Applying R-Drop on ViT and CNN Models in CIFAR Image Classification

Anonymous submission

Abstract

This study investigates the effectiveness of R-Drop regularization (Liang et al. 2021) in enhancing accuracy across Vision Transformer (ViT) and Convolutional Neural Network (CNN) architectures on the CIFAR-10 and CIFAR-100 datasets. Four sets of experiments were performed, each comparing the performance of models trained with and without R-Drop regularization. Specifically, ViT-B/16 and ViT-L/16 models were tested on both datasets, and VGG-16 was evaluated on CIFAR-100 with different R-Drop coefficients. While R-Drop achieved some accuracy improvements in certain configurations—such as a 0.22 percentage point increase on ViT-B/16 for CIFAR-100, its overall impact could not be fully gauged due to limited computational resources. In the VGG-16 experiment, a lower R-Drop coefficient showed slight improvement, while a higher coefficient suggested that too much regularization may hinder performance. These findings confirm that R-Drop does offer improvements in accuracy, but further investigation with extended training durations is needed to assess the true measure of its effectiveness across models and datasets.

Repository — https://anonymous.4open.science/r/R-Drop-ECE570-C73F
Original R-Drop repository — https://github.com/dropreg/R-Drop

Introduction

Overfitting is a persistent challenge in machine learning, especially in neural networks, where models can become overly attuned to specific training examples rather than learning generalizable patterns. This phenomenon hampers model performance on unseen data, reducing its practical utility. Regularization techniques are essential in mitigating overfitting, with dropout (Srivastava et al. 2014) being one of the most widely adopted methods for neural networks. Dropout works by randomly deactivating, or "dropping," neurons during each training iteration, which forces the network to learn distributed representations that are less reliant on individual neurons. By creating a series of sub-networks through dropout, the model benefits from an ensemble effect, leading to enhanced generalization. However, conventional dropout introduces inconsistencies between the training and inference phases, as the randomly dropped neurons during training cause variations in outputs that can negatively impact model behavior at inference. The R-Drop method (Liang et al. 2021) addresses this inconsistency by incorporating a regularization term that minimizes the bidirectional Kullback-Leibler (KL) divergence between the outputs of two forward passes of the same input, each with a different dropout mask. This approach maintains dropout's regularization benefits while enforcing output consistency, a critical aspect in achieving stable performance in inference.

This project explores the efficiency of R-Drop within the scope of image classification tasks and is structured in two phases. The first phase applies R-Drop to ViT models, aiming to reproduce results comparable to those reported in the original R-Drop paper (see Table 1). The second phase extends R-Drop to a CNN architecture, VGG-16. While the original R-Drop study focused on transformer models, its effect on CNNs was left as future work, leaving an open question on its utility in this type of network. Regularization in CNNs is especially relevant due to their reliance on spatial structures in images, which limits the applicability of standard dropout in convolutional layers (Cai et al. 2019). The second phase of this project thus aims to investigate whether R-Drop can effectively improve generalization in CNNs. The training configuration in phase 1 was set to closely mirror the one used in the original R-Drop paper to ensure comparable results on the ViT model. In phase 2, the configuration is modified slightly to better align with the characteristics of CNNs.

Method	CIFAR-100
ViT-B/16	92.64
ViT-B/16 + RD	**93.29**
ViT-L/16	93.44
ViT-L/16 + RD	**93.85**

Table 1: R-Drop's performance on Image Classification

Problem Definition

In deep neural networks, regularization techniques are essential to counteract overfitting and improve generalization. Dropout is one of the most widely used regularization methods, where random subsets of units in the network are set to zero during training to prevent units from co-adapting too closely. This approach effectively combines a large number of network architectures by introducing random sub-

networks throughout training, helping to reduce generalization error.

However, dropout introduces a significant inconsistency between the training and inference phases. During training, the model is exposed to randomly altered architectures due to the dropout mechanism, effectively training a distribution of sub-models. In contrast, at inference time, dropout is disabled, and the model operates in its full state. This discrepancy creates a misalignment between the models seen during training and those deployed at inference, leading to a drop in performance due to the model's inability to generalize from its training sub-models to its full architecture. To address this inconsistency, the R-Drop technique was introduced to enforce consistency among the predictions generated by different sub-models. The core idea behind R-Drop is to regularize the model's output by minimizing the difference between predictions from two forward passes of the same input. Formally, given an input x, the model performs two separate forward passes, yielding two output distributions $P_1(x)$ and $P_2(x)$. Due to the independent dropout operations in each pass, $P_1(x)$ and $P_2(x)$ are generated by two distinct sub-models of the network.

The R-Drop method encourages consistency between these two distributions by adding a bidirectional KL divergence term to the loss function. Specifically, the KL divergence between $P_1(x)$ and $P_2(x)$ is calculated as follows:

$$L_{KL} = \frac{KL(P_1(x)\|P_2(x)) + KL(P_2(x)\|P_1(x))}{2}$$

Consequently, the overall loss function becomes:

$$L = L_{CE} + \alpha \cdot L_{KL}$$

where L_{CE} represents the standard cross-entropy loss, and α is a coefficient that controls the strength of the KL divergence regularization. This KL divergence term penalizes differences between the two distributions, effectively enforcing a constraint on the model to produce similar outputs from different dropout-influenced sub-models. This method reduces the impact of randomness introduced by dropout, thereby enhancing the model's generalization capability beyond that of traditional dropout regularization alone.

Related Work

Dropout with Expectation-linear Regularization

The Expectation-linear Dropout (ELD) technique (Ma et al. 2016), introduces a similar approach to regularizing dropout by trying to analyze and control the inference gap—the discrepancy between training and inference behaviors. The paper does this by formulating dropout as a tractable approximation of a latent variable model. ELD introduces a regularization term to the standard dropout objective, explicitly controlling this inference gap during training. This approach minimizes the performance loss often observed when transitioning from training to inference. ELD is designed to be as computationally efficient as standard dropout, and the paper's experiments demonstrate that this explicit control of the inference gap reliably improves performance across benchmark image classification tasks.

Fraternal Dropout

Fraternal Dropout (FD) (Zolna et al. 2018) is another method that trains two identical networks with different dropout masks and encourages them to behave similarly by minimizing the discrepancy in their predictions. FD was designed with a focus on recurrent neural networks (RNNs), where training is particularly challenging. By minimizing the divergence between the outputs of these "fraternal" networks, FD regularization aims to make the model less sensitive to the dropout configuration, which narrows the gap between training and inference behaviors.

Comparisons with R-Drop

As stated in the original R-Drop paper, R-Drop, ELD, and FD all pursue consistency in dropout training, but they do so through distinct mechanisms. ELD focuses on reducing the gap between the submodel with dropout in training and the expected full model in inference. In contrast, R-Drop and FD emphasize consistency between multiple sub-models created by dropout during training. Furthermore, both ELD and FD use the L2 distance in the regularization loss term, whereas R-Drop uses the KL divergence.

Other Dropout Techniques

While random neuron deactivation has been proven to be an effective regularization technique, the authors of "AutoDropout: Learning Dropout Patterns to Regularize Deep Networks" (Pham and Le 2021) point out that imposing specific patterns that govern which neurons are dropped can enhance performance. However, designing these dropout patterns manually for different architectures, tasks, and domains is a time-consuming and difficult process. The key contribution of the AutoDropout paper is the development of an automated system that learns specialized dropout patterns for various model architectures through a reinforcement learning (RL) controller. This system searches for optimal dropout patterns that maximize validation performance on a target network and dataset. This automation eliminates the need for manual design and allows for efficient exploration of dropout patterns tailored to specific architectures like CNNs and Transformers.

The "Group-Wise Dynamic Dropout Based on Latent Semantic Variations" paper (Ke et al. 2020) introduces a dropout technique designed to address the limitations of conventional dropout in handling feature co-adaptations in CNNs. Unlike traditional methods that drop individual activations with a fixed probability, this approach adaptively adjusts dropout rates for groups of neurons based on their latent semantic relationships. By leveraging the variations in semantic representations, this method enhances model robustness and performance, particularly in tasks involving complex and nuanced data.

Moving beyond traditional dropout techniques, the "Beyond Dropout: Feature Map Distortion to Regularize Deep Neural Networks" paper (Tang et al. 2020) introduces another approach to improving the generalization ability of CNNs by introducing feature map distortions. Instead of disabling or zeroing out parts of the network, the paper introduces the idea of applying perturbations to the feature maps.

This helps decrease the Rademacher complexity of intermediate network layers, which in turn improves the generalization ability of the network. By distorting feature maps, the method introduces a more flexible way to obstruct information flow in the network, causing a regularizing effect.

Methodology

The methodology employed in this study involved four distinct experiments to evaluate the impact of R-Drop regularization on different neural network architectures and datasets. The primary objective of each experiment was to determine whether the application of R-Drop resulted in improved model accuracy. For each architecture-dataset pair listed below, two sets of models were fine-tuned: one with R-Drop and one without. Specifically, the following experiments were conducted:

- ViT-B/16 on CIFAR-10
- ViT-B/16 on CIFAR-100
- ViT-L/16 on CIFAR-100
- VGG-16 on CIFAR-100

Models and Datasets

The ViT-B/16 and ViT-L/16 models are Vision Transformer (Dosovitskiy et al. 2020) architectures with approximately 85 million and 307 million parameters, respectively. VGG-16, a widely-used CNN network with 138 million parameters, was selected to assess the effectiveness of R-Drop on CNNs. Experimentation with the ResNet-50 architecture was initially considered but was not feasible due to limited computational resources. The CIFAR-10 and CIFAR-100 datasets were used, each containing 60,000 images of 32x32 resolution. CIFAR-10 includes 10 classes, while CIFAR-100 spans 100 classes, posing a greater classification challenge due to increased label complexity. The ImageNet dataset used in the original R-Drop paper would have provided additional insights, but computational constraints precluded its use.

R-Drop Training Process

R-Drop training differs from standard training in that it requires two forward passes per training step. This additional computation is necessary to obtain two distinct prediction distributions for each input, between which the bidirectional KL divergence is calculated and added to the loss function. This dual-pass structure increases computational demand and extends training time. The resulting R-Drop loss is the sum of the cross-entropy loss and the scaled KL divergence between the two forward passes.

Hyperparameters

The hyperparameters for the ViT models were aligned as closely as possible with those specified in the R-Drop and Vision Transformer papers, considering available computational resources. For the VGG-16 model, the same configuration was used with minor adjustments to account for the model's architectural differences. The default configuration applied across experiments is as follows:

- R-Drop coefficient α of 0.6
- Training batch size of 256 (reduced from the recommended 512 for ViT due to memory limitations)
- Evaluation batch size of 64
- SGD optimizer with a learning rate of 0.01, momentum of 0.9, and no weight decay
- Cosine annealing scheduler with a warm-up phase of 500 steps
- Base learning rate of 0.01
- Gradient accumulation steps set to 16
- Gradient clipping to a maximum norm of 1.0 for training stability

If any deviations from this configuration occurred, they are specified in the relevant sections of each experiment.

Training Epochs and Early Stopping

Based on the guidelines reported in the ViT paper (Dosovitskiy et al. 2020), fine tuning the models on CIFAR-10 or CIFAR-100 would have required approximately 10,000 steps. However, computational constraints ultimately determined the number of epochs that were used for training. In order to save computational costs, R-Drop models were trained first with early stopping enabled, typically with a patience of 5. After early stopping was triggered, the best accuracy attained by the R-Drop model was noted and the recorded epoch count was used for training the non-R-Drop models without early stopping.

Data transformation

To improve model generalization, data transforms were applied on each dataset. For training, resizing, random horizontal flips, cropping, and normalization based on the mean and standard deviation values of each dataset were applied. For validation, only resizing and normalization were applied.

Experimental Results

Experiment 1: ViT-B/16 on CIFAR-10

CIFAR-10 was not used in the original R-Drop paper, but it was decided to test the R-Drop implementation on it before moving to the more complex CIFAR-100. Two ViT-B/16 models were fine-tuned on the CIFAR-10 dataset, with one employing R-Drop training and the other employing standard training. The results revealed that R-Drop offered no tangible accuracy improvement in this configuration. The R-Drop model achieved a validation accuracy of 98.60% before stopping at epoch 55 (the training was set for 60 epochs), while the standard model achieved 98.59%. On a simpler dataset like CIFAR-10, using R-Drop did not translate into a significant accuracy gain.

Experiment 2: ViT-B/16 on CIFAR-100

Two ViT-B/16 models were trained on the CIFAR-100 dataset, with one employing R-Drop training and the other employing standard training. The training process was initially set for 90 epochs, however early stopping triggered

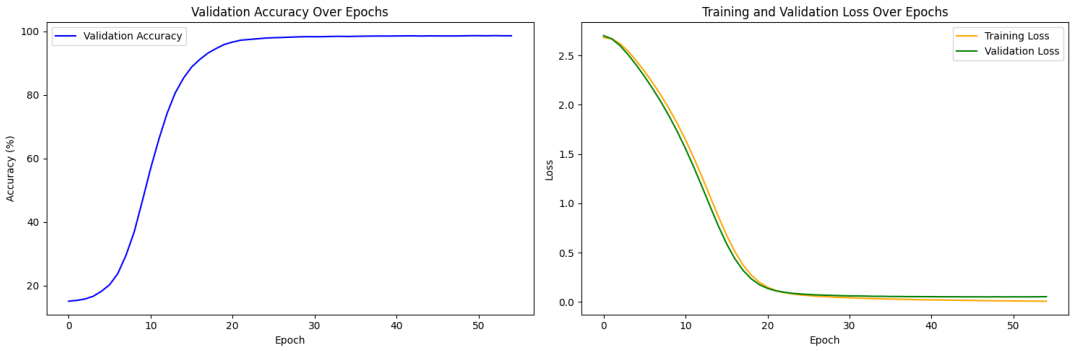

Figure 1: ViT-B/16 Performance on CIFAR-10 with R-Drop (α=0.6)

at epoch 86. The model trained with R-Drop achieved a final validation accuracy of 91.49%, whereas the non-R-Drop model reached an accuracy of 91.27%, marking a performance improvement of 0.22 percentage points when R-Drop was applied. Although the improvement is modest compared to the 0.65 point increase reported in the original R-Drop paper, it nonetheless underscores the efficacy of R-Drop in enhancing model generalization and accuracy in a more challenging dataset like CIFAR-100.

Method	CIFAR-10	CIFAR-100
ViT-B/16	98.59	91.27
ViT-B/16 + RD	**98.60**	**91.49**

Table 2: ViT-B/16 Accuracy Results

Experiment 3: ViT-L/16 on CIFAR-100

In this experiment, two ViT-L/16 models were fine-tuned on the CIFAR-100 dataset, with one employing R-Drop training and the other employing standard training. Due to the ViT-L/16 model's larger size and its high computational demands, both models were trained using a batch size of 64, smaller than the originally planned batch size of 512. The training was initially intended to run for 85 epochs, however, resource limitations necessitated a premature termination after 24 epochs. Consequently, neither model reached convergence, preventing any definitive conclusions on the effectiveness of R-Drop regularization.

At the end of 24 epochs, the R-Drop model achieved a validation accuracy of 86.34%, while the non-R-Drop model achieved a nearly identical accuracy of 86.31%. Given the similarity in performance and the limited training duration, these results are inconclusive regarding the advantages of R-Drop in improving accuracy for this configuration. Although there was a slight numerical advantage for the R-Drop model, it is likely due to the early termination. Further training would be needed to accurately assess R-Drop's potential impact on accuracy for the ViT-L/16 architecture on CIFAR-100.

Method	CIFAR-100
ViT-L/16	86.31
ViT-L/16 + RD	**86.34**

Table 3: ViT-L/16 Accuracy Results

Experiment 4: VGG-16 on CIFAR-100

Three VGG-16 models were trained on the CIFAR-100 dataset to assess the impact of the R-Drop regularization technique on CNNs. The models included a baseline that did not include R-Drop, one which used R-Drop with a coefficient of 0.6, and another which used R-Drop with a coefficient of 0.9. All models were trained using the same configuration applied in earlier experiments with ViT-B/16 models, except with the addition of a weight decay of 0.0001 in the SGD optimizer. Training was conducted for a modest 60 epochs due to computational limitations, which has unfortunately prevented full convergence of the models. Despite these constraints, preliminary results suggest a nuanced impact of R-Drop on VGG-16 performance. The model with a moderate R-Drop coefficient of 0.6 achieved a validation accuracy of 62.68%, a slight improvement over the baseline model's 62.55%. This modest gain indicates that R-Drop can contribute positively to CNN regularization when the coefficient is tuned carefully. However, the model with a higher R-Drop coefficient of 0.9 performed worse than both the baseline and the 0.6 coefficient model, achieving an accuracy of 62.06%. This decline at higher regularization strengths may indicate that excessive consistency enforcement can limit a CNN's capacity to capture complex patterns.

These results are promising but inconclusive, primarily due to the limited number of training epochs. Had the models reached convergence, the observed differences might have been more pronounced, providing clearer insights into R-Drop's efficacy within CNNs. Further studies with more compute resources and longer training durations may better assess R-Drop's utility for CNN architectures in image classification tasks.

Conclusion and Future Directions

This project aimed to assess the impact of R-Drop regularization on the accuracy of neural networks, specifically Vi-

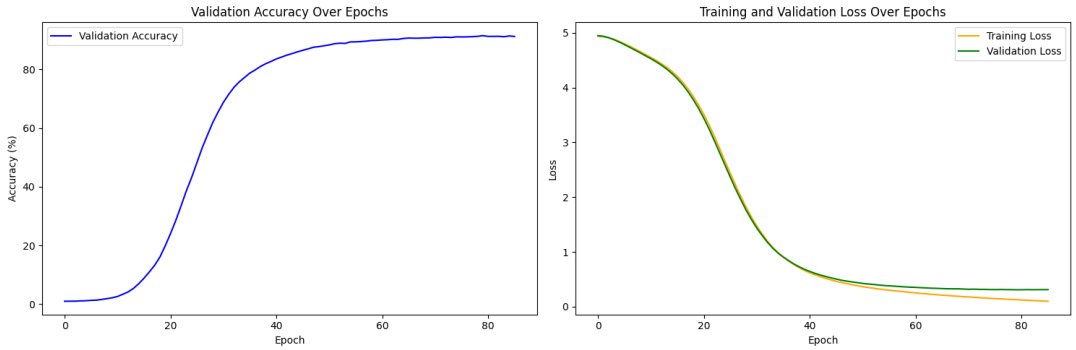

Figure 2: ViT-B/16 Performance on CIFAR-100 with R-Drop (α=0.6)

Method	CIFAR-100
VGG-16	62.55
VGG-16 + RD (α=0.6)	**62.68**
VGG-16 + RD (α =0.9)	62.06

Table 4: VGG-16 Accuracy Results

sion Transformers and VGG-16, on CIFAR datasets. Across four experiments, R-Drop generally led to slight improvements in accuracy, as seen in the ViT-B/16 model on CIFAR-100, which achieved a 0.22 percentage point gain. However, in most cases, the improvements were negligible, as with the ViT-B/16 on CIFAR-10, and the ViT-L/16 on CIFAR-100, which showed no significant difference between models trained with and without R-Drop. The VGG-16 experiment yielded interesting results, revealing that a lower R-Drop coefficient (α=0.6) yielded a slight improvement, whereas a higher coefficient (α=0.9) hindered accuracy, suggesting that too much regularization may adversely impact model performance.

Given the limited training epochs due to computational constraints, some models in these experiments did not fully converge, likely underestimating the true effect of R-Drop on model performance. Future work should focus on retraining these models with an extended number of epochs, particularly for VGG-16, to better understand the potential of R-Drop in different neural networks. Varying the α coefficient accross multiple model trainings would also be an interesting avenue to better assess its effect.

References

Cai, S.; Gao, J.; Zhang, M.; Wang, W.; Chen, G.; and Ooi, B. C. 2019. Effective and Efficient Dropout for Deep Convolutional Neural Networks. *CoRR*, abs/1904.03392.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *CoRR*, abs/2010.11929.

Ke, Z.; Wen, Z.; Xie, W.; Wang, Y.; and Shen, L. 2020. Group-Wise Dynamic Dropout Based on Latent Semantic Variations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07): 11229–11236.

Liang, X.; Wu, L.; Li, J.; Wang, Y.; Meng, Q.; Qin, T.; Chen, W.; Zhang, M.; and Liu, T.-Y. 2021. R-Drop: Regularized Dropout for Neural Networks. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 10890–10905. Curran Associates, Inc.

Ma, X.; Gao, Y.; Hu, Z.; Yu, Y.; Deng, Y.; and Hovy, E. H. 2016. Dropout with Expectation-linear Regularization. *CoRR*, abs/1609.08017.

Pham, H.; and Le, Q. 2021. Autodropout: Learning dropout patterns to regularize deep networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 9351–9359.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(56): 1929–1958.

Tang, Y.; Wang, Y.; Xu, Y.; Shi, B.; Xu, C.; Xu, C.; and Xu, C. 2020. Beyond dropout: Feature map distortion to regularize deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 5964–5971.

Zolna, K.; Arpit, D.; Suhubdy, D.; and Bengio, Y. 2018. Fraternal Dropout. arXiv:1711.00066.