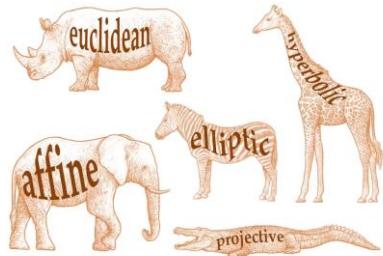
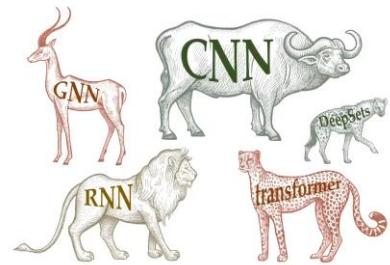

19th Century Zoo of Geometries



20th Century Zoo of Neural Network Architectures



几何深度学习

Michael M. Bronstein, Joan Bruna, Taco Cohen,

Petar Veličković

著

王广福, 安子玄

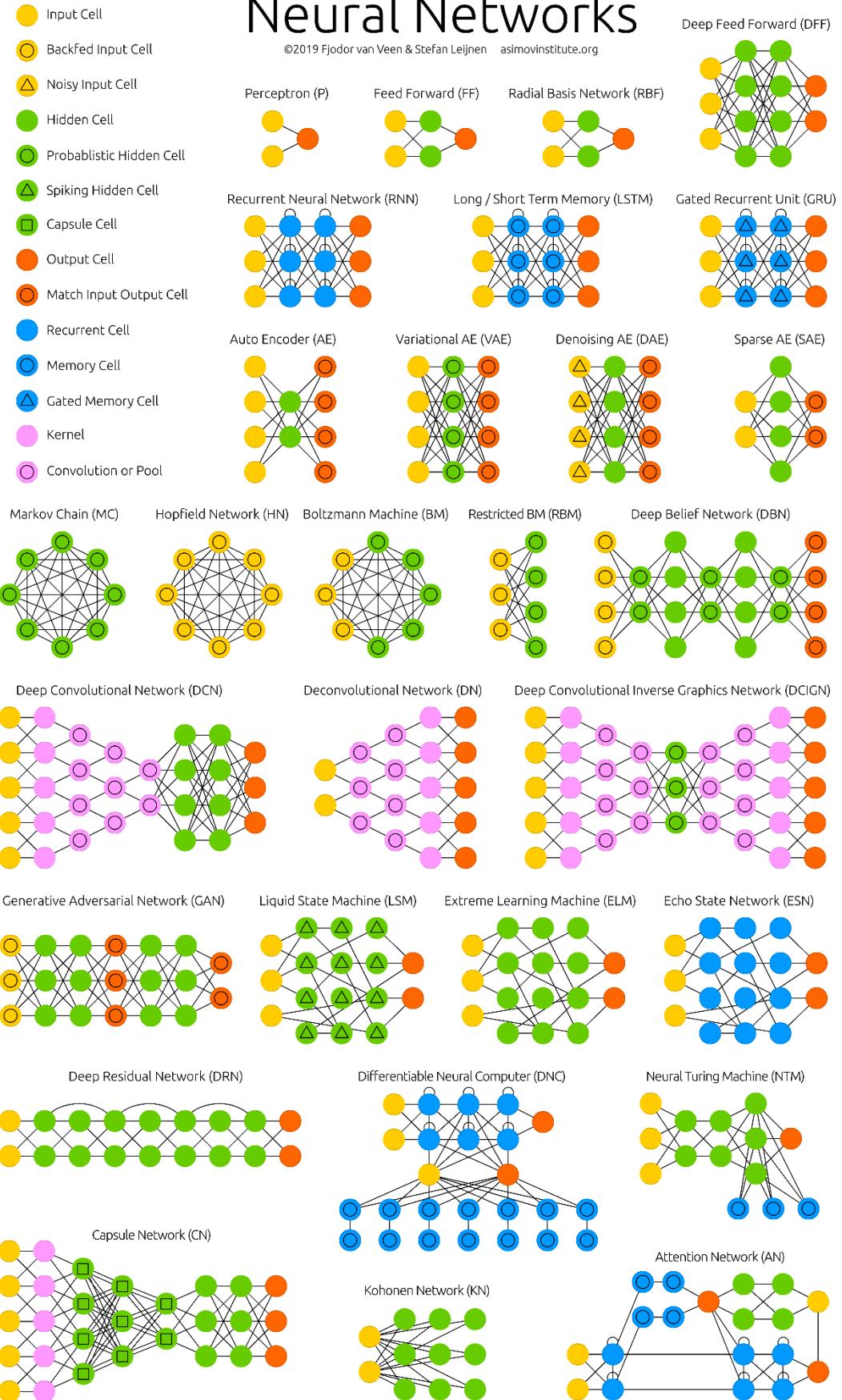
译

2021年5月4日

A mostly complete chart of
Neural Networks

©2019 Fjodor van Veen & Stefan Leijnen asimovinstitute.org

- Input Cell
- Backfed Input Cell
- △ Noisy Input Cell
- Hidden Cell
- Probabilistic Hidden Cell
- △ Spiking Hidden Cell
- Capsule Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- △ Gated Memory Cell
- Kernel
- Convolution or Pool



目 录

前 言	V
符号及标记	VII
1 引言	1
2 高维空间学习	3
2.1 函数正则化到归纳偏置 (Inductive Bias via Function Regularity)	3
2.2 维数灾难 (The Curse of Dimensionality)	5
3 几何先验	7
3.1 对称性、表示、不变性 (Symmtry, Representations and Invariance)	8
3.1.1 对称群 (Symmtry Groups)	8
3.1.2 群作用与群表示 (Group Action and Group Representations)	9
3.1.3 不变与等变函数 (Invariant and Equivariant Functions)	11
3.2 同构与自同构 (Isomorphism and Automorphism)	12
3.2.1 子群与结构层次 (Subgroups and Level of Structure)	12
3.2.2 同构与自同构 (Isomorphisms and Automorphisms)	13
3.3 变形稳定性 (Deformation Stability)	14
3.3.1 信号变形的稳定性 (Stability to Signal Deformations)	15
3.3.2 域变形的稳定性 (Stability to Domain Deformations)	16
3.4 尺度分离 (Scale Separation)	17
3.4.1 傅里叶变换与全局不变性 (Fourier Transform and Global Invariants)	17
3.4.2 多尺度表示 (Multiscale Representations)	18
3.4.3 多尺度表示的变形稳定性 (Deformation Stability of Multiscale Representations)	19
3.4.4 分离表示的先验 (Scale Separation Prior)	20
3.5 几何深度学习蓝图 (Geometric Deep Learning Blueprint)	20
3.5.1 几何深度学习蓝图表述	20
3.5.2 不同配置的几何深度学习	22
4 几何域: 5G	24
4.1 图 (Graphs) 与集合 (Sets)	24
4.2 网格 (Grids) 与欧氏空间 (Euclidean Space)	28
4.2.1 循环矩阵及卷积 (Circulant Matrices and Convolutions)	28

4.2.2	离散傅里叶变换推导 (Derivation of the Discrete Fourier Transform)	29
4.2.3	连续傅里叶变换推导 (Derivation of the Continuous Fourier Transform) ..	31
4.3	群 (Groups) 与齐次空间 (Homogeneous Space)	32
4.3.1	群卷积 (Group Convolution)	32
4.3.2	球状卷积 (Spherical Convolution)	33
4.4	测地线 (Geodesics) 与流形 (Manifolds)	35
4.4.1	黎曼流形 (Riemann Manifolds)	36
4.4.2	标量及矢量场 (Scalar and Vector Fields)	38
4.4.3	内参梯度 (Intrinsic Gradients)	39
4.4.4	测地线 (Geodesics)	39
4.4.5	平行移动 (Parallel Transport)	40
4.4.6	指数映射 (Exponential Map)	41
4.4.7	测地线距离 (Geodesic Distance)	42
4.4.8	等距同构 (Isometries)	42
4.4.9	内参对称 (Intrinsic Symmetries)	43
4.4.10	流形傅里叶分析 (Fourier Analysis on Manifolds)	43
4.4.11	流形谱域卷积 (Spectral Convolution on Manifolds)	45
4.4.12	流形空域卷积 (Spatial Convolution on Manifolds)	46
4.5	度规 (Gauges) 与丛 (Bundles)	47
4.5.1	切丛及结构群 (Tangent Bundles and the Structure Group)	48
4.5.2	度规对称 (Gauge Symmetries)	49
4.6	几何图 (Geometric Graph) 与面片模型 (Meshes)	51
4.6.1	拉普拉斯矩阵 (Laplacian Matrices)	51
4.6.2	Mesh 谱分析 (Spectral Analysis on Meshes)	54
4.6.3	算子及函数映射 Mesh (Meshes as Operators and Functional Maps)	55
5	几何深度学习模型	58
5.1	卷积神经网络 CNN	59
5.1.1	多尺度高效计算 (Efficient Multiscale Computation)	59
5.1.2	深度及残差网络 (Deep and Residual Network)	61
5.1.3	正则化 (Normalization)	62
5.1.4	数据增强 (Data Augmentation)	62
5.2	群等变卷积神经网络 (Group-equivariant CNNs)	63

5.2.1	离散群卷积 (Discrete Group Convolution)	63
5.2.2	变换+卷积方法 (Transform + Convolve Approach)	64
5.2.3	傅里叶频域球状 CNN (Spherical CNNs in the Fourier Domain)	65
5.3	图神经网络 GNN	65
5.3.1	图卷积网络 (Graph Convolution)	66
5.3.2	图注意力网络 (Graph Attention)	67
5.3.3	图信息传递网络 (Massage Passing)	67
5.4	Deep Sets、Transformer、潜图推理 (Latent Graph Inference)	67
5.4.1	空边集合 (Empty Edge Set)	68
5.4.2	完备边集合 (Complete Edge Set)	68
5.4.3	推断边集合 (Inferred Edge Set)	69
5.5	等变消息传递网络 (Equivariant Message Passing Networks)	69
5.5.1	不可规约表示 (Irreducible Representations)	71
5.5.2	正规表示 (Regular Representations)	71
5.6	内参面片模型卷积网络 (Intrinsic Mesh CNNs)	72
5.6.1	测地线片丁 (Geodesic Patch)	72
5.6.2	各向同性滤波器 (Isotropic Filter)	73
5.6.3	固定化度规 (Fixed Gauge)	73
5.6.4	角度聚合 (Angular Pooling)	73
5.6.5	度规等变滤波器 (Gauge-Equivariant Filter)	74
5.7	循环神经网络 RNN	75
5.7.1	简单 RNNs	76
5.7.2	RNN 平移等变群作用	77
5.7.3	RNN 深度	78
5.8	长短时记忆网络 LSTM	79
5.8.1	门控 RNN 的时间扭曲不变性	81
5.8.2	RNN 序列-序列学习	84
6	应用与分析	86
6.1	化学及药物研发 (Chemistry and Drug Design)	86
6.2	药物重定位 (Drug Repositioning)	87
6.3	生物蛋白质 (Protein Biology)	87
6.4	推荐系统及社交网络 (Recommender Systems and Social Networks)	88

6.5 流量预测 (Traffic Forecasting)	89
6.6 目标识别 (Object Recognition)	90
6.7 游戏 (Game Playing)	91
6.8 文字及音频综合 (Text and Speech Synthesis)	92
6.9 健康保险 (Healthcare)	93
6.10 粒子物理及天文物理 (Particle Physics and Astrophysics)	94
6.11 VR 及 AR (Virtual and Augmented Reality)	96
7 展望	97
7.1 数学及物理学中对称性 (Symmetry in Mathematics and Physics)	97
7.2 机器学习早期应用的对称 (Early Use of Symmetry in Machine Learning) ..	98
7.3 图神经网络 (Graph Neural Networks)	99
7.4 计算化学 (Computational Chemistry)	99
7.5 节点嵌入 (Node Embeddings)	100
7.6 概率图形学模型 (Probabilistic Graphical Models)	100
7.7 Weisfeiler-Lehman 正式化 (The Weisfeiler-Lehman Formalism)	101
7.8 高维方法 (High-Order Methods)	102
7.9 信号处理及调和分析 (Signal Processing and Harmonic Analysis)	103
7.10 图及 Mesh 上的信号处理 (Signal Processing on Graph and Meshes) ...	103
7.11 图形学及几何处理 (Computer Graphics and Geometry Processing)	104
7.12 算法推理 (Algorithmic Reasoning)	105
7.13 几何深度学习 (Geometric Deep Learning)	106
致 谢	108
参考文献	109

前 言

在欧几里得所处时代后接近两千年里，几何（“Geometry”）这一术语几乎是欧式几何的同义词，因为没有其他类型的几何度量出现。Lobachevesky、Bolyai、Gauss、Riemann 等人构建了非欧氏几何的诸多示例，宣告了欧氏几何的“独裁统治”在 19 世纪终于走到了尽头。在 19 世纪末期，这些欧氏空间及非欧空间几何的研究细分出了不同的领域，然而也伴随着数学家与哲学家们关于这些非欧几何与“唯一真正几何”（“One true Geometry”）本质（指欧式几何，译者注）的有效性及关系的争论。

德国一位在公元 1872 年、其 21 岁就被德国一所规模较小的大学—Bavarian University of Erlangen 任命为教授的年轻数学家 Felix Klein 明确地指明了这一争论的出路。在一份其编写的、后被收录到数学年刊的研究计划书 Erlangen Programme（德语，译者注）中，Klein 提出可以通过研究不变性来研究几何问题，不变性即在某些变换后保持不变的特性或者几何量，常常也被称为几何的对称性。这一方法使用群论作为几何的数学描述语言，通过定义合适的变换，Klein 清晰明确地展示了其所处时代的多种多样的几何度量。例如欧氏空间几何度量主要关注长度及角度，这也是由于欧氏变换（包括刚体旋转变换与刚体平移变换）将不改变这些几何度量的量；与此对应，仿射变换则主要关注角度，因为角度不随仿射变换群中元素（也即仿射变换，译者注）而变化。考虑对应的变换群时，不同的几何之间（指欧式几何与各种非欧几何，译者注）的关系就立马变得显而易见了，就像前述欧氏变换与仿射变换进行对比时，欧氏变换群（即李群，译者注）是仿射变换群的子群，而仿射变换群是投影变换群的子群。

Erlangen Programme 在几何领域的影响是十分深远的。另外，这种影响也传递到了其他研究领域，尤其是物理研究领域，在物理领域中，对称性原则不仅使得可以从第一对称性原则（一项令人震惊的结果，也被称为 Noether 定理）导出守恒定理，还使得基本粒子分类为对称群的不可约表示。分类理论（Category Theory）的创造者 Samuel Eilenberg 与 Saunders Mac Lane 指出，今天在纯数学研究领域已被普遍接受的分类理论，站在一个几何空间及其变换群可以表示为一个有着代数映射的类的角度看，也可以将其视为是一种 Klein 的 Erlangen Programme 的一种延续。

在本文写作之时（约指 2021 年 5 月前一段时间，译者注），深度学习研究领域重燃了 19 世纪广泛流形的几何研究。神经网络架构已经有不少方案，但是却缺少相应统一的架构构建法则。随着时间的流逝，人们也越来越难以理解不同神经网络架构之间关系，这也将难以避免地导致一些概念被不同应用领域的研究人员重新“发明”或者贴上不一样的标签。这一现状对于一个想要进入深度学习领域的初学者而言，理解这些冗余的概念与想法也绝对是一个梦魇。

前 言

在本文中，我们（指著作者们，译者注）谨慎地尝试在深度学习领域利用 Erlangen Programme 的思维框架，旨在提出一个系统化的深度学习方法并且这一方法能够囊括现有的其他深度学习框架（“Connecting the Dots”）。我们称这一几何化思想的尝试为几何深度学习，与 Klein 的想法一致，我们提出几何深度学习框架，通过从第一对称性及不变性的角度部署不同归纳偏置以及神经网络来导出不同架构。特别地，我们关注一系列用于分析非结构化的集合、网格、图、流形的神经网络，展示这些神经网络可以由我们提出的系统化几何深度学习框架以一种统一的方式进行理解，尤其是从其数据的结构与域的角度。

我们坚信本文将吸引包含深度学习研究人员、工程人员、以及一些技术爱好者们的目光。一个初学者可以把本文视为是一篇综述与几何神经网络的介绍；一个拥有多年神经网络研究经验的专家也可以根据一些基本的原则把本书当成一个对已熟悉的神经网络架构再次审视，也许能够发现不同网络之间令人惊讶的联系；工程人员也可以通过本书获得更多如何解决本领域面临的问题的认知。

在现代机器学习这样一个快节奏的领域，写一本试图系统化统一所有神经网络方法的书的风险在于，本书可能很快就过时了，甚至在没问世之前已经被落后于时代了。通过关注于基础的内容，我们希望这些讨论的关键概念能够超出他们目前应用的领域，正如 Claude Adrien Helvétius 说的，“The knowledge of certain we discuss will transcend their specific realisations principles easily compensates the lack of knowledge of certain facts.”（译文，原话为“la connaissance de certains principes supplée facilement à la connaissance de certains faits.”）。

符号及标记

Ω, u	域 (Domain), 域中一点 (Point on Domain)
$x(u) \in \mathcal{X}(\Omega, \mathcal{C})$	作用在域上的信号 $x: \Omega \rightarrow \mathcal{C}$
$f(x) \in \mathcal{F}(\mathcal{X}(\Omega))$	作用在域中信号上的函数 $f: \mathcal{X}(\Omega) \rightarrow \mathcal{Y}$
$\mathfrak{G}, \mathfrak{g}$	群, 群中元素
$\mathfrak{g}.u, \rho(\mathfrak{g})$	群作用, 群表示
$\mathbf{X} \in \mathcal{C}^{ \Omega \times s}$	离散域上的信号构成的矩阵
$\mathbf{x} \in \mathcal{C}^s$	离散域上信号构成的矢量
$x_{uj} \in \mathcal{C}$	离散域中信号 \mathbf{X} 作用于元素 $u \in \Omega$ 时的第 j 个分量
$\mathbf{F}(\mathbf{X})$	作用于离散域信号的方程, 返回一个矩阵表示的离散域信号
$\tau: \Omega \rightarrow \Omega$	域上自同构映射
$\eta: \Omega \rightarrow \Omega'$	两个域的同构映射
$\sigma: \mathcal{C} \times \mathcal{C}'$	激活方程 (逐元素非线性方程)
$G = (\mathcal{V}, \mathcal{E})$	图, 其中 \mathcal{V} 表示节点, \mathcal{E} 表示边
$\mathcal{T} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$	多边形 Mesh, 其中 \mathcal{V} 为顶点, \mathcal{E} 表示边, \mathcal{F} 为面
$x \star \theta$	信号 x 与滤波单元 θ 的卷积运算
S_v	左移 (或右移) 算子
φ_i	基函数
$T_u \Omega, T \Omega$	点处切矢量空间, 切矢量空间集合
$X \in T_u \Omega$	切矢量
$g_u(X, Y) = \langle df(X), df(Y) \rangle_u$	黎曼度量
$\ell(\gamma), \ell_{uv}$	曲线 γ 长度, 边 (u, v) 的离散度量

1 引言

过去的数十年已经见证了数据科学及机器学习领域的实验性的革命，以众多深度学习百花齐放为典型。事实上，很多在高维空间中学习的任务在过去曾被认为是难以处理的，例如计算机视觉（Computer Vision）、围棋比赛（Go）、蛋白质折叠，然而这些任务在合理的计算尺度的选择下（选择合理的维度，译者注）变得较为容易处理了。令人瞩目的是，深度学习的精髓思想构建在两个非常简单的算法原则上：（1）在应用表示学习或者特征学习的地方，其主要解决思路为层次性，对于特定任务而言，特征抓住了描述任务正则性的合适的构建元素。（2）通过局部的梯度下降来学习，典型的部署为后向传播。

尽管在高维空间中学习通用表达函数是一个艰难的估计问题，但是大多数的任务关注点并非寻求通用，由此也使得利用数据内在的低纬度特性以及物理世界数据本身的结构来精心预定义正则性成为可能。本书旨在可应用在广阔的应用中的统一几何原则明确这些正则性。

探寻并利用大规模系统的已知的对称性是一项应对维数灾难的有力且经典的良方，这也形成了大多数物理理论的基础。深度学习系统也不例外，在深度学习研究早期，学者们就曾利用神经网络来研究源自物理度量的低纬度几何特性，例如研究图像的网格、时间序列数据（例如音频，译者注）、分子的位置及动量，以及研究这些对象的对称性，诸如关于旋转与平移的对称性。通过我们对这些研究内容及对称性的呈现，我们将描述包含其他很多模型在内的这些常见模型，并分析这些模型蕴含的集合正则性的规律。

这样一个“几何统一”（“Geometric Unified”，也可译为度量统一，译者注），致力于传承 Erlangen Programme 项目的精髓思想，包含着两层目的：（1）这一统一框架为当前一些最为成功的神经网络架构（诸如各种 CNN、RNN、GNN、Transformer）的研究提供了统一的数学框架；（2）这一框架提供了整合以前的物理知识到深度神经网络的建设性流程规范以及提供了未来尚待发明的新架构的构建的主要原则。

继续阅读之前，值得说明的是我们的工作关注于表示学习的架构并利用所用数据的内在对称性。但诸如自监督学习、生成式模型、强化学习等众多令人激动的模型处理流程可能应用的领域并非我们关注的重点。因而，我们也不会深入地对诸如变分自编码器（variational autoencoder，[King and Welling, 2013]）、生成式对抗网络（generative adversarial networks，[Goodfellow et al., 2014]）、标准化流（Normalizing Flow，[Rezende and Mohamed, 2015]）、深度 Q 网络（deep Q-networks，[Mnih et al., 2015]）、趋近策略优化（proximal policy optimization，[Schulman et al., 2017]）、深度共有信息最大化（Deep mutual information maximisation，[Hjelm et al., 2019]）等这些有影响力的神经网络处理管线进行综述，我们也不会花费多数精力探讨各种优化及正则技术，例如 Adam（[Kingma and Ba, 2014]）、Dropout（[Srivastava, 2014]）、Batch Normalization（[Ioffe and Szegedy, 2015]）等方案。也就是说，我们相信我们聚焦的这

引言

些重要的原则对于所有领域均是十分重要的。

进一步来讲，尽管我们尽全力来展示利用我们提出的几何深度学习范本可应用于合理且广阔的网络，但是我们并不会尝试精确地描述现有的几何深度学习方法的全部研究内容。相对地，我们将深入研究几种典型且广为人知的架构，并藉此来展示这些已有的典型架构内含的主要原则，希望这些分析能够为读者提供足够且清晰对于几何深度学习的认知，并有助于读者利用这些认知来解决未来他们遇到的问题或者提出新的神经网络方案。

2 高维空间学习

对于最简单的表示形式的监督机器学习，通常其考虑一个包含 N 个观察的数据集 $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ ，并且通常假定这些数据是对待研究对象数据分布 P 的独立同分布的（Independent identical distributed, i.i.d）观测，这些数据通常记为 $\mathcal{X} \times \mathcal{Y}$ ，其中 \mathcal{X} 表示特征数据所在域， \mathcal{Y} 表示标记所在域。这样的数据通常假定 \mathcal{X} 在一个高维的空间中，通常人们假定 \mathcal{X} 在维度为 d 的欧式空间中，可记为： $\mathcal{X} = \mathbb{R}^d$ 。

在本书中，特征、信号等名词均指代同一事物，使用不同的说法是综合考虑到来自机器学习领域以及信号处理领域的研究人员的需要，不同领域的人员对于同一事物也有着不同的名词，因而我们交替使用这些名词，并希望读者尽快熟悉这些领域。欧氏几何通常也被称为“平直的”空间，这意味着两点之间直线最短，两点之间的直线上的点均在该空间中；而非欧氏空间则有不同，例如一个球面构成的空间，球面上两点之间的最短距离并非之间连成直线，因为这将意味着球体破一个洞，在连线上的点也不在属于球面这个空间了，球面表示的空间可以用黎曼几何来研究。总结来看，欧氏空间与黎曼几何研究的空间均是 Hausdorff 空间，也就是空间没有层叠，本段文字为译者注，有兴趣的读者可参考 Loring 编写的 *Introduction to Manifolds* 一书以及陈省身所著的微分几何入门一书。

进一步假定标签 y 是由一个位置的方程得到，也即 $y_i = f(x_i)$ ，则学习的问题就演变成了找到这样一个方程，通常可以用参数化的方程集合 $\mathcal{F} = \{f_{\theta \in \Theta}\}$ 来分段表示待求的方程 f 。神经网络就是一种常见的实现这样一个分段参数化的方程集合的方法，其中 $\theta \in \Theta$ 表示网络的权重。这样一个理想的假定下，实际上是要求数据 y 是没有噪声的，并且现代神经网络均采用了插值策略，也就是说估计的 $\tilde{f} \in \mathcal{F}$ 满足： $\tilde{f}(x_i) = f(x_i)$, $i = 1, \dots, N$ 。学习算法的性能度量通常使用来自统一分布 P 的测试数据的网络估计结果与我们预期的结果进行对比，即采用损失函数 $L(\cdot, \cdot)$ 对比这两项：

$$\mathcal{R}(\tilde{f}) := \mathbb{E}_P L(\tilde{f}(x), f(x)) \quad (2-1)$$

通常平方损失项在大多数神经网络方法中应用较多，即：

$$L(y, y') = \frac{1}{2} |y - y'|^2 \quad (2-2)$$

因而一个成功的深度学习策略应用包含有对于待估计函数 f 的合适正则化策略或者归纳偏置，这些策略正是贯穿在构建方程集合 \mathcal{F} 以及利用正则化的过程中。我们将在接下来的一节简要地介绍这一概念。

2.1 函数正则化到归纳偏置（Inductive Bias via Function Regularity）

大规模、高质量的数据集，以及结合合适的计算资源（MCU、CPU、GPU、FPGA

等, 译者注), 催生了众多能够针对这些海量数据插值计算的待求函数集合 \mathcal{F} 的设计, 现代的神经网络正是处理这样的数据而生。这种思维模式在神经网络方面起到了很好的作用, 因为即使采用简单的架构也能生成紧致的函数集合 (即万能函数生成器, 例如 Multi-Layer Perceptron, 译者注)。模拟任意的函数的能力的研究正是万能趋近定理 (Universal Approximation Theorems) 描述的内容; 在 1990 年左右一些重要的结果已经被一些应用数学家即计算机科学家们证明, 且广为流行, 诸如 [Cybenko, 1989]、[Hornik, 1991]、[Barron, 1993]、[Leshno et al., 1993]、[Marinov, 1999]、[Pinkus, 1999] 等人的研究。

紧致 (英文为 Compact) 这一概念等价于对于有限维度的希尔伯特空间中, 满足连续性要求以及有界性要求即可 (对于无穷维的情况, 由于本书研究的机器学习算法并不涉及, 因而在此不再给出说明, 有兴趣的读者可以参考泛函分析有关书籍), 本段为译者注。

但是万能模拟函数并没有暗示归纳偏置的不存在。给定一个假设空间 \mathcal{F} 及其万能趋近函数, 我们可以定义一个复杂度度量 $c: \mathcal{F} \rightarrow \mathbb{R}_+$, 则可重新定义我们的插值问题:

$$\tilde{f} \in \arg \min_{g \in \mathcal{F}} c(g) \quad s.t. \quad g(x_i) = f(x_i) \quad \text{for } i = 1, \dots, N \quad (2-3)$$

也就是说, 我们在假设空间中寻找最满足正则条件的函数。对于标准的函数空间, 可以将这一复杂度度量定义为范数 (norm), 使得 \mathcal{F} 为一个 Banach 空间, 从而我们可以利用泛函分析中的众多理论成果。在低维度空间中, 样条曲线是一个模拟函数的有力武器。样条曲线也可以定义为上述形式, 其中范数典型地反应了光滑性的思想, 例如三次样条曲线可以使用二阶切触阶 (即二次导函数, 切触阶这一概念多用于中文翻译的曲线研究中, 例如非均匀有理 B 样条曲线等, 译者注) 的平方和形式:

$$\int_{-\infty}^{+\infty} |f''(x)|^2 dx \quad (2-4)$$

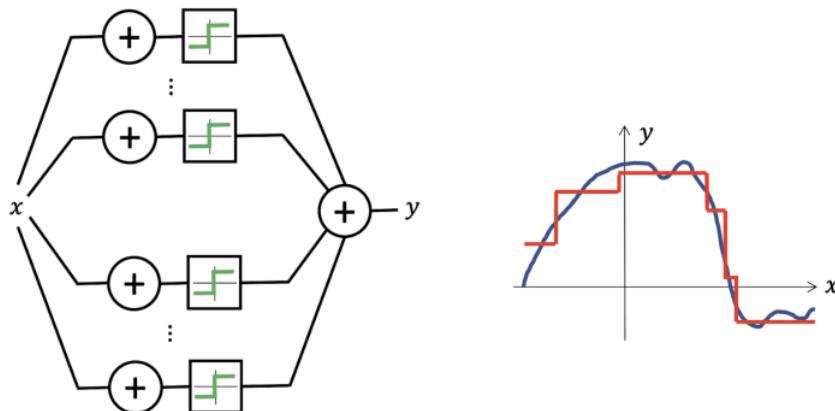


图 2.1 多层感知机 (Multi-Layer Perceptron, [Rosenblatt, 1958]), 最简单的前馈神经网络, 也可以作为万能函数模拟器, 其仅有一层, 可以模拟任意的阶跃函数的组合, 由此可以任意精度模拟任意的连续函数

对于神经网络而言, 复杂度度量可以用网络权重表示, 即: $c(f_\theta) = c(\theta)$ 。网络权重的 L_2 范数, 也被称为权重衰减, 与所谓的路径范数 (Path-norm) 等都是一些深度学习方中广为使用的选项。站在贝叶斯估计的角度, 这样的复杂度度量也可以解读为是目标函数的先

验的负对数度量。更一般地，这种复杂度度量可以由显式地集成在经验损失函数中（导出了所谓的结构风险最小化项）或者隐式地作为一个特定优化策略的结果。例如，众所周知，梯度下降方法可以使得待求解的最小二乘方程达到 L_2 范数最小化的效果。这一隐式的正则化方法的扩展应用的结果对于神经网络的影响正是当前研究的内容（例如，[Blanc et al., 2020]、[Shamir and Vardi, 2020]、[Razin and Cohen, 2020]、[Gunaseker et al., 2017]）。总而言之，我们要回答一个直观基本的问题：对于实际生活中的预测任务而言，我们怎样定义一个能够有效捕捉期望的正则性及复杂度的先验知识呢？

2.2 维数灾难 (The Curse of Dimensionality)

在低维空间中（通常指维数不大于 3 时），使用插值进行处理是经典信号处理任务的解决方法，并且可以通过使用愈加复杂的正则项非常精确地进行估计误差的数学控制（例如样条插值项、小波基函数（wavelet）、曲波基函数（curvelet）、脊波基函数（ridgelet）），然而对于高纬度空间中的任务而言情况完全不同了。

为了展示高纬度空间中任务处理与低维情况大相径庭这一想法的精髓，考虑一个可轻易地用于高纬度空间的经典正则思路：1-Lipschitz 函数（利普希茨函数，可参考实分析相关材料，译者注） $f: \mathcal{X} \rightarrow \mathbb{R}$ ，也即对于所有的 $x, x' \in \mathcal{X}$ 满足 $|f(x) - f(x')| \leq \|x - x'\|$ 条件的函数。这一假设仅要求目标函数在局部是光滑的，换句话说，如果给 x 一个扰动，使之到 x' ，则可用 $\|x - x'\|$ 度量扰动大小，在此扰动下，函数的输出变化不能比扰动变化大。若我们仅知道目标方程是满足 1-Lipschitz 条件的，那么我们需要多少观测数据才能充满信心地保证估计的函数 \tilde{f} 与待求函数 f 相近或一致呢？图 2.2 展示了一般对于上述问题的答案，即我们需要维度的指数次幂的常数倍的观测数据，满足 1-Lipschitz 条件的信号（即函数，不同说法，译者注）数量随着维度的增加而爆炸式增加：在一些尽管看起来维度不大的应用条件下，需要观测的数据规模也要比整个宇宙中所有原子的总数还要多！即使我们提高要求，不仅仅要求满足 1-Lipschitz 条件，更要求其是全局光滑的，例如要求处于可积的空间 Sobolev 空间 $\mathcal{H}^s(\Omega_d)$ ，这种维度增加带来所需观测数据规模的变化的问题仍然没什么改观。实际上，经典结果（[Tsybakov, 2008]）建立了最小化最大化趋近的方法，此外在当度为 $\epsilon^{-d/s}$ 的 Sobolev 空间中学习函数仅仅带来了额外的光滑性假设并优化了统计视角下的图景，Sobolev 空间学习也假设 $s \propto d$ ，这在现实中是一种不切实际的假设。

Sobolev 空间即二次可积的空间，二次可积是一项不弱的要求，并且对于诸多应用而言，我们均默认映射处于该空间，尤其是在应用 Dirichlet 能量法进行求解时。

全连接神经网络定义了更加灵活地正则化的函数空间，并通过考虑复杂度方程 c 关于权重系数的正则项来进行空间中函数的趋近。特别地，通过优先选择偏向稀疏的正则项，全连接神经网络能够打破维数灾难难题（[Bach, 2017]）。然而这也带来了相应的问题：这样的正则项相当于对目标函数 f 施加了很强的假定，例如目标函数 f 取决于输入数据在低

维度空间中投影的集合（如图 3.1）。在大多数现实应用中（诸如计算机视觉、语音分析、物理学、化学等），待求目标函数 f 展现出了长时空域的耦合，这种耦合不能够表示在低维度空间中的投影的集合中，因而全连接神经网络的假定变得不切实际了。由此，通过探究关于目标函数 f 的物理域的空间结构以及几何先验，我们有必要定义替代性的正则项意义及表示，这将在第 3 章中进行阐释。

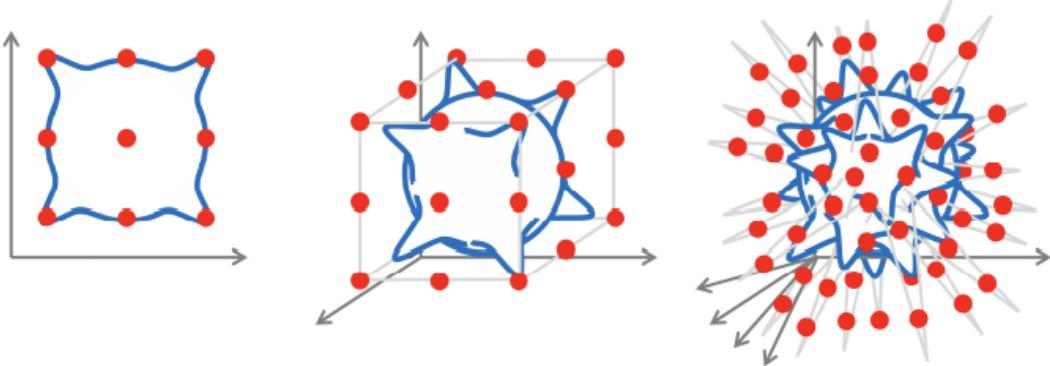


图 2.2 考虑一个 Lipschitz 方程： $f(x) = \sum_{j=1}^{2^d} z_j \phi(x - x_j)$, $z_j = \pm 1$, $x_j \in \mathbb{R}^d$

分布在各个象限， ϕ 表示局部支撑的 Lipschitz 突起方程（类似 Lagrange hat 函数，译者注），除非我们在 2^d 象限的空间中对方程进行观测，否则我们在预测时将遇到一个常量的误差而无法消除该误差。这一简单的集合论证由论文 Maximum Discrepancy [von Luxburg and Bousquet, 2004] 给出，其定义了

$$\text{Lipschitz 类: } \kappa(d) = \mathbb{E}_{x, x'} \sup_{f \in Lip(1)} \left| \frac{1}{N} \sum_i f(x_i) - \frac{1}{N} \sum_i f(x'_i) \right| \simeq N^{-1/d}$$

并以此度量 N 个采样得到的均值的差异性。确保 $\kappa(d) \simeq \epsilon$ 要求 $N = \Theta(\epsilon^{-d})$ 复杂度；对应的采样定义了域上一个 $\epsilon - net$ 即小邻域。对于直径为 1 的欧氏空间域而言，采样的数量正比于 ϵ^{-d} 。

3 几何先验

现代的数据分析（Data analysis）与高纬度学习（High-dimensional learning）是同一概念。虽然在 2.1 节中已经介绍了由于维数灾难在高维空间中数据学习通用范式通常是不可能实现的，对于具有物理结构的数据而言，在高维空间中学习其表示仍然有希望：我们可以使用两个根本的原则：（1）对称性；（2）尺度分离。在本书考虑的所有情况下，这一物理结构的数据通常在输入信号激励下来自研究对象所处的结构化域：我们将假定机器学习算法正是在作用在某些域的信号（函数）上。尽管在很多情况下域中点的线性组合并非是良好定义的（well-defined），我们仍然可以将信号线性地组合在一起并作用在域中点上，也就是说，信号空间形成了矢量空间。此外，由于我们可以定义信号之间的内积，因此信号形成的空间为 Hilbert 空间。

良好定义通常要求有界性以及可重复性。一个矢量空间必须满足的条件是该空间对于矢量加法是封闭的，并且该空间为一个线性空间。我们也将不断地重复，线性空间实际上是我们研究绝大多数问题时优先考虑的空间，本段译者注。



图 3.1 如果待求函数假定为可以逼近为 $f(\mathbf{x}) \approx g(\mathbf{A}\mathbf{x})$, $\mathbf{A} \in \mathbb{R}^{k \times d}$, s.t. $k \ll d$, 则浅层神经网络即可捕获图中归纳偏置, 见[Bach, 2017]。在典型的应用中, 这种对于低纬度空间投影的依赖是不现实的, 正如图中展示的, 一个低通滤波器可以将图中卷带投影至低纬度子空间, 虽然投影后多数信息被投影到了子空间, 但相当的信息丢失了。

一个度量空间上若柯西序列为收敛的，则该空间为 Banach 空间，实际上并非所有的度量空间均是巴那赫空间，例如空间 $(0, 1]$ ，并使用度量为欧氏度量，则该空间并非巴那赫空间，这是因为考虑 $1/n$ 的柯西序列时，序列收敛点为 0，该点不在空间里，因而该空间不为巴那赫空间。当一个空间是巴那赫空间后，若施加内积定义，则该空间就成为了希尔伯

特空间，也即内积空间。本段译者注。

这里以域 $\Omega = \mathbb{Z}_n \times \mathbb{Z}_n$ 为典型例子来说明，这一域即为二维的一个 $n \times n$ 网格（如灰度图片、深度图片等所在域，例如 256×256 大小，译者注）， x 表示一个RGB图片，也即：

$$x: \Omega \rightarrow \mathbb{R}^3 \quad (3-1)$$

f 表示一个函数，例如单层感知机并作用于维数为 $3n^2$ 的输入上。我们将在接下来进行详细地介绍，在细致介绍前值得说明的是，域 Ω 通常附加着特定的几何结构与对称性。尺度分离使得我们可以在将数据进行池化处理（粗糙化处理）时能够保留重要的特质（例如对图片进行降采样）。

我们将介绍这两条重要的原则（并将其称为几何先验）在现代深度学习网络架构中的绝对地位。对于上述所说的图片的示例，几何先验通过共享卷积核权重的方式（利用了平移不变性）以及池化方法（利用尺度分离）被构建在了卷积神经网络中（CNN）。扩展这些构建几何先验的想法到其他域中，例如图、流形中，并展示几何先验如何从这两条根本的原则中萌生，这些正是几何深度学习的主要目标，也是我们写就本书的主旋律（leitmotif of our text）。

3.1 对称性、表示、不变性 (Symmetry, Representations and Invariance)

非正式地，一个对象的对称性是指在某种变换的作用下该对象或系统保持某些关注的性质不发生变化或者恒定的现象。这些变换本身可以是光滑的、连续的，也可以是离散的。对称性对于众多机器学习任务而言是一种普遍的现象。例如，在计算机视觉中，物体的分类不会随着平移而发生改变，因而平移变换就是图像分类处理任务中的对称性。再如，在计算化学中，预测化学分子的性质与化学分子在空间中的旋转变换无关。在讨论粒子系统时，离散对称性自然出现了：由于粒子没有标准的次序，因而其在排列变换下保持不变。在其他动态系统中系统具有时逆对称性，例如保持平衡态的系统抑或是牛顿第二定律下的运动。我们将在4.1节中看到，排列对称性也是分析基于图的数据的核心想法。

3.1.1 对称群 (Symmetry Groups)

研究对象的对称性变换的集合满足一系列的性质。首先，对称变换可能组合起来形成新的对称变换，例如 g, h 表示两个群对称变换（即群中两个元素，译者注），则两者的组合作用 $g \circ h$ 及 $h \circ g$ 均为对称群变换（事实上这也是群定义中要求的四个性质之一，译者注）。原因在于，既然单个群作用作用在对象时是对称的，那么他们的组合也应当是对称的。进一步来说，对称性总是可逆的变换，因而对称群变换的逆仍然是对称的。这些性质表明对称变换形成了一个代数结构---群（对群的理论有兴趣的读者，推荐参看[Donald L. Kreher, 2020]，以及YouTube视频<https://www.youtube.com/watch?v=UwTQdOop->

[nU&list=PLwV-9DG53NDxU337smpTwm6sef4x-SCLv](#), 译者注)。既然这些对称元素作为集合深度学习网络的数学基础组件, 他们值得给出正式的定义以及细致的讨论。

群是一个高度抽象的概念, 对应的具象化就是群作用。群中元素具象化为群作用后, 即代表了某种变换作用, 例如旋转变换、平移变换等作用。正式地讲, 群 \mathfrak{G} 是一个集合, 该集合支持一种双元运算符 \circ , 并满足四条性质: (1) 任意的群中必有单位群元素, 即必存在 e : $e \circ g = g \circ e = g$; (2) 群运算封闭: $g \circ h \in \mathfrak{G}$; (3) 满足结合律; (4) 任意群元素均存在逆, 即任意元素 g , 均有 h 使得 $g \circ h = e$ 。本段译者注。

注意群运算中并没有规定交换律, 也就是说, 一般情况下 $h \circ g \neq g \circ h$, 对于满足交换律的群通常称为阿贝尔群 (Abelian Group), 这一命名源自挪威数学家 Niels Henrik Abel (1802-1829)。

尽管一些群非常大甚至包含无穷多元素, 这些元素通常仅仅来自于少数几个群袁术的组合, 这些组合形成群中所有元素的元素被称为群的生成器元素。正式地, 若群 \mathfrak{G} 中所有元素 $g \in \mathfrak{G}$ 均可以表示成一个其子集 $S \subseteq \mathfrak{G}$ (即为生成器) 的有限个元素的有限组合, 则我们说群 \mathfrak{G} 由生成器 S 生成。例如, 等边三角形的对称群 (也被成为 Dihedral Group D_3 , D 表示德语中 “Drei” 意为 3, 译者注) 是由 60° 旋转及镜生成的 (图 3.2)。一维的平移群是由无穷小位移变换生成, 这也是一个可微分李群的典型案例, 我们将在以下详细讨论。

李群是一种特殊的群, 其特殊在于其既是群, 也是流形, 换言之, 其代表了光滑的群, 而非离散表示的群。译者注。

可能有读者已经注意到, 我们将群定义为了一个抽象的对象, 并没有说群中元素究竟是什么 (例如一些域上的变换), 而是仅仅介绍了群元素怎么组合。因此一些大不相同的对象可能具有相同的对称群。例如前述的三角形的旋转变换以及镜像变换群与三个元素的排列群完全一样 (我们可以排列三角形中的顶点, 正如旋转及镜像能够达到的效果一样, 如图 3.2)。

3.1.2 群作用与群表示 (Group Action and Group Representations)

相对于将群看成是一个抽象的实体对象, 我们通常关心群怎么作用在数据上 (即群元素作用数据上等于什么样的变换, 译者注)。由于我们已经假定了数据蕴含于某个域, 因此我们主要关心的是群将怎么作用在该域中 (例如平面中点的平移变换), 并且由此得到相同群对于信号空间 $\mathcal{X}(\Omega)$ 的作用 (例如平面相片以及特征映射的平移变换)。

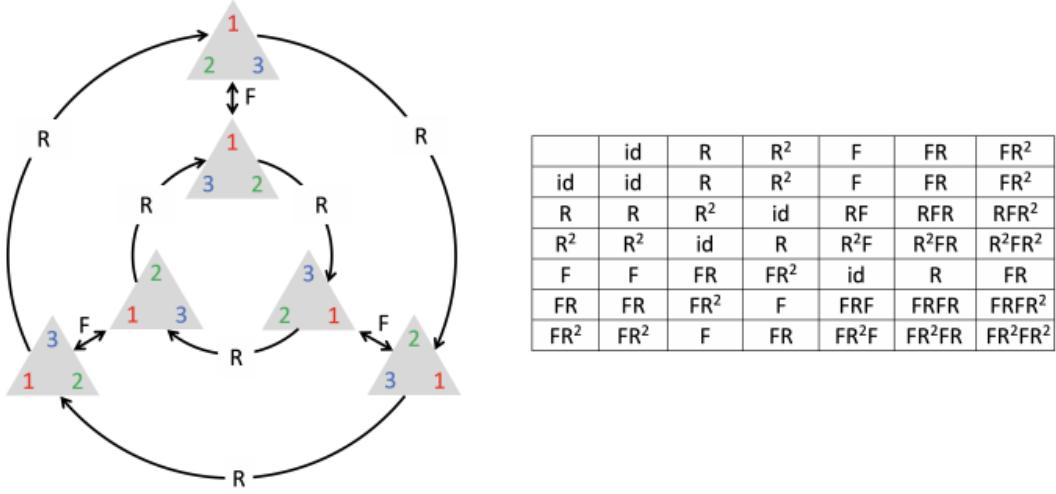


图 3.2 凯莉图 (Cayley Diagram)。一个等边三角形的对称变换可以由三种基本变换组合而来：旋转 60° 、旋转 120° 、镜像，图中使用 1、2、3 以及符号 (R, R^2, F) 分别表示这三种变换，这一变换群与包含三个元素的排列群完全一样。

一个群 \mathfrak{G} 对于一个集合（即域） Ω 的群作用定义为映射 $(g, u) \mapsto g.u$ ，包括一个群元素 $g \in \mathfrak{G}$ 以及一个点 $u \in \Omega$ ，以及该点与域内其他点作用效果等同且兼容，群元素组合作用兼容等，即： $g.(h.u) = (g \circ h).u, \forall g, h \in \mathfrak{G}, \forall u \in \Omega$ 。我们将在随后的章节中看到众多群作用的示例。例如在平面中，欧式变换群 $E(2)$ 就是一个能够保持变换前后线的长度不变的变换群，群中元素包含旋转、平移及镜像。这一欧式变换群也可以作用于二维图像中，旋转、平移、反转像素点等，此外这一群也可以作用于神经网络的表示空间。更确切地说，若我们有一个作用于域 Ω 的群 \mathfrak{G} ，我们可以自动地得到群 \mathfrak{G} 作用在信号空间 $\mathcal{X}(\Omega)$ 上的表达：

$$(g.x)(u) = x(g^{-1}.u) \quad (3-2)$$

由于群元素 g 的逆，这的确是一个正确的群作用，并且我们可以得到：

$$(g.(h.x))(u) = ((g \circ h).x)(u) \quad (3-3)$$

最重要的一类群作用，也是我们将在本书中不停地遇到的群作用，就是线性群作用，也被称为群表示。式 (3.2) 中的群元素作用于信号上也是线性的作用，由于：

$$g.(\alpha x + \beta x') = \alpha(g.x) + \beta(g.x') \quad (3-4)$$

其中 $\alpha, \beta \in \mathbb{R}$ ，信号 $x, x' \in \mathcal{X}(\Omega)$ 。我们既可以用对信号 x 线性的映射 $(g, x) \mapsto g.x$ 来描述线性群作用，也可以等效地，使用映射 $\rho: \mathfrak{G} \rightarrow \mathbb{R}^{n \times n}$ 来描述，其中映射 ρ 把群中的每个元素 g 作为输入，输出 $\rho(g)$ 表示的矩阵。矩阵的维度通常可以是任意的，并且与群 \mathfrak{G} 本身的维度或者域 Ω 本身的维度不相关，不过通常情况下机器学习算法采用群作用的特征空间的维度最为输出的矩阵维度。例如，我们有维度为 2 的平移群作用于 n 个像素点的图像空间。如果域 Ω 本身是无穷维的，那么信号空间 $\mathcal{X}(\Omega)$ 也将是无穷维的，在这种情况下 $\rho(g)$ 表示该域空间的线性算子，而不是一个有限维的矩阵，不过在现实中，人们通常需要

离散化表示一个空间，也就是维度并非无穷。

对于一般的群作用而言，将群元素赋予一个矩阵表示时应当注意与群作用兼容。更确切地说，群元素的组合作用的矩阵表示应当与单个群元素作用的矩阵表示的乘积（对于阿贝尔群为加）相等，以满足群运算法则。

群是一个抽象的表示方式，而在具体地进行作用时，需要将抽象的表示换成具体的作用函数，这就是群表示研究内容，由于众多群的元素实际上有无穷多个，因此表示这样的作用时无法通过枚举的方式来实现，对此研究人员仔细分析了群元素的特点，并指出了群元素的若干生成器来生成的，因而可以用生成器来代表整个群，并以此研究群作用的表示。本段译者注。

当写成群表示的形式时，群 \mathfrak{G} 中元素 \mathfrak{g} 对于信号 $x \in \mathcal{X}(\Omega)$ 的作用可以写为式(3.5)，并且我们可以验证式(3.6)成立。

$$\rho(\mathfrak{g})x(u) = x(g^{-1}u) \quad (3.5)$$

$$(\rho(\mathfrak{g}))(\rho(\mathfrak{h})x)(u) = (\rho(\mathfrak{g} \circ \mathfrak{h})x)(u) \quad (3.6)$$

3.1.3 不变与等变函数 (Invariant and Equivariant Functions)

域 Ω 在信号 $x \in \mathcal{X}(\Omega)$ 作用下的对称性使得作用在这些信号上的函数 f 具有了结构。实际上这是一种非常强的归纳偏置，通过减小可能的插值函数的空间 $\mathcal{F}(\mathcal{X}(\Omega))$ 的大小来有效地提高了学习的效率，也就是限定了插值函数必须要满足对称性先验。在本文中我们将探讨的两个重要的先验示例就是不变与等变函数。

【介绍不变与等变函数概念】

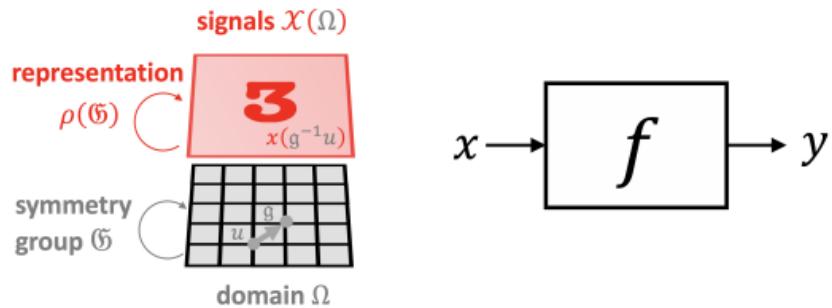


图 3.3 几何深度学习关注的三个方面：域、信号、以及假设空间

不变性的一个经典的示例的平移不变性，源自于计算机视觉领域以及诸如图像分类这样的模式识别应用中。值得注意的是众多信号处理书籍中的不变性实际上是等变形，我们将在下述位置正式介绍两者及其数学表达，以作为区分。这样的平移不变性对称变换中，函数 f （常常被部署于卷积神经网络 CNN 中）接受图片作为输入信息，并输出一个代表图片中包含某物体的概率（例如猫、狗等物体）。通常我们假定图片包含某种物体的概率与物体在图片中的位置无关，换言之，函数 f 应具有移动不变性（shift-invariant）。可以任意精

度逼近任意函数的网络多层感知机没有这样的移动不变性，这也是为什么 19 世纪 70 年代众多利用多层感知机架构进行模式识别的方法最后失败了。神经网络架构的发展，尤其是局部共享权重的应用，虽然有众多原因促使其出现并发展，实现移动不变性的模式识别在过去是一个重要的原因，这一典型案例是卷积神经网络（CNN）。

当我们进一步审视 CNN 的卷积层，我们就会发现其并不是移动不变的，而是移动等变的，换句话说，一个在卷积层输入端的移动变换将导致同样的输出特征映射的移动变换。

不变性即要求群作用与否不影响映射的结果，例如求和映射对于排序群作用可以保持不变性；而等变性指群元素作用于输入等价于群元素直接作用于输出。本段译者注。

回到计算机视觉，一个要求移动等变性的原型性应用场景为图像分割任务，其输出为一个逐像素点的图像掩码函数 f 。显然，分割掩码必须随着物体在图像中的移动而移动。在这一例子中，输入输出的域是一样的，但是输入是三个通道（RGB 三通道），但输出仅有一个通道，因而其表示 $(\rho, \mathcal{X}(\Omega, \mathcal{C}))$ 与 $(\rho', \mathcal{X}(\Omega, \mathcal{C}'))$ 是不一样的。

然而即使前述的图像分类的任务也通常可使用卷积层作为部署方法，并应用全局池化（池化操作是移动不变的）。我们将在 3.5 节中看到，这正是大多数深度学习框架的一般蓝本，包含 CNN、GNN 等。

3.2 同构与自同构（Isomorphism and Automorphism）

3.2.1 子群与结构层次（Subgroups and Level of Structure）

正如前边所言，对称性即指在某种变换的作用下保持某些性质或结构不变的性质，对于一个给定结构，这样的变换的集合形成了一个对称群。通常的情况是我们不仅关注结构，而是关注多个结构，因而我们常考虑域上的多个有层次的结构。因而，什么变换能够称为对称性，取决于我们考虑的结构是什么，不过对于所有的结构而言，对称性变换总是可逆的。一个对不同研究对象的可逆映射并且能够保留结构的变换通常称为同构变换，或简称同构（Isomorphism，希腊语中“同样外形”之意，译者注），一个对研究对象自身的同构变换构成的映射群称为自同构（Automorphism）或是对称。

从最基本的结构考虑，假定域 Ω 是一个集合，因此域有着最少的结构（连集合中元素的连接关系换言之拓扑关系，都没有，译者注）：我们仅仅能够说一个集合具有一些特定的度（Cardinality，对于有限元素的集合而言指集合中相同元素的个数，对于包含无穷多元素的集合而言指集合中元素的种类，例如实数集合可以看成仅有有理数与无理数两类，译者注）。一个能够保存该度结构的自身对自身的映射是双射的（Bijective）且可逆的，这一特点也可以视为是集合层次的对称性（如集合在排列（Permutation）变换下的双射性与可逆性）。读者也可以轻易地验证这一双射形成了一个群：（1）幺元，指集合自身不做任何变

换；（2）封闭性，由双射的组合仍然是双射来验证；（3）逆，双射总存在逆映射；（4）结合律，生设的组合可以形成新的双射。

取决于具体的应用，不同的域可以考虑进一步的结构，例如对于一个拓扑空间域 Ω 来说，我们可以考虑能够维持连续性及连续性的逆的映射：这样的映射也被称为同胚（Homeomorphism）加上不同的集合间的双射映射群。直观来讲，连续函数有一系列“好”的性质并能够将原空间相邻点 u 映射到新空间的相邻位置 $\tau(u)$ 。

人们还可以进一步提出要求：要求上述微分同胚映射的逆是可微分的，也就是说，映射及其逆映射在每一点都有微分，且微分是连续的。如果映射及其逆无穷次可微分，则称其为光滑的，或者是 C^∞ 的，如果其仅有限次可微，例如 r 次可微分，则称其为 C^r 的。（实际上同胚与无穷次可微分有所区别，一个知名的例子是 John Minor 提出的 Minor 怪球，Exterior Sphere，译者注）。可微分的映射引出了可微分流形（Manifolds）的概念，这样的映射也被称为微分同胚映射，简称微分同胚，记为 $\text{Diff}(n)$ ， n 为维度。另一个我们将会遇到的结构是距离度量、度量（metric）（能够维持这一结构不变的变换称为 Isometries）或者旋转变换（orientation）（据我们所知，这一变换没有一个希腊语对应，著者语）。

需要考虑的正确的层次结构取决于具体的问题。例如，在分割组织学病理图像时，我们可能希望原图的反转版本与原图等变（由于将其放置在显微镜上是看到的是翻转的图像），但是若我们想要分类路标时，我们仅希望考虑能够保留朝向信息的变换最为对称性（这是由于朝向的不同代表了不同的标志含义，例如左拐、右拐基本是镜像对称的，但其朝向不同，译者注）。

当我们设置更多待保留的结构时，变换群的规模将越来越小（因而可以做的变换越来越少，译者注）。的确，添加结构意味着从原群中挑选子群，子群正式原有的更大群的子集且满足群的定义。

若一个群中所有元素均存在于另一个群中，则称该群为子群。本段译者注。

例如，欧式同构变换群 $E(2)$ 就是一个平面微分同胚群的子群，朝向维持的群 $SE(2)$ 是欧式同构变换群 $E(2)$ 的子群。这一层次性的结构由 Klein 在其 Erlangen Programme 的项目书中构建，投影变换群、仿射变换群、以及欧式同构变换群有着越加多的不变性结构，因而其群规模也越来越小。

3.2.2 同构与自同构（Isomorphisms and Automorphisms）

前述中我们将对称性当成是一个对象对自身对象的能够维持结构不变以及可逆的映射。这样的映射也被称为自同构（Automorphism），这一变换群描述了一个对象怎么等价地与自身进行映射。然而另一类非常重要的映射是同构映射（Isomorphism），其描述了不同对象间的映射的对称性。这两个概念经常混淆，不过理清楚这两个概念的区别与联系将为我们后续的讨论提供帮助。

为理解两者不同，考虑一个集合 $\Omega = \{0, 1, 2\}$ ，一个自同构变换是映射 $\tau: \Omega \rightarrow \Omega$ ，例如一个循环右移操作 $\tau(u) = (u + 1) \bmod 3$ 。这样的一个映射保留了集合的度结构信息，将域映射到自身。若我们考虑另一个域 $\Omega' = \{a, b, c\}$ ，与域 Ω 有着相同的元素个数，则双向映射 $\eta: \Omega \rightarrow \Omega'$ ，例如具体地令 $\eta(0) = a, \eta(1) = b, \eta(2) = c$ ，这样的映射变换构成的群即为同构变换群。

研究对象的结构这一想法不仅仅包含一系列的节点，也包含节点之间的连接性（connectivity），我们将在 4.1 小节中详细介绍这一说法。一个图 $\mathcal{G}(\mathcal{V}, \mathcal{E})$ ，以及另一个图 $\mathcal{G}(\mathcal{V}', \mathcal{E}')$ 之间的同构映射 $\eta: \mathcal{V} \rightarrow \mathcal{V}'$ 是一个双射，这点不仅对于有连接关系节点之间，对于没有连接关系的节点也是如此。因此这两个同构图有着相同的结构，他们仅仅在节点的次序分布方面有所不同。另一方面，一个图的自同构，或者说一个对称映射 $\tau: \mathcal{V} \rightarrow \mathcal{V}$ ，将图中节点映射为图中的另一个节点，这一映射保留了连接性。一个非平凡的自同构映射能够保留对称性（非平凡这里指变换后与变换前不完全一样，从群作用角度看，即映射不等于群幺元，译者注）。

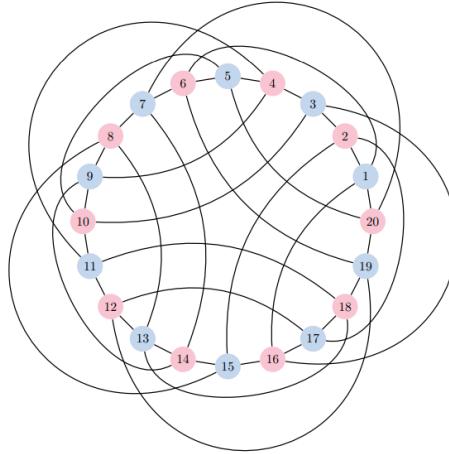


图 3.4 Folkman Graph [Folkman, 1967]，该图是具有 3840 个 Automorphism 的典型例子

3.3 变形稳定性 (Deformation Stability)

上述 3.1-3.2 小节中介绍的对称性来源于我们所知道的理想世界中的各种变换对称性，我们也希望能够精确地保有这些对称性。例如在计算机视觉研究中，我们通常假定平移对称性，然而现实世界中数据是充满噪声的，由此模型可能在两方面难以达到我们的标准。

首先，尽管这些简单的群为理解我们所研究的域 Ω 以及作用的信号空间 $\mathcal{X}(\Omega)$ 的全局对称性提供了一种方法，但是这些简单群并不能提供局部的对称性信息。例如，考虑一个包含有几个物体的视频流，视频中每个物体向着各自不同的方向运动，则在接下来的帧中，视频中包含着集合一样的语义信息，但是没有一个全局的变换能够提供从运动前到运动后的变换作用。在一些其他的例子中，例如被相机捕获到的一个变形的三维模型，我们也十

分难以描述作用在变形的模型上的群作用（这是因为变形可能仅发生在局部，而非全局变形，译者注）。这些例子说明了，在实际中我们更加关心一个全局变换被局部的、模糊的变换取代了的一系列对称变换。在我们的探讨中，我们将取分两类场景：（1）域 Ω 是固定不变的，作用其上的信号空间 $\mathcal{X}(\Omega)$ 可能发生变化；（2）域 Ω 本身也是可能发生变形的。

3.3.1 信号变形的稳定性 (Stability to Signal Deformations)

在多数应用中，我们知道一个先验信息：一个信号 x 的微小变化不应当引起输出 $f(x)$ 的变化，因而我们可以尝试将这种信号的微小变化看作对称性群作用。例如，我们可以将微分同胚群作用 $\tau \in \text{Diff}(\Omega)$ ，甚至更小的双射作用，看成是对称性。然而，小的变化可以组合起来形成大的变化，因而小的变形不能形成一个群，我们也不能要求小变形（即信号的微小变化，同一意思，译者注）的不变性或等变形。由于大的变形实际上可以改变输入信息的语义内容，因而使用全部的微分同胚群作为对称群也是不恰当的主意。

一个更好的想法是使用复杂度度量函数 $c(\tau)$ 量化一个给定映射 $\tau \in \text{Diff}(\Omega)$ 距离形成一个对称子群 $\mathfrak{G} \subset \text{Diff}(\Omega)$ 的差距有多大，并满足当 $\tau \in \mathfrak{G}$ 时 $c(\tau) = 0$ 。我们现在就可以将我们之前定义的在群作用下严格的不变性与等变形替换为新的形式，借助于一个更加“软”的变形稳定性（或称逼近性的不变性），应用如下公式：

$$\|f(\rho(\tau)x) - f(x)\| \leq Cc(\tau) \|x\|, \forall x \in \mathcal{X}(\Omega) \quad (3-7)$$

其中 $\rho(\tau)x(u) = x(\tau^{-1}u)$ ，这与前述一致， C 是依赖于信号的某常量，满足式(3.7)的函数 $f \in \mathcal{F}(\mathcal{X}(\Omega))$ 称为几何稳定的。我们将在3.4节中看到一些示例。

由于当 $\tau \in \mathfrak{G}$ 时有 $c(\tau) = 0$ ，式(3.7)中的定义泛化了群不变性的定义。应用时需要根据应用情况选择合适的变形惩罚项，即式中 $c(\tau)$ 。例如定义在欧氏空间上的平面图片，一个流形的选择是：

$$c^2(\tau) := \int_{\Omega} \|\nabla \tau(u)\|^2 du \quad (3-8)$$

这一公式度量了映射 τ 的“弹性”（也可以视为平稳性，例如在其梯度全为0，则该积分惩罚项为0，译者注），也即衡量了一个常量矢量场作用下的平移前后有多大区别。这一变形惩罚项实际上是一个二范数，常被称为 Dirichlet 能量（实际上狄利赫雷能量在未被严格证明前已经被 Green 等知名数学家广泛使用，后来有数学家 Warssmann 指出其没有严格的证明，导致了随后一百年间这一公式没人敢用，这一证明最后由 Hilbert 给出，译者注），其可以用来衡量映射 τ 与平一群差距有多远。

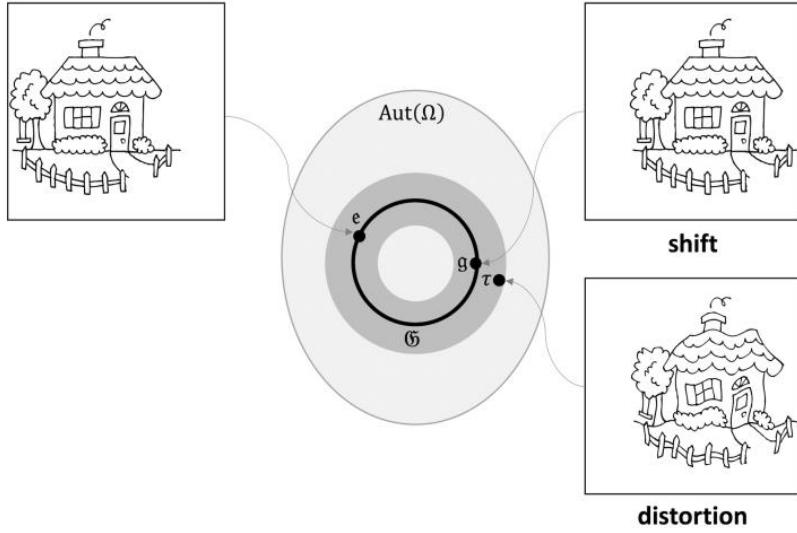


图 3.5 一个域映射到自身的作用形成的双射空间称为 **Automorphism**, 即 $\text{Aut}(\Omega)$, 其中群 \mathfrak{G} 是一个子群。几何稳定性扩展了群不变性以及群等变形, 并以某种度量来进行量化, 本例中, 光滑的畸变作用与移动作用有类似的效果

3.3.2 域变形的稳定性 (Stability to Domain Deformations)

在众多的应用中, 变形的对象并非是信号, 而是几何域本身。这样情况的一个标准的例子是处理图以及流形的应用: 一个图可以对不同时刻的社交网络进行建模, 不同时刻的社交网络会有稍微不同的社交间关系; 流形可以对一个非刚性变形进行建模。着这些变形可以按照如下的方式进行量化。若 \mathcal{D} 表示所有可能域变化形成的空间 (例如所有可能形成的图网络, 或者全部黎曼流形构成的空间), 我们可以针对两个域 Ω 及 Ω' 定义合适的度量 $d(\Omega, \Omega')$ 并满足条件: 若在某种结构下 Ω 域与 Ω' 域是一样的, 则 $d(\Omega, \Omega') = 0$ 。例如当两个图是同构时, 图之间的编辑距离等于 0; 当两个流形是同构时, 使用测地线度量的黎曼流形间的 Gromov-Hausdorff 距离等于 0。

一个这样的距离函数构建方法是使用一个试图将两个域对应起来并变成一致的可逆映射 $\eta: \Omega \rightarrow \Omega'$, 这一映射同时还试图保持关联的结构。例如, 在图的例子或者黎曼流形 (使用测地线度量作为度量空间) 的例子中, 这种试图对应的操作可以用逐点邻域或者距离结构来表示 (分别用 d 及 \tilde{d} 表示):

$$d_{\mathcal{D}}(\Omega, \Omega') = \inf_{\eta \in \mathfrak{G}} \|d - \tilde{d} \circ (\eta \times \eta)\| \quad (3-9)$$

其中 \mathfrak{G} 为表示双射或者同构映射的群, 上述的范数就是定义在了 $\Omega \times \Omega$ 上。换句话说, 两个域 Ω 及 Ω' 之间的元素距离被提高了一个层次, 到了两个域之间的距离, 这是通过考虑为保留内部结构的所有可能的对齐。给定一个信号 $x \in \mathcal{X}(\Omega)$ 以及一个变化的域 $\tilde{\Omega}$, 我们可以考虑变形后的信号为:

$$\tilde{x} = x \circ \eta^{-1} \in \mathcal{X}(\tilde{\Omega}) \quad (3-10)$$

稍微滥用一下记号，我们定义在给定变化域上的所有可能输入信号的组合为

$\mathcal{X}(\mathcal{D}) = \{(\mathcal{X}(\Omega), \Omega) : \Omega \in \mathcal{D}\}$ ，称一个函数映射 $f: \mathcal{X}(\mathcal{D}) \rightarrow \mathcal{Y}$ 对与变形稳定，当且仅当：

$$\|f(x, \Omega) - f(\tilde{x}, \tilde{\Omega})\| \leq C \|x\| d_{\mathcal{D}}(\Omega, \tilde{\Omega}) \quad (3-11)$$

对所有的 $\Omega, \tilde{\Omega} \in \mathcal{D}$ 以及 $x \in \mathcal{X}(\Omega)$ 均成立。考虑到流形中的同构映射变形发挥着重要的作用，我们将在 4.4-4.6 节中用流形为例详细介绍这一想法。此外，通过从小体素形式 [Gama et al., 2019] 的变形作用角度看信号作用下的变形稳定性，我们可以说域变形的稳定性是对信号变形稳定性分析的泛化。

3.4 尺度分离 (Scale Separation)

尽管变形稳定行极大地强化了全局对称的先验信息，它本身的应用并不能克服维数灾难，从某种角度说，尤其不正式地说，随着维数的增加存在太多可供选择的函数了。一个克服维数灾难的关键性的直觉是探索利用物理任务的多层次结构。在进一步介绍多层次表示之前，我们需要一些傅里叶变换的一些主要元素，这些元素依赖于频率而非尺度。

3.4.1 傅里叶变换与全局不变性 (Fourier Transform and Global Invariants)

可以说最知名的信号分解方法就是傅里叶变换，这也是调和分析的基石。经典的一维傅里叶变换为：

$$\hat{x}(\xi) = \int_{-\infty}^{+\infty} x(u) e^{-i\xi u} du \quad (3-12)$$

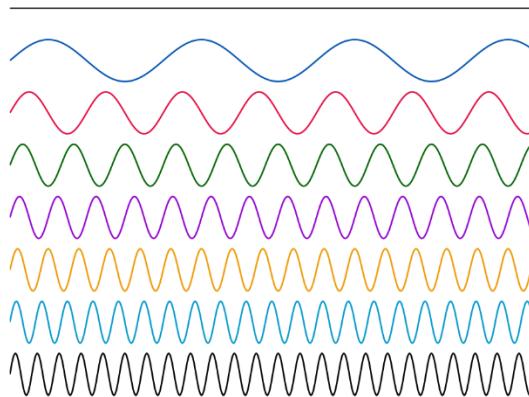


图 3.6 傅里叶变换的基底是全局支撑的，因此局部的信号也将在全局所有频段产生能量

其中函数 $x(u)$ 为可积分函数 $x(u) \in L^2(\Omega)$ 并且作为正交震荡的基函数 $\varphi_\xi(u) = e^{i\xi u}$ 的线性组合作用在域 $\Omega = \mathbb{R}$ 上，基函数由其震荡频率 ξ 作为索引。这一将原有时域信号转化到频域的方法揭示了信号本身的重要信息，例如信号的平滑性及局部性。傅里叶基函数自身也有深入几何基础，可以降级函数解释为域的自然震荡，这与其几何结构有紧密关联

[Berger, 2012]。

傅里叶变换在信号处理中发挥着基础性作用，它提供了卷积的对偶形式作为一个线性信号滤波的标准模型（我们使用 x 表示信号，使用 θ 表示滤波器）：

$$(x \star \theta)(u) = \int_{-\infty}^{+\infty} x(v) \theta(u - v) dv \quad (3-13)$$

值得注意的是，在后续的章节中，我们将使用协变形式的卷积表示（式 (3.14)），并将协变形式的卷积域式 (3.13) 表示混着使用，由于在机器学习中这两者的差别主要是由滤波器决定的，而滤波器通常是待学习的参数，因此两者的不同仅仅是记法的不同。

$$(x \star \theta)(u) = \int_{-\infty}^{+\infty} x(v) \theta(u + v) dv \quad (3-14)$$

我们将看到，卷积算子在傅里叶变换基函数中被对角化了，使得可以将卷积运算表示成对应的傅里叶变换后的乘积，也即：

$$\widehat{(x \star \theta)}(\xi) = \hat{x}(\xi) \cdot \hat{\theta}(\xi) \quad (3-15)$$

这一公式也被称为卷积定理。

事实上，很多基本的微分算子，例如 Laplace 算子，也可以描述为欧氏空间上的卷积变换。正是微分算子也可以内在地定义到非常一般的几何上，故而这些微分算子扩展了欧氏空间的傅里叶变换并提供了一个规范化的处理流程，例如对图、群、流形的处理。我们将在 4.4 节详细探讨。

傅里叶变换本质的一面是它能够揭示信号以及域的全局特性，例如平滑性、电导性。在域的全局对称性（例如平移变换）存在的情况下，这样的全局表现对于很多任务而言是很方便的，但这样的全局表现对于微分同胚的研究没什么助力。这就要求我们找到一个能够折中考虑空间及频率域定位的表示，我们将马上讨论这一点。

3.4.2 多尺度表示 (Multiscale Representations)

局部的不变性的想法可被描述为将傅里叶频域基础上的表示切换为多尺度层次化的表示，这也是多尺度分解方法的基石，例如可以采用小波基函数 (wavelets)。多尺度表示方法的核心是将定义在域上的函数分解为基本的单元函数，这些单元函数是位于空间及频域的。以小波基函数为例，通过关联一个平移以及扩张滤波器 (mother wavelets) ψ ，生成一个空域与频域复合的表示，这种表示也被称为连续小波变换 (continuous wavelet transformation)：

$$(W_\psi x)(u, \xi) = \xi^{-1/2} \int_{-\infty}^{+\infty} \psi\left(\frac{v-u}{\xi}\right) x(v) dv \quad (3-16)$$

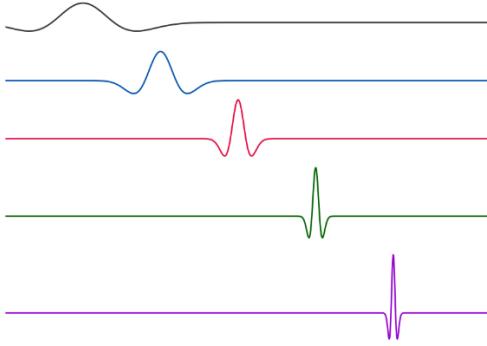


图 3.7 Wavelet 变换基函数[Mallat, 1999]

移动及扩张滤波器也被称为小波原子，他们的空间位置与扩张跟小波变换的两个坐标 u 和 ξ 关联。通常以特定的方式进行采样来获得坐标：若 j 表示尺度，则坐标采样为 $\xi = 2^{-j}$ 及 $u = 2^{-j}k$ 。多尺度的信号表示在全局平滑性之外带来了捕捉正则特质的益处，例如逐元素的光滑性，这也使得多尺度信号表示方法在 20 世纪 90 年代时在信号及图像处理、数值分析中愈加流行。

3.4.3 多尺度表示的变形稳定性 (Deformation Stability of Multiscale Representations)

相对于傅里叶分解，多尺度局部性小波分解的优势在于考虑了内在的对称群作用下的小的邻域变形的作用。首先我们在欧氏空间及平移变换群中展示这一重要的概念。由于傅里叶变换对角化了移动算子（可视为是一种卷积，我们将在 4.2 节中详细叙述），因而傅里叶分解的表示对于平移变换群作用是一种高效表示。但是傅里叶变换在高频变形的不稳定的，与之相反，小波分解在高频段则稳定的多。

实际上，我们可以考虑 $\tau \in \text{Aut}(\Omega)$ 及其关联的线性表示 $\rho(\tau)$ 。当 $\tau(u) = u - v$ ，代表其是一个移动算子，我们将在 4.2 节中验证，这一移动算子的表示 $\rho(\tau) = S_v$ 与卷积算子可交换。由于卷积算子经由傅里叶变换对角化了，频域中移动算子等效于傅里叶变换中的一个复项：

$$\widehat{(S_v x)}(\xi) = e^{-i\xi v} \hat{x}(\xi) \quad (3-17)$$

因而去掉了复数项的傅里叶的模 $f(x) = |\hat{x}|$ 就是一个移动不变性的函数，即有：

$$f(S_v x) = f(x) \quad (3-18)$$

但是如果我们仅有趋近的移动，即有 $\tau(u) = u - \tilde{\tau}(u)$ ， $\|\nabla \tau\|_\infty = \sup_{u \in \Omega} \|\nabla \tilde{\tau}(u)\| \leq \epsilon$ 情况就完全不一样的。我们可以证明：

$$\frac{\|f(\rho(\tau)x) - f(x)\|}{\|x\|} = \mathcal{O}(1) \quad (3-19)$$

这与 ϵ 多么小都无关（也就是说， τ 表示的移动有多大）。因而所有的傅里叶表示均是

在变形下不稳定的，无论变形有多么小。这种失稳定性不仅仅发生在非刚性移动的域中，而是会发生在一般的域上，我们也将再 4.4 节中使用一种自然扩展的傅里叶变换进行 3D 分析时展示这种不稳定性作为示例。

小波基函数为这一不稳定的表示问题（不稳定的表示问题也揭示了多尺度表示的重要作用）提供了一种改良方法。在上述的实例中，我们已经[Mallat, 2012]展示了小波分解 $W_\psi x$ 对变形是等变的，下式记法暗示了 $\rho(\tau)$ 是作用在空间的坐标 $(W_\psi x)(u, \xi)$ 上：

$$\frac{\|\rho(\tau)(W_\psi x) - W_\psi(\rho(\tau)x)\|}{\|x\|} = \mathcal{O}(\epsilon) \quad (3-20)$$

换句话说，将信号使用局部滤波分解在不同尺度中而不是频域中能够将全局不稳定的表示替换为一类局部稳定的特征。重要的是，这种在不同的尺度进行度量的表示并非是不变性的，并且需要由高频到低频进行处理，这暗示了现代神经网络的深度组合的本质。我们将在下面介绍了几何深度学习正是捕捉到了这一点，这将在接下来进行探讨。

3.4.4 分离表示的先验 (Scale Separation Prior)

我们可以通过考虑将多尺度粗糙化域 Ω 的数据计算为层次化的新的域 $\Omega_1, \Omega_2, \dots, \Omega_J$ 实现这一洞察。实际上，这种粗糙化处理可以定义到一般的域上，包括网格域、图域、流形等。非正式地说，粗糙化过程是综合平均化域 Ω 中的相邻点 $u, u' \in \Omega$ ，因而进而要求一种合适的域的度量的定义。若 $\mathcal{X}_j(\Omega_j, \mathcal{C}_j) := \{x_j : \Omega_j \rightarrow \mathcal{C}_j\}$ 表示定义在粗糙化后的域 Ω_j 上的信号，我们可以非正式地说一个方程 $f : \mathcal{X}(\Omega) \rightarrow \mathcal{Y}$ 在尺度 j 上是局部稳定的，若满足函数 f 能够被因子化表示为 $f \approx f_j \circ P_j$ ，其中 P_j 表示 $P_j : \mathcal{X}(\Omega) \rightarrow \mathcal{X}_j(\Omega_j)$ 是一个非线性的粗糙化粒子，函数 $f_j : \mathcal{X}_j(\Omega_j) \rightarrow \mathcal{Y}$ 。换句话说，尽管目标方程 f 可能依赖于域上复杂的远程耦合特征，在局部稳定的方程上，根据不同尺度对这种耦合作用进行隔离仍然是可能的，可通过首先关注那些局部的能够在其后传递到粗糙尺度上的耦合作用实现。

这些原则在诸多领域中（例如物理、数学）尤为重要，例如在统计物理中应用的重正规范化群（Renormalisation Group）、在数值分析中使用的 FMM 方法（Fast Multipole Method，一种数值方法，最初用于在多体动力学中加速长程力的计算，FMM 将靠近的源集合在一块并视为一个源）。在机器学习中，多尺度表示（Multiscale Representation）与局部不变性（Local Invariance）是保证卷积神经网络（CNN）、图神经网络（GNN）的高效性的基础性数学原则，并且通常部署为局部池化。在未来的工作中，我们将发展来自于调和分析的工具以便于统一这些原则到我们的几何域并且为多尺度分割的统计学习优势提供一些见解性的分析。

3.5 几何深度学习蓝图 (Geometric Deep Learning Blueprint)

3.5.1 几何深度学习蓝图表述

在上述 3.1-3.4 节中讨论的对称性、几何稳定性、尺度分离等的几何原则可结合起来以为学习高维空间数据的稳定表示提供一个统一的蓝图。这些表示将由函数 f 生成，函数 f 作用于定义在域 Ω 的信号 $\mathcal{X}(\Omega, \mathcal{C})$ ，并且函数处于一个对称群 \mathfrak{G} 中。

我们目前讨论到的几何先验并没有描绘一个构建这样的表示的特定的架构，而是描绘了一系列必要条件。然而，这些几何先验暗示了一种公理性的构建模式，这种构建模式以可证明的方式满足这些几何先验，同事保证了高度表达性的表示，这些表示可以趋近任意满足几何先验的函数。

一个简单的初始观察就是，为获得高度表达性的表示，要求我们引入非线性单元，否则若 f 是线性的并且是群 \mathfrak{G} 不变的，那么对于所有的 $x \in \mathcal{X}(\Omega)$ ，有：

$$f(x) = \frac{1}{\mu(\mathfrak{G})} \int_{\mathfrak{G}} f(g.x) d\mu(g) = f\left(\frac{1}{\mu(\mathfrak{G})} \int_{\mathfrak{G}} (g.x) d\mu(g)\right) \quad (3-21)$$

表明函数 f 仅依赖于 x ，并满足一个群平均 (\mathfrak{G} -average)：

$$Ax = \frac{1}{\mu(\mathfrak{G})} \int_{\mathfrak{G}} (g.x) d\mu(g) \quad (3-22)$$

对于图像以及平移，这使得我们仅需要使用输入的平均 RGB 值即可！

尽管这一推理表明线性不变的方程簇并不多么密集（方程簇中方程的种类不多），但线性等变的函数簇却提供了强大的多的函数工具可供选择，这是由于其通过组合合适的非线性单元使得更加丰富且稳定的特征的构建易于实现。实际上，若 $B: \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{X}(\Omega, \mathcal{C}')$ 表示群等变 \mathfrak{G} -Equivariant 的变换，并对所有 $x \in \mathcal{X}, g \in \mathfrak{G}$ 满足 $B(g.x) = g.B(x)$ ，并且非线性映射 $\sigma: \mathcal{C}' \rightarrow \mathcal{C}''$ 表示任意的非线性映射，那么我们可以非常容易地验证组合映射也是群等变的变换，即 $U := (\sigma \circ B): \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{X}(\Omega, \mathcal{C}'')$ 也为群等变作用，其中粗体表示的 σ 意为逐元素的非线性单元的实现，即 $\sigma: \mathcal{X}(\Omega, \mathcal{C}') \rightarrow \mathcal{X}(\Omega, \mathcal{C}'')$ ，满足 $(\sigma(x))(u) := \sigma(x(u))$ 。

这一简单的性质使得我们可以定义一个非常一般的群不变函数簇，可通过将映射 U 与群平均作用 A 组合而得到，即 $A \circ U: \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{C}''$ 。一个直观的问题是：通过选择合适的 B 和 σ ，任意的群不变函数是否能由该模型以任意精度趋近？应用标准万能逼近定理在未结构化的输入矢量数据上来展示浅层“几何”网络也是一种万能逼近器并非难事，可以通过将泛化群平均到一个通用的非线性群不变作用上来实现。不过，正如前述已经探讨过的傅里叶变换与小波变换的对比中，存在根本的矛盾方面：全局不变性与变形稳定性（这一结论被[Zaheer, 2017]证明）。这促进了一种替代性的表示，该替代性的表示考虑局部等变的映射。假定域 Ω 附带一个距离度量 d （对流形、网格、图、群等均可定义度量，但对于集合而言，没有合理的预定义的度量），我们称等变映射 U 是局部的，当满足条件 $(Ux)(u)$ 仅依赖于 $x(u)$ 的值，对邻域 $\mathcal{N}_u = \{v: d(u, v) \leq r\}$ ，其中 r 表示给定的某邻域半径，后者邻域也被称为感受野 (Receptive Field，该词源于神经学，起初指能够影响指定神经元的空间域)。

一个单层的局部等变的映射 U 无法逼近具有长程耦合的函数，但若干局部等变的映射

的组合 $U_J \circ U_{J-1} \cdots \circ U_1$ 可增加感受野并且保持稳定的局部等变性质。感受野可通过遍历粗糙化域（假定具有度量结构）的下采样算子进一步增加，这与 MRA（Multiresolution Analysis, [Mallat, 1999]）一道完成了多分辨率的分析。

总之，输入域的集合结构，结合蕴含的对称群作用知识，为我们提供了三个基本单元：(1) 局部等变映射；(2) 全局不变映射；(3) 粗糙化算子。我们称利用这些基本单元提供的丰富的逼近函数空间辅以预定义的不变性及稳定性性质并组合起来，形成的架构称为集合深度学习蓝图，如图 3.8 所示。

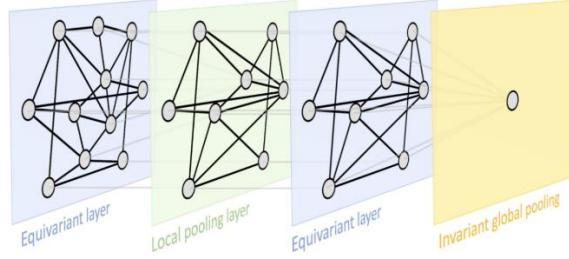


图 3.8 几何深度学习蓝图，以图 (Graph) 为例，一个典型的图神经网络架构可能包含排列等变层（计算逐节点的特征）、局部池化层（图粗糙化）、及排列不变全局池化层（读出层，Readout Layer）。

几何深度学习蓝图 (Geometric Deep Learning Blueprint)

假定 Ω 及 Ω' 为域， \mathfrak{G} 为域 Ω 上对称群，且当 Ω' 为域 Ω 的一个紧致域时，记作 $\Omega' \subseteq \Omega$ 。

我们定义如下基本单元：

线性群 \mathfrak{G} 等变层： $B: \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{X}(\Omega', \mathcal{C}')$ 并满足对于 $\mathfrak{g} \in \mathfrak{G}, \mathbf{x} \in \mathcal{X}(\Omega, \mathcal{C})$ ，有 $B(\mathfrak{g} \cdot \mathbf{x}) = \mathfrak{g} \cdot B(\mathbf{x})$ ；

非线性单元 σ ： $\sigma: \mathcal{C} \rightarrow \mathcal{C}'$ ，并逐元素应用 $(\sigma(x))(u) = \sigma(x(u))$ ；

局部聚合池化： $P: \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{X}(\Omega', \mathcal{C})$ 并满足 $\Omega' \subseteq \Omega$ ；

群 \mathfrak{G} 不变层（全局聚合池化）： $A: \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{Y}$ ，并满足对于任意的 $\mathfrak{g} \in \mathfrak{G}$ 及 $x \in \mathcal{X}(\Omega, \mathcal{C})$ ，有： $A(\mathfrak{g} \cdot x) = A(x)$ ；

使用这些基本组成单元可以构建群 \mathfrak{G} 不变函数 $f: \mathcal{X}(\Omega, \mathcal{C}) \rightarrow \mathcal{Y}$ ，满足：

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

可选定这些组件并使其满足上一个组件的输出可以作为下一个组件输入；不同的任务也许选择不同的群。

3.5.2 不同配置的几何深度学习

我们可以区分出不同的重要的深度学习配置方式，如研究对象域固定的情形、我们仅关注域上变化的信号的情形，或者域也是输入变化的一部分，与信号变化一道构成输入整

体的变化。一个固定的域的经典例子是计算机视觉的应用，例如图片经常被假定再一个固定的域（网格域，Grid）上。图分类是一个域也随着输入变化的样例，其中图的结构以及图上的信号均非常重要（图的结构亦即拓扑结构，信号即输入几何结构，译者注）。在变化的域的例子中，几何稳定性（即对于域变化的不敏感程度）在几何深度学习中扮演者十分关键的角色。

我们构建的几何深度学习蓝图有着通用的特性，可以用于多种几何度量领域。不同的几何深度学习方法在域的选择、对称群、以及特定的部署结构上有些区分，但均可由前述基本的构建单元所描述。正如我们将介绍的，目前应用的大多数类型的深度学习架构均与我们提出的几何深度学习蓝图有着一致性，因而也可以由我们的蓝图导出。

在接下来的章节中（4.1-4.6），我们将介绍多种不同的几何域，聚焦于“5G”，并且在5.1-5.8节中，这些域的特定几何深度学习部署也将一一呈现。

架构	域 Ω	对称群 \mathfrak{G}
CNN	Grid	Translation
Spherical CNN	Sphere / SO(3)	Rotation SO(3)
Intrinsic / Mesh CNN	Manifold	Isometry Iso(Ω) / Gauge symmetry SO(2)
GNN	Graph	Permutation Σ_n
Deep Sets	Set	Permutation Σ_n
Transformer	Complete Graph	Permutation Σ_n
LSTM	1D Grid	Time warping

4 几何域: 5G

本书主要关注图 (Graphs)、网格 (Grids)、群 (Groups)、测地线 (Geodesic)、度规 (Gauges)。本书中的群指的是齐次空间的全局对称变换集合；测地线指流形上的度量结构；度规指定义在局部切丛上的局部参考坐标系。这些概念将在下述内容中详细阐述。在接下来的小节中，我们将深入讨论这些域上的共同的重要元素以及域之间具有区分度的特质以及域上关联的对称群。我们的介绍并非按照由特定到通用的结构进行组织的（实际上网格可以看成是特定的图），相反是按照一种能够凸显几何深度学习蓝图的重要概念的方式展开的。

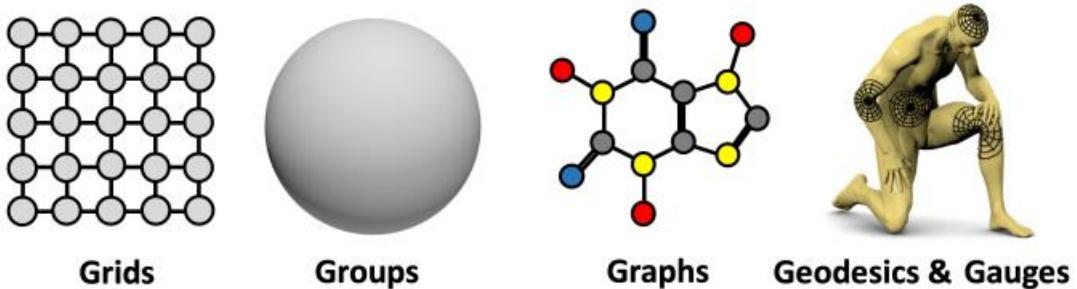


图 4.1 几何深度学习的 5G: 网格、群、对称群上齐次空间、图、测地线与流形度量、度规 (切空间或者特征空间上的局部坐标系)

4.1 图 (Graphs) 与集合 (Sets)

在多种科学分支中，从社会学到粒子物理，图均可以用于关系及作用的建模。在我看来，图衍生了一种由变换群排列模型描述的基本的不变性类型。进一步来说，其他我们关心的研究对象，包括网格即集合，均可以视为一种特定结构的图。

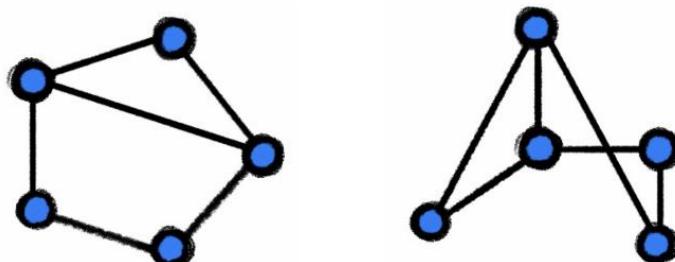


图 4.2 图之间的同构即表示两个图是双射并且仅存在节点序列的不同。

一个图 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ 是一个由节点 \mathcal{V} (Nodes) 与边 \mathcal{E} (Edges) 构成的集合，边描述了具备连接关系的节点作用。在不同的研究领域，节点有时也被称为顶点 (Vertices)，边也可以称为链接 (Links) 或者关联 (Relations)。本书将这些名词等同不加区分地使用。为便于进一步讨论，我们进一步假定节点携带着 s 维的节点特征，用 $\mathbf{x}_u, u \in \mathcal{V}$ 表示。社交网络

也许是图研究中最常应用的示例，其节点表示用户，边表示用户之间的友情连接关系，节点特征描述了用户的特征，例如年龄、主页图片等内容。我们也可以假定边也携带着特征，或者整个图携带着特征。由于边特征的假定并不影响本节的一些研究结论，因而我们将边特征即图特征的研究放在未来的研究工作中予以阐述。

图上一个关键的特征是节点通常并不会按照某种排列顺序提供出来，因而任意对于图的操作均应当不受图中节点的序列次序影响。因而作用于图上的方程应当满足排列不变性(Permutation Invariance)。这也意味着对于两个同构的图而言，函数的输出应当一模一样。我们可以将这一特点视为我们勾画的蓝图的一种特定的设定，即当域 $\Omega = \mathcal{G}$ 且研究空间 $\mathcal{X}(\mathcal{G}, \mathbb{R}^d)$ 是 d 维的逐节点的信号(即节点特征)。我们在这里考虑的群变换就是排列群变换 $\mathcal{G} = \Sigma_n$ ，群元素维所有可能的排列。

我们首先介绍一下集合上的排列不变性，这也是图上排列不变性的一种特例，因为集合可以视为没有边的图，即 $\mathcal{E} = \emptyset$ 。当通过将节点特征按照行进行排列可形成 $n \times d$ 维的特征矩阵 $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)^\top$ ，我们事实上已经给定了节点的排序。排列 $\mathbf{g} \in \Sigma_n$ 作用于节点集合上等同于对矩阵 \mathbf{X} 按照行进行重排列，因而可以表示为一个 $n \times n$ 维的变换矩阵 $\rho(\mathbf{g}) = \mathbf{P}$ ，矩阵中每一行每一列均仅包含一个1，其他全为0(最简单的排序矩阵即为单位矩阵，即什么排序也不做，译者注)。但由于共存在 $n!$ 种变换作用，因而这一群中将有 $n!$ 个元素，即使 n 较小群元素数目也相对较大(由Starling公式可以观察阶乘的阶，译者注)。

一个作用于这样的集合上的函数被称为满足排列不变性，当且仅当对于任意的排列变换矩阵 \mathbf{P} ，总有 $f(\mathbf{P}\mathbf{X}) = f(\mathbf{X})$ 。一个典型的例子是：

$$f(\mathbf{X}) = \phi\left(\sum_{u \in \mathcal{V}} \psi(\mathbf{x}_u)\right) \quad (4-1)$$

其中函数 ψ 可独立地作用于每一个节点的特征，函数 ϕ 是一种和积聚方法，由于和积聚函数对于输入节点的次序没有依赖关系，因而函数 ϕ 对于节点集合的排列作用能够保持不变性，并且对于任何的节点排列均可以返回同样的结果。

类似上述性质的函数提供了一个全局的图尺度的输出，但是很多时候我们关心哪些作用于局部的函数，也就是用于节点层次的计算。例如，我们有时想要应用某些函数对图中的节点特征进行更新，以便于获得节点的潜特征表示。如果我们将这些潜特征表示构造成矩阵的形式 $\mathbf{H} = \mathbf{F}(\mathbf{X})$ (我们使用加粗的表示函数 \mathbf{F} ，目的是强调其是一个逐节点输出的矢量，因而是一个矩阵数据的函数)就不再满足排序不变性了：因为这意味着矩阵 \mathbf{H} 中节点的次序与某一个特定的节点排列绑定了，这样的绑定使得我们能够知道输出的某特征属于哪一个节点。因而我们需要一种更加精细化的排列等变性的定义思想，也就是说，一旦我们对于输入进行一个排序作用，那么这种排序作用将同样地作用域输出，正式的讲， $\mathbf{F}(\mathbf{X})$ 是一个排序等变的函数，满足对于任意的排序矩阵 \mathbf{P} ，满足 $\mathbf{F}(\mathbf{P}\mathbf{X}) = \mathbf{P}\mathbf{F}(\mathbf{X})$ 。一个逐节点的线性变换可以表示为：

$$\mathbf{F}_\Theta(\mathbf{X}) = \mathbf{X}\Theta \quad (4-2)$$

由一个权重矩阵 $\Theta \in \mathbb{R}^{d \times d'}$ 定义, 该函数也是等变函数的一种可能的构造, 导出的潜特征表示为 $\mathbf{h}_u = \Theta^\dagger \mathbf{x}_u$ 。

这样的构建是由我们的几何深度学习蓝图自然地导出的。我们可以首先尝试分析一下线性等变函数 (即形式为 $\mathbf{FPX} = \mathbf{PFX}$ 的函数), 我们可以轻易地验证, 线性等变函数可以视为两个生成器的组合: 单位生成器 $\mathbf{F}_1 \mathbf{X} = \mathbf{X}$ 、平均生成器。

$\mathbf{F}_2 \mathbf{X} = 1/n \times \mathbf{1} \mathbf{1}^\dagger \mathbf{X} = 1/n \times \sum_{u=1}^n \mathbf{x}_u$ 。正如我们将要在 5.4 节中介绍的, 流形的 Deep Sets 架构正式严格地遵循这一蓝图。

我们现在可以将集合上的排序不变性与排序等变性泛化到图上面了。在一般的设定中, 边集合 $\mathcal{E} \neq \emptyset$, 图的连接性可以表示为一个 $n \times n$ 的邻接矩阵 \mathbf{A} , 定义为:

$$a_{uv} = \begin{cases} 1 & (u, v) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \quad (4-3)$$

当图为无向图时, 矩阵 \mathbf{A} 为实对称矩阵。

注意此时邻接矩阵与节点信号矩阵应当是同步关联的, 亦即 a_{uv} 表示的是节点 u 与节点 v 的连接信息, 这与特征矩阵 \mathbf{X} 的第 u 行与第 v 行关联。因此, 将排序矩阵 \mathbf{P} 应用于特征矩阵等效于将其应用于邻接矩阵 \mathbf{A} 的行以及列上, 即 \mathbf{PAP}^\dagger (这一表示正是排序群作用的作用形式)。正式地说, 一个图层次的函数是不变的。满足:

$$f(\mathbf{PX}, \mathbf{PAP}^\dagger) = f(\mathbf{X}, \mathbf{A}) \quad (4-4)$$

对于任意的排序矩阵 \mathbf{P} 成立。

我们说一个函数是排序等变的, 要求其对于任意排序矩阵 \mathbf{P} 满足:

$$\mathbf{F}(\mathbf{PX}, \mathbf{PAP}^\dagger) = \mathbf{PF}(\mathbf{X}, \mathbf{A}) \quad (4-5)$$

这里我们可以首先分析线性等变的函数。正如 [Maron et al, 2018] 指出的, 任意的满足式 (4-5) 的线性函数 \mathbf{F} 可以表示为 15 个线性的生成器, 并且令人印象深刻的是, 这些生成器不依赖于节点数量 n 。这与 Bell 数 (Bell Number) B_4 有关, Bell 数 B_4 即分割四个元素的可分割总数, 在这里即 4 个索引: (u, v) 、 (u', v') , 索引作用域排序矩阵。在这些生成器中, 我们的蓝图特别关注那些有局部性质的生成器, 也就是说, 图中节点 u 的输出直接依赖于周围节点。我们可以在我们的模型中显式地施加这种约束, 这可以通过定义一个节点对于周围节点的作用而实现。

一个 (无向) 图的节点 u 的邻居节点, 通常节点本身也视为其邻居节点, 有时也称为 $1-hop$, 定义为:

$$\mathcal{N}_u = \{v : (u, v) \in \mathcal{E} \text{ or } (v, u) \in \mathcal{E}\} \quad (4-6)$$

邻居节点特征矩阵定义为:

$$\mathbf{X}_{\mathcal{N}_u} = \{\{\mathbf{x}_v : v \in \mathcal{N}_u\}\} \quad (4-7)$$

多集合表示的形式中可以有相同的特征元素出现, 这意味着邻居特征矩阵中元素矢量

不具有唯一性, 当特征相同时这就会发生。邻居特征矩阵作用于 $1-hop$ 邻居的效果与我们的蓝图中的局部性有着很好的关联: 即定义我们的图上的度量为经过若干边的节点间的最短路径。

GDL 蓝图 (即几何深度学习蓝图, 以下将混合使用这两个记法) 生成了一个通用的应用于图上的排序等变函数的构建的菜谱, 这通过定义一个局部函数 ϕ 实现, 该函数作用于节点特征及其邻居, $\phi(\mathbf{x}_u, \mathbf{X}_{\mathcal{N}_u})$ 。然后一个排序等变的函数 \mathbf{F} 可以通过分别应用 ϕ 到每一个节点的邻居来构建, 亦即 (见图 4.3):

$$\mathbf{F}(\mathbf{X}, \mathbf{A}) = \begin{bmatrix} -\phi(\mathbf{x}_1, \mathbf{X}_{\mathcal{N}_1})- \\ -\phi(\mathbf{x}_2, \mathbf{X}_{\mathcal{N}_2})- \\ \vdots \\ -\phi(\mathbf{x}_n, \mathbf{X}_{\mathcal{N}_n})- \end{bmatrix} \quad (4-8)$$

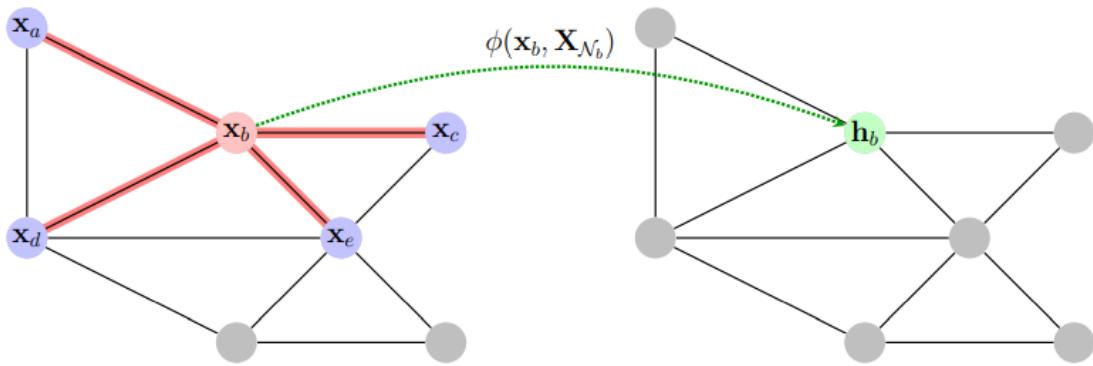


图 4.3 一个构建图上的排列等变函数展示, 通过应用排列不变函数到每一个节点的邻域来实现。在图 (4.3) 中, 函数 ϕ 应用于节点 b 的邻居特征 $\mathbf{x}_{\mathcal{N}_b}$, 其等于 $\mathbf{X}_{\mathcal{N}_b} = \{\{\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c, \mathbf{x}_d, \mathbf{x}_e\}\}$ 。通过应用函数 ϕ 到每一个节点的邻居我们可以得到潜特征表示 $\mathbf{H} = \mathbf{F}(\mathbf{X}, \mathbf{A})$

由于函数 \mathbf{F} 由共享的函数 ϕ 以局部的方式作用到每一个节点来构建, 其排列等变形依赖于函数 ϕ 的输出不独立于节点邻居的次序。因而, 如果函数 ϕ 为排列不变的, 那么这样的特质就能够得到满足。正如我们将在未来的工作中进一步介绍的, 函数 ϕ 的不同选择在表达我们这一蓝图中扮演着至关重要的角色。当 ϕ 是一对一映射时, 其等价于经典图同构判定算法 Weisfeiler-Lehman 算法的第一步 (该算法是通过迭代式细化流程判定两个图是否同构的必要条件的经典算法)。

值得一提的是, 定义在集合上的函数与定义在图上的函数的不同之处在于, 图上的函数需要明确地说明域的结构。因此, 图不同于集合, 其域也是机器学习问题输入的一部分, 但在网格及集合中时, 我们通常仅关心特征并假定域是固定的。这一区别将在本文中作为一个经常出现的主题来说明。因而, 几何稳定性 (Geometric Stability) 的概念 (即对于图的扰动保持不变的能力) 对于大多数图上的学习算法至关重要。几何稳定性直接地遵循了我们的蓝图构建, 因为排列等变函数及排列不变函数能够在同构的图上输出一样的结果。这些结果也可以应用于准同构的图上, 以及在图上扰动存在的条件下能够生成若干稳

定的结果[Levie et al, 2018]。我们将在介绍流形时详细介绍这一点，并且将其作用一个有力的武器来进一步研究不变性。

其次，由于图、网格、的额外的结构（与集合相比），他们可以粗糙化（Coarsening）表示，这也诞生了众多池化算法（Pooling）。对于集合而言我们不能够定义非朴素的池化操作，当然也存在具有拓扑连接性的集合，在这种情况下其可以应用非朴素（Non-trivial）的池化操作。

4.2 网格（Grids）与欧氏空间（Euclidean Space）

我们考虑的第二种研究对象就是网格（Grids）。我们可以深度学习的影响力在计算机视觉领域、自然语言处理领域、语音识别领域等非常显著。这些应用均共享一个几何分母：一个蕴含的网格结构。正如前述提及的，网格是一种特殊的图，其邻接矩阵有着特殊的结构。然而，由于网格中节点的次序是固定的，因此应用于网格的机器学习模型仅考虑特征，也无需关心排列不变性要求，因而有一个更强的几何先验约束：平移不变群。

4.2.1 循环矩阵及卷积（Circulant Matrices and Convolutions）

我们来深入分析一下平移不变群的作用。为简单起见，假定周期性的边界条件，我们可以认为一维的网格是一个环图，该环图上附带着节点特征，索引为 $0, 1, \dots, n-1 \bmod n$ （我们将在后面忽略取模操作），其邻接矩阵为 $a_{u,u+1 \bmod n} = 1$ ，其他索引情况为 0。这与我们前边讨论的图结构有两点的主要区别。第一，每一个节点 u 均有一样的连接性，即与节点 $u-1$ 及节点 $u+1$ 连接，因而其内部结构与其他类似的一维网格无法区分。第二点，也更重要的是，既然节点有着固定的次序，其邻居节点也是固定的，我们通常称为左邻居节点与右邻居节点。若我们应用我们前述的配方去设计等变函数 \mathbf{F} ，使用一个局部聚合的函数 ϕ ，我们现在有： $\mathbf{f}(\mathbf{x}_u) = \phi(\mathbf{x}_{u-1}, \mathbf{x}_u, \mathbf{x}_{u+1})$ 作用于网格中的每一个节点，函数 ϕ 也无需是排列不变性的函数了。对于一个特定的选择：

$\phi(\mathbf{x}_{u-1}, \mathbf{x}_u, \mathbf{x}_{u+1}) = \theta_{-1}\mathbf{x}_{u-1} + \theta_0\mathbf{x}_u + \theta_1\mathbf{x}_{u+1}$ 我们可以将函数 \mathbf{F} 写为：

$$\mathbf{F}(\mathbf{X}) = \begin{bmatrix} \theta_0 & \theta_1 & & \theta_{-1} \\ \theta_{-1} & \theta_0 & \theta_1 & \\ & \ddots & \ddots & \ddots \\ & & \theta_{-1} & \theta_0 & \theta_1 \\ \theta_1 & & \theta_{-1} & \theta_0 & \theta_0 \end{bmatrix} \begin{bmatrix} -\mathbf{x}_0 & - \\ -\mathbf{x}_1 & - \\ \vdots \\ -\mathbf{x}_{n-2} & - \\ -\mathbf{x}_{n-1} & - \end{bmatrix} \quad (4-9)$$

值得一提的是这一个特定的多对角线结构的函数在一些机器学习文献中被称作参数共享（Weight Sharing）。

更一般地，给定一个矢量 $\boldsymbol{\theta} = (\theta_0, \theta_1, \dots, \theta_{n-1})$ ，一个循环矩阵 $\mathbf{C}(\boldsymbol{\theta}) = (\theta_{u-v \bmod n})$ 可以由追加循环左移或右移的矢量 $\boldsymbol{\theta}$ 来构造。由于 $\mathbf{C}(\boldsymbol{\theta})\mathbf{x} = \mathbf{x} \star \boldsymbol{\theta}$ ，循环矩阵与离散卷积是同义

词。正是由于边界的存在，因而是循环的，在信号处理中， θ 也通常被称为滤波器，在CNN中其参数为可学习的参数。

$$(\mathbf{x} \star \boldsymbol{\theta})_u = \sum_{v=0}^{n-1} x_{v \bmod n} \theta_{u-v \bmod n} \quad (4-10)$$

一个特殊的选择是 $\boldsymbol{\theta} = (0, 1, 0, \dots, 0)^\dagger$ ，可以获得一个特殊的循环矩阵，该矩阵表示右移矢量一个位置。这一矩阵被称为右移算子或移动算子，用 \mathbf{S} 表示。通常左移算子用 \mathbf{S}^\dagger 表示，并且左右移算子属于正交群 $O(n)$ ，有： $\mathbf{S}^\dagger \mathbf{S} = \mathbf{S} \mathbf{S}^\dagger = \mathbf{I}$ 。循环矩阵有着可交换的良好性质，也即： $\mathbf{C}(\boldsymbol{\theta})\mathbf{C}(\boldsymbol{\eta}) = \mathbf{C}(\boldsymbol{\eta})\mathbf{C}(\boldsymbol{\theta})$ 对任意的 $\boldsymbol{\eta}, \boldsymbol{\theta}$ 均成立。由于移动操作可形成循环矩阵，我们可以得到熟悉的平移等变或左右移等变的卷积算子：

$$\mathbf{S}\mathbf{C}(\boldsymbol{\theta})\mathbf{x} = \mathbf{C}(\boldsymbol{\theta})\mathbf{S}\mathbf{x} \quad (4-11)$$

这一可交换特性并不令人意外，因为蕴含的对称群（平移群）正是一个阿贝尔群。此外，逆向的命题也是对的，亦即一个矩阵对移动可交换时其一定为循环矩阵。这反过来也使得我们可以定义卷积算子为平移等变线性操作，这也是一个非常好的几何先验的展示，同时也是整个几何机器学习的哲学所在：卷积起源于平移对称的第一准则。

值得注意的是，不像图及集合，网格的线性无关的移动等变函数（卷积）的数量随着域的增长而增加（这是由于循环矩阵仅有一个对角线的自由度）。然而尺度分离先验保证了滤波器是局部的，使得同一层具有 $\Theta(1)$ 的复杂度。当讨论卷积神经网络架构的部署时我们将细致讨论这些准则。

4.2.2 离散傅里叶变换推导 (Derivation of the Discrete Fourier Transform)

我们已经提到过傅里叶变换及其与卷积的联系了：傅里叶变换对角化了卷积操作是一个信号处理领域重要的性质，这使得可以在频域以逐元素相乘的方式进行卷积运算。然而，众多书籍中通常仅仅介绍一下这样的一个事实，很少介绍傅里叶变换是怎么来的，为什么傅里叶基底这么特殊等问题。在这里我们可以展示，这些一般的对称性原则对于傅里叶变换将有多么基础的作用。

为此，首先要回想一个事实：当且仅当线性矩阵之间可交换时，线性矩阵可被联合对角化。换句话说，循环矩阵存在一种公共的基底，这些循环矩阵仅存在特征值的不同。我们通常额外地假设循环矩阵有着不同的特征值，否则的话其将有不同的对角化表示，我们可以通过选择矩阵 \mathbf{S} 来使得满足这样的额外要求。因此我们可以选择一个循环矩阵进而计算其特征向量，然后我们就有了其他循环矩阵的特征向量。为方便起见通常可以选择移动算子的循环矩阵，其特征矢量正是离散傅里叶基底：

$$\boldsymbol{\varphi}_k = \frac{1}{\sqrt{n}} \left(1, e^{\frac{2\pi i k}{n}}, e^{\frac{4\pi i k}{n}}, \dots, e^{\frac{2\pi i (n-1)k}{n}} \right)^\dagger, \quad k = 0, 1, 2, \dots, n-1 \quad (4-12)$$

此处 \mathbf{S} 为正交矩阵但并不是对称矩阵，因此其特征向量是正交的，但是特征值是复数。我们可以组合式 (4-12) 形成矩阵 $\Phi = (\boldsymbol{\varphi}_0, \boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_{n-1})$ ，由于该矩阵为复数矩阵，因此用 Φ^* 表示其转置共轭矩阵，乘以转置共轭矩阵就得到了离散傅里叶变换 (Discrete Fourier Transform, DFT)，乘以矩阵本身就得到了傅里叶逆变换：

$$\begin{aligned} \hat{x}_k &= \frac{1}{\sqrt{n}} \sum_{u=0}^{n-1} x_u e^{-\frac{2\pi k u i}{n}} \\ x_u &= \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} \hat{x}_k e^{\frac{+2\pi k u i}{n}} \end{aligned} \quad (4-13)$$

由于傅里叶变换是一个正交矩阵 $\Phi^* \Phi = \mathbf{I}$ ，因而在几何上其相当于以旋转的方式改变系统的坐标，在傅里叶的域中，循环矩阵的作用就变成了逐元素的乘积。由于所有的循环矩阵均可被联合对角化，所以循环矩阵对角化地表示为其傅里叶变换，仅有的区别在于特征值。由于循环矩阵的特征值正是傅里叶变换的滤波器 [Bamieh, 2018]，即 $\hat{\theta} = \Phi^* \theta$ ，由此我们得到了著名的卷积定理：

$$\mathbf{C}(\theta) \mathbf{x} = \Phi \begin{bmatrix} \hat{\theta}_0 \\ \ddots \\ \hat{\theta}_{n-1} \end{bmatrix} \Phi^* \mathbf{x} = \Phi (\hat{\theta} \odot \hat{\mathbf{x}}) \quad (4-14)$$

由于傅里叶矩阵 Φ 有着特殊的代数结构，积式 $\Phi \mathbf{x}$ 以及 $\Phi^* \mathbf{x}$ 可以应用快速傅里叶变换 (Fast Fourier Transform, FFT) 来计算，计算的时间复杂度为 $\mathcal{O}(n \log n)$ 。这也是为什么频域滤波在信号处理中是如此流行的原因之一，此外，信号典型地定义在频域，因而傅里叶变换 $\hat{\theta}$ 几乎从来无需显式计算。

除了对于傅里叶变换的推导以及卷积运算，傅里叶变换提供了一个系统性的架构，用以将这些概念泛化到图上。注意环图的邻接矩阵正是移动算子，我们可以推出图上的傅里叶变换以及卷积运算的类比算子，这可以通过计算邻接矩阵的特征向量实现（参见 [Sandryhaila and Moura, 2013]）。早期的发展图上的神经网络的尝试是通过类比 CNN 来实现的，有时也被称为“Spectral GNNs”，这也正是我们提出的蓝图的应用。在图信号处理中，我们通常用图拉普拉斯矩阵来代替邻接矩阵，用以计算傅里叶变换，参见 [Shuman et al, 2013]，在网格中两个矩阵有着相同的特征向量，但在图中二者的特征向量不同了，图中图拉普拉斯矩阵的特征向量与图拉普拉斯矩阵的构造方式有关。正如我们将在 4.4-4.6 节中将详细展现的，这样的类比也引入了重要的局限性。第一个局限性来自于网格是固定的结构，因而所有作用在其上的信号均可以写成固定的傅里叶基底，与之对比，在一般的图上，傅里叶基底取决于图的结构，因而我们无法对来自不同的图上的傅里叶变换基底直接进行比较，这也导致了在机器学习任务中缺乏一定的泛化能力。第二，多维的网格，通常

可表示为一维的网格的张量积的形式，保持着内在的结构：傅里叶基底以及关联的频率可以划分到不同的维度。例如在图像中，我们可以自然地讨论横向或者纵向的频率，其上的滤波器也有着方向性。在图上，由于我们仅仅可以将傅里叶基底方程以相关的频率以及强度进行分类，这导致了傅里叶域的结构是一维的。因而，图滤波器没有方向性或者说是各向同性。

4.2.3 连续傅里叶变换推导 (Derivation of the Continuous Fourier Transform)

为了介绍完整，同时也为了接下来讨论的方便，我们也对连续的情况进行分析。正如 3.4 节中介绍的一样，考虑定义在域 $\Omega = \mathbb{R}$ 上的函数 f ，以及可将函数平移一个距离的平移算子 $(S_v f)(u) = f(u - v)$ ，在傅里叶基函数 $\varphi_\xi(u) = e^{i\xi u}$ 上应用平移算子 S_v ，利用指数函数的结合律，可得到：

$$S_v e^{i\xi u} = e^{i\xi(u-v)} = e^{-i\xi v} e^{i\xi u} \quad (4-15)$$

即 $\varphi_\xi(u)$ 为平移算子 S_v 的复数特征向量，其特征值为 $e^{-i\xi v}$ ，这与我们在离散的情况获得的结果有着高度对称性。由于平移算子 S_v 为单位算子（即对于任意的 p 以及 $x \in L_p(\mathbb{R})$ ，满足条件： $\|S_v x\|_p = \|x\|_p$ ），则对于其任意一个特征值而言必须满足： $|\lambda| = 1$ ，这与式 (4-15) 中的 $e^{-i\xi v}$ 相吻合。此外，平移算子的谱分解是单纯的 (Simple)，这意味着两个共享特征值的算子必须是协线性的 (Colinear)。确实，假定对于一些 ξ_0 有 $S_v f = e^{-i\xi_0 v} f$ ，在式子两侧做傅里叶变换，我们得到：

$$\forall \xi, e^{-i\xi v} \hat{f}(\xi) = e^{-i\xi_0 v} \hat{f}(\xi) \quad (4-16)$$

这表明对于 $\xi \neq \xi_0$ 的情况， $\hat{f}(\xi) = 0$ ，因此 $f = \alpha \varphi_{\xi_0}$ 。

对于一个一般的满足平移等变的线性算子 C （即满足 $S_v C = C S_v$ ），我们得到：

$$S_v C e^{i\xi u} = C S_v e^{i\xi u} = e^{-i\xi v} C e^{i\xi u} \quad (4-17)$$

这表明 $C e^{i\xi u}$ 也是平移算子 S_v 的特征方程（特征方程与特征向量是同义词，但特征向量多用于离散情况，特征方程多用于连续情况），其也满足谱的简化表示 $C e^{i\xi u} = \beta \varphi_\xi(u)$ 。换句话说，傅里叶基底正是所有平移等变算子的基底。所以， C 在傅里叶频域被对角化了并且可以表示成 $C e^{i\xi u} = \hat{p}_C(\xi) e^{i\xi u}$ ，其中 $\hat{p}_C(\xi)$ 是作用在不同频率 ξ 上的转移方程。最终，对于任意的函数 $x(u)$ ，根据线性：

$$\begin{aligned} (Cx)(u) &= C \int_{-\infty}^{+\infty} \hat{x}(\xi) e^{i\xi u} d\xi = \int_{-\infty}^{+\infty} \hat{x}(\xi) \hat{p}_C(\xi) e^{i\xi u} d\xi \\ &= \int_{-\infty}^{+\infty} p_C(v) x(u - v) dv = (x \star p_C)(u) \end{aligned} \quad (4-18)$$

这一平移群的频域特征是一种更加一般的泛函分析中结果的一种特例，即斯通定理

(Stone Theorem), 这一定理分析了对于任意单参数的单位群的特点。其中 $p_C(u)$ 是 $\hat{p}_C(u)$ 的傅里叶逆变换，因而任意的线性平移等变算子均可视为卷积。

4.3 群 (Groups) 与齐次空间 (Homogeneous Space)

我们上述的讨论指出了左移与右移和卷积的密切关联：卷积就是线性移动等变的操作，并且相反，任意的移动等变线性算子可视为一种卷积。进一步说，移动算子可以为傅里叶变换联合对角化。正如其即将展现的，这一特性是一个更大的场景的一部分：卷积以及傅里叶变换可以被定义为任意我们可以加和或者积分的对称群。严格来讲，我们需要群是局部紧致的，从而存在一个左移不变的 Haar 测量。对该测量进行积分，我们可以应用群元素移动积分项并将获得同样的结果，这是由于：

$$\int_{-\infty}^{+\infty} x(u) du = \int_{-\infty}^{+\infty} x(u-v) du, \quad x: \mathbb{R} \rightarrow \mathbb{R} \quad (4-19)$$

考虑欧式空间 $\Omega = \mathbb{R}$ ，我们可将卷积理解为一种模式匹配算子：我们将输入信号 $x(u)$ 与移动后的滤波器 $\theta(u)$ 进行匹配。卷积运算 $(x * \theta)(u)$ 在一点 u 的值就是信号 x 与移动后的滤波器的内积：

$$(x * \theta)(u) = \langle x, S_v \theta \rangle = \int_{\mathbb{R}} x(v) \theta(u+v) dv \quad (4-20)$$

注意我们这里定义的卷积并非数学上的卷积，而是跨-协关联 (Cross-Correlation)，这样的卷积定义通常用于机器学习中。为了统一我们在接下来也称之为卷积，在我们的记法中分别为 $(\rho(g)x)(u) = x(u-v)$ 以及 $(\rho(g^{-1})x)(u) = x(u+v)$ 。并且注意此时 u 既是欧式空间中的一个点，也是一个平移群的元素，这时我们把群与域等同起来，即 $\mathfrak{G} = \mathbb{R}$ 。我们现在将展示怎么泛化这一构建，这可以简单地通过用另一个作用于域 Ω 上的群 \mathfrak{G} 来替换平移群。

4.3.1 群卷积 (Group Convolution)

正如第 3 章中讨论的，群 \mathfrak{G} 在域 Ω 上的作用引出了群 \mathfrak{G} 的表示 ρ ，作用于信号空间 $\mathcal{X}(\Omega)$ 上，即 $\rho(g)x(u) = x(g^{-1}u)$ 。在上述的例子中，群 \mathfrak{G} 就是平移群，其元素通过左右移作用到点 u 的坐标上，其中 $\rho(g)$ 就是作用在信号上的移动算子，形如 $(S_v x)(u) = x(u-v)$ 。最后，为在信号上应用滤波器，我们假定信号空间 $\mathcal{X}(\Omega)$ 为希尔伯特空间，其具有内积结构：

$$\langle x, \theta \rangle = \int_{\Omega} x(u) \theta(u) du \quad (4-21)$$

这一积分是通过对域上的不变度量进行积分的，对于离散的情况，积分意味着在域上进行求和运算。为简单起见，我们假定信号空间数据为标量数据，即 $\mathcal{X}(\Omega, \mathbb{R})$ ，一般而言内积具有式第 3 章对于内积的介绍中的结构形式。

当定义好了如何对信号进行变换并于滤波器进行匹配后，我们可以定义域 Ω 上信号的群卷积为：

$$(x \star \theta)(g) = \langle x, \rho(g)\theta \rangle = \int_{\Omega} x(u) \theta(g^{-1}u) du \quad (4-22)$$

注意 $x \star \theta$ 的输入是群元素 g 而非域上的点。因此在下一层接受 $x \star \theta$ 作为输入信息时，应当作用在定义在群 \mathfrak{G} 上的信号，这一要点我们将在后续回过头来进行分析。

正如传统的欧式卷积是移动等变的，更加一般的群卷积是 \mathfrak{G} -等变的。一个关键的观察是将信号 x 结合群元素 g 作用的滤波器 $\rho(g)\theta$ 的运算与逆变换信号 $\rho(g^{-1})x$ 结合未变换的滤波器进行了匹配。从数学角度看，这一观察可以表示为： $\langle x, \rho(g)\theta \rangle = \langle \rho(g^{-1})x, \theta \rangle$ 。根据这一洞察，群 \mathfrak{G} -等变的群卷积依据其定义遵循如下式 (4-23)，同时也遵循群表示的定义特征式 $\rho(h^{-1})\rho(g) = \rho(h^{-1}g)$ 。

$$(\rho(h)x \star \theta)(g) = \langle \rho(h)x, \rho(g)\theta \rangle = \langle x, \rho(h^{-1}g)\theta \rangle = \rho(h)(x \star \theta)(g) \quad (4-23)$$

让我们观察一些例子。我们曾研究过的一维的网格，选定域

$\Omega = \mathbb{Z}_n = \{0, 1, \dots, n-1\}$ ，以及循环移动群 $\mathfrak{G} = \mathbb{Z}_n$ 。群元素就是索引的循环移动，也就是说，一个 $g \in \mathfrak{G}$ 元素可以视为 $u = 0, 1, \dots, n-1 \bmod n$ 。重要的是，这个例子中群的元素也是域的元素。放松一下记法的要求，我们因而可以认为两个结构相等；在这种情况下我们的群卷积为：

$$(x \star \theta)(g) = \sum_{v=0}^{n-1} x_v \theta_{g^{-1}v} \quad (4-24)$$

这引出了我们熟悉的卷积式： $(x \star \theta)_u = \sum_{v=0}^{n-1} x_v \theta_{v+u \bmod n}$ 。（注意这里仍然并非数学上的卷积）。

4.3.2 球状卷积 (Spherical Convolution)

现在考虑一下二维的球面 $\Omega = \mathbb{S}^2$ ，其上的对称群为旋转群，特定地，我们考虑特殊正交群 $\Omega = \text{SO}(3)$ ，尽管这一选择是出于本节示范性的原因，实际上这一选择也是非常具备实际价值的，其在众多应用中出现。例如，在天文物理研究中，观测数据通常天然地具有球状几何。此外，球状对称在化学学科的众多应用中也具有重要的价值，尤其是在对分子进行建模以及尝试预测其性质时，这可以用于虚拟药物的筛查。

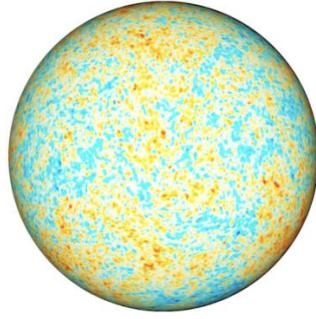


图 4.4 宇宙微波辐射就是在球状域上的信号，图为普朗克望远镜观测数据

使用一个三维的单位矢量 \mathbf{u} 表示球面上的一点，即要求: $\|\mathbf{u}\|=1$ ，群作用可以表示程一个 3×3 的正交群 \mathbf{R} 并且要求 $\det(\mathbf{R})=1$ (这一要求即约束了为特殊正交群, 译者注)。球状卷积因而可以写成信号与旋转后的滤波器的内积:

$$(x \star \theta)(\mathbf{R}) = \int_{\mathbb{S}^2} x(\mathbf{u}) \theta(\mathbf{R}^{-1} \mathbf{u}) d\mathbf{u} \quad (4-25)$$

首先值得注意的是，现在群就与域不再相同了：群 $SO(3)$ 是一个李群，也是一个三维的流形，但域 \mathbb{S}^2 是二维的一个空间。因此在这样的情况下，情况就与之前的示例不同了，卷积操作是定义在群空间 $SO(3)$ 而不是域 Ω 上了。

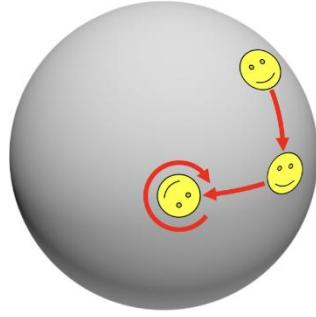


图 4.5 群 $SO(3)$ 的作用有三种类型 (图中红色线)，群本身也是一个三维流形

群与域的不同引入了严重的实际问题：在我们的几何深度学习蓝图中，我们将多个等变映射（即深度学习术语中的层）串联起来，下一层的输入就是上一层的输出。在平移群的情况下，由于其函数均定义在相同的域上，我们可以按照次序依次执行卷积操作。但在一般的情况下，由于 $x \star \theta$ 是一个定义在群 \mathfrak{G} 而非域 Ω 上的，我们不能依次做同样的操作，也就是说下一层必须处理定义在 \mathfrak{G} 上的信号，即 $x \in \mathfrak{G}$ 。我们定义的群卷积就排上用场了：我们假定群 \mathfrak{G} 本身作用于的域 $\Omega = \mathfrak{G}$ ，即假定函数空间与函数作用的空间均为群，这可以通过群作用实现: $(g, h) \mapsto gh$ ，即使用组合群作用。这使得群作用的表示 $\rho(g)$ 作用于信号 $x \in \mathcal{X}(\mathfrak{G})$ 上，作用形式为: $(\rho(g)x)(h) = x(g^{-1}h)$ 。群作用于群元素本身的表示也称为群的正规表示 (Regular Representation)。正如之前定义的一样，定义在群空间的内积等于对信号以及域上的滤波器进行逐元素乘积求和，只不过此处信号的域为 $\Omega = \mathfrak{G}$ 。在我们的球状卷积的实例中，第二层卷积将有如下形式：

$$((x \star \theta) \star \phi)(\mathbf{R}) = \int_{SO(3)} (x \star \theta)(\mathbf{Q}) \phi(\mathbf{R}^{-1} \mathbf{Q}) d\mathbf{Q} \quad (4-26)$$

由于卷积引入的内积运算要求对整个域进行积分或求和，我们仅可以在域本身较小（离散的情况）或者低维度时（连续的情况）进行运算。例如，我们可以在二维平面 \mathbb{R}^2 或者三维的特殊正交群 $SE(3)$ 上进行卷积运算，也可以有限个节点的图上进行卷积运算，但是我们难以在排列群 Σ_n 上进行运算，这是由于其包含 $n!$ 个元素。在高维中进行积分运算，例如在仿射变换群（包含平移群、旋转群、切变群、放缩群等，共 6 个维度）中也是不具备现实的操作性的。不管怎样，正如我们将在 5.3 节中介绍的，我们仍然可以通过定义在低纬度域 Ω 并且由群 \mathfrak{G} 作用的信号上进行等变的卷积操作。事实上，任意作用在两个不同的域 Ω, Ω' 上的等变线性映射 $f: \mathcal{X}(\Omega) \rightarrow \mathcal{X}(\Omega')$ 均可以写成类似此处定义的群卷积形式的一般卷积表示。

其次，我们注意到之前从卷积的移动等变性推导的傅里叶变换也可以扩展为更一般的形式，这可以通过将信号投影至对称群的不可约的矩阵元素上。我们将在未来对此进行深入分析。在此处探讨的 $SO(3)$ 的例子中，这一球状卷积形式使得球状调和（Spherical Harmonics）以及 Wigner-D 函数（Wigner D-function）出现了，这两种技术在量子物理及化学中有诸多应用。

最后我们也指出知道目前的讨论中构成我们假设的基础：不论域 Ω 是一个网格，还是球，我们均可以将任意一点变换到另一点，直觉来讲，这意味着域上的所有点都看起来一样（“look the same”）。具有这样的特质的域 Ω 通常称之为齐次空间，正式来说即：对于任意的点 $u, v \in \Omega$ 存在群变换 $g \in \mathfrak{G}$ 满足 $g.u = v$ ，对于群中幺元，有 $e.u = u$ ，及复合群作用： $g(h.u) = (gh).u$ 。我们将在下一节尽量放松这一齐次性假定。

4.4 测地线 (Geodesics) 与流形 (Manifolds)

在我们上一个例子中，球 S^2 是一个流形，尽管其由于齐次结构是一个具有全局群的特殊的一个流形。李群 $SO(3)$ 也是一个流形，实际上所有的李群均是流形，这也是李群的定义要求。不幸的是，其他大多数的流形并非均具备全局的对称群。在这样的情况下，我们也不能直接地在流形域 Ω 上的信号上定义群 \mathfrak{G} 的作用，也难以直接地应用群作用在周围邻域移动滤波器去定义经典的卷积运算的推广形式。无论如何，流形的确存在两种类型的不变性，正如我们将要在本节中介绍的：保持度量结构的变换、局部坐标系的变化。

尽管对于很多机器学习相关方向的读者而言流形似乎是一种陌生的研究对象，他们实际上在众多的科学领域经常出现。在物理中，流形在对我们所处的宇宙进行建模中起着中心作用：根据爱因斯坦广义相对论，重力起因于时空的弯曲，这可以用黎曼流形来描述。在一些稍显“平庸”的研究领域，例如计算机图形学及计算机视觉中，流形也是一种 3 维物体常用的建模方式。这种三维模型更加广阔的应用包含 VR、AR、运动捕捉的特殊效应、像是三维迷宫形态的蛋白质粘结作用的生物结构等。这些应用的共同分母就是应用流形来表示一些三维物体的边界。

这些模型如此方便的原因根植于一些起因。首先，流形描述方式提供了一个三维模型的紧致描述，消除了需要为空的空间分配内存的需要（在基于网格的表示中需要为空的空间分配内存）。其次，应用流形使得我们可以不关注内部的结构，这是一个有用的性质，例如在生物结构中，蛋白质分子的内部折叠通常与分子表面的相互作用无关，第三，也是最重要的是，人们经常需要处理变形的物体，尤其是塑性变形。我们自身的身体就是这样一个例子，其他计算机图形学及计算机视觉中众多的应用，例如前述提及的运动捕捉、虚拟化身等均要求变形稳定性。在这样的变形可以通过保有内参结构的流形的变换来建模，黎曼度量也即流形上点之间的距离，无需关心流形本身是怎么嵌入到环境空间中的。

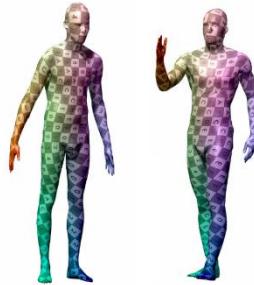


图 4.6 人体也是三维流形，可以以准同构的方式塑性变形

我们将在几何深度学习蓝图中着重分析变化域的流形（也就是流形发生形变，译者注），这与图的变化类似。我们将着重分析域变形的不变性概念，我们在 3.3 节中称这些特性为几何稳定性（Geometric Stability）。由于很多机器学习相关的读者可能对于微分几何较为陌生，我们将介绍一些所需的基本概念，对于有兴趣的读者可以阅读 [Penrose, 2005]。

4.4.1 黎曼流形（Riemann Manifolds）

由于流形的正式定义需要较多的背景知识，因而我们将从直观角度出发介绍流形，当然这种介绍将牺牲一些严谨性。在本文中，我们将局部为欧式空间的多维度曲面的光滑流形介绍为流形（光滑也意味着无穷次可微，对于有穷次可微的流形而言，需要指明其可微阶数，译者注），局部欧式空间意味着流形上的任意一个小的邻域可以看成是欧式空间 \mathbb{R}^s 上的一个邻域经过变形得到的，我们也称此流形为 s 维流形。这一定义使得我们可以在流形上的一点 u 的邻域内用欧式空间来趋近，即使用切面空间 $T_u \Omega$ 。后者可以考虑一个典型的二维流形：球来进行可视化，切面空间就是球上的切面（图 4.7）。所有切面空间的集合也叫做切丛（更一般地，切面空间称为图册（Chart），所有图册的集合称为图册集（Atlas），译者注），用符号 $T\Omega$ 表示。正式地说，切丛指的是不相关的切面空间集合（Disjoint Union，如式（4-27））。我们将在 4.5 节中深入分析这一概念。

$$T\Omega = \bigsqcup_{u \in \Omega} T_u \Omega \quad (4-27)$$

切面上一个切矢量用 $X \in T_u \Omega$ 表示，可以看成是流形上从点 u 出发的一个位移。为了

度量切矢量的模长以及不同切矢量之间的夹角，我们需要切空间附加额外的结构，定义为正定的与点 u 有关的双线性映射 $g_u: T_u \Omega \times T_u \Omega \rightarrow \mathbb{R}$ 。正定性意味着对于任意非零的输入，输出将大于 0，而输出等于 0 当且仅当输入为 0，如果这样的映射用矩阵 \mathbf{G} 表示，则使用符号 $\mathbf{G} \succ 0$ 表示正定性，其行列式的开平方代表着体积元素，与基底的选择无关。这样一个映射称为黎曼度量 (Riemann Metric)，这一命名是为了致敬在 1856 年引出该概念的 Bernhardt Riemann。这一映射可以视为是一种切面空间上的内积： $\langle X, Y \rangle_u = g_u(X, Y)$ ，其中切矢量 $X, Y \in T_u \Omega$ 。这一度量也引出了局部空间上切矢量模长的度量方法：

$$\|X\|_u = g_u^{1/2}(X, X)。$$

我们也必须指出，切矢量是一种抽象的几何实体，与坐标系无关。如果我们用一系列数来表示切矢量，我们则必须将其表示为一系列的坐标分量 $\mathbf{x} = (x_1, x_2, \dots, x_s)$ ，这一表示与某一局部坐标系 $\{X_1, X_2, \dots, X_s\} \subseteq T_u \Omega$ 有关。由于矢量经常与其坐标系混淆在一起，因此我们使用 X 表示坐标系，而 \mathbf{x} 表示在此坐标系中的坐标矢量表示。类似地，度量也可以表示为一个 $s \times s$ 的矩阵 \mathbf{G} ，其元素 $g_{ij} = g_u(X_i, X_j)$ ，我们将在 4.5 节中深入介绍这一点。

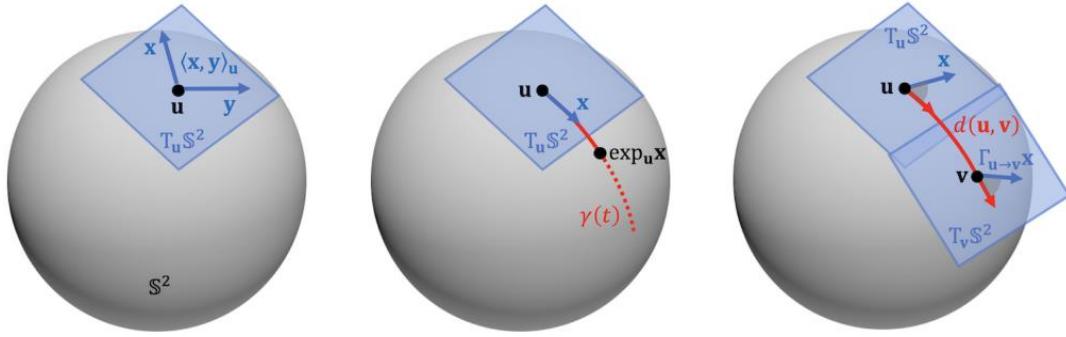


图 4.7 二维球面 $S^2 = \{\mathbf{u} \in \mathbb{R}^3 : \|\mathbf{u}\| = 1\}$ 上的黎曼几何的展示，这一球面由一系列三维平面包裹。切平面可以表示为： $T_u S = \{\mathbf{x} \in \mathbb{R}^3 : \mathbf{x}^\dagger \mathbf{u} = 0\}$ ，切面为二维平面，因而流形维度为 2。测地线就是大圆的弧长： $d(\mathbf{u}, \mathbf{v}) = \cos^{-1}(\mathbf{u}^\dagger \mathbf{v})$ 。指数映射表达式为： $\exp_u(\mathbf{x}) = \cos(\|\mathbf{x}\|)\mathbf{u} + \mathbf{x}\sin(\|\mathbf{x}\|)/\|\mathbf{x}\|$ ， $\mathbf{x} \in T_u S^2$ 。黎曼度量就是切面上两个矢量的内积： $\langle \mathbf{x}, \mathbf{y} \rangle_u = \mathbf{x}^\dagger \mathbf{y}$ ， $\mathbf{x}, \mathbf{y} \in T_u S^2$ 。

附带着度量的流形称为黎曼流形，一些仅依赖于度量的量称为内参。这一概念对我们的讨论至关重要，因为根据几何深度学习蓝图，我们将探索构造作用在域 Ω 的信号上的映射，并要求该映射对度量保持的变换具有不变性，这样的度量保持的变换也称为等距等角变换，这样的变换将流形进行变形时也不改变流形的局部结构。如果这样的映射能够写为仅依赖内参量的形式，他们则自动地就满足了等距同构不变性 (Isometry-Invariant) 因而能够不受等距同构变形影响。这些结论也可以扩展到准等距同构的变换中，也就是我们蓝图中讨论的对于域变形的几何稳定性的例子。



图 4.8 维持度量结构的变换由嵌入定理给定[Nash, 1956]，图中表示平面的等距同构的嵌入表示，来自 Shutterstock/300 libs

正如我们提到的，尽管黎曼流形的定义不要求一个具体的空间的几何实现（没有具体的实现的黎曼流形也称为抽象黎曼几何，Abstract Riemann Geometry），实际上任意的光滑的黎曼流形均可以由欧氏空间的一个子集来实现，当然该欧氏空间的维度可能是高维的，这时流形称为嵌入（“Embedding”），这可以通过欧氏空间的结构来引出黎曼度量。然而这样的嵌入却并非是唯一的，我们将在随后看到，两个同一黎曼度量的等效实现也是可能的。

4.4.2 标量及矢量场 (Scalar and Vector Fields)

由于我们关心定义在域 Ω 上的信号，我们需要说明一下流形上的标量场以及矢量场的函数。一个（光滑）标量场是一个函数： $x:\Omega \rightarrow \mathbb{R}$ 。标量场形成了一个矢量空间 $\mathcal{X}(\Omega, \mathbb{R})$ 附带着内积结构：

$$\langle x, y \rangle = \int_{\Omega} x(u) y(u) du \quad (4-28)$$

其中 du 表示黎曼度量引出的体积元。一个（光滑）的切矢量场是一个将每一个流形上点分配一个切矢量的函数： $X:\Omega \rightarrow T\Omega$ ，元素表示为： $u \mapsto X(u) \in T_u\Omega$ 。矢量场也形成了一个矢量空间 $\mathcal{X}(\Omega, T\Omega)$ 及其定义在黎曼度量上的内积结构：

$$\langle X, Y \rangle = \int_{\Omega} g_u(X(u), Y(u)) du \quad (4-29)$$

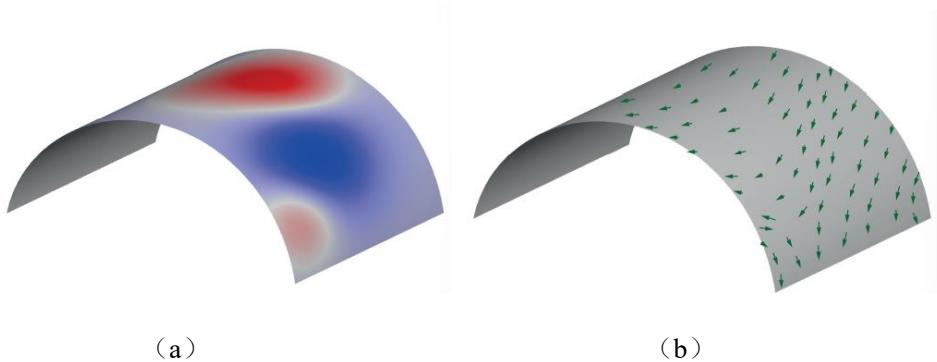


图 4.9 场示例，(a) 表示标量场，(b) 表示矢量场

4.4.3 内参梯度 (Intrinsic Gradients)

另一种考虑（严谨来说即定义）矢量场的方式是将其认为导数的一种一般的形式。在经典的微积分中，人们可以在局部用线性表示一个光滑函数，这可以用微分来表示：
 $dx(u) = x(u + du) - x(u)$ ，这也显示了任意无穷小的扰动 du 对函数 x 在点 u 处的变化。然而在我们的应用中，应用这一朴素的定义方式几乎是不可能的，这是因为缺乏全局的矢量空间结构的基础上 $u + du$ 在流形上通常是无意义的。

解决这一问题的办法就是使用切面空间上的切矢量来模拟局部无穷小的位移。给定一个光滑标量场 $x \in \mathcal{X}(\Omega, \mathbb{R})$ ，我们可以把矢量场看成是一个线性映射

$Y: \mathcal{X}(\Omega, \mathbb{R}) \rightarrow \mathcal{X}(\Omega, \mathbb{R})$ 并满足微分性质：(1) 对任意的常数函数 c 有 $Y(c) = 0$ ；(2) 线性性： $Y(x + z) = Y(x) + Y(z)$ ；(3) 莱布尼兹法则：

$Y(xz) = Y(x)z + xY(z)$, $x, z \in \mathcal{X}(\Omega, \mathbb{R})$ 。可以证明的是，我们可以用这些性质去定义矢量场。微分 $dx(Y) = Y(x)$ 也可以视为一种算子： $(u, Y) \mapsto Y(x)$ ，并可用如下来解释：切矢量 $Y \in T_u \Omega$ 的微小位移引起的 x 的变化可以表示为 $d_u x(Y)$ ，这也是经典的方向导数概念的推广。值得注意的是我们并没有使用黎曼度量，并且这一概念也可以推广到更一般的构建丛中，我们将在 4.5 节中详细介绍。

相应的替代选择是，在每一个点 u 上，微分可以视为一个线性的泛函 $dx_u: T_u \Omega \rightarrow \mathbb{R}$ ，其输入就是切矢量 $X \in T_u \Omega$ 。定义在矢量空间上的线性泛函也称为对偶矢量 (Dual Vector) 或者协矢量 (Covector)。若我们额外地给定内积结构 (即黎曼度量)，则对偶矢量总是可以表示成：

$$dx_u(X) = g_u(\nabla x(u), X) \quad (4-30)$$

这一表示是由 Riesz- Fréchet 表示定理导出，其中任意的对偶矢量均可以表示成原矢量的内积。

在不同的点 u 上的不同微分表示是一个切矢量 $\nabla x(u) \in T_u \Omega$ ，称之为 x 的内参梯度。这与经典微积分中梯度概念类似，其可以视为 x 增长最陡的方向。梯度可以视为一种算子：

$\nabla: \mathcal{X}(\Omega, \mathbb{R}) \rightarrow \mathcal{X}(\Omega, T\Omega)$ ，该算子在任一点 u 的矢量场 (或标量场) $x(u)$ 处变换为：
 $x(u) \mapsto \nabla x(u) \in T_u \Omega$ ，因此一个标量场的梯度就是一个矢量场 ∇x 。

4.4.4 测地线 (Geodesics)

考虑一个定义在流形上的光滑曲线 $\gamma: [0, T] \rightarrow \Omega$ ，其端点为 $u = \gamma(0), v = \gamma(T)$ 。该曲线在点 t 对应的导数是 $\gamma'(t) \in T_{\gamma(t)} \Omega$ ，称为速度矢量。在所有连接 u 与 v 的曲线上的点中，我们关心那些能够得到最小路径的点集，也就是说我们希望找到一个曲线 γ 使其是最小路径的泛函：

$$\ell(\gamma) = \int_0^T \|\gamma'(t)\|_{\gamma(t)} dt = \int_0^T g_{\gamma(t)}^{1/2}(\gamma'(t), \gamma'(t)) dt \quad (4-31)$$

最小化式 (4-31) 的曲线称为测地线 (希腊语发展而来, 意为 “Division of Earth”), 它们在微分几何中起着重要的作用。对我们的讨论非常重要的是, 我们定义的测地线是内参的, 因为其通过计算泛函的模长仅仅依赖于黎曼度量。

熟悉微分几何的读者可能已经回想起了测地线实际上是一个非常一般的概念, 他们的定义实际上也可以不依赖于黎曼度量, 而是依赖联络 (Connection), 也被称为协变导数 (Covariant Derivative), 其概念可扩充到矢量场及张量场, 其一般的定义也是不言自明的, 类似于我们对于微分的构建。给定一个黎曼度量, 存在一个唯一的联络, 称为 Levi-Civita 联络, 这一特殊联络在黎曼几何中也被视为默认选项。起源于这一联络的测地线就是我们前边讨论的最小模长的曲线。Levi-Civita 联络是一种无扭曲并且完备的, 黎曼几何的基本定理保证了其存在性及唯一性。

我们将在下面展示如何使用测地线来定义一种在流形上移动切矢量的方法, 即平行移动 (Parallel Transport), 这构建了从流形到切矢量空间的局部内参映射, 也成为指数映射 (Exponential Map), 也定义了流形上距离, 也称为测地线度量。这允许我们在切矢量空间上通过应用一个局部滤波器来构建类似卷积的操作。

4.4.5 平行移动 (Parallel Transport)

我们已经遇到了在处理流形问题时面临的一个难题: 我们无法直接对流形上的点直接应用加减法。在比较不同点处的切矢量时也会遇到同样的问题: 尽管这些切矢量属于同一维度, 他们处于不同的空间中, 也就是 $X \in T_u \Omega$ 及 $Y \in T_v \Omega$, 因而其无法直接进行比较。测地线提供了将矢量从一点移动到另一点的机制, 方法为: 假定 γ 表示测地线上点构成的曲线, 满足 $u = \gamma(0), v = \gamma(T)$, 并且假定 $X \in T_u \Omega$, 我们可以沿着测地线定义一个新的切矢量集合: $X(t) \in T_{\gamma(t)} \Omega$, 其模长以及与曲线切线的夹角、曲线的速度矢量均为常量, 也即:

$$g_{\gamma(t)}(X(t), \gamma'(t)) = g_u(X, \gamma'(0)) = \text{const}, \quad \|X(t)\|_{\gamma(t)} = \|X\|_u = \text{const} \quad (4-32)$$

我们可以在点 v 处得到一个点 u 处切矢量移动过来的唯一的矢量 $X(T) \in T_v \Omega$ 。

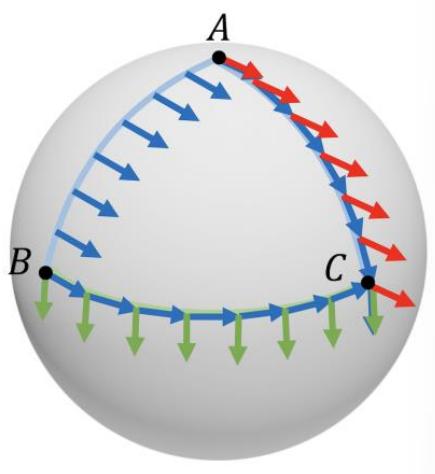


图 4.10 欧式空间下的移动对于流形而言是无意义的，因为移动后矢量可能不再是切矢量（图中红色），因而从 A 到 C 的平行移动随着路径转动，其始终是切矢量。注意其平行移动是路径相关的，从 A 直接到 C 与从 A 先到 B 再到 C 是不同的。

应用上述的记法定义映射 $\Gamma_{u \rightarrow v}(X): T_u \Omega \rightarrow T_v \Omega$ ，且满足 $\Gamma_{u \rightarrow v}(X) = X(T)$ ，该映射即为平行移动，或称为联络，后一种叫法暗示了其是一种连接不同的切矢量空间的机制。由于模长及转角保持的条件，平行移动仅仅对矢量进行旋转，因而其可以视为是一个特殊正交群 $SO(3)$ 的元素，该元素也因此被称为切丛结构群，我们使用 $\mathfrak{g}_{u \rightarrow v}$ 来表示联络，并将在 4.5 节中深入分析。

正如我们前述提到的，一个联络可以被公理化地定义为不依赖黎曼度量的形式，这给我们了一种任意光滑曲线上抽象平行移动的概念。然而这种移动依赖于路径。

4.4.6 指数映射 (Exponential Map)

在一点 u 的一个邻域中，我们总可能在给定一个方向 $X \in T_u \Omega$ 时定义一个唯一的测地线，满足 $\gamma(0) = u$ 以及 $\gamma'(0) = X$ 。当我们可以在所有的 $t \geq 0$ 定义这样的曲线 $\gamma_x(t)$ 时，流形被称为测地线完备 (Geodesically Complete)，这时指数映射定义在整个其空间上。由于紧致流形总是测地线完备的，因此我们可以默认这一方便的特性总是成立的。

给定点以及方向的测地线给出了一个由切空间到流形空间本身的映射，称为指数映射： $\exp: B_r(0) \subset T_u \Omega \rightarrow \Omega$ ，其通过沿着切矢量的一个单位矢量来定义，即： $\exp_u(X) = \gamma_x(1)$ 。指数映射 \exp_u 是一个局部微分同胚，由于其将切空间上原点附近邻域 $B_r(0)$ 映射到点 u 的邻域。相反，人们也可以将指数映射看成一种流形上内参局部变形到切空间中。

注意测地线完备性并不能保证指数映射 \exp 是全局微分同胚的，点 u 处指数映射 $\exp_u(B_r(0) \subseteq T_u \Omega)$ 能够微分同胚地映射的最大的半径 r 称为单射半径 (Injectivity Radius)。

4.4.7 测地线距离 (Geodesic Distance)

Hopf-Rinow 定理保证了测地线完备的流形也是完备度量空间，该定理建立了测地线与度量完备性之间的等价关系，度量完备意味着任意的柯西序列均在测地线距离度量下收敛。在此流形上我们可以定义一个距离（称为测地线距离或者测地线度量）为点之间的总是存在的最短路径：

$$d_g(u, v) = \min_{\gamma} \ell(\gamma), \text{ s.t. } \gamma(0) = u, \gamma(T) = v \quad (4-33)$$

值得注意的是，术语 u “度量” (“Metric”) 在本书有两层意思：(1) 黎曼度量；(2) 距离。为了不至于引起歧义，我们使用距离指代后一种意思，符号 d_g 表示依赖于黎曼度量的距离，这是通过测地线长度来定义的。

4.4.8 等距同构 (Isometries)

现在考虑我们的流形 Ω 变形到另一个有着黎曼度量 h 的流形 $\tilde{\Omega}$ 上，我们假定这种变形是微分同胚的： $\eta: (\Omega, g) \rightarrow (\tilde{\Omega}, h)$ 。其微分： $d\eta: T\Omega \rightarrow T\tilde{\Omega}$ 则相应地定义了两个切面丛的映射（称为前推，Pushforward），使得对于任意的点 u ，我们得到 $d\eta_u: T_u\Omega \rightarrow T_{\eta(u)}\tilde{\Omega}$ ，这可以如下解释：若我们沿着点 u 上切矢量 $X \in T_u\Omega$ 给一个小的位移，映射效果为将沿着切矢量 $d\eta_u(X) \in T_{\eta(u)}\tilde{\Omega}$ 位移 $\eta(u)$ 。

由于前推提供了两个流形上的切矢量空间的一种关联机制，其也支持从黎曼度量 h 的域 $\tilde{\Omega}$ 到黎曼度量为 g 的域 Ω 的回拉 (Pullback)：

$$(\eta^* h)_u(X, Y) = h_{\eta(u)}(d\eta_u(X), d\eta_u(Y)) \quad (4-34)$$

前推与回拉是一对自伴算子，即对对偶空间的矢量 $\alpha \in T^*\Omega$ ，有 $\langle \eta^* \alpha, X \rangle = \langle \alpha, \eta_* X \rangle$ 对偶矢量作为一种线性映射定义在每一切空间的点上，内积分别定义在矢量场与对偶矢量场。

若对于域 Ω 上的每一点，回拉度量均与黎曼度量已知，即 $g = \eta^* h$ ，则这样的映射 η 被称为（黎曼）等距同构。对于二维流形而言，等距同构可以直觉地理解为流形变形是没有拉伸或者剪切的非弹性变形。

根据其定义，等距同构能够保持内参结构不发生变化，例如完全由黎曼度量导出的测地线距离。因而我们可以从度量集合的角度去理解等距同构，即其是度量空间之间的距离保持的映射， $\eta: (\Omega, d_g) \rightarrow (\tilde{\Omega}, d_h)$ ，满足条件：

$$d_g(u, v) = d_h(\eta(u), \eta(v)) \quad (4-35)$$

上式对于任意的 $u, v \in \Omega$ 均成立，或者更严谨一点 $d_g = d_h \circ (\eta \times \eta)$ 。换言之，黎曼等距同构也是度量等距同构。在联通的流形中，逆命题也是正确的：任意的度量等距同构也是黎曼等距同构。这一结论由 Myers-Steenrod 定理给出，本书中我们默认流形是联通的。

在我们的几何深度学习蓝图中, η 表示域的变形的映射。当 η 是一个等距同构时, 所有的内参量均不会在 η 作用前后发生变化。我们也可以通过度量扩张 (Metric Dilation) 的概念来扩展度量等距同构:

$$\text{dil}(\eta) = \sup_{u \neq v \in \Omega} \frac{d_h(\eta(u), \eta(v))}{d_g(u, v)} \quad (4-36)$$

或用度量畸变 (Metric Distortion) 的概念来扩展:

$$\text{dis}(\eta) = \sup_{u, v \in \Omega} |d_h(\eta(u) - \eta(v)) - d_g(u, v)| \quad (4-37)$$

度量畸变定义式由 Gromov-Hausdorff 距离给出, 我们在 3.2 节中提及过, 其可以视为可能的最小度量距离。两个式子分别计算了测地线距离在变形 η 作用下时的相对亦即绝对变化量。式 (3-11) 要求的对于任意的函数 $f \in \mathcal{F}(\mathcal{X}(\Omega))$ 在域变形情况下稳定性的要求可以表示为:

$$\|f(x, \Omega) - f(x \circ \eta^{-1}, \tilde{\Omega})\| \leq C \|x\| \text{dis}(\eta) \quad (4-38)$$

4.4.9 内参对称 (Intrinsic Symmetries)

上述变形的一个特殊的例子就是域对自己的微分同胚映射 (我们在 3.2 节中称之为自同构映射, Automorphism), 用 $\tau \in \text{Diff}(\Omega)$ 表示这样的映射。若其回拉度量满足 $\tau^* g = g$ 时我们称其为黎曼 (自) 等距同构, 满足 $d_g = d_g \circ (\tau \times \tau)$ 时我们称其为度量 (自) 等距同构。意料之中的是, 等距同构形成了一个以组合算子为作用算子的群, 该算子用 $\text{Iso}(\Omega)$ 表示, 该群称为等距同构群, 群中幺元就是映射 $\tau(u) = u$, 根据映射 τ 为微分同胚映射, 其逆也总是存在。流形上的连续对称群是由无穷小的切矢量场生成的, 这样的矢量场称为 Killing 场 (Killing Fields), 这一命名由 Wilhelm Killing 给出。自等距同构也是流形的内参对称。

4.4.10 流形傅里叶分析 (Fourier Analysis on Manifolds)

我们现在介绍如何在流形上构造类似卷积的操作, 这样的操作也是对等距同构变形保持不变性的。为实现保持不变性这一点, 我们有两种选择: (1) 类比傅里叶变换的方法在傅里叶频域定义内积; (2) 通过在局部将滤波器与信号进行某种关联定义空域卷积。我们首先研究谱域方法。

欧氏空间上的傅里叶变换是由循环矩阵的特征向量得到, 利用了循环矩阵满足交换律并可联合对角化的特点。因此, 任意的循环矩阵, 或者特殊地, 微分算子, 均可以作为傅里叶变换的类比定义在一般的域上。在黎曼几何中, 最长使用的是拉普拉斯算子的正交基底, 我们将在此对该算子及基底进行介绍。

为说明这些, 回想一下我们的内参梯度算子: $\nabla: \mathcal{X}(\Omega, \mathbb{R}) \rightarrow \mathcal{X}(\Omega, T\Omega)$, 其能够在流

形域上一点得到指向最陡变化方向的标量场的一个矢量场。类似地，我们可以定义一个散度算子 $\nabla^*: \mathcal{X}(\Omega, T\Omega) \rightarrow \mathcal{X}(\Omega, \mathbb{R})$ 。若我们将矢量场假想为流形上一种流动，散度衡量了一个场在某一点的净流量，这使得我们得以区分哪里是流动的源泉，哪里是流动的终点。我们使用 ∇^* 的记法（没有采用常用的 div ）来强调这一算子与梯度算子是自伴的：

$$\langle X, \nabla x \rangle = \langle \nabla^* X, x \rangle \quad (4-39)$$

始终我们使用了标量与矢量之间的内积来表示。

拉普拉斯算子（Laplacian Operator），有时也被称为拉普拉斯-贝尔特拉米算子（Laplacian-Beltrami Operator）是定义在域 $\mathcal{X}(\Omega)$ 上的一种算子，定义式为 $\Delta = \nabla^* \nabla$ ，其可以视为是一点的无穷小的球域中的函数平均值与该点本身的函数值的差值。这一算子在数学物理中是一种非常重要的算子，可以用来描述诸如热扩散、量子震动、波传播等诸多现象。对我们而言重要的是，拉式算子是内参的，因此其能够在等距同构作用下保持不变。拉式算子是各向同性的，这一点我们将在 4.6 节中进行介绍，我们也可以定义各向异性的拉式算子： $\nabla^*(A(u))\nabla$ ，其中 $A(u)$ 表示与位置有关的张量，其决定了局部的方向 [Andreux et al., 2014]、[Boscaini et al., 2016b]。

显然拉式算子是自伴算子（也是实对称矩阵）：

$$\langle \nabla x, \nabla x \rangle = \langle x, \Delta x \rangle = \langle \Delta x, x \rangle \quad (4-40)$$

上式的左侧的平方形式也就是狄利赫雷能量：

$$c^2(x) = \|\nabla x\|^2 = \langle \nabla x, \nabla x \rangle = \int_{\Omega} \|\nabla x(u)\|_u^2 du = \int_{\Omega} g_u(\nabla x(u), \nabla x(u)) du \quad (4-41)$$

狄利赫雷能量衡量了函数 x 的平滑程度。

拉式算子可以进行特征分解：

$$\Delta \varphi_k = \lambda_k \varphi_k, \quad k = 0, 1, \dots \quad (4-42)$$

当流形是紧致（我们在本书中默认这一点）时，其谱域是可列集。由于其为自伴算子，拉式算子特征方程为正交的，其正交特征方程可表示为： $\langle \varphi_k, \varphi_l \rangle = \delta_{kl}$ 。其正交基底也可以构建成狄利赫雷能量最小化的正交基：

$$\varphi_{k+1} = \arg \min_{\varphi} \|\nabla \varphi\|^2 \quad s.t. \quad \|\varphi\| = 1, \langle \varphi, \varphi_j \rangle = 0 \quad (4-43)$$

上式对于任意的 $j = 0, 1, \dots, k$ 均成立，由此可将拉式算子特征方程视为最平滑的正交基底。特征方程 $\varphi_0, \varphi_1, \dots$ 及对应的特征值 $0 = \lambda_0 \leq \lambda_1 \leq \dots$ 可以视为经典傅里叶变换中的原子及频率的类比。实际上 $e^{i\xi u}$ 也是欧氏空间拉式算子的特征方程，这与傅里叶变换一致。

正交基底可以使得任意平方可积函数分解为傅里叶基底：

$$x(u) = \sum_{k \geq 0} \langle x, \varphi_k \rangle \varphi_k(u) \quad (4-44)$$

其中常使用 $\hat{x}_k = \langle x, \varphi_k \rangle$ 来代替内积，该项被称为函数 x 的傅里叶系数或者傅里叶变换。截断傅里叶序列可以得到一个带有误差的原函数 x 的逼近表示，其与原函数的差距可以表示为 [Aflalo and Kimmel, 2013]：

$$\left\| x - \sum_{k=0}^N \langle x, \varphi_k \rangle \varphi_k \right\|^2 \leq \frac{\|\nabla x\|^2}{\lambda_{N+1}} \quad (4-45)$$

[Aflalo et al., 2015] 进一步展示了没有其他基底能过够得到更低的误差, 这也使得拉式基底对于流形上平滑信号是最优的表示。

4.4.11 流形谱域卷积 (Spectral Convolution on Manifolds)

谱域卷积可以定义成傅里叶谱域中信号与滤波器的乘积, 也即:

$$(x \star \theta)(u) = \sum_{k \geq 0} (\hat{x}_k \cdot \hat{\theta}_k) \varphi_k(u) \quad (4-46)$$

注意我们在此处应用了经典傅里叶变换的性质 (亦即卷积定理) 来定义非欧空间上的卷积。由于其构建定义, 这样的谱域卷积是内参的, 因而也是等距同构不变的。此外由于拉式算子是各向同性的, 它无法区分方向, 这与我们在 4.1 节中遇到的图上由于邻域的排序变换不变性情况类似。

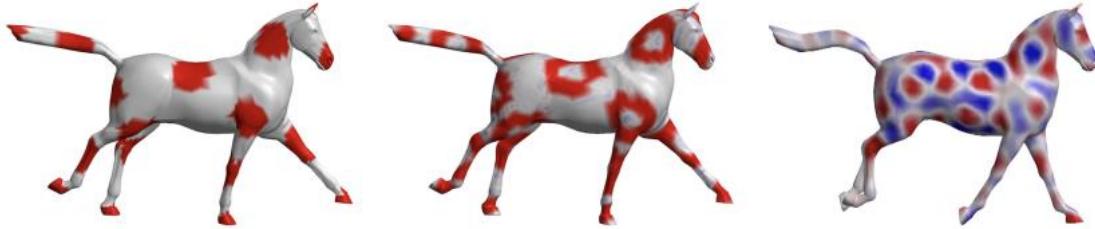


图 4.11 谱域滤波器在域变化时的不稳定性。左: 一个作用在网格模型上的信号; 中: 网格模型域上的谱域滤波后的拉式特征基底; 右: 同样的滤波器应用到准等距同构变换后的域上后的拉式算子的特征向量。可以看出轻微的等距同构变换将使得变换前后滤波信号有较大不同。

在实际中, 直接计算式 (4-46) 是计算开销巨大的, 这是由于需要对角化拉式算子矩阵。更糟的是, 其呈现出几何上的不稳定性: 在等距同构变换的作用下, 域的变形使得拉式算子的高频分量对应的特征方程将发生巨大变动 (图 4.11)。一个更加稳定的计算方式是使用一个形式为 $\hat{p}(\Delta)$ 的谱域传递函数作为滤波器:

$$\begin{aligned} (\hat{p}(\Delta)x)(u) &= \sum_{k \geq 0} \hat{p}(\lambda_k) \langle x, \varphi_k \rangle \varphi_k(u) \\ &= \int_{\Omega} x(v) \sum_{k \geq 0} p(\lambda_k) \varphi_k(v) \varphi_k(u) dv \end{aligned} \quad (4-47)$$

上式可以从两方面来解读: (1) 作为一个谱域滤波器, 这正是第一个等式的内容, 此时 $\hat{\theta}_k = \hat{p}(\lambda_k)$; (2) 作为一个空域滤波器, 此时具有一个位置相关的核 $\theta(u, v) = \sum_{k \geq 0} \hat{p}(\lambda_k) \varphi_k(v) \varphi_k(u)$ 。式 (4-47) 的优势在于, $\hat{p}(\lambda)$ 可以表示成数个系数的参数化形式, 例如可以选用多项式参数方程 $\hat{p}(\lambda) = \sum_{l=0}^r \alpha_l \lambda^l$, 选择这样的多项式表示可以使计算滤波变得高效:

$$(\hat{p}(\Delta)x)(u) = \sum_{k \geq 0} \sum_{l=0}^r \alpha_l \lambda_k^l \langle x, \varphi_k \rangle \varphi_k(u) = \sum_{l=0}^r \alpha_l (\Delta^l x)(u) \quad (4-48)$$

这样的计算避免了谱域分解。我们将在 4.6 节中介绍这一公式的构建。基于谱域卷积方法的几何深度学习通常表示为傅里叶变换的形式并于空域方法相对应，我们在这里看到两种方法实际上在某种程度上的等价的，因此将两种方法视为截然不同的方法也许是不合适的。

4.4.12 流形空域卷积 (Spatial Convolution on Manifolds)

定义流形上的卷积的另一种选择是在不同点上匹配一个滤波器，正如式 (4-22) 中定义的：

$$(x * \theta)(u) = \int_{T_u \Omega} x(\exp_u Y) \theta_u(Y) dY \quad (4-49)$$

但是我们在式 (4-49) 中使用指数映射来从切空间映射到标量场并获得标量场的数据，滤波器 θ_u 是定义在切空间中，因此依赖于点的位置（由于切平面依赖于点的位置，译者注）。如果我们定义的滤波器是内参的，那么对应的卷积也将是等距同构不变的，这一性质在众多的计算机视觉及计算机图形学应用中有着重要且广泛的应用。

然而我们需要指出与 4.2-4.3 节中的构造的一些重要的区别。首先，由于流形通常并非齐次空间，我们没有一个全局的群结构，使得我们没有一个共享的滤波器（也就是说滤波器参数将依赖于位置，如式 (4-49) 中定义的），点的位置在流形上移动时，滤波器也将发生变化。这一流形上依赖于位置的卷积运算操作需要平移移动，这使得我们可以在其他的切空间 $T_v \Omega$ 上应用共享的滤波器，并将其定义为切空间上 $T_u \Omega$ 的函数。其次，然而正如我们曾遇到过的，这将依赖于点 u 与点 v 之间的路径，因此我们移动滤波器的方式也是至关重要的（也就是沿着哪一条曲线移动滤波器，译者注）。第三，由于我们只能在局部使用指数映射，因此滤波器也将是局部的，局部的范围就是单射半径范围。第四，也是最重要的是，由于切矢量 X 是一个抽象的对象，我们无法处理 $\theta(X)$ ，为使得切矢量能够应用于计算当中，我们必须将其写成某个相对坐标的形式： $\omega_u : \mathbb{R}^s \rightarrow T_u \Omega$ ，即表示成一个 s 维坐标系的坐标 $\mathbf{x} = \omega_u^{-1}(X)$ ，应用这一相对坐标记法，我们得到：

$$(x * \theta)(u) = \int_{[0,1]^s} x(\exp_u(\omega_u \mathbf{y})) \theta(\mathbf{y}) d\mathbf{y} \quad (4-50)$$

此时滤波器定义在了单位立方体上。由于指数映射是内参的（由于通过测地线导出），所以得到的卷积是等距同构不变的。

不过这也默认了我们可以在另一个流形上定义坐标系 ω_u ，也就是 $\omega'_u = d\eta_u \circ \omega_u$ ，该坐标系物理学中也称为度规 (Gauge)。如何在给定的一个流形上以一种统一的方式获得这样一个坐标系却是十分困难的。首先，一个平滑的全局度规可能压根不存在：这种情况对应

于非平行的流形，在这样的流形上我们无法定义一个光滑非消失的切矢量场；第二，我们也没有一个流形上标准的度规，因为选择就是任意的，但是我们的卷积又依赖于相对坐标系 ω ，因此当选取另一个相对坐标系时，结果将大不一样。球面 S^2 就是一个非平行的流形的例子，这是 Poincare-Hopf 定理所证明的。

我们应当注意到这就是理论与实践出现分歧的地方：在实践中，我们可以构造最为光滑的坐标系，当然可能存在少数奇点，例如可以流形上一些内参标量场的内参梯度作为参考坐标系。此外，这样的构建也是稳定的，也就是说这样的构建在等距同构的流形间将能够保持一致，在准等距同构的流形间也即为相似。这些方法在早期的流形研究的深度学习方法中得到应用 [Masci et al., 2015]、[Monti et al., 2017]。

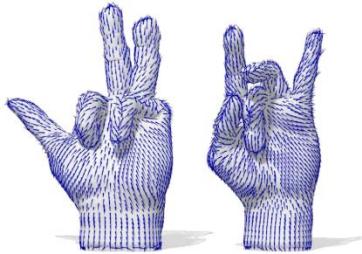


图 4.12 准等距同构的流形间稳定度规的示例，应用了 GFrames 算法 [Melzi et al., 2019]

然而，在奇点附近的结果并不能让人满意，滤波器方向（相对于度规的固定方向）将发生极大变化，导致即使在输入信号亦即滤波器均为光滑的情况下特征映射也是不光滑的。此外，给定一个点 u 的一个方向为什么应当视为与另一个点 v 对应的方向等同仍然不清楚，因此，抛开实践应用中的方法，我们将在下面深入分析在理论上完善的方法，该方法也将独立于度规的选择。

4.5 度规 (Gauges) 与丛 (Bundles)

正如我们定义的度规是作为切空间上的坐标系，度规的概念在物理中是一个更加一般的概念。从历史来看，纤维丛 (Fibre Bundles) 是在 Élie Cartan 的现代微分几何中提出的，但其并没有明确地定义他们，这一概念在 1930 年左右的拓扑学研究中作为独立的研究对象进一步被研究扩展了。度规可以指代任意的矢量丛的坐标系，并不仅仅指代切矢量丛。非正式地说，一个矢量丛描述了一类被另一个矢量空间参数化的矢量空间，即矢量丛包含有基矢量空间 Ω 以及在每一点 $u \in \Omega$ 出一个一模一样的矢量空间 \mathbb{V} （也成为纤维，即 Fibre）。对于切丛而言矢量空间 \mathbb{V} 就是切空间 $T_u\Omega$ 。粗略地讲，一个丛可以视为点 u 局部邻域上的积 $\Omega \times \mathbb{V}$ ，但是从全局来看，其一定会扭曲并因此具有完全不同的结构。在几何深度学习中，从用来描述流形上点 u 出的特征空间，其维度与该点的特征通道数相同。在这种语义下，一种新的并且令人激动的对称性，称为度规对称，就显现出来了。

让我们考虑一个 s 维的流形以及其切丛 $T\Omega$ ，一个矢量场 $X:\Omega \rightarrow T\Omega$ （在这种术语下

指代切丛上的一个片段 (Section))。相对于应用于切丛的度规 ω , 切矢量可以表为一个函数: $\mathbf{x}: \Omega \rightarrow \mathbb{R}^s$ 。然而值得注意的是, 我们真正关心的是蕴含的几何对象 (即矢量场), 其表示为一个依赖于度规选择的函数 $\mathbf{x} \in \mathcal{X}(\Omega, \mathbb{R}^s)$, 如果我们改变了度规, 我们也需要改变 \mathbf{x} 以使得蕴含的矢量场能够正确地表示出来。

4.5.1 切丛及结构群 (Tangent Bundles and the Structure Group)

当我们改变度规时, 我们需要应用一个可逆矩阵将每一点的旧的度规变换到新的度规上, 这个可逆矩阵对于在每一点的一对新旧度规而言都是唯一的, 但是在不同的点可能不一样。换言之, 度规变换是一个映射 $\mathbf{g}: \Omega \rightarrow \mathrm{GL}(s)$, 其中 $\mathrm{GL}(s)$ 表示一般线性群, 其为一个 $s \times s$ 的可逆矩阵。该映射作用于度规 $\omega_u: \mathbb{R}^s \rightarrow T_u \Omega$ 从而能够得到新的度规 $\omega'_u = \omega_u \circ \mathbf{g}_u: \mathbb{R}^s \rightarrow T_u \Omega$ 。度规变换通过 $\mathbf{x}'(u) = \mathbf{g}_u^{-1} \mathbf{x}(u)$ 作用在每一点的坐标系下的矢量场以生成相对于新的度规的 X 的表示 \mathbf{x}' , 内在的矢量场不发生变化, 仅仅表示发生了变化:

$$X(u) = \omega'_u(\mathbf{x}'(u)) = \omega_u(\mathbf{g}_u \mathbf{g}_u^{-1} \mathbf{x}(u)) = w_u(\mathbf{x}(u)) = X(u) \quad (4-51)$$

这正是我们想要的性质。更一般地, 我们又一个根据一般线性群 $\mathrm{GL}(s)$ 的表示 ρ 进行变换的一个几何量的场, 也就是说, 一个二维张量场 (2-Tensor, 也就是矩阵) $\mathbf{A}(u) \in \mathbb{R}^{s \times s}$ 发生的变换为 $\mathbf{A}'(u) = \rho_2(\mathbf{g}_u^{-1}) \mathbf{A}(u) \rho_1(\mathbf{g}_u)$, 在这样的情况下, 度规变换 \mathbf{g}_u 通过 $\rho(\mathbf{g}_u)$ 进行作用。

有时我们希望关注含有某些性质的坐标系选择, 例如正交的坐标系, 右手坐标系等。意料之中的是, 我们关注那些能够保持某些性质的变换形成的群。例如, 能够保持正交的群为正交群 $\mathrm{O}(s)$ (包含旋转与镜像), 额外地保持着方向或者说左右手性的群为 $\mathrm{SO}(s)$ (仅包含旋转)。因此一般而言我们称一个从带有的群为结构群, 度规变换就是映射 $\mathbf{g}: \Omega \rightarrow \mathfrak{G}$ 。一个关键的观察是, 在所有情况的给定性质中, 给定点之间的两个坐标系仅存在一个与其有关的度规变换。

我们使用 s 代表基空间的维度, d 代表纤维的维度, 对于切丛而言, $d = s$ 正是流形的维度, 对于 RGB 图像而言 $s = 2, d = 3$ 。

正如前面介绍过的, 度规理论可以扩展到切丛以外的领域, 一般地, 我们可以考虑一个矢量空间的丛并且其结构与维度并不一定与基空间的结构及维度有关。例如, 彩色图片像素有着二维网格结构的坐标 $u \in \Omega = \mathbb{Z}^2$ 以及 RGB 空间的 $\mathbf{x}(u) \in \mathbb{R}^3$ 的颜色值, 因而像素的空间可视为附带基空间 \mathbb{Z}^2 以及每一点的纤维 \mathbb{R}^3 的矢量丛。将 RGB 图像表示为相对于一种有着 R, G, B 三个通道的基矢的度规是可以由使用者自行选择的, 因此图像的坐标表示为 $\mathbf{x}(u) = (r(u), g(u), b(u))^\top$ 。但我们可以对不同的颜色通道进行独立地排序, 只要我们自己知道使用的颜色通道的次序即可。在这一例子中, 我么选择了群 $\mathfrak{G} = \Sigma_3$ 来作为变换群, 也可以选择其他类型的群, 例如 Hue 旋转群 $\mathfrak{G} = \mathrm{SO}(2)$ 。从计算的角度看这样做可能

是没什么意义的，但我们马上将看到，这一理论对于思考度规变换概念有着明确地提示概念的作用，因为这允许我们表达度规对称---在这样的情况下即颜色之间的等价性的关联，并且构造作用在图像上的能够遵循这一对称性的函数（对不同的颜色通道独立地处理）。

在流形上矢量场的例子中，一个 RGB 度规变换改变了一个图片的数值表示（这是通过对每个像素点的 RGB 三通道的数据进行重排列变换）但并没有改变图片本身。在机器学习应用中，例如在进行图片分类或者分割的任务中，我们关心构建作用在图片上的函数 $f \in \mathcal{F}(\mathcal{X}(\Omega))$ ，该函数通常部署为神经元的层。其遵循着若我们对图片应用度规变换，为保持其原有的意义，相应的函数 f （也就是神经网络层）也需要随之改变。为简单起见考虑一个 1×1 的卷积，也就是一个将图像中像素点作为输入 $\mathbf{x}(u) \in \mathbb{R}^3$ 将特征矢量 $\mathbf{y} \in \mathbb{R}^C$ 作为输出的映射函数。根据我们的几何深度学习蓝图，输出关联与一个群表示 ρ_{out} ，在本例中就是一个结构群 $\mathfrak{G} = \Sigma_3$ （即颜色通道排列变换）的 C 维的表示，类似地，输入与 $\rho_{in}(\mathfrak{g}) = \mathfrak{g}$ 的表示关联。然后，如果我们对输入作用一个度规变换，我们需要改变线性映射 (1×1 卷积): $f: \mathbb{R}^3 \rightarrow \mathbb{R}^C$ 到 $f' = \rho_{out}^{-1}(\mathfrak{g}) \circ f \circ \rho_{in}(\mathfrak{g})$ 以使得输出的特征矢量 $\mathbf{y}'(u) = f'(\mathbf{x}(u))$ 变换为 $\mathbf{y}'(u) = \rho_{out}(\mathfrak{g}_u)\mathbf{y}(u)$ 。实际上我们可以验证：

$$\mathbf{y}' = f'(\mathbf{x}') = \rho_{out}^{-1}(\mathfrak{g})f(\rho_{in}(\mathfrak{g})\rho_{in}^{-1}(\mathfrak{g})\mathbf{x}) = \rho_{out}^{-1}(\mathfrak{g})f(\mathbf{x}) \quad (4-52)$$

其中 $\rho^{-1}(\mathfrak{g})$ 指代群的逆表示。

4.5.2 度规对称 (Gauge Symmetries)

将度规变换视为一种对称意味着我们认为由度规变换关联的两个度规是等价的。例如，若我们考虑群 $\mathfrak{G} = SO(d)$ ，则任意两个右手正交坐标系都可以视为等价，因为我们总可以把一个右手正交坐标系经过旋转变换到另一个右手正交坐标系。换言之，局部并不存在明显的方向，例如上方或者右方（因为旋转操作可以改变方向，译者注）。类似地，如果选择 $\mathfrak{G} = O(d)$ 则任意左手或者右手的正交坐标系都可以视为等价的。在这样的情况下，也没有一个有限的方向。总之，我们可以考虑一个群以及每一点处坐标系构成的集合，满足对于任意的两个坐标系，存在唯一的群作用 $\mathfrak{g}(u) \in \mathfrak{G}$ 将一个坐标系变换到另一个坐标系。

在我们的几何深度学习蓝图中间度规变换视为对称，我们考虑构建作用在域 Ω 中信号上的函数 f 使之对度规变换能够具有等变的效果。具体来说，如果我们对输入进行了度规变化，则输出应该产生对应的变换（也许是经过不同的群作用）。我们之前注意到当我们改变度规时，函数 f 也应当发生变化，但对于度规等变的映射而言并非如此：改变度规并不改变映射。为观察这一点，考虑 RGB 颜色空间的例子，定义映射 $f: \mathbb{R}^3 \rightarrow \mathbb{R}^C$ 满足条件：
 $f \circ \rho_{in}(\mathfrak{g}) = \rho_{out}(\mathfrak{g}) \circ f$ 时称为等变映射，在这种情况下对函数 f 施加度规变换将不起作用： $\rho_{out}^{-1}(\mathfrak{g}) \circ f \circ \rho_{in}(\mathfrak{g}) = f$ 。换言之，用度规坐标表示的等变映射将不依赖于度规的选择，这与图上的示例一样，无论图中输入节点怎么排序变换，我们都应用同样的函数。然

而, 与图的例子以及前述的其他例子不同的是, 度规变换并非作用于 Ω , 而是通过变换 $\mathbf{g}(u) \in \mathfrak{G}$, $u \in \Omega$ 作用于每一个特征矢量 $\mathbf{x}(u)$ 。

进一步考虑当我们研究具有较大空域支撑的流形上的滤波器情况。首先考虑一个较为简单的映射例子 $f: \mathcal{X}(\Omega, \mathbb{R}) \rightarrow \mathcal{X}(\Omega, \mathbb{R})$, 其将 s 维流形 Ω 上的一个标量场映射到另一个标量场。与矢量或者其他几何量不同的是, 标量没有方向, 因而一个标量场 $x \in \mathcal{X}(\Omega, \mathbb{R})$ 对于度规变换是不变的 (或者其根据么元进行变换)。因此, 任意的标量场到标量场的映射均是度规等变的 (或者不变的, 本例中相等同)。例如我们可以写成 f 类似式 (4-47) 的形式, 类似一个卷积的算子, 与位置依赖的滤波器 $\theta: \Omega \times \Omega \rightarrow \mathbb{R}$ 进行作用:

$$(x \star \theta)(u) = \int_{\Omega} \theta(u, v) x(v) dv \quad (4-53)$$

上式也暗示了对于不同的点将有不同的滤波器 $\theta_u = \theta(u, \cdot)$, 也就是没有空间权重共享, 这也是度规对称本身并不保证的。

现在考虑一种更加有趣的映射 $f: \mathcal{X}(\Omega, T\Omega) \rightarrow \mathcal{X}(\Omega, T\Omega)$, 即从矢量场到矢量场的映射。相对于度规而言, 输入输出矢量场 $X, Y \in \mathcal{X}(\Omega, T\Omega)$ 是矢量表示的函数 $\mathbf{x}, \mathbf{y} \in \mathcal{X}(\Omega, \mathbb{R}^s)$ 。这样的矢量函数之间的一般线性映射可以写为式 (4-53) 一样的形式, 尝试通过修改式 (4-53) 的方式定义卷积运算。把核函数 θ 改为矩阵形式 $\Theta: \Omega \times \Omega \rightarrow \mathbb{R}^{s \times s}$ 。矩阵 $\Theta(u, v)$ 是将切矢量 $T_v \Omega$ 映射到切矢量 $T_u \Omega$ 上, 但是这两点有着不同的度规, 这两处的度规可能任意的或独立地改变。也就是说, 对于所有的 $u, v \in \Omega$, 滤波器应该满足 $\Theta(u, v) = \rho^{-1}(\mathbf{g}(u)) \Theta(u, v) \rho(\mathbf{g}(v))$, 其中 ρ 表示对于矢量的群作用, 也就是一个 $s \times s$ 旋转矩阵。由于我们可以自由地选择 $\mathbf{g}(u)$ 以及 $\mathbf{g}(v)$, 这实际上对于滤波器而言是一个非常强的约束。此时要求 Θ 为 0。

一种更好的方法是首先使用联络将矢量移动到共同的切矢量空间中, 然后应用对于某一个点的单个度规变换的度规等变。我们不再使用式 (4-53), 而是定义如下的矢量场间的映射:

$$(\mathbf{x} \star \Theta)(u) = \int_{\Omega} \Theta(u, v) \rho(\mathbf{g}_{v \rightarrow u}) \mathbf{x}(v) dv \quad (4-54)$$

其中 $\mathbf{g}_{v \rightarrow u} \in \mathfrak{G}$ 指代从点 v 到点 u 沿着测地线联络的平行移动。其表示 $\rho(\mathbf{g}_{v \rightarrow u})$ 是一个 $s \times s$ 的旋转方阵, 其将矢量从点之间进行移动。值得注意的是两点之间的测地线认为是唯一的 (实际上这一假定在某些情况下并不成立, 但本书忽视测地线不唯一的情况, 译者注), 这一假定仅仅是局部空间上正确, 因此滤波器也应当具有局部性。在度规变换 \mathbf{g}_u 的作用下, 元素的变换关系为 $\mathbf{g}_{u \rightarrow v} \mapsto \mathbf{g}_u^{-1} \mathbf{g}_{u \rightarrow v} \mathbf{g}_v$, 滤波器本身的变换为 $\mathbf{x}(v) \mapsto \rho(\mathbf{g}_v) \mathbf{x}(v)$ 。如果滤波器与结构群表示满足交换律, 即 $\Theta(u, v) \rho(\mathbf{g}_u) = \rho(\mathbf{g}_u) \Theta(u, v)$, 式 (4-54) 定义了度规等变的卷积, 在前述的变换作用下其变换为:

$$(\mathbf{x}' \star \Theta)(u) = \rho^{-1}(\mathbf{g}_u) (\mathbf{x} \star \Theta)(u) \quad (4-55)$$

4.6 几何图 (Geometric Graph) 与面片模型 (Meshes)

我们以几何图 (Geometric Graphs, 也就是有某种几何空间的图) 以及面片模型 (Meshes) 来总结对于不同的几何域的讨论。在“5G”的几何域中, 面片模型介于图与流形之间: 在很多方面, 面片模型类似一个图, 但是其额外的结构又使得我们可以将其视为连续的物体表面。因此, 在我们的蓝图中认为面片模型并不是一类分立的研究对象, 实际上, 我们将着重强调在对于面片模型的处理的很多方法均可以直接受地应用于图的处理。

正如我们已经在 4.4 节提到过的, 二维流形是常见的对三维对象建模的方法 (或者更好地说, 是对三维对象的边界进行建模的方法)。在计算机视觉及计算机图形学中, 这样的流形表面往往被离散化表示为三角面片, 三角面片可以视为一片一片的平面缝合形成的对原有光滑流形的趋近。因此面片模型就是一类带有特殊结构的图: 除去节点及边外, 面片模型 $\mathcal{T} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ 还有着有序的三个顶点构成的面片

$\mathcal{F} = \{(u, v, q) : u, v, q \in \mathcal{V} \ \&\& \ (u, v) \in \mathcal{E}, (u, q) \in \mathcal{E}, (v, q) \in \mathcal{E}\}$, 顶点的顺序就是面的朝向。三角面片模型也被称为单纯复形 (Simplicial Complexes)。

通常我们还进一步假设每一条边均由 2 个三角形共享, 边界处的边形成单一的回环。这一假设条件约束了顶点的 $1-hop$ 邻居是一个盘型的结构, 因此面片模型形成一个离散的流形, 这样的离散流形也被称为流形面片模型 (Manifolds Meshes)。与黎曼流形类似, 我们可以在离散流形上定义一个度量结构。在最简单的定义中, 度量可以由面片模型的节点的嵌入表示导出, 即通过定义每条边的模长 $\ell = \|\mathbf{x}_u - \mathbf{x}_v\|$ 。这样定义的度量自然就满足了三角不等式, 也就是对于任意的 $(u, v, q) \in \mathcal{F}$ 满足 $\ell_{uv} \leq \ell_{uq} + \ell_{vq}$ 。这一性质仅在模长度量定义为内参时才成立, 任意的面片模型保模长 ℓ 的变形均是等距同构的, 很多读者应当对于等距同构相对熟悉了。

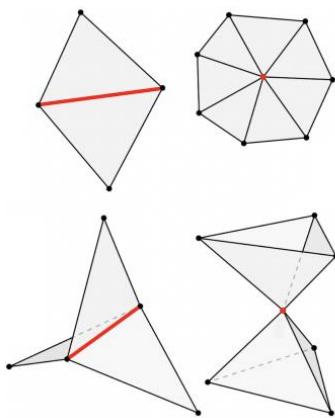


图 4.13 流形面片模型 (上侧) 与非流形面片模型 (下侧)

4.6.1 拉普拉斯矩阵 (Laplacian Matrices)

与对于图的处理类似, 我们假定流形面片模型具有 n 个节点, 每一个节点具有一个 d

维的特征矢量，我们将整个模型的数据写成矩阵的形式，即 $\mathbf{X} = \mathbb{R}^{n \times d}$ 。这一矩阵既可以存储节点的坐标信息，也可以存储额外的特征信息，例如颜色、法矢量等信息，或者存储特定应用需要的信息，例如在化学中几何模型需要存储原子序号（元素周期表中原子序号，译者注）信息。

首先看一下面片模型上的谱域卷积式 (4-46)，其起源于拉式算子。将离散的面片模型视为其代表的连续表面的离散化，我们可以将拉式算子离散化为：

$$(\Delta \mathbf{X})_u = \sum_{v \in \mathcal{N}_u} \omega_{uv} (\mathbf{x}_u - \mathbf{x}_v) \quad (4-56)$$

或者将其表示为矩阵的形式，作为一个 $n \times n$ 的实对称矩阵 $\Delta = \mathbf{D} - \mathbf{W}$ ，其中 \mathbf{D} 代表度矩阵 (Degree Matrix)，即 $\mathbf{D} = \text{diag}(d_1, d_2, \dots, d_n)$ ， $d_u = \sum_v \omega_{uv}$ 表示节点 u 的度。显然式 (4-56) 是一种排序不变性的局部聚合函数：

$$\phi(\mathbf{x}_u, \mathbf{X}_{\mathcal{N}_u}) = d_u \mathbf{x}_u - \sum_{v \in \mathcal{N}_u} \omega_{uv} \mathbf{x}_v \quad (4-57)$$

进而 $\mathbf{F}(\mathbf{X}) = \Delta \mathbf{X}$ 正是我们的蓝图的一个图上构造排序等变的示例。

值得注意的是，我们到目前为止没有讨论确定拉式矩阵的定义，实际上，这一构造对于任意的图都是成立的，其中邻接矩阵就是权重矩阵，即 $\mathbf{W} = \mathbf{A}$ ，也就是当 $(u, v) \in \mathcal{E}$ 时 $\omega_{uv} = 1$ ，其他情况均为 0。这样构造的拉式矩阵也被称为组合拉式算子，这反映了其仅仅就是图上的连接结构。对于几何图而言（也许不一定含有像面片模型具有的额外的结构，但节点具有空间位置，并可以据此定义度量），常常采用与度量成反向关系的权重，例如应用： $\omega_{uv} \propto e^{-\ell_{uv}}$ 。

在面片模型上，我们可以利用这些面带来的额外的结构，并应用余切公式定义权重 [Pinkall and Polthier, 1993]、[Meyer et al., 2003]：

$$w_{uv} = \frac{\cot \angle_{uqv} + \cot \angle_{upv}}{2a_u} \quad (4-58)$$

其中式 (4-58) 中的参数可以用图 4.14 表示。其中 \angle_{uqv} 以及 \angle_{upv} 是边 uv 对角的两个角， a_u 就是应用重心对偶时的面积，计算公式为：

$$a_u = \frac{1}{3} \sum_{v, q: (u, v, q) \in \mathcal{F}} a_{uvq} \quad (4-59)$$

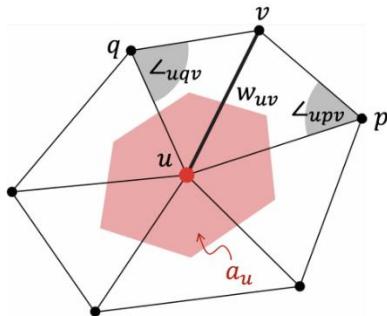


图 4.14 余切拉式算子，红色为对偶区域面积，源于 [Meyer et al., 2003] 的博

余切拉式算子具有一系列良好的性质 (参见[Wardetzky et al., 2007]): 其是一个正定的矩阵, $\Delta \succcurlyeq 0$, 因此有着非负的特征值, 便于将这些特征值与频率进行类比; 其也是实对称矩阵, 因此有着正交的特征向量; 余切拉式算子也是局部的, 其值仅依赖于节点邻居。也许余切拉式算子最重要的性质就是当面片模型不断精细化时其是连续的算子 [Wardetzky, 2008]。因此式 (4-58) 实际上构造了黎曼流形上的合适的离散拉式算子。值得注意的是精细化的过程需要满足一些条件, 例如避免三角网格变为病态, 一个又名的反例由德国数学家 Hermann Schwarz 给出, 称为 “Schwarz Stiefel” (“Schwarz Lantern”, 施瓦茨靴), 施瓦茨也是柯西-施瓦茨不等式的创造者。

式 (4-58) 并未显示处拉式算子是内参算子, 因此证明其内参性需要一些额外的努力。将拉式算子表示为仅有度量的形式:

$$w_{uv} = \frac{-\ell_{uv}^2 + \ell_{uq}^2 + \ell_{vq}^2}{8a_{uvq}} + \frac{-\ell_{uv}^2 + \ell_{up}^2 + \ell_{vp}^2}{8a_{uvp}} \quad (4-60)$$

使用海伦公式可得, 三角形的面积公式为:

$$a_{uvq} = \sqrt{s_{uvq}(s_{uvq} - \ell_{uv})(s_{uvq} - \ell_{vq})(s_{uvq} - \ell_{uq})} \quad (4-61)$$

这一分析表明了拉式算子的内参性, 这使得该算子具有等距同构不变性, 这一性质在计算机图形学以及几何处理中是十分好的性质 (参见[Wang and Solomon, 2019] 的介绍): 任意对于面片模型的变形只要不改变度量 (例如没有拉伸或者挤压), 就不会改变拉式算子

最后, 正如我们已经注意到的, 上述拉式算子的定义不受节点排序变换的影响, 这是由于其应用了聚合形式的运算。在一般的图上这也是一个必须的性质, 因为没有一个标准的图上节点的次序的定义, 不过实际上在面片模型上时我们可以按照某种方向对节点的次序进行设定 (例如顺时针), 这时唯一的不确定性就是哪个节点作为第一个节点。因而, 我们并非需要考虑所有可能的排序变换, 而是仅需要考虑循环移动变换即可, 这对应于 SO(2) 度规变换的歧义性 (也即无法区分谁应当是第一个节点, 译者注)。对于一个固定的度规, 定义一个各向异性拉式算子也是可能的, 这样的拉式算子对于方向具有敏感性, 在不同方向上具有不同的度量或者权重。[Andreux et al., 2014]、[Boscaini et al., 2016b] 等人使用形状算子定义了这种类型的拉式算子, [Boscaini et al., 2016a] 也在几何深度学习的早期架构中在面片模型上定义过这样的算子。

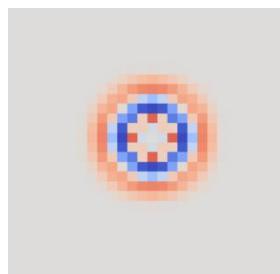


图 4.15 拉式算子是各向同性的, 在平面上其具有沿着半径方向的对称性

4.6.2 Mesh 谱分析 (Spectral Analysis on Meshes)

将拉式矩阵进行对角化的正交特征矢量构成的矩阵 $\Phi = (\varphi_1, \varphi_2, \dots, \varphi_n)$, 则应用该矩阵可将拉式矩阵对角化为: $\Delta = \Phi \Lambda \Phi^\top$, 其中 $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ 是由拉式矩阵特征值后成的对角线矩阵。该分解通常用于非欧式空间中, 作为欧氏空间的傅里叶变换基底的类比, 使得在面片模型上进行谱域卷积运算成为可能, 这也正如对应的傅里叶变换的卷积操作一样:

$$\mathbf{X} * \theta = \Phi \text{diag}(\Phi^\top \theta) (\Phi^\top \mathbf{X}) = \Phi \text{diag}(\hat{\theta}) \hat{\mathbf{X}} \quad (4-62)$$

其中滤波器 $\hat{\theta}$ 表示谱域的滤波器。值得注意的是该式是专门为面片模型设计的, 我们可以将其应用到一般的无向图中 (由于拉式矩阵是实对称的, 这要求图为无向图)。将这一谱域的卷积定义扩展到图上的 CNN 架构中是一项值得尝试的事, 本书的作者之一就做过这样的工作 [Bruna et al., 2013]。然而, 在非欧式空间中进行傅里叶变换似乎是对微小的扰动特别敏感的, 参见 4.4 节中的图片, 因而这一方面仅仅在域是固定不变的情况下才能使用, 可以用于分析不同的信号的作用, 但对于不同的域上进行直接地扩展则是不合理的。不幸的是, 很多计算机图形学的问题均需要处理变化的域的情况, 这样的任务中人们通常在一个域上进行训练并在另一个不同的域上进行测试, 使得应用基于傅里叶变换的方法显得有些不合适。

正如在 4.4 小节中提到的, 使用式 (4-47) 中的形式的谱域滤波器并应用一些传递函数 $\hat{p}(\lambda)$ 到拉式矩阵上是流行的做法:

$$\hat{p}(\Delta) \mathbf{X} = \Phi \hat{p}(\Lambda) \Phi^\top \mathbf{X} = \Phi \text{diag}(\hat{p}(\lambda_1), \hat{p}(\lambda_2), \dots, \hat{p}(\lambda_n)) \hat{\mathbf{X}} \quad (4-63)$$

当 \hat{p} 可以表示为矩阵-矢量乘积的形式时, 拉式矩阵的特征分解就可以避免了, 特征那个分解通常时间复杂度较高, 为 $\mathcal{O}(n^3)$ 。例如 [Defferrard et al., 2016] 使用多项式作为滤波函数:

$$\hat{p}(\Delta) \mathbf{X} = \sum_{k=0}^r \alpha_k \Delta^k \mathbf{X} = \alpha_0 \mathbf{X} + \alpha_1 \Delta \mathbf{X} + \dots + \alpha_r \Delta^r \mathbf{X} \quad (4-64)$$

这等价于将 $n \times d$ 维的特征矩阵 \mathbf{X} 与拉式矩阵乘若干次。由于拉式矩阵通常而言是稀疏的, 有 $\mathcal{O}(|\mathcal{E}|)$ 的非零元素, 因此该乘积操作具有较低的复杂度 $\mathcal{O}(|\mathcal{E}|dr) \sim \mathcal{O}(|\mathcal{E}|)$ 。面片模型是准正规的图, 每一个节点均有 $\mathcal{O}(1)$ 的邻居节点, 因此导致了 $\mathcal{O}(n)$ 的非零项。此外, 由于拉式矩阵是局部的, 一个 r 阶的多项式也是局部邻域的局部作用。

然而, 当处理面片模型时, 这一性质也有一定的弊端, 因为滤波器的支持区域依赖于面片模型的分辨率。我们必须明白面片模型来自于光滑表面的离散化表示, 因此我们可能得到两个完全不同的表示统一光滑表面的离散面片模型。在更精细化的表面上, 相对于粗糙的离散表示, 我们必须使用更大的邻域。

因此, 在计算机图形学应用中更常使用有理滤波器, 这样的滤波器依赖于分辨率。实

践中有很多定义这样的滤波器的方法（参考[Patanè, 2020]），最常用的就是使用一些有理函数的多项式，例如 $\lambda - 1/\lambda + 1$ 。更一般地，我们可以使用一个复数函数，例如应用将实数映射到复数域上平面的凯勒变换函数（Cayley Transform）： $\lambda - i/\lambda + i$ 。[Levie et al., 2018] 使用了作为凯勒多项式（Cayley Polynomials）的函数，以及带有复数系数的实有理函数 $\alpha_l \in \mathbb{C}$ ：

$$\hat{p}(\lambda) = \operatorname{Re} \left(\sum_{l=0}^r \alpha_l \left(\frac{\lambda - i}{\lambda + i} \right)^l \right) \quad (4-65)$$

在应用到矩阵时，凯勒多项式的计算要求矩阵的求逆操作：

$$\hat{p}(\Delta) = \operatorname{Re} \left(\sum_{l=0}^r \alpha_l (\Delta - i\mathbf{I})^l (\Delta + i\mathbf{I})^{-l} \right) \quad (4-66)$$

在信号处理中，多项式的滤波器也使用术语有限脉冲响应（Finite Impulse Response, FIR）来表示，有理滤波器也是用无限脉冲响应（Infinite Impulse Response, IIR）来表示。

式 (4-66) 可以在近似线性时间复杂度中计算。与多项式滤波器不同的是，有理滤波器没有局部支撑，而是有着指数衰减（参考[Levie et al., 2018]）。与直接计算傅里叶变换的一个关键的区别是多项式及有理滤波器在域上进行准等距同构的变换时能够保持稳定，论文[Levie et al., 2018]、[Levie et al., 2019]、[Gama et al., 2020]、[Kenlay et al., 2021] 展示了相关的众多结果。

4.6.3 算子及函数映射 Mesh (Meshes as Operators and Functional Maps)

泛函映射的示例促使我们思考将面片模型本身视为算子。正如我们即将展示的，这将使我们可以利用面片模型上额外的结构并获得更有趣类型的不变性。为便于讨论，假定面片模型 T 具有坐标特征矩阵 \mathbf{X} 。如果我们构建一个像拉式算子的内参算子，可以证明的是其也将面片模型的结构信息完全地进行了编码，我们也可以由算子恢复出模型本身（对等距同构不变的嵌入表示，正如[Zeng et al., 2012] 展示的）。这对于其他一些算子也是正确的命题（例如[Boscaini et al., 2015]、[Corman et al., 2017]、[Chern et al., 2018]），因而我们可以假定一个一般的算子，或者说一个 $n \times n$ 的矩阵 $\mathbf{Q}(T, \mathbf{X})$ 作为我们的面片模型本身的表示。

在这种视角下，4.1 节中对于形如 $f(\mathbf{X}, T)$ 的学习函数可以重新表示为 $f(\mathbf{Q})$ 。与图以及集合类似，面片模型的节点也没有标准的次序排列方式，也即定义在面片模型上的函数必须对于任意的排序矩阵 \mathbf{P} ，满足排列不变性或者排列等变性条件：

$$\begin{aligned} f(\mathbf{Q}) &= f(\mathbf{PQP}^\top) \\ \mathbf{PF}(\mathbf{Q}) &= \mathbf{F}(\mathbf{PQP}^\top) \end{aligned} \quad (4-67)$$

然而, 与一般的图相比我们的面片模型有着更多的结构: 我们可以假定面片模型是某个连续曲面 Ω 的离散化表示。我们因此也可以定义另一个面片模型 $\mathcal{T}' = (\mathcal{V}', \mathcal{E}', \mathcal{F}')$ 来表示和 \mathcal{T} 同一个连续的曲面 Ω , 但有着不同的节点数 n' 以及坐标矩阵 \mathbf{X}' 。更重要的是, 两个面片模型可以有着不同的连接结构 (例如一个模型采用三角网格, 另一个采用多边形网格, 译者注) 以及不同的节点数。因此我们不能仅根据排序矩阵 \mathbf{P} 对于节点的重排序作用来考虑两个面片模型是否是等距同构的。

[Ovsjanikov et al., 2012] 引出了泛函映射作为上述情况的一般性的关联关系的实现, 取代了以往的在两个域之间根据节点之间的关联关系来作为等距同构映射的判断, 而是采用映射函数 $\mathbf{C}: \mathcal{X}(\Omega) \rightarrow \mathcal{X}(\Omega')$ 来进行判断, 如图 4.16。一个泛函映射就是一个线性算子 \mathbf{C} , 使用 $n' \times n$ 的矩阵进行表示, 建立了两个不同离散化方式形成的面片模型之间的信号 \mathbf{x} 以及 \mathbf{x}' 的映射:

$$\mathbf{x}' = \mathbf{Cx} \quad (4-68)$$

通常而言, 泛函映射部署在谱域, 作为一个 $k \times k$ 的矩阵 $\hat{\mathbf{C}}$ 并作用与傅里叶变换后的信号上 $\mathbf{x}' = \Phi' \hat{\mathbf{C}} \Phi^\top \mathbf{x}$, 其中 Φ 以及 Φ' 分别表示截断的拉式矩阵的 $n \times k$ 的以及 $n' \times k$ 的基底, 其中 $k \ll n, n'$ 。

[Rustamov et al., 2013] 展示了使得映射前后局部上面积保持不变, 泛函映射必须是正交的, 也就是 $\mathbf{C}^\top \mathbf{C} = \mathbf{I}$, 也就是矩阵 \mathbf{C} 必须属于正交群 $O(n)$ 。在这样的情况下我们可以应用时 $\mathbf{C}^{-1} = \mathbf{C}^\top$ 进行求逆。

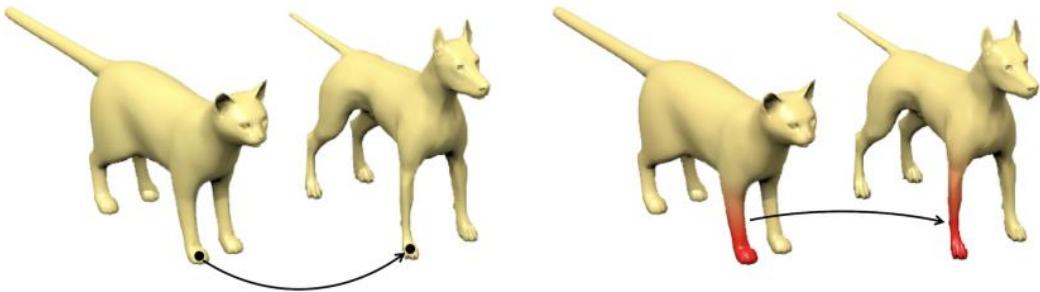


图 4.16 逐点对应 (左) 与泛函映射 (右)

泛函映射构建了两个算子表示的面片模型之间的对应关系:

$$\mathbf{Q}' = \mathbf{C} \mathbf{Q} \mathbf{C}^\top, \quad \mathbf{Q} = \mathbf{C}^\top \mathbf{Q}' \mathbf{C} \quad (4-69)$$

我们可以对此做如下解读: 给定一个面片模型 \mathcal{T} 的算子表示 \mathbf{Q} 以及一个泛函映射 \mathbf{C} , 我们可以首先将 \mathcal{T}' 中的信号用 \mathbf{C}^\top 映射到 \mathcal{T} , 然后再应用算子 \mathbf{Q} , 最后再使用 \mathbf{C} 将信号映射回 \mathcal{T}' 。这促使我们得到一种面片模型上面一类更加一般的重新离散化 (Remeshing) 不变 (或者等变) 的函数, 满足:

$$\begin{aligned} f(\mathbf{Q}) &= f(\mathbf{C} \mathbf{Q} \mathbf{C}^\top) = f(\mathbf{Q}') \\ \mathbf{C} \mathbf{F}(\mathbf{Q}) &= \mathbf{F}(\mathbf{C} \mathbf{Q} \mathbf{C}^\top) = \mathbf{F}(\mathbf{Q}') \end{aligned} \quad (4-70)$$

其中 $\mathbf{C} \in \mathrm{O}(n)$ 。可以看出之前设置的排序不变性以及排序等变性均是这一表示的特例形式（由于 $\mathbf{P}^\top \mathbf{P} = \mathbf{I}$ ），当节点发生排序作用的时候可以看成是一种微不足道（Trivial）的重新离散化。

[Wang et al., 2019a] 等人展示了，给定算子 \mathbf{Q} 的特征分解，任意的重新离散化不变或者等变的函数可以表示成 $f(\mathbf{Q}) = f(\mathbf{\Lambda})$ 以及 $\mathbf{F}(\mathbf{Q}) = \mathbf{V}\mathbf{F}(\mathbf{\Lambda})$ ，或者换句话说，重新离散化不变的函数仅仅与 \mathbf{Q} 的谱有关。实际上，拉式算子特征值的函数已经被证明对于曲面离散化方式以及微小的扰动鲁棒，这也解释了在计算机图形学中、图深度学习中 [Defferrard et al., 2016] 、[Levie et al., 2018] 应用拉式算子进行谱域构建是如此流行的原因。这一结论对于一般的算子 \mathbf{Q} 均有效，因此实际中有多种除了最常用到的拉式算子外的其他算子选择，例如狄拉克算子（Dirac Operator）[Liu et al., 2017] 、[Kostrikov et al., 2018] 斯特克洛夫算子（Steklov Operator）[Wang et al., 2018] 、可学习的参数化算子[Wang et al., 2019a] 等。

5 几何深度学习模型

我们已经在前边研究过几何深度学习蓝图的一些样例，例如对于域的选择不同、对称群不同以及局部性的概念等，现在讨论如何应用这些“处方”来导出目前一些流形的深度学习框架。

再次强调，在我们的展示中并不会按照严格的从特殊到一般的次序展开。我们首先介绍包括三种架构，这三种架构的实现能够直接地熊我们前述的讨论中得到。三种架构分别为：卷积神经网络（CNN）、群等变卷积神经网络（Group-equivariant CNN）、图神经网络（Graph Neural Network）。

随后我们也将对图神经网络的多种变体展开进一步分析，尤其是对于事前不知道图的结构的情况（也即无序集合），在讨论中我们也会介绍目前流行的图神经网络的例子，包括深度集合（DeepSets）架构以及 Transformer 架构。

在对于几何图以及面片模型的讨论中，我们首先介绍等变消息传递网络，其也是在图神经网络的计算中明确地引入了几何对称性。进而，我们将介绍多种将测地线及度规对称性应用到实际的深度学习网络中的方法，这包括了一系列内参卷积神经网络，如测地线 CNN（Geodesic CNN）、MoNet 以及度规等变面片 CNN（Gauge-Equivariant Mesh CNN）。

最后，我们从时域的角度再次审视网格域。这一讨论将引导我们对循环神经网络（RNN）进行探讨。我们将从 RNN 对于时域网格的平移等变形角度来介绍，当然也会介绍其对于时间扭曲变换的稳定性。这一性质对于处理许多具有远程依赖关系的情况尤其是有用的，保持对于时间扭曲的变换的类别的不变性正是门控 RNN（Gated RNN），包括非常流行的 RNN 模型，例如 LSTM 与 GRU。

尽管我们希望上述的内容已经能够尽可能包含目前在使用的深度学习关键架构，我们也清醒地知道新颖的神经网络架构也在不断地出现。因此，相对于囊括所有可能的架构，我们更关心介绍的内容是足够具有启发性的展示的，并促使读者能够对未来出现的任何的几何深度学习网络能够根据不变性与对称性轻易地进行分类。

$$\mathbf{x} \quad \underbrace{\mathbf{C}(\theta)}_{\theta_{11} + \theta_{13} + \theta_{22} + \theta_{31} + \theta_{33}} \quad \mathbf{x} * \theta$$

图 5.1 使用卷积核对信号进行卷积运算的过程，滤波器卷积核可以表示为生成器线性的组合

5.1 卷积神经网络 CNN

卷积神经网络也许是最早也是最广为人知的能够由几何深度学习导出的深度学习架构的例子（在 3.5 节中有所叙述）。在 4.2 节中我们已经完全介绍了线性与几步平移等变算子的特性，亦即通过信号与局部滤波器的卷积运算 $\mathbf{C}(\boldsymbol{\theta}) = \mathbf{x} * \boldsymbol{\theta}$ 。我们首先着眼于标量的图片（单通道图片，亦即灰度图，译者注），其中域就是 $\Omega = [\mathbf{H}] \times [\mathbf{W}]$ ，像素坐标 $\mathbf{u} = (u_1, u_2)$ ，信号 $\mathbf{x} \in \mathcal{X}(\Omega, \mathbb{R})$ 。

任何与尺寸为 $H^f \times W^f$ 的紧致局部滤波器的卷积可以写为生成器 $\boldsymbol{\theta}_{1,1}, \dots, \boldsymbol{\theta}_{H^f, W^f}$ 的线性组合的形式，例如使用单位峰值形式 $\boldsymbol{\theta}_{vw}(u_1, u_2) = \delta(u_1 - v, u_2 - w)$ 。任意局部线性等变映射可以表示为：

$$\mathbf{F}(\mathbf{x}) = \sum_{v=1}^{H^f} \sum_{w=1}^{W^f} \alpha_{vw} \mathbf{C}(\boldsymbol{\theta}_{vw}) \mathbf{x} \quad (5-1)$$

应用坐标表示时，上式就是我们熟悉的二维卷积，参考图 5.1：

$$\mathbf{F}(\mathbf{x})_{vw} = \sum_{a=1}^{H^f} \sum_{b=1}^{W^f} \alpha_{ab} x_{u+a, v+b} \quad (5-2)$$

选择其他类型的基生成器 $\boldsymbol{\theta}_{vw}$ 也是可能的并且会产生等价的操作（可能选择的 α_{vw} 也不同）。一种流行的选择是使用方向导数： $\boldsymbol{\theta}_{vw}(u_1, u_2) = \delta(u_1, u_2) - \delta(u_1 - v, u_2 - w)$ ，其中坐标 $(v, w) \neq (0, 0)$ ，并经过局部平均 $\boldsymbol{\theta}_0(u_1, u_2) = 1/H_f W_f$ 。实际上，方向导数也可以视为是一种图上的扩散过程对应的在网格域上的类比，其中我们假设了网格中每一个像素点是一个与其邻居相连接的节点。

当使用多个通道的输入数据而非标量数据时（例如 RGB 图像，或者更一般地任意数量的特征映射），卷积滤波器就成了卷积张量，该卷积张量能够表示为任意的输入特征到输出特征的线性组合的形式。用坐标表示为：

$$\mathbf{F}(\mathbf{x})_{uvj} = \sum_{a=1}^{H^f} \sum_{b=1}^{W^f} \sum_{c=1}^M \alpha_{jabc} x_{u+a, v+b, c}, \quad j \in [N] \quad (5-3)$$

其中 M 、 N 分别代表输入与输出的通道数。这一基本的操作涵盖了众多神经网络架构，这对众多领域（例如计算机视觉、信号处理以及其他）均产生了深远的影响，我们将在下一节详细叙述。相较于介绍卷积神经网络的众多可能的架构形成的金字塔，我们更愿意关注于那些使得他们广泛流行的核心创新点。

5.1.1 多尺度高效计算 (Efficient Multiscale Computation)

正如在几何深度学习蓝图对于一般的对称性的介绍中所述，从卷积算子 \mathbf{F} 中提取平移不变性的特征要求非线性的单元。卷积特征是通过一个非线性的激活函数 σ 来进行计算的，该函数逐元素地作用于输入，也就是 $\sigma: \mathcal{X}(\Omega) \rightarrow \mathcal{X}(\Omega)$ 满足 $\sigma(\mathbf{x})(u) = \sigma(\mathbf{x}(u))$ 。在本

书写作之时也许作为流行的激活函数就是修正线性单元（Rectified Linear Unit, ReLU），其定义为： $\sigma(x) = \max(x, 0)$ 。该非线性单元有效地修正了信号，将信号的能量朝着低频率段移动，并通过迭代构建过程使得计算不同尺度上的高频关联成为可能。

ReLU 经常被视为“现代的”激活函数选择，其在[Fukushima and Miyake, 1982] 已经被使用了。在他们的应用中修正等价于反模块化的原则，这在电子工程中是基础性的原则并被广泛用于传输协议中，例如 FM 音频广播。该函数在神经元的激活模型中也有着至关重要的作用。

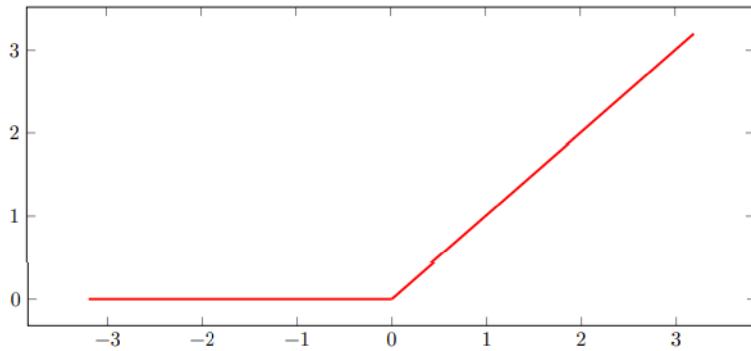


图 5.2 ReLU 激活函数

卷积神经网络以及类似的架构有责多尺度的结构，在每一层卷积层后面人们通常使用一层网格粗糙化层 $\mathbf{P}: \mathcal{X}(\Omega) \rightarrow \mathcal{X}(\Omega')$ ，其中网格域 Ω' 有着更粗糙的分辨率，这样的结构最早在[Fukushima and Miyake, 1982] 以及[LeCun et al., 1998] 中有所应用。粗糙化层使得多尺度滤波并且有效地增加了感受野，并能够在每一个尺度上维持常数个参数。若干种粗糙化策略（也叫池化、聚合等，译者注）可供选择，最常见的是在网格降采样后应用一个低通反锯齿（Anti-Aliasing）滤波器（如局部平均）或者非线性的最大化池化（Max-Pooling）。

总之，“寻常的”卷积神经网络（”Vanilla” CNN）可视为在我们的几何深度学习蓝图中已经介绍国的基本对象的组合：

$$\mathbf{h} = \mathbf{P}(\sigma(\mathbf{F}(\mathbf{x}))) \quad (5-4)$$

也即一个等边线性层 \mathbf{F} ，聚合池化层 \mathbf{P} 以及非线性激活函数 σ 组合而成。在 CNN 中应用全局的平移不变性的聚合池化也是可行的。直观来看，这关系到每一个像素点，每个像素点在经过若干卷积层后聚合为一个片（Patch），每一个像素点均对最后的图像表示有所贡献，并在最终的激活函数选择后以一种聚合的形式表达。一种流行的聚合函数选择就是平均化函数，这是由于其能够保持输出的幅度与图像尺寸无关[Springenberg et al., 2014]。此处介绍的 CNN 网络也被成为全卷积网络，与之对应，很多 CNN 网络架构在足够多的等边线性变换层以及池化聚合层后将图片扁平处理并把其传输到多层感知机中，这样的卷积操作失去了平移不变性。

一些流行的 CNN 架构例子（我们将会在随后讨论其中一些）如图 5.3 所示。

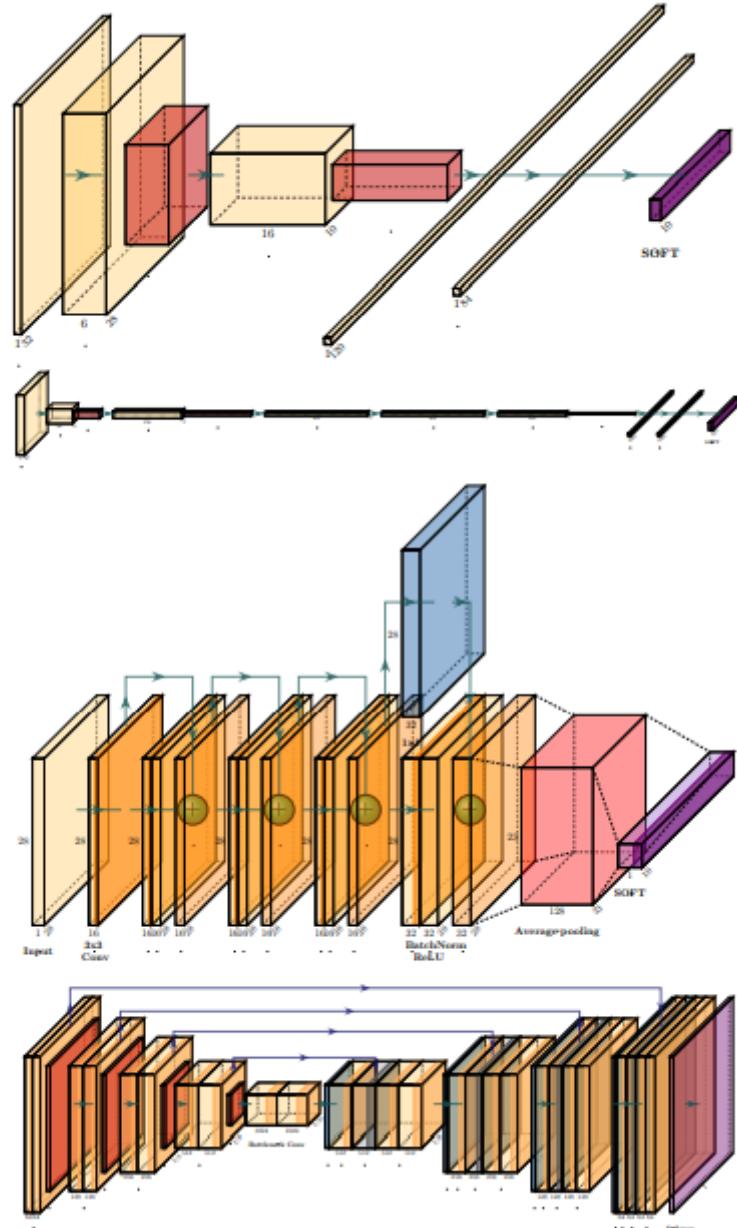


图 5.3 流行的 CNN 架构：从上而下：LeNet[LeCun et al., 1998]、AlexNet[Krizhevsky et al., 2012]、ResNet[He et al., 2016]、U-Net[Ronneberger et al., 2015]，图片采用 PlotNeuralNet[Iqbal, 2018] 包

5.1.2 深度及残差网络（Deep and Residual Network）

最简单形式的卷积神经网络由其超参数 $(H_k^f, W_k^f, N_k, p_k)_{k \leq K}$ 来确定，且 $M_{k+1} = N_k$, $p_k = 0, 1$ 来表明网格池化聚合是否使用。尽管这些所有的参数在应用中均十分重要，一个异常重要的问题就是理解卷积神经网络架构中网络深度所扮演的角色，以及在选择这一超参数时我们做了怎么的基础性的折中，尤其是与滤波器尺寸 (H_k^f, W_k^f) 有关的折中。

尽管从理论上进行严格证明仍然困难重重，观察近几年来收集的经验证据可以发现，我们倾向于一种更深的网络（更大的 K ）、更薄的模型（更小的 (H_k^f, W_k^f) ）的折中。这一折中下，一个关键性的洞察[He et al., 2016] 就是重新参数化每一层卷积层以对过去特征的扰动进行建模，而非一般的非线性变换。这一残差网络模型可以追溯到高速路网络 [Srivastava et al., 2015]，其使得更一般的门控机制以控制残差信息流：

$$\mathbf{h} = \mathbf{P}(\mathbf{x} + \sigma(\mathbf{F}(\mathbf{x}))) \quad (5-5)$$

得到的残差网络相对于在其之前的架构一些关键性的优势。本质上，残差参数化与深度网络是一种其内含的连续动态系统的离散化表示的观点一致，该观点常使用微分方程来建模（ODE）。重要的是，通过对速度进行建模而非对位置进行建模对于学习一个动态系统而言容易得多。在这样的观点下，残差网络正是一个微分方程的前向欧拉离散化表示 $\dot{\mathbf{x}} = \sigma(\mathbf{F}(\mathbf{x}))$ 。在我们的深度学习的设定中，这一做法引导我们走到了一种有着更好的几何先验的优化中，使得训练更深的网络成为可能。正如将要在未来的工作中进一步介绍的，使用深度神经网络进行学习等价于一个非凸优化问题，该问题可以在简化后使用基于梯度下降的方法有效地求解。残差网络参数化的关键优势已经在一些简单的场景下被严格地分析了[Hardt and Ma, 2016]，并且仍然是一个较为活跃的理论分析方向。最后神经 ODE[Chen et al., 2018]（Neural ODE）是最近的一个流行的架构，其利用了更多 ODE 的相似性，通过直接学习 ODE 的参数 $\dot{\mathbf{x}} = \sigma(\mathbf{F}(\mathbf{x}))$ 并依赖于标准数值积分方法。

5.1.3 正则化 (Normalization)

另一个极大地促进了卷积神经网络的经验表现提升的重要的算法创新是正则化的思想。在一些早期神经元活动的模型中，人们通常假定神经元执行了某种形式的局部“增益控制”，其中层参数 \mathbf{x}_k 被更新为了 $\tilde{\mathbf{x}}_k = \sigma_k^{-1} \odot (\mathbf{x}_k - \boldsymbol{\mu}_k)$ 。在此， $\boldsymbol{\mu}_k$ 与 σ_k 分别编码了 \mathbf{x}_k 的第一及第二阶距信息。进一步讲，其可以全局地进行计算，也可以局部地计算。

在深度学习的内容中，该准则被广泛应用，如通过批量正则化层[Ioffe and Szegedy, 2015] 以及一些变体[Ba et al., 2016]、[Salimans and Kingma, 2016]、[Ulyanov et al., 2016]、[Cooijmans et al., 2016]、[Wu and He, 2018]。我们注意到正则化激活的神经网络甚至在批量正则化出现之前已经吸引了一些注意[Lyu and Simoncelli, 2008]。尽管存在一些试图从更好的处理优化方法出发严格地解释正则化的好处的资料[Santurkar et al., 2018]，在本书写作之时尚且没有一个通用的理论能够提供良好的解释。

5.1.4 数据增强 (Data Augmentation)

尽管卷积神经网络对与平移不变性及尺度分离的几何先验进行了编码，他们并没有明确地解释其他种类的能够保持语义信息的变换，例如光照或颜色变化，或者小的转动与扩

张等变换。一种结合这些变换并且仅对原有架构进行微调的实用方法是应用数据增强，其中我们手动地应用前述变换到输入图像，并将其视为训练集。

数据增强取得了经验角度的成功并被广泛实用，不仅仅用于训练当前最优的视觉架构，也催生了一些自监督学习方法以及因果表示学习的发展[Chen et al., 2020]、[Grill et al., 2020]、[Mitrovic et al., 2020]。然而，其被证明了对于样本的复杂度而言是次优的[Mei et al., 2021]，一种更加高效的策略就是考虑更丰富发不变性群作用，我们将在接下来讨论。

5.2 群等变卷积神经网络 (Group-equivariant CNNs)

正如 4.3 节中讨论的，我们可以将应用于欧氏空间的信号的卷积操作通过群 \mathfrak{G} 的群作用泛化到任意的齐次空间 Ω 的信号上。齐次空间指代一个集合 Ω 及转移群作用，意即对任意的 $u, v \in \Omega$ ，存在 $g \in \mathfrak{G}$ 使得 $g.u = v$ 。类比于移动滤波器与信号进行运算的欧氏空间卷积，群卷积的思想是在域中实用群作用移动滤波器，例如实用旋转或者平移群作用。由于群作用的传递性，我们可以将滤波器移动到域中任意的位置上。在本节中，我们将讨论这一群卷积的一般性想法的实际的例子，包括部署时的一些方面的问题以及架构的选择。

5.2.1 离散群卷积 (Discrete Group Convolution)

我们首先考虑群 \mathfrak{G} 以及域 Ω 均为离散的情况。作为第一个实例，我们考虑以三维网格上的信号表示的医学空间图像以及其上离散的平移与旋转对称性。该域是三维的立方网格 $\Omega = \mathbb{Z}^3$ ，图像（例如 MRI 或者 CT 3D 扫描）可视为是映射 $x: \mathbb{Z}^3 \rightarrow \mathbb{R}$ ，其中 $x \in \mathcal{X}(\Omega)$ 。尽管实际上这样的图像是在有限的三维网格 $[W] \times [H] \times [D] \subset \mathbb{Z}^3$ 中，我们通常将其视为无穷的整数域三维网格，某些区域填充 0。作为我们的对称群，考虑群 $\mathfrak{G} = \mathbb{Z}^3 \rtimes O_h$ ，即为距离以及方向保持的变换群。该群包括平移变换、绕三个坐标轴的 90° 离散旋转变换，如图 5.4。

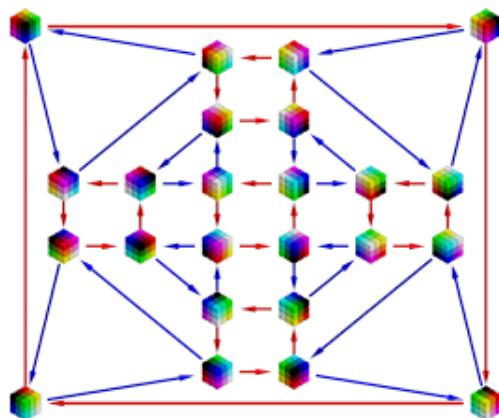


图 5.4 一个 3×3 卷积核，经过离散旋转群的 24 个群元素作用，红色表示
90° 旋转作用，蓝色表示 120° 旋转作用

我们第二个例子是 DNA 的序列结构，其是一种双螺旋结构，由四种碱基构成，碱基配对关系为 (A/T, C/G)。DNA 序列可视为一维网格 $\Omega = \mathbb{Z}$ ，其上信号 $x: \mathbb{Z} \rightarrow \mathbb{R}^4$ ，其中每一个碱基均被编码为 \mathbb{R}^4 空间的热点编码。自然而然地，我们有了 1 维的平移对称群，但是 DNA 也有额外的有趣的对称性。这种对称性源于物理上 DNA 呈现出双螺旋结构以及其被细胞解析的分析机制。双螺旋结构的每一根条带均已被称为 5' 端开始，并以被称为 3' 的位置结尾，并且一个条带的 5' 与另一个互补的条带的 3' 位置对应。换言之，两个条带有着相反的方向。虽然 DNA 条带总是从 5' 开始读，但是我们并不知道是哪一条，例如一个序列 ACCCTGG 与其逆的补 CCAGGGT 等价。这一现象称为 DNA 碱基字母序的逆向互补对称性。我们因而有了含有两个元素的群 $\mathbb{Z}_2 = \{0, 1\}$ 分别与原序列及其逆向互补序列。全部群构成了平移以及逆向互补变换。

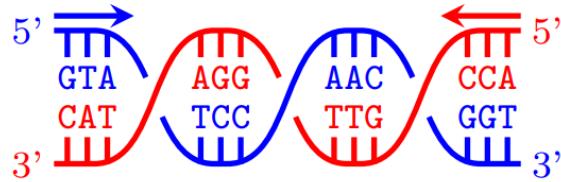


图 5.5 DNA 双螺旋碱基对配对情况，两个条带用不同颜色标识

在这个例子中，我们在 4.3 节中定义的群卷积可表示为：

$$(x * \theta)(g) = \sum_{u \in \Omega} x_u \rho(g)\theta_u \quad (5-6)$$

单通道的输入信号 x 与滤波器 θ 的内积经过群作用 $g \in \mathfrak{G}$ 变换，即为 $\rho(g)\theta_u = \theta_{g^{-1}u}$ ，卷积输出 $x * \theta$ 是一个定义在 \mathfrak{G} 的函数。注意到由于 Ω 是离散的，我们采用求和的方式而非积分的方式。

5.2.2 变换+卷积方法 (Transform + Convolve Approach)

群卷积可以分两步来进行实现部署：(1) 滤波变换步骤；(2) 平移卷积步骤。滤波变换步骤包含构造基滤波器的旋转后的拷贝，而平移卷积步骤与标准的卷积神经网络相同因而可以使用诸如 GPU 之类的硬件进行高效的计算。为明确这一点，注意到在我们的两个例子中均可以把一般性的变换 $g \in \mathfrak{G}$ 表示为一个变换 $h \in \mathfrak{H}$ （例如旋转或者逆向互补变换）与另一个平移群变换 $l \in \mathbb{Z}^d$ 的组合，即 $g = lh$ 。实用群表示的性质，我们有

$$\rho(g) = \rho(lh) = \rho(l)\rho(h)，\text{ 因而：}$$

$$\begin{aligned} (x * \theta)(lh) &= \sum_{u \in \Omega} x_u \rho(l)\rho(h)\theta_u \\ &= \sum_{u \in \Omega} x_u (\rho(h)\theta)_{u-l} \end{aligned} \quad (5-7)$$

我们认识到后一个等式正是平面欧氏空间的卷积运算，即信号 x 与变换后的滤波器 $\rho(h)\theta$ 的卷积。因此，为这些群部署实现群卷积，我们首先使用标准的滤波器 θ 并计算对

于任意的 $\mathfrak{h} \in \mathfrak{H}$ (例如旋转群 $\mathfrak{h} \in O_h$ 或者 DNA 的逆向互补群 $\mathfrak{h} \in \mathbb{Z}_2$) 其变换后的滤波器 $\theta_{\mathfrak{h}} = \rho(\mathfrak{h})\theta$, 然后将信号 x 与变换后的滤波器进行卷积运算 $(x * \theta)(\mathfrak{l}\mathfrak{h}) = (x * \theta_{\mathfrak{h}})(\mathfrak{l})$ 。对于我们提到的两个例子, 对称群作用在滤波器上时仅仅表现为对滤波器参数进行重排序, 正如图 5.4 所示。因此, 这些运算可以通过预计算的索引使用索引操作来高效地计算。

尽管之前已经利用定义在 \mathfrak{G} 上的群卷积作为特征映射输出, 我们可以将群作用 \mathfrak{g} 分解为两个群作用 \mathfrak{h} 与 \mathfrak{l} 的事实意味着我们可将群卷积视为欧式空间特征映射的叠加 (有时也被称为方向通道, Orientation Channels), 每一个特征映射对应一个滤波变换及方向 \mathfrak{l} 。例如, 在第一个例子中, 我们将每一个滤波旋转与一个特征映射关联起来, 这是通过与旋转后的卷积运算实现的。这些特征映射因此可以存储为 $W \times H \times C$ 的数组, 其中通道 C 等于独立的滤波器数目乘以群元素 $\mathfrak{h} \in \mathfrak{H}$ 数目。

正如 4.3 节中介绍的, 群卷积是等变作用: $(\rho(\mathfrak{g})x) * \theta = \rho(\mathfrak{g})(x * \theta)$ 。对于方向通道而言这意味着在群元素 \mathfrak{h} 作用下, 每一个方向通道均被变换了, 方向通道本身发生了排序作用。例如, 若我们将一个方向通道与一个变换进行关联 (如图 5.4), 并且应用一个绕着 z 轴的 90° 旋转作用, 特征映射将按照红色箭头所示的方向进行排序作用。这一描述指明了群卷积的神经网络与传统 CNN 网络有非常多的相似之处。因此 5.1 节中讨论的很多网络的设计模式, 例如残差网络, 也可以使用群卷积。

5.2.3 傅里叶频域球状 CNN (Spherical CNNs in the Fourier Domain)

对于在 4.3 中介绍的连续的对称群, 可以在谱域中进行群卷积的部署实现, 使用合适的傅里叶变换 (值得一提的是对于 \mathbb{S}^2 上的卷积是一个作用在 $SO(3)$ 的函数, 因此为部署多层球状卷积神经网络我们需要在两个域中分别定义傅里叶变换)。球状调和函数是二维球面上的正交基底, 这与复指数域上经典的傅里叶基底类似。在特殊正交群中, 傅里叶基底也被称为 Wigner D-functions。在两种情况中, 傅里叶变换均使用与基函数的内积来计算, 卷积基本定理的类比定理也成立: 在谱域进行逐元素的乘积等价于时域卷积。进一步来说, 快速傅里叶算法也可应用于高效地计算 \mathbb{S}^2 以及 $SO(3)$ 上的傅里叶变换。若要进一步了解, 我们推荐读者阅读 [Cohen et al., 2018]。

5.3 图神经网络 GNN

图神经网络是我们提出的几何深度学习蓝图在图上的应用, 利用了排序群作用的性质。图神经网络也是现存深度学习架构的最一般的表示形式, 正如我们将马上看到的, 大多数其他类型的深度神经网络均可以视为图神经网络的一种带有额外几何结构的特例。

正如在 4.1 节中探讨的一样, 考虑一个图以及其邻接矩阵 \mathbf{A} 与节点特征矩阵 \mathbf{X} , 我们将研究排序等变函数 $\mathbf{F}(\mathbf{X}, \mathbf{A})$ 作用的图神经网络, 该函数可以通过应用在某一结点的邻域的共享排序不变性函数 $\phi(\mathbf{x}_u, \mathbf{X}_{N_u})$ 构建。该排序不变函数有多种表现形式, 其可以视为

“扩散” (“Diffusion”)、“传播” (“Propagation”)、或者“消息传递” (“Message Passing”), 整体的排序等变函数的计算则称为图神经网络层 (GNN Layer)。

在本书写作之时设计以及研究图神经网络的每一层是深度学习中最热门的研究领域,这也使得其成为充满挑战性的地标。幸运的是,我们发现大多数文献可以由三种不同的图神经网络层来导出 (如图 5.6)。这三种不同的类型是根据排序等变函数在某种程度上是如何对邻域特征进行变换而划分的,这样的划分使得可以对图上各种复杂度的变化相互作用关系进行建模。

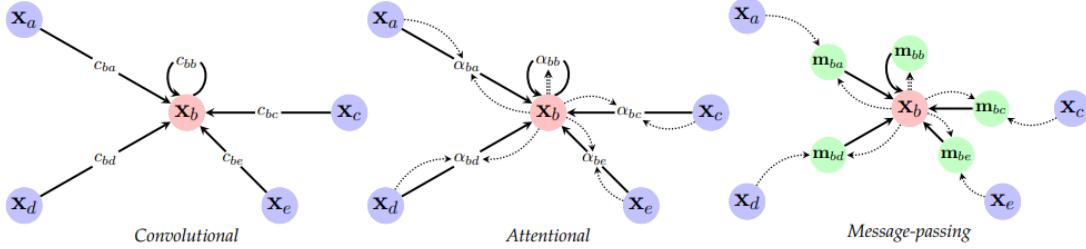


图 5.6 三种不同的图神经网络层计算方式。从左至右: 图卷积神经网络、图注意力网络、图消息传递网络

在这三种类型中,排序不变性是通过使用聚合邻域特征 $\mathbf{X}_{\mathcal{N}_u}$ (该邻域特征可能经过某种函数 ψ 的变换作用,即经过数据增强,译者注) 的方式实现的,聚合的函数 \oplus 也有多种选择,聚合之后再对节点 u 特征通过某种函数 ϕ 进行更新。典型的函数 ψ 与 ϕ 均为可学习的参数,聚合函数 \oplus 可使用非参数化的操作,例如求和、平均、最大、最小等来构建,当然其也可以使用循环神经网络来构建 [Murphy et al., 2018]。通常来看, ψ 与 ϕ 均为可学习的仿射变换,即可以表示为 $\psi(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{b}$ 以及 $\phi(\mathbf{x}, \mathbf{z}) = \sigma(\mathbf{W}\mathbf{x} + \mathbf{U}\mathbf{z} + \mathbf{b})$, 其中 $\mathbf{W}, \mathbf{U}, \mathbf{b}$ 均为可学习的参数, σ 为激活函数,例如使用 ReLU 激活。函数 ϕ 中额外的输入 \mathbf{x}_u 表示可选的跳跃联络 (Skip-connection),该项通常是非常有用的。

5.3.1 图卷积网络 (Graph Convolution)

在图卷积网络中 ([Kipf and Welling, 2016a]、[Defferrard et al., 2016]、[Wu et al., 2019]),邻域节点的特征直接地用于当前节点的聚合,其权值为固定参数:

$$\mathbf{h}_u = \phi \left(\mathbf{x}_u, \bigotimes_{v \in \mathcal{N}_u} c_{uv} \psi(\mathbf{x}_v) \right) \quad (5-8)$$

该式中 c_{uv} 表明了节点 v 的特征对于当前节点 u 特征的重要程度。其使用常数表示并且直接依赖于表示整个图结构的邻接矩阵。值得注意的是,聚合算子 \oplus 使用了求和,因而可以视为线性扩散或者位置相关的线性滤波,这也是一种卷积操作的扩展。特别地,再 4.4 节及 4.6 节中介绍过的谱域滤波也属于图卷积形式,这是由于其应用了固定的局部算子 (例如拉式矩阵) 到逐节点的特征信号上。注意此处介绍的图卷积形式并未包括所有的图上卷积操作,但是覆盖了实践中应用的大多数架构,我们将在未来的工作中对此进行详细

阐述。

5.3.2 图注意力网络 (Graph Attention)

在图注意力网路中 ([Veličković et al., 2018]、[Monti et al., 2017]、[Zhang et al., 2018]), 当前节点与邻域节点的相互作用关系为:

$$\mathbf{h}_u = \phi \left(\mathbf{x}_u, \bigotimes_{v \in \mathcal{N}_u} a(\mathbf{x}_u, \mathbf{x}_v) \psi(\mathbf{x}_v) \right) \quad (5-9)$$

其中 a 表示可学习的自注意力机制, 通过隐式的关系 $a_{uv} = a(\mathbf{x}_u, \mathbf{x}_v)$ 来计算重要性参数, 他们也经常使用 Softmax 函数对邻域节点特征进行正则化。当 \oplus 表示求和时, 聚合仍然是邻域节点信号的线性组合, 但是与图卷积形式相比, 现在的权重与特征有关。

5.3.3 图信息传递网络 (Message Passing)

最后一类为消息传递网络, 该类型计算不同的边之间的任意的矢量 (也称消息):

$$\mathbf{h}_u = \phi \left(\mathbf{x}_u, \bigotimes_{v \in \mathcal{N}_u} \psi(\mathbf{x}_u, \mathbf{x}_v) \right) \quad (5-10)$$

其中 ψ 为可学习的消息函数, 其计算节点 v 的特征并将发送给节点 u , 聚合可视为图上的一种消息传递。

对于三种类型而言, $convolution \subseteq attention \subseteq message-passing$ 。确实, 图注意力网络可以用来表示图卷积网络, 仅需要设定参数为常数即可, 这两种类型均可以使用消息传递网络来表示。

然而这并不意味着消息传递总是最有用处的变体选择, 由于消息传递类型总是计算基于矢量的表示, 因为训练较为困难, 也会需要更多的内存。此外, 对于众多自然出现的图而言, 图上的边编码了下游的类型的相似程度 (例如边 (u, v) 暗示了节点 u 与节点 v 极有可能有着类似的输出)。对于这类图 (通常被称为强关系与同质构成, Homophilous), 邻域上的卷积聚合通常是一个更好的选择, 这对于正规化以及尺度分离角度均是如此。注意力网络提供了一种折中选择, 他们可以对邻域的复杂关系进行建模但又仅仅计算边之间的标量的影响量, 使得注意力网络比消息传递网络更能适应大规模的图。

这三种类型在此处介绍的较为简略, 因而不可避免地忽视众多的细微差别、洞察, 一般性以及图神经网络的历史发展。重要的是, 这样的分类排除了基于 Weisfeiler-Lehman 结构的高纬度图神经网络以及需要明确计算图傅里叶变换的谱域图神经网络。

5.4 Deep Sets、Transformer、潜图推理 (Latent Graph Inference)

我们通过说明对于学习无序集合的排序等变形网络架构来结束图神经网络的探讨。尽管这样的集合相较前述讨论的域而言有着最少的结构，他们的重要性被众多的神经网络框架重点尽心了强调，诸如流行的 Transformer[Vaswani et al., 2017]、Deep Sets[Zaheer et al., 2017] 等架构。在 4.1 节中的描述中，我们假定了给定一个节点特征矩阵 \mathbf{X} ，但是并未假定邻接矩阵或者节点的排序信息。具体的架构将源自于决定在何种程度上对节点之间的相互关系进行建模。

5.4.1 空边集合（Empty Edge Set）

无序集合没有额外的结构或者任意的几何形式，因此最自然的处理方式就是将集合中的元素独立地进行处理。这指向了对输入特征矩阵应用排序等变函数，也就是在 4.1 节中介绍的：应用一个共享的变换，并作用到集合中每一个孤立的节点上。应用与图神经网络一样的符号，这样的模型可以表示为：

$$\mathbf{h}_u = \psi(\mathbf{x}_u) \quad (5-11)$$

其中 ψ 为可学习的变换函数。也可以将上式视为图神经网络的一个特例，即 $\mathcal{N}_u = \{u\}$ 的特例，或者说邻接矩阵 $\mathbf{A} = \mathbf{I}$ 。这样的架构通常被称为是深度集合（Deep Sets）[Zaheer et al., 2017]，该架构也从理论上证明了若干能逼近的性质。值得一提的是，处理无序集合的需求常见于计算机视觉以及图形学中对于点云数据的处理中，在这些领域也有称为 PointNets[Qi et al., 2017] 的模型存在。

5.4.2 完备边集合（Complete Edge Set）

尽管假定空边集合对于无序集合而言有着计算高效的优势，我们经常集合中元素呈现出某种形式的关联关系，例如节点之间存在一个潜图。将邻接矩阵设为 $\mathbf{A} = \mathbf{I}$ 则直接忽视了任意的结构，因而可能导致次优的表现。相反，在没有任何先验知识的前提下，我们可以假定不去除任意一条可能的节点间关联。在这样的方法中，我们假定一个完整的图，其邻接矩阵 $\mathbf{A} = \mathbf{1}\mathbf{1}^\top$ ，等价地， $\mathcal{N}_u = \mathcal{V}$ 。我们并没有假定计算所有的相互作用的参数，在这样的图上应用图卷积操作：

$$\mathbf{h}_u = \phi\left(\mathbf{x}_u, \bigotimes_{v \in \mathcal{V}} \psi(\mathbf{x}_v)\right) \quad (5-12)$$

其中第二项 $\bigoplus_{v \in \mathcal{V}} \psi(\mathbf{x}_v)$ 对于所有的节点均相同，因而这使得完备边集合网络与与忽视空边集合作用等价，也即 $\mathbf{A} = \mathbf{I}$ ，这也是聚合操作的排序不变性的直接结果。

这促使我们使用更具有表达能力的图神经网络，即使用注意力网络：

$$\mathbf{h}_u = \phi \left(\mathbf{x}_u, \bigotimes_{v \in \mathcal{V}} a(\mathbf{x}_u, \mathbf{x}_v) \psi(\mathbf{x}_v) \right) \quad (5-13)$$

这导出了自注意力算子，该算子也是 Transformer 架构的核心[Vaswani et al., 2017]。假定对注意力参数施加某种形式的正则化（例如 Softmax），我们可以将注意力系数限制在 $[0, 1]$ 之间，并且可以将这样的自注意力视为推断一个“软性”的邻接矩阵， $a_{uv} = a(\mathbf{x}_u, \mathbf{x}_v)$ ，这也可以作为下游的任务中的一个梯度下降优化中的副产品。

上述分析视角意味着可以将 Transformer 视为完备图上的注意力图神经网络[Joshi, 2017]。然而，这与 Transformer 架构最初用来对序列建模的初心相矛盾了，潜层表示 \mathbf{h}_u 应当对序列中节点 u 的位置进行记忆，但是完备图的聚合操作将其信息丢弃了。Transformer 通过使用位置编码来解决这一问题：节点特征矢量 \mathbf{x}_u 同时编码了节点的位置信息，通常采用频率依赖于节点 u 的位置的正弦波进行采样。处理应用注意力类型的操作外，也可以应用消息传递类型的操作，尽管在物理仿真以及关系推理中应用较多，但他们并不像 Transformer 一样被广为使用，这可能与内存问题有关，并且基于矢量的消息相较于注意力机制的软邻接矩阵也更加难以明确地解释。

尽管在图上没有自然的节点排序的存在，但也存在若干对位置进行编码的替代性的方案。虽然我们会将这些替代性的选择在后面才进行介绍，在此提醒读者，一个涉及到 Transformer 中应用的位置编码方法的实现的很有希望的方向直接地与离散傅里叶变换 (DFT) 有关，也即与循环网格的图拉式矩阵的特征矢量有关。因此 Transformer 的位置编码隐式地表示了对于输入节点再循环的网格中的假设。对于更加一般的图结构，也可以简单地应用图的拉式矩阵特征向量，这也是[Dwivedi and Bresson, 2020] 等人构造由经验出发构造的 Graph Transformer 模型的一个结论。

5.4.3 推断边集合 (Inferred Edge Set)

最后，我们可以尝试学习潜在的关系结构，从而这时邻接矩阵不再是 \mathbf{I} 也不再是 $1\mathbf{1}^\top$ 。推断一个 GNN 可用的潜邻接矩阵的问题对于图表示学习是非常重要的。这是因为假定 $\mathbf{A} = \mathbf{I}$ 时表示学习可能过于低阶，而 $\mathbf{A} = 1\mathbf{1}^\top$ 又由于内存消耗以及大量的邻域节点的聚合问题而难于部署。此外，推断邻接矩阵也与真正的问题紧密相关：推断邻接矩阵 \mathbf{A} 意味着检测特征矩阵 \mathbf{X} 的行之间的结构关系，这也可能帮助更好地进行假定，例如假定变量之间的因果关系。

不幸的是，这样的一个机构引入了更多的复杂度。具体来说，其要求在结构学习目标（由于是离散的，所以应用梯度下降优化时困难重重）与下游的图上任务之间做出合适的权衡，这使得图推理是十分具有挑战性并且是敏感的问题。

5.5 等变消息传递网络 (Equivariant Message Passing)

Networks)

在图神经网络的众多应用中，节点特征并非一些随意的矢量，而是几何量的坐标。例如，在处理分子图时，节点代表了原子，包含着诸如原子类型、空间三维坐标等信息。我们希望在处理后者的信息时能够按照分子整体进行变换的方式对其进行变换，换言之，对欧氏空间刚体运动群 $E(3)$ （旋转、平移、镜像三类群作用）以及标准的排序等变群均保持等变性。

为便于分析，我们讲节点的特征 $\mathbf{f}_u \in \mathbb{R}^d$ 与节点的坐标 $\mathbf{x}_u \in \mathbb{R}^3$ 进行区分，坐标满足欧式对称群。在这样的设定下，等变层分别作用于这两类输入信息，生成修正的节点特征矢量 \mathbf{f}'_u 以及 \mathbf{x}'_u 。

现在我们就可以根据几何深度学习蓝图介绍一下期待的等变性质。如果输入信息的空间信息（即坐标信息部分，译者注）由欧式变换群 $\mathbf{g} \in E(3)$ 进行了变换（可使用 $\rho(\mathbf{g})\mathbf{x} = \mathbf{Rx} + \mathbf{b}$ 表示，其中 \mathbf{R} 表示旋转及镜像作用矩阵， \mathbf{b} 表示平移变换），输出信息的空间分量 \mathbf{x}'_u 也应当经过同样的变换，但是节点特征信息 \mathbf{f}'_u 保持不变。

正如我们在一般性的图上讨论的排序等变的函数一样，存在众多满足上述约束的欧式变换群 $E(3)$ 等变网络层，但是并不是所有的这些层均满足几何稳定性，或者并非所有的层均易于部署。事实上，事件中易用的等变性可以简单地使用一类来描述，并不像空域图神经网络中的三类不同从层一样。[\[Satorras et al., 2021\]](#) 等人介绍了一种优雅的实现，使用了等变消息传递网络的形式，其模型为：

$$\begin{aligned}\mathbf{f}'_u &= \phi \left(\mathbf{f}_u, \bigotimes_{v \in \mathcal{N}_u} \psi_f(\mathbf{f}_u, \mathbf{f}_v, \|\mathbf{x}_u - \mathbf{x}_v\|^2) \right) \\ \mathbf{x}'_u &= \mathbf{x}_u + \sum_{v \neq u} (\mathbf{x}_u - \mathbf{x}_v) \psi_c(\mathbf{f}_u, \mathbf{f}_v, \|\mathbf{x}_u - \mathbf{x}_v\|^2)\end{aligned}\quad (5-14)$$

其中 ψ_f 与 ψ_c 是两类不同的可学习的函数。可以证明的是，这样的聚合运算在空余分量的欧式变换下是等变的。这是由于特征信息对于节点空间信息的依赖表示为 $\|\mathbf{x}_u - \mathbf{x}_v\|^2$ ，该表示与欧式变换作用无关。此外，这一层的计算也可以视为一种特殊的消息传递的图神经网络，因此他们可以高效地部署。

总结来看，与常规图神经网络相比，[\[Satorras et al., 2021\]](#) 使得正确地对待了图中节点的坐标信息。这些信息经由欧式变换群元素进行变换作用，这意味着网络能够在欧式旋转、镜像、平移作用下表现得正常。而特征信息则以一种逐通道式的方式进行处理，并且假定了其为标量以及其在这些欧式作用下保持不变的特性。这也限制了可以被该架构处理的空域信息的类型。例如，一些特征可能被编码成矢量，例如点的速度，该节点特征的方向会随着欧式变换群的作用而发生改变。[\[Satorras et al., 2021\]](#) 通过在其提出的一个架构的变体种中引入速度的概念来部分地缓解了这一问题。速度是关联与每一点的三维矢量，其随着旋转作用而变换。然而，这仅仅是一个能够被欧式变换群等变网络所学习的一般表示的一个子空间。更一般地，节点特征可被编码为任意维度的张量，其能够随着欧式变换作

用而进行合理的变换。

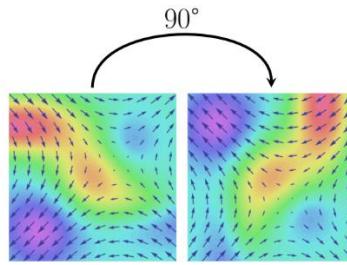


图 5.7 尽管标量场不会随着欧式作用而变换，但矢量场的方向则会随着旋转
变换而发生改变，在此处介绍的架构中并未考虑这一情况

因此尽管上述的架构已经为很多实际的问题提供了一个较为优雅的解决方案，在一些情况下我们仍然希望探索满足等变性的更为广阔的映射集合。现有的试图解决这一问题的方法可以分为两类：(1) 不可规约表示（前述的例子是该类的一个简单实例化）；(2) 正规表示。我们将简要地介绍一下这两类，详细的介绍将在未来的工作中进一步给出。

5.5.1 不可规约表示 (Irreducible Representations)

不可规约表示的理论建立在对任意的旋转-平移群作用可被视为一个不可规约表示的发现之上：其可表示为一个被块状对角矩阵旋转后的矢量。重要的是，该块状对角矩阵种每一个块都是一个 Wigner D-matrix（也就是前述的球状矩阵内积神经网络的傅里叶基底）。署于该类的方法将一个不可规约表示使用等变核函数映射到另一个不可规约表示。为得到全部的等变映射，我们可以直接求解这些核函数上的等变约束。这样求得的结果构成了一个由 Clebsch-Gordan 矩阵以及球状调和函数导出的等变基底矩阵的线性组合。

不可规约表示的早期例子是张量场网络 (Tensor Field Networks, [Thomas et al., 2018]) 以及三维可操控卷积神经网络 (3D Steerable CNNs, [Weiler et al., 2018])，两类卷积模型均作用域点云数据上。SE(3) Transformer 架构[Fuchs et al., 2020] 将这一理念扩展到了图上，使用了基于注意力的层而非是卷积层。此外，尽管我们仅关注了[Satorras et al., 2021] 的一些特殊例子，我们也注意到在一些其他领域的图上旋转或者平移等变预测任务已经有所探索，包括点云的动态图卷积网络 ([Wang et al., 2019b])、量子化学的高效信息传递模型 (SchNet [Schütt et al., 2018]、DimeNet [Klicpera et al., 2020])。

5.5.2 正规表示 (Regular Representations)

尽管不可规约表示看起来很吸引人，但是其直接要求对内含的群表示进行推理表示，这有时是费力的事，并且仅适用于紧致群。正规表示方法则更加通用，但也带来了额外的计算开销，对于特定的等变性其要求存储所有的群元素的潜特征嵌入表示。这一方法实际上出现于卷积神经网络之前。

在正规表示中一个富有希望的方法旨在观察对李群的等变形，并通过指数映射以及对数映射来实现，并可快速地应用于多种对称群。不过本书不对李群做过多的介绍，我们推荐有兴趣的读者阅读该方向最近的两个成功的例子：LieConv[Finzi et al., 2020]、LieTransformer[Hutchinson et al., 2020]。

本节介绍的方法代表着几何图上数据处理的流行方法，并以一种显式的内含几何上的等变性进行处理。正如 4.6 节中讨论的，面片模型是几何图的一个典型例子，其可视为内含的连续曲面的离散化，我们将在下一节深入分析面片模型等变神经网络。

5.6 内参面片模型卷积网络 (Intrinsic Mesh CNNs)

面片模型，尤其是三角面片构成的模型，常被称为计算机图形学的“Bread and Butter”，其也是最常用来对三维物体进行建模的方式。计算机视觉中通用的深度学习以及卷积神经网络的巨大成功在 2010 年前后也吸引了图形学以及几何处理领域的兴趣，这些领域也希望借助类似的处理架构来处理面片模型。

5.6.1 测地线片丁 (Geodesic Patch)

多数深度学习架构均是通过离散化或者逼近指数映射的方式来在面片模型上部署形如式 (4-50) 的卷积滤波器并将滤波器定义在切平面的一个坐标系上。从一个点 $u = \gamma(0)$ 发射一个射线 $\gamma: [0, T] \rightarrow \Omega$ 到邻域点 $v = \gamma(T)$ 就定义了一个测地线局部极坐标系 $(r(u, v), \theta(u, v))$ ，其中 r 与 θ 分别表示测地线距离以及 $\gamma'(0)$ 与局部参考系的转交。这使得定义一个测地线片丁 $x(u, r, \theta) = x(\exp_u \tilde{\omega}(r, \theta))$ ，其中 $\tilde{\omega}_u: [0, R] \times [0, 2\pi] \rightarrow T_u \Omega$ 表示局部极坐标系。



图 5.8 测地线片丁示例，为使得片丁为拓扑盘，其半径应当小于单射半径

在一个用面片模型离散化表示的表面上，一个测地线就是经过三角面片的折线。传统上，测地线一般使用一种非线性偏微分方程的高效的数值计算逼近方法进行计算，称为快速行进算法 (Fast Marching Algorithm, [Kimmel and Sethian, 1998])，该偏微分方程是在

波传播的物理现象中的 Eikonal Equation。这一方法[Kokkinos et al., 2012] 被等人用来计算局部测地线片丁并在随后被[Masci et al., 2015] 用来构建测地线卷积神经网络，这也是第一个面片模型上类似卷积神经网路的架构。

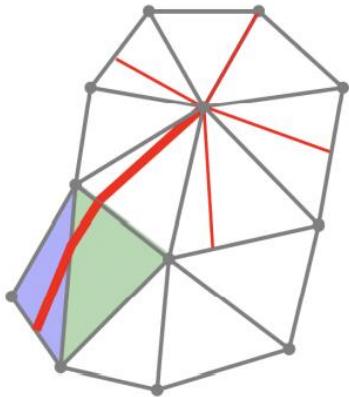


图 5.9 在面片模型上构建离散测地线示例

5.6.2 各向同性滤波器 (Isotropic Filter)

值得一提的是，在测地线片丁的定义中我们并未对参考方向以及片丁的转向进行指定，因此存在不确定性。这也正是度规选择的歧义性，我们的局部坐标系定义在对任意的旋转上，或者一个角度的移动上，这可能使得旋转方向在不同的点不同。也许最简单粗暴的方法就是使用各向同性滤波器 $\theta(r)$ ，其操作应用了对方向无关的邻域特征的聚合函数：

$$(x \star \theta)(u) = \int_0^R \int_0^{2\pi} x(u, r, \theta) \theta(r) dr d\theta \quad (5-15)$$

在 4.4-4.6 节中讨论的谱域滤波器正是这种类型：他们基于了各向同性的拉式算子。然而这一方法由于丢弃了重要的方向性的信息，因而可能在提取类似边缘的特征时失败。

5.6.3 固定化度规 (Fixed Gauge)

我们已经在 4.4 节中提及的可替代各向同性滤波器的方法是使用固定度规。[Monti et al., 2017] 等人使用了主曲率方向，尽管主曲率方向并不是内参信息并且在平面以及曲率相等的表面（例如球面）会引起歧义，作者们展示了其可以合理地应用于人体变形分析，可用来按照逐片刚性的方式来逼近变形前后形体。后续的工作，例如[Melzi et al., 2019] 展示了面片模型上可靠的内参数构建，即通过计算内参函数的梯度实现的。尽管切面场可能具有奇异性（例如在一些点处消失），总体来看整个处理流程对噪声鲁棒并可用于重渲染。

5.6.4 角度聚合 (Angular Pooling)

另一种替代各向同性滤波器的方法被称为角度最大化池化[Masci et al., 2015]。在这一方法中，滤波器 $\theta(r, \theta)$ 是各向异性的，但是其与各个方向的函数进行匹配并聚合：

$$(x \star \theta)(u) = \max_{\theta_0 \in [0, 2\pi)} \int_0^R \int_0^{2\pi} x(u, r, \theta) \theta(r, \theta + \theta_0) dr d\theta \quad (5-16)$$

从概念上来看，这可以视为将测地线片丁与旋转滤波器进行关联并且收集最强的响应。

在面片模型上可以将连续的积分运算使用片丁算子进行离散化表示[Masci et al., 2015]。在一个节点 u 附近的一个测地线片丁以及使用局部极坐标 (r_{uv}, θ_{uv}) 进行表示的邻域 \mathcal{N}_u 一道被加权函数 $w_1(r, \theta), \dots, w_K(r, \theta)$ 加权并聚合（如图 5.10 所示）。计算如下：

$$(x \star \theta)_u = \frac{\sum_{k=1}^K w_k \sum_{v \in \mathcal{N}_u} (r_{uv}, \theta_{uv}) x_v \theta_k}{\sum_{k=1}^K w_k \sum_{v \in \mathcal{N}_u} (r_{uv}, \theta_{uv}) \theta_k} \quad (5-17)$$

此处 $\theta_1, \dots, \theta_K$ 是滤波器可学习的参数。多通道的数据通过一个合适的滤波器组逐通道的方式进行处理。[Masci et al., 2015]、[Boscaini et al., 2016a] 使用了预定义的加权函数 w ，然而[Monti et al., 2017]进一步设其为可学习的参数。

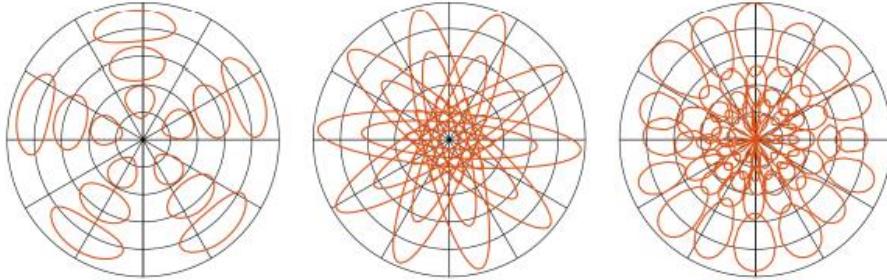


图 5.10 从左至右：测地线卷积网络（Geodesic CNN）片丁算子[Masci et al., 2015]、各向异性卷积网络[Boscaini et al., 2016b]、MoNet[Monti et al., 2017]，图中展示了权重函数 $w_k(r, \theta)$

5.6.5 度规等变滤波器（Gauge-Equivariant Filter）

各向同性滤波器以及角度最大化池化方法均使得其对度规变换保持不变性，他们在单位变换 $\rho(\mathbf{g}) = 1$ 的作用下进行变换（其中 $\mathbf{g} \in \text{SO}(2)$ 是局部坐标系的旋转变换）。这一观点也引出了前述 4.5 节中提到过的另一种方法[Cohen et al., 2019]、[de Haan et al., 2020]，其中网络计算出的特征与结构群 \mathfrak{G} 的任意的表示进行关联（例如结构群分别为坐标系的旋转活旋转镜像群 $\text{SO}(2)$ 或者 $\text{O}(2)$ ）。切矢量也随着标准的表示 $\rho(\mathbf{g}) = \mathbf{g}$ 进行变换。另一个例子是，通过同一滤波器的若干个旋转后的拷贝形成的特征矢量在队列旋转作用下循环移动进行变换，这也常被称为循环群 C_n 的正规表示。

正如 4.5 节中讨论的，当处理那些与非单元群变换关联的几何特征时，我们必须在应用滤波操作之前将其平行移动到同一矢量空间。在一个面片模型上，可[de Haan et al.,

[\[2020\]](#) 通过提出的消息传递机制来部署实现。设在面片模型的节点 u 的特征矢量表示为 $\mathbf{x}_u \in \mathbb{R}^d$ ，该特征相对于任意选定的度规，并且随着在度规旋转作用下特殊正交群 $\mathfrak{G} = \text{SO}(2)$ 的元素表示 ρ_{in} 进行变换。类似地，输出的特征矢量 \mathbf{h}_u 是维度为 d' 的矢量并且随着 ρ_{out} 进行变换（该变换可由网络设计者选择）。

类比图神经网络（GNN），我们可以在面片模型上通过向节点 u 的邻域节点 \mathcal{N}_u 发送消息来部署实现度规等变的卷积：

$$\mathbf{h}_u = \Theta_{\text{self}} \mathbf{x}_u + \sum_{v \in \mathcal{N}_u} \Theta_{\text{neigh}}(\vartheta_{uv}) \rho(\mathfrak{g}_{v \rightarrow u}) \mathbf{x}_v \quad (5-18)$$

其中 $\Theta_{\text{self}}, \Theta_{\text{neigh}}(\vartheta_{uv}) \in \mathbb{R}^{d' \times d}$ 均为可学习的滤波矩阵。结构群元素 $\mathfrak{g}_{v \rightarrow u} \in \text{SO}(2)$ 表示从 v 到 u 的平行移动作用，其表示节点之间的相对度规变换，并且可以预计算。群作用可以表示为平行移动矩阵（Transporter Matrix） $\rho(\mathfrak{g}_{v \rightarrow u}) \in \mathbb{R}^{d \times d}$ ，其中 d 为特征维度，而不一定等于2。矩阵 $\Theta_{\text{neigh}}(\vartheta_{uv})$ 依赖于邻域节点 v 相对节点 u 处的参考坐标系方向的夹角 ϑ_{uv} （例如与x轴夹角），因此这种类型的核是各向异性的，不同的邻域节点处理不同。

正如在4.5节中阐述的那样，若要使得 $\mathbf{h}(u)$ 为良性定义的几何量，则其在度规变换下应当按照 $\mathbf{h}(u) \mapsto \rho_{\text{out}}(\mathfrak{g}^{-1}(u))\mathbf{h}(u)$ 。此处滥用一下符号表示，用 ϑ 代表二维旋转变换时，对于任意的 $\vartheta \in \text{SO}(2)$ 的作用，式 $\Theta_{\text{self}} \rho_{\text{in}}(\vartheta) = \rho_{\text{out}}(\vartheta) \Theta_{\text{self}}$ 及 $\Theta_{\text{neigh}}(\vartheta_{uv} - \vartheta) \rho_{\text{in}}(\vartheta) = \rho_{\text{out}}(\vartheta) \Theta_{\text{neigh}}(\vartheta_{uv})$ 自然能够满足良性定义的条件。由于这些约束是线性的，满足约束的矩阵 Θ_{self} 以及矩阵 Θ_{neigh} 的空间也是线性子空间，因此我们可以使用可学习的系数及基核函数来参数化地表示该矩阵：

$$\Theta_{\text{self}} = \sum_i \alpha_i \Theta_{\text{self}}^i, \Theta_{\text{neigh}} = \sum_i \beta_i \Theta_{\text{neigh}}^i \quad (5-19)$$

5.7 循环神经网络 RNN

我们的讨论事实上总是假定输入为给定域上的空域特征。然而在众多的情况下，输入可以被视为是序列性的（例如视频流、文字段落、演讲稿等），在这样的情况下，我们假定输入包含着任意的若干步，在一个时间步 t 有一个输入信号，表示为 $\mathbf{X}^{(t)} \in \mathcal{X}(\Omega^{(t)})$ 。其中域是静态固定的还是动态变换的取决于时间尺度，例如道路表示的域在小的时间尺度下可视为静态的，但在大的时间尺度下由于道路发生维修、改建等，变为动态的域。类似地在社交网络中，用户交互的变化通常也比交互网络域要频繁的多，此时也可将域视为静态的。

尽管一般情况下域会随着输入信号的变化而发生着缓慢的变化，通常假定域在所有时间步中为固定不变的，即 $\Omega^{(t)} = \Omega$ 。此处我们也仅考虑这种情况，但也应注意例外的情况也是随处可见的。社交网络就是一个典型例子，在该网络中我们总是需要考虑随着时间变换网络本身也在发生变化，新的社交关系在不断新建，旧的社交连接可能被消除。这样的域的设定通常被称为动态图（Dynamic Graph）[\[Xu et al., 2020a\]](#)、[\[Rossi et al., 2020\]](#)。

通常每一个输入特征 $\mathbf{X}^{(t)}$ 会表现出有用的对称性，因此可以被前述所表述的架构有效

处理。这样的例子包括：视频流（域为固定，信号为序列性的帧）、fMRI 扫描（域为固定的面片模型，用于表示大脑皮层的几何结构，不同的区域会在不同的刺激下点亮）、道路交通流网络（域固定，表示道路网络，在每一个节点中记录了平均流量）等。

假定一个编码函数 $f(\mathbf{X}^{(t)})$ 可将输入信号变为合适粒度的潜层表示，并且符合输入域的对称性。以处理视频流为例，在每一个时间步中，我们给定一个网格状的输入信号，表示为一个 $n \times d$ 的矩阵 $\mathbf{X}^{(t)}$ ，其中 n 表示像素点的个数， d 表示通道数（对于 RGB 图像为 3）。此外，我们还对整体所有时间步的帧进行分析，在这样的情况下，我们的编码函数应当是平移不变性的 CNN 网络，每一步均输出一个 k 维的表示 $\mathbf{z}^{(t)} = f(\mathbf{X}^{(t)})$ 。在这一例子中我们并没有丢掉一般性，等变形可以通过分析时空域图上节点的输出来进行，不同的设定的区别仅在于函数 f 的选择。

现在我们的任务只剩下将所有的时间步的潜层表示收集起来。一种规范性的方法是动态地聚合这些信息，聚合时应当保有输入的时间进程信息以及允许轻易地处理在线到达的新数据帧，实践中可通过循环神经网络（Recurrent Neural Networks, RNN）来实现。潜层表示 $\mathbf{z}^{(t)}$ 可视为时间序列上的节点，因此使用 CNN 对其处理在某些应用中也是可行的，Transformer 也是处理这类信息的日渐流行的架构。我们在此处将说明的是，RNN 具有有趣的几何结构，这时由于他们部署了一个对输入信息 $\mathbf{z}^{(t)}$ （注意此处指聚合时的输入，因而维潜层表示，译者注）不平常的对称类型。

5.7.1 简单 RNNs

在每一个时间步中，RNN 计算一个对所有输入直到 t 的时间步的 m 维聚合矢量 $\mathbf{h}^{(t)}$ ，这一部分聚合是通过过去的聚合表示与当前的节点表示的共同计算来实现的，计算映射维：
 $R: \mathbb{R}^k \times \mathbb{R}^m \rightarrow \mathbb{R}^m$ ，计算为：

$$\mathbf{h}^{(t)} = R(\mathbf{z}^{(t)}, \mathbf{h}^{(t-1)}) \quad (5-20)$$

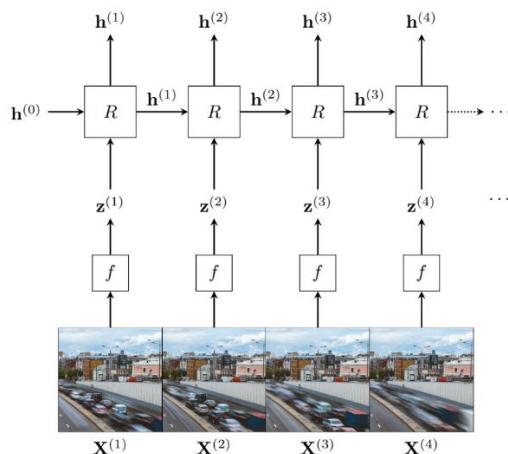


图 5.11 RNN 视频流处理。每一个时间步输入帧通过 f 进行处理，例如平移不变的卷积网络，处理为 $\mathbf{z}^{(t)}$ ，再经聚合作用表为 $\mathbf{h}^{(t)}$

由于 $\mathbf{z}^{(t)}, \mathbf{h}^{(t-1)}$ 均为矢量表示，因此 R 可部署为一个简单全连接神经网络层（也被称为简单 RNN，SimpleRNN[Elman., 1990]、[Jordan, 1997]）。尽管名字中带着“简单”，但其也拥有强大的函数模拟能力，例如[Siegelmann and Sontag, 1995] 等人证明了该模型是图灵完备（Turing-Complete），意味着其可以模拟计算机上任意的运算。该简单 RNN 计算为：

$$\mathbf{h}^{(t)} = \sigma(\mathbf{W}\mathbf{z}^{(t)} + \mathbf{U}\mathbf{h}^{(t-1)} + \mathbf{b}) \quad (5-21)$$

其中 $\mathbf{W} \in \mathbb{R}^{k \times m}$, $\mathbf{U} \in \mathbb{R}^{m \times m}$ 及 $\mathbf{b} \in \mathbb{R}^m$ 均为可学习的参数， σ 为激活函数。尽管这在计算图中引入了循环，实践中网络通常展开为若干合适的步，使得后向传播得以计算 [Robinson and Fallside, 1987]、[Werbos, 1988]、[Mozer, 1989]。

聚合后的矢量就可以被后续下游的任务合理利用，例如若每一个时间步中均需要进行某种预测，那么一个共享的判别器就可以用来对每一个 $\mathbf{h}^{(t)}$ 进行判别，对于将整个序列进行分类的任务，典型的方案是应用最后聚合形成的表示 $\mathbf{h}^{(T)}$ 并将该矢量输入一个分类器中，其中 T 表示序列长度。

特殊地，其实的聚合表示被设置为 0 矢量，也即 $\mathbf{h}^{(0)} = 0$ ，或者也可以设为可学习的参数。研究初始的参数设置也使得我们可以推理出 RNN 网络所呈现的平移等变性。5.7.2 节马上对此进行介绍。

5.7.2 RNN 平移等变群作用

由于我们把每一步视为离散的时间步，那么输入特征潜层表示 $\mathbf{z}^{(t)}$ 可以视为一维网格上的节点。值得注意的是这种设定可扩展至高维的网格中，使得我们可以按照扫描线的方式处理图像上的信号，这样的设定也催生了一些流行的架构，例如[Van den Oord et al., 2016b]。尽管将前述在卷积网络上讨论的平移等变性扩展至 RNN 中是极具吸引力的，这却难以直观地实现。

为展示原因，假定我们通过左移我们的序列一个时间步处理得到了一个新的帧潜层表示 $\mathbf{z}'^{(t)} = \mathbf{z}^{(t+1)}$ ，由于平移等变形，我们希望看到 $\mathbf{h}'^{(t)} = \mathbf{h}^{(t+1)}$ ，然而这一式子在一般情况下并不成立。考虑 $t=1$ ，应用更新函数以及将其展开，有：

$$\mathbf{h}'^{(1)} = R(\mathbf{z}'^{(1)}, \mathbf{h}^{(0)}) = R(\mathbf{z}^{(2)}, \mathbf{h}^{(0)}) \quad (5-22)$$

$$\mathbf{h}^{(2)} = R(\mathbf{z}^{(2)}, \mathbf{h}^{(1)}) = R(\mathbf{z}^{(2)}, R(\mathbf{z}^{(1)}, \mathbf{h}^{(0)})) \quad (5-23)$$

因而除非我们保证 $\mathbf{h}^{(0)} = R(\mathbf{z}^{(1)}, \mathbf{h}^{(0)})$ ，否则我们将无法保证平移等变形。在时间步 $t > 1$ 也可得到类似的结论。

幸运的是，将潜层表示稍作变形，以及选择合适的聚合函数 R ，平移等变性的等式可以得到满足，因此可以展示 RNN 满足平移等变性的设定。问题在于边界条件：上述等式包含 $\mathbf{z}^{(1)}$ ，但平移操作使得其不存在。为便于讨论一般情况，使用抽象的记号：

$$\bar{\mathbf{z}}^{(t)} = \begin{cases} 0 & t \leq t' \\ \mathbf{z}^{(t-t')} & t > t' \end{cases} \quad (5-24)$$

这样的一个序列允许左移 t' 时间步，并且不会破坏原始输入潜层表示。注意若我们对于 $t \leq t'$ 使用不同于0的矢量表示时应当分析等变性。此外，我们无需专门处理右移的情况，实际上，根据RNN的方程右移等变形是自然成立的。

现在可以分析在左移操作下RNN的表示 $\bar{\mathbf{z}}^{(t)}$ 的情况，我们表示为 $\bar{\mathbf{z}}^{(t)} = \bar{\mathbf{z}}^{(t+1)}$ ，正如在式(5-22)及式(5-23)中计算的，有：

$$\mathbf{h}'^{(1)} = R(\bar{\mathbf{z}}^{(1)}, \mathbf{h}^{(0)}) = R(\bar{\mathbf{z}}^{(2)}, \mathbf{h}^{(0)}) \quad (5-25)$$

$$\mathbf{h}^{(2)} = R(\bar{\mathbf{z}}^{(2)}, \mathbf{h}^{(1)}) = R(\bar{\mathbf{z}}^{(2)}, R(\bar{\mathbf{z}}^{(1)}, \mathbf{h}^{(0)})) = R(\bar{\mathbf{z}}^{(2)}, R(0, \mathbf{h}^{(0)})) \quad (5-26)$$

其中当 $t' \geq 1$ 时 $\bar{\mathbf{z}}^{(1)} = 0$ ，也即任意的补足位（对于 $t \leq t'$ 时 $\bar{\mathbf{z}}^{(t)}$ 的取值称为补足位）均可以应用。现在我们可以保证左移一步操作的等变形，这只要 $\mathbf{h}^{(0)} = R(0, \mathbf{h}^{(0)})$ 成立即可，对于左移 s 的情况只要 $t' \geq s$ 即可保证成立。

换言之，初值 $\mathbf{h}^{(0)}$ 必须选为函数的不动点，也即 $\gamma(\mathbf{h}) = R(0, \mathbf{h})$ 。如果更新方程选择得当，我们不仅可以保证不动点的存在，也可以直接迭代得到他们：

$$\mathbf{h}_0 = 0 \quad \mathbf{h}_{k+1} = \gamma(\mathbf{h}_k) \quad (5-27)$$

其中索引 k 代表了迭代计算的轮数，这与上边用 t 表示时间步不同。如果我们选择函数 R 使得 γ 为收缩映射（Contraction Mapping），这样的迭代会收敛到唯一的点，由此，我们就可以利用式(5-27)，直到 $\mathbf{h}_{k+1} = \mathbf{h}_k$ ，这时我们可设定 $\mathbf{h}^{(0)} = \mathbf{h}_k$ 。注意这一计算等价于将序列左移并补充足够的0的变换。

收缩映射为函数 $\gamma: \mathcal{X} \rightarrow \mathcal{X}$ ，满足对于某种范数 $\|\cdot\|$ ，应用 γ 时范数将收缩，意即对于任意的 $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ ，存在 $q \in [0, 1)$ ，满足 $\|\gamma(\mathbf{x}) - \gamma(\mathbf{y})\| \leq q \|\mathbf{x} - \mathbf{y}\|$ 。根据巴拿赫不动点定理迭代这样的函数将收敛到一个唯一的点上[Banach, 1922]。

5.7.3 RNN 深度

堆叠若干层RNN也是可以实现的，这可通过将每一时间步得到的表示 $\mathbf{h}^{(t)}$ 作为输入信息传到下一层RNN。这种结构的构建称为深度RNN（“Deep RNN”），当然这样的叫法可能会引起歧义（这是因为可能和单个RNN但是具有非常多的层混淆，译者注）。由于循环操作的重复应用，即使简单的RNN层也将有着和输入时间步相等的深度。

这样的深度RNN在学习动态中进行优化时引入特有的挑战，这时由于每一个训练例子均对更新网络的共享参数引入了大量的梯度更新。因此我们将聚焦于最主要的问题，也就是梯度消失（Vanishing）和梯度爆炸（Exploding）问题[Bengio et al., 1994]，由于其深度以及参数共享，这两个问题在RNN网络中尤其棘手。此外，这也激起了针对RNN的一些重要的研究成果，对于想要了解更多的读者，我们推荐阅读[Pascanu et al., 2013]，其细致地研究了训练RNN网络的动态，并且展示了多方面的困难：解析方面、几何方面以及动态系统的长度方面。

为展示梯度消失问题，考虑使用了sigmoidal激活函数类型的简单RNN，其激活函数

导函数取值在 0 到 1 之间。乘以很多这样的数将导致梯度变为 0，意味着输入序列中早期的数据将对网络参数的更新完全不起作用。

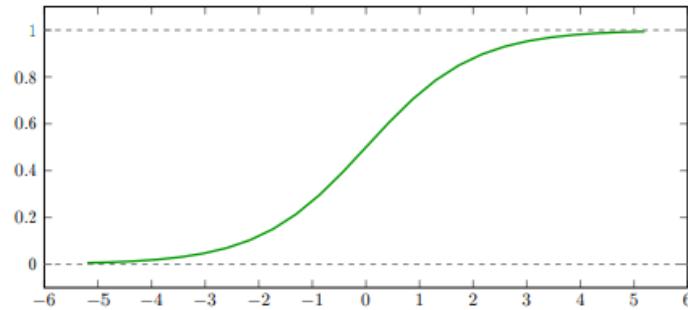


图 5.12 sigmoid 函数包括 logistic 函数以及 tanh 函数，其均具有 S 型曲线

例如，考虑下一个对序列中单词预测的任务（通常用在预测关键字等任务中），其输入为“Petar is Serbian, he was born on ...[long paragraph]...Petar currently lives in __”。在此处，预测下一个单词为“Serbia”尽可能通过对序列中开头位置进行推理得到，其梯度可能消失了，因而对于学习这类的任务而言十分困难。

深度前向神经网络也深受梯度消失的困扰，直到 ReLU 激活函数（其梯度等于 0 或者 1，因此不存在梯度消失问题）的发明，这一问题才得到初步解决。然而在 RNN 中直接应用 ReLU 激活函数很有可能导致梯度爆炸。这是因为更新函数的输出空间是无界的了，梯度下降会为每一个输入步骤进行更新，从而快速地提高了更新的尺度。从历史来看，梯度消失现象早期被视为应用循环网络的一大绊脚石。为解决这一问题促进了更加复杂的 RNN 层的出现，我们将在下面进行介绍。

5.8 长短时记忆网络 LSTM

RNN 中一个关键性的降低梯度消失效应的发明就是门控机制，该机制使得网络可以选择性地以数据驱动的方式重写信息。流行的门控 RNN 包括长短时记忆网络（Long-Short Term Memory, LSTM）[\[Hochreiter and Schmidhuber, 1997\]](#)、门控循环单元（Gated Recurrent Unit, GRU）[\[Cho et al., 2014\]](#)。在此我们主要介绍 LSTM 的变体形式[\[Graves, 2013\]](#)，以便于呈现这样的模型的操作流程。LSTM 中的概念也同样适用于其他门控 RNN 中。

在本节中，读者可参见图 5.13 的结构，其中展现了我们讨论的 LSTM 的操作。

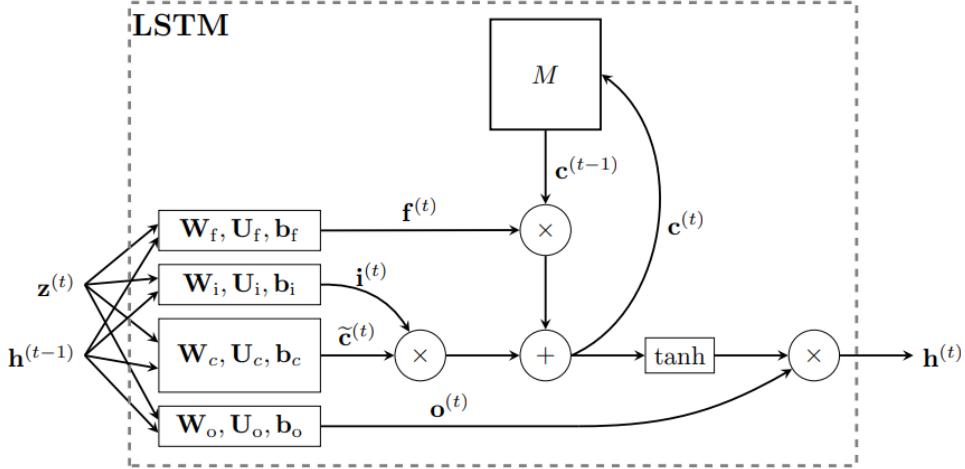


图 5.13 LSTM 结构，其中 M 为记忆单元（Memory Cell），基于现有输入 $z^{(t)}$ 以及上一时间步表示 $h^{(t-1)}$ 以及记忆单元 $c^{(t-1)}$ ，LSTM 预测更新记忆单元 $c^{(t)}$ 以及表示 $h^{(t)}$

LSTM 通过引入记忆单元（Memory Cell）来增强了循环计算，记忆单元存储了状态矢量 $c^{(t)} \in \mathbb{R}^m$ ，其在计算时将被保留。LSTM 网络直接基于 $c^{(t)}$ 计算表示 $h^{(t)}$ ，而记忆单元的状态表示 $c^{(t)}$ 则是根据 $z^{(t)}, h^{(t-1)}, c^{(t-1)}$ 来计算的。关键的是，记忆单元并非全部被 $z^{(t)}, h^{(t-1)}$ 重写，这避免了简单 RNN 中面临的远程依赖。相反，记忆单元中过去的某些状态被保留着，而被保留的部分则是从数据中学习到的。

正像简单 RNN 中，我们应用一个全连接神经网络层根据当前时间步输入以及过去聚合来计算新的特征聚合表示：

$$\tilde{c}^{(t)} = \tanh(\mathbf{W}_c z^{(t)} + \mathbf{U}_c h^{(t-1)} + \mathbf{b}_c) \quad (5-28)$$

注意此处我们使用了 \tanh 激活函数，由于 LSTM 设计上已经减缓了梯度消失问题，因此可以使用 sigmoidal 类型的激活函数了。

但是正如前述提及到的，我们并不允许所有的该矢量进入记忆单元，这也是为什么我们将该矢量称为特征候选，并使用了带有波浪线的表示 $\tilde{c}^{(t)}$ 。相反，LSTM 直接学习门控矢量，该矢量元素是取值为 $[0, 1]$ 之间的实数，决定了信号被允许进入、退出以及重写记忆单元的程度。

总共需要计算三个这样的门控，这三个均基于 $z^{(t)}, h^{(t-1)}$ 来计算。第一个门控为输入门 $i^{(t)}$ ，其计算了特征候选矢量中被允许进入记忆单元的比例；第二个门为遗忘门 $f^{(t)}$ ，其计算了过去记忆单元状态被保留的比例；第三个门为输出门 $o^{(t)}$ ，其计算的新记忆单元状态矢量被用于最后的求和矢量的比例。典型地这三个门均使用单个全连接层，以及 logistic 激活函数，以便于保证输出处于 $[0, 1]$ 之间。注意这三个门均用 m 维矢量表示，他们用来决定每一个维度通过门的程度。三个门的更新为：

$$i^{(t)} = \text{logistic}(\mathbf{W}_i z^{(t)} + \mathbf{U}_i h^{(t-1)} + \mathbf{b}_i) \quad (5-29)$$

$$f^{(t)} = \text{logistic}(\mathbf{W}_f z^{(t)} + \mathbf{U}_f h^{(t-1)} + \mathbf{b}_f) \quad (5-30)$$

$$\mathbf{o}^{(t)} = \text{logistic}(\mathbf{W}_o \mathbf{z}^{(t)} + \mathbf{U}_o \mathbf{h}^{(t-1)} + \mathbf{b}_o) \quad (5-31)$$

最后，这些门被合理地应用于解码新的记忆单元 $\mathbf{c}^{(t)}$ ，然后被输出门模块化以产生聚合表示 $\mathbf{h}^{(t)}$ ，如下：

$$\mathbf{c}^{(t)} = \mathbf{i}^{(t)} \odot \tilde{\mathbf{c}}^{(t)} + \mathbf{f}^{(t)} \odot \mathbf{c}^{(t-1)} \quad (5-32)$$

$$\mathbf{h}^{(t)} = \mathbf{o}^{(t)} \odot \tanh(\mathbf{c}^{(t)}) \quad (5-33)$$

其中 \odot 表示逐元素相乘。应用式 (5-28) - 式 (5-36) 完全明确定义了 LSTM 的更新计算公式，现在同时考虑记忆单元的状态 $\mathbf{c}^{(t)}$ ，有：

$$(\mathbf{h}^{(t)}, \mathbf{c}^{(t)}) = R(\mathbf{z}^{(t)}, (\mathbf{h}^{(t-1)}, \mathbf{c}^{(t-1)})) \quad (5-34)$$

这种构建方式仍然与式 (5-20) 兼容，因为将聚合函数考虑为表示 $\mathbf{h}^{(t)}$ 与记忆单元的连接聚合（即直接将两个矢量放在一起形成维度为原矢量维度和的矢量），有时用 $\mathbf{h}^{(t)} \parallel \mathbf{c}^{(t)}$ 表示。

注意由于遗忘门元素直接由 $\mathbf{z}^{(t)}, \mathbf{h}^{(t-1)}$ 来计算，因此可以从数据中学习，导致 LSTM 可直接学习如何合理地遗忘过去的经验。实际上，遗忘门 $\mathbf{f}^{(t)}$ 的数据直接出现在 LSTM 的后向传播参数更新中，允许网络可以以数据驱动的方式显式地控制不同时间步的梯度消失。

除了处理梯度消失问题，门控 RNN 也催生了另一种非常有用的不变性，即对时间扭曲变换的不变性，这在简单 RNN 中没有出现。

5.8.1 门控 RNN 的时间扭曲不变性

我们首先用连续时间的设定来展示什么叫做时间扭曲，以及为实现对时间扭曲这样的变换保持不变性时循环神经网络需要满足怎样的条件。连续时间是设定对于时间操纵来说更加方便，因而我们采用这样的设定。我们的介绍大体上是根据 [Tallec and Ollivier, 2018] 的工作来展开的，他们最早谈论了这一现象，实际上他们也是最早从不变性角度研究 RNN 的学者。

假定连续时间域的信号 $z(t)$ ，我们将有这样的信号输入 RNN 中。为参考离散设定时的聚合表示 $\mathbf{h}^{(t)}$ 的计算，并类比到连续时间域上，用 $h(t)$ 表示连续时间域上时间 t 的聚合表示，并注意到如下泰勒展开：

$$h(t + \delta) \approx h(t) + \delta \frac{dh(t)}{dt} \quad (5-35)$$

将 δ 设为 1 时得到 $h(t)$ 与 $h(t+1)$ 之间的关系，这正是式 (5-20) 中更新方程定义的。也就是说，RNN 更新函数满足如下微分方程：

$$\frac{dh(t)}{dt} = h(t+1) - h(t) = R(z(t+1), h(t)) - h(t) \quad (5-36)$$

我们希望 RNN 能够对信号采样的方式（例如改变采样的时间间隔）保持稳定，以便于应对信号的缺陷或者异常。正式地，我们使用 τ 表示时间扭曲 $\tau: \mathbb{R}^+ \rightarrow \mathbb{R}^+$ ，其中任意的单调递增并且可微分的映射均可作为时间扭曲映射，选用 τ 来表示时间扭曲的原因在于是

时间扭曲代表着时间的自同构。

这样的时间扭曲可能是非常简单的，例如 $\tau(t) = 0.7t$ ，如图 5.14 中所示，在离散的设定中，该时间扭曲相当于每 1.43 个时间步接收到一个信号。然而时间扭曲使得可调的采样速度。

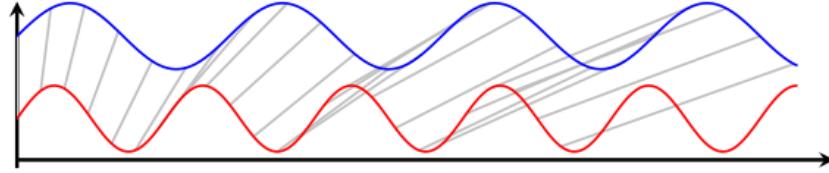


图 5.14 $\tau(t) = 0.7t$ 时的时间扭曲，红色为扭曲后

此外，我们称对于某类型的模型及任意的时间扭曲，存在另一个该类型的模型处理扭曲后的数据与原模型对于未扭曲的数据的方式一样，这种现象称为某类模型对时间扭曲的不变性。

这是一个极具潜在价值的性质。如果我们有一个能够对短程依赖建模的 RNN，并且我们也可以展示其对时间扭曲保持不变性，那么我们知道可以训练这个模型以应对长程依赖关系（由于他们与时间的膨胀扭曲的信号的短程依赖有关）。我们将马上看到，门控 RNN，诸如 LSTM 能够处理长程依赖并非偶然。实现时间扭曲不变性正是与门控机制密切相关，例如使用了输入门、遗忘门、输出门的 LSTM 模型。

当时间域被扭曲函数 τ 作用后，在时间步 t 被 RNN 观察的信号变为 $z(\tau(t))$ ，为保持对时间扭曲的不变性，模型应当预测等变扭曲的聚合函数 $h(\tau(t))$ ，使用泰勒展开，我们可以推导式 (5-36) 时间扭曲的版本，及 RNN 的更新函数 R 应当满足：

$$\frac{dh(\tau(t))}{d\tau(t)} = R(z(\tau(t+1)), h(\tau(t))) - h(\tau(t)) \quad (5-37)$$

然而，上式是对扭曲后的时间进行求导的，因而没有考虑原有的信号。为使我们的模型显式地考虑扭曲变换，我们需要对时间 t 的扭曲聚合函数的导数，应用链式法则，得到：

$$\frac{dh(\tau(t))}{dt} = \frac{dh(\tau(t))}{d\tau(t)} \frac{d\tau(t)}{dt} = \frac{d\tau(t)}{dt} R(z(\tau(t+1)), h(\tau(t))) - \frac{d\tau(t)}{dt} h(\tau(t)) \quad (5-38)$$

为保证对于连续时间的任意时间扭曲均保持不变性，时间扭曲导数应当明确地表示，并且其并不是假定实现知道的。我们引入一个可学习的函数 Γ 来逼近表示该导数。例如 Γ 可以为一个考虑 $z(t+1)$ 及 $h(t)$ 的神经网络，预测一个标量输出。

现在考虑离散的情况。在离散的 RNN 模型中，输入 $\mathbf{z}^{(t)}$ 对应于 $z(\tau(t))$ ，聚合表示 $\mathbf{h}^{(t)}$ 对应于 $h(\tau(t))$ 。为得到 $\mathbf{h}^{(t)}$ 与 $\mathbf{h}^{(t+1)}$ 的关系并同时保持对时间扭曲的不变性，我们使用泰勒展开：

$$h(\tau(t+\delta)) \approx h(\tau(t)) + \delta \frac{dh(\tau(t))}{dt} \quad (5-39)$$

并且设 $\delta=1$, 并且用式 (5-38) 来做变量替换, 有:

$$\begin{aligned}\mathbf{h}^{(t+1)} &= \mathbf{h}^{(t)} + \frac{d\tau(t)}{dt} R(\mathbf{z}^{(t+1)}, \mathbf{h}^{(t)}) - \frac{d\tau(t)}{dt} \mathbf{h}^{(t)} \\ &= \frac{d\tau(t)}{dt} R(\mathbf{z}^{(t+1)}, \mathbf{h}^{(t)}) + \left(1 - \frac{d\tau(t)}{dt}\right) \mathbf{h}^{(t)}\end{aligned}\quad (5-40)$$

最后我们使用前述的学习函数 Γ 代替时间扭曲导数, 这导出了我们需要的时间扭曲不变性的 RNN 的形式:

$$\mathbf{h}^{(t+1)} = \Gamma(\mathbf{z}^{(t+1)}, \mathbf{h}^{(t)}) R(\mathbf{z}^{(t+1)}, \mathbf{h}^{(t)}) - (1 - \Gamma(\mathbf{z}^{(t+1)}, \mathbf{h}^{(t)})) \mathbf{h}^{(t)} \quad (5-41)$$

我们很容易推导简单 RNN 并不具备时间扭曲不变性, 因为该模型没有式 (5-41) 中的第二项, 相反, 其使用 $R(\mathbf{z}^{(t+1)}, \mathbf{h}^{(t)})$ 来完全重写 $\mathbf{h}^{(t+1)}$, 这相当于假定了 $\Gamma=1$, 换句话说假定了 $\tau(t)=t$ 。

此外, 我们的连续时间 RNN 与离散时间 RNN 的推导是基于泰勒展开的, 而泰勒展开仅在时间扭曲导数比较小时才成立, 也即 $d\tau(t)/dt \lesssim 1$ 。直观的解释为: 若时间扭曲对时间进行了压缩, 使得时间间隔 ($t \rightarrow t+1$) 过大, 以至于中间时间的信息无法采样到, 那么模型当然也就无从学习这些没有采样到的信息了。相反, 任意形式的时间扩张 (在离散时间设定下通过在时间序列中填充 0 信号即可) 均是允许的。

结合我们对于时间扭曲函数必须是单调递增的要求, 也就是导数大于 0 的要求, 我们可以令其逼近函数 Γ 满足 $0 < \Gamma < 1$, 这促使使用 logistic 激活函数来代表 Γ :

$$\Gamma(\mathbf{z}^{(t+1)}, \mathbf{h}^{(t)}) = \text{logistic}(\mathbf{W}_\Gamma \mathbf{z}^{(t+1)} + \mathbf{U}_\Gamma \mathbf{h}^{(t)} + \mathbf{b}_\Gamma) \quad (5-42)$$

这正是 LSTM 门控机制! 与 LSTM 门控主要的区别在于, LSTM 计算门控矢量, 而式 (5-41) 暗示了 Γ 应当为标量。矢量化门控 [Hochreiter, 1991] 允许应用不同的扭曲导数到 $\mathbf{h}^{(t)}$ 的不同维度, 使得同时推理多个时间线成为可能。

在此值得停顿一下总结一下我们做了什么。通过要求我们的 RNN 类具有对时间扭曲不变性, 我们推导了必要的形式, 即式 (5-41), 并且表明了这正是门控 RNN 中实现的。门控的最初角色是准确地逼近扭曲导数。

类的不变性这一概念与前述我们讨论的不变性有所不同, 也就是说, 一旦我们在一个时间扭曲为 $\tau_1(t)$ 的输入上训练了一个门控 RNN, 一般情况下我们不能不加训练直接应用到另一个时间扭曲为 $\tau_2(t)$ 的输入上。存在一种特殊的情况, 可以将一个时间扭曲的与训练模型直接应用到另一个时间扭曲的门控 RNN 上, 这要求 $\tau_2(t) = \alpha \tau_1(t)$, 这时更新函数可保持不变, 仅改变门控即可, 即 $\Gamma_2(\mathbf{z}^{(t+1)}, \mathbf{h}^{(t)}) = \alpha \Gamma_1(\mathbf{z}^{(t+1)}, \mathbf{h}^{(t)})$ 。

相对的, 类不变性仅保证了门控 RNN 对这些不同的输入信号均能够以同样的方式有力地拟合, 但模型之间可能存在着完全不同的参数。也就是说, 不同的有效门控与不同的扭曲导数紧密关联, 这些不同扭曲函数实现也对门控 RNN 的优化提供了有用的处方, 我们将简要介绍一下这一点。

例如我们通常假定感兴趣的依赖范围与信号范围一致, 为 $[T_l, T_h]$ 时间步之间。通过分

析式 (5-38) 的解析解表明, 门控 RNN $\mathbf{h}^{(t)}$ 的遗忘时间特性与 $1/\Gamma(\mathbf{z}^{(t+1)}, \mathbf{h}^{(t)})$ 。因此为使得模型能够有效地记住给定区间的信息, 我们希望门控值控制在 $[1/T_h, 1/T_l]$ 之间。此外, 如果我们假设 $\mathbf{z}^{(t)}, \mathbf{h}^{(t)}$ 均是均值为 0 的分布 (这也是一个正则化的一个常见的副作用 [Ba et al., 2016]), 我们可以假定 $\mathbb{E}(\Gamma(\mathbf{z}^{(t+1)}, \mathbf{h}^{(t)})) \approx \text{logistic}(\mathbf{b}_r)$ 。因此控制门控的偏置矩阵称为了一个控制整个门控值的有力工具, 这一洞见由 [Gers and Schmidhuber et al., 2000] 等人给出, [Jozefowicz et al., 2015] 也从经验上推荐了可将门控偏置矢量设为正的常矢量, 例如 1。

结合这两个观察, 我们可以下结论地说, 可通过初始化设定 $\mathbf{b}_r \sim -\log(\mathcal{U}(T_l, T_h) - 1)$ 来得到门控值在合适的区间, 其中 \mathcal{U} 表示实数的均匀分布。这样的建议被 [Tallec and Ollivier, 2018] 等人命名为时间初始化 (Chrono Initialization), 并且已经被从经验角度分析验证了其对于长程依赖建模的有效提升性能。

5.8.2 RNN 序列-序列学习

应用 RNN 进行计算的历史上突出的一个例子就是序列到序列的翻译任务, 例如自然语言的机器翻译, 领先的序列到序列模型是 [Sutskever et al., 2014] 等人实现的, 其通过将聚合矢量作为输入传到解码器 RNN 中, 输出中的每一个 RNN 块的输出也被下一步用作了输入, 如图 5.15 所示。

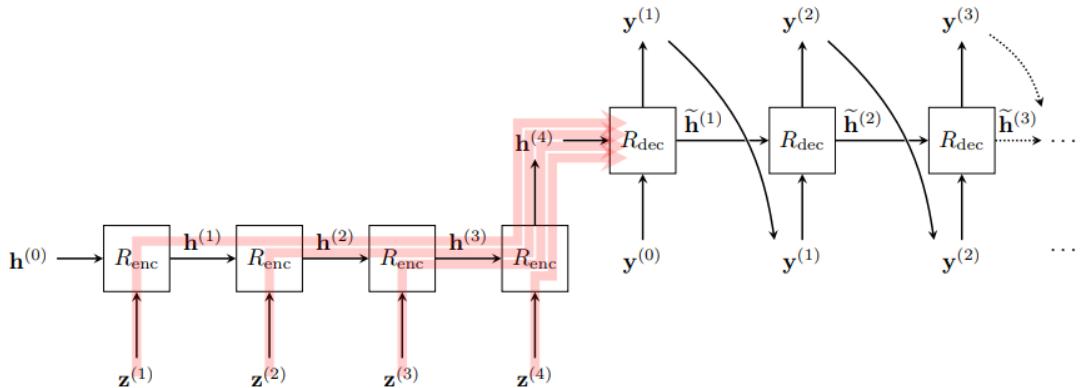


图 5.15 一个典型的序列-序列模型结构, 解码器的初始输入为聚合表示 $\mathbf{h}^{(T)}$, 在解码器中以一种自回归的方式进行处理: 即在每一步中上一解码器单元的输出同时用作下一解码器的输入, 红色线表明了卡脖子的环节, 即聚合表示用以代表所有的输入信息, 当输入序列较长时这尤其具有挑战性

这对于聚合表示而言带来了很大的表示压力。在深度学习的文献中, $\mathbf{h}^{(T)}$ 有时也被称为卡脖子环节 (Bottleneck)。其固定的容量长度必须能够表示输入的整个序列信息, 同时也需要支持输入序列的长度变化。卡脖子环节最近在图表示学习中 [Alon and Yahav, 2020] 以及神经算法推理中 [Cappart et al., 2021] 也得到了重视。

在实际中, 输出的不同部分也许需要关注输入的不同部分上, 这样的问题使用卡脖子

的聚合矢量难以处理。根据这样的观察，流行的循环注意力模型被提出了[Bahdanau et al., 2014]。在处理过程中的每一个步骤中，RNN 生成一个查询矢量，查询矢量进而与每一步的聚合矢量相互作用，主要通过计算加权和的方式进行作用。这一模型领先于基于内容的神经注意力模型并早于 Transformer 模型的大获成功。

最后，尽管注意力机制提供了一种软性的动态聚焦于输入信息的不同部分的能力，一些重要的成果也通过显式方法来导向输入的注意力。一个算法层面已被验证过的有利方法是指针网络（Pointer Network）[Vinyals et al., 2015]，其提出了一个循环注意力网络的改进版本，使得该网络具有指向可变长度输入的不同元素的能力。这些发现进而被扩展到了 set2set 架构[Vinyals et al., 2016] 上，其泛化了序列-序列模型使之支持了无序集合，并使用了指针网络构成的 LSTM 来实现。

6 应用与分析

不变性与对称性在自然界的数 据中随处可见。因此，并不奇怪的是 21 世纪的机器学习的诸多流行应用均来自几何深度学习直接的副产品，也许这些机器学习应用并未意识到这一点。因而我们希望为读者们提供一份综述，当然并非全面的综述，来介绍几何深度学习的重要的研究成果以及一些令人激动或者富有前景的应用。我们的动机可以分为两部分：

(1) 展示本文介绍的 5 种几何域上的科学以及工业界问题的具体例子；(2) 推动几何深度学习原则及架构的未来发展。

6.1 化学及药物研发（Chemistry and Drug Design）

表示学习的一个最具前景的应用领域就是计算化学以及药物研发。传统的要去就是小分子，其被设计为可以在化学上能够与目标分子进行靶向作用，典型的目标分子就是蛋白质，作用的目的就是为了激活或者破坏某种与疾病有关的化学过程。很多药物实际上并非是设计出来的，而是从自然界发现到的，药物的发现历史也反应在了其名字之中，例如乙酰水杨酸（Acetylsalicylic Acid）也被称为阿司匹林（Aspirin），就是柳树（*Salix alba*）的树皮中提取到的，其药用价值在古代已经被发现了。不幸的是，药物研发是一种极其漫长并且成本巨大的过程：在本书写作时，一款药物研发并推向市场经常花费超过十年并且成本超过十亿美元。时间及金钱成本巨大的一个原因就是测试成本过高，众多药物在不同的测试阶段失败了，只有约 5% 的候选药物能够最终走向市场，参见[Gaudel et al., 2020]。

所有化学上可综合的分析形成的可能空间是巨大的（估计约 10^{60} ），因此探索满足正确性质的组合的候选分子，例如满足目标靶向亲和性的分子、低毒性分析、可溶解分子等，这一过程难以通过实验逐个探索，因而虚拟或网络筛查的方法在实践中应用了，例如应用计算方法来寻找具有前景的分子。机器学习算法在这样的任务中正扮演者越来越重要的角色。一个使用几何深度学习实现虚拟药物筛查的突出的例子是[Stokes et al., 2020]，其使用了训练的图神经网络预测备选分子是否阻止了一种细菌，即大肠杆菌的生长，他们有效地发现了一种原用于治疗糖尿病的分子，即 Halicin，具有极具活性的抗生素特性，甚至能够对付具有抗生素抗性的细菌。他们的发现在科学以及流行媒体刊物中广为传播。

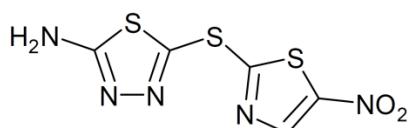


图 6.1 Halicin 分子结构

更广泛地说，应用图神经网络到以图为建模对象的分子上是一个非常活跃的领域，最近也出现了针对该领域任务的多个特定的架构，这些架构结合了物理，例如几何了对于旋转以及平移的等变性[Thomas et al., 2018]、[Anderson et al., 2019]、[Fuchs et al., 2020]、

[Satorras et al., 2021]。此外，[Bapst et al., 2020] 成功地展示了应用 GNN 网络来预测玻璃的动态建模过程，该网络的性能也击败了基于物理的模型，从历史角度看，众多计算化学领域的研究成果均是现代图神经网络架构的前身，他们也有着众多共同的特质。

6.2 药物重定位 (Drug Repositioning)

尽管生成全部的新型药物候选分子是具有潜在可行性的方法，研发新药一个更快也是更为经济的方法是药物重定位，该方法就是评估现有的已经经过验证的药物（或者几种药物的结合）是否可用于某种新的目的。这也极大地降低了推向市场所必须的临床评估工作量。在一定的抽象描述下，药物对于身体的生物化学作用以及它们之间的相互作用以及对其他分子的作用可以建模为图，这也催生了网络药物（“Network Medicine”）的发展，该名词由 Albert-László Barabási 给出，其也在呼吁使用生物网络（例如蛋白-蛋白相互作用和新陈代谢过程）来研发新的处方[Barabási et al., 2011]。

几何深度学习为该类方法提供了一个现代的解决方法。一个早期突出的例子是[Zitnik et al., 2018] 等人的工作，他使用了图神经网络来预测以一种被称为组合处方或者多药治疗的药物重定位为形式的副作用，也即在一个药物-药物图中预测边的特征。在本书写作之时正在流行的新冠疫情，也激发了对于应用类似方法来开发药物以应对 COVID-19 的特别的兴趣[Gysi et al., 2020]。最后我们应当指出，药物重定位并不局限在合成新的分子：[Veselkov et al., 2019] 等人应用类似的方法分析食物中的类似药物的小分子（正如我们提到的，很多事物中也有类似药物分析的生物分子）。本书的作者之一也以创新性的方法参与了该合作研究的一部分，与一位分子研发的负责人一道设计了基于药物类似的分子中富含的超食物的组分的令人激动的处方。

6.3 生物蛋白质 (Protein Biology)

虽然我们已经将蛋白质描述为药物的靶向目标，但是蛋白这一研究对象仍然值得更多关注。蛋白按理说是我们体内最重要的生物分子，其在我们体内有无数功效，包括防护病原体的攻击 (Antibodies)、是我们的皮肤具有特定的结构 (胶原, Collagen)、向细胞输送氧气 (血红蛋白, Haemoglobin)、催化化学反应 (酶, Enzymes) 以及用作信号 (许多荷尔蒙也是蛋白)。从化学角度讲，蛋白质就是生物聚合物，或者说是一系列小的构成单元 (被称为氨基酸, Aminoacids) 在静电力作用下折叠形成三维结构的组合。这样的结构也使得但被具有特定的功能，这对理解蛋白是如何工作的以及它们能做什么是至关重要的 (追溯到诺贝尔奖获得者 Emil Fischer 的一个典型说法就是 Schlüssel-Schloss-Prinzip，也即 “key-lock principle”：当蛋白具有几何上或者化学上互补的结构式他们才能相互作用)。由于蛋白是药物作用常见的靶向目标，药物工业领域对此一直有兴趣。

在蛋白生物信息研究领域的问题的典型层次来自于蛋白序列 (也即一维序列，包含 20 中氨基酸的排序) 到三维结构 (也被称为蛋白折叠) 再到功能 (蛋白功能预测)。最近的研

究方法，例如 DeepMind 提出的 AlphaFold[Senior et al., 2020] 使用了紧致图来表示蛋白结构。[Gligorijevic et al., 2020] 等人展示了应用图神经网络到这样的图上可比仅使用序列方法有更好的功能预测准确度。

[Gainza et al., 2020] 等人提出了一个被称为 MaSIF 的几何深度学习处理管线，从蛋白的三维结构来预测蛋白之间的相互作用。MaSIF 将蛋白建模为分子表面并离散化表示为面片模型，并论证到该表示在处理相互作用时具有优势，因为该表示使得抽象化了内部的折叠结构。该架构基于面片模型的卷积神经网络并作用于预先计算的局部小测地线片丁的化学以及几何特征上。该网络使用了数千个共晶蛋白三维结构进行了训练，该数据来源于 Protein Data Bank，进而处理多种任务，包括预测接口预测、配体分类以及作用位置预测等，并且也允许全新设计蛋白使之原则上能够表现出对癌症的生物免疫的药物。这种蛋白也在编程得到的细胞死亡蛋白复合物之间阻止了蛋白-蛋白之间的作用（Protein-Protein Interaction, PPI），并使得免疫系统有能力攻击癌细胞。

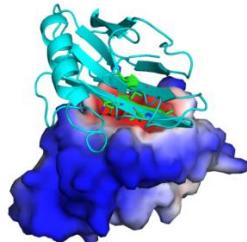


图 6.2 PD-L1 蛋白表面，热力图表明了作用的位置，设计的作用分子为图中条带

6.4 推荐系统及社交网络 (Recommender Systems and Social Networks)

第一个基于图的表示学习的大型且广为流行的应用正是在社交网络中产生，主要就是推荐系统。推荐者的任务就是向用户推荐什么内容，这可能是基于以往客户的交互历史来提供服务。这一问题通常采用边预测目标来解决：监督多个节点的嵌入，并使得若他们有密切关联时（例如他们看的内容很相近）推荐的内容接近。然后这两种嵌入表示的逼近可以视为在内容图中他们被连接在一起的概率，因此对于用户检索的任意的内容，一个简单的方式是推送 k 个相近邻居的嵌入空间的内容。

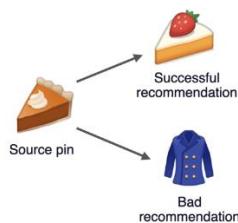


图 6.3 推荐系统的推荐

在这一方法之前是美国图像共享以及社交媒体平台 Pinterest：除了第一个成功在生产

环境部署了 GNN 网络，他们的方法，即 PinSage[Ying et al., 2018] 也成功地使得图表示学习可以应用到百万级节点以及十亿量级的边的任务中。其最后的成果，即 PinnerSage[Pal et al., 2020] 也将用户的特定上下文内容嵌入到了推荐系统中。相关的应用，尤其是在商品推荐领域也都广泛应用于这一方法。流行的 GNN 类型的推荐系统也被广泛部署到了生产环境，包括 Alibaba Aligraph[Zhu et al., 2019]、Amazon P-Companion[Hao et al., 2020] 等。这样，图深度学习每天都在影响着世界上数百万的人们。

在社交网络的内容分析研究中，另一个值得一提的成果就是 Fabula AI，其为第一个被收购的基于 GNN 的初创企业（2019 年被推特收购）。该初创公司是由本书的作者之一极其团队创建，他们发明了新的能够探测社交网络中假消息的方法[Monti et al., 2019]。Fabula 的解决方案是对网络中传播特定消息的用户以及分享该消息的用户进行建模。用户之间若有人重新分享了该信息，则分享者与被分享者就会连接起来，当然他们互相关注时也会建立连接。该图进而作为输入进入一个图神经网络，该网络将整个图分类为真或假，使用了基于事实查验主体提供的约定作为标签。除了能够快速稳定地预测（经常在谣言发生数小时内），分析每一个用户节点的嵌入表示也能揭示那些倾向于分享不实信息的用户群体，正是“Echo Chamber”例子中所描述的那样。

6.5 流量预测 (Traffic Forecasting)

交通网络是几何深度学习可以应用的另一个领域，其应用实际上已经对世界上数十亿的用户造成了积极影响。例如，在一个道路网络中，我们可以将交叉点视为节点，道路段作为连接节点的边，这些边可以使用道路长度、流量或者历史速度等来作为特征描述，或者类似的度量指标也可。

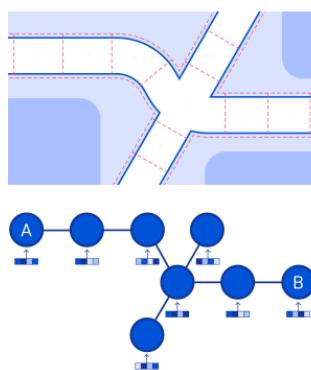


图 6.4 道路的图表示

这样的场景下一个标准的预测任务就是预测到达时间 (Estimated Time of Arrival, ETA)：给定一个选择路径，预测期望的通过该路径的时间。这样的问题也是基本需求，不仅可应用于面向用户的流量推荐应用中，也可用于利用这些预测的企业当中。

图神经网络在这一领域中也展现出了极具前景的价值：例如图神经网络可以针对一个给定道路网络的子图直接应用于预测预期到达时间（这也是一个图回归任务）。这样的方法

被 DeepMind 充分进行了利用，并部署了基于 ETA 预测器的预测系统到 Google Maps 中 [Derrow-Pinion et al., 2021]，并未若干世界上的特大都市提供服务。百度团队也应用了类似的方法，其使用了 ConSTGAT 模型来预测旅行时间，该模型是基于时空域融合的变体图注意力网络模型[Fang et al., 2020]。



图 6.5 Google Maps 中应用图神经网络来进行预测的一些城市，该模型的应用提高了预测质量（例如提高了 40% 的悉尼道路预测质量）

6.6 目标识别（Object Recognition）

机器学习方法的一个主要的校准方法就是看其在给定图像上的分类某些物体的能力。大型的视觉识别挑战赛 ImageNet[Russakovsky et al., 2015] 是每年一次的目标识别的赛事，其推动了几何深度学习的发展。ImageNet 要求模型能够从网络上获得的图片分类到 1000 种类目标的某一类中：这些目标也是多样的，有些是动态有些是静态，并且也具有特殊性，例如众多分类图片要求分类为猫或者狗。因此，在 ImageNet 上取得较好的表现也意味着对于一般的摄影图片有着更好的特征提取，这也形成了来自 ImageNet 的预训练模型的众多迁移学习设定的基础。



图 6.6 输入图像的示例

在 ImageNet 比赛中应用的卷积神经网络的大获成功，尤其是在 2012 年比赛中以较大领先榜的 AlexNet[Krizhevsky et al., 2012]，在很大程度上将深度学习方法带到了台前，在工业界与学术界均是如此。在此之后，卷积网络一直在 ILSVRC 比赛中领先，也衍生了一些流行的架构，例如 VGG-16[Simonyan and Zisserman, 2014]、Inception[Szegedy et al., 2015]、ResNet[He et al., 2016]等，这些架构也超越了人类在同样数据集上的表现。设计原则以及正则化方法也同样应用到了这些架构中，包括使用 ReLU[Nair and Hinton, 2010]、

Dropout[Srivastava et al., 2014]、Skip Connection[He et al., 2016]、以及批量正则化[Ioffe and Szegedy, 2015]等，这些正则化方法形成了现在使用的卷积神经网络的基础。有趣的是VGG使用了16层深度的网络，但作者们称其网络特别深，后续的研究中出现了多大数百层的网络。

与目标识别一道，目标检测领域也出现了重要的研究成果。也就是说，孤立化图片中我们感兴趣的对象所在的区域，并且为感兴趣的对象贴上标签。这样的任务对于下游的许多处理而言都是必要的，从图像的字幕到自动驾驶汽车等。这一任务使得更为精细的方法成为必要，预测也需要局部进行，因此，平移等变性的模型在这样的任务中发挥出了他们的作用。这一领域一些有影响力的例子是使用RNN类的模型，包括[Girshick et al., 2014]、[Girshick et al., 2015]、[Ren et al., 2015]、[He et al., 2017]等，在相关的语义分割中，[Badrinarayanan et al., 2017]模型使用VGG-16架构的编码器-解码器结构证明了其作用。

6.7 游戏（Game Playing）

当观测到的状态可以表示为一个网格域时，卷积神经网络也在增强学习环境中起着主导性的作用，这是由于其平移不变性的特征提取特点，例如可以用到通过图像的每一个像素点来学习玩电脑游戏时，这时卷积网络的任务就是将图像像素点表示为一个矢量化表示，然后这样的表示被应用于策略或者得分函数中并驱动RL主体的动作。尽管增强学习的特定内容并非本书研究重点，对于过去十年中深度学习的一些有影响力的一些成果正是通过构建基于卷积神经网络的增强学习实现的。

一个最值得提到的例子就是DeepMind的AlphaGo[Silver et al., 2016]。其对围棋游戏（Go）当前状态进行编码，使用了 19×19 的网格对其进行卷积神经网络编码，用以表示当前已经下的棋子的位置。然后使用一个来自于过去的专家的下棋位置、蒙特卡洛数搜索、自行对弈等方法得到的组合学习策略，其实现了大师级水平并击败了当今最强的围棋大师李世石，比赛过程也进行了全球直播。



图 6.7 围棋比赛的棋盘，其合法状态约有 2×10^{170} 个[Tromp and Farnebäck, 2006]，比宇宙中所有粒子加一起还要多得多

尽管这已经代表了一个广泛的人工智能的一个重要里程碑（由于围棋相比象棋有着巨大的搜索空间），AlphaGo的研发并未止步于此。该架构的作者们去掉了越来越多的围棋相关的偏置设置，实现了没有人类偏置的AlphaGo Zero[Silver et al., 2017]，并且仅仅使用自

对弈策略来进行学习。AlphaZero 也扩展了可以玩的游戏种类，例如象棋与军棋。最后 MuZero [Schrittwieser et al., 2020] 实现了一个可以在线学习游戏策略的模型，并在 Atari 2600 比赛中实现了很强的表现，同时在围棋、象棋、军棋中也可使用，并且事先不需要任何关于游戏规则的知识。所有这些研究进展，CNN 均是这些重要的架构的输入表示的组成部分。

尽管一些高性能的增强学习主体的研究成果在近些年陆续出现 [Mnih et al., 2015], [Mnih et al., 2016], [Schulman et al., 2017]，在很长一段时间里他们在提供的 57 个游戏中仍然无法达到人类玩家的水平。这一障碍最后被 Agent57 [Badia et al., 2020] 打破了，其使用了参数化族的策略，从强探索性策略到仅仅只有利用性策略，并使得这些策略在不同的训练阶段有着不同的优先级。该架构也在电子游戏的帧缓冲处理中大量使用了卷积网络来做计算。

6.8 文字及音频综合（Text and Speech Synthesis）

除了图片（图片也经常被视为二维网格）之外，一些其他的几何深度学习的架构也在一维的网格上取得了巨大成功。一维网格的自然的例子就是文字和演讲，促使几何深度学习蓝图可以应用到众多不同的领域，例如自然语言处理以及数字信号处理中。

一些广为应用并且公开的该领域成果聚焦于综合：也即或有条件地或无条件地早特定的激励下生成演讲或者文字。这样的设定已经可以支持非常多的有用的处理任务，例如文字转演讲语音（Text-To-Speech, TTS）、预测性文字补全、以及机器翻译等。过去数十年里众多架构已经出现来处理文字以及演讲语音的生成问题，最初仅仅基于 RNN ([Sutskever et al., 2014]) 或者循环注意力网络 ([Bahdanau et al., 2014])。然而，这些架构逐渐被基于卷积神经网络的方案或者基于 Transformer 的方案取代了。

这样的设定下简单的一维卷积网络的一个特殊的局限就是他们线性增长的感受野，这需要很多的层来捕获生成的截至当前所有文字。扩张卷积（Dilated Convolution）在等效参数数量的情况下提供了一个指数级增长的感受野，其也被称为带洞卷积（Trous Convolution or Holed Convolution）。因此，该方案提供了一个较强性能表现的替代性选择，在机器学习任务中最终比 RNN 网络更加具有竞争力了 [Kalchbrenner et al., 2016]，由于其对所有输入的可并行运算，因此进一步降低了计算的复杂度。应用扩张卷积一个最典型的例子就是 WaveNet 架构 [Van den Oord et al., 2016a]。这一架构展示了使用扩张方法，有可能在原始波形式（典型的每秒不少于 16000 采样）的层面实现演讲语音的合成，实现了比以往 TTS 方法的演讲语音更加像人类的演讲语音，并且该架构可以生成小提琴的演奏（实际上该方案在蛋白-蛋白预测中也击败了 RNN 网络） [Deac et al., 2019]。WaveNet 也进一步展示了计算 WaveNet 可以蒸馏为更简单的模型，即 WaveRNN [Kalchbrenner et al., 2018]，该模型有效地在工业界得到了部署。这不仅使得类似于 Google Assistant 这样的语

音助手的语音生成服务的大规模部署使用，并且使得在终端设备上进行计算，例如端到端的加密 Google Duo。

Transformer 架构[Vaswani et al., 2017] 越过了循环网络以及卷积网络的局限性，展示了自注意力机制在机器翻译任务中足以达到最领先水平的性能表现。进而，该架构革命性地重塑了自然语言处理。通过使用诸如 BERT 模型[Devlin et al., 2018] 预先训练的嵌入表示作为输入，Transformer 的计算已经对自然语言处理的下游一系列任务战胜了重要的影响，例如 Google 使用 BERT 嵌入表示来驱动其搜索引擎。

可以说过去几年间 Transformer 最多的公开出版应用就是文字生成了，这主要是来自 OpenAI 的 Generative Pre-Trained Transformer（即 GPT）模型[Radford et al., 2018]、[Radford et al., 2019]、[Brown et al., 2020] 所激发。特殊地，GPT-3[Brown et al., 2020] 使用了 1750 亿个学习参数来训练网络，使其可以成功应用到大规模语言任务中，训练是在网络级别的文字块上进行下一个单词的预测来进行的。这不仅使得 GPT-3 在众多的语言的任务中成为了高度有效的少样本学习器，也是其成为了能够生成与人类语言协调的文字的文字生成器。这样的处理能力不仅仅催化这下游的诸多任务，也涵盖了众多的媒体内容。

6.9 健康保险 (Healthcare)

几何深度学习的另一个富有潜力的应用场景就是在医疗领域中。这些方法可以以多种方式用到这些领域。首先，一些更加传统的架构，例如卷积神经网络，已经用到网格状数据上，例如用来预测在 ICU 中停留时间的长短[Rocheteau et al., 2020]、通过视网膜扫面来预测致命性的疾病[De Fauw et al., 2018] 等。[Winkels and Cohen, 2019] 等人展示了相对于使用传统卷积神经网络，使用对旋转即平移不变的群卷积神经网络可以有效提升肺疣的检测。

第二，将器官建模为几何表面，面片模型卷积神经网络展现出了处理多样任务的能力，从面部结构重建到基因相关信息[Mahdi et al., 2020]，从大脑皮质分割[Cucurull et al., 2018] 到人口皮质表面结构特点退化[Besson et al., 2020]。后一个例子代表了一种神经科学中增长的趋势，即考虑大脑为一个复杂折叠而成的表面，形成了强烈的非欧特性。在一些解剖学文献中这样的结构也称为脑沟 (Sulci) 或脑回 (Guri)。

同时，神经科学家也经常尝试构建及分析大脑的功能性网络来表示大脑的不同区域，进而分析在进行某种认知功能的时候哪些区域被激活了，这样的网络通常使用功能性核磁共振成像 (fMRI) 来构建，该成像能够实时显示大脑的哪些区域消耗的血更多（通常使用血氧水平依赖对比图，即 BOLD）。这些功能性网络揭示了病人的人口学特点（例如可以区分男女，[Arslan et al., 2018]），也可以用来做神经病理学诊断，这也是几何深度学习可以应用的第三个领域。在本文中，[Ktena et al., 2017] 等人使用图神经网络来做神经学上状况的预测，例如自闭症谱系疾病 (Autism Spectrum Disorder)。大脑的几何与功能性机构似乎

是紧密联系在一起的，最近[Itani and Thanou, 2021] 等人在联合神经学病理分析中指出了利用这一点的优势。

第四，在基于机器学习的药物诊断中病人网络变得愈加重要。这些方法背后的基本原理在于，病人口学信息、遗传性、外表相似性可以用来预测他们的疾病。[Parisot et al., 2018] 等人在由神经学病理诊断方面得到的人口学特征进而得到的病人网络上使用图神经网络，表明了使用图可以提高预测的结果。[Cosmo et al., 2020] 等人展示了使用潜图学习（网络学习未知的病人图）的重要用处。后者的研究数据来源于英国 Biobank 数据库，该库也包含大规模的医学数据，例如大脑图像[Miller et al., 2016]。

关于住院病人的大量的数据可在健康电子记录（Electronic Health Records, EHRs）中找到。面向公众开放的关于病人匿名处理的 EHR 数据集包括 MIMIC-III[Johnson et al., 2016] 以及 eICU[Pollard et al., 2018]。处理给出了病人疾病过程的全面的视角，HER 分析也使得可以将类似的病人关联起来。这与模式识别一样，模式识别也常常被用来诊断中。其中，医师使用经验来判断医学特性图像的模式，当医师的经验使得他们能够快速地定位诊断时该方法（指根据经验来做模式判断，译者注）可能是主要的方法。在这一情况下，一些工作试图基于 EHR 数据来构建病人图，这或是通过分析他们的医生的笔记[Malone et al., 2019] 的嵌入表示，诊断相似性[Rocheteau et al., 2021] 等来实现，或是甚至假定一个全连接图[Zhu and Razavian, 2019]。在所有的情况下，使用图表示学习来处理 EHR 数据均显示出来了极具前景的未来。

6.10 粒子物理及天文物理(Particle Physics and Astrophysics)

高能物理学家也许是第一类自然科学领域中使用深度学习这一闪亮工具的人了。在最近的一篇综述中，[Shlomi et al., 2020]等人指出机器学习算法在历史上就已被多次使用到了粒子物理测量中，或是用来学习复杂的函数的逆，并以此根据采集到的信息来推导物理过程，或是用来执行分类或回归任务。对于后者而言，通常需要使得数据以一种非自然的表示方式中，例如网格形式，进而才能被一些标准的深度学习架构所处理，例如卷积神经网络。然而实际上物理中的很多数据都有着无序的形式并且数据间存在这关联与相互作用，这可以自然地表示为图的形式。

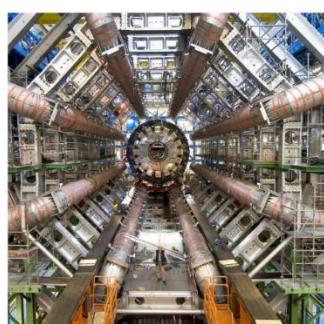


图 6.8 大型强子对撞机图

高能物理中一个重要的应用就是重建或者分类粒子束，这些粒子束源于单一事件导致的粒子衰变或者多源相互作用而形成的稳定的粒子流。在大型强子对撞机中，CERN 建设了最大的也是最知名的粒子加速器，这样的粒子束也是质子在接近光速下对撞而产生的。这些撞击产生了大量的粒子，例如希格斯玻色子以及夸克等。对对撞时间进行识别以及分类是至关重要的任务，其能够提供发现新粒子的实验性证据。

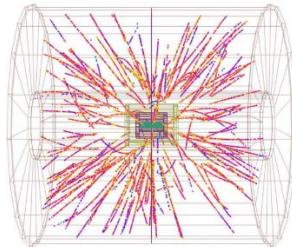


图 6.9 粒子束示意图

针对粒子束的分类任务最近学者们提出了多种几何深度学习的方法，例如[Komiske et al., 2019] 和[Qu and Gouskos, 2019]，分别基于 Deep Set 架构与动态图卷积网络。最近，根据物理方面的原理并结合与汉密尔顿或者拉格朗日方程兼容的归纳偏置进行特定的架构设计也吸引了不少兴趣。例如[Sanchez-Gonzalez et al., 2019] 和[Cranmer et al., 2020]，等效地使用洛伦兹群[Bogatsiy et al., 2020]，或者甚至结合符号推理的架构[Cranmer et al., 2019] 从而实现从数据中学习物理定律。

除了粒子加速器外，粒子探测器也被天文物理学家用作多信使天文（一种新的不同信号观测定位方法），例如来自同一发射源的电磁辐射、引力波以及中微子辐射。中微子天文学获得了额外的研究兴趣，这时由于中微子极少与物质发生相互作用，因而即使经历了很长很长的路程也不会受到影响。探测中微子使得观测那些使用天文光学望远镜无法观测到的星体成为了可能，但也要求使用非常大的尺寸的探测器---IceCube 中微子探测器是建设在南极大西洋冰层上并且使用了数千米的立方体结构。探测高能的中微子将对于认识宇宙提供帮助，例如耀变体和黑洞。等人使用几何深度学习来 IceCube 中微子探测器的不规则外观进行建模，展示了探测来自于外太空的中微子的较好的性能提升并且能够把中微子同背景事件分离开来。来自背景辐射的光学沉积的特性曲线以及来自外太空的源的不同

[Choma et al., 2018]如图 6.10。

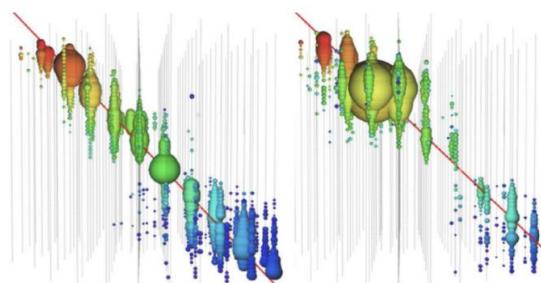


图 6.10 左为背景事件特性曲线，右为外太空中微子辐射特性曲线

尽管天文学中微子为宇宙学的研究提供了无限可能，传统的光学望远镜以及声学设备仍然是天文学家的实用武器。在这些传统的设备的使用中，几何深度学习仍可以为分析其数据提供有力工具。例如[Scaife and Porter, 2021] 等人使用旋转等变形卷积神经网路进行声学上星系的分类，[McEwen et al., 2021] 等人使用球状卷积神经网络来分析宇宙微波背景辐射，该辐射也是宇宙大爆炸的遗物辐射，可能为了解宇宙最初的形态提供帮助。正如我们已经提到的，这样的粒子可在球状几何上自然地表示并且等变神经网络也是分析这样的数据的合适工具。

6.11 VR 及 AR (Virtual and Augmented Reality)

另一个应用领域就是计算机视觉以及计算机图形学了，这也是促进了大量几何深度学习方法的发展的动力，尤其是对于 AR 及 VR 中对于三维形体模型的处理。用于产生诸如 Avatar 中的特效的运动捕捉技术作用于两个阶段：第一，三维扫描传感器将捕捉到形体的运动或者运动者的人脸作为输入信息，并与某一个标准的形体进行关联，典型的标准形体使用离散的流行或者面片模型来建模表示。第二，生成一个新的形体来跟踪输入的运动（也被称为综合，“Synthesis”）。计算机视觉及图形学中最初的几何深度学习相关的工作（[Masci et al., 2015]、[Boscaini et al., 2016a]、[Monti et al., 2017]）是作用于面片模型卷积神经网络中处理分析任务，更具体地说，处理变形形体的关联。

针对三维形体综合的第一个几何自编码器架构是被[Litany et al., 2018] 和[Ranjan et al., 2018] 独立地提出来的，在这样的架构中，人体、脸、或者手臂的标准面片模型假定是已知的，合成的工作包括对三维坐标使用微分几何的方法进行回归分析。[Kulon et al., 2020] 等人展示了三维手臂姿态估计的混合处理管线，该方法基于 CNN 的编码器以及几何的解码器。与英国初创公司 Ariel AI 合作研发并在 CVPR2020 上进行展示的一个该系统的演示表明该系统可以从视频输入流以超实时的方式创建手姿态的实景的形体阿凡达。Ariel AI 被 Snap 于 2020 年收购，在本文写作之时其技术也被应用到了 Snap 的增强现实产品中。



图 6.11 根据二维图片输入进行复杂三维手部姿态重建

7 展望

“根据你的定义对称性可大可小，其是一个人们穷尽一生来理解并构造条理性、美感、追求完美的典范。”（译文，原文为 ““Symmetry, as wide or as narrow as you may define its meaning, is one idea by which man through the ages has tried to comprehend and create order, beauty, and perfection.”），这句话正是伟大的数学家 Hermann Weyl 在其同名书中给出的对称性的诗意般的定义[Weyl, 2015]，该书为其从普林斯顿高等研究院退休前夕的呕心沥血之作。Weyl 研究了对称性在科学与艺术中占据的特殊位置，从苏美尔人的对称性的创作到笃信圆由于其对称性是最完美的形状的毕达哥拉斯学派。柏拉图研究了五种正多面体并认为这些多面体特别重要并且也是其他现实的物质直接的组成基本单元，这些多面体现在以他的名字命名。然而尽管柏拉图给出了专业术语 “”，意为相同的度量，其仅仅从美观上传达了艺术中比例的对称性以及音乐中和谐。天文学家兼数学家开普勒 (Johannes Kepler) 是第一个试图严格证明水分子的对称性的形状的人，在其论文 “On the Six-Cornered Snowflake” 中（改论文全名为 “Strena, Seu De Nive Sexangula”，即 “New Year’s Gift, or on the Six-Cornered Snowflake”，该书是一个其在圣诞节送给朋友 Johannes Matthaeus Wackher von Wackenfels 的小册子礼物），他将雪花中六边形折叠的二面角结构视为粒子的六方晶胞，这一想法也促进了对于物质是如何形成的理解，并且这一论断至今在结晶学研究中仍然成立[Ball, 2011]。

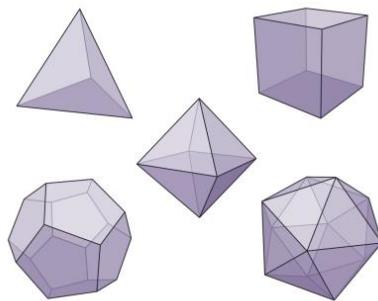


图 7.1 柏拉图立方体

7.1 数学及物理学中对称性 (Symmetry in Mathematics and Physics)

在现代数学中，对称性独家地使用了群论的表述方式。群论的研究起源于 Évariste Galois，1830 年左右他给出了这一命名并且用其来研究多项式函数的可解性问题。其他两个与群论紧密关联的两个名字就是 Sophus Lie 和 Felix Klein 了，他们遇到对方并一同工作了相当长的时间[Tobies, 2019]。Sophus Lie 发展了连续对称性并提出了以其名字命名的李群，Klein 在其 Erlangen Program 中提出群论是几何的主要组织原则，我们在本书开头也曾

提到过这一点。黎曼几何被 Klein 明确地排除在了他所说的统一框架之外，但由于 Élie Cartan 在 1920 年潜的不懈努力，近五十年的时间才将黎曼几何整合进入统一的框架。

Klein 在哥廷根大学的同事 Emmy Noether 证明了一个物理系统中任意的可微对称的作用均有着守恒定理[Noether, 1918]。在物理中，这是一项惊人的发现：在以往的情况下，大家通过细心的实验来观察并总结基本定理，例如能量守恒定律，这些守恒定理均是来自经验而非理论推到或证明，Emmy Noether 定理也被诺贝尔奖获得者 Frank Wilczek 称为 20 世纪到 21 世纪物理的指路明灯，展现出了能量守恒定理来自于时间的平移对称性，这也是一个非常直观的想法：一个实验的结果应当不依赖于什么时候做的实验。

电荷守恒相关的对称性就是电磁场的全局度规不变性，首次出现在麦克斯韦电动力学方程中[Maxwell, 1865]，然而其重要性最初并未被发现。前述提到的 Hermann Weyl 极富激情地在 20 世纪第一个介绍了物理中的度规不变性，强调了度规不变性在电磁场是如何导出的这一问题中的重要作用。Weyl 在 1919 年第一个提出了在尺度或者度规下的变化的不变性正是电磁的局部对称性，这一观点并不正确，但其定义的名词“度规”(Gauge, 德语的 Eich) 获得了广泛使用，该名词是仿照了铁轨的轨道来定义的。在研究了量子机制后，Weyl 修改了度规选择[Weyl, 1929]，将尺度因子改为了波相位的改变量，参见[Straumann, 1996]。

数年后，这一基本的原则被扩展了，也就是杨-米尔斯方程 (Yang-Mills) [Yang and Mills, 1954]，该方程成功地提供了一个统一的电磁的量子力学表现以及弱相互作用与强相互作用的框架，最终发展成熟为标准模型 (Standard Model)，并用以描述除了重力以外的其他任意的力。我们可以使用另一位诺贝尔物理学奖获得者，即 Philip Anderson 的话来总结[Anderson, 1972]，就是“说物理就是研究对称性可能并不为过”(译文，原文 “it is only slightly overstating the case to say that physics is the study of symmetry.”)。

7.2 机器学习早期应用的对称 (Early Use of Symmetry in Machine Learning)

在机器学习及其在模式识别与计算机视觉的应用中，对称性的重要性早已被认可了。早期为模式识别而设计的等变特征探测器由[Amari, 1978]、[Kanatani, 2012]、[Lenz, 1990] 等人给出。Shun’ichi Amari 也被认为将信息几何应用到黎曼几何模型上的开创者，信息几何主要的研究对象就是统计流形，在该流形上任意一个点关联与一个分布。在神经网络的文献中，[Minsky and Papert, 2017] 等提出的著名的感知机群不变定理也指明了单层感知机在学习不变性能力根本的局限性。这也是最初研究多层架构的动力来源[Sejnowski et al., 1986]、[Shawe-Taylor, 1989]、[Shawe-Taylor, 1993]，最终导致了深度学习的出现。

在神经网络社区中，神经认知 (NeoCognition) [Fukushima and Miyake, 1982] 是第一个部署移动不变性的神经网络，实现了不受移动变换影响的模式识别。这一方案使用了层

次化的神经网络结构并具有局部连接性，并利用了神经学家 David Hubel 以及 Torsten Wiesel 在二十多年前的视觉皮层的感受野的启发[Hubel and Wiesel, 1959]（Hubel 等人这一重要的工作也被授予了 1981 年诺贝尔医学奖）。这些想法最终发展成熟为了 Yann LeCun 以及合作者们提出的卷积神经网络[LeCun et al., 1998]。第一个从表示理论角度研究不变性与等变形神经网络的工作由[Wood and Shawe-Taylor, 1996]等人给出，但其工作却少有人引用。这些想法的最近化身包括[Makadia et al., 2007]、[Esteves et al., 2020] 以及本书的一个作者[Cohen and Welling, 2016]。

7.3 图神经网络 (Graph Neural Networks)

定位图神经网络是从什么时间开始的是十分困难的，这也是由于早期的一些工作并没有把图当成首要的研究对象，另一个原因是图神经网络仅仅在 2010 年后才开始能够实际使用，以及图神经网络也是多个研究领域的合流。也就是说，早期的图神经网络的形式至少可以追溯到 1990 年左右，以 Alessandro Sperduti's Labeling RAAM[Sperduti, 1994]、结构后向传播[Goller and Kuchler, 1996]、以及数据结构的适应性处理[Sperduti and Starita, 1997]、[Frascono et al., 1998] 等工作为代表。尽管这些工作最初主要关心在结构上的操作，并用于树或者有向无环图，这些架构中的很多不变性在今天用到的 GNN 模型中仍旧有所体现。

第一个合理地处理一般图结构的架构（并明确了术语“Graph Neural Networks”）出现于 21 世纪之后，在意大利的 Università degli Studi di Siena 大学的人工智能实验室中，Marco Gori 与 Franco Scarselli 第一个提出了“GNN”[Gori et al., 2005]、[Scarselli et al., 2008]。同时 Alessio Micheli 提出了图上神经网络（Neural Network for Graph, NN4G），使用了前向传播网络而非循环网络的方法[Micheli, 2009]。Gori 等人使用了循环机制，要求网络参数满足收缩映射，因而通过计算不动点来计算节点表示，这一方法形成了一种特殊的后向传播并且可以完全不依赖于节点特征[Almeida, 1990]、[Pineda, 1988]。上述的所有问题均被门控 GNN（Gated GNN, GGNN）所修正[Li et al., 2015]。GGNN 吸收应用了许多现代 RNN 的优点，例如门控机制[Cho et al., 2014]，以及随时间的后向传播，因而知道今天依然流行。

7.4 计算化学 (Computational Chemistry)

图神经网络的另一条值得注目的发展线就是来自于计算化学领域的需求驱动的研究，其中分子被表示为图，原子表示节点，化学键表示边。这样的表示促进了应用计算技术直接在这样的结构上进行分子性质的预测，并在 1990 年左右在机器学习领域出现了相关的研究，包括[Kireev, 1995] 和[Baskin et al., 1997]。令人吃惊的是，[Merkwirth and Lengauer, 2005] 提出的分子图网络（Molecular Graph Network）提出了众多在现代的 GNN 中应用的特征，例如边类型约束的权重、全局聚合池化等，而该成果是在 2005 年就已经提出了。化

学上的世纪需求也推动着 GNN 的发展，知道 2010 年，两个旨在提升分子指纹 [Duvenaud et al., 2015] 以及预测量子-化学性质 [Gilmer et al., 2017] 的重要的 GNN 网络提出。在本书写作之时，分子性质预测也是 GNN 研究的重要应用，在虚拟抗生素筛查中也扮演者重要的角色 [Stokes et al., 2020]。

7.5 节点嵌入 (Node Embeddings)

图上的深度学习早期成功的几个架构均与以基于图本身的结构并以无监督学习的方式来学习节点的表示有关。在这样的结构性特点的启发下，这一方向也提供了图表示学习与神经科学社区的关联。在这一领域中关键的一个方法就是基于随机漫步的嵌入表示：如果节点出现在短程的随机漫步时应当将其表示的更近一点。表示学习方法包括 DeepWalk [Perozzi et al., 2017]、node2vec [Grover and Leskovec, 2016]、以及 LINE [Tang et al., 2015] 等，这些均是纯粹的自监督学习。Planetoid 方法 [Yang et al., 2016] 是第一个集成了监督学习标签的方法。

在一些场景下希望将 GNN 的编码器与随机漫步的目标统一起来，这包括使用了诸如变分自编码器 (Variational Graph Autoencoder, VGAE) [Kipf and Welling, 2016b]、嵌入传播 (Embedding Propagation) [García-Durán and Niepert, 2017]、以及 GraphSAGE 的无监督变体 [Hamilton et al., 2017] 等方法的架构。[Srinivasan and Ribeiro, 2019] 等人最近提出了一个理论框架，研究了结构以及位置表示的等变形，此外，[Qiu et al., 2018] 等人展示了基于随机漫步方法的嵌入表示等价于一个合适条件的矩阵分解任务。然而，这产生了一些混合的结果，不久人们发现将邻域节点的特征表示聚传播到本届点正是 GNN 的归纳偏置的关键部分。实际上，有学者指出未经任何训练的 GNN 网络在节点特征已知的情况下已经能够取得比 DeepWalk 更好的性能 [Veličković et al., 2019]、[Wu et al., 2019]。这使得 GNN 的研究方向与随机漫步目标渐行渐远了，并且 GNN 的研究方向走向了由共有信息最大化所启发的对比学习方法，这也与图像域中的成功方法相吻合。这一方向一些重要的成果包括 Deep Graph Informax (DGI) [Veličković et al., 2019]、GRACE [Zhu et al., 2020]、BERT-like Objectives [Hu et al., 2020] 以及 BGRL [Thakoor et al., 2021] 等。

7.6 概率图形学模型 (Probabilistic Graphical Models)

同时，图神经网络也激发了概率图形学模型 (PGM) 的嵌入计算 [Wainwright and Jordan, 2008]。PGM 是处理图形学模型的有力工具，他们的易用性源于对图中的边的概率分析视角，也就是说，节点被视为随机变量，图的结构就编码了条件概率假设，这使得极大地降低了计算量以及可以从联合分布中采样。实际上，众多依赖于 PGM 进行学习以及推理的算法均依赖于从边之间传递消息 [Pearl, 2014]，这样的算法包括变分均值场推理 (Variational Mean-Field Inference) 以及奇异信念传播 (Loopy Belief Propagation) [Yedidia et al., 2001]、[Murphy et al., 2013]。

PGM 与消息传递架构的联系在 GNN 网络中被发展应用，例如等人建立了理论上的关联的 structure2vec 架构[Dai et al., 2016]。也就是说，通过将图表示学习视为马尔可夫随机场，节点关联着输入特征以及潜表示，该架构的作者们直接将均值场的计算与奇异信念传播关联起来，这与常见使用的 GNN 有所不同。

PGM 中使得 GNN 的潜层表示与概率分布关联起来的关键性的技巧就是使用了分布的希尔伯特空间嵌入[Smola et al., 2007]。给定精心挑选的特征 \mathbf{x} 的嵌入表示函数 ϕ ，这使得将其概率分布嵌入表示为期望 $\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \phi(\mathbf{x})$ 成为了可能。这样的关联关系使得我们能够进行类似 GNN 的计算，并且知道 GNN 计算出的表示将于一些节点特征上概率分布的嵌入表示相关联。

structure2vec 模型本身也是一个很容易符合我们的蓝图的 GNN 架构，但其设定也启发了一系列的 GNN 架构，这些架构们直接地与 PGM 中的计算进行整合。这一方向出现了成功将条件随机场与 GNN 结合的架构（[Gao et al., 2019]、[Spalević et al., 2020]）、关联马尔可夫网络[Qu et al., 2019] 以及马尔可夫逻辑网络[Zhang et al., 2020] 等架构。

7.7 Weisfeiler-Lehman 正式化 (The Weisfeiler-Lehman Formalism)

图神经网络的再度兴起与想要理解他们的根本性局限密不可分，尤其是他们的表达能力。GNN 是一种有效对图结构数据进行建模的工具这一事实已得到人们的认可，但人们也意识到其无法完美地解决图上的所有的问题。这一描述的一个规范化的说法就是图的同构：GNN 模型是否能够在给定的两个非同构的图后生成两个不同的表示呢？从两方面来看这均是有用的问题。如果 GNN 模型无法为不同构的图生成不同表示，那么对于任意的想要区分这两个图的任务而言都将束手无策。此外，目前仍然不清楚是否能在多项式时间内确定两个图是否同构，多项式时间也是所有 GNN 模型所需的计算时间开销。对于等距同构的图而言，GNN 总是能够生成一样的表示。目前判断图的同构的算法最好的是由[Babai and Luks, 1983] 等人提出，最近 Babai 提出了可以在准多项式时间来求解[Babai, 2016]，但这一论断尚未得到充分的评议。

判断 GNN 与图同构的重要的框架就是 Weisfeiler-Lehman 图同构测试[Weisfeiler and Lehman, 1968]。这一测试通过迭代地沿着图中的所有边遍历所有节点特征来生成图的表示，然后对邻域的和使用随机哈希算法。这与随机初始化的卷积图神经网络有着明显的关联，并且也被早就观察到了关联关系，例如 GCN 模型[Kipf and Welling, 2016a]。除了关联关系外，WL 迭代早前被[Shervashidze et al., 2011] 等人在图的核函数中引入了，并且他们的成果至今是整个图表示的无监督学习的参考基准。

尽管 WL 测试概念上非常简单，实际中也存在许多其不能有效地区分同构性的图，其表达能力也与 GNN 模型深度绑定了。[Morris et al., 2019] 等人及 [Xu et al., 2018] 等人的

分析得到了令人惊讶的结论：任意的符合我们在 5.3 节中介绍的三种类型的 GNN 网络的表达能力均不比 WL 测试的表达能力强。

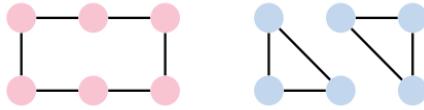


图 7.2 WL 测试无法区分图中两个结构

为了达到 WL 测试这样的表达能力，在 GNN 的更新规则中必须满足若干约束。[Xu et al., 2018] 等人展示了在分立特征域中，GNN 使用的聚合函数一定要是单射的，其和就是一个关键的表示，对于流行的直接求和以及取平均这样的聚合函数就不满足要求了，因为他们无法区分集合 $\{a, b\}, \{a, a, b, b\}$ 。基于这一出色分析，[Xu et al., 2018] 等人提出了图同构网络（Graph Isomorphism Network，GIN），该架构是一个非常简单但达到了 GNN 最大表达能力的模型。其也可以使用卷积形式的 GNN。

最后值得一提的是，这些分析结论对于连续特征空间并不成立。实际上，使用 Borsuk-Ulam 定理[Borsuk, 1933]、[Corso et al., 2020] 已经可以展示，假定节点特征为实数，得到单射的聚合函数要求多个聚合器（特定地要求聚合器数目等于节点的度，一个例子就是邻域多个集合的距）。这些发现也激发了主邻域聚合架构的出现，该架构提出使用多个从表示能力看经验上强大并且稳定的聚合器 GNN。

7.8 高维方法 (High-Order Methods)

前边几段的发现并不与 GNN 的实际用处相矛盾。实际上，在很多现实世界中图结构上的输入节点特征是足以支持有用的区分性的计算的，尽管上述提到的局限性的存在。在区分性的计算中总是考虑无特征或者分类特征的图。

然而一个关键的引理展示了 GNN 在探测图中一些基本的结构时实际能力是较弱的。在 WL 测试失败样例的局限性的指导下，一些著作提供了比 WL 测试具备更强能力的 GNN 变体，因此在一些需要进行结构探测的任务中能够发挥作用。一个主要的例子就是计算化学中，分子的化学性质受到原子环的存在与否的极大影响。

也许寻求更大的表达能力的需求正是 WL 测试自己。实际上，原始的 WL 测试的能力可以通过多层次 WL 测试来进一步提升，例如关联了节点的 k -tuple 的 k -WL 测试[Morris et al., 2017]。 k -WL 测试被等人直接翻译为了高度 k -GNN 架构[Morris et al., 2019]，可以证明这一架构比前述的 GNN 更具表达能力。此外也有一些诸如 δ - k -LGNN 的架构[Corso et al., 2020] 来稀疏化 k -GNN 的计算。然而， k -GNN 架构由于需要维护元组表示，导致其在实际中难以用到 $k > 3$ 的情况中。

与此同时，Maron 等人研究了基于 k -tuple 的节点的图网络的不变性与等变形[Maron et al., 2018]、[Maron et al., 2019]。除了展示出任意的不变与等变图网络均可视为有限个生成器的线性组合这一令人惊讶的结果外（生成器的个数依赖于 k ），作者们也展示了这样

的网络层的表达能力等价于 $k-WL$ 测试，并且提出了可证明的满足 $3-WL$ 表达能力的经验规模可扩展的变体。

除了泛化表示的计算所处的域外，分析 $1-WL$ 测试失败的一些特定例子并且使用 GNN 来增强输入以提升其可区分性能的研究方向也有一些重要的尝试。一个常见的例子是将一样的特征附加到节点上，这样可以帮助结构的探测例如若一个节点在 k 步之外找到与其一样的特征，则意味着其在 k 邻域内。这一方法的架构包括 one-hot 表示 [Murphy et al., 2019] 以及纯粹随机特征 (Purely Random Features) [Sato et al., 2020]。

更广泛地说，在尝试将结构信息与消息传递过程结合方面有不少的研究探索，或是通过模块化消息函数。或是模块化计算所在的图。一些有趣的成果包括采样锚框节点集合 [You et al., 2019]、基于拉普拉斯特征适量的聚合 [Stachenfeld et al., 2020]、[Beaini et al., 2020]、[Dwivedi and Bresson, 2020] 等方案，这些方案或是通过拓扑数据分析，或是通过位置编码 [Bouritsas et al., 2020]，还有消息传递 [Bodnar et al., 2021] 的方式来实现。在计算化学领域，通常人们假定分子的功能是由子结构（即功能性群，Functional Group）来决定的，这使得将分子建模为模体的层次，参见 [Jin et al., 2018]、[Jin et al., 2020] 及 [Fey et al., 2020]。

7.9 信号处理及调和分析 (Signal Processing and Harmonic Analysis)

由于卷积神经网络的成功应用，研究者们重新研究起了调和分析、图像处理以及计算神经学，以尝试为解释卷积网络的效果提供一个理论上的框架。Tomaso Poggio 及其合作者们 [Riesenhuber and Poggio, 1999]、[Serre et al., 2007] 提出的 M-Theory 就是受此启发，该理论基于可以被某些对称群处理的模板的思想。另一个值得注意的模型起源于计算神经学，称为 Steerable Pyramids，这是一种在特定输入变换作用下期望性质的多尺度小波分解的一种形式，该研究由 [Simoncelli and Freeman, 1995] 等人进一步发展。他们也是材质生成式模型的中心组件 [Portilla and Simoncelli, 2000]，并在后来被 Deep CNN 特征的可导向小波特征方法取代 [Gatys et al., 2015]。最后，Stephane Mallat 等人提出的散点变换也提供了一个理解 CNN 网络的框架 [Mallat, 2012]，这是通过把若干可训练的滤波器用多尺度小波分解替换来实现的，并且展示了变形稳定性以及在架构中深度的重要作用，[Bruna and Mallat, 2013] 等人发展了该方案。

7.10 图及 Mesh 上的信号处理 (Signal Processing on Graph and Meshes)

图神经网络的另一类重要内容，也即常被称为谱的内容，正是本书的作者之一的研究方向 [Bruna et al., 2013]，其使用了图傅里叶变换的思想。这样构建的根基在于信号处理以及计算调和分析社区，在这些领域里 2000 年左右以及 2010 年早期的时候非欧信号已经称

为研究的主要对象了。来自 Pierre Vandergheynst [Shuman et al., 2013] 以及 José Moura [Sandryhaila and Moura, 2013] 的研究团队在其极具影响力论文中提出了图信号处理的概念（“Graph Signal Processing”，GSP）并且基于图邻接矩阵以及拉式矩阵的特征向量的傅里叶变换。由于近年来图上机器学习的火热图卷积神经网络依赖的谱滤波算法 [Defferrard et al., 2016] 、[Kipf and Welling, 2016a] 成为了该领域中引用量最大的内容。

值得一提的是，在计算机图形学及几何处理领域，非欧调和分析可以早于 GSP 至少十年。我们可以追溯到 1996 年 Taubin 对于流形以及面片模型上的谱滤波方法 [Taubin et al., 1996]。这些方法在 2000 年左右成为了主流的方法，以 Karni 等人 [Karni and Gotsman, 2000] 对谱域几何压缩的算法以及 Lévy 等人 [Lévy, 2006] 使用拉式特征向量左为非欧空间傅里叶基底的算法为代表。谱域方法已经在众多应用中进行了部署，例如形体描述符的构建 [Sun et al., 2009]（Roe Litman 与 Alex Bronstein 提出了类似谱域图 CNN 的可学习形体描述符 [Litman and Bronstein, 2013]，Alex 也是本书作者之一的孪生兄弟）、泛函映射 [Ovsjanikov et al., 2012] 等，这些模型在本书写作之时仍在计算机图形学中广泛使用。

7.11 图形学及几何处理 (Computer Graphics and Geometry Processing)

基于内参度量不变性的形体分析模型被多位计算机图形学以及几何处理领域的研究人员独立提出了 [Elad and Kimmel, 2003] 、[Mémoli and Sapiro, 2005] 、[Bronstein et al., 2006]，并且由其中一位作者在其著作中深入地进行了分析 [Bronstein et al., 2008]。内参对称性的概念在这些领域中也有一定的探索 [Raviv et al., 2007] 、[Ovsjanikov et al., 2008]。第一个应用于面片模型的深度学习架构，也就是测地线 CNN (Deodesic CNN) 也是本书一位作者的团队提出的 [Masci et al., 2015]。该模型使用共享权重的局部滤波器，并将其应用到测地线片丁上，该设计也是后续本书另一位作者提出的另一个架构的一种特例 [Cohen et al., 2019]。拥有可学习聚合操作的测地线 CNN 架构的扩展架构，也即 MoNet，由来自同一团队的 Federico 提出 [Monti et al., 2017]，该方案在面片模型的局部结构特征上使用了类似于注意力的机制，展现出了可用于一般图上的能力。图注意力网络 (Graph Attention Network, GAT) 在技术视角看来也可以视为 MoNet 的一种特例，该方案是由本书另一位作者提出的 [Veličković et al., 2018]。GAT 通过整合节点特征信息扩展了 MoNet 的注意力机制，与之前相关的以纯粹结构驱动的方案分道扬镳了。该方案也是目前应用中最为流行的选择。

在计算机图形学中，值得一提的是来自斯坦福大学的 Leo Guibas 组发展了集合上学习的想法 [Zaheer et al., 2017]，提出了 PointNet 来分析三维点云 [Qi et al., 2017]。这一架构也有一系列跟随的成果，包括本书作者之一提出的动态图卷积神经网络 (Dynamic Graph CNN, DGCNN) [Wang et al., 2019b]。DGCNN 使用了最近邻图来捕获点云局部的结构以

使得节点之间可以交换信息。该架构的关键特点是神经网络之间的层与下游任务之间的图是在线构建并被更新的。后者使得 DGCNN 是潜图学习的重要化身之一，这也引起了众多跟踪研究。对 DGCNN 的 k 近邻图方法进行扩展的研究包括进行显式的图上边的控制，或是通过双边优化[Franceschi et al., 2019]、强化学习[Kazi et al., 2020]，或是通过直接监督学习[Veličković et al., 2009]。一个变分方向（从计算后验概率中以概率性的方式采样边）独立地以 NRI 模型的形态出现[Kipf et al., 2018]，尽管该方案依然依赖节点数的平方形式的复杂度，其使得显式地对所选边的不确定度进行编码。

另一个在并不给定图的情况下进行图上进行学习的非常流行的方向依赖对于整个完备图的 GNN 风格的计算，使得其自行推断连接性。这一需求来源于自然语言处理，其中一句话中多个单词以一种重要的且非顺序性的方式进行相互作用。对单词形成的完备图进行操作导出了 Transformer 的第一个化身[Vaswani et al., 2017]，该架构将神经及其翻译中居于顶峰王座的循环模型以及卷积模型赶了下来，并引起了相关工作的崩溃，使得 NLP 与其他领域的边界贯通了。全连接 GNN 的计算同时也出现在了仿真[Battaglia et al., 2016]、推理[Santoro et al., 2017]、多主体[Hoshen, 2017]应用中，并且在节点数较小的情况下目前仍是一种流行的选择。

7.12 算法推理 (Algorithmic Reasoning)

对于本节中大多数的讨论，我们均会给出空间几何的例子，这反过来也勾勒了内在的域以及其不变性与对称性。然而，在计算设定中也有大量的不变性与对称性出现。几何深度学习的众多常见设定的关键不同之处在于无需为任何类型的相似性、逼近或者关系类型进行编码以便于关联，他们仅仅指明了连接的数据点之间数据流的方式。

相反，神经网络的计算正是模拟了算法的推理过程[Cormen et al., 2009]，以及算法控制流与中间结果引起的额外的不变性。例如在最短路径的计算算法 Bellman-Ford 算法[Bellman, 1958]中，在 K 步骤后，该算法总是计算到源节点的最短路径，并且使用不超过 K 的边的数量。在算法研究中，假定的输入不变性通常也被称为前置条件(Precondition)，算法本身保持的不变性称为后置条件(Postcondition)。

同名的一个研究方向就是算法推理 (Algorithmic Reasoning) [Cappart et al., 2021]，该方向旨在生成能够合理地保持算法不变性的神经网络。这一研究领域已经研究了通用目的神经计算机的构建，即图灵机[Graves et al., 2014]，以及可微神经计算机[Graves et al., 2016]。尽管这些架构有着通用计算机的所有特征，他们每次仅引入了若干组件，使其难以优化，并且实际上，这些通用计算机通常还不如小的关系型推理单元，例如[Santoro et al., 2017]、[Santoro et al., 2018]等提出的计算方法。

由于对复杂的后置条件进行建模是极具挑战性的，归纳偏置的学习执行中大量的研究集中于基础算法[Zaremba and Sutskever, 2014]，例如简单的算法。这一方向主要的例子包

括神经 GPU (Neural GPU) [Kaiser and Sutskever, 2015]、神经程序员解释器 (Neural Programmer-Interpreters) [Reed and De Freitas et al., 2015]、神经算法逻辑单元 (Neural Arithmetic-Logic Unit) [Trask et al., 2018]、[Madsen and Johansen, 2020] 以及神经执行引擎 (Neural Execution Engine) [Yan et al., 2020]。

随着 GNN 架构的快速发展枚举超线性复杂度的算法组合已经成为了可能。算法校准框架 (Algorithmic Alignment Framework) [Xu et al., 2019] 已经从理论上展现了 GNN 可很好地模仿动态规划[Bellman, 1966]，而动态规划则是众多算法的基本原理。同时根据本书其中一位作者的经验表明，实际上也可以设计并训练 GNN 以使得可以与算法不变性一致 [Veličković et al., 2019]。在这之后，神经网络与算法对其的研究有迭代算法 (Iterative Algorithms) [Tang et al., 2020]、线性化算法 (Linearithmic Algorithms) [Freivalds et al., 2019]、数据结构 (Data Structures) [Veličković et al., 2020]、持久记忆 (Persistent Memory) [Strathmann et al., 2021] 等。这样的一些模型也在隐式规划 (Implicit Planners) [Deac et al., 2020] 中得到了实际的应用，并进入了强化学习的研究领域中。

同时，物理仿真领域也是用 GNN 做出了一些突出的成果 [Sanchez-Gonzalez et al., 2020]、[Pfaff et al., 2020]。这一方向对 GNN 扩展的设计提出了一致的建议。这样的关联是共同预期的：给定一个可以表述为离散时间仿真的算法，以及可以一步一步进行部署的算法，这两个方向均需要保持相似的不变性。

与算法推理紧密相关的是外插的度量。考虑到这些外插的成功应用均是在分布内部才可以进行，这对于神经网络而言是一个臭名昭著的痛点，也即当在训练数据中发现了合适的模式时才可以预见测试数据中也可能有类似的模式。然而，算法不变性必须保持，这与输入分布的数量或者生成分布无关，意味着测试集可能难以覆盖实际中使用的所有场景。[Xu et al., 2020b] 等人通过使用 ReLU 激活函数的外插 GNN 的要求提出了几何增强方法：其组件以及特征化必须按照满足组成模块仅通过线性目标函数进行学习的条件。[Bevilacqua et al., 2021] 等人提出在因果推理中观测外插，从而生成与环境无关的图表示。

7.13 几何深度学习 (Geometric Deep Learning)

最后从历史发展的角度的评论将以本书的名字结束。几何深度学习这一术语由本书在作者之一于 2015 年 ERC 启动资金资助过程中率先提出的，并且在同名的 IEEE 信号处理杂志文章中开始流行 [Bronstein et al., 2017]。这篇文章谨慎地宣布“一个新的研究领域诞生了”。考虑到最近的图神经网络的流行，机器学习应用中越来越多地应用了不变性与等变形，这也是我们写作本书的目的，也许将刚才所述的新的领域已然诞生这一预言视为起码部分的成立的。本书名字中 4 个 G，即 Grids、Graphs、Groups、Gauges 是由 Max Welling 为几何深度学习的 ELLIS 项目敲定的，本书两位作者参与联合指导了该项目。需要承认的是，最后一个 G，也就是 Gauges 有点勉强，这是由于内含的结构是流形与从而非度规。对

展 望

本书而言，我们增加了一个 G，也即 Geodesics，用以指代度量不变性以及流形的内参系统。

致 谢

本书通过从几何视角的不变性与对称性的角度，对于总结与综合深度学习架构数十年的研究展示了一个谦逊的尝试，我们希望这一视角能够使得学习该领域的新手以及在该领域工作的人们更容易地理解该领域，同时为综合新颖的架构的科学家们提供一个我们构造的蓝图范本。这样，我们希望展示深受 Vaswani 启发[Vaswani et al., 2017] 的一句话：“all you need to build the architectures that are all you need.”。

本书写于 2020 年末及 2021 年初，我们对描述的整个蓝图是否是有意义的也有着无数的疑虑，并且在同事的帮助下帮我们克服了推出本书的怯场情绪，并在 Petar 在剑桥的演讲、Michael 在牛津以及帝国理工的讲座中首先介绍了一下我们的这一工作。在 Andreas Maier 的盛情邀请下，Petar 也得以有机会在 Friedrich-Alexander-Universität Erlangen-Nürnberg 介绍了我们的工作，而该地正是 Erlangen Programme 发源地！关于这些讲座我们受到的众多的反馈也使得我们的情绪极度高涨，并促使我们将本书进一步打磨。最后，我们想要感谢 ICLR 2021 组织委员们，由于他们 Micheal 才得以有机会以 Keynote 的形式进行讲座呈现。

我们也应当指出调和如此大量的研究内容仅凭我们作者 4 人肯定不够，因此我们也要对本书进行深入研究并给出了审慎评论以及引用的研究者们表示感谢，他们有：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, Ngan Vu, Peter Wirsberger and David Wong。这些研究者们专业的反馈对于坚定我们的信念以及使得本书对于更多人有用也是无价的。当然本书中出现任何的异常均是我们四人的责任。目前本书也正处于打磨当中的状态，欢迎在任意的阶段给出评论。如果您发现本书有任何错误或者遗漏也可联系我们。

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.
- SI Amari. Feature spaces which admit and detect invariant signal transformations. In *Joint Conference on Pattern Recognition*, 1978.
- Brandon Anderson, Truong-Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. *arXiv:1906.04015*, 2019.
- 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.
- Victor Bapst, Thomas Keck, A Grabska-Barwińska, Craig Donner, Ekin Dogus Cubuk, Samuel S Schoenholz, Annette Obika, Alexander WR Nelson, Trevor Back, Demis Hassabis, et al. Unveiling the predictive power of static structure in glassy systems. *Nature Physics*, 16(4):448–454, 2020.
- 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, 20.
- 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 Bougleux, 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 Gülcühre, 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 Drozdzal, 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. Convolutional 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.
- Karlis Freivalds, Emīls Ozolinš, 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 rotation-equivariant attention networks. arXiv:2006.10503, 2020.

- Kunihiro 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, Frederico 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 Kosciolak, 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 WardeFarley, 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.
- Mikhail Gromov. Structures métriques pour les variétés riemanniennes. Cedic, 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 selfattention 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. Harisiqb88/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.
- Łukasz 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, Roee 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, Elvin 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.
- Roee 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 Sotiroopoulos, 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 Gouskos. 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 rcnn: 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 Žídek, 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.
- Balasubramanian 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 Barekatian, 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 Kälberer, 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 für 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 Hesemi. 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 semisupervised 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.

参考文献

- 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.