5. Sequence modeling with recurrent neural networks

5.6. Machine translation with encoder-decoder structures

Manel Martínez-Ramón
Meenu Ajith
Aswathy Rajendra Kurup

Introduction

- A translation machine consists of a deep structure whose inputs are sequences with variable lengths in a given *input language* (e.g. English):

  Spring begins today.

  and whose outputs are sentences of a different length in a *target language* (e.g. French):

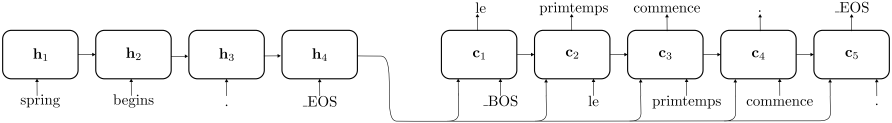
  Le printemps commence aujourd'hui.

- Feedforward deep structures are limited in this task because both their inputs and outputs are of fixed length.
- Recurrent structures have been used for this task.

Introduction

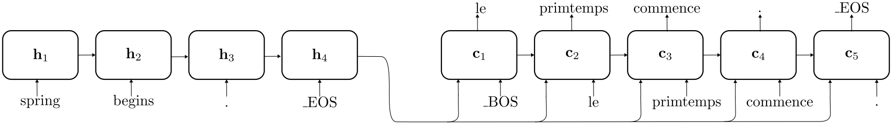
- One of the first successful variable length machine translation schemes is given by Ilya Sutskever et al. (2014) while working at Google Inc.
- The machine was constructed by a pair of input (encoder) LSTM and output (decoder) LSTM.
- The input uses a word embedding technique to code the words in vectors.
- The word embedding matrix consists of a matrix of dimensions $D_d \times D_c$ corresponding to the number of words and the dimension of the embedding vector.
- This is, each word in a dictionary for each language is encoded in a fixed length vector.

Structure



- The first word *spring* is entered, which generates state $\mathbf{h}_1$, which is fed back with the second word, to generate the next state. The operation is repeated to the _EOS token.
- State $\mathbf{h}_4$ is a fixed length code of the sentence.
- This state is used in another RNN.

Structure



- The output state $\mathbf{h}_4$ is used as input of the next RNN, concatenated with a _BOS token.
- This produces state $\mathbf{c}_1$. This is used to decode the first word *le*.
- State $\mathbf{c}_1$ is fed back together with $\mathbf{h}_4$ and the first decoded word to produce state $\mathbf{c}_2$, which is used to produce the second word.
- When the state is decoded as _EOS, the translation is finished.

Structure

- Usually, for the encoding and decoding sections, LSTM or GRU units are used.
- The words in the target language (French here) are obtained after the transformation of the hidden states $\mathbf{c}_t$ of the decoder RNN through a dense neural network.
- The number of outputs of the NN corresponds to the number of words in the target dictionary.
- The dense NN output estimates the probability of each word in the dictionary.
- Criterion: Minimize the correct translation negative log-likelihood (NLL)

$$J_{ML} = -\frac{1}{|S|}\sum_{T,S} \log p\left(T|S\right) \tag{1}$$

where $|\cdot|$ denotes the cardinality of s.

Greedy search versus beam search

Assume a dictionary with only four words: A, B, C, and _EOS.

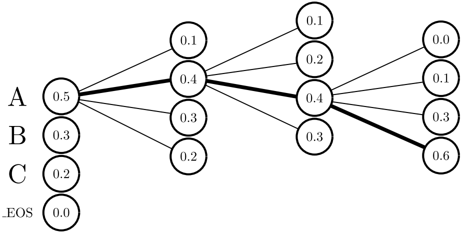
- An encoder-decoder system outputs 4 words in 4 consecutive steps. In the first step, the most probable word is A.
- When this word is fed back, the most probable word is B. The next most probable words are C and _EOS.
- The sequence is illustrated in the table:

	1	2	3	4
A	**0.5**	0.1	0.1	0.0
B	0.3	**0.4**	0.2	0.1
C	0.2	0.3	**0.4**	0.3
_EOS	0.0	0.2	0.3	**0.6**

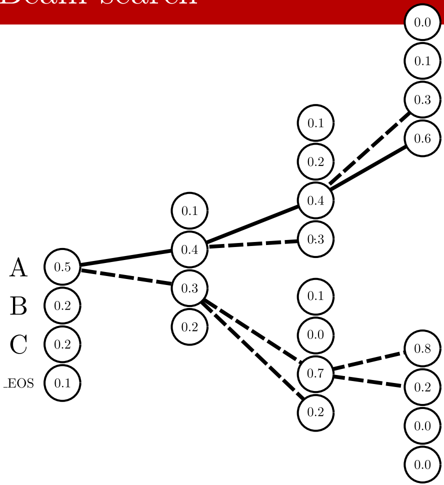
Greedy search versus beam search

- A greedy search would output the sequence "A", "B", "C", _EOS

  This gives a probability $p(T|S) = 0.5 \cdot 0.4 \cdot 0.4 \cdot 0.6 = 0.048$.



- A beam search uses the k words with the highest probability as input for the next steps.
- This changes the probabilities in the next steps



- With a beam $k = 2$, the probabilities are

Path Probability	NLL
$p(\mathrm{ABCC}\vert\mathrm{S}) = 0.024$	0.4
$p(\mathrm{ABC\_EOS}\vert\mathrm{S}) = 0.048$	**0.43**
$p(\mathrm{AB\_EOS}\vert\mathrm{S}) = 0.060$	~~0.61~~
$p(\mathrm{ACCA}) = 0.084$	0.01
$p(\mathrm{ACCB}\vert\mathrm{S}) = 0{,}021$	0.41
$p(\mathrm{AC\_EOS}) = 0.030$	~~0.76~~

The bold one is the finished sequence with the lowest NLL. The unfinished ones have lower NLL, so the search on their paths must be continued. The rest of the paths are discarded.

Experiment

Sutskever et al describe the experimental setup as follows:

- Dataset: WMT 14 dataset, with 12 million sentences, 284 million French words, and 304 million English words.
- Dictionary: 160.000 English words (source) and 80.000 French words (target).
- Out-of-dictionary words were changed by a UNK (unknown) token.
- Translations produced by beam search.
- Source sentences reversed.

Experiment

- Deep LSTMs with 4 layers of 1000 cells.
- 1000 dimension word embeddings.
- Softmax output over 80.000 words (without specification of the structure of the dense layers.)
- LSTM parameters initialized uniformly between -0.08 and 0.08.
- Momentum gradient descent with $\mu = 0.7$. 7.5 epochs.
- Batches of 128 sequences.
- Hard constraint on the norm of the weights.

Results

- Evaluation metric: Bilingual Evaluation Understudy. It is a number between 0 and 100 that compares automatic translations to high-quality reference translations.

BLEU Score*	**Interpretation**
< 10	Almost useless
10 - 19	Hard to get the gist
20 - 29	Clear gist, significant grammatical errors
30 - 40	Understandable to good translations
40 - 50	High quality translations
50 - 60	Very high quality, adequate, fluent translations
> 60	Quality often better than human

$*$ See https://cloud.google.com/translate/automl/docs/evaluate.
Google's current BLEU for French as a target is higher tan 90.

Results

Method	Test BLEU Score
Bahdanau et al. [1]	28.45
Baseline System [2]	33.30
Single forward LSTM, beam size 12	26.17
Single reversed LSTM, beam size 12	30.59
Ensemble of 5 reversed LSTMs, beam size 1	33.00
Ensemble of 2 reversed LSTMs, beam size 12	33.27
Ensemble of 5 reversed LSTMs, beam size 2	34.50
Ensemble of 5 reversed LSTMs, beam size 12	34.81

1. D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate.arXiv preprint arXiv:1409.0473, 2014
2. H. Schwenk. Université Le Mans.