Improvment of ResNet-18 with pre-trained weight using CIFAR-10 Dataset

Salman Khalil
Term Project, June 2023
Professor: Jung Heechul

***Abstract*—The goal of this term project is to improve the predicted accuracy of the ResNet-18 model, which is a deep learning architecture noted for its ability to handle complicated picture categorization problems. The project uses pre-trained ResNet-80 model weights as a starting point and then fine-tunes these weights to fit the model to a given job and dataset. The CIFAR-10 dataset is used, which is well-known and frequently used in machine learning and computer vision. The findings suggest that fine-tuning pre-trained weights may greatly enhance ResNet-18 performance on CIFAR-10.**

***Index Terms*—**

I. Introduction

Neural networks, a pillar of artificial intelligence, have transformed several industries by providing strong tools for modeling complicated patterns and relationships in data. Convolutional Neural Networks (CNNs) have emerged as a particularly significant architecture among them because to their extraordinary performance in image and video processing applications. CNNs, which are inspired by biological processes in the human brain, are meant to learn spatial hierarchies of characteristics from data automatically and adaptively, making them extremely successful for tasks such as image classification, object recognition, and semantic segmentation(1; 2). Convolutional Neural Networks (CNNs) have shown great potential in medical sciences, notably in medical imaging, illness diagnosis, and patient care. The use of CNNs in medical sciences is extensive and expanding as technology advances and big datasets for training these models become available(3).

CNNs, despite their effectiveness, are not without obstacles. One of the key concerns is their susceptibility to adversarial assaults, in which slight, purposefully created changes to the input might result in inaccurate results. Furthermore, because to their complicated, black-box nature, CNN interpretability remains an issue. However, these concerns have been addressed by a variety of measures, including the construction of robust models, the use of regularization techniques, and the investigation of network interpretation approaches(4; 5).

Convolutional Neural Networks (CNNs) have proven useful in medical sciences, CNNs, for example, have been used to detect cancers in brain MRIs, detect diabetic retinopathy in retinal pictures, and diagnose skin cancer using dermoscopic images(6). A four-layer CNN, for example, was used to forecast congestive heart failure and chronic obstructive pulmonary disease, yielding considerable gains over older approaches(7).

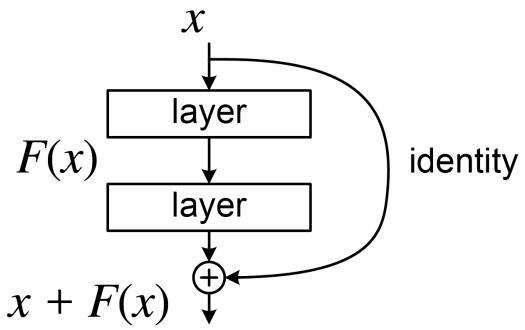

Fig. 1: Skip connections or a residual block

Over the years, several CNN models have been proposed, each improving upon its predecessors in terms of accuracy and efficiency. Among the most famous are LeNet, AlexNet, VGGNet, GoogLeNet, and ResNet. ResNet, or Residual Network. The ResNet18 model, a variant of ResNet, has been successfully applied in various domains, including medical imaging. For instance, it has been used for the detection and grading of Diabetic Retinopathy, a major cause of blindness, achieving impressive results(8).

ResNet, or Residual Network, was proposed by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun in their 2015 paper, "Deep Residual Learning for Image Recognition" (9). In this study, we come up with an approach to enhance the predictive accuracy of the ResNet18 model, which is a deep learning architecture renowned for its proficiency in handling complex image classification tasks. Recognizing the potential of this model, we have developed a unique code that leverages the pre-trained weights of the ResNet80 model as a starting point. The crux of our methodology lies in the fine-tuning of these weights, a process that allows us to tailor the model to our specific task and dataset. We have used CIFAR-10 dataset, which is a widely recognized and utilized dataset in the field of machine learning and computer vision.

II. Methodoloyg

A. Resnet18

The ResNet (Residual Network) architecture, particularly the ResNet18 variation, is a deep learning model that has

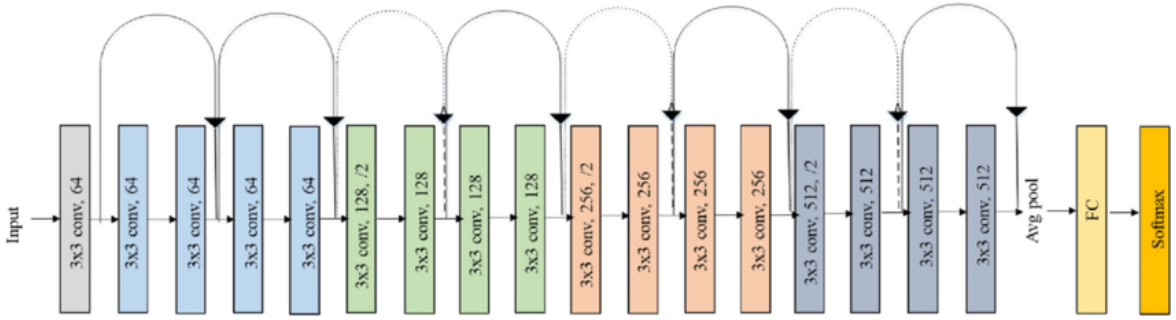

Fig. 2: ResNet-18 Architecture

achieved substantial advances in image recognition. The objective for developing ResNet was to solve the difficulty of training extremely deep neural networks. As the depth of these networks developed, they began to encounter issues such as disappearing gradients and degradation, resulting in a decline in performance. The architecture was designed to solve the problem of vanishing and exploding gradients and the difficulty of training deep neural networks. ResNet's main innovation is the introduction of "skip connections" or "shortcut connections", which allow the gradient to be directly backpropagated to earlier layers in Fig 1. These connections essentially create a shorter path for the gradient during backpropagation, mitigating the vanishing gradient problem and enabling the training of much deeper networks. The authors of the original paper demonstrated that networks of depth 152 could be trained using this architecture, significantly deeper than previous architectures(9).

ResNet18, a ResNet version, has numerous "residual blocks" in its construction. Each residual block is made up of a set of convolutional layers connected by a shortcut. The shortcut link allows the input to a block to be appended to the block's output, resulting in a feedback loop. This addition operation is done element by element, and if the dimensions of the input and output do not match, a linear projection is used to match the dimensions. ResNet's unique skip connection architecture considerably alleviates the vanishing gradient problem. It facilitates the training of very deep networks by enabling gradients to travel straight across multiple layers without attenuation. The gradient can be immediately backpropagated through the skip connections, bypassing the stacked layers, when the network conducts backpropagation to compute the gradient for updating the weights. This permits the gradient to propagate further without becoming too tiny, decreasing the effects of the vanishing gradient problem. This is particularly useful in deeper networks, where the vanishing gradient problem might be severe.

ResNet transformed the deep learning scene in 2015 when it won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The ImageNet dataset is a massive visual database intended for use in visual object identification research, with over 14 million photos classified into over 20,000 categories. The release of ResNet was a notable advance since the architecture was capable of training a 152-layer network for the ImageNet dataset, which was far deeper than earlier models while having fewer parameters and being easier to optimize.

ResNet's core building block is the "residual block." ResNet18, as a particular instance of ResNet, is made up of numerous stacked residual blocks. Each residual block is made up of the following elements:

- A convolutional layer with a tiny (typically 3x3) filter size.
- A layer of batch normalization.
- A function for activating ReLU.
- Another layer of convolution.
- Another layer of batch normalization.

This is followed by a skip (or shortcut) link that bypasses these levels and creates a parallel channel before the outputs are combined and transmitted via another ReLU activation.

ResNet-18's design begins with a convolutional layer, followed by a succession of residual blocks (each with two layers) with a progressive rise in the number of filters. There are two blocks with 64 filters, two blocks with 128 filters, two blocks with 256 filters, and lastly two blocks with 512 filters. Each of these blocks has two convolutional layers, which account for the vast bulk of the network's depth. The network is completed by a completely linked layer. The design has 18 layers in total, including two convolutional layers in each of the eight blocks, as well as the initial convolutional and final fully linked layers. The comprehensive architecture of the ResNet model is vividly illustrated in Figure 2.

B. Code Structure

PyTorch is a popular open-source machine learning framework that provides tools for developing deep learning models. The purpose of this research is to explain the operation of a specific code snippet that utilizes PyTorch to train the ResNet18 model on the CIFAR10 dataset. The code begins by importing the required libraries, which are torch, torchvision, and numpy. Following that, the CUDA device's environment variables are configured, and parameters such as batch size and learning rate are defined. A data directory for storing and retrieving the CIFAR10 dataset is also created.

Following that, the code describes two sets of transformations, one for training data and one for testing data.

```
print("loss:", np.mean(test_loss))
print('Accuracy', np.mean(test_acc))
```

```
loss: 0.3683237129598856
Accuracy 91.43209
```

(a)

```
print("loss:", np.mean(test_loss))
print('Accuracy', np.mean(test_acc))
```

```
loss: 0.20973228251188994
Accuracy 94.98182
```

(b)

Fig. 3: (a) Prediction Accuracy with Adam Optimizer and (b) Prediction Accuracy with Stochastic Gradient Descent

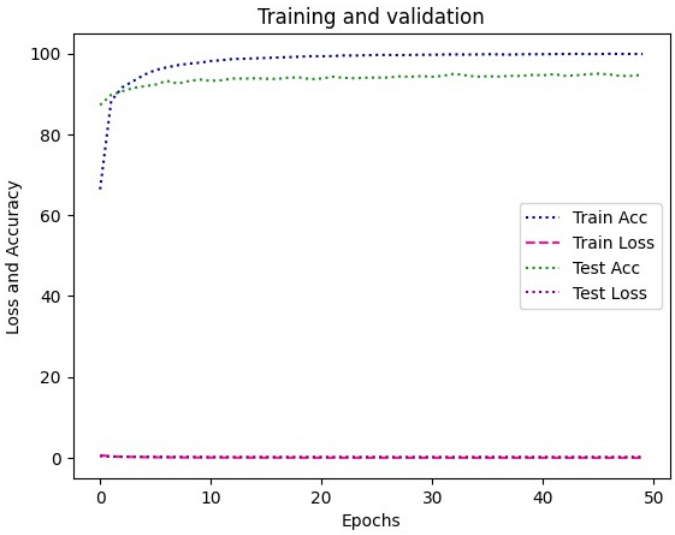

Fig. 4: Training Loss and Accuracy, Validation Loss and Accuracy

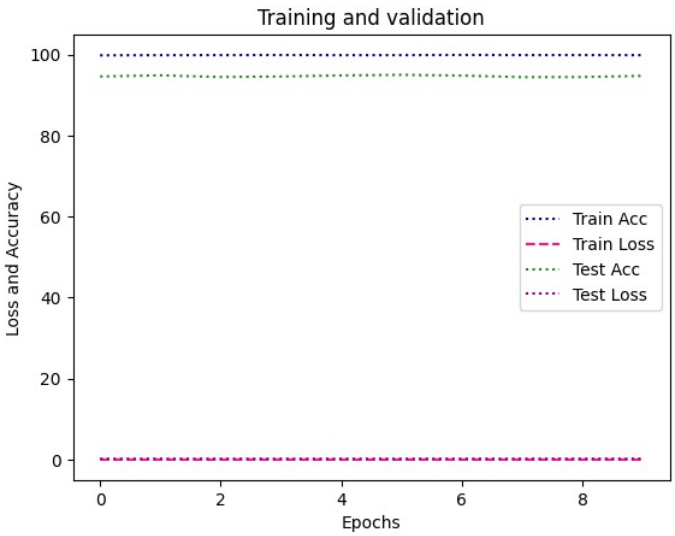

Fig. 5: Training Loss and Accuracy, Validation Loss and Accuracy of last 10 Epochs

These changes include scaling the photos to 224x224 pixels, leveling the RGB channels, and adding random horizontal and vertical flips to the training data. The CIFAR10 dataset is downloaded after the data modifications. This enables the model to learn from a broader set of samples and, as a result, detect unexpected cases during testing. This can assist prevent overfitting, which occurs when a model learns to perform very well on training data but badly on unobserved data. Normalization, which adjusts pixel values in pictures, is a preprocessing technique that ensures all input characteristics have a comparable data distribution, making the model easier to learn and frequently resulting to improved performance. The dataset is divided into two parts: training and testing, which are then changed using the given transformations. The modified datasets are subsequently loaded into DataLoader objects, which produce batches of data effectively during model training and assessment.

Typically, the pre-trained ResNet18 model is trained on ImageNet, which contains 1000 output classes. As a result, the fully connected layer must be updated in order to adapt the model to the CIFAR10 dataset, which includes ten classes. The method *initialize model* accepts as a parameter the number of classes and configures the ResNet18 model with a modified fully connected layer. The final fully connected layer of the pre-trained ResNet18 model is replaced with a new linear layer (nn.Linear) that corresponds to the number of classes in the CIFAR10 dataset, which is 10. The model is adjusted by changing the fully connected layer to produce a probability distribution across the CIFAR10 dataset's ten classifications. During training, the new fully connected layer will learn to map the extracted features from the ResNet18 backbone to the specific classes of CIFAR10, allowing accurate predictions for this specific job.

III. Results

During the early portion of my study, I trained the model using the Adam optimizer, a popular optimization method noted for its good handling of sparse gradients and adaptive learning rates. Adam adjusts the learning rate for each weight of the neural network based on estimations of the first and second moments of the gradient, which can speed up convergence. The learning rate was set to 0.01, the exponential decay rates for the first and second moment estimations, betas, were set to 0.9 and 0.999, and epsilon, a tiny constant for numerical stability, was set to 1e-08. No regularization, such as weight decay, was used in this initial setup to keep the model from overfitting. This training arrangement produced an accuracy of

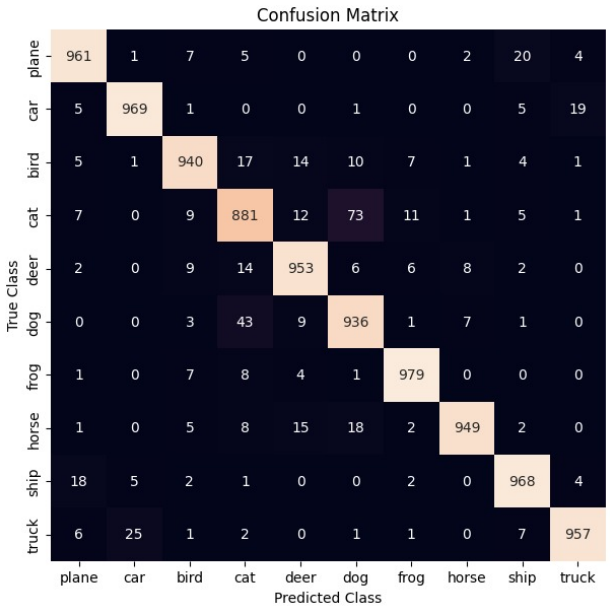

Fig. 6: Confusion Matrix

91.43% as shown in Fig 3a.

To boost the model's performance even more, I changed the optimizer to Stochastic Gradient Descent (SGD) with a lower learning rate of 0.001 and a momentum of 0.9 to suppress oscillations. More crucially, 0.0001 weight decay was used as a sort of L2 regularization. Regularization is a strategy for preventing overfitting by introducing a penalty term into the loss function, which encourages smaller weights and, as a result, a simpler model. Weight decay particularly does this by pushing the model's weights towards (but not precisely to) zero. Following these changes, the model's accuracy increased substantially, reaching 94.98% as shown in Fig 3b. This improvement in accuracy might be ascribed to the weight decay's regularizing impact, which helps to minimize overfitting, as well as the robustness of SGD, especially when paired with momentum, which can negotiate rougher regions of the loss surface and converge to better minima.

We started the training phase with the pre-set weights using a ResNet18 base model and saw an instant accuracy of around 65% as shown in Figure 4. This quickly increased, topping 90% as we moved from the first epoch to the 50th. After passing through all of the epochs, the training accuracy reached an astounding 99.844%. Concurrently, the performance of the model was validated by a number of tests. The first accuracy of these validation tests began at a promising 85% and steadily increased over numerous epochs to reach 94.98%. This indicates our model's effective convergence and performance improvement via the iterative learning process.

Figure 5 depicts a comprehensive study of the final 10 epochs of the training process. The consistency of validation accuracy and loss throughout various epochs is plainly visible. The lack of large variations in these measures clearly suggests that our model avoids the typical issue of over-fitting, highlighting its resilience and good applicability to unknown validation data.

Finally, the confusion matrix shown in Figure 6, which maps the predicted classes against the actual classes for each of the 10 categories, from 'plane' through 'vehicle,' and so on, finishing with 'truck,' demonstrates the model's performance. The primary diagonal of the confusion matrix, which reflects situations where the anticipated and actual classes align, clearly performs well. This confirms that our model is correctly recognizing and categorizing the various categories.

IV. Conclusion

Finally, this term project illustrates that fine-tuning pre-trained weights may greatly enhance ResNet-18 performance on CIFAR-10. To attain better results, this method may be applied to additional deep learning architectures and datasets. The study emphasizes the value of using pre-trained models as a foundation for constructing new models, which can save time and money while enhancing accuracy. Overall, this effort advances the field of machine learning and computer vision by giving a practical strategy for increasing the performance of deep learning models on picture classification tasks.

REFERENCES

[1] V. Chouhan, S. K. Singh, A. Khamparia, D. Gupta, P. Tiwari, C. Moreira, R. Damaševičius, and V. H. C. De Albuquerque, "A novel transfer learning based approach for pneumonia detection in chest x-ray images," *Applied Sciences*, vol. 10, no. 2, p. 559, 2020.

[2] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, *et al.*, "Recent advances in convolutional neural networks," *Pattern recognition*, vol. 77, pp. 354–377, 2018.

[3] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE transactions on neural networks and learning systems*, 2021.

[4] P. Korfiatis, T. L. Kline, D. H. Lachance, I. F. Parney, J. C. Buckner, and B. J. Erickson, "Residual deep convolutional neural network predicts mgmt methylation status," *Journal of digital imaging*, vol. 30, pp. 622–628, 2017.

[5] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM computing surveys (CSUR)*, vol. 51, no. 5, pp. 1–42, 2018.

[6] J.-G. Lee, S. Jun, Y.-W. Cho, H. Lee, G. B. Kim, J. B. Seo, and N. Kim, "Deep learning in medical imaging: general overview," *Korean journal of radiology*, vol. 18, no. 4, pp. 570–584.

[7] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: review, opportunities and challenges," *Briefings in bioinformatics*, vol. 19, no. 6, pp. 1236–1246, 2018.

[8] M. S. Sallam, A. L. Asnawi, and R. F. Olanrewaju, "Diabetic retinopathy grading using resnet convolutional neural network," in *2020 IEEE Conference on Big Data and Analytics (ICBDA)*, pp. 73–78, IEEE, 2020.

[9] 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*, pp. 770–778, 2016.