A Simple VQA Model with a Few Tricks and Image Features from Bottom-up Attention

Damien Teney¹, Peter Anderson²*, David Golub⁴*, Po-Sen Huang³, Lei Zhang³, Xiaodong He³, Anton van den Hengel¹

¹University of Adelaide    ²Australian National University
³Microsoft Research     ⁴Stanford University
*Work performed while interning at MSR
Proposed model

Straightforward architecture

- **Joint embedding** of question/image
- Single-head, question-guided attention over image
- Element-wise product

The devil is in the details

- Image features from Faster R-CNN
- Gated tanh activations
- Output as regression of answer scores, soft scores as target
- Output classifiers initialized with pretrained representations of answers
Non-linear layers: **gated hyperbolic tangent** activations

- Defined as: input $x$, output $y$
  
  $\tilde{y} = \tanh(Wx + b)$ \hspace{1cm} intermediate activation
  
  $g = \sigma(W'x + b')$ \hspace{1cm} gate
  
  $y = \tilde{y} \circ g$ \hspace{1cm} combine with element-wise product

- Inspired by gating in LSTMs/GRUs
- Empirically better than ReLU, tanh, gated ReLU, residual connections, etc.
- Special case of highway networks; used before in:
  
  
Question encoding

Chosen implementation
- Pretrained GloVe embeddings, d=300
- GRU encoder

Better than....
- Word embeddings learned from scratch
- GloVe of dimension 100, 200
- Bag-of-words (sum/average of embeddings)
- GRU backwards
- GRU bidirectional
- 2-layer GRU
Classical “top-down” attention on image features

Chosen implementation

- Simple attention on image feature maps
- One head
- Softmax normalization of weights

Better than....

- No L2 normalization
- Multiple heads
- Sigmoid on weights
Chosen implementation

- **Sigmoid** output (regression) of answer scores:
  - allows **multiple answers** per question
- **Soft targets** in [0,1]
  - allows **uncertain answers**
- **Initialize classifiers** with representations of answers

\[ y = \sigma(Wx) \quad W \text{ of dimensions } n\text{Answers} \times d \]

Better than....

- **Softmax classifier**
- **Binary targets** \{0,1\}
- **Classifiers learned from scratch**
Chosen implementation

- **Sigmoid** output (regression) of answer scores:
  - allows multiple answers per question
- **Soft targets** in [0,1]
  - allows uncertain answers
- **Initialize classifiers** with representations of answers

$$y = \sigma(W^{\text{text}}x^{\text{text}} + W^{\text{img}}x^{\text{img}})$$

Initialize $W^{\text{text}}$ with GloVe word embeddings

Initialize $W^{\text{img}}$ with Google Images (global ResNet features)
Training and implementation

- Additional training data from Visual Genome: questions with matching answers and matching images (about 30% of Visual Genome, i.e. ~485,000 questions)
- Keep all questions, even those with no answer in candidates, and with 0 < score < 1
- Shuffle training data but keep balanced pairs in same mini-batches
- Large mini-batches of 512 QAs; sweet spot in {64, 128, 256, 384, 512, 768, 1024}
- 30-Network ensemble: different random seeds, sum predicted scores
- Equally applicable to VQA and image captioning
- Significant relative improvements: 6 – 8 % (VQA / CIDEr / SPICE)
- Intuitive and interpretable (natural approach)
Typically, attention models operate on the spatial output of a CNN. We calculate attention at the level of objects and other salient image regions.
Can be implemented with Faster R-CNN\(^1\)

- Pre-train on 1600 objects and 400 attributes from Visual Genome\(^2\)
- Select salient regions based on object detection confidence scores
- Take the mean-pooled ResNet-101\(^3\) feature from each region

\(^1\)NIPS 2015, \(^2\)http://visualgenome.org, \(^3\)CVPR 2016
Qualitative differences in attention methods

Q: Is the person wearing a **helmet**?

Q: What **foot** is in front of the other **foot**?
VQA failure cases: counting, reading

Q: **How many** oranges are sitting on pedestals?

Q: **What is the name** of the realtor?
Equally applicable to Image Captioning

ResNet baseline: A man sitting on a **toilet** in a bathroom.

Up-Down attention: A man sitting on a **couch** in a bathroom.
**MS COCO Image Captioning Leaderboard**

<table>
<thead>
<tr>
<th>#</th>
<th>User</th>
<th>BLEU-1</th>
<th>BLEU-2</th>
<th>BLEU-3</th>
<th>BLEU-4</th>
<th>METEOR</th>
<th>ROUGE-L</th>
<th>CIDEr-D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>c5 ▲  c40 ▲</td>
<td>c5 ▲  c40 ▲</td>
<td>c5 ▲  c40 ▲</td>
<td>c5 ▲  c40 ▲</td>
<td>c5 ▲  c40 ▲</td>
<td>c5 ▲  c40 ▲</td>
<td>c5 ▲  c40 ▲</td>
<td>c5 ▲  c40 ▲</td>
</tr>
<tr>
<td>1</td>
<td>panderson_msr</td>
<td>0.802</td>
<td>0.952</td>
<td>0.641</td>
<td>0.888</td>
<td>0.491</td>
<td>0.794</td>
<td>0.369</td>
</tr>
<tr>
<td></td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
<td>(1)</td>
</tr>
<tr>
<td>2</td>
<td>TencentAI.v1</td>
<td>0.793</td>
<td>0.943</td>
<td>0.633</td>
<td>0.880</td>
<td>0.483</td>
<td>0.781</td>
<td>0.363</td>
</tr>
<tr>
<td></td>
<td>(2)</td>
<td>(2)</td>
<td>(2)</td>
<td>(2)</td>
<td>(2)</td>
<td>(4)</td>
<td>(2)</td>
<td>(4)</td>
</tr>
<tr>
<td>3</td>
<td>xxzhu</td>
<td>0.786</td>
<td>0.935</td>
<td>0.629</td>
<td>0.871</td>
<td>0.485</td>
<td>0.778</td>
<td>0.368</td>
</tr>
<tr>
<td></td>
<td>(5)</td>
<td>(4)</td>
<td>(3)</td>
<td>(3)</td>
<td>(2)</td>
<td>(2)</td>
<td>(2)</td>
<td>(2)</td>
</tr>
<tr>
<td>4</td>
<td>CASIA_IVA</td>
<td>0.786</td>
<td>0.934</td>
<td>0.629</td>
<td>0.870</td>
<td>0.484</td>
<td>0.776</td>
<td>0.368</td>
</tr>
<tr>
<td></td>
<td>(4)</td>
<td>(5)</td>
<td>(4)</td>
<td>(4)</td>
<td>(3)</td>
<td>(4)</td>
<td>(3)</td>
<td>(4)</td>
</tr>
<tr>
<td>5</td>
<td>Anonymous</td>
<td>0.787</td>
<td>0.937</td>
<td>0.627</td>
<td>0.867</td>
<td>0.476</td>
<td>0.765</td>
<td>0.356</td>
</tr>
<tr>
<td>6</td>
<td>etienmem</td>
<td>0.781</td>
<td>0.931</td>
<td>0.619</td>
<td>0.860</td>
<td>0.470</td>
<td>0.759</td>
<td>0.352</td>
</tr>
</tbody>
</table>

- **Bottom-up attention adds 6 – 8% improvement** on SPICE and CIDEr metrics
  
  (see arXiv: [Bottom-Up and Top-Down Attention for Image Captioning and VQA](http://arxiv.org/))

- First place on almost all MS COCO leaderboard metrics
VQA experiments

- **Current best results**  Ensemble, trained on tr+va+VG, eval. on test-std
  Yes/no: 86.52  Number: 48.48  Other: 60.95  **Overall: 70.19**

- **Bottom-up attention** adds 6% relative improvement (even though the baseline ResNet has twice as many layers)

  Single-network, trained on tr+VG, eval. on va

<table>
<thead>
<tr>
<th>Model</th>
<th>Yes/No</th>
<th>Number</th>
<th>Other</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours: ResNet (1×1)</td>
<td>76.0</td>
<td>36.5</td>
<td>46.8</td>
<td>56.3</td>
</tr>
<tr>
<td>Ours: ResNet (14×14)</td>
<td>76.6</td>
<td>36.2</td>
<td>49.5</td>
<td>57.9</td>
</tr>
<tr>
<td>Ours: ResNet (7×7)</td>
<td>77.6</td>
<td>37.7</td>
<td>51.5</td>
<td>59.4</td>
</tr>
<tr>
<td>Ours: Up-Down</td>
<td>80.3</td>
<td>42.8</td>
<td>55.8</td>
<td>63.2</td>
</tr>
<tr>
<td>Relative Improvement</td>
<td>3%</td>
<td>14%</td>
<td>8%</td>
<td>6%</td>
</tr>
</tbody>
</table>
Take-aways and conclusions

- Difficult to predict effects of architecture, hyperparameters, ...
  
  Engineering effort: good intuitions are valuable, then need **fast experiments**
  
  Performance \( \approx (\# \text{ Ideas}) \times (\# \text{ GPUs}) / (\text{Training time}) \)

- Beware of experiments with reduced training data

- Non-cumulative gains, performance saturates
  
  Fancy tweaks may just add more capacity to network
  
  May be redundant with other improvements

- Calculating attention at the level of **objects and other salient image regions** (bottom-up attention) significantly improves performance
  
  Replace pretrained CNN features with pretrained bottom-up attention features
Questions?


arXiv:1707.07998:  **Bottom-Up and Top-Down Attention for Image Captioning and VQA**

Damien Teney, Peter Anderson, David Golub, Po-Sen Huang, Lei Zhang, Xiaodong He, Anton van den Hengel