BASENET: MODIFYING THE PARAMETERS OF BASIC CONVOLUTIONAL NETWORK [1] [1] [1]

Alec Braynen

ABSTRACT

Convolutional neural networks (CNNs) are a type of deep learning algorithm frequently used for image classification and computer vision tasks. The CIFAR-10 dataset is a commonly used image dataset for training and testing deep learning neural networks. Currently, state of the art performance on the CIFAR-10 dataset is ~99.5 *[1]* In this paper however, we aim to explore how changing the parameters of a baseline CNN implementation affects its accuracy and loss on the CIFAR-10 dataset. The baseline CNN implementation achieved an accuracy of around ~70%. Changing various parameters (epochs, loss functions, optimizers), either negligibly affected this accuracy, or made the accuracy worse. However, adding regularization, was found to increase the accuracy by 1-2%. Investigating further into the effects of regularization on the accuracy of CNNs may lead to fruitful results.

1. INTRODUCTION

Image classification and visual object recognition are important problems in the field of artificial intelligence and computer science, specifically computer vision. Implementing these computer vision algorithms effectively greatly affects society; self-driving cars, bionic eyes and smartphones all depend on computer vision in nontrivial ways.

CNNs are a revolutionary solution to the problem of computer vision and vision related AI problems [2]. This neural network is most simply, composed as a series of convolutional blocks/layers, followed by poolings layer(s) into a fully connected layer(s). [3]

This neural network architecture and it's advances in computer vision, is, in no small part, a contributor to the current explosion of deep learning in the scientific and corporate communities; their rate of deployment has exploded in the fields of agriculture, radiology, medicine and many more.[4]

State of the art architectures of CNNs have achieved 99% accuracy on the CIFAR-10 dataset [1]. While we don't obtain such results, in this paper we explore how adjusting the architecture of a basic CNN affects the accuracy of the network on the CIFAR-10 dataset. Various adjustments and experiments were done such as changing the number of epochs, the optimizer function, the loss function, the learning rate and batch size. These changes either greatly diminished or negligibly affected the accuracy of the network. Regularization however, was able to increase the accuracy by 1-2% and may be a worthwhile area of research to pursue further.

2. RELATED WORKS

Much scientific research focuses on improving the performance of neural networks on tasks and datasets. Currently, the top performing neural network on the CIFAR-10 data set benchmark, is the Big Transfer (BiT) network. [5] BiT achieves an accuracy of 99.37% with an architecture that uses pre-training to replace task-specific routines. The network is first trained once on a large dataset and then from this training, smaller tasks are able to be tackled. [6]

EfficientNet is another neural network with state of the art performance on CIFAR-10. EfficientNet obtains an accuracy result of 98.9% with an architecture that was built with a neural architecture search and scaled with compound coefficient φ. [7] This compound coefficient acts as a reliable guide for effectively scaling neural networks within the constraints of the network environment (hardware, dataset, etc.)

In [8], the authors create/use a new type of pooling layer called Fractional Max Pooling(FMP). Instead of simply using a 2x2 or 3x3 pooling layer, the authors use layers with parameters such as 3/2, 6/4 and 10/7 layers. Using FMP along with dropout, the authors achieved an accuracy of 96.53% on the CIFAR-10 dataset.

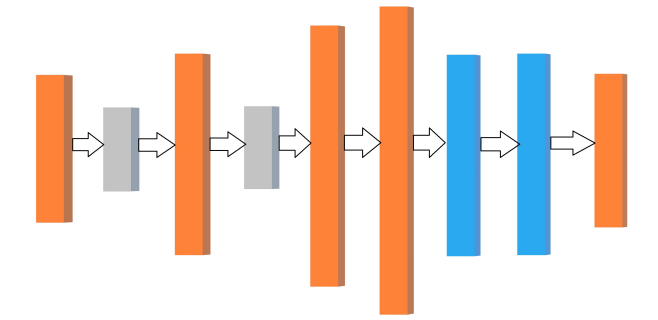
Another work that demonstrates high accuracy on the CIFAR-10 dataset, achieves an accuracy of 96.2% using Wide Residual Networks [9]. Instead of making the network deeper (adding more sequential layers to feed data through), the authors stacked more

layers at particular stages of the network (making it wider).

And interestingly, in [10], the authors achieve an accuracy of 95.59% by replacing the pooling layers of the CNN with convolutional layers with stride two and data augmentation.

3. METHOD

To implement our base CNN, we lay a convolutional layer, followed by a max pooling layer, followed by a convolutional layer, then another max pooling layer, followed by two more convolutional layers, then two dense layers followed finally by the output layer (layout shown in Fig 1). Each convolutional layer's neurons doubles in size of the previous layers' neurons (The first convolutional layer has 32 neurons, the second 64 neurons, etc.)



Conv Layer 1 -> Pooling Layer -> Conv Layer 2 -> Pooling Layer -> Conv Layer 3 -> Conv Layer 4 -> Dense Layer -> Dense Layer -> Output Layer

The dense layers have 2048 neurons, and the output layer has 10 neurons (for the ten classes).

The network uses the Adam optimizer function and has a learning rate of 0.001. Each layer uses the ReLU activation function, except for the output layer which uses the SoftMax function and the pooling layers which have no activation function. The network is trained via 25 epochs and uses a batch size of 128.

3.1 Baseline Results

This baseline CNN recorded a loss of 1.928 and an accuracy of 71.3%.

4. EXPERIMENTS AND RESULTS

From our baseline implementation of the CNN, we then carried out several experiments, changing various parameters to discover how the accuracy and loss of our CNN would change.

We experimented with the number of epochs (1, 5, 10, 15, 20, 25), the optimizer functions (Adam, SGD, RMSprop), the loss function (Sparse Categorical Cross Entropy, Poisson, KL Divergence), the learning rate (0.1, 0.01, 0.001), the batch sizes (128, 256, 512) and regularizing the dense layers (dropout, l1 regularization, l2 regularization).

4.1 Results

From these experiments, the parameters with the highest accuracy were 72.68% with: 25 epochs, the Adam optimizer, Sparse Categorical Entropy loss, a learning rate of 0.001, a batch size of 128, l1 and l2 regularization with a dropout of 0.2, and a regularization of 0.001.

Additionally, the parameters with the lowest loss were: 25 epochs, the Adam optimizer, Sparse Categorical Entropy loss, a learning rate of 0.001, a batch size of 128, l1 and l2 regularization with a dropout of 0.2, and a regularization of 0.001. The loss for these parameters was 1.2082.

The worst set of parameters were: 15 epochs, the Adam optimizer, KL Divergence loss, learning rate of 0.001, and a batch size of 128. The accuracy for this experiment was 9.996% and the loss was 20.7233.

Figure 2 below shows the performance of BaseNet against some other neural networks in the CIFAR-10 AI benchmark.

Method	Accuracy
GPipe: Efficient Training of Giant Neural Networks using Pipeline Parallelism [11]	99%
ShakeDrop Regularization [12]	97.69%
Improved Regularization of Convolutional Neural Networks with Cutout [13]	97.44%
Enhanced Image Classification With a Fast-Learning Shallow Convolutional Neural Network[14]	75.86%
PCANet: A Simple Deep Learning Baseline for Image Classification? [15]	79.60%
BaseNet	**72.68%**

Figure 2: Table of BaseNet Accuracy against Neural Networks in the CIFAR10 benchmark

4.2 Charts and Data Results

The figures below show how various parameters affected the accuracy of BaseNet.

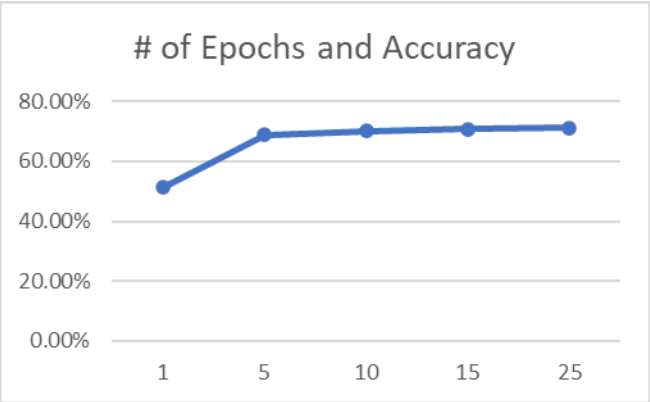


Figure 3: # of Epochs vs Accuracy Percentage

As shown in Figure 3, the accuracy of BaseNet increases sharply from 1 epoch to 5 epochs. Then it increases more subtly as the number of epochs increases from 5 to 25, with 25 giving the highest accuracy: 71.3%.

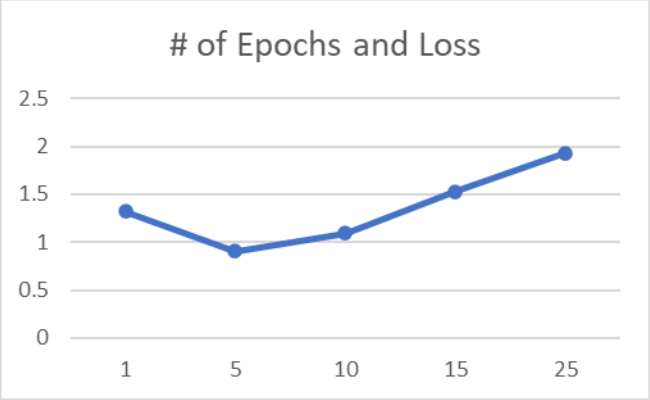


Figure 4: # of Epochs vs Loss

In Figure 4 however, we see that 5 epochs translates to the smallest loss, with the amount of loss increasing as the number of epochs increase. This data guided us to use 15 epochs for a lot of our testing since the accuracy of 15 epochs was very close to the accuracy of 25 epochs, and the loss of 15 epochs (1.5263) was smaller than the loss of 25 epochs (1.928).

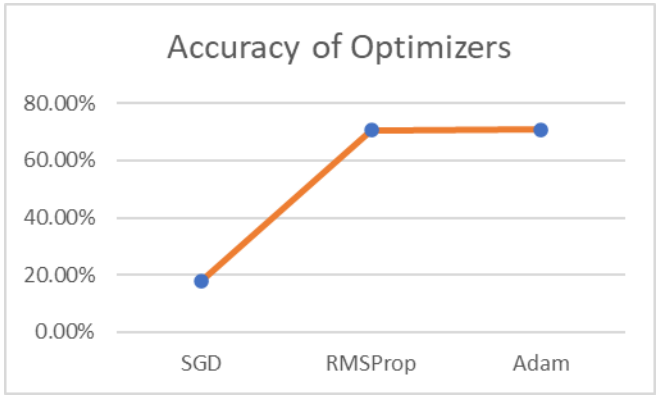


Figure 5: Optimizers (SGD, RMSProp and ADAM) and the Accuracy Percentage

In Figure 5 we see that the SGD optimizer gives very poor performance on the CIFAR-10 dataset with an accuracy of 18.06%, while RMSProp gives a slightly lower performance than Adam, with accuracies of 70.68% and 70.86% respectively.

In figure 6, we show that there is a drop off in accuracy as the learning rate becomes slower, with 0.001 giving the best accuracy (71.24%) on the dataset.

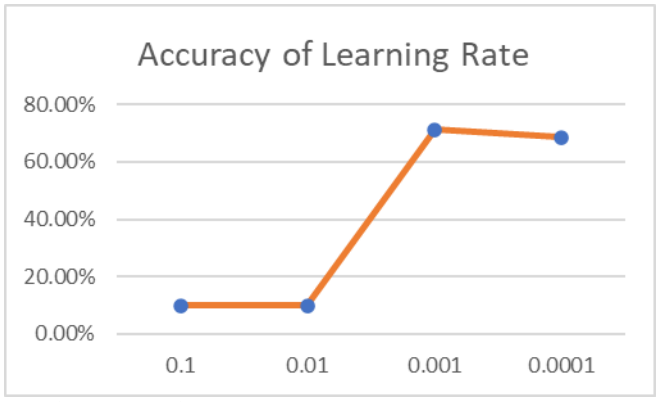


Figure 6: Learning Rate (0.1, 0.01, 0.001) vs Accuracy Percentage

Finally, Figure 7 shows how regularization improves the accuracy of the network, leading to the biggest change out of all the parameters (1-2%). This led to our highest accuracy of 72.68%.

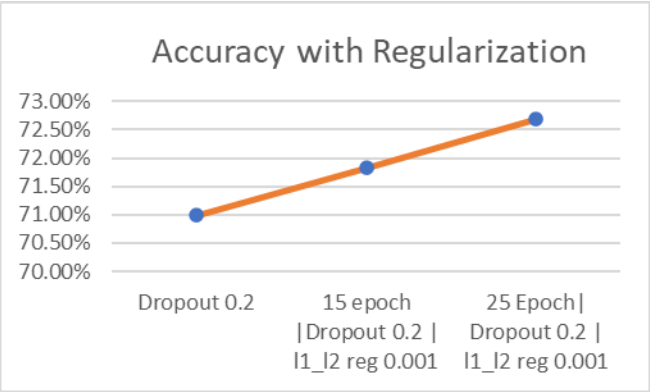


Figure 7: Regularization Parameters and the effect on Accuracy

5. CONCLUSION

CNNs continue to represent a giant leap forward for the fields of AI and Computer Vision. In this paper, we explored how changing parameters affected the accuracy of our basic CNN, BaseNet, on the CIFAR-10 dataset. Despite not approaching the results that state-of-the-art CNNs achieve, this paper and our experiments show how modifying easily changeable parameters can improve or degrade the performance of a CNN. Furthermore, this paper, along with the works [5],[6],[7],[8], and [9] can serve as guidance to those hoping to improve their CNN performance. Ultimately, we discovered that adding regularization to a CNN is an effective way to improve the accuracy of the network.

6. REFERENCES

[1] Alexey Dosovitskiy, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *ICLR 2021*, 2020.

[2] A. T. &. R. Cusack, "CONVOLUTIONAL NEURAL NETWORKS AS A MODEL OF VISUAL ACTIVITY IN THE BRAIN: GREATER CONTRIBUTION OF ARCHITECTURE THAN LEARNED WEIGHTS," in *Bridging AI and Cognitive Science*, 2020.

[3] S. Canavan, "CONVOLUTIONAL NEURAL NETWORKS," Tampa, 2022.

[4] J. Raitoharju, "Chapter 3 - Convolutional neural networks," in *Deep Learning for Robot Perception and Cognition*, Academic Press, 2022, pp. Pages 35-69.

[5] "Benchmarks AI," [Online]. Available: https://benchmarks.ai/cifar-10.

[6] Alexander Kolesnikov, "Big Transfer (BiT): General Visual Representation Learning," Google Research, Brain Team, 2019.

[7] "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," 2019.

[8] B. Graham, "Fractional Max-Pooling," 2015.

[9] Sergey Zagoruyko, "Wide Residual Networks," 2017.

[10] Jost Tobias Springenberg, "STRIVING FOR SIMPLICITY: THE ALL CONVOLUTIONAL NET," 2014.

[11] "GPipe: Easy Scaling with Micro-Batch Pipeline Parallelism," 2018.

[12] YOSHIHIRO YAMADA, "ShakeDrop Regularization for Deep Residual Learning," 2018.

[13] Terrance DeVries, "Improved Regularization of Convolutional Neural Networks with Cutout," 2017.

[14] Mark D. McDonnell, "Enhanced Image Classification With a Fast-Learning Shallow Convolutional Neural Network," 2015.

[15] Tsung-Han Chan, "PCANet: A Simple Deep Learning Baseline for Image Classification?," 2014.