Structural Compression of ResNet-like Convolutional Neural Networks

Nicholas Kastanos (nk569), Queens' College
Department of Computer Science and Technology
University of Cambridge
Cambridge, CB3 0FD
nk569@cam.ac.uk

Abstract

abstract

1 Introduction

2 Related Work

2.1 ResNet

The residual block first postulated for use in ResNet has become a common-place feature in many subsequent networks (DenseNet, Inception).

2.2 Separable convolutions

Convolution Layers contain a vast majority of the parameters in modern CNNs. By targeting parameter reductions to these layers, the compression can be spread throughout the network. Separable convolutions reduce the number of parameters by separating the convolution into multiple stages through spatially and depthwise separable convolutions. While these convolutions reduces the memory and computation requirements of the system, the reduction in parameters reduces the number of possible kernels explored in training, and the resulting network may be suboptimal.

2.2.1 Spatially separable convolutions

A convolution kernel can be decomposed on its 2D spatial axis, i.e. height and width. Conceptually, the $n \times n$ kernel can be separated into two smaller kernels, a $n \times 1$ followed by a $1 \times n$ kernel. These kernels can be applied in sequential convolutions to obtain the same output shape as the single convolution. These decomposed kernels scale the parameters required by the convolution by a factor $P_s(n)$ (see Equation 1).

Similarly, the multiplication operations of a spatially separated convolution are reduced. For a $M \times M$ input convolved with a $n \times n$ kernel, the number of multipications are reduced by a factor of $M_s(n)$ (see Equation 2).

Equations 1) and 2 show that spatially separable convolutions show computational benefits when $n > 2$.

$$P_s(n) = \frac{2}{n} \tag{1}$$

$$\begin{aligned} M_s(M,n) =& \frac{2}{n} + \frac{2}{n(M-2)} \\ \Rightarrow M_s(n) =& \frac{2}{n}, \text{ where } M >> n \end{aligned} \tag{2}$$

2.2.2 Depthwise separable convolutions

Depthwise separable convolutions separate the spatial convolution from the depth of the filters. This is accomplished by an initial depthwise convolution, followed by a pointwise convolution. The initial depthwise convolution separates the channels of the input and kernel, and convolves them independently. The pointwise convolution is a $N_F \times 1 \times 1 \times N_C$ convolution where N_F and N_C are the number of filters and channels respectively.

The number of parameters $P_d(n)$ and multiplications $M_d(n)$ are reduced by the same factor, which can be seen in Equation 3. Many CNNs have $N_F >> 1$, therefore depthwise convolutions show compression kernel sizes greater than 1.

$$P_d(n) = M_d(n) = \frac{1}{n^2} + \frac{1}{N_F} \approx \frac{1}{n^2} \tag{3}$$

2.3 Quantization and datatype compression

Tensorflow, by default, uses 32-bit floating point precision for its network layers and training. Many resource-constrained devices do not have sufficient memory to use large neural networks, or may not have access to floating-point arithmetic units. Both of these factors can be mitigated by using low-precision integer datatypes, such as 8-bit integers. This effectively reduces the memory required for each parameter by ¼.

3 Methodology

4 Evaluation and results

5 Conclusion