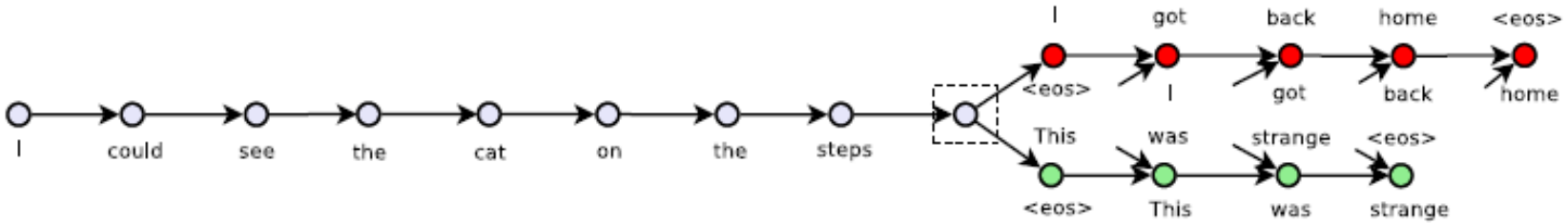
Skip-Thought Vectors

jianlincheng

Abstract

We describe an approach for unsupervised learning of a generic, distributed sentence encoder. Using the continuity of text from books, we train an encoder-decoder model that tries to reconstruct the surrounding sentences of an encoded passage. Sentences that share semantic and syntactic properties are thus mapped to similar vector representations. We next introduce a simple vocabulary expansion method to encode words that were not seen as part of training, allowing us to expand our vocabulary to a million words. After training our model, we extract and evaluate our vectors with linear models on 8 tasks: semantic relatedness, paraphrase detection, image-sentence ranking, question-type classification and 4 benchmark sentiment and subjectivity datasets. The end result is an off-the-shelf encoder that can produce highly generic sentence representations that are robust and perform well in practice. We will make our encoder publicly available.


I
could
see
the
cat
on
the
steps
I
got
back
home
<eos>
<eos>
I
got
back
home
This
was
strange
<eos>
<eos>
This
was
strange

Encoder

Encoder. Let $w_i^1, \ldots, w_i^N$ be the words in sentence s_i where N is the number of words in the sentence. At each time step, the encoder produces a hidden state $\mathbf{h}_i^t$ which can be interpreted as the representation of the sequence $w_i^1, \ldots, w_i^t$. The hidden state $\mathbf{h}_i^N$ thus represents the full sentence. To encode a sentence, we iterate the following sequence of equations (dropping the subscript i):

$$\mathbf{r}^t = \sigma(\mathbf{W}_r \mathbf{x}^t + \mathbf{U}_r \mathbf{h}^{t-1}) \quad (1)$$

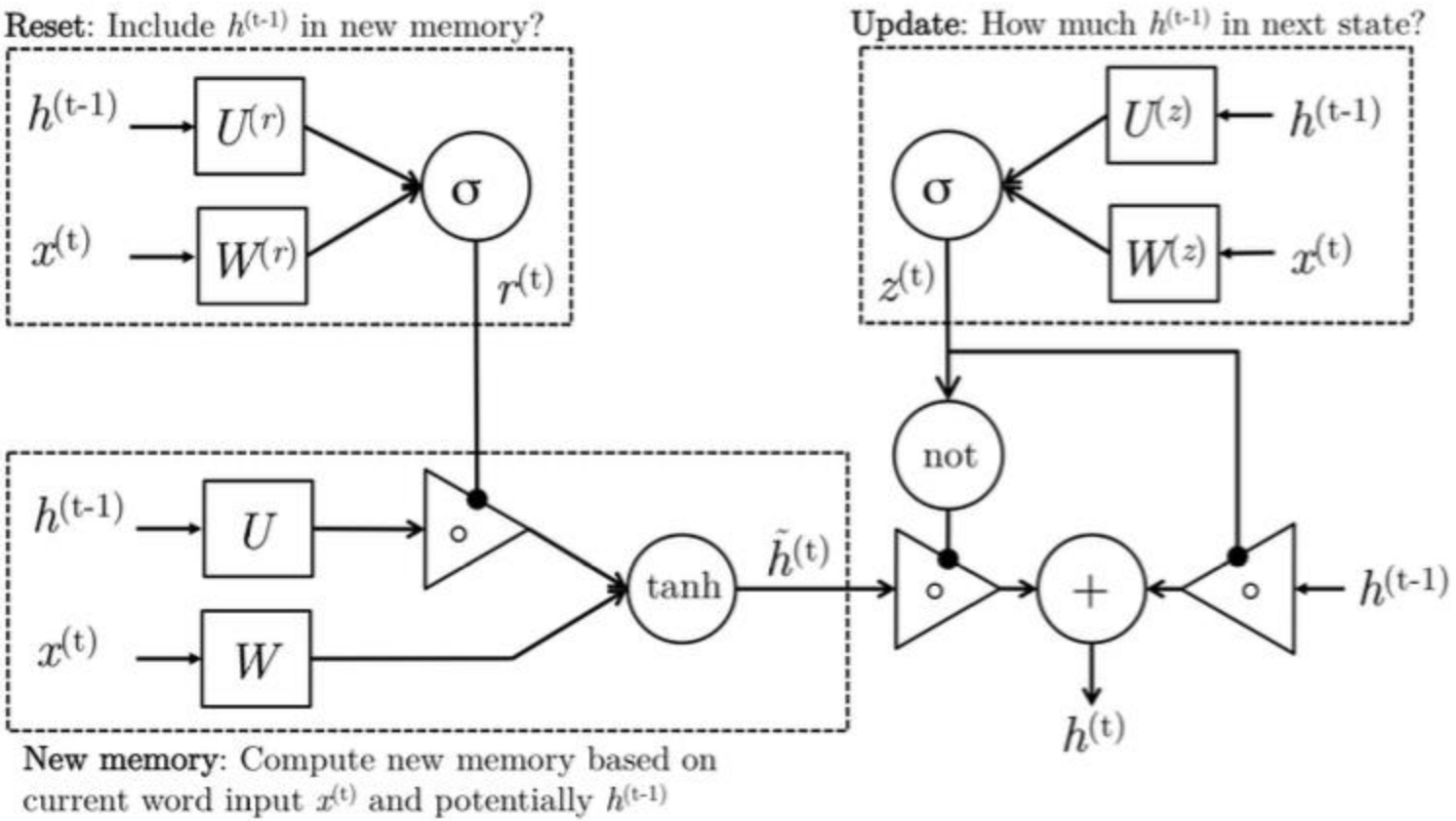
$$\mathbf{z}^t = \sigma(\mathbf{W}_z \mathbf{x}^t + \mathbf{U}_z \mathbf{h}^{t-1}) \quad (2)$$

$$\bar{\mathbf{h}}^t = \tanh(\mathbf{W}\mathbf{x}^t + \mathbf{U}(\mathbf{r}^t \odot \mathbf{h}^{t-1})) \quad (3)$$

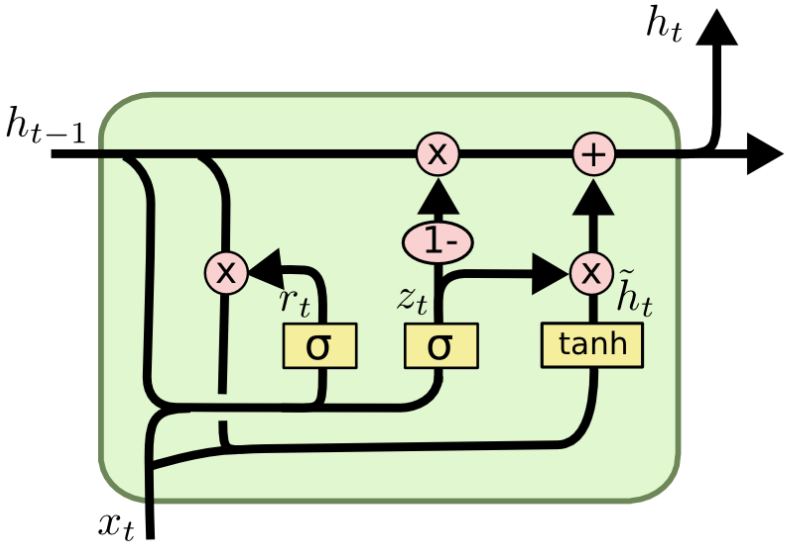
$$\mathbf{h}^t = (1 - \mathbf{z}^t) \odot \mathbf{h}^{t-1} + \mathbf{z}^t \odot \bar{\mathbf{h}}^t \quad (4)$$

where $\bar{\mathbf{h}}^t$ is the proposed state update at time t, $\mathbf{z}^t$ is the update gate, $\mathbf{r}_t$ is the reset gate ($\odot$) denotes a component-wise product. Both update gates takes values between zero and one.

Gated Recurrent Unit



Gated Recurrent Unit



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

DEcoder

Decoder. The decoder is a neural language model which conditions on the encoder output $\mathbf{h}_i$. The computation is similar to that of the encoder except we introduce matrices $\mathbf{C}_z$, $\mathbf{C}_r$ and $\mathbf{C}$ that are used to bias the update gate, reset gate and hidden state computation by the sentence vector. One decoder is used for the next sentence s_{i+1} while a second decoder is used for the previous sentence s_{i-1}. Separate parameters are used for each decoder with the exception of the vocabulary matrix $\mathbf{V}$, which is the weight matrix connecting the decoder's hidden state for computing a distribution over words. In what follows we describe the decoder for the next sentence s_{i+1} although an analogous computation is used for the previous sentence s_{i-1}. Let $\mathbf{h}_{i+1}^t$ denote the hidden state of the decoder at time t. Decoding involves iterating through the following sequence of equations (dropping the subscript $i+1$):

$$\mathbf{r}^t = \sigma(\mathbf{W}_r^d \mathbf{x}^{t-1} + \mathbf{U}_r^d \mathbf{h}^{t-1} + \mathbf{C}_r \mathbf{h}_i) \quad (5)$$

$$\mathbf{z}^t = \sigma(\mathbf{W}_z^d \mathbf{x}^{t-1} + \mathbf{U}_z^d \mathbf{h}^{t-1} + \mathbf{C}_z \mathbf{h}_i) \quad (6)$$

$$\bar{\mathbf{h}}^t = \tanh(\mathbf{W}^d \mathbf{x}^{t-1} + \mathbf{U}^d (\mathbf{r}^t \odot \mathbf{h}^{t-1}) + \mathbf{C}\mathbf{h}_i) \quad (7)$$

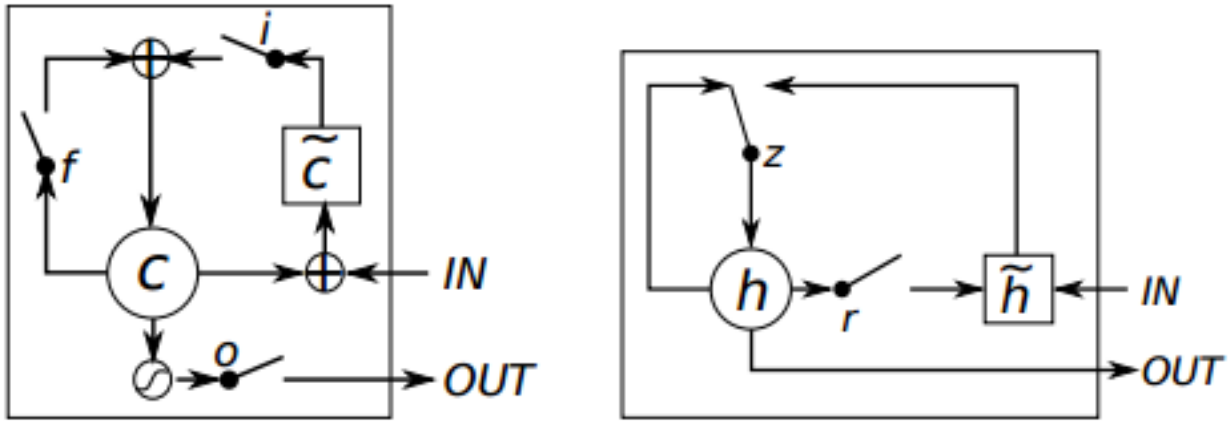
$$\mathbf{h}_{i+1}^t = (1 - \mathbf{z}^t) \odot \mathbf{h}^{t-1} + \mathbf{z}^t \odot \bar{\mathbf{h}}^t \quad (8)$$

Given $\mathbf{h}_{i+1}^t$, the probability of word w_{i+1}^t given the previous $t-1$ words and the encoder vector is

$$P(w_{i+1}^t | w_{i+1}^{<t}, \mathbf{h}_i) \propto \exp(\mathbf{v}_{w_{i+1}^t} \mathbf{h}_{i+1}^t) \quad (9)$$

where $\mathbf{v}_{w_{i+1}^t}$ denotes the row of $\mathbf{V}$ corresponding to the word of w_{i+1}^t. An analogous computation is performed for the previous sentence s_{i-1}.

LSTM and GRU



(a) Long Short-Term Memory (b) Gated Recurrent Unit

Figure 1: Illustration of (a) LSTM and (b) gated recurrent units. (a) i, f and o are the input, forget and output gates, respectively. c and $\tilde{c}$ denote the memory cell and the new memory cell content. (b) r and z are the reset and update gates, and h and $\tilde{h}$ are the activation and the candidate activation.

LSTM vs GRU

·A GRU has two gates, an LSTM has three gates

· GRU don't posses and internal memory(ct)that is different from the exposed hidden state.They don't have the output gate that is present in LSTMs

GRUs have fewer parameters (U and W are smaller) and thus may train a bit faster or need less data to generalize. On the other hand, if you have enough data, the greater expressive power of LSTMs may lead to better results.

The Future

Deep encoders and decoders

Larger context windows

encoding and decoding paragraphs

other encoders, such as convnets