



中国科学院大学
University of Chinese Academy of Sciences

模式识别与机器学习大作业报告

CIFAR-10 图片分类

小组成员 1:	邓璇	2023E8016082026	计算机技术
小组成员 2:	郑子豪	2023E8016082043	计算机技术

2024 年 6 月 18 日

目录

- 1. 任务.....3
- 2. 小组成员.....3
- 3. 问题描述.....3
- 4. 数据集介绍.....4
- 5. 数据预处理.....6
- 6. 模型选择.....7
 - 6.1 线性方法：线性支持向量机7
 - 6.2 线性方法：LOGISTIC 回归.....8
 - 6.3 非线性方法：KERNEL SVM.....9
 - 6.4 集成学习：随机森林10
 - 6.5 神经网络：RESNET34.....11
 - 6.6 神经网络：RESNET50.....13
 - 6.7 神经网络：GOOGLENET.....14
- 7. 试验结果.....17
 - 7.1 线性支持向量机18
 - 7.2 LOGISTIC 回归.....19
 - 7.3 KERNEL SVM20
 - 7.4 随机森林.....21
 - 7.5 RESNET34.....22
 - 7.6 RESNET50.....23
 - 7.7 GOOGLENET.....24
- 8. 结论.....25

1. 任务

在下列 4 类机器学习算法中，每类选一种算法，对 CIFAR-10 数据集中图片进行分类：

- 线性方法: 线性 SVM、对数几率回归
- 非线性方法: Kernel SVM, 决策树
- 集成学习: Bagging、Boosting
- 神经网络: 自定义合适的网络结构

2. 小组成员

姓名	学号	分工
邓璇	2023E8016082026	训练优化, 撰写报告
郑子豪	2023E8016082043	编写代码, 撰写报告

3. 问题描述

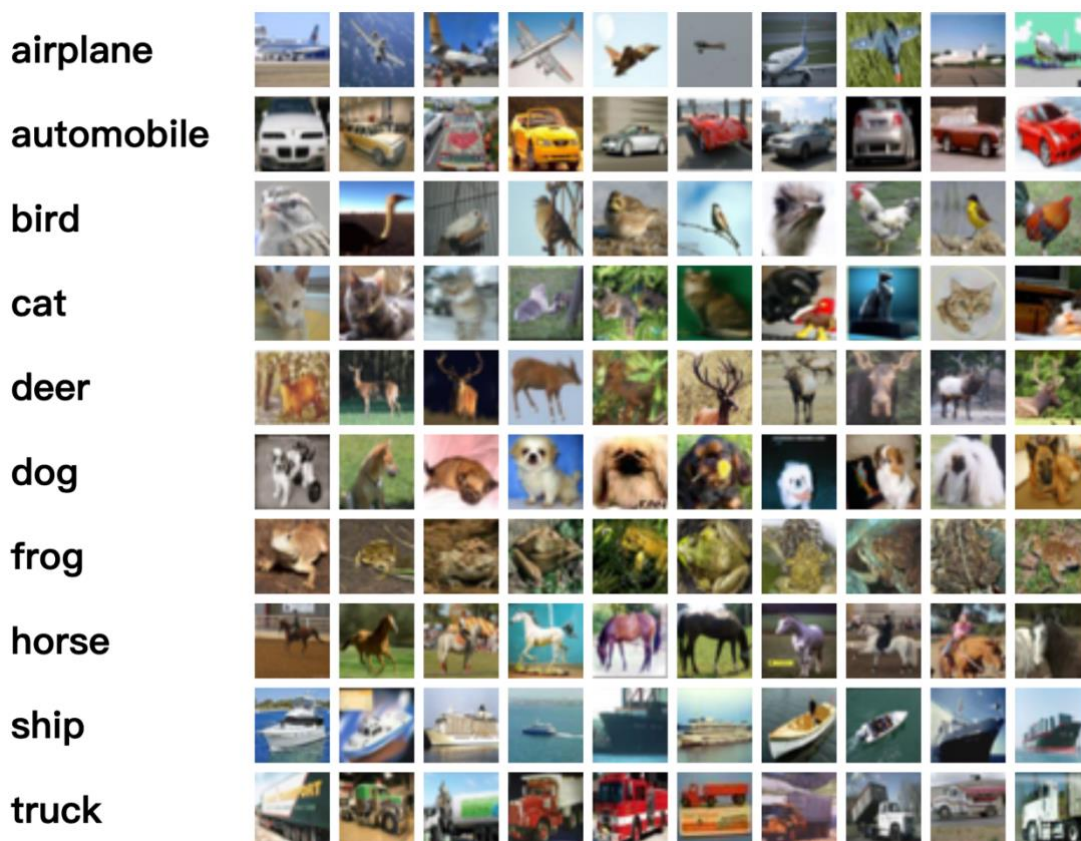
设计和实现一个高效的分类模型，使用 CIFAR-10 数据集中的训练集进行训练，通过特征提取和模型训练，使用测试集进行测试，达到尽可能高的分类准确率。

4. 数据集介绍

CIFAR-10 数据集由 60000 张 32x32 像素的彩色图像组成，分为 10 个不同的类别，每个类别包含 6000 张图像。这些类别包括飞机、汽车、鸟类、猫、鹿、狗、青蛙、马、船和卡车，每个类别的图像都具有高度的多样性和复杂性。

CIFAR-10 数据集的图像被分为训练集和测试集。训练集包含 50000 张图像，用于模型的训练和调优；测试集包含 10000 张图像，用于评估模型的最终性能。这种划分确保了模型能够在未见过的数据上进行有效的性能测试。

CIFAR-10 数据集的每张图片是以被展开的形式存储，每一类的数据表示格式为 uint8，前 1024 个数据表示红色通道，接下来的 1024 个数据表示绿色通道，最后的 1024 个通道表示蓝色通道。数据集展示如下图：



数据集中包含 8 个文件，如下表所示：

文件名	描述
batches.meta	文件存储了每个类别的英文名称。可以用记事本或其他文本文件阅读器打开浏览查看
data_batch_1	这 5 个文件是 CIFAR- 10 数据集中的训练数据。每个文件以二进制格式存储了 10000 张 32×32 的彩色图像和这些图像对应的类别标签。一共 50000 张训练图像
data_batch_2	
data_batch_3	
data_batch_4	
data_batch_5	
test_batch	这个文件存储的是测试图像和测试图像的标签，一共 10000 张
readme.html	数据集介绍文件

5. 数据预处理

1. 随机裁剪

通过对原始图片加入边界填充后进行随机裁剪到 32x32 像素，以增加数据集的多样性并模拟不同的观察角度。填充 4 像素意味着在每个边界上添加 4 像素的空间，然后从这个稍大的图像中随机裁剪出 32x32 的区域。

2. 随机水平翻转

图像有 50% 的概率被水平翻转。这种方法可以增加模型对图像方向的不敏感性，特别是对于不依赖于特定方向的对象分类任务。

3. 转换为张量

这一步将图像从 PIL 格式或 numpy 数组转换为 PyTorch 张量，并自动将像素值从 [0, 255] 缩放到 [0.0, 1.0]，即进行了归一化处理。

4. 归一化

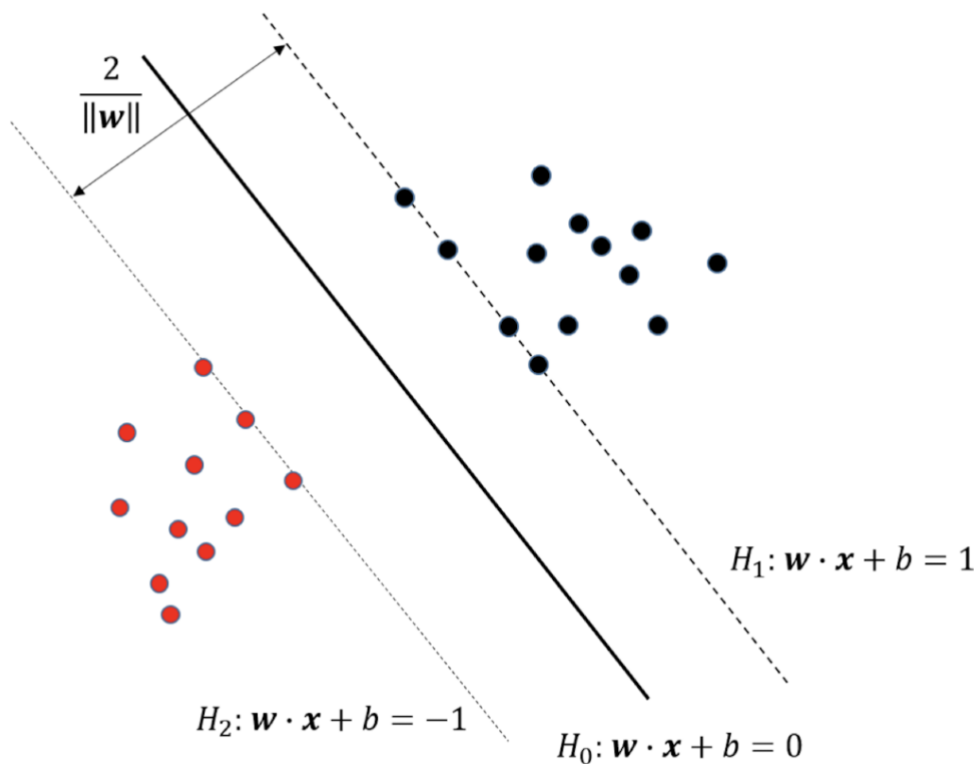
使用固定的均值 (0.5, 0.5, 0.5) 和标准差 (0.5, 0.5, 0.5) 对每个通道的数据进行标准化。这使得输入数据的分布接近均值为 0，标准差为 1，有助于模型训练的稳定性和效率。

6. 模型选择

为了在 CIFAR-10 数据集上执行图像分类任务, 我们选择了线性 SVM、Logistic 回归、Kernel SVM、随机森林以及神经网络五种不同的机器学习算法。每种算法都有其独特的特点和优势, 适用于解决不同的问题。以下是对每种选择的详细解释和理由。

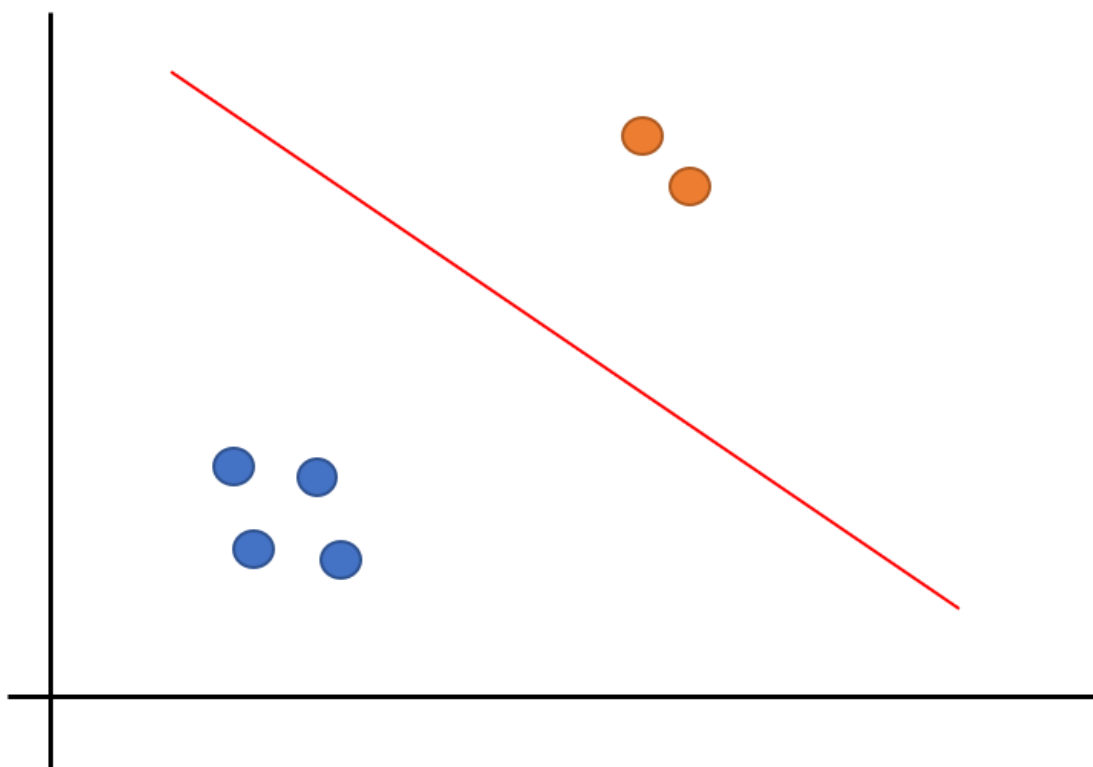
6.1 线性方法：线性支持向量机

线性支持向量机 (SVM) 是一种有效的分类器, 特别适合于特征维度高于样本数的情况。在高维空间中, 线性 SVM 通过寻找最大化类别间边缘的超平面来进行分类。对于 CIFAR-10 数据集, 虽然数据复杂度较高, 但选择线性 SVM 可以帮助我们评估在基础线性假设下模型的表现, 作为对比非线性模型的基准。



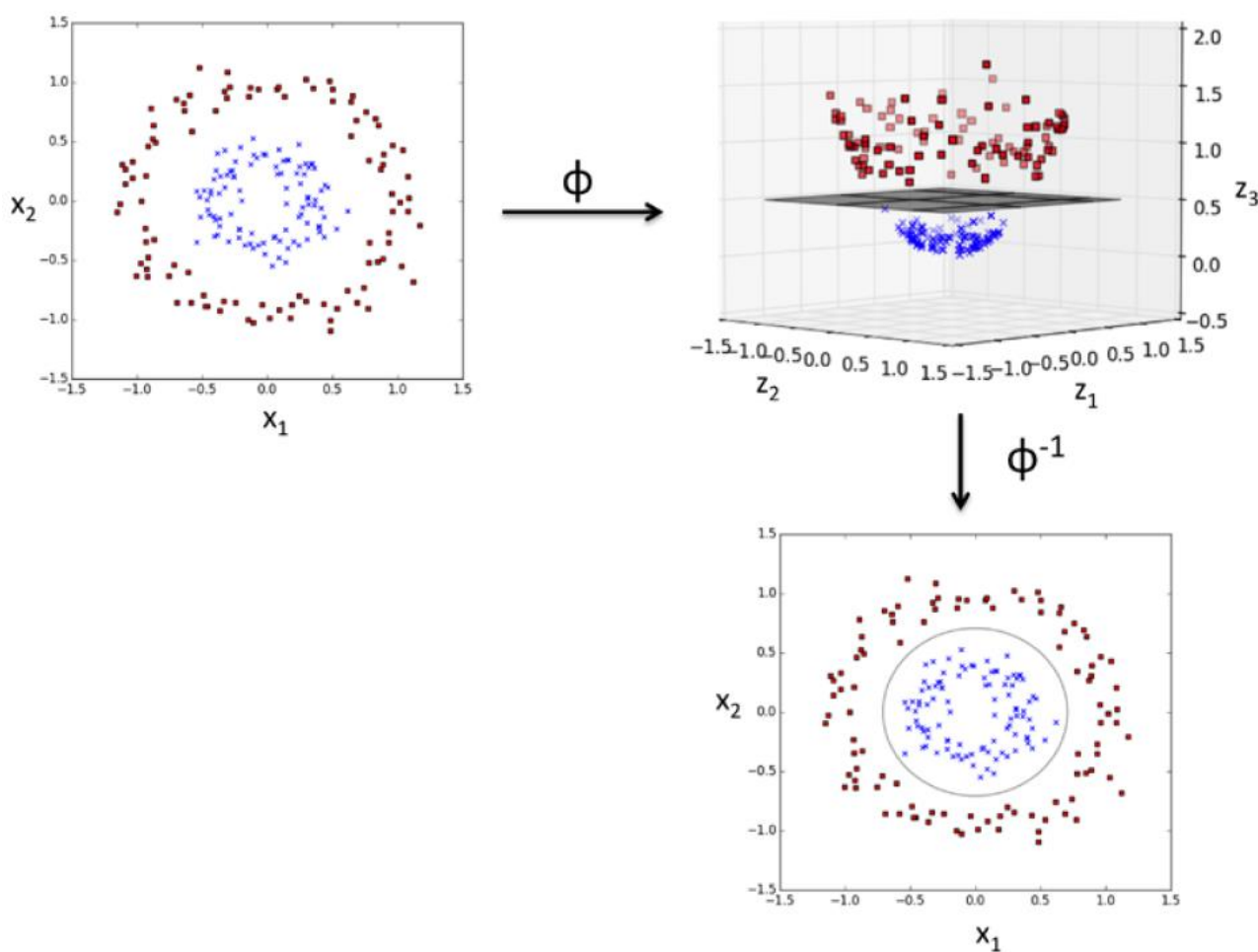
6.2 线性方法：Logistic 回归

Logistic 回归是一种广泛使用的线性分类方法，它通过估计概率来预测离散结果。尽管它本质上是线性的，但它在一些多类分类问题中表现良好，特别是当配合适当的正则化技术时。在 CIFAR-10 上使用 Logistic 回归可以帮助我们理解数据在最基本线性决策边界下的分布情况。



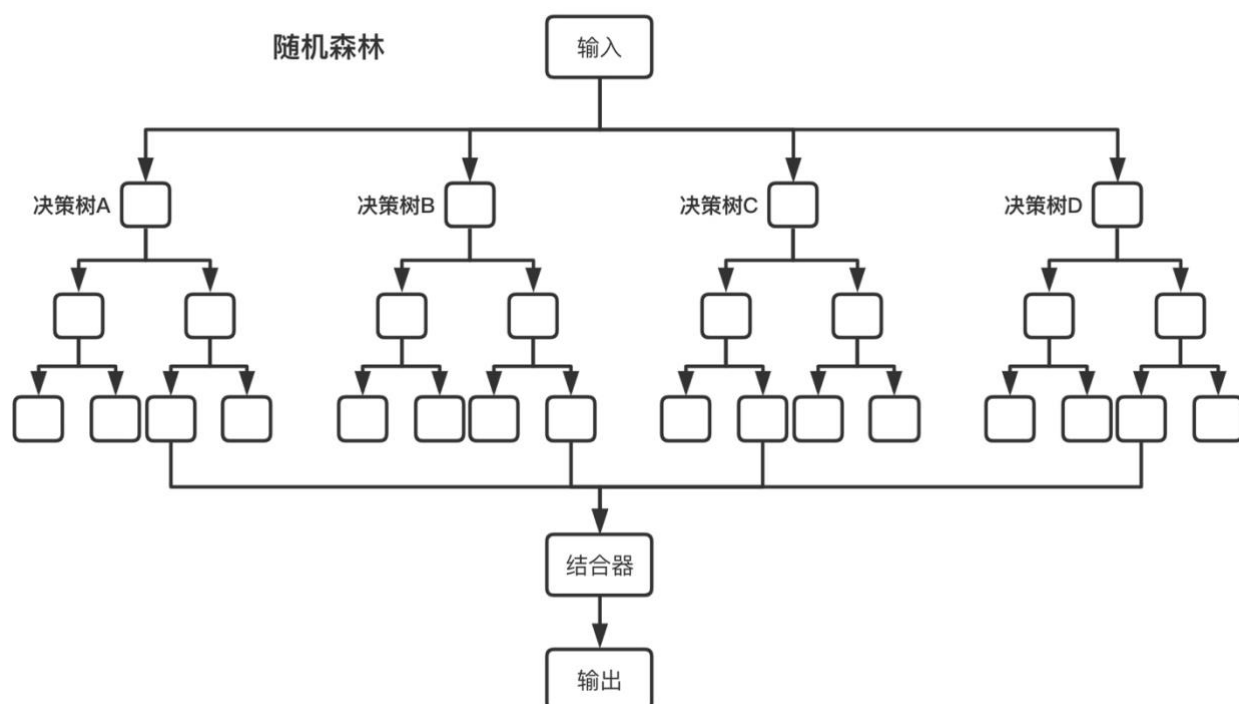
6.3 非线性方法：Kernel SVM

考虑到 CIFAR-10 数据集的复杂性，可能存在线性模型无法有效解决的问题。Kernel SVM 通过引入核函数，允许数据在更高维的空间中被有效分割，从而处理非线性分类问题。常用的核函数包括径向基函数 (RBF)，多项式核等。这种方法在处理图像和其他复杂数据集时表现出更好的性能，因其能捕捉数据中的非线性关系。



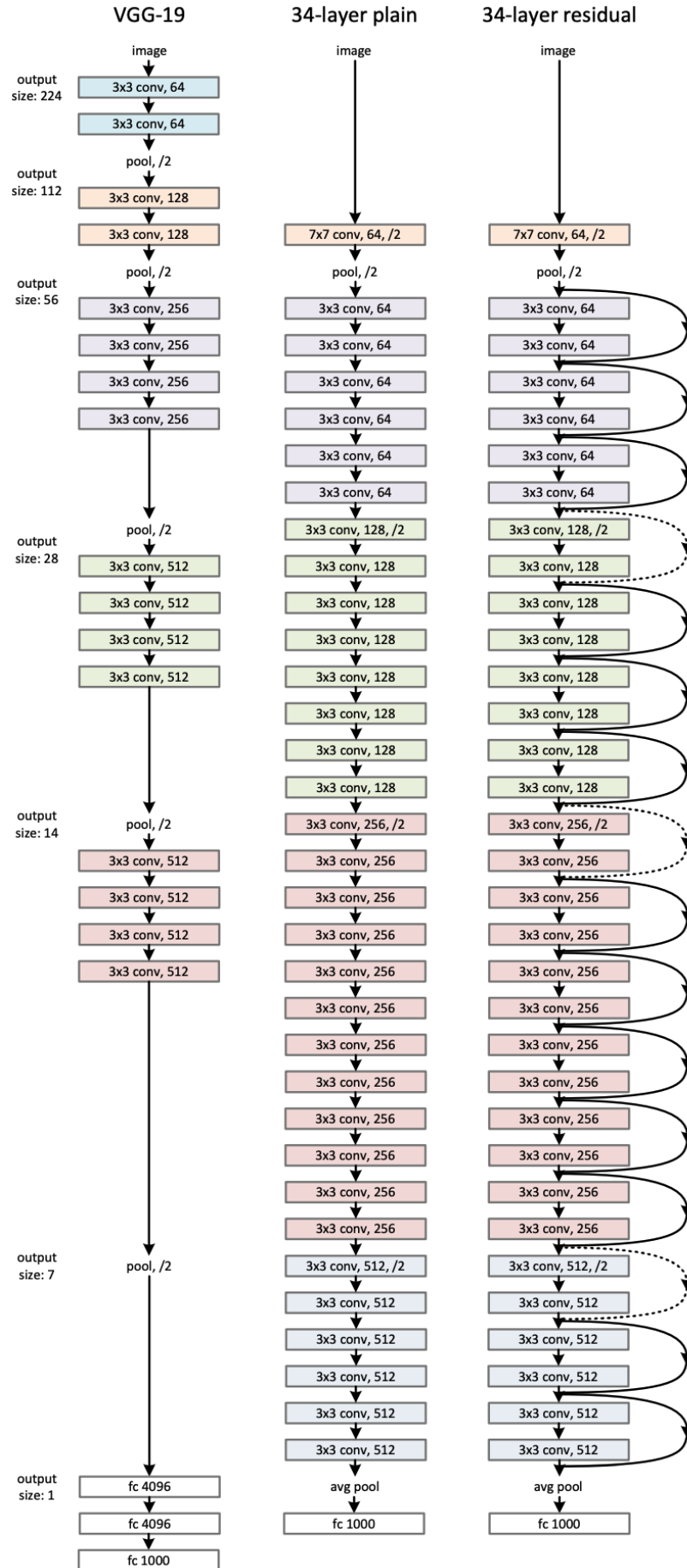
6.4 集成学习：随机森林

随机森林是一种强大的集成学习技术，通过构建多个决策树并汇总它们的预测结果来提高整体预测的准确性和鲁棒性。这种方法对于具有复杂决策边界的分类任务尤为有效，因为它可以自动进行特征选择并处理特征间的相互作用。在 CIFAR-10 这样的复杂图像数据集上，随机森林也许会更适用，因为它能有效地管理大量的特征并防止过拟合。



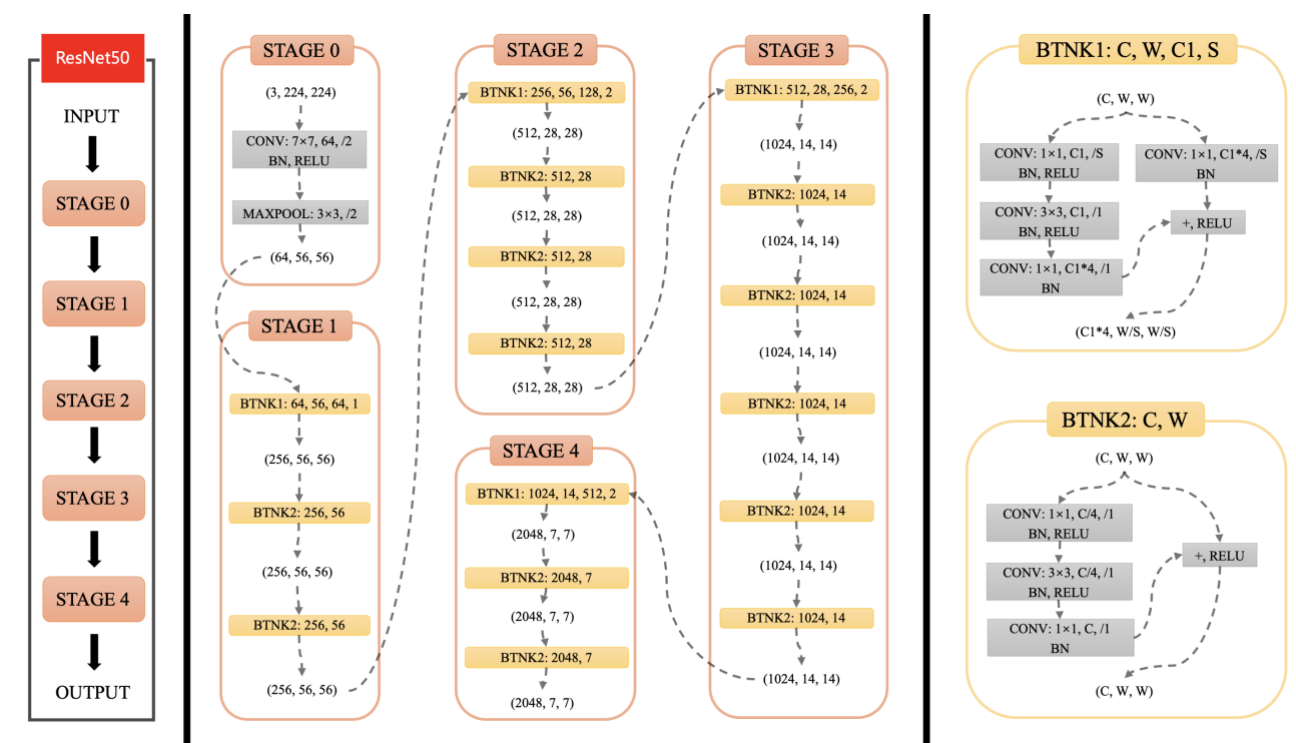
6.5 神经网络：ResNet34

ResNet34 是一种基于残差网络（ResNet）架构的深度学习模型，由 Kaiming He 和他的团队在 2015 年设计。这种模型具备 34 层的网络结构，使用了简单的残差块来构建，其中每个残差块包含两个卷积层。这些残差块中的跳跃连接允许一部分输入直接传递到后续层，帮助解决了在训练深度模型时常见的梯度消失问题。ResNet34 因其较浅的网络深度而在计算资源有限的环境下表现良好，适合用于图像分类、物体检测和图像分割等基本计算机视觉任务，能够有效地进行训练并提供可靠的性能。



6.6 神经网络：ResNet50

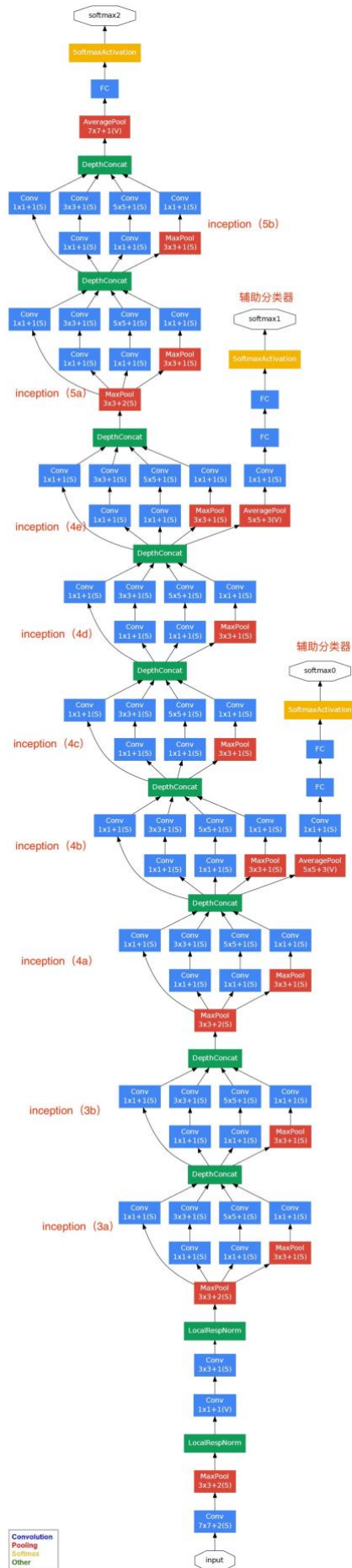
ResNet50 是 ResNet 系列中更深的模型，包含 50 层网络。与 ResNet34 不同的是，这个版本使用了包含三层卷积(1×1 , 3×3 , 1×1)的瓶颈结构的残差块，旨在通过降维和升维操作减少计算负担同时增加网络的深度。这种瓶颈结构使得 ResNet50 不仅在参数效率上得到改善，也能够捕捉更复杂的特征，从而提高了模型处理复杂视觉任务的能力。由于其深度和复杂度，ResNet50 在需要高精度解决方案的应用中尤为有效，如大规模图像分类和精细的物体识别，是许多高级视觉应用的首选模型。



6.7 神经网络：GoogLeNet

GoogLeNet 的主要特点是它的 Inception 模块,这一模块允许网络在不同尺度上处理数据,同时保持计算效率。每个 Inception 模块包含了多个并行的卷积层和池化层,这些层的不同尺寸的滤波器可以捕获不同层次的图像细节。此外,该模型使用了全局平均池化层而不是全连接层来减少模型的参数数量,这样做不仅有助于减少过拟合,还能减轻模型的计算负担。

在训练过程中,GoogLeNet 还引入了辅助分类器来加速早期层的训练,并提供额外的梯度信号来优化学习过程,虽然这些辅助分类器在实际推理阶段不被使用。此外,由于 GoogLeNet 没有采用局部响应归一化 (LRN),其结构简化同时还保持了良好的性能。



通过这些模型，我们旨在探索不同算法在处理图像识别任务时的表现，从简单的线性模型到复杂的非线性和集成方法，乃至神经网络。这种多样化的方法不仅可以帮助我们识别出最适合 CIFAR-10 数据集的模型，还可以深入了解不同算法在实际应用中的优缺点，为后续的模型选择和优化提供实验依据。

7. 试验结果

为了全面评估我们所选择模型在 CIFAR-10 数据集上的分类性能，我们计划使用三种主要的评价指标：混淆矩阵 (confusion matrix)、分类报告 (classification report) 和准确率 (accuracy score)。

- **混淆矩阵**

将帮助我们直观地观察模型在各个类别上的预测正确与错误情况，这对于理解模型在特定类别上的强项和弱点非常有用。

- **分类报告**

将提供包括精确率、召回率、F1 分数等关键指标的详细信息，这些都是衡量模型在各个类别预测性能的重要工具。

- **准确率**

给出模型对所有测试数据进行正确分类的比例，是评价模型整体性能的一个基本指标。

综合这些评价方法，我们能够得到一个关于模型表现的详细和全面的分析，进而为模型优化提供依据。

7.1 线性支持向量机

- 混淆矩阵

```
[[622  24  57  17  21  15  20  27 143  54]
 [ 32 646  19  42   9  17  16  21  49 149]
 [ 83  23 410  89 132  62 112  50  24  15]
 [ 32  30  91 386  56 166 121  42  25  51]
 [ 48  14 164  66 435  43 124  68  22  16]
 [ 22  14  87 199  70 431  84  48  21  24]
 [ 11  18  64  84  95  45 642  14  13  14]
 [ 36  22  50  76  74  68  34 566  16  58]
 [ 83  65  16  22  20  19  15  18 689  53]
 [ 43 150  10  40  11  14  28  38  52 614]]
```

- 分类报告及准确率

class	precision	recall	f1-score	support
0	0.61	0.62	0.62	1000
1	0.64	0.65	0.64	1000
2	0.42	0.41	0.42	1000
3	0.38	0.39	0.38	1000
4	0.47	0.43	0.45	1000
5	0.49	0.43	0.46	1000
6	0.54	0.64	0.58	1000
7	0.63	0.57	0.60	1000
8	0.65	0.69	0.67	1000
9	0.59	0.61	0.60	1000
accuracy			0.54	10000
macro avg	0.54	0.54	0.54	10000
weighted avg	0.54	0.54	0.54	10000

7.2 Logistic 回归

- 混淆矩阵

```
[[493  44  47  29  17  25  29  51 193  72]
 [ 60 465  31  31  19  38  39  49  83 185]
 [ 99  48 273  74 112  77 157  82  51  27]
 [ 42  66  86 225  64 186 158  51  49  73]
 [ 61  31 121  56 278  96 184 105  36  32]
 [ 41  60  89 150  85 319  94  69  55  38]
 [ 25  46  66 107  86  88 485  45  21  31]
 [ 49  56  67  58  86  77  38 438  47  84]
 [141  74  19  24   7  34  16  21 566  98]
 [ 66 180  18  25  13  25  47  54  94 478]]
```

- 分类报告及准确率

class	precision	recall	f1-score	support
0	0.46	0.49	0.47	1000
1	0.43	0.47	0.45	1000
2	0.33	0.27	0.30	1000
3	0.29	0.23	0.25	1000
4	0.36	0.28	0.31	1000
5	0.33	0.32	0.32	1000
6	0.39	0.48	0.43	1000
7	0.45	0.44	0.45	1000
8	0.47	0.57	0.52	1000
9	0.43	0.48	0.45	1000
accuracy			0.40	10000
macro avg	0.40	0.40	0.40	10000
weighted avg	0.40	0.40	0.40	10000

7.3 Kernel SVM

- 混淆矩阵

```
[[622  24  57  17  21  15  20  27 143  54]
 [ 32 646  19  42   9  17  16  21  49 149]
 [ 83  23 410  89 132  62 112  50  24  15]
 [ 32  30  91 386  56 166 121  42  25  51]
 [ 48  14 164  66 435  43 124  68  22  16]
 [ 22  14  87 199  70 431  84  48  21  24]
 [ 11  18  64  84  95  45 642  14  13  14]
 [ 36  22  50  76  74  68  34 566  16  58]
 [ 83  65  16  22  20  19  15  18 689  53]
 [ 43 150  10  40  11  14  28  38  52 614]]
```

- 分类报告及准确率

class	precision	recall	f1-score	support
0	0.61	0.62	0.62	1000
1	0.64	0.65	0.64	1000
2	0.42	0.41	0.42	1000
3	0.38	0.39	0.38	1000
4	0.47	0.43	0.45	1000
5	0.49	0.43	0.46	1000
6	0.54	0.64	0.58	1000
7	0.63	0.57	0.60	1000
8	0.65	0.69	0.67	1000
9	0.59	0.61	0.60	1000
accuracy			0.54	10000
macro avg	0.54	0.54	0.54	10000
weighted avg	0.54	0.54	0.54	10000

7.4 随机森林

- 混淆矩阵

```
[[567  42  57  23  28  19  25  24 165  50]
 [ 32 540  22  43  23  27  43  32  57 181]
 [ 93  42 350  68 143  71 117  54  31  31]
 [ 48  42  77 276  81 189 134  66  22  65]
 [ 57  23 143  55 406  38 149  83  20  26]
 [ 34  22  85 156  75 395  86  85  29  33]
 [ 17  42  75  75 109  47 563  24   8  40]
 [ 53  44  52  68  92  82  52 450  18  89]
 [ 86  89  18  20  27  34   8  20 622  76]
 [ 46 165  24  28  22  23  26  44  77 545]]
```

- 分类报告及准确率

class	precision	recall	f1-score	support
0	0.55	0.57	0.56	1000
1	0.51	0.54	0.53	1000
2	0.39	0.35	0.37	1000
3	0.34	0.28	0.30	1000
4	0.40	0.41	0.40	1000
5	0.43	0.40	0.41	1000
6	0.47	0.56	0.51	1000
7	0.51	0.45	0.48	1000
8	0.59	0.62	0.61	1000
9	0.48	0.55	0.51	1000
accuracy			0.47	10000
macro avg	0.47	0.47	0.47	10000
weighted avg	0.47	0.47	0.47	10000

7.5 ResNet34

- 混淆矩阵

```
[[960  0  11  5  1  0  1  2 17  3]
 [  2 982  0  0  1  0  0  0  4 11]
 [ 11  0 939 10 12 14  9  3  2  0]
 [  6  1  9 887  9 56 21  6  3  2]
 [  2  0  7  6 967  6  6  5  1  0]
 [  2  1 12 44 12 918  1  9  0  1]
 [  3  0 14 11  2  0 968  1  0  1]
 [  4  0  1  9  7 13  0 964  2  0]
 [ 13  4  3  1  1  1  1  0 971  5]
 [  4 19  1  3  0  1  0  0 10 962]]
```

- 分类报告及准确率

class	precision	recall	f1-score	support
0	0.95	0.96	0.96	1000
1	0.98	0.98	0.98	1000
2	0.94	0.94	0.94	1000
3	0.91	0.89	0.90	1000
4	0.96	0.97	0.96	1000
5	0.91	0.92	0.91	1000
6	0.96	0.97	0.96	1000
7	0.97	0.96	0.97	1000
8	0.96	0.97	0.97	1000
9	0.98	0.96	0.97	1000
accuracy			0.95	10000
macro avg	0.95	0.95	0.95	10000
weighted avg	0.95	0.95	0.95	10000

7.6 ResNet50

- 混淆矩阵

```
[[951  0 10  5  2  0  0  2 23  7]
 [ 3 978  0  0  0  0  0  0  2 17]
 [10  0 933  9 15 12 12  6  3  0]
 [ 4  1  9 877 15 64 17  5  5  3]
 [ 3  0 10  9 963  6  4  5  0  0]
 [ 1  0  8 51 13 913  3  9  1  1]
 [ 2  0  3  8  4  3 977  1  0  2]
 [ 5  0  2  5 12 10  3 963  0  0]
 [11  3  3  2  2  0  0  0 975  4]
 [ 3 17  2  3  0  0  2  0  3 970]]
```

- 分类报告及准确率

class	precision	recall	f1-score	support
0	0.96	0.95	0.95	1000
1	0.98	0.98	0.98	1000
2	0.95	0.93	0.94	1000
3	0.91	0.88	0.89	1000
4	0.94	0.96	0.95	1000
5	0.91	0.91	0.91	1000
6	0.96	0.98	0.97	1000
7	0.97	0.96	0.97	1000
8	0.96	0.97	0.97	1000
9	0.97	0.97	0.97	1000
accuracy			0.95	10000
macro avg	0.95	0.95	0.95	10000
weighted avg	0.95	0.95	0.95	10000

7.7 GoogLeNet

- 混淆矩阵

```
[[975  3  5  3  0  0  1  0 10  3]
 [  4 972  0  0  0  0  0  0  5 19]
 [ 11  0 936 16 12  8 14  2  1  0]
 [  3  1  9 886 15 60 15  6  2  3]
 [  4  0  8 10 960  6  5  6  1  0]
 [  0  1  7 44  9 931  1  5  0  2]
 [  2  0  8 11  3  4 970  1  0  1]
 [  5  0  4 13  7 10  1 959  1  0]
 [ 16  3  1  3  0  0  0  0 971  6]
 [  6 19  0  2  0  0  1  1  4 967]]
```

- 分类报告及准确率

class	precision	recall	f1-score	support
0	0.95	0.97	0.96	1000
1	0.97	0.97	0.97	1000
2	0.96	0.94	0.95	1000
3	0.90	0.89	0.89	1000
4	0.95	0.96	0.96	1000
5	0.91	0.93	0.92	1000
6	0.96	0.97	0.97	1000
7	0.98	0.96	0.97	1000
8	0.98	0.97	0.97	1000
9	0.97	0.97	0.97	1000
accuracy			0.95	10000
macro avg	0.95	0.95	0.95	10000
weighted avg	0.95	0.95	0.95	10000

8. 结论

本项目的目标是使用不同的机器学习模型对 CIFAR-10 数据集进行图像分类，并评估这些模型的性能。我们选择了线性 SVM、Logistic 回归、Kernel SVM、随机森林以及神经网络模型，包括 ResNet34、ResNet50 和 GoogLeNet，每种模型各有所长，最终的准确率如下所示：

模型	SVM	Logistic	kernelSVM	RandomForest	ResNet34	ResNet50	GoogLeNet
accuracy(%)	54.0	40.2	54.41	47.14	95.18	95.0	95.27

通过对比不同模型的表现，我们发现所选的三种神经网络模型在分类准确率和处理图像特征的能力上明显优于其他模型。这些深度学习模型能够有效地从图像中提取复杂的特征，并且具有更好的泛化能力，这在 CIFAR-10 的多样化图像分类任务中尤为重要。

线性模型如线性 SVM 和 Logistic 回归虽然在运算速度上较快，但在准确率和模型泛化方面不如深度学习模型。Kernel SVM 通过引入核函数提高了模型的非线性分类能力，但仍然无法与最先进的深度学习模型相匹配。随机森林作为一种集成学习方法，在处理非线性问题时表现较好，但在图像分类的复杂性面前，效果仍有限。

在深度学习模型中，ResNet34 和 ResNet50 的表现极好，这得益于它们的残差学习架构，该架构通过引入跳跃连接解决了深度网络中的梯度消失问题，允许模型学习更深层的特征。GoogLeNet 通过其独特的 Inception 模块，有效地在不增加计算负担的前提下增加了网络的宽度和深度，显示了卓越的分类性能。

总结而言，尽管传统机器学习模型在某些简单任务上表现可靠，但在处理复杂的图像数据集，如 CIFAR-10 时，深度学习模型特别是具有先进架构的神经网络显示出了更高的效率和更强的性能。因此，对于未来的图像分类任务，推荐使用深度学习方法，如 ResNet 和 Inception 系列模型，以获得更准确的结果和更高的效率。