PDL Challenge 0: Investigating ResNet18 Performance on Heterogeneous CIFAR-10 Dataset

SR+ITL

March 6, 2024

Abstract

The CIFAR-10 dataset is a well-known benchmark in the field of machine learning, consisting of 60,000 32x32 color images in 10 different classes, with 6,000 images per class. However, real-world datasets often exhibit imbalances and heterogeneity in class distributions, which can significantly affect the performance of deep learning models. Your task is to investigate how the performance of a ResNet10 model varies over the CIFAR-10 dataset when the dataset is modified to have heterogeneous distributions of classes. You will create datasets with different proportions of images for each class and record the change in accuracy, especially on a per-class basis.

1 Problem Setting

In the realm of machine learning, the assumption of a balanced dataset often remains an ideal rather than a reality. In many real-world applications, data is inherently imbalanced, with some classes significantly underrepresented compared to others. This imbalance poses a significant challenge to the robustness and fairness of machine learning models, especially in the context of deep learning where large amounts of data are typically required for optimal performance.

The CIFAR-10 dataset, a staple in the machine learning community for benchmarking image classification algorithms, presents an opportunity to explore this challenge. Under normal circumstances, this dataset is perfectly balanced, with equal representation of its ten diverse classes. However, this ideal balance is rarely encountered in practical scenarios. The impact of data imbalance on model performance is a critical area of study, particularly for convolutional neural networks (CNNs) which are widely used in image classification tasks.

In this project, you will embark on a journey to investigate how the performance of a CNN, specifically a ResNet18 model, is influenced by varying degrees of data imbalance in the CIFAR-10 dataset. You will start by establishing a baseline, training and evaluating the ResNet18 on the original, balanced CIFAR-10 dataset to understand its performance under standard conditions. Following this, you will delve into the core of the challenge by creating and experimenting with different versions of the CIFAR-10 dataset, each exhibiting a distinct level of class imbalance. This exploration will enable you to observe and analyze the resilience of the ResNet18 model as it confronts incThe final phase of your exploration will involve a strategic intervention. You will modify the loss function used during the training process, implementing a scheme designed to mitigate the adverse effects of data imbalance.

This challenge is not just about coding and data manipulation. It is an opportunity to develop a deeper understanding of the nuances and challenges inherent in machine learning, particularly in dealing with imbalanced datasets, which are ubiquitous in real-world scenarios. Your findings and insights will contribute to the broader conversation about building robust and fair machine learning models, capable of performing well across a variety of data landscapes.

2 Problem statement

Step 1: Load and Preprocess CIFAR-10 Dataset:

- prepare a loader for the CIFAR-10 dataset in its original, balanced form.

- let N_k be the total number of elements of elements for the k-th class, for $k \in [1...10]$
- Apply necessary preprocessing to the dataset: for instance normalize the pixel intensity level to be in the range $[0, 1]$, instead of $[0, 255]$
- Let the loader function take in input a vector

$$\mathbf{N} = [N_1, N_2, \ldots, N_{10}], \tag{1}$$

  which specifies how many elements are loaded for each class
- Split the data in train and test data, proportially for each class according to $\mathbf{N}$.

Step 2: Define ResNet18 Model:

- Utilize the ResNet18 model, which can be readily imported from PyTorch's model zoo.
- Make sure you understand the standard optimizer setting such as loss, learning rate, and momentum.
- Make sure you understand how the weights of the network are initialized and how the random seed for generating these weights is controlled.

Step 3: Train and Evaluate the Model:

- Train the ResNet18 model on the CIFAR-10 training dataset for a 20 epochs for .

$$\mathbf{N} = [4000, 4000, ..., 4000] \tag{2}$$

  that is the setting in which each class has 4000 entries in the dataset.
- Evaluate the model over the 25 iterations and repeat the training process 10 times for different random initializations of the Resnet18 weights.
- A curve plotting the train accuracy at each iteration as averaged over the 10 repetitions of the training process for different random weights at initialization.
- Plot the train and test accuracy as above and verify that the model is not over-fitting
- Make sure that you save the models once trained so you can use them for the next step. After that make sure you free your memory.

Step 4: Investigate the per-class accuracy

Given the ten trained models from step 2, evaluate the accuracy of the models for the 10 classes. That is evaluate the loss function over data points from the same class.

plot the pre-class accuracy for the 10 classes as averaged over the ten trained models

Step 5: Investigate the Impact of Dataset Imbalance

- Generate ten versions of the data loaders in which, for the k version, we have $N_k = 6000$ while $N_j = 3777$ for $j \neq k$.
- for each of the datasets, train ten models using ten different random initializations
- for each of the trained models, evaluate the accuracy of the ten classes
- comment on how the imbalance in the dataset
- DO NOT ATTEMPT TO SAVE ALL THE MODELS: collect the relevant statistics and then free your computer memory

Step 6: Find an effective way to illustrate the effect of data heterogeneity

- Try to clearly illustrate and summarize the results of the simulation campaign above
- Experiment with different representations of the results to provide a comprehensive understanding of the model's performance. Some suggested visualization methods include:
  - **Radar Plots:** Useful for displaying multivariate data in a way that is easy to understand. They can be particularly effective for comparing the per-class accuracies of different models.
  - **Confusion Matrices:** To visually represent the model's predictions versus the actual labels, highlighting where the model performs well and where it struggles.
  - **Precision-Recall Curves:** Especially useful in the context of imbalanced datasets, these curves can help in assessing the trade-offs between precision and recall for different classes.
  - **Heatmaps:** For representing data matrices where coloring the cells can provide an immediate visual summary of the information, such as the variation in accuracy across different classes and datasets.

  Encourage creativity and exploration in visualizing the data, as this can lead to deeper insights and understanding of the model's performance under varying conditions.

Step 7: Mitigate Data Heterogeneity with Loss Function Modification

- For this part, you are free to modify any aspect of the training – loss function, batch composition, batch scheduling, model updates.... so as to minimize the impact of the data heterogeneity for a certain imbalanced setting
- Find an effective way to demonstrate the performance improvement with respect to the baseline.