

基于多任务持续学习的医学图像分类算法研究

摘要

肺癌是发病率和死亡率增长最快，对人群健康和生命威胁最大的恶性肿瘤疾病之一，尽早地发现和预防对于疾病诊断和延缓病情有着重要的帮助。肺结节医学图像在不同 CT 征象如恶性度（Malignancy）、分叶征（Lobulation）以及钙化程度（Calcification）在不同的临床阶段有着不同的表现，能够对疾病的诊断和预防起到不同的辅助作用，因此需要同时参考多种 CT 征象进行多种分类任务的探究。

神经网络天然存在灾难性遗忘的问题，即在后面任务的学习过程中由于参数的变化会导致之前所学任务的遗忘，因此无法同时学习多种任务，而多任务持续学习克服了这一问题，能够同时对多种 CT 征象分类任务进行学习。本文基于多任务持续学习，针对肺结节医学图像的三种分类任务（恶性 Malignancy、分叶征 Lobulation、钙化程度 Calcification）进行了探究。文章首先构建了卷积神经网络，针对肺结节医学图像进行了特征提取，在 LIDC 数据集上进行了预处理和训练，并取得了不错的效果。在此基础上，文章为了实现多任务持续学习，使用了基于 EWC（elastic weight consolidation）算法的网络分类模型，此外我们还对学习上述几种持续学习分类任务的顺序做了进一步研究。最后文章对目前算法存在的问题以及未来努力的方向做了进一步的阐述，并对多任务持续学习辅助的肺癌疾病诊断的前景进行了展望。

关键词：肺结节 CT 医学图像，多任务持续学习，卷积神经网络，LIDC，EWC 弹性权重合并

装
订
线

Research on Medical Image Classification Algorithm Based on Multi-task Continuous Learning

ABSTRACT

Lung cancer is one of the malignant diseases with the fastest increase in morbidity and mortality, and the greatest threat to the health and life of the population. Early detection and prevention is of great help to the diagnosis and delay of the disease. The medical images of lung nodules have different manifestations in different CT signs such as Malignancy, Lobulation and Calcification at different clinical stages, which can assist in the diagnosis and prevention of diseases. Therefore, it is necessary to simultaneously refer to multiple CT signs for exploration of multiple classification tasks.

Neural networks naturally have the problem of catastrophic forgetting, that is, the change of parameters in the learning process of the following tasks will cause the forgetting of the previously learned tasks, so they cannot learn multiple tasks at the same time. Multi-task continuous learning overcomes this problem. It can learn multiple CT sign classification tasks at the same time. Based on multi-task continuous learning, this thesis explores the three classification tasks (Malignancy, Lobulation, Calcification degree) of lung nodules medical images. The thesis first constructed a convolutional neural network, carried out feature extraction for medical images of lung nodules, preprocessed and trained on the LIDC data set, and achieved good results. On this basis, in order to achieve multi-task continuous learning, the article uses a network classification model based on the EWC (elastic weight consolidation) algorithm. In addition, we have further studied the effects of different order of the above-mentioned classification tasks. Finally, the thesis further elaborates on the existing problems of the current algorithm and the direction of future efforts, and puts forward a prospect of lung cancer diagnosis assisted by multi-task continuous learning.

Key words: CT image, multi-task continuous learning, CNN, LIDC, EWC

装
订
线

目 录

1 引言	1
1.1 简要介绍	1
1.2 肺结节检测和医学影像	2
1.3 多任务持续学习	3
1.4 文章主要工作	4
1.5 文章内容安排	5
2 相关工作	6
2.1 相关工作介绍与综述	6
2.2 不同方法相关工作介绍	7
2.2.1 多任务持续学习方法	7
2.2.2 SGD 最速随机下降法	8
2.2.3 L2 正则化	12
3 基于 EWC 弹性权重合并的多任务肺结节 CT 图像分类算法	18
3.1 网络框架介绍	18
3.2 EWC 优化算法介绍	20
3.3 网络的训练和测试	23
4 实验结果和分析	25
4.1 LIDC 数据集	25
4.1.1 数据来源	25
4.1.2 解析结果	25
4.2 数据预处理和数据增强	27
4.2.1 肺结节图像数据预处理	27
4.2.2 肺结节图像数据增强	30
4.2.3 肺结节图像多任务分类	31
4.3 多任务持续学习实验方案介绍	34
4.4 多任务持续学习实验结果分析和讨论	35
4.4.1 分类模型评估参数	35
4.4.2 分类模型实验结果	36
4.4.3 分类模型实验分析	43
4.5 不同任务顺序对实验结果影响的分析和讨论	47
4.5.1 实验设计	47
4.5.2 实验结果	47
4.5.3 实验分析	53
5 结论和展望	54
5.1 结论	54
5.2 展望	54
参考文献	55
谢 辞	57

装
订
线

1 引言

1.1 简要介绍

肺癌是我国发病率和死亡率排名均为第一的癌症，对人们的生命健康具有很大的威胁，是人尽皆知的恶性肿瘤之一。有临床数据表明，肺癌患者中，超过七成的患者在确诊时病情已到了中晚期，这种程度的肺癌往往早已经错过了最佳治疗时机，这也是造成肺癌死亡率如此之高不可忽视的重要原因。之所以肺癌难以在早期被及时的发现，是因为肺癌的发病过程比较隐匿，许多的肺癌患者在出现早期的肺癌症状时误以为是其他疾病，从而导致病情被忽视，等到发现却为时已晚。因此伴随着科学技术的进步，越来越多先进的技术与医学图像分类相结合，作为辅助手段帮助医生进行早期肺癌的诊断。

对肺癌患者进行早期有效的诊断是一项重要的工作，如果能够做到及时的诊断肺癌患者的病情，做到早发现早治疗，那么肺癌将会得到有效的控制，死亡率也会大大降低。

近日，世卫组织国际癌症研究机构（IARC）发布了2020年全球最新癌症负担数据，统计了全球185个国家36中癌症类型的最新发病率、死亡率情况，以及癌症发展趋势。

这份报告显示：2020年全球1930万人新确诊癌症，近1000万人死亡；每5人中就有1人将在其一生中患癌症；每8名男性，每11名女性中就有1人将因癌症而死亡；癌症诊断后5年生存人数约为5060万；乳腺癌成为全球最常见癌症；2020年，以女性为主的乳腺癌发病人首次超过全人群肺癌，成为全球最常见的癌症。但是，从死亡人数来看，肺癌依然以180万排名第一，远超其他癌症类型。

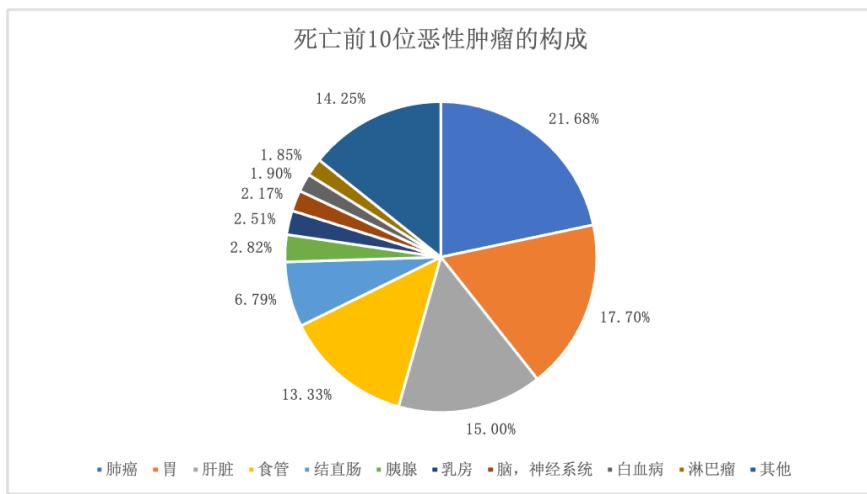


图 1.1 死亡前 10 位恶性肿瘤的构成示意图

肺癌会引起一系列的危害，比如导致患者出现咳嗽，胸痛，呼吸困难等症状，严重影响病人的日常生活；肺癌的病人可能出现咯血，严重情况下会出现低血容量性的休克，甚至可以引起窒息死亡；肺癌如果发展到了晚期，将导致严重的营养不良，内环境紊乱，甚至引起病人死亡等等。而且到目前为止，治愈肺癌仍是一个医学难题，但是及早的发现和干预病症，将会有效地延缓发

病进程，控制疾病发展。对于肺结节良恶性分类和预测研究的意义在于，通过结合病人的有效信息，辅助医生给出更加优质的诊断结果，合理掌握病人未来的发病状态和发病时间，尽早地预防和治疗病人。

在肺结节良恶性的早期诊断方法中，医学影像检测是目前临床使用最为广泛的方法，并且发展出了一门完整的神经影像学科。自上世纪 70 年代以来，计算机辅助诊断系统就被提出来辅助医生、提高诊断的准确率。这些计算机辅助系统属于从医学图像数据中提取特征向量并训练的监督学习系统。而提取这些特征向量需要人类专家大量的先验知识，这通常会耗费大量的人力物力，并且受限于数据本身的数量和质量。随着现代医学诊断技术与计算机科学技术的不断发展，医生对病人信息的获得更加全面，并且在医学辅助诊断自动化领域逐渐获得了一套标准化的方法。现在医学可以通过 X 射线、CT 扫描、超声诊断仪、和磁共振成像（Magnetic Resonance Imaging, MRI）等技术获取病人的信息。随着深度学习的发展，提取特征曾经这一需要人类专家参与的关键步骤也正在逐步趋于自动化。结合深度学习方法来研究肺结节的医学影像，有助于帮助医生获取更加全面的信息，减轻医生的负担，并且能够在发病早期就给出一些合理的信号来辅助医生干预肺癌的早期治疗。

1.2 肺结节检测和医学影像

随着机器学习技术与神经影像学技术的发展，利用计算机分析医学影像，获取医学影像的统计学特征今儿从医学影像中获取更深层次的信息已经逐步发展成为一项热门研究课题。早在 2012 年，就有学者提出了影像组学的概念，它强调从医学影像中发现更深层次的信息用来辅助医生做出准确的诊断。最早影像组学概念的提出主要是为了服务肿瘤疾病的诊断和治疗，其主要的处理流程可以总结为：影像数据的获取、肿瘤区域的标定、肿瘤区域的分割、特征的提取和量化、影像数据库的建立、分类和预测。在使用的方法上，传统的影像组学也会更加偏向于使用传统的机器学习算法来解决问题。随着深度学习技术的发展，传统影像组学与计算机领域的较差越来越多，影像组学这一概念也更加的广泛。总结来说，目前的影像组学的梳理流程可以大致归纳为影像数据的获取、目标区域的标定与分割、特征提取、分类和预测这几个步骤。

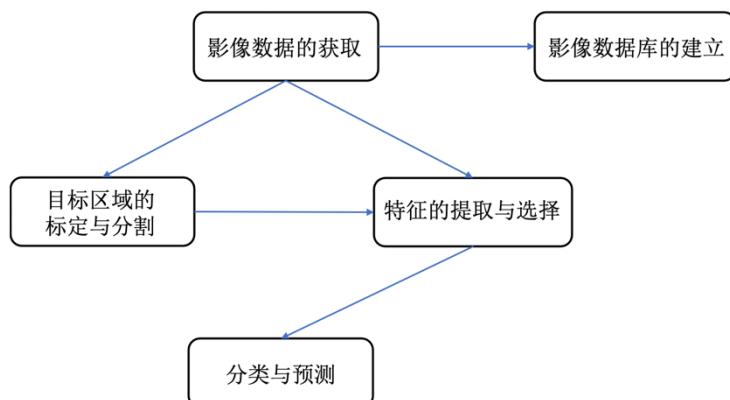


图 1.2 影像组学的基本流程

在影像数据的选择方面，一般希望它满足两方面的要求：一方面数据应该具有良好的代表性并且在不同类别特征之间表现出明显的差异，这将有利于模型能够较好的学习不同特征的表达；另一方面数据量应尽可能的充分，这不仅有利于模型的训练也有利于模型不受个别有差异个体的影响，从而提高模型的泛化能力。

在特征选择问题上，传统的影像组学方法需要人类专家提供感兴趣区域（Region of Interest, ROI），并在此基础上提取特征。而深度学习作为机器学习的一个重要分支，可以自动地从医学影像中学习潜在的特征，这一工作可以大量节省人力物力，并且能够发现一些潜在的生物学特征，为医生提供更多有效的信息。

在分类和预测的过程中，由于临幊上采集到的原始医学影像一般维度较高，不能够直接应用于分类或学习的模型当中，因此特征提取成为了分类与预测的主要问题，一些学者通过将原始数据变换到更合适的特征子空间，达到降低特征维度的目的，代表性的工作有主成分分析（Principal Component Analysis, PCA）、独立判别分析（Linear Discriminant Analysis, LDA）等方法；其他的学者通过特征选择，丢弃一些不相关特征，构建出判别效果最好的特征子集来达到降低特征维度的目的，代表性的工作有支持向量机（Support Vector Machine, SVM）、稀疏特征学习、深度学习等方法。

众多研究结果表明，在数据量充足的情况下，结合深度学习对肺结节图像进行分析得到的结果往往优于许多传统影像组学的方法，深度学习甚至可以挖掘一些潜在的生物学特征，为肺结节的疾病发展提供更多有效的信息。

1.3 多任务持续学习

我们人类有能够将一个任务的知识运用到另一个任务的能力，学习后一个任务时也不会忘记如何做前一个任务，这种能力叫持续学习（continual learning）。而想要深度学习神经网络拥有这个能力，归结起来主要有两个问题：如何能把之前任务的经验用上，使得网络能够更快更好的学习当前任务；以及在学习当前任务时，如何保证不会忘记之前已经学会的任务。用更专业的数据来讲就是可塑性（学习新知识的能力）和稳定性（旧知识的记忆能力）。

神经网络不同于人类，由于其自身的设计天然存在灾难性遗忘的问题。当学习一个新任务的时候，需要更新网络中的参数，但是上一个任务提取出来的知识也是存储在这些参数上的。于是，神经网络在学习新任务的时候，旧任务的知识就会被覆盖，这就导致了灾难性遗忘。

持续学习（continual learning）研究从无线数据流中学习的问题，其目的是逐渐扩充知识并将其运用于未来的学习过程中。这类学习的一个关键特征是其序列特性（sequential nature）：在某一时刻只有一个或部分数据/任务可获得。持续学习的主要挑战是灾难性遗忘（catastrophic forgetting）：当新任务被学习时过去获得的知识被遗忘，即之前任务的表现不如以往。

Sebastian Ruder^[1]介绍了深度学习中最常用的两种多任务持续学习方法：隐层参数的硬共享和软共享。

（1）参数的硬共享机制：参数的硬共享机制是神经网络的多任务学习中最常见的一种方式。一般来讲，它可以应用到所有任务的所有隐层上，而保留任务相关的输出层。硬共享机制降低了过拟合的风险，这一点是非常有意义的。越多任务同时学习，模型就能捕捉到越多任务的同一个

表示，从而导致我们在原始任务上的过拟合风险越小。

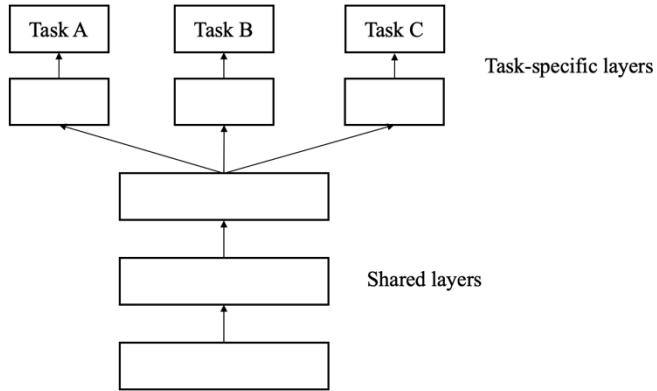


图 1.3 参数的硬共享机制示意图

(2) 参数的软共享机制：每个任务都有自己的模型，自己的参数。通过对模型参数的距离进行正则化来保证参数的相似，比如 L2 正则化方式和迹正则化方式。用于深度神经网络中的软共享机制的约束很大程度上是受传统多任务学习技术中正则化技术的影响。

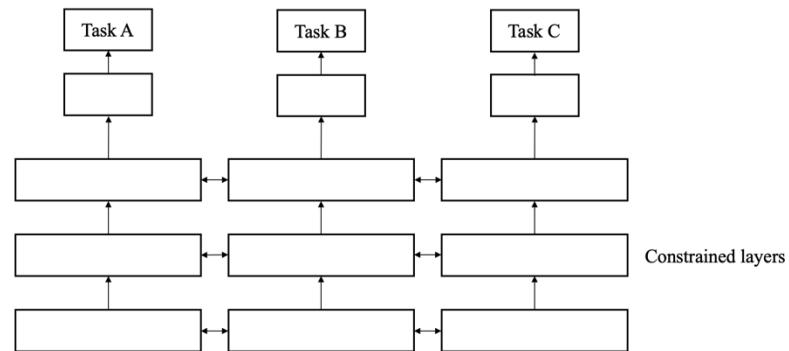


图 1.4 参数的硬软共享机制示意图

1.4 文章主要工作

文章在调研了目前各项基于医学图像的肺结节分类研究后，基于多任务持续学习和神经网络，解决了多任务肺结节图像分类中存在的灾难性遗忘问题，实现了肺结节图像在不同 CT 征象分类的持续学习。文章的主要工作如下：

- (1) 总结了近年来给予多任务持续学习在肺结节图像分类中的各类研究，并进行了相关分析。
- (2) 构建了深度卷积残差神经网络（ResNet）用于解决肺结节图像的分类问题。

(3) 由浅入深，在上述 ResNet 的基础上，使用 EWC 弹性权重合并算法实现多任务持续学习，让此神经网络能够同时进行肺结节图像在多种 CT 征象上的分类工作，如恶性度(Malignancy)、分叶征 (Lobulation)、钙化程度 (Calcification) 等等。在此基础上，对上述的几种分类问题的组合顺序进行进一步的探究。

(4) 分析研究的实验结果，并对文本的实验得出结论与展望。

1.5 文章内容安排

本文主要内容分为六章，内容安排如下：

第一章，介绍肺结节、深度学习与医学图像的背景，概括了本文的主要研究工作；

第二章，总结近年来基于多任务持续学习的肺结节图像分类的各类研究，并进行相关分析；

第三章，介绍了本文所使用或借鉴的各类技术的理论与方法，对比和分析了各类方法的特性；

第四章，介绍本文的主要研究工作，首先介绍本文实验所选取的数据集，进行数据预处理和数据增强工作；然后对肺结节图像进行多任务的分类，进而通过 EWC 算法实现多任务持续学习的实验，同时以 SGD 随机梯度下降和 L2 正则化两种方法进行对比实验，对 EWC 算法在多任务持续学习上的表现进行进一步探究；

第五章，对实验结果进行分析和讨论；

第六章，总结全文，并对未来研究工作进行展望。

装
订
线

2 相关工作

2.1 相关工作介绍与综述

近年来，随着医学和计算机技术的发展，计算机辅助诊断（Computer Aided Diagnosis, CAD）系统在临幊上发挥着越来越重要的作用，在一些容错率高、工作量大的任务方面甚至可以代替人类医生取得不错的效果。在肺结节图像分类方面，不断有研究者在探索如何通过一些原始的医学数据（影响、代谢特征等生物学信息），利用计算机，分析并给出病人的诊断结果。根据现有的研究状况，可以发现计算机辅助诊断系统在肺结节图像的一些分类问题上已经取得了不错的結果，但在多任务持续学习方面的研究仍然面临着较多的问题。

由于原始的医学图像通常是高维数据，一般的卷积神经网络不能直接对这些数据进行分析。针对肺结节 CT 图像数据，目前较为普遍的处理方法有 3 种模式，一种模式是将 3D 数据划分为许多的 2D 切片，通过卷积神经网络对这些 2D 切片进行数据分析；另一种模式是取 3D 图像的 3 张不同的 2D 切片进行分析（冠状面、矢状面和横断面）；最后一种模式是构建 3D 卷积神经网络（3D CNN）直接对 3D 数据进行处理。使用 2D 卷积神经网络的优点在于训练速度快，有利于模型调优；使用 3D 卷积神经网络的优点在于保留了更多的空间信息，对分类结果更加有力，但同时也提高了对机器计算能力的要求。

装
订
线

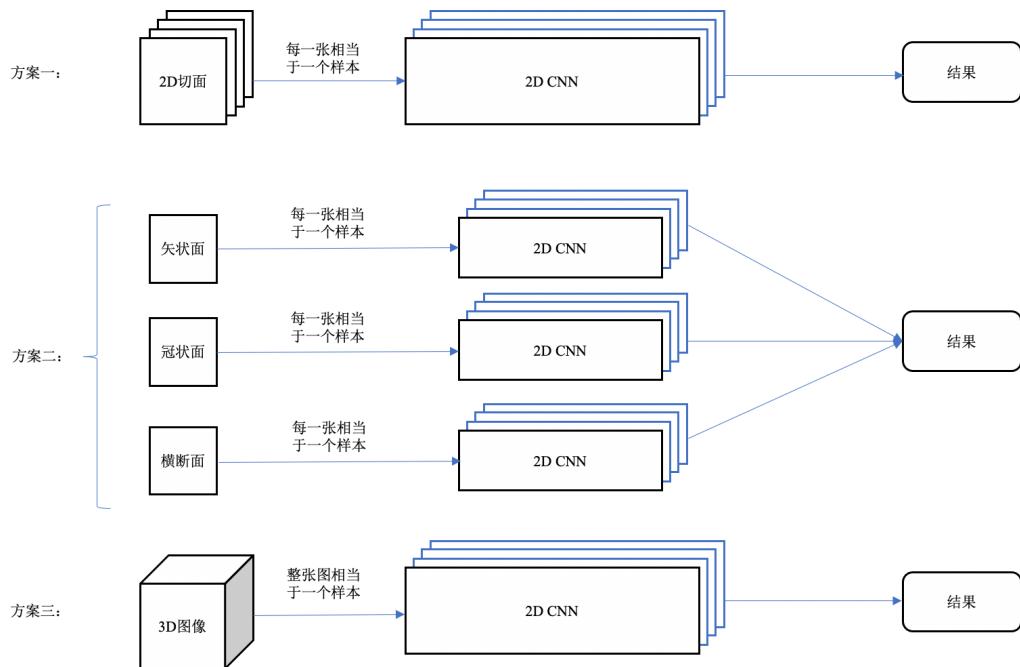


图 2.1 多种处理 CT 图像的方式

2.2 不同方法相关工作介绍

2.2.1 多任务持续学习方法

Amanda Rios 等人^[2]提出了一种使用 AC-GAN 架构的累积式闭环生成器和嵌入式分类器，该架构通过小缓冲区提供外部正则化。首先他们证明在多任务和持续学习设置中，一小部分数据集（内存缓冲区）中包含的可变性占准确性的很大一部分；其次，他们证明了在训练时使用生成器输出新图像可以对缓冲区进行采样，与固定缓冲区相比，该方法可以防止灾难性遗忘并达到更高的性能。

Sebastien Jean 等人^[3]在多任务持续学习优化方面提出使用不同的任务调度方法。通常的多任务方法对每个任务进行统一采样或与数据大小成比例，几乎无法控制性能折衷。他们使用隐式时间表，避免一项任务被过度采样情况下造成的不良影响，并且调整学习率和单个任务的梯度。他们的模型对于低资源任务表现更好，同时很大程度地减少了对高资源任务的负面影响。

Shikun Liu 等人^[4]提出了一种多任务注意力网络（MTAN），由一个包含全局功能池的共享网络以及应用于每个任务的软注意力模块组成。这些模块允许从全局功能中学习特定于任务的功能，同时允许在不同任务间共享功能。该体系结构可以端到端的有效训练，并且对多任务丢失函数中的各种加权方案也不太敏感。

Mihai Suteu 等人^[5]通过减少任务间的竞争干扰实现多任务持续学习，由于任务经常争夺模型的有限资源，从而导致整体性能降低。他们通过对任务的训练进行全面分析来解决干扰任务的问题，这些任务是通过查看它们共享参数内的梯度之间的相互作用而得出的。经验结果表明，性能良好的模型在任务梯度之间的角度上具有较低的方差。他们基于此提出了一个新颖的梯度正则项，通过强制执行接近正交的梯度来最大程度地减少任务干扰，从而减少任务间的竞争。

Jonathan Schwarz 等人^[6]为持续学习领域引入了一种概念上简单且可扩展的框架。在持续学习领域中，任务是顺序学习的。他们方法的参数数量恒定，意在保留先前遇到的任务的性能，同时加快后续问题的学习进度。这可以通过训练具有两个组件的网络来实现：能够解决先前遇到的问题的知识库，知识库连接到活动列，该活动列用于有效的学习当前任务。在学习完一项新的任务之后，活动列将被提炼到知识库中并保护起来，从而避免了灾难性遗忘。

Alex Kendall 等人^[7]观察到多任务学习系统的性能很大程度上取决于每个任务损失之间的相对权重，针对手动调整这些权重的困难性，提出了一种多任务学习的方法，该方法通过考虑每个任务的同方差不确定性来权衡多个损失函数，从而同时学习不同单位和尺度的任务。

Pengfei Liu 等人^[8]使用多任务学习框架来跨多个相关任务联合学习。基于循环神经网络，提出了三种不同的信息共享机制，以对具有特定任务和共享层的文本进行建模。整个网络在所有任务上联合训练，实验表明该模型可以在其他相关任务的帮助下提高任务的性能。

Ishan Misra 等人^[9]提出了一种使用多任务学习在 ConvNets 中学习共享表示的原则性方法，他们构造了一个新的共享单元：“十字绣”单元。这些单元结合了来自多个网络的激活，并且可以进行端到端的训练，具有十字绣单元的网络可以学习共享和特定于任务的表示的最佳组合。

O. Sener 等人^[10]认为多任务学习本质上是一个多目标问题，因为不同的任务可能会发生冲突，需要在它们之间进行权衡。这种权衡的常用方法是优化代理目标，以最小化每个任务损失的加权

线性组合。他们将多任务学习转换为多目标优化，其总体目标是找到帕累托最优解，并使用在基于梯度的多目标优化文献中开发的算法训练出了更好的模型。

David Lopez-Paz 等人^[11]提出了一种持续学习的模型，称之为梯度情景记忆（GEM），它可以减轻遗忘，同时允许将知识有益地转移到以前的任务中。

Friedemann Zenke 等人^[12]认为生物神经网络不断适应不断变化的领域，可能是通过利用复杂的分子机制同时解决许多任务。他们引入了智能突触，将这种生物复杂性的一部分带入人工神经网络。随着时间的推移，每个突触都会积累任务相关信息，并利用这些信息快速存储新记忆而不会忘记旧记忆。

T. Adel 等人^[13]提出了一种称为自适应权重持续学习（CLAW）的方法，该方法基于概率建模和变分推理，在整体持续学习性能方面表现良好并能够解决灾难性遗忘。

Cuong V Nguyen 等人^[14]开发了一种简单但通用的持续学习框架，变分持续学习（VCL）。它融合了在线变分推理（VI）和 Monte Carlo VI 神经网络，可以在复杂的持续学习环境中成功训练深度判别模型和深度生成模型。

Hanul Shin 等人^[15]提出了一种具有协作双模型架构的新型框架，称之为深度生成重放（Deep Generative Replay），由深度生成模型（generator）和任务解决模型（solver）组成。使用这两个模型可以对先前任务的训练数据进行采样，并将其与新任务的训练数据交错。

D. Rolnick 等人^[16]通过对所有过去的任务使用经验重播缓冲区，大大减少了领域中的灾难性遗忘。

Mengyao Zhai 等人^[17]提出了一种更通用的框架 Lifelong GAN，用于在不同条件图像生成设置下连续学习生成模型，通过使用知识蒸馏将学到的知识从以前的网络转移到新网络，使得在终身持续学习成为可能。

2.2.2 SGD 最速随机下降法

在我们优化一个函数 $f(x)$ 的过程中，我们要找到它的最小值，常用的方法叫做 Gradient Descent(GD)，也就是最速下降法。这种方法的表述比较简单，就是每次沿着当前位置的导数方向走一小步，在不断的进行中就可以到达最优解。

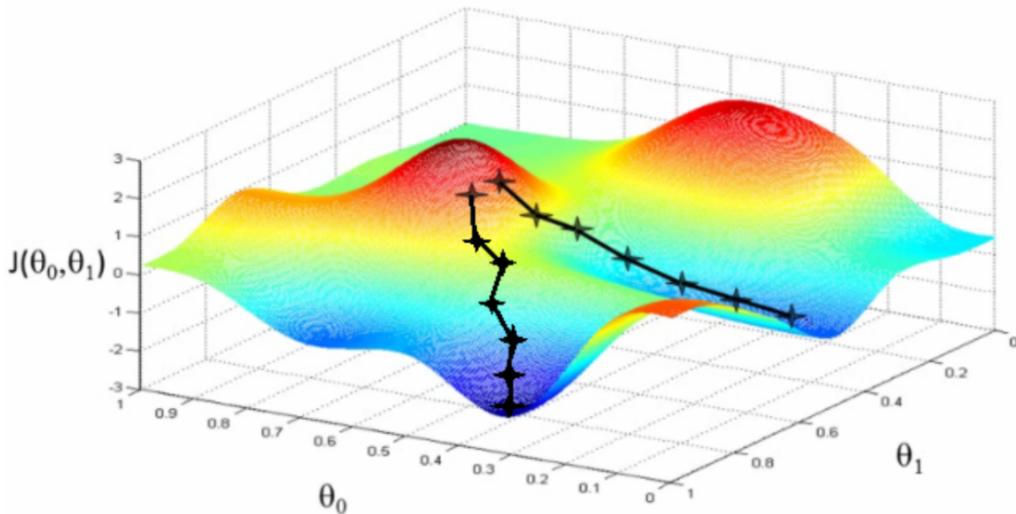


图 2.1 SGD 随机梯度下降示意图

装

订
线

如图，这种方法像是在下山一样，在进行过程中，每一步都挑选最陡峭的路走，一般来说，通过这种方法很快就能走到最低点，数学表示如下：

$$x_{t+1} = x_t - \eta_t f(x_t) \quad (2.1)$$

在这里 x_t 代表我们在第 t 步的位置， x_{t+1} 代表我们在第 $t + 1$ 步的位置， $f(x_t)$ 代表第 t 步位置的导数， η_t 代表步长。这种算法非常简单，通过反复进行上述迭代取得最优解。

但是，虽然简单优美，GD 算法也至少有两个明显的缺陷。

首先，在我们使用 GD 算法的时候，尤其是在机器学习的应用中，常常都会面临非常大的数据集。在数据集非常庞大的时候，如果要强行计算 $f(x)$ 的精确导数，这种计算量是非常庞大的，往往意味着把整个数据集扫描一遍就要花费几个小时甚至更多的时间。不仅如此，在如此巨大的时间代价下，这种方法还只能让我们前进一小步，然而一般使用 GD 算法时需要几千步甚至几万步才能达到收敛。所以在庞大的数据量下，GD 算法几乎是无法执行完成的，因此不具备很高的实用性。

其次，如果在我们使用 GD 算法时候，在不确定的情况下由于巧合陷入了鞍点，或者是比较差的局部最优点，GD 算法就无法继续执行下去了。因为 GD 算法依赖于当前位置的导数来获取下一步的位置，在这些地方，它们的导数是 0，这也就意味着当前位置的下一位位置仍然是它们本身，所以当这种带有不确定的巧合情况发生时，GD 算法会停留在这样的局部最优点，不能继续进行下去，整个算法也就无效了。

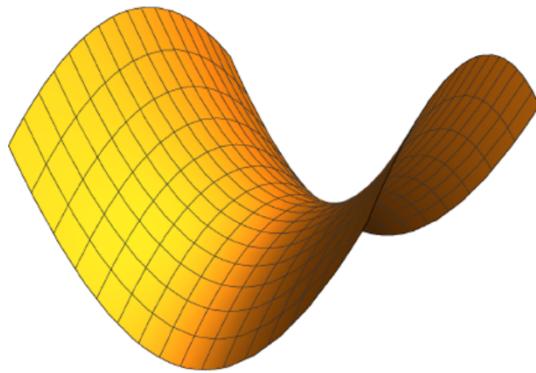


图 2.2 鞍点示意图

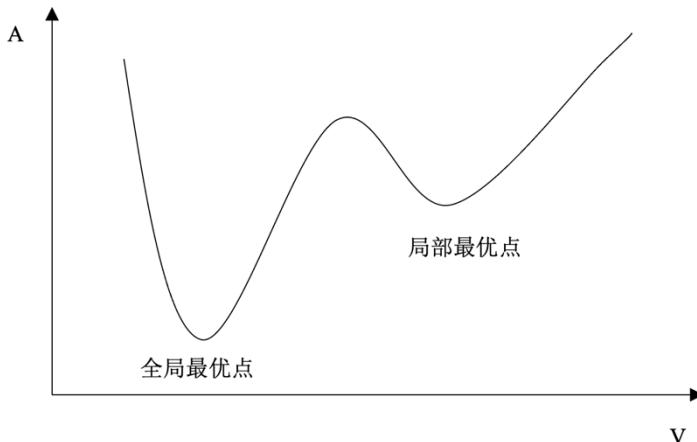
装
订
线

图 2.3 局部最优点示意图

不过上述两大缺陷可以通过同一个方法解决，那就是经典的 Stochastic Gradient Descent(SGD) 算法。SGD 算法的数学表达式和 GD 算法十分相似：

$$x_{t+1} = x_t - \eta_t g_t \quad (2.2)$$

在这里 x_t 代表我们在第 t 步的位置， x_{t+1} 代表我们在第 $t + 1$ 步的位置， η_t 代表步长， g_t 代表一种随机梯度 Stochastic Gradient，它满足 $E[g_t] = \nabla f(x_t)$ 。

也就是说，在 SGD 随机梯度下降算法中，虽然包含了一定的随机性，但是从数学期望上来看，它是等于正确的导数的。如下图所示，SGD 相比于 GD 来说，由于随机因素的影响，每一步的进行并不像 GD 那样精准，有时甚至可能会偏离正确的方向，但是由于它在期望上是等于正确的导数的，所以 SGD 的路线虽然曲折，最终也能到达 GD 所期望的收敛点。

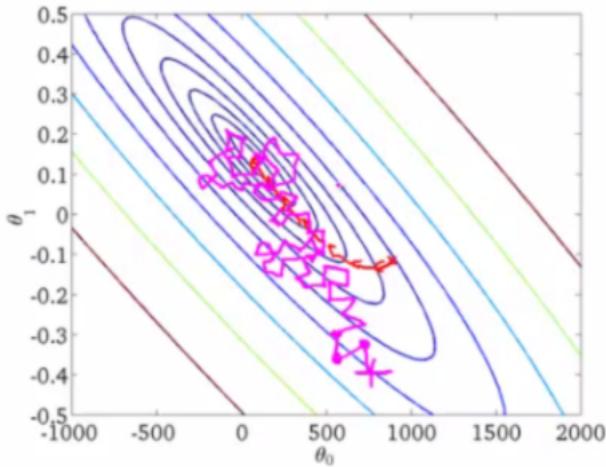


图 2.4 GD 与 SGD 路线对比示意图

装

订
线

如图所示，红色的是 GD 的路线，偏粉红色的是 SGD 的路线。

通过上图我们可以了解到，相比于 GD 来说，SGD 是需要更多的步数才能够达到收敛的，但它却能够带来很大的好处。由于 SGD 对于导数的要求非常低，可以包含大量的噪声，只需要满足期望正确，所以 SGD 在导数的计算上面非常的快。在上述提及到的数据量较大的情况下，通过利用 SGD 的随机性计算导数，要比传统的 GD 方法快成百上千倍，在这种情况下，就算进行的步数多了几倍，整体上看来也是非常高效的。

因此，SGD 算法可以完美地解决 GD 算法存在的第一个问题——计算速度太慢，时间代价太高。这也是最初人们使用 SGD 算法的主要目的。而且不需要担心 SGD 算法的导数中包含的噪声会造成负面影响，大量理论工作表明，只要噪声的程度比较合理，SGD 都能够进行很好的收敛。在不断的实践中，人们也发现很多情况下即便 GD 的训练要比 SGD 多几百倍甚至几千倍的时间，最后的结果往往是 SGD 得到的网络表现比 GD 得到的网络表现要更好。

此外，SGD 还可以逃离鞍点和局部最优解，这是 GD 算法所不能做到的。

鞍点的数学表达：

首先，我们考虑的情况是导数为 0 的点，这些点被称为 Stationary points，即稳定点。稳定点可以是（局部）最小值，（局部）最大值，也可以是鞍点。我们可以通过计算稳定点的 Hessian 矩阵 H 来判断它的类型。

1. 如果 H 是负定的，说明所有的特征值都是负的。在这种情况下，无论下一步往任何方向进行，导数都会变为负数，也就是说函数值会下降，所以这是（局部）最大值。
2. 如果 H 是正定的，说明所有的特征值都是正的。在这种情况下，无论下一步往任何方向进行，导数都会变为正数，也就是说函数值会上升，所以这是（局部）最小值。
3. 如果 H 既包含正的特征值，又包含负的特征值，那么这个稳定点就是一个鞍点，具体参考上文的图片。也就是说，在这种情况下，有些方向的函数值会上升，有些方向的函数值

会下降。

研究者通过在适当的时候给当前位置加入一个随机扰动，使得迭代能够快速的从鞍点跳出。杜克大学的 Rong Ge^[18]首先研究了这种逃离鞍点的方法，他证明通过加上随机扰动，GD 算法可以在多项式时间内对于满足 strict saddle property 的函数收敛到局部最小值。strict saddle property 是作者自创的一个概念，表示对于某个函数而言，在每一个位置 x ，都满足下面三个条件中的一个：

1. 在 x 处它的 gradient 足够大
2. 在 x 处它的 Hessian 至少包含一个负的特征值
3. x 离一个局部最小值很近

在这个假设下，所有的鞍点都是 strict saddle。如果算法进行到情况 3，那么我们就完成任务了，如果算法在情况 1，那么普通的 gradient descent 就能有效的下降函数值，难点在于情况 2，这时候如果有一个负的特征值，那说明我们至少可以找到一个有效的下降方向，也就给我们逃离鞍点带来了可能。为了寻找这个下降方向，只需要给每一步的 gradient descent 加上一个随机扰动 ε

$$x_{t+1} = x_t - \eta \nabla f(x_t) + \varepsilon \quad (2.3)$$

事实上，与人们的直觉相反，在很多优化算法里面，给梯度加上噪声非但没有阻碍收敛，反而加速了收敛，原理就是它能通过保证震荡幅度的方式避免算法在一些不好的地方停留过久。给梯度加上噪声使得 gradient descent 在速度和效果上都有很大的提升，这也是 SGD 广受欢迎的主要原因。

2.2.3 L2 正则化

在神经网络进行训练学习的过程中，可能会因为网络模型的复杂程度过高从而引起过拟合 (overfit)。过拟合是指为了得到一致假设而使假设变得过度严格的情况，往往表现为网络模型在训练集中准确率等表现效果很好，但是在测试集中的表现大打折扣，即网络的泛化能力比较差。我们在优化网络模型的过程中往往需要避免过拟合的发生，对此有很多常用的方法，比如本文提到的正则化方法，例如 L1 和 L2 正则化。

A. L2 正则化直观解释

L2 正则化公式如下，通过在原来损失函数的基础上加上权重的平方和：

$$L = E_{in} + \lambda \sum_j w_j^2 \quad (2.4)$$

其中， E_{in} 是未包含正则化项的训练样本误差， λ 是正则化参数，可调。

在网络模型中往往有很多的参数，通过对这些参数进行限制，使得参数在一定约束范围内无法过多或过大，从而控制网络模型的复杂程度，这是正则化方法的主要目的。例如，在使用多项式模型时，如果使用 10 阶多项式，多项式的阶数过很容易导致网络模型的复杂程度过高，这往往就容易造成过拟合的发生。为了避免这种情况，可以通过限制多项式模型中高阶部分的权重，比如将其高阶权重 w 设置为 0，从而完成多项式模型中高阶到低阶的形式转换。

通过限制高阶权重 w 的个数来避免过拟合的发生是一种非常直观的方法，但它也有一定的缺

陷，因为这种问题的复杂程度非常高，属于一种 NP-hard 问题，如何对其进行有效的求解往往是一个非常困难的问题。在这种情况下，往往通过寻找更宽松的限制条件来获取一种更具有般性的方法，公式如下：

$$\sum_j w_j^2 \leq C \quad (2.5)$$

下图说明如何在限定条件下，对 E_{in} 进行最小化的优化：

装
订
线

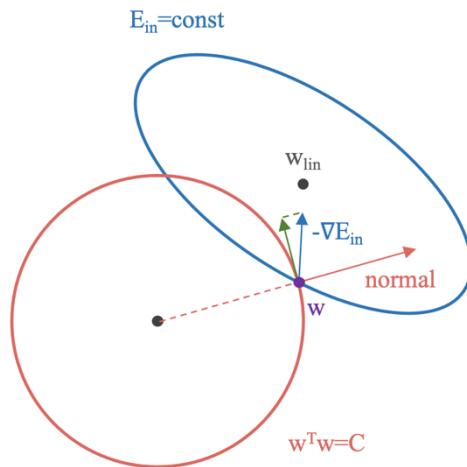


图 2.5 L2 限定条件下 E_{in} 最小化优化示意图

如上图所示，红色圆形区域代表 w 的限定条件区域，蓝色椭圆区域代表最小化 E_{in} 区域。如果对于模型中的参数没有限定条件，根据梯度下降算法，点 w 在蓝色椭圆区域内会一直沿着 w 梯度的反方向前进，直到沿着这个方向到达全局最优值 w_{lin} 。例如空间中有一紫色点 w ，在这种情况下， w 会沿着 $-\nabla E_{in}$ 的方向移动，如图中蓝色箭头所示。但是，由于存在限定条件， w 不能离开红色圆形区域，最多只能位于圆上边缘位置，沿着切线方向运动。 w 的方向如图中红色箭头所示， w 的运动方向如图中绿色箭头所示。

那么在存在限定条件的情况下， w 最终会在什么位置取得最优解呢？也就是说在满足限定条件的基础上，尽量的让 E_{in} 最小。

w 是沿着圆的切线方向运动，如上图绿色箭头所示。运动方向与 w 的方向（红色箭头方向）垂直。运动过程中，根据向量知识，只要 $-\nabla E_{in}$ 与运行方向有夹角，不垂直，则表明 $-\nabla E_{in}$ 仍会在 w 的切线方向上产生分量，那么 w 就会继续运动，寻找下一步的最优解。只有当 $-\nabla E_{in}$ 与 w 的切线方向垂直时， $-\nabla E_{in}$ 在 w 的切线方向才没有分量，这时候 w 才会停止更新，到达最接近 w_{lin} 的位置，且同时满足限定条件。

图 2.6 $-\nabla E_{in}$ 与 w 方向说明示意图

$-\nabla E_{in}$ 与 w 的切线方向垂直, 即 $-\nabla E_{in}$ 与 w 的方向平行。如上图所示, 蓝色箭头和红色箭头互相平行。这样, 根据平行关系得到:

$$-\nabla E_{in} + \lambda w = 0 \quad (2.6)$$

由于 λ 代表常数系数, 故上式相当于:

$$\nabla E_{in} + \lambda w = 0 \quad (2.7)$$

这样, 我们就把优化目标和限定条件整合在一个式子中了。也就是说只要在优化 E_{in} 的过程中满足上式, 就能实现正则化目标。

根据最优化算法的思想中: 梯度为 0 的时候, 函数取得最优值。已知 ∇E_{in} 是 E_{in} 的梯度, 观察上式, 将 λw 可以看成是 $\frac{1}{2} \lambda w^2$ 的梯度:

$$\frac{\partial}{\partial w} \left(\frac{1}{2} \lambda w^2 \right) = \lambda w \quad (2.8)$$

这样, 我们根据平行关系求得的公式, 构造一个全新的损失函数:

$$E_{aug} = E_{in} + \frac{1}{2} \lambda w^2 \quad (2.9)$$

之所以这样定义, 是因为对 E_{aug} 求导, 正好得到上面所求的平行关系式。上式中等式右边第二项就是 L2 正则化项。

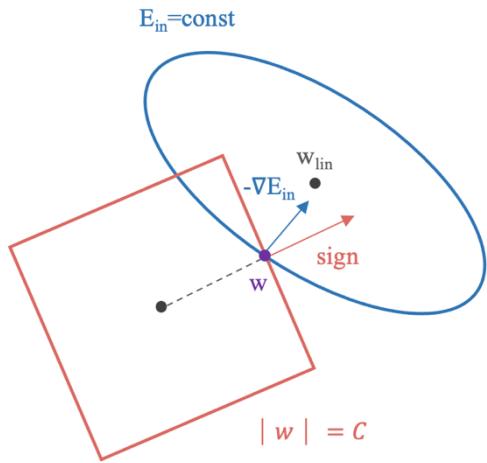
这样, 我们从图像化的角度, 分析了 L2 正则化的物理意义, 解释了带 L2 正则化项的损失函数是如何推导而来的。

B. L1 正则化直观解释

L1 正则化的公式如下, 通过在原来损失函数基础上加上权重参数的绝对值:

$$L = E_{in} + \lambda \sum_j |w_j| \quad (2.10)$$

用一张图来说明如何在 L1 正则化下, 对 E_{in} 进行最小化的优化。

装
订
线图 2.7 L1 限定条件下 E_{in} 最小化优化示意图

E_{in} 优化算法不变, L1 正则化限定了 w 的有效区域是一个正方形, 且满足 $|w| < C$ 。空间中的点 w 沿着 $-\nabla E_{in}$ 的方向移动。但是, w 不能离开红色正方形区域, 最多只能位于正方形边缘位置, 其推导过程与 L2 类似, 此处不再赘述。

C. L1 和 L2 解的稀疏性

介绍完 L1 和 L2 正则化的物理解释和数学推导之后, 我们再来看看它们解的分布性。

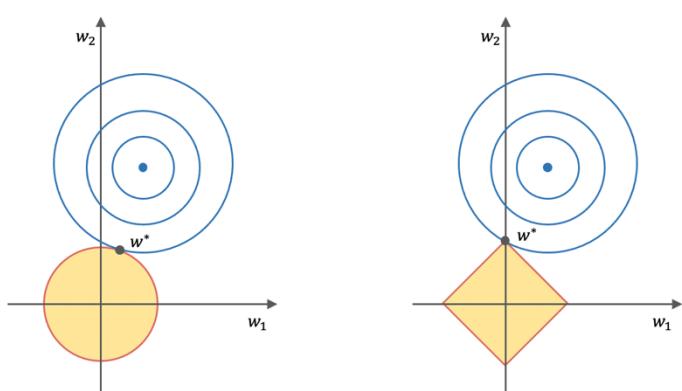


图 2.8 二维情况下 L1 和 L2 解的分布性示意图

以二维情况讨论, 上图左边是 L2 正则化, 右边是 L1 正则化。从另一个方面来看, 满足正则化条件, 实际上是求解蓝色区域到黄色区域的交点, 即同时满足限定条件和 E_{in} 最小化。对于 L2 来说, 限定条件是圆, 这样, 得到的解 w_1 或 w_2 为 0 的概率很小, 很大概率是非零的。

对于 L1 来说，限定区域是正方形，方形与蓝色区域相交的交点是顶点的概率很大，这从视觉和常识上来看是很容易理解的。也就是说，方形的凸点会更接近 E_{in} 最优解对应的 w_{lin} 位置，而凸点处必有 w_1 或 w_2 为 0。这样，得到的解 w_1 或 w_2 为 0 的概率就很大了。所以，L1 正则化的解具有稀疏性。

扩展到高维的情况下，同样的道理，L2 的限定区域是平滑的，与中心点等距；而 L1 的限定区域是包含凸点的，尖锐的。这些凸点更接近 E_{in} 的最优解位置，而在这些凸点上，很多 w_j 为 0。

D. L2 正则化和过拟合的关系

一般来说，为了能够在一定程度上避免过拟合现象的发生，会倾向于构造参数偏小的网络模型。参数比较小的模型相对来说比较简单，并且对不同数据集的适应性更好。举一个简单的例子，设想对于一个线性回归方程来说，如果方程中的参数很大，那么即使是数据发生了微小变化，最终体现在结果上的影响也会变得很大；与之相反，如果方程中的参数很小，就算是数据发生了比较大的变化，最终体现在结果上的影响也会很小，这样的模型具有更好的抗扰动能力。

L2 正则化优化参数的方式如下：

Andrew Ng^[19]提出了一种在深度学习训练中的参数表示方法。假设要求解的参数为 θ ，假设函数为 $h_\theta(x)$ 。在线性回归中，一般使用平方差损失函数。单个样本的平方差是 $(h_\theta(x) - y)^2$ ，在考虑到所有样本的情况时，损失函数变为对每个样本的平方差求和。假设有 m 个样本，线性回归的代价函数如下：

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2 \quad (2.11)$$

首先，对于单个样本中的某个函数 θ_j 求导，结果如下：

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} (h_\theta(x) - y) \frac{\partial}{\partial \theta_j} h_\theta(x) \quad (2.12)$$

在这里， $h_\theta(x)$ 的表达式为 $h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \dots + \theta_n x_n$ 。单个样本对某个参数 θ_j 求导时， $\frac{\partial}{\partial \theta_j} h_\theta(x) = x_j$ 。最终上式结果如下：

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} (h_\theta(x) - y) x_j \quad (2.13)$$

接下来，对于全部样本的情况来说，把每个样本对 θ_j 的导数求和，所得到的结果如下：

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \quad (2.14)$$

梯度下降算法会沿梯度的反方向更新参数，我们对上式的结果乘以学习率 α 并取反，所得结果为迭代计算参数 θ_j 的公式，如下：

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \quad (2.15)$$

其中 α 是学习率（learning rate）。上式的迭代公式中还没有添加 L2 正则化项，如果在上式中添加 L2 正则化项，那么迭代公式如下：

$$\theta_j := \theta_j \left(1 - \alpha \frac{\lambda}{m}\right) - \alpha \frac{1}{m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \quad (2.16)$$

在上式中， λ 为正则化参数。对于添加 L2 正则化的迭代公式来说，在每次迭代之前 θ_j 会先乘以一个因子 $(1 - \alpha \frac{\lambda}{m})$ ，因为这个因子是小于 1 的，所以 θ_j 在迭代过程中会不断减小，故从总体上而言，参数 θ 是不断减小的，这也就达到了 L2 正则化的目的。

E. 正则化参数 λ

正则化是结构风险最小化的一种策略实现，能够有效降低过拟合。损失函数实际上包含了两个方面：一个是训练样本误差，一个是正则化项。其中，参数 λ 起到了权衡的作用，如下图所示。

装
订
线

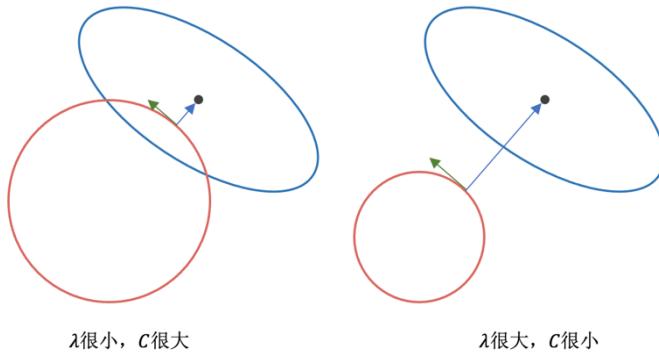


图 2.9 参数 λ 的权衡作用示意图

以 L2 正则化为例，若 λ 很小，对应上文中的 C 值就很大。这时候，红色圆形区域很大，能够让 w 更接近 E_{in} 最优解的位置。若 λ 近似为 0，相当于圆形区域覆盖了最优解的位置，这时候会导致正则化失效，容易造成过拟合。相反，若 λ 很大，对应上文中的 C 值就很小。这时圆形区域很小， w 离 E_{in} 最优解的位置较远。 w 被限制在一个很小的区域内变化， w 普遍较小且接近 0，起到了正则化的效果。但是， λ 过大容易造成欠拟合。欠拟合和过拟合是两种对立的状态。

3 基于 EWC 弹性权重合并的多任务肺结节 CT 图像分类算法

3.1 网络框架介绍

在 EWC (elastic weight consolidation) 算法对比实验网络结构选取方面，本文构建了一个基于 ResNet 的 2D 卷积神经网络。ResNet 可以说是目前应用最为广泛的 CNN 特征提取网络，它由何恺明、孙剑、任少卿等人^[20]在 2015 年首次提出，通过残差学习的方式，将网络层级扩展到一个很深的层次，达到深度学习的目的。

由于卷积神经网络可以提取到不同层次的信息，网络的层数越多，也就意味着提取到不同层级的信息越丰富。浅层的网络可能会提取一些局部的信息，而深层的网络则会提取更为抽象的信息，更多的包含一些语义信息，因此深层的网络从常理上来说应该优于浅层网络。

但是如果仅仅只是简单地增加网络层数，就会导致梯度在传递的过程中过小或过大，引发梯度消失或者梯度爆炸的问题。解决梯度消失或梯度爆炸问题的一个较好的解决方案是加入正则初始化以及批标准化 (Batch Normalization, BN)。然而网络层次较深也导致了网络更加难以拟合数据，从理论上来讲在一个训练集上，深层网络不应该比千层网络性能表现更差，因为只要将多出来的层全部优化为恒等映射，即部分的层表示为 $H(x) = x$ ，深层网络就会等价于浅层网络。但是在实际的实验过程当中，深层网络表现出的性能要远远不如浅层网络，这就是深度学习中经常提到的退化问题。

为了解决这样的退化问题，令深层网络结构具有浅层网络一样的信息传递能力，而不被限制权重更新的自由度，作者提出了残差这样一个概念，残差的结构如图所示。

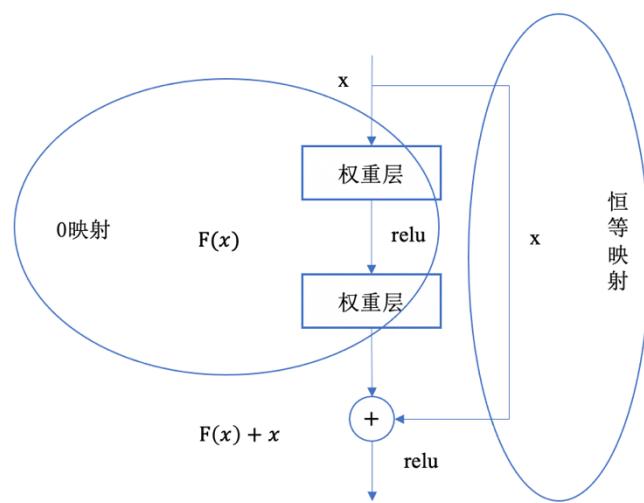


图 3.1 残差块结构示意图

令 x 表示原始输入， $F(x)$ 表示普通神经网络的输出， $H(x)$ 表示期望神经网络的输出。ReLU 表示激活函数。假设没有快捷连接，则此网络就是一个简单的深层网络，即上图的左侧部分。如果期望此深层网络结构具有浅层网络一样的信息传递能力而不发生退化，则需要在一些层具有恒等

映射：

$$H(x) = F(x) = x \quad (3.1)$$

然而这在训练过程中是很难实现的。原因在于假设 x 为 10, $H(x)$ 从 10 学习到 10.1 那么全中的学习过程将表示为：

$$H(10) = F(10) = 10.1 \quad (3.2)$$

权重的变化率为：

$$\frac{10.1 - 10}{10} = 1\% \quad (3.3)$$

可以看到权重的更新效果是很小的。但是在网络加入残差结构以后，仍然用之前的表示计算权重的学习过程，权重的学习过程将表示为：

$$H(x) = F(x) + x \quad (3.4)$$

$$H(10) = F(10) + 10 = 10.1 \quad (3.5)$$

$$F(10) = 0.1 \quad (3.6)$$

权重的变化率为：

$$\frac{0.1 - 10}{10} \approx 100\% \quad (3.7)$$

这种大的权重变化率对于权值的调节有很大的好处。因为希望将 $F(x)$ 优化为 0，因此上图的左侧部分又称为 0 映射，右侧部分称为恒等映射。

ResNet 在增加网络深度的同时，没有增加网络的复杂度，并且它的快捷连接使得网络更容易优化。这些优秀的特性也使得其应用广泛，在拟合各种类型的数据方面具有良好的效果。

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112			7×7, 64, stride 2		
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1			average pool, 1000-d fc, softmax		
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

图 3.2 经典的 Resnet 结构示意图

在另一篇文章中，何恺明等人^[21]通过调整激活函数（ReLU）和批标准化（BN）的位置，将激活项区分为预激活项和后激活项，对比二者的性能。调整后的结构如图所示。实验证明了将 ReLU 与 BN 都放置在预激活部分将会取得更好的结果，因此本文也采用这种调整后的网络结构来进行

实验。

装

订

线

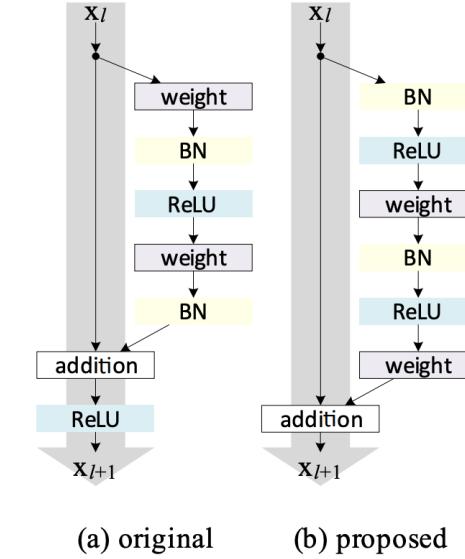


图 3.3 预激活项放置 BN 和 ReLU 的残差块结构示意图

3.2 EWC 优化算法介绍

解决灾难性遗忘：弹性权重合并算法 EWC (elastic weight consolidation)。

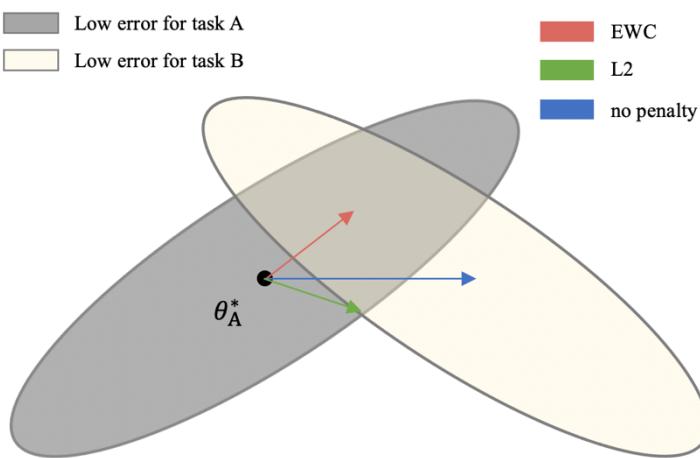


图 3.1 EWC 算法说明图

在学习完任务 A 后，EWC 会对参数进行一定程度的约束，将其约束在一个对任务 A 表现影响很小的低误差区域内（图中灰色椭圆部分），从而在后续学习任务 B 的过程当中保护任务 A 的性

能。

EWC 算法通过对参数的有效限制，能够在多任务持续学习 A、B 的情况下，同时保持在 A、B 两个任务上的优秀表现，避免了学习任务 B 时对任务 A 的灾难性遗忘，同时也对任务 B 保留了有效充分的学习空间。在第一个任务 A 学习完成后，参数为 θ_A^* ，如果使用梯度下降方式（如图中蓝色箭头所示），会对任务 B 进行有效的学习，网络在任务 B 上的损失会被降到最低，但如图所示，这种方式会离开任务 A 的低误差区域（图中灰色椭圆区域），这会导致网络对先前任务 A 所学习的内容产生灾难性遗忘。如果使用 L2 正则化方式（如图中绿色箭头所示），由于我们学习完任务 A 后，为了记住任务 A，对于权重施加的约束过于严格，这会导致在任务 B 上的学习空间被大大压缩，最终无法在任务 B 上进行有效的学习。EWC 算法（如图中红色箭头所示）通过显式计算参数权重的重要性，能够保护那些对任务 A 影响较大的权重，而对那些对任务 A 影响较小的权重保有高度的灵活性，从而在保护任务 A 的同时完成对任务 B 的有效学习。

实现人工通用智能要求智能体能够学习和记忆许多不同的任务，这在现实世界中面临着许多的困难：任务的顺序可能没有被明确标记，任务可能被不可预知的切换，任何单个任务都不可能长时间的重复。因此，至关重要的是人工智能必须获得一种持续学习的能力：即在不忘记如何执行先前训练的任务的情况下学习连续任务的能力。

与人工神经网络形成鲜明对比的是，人类和其他动物能够进行持续的学习。有研究证据表明，哺乳动物的大脑可能通过保护新皮层回路中先前获得的知识来避免灾难性遗忘。当老鼠获得一项新技能时，兴奋性突触的比例会增强；这表现为神经元个体树突棘体积的增加。关键的是，尽管随后学习了其他的任务，这些增大的树突棘依然存在，这就解释了对相应技能的记忆能力。当这些树突棘被选择性地抹去时，相应的技能就被遗忘了。这为保护这些强化突触的神经机制对保持任务性能提供了至关重要的因果证据。这些实验结果和神经生物学模型表明，在新大脑皮层中持续的学习依赖于特定任务的突触巩固，通过改变可塑性较低的突触的比例，从而在长时间内稳定的进行知识编码。因此，知识是通过对一部分可塑性较低的突触进行持久编码，从而达到在很长的时间尺度上的稳定性。

James Krikpatrick 等人^[22]以此为灵感开发了一种人工神经网络突触整合的算法，命名为 elastic weight consolidation (EWC)。该算法根据某些权重对先前看到的任务的重要性来降低学习率，减慢学习速度从而避免后续学习对先前任务重要权重的过大影响。EWC 能够在监督学习和强化学习问题中使用，在不忘记旧任务的情况下连续训练多个任务。

一个深度神经网络由多层线性投影组成。一个任务的学习过程由调整线性投影中的权重和偏移组成，通过对其进行不断的调整来优化神经网络的性能。过参数化使得任务 B 的参数 θ_B 和任务 A 的参数 θ_A 很相似。在学习任务 B 的时候，EWC 通过把参数限制在以 θ_A 为中心的一个小范围的误差内，保护了任务 A 的性能。这种对于参数的限制执行的效果类似于一种惩罚，之前任务的参数像是被弹簧锚定住了，因此得名 elastic。更重要的是，并不是所有的参数都具有相同的弹簧刚度，对于任务更重要的参数应该具有更大的约束，使得它们更加难以被改变，以此来保证对之前任务的记忆。

为了找到在任务中最重要的权重，从概率论观点考虑神经网络训练，优化参数相当于在数据

D 中寻找最重要的值。使用贝叶斯规则通过参数先验概率 $p(\theta)$ 和数据概率 $p(D|\theta)$ 计算条件概率 $p(\theta|D)$:

$$\log p(\theta|D) = \log p(D|\theta) + \log p(\theta) - \log p(D) \quad (3.8)$$

注意，在这里给定参数的数据对数概率 $\log p(D|\theta)$ 可以简单的定义为负损失函数 $-L(\theta)$ 。假设数据被分为两个部分，一个定义为任务 A，另一个定义为任务 B。重写等式:

$$\log p(\theta|D) = \log p(D_B|\theta) + \log p(\theta|D_A) - \log p(D_B) \quad (3.9)$$

等式左边仍然描述了整个数据集参数的后验概率，而等式右边只依赖于任务 B 的损失函数 $\log p(D_B|\theta)$ 。

因此，关于任务 A 的所有信息必须被吸收进后验概率分布 $p(\theta|D_A)$ 中。后验概率必须包含任务 A 的参数重要性信息，这也是 EWC 执行的关键。但真实的后验概率是很棘手的，因此使用 Laplace 近似。将后验概率作为由参数 θ_A 给出的高斯分布和 Fisher 信息对角矩阵 F 的对角线精度。

F 有三个关键性质：

1. 它相当于最小损失函数的二阶导数
2. 它可以单独的计算一阶导数，因此可应用于大规模计算
3. 可以保证是半正定矩阵

这种方法类似于期望传播，每个子任务被看作是一个后验的因素。近似地认为 EWC 中最小化的函数 L 为：

$$L(\theta) = L_B(\theta) + \sum_i \frac{\lambda}{2} F_i (\theta_i - \theta_{A,i}^*)^2 \quad (3.10)$$

其中 $L_B(\theta)$ 是任务 B 的损失函数， λ 代表相比于新任务来说旧任务的重要程度， i 代表每个参数。

当有第三个任务任务 C 时，EWC 也试图确保参数同样适用于任务 A 和任务 B。

装
订
线

贝叶斯视角下 EWC 的理论推导：

机器学习在贝叶斯的视角下，目的都是为了最大化后验概率 $p(\theta|D)$ ，但是因为后验概率难以估计（不可能尝试所有的 θ ），因此用贝叶斯公式 $p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}$ 转化为最大化 $p(D|\theta)$ ，然后取一个 \log 转成似然函数用点估计方法做极大似然估计来达成目的。

在这里，我们把 D 分为 D_A 和 D_B ，因此：

$$\begin{aligned} \log p(\theta|D) &= \log p(D_A, D_B|\theta) + \log p(\theta) - \log p(D_A, D_B) \\ &= \log p(D_A|\theta) + \log p(D_B|\theta) + \log p(\theta) - \log p(D_A) - \log p(D_B) \\ &= \log p(D_B|\theta) + \log p(\theta|D_A) - \log p(D_B) \end{aligned} \quad (3.11)$$

因此，我们的目标在贝叶斯视角下来看，只是一边最大化任务 B 的似然概率 $\log p(D_B|\theta)$ ，一边保持任务 A 的后验概率 $\log p(\theta|D_A)$ 的最大化。最大化任务 B 的似然概率推导到后面任务的损失函数，下面推导保持任务 A 的后验概率的最大化的损失函数。

先进行拉普拉斯近似，假设 $\log p(\theta|D_A)$ 关于参数服从高斯分布，均值为 μ ，方差为 σ ，都是长度为 N 的一维向量。

把 $\log p(\theta|D_A)$ 写作一个关于 θ 的函数：

$$p(\theta|D_A) = \frac{1}{Z} f(\theta) \quad (3.12)$$

其中， Z 为归一化系数（概率的积分必须为 1）。

泰勒展开公式：

$$F(x) = \frac{F(x_0)}{0!} + \frac{F'(x_0)}{1!} \cdot (x - x_0) + \frac{F''(x_0)}{2!} \cdot (x - x_0)^2 + R_n(x) \quad (3.13)$$

$$F(x) = \frac{F(x_0)}{0!} + \frac{J_F(x_0)}{1!} \cdot (x - x_0) + (x - x_0)^T \cdot \frac{H_F(x_0)}{2!} \cdot (x - x_0) + R_n(x) \quad (3.14)$$

然后，对它的似然函数进行泰勒展开：

$$\ln f(\theta) = \ln f(\theta^*) + J_{\ln f}(\theta^*) \cdot (\theta - \theta^*) + \frac{1}{2} \cdot (\theta - \theta^*)^T \cdot H_{\ln f}(\theta^*) \cdot (\theta - \theta^*) + R_n(\theta) \quad (3.15)$$

因为在任务 A 上，模型优化到了最优，损失函数很小，一阶导很小，近似为 0，并且三阶及以上的项，在 θ 变化不大的情况下都可以近似为 0，因此只保留二阶项和零阶项。

$$\ln f(\theta) \approx \ln f(\theta^*) - \frac{1}{2} \cdot (\theta - \theta^*)^T \cdot F_{\ln f}(\theta^*) \cdot (\theta - \theta^*) \quad (3.16)$$

$$F_{\ln f} = -H_{\ln f}(\theta^*) \quad (3.17)$$

这里，把 Hessian 矩阵取负，就得到了费雪信息矩阵。

对公式取对数，得：

$$f(\theta) \approx \frac{1}{Z} f(\theta^*) \cdot e^{-\frac{1}{2}(\theta-\theta^*)^T \cdot F_{\ln f}(\theta^*) \cdot (\theta-\theta^*)} \quad (3.18)$$

因此， $p(\theta|D_A)$ 是服从一个 $\mu = \theta^*$, $\frac{1}{\sigma^2} = F_{\ln f}(\theta^*)$ 的高斯分布，而最大化 $p(\theta|D_A)$ 就等价于最大化 $\ln f(\theta)$ 。

因为 $\ln f(\theta^*)$ 是个常数（ θ^* 是给定的），因此，最大化 $p(\theta|D_A)$ 最终等价于最大化 $-\frac{1}{2}(\theta - \theta^*)^T \cdot F_{\ln f}(\theta^*) \cdot (\theta - \theta^*)$ 。

此外， θ^* 都是长度为 N 的一维向量， $F_{\ln f}(\theta^*)$ 为 $N * N$ 大小，计算的时间和空间复杂度难以接受，于是和优化的领域一样，取了费雪信息矩阵的对角线，这等于假设参数之间相互独立没有影响，最终导出损失函数：

$$\mathcal{L}_{old} = \frac{\lambda}{2} \sum_i F_{\ln f}(\theta^*)_{i,i} (\theta_i - \theta_i^*)^2 \quad (3.19)$$

$$F_{\ln f}(\theta^*)_{i,j} = \frac{\partial^2 \mathcal{L}(d, \theta)}{\partial \theta_i \partial \theta_j} \quad (3.20)$$

3.3 网络的训练和测试

为了对多任务持续学习的效果进行直观有效的测试，本文设计了三种不同的分类任务记为 Task A、Task B、Task C。分别使用不同多任务持续学习方法的网络结构（EWC、L2、SGD）顺序进行三种任务 A、B、C 的学习，在学习任务 A 的过程中，对任务 A 的各项指标进行测试；在学习

完任务 A，学习任务 B 的过程中，对任务 A、B 的各项指标都进行测试；在学习完任务 A、B，学习任务 C 的过程中，对任务 A、B、C 的各项指标都进行测试。并对测试结果绘制表格和折线图，以此在多种维度（任务类型，训练过程）上直观的发现三种方法的优劣与不同，具体内容将在实验部分展开描述。

装
订
线

4 实验结果和分析

4.1 LIDC 数据集

4.1.1 数据来源

数据集采用为 LIDC-IDRI (The Lung Image Database Consortium)，该数据集由胸部医学图像文件（如 CT、X 光片）和对应的诊断结果病变标注组成。该数据是由美国国家癌症研究所（National Cancer Institute）发起收集的，目的是为了研究高危人群早期癌症检测。

该数据集中，共收录了 1018 个研究实例。对于每个实例中的图像，都由 4 位经验丰富的放射科医师进行两阶段的诊断标注。在第一阶段，每位医师分别独立诊断并标注病患位置，其中会标注三种类别：

1. $\geq 3\text{mm}$ 的结节
2. $< 3\text{mm}$ 的结节
3. $\geq 3\text{mm}$ 的非结节

在随后的第二阶段中，每位医师都分别独立的复审其他三位医师的标注，并给出自己最终的诊断结果。这样的两阶段标注可以在避免强制共识的前提下，尽可能完整的标注所有的结果。

4.1.2 解析结果

A. 图像矩阵像素信息

模块处理的数据位 $slicer * rows * cols$ 大小的三维矩阵 D。D 中第 z 个切片 y 行 x 列的元素对应的位置为： $(z * rows * cols + y * cols + x) * \text{sizeof}(\text{data_type})$ 。其中 rows 表示图像对应的行数，cols 表示图像的列数，默认均为 512；data_type 代表数据类型，默认为 short。

eg: 对于病例 LIDC-IDRI-0001，即为 $133 * 512 * 512$ 的矩阵，一共 133 张切片，每张大小 $512 * 512$ ，依次按顺序存入二进制文件，每个像素大小为 2 字节（对应 short 类型）。

B. 结节区域类型标注信息

第一行：slicer rows cols data_type pixel_space_x pixel_space_y slice_thickness

slicer: 切片个数；

rows: 矩阵行数，默认 512；

cols: 矩阵列数，默认 512；

data_type: 数据类型标签。为以下枚举类型中的一种（默认 SHORT_TYPE, 4）：enum DATA_TYPE {CHAR_TYPE, UCHAR_TYPE, INT_TYPE, UINT_TYPE, SHORT_TYPE, USHORT_TYPE, FLOAT_TYPE, DOUBLE_TYPE};

pixel_space_x : x 线列扫描步长，单位：毫米；

pixel_space_y : y 线行扫描步长，单位：毫米；

slice_thickness : z 轴扫描步长(即切片厚度)，单位：毫米。

其他行：type num x1 y1 z1 x2 y2 z2 ... xi yi zi ... xn yn zn

type: “1”表示“nodules”，“2”表示“small_nodules”，“3”表示“non_nodules”；

num: 该行 x, y, z 数字的个数（由于一个点有三个坐标，所以 num 为 3 的倍数）；

X_i, Y_i, Z_i : 该肺结节第 i 个点的空间坐标, Z_i 为切片序号。

C. XML 标注信息说明

XML 标注信息结构说明如下图所示:

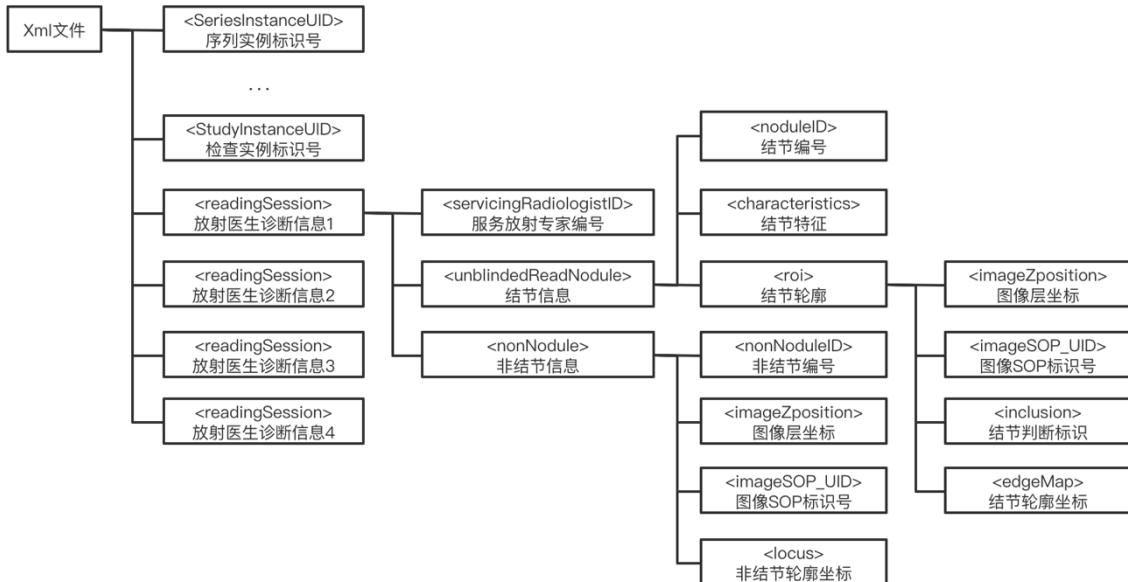


图 4.1 XML 标注信息结构说明图

D. 数据信息字段说明

数据信息中各字段说明如下:

表 4.1 数据信息字段说明表

字段名	类型	大小	说明
ExpertID	文本	50	专家编号, 4 位放射学专家定义为 A~D
NoduleID	文本	255	结节编号
Type	文本	255	病变分类: 大结节、小结节及非结节
Subtlety	数字	整型	精细度, 1~5
InternalStructure	数字	整型	内部结构, 1~4
Calcification	数字	整型	钙化, 1~6
Sphericity	数字	整型	球形度, 1~5
Margin	数字	整型	边缘, 1~5
Lobulation	数字	整型	分叶征, 1~5
Spiculation	数字	整型	毛刺征, 1~5
Texture	数字	整型	纹理, 1~6

续表 4.1

字段名	类型	大小	说明
Malignancy	数字	整型	恶性度, 1~5
imageSOP_UID	文本	255	CT 号
xCoord	数字	整型	X 坐标
yCoord	数字	整型	Y 坐标
fileCoord	文本	255	保存大结节轮廓点坐标的文件名

4.2 数据预处理和数据增强

4.2.1 肺结节图像数据预处理

在进行图像分类任务时, 原始的图像数据中所包含的信息往往过于复杂, 其中许多信息与图像分析任务是无关的, 因此, 需要对图像进行预处理, 从图像中提出关键性的有助于分类任务的信息, 并且对图像数据进行简化, 从而让图像信息更具有针对性和可靠性。

A. DICOM 和 NIFTI

原始的医学数据一般提供的数据格式为 2D 的 DICOM (Digital Imaging and Communications in Medicine, 医学数字成像和通信标准) 格式, 文件标识符一般为 “.dcm” 。DICOM 是医学图像与其相关信息的国际标准 (ISO 12052), 它存储原始图像数据的形式是存储 3D 图像的多个 2D 片段。DICOM 文件是由数据头和图像数据共同组成的, 数据头包括了病人编号、图像帧数以及分辨率等附加描述信息。由于医学图像的扫描一般是高维的 (通常是 3D 或 4D), 即使是对一个单一的图像获取, 也会产生许多 DICOM 文件。DICOM 和 NIFTI (Neuroimaging Informatics Technology Initiative, 神经影像信息技术) 之间最主要的区别在于 NIFTI 中的原始图像数据是以 3D 的形式储存的, 文件标识符一般为 “.nii” 。处理一个单个的 NIFTI 文件, 与处理上百个 DICOM 文件相比要轻松得多, 这也是 NIFTI 在研究工作中比 DICOM 更受欢迎的主要原因。遗憾的是, LIDC 数据集只提供了 DICOM 原始图像文件, 并没有提供 NIFTI 3D/4D 图像文件, 因此本文选择 DICOM 图像作为研究对象。

B. 生成肺结节 Mask 掩膜图像

提取数据文件中的坐标和直径, 以坐标为中心, 直径为长, 生成正方体区域, 最后输出成 Mask 图像文件。

掩膜是一种二进制图像, 内部由 0 和 1 两种值所组成。若某区域的掩膜值为 1, 则代表这部分区域是有效的, 被包括在后续的计算中。同理, 若某区域的掩膜值为 0, 则代表这部分区域是无效的, 不被包括在后续的计算中。

下图表示原始图在对应像素与掩膜图进行与运算后, 所得到的结果图示例:

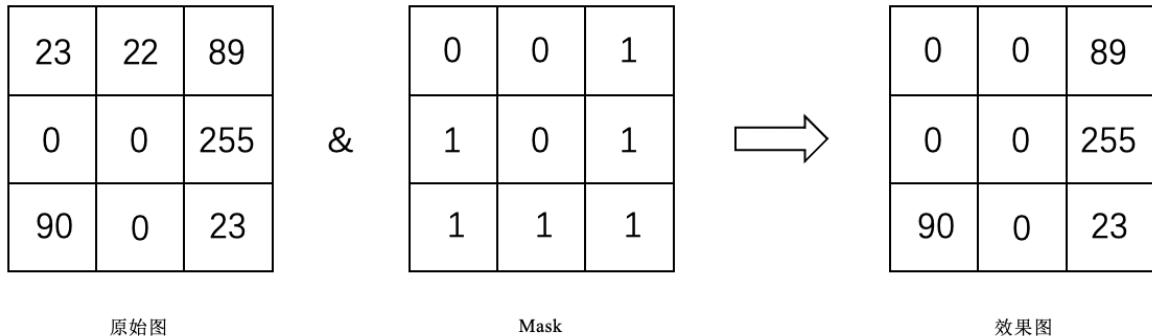


图 4.2 Mask 掩膜运算说明图

C. 图像去噪

设置窗宽窗位 (-1000, 600) 去除 CT 图像中的噪声，例如骨头的亮点，CT 床的金属线等，并将图像归一化到 (0, 1)。

CT 的识别能力远远强于人类的肉眼。必须进行分段观察，才能使 CT 的优点反映出来。观察的 CT 值范围称为窗宽，窗宽范围内观察的中心 CT 值即为窗位。

图像归一化不会改变图像本身的信息存储，并且可以将取值范围转化到 0~1 之间，对于后续神经网络处理有很大好处。对于图像归一化处理，采用最大最小值归一化方法，公式如下：

$$norm = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (4.1)$$

其中 x_i 表示图像像素点值， $\min(x)$, $\max(x)$ 分别表示图像像素的最大与最小值

装
订
线

$[[[4 \ 24 \ 19]$	$[[[0.01568628 \ 0.09411766 \ 0.07450981]$
$[31 \ 48 \ 44]$	$[0.12156864 \ 0.18823531 \ 0.17254902]$
$[34 \ 51 \ 48]$	$[0.13333334 \ 0.20000002 \ 0.18823531]$
$\dots,$	$\dots,$
$[0 \ 28 \ 22]$	$[0.00000000 \ 0.10980393 \ 0.08627451]$
$[1 \ 29 \ 33]$	$[0.00392157 \ 0.11372550 \ 0.09019608]$
$[3 \ 31 \ 25]]$	$[0.01176471 \ 0.12156864 \ 0.09803922]]$
$[[12 \ 32 \ 27]$	$[[0.04705883 \ 0.12549020 \ 0.10588236]$
$[30 \ 47 \ 43]$	$[0.11764707 \ 0.18431373 \ 0.16862746]$
$[32 \ 49 \ 46]$	$[0.12549020 \ 0.19215688 \ 0.18039216]$
$\dots,$	$\dots,$

原图像素值输出

归一化后像素值输出

图 4.3 归一化前后像素值输出对比图

D. 插值采样

对 Mask 图像采用最近邻插值法进行插值采样。最近邻插值法在放大图像时补充的像素时最近邻像素的值。优点是处理速度快，但缺点是放大后的图像会被劣化，常常含有锯齿边缘。

设 $i + u, j + v$ (i, j 为正整数, u, v 为大于零小于 1 的小数, 下同) 为待求象素坐标, 则待求象素灰度的值 $f(i + u, j + v)$ 。如下图所示:

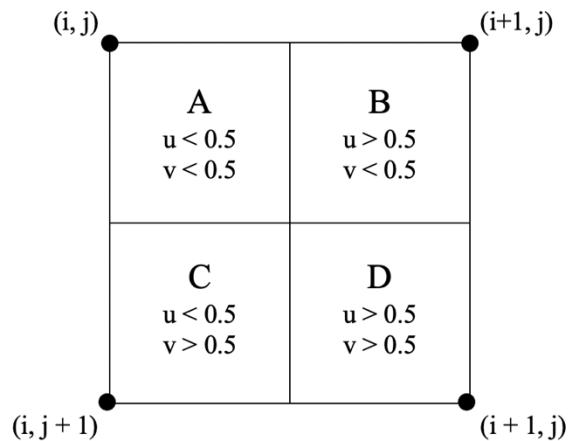


图 4.4 最近邻插值法说明图

装
订
线

如果 $(i + u, j + v)$ 落在 A 区, 即 $u < 0.5, v < 0.5$, 则将左上角象素的灰度值赋给待求象素, 同理, 落在 B 区则赋予右上角的象素灰度值, 落在 C 区则赋予左下角象素的灰度值, 落在 D 区则赋予右下角象素的灰度值。

最近邻插值算法原理:

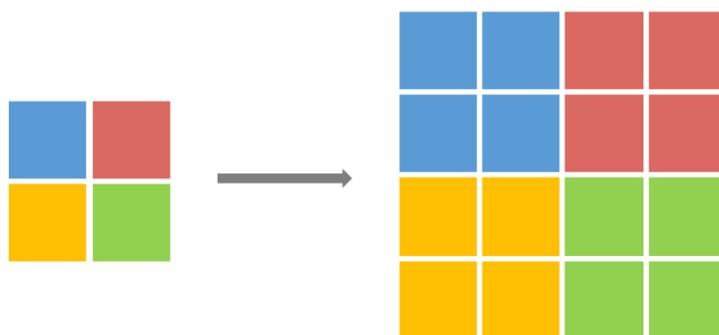


图 4.5 最近邻插值法原理图

4.2.2 肺结节图像数据增强

一般情况下，大量的参数往往能够训练出效果更好的网络模型，许多神经网络的参数都是数以百万计的。大量的数据可以保证这些参数可以正确地工作，然而实际情况中数据往往并没有想象中的那么多，为了提高模型的泛化能力，加入噪声提高模型的鲁棒性，一般地解决方法是对实验的数据进行扩充。扩充的办法主要是获取新数据或者进行数据增强。而获取新数据往往比较困难，需要消耗很多的资源和精力，因此数据增强是深度学习中对数据进行扩充最高效的方案。

在进行实验过程中，本文对肺结节图像数据进行了数据增强。文章采用的扩充方法有 2D 方向上的随机裁切（针对去除时间维度的 2D 图像进行裁切）、三种不同方向的翻转（针对去除时间维度的 2D 图像进行上下翻转、左右翻转以及对角线翻转）、旋转（在一定的角度约束下以图像中心为旋转中心进行随机旋转）和平移（在 x、y 两种方向上进行向左、向右、向上、向下的平移）。

为了解决正负样本相差悬殊的问题，对肺结节图像进行数据增强处理，通过旋转、平移、翻转等方式对肺结节图像进行扩充，对非肺结节图像进行随机采样。

数据增强也叫数据扩增，意思是在不实质性的增加数据的情况下，让有限的数据产生等价于更多数据的价值。

A. 随机裁切 (random crop)

随机裁切几乎是所有深度学习框架训练都具有的数据增强方法，在很多有名的深度学习网络（VGG, AlexNet, GoogleNet, ResNet...）的训练中，对输入 $256 * 256$ 的图像，通常会以 224 或 227 的窗口随机获得子图像作为训练，而在测试时则是以图像中心的子块（Patch）进行测试。最终使用则直接将图像 resize 到训练时用的 crop size 大小。实现上比较简单，在红色区域内随机取点作为滑窗左上角顶点。

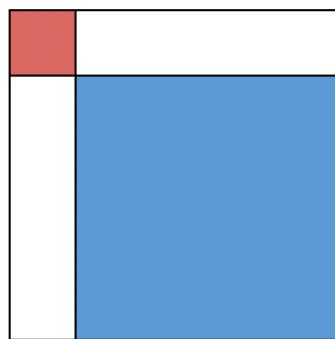


图 4.6 随机裁切原理示意图

B. 翻转（左右上下）

左右翻转也叫做水平翻转或镜像（mirror），将图像的左右部分以图像垂直中轴线为中心进行镜像对换。假设原图像高度为 h ，宽度为 w ，原图某一像素点 $P(x_0, y_0)$ ，经过水平镜像变换

后为 $P(w-1-x_0, y_0)$, 矩阵表示:

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 1 & w-1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

C. 旋转

图像旋转一般是以图像中心为旋转中心进行随机旋转（一般有一个正负角度约束）。

D. 平移

图像沿着 x 轴或 y 轴的平移有以下四种情况:

- (1) 向左平移
- (2) 向右平移
- (3) 向上平移
- (4) 向下平移

装
订
线

4.2.3 肺结节图像多任务分类

A. 肺结节图像的特征值描述

在 LIDC 数据集中，每一个肺结节 CT 图像都有许多特征值，每个特征值是由放射线医师注释的数值。这些属性中的每一个属性旁边都有一个与之相对应的计算属性，计算属性提供了给定特征数值的语义解释。

Subtlety 精细度

int, range = {1,2,3,4,5} - Difficulty of detection. Higher values indicate easier detection. 检测的困难度。越高的数值表示越容易检测。

1. 'Extremely Subtle'
2. 'Moderately Subtle'
3. 'Fairly Subtle'
4. 'Moderately Obvious'
5. 'Obvious'

InternalStructure 内部结构

int, range = {1,2,3,4} – Internal composition of the nodule. 结节的内部组成。

1. 'Soft Tissue'
2. 'Fluid'
3. 'Fat'
4. 'Air'

Calcification 钙化

int, range = {1,2,3,4,6} – Pattern of calcification, if present. 钙化模式（如果存在）

钙化对肺结节的诊断方面具有重要作用。钙化的分布、形态、以及含量都能够传递很多重要的信息。稠密、中心、层状、爆米花样及散在的钙化多为良性，而点状、网状、不定形的钙化多为恶性。

1. 'Popcorn'
2. 'Laminated'
3. 'Solid'
4. 'Non-central'
5. 'Central'
6. 'Absent'

Sphericity 球形度

int, range = {1,2,3,4,5} – The three-dimensional shape of the nodule in terms of its roundness. 结节的三维形状的球形度。

1. 'Linear'
2. 'Ovoid/Linear'
3. 'Ovoid'
4. 'Ovoid/Round'
5. 'Round'

Margin 边缘

int, range = {1,2,3,4,5} – Description of how well-defined the nodule margin is. 对结节边缘的清晰度的描述。

1. 'Poorly Defined'
2. 'Near Poorly Defined'
3. 'Medium Margin'
4. 'Near Sharp'
5. 'Sharp'

Lobulation 分叶征

int, range = {1,2,3,4,5} – The degree of lobulation ranging from none to marked. 分叶的程度（从无到标记）

分叶征 (lobulation)，是指肿块的轮廓并非纯粹的圆形或椭圆形，表面常呈现为凹凸不平的多个弧形，形似多个结节融合而成，通常可分为深分叶和浅分叶，以分叶部分的弧度为标准：弦距与距长之比 $> 2/5$ 为深分叶。病理基础一是与肿瘤边缘各部位肿瘤细胞分化程度不一，生长速度不同有关。二是肺的结缔组织间隔，进入肿瘤的血管、支气管分支、从肿瘤内向外生长的血管和结缔组织等可引起肿瘤生长受限，产生凹陷，从而形成分叶的形态。深分叶征在肺癌诊断中具有重要的意义。

1. 'No Lobulation'
2. 'Nearly No Lobulation'

- 3. 'Medium Lobulation'
- 4. 'Near Marked Lobulation'
- 5. 'Marked Lobulation'

Spiculation 毛刺征

int, range = {1,2,3,4,5} – The extent of spiculation present. 存在针刺的程度。

- 1. 'No Spiculation'
- 2. 'Nealy No Spiculation'
- 3. 'Medium Spiculation'
- 4. 'Near Marked Spiculation'
- 5. 'Marked Spiculation'

Texture 纹理

int, range = {1,2,3,4,5} – Radiographic solidity: internal texture (solid, ground glass, or mixed). 射线照相硬度：内部纹理（固体、磨砂玻璃或混合）

- 1. 'Non-Solid/GGO'
- 2. 'Non-Solid/Mixed'
- 3. 'Part-Solid/Mixed'
- 4. 'Solid/Mixed'
- 5. 'Solid'

Malignancy 恶性度

int, range = {1,2,3,4,5} – Subjective assessment of the likelihood of malignancy, assuming the scan originated from a 60-year-old male smoker. 假设扫描图像来自于一位 60 岁的男性吸烟者，主观评估恶性肿瘤的可能性。

- 1. 'Highly Unlikely'
- 2. 'Moderately Unlikely'
- 3. 'Indeterminate'
- 4. 'Moderately Suspicious'
- 5. 'Highly Suspicious'

B. 肺结节图像的多任务分类

结合上述对于肺结节图像特征值的具体描述，本文按照特征值的不同对肺结节图像的分类进行多任务的划分。例如对于 Malignancy、Lobulation、Calcification 三种特征，对其每一个的分类任务都可以被视为多任务中的一个任务，对这三个任务，分别将其记为 Task A, Task B, Task C。

由于放射性医师对数据集中特征值标注的分数数值都在一个范围内取值，为了简化分类工作，本文根据放射性医师对特征信息的分数将特征值分为两部分，较低分和较高分。例如，Margin 特征的打分范围在 1~5，本文通过将得分在 1, 2 的 'Poorly Defined' 和 'Near Poorly Defined' 数据合并，将得分在 4, 5 的 'Near Sharp' 和 'Sharp' 数据合并，前者为较低分的数据

组，在此表示图像的边界相对模糊，后者为较高分的数据组，在此表示图像的边界相对锐利。这样做一方面扩大了数据样本的量，另一方面简化了分类工作的复杂程度，使得实验能够更加高效的进行。同理，对于其他的特征值，也采取了同样的方式进行数据的合并与分类任务的简化工作。

4.3 多任务持续学习实验方案介绍

目前有相当一部分进行肺结节图像分类任务的研究都是基于 LIDC 数据的。原因主要是 LIDC 图像分辨率高，图像清晰，利于数据分析；另外它可以良好的反应患者肺部结构信息的变化，从多种不同的角度对多种特征值进行了评分标注，这些特征的变化往往是和肺癌的疾病进展直接相关的。因此，LIDC 数据集是研究肺结节图像分类任务的良好工具，本文所进行的多任务持续学习实验也是基于 LIDC 数据集开展的。

本文首先从 LIDC 数据集获取到了患者的肺结节 CT 图像，数据集为 DICOM 数据格式。通过上文介绍的处理方式，对数据进行了数据预处理和数据扩增。同时以不同的特征值为分类依据，设定了多个不同的肺结节图像分类任务，并将这些数据存储到磁盘中来进行后续的实验步骤。

在和深度学习相关的学术研究与教学领域中，很多算法都有数据分布均匀这样一个基本假设。但是当这些算法被直接应用于实际数据时，大多数情况下都无法取得理想的效果。解决类间数据不均衡问题的方法多种多样，但是解决问题的基本思路就是让实验数据当中的正负样本在训练过程中都可以对模型产生相同力度的影响，比如利用采样或加权等方法，就可以比较好地处理这一问题。本文使用了上采样的方案来解决此问题。

在患者数据当中，以 5:1 的比例区分训练集和验证集。在训练集中，取 20% 的数据作为验证集实施五折交叉验证。实验数据分配如图所示。

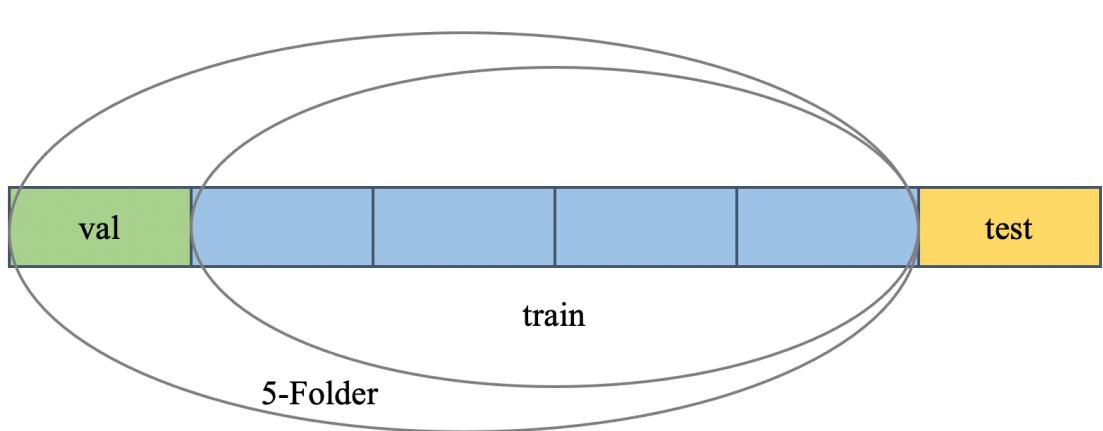


图 4.7 实验数据分配示意图

在区分好训练集、验证集和测试集后，文章搭建了 Resnet 网络，在网络结构搭建结束后，通过结合使用 EWC 算法，达到了对多任务分类问题的训练，最终实验可以实现多任务持续学习，同时完成在多种特征值（如 Malignancy、Lobulation、Calcification）上的分类问题。

此外，为了验证 EWC 算法在医学图像分类上进行多任务持续学习的有效性，本文还进行了使用 SGD 随机梯度下降方式和使用 L2 正则化方式的神经网络搭建，并同样的将多任务应用在这两种网络上，将其学习效果与使用 EWC 算法搭建的网络进行对比分析，进一步验证 EWC 算法在多任务持续学习方面的有效性。

4.4 多任务持续学习实验结果分析和讨论

4.4.1 分类模型评估参数

文章保存了模型中验证集损失函数最低的部分用以评估模型，以准确率来判别模型的好坏。首先选取总数据量的六分之一当作测试集留做测试来表示测试集准确率，剩余的数据采用五折交叉验证的方式，取验证集的五次平均准确率作为评估模型的一个标准（后文的其他标准也是通过取平均进行计算）。测试集和验证集都不会参与到训练当中以保证模型的泛化能力以及模型评估的有效性。

除了使用准确率（accuracy）作为模型的评估标准以外，在分类问题中，文章计算了模型的灵敏度（sensitivity，也称为真阳性率），特异度（specificity，也称为真阴性率）等一系列评估参数。通过这些参数，文章更全面的分析了试验结果。

首先介绍几个常见的模型评价术语，现在假设我们的分类目标只有两类，记为正例（positive）和负例（negative）分别是：

1. True Positive (TP): 被正确地划分为正例的个数，即实际为正例且被判定为正例的样本数
2. False Positive (FP): 被错误地划分为正例的个数，即实际为负例但被判定为正例的样本数
3. False Negative (FN): 被错误地划分为负例的个数，即实际为正例但被判定为负例的样本数
4. True Negative (TN): 被正确地划分为负例的个数，即实际为负例且被判定为负例的样本数

具体的评估标准介绍如下：

准确率和加权准确率

分类准确率指的是所有分类正确的样本占总样本的百分比。公式表示为：

$$acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.2)$$

由于分类时类间数据可能并不是完全均衡的，加权准确率就是将每个类别的数量所占百分比考虑而计算的准确率。

灵敏度和特异度

灵敏度（sensitivity，也称为真阳性率）是指在实际为阳性的样本中，判断为阳性的比例，计算方式是真阳性除以真阳性+假阴性（实际为阳性，但判断为阴性）的比值（能将实际患病的病例正确地判断为患病的能力）。公式表示为：

$$sensitivity = \frac{TP}{TP + FN} \quad (4.3)$$

特异度 (specificity, 也称为真阴性率) 是指在实际为阴性的样本中, 判断为阴性的比例, 计算方式是真阴性除以真阴性+假阳性 (实际为阴性, 但判断为阳性) 的比值 (能正确判断实际未患病的病例的能力)。公式表示为:

$$specificity = \frac{TN}{TN + FP} \quad (4.4)$$

4.4.2 分类模型实验结果

本文按照上述的一系列解决方案, 实验了多种 Resnet 结构, 最终选择了 Resnet101 作为本文的实验模型。在分别结合使用 SGD、L2、EWC 方法对多任务持续学习分类问题进行了探究。在顺序学习三个任务 Task A、B、C 的过程中, 对三种方法网络的表现进行对比分析。三种方法对于多任务持续学习的各项评估指标的具体结果如下。

Task A 分类结果

在学习 Task A 时, Task A 的分类结果:

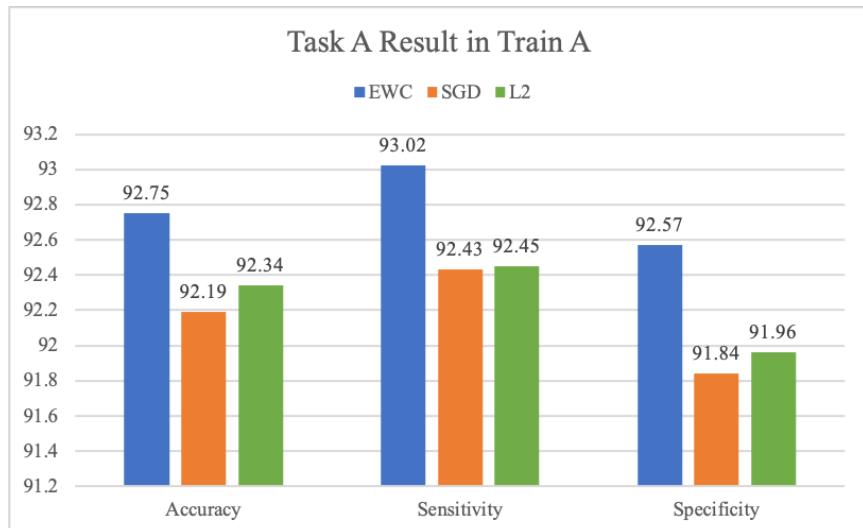


图 4.8 学习完成 Task A 后 Task A 测试集的各项指标柱状图

在学习 Task A 时, Task A 的准确率:

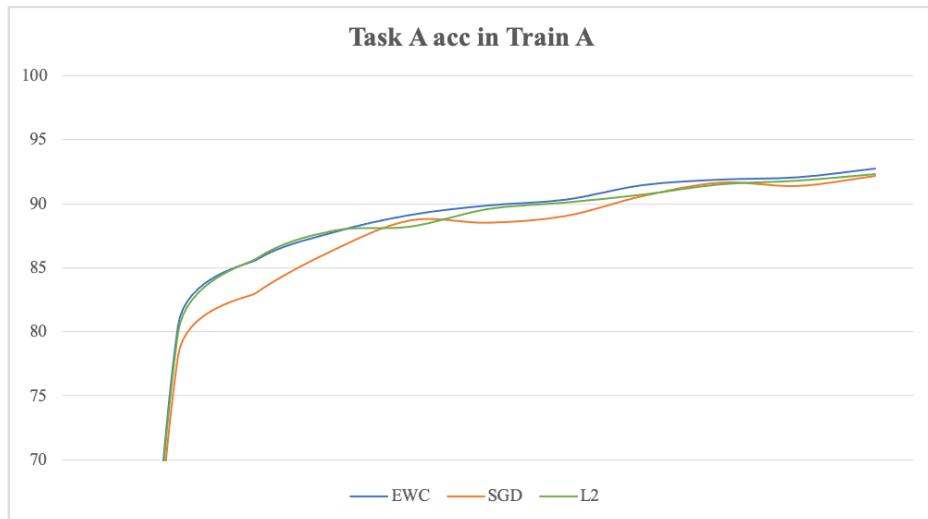


图 4.9 学习 Task A 过程中 Task A 的准确率折线图

装
订
线

在 Task A 的学习完成后，对 Task A 测试集进行测试的各项指标如下：

测试集总准确率：92.75% (EWC)、92.19% (SGD)、92.34% (L2)

测试集加权总准确率：92.75% (EWC)、92.19% (SGD)、92.34% (L2)

测试集各项指标整合表：

表 4.1 学习完成 Task A 后 Task A 测试集各项指标整合表

类别	Accuracy	Sensitivity	Specificity
EWC	92.75%	93.02%	92.57%
SGD	92.19%	92.43%	91.84%
L2	92.34%	92.45%	91.96%

在学习完 Task A，顺序学习 Task B 时，Task A 的分类结果：

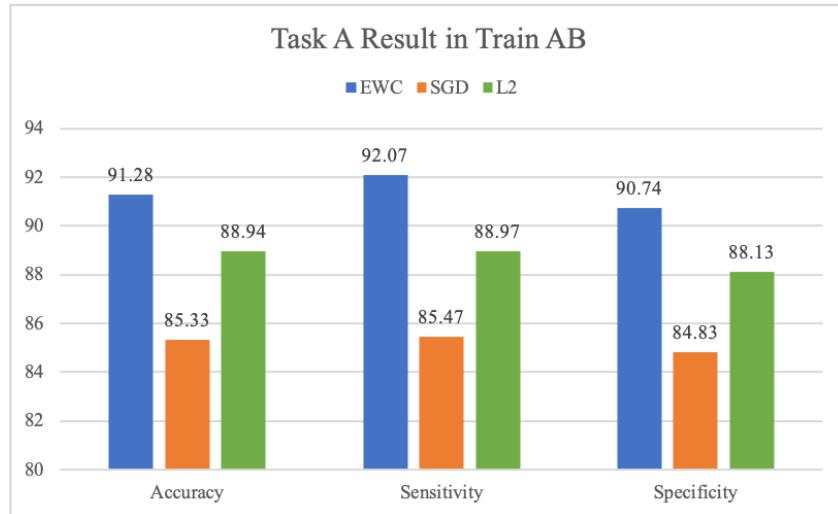


图 4.10 顺序学习完成 Task AB 后 Task A 测试集的各项指标柱状图

装

订

线



图 4.11 学习 Task B 过程中 Task A 的准确率折线图

在 Task A、B 的学习顺序完成后，对 Task A 测试集进行测试的各项指标如下：

测试集总准确率：91.28% (EWC)、85.33% (SGD)、88.94% (L2)

测试集加权总准确率：91.28% (EWC)、85.33% (SGD)、88.94% (L2)

测试集各项指标整合表：

表 4.2 顺序学习完成 Task AB 后 Task A 测试集各项指标整合表

类别	Accuracy	Sensitivity	Specificity
EWC	91.28%	92.07%	90.04%
SGD	85.33%	85.47%	84.83%
L2	88.94%	88.97%	88.13%

在顺序学习完 Task A、B，继续学习 Task C 时，Task A 的分类结果：

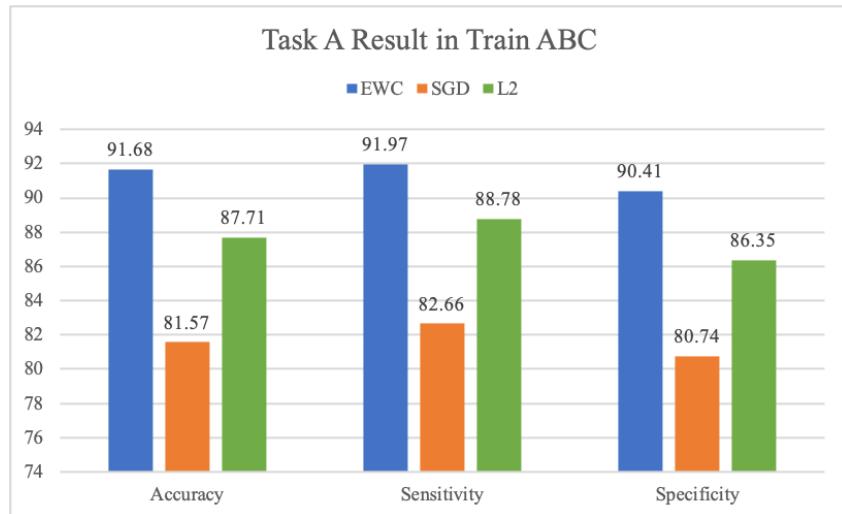


图 4.12 顺序学习完成 Task ABC 后 Task A 测试集的各项指标柱状图

装
订
线

在顺序学习完 Task A、B，继续学习 Task C 时，Task A 的准确率：



图 4.13 学习 Task C 过程中 Task A 的准确率折线图

在 Task A、B、C 的学习顺序完成后，对 Task A 测试集进行测试的各项指标如下：

测试集总准确率：91.68% (EWC)、81.57% (SGD)、87.71% (L2)

测试集加权总准确率：91.68% (EWC)、81.57% (SGD)、87.71% (L2)

测试集各项指标整合表：

表 4.3 顺序学习完成 Task A、B、C 后 Task A 测试集各项指标整合表

类别	Accuracy	Sensitivity	Specificity
EWC	91.68%	91.97%	90.41%
SGD	81.57%	82.66%	80.74%
L2	87.71%	88.78%	86.35%

在学习完 Task A，顺序学习 Task B 时，Task B 的分类结果：

装
订
线

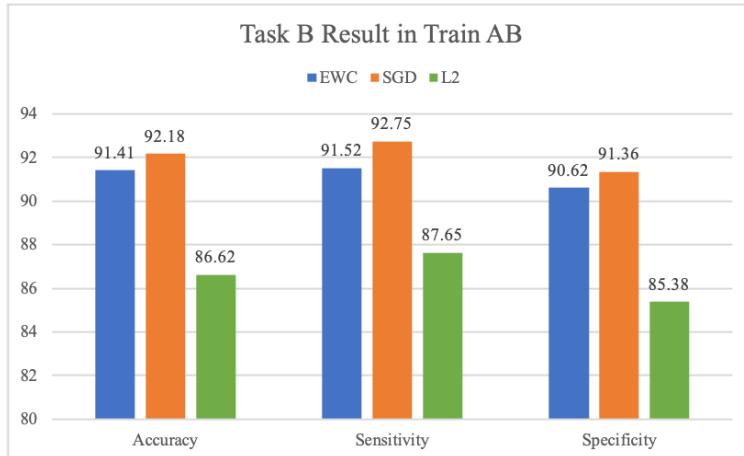


图 4.14 顺序学习完成 Task AB 后 Task B 测试集的各项指标柱状图

在学习完 Task A，顺序学习 Task B 时，Task B 的准确率：



图 4.15 学习 Task B 过程中 Task B 的准确率折线图

在 Task A、B 的学习顺序完成后，对 Task B 测试集进行测试的各项指标如下：

测试集总准确率：91.41% (EWC)、92.18% (SGD)、86.62% (L2)

测试集加权总准确率：91.41% (EWC)、92.18% (SGD)、86.62% (L2)

测试集各项指标整合表：

表 4.4 顺序学习完成 Task A、B 后 Task B 测试集各项指标整合表

类别	Accuracy	Sensitivity	Specificity
EWC	91.41%	91.52%	90.62%
SGD	92.18%	92.75%	91.36%
L2	86.62%	87.65%	85.38%

在顺序学习完 Task A、B，继续学习 Task C 时，Task B 的分类结果：

装
订
线

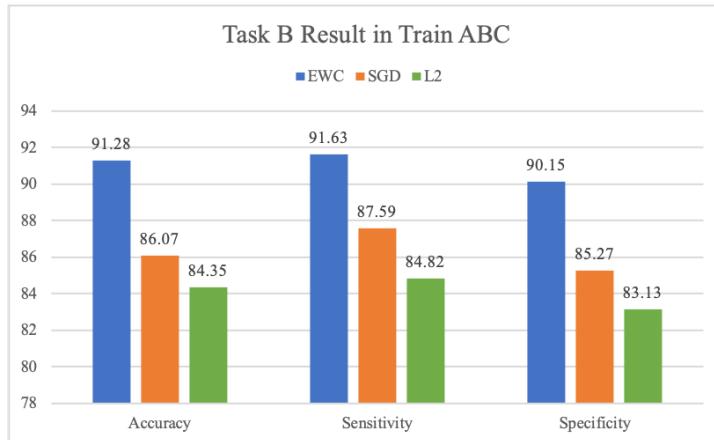


图 4.16 顺序学习完成 Task ABC 后 Task B 测试集的各项指标柱状图

在顺序学习完 Task A、B，继续学习 Task C 时，Task B 的准确率：

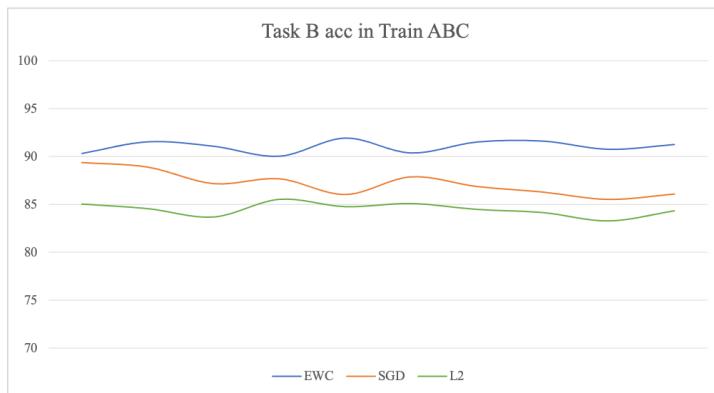


图 4.17 学习 Task C 过程中 Task B 的准确率折线图

在 Task A、B、C 的学习顺序完成后，对 Task B 测试集进行测试的各项指标如下：

测试集总准确率：91.28% (EWC)、86.07% (SGD)、84.35% (L2)

测试集加权总准确率：91.28% (EWC)、86.07% (SGD)、84.35% (L2)

测试集各项指标整合表：

表 4.5 顺序学习完成 Task A、B、C 后 Task B 测试集各项指标整合表

类别	Accuracy	Sensitivity	Specificity
EWC	91.28%	91.63%	90.15%
SGD	86.07%	87.59%	85.27%
L2	84.35%	84.82%	83.13%

在顺序学习完 Task A、B，继续学习 Task C 时，Task C 的分类结果：

装
订
线

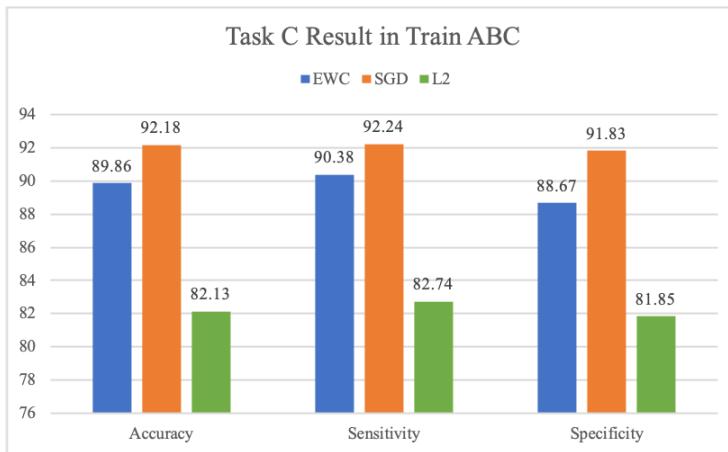


图 4.18 顺序学习完成 Task ABC 后 Task C 测试集的各项指标柱状图

在顺序学习完 Task A、B，继续学习 Task C 时，Task C 的准确率：

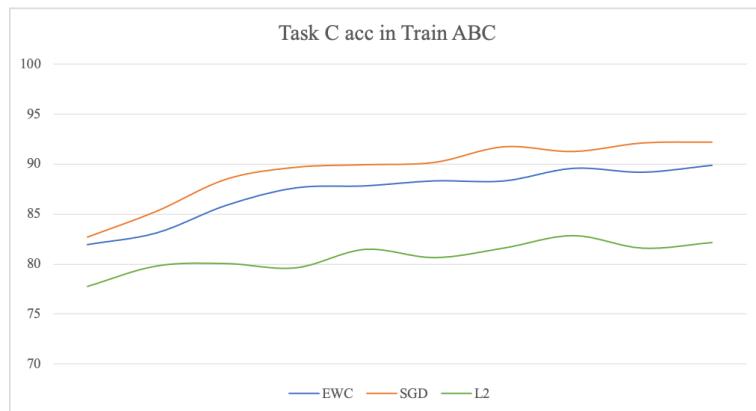


图 4.19 学习 Task C 过程中 Task C 的准确率折线图

在 Task A、B、C 的学习顺序完成后，对 Task B 测试集进行测试的各项指标如下：

测试集总准确率：89.86% (EWC)、92.18% (SGD)、82.13% (L2)

测试集加权总准确率：89.86% (EWC)、92.18% (SGD)、82.13% (L2)

测试集各项指标整合表：

表 4.6 顺序学习完成 Task A、B、C 后 Task C 测试集各项指标整合表

类别	Accuracy	Sensitivity	Specificity
EWC	89.86%	90.38%	88.67%
SGD	92.18%	92.24%	91.83%
L2	82.13%	82.74%	81.85%

4.4.3 分类模型实验分析

依据本文 5.1.2 节各个分类问题的表现结果，汇总图如图所示：

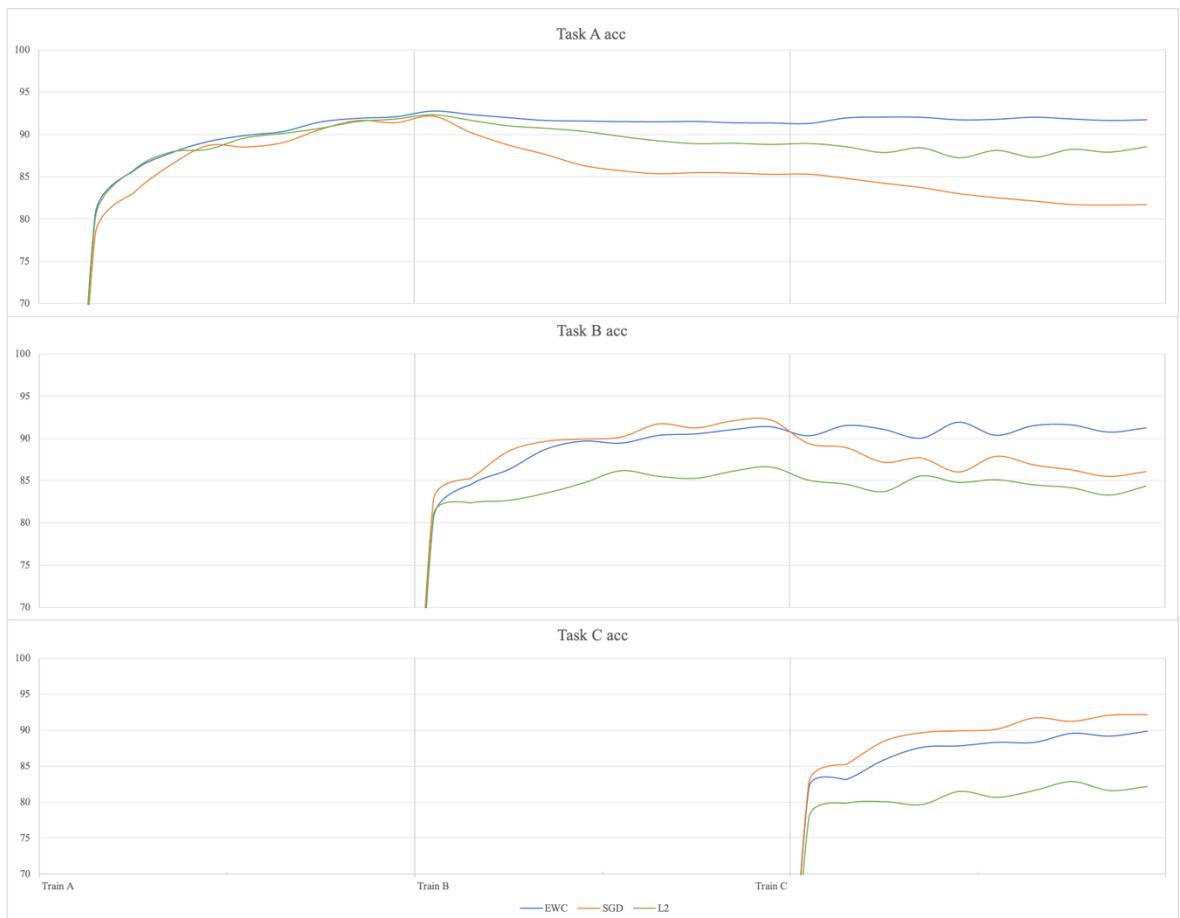


图 4.20 顺序学习 Task ABC 过程中各任务的准确率折线图

表 4.7 顺序学习 Task ABC 过程中各任务测试集准确率整合表

类别	Train A	Train B	Train C
EWC (Task A)	92.75%	91.28%	91.68%
SGD (Task A)	92.19%	85.33%	81.57%
L2 (Task A)	92.34%	88.94%	87.71%
EWC (Task B)	-	91.41%	91.28%
SGD (Task B)	-	92.18%	86.07%
L2 (Task B)	-	86.62%	84.35%
EWC (Task C)	-	-	89.86%
SGD (Task C)	-	-	92.18%
L2 (Task C)	-	-	82.13%

A. 任务 A 在顺序学习任务 ABC 过程中的表现

汇总图最上面部分的折线图代表在整个顺序学习 Task A、B、C 的过程中，网络模型对于 Task A 的准确率表现。其中纵轴代表 Task A 的准确率，横轴代表训练进度，依据顺序学习的任务顺序分为 Train A、Train B、Train C 三个时间跨度。具体如图所示。

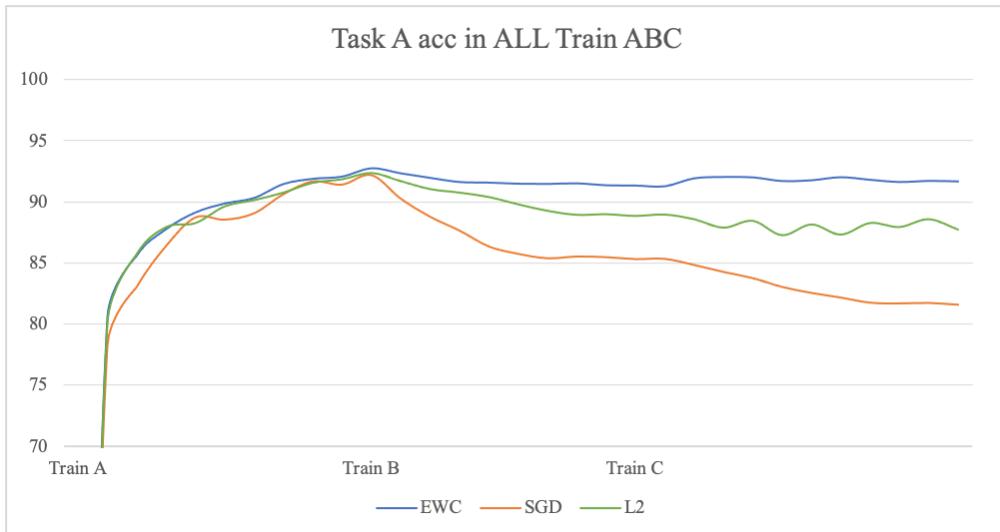


图 4.21 顺序学习 Task ABC 过程中 Task A 的准确率折线图

从图中我们可以看到，在折线图最左边的部分，代表多任务持续学习在训练第一个任务，即 Train A 的训练过程中，SGD、L2 和 EWC 三种方法都得到了较好的表现。这是因为此时只训练了 Task A，这部分在概念上其实相当于一个单任务，不会出现灾难性遗忘的问题。所以，三种方法在此时 Task A 的准确率上都得到了比较好的表现。

接下来，在进行完 Task A 的学习之后，进入到 Task B 的学习过程，即折线图的中间部分。这部分代表在学习 Task B 的过程当中，网络对于 Task A 准确率的表现变化。由图中信息我们可

以看出，在学习 Task B 的过程中，使用 EWC 方式的网络结构在学习 Task B 的同时保持了 Task A 的准确率，即实现了多任务持续学习。而使用 L2 正则化和 SGD 随机梯度下降方式的网络结构则在 Task A 的表现上有所下降，且使用 SGD 方式的网络在 Task A 表现上的下降比使用 L2 正则化方式的网络更为明显。通过本文之前提及的原理可以推测，这是因为结合 SGD 方式和 L2 正则化方式的网络结构都具有灾难性遗忘的问题，但是由于 L2 正则化对于网络的参数有一种约束效果，所以灾难性遗忘的程度比 SGD 要小，这也就解释了为什么结合 SGD 方式的网络在 Task A 表现上下降的更快。

最后，在顺序完成了 Task A、B 的学习之后，进入到 Task C 的学习过程，即折线图的最右侧部分。这部分代表在学习 Task C 的过程当中，网络对于 Task A 准确率的表现变化。从图中信息我们可以看出，在学习 Task C 的过程中，使用 EWC 方式的网络结构依旧在学习 Task C 的同时保持了 Task A 的准确率，即实现了多任务持续学习。而使用 L2 正则化和 SGD 随机梯度下降方式的网络结构则在 Task A 的表现上进一步的下降，且 SGD 的下降程度更为明显。整体上来看，在从两个任务 Task A、B 到三个任务 Task A、B、C 的过程中，三种网络结构的表现仍然类似，其中 EWC 保持高准确率且稳定，SGD 和 L2 的表现下滑且 SGD 下滑更加严重。其中原理同上文所述，在此不再赘述。

B. 任务 B 在顺序学习任务 ABC 过程中的表现

汇总图中间部分的折线图代表在整个顺序学习 Task A、B、C 的过程中，网络模型对于 Task B 的准确率表现。其中纵轴代表 Task B 的准确率，横轴代表训练进度，依据顺序学习的任务顺序分为 Train A、Train B、Train C 三个时间跨度。具体如图所示。

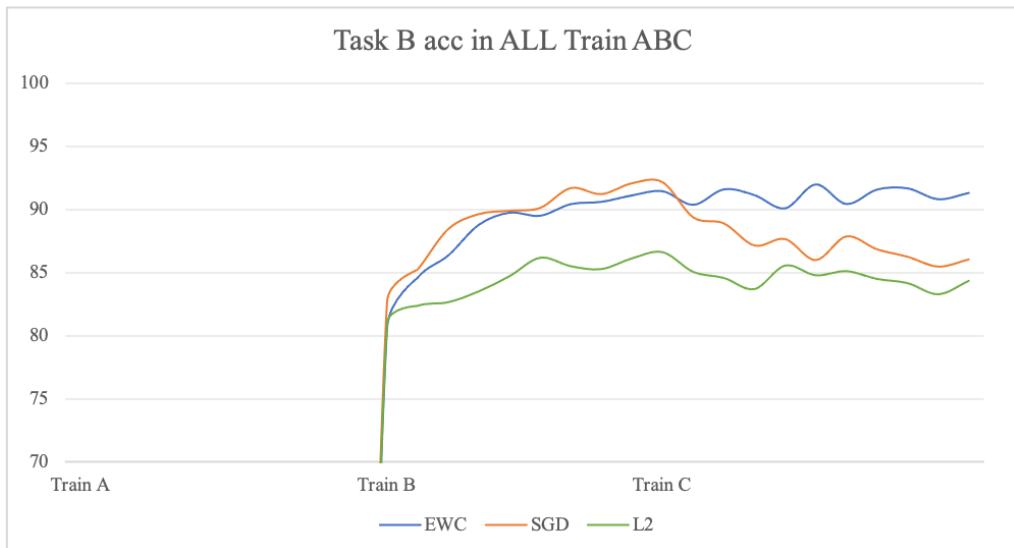


图 4.22 顺序学习 Task ABC 过程中 Task B 的准确率折线图

由于多任务 Task A、B、C 是顺序学习的，所以在折线图的左边部分，即 Train A 部分，是不涉及 Task B 的表现的，故此部分图为空，在这部分时间段中，网络正在进行对于 Task A 的训练。

接下来，在进行完 Task A 的学习之后，进入到 Task B 的学习过程，即折线图的中间部分。这部分代表在学习 Task B 的过程当中，网络对于 Task B 准确率的表现变化。结合上文，由图中信息我们可以看出，在学习 Task B 的过程中，使用 EWC 的网络结构在学习完 Task A 后仍旧能够对 Task B 进行有效的学习，同时保持了网络在 Task A、B 两种任务的优秀表现，即实现了多任务持续学习。而虽然结合 SGD 方式的网络在 Task B 上有较好的表现，但考虑到它存在灾难性遗忘的问题，即在 Task B 的学习过程中其在 Task A 上的准确率迅速下降，所以其在 Task B 上的优秀表现是以遗忘 Task A 为代价的。最后，结合 L2 正则化方式的网络在 Task B 上的学习效果并不是很好，这是因为在学习了 Task A 之后，由于 L2 正则化对于网络中的参数有一种约束限制，导致了网络在 Task B 中的学习空间不足，在 Task A 的参数约束限制下，无法对 Task B 进行有效的学习。换言之，L2 正则化方法在保证对 Task A 任务记忆能力的同时，其约束付出的代价是限制了后续学习的学习力。

最后，在顺序完成了 Task A、B 的学习之后，进入到了 Task C 的学习过程，即折线图的最右侧部分。这部分代表在学习 Task C 的过程当中，网络对于 Task B 准确率的表现变化。从图中信息我们可以看出，在学习 Task C 的过程当中，使用 EWC 方式的网络结构依旧在学习 Task C 的同时保持了 Task B 的准确率，即实现了多任务持续学习。而使用 L2 正则化和 SGD 随机梯度下降方式的网络结构则在 Task A 的表现上有所下降，且 SGD 的下降程度更加明显。其中原理同上文所述，在此不再赘述。

C. 任务 C 在顺序学习任务 ABC 过程中的表现

汇总图最下面部分的折线图代表在整个顺序学习 Task A、B、C 的过程中，网络模型对于 Task C 的准确率表现。其中纵轴代表 Task C 的准确率，横轴代表训练进度，依据顺序学习的任务顺序分为 Train A、Train B、Train C 三个时间跨度。具体如图所示。

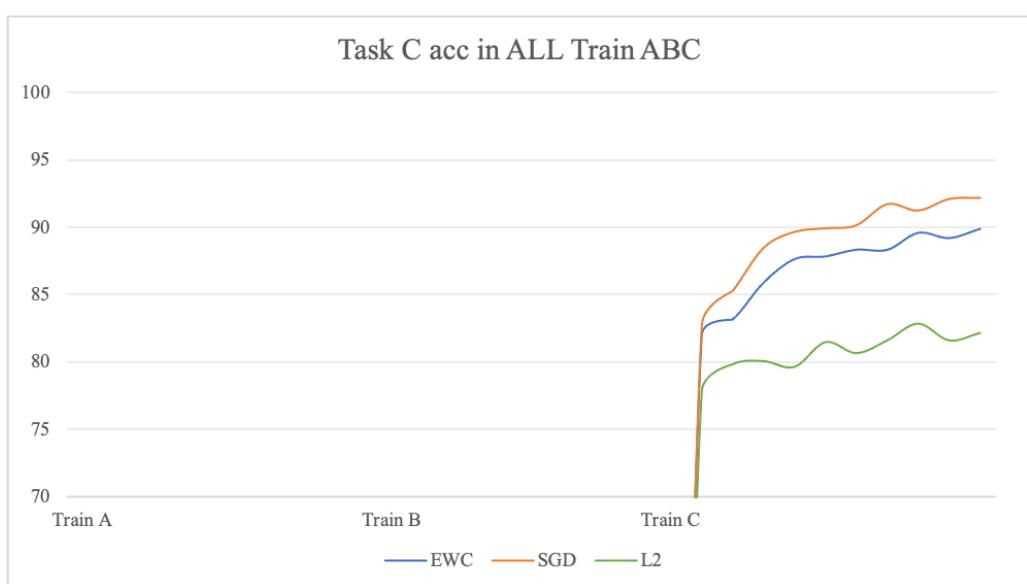


图 4.23 顺序学习 Task ABC 过程中 Task C 的准确率折线图

由于多任务 Task A、B、C 是顺序学习的，所以在折线图的左边部分和中间部分，即 Train A 部分和 Train B 部分，是不涉及 Task C 的表现的，故此部分图为空，在这部分时间段中，网络正在顺序进行对于 Task A 和 Task B 的训练。

最后，在顺序完成了 Task A、B 的学习之后，进入到了 Task C 的学习过程，即折线图的最右侧部分。这部分代表在学习 Task C 的过程当中，网络对于 Task B 准确率的表现变化。从图中信息我们可以看出，在学习 Task C 的过程当中，使用 EWC 方式的网络结构在学习完 Task A、B 后仍旧能够对 Task C 进行有效的学习，同时保持了网络在 Task A、B、C 三种任务的优秀表现，即实现了多任务持续学习。而虽然结合 SGD 方式的网络在 Task C 上有较好的表现，但它存在灾难性遗忘问题，即在 Task C 的学习过程中其在 Task A、B 上的准确率迅速下降，所以其在 Task C 上的优秀表现是以遗忘 Task A、B 为代价的。最后，结合 L2 正则化方式的网络在 Task C 上的学习效果并不是很好，这是因为在学习了 Task A、B 之后，由于 L2 正则化对于网络中的参数有一种约束限制，这导致了网络在后续任务 Task C 上的学习空间不足，在 Task A、B 的参数约束限制下，无法对 Task C 进行有效的学习。其中原理同上文所述，在此不再赘述。

装
订
线

4.5 不同任务顺序对实验结果影响的分析和讨论

4.5.1 实验设计

在上述实验过程中，我们所设定的多任务持续学习实验任务顺序为 Task A、Task B、Task C，为了进一步验证实验结果的普适性，避免由于不同任务学习顺序的变化对实验准确性所造成的影响，本文在此部分对其他五种不同任务顺序下的学习情况进行实验分析，任务的学习顺序分别为 Task ACB、Task BAC、Task BCA、Task CAB、Task CBA，所得到的实验结果如下。

4.5.2 实验结果

A. 顺序学习任务 A、C、B 过程中的网络表现

装订线

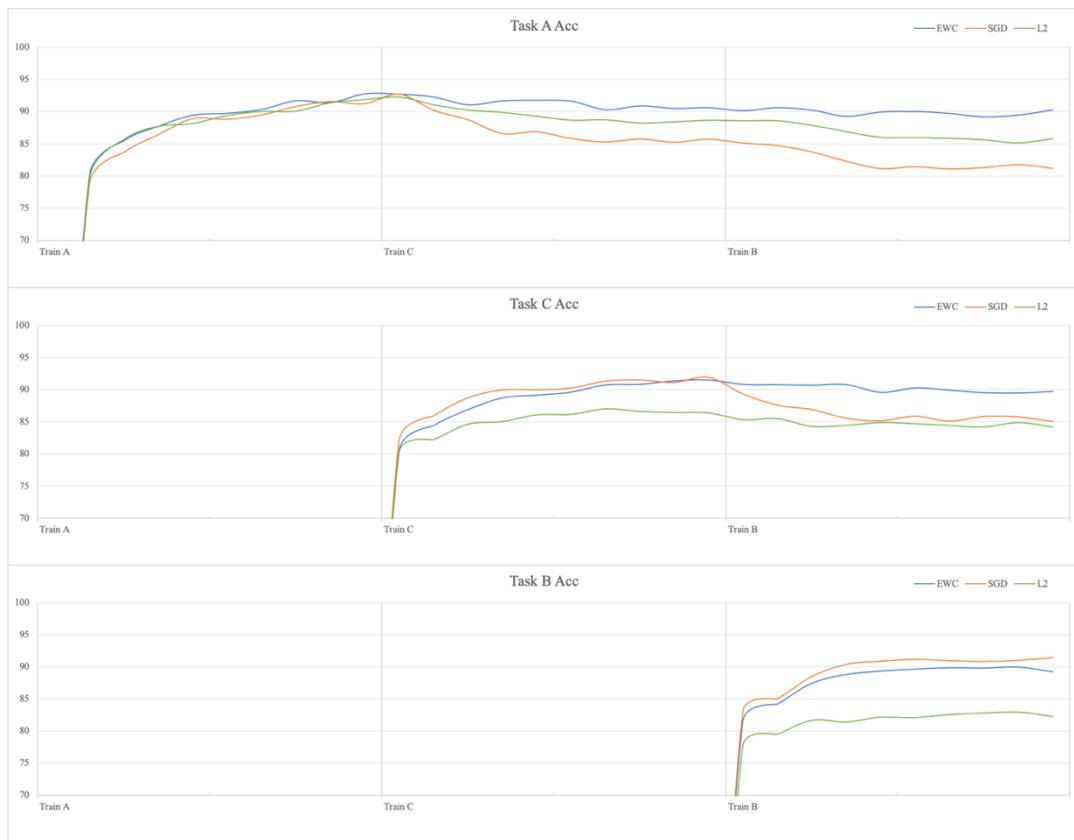


图 4.24 顺序学习 Task ACB 过程中各任务的准确率折线图

表 5.7 顺序学习 Task ACB 过程中各任务测试集准确率整合表

类别	Train A	Train C	Train B
EWC (Task A)	92.77%	90.63%	90.31%
SGD (Task A)	91.18%	85.66%	81.76%
L2 (Task A)	91.87%	88.65%	85.53%
EWC (Task C)	-	91.48%	89.71%
SGD (Task C)	-	91.91%	85.03%
L2 (Task C)	-	86.37%	84.18%
EWC (Task B)	-	-	89.21%
SGD (Task B)	-	-	91.42%
L2 (Task B)	-	-	82.23%

B. 顺序学习任务 B、A、C 过程中的网络表现



图 4.25 顺序学习 Task BAC 过程中各任务的准确率折线图

表 5.8 顺序学习 Task BAC 过程中各任务测试集准确率整合表

类别	Train B	Train A	Train C
EWC (Task B)	92.19%	91.29%	90.73%
SGD (Task B)	91.74%	85.67%	81.90%
L2 (Task B)	91.53%	87.02%	85.06%
EWC (Task A)	-	91.25%	90.61%
SGD (Task A)	-	92.15%	86.40%
L2 (Task A)	-	85.06%	84.73%
EWC (Task C)	-	-	90.84%
SGD (Task C)	-	-	92.54%
L2 (Task C)	-	-	82.73%

C. 顺序学习任务 B、C、A 过程中的网络表现

装
订
线

图 4.26 顺序学习 Task BCA 过程中各任务的准确率折线图

表 5.9 顺序学习 Task BCA 过程中各任务测试集准确率整合表

类别	Train B	Train C	Train A
EWC (Task B)	91.97%	91.76%	90.67%
SGD (Task B)	92.47%	85.89%	81.09%
L2 (Task B)	91.88%	88.89%	86.33%
EWC (Task C)	-	90.72%	90.15%
SGD (Task C)	-	91.82%	86.42%
L2 (Task C)	-	85.64%	83.37%
EWC (Task A)	-	-	89.72%
SGD (Task A)	-	-	91.14%
L2 (Task A)	-	-	80.75%

D. 顺序学习任务 C、A、B 过程中的网络表现

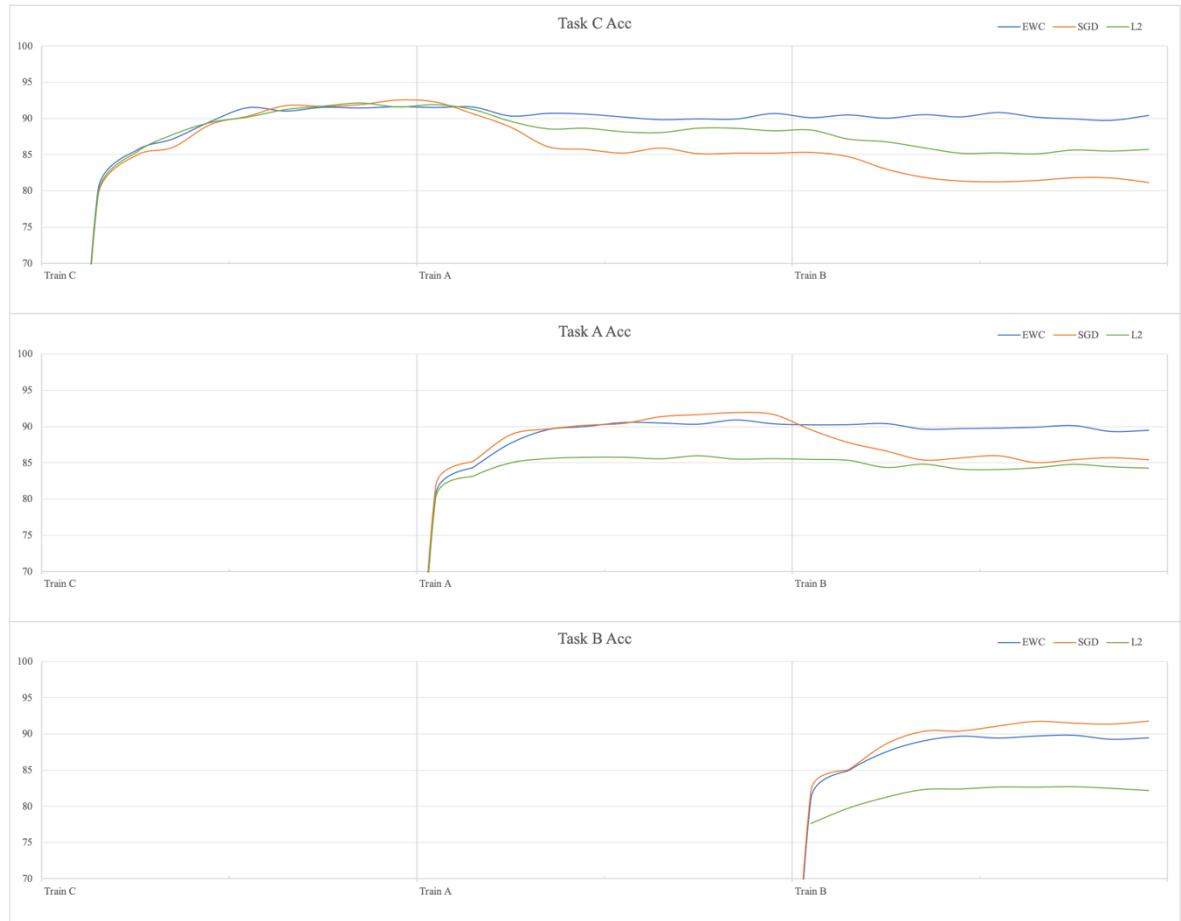


图 4.27 顺序学习 Task CAB 过程中各任务的准确率折线图

表 5.10 顺序学习 Task CAB 过程中各任务测试集准确率整合表

类别	Train C	Train A	Train B
EWC (Task C)	91.62%	90.68%	89.87%
SGD (Task C)	92.57%	85.22%	81.60%
L2 (Task C)	91.57%	88.27%	85.50%
EWC (Task A)	-	90.37%	89.47%
SGD (Task A)	-	91.56%	85.40%
L2 (Task A)	-	85.55%	84.23%
EWC (Task B)	-	-	89.45%
SGD (Task B)	-	-	91.80%
L2 (Task B)	-	-	82.18%

E. 顺序学习任务 C、B、A 过程中的网络表现



图 4.28 顺序学习 Task CBA 过程中各任务的准确率折线图

表 5.11 顺序学习 Task CBA 过程中各任务测试集准确率整合表

类别	Train C	Train B	Train A
EWC (Task C)	91.87%	90.57%	90.33%
SGD (Task C)	92.90%	85.65%	81.40%
L2 (Task C)	91.13%	88.40%	85.90%
EWC (Task B)	-	90.17%	89.34%
SGD (Task B)	-	91.65%	85.46%
L2 (Task B)	-	85.87%	84.72%
EWC (Task A)	-	-	89.37%
SGD (Task A)	-	-	90.11%
L2 (Task A)	-	-	81.33%

4.5.3 实验分析

通过对其他五种不同的任务顺序下的学习情况进行实验对比分析，我们可以发现即便网络在不同任务顺序下对每个任务学习情况的数据表现稍有波动，但是从整体上来看，不同的任务顺序对于网络在学习情况上的变化影响不大，即我们可以认为在多任务持续学习的过程中，不同的任务学习顺序对网络的最终表现几乎没有影响，从而验证了网络在多任务持续学习上的有效性。

不管 Task A、Task B、Task C 学习的先后顺序如何。总结实验结果中的网络表现情况可知，EWC 方法在任何任务顺序下都能完成良好的多任务持续学习效果，即在对后继任务进行有效的学习时同时保持对先前任务的有效记忆，能够同时在多种任务上有优秀的表现。而 SGD 方法由于存在灾难性遗忘的问题，其对于每个人物的学习空间都较大，从而在学习当前任务时能在当前任务上有较好的表现，但这种优秀的表现会以忘掉先前学习的任务为代价，即其在进行新的任务学习时对于先前任务的表现会大幅下降。最后，L2 正则化方法在学习后续任务时，虽然由于其对参数的约束限制能够在一定程度上保持对先前任务的记忆，但这种参数上的约束限制同时限制了对于后续任务的学习空间，以至于在多任务学习情况下，L2 正则化方法无法对后续任务的学习有较好的表现。

装
订
线

5 结论和展望

5.1 结论

本文基于 CNN 网络结构，首先针对肺结节医学图像的分类任务搭建了 Resnet 网络结构，通过预处理和调整网络参数，在三种医学图像的分类问题中都取得了良好的效果。在此基础上，本文提出了基于 EWC 弹性权重合并算法的 Resnet 网络模型，针对多任务持续学习的情况，同时学习三种肺结节医学图像的分类问题，并辅以经典的 SGD 随机梯度下降方法和 L2 正则化方法进行实验对比分析。通过实验，本文基于 EWC 方式的 Resnet 网络在三种分类问题任务的多任务持续学习上达到了比较好的效果。在未来数据量增大的情况下，本文也相信这种基于 EWC 的 Resnet 网络能够在多任务持续学习方面作出更好的表现。最终本文通过一系列的实验说明了几个主要事实：

- (1) 目前针对肺结节医学图像的多任务持续学习研究正在逐步完善，在辅助医生诊断肺结节医学图像方面已经可以给出一些较为有价值的信息
- (2) 在合理并有效的利用以往的网络参数进行优化的情况下，在多任务持续学习中得到的分类结果往往要好于仅仅基于当前任务进行参数优化的分类结果。
- (3) 本文的实验结果证明了卷积神经网络结构可以充分的挖掘空间信息，有利于解决肺结节医学图像的分类问题。另外本文也表明 EWC 弹性权重算法可以充分的利用以往任务的学习信息，有利于解决多任务持续学习中灾难性遗忘的问题。

5.2 展望

随着计算机科学技术的发展，多任务持续学习学习在医学领域的应用也在逐步增加，本文结合多任务持续学习，通过 CNN 卷积神经网络，针对医学图像数据挖掘特征，进行多种分类问题的持续学习，展示了多任务持续学习在医学方面的巨大潜力。

针对肺癌的诊断问题来说，本文仅仅解决了肺结节医学图像的部分分类问题，更多的问题如肺癌的发病进程预测，肺癌发病位置的定位，肺癌与医学量表（如 MMSE）结合的回归还有待探究。在未来，本文将继续深入肺结节医学图像在多任务持续学习上的分类研究，并将会结合肺癌的分类与疾病定位。同时，本文会更加关注一些临床需求，将计算机领域的多任务持续学习技术真正作用于实际的医学问题，达到理论用于实践，真正解决病人问题的目的。

参考文献

- [1] Sebastian Ruder. An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098, 2017.
- [2] A. Rios and L. Itti. Closed-loop GAN for continual learning. arXiv preprint arXiv:1811.01146, 2018.
- [3] Sébastien Jean, Orhan Firat, and Melvin Johnson. Adaptive scheduling for multi-task learning. arXiv preprint arXiv:1909.06434, 2019.
- [4] S. Liu, E. Johns, and A. J. Davison. End-to-end multi-task learning with attention[C]. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 1871–1880, 2019.
- [5] Suteu M, Guo Y. Regularizing deep multi-task net- works using orthogonal gradients. arXiv preprint arXiv:1912.06844, 2019.
- [6] Jonathan S, Wojciech C, Jelena L, et al. Progress & compress: A scalable framework for continual learning[C]. Proceedings of ICML, 2018.
- [7] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics[C]. Conference on Computer Vision and Pattern Recognition, 2018.
- [8] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Recurrent neural network for text classification with multi-task learning[C]. Proceedings of International Joint Conference on Artificial Intelligence.2016.
- [9] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning[C]. CVPR, 2016.
- [10] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization[C]. Advances in Neural Information Processing Systems, pages 525–536, 2018.
- [11] David Lopez-Paz and Marc' Aurelio Ranzato. Gradient episodic memory for continual learning[C]. Conference and Workshop on Neural Information Processing Systems(NIPS), 2017.
- [12] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence[C]. International Conference on Machine Learning (ICML), 2017.
- [13] Adel, T., Zhao, H., and Turner, R. E. Continual learning with adaptive weights (CLAW)[C]. 8th International Conference on Learning Representations, ICLR 2020.
- [14] Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. Variational continual learning[C]. International Conference on Learning Representations, ICLR, 2018.
- [15] Shin, Hanul, Lee, Jung Kwon, Kim, Jaehong, and Kim, Jiwon. Continual learning with deep generative replay[C]. Advances in Neural Information Processing Systems, pp. 2994–3003, 2017.
- [16] Rolnick, D. Ahuja, A. Schwarz, J. Lillicrap, T. P. and Wayne, G. Experience Replay for Continual Learning[J]. CoRR abs/1811.11682, 2018.

装订线

-
- [17] Zhai M, Chen L, Tung F, He J, Nawhal M, Mori G. Lifelong gan: Continual learning for conditional image generation[C]. IEEE International Conference on Computer Vision (ICCV), 2019.
 - [18] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. How to escape saddle points efficiently[C]. Proceedings of the 34th International Conference on Machine Learning, pages 1724–1732, 2017.
 - [19] Andrew Y.Ng. Feature selection, l1 vs l2 regularization, and rotational invariance[C]. Proceedings of the 21st Internal Conference on Machine Learning, 2004.
 - [20] He Kaiming, et al. Deep Residual Learning for Image Recognition[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR): 770-778, 2016.
 - [21] He Kaiming, Zhang Xiangyu, Ren Shaoqing, Sun Jian. (2016). Identity Mappings in Deep Residual Networks. 9908. 630-645. 10.1007/978-3-319-46493-0_38.
 - [22] James Kirkpatrick et al. Overcoming catastrophic forgetting in neural networks[J]. Proceedings of the National Academy of Sciences(PNAS). 114(13):3521–3526, 2017.

装
订
线

謝 辭

衷心地感谢罗烨老师和张国凯学长对我在肺结节医学图像结合多任务持续学习研究项目上的指导和支持。罗烨老师和张国凯学长为我指明了我的研究方向，在研究的过程中，老师提出了许许多多的宝贵意见。在这段时间的项目过程中，每周实验室组会我们都会定期汇报项目进展，罗烨老师也会时时刻刻的关注和指导我们的研究。正是有了罗烨老师的帮助，才使得我的毕业设计可以顺利进行，因此在这里首先要衷心感谢罗烨老师的帮助和指导！

在论文的实验过程中，我遇到了许多问题，在 iLab 实验室张国凯学长的帮助下，我得以顺利地完成各项实验，每次我遇到困难向学长请教时，他会不厌其烦地给我讲解。学长的这种精神给我树立了良好的榜样，我也由衷的感谢学长学姐在我的学习生涯中给予我的各种支持和帮助。

最后，我要感谢各位参与评价或指导我文章的专家老师们，感谢各位的批评和指导，我也会虚心接受建议，在未来进一步提高自己的研究能力和文章水平！

装
订
线