分类号_____ 密级_____

U D C_____ 编号_____

中国科学院自动化研究所

博 士 后 研 究 工 作 报 告

人工神经网络中连续学习与情境学习的算法设计与研究

陈 阳

工作完成日期 2017 年 7 月 — 2019 年 3 月

报告提交日期 2019 年 3 月

中国科学院自动化研究所

2019 年 3 月

人工神经网络中连续学习与情境学习的算法设计与研究

Continual Learning of Context-dependent Processing in Neural Networks

博 士 后 姓 名　　陈阳

流动站（一级学科）名称　控制科学与工程

专　业（二级学科）名称　模式识别与智能系统

研究工作起始时间　2017 年 7 月 3 日

研究工作期满时间　2019 年 4 月 3 日

中国科学院自动化研究所

2019 年 3 月

摘 要

深度人工神经网络（deep neural networks, DNN）已经在识别任务和分类任务中展示了其强大威力，可以学习输入和输出之间十分复杂的映射关系。然而，一方面，当前的人工神经网络不具备连续学习的能力。所有任务要在训练过程中同时完成。如果此后要学习新的任务，就会将以往的训练成果“灾难性”遗忘。另一方面，当前大多数任务中 DNN 所学习的规则是固定的，一旦学习完毕就不再改变。这些不足将大大限制人工神经网络在复杂多变的环境下工作的能力。因为在实际情况下，输出不仅仅由输入决定，其映射规则会根据情境的不同（例如不同的环境和目标）而发生变化。这不仅需要智能体要具有连续学习的能力，在积累经验的同时解决新的问题，还需要根据不同的情况灵活运用这些知识。

针对这些问题，我们提出了一种新的连续学习算法——正交权重修改（orthogonal weights modification，OWM）算法，并受大脑前额叶的启发，在 OWM 算法的基础上提出了情境信息处理（context-dependent processing， CDP）模块，能够使人工神经网络获得连续学习和情境学习的能力。通过数值实验我们证明，OWM 算法可以有效保护网络已获得的知识，连续学习多达数千种不同的新任务而不会干扰已学的知识结构。同时，学习过程只需要少量样本，展示了强大的在线、连续学习的能力。此外，通过使用 CDP 模块，神经网络可以根据情境信息调制输入特征的表示，进而辅助网络学习上千种任务，特别是可以对相同输入作出不同的响应。总之，这些算法相结合可以使人工神经网络更加灵活的应对和适应复杂环境，使其在保证结构紧凑的同时，逐渐学习现实世界不断出现的新规律、新任务。

关键词：类脑智能，连续学习，情境学习，正交权重修改算法，情境信息处理模块

Abstract

Deep artificial neural networks (DNNs) are powerful tools for recognition and classification as they learn sophisticated mapping rules between the inputs and the outputs. However, the rules that learned by the majority of current DNNs used for pattern recognition are largely fixed and do not vary with different conditions. This limits the network’s ability to work in more complex and dynamical situations in which the mapping rules themselves are not fixed but constantly change according to contexts, such as different environments and goals.

Inspired by the role of the prefrontal cortex (PFC) in mediating context-dependent processing in the primate brain, here we propose a novel approach, involving a learning algorithm named orthogonal weights modification (OWM) with the addition of a context-dependent processing (CDP) module, that enables networks to continually learn different mapping rules in a context-dependent way. We demonstrate that with OWM to protect previously acquired knowledge, the networks could sequentially learn up to thousands of different mapping rules without interference, and needing as few as samples to learn each, demonstrating powerful ability of online, continual learning. In addition, by using the CDP module to enable contextual information to modulate the representation of sensory features, a network could sequentially learn different, context-specific mappings for even identical stimuli. Taken together, these approaches allow us to teach a single network numerous context-dependent mapping rules in an online, continual manner. This would enable highly compact systems to gradually learn myriad of regularities of the real world and eventually behave appropriately within it.

Key Words: Brain-like intelligence, Continual learning, Context-dependent learning, orthogonal weights modification (OWM), **context-dependent processing** (CDP)

目录

第1章.绪论

1. 研究背景与意义

人们对人工智能的向往由来已久。在早期，人工智能多出现在文学之中，是作家想象的产物，如著名的科学怪人弗兰肯斯坦。直到上个世纪 50 年代左右，Rosenblatt 在 MP 神经元数学模型的基础上，提出了第一代神经网络单层感知机[1,2]。至此，人工智能进入科学工作者的研究视野。单层神经感知机的提出也掀起了人工智能的第一次高潮。然而在 1969 年，Minsky 证明单层感知器无法解决异或问题[3]，导致这一次转入低谷。不过，这一次的发展还是吸引了科学领域的广大关注，也为之后的研究打下基础。1986 年，Hinton 等人提出多层感知机神经网络[4]，并利用误差的反向传播算法来训练模型，有效的解决了之前非线性分类难题。紧接着后来 Cybenko 和 Hornik 等人证明任何函数都可以被三层神经网络以任意精度逼近[5,6]，掀起了人工智能研究的第二波高潮。但好景不长，人们发现误差反传算法存在梯度消散问题，加之当时数据、算力等条件的限制，多层神经网络被认为难以实用，人们的注意力转移到了支持向量机等浅层学习模型上。到了 2010 年左右，大数据已经成为现实，而算力也因硬件的高速发展而迅速提升。同时，人们也提出各种方法抑制梯度消失问题。深度网络在语音识别和图片分类问题上先后取得重大突破[7-10]，取得了远超以往模型的识别和分类效果，从此深度学习进入爆发期。从 2010 年至今是人工智能高速发展的 10 年，新成果层出不穷，工业应用如火如荼。然而，大家也认识到目前的人工智能距真正的智

能还有相当的距离。自然，人们将目光投向了人工智能的天然参照物——大脑。

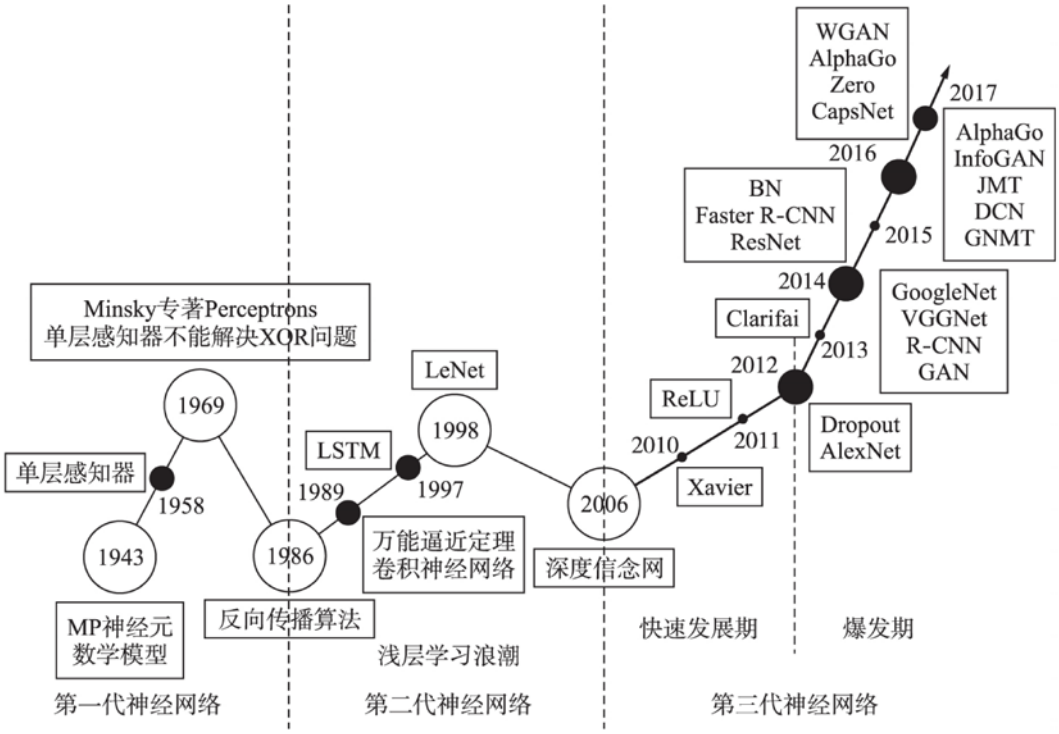

图 1-1 深度学习发展史[11]

过去的十年也是脑科学高速发展的十年。我们的大脑包含高达 10^{11}-10^{12} 个神经元。这些神经元通过约 10^{15} 量级的突触联系在一起，构成了这个星球上结构最复杂和功能最丰富的网络之一。一直以来，人们在不同层次和不同角度对大脑的功能和机理不断探索。在早期，脑研究主要集中于认知心理学方面的讨论，而随着科学技术的进步，大脑功能的生物基础和机制逐渐成为了关注的焦点。近 10 年来，对大脑的研究更是受到了前所未有的重视，美国、欧盟、日本、加拿大、澳大利亚、韩国等国相继提出了各自版本的脑研究计划[12]。这些脑计划从不同的方面增进人们对大脑工作机理的了解，典型如：欧盟的 “人类脑计划”（Human Brain Project）[13]，美国推出的“推进创新神经技术脑研究计划

(BRAIN)”[14]，等等。中国的脑计划[15]也在酝酿之中，提出了“一体两翼”的研究思路，而其中一翼就是类脑人工智能研究，强调脑研究与人工智能的相互启发和相互促进。第三次人工智能热潮方兴未艾，大脑可以为通用人工智能提供一个绝佳的参照，而人工智能的发展也可以为脑研究提供新的工具或思路。因此，脑科学与人工智能之间的相互借鉴与相互促进，对二者都有积极的意义。

2. 研究内容概述

类脑智能在近来越来越受到人工智能领域的重视。一方面，大脑是人工智能的绝佳参照。大脑在彰显当前人工神经网络不足的同时，也为人工神经网络的改进和发展指明了方向。另一方面，脑科学的发现可以启发人工智能的新思路。通过对大脑神经机制的借鉴，我们可以进一步改进现有算法。实际上，人工智能对脑科学的借鉴由来已久。例如，MP 神经元和单层神经感知机最早提出来是为了解释大脑的信息存储和处理机制的，而后来成为了人工智能连接主义的基石。

本人在博士后期间主要开展了类脑智能方面的研究。和大脑相比，连续学习和情境学习能力的缺失是当前人工神经网络的重大不足。针对这个问题，我们首先提出了正交权重修改（orthogonal weights modification，简称 OWM）算法，实现了人工神经网络的连续学习。在此基础之上，我们受大脑前额叶皮层的启发，提出了情境信息处理（context-dependent processing ，简称 CDP）模块，实现人工神经网络的情境学习。我们在不同的数据集、网络构架、任务类型上测试对比，进一步证明了我们算法的有效性。本研究将为实现人工神经网络的连续学习和情境学习提供崭新的解决方案。而这两种能力的具备将为我们开发更加类脑的人工神经网络构架和算法打下基础。

3. 报告组织结构

本文分为六个章节，每章的内容如下：

第一章：绪论。本章总述报告的研究内容，选题的背景和意义，并给出文章的章节安排。

第二章：研究现状简介。本章介绍连续学习和情境学习的研究现状，包括神经科学的研究发现和人工神经网络的已有方案。

第三章：OWM 算法的原理。本章详细介绍 OWM 算法实现连续学习的基本原理，给出算法的具体执行流程，并将分析人工智能领域内的其他算法的区别与联系。最后我们将讨论该算法在大脑中的生物可实现性。

第四章：OWM 算法的验证和对比。本章介绍 OWM 算法的实际测试效果，并与主流方法进行比较。我们首先在基于 MNIST 数据集的标准连续学习任务对 OWM 测试，并将结果与其他算法的结果比较，证明该算法的优越性。然后我们将在大数据集上进一步研究 OWM 算法的有效性 。最后本章将总结 OWM 算法的性能特点，并讨论可能的改进方向。

第五章：基于 OWM 算法的 CDP 模块。本章将介绍 CDP 模块的构建以及实现情境学习的原理。并以一个人脸属性识别任务为例，介绍该模块如何配合 OWM 算法实现人工神经网络的情境学习

第六章：总结与展望。本章总结本报告研究内容，并分析和展望未来可能的研究方向。

第2章.研究现状简介

1. 连续学习的研究现状

连续学习是大脑的一项非常重要的能力。所谓连续学习，是指系统在不忘记已获得知识或技能的前提下，学习新的知识或技能的能力[16]。人和动物在其一生中，会不断遇到新的境遇。这意味着大脑需要不断学习新的规则应对。在学习新知识的过程中，大脑还要保证不忘记旧的知识，正如我们学会了游泳但不能忘记如何走路。因此，连续学习的能力是人和动物不断适应环境乃至成长进化的关键之一。因此，连续学习能力对于智能体在与复杂的现实环境交互、处理不断变化的信息流时至关重要。在人工智能领域，人们也已经认识到连续学习能力是获得通用人工智能的必要条件之一。然而，连续学习对于机器学习和人工神经网络模型仍然是一个长期的挑战。当今主流的人工神经网络模型是基于连接主义的，即其全部信息都存储在权重之中。当神经网络在面对全新的数据和任务时，会不可避免的对权重修改、甚至覆盖，进而导致以往存储的信息被擦除。因此，当该人工神经网络面对以前已经学习过的任务时，其性能对大大下降，即发生“灾难性遗忘”（catastrophic forgetting，请见图 2-1）。灾难性遗忘是当前深层神经网络模型的一个重要缺点，也是大家关注的热点和难点之一。

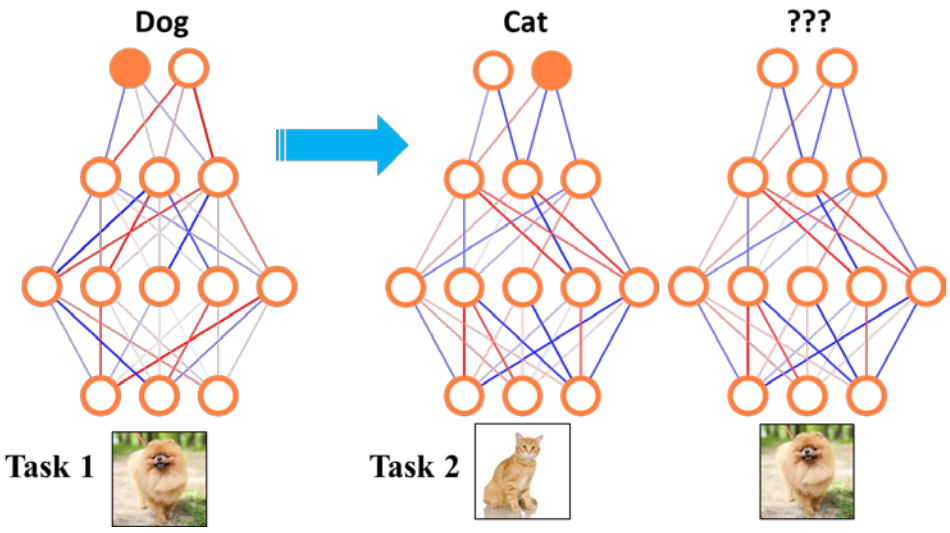

图 2-1 灾难性遗忘示意图

2. 神经科学对连续学习的讨论

人类在一生能够不断的学习知识和技能，在新的境遇下提炼新的经验，并实现跨领域的知识迁移，因而表现出惊人的适应能力[17]。虽然一生中，我们往往也会逐渐忘记过往的一些细节信息，但在学习新信息的时候，却很少对已掌握知识造成重大干扰[18]。例如，体感皮层可以在运动学习任务中吸收新信息，而不会破坏已获得的运动技能的稳定性[19]。显然，大脑已经找到一套有效的神经生理机制实现终身学习。这套机制的核心在于神经元权重可塑性和稳定性之间的微妙平衡[20-23]。可塑性体现为神经元为学习新任务对权重做出必要的修改；而稳定性则要求这些修改不能造成神经系统在旧任务上的性能损失。二者体现出对神经元可塑性相反的诉求，因此也被称为稳定性-可塑性悖论（stability－plasticity dilemma）[24]。

神经突触可塑性是大脑的一个基本特征。由于这种能力，大脑可以不断调整其物理结构，使我们能够不断学习、记忆和适应复杂多变的外部环境[21]。大脑

在早期发育时期尤其具有可塑性，之后通过一系列特定的发育阶段稳定下来，可塑性变得不那么突出，但保持了一定程度的可塑性，以便小范围内重组结构以适应环境[21,25 - 27]。神经科学已经发现，树突棘的动态变化和学习记忆在突触水平上的机制紧密相关[28]。树突棘是神经元树突质膜上形成的微小膜状突起，是兴奋性突触信号的主要接收位点。树突棘的形成/增大经常预示着突触联接的增强，而树突棘的缩小/ 消失通常意味着突触联接的消失[27]。实验工作表明，小鼠在学习新任务（技能）时，会生成新的树突棘，并只有其中一部分可以之后长时间存留下来。这说明在新任务中，神经系统会在已有结构的基础上对突触做出特异性的修改[28]。然而，神经元具体如何“特异性”的修改权重才能同时满足稳定性和可塑性的要求？这样的修改机制又是通过什么样的机制实现？这些问题尚缺少系统的研究

此外，还有部分神经科学家认为大脑存在可以长期和短期学习记忆的系统，二者相互补充来实现连续学习。互补学习系统（CLS）理论[29]（McClelland 等，1995）认为海马体可以快速学习并对新信息稀疏编码以减少其他信息干扰。相反，新皮层学习速度慢，并允许不同知识的共享权重。因此，海马和新皮质可以相互补充。海马系统具有短期适应性，快速学习新信息，然后将信息会不断回放从而将这些知识转移到新皮质中，实现新皮质的连续学习。

3. 人工神经网络现有算法简介

在人工智能领域，人们也已经认识 “灾难性遗忘”（catastrophic forgetting）问题的严重性。最近几年来，为了克服灾难性遗忘，人们已经提出了诸多算法[16]。在人工神经网络上，这些方法可以不同程度上实现连续学习。这些算法大概可以分为三类：一、增加结构，即学习新任务时以特定方式增加新的神经元，扩展原有的神经网络；二、保存样本，即已某种方式保存旧任务的数据样本，典

型的如利用存储模块保存关键数据或采用生成网络（generative networks）生成旧任务的样本；三、权重约束，即对权重的更新过程加以约束来实现连续学习。

3.1. 增加结构类算法

这类方法通过动态地引入新的神经结构单元（例如，神经元或网络层）改变网络构架来容纳新信息[30-36]。其中，最典型的是 Rusu 等人提出 Progressive Neural Networks[30]。该方法不对先前训练的网络做任何改变，只通过分配固定容量的新子网络来扩展架构来学习新的知识。该方法在训练过程会保留了已训练的模型。当有新任务时，则创建新的神经网络进行训练。不过新的子网络会与已训练的网络有连接，会接受旧模型传来的信息。由于旧任务中学习到的结构得到完全的保留，自然好避免灾难性遗忘。不过，这种方法虽然可以防止灾难性遗忘发生，但架构的复杂性随着学习任务的数量增多而不断增长，因此实际较难实现。

3.2. 保存样本类算法

CLS 理论[29]为这类算法提供了基础。因此这类算法通常会有两个子系统：一个负责完成任务，一个负责“记忆”所有任务内容。一旦遇到新任务，就记忆系统提取旧任务的训练样本与新样本混合，以训练任务系统，达到连续学习的目的[37-40]。例如，Shin 等人提出了一种由深度生成网络和任务求解器网络构成的双体系神经网络结构[38]。在新任务中，生成网络生成伪数据对先前学习任务的训练数据进行采样，并且与新任务的样本信息混合产生一个大的样本集，来重新训练整个网络。另一类则直接保存旧任务中特定样本，如 Sylvestre-Alvise Rebuffi 等人提出的 iCaRL 模型[37]，在学习过程过程会存储以往任务的关键样本子集。实际上，该类方法在每个新任务中都要把所有的任务训练一次，随着任务的增多，

会带来一定的效率问题。

3.3. 权重约束类算法

弹性权重巩固 (Elastic Weight Consolidation，EWC)[41] 。EWC 算法由 Deepmind & Google 的研究组提出，是这类算法中经典的代表之一。该算法的基本思想是用 Fisher 矩阵衡量历史任务中神经网络权重的重要性，并在新任务，降低那些重要性较高权重的学习率。显然，该方法可以保护神经网络已学习的知识。但是，该方法并不能从理论上并保证连续学习一定可以达成。原因在于对历史任务不重要的权重会因为在新任务中比较重要而变大，从而会对历史任务产生重要影响。若任务的输入空间重叠较小，该方法会有比较好的表现；反之则很难发挥作用。类似的还有突触智能(Synaptic Intelligence，SI)[42]方法，差别在于衡量权重重要性的矩阵不同。

增量矩匹配，即(Incremental Moment Matching，IMM)[43]。IMM 方法从另外一个角度考虑问题：如何融合新旧任务的最优网络结构，从而找到一个平衡点。该方法发现一个操作起来十分简单的办法，对新旧网络的最优网络结构加权平均作为最终的结果。这个过程被作者们称为增量匹配，即用新任务的最优解作为约束条件对原网络结构进行修改。作者提出不同的技巧来实现这一过程。该方法在一些简单的任务上表现良好，但是否具有可扩展性尚未知。

基于任务的硬注意力方法(Hard Attention to the Task，HAT)[44]。HAT 方法在任务训练时会计算当前任务下神经网络的注意力矩阵以及包括当前和历史任务信息的累计注意力矩阵。前者主要控制信息在网络中的流向，相当于对于特定任务选择相应的子结构；后者帮助权重更新，利用注意力信息决定哪些权重对历史任务重要，从而减小其学习率。该方法的一个缺陷是需要根据不同的任务选择不

同的注意力矩阵，即需要提前知道任务标签，而在实际当中这往往比较难办到。

Conceptor 辅助反传算法 (Conceptor-aided Backprop， CAB)[45]。作者提出 Conceptor 矩阵辅助神经网络的更新。该方法和本报告的 OWM 方法虽然相互独立提出，但是思想非常类似。我们会在后边详细讨论这两种方法的异同。

Learning without Forgetting（Li Z，2016ECCV）[46]。作者巧妙运用了知识蒸馏(Knowledge Distillation)[47]技术来缓解灾难性问题。该方法的主要思想是在新任务中先利用新数据在旧任务的网络上的输出结果作为伪标签，然后训练网络时将这些伪标签作为约束加入到 Loss 函数中，尽可能在训练过程中保持旧任务输出端的结果与伪标签接近。该方法希望以此保持神经网络在旧任务输入空间上的函数流形保持不变。然而新任务的输入空间不能保证对旧任务空间的全覆盖，因此很难取得很好的保护效果。

在上述三类算法中，第一类算法、第二类算法在新任务中不断增加神经元或存储数据，而实际中智能体会不断遇到新情况，需要学习的任务不可穷尽，故较难实用。第三类具有较强的实用价值，关键在于如何充分利用神经网络的容量，并达到新旧任务性能的平衡。

4. 情境学习的研究现状

情境学习能力，即灵活动态响应环境的能力，是高级智能的标志之一[48]。人类可以根据不同的目标、环境和内部状态等条件对相同的刺激做出不同的反应[49–53]。前额叶皮层（prefrontal cortex，PFC）在灵长类动物大脑中高度精细的结构，是实现这种能力的关键生物基础[53–57]。 PFC 可以快速学习输入到输出的映射规则，并可以根据环境的不同，灵活的用不同的动作输出响应感官输入 [56–58]

（参见图 2-2）。该过程在神经和认知科学领域被称之为认知控制。该过程允许灵长类动物灵活应对不确定、不可穷举的环境变化 [56,57,57]。在人类的认知实验中，若受试者 PFC 受损，其对输入的响应很大程度上取决于感觉输入刺激的强度，失去了对任务高度相关但刺激强度较弱信号的反应能力[59]。此外，这些受试者往往顽固地遵循已学到的既定规则，即便这时候情况已发生变化，该规则不再带来原本预期的结果。这即意味着他/她们失去根据情境动态调整输入和输出之间映射的能力[60]。不仅 PFC 受损的人类患者的实验证明 PFC 是情境学习的关键，非人类灵长类动物的许多电生理学研究也证明了 PFC 神经元确实可以表征各种与情境相关的信息[56,57]。

显然，情境学习能力是实现通用人工智能的必备能力之一。虽然神经科学对情境学习的研究由来已久，然而，在人工智能领域，大家对情境学习的关注远不及连续学习。可能是当下人工智能领域涉及的任务往往比较类型化的缘故。

当前的人工深度神经网络（deep neural networks，DNN） 尚缺少这样的能力，更缺少像 PFC 这样的功能模块。 DNN 非常擅长从原始传感数据中提取高级特征并进行模式检测、识别和分类等任务，在复杂映射规则学习方面非常强大[61]。然而，在大多数人工神经网络中，其输出响应完全由输入决定，表现出刻板的输入输出映射。而且这些映射一旦训练完成就不再改变。因此，当前的 DNN 缺乏足够的灵活性来应对复杂环境变化。这种灵活性表现为两个方面：1）映射规则可根据情境信息而改变; 2）这些规则经过少量训练就可以获得。这些能力的缺乏构成了 DNN 与人类大脑之间显着的差距。

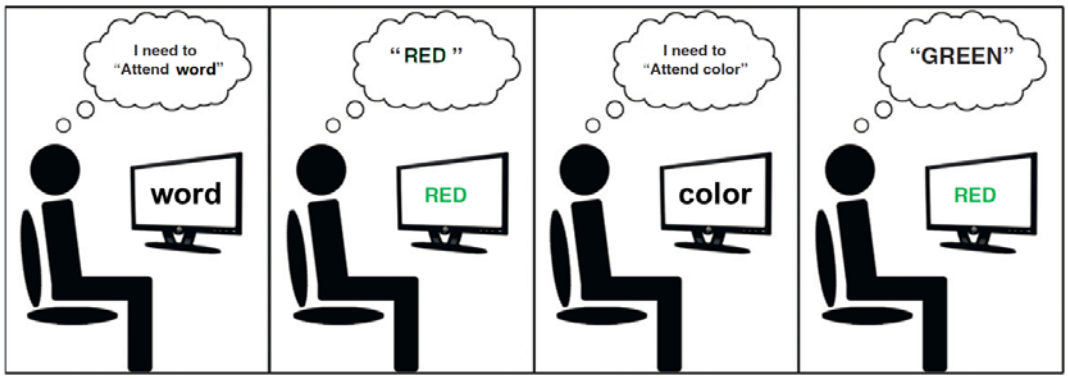

图 2-2 情境学习示意图。人可以根据提示，对同一个单词作出不同的相应（回答）。

5. 本章小结

连续学习和情境学习是本报告的研究核心。本章综述了这两个课题的研究现状。其中连续学习近几年已成为人工智能领域的热点之一，已经有了较多的成果。但是情境学习在神经科学领域一直是重点之一，但在人工神经网络上如何实现还有待进一步深入的研究。后面几章，我们将提出新的方案有效解决这两个问题，为实现更加类脑的人工神经网络打下基础。

第3章.OWM 算法的原理与内容

灾难性遗忘往往发生在连续学习任务中。目前，在人工智能领域，人们在训练人工神经网络时，会将所有任务的数据样本混合后同时“投喂”给系统。而对于人来说，我们学习知识时通常是学完一项再进行下一项。这样顺序的学习方式对人工神经网络是个困难。因为当前的人工神经网络是基于连接主义的，在训练时，若不特别处理的话会导致灾难性遗忘的发生。那么我们是否可以无视连续学习这种训练方式呢？答案是不能。在实际当中，要处理的问题的往往不可预知，必须不断学习才能有效应对各种新情况。

总之，连续学习是大脑的核心能力之一，也是实现通用人工智能的关键之一。到目前为止，现有的连续学习算法的效果并不令人满意，也不能从理论上完全保证可以克服灾难性遗忘。因此，我们提出一种全新的算法——正交权重修改算法（orthogonal weights modification，OWM），理论上可以完全克服灾难性遗忘，实现连续学习。

1. 基本思想

针对人工神经网络的灾难性遗忘问题，我们提出了正交权重修改(Orthogonal Weights Modification，OWM)算法。其基本思想是，人工神经网络在新任务连续训练过程中，其权重只允许在与旧任务输入空间正交的方向上修改。所谓旧任务的输入空间，是指以往所有训练任务中神经网络各层的所有输入向量张成的空间。对于神经网络的每一层，都有这样的一个空间。这样的操作确保了新任务的

权重更新不会干扰已经学到的任务。因为正交性意味着整个网络的权重增量几乎不会与旧的输入发生相互作用，因此在输入旧任务数据时，即便网络结构已发生变化，其输出几乎保存不变。在 OWM 中，上述正交操作是用正交投影矩阵实现的。其定义为[62–64]

$$\mathbf{P} = \mathbf{I} - \mathbf{A}\left(\mathbf{A}^{\mathrm{T}}\mathbf{A} + \alpha\mathbf{I}\right)^{-1}\mathbf{A} \qquad \text{式 3-1}$$

其中，$\mathbf{A} = [\mathbf{x}_1, \cdots, \mathbf{x}_n]$为输入的特征向量，$\mathbf{I}$是单位矩阵，常量$\alpha$用于调整投影的正交程度，若特征向量个数大于其维数还可以保证括号内矩阵可逆。得到正交投影矩阵后，我们将其作用于传统算法得到的权重增量，得到最终权重更新量$\Delta\mathbf{W}^{\mathrm{OWM}}$：

$$\Delta\mathbf{W}^{\mathrm{OWM}} = \kappa\mathbf{P}\Delta\mathbf{W}^{\mathrm{BP}} \qquad \text{式 3-2}$$

式中κ是学习率，$\Delta\mathbf{W}^{\mathrm{BP}}$是反向传播(BackPropogation，BP)[4]算法计算得到的权重增量。图 3-1（A）较形象的展示了这一过程，在权重空间中，该算法首先计算传统反向传播算法得到的权重增量$\Delta\mathbf{W}^{\mathrm{BP}}$，然后与投影矩阵$\mathbf{P}$作用，只保留与历史任务输入空间正交的部分，即实际权重增量$\Delta\mathbf{W}^{\mathrm{OWM}}$。如此，可以将新任务解的搜索范围约束在旧任务的解空间中 (图 3-1（A）中的深绿色曲面以及（B）的蓝色区域)。这确保了系统的权重最终会在所有任务解空间的重叠区域内，保证在不忘记旧任务的同时完成新任务，实现连续学习。

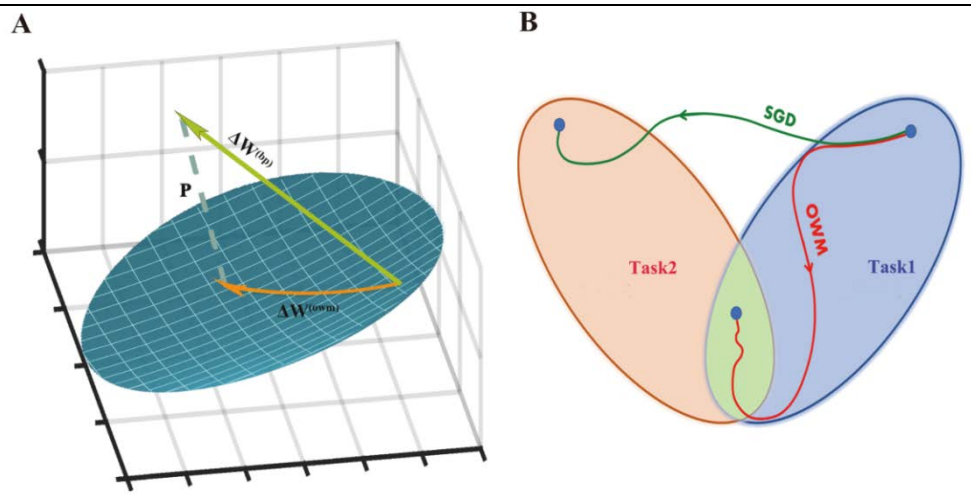

图 3-1　OWM 算法基本原理示意图。A. 正交投影矩阵对权重增量的投影过程。B. 使用 OWM 算法，可以保证网络在训练过程过中搜索可任务 2 的解空间时（桔黄色区域），不会离开任务 1（淡蓝色区域）的解空间。当成功搜索到两个空间的重叠之处（浅绿色区域）时，算法可以停止（红色曲线代表搜索路径）。相比之下，随机梯度下降(SGD)搜索获得的解，往往不能保证其处于两个任务解空间的重合区域之中（绿色曲线）。

2. 算法内容

这一节我们将具体介绍 OWM 的具体实现流程。从上一节可以看到 OWM 算法与 BP 算法的主要区别在于引入了正交投影算子。因此如何计算该算子是内容的重点。按照定义（式 3-1）计算投影算子是十分不便的，因为需要所有历史任务的输入数据。实际上，正交投影算子由于有正交这个优良的性质，是可以通过递归的方式更新，避免了数据存储这一麻烦。接下来，我们详细介绍整个算法的实施流程。

我们以一个 L+1层($l=0,1,\cdots,L$)的前馈神经网络为例，具体如图 3-2 所示。第 $l=0$ 层和 $l=L$ 层分别表示神经网络的输入层和输出层。所有隐藏层的激活函数的都相同，用函数 $g(\bullet)$ 表示。$\mathbf{W}_l$ 表示第 $(l-1)$ 层与第 l 层之间的连接权重矩阵，满足 $\mathbf{W}_l \in \mathbb{R}^{s\times m}$。向量 $\mathbf{x}_l$ 和 $\mathbf{y}_l$ 分别表示第 l 层的输出和输入，并满足 $\mathbf{x}_l = g(\mathbf{y}_l)$，

$\mathbf{y}_l = \mathbf{W}_l^T \mathbf{x}_{l-1}$ ($\mathbf{x}_{l-1} \in \mathbb{R}^s$， $\mathbf{y}_l \in \mathbb{R}^m$)。网络参数具体更新过程如下：

i. 参数初始化:。随机初始话各层的权重矩阵 $\mathbf{W}_l(0)$，且 $\mathbf{P}_l(0) = \mathbf{I}_l / \beta$。其中， $l = 1, \cdots, L$；通常令 $\beta = 1$ 即可

ii. 前馈第 j 个任务中第 i 批次数据到网络中, 再利用 BP 算法反馈误差并计算权重 $\mathbf{W}_l(i-1, j)$ 的增量 $\Delta \mathbf{W}_l^{BP}(i, j)$。

iii. 更新各层的权重:

$$\mathbf{W}_l(i, j) = \mathbf{W}_l(i-1, j) + \kappa(i, j) \Delta \mathbf{W}_l^{BP}(i, j) \quad \text{if} \quad j = 1 \qquad \text{式 3-3}$$

$$\mathbf{W}_l(i, j) = \mathbf{W}_l(i-1, j) + \kappa(i, j) \mathbf{P}_l(j-1) \Delta \mathbf{W}_l^{BP}(i, j) \quad \text{if} \quad j = 2, 3, \cdots \qquad \text{式 3-4}$$

其中 $\kappa(i, j)$ 表示学习率，是超参数.

iv. 对于第 j 个任务中各批次数据重复步骤 (ii) 到 (iii) 直到任务完成。

v. 如果第 j 个任务训练完成, 逐个前馈该任务中各批次数据的均值($i = 1, \cdots, n_j$)，并更新各层权重矩阵 $\mathbf{W}_l$ 所对应的正交投影算子 $\mathbf{P}_l$ 。令 $\mathbf{P}_l(j) = \mathbf{P}_l(n_j, j)$，计算过程如下

$$\mathbf{P}_l(i, j) = \mathbf{P}_l(i-1, j) - \mathbf{k}_l(i, j) \bar{\mathbf{x}}_{l-1}(i, j)^T \mathbf{P}_l(i-1, j) \qquad \text{式 3-5}$$

$$\mathbf{k}_l(i, j) = \mathbf{P}_l(i-1, j) \bar{\mathbf{x}}_{l-1}(i, j) / [\alpha + \bar{\mathbf{x}}_{l-1}(i, j)^T \mathbf{P}_l(i-1, j) \bar{\mathbf{x}}_{l-1}(i, j)] \qquad \text{式 3-6}$$

其中 $l-1$ 层在输入第 i 批次数据均值后的输出为 $\bar{\mathbf{x}}_{l-1}(i)$，$n_j$ 代表前

馈第 j 次任务训练数据的批次。之后令 $\mathbf{P}_l(0, j) = \mathbf{P}_l(j-1)$.

vi. 在下一个任务中重复步骤 (ii) 到 (v)，直到序列中的所有任务完成。

上述过程是在整个任务完成后更新正交投影算子 $\mathbf{P}_l$。实际上，我们可以在任务中完成每个批次数据的更新后立即按照步骤 (v) 更新该算子。 α 为遗忘系数；第 l 层的遗忘系数 $\alpha_l = \alpha_l(0)\lambda^{i/n_j}$，$\alpha_l(0)$ 为第 l 层遗忘系数的初值，通常设为 1；λ 是其衰减项的底数。任务刚开始训练时，隐藏层的输入空间尚未收敛到与正确解对应的空间重合，可以令遗忘系数 α 大一些。随着训练的进行，神经网络逐步找到了正确解，减小 α 保证正确计算正交投影算子。

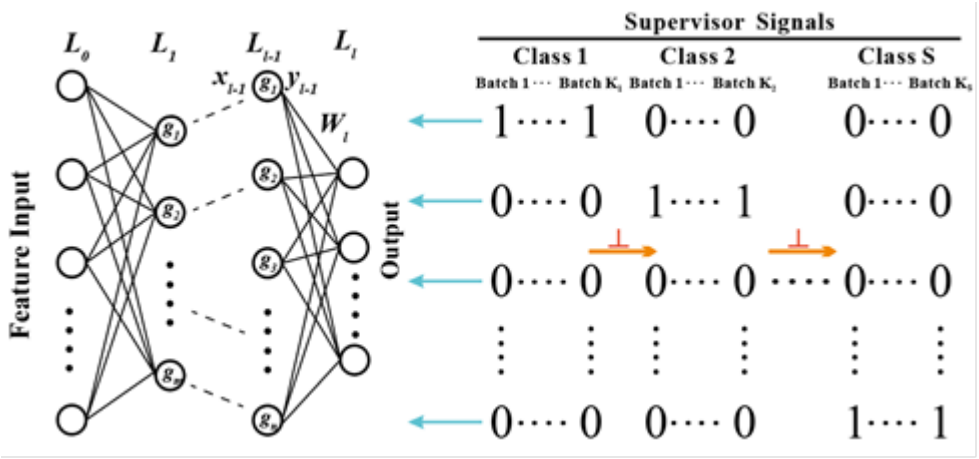

图 3-2　OWM 算法流程示意图

3. 与其他算法的联系与区别

本节将比较 OWM 算法与一些主流算法的区别与联系。首先我们将说明为何 EWC 算法是 OWM 算法的特例，然后说明 OWM 算法与 CAB 算法的区别。最

后将介绍 OWM 算法与信息处理领域的 RLS 算法的联系。

3.1. EWC 算法是 OWM 算法的特例

第 2 章已经较详细的介绍了 EWC 算法[41]的基本思想与具体内容。本节我们将证明 EWC 算法实则是 OWM 的一个特例。

EWC 的损失函数为:

$$L = L_{err} + \frac{1}{2}\lambda L_{reg} \qquad \text{式 3-7}$$

其中

$$L_{reg} = \sum_i (\boldsymbol{\theta}_i - \boldsymbol{\theta}_i^*)^T \mathbf{F}_i (\boldsymbol{\theta}_i - \boldsymbol{\theta}_i^*) \qquad \text{式 3-8}$$

L_{err} 为一般的误差项。$\boldsymbol{\theta}_i$ 和 $\boldsymbol{\theta}_i^*$ 分别为当前任务和上一个任务中神经元 i 的输入权重向量，$\mathbf{F}_i$ 为相应的 Fisher 矩阵。

在式 3-7 两边分别对 $\boldsymbol{\theta}_i$ 取梯度：

$$\begin{aligned} \nabla_{\boldsymbol{\theta}_i} L &= \nabla_{\boldsymbol{\theta}_i} L_{err} + \frac{1}{2}\lambda \nabla_{\boldsymbol{\theta}_i} L_{reg} \\ &= \nabla_{\boldsymbol{\theta}_i} L_{err} + \lambda \mathbf{F}_i (\boldsymbol{\theta}_i - \boldsymbol{\theta}_i^*) \\ &\approx \nabla_{\boldsymbol{\theta}_i} L_{err} - \lambda \mathbf{F}_i \nabla_{\boldsymbol{\theta}_i} L_{err} \\ &= (\mathbf{I} - \lambda \mathbf{F}_i) \nabla_{\boldsymbol{\theta}_i} L_{err} \end{aligned} \qquad \text{式 3-9}$$

在信息通过激活函数出现之前，神经元的输入输出映射可以看作是一个带有白噪声(误差)的线性模型 $y_i = \boldsymbol{\theta}_i^T \mathbf{x}_i + \epsilon_i$。其中，Fisher 信息矩阵 $\mathbf{F}_i$ 和输入的相关矩阵 $\mathbf{R}_i$ 之间存在以下关系，

$$\mathbf{F}_i = \frac{1}{\sigma^2}\mathbf{R}_i \qquad \text{式 3-10}$$

其中σ为白噪声ϵ_i的方差(或误差)。

如果对正交投影矩阵的定义略作改动$\mathbf{P}_i^{(owm)} = \mathbf{I} - (1+\alpha)\mathbf{A}(\mathbf{A}^T A + \alpha\mathbf{I})^{-1}\mathbf{A}^T$。注意这样的改动不会对投影矩阵造成很大影响。假设$\lambda \sim \sigma^2$，那么将式 3-10 代入式 3-9，可以得到

$$\begin{aligned} \nabla_{\boldsymbol{\theta}_i} L &= (\mathbf{I} - \lambda\mathbf{F}_i)\nabla_{\boldsymbol{\theta}_i} L_{err} \\ &= (\mathbf{I} - \frac{\lambda}{\sigma^2}\mathbf{R}_i)\nabla_{\boldsymbol{\theta}_i} L_{err} \\ &\approx \lim_{\alpha\to\infty}\mathbf{P}^{(owm)}\Delta\boldsymbol{\theta}_i^{\,bp} \end{aligned} \qquad \text{式 3-11}$$

可以看到，当α接近无穷大时，EWC 的梯度更新与 OWM 算法等价。EWC 可以近似为 OWM 当$\alpha \to \infty$时的一种特殊情况。

3.2. OWM 算法与 CAB 算法的比较

CAB 是新近提出的连续学习算法[45]。其与 OWM 算法虽然是相互独立提出的，但是二者解决灾难性遗忘的思路十分接近，即权重更新约束在与学习任务的输入空间正交的方向上。此外，CAB 提出的 Conceptor 的作用也类似 OWM 中的正交投影算子。因此有必要比较二者的异同。在接下来，本节将论述这两种方法的关键差异，特别是 CAB 提出的 Conceptor 存在较大缺陷，导致它们之间存在显著的性能差距。

两个算法的核心都是只允许网络权重在历史任务输入空间的正交方向上修改，而关键是如何构造投影矩阵实现这一步骤。投影算子的构造方式不同是 CAB 与 OWM 的关键差异。如前所述，OWM 构造的投影算子为$\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$（$\mathbf{A}$为

历史任务中输入向量构成的矩阵 $\mathbf{A}=[\mathbf{x}_1,\cdots,\mathbf{x}_n]$）。这是数学中正交投影算子的标准定义[62–64]。此外，这里在投影算子计算中引入了 $\alpha\mathbf{I}$，进而投影算子改写为 $\mathbf{A}(\mathbf{A}^T\mathbf{A}+\alpha\mathbf{I})^{-1}\mathbf{A}^T$。这样做的好处是我们可以通过调节参数 α 控制权重增量与输入空间的正交程度，并降低噪声的干扰[64,65]。

在 CAB 算法所用的投影算子则是作者构建的 Conceptor。在原文章[45]中 Conceptor 矩阵被记为 $\mathbf{C}$，其通过最小化损失函数 $\mathbb{E}_x[\|\mathbf{x}-\mathbf{C}\mathbf{x}\|^2]+\alpha^{-2}\|\mathbf{C}\|_{fro}$ 得到。不幸的是，这不是一个合适的损失函数，特别是第二项正则项，不仅无引入的必要，而且还可能带来不必要的副作用。接下来，我们将说明只需要损失函数的第一项 $\mathbb{E}_x[\|\mathbf{x}-\mathbf{C}\mathbf{x}\|^2]$ 中的即可构建的一个合适的投影算子。

显然，若数据向量 $\mathbf{x}$ 不能构成所在空间的完备基并张满整个空间，线性方程

$$\mathbf{x}=\mathbf{C}\mathbf{x} \qquad \text{式 3-12}$$

会有无数多个解 $\mathbf{C}$。特别是连续学习的前期，这种情况会经常存在的。CAB 的作者引入正则项正是为了克服这一问题。但是，如果我们求解方程 $\mathbf{x}=\mathbf{C}\mathbf{x}$ 的极小范数解时，此时方程的解则是唯一的。即存在唯一的精确解 $\mathbf{C}$ 满足 $\min_{\mathbf{x}-\mathbf{C}\mathbf{x}=0}\|\mathbf{C}\|$。我们可以把式 3-12 写为矩阵形式

$$\mathbf{A}=\mathbf{C}\mathbf{A} \qquad \text{式 3-13}$$

其中，$\mathbf{A}=[\mathbf{x}_1,\cdots,\mathbf{x}_n]$，即由历史任务数据组成的矩阵。向量化式 3-14[62]

$$(\mathrm{I}\otimes\mathbf{A}^{\mathrm{T}})\mathrm{vec}(\mathbf{C})=\mathrm{vec}(\mathbf{A}) \qquad \text{式 3-14}$$

这里 $\otimes$ 代表 Kronecker 内积，

$vec(\mathbf{C}) = vec([\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_n]) = [\mathbf{c}_1^T, \mathbf{c}_2^T, \cdots, \mathbf{c}_n^T]^T$ ($vec(\mathbf{A})$ 依此类推)。求解上述方程极小范数解[62]，可以得到：

$$\mathbf{C} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T \qquad \text{式 3-15}$$

式 3-15 正是 OWM 算法中的投影矩阵。然而 CAB 算法并没有选择对式 3-12

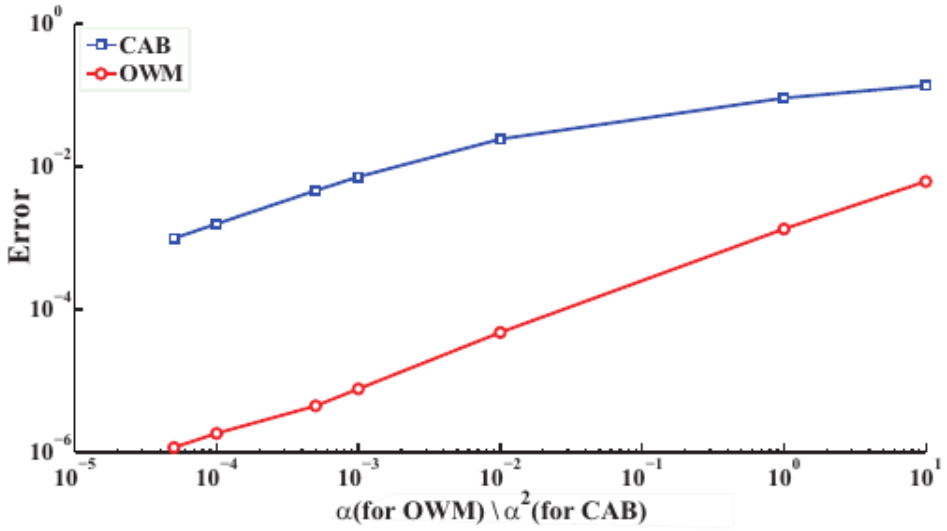

图 3-3　OWM 算法与 CAB 算法的投影效果比较

精确求解，而是选择最小化损失函数 $\mathbb{E}_x[\|\mathbf{x}-\mathbf{C}\mathbf{x}\|^2]+\alpha^{-2}\|\mathbf{C}\|_{fro}$ 求解最小二乘解（该解非精确解，一般常用在精确解不存在的情形）。该函数引入不必要的正则项 $\alpha^{-2}\|\mathbf{A}\|_{fro}$，将不可避免的给投影带来了误差。这意味着，CAB 方法不能保证投影真正正交，也不能很好地保护已经学过的知识。如果想要去除正则项带来的误差，就需要 α^{-2} 趋近于零。但是这样做会导致 Conceptor 无法定义，因为括号中的项将不可逆。特别在连续学习中的早期训练中，CAB 中的关联矩阵 $\mathbf{R}$ 通常不是满秩，也不可逆，进一步加剧上述问题。因此，α^{-2} 将成为一个非常敏感的参数，太小将导致 Conceptor 成为病态矩阵，无法定义；太大则引入较大的不必

要误差，不能保证正交性。这里要特别说明的是，CAB 中的 Conceptor 与 OWM 中的正交投影算子并不等价，也不能通过修改参数来取得正交投影算子的等价效果。

此外，为了证明上述分析，我们用 MNIST 数据集实验的 Conceptor 和正交投影算子的投影效果。我们采用原始输入和其投影之间的误差作为指标。该误差定义为：$\frac{1}{N}\sum_{i=1}^{N}\sqrt{\frac{1}{s}\|\mathbf{x}_i-\mathbf{x}_i^{pro}\|}$，其中 $\mathbf{x}^{pro}$ 是 $(\mathbf{I}-\mathbf{P}^{owm})\mathbf{x}$（对应于 OWM 的情况）或者 $\mathbf{Cx}$（对应与 CAB 的情况），N 是数据样本数，s 是输入向量的维度。误差越小说明投影的效果越好，反之亦然。

在图 3-3 中，我们以 MNIST 数据集中的数字“0”和“1”的图像数据为例，选择了不同的参数绘制了 CAB 和 OWM 的误差曲线。从图中可以看到，与 OWM 相比，上述理论差异确实导致 CAB 的投影效果较差。同时，从图还可以看到，给定精度阈值，OWM 具有更大的可调参数范围。注意，这里横坐标的 α 和 α^2 之间没有任何关系，只是两种算法各自定义的调节参数。第四章将进一步对比二者的连续学习效果。

3.3. OWM 算法与递归最小二乘法之间的联系

递归最小二乘法（recursive least-square, RLS）是信号处理和滤波理论中的最重要的算法之一[64]。本节我们将说明 OWM 算法与 RLS 算法的联系。RLS 算法的计算流程见图 3-4。

Initialize the algorithm by setting

$$\hat{\mathbf{w}}(0) = \mathbf{0},$$
$$\mathbf{P}(0) = \delta^{-1}\mathbf{I},$$

and

$$\delta = \begin{cases} \text{small positive constant for high SNR} \\ \text{large positive constant for low SNR} \end{cases}.$$

For each instant of time, $n = 1, 2, \ldots$, compute

$$\boldsymbol{\pi}(n) = \mathbf{P}(n-1)\mathbf{u}(n),$$
$$\mathbf{k}(n) = \frac{\boldsymbol{\pi}(n)}{\lambda + \mathbf{u}^H(n)\boldsymbol{\pi}(n)},$$
$$\xi(n) = d(n) - \hat{\mathbf{w}}^H(n-1)\mathbf{u}(n),$$
$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)\xi^*(n),$$

and

$$\mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{u}^H(n)\mathbf{P}(n-1).$$

图 3-4 RLS 算法的计算流程[64]

下面我们将说明 OWM 中的投影算子与 RLS 算法中的矩阵 $\mathbf{P}^{(RLS)} = (\sum_{i=1}^{n}\mathbf{x}(i)\mathbf{x}^T(i) + \alpha\mathbf{I})^{-1}$ 在一定意义下是等价的。$\boldsymbol{\Phi}(n) = \sum_{i=1}^{n}\gamma^{-1}\mathbf{x}(i)\mathbf{x}(i)^T + \alpha\gamma^n\mathbf{I}$，表示输入信号的相关矩阵。在 $\sum_{i=1}^{n}\mathbf{x}(i)\mathbf{x}^T(i) + \alpha\mathbf{I}$ 可逆的情况下，$\boldsymbol{\Phi}$ 的逆表示为 $\mathbf{P}^{(RLS)}$，即 $\mathbf{P}^{RLS}(n) = \boldsymbol{\Phi}^{-1}(n)$，其中。假设遗忘因子 $\gamma = 1$，并令 $\mathbf{A}(n) = [\mathbf{x}(1), \mathbf{x}(2), \cdots, \mathbf{x}(n)]$,其中 $\mathbf{x}(i)$ 表示第 i 个输入向量。可以将 $\boldsymbol{\Phi}$ 写成矩阵形式：$\boldsymbol{\Phi}(n) = \mathbf{A}(n)\mathbf{A}^T(n) + \alpha\mathbf{I}$。由 Woodbury 矩阵恒等式可得：

$$\begin{aligned} \mathbf{P}^{RLS}(n) &= \alpha^{-1}\mathbf{I} - \alpha^{-1}\mathbf{A}(\mathbf{I} + \alpha^{-1}\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\alpha^{-1} \\ &= \alpha^{-1}[\mathbf{I} - \mathbf{A}(\alpha\mathbf{I} + \mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T] \end{aligned} \tag{3.3}$$

正是OWM算法中的正交投影算子。

早些时候 RLS 算法也被用于神经网络的训练以加速求解的收敛速度[66]，并被称为增强 BP 算法(Enhanced Back Propagation，EBP)。由于二者之间的相似性，OWM 算法和 EBP 有相同的计算复杂度 O($N_n N_w^2$)，这里的 N_n 是所有神经元的数量， N_w 是每个神经元的输入权重[66]。

4. OWM 算法的容量问题

在神经网络结构给定的情况下，一个神经网络所能学习的任务量终归是有限的。神经网络可以连续学习的任务数量上限，即为该网络的容量。连续学习的算法对网络容量有着重要影响。本小节 OWM 相应的网络容量做一个粗略的估计。在神经网络中，每一层神经元的容量可以用该层相应的正交投影算子 $\mathbf{P}^{(owm)}$ 的秩来度量。首先定义 $\mathbf{P}_i^{(owm)}$ 为完成第 i 个任务后得到的正交投影算子， $\Delta\mathbf{P}_{i+1}^{(owm)}$ 为算子下一个任务中所更新的矩阵增量。由于投影矩阵的正交性质，$range(\mathbf{P}_i^{(owm)}) \cap range(\Delta\mathbf{P}_{i+1}^{(owm)}) = \varnothing$，上述满足等式 $\mathbf{P}_{i+1}^{(owm)} = \mathbf{P}_i^{(owm)} - \Delta\mathbf{P}_{i+1}^{(owm)}$，同时有：

$$rank(\mathbf{P}_{i+1}^{(owm)}) = rank(\mathbf{P}_i^{(owm)}) - rank(\Delta\mathbf{P}_{i+1}^{(owm)}) \qquad \text{式 3-16}$$

可以看到，随着学习过程的进行，该层神经元的容量在减小。若 $rank(\mathbf{P}_l^{(owm)}) = 0$，表明该层已耗尽其容量。显然，该层的容量上限可以用投影算子的满秩来衡量。那么整个网络的容量大致可以用各层容量之和 $\sum_{l=1}^{L} rank(\mathbf{A}_l)$ 来近似。

如果整个网络已耗尽其容量，可以考虑两种解决方案:一是引入遗忘因子，通过遗忘一些旧任务为新任务腾出空间。二是添加更多的层或神经元。

5. 权重协同更新

本小节简单讨论 OWM 算法对 BP 算法带来的影响。从第 1 和第 2 小节的讨论可以看出，OWM 算法是基于 BP 算法的。但是从最终效果来看，二者却有着本质的不同。其不同体现在：OWM 会出现权重协同更新的现象。在 OWM 算法，若两个神经元的发放具有相关性，则两个神经元会共享彼此的权重增量，因此更新时有协同效应。其原因在于正交投影矩阵往往不是一个对角证，存在非零非对角项。这些非对角项会产生比较重要的影响，例如产生的协同更新效应会显著的增加网络容量。但是，这样的更新方式是非局域的，在生物上实现似乎有一定的困难。OWM 算法类似的机制是否存在与真实的神经系统中是一个有趣的问题。我们会在今后的研究工作中进一步探讨。

6. 本章小结

OWM 算法是实现连续学习的新算法。OWM 算法的基本思想和原理清晰明确，可以从原理上保证完全克服灾难性遗忘。而其实现过程也较为简单方便，算法的关键——正交投影算子迭代实现，不需要保存历史数据。本章还讨论了算法的容量问题，比较了与关联算法的区别和联系。特别是我们证明了 EWC 是 OWM 算法的一种特例，也说明了对于 CAB 算法的优势。在下一章中，我们将通过数值实验进一步说明本算法的优越性。

第4章.OWM 算法的效果测试

本章将介绍 OWM 算法的实际测试效果。我们首先在连续学习的标准任务——手写体数字的连续学习任务上测试其效果。该任务是非常经典的连续任务，几乎所有的连续算法都会在该任务上进行测试，这也方便我们与其他算法进行比较。我们还在大数据集上——ImageNet 和手写体汉字数据上验证了 OWM 算法的有效性，特别是文字学习是人类生活中一个典型的连续学习任务，方便 OWM 算法与人类的学习效果进行比较。

1. MNIST 连续识别任务

MNIST 数据集是人工智能领域最基础、最常用的数据集。MNIST [67]数据库是由美国国家标准与技术研究院（NIST）收集的手写阿拉伯数字图片库，包括数字 0 到 9。 MNIST 拥有 60,000 个训练样本和 10000 个测试样本。 每个样本都是 28×28 灰度图片。在连续学习的研究中，有两个标准的任务模式，都是基于 MNIST 数据的。一个是 Shuffled MNIST 任务，另一个是 Disjoint MNIST 任务。本节将在这两个任务上测试 OWM 算法的效果，并与其他算法进行比较。

1.1. Shuffled MNIST 任务

Shuffled MNIST 任务是指连续学习多个基于 MNIST 数据集的图片识别任务。对于每一个任务，其目标都是将图片中的数字正确识别出来。但是，在不同任务中，图片会以不同但固定的置换规则将图片的像素随机置换(图像标签保持

不变，即 1 的像素被打乱重置后其标签仍为 1)。在同一个任务中，所有图片的置换方式都是相同的。不同任务的差别在于像素的置换方式不同。具体操作如图 4-1[68]所示

在这个任务中，我们分别采用了用[784-800-10]（3 层）、[784-800-800-10]（4 层）、[784-2000-2000-10]（4 层）前馈神经网络完成识别任务。这些都是其他的连续学习算法实现该任务时采用的结构，我们采用同样的结构以方便比较。采用的损失函数为常用的交叉熵，且 ReLU 作为激活函数。

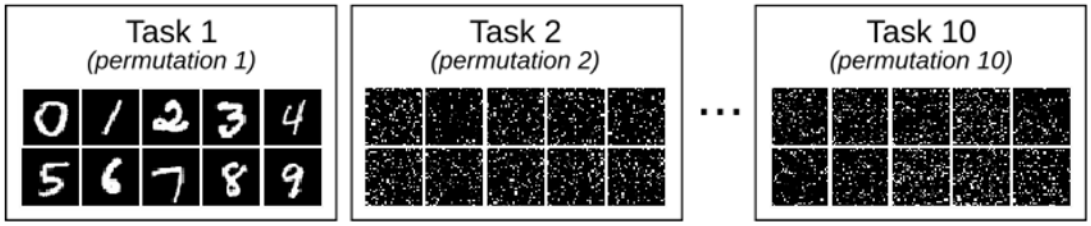

图 4-1 Shuffled MNIST 任务示意图

在表 4-1 中，我们在不同的网络结构上，将 OWM 算法与其他主流算法的连续学习效果做了比较。从表中可以看到，OWM 算法优于其他算法，特别是相对于 EWC、CAB 算法优势较为明显。另一方面，由于 SGD 算法在这个任务中也有不俗的表现（正确率高达 70%），所以该任务的灾难性遗忘问题不是特别严重，所以各算法较容易达到正确率上限。

Shuffled MNIST实验			
3 tasks	Accuracy (%)	10 tasks	Accuracy (%)
SGD#	71.32 ± 1.54*	EWC#	~ 97.0
IMM#	98.30 ± 0.08n.s	**OWM#**	**97.52 ± 0.03**
EWC#	~ 98.2	EWC†	~ 89.0
MA(pk)	98.14	CAB†	~ 95.2
OWM#	**98.34 ± 0.02**	OWM†	95.15 ± 0.08
		Synaptic Intelligence‡	~ 97.2
		OWM‡	**97.64 ± 0.03**

表 4-1　在 Shuffled MNIST 任务中，OWM 算法与其他主流算法的比较。†代表网络结构为[784-100-10]；#代表网络结构为[784-800-800-10]；‡代表网络结构为[784-2000-2000-10]。其他方法的结果来自对应的文献。表中的均值和方差都是重复 10 次试验后统计得到的结果。若原始文献中未统计方差表中也不予显示。

1.2. Disjoint MNIST 任务

Disjoint MNST 任务是指在连续学习多个任务，每个任务只学习一部分的数字。例如，Disjoint MNST 2 将十个数字分为两个任务来学习：第一个任务识别数字{0,1,2,3,4}；第二个任务是识别数字{5,6,7,8,9}；Disjoint MNST 10 分为 10 个任务分别学习各个数字。这两个任务的学习流程如图 4-2 所示。

我们首先在 Disjoint MNIST 2 任务测试了 OWM 算法，并与其他算法进行了比较。表 4-2 显示了各算法在连续学习任务中识别的正确率。可以看到，OWM 优于其他算法，且具有较大的优势。另外 HAT 算法采用独有的的网络结构，即

[784-800-800-(5x2)]，故未放在表中比较。其输出是两个独立的模块，因此做任务需要任务标签，以决定使用那个输出模块查看结果。在第 2 章中我们已经论述过，这样的结构并不实用。而且预先知道任务信息，将大大提高正确率，但这种性能提升不是由算法本身带来的。若使用该结构，HAT 算法准确度高达 99.03%，效果甚至超过 OWM 算法。但是，如果采用表 4-2 的结构，其正确率只有 91.50%左右。

图 4-2 Disjoint MNIST 任务示意图。A. Disjoint MNIST 2 任务。 B. Disjoint MNIST 10 任务。

Disjoint MNIST实验	
Methods	Accuracy (%)
EWC#	52.72 ±1.36*
IMM#	94.12 ±0.27*
OWM#	**96.59 ±0.06**
SGD†	53.85 ±0.14*
CAB†	94.91 ±0.30*
OWM†	**96.30 ±0.03**

表 4-2 在 Disjoint MNIST 2 任务中，OWM 算法与其他算法的连续学习效果比较。† 代表网络结构为[784-800-10]；#代表网络结构为[784-800-800-10]。其他方法的结果来自对应的文献。表中的均值和方差都是重复 10 次试验后统计得到的结果。

在表 4-2 中可以看到，除 OWM 算法以外，CAB 在这个任务上的性能最优。考虑到该算法的思想与 OWM 接近，因此有必要做进一步的比较。图 4-3 展示了 OWM、CAB 和 SGD 在 Disjoint MNST 2 训练过程中的学习曲线。若没有其他操作，传统的算法 SGD 会在第 2 个任务中迅速忘记第一个任务的学习成果，发生灾难性遗忘。OWM（红线）和 CAB（绿线）算法都可以克服灾难性遗忘。但 OWM 的性能显然优于 CAB，并且收敛的更快。图 4-4 测试了连续学习更多任务时三者的性能表现。图中每个点表示学习完横坐标对应的数字后，对所有历史任务（包括刚完成的数字学习任务）测试的平均结果。可以看到，随着任务数目的增加，OWM 算法的性能优势更加明显。在上述任务中，所有算法采用相同的网络结构。这进一步证实了 OWM 算法的优越性。

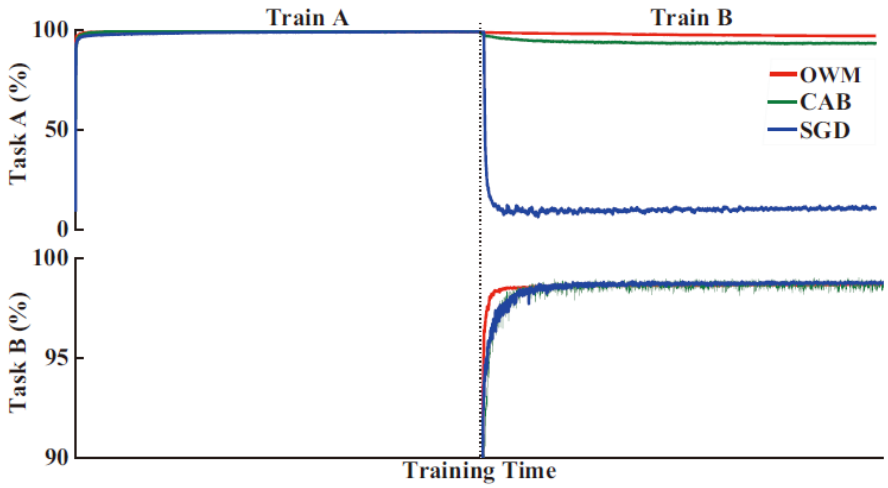

图 4-3 Disjoint MNIST(2 个任务)中 OWM、CAB 和 SGD 的学习曲线

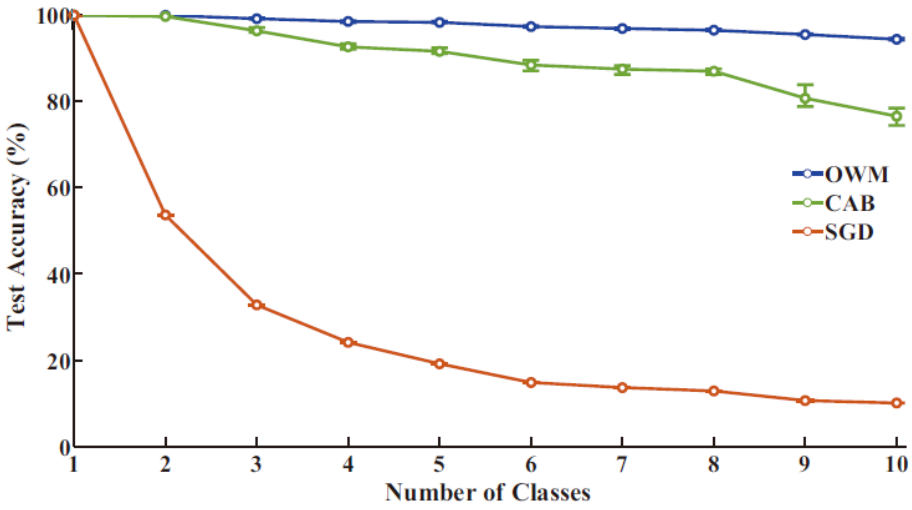

图 4-4 Disjoint MNIST 10 任务中 OWM、CAB 和 SGD 的性能曲线。

2. Image Net 连续识别任务

我们选用 ImageNet 数据集进一步测试测试 OWM 算法在大数据集上的适用性。ImageNet 是世界上最大的图像识别数据库[69]，其总共搜集了高达 1500 万张图片，涉及 2.2 万个分类，并且每张图片都经过了严格的筛选和标记。不过人们通常会使用它的一个子集——ILSVR2012 数据集。 其中训练集共有 120 万张图片，1000 个分类。验证集包含相同分类的共 50,000 张图片。本节将采用 OWM 算法，连续的学习这 1000 个分类。由于 ImageNet 数据集中的图片通常涉及复杂场景，目标物也具有十分复杂的特征，因此需要提取特征。在实际当中，由于特征提取器的训练十分耗时，通常会使用预训练好的提取器，然后输出后用多层感知机做微调。这里我们使用深度卷积神经网络提取特征，然后在输出的多层感知机上使用 OWM 算法（如图 4-5 所示）。这样可以避免繁冗的特征提取工作。具体的，我们使用 ResNet[70]作为特征提取器，并将提取好的特征向量馈送到一个 2 层的分类器进行分类。

Data Set	Classes	Feature Extractor	CT by SGD (%)	ST by OWM (%)	ST by SGD (%)
ImageNet	1000	ResNet152	78.31	73.80	0.69

表 4-3 OWM 算法在 ImageNet 数据集上进行 1000 类图片连续学习的性能。CT：Concurrent Training，即所有分类的数据混在一起训练；ST：Sequential Training，即各类的数据依次训练。

在表 4-3 中，OWM 算法在连续学习 1000 类图片后测试的平均正确率达 73.8%，与这 1000 类同时训练得到的结果十分接近。如果不使用 OWM 算法，只利用传统的 SGD 算法的话，正确率不到 1%。上述结果表明我们的算法可以有效的进行大数据集上的多任务连续学习。

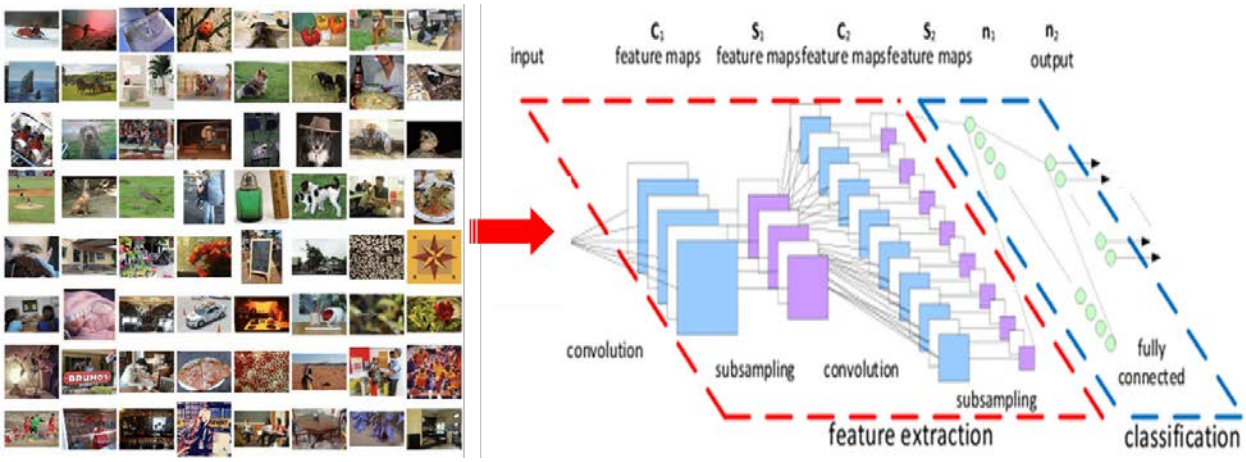

图 4-5 基于 ImageNet 的连续学习任务网络构架示意图。

3. 手写体汉字连续识别任务

文字的学习是人类的一项十分典型的连续学习任务，也十分具有挑战性。为测试 OWM 算法是否具备类似的连续学习能力，我们在一个汉字的手写体数据集上对其性能进行了测试。该数值实验采用中文手写体的离线数据库

CASIA-HWDB[71]。该数据集有中国科学院自动化研究所国家模式识别重点实验室（NLPR）收集整理。这里我们使用它的 CASIA-HWDB1.1 子集。该子集收录了 300 多人手写的一百万个汉字样本，涉及 3755 个常用汉字。 每个汉字都有 240 个训练图像和 60 个测试图像。这里采用了与上一节 ImageNet 任务类似的网络构建，只是特征提取器略有不同。

Data Set	Classes	Feature Extractor	CT by SGD (%)	ST by OWM (%)	ST by SGD (%)
CASIA-HWDB1.1	3755	ResNet18	97.46	92.11	8.07

表 4-4 OWM 算法在 CASIA-HWDB1.1 数据集上进行 3000 多类汉字手写体图片连续学习的性能。CT、ST 的含义同表 4-3。

测试的效果见表 4-4。可以看到，OWM 可以有效的实现多达 3755 类汉字的连续学习，其正确率与传统的所有样本同时训练时接近。若只使用传统 SGD 算法连续学习其正确率只有 8%。值得一提的是，该正确率人类的测试水平基本接近(~96%)[72]。这 3755 个字属于汉字一类字库，基本覆盖小学阶段所需要学习的汉字。这意味着即便对于人类而言，连续学习这些字也需要数年的时间，且期间还需要不断地复习。上述结果充分说明了 OWM 算法的有效性。

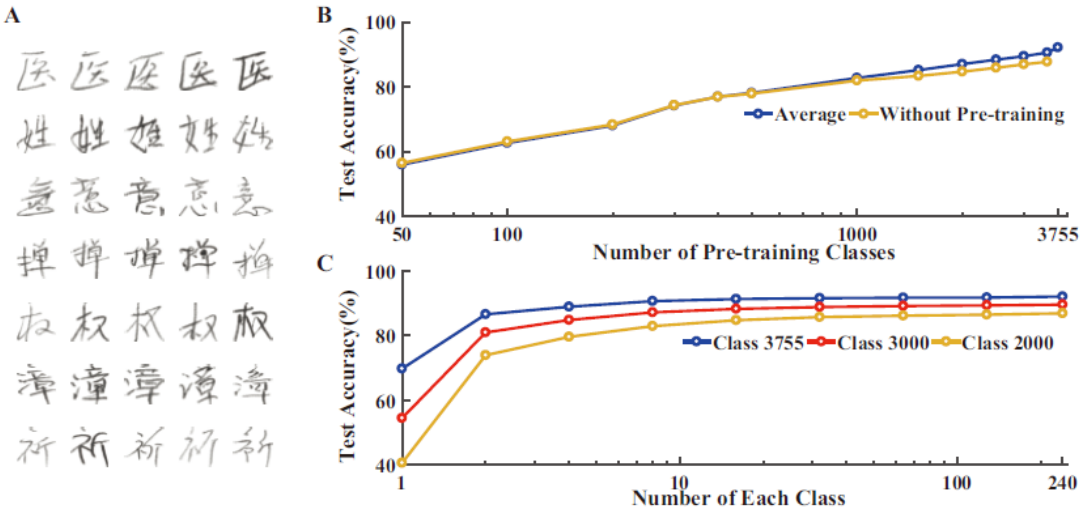

图 4-6　OWM 辅助汉字手写体的连续学习任务。A. CASIA-HWDB1.1 **数据样本示意图。**B. **特征提取器预训练样本种类数目对连续学习效果的影响。**C. **每个任务所用的样本数目对连续学习效果的影响。**

我们在实验中使用了预训练过的特征提取器。在实际中，特征提取器在预训练的过程很难穷尽所有的样本种类，也不能保证见过的种类足够丰富。考虑到这些情形，我们进一步研究预训练过程中样本种类对 OWM 算法的影响。图 4-6（B）显示了预训练样本的种类对连续学习效果的影响。只用 50 类样本训练特征提取器时，最终连续学习所有任务的平均正确率（蓝线）就达 50%。当预训练种类为所有种类的 15%时，正确率可达 80%以上。这说明在连续学习任务中，特征提取器不起主要作用，也不需要其在预训练过程中见过所有任务种类。

另一个很重要的问题是，OWM 算法需要多少样本来完成连续学习。图 4-6（C）显示，每个任务类别只用一个样本进行训练，连续学习就可以达到 50%左右正确率。而样本到 10 个时，OWM 算法的性能基本就可以很好的发挥出来，达到 80%以上的正确率。图中三种不同颜色的曲线代表预训练特征提取器时使用的样本种类，三条曲线显示了一致的变化趋势。该结果一方面说明 OWM 算法具备快速学习的能力，即便只看到过少量样本，OWM 也可以高效完成连续学习；

另一方面说明特征提取器在预训练中是否遇到过某个任务，在该任务以后的连续学习中不起主要作用。

4. 网络参数

Shuffled MNIST 任务使用经典的多层感知机神经网络。其中神经元激活函数为 ReLU(Rectified Linear Units)，输出层使用了 Dropout，drop 的概率为 0.2；损失函数加入 L2 正则项，约束因子为 0.001。Disjoint MNIST 使用经典的多层感知机神经网络，其中神经元激活函数为 ReLU，训练时损失函数为交叉熵，而且使用了动量技术，动量因子为 0.9。ImageNet 和 Chinese Handwriting 两个任务使用均方根损失函数。

在以上所有实验中，连续训练任务时，我们没有使用负样本。换句话说，在训练过程中，只有一个与任务对应类别的正样本被输入给网络。表 4-5 和表 4-6 为各种超参数详细情况。表 4-6 中 ResNet 来自文章[70]，使用的优化方法是 RMSprop，学习率都为 0.1，权重衰减率都为 0.0001，全都使用 Pytorch 框架实现。

Experiment	α	λ	κ	**Batch Size**
Shuffled MNIST (3 Tasks)	1.0	1.0	1.0	100
Shuffled MNIST (10 Tasks)	1.0	1.0	4.0	100
Disjoint MNIST (3 Layers)	0.9/0.6	0.001/1.0	0.2	40
Disjoint MNIST (4 Layers)	0.9/1.0/0.6	0.001/0.1/1.0	0.2	40
Disjoint MNIST (SGD)	NA	NA	0.01	50
CASIA-HWDB1.1 (3 Layers)	1.0/0.5	0.02/1.0	2.0	50
CASIA-HWDB1.1(SGD)	NA	NA	0.0001	200
ImageNet ILSVR2012 (3 Layers)	1.0/1.0	0.005/1.0	2.0	30
ImageNet ILSVR2012 (SGD)	NA	NA	0.0001	1000
CelebA (2 Layers in Fig .4.4A)	1.0	1.0	1.5	20
CelebA (2 Layers in Fig .4.4C)	1.0	1.0	0.1	1

表 4-5　使用 OWM 方法任务中的超参数

Experiment	Feature Extractor/Output Size	Batch Size
ImageNet ILSVR2012	ResNet152 / 2048	512
CASIA-HWDB1.1	ResNet18 / 1024	512
CelebA	ResNet50 / 2048	256

表 4-6 不同任务中特征提取器的超参数

5. 本章小结

本章我们在不同的数据集、不同的网络结构和不同的任务类型上测试了连续学习的效果。在基于 MNIST 数据集的标准连续学习任务上，OWM 算法的性能达到了 State of art 的水平，优于现有的其他主流算法。在 ImageNet 等大数据集上，面对更多的任务数量和更复杂的图片结构，OWM 算法也可以高效的实现连续学习，其性能与所有任务同时训练的结果有可比性。在汉字的连续学习任务中，OWM 算法取得了接近人类的水准，具备快速学习的能力。以上结果表明，OWM 算法是实现连续学习的高效算法。

第5章.基于 OWM 算法的情境学习算法

我们所处的环境往往复杂多变。一个理想的智能体需要具备灵活响应这样环境的能力。该能力显然为人类所擅长。神经科学的研究表明，前额叶是人类情境学习能力的生物基础。本章我们将介绍如何通过借鉴前额叶的功能，在 OWM 算法实现连续学习的基础上，让人工神经网络具备情境学习的能力。

1. 情境学习模块

认知控制是大脑的核心能力之一，并由前额叶脑区实现。本节，我们将提出情境学习模块（context-dependent processing， CDP），模拟前额叶的功能，根据情境信息调整神经网络的信息处理过程。

CDP 模块包含两个部分：第一部分是编码模块，它将情境信息转换为适当的控制信号;第二部分是调制模块，其使用编码模块输出的控制信号对系统输入进行调制。具体如图 5-1（A）所示。这两个模块代表了情境学习的两个核心问题：❶ 情境信号的编码问题，包括如何在实际中界定情境和非情境信号，如何从输入信号快速提取情境信号以及情境信号如何编码。这些问题都是非常有挑战性的问题。当前我们不对这个问题做详细的探讨，默认已知情境信号，并直接将情境信号编码为一个随机信号。❷ 情境信号对神经网络的控制方式。这个问题是我们探讨的重点。我们通过很多实验尝试，发现直接调整神经元权重并旋转输入信号在输入空间的表示是最为有效的方式。

情境信号对神经网络的调制通过以下方式实现。如图 5-1（A）,向 CDP 模块输入情境信号，通过编码模块转换为控制信号 $\mathbf{C}=[c_1,c_2,...,c_m]^T\in\mathbb{R}^m$，然后控制

信号作为输入层的输出权重调制神经网络的信息处理过程。输入层接受任务的特征向量 $\mathbf{F}=[f_1, f_2,\cdots,f_k]^{\mathrm{T}}\in\mathbb{R}^k$，经过一个隐藏层然后产生输出 $\mathbf{Y}^{\mathrm{out}}=[y_1,y_2,...,y_m]^{\mathrm{T}}\in\mathbb{R}^m$，满足 $y_i=c_i\,\mathrm{g}(net_i)$，$net_i=\left(\mathbf{w}_{\mathrm{i}}^{\mathrm{in}}\right)^{\mathrm{T}}\mathbf{F}=\sum_{j=1}^{k}f_j w_{ji}^{\mathrm{in}}$，$\mathrm{g}=\max(0,x)$。隐藏层的输入权重矩阵 $\mathbf{W}^{\mathrm{in}}=[\mathbf{w}_1^{\mathrm{in}},\mathbf{w}_2^{\mathrm{in}},..,\mathbf{w}_m^{\mathrm{in}}]\in\mathbb{R}^{k\times m}$ 随机产生且在以后的学习过程中不做修改。系统之后的输出部分（图中为输出权重矩阵 $\mathbf{W}^{\mathrm{out}}=[w_1,w_2,...,w_m]^{\mathrm{T}}\in\mathbb{R}^m$）是以后的训练过程中需要学习的部分；不同信号矩阵 $\mathbf{C}$ 代表不同的情境信息。

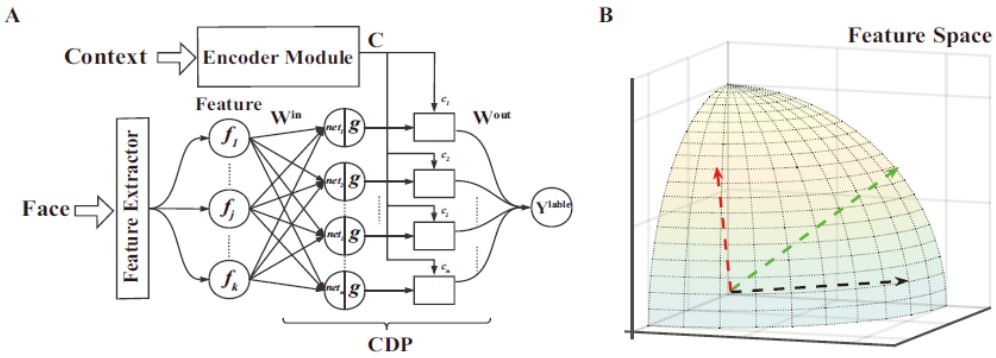

图 5-1　CDP 模块的结构图以及三维示意图

CDP 模块层的数学形式可以表示为：

$$
\begin{aligned}
\mathbf{Y} &= \mathrm{g}\left(\left(\mathbf{W}^{\mathrm{in}}\right)^{\mathrm{T}}\mathbf{F}\right)\bullet\mathbf{C} \\
&= \mathrm{g}\left(\left[\mathbf{w}_1^{\mathrm{in}},\ \mathbf{w}_2^{\mathrm{in}},\ \cdots,\ \mathbf{w}_p^{\mathrm{in}}\right]^{\mathrm{T}}\mathbf{F}\right)\bullet\mathbf{C} \\
&= \mathrm{g}\left(\left[\sum_{j=1}^{k} f_j w_{j1}^{\mathrm{in}},\ \sum_{j=1}^{k} f_j w_{j2}^{\mathrm{in}},\ \cdots,\ \sum_{j=1}^{k} f_j w_{jp}^{\mathrm{in}}\right]^{\mathrm{T}}\right)\bullet\mathbf{C} \\
&= \mathrm{g}\left(\left[c_1\|\mathbf{F}\|\left\|\mathbf{w}_1^{\mathrm{in}}\right\|\cos\theta_1,\ c_2\|\mathbf{F}\|\left\|\mathbf{w}_2^{\mathrm{in}}\right\|\cos\theta_2,\ \cdots,\ c_p\|\mathbf{F}\|\left\|\mathbf{w}_p^{\mathrm{in}}\right\|\cos\theta_p\right]^{\mathrm{T}}\right) \\
&= \mathrm{g}\left(\left[c_1\left\|\mathbf{w}_1^{\mathrm{in}}\right\|\cos\theta_1,\ c_2\left\|\mathbf{w}_2^{\mathrm{in}}\right\|\cos\theta_2,\ \cdots,\ c_p\left\|\mathbf{w}_p^{\mathrm{in}}\right\|\cos\theta_p\right]^{\mathrm{T}}\right)\|\mathbf{F}\| \\
&= \mathrm{g}\left(\left[c_1\cos\theta_1,\ c_2\cos\theta_2,\ \cdots,\ c_p\cos\theta_p\right]^{\mathrm{T}}\right)\|\mathbf{F}\|
\end{aligned}
\qquad \text{式 5-1}
$$

其中$\odot$代表向量对应元素相乘；θ_i表示向量$\mathbf{w}_i^{\mathrm{in}}$与向量$\mathbf{F}$之间的角度。对于任何$\upsilon\geq 0$, $\mathrm{g}(\upsilon x)=\max(0,\upsilon x)=\upsilon\max(0,x)=\upsilon\,\mathrm{g}(x)$。对于特定样本而言，其特征向量$\mathbf{F}$是固定的。那么，输入层经 CDP 模块调整后的输出$\mathbf{Y}$只受情境控制信号$\mathbf{C}$的影响。如果对$\mathbf{Y}$进行归一化$\sqrt{\sum_{i=1}^{m}\left(c_i\left\|\mathbf{w}_{\mathrm{i}}^{\mathrm{in}}\right\|\mathrm{g}\left(\cos\theta_i\right)\right)^2}$，可以发现，CDP 模块相当于“旋转”了特征空间中的输入向量(见图 5-1 （B）)。因此，CDP 模块可以改变输入的表示，同时由保持信息内容不变。

2. CDP 模块的情境学习性能测试

本节我们将介绍如何把 OWM 算法和 CDP 模块结合起来实现人工神经网络的连续学习。该实验目的是判别人脸的属性特征。传统这属于多任务学习的范畴，需要使用多个分类模块，识别同一张人脸的不同属性。实际上，联合使用 OWM 算法和 CDP 模块，只需要一个判别器就可以实现上述目的。实现方式的核心是基于基于情境信号的连续学习。本研究中使用明星脸数据集 CelebA[73]。该数据集包含 10177 个明星共 202599 张人脸图像，图片中涉及不同的姿态、角度和复杂的背景图案。每张图像有 40 个属性标签。

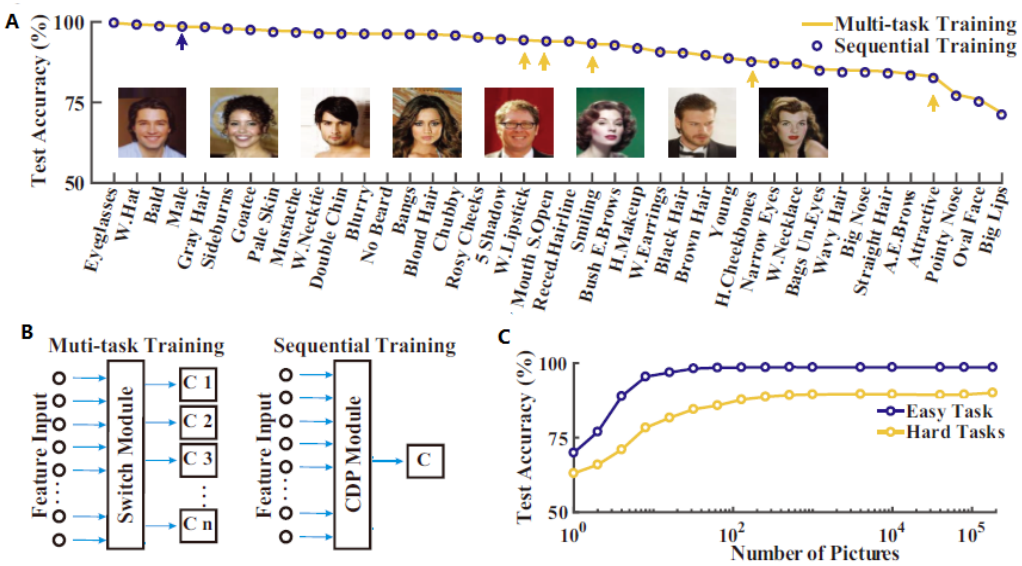

图 5-2 人脸属性识别任务。A. 多任务学习和基于情境信号的连续学习效果的比较。 B.多任务学习和情境学习的网络架构的比较。 C. 预训练中样本的数量对结果的影响

和之前一样，我们首先训练 ResNet50 做特征提取，输出 2048 维的特征向量。然后将提取好的特征向量输入到输入层，同时 CDP 模块接受情境信号对输入进行调制。本实验的情境信号是任务要识别的属性的英文单词，再通过 gensim 工具包(Radim,2010)转换为 200 维词向量后，然后通过维度匹配输出 5000 维的控制信号向量 **C** 。实验中识别模块的结构为[2048-5000-1]。由于只有一个判别器和一个输出端，因此需要采取连续学习的训练方式。而每次训练什么任务则由情境信号决定。判别器采用 OWM 算法更新权重。相应网络参数请参加表表 5-1 和表 5-2。

Experiment	a	λ	κ	Batch Size
CelebA (2 Layers in Fig .4.4A)	1.0	1.0	1.5	20
CelebA (2 Layers in Fig .4.4C)	1.0	1.0	0.1	1

表 5-1 人脸属性识别任务的中参数设置

Experiment	Feature Extractor/Output Size	Batch Size
CelebA	ResNet50 / 2048	256

表 5-2 人脸属性识别任务中特征提取器的参数设置

这样的训练方式与传统的多任务模式显著不同。如图 5-2（B）所示，相对于多任务模式（左图），我们的采用的方式（右图）可以得到更为紧凑的网络结构 。图 5-2（A）显示了两种方式下测试结果的比较。可以看到，虽然连续学习（蓝点）的方式只用了一个判别器，但效果与使用多个判别器基本相当。表 5-1 是二者在不同属性判别任务中的具体正确率，之间只有不到 1%的差距，充分说明了 CDP 模块的有效性。

我们进一步中讨论了样本数对性能的影响。我们特别选取了 1 个较容易的任务（判别人脸的性别）和 5 个较难的任务（判别人脸是否具有吸引力、是否微笑、嘴唇厚度、是否张嘴、颧骨高低）。这几个任务比较具有代表性且样本分布比较均衡。在图 5-2（C）可以看到与之前类似的结果，我们的方法可以小样本的学习。无论任务难易，只用少量的样本就可以达到较好的学习效果。

Attributes	5 Shadow	Arched E.Brows	Attractive	Bags Un.Eyes	Bald	Bangs	Big Lips	Big Nose	Black Hair	Blond Hair	Blurry	Brown Hair	Bush E.Brows	Chubby
ST	94.50	82.29	82.47	84.24	98.58	96.22	71.10	83.93	90.22	95.90	96.11	89.55	92.63	95.68
MT	94.82	83.83	83.01	85.28	90.00	96.27	71.66	84.89	90.55	96.10	96.28	89.56	92.95	95.83
Attributes	Double Chin	Eyeglasses	Goatee	Gray Hair	Heavy Makeup	H.Cheekbones	Male	Mouth Small Open	Mustache	Narrow Eyes	No Beard	Oval Face	Pale Skin	Pointy Nose
ST	96.17	99.63	97.41	98.29	91.69	87.53	98.45	93.88	96.59	87.12	96.04	75.16	96.73	76.96
MT	96.41	99.67	97.67	98.36	91.98	87.96	98.56	93.99	97.08	87.35	96.28	75.91	97.25	77.44
Attributes	Reced.Hairline	Rosy Cheeks	Sideburns	Smiling	Straight Hair	Wavy Hair	Wear Earrings	Wear Hat	Wear Lipstick	Wear Necklace	Wear Necktie	Young	Average	
ST	93.87	95.10	97.76	93.19	84.26	84.75	90.53	99.07	94.24	86.90	96.24	88.59	91.26	
MT	93.97	95.12	97.88	93.13	84.60	85.05	90.70	99.15	94.32	87.11	96.61	88.69	91.56	

表 5-3 **两种学习模式下识别不同人脸属性的正确率。**ST：sequential training。 MT：Multi-task training。

3. 本章小结

本章详细介绍了 CDP 模块实现情境学习的基本原理和思想，给出了有效的处理框架，并理论分析了其作用机制。通过配合 OWM 算法，CDP 模块可以只使用一个判别器的情况下有效对相同输入做不同的处理。二者结合，一方面大大压缩了网络结构的复杂性，使系统更加紧凑；另一方面，将为我们实现更加类脑的通用人工智能打下基础。

第6章.总结与展望

本研究的最终目标是实现人工神经网络的连续学习和情境学习。对于连续学习，我们提出 OWM 算法。该算法的核心在于控制网络权重的更新方向，保证系统在旧任务的解空间中寻找新任务的解。该算法不需要增加网络结构，不需要保存数据，计算方便简洁。数值实验证明该方法可以高效的实现人工神经网络的连续学习，效果优于同类主流的算法。对于情境学习，我们提出 CDP 模块。该模块受前额叶皮层的功能启发，是一个类脑的计算模块。配合 OWM 算法，该模块可以让人工神经网络依据情境信息灵活调制信息的处理过程，同时使网络结构更加紧凑。总之二者结合将为我们实现更加类脑的人工神经网络打下基础。

在未来的研究中，我们进一步深化现有的研究结果。具体有两个方向：一、研究 OWM 算法在动力学神经系统中的实现方案。由于 OWM 算法是一种非局域的算法，这样的非局域效应在大脑中是否存在，若存在如何实现是十分有趣的问题。通过对这些问题的研究将进一步加深我们对大脑连续学习机制的了解，同时也会对开发更加高效的连续学习算法提供启示。二、研究 CDP 模块主动识别、匹配情境信息的机制。现在的方案中，情境信号是给定的。显然在真实情形中情境也不可穷数，也不可预知，需要神经网络自身学习得到。这是非常重要的问题。无论神经科学领域还人工智能领域都非常关注。我们希望能提出有效算法实现这一目的。

参 考 文 献

[1] MCCULLOCH W S, PITTS W. A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY* (reprinted from 1943)[J]. Bulletin of Mothemnticnl Biology, 1990, 52(l2): 99–115.

[2] MCCULLOCH W S, PITTS W H. originally published in: Bulletin of Mathematical Biophysics, Vol. 5, 1943, p. 115-133[J]. Bulletin of Mathematical Biophysics, 1943, 5: 115–133.

[3] PAPERT M M and S. Perceptrons. An Introduction to Computational Geometry.[J]. Science, 1969, 165: 780–782.

[4] D.E. RUMELHART, G. HINTON, R. J. WILLIAMS. Learning representations by back-propagation errors[J]. Nature, 1986, 323: 533–536.

[5] CYBENKO G. Degree of approximation by superpositions of a sigmoidal function[J]. Approximation Theory and its Applications, 1993, 9(3): 17–28.

[6] KURT HORNIK, MAXWELL STINCHCOMBE, HALBERT WHITE. Multilayer Feedforward Networks are Universal Approximators[J]. Neural Networks, 1989, 2: 359–366.

[7] HINTON, G., DENG, L., YU, D., DAHL, G., MOHAMED, A., JAITLY, N., … KINGSBURY B. Deep Neural Networks for Acoustic Modeling in Speech Recognition.[J]. IEEE Signal Processing Magazine, 2012(November): 82–97.

[8] DAHL G E, YU D, DENG L等. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition[J]. IEEE Transactions on Audio, Speech and Language Processing, 2012.

[9] LI FEI-FEI, WEI DONG, JIA DENG等. ImageNet: A large-scale hierarchical image database[J]. researchgate.net, 2009: 248–255.

[10] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet Classification with Deep Convolutional Neural Networks[J]. ImageNet Classification with Deep Convolutional Neural Networks, 2012: 1097–1105.

[11] 张荣， 李伟平， 莫同. 深度学习研究综述[J]. 信息与控制, 2018, 47(4): 385–397.

[12] BROSE K. Global Neuroscience[J]. Neuron, 2016, 92(3): 557–558.

[13] AMUNTS K, EBELL C, MULLER J等. The human brain project: creating a European research infrastructure to decode the human brain[J]. Elsevier, .

[14] MARTIN C, NEURON M C-, 2016 undefined. The BRAIN initiative: building, strengthening, and sustaining[J]. Elsevier, .

[15] XIONG Z-Q, TAN T, POO M等. China Brain Project: Basic Neuroscience, Brain Diseases, and Brain-Inspired Computing[J]. Neuron, 2016, 92(3): 591–596.

[16] PARISI G I, KEMKER R, PART J L等. Continual Lifelong Learning with Neural Networks: A Review[J]. Neural Networks, Pergamon, 2018, 113: 54–71.

[17] BARNETT S, BULLETIN S C-P, 2002 undefined. When and where do we apply what we learn[J]. psycnet.apa.org, .

[18] FRENCH R M. Catastrophic forgetting in connectionist networks[J]. Trends in Cognitive Sciences, Elsevier Current Trends, 1999, 3(4): 128–135.

[19] BRAUN C, HEINZ U, SCHWEIZER R等. Dynamic organization of the somatosensory cortex induced by motor activity[J]. Brain, Narnia, 2001, 124(11): 2259–2267.

[20] LEWKOWICZ D J. Early experience and multisensory perceptual narrowing[J]. Developmental Psychobiology, John Wiley & Sons, Ltd, 2014, 56(2): 292–315.

[21] POWER J D, SCHLAGGAR B L. Neural plasticity across the lifespan[J]. Wiley Interdisciplinary Reviews: Developmental Biology, 2017, 6(1): e216.

[22] MURRAY M M, LEWKOWICZ D J, AMEDI A等. Multisensory Processes: A Balancing Act across the Lifespan[J]. Trends in Neurosciences, 2016, 39(8): 567–579.

[23] ZENKE F, GERSTNER W, GANGULI S. The temporal paradox of Hebbian learning and homeostatic plasticity[J]. Current Opinion in Neurobiology, 2017, 43: 166–176.

[24] MERMILLOD M, BUGAISKA A, BONIN P. The stability-plasticity dilemma: investigating the continuum from catastrophic forgetting to age-limited learning effects[J]. Frontiers in Psychology, 2013, 4.

[25] QUADRATO G, ELNAGGAR M Y, DI GIOVANNI S. Adult neurogenesis in brain repair: cellular plasticity vs. cellular replacement[J]. Frontiers in Neuroscience, 2014, 8.

[26] KIYOTA T. Neurogenesis and brain repair[G]//Neuroimmune Pharmacology.

Cham: Springer International Publishing, 2016: 575–597.

[27] HENSCH T K, FAGIOLINI M, MATAGA N等. Local GABA circuit control of experience-dependent plasticity in developing visual cortex[J]. Science, American Association for the Advancement of Science, 1998, 282(5393): 1504–1508.

[28] YANG G, PAN F, GAN W B. Stably maintained dendritic spines are associated with lifelong memories[J]. Nature, 2009, 462(7275): 920–924.

[29] MCCLELLAND J L, MCNAUGHTON B L, O'REILLY R C. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory[J]. Psychological Review, 1995, 102(3): 419–457.

[30] RUSU A A, RABINOWITZ N C, DESJARDINS G等. Progressive Neural Networks[J]. 2016.

[31] YUSTE R. Electrical Compartmentalization in Dendritic Spines[J]. Annual Review of Neuroscience, Annual Reviews , 2013, 36(1): 429–449.

[32] DRAELOS T J, MINER N E, LAMB C C等. Neurogenesis Deep Learning[C]//2017 International Joint Conference on Neural Networks (IJCNN). IEEE, 2016: 526–533.

[33] PARISI G I, TANI J, WEBER C等. Lifelong learning of human actions with deep neural network self-organization[J]. Neural Networks, 2017, 96: 137–149.

[34] PART J L, LEMON O. Incremental online learning of objects for robots operating in real environments[C]//2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob). IEEE, 2017: 304–310.

[35] XIAO T, ZHANG J, YANG K等. Error-Driven Incremental Learning in Deep Convolutional Neural Network for Large-Scale Image Classification[C]//Proceedings of the ACM International Conference on Multimedia - MM '14. New York, New York, USA: ACM Press, 2014: 177–186.

[36] ZHOU G, SOHN K, LEE H. Online Incremental Feature Learning with Denoising Autoencoders[J]. 2012: 1453–1461.

[37] REBUFFI S-A, KOLESNIKOV A, SPERL G等. iCaRL: Incremental Classifier and Representation Learning[C]//2017 IEEE Conference on Computer Vision and Pattern

Recognition (CVPR). IEEE, 2017: 5533–5542.

[38] SHIN H, LEE J K, KIM J等. Continual Learning with Deep Generative Replay[J]. 2017.

[39] KEMKER R, KANAN C. FearNet: Brain-Inspired Model for Incremental Learning[J]. 2017.

[40] UDERS B, SCHLÄGER M, RISI S. Continual Learning through Evolvable Neural Turing Machines[J]. NIPS 2016 Workshop on Continual Learning and Deep Networks, 2016: 1–5.

[41] KIRKPATRICK J, PASCANU R, RABINOWITZ N等. Overcoming catastrophic forgetting in neural networks[J]. Proceedings of the National Academy of Sciences, National Academy of Sciences, 2017, 114(13): 3521–3526.

[42] ZENKE F, POOLE B, GANGULI S. Continual learning through synaptic intelligence[J]. Proceedings of the 34th International Conference on Machine Learning - Volume 70, JMLR.org, 2017: 3987–3995.

[43] LEE S-W, KIM J-H, JUN J等. Overcoming Catastrophic Forgetting by Incremental Moment Matching[J]. 2017: 4652–4662.

[44] SERRÀ J, SURÍS D, MIRON M等. Overcoming catastrophic forgetting with hard attention to the task[J]. 2018.

[45] HE X, JAEGER H. Overcoming Catastrophic Interference by Conceptors[J]. 2017.

[46] LI Z, HOIEM D. Learning without Forgetting[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(12): 2935–2947.

[47] HINTON G, VINYALS O, DEAN J. Distilling the Knowledge in a Neural Network[J]. 2015.

[48] MINSKY M. Unified theories of cognition[M]. Encyclopedia of Cognitive Science, {Harvard University Press}, 2003, 59(1–2).

[49] MILLER G A, HEISE G A, LICHTEN W. The intelligibility of speech as a function of the context of the test materials.[J]. Journal of Experimental Psychology, 1951, 41(5): 329–335.

[50] MCCLELLAND J L, RUMELHART D E. An interactive activation model of

context effects in letter perception: I. An account of basic findings.[J]. Psychological Review, 1981, 88(5): 375–407.

[51] DESIMONE R, DUNCAN J. Neural Mechanisms of Selective Visual Attention[J]. Annual Review of Neuroscience, Annual Reviews 4139 El Camino Way, P.O. Box 10139, Palo Alto, CA 94303-0139, USA , 1995, 18(1): 193–222.

[52] FRIES P. Neuronal Gamma-Band Synchronization as a Fundamental Process in Cortical Computation[J]. Annual Review of Neuroscience, Annual Reviews , 2009, 32(1): 209–224.

[53] SIEGEL M, BUSCHMAN T J, MILLER E K. Cortical information flow during flexible sensorimotor decisions.[J]. Science (New York, N.Y.), American Association for the Advancement of Science, 2015, 348(6241): 1352–5.

[54] FUSTER J M. The prefrontal cortex[M]. .

[55] PASSINGHAM R E, WISE S P. The neurobiology of the prefrontal cortex : anatomy, evolution, and the origin of insight[M]. Oxford Univ. Press, 2012.

[56] MILLER E K. The Prefrontal Cortex: Complex Neural Properties for Complex Behavior[J]. Neuron, Cell Press, 1999, 22(1): 15–17.

[57] MILLER E K. The prefontral cortex and cognitive control[J]. Nature Reviews Neuroscience, Nature Publishing Group, 2000, 1(1): 59–65.

[58] PASSINGHAM R E. The frontal lobes and voluntary action[M]. Oxford University Press, 1995.

[59] DIAS R, ROBBINS T W, ROBERTS A C. Primate analogue of the Wisconsin card sorting test: Effects of excitotoxic lesions of the prefrontal cortex in the marmoset.[J]. Behavioral Neuroscience, 1996, 110(5): 872–886.

[60] LECUN Y, BENGIO Y, HINTON G. Deep learning[J]. Nature, Nature Publishing Group, 2015, 521(7553): 436–444.

[61] ROHRBACH M, STARK M, SCHIELE B. Evaluating knowledge transfer and zero-shot learning in a large-scale setting[C]//CVPR 2011. IEEE, 2011: 1641–1648.

[62] BEN-ISRAEL A, GREVILLE T N E (Thomas N E. Generalized inverses : theory and applications[M]. Springer, 2003.

[63] YANAI H, TAKEUCHI K, TAKANE Y. Projection matrices, generalized inverse

matrices, and singular value decomposition[M]. Springer, 2011.

[64] HAYKIN S S. Adaptive filter theory[M]. .

[65] MOUSTAKIDES G V. Study of the transient phase of the forgetting factor RLS[J]. IEEE Transactions on Signal Processing, 1997, 45(10): 2468–2476.

[66] SHAH S, PALMIERI F, DATUM M. Optimal filtering algorithms for fast learning in feedforward neural networks[J]. Neural Networks, 1992, 5(5): 779–787.

[67] LECUN Y, BOTTOU L, BENGIO Y等. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, Institute of Electrical and Electronics Engineers Inc., 1998, 86(11): 2278–2324.

[68] VAN DE VEN G M, TOLIAS A S. Generative replay with feedback connections as a general strategy for continual learning[J]. 2018.

[69] RUSSAKOVSKY O, DENG J, SU H等. ImageNet Large Scale Visual Recognition Challenge[J]. 2014.

[70] HE K, ZHANG X, REN S等. Deep Residual Learning for Image Recognition[J]. 2015.

[71] LIU C-L, YIN F, WANG D-H等. Online and offline handwritten Chinese character recognition: Benchmarking on new databases[J]. Pattern Recognition, Elsevier Science Inc., 2013, 46(1): 155–162.

[72] YIN F, WANG Q-F, ZHANG X-Y等. ICDAR 2013 Chinese Handwriting Recognition Competition[C]//2013 12th International Conference on Document Analysis and Recognition. IEEE, 2013: 1464–1470.

[73] LIU Z, LUO P, WANG X等. Deep Learning Face Attributes in the Wild[C]//2015 IEEE International Conference on Computer Vision (ICCV). IEEE, 2015: 3730–3738.

致 谢

首先衷心感谢我的合作导师余山研究员。在博士后的研究中，余老师给予了很多支持和帮助，让我受益良多。而在学术讨论中，余老师经常一语道破问题的关键，令我茅塞顿开。同时在这些讨论中，我学习到了很多神经科学的理论方法，启发和推动本研究的开展。

感谢蒋田仔研究员为我们提供了良好的学术环境。蒋老师组织的组会讨论和邀请报告，让我接触到了很多脑科学领域的前沿课题，开阔了研究视野。感谢脑网络中心各位老师在学术上和生活上给我的诸多帮助。

感谢黄旭辉副研究员在博后期间对我的支持和帮助，一直以来黄师兄给了我很多的建议，帮助我顺利完成了本课题。

感谢曾冠雄硕士，本研究是我们倾力合作的结果。他有很多优秀的品质值得我学习，特别是合作中，他的勤奋努力让我印象深刻。感谢张金鹏博士、牛威坤博士、崔波博士、胡古月博士，在与他们的讨论中我受益匪浅。感谢组里每一位师弟、师妹的帮助与支持。

衷心感谢所有关心和帮助我的老师、同学和朋友们!

陈阳

2019 年 3 月

个人简历、在学期间发表的论文与研究成果

研究论文：

1. Guanxiong Zeng, **Yang Chen**, Shan Yu, Modeling of Brain-Computer Interfaceaided Training in Rehabilitation [C]//2018 2th IEEE International Conference on Computer Systems, Electronics and Control (ICCSEC). IEEE, 2018.
2. Zeng G*, **Chen Y***, Cui B, et al. Continuous Learning of Context-dependent Processing in Neural Networks[J]. arXiv preprint arXiv:1810.01256, 2018. （共同第一作者，under review）

专利

1. 曾冠雄，**陈阳**，余山.基于情景信号类前额叶网络的信息处理方法、系统、装置，发明专利，已受理，申请号：201910058284.2；
2. **陈阳**，曾冠雄，余山.基于正交投影矩阵的人工神经网络优化方法、系统、装置，发明专利，已受理，申请号：201910138155.4；