Introduction to Deep Learning

9. Contextual Representations

Haibin Lin and Leonard Lausen

gluon-nlp.mxnet.io

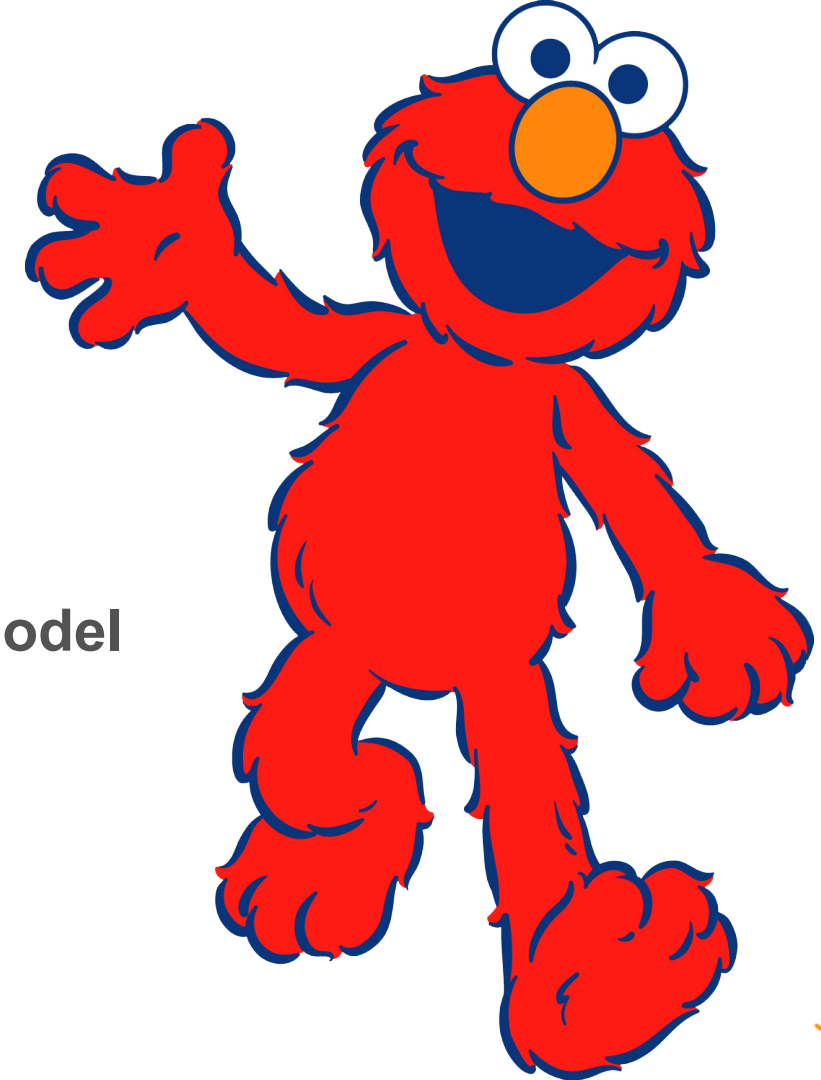
aws

8:30-9:00	Continental Breakfast
9:00-9:45	Introduction and Setup
9:45-10:30	Neural Networks 101
10:30-10:45	Break
10:45-11:15	Machine Learning Basics
11:15-11:45	Context-free Representations for Language
11:45-12:15	Convolutional Neural Networks
12:15-13:15	Lunch Break
13:15-14:00	Recurrent Neural Networks
14:00-14:45	Attention Mechanism and Transformer
14:45-15:00	Coffee Break
15:00-16:15	Contextual Representations for Language
16:15-17:00	Language Generation

Word2Vec/FastText/GloVE is great. Can we do better?

ELMo

Embedding from Language Model



Elmo Abby Cadabby Zoe Cookie Monster Oscar the Grouch - sesame street

gluon-nlp.mxnet.io

aws

ELMo

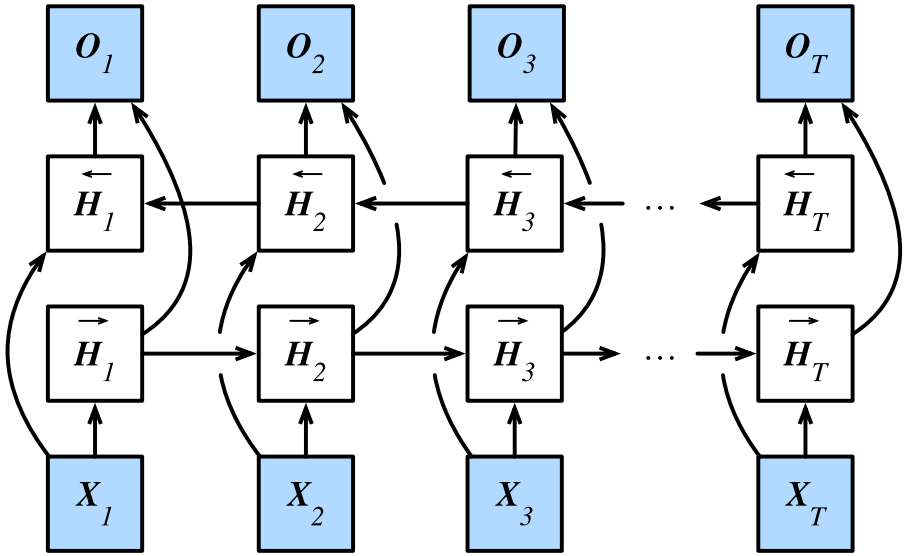
Model Architecture

- Character CNN embedding

ELMo

Model Architecture

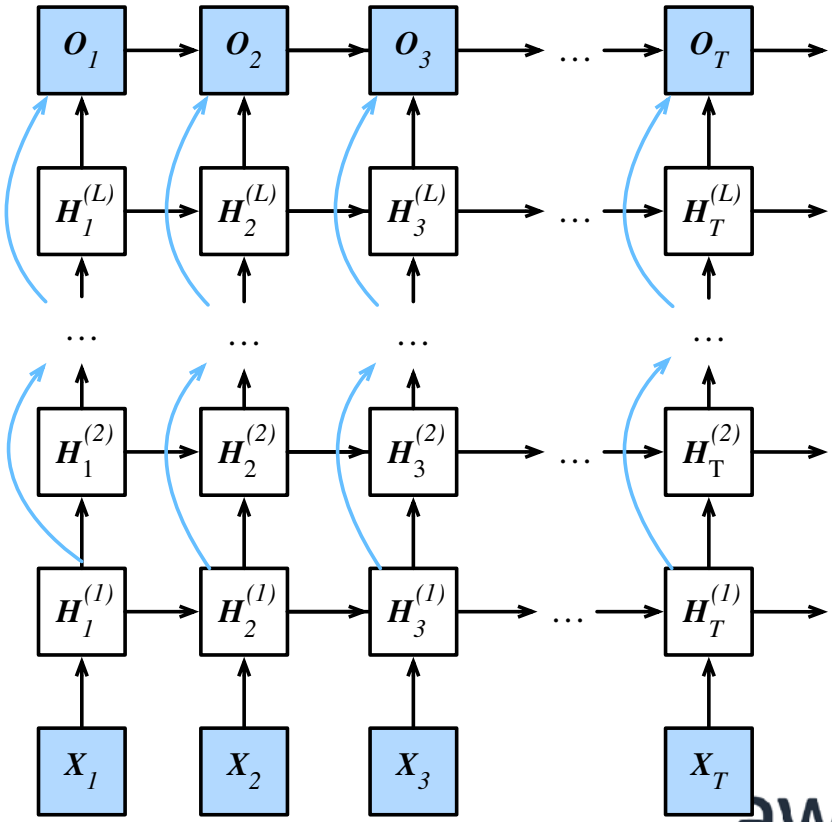
- Character CNN embedding
- Bidirectional



ELMo

Model Architecture

- Character CNN embedding
- Bidirectional
- Residual connections



ELMo

Training Procedure

- Pre-train bidirectional language model on large corpus (1B words)

$$p(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} p(t_k \mid t_{k+1}, t_{k+2}, \ldots, t_N).$$

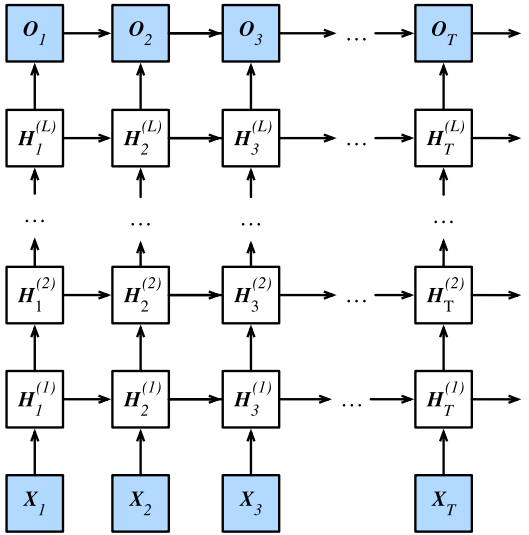
$$p(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} p(t_k \mid t_1, t_2, \ldots, t_{k-1}).$$

ELMo

Training Procedure

- Pre-train bidirectional language model on large corpus
- Extract features (weight sum of hidden outputs) for downstream tasks

$$\mathbf{E}_t = \sum_i w_i \mathbf{H}_{i,t}$$



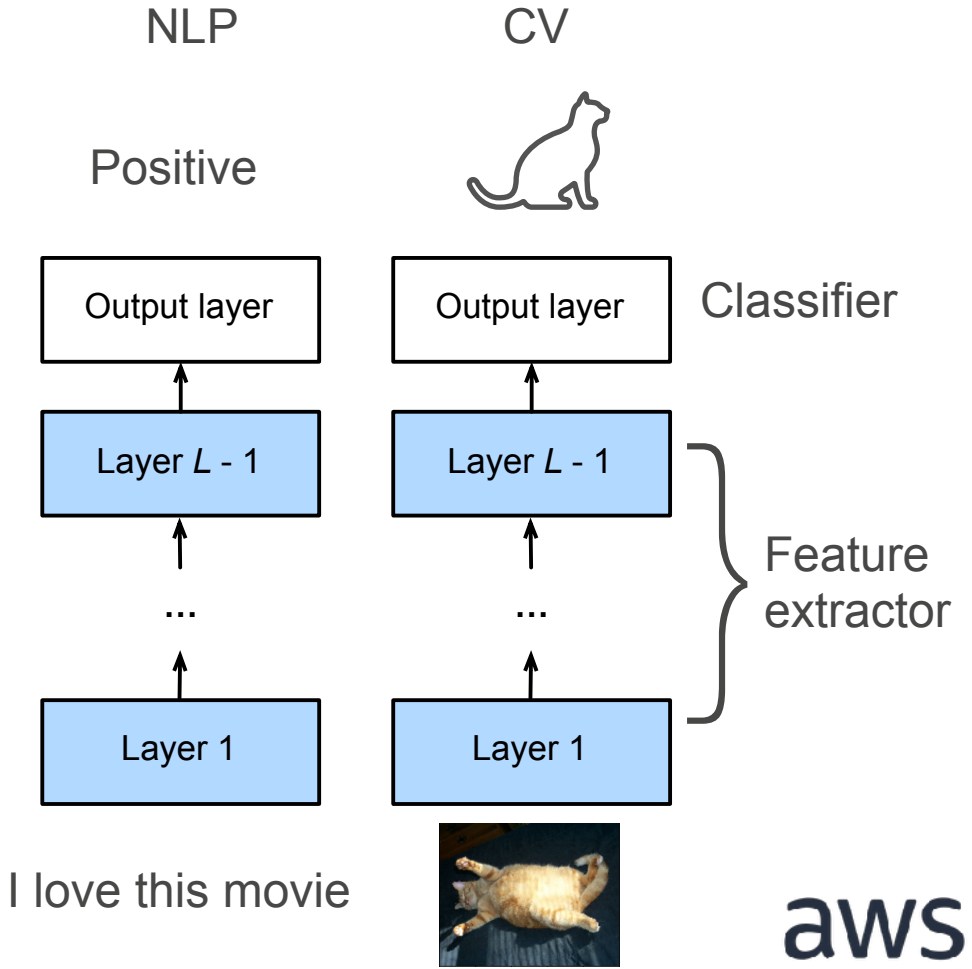
BERT

Bidirectional Embedding from Transformers

gluon-nlp.mxnet.io

Motivation of BERT

- A fine-tuning based approach
- The pre-trained model capture sufficient data information
- Only need to add a simple output layer for a new task

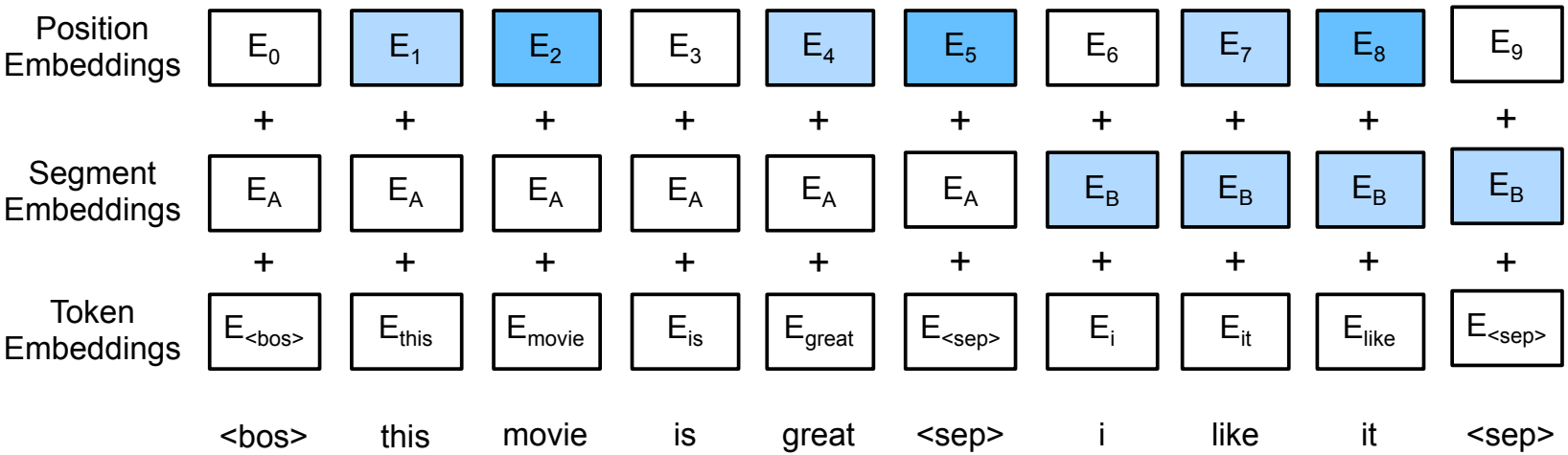


BERT Architecture

- A (big) Transformer encoder (without the decoder)
- Two variants:
  - Base: #blocks = 12, hidden size = 768, #heads = 12, #parameters = 110M
  - Large: #blocks = 24, hidden size = 1024, #heads = 16, #parameter = 340M
- Train on large-scale corpus (books and wikipedia) with > 3B words

Modification of inputs

- Each example is a pair of sentences
- Add an additional segment embedding



Pre-training Task 1: Masked Language Model

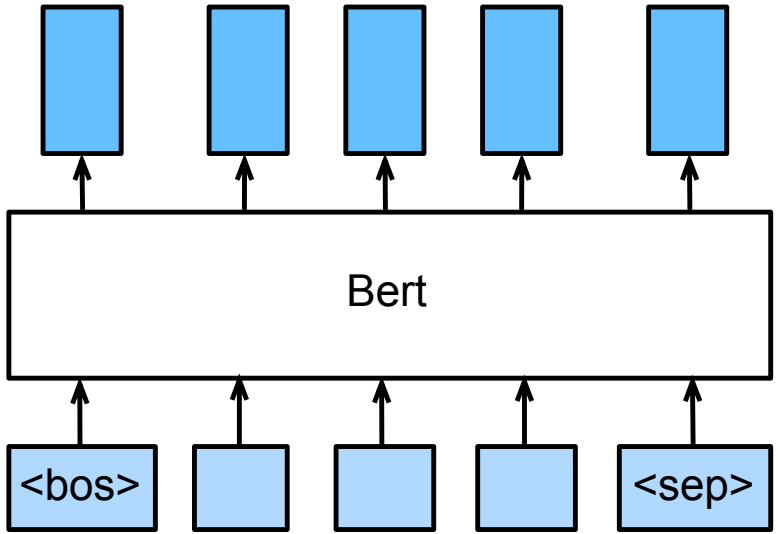
- Randomly mask (e.g. 15%) tokens in each sentence, predict these masked tokens
  - Transformer is bidirectional, which breaks the unidirectional limit of standard LM
- No mask token (<mask>) in fine turning tasks
  - 80% of the time, replace selected tokens with <mask>
  - 10% of the time, replace with randomly picked tokens
  - 10% of the time, keep the original tokens

Pre-training Task 2: Next Sentence Prediction

- 50% of time, choose a sequential sentence pair
  - <bos> this movie is great <sep> i like it <sep>
- 50% of time, choose a random sentence pair
  - <bos> this movie is great <sep> hello world <sep>
- Feed the Transformer output of <bos> into a dense layer to predict if it is a sequential pair

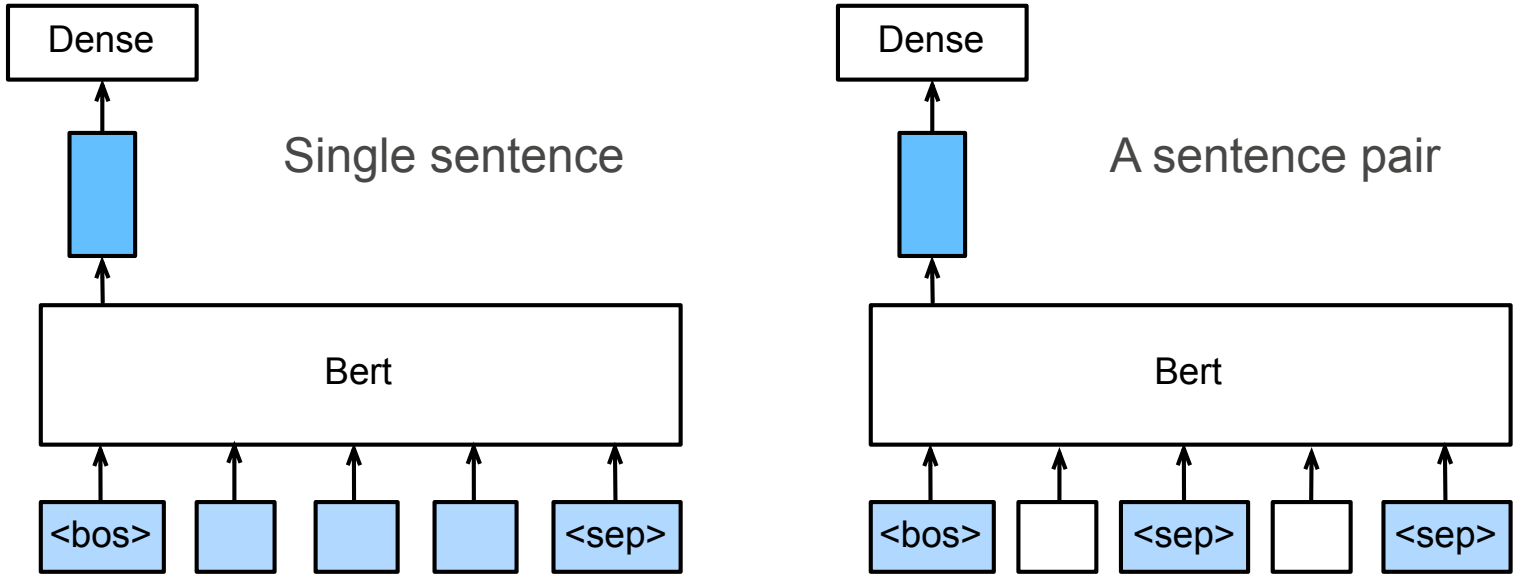
Bert for Fine Tuning

- Bert returns a feature vector for each token that captures the context information
- Different fine-tuning tasks use a different set of vectors



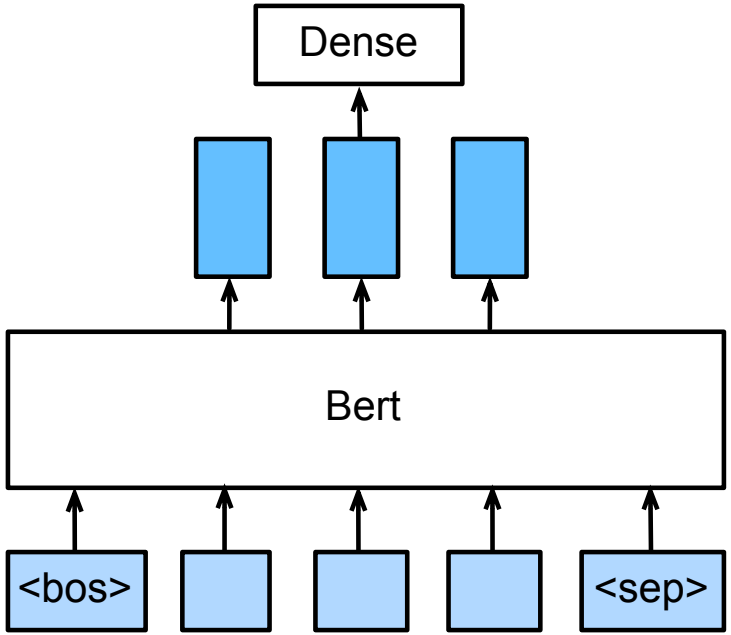
Sentences Classification

- Feed the <bos> token vector into a dense output layer



Named Entity Recognition

- Identify if a token is a named entity such as person, org, and locations…
- Feed each non-special token vector into a dense output layer



Question Answering

- Given a question and a description text, find the answer, which is a text segment in the description
- Given $\mathbf{p}_i$ the *i*-th token in the desperation, learn $\mathbf{s}$ so that

$$p_1, \ldots, p_T = \text{softmax}(\langle \mathbf{s}, \mathbf{v}_1 \rangle, \ldots, \langle \mathbf{s}, \mathbf{v}_T \rangle)$$

p_i is the probability *i*-th token is the segment start. Same for the end