Local Binary Convolutional Neural Networks

Cheng Guan

June 5, 2018

1 Introduction

Deep learning has been overwhelmingly successful in a broad range of applications, such as computer vision, speech recognition / natural language processing, machine translation, bio-medical data analysis, and many more. Deep convolutional neural networks (CNN), in particular, have enjoyed huge success in tackling many computer vision problems over the past few years, thanks to the tremendous development of many effective architectures, AlexNet [2], VGG [5] and Inception [6] to name a few. However, training these networks end-to-end with fully learnable convolutional kernels (as is standard practice) is (1) computationally very expensive, (2) results in large model size, both in terms of memory usage and disk space, and (3) prone to over-fitting, under limited data, due to the large number of parameters.

On the other hand, there is a growing need for deploying, both for learning and inference, these systems on resource constrained platforms like, autonomous cars, robots, smartphones, smart cameras, smart wearable devices, etc. To address these drawbacks, several binary versions of CNNs have been proposed [1, 4] that approximate the dense real-valued weights with binary weights. Binary weights bear dramatic computational savings through efficient implementations of binary convolutions. Complete binarization of CNNs, though, leads to performance loss in comparison to real-valued network weights. In this paper, we present an alternative approach to reducing the computational complexity of CNNs while performing as well as standard CNNs. We introduce the local binary convolution (LBC) layer that approximates the non-linearly activated response of a standard convolutional layer. The LBC layer comprises of fixed sparse binary filters (called anchor weights), a non-linear activation function and a set of learnable linear weights that computes weighted combinations of the activated convolutional response maps. Learning reduces to optimizing the linear weights, as opposed to optimizing the convolutional filters. Parameter savings of at least 9×to 169× can be realized during the learning stage depending on the spatial dimensions of the convolutional filters (3×3 to 13×13 sized filters respectively), as well as computational and memory savings due to the sparse nature of the binary filters. CNNs with LBC layers, called local binary convolutional neural networks (LBCNN)1, have much lower model complexity and are as such less prone to over-fitting and are well suited for learning and inference of CNNs in resource-constrained environments.

Our theoretical analysis shows that the LBC layer is a good approximation for the non-linear activations of standard convolutional layers. We also demonstrate empirically that CNNs with LBC layers performs comparably to regular CNNs on a range of visual datasets (MNIST, SVHN, CIFAR-10, and ImageNet) while enjoying significant savings in terms of the number of parameters during training,computations, as well as memory requirements due to the sparse and pre-defined nature of our binary filters, in comparison to dense learnable real-valued filters.

Related Work The idea of using binary filters for convolutional layers is not new. BinaryConnect [1] has been proposed to approximate the real-valued weights in neural networks with binary weights. Given any real-valued weight, it stochastically assigns +1 with probability p that is taken from the hard sigmoid output of the real-valued weight, and -1 with probability $1-p$. Weights are only binarized during the forward and backward propagation, but not during the parameter update step, in which high-precision real-valued weights are necessary for updating the weights. Therefore, BinaryConnect alternates between binarized and real-valued weights during the network training process. Building upon BinaryConnect [1], binarized neural network (BNN) and quantized neural network (QNN) have been proposed, where both the weights and the activations are constrained to binary values. These approaches lead to drastic improvement in run-time efficiency by replacing most 32-bit floating point multiply-accumulations by 1-bit XNOR-count operations.

2 Forming LBP with Convolutional Filters

Local binary patterns (LBP) is a simple yet very powerful hand-designed descriptor for images rooted in the face recognition community. LBP has found wide adoption in many other computer vision, pattern recognition, and image processing applications [3].The traditional LBP operator operates on image patches of size 3×3, 5×5, etc. The LBP descriptor is formed by sequentially compare the intensity of the neighboring pixels to that of the central pixel within the patch. Neighbors with higher intensity value, compared to the central pixel, are assigned a value of 1 and 0 otherwise. Finally, this bit string is read sequentially and mapped to a decimal number (using base 2) as the feature value assigned to the central pixel. These aggregate feature values characterize the local texture in the image. The LBP

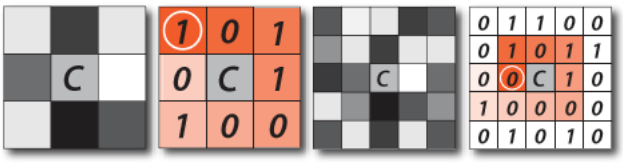

Figure 1: (L-R) 3 × 3 patch and its LBP encoding, 5 × 5 patch and its LBP encoding.

for the center pixel (x_c, y_c) within a patch can be represented as $\text{LBP}(x_c, y_c) = \sum_{n=0}^{L-1} s\left(i_n, i_c\right) \times 2^n$ where i^n denotes the intensity of the nth neighboring pixel, i^c denotes the intensity of the central pixel, L is the length of the sequence, and $s\left(\cdot\right) = 1$ if $i_n \geqslant i_c$ and $s\left(\cdot\right) = 0$ otherwise. For example, a N × N neighborhood consists of $N^2 - 1$neighboring pixels and therefore results in a $N^2 - 1$ long bit string. Figure 1 shows examples of LBP encoding for a local image patch of size 3 × 3 and 5 × 5.

References

[1] M. Courbariaux, Y. Bengio, and J.-P. David. BinaryConnect: Training deep neural networks with binary weights during propagations. In *NIPS*, 2015. 1

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 1

[3] M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen. *Computer vision using local binary patterns*. Springer, 2011. 1

[4] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. XNOR-Net: ImageNet classification using binary convolutional neural networks. In *ECCV*, 2016. 1

[5] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 1