PART 3

Language Models

Example: what is the problem?

Predict the next word

I are a student

P = 0

i	1	2	3	4
X_i	what	is	the	problem
Y_i	is	the	problem	EOS

* $P(W_K \mid W_1, \cdots W_{K-1})$ NLG $P(W_1) = P(W_1 \mid \text{Begin})$

* $P(W_1, W_2, \cdots W_n) = P(W_1)\, P(W_2 \mid W_1) \cdots P(W_n \mid W_1 \cdots W_{n-1})$

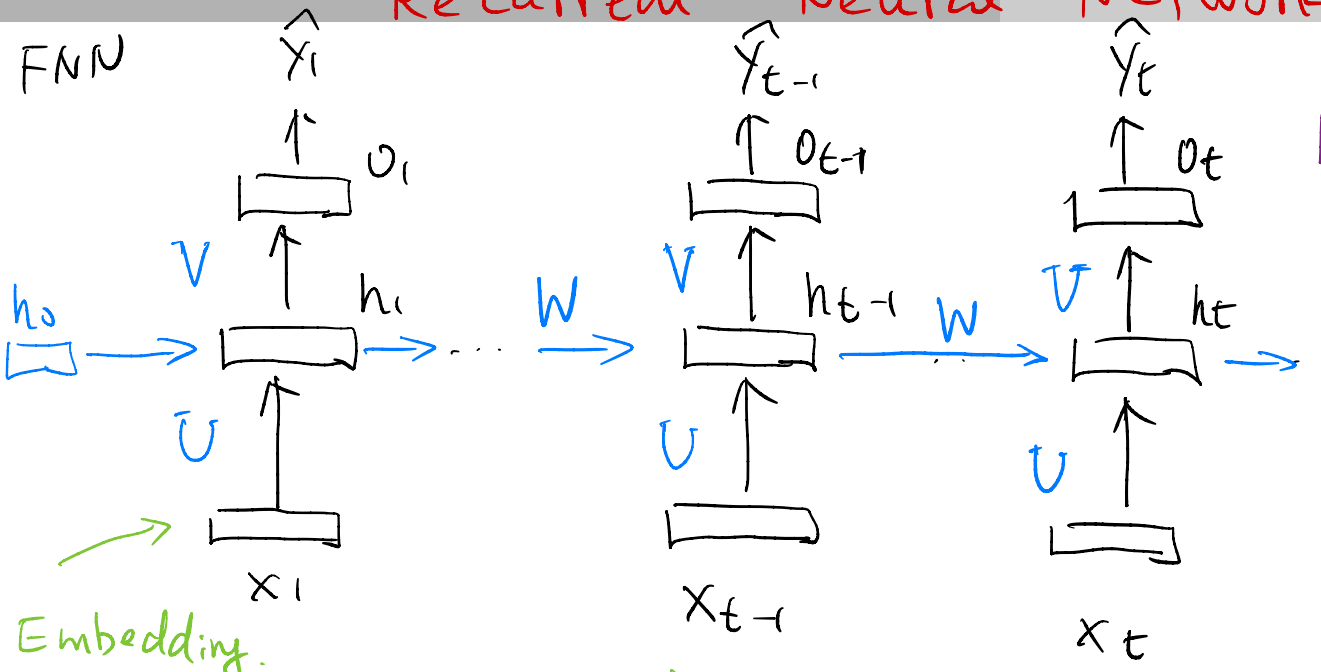
↖ prob of sentence

Dialogue: * How are you ?

* I am fine

i	1	2	3	4	5	6
X_i	How	are	you	–	–	–
Y_i	–	–	–	I	am	fine

Recurrent Neural Network (RNN)



$h_t = \tanh(U x_t + W h_{t-1} + b)$

z

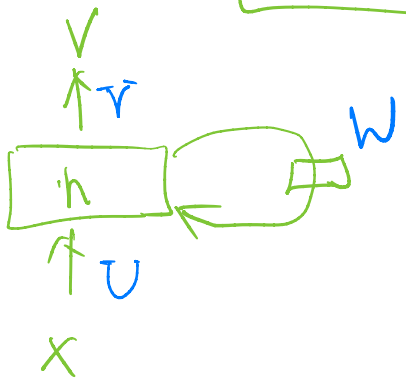
$o_t = V h_t + c$

$P(y_t \mid x_t, x_{t-1}, \dots x_1)$

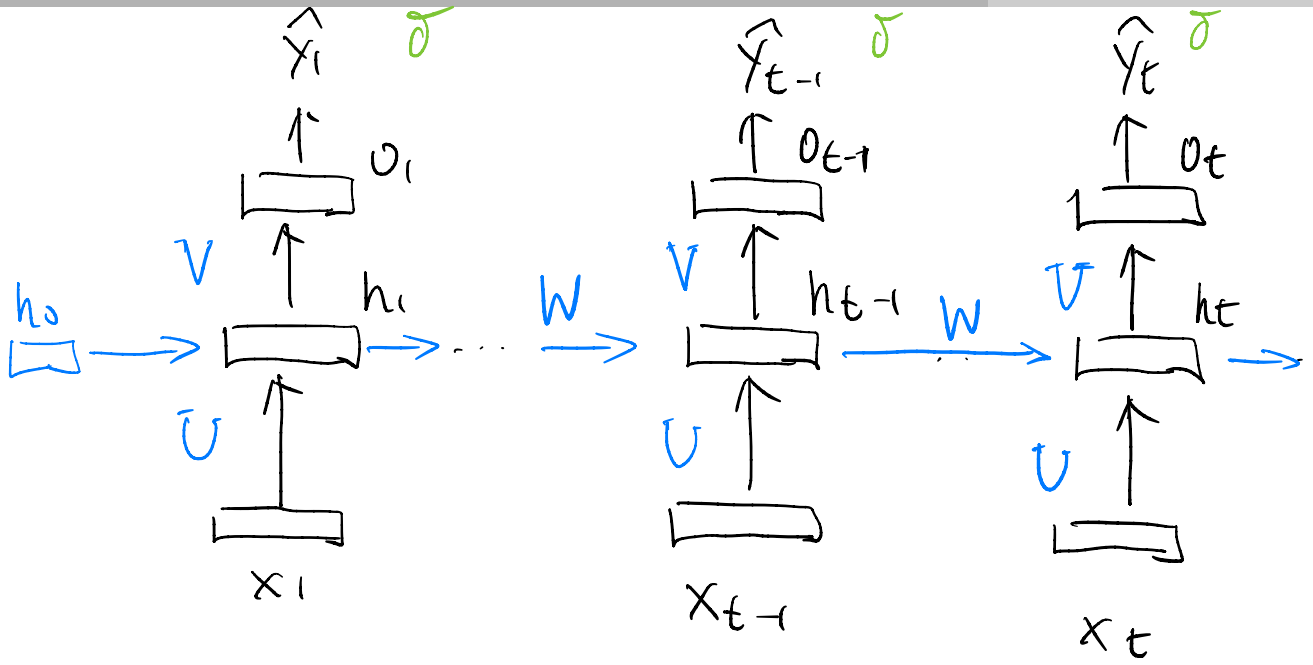
$= \text{Softmax}(o_t)$

rolled-out model

Circuit diagram



Time delayed dependence



$$h_t = \tanh(U x_t + W h_{t-1} + b)$$

$$\Rightarrow o_t = V h_t + c$$

$$P(y_t \mid x_t, x_{t-1}, \ldots x_1) = \text{Softmax}(o_t)$$

Parameters: $W, U, V, b, c, \theta_{em}$ (embedding vectors)

$$L = E\left[\frac{1}{T}\sum_{t=1}^{T} -\log P(y_t \mid x_t, \ldots x_1)\right]$$

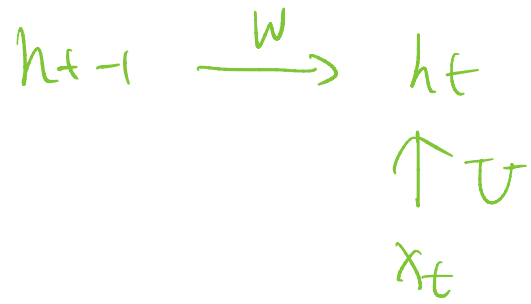
$\nabla_W L, \nabla_U L \ldots$ Average over pairs of training sequence

LSTM

Standard RNN Cell

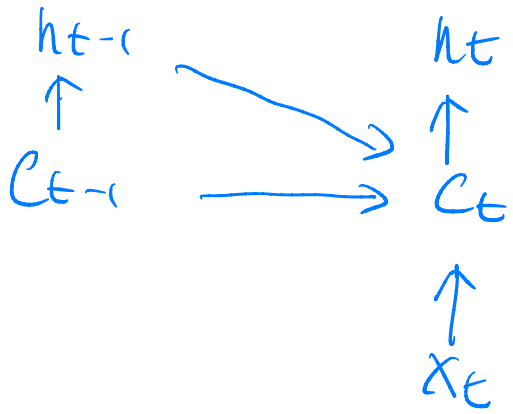
$$a_t = U x_t + W h_{t-1} + b$$

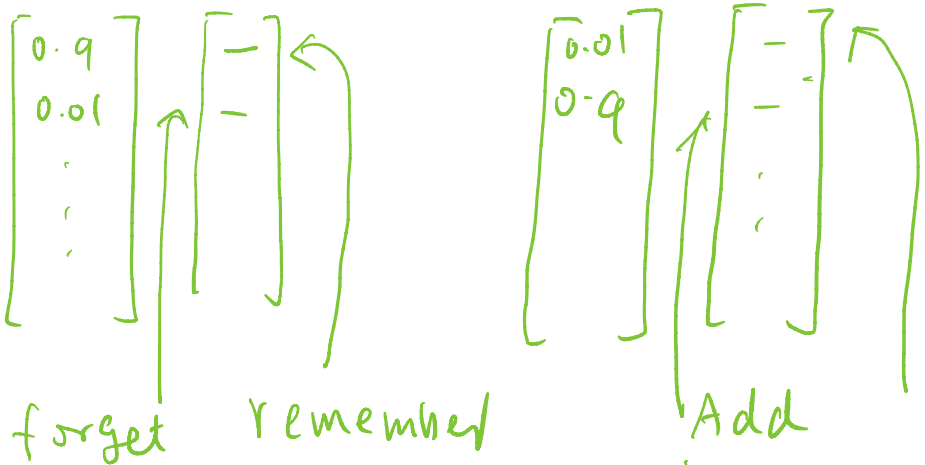
$$h_t = \tanh(a_t)$$



LSTM Cell

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \tanh(a_t)$$



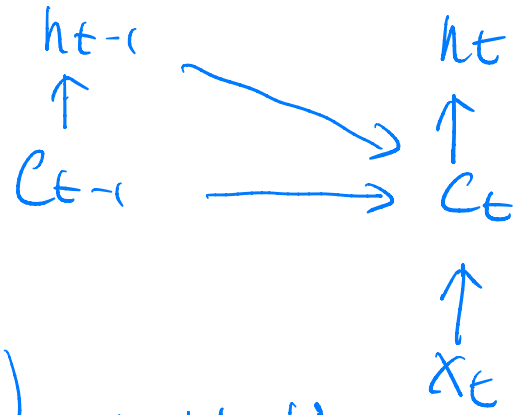


f_t : forget gate

i_t : input gate

LSTM cell

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \tanh(a_t)$$

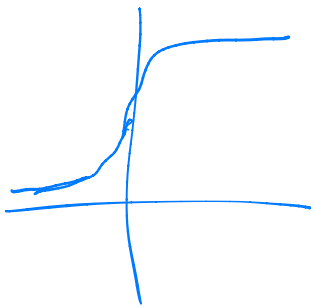


$$f_t = \sigma\left(W_f h_{t-1} + U_f x_t + b_f\right)$$

$$i_t = \sigma\left(W_i h_{t-1} + U_i x_t + b_i\right)$$

* W_f, U_f, b_f learnable para

* Sigmoid good as gate



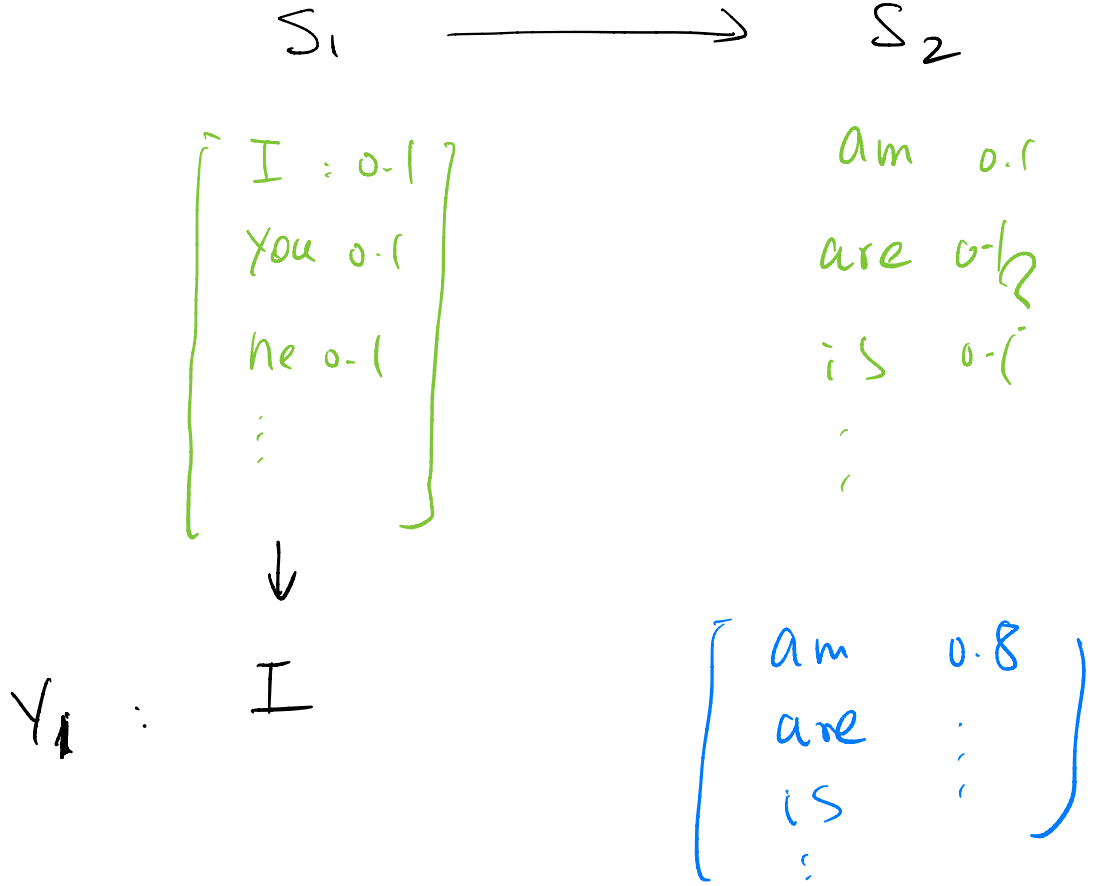
$$h_t = O_t \otimes \tanh(C_t)$$

O_t : output gate $\begin{bmatrix} 0.9 \\ 0.01 \\ 0.7 \\ \vdots \end{bmatrix}$

$$O_t = \sigma\left(W_o h_{t-1} + U_o x_t + b_o\right)$$

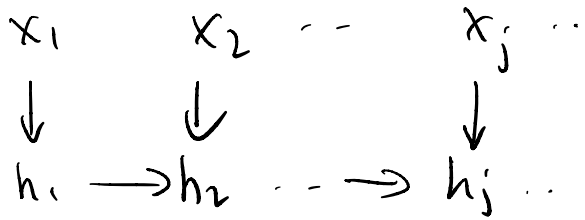
Encoder - Decoder / Seq 2 Seq Model Architecture

Decoder



Attention

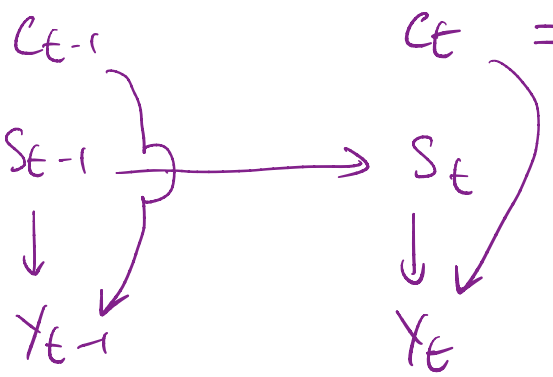
Encoder



$$\alpha_{tj} \leftarrow c_{T-1}^{\top} \cdot h_j$$

$$\alpha_{tj} \leftarrow \frac{e^{\alpha_{tj}}}{\sum_j e^{\alpha_{tj}}}$$

Decoder



$c_t = \sum_j \alpha_{tj} h_j$: α_{tj} – how much attention to h_j at t (decoder)

① c_{t-1} (what focused on at $t-1$)

Query

- held
- drink

c_t (what to focus on at t)

talk, meeting, cup, ...

water, juice - -

② what info at j: h_j

Key

$z_1 \dots z_i \dots$

$x_1 \dots x_i \dots x_j \dots$

$1 \times d_m$

512

At z_i, how much attention to pay to x_j ?

Query: $q_i = x_i W^Q \rightarrow d_m \times d_k$ (64)

$1 \times d_k \quad 1 \times d_m$ learnable

key: $k_j = x_j W^K$

$\alpha_{ij} = q_i k_j^T / \sqrt{d_k}$ scaled ...

↑ Dot product attention

$$\alpha_{ij} \leftarrow \frac{e^{\alpha_{ij}}}{\sum_j e^{\alpha_{ij}}}$$

$$z_i = \sum_j \alpha_{ij} v_j$$

$1 \times d_v$

64

Value: $v_j = x_j W^V \rightarrow d_m \times d_v$ (64)

$1 \times d_v$

Je

suis

etudiante

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \quad Q = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \end{bmatrix} \quad K = \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_N \end{bmatrix} \quad V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix} \quad Z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix}$$

$N \times d_m$ $N \times d_k$ $N \times d_k$ $N \times d_v$ $N \times d_v$

$$Q = X W^Q$$
$$K = X W^K$$
$$V = X W^V$$

$$Z = \text{Softmax}\left(\frac{Q K^T}{\sqrt{d_k}}\right) V$$

$N \times N$

Softmax

Single-head Self-attention

$X \xrightarrow{W^Q, W^K, W^V} Z$

$N \times d_m$ $\quad$ $N \times d_v$

Multi-head self-attention

$X \xrightarrow{W_1^Q, W_1^K, W_1^V} Z_1$

$\vdots$

$\xrightarrow{W_h^Q, W_h^K, W_h^V} Z_h$

$h = 8$

$h = 16$

$Z = \text{Cat}(Z_1, \cdots, Z_h)$

$N \times h d_v$

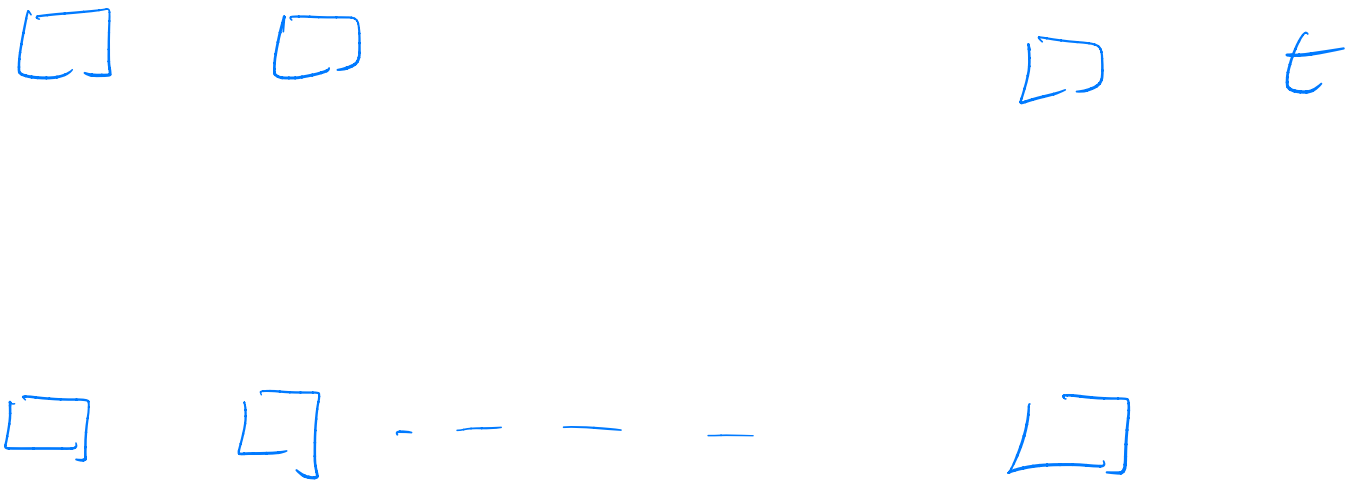
$d_m = 512$

$512 = 8 \times 64$

$1024 = 16 \times 64$

$Z \leftarrow Z W^O$

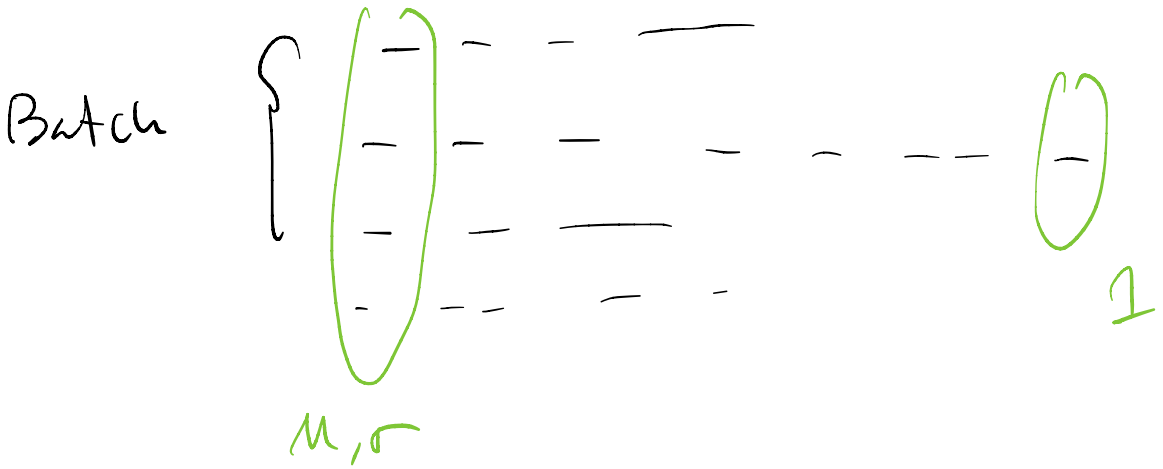
$N \times d_m \quad N \times h d_v \quad h d_v \times d_m$


t
t
Complexity : t^2
512 x 512

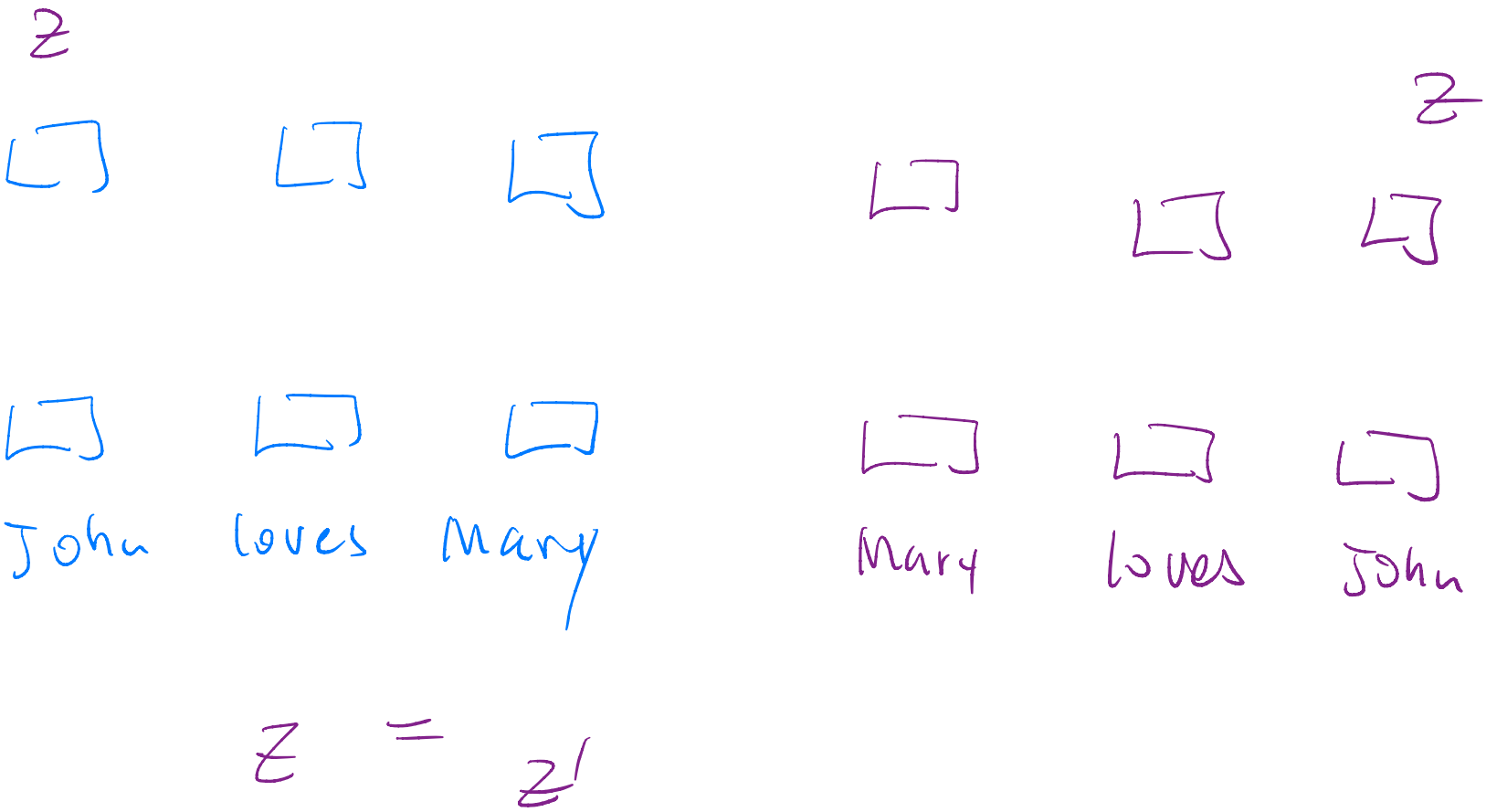
Why Not Batch Normalization in NLP (Transformer)

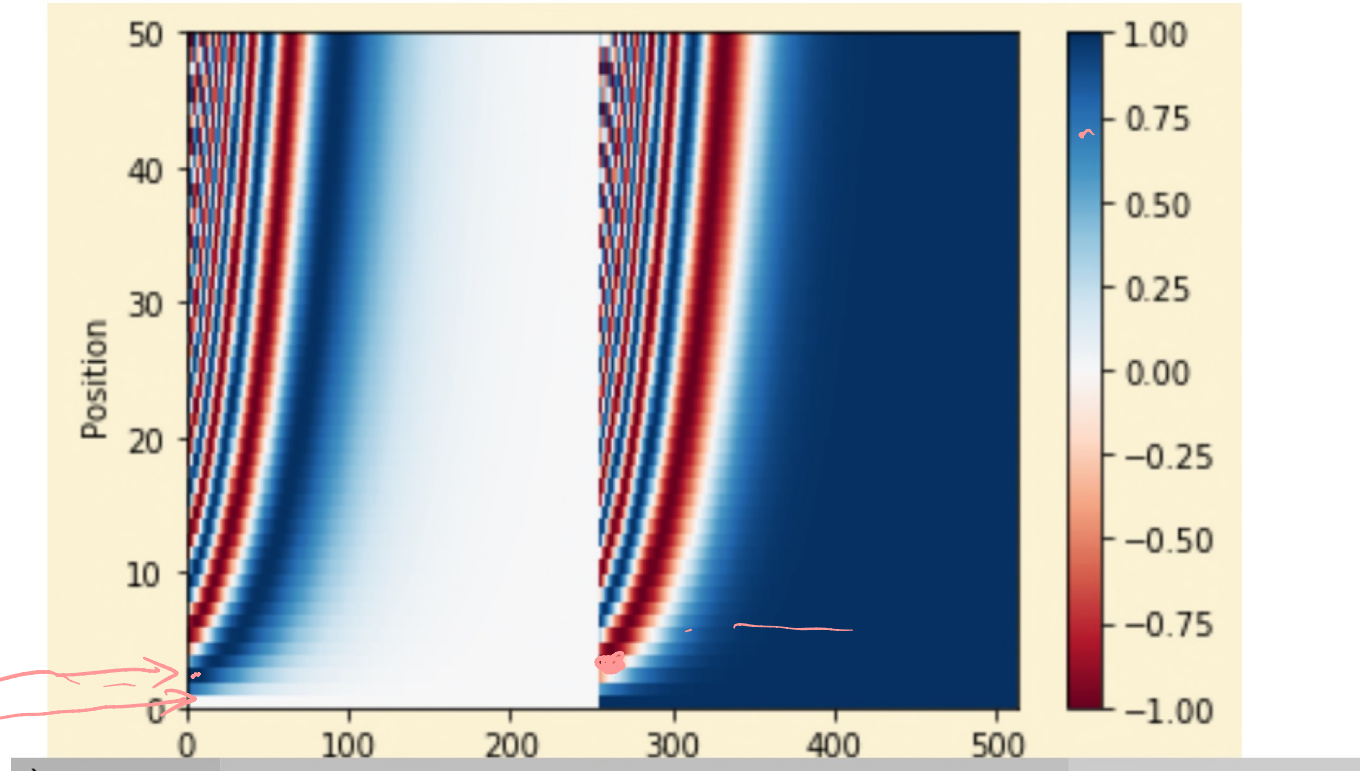
* BN destroys sequential dependence

* Sentence length varies



* Memory


z
John loves Mary
z'
Mary loves John
z = z'



Je [0, ... 0, 1, 1 ... 1]

suis

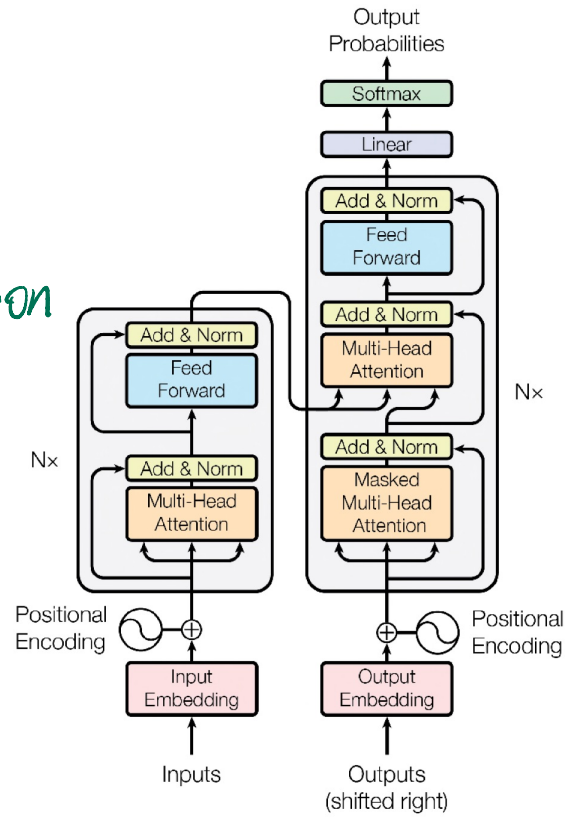
'etudiant [0.7 ... 0 -0.7 ... 1]

Transformer

* Encoder + Decoder
* Seq2seq model

BERT: Bidirectional Encoder Representation from Transformer

* Encoder only
* Representation leaning
  ⇒ Downstream NLU tasks



GPT: Generative Pre-trained Transformer

* Decoder only
* For NLG tasks