第 3 章

CHAPTER 3

自动机器学习

3.1 引言

随着人工智能技术的快速发展，机器学习和深度学习已逐渐成为解决各种复杂问题的关键工具。在智能模型的开发与训练过程中，会存在大量参数调整、深度网络模型层级组合等耗时且繁杂的工作，而自动机器学习旨在采用数据驱动的方法，以任务为导向，实现模型开发过程中烦琐且耗时的重复性任务的自动化处理，如超参数调优和模型结构选择等。自动机器学习的发展很大程度降低了机器学习、深度学习的开发与使用门槛，同时将专家工作从繁杂的调参、架构选择等试错过程中解脱出来。本章将围绕自动机器学习的方法展开讨论，为读者提供前沿技术概述。

3.2 算法选择

在深度学习领域中，算法选择是决定模型表现优劣的关键因素之一[1]。不同的算法具有不同的架构和学习机制，因此在面对不同的数据集和任务时，表现会有所不同。例如，卷积神经网络[2]具有处理空间关系的能力而被广泛应用于图像处理任务；循环神经网络[3]能够捕捉数据的序列关系，从而在时间序列和自然语言处理任务中表现突出。事实上，不当的算法选择不仅可能导致模型性能下降，还会导致训练时间增加、计算资源浪费，甚至在某些情况下模型无法收敛，无法完成既定任务。因此，掌握每种算法的适用场景和局限性，以及任务数据的特性至关重要[4]。

算法的选择并非一个简单的流程，它涉及对任务本质的深入理解，以及对数据和模型特征的充分分析。

首先，数据集特征的分析是基础。数据集的大小、维度、数据类型（如文本、图像或时间序列数据）以及数据中的噪声程度，都会影响算法的适用性。例如，面对小样本数据集，深度神经网络可能会导致过拟合，此时基于传统机器学习方法或小型网络的迁移学习可能更加合适[5]。反之，对于拥有大量数据的应用场景，深度学习模型尤其是深层网络会更好地发挥其潜力。因此，选择算法前应对数据集的规模、数据质量及数据结构等方面进行详细分析，以便确定合适的算法类别。

其次，对于算法本身特性的了解是算法选择的重要依据，以各类深度学习算法为例，如前馈神经网络（Feedforward Neural Network，FNN）、卷积神经网络（CNN）[2]、循环神经网络（RNN）[3]及其衍生模型，如长短期记忆（LSTM）网络[6]和门控循环单元（Gated Recurrent Unit，GRU）[7]，它们均有独特的结构和适用领域。FNN 的结构简单，但其对数据的空间关系和序列信息捕捉能力较弱，适用于一些非时序数据。CNN 因其卷积操作的特性，对图像和视频数据具有强大的处理能

力。RNN 及其改进版本 LSTM 网络和 GRU 因具有记忆功能，适合处理连续的时间序列数据。然而，不同的算法不仅在架构上有所不同，它们在学习机制、计算复杂度、对硬件资源的要求上也有差异。例如，CNN 的卷积操作需要大量计算资源，而 RNN 需要更高的内存占用[8]。因此，应结合算法优缺点选择算法，并要同时考虑到硬件和时间等方面的资源限制。

最后，在评估算法选择效果时，合理的评估指标是必不可少的。通常情况下，深度学习模型的评估不仅关注其准确率，还要考虑其泛化能力、训练时间、资源消耗等因素。准确率高的模型若泛化能力弱，则在实际应用中可能表现不佳，尤其是在数据分布有所变化的情况下。此外，对于一些实时应用场景，模型的响应速度也至关重要，计算复杂度高的模型虽然精度可能更优，但在计算资源有限的情况下或许无法满足要求。评估指标的选择还应根据具体的任务需求进行调整。例如，在分类任务中，准确率、召回率、F1 值等是常用指标；而在生成模型中，图像生成质量、模型稳定性等也是重要的考量因素。通过系统地评估模型表现，能够更好地指导算法选择的优化过程。

算法选择贯穿机器学习模型开发的始终，决定了模型的适用性和性能表现。本节将系统讨论算法选择的各个方面，包括数据集特征分析、算法特征解读及评估指标的选取等。通过对这些内容的深入理解，将能够根据实际应用场景，有效地选择适合的算法，实现任务性能的优化。这不仅是 AI 算法应用成功的基础，也是推动智能技术发展的重要步骤。

3.2.1 特征选择

在深度学习和机器学习中，特征选择是数据处理的核心步骤。特征的适当选择不仅能优化模型性能，还能显著提高训练效率，帮助算法更好地理解数据的内在结构[9]。下面将从数据类型、数据集规模、数据分布、数据的特征选择与降维、数据的噪声与缺失值等的角度进行详细讨论[10]。

1. 数据类型

数据通常可以分为结构化数据和非结构化数据。结构化数据通常呈现为表格或二维矩阵形式，有明确的行列和数据类型。例如，表格中的每行可以代表一个数据实例，每列则对应一个特征。结构化数据适合使用传统的机器学习算法，如决策树、随机森林和逻辑回归。这些算法在处理小型数据集时表现更佳，同时由于其模型结构相对简单，往往具有较强的可解释性。

非结构化数据包括图像、文本和音频等，没有固定的格式。非结构化数据的处理通常需要借助深度学习模型。例如，CNN 的卷积层可以捕捉图像的局部特征，如边缘、纹理等，从而对复杂图像内容进行有效建模，更适合于处理图像数据；RNN 通过循环结构记忆前一时刻的状态，能够捕捉到序列信息的依赖性，因此更适合处理语音、文本、传感监督信号等时间序列数据。

2. 数据集规模

数据集的规模是影响模型选择的重要因素，不同的数据规模适合不同的算法。对于数据量较少的小型数据集，传统的基于统计学习的智能算法（如 K 最近邻、逻辑回归、朴素贝叶斯等）往往表现优异。这是由于深度学习模型参数量较大，数据规模小时容易过拟合，而传统机器学习模型则由于参数较少，能在小规模数据集上获得更优的性能。此外，传统统计学习模型往往具有较好的可解释性，有助于理解特征的重要性及其对模型预测的影响。而对于较大规模的数据集，深度学习模型（如卷积神经网络和递归神经网络）则会表现更好。例如，ImageNet[11] 等大型图像数据集在图像分类领域中成为标准测试集。ResNet[12]、Inception[13] 等深度学习模型在 ImageNet 上的表现表明，深度卷积神经网络能充分利用大量数据，提取出复杂特征，实现高精度的分类。对于深度学习模型，大规模数据不仅可以提供更多的样本用于特征学习，还可以缓解过拟合问题，因为数据的多样性能够增强模型的泛化能力。

3. 数据分布

数据分布是影响算法选择的关键因素之一，包含类别分布和特征分布。数据集中不同类别样本量差异较大时，即类别分布不平衡，模型的性能往往会受到影响。在这种情况下，模型可能会倾向于预测样本量较多的类别，导致小类别的预测效果较差[14]。例如，在欺诈检测中，正常交易和欺诈交易的比例严重失衡，导致模型难以正确学习与识别到异常以及欺诈交易的模式，预测性能从而受限。为应对类别不平衡问题，可以在选择算法时考虑使用综合评估指标，如 F1 值，而不仅仅是准确率，以更准确地反映模型的分类效果。同时，算法选择上也应偏向对不平衡数据敏感的模型，如支持向量机和决策树等。此外，还可以采取一些补充方法，如欠采样、过采样或利用生成对抗网络(GAN)[15]生成小类别样本来改善不平衡问题。此外，不同算法在面对不同的特征分布时表现各异。对于线性可分的数据，逻辑回归、线性支持向量机等线性模型通常能提供良好的效果；而对于复杂的非线性数据，决策树、随机森林等非线性模型更为合适。此外，深度学习模型在处理高维、非线性数据时具有优势，如 CNN 在图像分类中可以通过卷积层自动学习数据的复杂分布特性。此外，还可以使用数据标准化和归一化等预处理技术改善数据分布不一致的问题，使数据更符合模型的输入要求。

4. 数据的特征选择与降维

数据的特征选择与降维方法也是提高模型效率和精度的重要手段。对于高维数据，通过特征选择与降维，可以减少数据的冗余信息，降低模型的复杂度。在特征选择阶段，可以采用一些统计方法(如方差分析、卡方检验)或基于模型的特征重要性度量(如决策树的特征重要性评分)选择最具代表性的特征。特征选择可以显著减少输入数据的维度，减少计算开销，提高模型的泛化性能。选择合适的特征还能够增强模型的可解释性，有助于系统地解析各特征对预测结果的影响。当数据维度较高且特征间存在冗余时，降维技术如主成分分析、线性判别分析等可以有效地降低数据维度。降维不仅可以简化数据结构，还能缩短模型的训练时间。深度学习中，也可以通过自编码器等无监督学习模型实现非线性降维。降维后的特征子集仍需满足数据的代表性要求，因此需要仔细设计降维过程以确保重要信息的保留。

5. 数据的噪声与缺失值

数据的噪声和缺失值也是影响特征选择的重要因素。噪声数据会干扰模型的学习，降低模型性能。对于传统机器学习算法，可以使用一些方法去除或平滑噪声，如均值平滑、移动平均等。对于深度学习模型，可以采用数据增强或正则化技术减轻噪声的影响。当数据中存在缺失值时，需要根据任务需求和数据特征选择处理方法。例如，可以使用均值填充、中位数填充或插值等方法处理连续变量的缺失值；对于分类变量，可以使用众数填充或增加缺失类别等处理方法。处理缺失值时还需考虑特征的分布和缺失模式，确保处理结果不影响数据整体分布和模型性能。

综上，特征选择在数据预处理中占据核心地位，合理的特征选择不仅可以显著提高模型的性能，还能降低数据的维度和复杂度，使模型训练更加高效。本节详细讨论了特征选择在不同数据类型和特性下的实现方式，并结合机器学习和深度学习方法，分析了在不同数据集规模、数据分布和噪声处理方面的差异化方案。

3.2.2　算法评估

在构建深度学习和机器学习模型的过程中，算法评估是不可或缺的环节。通过评估可以量化模型的性能，从而判断其在实际应用场景中的有效性[16]。评估的准确性直接影响算法选择和模型优化的方向。不同的算法评估方法侧重于对不同的性能维度进行刻画，如准确性、稳定性、速

度、资源消耗等，这些指标能够全面评估模型的具体表现。如图 3.1 所示，在算法库中选择算法对数据集库中的数据集进行测试，根据模型表现为每个数据集匹配对应合适的算法。本节将介绍常用的评估指标及其适用的算法类型，帮助读者在模型开发过程中更加科学、客观地评估算法的优劣。

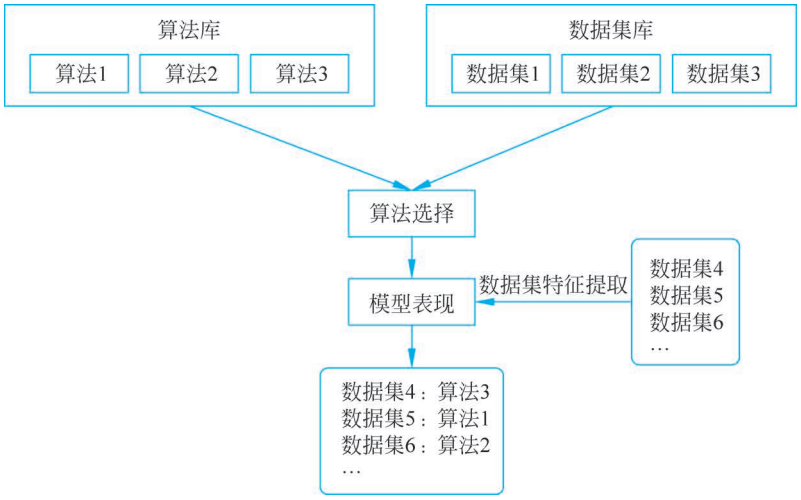

图 3.1 算法选择

在选择算法时，不同任务对模型的要求会有所不同，尤其是在训练效率、推理效率和模型可解释性方面。对于需要高效训练和快速推理的任务，通常选择简单的模型，如线性回归和决策树等传统算法。在需要更高准确率的任务中，则会考虑如深度神经网络等复杂模型以提供更强的特征表示能力，应对更加复杂的数据模式。对于智慧医疗、智能交通决策等涉及人身安全的 AI 任务，模型的可解释能力则至关重要，因此这类问题多使用随机森林等可解释性高的算法。此外，对于如金融风险评估等任务，既要求较高的预测准确率，又要求模型结果具有可解释性。在这种情况下，传统模型可能不够灵活，而深度学习模型的可解释性差可能无法满足需求。此时，可以考虑使用诸如模型无关的局部解释（Local Interpretable Model-Agnostic Explanations，LIME）[17] 等技术对深度学习模型进行后处理，增加其可解释性，帮助用户理解模型的决策过程。

1. 算法效率评估

在评估算法效率时，时间效率和空间效率是两个关键的维度。时间效率主要关注算法执行所需的时间，通常通过时间复杂度来衡量。时间复杂度反映了算法在输入数据规模增大时，执行时间的增长规律。然而，时间复杂度虽然能够提供理论上的预测，但实际运行时间也受到硬件平台、编程语言、数据存储等多方面因素的影响。因此，评估时通常还需要结合实验数据，测量算法在不同规模数据集上的执行时间，进而更加准确地理解其性能表现。空间效率则关注算法在执行过程中所消耗的内存空间，通常通过空间复杂度来描述。空间复杂度衡量了算法所需额外内存的变化趋势，随着输入数据量的增加，算法所需的内存会以某种方式增长。除此之外，内存访问模式和缓存局部性等因素也会影响空间效率。

实际应用中，时间效率与空间效率通常是相互制约的。优化其中一方面可能会导致另一方面的牺牲。因此，在评估算法的效率时，需要在时间和空间之间作出权衡。一般而言，对于某些内存受限的嵌入式系统、边缘计算设备上的应用程序等，可能需要更多地关注空间效率，而对于实时系统或大规模计算任务，时间效率可能更为重要。因此，综合考虑这两个维度的表现，有助于在不同

的应用场景中选择或设计最优的算法[18]。

2. 算法性能评估

为了准确评估算法的性能，针对不同类型的任务提出了不同的性能评估指标，下面将依次介绍准确率、精确率、召回率、F1 值、受试者工作特征（Receiver Operating Characteristic，ROC）曲线等。在介绍上述指标之前，需要先明确一下混淆矩阵中的元素，如表 3.1 所示。混淆矩阵是用于描述分类模型性能的工具，用于评估模型在不同类别上的预测准确性，显示了模型在正类和负类上的预测结果，包括下面四部分。

（1）真正例（True Positive，TP）：模型正确预测为正的样本数量。

（2）假反例（False Negative，FN）：实际为正，但模型预测为负的样本数量。

（3）假正例（False Positive，FP）：实际为负，但模型预测为正的样本数量。

（4）真反例（True Negative，TN）：模型正确预测为负的样本数量。

表 3.1　混淆矩阵

真实情况	预测结果	
	正　例	反　例
正例	TP（真正例）	FN（假反例）
反例	FP（假正例）	TN（真反例）

准确率是评估分类模型最常用的指标，表示模型正确分类的样本比例，适用于类别平衡的数据集，如式（3.1）所示。

$$\text{accuracy}=\frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \tag{3.1}$$

例如，在鸢尾花数据集[19]的分类任务中，准确率可以作为初步评估指标，但在类别不平衡的情况下需要结合其他指标。

精确率表示在所有被预测为正例的样本中真正为正例的比例，也称为查准率，如式（3.2）所示；召回率则表示在所有真实为正例的样本中，被正确预测为正例的比例，也称为查全率，如式（3.3）所示。对于类别不平衡的数据评估，单独使用准确率可能会误导决策，引入精确率和召回率可以更全面地评估模型性能。

$$\text{precision}=\frac{\text{TP}}{\text{TP}+\text{FP}} \tag{3.2}$$

$$\text{recall}=\frac{\text{TP}}{\text{TP}+\text{FN}} \tag{3.3}$$

F1 值是精确率和召回率的调和平均数，如式（3.4）所示。F1 值能够综合评估模型的性能，尤其适用于类别不平衡的情况。在医疗诊断中，F1 值常用于评估疾病预测模型的表现。

$$\text{F1}=2\times\frac{\text{precision}\times\text{recall}}{\text{precision}\times\text{recall}} \tag{3.4}$$

ROC 曲线常被用来评价二值分类器的优劣，即评估模型预测的准确度，绘制的是假阳性率（False Positive Rate，FPR）与真阳性率（True Positive Rate，TPR）的关系。如图 3.2 所示，绘制 100 个二分类样本的 ROC 曲线与曲线下面积（Area Under the Curve，AUC）的值。AUC 值表示模型的综合性能，AUC 值越接近 1，模型性能越好。例如，在信用卡欺诈检测中，使用 ROC 曲线分析不同阈值下的模型表现，有助于选择最优的决策阈值。

然而，对于多分类任务，上述常规的评估指标往往不能充分反映模型在各类别上的表现，尤其

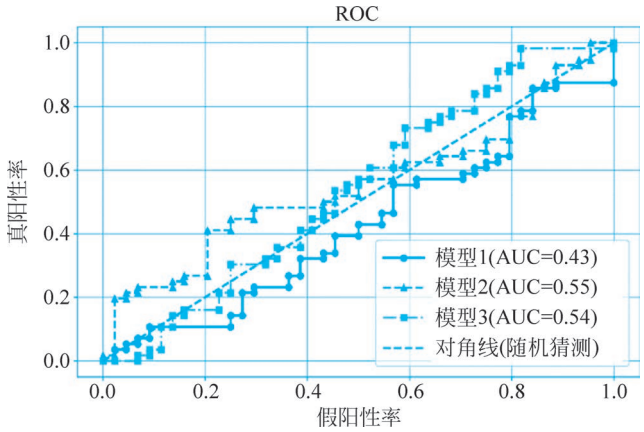

图 3.2 ROC 曲线示例

是在数据集中某些类别的样本数量极少的情况下。为了更好地评估模型在不平衡数据集上的性能，通常采用宏平均(Macro Average)和加权平均(Weighted Average)等方法。宏平均是通过分别计算每个类别的精确率、召回率或 F1 值，然后对这些指标进行算术平均得到的。这种方法的优点是不会因为某个类别样本数多而偏向该类别，而是对所有类别的评估给予同等权重。宏平均能够提供各个类别表现的公平视角，但在类别不平衡的情况下，可能会过于强调少数类的表现，从而忽略了大类别的优势。相比之下，加权平均则是根据每个类别的样本数量对各类别的评估指标进行加权求平均，这意味着样本较多的类别对整体评估的影响较大。这种方法能够更好地反映模型对大类别的表现，但可能会掩盖少数类别的性能，尤其是当少数类的预测效果较差时。

此外，在如信息检索和推荐系统等任务中，AI 模型输出结果的分数排序也尤为重要，而传统的精确率和召回率等指标难以刻画排序质量，因此引入了一系列基于排序的评价指标，如平均倒数排名(Mean Reciprocal Rank，MRR)、归一化折扣累计增益(Normalized Discounted Cumulative Gain，NDCG)等。这些指标在继承精确率和召回率基本思想的同时，进一步考虑了结果的排序位置，从而能够更有效地评估对排序敏感的算法性能。

在评估 AI 模型的过程中，考虑到深度学习等方法在训练过程中引入了随机梯度下降、概率采样等随机过程，模型的实际效果会产生一定的波动，给模型评估带来了稳定性与泛化性问题。交叉验证是一种用于评估模型稳定性和泛化能力的技术，能够有效减少因数据划分带来的偶然性影响。其核心思想是将数据集多次划分，分别用于训练和验证，从而更全面地评估模型的性能。常用的交叉验证方法之一是 K 折交叉验证(K-Fold Cross-Validation)。在 K 折交叉验证中，数据集被划分成 K 个大小相似的子集(或称“折”)。然后，交替地将其中一个子集作为验证集，其余 $K-1$ 个子集作为训练集，模型依次在每轮进行训练和评估。这一过程会持续 K 次，以确保每个子集都被用作一次验证集。记录每次验证的结果(如准确率、F1 值等)，最后计算 K 次结果的平均值，作为模型的整体评估指标，如图 3.3 所示，其中阴影框所代表的子集被用作验证集，验证每次模型训练的效果。通过 K 折交叉验证可以减少结果的偶然性，能够更高效地利用数据，获得更全面的性能评估，但是其计算开销较大，使得模型选择过程变得复杂。

除了 K 折交叉验证，还有几种常用的交叉验证方法，适用于不同的场景。留一交叉验证将数据集分成与样本数量相同的子集，每次只用一个样本作为验证集，其他样本用于训练。这种方法特别适用于数据集较小的情况，但计算开销较大。分层 K 折交叉验证用于分类任务中数据不均衡的情况。在分层 K 折交叉验证中，数据集在划分时会保持各个类别的比例一致，保证每个折中的

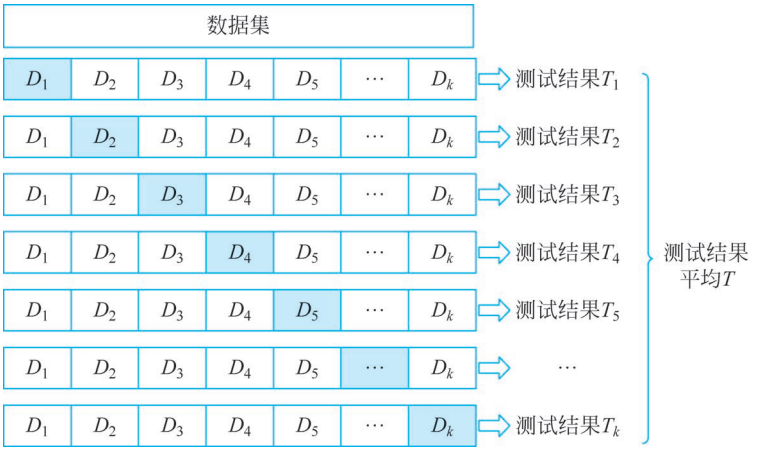

图 3.3　K 折交叉验证

类别分布与整体数据集一致，从而使模型评估结果更加可靠。时间序列交叉验证适用于时间序列数据，训练集和验证集的划分顺序严格按照时间排序，以保证模型能够反映实际应用中的时间依赖关系。

3. 其他评估指标

在算法评估中，除了常规的准确率、精确率和召回率等指标外，其他评估维度同样重要。尤其是在复杂任务中，衡量 AI 算法能否有效解决实际问题，不仅要关注性能指标，还需考虑模型的可解释性、稳健性等关键因素，以确保其在不同场景下的可靠性和适用性。其中，可解释性是重要评估方面之一，尤其在医疗、金融和法律等领域，模型不仅要求具备高精度，还需要提供透明、易于理解的决策依据供人类参考。为此，研究者提出了保真度、逻辑一致性、可解释度量和稳定性等指标量化模型的可解释性。保真度(Fidelity)是指模型的解释是否能够准确地反映其真实的决策过程。较高的保真度意味着解释能够真实地反映模型的预测行为。在局部可解释模型中，保真度可以通过比较模型原始输出与解释生成的预测结果的差异来衡量，如式(3.5)所示。

$$\text{Fidelity}=\frac{1}{n}\sum_{i=1}^{n}\left|f_\theta(x_i)-f(x_i)\right| \tag{3.5}$$

其中，$f_\theta(x_i)$是原始模型在输入 x_i 上的预测值；$f(x_i)$是解释模型在相同输入 x_i 上的预测值；n 是样本数量。

逻辑一致性(Logical Consistency)衡量的是模型的解释是否遵循一定的逻辑规则。例如，沙普利增量解释(SHapley Additive exPlanations，SHAP)[20]值常被用来衡量各特征对预测结果的贡献是否符合某种一致性原则，即每个特征的贡献值应该合理反映其在预测中的作用。若特征的贡献值出现异常波动，则说明解释模型存在逻辑不一致问题。如式(3.6)所示，如果两个输入样本的 SHAP 值差异过大，可能意味着模型在同样特征下的解释不一致。为了保持逻辑一致性，解释应该在相似输入下保持相对稳定。

$$\Delta\phi(i,j)=\left|\phi(f,x_i)-\phi(f,x_j)\right| \tag{3.6}$$

其中，$\phi(f,x)$表示在输入 x 上的 SHAP 值；x_i 和 x_j 是不同的输入样本。

可解释(Interpretability)度量指标旨在量化解释的清晰度和易懂性。通常，解释模型的复杂度越低，其可解释性越强。例如，基于决策树的解释往往比神经网络更易理解，因为决策树提供了明确的规则和路径。神经网络的解释需要通过复杂的技术(如 LIME[17]或 SHAP[20])进行，而这

些方法本身可能并不直观。一般而言，可解释度量可以通过用户评估的方式进行，即通过问卷调查或实验验证，评估用户对解释结果的理解程度。

稳定性(Stability)则用来评估模型解释结果是否会受到微小输入扰动的干扰。稳定的解释意味着在相似的输入条件下，解释结果应该保持一致，如式(3.7)所示。模型解释的稳定性是一个重要的质量指标，如果解释模型对输入的轻微变化过于敏感，可能会导致解释结果不可靠，难以被用户所接受。

$$\text{Stability}=\frac{1}{n}\sum_{i=1}^{n}\mid g(x_i)-g(x_i')\mid \tag{3.7}$$

其中，$g(x_i)$是对输入 x_i 的预测结果；$g(x_i)$是对扰动后的输入 x_i'的预测结果。

通常，可解释性指标相互关联，共同影响模型的可解释性。例如，较高的保真度通常伴随着较好的逻辑一致性，而可解释度量和稳定性则可以通过用户的反馈进一步验证。如果在保证模型性能的同时，能够通过优化这些可解释性指标，提高 AI 算法的透明度和可靠性，便能更好地使 AI 模型在高风险领域获得信任。

另一个重要的评估指标是生成质量，这一指标主要适用于生成模型，如生成对抗网络和变分自编码器等。生成质量衡量的是模型生成数据的真实性和多样性，通常通过一些标准来评估，如 Inception 分数(Inception Score，IS)[21] 和弗雷歇初始距离(Frechet Inception Distance，FID)[22] 等。IS 主要评估生成图像的多样性和质量，如式(3.8)所示。

$$\text{IS}=\exp(E_x[D_{\text{KL}}(P(y\mid x)\parallel P(y))]) \tag{3.8}$$

其中，$P(y\mid x)$是生成图像类别预测的分布，表示生成图像属于不同类别的概率；$P(y)$是所有生成图像类别的平均概率分布，反映了生成模型的多样性。

而 FID 则通过衡量生成图像和真实图像之间的特征差异评估生成质量，如式(3.9)所示。

$$\text{FID}=\parallel\mu_{\text{r}}-\mu_{\text{g}}\parallel^{2}+\text{Tr}(\sigma_{\text{r}}+\sigma_{\text{g}}-2(\sigma_{\text{r}}\sigma_{\text{g}})^{1/2}) \tag{3.9}$$

其中，μ_{r} 和 σ_{r} 是来自真实图像特征的均值向量和方差矩阵；μ_{g} 和 σ_{g} 是来自生成图像特征的均值向量和方差矩阵；Tr 表示矩阵的迹。

总的来说，模型的评估涉及的维度十分广泛，不仅仅局限于传统的效率、准确性等方面，还涉及模型的可解释性、生成质量等多个层面。这些指标在不同维度上的综合评估能够进一步帮助开发者以及自动化工具更全面地理解算法的优缺点，从而选择最合适的算法。

3.3 自动超参数优化

机器学习模型的性能不仅依赖于算法的选择和数据的质量，还深受超参数设置的影响。超参数是模型在训练前需要手动设定的参数，如学习率、正则化强度和网络结构参数等。这些超参数的选择往往直接决定了模型的学习能力、泛化能力和最终的性能表现。然而，手动调整超参数既耗时又需经验，且在面对复杂模型和高维超参数空间时，寻找最优超参数组合的难度急剧增大。因此，如何高效地进行超参数优化，成为提升机器学习模型性能的关键。

3.3.1 问题定义

自动超参数优化(Automated Hyperparameter Optimization)旨在通过算法化手段，自动搜索超参数空间，找到最优或近似最优的超参数组合。这种方法不仅提高了模型的训练效率，还在一定程度上降低了对领域专业知识的依赖。通过结合多种优化策略和工具，自动超参数优化为机器

学习应用提供了更为广泛和深刻的支持，尤其在大规模数据集和复杂模型中显示出不可或缺的价值。

3.3.2　免模型超参数优化

免模型超参数优化方法不依赖于模型的具体结构或训练过程，而是通过系统地探索超参数空间，以寻找最佳参数组合[23,24]。这类方法往往无须执行完整模型训练流程，而是依托性能指标的快速评估机制实现参数调优。下面讨论几种常用的免模型超参数优化方法。

1. 网格搜索

网格搜索(Grid Search)[25,26]是一种常用的超参数优化方法，通过系统地遍历预定义的超参数组合，以寻找实现模型性能最佳的参数配置。在免模型超参数优化的背景下，网格搜索的简洁性和易于实现使其成为许多机器学习任务中的首选策略，尤其是在计算资源充足时。网格搜索的基本思想是针对每个超参数设定一组离散的取值，然后生成所有可能的组合进行模型训练和评估。在实际操作中，网格搜索的步骤包括以下两方面。首先，确定要优化的超参数及其对应的取值范围。例如，对于深度学习模型，常见的超参数包括学习率、批次大小、正则化系数以及网络层数等；其次，生成超参数的组合，通过设定每个超参数的具体取值，网格搜索创建一个超参数空间，其中每个点代表一种模型配置。

网格搜索的优点在于其全面性和可解释性。通过穷举所有可能的超参数组合，网格搜索能够找到全局最优解，特别是在搜索空间较小的情况下。模型性能的提升通常可以通过实验结果直观反映出来，使得超参数优化过程透明且易于理解。此外，网格搜索的实现较为简单，许多机器学习框架都提供了现成的网格搜索工具，用户可以轻松地进行超参数调优。

然而，网格搜索也存在显著的缺点。由于其穷举式的特性，当超参数空间较大时，网格搜索的计算开销随超参数数量和取值范围的增加而迅速上升。在大多数实际应用中，超参数空间往往是高维的，这导致网格搜索可能需要数小时、数天甚至更长的时间才能完成。这样一来，计算资源的浪费和时间的消耗使得网格搜索在复杂模型和大规模数据集上的应用受到限制。尽管网格搜索在效率上存在局限性，但它在超参数优化中的重要性不可忽视。特别是在处理小规模数据集或模型时，网格搜索仍然能够提供可行的解决方案，帮助研究者快速识别最佳超参数配置。在实际应用中，网格搜索常与其他方法结合使用，形成混合优化策略，以便在保证性能的同时降低计算成本。

2. 随机搜索

随机搜索(Random Search)[27]是一种用于超参数优化的简单而高效的方法，通过在超参数空间内随机采样多个组合寻找模型的最佳配置。与网格搜索的穷举策略不同，随机搜索不会遍历所有可能的超参数组合，而是从每个超参数的预定义取值范围中随机选择若干样本进行评估。这种策略降低了计算成本，提升了优化效率，尤其在高维度或超参数空间较大时表现优异。随机搜索的核心思想是通过概率性探索取代确定性穷举。在实际操作中，需为每个超参数定义连续型或离散型概率分布。系统在搜索过程中随机抽取若干组超参数组合，训练模型并在验证集上评估其性能。随机搜索的灵活性使其能够跳过大量冗余组合，避免了网格搜索中相同超参数之间的重复计算。每次采样得到的模型配置都是独立的，因此在计算资源允许的情况下，随机搜索的过程可以并行化，显著提升搜索速度。

与网格搜索相比，随机搜索的主要优势在于其效率和适应性。在高维搜索空间中，网格搜索的计算开销呈指数级增长，而随机搜索能够在有限计算预算内找到高性能的模型配置。研究表明，当部分超参数对模型性能的影响较大，而其他超参数影响较小时，随机搜索能够比网格搜索更

快地找到优质解[27]。因为随机搜索更倾向于探索不同区域的超参数组合，而不是集中于某些固定的网格点。随机搜索还具有较好的扩展性，它可以轻松处理连续型和离散型超参数，并且适用于多种机器学习模型和深度学习网络的优化任务。例如，在优化卷积神经网络的过程中，随机搜索可以同时调节学习率、卷积核大小和层数，而不需要为每个参数设定严格的固定组合。由于每个采样都是独立的，随机搜索也非常适合分布式计算环境，在多个计算节点上同时评估不同的超参数组合。

尽管随机搜索在效率和灵活性上具有明显优势，但其效果依赖于采样数量和超参数空间的设计。在搜索预算有限的情况下，随机搜索的采样结果可能存在较大的方差，导致部分优质配置被忽略。因此，在实际应用中，用户通常需要根据计算资源和任务需求调整采样数量，以找到性能与资源之间的最佳平衡。

3. 树结构贝叶斯优化

树结构帕尔逊估计器（Tree-structured Parzen Estimator，TPE）[28-30]是一种基于概率模型的超参数优化方法，相较于网格搜索和随机搜索，它能够更智能地探索超参数空间。TPE 通过构建概率模型，对超参数空间中的不同区域进行密度估计，从而有针对性地选择更有潜力的参数组合。这种方法尤其适用于高维搜索空间和复杂模型，因为它可以动态地调整探索方向，提高优化效率。TPE 的基本思想是将超参数的采样过程视为一个贝叶斯优化问题。在每次迭代中，TPE 会根据已有评估结果更新参数的概率分布，重点探索那些在历史评估中表现较好的区域。具体来说，TPE 将超参数空间中的采样点划分为两部分：一部分是性能较优的点集，另一部分是性能较差的点集。通过对这两部分点集的概率密度进行估计，TPE 构建出一个条件概率模型，用于引导下一次的超参数选择。相比于随机搜索的无序采样，TPE 能够集中资源探索到更具潜力的参数组合。

在实现过程中，TPE 使用两种密度估计函数描述性能优良点集和差表现点集的分布。根据这些密度估计，TPE 计算出一个采集函数（Acquisition Function），用于选择下一个待评估的超参数组合。这个采集函数通过在参数空间中寻找使条件概率最大化的点，从而引导搜索过程不断向更优解靠近。由于每次采样都依赖于当前的模型评估结果，TPE 具有一定的自适应能力，能够根据任务进展动态调整搜索方向。TPE 优势在于其高效性和精确性，在高维超参数空间中，传统的穷举法和随机法容易陷入计算瓶颈，而 TPE 能通过概率模型的指导，快速缩小搜索范围。特别是在面对计算成本较高的模型（如深度神经网络）时，TPE 可以在较少的评估次数下找到性能接近最优的超参数组合。此外，由于 TPE 的优化过程是渐进式的，每轮迭代的结果都会用于更新概率模型，这使得它在长期的优化过程中表现稳定。

TPE 在多种机器学习和深度学习任务中得到了广泛应用。例如，在神经网络的超参数优化中，TPE 能够有效地调整学习率、网络深度、激活函数类型等关键参数，从而提升模型性能[30]。在自动化机器学习系统中，TPE 常用于复杂搜索任务，如集成学习模型的参数选择和混合模型的优化。TPE 的高效性使其成为超参数优化工具中的核心算法，并在大规模数据集和分布式计算环境中展现出良好的适应性。尽管 TPE 的计算复杂度比随机搜索略高，但它在评估次数有限的情况下能够显著提升搜索效果，这使得它特别适合计算资源受限的场景。在实际应用中，TPE 常被用作超参数优化流程中的关键组件，为用户提供更高效、更智能的优化方案。通过动态调整搜索策略，TPE 不仅能够减少不必要的计算，还能更快地找到高性能的超参数配置，使其在大规模模型优化中展现出重要价值。

4. 遗传算法

遗传算法（Genetic Algorithm，GA）是一种基于自然选择和生物进化启发的优化方法，在超参

数优化任务中通过模拟遗传和进化过程，逐步筛选出高性能的参数组合[31-35]。它将超参数的搜索视为种群演化的过程，通过选择、交叉和变异等操作优化参数配置，使得经过多代进化后的参数组合在目标任务上达到最优。

遗传算法在超参数优化中的基本步骤如下。首先，系统会随机初始化一个包含若干超参数组合的种群，每个个体（即参数组合）代表一种模型配置；接着，对种群中的每个个体进行评估，计算其适应度（Fitness），即在验证集上的性能表现；随后，系统根据个体的适应度选择优秀的个体作为父代，并通过交叉和变异操作生成新的参数组合，构成下一代种群。这个过程不断重复，直至找到最佳超参数组合或达到预设的迭代次数。

1）初始化与种群生成

初始化阶段生成多个超参数组合，组成初始种群。为了保证种群的多样性，系统通常会在各个超参数的取值范围内随机采样。例如，对于一个卷积神经网络的优化任务，种群中的个体可能包括不同的学习率、卷积核大小、层数和正则化系数的组合。

2）适应度评估与选择

在每代中，系统会评估种群中每个个体的适应度。适应度通常由模型在验证集上的性能决定，如准确率或损失值。适应度高的个体有更高的概率被选为下一代的父代。常见的选择策略包括轮盘赌选择（Roulette Wheel Selection）和锦标赛选择（Tournament Selection），它们分别基于概率和竞争决定哪些个体进入下一代。

3）交叉操作与新个体生成

交叉（Crossover）操作模拟生物遗传中的基因重组，将两个或多个父代个体的超参数部分组合成新的个体。典型的交叉方法包括单点交叉和多点交叉，例如可以将两个父代的不同层数或学习率组合成一个新的超参数配置。交叉操作使得种群能够保留父代的优秀特性，同时探索新的参数组合。

4）变异操作与多样性保持

变异（Mutation）操作通过随机修改个体的某些超参数值，引入新的多样性，以避免种群陷入局部最优。例如，在变异操作中，系统可能会将某个个体的学习率从 0.01 调整为 0.001，或改变卷积核的大小。在算法使用过程中，通常将变异的概率设置得较低，以确保搜索过程以父代的优秀特性为基础进行优化。

遗传算法的优势在于其强大的全局搜索能力和适应性，它能够在复杂的高维搜索空间中发现优秀的超参数组合，即使在搜索空间包含大量不规则或非凸结构时，遗传算法也能通过代际进化逐步逼近全局最优解。由于每代种群中的个体彼此独立，遗传算法还非常适合并行计算环境，可以将不同个体的评估任务分配给多个计算节点，进一步提升搜索效率。

在实际应用中，遗传算法已广泛用于神经网络的超参数优化和机器学习模型的参数调优。例如，在 CNN 中，遗传算法能够同时优化学习率、批次大小和层数，使得模型在多个目标上取得平衡。

5. 贝叶斯优化

贝叶斯优化[36-40]是一种基于概率模型的超参数优化方法，旨在通过构建代理模型，对超参数空间中的未知区域进行智能探索，以便找到性能最优的参数配置。与网格搜索和随机搜索不同，贝叶斯优化不会依赖于盲目的遍历或随机采样，而是通过利用已评估参数的结果来推断未来探索方向，从而在尽可能少的评估次数内找到高性能的超参数组合。这种方法在计算资源有限、评估代价较高的场景中表现尤为出色。贝叶斯优化的核心思想是将超参数优化问题视为一个黑箱优

化问题。在黑箱模型中，无法直接获取目标函数的显式形式，只能通过离散的超参数组合进行查询（即训练和验证模型），得到其在验证集上的表现。贝叶斯优化通过使用概率模型（如高斯过程）估计不同超参数组合的性能分布，并在此基础上选择最优的采样点进行下一轮评估[41]。其优势在于能够智能地平衡探索和开发，既探索未知的高潜力区域，又充分利用已有的评估结果。

1）代理模型与性能估计

贝叶斯优化的第一步是构建代理模型，用于估计超参数空间中的性能分布。最常用的代理模型是高斯过程，它能够捕捉超参数与模型性能之间的非线性关系，并提供不确定性估计。具体来说，高斯过程为每个待评估的超参数组合提供一个性能预测值和相应的不确定性，从而帮助系统确定下一步探索的方向。

2）采集函数与决策机制

贝叶斯优化的关键在于使用采集函数平衡探索和开发。常见的采集函数包括期望改进、最大值概率和上置信限。这些函数根据代理模型的预测结果和不确定性，选择那些既有可能提升模型性能，又具备较高探索价值的超参数组合。例如，期望改进函数会优先选择那些预测值较高且不确定性较大的超参数点，从而避免陷入局部最优。

3）迭代优化过程

贝叶斯优化采用迭代式的过程。在每轮迭代中，系统根据代理模型的预测和采集函数的结果，选择一个新的超参数组合进行模型训练和验证，并将评估结果反馈给代理模型用于更新。随着迭代的进行，代理模型会逐渐对超参数空间形成更精确的估计，从而提高采样点的质量。最终，当达到预设的评估次数或性能目标时，系统输出最优的超参数配置。

贝叶斯优化的主要优势在于其计算效率和智能探索能力。它通过构建性能代理模型和使用采集函数，有效减少了不必要的模型评估，特别适用于那些单次评估代价高昂的深度学习任务。在实践中，贝叶斯优化已广泛应用于神经网络的超参数优化，如用于调整学习率、正则化参数和网络结构的关键超参数。此外，贝叶斯优化在强化学习、自然语言处理等领域也展现了良好的适应性。由于贝叶斯优化依赖于代理模型进行性能预测，它在高维度和复杂的超参数空间中可能面临一些局限，如维度灾难[40]、样本效率[36]等。然而，改进版本的贝叶斯优化算法（如基于梯度的贝叶斯优化和分布式贝叶斯优化）正在逐步解决这些问题[42]，使其在大规模任务中得到更广泛的应用。

3.3.3 近似超参数优化方法

在现代机器学习实践中，面对大规模数据和复杂模型，超参数优化的计算成本和时间开销往往不可忽视。为了提高超参数优化的效率，近似超参数优化方法应运而生。这些方法通过借助智能策略，避免每次都进行完整的模型训练，从而实现高效的超参数调优。下面将详细介绍几种主要的近似超参数优化方法。

1. 早停

早停（Early Stopping）[43-45]是一种常用的超参数调优技术，旨在防止模型在训练过程中的过拟合。具体而言，在模型训练期间，持续监测验证集的性能。当验证集的性能在经过若干次迭代后不再提升时，便停止训练。这种方法不仅能够节省计算资源，还能避免因过拟合而导致的模型性能下降。在超参数优化过程中，早停策略可以与其他优化算法结合使用，作为评估模型性能的有效手段，减少不必要的训练次数。

2. 连续对半算法

连续对半算法（Successive Halving）[46]是一种高效的超参数优化方法，适用于大规模超参数

空间的快速探索。其核心思想是通过逐步缩小候选超参数的集合寻找最优解。具体步骤如下：首先，初始时，在超参数空间内随机选择多个候选组合，并对每个组合进行一定数量的训练；其次，根据验证集的性能，保留表现最佳的一部分组合，其他组合则被淘汰；接下来，针对保留的组合进行更多训练，并重复这一过程，直至最终得到最佳超参数组合。连续对半算法的优势在于通过逐步筛选，显著减少了模型评估的次数，使得在有限的计算资源下，能够更高效地探索超参数空间。

3. 超带

超带(Hyperband)[47,48]是一种结合了随机搜索和连续对半算法的超参数优化方法，旨在最大化资源的使用效率。该方法通过分配资源(如训练时间)评估超参数组合的表现。与连续对半算法不同，超带同时进行多组超参数组合的评估，快速淘汰表现不佳的组合，并将资源集中于表现良好的组合上。超带利用了多臂老虎机问题的思想，在不同的超参数组合之间分配资源，从而加速收敛，减少不必要的计算开销。这一方法在处理大规模超参数空间时展现出很大的灵活性和有效性，成为现代机器学习中一种颇具前景的优化策略。

4. 集成近似优化方法

集成近似优化方法[49]通过结合多种优化策略，进一步提升超参数优化的效果。这些方法的核心在于利用多种算法的优势，动态平衡不同算法的探索能力与局部最优解的风险。例如，可以将贝叶斯优化与随机搜索结合，利用贝叶斯模型的全局探索能力以及随机搜索的随机性防止局部最优解。通过这种集成策略，研究者能够在更广泛的超参数空间中进行高效探索，从而更有可能找到优质的超参数组合。

近似超参数优化方法为机器学习实践提供了高效的解决方案，尤其在面临大规模数据和复杂模型时。这些方法通过智能化的策略，不仅提高了超参数优化的效率，还在一定程度上降低了对计算资源的需求。随着机器学习技术的不断进步，未来的研究将继续探索更加高效、智能的超参数优化方法，以推动机器学习领域的进一步发展和应用。在此基础上，研究者和从业者可以更自信地开展各类机器学习任务，挖掘更深层次的数据潜力。

3.4　元学习

在传统的机器学习中，模型通常依赖于大量数据学习特定场景下的任务。然而，当应用场景发生变化时，这些模型往往需要重新进行训练，甚至可能无法有效适应新的数据。这种现象与人类学习的方式形成了鲜明对比。想象一下，一个小朋友在成长过程中接触了各种各样的物体和照片，即使他只看了几张狗的图片，也能够很快将狗与其他物体区分开来。这种快速学习和适应新任务的能力，正是元学习(Meta Learning)所追求的目标[50]。元学习也被称为“学习如何学习”(Learning to Learn)，其核心在于提升模型的适应性和泛化能力，使模型能够在有限的数据和经验下快速掌握新任务[51]。如图 3.4 所示，元学习通过在多个已知任务上进行学习，实现模型在新任务上的快速适应。

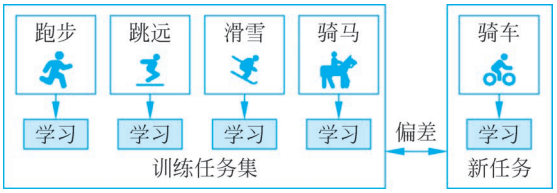

图 3.4　元学习

传统的机器学习和深度学习模型通常依赖于特定任务的大量数据进行训练，模型对这些特定任务的表现虽出色，但一旦面临新任务或新环境，就需要大量数据和重新训练保持效果，成本和时间花费都很高。而元学习通过“跨任务学习”将学习目标从解决单一任务提升为一种通用的“学习能力”，能够以高效、灵活的方式快速适应新任务，显著减少了重新训练的代价。

元学习通过构建一种更通用的学习机制，突破了传统模型在应对新任务上的局限，其目的是“培养模型的学习能力”，使智能算法不仅能够针对特定任务提出解决方案，还能从少量新数据中迅速提取有用信息并适应新的任务。例如，在图像识别任务中，传统的图像分类模型需要成千上万张样本图片的数据训练学习新类别的特征，而元学习模型只需少数几张示例图像即可达到相当的准确率，从而节省大量的标注成本和数据资源[52]。这种“少样本学习”(Few-Shot Learning)的能力使得元学习在数据稀缺、任务多变、个性化需求强的场景中表现出色。元学习的价值不仅体现在其理论上的创新性，也在应用上展示出很大的潜力，其思想成功应用于医疗诊断[53]、实时监控[54]、无人驾驶[55,56]、个性化推荐[57]等多个领域，显著地降低了对大数据的依赖并提升了适应能力。例如，在医疗领域，元学习通过多任务经验积累实现对新病例的关联学习，从而让机器具备类似于医生对新病例的快速适应与诊断能力[58]。在图像识别任务中，元学习方法使模型能够在有限数据下准确识别新类别，这在医疗、安防和自动驾驶等需要少样本学习的场景中表现尤为突出。此外，元学习在自然语言处理中的快速语言适应和迁移学习也逐步得到应用，如通过记忆过去的语言结构来推断新的语言模式，或通过在多语言任务上的训练来增强语言转换能力，使模型在新语言上具有较强的迁移能力。

本节将从三个主要方面对元学习进行介绍：基于优化的元学习方法、基于模型的元学习方法和基于度量的元学习方法[59]。基于优化的元学习方法专注于改进学习算法本身，以提高模型在新任务上的学习速度和准确性。基于模型的元学习方法则通过设计特定的学习架构，使得模型能够更有效地提取和利用先前的经验，进而应对新的挑战。而基于度量的元学习方法则强调通过比较样本之间的相似性，实现快速学习和高效分类。

从更广泛的角度来看，元学习的崛起为 AI 的发展带来了新的方向，即不再仅依赖于数据量的增加和模型复杂度的提升获得更好的性能，而是更多地强调模型的学习适应能力。在未来的应用场景中，元学习将推动机器学习从大数据时代向“小数据”时代迈进，帮助智能系统高效、迅速地适应新任务和新场景，减少数据标注成本和训练时间，显著提高 AI 系统的应用效率[60]。进一步，本节将更深入地讨论元学习的三大核心方法，并详细介绍其在图像识别、自然语言处理、机器人学习等多领域的应用前景。

3.4.1 基于优化的元学习方法

基于优化的元学习方法旨在通过改进学习算法和训练过程，使模型能够在新任务上快速适应。与传统的机器学习方法依赖于大量数据进行训练不同，基于优化的元学习方法强调的是在少量数据的基础上提升学习效率。这种方法通常通过优化学习过程中的参数更新规则，加速模型的适应能力。

一个典型的基于优化的元学习方法是模型无关元学习(Model-Agnostic Meta-Learning，MAML)[61]。MAML 的核心思想是通过在多个任务上进行训练，使得模型能够通过少量的梯度更新就适应新任务。具体而言，是指训练一组初始化参数，模型通过初始化参数，仅用少量数据就能实现快速收敛的效果。为了达到这一目的，模型需要大量的先验知识不停修正初始化参数，使其能够适应不同种类的数据。MAML 训练过程中使用了一组任务，每个任务都从一个共享的初始化参数开始。

模型在每个任务上进行多次更新，之后再对这些任务的性能进行评估。通过这种方式，模型学会了如何在任务之间共享知识，从而在面对新任务时，能够迅速调整参数，达到较高的准确性。

MAML 的优势在于其通用性和有效性。它可以应用于多种类型的模型和任务，无论是分类、回归，还是强化学习等。此外，MAML 已在许多实际应用中表现出色，如图像分类、自然语言处理等领域[62]。

另一个重要的优化方法是自适应学习率调整[63]。在基于优化的元学习方法中，模型可以根据不同任务的特点动态调整学习率。这种自适应机制使得模型能够在面对不同的任务时，选择最合适的学习策略，从而加快收敛速度并提高性能。具体实现方式是利用梯度信息估计任务的难度，从而使模型根据任务的特性动态选择学习率。对于难度较大的任务，模型使用较小的学习率，以避免在优化过程中产生较大的波动；而对于简单的任务，模型则可以使用较大的学习率加快收敛速度；自适应学习率的实现也可以通过不同的优化算法来达到。例如，使用 Adam[64]、RMSprop[65]等自适应优化算法，这些算法能够根据历史梯度信息自动调整学习率，从而实现更快速地收敛。学习率调度策略可以在训练过程中动态调整学习率。例如，可以在训练的初期使用较大的学习率，在模型逐渐收敛时逐步减小学习率，以确保模型能够找到全局最优解。

基于优化的元学习方法还包括几种其他策略，如记忆增强的优化方法，这些方法通过引入记忆机制，使得模型能够保留和利用过去的经验，以便在新任务上更快地学习[66]。此外，记忆增强的优化方法也能够结合迁移学习的理念，在多个相关任务上进行训练，可以提高模型的泛化能力。通过共享不同任务间的知识，模型能够在新任务上更快地收敛，提升整体性能。

总之，基于优化的元学习方法为机器学习模型赋予了快速适应新任务的能力，这一能力在现实世界中具有广泛的应用前景。无论是在医疗诊断、个性化推荐还是自动驾驶等领域，基于优化的元学习都能显著提高模型的灵活性和效率，使其能够在变化莫测的环境中迅速响应并作出决策。通过不断优化学习过程，这些方法为未来智能系统的设计和实现提供了新的思路。

3.4.2　基于模型的元学习方法

基于模型的元学习方法是元学习领域的重要分支之一。其核心思想是通过设计具有灵活结构和适应性的模型，使其在处理新任务时能够快速更新和调整[67,68]。与其他元学习方法相比，基于模型的元学习方法更加注重模型的结构设计和信息存储机制，以便满足不同任务需求[69]。这种方法特别适用于少样本学习和在线学习环境[70,71]，因为它能够在数据有限的情况下快速适应新任务，从而在图像识别、自然语言处理等领域展示出显著的应用潜力[72]。

1. 基于模型的元学习步骤

在元学习的框架下，基于模型的元学习方法通常会包含一个具有记忆功能的结构，如特定的神经网络组件或外部记忆单元，使模型能够“记住”先前任务的知识，并在新任务上迅速进行调整。通过任务间的共享和任务内的微调，这些模型能够在多种任务场景中表现出很高的适应性。通常，基于模型的元学习过程可以分为以下几个步骤。

(1) 任务表示和学习。将多个任务表示为任务集合$\{T_1, T_2, \cdots, T_n\}$，通过少样本数据集定义每个任务。任务的数据通过元模型进行训练，以期在新任务上也能取得较好的表现。

(2) 模型设计和更新机制。基于模型的元学习方法依赖于模型的结构设计，使其能够在少量数据下快速更新。典型的做法包括添加可更新的权重模块、任务特定的参数调整机制，以及集成外部记忆模块来增强适应性。

(3) 任务快速适应和微调。在新任务上进行微调，基于模型的元学习方法一般会利用先前任

务的学习经验，通过少量迭代将模型参数调整到最优状态，快速实现高精度预测。

2. 典型模型与应用

在基于模型的元学习方法中，有几种广泛应用的模型，包括动态权重生成模型、神经注意力机制和存储强化模型。下面将讨论这些代表性模型及其特点。

动态权重生成模型通过动态生成或调整权重，实现模型在不同任务间的快速切换。一个典型的例子是元网络(Meta Network)，该模型通过一个元网络生成任务特定的权重，以适应每个新任务的数据分布[67]。元网络的核心思想是为每个任务生成不同的模型参数，类似于快速的参数调节机制，使模型能够在不同任务之间进行切换。例如，在图像识别中，当模型面对新类别的物体时，可以利用少量新类别的图像生成适应当前任务的权重，从而实现精准的识别[71]。这种动态权重生成模型还在其他领域得到了广泛应用。例如，在自然语言处理领域，生成不同的权重可以让模型在不同的文本分类任务中快速适应新的语义模式，而无须大规模数据微调。通过将任务特定的知识嵌入模型中，动态权重生成模型能够在数据不足的情况下表现出良好的泛化能力。

神经注意力机制赋予模型对任务数据的重点关注能力，使模型能够有选择性地提取任务相关信息。一个典型的应用是 Meta-LSTM 模型，在其结构中融入了长短期记忆网络，以便于记忆先前任务的信息[73]。当遇到新任务时，模型可以通过记忆机制选取关键特征，快速适应新的任务需求。神经注意力机制在机器人学习中显示出很大的潜力。机器人在执行不同任务时，如抓取物体或避障，需要快速适应新环境的变化。通过在任务之间共享注意力权重，机器人可以识别并记忆特定的场景模式，从而在新任务中表现出更高的适应性。例如，机器人可以通过观察先前任务中的操作，将注意力集中在新任务的关键步骤上，从而快速执行新操作。

存储强化模型将外部存储单元集成到模型结构中，以便为模型提供一个长期记忆模块，使其在处理多个任务时具有较强的记忆和泛化能力。这类模型中最著名的例子之一是记忆增强神经网络(Memory-Augmented Neural Networks，MANN)[66]，利用神经网络与外部存储单元的结合，使模型能够在多个任务之间进行知识共享。当遇到新任务时，模型能够从存储单元中读取相关的信息，并迅速适应新任务。

在计算机视觉任务中，基于模型的元学习方法可以显著提高模型对新类别的适应能力，特别是在少样本环境下。通过在存储单元中保存不同类别的特征表示，模型能够快速将输入图像与存储特征进行匹配，从而实现准确分类。元学习方法(如元网络[67]和 MANN 模型[66])能够在少量标注数据下，通过记忆或动态生成权重，快速适应新任务。这种结合存储和元学习的策略提高了模型的扩展性和稳健性，使其在小样本学习任务中表现出色，并能够更好地匹配新类别特征，提升识别精度。此外，在视频跟踪中，基于模型的元学习可以实现更优的在线更新效果[74]。

在自然语言处理领域，基于模型的元学习方法同样显示出强大的适应能力。在图像分类、情感分析和机器翻译等任务中，基于模型的元学习方法可以在少样本的条件下进行微调和快速适应[68]。这样的能力为模型在多领域、多语种的迁移学习提供了新的可能性。

在机器人学习中，基于模型的元学习方法为机器人快速适应新任务提供了有效的途径。基于模型的元学习方法通过记忆先前任务的关键特征或生成适应新任务的权重，使机器人能够在短时间内掌握新任务的技能。通过存储强化模型，机器人可以将之前任务中的特征嵌入外部存储单元中，在执行新任务时快速调用相关信息。这种方式特别适用于动态环境或未知环境，如在灾难救援、搜索和导航等任务中，机器人可以基于过去任务经验快速调整和优化行为模式，展现更高的学习和适应能力[75]。

3.4.3　基于度量的元学习方法

基于度量的元学习(Metric-Based Meta-Learning)方法是元学习领域的另一个重要方向[76]，其核心思想是通过学习一个适应性强的度量空间，使模型能够在不同任务之间进行有效的特征匹配和相似性计算。这种方法通常不需要显著调整模型参数，而是通过构建适当的度量标准，使模型能够快速地对新任务的数据进行分类或预测。基于度量的元学习方法具有实现相对简单、效率高的特点，尤其适合少样本学习和零样本学习等场景。

1. 基于度量的元学习步骤

在元学习的框架下，基于度量的元学习模型通过学习一个特定的度量空间度量新样本与已有样本之间的相似性。该方法通常涉及下面几个步骤。

(1) 任务表示和样本构建。将任务表示为多个不同类别的少样本集合，称为支持集(Support Set)，并引入待分类或预测的查询集(Query Set)；通过在支持集样本之间建立相似性度量模型，可以将查询集中样本与支持集进行对比，完成任务。

(2) 度量空间学习。基于支持集样本，模型学习一个度量空间，能够量化新样本和支持集样本之间的距离或相似性。常见的度量空间包括欧氏距离、余弦相似度等，模型可以通过这些度量快速进行样本间的相似性计算。

(3) 少样本预测和分类。当模型完成度量空间的学习后，可以通过在查询集中寻找与支持集最相似的样本类别完成预测。模型无须通过大量数据更新参数，而是直接依赖学习的度量空间，能够在新的任务上快速获得准确结果。

2. 典型算法与应用

1) 三种典型算法

基于度量的元学习方法中最经典的算法包括原型网络[77]、匹配网络[78]、关系网络[76]等。其中，原型网络是一种简单而高效的基于度量的元学习方法，该方法假设每个类别都可以由支持集样本的特征均值表示为一个原型向量[77]。具体而言，对于一个任务内的每个类别，模型首先计算该类别样本的均值向量，并将其视为该类别的原型。在分类过程中，查询集中的每个样本都与每个类别的原型计算距离(通常为欧氏距离)，将其分配给距离最小的类别。原型网络的优势在于，它不依赖大量样本训练参数，而是通过一个简单的原型向量就可以表示每个类别，从而适用于少样本场景。该方法特别适用于图像识别任务，如在半监督任务中，原型网络可以有效地对发生的变化或新出现的类别进行快速识别[79]。

匹配网络是一种利用度量学习的神经网络，通过计算查询样本和支持集样本之间的相似度实现分类。不同于原型网络，匹配网络采用了注意力机制，允许模型在每个任务中为支持集中的每个样本分配权重，从而对查询集样本进行分类[78]。具体而言，匹配网络通过一种“上下文嵌入”机制，将每个支持集样本和查询集样本嵌入同一个度量空间，然后通过余弦相似度或其他度量方法，计算查询样本与支持样本的匹配得分。匹配网络的创新之处在于引入了注意力机制，使模型可以在不同任务中适应不同的类别分布。其效果尤其适用于零样本或小样本的分类任务，如文本匹配和图像分类等场景。通过注意力机制，模型能够根据每个任务的特征自适应地调整度量标准。

关系网络不同于前两种方法，关系网络不是直接使用传统的距离度量(如欧氏距离或余弦相似度)，而是通过一个神经网络学习样本之间的关系[76]。具体而言，关系网络首先使用卷积神经网络将支持集样本和查询样本映射到高维特征空间，然后通过一个关系模块(通常是全连接网络)预测查询样本与支持集样本之间的相似性分数。这种方法的优势在于，通过关系模块的学习，模

型可以自动找到支持样本和查询样本之间的最佳关系特征，适用于复杂的样本关系匹配任务。关系网络在图像识别、视频分类等领域表现出色，因为其关系模块可以灵活地捕捉图像或视频帧之间的复杂相似性关系。

2）三种典型应用

在图像识别领域，基于度量的元学习方法因其高效的相似性度量能力，被广泛应用于少样本和零样本分类任务[71]。传统的深度学习模型往往需要大量样本来训练，但基于度量的元学习方法可以通过学习到的度量空间，在少样本的情况下实现对新类别的分类。以原型网络为例，在图像分类任务中，模型通过支持集构建每个类别的原型向量，当遇到新的查询样本时，模型可以快速地根据样本与原型的距离进行分类。这种方法显著减少了标注数据的需求，并且在新类别频繁出现的应用场景中表现出色。

在自然语言处理领域，基于度量的元学习方法同样展现出强大的适应性。在文本分类、情感分析和问答系统等任务中，基于度量的方法通过度量空间量化文本之间的相似性，从而实现快速的少样本学习。基于度量的元学习方法在多语言文本匹配和文本相似度计算方面也表现出色。尤其是在多语言环境下，匹配网络可以通过预训练嵌入模型将不同语言的文本映射到同一个度量空间，从而实现跨语言的文本匹配和分类。这种方法能够帮助模型在不同语言和领域之间进行快速迁移，提升语言模型的跨领域适应性。

在机器人学习领域，基于度量的元学习方法为机器人快速适应新环境和新任务提供了有效的工具。机器人通常需要在不同场景中执行任务，如物体抓取、路径规划等。传统的深度学习方法往往需要大量数据进行训练，而基于度量的元学习方法能够通过少量样本快速完成新任务的学习。关系网络在机器人学习中的应用尤其广泛，机器人可以利用其关系模块识别不同物体或场景之间的相似性，从而快速适应新任务。通过这种方式，机器人可以更灵活地适应动态环境，提高操作的精确度和效率。

基于度量的元学习方法具备多个显著优势。首先，基于度量的模型具有较高的效率，能够在不需要大量训练样本的情况下实现新任务的快速适应。其次，这些方法对任务参数调整的需求较低，通过度量空间的学习即可完成样本匹配，非常适合少样本学习任务。此外，基于度量的元学习方法结构简单，计算效率高，在推理过程中能够快速处理查询样本和支持集样本的相似性计算，从而降低了计算成本。

尽管基于度量的元学习方法具有较高的适应性和效率，但也面临一些挑战。首先，度量空间的构建比较依赖数据分布，当支持集样本数量过少或分布不均时，模型的泛化能力可能受到影响。其次，不同任务之间的特征差异较大时，固定的度量空间可能无法适应所有任务的需求，导致模型在新任务上的准确性下降。

3.5 神经架构搜索

随着深度学习的发展，神经网络的性能高度依赖于其架构的设计。然而，手工设计神经网络不仅费时费力，还需要大量的专家知识。因此，神经架构搜索（NAS）逐渐成为一种新兴的自动化技术，旨在自动探索神经网络架构，以替代传统人工设计过程[80]。这一方法不仅能减少人工干预，还能发现比人工设计更优的网络结构。NAS 的目标是在给定的任务和数据集上自动搜索出最佳的神经网络架构，从而优化性能，如分类准确率、模型复杂度或推理速度。

在 NAS 的框架中，搜索过程通常包含三个核心部分：搜索空间（Search Space），定义候选架构

的所有可能组合；搜索策略(Search Strategy)，探索候选架构以找到最优解的方法；评价策略(Evaluation Strategy)，评估候选架构的质量。

3.5.1　搜索空间

在NAS中，搜索空间是决定整个搜索过程效果和效率的关键因素，它决定了架构搜索的范围、复杂度和灵活性。如果搜索空间设计不合理，即使有再好的搜索算法也难以找到最优的架构。因此，如何合理地设计和优化搜索空间，已成为NAS研究中的关键问题。

1. 搜索空间的基本概念

在数学上，搜索空间可以被视为一个高维离散优化空间。每维代表一个决策变量，如某一层的操作类型、网络的层数或参数选择。这些变量之间的组合构成了神经网络架构的所有可能形式。搜索空间的设计决定了NAS算法的灵活性和效率。更大、更复杂的搜索空间允许找到潜在更优的架构，但也会导致搜索时间和资源的显著增加。搜索空间需要在灵活性与计算复杂性之间取得平衡。

以卷积神经网络的NAS为例，搜索空间可能包括卷积层、池化层、归一化层和激活函数等基本构件，并定义这些构件的堆叠顺序、连接方式及其超参数设置。NAS算法在这个多维空间内搜索，选择最优的模型架构以满足特定任务需求。如图3.5所示，①和②代表了两种不同的神经网络连接模式，③表示的是两种模式的组合。从图中可看出，①和②构成了一个小型的NAS搜索空间，而③则是NAS搜索出来的其中一种神经网络的组合架构。

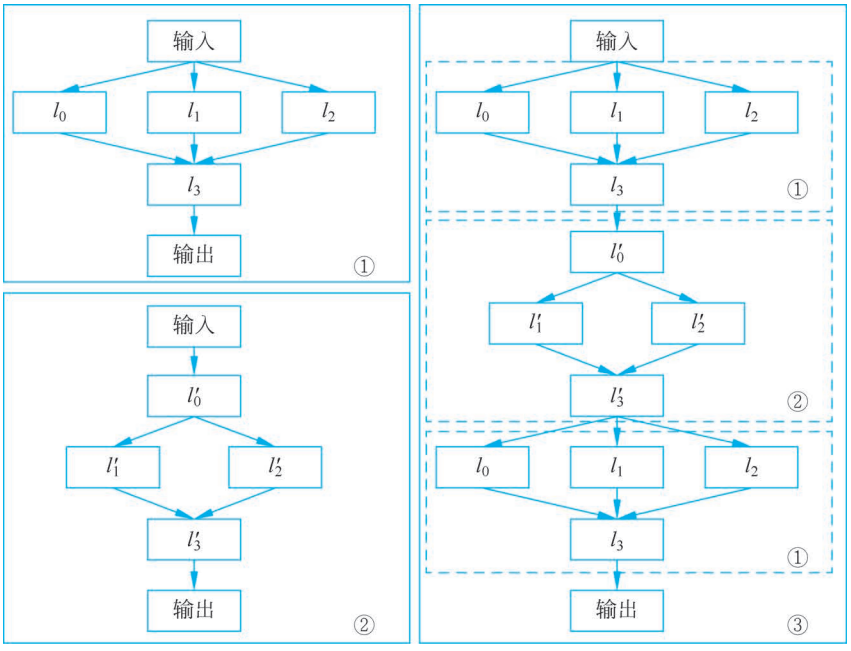

图3.5　搜索空间

2. 搜索空间的基础构成

搜索空间的基础构成决定了神经架构搜索系统可以生成和优化的架构类型。它涵盖了网络模型的基本组件、层级设计、超参数选择以及连接方式，这些要素共同定义了网络的结构与功能。合理的基础构成不仅确保了搜索空间的覆盖范围，还能避免冗余和无效架构的探索，提高搜索的效率和质量。

1）基础操作与层类型

NAS的候选架构通常由多种基本操作构成，这些操作在网络的不同层级上负责特定的功能。卷积层用于提取局部空间特征，是图像处理任务中的核心组件；池化层通过下采样减小特征图尺寸，减少计算开销；跳跃连接和残差连接则改善梯度传播，缓解深层网络的梯度消失问题。此外，归一化层（如批归一化、层归一化等）和激活函数（如ReLU、Sigmoid）在稳定训练过程和引入非线性方面也发挥着重要作用。NAS系统通常在这些基础操作中进行选择和组合，以找到满足任务需求的最佳架构。

2）连接方式与拓扑结构

层与层之间的连接方式决定了信息如何在网络中流动，不同的拓扑结构会显著影响模型的表达能力和训练效率。顺序连接是一种简单且直观的方式，将各层依次堆叠，但这种方式容易导致深层网络中的梯度消失。跳跃连接（残差连接）通过跨层连接减缓了梯度消失问题，改善了训练效果。密集连接通过让每层接收所有之前层的输出，进一步增强了信息流动和特征复用。NAS系统在搜索过程中需要探索不同的拓扑结构，以找到在任务上表现最优的连接方式。

3）模块化设计与基于单元的搜索

模块化设计是NAS中常用的策略，它将网络分为多个可重复使用的单元，从而降低搜索空间的复杂度。每个单元都是一个小型网络模块，包含多层基础操作和连接方式。NAS系统通过优化这些单元的结构和排列组合构建完整网络。这种基于单元的设计不仅减少了搜索维度，还提高了架构的可扩展性。

4）参数空间与超参数设置

除了网络的结构设计，超参数的选择也是搜索空间的重要组成部分。常见的超参数包括卷积核大小、学习率、批次大小和正则化系数等，这些参数直接影响模型的训练速度和性能。在NAS中，一些系统不仅优化网络结构，还同时优化超参数，形成联合搜索空间。此外，卷积核大小等架构参数也在搜索空间中占据重要地位。例如，在某些任务中，较大的卷积核更有利于捕捉全局特征，而较小的卷积核则更适合提取局部信息。超参数空间的设计需要在灵活性和可控性之间取得平衡，以避免搜索维度过大而增加计算负担。

搜索空间的基础构成不仅影响模型的性能上限，还决定了NAS算法的搜索效率。在实践中，合理的基础构成需要综合考虑任务需求和计算资源的限制。例如，对于需要实时推理的嵌入式系统，搜索空间可能更倾向于轻量化架构和低资源消耗的操作；而在云端任务中，搜索空间则可以包含更多复杂的连接方式和大型卷积核，以追求最高的模型精度。通过合理设计搜索空间的基础构成，NAS系统能够在性能和计算资源之间找到最佳平衡，为后续的架构搜索奠定坚实基础。

3. 搜索空间的类型与特性

搜索空间依据其组成形式的不同，大致可以分为四类，每种类型的搜索空间在设计和应用中具有独特的特性和优势，能够满足不同的需求。下面将针对性介绍这几类搜索空间。

1）全结构搜索空间（Entire-Structured Search Space）

全结构搜索空间是最简单直观的搜索空间之一[81,82]。在全结构搜索中，NAS算法尝试从头开始构建完整的神经网络架构。这意味着需要对网络的每层和每个连接方式进行选择和优化。虽然整个结构很容易实现，但它也有一些缺点。例如，人们普遍认为，模型越深，其泛化能力就越强；然而，寻找这样一个深层网络的工作繁重且计算成本高昂。此外，生成的架构缺乏可移植性；也就是说，在小数据集上生成的模型可能不适合更大的数据集，这就需要为更大的数据集生成新的模型。因此，全结构搜索常用于小规模任务或在资源充裕的条件下进行研究。

2）基于单元的搜索空间（Cell-Based Search Space）

基于单元的搜索空间通过设计可重复使用的模块降低搜索的复杂度。NASNet[83]和ENAS[82]是这种方法的经典案例。它们通过搜索单元的结构，并将其堆叠形成完整网络，从而避免了全局搜索的计算瓶颈。这种方法在计算效率和性能之间取得了良好平衡，因此常用于大规模数据集和实际应用。

3）层次化搜索空间（Hierarchical Search Space）

层次化搜索空间将网络架构视为多个层次的组合。每个层次都可以独立优化，逐步构建复杂的网络结构，能够显著降低搜索空间的复杂度，并使模型适应不同层次的特征表示需求。

4）形态变换搜索空间（Morphisms-Based Search Space）

形态变换搜索空间是一种从现有网络结构出发，通过小幅调整（如改变层数、调整卷积核大小或修改连接方式）生成新模型的策略。这种方法特别适合在已有成功架构的基础上进行微调优化。它能够有效利用已有的知识，并快速找到更优的模型。

合理的搜索空间设计对提升NAS系统的效率和效果至关重要。研究者在设计搜索空间时，既要考虑任务的复杂性和特性，也要平衡计算资源与搜索精度之间的关系。随着NAS技术的发展，未来的搜索空间可能会更多地采用混合策略，将不同类型的搜索空间优势结合起来，以应对更加复杂和多样化的任务需求。

4. 搜索空间优化与约束设计

为了提高NAS的效率和效果，搜索空间的优化与约束设计至关重要。过于庞大的搜索空间虽然能够涵盖更多潜在的优质模型，但会导致计算成本的急剧上升，并增加搜索算法陷入局部最优解的风险。因此，合理优化搜索空间的大小和复杂度，同时通过适当的约束排除无效或不合理的架构，是提升NAS系统性能的重要策略。

1）搜索空间的规模控制

大规模搜索空间带来了显著的计算开销，因此需要对搜索空间的规模进行合理控制，同时确保搜索方向更加明确，避免不必要的计算资源浪费。常用的缩减搜索空间的方法有以下几种。

（1）基于经验的先验设计。根据任务特点和已有经验，限定搜索空间的范围。例如，在图像分类任务中限制卷积核大小为3×3或5×5，从而减少无意义架构的生成。

（2）模块化与分层设计。通过将网络拆分为多个单元或模块，并仅优化这些基本模块的结构来简化搜索。

（3）随机采样和代理模型。在大规模搜索空间中，系统可以通过随机采样初步探索潜力区域，然后集中资源对这些区域进行深入优化。代理模型也常用于大规模搜索空间的初期筛选。

2）合法性约束与结构优化

为了避免生成无效的网络结构，搜索空间通常需要设计合法性约束，从而能够排除无意义的架构设计，确保搜索空间中的每个候选模型都是合理且可训练的。常见的约束有以下几种。

（1）拓扑约束。限制层与层之间的连接方式，防止出现无效回路或信息断层，如禁止在某些层之间出现循环依赖。

（2）层数和宽度限制。控制网络的层数和每层的节点数，以保证生成的网络在计算复杂度上可控。

（3）类型兼容性约束。确保不同类型的操作能够正确组合。例如，某些池化操作可能不适用于特定的特征图尺寸，在搜索过程中需要排除这种组合。

3）多目标优化设计

在实际应用中，NAS 系统不仅需要优化模型性能，还要同时考虑资源消耗（如计算时间、内存占用、能耗等）。此外，NAS 还需具备适应多样化部署场景的能力，使其能够针对不同设备部署既轻量化又高性能的模型，以满足不同应用需求。因此，搜索空间的设计通常需要支持多目标优化，这里讨论两种代表性方法。

（1）加权多目标优化。为不同目标分配权重，将它们合成为一个综合评分函数；在此框架下，NAS 系统可以在性能和资源消耗之间找到最佳平衡点。

（2）帕累托前沿优化。通过绘制性能与资源消耗之间的帕累托前沿曲线，系统选择在多个目标上均衡表现的模型，这种方法广泛应用于嵌入式系统和移动设备的模型优化。

4）自适应约束设计

在搜索过程中，自适应约束设计能够根据任务需求和搜索结果动态调整约束条件。初期阶段，系统可能采用较宽松的约束进行广泛探索，而在后期阶段逐步收紧约束，集中优化表现良好的区域。自适应约束设计使搜索过程更加灵活，避免因固定约束导致的潜在优质架构被遗漏。

通过搜索空间的优化与约束设计，NAS 系统能够在性能和计算资源之间找到最佳平衡，提高搜索效率并确保模型质量。合理的搜索空间优化不仅减少了不必要的探索，还能引导系统更快速地找到优质模型。

3.5.2 搜索策略

在 NAS 中，如何高效地探索庞大的搜索空间是关键问题。搜索策略决定了算法如何在给定的搜索空间内找到性能优异的架构，同时控制计算成本和时间开销。由于搜索空间的复杂性和维度的高阶性，NAS 算法需要在探索（Exploration）和开发（Exploitation）之间取得平衡。这意味着 NAS 既要能发掘尚未探索的新颖框架，也要充分利用已知的有效设计优化现有结果。不同的搜索策略各有其特点和适用场景，如强化学习（RL）、进化算法（Evolutionary Algorithm，EA）、基于梯度的搜索、随机搜索，以及多种混合策略的结合。本节将从这些策略的原理、应用场景、优势和挑战等方面进行详细讨论。

1. 基于强化学习的搜索

RL 是一种常用于 NAS 的搜索策略[83,84]。它将网络架构的搜索过程建模为一个序列决策问题，其中智能体（Agent）通过与环境交互逐步学习最优策略，以生成高性能的神经网络模型。在 RL 框架下，一个策略网络（Policy Network）生成一系列决策，这些决策决定了每层网络的操作类型及其连接方式。例如，策略网络可能选择在某一层使用 3×3 卷积或最大池化，并决定该层与前一层之间是否添加跳跃连接。这些决策组合成一个完整的网络架构。RL 算法通过反复试验调整策略，每次生成的架构都经过训练，并根据其在验证集上的表现更新策略网络的参数。

NAS 的目标是找到能够在特定任务上取得最佳性能的神经网络架构，而强化学习为此提供了系统化的搜索策略。在强化学习框架中，网络架构的设计被视为一个马尔可夫决策过程（Markov Decision Process，MDP）。其中，状态（State）表示当前生成的部分网络架构；动作（Action）表示在当前状态下选择的网络操作（如添加一个卷积层或池化层）；奖励（Reward）评估当前生成架构在验证集上的性能（如精度或损失）；策略（Policy）由策略网络生成，用于指导下一步的架构设计。强化学习的过程可视为一种迭代试错，每次生成一个新架构，评估其性能，并根据结果更新策略网络的参数。

Zoph 等于 2016 年首次提出使用 RNN 控制器执行 NAS[81]。在这种方法中，RNN 控制器作

为策略网络，逐步生成神经网络的每层的配置，包括操作类型（如卷积或池化）、卷积核大小以及连接模式。生成的架构随后被训练和验证，验证集的精度作为奖励信号反馈给 RNN 控制器。基于策略梯度方法（Policy Gradient），RNN 控制器的参数通过梯度上升法更新，以提高未来生成优良架构的概率。这种方法的优点在于其高度自适应性，即 RNN 控制器能够随着搜索过程的推进不断调整策略，生成越来越好的架构。此外，RNN 能够处理长时间依赖关系，因此非常适用于多层复杂网络的设计。

强化学习中的一个重要问题是如何在探索和开发之间取得平衡。探索是指智能体尝试新架构，以获取更多未知信息；开发则是指智能体集中搜索已知表现良好的架构，以进一步提升性能。为了控制这两者之间的关系，常用的策略概括如下。

（1）ε-贪心策略（ε-Greedy Strategy）：智能体以一定概率 ε 随机选择动作进行探索，其余时间则选择已知的最佳策略，这种方法能够有效避免陷入局部最优。

（2）基于策略梯度的优化：NAS 中的策略网络通常使用策略梯度算法进行更新。具体来说，通过对架构表现（奖励）与动作概率的乘积进行梯度上升，可以逐步提高生成高性能架构的概率。

（3）熵正则化（Entropy Regularization）：为了进一步鼓励探索，可以在损失函数中加入熵正则化项，使得智能体在搜索初期更加倾向于多样化探索。

RL 策略适用于处理复杂的网络设计任务，尤其是多层网络中的长时间依赖（如 RNN）。然而，RL 的计算成本较高，每次生成的新架构都需要完整训练，因此往往需要大量计算资源。为了降低成本，一些方法采用权重共享机制，即不同架构共享同一组权重，从而减少训练时间。强化学习为 NAS 提供了一种系统化的优化策略，能够在复杂的搜索空间内高效探索。然而，其高计算成本和收敛速度慢的缺点也限制了其应用范围。未来，随着更高效的 RL 算法和资源感知优化技术的发展，RL 在 NAS 中的应用有望进一步扩展。

2. 基于进化算法的搜索

进化算法是一类受自然选择和生物进化启发的优化算法。在 NAS 中，进化算法模拟种群的繁衍与竞争，通过代际更新不断优化架构[85-87]。与强化学习不同，进化算法更注重群体多样性和全局探索能力，适用于高维复杂搜索空间。NAS 中使用的进化算法包括遗传算法、遗传规划（Genetic Programming）、差分进化（Differential Evolution）等。

在进化算法框架中，候选网络架构被视为个体（Individuals），这些个体组成一个种群（Population）。通过不断地选择、交叉和变异，种群中的个体逐代优化，最终找到在特定任务上性能最佳的架构。

进化算法的优势在于其并行搜索能力：多个候选架构同时参与进化，可以在大规模搜索空间中高效探索多种可能性。AmoebaNet[87]将遗传算法应用在 NAS 中，通过多代进化优化了网络层的结构和连接方式。每代中，种群中的架构都会接受变异和交叉操作，并根据适应度选择进入下一代，最终在 ImageNet 分类任务上表现优异，证明了遗传算法在大型神经网络设计中的有效性。

由于每次迭代都需要评估大量架构的性能，进化算法在计算效率上面临挑战。NAS 中的适应度评价通常通过训练候选架构并在验证集上测试来实现。然而，这种完全训练的方式成本很高，因此研究者提出了多种优化策略，如权重共享，即在不同架构共享同一组模型权重，从而减少每次训练的时间开销[82]。或者使用代理模型预测架构性能，避免对每个候选模型的完全训练。例如，基于高斯过程的代理模型可以快速估计新架构的适应度。此外，常见的早停机制在训练过程中，如果发现某个候选架构的性能无法达到预期，可以提前停止训练，节省计算资源。

进化算法的优势在于可以同时探索多个候选架构，降低陷入局部最优的风险。并且它可以适

应大规模搜索空间，在处理复杂的搜索任务，特别是在多目标优化场景下表现出色。但是与此同时，由于需要评估大量候选架构，进化算法的计算开销较大。同时，随着种群的扩展，管理和追踪每代的候选架构变得困难，需要高效的存储和计算资源应对这些挑战。基于进化算法的搜索策略为 NAS 提供了强大的全局优化能力。它通过模拟生物进化过程，在复杂的搜索空间内高效探索高质量架构。尽管进化算法的计算成本较高，但通过权重共享、代理模型和混合策略等优化方法，其应用范围不断扩大。

3. 基于梯度的搜索

基于梯度搜索的 NAS 方法通过将搜索空间参数化为连续形式，使梯度下降法能够直接用于优化网络架构。传统的 NAS 方法依赖于离散搜索空间，需要对每个候选架构进行独立训练，而基于梯度搜索的 NAS 方法通过连续化操作，将候选架构的选择与权重训练过程结合，显著提高了搜索效率。其核心思路是将搜索问题转换为一个可以求导的优化问题，使得在训练网络权重的同时，优化出最优的网络结构。

在基于梯度的搜索框架中，所有可能的网络操作都会被赋予一个可微的参数。这些参数控制不同操作在网络中的重要性，如卷积层、池化层或跳跃连接的权重。在训练过程中，通过反向传播计算这些参数的梯度，并更新它们的值，使得整个网络架构逐渐趋向于最优状态。可微架构搜索（Differentiable Architecture Search，DARTS）[88]是这种方法的代表，它将所有候选操作嵌入一个超网络（Supernet）中，并通过优化超网络的权重和架构参数来进行搜索。DARTS 的搜索过程主要包括以下几个步骤：①定义搜索空间并将所有候选操作映射到超网络中；②构建参数化搜索空间，将每层的候选操作与一个概率分布关联；③通过梯度下降同时更新网络权重和架构参数；④在完成搜索后，根据优化出的参数选择最优架构，并对其进行独立训练。相比于离散搜索，DARTS 避免了对每个候选架构的独立训练，显著降低了计算资源的消耗。

基于梯度搜索的 NAS 方法适用于大规模任务，如 ImageNet 图像分类，其高效的搜索过程使其能够在较短时间内完成架构优化。然而，这种方法也面临一些挑战：由于采用了连续化的搜索空间，DARTS 可能会陷入局部最优解，难以找到真正的全局最优结构；此外，DARTS 假设超网络中所有候选操作都能够同时学习，这可能导致网络的训练动态变得复杂，出现梯度冲突的问题。针对这些问题，研究者提出了一些改进策略。例如，部分研究引入了正则化技术，以防止架构参数的过度拟合[89]；也有研究采用多级 DARTS，将搜索过程分为多个阶段，每个阶段优化不同的层次结构，从而避免单一搜索过程中的局部最优[90]。此外，权重共享技术也被用于进一步减少搜索时间和资源开销，使在资源受限环境中部署 NAS 成为可能。

在实际应用中，基于梯度的搜索已成功应用于卷积神经网络、循环神经网络和图神经网络的架构设计[91]。基于梯度的搜索在提升搜索效率的同时，也提出了新的挑战，如超网络的构建和训练复杂度的管理。未来研究的重点将是如何更好地处理梯度冲突，以及如何将这种方法扩展到更多类型的任务和搜索空间中。同时，结合其他优化策略（如进化算法和强化学习），混合搜索方法可能成为下一代 NAS 的主流方案，为网络架构设计提供更强大的支持。

4. 基于代理模型的搜索

基于代理模型的搜索（Surrogate Model-Based Search）是提升 NAS 效率的重要方法。传统 NAS 的挑战在于需要对每个候选架构进行完整的训练和验证，这种做法在大规模搜索空间中会消耗大量计算资源。代理模型通过构建一个性能预测模型，快速估计候选架构的表现，减少了对完全训练的依赖，从而显著降低搜索时间和成本[92]。这一方法已广泛应用于计算机视觉、自然语言处理等领域，尤其适用于任务复杂、搜索空间庞大的场景。代理模型的核心思路是通过历史数据

中的评估结果学习架构特征与性能之间的关系，对新生成的架构进行快速预测。

1）代理模型构建与架构搜索过程

在构建代理模型时，需要首先将候选架构转换为适当的特征表示。这些特征可以包括网络的层数、卷积核的大小、层之间的连接方式等。例如，一个卷积神经网络的特征可能是“3×3 卷积层＋最大池化层＋全连接层”这样的结构序列。代理模型通过学习这些特征与验证集表现之间的映射关系，从而能够推断出未训练架构的性能[93]。常见的代理模型包括高斯过程、随机森林、贝叶斯神经网络以及深度学习回归模型等，其中高斯过程尤为常用，因为它能够量化预测的不确定性，帮助系统更好地管理探索与开发的平衡。

2）基于代理模型的搜索效率讨论

代理模型的准确性和稳健性直接影响 NAS 的搜索效率。在搜索初期，代理模型通常会出现一定程度的偏差，因此需要通过动态更新不断优化。在每次架构评估后，将新的性能数据添加到训练集中，以便模型逐步改进其预测能力。这种动态更新机制有助于提升代理模型的精度，尤其是在搜索空间复杂、候选架构多样性大的情况下。一些研究还提出了混合代理模型的策略，即同时使用多个模型（如高斯过程与随机森林）进行性能预测，并通过集成多个预测结果减小误差[38]。这种多模型融合的方法能够有效提高预测的稳健性。

虽然代理模型减少了计算资源的消耗，但其构建和训练也存在挑战。首先，代理模型需要大量的历史数据作为训练集，而获取这些数据本身可能代价高昂。此外，在高维搜索空间中，代理模型的预测精度可能会下降，尤其是在架构多样性较高的情况下。为了解决这些问题，研究者提出了一些优化策略，如使用不确定性驱动的探索方法，即优先验证代理模型预测不确定性较大的架构，以改善模型在高风险区域的表现。此外，还有研究探索如何通过元学习预训练代理模型，使其在少量数据的情况下也能提供准确的预测。

尽管代理模型大幅提升了搜索效率，但仍存在一些亟待解决的问题。随着搜索任务的复杂性增加，如何提高代理模型的泛化能力，确保其在大规模搜索空间中的准确性，是一个重要研究方向。此外，将代理模型与其他搜索策略（如强化学习和进化算法）结合，有望在更复杂的多目标优化任务中取得突破。

基于代理模型的搜索为 NAS 提供了高效、灵活的架构优化方案。在未来的发展中，随着模型精度和资源管理能力的提升，这一方法有望进一步拓展其应用范围，为更复杂的任务和场景提供支持。在与其他优化方法的融合中，代理模型的价值将得到进一步放大。

5. 混合搜索策略

混合搜索策略结合了多种搜索方法的优点，在 NAS 中展现了强大的灵活性与适应性。单一的搜索方法，如强化学习、进化算法或代理模型，往往在特定任务或特定条件下表现良好，但面对复杂、多目标的任务时可能存在局限。混合搜索策略通过整合不同方法的优势，有效平衡全局探索与局部开发，提升搜索效率和架构质量。这类方法尤其适用于大规模搜索空间、多目标优化以及需要在有限资源内完成的搜索任务。

1）混合搜索策略框架优势

混合搜索策略的核心在于通过合理地分工与协作，将各个子策略的强项发挥到极致。例如，在强化学习与进化算法的结合中，强化学习负责指导全局搜索方向，提供策略网络生成的架构初步设计，而进化算法则对这些设计进行局部优化与多样化处理[80]。具体而言，在这种组合中，NAS 系统可以利用强化学习高效找到潜在优良架构，然后通过进化算法进一步迭代改进这些架构，确保不会陷入局部最优。此外，混合方法能够在系统训练的不同阶段切换策略，根据任务进展

动态调整搜索方案。另一种常见的混合搜索策略是代理模型与强化学习的融合[81]。在这种框架下，代理模型通过性能预测引导强化学习的搜索过程。代理模型快速筛选出可能表现优异的架构，减小 RL 控制器生成无效架构的概率，从而提高搜索效率。在搜索初期，代理模型帮助强化学习缩小搜索空间，而随着训练数据的积累，RL 控制器逐步接管搜索任务，进行更加精细的探索与优化。混合策略在这种动态交替过程中，实现了初期快速筛选与后期精确优化的良好平衡。

混合搜索策略的优势还体现在多目标优化上。在实际应用中，NAS 系统通常需要在模型性能、计算时间、能耗等多个目标之间取得平衡。单一的搜索策略往往无法同时兼顾所有目标，而混合搜索策略通过分工合作，实现多目标之间的动态权衡。例如，进化算法善于处理多目标优化问题，可以通过构建帕累托前沿，找到在性能与资源之间的最佳折中点；而强化学习则负责实时调整搜索方向，确保在不同优化目标之间灵活切换。这样的多目标优化框架在嵌入式设备、移动计算和云服务等场景中展现了广泛的应用潜力。混合搜索策略在 NAS 的实践中已有多项成功应用。例如，在高效神经架构搜索(Efficient Neural Architecture Search，ENAS)网络[82]中，系统通过权重共享减少了架构评估的计算成本，并结合强化学习的策略生成能力实现了高效搜索。在 AmoebaNet[87]中，进化算法与强化学习的结合使得系统能够在探索过程中持续改进架构，最终在 ImageNet 等大型数据集上取得了领先的分类性能。此外，一些研究通过将代理模型与梯度优化相结合，使 NAS 系统能够在高维搜索空间中快速找到高性能架构[80]，并通过梯度优化进一步提高模型精度。

2) 混合搜索策略框架挑战与建议

然而，混合搜索策略的设计也面临挑战。首先，不同搜索方法之间的协作需要精确的策略切换与资源分配，否则可能导致系统效率降低甚至产生冲突。其次，混合搜索策略中的信息交互和同步机制对系统性能至关重要，如何高效管理这些交互，避免不同子策略之间的冗余操作，是一个重要的研究课题。最后，由于混合搜索策略往往涉及多种算法的联合优化，NAS 系统的复杂性和调试难度也相应增大，需要更加完善的管理和监控机制。为了进一步提升混合搜索策略的效果，研究者提出了多种优化方案，如引入自适应混合策略框架，根据搜索过程的进展动态调整不同算法的权重和优先级；还有使用元学习技术预训练各子策略模型，使得系统在初期就能高效搜索。此外，分布式混合搜索策略也在 NAS 中得到应用，通过将不同子策略分配到多个计算节点并行执行，进一步提升了搜索效率。混合搜索策略为 NAS 提供了更强大的灵活性和适应性，使其能够在复杂、多变的任务环境中保持高效。未来，混合搜索策略有望在更多领域展现其潜力，尤其是在多任务学习[94]、联邦学习[95,96]以及资源受限的嵌入式系统[97-99]中；同时将进一步探索如何实现更精细的策略融合，推动混合搜索策略在 NAS 中的广泛应用。

3.5.3 评估策略

评估策略在神经架构搜索中起着至关重要的作用，它决定了如何衡量候选架构的表现，并为搜索过程提供反馈。在大多数情况下，NAS 需要在庞大的搜索空间内评估大量的架构，因此评估过程的效率直接影响整个搜索的速度和效果。然而，准确的评估通常伴随着高昂的计算成本，这使得性能与资源消耗之间的权衡成为 NAS 中的关键挑战。因此，设计合理的评估策略不仅要追求结果的精度，还要考虑时间、资源的开销以及实际应用中的约束条件。一个理想的评估策略应满足几个要求：第一，应能在不影响模型性能的前提下准确衡量候选架构的表现；第二，应尽量减少计算资源和时间的消耗，以应对大规模搜索任务的需求；第三，应具备适应不同任务的灵活性，如小规模数据集的精确评估与大规模数据集的快速筛选。此外，对于嵌入式系统和移动设备等受

限资源的应用场景，评估策略还需要考虑模型的推理速度、能耗以及存储需求等。

1. 完全训练评估

完全训练评估是一种最直接且精确的评估策略，即对每个候选架构都进行完整的训练和测试。在此过程中，候选模型从初始状态开始，经过完整的训练周期，并在验证集或测试集上进行性能评估。这种方法能够提供接近真实任务表现的结果，确保搜索到的网络架构具备强大的泛化能力。因此，完全训练评估通常被视为 NAS 中的黄金标准。然而，这一策略的计算开销很高，尤其是在处理大规模数据集或复杂网络架构时。

完全训练评估的过程通常包括以下几个步骤。首先，针对每个生成的候选网络架构，从头初始化所有参数。接着，在完整的数据集上训练该网络，训练轮次、学习率、批次大小等超参数的设定通常与实际部署时保持一致，以确保模型性能的可靠性。训练完成后，在验证集或测试集上评估该模型的表现，并将性能指标（如准确率、F1 值或损失值）反馈给搜索算法。NAS 系统根据这些评估结果指导下一轮的架构生成与筛选。这种评估方法的最大优势在于结果的准确性。由于每个候选架构都经过完整的训练，其性能评估几乎没有误差。对于规模较小的任务，如 CIFAR-10[100]、Fashion-MNIST[101] 等数据集，完全训练评估能够在合理的时间内完成，并为搜索提供高精度的反馈。

然而，完全训练评估的局限性也非常明显。当面对大规模数据集（如 ImageNet）或复杂架构（如深层卷积网络、Transformer 网络）时，这一策略的时间和资源消耗会成倍增长。例如，在 ImageNet 数据集上，对每个候选架构进行完整训练通常需要几天甚至几周的时间，即使使用高性能 GPU 集群也难以支撑大规模 NAS 任务。这样的计算开销使得完全训练评估在实际应用中的适用性显著降低。为了解决这一问题，研究者提出了一些优化措施。例如，可以在小规模数据集上进行初步搜索，并将筛选出的架构迁移到更大数据集上进行完整训练[102]。虽然这种做法在一定程度上减少了计算成本，但也存在跨数据集迁移时性能下降的风险。此外，还有一些研究尝试通过早停机制加快完全训练评估的速度[103]。在训练过程中，如果某个候选架构在早期的训练阶段表现不佳，则提前停止训练，从而节省计算资源。然而，这一策略也需要谨慎使用，因为提前终止可能会漏掉一些在后期表现良好的架构。

尽管完全训练评估的计算成本高昂，它仍然在一些特定场景下具有重要价值。首先，在验证新型搜索算法的有效性时，完全训练评估是不可或缺的基准。其次，在小规模或资源丰富的实验环境中，它能够为 NAS 系统提供最准确的性能反馈。此外，在最终模型的筛选阶段，开发者往往会对经过多轮筛选的优质架构进行完全训练，以确保模型在实际部署中的稳定性和可靠性。未来，随着 NAS 技术的发展，完全训练评估可能会与其他高效评估策略结合使用。例如，在搜索的初期阶段采用低保真评估快速筛选出一批候选架构，随后再对这些架构进行完全训练，以获取精确的性能反馈。混合使用评估策略不仅能提高搜索效率，还能在一定程度上保留完全训练评估的准确性。此外，随着计算硬件的发展和分布式训练技术的进步，完全训练评估的时间成本有望进一步降低，使其在更多场景中得到应用。

综上，完全训练评估是一种准确性最高的评估策略，适合在规模较小的任务中使用，或用于验证最终模型的性能。尽管其计算成本较高，但在某些关键场景下仍然不可替代。通过与其他评估策略的结合，完全训练评估将继续在 NAS 系统中发挥重要作用，并为发现高性能网络架构提供坚实的支撑。

2. 低保真评估

低保真评估[104,105]是一种通过缩短训练时间或简化训练过程快速评估候选网络架构的方法。

与完全训练评估相比，低保真评估牺牲了部分训练的完整性和精确性，换取更低的计算成本和更快的评估速度。这种方法适用于大规模搜索任务或资源受限的环境，如在处理大数据集、深层网络时的初期筛选阶段。低保真评估关键在于如何在减少计算开销的同时，最大限度保持与完全训练结果的相关性，以确保筛选出的架构能够在最终训练中表现良好。

低保真评估可以通过多种方式实现，最常见的方法包括减少训练轮次、缩小数据集规模、简化网络结构和降低模型复杂度。例如，在评估一个架构时，只使用原数据集的一个子集进行训练，这不仅减少了数据加载和计算的时间，也能快速筛选掉表现不佳的模型。此外，降低训练轮次是一种常见策略，通过减少训练的迭代次数，模型的初步性能可以在较短时间内得到验证。尽管这种方法可能无法完全展现模型的最终表现，但它能为后续筛选提供有价值的参考。使用较浅的网络结构进行评估也可以实现低保真评估。在这种策略下，搜索系统会优先评估浅层模型，并根据其表现推断更深层网络的潜力。这种做法在多层卷积神经网络中尤为常见，因为浅层网络的训练速度更快，可以快速探索架构设计的多种组合。此外，某些研究还采用代理数据集进行评估，即在规模较小但特征分布与目标任务相似的数据集上进行初步搜索，筛选出的优质架构再迁移到目标数据集上进行完整训练[106]。尽管跨数据集迁移可能引入一定误差，但这种方法在缩短搜索时间方面效果显著。

低保真评估的优势在于其计算效率高、适用性广，特别是在搜索的初期阶段能够快速筛选掉性能不佳的模型。在大多数情况下，低保真评估的结果与完全训练结果具有较高的相关性，这使得它成为大规模 NAS 任务中不可或缺的策略之一。然而，由于训练过程的简化，低保真评估的结果难免存在一定的误差。这种误差可能导致某些潜在优质架构在初期筛选中被误判为不佳，或者筛选出的模型在完全训练时表现不如预期。为了解决这一问题，研究者提出了多阶段的低保真评估方案[107]。在多阶段评估中，系统会逐步增加训练资源和训练复杂度。例如，第一阶段使用子集数据和少量训练轮次进行初步筛选，第二阶段对表现较好的模型增加数据量或训练轮次，最终在筛选出的优质架构上进行完整训练。这种分阶段的策略能够在保证筛选效率的同时，减少误差带来的影响。

低保真评估方法可以快速筛选大量候选架构，仅对少数表现优异的架构进行完整训练，显著地提升了搜索效率。此外，在资源受限的移动设备上，低保真评估能够帮助系统快速选择计算成本较低且性能良好的模型，实现模型在实际场景中的快速部署[108]。在某些时间敏感的任务中，如实时检测或预测，低保真评估还能显著缩短模型迭代周期，提升系统的响应速度。尽管低保真评估在 NAS 任务中展现了其独特价值，但仍然面临一些挑战。首先，不同任务和数据集之间的低保真评估效果可能存在差异，这需要针对具体任务调整评估策略。其次，如何设计更精确的低保真评估指标，使其尽可能接近完全训练的结果，是一个需要进一步探索的问题。最后，低保真评估的结果是否具备足够的稳健性，也影响到后续筛选与搜索的质量。总之，低保真评估是解决 NAS 计算瓶颈的重要策略之一，它以高效、灵活的特点在大规模搜索任务中占据了重要地位。

3. 一次性架构搜索评估

一次性架构搜索评估[109]是一种通过共享权重的方式，在同一训练过程中同时评估多个候选架构的策略。这种方法大幅地减少了 NAS 中的计算成本，使得在大规模数据集中也能快速探索复杂的空间。一次性架构搜索的核心思想是在一个超网络中嵌入所有候选架构，并通过共享参数的方式避免对每个架构进行独立训练。该策略首次在 ENAS[82] 中得到成功应用，并逐渐成为 NAS 领域的重要方法。

在一次性架构搜索评估中，超网络的构建是关键。这个超网络包含了搜索空间内所有可能的

架构组合，每个候选架构都是超网络的一部分。在训练过程中，不同架构通过选择超网络的子路径共享相同的参数。具体来说，超网络的每层包含多个操作，如 3×3 卷积、5×5 卷积、最大池化等。候选架构通过激活特定路径中的操作组合，形成独特的网络结构。这种权重共享机制显著减少了训练所需的时间和资源，使得搜索算法能够在较短时间内评估大量候选架构。一次性架构搜索的优势在于其计算效率高、训练时间短。与传统的 NAS 方法相比，这种策略避免了对每个候选架构的独立训练，从而将原本需要数天甚至数周的搜索任务缩短为几小时甚至几分钟。这一特性使得一次性架构搜索成为处理大规模数据集时的理想选择。此外，通过共享权重，系统能够利用少量的数据和资源快速找到性能良好的架构，特别适用于资源有限的环境。

然而，一次性架构搜索也面临一些挑战。所有候选架构共享同一组参数，不同架构之间的梯度会相互干扰，导致权重更新的冲突。这种梯度冲突可能影响模型的训练效果，导致某些架构的性能评估出现偏差。此外，共享权重的架构在性能上往往存在高方差，即在超网络中表现良好的架构，独立训练后未必能达到同样的性能。这是因为在超网络中的共享参数未经过特定架构的单独优化，而是服务于所有可能的架构组合。为了解决这些问题，研究者提出了多种改进策略。例如，通过调控不同架构之间的权重更新频率，减少梯度冲突的影响。此外，一些研究采用了逐层权重共享的方式[110]，使不同架构在部分层共享参数，而在关键层上保持独立，以提升评估的准确性。另一种改进方法是使用后训练阶段，即在搜索结束后，对筛选出的优质架构进行独立训练，确保其在实际应用中的表现稳定[111]。

一次性架构搜索的实际应用十分广泛。ENAS 作为该策略的代表性案例，在 ImageNet 数据集上的搜索速度较传统方法提升了数百倍。此后，许多 NAS 系统都采用了类似的权重共享机制，进一步优化了搜索过程。此外，在嵌入式系统和移动设备上，研究者通过一次性架构搜索找到计算效率高且占用资源少的网络[85]，使模型在实际应用中能快速部署。总的来说，一次性架构搜索评估是解决 NAS 计算成本问题的有效策略。通过在超网络中共享权重，这一方法大幅缩短了搜索时间，并使 NAS 在大规模任务中变得更加可行。未来，随着动态权重共享和混合策略的进一步发展，一次性架构搜索有望在更多复杂场景中展现其潜力，为 NAS 的广泛应用提供有力支持。

4. 代理模型评估

代理模型评估[112]通过构建性能预测模型，减少了神经架构搜索过程中对完全训练的依赖，使得搜索过程变得更加高效。与传统的完整训练评估不同，代理模型学习候选架构的特征与其表现之间的映射关系，在不进行完全训练的情况下快速预测每个架构的性能。代理模型评估在大规模搜索任务中表现尤为突出，尤其在需要高效筛选的场景，如计算机视觉和自然语言处理任务上，已经展现出重要价值。

代理模型的构建通常包括两个关键步骤：特征表示和性能预测。特征表示是将候选架构转换为适合代理模型输入的向量形式，这些特征可以包括层的数量、卷积核大小、连接方式以及激活函数等。例如，一个卷积神经网络的特征可能是“3×3 卷积＋最大池化＋全连接层”这样的结构序列。性能预测则是通过拟合这些特征与已知架构表现之间的关系，估计未评估架构的性能。常用的代理模型包括高斯过程、随机森林、贝叶斯神经网络和深度学习回归模型等[113]。高斯过程不仅能够预测性能，还能提供预测的不确定性，在 NAS 中被广泛应用。在贝叶斯优化框架下，系统会基于高斯过程的预测结果和不确定性值选择下一个评估对象。对于预测结果不确定性较大的架构，系统会优先进行完整训练，以更新代理模型并提高其预测精度。随机森林和贝叶斯神经网络也是常见的代理模型选择，这些模型能够处理更高维度的数据，适应复杂的搜索空间。深度学习模型作为代理模型则进一步提升了预测性能，特别是在搜索空间庞大、非线性关系复杂的场景中。

代理模型评估的主要优势在于其大幅减少了对计算资源的消耗。在传统的 NAS 流程中，完全训练每个候选架构往往需要数小时甚至数天的时间，而代理模型可以在几毫秒内就完成对架构性能的预测。这使得代理模型评估特别适用于大规模搜索任务和时间敏感的应用场景。此外，代理模型还可以与其他策略结合使用，如结合早停机制，在中期训练阶段预测模型最终的表现，以避免不必要的资源浪费。然而，代理模型评估的精度和稳健性受到多种因素的影响。首先，代理模型的预测质量依赖于训练数据的数量和多样性。如果训练集中的架构样本不足或分布不均匀，代理模型的预测结果可能偏差较大。其次，代理模型在处理十分复杂的搜索空间时，可能无法捕捉所有架构特征的细微差异，这会导致某些潜在优质架构被忽略。因此，在搜索过程中，系统通常需要不断更新代理模型，以融入最新评估的数据，提升模型的准确性。

为了提高代理模型的预测效果，研究者提出了多种优化策略。一种常见的方法是使用多模型集成，将高斯过程、随机森林和神经网络等模型的预测结果进行融合，以减小单一模型的误差[114]。此外，动态更新代理模型也是提高其性能的重要手段。在搜索过程中，每次新增架构的评估数据都会实时更新代理模型，使其始终反映当前搜索空间的最新信息。这些优化策略使代理模型评估在不同任务和数据集上展现出较好的泛化能力。在 NAS 的实践中，代理模型评估已经在多个领域取得成功。例如，在计算机视觉任务中，NAS-Bench-101[115] 和 NAS-Bench-201[116] 等基准数据集通过代理模型加速了架构搜索，并大幅减少了实验次数。在自然语言处理和语音识别任务中，代理模型帮助生成了高效的 RNN 和 Transformer 变体，并提升了模型的性能。在嵌入式系统和移动设备上，代理模型评估还能够快速预测不同架构的计算开销和能耗，帮助系统选择最优架构，实现模型的高效部署。总之，代理模型评估为 NAS 提供了强大的支持，它以高效、灵活的特点在不断扩展的搜索空间中发挥着不可替代的作用。

5. 早停机制

早停机制是一种在训练过程中根据模型的中期表现提前终止不佳架构训练的策略，旨在减少计算资源浪费并加快搜索过程。其核心思想是通过监控候选架构在部分训练轮次中的性能，预测其最终表现，并在判断模型潜力有限时及时停止训练。早停机制尤其适用于大规模 NAS 任务和资源受限的环境，因为它能够有效避免对表现不佳模型的冗余训练，提高搜索效率。

在实际应用中，早停机制通常基于一些中期指标进行决策，如验证集上的损失、准确率或梯度变化情况。当这些指标在若干轮训练后没有显著提升，或出现停滞甚至恶化时，系统会触发提前停止逻辑。这种策略的前提是假设模型在早期阶段的表现能部分反映其最终潜力。通过及时终止那些无法达到预期性能的架构训练，系统可以将更多资源集中在更具潜力的模型上，加快整个搜索流程。为了实现更准确的早停决策，一些 NAS 系统会设置阈值条件或采用动态监控策略。最常见的做法是在若干轮训练后计算模型的损失变化率或性能提升幅度，并与预设的阈值进行比较。如果模型的性能提升低于设定值或停滞在某一水平，则终止该模型的训练。另一种方法是使用滑动窗口技术，即在最近几轮训练中评估模型的趋势变化，避免因偶然的波动触发早停。此外，有些系统还会采用自适应的早停策略，根据当前搜索阶段的进展或系统资源的情况调整早停标准，使其更灵活地适应不同任务。

早停机制在 NAS 中发挥着重要作用，不仅能显著节省计算资源，还能大幅缩短整个搜索过程的时间。对于大规模数据集而言，完整训练一个复杂的模型可能需要数天甚至数周，而通过引入早停机制，可以将这一过程压缩至几个小时甚至几分钟。在搜索的初期阶段，NAS 系统通常会生成大量的候选架构，其中大部分架构在早期表现不佳。此时，早停机制能够迅速淘汰这些低质量架构，避免无谓的计算消耗，为后续阶段的优质架构腾出更多资源。特别是在结合代理模型的

NAS 系统中，早停机制的应用还能进一步优化性能预测。代理模型可以通过早停机制提供的反馈数据提升对架构性能的预测准确性。在资源受限的环境中，如移动设备或嵌入式系统，早停机制的作用尤为突出。由于这些设备的计算能力和电池寿命有限，如何高效地进行模型搜索和训练是关键。通过早停机制，系统能够快速筛选出有潜力的架构，及时停止低效的训练，从而在最短时间内找到最优解，确保选出的模型既高效又具有实际应用价值。此外，早停机制还可以与多阶段评估策略相结合，进一步提高搜索的精准度和效率。例如，第一阶段可以使用比较宽松的早停标准进行初步筛选；而在第二阶段，则可以通过更严格的标准对优质模型进行深入评估。总体而言，早停机制为 NAS 中的高效评估提供了一种实用且灵活的解决方案。它通过快速淘汰不佳模型和集中资源训练优质架构，大幅提升了搜索过程的速度和资源利用率。在实际应用中，早停机制已成为多种 NAS 系统中的标准配置，为大规模架构搜索任务提供了关键支持。

6. 资源感知评估

资源感知评估是一种在 NAS 中同时关注模型性能和计算资源消耗的评估策略。传统的 NAS 方法通常只以性能指标（如准确率、F1 分数等）作为优化目标，而在实际应用中，尤其是在移动设备、嵌入式系统或云端部署环境中，计算资源和运行效率同样至关重要。资源感知评估通过在搜索阶段引入资源约束，确保选出的模型不仅具有优异的性能，还能在指定的资源限制内高效运行。这种策略已广泛应用于需要兼顾性能和效率的场景，如自动驾驶、物联网和智能手机中的实时应用。

资源感知评估的核心是在 NAS 过程中将资源消耗指标纳入优化目标。这些指标可以包括推理时间、内存占用、计算延迟、能耗以及模型参数的数量。在实际应用中，系统可能根据不同场景的需求对这些资源指标进行加权，以实现多目标优化。例如，在移动设备上，能耗和延迟是关键的指标，而在云端环境中，系统可能更关注内存使用和计算效率。在资源感知 NAS 中，评估模型时常采用多目标优化的方法。通过构建多目标函数，系统能够在不同目标之间找到最优的平衡点。具体实现方式包括使用加权和法或帕累托前沿法。在加权和法中，系统为每个目标分配一个权重，将多个目标组合成一个标量进行优化；而在帕累托优化中，系统寻找在所有目标上均衡表现的模型，形成帕累托前沿，用于指导后续的模型选择。

为了高效进行资源感知评估，研究者还提出了一些近似计算方法[117]。在 NAS 的初期阶段，系统可以采用代理模型或低保真评估方法快速估计模型的资源消耗。例如，通过建立回归模型预测每个架构的推理时间和内存使用，避免对每个候选模型进行实际部署和测试。代理模型在资源感知评估中的应用大幅降低了计算成本，使得系统能在短时间内评估大量架构。此外，资源感知评估的实现还需要结合特定硬件平台的特性。在嵌入式系统或移动设备上，不同的硬件平台对模型的运行速度和能耗表现有显著影响。例如，同一卷积网络在 GPU、CPU 或 NPU 上的推理速度可能存在较大差异。因此，系统在搜索阶段通常会模拟目标平台的运行环境，确保评估结果能反映模型在实际部署中的表现。这种硬件感知的评估方法能够帮助系统筛选出与目标硬件最兼容的模型。

资源感知评估的策略也与 NAS 搜索算法紧密结合。例如，在进化算法中，种群中的个体不仅需要根据性能指标选择，还需要考虑资源消耗的约束；在强化学习 NAS 中，控制器会在生成新架构时将资源指标作为奖励函数的一部分，指导策略优化。在一次性架构搜索（如 ENAS）中，系统可以通过共享参数的超网络快速估计不同架构的资源开销，并在搜索过程中不断更新这些估计结果。在实践中，资源感知评估的效果已经在多种应用中得到了验证。另一个典型案例是 NAS 在自动驾驶系统中的应用，研究者在模型设计时同时考虑了精度和推理延迟，以确保车辆的检测系

统能够实时响应。在这些应用中，资源感知评估不仅优化了模型的性能，还使模型在实际场景中更具实用性。

总之，资源感知评估通过将计算资源与性能目标结合，为 NAS 提供了一种更全面的优化方案。它特别适用于需要实时响应或运行在资源受限环境中的任务，如物联网设备、智能家居和移动应用。随着硬件技术的发展和 NAS 算法的进步，资源感知评估将在未来的深度学习模型设计中扮演越来越重要的角色。

7. 真实部署评估

真实部署评估是一种在 NAS 流程结束后，将筛选出的模型部署于实际环境中，以测试其在真实运行条件下表现的评估策略。这一过程超越了单纯的性能指标评估，重点考查模型在实际硬件、应用场景和用户交互中的综合表现，包括推理速度、响应延迟、能耗、存储开销以及稳定性等。真实部署评估是确保 NAS 选出的模型不仅在离线测试中表现良好，而且能在复杂的实际环境中可靠运行的关键步骤。

在 NAS 任务中，离线训练和验证阶段往往采用标准数据集与测试环境，但这些条件未必能完全反映模型的最终应用场景。例如，在智能手机、自动驾驶汽车、物联网设备等实际应用中，硬件性能、计算能力、网络延迟、功耗限制和存储空间都会对模型运行产生显著影响。真实部署评估通过模拟这些复杂的运行条件，全面评估模型的性能和适应性，确保模型能满足应用的实际需求。真实部署评估通常涵盖以下几方面。首先是推理速度和响应延迟，这是所有实时系统的关键指标；特别是在自动驾驶、监控系统和金融交易等任务中，系统需要在毫秒级时间内作出响应；通过在实际硬件上运行模型，评估过程能够准确测量模型的推理时间和延迟，确保其符合应用场景的响应要求。其次是能耗评估，在移动设备和物联网设备上，电池寿命至关重要，因此，真实部署评估还会记录模型在目标设备上运行时的功耗，以保证其能在电池供电环境下长时间运行。存储开销是另一个重要指标，尤其是在嵌入式系统和边缘设备上，模型的大小直接影响其部署可行性，真实部署评估通过测量模型的参数数量和内存占用情况，帮助系统设计者在模型精度与存储需求之间找到最佳平衡。此外，在某些任务中，如自然语言处理或语音识别，模型还需要处理连续的输入数据流。此时，评估过程会重点考查模型的稳定性和稳健性，确保其在长时间运行中不会出现性能下降或错误累积。因此，在实际应用中，真实部署评估的具体实施方式会根据任务和硬件平台的不同而有所差异。在云计算环境中，系统通常会评估模型的多线程性能、负载均衡能力和容错性，以确保模型能在大规模数据处理任务中稳定运行。在移动设备上，评估则侧重于测量模型在不同使用场景中的表现，如手机待机模式与高负载应用中的能耗对比。在自动驾驶系统中，模型的部署需要经过严格的实时性测试，以确保其在各种复杂路况下都能作出快速、准确的判断。

为了实现更高效的评估，一些系统还采用了在线 A/B 测试的方式，即将新模型与现有模型同时部署在真实环境中，比较其在同一任务上的表现[118]。这种方式不仅能够直接反映新模型的改进效果，还能通过收集用户反馈优化模型设计。此外，研究者还提出了逐步部署的策略，在小规模环境中先行测试，再逐步扩展到更大的应用范围，以降低部署风险。真实部署评估已在多个领域取得了显著成效。在自动驾驶领域，通过真实部署评估，研究者确保了模型在各种天气、光照和交通条件下的稳定性。在金融系统中，交易模型的部署经过了严格的延迟和可靠性测试，以满足高频交易的苛刻要求。这些案例表明，真实部署评估是将 NAS 研究成果转化为实际应用的重要环节。

总之，真实部署评估在确保模型适应性和运行可靠性方面发挥着关键作用。它弥补了离线评估的不足，使 NAS 系统能够找到不仅在理论上优异，而且在实际应用中高效的模型。通过这一过

程，系统设计者能够更好地应对复杂多变的运行环境，确保模型在真实世界中的稳定表现。随着NAS技术和硬件设备的不断发展，真实部署评估将继续为模型的优化与应用提供坚实的保障。

8. 评估策略的未来发展方向

评估策略的未来发展方向将进一步聚焦于提升效率、准确性和适应性，以应对日益复杂的NAS任务和多样化的应用场景。随着NAS应用的扩大，传统评估策略的瓶颈逐渐显现，如计算资源的限制、评估结果的偏差以及无法及时响应实际部署需求等。未来的研究将探索更加智能、灵活的评估方法，通过动态调整、跨任务迁移和资源优化，提升NAS的整体性能。其中，一个重要的发展方向是自适应评估策略，传统评估策略往往在搜索的每轮中采用固定的评估方式，而自适应评估根据当前的搜索进展动态调整评估方案。例如，在初期阶段使用低保真评估快速筛选潜力架构，随着搜索的深入逐步引入高保真评估进行更精细的验证。此外，自适应策略还能根据当前的计算资源和时间限制，实时调整评估的复杂度。这样的机制能够显著提升NAS系统的效率，使其更灵活地应对不同规模和场景的任务。另一个值得关注的方向是跨任务评估与迁移学习。NAS系统通常需要针对每个新任务重新进行搜索和评估，这一过程既耗时又费力。未来的研究应关注如何在不同任务之间迁移已有的评估结果和搜索经验。例如，通过构建任务之间的性能映射模型，将在某一任务上获得的架构性能预测迁移到其他相关任务中。这种跨任务评估策略能够减少重复计算，并提升搜索的泛化能力，特别是在多任务学习和联邦学习中具有广阔的应用前景。分布式评估也是未来的研究重点之一。随着NAS任务规模的不断扩大，单节点计算难以满足复杂搜索的需求。通过将评估任务分配到多个计算节点并行进行，分布式评估能够大幅缩短搜索时间，并更好地利用计算资源。此外，在分布式环境中，如何协调多个节点之间的信息交互与同步，避免重复计算和数据冲突，也是需要解决的重要问题。未来的分布式NAS系统有望结合云计算和边缘计算技术，实现更高效、更稳定的评估体系。最后，NAS系统与用户反馈的结合也是一个值得探索的发展方向。在某些应用中，如推荐系统或个性化服务，用户反馈能够提供额外的性能指标。未来的评估策略可以将用户反馈纳入NAS优化的闭环中，使系统能够根据实际使用情况不断调整和优化模型，从而提升用户体验。

综上，未来的NAS评估策略将向智能化、动态化和分布式方向发展，并更加注重资源感知与跨任务的灵活应用。这些发展将推动NAS系统在更加复杂和多样化的应用中发挥更大的价值，为智能系统的设计和优化提供有力支持。

小结与展望

本章围绕自动机器学习展开讨论，重点讨论了算法选择、自动参数优化这两部分自动机器学习算法的重要流程；并进一步介绍了元学习与神经架构搜索两类重要的自动机器学习方法。其中，元学习旨在介绍如何让机器自己学会学习，如通过评价机器学习模型在已有数据中的经验来为新数据设计新的学习方法；神经架构搜索旨在应对深度学习模型构件中网络层级设计空间大且神经网络模型训练评估耗时等挑战。自动机器学习不仅能够自动设计高效的模型架构，还能优化训练过程、加速推理并降低资源消耗，具有重要的研究意义与应用价值。随着大模型、多模态学习等技术的发展，自动机器学习的应用前景与空间将会更加广泛。未来，自动机器学习研究的重要方向可能包括如下几方面。

1. 与基础模型的深度融合

随着大规模预训练模型（如GPT[119]、BERT[120]等）的广泛应用，如何高效地训练、微调和部

署这些模型成为关键挑战。未来的自动机器学习研究将更加注重与基础模型的深度融合[121]，通过自动化手段优化其全生命周期。例如，利用元学习技术自动选择适合特定下游任务的微调策略（如适配器微调或提示学习）[122,123]，或通过神经架构搜索（NAS）优化模型结构以适应不同计算资源限制[121]。此外，自动化超参数优化技术可以显著减少大模型训练和推理中的调参成本。未来，这一方向的研究将进一步推动大模型的高效应用与普及。

2. 通用化与可扩展性研究

当前的自动机器学习方法通常针对特定任务或领域设计，缺乏通用性和可扩展性。未来的研究将致力于开发能够适应多种任务、模态和领域的通用自动机器学习框架。例如，通过元学习技术提取跨任务的共性知识，构建通用的超参数优化策略或架构搜索空间。此外，结合多模态学习技术，自动机器学习可以自动优化跨模态（如图像、文本、语音）任务的模型架构与训练策略[124,125]。通用化的自动机器学习框架不仅能够降低开发成本，还能探索其在复杂任务中的潜力，进一步推动 AI 在更广泛场景中的应用。

3. 绿色 AI 与可持续发展

随着深度学习模型参数规模的不断扩大和训练数据量的急剧增长，计算资源消耗大幅提升，导致高能耗问题日益突出。这一挑战促使机器学习领域深入探讨绿色 AI（Green AI），推动针对能效优化、计算资源节约和环境可持续性的研究。未来的自动机器学习研究将更加注重可持续性，通过优化算法减少模型训练和推理中的能耗与碳排放。例如，开发能耗感知的 NAS 方法，在搜索过程中引入能耗约束，自动设计高效且低功耗的模型架构[126]。此外，自动化模型压缩技术（如剪枝、量化和知识蒸馏）可以显著降低大模型的存储与计算需求，从而减少资源消耗[121]。

4. 人机协作与可解释性

随着自动机器学习的普及，如何使其与人类专家协作并增强模型的可解释性成为重要研究方向[123]。未来的自动机器学习系统将更加注重人机协作，通过结合领域知识和自动化能力，实现更高效的模型开发与优化。例如，开发交互式算法工具，允许用户参与算法选择与搜索过程并提供反馈，从而加速优化并提高结果的可信度。此外，自动化可解释性分析技术（如注意力可视化、特征重要性分析）可以帮助用户理解模型的决策过程，从而提高模型的可信度和可接受性。未来，该研究方向有望进一步推动自动机器学习应用与落地，并在对算法结果可解释要求高的应用场景如金融、医疗等任务中应用。

参考文献

[1] YUEM S Y，CHOW C K，ZHANG X，et al. Which algorithm should I choose：An evolutionary algorithm portfolio approach[J]. Applied Soft Computing，2016，40：654-673.

[2] LECUN Y，BOSER B，DENKER J S，et al. Backpropagation applied to handwritten zip code recognition[J]. Neural Computation，1989，1(4)：541-551.

[3] ZAREMBA W，SUTSKEVER I，VINYALS O. Recurrent Neural network regularization[EB/OL]. (2025-02-19)[2025-02-21]. https://arxiv.org/abs/1409.2329.

[4] 周志华. 机器学习[M]. 北京：清华大学出版社，2016.

[5] 李航. 机器学习方法[M]. 北京：清华大学出版社，2022.

[6] HOCHRETIER S，SCHMIDHUBER J. Long short-term memory[J]. Neural Computation，1997，9(8)：1735-1780.

[7] CHO K，VAN M B，GULCEHRE Ç，et al. Learning phrase representations using RNN encoder-decoder for

statistical machine translation[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. Doha, Qatar: Association for Computational Linguistics(ACL), 2014: 1724-1734.

[8] 程艳,尧磊波,张光河,等. 基于注意力机制的多通道 CNN 和 BiGRU 的文本情感倾向性分析[J]. 计算机研究与发展,2020,57(12): 2583-2595.

[9] DASH M, LIU H. Feature selection for classification[J]. Intelligent Data Analysis, 1997, 1(1-4): 131-156.

[10] DY J G, BRODLEY C E. Feature selection for unsupervised learning[J]. Journal of machine learning research, 2004, 5(8): 845-889.

[11] DENG J, DONG W, SOCHER R, et al. ImageNet: A large-scale hierarchical image database[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2009: 248-255.

[12] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2016: 770-778.

[13] SZEGEDY C, LIU W, JIA Y Q, et al. Going deeper with convolutions[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2015: 1-9.

[14] 张旭东. 机器学习导论[M]. 北京: 清华大学出版社, 2022.

[15] GOODFELLOW I, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial nets[C]//Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2. Cambridge: MIT Press, 2014: 2672-2680.

[16] 何弦,李佳宸,金立,等. 三维模板跟踪的基准合成数据集构建及算法评估[J]. 计算机学报, 2022, 45(3): 585-600.

[17] RIBEIRO M T, SINGH S, GUESTRIN C. "Why should I trust you?" Explaining the predictions of any classifier[C]//Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2016: 1135-1144.

[18] 李航. 统计学习方法[M]. 北京: 清华大学出版社, 2019.

[19] FISHER R A. The use of multiple measurements in taxonomic problems[J]. Annals of Eugenics, 1936, 7(2): 179-188.

[20] LUNDBERG S M, LEE, S I, A unified approach to interpreting model predictions[C]//Proceedings of the International Conference on Neural Information Processing Systems. New York: Curran Associates Inc., 2017: 4768-4777.

[21] SALIMANS T, GOODFELLOW I, ZAREMBA W, et al. Improved techniques for training GANs[C]// Proceedings of the International Conference on Neural Information Processing Systems. New York: Curran Associates Inc., 2016: 2234-2242.

[22] HEUSEL M, RAMSAUER H, UNTERTHINER T, et al. GANs trained by a two time-scale update rule converge to a local nash equilibrium[C]//Proceedings of the International Conference on Neural Information Processing Systems. New York: Curran Associates Inc., 2017: 6629-6640.

[23] BERGSTRA J, BARDENET R, BENGIO Y, et al. Algorithms for hyper-parameter optimization[C]// Proceedings of the International Conference on Neural Information Processing Systems. New York: Curran Associates Inc., 2011: 2546-2554.

[24] YU T, ZHU H. Hyper-parameter optimization: A review of algorithms and applications[EB/OL]. (2020-03-12) [2025-02-21]. https://arxiv.org/abs/2003.05689.

[25] HSU C W, CHANG C C, LIN C J. A practical guide to support vector classification[R]. Department of Computer Science, NTu, 2003.

[26] HESTERMAN J Y, CAUCCI L, KUPINSKI M A, et al. Maximum-likelihood estimation with a contracting-grid search algorithm[J]. IEEE Transactions on Nuclear Science, 2010, 57(3): 1077-1084.

[27] BERGSTRA J, BENGIO Y. Random search for hyper-parameter optimization[J]. Journal of Machine Learning Research, 2012, 13(1): 281-305.

[28] OZAKI Y, TANIGAKI Y, WATANABE S, et al. Multiobjective tree-structured parzen estimator[J]. Journal of Artificial Intelligence Research, 2022, 73: 1209-1250.
[29] FALKNER S, KLEIN A, HUTTER F. BOHB: Robust and efficient hyperparameter optimization at scale[C]// Proceedings of International Conference on Machine Learning. New York: PMLR, 2018: 1437-1446.
[30] WATANABE S. Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance[EB/OL]. (2023-04-21)[2025-02-21]. https://arxiv.org/abs/230 4.11127.
[31] STANLEY K O, MIIKKULAINEN R. Evolving neural networks through augmenting topologies[J]. Evolutionary Computation, 2002, 10(2): 99-127.
[32] REAL E, MOORE S, SELLE A, et al. Large-scale evolution of image classifiers[C]//Proceedings of International Conference on Machine Learning. New York: PMLR, 2017: 2902-2911.
[33] REAL E, AGGARWAL A, HUANG Y, et al. Regularized evolution for image classifier architecture search[C]// Proceedings of the 33rd AAAI Conference on Artificial Intelligence. Menlo Park: AAAI Press, 2019: 4780-4789.
[34] ELSKEN T, METZEN J H, HUTTER F. Efficient multi-objective neural architecture search via lamarckian evolution[C]//Proceedings of International Conference on Learning Representations. Puerto Rico: Openreview. net, 2018: 1-23.
[35] MIIKKULAINEN R, LIANG J, MEYERSON E, et al. Evolving deep neural networks[M]//Artificial Intelligence in the Age of Neural Networks and Brain Computing. New York: Academic Press, 2024: 269-287.
[36] SHAHRIARI B, SWERSKY K, WANG Z, et al. Taking the human out of the loop: A review of bayesian optimization[J]. Proceedings of the IEEE, 2015, 104(1): 148-175.
[37] VICTORIA A H, MARAGATHAM G. Automatic tuning of hyperparameters using Bayesian optimization[J]. Evolving Systems, 2021, 12(1): 217-223.
[38] WANG X, JIN Y, SCHMITT S, et al. Recent advances in Bayesian optimization[J]. ACM Computing Surveys, 2023, 55(13s): 1-36.
[39] JIN H, SONG Q, HU X. Auto-Keras: An efficient neural architecture search system[C]//Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: ACM Press, 2019: 1946-1956.
[40] BINOIS M, WYCOFF N. A survey on high-dimensional Gaussian process modeling with application to Bayesian optimization[J]. ACM Transactions on Evolutionary Learning and Optimization, 2022, 2(2): 1-26.
[41] 崔佳旭, 杨博. 贝叶斯优化方法和应用综述[J]. 软件学报, 2018, 29(10): 3068-3090.
[42] WU J, POLOCZEK M, WILSON A G, et al. Bayesian optimization with gradients[C]//Proceedings of the International Conference on Neural Information Processing Systems. New York: Curran Associates, 2017: 1-12.
[43] MAHSERECI M, BALLES L, LASSNER C, et al. Early stopping without a validation set[EB/OL]. (2017-06-06)[2025-02-21]. https://arxiv.org/abs/1703.09580.
[44] DENG B, YAN J, LIN D. Peephole: Predicting network performance before training[EB/OL]. (2017-12-09)[2025-02-21]. https://arxiv.org/abs/1712.03351.
[45] DOMHAN T, SPRINGENBERG J T, HUTTER F. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves[C]//Proceedings of the International Conference on Artificial Intelligence. Menlo Park: AAAI Press, 2015: 3460-3468.
[46] SCHMUCKER R, DONINI M, ZAFAR M B, et al. Multi-objective asynchronous successive halving[EB/OL]. (2021-06-23)[2025-02-1]. https://arxiv.org/abs/2106.12639.
[47] LI L, JAMIESON K, DESALVO G, et al. Hyperband: A novel bandit-based approach to hyperparameter optimization[J]. Journal of Machine Learning Research, 2018, 18(185): 1-52.
[48] AWAD N, MALLIK N, HUTTER F. DEHB: Evolutionary hyperband for scalable, robust and efficient hyperparameter optimization[EB/OL]. (2021-05-20)[2025-02-21]. https://arxiv.org/abs/2105.09821.
[49] WHITE C, NEISWANGER W, SAVANI Y. BANANAS: Bayesian optimization with neural architectures for

neural architecture search[C]//Proceedings of the 35th AAAI Conference on Artificial Intelligence. Menlo Park: AAAI Press,2021: 10293-10301.

[50] VILALTA R,DRISSI Y. A perspective view and survey of meta-learning[J]. Artificial Intelligence Review, 2002,18: 77-95.

[51] HOSPEDALES T,ANTONIOU A,MICAELLI P,et al. Meta-learning in neural networks: A survey[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2021,44(9): 5149-5169.

[52] LI D Z,LI R Q,WANG L J,et al. You only infer once: Cross-modal meta-transfer for referring video object segmentation[C]//Proceedings of the 36th AAAI Conference on Artificial Intelligence. Menlo Park: AAAI Press,2022: 1297-1305.

[53] SINGH R,BHARTI V,PUROHIT V,et al. MetaMed: Few-shot medical image classification using gradient-based meta-learning[J]. Pattern Recognition,2021,120: 108111.

[54] CHEN J,TANG J,LI W. Industrial edge intelligence: Federated-meta learning framework for few-shot fault diagnosis[J]. IEEE Transactions on Network Science and Engineering,2023,10(6): 3561-3573.

[55] ZENG M,SUN W,WANG Z,et al. Evaluation and application algorithm of artificial intelligence unmanned vehicle control device based on IoT intelligent transportation[J]. Computing and Informatics,2024,43(4): 944-973.

[56] LUPU E S,XIE F,PREISS J A,et al. Magic VFM-meta-learning adaptation for ground interaction control with visual foundation models[J]. IEEE Transactions on Robotics,2024,41(10): 180-199.

[57] XU J,SONG J,SANG Y,et al. CDAML: A cluster-based domain adaptive meta-learning model for cross domain recommendation[J]. World Wide Web,2023,26(3): 989-1003.

[58] TAN Z P,ZHANG H Y,TIAN F,et al. Progressively coupling network for brain MRI registration in few-shot situation[C]//Processing of the Medical Image Computing and Computer Assisted Intervention. Berlin: Springer,2023: 623-633.

[59] VETTORUZZO A,BOUGUELIA M R,VANSCHOREN J,et al. Advances and challenges in meta-learning: A technical review[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, 46: 4763-4779.

[60] HUISMAN M, VAN RIJN J N, PLAAT A. A survey of deep meta-learning[J]. Artificial Intelligence Review,2021,54(6): 4483-4541.

[61] FINN C,ABBEEL P,LEVINE S. Model-agnostic meta-learning for fast adaptation of deep networks[C]// Proceedings of the International Conference on Machine Learning. New York: PMLR,2017: 1126-1135.

[62] ANTONIOU A, EDWARDS H, STORKEY A. How to train your MAML[C]//Proceedings of the International Conference on Learning Representations. San Juan,Puerto Rico: Openreview. net,2018: 1-11.

[63] AMID E, ANIL R, FIFTY C, et al. Step-size adaptation using exponentiated gradient updates[EB/OL]. (2022-01-31)[2025-02-21]. https://arxiv. org/abs/2202. 00145.

[64] KINGMA D,BA J. Adam: A method for stochastic optimization[EB/OL]. (2014-12-22)[2025-03-26]. https://arxiv. org/abs/1412. 6980.

[65] TIELEMAN T. Lecture 6. 5-RMSProp: Divide the gradient by a running average of its recent magnitude[Z]. COURSERA: Neural Networks for Machine Learning 4. 2(2012): 26.

[66] SANTORO A, BARTUNOV S, BOTVINICK M, et al. Meta-learning with memory-augmented neural networks[C]//Proceedings of the International Conference on Machine Learning. New York: PMLR,2016: 1842-1850.

[67] BERTINETTO L,HENRIQUES J F,TORR P,et al. Meta-learning with differentiable closed-form solvers[C]// Proceedings of the International Conference on Learning Representations. Puerto Rico: Openreview. net, 2018: 1-15.

[68] MUNKHDALAI T,YU H. Meta networks[C]//Proceedings of the International Conference on Machine Learning. New York: PMLR,2017: 2554-2563.

[69] MISHRA N,ROHANINEJAD M,CHEN X,et al. A simple neural attentive meta-learner[C]//Proceedings of the International Conference on Learning Representations. Puerto Rico: Openreview. net,2018: 1-17.

[70] 赵一凡,李甲,田永鸿. 局部关系泛化表征的小样本增量学习[J]. 中国科学: 信息科学,2023,53(6): 1132-1146.

[71] 葛轶洲,刘恒,王言,等. 小样本困境下的深度学习图像识别综述[J]. 软件学报,2022,33(1): 193-210.

[72] PIAO Y R, LU C Y, ZHANG M, LU H C. Semi-supervised video salient object detection based on uncertainty-guided pseudo labels [C]//Proceedings of the 36th International Conference on Neural Information Processing Systems. New York: Curran Associates,2022: 5614-5627.

[73] RAVI S,LAROCHELLE H. Optimization as a model for few-shot learning[C]//Proceedings of the 5th International Conference on Learning Representations. Puerto Rico: Openreview. net,2017: 1-11.

[74] DAI K N,ZHANG Y H,WANG D,et al. High-performance long-term tracking with meta-updater[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press,2020: 6297-6306.

[75] 郝肇铁,郭斌,赵凯星,等. 从规则驱动到群智涌现: 多机器人空地协同研究综述[J]. 自动化学报,2024,50(10): 1877-1905.

[76] SUNG F,YANG Y,ZHANG L,et al. Learning to compare: Relation network for few-shot learning[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press,2018: 1199-1208.

[77] SNELL J,SWERSKY K,ZEMEL R. Prototypical networks for few-shot learning[C]//Proceedings of the International Conference on Neural Information Processing Systems. New York: Curran Associates,2017: 4080-4090.

[78] VINYALS O, BLUNDELL C, LILLICRAP T, et al. Matching networks for one shot learning [C]//Proceedings of the International Conference on Neural Information Processing Systems. New York: Curran Associates,2016: 3637-3645.

[79] JIANG X,LI G,ZHANG X P,HE Y. A semisupervised Siamese network for efficient change detection in heterogeneous remote sensing images[J]. IEEE Transactions on Geoscience and Remote Sensing,2021,60: 1-18.

[80] ELSKEN T,METZEN J H,HUTTER F. Neural architecture search: A survey[J]. Journal of Machine Learning Research,2019,20(55): 1-21.

[81] ZOPH B,LE Q. Neural architecture search with reinforcement learning[EB/OL]. (2017-02-15)[2025-02-21]. https://arxiv. org/abs/1611. 01578.

[82] PHAM H,GUAN M,ZOPH B,et al. Efficient neural architecture search via parameters sharing[C]//Proceedings of the 35th International Conference on Machine Learning. New York: PMLR,2018: 4095-4104.

[83] ZOPH B, VASUDEVAN V, SHLENS J, et al. Learning transferable architectures for scalable image recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press,2018: 8697-8710.

[84] ZHONG Z, YAN J, WU W, et al. Practical block-wise neural network architecture generation [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press,2018: 2423-2432.

[85] YAN B,PENG H W,WU K,et al. LightTrack: Finding lightweight neural networks for object tracking via one-shot architecture search[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press,2021: 15175-15184.

[86] LV Z,QIAN C,YEN G G,et al. Analyzing the expected hitting time of evolutionary computation-based neural architecture search algorithms [J]. IEEE Transactions on Emerging Topics in Computational Intelligence,2024,8(6): 3899-3911.

[87] REAL E,AGGARWAL A,HUANG Y,et al. Regularized evolution for image classifier architecture search[C]//Proceedings of the 33rd AAAI Conference on Artificial Intelligence. Menlo Park: AAAI Press,2019:

4780-4789.

[88] LIU H,SIMONYAN K,YANG Y. DARTS: Differentiable architecture search[C]//Proceedings of the 6th International Conference on Learning Representations. Puerto Rico: Openreview. net,2018: 1-13.

[89] KUKAKA J,GOLKOV V,CREMERS D. Regularization for deep learning: A taxonomy[EB/OL]. (2017-10-29)[2025-02-21]. https://arxiv. org/abs/1710. 10686.

[90] CHEN X, XIE L X, WU J, et al. Progressive differentiable architecture search: Bridging the depth gap between search and evaluation[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway: IEEE Press,2019: 1294-1303.

[91] ZHANG M,LIU T W,PIAO Y R,et al. Auto-MSFNet: Search multi-scale fusion network for salient object detection[C]//Proceedings of the 29th ACM International Conference on Multimedia. New York: ACM Press,2021: 667-676.

[92] LIU C, ZOPH B, NENMANN M, et al. Progressive neural architecture search[C]//Proceedings of the European Conference on Computer Vision. Berlin: Springer,2018: 19-35.

[93] LUO R Q,TIAN F,QIN T,et al. Neural architecture optimization[C]//Proceedings of the 32nd International Conference on Neural Information Processing Systems. New York: Association for Computing Machinery, 2018: 7827-7838.

[94] WU X,ZHANG Y T,LAI K W,et al. A novel centralized federated deep fuzzy neural network with multi-objectives neural architecture search for epistatic detection[J]. IEEE Transactions on Fuzzy Systems,2025, 33(1): 94-107.

[95] LIU Y,GUO S,ZHANG J,et al. Collaborative neural architecture search for personalized federated learning[J]. IEEE Transactions on Computers,2025,74(1): 250-262.

[96] WU R,LI C,ZOU J,et al. Generalizable reconstruction for accelerating MR imaging via federated learning with neural architecture search[J]. IEEE Transactions on Medical Imaging,2025,44(1): 106-117.

[97] WU X,WANG D,CHEN H H,et al. Neural architecture search for text classification with limited computing resources using efficient cartesian genetic programming[J]. IEEE Transactions on Evolutionary Computation,2023,28(3): 638-652.

[98] WU Y, TANG B, DENG L, et al. Hardware-resource-constrained neural architecture search for edge-side fault diagnosis of wind-turbine gearboxes[J]. IEEE Transactions on Industrial Electronics, 2023, 71(8): 9812-9822.

[99] PENG C,LI Y Y,SHANG R H,et al. Recnas: Resource-constrained neural architecture search based on differentiable annealing and dynamic pruning[J]. IEEE Transactions on Neural Networks and Learning Systems,2022,35(2): 2805-2819.

[100] KRIZHEVSKY A,HINTON G. Learning multiple layers of features from tiny images[R]. Techaical Report TR-2009,University of Toronto,2009.

[101] XIAO H,RASUL K,VOLLGRAF R. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms[EB/OL]. (2017-08-25)[2025-02-21]. https://arxiv. org /abs/1708. 07747.

[102] FANG J M,CHEN Y K,ZHANG X B,et al. EAT-NAS: Elastic architecture transfer for accelerating large-scale neural architecture search[J]. Science China Information Sciences,2021,64: 192106.

[103] DAI X Y,CHEN D D,LIU M C,et al. DA-NAS: Data adapted pruning for efficient neural architecture search[EB/OL]. (2020-03-27)[2025-02-21]. https://arxiv. org/abs/2003. 12563.

[104] RUNGE F, STOLL D, FALKNER S, et al. Learning to design RNA[C]//Proceedings of International Conference on Learning Representations. Puerto Rico: Openreview. net,2019: 1-29.

[105] KLEIN A, FALKNER S, BARTELS S, et al. Fast Bayesian optimization of machine learning hyperparameters on large datasets[C]//Proceeds of the 20th International Conference on Artificial Intelligence and Statistics. New York: PMLR,2017: 528-536.

[106] NA B,MOK J,CHOE H,et al. Accelerating neural architecture search via proxy data[C]//Proceedings of

the International Joint Conference on Artificial Intelligence. Montreal: IJCAI,2021: 2848-2854.
[107] YING G H,HE X,GAO B,et al. EAGAN: Efficient two-stage evolutionary architecture search for GANs[C]// Proceedings of the European Conference on Computer Vision. Berlin: Springer,2022: 37-53.
[108] 蒋鹏程,薛羽. 基于排序得分预测的演化神经架构搜索方法[J]. 计算机学报,2024,47(11): 2522-2535.
[109] BROCK A, LIM T, RITCHIE J M, et al. SMASH: One-shot model architecture search through hypernetworks[C]//Proceedings of the International Conference on Learning Representations. Puerto Rico: Openreview. net,2018: 1-22.
[110] ZHANG Y G,LIN Z J,JIANG J Y,et al. Deeper insights into weight sharing in neural architecture search [EB/OL]. (2020-06-06)[2025-02-21]. https://arxiv. org/abs/2001. 01431.
[111] NIU S C,WU J X,ZHANG Y F,et al. Disturbance-immune weight sharing for neural architecture search[J]. Neural Networks,2021,144(C): 553-564.
[112] MENDOZA H, KLEIN A, FEURER M, et al. Towards automatically-tuned neural networks [C]// Workshop on Automatic Machine Learning. New York: PMLR,2016: 58-65.
[113] NEGRINHO R,GORDON G. Deeparchitect: Automatically designing and training deep architectures[EB/OL]. (2017-04-28)[2025-02-21]. https://arxiv. org/abs/1704. 08792.
[114] 乔少杰,薛骐,杨国平,等. 基于动态自适应时空图的多元时序预测模型[J]. 计算机学报,2024,47(12): 2925-2937.
[115] YING C,KLEIN A,REAL E,et al. NAS-Bench-101: Towards reproducible neural architecture search[C]// Proceedings of the International Conference on Machine Learning. New York: PMLR,2019: 7105-7114.
[116] DONG X Y, YANG Y. NAS-Bench-201: Extending the scope of reproducible neural architecture search [EB/OL]. (2020-06-02)[2025-02-21]. https://arxiv. org/abs/2001. 00326.
[117] CAI H,ZHU L G,HAN S. ProxylessNAS: Direct neural architecture search on target task and hardware [EB/OL]. (2018-12-02)[2025-02-21]. https://arxiv. org/abs/1812. 00332.
[118] 蔡超丽,李纯纯,黄琳,等. ED-NAS: 基于神经网络架构搜索的陶瓷晶粒 SEM 图像分割方法[J]. 电子学报,2022,50(2): 461-469.
[119] RADFORD A,NARASIMHAN K. Improving language understanding by generative pre-training[EB/OL]. (2018-06-11)[2025-03-28]. https://www. mikecaptain. com/resources/pdf/GPT-1. pdf.
[120] DEVLIN J, CHANG M W, LEE K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding [C]//Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics. Cambridge: Association for Computational Linguistics, 2019: 4171-4186.
[121] SUKTHANKER R S,STAFFLER B,HUTTER F,et al. LLM compression with neural architecture search [EB/OL]. (2024-10-09)[2025-02-21]. https://arxiv. org/abs/2410. 06479v1.
[122] FANG Y,WU Y,YU X,et al. Few-shot learning on graphs: From meta-learning to LLM-empowered pre-training and beyond companion[C]//Proceedings of the ACM on Web Conference 2025. New York: ACM Press,2025: 9-12.
[123] RAJAPAKSHA P,CRESPI N. Explainable attention pruning: A meta-learning-based approach[J]. IEEE Transactions on Artificial Intelligence,2024,5(6): 2505-2516.
[124] MA H,YANG K. MetaSTNet: Multimodal meta-learning for cellular traffic conformal prediction[J]. IEEE Transactions on Network Science and Engineering,2023,11(2): 1999-2011.
[125] TU G,WU T,LUO X,et al. Meta-learning for incomplete multimodal sentiment analysis[C]//Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press,2025: 2911-2915.
[126] DOU Z W,DONG Y. Multi-objective neural architecture search for efficient and fast semantic segmentation on edge[J]. IEEE Transactions on Intelligent Vehicles,2023,9(1): 1346-1357.