机器学习模型的可视分析

（申请清华大学工学博士学位论文）

培 养 单 位：高 等 研 究 院
学　　科：计 算 机 科 学 与 技 术
研　究　生：刘　梦　尘
指 导 教 师：沈　向　洋　教　授
联 合 导 师：张　钹　教　授

二〇一八年六月

Visual Analytics of Machine Learning Models

Dissertation Submitted to

Tsinghua University

in partial fulfillment of the requirement

for the degree of

Doctor of Philosophy

in

Computer Science and Technology

by

Liu Mengchen

Dissertation Supervisor : Professor Shen Xiangyang

Associate Supervisor : Professor Zhang Bo

June, 2018

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：

清华大学拥有在著作权法规定范围内学位论文的使用权，其中包括：（1）已获学位的研究生必须按学校规定提交学位论文，学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；（2）为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆、资料室等场所供校内师生阅读，或在校园网上供校内师生浏览部分内容；（3）根据《中华人民共和国学位条例暂行实施办法》，向国家图书馆报送可以公开的学位论文。

本人保证遵守上述规定。

（保密的论文在解密后应遵守此规定）

作者签名：_____　　导师签名：_____

日　　期：_____　　日　　期：_____

摘 要

机器学习取得的显著成功催生了众多人工智能应用，例如个性化推荐，汽车自动导航等。在这些应用中，机器学习模型常常被当作一个黑盒子。由于不能理解这些模型的工作机理，高效模型的开发常常依赖一个冗长又昂贵的反复实验过程。因此，研究人员和从业人员（专家）迫切需要一个透明和可解释的机制，帮助他们更好地理解和分析学习模型，从而快速设计出符合需求的模型。为此，本论文以提升机器学习模型的可解释性为目标，研究机器学习模型的可视分析技术与方法，从而使人工智能系统能够生成可解释的分析结果。

具体来说，本论文提出了三个可视分析方法，帮助专家更高效地完成模型开发过程中三个主要任务：（1）理解模型的工作机理；（2）诊断模型的训练过程；以及（3）改进模型的预测性能。

在模型理解方面，本论文提出了卷积神经网络工作机理分析与理解的可视分析方法，帮助专家理解训练过程中单个时间片上的训练状态。为了分析大规模网络，本论文将卷积神经网络建模为有向无环图，并进行多层次聚类。为了展现聚类后的网络的多方面信息，本论文提出了一个混合可视化方法，包括：层次化矩形布局、矩阵重排和基于双聚类的边绑定。

在模型诊断方面，本论文提出了深度生成模型训练过程诊断的可视分析方法，帮助专家交互地探索模型性能不佳或训练失败的原因。在时间片层次，本论文结合有向无环图和折线图，有效展现数据在网络中的流动。在网络层次，本论文利用基于蓝噪声的折线采样算法，减少由大量训练动态数据带来的视觉混乱并保留异常值。在神经元层次，本论文提出了责任分配算法，揭示神经元之间的相互影响，帮助专家诊断模型训练失败的根本原因。

在模型改进方面，本论文提出了基于不确定性的模型改进可视分析方法，帮助专家将人的知识集成到检索模型中，提高模型整体性能。本论文以微博数据为例，将其检索问题建模为互增强图模型，并计算检索结果的不确定性，以及不确定性在图上的传播。相应地，本论文紧密结合图可视化、不确定性符号以及流向图等多种可视化技术有效展现这些信息，帮助专家找到最不确定的检索结果，并交互地修改。另外，本论文提出了增量式模型更新算法，根据专家的修改逐步改进模型，形成一个迭代循环的模型改进过程。

关键词： 可视化；交互式模型分析；可解释人工智能

Abstract

The success of machine learning has triggered many AI applications, such as personalized recommendation and automatic navigation. In these applications, machine learning models are often treated as black boxes. Due to the limited understanding of the working mechanism of these models, developing effective models typically relies on a substantial amount of trial-and-error. Consequently, machine learning researchers and practitioners (experts) desire a transparent and explainable mechanism, to help them better analyze machine learning models and accelerate the development of effective models. To this end, this thesis studies visual analytics of machine learning models, which aims at generating more explainable results for AI systems.

In particular, we develop three visual analytics approaches that help experts with the major tasks in model development: (1) understanding the working mechanism of a model; (2) diagnosing a failed training process; (3) refining a model to improve its performance.

For model understanding, we propose a visual analytics approach to better understanding the working mechanism of a convolutional neural network (CNN). The approach helps experts analyze one snapshot in a training process of a CNN. To effectively analyze a large CNN, we formulate a CNN as a directed acyclic graph (DAG), and aggregate the generated DAG by the developed multi-level clustering. To visualize the multiple facets of the aggregated DAG, we design a composite visualization, including hierarchical rectangle packing, matrix reordering, and biclustering-based edge bundling.

For model diagnosis, a multi-level visual analytics approach is proposed to diagnosing the training process of a deep generative model (DGM). The approach helps experts interactively discover the reason of a training failure or an unsatisfactory performance of a DGM. At the snapshot level, we combine DAG visualization and line charts to effectively present the dataflow in the network. At the network level, we employ blue noise polyline sampling to preserve outliers and reduce visual clutter caused by the large amount of training dynamics. At the neuron level, we propose a credit assignment algorithm that indicates how neurons influence each other, to help diagnose the root cause of a training failure.

For model refinement, we develop an uncertainty-aware visual analytics approach to helping users interactively refine microblog retrieval model by incorporating domain

knowledge into the retrieval model. We model the microblog retrieval problem as a mutual reinforcement graph. We calculate the uncertainty of the retrieval results, and the propagation of the uncertainty. To illustrate these facets, we design a composite visualization with three visual components: a graph visualization, an uncertainty glyph, and a flow map. The visualization enables users to effectively find the most uncertain retrieval results and interactively refine them. The feedback is integrated into the model by the developed incremental model update algorithm, which enables model refinement to be an iterative process.

Key Words: visualization; interactive model analysis; explainable AI

目 录

第 1 章　引言

1.1　研究目标

机器学习取得的显著成功催生了众多人工智能应用，如个性化推荐、汽车自动导航、垃圾邮件过滤等等。随着需要处理的数据量的增大，机器学习模型的结构越来越复杂，参数越来越多。例如，在图像识别中表现优异的卷积神经网络[1]，可能含有上百个网络中间层。每个网络中间层中可能含有上百万参数。研究人员和从业人员 (以下简称机器学习专家或专家）很难理解这些复杂模型的内部工作机理。他们在使用这些模型的时候，往往将其当做一个黑盒子。由于缺乏对这些模型工作机理的深刻理解，高效模型的开发常常是一个冗长又昂贵的反复实验过程。例如，为了开发符合需求的深度神经网络，需要选择合适的网络结构。而网络结构的选择目前主要依赖于专家的经验。不仅如此，网络结构对网络性能的影响，目前还缺乏深层次的认识。这些都极大地提高了高效模型开发的难度。

为了帮助机器学习专家快速设计出符合需求的模型，迫切需要一个可解释的机制，帮助他们更好地理解和分析机器学习模型。机器学习的可解释性主要指：机器学习模型在给出预测结果的同时可以提供相应的原因，使用户更好地理解机器的决策过程；同时通过交互式分析方法，利用可视化手段，帮助用户理解模型的工作机理，对模型的训练和决策过程进行诊断，进而实现模型的改进。由于可解释机器学习在未来人工智能领域的重要意义，世界各国都高度重视可解释机器学习模型的研究。2016 年 8 月，美国国防高级研究计划局（DARPA）发布了一份关于“可解释人工智能”（Explainable Artificial Intelligence, XAI）项目的征询建议书，认为可解释人工智能将引领“手工智能”、“统计学习”之后的第三波人工智能浪潮。可解释人工智能旨在寻求建立一套具有可解释模型的机器学习技术，探索新一代人机双向沟通的新技术和新工具，解决人与机器之间相互信任和顺畅交流等问题，便于用户理解和管理日益复杂的人工智能系统。我国在《新一代人工智能发展规划》中也明确将“实现具备高可解释性、强泛化能力的人工智能”作为未来我国人工智能发展的重要突破口。

近年来，人工智能和机器学习领域的顶级国际会议 NIPS、ICML 和 IJCAI 等都纷纷设立了可解释机器学习的专题研讨会，吸引了领域内大量的研究者参与讨论。2017 年，机器学习领域最具影响力的学术会议之一的 ICML 2017 将本次会议的最佳论文授予了“Understanding Black-box Predictions via Influence Functions”[2]。与此同时，可视化领域最具影响力的学术会议之一的 IEEE VIS 2017 将本次会

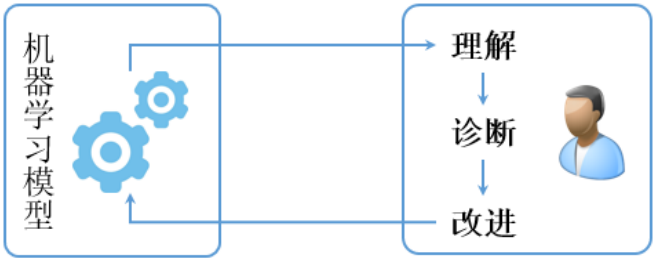


图 1.1　机器学习模型开发过程中的主要任务

议的最佳论文授予了 Google 研究人员基于 TensorFlow 的深度模型可视分析工作“Visualizing Dataflow Graphs of Deep Learning Models in TensorFlow”[3]。这些都有力地说明可解释机器学习极大地吸引了学术界和产业界关注。

可视分析是可解释机器学习研究的重要手段。美国国防高级研究计划局（DARPA）将基于可视分析的可解释机器学习作为“可解释人工智能”计划的三大研究方向之一[4]。基于可视分析的可解释机器学习利用可视化方法，帮助专家更好地理解机器学习模型的工作机理，诊断模型训练过程中可能出现的问题，为进一步改进和完善模型提供必要的信息。**本论文以提升机器学习模型的可解释性为目标，研究机器学习模型的可视分析技术与方法**，帮助专家快速设计出符合需求的模型。

1.2　研究思路与技术挑战

为了帮助专家快速设计出符合需求的模型，本文主要研究如何利用可视分析技术帮助专家完成开发过程中的主要任务（图1.1），即：

- **理解**模型的工作机理；
- **诊断**模型的训练过程；
- **改进**模型的预测性能。

下面，本文针对这三个主要任务中机器学习专家面临的问题，以及利用可视分析技术帮助专家完成相应任务面临的技术挑战进行探讨。

在机器学习模型的开发过程中，专家首先需要**理解**现有的模型的工作机理与运行状态。为此，专家往往会查看一些数值统计信息，例如，模型的准确率、模型的损失函数值等。然而这些数值统计信息不足以帮助专家深入理解模型的工作机理。例如，对于深度神经网络，无法仅通过其模型准确率，理解网络中每个神经元学到的特征，以及这些神经元之间的相互影响。为了深入理解模型的工作机理，专家有时会浏览底层的模型参数。例如，专家有时会浏览一个感兴趣的神经元所有连边的权值，以理解这个神经元在网络中的作用。但是，这种浏览方式很不直

观。另外，由于机器学习模型结构越来越复杂，专家无法浏览所有的模型参数。利用可视分析技术，可以将复杂机器学习模型的工作机理转换成易于用户理解的直观展现形式，帮助专家更好地理解模型的工作机理。为此，我们需要解决两个技术挑战。第一个挑战是有效处理复杂的机器学习模型。随着深度学习的广泛应用，机器学习模型的结构越来越复杂。最先进的深度神经网络可能含有上百个网络中间层。有效地展现如此复杂的机器学习模型的工作机理，是目前仍待解决的问题。第二个挑战是将模型工作机理转换成易于用户理解的直观展现形式。现在，以深度神经网络为代表的深度学习技术广泛应用在计算机视觉，自然语言处理等应用中。但是对这些模型工作机理的研究还处于滞后阶段。因此，将这些模型抽象的工作机理用可视化的语言“翻译”成易于用户理解的视觉语言，是具有挑战性的。

在理解了现有模型的工作机理与工作状态之后，如果模型的准确率无法满足需求或者模型训练失败，专家需要**诊断**模型的训练过程。为此，在现在的工作流程中，专家往往会在模型训练过程中或者训练之后输出训练过程的一些高层统计信息，例如，模型的损失函数值随时间的变化。这能够帮助他们找到感兴趣的时间点。在找到感兴趣的时间点之后，专家会浏览网络中每个中间层的一些统计信息来找到感兴趣的中间层。为了找到出现问题的神经元，专家会浏览感兴趣的网络中间层中的部分训练动态数据，例如该层神经元响应在训练过程中的变化。在发现异常的神经元之后，专家通常会用领域知识分析网络训练失败的根本原因。网络训练失败可能由多种错误引起，包括代码中的错误，数值上的不稳定性，或者是网络结构不合适等等。因此，这个步骤极大地依赖专家的专业知识。利用可视分析技术能够帮助专家交互地探索模型性能不佳或训练失败的原因，减少对专家专业知识的依赖。为此，我们需要解决两个技术挑战。第一个是有效处理大量的训练动态数据。由于一个机器学习模型可能很复杂（例如神经网络中可能含有上百万神经元连边），直接展示所有训练动态数据会导致严重的视觉混乱。第二个是帮助专家交互地找到找到训练失败的根本原因。由于模型中的组件可能相互影响，找到真正导致训练失败的模型组件是不容易的。

在诊断的基础上，专家会进一步提出**改进**模型的方案，使模型预测性能提高。现在，典型的模型改进工作流程是：专家从输出的统计信息中定位到具体出现问题的代码上，修改代码，然后训练改进后的模型。例如，为了改进深度神经网络，专家往往会浏览每一个网络中间层的平均响应、权值平均更新量等统计信息，以定位到性能不满足需求的网络中间层。该浏览过程很耗时。在此之后，专家在文本编辑器中浏览该网络中间层对应的代码，找到可以尝试改进的地方，并进行相应的修改。在修改之后，专家会用训练脚本重新训练修改好的模型。但是由于模

图 1.2　本文主要研究思路：利用可视分析帮助专家完成开发过程中的主要任务

型训练时间较长，每次修改都重新训练模型会极大地减慢开发过程。为了帮助专家更高效地改进模型，本文研究交互式模型改进技术，利用可视分析帮助专家有效提高模型整体性能。与现有模型改进工作流程中遇到的主要问题类似的，我们需要解决两个技术挑战。第一个挑战是帮助专家高效地找到需要修改的模型组件。由于机器学习模型结构越来越复杂，需要查看的模型信息越来越多。如何将众多的模型信息有效地组织起来并展现给专家，帮助专家高效地找到需要修改的模型组件是不容易的。第二个技术挑战是有效地将专家的分析结果集成到模型中。随着大数据时代的到来，模型训练数据集的数据量越来越大。这导致模型的训练时间越来越长。因此，不能在每次修改之后都重新训练模型。而如何有效地利用专家的分析结果还亟待解决。

1.3　论文的主要工作

为了应对上述技术挑战，本论文提出了三个可视分析方法，帮助专家理解、诊断以及改进机器学习模型（图1.2）：(1) 在模型理解方面，本论文提出了卷积神经网络工作机理分析与理解的可视分析方法，帮助专家理解训练过程中单个时间片上的训练状态。(2) 在模型诊断方面，本论文提出了深度生成模型训练过程诊断的可视分析方法，帮助专家交互地探索模型性能不佳或训练失败的原因。(3) 在模型改进方面，本论文提出了基于不确定性的模型改进可视分析方法，帮助专家将人的知识集成到检索模型中，提高模型整体性能。

1.3.1　模型理解：卷积神经网络工作机理分析与理解的可视分析

首先，本论文研究卷积神经网络（convolutional neural network）工作机理分析与理解的可视分析。卷积神经网络在很多模式识别任务上相较于传统方法有了很

大提高。但是在应用中往往被当做一个黑盒子。因此，本文利用可视分析技术帮助专家分析与理解训练过程中单个时间片上的训练状态。为了有效地分析与理解深层卷积神经网络，有两个技术挑战。第一，网络中可能含有数十乃至上百个网络中间层，每个网络中间层中可能含有数千乃至数百万神经元。这些神经元之间可能有数百万连边。有效地展现大量的神经元以及连边是具有挑战性的。第二，卷积神经网络中含有大量作用未知的模型组件。现在专家对这些组件的作用还缺乏深入的理解[5]。这些组件还会相互影响。这导致这些组件整体的作用更加难以理解。

为了应对上述技术挑战，本文提出了基于多层次聚类和有向无环图（directed acyclic graph）可视化的可视分析方法。根据卷积神经网络中神经元的连边不存在回路这一结构特点，本文将网络建模为一个有向无环图。为了有效处理大规模网络，本文提出了网络层次和神经元层次的多层次聚类方法，将该有向无环图聚合为一个更加紧凑的图。聚合后的有向无环图中，每一个节点是一个神经元聚类，而图中的边表示神经元聚类间的连边。为了帮助专家理解网络中各个组件的作用，我们提出了一个基于有向无环图的混合可视化方法，展现神经元聚类的多方面信息（神经元学到的特征、响应和对网络的贡献），以及神经元聚类间的连边。

对于神经元聚类，已有的研究工作主要集中在展示神经元学到的特征上[6]。除了神经元学到的特征，从更多方面分析神经元，能够帮助专家全面地分析神经元在网络中起到的作用，从而更好地理解网络的工作机理。例如，神经元的响应（activation）等其他数值特征，也可以帮助专家更好地理解神经元在网络处理不同类别样本时起到的不同作用。因此，对于神经元聚类，本文展现了其多方面的信息，帮助专家从多个侧面分析神经元聚类。在神经元学到的特征方面，将神经元响应最大的前 k 个图片块作为这个神经元学到的特征。为了强调神经元对预测结果的贡献，我们使用图片块的大小表示神经元的贡献，并用矩形布局方法计算每个神经元的位置。然而，现有的最优矩形布局算法（rectangle packing）只能处理少量的矩形。为了解决这个问题，本文提出了层次化矩形布局算法，近似求解，以加速算法。与最优矩形布局算法相比，该算法能够有效地处理含有大量神经元的神经元聚类。为了比较不同神经元的响应，我们利用矩阵可视化展示神经元聚类中的响应。矩阵中每一个元素的颜色表示一个神经元在一个类别的样本上的平均响应。为了揭示神经元聚类内部更细粒度的响应模式，本文提出了并提出了矩阵重排算法，将聚类中的神经元重新排序，让相似的神经元位置相近。因此，本文将该问题建模为旅行商问题，最大化相邻神经元相似度的和。精确求解该问题很耗时。为了进一步加速，我们采用了分治算法。该分治算法的主要思想是将聚类中的神经元通过其相似度建模为图，并利用图聚类算法将这个图分解为若干容易求解的

子图，从而加快求解速度。

有效展示神经元聚类间的连边，能够展示底层的特征（例如，边角）如何结合在一起组成高层特征（例如，人脸）。然而，一个深度卷积神经网络可能含有上万神经元聚类间的连边。如果简单地展示所有连边，必然导致严重的视觉混乱现象。为了解决这个问题，本文提出了基于双聚类的边绑定（biclustering-based edge bundling）算法，以减少神经元聚类之间海量连边带来的视觉混乱。

1.3.2　模型诊断：深度生成模型训练过程诊断的可视分析

其次，本论文研究深度生成模型（deep generative model）训练过程诊断的可视分析方法。深度生成模型在无监督和半监督学习中有着广泛的应用[7]。但是，其训练过程由于结合了自顶向下的生成过程以及自底向上的贝叶斯推理过程，相对复杂，容易失败。因此，诊断深度生成模型的训练过程具有重要的理论和实践意义。诊断诊断深度生成模型的训练过程有两个主要的技术挑战。第一个挑战是有效处理深度生成模型训练过程中产生的大量训练动态数据。典型的训练动态数据包括：响应/梯度/权值随时间的变化。由于深度生成模型中往往包含上百万响应、梯度和权值，其训练过程中可能包含上百万响应/梯度/权值随时间的变化。直接用机器学习专家熟悉的折线图展示所有的训练动态数据会导致严重的视觉混乱。第二个技术挑战是找到训练失败的根本原因。由于深度生成模型中，神经元相互影响，因此找到真正导致网络训练失败的神经元较为困难。

为了应对上述技术挑战，我们提出了一个多层次可视分析方法，帮助专家交互地探索模型性能不佳或训练失败的原因。该多层次可视分析方法与专家的典型诊断过程是一致的，其包含三个层次：时间片层次，网络层次及神经元层次。作为分析过程的开始，专家可以浏览损失函数随时间的变化，以找到感兴趣的时间片。在时间片层次，我们结合有向无环图和折线图，有效地展现数据在网络中的流动情况，帮助专家找到感兴趣的网络中间层。在网络层次，我们采用基于蓝噪声的折线采样算法[8]，挑选出该网络中间层中具有代表性的训练动态数据，例如该层响应/权值/梯度随时间的变化。该采样算法能够保留异常值，并有效减少大量训练动态数据带来的视觉混乱。浏览采样结果能够帮助专家定位到可能导致训练失败的神经元。在神经元层次，我们提出了责任分配算法（credit assignment），揭示神经元之间的相互影响。我们利用层级相关性传播算法（layerwise relevance propagation）计算神经元的前向影响，并基于后向传播算法（back-propagation）计算神经元的后向影响。通过浏览神经元之间的相互影响，专家能够有效分析网络训练失败的根本原因。

1.3.3　模型改进：基于不确定性的模型改进可视分析方法

最后，本论文研究基于不确定性的模型改进可视分析方法，帮助专家将人的知识集成到检索模型中，提高模型整体性能。本论文以微博检索为例展开研究。交互地改进微博检索模型主要面临两个技术挑战。第一个挑战是高效地找到需要修改的模型组件。现在，为了改进一个模型，专家需要从自己输出的统计信息中，定位到检索结果中不正确的部分。该定位过程很耗时而且极大地依赖于专家的领域知识。专家往往需要查看众多信息才能定位到需要修改的模型组件。第二个挑战是有效地将专家的分析结果集成到模型中。现在，在定位到需要修改的模型组件之后，专家需要修改对应的代码，并用训练脚本重新训练修改好的模型。随着大数据时代的到来，机器学习模型的训练时间越来越长。每次修改后都重新训练模型会极大地减慢开发过程。

为了应对这两个挑战，我们提出了基于不确定性的模型改进可视分析方法。我们将微博检索问题建模为互增强图模型（mutual reinforcement graph）。该建模能够有效考虑微博数据独有的特性，即微博数据不仅仅包含微博文本，还包括微博用户和微博标签，而且该模型还能有效地处理这三个维度的相互影响。由于蒙特卡洛（Monte Carlo）采样方法具有收敛速度快，可局部更新等特点，我们利用该采样方法求解互增强图模型[9]。根据采样结果，我们对检索结果的不确定性进行建模，并将不确定性在该互增强图模型上的传播建模为一个马尔科夫链。为了展现上述多方面的信息，帮助专家分析检索结果的不确定性并定位到需要修改的检索结果，我们设计了一个混合可视化。具体地说，我们将密度图（density map）与节点连接图相结合，以展现微博消息，用户和标签，以及他们之间的关系。我们设计了表示不确定性的符号，来展现不确定性的分布。我们提出了基于力导向的流向图布局算法，展现不确定性在图上的传播情况。上述混合可视化与不确定性分析有机地结合在一起，帮助专家快速找到检索结果中最不确定的部分，并交互地进行修改。为了有效地将专家的修改融入模型，我们提出了增量式模型更新算法。该算法通过改变采样结果的权重，实现对模型的局部更新，从而高效地将专家的修改融入到图模型中，满足实时交互的需求。所提出的可视化方法会自动地根据修改后的模型更新可视化结果，从而形成一个迭代循环的模型改进过程。

1.4　论文概览

本论文后续章节组织总结如下。第 2 章介绍了机器学习模型可视分析的的研究现状，对现有工作的优缺点进行了分析探讨。第 3 章、第 4 章、第 5 章分别介绍：卷积神经网络工作机理分析与理解的可视分析，深度生成模型训练过程诊断

的可视分析，以及基于不确定性的模型改进可视分析这三份工作。第 6 章进行总结，并对未来研究方向进行讨论。

第 2 章　相关工作

基于可视分析的可解释机器学习是可解释人工智能的三大研究方向之一[4]。基于可视分析的可解释机器学习利用可视化方法，帮助专家更好地理解机器学习模型的工作机理，诊断模型训练过程中出现的问题，为进一步改进和完善模型提供必要的信息。现有相关研究按照研究目的可以分为三类：模型理解的可视分析、模型诊断的可视分析和模型改进的可视分析。

2.1　模型理解的可视分析

机器学习模型的理解是诊断和改进的基础。研究者们提出了一系列可视分析方法帮助专家更好地理解不同机器学习模型，例如分类模型[10-11] 以及回归模型[12]。在所有模型之中，神经网络由于其优异的性能和难以理解的工作机理获得了最为广泛的关注。相关研究工作可以分为两类：基于散点图的模型理解方法[12-13] 和基于图可视化的模型理解方法[11,14]。

2.1.1　基于散点图的模型理解方法

基于散点图的模型理解方法[12-13] 利用散点图（scatterplot）展示样本间的关系。该方法将每个样本表示为一个高维数据点，例如可以将样本的特征表示作为高维数据点。在此基础上，该方法利用降维技术将这个高维数据点降维至 2 维平面，然后利用散点图进行展示。典型的降维技术包括主成分分析 (principal component analysis, PCA)[15] 和 t-SNE (t-distributed stochastic neighbor embedding)[16] 等。基于散点图的模型理解方法可以帮助专家验证关于模型的假设[12]，找到异常样本等[13]。

一个典型的例子是 Rauber 等[13] 提出的，基于 t-SNE 降维技术的模型理解方法。如图2.1所示，每个点代表一个测试样本的特征表示。每个点的颜色表示这个测试样本的类别标签。从图中可以看出，在模型训练之后，不同类别样本的特征表示能够区分地更好。该方法还可以帮助专家发现分类错误的样本。很多分类错误的样本是视觉上的离群点，即周围有很多其他类别的样本。如图2.1(b) 所示，这些分类错误的样本用三角形表示。通过该方法的可视化结果可以发现，很多离群点对应的样本类别判断比较困难，连人都难以分辨它们的类别。例如，如图2.1(b) 所示，一个数字 3 的图片之所以被分错，是因为它跟数字 5 特别相似。

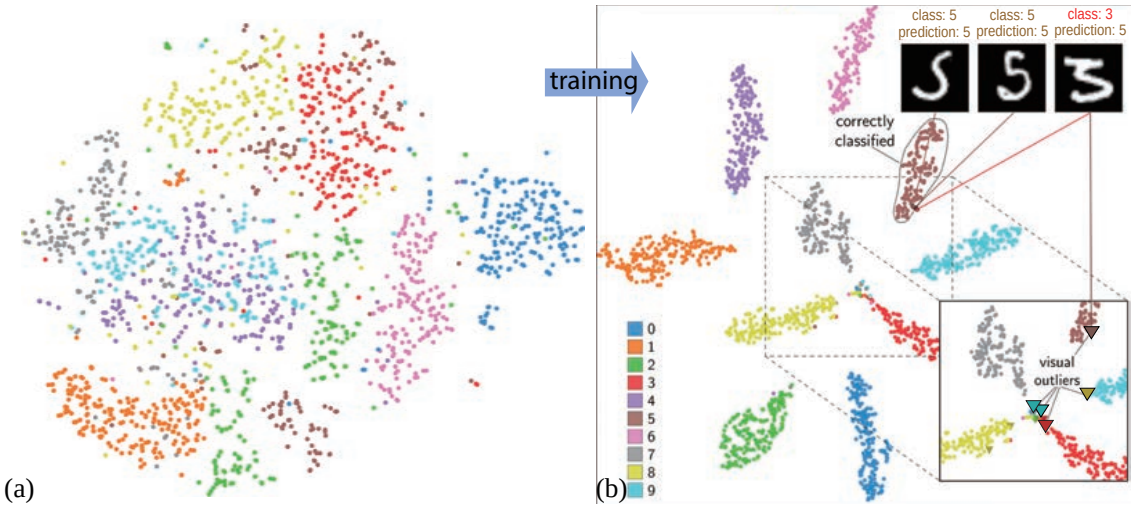


图 2.1　基于散点图的模型理解方法样例[13]：(a) 训练前样本的特征表示；(b) 训练后样本的特征表示（图片引自[13]）

2.1.2　基于图可视化的模型理解方法

基于散点图的模型理解方法能够展示数据集中样本间的关系，但是这些方法无法展示出神经网络的拓扑结构。而神经网络与传统的机器学习模型的重要不同之处就在于其拓扑结构不同。在神经网络中，很多网络中间层相互连接，相邻网络中间层中的神经元也有大量的连接关系[17]，这些中间层和神经元共同完成一定的机器学习任务。而基于散点图的模型理解方法无法展现出这些中间层以及中间层中神经元的相互连接。为了解决这个问题，研究者们提出了一系列基于图可视化的模型理解方法，将神经网络建模为一个图，利用图可视化方法展现网络的拓扑结构[14,18-19]。在这些方法中，常用图中节点和边的颜色以及大小等属性，表示神经网络除拓扑信息之外的其他信息，例如神经元的响应以及神经元之间的相互影响等。这些信息与神经网络的拓扑信息帮助专家从多个角度分析神经网络的工作机理。

一个典型的例子是 Tzeng 等[11] 提出的浅层神经网络可视分析方法。Tzeng 等[11] 将神经网络表示成一个有向无环图，并用有向无环图可视化展现网络的拓扑结构。网络中的其他重要信息通过有向无环图中点（边）的大小（粗细）、颜色以及可视化图标来表示。一个样例可视化结果如图2.2所示。这个图展现的神经网络有一个隐藏层，其目的是判断一个头部体素 (voxel) 是否是脑内物质。这里，每个体素用它的标量值 s、梯度大小 g、它邻居的标量值 n 以及它的位置 p 来表示。边的宽度表示每条边在给定输入体素的情况下的重要性，输入和输出层点的颜色代表对应层神经元的响应。如图2.2(a) 和 (c) 所示，从输出层点的颜色可以看出，该神经网络可以正确地将左边的体素分类成非脑物质（低输出值），将右边的体素分

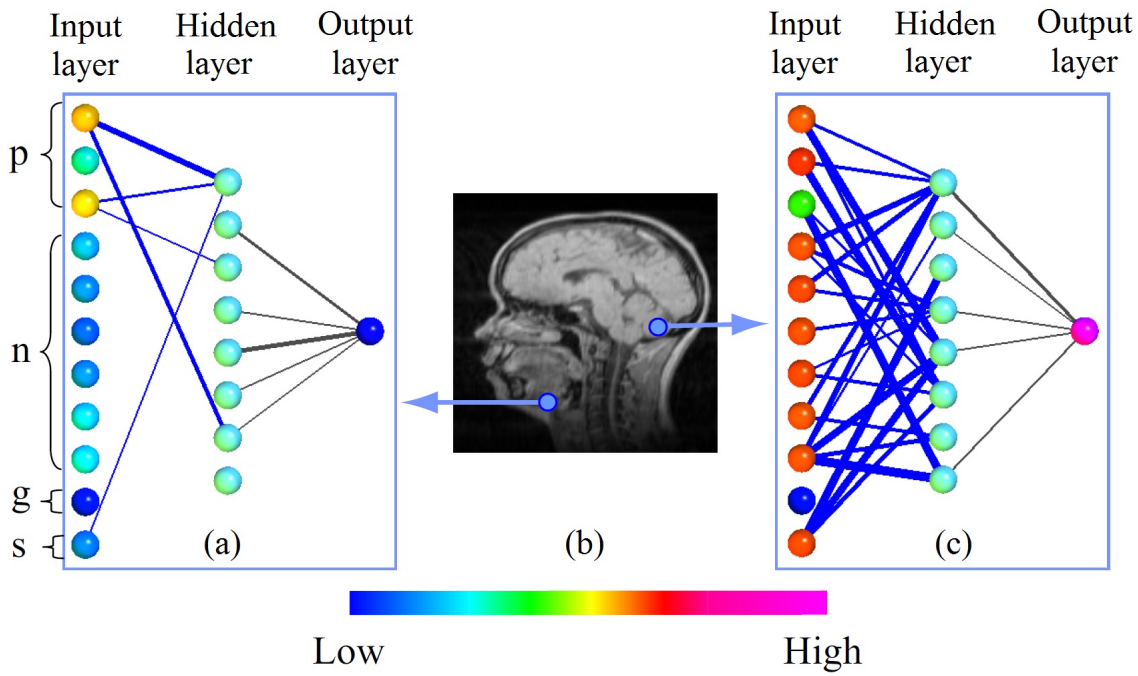


图 2.2　基于图可视化的神经网络理解方法样例[11]：(a) 非大脑成分在网络中的流动；(b) 物质来源实例；(c) 大脑成分在网络中的流动 (图片引自[11])

类成脑物质（高输出值）。另外，从图2.2(a) 和 (c) 中可以看出，要将左边的体素分类成非脑物质，只需要考虑它的位置就可以了；而要将右边的体素识别为脑内物质，则需要考虑除了梯度大小 g 以外的所有输入信息。

Tzeng 等[11] 所提出的方法能够处理含有几个网络中间层，几十个神经元的浅层神经网络。但是，随着深度学习的发展，以深度卷积神经网络为代表的神经网络的深度和宽度都不断增加，给上述方法带来了可扩展性上的挑战。例如，在图像识别领域表现最好的深度卷积神经网络 ResNet[1] 具有上百个网络中间层，数百万神经元。该方法在处理如此大规模的深度卷积神经网络时，会产生严重的视觉混乱。为了解决这个问题，本文提出了基于多层次聚类和大规模有向无环图可视化的可视分析方法，帮助专家理解深度卷积神经网络训练过程中单个时间片上的训练状态（第 3 章）。在将深度卷积神经网络建模为有向无环图的基础上，为了有效处理大规模网络，本文提出了网络层次和神经元层次的多层次聚类方法，将该有向无环图聚合为一个更加紧凑的图。聚合后的有向无环图中，每一个节点是一个神经元聚类，而图中的边表示神经元聚类间的连边。为了帮助专家理解网络中各个组件的作用，我们提出了一个大规模有向无环图可视化方法，展现神经元聚类的多方面信息（神经元学到的特征、响应和对网络的贡献），以及神经元聚类间的连边。

2.2　模型诊断的可视分析

模型诊断的可视分析技术能够帮助专家交互地探索模型性能不佳或训练失败的原因。机器学习模型的训练过程往往是一个迭代过程，由多个时间片组成。按照分析对象为训练过程中单个时间片还是多个时间片，可以将现有研究分为两个部分：训练过程中单个时间片的诊断方法，以及模型整个训练过程的诊断方法。需要注意的是，对模型训练结果的诊断也可以看做是对单个时间片（即训练过程最后一个时间片）的诊断。

2.2.1　训练过程中单个时间片的诊断方法

训练过程单个时间片的诊断方法旨在展示机器学习模型在训练过程中单个时间片上的情况，例如训练结果对应的时间片，帮助专家探索模型训练失败的原因。混淆矩阵（confusion matrix）是机器学习领域常用的诊断训练结果的手段之一。在一个混淆矩阵中，每一个元素 $c[i,j]$ 表示将第 i 类样本分为第 j 类的个数或者概率。混淆矩阵可以提供数据集中所有样本预测结果的一个概览，帮助专家定位到需要进一步浏览的类别上。研究者们提出了一系列可视分析方法帮助专家有效地浏览混淆矩阵。

一个典型的例子是 Alsallakh 等[20] 开发的分类模型诊断工具。该工具包括一个混淆轮 (Confusion wheel) 视图（图2.3(a)）以及一个特征分析视图（图2.3(b)）。混淆轮视图通过直方图来展示预测得分分布，对于每一个类 c_i，装有预测得分值低（高）的样本的区间（bin）被放在内（外）环。混淆轮内部连边的粗细代表了类别标签为 c_i 的样本被分到了 c_j 类的个数，体现出了类与类的混淆程度。这个视图可以帮助专家快速发现那些以较低概率被分错的样本，例如，c_7 中的假阴性样本（false negative，FN）。特征分析视图帮助专家分析和比较两组样本，例如真阳性 (true-positive，TP) 样本和假阳性 (false-positive，FP) 样本。通过特征分析视图，专家能够发现这两组样本可以通过哪些特征进行更好地分离，从而帮助专家在进行特征选择的时候做出更好的决定。

尽管上述工具能够提供有益的诊断信息，但是混淆轮中径向布局的直方图可能会造成一定失真。另外，也有研究人员表示，多个视图可能造成诊断过程较为复杂，专家的诊断负担过重[21]。为了解决这个问题，Ren 等[21] 开发了利用单个视图帮助专家诊断模型的可视分析工具 Squares。如图2.4所示, Squares 从多个粒度上展现混淆矩阵。在最细粒度上，每个样本用一个方块表示（c_3 和 c_5），每个方块的颜色表示对应样本的类别。方块的纹理表示这个样本的预测结果是否正确（实心表示正确，条纹表示错误）。在最粗粒度上，每个类别中的样本用堆叠起来的方块

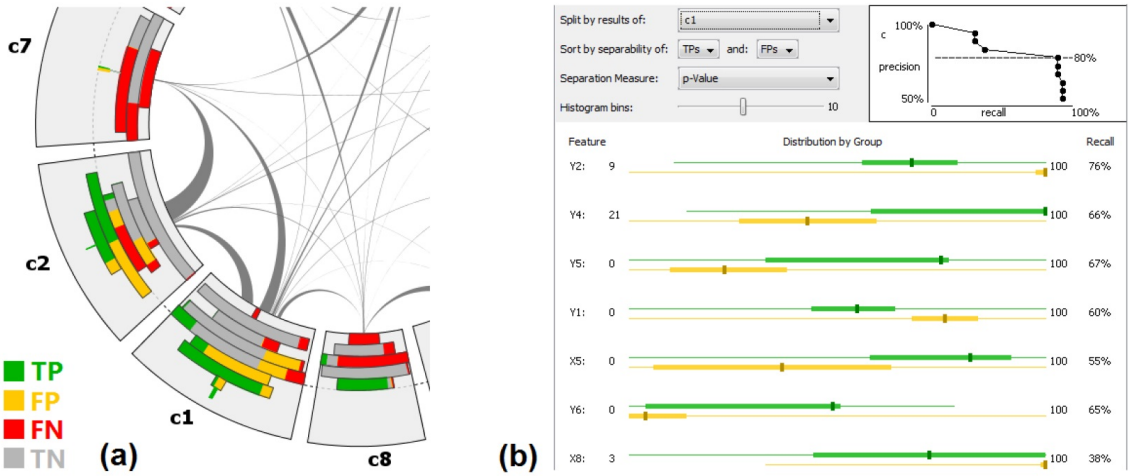


图 2.3　Alsallakh 等[20] 开发的分类模型诊断工具：(a) 提供混淆矩阵概览的混淆轮可视化；(b) 特征可视化 (图片引自[20])

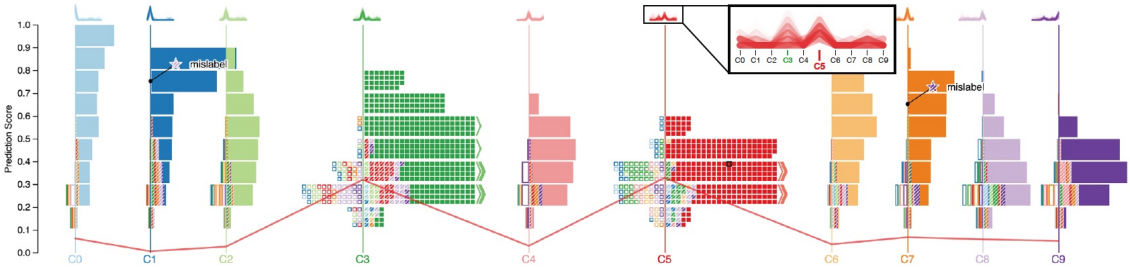


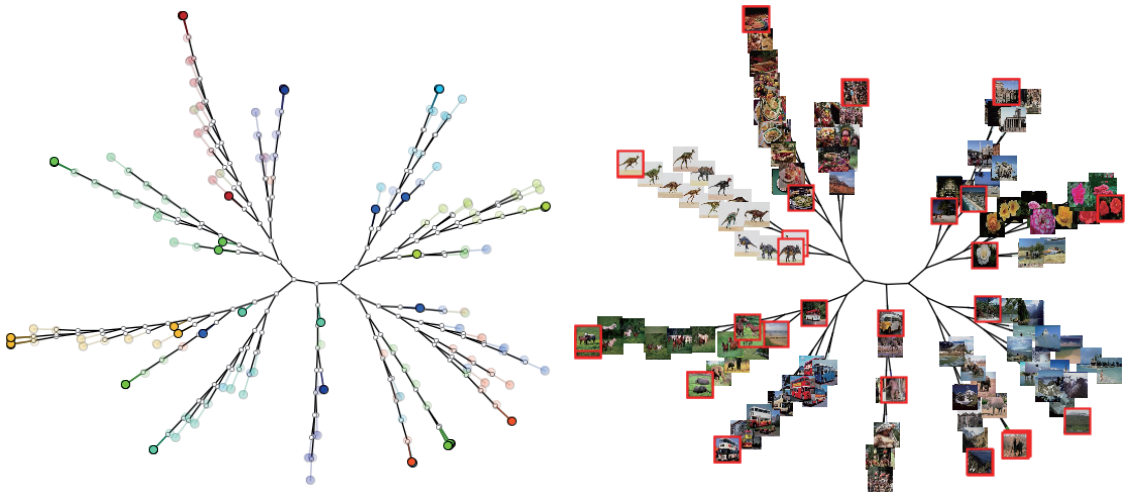
图 2.4　用单视图可视化展现机器学习模型分类结果的可视分析工具 Squares[21](图片引自[21])

表示（c_0 和 c_1）。除了单个类别上的信息，Squares 还能够帮助专家浏览多个类别之间的信息。这部分信息用贯穿所有类别的折线表示（图2.4）。

2.2.2　模型整个训练过程的诊断方法

上述基于单个时间片的模型诊断方法，能够在一定程度上帮助专家诊断模型训练失败的原因。但是，当训练过程过长时，专家无法预知要浏览的时间片。另外，这些方法也不能有效地刻画模型在训练过程中的演化过程。这导致专家无法将训练结果与训练各个阶段的中间结果相比较。为了解决上述问题，研究者们提出了展示模型整个训练过程的可视分析方法。这些方法可以分为两大类：基于投影的方法和非投影方法。

受训练过程中单个时间片诊断方法的启发，基于投影的方法将多个时间片上的高维信息用类似的投影方法投影到低维（2D）空间中。例如，Rauber 等[13] 提出了一个基于 t-SNE 的可视化方法，展现神经网络中每个样本在训练过程中特征表



(a)　　(b)

图 2.5　支持交互式选取需要修改样本的可视分析工具[10]：(a) 圆圈表示待选取样本；(b) 缩略图表示待选取样本 (图片引自[10])

示的变化。具体地说，每个时间片上的信息用 t-SNE 降维技术[16] 投影至二维平面上。为了减少多个投影重叠带来的视觉混乱，将多个投影结果对齐，并用二维的流线展现样本特征表示的变化。上述基于 t-SNE 的可视化能够有效展示出，神经网络经过训练能够更好地区分不同类别的图片这一事实。虽然基于投影的方法能够有效地展现在训练过程中样本特征表示的演化情况，但是，这些方法无法提供网络训练的概览，也无法展现在网络训练过程中响应，梯度以及网络权重的变化情况。浏览上述训练动态数据对于找到真正导致训练失败的神经元是很重要的[22]。

展现整个训练过程中训练动态数据更有效的方法是非投影的方法，例如折线图等。现在已经有几种诊断工具，采用基于非投影方法来展现整个训练过程中的训练动态数据[23-24]。例如，TensorFlow 中提供的诊断工具[23] 支持专家利用折线图浏览整体的训练情况，诸如网络的损失函数变化，每个中间层响应的平均值变化。该诊断工具能够给专家提供一个训练过程的概览。但是不足以帮助专家定位真正导致训练失败的一个或几个神经元。

与上述工具相比，本文提出的多层次可视分析算法（第 4 章）不仅提供了整体的训练情况，还建立了沟通整体训练情况与具体训练动态数据的桥梁。该方法支持专家浏览损失函数的变化作为分析的入口，帮助专家找到感兴趣的时间片。在时间片层次，本文结合有向无环图和折线图，有效展现数据在网络中的流动。在网络层次，本论文利用基于蓝噪声的折线采样算法，减少由大量训练动态数据带来的视觉混乱并保留异常值。在神经元层次，本论文提出了责任分配算法，揭示神经元之间的相互影响，帮助专家诊断模型训练失败的根本原因。

2.3　模型改进的可视分析

在理解了模型的工作机理并诊断出模型训练失败的原因之后，专家往往需要根据自己分析的结果改进这个模型。为了方便地将专家的知识融入到模型之中，研究者提出了一系列交互式模型改进方法。这些方法的优点在于能够有机地结合人的推理能力和机器的运算能力。早期的交互式模型改进方法通常提供简单的用户界面（一个需要标注的文档，一张需要做分割的图片等）让专家进行标注。这一时期的典型工作包括交互式图像分割[25-26] 和交互式图像检索[27] 等。

随着机器学习的发展，机器学习模型越来越复杂，简单的交互界面已不能满足用户深入分析算法和改进模型的需求。为了应对越来越复杂的机器学习模型，研究者提出了一系列模型改进可视分析方法[10,28-29]。模型改进可视分析方法与之前的技术相比，其优点在于允许专家主动进行分析和探索。随着探索不断进行，专家能够不断加深对模型的理解，从而更好地分析出改进的方向。在修改模型之后，系统会根据用户的反馈，得到性能更好的模型，从而形成一个迭代渐进的模型改进过程。针对不同的机器学习模型，研究者构建了不同的模型改进可视分析方法。相关研究工作按照处理的模型可以大致分为两个部分：针对有监督学习的模型改进可视分析方法[10,20] 和针对无监督学习的模型改进可视分析方法[28-29]。

在有监督学习方面，研究者们主要研究如何帮助专家找到对有监督分类器性能影响很大的因素[10,20]，并有针对性的修改。这些因素包括训练样本、特征、分类器种类、训练参数等。例如，Paiva 等[10] 针对图片分类模型开发了一个可视分析工具，支持专家交互地选择需要修改的训练样本，并调整它们的类标。在专家反馈的基础上，该工具能够增量式地更新模型。图2.5展示了该工具如何帮助用户进行有指导的训练样本选择/标注。图2.5(a) 中，每个样本用一个点表示。在图2.5(b) 中，每个样本用更具体的缩略图表示。采用邻居接入树（neighbor joining trees）[30] 对样本进行布局。

在无监督学习方面，为了将专家的反馈融入模型，研究者们往往将问题建模为一个半监督学习问题[28-29,31]。在这些方法中，专家的反馈往往被用作少部分的有监督信息，与原始的无监督数据综合在一起，以提高无监督学习模型的性能。一个典型的例子是 Choo 等[29] 开发的用于改进主题模型的可视分析工具 UTOPIAN。在 UTOPIAN 工具中，主题是用非负矩阵分解（Nonnegative Matrix Factorization，NMF）[32] 生成的。生成的主题采用散点图进行展示。如图2.6所示，UTOPIAN 工具支持专家交互地合并，分解主题，以及基于样例文档或者用户给定的关键词生成新的主题。另外，UTOPIAN 工具还支持主题中关键词的改进。这些交互通过半监督 NMF 算法，与原始的主题模型相融合，增量式地改进原始的主题模型。

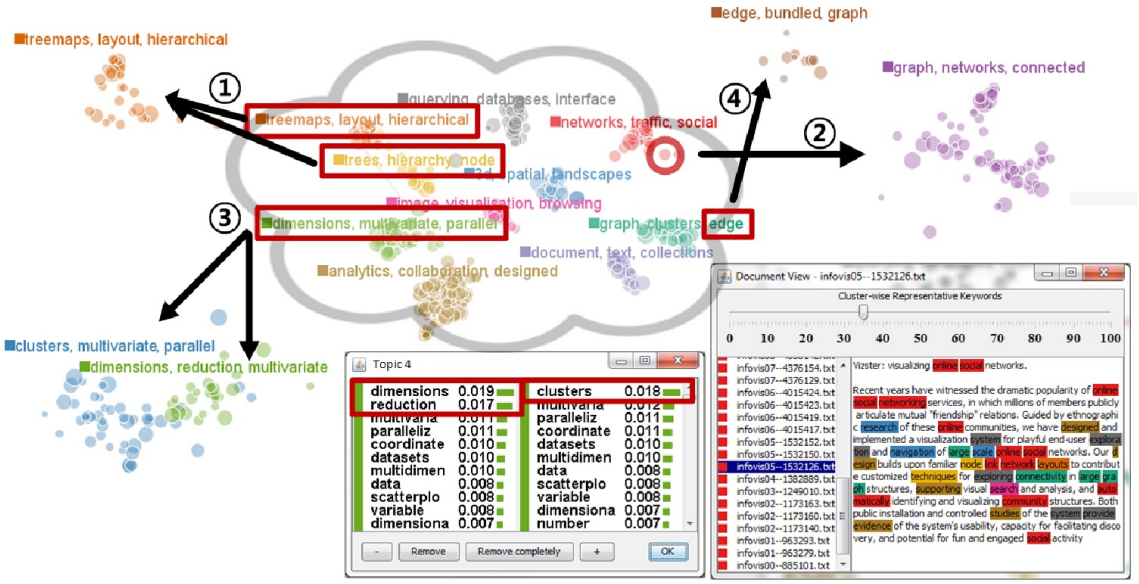


图 2.6　交互修改主题模型的可视分析工具 UTOPIAN[29](图片引自[29])

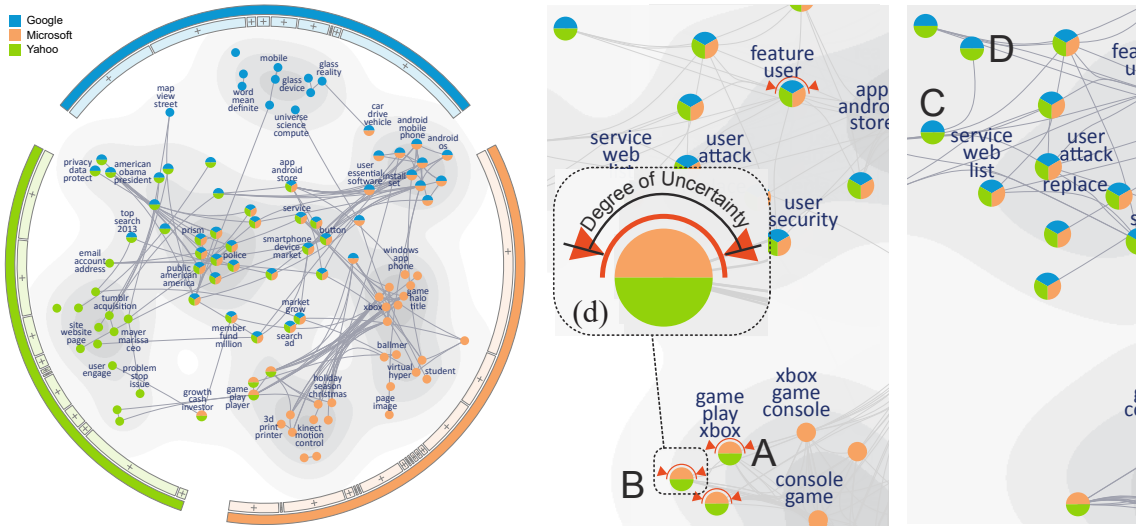


图 2.7　交互修改多源文本数据主题全景图的可视分析工具 TopicPanorama[28]：(a) 全景图可视化；(b) 两个错误的匹配 A 和 B；(c) 更新后改正的错误 C 和 D；(d) 不确定性符号 (图片引自[28])

虽然上述方法能够帮助专家交互地改进机器学习模型，但是从可视化结果中找到待修改的地方需要专家大量的浏览和探索。为了尽量减少专家的工作时间，后续的研究者们，利用机器学习领域主动学习（active learning）[33] 的思想，计算并展现机器学习模型结果的不确定性。这样，专家可以更有针对性地关注不确定性较大的部分，方便地找到需要修改的地方，并进行修改。

例如，Wang 等[28] 开发了 TopicPanorama 工具，帮助商业领域专家改进主题模

型。TopicPanorama 工具的核心思想是展现多源文本中主题全景图，并支持专家交互地进行修改。多源文本中的主题图采用相关主题模型（correlated topic models[34]）抽取。多源文本中抽取出的多个主题图经过图匹配算法，计算出相同以及不同的主题，并拼接成全景图。图匹配的结果采用散点图的可视化形式进行展现。为了帮助专家更容易地找到需要修改的主题，TopicPanorama 工具计算了每一个匹配结果的不确定性。专家可以根据展现出的不确定性方便地找到需要修改的地方并进行修改。专家的修改经过度量学习（metric learning），增量式地改进原有的匹配模型。图2.7(a) 展示了关于谷歌，微软和雅虎的主题全景图。图中，关于不同科技公司的主题用不同颜色的节点表示。多个科技公司共同相关的主题用饼图表示。一个公共关系经理尤其关心游戏方面的主题。因此，她从不确定性最高的游戏相关的匹配入手，以期找到可能不正确的匹配。经过浏览，她发现两个不正确的匹配：A 和 B。A 和 B 把微软 XBOX 游戏相关的主题匹配到了雅虎游戏相关的主题上 (图2.7(b))。经过解除相应的匹配 B，她发现匹配 A 改成了 C，而 B 改成了 D。这两个修改都正确的将谷歌关于运动游戏的主题匹配到了雅虎关于运动的主题上 (图2.7(c)).

与 TopicPanorama 工具相比，本论文提出的基于不确定性的交互式模型改进方法（第 5 章），不仅计算并展示了模型输出结果的不确定性，还计算并展示了不确定性的传播。利用不确定性的传播，专家能从当前的修改出发，方便地找到受该修改影响的其他需要修改的部分。因此，该方法能够更好地节省专家的工作时间。

第 3 章　模型理解：卷积神经网络工作机理分析与理解的可视分析

本章研究卷积神经网络（convolutional neural network）工作机理分析与理解的可视分析方法，帮助专家研究模型训练过程中单个时间片的理解与诊断。卷积神经网络是当前深度学习研究的主流框架之一。其在很多模式识别任务上相较于传统方法有了很大提高[17]，例如语音识别[35-36]、图像分类[1,37-38]、视频分类[6,39] 等。最近，卷积神经网络作为通用的函数近似方法，被应用于深度强化学习中（deep reinforcement learning）。基于卷积神经网络的深度强化学习方法在一系列人工智能任务上，达到甚至超过了人类水平，例如 Atari 游戏[40] 和围棋[41]。然而，在这些应用中，由于卷积神经网络本身复杂的结构和难以理解的工作机理，经常被当做一个“黑盒子”[5]。对于机器学习专家来说，网络中含有大量非线性而且相互作用的组件（神经元以及神经元之间的连接），理解每个组件的作用是比较困难的。由于缺乏对卷积神经网络工作机理的有效理解，开发一个高性能的网络常常是一个冗长又昂贵的反复实验过程[5,42-43]。为了解决这个问题，本章研究如何利用可视分析技术分析与理解卷积神经网络训练过程中一个时间片上的工作机理。

3.1　背景介绍：卷积神经网络

本节简要介绍卷积神经网络的基本结构和相关概念。一个典型的卷积神经网络由几个连续的模块组成（图3.1）。前几个模块由两种网络中间层组成：卷积层（convolutional layer）和池化层（pooling layer）。在卷积层中，每个神经元与上一层的神经元通过一系列权值相连。经过线性加权后的结果作为这个神经元响应函数（activation function）的输入，最终生成神经元的响应，即其输出。响应函数是一个非线性函数，用于避免卷积神经网络仅是输入的线性变换而无法抽取出非线性的特征。通过线性加权与非线性的响应函数，卷积层能够将上一个网络中间层中的神经元学到的不同特征结合得到新的特征。池化层通过计算输入样本中一个较小区域内的局部统计信息（例如，最大值），将相似的特征合并为一个特征。卷积神经网络中使用池化层的优势在于：其一，池化层能够保证网络的输出，在输入平移和形变之后依然保持稳定；其二，池化层能在保留主要特征的同时减少网络参数从而减少计算量。

若干卷积层和池化层之后，卷积神经网络通常中通常包含一个或几个全连接层（fully connected layer）。作为网络的最后组成部分，损失函数（loss function）用

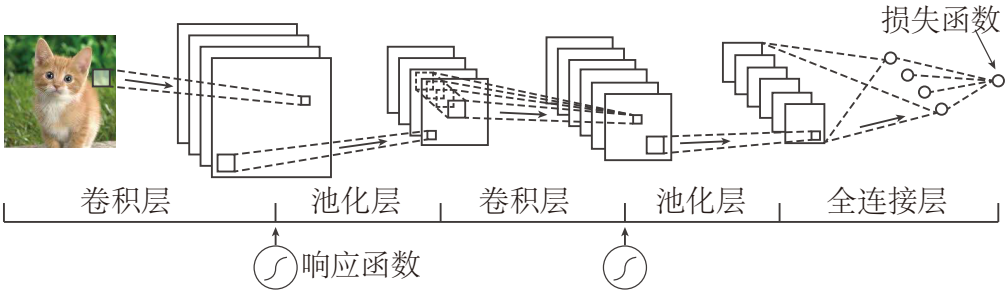


图 3.1　卷积神经网络的典型结构示意图

来衡量网络的性能，即网络的输出和真实类标的差异。

卷积神经网络的训练目标是利用训练数据，最小化损失函数。通常采用基于随机梯度下降（stochastic gradient decent）[44] 的优化方法训练一个网络。具体地说，随机梯度下降是一个迭代的过程。在迭代的每一步，计算神经元连边权值的梯度（gradient），并利用梯度更新神经元连边的权值。

3.2　问题分析与建模

为了有效地分析与理解卷积神经网络训练过程中一个时间片上的运行情况，有两个技术挑战。第一个技术挑战是，最先进的卷积神经网络[1] 中可能含有数十乃至上百个网络中间层（深度），每层中可能含有数百万神经元（宽度），这些神经元之间可能有上百万的连边。如此大量的神经元以及连边，专家无法逐个查看，以找到出现问题的网络组件，例如一组神经元。第二个挑战是，卷积神经网络中含有大量作用未知的组件。现在专家对于这些组件的作用没有一个清晰的认识[5]。例如，具有旁路的 CNN[1] 中旁路的工作机理还存在争议。更何况这些组件还会相互影响，导致这些组件的作用更难以理解。

针对上述挑战，本文提出了基于多层次聚类和有向无环图（directed acyclic graph）的可视分析方法，帮助专家分析与理解卷积神经网络训练过程中一个时间片上的工作机理。典型的时间片包括训练结束对应的时间片，或者是训练异常终止的时间片。图3.2展示了该算法的主要流程。为了应对第一个挑战，将卷积神经网络建模一个有向无环图。为了有效处理大规模网络，本文提出了网络层次和神经元层次的多层次聚类方法，将该有向无环图聚合为一个更加紧凑的图。聚合后的有向无环图中，每一个节点是一个神经元聚类，而边表示神经元聚类间的连边。为了应对第二个挑战，本文提出了一个有向无环图可视化方法，帮助专家浏览神经元聚类的不同方面的信息（学到的特征、响应和对网络的贡献），以及神经元聚类的连边，从而让专家更好地理解这些网络组件的作用，进而揭示卷积神经网络

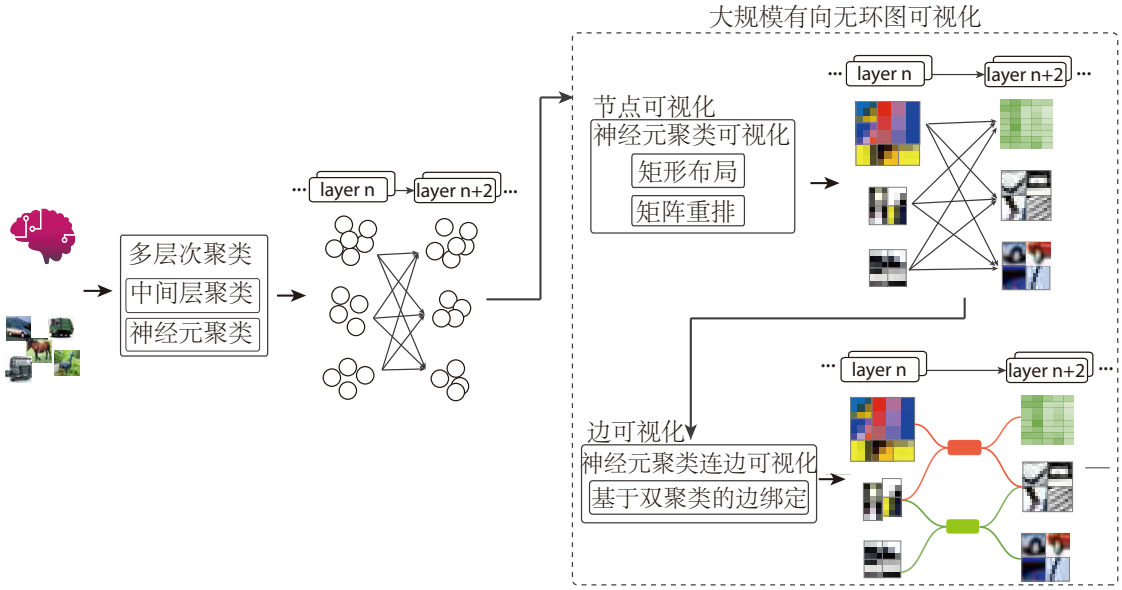


图 3.2　卷积神经网络工作机理分析与理解的可视分析方法概览

的工作机理。

多层次聚类。为了展示卷积神经网络的结构，根据卷积神经网络中神经元连边无回路的结构特点，将卷积神经网络建模为一个有向无环图。该有向无环图中每一个节点代表一个神经元，每一个边代表神经元之间的连边。一个大规模卷积神经网络可能含有数十乃至数百个网络中间层，每一个中间层可能含有数千乃至数百万神经元。相应的，建模得到的有向无环图也可能含有上百层，上百万个节点和边。这就导致，直接展示该有向无环图会导致严重的视觉混乱。

因此，本文提出多层次聚类方法，将该大规模有向无环图聚合成一个更加紧凑的图。具体地说，首先在中间层层次利用网络拓扑结构进行聚类，其次在神经元层次按照神经元起的作用进行聚类。聚合后的有向无环图中，每一个节点是一个神经元聚类，而图中的边表示神经元聚类间的连边，为有向无环图可视化建立了基础。

有向无环图可视化。针对聚合后的有向无环图，本文提出了一个混合可视化方法，展现图中节点（神经元聚类）和图中边（神经元聚类间连边）的多方面信息，帮助专家理解卷积神经网络中神经元与其连边在网络中起到的作用。

对于神经元聚类，已有的研究工作主要集中在展示神经元学到的特征上[6]。展示网络中神经元学到的特征对于专家开始分析是必不可少的。检查神经元的特征能够帮助专家确定卷积神经网络是否学到了具有明确含义的特征，例如汽车轮子等。除了神经元学到的特征，从更多的侧面研究神经元能够更好地帮助专家理解网络的工作机理。例如，神经元的响应（即神经元的输出）等其他数值特征，也

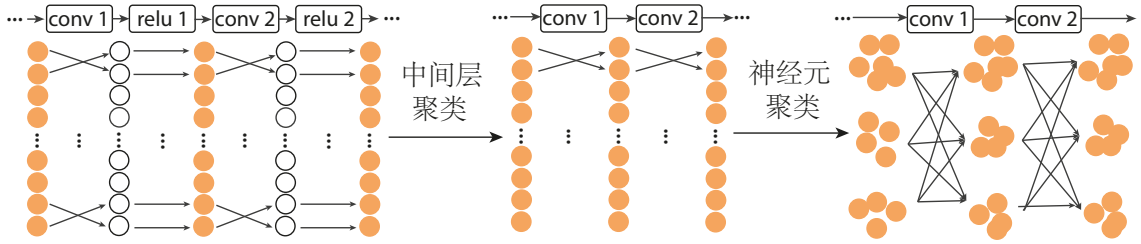


图 3.3　多层次聚类算法概览

可以帮助专家更好地理解神经元针对不同类别样本起到的不同作用。经过与机器学习专家的讨论，我们逐渐总结出专家希望检查的神经元不同侧面：学到的特征，响应，以及对网络性能的贡献。因此，对于神经元聚类，我们展现了聚类这些方面的信息，帮助专家从多个侧面分析神经元。具体地说，提出了一个层次化矩形布局算法（hierarchical rectangle packing）展示神经元聚类中神经元学到的特征以及神经元对网络性能的贡献。另外，提出了基于矩阵重排算法（matrix reordering）的矩阵可视化以展现神经元聚类中神经元在不同类别样本上的响应。专家可以在这些侧面之间切换，深入地分析比较神经元聚类。

神经元间的连边，能够展示底层的特征如何结合在一起组成高层特征。在一个卷积神经网络中，底层的神经元往往学到的是简单的特征，例如条纹，边，角等。中层的神经元往往学到的是物体的一部分，例如车的轮子等。而高层的神经元往往学到的是一个概念，例如猫。这个特性是卷积神经网络能够成功的重要保证之一。这个特性来自于卷积神经网络中神经元的局部连接关系，即，*m* 层的神经元只和 *m-1* 层的神经元中的一小部分相连接。因此，展示相邻层中的神经元的连接关系，有助于分析底层的特征如何结合在一起组成高层特征。然而，两个中间层之间可能含有数以百万计的连接关系，简单地展示所有的连接关系必然导致严重的视觉混乱现象。因此，需要展示这些连接关系的整体趋势，并且提供必要的交互手段来检查单独的连接关系。相应的，本文提出了基于双聚类的边绑定（biclustering-based edge bundling）算法，以减少神经元聚类之间大量连边带来的视觉混乱。

接下来，详细阐述多层次聚类，和有向无环图可视化中的两个主要部分：神经元聚类及神经元聚类连边的可视化。

3.3　多层次聚类

为了有效地展示一个大规模卷积神经网络对应的有向无环图，本文提出了多层次聚类，将该大规模有向无环图聚合成一个更加紧凑的图。

首先，对临近的中间层进行聚类（图3.3）。可能的聚类的方法不止一种。例如，

可以利用两个相邻网络中间层神经元响应的差别判断。这个差别可以用来大致衡量两个网络中间层的整体作用的差别。当这个差别很小的时候，可以将这样两个整体作用相似的中间层聚为一类。除此以外，还可以利用池化层来聚类，即将两个池化层之间的中间层聚为一类。

其次，将每个中间层中的神经元按照其在网络中的作用进行聚类（图3.3）。神经元的响应可以表示网络在处理一个样本时，该神经元起的作用。也就是说拥有相似响应的神经元在网络中起的作用相似。一个训练数据集中可能有上百万个样本，直接用一个神经元在每张图片上的响应作为特征，对神经元聚类是非常耗时的。因此，将这些响应按照样本的类别聚合为平均响应向量。

具体地说，假设训练样本分为 m 类 $c_1, c_2, ..., c_m$。类 c_i 中的训练样本可以表示为：$S_i = \{s_1^{(i)}, s_2^{(i)}, \cdots, s_{N_i}^{(i)}\}$。其中，$N_i$ 是类 c_i 中的训练样本的个数。为了获取响应，将训练样本 $s_j^{(i)}$ 作为网络输入，并获取其在每个神经元 n 上的响应 $a_n(s_j^{(i)})$。然后，计算神经元 n 在类 c_i 上的平均响应 $a_n(c_i)$：

$$\frac{1}{N_i}\sum_{j=1}^{N_i} a_n(s_j^{(i)}) \tag{3-1}$$

在此基础上，将每一个平均响应 $a_n(c_i)$ 聚合成一个平均响应向量：$\vec{a_n} = [a_n(c_1), a_n(c_2), ..., a_n(c_m)]$。最后，将这个向量作为特征将神经元进行聚类。在聚类算法的选择上，采用两种广泛使用的聚类方法：K-Means[45]（含参聚类）以及 Mean-Shift[46]（无参聚类）。其中，第二种聚类方法不需要提前知道聚类的数量。因此，第二种方法适用于专家预先不知道一层中神经元聚类数量的情况。为了提供神经聚类的概览，我们从每一个神经元聚类中计算出离类中心最近的若干神经元作为代表神经元。

3.4　神经元聚类可视化：混合展现聚类的不同侧面

基于多层次聚类的结果，本文设计了一个混合可视化（图 3.4）来展示神经元不同方面的信息，包括神经元学到的特征、响应以及对网络性能的贡献。每一个神经元聚类用一个矩形代表（图 3.4A）。聚类中，一个矩形表示一个神经元学到的特征（图 3.4B1）。矩形的大小表示其对网络性能的贡献。因此，采用矩阵布局算法来计算神经元聚类中神经元学到的特征的位置。另外，本文采用矩阵可视化展示神经元聚类中神经元的响应（图3.4B2）。矩阵中每一个元素的颜色表示一个神经元在一个类别样本上的平均响应。专家可以选择在学到的特征和响应之间切换以便更好地分析神经元起到的作用。

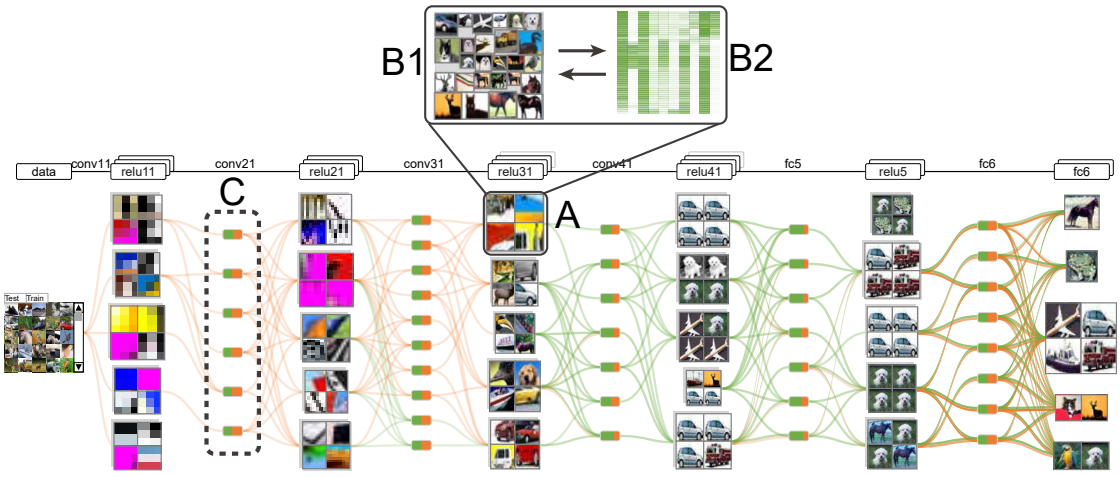


图 3.4　神经元聚类及其连边可视化概览

3.4.1　神经元学到的特征可视化

揭示神经元学到的特征。本文采用 Girshick 等[47] 提出的方法来揭示神经元学到的特征。采用这个方法的原因是其速度快，而且结果易于人类理解。具体地说，对于每一个神经元，计算其在大量图片块上的响应。经过对这些响应排序，找到若干能使这个神经元响应最大的图片块。将这些图片块作为这个神经元学到的特征展现给专家。其他揭示神经元学到的特征的方法[6,48] 也可以很容易地融入所开发可视分析算法中。

布局计算。展示上一步计算出特征的一个直接想法是用等大的矩形代表计算出的图片块，并且用基于网格的布局方法计算位置[6,43]。然而这个方法不能强调在网络中起到重要作用的神经元。

为了解决这个问题，首先计算出每个神经元的重要性。本文提出可以用一系列方法来计算一个神经元的重要性，比如其最大的或者平均响应，亦或是对于分类的贡献[49]。在此基础上，用矩形的大小表示一个神经元的重要性，并将计算矩形位置的问题建模成一个矩形布局问题（rectangle packing），目的是让布局结果尽量紧凑。

现有的最优矩形布局算法[50-51] 能够处理少量的矩形布局问题（例如，15 个矩形的位置可以在 0.1 秒之内计算出来）。但是当矩形的个数线性增长的时候，计算时间会呈指数级增长。例如，25 个矩形的位置需要超过一个小时计算出来[50]。由于一个神经元聚类可能含有数十乃至数百神经元，现有的矩形布局算法无法直接使用。

为了解决这个问题，本文提出了层次化矩阵布局算法。该算法的基本思想是将大量矩形的布局问题分解为一系列小的矩形布局问题。其中每个小的矩形布局

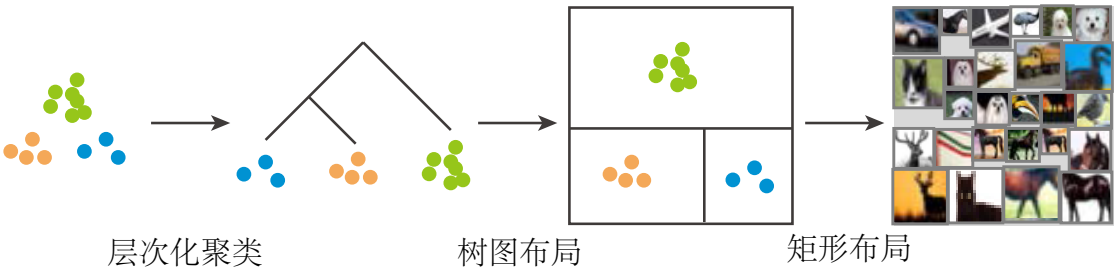


图 3.5　层次化矩形布局算法概览

问题都可以用传统的最优矩形布局算法[50] 求解。最后将这些结果合并为最终的布局。具体地说，层次化矩形布局算法包含以下步骤（图3.5）。

第一步：层次化聚类。在这一步中，将得到的图片块进行分裂式层次化聚类，即首先将所有图片块聚合为一个大类，然后逐步将这个类分裂为更小的类，直到每个小类中的图像块个数小于阈值。本文采用广泛使用的图聚类算法[52] 进行聚类的分裂操作。该算法利用神经元间的相似度将聚类中的神经元建模为图。在聚类分裂的时候，该算法会最大化聚类内部图像块的相似度，并同时最小化不同聚类的图像块的相似度。

第二步：计算每个小类所占的空间以及位置。基于上述层次化聚类的结果，采用树图（treemap)[53] 计算每个小类所占的空间以及位置。与上述图聚类算法相似的，树图算法根据输入的层次化聚类结果，不断分割待布局的区域，以得到每个小类所占的空间及位置。

第三步：计算每个小类中图片块的位置。最后，利用最先进的矩形布局算法[50] 计算每个小类中图像块的位置。该算法是一种树搜索算法，不断尝试搜索可能的矩形放置位置。为了尽量减少需要搜索的矩形放置位置，该矩形布局算法利用了一系列搜索剪枝技术。

3.4.2　神经元响应可视化

在最初的实现中，我们简单地用每个神经元的大小（即其图片块的大小）表示它在所有类别样本上的平均响应。然而，合作的专家们对这个表示方法不满意，原因是没有办法分析神经元对于不同类别样本起到的不同作用。为了满足这个需求，将一个神经元聚类中平均响应向量聚合成一个响应矩阵，其中每一行都是一个平均响应向量。基于这种表达，本文采用矩阵可视化的方法展示这个响应矩阵。具体地说，第 i 行第 j 列的元素表示第 i 个神经元 n_i 在第 c_j 类上的平均响应。专家们整体上对这个设计比较满意，因为这个设计能够帮助他们宏观地看到一个神经元聚类中的神经元在不同类上的响应模式。他们对这个设计不满意的地方在于，

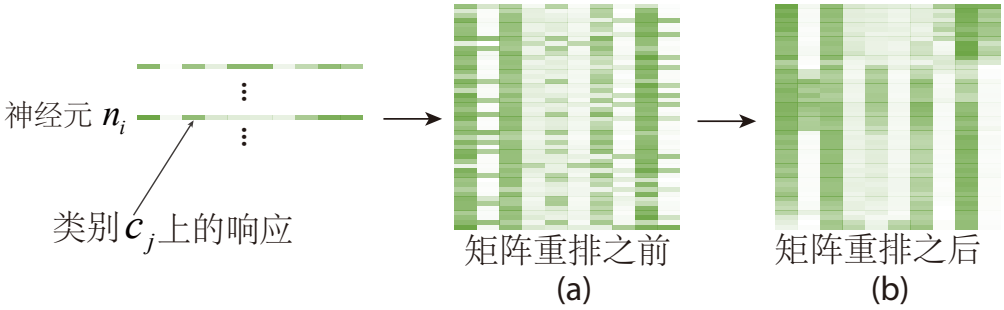


图 3.6　矩阵重排算法效果示意图：（a）重排之前；（b）重排之后

现在的可视化结果不能展示，在一个聚类内部，神经元响应更细粒度的聚类特征。为了解决这个问题，本文提出了一个矩阵重排算法（图3.6），以揭示矩阵中神经元响应更细粒度的聚类特征。

矩阵重排。为了比较不同的神经元聚类，在不同的神经元聚类之间，列（类别）的顺序应该是相同的。因此，只对矩阵中的行（神经元）进行重排。

矩阵重排算法的基本思想是找到一种排列方式能够最大化相邻两个神经元的相似度。这样能够将具有相似响应的神经元尽量排在一起，从而揭示出神经元聚类中的聚类特征。具体地说，给定一个神经元聚类中 $C = \{n_1, n_2, \cdots, n_{N_C}\}$，重排的目标是对于任意神经元 n_i 找到其最适合的位置 $\pi(i)$ 使得最终的排列结果能够更好地揭示聚类特征。对于矩阵中的第 r 列，定义其对应的神经元为 $n_{\pi^{-1}(r)}$。为了达到这个目标，希望最大化相邻两个神经元的相似度：

$$max \sum_{r=1}^{N_C-1} sim(n_{\pi^{-1}(r)}, n_{\pi^{-1}(r+1)}) \tag{3-2}$$

其中，$sim(\cdot,\cdot)$ 是相邻两个神经元的相似度。本文采用广泛应用的余弦距离计算神经元的相似度。这个组合优化问题可以用 Held-Karp 算法[54] 在 $O(2^{N_C} \cdot N_C^2)$ 的时间复杂度内求解，其中 N_C 是神经元的个数。直接用 Held-Karp 算法的问题是一个神经元聚类中可能含有上百个神经元，这样计算时间会很长。因此，本文提出了一个基于分治算法的加速策略，具体包含以下几步：

分解。如果一个神经元聚类中神经元个数过多，以至于不能直接用 Held-Karp 算法求解，那么首先将这个神经元聚类用广泛应用的图聚类算法[52] 分解为若干小类。

解决。利用 Held-Karp 算法计算每个小类中神经元的顺序。该算法利用动态规划，根据 $n-1$ 个神经元的顺序求出 n 个神经元的顺序。即从 2 个神经元开始，逐步计算出小类中神经元的顺序。

合并。将每个小类的顺序合并为最终的顺序。即将每个小类首尾相接，形成最终的矩阵重排结果。

图3.6展示了矩阵重排算法结果的一个例子。能够看到重排之后的矩阵中出现了若干类不同模式的响应。

3.4.3　交互

为了帮助专家更好地理解神经元聚类的不同侧面，支持以下交互：

交互式修改聚类结果。聚类算法可能不完美，另外不同的用户可能有不同的需求。因此应该支持专家交互式地修改聚类的结果。例如，在一个主要检测黑白局部特征的神经元聚类中，出现了一个检测彩色局部特征的神经元。为了更好地比较这些聚类，专家可以将这个神经元移动到主要检测彩色特征的神经元聚类中去。

选择并展示一部分神经元。一个卷积神经网络中可能包含数以千计的神经元。因此，支持专家选取并展示一部分神经元是很有必要的。我们支持专家选取一部分的图片类别，并着重展示在这部分图片上响应很大的神经元。其他神经元作为上下文，被设定为半透明状态。

在神经元不同方面信息间切换。浏览神经元的不同方面的信息有助于专家全面理解神经元在网络中的作用。显示一个物体多个方面信息的方法之一是把一个方面的信息叠加在另一个方面之上[55-56]。然而，每一个神经元的某一方面已经包含足够大量的信息了。将若干方面叠在一起，会导致专家没法看清每一个方面的信息。专家也表示，他们更注重分析每一个单独方面上的信息。因此，我们支持专家在不同方面的信息之间切换。例如，专家可以从默认的学到的特征视图切换浏览响应矩阵。

3.5　神经元聚类连边可视化：基于双聚类的边绑定

为了减少神经元聚类之间大量连边导致的视觉混乱，本文提出了一个基于双聚类的边绑定技术。对于每一个网络中间层，首先计算其输入神经元聚类与输出神经元聚类的双聚类。受 BiSet 系统[57] 启发，同样地在输入神经元与输出神经元之间添加了一个虚拟中间层（图 3.4C）。在这个虚拟中间层中，每一个双聚类都被看做一个有向无环图中的节点，并用一个小矩形表示。

3.5.1　基于双聚类的边绑定

在最初的实现中，我们将每两个神经元之间的连接（DAG 中的边）用一条曲线表示。但是，数百万条边会导致严重的视觉混乱。

为了减少视觉混乱，首先尝试了基于位置的边绑定技术[58-59]。经过试用，合作的专家们表示这种方法的确能显著地降低视觉混乱。然而他们认为基于位置的边

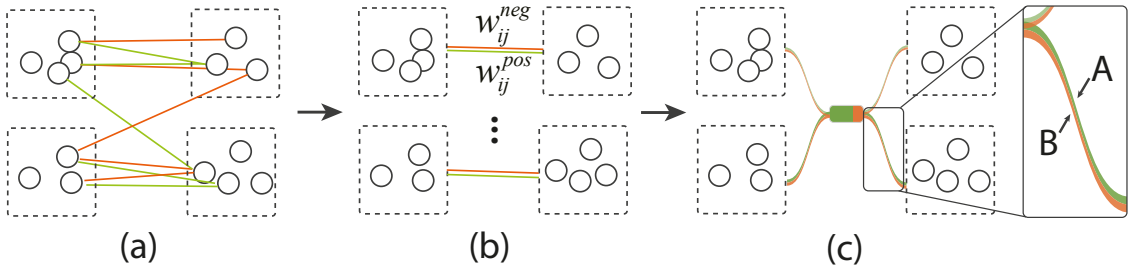


图 3.7　利用基于双聚类的边绑定方法展示神经元聚类连边

绑定技术计算出的边聚类不具有实际意义，不能够将相似权值的边聚在一起。另外，他们对权值较大的边更感兴趣，因为这样的边表示输入神经元对输出神经元有较大的影响。

为了满足这个需求，本文提出了一个基于双聚类的边绑定算法，将具有相似权重的边绑定为一个聚类。具体地说，一个双聚类表示一些输入神经元聚类以及一些输出神经元聚类。这个方法能够在视觉上将不同的神经元聚类连接在一起。本文所提出的双聚类算法是基于 BiSet 系统中提出的双聚类算法。经过调研，其不能直接应用在神经元连边上，因为 BiSet 中的双聚类算法要求边是没有权重的。然而在卷积神经网络中，每一条边都是带权的而且需要将相似权重的边聚在一起。如果简单地采用 BiSet 中的聚类算法，有可能无法找到一些具有较大权值，但是边数较少的聚类。为了解决这个问题，本文提出了一个能够处理带权边的双聚类算法。如图3.7所示，该算法含有以下几步：

第一步：计算两个神经元聚类的连接强度。 首先计算两个神经元聚类 C_i 和 C_j 之间的连接 e_{ij} 的强度 w_{ij}。$E = \{e_{ij}\}$ 表示两层神经元之间的边集。计算 w_{ij} 的一个简单想法是将连接 C_i 和 C_j 的连边的权值进行平均。这个算法的问题是如果两个神经元聚类之间有权值为正和负的边，且这两种边的数量相当，那么计算出的权值会接近于零。在这种情况下，这种计算方式会导致误解。因此，将两个神经元聚类之间的连接强度表示为一个二维向量：$\vec{w}_{ij} = [w_{ij}^{pos}, w_{ij}^{neg}]$。其中，$w_{ij}^{pos}$ 表示两个神经元聚类之间权值为正的边的权值的平均值，w_{ij}^{neg} 表示两个神经元聚类之间权值为负的边的权值的平均值。

第二步：双聚类。 基于上一步中计算出的神经元聚类的连接强度，将网络中一层对应的输入和输出神经元进行双聚类。因为专家们对权值的绝对值大的边比较感兴趣，不能简单地采用针对无权值边的双聚类算法。因此，首先在 $W = \{w_{ij}^{pos}\} \cup \{|w_{ij}^{neg}|\}$ 中计算权值的绝对值最大的边 w_{max}。如果 $w_{max} \in \{w_{ij}^{pos}\}$，在权值为正的边中找满足下列条件的边：$|w_{ij}^{pos} - w_{max}| < \tau$。其中，$\tau$ 是用户给定的参数，表示权值相似度的容忍度。如果 $w_{max} \in \{|w_{ij}^{neg}|\}$，在权值为负的边中进行类似操作。对

于这些边，利用广泛使用的 Apriori 算法[60] 计算闭合项集（closed item set），并将找到的闭合项集作为双聚类。在此之后，将计算出的双聚类相关的边从边集 E 中去掉，并重复上述过程直到 w_{max} 小于用户给定阈值。

第三步：边绑定。在这一步中，利用计算出的双聚类进行边绑定，以减少视觉混乱。受 BiSet 系统[57] 启发，我们在输入和输出神经元之间也添加了个“中间”层（图3.7(c)）。在这一层中，每一个双聚类都用一个矩形表示。我们将每一个双聚类分为两个部分（红色和绿色）。其中，红色的部分与绿色的部分面积的比值，表示这个双聚类中负权值边与正权值边的数量的比值。每一条连接双聚类与神经元聚类的边都由两条曲线构成（图3.7A，B 所示）。其中，红色和绿色的曲线分别表示负权值边与正权值边的平均值。为了进一步减少视觉混乱，专家可以过滤掉一部分平均权值较小的边。

3.5.2　交互

展示调试信息。调试信息可以帮助专家诊断一个失败的训练过程。而调试网络的时候，他们会经常检查诸如梯度，响应等调试信息。除此之外，其他衍生出来的调试信息也经常被其他专家使用，例如网络中权值的相对变化。他们所遇到的问题是这部分调试信息往往非常繁杂，例如，网络训练中的梯度可能有数百万个。逐个检查这些信息非常费时费力。因此，本文提供这些调试信息的概览，并支持专家从不同粒度分析调试信息。例如，可以用边的颜色编码导数信息。另外，还可以将这部分信息进一步聚合，显示为折线图，从而浏览每一层上整体的导数情况。

3.6　算法应用：CNNVis 系统

为了验证所提出可视分析方法的有效性，我们基于该算法，开发了 CNNVis 系统。该系统能够帮助专家分析与理解卷积神经网络训练过程中一个时间片上的运行状态。我们与两位深度学习专家合作，以案例分析的形式用该系统分析了一系列卷积神经网络。专家 E_1 的主要研究方向为深度生成模型。由于深度生成模型中经常会以卷积神经网络作为子模块，其对卷积神经网络相对熟悉。专家 E_2 的研究方向是利用深度神经网络进行概念学习。案例分析的结果表明，所提出的可视分析方法能够有效地展现卷积神经网络训练过程的单个时间片，帮助专家理解影响卷积神经网络性能的重要因素，以及诊断一个失败的训练过程。

3.6.1　系统简介

CNNVis 包含以下组成部分：

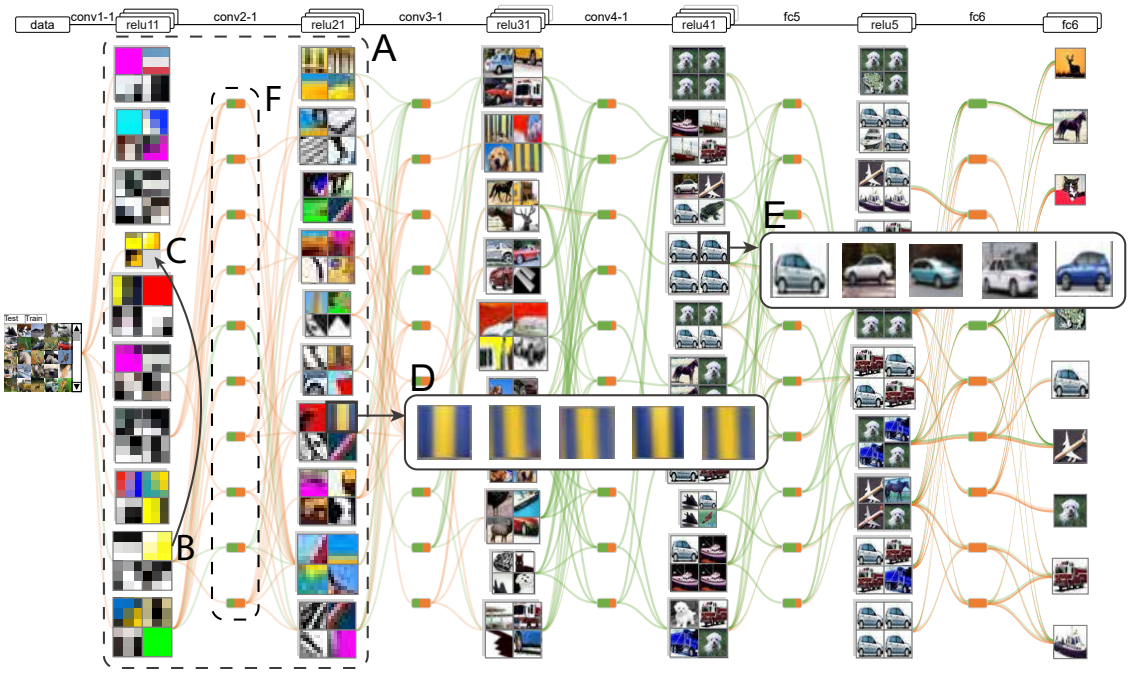


图 3.8　CNNVis 可视化样例

- **有向无环图建模模块**：将卷积神经网络的结构转化为一个有向无环图，并利用多层次聚类算法对网络中间层以及每层的神经元进行聚类，从而提供网络结构的概览；
- **神经元聚类可视化模块**：展示神经元的不用方面的信息：神经元学到的特征、响应和对网络性能的贡献；
- **基于双聚类的边绑定模块**：减少大量的神经元聚类连边带来的视觉混乱；
- **交互模块**：提供一系列的交互帮助专家更好地分析与理解网络，例如聚类调整以及根据需求显示调试信息。

一个可视化结果如图3.8所示。最左侧是网络的输入，最右侧是网络的输出；每一列代表一个代表性的中间层；每一列中一个大矩形代表一个神经元聚类；其中每一个小矩形代表一个神经元。在两层神经元中间，我们增加了一个中间层，来代表基于双聚类的边绑定结果（图3.8F）。利用这个可视化，可以很快地了解网络运行状态的概况，例如每层神经元学到的特征。注意到，在这个网络中，底层神经元学到的是底层特征，例如色块等（图3.8A）。可以看到在一个主要检测黑白局部特征的神经元聚类中，出现了一个检测彩色局部特征的神经元（图 3.8B）。为了更好地比较这些聚类，可以将这个神经元移动到了主要检测彩色特征的神经元聚类中去（图3.8C）。

3.6.2　案例分析

案例分析的基本思路是利用一个广泛应用的卷积神经网络，在其网络结构上进行一系列修改，利用 CNNVis 比较分析修改带来的影响。如果修改之后的网络结构不尽如人意，便利用 CNNVis 找到出现问题的原因，并进行修改。

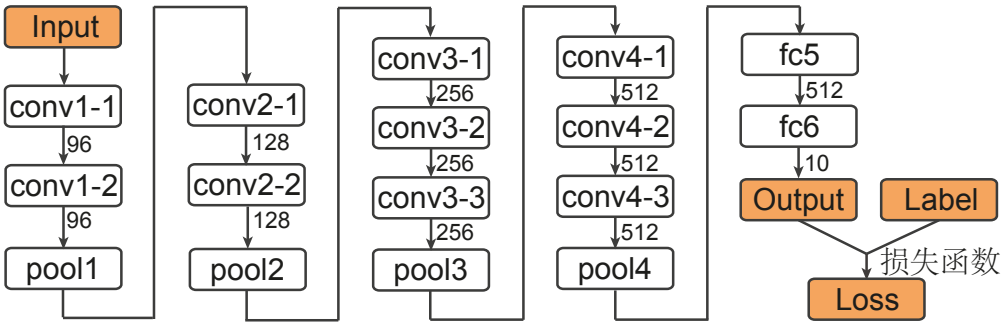


图 3.9　BaseCNN 的结构示意图

基本的卷积神经网络。专家 E_2 提供了该案例分析所需的基本卷积神经网络。为了简单起见，将其称为 BaseCNN。BaseCNN 来源于一种广泛应用的网络结构：VGGNet[61]。该结构广泛地应用于图像分类等一系列任务。BaseCNN 含有 10 个卷积层和 2 个全连接层。这些卷积层被组织成 4 个卷积层组，每个组分别包含 2,2,3,3 个卷积层。每一个卷积层组的最后都有一个最大池化层（max-pooling layer）。专家使用了广泛使用的 ReLU（rectified linear unit)[62] 作为响应函数。为了衡量网络的性能，专家使用了常用的交叉熵（cross-entropy）作为损失函数。BaseCNN 的整体结构如图3.9所示。图中，每个中间层下面的数字表明中间层中的通道数

为了验证 BaseCNN 的性能，专家在标准数据集 CIFAR10[63] 上做了实验。CIFAR10 数据集含有 60000 张 32*32 的彩色图片。每张图片隶属于 10 个类中的一个（例如，飞机，猫等）。每个类含有 6000 张图片。整个数据集被分为 50000 个训练样本（训练集）以及 10000 个测试样本（测试集）。整个实验是在业界广泛使用的深度学习框架 Caffe[64] 上进行的。在测试集上，BaseCNN 的错误率为 11.32%。

案例分析的设计思路。在第一个案例分析中，专家 E_1 设计了若干 BaseCNN 的变形网络，并利用 CNNVis 研究了网络结构如何影响网络性能。专家表示，这方面的分析有助于理解为什么不同的网络结构会导致不同的网络性能。

在第二个案例分析中，专家利用 CNNVis 系统诊断了一个失败的训练过程。在这个训练过程中，专家 E_2 尝试更换 BaseCNN 的损失函数。然而，新的网络的训练过程不能收敛。经过一些未能成功的尝试之后，专家 E_2 希望用 CNNVis 系统辅

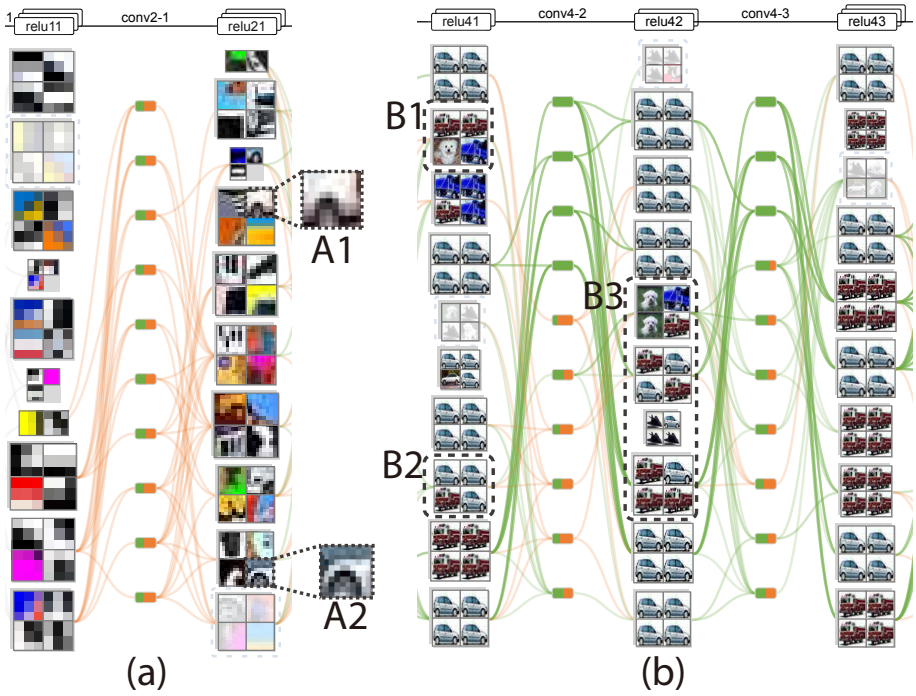


图 3.10　BaseCNN 中神经元学到的特征：(a) 底层特征，例如轮子；(b) 高层特征，例如整个车辆

助进行调试过程。

3.6.2.1　理解网络结构对网络性能的影响

我们与专家 E_1 合作完成了这个案例分析。在这个案例分析中，E_1 在一系列的 BaseCNN 变形网络上测试并分析了网络结构对网络性能的影响。在此基础上，E_1 还探寻了利用 CNNVis 选择一个合适的网络结构的可能性。虽然在标准数据集上有很多高性能的网络可供选择，但是面对一个新的数据集的时候，研究者们还是需要大量的时间来选择一个合适的网络结构。因此，E_1 认为研究网络结构对网络性能的影响，能够帮助他选择合适的网络结构。

BaseCNN 概览。 E_1 首先从 BaseCNN 开始分析。从 BaseCNN 的概览中，专家发现，底层的神经元学出的特征往往是低层的特征，例如边，角，色块等（图3.8A）。在现有的工作中，也有类似的现象出现[37]。通过在能使这部分底层神经元响应最大的图片块中浏览，专家发现这些图像块的差别很小（图3.8D）。而在高层的神经元中，专家发现这部分神经元能够检测出高层的特征，例如一辆车（图 3.8E）。基于这些观察，专家总结道：“能够从底层逐渐检测出高层的特征是卷积神经网络一个很重要的性质，而 CNNVis 能够很好地展现这个性质”。

为了从更细粒度分析 BaseCNN，E_1 进一步地选择两个相似的类：轿车与卡车，

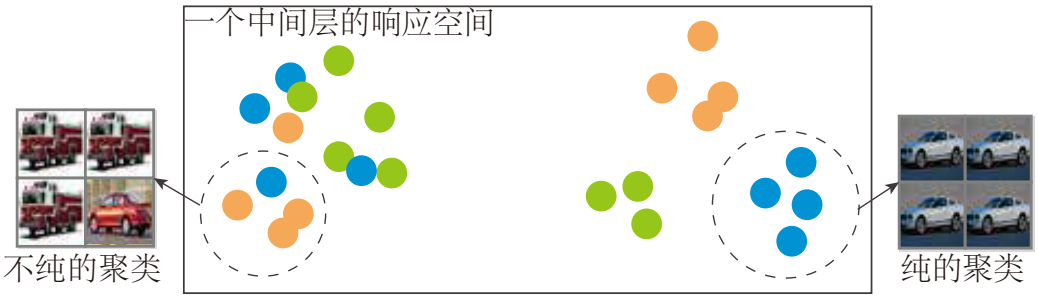


图 3.11　响应空间概览：‘‘不纯’’ 的神经元聚类，以及 ‘‘纯’’ 的聚类。

表 3.1　不同深度的 BaseCNN 变种的性能比较

网络结构	错误率	卷积层数	总层数
ShallowCNN	11.94%	7	30
BaseCNN	11.33%	10	40
DeepCNN	14.77%	20	70

并观察与这两类相关的神经元响应特征。从底层神经元学到的特征上，专家发现了一些轿车和卡车共有的部分，例如轮子（图3.10(a) 中的 A1 和 A2）。但是，专家认为这些特征不足以区分这两个类。因此，他展开了第四个卷积层组进行进一步检查（图3.10(b)）。专家 E_1 注意到 “不纯” 的神经元聚类越来越少（图3.10(b) 中的 B1-B3）。这里所说的 “不纯” 的聚类指的是能使一个神经元聚类中的神经元相应最大的图片块来自于不同的类。检查聚是否 “纯净” 能够帮助专家判断一个网络区分不同类别图片的能力。在一个 “纯” 的聚类中，所有相同类别的图片块在一层对应的响应空间中聚集在一起。在网络的底部中间层中，希望看到 “不纯” 的聚类，因为希望神经元尽可能地检测多样化的特征。而在网络的顶部中间层中，希望看到 “纯” 的聚类，因为希望一个网络能够将不同的类别分的比较开。在这种情况下，属于不同类别的图片块几乎不会出现在同一个聚类中。专家表示，这个准则也适用于其他结构的卷积神经网络。图3.11展示了这个准则。例如，在 BaseCNN 的最顶层卷积层中，所有的聚类都是 “纯” 的。这表示这些图片的响应和它们的类别对应的比较好，也间接证明了 BaseCNN 良好的分类性能。

网络深度。 E_1 进一步研究了网络深度对神经元学到的特征的影响。他基于 BaseCNN 设计了若干变种网络，并与 BaseCNN 进行比较。具体地说，他去掉 BaseCNN 最后一个卷积层组，得到了更浅的网络 ShallowCNN，以及将 BaseCNN 中所有卷积层的数量加倍，得到了 DeepCNN。Table 3.1展示了以上变种网络的结构和性能。

基于 BaseCNN 的分析，他选择了卡车和轿车这两个类，并展开了 ShallowCNN

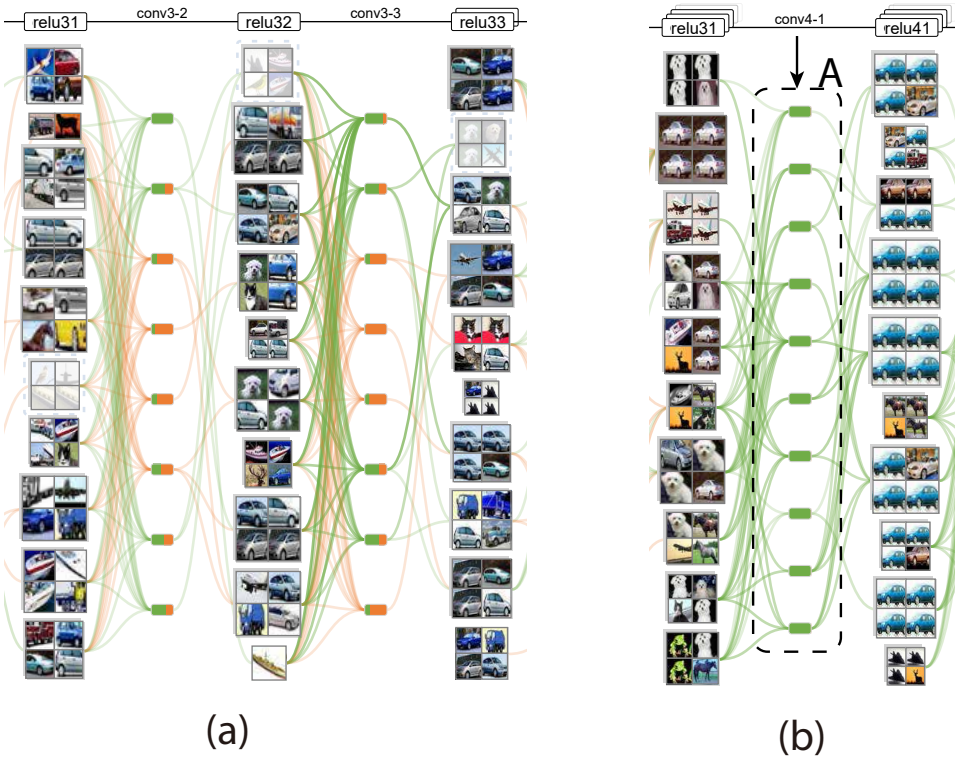


图 3.12　网络深度对模型的影响：(a)ShallowCNN 高层的特征，含有很多不纯的神经元聚类；(b)DeepCNN 中含有一个权值几乎全为正的中间层

中最后一个卷积层组（图3.12(a)）。他注意到，相比于 BaseCNN，这个浅层网络的顶层中的确含有更多“不纯”的神经元聚类。这意味着，一个卷积神经网络如果深度不足，会导致它不能有效地区分相似类别的图片，也就意味着性能上会有所欠缺。在 DeepCNN 中，专家 E_1 注意到，最后一个卷积层组中第一个卷积层的边几乎都是绿色的（图3.12(b) 中的 A）。这意味着这一层中所有的权值都是正的。由于在 DeepCNN 中，每一个卷积层的输入都是上一个卷积层的响应函数（ReLU）的输出。而 ReLU 的输出一定为非负，因此每一个卷积层的输入都是非负的。如果这个卷积层的权值几乎都是正的，那么响应函数 ReLU 会几乎不起作用，也就导致这个卷积层可以看做一个近似线性函数。进一步展开对应的卷积层组，可以发现，有若干卷积层出现了类似的现象，即权值几乎都是正的（图3.13）。由于线性函数的复合依然是线性函数，专家表示这说明网络中存在着冗余现象。这种冗余会减慢训练过程，并使网络更容易陷入不好的局部最优解中，从而导致性能下降。这个发现与之前的研究也是一致的[65]。专家 E_1 最后表示，CNNVis 可以用来方便地定性检查 CNN 抽取的特征的抽象程度。

网络宽度。网络的宽度是另一个影响网络性能的重要因素。为了深入理解网络宽度的影响，专家 E_1 在 BaseCNN 的基础上构建了一系列不同宽度的网络 BaseCNN×w，

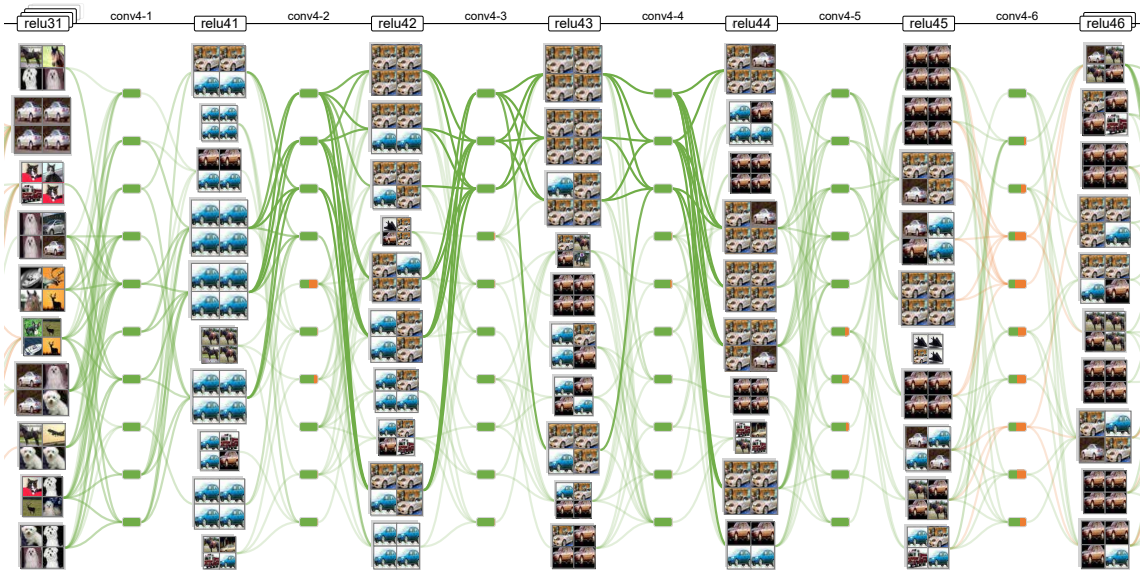


图 3.13　DeepCNN 网络冗余现象示意图

表 3.2　BaseCNN 不同宽度的变种网络的性能比较。

网络结构	错误率	参数数量（百万）	L_{train}	L_{test}
BaseCNN×4	12.33%	4.22M	0.04	0.51
BaseCNN×2	11.47%	2.11M	0.07	0.43
BaseCNN	11.33%	1.05M	0.16	0.40
BaseCNN×0.5	12.61%	0.53M	0.34	0.40
BaseCNN×0.25	17.39%	0.26M	0.65	0.53

并与 BaseCNN 进行比较。w 表示对应网络中每层神经元与 BaseCNN 每层神经元数量的比值。例如，BaseCNN×4 表示其神经元数量为 BaseCNN 的四倍。在案例分析中，w 取自 $\{4, 2, 0.5, 0.25\}$。表3.2列出了上述 BaseCNN 变种网络的结构和性能。其中，L_{train} 和 L_{test} 分别表示训练集和测试集上的损失函数值。

与 BaseCNN 相比，更宽的网络 (BaseCNN×4) 在训练集上的损失函数值更小，而在测试集上的损失函数值更大。这个现象称为过拟合（overfitting）。过拟合现象来源于模型尝试拟合数据中很小的变化，而这种很小的变化往往是噪音。过拟合现象常常出现在与训练数据相比网络含有过多参数的情况。当过拟合现象出现的时候，一个直接的表现就是训练集上的损失函数会远大于测试集上的损失函数。E_2 想要利用 CNNVis 研究卷积神经网络中过拟合的影响。因此，他首先载入了过拟合现象最严重的的网络 BaseCNN×4。

在观察了高层的特征之后，专家没有找到与 BaseCNN 相比不同的地方。所以他转而检查底层的特征，进而发现有若干神经元学到了相同的特征（图3.14(a) 中

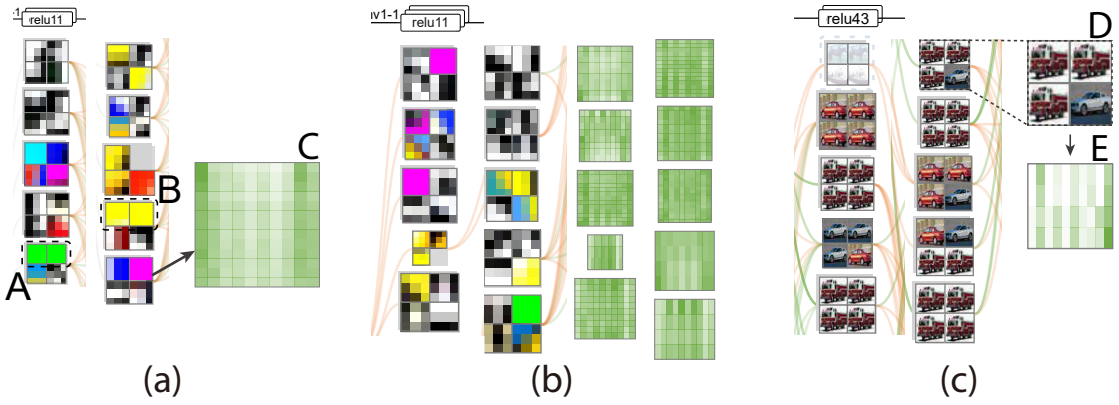


图 3.14　不同宽度网络学到的特征：(a) 宽网络 BaseCNN×4 学到的底层特征；(b)BaseCNN 的低层特征；(c) 窄网络 BaseCNN×0.25 的高层特征

的 A 和 B）。这意味着在一个过拟合的卷积神经网络中可能出现冗余的神经元。为了验证这个猜想，专家进一步检查了出现冗余神经元的聚类。与 BaseCNN 的底层响应（图3.14(b)）相比，他发现很多神经元的响应非常相似（图3.14(a) 中的 C）。这个现象进一步验证了在一个过宽的网络中，可能出现冗余的神经元。

E_1 表示，他们经常使用一些定量的信息，比如准确率，来衡量一个网络的质量。但是这种定量的信息不足以清楚地指导如何修改网络。即使当专家知道一个卷积神经网络出现了过拟合，也很难决定要去减少每层神经元的数量还是删掉某些层。而 CNNVis 可以帮助他们找到可能要修改的中间层。这对于加快他们的开发是很有帮助的。

在这之后，E_1 比较了 BaseCNN 与窄模型（BaseCNN×0.5 和 BaseCNN×0.25）的性能。这些窄模型在训练集和测试集上的损失函数值相差很小。这意味着这些模型能够很好地将训练集上学到的知识迁移到测试集上。但是他们的性能比 BaseCNN 差一些 (表3.2)。专家解释说这个现象被称为欠拟合（underfitting）。欠拟合现象发生在任务比所使用的模型更复杂的情况。这导致网络的准确度不能满足要求。为了检查除了准确度下降，欠拟合还会带来什么影响，专家载入了 BaseCNN×0.25 网络进一步分析。

受 BaseCNN 分析的启发，他选择了两个相似的类：卡车和轿车。分析了底层的神经元之后，专家没有发现与 BaseCNN 过多不同的地方。因此他转而分析高层的神经元。专家发现在最后一个卷积层中有若干“不纯”的神经元聚类。例如，图3.14(c) 中的神经元聚类 D 的代表图片块中含有三个卡车的图片块和一个轿车的图片块。进一步检查这个神经元聚类的响应（图3.14(c) 中的 E），专家发现这个聚类的响应可以分为两个子类（卡车和轿车）。这意味着，这个神经元聚类很难区分这两个类的图片。也就导致这个窄网络的性能不如 BaseCNN。

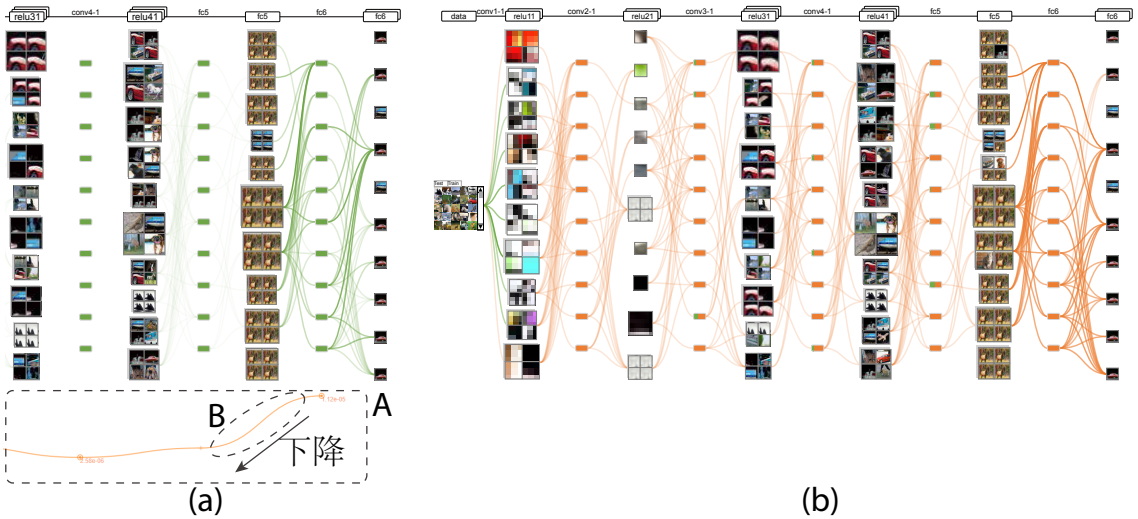


图 3.15　神经元间连边的概览：(a) 边的颜色表示相对上一个时间点，权值的相对变化，下方对应的折线图也展现了这个信息在中间层级别的概况；(b) 边的颜色表示权值本身

专家 E_1 表示，选择一个合适的网络结构很难。尤其是面对一个新的数据集时，由于类似成熟的网络参考，选择网络的深度和宽度更加困难。在这种情况下，专家往往不得不尝试一系列参数，来找到一个满意的网络结构。CNNVis 能够帮助专家直观地检查一个网络质量，并提供进一步修改的依据。

3.6.2.2　诊断失败的卷积神经网络训练过程

这个案例分析展示 CNNVis 如何帮助专家 E_2 诊断一个卷积神经网络失败的训练过程。最近 E_2 根据之前的研究成果[66]，设计了一个 BaseCNN 的变种网络，以期提高 BaseCNN 的性能。具体地说，他把损失函数换成了合页损失函数（hinge loss）。然而，新的网络训练失败了，具体体现在训练过程会卡在损失函数值约为 2.0 的位置。这个时候网络远未达到满意的性能。

为了帮助专家诊断这个失败的训练过程，他载入了训练卡住之后的一个时间片。由于在平时的诊断过程中，他一般从权值的相对变化入手，因此 E_3 将边的颜色设定成表示权值相对于上一个时间片的相对变化。

专家发现从网络的顶层开始，权值的相对变化迅速减小，到倒数第二层之后就几乎为 0 了 (图3.15(a))。网络下方表示每层中权值的平均变化的折线图 (图3.15(a) 中的 A 和 B) 也验证了这个现象。这个现象导致网络的训练卡在这个地方。E_2 想知道是什么原因导致了权值的变化如此小，因此他将边的颜色设定为表示权值本身。他立刻发现几乎所有的权值都是负的 (图3.15(b))。为了进一步分析负的权值对网络训练带来的影响，他首先观察了网络中神经元学出的特征。但是由于网络训练失败，神经元学到的特征没有太大意义。因此，E_2 切换显示神经元的响应，并发现

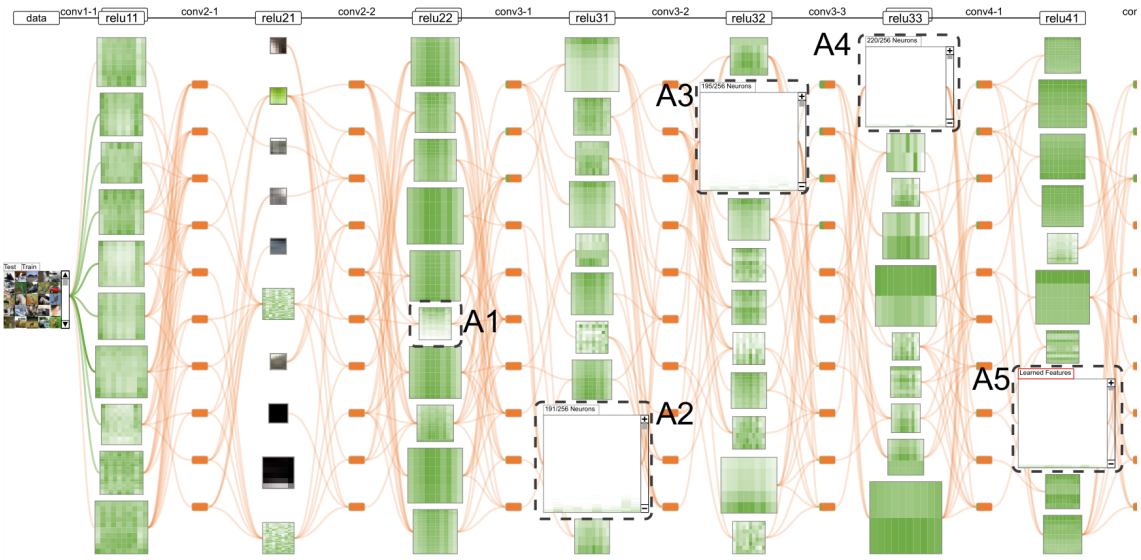


图 3.16　出现非激活神经元的网络示意图

有些神经元的响应矩阵中有若干行是白色。这意味着对应的神经元在所有类上的响应都是 0。进一步分析这个现象，他展开了网络中第二、三、四卷积层组。他发现从网络的浅层到深层，响应为 0 的神经元的数目不断增长 (图3.16中的 A1-A5)。类似神经元被称作非激活神经元。由于这些神经元的响应函数为 ReLU，专家进一步检查了这些神经元对应的响应函数的输入，发现它们全为负数。至此，E_2 分析出了为什么训练过程为什么卡在这个地方。

他向我们解释，由于每个卷积层的输入是上一层响应函数 ReLU 的输出，这个输入必为非负（$ReLU(x) = max(0, x)$）。由于这些卷积层的权值基本为负，这些层的 ReLU 的输入基本为负。进而，ReLU 的输出基本为 0，也就是神经元的响应基本为 0。又因为采用了基于随机梯度下降[44] 的训练方法，神经元的 0 响应意味着权值的 0 更新。

在分析了为什么网络训练会卡在这个地方之后，E_2 提出了一个让网络训练继续进行的方法。具体地说，他在每个卷积层以及全连接层的的响应函数之间，增加了一个批归一化（batch-normalization）操作。一个批归一化操作可以将响应函数的输入归一化到均值为 0，方差为 1。这样，响应函数 ReLU 的输入不会几乎都是负值了。也就是说，即使网络中的权重都是负的，模型也可以继续训练。

修改后的模型在 CIFAR-10 数据集上达到了 9.43% 的错误率，相较于 BaseCNN 提高了 1% 以上。专家 E_2 表示，为了调试这个网络，他之前用了很多方法，比如打印各种调试信息。看这些调试信息很费时费力。不仅如此，他也尝试了很多修改方法，但是这些修改方法都没有用。而 CNNVis 的确帮助他成功地改好了这个网络。专家进一步表示，CNNVis 的优点之一在于它能够帮助用户从多个侧面方便地

浏览一个网络，也能够方便地检查一些训练中的调试信息，而不用手动插入代码。

3.7　讨论及小结

局限性。 虽然上述案例分析证明了所提出可视分析方法的有效性，但是该算法也有一些局限。

首先，该算法不能用于分析无法建模成有向无环图的深层模型。循环神经网络（recurrent neural network，RNN）[17] 是其中一例。在一个循环神经网络中，神经元之间的连边形成一个有向环，因此循环神经网络的结构无法之间建模成一个有向无环图。对于循环神经网络，这个问题可以用将其展开成一个非常深的有向无环图来解决[17]。但是展开后的有向无环图过深会引入额外的计算开销。

其次，响应矩阵的可扩展性有所欠缺。当数据集中含有过多类时，响应矩阵会含有太多的列而无法有效地浏览。随着卷积神经网络的发展，机器学习专家会接触到含有成百上千类别的数据集。例如，ImageNet 数据集中就含有超过 1000 个类[37]。这个问题可以通过将类别聚类，并将同一类别中的响应聚合来解决。

最后，论文中采用的 K-Means 算法，无法展示出神经元聚类中的异常值（例如异常的神经元）。而异常值在很多情况下对于分析来说是非常重要的。这个问题可以用两种方法解决。其一是利用第五章所述的不确定性符号，将聚类结果的不确定性（含有异常值的聚类不确定性高）展示给专家。专家可以主动浏览不确定性高的聚类，从而分析聚类中的离群点。其二是利用类似第四章所述的蓝噪声采样技术，在展示整体趋势的同时保留异常值。

小结。 本章提出了基于多层次聚类和有向无环图的可视分析方法，帮助专家分析与理解卷积神经网络训练过程中单个时间片上的运行状态。该方法将一个卷积神经网络建模成一个有向无环图，并将该有向无环图利用多层次聚类聚合为更紧凑的图，以处理大规模卷积神经网络。聚合后的有向无环图中，每个节点是一个神经元聚类，边表示神经元聚类间的连边。对于神经元聚类，展现了聚类的多种信息，帮助专家从多个侧面分析神经元。具体地说，本文提出了一个层次化矩形布局算法展示神经元聚类中神经元学到的特征。本文还提出了矩阵重排算法，以展现神经元聚类内部的响应。对于神经元聚类间的连边，本文提出了基于双聚类的边绑定算法，以减少神经元之间大量连边带来的视觉混乱。我们基于该算法开发了 CNNVis 系统，帮助专家理解了影响网络性能的重要因素（深度，宽度），并帮助其成功地诊断了一个卷积神经网络失败的训练过程。

第 4 章　模型诊断：深度生成模型训练过程诊断的可视分析

本章研究深度生成模型（deep genrative model）训练过程诊断的可视分析方法，帮助专家交互地探索模型性能不佳或训练失败的原因。作为一种新兴的用于无监督以及半监督学习的深度学习模型，深度生成模型的主要目的是，在没有带类标训练样本[67] 或只有很少带类标训练样本[68] 的情况下，揭示数据中的主要特征，从而完成生成类似数据等任务。由此可见，深度生成模型能够克服用于有监督学习的深度学习模型（例如，卷积神经网络）需要大量类标这一缺点。因此，深度生成模型有着广泛的应用，例如数据聚类，图片去噪，3D 场景重建，场景理解，密度估计，数据压缩，特征学习以及半监督分类[17,69]。

然而，由于深度生成模型的结构相比于其他深度学习模型（例如，卷积神经网络）更加复杂，训练深度生成模型往往需要更多的技巧。首先，卷积神经网络中主要含有确定性的操作（例如，卷积和池化）。而深度生成模型中既含有确定性操作，又包含大量的随机变量，这加大了训练的难度。其次，卷积神经网络是一个自底向上的判决过程。具体地说，卷积神经网络在底层接收一个样本（例如，图片），并逐渐计算出更高层次的特征，最后输出一个判决结果（例如，图片的类别）。而深度生成模型既包含一个自顶向下的生成过程，又包含一个自底向上的贝叶斯推理（Bayesian inference）过程。生成过程通过顶层的特征生成相应的底层输入。而贝叶斯推理过程用于在给定输入（例如，图片）的情况下揭示其顶层的特征。虽然大规模贝叶斯推理已经有了一定进展，但是在实际应用中还需要解决很多问题[70]。因此，诊断深度生成模型的训练过程，对于机器学习专家来说具有重要的的理论和实践意义。

4.1　背景介绍：深度生成模型

这一节简要介绍深度生成模型[7] 的基本概念和主要结构。这里用一个生成图片的深度生成模型作为例子阐述深度生成模型的工作原理。

假设有一些图片 X，目标是根据这些图片，产生相似的图片。在数学上，这个问题可以建模成找到用于采样 X 的真实分布 $P_t(X)$。找到准确的 $P_t(X)$ 是不可行的，因为我们只知道 $P_t(X)$ 中采出来的有限样本集 X。而深度生成模型的主要目标就是找到一个近似分布 $P_g(X)$，以期最好地逼近 $P_t(X)$。为此，深度生成模型利用一个简单分布（例如高斯分布），将这个分布经过一个深度神经网络 $f(\cdot)$，变换成所需的 $P_g(X)$。这背后的数学原理是，任何一个 d 维分布都可以由另一个 d 维简单

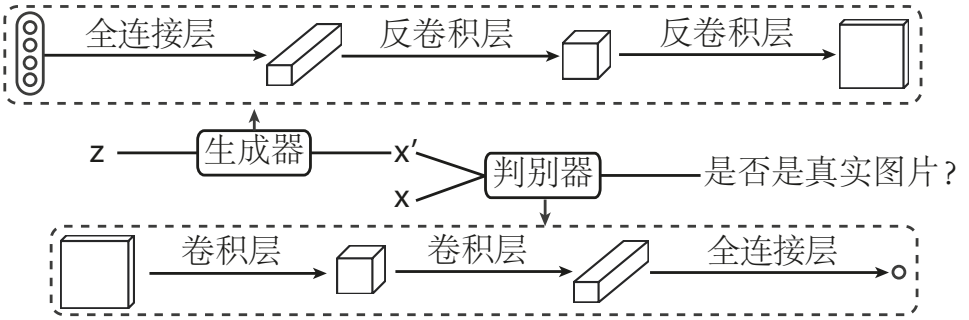


图 4.1　生成式对抗网络的基本结构示意图

分布，通过一个足够复杂的函数变换得到[71]。深度生成模型正是用一个深度神经网络作为这个足够复杂的函数，实现这一变换。上述过程可以看做一个解码过程。具体地说，生成的图片 $\mathbf{x}'$ 可以看做由高斯分布的样本 $\mathbf{z}$ 通过深度神经网络 $f(\mathbf{z};w)$ 解码得到。这里高斯分布的样本 $\mathbf{z}$ 可以看做是生成图片的特征。接下来介绍两种最经典的深度生成模型：变分自动编码器（variational autoencoders，VAE）[72]，以及生成式对抗网络（generative adversarial net，GAN）[73]。这两种深度生成模型在无监督以及半监督学习中都有着广泛应用。

变分自动编码器。变分自动编码器的结构与自动编码器（autoencoder，AE）的结构相近。自动编码器是一种传统的无监督学习模型，用于根据输入，以最小信息损失的方式重建输入[7]。自动编码器由两部分组成：编码器（encoder）和解码器（decoder）。编码器将输入 $\mathbf{x}$ 变换为其特征 $\mathbf{z}_a$。而解码器将 $\mathbf{z}_a$ 变换为重建后的输入 $\mathbf{x}'$。变分自动编码器可以看做是概率化的自动编码器。具体地说，在自动编码器中，特征 $\mathbf{z}_a$ 是实数向量，而在变分自动编码器中，特征 $\mathbf{z}_v$ 是一个由随机变量（例如，服从高斯分布的随机变量）组成的向量。相对应的，变分自动编码器也由两部分组成：一个概率化的编码器，用于估计真实的后验概率 $P(\mathbf{z}_v|\mathbf{x})$，和一个生成式的解码器，用于从特征 $\mathbf{z}_v$ 重建输入 $\mathbf{x}'$。

生成式对抗网络。如图4.1所示，生成式对抗网络由两部分组成：生成器（generator）和判别器（discriminator）。生成器用于从特征 $\mathbf{z}$ 生成图片 $\mathbf{x}'$。而判别器尝试将生成的图片与真实图片相区分。判别器常常是一个卷积神经网络[74-75]，而生成器往往由一系列全连接层组成。生成式对抗网络的训练过程可以看做一个双人博弈（two-player game）。在博弈中，生成器需要生成判别器无法区分的图片，而判别器需要尽可能区别生成的图片和真实图片。这种博弈使得生成器和判别器共同进步，直到判别器无法区分生成的图片与真实图片。相比于变分自动编码器，生成式对抗网络的训练由于其不稳定性更加困难[75]。

4.2　问题分析与建模

现有展示深度生成模型训练过程的工具[23]支持专家以折线图的形式浏览整体训练情况，诸如网络损失函数或是每个网络中间层响应平均值随时间的变化。类似工具能够给专家提供一个训练过程的概览，但是不足以帮助专家定位真正导致训练失败的原因，例如，网络中一组行为异常的神经元。为了解决这个问题，需要展示训练动态数据，例如网络中神经元连边权值随时间的变化。而现有的工具在利用折线图展示大量细节的训练动态数据时，会产生严重的视觉混乱。因此，诊断深度生成模型训练过程的关键在于建立起沟通整体统计信息与训练动态数据之间的桥梁。为了解决这个问题，本文提出了一个基于责任分配的多层次可视分析方法，支持专家从整体统计信息出发，逐层次浏览，最终通过浏览训练动态数据找到真正导致训练失败的一个或几个神经元。

为了确定从哪些层次诊断一个训练过程，我们首先观察了机器学习专家的典型诊断过程并与相关专家进行讨论。这里我们选择了三组不同背景的机器学习专家，使得总结出的典型诊断过程更具有普遍性。第一组机器学习专家的主要研究方向是，利用深度生成模型，卷积神经网络等深度学习方法处理有监督，无监督，半监督，增强学习问题。第二组机器学习专家的主要研究方向是，利用深度生成模型，残差网络[1]，区域卷积神经网络[47]（region convolutional neural network）等深度神经网络解决计算机视觉问题。第三组专家主要利用区域卷积神经网络等深度模型研究行人检测以及图片分割等经典计算机视觉问题。

根据我们的观察和与专家的讨论，我们发现专家在调试过程的开始，往往会浏览损失函数在训练过程中的变化情况，以便找到不正常的时间片。在找到感兴趣的时间片之后，专家通常浏览在该时间片上网络中每个中间层的一些统计信息以找到感兴趣的中间层（时间片层次分析）。在定位可能导致训练失败的中间层后，为了找到出现问题的神经元，专家通常打印出感兴趣的网络中间层中的部分训练动态数据，例如该层神经元响应在训练过程中的变化（网络层次分析）。在此之后，专家通常会用领域知识分析网络训练失败的根本原因。这个步骤极大地依赖于专家的专业知识（神经元层次分析）。其原因是训练失败可能由多种错误引起，找到失败的根源是很难的。例如，在深度生成模型的训练过程中，损失函数值偶尔会变为 NaN（not a number）或是 Inf（infinity）。这会直接导致训练失败。可能的原因包括代码中的错误，数值上的不稳定性，或者是网络结构不合适。即使知道是哪种原因导致的这个错误，还是很难定位到具体导致这个错误的一组神经元上，因为神经元之间是相互影响的。综上所述，专家的调试过程是一个三层的分析过程：时间片层次-网络层次-神经元层次。

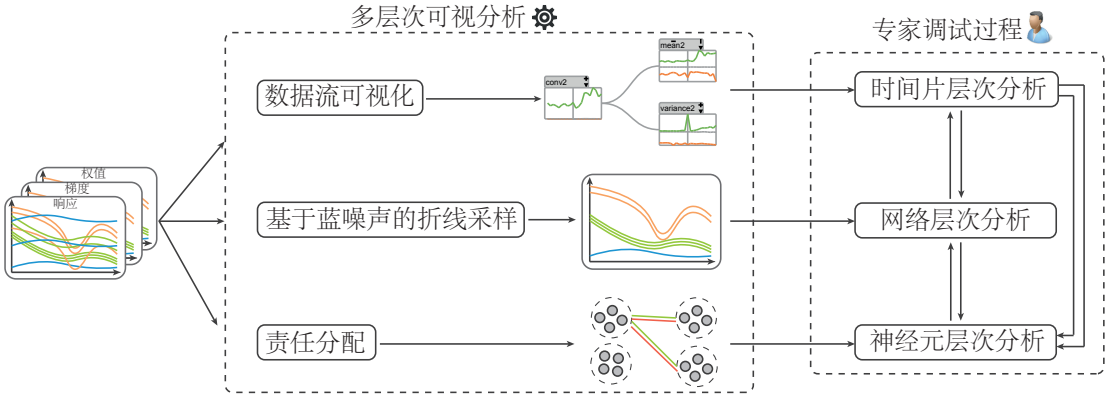


图 4.2　深度生成模型训练过程诊断的可视分析方法概览

为了支持上述多层次分析流程，本文提出了相应的多层次可视分析方法（图4.2）。该算法包含三个层次：时间片层次，网络层次，以及神经元层次。作为分析过程的开始，该算法支持专家以不同的时间粒度浏览损失函数的变化[76]。在专家选择感兴趣的时间片之后，时间片层次可视化方法结合有向无环图和折线图，有效展现数据在网络中的流动。通过这个数据流可视化，专家能够方便地找到要进一步浏览的中间层。在此基础上，网络层次可视化方法利用基于蓝噪声的折线采样算法[8]，挑选出该中间层中具有代表性的训练动态数据，例如该层响应/权值/梯度随时间的变化。浏览采样结果能够帮助专家定位到可能导致训练失败的神经元。在此之后，所开发的基于责任分配的神经元层次可视化方法，会展现出神经元之间的相互影响，帮助专家分析网络训练失败的根本原因。

接下来，具体介绍所提出的多层次可视分析方法的各个组成部分。

4.3　时间片层次：结合有向无环图与折线图展现数据流

在时间片层次，着重展示数据在网络中的流动情况。作为一种深度学习模型，深度生成模型与传统浅层机器学习模型的重要区别之一是该模型由很多网络中间层组成。这些中间层各自起到不同的作用，共同完成相应的任务，例如生成一个图片。理解数据在网络各个中间层中的流动情况，对于理解网络中间层的不同作用是非常重要的[13]。另外，一个失败的训练过程往往是由一个中间层导致的。专家们表示在深度学习库（例如 TensorFlow）提供的中间层一般都很鲁棒，容易出现错误的层是自己构建的那些。因此，检查在这些中间层，尤其是专家自己构建的中间层中的数据流动情况，对于定位导致训练失败的中间层是很有帮助的。但是一个深度学习模型中可能含有上百层，每层中含有上百万神经元。直接展示所有的中间层，和所有中间层上的数据流动情况会导致严重的视觉混杂。因此，需要一个有效的可视化工具帮助理解数据在网络中的整体流动情况[13]。

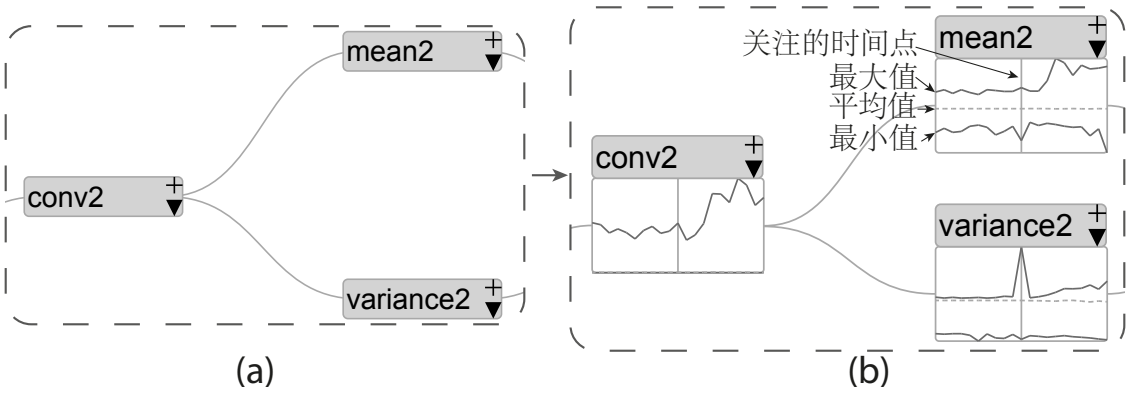


图 4.3　时间片层次可视化，以混合可视化形式展现的数据在网络中的流动情况：(a) 利用有向无环图可视化展示深度生成模型的网络结构；(b) 利用折线图展现数据流

为了解决这个问题，Rauber 等[13] 利用 t-SNE 降维技术展示数据在网络中的流动情况。具体地说，他们利用一条轨迹展示每一个输入数据在网络中的流动情况。因此，这种方法只能处理含有单链结构的深度神经网络。然而深度生成模型可能有更复杂的网络结构，例如在变分自动编码器中网络中间层可能分裂和合并。本文设计了一个混合可视化以处理类似复杂的网络结构。具体地说，本文用有向无环图可视化来展现网络中间层之间的连接关系，并用一系列的折线图与之相结合来展现数据在每一层中的流动情况。

有向无环图可视化。本文将一个深度生成模型的结构表示为一个有向无环图。图中每一个节点是一个网络中间层，中间层之间的连边用有向无环图中的边表示（图4.3(a)）。在此基础上，利用 TextFlow 系统[77] 中的布局算法计算图中每一个节点的位置。为了处理含有数十乃至上百层的深度生成模型，利用 TensorFlow[23] 中的方法用一棵树组织这些中间层。对于大的深度生成模型，只默认展示这棵树的上层节点，专家可以展开这个节点来浏览单个的网络中间层。

利用折线图展现数据流。本文用一系列折线图展现数据流。具体地说，在每一个网络中间层中放置了一个直线图，来展示数据在当前中间层的流动情况（图4.3(b)）。在这个折线图中，中间的竖线代表选定的时间点，每一条折线代表在选定时间点 S_t 附近训练动态数据的变化情况，例如在当前层中平均响应从 S_{t-k} 到 S_{t+k} 的变化。默认设定 $k = 10$，专家可以交互地修改这个默认值。

图4.4展示了若干在开发过程中发现的有关响应的典型数据流模式。图4.4(a) 表示在关注的时间点上，这个中间层中的响应有了突变。图4.4(b) 表示大部分的响应在关注的时间点上保持稳定，而少部分的响应突然增大。这两个现象表明这个中间层可能出现了问题。一个中间层出现了问题，也可能导致其他中间层有相应的变化。如图4.4(c) 所示，这层的响应在关注的时间点之前都很稳定，在关注的时间

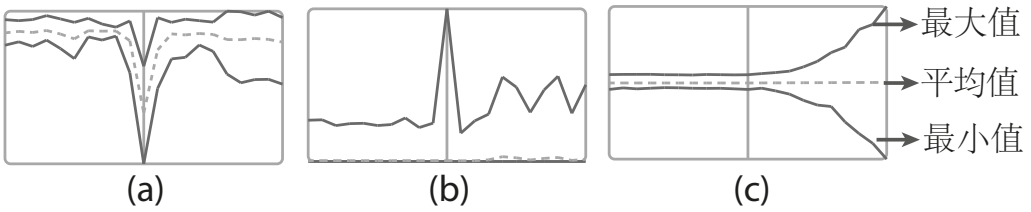


图 4.4　单个中间层中数据流的典型模型：(a) 大部分响应突然改变；(b) 少部分响应突然变化；(c) 在关注的时间点后响应逐渐变得不稳定

点之后逐渐变得不稳定。这个现象意味着别的中间层很可能出现了问题。

4.4　网络层次：基于蓝噪声采样的训练动态数据可视化

当专家选定一个网络中间层具体浏览时，希望展现出相应的训练动态数据帮助他们找到感兴趣的神经元进一步分析。找到感兴趣的神经元的关键在于保留训练动态数据中的异常值。异常值（outlier）检测旨在找到与预想的行为不同的数据[78]。合作的专家表示，一般来说，异常的神经元是最有可能导致训练失败的神经元。而根据他的经验，最难调试的错误正是由一个或几个神经元引起而传播到整个网络上，因为他很难从海量的神经元中准确地找出他们。这个时候只能依靠现有工具中的调试模式或者数值检查模式（例如 Theano[79] 和 TensorFlow[23]）找到可能出错的神经元。该过程费时费力。因此，希望能有一个自动检测异常值的算法。但是自动地检测异常值在机器学习领域依然是极具挑战性的[80]，所以检测异常值还需要有专家的参与。

为了解决这个问题，最直接的办法是用折线图来展示训练动态数据（也就是一些时间序列数据），因为折线图对机器学习专家来说是一种熟悉的可视化形式，能够帮助他们更专注于分析任务。但是直接用折线图展现大量的时间序列数据会产生严重的视觉混乱。因此，采用基于蓝噪声的折线采样算法[8]，选取具有蓝噪声特性的折线，以保留可能的异常折线并减少视觉混乱。

4.4.1　蓝噪声采样

受蓝噪声采样在计算机图形学领域的广泛应用（例如图像重建以及颜色点彩化[81]）的启发，选用蓝噪声采样选取具有代表性的折线。蓝噪声采样是指采出的样本具有蓝噪声特性，即采出的样本在空间中是均匀且随机分布的[82]。这种均匀性在可视化的应用中是非常重要的[83]。具体地说，均匀性可以使高密度的区域中采样率低，而低密度区域中采样率高。这种性质导致蓝噪声采样能够减少视觉混乱以及保留可能的异常值。因此，我们决定采用基于蓝噪声的折线采样选取具有

代表性的时间序[8]。

4.4.2　基于蓝噪声的折线采样算法

采样算法的本质是从数据集中选择一部分样本作为样本集。传统的基于蓝噪声的点采样中，核心的步骤是从数据集中选择一个数据点，并根据这个数据点与样本集中样本的距离，决定是否将这个数据点放入样本集中。这也是基于蓝噪声的折线采样算法的基本思想。与传统的基于蓝噪声的点采样算法不同的是，每一次选择数据集中的一个折线（时间序列）而非一个点，并计算这条折线与样本集中其他折线的距离。由于一条折线是由一组首尾相连的线段组成，折线 L_1 和 L_2 间的距离 d 定义为：

$$d(L_1, L_2) = \frac{1}{N_S} \sum_{i=1}^{N_S} d_C(s_1^i, s_2^i), \tag{4-1}$$

其中，N_S 是折线中的线段数（即对应时间序列的时间点数）。$d_C(\cdot,\cdot)$ 是两个线段中点的距离。s_1^i 和 s_2^i 是对应折线中相同时间点处的线段。这个距离计算方式能够有效地保留异常值（图4.5(c)）。如果两个线段之间距离很大，那么相应的两条折线的距离主要由这两个线段的距离决定，也会较大。

采样结果实例. 图4.5比较了不采样（图4.5(a)），随机采样（图4.5(b)）以及蓝噪声采样（图4.5(c)）的可视化效果。这里选择第二个案例分析中使用的变分自动编码器中的第一个中间层的权值变化作为待采样数据。我们从 1,728 个时间序列中采样出 5% 的时间序列进行展示。这里选择了两种采样方法：随机采样和基于蓝噪声的折线采样。

如图4.5(a) 所示，不进行采样会导致红色矩形框内的高密度区域产生严重的视觉混乱。与随机采样的结果（图4.5(b)）相比，蓝噪声采样方法能够有效地减少大量时间序列堆积导致的视觉混乱现象（图4.5(b) 和 (c)）。另外，从图4.5(a) 能够发现，有少部分时间序列从训练开始就呈现逐渐下降的趋势（图4.5(a) 中绿色方框部分）。这些时间序列可能是异常值，需要专家进一步检查。通过比较图4.5(b) 和 (c)，能够发现蓝噪声采样方法可以更好地保留这部分可能的异常值。

4.4.3　交互

为了帮助专家更方便地浏览一个网络中间层中的训练动态数据，我们将以下交互融入网络层次的可视化中：

维度聚合。 中间层的响应/梯度/权值可以用一个张量（tensor）表示。例如，在卷积层中一张图片的响应可以表示为一个三维张量 $\mathbf{T} \in \mathbb{R}^{H\times W\times C}$，其中 H，W 和 C 分

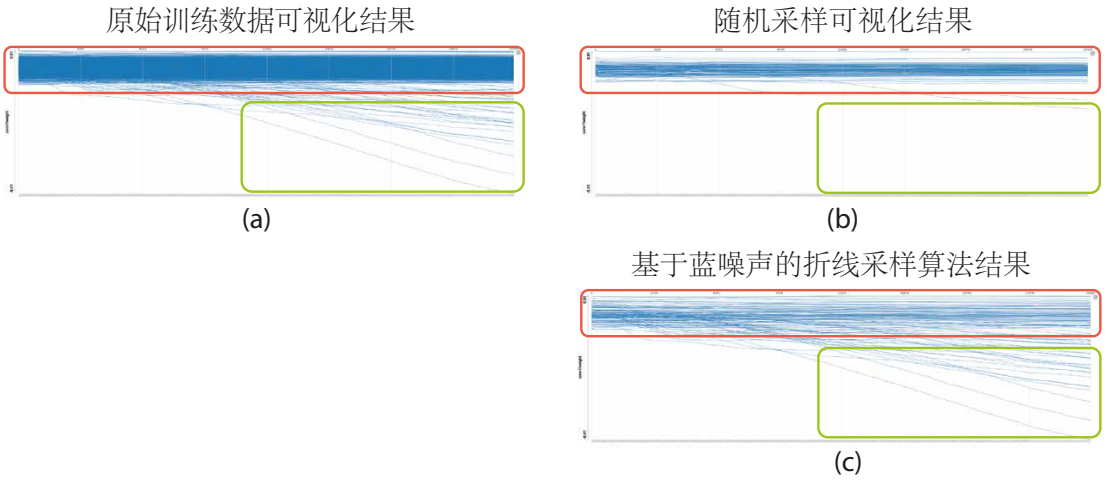


图 4.5　不同采样方法结果比较：(a) 没有采样的原始训练动态数据可视化；(b) 随机采样的可视化结果；(c) 基于蓝噪声的折线采样结果，蓝噪声采样能够更好地保留数据中的异常值（绿色矩形），同时有效减少视觉混杂（红色矩形）

别为高度，宽度和图片通道数。将一个响应/梯度/权值中的几个维度聚合起来可以有效地减少需要展示的时间序列数。因此我们支持专家交互式地聚合响应/梯度/权值的某些维度，并在聚合的基础上进行采样。

焦点 + 上下文时间轴。 由于在训练过程中有数万个时间点，展示所有时间点上会产生严重的视觉混杂。为了解决这个问题，采用了焦点 + 上下文时间轴技术[76]，帮助专家从不同的时间粒度上浏览中间层具体的训练动态数据。该技术首先以粗粒度展示具体的训练动态数据。专家可以找到感兴趣的部分并以更细粒度进行浏览这部分训练动态数据。

4.5　神经元层次：神经元相互影响可视化

在神经元层次，本文计算并展示神经元之间的相互影响。这能够帮助专家分析网络训练失败的根本原因（神经元层次分析）。

现在研究者们还无法有效理解深度生成模型中神经元的相互影响。即使专家找到了导致训练失败的神经元，也很难找到训练失败的根本原因。合作的专家表示，有的时候即使他发现一个神经元的响应异常，也很难分析出到底是什么引发这个问题。有的时候这样的问题的根源是别的层，然后传播到这一层的这个神经元上。因此专家希望能直观地浏览神经元之间的相互影响，尤其是对于其他神经元对一个神经元响应（输出）的影响。

为了解决这个问题，本文借用了机器学习领域的责任分配这一概念（credit assignment）[84]，计算前后层神经元对当前层神经元的影响。责任分配能够决定当

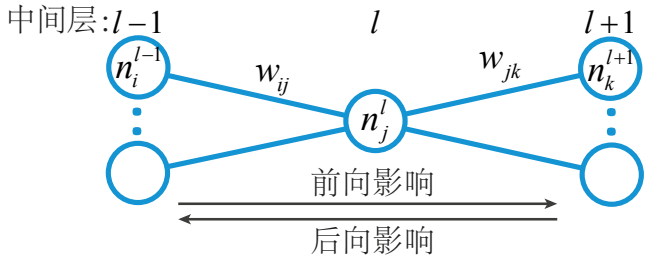


图 4.6　神经元前向影响与后向影响示意图

网络的输出与所预想的不同时，哪些神经元应该被修改。在此基础上，将计算出的影响以直观的形式展现出来，辅助专家分析。

4.5.1　责任分配计算

如图4.6所示，l 层神经元 n_j^l 的输出不仅仅受前层神经元的影响（前向影响），还会受到后一层神经元的影响（后向影响）。

这两种影响共同决定了一个神经元的输出。在这两种影响中，前向影响已经被机器学习领域的研究者们深入地研究过了[6,85]。本文采用当下最先进的层级相关性传播算法（Layer-wise Relevance Propagation，LRP）[85] 计算神经元之间的前向影响。另外，本文采用后向传播算法[86] 计算后向影响。接下来具体介绍如何计算前向和后向影响。

前向影响。 如图4.6所示，l 层神经元 n_j^l 的输出是由 $l-1$ 层神经元决定的：$a_j^l = \sigma(\sum_i w_{ij} a_i^{l-1})$，其中 $\sigma(\cdot)$ 是响应函数，w_{ij} 是连接 n_j^l 和 n_i^{l-1} 的权值。在 LRP 中，神经元 n_i^{l-1} 对 n_j^l 的贡献 $C(a_i^{l-1} \to a_j^l)$ 可以计算为：

$$C(a_i^{l-1} \to a_j^l) = w_{ij} a_i^{l-1} / Z, \tag{4-2}$$

其中 $Z = \sum_h w_{hj} a_h^{l-1}$ 是一个归一化项。

后向影响。 根据后向传播（back-propagation）算法[86]，$l+1$ 层神经元 n_k^{l+1} 的输出 a_k^{l+1} 对权值 w_{ij} 的梯度 g_{ij} 有一个后向的影响。在权值 w_{ij} 根据梯度更新之后，这个权值会影响神经元 n_j^l 的输出 a_i^l。上述分析过程可以概括为：

$$a_k^{l+1} \Rightarrow g_{ij} \Rightarrow w_{ij} \Rightarrow a_i^l, \tag{4-3}$$

其中 $A \Rightarrow B$ 表示 A 能影响 B。因此，$l+1$ 的神经元的输出也会影响 l 层神经元的输出。为了计算这个后向影响，将式4-3中每一步的影响综合考虑：

$$C(a_k^{l+1} \to a_j^l) = w_{kj} a_k^{l+1} / Y, \tag{4-4}$$

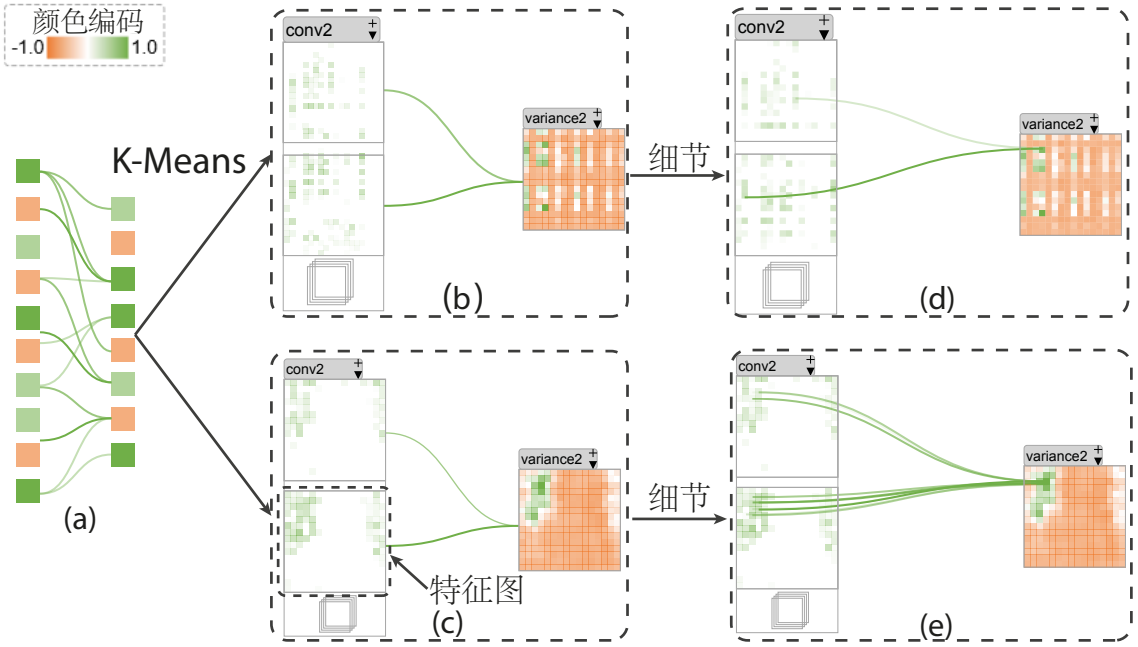


图 4.7　展示神经元之间相互影响的责任可视化示意图：(a) 神经元聚类之前；(b)K-Means 算法聚类的结果；(c) 以特征图组织神经元的可视化结果；(d) 和 (e) 展现具体地影响

其中 $Y = \sum_h w_{jh} a_h^{l+1}$ 是归一化项。可以看出两种影响的计算方式相似，具有一定对称性。

4.5.2　责任可视化

基于计算出的前向和后向影响，本文展现了在专家感兴趣的时间片上神经元之间的相互影响，方便专家确定导致网络训练失败的神经元。这里以前向影响为例阐述可视化设计。如图4.7(a) 所示，每一个神经元用一个矩形代表。这个矩形的颜色表示这个神经元的响应（红色：负的响应，绿色：正的响应）。神经元之间的相互影响用矩形之间的边来表示（图4.7(a)）。在边上，采用相同的颜色表示影响的大小以及极性（绿色：正面影响，红色：负面影响）。由于一个中间层中神经元数量过多，直接展示所有的神经元及影响会导致严重的视觉混乱。为了解决这个问题，首先用 K-means 算法[86] 对神经元进行聚类，并只展示对所选神经元有重要影响的神经元聚类（图4.7(b)）。为了节省空间，以网格的形式展现神经元聚类。另外，默认情况下，只展示两个神经元聚类之间的平均影响。专家可以用鼠标点击一个神经元来具体观察这个神经元与别的神经元的相互影响（图4.7(d)）。为了给专家提供上下文，我们将责任可视化与时间片层次的可视化以焦点 + 上下文的形式相结合。

值得注意的是，经过与机器学习专家的讨论，对于卷积层以及解卷积层（de-

convolutional layer），用特征图[17] 来组织神经元（图4.7(c)），而不使用 K-Means 聚类方法。这种方法有两个好处：首先，这种组织形式对于机器学习专家更为熟悉[6]；其次，这种组织形式能够帮助专家更好地检测出输入图片的哪一部分可能导致网络训练失败。如图4.7(e) 所示，选定的神经元主要受到前一层第二张特征图左上角神经元的影响。经过逐层回溯，能够进一步定位到输入图片的哪一部分影响了这个神经元。而 K-means 算法产生的聚类结果（图4.7(d)）不能有效的揭示这个现象。

4.6　算法应用：DGMTracker 系统

为了验证所提出的多层次可视分析方法的有效性，我们基于该方法开发了 DGMTracker 系统，交互地探索深度生成模型性能不佳或训练失败的原因。我们与两位研深度学习专家合作完成了两个案例分析。专家 E_1 的研究方向是将深度生成模型应用于深度增强学习中。专家 E_2 的研究方向是深度生成模型训练过程中变分推理（variational inference）。在这两个案例分析中，我们帮助专家理解生成式对抗网络的训练过程，以及诊断一个失败的变分自动编码器训练过程。

4.6.1　系统概览

DGMTracker 系统包含三个模块：

- 时间片层次可视化模块：展示深度生成模型中数据流动的情况；
- 网络层次可视化模块：从大量的时间序列中选择具有代表性的时间序列；
- 神经元层次可视化模块：展示神经元之间的相互影响

作为诊断过程的开始，专家可以以不同的时间粒度浏览损失函数的变化（图4.8(a)）。转接也可以点击损失函数上的某一个时间点选择自己感兴趣的时间点进一步分析。专家可以浏览在该时间点周围数据在网络中的流动情况，以找到需要进一步分析的中间层（图4.8(b)）。在找到感兴趣的中间层之后，专家可以利用网络层次可视化模块，浏览该中间层中的训练动态数据，例如，相应随时间的变化（图4.8(c)）。DGMTracker 用折线图展示训练动态数据。折线图中每一个折线都是一个神经元或神经元组。根据训练动态数据的折线图，专家能够找到可能导致网络失败的神经元 (图4.8A)。在此基础上，他可以利用神经元层次可视化模块查看神经元之间的相互影响（图4.8B），从而分析出网络训练失败的根本原因。

4.6.2　案例分析

在第一个案例分析中，我们与专家 E_1 合作，更好地理解生成式对抗网络的训练过程。第二个案例分析展示了我们如何帮助专家 E_2 诊断一个变分自动编码器的

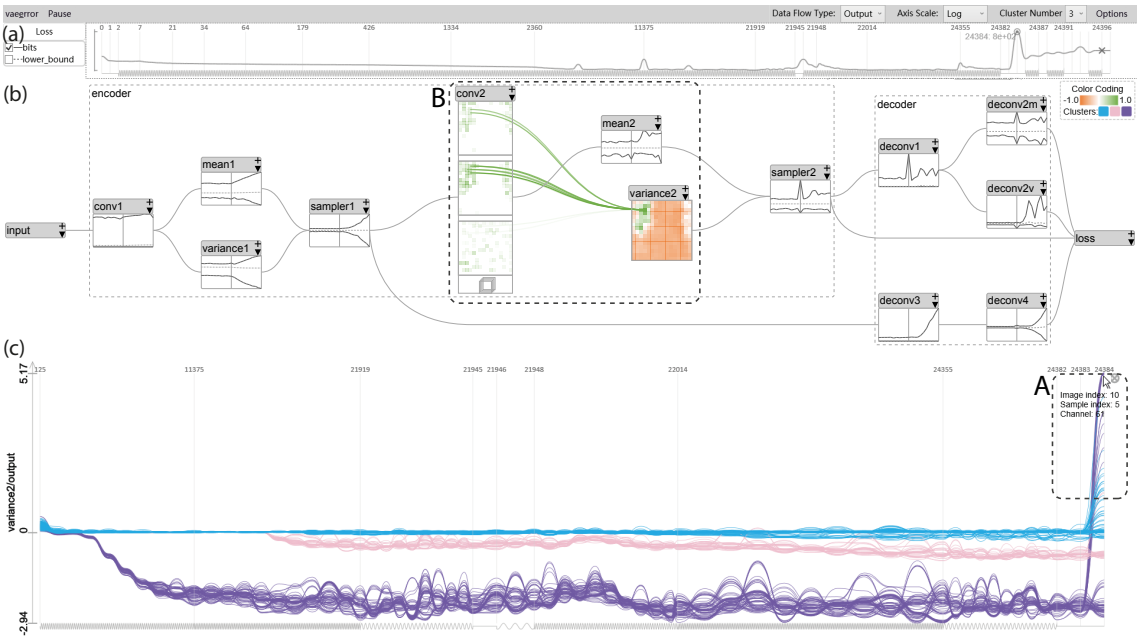


图 4.8　DGMTracker 系统概览：(a) 损失函数的变化；(b) 数据流可视化；(c) 训练动态数据可视化

失败训练过程。

4.6.2.1　理解生成式对抗网络的训练过程

生成式对抗网络是深度生成模型的研究热点之一[73-74]。影响生成式对抗网络在实际中应用的最大问题之一是其训练过程非常不稳定。为了解决这个问题，Arjovsky 等[74] 基于生成式对抗网络开发了 Wasserstein 生成式对抗网络（Wasserstein Generative Advesarial Net，以下简称 WGAN）。在 Wasserstein 生成式对抗网络中，作者用 Wasserstein 距离替代了原先的距离计算方式，因为他们发现原先的距离计算方式会导致梯度消失的问题。采用 Wasserstein 距离的优势在于，它是一个连续几乎处处可微的距离计算方式，因此它能够使梯度的计算更加稳定。在 E_1 的调查研究中，他对于 Wasserstein 的相关文献中介绍但没有完全解释的两个现象很感兴趣[73-74]，想尝试用 DGMTracker 帮助他理解这两个现象。

不合适的损失函数。Goodfellow 等[73] 根据生成式对抗网络工作原理给出了一个基于双人极大极小博弈[87] 的损失函数定义：

$$\min_G \max_D \mathbb{E}_{x\sim p_{data}(x)} \log D(x) + \mathbb{E}_{z\sim p(z)} \log(1 - D(G(z))), \tag{4-5}$$

其中，G 表示生成器，D 表示判别器。

Goodfellow 等[73] 指出虽然上述损失函数在实际应用中是不合适的，会导致训

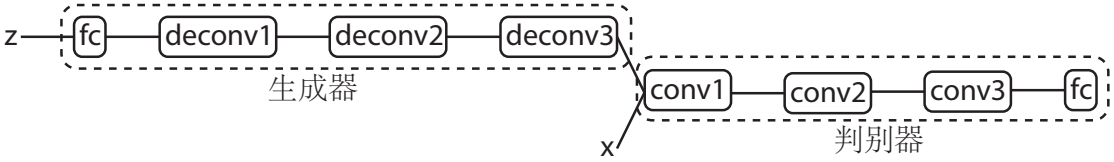


图 4.9　案例分析中使用的生成式对抗网络结构示意图

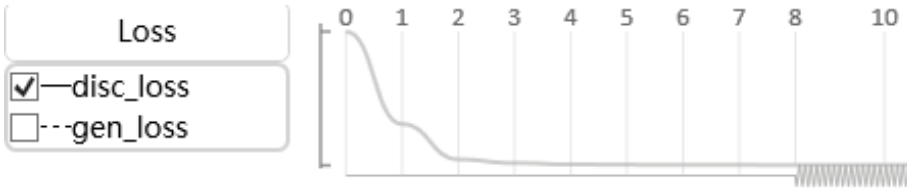


图 4.10　训练无法进行：判别器的损失函数在训练开始后的几个时间点内迅速停止变化

练失败。但是 E_1 并不理解其中原因。

因此，他构建了一个生成式对抗网络，其结构如图4.9所示。这个模型含有 5.48 兆个参数。该模型在 CIFAR10 数据集[63] 上进行训练。从损失函数的曲线中，E_1 发现判别器上的损失函数值在训练开始几个时间点后就很快停止了（图4.10）。这意味着训练迅速卡在这个地方。

为了理解为什么卡住，E_1 选择检查在训练卡主之后的时间片层次的数据流动情况。具体地说，专家点击了第八个时间点上损失函数，并选择浏览梯度的数据流。他发现在训练的一开始梯度是正常的，但是在几个时间点之后，梯度都变为 0（图4.11）。为了检查从哪个中间层开始，梯度消失的，专家从网络的输出端逐层检查梯度的数据流，发现甚至在最后一个全连接层，梯度都为 0 (图4.11A)。根据后向传播算法，梯度是从最后一个中间层逐层向输入层传播。因此，他认为是这个全连接层导致训练卡住。这使得 E_1 检查了这一层的输出。他发现了一个异常现象，在训练开始几个时间点之后，生成出的图片在判别器上的输出都很接近 0（图4.12A），而真实图片在判决器上的输出都接近 1（图4.12B）。

进一步检查了产生出的图片，E_1 明白了这个现象出现的原因。在训练的开始，生成器由于训练不充分，无法生成真实的图片。在这种情况下，判别器能够很容易地区分真实的图片和生成的图片。又因为判别器的输出表示一张图片是真实图片的概率，在训练的开始这些生成出的图片是真实图片的概率接近 0。

在理解了这个现象出现的原因之后，专家进一步分析了这个现象对训练的影响。E_1 检查了出现这个异常现象的全连接层上的导数，发现损失函数对于这个中间层输出的导数，当这个现象发生时，接近 0。根据后向传播算法，这会使得其他所有层上的权值的梯度接近 0，也就导致这些权值无法更新，进而导致训练无法正

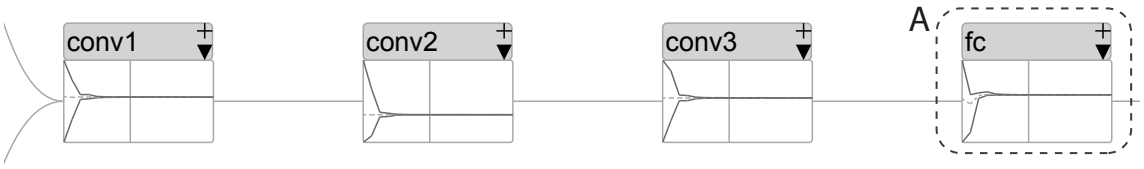


图 4.11　使用不合适损失函数导致的梯度消失现象示意图

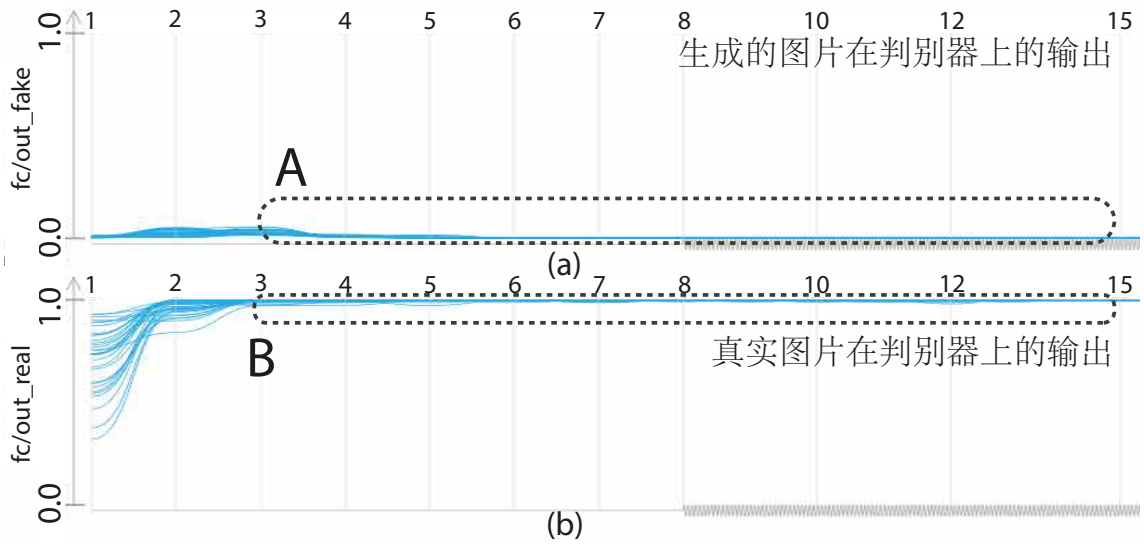


图 4.12　判别器输出随时间变化的示意图：(a) 由生成器生成的图片在判别器上的输出在训练开始后很快变为 0；(b) 真实图片的输出很快变为 1

常进行。

基于动量的训练方法的不稳定性。在深度学习模型的训练中，基于动量的方法是很常用的[88]。但是，在训练 WGAN 的时候，采用基于动量的训练方法会导致训练不稳定[74]。虽然动量被认为是导致训练不稳定的可能原因，但是为什么会导致训练不稳定在文章中没有完全解释。为了分析为什么基于动量的训练方法会导致 WGAN 训练不稳定，E_1 构建了一个 WGAN。该模型结构和上个案例分析中使用的 GAN 结构相同。相同地，E_1 在 CIFAR10 数据集上用一种基于动量的训练方法 Adam[89] 训练了这个网络。

E_1 首先检查了判别器和生成器上的损失函数。他注意到在判别器上的损失函数在两个时间点上突然增大了（图4.13A 和 B）。为了确定突然增大的损失函数的影响，E_1 首先研究了 Adam 训练方法中权值的更新方式。在传统的训练方法随机梯度下降（stochastic gradient descent，SGD）中，权值 w_i 的更新量直接由梯度 g_i 与学习率 α ($\alpha > 0$) 决定：

$$w_i^{t+1} = w_i^t - \alpha g_i^t. \tag{4-6}$$

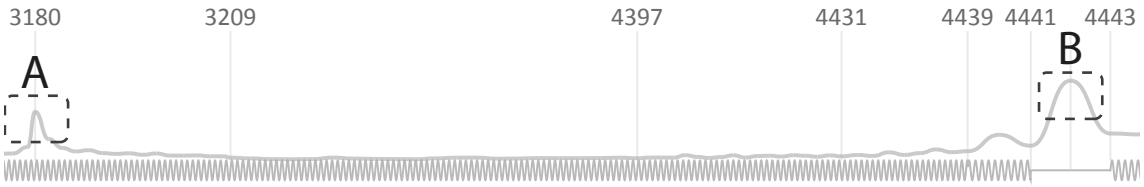


图 4.13　在 WGAN 训练过程中采用基于动量的训练方法-损失函数随时间的变化

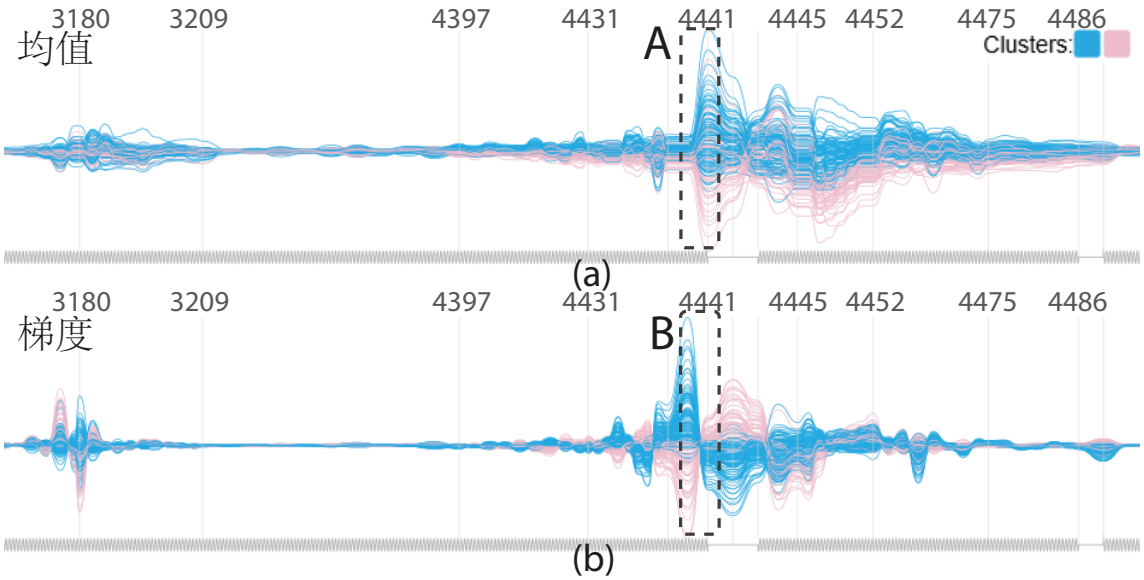


图 4.14　训练过程中梯度的符号突然发生改变后其余估计量变化的示意图：(a)Adam 训练方法对其均值估计的改变；(b) 梯度的改变

而 Adam 利用动量的概念，首先自适应地估计梯度的均值和方差，并用估计出的均值和方差更新权值。因此，专家检查了网络最底层卷积层中，梯度的估计均值和方差的变化。检查这层的原因是这层最容易受到梯度消失的影响[7]。

专家注意到在时间点 4441，梯度的正负突然发生了改变（图4.14B），但是 Adam 对梯度均值的估计的正负没有变化（图4.14A）。这个是解释基于动量的训练方法的不稳定性的关键。E_1 进一步解释道，当梯度本身的正负发生改变之后，均值的估计却没有相应的改变。这是因为均值是由之前若干时间点上的梯度共同决定的（图4.15）。因此，训练过程会向着不正确的方向进行训练，也就会比采用不基于动量的训练方法（例如，RMSprop）的训练过程更不稳定。

为了进一步验证这个分析结果，E_1 对每个权值计算并可视化 $(w_i^{t+1} - w_i^t)g_i^t$。验证这个数值是受式4-6的启发。在式4-6中，$(w_i^{t+1} - w_i^t)g_i^t = -\alpha(g_i^t)^2 \leq 0$。因此，当这个值为正的时候，权值的变化与梯度的变化是不一致的。如图4.16所示，在使用 Adam 的训练过程中的某些时间点上，出现了一些正值（图4.16A）。而使用非基于动量的训练方法时（RMSprop，WGAN 提出者推荐使用[74]），没有出现正值

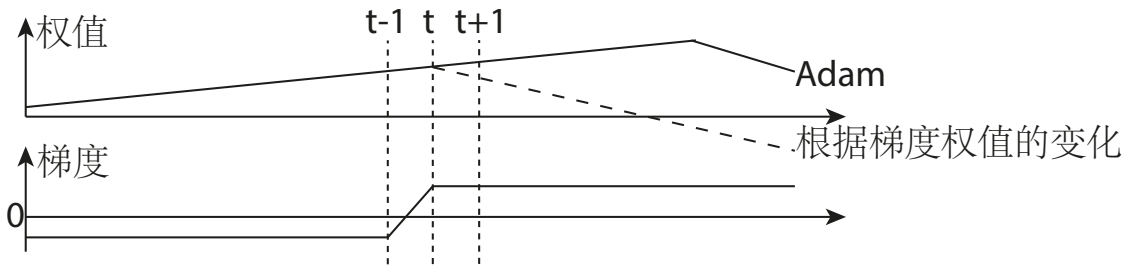


图 4.15　在使用 Adam 训练方法的训练过程中，梯度的符号改变时，对应权值的改变示意图

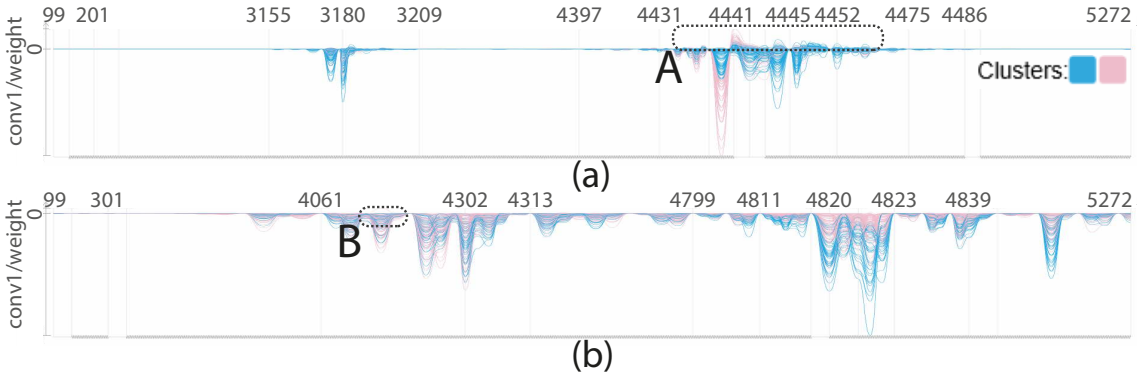


图 4.16　$(w_i^{t+1} - w_i^t)g_i^t$ 在训练过程中的改变示意图：(a) 使用基于动量的训练方法 Adam；(b) 使用非基于动量的训练方法 RMSprop

(图4.16B)。WGAN 的原始文章[74] 中也展示了这个现象，但是没能解释。

经过这些分析之后，专家表示，现在他理解了为什么基于动量的方法不能达到满意的效果。主要的原因是，梯度会在训练过程中突然改变，导致基于动量的方法失效。而这样的情况在其他类型的深度学习模型，例如卷积神经网络，的训练过程中出现地更少。因此，对于这些深度学习模型，基于动量的方法比较好用。

4.6.2.2　诊断变分自动编码器的训练过程

这个案例展示了 DGMTracker 如何帮助专家 E_2 诊断一个变分自动编码器失败的训练过程。变分自动编码器的研究是深度生成模型领域重要方向之一[90-92]，在一系列无监督学习问题中都有着广泛的应用。E_2 在其研究中设计了一个层次化的变分自动编码器，网络结构如图4.17所示。整个网路由两部分组成：一个编码器，和一个解码器。这两部分都是由一系列卷积/解卷积层，以及高斯采样层组成。E_2 在 CIFAR10 数据集上进行训练。但是，在训练过程的 10,000 到 30,000 时间点之间的某个时间点，损失函数会变为 NaN。具体的时间点取决于网络初始化使用的随机种子。

为了帮助专家找到可能的原因，我们载入了一个失败的训练过程样例。E_2 首

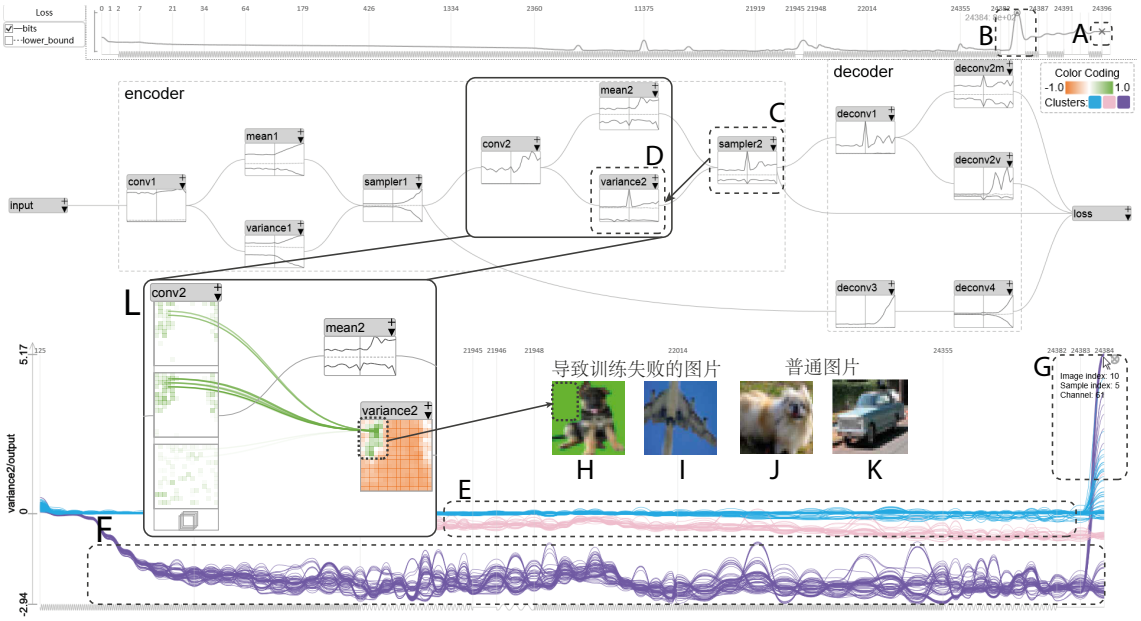


图 4.17　利用 DGMtracker 诊断一个变分自动编码器失败的训练过程

先浏览了损失函数在整个训练过程中的变化，并发现在时间点 24,397，损失函数变为 NaN（图4.17A），导致训练失败。接着，专家注意到在时间点 24,384，损失函数突然变大（图4.17B）。在这之后很短的时间内，损失函数就变为 NaN。因此，专家点击了损失函数曲线上的这个时间点，并检查在时间片层次的数据流。由于损失函数的值由中间层的响应决定，E_2 检查了每个中间层中最大/平均/最小的响应。检查了会直接影响损失函数的三个中间层之后，专家发现，在第二个高斯采样层上，出现了一个异常现象，即响应突然增大 (图4.17C)。沿着数据流回溯，专家发现在决定这个高斯采样层的对数方差的卷积层上，相应的时间点上出现了响应突然增大的情况（图4.17D）。这意味着，这个卷积层响应的变化，引起了高斯采样层的突然变化。

为了研究这个卷积层响应突然变化的原因，E_2 打算浏览一下这一层的响应在整个训练过程中的变化情况。由于这一层中含有过多的响应（约两百万），E_2 决定将响应张量中的高度和宽度两个维度聚合在一起，这样就对每张图片的每个通道产生了一个时间序列。如图4.17E 所示，大部分的响应在整个训练过程中变化不大。然而，有一些从训练开始就呈现奇怪的下降趋势（图4.17F）。这些用紫色高亮的响应引起了专家的注意。另外，他发现，在这些响应中，有少部分在时间点 24,384 处，突然增大（图4.17G）。专家通过检查这些响应发现了一个有趣的现象，所有这些响应都来自于这个训练批次（batch）中的第十张图片 (图4.17G)。E_2 进一步从数据集中找到了这张图片，并发现这张图片有一个不同寻常的纯绿色背景 (图4.17H)。由于 CIFAR10 中的图片均为 RGB 格式，纯绿色背景的像素上，绿色通

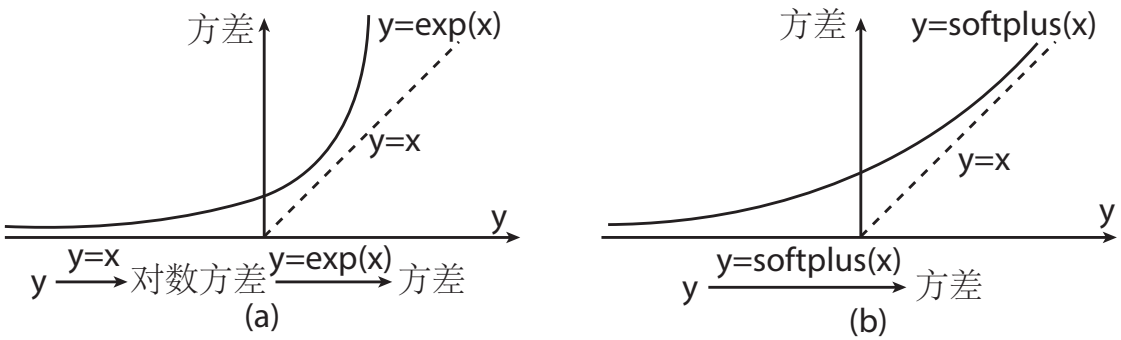


图 4.18　比较计算高斯采样层方差的两种方式：(a) 将对数方差利用指数函数变换为方差；(b) 直接计算方差

道的值会非常大（若数值取自 [0, 255]，那这些像素在绿色通道上几乎为 255）。在一张图片中集中地出现这么多极端值在 CIFAR10 这样的自然图片数据集中是不常见的（图4.17J 和 K）。E_2 怀疑是这张不寻常的图片导致的训练失败。为了验证他的猜想，E_2 选择了这张图片中响应最大处对应的神经元组（特征图），并检查了前一层神经元对这一层神经元的影响。在查看具体的影响之后，专家发现，上一层中来自绿色的背景处的神经元对这一层响应最大的神经元有非常大的影响（图4.17L）。这个进一步验证了他的猜想。

在分析了训练失败的原因是一张不寻常的图片之后，E_2 提出了一个直接的解决方案，即将这张图片换成一张普通图片，并重新训练了这个网络。但是，网络依然会以相似的方式训练失败，只是失败的时间点大大延后了（时间点 300,000 左右）。这意味着这张图片只是训练失败的原因之一。经过相同的分析，专家发现这次训练失败，来自于一张类似的图片。这张图片有一个蓝天作为背景（Fig.4.17I）。这种图片在数据集中很常见，不能全部去除。因此，E_2 决定从网络结构入手。他首先研究了为什么网络会对输入的极端值如此敏感。根据 DGMTracker 的分析结果，他重点关注产生对数方差的卷积层。他发现主要的敏感性，来源于对数方差到方差的指数变换。如图4.18(a) 所示，如果对数方差 y（即卷积层的输出）大于 0，而且发生了很小的改变，对应的方差会因为指数爆炸的原因，发生很大的改变。在这种情况下，对应的高斯采样层可能产生很大的样本。一旦产生了数值很大的样本，损失函数就会响应突然增大，导致训练很快失败。因此，解决这个问题的关键在于替换会爆炸的指数变换函数。相应的，E_2 提出将卷积层的输出直接作为高斯采样层的方差。为了保证方差大于 0 这个限制条件，专家采用了 softplus 响应函数将卷积层的输出变换为大于 0 的实数：$f(x) = \log(1 + e^x)$ (图4.18(b))。这个函数相比于指数变换在正半轴上更加平缓。

在修改之后，训练可以正常进行了。最终的损失函数值用比特描述为 4.9[7]。

专家对这个结果比较满意，他表示：“在变分自动编码器现在的结构中，使用对数方差是很常见的。一直以来我们也发现这样的网络会很容易出现 NaN 的情况。为了避免类似的错误，我们一般会尝试着限制梯度以及网络权重的大小。这种尝试有的时候会成功，有的时候不会。即使成功训练出来了，但是训练的时间比正常的训练要长很多，因为梯度被限制了。现在，我们知道了这个问题的来源是对数方法的使用，就可以有针对性地避免这个问题。不仅如此，这个例子还让我们在其他的网络设计中对于对数方差更加小心”。

4.7　讨论及小结

局限性。 上述案例分析证明了所提出的多层次可视分析方法的有效性。但是，该算法也有几个可提高的地方。

首先，上述案例分析主要集中在分析深度生成模型的训练过程。而所开发算法也可以直接应用于其他一些深度学习模型，例如卷积神经网络和多层感知器（MLP）。例如，在第一个案例中，所开发算法分析了生成式对抗网络的工作机理。而采用的生成式对抗网络的判决器本身就是一个卷积神经网络。因此，这个案例证明了所开发算法能够用于分析卷积神经网络的训练过程。更准确地说，所开发的算法能够分析神经元之间不形成有向环的网络（有环网络）。这类网络被称为深度前馈网络（deep feedforward networks）[7]。这类网络包含很大一部分深度学习模型，也是很多商业应用中的基础网络[7]。例如，卷积神经网络就是一种经典的深度前馈网络，并广泛地应用于人脸识别系统中。限制所提出算法应用于有环网络的可视化组件是，时间片层次的可视化。原因是本文采用了有向无环图布局算法计算每个网络中间层的位置。可以比较容易地将该部分拓展为可以分析有环网络，例如循环神经网络（RNN）。具体地说，可以将循环神经网络展开为深度前馈网络[17]，并利用所开发算法分析其训练过程。

其次，所提出算法需要专家提前训练完成一个模型，并将整个训练过程送入分析。这种离线的分析方式能够在很多情况下帮助专家理解和诊断训练过程。但是在于专家的讨论中，他们也表达了有的时候需要在线地分析一个训练过程。有的时候，深度生成模型的训练时间会长达几天[93]。在这种情况下，在线地分析训练过程能够帮助专家实时地检查训练情况，并在必要的时候停止训练以节省时间。解决这个问题的关键在于设计一系列能够有效展示流数据的可视化，并且开发配套的数据挖掘算法，例如在线蓝噪声采样和在线异常值检测等。

小结。 本章提出了一个基于责任分配的多层次可视分析方法，用于诊断深度生成模型整个训练过程，帮助专家交互地探索模型性能不佳或训练失败的原因。该多

层次可视分析方法与专家的典型多层次诊断过程（时间片，网络，以及神经元）是一致的。在时间片层次，提出了混合数据流可视化方法，结合有向无环图和折线图，有效展现数据在网络中的流动。在网络层次，采用了基于蓝噪声的折线采样算法，展现一个网络中间层中训练动态数据在训练过程中的变化，例如一个中间层中神经元的响应随时间的变化。在神经元层次，提出了责任分配算法，计算并展示了神经元之间的相互影响。基于该多层次可视分析方法，我们开发了 DGMTracker 系统，并用该系统帮助专家理解了生成式对抗网络训练过程中不同训练方法的影响，在此基础上诊断了一个失败的变分自动编码器训练过程。这两个案例证明了所提出的可视分析方法的有效性。

第 5 章　模型改进：基于不确定性的模型改进可视分析

本论文以微博检索为例，研究基于不确定性的模型改进可视分析方法，帮助专家将人的知识集成到检索模型中，提高模型整体性能。采用微博检索作为应用实例的原因是随着 Web2.0 时代的到来，微博逐渐成为人们分享想法的重要平台之一。研究者们也逐渐意识到这些数据能够帮助他们获取产品评价，分类消费者，衡量公民情绪，预测证券市场等[94-98]。上述研究能够顺利开展的前提是能够从海量的微博中检索出特定话题相关的重要微博。为此，研究者们开发了一系列的微博数据检索方法[99-100]。尽管这些方法已经能够在一定程度上帮助专家完成检索重要微博的任务，但是采用的数据检索方法可能产生错误。为了提高检索结果的质量，专家往往需要手工地检查检索结果，并利用一系列启发式算法去除不相关的内容。这个工作非常耗时以及其效果极大地依赖于专家的领域知识。

5.1　问题分析与建模

为了帮助专家更高效地改进检索模型，本章研究交互式模型改进技术，利用可视分析帮助专家有效提高模型整体性能。有效地交互改进检索模型，主要面临两个技术挑战。第一个挑战是高效地找到需要修改的模型组件。为了改进一个模型，目前专家需要逐个浏览感兴趣的模型统计信息，以定位检索结果中不正确的部分。该定位过程很耗时而且极大地依赖于专家的领域知识。专家往往需要查看众多信息才能定位到需要修改的模型组件。第二个挑战是有效地将专家的分析结果集成到模型中。在定位到需要修改的模型组件之后，专家需要修改对应的代码，并用训练脚本重新训练修改好的模型。随着大数据时代的到来，机器学习模型的训练时间越来越长。每次修改后都重新训练模型会极大地减慢开发过程。

为了应对这两个挑战，本文提出了基于不确定性的模型改进可视分析方法（图5.1）。给定模型的输出结果，该方法首先对输出结果的不确定性进行建模，使得专家可以更关注不确定性高的输出结果，从而减少专家的工作时间（挑战一）。除了输出结果的不确定性，该方法还分析了不确定性的传播，以期帮助专家从当前修改出发，找到其他需要修改的地方，进一步节省其工作时间。在分析了不确定性及其传播之后，该算法将分析结果（模型输出，不确定性及其传播）利用可视化直观地展现给专家。专家可以浏览这个可视化结果方便地定位到需要修改的地方。在专家修改之后，该算法增量式更新模型和对应的可视化，并支持专家进一步修改，从而使专家的改进过程成为一个完整的闭环。

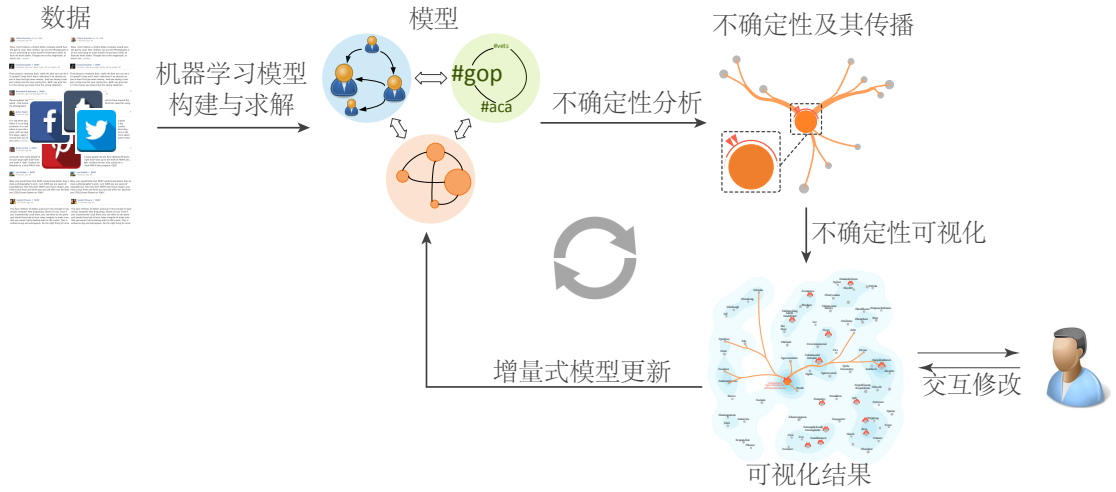


图 5.1　基于不确定性的模型改进可视分析方法概览

我们将这个算法框架应用到微博检索中，帮助专家交互地改进一个微博检索模型。本文将微博检索问题建模为互增强图模型（mutual reinforcement graph）。该建模能够有效考虑微博数据独有的特性，即微博数据不仅仅包含微博消息，还包括用户和标签。并且，这三个维度或者说三种元素（消息，用户，标签）是相互影响的。例如，一个有影响力的用户所发的消息往往比较重要。然而，现有的方法不能有机地结合这三个维度，它们抑或只考虑了微博消息，抑或是将微博消息作为主要的维度进行处理。例如，ScatterBlogs2 系统[99] 利用微博消息中是否含有专家感兴趣的关键词来检索专家感兴趣的微博。本文利用蒙特卡洛（Monte Carlo）采样方法求解该模型[9]，并根据采样结果分析检索结果的不确定性。另外，本文将不确定性在该图模型上的传播建模为一个马尔科夫链。为了帮助专家分析检索结果的不确定性并定位到需要修改的检索结果，本文设计了一个混合可视化。具体地说，将密度图与点线图相结合展现微博消息，用户和标签，以及他们之间的关系。用符号和流向图（flow map）分别表示图上元素（微博消息，用户和标签）的不确定性以及不确定性的传播情况。上述可视化与不确定性分析有机地结合在一起，能够帮助专家快速找到检索结果中最不确定的部分，并交互地进行修改。为了有效地将用户的修改融入模型，本文提出了一个增量式模型更新算法。该算法对图模型局部更新，能够高效地将用户的修改融入到图模型中，满足实时交互的需求。混合可视化会自动地根据更新后的模型更新可视化结果，从而使专家交互的修改过程形成一个有机的闭环。

接下来，具体介绍检索模型的不确定性分析，不确定性混合可视化方法，以及增量式模型更新算法。

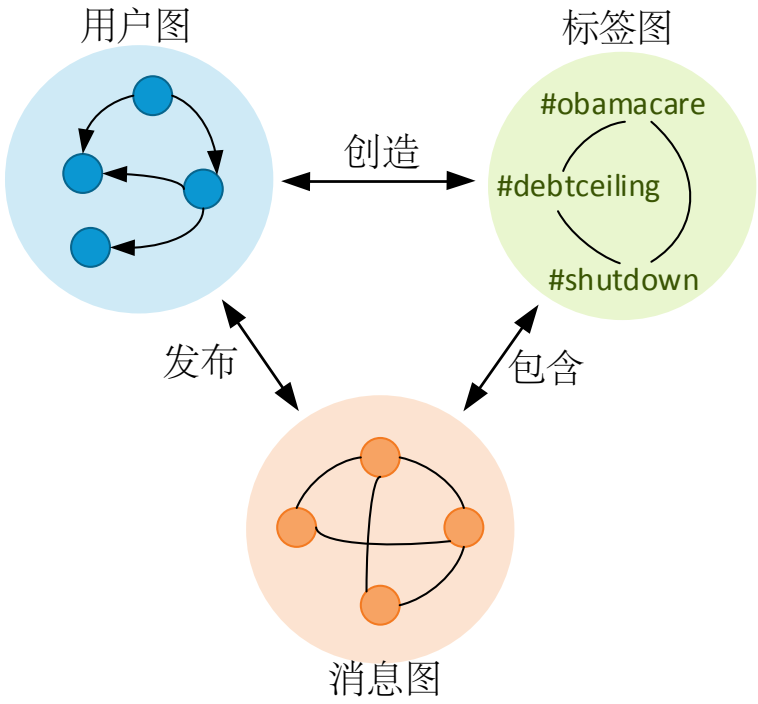


图 5.2　互增强图模型示意图

5.2　微博检索模型不确定性分析

本节介绍如何构建求解微博检索模型，以及如何对不确定性和传播建模。

5.2.1　基于互增强图模型的微博检索模型构建

互增强图模型[101-102] 的主要特征是它借助消息、标签和用户每一维度的关联图及其之间的关系来计算每一维度数据的重要度。这个特征能够减少专家与可视分析工具交互所花的时间。例如，如果一个专家修改了一个微博标签的重要性。互增强图模型不仅仅会修改与之相关的微博标签的重要性，还会修改相关的用户以及消息的重要性，从而减少所需的用户输入。这也是采用互增强图模型的主要原因。

互增强图模型的输入时三个图：消息图，微博用户图，标签图，以及三个图之间的关系。图5.2展示了这三个图以及他们之间的关系。用微博消息之间的消息相似度（余弦距离）构建微博消息图[101] 选取用户的关注与被关注关系构建微博用户图。标签图是利用两个标签之间共同在一条消息中出现的次数构建的。在此基础上，利用从属关系构建图与图之间的关系。具体地说，将每一条消息与其作者和其包含的标签相连接。另外，还将每一个用户与其在消息中提到的标签相连接。为了简便起见，在接下来的叙述中将消息，微博用户以及标签统称为微博元素。

基于上述输入，互增强图模型采用类似于 PageRank 算法[103] 的一种方法对微

博元素之间的相互影响进行建模：

$$\begin{bmatrix} R_p \\ R_u \\ R_h \end{bmatrix} = d \begin{bmatrix} \alpha_{pp}M_{pp} & \alpha_{up}M_{up} & \alpha_{hp}M_{hp} \\ \alpha_{pu}M_{pu} & \alpha_{uu}M_{uu} & \alpha_{hu}M_{hu} \\ \alpha_{ph}M_{ph} & \alpha_{uh}M_{uh} & \alpha_{hh}M_{hh} \end{bmatrix} \begin{bmatrix} R_p \\ R_u \\ R_h \end{bmatrix} + (1-d) \begin{bmatrix} W_p \\ W_u \\ W_h \end{bmatrix} \tag{5-1}$$

R_p，R_u 以及 R_h 是微博消息，用户以及标签的重要性组成的向量。M_{xy} 表示从 x 到 y 的连接强度，α_{xy} 是用来平衡各种元素之间增强关系的权重，d 是网页排序算法中的参数，与传统的网页排序算法采用相同参数取值 $d = 0.85$。W_p，W_u 以及 W_h 表示消息、标签和用户的先验重要度。为了进一步化简式 (5-1)，令 $R = \begin{bmatrix} R_p \\ R_u \\ R_h \end{bmatrix}$，$W = \begin{bmatrix} W_p \\ W_u \\ W_h \end{bmatrix}$ 以及 $M = \begin{bmatrix} \alpha_{pp}M_{pp} & \alpha_{up}M_{up} & \alpha_{hp}M_{hp} \\ \alpha_{pu}M_{pu} & \alpha_{uu}M_{uu} & \alpha_{hu}M_{hu} \\ \alpha_{ph}M_{ph} & \alpha_{uh}M_{uh} & \alpha_{hh}M_{hh} \end{bmatrix}$，式 (5-1) 可以被化简为：

$$R = dMR + (1-d)W \tag{5-2}$$

5.2.2　基于蒙特卡洛采样的微博检索模型求解

Duan 等[101] 基于式 (5-1) 利用矩阵乘法迭代地求解互增强图模型。这个算法是一个全局的算法。这意味着，如果需要更新某一个微博元素的重要性，需要更新所有微博元素的重要性，导致更新非常耗时。为了解决这个问题，本文采用蒙特卡洛采样近似求解互增强图模型该方法相较于 Duan 等的方法的优点如下[9]：

- 当输入数据局部变动时，只会影响局部元素的重要度；
- 对于重要度较高的元素，蒙特卡洛算法计算收敛速度更快；
- 因为蒙特卡洛算法可以从统计意义上计算出方差，所以可以准确建模每个元素重要度估计的不确定性。

为了采用蒙特卡洛算法求解，首先将式 (5-2) 转化为：

$$R = (1-d)(I - dM)^{-1}W = (1-d)(\sum_{k=0}^{\infty} d^k M^k)W \tag{5-3}$$

在此之后，以每个微博元素为起始点进行一系列随机游走。在每一个随机游走中，以 $1-d$ 的概率停止。如果没有停止，则根据矩阵 M 选择下一步的目的地。具体地说，矩阵 M 中的每一个元素 m_{ij} 表示从微博元素 i 到 j 的转移概率。

在 Duan 等[101] 的工作中，转移概率只取决于两个微博元素的相似度。这个方法的缺点是，如果一个用户发了大量毫无意义的消息，他的重要性有可能被高估。

为了解决这个问题，在计算转移概率的时候考虑微博元素的先验重要性，即将转移概率 m_{ij} 定义为 $similarity(i,j)\cdot w_j$。

在此基础上，根据获得的随机游走的样本计算每个微博元素的重要性。具体地说，令$Z=\sum_{k=0}^{\infty} d^k M^k$ 每个微博元素的重要性可以表示为：

$$r_j = (1-d)\sum_{i=1}^{N} w_i z_{ij} \tag{5-4}$$

其中，z_{ij} 表示从元素 i 开始的随机游走访问元素 j 的次数。w_i 表示每个微博元素的先验重要度。由此可见，可以利用获得的随机游走样本估计 z_{ij}，从而得到每个微博元素在检索中的重要性。

5.2.3　基于泊松混合模型的不确定性建模

由于采用蒙特卡洛采样近似地求解互增强图模型，检索结果具有一定不确定性。接下来，阐述如何计算每个微博元素计算出的重要性的不确定性。

在蒙特卡洛采样方法中，每个微博元素重要性的取值分布是已知的。因此，本文提出了一种基于概率的算法对每个微博元素重要性的不确定性进行建模。不确定性表示估计值在不同的情况下可能的差异大小[104]。

传统的方法将估计值建模为一个高斯随机变量[105-106]，并利用方差[105] 和标准差[106] 计算不确定性。这种方法适合估计值能够取正值和负值的情况。然而在微博检索这一应用中，重要性只能取正值，因此，传统方法无法使用。

根据 Avrachenkov 等[9] 的工作，式 (5-4) 中的 z_{ij} 服从泊松分布。由于计算出的重要性是 z_{ij} 的加权平均，每一个计算出的重要性服从混合泊松分布。对于混合波松分布来说，其方差和均值近似成线性关系。因此，如果采用方差衡量不确定性，会出现一个微博元素重要性越高，其不确定性越高的情况。标准差由于是方差的平方根也出现类似的情况。这是不可接受的。因此，方差和标准差都不能用来刻画泊松混合分布的不确定性。为了解决这个问题，本文采用了方差均值比（variance-mean-ratio）。对于一个微博元素来说，其重要性分布的方差均值比越高，其不确定性越高。具体地说，对于微博元素 j，其方差均值比的定义是：

$$u_j = v_j / r_j \tag{5-5}$$

其中，v_j 是元素 j 的重要性分布的方差。根据 Avrachenkov 等[9] 的工作，v_j 可

以用下式计算：

$$v_j = (1-d)^2 \sum_{i=1}^{N} w_i^2 v_{z_{ij}} \tag{5-6}$$

其中，$v_{z_{ij}}$ 是 z_{ij} 的方差。由于每个 z_{ij} 都服从泊松分布，它的方差可以用期望得到。

微博数据中海量的元素决定了不能将数据集中的所有元素都以单个的形式展现出来。因此，本文将相似的元素聚合在一起。对于一个聚类，其平均重要性 r_c 定义为其中元素的重要性之和[107]。

由于每个元素的重要性是独立的，那么一个聚类的方差 v_c，也可以表示为其中所有元素方差的和。进而，一个聚类的不确定性 u_c 可以用 v_c 除以 r_c 来计算：

$$u_c = v_c / r_c = \sum_{j \in c} (r_j / r_c)(v_j / r_j) = \sum_{j \in c} k_j u_j \tag{5-7}$$

式5-7表明，u_c 可以用聚类中元素的不确定性加权得到。而其权值 k_j 表示在这个聚类内部其重要性所占比重因此，一个聚类的不确定性主要由其中重要的元素决定，也是很合理的。

5.2.4　基于马尔科夫链计算不确定性传播

当专家找到了一个重要性计算错误的元素之后，他可以根据自己的领域知识进行修改。修改之后，本文计算了这个元素（聚类）的不确定性到别的元素（聚类）的传播，以期帮助专家找到被这个元素影响的其他需要修改的元素，进一步节省其工作时间。为此，我们显式地计算了互增强图上的不确定性传播。

在互增强图中，一个元素的重要度可以表示为其他相关元素重要度的加权平均。因此，其方差也可以表达为相关元素的方差的加权平均。又因为一个元素的不确定性由方差及重要度决定，因此一个元素的不确定性也可以由其他相关元素的不确定性线性表示为：

$$u_j = \sum_{i=1, i \neq j}^{N} m_{ij}^* u_i \tag{5-8}$$

其中 $m_{ij}^* = (d^2 m_{ij}^2 r_i)/((1-d^2 m_{jj}) r_j)$，表示两个元素不确定性之间的传播强度。式 (5-8) 表明元素之间的不确定性不是独立的，而是以线性形式在图上传播。因此，对于任意一对元素 i 和 j，$m_{ij}^* u_i$ 可以看做是从元素 i 到元素 j 的不确定性，并用 $u_{i \to j}$ 表示。

将式 (5-8) 重写为矩阵形式，可以将不确定性的传播建模为一个马尔科夫链：

$$UM^* = U, \tag{5-9}$$

其中 $U = [u_j]_{1\times N}$，$M^* = [m^*_{ij}]_{N\times N}$。

5.3 不确定性混合可视化方法

为了帮助专家在海量的微博数据中根据不确定性分析的结果找到需要修改的元素，我们设计了一个混合可视化，包括图可视化，不确定性符号，以及流向图。

具体地说，不确定性分析的结果不能单独展现，需要在检索结果作为上下文的情况下才有意义。为此，本文将检索结果以基于密度图的图可视化的形式展现出来，并支持专家从多个角度浏览重要的微博数据。这是因为微博数据中的多个维度会相互影响，更好地理解这种影响，能够帮助专家连接不同维度上的重要数据。例如，当找到一条重要的微博之后，检查相同作者的其他微博，以及含有相同标签的微博，能够帮助专家更快地找到相关的微博。

受箱线图（box plot）的启发，我们设计了表示不确定性的符号，来展示每一个微博元素聚类的不确定性分布情况。通过不确定性的展示，专家可以方便地找到不确定性最大的元素，并通过系统进行交互修改。

为了进一步节省专家的工作时间，本文还根据不确定性传播的情况，提出了基于力导向的流向图。该流向图能够帮助专家理解不确定性在微博不同元素之间的传播情况，以及找到受到这个检索结果的影响的其他检索结果。

5.3.1 基于密度图的图可视化展现检索结果

由于一条微博消息只与一个用户和几个标签相关，一条微博的影响比一个用户或者是一个标签要小。也就是说，修改一条微博消息的重要性只会影响到对应作者的重要性，及几个标签的重要性。而修改一个用户或是一个标签的重要性，会直接影响成百上千的微博。另外，微博消息的数量往往是用户数和标签数的十倍乃是上百倍。这就需要专家花更多的时间修改微博消息的重要性。因此，在可视化中，用户和标签作为主要元素，而微博消息作为次要元素进行显示。相应的，采用节点连接图的展示用户和标签，用列表展示微博。为简单起见，这里以标签关联图为例介绍所提出的可视化设计。为了用户能够更有效地浏览大图，本文对标签利用贝叶斯多分支树[108] 进行层次化聚类，聚类结果中的每一个非叶子节点都表示一个标签类。本文用一个堆积树来表示标签聚类的层次结构，并且结合密度图和节点连接图来展示所选层次结构上的标签。具体地说，在所选的层次结构上，

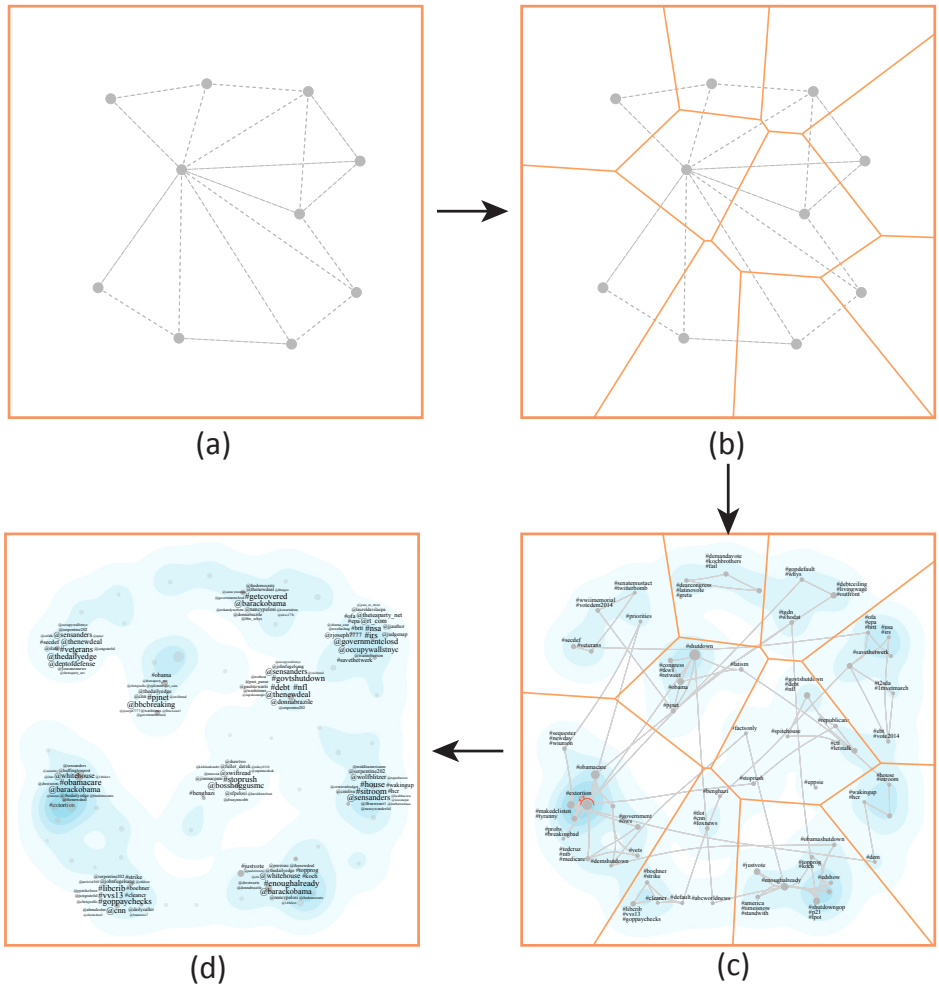


图 5.3　基于密度图的图可视化布局基本思想：(a) 放置标签聚类，并计算每个每个标签聚类的中心位置；(b) 根据标签聚类的中心计算对应的 Voronoi 剖分，并将每个剖分单元当做对应聚类的布局区域；(c) 计算代表性标签和非代表性标签的位置；(d) 生成带有用户上下文的最终可视化结果

首先从每一个标签类中选择若干代表性标签，并且将该类中其他非代表性标签分配给与之最近的代表性标签[109]。本文用节点连接图来展示代表性标签，用密度图表示非代表性标签。在布局中，用节点之间的距离表示它们所代表标签之间内容的相似度，每个节点的大小表示对应标签类的重要程度，与之相关的用户作为上下文布在其周围（图5.3(d)）。

布局。堆积树的布局比较直白，因此只介绍基于密度图的图可视化布局算法：

步骤一：计算在选定层次上聚类中心的布局。 我们首先建立了聚类之间的图。图中每个节点都表示一个聚类。如果两个聚类内部元素之间有足够多的连边（10），两个聚类之间也有边相连。在此基础上，用力导向的布局算法[110] 计算每个聚类的位置（图5.3 (a)）。

*步骤二：计算每个聚类所占区域。*在这一步中，利用 Voronoi 剖分计算每个计算每个聚类所占区域。上一步中计算出的每个聚类中心当做 Voronoi 剖分的起始点。在剖分之后，对应的 Voronoi 区域当做是聚类所占区域 (图5.3(b))。

*步骤三：计算代表性元素和非代表性元素的位置。*在这一步中，首先用力导向的布局算法计算代表性元素的位置。为了保证每个聚类中的代表元素在上一步计算出的聚类所占区域内，本文在每个区域的边界上加了一个对内部元素的排斥力。其他非代表性元素的分布用核密度估计算法[111]（kernel density estimation）计算 (图5.3(c))。

*步骤四：计算作为上下文的词云的布局。*同时展现标签图和用户图会导致严重的视觉混杂。为了解决这个问题，在主要关注标签图的时候，将标签图作为主要可视元素，而其用户信息显示为上下文，并支持专家在两个图之间切换。具体地说，当专家选中一个标签节点时，我们用一个词云（word cloud）作为上下文来显示对应的用户。在词云中。选中的标签节点被放置在正中间，对应的用户用基于扫描线的算法[112] 布局在其周围（图5.3 (d)）。

交互。为了支持专家从不同角度浏览排序结果，我们提供以下的用户交互操作。

浏览已排序的微博数据及其关系 (**R2**). 本文将基于密度分布的节点连接图以及堆积树结合在一起，帮助用户从不同视角和不同粒度浏览和理解排序结果。另外，本文还提供了若干过滤器，支持专家自定义所关心的数据。

在不同的数据维度之间切换 (**R3**). 受上下文弹出交互[113]（context popup interaction）的启发，本文在选定的元素周围提供进一步浏览的线索。例如，当一个专家在用户关联视图选择了一用户元素，其对应的标签元素就会以标签云的形式放置在其周围 (图5.9(a))。如果用户看到感兴趣的内容，就可以从用户关联图平滑地过渡到标签关联图 (图5.9(b))。

5.3.2　不确定性符号设计

不确定性的展现在可视分析过程中非常重要[105-106]。通过不确定性的展示，用户可以找到不确定性最大的元素，并通过系统进行交互修改。由于排序结果是以聚类的形式展现出来的，因此，需要展现每一个聚类的不确定性分布情况，其中包括这个分布的最大值，最小值，下四分位数（25%），上四分位数（75%）。

受箱线图（box plot，图5.4(a)）的启发，我们设计了表示不确定性的符号 (图5.4(b))。如图5.4(a) 所示，箱线图可以展现数据分布的最大值、最小值、极大值、极小值和上下四分位数值。为了将其与图上的节点紧密结合，将传统的箱线图转化为一个基于线的形式，然后将其包绕在每个节点的外侧（图5.4(b)）。我们还

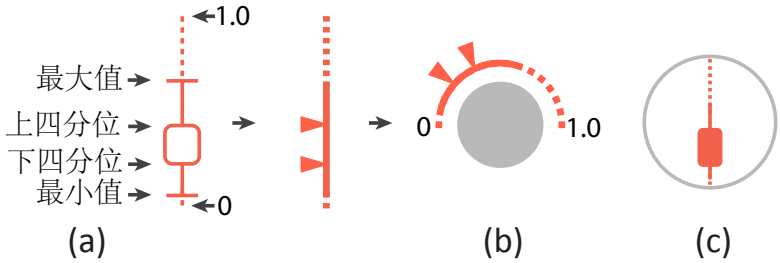


图 5.4　不确定性符号设计：(a) 箱线图；(b) 将箱线图变换为不确定性符号；(c) 另一种最终被舍弃的设计

图 5.5　四种不确定性分布的典型模式：(a) 聚类中大部分元素的不确定性较低，而少部分元素不确定性非常高；(b) 聚类中大部分元素的不确定性较高，而少部分元素不确定性非常高；(c) 均匀的不确定性分布；(d) 大多数元素不确定性很低

设计了别的不确定性符号以供专家选择。图5.4(c) 是其中一个例子。在于相关专家讨论之后，他们表示这个设计没有第一个设计好。他们认为中间部分大的聚类更能吸引他们的关注。但是这样的聚类只是因为上下四分位数相差比较大，而非不确定性大。经过充分的讨论，我们决定用图5.4(b) 中的符号表示不确定性分布。

基于这个符号，专家可以获得一个聚类中不确定性分布的概览，并找到感兴趣的聚类。图5.5展示了几个典型的不确定性分布模式。例如，如图5.5(a) 所示，聚类中的大部分元素的不确定性较低，而少部分元素不确定性很高。因此，专家会倾向于只浏览该聚类中不确定性较高的这些元素，从而减少工作时间。

5.3.3　利用流向图展示不确定性传播

在可视化领域，流向图[114-115] 被用来展现事物或人从一个地点向多个地点流动的情况。受此启发，本文提出了不确定性传播的展现方法 (图5.6)，该方法能够帮助专家快速地从已知的不确定性值比较高的节点找到更多未知的不确定的节点。**布局。** 在实际应用中，用户经常比较多个节点的不确定性传播图。为了减少展现多个传播图引起的视觉混乱并且方便用户比较多个节点不确定性传播模式，本文利用基于单个流向图[115] 和图的边绑定技术[116] 展现多个节点的不确定性传播情况的算法。该布局算法分为以下三步：

步骤一：不确定性传播计算及初始单个流向图布局。. 给定一些选择的节点，首

先计算不确定性传播路径。然后生成以每一个点为起始点的流向图的初始布局(图5.6(a))。

步骤二：计算不同流向图中边之间的相容性。在这一步中，利用边绑定算法中的相容性计算方法[116]计算并绑定相容性相近的边。相容性表示两个边的相似程度，其包含以下几个方面。第一个是角度相容性。其目的是绑定具有相似夹角的边，定义是：

$$C_\alpha(e_i, e_j) = |cos(\alpha)|.$$

第二个是长度相容性。其目的是绑定相似长度的边：

$$C_s(e_i, e_j) = 2/(l_{avg} \cdot min(|e_i|, |e_j|) + min(|e_i|, |e_j|)/l_{avg}),$$

这里，$l_{avg} = (|e_i| + |e_j|)/2$。

第三个是位置相容性。其目的是绑定相似位置的边，定义为：

$$C_p(e_i, e_j) = l_{avg}/(l_{avg} + \|Q_{m1} - Q_{m2}\|),$$

其中，Q_{m1} 和 Q_{m2} 分别是是边 e_i 和 e_j 的中点。

综上所述，整体的边相容性的定义是：

$$C_e(e_i, e_j) = C_\alpha(e_i, e_j) \cdot C_s(e_i, e_j) \cdot C_p(e_i, e_j).$$

图5.6(b) 展示了多个流向图匹配之后的结果。

步骤三：利用力导向技术绑定匹配后的多个流向图。基于匹配结果，用力导向技术绑定匹配后的多个流向图。边 e_i 上的点 p_i 所受的力定义为：

$$F_{p_i} = K_i(\|p_{i-1} - p_i\| + \|p_i - p_{i+1}\|) + \sum_{e_j \in E} \|p_i - p_j\| \cdot C_e(e_i, e_j)$$

其中，K_i 是弹簧系数，E 是匹配的所有边 e_i 的集合。上式中第一项用于保证生成平滑的流向图，第二项用于将匹配的边绑定起来。图5.6(c) 展示了将多个流向图绑定之后的结果。

5.4　增量式模型更新

为了有效地将专家的反馈融入到模型之中，本文根据专家的反馈增量式地更新模型，而非重新计算，以减少专家等待的时间。能够增量式更新模型的关键在于采用了蒙特卡洛采样算法计算各个元素的重要性。增量式更新模型算法的基本

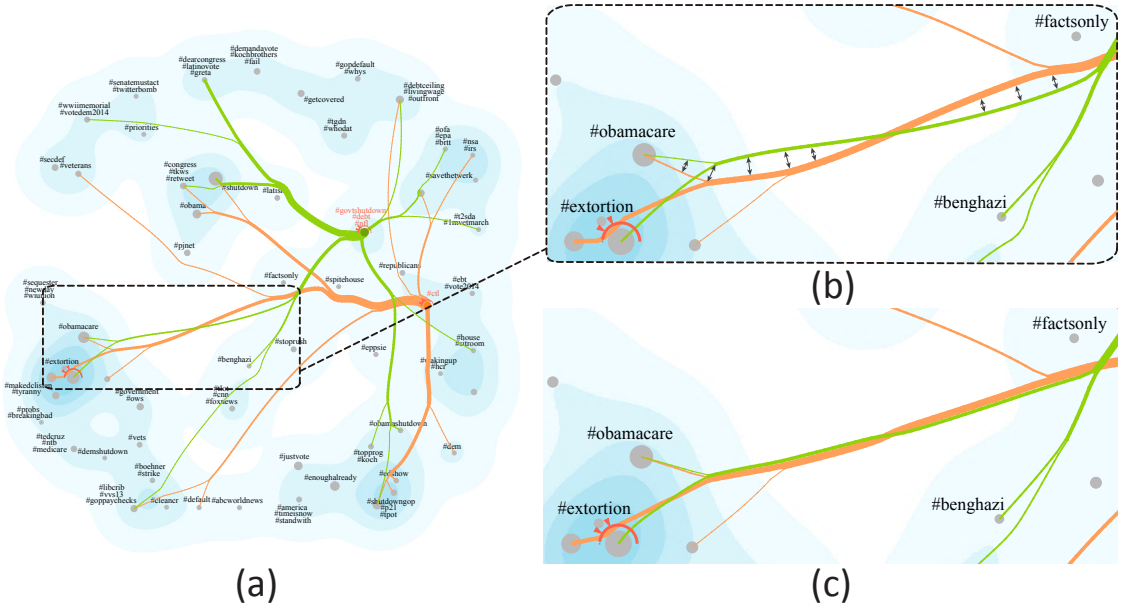


图 5.6　多个不确定性传播的共同布局：(a) 基于流向图算法生成的初始布局；(b) 利用边的相容性的匹配结果；(c) 绑定匹配的边的可视化结果

思想是，当专家修改选定元素的重要性之后，该算法会增量式更新元素的先验重要性。相应的，相似度矩阵 M 会变为 M'。这个改变会影响蒙特卡洛采样算法采出的少部分样本（随机游走）。

对于这部分被影响的随机游走，现有的增量式算法[117] 首先重采样这部分随机游走，然后计算新的元素重要性。该算法主要的问题在于重采样过程相对耗时，无法满足专家实时交互的需求。具体地说，假设 n 是一个元素邻居数量的平均值，以及 l 是随机游走的平均长度。在随机游走的每一步，都需要从一个 n 维的多项式分布（multinomial distribution）中采样出下一步的目的地，其开销为 $O(n)$。因此，完整地采样一个随机游走的开销为 $O(nl)$。计算这个随机游走样本对元素重要性的影响的开销为 $O(l)$。综上所述，采样一个随机游走样本并并计算其影响的开销为 $O(nl) + O(l) = O(nl)$。

需要注意的是，在交互模型修改这个应用中，不会删除或者添加图上的边。因此，不需要进行重采样，而只需要根据改变后的相似度矩阵，更新受影响的随机游走样本对元素重要性的影响。这样，也就避免了高开销的重采样过程，使得更新一个受影响的随机游走样本的开销减小到 $O(l)$。

给定一个随机游走样本 $path = \{i \to n_1 \to \ldots \to n_{k-1} \to j\}$。在其上定义一个新的随机变量 x_{ij}^k。$x_{ij}^k = 1$ 表示这个随机游走样本从 i 出发，经过 k 步，到 j 终止。在更新前，随机游走样本中的每一步的权重都为 1。在更新之后，该权重可以重新计算为 $P'(x_{ij}^k = 1)/P(x_{ij}^k = 1)$。其中，$P(x_{ij}^k = 1)$ 为根据 M，$x_{ij}^k = 1$ 的概率。相应的，

$P'(x_{ij}^k = 1)$ 表示根据 M'，$x_{ij}^k = 1$ 的概率。因此，$P(x_{ij}^k = 1)$ 可以由下式得到：

$$P(x_{ij}^k = 1) = m_{in_1} m_{n_1 n_2} ... m_{n_{k-1} j} \tag{5-10}$$

$P'(x_{ij}^k = 1)$ 可以用相同的计算方式得到。

图5.8(c)- (f) 展示了专家修改和增量式更新的一个实例。可以看到，若干标签聚类的重要性都改变了（反映在节点大小上）。这里我们设计了一个符号表示重要性的改变。具体的说，橙色虚线圈表示更新之前的重要性，灰色实心圈表示更新后的重要性（图5.8(d)-(f)）。

5.5 算法应用：MutualRanker 系统

我们根据基于不确定性的交互式模型改进算法，开发了可视分析工具 MutualRanker。该工具能够有效地展示微博检索结果及其不确定性，并支持专家交互地修改模型的检索结果。我们利用该系统进行了定量分析和案例分析，以证明所开发算法的有效性。

5.5.1 系统概览

MutualRanker 包含以下模块：

- 一个互增强图模型用来检索重要的微博消息，微博用户，和标签；
- 一个不确定性模型用来估计检索结果的不确定性以及在图上的传播情况；
- 一个混合的可视化用来展现检索结果，不确定性以及不确定性的传播。

MutualRanker 的主要目标是根据专家给定的检索条件检索出相关而且重要的若干微博消息，微博用户以及标签。给定一个微博数据库，预处理模块首先构建微博消息图，微博用户图以及标签图。互增强图模型利用这三个图计算出每个微博消息，用户以及标签的重要性，并按照重要性将这些元素进行排序，从而产生检索结果。不确定性分析模块之后会估计出检索结果的不确定性以及不确定性的传播情况。基于检索结果和不确定性分析结果，可视化模块利用一个混合可视化展现这些分析结果。具体地说，这个混合可视化包含一个图的可视化，一个不确定性符号和一个流向图。专家可以根据可视化结果交互地调整每个微博元素的重要性。MutualRanker 工具会根据专家的输入增量式地更新每个微博元素的重要性。

5.5.2 定量分析

本文首先验证所开发增量式模型更新算法的有效性。这里采用了两组推特数据集，分别是 Shutdown 数据集，和 Ebola 数据集。Shutdown 数据集包含 2013 年美

表 5.1　增量式模型更新算法的数值验证

修改步数	Ebola 数据集			Shutdown 数据集		
	消息	用户	标签	消息	用户	标签
0(初始结果)	0.800	0.710	0.870	0.900	0.875	0.860
1	0.815	0.715	0.875	0.910	0.875	0.865
2	0.840	0.720	0.880	0.915	0.875	0.865
3	0.855	0.720	0.885	0.925	0.885	0.875
4	0.855	0.720	0.890	0.925	0.885	0.880
5	0.855	0.720	0.895	0.925	0.885	0.885

国政府关门相关的推特（5,132,510 条推特，2013 年 10 月 1 日-10 月 16 日）。数据集由关键词 “shutdown’’ 抽取，并去除了标签中的一系列停用词，例如 “#retweet’’，“#rt’’，“#path’’，以及 “#road’’。Ebola 数据集包含关于 2014 年埃博拉病毒爆发相关的推特（1,425,017 条推特，2014 年 1 月 1 日-12 月 25 日）。

为此，我们邀请了两名专家使用所开发的 MutualRanker 工具。其中一位专家（S）的研究方向是社会学，另一位专家（C）的研究方向是传媒学。这两位专家都有丰富的微博检索经验。在他们的研究项目中，经常需要检索与某事件相关的重要推特消息。在本实验中，一名专家分析 Shutdown 数据集，另一名专家分析 Ebola 数据集。他们分别根据互增强图算法给出的初始重要性，按照个人的理解交互地修改模型。具体地说，当一名专家找到一个重要性被低估的元素，便增大其重要性。反之降低其重要性。在每一次修改之后，我们检查重要性排名前两百的推特，用户和标签是否真的重要。为了方便地比较，我们用 $n-$ 准确率衡量结果的好坏。$n-$ 准确率的定义是在排名前 n 个元素中，正确的元素所占比例。该衡量方式常常用于召回率 (recall) 很难计算的情况[118]。在微博检索中，由于不知道全部重要的元素（例如微博），召回率无法计算。因此使用 $n-$ 准确率衡量检索结果的质量。

通过观察，我们发现最多经过五次修改，结果几乎不变。因此，将修改限制为五步。每次修改后的 200-准确率如表5.1所示。可以看出，检索结果的质量随着修改逐渐提高。这个结果验证了所开发的基于不确定性的交互式模型修改算法，能够根据专家的反馈改进模型的结果。

进一步，我们观察到在某些修改之后，不止一种元素的检索结果改变了。例如，在 Ebola 数据集中，专家修改了第一个元素的重要性之后，消息/用户/标签检索结果的质量都提高了。这个结果表明，采用互增强的图模型，能够更好地将专家地反馈传播到不同种类的元素上，从而减少他们的工作时间。

5.5.3　案例分析

进一步验证所开发算法的有效性，在 Shutdown 数据集上我们邀请传媒学方向的研究员（C），使用 MutualRanker 系统，完成：(1) 利用不确定性分析的结果找到重要性计算错误的元素。(2) 根据其领域知识逐渐改进排序模型。(3) 最终检索出和美国政府关门事件中最重要的推特/用户/标签。

排序结果概览。根据标签排序结果的概览 (图5.7(a))，专家发现在政府关门这个话题下面，有若干子话题。例如，奥巴马医改 (图5.7A)，关于两党的讨论 (图5.7B)，希望结束关门 (图5.7C)，政府关门的影响 (图5.7D)，媒体对于政府关门的报道 (图5.7E)，债务相关的讨论 (图5.7F)，以及对政府关门的批评 (图5.7G)。

不确定性分析。从标签的概览中，“#shutdown’’ 这个标签聚类吸引了专家的注意力，因为这个聚类的不确定性很高，而且重要性很大 (图5.7(b))。专家查看了这个聚类中具体的标签和推特。他发现除了常见的一些标签：#govtshutdown，#obamashutdown 和 #shutdowngop，推特用户创建了多种多样批评政府关门的标签，例如 #shutdown-harry 和 #dontcutkids（公众运动）。专家想从最不确定的标签开始分析。因此对标签按照不确定性排序。有趣的是，标签 #lewinsky 的不确定性最高 (图5.7(c))。专家浏览了相关的推特，发现这个标签是和克林顿政府 1995 年关门事件相关的。由于专家只关心这次政府关门，因此他降低了这个标签的重要性。由此可见，不确定性分析能够一定程度上帮助专家找到最需要修改的地方。

不确定性传播。接下来，专家想检查 “#shutdown’’ 聚类的不确定性，如何影响其他的聚类。因此，他使用 MutualRanker 提供的流向图，查看了该聚类对其他聚类的影响。另外，他一并选择了民主党和共和党两个与 “#shutdown’’ 聚类紧密相关的标签聚类，一同查看这三个标签聚类的影响（图5.8）。如图5.8(b) 所示，“#nationalparks’’ 这个聚类受到三个聚类的共同影响。国家公园的关闭正是政府关门的后果之一，并且在推特上引发了对两党广泛的批评。专家决定提高其重要性（由 4 变大为 6）。在 MutualRanker 中，重要性归一化到 1-10。10 表示重要性最高。

在这样修改之后，专家注意到另外两个标签聚类 “#spitehouse’’ 和 “#teaparty.’’ 的重要性也自动地提高了。通过查阅其中推特，发现这两个聚类是和批评两党相关的，其中一个批评的原因就是国家公园关闭导致游人无法进入，例如：“@RepBradWenstrup @sarahlance #shutdown #Nationalpark Here’s what my tea-party-backed #Republican did to my vacation.” 第一个聚类中是和民主党相关的，其中诸如 “#spitehouse’’ 和 “#demshutdown’’ 这样的标签的重要性有所提高。在第二个聚类是和共和党相关的，其中诸如 “#teaparty’’ 和 “#defundgop’’ 的重要性有所提高。因为专家选择了民主党和共和党这两个标签聚类，与批评这两个党相关的标签的重要性提

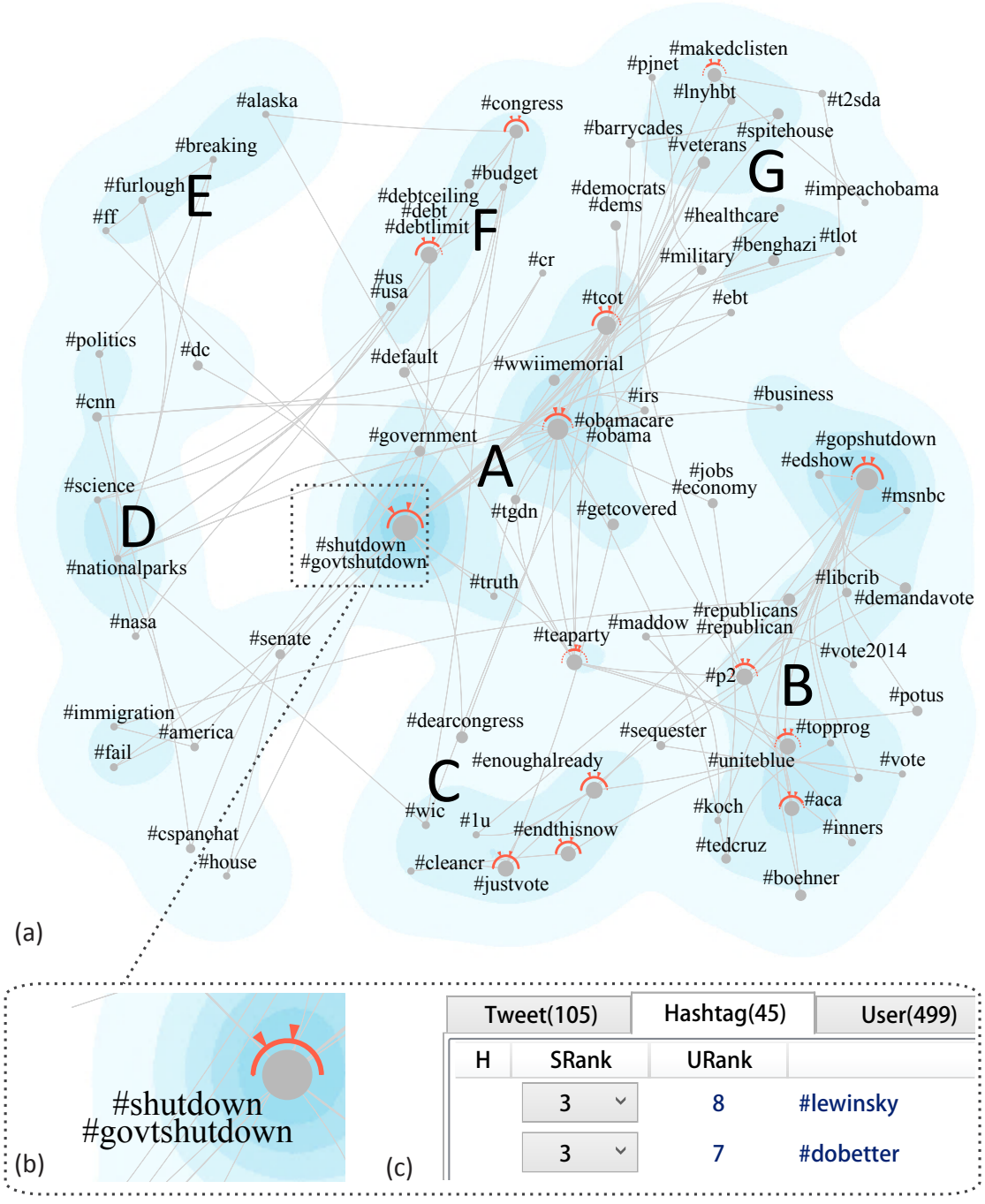


图 5.7　Shutdown 数据集检索结果概览

高是合理的。专家表示，他开始只是想找到批评两党的原因的标签聚类。“#nationalparks’’ 这个聚类很自然地被找到了。他没有想到的是，通过修改这个标签聚类的重要性，还会对批评两党本身的标签聚类的重要性产生影响。在他的计划中，他想从类似 “#nationalparks’’ 这样的标签聚类出发找到更多批评两党的标签聚类，而算法自动地帮助他完成了这个操作，的确有效地减少了他所花的时间。

另外，“#ebt’’ 这个聚类的重要性自动地降低了。具体地说，其中诸如 “#ebt’’ 这

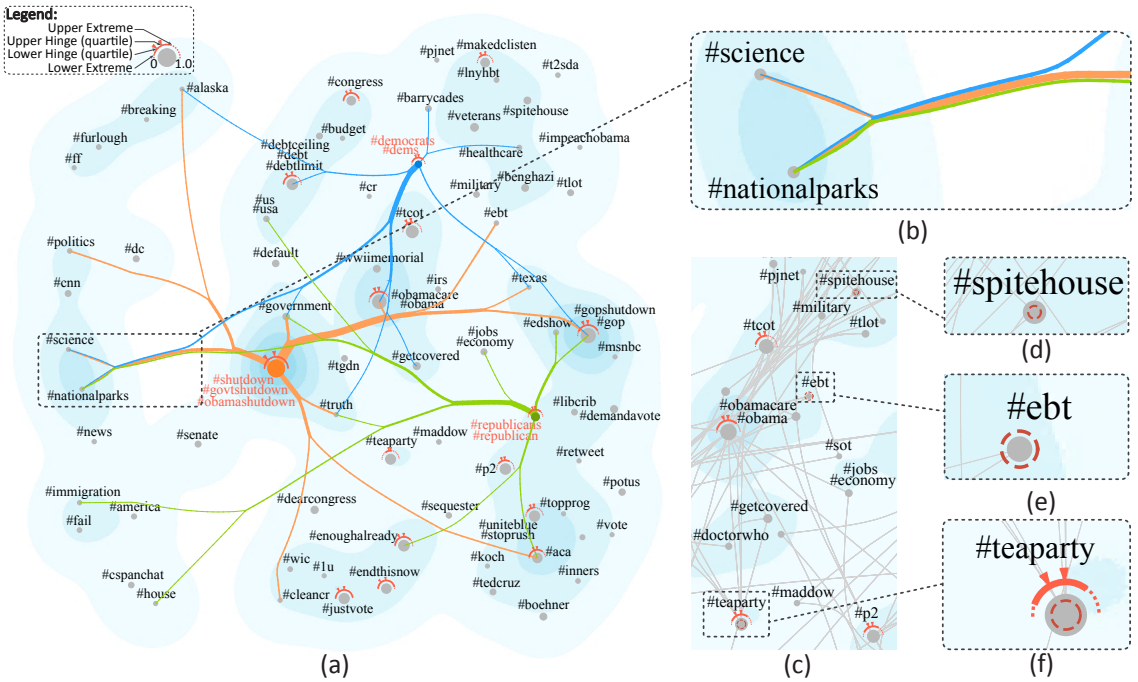


图 5.8　交互式修改 Shutdown 数据集对应的检索模型：(a) 标签图及图上不确定性分布和不确定性的传播；(b) 不确定性传播；(c)-(f) 交互修改检索模型

样的标签的重要性降低了。通过查阅相关推特，发现 EBT 系统在政府关门期间宕机了。很多人怀疑这也是政府关门的后果：“Ahh... #ebt not working cause if a #governmentshutdown? How sad you can’t spend money taken from me against my will that I worked for...” 之后，官方辟谣称此次宕机只是技术问题，并不涉及政府关门：“According to NBC, #ebt is down because of a technical issue, NOT #governmentshutdown”. 因此，这个自动的修改也是合理的。

在用户和标签视图间切换。 除了重要的标签，专家还想找到政府关门讨论中重要的推特用户。因此，他以词云的形式显示了与 “#shutdown’’ 聚类的相关的用户 (图5.9(a))。接着，他切换到这些用户对应的用户视图上 (图5.9(b) 和图5.9(c))。专家从用户视图的概览中找到了一些重要的用户图5.9(b) 和图5.9(c)。这些重要的用户可以分为两组：1) 政府官方账号：“@barackobama’’，“@whitehouse’’ 等。(图5.9(b))；2) 媒体官方账号：“@nytimes’’，“@guardian’’，“@bloombergnews’’ 等 (图5.9(c))。除了这两类账号，专家还对重要的政党领袖比较感兴趣。但是她发现有些政党领袖的账号的重要性低估了，例如 @speakerboehner (Rank 8), @whiphoyer (Rank 8), @nancypelosi (Rank 7)。专家认为在实际生活中，这些政党领袖很重要。但是他们在推特上，往往不是很活跃，导致其重要性被低估。因此，专家手动将政党领袖账号的重要性调高。具体地说，他将 “@speakerboehner’’，“@whiphoyer’’ 以及 “@nancypelosi,’’ 的重要性都调为最高等级 10. 图5.9(d) 展示了修改之后其他元素

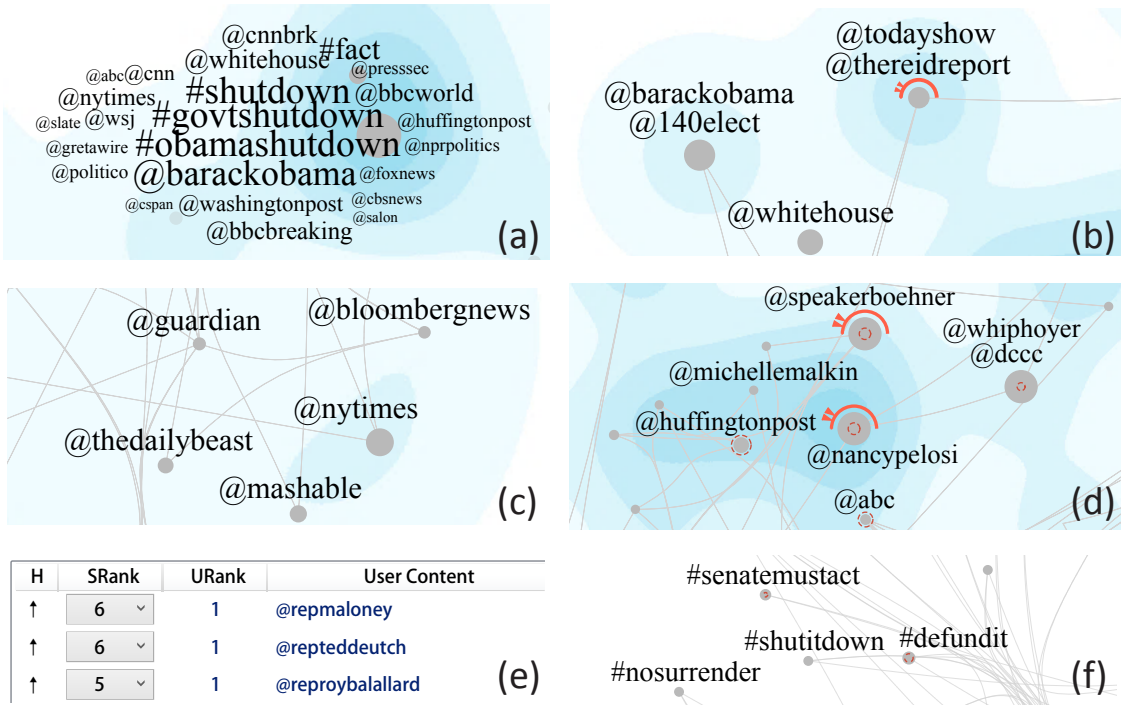


图 5.9　在标签视图和用户视图间切换：(a) 在关注的标签聚类上将用户以上下文形式展示；(b) 用户视图中关键的政府官方账号；(c) 用户视图中的关键媒体账号；(d) 修改用户的重要性之后；(e) 一些政治家用户更新后的重要性；(f) 切换回标签视图

的重要性的变化。

在修改之后，专家注意到 “@whiphoyer’’ 这个聚类的重要性显著地提高了。其中几个用户的重要性自动地提高了 (图5.9(e))。例如，“@repmaloney,’’ 和 “@repted-deutch,’’ 的重要性都由 5 变为 6。专家表示，这些都是国会的议员，在调高了政党领袖的重要性之后，这些议员重要性的提高是很合理的。他进一步表示，这种自动地修改相关元素重要性的机制，能够帮助他在只知道一少部分重要用户的情况下，也能将大多数重要的用户自动地检索出来。

在修改了用户的重要性之后，专家切换回标签视图，检查用户上的修改对标签的影响。他发现了一个新的标签聚类 “#senatemustact’’。通过查阅聚类相关的推特，他发现这个聚类主要是在批评重要的国会议员（图5.9(f)）：“@PeteSessions #DefundObamacare #shutdown #MakeDCListen #senatemustact Stand for the American People!” 而这个聚类的出现也正是提高了国会议员重要性的结果。

5.6　讨论及小结

局限性。 对于所开发算法，有两个可以改进的地方。其一是目前该算法只能分析离线数据。将该算法拓展到分析流数据，能够有效地检索过去正在发生的事件中

重要的微博数据。该拓展的关键技术挑战在于，管理用户的修改和流数据的不断流入。具体地说，用户的修改和流数据的不断流入都会导致可视化发生改变。需要有一个管理机制在用户修改的时候阻塞数据的流入，但同时还要提醒用户数据正在流入但被阻塞。第二个可以改进的地方是，可视化设计稍显复杂。虽然该算法的目标用户是有一定领域知识的专家，但是和我们合作的专家相信该算法也能帮助普通用户检索感兴趣的信息。专家建议采用更加直观的可视化设计。例如，不确定性符号可以修改为只展示一个聚类中的平均不确定性，而非更加细节的不确定性分布情况。第三个可以改进的地方是，在案例分析和数值实验的基础上，采用更多样化的验证方法以验证所提出方法的有效性。可以通过用户研究，即在测试数据集上与同类或者类似的可视化系统（例如 ScatterBlogs2[99]）从性能和效率等多个方面进行分析比较，验证所提出算法可以更好地帮助专家交互地修改。对于第三章和第四章所提出算法也可以采用类似的方法，加强对算法有效性的验证。

小结。 本章提出了基于不确定性的模型改进可视分析方法，帮助专家将人的知识集成到微博检索模型中，提高模型整体性能。。该方法能够检索专家感兴趣的微博消息，用户，标签，并分析检索结果的不确定性。具体地说，本文利用了社交网络上微博消息，用户和标签，三者相互影响的特性，将微博检索问题建模为互增强图模型。在该模型中，微博消息的重要性，微博用户的社会影响力，以及微博标签的流行程度三者相互影响。我们利用蒙特卡洛采样方法求解该模型，并根据采样结果分析该求解过程的不确定性。另外，本文将不确定性在该图模型上的传播建模为一个马尔科夫链。为了帮助专家理解检索结果和其不确定性。我们设计了一个混合可视化。具体地说，本文将密度图与点线图相结合展现微博消息，用户和标签，以及他们之间的关系。本文用符号和流向图分别表示图上元素（微博消息，用户和标签）的不确定性以及不确定性的传播情况。上述可视化与不确定性分析有机地结合在一起，能够帮助专家快速找到检索结果中最不确定的部分，并交互地进行修改。我们基于该交互改进算法开发了 MutualRanker 系统，帮助专家交互地改进微博检索模型。我们利用该系统完成了数值实验和案例分析，以验证所提出框架和所开发算法的有效性。

第 6 章　总结与展望

6.1　本文工作总结

本论文针对机器学习模型开发过程中三个主要任务：理解、诊断和改进，提出了三个对应的可视分析方法，帮助专家直观地理解机器学习模型的工作机理，更方便地诊断模型的训练过程，以及更高效地改进模型的预测性能。

首先，本论文研究卷积神经网络工作机理分析与理解的可视分析，帮助专家理解训练过程中单个时间片上的训练状态。本文提出了基于大规模有向无环图和多层次聚类的可视分析方法。根据卷积神经网络的结构特点，本文将网络建模为一个有向无环图。为了有效处理大规模网络，本文提出了网络级别和神经元级别聚类方法，将该有向无环图聚合为一个更加紧凑的图。聚合后的有向无环图中，每一个节点是一个神经元聚类，而边表示神经元聚类间的连边。根据聚合后的有向无环图，本文利用节点链接图提供网络结构的概览，并允许专家从概览出发，交互地探索网络的内部工作机理。为了帮助专家理解网络中各个组件的作用，本文提出了一个大规模有向无环图可视化方法，帮助专家浏览神经元聚类不同方面的信息以及神经元聚类的连边。该可视化方法紧密结合矩形布局，矩阵重排和基于双聚类的边绑定技术，有效展现神经元聚类学到的特征、响应和对网络的贡献，以及神经元聚类之间的连接关系。

其次，本论文提出了深度生成模型训练过程诊断的可视分析方法，帮助专家交互地探索模型性能不佳或训练失败的原因。为了支持专家的诊断过程，本文根据专家的典型诊断过程提出了多层次可视分析方法，建立起沟通整体统计信息与细节的训练动态数据之间的桥梁。在时间片层次，本论文结合有向无环图和折线图，有效展现数据在网络中的流动。在网络层次，本论文利用基于蓝噪声的折线采样算法，减少由大量训练动态数据带来的视觉混乱并保留异常值。在神经元层次，本论文提出了责任分配算法，揭示神经元之间的相互影响，帮助专家诊断模型训练失败的根本原因。

最后，本论文提出了基于不确定性的模型改进可视分析方法，帮助专家将人的知识集成到检索模型中，提高模型整体性能。本论文以微博数据为例，将其检索问题建模为互增强图模型。该图模型能够有效考虑微博数据独有的特性，即微博数据不仅仅包含微博消息，还包括用户和标签。并且，这三个维度或者说三种元素（消息，用户，标签）是相互影响的。本文提出了基于蒙特卡洛采样的模型求解方法，并计算了检索结果的不确定性，以及不确定性在图上的传播。相应地，本

论文紧密结合图可视化、不确定性符号以及流向图等多种可视化技术有效展现这些信息，帮助专家找到最不确定的检索结果，并交互地修改。另外，本论文提出了增量式模型更新算法，根据专家的修改逐步改进模型，形成一个迭代循环的模型改进过程。

基于这三个可视分析方法，我们开发了三个相应的可视分析工具 CNNVis, DGMTracker 和 MutualRanker 帮助专家理解、诊断和改进机器学习模型。为了验证所提出方法的有效性，本论文进行了数值实验和一系列案例分析。这些实验表明所提出方法能够有效地帮助专家理解卷积神经网络的工作机理，诊断深度生成模型的训练过程，以及交互地改进微博检索模型，提高了机器学习模型的可解释性，帮助专家快速设计符合需求的机器学习模型。

6.2　未来工作展望

本文的研究工作能够帮助机器学习专家完成模型开发过程中的主要任务。从另外一个方面，也就是分析对象（例如模型和训练过程）上看，机器学习模型的可视分析依然有着广阔的研究前景。一个机器学习模型可以看做由三部分组成：数据、模型和训练过程（将数据与模型相结合的过程）。下面我们简单讨论这三个方面未来可能的研究方向。

数据。在机器学习模型的可视分析中，现在研究者主要关注正常的数据。最近，研究者发现通过在正常数据上叠加一个恶意的噪声，能够产生一个人类很容易识别正确，但是机器学习模型会以很高置信度判错的样本[119]。这类样本称为对抗性样本（adversarial example）。例如，通过在一张人和模型都能轻易识别的熊猫图片上，叠加一个恶意噪音，会产生一个对抗性图片。该图片人依然可以轻易识别为熊猫，但是模型会将其识别为猴子。这类对抗性样本对机器学习模型在安全性要求高的领域（例如无人驾驶）中的应用提出了挑战。因此，一个可能的研究方向是利用可视分析比较对抗性样本与正常样本在网络中的流动情况，从而揭示对抗性样本的工作机理。这也能对专家开发更鲁棒的机器学习模型有一定指导意义。

模型。由于深度神经网络的优异性能，现在研究者主要研究深度神经网络的可视分析。在多种多样的深度神经网络中，卷积神经网络受到了最广泛的关注[13,120]。最近，也有部分研究者开始研究其他类型的深度神经网络的可视分析，例如循环神经网络[121-122] 和深度生成模型[123] 等等。除此以外，还有大量各具特点而且广泛应用的深度神经网络值得研究，例如，深度置信网络 (deep belief network)。如果将视野扩展到整个机器学习领域，深度神经网络只是机器学习模型大家族中很小的一部分。实际生产生活中，诸如决策树等其他传统机器学习模型由于其鲁棒性容

易训练等特点也有着广泛的应用。这些传统的机器学习模型也是值得研究的。研究重点可以放在，在实际应用中如何利用可视分析帮助领域知识较少的用户理解和信任这些模型。

训练过程。目前，研究者主要研究训练过程的离线分析。而训练过程的在线分析也是值得研究的。如第四章所述，训练过程的在线分析，能够帮助专家实时地检查训练情况，并在必要的时候停止训练以节省时间。另外，现在研究者主要研究训练过程中时间序列数据（例如权值随时间的变化）的有效展示和分析。虽然这部分时间序列数据已经能够一定程度上帮助专家理解与诊断训练过程了。但是，要更加深入地理解一些机器学习模型的训练过程，只展示这些时间序列是不够的。例如，在深度强化学习中，训练过程实际是训练个体和环境的不断交互与反馈学习过程。想要深入地理解这个过程，需要展现出个体的决策变化等信息。如何将抽象的决策变化以直观的方式展现出来是很值得研究的。再例如，生成式对抗网络的训练过程，可以看做是判决器和生成器二者的博弈。有效地展现这个博弈过程能够帮助专家更深入地理解生成式对抗网络的工作机理。

参考文献

[1] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//IEEE Conference on Computer Vision and Pattern Recognition. 2016: 770-778.

[2] Koh P W, Liang P. Understanding black-box predictions via influence functions[C]//International Conference on Machine Learning. 2017: 1885-1894.

[3] Wongsuphasawat K, Smilkov D, Wexler J, et al. Visualizing dataflow graphs of deep learning models in tensorflow[J]. IEEE Transactions on Visualization and Computer Graphics, 2018, 24 (1): 1-12.

[4] Gunning D. Explainable artificial intelligence (xai)[J]. Defense Advanced Research Projects Agency (DARPA), 2017.

[5] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(8): 1798-1828.

[6] Zeiler M D, Fergus R. Visualizing and understanding convolutional networks[C]//European Conference on Computer Vision. 2014: 818-833.

[7] Goodfellow I, Bengio Y, Courville A. Deep learning[M]. Cambridge: MIT Press, 2016

[8] 曹科垒. 折线图的蓝噪声采样及其在可视化中的应用[硕士学位论文]. 北京: 清华大学计算机系, 2017.

[9] Avrachenkov K, Litvak N, Nemirovsky D, et al. Monte Carlo methods in PageRank computation: When one iteration is sufficient[J]. Journal of the Society for Industrial and Applied Mathematics, 2007, 45(2): 890-904.

[10] Paiva J G S, Schwartz W R, Pedrini H, et al. An approach to supporting incremental visual data classification[J]. IEEE Transactions on Visualization and Computer Graphics, 2015, 21 (1): 4-17.

[11] Tzeng F Y, Ma K L. Opening the black box - data driven visualization of neural networks[C]// IEEE Visualization. 2005: 383-390.

[12] Zahavy T, Ben-Zrihem N, Mannor S. Graying the black box: Understanding DQNs[C]// International Conference on Machine Learning. 2016: 1899-1908.

[13] Rauber P E, Fadel S G, Falcao A X, et al. Visualizing the hidden activity of artificial neural networks[J]. IEEE Transactions on Visualization and Computer Graphics, 2017, 23(1): 101-110.

[14] Harley A W. An interactive node-link visualization of convolutional neural networks[C]// International Symposium on Visual Computing. 2015: 867-877.

[15] Wold S, Esbensen K, Geladi P. Principal component analysis[J]. Chemometrics and Intelligent Laboratory Systems, 1987, 2(1): 37 - 52.

[16] Maaten L v d, Hinton G. Visualizing data using t-SNE[J]. Journal of Machine Learning Research, 2008, 9: 2579-2605.

[17] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553): 436-444.

[18] Streeter M J, Ward M O, Alvarez S A. NVIS: an interactive visualization tool for neural networks [J]. SPIE, 2001, 4302: 234-241.

[19] Craven M W, Shavlik J W. Visualizing learning and computation in artificial neural networks [J]. International Journal on Artificial Intelligence Tools, 1992, 1(03): 399-425.

[20] Alsallakh B, Hanbury A, Hauser H, et al. Visual methods for analyzing probabilistic classification data[J]. IEEE Transactions on Visualization and Computer Graphics, 2014, 20(12): 1703-1712.

[21] Ren D, Amershi S, Lee B, et al. Squares: Supporting interactive performance analysis for multiclass classifiers[J]. IEEE Transactions on Visualization and Computer Graphics, 2017, 23 (1): 61-70.

[22] Orr G B, Müller K R. Neural networks: tricks of the trade[M]. Berlin: Springer, 2003

[23] Google. Tensorflow[EB/OL]. 2017. https://www.tensorflow.org.

[24] NVIDIA. NVIDIA DIGITS: Interactive deep learning GPU training system[EB/OL]. 2017. https://developer.nvidia.com/digits.

[25] Boykov Y Y, Jolly M P. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images[C]//IEEE International Conference on Computer Vision. 2001: 105-112.

[26] Rother C, Kolmogorov V, Blake A. Grabcut: Interactive foreground extraction using iterated graph cuts[J]. ACM Transactions on Graphics, 2004, 23(3): 309-314.

[27] Buckley C, Salton G. Optimization of relevance feedback weights[C]//ACM Conference on Research and Development in Information Retrieval. 1995: 351-357.

[28] Wang X, Liu S, Liu J, et al. TopicPanorama: A full picture of relevant topics[J]. IEEE Transactions on Visualization and Computer Graphics, 2016, 22(12): 2508-2521.

[29] Choo J, Lee C, Reddy C K, et al. UTOPIAN: User-driven topic modeling based on interactive nonnegative matrix factorization[J]. IEEE Transactions on Visualization and Computer Graphics, 2013, 19(12): 1992-2001.

[30] Paiva J G, Florian L, Pedrini H, et al. Improved similarity trees and their application to visual data classification[J]. IEEE Transactions on Visualization and Computer Graphics, 2011, 17 (12): 2459-2468.

[31] 巫英才, 崔为炜, 宋阳秋, 等. 基于主题的文本可视分析研究[J]. 计算机辅助设计与图形学学报, 2012, 24(10): 1266-1272.

[32] Lee D D, Seung H S. Learning the parts of objects by non-negative matrix factorization[J]. Nature, 1999, 401(6755): 788-791.

[33] Settles B. Synthesis lectures on artificial intelligence and machine learning: Active learning [M]. California: Morgan & Claypool, 2012

[34] Chen J, Zhu J, Wang Z, et al. Scalable inference for logistic-normal topic models[C]//Advances in Neural Information Processing Systems. 2013: 2445-2453.

[35] r. Mohamed A, Dahl G E, Hinton G. Acoustic modeling using deep belief networks[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2012, 20(1): 14-22.

[36] Seide F, Li G, Yu D. Conversational speech transcription using context-dependent deep neural networks[C]//Interspeech. 2011: 437-440.

[37] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in Neural Information Processing Systems. 2012: 1097-1105.

[38] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]//IEEE Conference on Computer Vision and Pattern Recognition. 2015: 1-9.

[39] Karpathy A, Toderici G, Shetty S, et al. Large-scale video classification with convolutional neural networks[C]//IEEE Conference on Computer Vision and Pattern Recognition. 2014: 1725-1732.

[40] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.

[41] Silver D, Huang A, Maddison C J, et al. Mastering the game of go with deep neural networks and tree search[J]. Nature, 2016, 529(7587): 484-489.

[42] Bengio Y. Learning deep architectures for AI[J]. Foundations and Trends in Machine Learning, 2009, 2(1): 1-127.

[43] Yosinski J, Clune J, Nguyen A, et al. Understanding neural networks through deep visualization [C]//ICML Workshop on Deep Learning. 2015.

[44] Bottou L. Stochastic gradient learning in neural networks[J]. Neuro-Nımes, 1991, 91(8).

[45] Marsland S. Machine learning: an algorithmic perspective[M]. Florida: CRC press, 2015

[46] Comaniciu D, Meer P. Mean shift: a robust approach toward feature space analysis[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(5): 603-619.

[47] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//IEEE Conference on Computer Vision and Pattern Recognition. 2014: 580-587.

[48] Mahendran A, Vedaldi A. Understanding deep image representations by inverting them[C]// IEEE Conference on Computer Vision and Pattern Recognition. 2015: 5188-5196.

[49] Li H, Jiang T, Zhang K. Efficient and robust feature extraction by maximum margin criterion [J]. Neural Networks, 2006, 17(1): 157-165.

[50] Huang E, Korf R E. Optimal rectangle packing: An absolute placement approach[J]. Journal of Artificial Intelligence Research, 2012, 46: 47-87.

[51] Korf R E, Moffitt M D, Pollack M E. Optimal rectangle packing[J]. Annals of Operations Research, 2010, 179(1): 261-295.

[52] Newman M E. Fast algorithm for detecting community structure in networks[J]. Physical review E, 2004, 69(6): 066133.

[53] Johnson B, Shneiderman B. Tree-maps: A space-filling approach to the visualization of hierarchical information structures[C]//Visualization. 1991: 284-291.

[54] Held M, Karp R M. A dynamic programming approach to sequencing problems[J]. Journal of the Society for Industrial and Applied Mathematics, 1962, 10(1): 196-210.

[55] Chen M, Jaenicke H. An information-theoretic framework for visualization[J]. IEEE Transactions on Visualization and Computer Graphics, 2010, 16(6): 1206-1215.

[56] Chen M, Walton S, Berger K, et al. Visual multiplexing[J]. Computer Graphics Forum, 2014, 33(3): 241-250.

[57] Sun M, Mi P, North C, et al. BiSet: Semantic edge bundling with biclusters for sensemaking [J]. IEEE Transactions on Visualization and Computer Graphics, 2016, 22(1): 310-319.

[58] Cui W, Zhou H, Qu H, et al. Geometry-based edge clustering for graph visualization[J]. IEEE Transactions on Visualization and Computer Graphics, 2008, 14(6): 1277-1284.

[59] Holten D, Van Wijk J J. Force-directed edge bundling for graph visualization[J]. Computer Graphics Forum, 2009, 28(3): 983-990.

[60] Agrawal R, Srikant R. Fast algorithms for mining association rules in large databases[C]// International Conference on Very Large Data Bases. 1994: 487-499.

[61] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition [J]. arXiv preprint arXiv:1409.1556, 2014.

[62] Nair V, Hinton G E. Rectified linear units improve restricted Boltzmann machines[C]// International Conference on Machine Learning. 2010: 807-814.

[63] Krizhevsky A. Learning multiple layers of features from tiny images[R/OL]. University of Montreal, 2009. http://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf.

[64] Jia Y, Shelhamer E, Donahue J, et al. Caffe: Convolutional architecture for fast feature embedding[J]. arXiv preprint arXiv:1408.5093, 2014.

[65] Sun S, Chen W, Wang L, et al. On the depth of deep neural networks: A theoretical view[C]// the Association for the Advance of Artificial Intelligence (AAAI). 2016: 2066-2072.

[66] Li C, Zhu J, Shi T, et al. Max-margin deep generative models[C]//Advances in Neural Information Processing Systems. 2015: 1828-1836.

[67] Li C, Zhu J, Zhang B. Learning to generate with memory[C]//International Conference on Machine Learning. 2016: 1177–-1186.

[68] Kingma D P, Mohamed S, Rezende D J, et al. Semi-supervised learning with deep generative models[C]//Advances in Neural Information Processing Systems. 2014: 3581-3589.

[69] Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks[J]. Science, 2006, 313(5786): 504-507.

[70] Zhu J, Chen J, Hu W, et al. Big learning with Bayesian methods[J]. National Science Review, 2017, 4(3): 1-25.

[71] Devroye L. Sample-based non-uniform random variate generation[C]//Conference on Winter Simulation. 1986: 260-265.

[72] Kingma D P, Welling M. Auto-encoding variational Bayes[J]. arXiv preprint arXiv:1312.6114, 2013.

[73] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[C]//Advances in Neural Information Processing Systems. 2014: 2672-2680.

[74] Arjovsky M, Chintala S, Bottou L. Wasserstein GAN[J]. arXiv preprint arXiv:1701.07875, 2017.

[75] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks[J]. arXiv preprint arXiv:1511.06434, 2015.

[76] Wang X, Liu S, Chen Y, et al. How ideas flow across multiple social groups[C]//IEEE Visual Analytics Science and Technology. 2016: 770-778.

[77] Cui W, Liu S, Tan L, et al. Textflow: Towards better understanding of evolving topics in text[J]. IEEE Transactions on Visualization and Computer Graphics, 2011, 17(12): 2412-2421.

[78] Han J, Pei J, Kamber M. Data mining: Concepts and techniques[M]. Amsterdam: Elsevier, 2011

[79] Bergstra J, Breuleux O, Bastien F, et al. Theano: a CPU and GPU math expression compiler [C]//SciPy. 2010.

[80] Tam G K L, Kothari V, Chen M. An analysis of machine- and human-analytics in classification [J]. IEEE Transactions on Visualization and Computer Graphics, 2017, 23(1): 71-80.

[81] Balzer M, Schlömer T, Deussen O. Capacity-constrained point distributions: A variant of Lloyd's method[J]. ACM Transactions on Graphics, 2009, 28(3): 86:1-86:8.

[82] Sun X, Zhou K, Guo J, et al. Line segment sampling with blue-noise properties.[J]. ACM Transactions on Graphics, 2013, 32(4): 127-1.

[83] Chen H, Chen W, Mei H, et al. Visual abstraction and exploration of multi-class scatterplots[J]. IEEE Transactions on Visualization and Computer Graphics, 2014, 20(12): 1683-1692.

[84] Rumelhart D E, Widrow B, Lehr M A. The basic ideas in neural networks[J]. Communications of the ACM, 1994, 37(3): 87-93.

[85] Lapuschkin S, Binder A, Montavon G, et al. The LRP toolbox for artificial neural networks[J]. Journal of Machine Learning Research, 2016, 17(114): 1-5.

[86] Bishop C M. Pattern recognition and machine learning[M]. Berlin: Springer, 2006

[87] Osborne M J. An introduction to game theory[M]. New York: Oxford University Pres, 2002

[88] Sutskever I, Martens J, Dahl G, et al. On the importance of initialization and momentum in deep learning[C]//International Conference on Machine Learning. 2013: 1139-1147.

[89] Kinga D, Adam J B. Adam: A method for stochastic optimization[C]//International Conference on Learning Representations. 2015.

[90] Bachman P. An architecture for deep, hierarchical generative models[C]//Advances in Neural Information Processing Systems. 2016: 4826-4834.

[91] Kingma D P, Salimans T, Welling M. Improving variational inference with inverse autoregressive flow[J]. arXiv preprint arXiv:1606.04934, 2016.

[92] Sønderby C K, Raiko T, Maaløe L, et al. Ladder variational autoencoders[C]//Advances in Neural Information Processing Systems. 2016: 3738-3746.

[93] Pu Y, Gan Z, Henao R, et al. Variational autoencoder for deep learning of images, labels and captions[C]//Advances in Neural Information Processing Systems. 2016: 2352-2360.

[94] Liu B, Zhang L. A survey of opinion mining and sentiment analysis[M]//Mining text data. Berlin: Springer, 2012: 415-463

[95] Ruiz E J, Hristidis V, Castillo C, et al. Correlating financial time series with micro-blogging activity[C]//ACM International Conference on Web Search and Data Mining. 2012: 513-522.

[96] Zhao X W, Guo Y, He Y, et al. We know what you want to buy: A demographic-based system for product recommendation on microblogs[C]//ACM International Conference on Knowledge Discovery and Data Mining. 2014: 1935-1944.

[97] 刘翠娟, 刘箴, 柴艳杰, 等. 基于微博文本数据分析的社会群体情感可视计算方法研究[J]. 北京大学学报 (自然科学版), 2016, 52(1): 178-186.

[98] 王臻皇, 陈思明, 袁晓如. 面向微博主题的可视分析研究[J]. 软件学报, 2018, 29(4): 1115-1130.

[99] Bosch H, Thom D, Heimerl F, et al. ScatterBlogs2: Real-time monitoring of microblog messages through user-guided filtering[J]. IEEE Transactions on Visualization and Computer Graphics, 2013, 19(12): 2022-2031.

[100] Efron M. Information search and retrieval in microblogs[J]. Journal of the American Society for Information Science and Technology, 2011, 62(6): 996-1008.

[101] Duan Y, Wei F, Chen Z, et al. Twitter topic summarization by ranking tweets using social influence and content quality[C]//Coling. 2012: 763-780.

[102] Wei F, Li W, Lu Q, et al. Query-sensitive mutual reinforcement chain and its application in query-oriented multi-document summarization[C]//ACM Conference on Research and Development in Information Retrieval. 2008: 283-290.

[103] Brin S, Page L. The anatomy of a large-scale hypertextual web search engine[J]. Computer networks and ISDN systems, 1998, 30(1): 107-117.

[104] BIPM I, IFcc I, IuPAc I. Oiml, guide to the expression of uncertainty in measurement[J]. International Organization for Standardization, Geneva. ISBN, 1995: 92-67.

[105] Correa C, Chan Y H, Ma K L. A framework for uncertainty-aware visual analytics[C]//IEEE Visual Analytics Science and Technology. 2009: 51-58.

[106] Wu Y, Yuan G X, Ma K L. Visualizing flow of uncertainty through analytical processes[J]. IEEE Transactions on Visualization and Computer Graphics, 2012, 18(12): 2526-2535.

[107] Bianchini M, Gori M, Scarselli F. Inside PageRank[J]. ACM Transactions on Internet Technology, 2005, 5(1): 92-128.

[108] Liu X, Song Y, Liu S, et al. Automatic taxonomy construction from keywords[C]//ACM International Conference on Knowledge Discovery and Data Mining. 2012: 1433-1441.

[109] Liu S, Wang X, Chen J, et al. TopicPanorama: A full picture of relevant topics[C]//IEEE Visual Analytics Science and Technolog. 2014: 183-192.

[110] Kamada T, Kawai S. An algorithm for drawing general undirected graphs[J]. Information Processing Letters, 1989, 31(1): 7-15.

[111] Lampe O D, Hauser H. Interactive visualization of streaming data with kernel density estimation [C]//IEEE Pacific Visualization Symposium. 2011: 171-178.

[112] Shi L, Wei F, Liu S, et al. Understanding text corpora with multiple facets[C]//IEEE Visual Analytics Science and Technolog. 2010: 99-106.

[113] Ghani S, Kwon B, Lee S, et al. Visual analytics for multimodal social network analysis: A design study with social scientists[J]. IEEE Transactions on Visualization and Computer Graphics, 2013, 19(12): 2032-2041.

[114] Phan D, Xiao L, Yeh R, et al. Flow map layout[C]//IEEE Information Visualization. 2005: 219-224.

[115] Verbeek K, Buchin K, Speckmann B. Flow map layout via spiral trees[J]. IEEE Transactions on Visualization and Computer Graphics, 2011, 17(12): 2536-2544.

[116] Holten D, Van Wijk J J. Force-directed edge bundling for graph visualization[J]. Computer Graphics Forum, 2009, 28(3): 983-990.

[117] Bahmani B, Chowdhury A, Goel A. Fast incremental and personalized pagerank[J]. International Conference on Very Large Data Bases, 2010, 4(3): 173-184.

[118] Chandramouli A, Gauch S. A co-operative web services paradigm for supporting crawlers[C]// Large Scale Semantic Access to Content (Text, Image, Video, and Sound). 2007: 475-489.

[119] Moosavi-Dezfooli S M, Fawzi A, Fawzi O, et al. Universal adversarial perturbations[C]//IEEE Conference on Computer Vision and Pattern Recognition. 2017: 86-94.

[120] Liu S, Wang X, Liu M, et al. Towards better analysis of machine learning models: A visual analytics perspective[J]. Visual Informatics, 2017, 1(1): 48-56.

[121] Strobelt H, Gehrmann S, Pfister H, et al. LSTMVis: A tool for visual analysis of hidden state dynamics in recurrent neural networks[J]. IEEE Transactions on Visualization and Computer Graphics, 2018, 24(1): 667-676.

[122] Ming Y, Cao S, Zhang R, et al. Understanding hidden memories of recurrent neural networks [C]//IEEE Visual Analytics Science and Technology. 2017.

[123] Wang J, Gou L, Yang H, et al. Ganviz: A visual analytics approach to understand the adversarial game[J]. IEEE Transactions on Visualization and Computer Graphics, 2018, 24(6): 1905-1917.

致 谢

五年多的博士生活是我人生中一段重要的时光。在博士学习生活即将结束的时候，回首过去那个连代码都不太会写的自己，感慨良多。

首先要衷心感谢导师沈向洋教授和张钹教授对我的精心指导。他们的谆谆教诲将使我受益终生。沈向洋教授以其不知疲倦的科研态度鼓舞着我在二年级最迷茫的时候砥砺前行。张钹教授以其自身严谨的治学之道，为我树立了一生学习的典范，也鼓励着我在今后的科研道路上研究求是。

特别感谢这几年一直悉心指导我的刘世霞教授。在跟刘老师学习的这几年中，刘世霞教授严谨求实的科研态度，卓越的科研能力都让我受益良多。在刘老师的指导下，我从当年那个不靠谱的学生，变得稍微靠谱了一些。可以说，没有她的指导，我不可能完成博士学业。在未来的日子里，也会秉承刘老师一贯的指导，做一名靠谱的研究员。

感谢高等研究院，微软亚洲研究院，计算机系，软件学院为我们创造了良好的学术氛围。高等研究院时刻以其学院派的气质熏陶着我。感谢吴念乐老师关怀，让我虽然身不在高研，也能体会到学院的温暖。感谢李丽老师细致的工作，能让我们这一批学生顺利地完成学业。在微软亚洲研究院的两年中，我深刻地感到做学问只要有心，无论是身在高校还是身在企业都是没有问题的。感谢计算机系和软件学院提供了优良的工作环境。

感谢和我合作过的各位老师同学。感谢王希廷学姐一直以来的帮助。学姐阳光向上的心态一直是我学习的榜样。感谢朱锡洲学弟，我惊叹于学弟的科研能力，敏捷的思路和坚实的基础。感谢李振，和他合作的一年时间是我最开心的一段时间。愿我们的友谊永不退色。感谢曹科垒，作为一名靠谱的学弟，让我的科研之路顺利很多。也感谢实验室的师弟师妹们，是他们的陪伴让我开心地度过了这五年博士学习时光。

感谢我的家人。虽然我已经长大在外已久，但是我知道任何时间我回去，那里一定有一个温暖的家。没有家人的默默支持，将没有我的今天希望家人能够一直平安喜乐。

最后，谨以此文献给我的夫人。是她让我一直不断努力，做一个更好的自己。

声　明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签　名：_____ 日　期：_____

个人简历、在学期间发表的学术论文与研究成果

个人简历

1990 年 3 月 14 日出生于辽宁省大连市。

2009 年 9 月考入清华大学电子工程系电子信息科学与技术专业，2013 年 7 月本科毕业并获得工学学士学位。

2013 年 9 月免试进入清华大学高等研究院攻读工学博士学位至今。

发表的学术论文

[1] Liu M, Shi J, Cao K, et al. Analyzing the training processes of deep generative models[J]. IEEE Transactions on Visualization and Computer Graphics, 2018, 24(1): 77-87. (SCI 收录, 检索号:000418038400010, 影响因子:2.84).

[2] Liu M, Shi J, Li Z, et al. Towards better analysis of deep convolutional neural networks[J]. IEEE Transactions on Visualization and Computer Graphics, 2017, 23(1): 91-100. (SCI 收录, 检索号:000395537600012, 影响因子:2.84).

[3] Liu M, Jiang L, Liu J, et al. Improving learning-from-crowds through expert validation[C] //International Joint Conference on Artificial Intelligence. 2017: 2329-2336.

[4] Liu M, Liu S, Zhu X, et al. An uncertainty-aware approach for exploratory microblog retrieval[J]. IEEE Transactions on Visualization and Computer Graphics, 2016, 22(1): 250-259. (SCI 收录, 检索号:000364043400030, 影响因子:2.84).

[5] Liu S, Wang X, Liu M, et al. Towards better analysis of machine learning models: A visual analytics perspective[J]. Visual Informatics, 2017, 1(1): 48-56.

[6] Peng T Q, Liu M, Wu Y, et al. Follower-followee network, communication networks, and vote agreement of the US members of congress[J]. Communication Research, 2016, 43(7): 996-1024.

[7] Meng Y, Zhang H, Liu M, et al. Clutter-aware label layout[C]//IEEE PacificVis. 2015: 207-214.

[8] Wu Y, Liu S, Yan K, et al. Opinionflow: Visual analysis of opinion diffusion on social media[J]. IEEE Transactions on Visualization and Computer Graphics, 2014, 20(12): 1763-1772.

[9] Liu S, Cui W, Wu Y, et al. A survey on information visualization: recent advances

and challenges[J]. The Visual Computer, 2014, 30(12): 1373-1393.

[10] Liu S, Wu Y, Wei E, et al. Storyflow: Tracking the evolution of stories[J]. IEEE Transactions on Visualization and Computer Graphics, 2013, 19(12): 2436-2445.