GoogleNet 在 Fashion_MNIST 数据集上的应用

1. 研究背景和介绍

1.1 关于 GoogleNet 的简单介绍

GoogleNet（Inception V1）是由 Google 提出的深度卷积神经网络架构，于 2014 年发布。其特点是采用了“Inception 模块”来提高网络宽度和深度，以减少参数数量，并且能够更有效地学习多层抽象特征。

1.2 实验目的

本次实验旨在探索和应用 GoogleNet 模型在 Fashion_MNIST 数据集上进行图像分类任务，并分析模型在该数据集上的性能表现。

2. 数据集

2.1 数据集介绍

Fashion MNIST 是一个包含 10 个类别的图像数据集，每个类别对应一种服饰或配件。数据集中的图像尺寸为 28x28 像素，适用于图像分类任务。

2.2 数据预处理

加载数据集后，将数据集划分为训练集、验证集和测试集。通过对图像进行标准化和预处理，使其适应 GoogleNet 模型的输入要求。

```
#调整输入数据的维度
train_images_rgb = np.repeat(train_images, 3, axis=3)  # 复制灰度通道，创建 RGB 图片
test_images_rgb = np.repeat(test_images, 3, axis=3)

train_images_resized = tf.image.resize(train_images_rgb, [224, 224])
test_images_resized = tf.image.resize(test_images_rgb, [224, 224])
```

3. 模型架构

3.1 GoogleNet 模型结构

GoogleNet 采用了 Inception 模块，其中包含多个并行卷积层和池化层，通过不同尺寸的卷积核来捕获多尺度的特征。模型具有较深的层级结构，利用全局平均池化和多个全连接层来进行分类。

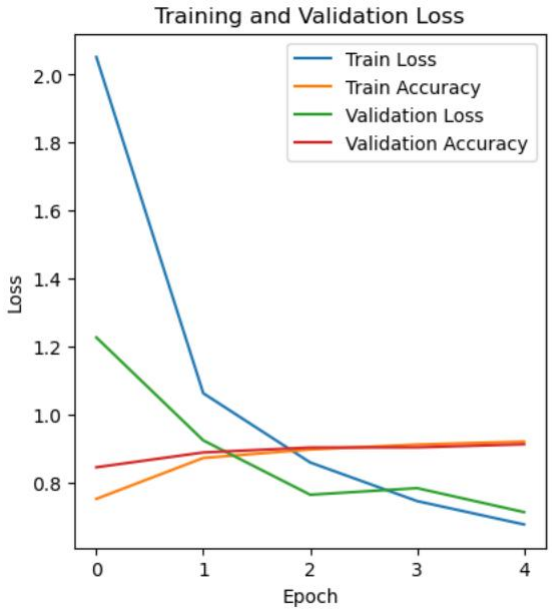
下面是模型各个层的参数。详细的代码见附件 GoogLeNet.ipynb

Layer (type)	Output Shape	Param #	Connected to
Inputs (InputLayer)	[(None, 224, 224, 3)]	0	[]
convolution-1 (Conv2D)	(None, 112, 112, 64)	9472	['Inputs[0][0]']
max-pool-1 (MaxPooling2D)	(None, 56, 56, 64)	0	['convolution-1[0][0]']
convolution-2 (Conv2D)	(None, 56, 56, 192)	110784	['max-pool-1[0][0]']
max-pool-2 (MaxPooling2D)	(None, 28, 28, 192)	0	['convolution-2[0][0]']
inception-3a (Inception)	(None, 28, 28, 256)	163696	['max-pool-2[0][0]']
inception-3b (Inception)	(None, 28, 28, 480)	388736	['inception-3a[0][0]']
max-pool-3 (MaxPooling2D)	(None, 14, 14, 480)	0	['inception-3b[0][0]']
inception-4a (Inception)	(None, 14, 14, 512)	376176	['max-pool-3[0][0]']
inception-4b (Inception)	(None, 14, 14, 512)	449160	['inception-4a[0][0]']
inception-4c (Inception)	(None, 14, 14, 512)	510104	['inception-4b[0][0]']
inception-4d (Inception)	(None, 14, 14, 528)	605376	['inception-4c[0][0]']
inception-4e (Inception)	(None, 14, 14, 832)	868352	['inception-4d[0][0]']
max-pool-4 (MaxPooling2D)	(None, 7, 7, 832)	0	['inception-4e[0][0]']
inception-5a (Inception)	(None, 7, 7, 832)	1043456	['max-pool-4[0][0]']
inception-5b (Inception)	(None, 7, 7, 1024)	1444080	['inception-5a[0][0]']
avg-pool (AveragePooling2D)	(None, 1, 1, 1024)	0	['inception-5b[0][0]']
flatten (Flatten)	(None, 1024)	0	['avg-pool[0][0]']
dropout (Dropout)	(None, 1024)	0	['flatten[0][0]']
linear (Dense)	(None, 1000)	1025000	['dropout[0][0]']
AuxiliaryClassifier-1 (AuxiliaryClassifier)	(None, 1000)	3188840	['inception-4a[0][0]']
AuxiliaryClassifier-2 (AuxiliaryClassifier)	(None, 1000)	3190888	['inception-4d[0][0]']
softmax (Softmax)	(None, 1000)	0	['linear[0][0]']

Total params: 13374120 (51.02 MB)
Trainable params: 13374120 (51.02 MB)
Non-trainable params: 0 (0.00 Byte)

4. 实验结果

下图是 GoogleNet 模型在 Fashion MNIST 数据集上的训练过程中的训练损失、准确率随 epoch 的变化。



下图是模型在测试集上的表现

```
313/313 [==============================] - 229s 733ms/step - loss: 0.7130 - AuxiliaryClassifier-1_loss: 0.2389 - AuxiliaryClassifier-2_lo
ss: 0.2355 - softmax_loss: 0.2385 - AuxiliaryClassifier-1_accuracy: 0.9120 - AuxiliaryClassifier-2_accuracy: 0.9150 - softmax_accuracy:
0.9127
Test accuracy: 0.23892414569854736
```

6. 分析和讨论

模型表现：

模型在训练集上表现得非常好，但在测试集上的性能并不理想。GoogleNet 拥有强大的特征提取能力，但在 Fashion MNIST 这样相对简单的数据集上，其过度复杂的模型架构会导致训练过程中出现过拟合，导致模型的泛化能力比较差。

由于 GoogleNet 模型复杂，其收敛速度可能较慢。在小型数据集上，需要更多的训练时间才能获得可接受的性能。

相对于数据集规模，GoogleNet 模型过于复杂，会消耗过多的计算资源和内存。这在小型数据集上并不是一个高效的选择。

综上所述，尽管 GoogleNet 在大型数据集上表现出色，但在 Fashion MNIST 这样的小型数据集上，其性能可能不如其他简单的模型，比如之前的 LeNet5。

实验结果分析：

模型在训练集和验证集上表现良好，但在测试集上准确率较低。这表明模型在训练过程中出现了过拟合。改进空间在于减少模型复杂度或引入正则化方法来降低过拟合。

损失函数分析：观察训练集和验证集上的损失函数变化。验证集上的损失函数在后两个 epoch 上高于训练集，表明了模型过拟合的情况。可以考虑使用更适当的正则化方法或减少模型的复杂度。

7. 结论

GoogleNet 在训练集和验证集上表现出较高的准确率，但在测试集上的性能不佳。这表明模型在训练过程中出现了过拟合现象，无法很好地泛化到未见过的数据。这是因为模型在 Fashion MNIST 这样相对简单的数据集上可能过于复杂，导致过拟合。训练过程中，模型可能过度学习了训练集上的特定特征和噪声，而泛化能力不足。可以考虑引入正则化方法，如 L1、L2 正则化、Dropout 等，以减少过拟合。这有助于提高模型的泛化能力。

总体来说，GoogleNet 在 Fashion MNIST 上的性能可能不如其他更简单的模型。未来的工作方向可以着重于降低模型复杂度、增强模型的泛化能力，并尝试其他更适合小型数据集的模型架构和训练方法。