L9-1

SLIDE 1 : RECAP CNNs, RNNs

SLIDE 2:

① Diff BETWEEN SEQ → LABEL
SEQ → SEQ

SEQ-2 LABEL

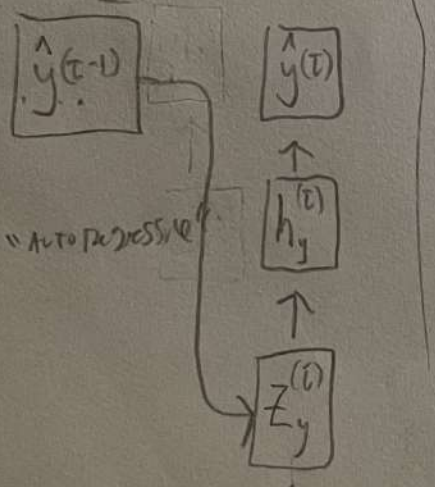
Model: $P(y^{(T)} \mid x^{(1)}, \ldots, x^{(T)})$

Seq2Seq: $P(y^{(1)}, \ldots, y^{(T_y)} \mid x^{(1)}, \ldots, x^{(T_x)})$ ①

EXAMPLES: MACHINE TRANSLATION $x^{(1)}, \ldots, x^{(T_x)}$ FRENCH SENTENCE
$y^{(1)}, \ldots, y^{(T_y)}$ ENGLISH SENTENCE

TEXT GENERATION $x^{(1)} \ldots x^{(t)}$ CONTEXT
$y^{(1)}, \ldots, y^{(T_y)} = x^{(t+1)}, \ldots, x^{(T_x)}$

② → Refer to SLIDE



NOTES: • $\hat{y}^{(0)}$ = <START>

• DECODER STOPS WHEN
↳ WHEN $\hat{y}$ = <STOP>

The Autoregressive Decoder

3

Algorithm

1. Compute $h_x^{(T_x)}(x^{(1)}, \dots, x^{(T_x)})$
2. Set $t = 1$, $\hat{y}^{(0)} = \langle \text{START} \rangle$, $h_y^{(0)} = h_x^{(T_x)}$
3. While TRUE:

$h^{(t)} = f_h(h_y^{(t-1)}, \hat{y}^{(t-1)})$ → Compute Hidden State

$y^{(t)} = f_y(h^{(t)})$ → Predict t^{th} word

If $y^{(t)} =$ "$\langle \text{STOP} \rangle$":

Break;

$t = t + 1$

NOTE:

Seq2Label

$h^{(t)} = f_h(h^{(t-1)}, x^{(t)})$

$x^{(t)}$ comes from the data

Seq2Seq

$h^{(t)} = f_h(h^{(t-1)}, y^{(t-1)})$

$y^{(t-1)}$ comes from the model

The "Inference Gap"

4

Training task: $\hat{\theta} = \underset{\theta}{\text{argmax}}\ P(y^{(t)} \mid f_h(h^{(t-1)}, y^{(t-1)}); \theta)$

Prediction task: $\hat{y}^{(1)}, \dots, \hat{y}^{(T_y)} = \underset{y^{(1)}, \dots, y^{(T_y)}}{\text{argmax}}\ P(y^{(1)}, \dots, y^{(T_y)} \mid f_h(h_x^{(T_x)}, y^{(0)}; \theta)$

Our training and prediction tasks have diverged!

INFERENCE GAP Cont. –

TEACHER FORCING:

$$L_{CE}\left(y^{(t)}, P(y^{(t)} \mid y^{(1)}, \ldots, y^{(t-1)})\right)$$

y^t = t^{TH} word in training example

$\hat{y}^t$ = t^{th} prediction

⇒ trained only on ground truth context words

SCHEDULED SAMPLING:

$$L_{CE}\left(y^{(t)}, P(y^{(t)} \mid \tilde{y}^{(1)}, \ldots, \tilde{y}^{(t-1)})\right)$$

(6)

where $\tilde{y}^{(t)} = \begin{cases} \hat{y}^{(t)} & \text{with probability } p \\ y^{(t)} & \text{with prob } 1-p \end{cases}$

↳ EVEN WITH SCHEDULED SAMPLING: THE TRAINING AND INFERENCE DISTRIBUTIONS ARE DIFFERENT, UNLESS θ IS A PERFECT MODEL OF THE DATA.

CAN'T WE JUST ESTIMATE $P(y^{(1)}, \ldots, y^{(T_y)})$ USING MLE?

(4)

$$P(y^{(1)}, \ldots, y^{(T_y)}) = \frac{1}{T_y} \prod_{t=1}^{T_y} P(y^{(t)} \mid y^{(1)}, \ldots, y^{(t-1)}; \theta)$$

↳ NO, SCALES EXPONENTIALLY WITH SEQ LEN T_y

↳ EX: $N = 100{,}000$ VOCAB SIZE, $T_y = 5$ ⇒ WE'D NEED TO EVAL $(10^5)^5 = 10^{25}$ POSSIBILITIES!

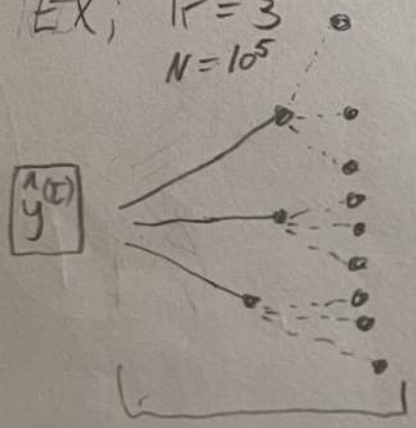
INFERENCE GAP Cont...

APPROXIMATE SEARCH ALGORITHMS:

BEAM SEARCH.

IDEA: At every autoregressive step, only consider top-K most probable words

EX; $K=3$
$N=10^5$



2 autoregressive steps

ORIGINAL PROBLEM:

$N^2 = 10^{10}$

$K^2 = 9$

↳ IN PRACTICE THIS WORKS FAIRLY WELL.

---

2 PRIMARY CHALLENGES WITH SEQ2SEQ MODELS:

① THE INFORMATION "BOTTLENECK"

② THE INFERENCE GAP

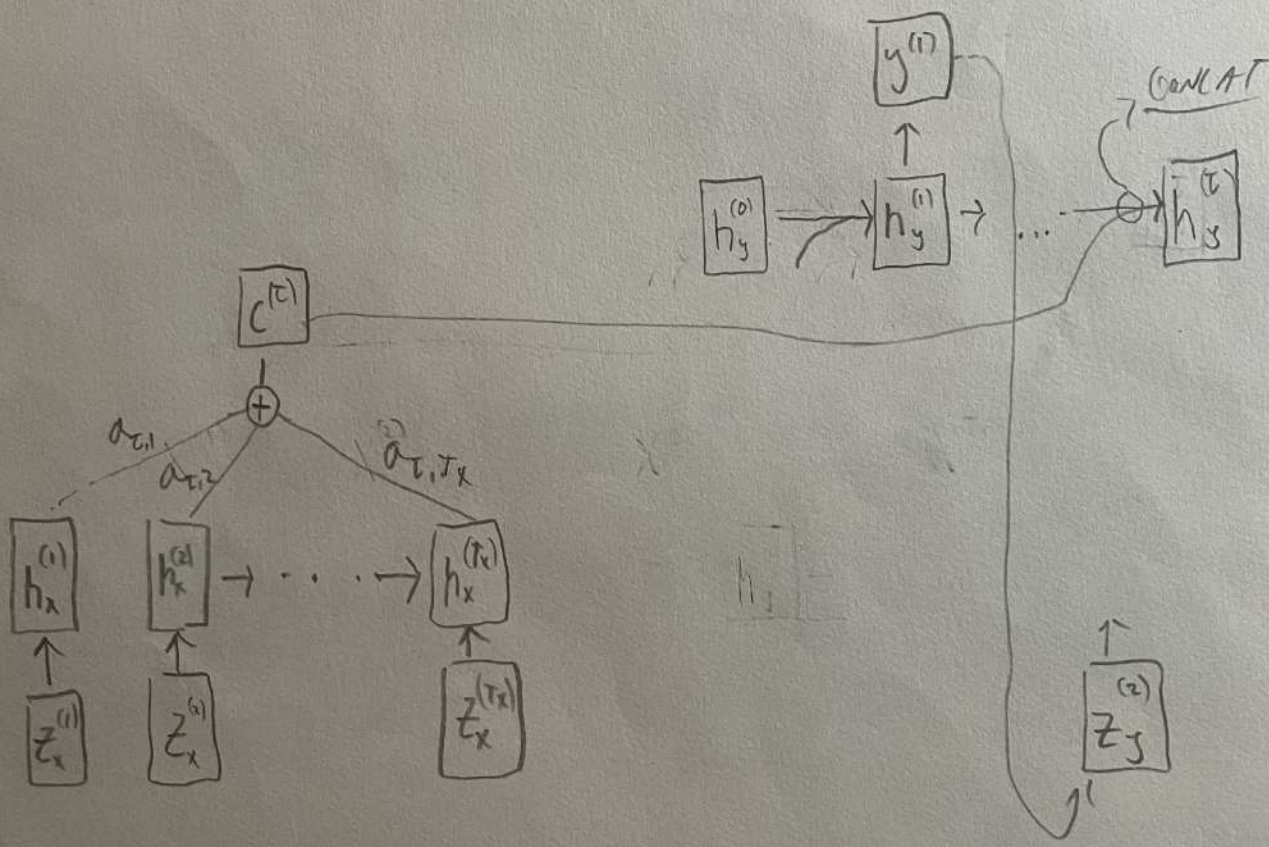
↳ BOTH OF THESE CHALLENGES HAVE BEEN ADDRESSED WITH THE ADVENT OF "ATTENTION"!

FIRST: THE INFORMATION BOTTLENECK

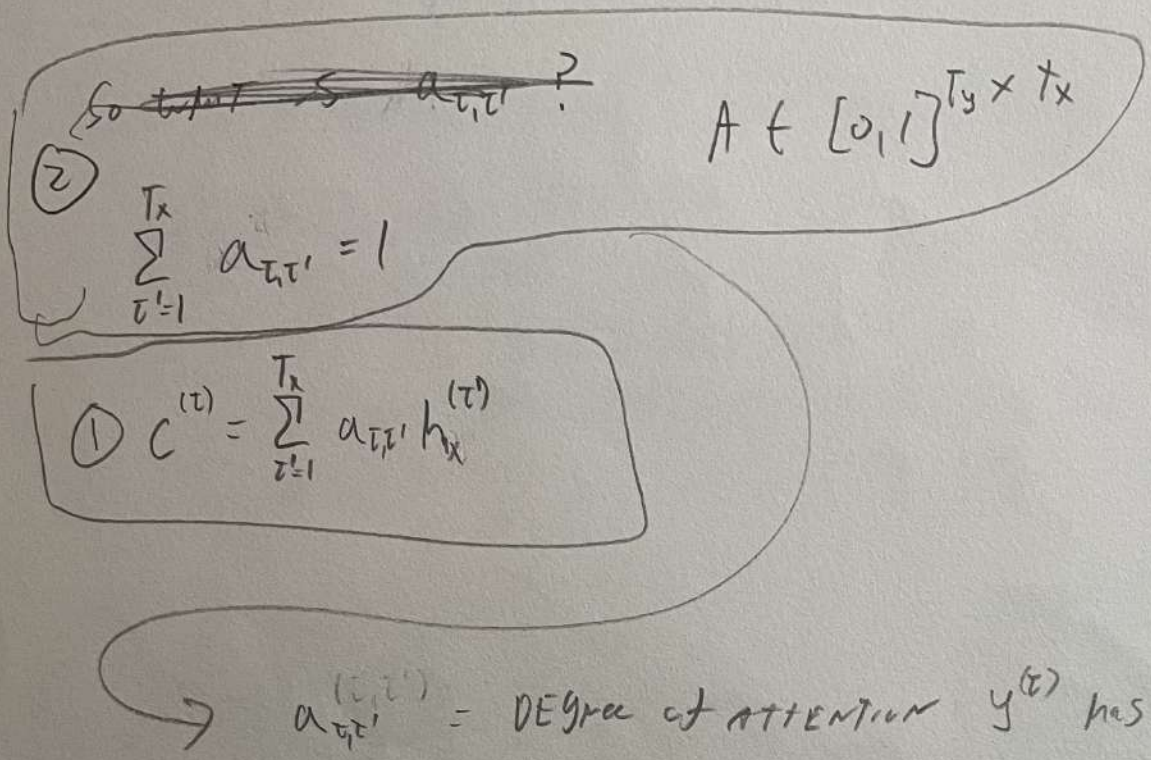
ATTENTION BASED SEQ2SEQ MODELING | L9-5

- STANDARD BIDIRECTIONAL LSTM BASED E-D ARCH WITH ONE KEY ADDITION: AN ATTENTION MECHANISM BETWEEN THE DECODER AND EACH HIDDEN STATE IN THE ORIGINAL SEQ X.

$$h^{(t)} = \{ \overrightarrow{h}^{(t)}, \overleftarrow{h}^{(t)} \}$$



Attention Based Seq2Seq Models Cont...



$a_{t,t'}^{(t,t')}$ = DEgree of ATTENTION $y^{(t)}$ has on $x^{(t')}$

$$a_{t,t'} = \frac{e^{\alpha_{t,t'}}}{\sum_{t''=1}^{T_x} e^{\alpha_{t,t''}}}$$

α = ① DOT-PRODUCT $\alpha_{tt'} = h_y^{(t)} \cdot h_x^{(t')}$

② NEURAL NET $\alpha_{tt'} = f_{h_y}(h_y^{(t)}) \cdot f_{h_x}(h_x^{(t')})$

↳ where $f_h(\cdot;\theta) = NN$

Attention is All You Need Cont. –

→ Notice that each input $h^{(t)}$ is used in 3 different ways!

(2) → Attention as a Fuzzy lookup table

For every $c^{(t)}$ there is

① a Query: $h^{(t)}$

② T separate keys: $h^{(1)}, \ldots, h^{(T)}$

③ T separate values: $h^{(1)}, \ldots, h^{(T)}$

→ $c^{(t)} = \text{DICT}[\text{Query}^{(t)}]$ ← fuzzy lookup

$$= \sum_{t'=1}^{T} \text{VALUE}^{(t')} \cdot \sigma_{\text{SOFTMAX}}(\text{Query}^{(t)} \cdot \text{Key}^{(t')})$$

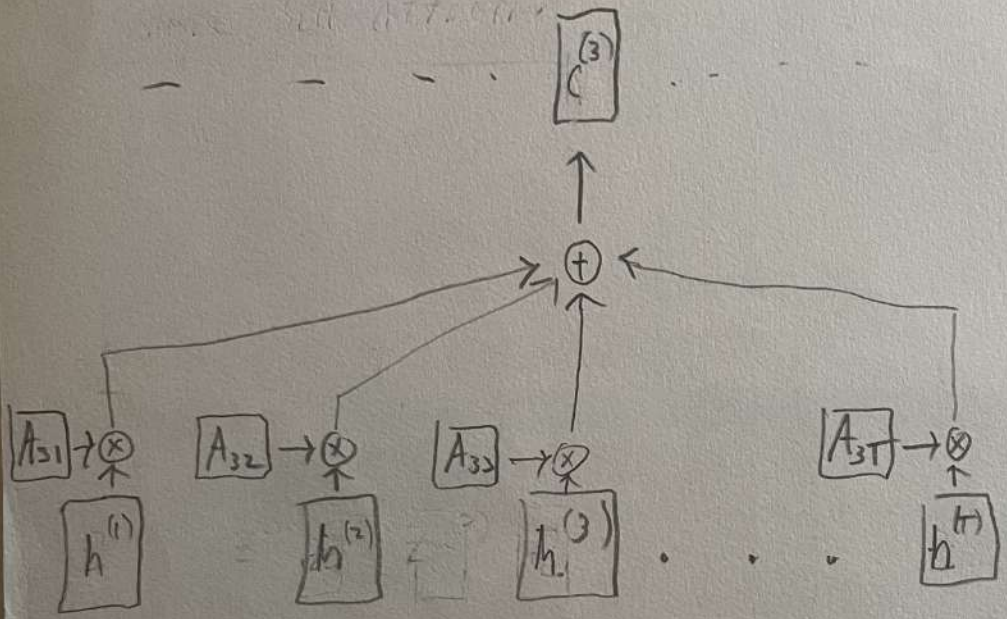
$$= \sum_{t'=1}^{T} h^{(t')} \cdot \sigma_{\text{SOFTMAX}}(h^{(t)} \cdot h^{(t')})$$

↳ The context vector at t^{th} step is the sum of all inputs $h^{(1)} \ldots h^{(T)}$ weighted by the similarity factor $\sigma_{\text{SOFTMAX}}(h^{(t)}, h^{(t')})$

Simple Self Attention

Slide 7

IDEA: Could we just rely on attention alone and do away with RNNs?



$$c^{(t)} = \sum_{t'} A_{t,t'} \cdot h^{(t')} \quad \text{where} \quad A_{t,t'} = \frac{e^{h^{(t)} \cdot h^{(t')}}}{\sum_{t''} e^{h^{(t)} \cdot h^{(t'')}}}$$

Where sum over any column of $A = 1$

Simple Self Attention Cont...

NOTE: Simple Self Attention is NOT a sequence model (necessarily), depends on how A is computed, the simple self attention model is a set model.

Attention is All You Need

[redacted]

Introduced the transformer model (Slide 8)

How'd they get it to work?

① Scaled Dot-Product Self Attention

↳ Problem: We're computing a Soft Max over dot products! The magnitude of the $\langle \cdot \rangle$ scales with $\sqrt{D}$

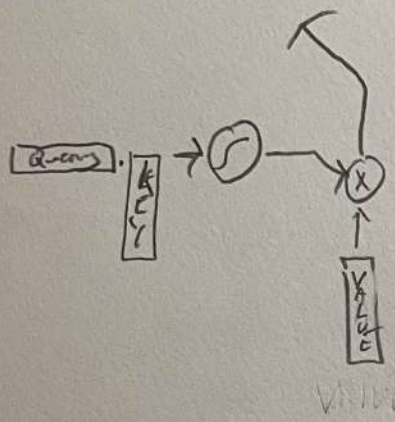
↳ or: $[1,1]^2 = \sqrt{2}$

$[1,1,1]^2 = \sqrt{3}$

$[1,1,1,1] = \sqrt{4}$

So: $A \Rightarrow \frac{A}{\sqrt{D}}$

THE K, Q, V IDEA MORE GENERICALLY:



EACH INPUT IS USED AS THE KEY, QUERY, AND VALUE

③ ~~[illegible]~~ A LEARNABLE SET OF KEYS, QUERIES, VALUES

$$\left.\begin{aligned} k_t &= K h^{(t)} + b_K \\ q_t &= Q h^{(t)} + b_Q \\ v_t &= V h^{(t)} + b_V \end{aligned}\right\} \quad \Theta \mathrel{+}= \{K, Q, V, b_K, b_Q, b_V\}$$

④ MULTI-HEAD SA

$$K = \{K^{(1)}, \dots K^{(n)}\}$$
$$Q = \{Q^{(1)}, \dots, Q^{(n)}\}$$
$$V = \{V^{(1)}, \dots, V^{(n)}\}$$

↳ THIS GIVES US n CONTEXT VECTORS AT EACH SEQUENCE POSITION t: $c^{(t)} = [c_1^{(t)}, \dots, c_n^{(t)}]$

↳ IN PRACTICE WE CAN MAKE EACH $K, Q, V \in \mathbb{R}^{D/n}$ SUCH THAT $c^{(t)} \in \mathbb{R}^D$