Efficient Pseudo Labelling and Generative Replay for Continual Incremental Learning

Program Instructions

Link: https://github.com/srirammandalika/R2R

1. Prerequisites

Ensure you have the following installed:

- Python 3.8+
- PyTorch (>= 2.0.0)
- CUDA (if using GPU acceleration)
- Additional dependencies (install via pip):

Command:

```
pip install numpy scikit-learn matplotlib torchvision diffusers transformers datasets tqdm
```

2. Data Preparation

The framework supports:

- CIFAR-10
- CIFAR-100
- SVHN
- CINIC-10
- TinyImageNet

Automatic Download (if supported)

Command:

```
python download_datasets.py --dataset <dataset_name>
```

3. Training the R2R Model

Basic Training Command:

```
python train_r2r.py \
  --dataset cifar10 \
  --num_tasks 5 \
  --batch_size 32 \
  --epochs 100 \
  --lr 0.001 \
  --gamma 0.9 \
  --use_gpu
```

Arguments:

- --dataset: Dataset name (cifar10, cifar100, etc.).
- --num_tasks: Number of continual learning tasks (default: 5).
- --batch_size: Batch size (default: 32).
- --epochs: Training epochs per task (default: 100)
- --lr: Learning rate (default: 0.001).
- --gamma: Decay rate for learning rate (default: 0.9).
- --use_gpu: Enable GPU training (optional)

4. Evaluation & Testing

After training, evaluate the model on all seen tasks:

```
python evaluate.py \
  --dataset cifar10 \
  --model_path ./saved_models/checkpoints.pth \
  --task_id 5
```

4. Expected Outputs

Training Logs: In terminal
Cluster Visualization (t-SNE plots):
Saved in ./results/tsne_plots/.

Generative Replay Samples:
Stored in ./results/generated_samples/.