Visual Question Answering with Deep Learning

Aahitagni Mukherjee
Shubham Agarwal
Visual Question Answering

Given an image, and a natural language-like question, find the correct answer to it

- Training on a set of triplets (image, question, answer).
- *Free-form* and *open-ended* questions.
- Answers can be single word or multiple word.

*Lin Ma et al 2015*
Datasets

- DAQUAR (DATaset for QUestion Answering on Real-world images) – 1450 images and 12468 questions related to them. On an average 12 words per question.
  
- VQA (Visual Question Answering) dataset – 254,721 images, 764,163 questions, 9,934,119 answers

- Wu-Palmer Similarity Measure (WUPS score) is used for performance evaluation – Script by Malinowski M.

Malinowski et al. 2014
Anton et al. 2015
Wu et al. 1994
Challenges

- The output is to be conditioned on both image and language inputs.
- A better representation of the image content is essential.

- Interactions between the two modalities need to appropriately modelled.

Lin Ma et al 2015
Previous Approaches

- **Neural-based approach** - image representation from a CNN is fed to each hidden layer of a single LSTM. The LSTM then models the concatenation of question and answer.

- **mQA approach** – 4 units - an LSTM to extract the *question representation*, a CNN to extract the *visual representation*, an LSTM for storing the linguistic context in an answer, and a *fusing component* to combine the information from the first three components and generate the answer.

- **VIS + LSTM** - Here the image is treated as a single word, and the intermediate representation of the input thus obtained is used for classification into the correct class, which is the single word answer.

- **CNN approach** - uses 3 CNN’s - one to extract sentence representation, one for image representation, and the third is a multimodal layer to fuse the two.

Malinowski et al. 2015
Gao et al. 2015
Kiros et al. 2015
Lin Ma et al. 2015
CNN model

Lin Ma et al 2015
Image CNN

\[ \nu_{im} = \sigma(w_{im}(CNN_{im}(I)) + b_{im}) \]

\( \sigma \): Nonlinear activation function.
\( w_{im} |_{d \times 4096} \): Mapping matrix
\( CNN_{im} \) takes image as input and outputs a fixed length vector.
Sentence CNN

1. For sequential input $\sigma$, convolution unit for feature map of type $f$ on the $l^{th}$ layer is

$$\nu^i_{(l,f)} \overset{\text{def}}{=} \sigma(w_{(l,f)} \tilde{\nu}^i_{(l-1)} + b_{(l,f)})$$

2. \(\tilde{\nu}^i_{(l-1)} \overset{\text{def}}{=} \nu^i_{(l-1)} \parallel \nu^i_{(l-1)} \parallel \nu^i_{l-1}\)

3. \(\tilde{\nu}^i_{(0)} \overset{\text{def}}{=} \nu^i_{wd} \parallel \nu^i_{wd} \parallel \nu^i_{wd}\)

4. Max-pooling after each convolution

$$\nu^i_{(l+1,f)} = \max(\nu^i_{(l,f)} , \nu^{2i+1}_{(l,f)})$$

$\nu^i_{wd}$: Skip-gram word embedding of $i$-th question word

Lin Ma et al 2015
Multimodal Convolutional Layer

input: $\nu_{qt} = [\nu_{(6)}^0 \ldots \nu_{(6)}^n]$

Capturing the interaction between two multimodal inputs

$v_6^i = v_6^i \parallel v_{im} \parallel v_6^{i+1}$

$v_{(mm,f)}^i = \sigma(\mathbf{w}_{(mm,f)} v_6^i + b_{(mm,f)})$
Proposed Modification
Input to LSTM

Skip-gram word embeddings from the question sentence

Mikolov et al. 2013
https://code.google.com/p/word2vec/
LSTM

\[
\begin{align*}
    i_t &= \sigma(W_{vi}v_t + W_{hi}h_{t-1} + b_i) \\
    f_t &= \sigma(W_{vf}v_t + W_{hf}h_{t-1} + b_f) \\
    o_t &= \sigma(W_{vo}v_t + W_{ho}h_{t-1} + b_o) \\
    g_t &= \phi(W_{vg}v_t + W_{hg}h_{t-1} + b_g) \\
    c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
    h_t &= o_t \odot \phi(c_t)
\end{align*}
\]

Malinowski et al. 2015
https://github.com/junhyukoh/caffe-lstm
References

CONVOLUTIONAL, LONG SHORT-TERM MEMORY, FULLY CONNECTED DEEP NEURAL NETWORKS

Tara N. Sainath, Oriol Vinyals, Andrew Senior, Hasim Sak

Google, Inc., New York, NY, USA
{tsainath, vinyals, andrewsenior, hasim}@google.com

ABSTRACT
Both Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) have shown improvements over Deep Neural Networks (DNNs) across a wide variety of speech recognition tasks. CNNs, LSTMs and DNNs are complementary in their modeling capabilities, as CNNs are good at reducing frequency variations, LSTMs are good at temporal modeling, and DNNs are appropriate for mapping features to a more separable space. In this paper, we take advantage of the complementarity of CNNs, LSTMs and DNNs by combining them into one unified architecture. We explore the proposed architecture, which we call CLDNN, on a variety of large vocabulary tasks, varying from 200 to 2,000 hours. We find that the CLDNN provides a 4-6% relative improvement in WER over an LSTM, the strongest of the three individual models.

<table>
<thead>
<tr>
<th>Method</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>18.0</td>
</tr>
<tr>
<td>CNN+LSTM</td>
<td>17.6</td>
</tr>
<tr>
<td>LSTM+DNN</td>
<td>17.6</td>
</tr>
<tr>
<td>CLDNN</td>
<td>17.3</td>
</tr>
</tbody>
</table>

Table 5. WER, CLDNN