同济大学

TONGJI UNIVERSITY

《机器学习》

实验报告（大作业）

实验名称	Lab2 分类实验
小组成员	(数据预处理，传统方法，手写贝叶斯，报告) 深度学习，模型对比，手动 ViT 架构，报告） （模型运行）
学院（系）	电子与信息工程学院
专　　业	计算机科学与技术
任课教师	李洁
日　　期	2023 年 11 月 16 日

机器学习实现Cifar10图片分类

本次任务基于 Cifar-10 数据集，使用传统的机器学习方法实现了 Cifar10 图片分类任务。传统的机器学习方法由于不支持 Cifar10 的众多特征，我们首先选择了 PCA 主成分分析、Hog 特征提取方法以及 Daisy 特征提取方法对原始的输入数据特征进行一定的取舍。在传统的机器学习方法中，我们使用了包括高斯贝叶斯方法，伯努利贝叶斯方法，支持向量机(SVM)、线性支持向量机(Linear SVM)、KNN、随机森林、逻辑斯蒂回归、决策树、XGBoost 和 AdaBoost。为了和传统的机器学习算法进行对比，我们使用了深度学习的策略，实现端到端的机器学习实现。在深度学习的模型中，我们使用 cnn，deepcnn，vgg 和 resnet 进行实现。同时，为了优化对比的效果和最终分类的准确率，我们使用了先进的 ViT 模型，手动构建视觉领域的 Transformer 架构实现图片分类问题。由于这也是当前 Cifar10 任务的 SOTA，我们也构筑优化了相应的模型。

I. 任务目标与数据集说明

本任务的本质是实现图像的分类任务。CIFAR-10 是一个流行的计算机视觉数据集，用于图像识别和机器学习领域的研究和测试。CIFAR-10 的主要任务目标是图像分类。CIFAR-10 包含 10 个类别的图像，例如猫、狗、飞机等。每个类别有 6000 张图像，数据集总共有 60000 张图像。其中 50000 张用于训练，10000 张用于测试。CIFAR-10 的广泛使用反映了其在提供实际、可挑战的机器学习问题方面的价值，同时其规模又适合于快速实验和学习。

图 1 数据集样态

II. 数据清洗与预处理

数据集使用方法：使用 tensorflow 中暴露的 cifar10 接口进行调用.这个调用的好处是只用下面的几行代码就可以实现数据集到 numpy 的 array 类型的转换：

```
# 加载CIFAR-10数据集
(x_train, y_train), (x_test, y_test) = datasets.cifar10.load_data()
```

在 CIFAR-10 数据集上进行机器学习算法分析之前进行特征选择和降维是有益的。这可以提高计算效率，降维可以减少数据集的大小，从而减少模型训练所需的计算资源和时间。这对于处理有限的计算资源或在较短时间内进行多次实验的情况尤为重要。特征选择有助于去除数据中的噪声和冗余信息。在 CIFAR-10 这样的图像数据集中，可能存在一些不重要或重复的像素值，它们对于分类任务并无帮助，甚至可能干扰模型的学习过程。并且通过选取最相关的特征和减少维度，可以提高模型的泛化能力。这意味着模型在未见数据上的表现可能会更好，减少过拟合的风险。在某些情况下，原始数据的处理速度可能非常慢。通过降维，可以加快数据处理的速度，尤其是在进行复杂的计算时。某些机器学习算法在处理高维数据时效果不佳或计算量过大。通过降维，可以使这些算法在处理后的数据集上运行得更有效。

虽然在 CIFAR-10 这样的标准数据集上进行特征选择和降维可能不是必需的，尤其是考虑到现代深度学习模型通常能够处理原始数据，但在特定情况下，这些步骤可能有助于提高效率和性能。

下面我们重点介绍我们在本次实验中使用的特征提取方法。

A. PCA方法

主成分分析（PCA）是一种统计技术，用于简化数据集的复杂性，同时尽可能保留原始数据中的重要信息。它通过线性变换将数据投影到较低维度的空间中，这个过程涉及到识别数据中的“主成分”，也就是最大化数据方差的方向。在实践中，这意味着 PCA 可以找到减少数据集维度的同时保留最多变异性的方法。这个技术特别有用于处理高维数据集，如图像或者复杂的统计数据，因为它可以揭示数据的内在结构，减少数据量，同时减少计算成本。PCA 的一个关键优点是它是非监督的，意味着它不依赖于数据的标签或结果。结果是一组线性无关的向量，通常称为主成分，它们定义了数据在新空间中的坐标系。通过这种方式，PCA 能够有效地揭示数据中的关键特征和模式，为后续的数据分析和机器学习提供了一个更简洁和更有洞察力的视角。

主成分分析（PCA）的数学原理主要基于线性代数和统计学。核心思想是通过正交变换将原始数据变换到一个新的坐标系统，使得在这个新坐标系统中，数据的方差最大化。下面是 PCA 的一些关键数学概念和公式：

协方差矩阵：首先计算数据的协方差矩阵。协方差矩阵描述了数据各维度间的相关性。对于数据矩阵 X（已中心化，即每个特征的均值为 0），其协方差矩阵 Σ 计算公式为：

$$\Sigma = \frac{1}{n-1} X^T X$$

其中n是样本数量。

特征值和特征向量：接下来，计算协方差矩阵的特征值和特征向量。特征向量表示数据中方差最大的方向，而特征值表示该方向的方差量。数学上，对于协方差矩阵Σ，我们要解以下方程以找到特征值λ和特征向量v：$\Sigma v = \lambda v$.

排序并选择主成分：将特征值按降序排列，并选择前k个最大的特征值对应的特征向量，这些特征向量构成了数据的主成分。这一步骤决定了新特征空间的维度。

数据投影：最后，将原始数据投影到选定的主成分上，形成降维后的数据。这可以通过以下公式实现：

$$Y = XV$$

其中V是包含选定特征向量的矩阵，Y是投影后的数据。

PCA 通过这种方式能够提取数据中最重要的特征，并将高维数据有效地降维到更低的维度空间中，有助于数据的可视化和进一步分析。

下面的代码实现了 PCA 的主成分分析。

```
x_train_pca = x_train.reshape(x_train.shape[0],-1)
x_test_pca = x_test.reshape(x_test.shape[0],-1)
pca = PCA()
pca.fit_transform(x_train_pca)
# 查找保留特征
k = 0
total = sum(pca.explained_variance_)
current_sum = 0
while(current_sum / total < 0.99):
    current_sum += pca.explained_variance_[k]
    k += 1
pca = PCA(n_components=k, whiten=True)
x_train_pca = pca.fit_transform(x_train_pca)
x_test_pca = pca.transform(x_test_pca)
```

B. Hog方法

特征提取中的 HOG(Histogram of Oriented Gradient)算法是一种用于图像处理和计算机视觉中的特征描述符。它通过计算和统计图像局部区域的梯度方向直方图来构建特征,可以有效描述图像的局部形状和纹理信息。

具体来说,HOG 算法首先将图像分割成小的连通区域即单元,然后计算每个像素点的梯度方向和梯度模值。对每个单元内的所有像素点的梯度方向直方图进行统计,得到该单元的 HOG 特征描述符。之后可以对重叠的局部单元组进行组合,形成更大局部区域的 HOG 描述符。整幅图像的 HOG 特征就是将所有局部单元的 HOG 描述符组合起来。

HOG 特征保留了局部形状和纹理信息,对光照变化和微小形变也有一定的鲁棒性。它经常被用来做图像分割、目标检测、行人检测等任务,尤其适用于需要对局部形状和边缘进行建模分析的问题。总体来说,HOG 是一种较为成熟有效的图像特征描述方法,广泛应用于计算机视觉与图像分析领域。

下面给出 Hog 方法的具体实现：

```
def getFeat_Hog(TrainData,TestData):
    train_feature=np.zeros(shape=(50000,324))
    test_feature=np.zeros(shape=(10000,324))
    cnt=0
    for data in TrainData:
        image = data
        gray = rgb2gray(image)/255.0
        fd = hog(gray, orientations=9, pixels_per_cell=(8,8), cells_per_block=(3,3))
        train_feature[cnt] = fd
```

```
        cnt = cnt + 1
    print("Train features are extracted and saved.")
    cnt=0
    for data in TestData:
        image = data
        gray = rgb2gray(image)/255.0
        fd = hog(gray, orientations=9, pixels_per_cell=(8,8), cells_per_block=(3,3))
        test_feature[cnt]=fd
        cnt=cnt+1
    print("Test features are extracted and saved.")
    return train_feature,test_feature
```

C. Daisy方法

DAISY 算法是一种用于图像处理和计算机视觉中的局部特征描述符,它可以有效地描述图像的纹理和形状信息。DAISY 算法的名称来源于它所提取的特征与花朵(Daisy Flower)相似。

具体来说,DAISY 会首先在提取特征点的位置构建多个不同大小的圆形区域,这些圆形区域类似花朵的花瓣。每个圆形区域内部则根据径向和角度分割出小块,对每个小块内像素的梯度直方图进行统计,得到这个区域块的方向特征。之后,对于不同半径的所有圆形区域,利用高斯加权的方法聚合所有区域块的特征,构成最终的 DAISY 特征描述符。

DAISY 特征对光照变化和图像旋转具有一定的鲁棒性,并且计算效率高,适合实时视觉应用。它经常被用于图像匹配、目标跟踪、图像立体匹配等任务中。总体上,DAISY 是一种计算高效且对光照和旋转变化鲁棒的局部特征描述符,可以充分描述图像的纹理和形状特征,适用于实时图像分析与理解任务。

由于 Daisy 方法实现和上面的 Hog 特征提取类似，所以不在这里讨论。

III. 特征提取机器学习算法

这一部分我们将论述特征提取的诸多机器学习算法。我们主要选取了贝叶斯算法，决策树算法，支持向量机算法，随机森林算法，KNN临近算法，逻辑斯蒂回归算法，XgBoost算法以及AdaBoost算法。在实现过程中我们针对其中的一些算法实现了手动的实现，也调整了一些算法对应的超参数。以下是对其中一些算法详尽的分析。

A. 贝叶斯算法

机器学习中的贝叶斯分类算法是基于贝叶斯定理与概率统计建立的一类分类模型。它利用先验知识对不同类别发生的概率进行建模,然后根据数据计算后验概率,将实例分配到后验概率最大的目标类别中。

具体来说,贝叶斯分类器会根据训练数据估计出每种类别的先验概率分布,例如不同类别的特征值出现的频率。在对新样本进行分类时,贝叶斯分类器计算样本属于每个类别的后验概率,即先验概率与 Likelihood 的乘积,然后将其归类到后验概率最大的类别中。

贝叶斯分类方法根据贝叶斯规则对 evidence 与 hypothesis 进行联合建模,分类决策规则朴素合理,也不易过拟合数据。常用的贝叶斯分类器包括高斯朴素贝叶斯与伯努利朴素贝叶斯。其中,高斯朴素贝叶斯假设每个特征符合高斯分布,用于处理连续值特征;而伯努利朴素贝叶斯假设特征符合伯努利分布,通常用于处理文本分类等离散特征空间问题。

a. 伯努利朴素贝叶斯

朴素贝叶斯算法是基于贝叶斯定理与特征条件独立假设而推导出的分类方法。具体来说,对于给定的训练数据集和未知类别的样本 X,贝叶斯定理指出后验概率$P(C_k|X)$与类先验概率$P(C_k)$以及类别条件概率$P(X|C_k)$相关。其中,$P(C_k|X)$表示在特征 X 条件下样本属于类C_k的概率。而朴素贝叶斯算法假设特征之间相互独立,即$P(X|C_k)$可以分解为每个特征条件概率的乘积。这简化了计算,是朴素的原因。伯努利朴素贝叶斯专门用于特征取离散值的情况,即每个$P(X_i|C_k)$服从伯努利分布。根据最大后验概率准则,样本X会被归类到$P(C_k|X)$最大的类,也就是后验概率最大的类。

通过求解联合概率并运用 Bayes 定理,我们可以得到朴素伯努利贝叶斯的分类决策规则:

将样本X归类到$P(C_k) * \Pi\ P(X_i|C_k)$ 最大的类C_k中。

这需要根据训练数据统计$P(C_k)$和 $P(X_i|C_k)$,即特征在各类别中的分布情况。这构成了朴素伯努利贝叶斯算法的数学原理基础。

b. 高斯朴素贝叶斯

朴素贝叶斯算法同样基于贝叶斯定理,并假设特征之间条件独立。不同的是,朴素高斯贝叶斯假定每个特征都符合高斯分布,适用于处理连续值特征。

具体来说,对于类别C_k,每个连续特征X_i都服从一个高斯分布,描述为参数μik和σik^2。其中μik是特征X_i在类C_k上的均值,σik^2是方差。

根据贝叶斯定理,后验概率$P(C_k|X)$与类先验概率$P(C_k)$及特征条件概率$P(X|C_k)$相关。这里$P(X|C_k)$可以表示为每个特征X_i的高斯分布概率密度函数的乘积。

将贝叶斯定理展开,我们可以得到朴素高斯贝叶斯的分类决策规则:

将样本X归类到$P(C_k) * \Pi P(X_i|C_k)$最大的类C_k中。

这里$P(X_i|C_k)$由对应高斯分布的概率密度函数表示。训练时需要估计出每个类C_k的均值μik和方差σik^2。分类时计算每个类的后验概率并归类到最大者。

这构成了朴素高斯贝叶斯的基本原理,通过假设特征服从高斯分布,建立类别条件概率,实现简单有效的分类。

综上,贝叶斯算法是一种基于概率与决策论的分类方法,通过最大后验概率规则对实例进行分类,能够有效地集成先验知识,对缺失数据也较为鲁棒。它是机器学习中一类简单实用且效果较好的分类算法。具体的实现可以参考我们的附录笔记本。同样的我们也实现了手动的贝叶斯算法,核心代码如下:

```
# 根据最小风险贝叶斯模型进行分类
def calculatePostrior(matrixE, matrixC, pc, test_data, feature_count):
    risk = []
    for i in range(10):
        r = 0
        for j in range(10):
            p_x_wj = 1
            for k in range(feature_count):
                p_x_wj = p_x_wj * math.exp(- (test_data[k] - matrixE[j][k]) ** 2 / (2 *
matrixC[j][k])) / math.sqrt(2 * math.pi * matrixC[j][k])
            r += matrixLambda[i][j] * pc[i] * p_x_wj
        risk.append(r)
    return risk.index(min(risk))
```

B. 决策树算法

决策树是一种基于特征递归二分的树形预测模型。它可以表示为一个流程图,包含内部节点(表示属性测试)和叶节点(表示类别)。决策树学习算法的目标是找出最优属性进行划分,使得同一叶节点的样本纯度最高。

决策树的最大深度会影响模型的复杂程度。深度小的决策树模型较简单,对训练数据的拟合程度较差,容易欠拟合。深度大的决策树能够适应训练数据的细微模式,可能会过拟合。

一般来说,较浅的决策树学习到的是数据中的更泛化的模式,这些关键的决策特征对最终结果影响大,树较易解释。而较深的决策树可以学习到训练数据细微的模式,拟合程度高,但也更容易过拟合和学习到噪声,同时树的解释性下降。

所以决策树学习需要在深度与泛化能力间进行 trade-off。控制最大深度可以避免决策树过深而过拟合。预剪枝和后剪枝也可起到正则化的效果。设置合适的决策树深度既能学到关键模式,又能避免过拟合,是决策树学习的关键。

在我们的实验中尝试了在三个预处理方式下的不同最大深度的决策树计算方法。下面是我们得到的结论:

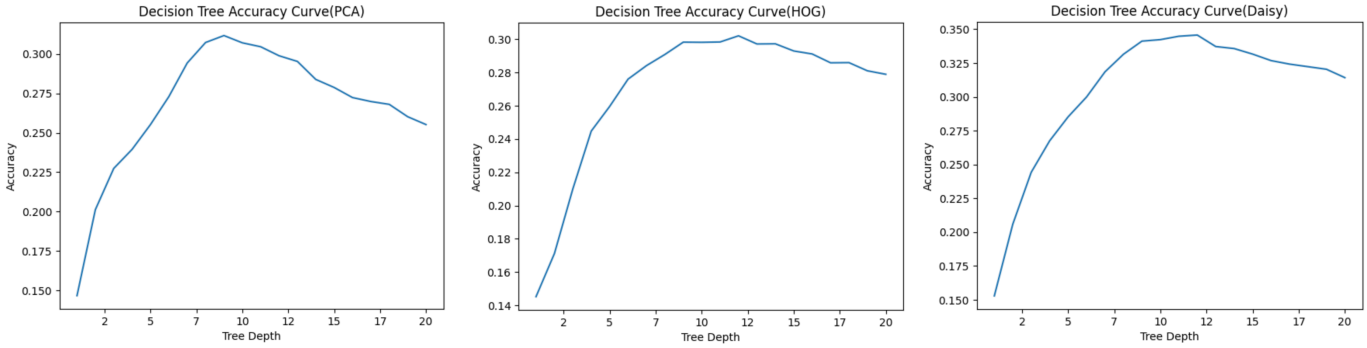

图 2 决策树不同树深度计算准确率结果(三种预处理方法)

我们最终得到的模型使用的最大深度参数为 10.

C. KNN算法

K 近邻算法是一种基于实例学习的非参数分类算法。它的核心思想是,将未知类别的样本点归类到其 K 个最近邻训练样本所属的类别中。K 近邻算法的 K 值即最近邻居的数量,对模型的复杂度和性能有重要影响。

当 K 值较小时,模型更容易学习训练数据的局部模式,并对训练数据拟合较好,但容易过拟合。当 K 值较大时,分类决策则依赖于较多的样本,模型相对简单,对个别样本点的噪声或异常值影响较小,有较好的泛化能力。但是 K 值过大会导致类别界限不明显,从而影响分类性能。

所以,K 值的选择需要平衡这两方面因素。一般通过交叉验证选择最优 K 值。较小的 K 值容易过拟合但性能较好;较大 K 值泛化能力强但性能可能降低。需要根据具体问题与数据集选择最佳的 K 值,以达到模型准确率与泛化能力的最优平衡。总体来说,K 值直接控制了 KNN 算法的复杂度,其选择对模型性能有重大影响。

下面是我们实验的 K 值情况：

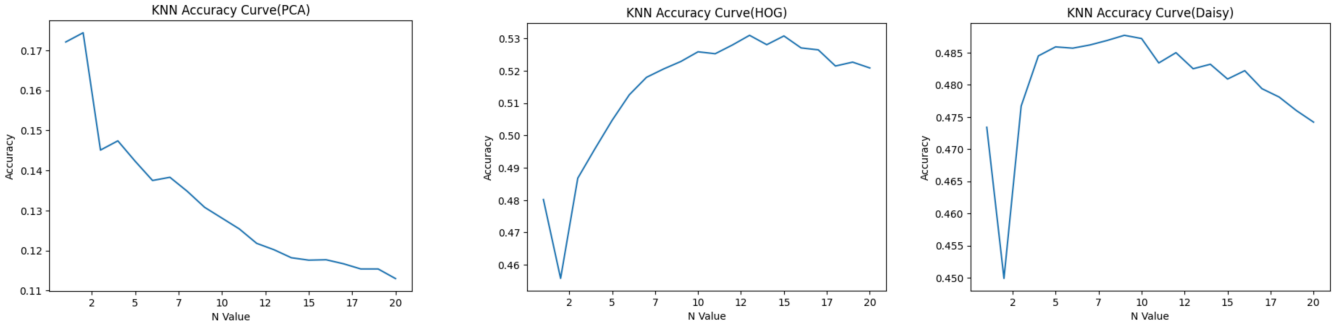

图 3 实验KNN的K值对三种预处理结果的ACC比对

D. 随机森林算法

随机森林是一种基于集成学习理论的分类与回归方法。它由多棵决策树组成,每棵决策树依据从原始训练集中通过有放回的随机采样得到的子样本进行训练。随机森林同时具有 bagging 的集成优势以及由特征随机采样引入的额外随机性。

决策树的最大深度影响着每棵决策树的复杂程度。树深过浅,模型欠拟合,无法充分学习数据模式;树深过深,每棵树都容易过拟合。通常来说,适当限制决策树的最大深度,可以避免单棵树过拟合,控制随机森林的整体复杂度。

一般建议随机森林中决策树的深度取值为数据特征数量的平方根附近。最大深度不宜过大或过小。适当控制单棵决策树的深度,既能保持较好的泛化能力,又不会丧失学习数据的功能。多棵树深度适中的集成,通常能够给出更稳健和准确的模型。

所以选择适当的决策树深度对随机森林性能有重要影响。需要在避免过拟合与保持学习能力间取得平衡。这取决于数据的复杂程度以及特征空间的大小。

随机森林算法参数变换之后的实验效果如下：

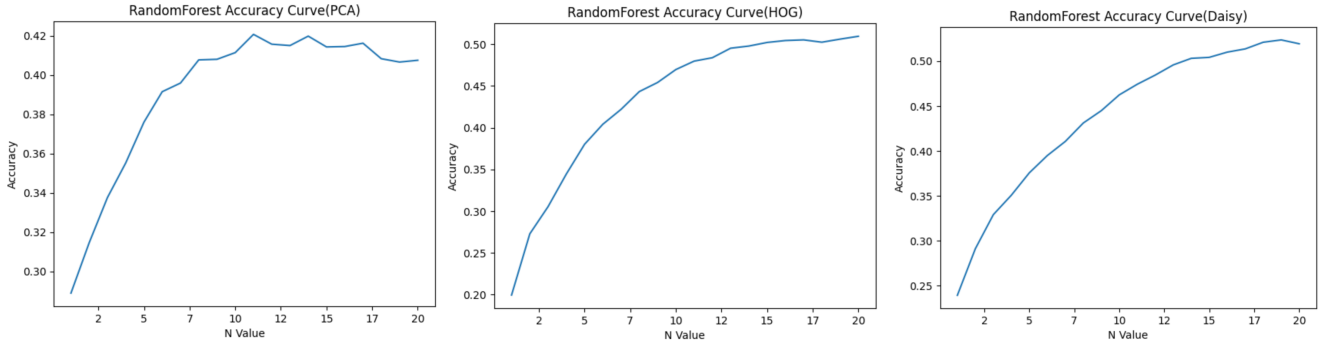

图 4 随机森林树深度在三种预处理下的比较

E. 实验结果

各种方法的最终 ACC 和预处理方法的不同呈现出了一些不同的情况，详细的数据参看下表。

	PCA	HOG	Daisy
BernoulliNB	0.3296	0.2216	0.1000
GaussianNB	0.2917	0.4537	0.3839
RandomForest	0.3834	0.5105	0.5229
KNN	0.1456	0.5047	0.4859
Logistic	0.4029	0.5095	0.3604
LinearSVM	0.3395	0.5047	0.4117
SVM	0.4862	0.6298	0.6427
DecisionTree	0.2393	0.2741	0.3123
XGBoost	0.4939	0.5826	0.6049
AdaBoost	0.3419	0.378	0.3648

我们的实验最终得到的结果输出参考下图：

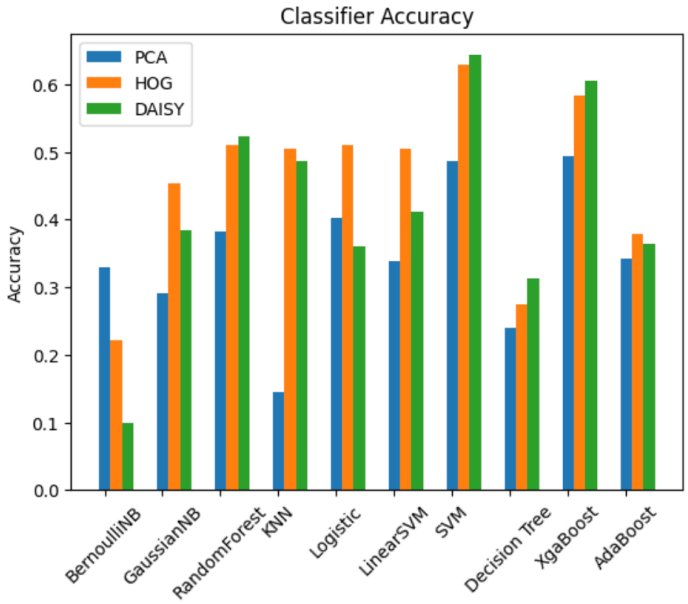

图 5 机器学习效果图

IV.深度学习方法

为与机器学习方法进行对比，此处采用深度学习方法对 Cifar10 数据集分类问题进行探索。我们主要实现了在图像分类领域比较经典且表现较好的深度学习模型，包括 CNN,AlexNet，LeNet,ResNet,VGG 以及采用 Transformer 架构的 ViT 等。我们记录了模型训练过程中的准确率和损失随训练轮数的变化曲线，以混淆矩阵的形式展示了图片分类结果，并对 CNN 的不同卷积层输出做可解释性分析。

A. CNN

卷积神经网络（CNN）专门设计用于处理具有网格状结构的数据，其主要组成部分包括卷积层、激活函数、池化层、全连接层和输出层。卷积层通过卷积操作提取输入数据的局部特征，激活函数引入非线性，池化层降低空间维度，全连接层将卷积层输出转化为最终预测，而输出层负责最终的分类或预测。

在图像分类中，CNN 取得了显著的成功。通过学习图像中的局部特征，如边缘、纹理等，CNN 能够高效地对复杂模式进行学习和识别。其卷积和池化操作使其对平移和变形具有不变性，提高了对输入数据的鲁棒性。

首先我们在 Cifar10 数据集上的 10 个类别 50000 张测试数据上进行训练，其准确率及损失随训练轮数变化如下图所示：

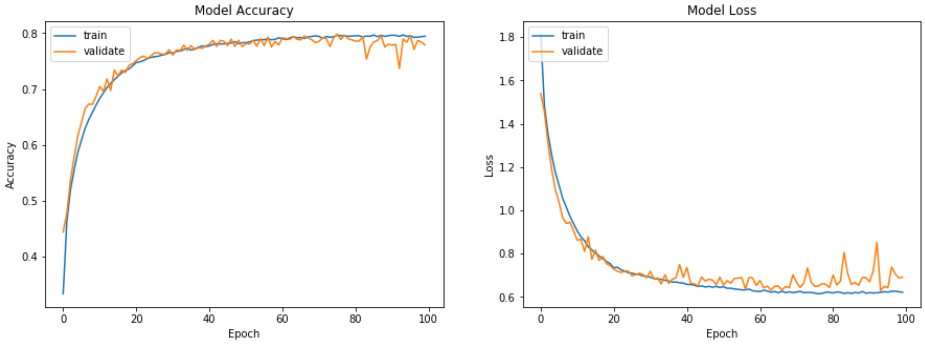

图 6 准确率与损失随训练轮数变化曲线

最终 CNN 模型训练准确率稳定在 0.794 左右，验证集上准确率稳定在 0.69 左右，模型没有过拟合，作为 baseline 模型表现稳定。同时为探索 CNN 模型的可解释性，我们对图片在 CNN 模型中变化过程做可解释性分析，如下图所示。

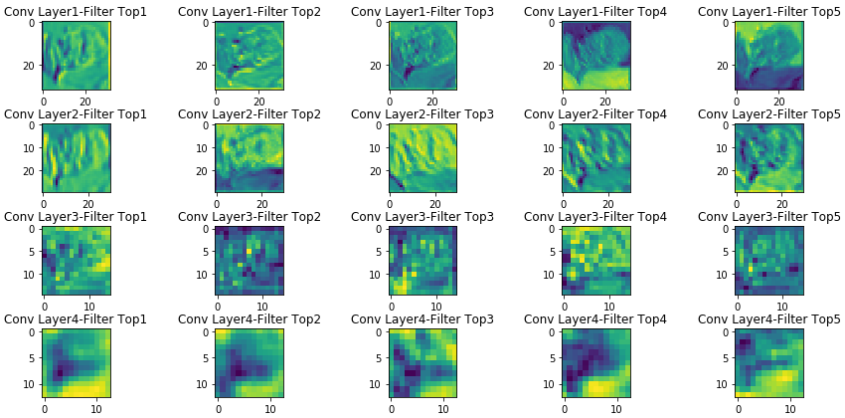

图 7 CNN 可解释性分析

其中从上到下为深度逐渐加深的卷积层输出可视化图像，每一行当前卷积层中前五个最显著的通道特征。可以注意到在底层卷积更接近于原始数据，可以观察到物体大致轮廓，而高层更关注物体的细节信息，通过对局部的分析对最后的决策产生贡献。对分类结果我们采用混淆矩阵进行展示，可以观察到，混淆矩阵对角线上分布的样本数最多，其中 Truck 类别和 Ship 类别的分类效果最佳。而将 Cat 类错误分类为 Dog 类和将 Airplane 类错误分类为 Ship 类相对较多。

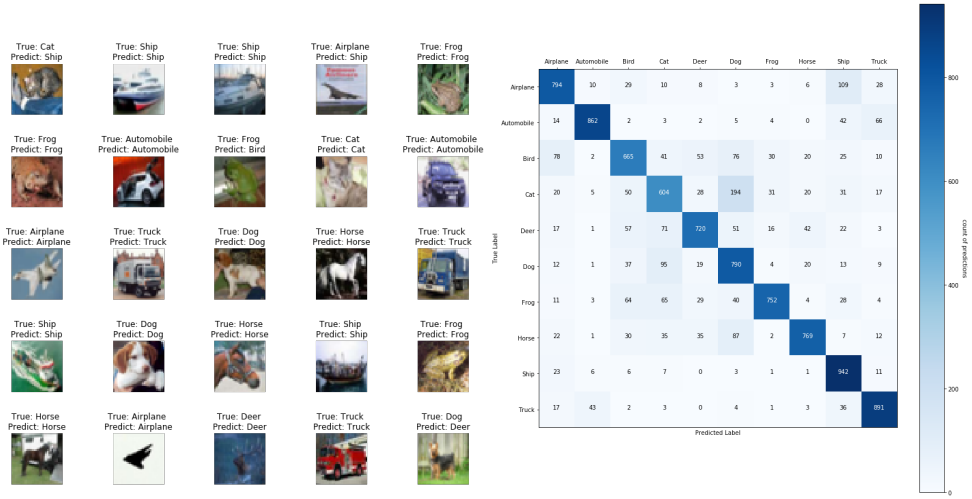

图 8 CNN 分类结果及混淆矩阵分析

B. AlexNet

AlexNet 是一种深度卷积神经网络，由五个卷积层和三个全连接层组成。它在 2012 年 ImageNet 图像分类挑战中取得了重大突破，采用较大的卷积核和 ReLU 激活函数，引入 Dropout 来防止过拟合，下图为 AlexNet 网络结构。

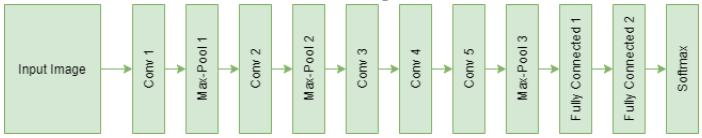

图 9 AlexNet 网络结构

我们同样对此网络进行训练与测试，其中准确率和损失变化如下图所示。可以较为明显的看出，在大约 20 轮以后，训练集和验证集的准确率出现较大差异，产生了明显的过拟合现象，在 loss 曲线可以观察到，此后验证集 loss 不断上升，此处如果引入 early stop 机制或修改目标函数，引入正则项等能够有效减少过拟合的发生。

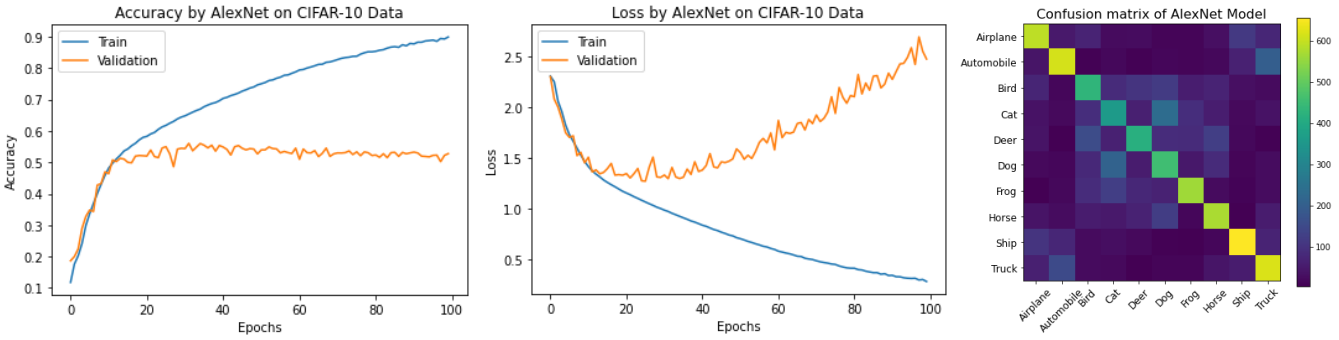

图 10 AlexNet 训练曲线与混淆矩阵

同理我们以混淆矩阵的方式展示分类结果如上图所示，可以观察到其分类结果与 CNN 相似，在 Ship 和 Truck 类别上的分类效果最好，且仍然存在 Cat 和 Dog 分类错误相对较多的情况。

C. VGG

VGG（Visual Geometry Group）是深度卷积神经网络模型，旨在通过使用深层卷积层来提高图像分类性能。VGG 模型的主要特点是采用相对较小的卷积核（3x3），通过多个卷积层和池化层的堆叠，形成 16 或 19 层的深度结构。VGG-16 是比较常见的 VGG 模型之一，其具有 16 个卷积和全连接层，其中包括 13 个卷积层和 3 个全连接层。这个模型的卷积层均使用 3x3 的小卷积核，每两个卷积层之间通过最大池化层降低空间分辨率。其结构示意图如下所示：

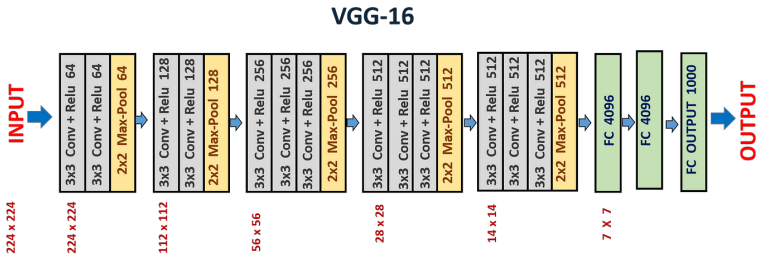

图 11 VGG-16 网络结构

在训练过程中收敛速度相对传统模型较快，同时在准确率方面有显著提升，训练过程如下图所示：

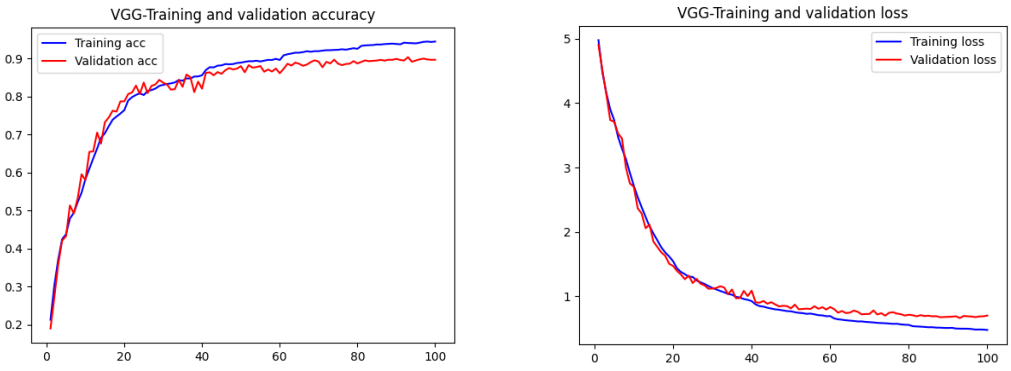

图 12 VGG 训练曲线

其中没有观察到明显的过拟合现象，模型约 40 轮后收敛。训练准确度稳定在 0.944，验证准确度稳定在 0.89 左右。训练过程中我们引入了动态学习率机制，模型学习率会随着训练轮数的提高而逐渐变小，由此根据模型的表现自适应地调整学习率，以加速收敛、提高性能，并避免陷入局部最优解。在我们的模型中，动态学习率定义为：

$$\mathrm{lr} := \mathrm{lr} \times 2^{-(\lfloor \mathrm{epoch} / \mathrm{lr\_drop} \rfloor)}$$

其中 lr 表示学习率，epoch 表示当前训练轮数，lr_drop 为控制学习率下降速率的超参数。

D. ResNet

ResNet 其创新之处在于引入了残差块，通过跨层的跳跃连接，允许信息直接传递而不经过多层堆叠的卷积层，解决了训练极深网络时的梯度消失和梯度爆炸问题。ResNet 的结构使得可以轻松地训练数百层的网络。ResNet 沿用了 VGG 完整的 卷积层设计。 残差块里首先有 2 个有相同输出通道数的卷积层。 每个卷积层后接一个批量规范化层和 ReLU 激活函数。 然后我们通过跨层数据通路，跳过这 2 个卷积运算，将输入直接加在最后的 ReLU 激活函数前。其中较为关键的残差块结构如下图示：

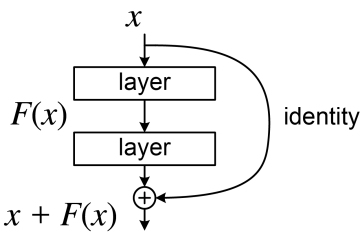

图 13 残差块结构

而整体网络由卷积层和残差块组成，在实现时采用了 19 层 ResNet，每个残差快由两个或三个卷积层和激活函数组成，在每个残差块中 tensor 通过卷积层、激活函数和池化层进行处理。最后通过平均池化层和全连接层完成下游的分类工作。

我们对 ResNet 网络进行训练，得到训练过程准确率变化和下图所示。ResNet 同样发生了快速收敛且过拟合的情况。但是相对于图 9 中 AlexNet 训练过程，在模型过拟合后其 loss 没有继续上升。

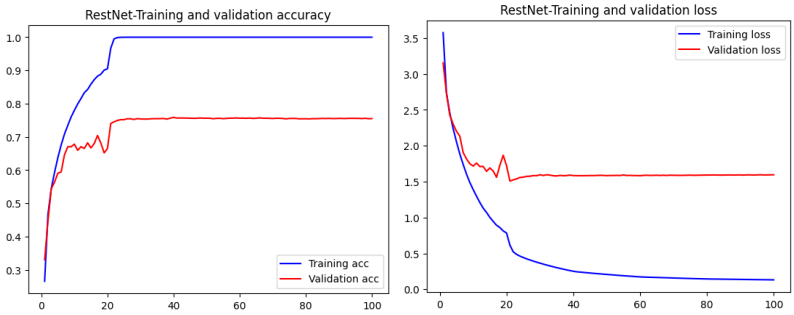

图 14 ResNet 训练过程

E. EfficientNet

EfficientNet 是卷积神经网络架构，其结合了网络深度、宽度和分辨率的缩放，通过使用复合系数来平衡这三个方面。复合系数包括了网络的深度、宽度和分辨率的乘法因子，使得在不同规模的网络中都能够取得较好的性能。其设计理念是通过自动调整网络结构的深度、宽度和分辨率，使得模型在资源有限的情况下能够更好地适应不同的任务。该网络使用了一种叫做“自动网络结构搜索” 的方法，通过对不同网络结构的组合进行评估，找到了一组优秀的复合系数。其网络结构如图所示：

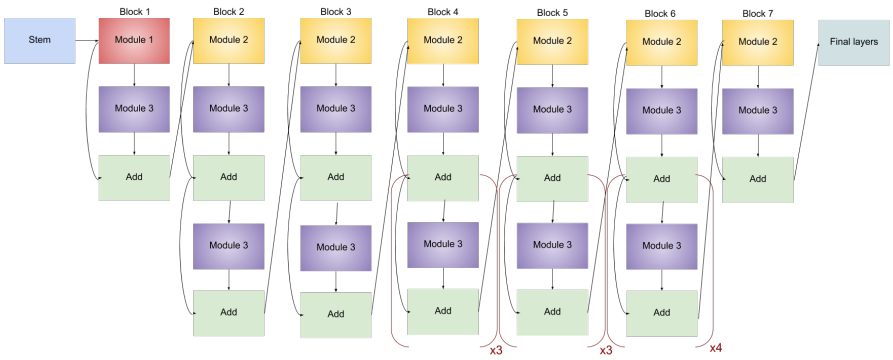

图 15 EfficientNet 网络结构

同时在训练 Efficient 过程中我们考虑到存在模型过拟合的情况，设置了 Early Stop 机制，实际在模型运行过程中确实出现了由于过拟合或其他因素导致准确率在最近的几个轮次中准确率持续下降，由此机制的存在，模型训练终止

在 13epoch，早于预先设置的 20epochs，有效防止模型过拟合情况的出现。该模型在训练集上的准确率可以达到 0.994，在验证集上的准确率可以达到 0.975

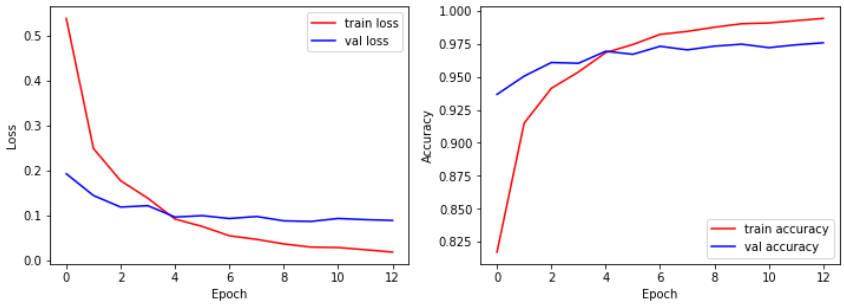

图 16 EfficientNet 训练（Early Stop）

F. Vision Transformer

VisionTransformer是一种基于Transformer架构的图像分类模型，与传统卷积神经网络不同，ViT将图像分割成一系列小块，然后将每个小块的像素值转换为序列形式，以便Transformer可以处理。ViT的核心思想是将图像信息转化为序列数据，然后通过Transformer模型来学习全局关系。ViT的基本结构包括一个嵌入层，用于将图像块的像素转换为向量序列，以及一系列Transformer块，用于学习图像中的全局特征关系。每个Transformer块包含多头自注意力机制和全连接前馈网络，允许模型在序列中捕捉不同位置的关系。其结构如下图所示：

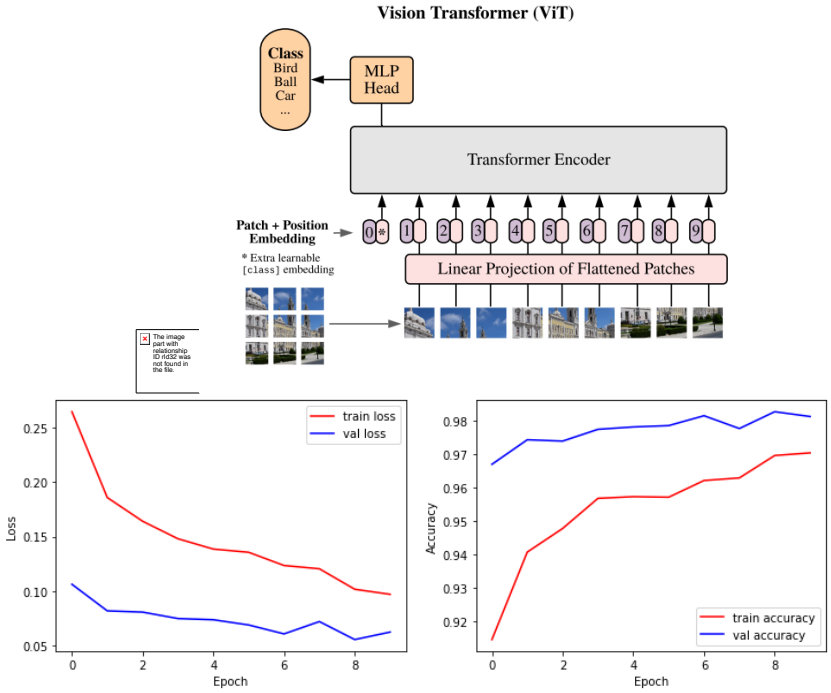

图 17 ViT 模型结构与训练过程

ViT 在 Cifar10 测试集上的表现是尝试过的模型中最好的，其准确率达到了 0.981，训练集准确率稍逊于 EfficientNet，但仍保持在较高的 0.9705.