Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding

Akira Fukui*[1,2] **Dong Huk Park*[1]** **Daylen Yang*[1]**
Anna Rohrbach*[1,3] **Trevor Darrell[1]** **Marcus Rohrbach[1]**
[1]UC Berkeley EECS, CA, United States
[2]Sony Corp., Tokyo, Japan
[3]Max Planck Institute for Informatics, Saarbrücken, Germany

*in EMNLP 2016

Motivation

- Multimodal pooling: efficiently and expressively fuse the visual and textual representations. (Image Caption, VQA, etc.)

$$\hat{a} = \operatorname*{argmax}_{a \in A} p(a|\mathbf{x}, \mathbf{q}; \theta)$$

- Conventional approach: vector concatenation or element-wise operations.
- Only capture first-order interactions or partial second-order interactions.
- Might not be expressive enough to fully capture the complex associations between the two different modalities.
- Second-order models are more powerful!

Bilinear Models

- Outer Product of two vectors:

$$\mathbf{u} \otimes \mathbf{v} = \mathbf{u}\mathbf{v}^\top = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} = \begin{bmatrix} u_1v_1 & u_1v_2 & u_1v_3 \\ u_2v_1 & u_2v_2 & u_2v_3 \\ u_3v_1 & u_3v_2 & u_3v_3 \\ u_4v_1 & u_4v_2 & u_4v_3 \end{bmatrix}.$$

- Encoding full second-order interactions.

- Bilinear Models: input $x \in \mathbb{R}^{n_1}, y \in \mathbb{R}^{n_2}$, output $z = W[xy^T] \in \mathbb{R}^{n_3}$

  where [] denotes linearizing the matrix as a vector, and $W \in \mathbb{R}^{n_1 * n_2 * n_3}$

- High dimensionality! 512*512*512=134M!

- Reduce dimension and approximation.

Compact Bilinear Pooling

- Illustration:

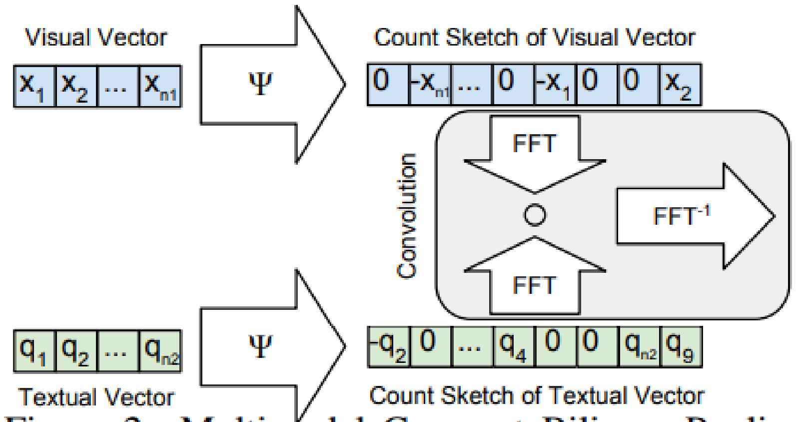

Figure 2: Multimodal Compact Bilinear Pooling (MCB)

- Count Sketch: randomly project $\mathbb{R}^n$ to $\mathbb{R}^d$
- Count sketch of the outer product of two vectors can be expressed as convolution of both count sketches:

$$\Psi(x \otimes q, h, s) = \Psi(x, h, s) * \Psi(q, h, s),$$

- Convolution in time domain equals element-wise product in frequency domain.

$$x' * q' \longrightarrow \mathrm{FFT}^{-1}(\mathrm{FFT}(x') \odot \mathrm{FFT}(q'))$$

Experiments

Method	Accuracy
Element-wise Sum	56.50
Concatenation	57.49
Concatenation + FC	58.40
Concatenation + FC + FC	57.10
Element-wise Product	58.57
Element-wise Product + FC	56.44
Element-wise Product + FC + FC	57.88
MCB ($2048 \times 2048 \to 16$K)	**59.83**
Full Bilinear ($128 \times 128 \to 16$K)	58.46
MCB ($128 \times 128 \to 4$K)	58.69
Element-wise Product with VGG-19	55.97
MCB ($d = 16$K) with VGG-19	**57.05**
Concatenation + FC with Attention	58.36
MCB ($d = 16$K) with Attention	**62.50**

Table 1: Comparison of multimodal pooling methods. Models are trained on the VQA train split and tested on test-dev.

Compact Bilinear d	Accuracy
1024	58.38
2048	58.80
4096	59.42
8192	59.69
16000	**59.83**
32000	59.71

Hadamard Product for Low-Rank Bilinear Pooling

Jin-Hwa Kim
Interdisciplinary Program in Cognitive Science
Seoul National University
Seoul 08826, Republic of Korea
jhkim@bi.snu.ac.kr

Kyoung-Woon On
School of Computer Science and Engineering
Seoul National University
Seoul 08826, Republic of Korea
kwon@bi.snu.ac.kr

Woosang Lim
School of Computing, KAIST
Daejeon 34141, Republic of Korea
quasar17@kaist.ac.kr

Jeonghee Kim & Jung-Woo Ha
NAVER LABS Corp. & NAVER Corp.
Gyeonggi-do 13561, Republic of Korea
{jeonghee.kim, jungwoo.ha}@navercorp.com

Byoung-Tak Zhang
School of Computer Science and Engineering & Interdisciplinary Program in Cognitive Science
Seoul National University & Surromind Robotics
Seoul 08826, Republic of Korea
btzhang@bi.snu.ac.kr

*in ICLR 2017

Low-Rank Bilinear Model

- Original: $f = W[xy^T] + b$

$$f_i = \sum_{j=1}^{N}\sum_{k=1}^{M} w_{ijk}x_j y_k + b_i = \mathbf{x}^T\mathbf{W}_i\mathbf{y} + b_i$$

- Assume W is low-rank (at most d), then:

$$\mathbf{W}_i = \mathbf{U}_i\mathbf{V}_i^T \qquad \mathbf{U}_i \in \mathbb{R}^{N\times d} \text{ and } \mathbf{V}_i \in \mathbb{R}^{M\times d}$$

- Re-write the bilinear model:

$$f_i = \mathbf{x}^T\mathbf{W}_i\mathbf{y} + b_i = \mathbf{x}^T\mathbf{U}_i\mathbf{V}_i^T\mathbf{y} + b_i = \mathbb{1}^T(\mathbf{U}_i^T\mathbf{x} \circ \mathbf{V}_i^T\mathbf{y}) + b_i$$

$$\boxed{\mathbf{f} = \mathbf{P}^T(\mathbf{U}^T\mathbf{x} \circ \mathbf{V}^T\mathbf{y}) + \mathbf{b}}$$

d is the dimension of joint embedding.

linear -> element-wise product -> linear

Experiments

Table 3: The VQA *test-standard* results for ensemble models to compare with state-of-the-art. For unpublished entries, their team names are used instead of their model names. Some of their figures are updated after the challenge.

	Open-Ended				MC
MODEL	**ALL**	**Y/N**	**NUM**	**ETC**	**ALL**
RAU (Noh & Han, 2016)	64.12	83.33	38.02	53.37	67.34
MRN (Kim et al., 2016b)	63.18	83.16	39.14	51.33	67.54
DLAIT (not published)	64.83	83.23	**40.80**	54.32	68.30
Naver Labs (not published)	64.79	83.31	38.70	54.79	69.26
MCB (Fukui et al., 2016)	66.47	83.24	39.47	**58.00**	70.10
MLB (ours)	**66.89**	**84.61**	39.07	57.79	**70.29**
Human (Antol et al., 2015)	83.30	95.77	83.39	72.67	91.54

Low-Rank Bilinear Pooling for Multi-Head

$$\mathbf{f} = \mathbf{P}^T(\mathbf{U}^T\mathbf{x} \circ \mathbf{V}^T\mathbf{y}) + \mathbf{b}$$

- Eight heads rather than two heads?
- Example: 3 heads, a, b, c
- Method 1: a -> a1, b->b1, c->c1, then a1∘b1∘c1, then linear projection.
    - Slow and not good performance.
- Method 2: d = [a, b, c], [] denotes concatenation, let x=d, y=d.
    - Fast and good performance: 28.35 (+0.71) on En-De test set.
- Also consider first-order interactions?
    - Concatenate with d before the linear projection.
    - 28.59 (+0.95) on En-De test set.