

持续学习大作业实验报告

小组成员

郁博文	201250070	负责代码编写和调试
邢佳勇	201250071	负责论文阅读
陈子凡	201250072	负责代码编写和调试
郑启睿	201250229	负责撰写实验报告

复现思路

我们复现的是iCaRL算法

初始化

首先，iCaRL需要一个初始的神经网络模型（如卷积神经网络）作为基础分类器。这个模型可以在一个基准数据集上进行训练，以学习初始的分类能力。

分类器选择

从训练集中选择一小部分样本（称为参考集），该集合包含每个类别的一些样本。参考集用于训练一个初始分类器，并为每个类别维护一个原型向量。

增量训练

接下来，iCaRL开始逐步接收新的样本数据。对于每个新的类别，它会将一些样本添加到参考集中，并使用参考集重新训练分类器。同时，它还会为新类别创建一个原型向量，该向量是该类别参考集样本的平均。

知识蒸馏

在每次增量训练后，iCaRL会对分类器进行知识蒸馏。这意味着它会使用所有先前训练过的分类器进行预测，并将它们的预测结果与真实标签进行比较。然后，根据比较结果调整分类器的权重，以提高对已知类别的分类准确性。具体来说，在每次before_task中，我们会将上一个iteration的模型进行深度拷贝，之后，再使用旧模型的fc层去更新模型的参数。

```
def update_representation(self):
    if self.old_model is not None:
        self.model.fc = nn.Linear(self.model.fc.in_features, self.n_classes)
        self.model.fc.weight.data[:self.old_model.fc.weight.size(0)] = self.old_model.fc.weight.data.clone()
        self.model.fc.bias.data[:self.old_model.fc.bias.size(0)] = self.old_model.fc.bias.data.clone()
```

遗忘旧类别

为了避免旧类别的累积性能下降，iCaRL使用一种称为“保留最重要样本”的策略来忘记旧类别。这个策略基于参考集的重要性进行排序，并保留每个类别的前几个样本。然后，iCaRL通过从参考集中移除不再重要的样本来删除旧类别。

```
def reduce_exemplar_set(self, m):
    for class_idx in self.exemplar_sets.keys():
        self.exemplar_sets[class_idx] = self.exemplar_sets[class_idx][:m]
```

重放

为了进一步改善遗忘性能，iCarl还使用了一种重放机制。它将之前训练过的样本添加到当前训练集中，以帮助分类器记住以前的类别。这样可以减少对旧数据的遗忘。

Early Stop

在实验中，我们发现，如果训练的epoch较小可能会发生欠拟合，使得在加入新的类别后模型容易遗忘先前学到的分类知识，但是如果epoch较大则会发生过拟合。在查阅相关资料后，我们使用了earlystop的策略。即，我们让每个iteration训练20个epoch，但是如果连续数个epoch训练结束后，test的loss都没有下降，。则停止当前iteration进行下一个。考虑到要使模型尽可能记住当前学到的知识，我们设定的耐心值是5。

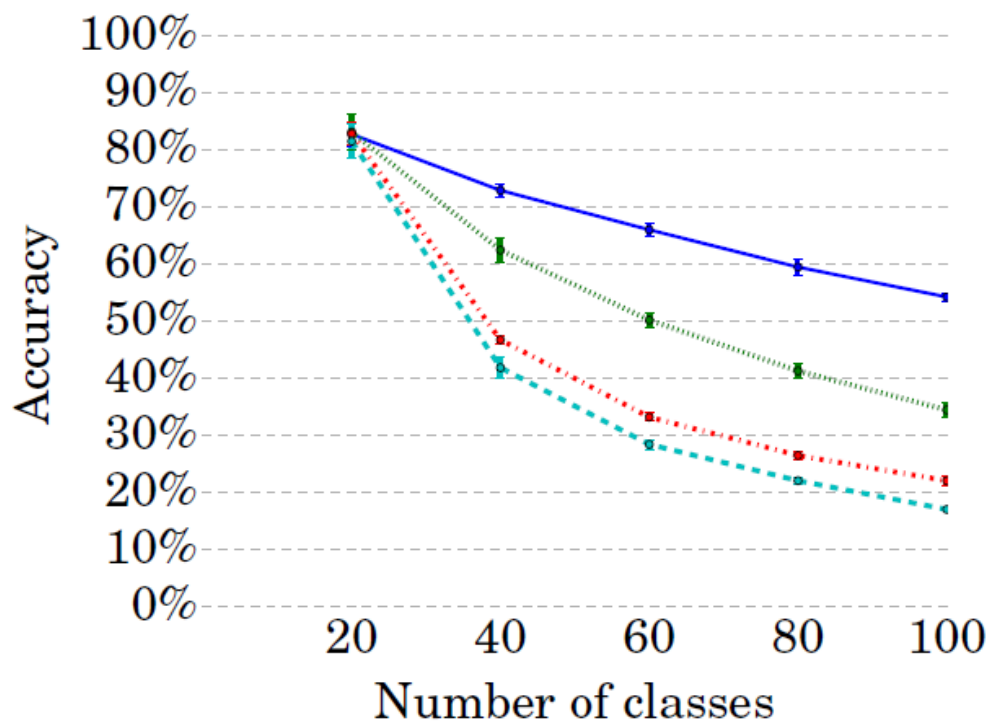
```
if eval_dict['avg_loss'] < best_loss:
    best_loss = eval_dict['avg_loss']
    patience = 0
else:
    patience += 1
if patience >= 5:
    break
```

复现结果

我们使用的数据集是CIFAR100.首先每次加入20个新的类别，得到的实验结果如下

Task0	71.3%
Task1	54.27%
Task2	39.32%
Task3	31.51%
Task4	28.31%

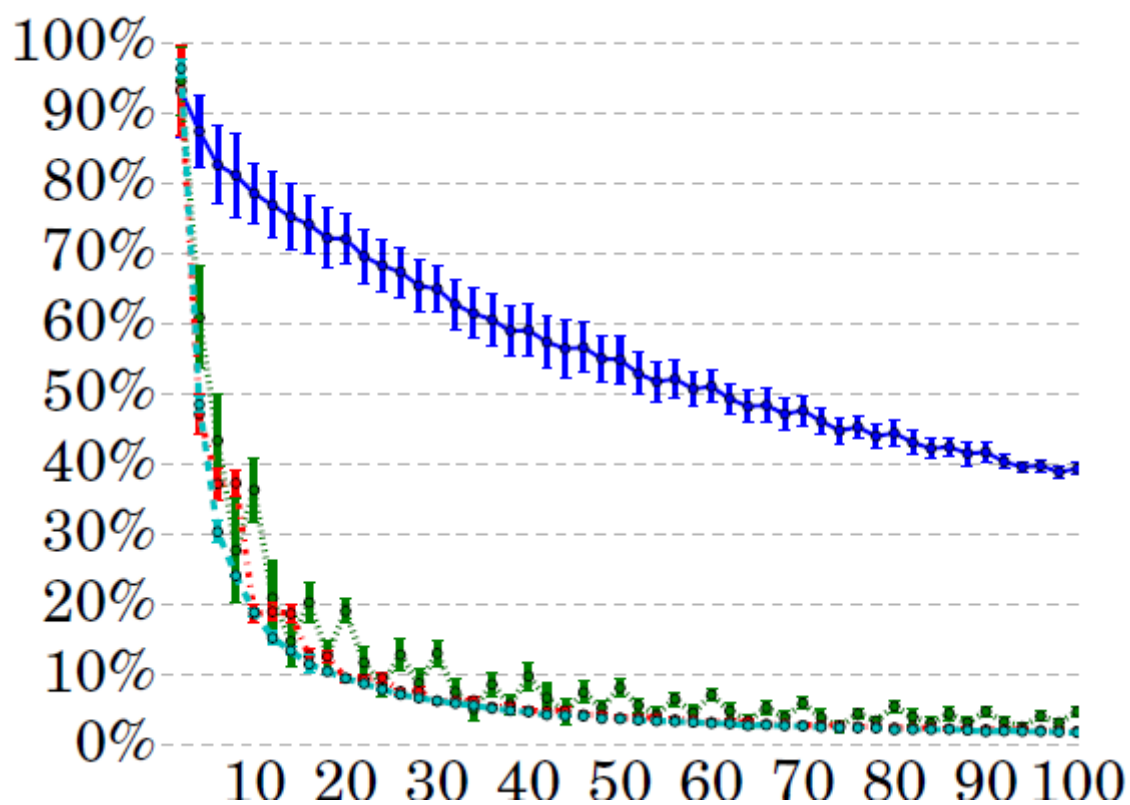
对比原论文的结果



我们又尝试了每次加入10个新的类别，得到的实验结果如下：

Task0	70.1%
Task1	51.65%
Task2	45.8%
Task3	39.97%
Task4	31.68%
Task5	29.83%
Task6	25.57%
Task7	21.49%
Task8	21.6%
Task9	18.51%

对比原文结果如下：



总体上，我们的效果比原文差了很多，可能的原因是，我们对于蒸馏模型的代码理解较为幼稚，在更新模型以往参数时出现问题。