LSTM for Language Translation and Image Captioning

Tel Aviv University Deep Learning Seminar
Oran Gafni & Noa Yedidia
Part I – LSTM for Language Translation

- Motivation
- Background (RNNs, LSTMs)
- Model (General, Specific)
- Dataset, Training, Prediction
- Results
- (Very) Recent Developments

Mainly based on: “Sequence to Sequence Learning with Neural Networks” By Ilya Sutskever, Oriol Vinyals & Quoc V. Le – Google, 2014
Motivation

- **Regular DNNs** can only be applied to **fixed input-output** dimensionality vectors.
- An **LSTM** architecture (actually 2) is introduced to **solve general sequence to sequence problems**.
- The **LSTM is a natural choice** for **long-range** temporal dependencies in language translation.
- A useful property of the **LSTM** is **mapping a variable length** input sentence into a **fixed dimensional** vector representation.
- As translations tend to be **paraphrases**, this **encourages the LSTM** to find a sentence representation **capturing the meaning**.
Background – Standard RNNs

Given a sequence of inputs \((x_1, ..., x_T)\), a sequence of outputs \((y_1, ..., y_T)\) is computed by iterating over the following:

\[
h_t = \text{sigm}(W^{hx}x_t + W^{hh}h_{t-1})
\]

\[
y_t = W^{yh}h_t
\]
RNNs can be used in many variations for different purposes, e.g.:

- Vanilla Neural Networks
- e.g. Image Captioning
- e.g. Sentiment Classification
- e.g. Language Translation
- e.g. Frame Level Video Classification

Figure source: Stanford's CS231n, by Fei-Fei Li & Andrej Karpathy & Justin Johnson
Background – LSTMs

- LSTMs tweak cells through additive interactions, rather than fully transforming them (as in RNNs)

Figures source: Stanford’s CS231n, by Fei-Fei Li & Andrej Karpathy & Justin Johnson
The goal of the model is to estimate the conditional probability of:

\[ p(y_1, \ldots, y_{T'}, x_1, \ldots, x_T) \]

By iterating over:

For a standard LSTM-LM the conditional probabilities are denoted by:

\[ h_t = \text{sigm} \left( W^{hx} x_t + W^{hh} h_{t-1} \right) \]
\[ y_t = W^{yh} h_t \]

While each \( p(y_t|v, y_1, \ldots, y_{t-1}) \) distribution is represented by a Softmax over all the words in the vocabulary
Specific Model

- The model consists of **2 main LSTMs**
  - **Encoder** – A multilayered LSTM to **map the input sequence to a vector** of a fixed dimensionality ($v$)
  - **Decoder** – Another deep LSTM to **decode the target sequence from the vector**
- This has **3 main advantages**
  - **Negligible cost** (parameter-wise) while multiple language pairs are being trained simultaneously in a **natural way**
  - Deep LSTMs (4 layers) outperform shallow LSTMs
  - **Reversing the source sentence order greatly boosts** the LSTM **performance**
    - i.e. instead of mapping $a, b, c \rightarrow \alpha, \beta, \gamma$, we map $c, b, a \rightarrow \alpha, \beta, \gamma$
WMT 14 English-French

Trained on a subset of 12M sentences (348M French 304M English words)

chosen due to the publicly available tokenized training & test set of 1000-best list from the baseline SMT

A fixed vocabulary was used for both languages (160k words source, 80k words target, 'UNK' o.w.) with a vector representation for each word

\[
\begin{align*}
W(\text{"woman")} - W(\text{"man")} &\approx W(\text{"aunt")} - W(\text{"uncle")} \\
W(\text{"woman")} - W(\text{"man")} &\approx W(\text{"queen")} - W(\text{"king")}
\end{align*}
\]
The experiment **core was training** a large deep LSTM on many pair sentences.

This was done by **maximizing the log probability of a correct translation**, given the source sentence:

- $T$ - Correct translation given the source sentence
- $S$ - Training set

\[
\frac{1}{|S|} \sum_{(T,S) \in S} \log p(T|S)
\]

\[
\hat{T} = \arg \max_T p(T|S)
\]
The "Encoder" LSTM transforms the input into a rich fixed vector representation

This is later used as the initial condition of the "Decoder" LSTM

An "<EOS>" symbol is required, it enables the model to define a distribution over sequences of all possible lengths
A translation is produced by finding the most likely translation.

This is done by a simple left-to-right beam search decoder.

The decoder maintains a small number ($B$) of partial hypothesis.

At each time-step, $B$ is extended in the beam with every possible word from the vocabulary.

As soon as "<EOS>" is appended to the hypothesis, it's removed from the beam.
Results

BLEU (BiLingual Evaluation Understudy) score evaluated for translation quality – 34.81

<table>
<thead>
<tr>
<th>Method</th>
<th>test BLEU score (ntst14)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bahdanau et al. [2]</td>
<td>28.45</td>
</tr>
<tr>
<td>Baseline System [29]</td>
<td>33.30</td>
</tr>
<tr>
<td>Single forward LSTM, beam size 12</td>
<td>26.17</td>
</tr>
<tr>
<td>Single reversed LSTM, beam size 12</td>
<td>30.59</td>
</tr>
<tr>
<td>Ensemble of 5 reversed LSTMs, beam size 1</td>
<td>33.00</td>
</tr>
<tr>
<td>Ensemble of 2 reversed LSTMs, beam size 12</td>
<td>33.27</td>
</tr>
<tr>
<td>Ensemble of 5 reversed LSTMs, beam size 2</td>
<td>34.50</td>
</tr>
<tr>
<td>Ensemble of 5 reversed LSTMs, beam size 12</td>
<td><strong>34.81</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our model</td>
<td>Ulrich <strong>UNK</strong> , membre du conseil d’administration du constructeur automobile Audi , affirme qu’il s’agit d’une pratique courante depuis des années pour que les téléphones portables puissent être collectés avant les réunions du conseil d’administration afin qu’ils ne soient pas utilisés comme appareils d’écoute à distance.</td>
</tr>
<tr>
<td>Truth</td>
<td>Ulrich Hackenberg , membre du conseil d’administration du constructeur automobile Audi , déclare que la collecte des téléphones portables avant les réunions du conseil , afin qu’ils ne puissent pas être utilisés comme appareils d’écoute à distance , est une pratique courante depuis des années .</td>
</tr>
</tbody>
</table>
Google’s GNMT (Oct. 16’)

- BLEU 38.95
- 8 layer deep LSTM
- Residual connection between layers
- Bi-directional encoder
- Attention module

“Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation” – Oct. 16’
Part II – LSTM for Image Captioning

- What is Image Captioning?
- The NIC Model
- Background – Word Embedding
- Training
- Inference
- Evaluation Metrics
- Results
- References

Mainly based on: “Show and Tell: A Neural Image Caption Generator”
By Oriol Vinyals, Alexander Toshev, Samy Bengio & Dumitru Erhan - Google
What is Image Captioning?

A person skiing down a snow covered slope.

A group of giraffes standing next to each other.
Challenges

• **Detection:**
  person, kite, beach

• **Relations Actions and Adjectives:**
  person on beach fly kite red.

• **Natural Language:**
  A person on a beach flying a red kite.
Neural Image Caption (NIC) - Model

- **Encoder - deep CNN**
  - Transforms the image into a fixed length vector
    - Pre-trained CNN – Inception V2, used for classification

- **Decoder - LSTM**
  - Takes the output of the encoder as input and generates the target sentence
    - LSTM is often used in machine translation
    - The same approach here, where the input is an image instead of a sentence
A mapping function from words in some language to a high dimensional vectors:

\[ W : \text{word} \rightarrow \mathbb{R}^n \]

Typically, the function is a lookup table with a row for each word.

For example: 
\[ W(\text{“cat”}) = (0.2, -0.4, 0.7, ...) \]

- Initialized with random vectors
- Learns to have meaningful vectors according to the desired task
- Independent of the size of the dictionary
Interesting Results of Word Embedding

- Similar words are close together

- Analogies between words are encoded in the difference vectors between words

\[
W(\text{"woman"}) - W(\text{"man"}) \cong W(\text{"aunt"}) - W(\text{"uncle"})
\]

\[
W(\text{"woman"}) - W(\text{"man"}) \cong W(\text{"queen"}) - W(\text{"king"})
\]
This is an unrolled form of LSTM – **all LSTM units share the same parameters.**

- $\{S_1, S_2, \ldots\}$ – words of the describing sentence, represented as one hot vector
  
  \(S_0 – \text{start word}, \quad S_N – \text{stop word}\)

- $p_i$ probability distributions at each step

- $\log(p_i(S_i))$ - log-likelihoods of the correct word at each step

- Loss function: $L(I, S) = -\sum_{t=1}^{N} \log p_t(S_t)$

---

**Image Description:**

- LSTM units are connected in a sequence, with each unit inputting the output of the previous unit.
- The input to the first LSTM unit is an image, represented as $S_0$.
- Each subsequent LSTM unit takes the output of the previous unit as input, represented as $S_1, S_2, \ldots, S_N$.
- The final output of the LSTM sequence, represented as $P_N$, is connected back to the input, indicating a form of feedback or auto-regression.
Training

The training process:

• \( x_{-1} = CNN(I) \) - the initial input is the CNN output vector

• \( x_t = W_e S_t \) - from one hot vector to word embedding

• \( p_t = LSTM(x_t) \)

• Loss function \( L(I, S) \) is then minimized

- The image \( I \) is only input once
- Both the image and the words are mapped to the same space, the image by using a CNN with additional FC layer, the words by using word embedding \( W_e \)
There are 2 phases of the training:

1) **Training only FC layer and LSTM**
   - The CNN is kept fixed
   - A single FC layer is added on top it - transform image embedding into word embedding
   - The model is trained w.r.t. the parameters of the **word embedding**, the **FC layer** on top of the CNN and the **LSTM**.
2) **Fine-tune the whole system**

- **All parameters** are trained to jointly fine-tune the image encoder and the LSTM.
- Allows the system to transfer information from the image that is useful for generating descriptive captions, but not necessary useful for classifying objects.
- This phase must occur after first one - the noisiness of the randomly initialization causes irreversible corruption to the CNN.
Training Details

- Used stochastic gradient descent with fixed learning rate and no momentum.
- The CNN’s weights were initialized with pre-trained model (on imageNet).
- All other weights were randomly initialized.
- The embedding and the size of the LSTM memory was 512.
- Dropout
- Ensembling models
Inference

**Sampling (Greedy Search)**
- Sample the first word according to $p_1$, then provide the corresponding embedding as input and sample $p_2$ and so on, until stop word is sampled.

**Beam Search**
- Captions are generated word-by-word.
- At each step $t$ we use the set of sentences already generated with length $t - 1$ to generate a new set of sentences with length $t$.
- We keep only the top $k$ candidates at each step.
- $k$ is called the beam size, optimal for NIC $k = 20$.
- Better approximates $S = \arg\max_{S'} p(S'|I)$.
**Evaluation Metrics**

- **Human Raters** - most reliable metric, but time consuming. Only used to verify correlation with automated metrics.

  All automatic metrics require reference description!

- **BLEU score** – most common (next slide)

- **Perplexity of the model** - the geometric mean of the inverse probability for each predicted word. Used to perform choices regarding model selection and hyper-parameter tuning.

- **METEOR** – more intelligent than BLEU (for example - synonyms)

- **Cider, and more…**

- **Evaluation metrics is still an open challenge**
The BLEU Score

- BLEU scores range from 0-100, the higher the score, the more the translation correlates to a human translation.
- Measures how many words overlap in a given translation, compared to a reference translation
- Favors sequential words
- Penalize short translation

$$\text{BLEU} = \min \left( 1, \frac{\text{output-length}}{\text{reference-length}} \right) \left( \prod_{i=1}^{4} \text{precision}_i \right)^{\frac{1}{4}}$$
### Results

<table>
<thead>
<tr>
<th>Dataset name</th>
<th>size (train: valid: test)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pascal VOC 2008</td>
<td>-: -: 1000</td>
</tr>
<tr>
<td>Flickr8k</td>
<td>6000: 1000: 1000</td>
</tr>
<tr>
<td>Flickr30k</td>
<td>28000: 1000: 1000</td>
</tr>
<tr>
<td>MSCOCO</td>
<td>82783: 40504: 40775</td>
</tr>
<tr>
<td>SBU</td>
<td>1M: -: -</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Approach</th>
<th>PASCAL (xfer)</th>
<th>Flickr 30k</th>
<th>Flickr 8k</th>
<th>SBU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Im2Text</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TreeTalk</td>
<td></td>
<td>55</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>BabyTalk</td>
<td></td>
<td>56</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td>Tri5Sem</td>
<td></td>
<td></td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>m-RNN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MNLM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SOTA</td>
<td>25</td>
<td>56</td>
<td>58</td>
<td>19</td>
</tr>
<tr>
<td>NIC</td>
<td>59</td>
<td>66</td>
<td>63</td>
<td>28</td>
</tr>
<tr>
<td>Human</td>
<td>69</td>
<td>68</td>
<td>70</td>
<td></td>
</tr>
</tbody>
</table>

**Table 1.** Scores on the MSCOCO development set.

**Table 2.** BLEU-1 scores. We only report previous work results when available. SOTA stands for the current state-of-the-art.
Results

A person riding a motorcycle on a dirt road.

Two dogs play in the grass.

A skateboarder does a trick on a ramp.

A dog is jumping to catch a frisbee.

A group of young people playing a game of frisbee.

Two hockey players are fighting over the puck.

A little girl in a pink hat is blowing bubbles.

A refrigerator filled with lots of food and drinks.

A herd of elephants walking across a dry grass field.

A close up of a cat laying on a couch.

A red motorcycle parked on the side of the road.

A yellow school bus parked in a parking lot.

Describes without errors | Describes with minor errors | Somewhat related to the image | Unrelated to the image
Questions?
References

- https://github.com/tensorflow/models/tree/master/im2txt#model-overview