持续学习策略

手工实现：（已实现3个+基础的策略）

1. Naïve

Naïve策略对于训练数据的选择没有任何的偏好，将训练数据打乱后顺序喂给分类网络进行10分类任务（在MNIST数据集上）

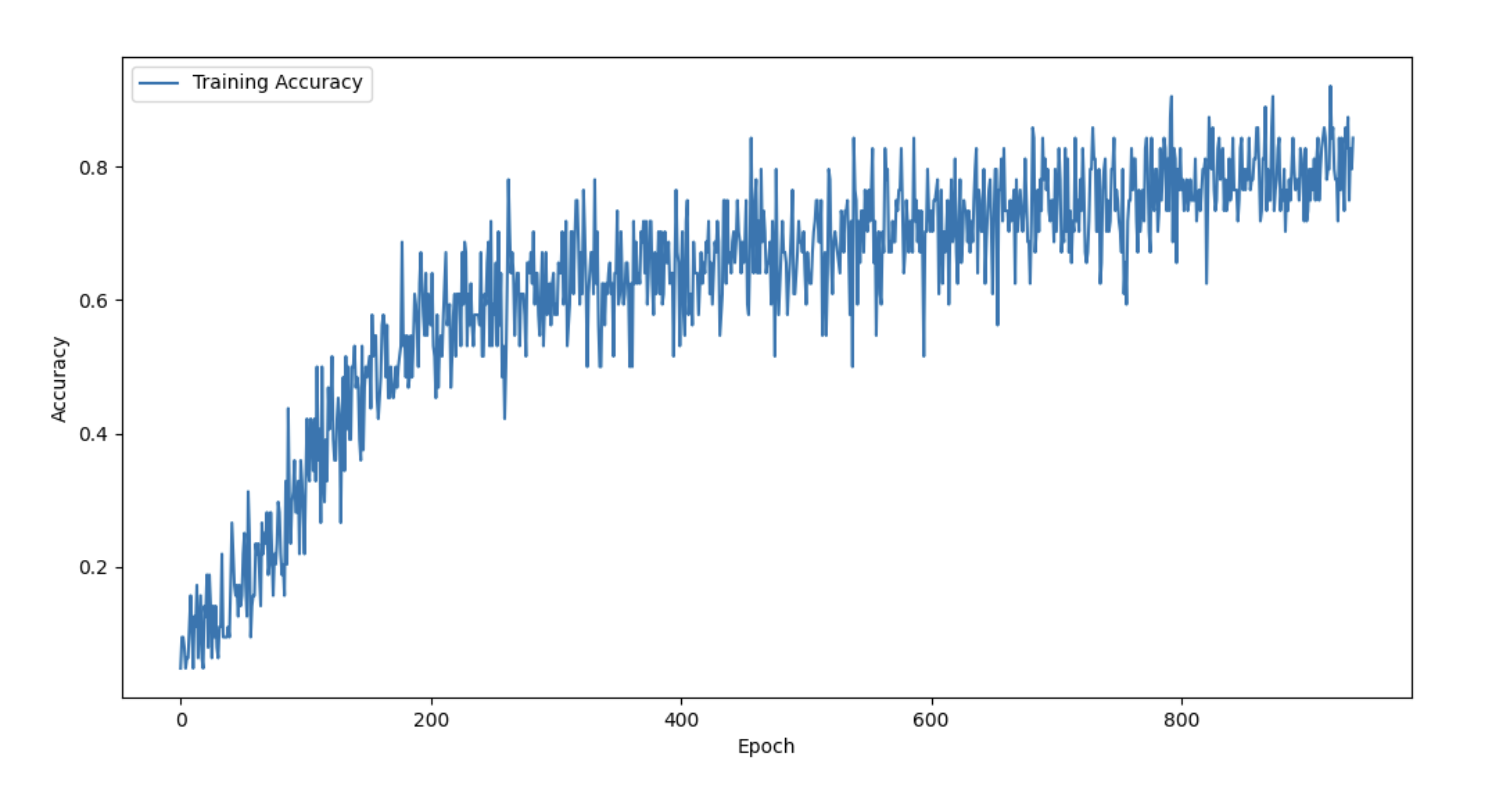
图表, 折线图

描述已自动生成

1. A-GEM

全称为：Averaged Gradient Episodic Memory (A-GEM)

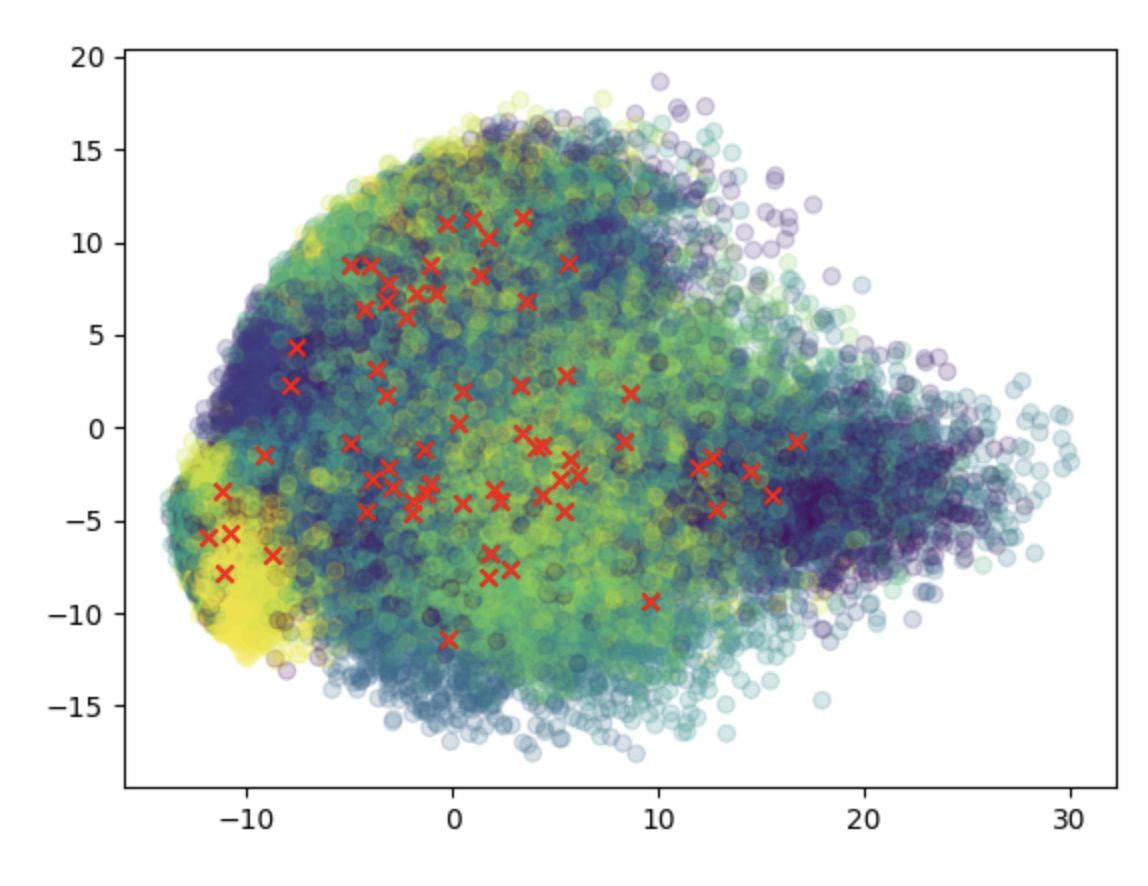
用当前训练的数据来更新episodic存储器，在每次训练的时候，不直接采用计算的gradient来更新网络参数，转而是通过当前的gradient还有episodic存储器里的数据计算到的gradient来平均之后共同更新参数，这样在训练的过程中，不仅会适应当前的数据，还可以避免之前学习到的参数的遗忘。



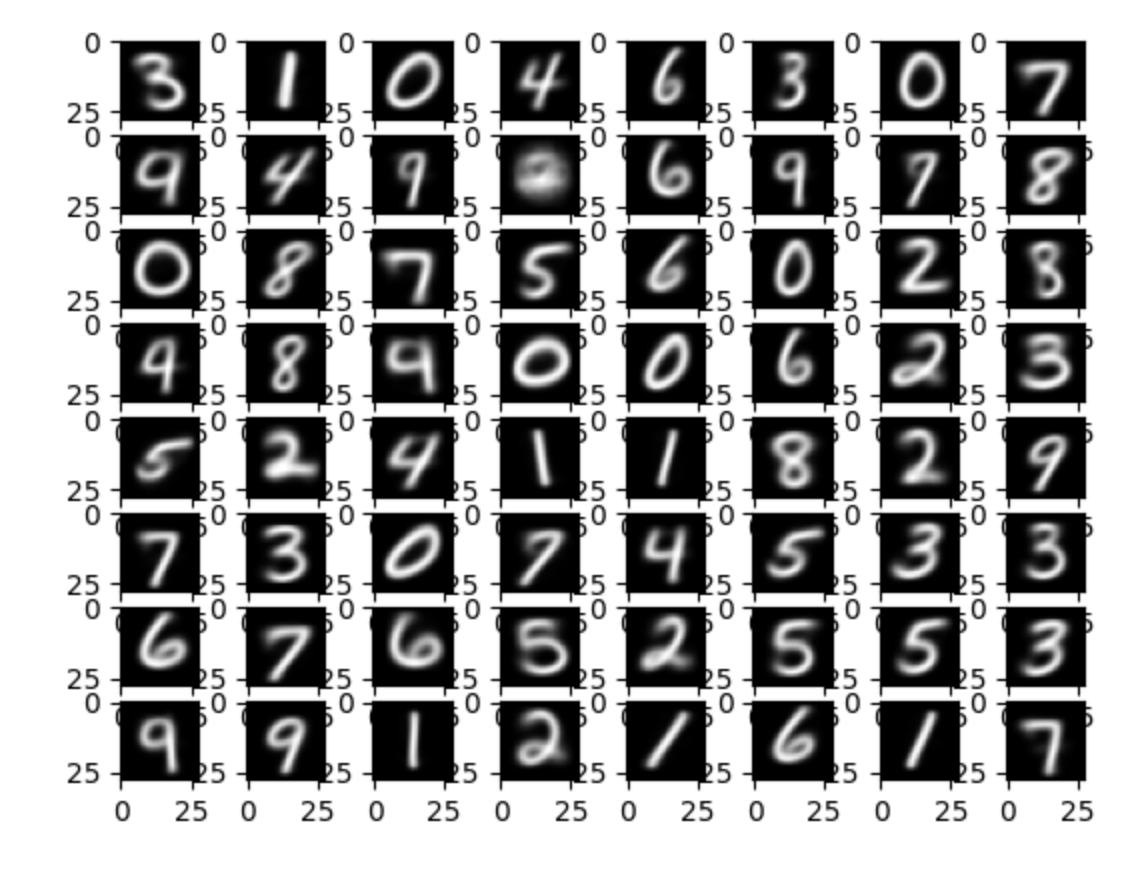
1. Replay

采用kmeans聚类算法对数据集聚出来64个类，将这64个类中心存储起来。

聚类展示：



聚类中心展示：



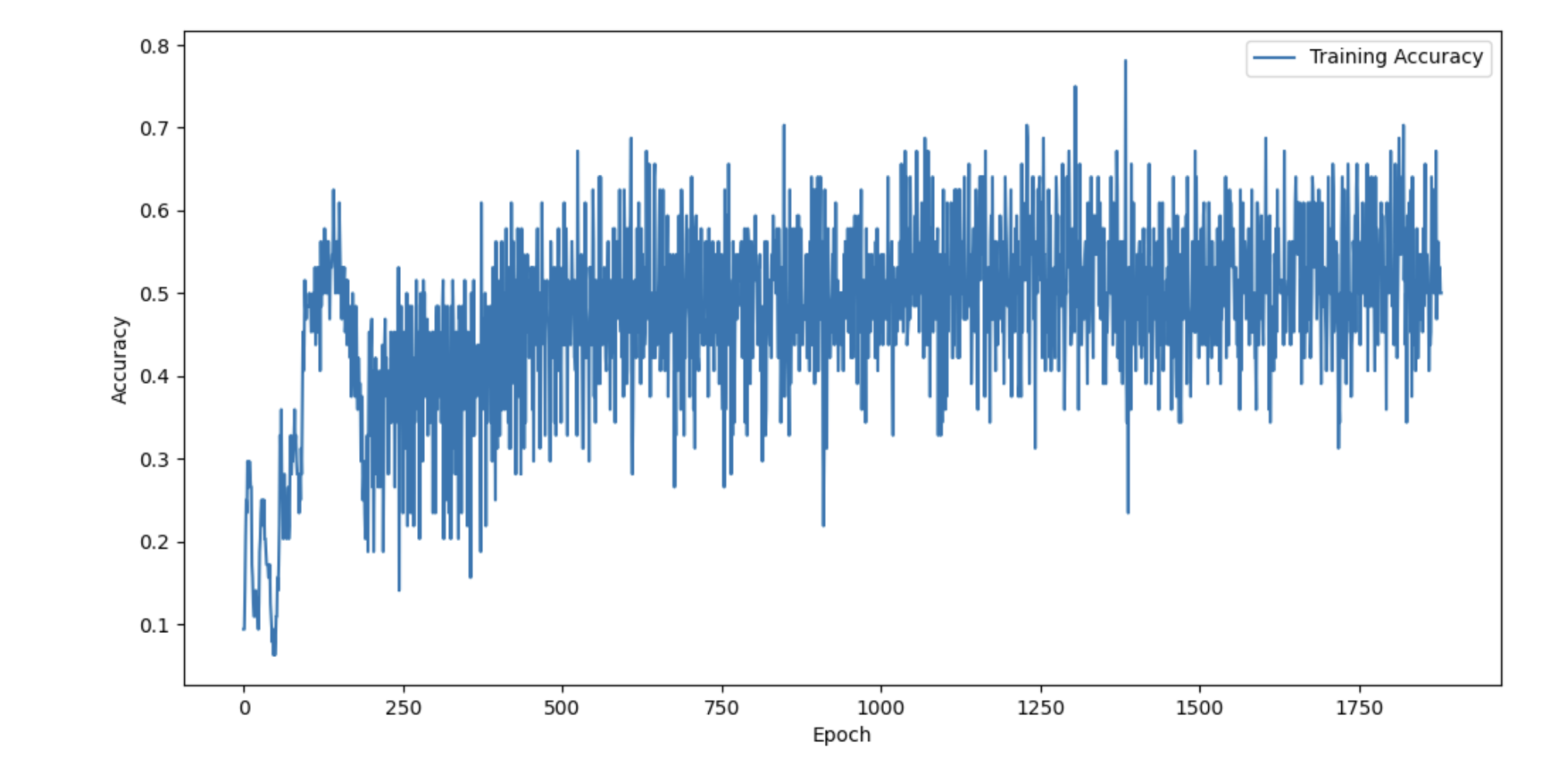
每次从数据集中挑出来32个数据样本，以及从64个类中心随机选择出32个，共同组成64个训练数据喂给网络进行学习。

图表, 条形图

描述已自动生成

1. Entropy

该策略将会选择信息熵高的样本点来作为训练数据训练十分类器，信息熵越高越说明不确定度越大，也就是分类器对于该点的预测越不稳定。选择每次信息熵最高的前64个样本点作为训练数据喂给网络进行训练。



其余手工实现代码：

问题：复现难度大，有两种实现方式1）从现有框架中剥离出来，但是框架集成度高，不太能顺利；2）手动实现，需要对照原始论文，原始论文并未开源。

使用框架：

1. Avalanche

问题：数据读取直接使用python内建的dataloader，并且包装成ncscenario，无法获取策略读取数据的index

1. PyCIL

问题：框架集成度太高，需要解构框架

1. GMvandeVen/continual-learning

问题：同上