Image recognition on CIFAR10 dataset using ResNet18 and Keras

Kamil Klosowski

November 2018

Contents

1 Introduction

Image recognition is a widely used technique to enable computers to recognise the objects inside a graphical input. It is used by a large amount of tasks including image labelling, image search, medical disease recognition and self driving cars. For a long time the best algorithm to use was a Support Vector Machine with several optimisations. This changed in 2012 when an annual ImageNet Large Scale Visual Recognition Challenge was won by a convolutional neural network [2]. Since then a large progress have been seen in the area.

In this report I have analysed several image recognition techniques with the main focus on the effectiveness of ResNet architecture [1]. I have also included a comparison with a 5-layer fully connected network, 5-layer convolutional network.

2 Methods

2.1 Fully connected network

For the sake of comparison, I have tested the performance of a fully connected neural network with 5 layers (1000, 800, 600, 400, 200 neurons on consecutive layers) and a 10 neuron softmax activation layer. This network started to overfit the augmented dataset after only 20 epochs reaching the maximum validation accuracy of 45%. Even with a relatively low accuracy this neural network presented accuracy similar to that of a SVM algorithm which shows how powerful a tool neural networks are. Because of it's simplicity and lack of normalisation, the network was very unstable with validation accuracy fluctuating +/- 5% even after reducing the learning rate. Further training didn't improve the achieved accuracy.

2.2 Convolutional network

The second type of algorithm that I've tested was a convolutional neural network. CNNs are similar to ordinary dense networks in a way they learn but differ in a structure significantly. Every layer is represented by a collection of feature maps with a certain height, width and depth which is represented by the amount of the maps. Input to the network is also three dimensional with

image having a height, width and 3 depth colour channels. While the convolutional architecture performs noticeably better than fully connected, larger image datasets introduce the need to use deeper networks which results in the appearance of a vanishing gradient problem. Because of how the back propagation algorithm works, the lower layers of the network in deep architectures are very hard to train. This is often addressed by changing the activation function or introducing transfer learning.

The architecture that I've tested consisted of 5 convolutional layers with batch normalisation, 3x3 kernel size and stride 2 followed by 2 fully connected layers with 1000 and 500 neurons and a 10 neuron softmax activation layer.

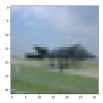
2.3 Residual network

To address the vanishing gradient problem and the slow training speed of traditional convolutional networks, residual neural networks have been introduced [1]. This new architecture attempts to fix the problem by introducing an identity connection between residual block. Each block takes an input from the previous block combined with the identity connection and is usually build with two convolutional layers with batch normalisation and dropout. The architectures vary slightly depending on implementation. This approach allowed for creation of much deeper networks with architectures over 1000 layers deep successfully trained and tested.

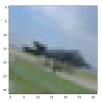
For my residual network I have followed the original ResNet 18 architecture and experimented with hyperparameters and regularisation. I have also tested 3 different depths of the ResNet with 4, 6 and 8 residual block each consisting of convolutional layer with 50% dropout. Each block used batch normalisation and the number of filters doubled every 2 blocks starting with 32 and finishing with 256 for the full ResNet18. The highest improvement was achieved by adding the dropout layers. The usage of L2 normalisation didn't seem to have any effect which was most likely compensated by the usage of decaying learning rate and high batch size of 500. The highest top 1 accuracy of 70% was achieved by both 4 and 6 block versions while 8 layer version seemed to stabilised at 66%, lack of processing power prevented me from training this version for longer. Each version of the network was trained for 500 epoch with the provided dataset and followed by 250 epochs of training on augmented dataset.

2.4 Data Augmentation

To improve the accuracy and ensure a better generalisation of the neural network, I have used several data augmentation techniques [3]. Firstly, I've flipped the images in the horizontal axis, creating mirror images. Then I've rotated both normal and flipped versions by 15 and -15 degrees. To compensate for the loss of data during rotation, I've used the nearest neighbour interpolation. Lastly, I've applied a slight noise to every version of the image. This resulted in dataset size increase from 10.000 to 120.000 samples. After testing with both original and augmented dataset, I've observed that augmentation greatly increased both accuracy and network stability, especially when using deeper residual networks. With final parameters of the network, this approach increased the accuracy from 60 to 70 percent.



(a) Base

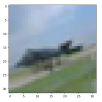


(b) Base +15°

(c) Base -15°

(d) Flip

(e) Flip +15°



(f) Flip -15°

Figure 1: Dataset Augmentation

3 Results

As predicted, the fully connected network was significantly to small to capture the differences between the categories which yielded a result of 45%. While the accuracy is significantly lower than the other tested approaches, the score is still on par or higher than other traditional methods used for image recognition.

The convolutional neural network showed an improvement over the fully connected network and produced a result of 52% accuracy. Because of a similar small size the network started to overfit very early in the training. While increasing the depth of the network would likely produce better results, the time to train the network increases disproportionately because of the vanishing gradient problem.

The residual neural network performed very well producing a 71-72% Top1 accuracy which is a great result. It easily outperformed the other approaches. The accuracy could still be improved for the 8 block architecture with further hyperparameter tweaking and more computational power. While ResNets improve the training time significantly, the amount of computation needed to train deeper networks is still high.

3.1 Comparison

	Accuracy	**Epochs**	**Training time**
Fully connected 5 layers	45%	25	2 min
Convolutional 5 layers	52%	25	3 min
Resnet 4 blocks/ 10 layers	70%	500+250	2 hours
ResNet 6 Blocks/ 14 Layers	71%	500+250	3.5 hours
Full ResNet 18 8 Blocks / 18 Layers	66%	500+250	5 hours

Table 1: Top 1 Accuracy comparison

3.2 Full CIFAR10 Dataset

The tested version of ResNet 18 was able to achieve 61% top1 accuracy and 95.5% top 5 accuracy on full version of CIFAR10.

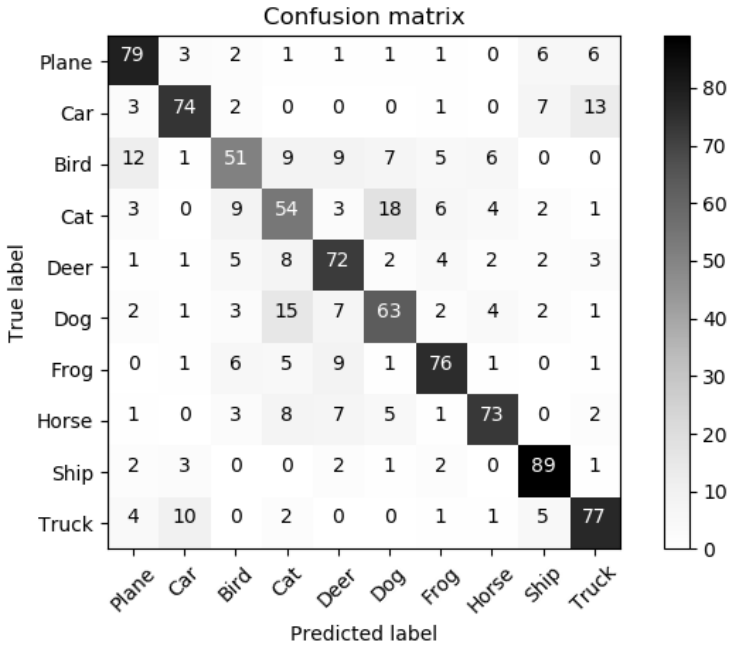


Figure 2: Confusion matrix for the best 6 block 14 layer ResNet

4 Conclusion

Out of the several architectures the ResNet with 8 blocks produced the best results. The accuracy of 71% is in my opinion a satisfactory result. As mentioned, this percentage can be further improved with hyperparameter tweaking which unfortunately requires a lot more expertise than I currently posses. Making this problem worse is the fact that each test of a parameter change takes a long time to assess making methods like grid search ineffective.

Another limiting factor is the size of the dataset. Use of a larger training set would significantly improve the accuracy, especially in the case of the larger networks. Although I have used data augmentation techniques to mitigate this, it introduced a new problem of increased computational complexity which made the training even harder using my hardware. Overall I consider this project a success and I have learned a lot while completing it which in this case is much more valuable then high model accuracy.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[3] Bharath Raj. Data augmentation — how to use deep learning when you have limited data - part 2, Apr 2018.