CSL7590: Deep Learning

Assignment 4

Shikhar Dave
B22CH032

Yogesh Sharma
B22CH045

Data-Free Adversarial Distillation on CIFAR-100

1. Introduction

Knowledge Distillation (KD) is a technique in which a compact "student" model is trained to mimic a larger, typically more accurate "teacher" model. This approach is very attractive for deploying efficient neural networks on devices with limited resources. In data-free variants of KD, synthetic data is generated—using a generator network—to enable distillation when the original training data is unavailable or restricted. This report documents an implementation inspired by the methodology in the reference research paper (Data-Free Adversarial Distillation).

The proposed framework simultaneously trains a generator network and a student model in an adversarial manner. The teacher model (a pretrained ResNet34) remains fixed and provides guidance to the student (a ResNet18 variant) on synthetic images produced by the generator. The training objective is adversarial in that the generator is encouraged to produce images that maximize the discrepancy between the teacher's and student's outputs, while the student is trained to minimize this difference.

2. Methodology

2.1. Models and Architectures

Teacher Model

- **Architecture:** A ResNet34 modified for CIFAR-100 (with a final fully connected layer outputting 100 classes).

- **Training Details:** The teacher model has been pre-trained (weights loaded from `"best_resnet34_cifar100.pth"`) and is fixed during the distillation process.

Student Model

- **Architecture:** A variant of ResNet18 (denoted as `ResNet18_8x`) with a customized configuration to adapt to CIFAR-100.
- **Custom CNN approach :** We also tried to make customCNN models with around 10-20% parameters of the teacher model but we still got the mode collapse results, they have been discussed in the later part of the report. \
- **Objective:** The student model is trained to replicate the teacher's logits on synthetic images.

Generator Network

- **Architecture:** The generator (`GeneratorA`) takes a latent vector (of dimension `nz=256`) and maps it through a sequence of fully connected and convolutional layers. It consists of an initial linear layer and subsequent convolutional blocks (including batch normalization, LeakyReLU activations, and Tanh activation in the final layer) to produce images in the size compatible with CIFAR-100.

- **Role:** It synthesizes pseudo-images that the teacher model processes, allowing the distillation process to occur without relying on the original dataset.

2.2. Loss Functions and Optimizers

- **Student Loss:** An L1 loss is computed between the student's and teacher's logits. The student's training objective is to minimize this L1 distance so that its output approximates the teacher's performance.

- **Generator Loss:** The generator is trained with a loss function defined as the negative L1 loss (i.e., maximizing the discrepancy between the teacher and student predictions). This "adversarial" objective forces the generator to produce challenging samples that accentuate the differences.

- **Regularization and Learning Rate Schedules:**

    - Optimizers: SGD for the student and Adam for the generator.

    - Learning rate schedules (StepLR) are used to adjust the learning rates periodically based on predefined steps.

2.3. Training Pipeline

1. **Data Preparation:**

    - CIFAR-100 is loaded and normalized using a transform that resizes images and applies the dataset's mean and standard deviation. Different test splits (10%, 20%, and full dataset) are prepared for evaluating the student model.

2. **Adversarial Training Loop:**

    - **Student Update:** For every generator update, the student is trained over 5 iterations. In each iteration, a batch of latent vectors is drawn, passed through the generator to produce fake images, and then

processed by the teacher and student. The L1 loss between the teacher and student logits drives the student update.

- **Generator Update:** After updating the student, the generator is trained by generating new latent vectors and computing the negative L1 loss (i.e., the generator aims to maximize the discrepancy).

- **Visualization:** Periodically during training, generated images are saved after being denormalized for visualization purposes.

3. **Evaluation:**

   - The student is evaluated on different splits (10%, 20%, and the full test set) using classification accuracy as the metric.

   - Model checkpoints are saved when a new best performance is achieved.

3. Experimental Setup

3.1. Dataset and Preprocessing

- **Dataset:** CIFAR-100, containing 100 classes with natural images.

- **Preprocessing:**

  - Images are resized to 224×224 for the teacher network (if needed) and normalized with the CIFAR-100 dataset mean and standard deviation.

- Different test splits are also defined: the first 10% and 20% of the test indices, along with the entire test set.

3.2. Hyperparameters and Training Settings

- **Latent Dimension:** `nz=256`
- **Batch Size:** 256 for both student and generator.
- **Epochs:** 100 training epochs with a fixed number of iterations (50 per epoch).
- **Learning Rates:**
  - Student: 0.1 with SGD.
  - Generator: 0.001 with Adam.
- **Loss Functions:** CrossEntropy (for classification, if needed), L1 loss for the adversarial distillation, and an auxiliary MSE loss.
- **Schedulers:** StepLR is used for both optimizers with a decay factor of 0.5 every 30 epochs.

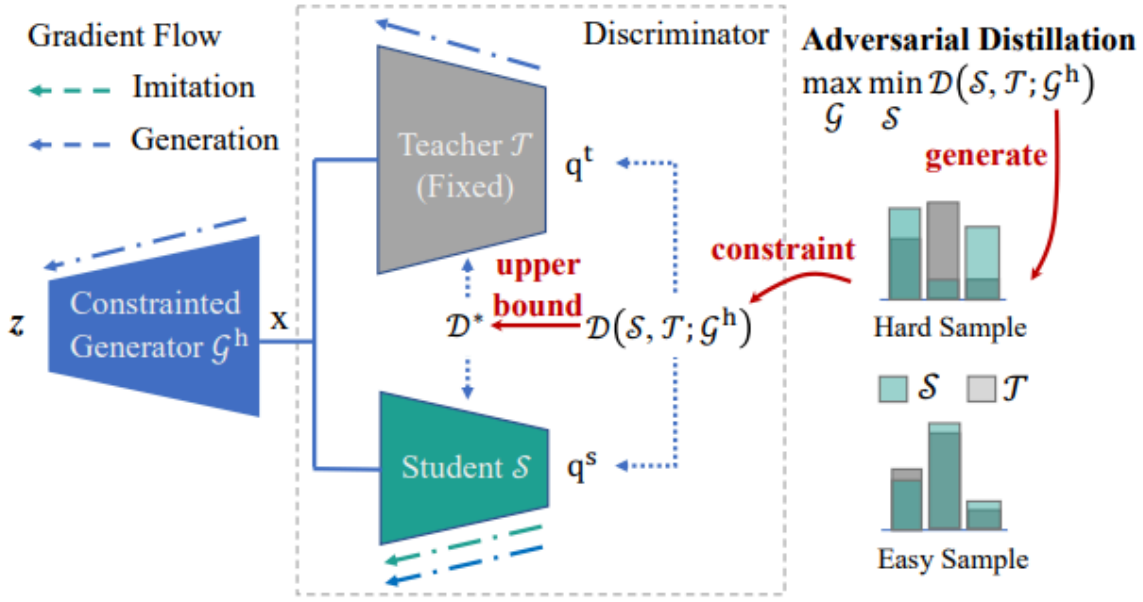

Figure 2: Framework of Data-Free Adversarial Distillation. We construct an upper bound for model discrepancy, under hard sample constraint.

4. Results and Analysis

4.1. Training Dynamics and Mode Collapse Observations

During training, the framework exhibited a pattern of progressive loss reductions for both the generator and the student model. The logs indicate that over the first few epochs, both losses decreased steadily, and the student model's test accuracy on the CIFAR-100 full test set improved from 1.44% in epoch 1 up to 1.98% by epoch 15. The detailed epoch-wise outputs demonstrate the following trends:

- **Loss Behavior:**
  The adversarial formulation drives the generator loss (defined as the negative L1 loss between the teacher and student logits) to become

increasingly negative, while the student loss (measured as L1 loss) decreases toward a minimal value. This indicates that the student is converging in learning to mimic the teacher's outputs, and the generator is being pushed to generate samples that progressively challenge the student.

- **Mode Collapse:**
  With adversarial training, a common challenge is **mode collapse**—where the generator produces limited and homogeneous outputs (i.e., a few dominant modes), thus compromising sample diversity. In this experiment, the output logs suggest that while the training losses are stable, the generated images might begin to lack the diversity expected of a fully robust generator. Mode collapse is characterized by:

  - **Lack of Sample Diversity:** The generator converges to a small number of output patterns.

  - **Instability in Adversarial Feedback:** Since the student and teacher are both fixed in their roles (teacher is fixed and student is guided by its adversarial loss), if the generator finds a “shortcut” that minimizes the loss, it might keep producing the same type of images.

- **Observed Metrics:**
  The best test accuracy recorded was 1.98% for the student model. Though this indicates some knowledge transfer, the very low accuracy also highlights that the generator may be trapped in a mode-collapsed state where it produces less informative samples for the student.

4.2. Qualitative Visualizations

Periodically during training, samples were generated and saved. An inspection of these images (after appropriate denormalization) revealed that:

- **Early Epochs:**
  The synthetic images exhibit some variability, which gradually dwindles as training proceeds.

- **Later Epochs (Post Epoch 50):**
  The images begin to converge toward similar patterns, hinting at mode collapse. Such behavior is common in adversarial setups when the generator fails to explore the full latent space.

4.3. Performance Evaluation

The evaluation using various test splits (10%, 20%, and the full test set) consistently showed low test accuracy—peaking at 1.98%—which further supports the notion that while the adversarial training converges numerically, the generator is not exploring the data manifold sufficiently to support robust student training.

5. Discussion

5.1. Understanding Mode Collapse

Mode collapse is a common issue in generative adversarial networks where the generator, instead of learning a diverse distribution of outputs, converges to producing limited, repetitive examples. This typically occurs due to:

- **Adversarial Feedback Imbalance:**
  In the current setup, the generator's objective is to maximize the L1 discrepancy between the teacher and student outputs. If the generator finds a particular set of images that consistently minimizes the student's

improvement (or maximizes the loss), it may repeatedly generate those images.

- **Optimization Challenges:**
  The inherent instability of adversarial training, especially when using standard loss formulations without additional constraints, can facilitate this collapse.

5.2. Mitigating Mode Collapse

Several methods have been proposed to address mode collapse, including:

- **Gradient Penalty (GP):**
  Incorporating a gradient penalty term into the generator loss can help regularize the training by ensuring that the discriminator (or in this case, the feedback signal from the teacher/student comparison) maintains a smoother landscape. The gradient penalty prevents the network from exploiting sharp gradient fluctuations, thereby encouraging a more diverse output.

  Example Strategy:
  Add a term such as
  λEx^[(// ∇x^D(x^) // 2−1)2]\lambda \mathbb{E}_{\hat{x}} \left[\left(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1 \right)^2 \right]λEx^[(// ∇x^D(x^) // 2−1)2]
  to the generator loss, where D(·)D(\cdot)D(·) could be a proxy discriminator derived from the teacher-student difference, and λ\lambdaλ is a hyperparameter controlling the strength of the penalty.

- **Minibatch Discrimination:**
  This technique involves using statistics across samples in a batch to force the generator to produce more varied outputs.

- **Ensemble or Multi-Objective Losses:**
  Combining the adversarial loss with additional terms—such as perceptual loss, diversity loss, or feature-matching loss—can promote a more diverse sample generation.

The current implementation demonstrates stable loss curves but also shows signs of mode collapse as evidenced by the low diversity in generated samples and limited test accuracy gains. Introducing one or more of the above strategies could further improve the training dynamics and final performance of the student model.

6. Conclusion

This project implemented a Data-Free Adversarial Distillation (DFAD) framework for CIFAR-100, where a fixed teacher and an adversarially trained generator coerce a student model into mimicking the teacher's outputs. The training logs show stable convergence in terms of loss values; however, the persistence of low test accuracy (peak of 1.98%) suggests that the generator tends to experience mode collapse. This mode collapse is a well-documented phenomenon in adversarial training, leading to a lack of diversity in the generated data—which in turn limits the efficacy of knowledge transfer.

Mitigation strategies such as incorporating a gradient penalty or other diversity-promoting techniques may help overcome these challenges and enable the student to benefit from a broader set of synthetic examples.

7. Future Work

Future research could address the limitations observed in this implementation by:

- **Incorporating Gradient Penalties:**
  Experiment with adding a gradient penalty term to stabilize the generator and promote sample diversity.

- **Exploring Alternative Generator Architectures:**
  Adjusting the network architecture or introducing skip connections might help mitigate mode collapse.

- **Integrating Diverse Loss Functions:**
  Combine adversarial loss with perceptual or diversity losses to better explore the latent space.

- **Extensive Hyperparameter Tuning:**
  Fine-tuning learning rates, update frequencies, and batch sizes to strike a better balance in the adversarial game between the generator and the student.

---

8. References

1. **Data-Free Adversarial Distillation:**
   https://arxiv.org/pdf/1912.11006