Code available

Local Binary Convolutional Neural Networks

Felix Juefei Xu [1], Vishnu Naresh Boddeti [2], Marios Savvides [1]

[1] Carnegie Mellon University [2] Michigan State University

felixu@cmu.edu, vishnu@msu.edu, msavvid@ri.cmu.edu

IEEE 2017 Conference on Computer Vision and Pattern Recognition

CVPR July 21-26 HONOLULU 2017

The Goal

The proposed **Local Binary Convolutional Neural Networks (LBCNN)** aims at:

- Statistical efficiency
1. Significant parameter savings: 9x to 169x in the number of learnable parameters.
2. Less prone to overfitting, converges faster, can learn from much fewer training samples.
- Computation efficiency
1. Significant model size savings: 9x to 169x.
2. Significant computation savings: due to sparse binary convolutions.
- On-par performance with standard CNNs (ImageNet, CIFAR-10, MNIST, SVHN)

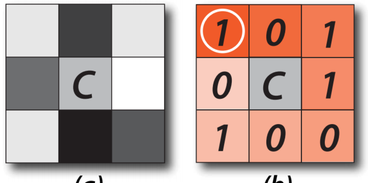
The Inspiration

The **LBCNN** is inspired by the Local Binary Patterns (LBP) descriptor: the LBP is one of the most successful and widely used feature descriptor in computer vision, especially in face recognition.

There are several tuning knobs that lead to different LBP configurations.

- Base $\mathbf{v} = [2^7, 2^6, 2^5, 2^4, 2^3, 2^2, 2^1, 2^0]$
- Pivot
- Ordering of the encoding neighbors

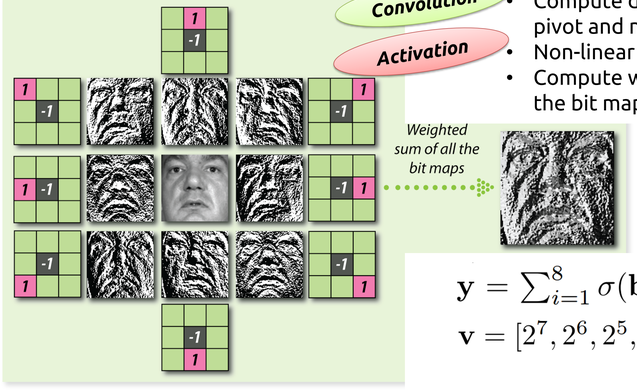
We will generalize these factors in a learnable framework for LBCNN.



(a) (b)

LBP operation is convolutional:

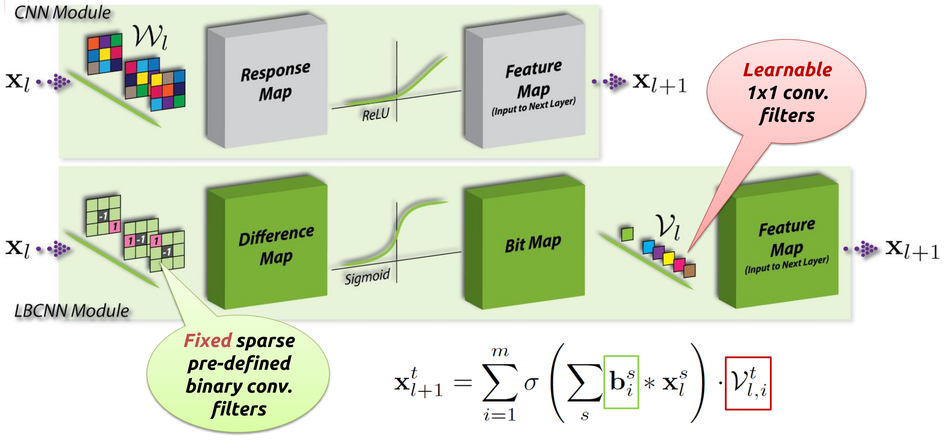
- Compute difference between pivot and neighboring pixels
- Non-linear thresholding
- Compute weighted sum of all the bit maps



$$\mathbf{y} = \sum_{i=1}^{8} \sigma(\mathbf{b}_i * \mathbf{x}) \cdot \mathbf{v}_i$$

$$\mathbf{v} = [2^7, 2^6, 2^5, 2^4, 2^3, 2^2, 2^1, 2^0]$$

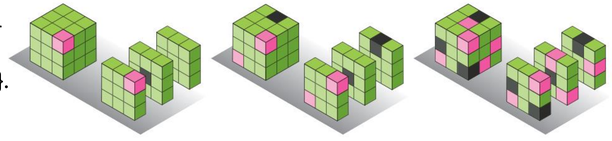
The Approach



$$\mathbf{x}_{l+1}^t = \sum_{i=1}^{m} \sigma\left(\sum_s \mathbf{b}_i^s * \mathbf{x}_l^s\right) \cdot \mathcal{V}_{l,i}^t$$

Generating LBCNN filters:

- Stochastically generated filter weights.
- Sparse binary weights {-1, 0, 1}.
- Efficient convolutions by +/- operations.



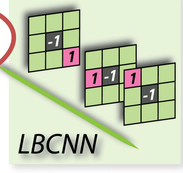
Theoretical analysis

- We have shown theoretically that the proposed LBCNN layer is a good approximation of the standard CNN layer.

Theorem 3.5. *Let* $\mathbf{B} \in \mathbb{R}^{m \times N}$ *be a Bernoulli random matrix with the same subgaussian parameter* c *in (5), and* $\mathbf{x} \in \mathbb{R}^N$ *be a fixed vector and* $\|\mathbf{x}\|_2 > 0$*, with* $N = p \cdot h \cdot w$*. Let* $\boldsymbol{\xi} = \mathbf{B}\mathbf{x} \in \mathbb{R}^m$*. Then, for all* $t \in (0, 1)$*, there exists a matrix* $\mathbf{B}$ *and an index* $i \in [m]$ *such that*

$$\mathbb{P}\left(\xi_i \geq \underbrace{\sqrt{(1-t)}\|\mathbf{x}\|_2}_{>0}\right) \geq 1 - 2\exp(-\tilde{c}t^2 m) \qquad (7)$$

Code Available!



xujuefei.com/lbcnn.html

Carnegie Mellon University

MICHIGAN STATE UNIVERSITY

The Experiments

(1) Experiments (small scale): CIFAR-10, MNIST, SVHN

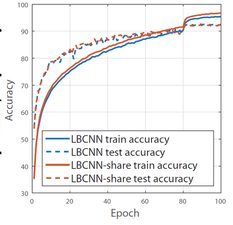
- LBCNN ResNet, CNN ResNet (baseline)
- We have experimented with a 8848-layer LBCNN (NetEverest), trainable on a single nVidia Titan X GPU.

	LBCNN	Baseline	BinaryConnect [6]	BNN [5, 14]	ResNet [12]	Maxout [9]	NIN [23]
MNIST	99.51	99.48	98.99	98.60	/	99.55	99.53
SVHN	94.50	95.21	97.85	97.49	/	97.53	97.65
CIFAR-10	92.99 *(93.66 NetEverest)*	92.95	91.73	89.85	93.57	90.65	91.19

- Sharing fixed local binary filters across all layers.
- Further saving on the model size.

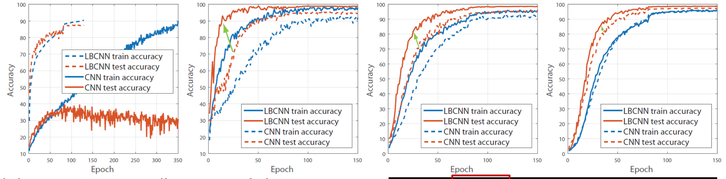
q	16	32	64	128	192	256	384	512
LBCNN	82.74	85.57	88.18	90.70	91.58	92.13	**92.96**	92.09
LBCNN-share	82.70	85.26	87.85	90.26	91.37	91.72	**92.91**	91.83
Baseline	84.13	86.30	88.77	90.86	91.69	92.15	**92.93**	91.87

Table: Classification accuracy (%) on CIFAR-10 with 20 convolution layers and 512 LBC filters on LBCNN, LBCNN-share, and CNN baseline.



(2) Experiments (statistical efficiency):

- Prevents over-fitting. Due to much smaller model complexity. Experiments (L1) are on small-sized CIFAR subset where CNN over-fits.
- Converges faster. Experiments on the FRGC face recognition dataset, and LBCNN converges faster than CNN (R3:10-, 50-, and 100-class).



(3) Experiments (large scale): ImageNet

- Architecture: AlexNet.
- Follow the LR schedule of the CNN (could be improved).
- LBCNN saves 6.622x learnable parameters

	LBCNN	AlexNet (ours)	AlexNet (BLVC) [1]
ImageNet	54.9454	56.7821	56.9

Layers	AlexNet [21]	LBCNN (AlexNet)
Layer 1	$96 \times (11 \times 11 \times 3) = 34,848$	$96 \times 256 = 24,576$
Layer 2	$256 \times (5 \times 5 \times 48) = 307,200$	$256 \times 256 = 65,536$
Layer 3	$384 \times (3 \times 3 \times 256) = 884,736$	$384 \times 256 = 98,304$
Layer 4	$384 \times (3 \times 3 \times 192) = 663,552$	$384 \times 256 = 98,304$
Layer 5	$256 \times (3 \times 3 \times 192) = 442,368$	$256 \times 256 = 65,536$
Total	$2,332,704\ (\sim 2.33M)$	$352,256\ (\sim 0.352M)$

Table: Comparison of the number of learnable parameters in convolutional layers in AlexNet and AlexNet with LBCNN modules. LBCNN saves 6.622x learnable parameters in the convolutional layers.