Robust Training with Noisy Labels using Normalized Losses and the APL Framework

S Anantha Krishnan

April 7, 2025

Abstract

Robustness in machine learning is crucial when working with real-world datasets prone to noisy labels, which can lead to overfitting and poor model performance. This project explores the effectiveness of normalized loss functions and the Active-Passive Loss (APL) framework in enhancing robustness to label noise. Experiments on the CIFAR-10 dataset demonstrate that APL strikes a balance between robustness and accuracy, even at high noise rates.

1 Introduction

Noise in data labels is a common issue in real-world datasets, significantly impacting model performance. Robust learning aims to handle such noisy labels effectively while maintaining good performance on clean data. However, robustness often comes at the cost of generalization. This work investigates loss normalization and the APL framework to address this trade-off.

2 Methodology

2.1 Data Preparation

We use the CIFAR-10 dataset and introduce symmetric noise at rates $\eta \in \{0.2, 0.4, 0.6, 0.8\}$. The labels are flipped randomly within each class to simulate noisy real-world data. For each sample, with probability η, its label is uniformly flipped to one of the remaining $C-1$ classes. The noise is applied only to the training set, while the validation/test sets remain clean.

2.2 Model Architecture

I use a lightweight convolutional neural network (CNN) implemented in PyTorch, with the following structure:

- **Feature Extractor:**
  - 3 blocks of 2 convolutional layers each (6 conv total)
  - Kernel size: 3×3 with padding=1
  - Channels: $64 \rightarrow 128 \rightarrow 256$
  - Batch normalization and ReLU after each convolution
  - MaxPool2d (2×2) after each block

- **Classifier:**
  - Flattened features: 256×4×4 = 4096 dimensions
  - Fully-connected layers: 4096 → 1024 → 10
  - Dropout (p=0.5) after first FC layer
- **Parameters:**
  - Input: 3×32×32 (CIFAR-10 images)
  - Output: 10-class probabilities
  - Total parameters: ∼4.2 million
- **Implementation Details:**
  - Initialized weights using Kaiming normal initialization
  - Trained on GPU using CUDA acceleration

2.3 Normalized Losses

Normalized Cross-Entropy (NCE) and Normalized Focal Loss (NFL) are used. These losses map the output within a fixed range to reduce sensitivity to outliers. In datasets with large noises incorret labled samples cause large losses, so if loss function is not normalised, these large values dominate optimization causing the model to focus on outliers instead of learning the general pattern. What normalization does is it scales down the loss making the gradients smaller. Here is the formula of NCE and NFL the two loss functions used in this paper:

$$\mathrm{NCE}(p) = \frac{-\frac{1}{N}\sum_{i=1}^{N}\log p(y_i \mid x_i)}{-\sum_{c=1}^{C} p(c)\log p(c)}$$

$$\mathrm{NFL}(p) = \frac{-\frac{1}{N}\sum_{i=1}^{N}(1-p(y_i))^{\gamma}\log p(y_i)}{\frac{1}{N}\sum_{i=1}^{N}(1-p(y_i))^{\gamma}}$$

Explanation of Terms:

- **N**: The number of samples in the batch.
- **C**: The total number of classes.
- $\mathbf{p(y_i \mid x_i)}$: The predicted probability for the true label y_i given the input x_i. This is obtained by applying a softmax function to the logits.
- $\mathbf{p(c)}$: The empirical probability (frequency) of class c in the current batch. It is computed as the mean of the one-hot encoded true labels.
- **Numerator of NCE:** $-\frac{1}{N}\sum_{i=1}^{N}\log p(y_i \mid x_i)$ represents the average negative log-likelihood (NLL) of the true labels.
- **Denominator of NCE:** $-\sum_{c=1}^{C} p(c)\log p(c)$ is the entropy of the empirical class distribution. Normalizing by this value scales the loss between 0 and 1.

- γ: The focusing parameter in the focal loss, which emphasizes hard-to-classify samples by modulating the weight $(1 - p(y_i))^\gamma$.
- **Numerator of NFL:** $-\frac{1}{N}\sum_{i=1}^{N}(1 - p(y_i))^\gamma \log p(y_i)$ computes the average weighted negative log-likelihood, where the weights reduce the loss contribution of easy examples.
- **Denominator of NFL:** $\frac{1}{N}\sum_{i=1}^{N}(1 - p(y_i))^\gamma$ is the normalization factor, ensuring the loss is properly scaled by the average weight over the batch.

2.4 APL Framework

APL combines:

- **Active losses:**
  - Focus on maximizing the probability of the true class.
  - Actively corrects model predictions by pushing them toward the correct labels.
  - It penalizes incorrect predictions heavily.
  - Examples: Normalised Cross Entropy (NCE), Normalised Focal Loss (NFL)
- **Passive losses:**
  - Focus on reducing the probability of incorrect classes.
  - Minimizes loss without actively pushing predictions toward specific labels.
  - Does not aggressively penalize incorrect predictions.
  - Examples: Mean Absolute Error (MAE), Reverse Cross Entropy (RCE)

  Given active loss $\mathcal{L}_A$ and passive loss $\mathcal{L}_P$, the total loss is:

$$\mathcal{L}_{\text{APL}} = \lambda \mathcal{L}_A + (1 - \lambda)\mathcal{L}_P$$

  where $\lambda \in [0, 1]$ balances the two terms. I set $\lambda = 0.6$ after grid search.

3 Experiments

- Model: 8 Layer CNN
- Optimizer: ADAM learning rate = 0.001
- Epochs: 30
- Metrics: Accuracy, Loss
- Noise Rates: 0.2, 0.4, 0.6, 0.8

4 Results and Discussion

4.1 Performance Comparison between NCE, NFL and CE, FL

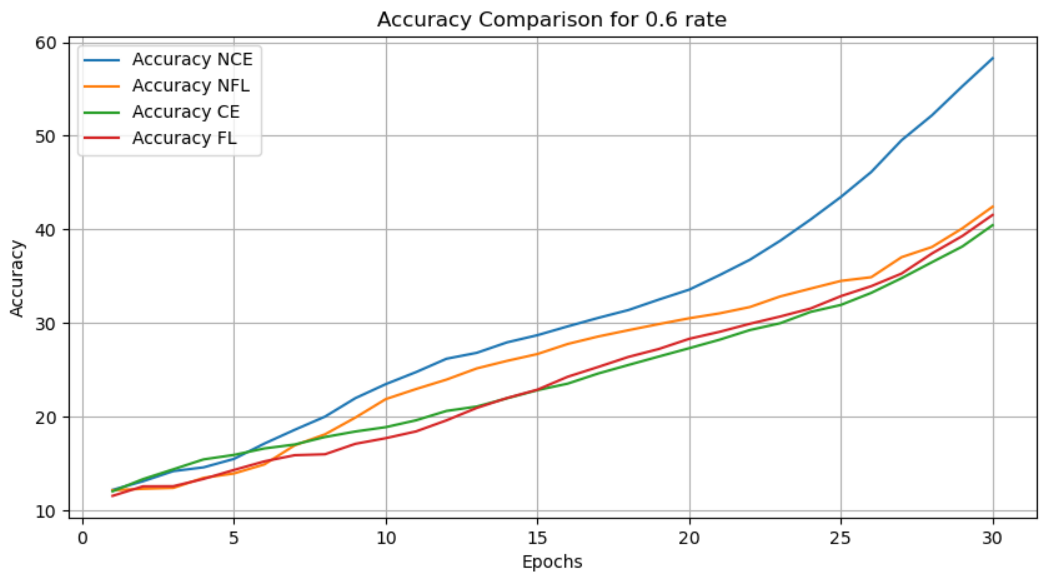

Figure 1: Accuracy vs. Epochs for noise rate of 0.6

4.2 Performance Comparison between NCE + MAE, NFL + MAE, NCE + RCE, NFL + RCE, CE, FL, NCE, NFL

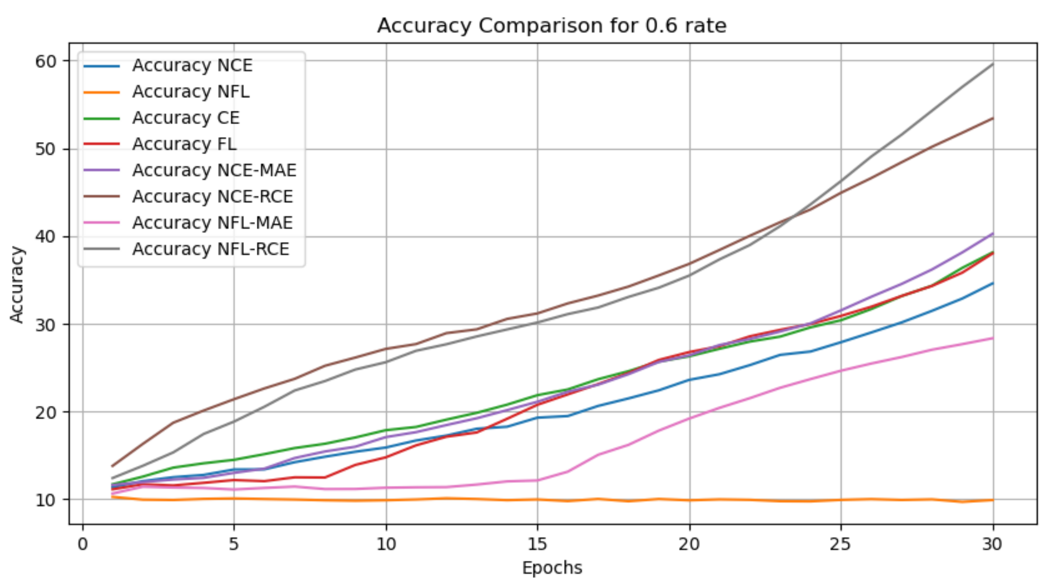

Figure 2: Accuracy vs. Epoch for noise rate of 0.6

4.3 Test Accuracy for all different noises of all the models

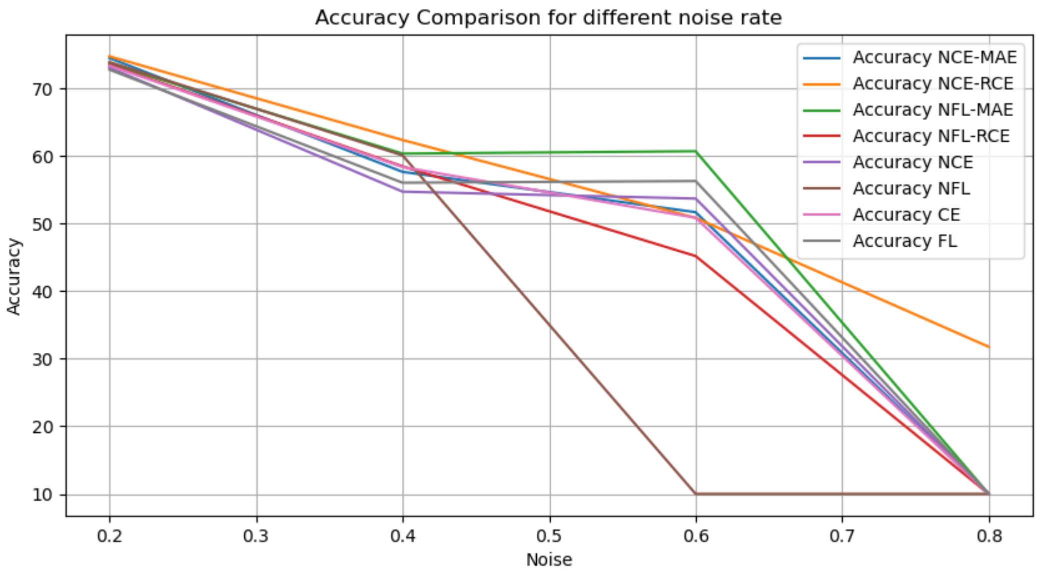

Figure 3: Accuracy vs. Noise Rate

Noise Rate	0.2	0.4	0.6	0.8
NCE	73.07	54.72	53.70	10.00
NFL	73.89	60.09	10.00	10.00
CE	73.35	58.36	50.85	10.00
FL	72.77	56.04	56.29	10.00
NCE + MAE	74.49	57.65	51.68	10.00
NFL + RCE	73.47	58.46	45.19	10.00
NFL + MAE	73.64	60.35	60.70	10.00
NCE + RCE	74.74	62.36	50.84	31.76

Table 1: Test Accuracy under Different Noise Rates

The table and graph (Figure 3) above shows that with higher noises one of the APL framework works the best for classification.

5 Conclusion

This project demonstrates that normalized loss functions and the APL framework can improve robustness to label noise.

References

[1] Ma, X., Huang, H., Wang, Y., Romano, S., Erfani, S., & Bailey, J. (2020, November). *Normalized Loss Functions for Deep Learning with Noisy Labels.* International Conference on Machine

Learning (ICML).