Deep Learning for Vision and Language

- Images
  - Fixed sizes (or resizable to fixed sizes)
  - Representation: convolutional neural networks
    - Feed-forward networks
- Sentences
  - Variable lengths
  - Representation: LSTM or GRU
    - Weight-sharing networks
- Videos
  - Variable lengths
  - Regarded as a sequence of frames or a set of frames
  - Representation: not clear yet
- Famous problems
  - Image caption generation
  - Visual question answering

Image Caption Generation

- Goal
  - Automatically describing the content of an input image using text
  - Generating a sentence relevant to an input image

- Problem formulation
  - Finding the sentence (a sequence of words) with maximum likelihood
  \[ S^* = \arg\max_S p(S|I) \quad \text{where} \quad S = \{S_1, S_2, ..., S_N\} \]

Show and Tell

- A simple encoder-decoder model
  - Motivated by a machine translation technique where encoder RNN reads the source sentence and decoder RNN generates the target sentence.
  - The encoder RNN is replaced by CNN.

[A group of people shopping at an outdoor market.]

There are many vegetables at the fruit stand.

**Image Encoder**

- Convolutional Neural Networks (CNNs)
  - AlexNet / VGG / GoogLeNet / ResNet

  - Representing images with fixed dimensional feature vectors

**Sentence Decoder**

- Long Short Term Memory (LSTM)

\[
\begin{align*}
  i_t &= \sigma(W_{ix}x_t + W_{ih}h_{t-1}) \\
  f_t &= \sigma(W_{fx}x_t + W_{fh}h_{t-1}) \\
  o_t &= \sigma(W_{ox}x_t + W_{oh}h_{t-1}) \\
  c_t &= f_t \odot c_{t-1} + i_t \odot g(W_{cx}x_t + W_{ch}h_{t-1}) \\
  h_t &= o_t \odot c_t \\
  p_t &= \text{softmax}(h_t)
\end{align*}
\]

**Inference**

\[
x_{t-1} = \text{CNN}(I) \\
x_t = W_xS_t \\
p_{t+1} = \text{LSTM}(x_t), \\
t \in \{1, ..., N - 1\}
\]

- Maximizing \( \log p_t(S_t) \) in each step.

**Training**

- Objective function
  - Sum of negative log-likelihood of the correct word at each step:
    \[
    L(S, I) = - \sum_t \log p_t(S_t)
    \]

- Training strategy: minimizing the loss w.r.t.
  - All the parameters of the LSTM
  - The top layer of the image encoder CNN
  - Word embedding matrix \( W_e \)

Generated caption: A group of people shopping at an outdoor market.
Results

Show, Attend and Tell

- Image caption generation with visual attention
  - Encoding is similar.
  - LSTM receives visual attention as an input for the prediction of each word.

Sentence Decoder

- Terminology
  - $y = \{y_1, ..., y_C\}$, where $y_i \in \mathbb{R}^K$: caption with $C$ words
  - $a = \{a_1, ..., a_L\}$, where $a_i \in \mathbb{R}^D$: image representations of $L$ locations
- Attention estimation
  - $f_{\text{att}}(\cdot)$: attention model based on multilayer perceptron
  - $\alpha = \{\alpha_1, ..., \alpha_L\}$: attention probability
    $$e_{ij} = f_{\text{att}}(a_j, h_{t-1}) \quad \alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^K \exp(e_{ik})}$$
- Decoding by LSTM
  - Context vector: representation considering visual attention
    $$\tilde{z}_t = \phi([a_i], \{\alpha_{ti}\})$$
  - Decoding
    $$(h_t, c_t) = \text{LSTM}(\{y_{t-1}, \tilde{z}_t\}, h_{t-1}, c_{t-1})$$

Attention model and LSTM are trained jointly end-to-end.

Attention

- Concept
  - Identifying where to observe for word generation
- Stochastic hard attention
  - One-hot vector: 1 for the attended region
  - Extracting feature from the position corresponding to one-hot value
  - Optimizing a variational lower bound on marginal log-likelihood $\log p(y|\alpha)$
- Deterministic soft attention
  - Real-valued vector whose elements represent amount of attention
  - Extracting feature from the weighted sum of multiple locations
  - Learning by using standard backpropagation since the whole model is smooth and differentiable
Results

Evaluation Metrics

- **BLEU**
  - Widely used to measure precision but inappropriate to estimate recall
  - The number of common words between generated caption and reference caption divided by the number of words in generated caption
  - Clipping the occurrences of the same words using a threshold
    - To avoid overestimating performance by counting trivial words
    - ex.) generated (the the the the) vs. reference (the cat is on the mat)
- **METEOR**
  - Harmonic mean of precision and recall with higher recall weight
- **CIDEr**
  - Sentence is transformed into Term Frequency Inverse Document Frequency (TF-IDF) to give lower weight for commonly occurring word

\[
CIDEr(g, r_1, \ldots, r_n) = \frac{1}{n} \sum_{i=1}^{n} \frac{tfidf(g) \cdot tfidf(r_i)}{|tfidf(g)||tfidf(r_i)|}
\]

Impact of Attention

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>BLEU-1</th>
<th>BLEU-2</th>
<th>BLEU-3</th>
<th>BLEU-4</th>
<th>METEOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flickr8k</td>
<td>Google NIC (Vinyals et al., 2014)†</td>
<td>65.0</td>
<td>44.2</td>
<td>27.7</td>
<td>17.7</td>
<td>17.31</td>
</tr>
<tr>
<td></td>
<td>Log Bilinear (Kiros et al., 2014a)§</td>
<td>65.6</td>
<td>42.4</td>
<td>27.7</td>
<td>17.7</td>
<td>17.31</td>
</tr>
<tr>
<td></td>
<td>Soft-Attention</td>
<td>67.0</td>
<td>44.8</td>
<td>29.9</td>
<td>19.5</td>
<td>18.93</td>
</tr>
<tr>
<td></td>
<td>Hard-Attention</td>
<td>67.0</td>
<td>45.7</td>
<td>31.4</td>
<td>21.3</td>
<td>20.30</td>
</tr>
<tr>
<td>Flickr30k</td>
<td>Google NIC</td>
<td>66.3</td>
<td>42.3</td>
<td>27.7</td>
<td>18.3</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Log Bilinear</td>
<td>60.0</td>
<td>38.0</td>
<td>25.4</td>
<td>17.1</td>
<td>16.88</td>
</tr>
<tr>
<td></td>
<td>Soft-Attention</td>
<td>66.7</td>
<td>43.4</td>
<td>28.8</td>
<td>19.1</td>
<td>18.49</td>
</tr>
<tr>
<td></td>
<td>Hard-Attention</td>
<td>66.9</td>
<td>43.9</td>
<td>29.6</td>
<td>19.9</td>
<td>18.46</td>
</tr>
<tr>
<td>COCO</td>
<td>CMU/MS Research (Chen &amp; Zitnick, 2014)*</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>20.41</td>
</tr>
<tr>
<td></td>
<td>MS Research (Fang et al., 2014)**</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>20.71</td>
</tr>
<tr>
<td></td>
<td>BRNN (Karpathy &amp; Li, 2014)*</td>
<td>64.2</td>
<td>45.1</td>
<td>30.4</td>
<td>20.3</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Google NIC†</td>
<td>66.6</td>
<td>46.1</td>
<td>32.9</td>
<td>24.6</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>Log Bilinear*</td>
<td>70.8</td>
<td>48.9</td>
<td>34.4</td>
<td>24.3</td>
<td>20.03</td>
</tr>
<tr>
<td></td>
<td>Soft-Attention</td>
<td>70.7</td>
<td>49.2</td>
<td>34.4</td>
<td>24.3</td>
<td>23.90</td>
</tr>
<tr>
<td></td>
<td>Hard-Attention</td>
<td>71.8</td>
<td>50.4</td>
<td>35.7</td>
<td>25.0</td>
<td>23.04</td>
</tr>
</tbody>
</table>

Visual Question Answering

**Characteristics**

- Providing an image and a question as inputs
- Involving significant amount of learning from data

**Motivation**

- General purpose image understanding: from low to high level
- Unified framework to solve the tasks defined on demand by questions
Visual Question Answering

- Technique to answer questions about an image
  - Input: image and question (sequence of words)
  - Output: answer (a word or sequence of words)

VQA system

```
Image
“What is the baby doing?”
Question

VQA system

Answer
Eating

Often formulated with deep neural networks
```

Sub-Problems in VQA

- Multi-domain classification
  - Classifiers are defined by questions.

Object classification: Person
Action classification: Jumping

- Different answers for the same semantic understanding of images

```
What is this? Elephant
Is this Elephant? Yes
```

Sub-Problems in VQA

- Reference problems

```
What color is the teapot? What is the woman holding? What is the man throwing?
```

- Understanding spatial relations

```
What is behind the horse? What is in front of the bed? What is beside the cat?
```

Sub-Problems in VQA

- Zero-shot learning
  - Object classifier: Elephant Lion Dog Cat Hippo
  - Yes/No classifier: Elephant Lion Dog Cat Hippo

```
Are these penguins? Is this a kitty?
```

Sub-Problems in VQA

- Weakly supervised learning: counting

```
How many people? How many dishes? How many snowboarders?
```
Problem Formulation

- Technique to answer questions about an image
  - Input: image and question (sequence of words)
  - Output: answer (a word or sequence of words)

VQA system

Image Encoder

- Convolutional Neural Networks (CNNs)
  - AlexNet / VGG / GoogLeNet / ResNet

Sentence Encoder

- Recurrent Neural Networks (RNNs)
  - LSTM / GRU

Answer Generation

- Flat classification
  - All classes are handled equivalently.
  - However, some classes may not be comparable, exclusive, and compatible.
**IMG+BOW**

- A simple baseline algorithm
  - Combination of CNN and BoW representations

**Multiple CNNs**

- 3 CNN-based approach
  - One to extract sentence representation
  - One for image representation
  - The third is a multimodal layer to fuse the two.

**VIS+LSTM**

- LSTM with CNN-based image representation
  - Image representation: treating the image as if it is the first word of the sentence
  - Word embedding: Treating the image as one word of the question

**Neural-Image-QA**

- LSTM with repeated CNN inputs
  - The question together with the visual representation is fed into LSTM.
  - CNN and LSTM are trained jointly and end-to-end.
  - Answers are obtained from the outputs of LSTM.

---


Stacked Attention Network

- Multi-step reasoning for image QA
  - Image model: using CNN to extract high level image representations
  - Question model: using CNN or LSTM to extract a semantic vector
  - Attention model: locating, via multi-step reasoning, the image regions that are relevant to the question for answer prediction

Image model:
- using CNN to extract high level image representations

Question model:
- using CNN or LSTM to extract a semantic vector

Attention model:
- locating, via multi-step reasoning, the image regions that are relevant to the question for answer prediction


NMN+LSTM

- Neural Module Network (NMN)
  - Assembling a network on the fly from a collection of modules
  - Modules: attention, reattention, combination, classification, measurement


DPPnet

- Dynamic parameter prediction network
  - Adapting network parameters depending on questions


Parameter Prediction Network


VQA: Dataset and Problem definition

- Statistics
  - Dataset size
    - Training: 248,349
    - Validation: 121,512
    - Testing: 244,302
  - Number of questions
    - One word: 89.32%
    - Two words: 6.91%
    - Three words: 2.74%
  - Length of answers
    - Evaluation is automated.


Evaluation is automated.
VQA: Dataset and Problem definition

• Examples

Q: How many dogs are seen?
Q: What color is the car?
Q: What animal is this?
Q: Is this vegetarian pizza?
Q: What is the mustache made of?

VQA Evaluation

• Evaluation

Two evaluation types

Q: What color is the water?

Answer candidates

(1) blue
(2) green
(3) blue
(4) blue
(5) green
(6) blue
(7) blue
(8) dark blue and aqua
(9) blue
(10) blue and green

Evaluation metric

\[
\text{Acc}(\text{ans}) = \min \left\{ \frac{\# \text{ of humans that said ans}}{3}, 1 \right\}
\]

VQA Summary

• Multi-modal deep learning is very popular in
  ▪ Semantic segmentation
  ▪ Image/video captioning
  ▪ Visual question answering
  ▪ Many others

• Recent advances of VQA is based on deep learning.
  ▪ Various combination of CNN and RNN
  ▪ Attention mechanism
  ▪ Adaptive networks

• Challenges in VQA
  ▪ Performance depends on too many factors.
  ▪ It is difficult to analyze existing limitations and proper directions.
  ▪ Solving isolated sub-problems doesn’t always improve the overall performance.