Investigating the Effects of Bag of Tricks for Object Classification Models: ResNet and ViT

Heejun Park
Department of Immersive Media Engineering, Sungkyunkwan University
25-2, Seonggyungwan-ro, Jongno-gu, Seoul, Republic of Korea
parkie0517@gmail.com

Abstract

Diverse methods exist for enhancing the performance of deep neural networks. These include various data preprocessing techniques and training strategies, collectively referred to as a "bag of tricks". The objective of this report is to identify optimal combinations of these tricks that can yield superior performance. These tricks were applied in combinations to two object classification models: ResNet and Vision Transformer (ViT). When trained using the CIFAR-10 dataset with appropriate combinations of tricks, I observed performance improvements of 9.77% and 10.14% for ResNet-50 and ViT-B/4, respectively, compared to their baseline versions. Thus, this empirical study underscores the importance of using different combinations of tricks in developing deep neural networks, highlighting their impact on model efficacy.

1. Introduction

In the field of computer vision, well-known papers often focus on enhancing the performance of specific tasks such as image classification. For instance, He et al. [3] introduced the concept of residual connections, whereas Dosovitskiy et al. [1] demonstrated the efficacy of Transformer [12] structures in image classification tasks, thereby improving the models' generalizability. In these cases, various methods collectively referred to as a "bag of tricks" are applied to further elevate model performance. These methods include data augmentation through preprocessing and the implementation of diverse training strategies.

However, such papers often fail to adequately introduce the bag of tricks methods they employ. Even if they conduct ablation studies, they do not explore in detail how their chosen tricks enhances model performance. Addressing this gap, He et al. [4] investigate the impact of various tricks on model performance and computational requirements during the training process of different object classification models. This report utilizes some of the methods from [4] and explores other basic tricks not mentioned in it, examining their influence on the training of the models.

The experiments are conducted using well-known Convolutional Neural Network (CNN) and Transformer-based models, specifically ResNet-50 and ViT-B/4, with the CIFAR-10 [6] dataset. I found significant performance improvements when appropriate combinations of tricks were applied to standard ResNet-50 and ViT-B/4 models. This suggests that choosing the right combination of tricks suitable for the model and dataset can greatly influence performance enhancement.

The structure of this report is as follows: Sec. 2 provides an introduction to the two models used in the experiments. Sec. 3 introduces the bag of tricks used in the experiments, examining how each method can influence the deep neural network models' training process in detail. Sec. 4 presents an introduction to the dataset used in the experiments and reports the experimental results. Sec. 5 discusses subjective interpretations of the results. Finally, Sec. 6 concludes the report.

I encourage the use of my code for those who wish to replicate the experiments. All codes used for this report are available through: https://github.com/parkie0517/Bag_of_Tricks.

2. Object classification models

2.1. ResNet-50

ResNet-50, a 50-layer model in the Residual Network family, is designed to overcome the vanishing gradient problem in deep neural networks through its unique residual connections. These skip connections enable efficient training by allowing the input of a layer block to be added to its output, facilitating residual learning. ResNet-50's architecture, which includes 1×1 and 3×3 convolution layers, not only addresses the vanishing gradient issue but also reduces model complexity and computational cost. This report evaluates ResNet-50's performance with the bag of tricks on the

CIFAR-10 dataset. Specifics about these tricks and their impact on the model's accuracy and training efficiency are detailed later in the report.

2.2. ViT-B/4

The Vision Transformer (ViT) adapts the Transformer architecture [12], initially used in NLP. It applies the Transformer's attention mechanism to image patches to analyze and classify images. ViT uses the Transformer's encoder with a classification token added during embedding. This token, alongside positional embeddings for each patch, passes through the encoder's two main blocks: one with Layer Normalization (LN), multi-head self-attention, and a residual connection, and another with LN, an MLP, and a residual connection. The study modifies the original ViT model which is introduced in [1], so that the modified ViT can be appropriately trained with the CIFAR-10 dataset. The patch size is adjusted to 4, embedding dimensions to 384, MLP size to 768, and numner of layers to 6. The modified model is called ViT-B/4. The report further evaluates the impact of various combinations of tricks when applied to the ViT-B/4 model.

3. Bag of tricks

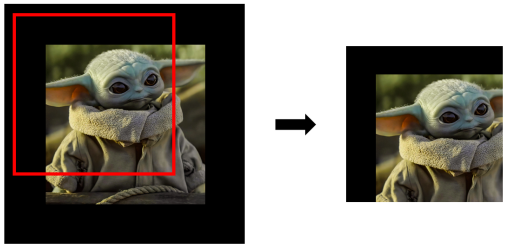
This section introduces the bag of tricks that are used for training and discusses how these methods can effect the training process in detail. The bag of tricks used in this experiment can be broadly classified into two groups. The first group consists of tricks used during the preprocessing stage. These are primarily techniques related to data augmentation. The second group comprises of various training strategies. These techniques are employed during the model's training process to enhance performance or to ensure stable learning.

3.1. Preprocessing

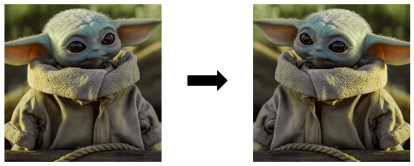
The tricks belonging to the preprocessing group include geometric transformation, normalization, and mixup.

3.1.1 Geometric transformation

The first preprocessing technique is geometric transformation. Geometric transformation involves resizing, cropping, rotating, and flipping the original images. The reason for applying these transformations is to perform data augmentation on the original images. Data augmentation is used to prevent machine learning models from overfitting. Overfitting occurs when the model is too well trained on the training dataset, which can lead to reduced generalizability when encountering unseen data. Data augmentation adds variation to the original data samples, preventing the model from overfitting to the training dataset. Perez et al. [9] reported that, when experimenting with a part of the ImageNet [10] dataset, geometric transformation was the most effective data augmentation method when training the model to find the best performance.



(a) Random crop. The black boundary that is around the original image is the padding and the red line is the desired crop size.



(b) Random horizontal flip.

Figure 1. Geometric transformations that are applied during the preprocessing step.

In this report, two types of geometric transformations are applied. The first is random crop, and the second is random horizontal flip. The random crop method, depicted in Fig. 1a, adds padding around the original image then cuts the padded image to a specified size. The random horizontal flip method, illustrated in Fig. 1b, randomly flips the original image horizontally. These data augmentation techniques were employed to attempt to enhance the model's generalizability to new data.

3.1.2 Normalization

Normalization is used to adjust the distribution of data to enhance the stability and efficiency of the model's training process. While image data typically ranges from 0 to 255, normalizing it to values between 0 and 1 aids in more effective model training. Without normalization, the scale of the input data can become too large or too small, leading to exploding or vanishing gradient problems during training. This can cause the model's output to rely excessively on larger values in the training data, leading to overfitting, or conversely, to underfitting issues with relatively smaller values.

This experiment performs normalization before training the model on the CIFAR-10 dataset. This process involves calculating the mean and standard deviation for each of the RGB channels in the training dataset. The calculated values were: mean = $(0.4914, 0.4822, 0.4465)$ and standard deviation = $(0.2023, 0.1994, 0.2010)$. These values are used for normalizing both the model's training and testing datasets.

Figure 2. An example of mixup, applied to alien and human images, when γ is set to 0.3.

3.1.3 Mixup

In addition to transforming geometric structure of the data samples for data augmentation, there exists a method of combining two data samples for data augmentation. This technique, introduced by Zhang et al. [13], is known as "mixup". Fig. 2 demonstrates the result of applying mixup to an alien and human. After combining two images, their labels are also combined. The combined image can be calculated using Eq. (1), and the labels using Eq. (2).

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j \tag{1}$$

$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j \tag{2}$$

In these formulas, x_i represents the first image, x_j the second image, and y_i and y_j are the corresponding labels for the first and second images, respectively. The parameter γ, which determines the ratio of combining the two data samples, takes values between 0 and 1, generated from a beta distribution.

3.2. Training strategies

Among the training strategies are learning rate decay, learning rate warmup, regularization, and initialization techniques.

3.2.1 Learning rate decay

Deep learning models typically employ the mini-batch gradient descent algorithm during training. This algorithm updates the model's parameters in the direction that minimizes the loss for the given mini-batch of the data. A critical hyperparameter in this process is the learning rate, which determines the magnitude of the parameter updates. However, the optimal value for the learning rate varies with each mini-batch, depending on the batch size, current parameter values, and the distance to the optimal solution. I will introduce three approaches to set the learning rate: constant, step decay, and cosine decay.

Constant learning rate scheduling maintains the initial learning rate unchanged throughout the training. This approach may lead to unstable training, especially with small mini-batches where the data distribution can vary significantly between batches. As the number of epochs increases, the model may oscillate around the optimal solution without proper convergence. Constant learning rate scheduling is primarily used to check the functionality of a deep learning model after it is defined.

The second learning rate scheduling technique is step decay, which reduces the learning rate at pre-determined epochs. For the training of ResNet, the learning rate was multiplied by 0.1 every 30 epochs. Step decay allows for rapid initial learning with a larger learning rate, gradually reducing the learning rate as the model approaches the optimal solution, aiding in better convergence.

The final method is cosine decay which was first introduced by Loshchilov et al. [8]. This technique employs a cosine function to decrease the learning rate. The training starts with the initial learning rate, which gradually diminishes to zero towards the end of the training. Unlike step decay, cosine decay offers a continuous reduction in the learning rate.

3.2.2 Learning rate warmup

At the beginning of the training process, all parameters in a model are initialized with random values. He et al. [4] note that using a large learning rate during these initial stages can lead to numerical instability. Goyal et al. [2] suggest a gradual warmup approach. This method starts with a very small learning rate for the first few epochs and then gradually increases it to the initial learning rate. The strategy is designed to enhance the stability of the training process by preventing the premature occurrence of numerical instabilities that may result from large updates.

3.2.3 Regularization

In deep learning, if the parameters of a model become excessively large, there is a risk that the results will depend too heavily on these inflated values. This can lead to overfitting, where the model becomes overly reliant on specific features of the data. Regularization is a concept that imposes constraints on the model to prevent overfitting. This report introduces three regularization techniques: weight decay, dropout, and label smoothing.

Weight Decay is one of the simplest regularization techniques, applying a penalty during the learning process to prevent the weights from growing too large. To implement this, either the L1 norm or the L2 norm of the weights is added to the loss function, thus preventing excessive increase in the weight values.

Dropout, a technique that randomly deactivates neurons with a probability between 0 and 1. This technique was first introduced by Hinton et al. [5]. Fig. 3 illustrates the changes in the structure of a Fully Connected Network (FCN) when assuming a dropout rate of 0.5. As shown in the figure,

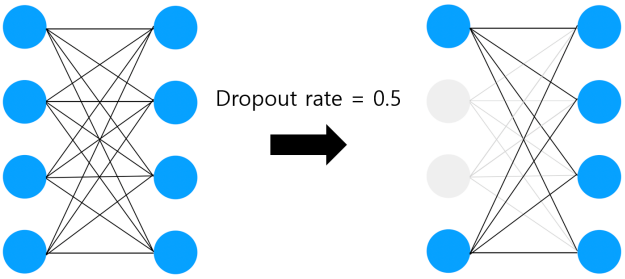


Figure 3. An example of dropout technique that is applied to a Fully Connected Network.

using dropout allows the model's parameters to learn various architectures, thereby reducing their dependence on any specific feature during training. Dropout is not only known to prevent overfitting but is also effective in averting underfitting. Liu et al. [7] reported that the use of dropout in various vision tasks can consistently improve generalization accuracy.

Label smoothing, a technique proposed by Szegedy et al. [11], refers to the process of transforming hard labels into soft labels. In image classification datasets, the default labels typically assign a probability of "1" to the correct class and "0" to all others. However, when label smoothing is applied, the probability for the correct class is reduced, while the incorrect classes are assigned with a small, equally distributed non-zero probability.

3.2.4 Initialization

When training deep learning models, appropriately assigning the initial values is a crucial step. Various methods exist for choosing the initial parameters, and this report will focus on the initialization method of the γ value used in Batch Normalization (BN). Zero Initialization, introduced in [4], involves setting the γ value of the last BN layer in a residual block to 0. This approach results in all residual blocks initially outputting values only through residual connections. According to He et al. [4], zero initialization emulates a network with fewer layers in the initial stages of training, thereby facilitating learning with a simpler model. The ViT-B/4 does not have a BN layer. Instead, it uses Layer Normalization (LN). Therefore, the γ values of the LN layers in a encoder block is set to 0 in the beginning.

3.3. Combinations of tricks

3.3.1 ResNet-50

For ResNet-50, 9 different combinations were used to maximize the models' performance, as detailed in Tab. 1. The "Basic" group uses random crop, random horizontal flip, data normalization, and Stochastic Gradient Descent (SGD). The terms "Const", "Step", and "Cos" stand for constant learning rate scheduling, step decay, and cosine decay, respectively. "Warm", and "Mix" indicate learning rate warmup, and mixup method. The terms "Drop" and "Zero" represent dropout and zero initialization. The baseline model for ResNet-50 uses two tricks which are basic, and cosine decay.

The implementation details for the tricks are as follows: For random cropping, padding value of 4 is applied, then the padded images are cropped to (32, 32). With SGD, the initial learning rate was 0.1, momentum 0.9, and weight decay 5e-4. Step decay approach involves multiplying the learning rate by 0.1 every 30 epochs, and for cosine decay, T is set to 90. A gradual warmup was used in the first 5 stages of the warmup technique, and the dropout rate was set at 0.5 for dropout. All experiments were conducted for 90 epochs.

3.3.2 ViT-B/4

ViT-B/4 was trained using 6 different combinations, as outlined in Tab. 2. These methods are 'Basic', 'Cos', 'Warm', 'Mix', 'Zero', and 'Drop', each corresponding to the techniques utilized in ResNet-50's training. The term 'Label' denotes the use of label smoothing. Unlike in the ResNet-50 training experiments, ViT-B/4 did not employ constant learning rate scheduling or step decay. This decision was based on the observation that cosine decay yielded better stability and performance than these methods in the ResNet-50 experiments. Basic and cosine decay are used for training the ViT-B/4 model.

The implementation details for these 6 methods are as follows: The basic group replicates the techniques used in ResNet-50's training. Similarly, cosine decay, warmup, and mixup use the same hyperparameters as those in ResNet-50's experiment. However, for the dropout method, a dropout rate of 0.1 was used. Dropout was applied at both the multi-head self-attention and MLP block.

4. Experiments

4.1. CIFAR-10 dataset

In this study, the CIFAR-10 dataset was used to train ResNet-50 and ViT-B/4. CIFAR-10 is a widely-used dataset in computer vision research, consisting of 60,000 small color images belonging to 10 different categories. These categories include airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks. The dataset is divided into 50,000 training images and 10,000 test images, each with a resolution of 32×32 pixels. Training dataset was not separated to create a validation dataset. All results are based on evaluations using the 10,000 test images.

Combination	Basic	Const	Step	Cos	Warm	Mix	Drop	Zero
Basic + Const (Baseline)	✓	✓						
Basic + Step	✓		✓					
Basic + Cos	✓			✓				
Basic + Cos + Warm	✓			✓	✓			
Basic + Cos + Mix	✓			✓		✓		
Basic + Cos + Warm + Mix	✓			✓	✓	✓		
Basic + Cos + Warm + Mix + Drop	✓			✓	✓	✓	✓	
Basic + Cos + Warm + Mix + Zero	✓			✓	✓	✓		✓
Basic + Cos + Warm + Mix + Drop + Zero	✓			✓	✓	✓	✓	✓

Table 1. Combination of different tricks used to train ResNet-50.

Combination	Basic	Cos	Warm	Label	Mix	Zero	Drop
Basic + Cos (Baseline)	✓	✓					
Basic + Cos + Label	✓	✓		✓			
Basic + Cos + Warm + Label	✓	✓	✓	✓			
Basic + Cos + Warm + Label + Mix	✓	✓	✓	✓	✓		
Basic + Cos + Warm + Label + Mix + Zero	✓	✓	✓	✓	✓	✓	
Basic + Cos + Warm + Label + Mix + Drop	✓	✓	✓	✓	✓		✓

Table 2. Combination of different tricks used to train ViT-B/4.

4.2. ResNet-50 results

Tab. 3 displays the performance and training duration of ResNet-50 when trained using 9 distinct combinations shown in Tab. 1. Notably, the judicious selection of technique combinations could enhance performance by up to 9.77% compared to the baseline model.

An overarching trend observed from the results is that employing a greater number of techniques typically results in better performance enhancement. Among the learning rate scheduling methods evaluated, cosine decay outperformed both the constant learning rate scheduling and step decay. Through this, it can be confirmed that gradually decreasing the learning rate following a differentiable curve is the most effective learning scheduling technique.

Interestingly, the use of warmup in conjunction with the basic and cosine decay methods resulted in a decrease in performance. However, when combined with the mixup technique, warmup significantly improved the accuracy. Through these results, we can see that the data augmentation technique called mixup is an effective trick in increasing the model's generalizability.

In scenarios where basic, cosine decay, warmup, and mixup were implemented together, of the two regularization techniques, dropout, and zero, employing zero alone yielded the most favorable results which was 94.79%. This means that not all regularization techniques have a positive impact on performance enhancement. However, initializing the γ of the BN layer to 0 has been found to significantly contribute to model performance improvement. From Fig. 4, which is the testing accuracy throughout the training process, it can be seen that among the regularization techniques, using zero with other tricks results in the better performance in the early stages of training compared to just using the other tricks without zero.

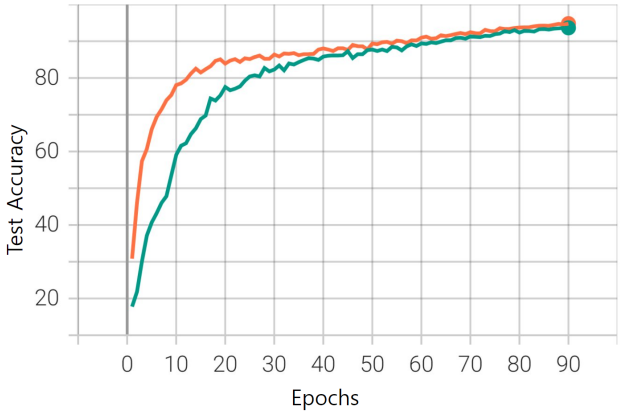


Figure 4. Comparison of ResNet-50's testing accuracy throughout the training process. The orange graph represents results from training with basic, cosine decay, warmup, mixup, and zero techniques. The green graph shows results from training with the zero technique excluded.

When analyzing the training time results, an intriguing observation emerged: the combination yielding the high-

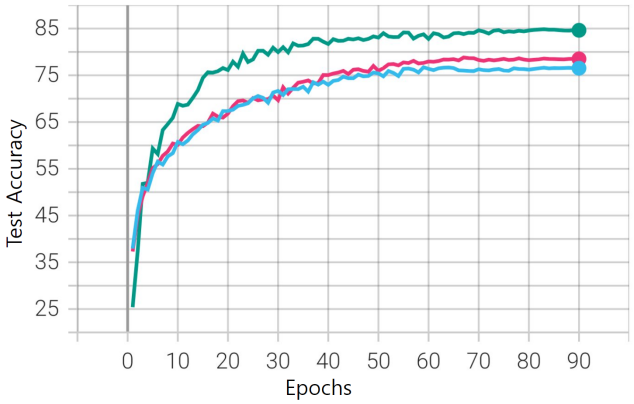


Figure 5. Comparison of ViT-B/4's testing accuracy throughout the training process. The green graph represents results from training with basic, cosine decay, warmup, and label smoothing techniques. The pink graph shows results from training with the warmup technique excluded. While the blue graph is trained with the warmup, and label smoothing techniques removed.

est performance, that involves basic, cosine decay, warmup, mixup, and zero, required less time than the baseline model, which employed only basic and cosine decay. This finding piqued interest and warrants further exploration. A more detailed discussion of these time-related findings will be presented in Sec. 5.

4.3. ViT-B/4

Tab. 4 displays the performance and training duration for ViT-B/4, trained using 6 different combinations. The baseline model of ViT-B/4 achieved a performance of 76.73%. Remarkably, with the mere addition of learning rate warmup and label smoothing, the performance escalated to 84.86%. When observing Fig. 5, it can be seen that adding warmup and label smoothing techniques initially results in lower performance compared to the other two combinations which do not use them all. However, the performance soon increases much more rapidly. It is evident that using these two combinations is highly effective when training ViT-B/4.

The best performance of 86.87% was attained by incorporating the mixup method to the basic, cosine decay, warmup, and label smoothing combined. This demonstrates that the mixup trick, is not only highly effective for the CNN-based model ResNet-50, but also for the Transformer-based model ViT-B/4.

Conversely, the application of the two distinct regularization techniques, zero and dropout, led to a diminishment in performance. It can be observed that regularization techniques do not always enhance the model's performance.

Regarding training time, the most rapid completion was observed with the basic and cosine decay combination, whereas the incorporation of warmup, label smoothing, mixup, and dropout extended the duration the most.

5. Discussion

5.1. Is regularization an effective trick?

The main purpose of using regularization is to prevent overfitting. However, in some cases, the performance was better when not using some regularization methods.

5.1.1 Regularization in ResNet-50

When dropout was used for trianing ResNet-50, the final performance decreased. The ResNet-50 model I used had a dropout layer right before the classification head, which is the final part of the model. However, this approach might prevent the model from correcting errors caused by dropout. I believe that this might have been the cause for the performance drop. If dropout had been applied somewhere else, *e.g.* at the beginning stages of the model, the results might have been different.

5.1.2 Regularization in ViT-B/4

When training the ViT-B/4 model, it was found that dropout degraded the model's performance. Particularly in the ViT-B/4 model, dropout was applied after Layer Normalization (LN). Applying dropout to the data that has been normalized by LN can alter the distribution of the features optimized through LN. For this reason, I believe that dropout had a negative impact on the results when used applied after LN.

Not only drop, but the zero technique, dimished the performance of ViT-B/4. The accuracy was found to be 6.68% lower than when it was trained without using zero. This was a surprising result, since applying zero to the ResNet-50 training had led to performance improvements.

The reason for such an outcome is believed to be related to the attention mechanism of the Transformer. In the multi-head self-attention of ViT, attention scores are first calculated. However, if the values of γ start at zero, the attention scores might not be effectively calculated. This is because the outputs of Layer Normalization (LN) are very small due to γ values being near 0. Fig. 6 shows the average values of γ throughout the training process. It can be observed that when ViT-B/4 is trained using the zero strategy, the γ values start very small from the beginning of the training. Even at the later stages of training, the converging values of γ are smaller when zero technique is used compared to when it is not. Thus, it is thought that the attention function might not have been properly executed due to these small γ values when training with the zero method.

Combination	Accuracy (%)	Time (sec/epoch)
Basic + Const (Baseline)	85.02	23.77
Basic + Step	88.04	25.02
Basic + Cos	90.87	24.07
Basic + Cos + Warm	90.49	23.72
Basic + Cos + Mix	92.87	24.78
Basic + Cos + Warm + Mix	93.64	25.44
Basic + Cos + Warm + Mix + Drop	93.45	23.66
Basic + Cos + Warm + Mix + Zero	**94.79**	23.61
Basic + Cos + Warm + Mix + Drop + Zero	94.51	24.48

Table 3. ResNet-50's performance and time required to train for various combinations.

Combination	Accuracy (%)	Time (sec/epoch)
Basic + Cos	76.73%	8.43sec/epoch
Basic + Cos + Label	78.81%	8.47sec/epoch
Basic + Cos + Warm + Label	84.86%	8.47sec/epoch
Basic + Cos + Warm + Label + Mix	**86.87%**	8.47sec/epoch
Basic + Cos + Warm + Label + Mix + Zero	80.19%	8.51sec/epoch
Basic + Cos + Warm + Label + Mix + Drop	85.96%	9.18sec/epoch

Table 4. ViT-B/4's performance and time required to train for various combinations.

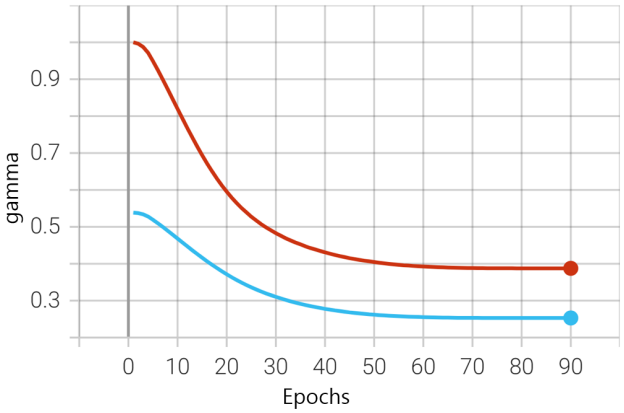


Figure 6. Comparison of the average gamma γ value in ViT-B/4, contrasting its training with and without the zero technique. The blue graph represents training with basic, cosine decay, warmup, label smoothing, mixup, and zero, while the red graph indicates training with the same techniques except for zero.

5.2. Are the results reliable?

ResNet-50 utilized 9 different bag of tricks combinations, while ViT-B/4 used 6. Each combination, though, was only trained a 1 time. This might be why some of the results appeared illogical. For instance, as seen in Tab. 3, the training duration for the combination using only basic and constant learning rate scheduling exceeded that of the combination which incorporated warmup, mixup, and dropout. Various factors could have influenced these unexpected outcomes, but repeating the experiments multiple times and averaging the results could potentially clarify them.

6. Conclusion and future work

In this report, I examined how various combinations of "bag of tricks" enhance the performance in training ResNet-50 and ViT-B/4. It was observed that stacking most of these tricks generally improves performance. Techniques like geometric transformation, data normalization, and mixup, which belong to the data preprocessing category, have typically been confirmed to boost model performance. However, we also noted that tricks aimed at preventing overfitting, such as regularization techniques, are only beneficial when used appropriately. Therefore, this report emphasizes the importance of using an appropriate combination of bag of tricks when training deep neural networks, as determined through empirical studies.

In the future, the focus will be on conducting the same experiment multiple times to enhance the reliability of the experiments. Additionally, there will be efforts to improve the performance of the models by applying regularization techniques in different ways, *e.g.* applying dropout at the Convolutional Layers. Finally, I am interested in verifying whether backbones like ResNet-50 or ViT-B/4 trained using the bag of tricks, positively impact the performance in more practical computer vision tasks such as object detection or semantic segmentation.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2

[2] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 3

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[4] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 558–567, 2019. 1, 3, 4

[5] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012. 3

[6] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1

[7] Zhuang Liu, Zhiqiu Xu, Joseph Jin, Zhiqiang Shen, and Trevor Darrell. Dropout reduces underfitting. *arXiv preprint arXiv:2303.01500*, 2023. 4

[8] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 3

[9] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017. 2

[10] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. 2

[11] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 4

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1, 2

[13] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 3