CQNN:
Convolutional Quadratic Neural Networks

Pranav Mantini* & Shishir K. Shah
University of Houston, Department of Computer Science
*pmantini@cs.uh.edu

UNIVERSITY of HOUSTON
DEPARTMENT OF COMPUTER SCIENCE

Abstract

Image classification is a fundamental task in computer vision. A variety of deep learning models based on the Convolutional Neural Network (CNN) architecture have proven to be an efficient solution. Numerous improvements have been proposed over the years, where broader, deeper, and denser networks have been constructed. However, the atomic operation for these models has remained a linear unit (single neuron). In this work, we pursue an alternative dimension by hypothesizing the atomic operation to be performed by a quadratic unit. We construct convolutional layers using quadratic neurons for feature extraction and subsequently use dense layers for classification. We perform analysis to quantify the implication of replacing linear neurons with quadratic units. Results show a keen improvement in classification accuracy with quadratic neurons over linear neurons.

Introduction

A variety of deep learning models have gained popularity for classification and recognition: AlexNet [5], VGGNet [6], DenseNet [4], GoogleNet [7], and ResNet [3]. These models use a Convolutional Neural Network (CNN) at the core. The atomic operation for these models has remained a linear unit. In this work, we pursue an alternative dimension by hypothesizing the atomic operation to be performed by a quadratic unit.

Main Objectives

- We propose Convolutional Quadratic neural networks (CQNN) for representation learning and demonstrate its application for image classification.
- We implement popular image classification architectures that consist of convolutional layers with quadratic neurons as processing units.
- We compare the image classification performance of the CQNN version against the existing architectures.

Convolutional Quadratic Neural Networks

Let $X^T = \{x_1, x_2, ..., x_d\}$ be the input vector with d dimensions, where $\{\}^t$ is the transpose. A neuron performing a linear function is represented as

$$f(X) = WX + b, \tag{1}$$

where $W = \{w_1, w_2, .., w_d\}$ are the weights and b is the bias. A generalized quadratic functions for a neuron can be defined as.

$$q(X) = X'^T W_q X', \tag{2}$$

where $X'^T = \{X^T|1\} = \{x_1, x_2, ..., x_d, 1\}$ is the augmented vector, and

$$W_q = \begin{bmatrix} w'_{1,1} & w'_{1,2} & \cdot & \cdot & w'_{1,d+1} \\ w'_{2,1} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ w'_{d,1} & \cdot & \cdot & \cdot & w'_{d+1,d+1} \end{bmatrix} \tag{3}$$

are the weights. Equation 2 is mathematically equivalent to the quadratic neuron defined in [2].

CQNN:

We propose to use quadratic neurons for representation learning rather than for classification. The idea is to use quadratic neurons for image representation learning and subsequently use linear neurons for classification. We build networks with a combination of quadratic and linear neurons for image classification where the convolutions layers used for extracting image representation are constructed using quadratic neurons, and the dense layers used in classification at the latter stages use linear neurons. Let I be an image and consider a filter of size NXN used in the convolutional layer of CNNs. Let $X^T_{i,j} = \{x_1, x_2, ..., x_{N^2}\}$ be the pixels in the receptive field of the image spanned by the kernel at location (i, j). Then the output of the quadratic neuron is computed as

$$q(X_{i,j}) = X'^T_{i,j} W_q X'_{i,j}. \tag{4}$$

Experimental Design

We design experiments to quantify the capacity of CQNNs to perform image classification and compare them with CNNs. We consider AlexNet and ResNet as base architectures and construct CQNNs with similar.

1. CQNN (Proposed): The neurons in the convolutional layer are replaced with quadratic neurons.
2. CQNNFan: Fan *et al.* [1] proposed a quadratic neuron for neural networks. We build CQNN using this neural function.
$$f(x) = (W_1X + b_1)(W_2X + b_2) + (W_3X + b_3) \tag{5}$$
3. Bilinear Networks (BiCNNs): The bilinear implementation consists of two parallel networks consisting of only convolutional layers. The features extracted from the networks are used to compute an outer product.
4. Horizontally Scaled Networks (HSCNNs): We scale each CNN horizontally (i.e. increase the number of filters) to create horizontally scaled versions, such that they have the same number of parameters as the CQNN versions.

Datasets: We conduct experiments on two image classification datasets, Cifar-10 and Cifar-100.

Results

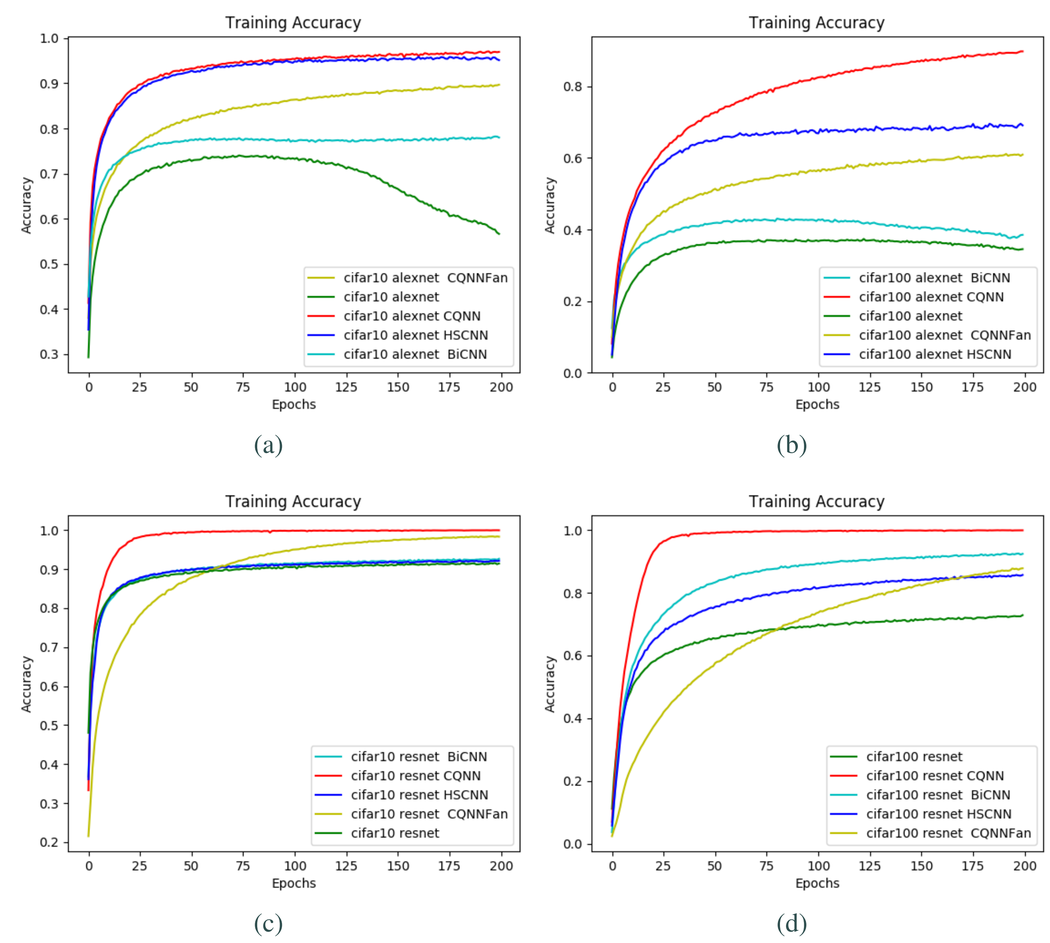

Figure 1: Training Accuracy on Cifar-10 dataset: (a) AlexNet variants; (c) ResNet variants; and Cifar-100 dataset: (b) AlexNet variants; (d) ResNet variants

Dataset	Model	Parameters	Accuracy
Cifar - 10	AlexNet	1.2M	0.63
	AlexNet HSCNN	32.2M	**0.88**
	AlexNet BiCNN	53.2M	0.79
	AlexNet CQNNFan	2.2M	0.87
	AlexNet CQNN	31.4M	**0.88**
	ResNet	.2M	0.85
	ResNet HSCNN	130.6M	0.87
	ResNet BiCNN	0.5M	0.84
	ResNet CQNNFan	0.8M	0.84
	ResNet CQNN	128M	**0.91**
Cifar - 100	AlexNet	1.2M	0.41
	AlexNet HSCNNs	32.2M	0.56
	AlexNet BiCNN	64M	0.41
	AlexNet CQNNFan	2.3M	0.58
	AlexNet Quadratic	31.4M	**0.63**
	ResNet	.2M	0.60
	ResNet HSCNN	130.8M	0.59
	ResNet BiCNN	0.9M	0.51
	ResNet CQNNFan	0.8M	0.55
	ResNet CQNN	128M	**0.66**

Figure 2: Testing accuracy of Existing architectures on Cifar-100 dataset.

Conclusions

- Training Behavior: Quadratic version shows a clear improvement in the training accuracy on both the datasets.
- Testing Accuracy: CQNN AlexNet versions show a 25% increase in the accuracy on Cifar-10 and 22% on Cifar-100. item The ResNet CQNN version shows a 6% improvement in accuracy on both the datasets.
- Experiments on two public datasets, Cifar-10 and Cifar-100 show a clear improvement in training behavior and testing accuracy compared to conventional CNN architecture.

References

[1] Fenglei Fan, Wenxiang Cong, and Ge Wang. A new type of neurons for machine learning. *International journal for numerical methods in biomedical engineering*, 34(2):e2920, 2018.

[2] Fenglei Fan, Hongming Shan, Lars Gjesteby, and Ge Wang. Quadratic neural networks for ct metal artifact reduction. In *Developments in X-Ray Tomography XII*, volume 11113, page 111130W. International Society for Optics and Photonics, 2019.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. Densenet: Implementing efficient convnet descriptor pyramids, 2014.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[7] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.