Learning with Label Noise

Effect of using APL(Active Passive Loss) and Dynamic Weights into Samples on Noise Label Dataset

Comp 433 Final Project
Deokyeong Kim (26942105) , Suin Kang(40129337)

Abstract

This project aims to create a robust model by exploring two different methods beyond the baseline model while considering label noise during training. With the CIFAR-10 dataset, the project focuses on addressing two types of noise in images: symmetric label noise and asymmetric label noise. The noise levels considered in this project are 10%, 30%, 60%, and 90%. The first approach involves applying the Active Passive Loss (APL), a modified loss function, to the Baseline Model. The second approach entails incorporating dynamic weights into the samples of the Baseline Model. When compared to the baseline model, the model augmented with both methods consistently demonstrated higher test accuracy across the dataset with different noise levels. Notably, the observed performance similarity on the additional dataset implies that models trained with these two methods exhibit robustness, proving their ability to generalize well to diverse datasets. This successful outcome underscores the effectiveness of both methods in enhancing the overall performance and robustness of the baseline model.

I. Introduction

Label noise is a common issue encountered during the data collection process in the real world, particularly in high-dimensional datasets such as image classification. In high-dimensional data like images, label noise can significantly degrade the performance of a model. If a model is trained with incorrect labels and fails to learn accurate patterns, it ultimately leads to a substantial reduction in the model's accuracy and generalization capability.

In the context of learning with label noise, the challenges arise from the increasing complexity of the model structure, which results in a higher number of parameters for CNNs. Training such complex models demands significant computational resources and time. Additionally, as the model becomes more intricate, the iterative training process for different levels of noise incurs substantial time overhead. To address these issues, employing more powerful hardware becomes essential for efficient model training. In this project, a model leveraging reasonable computational resources and time has been implemented to overcome the challenges posed by label noise and develop a robust solution.

Baseline models typically employ naive empirical loss minimization, such as cross-entropy, which is a standard loss function used in common classification problems. However, when label noise is present, this approach can make the model sensitive to incorrect labels. APL and dynamic weights offer innovative solutions to address label noise in image classification, providing a fresh perspective to mitigate the impact of noisy labels.

The introduction of APL as the first method aims to make the model robust against label noise. APL incorporates both Active Loss, which corrects labels for samples with noise, and Passive Loss, which performs regular training. This combination helps the model mitigate the influence of noise, maintaining high performance. The second approach involves applying dynamic weights to samples, enabling the model to adapt sensitively to samples with high noise levels. By employing dynamic weights for specific samples, the model can focus more on samples with accurate labels while responding more sensitively to samples with considerable noise.

In each section, we compare and analyze the training and testing performance accuracy of the baseline model and the two approaches across various noise levels using the CIFAR-10 dataset, considering both asymmetric and symmetric label noise. Additionally, we will conduct a thorough analysis, examining the intricate connection between time consumption and accuracy. This investigation aims to unveil nuanced trade-offs among training time, inference time, and overall performance, shedding light on the dynamic interplay between time investments and model effectiveness. Finally, we assess the generalization ability of both methods by utilizing an additional evaluation dataset.

II. Literature review

Many researchers have explored various methods to address the challenges posed by noisy labels when building robust models with deep neural networks. The significance of weight initialization techniques in improving the performance of deep learning has been highlighted (Boulila et al., 2021). Studies have delved into the application of different loss functions and performance metrics tailored to specific deep learning problems during model training and evaluation (Terven et al., 2023). Furthermore, it demonstrated the effectiveness of training deep neural networks on noisy labels by incorporating a loss function created with the application of simpler regularization (Ma et al., 2020). This literature is highly relevant to the implementation of the first method in this project. A proposed solution to the challenge of label memorization in deep learning, particularly in datasets with noisy labels, involves training by selectively using some data with clean labels (Han et al., 2018). Ren and colleagues propose various normalization methods and example reweighting algorithms as a means to prevent overfitting to biases in the training set and label noise. Moreover, they suggest an algorithm that allocates new weights to training examples (Ren et al., 2019). This literature is highly pertinent to the implementation of the second method in this project.

III. Dataset

The first dataset, CIFAR-10, was employed to train models under various symmetric and asymmetric label noise levels, including 10%, 30%, 60%, and 90%. These trained models were subsequently utilized for evaluation with the test dataset. The second dataset is MNIST. This test dataset was exclusively employed for testing to assess the generalization capabilities of the models.

The image datasets were preprocessed in advance to enhance learning and promote generalization. Each image was resized to 32x32x3 to align with the specifications of the CIFAR-10 dataset and normalized using the ImageNet means and standard deviations, [0.485, 0.456, 0.406], and [0.229, 0.224, 0.225]. Throughout this project, the datasets were split with an 80:20 ratio. This ratio was selected to ensure enough data for training and a sufficient amount for testing, enabling reliable evaluation.

Two label noise scenarios were used for model training. Symmetric Label Noise involves flipping labels with a probability ϵ during training. For example, with ϵ noise, $100(1-\epsilon)$ of 100 bird images keep the label "bird," while 100ϵ get a random label. Asymmetric Label Noise flips labels based on ϵ and class during training. If there are 100 bird images with ϵ noise, $100(1-\epsilon)$ retain the "bird" label, while 100ϵ receive a different random label.

IV. Presentation of Experiment

1) Baseline Model

Addressing label noise challenges in the CIFAR-10 dataset, the BaselineModel, a robust CNN, is created.

Comprising convolutional and activation layers, followed by max-pooling, the model processes input channels to output 32x32x8. Subsequent layers feature convolutions with ReLU activation and max-pooling to diminish spatial dimensions. Transitioning to fully connected layers, the flattened output connects to two linear layers with ReLU activation. The final layer yields 10 output classes for CIFAR-10 categories.

Xavier initialization enhances training stability by setting weights in the linear layers. Applied as a method to mitigate the risks of gradient vanishing or exploding, it plays a crucial role in the forward method, defining the data flow through the model (Boulila et al., 2021). This process involves passing input tensors through convolutional and fully connected layers, ultimately yielding the final classification output. The model's objective is to achieve robust performance, particularly in the presence of label noise challenges in CIFAR-10.

2) Angle 1 - APL

In reality, datasets often exhibit class imbalances and outliers, making it challenging to apply a one-size-fits-all loss function. The choice of loss function and metrics significantly impacts the performance of the model when designing and evaluating it (Terven et al., 2023). The commonly used Cross Entropy (CE) loss, as employed in our baseline model, is not robust to noise. However, it has been theoretically proven that any loss function can be made robust to noise through a simple normalization. APL is designed as a robust loss function, leveraging the properties enabled by such normalization (Ma et al., 2020).

In the context of addressing Noisy Labels in the Baseline model, our first approach involves modifying the model through the incorporation of the APL function. APL categorizes existing robust functions into "Active" and "Passive" classifications based on their optimization behaviors. Specifically, a loss is deemed "Active" if it optimizes exclusively when $q(k=y|x) = 1$ (Ma et al., 2020), meaning the model predicted the actual class y when inputted data x. Otherwise, set the "Passive" loss.

The APL function combines these losses, expressed as $LAPL = \alpha \cdot LActive + \beta \cdot LPassive$ (Ma et al., 2020) , with a balanced weighting of terms. In this experiment, we used two combinations of active and passive pairs. One combination with Normalized cross entropy(NCE) and Reverse Cross Entropy(RCE) then the second combination with Normalized Focal Loss(NFL) and RCE. The active and passive loss ratio is set to alpha=1 and beta=1.

3) Angle 2 - Dynamic Weights into Samples

Training deep neural networks poses challenges, especially when there's a risk of memorizing labels, particularly in the presence of noisy labels (Han et al., 2018). Deep neural networks function as potent tools for modeling complex input patterns. Nonetheless, they are prone to overfitting due to biases in the training set and label noise. Various forms of training set biases exist, and manually crafted weighting objectives come with their inherent biases (Ren et al., 2019). Suggesting the use of weighting algorithms is a common proposal to address these challenges, along with the application of various normalization techniques.

The second strategy to tackle the issue of noisy labels entails employing dynamic sample weights within the BaselineModel. The key aspect of this approach involves computing sample weights based on prediction confidence with a designated threshold of 0.6. This incorporation of weighted loss calculation is seamlessly embedded into the training loop of the Baseline Model.

This methodology enhances the training dynamics by dynamically adjusting the influence of each sample on the overall loss, giving prominence to samples with higher confidence while de-emphasizing those with lower confidence. The meticulous implementation of this mechanism is pivotal for effectively navigating the challenges posed by noisy labels during training, enabling the model to adapt and acquire more robust learning from the available data.

V. Result

1) Experiment setup

The baseline model involved training the CNN encoder from scratch. For this purpose, the CCE loss function and the Stochastic Gradient Descent (SGD) optimizer were chosen as the objective function and optimizer, respectively. To address regularization, weight decay was also implemented. Hyperparameter tuning was conducted to control the learning algorithm and achieve optimal results. The learning rate was set to 0.01, and training was performed for 7 epochs with a batch size of 256. Despite conducting training and testing for 20 and 30 epochs to avoid underfitting, the validation loss started to diverge after epoch 10. There is a tiny difference in loss between 7.5 and 10, and it is evident that the validation loss consistently converged over the 7 epochs as depicted in Figure 1. Due to computational complexity, we decided to set the number of epochs in training to 7 due to computational complexity.

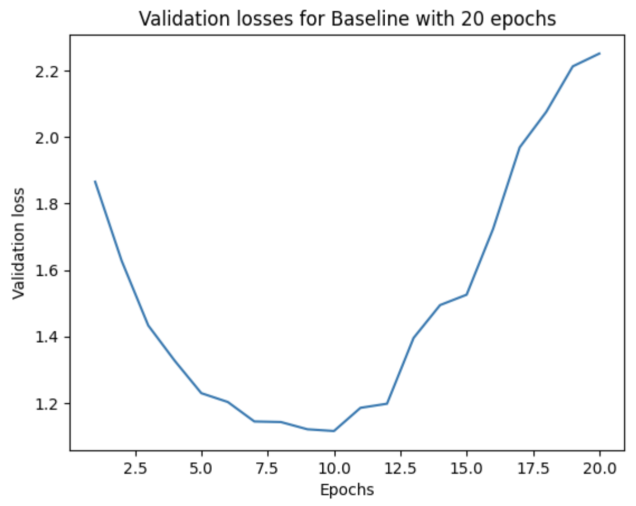

Figure 1. Validation loss for epoch 20 in the Baseline model

2) Main Result

After implementing the baseline model, training was carried out for 7 epochs. At each epoch, various levels of noise were introduced during the training process, and the corresponding training loss/accuracy, validation loss/accuracy, and training time were documented. As indicated in Figure 2, the validation loss exhibited convergence at noise levels 0.0, 0.1, and 0.3. However, noise levels beyond 0.6 displayed poor convergence, with the achieved training accuracy in the last epoch hovering around 10-20%. In contrast, stability was observed in both training and validation when compared to symmetric noise. Notably, the model with a noise level of 0.9 demonstrated effective training, achieving a high training accuracy and significantly smaller validation loss, as illustrated in Figure 3.

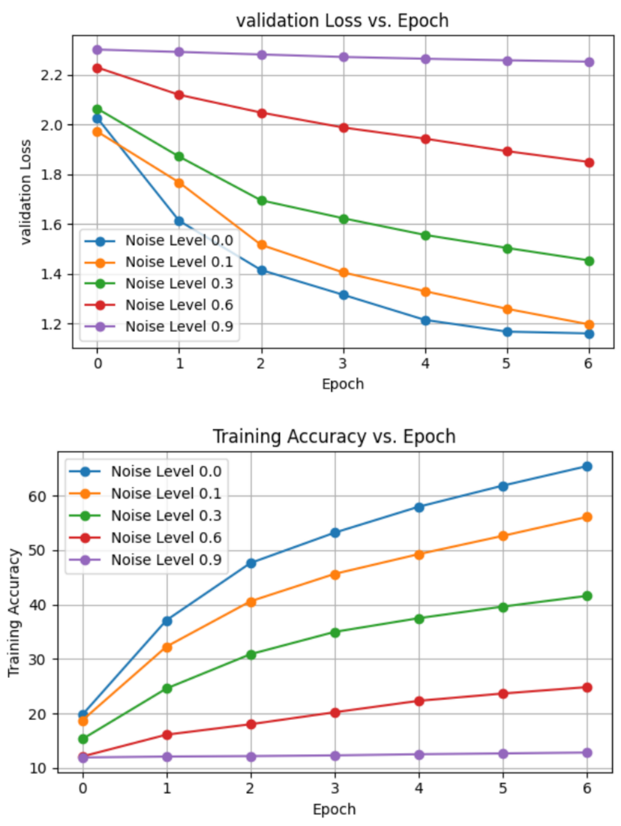

Figure 2. Validation loss and train accuracy for Baseline model with symmetric noise

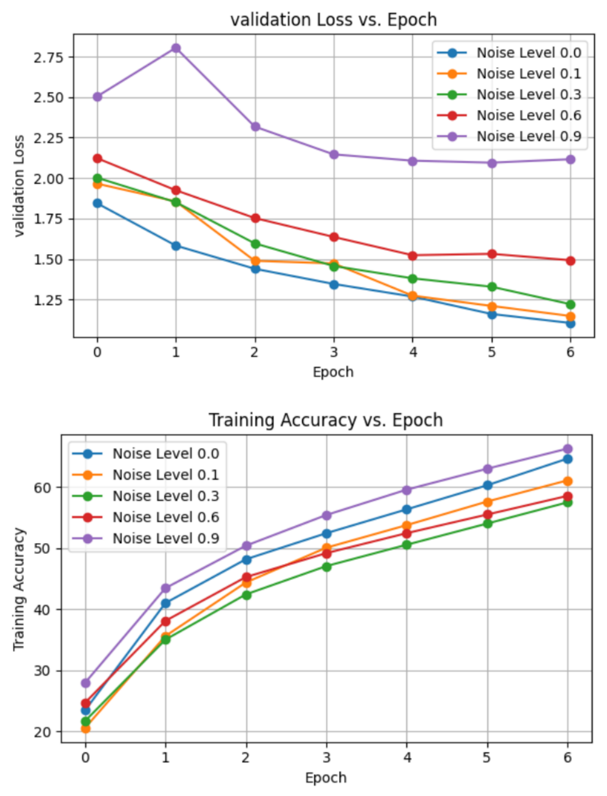

Figure 3. Validation loss and train accuracy for Baseline model with asymmetric noise

The hyperparameters alpha and beta are determined through hyperparameter tuning for the APL approach. Three sets of alpha and beta are used for testing: (1, 10), (1, 1), and (1, 0.5). Among these pairs, (1, 1) resulted in the highest validation accuracy during training and was selected for the remainder of the project.

The APL model with NFL active loss and RCE passive loss exhibits more than a 3% improvement in test accuracy for all symmetric noise levels, as illustrated in Table 3. APL(NFL, RCE) also demonstrates the highest improvement in asymmetric noise at level 0.6, as presented in Table 2. Overall, APL(NFL, RCE) outperformed APL(NCE, RCE), and both APL models show slight improvement compared to baseline models.

After conducting hyperparameter tuning for dynamic weight thresholds in the baseline model, the model's performance varies across different threshold values, as evident from Table 1. For a threshold of 0.4, the model achieved a training accuracy of 46.31% and a validation accuracy of 39.25%. With a threshold of 0.6, both training and validation accuracies improved, reaching 60.33% and 45.41%, respectively. Training with a threshold of 0.8 yielded a training accuracy of 58.75% and a validation accuracy of 45.00%. Considering these outcomes, we have chosen a threshold of 0.6 as the final selection, given its balanced improvement in both training and validation accuracies compared to other thresholds.

Threshold	Train Loss	Train Accuracy (%)	Validation Loss	Validation Accuracy (%)
0.4	1.6	46.31	2.05	39.25
0.6	1.12	60.33	1.58	45.41
0.8	1.13	58.75	1.49	45

Table 1. Hyperparameter tuning result for applying dynamic weight in the Baseline model

Based on hyperparameter tuning, training was performed with the threshold of 0.6 for both symmetric and asymmetric noise labels. The training and validation trends over 7 epochs resembled those of the baseline model, as depicted in Figures 1 and 2. As shown in Figure 4, the dynamic weight model demonstrates robustness primarily in low-noise scenarios for symmetric noise levels. In datasets with pure, noise-free instances, both models achieved the highest accuracy, with the dynamic weight model surpassing the baseline model (63.23% vs. 60.78%). At noise levels 0.0 and 0.1, the dynamic weight model showed a slight improvement in performance, from 53.39% to 54.26%, and at noise level 0.3, the accuracy increased from 53.39% to 54.26%. Unfortunately, beyond these noise levels, the dynamic weight model exhibited accuracy similar to the baseline model for symmetric noise.

In contrast, for asymmetric noise, the dynamic weight model showed a slightly enhanced test accuracy across all noise levels. As illustrated in Figure 5, at noise level 0.6, the accuracy increased from 44.26% to 45.50%, and at noise level 0.9, it rose from 41.22% to 42.32%. While the overall test accuracy improvements were modest for both noise types, the dynamic weight model demonstrated relative robustness in higher noise levels for asymmetric noise compared to the baseline model.

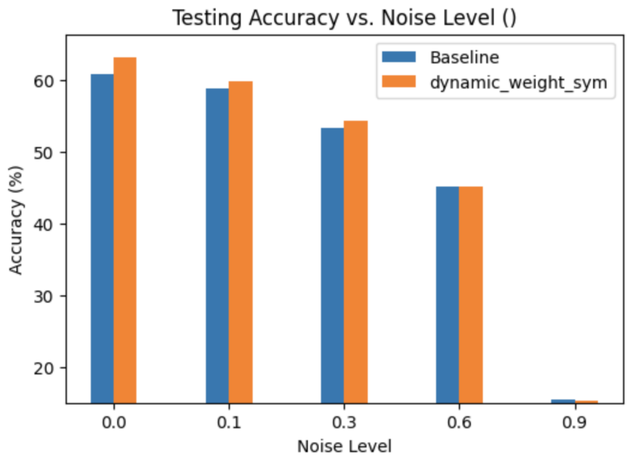

Figure 4. Test accuracy comparison between Baseline model and Dynamic Weight model trained with symmetric noise

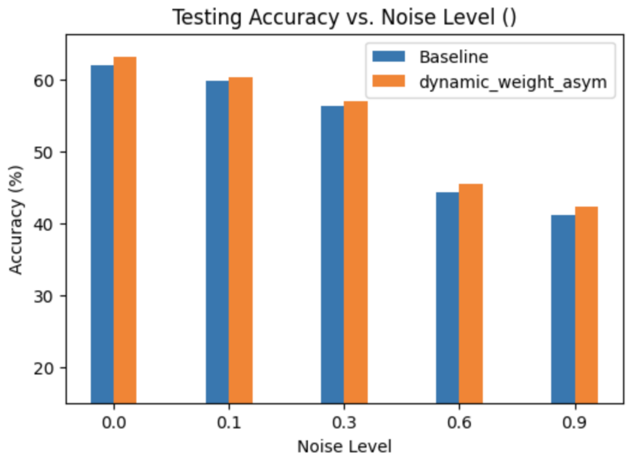

Figure 5. Test accuracy comparison between Baseline model and Dynamic Weight model trained with asymmetric noise

We used the MNIST test dataset to test the performance of our models.We have chosen three models that show improvement to test on new test data. The chosen models are APL(NFL, RCE) with symmetric noise 0.9, APL(NFL, RCE) with asymmetric noise 0.6 and Dynamic weight model asymmetric with noise 0.9 as shown in Table 2. All three models show improvement on the Test dataset.

	symmetric with noise level 0.9	
MNIST	Baseline	15.44
	APL(NFL, RCE)	19.12

	Asymmetric with noise level 0.6	
MNIST	Baseline	44.26
	APL(NFL, RCE)	47.2

	Asymmetric with noise level 0.9	
MNIST	Baseline	41.22
	Dynamic weight	42.32

Table 2. Test accuracy(%) comparison on MNIST test dataset with model trained on respective noise type and level on cifar10 dataset.

Dataset	Methods	Clean (0.0)	Symmetric Noise Level (epsilon)			
			0.1	0.3	0.6	0.9
CIFAR-10	Baseline CE	60.78	58.87	53.39	45.14	15.44
	Dynamic weight	63.23	59.86	54.26	45.21	15.31
	APL(NCE+RCE)	64.07	61.8	56.09	47.63	16.75
	APL(NFL+RCE)	**60.85**	**63.13**	**57.27**	**48.92**	**19.12**

Table 3. Test accuracy(%) comparison with Symmetric noise levels with epsilon 0.0, 0.1, 0.3, 0.6 and 0.9. Observe improvement of APL(nfl,rce) model with noise level 0.9 compared to the baseline model with noise level 0.9.

Dataset	Methods	Clean (0.0)	Asymmetric Noise Level (epsilon)			
			0.1	0.3	0.6	0.9
CIFAR-10	Baseline CE	61.91	59.83	56.31	44.26	41.22
	Dynamic weight	63.21	60.24	59.95	45.5	42.32
	APL(NCE+RCE)	63.18	61.39	57.54	46.3	40.89
	APL(NFL+RCE)	**63.57**	**60.88**	**58.59**	**47.2**	**40.16**

Table 4. Test accuracy(%) comparison with Asymmetric noise levels with epsilon 0.0, 0.1, 0.3, 0.6 and 0.9. Observe improvement of APL(nfl,rce) model with noise level 0.6 compared to the baseline model with noise level 0.6.

VI. Limitations

When observing the validation loss and training accuracy at different noise levels for the baseline and two alternative methods, it was evident that the validation loss did not significantly improve, especially at higher noise levels, such as 0.9. Consequently, this lack of improvement in validation loss translated to lower accuracy during testing. Despite our efforts to enhance the baseline model through two novel approaches, the test accuracy, varying with noise levels and types, did not show a significant improvement compared to the testing results of the baseline model.

The experimentation process has been challenging primarily because it requires training, validating, and testing models for each of the two different noise types at various noise levels. This extensive process demands a significant amount of time, and, most importantly, it poses difficulties due to the limited GPU usage in Colab. We also aimed to explore an alternative approach by incorporating semantic layers and engaging in semi-supervised learning within the architecture. However, training this architecture proved to be impossible due to the limited RAM capability in the Google Colab work environment.

VII. Conclusion

In this project, we performed various experiments to develop a robust model even in the presence of noise in data labels. The experiments included a method for weighting losses, an APL method combining two different losses, a method for performing semantic clustering, and an architectural approach using transfer learning. Among these, the results of applying Active Passive Loss showed performance improvement compared to the baseline model. Additionally, results from MINST data confirmed that our model had excellent generalization ability.

Throughout this project, as we devised and experimented with various methods to develop robust models against different noise levels, we gained a deeper understanding of the fundamental concepts that significantly impact deep neural networks. Our exploration and experimentation provided insights into addressing and mitigating challenges faced by CNNs, such as label noise, from various perspectives.

VIII. Reference

Boulila, W., Driss, M., Al-Sarem, M., Saeed, F., & Krichen, M. (2021, February 13). *Weight initialization techniques for deep learning algorithms in remote sensing: Recent trends and future perspectives*. arXiv.org. https://arxiv.org/abs/2102.07004v1

Terven, J., Cordova-Esparza, D. M., Ramirez-Pedraza, A., & Chavez-Urbiola, E. A. (2023, September 6). *Loss functions and metrics in deep learning*. arXiv.org. https://arxiv.org/abs/2307.02694

Ma, X., Huang, H., Wang, Y., Romano, S., Erfani, S., & Bailey, J. (2020, June 24). *Normalized loss functions for deep learning with noisy labels*. arXiv.org. https://arxiv.org/abs/2006.13554

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., & Sugiyama, M. (2018, October 30). *Co-teaching: Robust training of deep neural networks with extremely noisy labels*. arXiv.org. https://arxiv.org/abs/1804.06872

Ren, M., Zeng, W., Yang, B., & Urtasun, R. (2019, May 5). *Learning to reweight examples for robust deep learning*. arXiv.org. https://arxiv.org/abs/1803.09050

IX. Appendix

https://colab.research.google.com/github/suinkangme/comp433_project/blob/main/final_project_submit.ipynb