Image Classification
for the CIFAR-10 dataset

Kenny Aranda, Jose Riquelme Orgiles, Richfield Quist, and Matthew Ross
CAP4630 – Intro to Artificial Intelligence – University of North Florida
n01513222@unf.edu, n01534785@unf.edu, n01041942@unf.edu, and n01502347@unf.edu

Abstract

Image classification is a fundamental problem in artificial intelligence and machine learning. Image classification is significant because it dramatically expands the abilities of computers. The benefits of image classification range from medical diagnosis to autonomous driving. In this report, we present research on the implementation of image classification using convolutional neural networks (CNNs). For this project, we will be using the CIFAR-10 dataset, which consists of 60,000 32x32 colored images with 10 classes such as frogs, birds, cats, ships, etc. Even though the dataset can be solved efficiently, it can serve as a foundation for understanding and practicing the creation, assessment, and application of convolutional deep-learning neural networks for image categorization.

Introduction

The demand for image classification has significantly increased with the recent surge in artificial intelligence. Convolutional neural networks are one of the predominant image classification algorithms. In this paper, we investigate the effectiveness of convolutional neural networks for image classification. The purpose for the project is to further our understanding of image classification and explore different methodologies to improve the performance of CNNs.

The project is important because image classification serves a crucial role in many important fields. Image classification is used for autonomous driving, security, medical diagnosis, and much more. Improving image classification can save lives, enhance business operations, increase productivity, and improve overall quality of life. Therefore, research in image classification is extremely important to the improvement of society.

Related Work

Extensive research was conducted in machine learning algorithms and image classification. We primarily conducted research through resources provided within the course.

The lectures and slides taught us all the necessary fundamentals on machine learning. These fundamentals include supervised and unsupervised learning, classification, regression, and regularization. The lectures built a foundation and allowed us to understand the theory of machine learning. Furthermore, the lectures went into detail about convolutional neural networks, explaining the theoretical concepts and the reasoning behind their use for image classification.

Additionally, a hands-on assignment was created to introduce a simpler version of image classification. The assignment mainly differs from the project in that it was grayscale and colored. Despite these differences, the assignment showed the basics of implementing image classification using a convolutional neural network. From there, we were able to understand each segment of code, and apply these concepts to the project.

Aside from material provided within the course, external research was also conducted. The article "How To Develop a CNN from Scratch for CIFAR-10 Photo Classification" showed the creation of a basic image classification model with the CIFAR-10 dataset. The article highlights important information about the initial development of the model and methodologies that can improve it. Furthermore, many of the core concepts were explained in detail, allowing our group to gain a better conceptual understanding of the project. The article primary benefitted the project with the improvement of our model. Namely, we used regularization and data augmentation.

Methodologies

For this project, the CIFAR-10 dataset will be used to train and test our model. The CIFAR-10 dataset consists of 60,000 32x32 colored images with 10 classes. 50,000 of these images are training images, while the other 10,000 are testing images. Each image correlates with exactly one class. Additionally, to build the program, we used the Keras deep learning API to assist us in building our image classification model.

We first preprocessed the data received from the CIFAR-10 dataset and loaded the dataset into Keras. After preprocessing was completed, that data was divided into training sets and testing sets. The training set contains 50,000 in-color images, each 32x32 pixels. After that, we labeled the classes from 0 to 9 in the following order:

- 0: airplane
- 1: automobile
- 2: bird
- 3: cat
- 4: deer
- 5: dog
- 6: frog
- 7: horse
- 8: ship
- 9: truck

Now that the data is ready (see Figure 1), we built our convolutional neural network model that will serve as our baseline model and will establish a minimum model performance to which all of our other models can be compared, as well as a model architecture that we can use as the basis of study and improvement.

We are going to adopt the fundamental architectural principles of the VGG models, our baseline model is built with 2 blocks. In short, the model extracts features from the images, performs regularization, and connects the layers. The model has a relatively simple architecture that entails the sequential arrangement of convolutional layers featuring 3×3 filters, followed by a max-pooling layer. To maintain consistency in feature map dimensions, padding is applied to the convolutional layers. These layers are organized into blocks, which can be replicated, with an increase in the number of filters per block as the network deepens, typically starting at 32, 64, 128, and 256 for the initial four blocks but still performing to expectations. We assess models with 1, 2, 3, and 4 blocks for performance comparison. We experimented with two additional models, one with 3 blocks and another with 4 blocks. ber of correct classifications.

Figure 1: Sample of images in the CIFAR-10 dataset

Exploring Alternative Architectures: ResNet and Inception Models

While our current research predominantly utilizes VGG-style models, further studies could benefit from considering alternative architectures such as ResNet and Inception. These models have shown significant promise in handling image classification tasks with complex datasets due to their unique structural advantages:

ResNet (Residual Networks): This architecture introduces the concept of skip connections, which help in alleviating the vanishing gradient problem by allowing gradients to flow through an alternate pathway during backpropagation. This can be particularly advantageous for training deeper networks efficiently.

Inception Models: Known for their inception modules, these models employ convolutional operations at various scales concurrently within the same module. This setup allows the model to capture information at multiple scales and complexities, potentially improving performance on diverse datasets like CIFAR-10.

Including these architectures in future research could provide insightful comparisons and might offer improvements in both accuracy and training stability over the standard VGG configurations used in our current experiments. This exploration would enrich our understanding of how different architectural choices can be optimized for specific tasks in image classification.

Experiment and Results

Having established the baseline model with four blocks, we are set to test adjustments to the model and training algorithm, our main objective is to improve the performance. What we did is used regularization techniques, first, we used Dropout Regularization which is a straightforward technique that will randomly drop nodes out of the network, fostering regularization as the remaining nodes adjust to compensate for the removed ones. To implement Dropout, we need to add new Dropout layers to the model, with the extent of node removal specified as a parameter. We will integrate Dropout layers following each max-pooling layer and the fully connected layer, employing a consistent dropout rate of 20% (retaining 80% of the nodes).

Then, we incorporated Weight Regularization, this technique involves updating the loss function to penalize the model based on the magnitude of the model's weights. This produces a stabilizing effect, since heavy weights can lead to a model that is overly complex and unstable, whereas lighter weights tend to a more robust and generalized model.

However, to implement it to our convolutional and fully connected layers we need to establish the “kernel_regularizer” parameter and define the chosen regularization type. We will employ L2 weight regularization, this is a standard approach for neural networks with a regularization factor of 0.0001. We combined these two techniques into the 4-block VGG model, suggesting it as our final model.
Our first model: 1-block VGG style architecture convolutional neural network model, got an accuracy of 0.8739 shown in Figure 2.

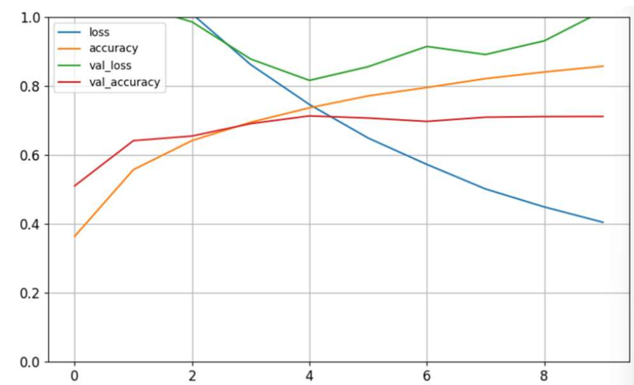

Figure 2: 1-block VGG accuracy graph

Our next model: 2-block VGG style architecture convolutional neural network model, got an accuracy of 0.8574, if we compare it with our first model we can notice that the second model’s accuracy is slightly less than the first one.

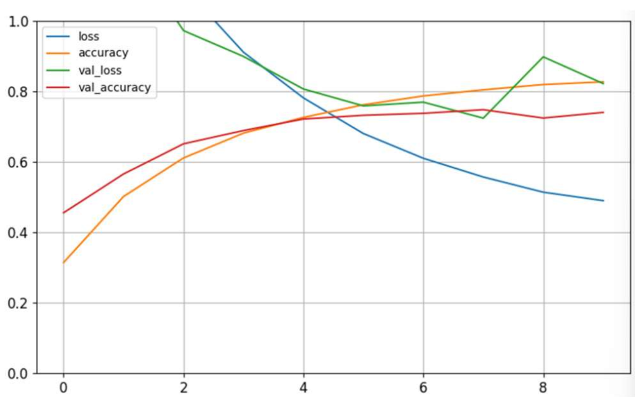

Figure 3: 2-block VGG accuracy graph

Our third model: 3-block VGG style architecture convolutional neural network model has an accuracy of 0.8274, having less accuracy than the first and second model.

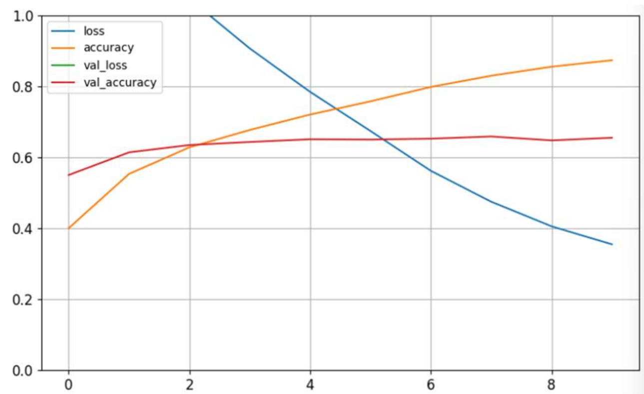

Figure 4: 3-block VGG accuracy graph

Our fourth model has an accuracy of 0.8382 as shown below in Figure 5. This accuracy is slightly higher than the 3-block model accuracy but less than the 1 and 2-block model accuracy, however, we are going to apply the Regularization techniques to this model.

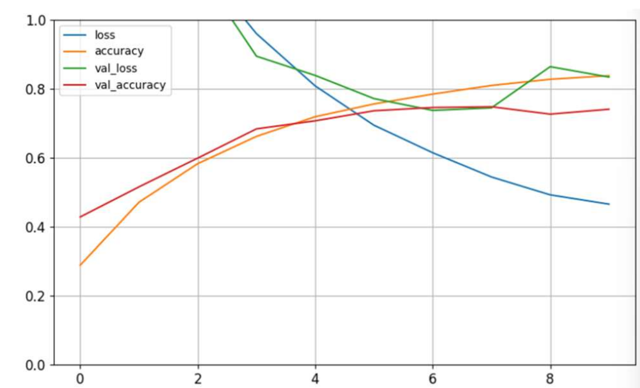

Figure 5: 4-block VGG accuracy graph

When we apply weight Regularization to the 4-block VGG model we have an accuracy of 0.67 which is the lowest accuracy we have received from this model.
Also, when we apply Dropout Regularization the accuracy received is 0.5940, concluding that both Regularization techniques drop the accuracy of the 4-block model.

Deeper Analysis of Model Performance

As we have observed, the CIFAR-10 dataset presents unique challenges that demand robust image classification solutions. Our experiments have demonstrated varying performances across different architectures of the VGG-style model, particularly as the number of convolutional blocks changes. Here, we provide a detailed analysis of how model complexity influences both accuracy and computational efficiency.

Accuracy vs. Model Depth: Our results indicate a trend where the model with one convolutional block achieved the highest accuracy (87.39%), and as additional blocks were introduced, there was a slight decline in accuracy. This suggests that while additional depth can theoretically enhance feature extraction capabilities, for the CIFAR-10 dataset, it might lead to overfitting or the model becoming too complex for the relatively simple images in the dataset. Furthermore, the increase in depth may not be capturing additional useful features but rather noise, which detracts from the model's performance.

Computational Efficiency: An increase in the number of convolutional blocks leads to a substantial increase in the number of parameters that need to be trained, which directly impacts the training time and computational resources required. The single-block model was not only the most accurate but also the most efficient in terms of computation time. This is a crucial consideration in practical applications where both accuracy and speed are essential. For instance, in real-time image processing scenarios such as video surveillance for security purposes, faster processing times are vital.

Trade-off Between Depth and Performance: It is essential to find a balance between the depth of the model and the performance it delivers. Our findings suggest that a deeper network does not always translate to better performance for all datasets. The CIFAR-10, being comprised of low-resolution images, might be efficiently modeled with simpler and shallower networks rather than very deep architectures.

Implications for Future Experiments: Based on these insights, future experiments could explore the optimization of layer configurations, considering not just the depth but also the arrangement and type of layers (e.g., the introduction of skip connections as seen in ResNet architectures) that might help in mitigating the diminishing returns on accuracy with increased depth while maintaining or even reducing the computational load.

By incorporating these findings into our ongoing research, we can better tailor the architectural choices to specific datasets and applications, optimizing both the effectiveness and efficiency of our image classification models.

References

Brownlee, J. 2020. How to develop a CNN from scratch for CIFAR-10photo classification. https://machinelearningmastery.com/how-to-develop-a-cnn-from-scratch-for-cifar-10-photo-classification/. Accessed: 2024-04-28