大作业

总体目标：把 MNIST 训练集（mnist_train）划分为 train_dataset 和 val_dataset，train_dataset 用于训练 CNN，val_dataset 用于选择 CNN 的超参数，最后在 MNIST 测试集（mnist_test）上测试。

条件：对于 train_dataset，只允许使用下列图片的标签，其他图片的标签不允许使用。允许使用标签的图片的索引（train_dataset 里的图片索引）为：[1173, 3336, 12529, 12785, 12979, 17351, 27048, 40579, 43128, 46498]，即对于每个类别，允许使用的有标签的图片个数为 1。

挑战：如果只使用上述 10 个图片及其标签训练 CNN，显然会导致过拟合（over-fitting），需要尽可能地使用更多的数据。

关键问题：对于给定的每个类别的 1 个图片，在 train_dataset 里尽可能地找出与其相似的图片（即属于同一个簇）。

思路：构建基于自编码机（或者 PCA）的聚类模型，在嵌入空间中找出和上述给定图片属于同一个簇的图片，把它们的标签设为给定图片的标签。另一个挑战是，聚类算法并不能 100%聚类正确。对于每一个簇，可以使用置信度（比如 DEC 中的每个图片属于每个簇的概率或在嵌入空间中使用高斯混合模型聚类，使用其γ_{ji}）>threshold 的图片（threshold 自己设定）或者每个簇中离簇中心距离排名前 60%（threshold，自己设定）的图片或者其他方式（这部分是开放性的）。最后，把给定的有标签的 10 个图片，以及与它们属于同一簇并且置信度>threshold 或者离簇中心距离排名>threshold 的图片作为训练集训练 CNN。

技巧：自编码机的编码器和解码器可以都是 CNN，上述聚类任务训练完的编码器可以作为分类任务的预训练模型。如果采用 PCA 降维，需要从头训练 CNN。选择什么样的 CNN（比如和前一个项目一样）也是开放问题。

把 mnist_train 数据集划分为 train_dataset 和 val_dataset 时，不同的 seed 会导致不同的划分。为了统一划分方式，使得上述给定的图片的索引在所有机器上一致，提供如下代码:

```
seed = 42
def seed_torch(seed=0):
    random.seed(seed)
    os.environ['PYTHONHASHSEED'] = str(seed)  # 为了禁止hash随机化，使得实验可复现
    np.random.seed(seed)
    torch.manual_seed(seed)
    torch.cuda.manual_seed(seed)
    torch.cuda.manual_seed_all(seed)  # if you are using multi-GPU.
    torch.backends.cudnn.benchmark = False
    torch.backends.cudnn.deterministic = True

seed_torch(seed)
transform = transforms.Compose([
    transforms.Resize(32),
    transforms.ToTensor(),
])

mnist_train = MNIST(root='./', download=True, transform=transform)
mnist_test = MNIST(root='./', train=False, download=True, transform=transform)
# split train/val subset
generator = torch.Generator().manual_seed(seed)
train_dataset, val_dataset = random_split(mnist_train, [50000, 10000], generator=generator)
train_loader = DataLoader(train_dataset, batch_size=batch_size, shuffle=True, num_workers=8, persistent_workers=True)
val_loader = DataLoader(val_dataset, batch_size=batch_size, shuffle=False)
```

提交代码和报告至：yeweiysh@qq.com，截止日期为：2023/7/2 23:59:59.