DRAFT

Training a ConvNet on the CIFAR-10 Dataset

Oscar Nilsson
Jacob Hernberg

Abstract

This report explores techniques to improve the accuracy of image classification models using the CIFAR-10 dataset. The objective is to construct a robust network architecture based on the VGG model and enhance its performance through regularization and optimization strategies. The report discusses the project objectives, importance of high accuracy in image classification, and the systematic approach taken to improve model performance. Various regularization techniques such as dropout, weight decay, and data augmentation are explored, along with the integration of batch normalization. Different optimisation strategies and learning rate schedulers are also tested. The target accuracy is set at 88% which was exceeded.

1. Introduction

In the ever-evolving landscape of deep learning and computer vision, the quest for improving the accuracy of image classification models remains paramount. This report delves into a comprehensive exploration of techniques to address this challenge, focusing on the CIFAR-10 dataset — a benchmark for image classification tasks. The core objective of this project is to construct a robust baseline network architecture, leveraging the principles of the VGG network, and progressively augment its performance through regularization and optimization strategies. **The target grade for this project is E.**

1.1 The Problem

The problem at hand involves constructing a baseline network architecture, initially derived from the VGG model, with the aim of solving the CIFAR-10 dataset. This baseline network will consist of three VGG blocks, forming a powerful foundation for subsequent enhancements. Our initial training phase will employ the SGD optimizer with momentum and no regularization, aiming to achieve results close to 73% accuracy. Enhancing the VGG model was done with strategies suggested in the tutorial, but also with techniques not covered in the tutorial.

1.2 Why it is Important

Achieving high accuracy in image classification tasks is vital for numerous applications, ranging from autonomous vehicles to medical diagnosis and content recommendation systems. The CIFAR-10 dataset, with its diverse range of images across ten classes, poses a unique challenge to the machine learning community. Solving this dataset serves as a litmus test for the effectiveness of various deep learning techniques.

The significance of this project lies in its systematic approach to improving model performance through regularization and optimization. By progressively introducing dropout, weight decay, data augmentation, and batch normalization, we aim to push the performance envelope. The target accuracy of 88% represents a substantial improvement over the baseline, and reaching this milestone could have far-reaching implications for practical image classification tasks.

1.3 Preview of Results

Where the tutorial was followed similar results were obtained as those that had been specified. The results were not exactly the same as the training and testing are stochastic processes which means that some deviation is expected. Where the tutorial was not followed we were able to improve upon the baseline model developed using the tutorial using different regularization strategies. We also found that some regularization strategies worsened the baseline model.

In essence, this report embarks on a journey to unravel the intricate nuances of deep learning regularization and optimization strategies, seeking to not only meet but surpass the target performance. The findings and insights derived from this research endeavor promise to contribute significantly to the ever-advancing field of image classification in the realm of artificial intelligence and machine learning.

2. Related Work

The work which this project is centered on outlines the construction of a baseline VGG-based network architecture for solving the CIFAR-10 dataset. This baseline network consists of three VGG blocks. Subsequently, various regularization techniques are introduced independently to the baseline network such as dropout, weight decay, and data augmentation. The network is then further refined by combining these regularization strategies and incorporating batch normalization. This comprehensive approach requires longer training but may enable the use of a higher learning rate. The final configuration aims for an 88% accuracy target (Brownlee, 2019).

3. Data

3.1 CIFAR-10

The CIFAR-10 dataset is a widely used benchmark dataset in the field of computer vision and machine learning. It is designed for the task of image classification. The CIFAR-10 dataset consists of labeled images. Each image is a 32x32 pixel color image, meaning it has three color channels (red, green, and blue). The dataset was created by the Canadian Institute for Advanced Research (CIFAR). It was collected and compiled for the purpose of advancing research in machine learning and computer vision. The dataset was first released in 2009 as an extension of the earlier CIFAR-100 dataset, which contains 100 classes. The CIFAR-10 dataset contains a total of 60,000 images. These images are divided into 10 distinct classes or categories, with each class containing 6,000 images. The dataset is evenly balanced, meaning that each class has an equal number of samples. These classes are commonly used for image classification tasks and include objects like airplanes, cars, birds, cats, and more. To use the data for training and testing the CNN model developed in this project we normalize the pixel values, by rescaling them to the range [0,1]. This involves first converting the data type from unsigned integers to floats, then dividing the pixel values by the maximum value.

3.2 Neural Network Architectures for CIFAR-10 Classification

In broad terms, a basic convolutional neural network, when coupled with an effective training approach, can attain accuracy levels ranging from 80% to 90% on the CIFAR-10 dataset. Meanwhile, traditional models like ResNet and DenseNet, which have been around for a while, have demonstrated the ability to achieve accuracy levels spanning from 90% to 95%. On the other hand, modern classic models like ViT have shown the potential to surpass the 95% accuracy mark (Franky, 2022).

4. Methods

Solving the complex challenge of improving image classification accuracy on the CIFAR-10 dataset necessitates a systematic and comprehensive approach. This methodology outlines the steps and reasoning behind each stage of our approach, detailing why each choice is appropriate for enhancing the baseline VGG-based network's performance.

4.1 Baseline Network Construction

The objective was to establish a foundation for subsequent enhancements by creating a baseline VGG-based network. The choice of VGG-based architecture is grounded in that it was what was suggested in the blogpost. Why this was chosen for the tutorial might lie in its proven effectiveness in image classification tasks. This architecture's depth and layer organization provide a robust starting point. To construct the deep learning model Keras was used. Keras is a high-level, deep learning API developed by Google for implementing neural networks. It is written in Python and is used to make the implementation of neural networks easy. To begin with, the network is trained with a stochastic gradient descent optimizer along with momentum, omitting regularization initially. This approach is adopted to gauge the network's raw performance potential, which was expected to reach around 73%.

4.2 Regularization Exploration

For this part the objective was to systematically assess the impact of different regularization techniques on network performance. To mitigate overfitting, dropout was introduced, regularly applied throughout the network. The target performance here was an accuracy of ~83%. Weight decay was then introduced as a regularization technique to penalize large weights, further reducing overfitting. The target performance here was an accuracy of ~72%. Finally, horizontal flipping and x-y translation shifts were applied to the training data to augment the dataset, enhancing the model's ability to generalize. The target performance was an accuracy of ~84%.

4.3 Regularization Strategies Combination

The primary goal of this phase was to investigate the impact of multiple regularization techniques, alongside the integration of batch normalization. Specifically, we combined dropout, data augmentation, and batch normalization to create a more resilient model. This strategic combination aimed to enhance training stability and efficiency, potentially enabling the utilization of a higher learning rate. The selection of dropout and data augmentation as the chosen regularization strategies for combination was grounded in their individual performance superiority during isolated testing. Our ultimate objective at this stage was to achieve an accuracy of approximately 88%. The model that

emerged from this phase, enriched through this synergy of techniques, served as the foundation for further experimentation and refinement.

4.4 Exploration Beyond Tutorial

To extend the investigation beyond the boundaries of the tutorial and gain deeper insights into network behavior. We first examined the effect of normalizing input data to have zero mean and standard deviation, in contrast to the traditional [0, 1] range normalization employed in the tutorial. We then moved on to replacing the SGD + momentum optimizer with Adam and AdamW to examine whether these alternatives lead to improved performance and/or faster convergence. After this various learning rate scheduling strategies, including warm-up + cosine annealing, step decay, and cosine annealing with re-starts, were tested to determine their impact on training dynamics.Finally, the order of applying dropout and batch normalization is altered to assess whether BatchNorm before dropout affects performance. Additionally, the complementary nature of Dropout and BatchNorm is explored to determine if having both in the network is superior to a network with just one of these regularization techniques. The baseline performance for all these experiments was ~88%, as this was the performance achieved by the network developed in the tutorial.

By methodically progressing through these stages and exploring beyond the tutorial's scope, we aim to gain a deep understanding of the factors influencing the network's performance and contribute valuable insights to the field of deep learning, particularly in the context of image classification tasks. This methodology ensured a structured and rigorous approach to solving the complex challenges presented by the CIFAR-10 dataset.

5. Experiments

The experiments ran focused on adding different types of regularizations to see how they impacted the performance of the network. Important to note at this stage are the aspects of the network that remained the same throughout the experiments. The model was always trained for 100 epochs and the batch size always remained 64. The full CIFAR-10 training set and test set were always used for training and evaluation respectively. Moreover, the most basic version of the model always consisted of 3 VGG-blocks.

5.1 Baseline Network with No Regularisation

The first experiment was to observe the results of the network without regularization. The baseline for this experiment was approximately 73% classification accuracy. The obtained accuracy was 72.5%. This was deemed to meet the baseline accuracy, as such this experiment was considered a success. The loss and cost curves for this version of the model showed signs of dramatic overfitting. One can see that the model’s performance on the training dataset (blue) continues to improve whereas the performance on the test dataset (orange) improves, then starts to get worse at around 20 epochs (See figure 1, will be added in final submission).

[INSERT FIGURE 1]

.

5.2 Dropout Regularly applied Throughout the Network.

The next experiment was to observe the results of the network with 20% dropout regularly applied throughout the network. The baseline for this experiment was around 83% classification accuracy. The obtained accuracy was 81.55%. Although further away from the baseline than in the first experiment the results of this one were still considered a success and the difference put down to the stochastic nature of training and testing neural networks. The loss and cost curves for this version of the model show that the drastic overfitting in experiment 1 has been somewhat addressed as the model now converges over 40-50 epochs rather than just 20 (See figure 2, will be added in final submission).

[INSERT FIGURE 2]

5.3 Network with Weight Decay.

The third experiment was to observe the results of the network with L2-regularization added to the loss function. The baseline for this experiment was around 72% classification accuracy. The obtained accuracy was 76.74%. Here the baseline was exceeded by almost four percent. The baseline accuracy was derived from the tutorial, where the accuracy for their version of the same network was 72.5%. It is possible that the model performed as well as it possibly could in this run and as such can be put down to the stochastic nature of this network. It could also have meant that weight decay is a far more effective regularization strategy than was initially thought. Regardless, the impact weight decay has on the network is less than that which 20% dropout regularly applied throughout the network has. The loss and cost curves tell a similar story. Weight decay does lead to a small reduction in overfitting but the reduction is not as large as the one caused by dropout (See figure 3, will be added in final submission).

[INSERT FIGURE 3]

5.4 Network with data augmentation.

In the fourth experiment augmentation techniques was used to add more training data for each class to see if it could improve performance by increasing training diversity. With the added augmented datapoints, the model reached 83,72% test accuracy, slightly beating the target of 83% accuracy, and an overall large improvement from the baseline of 72,5% accuracy, an increase of over 11 percentage points. The loss and cost curves for this version of the model show that… (See figure 4, will be added in final submission).

[INSERT FIGURE 4]

5.5 Network with Dropout, Data Augmentation, and Batch Normalisation.

In the fifth experiement, a combination of dropout, data augmentation and batch nomralisation was implemented in the network. The target accuracy was 88% and the combination of regularization techniques improved accuracy to 87,82%. The combination of regularization methods increased performance from standalone use of dropout and data augmentation, both previously mentioned reaching test accuracy results of 81,55% and 83,72% respectively. The loss and cost curves for this version of the model show that… (See figure 5, will be added in final submission).

[INSERT FIGURE 5]

5.6 Changing Normalization

The next experiment was to observe the results of normalizing the data to have 0 mean and standard deviation 1 instead of normalizing the data to be in the range [0, 1]. The baseline for this experiment was around 88% classification accuracy. The obtained accuracy was 88.53%. This makes it seem as though the change in normalization strategy improved the network. As such we chose to keep this normalization moving forward. The loss and cost curves for this version of the model show that… (See figure 6, will be added in final submission).

[INSERT FIGURE 6]

5.7 Changing Optimizer

In this experiment observations were made regarding the results of changing the method of model optimization. Up until this point in the project the network had only used SGD + momentum. Here this was changed to first ADAM and then ADAMW. The baseline for this experiment was once again approximately 88% classification accuracy. The obtained accuracy for ADAM was 89.39% and 89.13% for ADAMW. Once again the changes made seem to have improved the performance of the network. As ADAM was the better performing optimizer this was the optimisation strategy chosen to move forward with. The loss and cost curves for this version of the model show that… (See figure 7, will be added in final submission).

[INSERT FIGURE 7]

5.8 Introducing Learning Rate Schedules

The eighth experiment was to observe the effect of introducing a learning rate schedule. Three different learning rate schedules were tested: warm-up + cosine annealing, step decay, and cosine annealing + restarts. The baseline for each experiment was approximately 88% classification accuracy. Warm-up + cosine annealing resulted in 89.72% classification accuracy. Step decay produced a classification accuracy of 87.5%. Finally, cosine annealing + restarts resulted in 89.63% classification accuracy. Based on these results, cosine annealing seemed to help improve the model. Additionally, warm-up seems to improve the model more than what restarts do. As such we chose to keep cosine annealing + warm-ups as the learning rate schedule. The loss and cost curves for this version of the model show that… (See figure 8, will be added in the final submission).

[INSERT FIGURE 8]

5.9 Complementary Effects of Batch Normalisation and Dropout

In the final experiment complementary effects of batch normalization and dropout was investigated. n previous experiments, only batch normalization before dropout has been performed. Adam was used for optimisation and warm-up with cosine annealing was used as learning rate schedule since this resulted in the best accuracy in the latest tests (see 5.7 and 5.8). By changing the order to do dropout before batch normalization resulted in test accuracy of 88.64%, compared to 89.72% for the other way around. Waiting to do batch normalization after the dropout seemingly worsened the result by almost one percentage point. Thereafter, training with batch normalization and dropout one at a time was

tested to measure the individual performance of the regularizers. Solely using batch normalization reached a test accuracy of 89.25%, while only using dropout resulted in 85.53% accuracy. If choosing to train a model with only one of these reguarization techniques, batch normalization should be the obvious choice given the test results and that all other model settings are the same as those in the experiment. Using dropout only slightly improved performance compared to only implementing batch normalization, and only it was done after batch normalization, as dropout before batch normalization was inferior to batch normalization implemented alone. The loss and cost curves for this version of the model show that… (See figure 9, will be added in final submission).

[INSERT FIGURE 9]

6. Conclusion

In conclusion, Dropout and data augmentation solely implemented on the first baseline model both increase the models performance by around 10%. Batch normalization when removed caused the model performance to decrease by around 5%. As such, of the regularization strategies that have been tested, these three were the most impactful. It was also seen that the model can be improve further, although incrementally, by replacing SGD + momentum with ADAM, introducing warm-up + cosine annealing as a learning rate schedule, and normalizing the data to have 0 mean and standard deviation 1 rather than normalizing it to the range [0, 1]. All these regularization strategies together produce a CNN capable of ~90% classification accuracy. Another take-away from this project is that weight decay might not be well suited for CNNs. This was an interesting result as in other projects this has always been included as a regularization strategy purported to help the generalisability of the machine learning model. Considering that the past projects that the authors of this paper have done mostly concerned MLPs, the conclusion drawn from the results here are that weight decay is one of the worst regularization strategies that can be used to improve the performance of CNNs.

Based on some related work that has been covered in preparation for this project, CNNs and similar architectures are able to achieve 99% classification accuracy on the CIFAR-10 dataset (Franky, 2022). As such, future projects looking to investigate CNN performance on the CIFAR-10 dataset could look to add to the model created in this project in order to be able to replicate the 99% classification accuracy that other researchers have been able to achieve. It would also be interesting to see research done comparing the effect of weight decay on MLPs compared to CNNs as this was a slightly confounding result of the research conducted herein.

7. References

Brownlee, J. (2019, May 13). How to Develop a CNN From Scratch for CIFAR-10 Photo Classification. Machine Learning Mastery. https://machinelearningmastery.com/how-to-develop-a-cnn-from-scratch-for-cifar-10-photo-classification/

Franky. (2022, September 1). Once Upon a Time in CIFAR-10: The Good, The Bad, And The Ugly About Some Technical Tutorials. Medium. https://franky07724-57962.medium.com/once-upon-a-time-in-cifar-10-c26bb056b4ce