Image Classification with Neural Networks: An Experimental Study

Robert Nasuti
University of Colorado at Colorado Springs
rnasuti@uccs.edu

Abstract

This paper presents an exploration into the domain of image classification using artificial neural networks. We investigate the performance of custom CNN architectures and compare them with widely-recognized networks like VGG16 and ResNet50 on popular datasets including MNIST, CIFAR-10, and Tiny Imagenet.

Introduction

Image classification, a fundamental task in the domain of computer vision, involves categorizing an input image into one of several predefined classes. With the advent of deep learning, convolutional neural networks (CNNs) have been at the forefront of achieving remarkable performance on this task. In this study, we explore the capabilities of custom CNN architectures and benchmark them against renowned models on three datasets: MNIST, CIFAR-10, and Tiny Imagenet.

Approach

Datasets

MNIST MNIST is a large database of handwritten digits. It contains 60,000 training images and 10,000 testing images, each of size 28x28 pixels.

CIFAR-10 CIFAR-10 consists of 60,000 32x32 color images in 10 classes, with 6,000 images per class. There are 50,000 training images and 10,000 test images.

Tiny Imagenet Tiny Imagenet is a subset of the larger ImageNet dataset, containing 200 classes. Each class has 500 training images, 50 validation images, and 50 test images.

Models

1-Layer CNN Our custom 1-layer CNN architecture primarily consists of a convolutional layer followed by a dense layer, as illustrated in Figure 1.

LeNet-5 Inspired by the classic LeNet-5 architecture, our adapted model is illustrated in Figure 2.

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

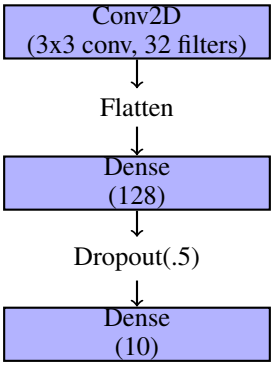

Figure 1: 1-Layer CNN architecture.

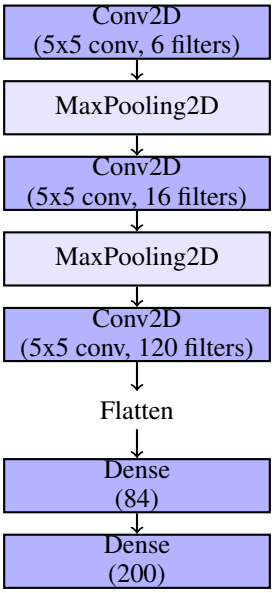

Figure 2: Customized LeNet-5 architecture.

Dataset & Model	Pre-processing
1-layer-cnn MNIST	Reshape to (60000, 28, 28, 1), Normalize
1-layer-cnn CIFAR-10	Normalize
1-layer-cnn & LeNet-5 Tiny Imagenet	Resize to 64x64, Batch size: 128
VGG16 MNIST	Resize to (32, 32), Normalize
VGG16 CIFAR-10	Normalize
VGG16 Tiny Imagenet	Resize to 64x64, Batch size: 64
ResNet50 MNIST	Resize to (32, 32), Normalize
ResNet50 CIFAR-10	Normalize
ResNet50 Tiny Imagenet	Resize to 64x64, Batch size: 64

Table 1: Pre-processing steps for each dataset and model.

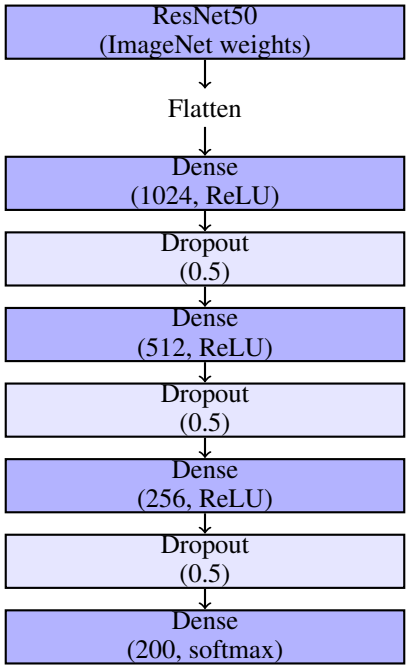

Figure 3: Modified ResNet50 architecture for Tiny ImageNet.

VGG16 and ResNet50 VGG16 and ResNet50 are widely-recognized architectures that have been extensively utilized in various computer vision tasks, especially image classification, due to their robustness and reliability. VGG16 is known for its simplicity, using only 3x3 convolutional layers stacked on top of each other in increasing depth. ResNet50 leverages a deep architecture with skip connections to prevent the loss of information through the network's depth.

Modified ResNet50 for Tiny ImageNet For the Tiny ImageNet dataset, we utilized a modified version of the ResNet50 model, integrating additional dense and dropout layers to manage the complexity and prevent overfitting. The architecture is illustrated in Figure 3.

Experimental Setup

In the experiments, we utilized the Adam optimizer and categorical crossentropy as the loss function, which is suitable for the multi-class classification problems presented by our datasets. The models were trained using an Nvidia 4090 GPU, leveraging its computational capability to efficiently train the deep learning models. The operating system used was Ubuntu 22.04 LTS, and models were implemented using Keras with a TensorFlow backend.

Specifically, the activation function for hidden layers was set to ReLU (Rectified Linear Unit) to introduce non-linearity into the model, while the output layer utilized a softmax activation function to produce a probability distribution over the target classes. To find the optimal batch size and learning rate for each model-dataset combination, several runs were performed with varied parameters, ensuring that model learning was stable and converged efficiently. In most scenarios, convergence was observed after approximately 20 epochs.

The datasets underwent a series of preprocessing steps, which were critical to ensure the consistency of input data and to enhance the training efficiency and model performance. The specific preprocessing steps for each dataset and model are detailed in Table 1.

Results

In this section, we present the performance of our models on the respective datasets in terms of accuracy. A comprehensive visual comparison of the accuracy achieved by each model on different datasets is provided in Figure 4.

1-Layer CNN:

- MNIST: 0.9852 maximum accuracy on the validation set.
- CIFAR-10: 0.6343 maximum accuracy on the validation set.
- Tiny ImageNet: 0.005 maximum accuracy on the validation set.

LeNet-5:

- Tiny ImageNet: 0.1737 maximum accuracy on the validation set.

VGG-16:

- MNIST: No pre-training, fine-tuned for 20 epochs, maximum accuracy of 0.9941 observed.
- CIFAR-10: No pre-training, model failed to learn after 20 epochs, resulting in a maximum accuracy of 0.1 against the validation set and lower on the training set.
- Tiny ImageNet: No pre-training, model failed to learn after 20 epochs, maximum accuracy of 0.005 against the validation set and lower on the training set.
- CIFAR-10 with ImageNet weights: Froze the base model and fine-tuned for 5 epochs, then unfroze bottom 4 layers and trained for another 20 epochs, achieving a maximum accuracy of 0.7433.
- Tiny ImageNet with ImageNet weights: Froze the base model and fine-tuned for 5 epochs, then unfroze bottom 4 layers and trained for another 20 epochs, achieving a maximum accuracy of 0.4229.

ResNet50:

- MNIST: Did not freeze any layers, fine-tuned for 20 epochs, achieving a maximum accuracy of 0.9932 against the validation set.

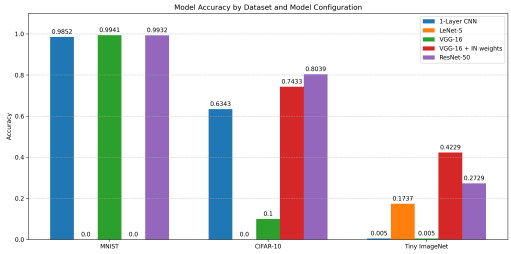

Figure 4: Comparison of classification accuracies across different neural network models and datasets. The models, of increasing complexity, are evaluated on three datasets: MNIST, CIFAR-10, and Tiny ImageNet. Accuracies are depicted as bar heights, and each model's performance on a dataset is represented by a distinct bar, facilitating direct visual comparison. Notably, model performance varies significantly across different datasets. A measurement value of 0.0 signifies that the respective dataset was not utilized for the evaluation of the corresponding model. In such instances, the absence of a recorded accuracy metric should not be misconstrued as a zero-accuracy outcome, but rather as an indication that the model was not subjected to testing against that specific dataset.

- CIFAR-10: Did not freeze any layers, fine-tuned for 20 epochs, achieving a maximum accuracy of 0.8039 against the validation set.
- Tiny ImageNet: Did not freeze any layers, fine-tuned for 20 epochs, achieving a maximum accuracy of 0.2729 against the validation set.

Observations

Building upon the results presented in the previous section, this section aims to provide insights and draw observations from the performance metrics of various models across different datasets.

- The 1-layer CNN was sufficient for MNIST but not for Tiny ImageNet, indicating a necessity for deeper architectures for more complex datasets.
- The LeNet-5 model performed surprisingly well on Tiny ImageNet, given its relatively simple architecture.
- VGG-16 was able to learn MNIST from scratch but struggled with CIFAR-10 and Tiny ImageNet, which was unexpected given the complexity of the network.
- Fine-tuning VGG-16 and ResNet-50 on MNIST and CIFAR-10 resulted in early overfitting, followed by a rapid increase in validation performance before plateauing, potentially exhibiting the "Grokking" phenomenon as described in "GROKKING: Generalization Beyond Overfitting on Small Algorithmic Datasets." (Power et al. 2022)
- ResNet-50 demonstrated overfitting when used alone, which is indicated by a disparity in validation and training data performance (see Figure 5). Introducing dropout layers as a form of regularization mitigated overfitting but resulted in "spiky" validation performance, as can be seen in Figure 6, warranting further investigation.
- Freezing layers during the training of ResNet-50 seemed to slow or prevent the network from learning. Initially, unfreezing only 4 layers, then 25, and finally all of them was attempted. Unfreezing the entire model yielded the best performance, suggesting that perhaps due to the "skip architecture" of ResNet-50, freezing layers might not be the most effective approach for this particular architecture.

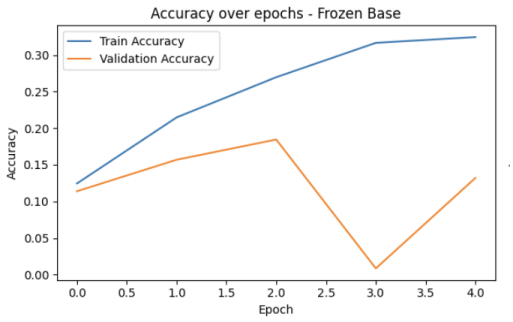

Figure 5: Training and validation accuracy of ResNet-50 on Tiny ImageNet, illustrating overfitting. The graph demonstrates a growing divergence between training and validation accuracy as epochs progress, indicative of the model memorizing the training data and losing generalization on the validation data.

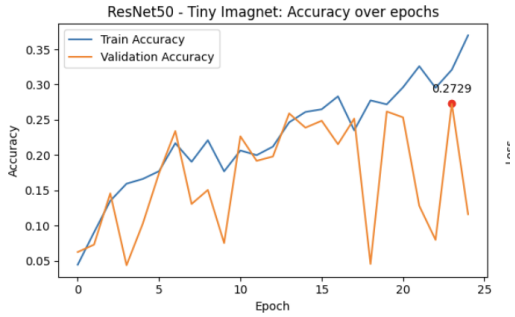

Figure 6: Training graph of ResNet-50 on Tiny ImageNet showing spikiness in validation performance.

State of the Art in Computer Vision

Recent advancements in the field of computer vision, particularly concerning image classification, have presented novel methodologies and optimizations that have pushed the boundaries of model accuracy and efficiency.

The **Lion** optimizer, for example, has demonstrated remarkable success, achieving 91.1% fine-tuning accuracy on ImageNet. Notably, Lion is engineered to amalgamate the advantages of both Adam and SGD optimizers, balancing rapid convergence with stable training (Chen et al. 2023).

Another noteworthy development is the **Contrastive Captioner (CoCa)** model, which achieved a noteworthy

91.0% top-1 accuracy on ImageNet with a fine-tuned encoder. CoCa incorporates a hybrid pretraining strategy, integrating contrastive loss and captioning loss, offering the model capabilities of both contrastive approaches like CLIP and generative methodologies like SimVLM (Yu et al. 2022).

Lastly, the concept of **Model Soups** offers a novel perspective on model optimization. Instead of discarding models from hyperparameter tuning sessions, it proposes averaging the weights of multiple fine-tuned models, which has empirically improved accuracy without increasing inference time. This methodology has seen particular success when fine-tuning large pre-trained models like CLIP, ALIGN, and a ViT-G pre-trained on JFT (Wortsman et al. 2022).

Conclusion and Future Work

This study offered an in-depth exploration into the realm of image classification using neural networks, investigating the performance of custom CNN architectures and comparing them against widely-recognized networks like VGG16 and ResNet50 on popular datasets. The experimental results, observations, and insights derived from this exploration can serve as a foundation for further research and optimization in the domain of image classification using deep learning models.

In future work, exploring additional architectures such as EfficientNet, exploring additional regularization techniques for handling complex datasets like Tiny ImageNet, and investigating further into the phenomenon of "Grokking" may offer further insights and improvements in the field of image classification using neural networks.

Acknowledgments

I'd like to thank ChatGPT (ChatGPT 2023) for assistance with formatting, editing, and LaTeX support throughout this assignment.

References

ChatGPT. 2023. Conversations and Assistance in Formatting, Editing, and LaTeX. Available at OpenAI.

Chen, X.; Liang, C.; Huang, D.; Real, E.; Wang, K.; Liu, Y.; Pham, H.; Dong, X.; Luong, T.; Hsieh, C.-J.; et al. 2023. Symbolic discovery of optimization algorithms. *arXiv preprint arXiv:2302.06675*.

Power, A.; Burda, Y.; Edwards, H.; Babuschkin, I.; and Misra, V. 2022. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*.

Wortsman, M.; Ilharco, G.; Gadre, S. Y.; Roelofs, R.; Gontijo-Lopes, R.; Morcos, A. S.; Namkoong, H.; Farhadi, A.; Carmon, Y.; Kornblith, S.; and Schmidt, L. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvari, C.; Niu, G.; and Sabato, S., eds., *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, 23965–23998. PMLR.

Yu, J.; Wang, Z.; Vasudevan, V.; Yeung, L.; Seyedhosseini, M.; and Wu, Y. 2022. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*.