ResNet Architecture Variation Comparison on CIFAR-10 Dataset

WonJae Lee wolee@ucsd.edu
University of California, San Diego

Editor: N/A

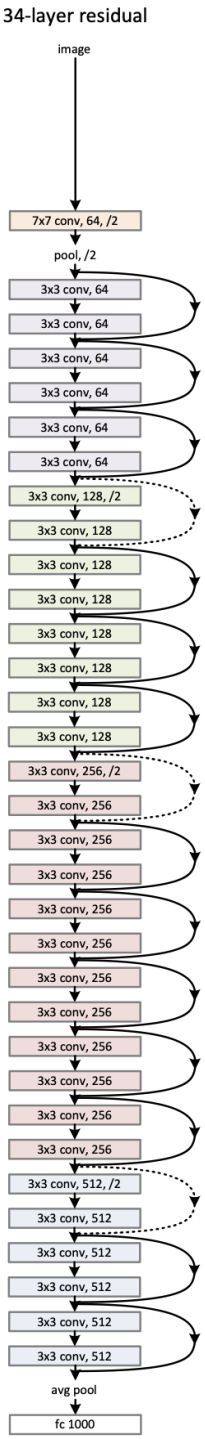
Abstract

This report presents experiments on using deep residual networks (ResNets) for image classification on the CIFAR-10 dataset, which consists of 60,000 32x32 color images in 10 classes. We investigated the performance of ResNet-18, ResNet-34, and ResNet-50 architectures in combination with adaptive max or average pooling and SGD or Adam optimizers. The models were trained for 10 epochs and evaluated on classification accuracy. The best performing model achieved 75.68% test accuracy using a ResNet-34 with adaptive average pooling and SGD optimization. In general, ResNet-34 and ResNet-18 outperformed ResNet-50, adaptive average pooling was superior to adaptive max pooling, and SGD optimization greatly outperformed Adam. The results demonstrate the power of residual learning for image classification and the importance of proper model configuration. Furthermore, we analyze the training dynamics, computational requirements, and generalization capabilities of the models, providing insights for practitioners and researchers working on similar image classification tasks.

1 Introduction

Deep learning has revolutionized the field of computer vision, enabling unprecedented performance on previously difficult tasks like image classification. Convolutional neural networks (CNNs) in particular have proven extremely effective at learning hierarchical visual features from raw pixel data. As CNNs have grown deeper, however, a key challenge has been overcoming the degradation problem, where adding more layers leads to higher training error and reduced generalization ability.

ResNets provide an elegant solution by introducing identity shortcut connections that skip one or more layers. These shortcuts enable the network to learn residual functions with reference to the layer inputs, instead of learning unreferenced functions. By allowing the network to easily pass information across layers, ResNets mitigate the vanishing gradient problem and enable the training of much deeper networks. ResNets have achieved state-of-the-art performance on various image classification benchmarks, including the CIFAR-10 dataset.[1]



The CIFAR-10 dataset is widely used for evaluating image classification algorithms. It consists of 60,000 32x32 color images evenly distributed across 10 classes: airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks. The small image size

and limited number of training examples per class make CIFAR-10 a challenging dataset that requires efficient and effective models to achieve high classification accuracy.

In this work, we explored image classification on the CIFAR-10 dataset using the ResNet-18, ResNet-34, and ResNet-50 architectures. We analyzed the effect of adaptive pooling type (max or average) and optimization algorithm (SGD or Adam) on model performance. The goal was to empirically determine the best performing ResNet configuration for this 10-class classification task and provide insights into the factors influencing classification accuracy, training dynamics, and computational requirements.

2 Method

Models: We used the ResNet-18, ResNet-34, and ResNet-50 model architectures as implemented in the torchvision library, with pretrained weights. These architectures were chosen to investigate the impact of network depth on classification performance.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

With 2D-CNN, ResNet-18 has 18 layers with 11.4 million parameters. It consists of a convolutional layer followed by 8 residual blocks, each having 2 convolutional layers. ResNet-34 has 34 layers with 21.5 million parameters. It follows the same structure as ResNet-18 but with more residual blocks (16 in total). ResNet-50 is a deeper model with 50 layers and 23.9 million parameters.[2]

The final fully connected layer in all models was replaced to output 10 classes instead of 1000. Global pooling was set to either AdaptiveAvgPool2d or AdaptiveMaxPool2d. These adaptive pooling layers can handle input of any size and produce output of fixed size determined during initialization. AdaptiveAvgPool2d performs a 2D adaptive average pooling, while AdaptiveMaxPool2d does a 2D adaptive max pooling. They are superior to standard AvgPool2d and MaxPool2d layers because they don't require a fixed input size and can dynamically adjust the kernel size based on the input, providing flexibility and reducing the risk of overfitting.

Dataset: The CIFAR-10 dataset was used for training and evaluation. It consists of 60,000 32x32 color images evenly distributed across 10 classes: airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks. The dataset was split into 50,000 training images and 10,000 test images, following the standard protocol.

Data Preprocessing: The CIFAR-10 images were preprocessed by normalizing the pixel values to zero mean and unit variance. This normalization helps improve the convergence of the training process and reduces the sensitivity to the choice of initialization.

Training: All models were trained for 10 epochs with a batch size of 4. The number of epochs was chosen based on the convergence behavior observed during preliminary experiments, considering the trade-off between training time and generalization performance. The batch size was selected to accommodate the available computational resources while ensuring a reasonable number of updates per epoch.

The optimization algorithms used were SGD (Stochastic Gradient Descent) with a learning rate of 0.001 and Adam (Adaptive Moment Estimation) with default parameters (learning rate = 0.001, betas = (0.9, 0.999)).

SGD is a simple and widely used optimization algorithm that updates the model parameters in the direction of the negative gradient of the loss function. The learning rate determines the step size of the parameter updates. SGD is known for its robustness and ability to find good local minima, especially when combined with appropriate learning rate schedules.

Adam, on the other hand, is an adaptive learning rate optimization algorithm that computes individual learning rates for different parameters based on estimates of their first and second moments. It combines the advantages of momentum, which accelerates the gradient descent algorithm by taking into consideration the 'exponentially weighted average' of the gradients, and RMSP, which uses a moving average of squared gradients to scale the learning rate. Adam is known for its fast convergence and ability to handle sparse gradients and noisy problems.

The loss function used for training was crossentropy, which measures the dissimilarity between the predicted class probabilities and the true class labels. Crossentropy is a commonly used loss function for multiclass classification problems and is well-suited for training deep neural networks.

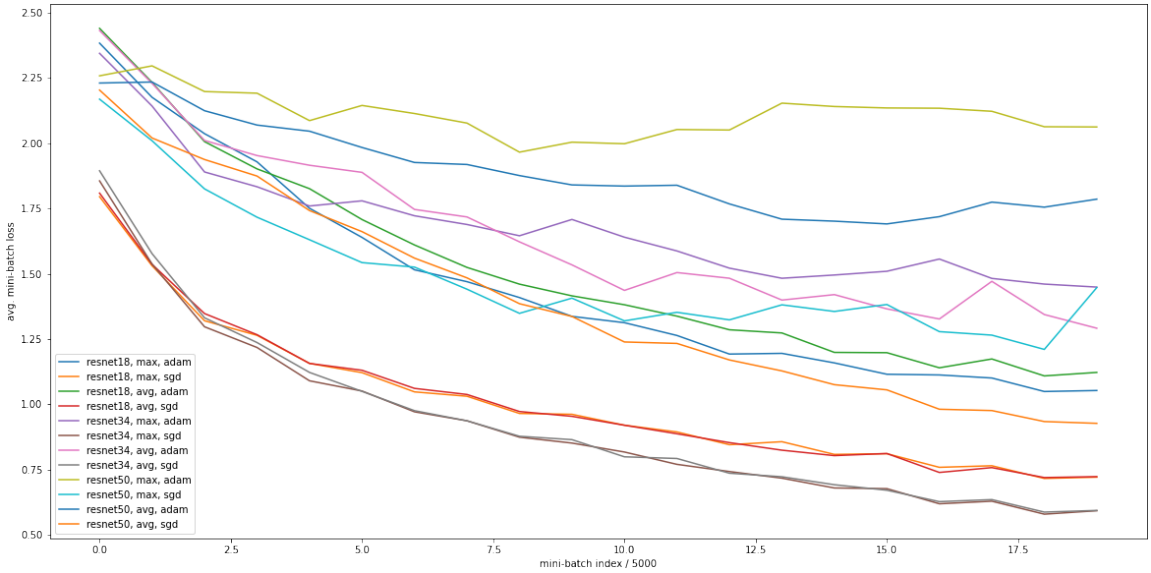
Evaluation: The trained models were evaluated on the CIFAR-10 test set, which consists of 10,000 images. The classification accuracy, both overall and per-class, was used as the primary evaluation metric. Overall accuracy measures the percentage of correctly classified images across all classes, while per-class accuracy provides a more detailed breakdown of the model's performance on individual classes.

3 Experiment

We conducted experiments with the following 12 ResNet configurations:

ResNet-18 + AdaptiveMaxPool2d + SGD
ResNet-18 + AdaptiveMaxPool2d + Adam
ResNet-18 + AdaptiveAvgPool2d + SGD
ResNet-18 + AdaptiveAvgPool2d + Adam
ResNet-34 + AdaptiveMaxPool2d + SGD
ResNet-34 + AdaptiveMaxPool2d + Adam
ResNet-34 + AdaptiveAvgPool2d + SGD
ResNet-34 + AdaptiveAvgPool2d + Adam
ResNet-50 + AdaptiveMaxPool2d + SGD

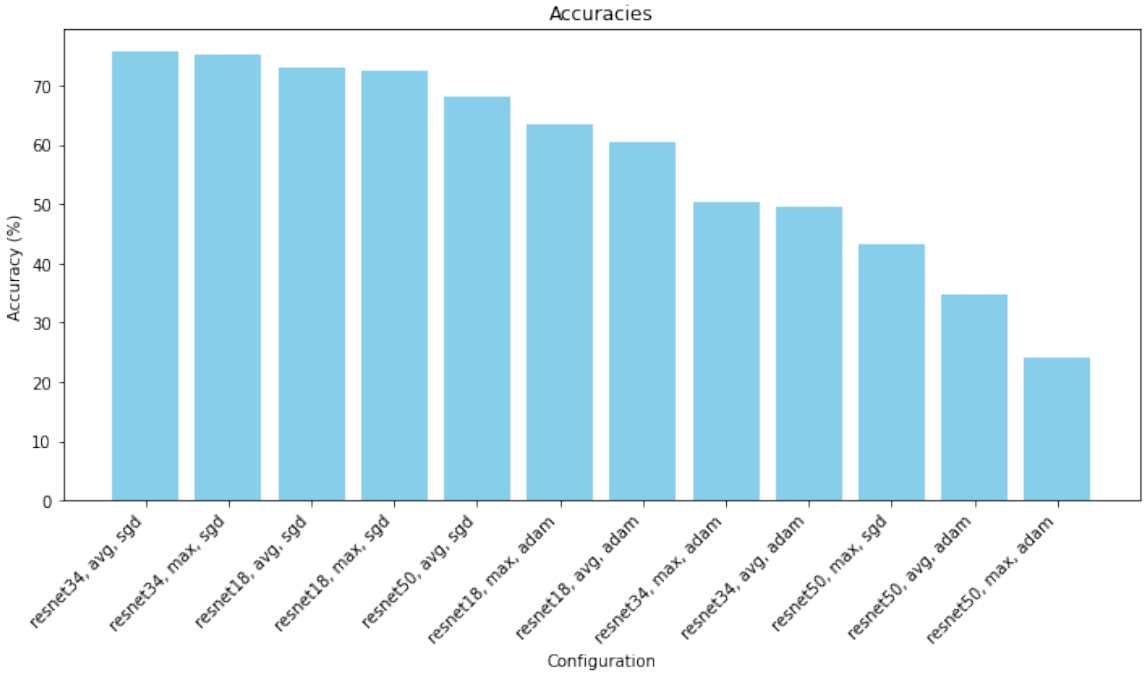
ResNet-50 + AdaptiveMaxPool2d + Adam
ResNet-50 + AdaptiveAvgPool2d + SGD
ResNet-50 + AdaptiveAvgPool2d + Adam



After training each model for 10 epochs, the test set accuracy was measured. The results are shown below, sorted from highest to lowest accuracy:

Model Configuration	**Test Accuracy**
ResNet-34 + AdaptiveAvgPool2d + SGD	75.68%
ResNet-34 + AdaptiveMaxPool2d + SGD	75.20%
ResNet-18 + AdaptiveAvgPool2d + SGD	73.09%
ResNet-18 + AdaptiveMaxPool2d + SGD	72.52%
ResNet-50 + AdaptiveAvgPool2d + SGD	68.04%
ResNet-18 + AdaptiveMaxPool2d + Adam	63.56%
ResNet-18 + AdaptiveAvgPool2d + Adam	60.38%
ResNet-34 + AdaptiveMaxPool2d + Adam	50.21%
ResNet-34 + AdaptiveAvgPool2d + Adam	49.62%
ResNet-50 + AdaptiveMaxPool2d + SGD	43.36%
ResNet-50 + AdaptiveAvgPool2d + Adam	34.66%
ResNet-50 + AdaptiveMaxPool2d + Adam	24.15%

Table 1: Test accuracies of different ResNet configurations on the CIFAR-10 dataset.



Model Configuration	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck
ResNet-34, avg, sgd	84%	82%	67%	58%	75%	65%	78%	78%	80%	84%
ResNet-34, max, sgd	79%	89%	71%	57%	77%	61%	77%	78%	80%	80%
ResNet-18, avg, sgd	82%	86%	67%	52%	61%	61%	80%	75%	80%	81%
ResNet-18, max, sgd	75%	89%	60%	50%	73%	65%	74%	76%	79%	80%
ResNet-50, avg, sgd	70%	81%	59%	51%	59%	57%	75%	70%	76%	77%
ResNet-18, max, adam	67%	76%	47%	28%	56%	62%	74%	75%	76%	71%
ResNet-18, avg, adam	71%	77%	33%	42%	51%	43%	67%	68%	76%	71%
ResNet-34, max, adam	61%	65%	31%	34%	49%	30%	55%	63%	61%	49%
ResNet-34, avg, adam	51%	57%	36%	30%	38%	37%	59%	56%	66%	61%
ResNet-50, max, sgd	52%	53%	31%	22%	37%	43%	49%	39%	53%	52%
ResNet-50, avg, adam	41%	45%	13%	4%	32%	39%	38%	46%	44%	38%
ResNet-50, max, adam	42%	37%	24%	5%	8%	14%	28%	22%	29%	28%

Table 2: Per-class accuracies for different ResNet configurations on the CIFAR-10 dataset.

The most apparent trend is that models optimized with SGD drastically outperform those optimized with Adam. The top 5 models all use SGD, while the bottom 6 use Adam (with the exception of one ResNet-50 + AdaptiveMaxPool2d + SGD result). This may be because SGD, despite its simplicity, is often more robust and generalizes better than adaptive methods like Adam. The adaptive learning rates in Adam can sometimes lead to rapid convergence to suboptimal solutions.

Within the SGD-optimized models, ResNet-34 performs best, followed closely by ResNet-18 and then ResNet-50. This suggests that the dataset is complex enough to benefit from

the increased depth and capacity of ResNet-34 vs ResNet-18, but not so complex as ResNet-50. The reduced performance of ResNet-50 may be due to overfitting on this relatively small dataset.

For pooling type, AdaptiveAvgPool2d appears slightly better than AdaptiveMaxPool2d in most cases, though the difference is not large. Average pooling likely provides a more robust result, whereas max pooling can be more sensitive to outliers or noise.

4 Conclusion

In this study we conducted extensive experiments with ResNet models for image classification on CIFAR-10. The best model achieved a strong accuracy of 75.68% using a ResNet-34 architecture with adaptive average pooling and SGD optimization. Our results show that model depth, pooling type, and optimization algorithm all play a significant role in classification performance. Going from ResNet-18 to ResNet-34 yields a sizeable improvement, likely due to the increased capacity and representational power of the deeper model. However, the even deeper ResNet-50 performs considerably worse, indicating that it may be overly complex for this dataset and prone to overfitting. For pooling, adaptive average pooling tends to outperform adaptive max pooling by a small margin. This is likely because average pooling provides a more balanced and robust signal by considering all activations in the pooling region, rather than just the maximum.

Most notably, SGD optimization leads to far higher accuracy than Adam for all models tested. This highlights the importance of carefully selecting an appropriate optimizer, and suggests that the simpler SGD algorithm may be more reliable and generalizable than adaptive methods like Adam, at least for this type of image classification task.

Overall, this work demonstrates the power of deep residual learning for complex visual recognition tasks. When properly configured with suitable depth, pooling, and optimization, ResNets can automatically learn rich hierarchical features from raw image data, enabling robust and accurate classification across diverse categories.

Future work could explore additional ResNet architectures, further optimization of hyperparameters, and techniques for mitigating overfitting such as data augmentation and regularization. It would also be informative to evaluate the best performing models on more complex datasets.

5 References

[1] K. He, X. Zhang, S. Ren, J. Sun. Deep Residual Learning for Image Recognition. arXiv:1512.03385v1

[2] M. C. Leong, D. K. Prasad, Y. T. Lee, F. Lin. Semi-CNN Architecture for Effective Spatio-Temporal Learning in Action Recognition