Computational Photography Course Project: Enhanced CIFAR-10 Classification Using ResNet with an MLP Head

Jason Pickering - 100439553

April 6, 2025

Abstract This project explores the effectiveness of enhancing a ResNet-based architecture with a multilayer perceptron (MLP) classifier for the CIFAR-10 image classification task. Using PyTorch Lightning, we fine-tuned a pre-trained ResNet18 model with a custom MLP head that incorporates batch normalization, dropout, and a hidden layer. Our results demonstrate improved validation accuracy and suggest that augmenting classic CNN pre-trained models with deeper classifiers can be beneficial in low-resolution image tasks.

1 Introduction

Deep-convolutional networks like ResNet have achieved significant success in image classification. However, using a simple linear layer at the output can limit the model's capacity to learn complex class boundaries. This project investigates the effect of replacing the final linear classification layer in a ResNet model with a multilayer perceptron (MLP) consisting of an additional hidden layer, dropout, and normalization.

The dataset used for this task is CIFAR-10, a well-known benchmark in the machine learning community. It consists of 60,000 color images sized 32x32 pixels across 10 classes, with 6,000 images per class. The dataset is divided into 50,000 training images and 10,000 test images. The ten classes include airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks. Its relatively small image size and class diversity make CIFAR-10 an ideal testbed for evaluating the representational power of different neural network architectures.

2 Related Work

Convolutional Neural Networks (CNNs) have long been a cornerstone in image classification tasks. However, as CNNs become deeper, they suffer from the vanishing gradient problem, which makes training difficult and leads to degraded performance. To mitigate this, He et al. introduced the **Residual Network (ResNet)** architecture in 2015 [2]. The key innovation in ResNet is the use of identity-based *skip connections*, allowing gradients to flow directly through the layers of a network and allowing the training of extremely deep networks (e.g., ResNet-50, ResNet-101).

The original ResNet paper demonstrated state-of-the-art performance on ImageNet and won the ILSVRC 2015 classification task. For smaller datasets such as CIFAR-10, the authors also proposed a variant of ResNet adapted for low-resolution 32x32 images. This architecture typically uses fewer layers and smaller kernel sizes, making it more computationally efficient.

Although ResNet has a robust feature extraction backbone, the default classifier head, a single linear layer, is often lacking. Several studies have explored improving this by adding MLP-style heads to enhance class separation in the final embedding space. For example, Dosovitskiy et al. [1] introduced Vision Transformers (ViT) that rely heavily on MLP heads for classification after self-attention blocks. Similarly, Raghu et al. [3] showed that deeper heads can sometimes help CNNs match the representational power of attention-based models.

Additionally, MLP layers with dropout and batch normalization have been shown to improve generalization by reducing overfitting and improving convergence during training. These techniques are especially effective on datasets like CIFAR-10, which contain fewer classes but relatively high variability.

My project builds upon this prior work by modifying the classifier head of a pre-trained ResNet18 model to include a hidden layer, ReLU activation, dropout, and batch normalization. This setup aims to leverage the strong feature representations learned by ResNet while enhancing the classifier's ability to distinguish between classes.

3 Methodology

Dataset Preparation

The CIFAR-10 dataset contains 60,000 32x32 px colour images, categorized into 10 distinct classes (e.g. airplane, automobile, bird, cat, etc.). Each class contains 6,000 images. The data set is divided into 50,000 training images and 10,000 test images. A validation split of 5,000 samples was extracted from the training set to tune model hyperparameters and prevent overfitting.

Images were normalized using the dataset's mean and standard deviation per channel. Data augmentation included random horizontal flips and random crops with padding to introduce minor variations and help generalize learning.

Model Architecture

The base model is a ResNet18 architecture, pre-trained on a variant of CIFAR-10-compatible data. ResNet18 utilizes residual connections that allow gradients to back-propagate more effectively across layers, making it ideal for deeper networks. For this project, we retained the pre-trained convolutional feature extractor and replaced the final fully-connected classifier head with a custom multi-layer perceptron (MLP) as follows:

```
nn.Sequential(
    nn.BatchNorm1d(num_in_features),
    nn.Linear(num_in_features, 1024),
    nn.ReLU(),
    nn.BatchNorm1d(1024),
```

```
    nn.Dropout(0.5),
    nn.Linear(1024, 10)
)
```

Here, `num_in_features` corresponds to the dimensionality of the final pooled feature vector from ResNet18, typically 1,024. Batch normalization stabilizes training and accelerates convergence, while dropout with a probability of 0.5 helps reduce overfitting by randomly disabling neurons during training.

Training Strategy

To reduce computational load and overfitting, the convolutional layers of the ResNet18 backbone were frozen during training—only the new MLP head was trained. This approach uses the pre-trained feature extractor as a fixed embedding generator.

The model was trained using the AdamW optimizer, known for decoupling weight decay from gradient updates. A learning rate of 1e-3 and weight decay of 1e-4 were used, along with a multi-step learning rate scheduler that decreased the learning rate at predefined epochs to encourage better convergence.

Training was performed over 20 epochs with early stopping monitored on the validation loss. Cross-entropy loss was used as the objective function, measuring the difference between the predicted and true label distributions.

Implementation

The training pipeline was implemented using PyTorch Lightning, which provides high-level abstractions for model training and checkpointing. This modular design helped maintain cleaner code and enabled reproducibility and ease of experimentation.

4 Results

The final test and validation accuracy scores were as follows:

- **Validation Accuracy:** 0.4832
- **Test Accuracy:** 0.4575

Training converged in under 20 epochs, and the model displayed improved generalization compared to the default linear classifier baseline. We observed that adding

dropout and normalization helped stabilize training.

5 Discussion

Replacing the simple linear layer with an MLP improved the model's performance, especially on the validation set. The hidden layer likely helped in learning non-linear boundaries in the feature space. While freezing the backbone limited overall learning capacity, it was a practical decision for faster training and reduced overfitting risk on CIFAR-10.

Future work could explore fine-tuning the full model end-to-end or adding more layers to the MLP to assess trade-offs in training time and accuracy.

6 Conclusions

We demonstrated that using an MLP as a classifier head in a ResNet18 model can enhance classification accuracy on CIFAR-10. Our approach offers a simple but effective way to boost model performance using standard deep learning techniques such as dropout and batch normalization.

A Hyperparameters and Training Setup

- Optimizer: AdamW
- Learning Rate: 1e-3
- Weight Decay: 1e-4
- Scheduler: MultiStepLR (milestones: [100, 150], gamma: 0.1)
- Epochs: 20

B Repository Link

The full source code for this project is available at the following GitHub repository:

`https://github.com/japickering42/CSCI_4220U_Project`

References

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[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)*, pages 770–778, 2016.

[3] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems (NeurIPS)*, 34:12116–12128, 2021.