Compact Bilinear Pooling

Yang Gao[1], Oscar Beijbom[1], Ning Zhang[1,2], Trevor Darrell[1]

Introduction

Bilinear models has been shown to achieve impressive performance on a wide range of visual tasks, such as semantic segmentation, fine grained recognition and face recognition. However, bilinear features are high dimensional, typically on the order of hundreds of thousands to a few million, which makes them impractical for subsequent analysis. We propose two compact bilinear representations with the same discriminative power as the full bilinear representation but with only a few thousand dimensions. Our compact representations allow back-propagation of classification errors enabling an end-to-end optimization of the visual recognition system. The compact bilinear representations are derived through a novel kernelized analysis of bilinear pooling which provide insights into the discriminative power of bilinear pooling, and a platform for further research in compact pooling methods. Extensive experimentation illustrate the applicability of the proposed compact representations, for image classification and few-shot learning across several visual recognition tasks.

Compact Bilinear Pooling

We propose a compact bilinear pooling method for image classification. In a typical case, our method could reduce 250 thousand dimensions required in bilinear pooling to only 4 thousand to 8 thousand dimensions without loss of classification accuracy when finetuned. Remarkably, this indicates a 98% redundancy in the original bilinear feature.

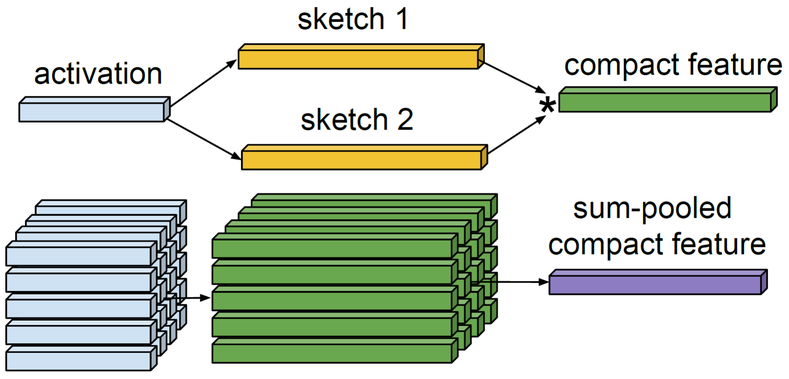

Figure 1. A plot illustrates Compact Tensor Sketch pooling method.

Bilinear Pooling

Bilinear pooling is a method to combine local feature descriptors to a global descriptor by:

$$B(X) = \sum_s x_s x_s^T$$

Where $X = \{x_s | s = 1,2 \cdots |S|, x_s \in R^c\}$ is a set of local descriptors and S is the set of spatial locations in the image. In this paper, we use the last convolutional activations (with ReLU) as the local descriptors.

Compact bilinear pooling from a Kernelized Viewpoint

Suppose we classify the bilinear feature using a linear kernel machine, then the machine compares two images by:

$$\begin{aligned}\langle B(\mathcal{X}), B(\mathcal{Y})\rangle &= \sum_{s\in\mathcal{S}}\sum_{u\in\mathcal{U}}\langle x_s, y_u\rangle^2 \\ &\approx \sum_{s\in\mathcal{S}}\sum_{u\in\mathcal{U}}\langle \phi(x), \phi(y)\rangle \\ &= \langle \sum_{s\in\mathcal{S}}\phi(x_s), \sum_{u\in\mathcal{U}}\phi(y_u)\rangle \\ &\equiv \langle C(\mathcal{X}), C(\mathcal{Y})\rangle,\end{aligned}$$

where

$$C(\mathcal{X}) := \sum_{s\in\mathcal{S}}\phi(x_s)$$

is the **compact bilinear feature** and $\phi(\cdot)$ is some low dimensional projection. We use Random Maclaurin Projection and Tensor Sketch Projection as two $\phi(\cdot)$ functions. Figure 1 shows a schematic plot of Tensor Sketch projections.

Compact Bilinear Pooling Could be Learnt End-To-End

The two compact bilinear pooling methods are both easy to backprop to get derivatives of the input data and derivatives of projection weights. Thus it's easy to fine tune the whole network.

	Compact	Highly discriminative	Flexible input size	End-to-end learnable
Fully connected	✓	✗	✗	✓
Fisher Encoding	✗	✓	✓	✗
Bilinear pooling	✗	✓	✓	✓
Compact bilinear	✓	✓	✓	✓

Table 1. Comparison of several pooling methods for CNN.

Results

We show theoretical and experimental comparisons among our proposed compact bilinear features (RM, TS) and the Fully Bilinear (FB), Fisher Vector (FV) and Fully Connected (FC) pooling layers.

Theoretical Comparisons

	Full Bilinear	Random Maclaurin (RM)	Tensor Sketch (TS)
Dimension	c^2 [262K]	d [10K]	d [10K]
Parameters Memory	0	$2cd$ [40MB]	$2c$ [4KB]
Computation	$O(hwc^2)$	$O(hwcd)$	$O(hw(c + d\log d))$
Classifier Parameter Memory	kc^2 [1000MB]	kd [40MB]	kd [40MB]

Table 2. Dimension, memory and computation comparison among bilinear and the proposed compact bilinear features. Parameters c; d; h; w; k represent the number of channels before the pooling layer, the projected dimension of compact bilinear layer, the height and width of the previous layer and the number of classes respectively. Numbers in brackets indicate typical value when it is applied after the last convolutional layer of VGG-VD model on a 1000-class classification task, i.e. c = 512; d = 10K; h = w = 13; k = 1000. All data are stored in single precision.

Compact Bilinear Pooling Configurations

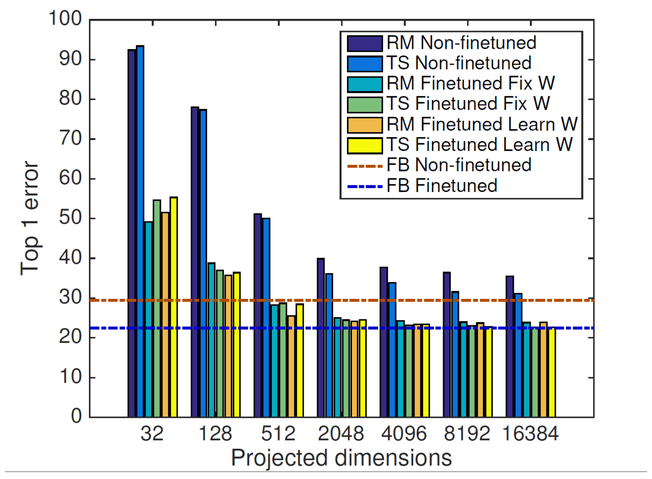

Figure 2. Classification error on the CUB dataset. Comparison of Random Maclaurin pooling and Tensor Sketch pooling for various combinations of projection dimensions and fine-tuning options. The two horizontal lines show the bilinear performance.

Evaluations Across Multiple Datasets

Data-set	Net	FC [4, 30]	Fisher [7]	FB [22]	RM (Alg. 1)	TS (Alg. 2)
CUB [36]	VGG-M [4]	49.90/42.03	52.73/NA	29.41/**22.44**	36.42/23.96	31.53/23.06
CUB [36]	VGG-D [30]	42.56/33.88	35.80/NA	19.90/**16.00**	21.83/16.14	20.50/**16.00**
MIT [27]	VGG-M [4]	39.67/35.64	32.80/NA	**29.77**/32.95	31.83/32.03	30.71/31.30
MIT [27]	VGG-D [30]	35.49/32.24	24.43/NA	**22.45**/28.98*	26.11/26.57	23.83/27.27*
DTD [6]	VGG-M [4]	46.81/43.22	42.58/NA	**39.57**/40.50	43.03/41.36	**39.60**/40.71
DTD [6]	VGG-D [30]	39.89/40.11	34.47/NA	32.50/35.04	36.76/34.43	**32.29**/35.49

Table 3. Top 1 errors of Fully Connected, Fisher Vector, Fully Bilinear, Random Maclaurin and Tensor Sketching methods on CUB bird recognition, MIT indoor scene recognition and Describable Texture datasets. Numbers before and after slash are non fine tuned and fine tuned errors. For RM and TS we use d=8192 and not learning the random weight configurations. Some fine tuning diverged, marked by "*".

Better Discriminative Power in Few Shots Learning

Many datasets are expensive to collect. For example, the bird species classification dataset (CUB) requires expert knowledge to label. Thus few shots learning is especially important in such case. We simulate a case where only 1, 2, 3, 7 or 14 images are available at training time. Table 4 shows the mAP by FB and TS methods.

# images	1	2	3	7	14
Bilinear	12.7	21.5	27.4	41.7	53.9
TS	15.6	24.1	29.9	43.1	54.3

Table 4. Few shots learning comparison (in mAP) between Bilinear and Tensor Sketch pooling.

Author information:
1. Department of Computer Science, University of California Berkeley
2. Snapchat