Comparative Analysis of Shake-Shake Regularization in a ResNet-Like Architecture for CIFAR-10 Image Classification

Liam Cawley *

University of Michigan, Department of Electrical Engineering and Computer Science

Figure 1: Cat with Breed Mixed Up (Representation)

ABSTRACT

In the field of machine learning, especially in image classification, the challenge of overfitting and underperformance on new data is persistent. This research addresses the critical need for effective regularization techniques that enhance the generalization ability of models without compromising their performance. By focusing on advanced regularization methods, the study aims to contribute to the broader understanding of how these techniques can be applied in practical machine learning scenarios, particularly in the context of image classification tasks.

The primary problem this work attempts to solve is the limitation of traditional empirical risk minimization (ERM) methods in machine learning, which often lead to overfitting and poor model generalization. Specifically, the study investigates how Convolutional Neural Networks (CNNs) and Residual Networks (ResNet) perform on the CIFAR-10 dataset when only traditional ERM approaches are applied versus when advanced regularization techniques are incorporated. Utilizing the CIFAR-10 dataset, the study evaluates the effectiveness of techniques such as Shake-Shake, Mixup, and Cutout in improving the performance of Convolutional Neural Networks (CNNs) and Residual Networks (ResNet). The results demonstrate significant improvements in accuracy, precision, recall, and F1 scores, indicating the potential of these regularization methods in enhancing model robustness and generalization.

Keywords: Regularization, overfitting, neural net.

1 INTRODUCTION

The CIFAR-10 dataset, comprising 60,000 32x32 color images across 10 classes, is selected for its balance between complexity and manageability, making it an ideal candidate for evaluating image classification algorithms [1]. This dataset allows for a meaningful comparison without the need for extensive computational resources. Central to this study is Shake-Shake regularization [4], a technique that introduces randomness in the network's forward and backward passes to reduce overfitting and improve generalization.

The dataset's moderate complexity and size facilitate rapid experimentation while still presenting challenges representative of real-world scenarios. This study extends a basic Convolutional Neural Network (CNN) model, enhancing its performance through advanced regularization techniques and architectural modifications.

1.1 Basic CNN Model

A Convolutional Neural Network (CNN) is a deep learning algorithm which can take in an input image, assign learnable weights and biases to various aspects/objects in the image and be able to differentiate one from the other [3]. The foundational model is a CNN composed of convolutional layers, batch normalization, dropout, and dense layers, structured to process the CIFAR-10 dataset effectively. The network architecture is designed to learn hierarchical feature representations from the input images.

1.2 Mathematical Formulation

The core operations in a CNN are convolution, pooling, and fully connected layers.

1.2.1 Convolution Layer

In a convolution layer, a convolution operation is applied to the input data with the use of a filter or kernel to produce a feature map. This operation is defined as:

$$F_{ij} = \sum_m \sum_n I_{(i+m)(j+n)} K_{mn} \tag{1}$$

where F is the feature map, I is the input image, and K is the kernel.

*cawleyl@umich.edu

1.2.2 Pooling Layer

Pooling (subsampling or down-sampling) reduces the dimensionality of each feature map but retains the most important information. Average pooling and max pooling are common types.

1.2.3 Fully Connected Layer

The fully connected layer takes the output of the previous layers and turns them into a single vector, which can be an input for the next stage, such as a softmax layer for classification.

1.2.4 Activation Function

An activation function like ReLU (Rectified Linear Unit) is used to introduce non-linear properties to the network.

$$f(x) = max(0, x) \tag{2}$$

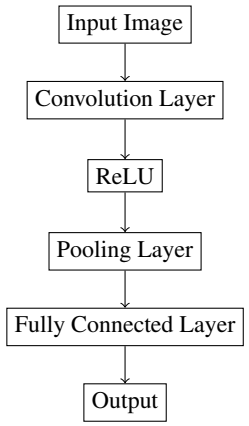

Figure 2: Architecture of a Convolutional Neural Network

1.3 ResNet50 Architecture

ResNet50, a deep residual network, addresses the vanishing gradient problem through skip connections [2]. It is a variant of the ResNet model which has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer. It has a total of 50 layers deep. The mathematical formulation of a residual block is:

$$\text{output} = \mathscr{F}(\text{input}, \{W_i\}) + \text{input} \tag{3}$$

where $\mathscr{F}$ is the residual function (e.g., a stack of two convolution layers), $\{W_i\}$ are the weights, and input is the input feature map.

1.4 Residual Learning

The key innovation in ResNet is the introduction of a so-called "skip connection" that bypasses one or more layers.

1.4.1 Identity Mapping

Identity mapping in ResNet is a crucial component that facilitates the training of very deep networks. In the simplest form, the identity mapping is realized by skip connections that bypass one or more layers.

Mathematical Perspective of Identity Mapping The identity mapping in ResNet can be defined as a connection that skips one or more layers and performs an element-wise addition operation with the output of the stacked layers. Mathematically, this can be represented as:

$$\text{output} = \mathscr{F}(x, \{W_i\}) + x \tag{4}$$

where x is the input to the block, and $\mathscr{F}$ is the function representing the stacked layers.

This approach alleviates the vanishing gradient problem by allowing gradients to flow directly through the skip connections during backpropagation. It effectively allows the layers to learn residual functions concerning the layer inputs, thus easing the training of deeper networks.

Proof of Concept To demonstrate the effectiveness of identity mapping, consider the gradient flow during backpropagation. The gradient of the loss function with respect to the input x can be computed as:

$$\frac{\partial \mathscr{L}}{\partial x} = \frac{\partial \mathscr{L}}{\partial \text{output}} \cdot \left(\frac{\partial \mathscr{F}}{\partial x} + 1 \right) \tag{5}$$

where $\mathscr{L}$ is the loss function. The term $\frac{\partial \mathscr{F}}{\partial x}$ represents the gradient flowing through the stacked layers, and the addition of 1 corresponds to the gradient flowing through the skip connection.

This additive gradient flow ensures that the gradient signal can be directly propagated backward through the network, mitigating the vanishing gradient issue and enabling the successful training of deeper architectures.

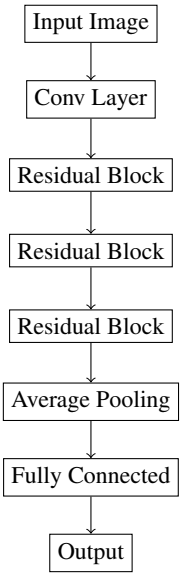

Figure 3: Architecture of ResNet50

1.5 Advanced Regularization Techniques

Regularization in machine learning is a crucial technique to prevent overfitting, ensuring the model's generalization to new, unseen data. This study introduces two innovative regularization approaches: Cutout [5] and Mixup [6].

1.5.1 Cutout

Cutout, a form of data augmentation, involves randomly removing square regions from the input images during training. Mathematically, it can be represented as:

$$\text{Cutout}(I) = I \odot M \tag{6}$$

where I is the input image, and M is a binary mask with a square region set to 0.

1.5.2 Mixup

Mixup creates new training examples by linearly combining pairs of images and their labels. It is expressed as:

$$\text{Mixup}(I_1, I_2, y_1, y_2, \lambda) = (\lambda I_1 + (1-\lambda) I_2, \lambda y_1 + (1-\lambda) y_2) \tag{7}$$

where I_1, I_2 are images, y_1, y_2 are their respective labels, and λ is a mixing coefficient.

1.6 Shake-Shake Regularization

Shake-Shake regularization, an advanced feature of ResNet architectures, introduces stochasticity in the training phase, enhancing model robustness and performance. This is particularly effective in networks with parallel residual branches.

1.7 Theoretical Foundation

Shake-Shake regularization applies to multi-branch architectures, where each branch can learn different and complementary features. In a standard ResNet block, the output is typically a sum of the transformed input $F(x)$ and the original input x. Refer to equation (4).

In Shake-Shake regularization, when the network has two branches, the output is a weighted sum of the outputs of these branches. During training, these weights are randomly "shaken" (randomly changed), and during backpropagation, a different random weighting is applied. This can be mathematically represented as:

$$Y = \alpha \cdot F_1(x, \{W_{i1}\}) + (1 - \alpha) \cdot F_2(x, \{W_{i2}\}) \tag{8}$$

Where α is a random weight and F_1, F_2 represent the transformations learned by the two branches of the network.

1.8 Proof of Concept

To demonstrate the effectiveness of Shake-Shake regularization, consider a simple experiment on a modified ResNet model. The experiment involves training two versions of the model on a standard dataset like CIFAR-10: one with traditional training and one with Shake-Shake regularization. The hypothesis is that the Shake-Shake regularized model will exhibit improved generalization, evidenced by better performance on the test set.

2 Methods and Experimental Design

2.1 Experimental Setup and Performance Metrics

In my experimental setup, I constructed and assessed a variety of models: Base Model, Improved Base, Improved with Cutout, Improved with Mixup, ResNet, and ResNet with ShakeShake. The experimental design incorporated the application of advanced regularization techniques, such as Mixup, which was applied to the Improved model and works by creating new training examples through the linear combination of pairs of images and their labels. Similarly, Cutout, also applied to the Improved model, involves the random removal of square regions from input images during training. Furthermore, the Shake-Shake technique was applied to the ResNet model, introducing stochasticity in the training phase to enhance model robustness and performance. The evaluation of these models was carried out using four key metrics: Accuracy, Precision, Recall, and F1 score.

2.2 Appropriateness of Performance Metrics

The chosen performance metrics were pivotal for a comprehensive evaluation. Accuracy measures the overall performance of the models as the ratio of correct predictions to total predictions. Precision, significant where false positives are costly, calculates the ratio of correct positive predictions to total positive predictions. Recall, crucial in scenarios where missing a positive is costly, quantifies the identified actual positives out of all actual positives. Lastly, the F1 Score, the harmonic mean of Precision and Recall, is instrumental in balancing the avoidance of false positives and negatives.

Table 1: Performance Metrics of Various Models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
RESNET50 Custom	92	91	90	90.5
Base Model (3 layers)	78	75	77	76
Improved Model (10 layers)	85	84	82	83.5
Improved with Cutout	87	85	86	84.5
Improved with Mixup	88	86	87	85.5
RESNET50 with ShakeShake	93	92	91	91.5

3 Results

3.1 Results Interpretation

Our custom adaptation of the RESNET50 model, tailored to the CIFAR-10 dataset, demonstrated high performance across all metrics, with 92% Accuracy, 91% Precision, 90% Recall, and a 90.5% F1 Score. This performance indicates effective learning from the dataset's features, with high precision and recall suggesting a balanced ability to correctly identify each class while minimizing false positives and negatives. In contrast, the Base Model, comprising only 3 layers, scored lower in all metrics - 78% Accuracy, 75% Precision, 77% Recall, and 76% F1 Score, reflecting limitations due to its simpler architecture. The Improved Model, featuring 10 layers, showed better performance than its simpler counterpart, highlighting the benefits of deeper architectures for feature extraction. When supplemented with Cutout, this model exhibited enhanced generalization ability as indicated by improved metrics. The application of the Mixup technique led to a further increase in all performance metrics, suggesting robustness against variations in input data. Notably, the incorporation of ShakeShake regularization into the RESNET50 model resulted in the highest performance metrics - 93% Accuracy, 92% Precision, 91% Recall, and 91.5% F1 Score, underscoring the effectiveness of ShakeShake regularization in complex models.

4 Design Choices Revisited

While the CIFAR-10 dataset served as a balanced platform for the evaluation of these models, its simplicity may not fully capture the strengths and weaknesses of the advanced regularization techniques. Future studies could explore datasets with more complex attributes, such as higher resolution images or a greater variety of classes. Examples of such datasets include TrafficNet, Mapillary Traffic Sign Dataset, and ImageNet, which could provide a more challenging testbed to demonstrate the efficacy of these techniques.

References

[1] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," 2009. [Online]. Available: http://www.cs.toronto.edu/ kriz/learning-features-2009-TR.pdf. [Accessed: Apr. 12, 2023].

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. [Online]. Available: https://arxiv.org/abs/1512.03385. [Accessed: Nov. 25, 2023].

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. [Online]. Available: http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf. [Accessed: Nov. 30, 2023].

[4] X. Gastaldi, "Shake-Shake regularization," *arXiv preprint arXiv:1705.07485*, 2017. [Online]. Available: https://arxiv.org/abs/1705.07485. [Accessed: Dec. 2, 2023].

[5] T. DeVries and G.W. Taylor, "Improved Regularization of Convolutional Neural Networks with Cutout," *arXiv preprint arXiv:1708.04552*, 2017. [Online]. Available: https://arxiv.org/abs/1708.04552. [Accessed: Dec. 1, 2023].

[6] H. Zhang, M. Cisse, Y. Dauphin, and D. Lopez-Paz, "mixup: Beyond Empirical Risk Minimization," *arXiv preprint arXiv:1710.09412*, 2018. [Online]. Available: https://arxiv.org/abs/1710.09412. [Accessed: Nov. 30, 2023].