IMAGE CLASSIFICATION OF CIFAR-10 DATASET USING RESNET ARCHITECTURE

Sai Navyanth Penumaka[1], Karthik Sunkari[2], Geethika Rao Gouravelli[3]

sp8138@nyu.edu, ks7929@nyu.edu, gg2879@nyu.edu

New York University

Code: Github

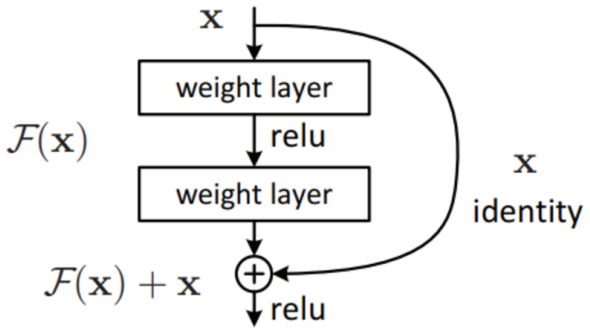
Abstract

In this project, we implemented a custom ResNet architecture with Squeeze-and-Excitation (SE) blocks to classify images from the CIFAR-10 dataset. ResNet, a well-known deep learning architecture, addresses the vanishing gradient problem by introducing residual connections, enabling the training of deeper networks. SE blocks further enhance the network's ability to recalibrate feature maps, allowing for better focus on relevant features. Our work incorporated a variety of techniques, including data augmentation, learning rate scheduling, and test-time augmentation (TTA), to maximize model performance and generalization. We performed 22 experiments with different architectures. The final model achieved a test accuracy of 94.10%, highlighting the effectiveness of SE blocks.

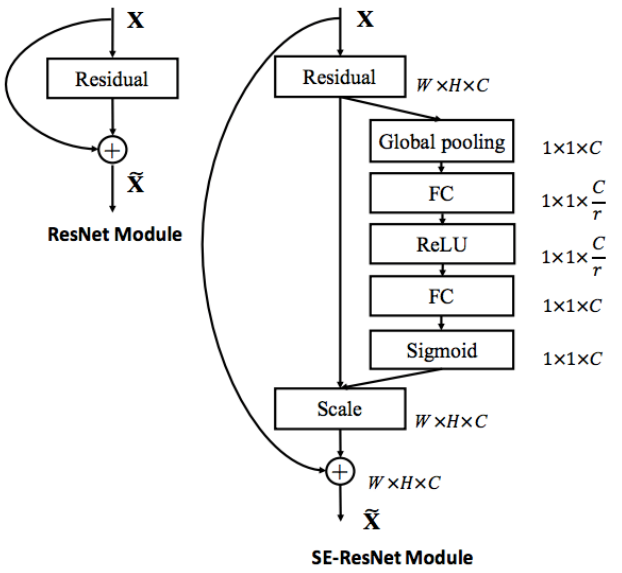
Introduction

Image classification is a fundamental task in computer vision, often used to benchmark the performance of deep learning models. One of the major challenges in training deep neural networks is the vanishing gradient problem, which occurs when gradients become increasingly small during backpropagation, hindering effective parameter updates in deeper architectures. ResNet, introduced by He et al. in 2015, addressed this issue by incorporating residual connections (also known as skip connections) into the network architecture. These residual connections allow the network to learn residual mappings rather than direct mappings, significantly mitigating gradient degradation and enabling the successful training of very deep networks. Consequently, ResNet achieved state-of-the-art performance across various image classification benchmarks.

The core idea behind ResNet's success lies in its unique architectural design. Each residual block within a ResNet model contains convolutional layers followed by batch normalization and activation functions, typically ReLU (Rectified Linear Unit). Crucially, these blocks introduce identity-based skip connections that directly add the input of a block to its output. Formally, instead of learning a direct function F(x), ResNet learns the residual function F(x) + x. This design facilitates efficient gradient flow during backpropagation, stabilizes training even in extremely deep networks, and allows models to capture complex hierarchical features effectively.



To further enhance ResNet's representational power, Squeeze-and-Excitation (SE) blocks have been integrated into its architecture. SE blocks adaptively recalibrate channel-wise feature responses by explicitly modeling interdependencies between channels. Through a two-step process: global average pooling ("squeeze") followed by fully connected layers and nonlinear activations ("excitation"): SE blocks generate channel-specific scaling factors. These factors emphasize informative features while suppressing less relevant ones, thereby improving model performance without significantly increasing computational complexity.



The CIFAR-10 dataset serves as a widely recognized benchmark for evaluating image classification methods. It comprises 60,000 color images of size 32x32 pixels distributed evenly across 10 distinct classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The dataset's moderate complexity and relatively small image resolution make it an excellent testbed for exploring novel architectural improvements and optimization strategies.

This project specifically focuses on designing an 80-layer ResNet-SE model tailored for CIFAR-10 classification tasks. To achieve high accuracy while maintaining computational efficiency, we experimented with advanced preprocessing techniques such as normalization and one-hot encoding of labels. Additionally, extensive data augmentation methods, including random cropping, horizontal flipping, rotation adjustments, and color jittering, were employed to enhance data diversity and improve generalization capabilities. Furthermore, optimization strategies such as adaptive learning rate scheduling and weight decay regularization were explored to facilitate efficient convergence and robust model performance. Through this comprehensive approach combining architectural innovations with sophisticated training methodologies, we aim to demonstrate significant improvements in CIFAR-10 classification accuracy while providing insights into effective practices for training deep convolutional neural networks.

Methodology

1. Data Preprocessing:

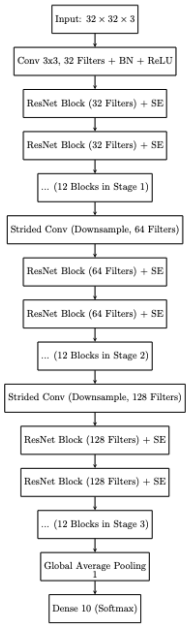
The CIFAR-10 dataset underwent several preprocessing steps to ensure better model performance. First, the images were normalized to the [0,1] range and standardized using the dataset's mean and standard deviation. This ensured that the input data had a consistent distribution. Labels were one-hot encoded to align with the categorical cross-entropy loss function used during training. To facilitate hyperparameter tuning, 10% of the training dataset was set aside as a validation set.

2. Data Augmentation:

To improve the model's generalization and robustness, we employed extensive data augmentation during training. The augmentations included:
- Random rotations up to ±15 degrees.
- Horizontal and vertical shifts up to 10% of the image size.
- Horizontal flips for orientation invariance.
- Shear transformations and zoom operations for added diversity.

For test-time augmentation (TTA), multiple augmented versions of the test images were generated. Predictions for each augmented version were averaged to produce the final classification result, improving robustness during inference.

3. Custom ResNet-SE Architecture:



The architecture used in this project is an 80-layer ResNet model enhanced with Squeeze-and-Excitation (SE) blocks. The components of the model are as follows:

Input Layer

The input to the network is a three-channel RGB image of size 32×32×32.

Convolutional Stem

The initial layers perform feature extraction through convolutional operations, followed by batch normalization and non-linearity:
- Conv2D (3×3, 64 filters, stride 1) – Extracts low-level features.
- Batch Normalization – Normalizes activations for stable training.
- ReLU Activation – Introduces non-linearity.

SE-ResNet Residual Blocks
The core of the model consists of SE-ResNet blocks, each containing:
- Two 3×3 Convolutional Layers with batch normalization and ReLU activations.
- Skip Connection (identity mapping) added to maintain gradient flow.
- Squeeze-and-Excitation (SE) Block:
  - Squeeze Phase: Global Average Pooling (GAP) to compute channel-wise statistics.
  - Excitation Phase: Two fully connected layers with non-linearity (ReLU and Sigmoid) to generate attention weights for each channel.
  - Scaling Phase: The original feature maps are reweighted using these attention scores.

Each residual block is structured as follows:
Conv2D (3×3, F filters, stride 1) → BatchNorm → ReLU
Conv2D (3×3, F filters, stride 1) → BatchNorm
SE Block (Global Pooling → FC → ReLU → FC → Sigmoid)
Element-wise Addition (Residual Connection)
ReLU Activation

The number of residual blocks varies based on model depth, with deeper networks having more stacked blocks.

Transition to Classification Head:
After passing through multiple SE-ResNet blocks, the feature maps are aggregated using:
- Global Average Pooling (GAP) – Reduces spatial dimensions.
- Fully Connected (FC) Layer – Maps pooled features to class probabilities.
- Softmax Activation – Outputs normalized probabilities for classification.

4. Training Configuration:

The model was compiled using the Adam optimizer with categorical cross-entropy loss. The training process was configured as follows:

- **Learning Rate Scheduler:** A custom learning rate schedule with the following stages:
  - Warm-up phase for the first 5 epochs, gradually increasing the learning rate.
  - Standard learning rate for epochs 5 to 20.
  - Learning rate decays at epochs 100 and 250.
- **Batch Size:** 128.
- **Callbacks:** The following callbacks were utilized during training:
  - Early Stopping: Halted training if validation loss plateaued for 200 epochs.
  - Model Checkpointing: Saved the best-performing model based on validation loss.
  - ReduceLROnPlateau: Dynamically reduced the learning rate when validation loss stopped improving.

5. Experimentation journey on design and training:

1. **Initial Approach: Vanilla ResNet as a Baseline**
   We began with Vanilla ResNet, leveraging residual connections to stabilize deep network training. The model was trained on CIFAR-10 with standard preprocessing (random cropping, flipping, normalization). Using fixed learning rates and SGD, it struggled with fine-grained class separation, highlighting the need for better feature prioritization.

2. **Enhancing Feature Selection: Transition to SE-ResNet**
   To improve feature representation, we integrated SE blocks, which use global pooling and fully connected layers to reweight feature channels dynamically. This boosted classification performance but increased parameters (~10%) and computational cost. To mitigate overfitting, we applied dropout (p=0.5) and L2 weight decay (1e-4).

3. **Hyperparameter Tuning: Optimizing Model Performance**
   Fixed learning rates led to stagnation, so we adopted adaptive scheduling:

   a. **Cosine Annealing** for gradual convergence.
   b. **ReduceLROnPlateau** for dynamic adjustments based on validation loss.
   c. **Batch size tuning** (optimal at 128) for balance between stability and efficiency.

   These optimizations significantly improved accuracy over Vanilla ResNet while keeping computational overhead manageable.

6. Test-Time Augmentation (TTA):

Test-time augmentation was implemented to enhance the model's robustness during inference. Predictions were averaged over multiple augmented versions of the test images, yielding a more accurate classification result.

Lessons Learned:

- Squeeze-and-Excitation blocks significantly improved the model's ability to classify challenging classes by recalibrating feature maps.
- Data augmentation played a crucial role in reducing overfitting and improving generalization.
- A well-designed learning rate schedule accelerated convergence and improved overall performance.

Results:

1. Test Accuracy:

The custom ResNet-SE model achieved a validation accuracy of 94.10%, demonstrating the effectiveness of SE blocks and advanced training techniques.

2. Model Architecture and Parameters:

The designed ResNet-SE model for CIFAR-10 classification comprises a total of 4,628,210 parameters. Out of these, 4,617,010 parameters are trainable. The remaining 11,200 parameters are non-trainable.

Model Architecture:

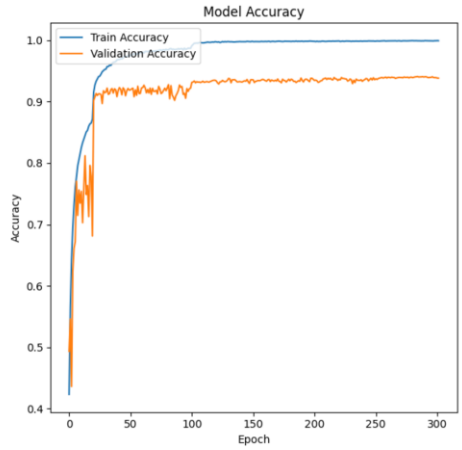
Layer Type	Output Shape	Description
Input	32×32×3	RGB image input
Conv2D (3×3, 64)	32×32×64	Initial feature extraction
BatchNorm + ReLU	32×32×64	Normalization and activation
SE-ResNet Block 1	32×32×64	Conv2D → BatchNorm → SE → Residual Add
SE-ResNet Block 2	32×32×128	Conv2D → BatchNorm → SE → Residual Add
SE-ResNet Block 3	16×16×256	Conv2D → BatchNorm → SE → Residual Add
SE-ResNet Block 4	8×8×512	Conv2D → BatchNorm → SE → Residual Add
Global Avg Pooling	1×1×512	Aggregates feature maps
Fully Connected (Dense)	10	Class scores
Softmax Activation	10	Final classification output

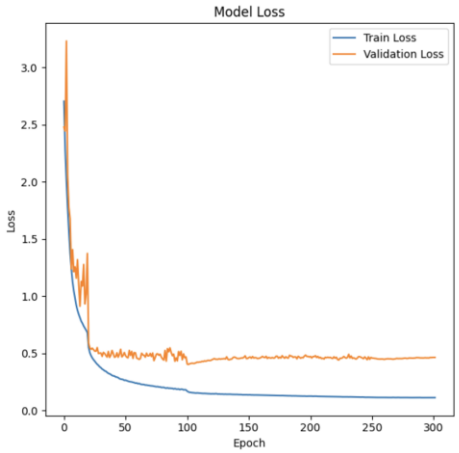
3. Training and Validation Performance:

The model achieved a training accuracy of 99.99% and a validation accuracy of 94.10%. The training and validation loss curves showed stable convergence, indicating effective training and minimal overfitting.

4. Comparison with Baseline:

The addition of SE blocks improved the test accuracy by approximately 10% compared to a baseline ResNet model without SE optimization, highlighting the importance of feature recalibration.





Citations:

1. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE CVPR

2. Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-Excitation Networks. Proceedings of the IEEE CVPR.