值得注意的例子包括 *Dirac* ([Liu et al., 2017](#); [Kostrikov et al., 2018](#)) 或 *Steklov* ([Wang et al., 2018](#)) 算子，以及可学习的参数算子 ([Wang et al., 2019a](#))。

5 几何深度学习模型

已经彻底研究了我们的几何深度学习蓝图的各种实例 (对于域、对称群和局部性概念的不同选择), 我们准备讨论如何执行这些规定可以生成一些最流行的深度学习体系结构.

我们的论述, 再一次, 将不会严格按照普遍性的顺序. 我们最初涵盖了三种体系结构, 其实现几乎直接遵循我们之前的讨论: 卷积神经网络 (*CNNs*)、群等变 *CNNs* 和图神经网络 (*GNNs*).

然后, 我们将仔细研究图形结构未知的情况下的 *GNNs* 变体 (即无序集), 通过我们的讨论, 我们将把流行的深度集和转换器架构描述为 *GNNs* 的实例.

根据我们对几何图形和格网的讨论, 我们首先描述等变信息传递网络, 它将显式几何对称性引入 *GNN* 计算. 然后, 我们展示了我们的测地线和规范对称理论可以在深度学习中实现的方法, 恢复了一族内在的格网神经网络 (包括测地线神经网络、*MoNet* 和规范等变格网神经网络).

最后, 我们从时序的 (temporal) 角度回顾格网领域. 这个讨论将引导我们到递归神经网络 (*RNNs*). 我们将展示 *RNNs* 在时间格网上平移等变的方式, 还将研究它们对时间规整变换的稳定性. 这个属性对于正确处理远程依赖关系是非常理想的, 并且对这样的转换执行类不变性产生了门控神经网络的类 (包括流行的 *RNN* 模型, 如 *LSTM* 或 *GRU*).

虽然我们希望以上讨论了在写作时使用的大多数关键的深度学习架构, 但我们很清楚每天都有新的神经网络实例被提出. 因此, 我们不打算覆盖每一个可能的架构, 我们希望下面的部分足够说明问题, 以至于读者能够使用不变性 (*invariances*) 和对称性 (*symmetries*) 的镜头轻松地对任何未来的几何深度学习发展进行分类.

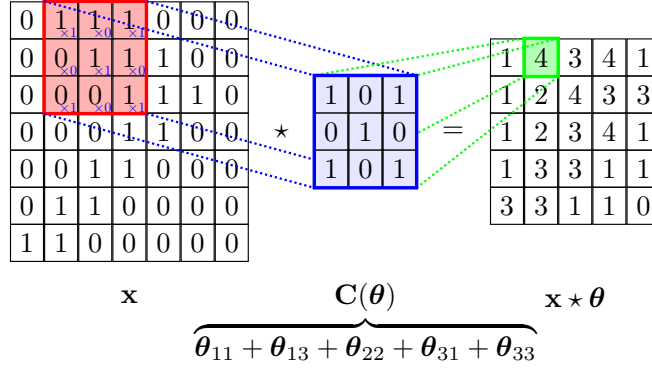


图 14: 用滤波器 $\mathbf{C}(\boldsymbol{\theta})$ 为图像 \mathbf{x} 做卷积的过程. 滤波器参数 $\boldsymbol{\theta}$ 可以表示为发生器 $\boldsymbol{\theta}_{vw}$ 的线性组合.

5.1 卷积神经网络

卷积神经网络可能是最早和最广为人知的深度学习体系结构的例子, 它遵循 3.5 节中概述的几何深度学习的蓝图. 在第 4.2 节中, 我们已经完全刻画了线性和局部平移等变算子的类, 由带局部滤波器 $\boldsymbol{\theta}$ 的卷积 $\mathbf{C}(\boldsymbol{\theta})\mathbf{x} = \mathbf{x} \star \boldsymbol{\theta}$ 给出. 让我们首先关注标量值 (“单通道” 或 “灰度”) 离散化图像, 其中域是格网 $\Omega = [H] \times [W]$, $\mathbf{u} = (u_1, u_2)$ 和 $\mathbf{x} \in \mathcal{X}(\Omega, \mathbb{R})$.

回想一下, $\mathbf{C}(\boldsymbol{\theta})$ 是一个带参数 $\boldsymbol{\theta}$ 的循环 (circulant) 矩阵.

与尺寸为 $H^f \times W^f$ 的紧支撑滤波器的任何卷积可以写成发生器 $\boldsymbol{\theta}_{1,1}, \dots, \boldsymbol{\theta}_{H^f, W^f}$, 例如由单位峰值 $\boldsymbol{\theta}_{vw}(u_1, u_2) = \delta(u_1 - v, u_2 - w)$ 给出. 因此, 任何局部线性等变映射都可以表示为

$$\mathbf{F}(\mathbf{x}) = \sum_{v=1}^{H^f} \sum_{w=1}^{W^f} \alpha_{vw} \mathbf{C}(\boldsymbol{\theta}_{vw}) \mathbf{x}, \quad (26)$$

其在坐标上对应于熟悉的 2D 卷积 (参见图 14 的概览):

$$\mathbf{F}(\mathbf{x})_{uv} = \sum_{a=1}^{H^f} \sum_{b=1}^{W^f} \alpha_{ab} x_{u+a, v+b}. \quad (27)$$

基础 $\boldsymbol{\theta}_{vw}$ 的其他选择也是可能的, 并且将产生等效操作 (对于 α_{vw} 的潜在不同选择). 一个常见的例子是方向导数 (directional derivatives): $\boldsymbol{\theta}_{vw}(u_1, u_2) =$

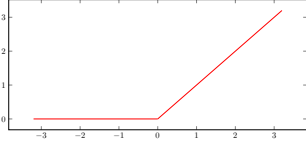
请注意, 我们通常想象 \mathbf{x} 和 $\boldsymbol{\theta}_{vw}$ 是 2D 矩阵, 但是在这个等式中, \mathbf{x} 和 $\boldsymbol{\theta}_{vw}$ 都将它们的两个坐标维展平为一维, 使得 \mathbf{x} 是一个向量, $\mathbf{C}(\boldsymbol{\theta}_{vw})$ 是一个矩阵.

$\delta(u_1, u_2) - \delta(u_1 - v, u_2 - w), (v, w) \neq (0, 0)$ 加上局部平均值 $\theta_0(u_1, u_2) = \frac{1}{H_f W_f}$. 事实上, 方向导数可以被看作是图形上扩散过程的格网特定模拟, 如果我们假设每个像素是连接到格网中紧邻像素的节点, 则可以恢复方向导数.

当标量输入通道被多个通道 (例如, RGB 颜色, 或者更一般地, 任意数量的特征映射) 代替时, 卷积滤波器变成卷积张量, 其将输入特征的任意线性组合表示成输出特征映射. 在坐标中, 这可以表示为:

$$\mathbf{F}(\mathbf{x})_{uvj} = \sum_{a=1}^{H^f} \sum_{b=1}^{W^f} \sum_{c=1}^M \alpha_{jabc} x_{u+a, v+b, c}, \quad j \in [N], \quad (28)$$

其中 M 和 N 分别是输入和输出通道的数量. 这一基本操作包含了一大类神经网络架构, 正如我们将在下一节中展示的那样, 这些架构对计算机视觉、信号处理等许多领域产生了深远的影响. 在这里, 与其剖析 $CNNs$ 的无数可能的架构变体, 我们更喜欢关注一些使其得以广泛使用的重要创新.



$ReLU$, 通常被认为是一种“现代”的架构选择, 已经被

用于 *Neocognitron*

(*Fukushima and Miyake, 1982*). 整流相当于解调原理, 在电气工程中是基础, 作为很多传输协议的基础, 比如 *FM* 收音机; 并且在神经元活动的模型中也具有突出的作用.

高效多尺度计算 正如在 *GDL* 通用对称模板中所讨论的, 从卷积算子中提取平移不变特征需要一个非线性步骤. 卷积特征通过非线性激活函数 σ 进行处理, 以元素方式作用于输入, 即 $\sigma: \mathcal{X}(\Omega) \rightarrow \mathcal{X}(\Omega)$, 如 $\sigma(\mathbf{x})(u) = \sigma(\mathbf{x}(u))$. 也许在撰写本文时最流行的例子是修正线性单位 ($ReLU$): $\sigma(x) = \max(x, 0)$. 这种非线性有效地校正 (rectifies) 了信号, 将它们的能量推向更低的频率, 并通过迭代结构来计算跨尺度的高阶相互作用.

早在 *Fukushima and Miyake (1982)* 和 *LeCun et al. (1998)* 的早期工作中, $CNNs$ 和类似架构就具有多尺度结构, 其中在每个卷积层 (28) 之后, 执行格网粗化 $\mathbf{P}: \mathcal{X}(\Omega) \rightarrow \mathcal{X}(\Omega')$, 其中格网 Ω' 具有比 Ω 更粗的分辨率. 这使得多尺度滤波器能够有效地增加感受野, 但每个尺度保持恒定数量的参数. 可以使用几种信号粗化策略 (称为池化), 最常见的是应用低通抗混叠滤波器 (例如局部平均), 然后是格网下采样或非线性最大池化.

总之, “普通” CNN 层可以表示为几何深度学习蓝图中已经介绍的基本对象

的组成:

$$\mathbf{h} = \mathbf{P}(\sigma(\mathbf{F}(\mathbf{x}))) , \quad (29)$$

即等变线性层 \mathbf{F} 、粗化操作 \mathbf{P} 和非线性 σ . 也可以在 $CNNs$ 内执行平移不变的全局池化操作. 直观地说, 这涉及到每个像素——经过几次卷积后, 总结出一个以它为中心的面片——提出图像的最终表示, 最终的选择由这些建议的聚合形式来指导. 这里一个流行的选择是平均函数, 因为它的输出将保持相似的大小, 而与图像大小无关 (*Springenberg et al., 2014*).

遵循 CNN 蓝图的突出例子 (其中一些我们将在下面讨论) 显示在图 15 中.

深度与残差网络 因此, 最简单形式的 CNN 架构由超参数 $(H_k^f, W_k^f, N_k, p_k)_{k \leq K}$ 指定, 其中 $M_{k+1} = N_k, p_k = 0, 1$ 表示是否进行格网粗化. 虽然所有这些超参数在实践中都很重要, 但一个特别重要的问题是理解深度 K 在 CNN 架构中的作用, 以及在选择这样一个关键超参数时涉及到哪些基本的权衡, 尤其是与滤波器大小 (H_k^f, W_k^f) 的关系.

虽然这个问题的严格答案仍然难以捉摸, 但近年来收集的越来越多的经验证据表明, 更深 (大 K) 但更薄 (小 $((H_k^f, W_k^f))$) 的模型是一个有利的折衷. 在这种情况下, *He et al. (2016)* 的一个关键见解是重新参数化每个卷积层, 以模拟先前特征的扰动, 而不是一般的非线性变换:

$$\mathbf{h} = \mathbf{P}(\mathbf{x} + \sigma(\mathbf{F}(\mathbf{x}))) . \quad (30)$$

所得的残差网络提供了优于先前公式的几个关键优点. 本质上, 残差参数化与深层网络是底层连续动力系统的离散化的观点一致, 建模为常微分方程 (ODE). 至关重要的是, 通过模拟其速度来学习动力系统比直接学习其位置要容易得多. 在我们的学习环境中, 这转化为具有更有利几何形状的优化景观, 从而能够训练比以前更深入的架构. 正如将在未来的工作中讨论的, 使用深度神经网络的学习定义了一个非凸优化问题, 它可以在某些简化的情况下使用梯度下降方法有效地解决. $ResNet$ 参数化的主要优势已经在简单场景中进行了严格分析 (*Hardt and Ma, 2016*), 并且仍然是理论研究的活跃领域. 最后, 神经微分方程组 (*Chen et al., 2018*) 是一种最近流行的体系结构, 它

只由本段提到的运算组成的神经网络通常被称为“全卷积”. 相比之下, 一旦应用了足够的等变层和粗化层, 许多中枢神经系统就会在空间轴上展平图像, 并将它们传递给 MLP 分类器. 这就失去了平移不变性.

从历史上看, $ResNet$ 模型早于高速网络 (*Srivastava et al., 2015*), 高速网络允许更通用的门控机制来控制残差信息流.

在这种情况下, $ResNet$ 正在对一个常微分方程进行前向欧拉离散化: $\dot{\mathbf{x}} = \sigma(\mathbf{F}(\mathbf{x}))$

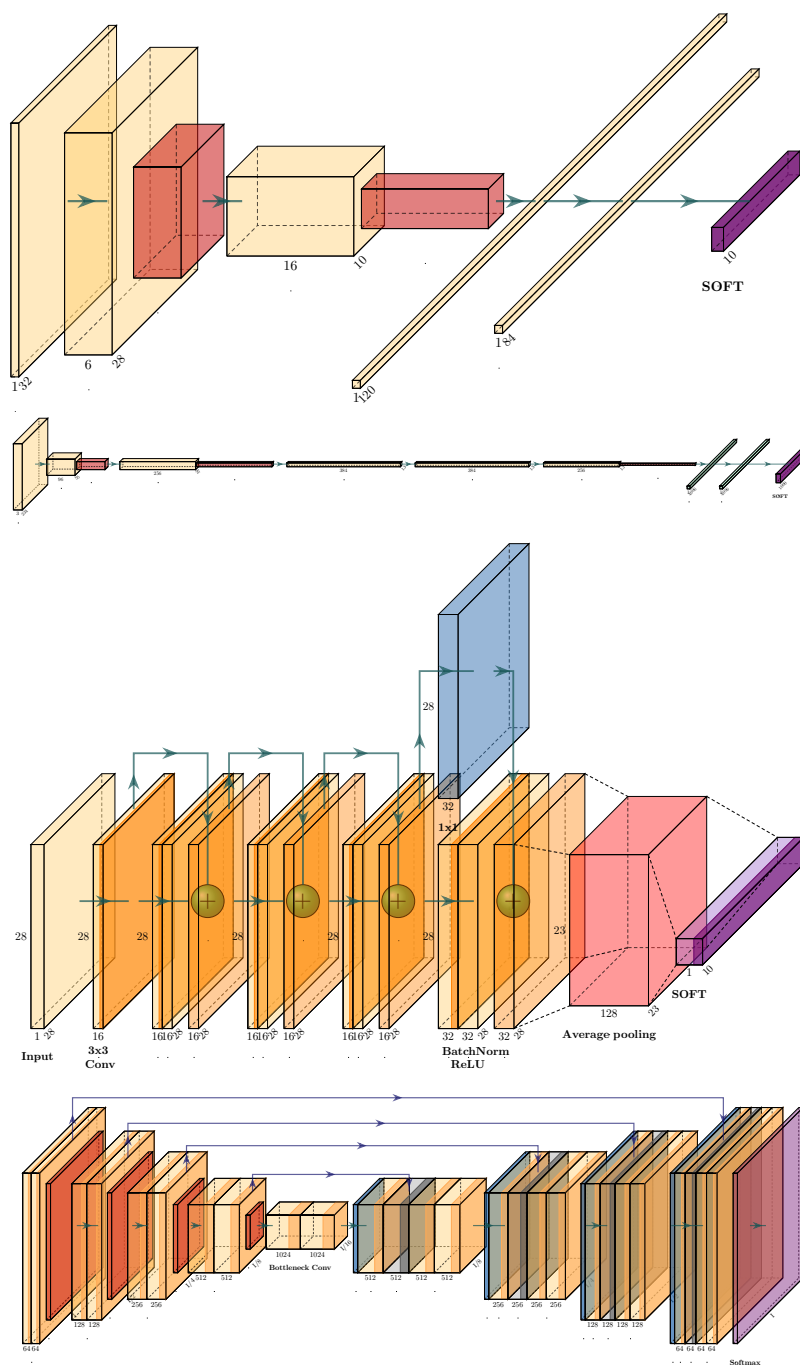


图 15: CNN 架构的突出例子. 自上而下: LeNet (LeCun et al., 1998)、AlexNet (Krizhevsky et al., 2012)、ResNet (He et al., 2016) 和 U-Net (Ronneberger et al., 2015). 使用绘图神经网络包绘制 (Iqbal, 2018).

通过直接学习 $ODE\dot{\mathbf{x}} = \sigma(\mathbf{F}(\mathbf{x}))$ 的参数并依靠标准数值积分, 将与微分方程组的类比推得更远.

归一化 另一个显著提高中枢神经系统经验性能的重要算法创新是标准化的概念. 在早期的神经活动模型中, 假设神经元执行某种形式的局部“增益控制”, 其中层系数 \mathbf{x}_k 由 $\mathbf{x}_k = \sigma_k^{-1} \odot (\mathbf{x}_k - \mu_k)$ 代替. 这里, μ_k 和 σ_k 分别表示 \mathbf{x}_k 的一阶和二阶矩信息. 此外, 它们可以是全局计算的, 也可以是局部计算的.

在深度学习的背景下, 这一原则通过批归一化 (batch normalisation) 层被广泛采用 (*Ioffe and Szegedy, 2015*), 随后还有几个变体 (*Ba et al., 2016*; *Salimans and Kingma, 2016*; *Ulyanov et al., 2016*; *Cooijmans et al., 2016*; *Wu and He, 2018*). 尽管有人试图从更好的条件优化景观的角度来严格解释归一化的好处 (*Santurkar et al., 2018*), 但在撰写本文时, 仍然缺乏一个可以提供指导原则的一般理论.

我们注意到, 甚至在批量标准化出现之前, 神经网络的标准活化激活就已经受到关注. 例如, 见 *Lyu and Simoncelli (2008)*.

数据增强 虽然中枢神经系统编码与平移不变性和尺度分离相关的几何先验, 但它们没有明确说明保留语义信息的其他已知变换, 例如闪电或颜色变化, 或小的旋转和膨胀. 用最小的体系结构变化来合并这些先验的实用方法是执行数据扩充, 其中人工地对输入图像执行所述转换, 并将它们添加到训练集中.

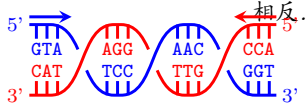
数据增强已经在实践中取得了成功, 并得到了广泛的应用——不仅用于训练最先进的视觉架构, 还用于支持自我监督和因果表示学习的若干发展 (*Chen et al., 2020*; *Grill et al., 2020*; *Mitrovic et al., 2020*). 然而, 就样本复杂度而言, 它是可证明次优的 (*Mei et al., 2021*); 更有效的策略是考虑具有更丰富不变性群的架构——正如我们接下来讨论的.

5.2 群等变卷积神经网络

如第4.3节所讨论的, 我们可以将卷积运算从欧几里得空间上的信号推广到由群 \mathfrak{G} 作用的任何齐次空间 (homogeneous space) Ω 上的信号. 类似于欧几里得卷积, 其中平移的滤波器与信号匹配, 群卷积的思想是使用群作用在域中移动滤波器, 例如通过旋转和平移. 借助群作用的传递性 (transitivity), 我们可以将滤波器移动到 Ω 上的任何位置. 在本节中, 我们将讨论群卷积一般思想的几个具体例子, 包括实现方面和架构选择.

离散群卷积 我们首先考虑域 Ω 和群 \mathfrak{G} 是离散的情况. 作为我们的第一个例子, 我们将医学体积图像表示为具有离散平移和旋转对称性的 3D 格网上的信号. 该域是 3D 立方体格网 $\Omega = \mathbb{Z}^3$, 并且图像 (例如 MRI 或 CT 3D 扫描) 被建模为函数 $x: \mathbb{Z}^3 \rightarrow \mathbb{R}$, 即 $x \in \mathcal{X}(\Omega)$. 虽然实际上这样的图像在有限长方体 $[W] \times [H] \times [D] \subset \mathbb{Z}^3$ 上有支撑, 但是我们更愿意将它们看作 \mathbb{Z}^3 上具有适当零填充的函数. 作为我们的对称性, 我们考虑 \mathbb{Z}^3 上的群 $\mathfrak{G} = \mathbb{Z}^3 \rtimes O_h$ 的距离和方向保留变换. 这个群包括平移 (\mathbb{Z}^3) 和绕三个轴旋转 90 度产生的离散旋转 O_h (见图 16).

DNA 是一种生物聚合物分子, 由四个称为核苷酸 (胞嘧啶、鸟嘌呤、腺嘌呤和胸腺嘧啶) 的重复单元组成, 排列成两条以双螺旋形式相互缠绕的链, 其中每个核苷酸与互补的核苷酸 (碱基对 A/T 和 C/G) 配对. 作为我们的第二个例子, 我们考虑由四个字母组成的 DNA 序列: C、G、A 和 T. 这些序列可以在 1D 格网 $\Omega = \mathbb{Z}$ 上表示为信号 $x: \mathbb{Z} \rightarrow \mathbb{R}^4$, 其中每个字母在 \mathbb{R}^4 被单热编码. 自然地, 我们在格网上有一个离散的 1D 平移对称性, 但是 DNA 序列有一个额外的有趣的对称性. 这种对称性源于 DNA 物理上体现为双螺旋的方式, 以及细胞的分子机制读取它的方式. 双螺旋的每一条链都从所谓的 5' 端开始, 以 3' 端结束, 一条链上的 5' 端由另一条链上的 3' 端补充. 换句话说, 两股具有相反的方向. 由于 DNA 分子总是从 5' 端开始被读出, 但我们不知道是哪一个, 所以像 ACCCTGG 这样的序列相当于每个字母都被它的互补序列 CCAGGGT 取代的反向序列. 这就是所谓的字母序列的反补对称 (reverse-complement symmetry). 因此, 我们有对应于恒等式 0 和逆补变换 1 (以及组合 $1 + 1 = 0 \pmod{2}$) 的二元群 $\mathbb{Z}_2 = \{0, 1\}$. 整个群包含了平移 (translations) 和逆补 (reverse-complement) 变换.



脱氧核糖核酸双螺旋结构的示意图, 两条链被染成蓝色和红色. 注意螺旋中的序列是如何互补和反向阅读的 (从 5' 到 3').

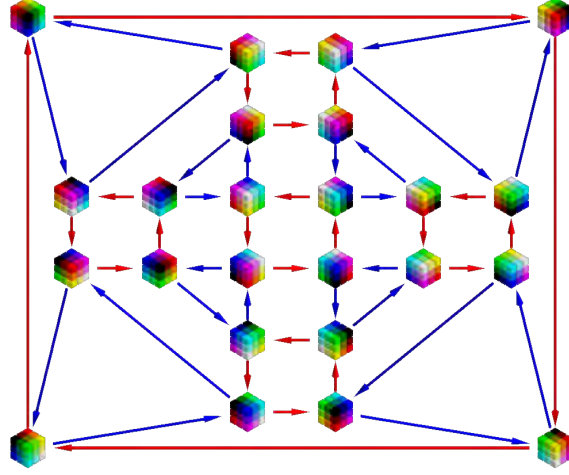


图 16: 3×3 滤波器, 由离散旋转群 O_h 的所有 24 个元素旋转, 由围绕垂直轴旋转 90 度 (红色箭头) 和围绕对角轴旋转 120 度 (蓝色箭头) 生成.

在我们的例子中, 我们在第 4.3 节中定义的群卷积 (14) 给出如下:

$$(x \star \theta)(\mathfrak{g}) = \sum_{u \in \Omega} x_u \rho(\mathfrak{g}) \theta_u, \quad (31)$$

(单通道) 输入信号 x 与 $\mathfrak{g} \in \mathfrak{G}$ 通过 $\rho(\mathfrak{g})\theta_u = \theta_{\mathfrak{g}^{-1}u}$ 转换的滤波器 θ 和输出的内积 $x \star \theta$ 是 \mathfrak{G} 上的一个函数. 注意, 由于 Ω 是离散的, 我们用一个和代替了等式 (14) 中的积分.

变换 + 卷积方法 我们将展示群卷积可以分两步实现: 一个滤波器变换步骤, 和一个平移卷积步骤. 滤波器变换步骤包括创建基本滤波器的旋转 (或逆补变换) 副本, 而平移卷积与标准 CNNs 相同, 因此可以在 GPU 等硬件上高效计算. 要看到这一点, 请注意, 在我们的两个例子中, 我们可以将一般变换 $\mathfrak{g} \in \mathfrak{G}$ 写成变换 $\mathfrak{h} \in \mathfrak{H}$ (例如旋转或逆补变换), 然后是平移 $\mathfrak{k} \in \mathbb{Z}^d$, 即 $\mathfrak{g} = \mathfrak{k}\mathfrak{h}$ (并置表示群元素 k 和 H 的组成). 通过群表示的性质, 我们得到了

$\rho(\mathfrak{g}) = \rho(\mathfrak{k}\mathfrak{h}) = \rho(\mathfrak{k})\rho(\mathfrak{h})$. 因此,

$$\begin{aligned}(x \star \theta)(\mathfrak{k}\mathfrak{h}) &= \sum_{u \in \Omega} x_u \rho(\mathfrak{k})\rho(\mathfrak{h})\theta_u \\ &= \sum_{u \in \Omega} x_u (\rho(\mathfrak{h})\theta)_{u-\mathfrak{k}}\end{aligned}\tag{32}$$

我们认为最后一个等式是信号 x 和变换滤波器 $\rho(\mathfrak{h})\theta$ 的标准 (平面欧几里得) 卷积. 因此, 为了实现这些群的群卷积, 我们采用正则滤波器 θ , 为每个 $\mathfrak{h} \in \mathfrak{H}$ (例如, 每个旋转 $\mathfrak{h} \in O_h$ 或反向互补 DNA 对称性 $\mathfrak{h} \in \mathbb{Z}_2$) 创建变换副本 $\theta_{\mathfrak{h}} = \rho(\mathfrak{h})\theta$, 然后将 x 与这些滤波器卷积: $(x \star \theta)(\mathfrak{k}\mathfrak{h}) = (x \star \theta_{\mathfrak{h}})(\mathfrak{k})$. 对于我们的两个例子, 对称性通过简单地置换滤波器系数来作用于滤波器, 如图 16 所示的离散旋转. 因此, 这些操作可以使用具有预先计算的索引的索引操作来有效地实现.

当我们定义由群卷积输出 $x \star \theta$ 的特征映射作为 \mathfrak{G} 上的函数时, 我们可以将 \mathfrak{g} 分为 \mathfrak{h} 和 \mathfrak{k} 的事实意味着我们也可以将它们视为欧几里得特征映射 (有时称为方向通道) 的堆栈, 每个滤波器变换/方向 \mathfrak{k} 有一个特征映射. 例如, 在我们的第一个示例中, 我们将每个滤波器旋转 (图 16 中的每个节点) 与一个特征映射相关联, 该特征映射是通过卷积 (在传统的平移意义下) 旋转的滤波器获得的. 因此, 这些特征映射仍然可以存储为 $W \times H \times C$ 数组, 其中通道 C 的数量等于独立滤波器的数量乘以变换 $\mathfrak{h} \in \mathfrak{H}$ 的数量 (例如旋转).

如 4.3 节所示, 群卷积是等变的: $(\rho(\mathfrak{g})x) \star \theta = \rho(\mathfrak{g})(x \star \theta)$. 这在定向通道方面的意思是, 在 \mathfrak{h} 的作用下, 每个定向通道都被变换, 定向通道本身被置换. 例如, 如果我们在图 16 中为每个变换关联一个方向通道, 并围绕 z 轴旋转 90 度 (对应于红色箭头), 则要素地图将被置换, 如红色箭头所示. 这种描述清楚地表明, 群卷积神经网络与传统的 CNN 有很大的相似性. 因此, 第 5.1 节中讨论的许多网络设计模式, 如残差网络, 也可以用于群卷积.

傅立叶域球面卷积神经网络 对于我们在第 4.3 节中看到的球体的连续对称群, 可以使用适当的傅里叶变换在谱域中实现卷积 (我们提醒读者 \mathbb{S}^2 上的卷积是 $\text{SO}(3)$ 上的函数, 因此我们需要在这两个域上定义傅里叶变换, 以便

实现多层球形 CNNs). 球谐函数 (Spherical harmonics) 是 $2D$ 球面上的正交基, 类似于复指数的经典傅里叶基. 在特殊的正交群上, 傅里叶基被称为维格纳函数 (Wigner D-functions). 在这两种情况下, 傅里叶变换 (系数) 都是作为基函数的内积来计算的, 卷积定理的一个类比成立: 可以将傅里叶域中的卷积作为傅里叶变换的元素积来计算. 此外, 还存在类似快速傅立叶变换的算法来有效地计算 S^2 和 S^3 上的傅立叶变换. 进一步细节可以参考 [Cohen et al. \(2018\)](#).

5.3 图神经网络

图神经网络是利用置换群的性质在图形上实现我们的几何深度学习蓝图. GNNs 是目前存在的最普通的深度学习体系结构, 正如我们将在本文中看到的, 大多数其他深度学习体系结构可以被理解为具有附加几何结构的 GNN 的特例.

根据我们在第 4.1 节中的讨论, 我们考虑用邻接矩阵 \mathbf{A} 和节点特征 \mathbf{X} 来指定图. 我们将研究 GNN 体系结构, 它们是通过在局部邻域上应用共享置换不变函数 $\mathbf{F}(\mathbf{X}, \mathbf{A})$ 而构造的置换等变函数 $\phi(\mathbf{x}_u, \mathbf{X}_{\mathcal{N}_u})$. 在各种伪装下, 这个局部函数 可以被称为“扩散”、“传播”或“消息传递”, 并且这种 \mathbf{F} 的整体计算被称为“GNN 层”.

GNN 图层的设计和研究是写作时深度学习最活跃的领域之一, 使其成为一个具有挑战性的景观. 幸运的是, 我们发现绝大多数文献可能仅来自三种“调料” (flavours) 的 GNN 层面 (图 17), 我们将在此介绍. 这些调料决定了 ϕ 变换邻域特征的程度, 从而允许在对整个图形的交互进行建模时有不同程度的复杂性.

在所有三种调料中, 置换不变性是通过用一些置换不变函数 l 聚集来自 $\mathbf{X}_{\mathcal{N}_u}$ 的特征 (潜在地, 通过一些函数 ψ 变换), 然后通过一些函数 \oplus 更新节点 u 的特征来保证的. 一般来说, ψ 和 ϕ 是可学习的, 其中 \oplus 实现为非参数运算, 如和、均值或最大值, 尽管它也可以使用递归神经网络来构建 ([Murphy](#)

最常见的是, ψ 和 ϕ 是具有激活函数的可学习仿射变换; 例如, $\psi(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{b}$; $\phi(\mathbf{x}, \mathbf{z}) = \sigma(\mathbf{W}\mathbf{x} + \mathbf{U}\mathbf{z} + \mathbf{b})$, 其中, $\mathbf{W}, \mathbf{U}, \mathbf{b}$ 是可学习参数, σ 是激活函数, 如修正线性单位. \mathbf{x}_u 到 ϕ 的额外输入代表可选的残差链接, 这通常非常有用.

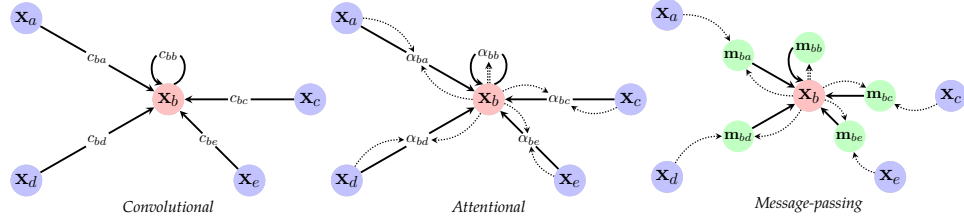


图 17: GNN 层三种调料的数据流可视化, 例如, 我们使用图10中节点 b 的邻域来说明这一点. 从左到右: 卷积, 其中发送者节点特征乘以常数 c_{uv} 注意, 其中该乘数是通过接收者对发送者的注意机制隐式计算的: $\alpha_{uv} = a(\mathbf{x}_u, \mathbf{x}_v)$; 和消息传递, 其中基于向量的消息是基于发送方和接收方计算的: $\mathbf{m}_{uv} = \psi(\mathbf{x}_u, \mathbf{x}_v)$.

et al., 2018).

在卷积层调料中 (Kipf and Welling, 2016a; Defferrard et al., 2016; Wu et al., 2019), 邻域节点的特征用固定权重直接聚合,

$$\mathbf{h}_u = \phi \left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{N}_u} c_{uv} \psi(\mathbf{x}_v) \right). \quad (33)$$

这里, c_{uv} 是指节点 v 对节点 u 的重要性. 它是一个常数, 经常直接依赖于 \mathbf{A} 中代表图形结构的元素. 请注意, 当聚合算子 \bigoplus 被选为求和时, 它可以被视为线性扩散或位置相关的线性滤波, 卷积的泛化. 特别是, 我们在第 4.4 节和第 4.6 节中看到的频谱滤波器属于这一类, 因为它们相当于将固定的局部算子 (例如拉普拉斯矩阵) 应用于逐节点信号.

在注意力层调料中 (Veličković et al., 2018; Monti et al., 2017; Zhang et al., 2018), 这些相互作用是隐式的

$$\mathbf{h}_u = \phi \left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{N}_u} a(\mathbf{x}_u, \mathbf{x}_v) \psi(\mathbf{x}_v) \right). \quad (34)$$

这里, a 表示可学习的自注意力机制, 它隐式计算重要性系数 $\alpha_{uv} = a(\mathbf{x}_u, \mathbf{x}_v)$. 它们通常在所有邻居中被 *softmax* 归一化. 当 \bigoplus 表示求和时, 聚集仍然是邻域节点特征的线性组合, 但是现在权重依赖于特征.

值得注意的是, 这种风格并不表示卷积的每个 GNN 层 (在与图结构交换的意义上), 而是涵盖了实践中提出的大多数这样的方法. 我们将在未来的工作中提供详细的讨论和扩展

最后, 信息传递的调料 (*Gilmer et al., 2017; Battaglia et al., 2018*) 相当于计算边的任意向量 (“消息”),

$$\mathbf{h}_u = \phi \left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{N}_u} \psi(\mathbf{x}_u, \mathbf{x}_v) \right). \quad (35)$$

这里, ψ 是一个可学习的消息函数, 计算发送给 u 的 v 的向量, 聚合可以被认为是在图上传递消息的一种形式.

需要注意的一件重要的事情是这些方法之间的表示包含: 卷积 \subseteq 注意 \subseteq 消息传递. 事实上, 注意力 $GNNs$ 可以通过一种被实现为查找表 $a(\mathbf{x}_u, \mathbf{x}_v) = c_{uv}$ 的注意力机制来表示卷积 $GNNs$, 并且卷积 $GNNs$ 和注意 $GNNs$ 都是消息传递的特殊情况, 其中消息仅仅是发送者节点的特征: 对于卷积 $GNNs$, $\psi(\mathbf{x}_u, \mathbf{x}_v) = c_{uv}\psi(\mathbf{x}_v)$, 对于注意力 $GNNs$, $\psi(\mathbf{x}_u, \mathbf{x}_v) = a(\mathbf{x}_u, \mathbf{x}_v)\psi(\mathbf{x}_v)$.

这并不意味着消息传递 $GNNs$ 总是最有用的变体; 由于它们必须计算边上的向量值消息, 它们通常更难训练, 并且需要庞大的内存. 此外, 在大量自然生成的图中, 图的边编码为类相似性 (即边 (u, v) 意味着 u 和 v 可能具有相同的输出). 对于这样的图 (通常被称为同构图), 从正则性 (*regularisation*) 和可扩展性 (*scalability*) 两个方面来看, 邻域之间的卷积聚合通常是更好的选择. 注意力 $GNNs$ 提供了一个 “中间地带”: 它们允许模拟邻里之间的复杂互动, 同时只计算边上的标量值, 使它们比消息传递更具可扩展性.

这里提出的 “三种调料” 分别考虑了简洁性, 不可避免地忽略了 GNN 模型的大量细微差别、洞察、泛化性和历史背景. 重要的是, 它排除了基于 *Weisfeiler-Lehman* 层次的高维 GNN 和依赖于显式计算图傅里叶变换的谱 $GNNs$.

5.4 Deep Sets, Transformers, 潜在图推理

我们通过评论用于学习无序集 (unordered sets) 表示的置换-等变神经网络结构来结束对神经网络的讨论. 虽然集合在我们在本文中讨论的领域中具有最少的结构, 但是它们的重要性最近被高度流行的架构所强调, 例如

Transformers (Vaswani et al., 2017) 和 *Deep Sets* (Zaheer et al., 2017).. 在第 4.1 节的语言中, 我们假设给定了一个节点特征矩阵, 但是节点之间没有任何指定的邻接或排序信息. 具体的架构将通过决定在多大程度上对节点之间的交互进行建模来产生.

空边集 无序集合是指没有任何额外结构或几何信息的——因此, 可以认为处理它们的最自然的方法是完全独立地处理每个集合元素. 这转化为在这些输入作用上置换等变函数, 这已经在第 4.1 节中介绍过了: 一个应用于每个孤立节点的共享变换. 假设采用与 GNNs 相同的符号 (第 5.3 节), 这样的模型可以表示为

$$\mathbf{h}_u = \psi(\mathbf{x}_u),$$

其中 ψ 是一个可学习的变换. 可以观察到, 这是卷积 GNN 的一种特殊情况, 其 $\mathcal{N}_u = \{u\}$ —或等价地, $\mathbf{A} = \mathbf{I}$. 这种体系结构通常被称为 *Deep Sets*, Zaheer et al. (2017) 的工作从理论上证明了这种体系结构的几个通用近似性质. 需要注意的是, 在处理点云时, 处理无序集合需要计算机视觉和图形学模型; 其中, 有类模型被称为 *PointNets* (Qi et al., 2017).

完全集 虽然假设空边集是在无序集上构建函数的一种非常有效的构造, 但我们通常会期望该集的元素表现出某种形式的关系结构, 即节点之间存在潜在的图. 假定 $\mathbf{A} = \mathbf{I}$ 会丢弃任何此类结构, 并可能产生次优性能. 相反, 我们可以假设, 在没有任何其他先验知识的情况下, 我们不能预先排除节点之间的任何可能的链接. 在这种方法中, 我们假设完全图, $\mathbf{A} = \mathbf{1}\mathbf{1}^\top$; 等价地, 由于我们不假设访问任何交互作用系数, 在这样的图上运行卷积型神经网络将等于:

$$\mathbf{h}_u = \phi\left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{V}} \psi(\mathbf{x}_v)\right),$$

这是 \bigoplus 置换不变性的直接结果. 其中, 第二个输入 $\bigoplus_{v \in \mathcal{V}} \psi(\mathbf{x}_v)$ 对于所有节点 u 都是相同的, 因此使得模型的表达等同于完全忽略该输入; 即上述情况 $\mathbf{A} = \mathbf{I}$.

这激发了人们使用更富表现力的 GNN “调料”，注意力

$$\mathbf{h}_u = \phi \left(\mathbf{x}_u, \bigoplus_{v \in \mathcal{V}} a(\mathbf{x}_u, \mathbf{x}_v) \psi(\mathbf{x}_v) \right) \quad (36)$$

这就产生了“自注意力”的算子, *Transformer* 架构的核心 (Vaswani et al., 2017). 假设对注意力系数 (例如 *softmax*) 进行某种归一化, 我们可以将所有标量 $a(\mathbf{x}_u, \mathbf{x}_v)$ 限制在 $[0, 1]$ 范围内; 因此, 我们可以把自注意力看作是推断一个软邻接矩阵 $a_{uv} = a(\mathbf{x}_u, \mathbf{x}_v)$, 作为一些下游任务梯度优化的副产品.

上述观点意味着我们可以在一个完全图上把 *Transformers* 精确地看作注意力 GNNs (Joshi, 2020). 然而, 这显然与 *Transformers* 最初被提议用于建模序列相冲突—— \mathbf{h}_u 的表示应该注意节点 u 在序列中的位置, 而完全图聚合会忽略这一点. *Transformers* 通过引入位置编码来解决这个问题: 节点特征 \mathbf{x}_u 被扩充以对序列中节点 u 的位置进行编码, 通常作为正弦波的样本, 其频率取决于 u .

在图中, 不存在节点的自然排序, 对这种位置编码提出了多种选择. 虽然我们将这些替代方案放到以后讨论, 但我们注意到一个有希望的方向是实现 *Transformers* 中使用的位置编码可以直接与离散傅立叶变换相关, 从而与“环形格网”的图拉普拉斯特征向量相关. 因此, *Transformers* 的位置编码隐含地代表了我们的假设, 即输入节点连接在一个格网中. 对于更一般的图形结构, 人们可以简单地使用 (假设的) 图拉普拉斯特征向量——这是 Dwivedi and Bresson (2020) 在其经验强大的图 *Transformer* 模型中采用的.

推断边集合 最后, 你可以试着学习潜在的关系结构, 得到一些既不是 \mathbf{I} 也不是 $\mathbf{1}\mathbf{1}^\top$ 的一般 \mathbf{A} . 推导出 GNN 使用的潜在邻接矩阵 \mathbf{A} (通常称为潜图推理 (latent graph inference)) 是图表示学习的一个重要问题. 这是因为假设 $\mathbf{A} = \mathbf{I}$ 可能表现较差, 并且由于内存要求和要聚合的大邻居, $\mathbf{A} = \mathbf{1}\mathbf{1}^\top$ 可能难以实现. 此外, 它最接近“真实”问题: 推断邻接矩阵意味着检测 \mathbf{X} 行之间的有用结构, 这可能有助于构造假设, 如变量之间的因果关系.

不幸的是, 这样的框架必然导致建模复杂性的增加. 具体来说, 它需要适当平

采用信息传递“调料”也是合适的. 虽然流行的物理模拟和关系推理 (e.g. Battaglia et al. (2016); Santoro et al. (2017)) 还没有像

Transformers 那样被广泛使用. 这可能是由于在一个完全图中计算向量消息的内存问题, 或者基于向量的消息比基于“自注意力”的“软邻接”更难解释.

衡结构学习目标 (这是离散的, 因此对基于梯度的优化具有挑战性) 和基于图下游任务. 这使得潜图推理成为一个极具挑战性的复杂问题.

5.5 等变消息传递网络

在图神经网络的许多应用中, 节点特征 (或其部分) 不是任意向量, 而是几何实体的坐标. 例如, 在处理分子图时就是这种情况: 表示原子的节点可能包含关于原子类型及其 3D 空间坐标的信息. 以与分子在空间中变换相同的方式来处理特征的后一部分是我们希望的, 换句话说, 除了之前讨论的标准置换等变性, 与描述刚体运动 (旋转, 平移, 镜像) 的欧几里得群 $E(3)$ 等价.

为了给我们的 (稍微简化的) 分析奠定基础, 我们将区分节点特征 $\mathbf{f}_u \in \mathbb{R}^d$ 和节点空间坐标 $\mathbf{x}_u \in \mathbb{R}^3$; 后者被赋予欧几里得对称结构. 在这种设置下, 等变图层分别显式转换这两个输入, 产生修改的节点特征 \mathbf{f}'_u 和坐标 \mathbf{x}'_u .

我们现在可以按照几何深度学习蓝图, 陈述我们期望的等变性质. 如果输入的空间分量由 $\mathbf{g} \in E(3)$ 变换 (表示为 $\rho(\mathbf{g})\mathbf{x} = \mathbf{R}\mathbf{x} + \mathbf{b}$, 其中 \mathbf{R} 是模拟旋转和反射的正交矩阵, \mathbf{b} 是平移向量), 输出的空间分量以相同的方式变换 (如 $\mathbf{x}'_u \mapsto \mathbf{R}\mathbf{x}'_u + \mathbf{b}$), 而 \mathbf{f}'_u 保持不变.

就像我们之前在一般图的上下文中讨论的置换等变函数的空间一样, 存在大量满足上述约束的 $E(3)$ -等变层, 但是并非所有这些层都是几何稳定的, 或者易于实现. 事实上, 实际上有用的等变层的空间可以很容易地通过简单的分类来描述, 这与我们的空间 GNN 层的“三种调料”并无不同. [Satorras et al. \(2021\)](#) 以等变消息传递的形式提出了一个优雅的方案. 他们的模式如下:

$$\begin{aligned}\mathbf{f}'_u &= \phi \left(\mathbf{f}_u, \bigoplus_{v \in \mathcal{N}_u} \psi_f(\mathbf{f}_u, \mathbf{f}_v, \|\mathbf{x}_u - \mathbf{x}_v\|^2) \right), \\ \mathbf{x}'_u &= \mathbf{x}_u + \sum_{v \neq u} (\mathbf{x}_u - \mathbf{x}_v) \psi_c(\mathbf{f}_u, \mathbf{f}_v, \|\mathbf{x}_u - \mathbf{x}_v\|^2)\end{aligned}$$

其中 ψ_f 和 ψ_c 表示两个不同的 (可学) 函数. 可以表明, 在空间坐标的欧几

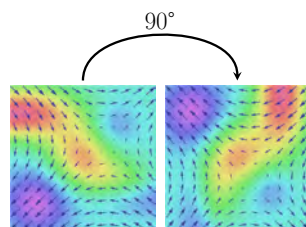
里得变换下, 这种聚集是等变的. 这是因为 \mathbf{f}_u on \mathbf{x}_u 的唯一依赖性距离 $\|\mathbf{x}_u - \mathbf{x}_v\|^2$, 而 $E(3)$ 的作用必然使节点之间的距离保持不变. 此外, 这种层的计算可以被视为“消息传递”GNN 调料的特定实例, 因此它们实现起来是有效的.

总之, 与普通的神经网络相比, [Satorras et al. \(2021\)](#) 能够正确处理图形中每个点的“坐标”. 它们现在被视为 $E(3)$ 群的成员, 这意味着网络输出在输入的旋转、反射和平移下表达正确. 然而, 特征 \mathbf{f}_u 是以通道方式处理的, 并且仍然被认为是在这些转换下不变的标量. 这限制了可以在这样的框架内捕获的空间信息的类型. 例如, 可能希望将某些特征编码为矢量 (例如点速度), 在这种变换下, 这些矢量会改变方向. [Satorras et al. \(2021\)](#) 通过在他们架构的一个变体中引入速度的概念, 部分缓解了这个问题. 速度是每个适当旋转的点的 $3D$ 矢量属性. 然而, 这只是一般表示的一个小的子空间, 可以用 $E(3)$ 等变网络来学习. 一般来说, 节点特征可以编码任意维数的张量, 这些张量仍然会以明确定义的方式根据 $E(3)$ 进行变换.

因此, 虽然上面讨论的体系架构已经为许多实际的输入表示提供了一个优雅的等变解决方案, 但在某些情况下, 可能需要探索满足等变属性的更广泛的函数集合. 处理这种场景的现有方法可以分为两类: 不可约表示 (其中前面提到的层是简化的实例) 和正则表示. 我们在这里简单介绍一下, 把详细的讨论留给以后的工作.

不可约表示 不可约表示建立在旋转平移群的所有元素都可以变成不可约形式的发现之上: 由块对角矩阵旋转的向量. 至关重要的是, 这些块中的每一个都是维格纳矩阵 (Wigner D-matrix) (前面提到的球面神经网络的傅里叶基). 使用等变核从一组不可约表示映射到另一组不可约表示的方法. 为了找到等变映射的全部集合, 可以直接求解这些核上的等变约束. 其解构成了由 Clebsch-Gordan 矩阵和球谐函数导出的等变基矩阵的线性组合.

不可约表示方法的早期例子包括张量场网络 (*Tensor Field Networks* ([Thomas et al., 2018](#))) 和 $3D$ 可控卷积神经网络 (*3D Steerable CNNs* ([Weiler et al.,](#)



虽然标量要素 (热图) 在旋转时不会改变, 但矢量要素 (箭头) 可能会改变方向. 之前给出的简单的 $E(3)$ 等变 GNN 没有考虑到这一点.

2018)), 这两种卷积模型都在点云上运行.*Fuchs et al. (2020)* 的 SE(3)-Transformer 将该框架扩展到图形域, 使用了注意力层而不是卷积层. 此外, 虽然我们的讨论集中在 *Satorras et al. (2021)* 的特例解上, 但我们注意到, 对图的旋转或平移等变预测的动机在历史上已经在其他领域得到了探索, 包括诸如用于点云的动态图 CNN(Dynamic Graph CNN (*Wang et al., 2019b*)) 和用于量子化学的高效消息传递模型的架构, 例如 SchNet (*Schütt et al., 2018*) 和 DimeNet (*Klicpera et al., 2020*).

正则表示 虽然不可约表示的方法很有吸引力, 但它需要直接推理底层的群表示, 这可能很繁琐, 并且只适用于完备群. 正则表示方法更为普遍, 但会带来额外的计算负担——为了精确的等变, 它们需要存储所有群元素的潜在特征嵌入的副本.

事实上, 这种方法是由我们在前面几节中介绍的群卷积神经网络开创的.

这个领域一个有前途的方法是通过指数和对数映射的定义来观察李群 (Lie groups) 的等变, 并有望在各种对称群之间快速原型化. 虽然李群不在本节讨论范围内, 但我们请读者参考这一方向的两个最近成功的例子: *LieConv Finzi et al. (2020)* 和 *LieTransformer Hutchinson et al. (2020)*.

本节中涉及的方法代表了在几何图形上处理数据的流行方法, 这种方法明显等同于基础几何图形. 如 4.6 节所述, 网格 (meshes) 是几何图形的一个特殊例子, 可以理解为连续曲面的离散. 接下来我们将研究特定网格的等变神经网络.

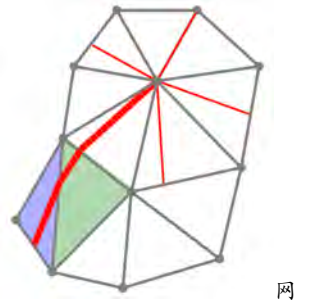
5.6 内在网格卷积神经网络

网格, 尤其是三角形网格, 是计算机图形学的“面包和黄油”, 也可能是建模 3D 对象的最常见方式. 深度学习的显著成功, 特别是计算机视觉中的中枢神经系统, 导致了 2010 年代中期图形和几何处理社区对构建网格数据的类似架构的浓厚兴趣.

测地线面片的例子. 为了使得到的网格成为拓扑圆盘, 其半径 R 必须小于内射半径 (injectivity radius).

测地线面片 大多数用于网格深度学习的体系结构通过离散化或近似指数映射并在三角平面的坐标系中表示滤波器来实现公式 (21) 的卷积滤波器. 跟踪测地线 $\gamma : [0, T] \rightarrow \Omega$, 从 $u = \gamma(0)$ 到领域点 $v = \gamma(T)$, 定义测地线极坐标 (geodesic polar coordinates) $(r(u, v), \vartheta(u, v))$ 的局部系统, 其中 r 是 u 和 v 之间的测地线距离 (测地线 γ 的长度), ϑ 是 $\gamma'(0)$ 和某个局部参考方向之间的角度. 这允许定义测地线面片 $x(u, r, \vartheta) = x(\exp_u \tilde{\omega}(r, \vartheta))$, 其中 $\tilde{\omega}_u : [0, R] \times [0, 2\pi) \rightarrow T_u\Omega$ 是局部极坐标系.

在离散为网格的曲面上, 测地线是穿过三角形面的折线. 传统上, 测地线是用快速行进 (Fast Marching) 算法 [Kimmel and Sethian \(1998\)](#) 计算的, 它是在介质中波传播的物理模型中遇到的非线性偏微分方程 (eikonal equation) 的有效数值近似. 该方案由 [Kokkinos et al. \(2012\)](#) 改编用于计算局部测地线面片, 后来由 [Masci et al. \(2015\)](#) 重新用于构建测地线 CNNs (Geodesic CNNs), 这是网格上第一个内在的 (intrinsic) 类似 CNN 的架构.



网格上离散测地线的构造.

各向同性滤波器 重要的是, 在定义中, 在参考方向和面片方向的选择上, 面片具有模糊性. 这正是量具选择的模糊性, 我们的局部坐标系被定义为任意旋转 (或角度坐标的移动, $x(u, r, \vartheta + \vartheta_0)$), 这在每个节点上都可能不同. 也许最直接的解决方案是使用 $\theta(r)$ 形式的各向同性滤波器, 该滤波器对相邻特征进行方向无关的聚集,

$$(x \star \theta)(u) = \int_0^R \int_0^{2\pi} x(u, r, \vartheta) \theta(r) dr d\vartheta.$$

第 4.4-4.6 节中讨论的谱滤波器属于这一类: 它们基于各向同性的拉普拉斯算子. 然而, 这种方法丢弃了重要的方向信息, 并且可能无法提取类似边的特征.

固定规范 我们已经在第 4.4 节中提到了另一种方法, 即固定一些规范. [Monti et al. \(2017\)](#) 使用了主曲率方向: 虽然这种选择不是内在的, 可能在平坦点 (曲率消失的地方) 或均匀曲率 (如在完美的球体上) 模糊不清, 但作者表明,

这对于处理近似分段刚性的可变形人体形状是合理的. 后来的工作, 如 [Melzi et al. \(2019\)](#), 显示了网格上的规的可靠的内在结构, 计算为内在函数的 (内在) 梯度. 虽然这样的切场可能具有奇异性 (即, 在某些点消失), 但是整个过程对于噪声和重网格化是非常鲁棒的.

角度池化 [Masci et al. \(2015\)](#) 使用了另一种方法, 称为角度最大池化 (angular max pooling). 在这种情况下, 滤波器 $\theta(r, \vartheta)$ 是各向异性的, 但它与函数的匹配是在所有可能的旋转上执行的, 然后这些旋转被聚合:

$$(x \star \theta)(u) = \max_{\vartheta_0 \in [0, 2\pi)} \int_0^R \int_0^{2\pi} x(u, r, \vartheta) \theta(r, \vartheta + \vartheta_0) dr d\vartheta.$$

从概念上来说, 这可以被可视化为将测地线面片与旋转滤波器相关联, 并收集最强的响应.

在网格上, 连续积分可以使用称为面片算子 (patch operators) 的结构来离散化 ([Masci et al., 2015](#)). 在节点 u 周围的测地线面片中, 在局部极坐标中表示为 (r_{uv}, ϑ_{uv}) 的相邻节点 \mathcal{N}_u 由一组加权函数 $w_1(r, \vartheta), \dots, w_K(r, \vartheta)$ 加权聚合 (如图 18 所示, 作为 “软像素”).

$$(x \star \theta)_u = \frac{\sum_{k=1}^K w_k \sum_{v \in \mathcal{N}_u} (r_{uv}, \vartheta_{uv}) x_v \theta_k}{\sum_{k=1}^K w_k \sum_{v \in \mathcal{N}_u} (r_{uv}, \vartheta_{uv}) \theta_k}$$

(这里 $\theta_1, \dots, \theta_K$ 是滤波器的可学习系数). 多通道功能是按通道处理的, 有一系列合适的滤波器. [Masci et al. \(2015\)](#); [Boscaini et al. \(2016a\)](#) 使用预定义的加权函数 w , 而 [Monti et al. \(2017\)](#) 进一步允许它们是可学习的.

规范等变滤波器 各向同性滤波器和角度最大化池化都导出对规范变换不变 (invariant) 的特征; 它们根据平凡表示 $\rho(\mathbf{g}) = 1$ (其中 $\mathbf{g} \in \text{SO}(2)$ 是局部坐标框架的旋转) 进行变换. 这一观点启发了 [Cohen et al. \(2019\)](#); [de Haan et al. \(2020\)](#) 提出的另一种方法, 在第 4.5 节中讨论过, 其中由网络计算的特征与结构群 \mathfrak{G} 的任意表示 ρ 相关联 (例如, $\text{SO}(2)$ 和 $\text{O}(2)$ 分别是坐标框架的旋转和旋转 + 镜像). 切向量根据标准表示 $\rho(\mathbf{g}) = \mathbf{g}$ 进行变换. 作为另一

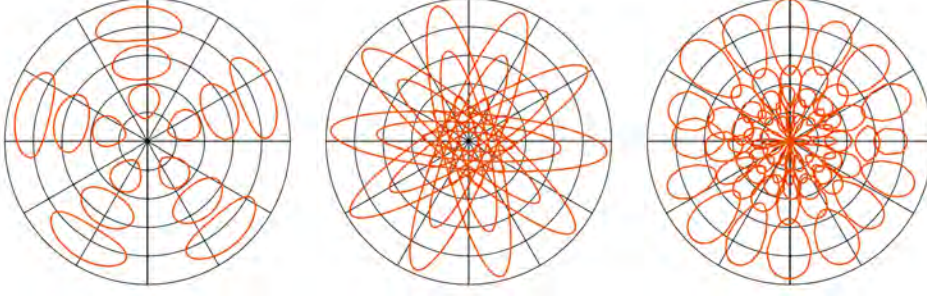


图 18: 从左到右: 在测地线 CNN (Geodesic CNN (Masci et al., 2015))、各向异性 CNN (Anisotropic CNN (Boscaini et al., 2016b)) 和 MoNet ((Monti et al., 2017)) 中使用的面片算子示例, 权重函数 $w_k(r, \vartheta)$ 显示为红色。

个示例, 通过匹配同一滤波器的 n 个旋转副本获得的特征向量, 在规范旋转下通过循环移位进行变换; 这就是循环群 C_n 的正则表示。

正如 4.5 节所讨论的, 当处理这样的几何特征 (与非平凡表示相关联) 时, 我们必须在应用滤波器之前首先将它们平行传输到相同的向量空间。在网格上, 这可以通过 de Haan et al. (2020) 描述的消息传递机制来实现。设 $\mathbf{x}_u \in \mathbb{R}^d$ 为网格节点 u 处的 d 维输入特征, 该特征可用相应于 u 处的规范 (任意) 来表示, 并假设根据 $\mathfrak{G} = \text{SO}(2)$ 在规范旋转下的表示 ρ_{in} 来变换。类似地, 网格卷积的输出特征是 d' 维的, 并且应该根据 ρ_{out} 进行变换 (ρ_{out} 可以由网络设计者随意选择)。

与图神经网络类似, 我们可以通过从 u 的邻居 \mathcal{N}_u (以及从 u 本身) 向 u 发送消息来实现网格上的规范等变卷积 (23);

$$\mathbf{h}_u = \Theta_{\text{self}} \mathbf{x}_u + \sum_{v \in \mathcal{N}_u} \Theta_{\text{neigh}}(\vartheta_{uv}) \rho(\mathbf{g}_{v \rightarrow u}) \mathbf{x}_v, \quad (37)$$

其中 $\Theta_{\text{self}}, \Theta_{\text{neigh}}(\vartheta_{uv}) \in \mathbb{R}^{d' \times d}$ 学习好的滤波器矩阵。结构群元素 $\mathbf{g}_{v \rightarrow u} \in \text{SO}(2)$ 表示从 v 到 u 的平行传输效果, 相对于 u 和 v 处的规范表示, 可以为每个网格预先计算。它的作用由变换矩阵 $\rho(\mathbf{g}_{v \rightarrow u}) \in \mathbb{R}^{d \times d}$ 编码。矩阵 $\Theta_{\text{neigh}}(\vartheta_{uv})$ 取决于邻居 v 相对于参考方向 (例如, 帧的第一轴) 的角度 ϑ_{uv} , 所以这个核是各向异性的: 不同的邻居被不同地对待。

请注意, d 是特征尺寸, 不一定等于 2, 网格的维数。

如第4.5节所述, 为了使 $\mathbf{h}(u)$ 成为一个定义明确的计量量, 在规范变换下, 它应该变换为 $\mathbf{h}(u) \mapsto \rho_{\text{out}}(\mathbf{g}^{-1}(u))\mathbf{h}(u)$. 对于所有 $\vartheta \in \text{SO}(2)$, 当 $\Theta_{\text{self}}\rho_{\text{in}}(\vartheta) = \rho_{\text{out}}(\vartheta)\Theta_{\text{self}}$, 代表所有 $\vartheta \in \text{SO}(2)$ 和 $\Theta_{\text{neigh}}(\vartheta_{uv}-\vartheta)\rho_{\text{in}}(\vartheta) = \rho_{\text{out}}(\vartheta)\Theta_{\text{neigh}}(\vartheta_{uv})$ 时, 情况就满足. 由于这些约束是线性的, 满足这些约束的矩阵 Θ_{self} 和矩阵值函数 Θ_{neigh} 的空间是线性子空间, 因此我们可以将它们参数化为具有可学习系数的基核的线性组合: $\Theta_{\text{self}} = \sum_i \alpha_i \Theta_{\text{self}}^i$ 和 $\Theta_{\text{neigh}} = \sum_i \beta_i \Theta_{\text{neigh}}^i$.

5.7 递归神经网络

到目前为止, 我们的讨论一直假设输入在给定的域中是唯一空间 (spatial) 的输入. 然而, 在许多常见的用例中, 输入也可以被认为是顺序的 (例如, 视频、文本或语音). 在这种情况下, 我们假设输入由任意多的步骤 (steps) 组成, 其中在每个步骤 t , 我们得到一个输入信号, 我们将其表示为 $\mathbf{X}^{(t)} \in \mathcal{X}(\Omega^{(t)})$.

该域是静态还是动态取决于时间尺度 (time scales): 例如,

道路网络确实随着时间而变化 (随着新道路的修建和旧道路的拆除), 但与交通流量相比明显较慢. 类似地, 在社交网络中, 参与度的变化 (例如推特用户转发推文) 发生的频率比下图中的变化高得多.

虽然一般来说, 该域可以随着其上的信号一起随时间演变, 但通常假设该域在所有 t 上保持固定, 即 $\Omega^{(t)} = \Omega$. 在这里, 我们将专门关注这种情况, 但请注意, 例外是常见的. 社交网络是一个例子, 人们经常不得不考虑随着时间的推移而变化的领域, 因为新的链接经常被创建和删除. 该设置中的域通常被称为动态图 (dynamic graph) (Xu et al., 2020a; Rossi et al., 2020).

通常, 单个的 $\mathbf{X}^{(t)}$ 输入会表现出有用的对称性, 因此可能会被我们之前讨论的任何体系结构所忽略. 常见的例子有: 视频 (Ω 是固定网格, 信号是帧序列); 功能磁共振成像扫描 (Ω 是一个代表大脑皮层几何形状的固定网格, 不同的区域在不同的时间被激活, 作为对呈现的刺激的反应); 和交通流网络 (Ω 是表示道路网络的固定图形, 例如在其上记录了各个节点的平均交通速度).

让我们假设一个编码器函数 $f(\mathbf{X}^{(t)})$ 在适合问题的粒度级别上提供潜在表示, 并尊重输入域的对称性. 例如, 考虑处理视频帧: 也就是说, 在每个时间步长, 我们得到一个格网结构的输入 (grid-structured input), 表示为 $n \times d$ 矩阵 $\mathbf{X}^{(t)}$, 其中 n 是像素数 (时间固定), d 是输入通道数 (例如, 对于 RGB 在我们的例子中, 我们失去一般性; 可以对例如时空图上的节点级输出进行等价分析; 唯一的区别在于编码器 f 的选择 (它将是置换等变 GNN).

帧, $d = 3$). 此外, 我们对整个帧级别的分析感兴趣, 在这种情况下, 将 f 实现为平移不变 CNN 是合适的, 在时间步长 t 输出帧的 k 维表示 $\mathbf{z}^{(t)} = f(\mathbf{X}^{(t)})$.

我们现在剩下的任务是在所有步骤中适当地汇总 (summarising) 一系列向量 $\mathbf{z}^{(t)}$. 一种规范的方法是使用递归神经网络 (RNN), 以相对输入的时序方式动态地 (dynamically) 聚合该信息, 并且还容易接收在线生成新的数据点. 我们将在这里展示的是, RNNs 本身是一种有趣的几何架构, 因为它们

请注意, $\mathbf{z}^{(t)}$ 向量可以被视为时序格网 (temporal grid) 上的点: 因此, 用 CNN 处理它们在某些情况下也是可行的. Transformers 也是越来越受欢迎的处理一般顺序输入的模式.

简单递归神经网络 在每一步, 递归神经网络计算所有输入步骤的 m 维汇总向量 $\mathbf{h}^{(t)}$, 直到包括 t . 这个 (部分) 汇总是根据当前步骤的特征和前一步的汇总, 通过一个共享的更新函数 $R: \mathbb{R}^k \times \mathbb{R}^m \rightarrow \mathbb{R}^m$, 如下所示 (见图 19):

$$\mathbf{h}^{(t)} = R(\mathbf{z}^{(t)}, \mathbf{h}^{(t-1)}) \quad (38)$$

因为 $\mathbf{z}^{(t)}$ 和 $\mathbf{h}^{(t-1)}$ 都是平面 (flat) 向量表示, 所以 R 最容易表示为单个完全连接的神经网络层 (通常称为简单递归神经网络 (SimpleRNNs); 见 *see Elman (1990); Jordan (1997)*):

$$\mathbf{h}^{(t)} = \sigma(\mathbf{W}\mathbf{z}^{(t)} + \mathbf{U}\mathbf{h}^{(t-1)} + \mathbf{b}) \quad (39)$$

其中 $\mathbf{W} \in \mathbb{R}^{k \times m}$, $\mathbf{U} \in \mathbb{R}^{m \times m}$ 和 $\mathbf{b} \in \mathbb{R}^m$ 可学习参数, σ 为激活函数. 虽然这在网络的计算图中引入了循环, 但在实践中, 网络被展开适当的步数, 允许通过时间反向传播 (*Robinson and Fallside, 1987; Werbos, 1988; Mozer, 1989*) 将被应用.

尽管名字很简单, 但它们非常有表现力. 例如, *Siegelmann and Sontag (1995)* 表明, 这种模型是图灵完备的 (Turing-complete), 这意味着它们可能代表我们可能在计算机上执行的任何计算.

然后下游任务可以适当地利用汇总向量——如果在序列的每一步都需要预测, 则可以将共享预测器单独应用于每个 $\mathbf{h}^{(t)}$. 为了对整个序列进行分类, 通常将最后的汇总 $\mathbf{h}^{(T)}$ 传递给分类器. 这里, T 是序列的长度.

特别地, 初始汇总向量通常要么被设置为零向量, 即 $\mathbf{h}^{(0)} = \mathbf{0}$, 要么被设置为可学习的. 分析初始汇总向量的设置方式也允许我们推导出由 RNNs 具有的一种有趣的平移等变 (translation equivariance) 形式.

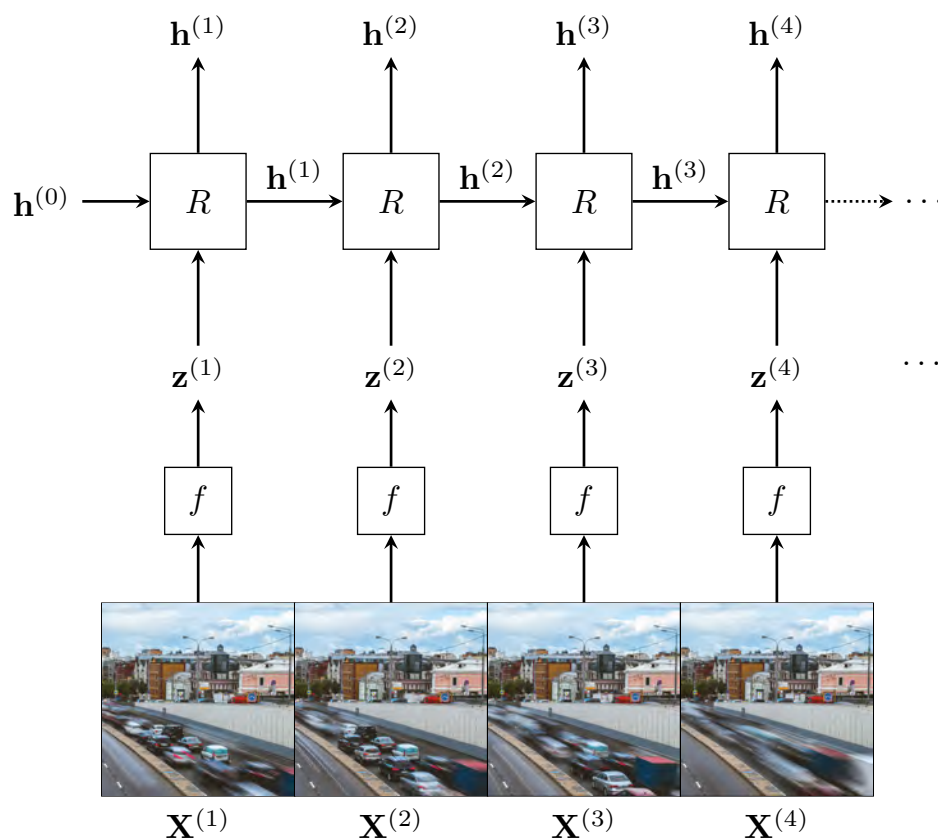


图 19: 使用 RNNs 处理视频输入. 每个输入视频帧 $\mathbf{X}^{(t)}$ 使用共享函数 f (例如, 平移不变 CNN) 处理成平面 (flat) 表示 $\mathbf{z}^{(t)}$. 然后, RNN 更新函数 R 在这些向量上迭代, 迭代地更新汇总 (summary) 向量 $\mathbf{h}^{(t)}$, 该向量汇总了直到并包括 $\mathbf{z}^{(t)}$ 的所有输入. 该计算以初始汇总向量 $\mathbf{h}^{(0)}$ 作为种子, 该向量可以是预先确定的或可学习的.

递归神经网络的平移等变 因为我们把单个的时间步长 t 解释为离散的时间步长 (discrete time-steps), 所以输入向量 $\mathbf{z}^{(t)}$ 可以被看作是时间步长上的一维格网. 虽然在这里尝试从 CNNs 扩展我们的平移等变分析可能是有吸引力的, 但它不能用平凡的方式得到.

Note that this construction is extendable to grids in higher dimensions, allowing us to, e.g., process signals living on images in a scanline fashion. Such a construction powered a popular series of models, such as the PixelRNN from van den Oord et al. (2016b).

为了了解原因, 让我们假设我们已经通过将序列左移一步产生了一个新的序列 $\mathbf{z}'^{(t)} = \mathbf{z}^{(t+1)}$. 尝试揭露 $\mathbf{h}'^{(t)} = \mathbf{h}^{(t+1)}$ 可能很有诱惑力, 正如人们对平移等变的期望; 然而, 这通常是不成立的. 考虑 $t = 1$; 直接应用和扩展更新函数, 我们可以得到以下内容:

$$\mathbf{h}'^{(1)} = R(\mathbf{z}'^{(1)}, \mathbf{h}^{(0)}) = R(\mathbf{z}^{(2)}, \mathbf{h}^{(0)}) \quad (40)$$

$$\mathbf{h}^{(2)} = R(\mathbf{z}^{(2)}, \mathbf{h}^{(1)}) = R(\mathbf{z}^{(2)}, R(\mathbf{z}^{(1)}, \mathbf{h}^{(0)})) \quad (41)$$

因此, 除非我们能够保证 $\mathbf{h}^{(0)} = R(\mathbf{z}^{(1)}, \mathbf{h}^{(0)})$, 否则我们将无法恢复平移等变, 然后可以对步骤 $t > 1$ 进行类似的分析.

幸运的是, 通过对我们如何表示 \mathbf{z} 的稍微重构, 选择一个合适的 R , 有可能满足上面的等式, 这表示一种移位 (shifts) 等变的 RNNs. 我们的问题主要是边界条件 boundary conditions 之一: 上面的等式包括 $\mathbf{z}^{(1)}$, 我们的左移位操作破坏了. 为了抽象出这个问题, 我们将关注 RNN 如何处理一个适当的左填充 (left-padded) 序列 $\bar{\mathbf{z}}^{(t)}$, 定义如下:

$$\bar{\mathbf{z}}^{(t)} = \begin{cases} \mathbf{0} & t \leq t' \\ \mathbf{z}^{(t-t')} & t > t' \end{cases}$$

这样的序列现在允许向左移位多达 t' 步, 而不破坏任何原始输入元素. 此外, 请注意, 如果我们使用不同于 $\mathbf{0}$ 的填充向量, 将需要等价分析.

我们现在可以再次分析 RNN 在 $\bar{\mathbf{z}}^{(t)}$ 的左移位版本上的运算, 我们用 $\bar{\mathbf{z}}'^{(t)} = \bar{\mathbf{z}}^{(t+1)}$ 表示, 正如我们在等式 40-41 中所做的:

$$\mathbf{h}'^{(1)} = R(\bar{\mathbf{z}}'^{(1)}, \mathbf{h}^{(0)}) = R(\bar{\mathbf{z}}^{(2)}, \mathbf{h}^{(0)})$$

$$\mathbf{h}^{(2)} = R(\bar{\mathbf{z}}^{(2)}, \mathbf{h}^{(1)}) = R(\bar{\mathbf{z}}^{(2)}, R(\bar{\mathbf{z}}^{(1)}, \mathbf{h}^{(0)})) = R(\bar{\mathbf{z}}^{(2)}, R(\mathbf{0}, \mathbf{h}^{(0)}))$$

其中只要 $t' \geq 1$, 即只要应用了任何填充, 替换 $\mathbf{z}^{(1)} = \mathbf{0}$ 就成立. 现在, 只要 $\mathbf{h}^{(0)} = R(\mathbf{0}, \mathbf{h}^{(0)})$, 我们可以通过一步 ($\mathbf{h}^{(t)} = \mathbf{h}^{(t+1)}$) 保证左移位等变.

换句话说, $\mathbf{h}^{(0)}$ 必须选择为函数 $\gamma(\mathbf{h}) = R(\mathbf{0}, \mathbf{h})$ 的不动点 (fixed point). 如果更新函数 R 选择方便, 那么不仅可以保证这类不动点的存在, 甚至可以通过迭代 R 直到收敛; 举例如下:

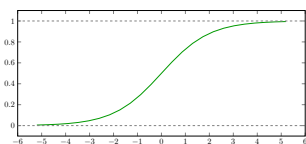
$$\mathbf{h}_0 = \mathbf{0} \quad \mathbf{h}_{k+1} = \gamma(\mathbf{h}_k), \quad (42)$$

其中指数 k 指的是在我们的计算中 R 的迭代, 与表示 RNN 时间步长的指数 (t) 相反. 如果我们选择 R 使得 γ 是压缩映射 (contraction mapping), 这样的迭代确实会收敛到唯一的不动点. 因此, 我们可以迭代方程 (42), 直到 $\mathbf{h}_{k+1} = \mathbf{h}_k$, 我们可以设置 $\mathbf{h}^{(0)} = \mathbf{h}_k$. 请注意, 这种计算相当于用“足够多的”零向量对序列进行左填充.

压缩是函数 $\gamma: \mathcal{X} \rightarrow \mathcal{X}$, 使得在 \mathcal{X} 上的某个范数 $\|\cdot\|$, 应用 γ 压缩点之间的距离: 对于所有的 $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, 以及某些 $q \in [0, 1)$, 它认为 $\|\gamma(\mathbf{x}) - \gamma(\mathbf{y})\| \leq q\|\mathbf{x} - \mathbf{y}\|$. 迭代这样的函数必然会收敛到一个唯一的不动点, 这是巴拿赫不动点定理的直接结果 (Banach's Fixed Point Theorem (Banach, 1922)).

递归神经网络的深度 堆叠多个神经网络也很容易——只需将 $\mathbf{h}^{(t)}$ 向量用作第二个 RNN 的输入序列. 这种架构通常被称为“深 RNN ”, 这可能会产生误导. 实际上, 由于循环操作的重复应用, 甚至单个 RNN “层”的深度也等于输入步骤的数量.

在优化神经网络时, 这通常会引入独特的挑战性学习动态, 因为每个训练示例都会对更新网络的共享 (shared) 参数进行多次梯度更新. 在这里, 我们将重点关注最突出的问题——消失 (vanishing) 和爆炸 (exploding) 梯度 (Bengio et al., 1994), 这在 $RNNs$ 中尤其成问题, 因为它们的深度和参数共享. 此外, 它还刺激了一些对神经网络最有影响力的研究. 为了获得更详细的概述, 我们请读者参考 Pascanu et al. (2013), 他们非常详细地研究了 $RNNs$ 的训练动力学, 并从多种角度揭露了这些挑战: 分析、几何和动力系统的视角.



这种激活的例子包括 *logistic* 函数 $\sigma(x) = \frac{1}{1+\exp(-x)}$, 以及双曲正切 (hyperbolic tangent), $\sigma(x) = \tanh x$. 他们被称为 *sigmoidal*, 因为他们的图形是截然不同的 S 形.

为了说明消失梯度, 考虑一个带有 σ 激活函数的 *SimpleRNNs*, 它的导数幅度 $|\sigma'|$ 总是在 0 和 1 之间. 将许多这样的值相乘会导致梯度迅速趋于零, 这意味着输入序列中的早期变化可能根本不能影响网络参数的更新.

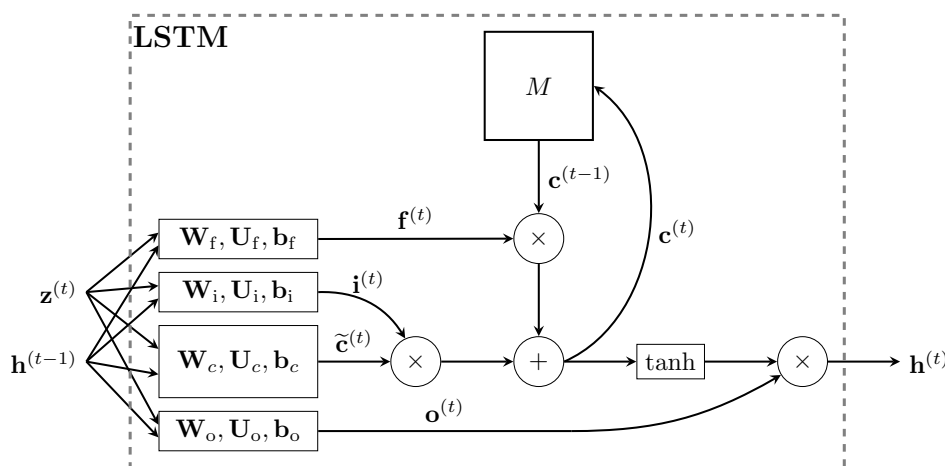


图 20: 长短期记忆的数据流 (LSTM), 其组成部分和记忆单元 (M) 清楚地突出. 基于当前输入 $\mathbf{z}^{(t)}$ 、先前汇总 $\mathbf{h}^{(t-1)}$ 和先前单元状态 $\mathbf{c}^{(t-1)}$, LSTM 预测更新的单元状态 $\mathbf{c}^{(t)}$ 和汇总 $\mathbf{h}^{(t)}$.

例如, 考虑下一个单词预测任务 (在预测键盘中常见), 并且输入文本 “Petar is Serbian. He was born on ...[long paragraph] ...Petar currently lives in _____”. 在这里, 预测下一个单词是 “Serbia” 可能只能通过考虑段落的开头来合理地得出结论——但是当梯度到达这个输入步骤时, 它们可能已经消失了, 使得从这样的例子中学习变得非常困难.

深度前馈神经网络也受到消失梯度问题的困扰, 直到 $ReLU$ 激活 (其梯度恰好等于零或一——从而解决了消失梯度问题) 的发明. 然而, 在 $RNNs$ 中, 使用 $ReLU$ s 可能很容易导致梯度爆炸 (exploding), 因为更新函数的输出空间现在是无界的 (unbounded), 梯度下降将为每个输入步骤更新一次单元格, 快速建立更新的规模. 历史上, 消失的梯度现象很早就被认为是使用递归网络的一个重大障碍. 解决这个问题推动了更复杂的 RNN 层的发展, 我们将在下面描述.

5.8 长短期记忆网络

显著降低 *RNNs* 中消失梯度影响的一项关键发明是门控机制 (gating mechanisms), 它允许网络以数据驱动的方式选择性地覆盖信息. 这些门控神经网络的突出例子包括长短期记忆 (*LSTM*; [Hochreiter and Schmidhuber \(1997\)](#)) 和门控递归单位 (*GRU*; [Cho et al. \(2014\)](#)). 在这里, 我们将主要讨论 *LSTM*——具体来说, [Graves \(2013\)](#) 提出的变体——以说明这种模型的操作. *LSTMs* 的概念很容易迁移到其他门控 *RNNs*.

在本节中, 参考图 20 可能会有所帮助, 它说明了我们将在文本中讨论的所有 *LSTM* 操作.

LSTM 通过引入记忆单元 (memory cell) 来增加递归计算, 该记忆单元存储在计算步骤之间保留的单元状态 (cell state) 向量 $\mathbf{c}^{(t)} \in \mathbb{R}^m$. *LSTM* 直接基于 $\mathbf{c}^{(t)}$ 计算汇总向量 $\mathbf{h}^{(t)}$, 而 $\mathbf{c}^{(t)}$ 又通过 $\mathbf{z}^{(t)}$, $\mathbf{h}^{(t-1)}$ 和 $\mathbf{c}^{(t-1)}$ 来计算. 重要的是, 基于 $\mathbf{z}^{(t)}$ 和 $\mathbf{h}^{(t-1)}$, 单元没有被完全覆盖, 这将使网络面临与 *SimpleRNNs* 相同的问题. 取而代之的是, 可以保留一定数量的先前单元状态——并且这种情况发生的比例是从数据中明确获知的.

就像在 *SimpleRNN* 中一样, 我们通过在当前输入步骤和之前的汇总向量上使用单个完全连接的神经网络层来计算特征:

注意我们这里已经把激活函数设置为 \tanh , 由于 *LSTMs* 的设计是为了改善消失的梯度问题, 现在适合使用 *sigmoidal* 激活函数.

$$\tilde{\mathbf{c}}^{(t)} = \tanh(\mathbf{W}_c \mathbf{z}^{(t)} + \mathbf{U}_c \mathbf{h}^{(t-1)} + \mathbf{b}_c) \quad (43)$$

但是, 如上所述, 我们不允许所有的向量进入“单元”——因此我们称之为候选 (candidate) 特征向量, 并将其表示为 $\tilde{\mathbf{c}}^{(t)}$. 相反, *LSTM* 直接学习选通向量 (gating vectors) (范围为 $[0, 1]$ 的实值向量), 并决定应该允许多少信号进入、退出和覆盖记忆单元.

计算三个这样的门, 全部基于 $\mathbf{z}^{(t)}$ 和 $\mathbf{h}^{(t-1)}$: 输入门 $\mathbf{i}^{(t)}$, 它计算允许进入单元的候选向量的比例; 忘记门 $\mathbf{f}^{(t)}$ 计算要保留的先前单元状态的比例, 输出门 $\mathbf{o}^{(t)}$ 计算要用于最终汇总向量的新单元状态的比例. 一般来说, 所有这些门都是使用单个完全连接的层导出的, 采用 *logistic sigmoid* 激活函数

$\text{logistic}(x) = \frac{1}{1+\exp(-x)}$, 以保证输出在 $[0, 1]$ 范围内:

$$\mathbf{i}^{(t)} = \text{logistic}(\mathbf{W}_i \mathbf{z}^{(t)} + \mathbf{U}_i \mathbf{h}^{(t-1)} + \mathbf{b}_i) \quad (44)$$

$$\mathbf{f}^{(t)} = \text{logistic}(\mathbf{W}_f \mathbf{z}^{(t)} + \mathbf{U}_f \mathbf{h}^{(t-1)} + \mathbf{b}_f) \quad (45)$$

$$\mathbf{o}^{(t)} = \text{logistic}(\mathbf{W}_o \mathbf{z}^{(t)} + \mathbf{U}_o \mathbf{h}^{(t-1)} + \mathbf{b}_o) \quad (46)$$

请注意, 三个门本身就是向量, 即 $\mathbf{i}^{(t)}, \mathbf{f}^{(t)}, \mathbf{o}^{(t)} \in [0, 1]^m$. 这允许它们控制 m 个维度中的每一个维度允许通过门的量.

最后, 这些门被适当地应用于解码新的单元状态 $\mathbf{c}^{(t)}$, 然后由输出门对其进行调制以产生汇总向量 $\mathbf{h}^{(t)}$, 如下所示:

$$\mathbf{c}^{(t)} = \mathbf{i}^{(t)} \odot \tilde{\mathbf{c}}^{(t)} + \mathbf{f}^{(t)} \odot \mathbf{c}^{(t-1)} \quad (47)$$

$$\mathbf{h}^{(t)} = \mathbf{o}^{(t)} \odot \tanh(\mathbf{c}^{(t)}) \quad (48)$$

其中 \odot 是逐元素向量乘法. 一起应用, 等式 (43)–(48) 完全指定了 *LSTM* 的更新规则, 现在也考虑了单元向量 $\mathbf{c}^{(t)}$:

$$(\mathbf{h}^{(t)}, \mathbf{c}^{(t)}) = R(\mathbf{z}^{(t)}, (\mathbf{h}^{(t-1)}, \mathbf{c}^{(t-1)}))$$

这仍然与等式 (38) 中的 *RNN* 更新蓝图兼容; 简单地认为汇总向量是 $\mathbf{h}^{(t)}$ 和 $\mathbf{c}^{(t)}$ 的连接; 有时用 $\mathbf{h}^{(t)} \parallel \mathbf{c}^{(t)}$ 表示.

请注意, 由于 $\mathbf{f}^{(t)}$ 的值是从 $\mathbf{z}^{(t)}$ 和 $\mathbf{h}^{(t-1)}$ 导出的, 因此可以直接从数据中学习, *LSTM* 有效地学会了如何适当地忘记过去的经历. 事实上, $\mathbf{f}^{(t)}$ 的值直接出现在所有 *LSTM* 参数 $(\mathbf{W}_*, \mathbf{U}_*, \mathbf{b}_*)$ 的反向传播更新中, 允许网络以数据驱动的方式明确控制梯度在时间步长上的消失程度.

除了正面解决消失的梯度问题, 门控 *RNNs* 还释放了一种非常有用的对时间规整 (time-warping) 的不变性, 这是 *SimpleRNNs* 无法实现的.

门控递归神经的时间规整不变性 我们将首先说明, 在连续时间场景中, 规整时间 (warp time) 意味着什么, 以及为了实现对这种变换的不变性, 需要一个递归模型. 我们的阐述将在很大程度上遵循 [Tallec and Ollivier \(2018\)](#) 的工作, 他们最初描述了这一现象——事实上, 他们是第一批从不变性的角度实际研究 *RNNs* 的人.

我们专注于连续的场景, 因为在那里更容易推理时间的操控.

让我们假设一个连续的时域信号 $z(t)$, 我们希望在其上应用 *RNN*. 对齐 *RNN* 的汇总向量 $\mathbf{h}^{(t)}$ 的离散时间计算. 用连续域中的一个类似物 $h(t)$, 我们

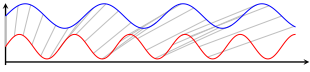
我们将使用 $h(t)$ 表示时间 t 的连续信号, $\mathbf{h}^{(t)}$ 表示时间步长 t 的离散信号.

将观察它的线性泰勒展开:

$$h(t + \delta) \approx h(t) + \delta \frac{dh(t)}{dt} \quad (49)$$

同时, 假定 $\delta = 1$, 我们恢复 $h(t)$ 和 $h(t+1)$ 之间的关系, 这正是 *RNN* 更新函数 (等式 38) 所计算的. 即, *RNN* 更新函数满足以下微分方程:

$$\frac{dh(t)}{dt} = h(t+1) - h(t) = R(z(t+1), h(t)) - h(t) \quad (50)$$



这种规整操作可以很简单, 例如时间重新缩放; 例如, $\tau(t) = 0.7t$ (如上所示), 在离散设置中, 这相当于每隔 ~ 1.43 步接收新的输入. 然而, 它也允许宽范围的可变 (variably-changing) 采样速率, 例如, 采样可以在整个领域中自由加速或减速.

我们希望 *RNN* 能够适应信号采样的方式 (例如, 通过改变测量的时间单位), 以解决其中的任何缺陷或不规则性. 在形式上, 我们将时间规整操作 $\tau: \mathbb{R}^+ \rightarrow \mathbb{R}^+$ 表示为时间之间任何单调递增的可微映射. 选择符号 τ 是因为时间规整表示时间的自同构 (automorphism).

此外, 我们声明一类模型对于时间规整是不变的 (invariant), 如果对于该类的任何模型和任何这样的 τ , 存在来自该类的另一个 (可能相同的) 模型, 该模型以与原始模型在非规整情况下相同的方式处理规整的数据.

这是一个潜在的非常有用的属性. 如果我们有一个能够很好地模拟短期相关性的 *RNN* 类, 并且我们还可以证明这个类对时间规整是不变的, 那么我们知道有可能以一种同样有效地捕捉长期相关性的方式来训练这样一个模型 (因为它们对应于具有短期相关性的信号的时间膨胀规整 (time dilation warping)). 正如我们将很快看到的那样, 像 *LSTM* 这样的门控 *RNN* 模型被提出来模拟长程相关性并不是巧合. 实现时间规整不变性与门控机制的存在密切相关, 例如 *LSTMs* 的输入/忘记/输出门.

当时间被 τ 规整时, *RNN* 在时间 t 观察到的信号是 $z(\tau(t))$, 并且为了保持对这种规整的不变性, 它应该预测等变规整的汇总函数 $h(\tau(t))$. 再次使用泰勒展开参数, 我们导出了规整时间的公式 50, *RNN* 更新 R 应满足:

$$\frac{dh(\tau(t))}{d\tau(t)} = R(z(\tau(t+1)), h(\tau(t))) - h(\tau(t)) \quad (51)$$

然而, 上述导数是相对于规整时间 $\tau(t)$ 计算的, 因此没有考虑原始信号. 为了使我们的模型明确地考虑规整变换, 我们需要区分关于 t 的规整汇总函数.

应用链式规则, 这产生以下微分方程:

$$\frac{dh(\tau(t))}{dt} = \frac{dh(\tau(t))}{d\tau(t)} \frac{d\tau(t)}{dt} = \frac{d\tau(t)}{dt} R(z(\tau(t+1)), h(\tau(t))) - \frac{d\tau(t)}{dt} h(\tau(t)) \quad (52)$$

同时, 为了使我们的 (连续时间) RNN 对于任何时间规整 (t) 保持不变, 它需要能够显式地表示导数 $\frac{d\tau(t)}{dt}$, 这是预先不知道的假设! 我们需要引入一个逼近这个导数的可学习函数 Γ . 例如, Γ 可以是考虑 $z(t+1)$ 和 $h(t)$ 并预测标量 (*scalar*) 输出的神经网络.

现在, 请注意, 从时间规整下的离散 RNN 模型的角度来看, 其输入 $\mathbf{z}^{(t)}$ 将对应于 $z(\tau(t))$, 其汇总 $\mathbf{h}^{(t)}$ 将对应于 $h(\tau(t))$. 为了获得 $\mathbf{h}^{(t)}$ 与 $\mathbf{h}^{(t+1)}$ 的所需关系, 以便保持对时间规整的不变性, 我们将使用 $h(\tau(t))$ 的一步泰勒展开:

$$h(\tau(t+\delta)) \approx h(\tau(t)) + \delta \frac{dh(\tau(t))}{dt}$$

同时, 再次假定 $\delta = 1$, 代入等式 52, 然后离散化

$$\begin{aligned} \mathbf{h}^{(t+1)} &= \mathbf{h}^{(t)} + \frac{d\tau(t)}{dt} R(\mathbf{z}^{(t+1)}, \mathbf{h}^{(t)}) - \frac{d\tau(t)}{dt} \mathbf{h}^{(t)} \\ &= \frac{d\tau(t)}{dt} R(\mathbf{z}^{(t+1)}, \mathbf{h}^{(t)}) + \left(1 - \frac{d\tau(t)}{dt}\right) \mathbf{h}^{(t)} \end{aligned}$$

最后, 我们用前面提到的可学习函数 Γ 交换 $\frac{d\tau(t)}{dt}$. 这给了我们时间规整不变 RNN 所需的形式:

$$\mathbf{h}^{(t+1)} = \Gamma(\mathbf{z}^{(t+1)}, \mathbf{h}^{(t)}) R(\mathbf{z}^{(t+1)}, \mathbf{h}^{(t)}) + (1 - \Gamma(\mathbf{z}^{(t+1)}, \mathbf{h}^{(t)})) \mathbf{h}^{(t)} \quad (53)$$

我们可以很快推导出 *SimpleRNNs* (等式 39) 不是时间规整不变的, 因为它们没有等式 53 中的第二项. 相反, 它们用 $R(\mathbf{z}^{(t+1)}, \mathbf{h}^{(t)})$ 完全覆盖 $\mathbf{h}^{(t)}$, 这对应于假设完全没有时间规整; $\frac{d\tau(t)}{dt} = 1$, 即 $\tau(t) = t$.

此外, 我们在连续时间 $RNNs$ 和基于 R 的离散 RNN 之间的联系建立在泰勒近似的准确性上, 只有当时间规整导数不太大, 即 $\frac{d\tau(t)}{dt} \lesssim 1$ 时, 泰勒近似才成立. 对此的直观解释是: 如果我们的时间规整操作 (*time warping operation*) 以使时间增量 $(t \rightarrow t+1)$ 足够大以至于中间数据变化没有被采样的方式压缩时间 (*contracts time*), 模型就永远不能希望以与原始输入相

同的方式处理时间规整的输入——它根本就不能访问相同的信息。相反，任何形式的时间膨胀 (dilations)(用离散术语来说，相当于在输入时间序列中散布零) 在我们的框架内都是完全允许的。

结合我们的单调递增 $\tau (\frac{d\tau(t)}{dt} > 0)$ 的要求，我们可以将 Γ 的输出空间限制为 $0 < \Gamma(\mathbf{z}^{(t+1)}, \mathbf{h}^{(t)}) < 1$ ，这激励我们为 Γ 使用逻辑 *sigmoid* 激活函数，例如：

$$\Gamma(\mathbf{z}^{(t+1)}, \mathbf{h}^{(t)}) = \text{logistic}(\mathbf{W}_\Gamma \mathbf{z}^{(t+1)} + \mathbf{U}_\Gamma \mathbf{h}^{(t)} + \mathbf{b}_\Gamma)$$

精确 (exactly) 匹配 *LSTM* 门控方程 (例如等式44)。主要区别在于，*LSTMs* 计算门控向量，而等式53暗示 Γ 应该输出标量。向量化门 (*Hochreiter, 1991*) 允许在 $h(t)$ 的每个维度上拟合不同的规整 (*warping*) 导数，允许在多个时间上同时推理。

这里值得停下来总结一下我们所做的工作。通过要求我们的 *RNN* 类对于 (非破坏性的) 时间规整是不变的，我们导出了它必须具有的必要形式 (等式53)，并且表明它精确地对应于门控类 *RNNs*。在这种观点下，门的主要作用是精确拟合规整变换的导数 $\frac{d\tau(t)}{dt}$ 。

类不变性 (class invariance) 的概念与我们先前研究的不变性有些不同。也就是说，一旦我们在 $\tau_1(t)$ 的时间规整输入上训练门控 *RNN*，我们通常不能将其零拍转移 (*zero-shot transfer*) 为被不同 $\tau_2(t)$ 规整的信号。相反，类不变性只能保证门控神经网络足够强大，能够以相同的方式拟合这两种信号，但可能具有非常不同的模型参数。也就是说，认识到有效的门控机制与拟合翘曲导数密切相关，可以为门控 *RNN* 优化提供有用的处方，正如我们现在简要展示的那样。

有可能进行零拍转移的一种情况是，假设第二次时间规整是第一次的时间重新缩放 ($\tau_2(t) = \alpha\tau_1(t)$)。将预先在 τ_1 上训练的门控 *RNN* 转换为由 τ_2 规整的信号需要重新调整门控： $\Gamma_2(\mathbf{z}^{(t+1)}, \mathbf{h}^{(t)}) = \alpha\Gamma_1(\mathbf{z}^{(t+1)}, \mathbf{h}^{(t)})$ 。 R 可以保留其参数 ($R_1 = R_2$)。

例如，我们可以经常假设，希望我们跟踪的信号中依存关系的范围将在 $[T_l, T_h]$ 时间步长内。

通过分析方程52的解析解，可以看出我们的选通 *RNN* 对 $\mathbf{h}^{(t)}$ 的特征遗忘时间与 $\frac{1}{\Gamma(\mathbf{z}^{(t+1)}, \mathbf{h}^{(t)})}$ 成正比。因此，我们希望我们的选通值在 $\left[\frac{1}{T_h}, \frac{1}{T_m}\right]$ 之间，以便有效地记住假设范围内的信息。

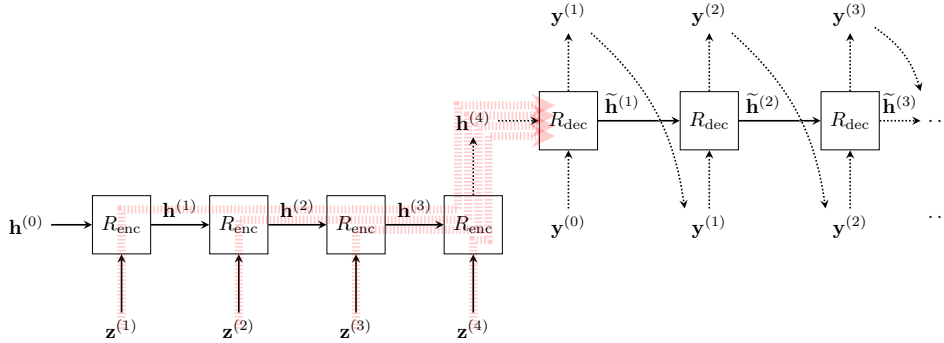


图 21: seq2seq 架构的一个典型例子是 RNN 编码器 R_{enc} 和 RNN 解码器 R_{dec} . 解码器以来自编码器的最终汇总向量 $\mathbf{h}^{(T)}$ 作为种子, 然后以自回归 (autoregressive) 方式进行: 在每一步, 来自前一步的预测输出被反馈作为 R_{dec} 的输入. 瓶颈问题也用红线表示: 汇总向量 $\mathbf{h}^{(T)}$ 被迫存储翻译输入序列的所有相关信息, 随着输入长度的增长, 这变得越来越具有挑战性.

此外, 如果我们假设 $\mathbf{z}^{(t)}$ 和 $\mathbf{h}^{(t)}$ 大致以零为中心——这是应用层归一化等变换的常见副产品 (Ba et al., 2016)——我们可以假设 $\mathbb{E}[\Gamma(\mathbf{z}^{(t+1)}, \mathbf{h}^{(t)})] \approx \text{logistic}(\mathbf{b}_\Gamma)$. 因此, 控制选通机制的偏置向量是控制有效选通值的非常有效的方法.

Gers and Schmidhuber (2000); Jozefowicz et al. (2015) 已经发现了这一观点, 他根据经验建议将 *LSTMs* 的遗忘门偏置初始化为一个恒定的正向量, 比如 1.

结合这两个观察结果, 我们得出结论, 通过初始化 $\mathbf{b}_\Gamma \sim -\log(\mathcal{U}(T_l, T_h) - 1)$ 可以获得适当范围的选通值, 其中 \mathcal{U} 是均匀实分布. 这样的建议被 Tallec and Ollivier (2018) 称为计时初始化 (chrono initialisation), 并已被经验证明可以改善门控 *RNNs* 的长期依赖性建模.

利用递归神经进行序列到序列学习 使用 *RNN* 支持的计算的一个突出的历史例子是序列到序列 (sequence-to-sequence) 的翻译任务, 例如自然语言的机器翻译. Sutskever et al. (2014) 的开创性工作通过传递汇总向量实现了这一点, $\mathbf{h}^{(T)}$ 作为解码器 *RNN* 的初始输入, *RNN* 块的输出作为下一步的输入给出.

这给汇总向量 $\mathbf{h}^{(T)}$ 带来了相当大的表征压力. 在深度学习的背景下, $\mathbf{h}^{(T)}$ 有



瓶颈效应最近在图形表示学习社区 (Alon and Yahav, 2020) 以及神经算法推理 (Cappart et al., 2021) 中受到了极大的关注。

时被称为瓶颈。它的固定容量必须足以表示整个输入序列的内容，以有助于生成相应序列的方式，同时还支持长度基本不同的输入序列 (图21)。

实际上，输出的不同步骤可能希望关注输入的不同部分，并且所有这样的选择很难通过瓶颈向量来表示。根据这一观察, Bahdanau et al. (2014) 提出了流行的递归注意力 (recurrent attention) 模型。在处理的每一步，查询向量由 RNN 生成；然后，这个查询向量与每个时间步长 $\mathbf{h}^{(t)}$ 的表示相互作用，主要是通过计算它们的加权和。这一模型开创了基于神经内容的注意力，并先于 Transformer 模型获得成功。

最后，虽然试图提出一种动态关注部分输入内容的软方法，但实质性的工作也让我们学会了用明确的 (explicit) 方法将注意力引向输入。一种强大的基于算法的方法是 Vinyals et al. (2015) 的指针网络 (pointer network)，该网络提出了一种简单的递归注意力修改模块，以允许指向可变大小的输入元素。这些发现随后被推广到 set2set 架构 (Vinyals et al., 2016)，该架构将 seq2seq 模型推广到无序集，由指针网络支持的 LSTMs 支持。

6 问题与应用

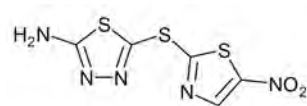
不变性和对称性在源自现实世界的数据中普遍存在。因此，毫不奇怪，21 世纪机器学习的一些最受欢迎的应用是作为几何深度学习的直接副产品出现的，也许有时没有完全意识到这一事实。我们想为读者提供一个概述——绝不是全面的——几何深度学习中有影响力的作品以及令人兴奋和有前途的新应用。我们的动机是双重的：展示五个几何领域共同出现的科学和工业问题的具体实例，并为进一步研究几何深度学习原则和体系结构提供额外的动机。

6.1 化学与药物设计

图形表示学习最有前途的应用之一是在计算化学和药物开发。传统药物是小分子，被设计成与某些目标分子（通常是蛋白质）化学连接（结合），以激活或破坏某些疾病相关的化学过程。不幸的是，药物开发是一个极其漫长和昂贵的过程：在撰写本文时，将一种新药推向市场通常需要十多年的时间，费用超过十亿美元。原因之一是许多药物在不同阶段失败的测试成本——不到 5% 的候选药物进入了最后阶段（例如，见 [Gaudelet et al. \(2020\)](#)）。

由于化学合成分子的空间非常大（估计在 10^{60} 左右），寻找具有正确结合特性的候选分子，如靶结合亲和力、低毒性、溶解性等。不能通过实验完成，而是采用虚拟或电子筛选（即使用计算技术来识别有希望的分子）。机器学习技术在这项任务中发挥着越来越重要的作用。[Stokes et al. \(2020\)](#) 最近展示了一个使用几何深度学习 (*Geometric Deep Learning*) 进行虚拟药物筛选的突出例子，该方法使用经过训练的图形神经网络来预测候选分子是否抑制模型细菌大肠杆菌的生长，他们能够有效地发现 Halicin（一种最初用于治疗糖尿病分子）是一种高效抗生素，即使是针对已知抗生素耐药性的细菌菌株。这一发现被科学和大众媒体广泛报道。

许多药物不是设计出来的，而是偶然发现的。植物王国的许多药物的历史来源反映在它们的名字上：例如，乙酰水杨酸，通常被称为阿司匹林 (aspirin)，包含在柳树皮（柳属）中，柳属植物的药用特性自古以来就为人所知。



Halicin 的分子图。

6.2 药物重新定位

虽然产生全新的候选药物是一种潜在的可行方法，但开发新疗法的更快、更便宜的途径是药物重新定位 (drug repositioning)，即寻求评估已经批准的药物 (单独或联合) 的新用途。这通常会显著减少将药物投放市场所需的临床评估量。在某种抽象层次上，药物对身体生物化学的作用以及它们彼此之间以及与其他生物分子之间的相互作用可以被建模为图形，从而产生了由著名网络科学家阿尔伯特-拉斯洛·巴拉斯 (*Albert-László Barabási*) 创造的“网络医学”概念，并倡导使用生物网络 (如蛋白质-蛋白质相互作用和代谢途径) 来开发新的疗法 (*Barabási et al., 2011*)。

几何深度学习为这类方法提供了一种现代的方式。一个突出的早期例子是 *Zitnik et al. (2018)* 的工作，他使用图形神经网络来预测药物重新定位形式的副作用，称为组合疗法或多药疗法，在药物-药物相互作用图中表示为边缘预测。在撰写本文时，新型冠状病毒大流行仍在进行中，这引发了人们对尝试将此类方法应用于新冠肺炎的特别兴趣 (*Gysi et al., 2020*)。最后，我们应该注意到，药物重新定位不一定局限于合成分子：*Veselkov et al. (2019)* 对食物中包含的药物样分子应用了类似的方法 (因为，正如我们提到的，许多植物基食物包含用于肿瘤治疗的化合物的生物类似物)。本文的一位作者参与了一项合作，通过与一位分子厨师合作，为这项研究增添了创造性，这位分子厨师根据富含这种药物样分子的“超级食物”成分设计了令人兴奋的食谱。

6.3 蛋白质生物学

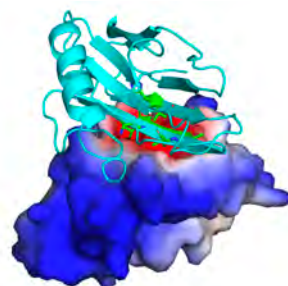
既然我们已经提到了蛋白质作为药物靶点，让我们在这个话题上多花一些时间。蛋白质可以说是我们身体中具有多种功能的最重要的生物分子之一，包括抵御病原体 (抗体)、赋予皮肤结构 (胶原蛋白)、向细胞输送氧气 (血红蛋白)、催化化学反应 (酶) 和发出信号 (许多激素是蛋白质)。从化学角度来说，蛋白质是一种生物聚合物，或者是一系列被称为氨基酸的小构件，在静电力的影响下折叠成复杂的 3D 结构。正是这种结构赋予了蛋白质功能，因

一个常见的隐喻，可以追溯到诺贝尔化学奖得主埃米尔·费舍尔 (*Emil Fischer*)，是施吕塞尔-施洛斯-普林齐普 (*Schlus-Schloss-Prinzip*) (“钥匙锁原理”，1894 年)：两种蛋白质通常只有在几何和化学结构互补的情况下才会相互作用。

此，理解蛋白质如何工作以及它们做什么至关重要。由于蛋白质是药物治疗的常见目标，制药工业对这一领域有着浓厚的兴趣。

蛋白质生物信息学中典型的问题层次是从蛋白质序列 (20 个不同氨基酸字母表上的 1D 字符串) 到 3D 结构 (一个被称为“蛋白质折叠”的问题) 到功能 (“蛋白质功能预测”)。[Senior et al. \(2020\)](#) 最近的方法，如 DeepMind 的 AlphaFold，使用接触图来表示蛋白质结构。[Gligorijevic et al. \(2020\)](#) 表明，将图形神经网络应用于此类图形，可以实现比使用纯粹基于序列的方法更好的功能预测。

[Gainza et al. \(2020\)](#) 开发了一个名为 MaSIF 的几何深度学习管道，从蛋白质的 3D 结构预测蛋白质之间的相互作用。MaSIF 将蛋白质建模为离散为网格的分子表面，认为这种表示在处理相互作用时是有利的，因为它允许抽象内部折叠结构。该体系结构基于网格卷积神经网络，该网络在小的局部测地线片中对预先计算的化学和几何特征进行操作。该网络使用蛋白质数据库中的几千个共晶体蛋白质 3D 结构进行培训，以解决多项任务，包括界面预测、配体分类和对接，并允许重新设计蛋白质 (从零开始)，这些蛋白质原则上可用作抗癌的生物免疫治疗药物——这些蛋白质旨在抑制程序性细胞死亡蛋白质复合物 (PD-1/PD-L1) 各部分之间的蛋白质-蛋白质相互作用 (PPI)，并赋予免疫系统攻击肿瘤细胞的能力。



肿瘤靶向 PD-L1 蛋白表面 (热图显示预测的结合位点) 和设计的结合剂 (如带状图所示)。

6.4 推荐系统与社交网络

图表示学习的第一个普及的大规模应用出现在社交网络中 social networks，主要是在推荐系统的环境中。推荐者的任务是决定向用户提供哪些内容，这可能取决于他们以前在服务上的交互历史。这通常是通过链接预测目标来实现的：监督各种节点 (内容片段) 的嵌入，使得如果它们被认为是相关的 (例如，通常一起观看)，则它们被保持在一起。然后，两个嵌入 (例如，它们的内积) 的接近度 (proximity) 可以被解释为它们通过内容图中的边链接的概率，因此对于用户查询的任何内容，一种方法可以服务于嵌入空间中的 k 个最近邻居。

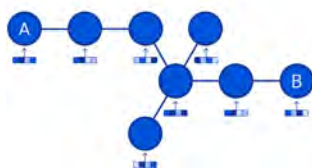


这种方法的先驱之一是美国图像共享和社交媒体公司 *Pinterest*: 除了展示 *PinnerSage*(Pal et al., 2020), 该工作有效地将用户特定的上下文信息集成到推荐器中。GNNs 在生产中的第一次成功部署之外, 他们的方法 *PinSage*成功地使图形表示学习可扩展到数百万个节点和数十亿条边的图形 (Ying et al., 2018)。相关的应用, 特别是在产品推荐领域, 很快接踵而至。目前投入生产的热门 GNNbacked 推荐产品包括阿里巴巴的 *Aligraph* (Zhu et al., 2019) 和亚马逊的 *P-Companion*(Hao et al., 2020)。这样, 图形深度学习每天都在影响着数百万人。

在社交网络内容分析方面, 另一个值得注意的努力是 *Fabula AI*, 它是第一批被收购的 GNN 初创企业之一 (2019 年, 被 *Twitter* 收购)。这家初创公司由该文本的一位作者及其团队创建, 开发了一种新技术来检测社交网络上的错误信息 (Monti et al., 2019)。Fabula 的解决方案包括通过分享某个特定新闻项目的用户网络对其传播进行建模。如果其中一个用户重新分享了另一个用户的信息, 并且他们在社交网络上相互关注, 那么这些用户就是有联系的。然后, 这个图形被输入到一个图形神经网络中, 该网络将整个图形分为“真”或“假”内容——标签基于事实核查机构之间的一致。除了表现出快速稳定的强大预测能力 (通常在新闻传播的几个小时内), 分析单个用户节点的嵌入揭示了倾向于共享不正确信息的用户的清晰聚类, 例证了众所周知的“回声室” (‘echo chamber’) 效应。

6.5 交通预测

交通网络是几何深度学习技术已经对全球数十亿用户产生切实影响的另一个领域。例如, 在道路网络上, 我们可以将交叉点视为节点, 将路段视为连接它们的边, 然后通过路段的道路长度、当前或历史速度等来表征这些边。



道路网络 (顶部) 及其相应的图形表示 (底部)。

这个空间中的一个标准预测问题是预测估计到达时间 (ETA): 对于给定的候选路线, 提供穿越它所需的预期行驶时间。在这个领域, 这样一个问题至关重要, 不仅对于面向用户的流量推荐应用程序, 而且对于在自己的运营中利用这些预测的企业 (如送餐或拼车服务) 也是如此。

图神经网络在这一领域也显示出巨大的潜力：例如，它们可以用于直接预测道路网络相关子图的预计到达时间（实际上是图形回归任务）。DeepMind 成功地利用了这种方法，产生了一个基于 GNNbased 的预计到达时间预测器，该预测器现已在谷歌地图 (Derrow-Pinion *et al.*, 2021) 上投入生产，为全球几个主要大都市地区的预计到达时间查询提供服务。百度地图团队也观察到了类似的回报，其中旅行时间预测目前由 ConSTGAT 模型提供，该模型本身基于图形注意力网络模型的时空变体 (Fang *et al.*, 2020)。



在一些大都市地区，全球导航卫星系统在谷歌地图上提供查询服务，预测质量有所提高（悉尼等城市提高了 40%）。

6.6 物体识别

计算机视觉中机器学习技术的一个主要基准是对所提供图像中的中心对象进行分类的能力。ImageNet 大规模视觉识别挑战 (Russakovsky *et al.*, 2015, ILSVRC) 是一项年度对象分类挑战，推动了几何深度学习的早期发展。ImageNet 要求模型将从网络上抓取的真实图像分为 1000 个类别之一：这些类别同时是多样的（包括有生命的和无生命的物体），以及特定的（许多类别侧重于区分各种猫和狗的品种）。因此，在 ImageNet 上的良好性能通常意味着从普通照片中提取特征的坚实水平，这为来自预先训练的 ImageNet 模型的各种迁移学习设置奠定了基础。



卷积神经网络在 ImageNet 上的成功——特别是 Krizhevsky *et al.* (2012) 的 AlexNet 模型，该模型在很大程度上横扫了 ILSVRC 2012 在很大程度上引领了学术界和工业界对深度学习的整体采用。自那以后，CNNs 一直在 ILSVRC 中名列前茅，产生了许多流行的架构，如 VGG-16 (Simonyan and Zisserman, 2014)、Inception (Szegedy *et al.*, 2015) 和 ResNets (He *et al.*, 2016)，这些架构在这项任务中的性能已成功超过了人类水平。这些架构所采用的设计决策和正则化技术（如校正线性激活 (Nair and Hinton, 2010)、dropout (Srivastava *et al.*, 2014)、跳跃连接 (He *et al.*, 2016) 和批量归一化 (Ioffe and Szegedy, 2015)) 构成了当今使用的许多有效 CNN 模型的主干。

一个输入图像的例子，类似的可以在 ImageNet 中找到，代表“斑猫” (“tabby cat”) 类。有趣的是，VGG-16 体系结构有 16 个卷积层，被作者称为“非常深”。随后的发展迅速将这种模型扩大到数百甚至数千层。

在物体分类的同时，物体检测也取得了重大进展；也就是说，隔离图像中所有感兴趣的对象，并用特定的类标记它们。这一任务与各种下游问题相关，

从图像字幕一直到自动驾驶。它需要一种更精细的方法，因为预测需要本地化；因此，通常情况下，平移等变模型已经证明了它们在这个领域的价值。这一领域一个有影响力的例子包括 *R-CNN* 模型家族 ([Girshick et al., 2014](#); [Girshick, 2015](#); [Ren et al., 2015](#); [He et al., 2017](#))，而在语义分割的相关领域，[Badrinarayanan et al. \(2017\)](#) 的 *SegNet* 模型被证明具有影响力，其编码器-解码器架构依赖于 *VGG-16* 主干。

6.7 博弈对抗

无论何时观察到的状态可以在网格域中表示，卷积神经网络在强化学习 (*RL*) 环境中作为平移不变特征提取器也发挥着突出的作用；例如，当从像素学习玩视频游戏时就是这种情况。在这种情况下，*CNN* 负责将输入减少到平面向量表示，然后用于导出驱动强化学习智能体行为的策略或值函数。虽然强化学习的细节不是本节的重点，但我们确实注意到，过去十年中深度学习的一些最有影响力的结果是通过 *CNN* 支持的强化学习产生的。

这里值得一提的一个特别的例子是 *DeepMind* 的 *AlphaGo* ([Silver et al., 2016](#))。它通过将 *CNN* 应用于代表放置的石头的位置的 19×19 网格来编码围棋游戏中的当前状态。然后，通过从以前的专家动作中学习、蒙特卡罗树搜索和自我游戏的结合，它成功地达到了围棋大师的水平，足以在一场在世界范围内广泛宣传的五轮挑战比赛中击败有史以来最强的围棋选手之一李世石 (*Lee Sedol*)。



围棋的游戏是在 19×19 的棋盘上进行的，两个玩家在空旷的场地上放置白色和黑色的石头。据估计，合法态的数量约为 $\approx 2 \times 10^{170}$ ([Tromp and Farnebäck, 2006](#))，远远超过宇宙中的原子数量。

虽然这已经代表了更广泛的人工智能的一个重要里程碑——围棋的状态空间比象棋复杂得多——但 *AlphaGo* 的发展并没有就此停止。作者逐渐从体系结构中移除了越来越多的围棋特定偏好，*AlphaGo Zero* 移除了人类偏好，纯粹通过自我游戏进行优化 ([Silver et al., 2017](#))，*AlphaZero* 将该算法扩展到相关的双人游戏，如象棋和将棋；最后，*MuZero* ([Schrittwieser et al., 2020](#)) 采用了一种能够动态学习游戏规则模型，这种模型允许在 *Atari 2600* 控制台以及围棋、象棋和将棋中达到强大的性能，而无需任何规则的前期知识。在所有这些发展过程中，中枢神经系统仍然是这些模型表示输入的基础。

虽然多年来为 *Atari2600* 平台提出了几种高性能的强化学习智能体 (*Mnih et al., 2015, 2016; Schulman et al., 2017*), 在很长一段时间里, 他们无法在其中提供的 57 款游戏中达到人类水平的表现。这一障碍最终被 *Agent57* (*Badia et al., 2020*) 打破, 该智能体使用了一系列参数化的策略, 从强烈的探索性到纯粹的剥削性, 并在不同的培训阶段以不同的方式对其进行优先排序。它通过一个卷积神经网络为视频游戏的帧缓冲区提供大部分计算能力。

6.8 文本与语音合成

除了图像 (自然地映射到二维网格), 一些 (几何) 深度学习的最大成功发生在一维网格上。自然的例子是文本和语音, 在自然语言处理和数字信号处理等不同领域折叠几何深度学习蓝图。

这个领域中一些最广泛应用和宣传的作品侧重于合成: 能够无条件地或以特定提示为条件合成 (synthesis) 语音或文本。这种设置可以支持大量有用的任务, 如文本到语音 (TTS)、预测文本补全和机器翻译。在过去十年中, 已经提出了用于文本和语音生成的各种神经架构, 最初主要基于递归神经网络 (例如, 前述 *seq2seq* 模型 (*Sutskever et al., 2014*) 或递归注意力 (*Bahdanau et al., 2014*)). 然而, 近年来, 它们已经逐渐被卷积神经网络和基于变压器的体系结构所取代。

在这种情况下, 简单 *1D* 卷积的一个特殊限制是它们线性增长的感受野 (receptive field), 需要许多层来覆盖到目前为止产生的序列。相反, 扩张卷积提供了一个指数增长的感受野, 具有相同数量的参数。因此, 它们被证明是一个非常强大的替代方案, 最终在机器翻译方面与 *RNNs* 竞争 (*Kalchbrenner et al., 2016*), 同时由于它们在所有输入位置的并行性, 大大降低了计算复杂性。最为人知的扩张卷积应用是 *van den Oord et al. (2016a)* 的 *WaveNet* 模型。*WaveNets* 证明, 使用扩展有可能在原始波形 (通常每秒 16,000 个样本或更多) 的水平上合成语音, 产生比以前最好的文本到语音 (TTS) 系统更“像人”的语音样本。随后, 进一步证明了 *WaveNets* 的计算可以在一

扩张卷积也被称为 *à trous* 卷积, 法语中字面意思是“有孔” (“holed”).

这种技术在蛋白质-蛋白质相互作用等各种问题上也优于 *RNNs* (*Deac et al., 2019*).

除此之外, *WaveNet* 模型被证明能够生成钢琴曲。

个更简单的模型——WaveRNN ([Kalchbrenner et al., 2018](#)) 中进行提炼——并且该模型能够在工业规模上有效地部署该技术。这不仅允许其部署大规模的语音生成服务，如谷歌助手，还允许有效的设备上的计算；例如使用端到端加密的 *Google Duo*。

Transformers ([Vaswani et al., 2017](#)) 已经设法超越了递归和卷积架构的限制，表明自注意力 (self-attention) 足以实现机器翻译的最新性能。随后，它们彻底改变了自然语言处理。通过 *BERT* ([Devlin et al., 2018](#)) 等模型提供的预先训练的嵌入，*Transformer* 计算已经能够用于自然语言处理的大量下游应用——例如，谷歌使用 *BERT* 嵌入来为其搜索引擎提供动力。

可以说，在过去几年里，*Transformer* 最广泛的应用是文本生成，主要是由创成式预先训练的 *Transformer* (*GPT*, [Radford et al. \(2018, 2019\)](#); [Brown et al. \(2020\)](#)) 来自 *OpenAI* 的模型家族。特别是，*GPT-3* ([Brown et al., 2020](#)) 成功地将语言模型学习缩放到 1750 亿个可学习参数，在刮擦的 (*scraped*) 文本语料库的网络规模的数量上训练下一个单词预测。这使得它不仅能够成为一个在各种基于语言的任务中非常有效的少样本学习器，而且能够生成连贯的、听起来像人的文本。这种能力不仅意味着大量的下游应用，还引发了广泛的媒体报道。

6.9 医疗保健

医学领域的应用是几何深度学习的另一个有前途的领域。这些方法有多种使用方式。首先，更多的传统架构 (如中枢神经系统) 已应用于网格结构的数据，例如，预测重症监护病房的住院时间 ([Rocheteau et al., 2020](#))，或通过视网膜扫描诊断威胁视力的疾病 ([De Fauw et al., 2018](#))。Winkels and Cohen (2019) 表明，与传统的中枢神经系统相比，使用 3D 旋转平移群卷积网络可以提高肺结节检测的准确性。

第二，将器官建模为几何表面，网格卷积神经网络被证明能够解决各种各样的任务，从遗传学相关信息重建面部结构 ([Mahdi et al., 2020](#)) 到大脑皮层

分组 (*Cucurull et al., 2018*) 到从皮层表面结构回归人口统计属性 (*Besson et al., 2020*)。后一个例子代表了神经科学中的一种增长趋势, 即认为大脑是一个具有复杂褶皱的表面, 从而产生了高度非欧几里得结构。

大脑皮层的这种结构在解剖学文献中被称为脑沟 (sulci) 和脑回 (gyri)。

与此同时, 神经科学家经常试图构建和分析大脑的功能网络 (functional networks), 这些功能网络代表大脑中在执行某些认知功能时被一起激活的各个区域; 这些网络通常使用功能性磁共振成像 (fMRI) 来构建, 该成像实时显示大脑的哪些区域消耗更多的血液。这些功能网络可以揭示患者的人口统计数据 (例如, 区分男性和女性, *Arslan et al. (2018)*), 并用于神经病理学诊断, 这是几何深度学习在医学中的第三个应用领域, 我们希望在此强调。在这种情况下, *Ktena et al. (2017)* 率先使用图形神经网络来预测神经疾病, 如自闭症谱系障碍。大脑的几何结构和功能结构似乎密切相关, 最近 *Itani and Thanou (2021)* 指出了在神经疾病分析中联合利用它们的好处。

典型地, 使用血氧水平依赖 (BOD) 对比成像。

第四, 患者网络 (patient networks) 在基于最大似然估计的医学诊断中变得越来越突出。这些方法背后的基本原理是, 患者人口统计学、基因型和表型相似性的信息可以改善对他们疾病的预测。*Parisot et al. (2018)* 将图形神经网络应用于根据人口统计学特征创建的患者网络, 用于神经疾病诊断, 表明图形的使用改善了预测结果。*Cosmo et al. (2020)* 展示了在这种情况下潜在图形学习 (网络通过它学习未知的患者图形) 的好处。后一项工作使用了英国生物银行的数据, 这是一个包括脑成像在内的大规模医学数据集合 (*Miller et al., 2016*)。

关于医院病人的大量数据可以在电子健康记录 (electronic health records (EHRs)) 中找到。除了给出患者病情进展的全面视图, EHR 分析还允许将相似的患者联系在一起。这与诊断中常用的模式识别方法 (pattern recognition method) 相一致。其中, 临床医生使用经验来识别临床特征的模式, 并且当临床医生的经验可以使他们快速诊断病情时, 这可能是使用的主要方法。沿着这些思路, 一些工作试图基于数据构建患者图, 或者通过分析他们的医生笔记的嵌入 (*Malone et al., 2018*), 入院时的诊断相似性 (*Rocheteau et al., 2021*), 或者甚至假设完全连通的图 (*Zhu and Razavian, 2019*)。在所有情况下, 有希望的结果已经显示出支持使用图形表示学习来处理 EHRs。

公开的匿名 EHR 重症监护数据集包括 MIMIC-III (*Johnson et al., 2016*) 和 eICU (*Pollard et al., 2018*)。

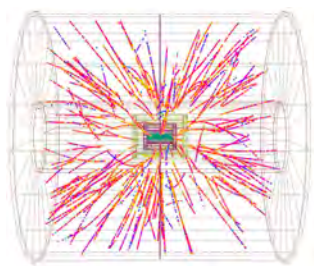
6.10 粒子物理和天体物理



大型强子对撞机探测器的一部分。

高能物理学家可能是自然科学领域第一批拥抱新的闪亮工具——图神经网络的领域专家之一。在最近的一篇综述论文中, [Shlomi et al. \(2020\)](#)指出, 机器学习在粒子物理实验中一直被大量使用, 要么学习复杂的反函数, 允许从探测器中测量的信息推断底层物理过程, 要么执行分类和回归任务。对于后者, 为了能够使用标准的深度学习架构, 如 *CNN*, 通常需要将数据强制转换为不自然的表示, 如格网 (*grid*)。然而, 物理学中的许多问题涉及到具有丰富关系和相互作用的无序集合形式的数据, 这些数据可以自然地表示为图形。

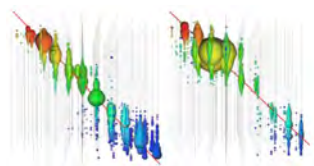
高能物理的一个重要应用是粒子射流 (particle jets) 的重建和分类——由多次连续相互作用产生的稳定粒子喷雾和由单个初始事件产生的粒子衰变。在欧洲粒子物理研究所建造的最大和最著名的粒子加速器大型哈顿对撞机中, 这种射流是质子以接近光速的速度碰撞的结果。这些碰撞产生了大量的粒子, 比如长粒子——希格斯玻色子 (*Higgs boson*) 或顶夸克 (*top quark*)。碰撞事件的识别和分类至关重要, 因为它可能为新粒子的存在提供实验证据。



粒子射流示例。

最近, 多种几何深度学习方法已经被提出用于粒子射流分类任务, 例如, 分别由 [Komiske et al. \(2019\)](#) 和 [Qu and Gouskos \(2019\)](#) 基于 *DeepSet* 和 *Dynamic Graph CNN* 架构。最近, 人们还对开发基于物理考虑的特殊架构感兴趣, 并结合符合哈密顿量或拉格朗日力学的归纳偏差 (例如, [Sanchez-Gonzalez et al. \(2019\)](#); [Cranmer et al. \(2020\)](#)), 洛伦兹群的等变 (物理学中空间和时间的基本对称性) ([Bogatskiy et al., 2020](#)), 甚至结合符号推理 ([Cranmer et al., 2019](#)), 能够从数据中学习物理规律。这种方法更容易解释 (因此被领域专家认为更“可信”), 也提供了更好的泛化性。

除了粒子加速器之外, 粒子探测器现在正被天体物理学家用于多信使天文学 (multi-messenger astronomy)——一种协调观察来自同一来源的不同信号 (如电磁辐射、引力波和中微子) 的新方法。中微子天文学尤其令人感兴趣, 因为中微子很少与物质相互作用, 因此可以不受影响地传播很远的距离。探测中微子可以观察光学望远镜无法观察到的物体, 但需要非常大的探测



自背景事件 (*muon* 子束, 左) 和天体物理中微子 (高能单 *muon* 子, 右) 的 IceCube 探测器中光沉积 (*light* 来

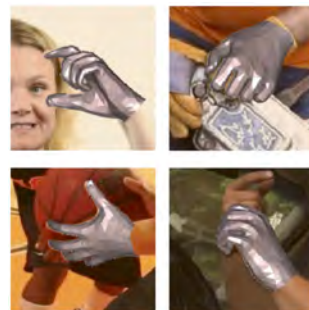
器——冰立方中微子天文台使用南极一立方公里的南极冰架作为探测器。探测高能中微子可能会照亮宇宙中一些最神秘的物体，如黑洞和黑洞。[Choma et al. \(2018\)](#) 使用几何神经网络对冰立方中微子探测器的不规则几何形状进行建模，在探测来自天体物理源的中微子并将它们与背景事件分离方面显示出显著更好的性能。

虽然中微子天文学为宇宙研究提供了巨大的希望，但传统的光学和射电望远镜仍然是天文学家的“战马”。有了这些传统的工具，几何深度学习仍然可以为数据分析提供新的方法。例如，[Scaife and Porter \(2021\)](#) 使用旋转等变氯化茶对射电星系进行分类，[McEwen et al. \(2021\)](#) 使用球形氯化茶对宇宙微波背景辐射进行分析，这是大爆炸的遗迹，可能会揭示原始宇宙的形成。正如我们已经提到的，这种信号自然地在球体上表示，等变神经网络是研究它们的合适工具。

6.11 虚拟与增强现实

另一个应用领域是计算机视觉和图形学，特别是处理虚拟和增强现实的三维人体模型，它是开发一大类几何深度学习方法的动力。在《阿凡达》等电影中用于产生特殊效果的运动捕捉技术通常分两个阶段运行：首先，捕捉演员身体或面部运动的 3D 扫描仪的输入与一些典型形状（通常建模为离散流形或网格）对应（这个问题通常称为“分析”）。其次，生成一个新的形状来重复输入的运动（“合成”）。计算机图形学与视觉中的几何深度学习 ([Masci et al., 2015](#); [Boscaini et al., 2016a](#); [Monti et al., 2017](#)) 开发了网状卷积神经网络来解决分析问题，或者更具体地说，可变形形状对应。

[Litany et al. \(2018\)](#) 和 [Ranjan et al. \(2018\)](#) 独立提出了第一个用于 3D 形状合成的几何自动编码器架构。在这些体系结构中，假设（身体、面部或手的）标准网格是已知的，并且合成任务包括回归节点的 3D 坐标（表面的嵌入，使用微分几何的行话）。[Kulon et al. \(2020\)](#) 展示了一种用于 3D 手姿态估计的混合管道，该管道具有基于 CNN 的图像编码器和几何解码器。该系统的演示是与英国初创公司阿里埃勒人工智能 (Ariel AI) 合作开发的，并

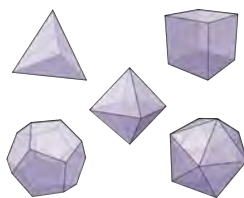


从野外 2D 图像重建的复杂 3D 手部姿势的示例 ([Kulon et al., 2020](#)).

在 *CVPR 2020* 上展示，它允许通过手机上的视频输入，以比实时更快的速度，用完全铰接的手创建逼真的身体化身。*Ariel AI* 于 2020 年被 *Snap* 收购，在撰写本文时，其技术用于 *Snap* 的增强现实产品。

7 历史视角

“对称, 无论你怎么定义它的含义, 它都是一种观念, 自古以来人类就试图通过它来理解和创造秩序、美和完美。”伟大数学家 *Hermann* 的同名著作中给出了这种有点诗意的对称定义 [Weyl \(2015\)](#), 他在普林斯顿高等研究院退休前夕去世。Weyl 将对称在科学和艺术中的特殊地位追溯到古代, 从苏美尔 (*Sumerian*) 的对称设计到毕达哥拉斯学派 (*Pythagoreans*), 他们相信圆是完美的, 因为它是旋转对称的。柏拉图认为今天以他的名字命名的五个正多面体是如此的基本, 以至于它们必须是塑造物质世界的基本构件。然而, 尽管柏拉图被认为创造了 *συμμετρία* 这个词, 字面意思是“同样的度量”, 但他只是模糊地用它来传达艺术中的比例美和音乐中的和谐美。天文学家和数学家约翰尼斯·开普勒首次尝试对水晶体的对称形状进行严格的分析。天文学家和数学家约翰尼斯·开普勒 (*Johannes Kepler*) 首次尝试对水晶体的对称形状进行严格的分析。在他的论文“六角雪花”中, 他将雪花的六重二面角结构归因于颗粒的六边形堆积——这一想法虽然先于对物质如何形成的清晰理解, 但今天仍然是结晶学的基础 ([Ball, 2011](#))。



四面体、立方体、八面体、十二面体和二十面体被称为柏拉图多面体。

书名为“新年礼物”, 或“六角雪花”, 是开普勒在 1611 年作为圣诞礼物送给他的赞助人和朋友 *Johannes Matthäus Wackher von Wackenfels* 的小册子。

7.1 数学与物理中的对称

在现代数学中, 对称性几乎是用群论的语言来表达的。这一理论的起源通常归功于埃瓦里斯特·伽罗瓦 (*Évariste Galois*), 他在 19 世纪 30 年代创造了这个术语, 并用它来研究多项式方程的可解性。另外两个与群论有关的名字是索福思·李 (*Sophus Lie*) 和费利克斯·克莱因 (*Felix Klein*), 他们相遇并一起工作了一段时间, 并取得了丰硕的成果 ([Tobies, 2019](#))。前者将发展至今以他的名字命名的连续对称理论; 后者在他的埃尔兰根纲领中宣称群论是几何学的组织原则, 我们在本文开头提到过。黎曼几何被明确排除在克莱因的统一几何图景之外, 又过了五十年才被整合, 这很大程度上要归功于 20 世纪 20 年代埃利·卡坦 (*Élie Cartan*) 的工作。

艾米·诺特 (*Emmy Noether*), 克莱因在哥廷根的同事, 证明了一个物理系统

的每一个可微的对称性都有相应的守恒定律 (*Noether, 1918*). 在物理学中, 这是一个惊人的结果: 在此之前, 需要细致的实验观察来发现能量守恒等基本定律, 即使在那时, 这也是一个来自任何地方的经验结果. 诺瑟定理——用诺贝尔奖得主弗兰克·维尔泽克 (*Frank Wilczek*) 的话说, 是“20 世纪和 21 世纪物理学的一颗指路之星”——表明能量守恒源于时间的平移对称性, 这是一个相当直观的想法, 即实验的结果不应取决于它是在今天还是明天进行.

1919 年, *Weyl* 第一次推测 (错误地) 尺度或“规范”变化下的不变性是电磁学的局部对称性. 规范一词, 或德语中的 *Eich*, 是通过类比铁路的各种规范来选择的. 在量子力学发展之后, *Weyl (1929)* 通过用波相的变化代替标度因子来修改规范选择. 见 *Straumann (1996)*.

与电荷守恒相关的对称性是电磁场的整体规范不变性, 首次出现在麦克斯韦的电动力学公式中 (*Maxwell, 1865*); 然而, 它的重要性最初并未引起注意. 同样是赫尔曼·韦尔 (*Hermann Weyl*) 如此犹豫不决地写下了对称性, 他是 20 世纪初第一个在物理学中引入规范不变性概念的人, 强调它作为一种可以导出电磁学的原理的作用. 直到几十年后, 这一基本原理——由杨和米尔斯 *Yang and Mills (1954)* 发展的广义形式——才被证明成功地提供了一个统一的框架来描述电磁学的量子力学行为以及强弱力, 最终形成了标准模型, 该模型捕捉了除重力之外的所有基本自然力. 因此, 我们可以与另一位诺贝尔奖获得者物理学家 *Philip Anderson (1972)* 一起得出结论, “说物理是对称性的研究只是稍微夸大其词.”

7.2 机器学习中早期使用对称

Shun'ichi Amari 被认为是将黎曼几何模型应用于概率的信息几何领域的创造者. 信息几何研究的主要对象是统计流形, 其中每个点对应一个概率分布.

在机器学习及其在模式识别和计算机视觉中的应用中, 对称性的重要性早已得到认可. 设计用于模式识别的等变特征检测器的早期工作是由 *Amari (1978)*, *Kanatani (2012)*, 和 *Lenz (1990)* 做的. 在神经网络文献中, *Minsky and Papert (2017)* 的著名感知器群不变性定理对 (单层) 感知器学习不变量的能力提出了基本限制. 这是研究多层架构的主要动机之一 (*Sejnowski et al., 1986; Shawe-Taylor, 1989, 1993*), 最终导致深度学习.

在神经网络社区中, *Neocognitron (Fukushima and Miyake, 1982)* 被认为是神经网络中“不受位置变化影响的模式识别”的移动不变性的第一个实现. 他的解决方案是以具有局部连通性的分层神经网络的形式出现的, 灵感来

自二十年前神经科学家大卫·胡贝尔 (David Hubel) 和托尔斯滕·威塞尔 (Torsten Wiesel) 在视觉皮层中发现的感受野 (Hubel and Wiesel, 1959). 这些想法在 Yann LeCun 和合著者 (LeCun et al., 1998) 的开创性工作中的卷积神经网络中达到顶峰. Wood and Shawe-Taylor (1996) 完成了对不变和等变神经网络的表示理论观点的第一项工作, 不幸的是很少被引用. 这些思想的最新体现包括 Makadia et al. (2007); Esteves et al. (2020) 和本文作者之一 (Cohen and Welling, 2016).

这项经典的工作得到了 1981 年诺贝尔医学奖的认可, 由 Hubel, Wiesel 与 Roger Sperry 分享.

7.3 图神经网络

图神经网络的概念是如何开始出现的很难解释——部分是因为大多数早期的工作没有将图形作为一等公民, 部分是因为 GNNs 直到 2010 年代后期才变得实用, 部分是因为这个领域是从几个研究领域的融合中出现的. 也就是说, 图神经网络的早期形式至少可以追溯到 20 世纪 90 年代, 例子包括 Alessandro Sperduti's Labeling RAAM (Sperduti, 1994), Goller and Kuchler (1996) 的“通过结构的反向传播”, 和数据结构的自适应处理 (Sperduti and Starita, 1997; Frasconi et al., 1998). 虽然这些工作主要关注于对“结构”(通常是树或有向无环图) 的操作, 但它们的架构中保留的许多不变性让人想起今天更常用的 GNNs.

第一次正确处理一般图形结构的过程 (以及“图神经网络”一词的诞生) 发生在 21 世纪之交. 在锡耶纳大学 (Università degli Studi di Siena) 的人工智能实验室里, 由 Marco Gori 和 Franco Scarselli 领导的论文提出了第一个“GNN” (Gori et al., 2005; Scarselli et al., 2008). 他们依赖循环机制, 需要神经网络参数来指定压缩映射, 从而通过搜索固定点来计算节点表示——这本身就需要一种特殊形式的反向传播 (Almeida, 1990; Pineda, 1988) 并且根本不依赖于节点特征. Li et al. (2015) 的门控模型纠正了上述所有问题. GGNNs 带来了现代神经网络的许多好处, 如 (Cho et al., 2014) 和通过时间的反向传播, 到 GNN 模型, 并保持流行至今.

与此同时, 阿莱西奥·米歇尔提出了图神经网络 (neural network for graphs, NN4G) 模型, 该模型侧重于前馈 (feedforward) 而不是递归范式 (Micheli, 2009).

7.4 计算化学

同样非常重要的是要注意到 GNNs 的一条独立且并行的发展路线：一条完全由计算化学需求驱动的路线，其中分子最自然地表达为由化学键（边）连接的原子（节点）的图形。这邀请了直接在这样的图形结构上操作的分子性质预测的计算技术，它在 20 世纪 90 年代已经出现在机器学习中：这包括 Kireev (1995) 的 ChemNet 模型 (1995) 和 Baskin et al. (1997) 的工作。引人注目的是，Merkwirth and Lengauer (2005) 的“分子图网络”早在 2005 年就明确提出了许多在当代 GNNs 中常见的元素——例如边缘类型条件权重或全局汇集。化学研究动机继续推动 GNN 发展到 2010 年代，GNN 的两项重大进展集中在改进分子指纹 (Duvenaud et al., 2015) 和预测小分子的量子化学性质 (Gilmer et al., 2017)。在撰写本文时，分子性质预测是 GNNs 最成功的应用之一，在新抗生素药物的虚拟筛选中产生了有影响力的结果 (Stokes et al., 2020)。

7.5 节点嵌入

一些关于图的深度学习的早期成功故事涉及到基于图结构以无监督的方式学习节点的表示。鉴于他们的结构灵感，这个方向也提供了图形表示学习和网络科学社区之间最直接的联系之一。这一领域的关键早期方法依赖于基于随机行走的嵌入：学习节点表示，如果节点在短时间内随机行走，则使它们更接近。这个空间中有代表性的方法有 DeepWalk (Perozzi et al., 2014)、node2vec (Grover and Leskovec, 2016) 和 LINE (Tang et al., 2015)，都是纯自监督的。Planetoid (Yang et al., 2016) 是第一个在这个空间纳入监管标签信息（如果有）。

最近，Srinivasan and Ribeiro (2019) 开发了一个理论框架，其中证明了结构和位置表示的等效性。此外，Qiu et al. (2018) 已经证明，所有基于随机行走的嵌入技术相当于适当设定的矩阵分解任务。曾多次尝试用 GNN 编码器统一随机行走目标，有代表性的方法包括变分图自动编码器 (VGAE, Kipf and Welling (2016b))、嵌入传播 (García-Durán and Niepert, 2017) 和 GraphSAGE 的无监督变体 (Hamilton et al., 2017)。然而，结果喜忧参半，人们很快发现，将相邻的节点表示推到一起已经是 GNNs 感应偏置的一个关键部分。事实证明，在节点功能可用的情况下，未

经训练的 GNN 已经表现出与 DeepWalk 相当的性能 (Veličković et al., 2019; Wu et al., 2019). 这开启了一个方向, 从将随机行走目标与 GNNs 相结合, 转向受互信息最大化启发的对比方法, 并与图像领域的成功方法相一致. 这方面的突出例子包括 Deep Graph Informax(DGI, Veličković et al. (2019))、GRACE(Zhu et al., 2020)、BERT 样目标 (Hu et al., 2020) 和 BGRL (Thakoor et al., 2021).

7.6 概率图模型

同时, 图神经网络也通过嵌入概率图模型的计算而复兴. 概念图模型 (PGMs, Wainwright and Jordan (2008)) 是处理图形数据的强大工具, 它们的效用来自于对图形边的概率: 也就是说, 节点被视为随机变量, 而图形结构编码条件独立假设, 允许显著简化联合分布的计算和采样. 事实上, 许多 (精确地或近似地) 支持 PGMs 上的学习和推理的算法依赖于通过它们的边传递消息的形式 (Pearl, 2014), 例子包括变分平均场推理和循环信念传播 (Yedidia et al., 2001; Murphy et al., 2013).

PGMs 和消息传递之间的这种联系随后被发展到体系结构中, 由 *structure2vec*(Dai et al., 2016) 的作者建立了早期的理论联系. 也就是说, 通过将图形表示学习设置提出为马尔可夫随机场 (对应于输入特征和潜在表示的节点), 作者直接将平均场推理和循环信念传播的计算与当今常用的 GNNs 模型进行了比较.

关键的“技巧”是使用分布的希尔伯特空间嵌入 (Hilbert-space embeddings), 它允许将 GNN 的潜在表示与概率模型保持的概率分布联系起来 (Smola et al., 2007). 给定 ϕ , 一个为特征 \mathbf{x} 适当选择的嵌入函数, 有可能嵌入它们的概率分布 $p(\mathbf{x})$ 作为期望嵌入 $\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \phi(\mathbf{x})$. 这种对应允许我们执行类似 GNN 的计算, 知道由 GNN 计算的表示将总是对应于在节点特征上的某种概率分布的嵌入.

最终, *structure2vec* 模型本身是一个 GNN 架构, 它很容易被放在我们的框

架中, 但是它的设置启发了一系列 GNN 架构, 这些架构更直接地结合了 PGMs 中的计算. 新兴的例子已经成功地将 GNNs 与条件随机场相结合 (Gao et al., 2019; Spalević et al., 2020), 关系马尔科夫网络 (Qu et al., 2019) 和马尔科夫逻辑网 (Zhang et al., 2020).

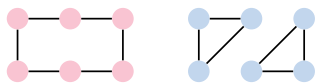
7.7 Weisfeiler-Lehman 形式化

图形神经网络的复兴紧随其后的是理解其基本局限性的努力, 特别是在表达能力方面. 虽然 GNNs 是一个强大的图形结构数据建模工具变得越来越明显, 但很明显, 它们不能完美地解决图形上指定的任何任务. 一个典型的例子是决定图的同构 (graph isomorphism): 我们的 GNN 能够给两个给定的非同构图附加不同的表示吗? 这是一个有用的框架, 原因有二. 如果 GNN 不能做到这一点, 那么它在任何需要区分这两个图形的任务上都是没有希望的. 此外, 目前还不知道决定图同构的是在 P 中, 这是所有 GNN 通常的计算复杂性.

由于它们的排列不变性, GNNs 会把相同的表示附加到两个同构的图上, 所以这种情况被简单地解决了.

目前最著名的决定图同构的算法是由 Babai and Luks (1983) 提出的, 尽管 Babai (2016) 最近的一个提议 (未被完全审查) 暗示了一个准多项式时间的解决方案.

将 GNNs 绑定到图同构的关键框架是 Weisfeiler-Lehman (WL) 图同构测试 (Weisfeiler and Leman, 1968). 该测试通过沿着图的边缘反复传递节点特征, 然后随机散列它们在邻近区域的总和, 来生成图表示. 与随机初始化的卷积神经网络的联系是显而易见的, 并且已经在早期观察到了: 例如, 在 Kipf and Welling (2016a) 的 GCN 模型中. 除了这种联系之外, WL 迭代以前是由 Shervashidze et al. (2011) 引入图核领域的, 并且它仍然为全图表示的无监督学习提供了强基线.



一个简单的例子: WL 测试无法区分 6 个环和 2 个三角形.

虽然 WL 测试在概念上很简单, 并且有许多它无法区分的非同构图的简单例子, 但它的表达能力最终与 GNNs 紧密相连. Morris et al. (2019) 和 Xu et al. (2018) 的分析都得出了一个惊人的结论: 任何符合我们在第 5.3 节中概述的三种口味之一的都不可能比 WL 测试更有效!

为了准确地达到这一表示水平, GNN 更新规则必须存在某些限制. Xu et al. (2018) 表明, 在离散特征域中, 使用的聚集函数必须是单射的 (injective), 而

summation 是一个关键表示. 基于他们的分析结果, [Xu et al. \(2018\)](#) 提出了图同构网络 (GIN), 它是在此框架下最大表达的的一个简单而有力的例子. 它也可以在我们提出的卷积 GNN 风味下表达.

最大化和平均化等流行的聚合器在这方面有所欠缺, 因为它们无法区分, 比如领域多集合 $\{\{a, b\}\}$ and $\{\{a, a, b, b\}\}$.

最后, 值得注意的是, 这些发现不能推广到连续的节点特征空间. 事实上, 使用 Borsuk-Ulam 定理 ([Borsuk, 1933](#)), [Corso et al. \(2020\)](#) 已经证明, 假设实值节点特征, 获得单射聚合函数需要多个聚合器 (具体地, 等于接收节点的度). 他们的发现推动了主要邻域聚合 (PNA) 架构, 该架构提出了一个经验上强大且稳定的多聚合器 GNN.

这种聚合器的一个例子是多集邻居的时刻.

7.8 高阶方法

前面几段的发现并不与神经网络的实际效用相矛盾. 事实上, 在许多现实世界的应用中, 输入特征足够丰富, 以支持对图形结构的有用的区分计算, 尽管有上述限制.

相比之下, 几乎总是认为无特征或分类特征的图形.

然而, 一个关键的推论是, 神经网络在检测图形中的一些基本结构方面相对较弱. 在 WL 试验的具体限制或失败案例的指导下, 一些工作已经提供了比 WL 试验更强有力的 GNNs 变体, 因此可能对需要这种结构检测的任务有用.

一个突出的例子是计算化学, 其中分子的化学功能会受到分子图中芳香环的强烈影响.

或许寻找更具表现力的神经网络最直接的地方是 WL 测试本身. 事实上, 原始 WL 测试的强度可以通过考虑 WL 测试的层次结构来增强, 使得 k -WL 测试将表示附加到节点的 k -元组 ([Morris et al., 2017](#)). [Morris et al. \(2019\)](#) 已经将 k -WL 测试直接转化为更高阶的 k -GNN 架构, 这种架构比我们之前考虑的 GNN 风格更强大. 然而, 它维护元组表示的要求意味着, 实际上很难扩展到 $k = 3$ 以上.

已经有一些努力, 例如 δ - k -LGNN ([Morris et al., 2020](#)), 来稀疏化 k -GNN 的计算.

同时, [Maron et al. \(2018, 2019\)](#) 研究了节点 k 元组上不变和等变图网络的特征. 除了证明任何不变的或等变的图形网络都可以表示为有限数量的生成器的线性组合的令人惊讶的结果——其数量仅取决于 k ——作者还证明了这种层的表达能力相当于 k -WL 测试, 并提出了一个经验上可扩展的变体, 该

变体可证明是 3-WL 强大的。

除了对计算表示的领域进行概括之外, 还投入了大量的精力来分析 1-WL 的具体故障情况, 并增加 GNN 的输入, 以帮助他们区分这种情况. 一个常见的例子是将识别特征附加到节点, 这可以帮助检测结构. 这样做的建议包括独热 (one-hot) 表示 (Murphy et al., 2019), 以及纯随机特征 (Sato et al., 2020).

更广泛地说, 已经有许多努力将结构信息结合到消息传递过程中, 或者通过调制消息函数或者通过传递计算的图形. 这里有几个有趣的工作包括对锚节点集 (anchor node sets) 进行采样 (You et al., 2019), 基于拉普拉斯特征向量 (Laplacian eigenvectors) 进行聚合 (Stachenfeld et al., 2020; Beaini et al., 2020; Dwivedi and Bresson, 2020), 或进行拓扑数据分析, 或用于位置嵌入 (Bouritsas et al., 2020) 或驱动信息传递 (Bodnar et al., 2021).

7.9 信号处理与调和分析

自从卷积神经网络的早期成功以来, 研究人员已经求助于谐波分析、图像处理和计算神经科学的工具, 试图提供一个理论框架来解释它们的效率. *M-theory* 是一个受视觉皮层启发的框架, 由托马索·波吉欧 (Tomaso Poggio) 及其合作者首创 (Riesenhuber and Poggio, 1999; Serre et al., 2007), 基于模板的概念, 可以在某些对称群下操作. 由计算神经科学产生的另一个值得注意的模型是可操纵金字塔, 这是一种多尺度小波分解的形式, 对某些输入变换具有有利的性质, 由 Simoncelli and Freeman (1995) 开发. 它们是早期纹理生成模型的核心元素 (Portilla and Simoncelli, 2000), 后来通过用深层 CNN 特征替换可控小波特征进行了改进 Gatys et al. (2015). 最后, 由 Stéphane Mallat (2012) 引入并由 Bruna and Mallat (2013) 开发的散射变换, 通过用多尺度小波分解代替可训练滤波器, 提供了一个理解中枢神经系统的框架, 也展示了变形稳定性和深度在体系结构中的作用.

7.10 图与网格上信号处理

图神经网络的另一个重要类别, 通常被称为谱 (spectral), 已经出现在本文作者之一的工作中 (*Bruna et al., 2013*), 使用了图傅立叶变换 (Graph Fourier transform) 的概念. 这种结构的根源在于信号处理和计算谐波分析领域, 在 2000 年代末和 2010 年代初, 处理非欧几里得信号变得非常突出. 来自 Pierre Vandergheynst (*Shuman et al., 2013*) 和 José Moura (*Sandryhaila and Moura, 2013*) 小组的有影响力的论文普及了“图信号处理” (GSP) 的概念和基于图形邻接和拉普拉斯矩阵的特征向量的傅立叶变换的推广. *Defferrard et al. (2016)* 和 *Kipf and Welling (2016a)* 的依赖于谱滤波器的图卷积神经网络是该领域引用最多的网络之一, 并且可能被认为是近年来重新点燃机器学习对图的兴趣的网络.

值得注意的是, 在计算机图形学和几何处理领域, 非欧调和与分析比图形信号处理至少早了十年. 我们可以追溯流形和网格上的谱滤波器到 *Taubin et al. (1996)* 的工作. 在 *Karni and Gotsman (2000)* 关于谱几何压缩和 *Lévy (2006)* 关于使用拉普拉斯特征向量作为非欧几里得傅立叶基础的有影响力的论文之后, 这些方法在 2000 年代成为主流. 谱方法已被广泛应用 其中最突出的是构造形状描述符 (*Sun et al., 2009*) 和函数映射 (functional maps) (*Ovsjanikov et al., 2012*); 在撰写本文时, 这些方法仍广泛用于计算机图形学.

Roe Litman 和 *Alex Bronstein (2013)* 提出了类似于谱图 CNNs 的可学习形状描述符, 后者是本文作者的学生兄弟.

7.11 计算机图形学与几何处理

基于内在度量不变量的形状分析模型由计算机图形和几何处理领域的不同作者引入 (*Elad and Kimmel, 2003; Mémoli and Sapiro, 2005; Bronstein et al., 2006*), 并在其中一个作者的早期著作 (*Bronstein et al., 2008*) 中得到深入讨论. *Raviv et al. (2007)*; *Ovsjanikov et al. (2008)* 在同一领域也探讨了内在对称性的概念. 网格深度学习的第一个体系结构是测地线 CNNs, 是由 (*Masci et al., 2015*) 该文的一位作者的团队开发的. 该模型使用具有共享权重的局部滤波器, 应用于测地线径向面片. 这是后来由该文本的另一位作者

(Cohen et al., 2019) 开发的规范等变 CNNs 的一种特例。Federico Monti et al. (2017) 在同一团队中提出了一种具有可学习聚合操作的测地线 CNNs 的一般化方法 MoNet, 该方法在网格的局部结构特征上使用了一种类似注意力的机制, 这种机制在一般的图形上也同样有效。图形注意网络 (GAT) 从技术上来说可以被认为是 MoNet 的一个特定实例, 它是由本文的另一位作者引入的 (Veličković et al., 2018)。GATs 囊括了 MoNet 的注意力机制, 也纳入了节点特征信息, 脱离了先前工作的纯粹结构衍生的相关性。它是目前使用的最受欢迎的 GNN 架构之一。

在计算机图形学的背景下, 还值得一提的是, 在集合上学习的想法 (Zaheer et al., 2017) 是由斯坦福大学的 Leo Guibas 团队以 PointNet (Qi et al., 2017) 的名义同时开发的, 用于分析 3D 点云。这种架构导致了多个后续作品, 包括本文作者之一的动态图形 CNN (DGCNN, Wang et al. (2019b))。DGCNN 使用最近邻图来捕捉点云的局部结构, 以便在节点之间交换信息; 这种体系结构的关键特征是图形是动态构建的, 并在与下游任务相关的神经网络层之间进行更新。后一个特性使 DGCNN 成为“潜在图形学习”的第一个体现, 这反过来又产生了重大的后续影响。扩展到 DGCNN 的 k -近邻图建议包括通过双层优化 (Franceschi et al., 2019)、强化学习 (Kazi et al., 2020) 或直接监督 (Veličković et al., 2020) 对这些图的边进行更明确的控制。独立地, 通过 NRI 模型出现了一个变化的方向 (从计算的后验分布中概率性地采样边)(Kipf et al., 2018)。虽然它仍然依赖于节点数量的二次计算, 但它允许对所选边的不确定性进行显式编码。

另一个非常流行的方向是在没有提供图形的情况下学习图形, 它依赖于在一个完全图形上执行 GNN 风格的计算, 让网络推断自己的方式来利用连通性。这种需求尤其出现在自然语言处理中, 在自然语言处理中, 句子中的各种单词以非常不寻常和不连续的方式相互作用。对一个完整的单词图进行操作带来了 Transformer 模型的第一个化身 (Vaswani et al., 2017), 它超越了递归和卷积模型, 成为神经机器翻译的最新技术, 并引发了相关工作的雪崩, 超越了自然语言处理和其他领域的界限。全连通 GNN 计算也同时出现在模拟 (Battaglia et al., 2016)、推理 (Santoro et al., 2017) 和多智能体 (Hoshen,

2017) 应用中, 并且当节点数量相当少时仍然是一种流行的选择.

7.12 算法推理

在本节提出的大部分讨论中, 我们给出了空间诱导几何的例子, 这些例子反过来又塑造了底层的定义域, 以及它的不变性和对称性. 然而, 大量不变性和对称性的例子也出现在计算环境中. 许多常见场景几何深度学习的一个关键区别是, 链接不再需要为任何类型的相似性、邻近性或关系类型进行编码——它们只是为它们连接的数据点之间的数据流指定“配方”.

相反, 神经网络的计算模拟算法的推理过程 (Cormen et al., 2009), 由算法的控制流和中间结果引起的附加不变性. 在算法空间中, 假设的输入不变量通常被称为前置条件, 而算法保留的不变量被称为后置条件.

例如, Bellman-Ford 寻路算法 (Bellman, 1958) 的一个不变量是, 在 k 步之后, 它将总是计算到使用不超过 k 条边的源节点的最短路径.

同名, 算法推理 (algorithmic reasoning) 的研究方向 (Cappart et al., 2021, 第 3.3 节) 是寻求产生适当保存算法不变量的神经网络体系结构. 该领域研究了通用神经计算机的构造, 例如神经测试机 (neural Turing machine) (Graves et al., 2014) 和可微分神经计算机 (differentiable neural computer) (Graves et al., 2016). 虽然这种架构具有一般计算的所有特征, 但它们同时引入了几个组件, 这使得它们经常难以优化, 并且在实践中, 它们几乎总是被简单的关系推理器超越, 例如 Santoro et al. (2017, 2018) 提出的推理器.

由于模拟复杂的后条件具有挑战性, 关于学习执行的归纳偏置的大量工作 (Zaremba and Sutskever, 2014) 集中在原始算法上 (例如简单的算术). 这一领域的突出例子包括神经 GPU (neural GPU) (Kaiser and Sutskever, 2015)、神经 RAM (neural RAM) (Kurach et al., 2015)、神经程序解释器 (neural programmer-interpreters) (Reed and De Freitas, 2015)、神经算术-逻辑单元 (neural arithmetic-logic units) (Trask et al., 2018; Madsen and Johansen, 2020) 和神经执行引擎 (neural execution engines) (Yan et al., 2020).

随着 GNN 体系结构的迅速发展, 模拟超线性 (superlinear) 复杂度的组合算法成为可能. Xu et al. (2019) 首创的算法对齐框架从理论上证明了 GNNs

与动态规划 ([Bellman, 1966](#)) 对齐, 动态规划是一种可以表达大多数算法的语言. 本文的一位作者同时从经验上表明, 设计和训练符合实践中算法不变量的 GNNs 是可能的 ([Veličković et al., 2019](#)). 此后, 通过迭代算法 (iterative algorithms ([Tang et al., 2020](#)))、线性算法 (linearithmic algorithms ([Freivalds et al., 2019](#)))、数据结构 (data structures ([Veličković et al., 2020](#))) 和持久存储器 (persistent memory ([Strathmann et al., 2021](#))) 实现了对齐. 这种模型在隐式规划器 (implicit planners) 中也有实际应用 ([Deac et al., 2020](#)), 突破了强化学习算法的空间.

同时, 在将神经网络用于物理模拟 (physics simulations) 方面取得了重大进展 ([Sanchez-Gonzalez et al., 2020](#); [Pfaff et al., 2020](#)). 这一方向为广义神经网络的设计提供了许多相同的建议. 这种对应是意料之中的: 给定算法可以被表述为离散时间模拟, 并且模拟通常被实现为逐步算法, 两个方向都需要保持相似种类的不变量.

与算法推理研究紧密相连的是外推法 (extrapolation). 对于神经网络来说, 这是一个臭名昭著的痛点, 因为它们的大部分成功故事都是在推广内分布时获得的; 即当在训练数据中发现的模式正确地预测了在测试数据中发现的模式时. 然而, 算法不变量必须被保留, 而与例如输入的大小或生成分布无关, 这意味着训练集可能不会覆盖实践中遇到的任何可能的场景. [Xu et al. \(2020b\)](#) 提出了一个几何论证, 证明了整流器激活对外推的要求: 需要对其组件和特征进行设计, 以使其组成模块 (如消息函数) 仅学习线性目标函数. [Bevilacqua et al. \(2021\)](#) 提出在因果推理的透镜下观察外推, 产生环境不变的图形表示.

7.13 几何深度学习

我们最后的历史评论是关于这篇文章的名字. “几何深度学习”一词最早是由本文的作者之一在 2015 年的 ERC 授权中提出的, 并在同名的《电气和电子工程师协会信号处理杂志》论文中得到推广 ([Bronstein et al., 2017](#)). 这篇论文宣告了“一个新领域正在诞生”的迹象, 尽管“有些谨慎”鉴于最近图

形神经网络的流行, 不变性和等方差思想在广泛的机器学习应用中的日益使用, 以及我们写这篇文章的事实, 认为这个预言至少部分实现可能是正确的. “4G: 格网、图形、群和规划” 这个名字是由马克斯·韦林 (*Max Welling*) 为 *ELLIS* 几何深度学习项目创造的, 该项目由本文的两位作者共同领衔. 诚然, 最后一个 “G” 有点牵强, 因为底层结构是流形 (*manifolds*) 和丛 (*bundles*), 而不是规划 (*gauges*). 对于这篇文章, 我们添加了另一个 “G”, 测地线, 参考度量不变量和流形的内在对称性.

致谢

本文通过不变性和对称性的几何透镜, 尝试总结和融合深度学习架构中几十年现有知识. 希望我们的观点将使新人和从业者更容易统揽这个领域, 使研究人员更容易融合新的架构, 作为我们蓝图的实例. 在某种程度上, 我们希望已经展示了“构建您所需要的架构所需的一切”——一部受启发的文字剧 *Vaswani et al. (2017)*.

正文的大部分是在 2020 年末和 2021 年初写的. 正如经常发生的那样, 我们对整本书是否有意义有成千上万的怀疑, 并利用同事提供的机会帮助我们打破“怯场”, 展示我们工作的早期版本, 这在 Petar 在剑桥的演讲 (由 Pietro Liò 提供) 和迈克尔在牛津 (由 Xiaowen Dong 提供) 和帝国理工学院 (由 Michael Huth 和 Daniel Rueckert 主持) 的演讲中看到了曙光. Petar 还在“埃尔兰根纲领”的发源地——弗里德里希-亚历山大-埃尔兰根-纽伦堡大学 (*Friedrich-Alexander-Universität Erlangen-Nürnberg*) 展示了我们的作品! ——感谢 Andreas Maier 的盛情邀请. 我们从这些会谈中得到的反馈对保持我们的高昂情绪以及进一步完善工作非常宝贵. 最后, 但同样重要的是, 我们感谢 ICLR 2021 的组委会, 我们的工作将在一个由迈克尔发表的主题演讲中被突出.

我们应该注意到, 协调如此大量的研究很少是由仅仅四个人的专业知识实现的. 因此, 我们要对所有认真研究了我们的案文的各个方面的研究人员给予应有的赞扬, 他们为我们提供了仔细的评论和参考资料: Yoshua Bengio, Charles Blundell, Andreea Deac, Fabian Fuchs, Francesco di Giovanni, Marco Gori, Raia Hadsell, Will Hamilton, Maksym Korablyov, Christian Merkwirth, Razvan Pascanu, Bruno Ribeiro, Anna Scaife, Jürgen Schmidhuber, Marwin Segler, Corentin Tallec, Ngân Vũ, Peter Wirnsberger and David Wong. 他们的专家式反馈对巩固我们的统一努力非常宝贵. 当然, 本文中的任何违规行为都是我们的责任. 这是一项正在进行的工作, 我们很高兴在任何阶段收到评论. 如果您发现任何错误或遗漏, 请联系我们.

参考文献

- Yonathan Aflalo and Ron Kimmel. Spectral multidimensional scaling. PNAS, 110(45):18052–18057, 2013.*
- Yonathan Aflalo, Haim Brezis, and Ron Kimmel. On the optimality of shape and data representation in the spectral domain. SIAM J. Imaging Sciences, 8(2):1141–1160, 2015.*
- Luis B Almeida. A learning rule for asynchronous perceptrons with feedback in a combinatorial environment. In Artificial neural networks: concept learning, pages 102–111. 1990.*
- Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. arXiv:2006.05205, 2020.*
- Sl Amari. Feature spaces which admit and detect invariant signal transformations. In Joint Conference on Pattern Recognition, 1978.*
- Philip W Anderson. More is different. Science, 177(4047):393–396, 1972.*
- Mathieu Andreux, Emanuele Rodola, Mathieu Aubry, and Daniel Cremers. Anisotropic Laplace-Beltrami operators for shape analysis. In ECCV, 2014.*
- Salim Arslan, Sofia Ira Ktena, Ben Glocker, and Daniel Rueckert. Graph saliency maps through spectral convolutional networks: Application to sex*

classification with brain connectivity. In Graphs in Biomedical Image Analysis and Integrating Medical Imaging and Non-Imaging Modalities, pages 3–13. 2018.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. arXiv:1607.06450, 2016.

László Babai. Graph isomorphism in quasipolynomial time. In ACM Symposium on Theory of Computing, 2016.

László Babai and Eugene M Luks. Canonical labeling of graphs. In ACM Symposium on Theory of computing, 1983.

Francis Bach. Breaking the curse of dimensionality with convex neural networks. JMLR, 18(1):629–681, 2017.

Adrià Puigdomènech Badia, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, and Charles Blundell. Agent57: Outperforming the atari human benchmark. In ICML, 2020.

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. Trans. PAMI, 39(12):2481–2495, 2017.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. arXiv:1409.0473, 2014.

Philip Ball. In retrospect: On the six-cornered snowflake. Nature, 480(7378):455–455, 2011.

Bassam Bamieh. Discovering transforms: A tutorial on circulant matrices, circular convolution, and the discrete fourier transform. arXiv:1805.05533, 2018.

- Stefan Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. Fundamenta Mathematicae, 3(1):133–181, 1922.*
- Albert-László Barabási, Natali Gulbahce, and Joseph Loscalzo. Network medicine: a network-based approach to human disease. Nature Reviews Genetics, 12(1):56–68, 2011.*
- Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. IEEE Trans. Information Theory, 39(3):930–945, 1993.*
- Igor I Baskin, Vladimir A Palyulin, and Nikolai S Zefirov. A neural device for searching direct correlations between structures and properties of chemical compounds. J. Chemical Information and Computer Sciences, 37(4):715–721, 1997.*
- Peter W Battaglia, Razvan Pascanu, Matthew Lai, Danilo Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. arXiv:1612.00222, 2016.*
- Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. arXiv:1806.01261, 2018.*
- Dominique Beaini, Saro Passaro, Vincent Létourneau, William L Hamilton, Gabriele Corso, and Pietro Liò. Directional graph networks. arXiv:2010.02863, 2020.*
- Richard Bellman. On a routing problem. Quarterly of Applied Mathematics, 16(1):87–90, 1958.*
- Richard Bellman. Dynamic programming. Science, 153(3731):34–37, 1966.*

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. *Learning long-term dependencies with gradient descent is difficult*. IEEE Trans. Neural Networks, 5(2):157–166, 1994.

Marcel Berger. *A panoramic view of Riemannian geometry*. Springer, 2012.

Pierre Besson, Todd Parrish, Aggelos K Katsaggelos, and S Kathleen Bandt. *Geometric deep learning on brain shape predicts sex and age*. BioRxiv:177543, 2020.

Beatrice Bevilacqua, Yangze Zhou, and Bruno Ribeiro. *Size-invariant graph representations for graph classification extrapolations*. arXiv:2103.05045, 2021.

Guy Blanc, Neha Gupta, Gregory Valiant, and Paul Valiant. *Implicit regularization for deep neural networks driven by an ornstein-uhlenbeck like process*. In COLT, 2020.

Cristian Bodnar, Fabrizio Frasca, Yu Guang Wang, Nina Otter, Guido Montúfar, Pietro Liò, and Michael Bronstein. *Weisfeiler and lehman go topological: Message passing simplicial networks*. arXiv:2103.03212, 2021.

Alexander Bogatskiy, Brandon Anderson, Jan Offermann, Marwah Roussi, David Miller, and Risi Kondor. *Lorentz group equivariant neural network for particle physics*. In ICML, 2020.

Karol Borsuk. *Drei sätze über die n -dimensionale euklidische sphäre*. Fundamenta Mathematicae, 20(1):177–190, 1933.

Davide Boscaini, Davide Eynard, Drosos Kourounis, and Michael M Bronstein. *Shape-from-operator: Recovering shapes from intrinsic operators*. Computer Graphics Forum, 34(2):265–274, 2015.

- Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In NIPS, 2016a.*
- Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Michael M Bronstein, and Daniel Cremers. Anisotropic diffusion descriptors. Computer Graphics Forum, 35(2):431–441, 2016b.*
- Sébastien Bogleux, Luc Brun, Vincenzo Carletti, Pasquale Foggia, Benoit Gaüzere, and Mario Vento. A quadratic assignment formulation of the graph edit distance. arXiv:1512.07494, 2015.*
- Giorgos Bouritsas, Fabrizio Frasca, Stefanos Zafeiriou, and Michael M Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. arXiv:2006.09252, 2020.*
- Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. PNAS, 103(5):1168–1172, 2006.*
- Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Numerical geometry of non-rigid shapes. Springer, 2008.*
- Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond Euclidean data. IEEE Signal Processing Magazine, 34(4):18–42, 2017.*
- Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. arXiv:2005.14165, 2020.*
- Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. IEEE transactions on pattern analysis and machine intelligence, 35(8):1872–1886, 2013.*

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. *Spectral networks and locally connected networks on graphs*. In *ICLR*, 2013.

Quentin Cappart, Didier Chételat, Elias Khalil, Andrea Lodi, Christopher Morris, and Petar Veličković. *Combinatorial optimization and reasoning with graph neural networks*. arXiv:2102.09544, 2021.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. *Neural ordinary differential equations*. arXiv:1806.07366, 2018.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. *A simple framework for contrastive learning of visual representations*. In *ICML*, 2020.

Albert Chern, Felix Knöppel, Ulrich Pinkall, and Peter Schröder. *Shape from metric*. *ACM Trans. Graphics*, 37(4):1–17, 2018.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. *Learning phrase representations using rnn encoder-decoder for statistical machine translation*. arXiv:1406.1078, 2014.

Nicholas Choma, Federico Monti, Lisa Gerhardt, Tomasz Palczewski, Zahra Ronaghi, Prabhat Prabhat, Wahid Bhimji, Michael M Bronstein, Spencer R Klein, and Joan Bruna. *Graph neural networks for icecube signal classification*. In *ICMLA*, 2018.

Taco Cohen and Max Welling. *Group equivariant convolutional networks*. In *ICML*, 2016.

Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. *Gauge equivariant convolutional networks and the icosahedral CNN*. In *ICML*, 2019.

Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. *Spherical cnns*. arXiv:1801.10130, 2018.

Tim Cooijmans, Nicolas Ballas, César Laurent, Çağlar Gülgehre, and Aaron Courville. Recurrent batch normalization. arXiv:1603.09025, 2016.

Etienne Corman, Justin Solomon, Mirela Ben-Chen, Leonidas Guibas, and Maks Ovsjanikov. Functional characterization of intrinsic and extrinsic geometry. ACM Trans. Graphics, 36(2):1–17, 2017.

Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. Introduction to algorithms. MIT press, 2009.

Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. Principal neighbourhood aggregation for graph nets. arXiv:2004.05718, 2020.

Luca Cosmo, Anees Kazi, Seyed-Ahmad Ahmadi, Nassir Navab, and Michael Bronstein. Latent-graph learning for disease prediction. In MICCAI, 2020.

Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. arXiv:2003.04630, 2020.

Miles D Cranmer, Rui Xu, Peter Battaglia, and Shirley Ho. Learning symbolic physics with graph networks. arXiv:1909.05862, 2019.

Guillem Cucurull, Konrad Wagstyl, Arantxa Casanova, Petar Veličković, Estrid Jakobsen, Michal Drozdal, Adriana Romero, Alan Evans, and Yoshua Bengio. Convolutional neural networks for mesh-based parcellation of the cerebral cortex. 2018.

George Cybenko. Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems, 2(4):303–314, 1989.

Hanjun Dai, Bo Dai, and Le Song. Discriminative embeddings of latent variable models for structured data. In ICML, 2016.

Jeffrey De Fauw, Joseph R Ledsam, Bernardino Romera-Paredes, Stanislav Nikolov, Nenad Tomasev, Sam Blackwell, Harry Askham, Xavier Glorot, Brendan O' Donoghue, Daniel Visentin, et al. Clinically applicable deep learning for diagnosis and referral in retinal disease. Nature Medicine, 24(9):1342–1350, 2018.

Pim de Haan, Maurice Weiler, Taco Cohen, and Max Welling. Gauge equivariant mesh CNNs: Anisotropic convolutions on geometric graphs. In NeurIPS, 2020.

Andreea Deac, Petar Veličković, and Pietro Sormanni. Attentive cross-modal paratope prediction. Journal of Computational Biology, 26(6):536–545, 2019.

Andreea Deac, Petar Veličković, Ognjen Milinković, Pierre-Luc Bacon, Jian Tang, and Mladen Nikolić. Xlvin: executed latent value iteration nets. arXiv:2010.13146, 2020.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. NIPS, 2016.

Austin Derrow-Pinion, Jennifer She, David Wong, Oliver Lange, Todd Hester, Luis Perez, Marc Nunkesser, Seongjae Lee, Xueying Guo, Peter W Battaglia, Vishal Gupta, Ang Li, Zhongwen Xu, Alvaro Sanchez-Gonzalez, Yujia Li, and Petar Veličković. Traffic Prediction with Graph Neural Networks in Google Maps. 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805, 2018.

David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Con-

- volutional networks on graphs for learning molecular fingerprints*. NIPS, 2015.
- Vijay Prakash Dwivedi and Xavier Bresson. *A generalization of transformer networks to graphs*. arXiv:2012.09699, 2020.
- Asi Elad and Ron Kimmel. *On bending invariant signatures for surfaces*. Trans. PAMI, 25(10):1285–1295, 2003.
- Jeffrey L Elman. *Finding structure in time*. Cognitive Science, 14(2):179–211, 1990.
- Carlos Esteves, Ameesh Makadia, and Kostas Daniilidis. *Spin-weighted spherical CNNs*. arXiv:2006.10731, 2020.
- Xiaomin Fang, Jizhou Huang, Fan Wang, Lingke Zeng, Haijin Liang, and Haifeng Wang. *ConSTGAT: Contextual spatial-temporal graph attention network for travel time estimation at baidu maps*. In KDD, 2020.
- Matthias Fey, Jan-Gin Yuen, and Frank Weichert. *Hierarchical inter-message passing for learning on molecular graphs*. arXiv:2006.12179, 2020.
- Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. *Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data*. In ICML, 2020.
- Jon Folkman. *Regular line-symmetric graphs*. Journal of Combinatorial Theory, 3(3):215–232, 1967.
- Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. *Learning discrete structures for graph neural networks*. In ICML, 2019.
- Paolo Frasconi, Marco Gori, and Alessandro Sperduti. *A general framework for adaptive processing of data structures*. IEEE Trans. Neural Networks, 9(5):768–786, 1998.

Kārlis Freivalds, Emīls Ozoliņš, and Agris Šostaks. Neural shuffle-exchange networks—sequence processing in $o(n \log n)$ time. arXiv:1907.07897, 2019.

Fabian B Fuchs, Daniel E Worrall, Volker Fischer, and Max Welling. SE(3)-transformers: 3D roto-translation equivariant attention networks. arXiv:2006.10503, 2020.

Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In Competition and Cooperation in Neural Nets, pages 267–285. Springer, 1982.

Pablo Gainza, Freyr Sverrisson, Federico Monti, Emanuele Rodola, D Boscaini, MM Bronstein, and BE Correia. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. Nature Methods, 17(2):184–192, 2020.

Fernando Gama, Alejandro Ribeiro, and Joan Bruna. Diffusion scattering transforms on graphs. In ICLR, 2019.

Fernando Gama, Joan Bruna, and Alejandro Ribeiro. Stability properties of graph neural networks. IEEE Trans. Signal Processing, 68:5680–5695, 2020.

Hongchang Gao, Jian Pei, and Heng Huang. Conditional random field enhanced graph convolutional neural networks. In KDD, 2019.

Alberto García-Durán and Mathias Niepert. Learning graph representations with embedding propagation. arXiv:1710.03059, 2017.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. arXiv preprint arXiv:1505.07376, 2015.

Thomas Gaudelet, Ben Day, Arian R Jamasb, Jyothish Soman, Cristian Regep, Gertrude Liu, Jeremy BR Hayter, Richard Vickers, Charles

- Roberts, Jian Tang, et al. Utilising graph machine learning within drug discovery and development. arXiv:2012.05716, 2020.*
- Felix A Gers and Jürgen Schmidhuber. Recurrent nets that time and count. In IJCNN, 2000.*
- Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. arXiv:1704.01212, 2017.*
- Ross Girshick. Fast R-CNN. In CVPR, 2015.*
- Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.*
- Vladimir Gligorijevic, P Douglas Renfrew, Tomasz Kosciolk, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C Taylor, Ian M Fisk, Hera Vlamakis, et al. Structure-based function prediction using graph convolutional networks. bioRxiv:786236, 2020.*
- Christoph Goller and Andreas Kuchler. Learning task-dependent distributed representations by backpropagation through structure. In ICNN, 1996.*
- Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. arXiv:1406.2661, 2014.*
- Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In IJCNN, 2005.*
- Alex Graves. Generating sequences with recurrent neural networks. arXiv:1308.0850, 2013.*
- Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. arXiv:1410.5401, 2014.*

Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. Nature, 538(7626): 471–476, 2016.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. arXiv:2006.07733, 2020.

Mikhael Gromov. Structures métriques pour les variétés riemanniennes. Cedric, 1981.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In KDD, 2016.

Suriya Gunasekar, Blake E Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Implicit regularization in matrix factorization. In NIPS, 2017.

Deisy Morselli Gysi, Ítalo Do Valle, Marinka Zitnik, Asher Ameli, Xiao Gan, Onur Varol, Helia Sanchez, Rebecca Marlene Baron, Dina Ghiassian, Joseph Loscalzo, et al. Network medicine framework for identifying drug repurposing opportunities for COVID-19. arXiv:2004.07229, 2020.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In NIPS, 2017.

Junheng Hao, Tong Zhao, Jin Li, Xin Luna Dong, Christos Faloutsos, Yizhou Sun, and Wei Wang. P-companion: A principled framework for diversified complementary product recommendation. In Information & Knowledge Management, 2020.

Moritz Hardt and Tengyu Ma. *Identity matters in deep learning*. arXiv:1611.04231, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Deep residual learning for image recognition*. In CVPR, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. *Mask r-cnn*. In CVPR, 2017.

Claude Adrien Helvétius. *De l'esprit*. Durand, 1759.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. *Learning deep representations by mutual information estimation and maximization*. In ICLR, 2019.

Sepp Hochreiter. *Untersuchungen zu dynamischen neuronalen Netzen*. PhD thesis, Technische Universität München, 1991.

Sepp Hochreiter and Jürgen Schmidhuber. *Long short-term memory*. Neural Computation, 9(8):1735–1780, 1997.

Kurt Hornik. *Approximation capabilities of multilayer feedforward networks*. Neural Networks, 4(2):251–257, 1991.

Yedid Hoshen. *Vain: Attentional multi-agent predictive modeling*. arXiv:1706.06122, 2017.

Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. *Strategies for pre-training graph neural networks*. In ICLR, 2020.

David H Hubel and Torsten N Wiesel. *Receptive fields of single neurones in the cat's striate cortex*. J. Physiology, 148(3):574–591, 1959.

- Michael Hutchinson, Charline Le Lan, Sheheryar Zaidi, Emilien Dupont, Yee Whye Teh, and Hyunjik Kim. LieTransformer: Equivariant self-attention for Lie groups. arXiv:2012.10885, 2020.*
- Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML, 2015.*
- Haris Iqbal. Harisqbal88/plotneuralnet v1.0.0, December 2018. URL <https://doi.org/10.5281/zenodo.2526396>.*
- Sarah Itani and Dorina Thanou. Combining anatomical and functional networks for neuropathology identification: A case study on autism spectrum disorder. Medical Image Analysis, 69:101986, 2021.*
- Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In ICML, 2018.*
- Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Hierarchical generation of molecular graphs using structural motifs. In ICML, 2020.*
- Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. Scientific Data, 3(1):1–9, 2016.*
- Michael I Jordan. Serial order: A parallel distributed processing approach. In Advances in Psychology, volume 121, pages 471–495. 1997.*
- Chaitanya Joshi. Transformers are graph neural networks. The Gradient, 2020.*
- Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In ICML, 2015.*
- Lukasz Kaiser and Ilya Sutskever. Neural GPUs learn algorithms. arXiv:1511.08228, 2015.*

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. *Neural machine translation in linear time*. arXiv:1610.10099, 2016.

Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman, and Koray Kavukcuoglu. *Efficient neural audio synthesis*. In ICML, 2018.

Ken-Ichi Kanatani. *Group-theoretical methods in image understanding*. Springer, 2012.

Zachi Karni and Craig Gotsman. *Spectral compression of mesh geometry*. In Proc. Computer Graphics and Interactive Techniques, 2000.

Anees Kazi, Luca Cosmo, Nassir Navab, and Michael Bronstein. *Differentiable graph module (DGM) graph convolutional networks*. arXiv:2002.04999, 2020.

Henry Kenlay, Dorina Thanou, and Xiaowen Dong. *Interpretable stability bounds for spectral graph filters*. arXiv:2102.09587, 2021.

Ron Kimmel and James A Sethian. *Computing geodesic paths on manifolds*. PNAS, 95(15):8431–8435, 1998.

Diederik P Kingma and Jimmy Ba. *Adam: A method for stochastic optimization*. arXiv:1412.6980, 2014.

Diederik P Kingma and Max Welling. *Auto-encoding variational bayes*. arXiv:1312.6114, 2013.

Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. *Neural relational inference for interacting systems*. In ICML, 2018.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. arXiv:1609.02907, 2016a.

Thomas N Kipf and Max Welling. Variational graph auto-encoders. arXiv:1611.07308, 2016b.

Dmitry B Kireev. Chemnet: a novel neural network based method for graph/property mapping. J. Chemical Information and Computer Sciences, 35(2):175–180, 1995.

Johannes Klicpera, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. arXiv:2003.03123, 2020.

Iasonas Kokkinos, Michael M Bronstein, Roe Litman, and Alex M Bronstein. Intrinsic shape context descriptors for deformable shapes. In CVPR, 2012.

Patrick T Komiske, Eric M Metodiev, and Jesse Thaler. Energy flow networks: deep sets for particle jets. Journal of High Energy Physics, 2019 (1):121, 2019.

Ilya Kostrikov, Zhongshi Jiang, Daniele Panozzo, Denis Zorin, and Joan Bruna. Surface networks. In CVPR, 2018.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.

Sofia Ira Ktena, Sarah Parisot, Enzo Ferrante, Martin Rajchl, Matthew Lee, Ben Glocker, and Daniel Rueckert. Distance metric learning using graph convolutional networks: Application to functional brain networks. In MICCAI, 2017.

Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In CVPR, 2020.

- Karol Kurach, Marcin Andrychowicz, and Ilya Sutskever. *Neural random-access machines*. arXiv:1511.06392, 2015.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. *Gradient-based learning applied to document recognition*. Proc. IEEE, 86(11):2278–2324, 1998.
- Reiner Lenz. *Group theoretical methods in image processing*. Springer, 1990.
- Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. *Multilayer feedforward networks with a nonpolynomial activation function can approximate any function*. Neural Networks, 6(6):861–867, 1993.
- Ron Levie, Federico Monti, Xavier Bresson, and Michael M Bronstein. *Cayleynets: Graph convolutional neural networks with complex rational spectral filters*. IEEE Trans. Signal Processing, 67(1):97–109, 2018.
- Ron Levie, Elwin Isufi, and Gitta Kutyniok. *On the transferability of spectral graph filters*. In Sampling Theory and Applications, 2019.
- Bruno Lévy. *Laplace-Beltrami eigenfunctions towards an algorithm that “understands” geometry*. In Proc. Shape Modeling and Applications, 2006.
- Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. *Gated graph sequence neural networks*. arXiv:1511.05493, 2015.
- Or Litany, Alex Bronstein, Michael Bronstein, and Ameesh Makadia. *Deformable shape completion with graph convolutional autoencoders*. In CVPR, 2018.
- Roe Litman and Alexander M Bronstein. *Learning spectral descriptors for deformable shape correspondence*. Trans. PAMI, 36(1):171–180, 2013.
- Hsueh-Ti Derek Liu, Alec Jacobson, and Keenan Crane. *A Dirac operator for extrinsic shape analysis*. Computer Graphics Forum, 36(5):139–149, 2017.

Siwei Lyu and Eero P Simoncelli. Nonlinear image representation using divisive normalization. In CVPR, 2008.

Richard H MacNeal. The solution of partial differential equations by means of electrical networks. PhD thesis, California Institute of Technology, 1949.

Andreas Madsen and Alexander Rosenberg Johansen. Neural arithmetic units. arXiv:2001.05016, 2020.

Soha Sadat Mahdi, Nele Nauwelaers, Philip Joris, Giorgos Bouritsas, Shunwang Gong, Sergiy Bokhnyak, Susan Walsh, Mark Shriver, Michael Bronstein, and Peter Claes. 3d facial matching by spiral convolutional metric learning and a biometric fusion-net of demographic properties. arXiv:2009.04746, 2020.

VE Maiorov. On best approximation by ridge functions. Journal of Approximation Theory, 99(1):68–94, 1999.

Ameesh Makadia, Christopher Geyer, and Kostas Daniilidis. Correspondence-free structure from motion. IJCV, 75(3):311–327, 2007.

Stéphane Mallat. A wavelet tour of signal processing. Elsevier, 1999.

Stéphane Mallat. Group invariant scattering. Communications on Pure and Applied Mathematics, 65(10):1331–1398, 2012.

Brandon Malone, Alberto Garcia-Duran, and Mathias Niepert. Learning representations of missing data for predicting patient outcomes. arXiv:1811.04752, 2018.

Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and equivariant graph networks. arXiv:1812.09902, 2018.

- Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. *Provably powerful graph networks*. arXiv:1905.11136, 2019.
- Jean-Pierre Marquis. *Category theory and klein’ s erlangen program*. In *From a Geometrical Point of View*, pages 9–40. Springer, 2009.
- Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. *Geodesic convolutional neural networks on Riemannian manifolds*. In *CVPR Workshops*, 2015.
- James Clerk Maxwell. *A dynamical theory of the electromagnetic field*. *Philosophical Transactions of the Royal Society of London*, (155):459–512, 1865.
- Jason D McEwen, Christopher GR Wallis, and Augustine N Mavor-Parker. *Scattering networks on the sphere for scalable and rotationally equivariant spherical cnns*. arXiv:2102.02828, 2021.
- Song Mei, Theodor Misiakiewicz, and Andrea Montanari. *Learning with invariances in random features and kernel models*. arXiv:2102.13219, 2021.
- Simone Melzi, Riccardo Spezialetti, Federico Tombari, Michael M Bronstein, Luigi Di Stefano, and Emanuele Rodolà. *Gframes: Gradient-based local reference frame for 3d shape matching*. In *CVPR*, 2019.
- Facundo Mémoli and Guillermo Sapiro. *A theoretical and computational framework for isometry invariant recognition of point cloud data*. *Foundations of Computational Mathematics*, 5(3):313–347, 2005.
- Christian Merkwirth and Thomas Lengauer. *Automatic generation of complementary descriptors with molecular graph networks*. *J. Chemical Information and Modeling*, 45(5):1159–1168, 2005.
- Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H Barr. *Discrete differential-geometry operators for triangulated 2-manifolds*. In *Visualization and Mathematics III*, pages 35–57. 2003.

Alessio Micheli. Neural network for graphs: A contextual constructive approach. IEEE Trans. Neural Networks, 20(3):498–511, 2009.

Karla L Miller, Fidel Alfaro-Almagro, Neal K Bangerter, David L Thomas, Essa Yacoub, Junqian Xu, Andreas J Bartsch, Saad Jbabdi, Stamatios N Sotiropoulos, Jesper LR Andersson, et al. Multimodal population brain imaging in the uk biobank prospective epidemiological study. Nature Neuroscience, 19(11):1523–1536, 2016.

Marvin Minsky and Seymour A Papert. Perceptrons: An introduction to computational geometry. MIT Press, 2017.

Jovana Mitrovic, Brian McWilliams, Jacob Walker, Lars Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms. arXiv:2010.07922, 2020.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, 2015.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In ICML, 2016.

Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In CVPR, 2017.

Federico Monti, Fabrizio Frasca, Davide Eynard, Damon Mannion, and Michael M Bronstein. Fake news detection on social media using geometric deep learning. arXiv:1902.06673, 2019.

Christopher Morris, Kristian Kersting, and Petra Mutzel. Glocalized

- Weisfeiler-Lehman graph kernels: Global-local feature maps of graphs. In ICDM, 2017.*
- Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In AAAI, 2019.*
- Christopher Morris, Gaurav Rattan, and Petra Mutzel. Weisfeiler and Leman go sparse: Towards scalable higher-order graph embeddings. In NeurIPS, 2020.*
- Michael C Mozer. A focused back-propagation algorithm for temporal pattern recognition. Complex Systems, 3(4):349–381, 1989.*
- Kevin Murphy, Yair Weiss, and Michael I Jordan. Loopy belief propagation for approximate inference: An empirical study. arXiv:1301.6725, 2013.*
- Ryan Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Relational pooling for graph representations. In ICML, 2019.*
- Ryan L Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs. arXiv:1811.01900, 2018.*
- Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In ICML, 2010.*
- John Nash. The imbedding problem for Riemannian manifolds. Annals of Mathematics, 63(1):20–63, 1956.*
- Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In COLT, 2015.*
- Emmy Noether. Invariante variationsprobleme. In König Gesellsch. d. Wiss. zu Göttingen, Math-Phys. Klasse, pages 235–257. 1918.*

Maks Ovsjanikov, Jian Sun, and Leonidas Guibas. Global intrinsic symmetries of shapes. Computer Graphics Forum, 27(5):1341–1348, 2008.

Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. ACM Trans. Graphics, 31(4):1–11, 2012.

Aditya Pal, Chantat Eksombatchai, Yitong Zhou, Bo Zhao, Charles Rosenberg, and Jure Leskovec. Pinnersage: Multi-modal user embedding framework for recommendations at pinterest. In KDD, 2020.

Sarah Parisot, Sofia Ira Ktena, Enzo Ferrante, Matthew Lee, Ricardo Guerrero, Ben Glocker, and Daniel Rueckert. Disease prediction using graph convolutional networks: application to autism spectrum disorder and alzheimer’s disease. Medical Image Analysis, 48:117–130, 2018.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In ICML, 2013.

Giuseppe Patanè. Fourier-based and rational graph filters for spectral processing. arXiv:2011.04055, 2020.

Judea Pearl. Probabilistic reasoning in intelligent systems: networks of plausible inference. Elsevier, 2014.

Roger Penrose. The road to reality: A complete guide to the laws of the universe. Random House, 2005.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In KDD, 2014.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning mesh-based simulation with graph networks. arXiv:2010.03409, 2020.

- Fernando J Pineda. Generalization of back propagation to recurrent and higher order neural networks. In NIPS, 1988.*
- Ulrich Pinkall and Konrad Polthier. Computing discrete minimal surfaces and their conjugates. Experimental Mathematics, 2(1):15–36, 1993.*
- Allan Pinkus. Approximation theory of the mlp model in neural networks. Acta Numerica, 8:143–195, 1999.*
- Tom J Pollard, Alistair EW Johnson, Jesse D Raffa, Leo A Celi, Roger G Mark, and Omar Badawi. The eicu collaborative research database, a freely available multi-center database for critical care research. Scientific Data, 5(1):1–13, 2018.*
- Javier Portilla and Eero P Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. International journal of computer vision, 40(1):49–70, 2000.*
- Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In CVPR, 2017.*
- Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In WSDM, 2018.*
- H Qu and L Goukos. Particlenet: jet tagging via particle clouds. arXiv:1902.08570, 2019.*
- Meng Qu, Yoshua Bengio, and Jian Tang. GMNN: Graph Markov neural networks. In ICML, 2019.*
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.*

- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9, 2019.*
- Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. Generating 3D faces using convolutional mesh autoencoders. In ECCV, 2018.*
- Dan Raviv, Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Symmetries of non-rigid shapes. In ICCV, 2007.*
- Noam Razin and Nadav Cohen. Implicit regularization in deep learning may not be explainable by norms. arXiv:2005.06398, 2020.*
- Scott Reed and Nando De Freitas. Neural programmer-interpreters. arXiv:1511.06279, 2015.*
- Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv:1506.01497, 2015.*
- Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In ICML, 2015.*
- Maximilian Riesenhuber and Tomaso Poggio. Hierarchical models of object recognition in cortex. Nature neuroscience, 2(11):1019–1025, 1999.*
- AJ Robinson and Frank Fallside. The utility driven dynamic error propagation network. University of Cambridge, 1987.*
- Emma Rocheteau, Pietro Liò, and Stephanie Hyland. Temporal pointwise convolutional networks for length of stay prediction in the intensive care unit. arXiv:2007.09483, 2020.*
- Emma Rocheteau, Catherine Tong, Petar Veličković, Nicholas Lane, and Pietro Liò. Predicting patient outcomes with graph representation learning. arXiv:2101.03940, 2021.*

- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In MICCAI, 2015.*
- Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. Psychological Review, 65(6):386, 1958.*
- Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs. arXiv:2006.10637, 2020.*
- Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. IJCV, 115(3):211–252, 2015.*
- Raif M Rustamov, Maks Ovsjanikov, Omri Azencot, Mirela Ben-Chen, Frédéric Chazal, and Leonidas Guibas. Map-based exploration of intrinsic shape differences and variability. ACM Trans. Graphics, 32(4):1–12, 2013.*
- Tim Salimans and Diederik P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. arXiv:1602.07868, 2016.*
- Alvaro Sanchez-Gonzalez, Victor Bapst, Kyle Cranmer, and Peter Battaglia. Hamiltonian graph networks with ODE integrators. arXiv:1909.12790, 2019.*
- Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In ICML, 2020.*
- Aliaksei Sandryhaila and José MF Moura. Discrete signal processing on graphs. IEEE Trans. Signal Processing, 61(7):1644–1656, 2013.*

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In NIPS, 2017.

Adam Santoro, Ryan Faulkner, David Raposo, Jack Rae, Mike Chrzanowski, Theophane Weber, Daan Wierstra, Oriol Vinyals, Razvan Pascanu, and Timothy Lillicrap. Relational recurrent neural networks. arXiv:1806.01822, 2018.

Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? arXiv:1805.11604, 2018.

Ryoma Sato, Makoto Yamada, and Hisashi Kashima. Random features strengthen graph neural networks. arXiv:2002.03155, 2020.

Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. $E(n)$ equivariant graph neural networks. arXiv:2102.09844, 2021.

Anna MM Scaife and Fiona Porter. Fanaroff-Riley classification of radio galaxies using group-equivariant convolutional neural networks. Monthly Notices of the Royal Astronomical Society, 2021.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. IEEE Trans. Neural Networks, 20(1):61–80, 2008.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. Nature, 588(7839):604–609, 2020.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv:1707.06347, 2017.

- Kristof T Schütt, Huziel E Sauceda, P-J Kindermans, Alexandre Tkatchenko, and K-R Müller. Schnet—a deep learning architecture for molecules and materials. The Journal of Chemical Physics, 148(24):241722, 2018.*
- Terrence J Sejnowski, Paul K Kienker, and Geoffrey E Hinton. Learning symmetry groups with hidden units: Beyond the perceptron. Physica D: Nonlinear Phenomena, 22(1-3):260–275, 1986.*
- Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Židek, Alexander WR Nelson, Alex Bridgland, et al. Improved protein structure prediction using potentials from deep learning. Nature, 577(7792):706–710, 2020.*
- Thomas Serre, Aude Oliva, and Tomaso Poggio. A feedforward architecture accounts for rapid categorization. Proceedings of the national academy of sciences, 104(15):6424–6429, 2007.*
- Ohad Shamir and Gal Vardi. Implicit regularization in relu networks with the square loss. arXiv:2012.05156, 2020.*
- John Shawe-Taylor. Building symmetries into feedforward networks. In ICANN, 1989.*
- John Shawe-Taylor. Symmetries and discriminability in feedforward network architectures. IEEE Trans. Neural Networks, 4(5):816–826, 1993.*
- Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. JMLR, 12(9), 2011.*
- Jonathan Shlomi, Peter Battaglia, and Jean-Roch Vlimant. Graph neural networks in particle physics. Machine Learning: Science and Technology, 2(2):021001, 2020.*
- David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs:*

Extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Processing Magazine, 30(3):83–98, 2013.

Hava T Siegelmann and Eduardo D Sontag. On the computational power of neural nets. Journal of Computer and System Sciences, 50(1):132–150, 1995.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. Nature, 529(7587):484–489, 2016.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. Nature, 550(7676):354–359, 2017.

Eero P Simoncelli and William T Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In Proceedings., International Conference on Image Processing, volume 3, pages 444–447. IEEE, 1995.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556, 2014.

Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A Hilbert space embedding for distributions. In ALT, 2007.

Stefan Spalević, Petar Veličković, Jovana Kovačević, and Mladen Nikolić. Hierachial protein function prediction with tail-GNNs. arXiv:2007.12804, 2020.

Alessandro Sperduti. Encoding labeled graphs by labeling RAAM. In NIPS, 1994.

- Alessandro Sperduti and Antonina Starita. *Supervised neural networks for the classification of structures*. IEEE Trans. Neural Networks, 8(3):714–735, 1997.
- Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. *Striving for simplicity: The all convolutional net*. arXiv:1412.6806, 2014.
- Balasubramaniam Srinivasan and Bruno Ribeiro. *On the equivalence between positional node embeddings and structural graph representations*. arXiv:1910.00452, 2019.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. *Dropout: a simple way to prevent neural networks from overfitting*. JMLR, 15(1):1929–1958, 2014.
- Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. *Highway networks*. arXiv:1505.00387, 2015.
- Kimberly Stachenfeld, Jonathan Godwin, and Peter Battaglia. *Graph networks with spectral message passing*. arXiv:2101.00079, 2020.
- Jonathan M Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M Donghia, Craig R MacNair, Shawn French, Lindsey A Carfrae, Zohar Bloom-Ackerman, et al. *A deep learning approach to antibiotic discovery*. Cell, 180(4):688–702, 2020.
- Heiko Strathmann, Mohammadamin Barekatain, Charles Blundell, and Petar Veličković. *Persistent message passing*. arXiv:2103.01043, 2021.
- Norbert Straumann. *Early history of gauge theories and weak interactions*. hep-ph/9609230, 1996.
- Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. *A concise and provably informative multi-scale signature based on heat diffusion*. Computer Graphics Forum, 28(5):1383–1392, 2009.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. arXiv:1409.3215, 2014.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In CVPR, 2015.

Corentin Tallec and Yann Ollivier. Can recurrent neural networks warp time? arXiv:1804.11188, 2018.

Hao Tang, Zhiao Huang, Jiayuan Gu, Bao-Liang Lu, and Hao Su. Towards scale-invariant graph-related problem solving by iterative homogeneous gnns. In NeurIPS, 2020.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In WWW, 2015.

Gabriel Taubin, Tong Zhang, and Gene Golub. Optimal surface smoothing as filter design. In ECCV, 1996.

Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Rémi Munos, Petar Veličković, and Michal Valko. Bootstrapped representation learning on graphs. arXiv:2102.06514, 2021.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds. arXiv:1802.08219, 2018.

Renate Tobies. Felix Klein—mathematician, academic organizer, educational reformer. In The Legacy of Felix Klein, pages 5–21. Springer, 2019.

Andrew Trask, Felix Hill, Scott Reed, Jack Rae, Chris Dyer, and Phil Blunsom. Neural arithmetic logic units. arXiv:1808.00508, 2018.

John Tromp and Gunnar Farnebäck. *Combinatorics of go*. In International Conference on Computers and Games, 2006.

Alexandre B Tsybakov. Introduction to nonparametric estimation. *Springer*, 2008.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. *Instance normalization: The missing ingredient for fast stylization*. arXiv:1607.08022, 2016.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. *Wavenet: A generative model for raw audio*. arXiv:1609.03499, 2016a.

Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. *Pixel recurrent neural networks*. In ICML, 2016b.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. *Attention is all you need*. In NIPS, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. *Graph Attention Networks*. ICLR, 2018.

Petar Veličković, Rex Ying, Matilde Padovano, Raia Hadsell, and Charles Blundell. *Neural execution of graph algorithms*. arXiv:1910.10593, 2019.

Petar Veličković, Lars Buesing, Matthew C Overlan, Razvan Pascanu, Oriol Vinyals, and Charles Blundell. *Pointer graph networks*. arXiv:2006.06380, 2020.

Petar Veličković, Wiliam Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. *Deep Graph Infomax*. In ICLR, 2019.

Kirill Veselkov, Guadalupe Gonzalez, Shahad Aljifri, Dieter Galea, Reza Mirnezami, Jozef Youssef, Michael Bronstein, and Ivan Laponogov. Hyperfoods: Machine intelligent mapping of cancer-beating molecules in foods. Scientific Reports, 9(1):1–12, 2019.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. arXiv:1506.03134, 2015.

Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. In ICLR, 2016.

Ulrike von Luxburg and Olivier Bousquet. Distance-based classification with lipschitz functions. JMLR, 5:669–695, 2004.

Martin J Wainwright and Michael Irwin Jordan. Graphical models, exponential families, and variational inference. Now Publishers Inc, 2008.

Yu Wang and Justin Solomon. Intrinsic and extrinsic operators for shape analysis. In Handbook of Numerical Analysis, volume 20, pages 41–115. Elsevier, 2019.

Yu Wang, Mirela Ben-Chen, Iosif Polterovich, and Justin Solomon. Steklov spectral geometry for extrinsic shape analysis. ACM Trans. Graphics, 38(1):1–21, 2018.

Yu Wang, Vladimir Kim, Michael Bronstein, and Justin Solomon. Learning geometric operators on meshes. In ICLR Workshops, 2019a.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph CNN for learning on point clouds. ACM Trans. Graphics, 38(5):1–12, 2019b.

Max Wardetzky. Convergence of the cotangent formula: An overview. Discrete Differential Geometry, pages 275–286, 2008.

- Max Wardetzky, Saurabh Mathur, Felix K lberer, and Eitan Grinspun. Discrete Laplace operators: no free lunch. In Symposium on Geometry Processing, 2007.*
- Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. arXiv:1807.02547, 2018.*
- Boris Weisfeiler and Andrei Leman. The reduction of a graph to canonical form and the algebra which appears therein. NTI Series, 2(9):12–16, 1968.*
- Paul J Werbos. Generalization of backpropagation with application to a recurrent gas market model. Neural Networks, 1(4):339–356, 1988.*
- Hermann Weyl. Elektron und gravitation. i. Zeitschrift f r Physik, 56(5-6): 330–352, 1929.*
- Hermann Weyl. Symmetry. Princeton University Press, 2015.*
- Marysia Winkels and Taco S Cohen. Pulmonary nodule detection in ct scans with equivariant cnns. Medical Image Analysis, 55:15–26, 2019.*
- Jeffrey Wood and John Shawe-Taylor. Representation theory and invariant neural networks. Discrete Applied Mathematics, 69(1-2):33–60, 1996.*
- Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In ICML, 2019.*
- Yuxin Wu and Kaiming He. Group normalization. In ECCV, 2018.*
- Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. arXiv:2002.07962, 2020a.*
- Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? arXiv:1810.00826, 2018.*

Keyulu Xu, Jingling Li, Mozhi Zhang, Simon S Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. *What can neural networks reason about?* arXiv:1905.13211, 2019.

Keyulu Xu, Jingling Li, Mozhi Zhang, Simon S Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. *How neural networks extrapolate: From feedforward to graph neural networks.* arXiv:2009.11848, 2020b.

Yujun Yan, Kevin Swersky, Danai Koutra, Parthasarathy Ranganathan, and Milad Heshemi. *Neural execution engines: Learning to execute subroutines.* arXiv:2006.08084, 2020.

Chen-Ning Yang and Robert L Mills. *Conservation of isotopic spin and isotopic gauge invariance.* Physical Review, 96(1):191, 1954.

Zhilin Yang, William Cohen, and Ruslan Salakhudinov. *Revisiting semi-supervised learning with graph embeddings.* In ICML, 2016.

Jonathan S Yedidia, William T Freeman, and Yair Weiss. *Bethe free energy, kikuchi approximations, and belief propagation algorithms.* NIPS, 2001.

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. *Graph convolutional neural networks for web-scale recommender systems.* In KDD, 2018.

Jiaxuan You, Rex Ying, and Jure Leskovec. *Position-aware graph neural networks.* In ICML, 2019.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. *Deep sets.* In NIPS, 2017.

Wojciech Zaremba and Ilya Sutskever. *Learning to execute.* arXiv:1410.4615, 2014.

- Wei Zeng, Ren Guo, Feng Luo, and Xianfeng Gu. *Discrete heat kernel determines discrete riemannian metric*. Graphical Models, 74(4):121–129, 2012.
- Jiani Zhang, Xingjian Shi, Junyuan Xie, Hao Ma, Irwin King, and Dit-Yan Yeung. *Gaan: Gated attention networks for learning on large and spatiotemporal graphs*. arXiv:1803.07294, 2018.
- Yuyu Zhang, Xinshi Chen, Yuan Yang, Arun Ramamurthy, Bo Li, Yuan Qi, and Le Song. *Efficient probabilistic logic reasoning with graph neural networks*. arXiv:2001.11850, 2020.
- Rong Zhu, Kun Zhao, Hongxia Yang, Wei Lin, Chang Zhou, Baole Ai, Yong Li, and Jingren Zhou. *Aligraph: A comprehensive graph neural network platform*. arXiv:1902.08730, 2019.
- Weicheng Zhu and Narges Razavian. *Variationally regularized graph-based representation learning for electronic health records*. arXiv:1912.03761, 2019.
- Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. *Deep graph contrastive representation learning*. arXiv:2006.04131, 2020.
- Marinka Zitnik, Monica Agrawal, and Jure Leskovec. *Modeling polypharmacy side effects with graph convolutional networks*. Bioinformatics, 34(13):i457–i466, 2018.