Deeper Networks for Image Classification

Dimitrios Mylonas (180326363)

School of Electronic Engineering and Computer Science Queen Mary University of London, UK

Deep neural networks have been of great interest to researchers and they have found great success in task like image classification. Many architectures have been proposed to enable such deeper networks. This paper explores 2 such implementations, VGG-16 and ResNet-18, applying them to the task of image recognition and proposing improvements based on architectural changes and alternative training strategies. The evaluation of the models is performed on the MNIST handwritten digits and CIFAR-10 datasets.

I. Introduction

The power of convolutional neural networks (CNN) [1] combined with current advances in processing power are partly responsible for the rise of Deep Learning. Models based on CNNs are particularly dominant in the field of computer vision, where tasks require the model to learn spatial patterns of an image. CNNs are especially good at this, requiring little pre-processing of an image to identify such patterns.

Over the past years, countless researchers have published papers attempting to build networks such as these to perform the task of image classification better, with the baseline being the performance of their network in the ImageNet Large Scale Visual Recognition Challenge. [2]

In the last decade, through this challenge, it has been identified that the network's depth is crucial to the performance of the network, with a huge part of research devoted to finding ways to build deeper and deeper networks. This however is a challenging task as adding more and more layers doesn't always have a positive effect on the accuracy, as vanishing gradient effects come into play.

In this paper two approaches of training such deep networks are explored, the VGG [3] and ResNet [4] architectures. The models are first explained in detail, and following that a critical analysis is performed

comparing them and proposing improvements on the existing models and their training process. Then, using the MNIST [1] and CIFAR-10 [5] datasets, experiments are performed on the original and improved versions of the models.

II. Model Description

A. VGG

In this paper, the authors introduce an architecture based on stacked 3x3 convolutional layers to reduce the parameters significantly which allows for a deeper network. These layers are meant to smartly replace the large 7x7 layers but preserve the important information from an image using the minimum receptive field to capture information about left/right, up/down and center. This change makes the decision function of the model more discriminative as there are now three ReLU non linearity units present instead of one. Furthermore, 1x1 convolutional layers were utilised in some versions of the model, preserving the dimensionality while also increasing the non linearity of the decision function.

All hidden layers of the network make use of the ReLU non-linearity and there are 2x2 max pooling layers with stride 2 within the stacks of layers. An adaptive average pooling layer with a window size of 7x7 pools the outputs of the last convolutional layer which are then passed to three fully connected layers which eventually are responsible for producing the prediction of the image classification.

The VGG network optimises a multinomial logistic regression utilising a size 256 mini batch stochastic gradient descent with 0.9 momentum. Dropout layers of 0.5 are also used to decrease the effects of overfitting of the model.The instability of the learning process was decreased by introducing initialisation techniques such as REF which initialise the weights from a distribution, accelerating training in the process.

B. ResNet

In the ResNet paper, the authors attempt to tackle the degradation problem (decrease in accuracy with deeper networks) by introducing residual networks. The degradation problem occurs because the deeper layers of the network are unable to learn even the identity mappings. This problem is fixed by focusing on optimising a new objective. The new objective of this type of training is to learn a residual mapping of the data. This means that provided the identity mapping can be learnt by the network, extra layers should produce an error no greater than that of a shallower version of the network.

The main idea is that the authors explicitly allow the layers to fit a residual

mapping. Given the underlying mapping is H(x), they let the stacked nonlinear layers fit another mapping F(x) := H(x)x, and the original mapping becomes F(x)+x. They do this because they believe that this mapping is easier optimised than the original mapping. This is done by utilising skip connections, which skip one or more layers during the feed-forward operation. The skip connections are responsible for performing identity mapping and their outputs are added to the outputs of the stacked layers. To avoid any problems with mismatching dimensions during the skipping of layers, a weight matrix is used to perform a linear projection.

The network consists mainly of 3x3 convolutional layers, with a max pooling layer at the end being fed into a dense layer for classification. After each convolutional layer, batch normalisation is applied, followed by a ReLU non-linearity. Similar to VGG, weight initialisation, stochastic gradient descent and the same objective function were used.

III. Critical Review

The aforementioned models have found success and are used extensively in computer vision applications, and researchers are working on using the ideas proposed in these models to improve the models efficiency. Some of these improvements are discussed in this section.

Improvements on the VGG are minimal, with most being small adjustments on the mode, to fit specific tasks. These changes could be for example removing or adding more convolutional layers to fit the complexity of the task's dataset. Other improvements could be adding batch normalisation layers (like they are included in the ResNet architecture), to counter the vanishing/exploding gradient problem.

On the other hand, ResNet has more potential for modifications and a lot of research has gone into improving the residual blocks. These include techniques such as Squeeze and Excitation [6], selective kernels [7] and anti-alias down-sampling [8], as well as label smoothing and dropout.

IV. Improved Models

This section describes the improvements made on the VGG and the ResNet models.

A modification made in both of the improved models is weight decay. This is a regularisation technique that penalises the weights and biases L2 norm to prevent the network from over-fitting. This happens because it forces it keep low values for the weights avoiding the exploding gradient problem.

Another modification that both the improved models use is data augmentation.

For both MNIST and CIFAR-10 the images are resized to 64x64 using bi-linear interpolation. The images are also randomly rotated in a range of 20 degrees, translated horizontally and vertically with a value of 0.2 and scaled randomly with a factor of 0.7 to 1.3. The images are also normalised with the respective datasets means and standard deviations.

Specifically for the VGG-16, batch normalisation layers were introduced before each activation function to fix the inputs to a range. Another change was that the 7x7 adaptive average pooling was changed to a 5x5 adaptive max pooling layer. This greatly reduced the networks parameters, making it more light without losing significant performance in the tasks.

For the improved version of the ResNet, the model makes use of 3x3 convolutional layers rather than a bigger 7x7 layer. This is in line with the principles used in VGG explained before about reducing the receptive field enough so that only the necessary information is encapsulated. This means that the stride was picked appropriately.

V. Experimentation

A. Datasets

MNIST: This dataset consists of 60000 greyscale images of handwritten digits with size 28x28 from numbers 0-9. The training/testing split is 50000-10000 images. The training set was split further into 4:1 training-validation split. Since the images are grayscale, the models were modified so that they can accept single channel inputs (unlike their original implementation intended for the 3 channel ImageNet dataset). Furthermore, the output of the dense layer was modified to 10, which is all the handwritten digits from 1000, which is the amount of classes in ImageNet.

CIFAR-10: This dataset consists of 60000 RGB images of 10 different classes. This time the images are 32x32 size. The last layer was changed to 10 units therefore but the input layer could be left untouched.

B. Training

For the original models of VGG-16 and ResNet-18, a learning rate of 0.001 was used and for every baseline model 30 epochs were used, except for VGG-16 for CIFAR where 50 epochs were used. Stochastic Gradient Descent with momentum of 0.9 was used on a batch size of 128 for all datasets and models.

For the improved models the same parameters are used for the training, with the added caveat that a weight decay is introduced during the optimisation.

VI. Results and Discussion

A. MNIST

Due to the low complexity of the dataset, both original and improved models perform extremely well, even with a small number of epochs. In fact, for all models, the loss reaches approximately 0, and so the MNIST dataset is not the best for testing the performance and effect of changes in architectures.

We can observe that for VGG, the improved version performs better on the MNIST dataset. The baseline model reaches a test accuracy of 99.05 percent, whereas the improved version reaches an accuracy of 99.59 percent. For such an easy task as the MNIST dataset, a 0.6 percent improvement is significant. This could be attributed to the data augmentation performed on the data and the weight decay introduced.

For the original ResNet, the accuracy on the MNIST dataset is 99.31, which suggests it is more powerful than VGG, as we would expect. The improved version of ResNet reaches an accuracy of 99.54, which is a very small change, not worthy of commenting on.

B. CIFAR-10

This dataset is a way better baseline to test our models, as its complexity allows us to analyse the loss and accuracies produced to a better degree. The effects of the improvements on the VGG-16 are clearer when looking at the accuracies on the CIFAR-10 test dataset. The original VGG-16 reaches an accuracy of 76.28 percent, whereas the improved version reaches an accuracy of 82.46 percent, a more than 4 percent increase. This confirms that the improvements made are valid and improve the training of the model.

For ResNet, the positive effects of data augmentation, weight decay and architectural changes are also obvious on the CIFAR-10 dataset. The accuracy of the baseline models reaches an accuracy of 75.46 percent, whereas the accuracy of the improved model reaches 82.55 percent, the biggest increase in performance we've seen so far, meaning the changes in ResNet are significant in improving performance.

VII. Conclusion and future work

This paper successfully implemented the baseline VGG-16 and ResNet-18 models, applying them to the MNIST and CIFAR-10 datasets and proposed and implemented improvements on the models that increased the performance of the models for the task of image recognition. For future work, inception based networks like GoogLeNet can be implemented to compare their perfor-

mance with the currect models. Different, more complex datasets with more than 10 classes could also be used.

References

[1] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, Proceedings of the IEEE **86**, 2278 (1998).

[2] O. Russakovsky *et al.*, International Journal of Computer Vision (IJCV) **115**, 211 (2015).

[3] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition.

[4] K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition.

[5] A. Krizhevsky *et al.*, (2009).

[6] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, Squeeze-and-excitation networks, 2017.

[7] X. Li, W. Wang, X. Hu, and J. Yang, Selective kernel networks, 2019.

[8] R. Zhang, Making convolutional networks shift-invariant again, 2019.