Introduction to Deep Learning

10. Sequence Sampling

Haibin Lin and Leonard Lausen

gluon-nlp.mxnet.io

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

I have a language model / machine translation model, how to generate texts?

Generating Text

- Language model

$$p(\text{text}) = \prod_t p(w_t | [w_{t-1} \ldots w_1])$$

- Sample from language model, one character/word at a time
- Need to **search** over lots of possible sequences

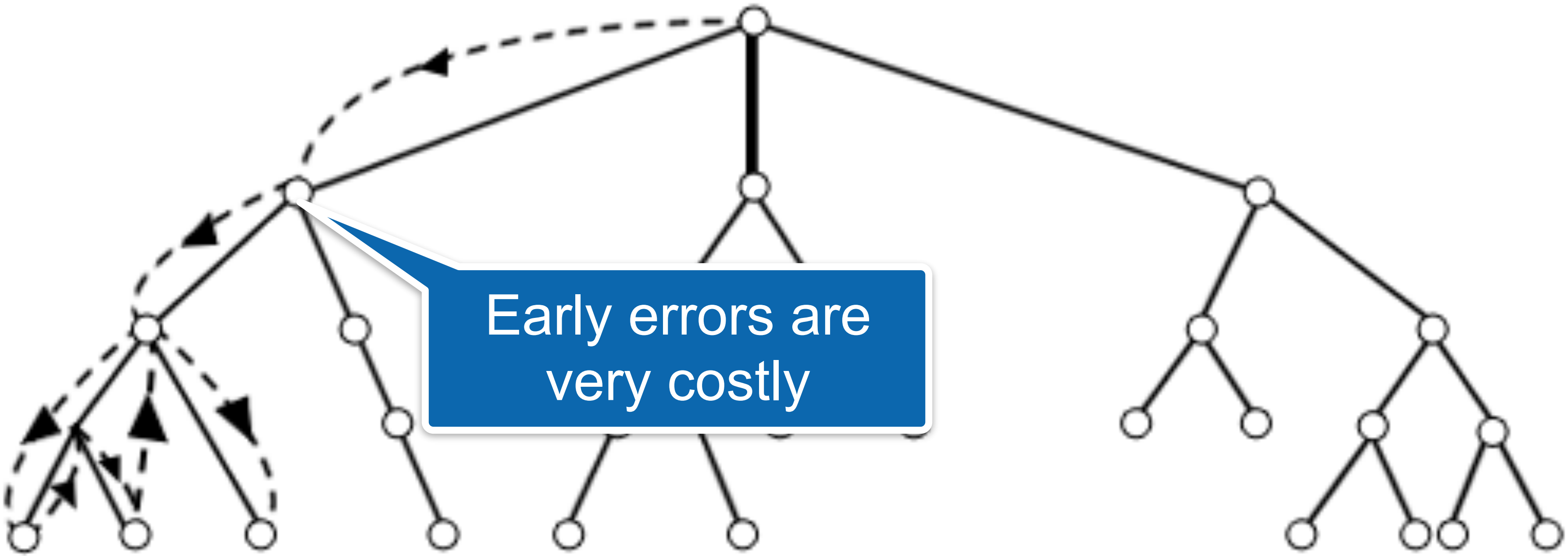
Greedy Search

- Greedy search during predicting could be suboptimal

Greedy search:
$0.5 \times 0.4 \times 0.4 \times 0.6 = 0.048$

Time step	1	2	3	4
A	**0.5**	0.1	0.2	0.0
B	0.2	**0.4**	0.2	0.2
C	0.2	0.3	**0.4**	0.2
<eos>	0.1	0.2	0.2	**0.6**

Depth first search



Greedy Search

- Greedy search during predicting could be suboptimal

Greedy search:
$0.5\times0.4\times0.4\times0.6=0.048$

Time step	1	2	3	4
A	**0.5**	0.1	0.2	0.0
B	0.2	**0.4**	0.2	0.2
C	0.2	0.3	**0.4**	0.2
<eos>	0.1	0.2	0.2	**0.6**

A better choice:
$0.5\times0.3\times0.6\times0.6=0.054$

Time step	1	2	3	4
A	**0.5**	0.1	0.1	0.1
B	0.2	0.4	**0.6**	0.2
C	0.2	**0.3**	0.2	0.1
<eos>	0.1	0.2	0.1	**0.6**

Exhaustive Search

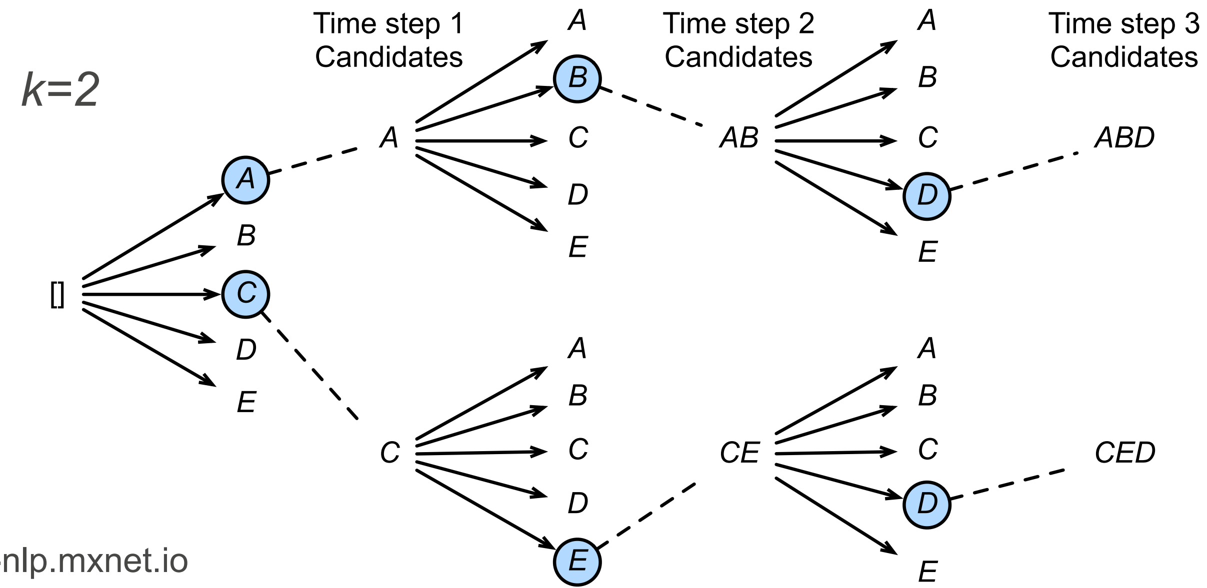
- For every possible sequence, compute its probability and pick the best one
- If output vocabulary size is *n,* and max sequence length *T,* then we need to examine n^T sequences
  - It's computationally infeasible

$$n = 10000, \quad T = 10: \quad n^T = 10^{40}$$

Beam Search

Beam Search

- We keep the best *k* (beam size) candidates for each time
- Examine *kn* sequences by adding an new item to a candidate, and then keep the top-*k* ones



Beam Search

- Time complexity is O(knT)

$$k = 5, \quad n = 10000, \quad T = 10: \qquad knT = 5 \times 10^5$$

- The final score for each candidate is

$$\frac{1}{L^\alpha} \log \mathbb{P}(y_1, \ldots, y_L) = \frac{1}{L^\alpha} \sum_{t'=1}^{L} \log \mathbb{P}(y_{t'} \mid y_1, \ldots, y_{t'-1}, \boldsymbol{c})$$

  - Often $\alpha = 0.75$

Goldilocks

- **Avoid pathological cases** (Wu et al, 2016)
  - “”
  - “La La La La La La La …”
  - Partial translations in machine translation
- Length penalty, such as $(l+5)^\alpha$ to normalize for variable segment lengths
- Submodular Coverage penalty avoids missing segments

$$\sum_i \log \min\Big(\sum_j \alpha_{ij}, 1\Big)$$