A Comparative Study of Self-Supervised and Supervised Learning Methods for Image Classification on CIFAR-100

Xiang Zheng
21307110169@m.fudan.edu.cn

YunNam Chan
21340980053@m.fudan.edu.cn

Abstract

This study evaluates the performance of different training methods for image classification on the CIFAR-100 dataset. We compare three approaches: fine-tuning a ResNet-18 model pretrained on ImageNet, self-supervised learning using the BYOL (Bootstrap Your Own Latent) algorithm, and training a ResNet-18 model from scratch. The results indicate that fine-tuning the pretrained ResNet-18 model achieves the highest validation accuracy of 78.07%, leveraging rich feature representations learned from the larger ImageNet dataset. The self-supervised BYOL method also demonstrates its effectiveness, achieving a validation accuracy of 71.26%, which surpasses the performance of the model trained from scratch. These findings highlight the significant impact of both self-supervised learning and supervised pretraining on improving classification performance. Our study underscores the superiority of pretrained models and self-supervised methods, particularly in scenarios with limited labeled data, and suggests future exploration into combining these techniques for enhanced performance.

Note: You can find the project files in the GitHub repo and the model parameters in Google Drive. For instructions on how to train and test the neural network, please refer to the readme file.

1 Introduction

The rise of Machine Learning and ultimately Deep Learning techniques has propelled the development and enhancement of image classification tasks, paving the way for more intelligent and sophisticated models. In recent years, interest in Self-supervised Learning has surged, promising performance advantages against traditional Supervised Learning techniques. This study aims to examine the application of one such self-supervised learning algorithm, BYOL (Bootstrap Your Own Latent), on ResNet-18 to test its performance through the Linear Classification Protocol on the CIFAR-100 dataset. Additionally, we juxtapose this against the results from two supervised learning methodologies: one utilizing pretrained weights of ResNet-18 and the other training ResNet-18 from scratch on the CIFAR-100 dataset. This comparative analysis seeks to establish a clear point of comparison between self-supervised and supervised learning techniques.

2 Dataset

In this study, we utilize two well-known datasets in the field of image classification: CIFAR-10 and CIFAR-100. Both datasets are commonly used benchmarks that consist of colored images with a size of 32x32 pixels, but they differ in the number of categories they contain. A snap view of these two datasets are shown in Figure 1.

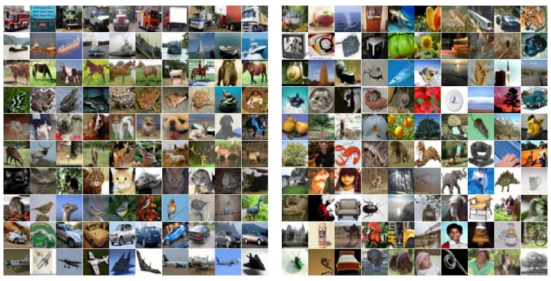

Figure 1: A snap view of (a) CIFAR-10, (b) CIFAR-100

2.1 CIFAR-10

CIFAR-10 is used as the dataset for self-supervised learning. It comprises 60,000 images divided into 10 classes, with 6,000 images per class. The dataset is split into 50,000 training images and 10,000 test images. Each class represents a distinct object, such as airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks.

For the purpose of this study, the CIFAR-10

dataset is used to pretrain the ResNet-18 model using the BYOL self-supervised learning algorithm. The images undergo a series of augmentations defined by the BYOL algorithm to improve the model's ability to learn robust feature representations. The augmentations applied include random color jitter, random grayscale conversion, random horizontal flip, random Gaussian blur, random resized crop, and normalization. For parameters of these transformations, we choose the recommended value offered by the author of BYOL paper.

The CIFAR-10 dataloader is prepared by first transforming the training images to tensors. If the dataset is not available locally, it is downloaded automatically. The dataloader is then created using the transformed dataset, with images loaded in batches of a specified size, shuffled, and with certain other settings like the number of workers and pin memory enabled for efficient loading.

2.2 CIFAR-100

CIFAR-100 serves as the dataset for the classification task. It consists of 60,000 images distributed across 100 classes, with each class containing 600 images. The dataset is divided into 50,000 training images and 10,000 test images. Each of the 100 classes falls under a broader superclass category, making CIFAR-100 a more challenging and fine-grained classification task compared to CIFAR-10.

In this study, the pretrained ResNet-18 model from the self-supervised learning phase is tested on CIFAR-100 using the Linear Classification Protocol. Additionally, two supervised learning methodologies are employed on CIFAR-100: one with ResNet-18 pretrained weights and another training ResNet-18 from scratch. The data transformation for CIFAR-100 includes random cropping, resizing, horizontal flipping, rotation for the training set, and resizing for the test set, followed by normalization. Specifically, transformations like `RandomCrop`, `Resize`, `RandomHorizontalFlip`, `RandomRotation`, and `Normalize` are used to prepare the images for training dataset.

The CIFAR-100 dataloader is prepared similarly to the CIFAR-10 dataloader, with specific transformations applied to the training and testing datasets to augment and normalize the images. If the dataset is not available locally, it is downloaded automatically. The dataloaders are then created using the transformed datasets, with images loaded in batches, shuffled for the training set, and with certain other settings like the number of workers enabled for efficient loading.

By using these datasets and their respective preprocessing steps, we aim to provide a comprehensive comparison of the performance of self-supervised and supervised learning techniques on a complex classification task.

3 Model Architecture

This study employs a two-stage model architecture approach to evaluate the performance of different learning techniques. The architecture can be decomposed into two main components: the base encoder and the linear classifier. The base encoder utilizes the ResNet-18 architecture, while the linear classifier is a simple Multi-Layer Perceptron (MLP).

3.1 Base Encoder

The base encoder, ResNet-18, is a widely used convolutional neural network architecture known for its effectiveness in image classification tasks. In this study, we obtain the weights for the base encoder using three different methods:

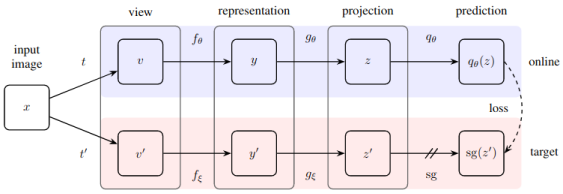

Figure 2: The procedure of the BYOL method

1. **Self-Supervised learning with BYOL:** The ResNet-18 model is pretrained on the CIFAR-10 dataset using the Bootstrap Your Own Latent (BYOL) self-supervised learning algorithm. This method does not rely on labeled data and focuses on learning useful feature representations through data augmentation and self-supervised training objectives. The procedure of the method is visualized in Figure 2.

2. **Supervised learning with pretrained weights:** The ResNet-18 model is initialized with weights pretrained on ImageNet. These pretrained weights provide a strong starting point for further training on the CIFAR-100 dataset.

3. **Supervised learning from scratch:** The ResNet-18 model is trained from scratch on the CIFAR-100 dataset. This method does not utilize any prior knowledge and learns the feature representations solely from the CIFAR-100 training data.

Once the base encoder weights are obtained through these methods, the ResNet-18 model is frozen to serve as the backbone for feature extraction. This means that the weights of the ResNet-18 model are not updated during the training of the linear classifier, ensuring that the learned feature representations remain constant.

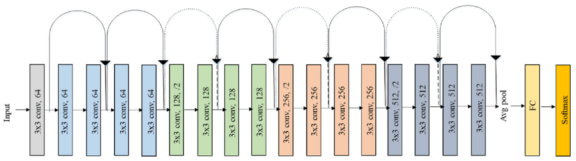

Figure 3: Model Architecture of ResNet-18

3.2 Linear Classifier

The second component of the model architecture is the linear classifier, which is used to classify the features extracted by the frozen ResNet-18 encoder. The linear classifier is implemented as a simple Multi-Layer Perceptron (MLP) with a single fully connected layer. The architecture of the MLP classifier is as follows:

- **Input layer:** The input dimension corresponds to the output feature dimension of the ResNet-18 encoder, which is 512.
- **Output layer:** The output layer consists of a fully connected layer with the number of output units equal to the number of classes in the CIFAR-100 dataset, which is 100.

The forward pass of the MLP classifier involves feeding the extracted features through the fully connected layer to obtain the class scores.

By combining the pretrained ResNet-18 encoder with the MLP classifier, the model can effectively leverage the learned feature representations to perform the classification task on the CIFAR-100 dataset. This approach allows us to compare the performance of different learning techniques in a consistent and structured manner.

4 Training and Results

In this section, we outline the training configuration and results for three different models evaluated in this study: BYOL (Bootstrap Your Own Latent), fine-tuning with pretrained ResNet-18 on ImageNet weights, and training ResNet-18 from scratch on CIFAR-100. We will detail the model configurations, analyze the training loss of BYOL, ResNet-18, and evaluate the performance of a linear classifier trained on these representations.

4.1 Model Configuration

The configurations for each model are summarized in Table 1. These configurations were selected based on empirical evidence to optimize performance, where LR stands for learning rate and WD stands for weight decay.

4.2 Training Loss of BYOL

We implemented BYOL with three different configurations, summarized in Table 2. The training loss for these configurations is visualized in Figure 4.

Figure 4: Training Loss in BYOL

Due to limited computational resources, we tested only three configurations to understand the impact of the projection (hidden) dimension, prediction (output) dimension, and dataset size on the method. As shown in Figure 4, larger feature dimensions and larger datasets result in faster convergence of the training loss, which implies better performance. Specifically, the training loss for CIFAR-10 plateaus around 0.03, while for Tiny-ImageNet, it plateaus around 0.01.

The faster convergence of training loss typically indicates that the model is learning representations more efficiently. In the context of self-supervised learning, this can lead to better downstream task

Model	Epochs	LR	WD	Optimizer	Feature Dimension / Update Rate
BYOL	100	0.0003	-	Adam	Hidden: 1024, Output: 128 / 0.99
Fine-tuning	20	0.001	0.0001	Adam	Pretrained / -
Training from Scratch	20	0.001	0.0001	Adam	Random Initialization / -

Table 1: Model Configuration and Training Process

Hidden Dim	Output Dim	Dataset
1024	128	CIFAR-10
2048	256	CIFAR-10
2048	256	Tiny-ImageNet

Table 2: Configurations for BYOL

performance because the model has learned to capture more useful and discriminative features from the data. Based on these observations, we proceed with the best configuration (2048, 256, Tiny-ImageNet) to fine-tune the trained encoder with 10% of the labels for 20 epochs.

4.3 Training Loss of ResNet-18

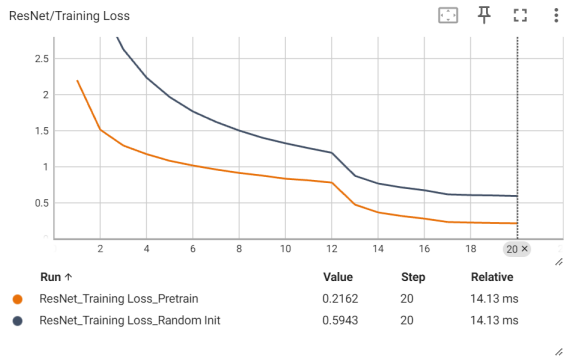

Figure 5: Training Loss in ResNet-18

From Figure 5, we observe that the training loss of the pretrained ResNet-18 model is lower compared to the randomly initialized ResNet-18. Both models show a similar rate of decrease in training loss, but the pretrained model starts with a significantly lower initial loss and maintains a lower loss throughout the training process.

This lower initial training loss for the pretrained model can be attributed to the fact that it starts with weights already optimized for a related task (ImageNet classification). These pretrained weights provide a strong foundation, capturing useful feature representations that are broadly applicable. Consequently, the pretrained model requires less adjustment during the fine-tuning process on CIFAR-100, resulting in a consistently lower training loss.

In contrast, the randomly initialized ResNet-18 model begins with weights that are randomly distributed, necessitating substantial modifications to learn useful features from scratch. Despite the similar rate of loss decrease, the initial high loss indicates the additional effort required for the model to adapt to the CIFAR-100 dataset from an untrained state.

The consistently lower training loss of the pretrained model does not only reflect its starting advantage but also suggests more efficient learning. The pretrained model can refine its weights with minimal adjustments, achieving better generalization with less effort compared to training from scratch.

In summary, the lower training loss for the pretrained ResNet-18 model underscores the advantage of using pretrained weights, allowing the model to leverage previously learned features and achieve superior performance with more efficient training. This highlights the benefits of transfer learning in enhancing model performance on related tasks.

4.4 Evaluation of Linear Classifier

To further assess the quality of the learned representations from BYOL and ResNet-18, we trained a linear classifier on top of these representations. Figures 6 and 7 display the loss and accuracy during the training of the linear classifier, respectively.

From Figure 7, it is evident that the supervised method with pretrained weights achieves the highest performance, boasting a validation accuracy of 78.07%. In comparison, the self-supervised method achieves an accuracy of 71.26%, slightly surpassing the supervised method without pre-training.

These results underscore the significant impact of both self-supervised learning and supervised pretraining on the performance of the linear classifier. While the self-supervised method falls short of matching the accuracy achieved by supervised pretraining, it demonstrates its effectiveness by outperforming the model trained without pre-training. This highlights the capability of self-supervised

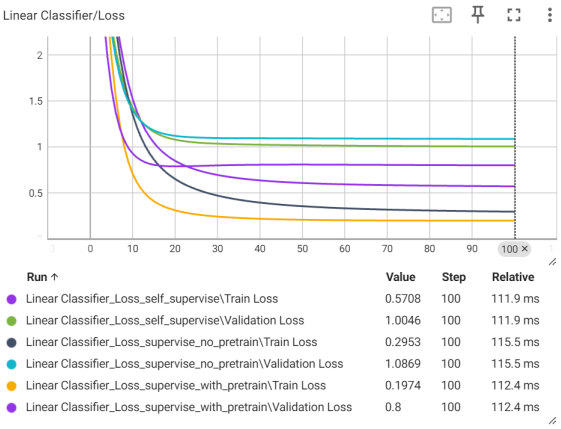

Figure 6: Losses in Linear Classifier

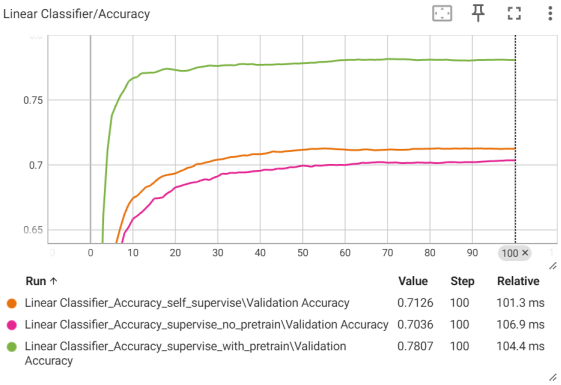

Figure 7: Accuracies in Linear Classifier

learning to derive meaningful representations from data without explicit labels.

Self-Supervised pre-training confers a distinct advantage by enabling the model to uncover intrinsic patterns and features within the data, which subsequently enhance performance in downstream tasks like classification. This approach establishes a robust foundation for the linear classifier, yielding superior results compared to training from scratch.

Conversely, supervised pretraining likely captures task-specific features directly relevant to the classification task, resulting in the highest observed accuracy.

Training the classifier without any pre-training yields the lowest accuracy, as the model must simultaneously learn both feature extraction and classification from scratch, which is typically less efficient and effective.

These findings emphasize the critical role of both self-supervised methods and pretrained models in enhancing the performance of classification tasks. The utilizations of self-supervised learning and supervised pretraining yield robust and effective feature representations, thereby improving overall performance significantly.

5 Discussion & Conclusion

From the results above, we observe that fine-tuning the pretrained ResNet-18 model on CIFAR-100 achieves the highest validation accuracy of 78.07%. This method outperforms both self-supervised learning with BYOL, which achieves an accuracy of 71.26%, and training from scratch, which has the lowest accuracy.

Fine-tuning a model pretrained on a larger dataset (ImageNet) provides a significant advantage in learning representations that generalize well to the CIFAR-100 dataset. The self-supervised BYOL method, while slightly less effective than fine-tuning, shows its strength by outperforming the model trained from scratch. This demonstrates the potential of self-supervised learning in scenarios where labeled data is scarce.

In conclusion, for CIFAR-100 classification, fine-tuning a pretrained ResNet-18 on ImageNet weights is the most effective approach. However, self-supervised learning with BYOL is a promising method, especially when labeled data is not available, and it significantly outperforms training from scratch.

Future work could explore combining self-

supervised pretraining with fine-tuning to leverage the benefits of both approaches. Additionally, testing more advanced self-supervised learning techniques on other datasets could further validate their effectiveness.