Team YAN — Report: Deep Learning Project 1

Written By
ar6316 Athul Radhakrishnan
nd2745 Nidhi Donuru
yt3150 Yumiko Tajiri
Kaggle team name: YAN
Repo: GitHub repository

Overview

This project is centered on the development of a custom, efficient residual network (ResNet) model designed to operate under a strict parameter constraint of fewer than 5 million parameters. The objective was to accurately classify images in the CIFAR-10 dataset while optimizing test accuracy within the specified architectural and parameter limitations. This work was conducted as part of a Kaggle-style competition for the Deep Learning Project-1 course (Spring 2025). During the initial phase, only 50% of the test dataset was labeled and available for training and validation, with the remaining 50% reserved for final evaluation following the competition's conclusion. Our team, consisting of Athul Radhakrishnan (ar6316), Nidhi Donuru (nd2745), and Yumiko Tajiri (yt3150), undertook extensive experimentation involving architectural modifications, hyperparameter tuning strategies, data augmentation techniques, and training optimizations to enhance model performance within the given constraints. Through rigorous experimentation, the final model integrated a Squeeze-and-Excitation (SE) block-enhanced ResNet architecture. This design significantly improved the model's representational capacity while maintaining compliance with the parameter limitation. The resulting model achieved a validation accuracy of 93.66% on the labeled test data and 81.55% on the complete unlabeled test dataset, effectively surpassing the baseline requirement of 80% test accuracy.

Code Repository

The code repository is available at this GitHub repository

Summary of Findings

Our project aimed to design a lightweight yet high-performing ResNet-based architecture for CIFAR-10 image classification, adhering to the strict constraint of having fewer than 5 million trainable parameters. After a thorough exploration of architectural enhancements, training strategies, and regularization methods, we achieved competitive results that surpassed the baseline test accuracy requirement.

Key findings include :

- **Model Architecture and SE Integration:** We developed a custom ResNet model augmented with Squeeze-and-Excitation (SE) blocks, which substantially enhanced the model's channel-wise feature recalibration without increasing the parameter count. The integration of SE blocks enabled the network to dynamically emphasize informative features, thereby improving its generalization capabilities.
- **Parameter Efficiency:** The final model was designed to maintain a total parameter count below 5 million by strategically managing the number of layers and filter sizes. Our model comprises 4,984,747 trainable parameters, ensuring computational efficiency while preserving robust classification performance.
- **Optimization and Training Strategy:** To achieve optimal performance, the final model was trained using the Adam optimizer with a learning rate of 0.0001 and a weight decay of 5e-3, which facilitated stable convergence. Additionally, a MultiStepLR scheduler was employed to reduce the learning rate at critical training milestones (epochs 100 and 150), further refining performance during the later stages of training.
- **Data Preprocessing:** Input images were normalized using the CIFAR-10-specific mean and standard deviation values. This transformation effectively regularized the data, mitigated gradient explosion issues, and enhanced model convergence stability.
- **Model Performance:** The final SE-enhanced ResNet model achieved a validation accuracy of 93.66% on labeled data and a test accuracy of 81.555% on the unseen 50% test dataset. These results surpassed the established baseline requirement of 80%, demonstrating the model's robustness and effectiveness within the defined resource constraints.
- **Custom Dataset Loader:** A tailored dataset loader class (Cifar10) was implemented to efficiently preprocess and manage the test data in accordance with the defined transformation pipeline.

  Overall, our experimental findings validate that a well-designed, parameter-efficient ResNet architecture enhanced with SE blocks can achieve strong classification

performance on CIFAR-10, even under stringent computational constraints.

Methodology

This section outlines the architectural choices, training strategies, and experiments conducted during model development. Our objective was to build a parameter-efficient ResNet-based model for CIFAR-10 classification, enhanced with Squeeze-and-Excitation (SE) blocks, and trained entirely from scratch while staying within the 5 million parameter constraint.

1. **Evolution from Initial Architecture**
   Our initial implementation followed a standard ResNet structure, where each residual block consisted of two sequential convolutional layers followed by batch normalization and ReLU activation which led to 4.3M parameters. While this setup performed adequately, we observed limitations in the model's ability to capture channel-wise dependencies and a lack of effective feature recalibration.

   To improve this, we introduced two major architectural changes:

   - **Integration of Squeeze-and-Excitation (SE) Blocks:** After the second convolutional layer in each residual block, we added an SE block to enhance channel-wise attention. The SE block performs:
     – *Squeeze:* Global average pooling across spatial dimensions to produce channel descriptors.
     – *Excitation:* A two-layer fully connected network with ReLU and Sigmoid activations to compute attention weights.
     – The output attention weights are multiplied (broadcasted) with the feature map to recalibrate the channels.

     This allowed the model to selectively emphasize informative features with minimal additional parameters.
   - **Modified Residual Block Flow:** The standard residual block was altered to incorporate the SE block before the residual addition:

```
    out = self.conv1(x)
    out = self.bn1(out)
    out = F.relu(out)

    out = self.conv2(out)
    out = self.bn2(out)

    out = self.se(out) # SE block

    out += self.shortcut(x)
    out = F.relu(out)
```

     This ensured the SE recalibration influences the final output of the block more effectively.

   These changes improved the model's representational capacity, validation and test performance while maintaining compliance with the parameter constraint.

2. **Data Preprocessing and Regularization**
   All input images were normalized using the standard CIFAR-10 dataset statistics (mean and standard deviation). This transformation helped regularize the data, improve training stability, and ensure consistent feature scaling throughout the network.

3. **Data Loading and Batch Size**

   We used a batch size of **128** for both the training and validation datasets during model training. This provided a good balance between training speed and convergence stability.

   For inference on the **unlabeled test data**, we used a smaller batch size of **32** to reduce memory load and improve prediction efficiency during the evaluation phase.

4. **Optimizer and Scheduler Experiments**
   We explored multiple optimizer-scheduler combinations to evaluate their impact on convergence and generalization.

   - **SGD with ReduceLROnPlateau Scheduler**
     Using SGD with lr=0.0001 and weight_decay=5e-3, paired with a ReduceLROnPlateau scheduler, led to poor performance. The model showed slow convergence and plateaued around 57.42% validation accuracy after 200 epochs, with minimal improvements in later stages.
   - **Adam with MultiStepLR Scheduler (Best Performance)**
     Using Adam optimizer (lr = 0.0001, weight_decay = 5e-3) with a MultiStepLR scheduler (milestones at epochs 100 and 150, gamma=0.1) resulted in faster convergence and consistently high validation accuracy, reaching a peak of 93.66%. This configuration provided the best balance between training stability and model performance.

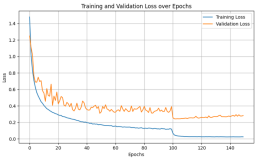
5. **Label Smoothing**
   We experimented with label smoothing to reduce overconfidence in predictions. However, models trained with label smoothing achieved slightly lower validation accuracy ($\sim$92.5%) compared to those trained without it. In this case, label smoothing did not provide a significant advantage.

6. **Regularization Tuning**
   To assess the effect of stronger regularization, we trained the model with a higher weight decay value (0.001). The performance remained consistent, with validation accuracy around 93.2%, indicating that the model was not highly sensitive to this change and was already well-regularized.

7. **Visualization of Training Behavior**

8. **Post-Training Image Visualization**
   To verify the correctness of data preprocessing and model predictions, we visualized sample images by reconstructing them from input vectors after training. The image tensors were reshaped and permuted to display RGB images using matplotlib. This helped validate that

(a) The accuracy rate over Epochs



(b) The loss rate over Epochs

Figure 1: Comparison of Accuracy and Loss over Epochs

the input pipeline was functioning as expected and ensured that no distortions occurred during transformation and normalization.

9. **Evaluation on Unlabeled Test Data**
   After training, the final model was evaluated on the held-out 50% test dataset, as per the competition setup. This allowed us to assess the generalizability of the model on unseen data.

Experiments

Model Search

We explored various iterations of the ResNet architecture during the model development process. Initially, we employed a ResNet-18 model with 4.3 million parameters as the baseline. Subsequently, we enhanced the model by incorporating Squeeze-and-Excitation (SE) blocks, resulting in a refined model with 4.98 million parameters.

Data Preparation For data preparation, we initially utilized the provided dataset files available on Brightspace. However, we later transitioned to the original CIFAR-10 repository to leverage PyTorch's built-in functions for efficient data downloading. PyTorch's DataLoader utility was employed to load both the training and test sets in batches. To further enhance model performance, we applied data normalization using CIFAR-10-specific mean and variance values. Additionally, data augmentation techniques such as random cropping and horizontal flipping were employed to improve generalization.

Optimizers and Schedulers We employed the Adam optimizer due to its established effectiveness in vision models and its ability to achieve faster convergence toward the global minimum compared to alternative optimizers. For learning rate adjustments, we utilized the MultiStepLR scheduler, which effectively reduced the learning rate at predefined milestones. Although we experimented with the ReduceOnPlateau scheduler, we observed that it significantly prolonged convergence, prompting us to discontinue its use.

The results of these experiments are presented in the subsequent section.

Results

The resulting model is a ResNet architecture with Squeeze-and-Excitation (SE) blocks, which improve feature representation by channel-wise attention.

The SE block includes a global average pooling to squeeze spatial dimensions. It also has two fully connected layers with ReLU and Sigmoid activations to learn channel-wise dependencies, and multiplicative channel-wise scaling of the original feature maps.

The backbone of the model uses ResNet, where residual connections are used to improve vanishing gradient problems and allow training of deeper networks.

- Each residual block is augumented with an SE block.
- An SE block performs the Squeeze, Excitation, and Scale.
- These blocks help the network learn inter-channel relationships, improving its ability to prioritize important features.

The model ends with a global average pooling layer and a fully connected output layer for classification.

The total number of trainable parameters in the model is 4.98 million.

Overall, the model achieved has a maximum test accuracy of 93.66%. The model predictions upon submission to Kaggle gave a score of 81.743 on 50% of the unlabeled data and 81.555% on 100% of the unlabeled test data.

Results Summary

The performance of various model configurations is summarized below:

Model Configuration	**50% Acc.**	**100% Acc.**
ResNetSE (lr=0.0001, wd=5e-3)	**81.555%**	**81.743%**
ResNetSE + Label Smoothing (smoothing = 0.1)	81.034%	80.945%
ResNetSE + Higher Reg. (weight_decay = 0.01)	79.731%	79.628%
ResNetSE Initial Version (lr=0.0001, wd=5e-3)	76.202%	76.017%
Baseline ResNet (lr=0.0001, wd=5e-3)	75.160%	74.820%

Table 1: Model Performance Summary

Citations

1. Kaiming He et al., "Deep Residual Learning for Image Recognition", CVPR 2016.
2. Jie Hu et al., "Squeeze-and-Excitation Networks", CVPR 2018.
3. Alex Krizhevsky, "Learning Multiple Layers of Features from Tiny Images", Technical Report, 2009.
4. Diederik Kingma and Jimmy Ba, "Adam: A Method for Stochastic Optimization", ICLR 2015.
5. H. Robbins and S. Monro, "A Stochastic Approximation Method", Annals of Mathematical Statistics, 1951.
6. L. N. Smith, "Cyclical Learning Rates for Training Neural Networks", 2017 (for LR scheduling inspiration).