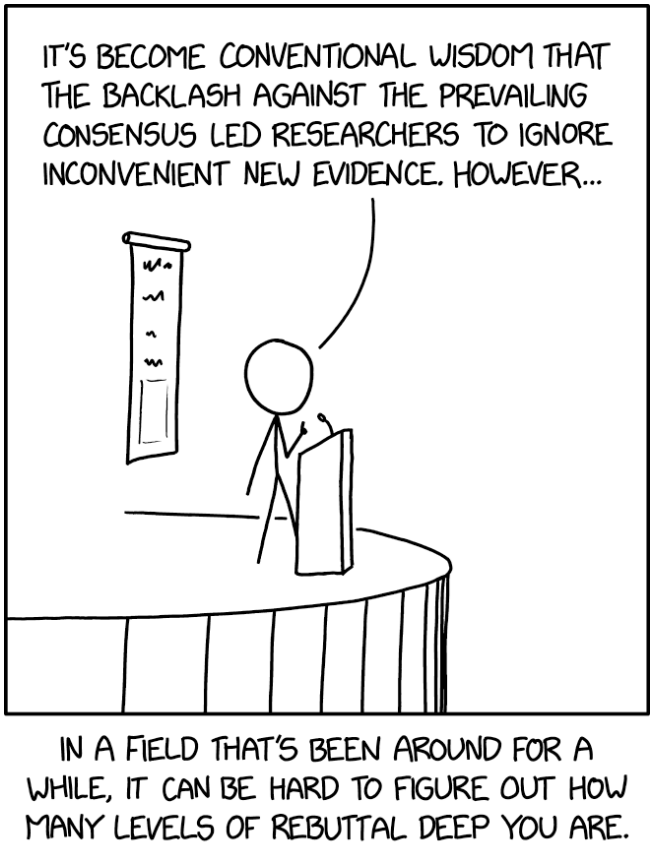
COMP 4446 / 5046
Lecture 7: Models – Transformer

Jonathan K. Kummerfeld

Semester 1, 2025

menti.com 2520 1730

Rebuttals




```
[The mainstream dogma sparked a wave of dogmatic
revisionism, and this revisionist mainstream
dogmatism has now given way to a more rematic
mainvisionist dogstream.]
```

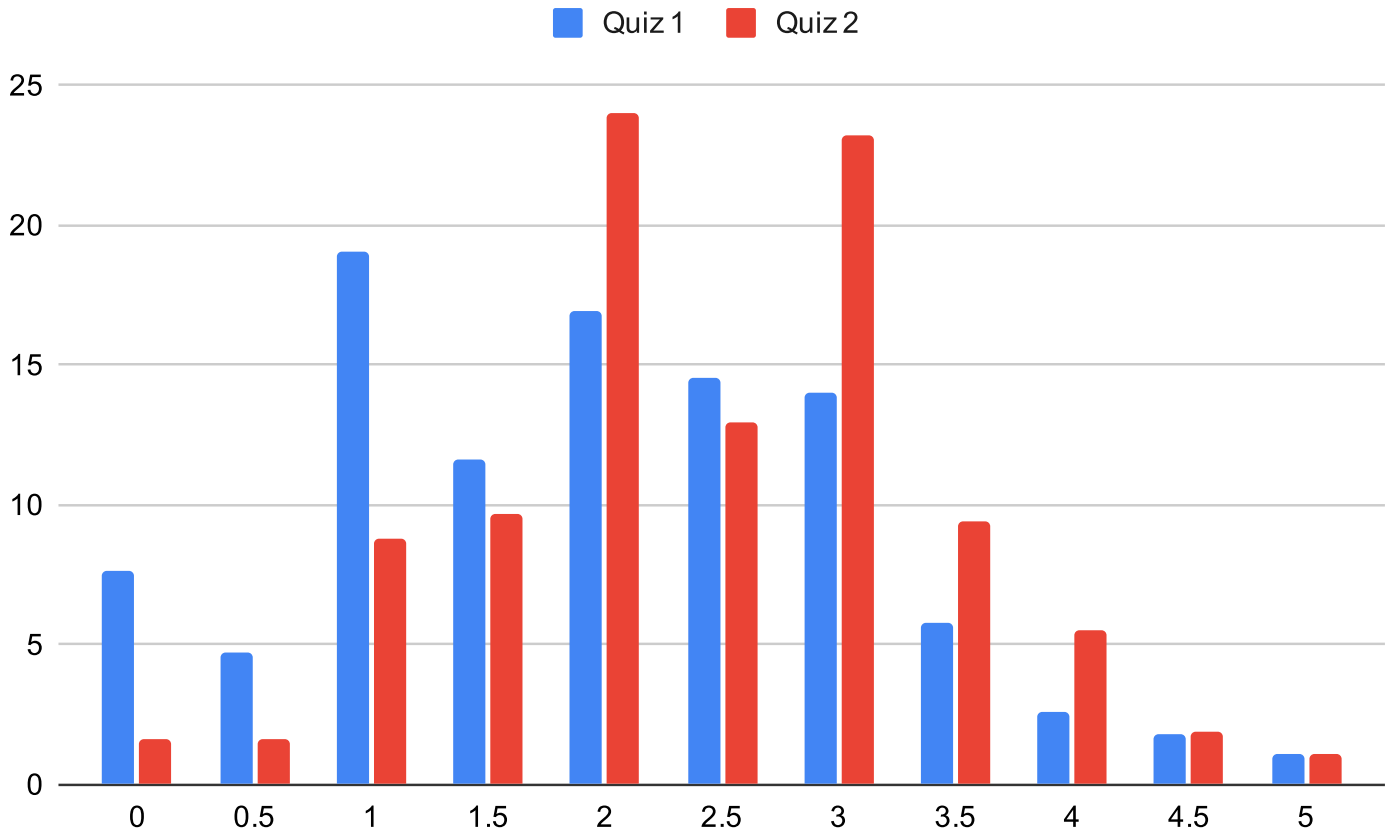
Source: https://xkcd.com/2857/

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Quiz results - an improvement! But still 22% below 2



To avoid the bugs from this quiz, I will be getting TAs to check them in future (just as exams are reviewed by another academic)

Note: resu ts
credit for t

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Special Consideration Reminder

Short release assignments - No simple extensions

Regular Special Consideration is not processed fast enough

Slip days instead!

For major issues (e.g., broke your arm and can’t type for a month), go through Special Consideration process and a more significant change will be determined

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Office Hours shift

This week Office Hours will be 2-3pm on Tuesday

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Self-Attention

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

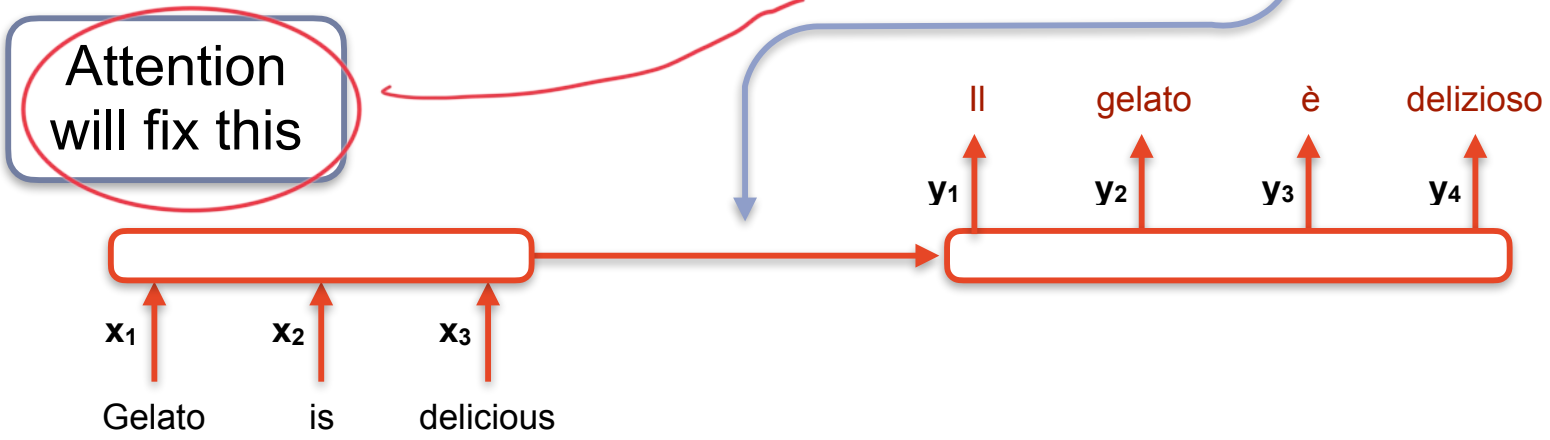
menti.com 2520 1730

Reminder: Issues with the encoder-decoder

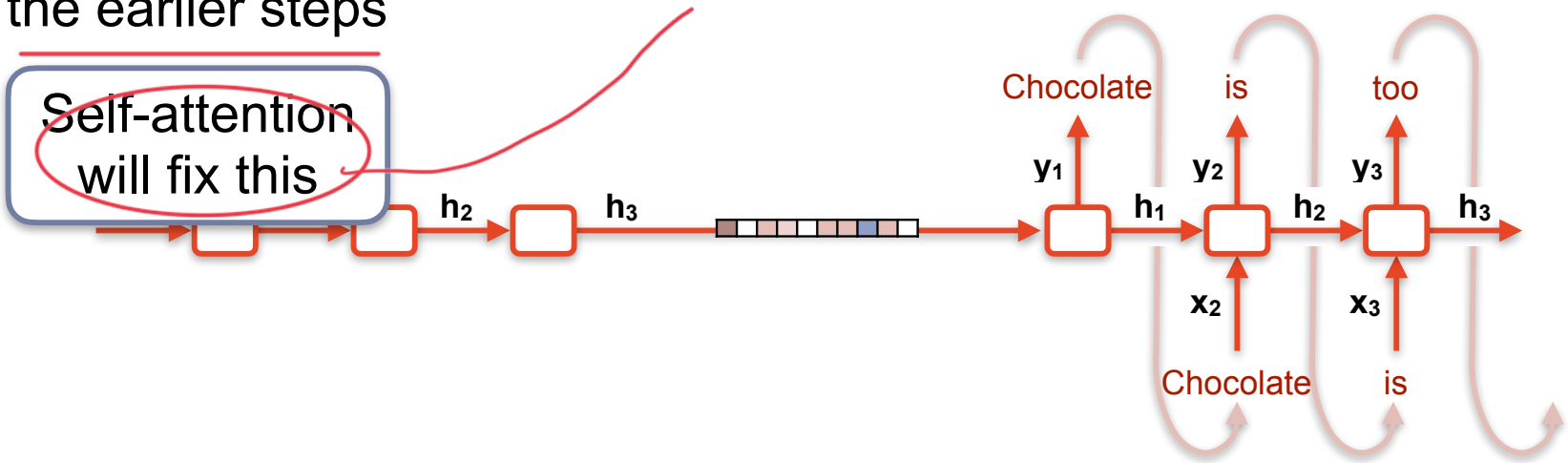
Review

Data loss

Bottleneck - all information has to go through one vector



Hard to parallelise - every step of the calculation depends on the earlier steps



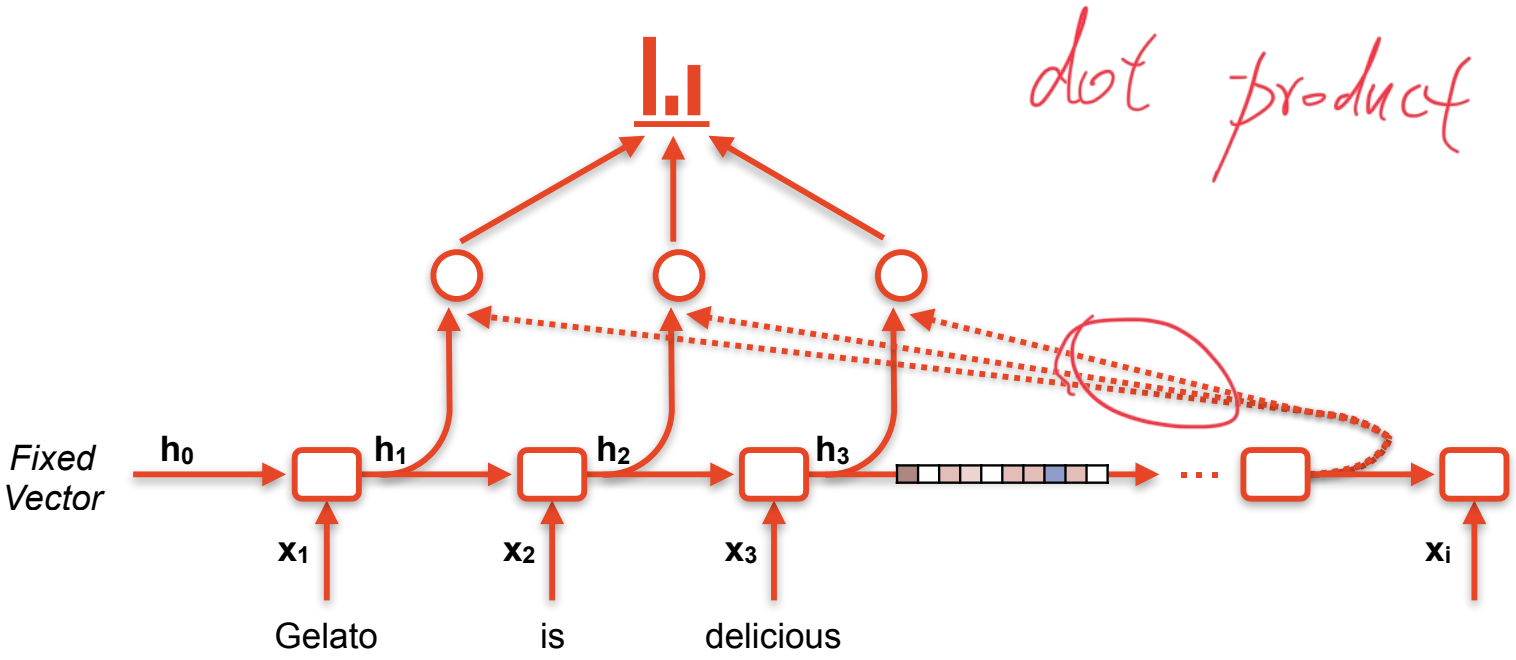
COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

We will solve this with ‘attention’

Give each output step
a representation of the input
that is most useful for that output decision



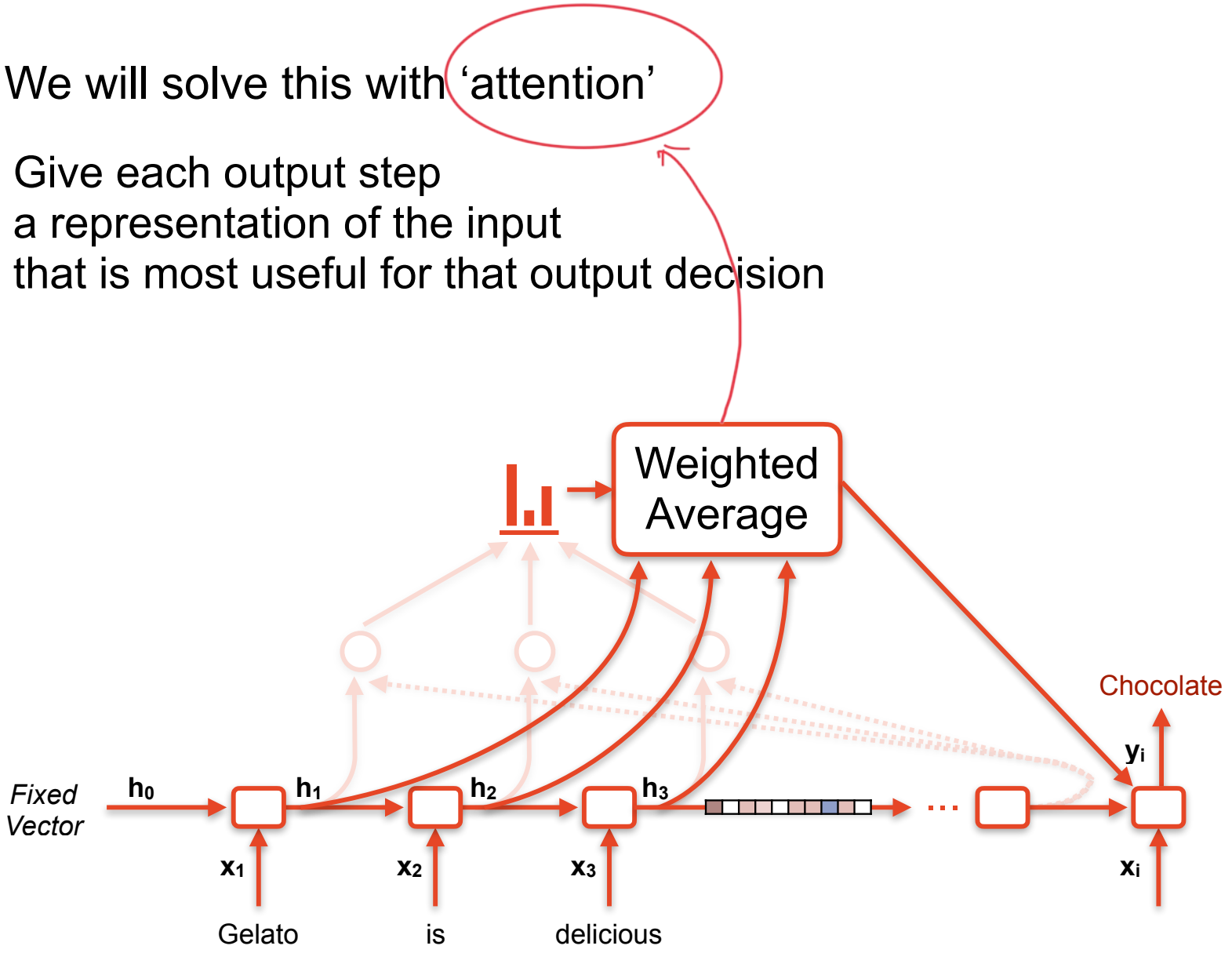
COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

We will solve this with 'attention'

Give each output step
a representation of the input
that is most useful for that output decision

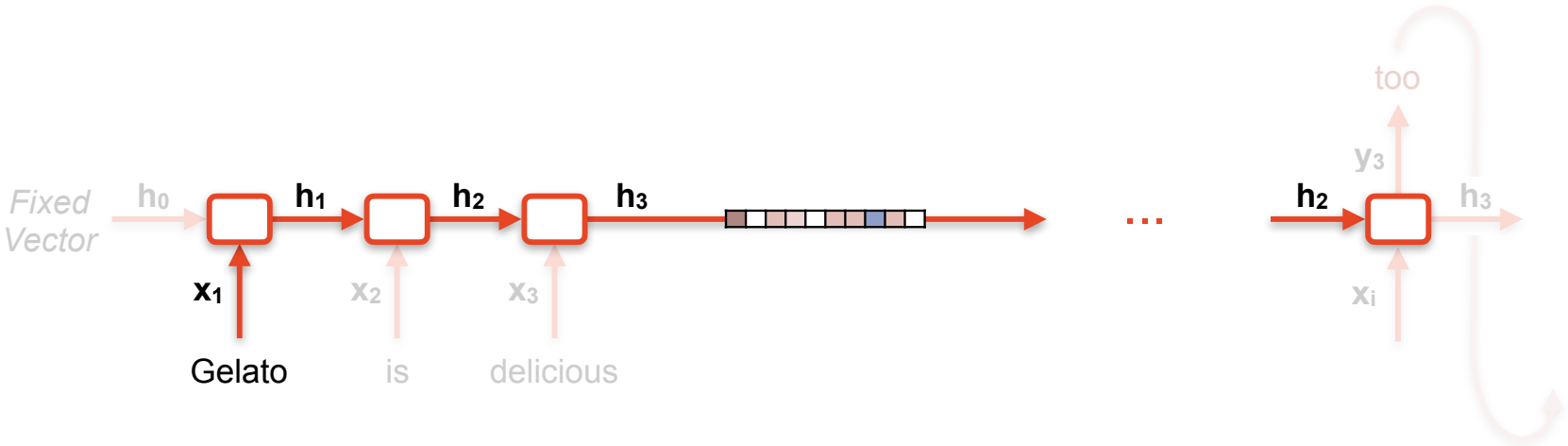


COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

This resolves the bottleneck problem without introducing many extra parameters

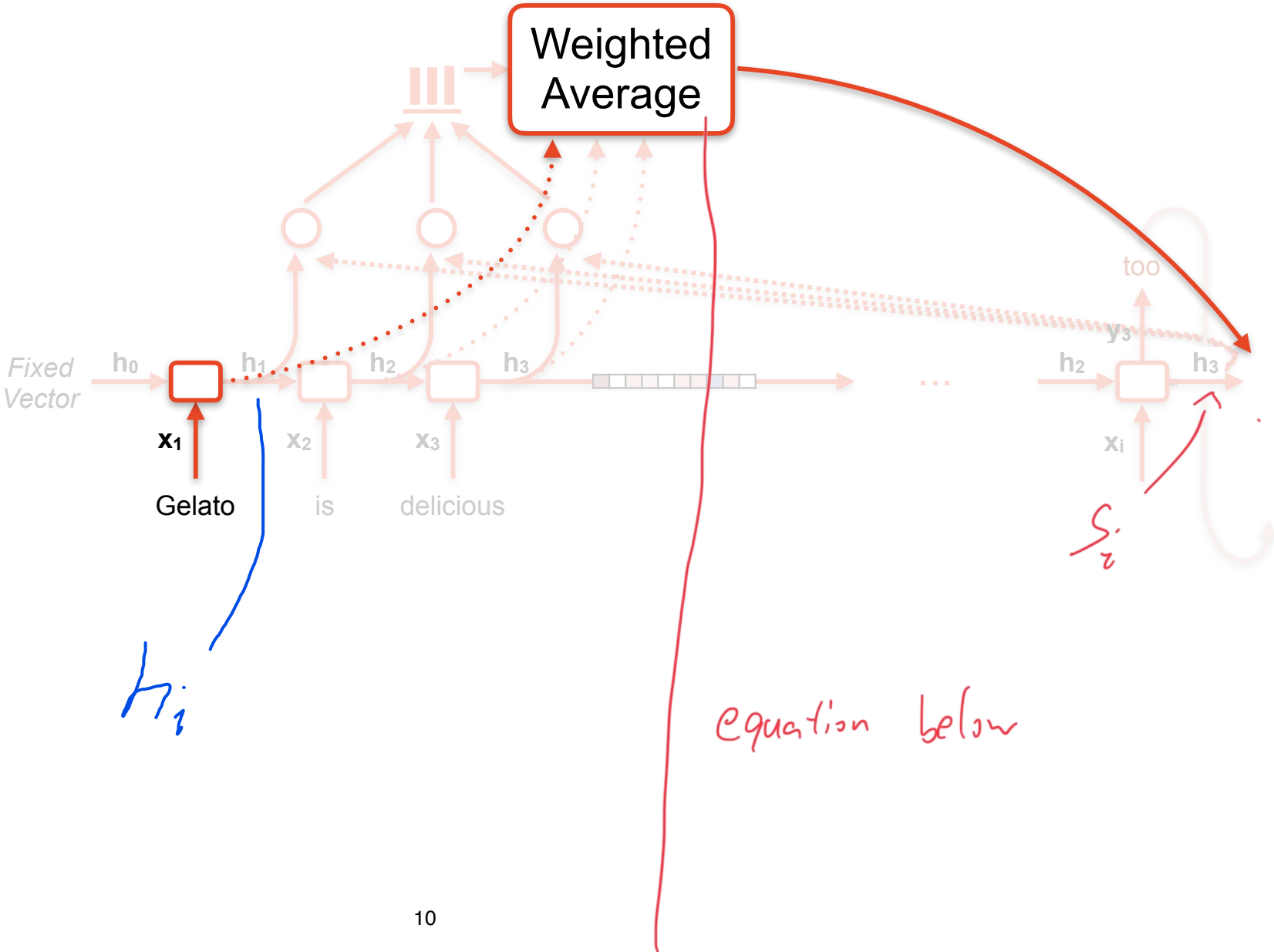


COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

This resolves the bottleneck problem without introducing many extra parameters



COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

We will solve this with ‘attention’

Give each output step
a representation of the input
that is most useful for that output decision

Hidden vectors from input $\mathbf{h_1}, \mathbf{h_2}, \mathbf{h_3} \ldots \mathbf{h_N} \in \mathbb{R}^{d_h}$

Hidden vector for current output step $\mathbf{s_i} \in \mathbb{R}^{d_h}$

Calculate attention scores $\mathbf{e_i} = [\mathbf{s_i}^\top \mathbf{h_1}, \mathbf{s_i}^\top \mathbf{h_2}, \mathbf{s_i}^\top \mathbf{h_3}, \ldots \mathbf{s_i}^\top \mathbf{h_N}] \in \mathbb{R}^N$

This is dot product attention

Normalise s $\ldots \text{ax}(\mathbf{e_i}) \in \mathbb{R}^N$

$\alpha_i = \text{softmax}(e_i) \in \mathbb{R}^N$

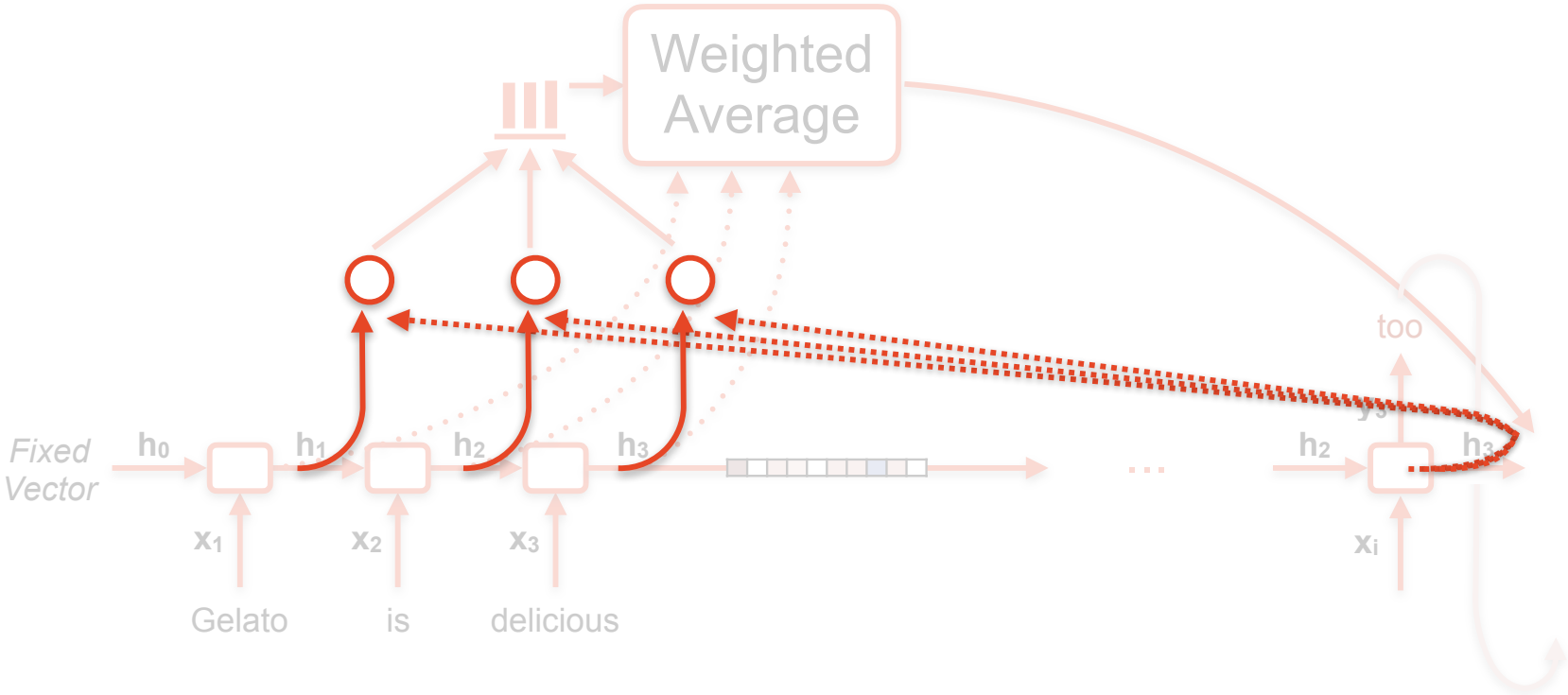
Calculate weighted average $\mathbf{a_i} = \sum_{j=1}^{N} \alpha_{i,j} \mathbf{h_j} \in \mathbb{R}^{d_h}$

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

This resolves the bottleneck problem without introducing many extra parameters



COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

There are many forms of attention

Dot product attention	$\mathbf{e} = \mathbf{s}^\top \mathbf{h}$	s, h must have same size
Scaled dot product attention	$\mathbf{e} = \frac{\mathbf{s}^\top \mathbf{h}}{\sqrt{d_h}}$	Motivated by statistical properties of dot products
Multiplicative attention / Bilinear attention	$\mathbf{e} = \mathbf{s}^\top \mathbf{W} \mathbf{h}$	s and h can be different sizes
Reduced-rank multiplicative attention	$\mathbf{e} = \mathbf{s}^\top (\mathbf{U}^\top \mathbf{V}) \mathbf{h}$ $= (\mathbf{s}\mathbf{U})^\top (\mathbf{V}\mathbf{h})$	Can improve efficiency
Additive attention / Feedforward attention	$\mathbf{e} = \mathbf{b} \tanh(\mathbf{W}_1 \mathbf{h} + \mathbf{W}_2 \mathbf{s})$	Powerful / Flexible

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

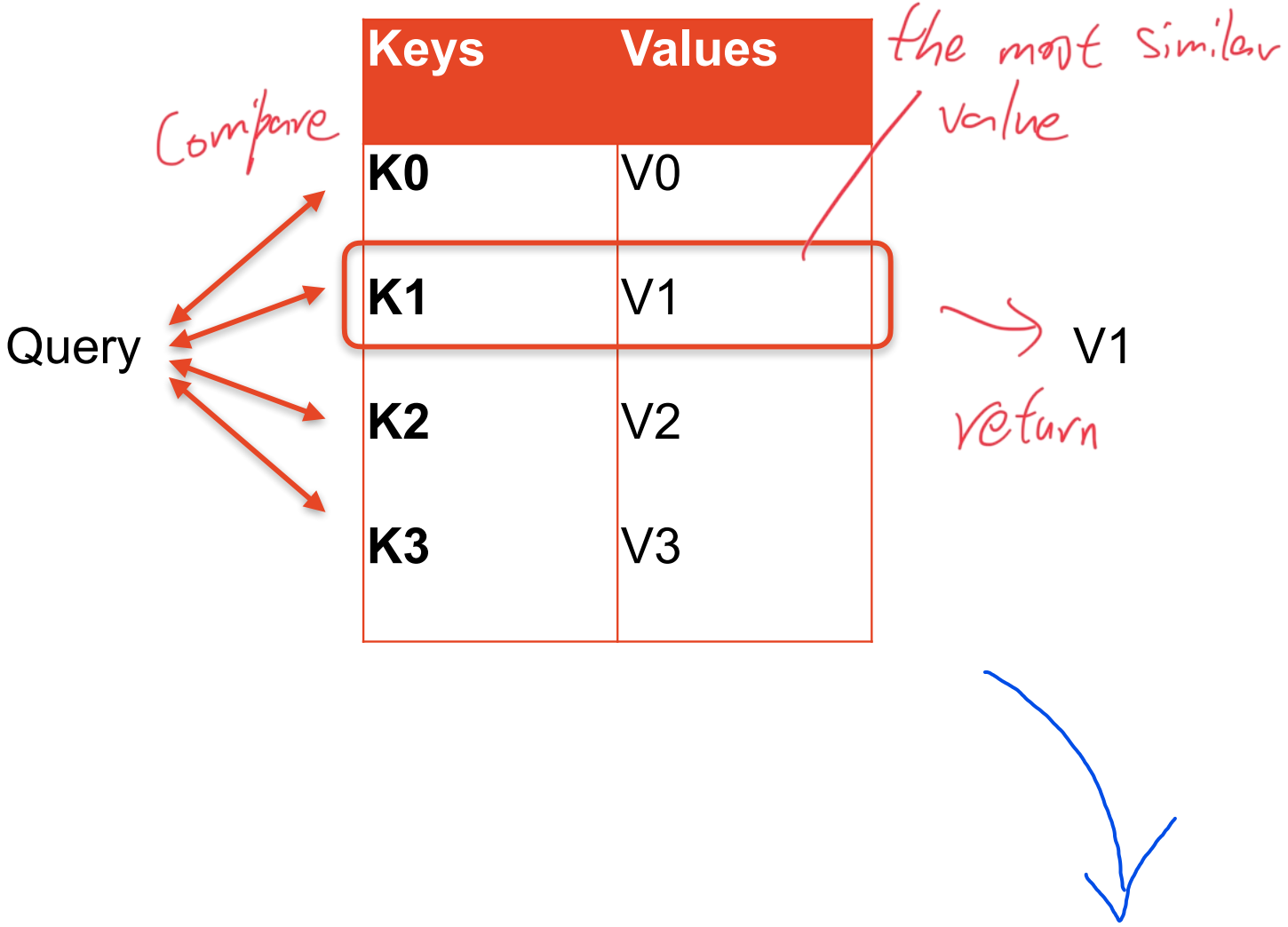
Another way of thinking about attention - looking up values

Consider a lookup table:

One Example

Input

Output

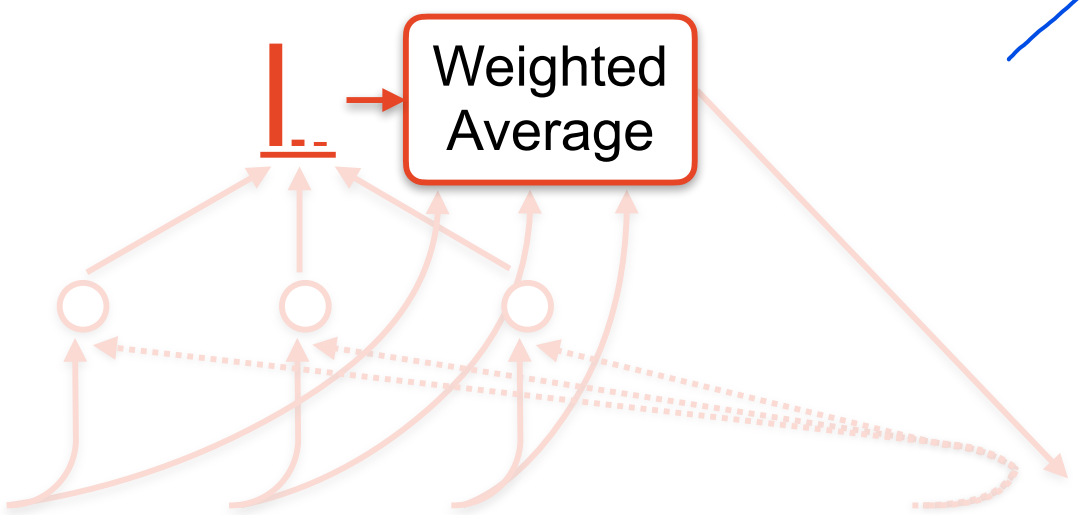


COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Another way of thinking about attention - looking up values

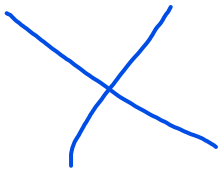


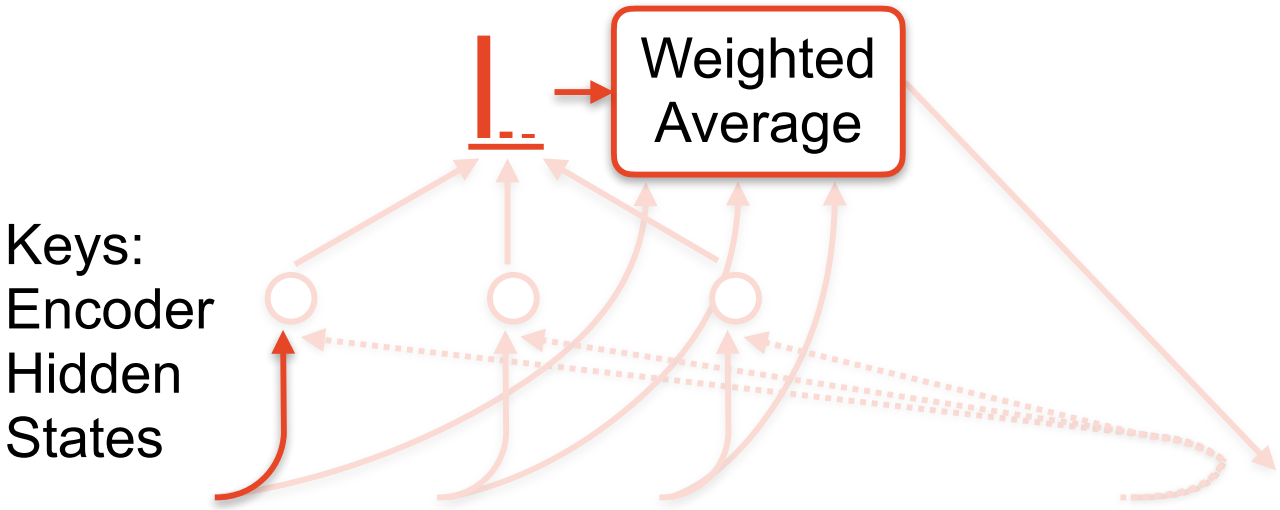
COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Another way of thinking about attention - looking up values



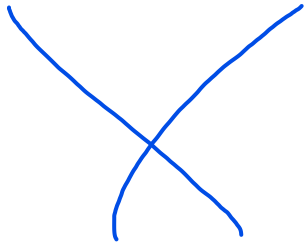


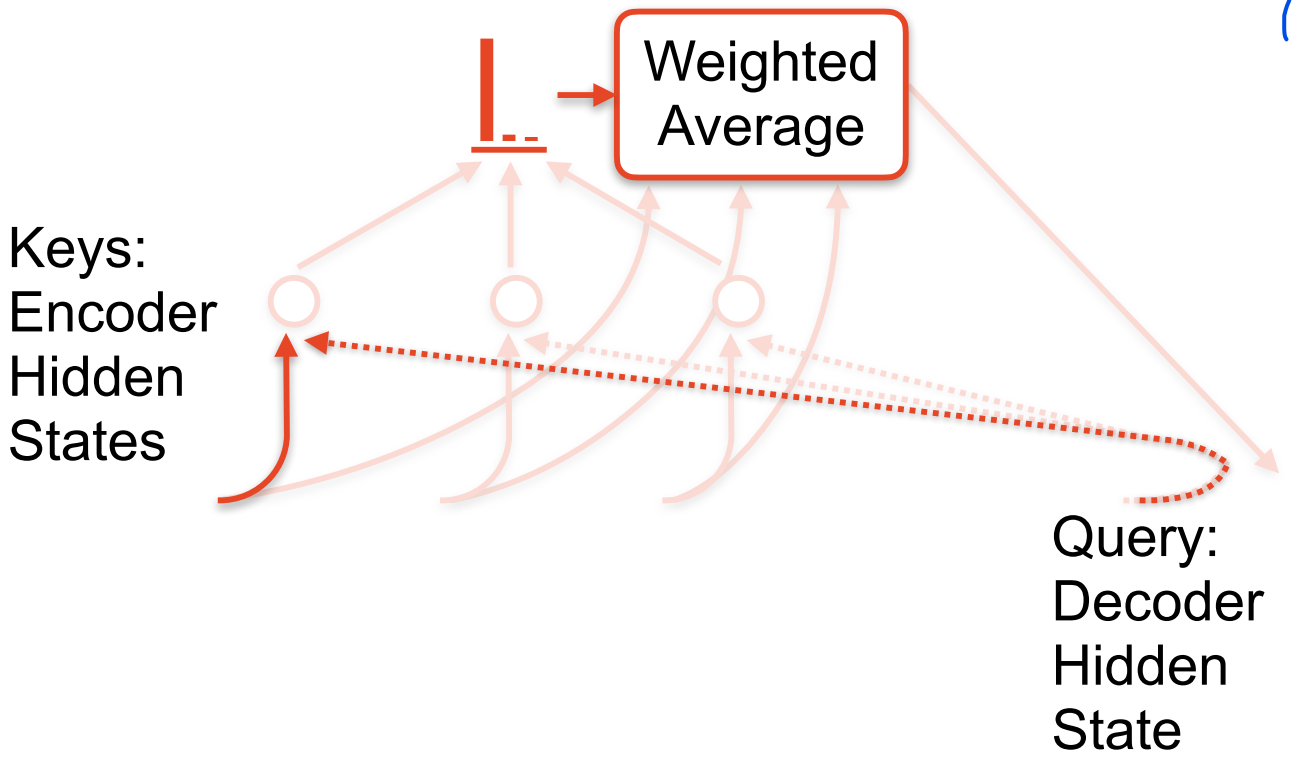
COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Another way of thinking about attention - looking up values



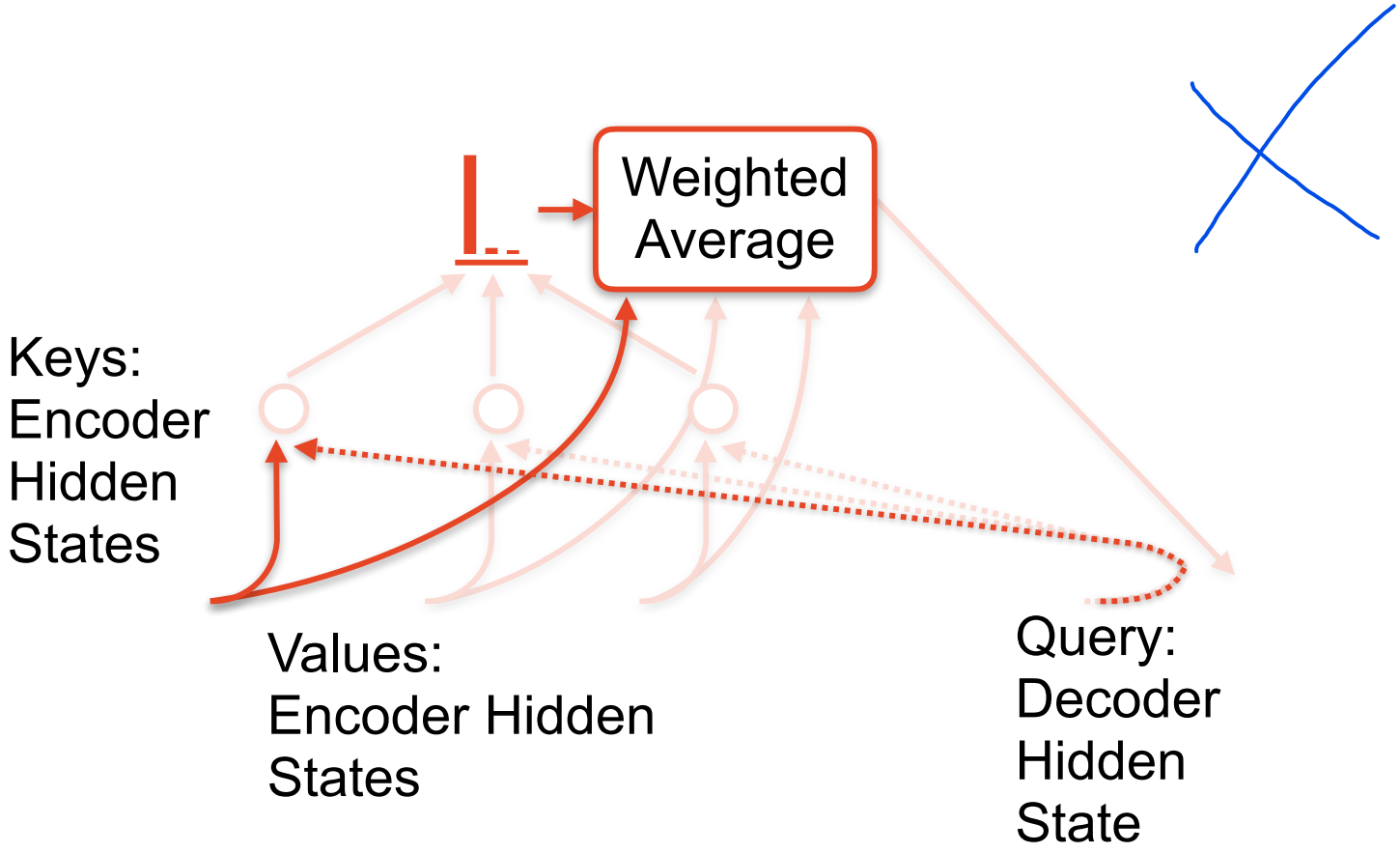


COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Another way of thinking about attention - looking up values

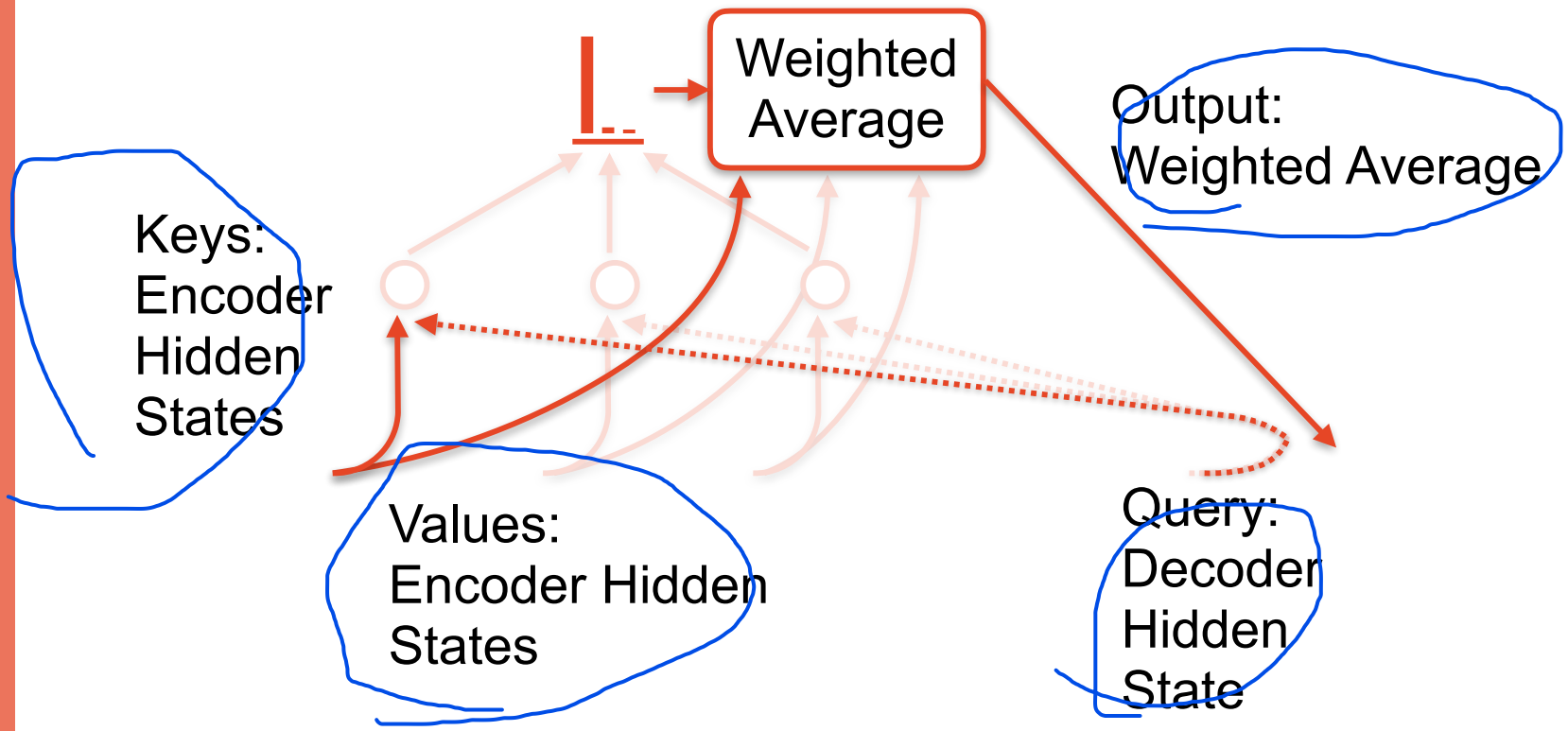


COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Another way of thinking about attention - looking up values



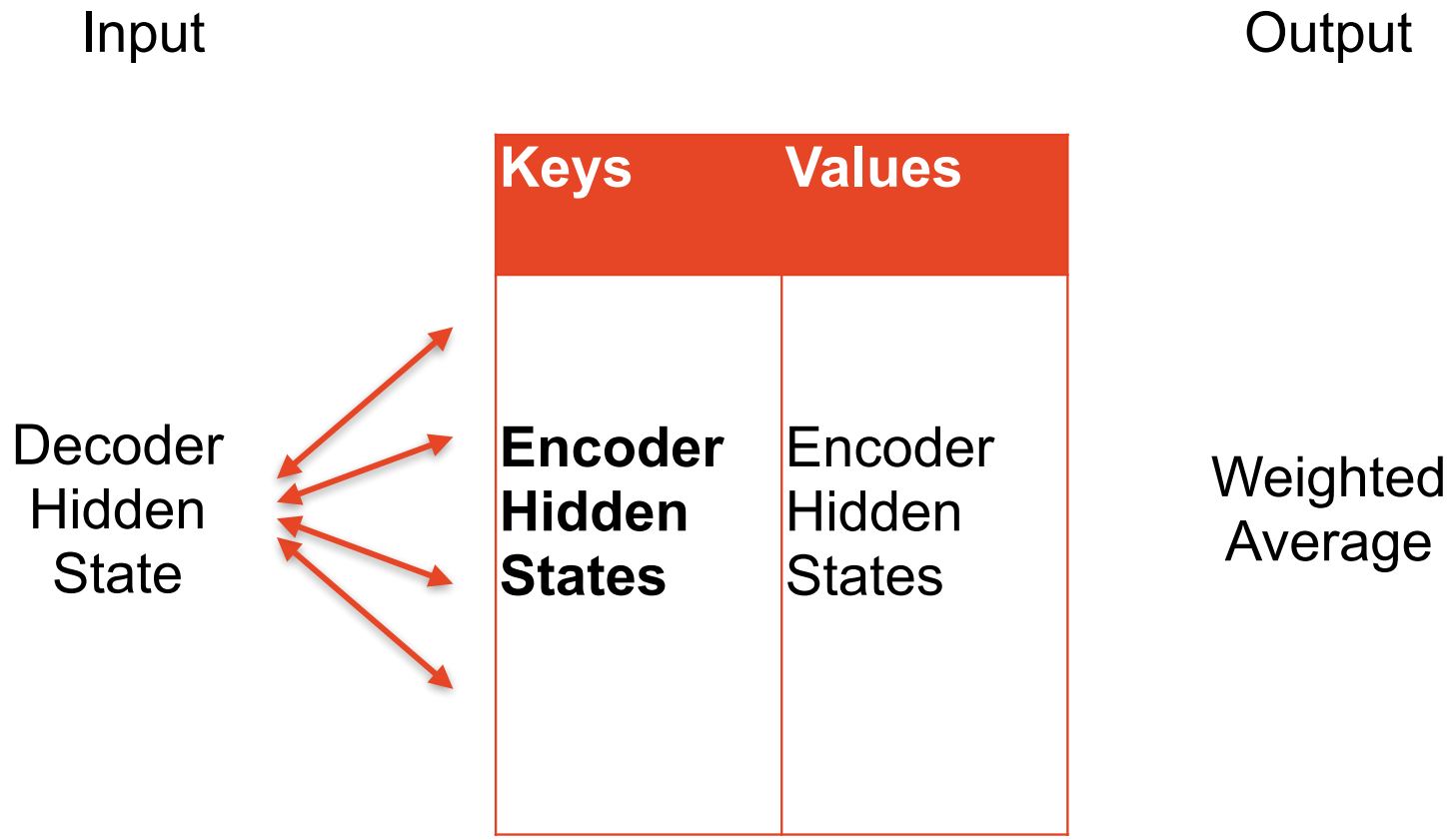
COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Another way of thinking about attention - looking up values

Consider a lookup table:



COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

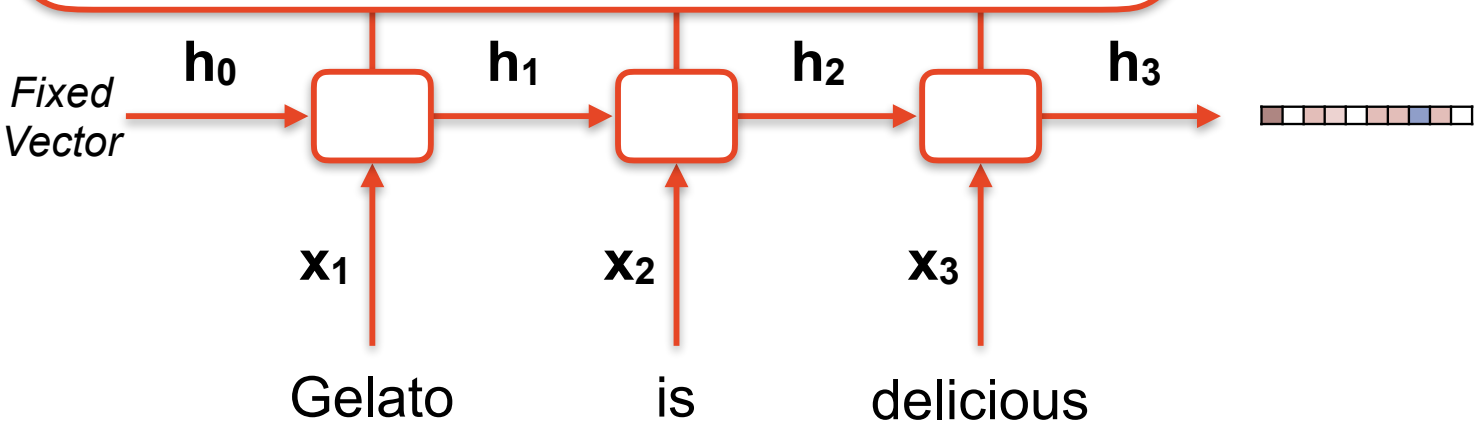
Do we need RNNs?

In lecture 3:
"How do we handle variable length input?"
"Recurrent Neural Networks (RNNs) address this issue with **context-dependent representations**"

hidden layer

In lecture 6, attention:
"Give each output step
a **representation of the input**
that is most useful for that output decision"

average weight

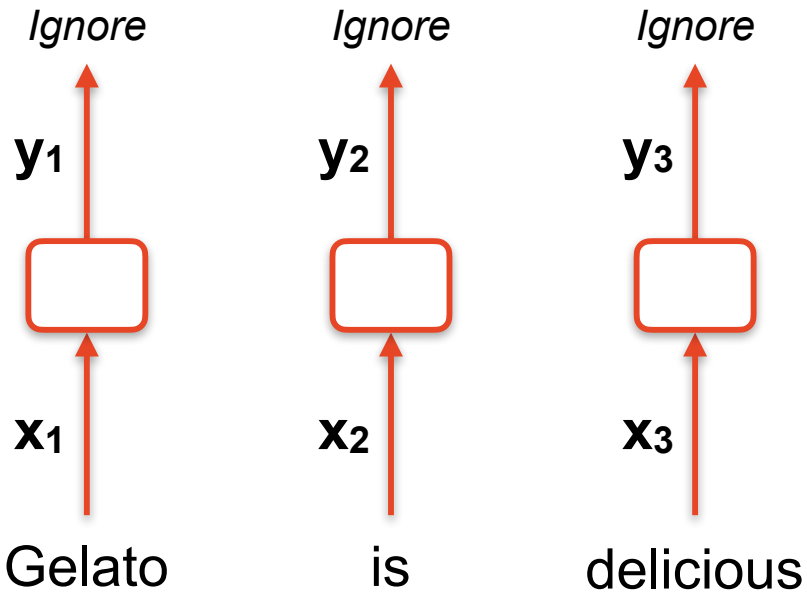


COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Cut the recurrent edges and use attention instead

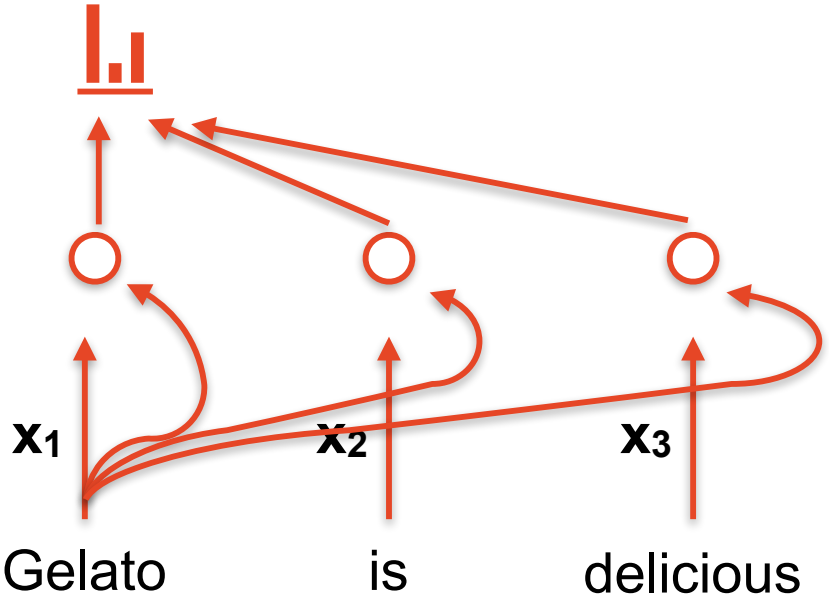


COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Cut the recurrent edges and use attention instead

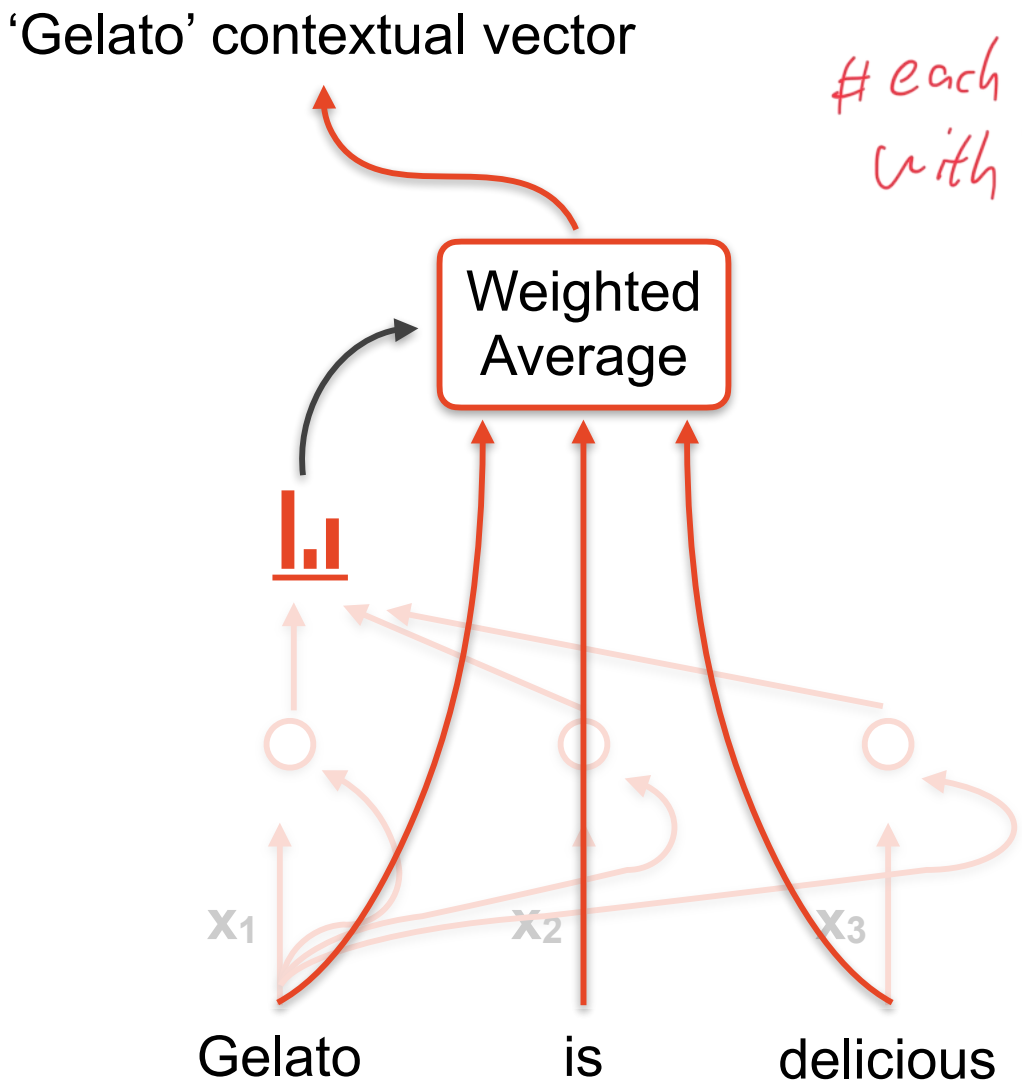


COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Cut the recurrent edges and use attention instead



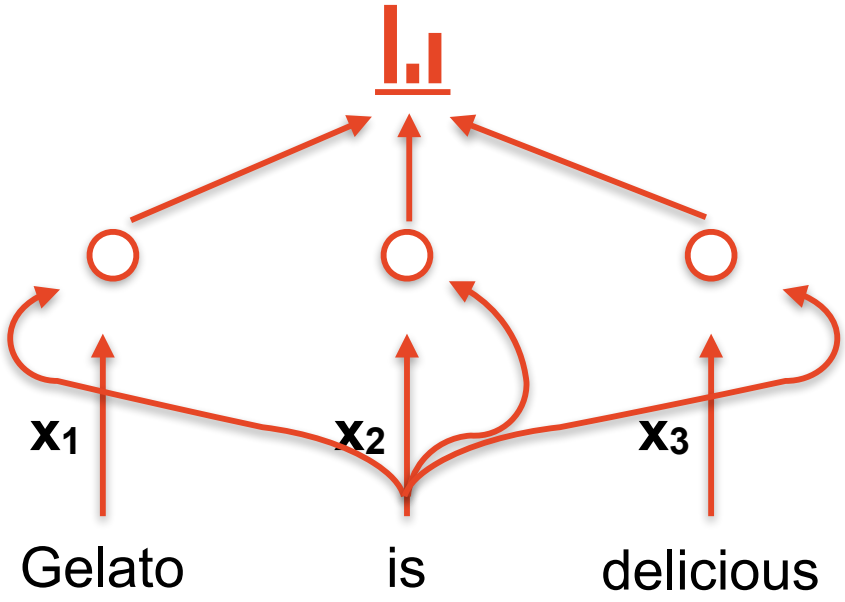
each word compare with 'Gelato'

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Cut the recurrent edges and use attention instead

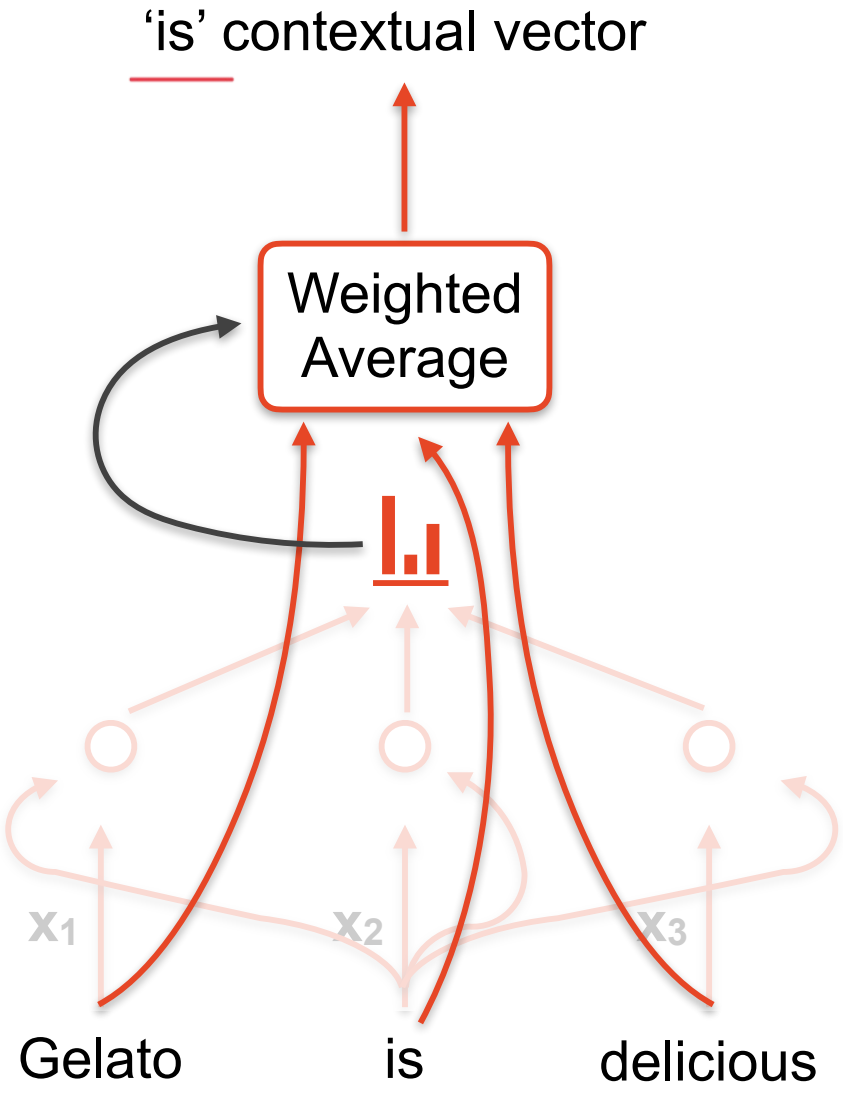


COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Cut the recurrent edges and use attention instead

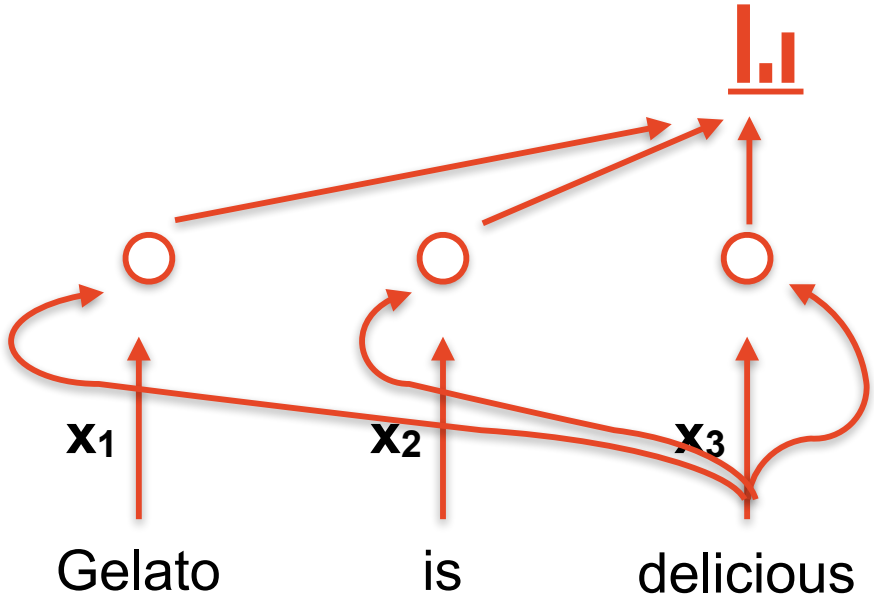


COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Cut the recurrent edges and use attention instead

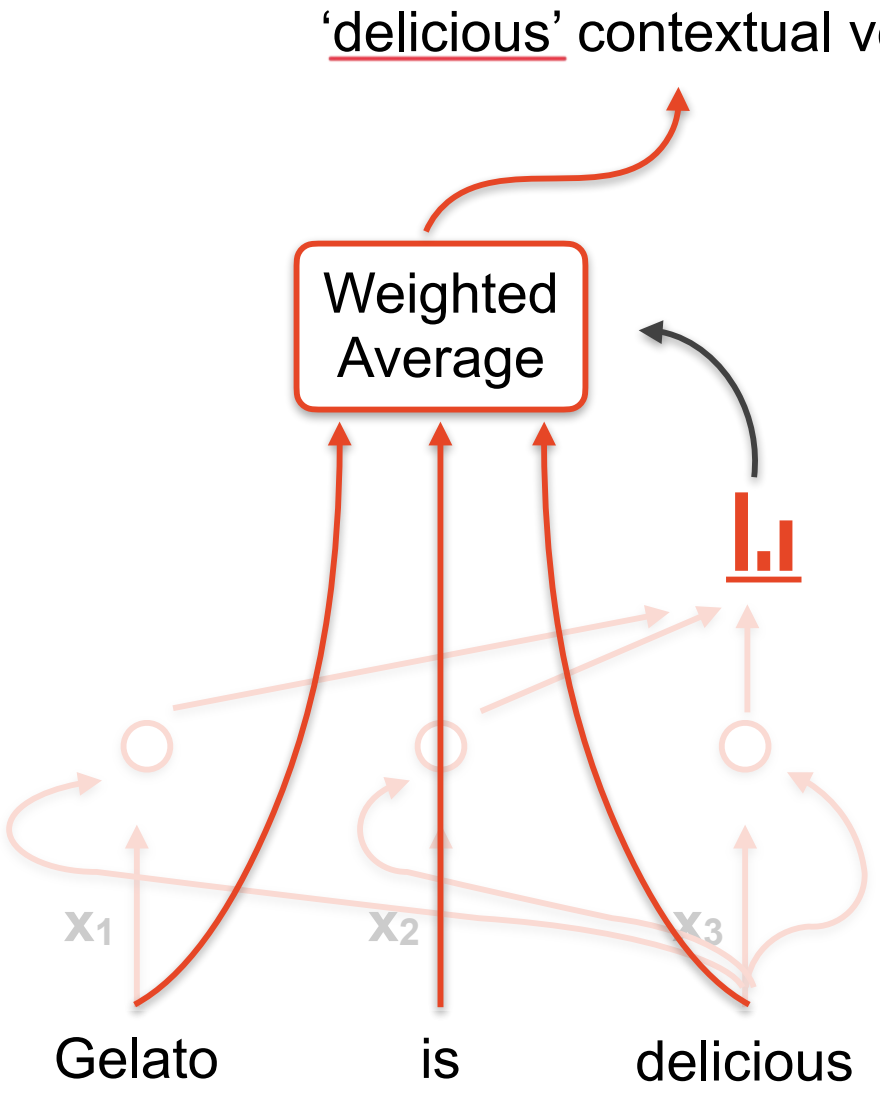


COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Cut the recurrent edges and use attention instead

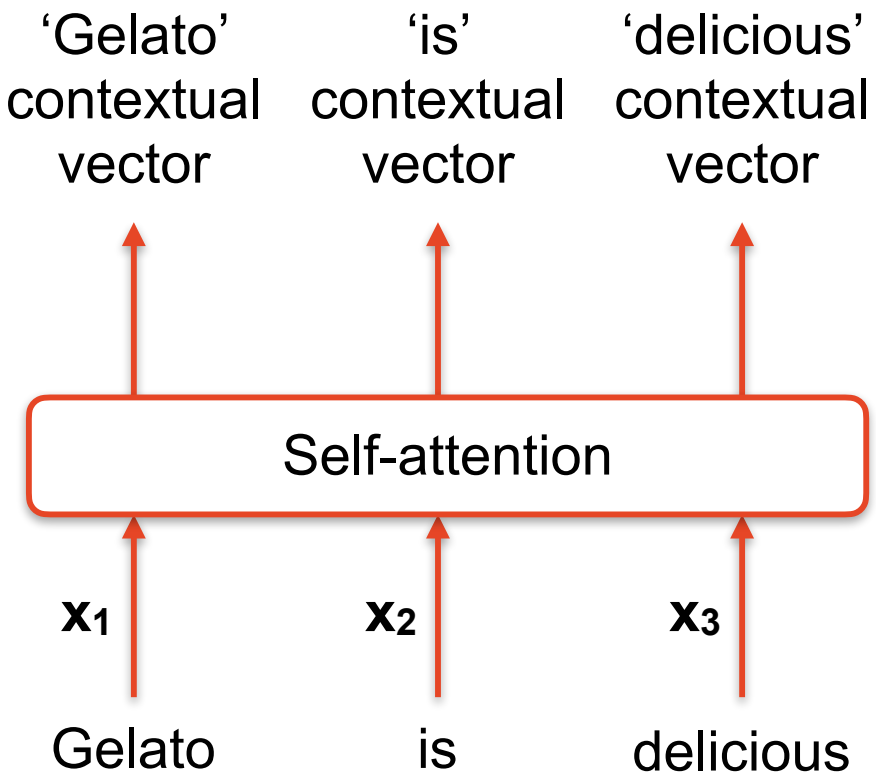


COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Self-attention gives contextual representations of input

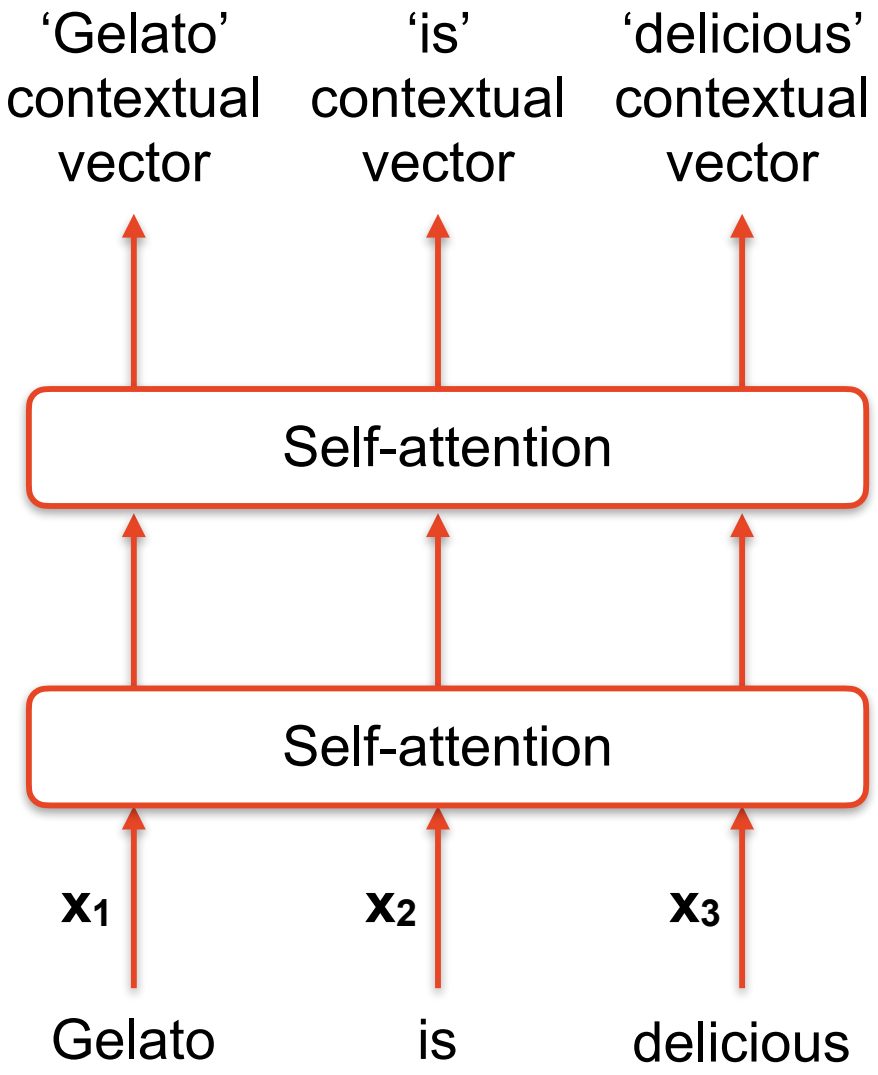


COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Self-attention gives contextual representations of input



COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Using equations, here is what we have done

self attention

Initial word vectors $\mathbf{x_i} \in \mathbb{R}^d$

current word

Do dot product to get attention scores $e_{ij} = \mathbf{x_i}^\top \mathbf{x_j}$

$\mathbf{x_i}$ is the query

$\mathbf{x_j}$ is the key

Softmax to get a distribution $\alpha_{ij} = \text{softmax}(e_{ij})$

all words include self

Weighted average $\mathbf{o_i} = \sum_j \alpha_{ij} \mathbf{x}_j$

$\mathbf{x_j}$ is the value

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Using equations, here is what we have done

Initial word vectors

$$\mathbf{x}_i \in \mathbb{R}^d$$

Transform into query, keys, and values

$$\mathbf{q_i} = Q\mathbf{x_i}$$
$$\mathbf{k_i} = K\mathbf{x_i}$$
$$\mathbf{v_i} = V\mathbf{x_i}$$

Q, K, and V are learned matrices

Do dot product to get attention scores

$$e_{ij} = \mathbf{q_i}^\top \mathbf{k_j}$$

Softmax to get a distribution

$$\alpha_{ij} = \text{softmax}(e_{ij})$$

Weighted average

$$\mathbf{o_i} = \sum_j \alpha_{ij}\mathbf{v_j}$$

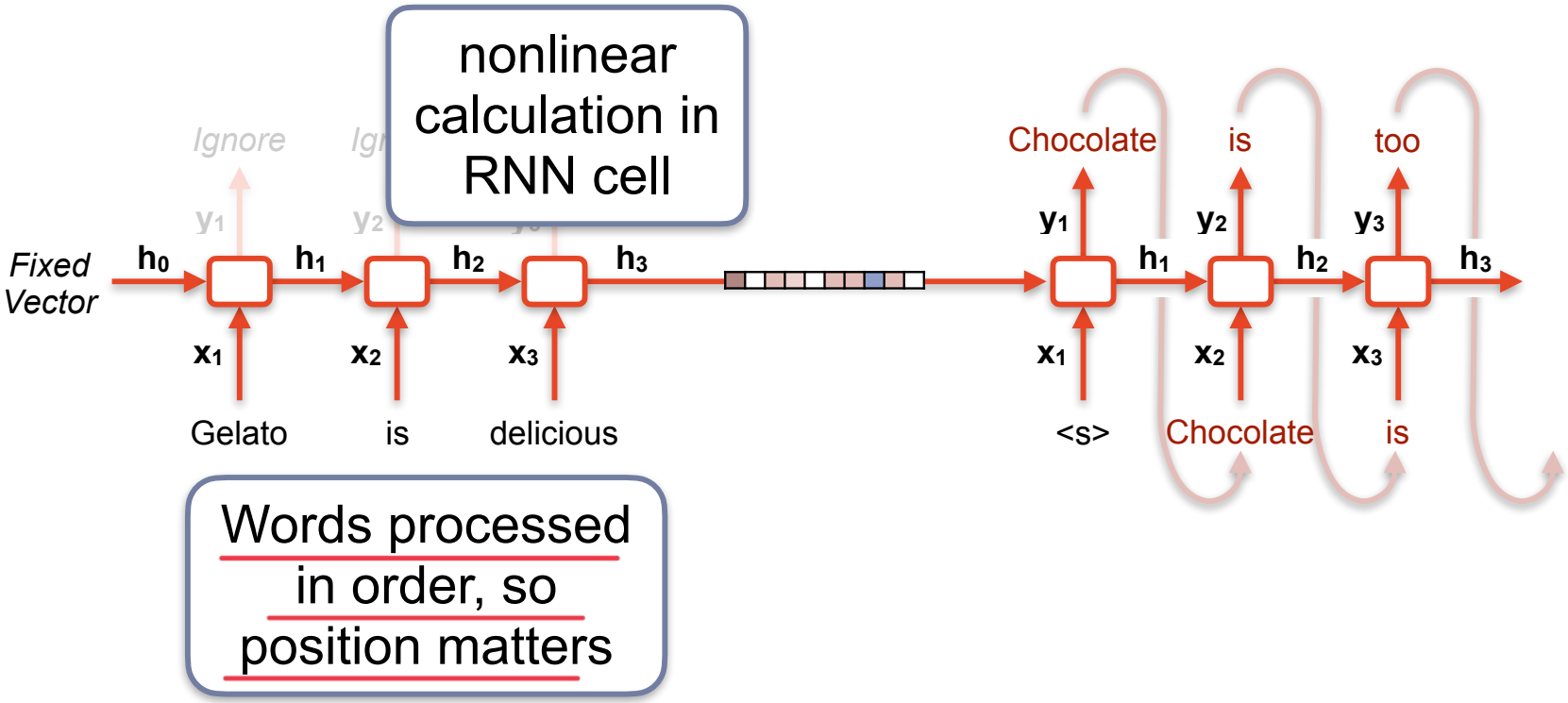
(above is linear)

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

To recover all benefits of the RNN we need nonlinearities and position representation

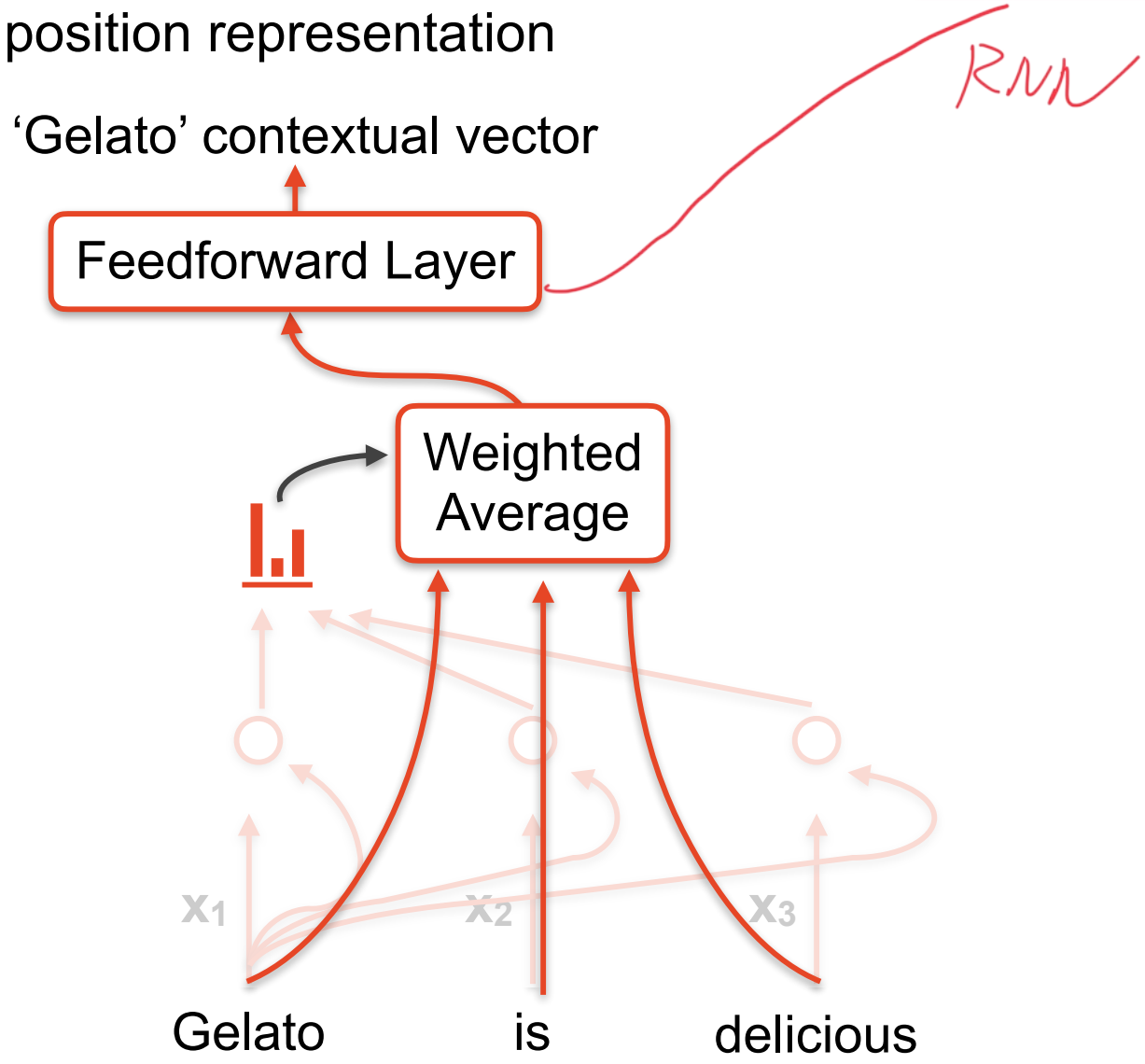


COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

To recover all benefits of the RNN we need **nonlinearities** and position representation



COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

To recover all benefits of the RNN we need **nonlinearities** and position representation

Initial word vectors $\mathbf{x}_i \in \mathbb{R}^d$

Transform into query, keys, and values

$$\mathbf{q_i} = Q\mathbf{x_i}$$
$$\mathbf{k_i} = K\mathbf{x_i}$$
$$\mathbf{v_i} = V\mathbf{x_i}$$

Do dot product to get attention scores $e_{ij} = \mathbf{q_i}^\top \mathbf{k_j}$

Softmax to get a distribution $\alpha_{ij} = \text{softmax}(e_{ij})$

Weighted average $\mathbf{t_i} = \sum_j \alpha_{ij} \mathbf{v_j}$

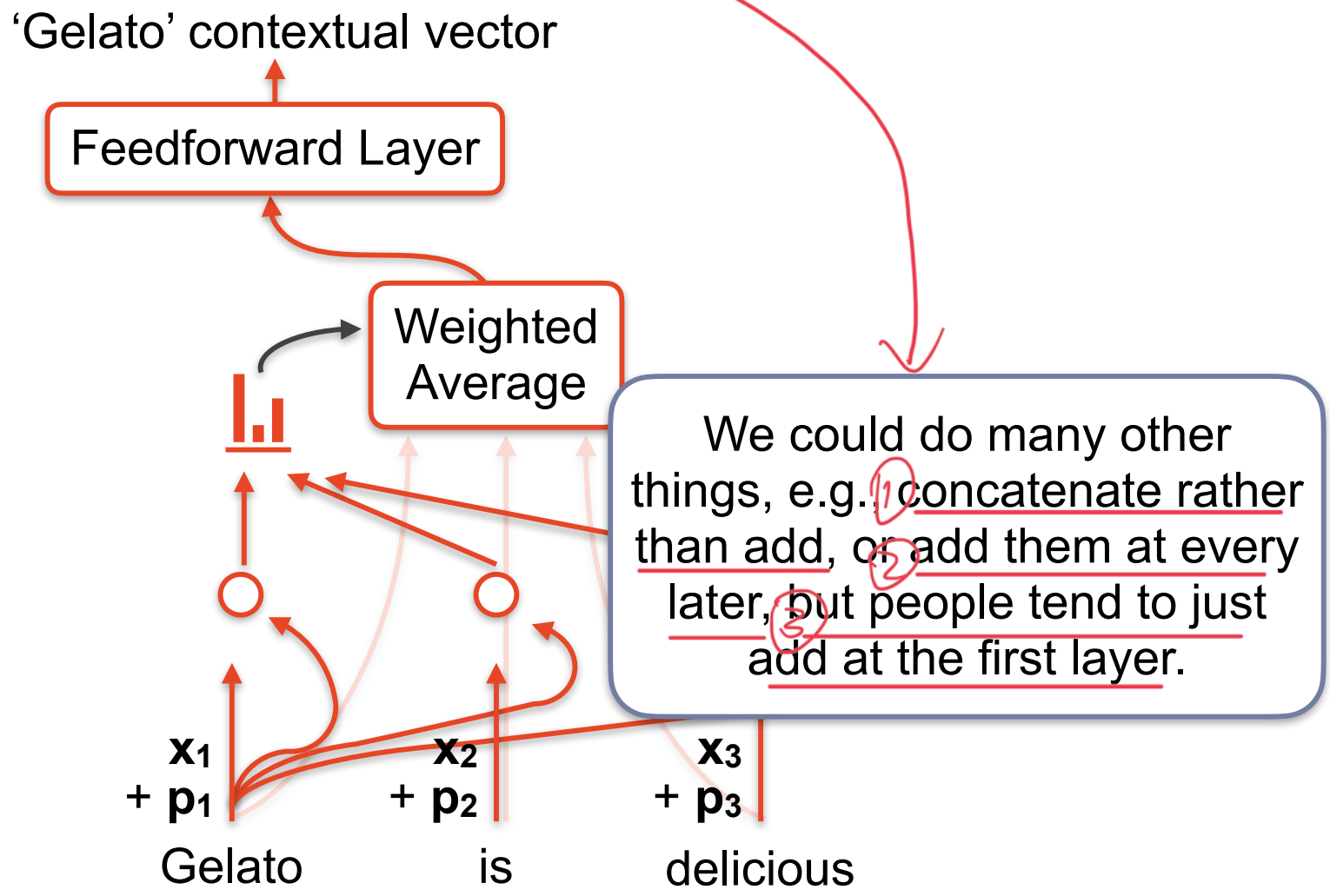
Feedforward layer $\mathbf{o_i} = W_2 \text{ReLU}(W_1 \mathbf{t_i} + \mathbf{b_1}) + \mathbf{b_2}$

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

To recover all benefits of the RNN we need nonlinearities and **position representation**



COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

To recover all benefits of the RNN we need nonlinearities and **position representation**

Original without position

Initial word vectors	$\mathbf{x}_i \in \mathbb{R}^d$
Transform into query, keys, and values	$\mathbf{q_i} = Q\tilde{\mathbf{x}}_\mathbf{i}$ $\mathbf{k_i} = K\tilde{\mathbf{x}}_\mathbf{i}$ $\mathbf{v_i} = V\tilde{\mathbf{x}}_\mathbf{i}$
Do dot product to get attention scores	$e_{ij} = \mathbf{q_i}^\top \mathbf{k_j}$
Softmax to get a distribution	$\alpha_{ij} = \text{softmax}(e_{ij})$
Weighted average	$\mathbf{t_i} = \sum_j \alpha_{ij}\mathbf{v_j}$
Feedforward layer	$\mathbf{o_i} = W_2\text{ReLU}(W_1\mathbf{t_i} + \mathbf{b_1}) + \mathbf{b_2}$

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

To recover all benefits of the RNN we need nonlinearities and **position representation**

Initial word vectors with positions

$$\tilde{\mathbf{x}}_i = \mathbf{x_i} + \mathbf{p_i}$$

Transform into query, keys, and values

$$\mathbf{q_i} = Q\tilde{\mathbf{x}}_\mathbf{i}$$

$$\mathbf{k_i} = K\tilde{\mathbf{x}}_\mathbf{i}$$

$$\mathbf{v_i} = V\tilde{\mathbf{x}}_\mathbf{i}$$

Do dot product to get attention scores

$$e_{ij} = \mathbf{q_i}^\top \mathbf{k_j}$$

Softmax to get a distribution

$$\alpha_{ij} = \text{softmax}(e_{ij})$$

Weighted average

$$\mathbf{t_i} = \sum_j \alpha_{ij}\mathbf{v_j}$$

Feedforward layer

$$\mathbf{o_i} = W_2\text{ReLU}(W_1\mathbf{t_i} + \mathbf{b_1}) + \mathbf{b_2}$$

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

To recover all benefits of the RNN we need nonlinearities and **position representation**

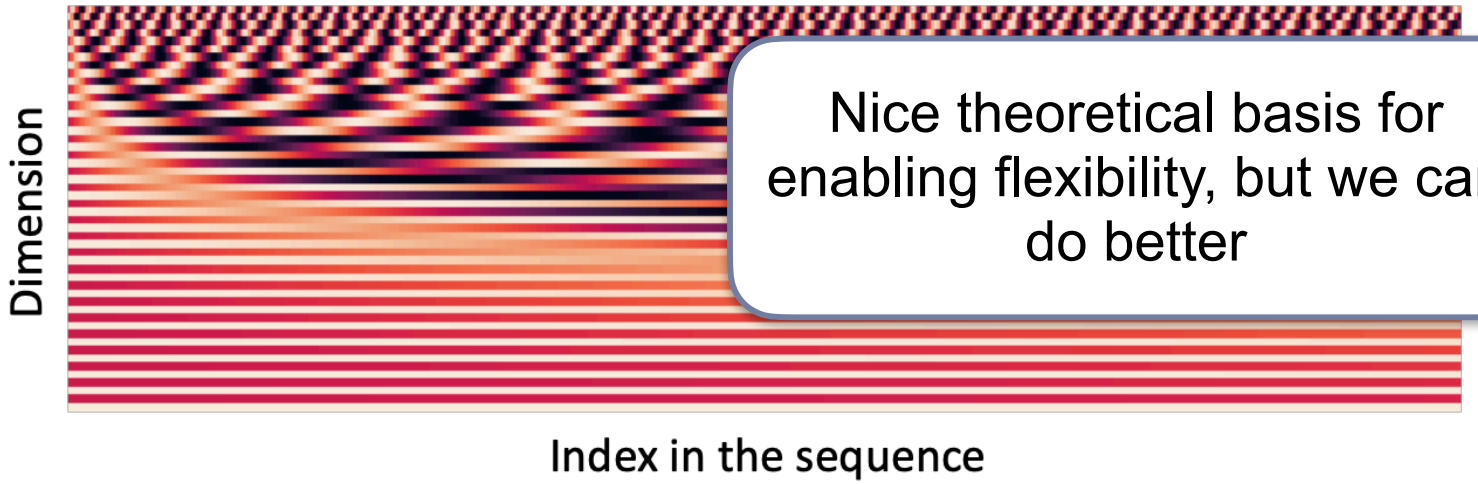
What is $\mathbf{p}_i$?

First idea:

$$\boldsymbol{p}_i = \begin{pmatrix} \sin(i/10000^{2*1/d}) \\ \cos(i/10000^{2*1/d}) \\ \vdots \\ \sin(i/10000^{2*\frac{d}{2}/d}) \\ \cos(i/10000^{2*\frac{d}{2}/d}) \end{pmatrix}$$

at top change quickly

at bottom change slowly



Nice theoretical basis for enabling flexibility, but we can do better

Stanford CS224 lectures

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

To recover all benefits of the RNN we need nonlinearities and **position representation**

What is $\mathbf{p}_i$?

Next idea:

1) Learned values - for each position, learn a vector

problem

Note: can't generalise to sequences longer than those seen in training!

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

To recover all benefits of the RNN we need nonlinearities and **position dependent representation**

Rotary Positional Embeddings

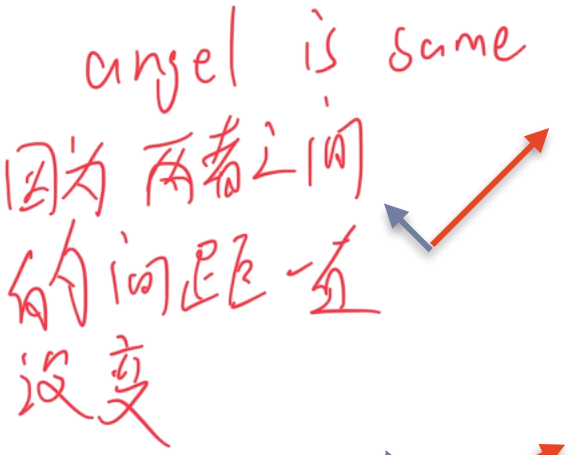
COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

To recover all benefits of the RNN we need nonlinearities and **position dependent representation**

Rotary Positional Embeddings



This chocolate is delicious

I think chocolate is delicious

Most people know chocolate is delicious

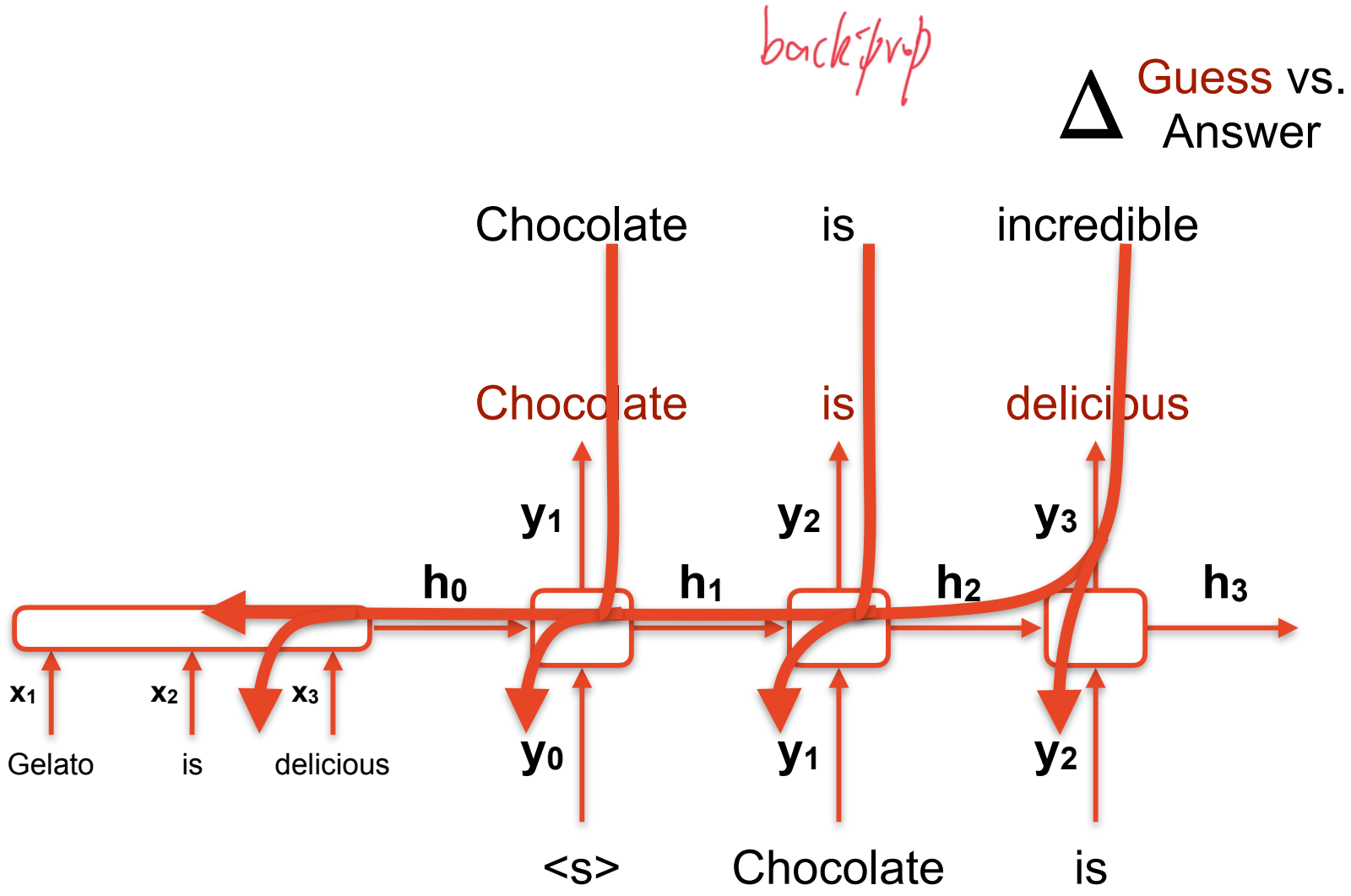
Cosine similarity remains the same for two words the same distance apart!

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Training needs to account for what is known at each step

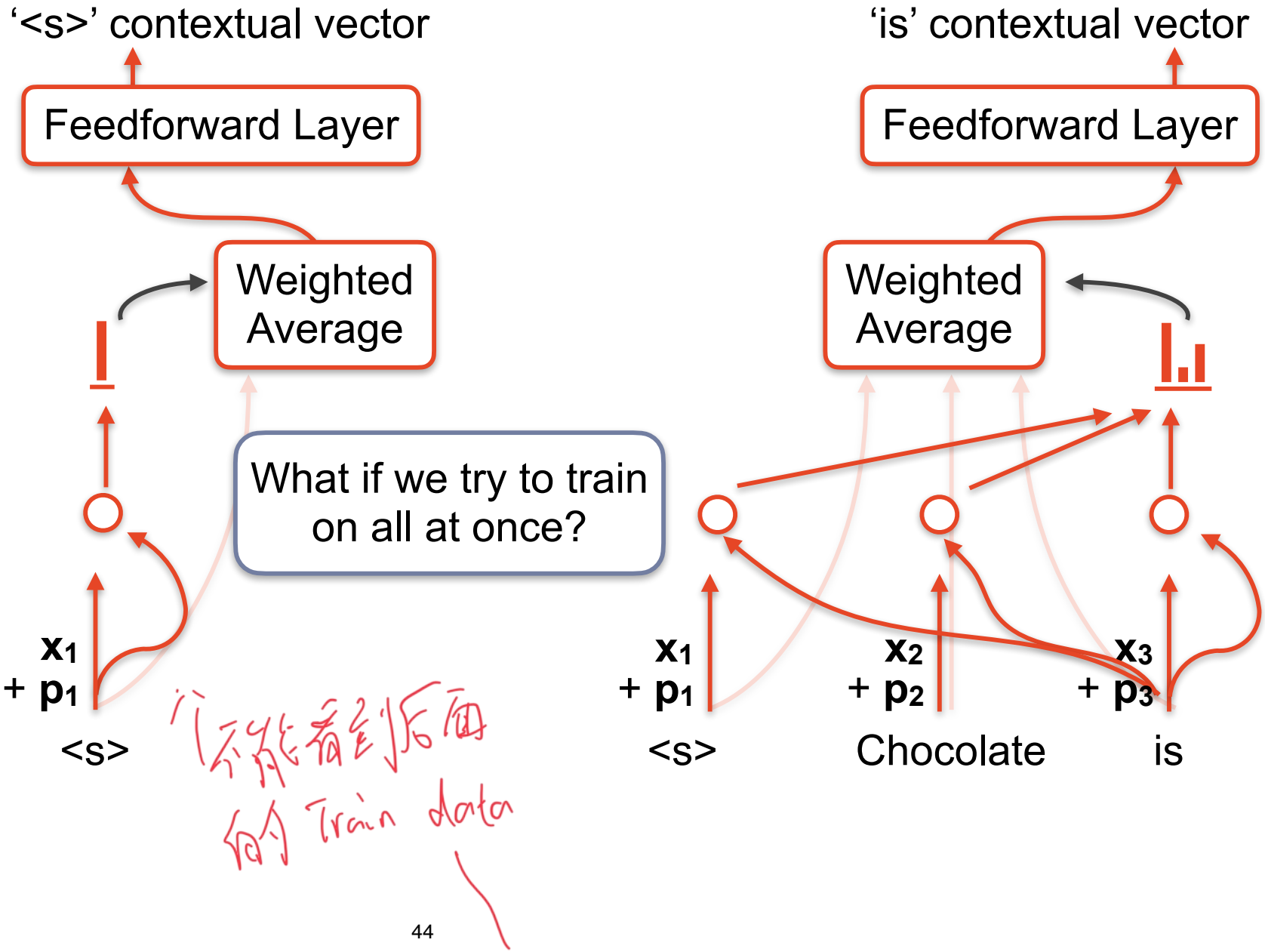


COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Training needs to account for what is known at each step

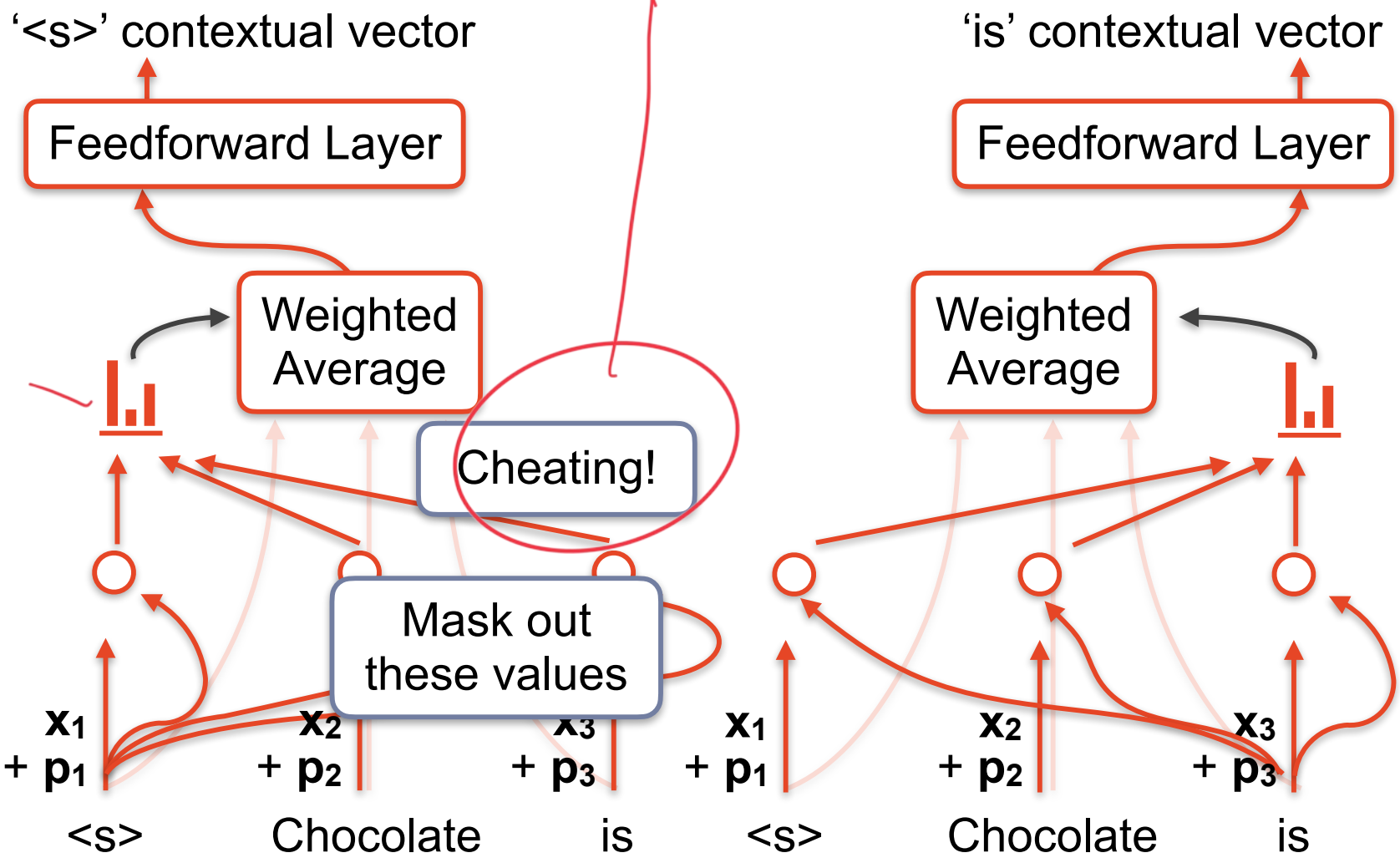


COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Training needs to account for what is known at each step



COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Training needs to account for what is known at each step

Initial word vectors with positions

$$\tilde{\mathbf{x}}_i = \mathbf{x_i} + \mathbf{p_i}$$

Transform into query, keys, and values

$$\mathbf{q_i} = Q\tilde{\mathbf{x}}_\mathbf{i}$$
$$\mathbf{k_i} = K\tilde{\mathbf{x}}_\mathbf{i}$$
$$\mathbf{v_i} = V\tilde{\mathbf{x}}_\mathbf{i}$$

Do dot product to get attention scores

$$e_{ij} = \mathbf{q_i}^\top \mathbf{k_j}$$

Training:

$$e_{ij} = \begin{cases} \mathbf{q_i}^\top \mathbf{k_j}, j \leq i \\ -\infty, j > i \end{cases}$$

Softmax to get a distribution

bar chart in Gradescope

$$\alpha_{ij} = \text{softmax}(e_{ij})$$

Weighted average

$$\mathbf{t_i} = \sum_j \alpha_{ij} \mathbf{v_j}$$

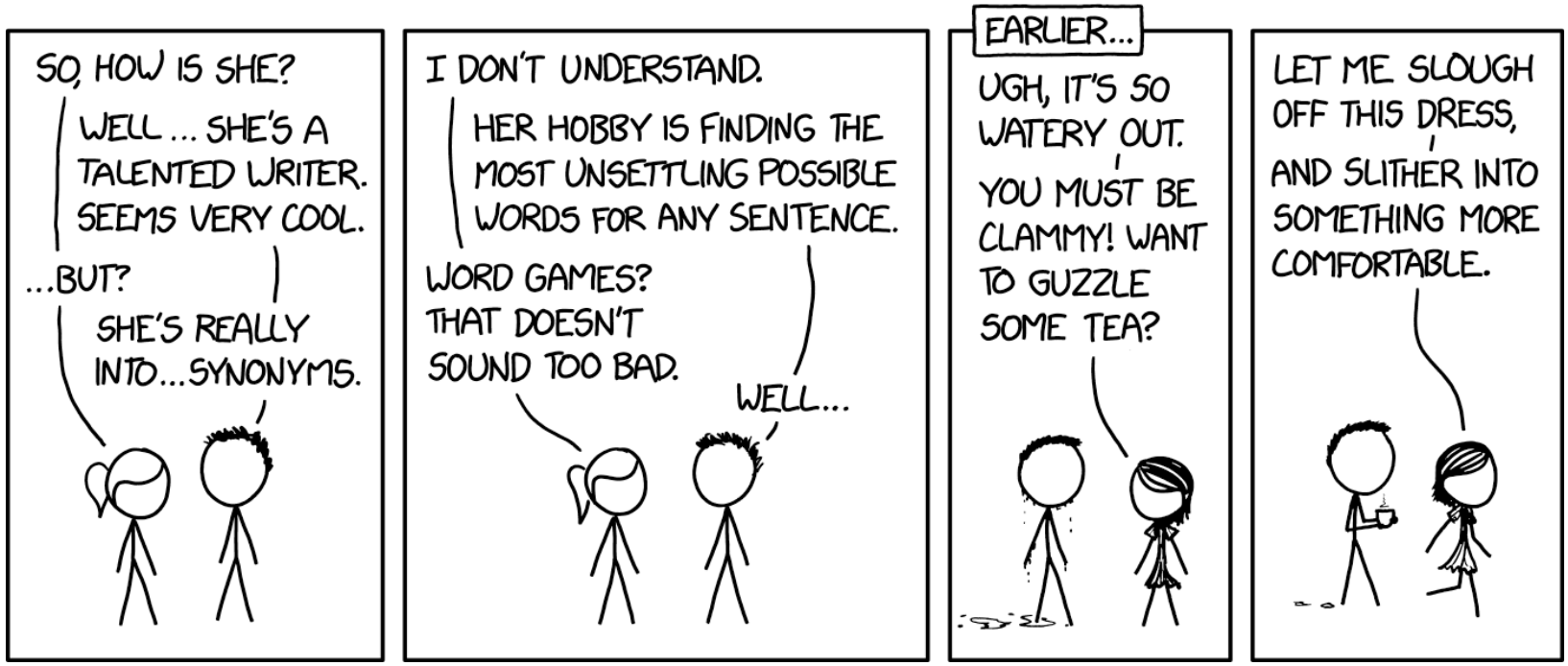
Feedforward layer

$$\mathbf{o_i} = W_2 \text{ReLU}(W_1 \mathbf{t_i} + \mathbf{b_1}) + \mathbf{b_2}$$

3 minute Break - stretch and visit Menti

menti.com
2520 1730

Synonym Date



[We need some grub to munch—I'll go slouch over to the kitchen.]

Source: https://xkcd.com/2352/

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Transformer

COMP 4446 / 5046
Lecture 7, 2025

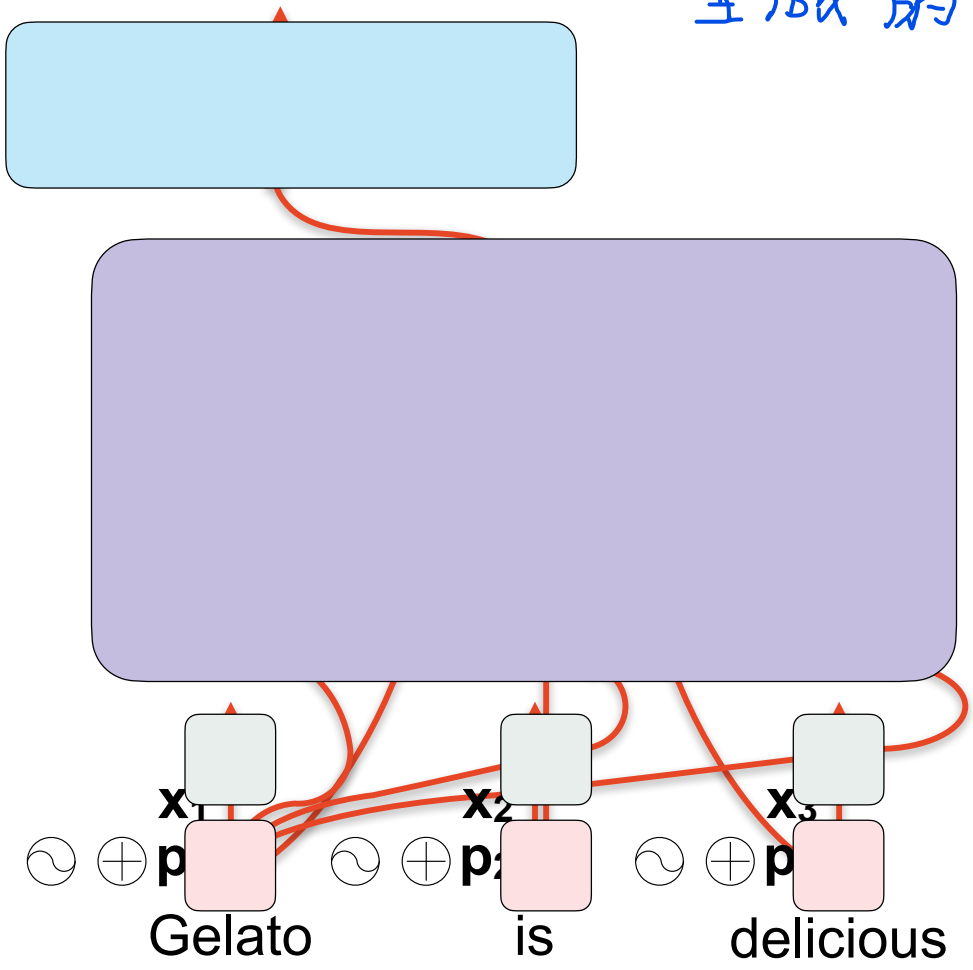
Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Going from Self-Attention to the Transformer

用box表示之前在self-attention里做的

‘Gelato’ contextual vector



Feed Forward

$$\max(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$$

⊕ Element-wise sum

Attention:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear transform

Look up vector

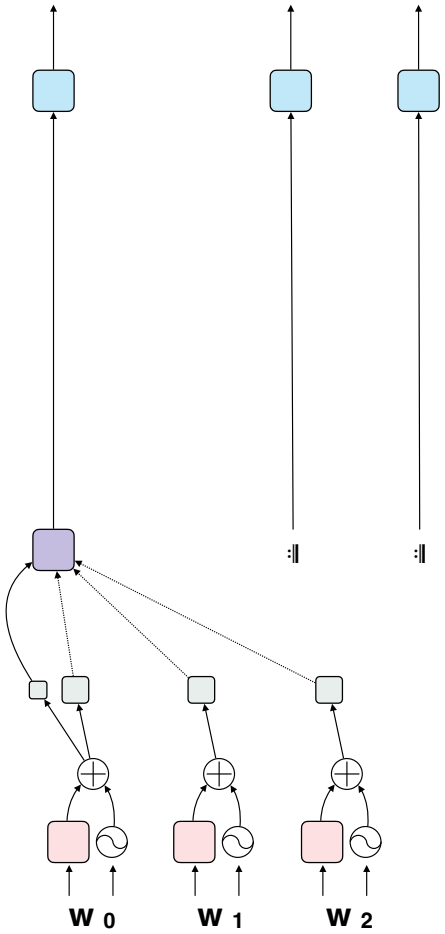
Position as vector

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Going from Self-Attention to the Transformer



Feed Forward

$$\max(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$$

⊕ Element-wise sum

Attention:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear transform

Look up vector

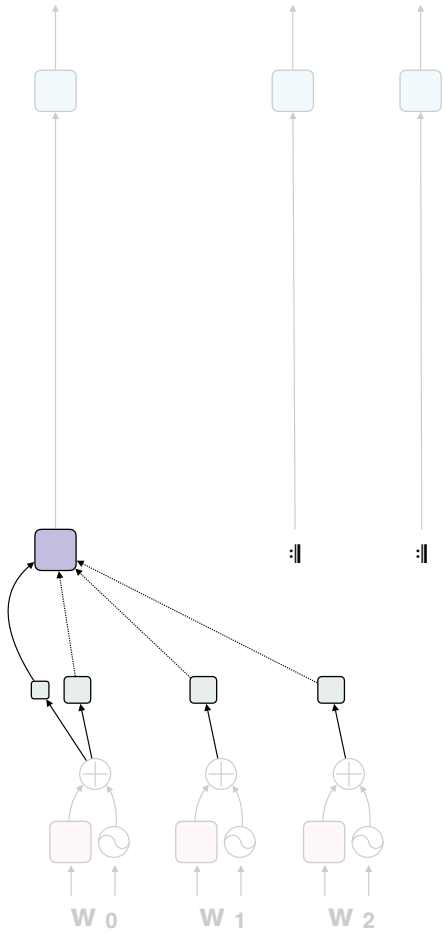
Position as vector

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Going from Self-Attention to the Transformer: Multi-Head Attention



Feed Forward

$$\max(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$$

⊕ Element-wise sum

Attention:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear transform

Look up vector

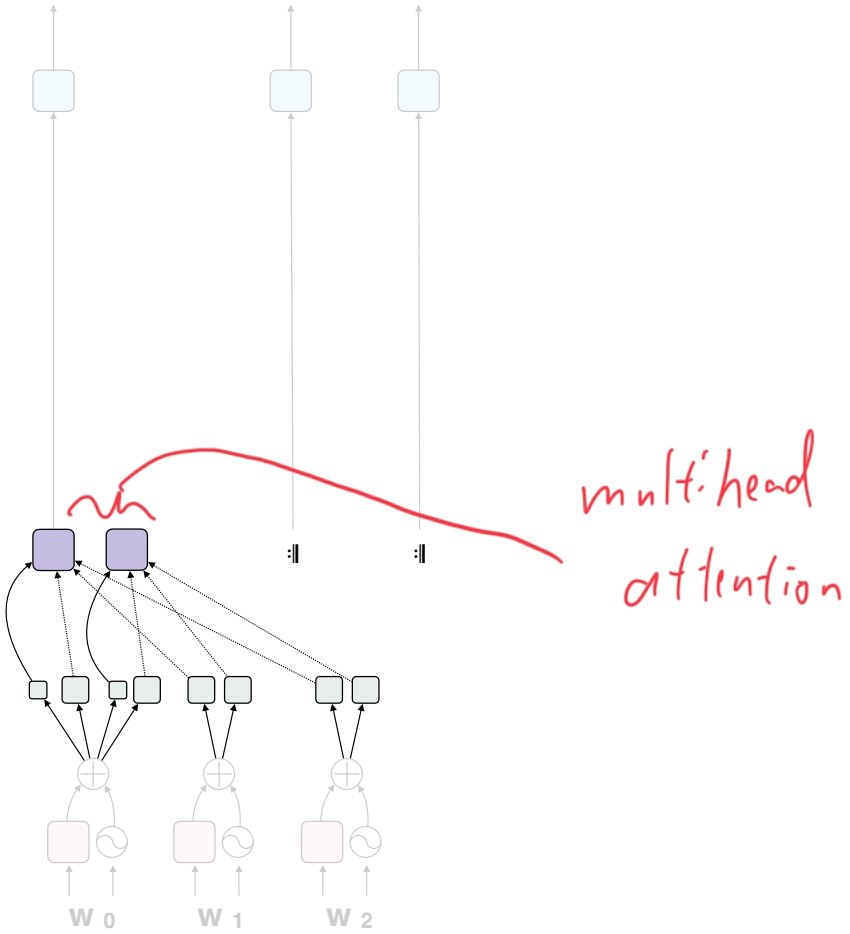
Position as vector

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Going from Self-Attention to the Transformer: Multi-Head Attention



Feed Forward

$$\max(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$$

⊕ Element-wise sum

Attention:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear transform

Look up vector

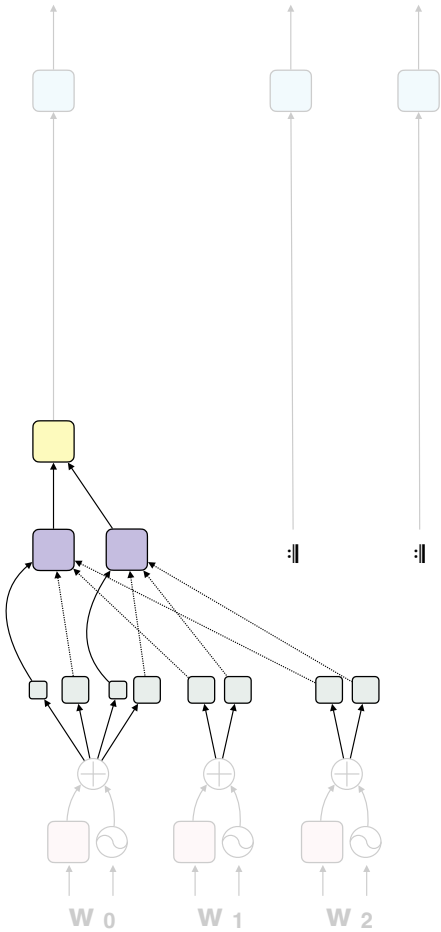
Position as vector

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Going from Self-Attention to the Transformer: Multi-Head Attention



Feed Forward

$$\max(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$$

⊕ Element-wise sum

Concatenate

Attention:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear transform

Look up vector

Position as vector

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Why do multi-head attention?

Attention is more than just a dot product here

Attention Head 1
(Entity focused)

Attention Head 2
(Syntax focused)

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Why do scaled dot-product attention?

Attention:

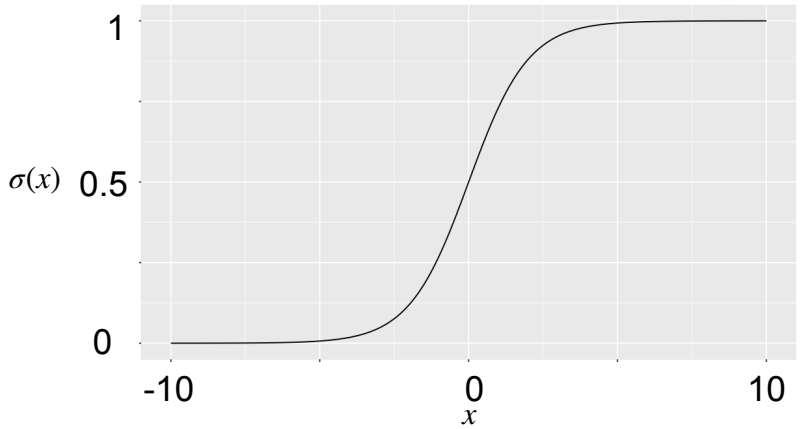
$$\text{softmax}\left(\frac{\mathbf{Q}\mathbf{V}^\top}{\sqrt{\mathbf{d_k}}}\right)\mathbf{V}$$

Can avoid

If we increase dimensionality, dot products become larger:

[1, 2] . [2, 3] = 8
[1, 2, 1] . [2, 3, 4] = 12
[1, 2, 1, 1] . [2, 3, 4, 8] = 20

Larger dot products mean smaller gradients after softmax:

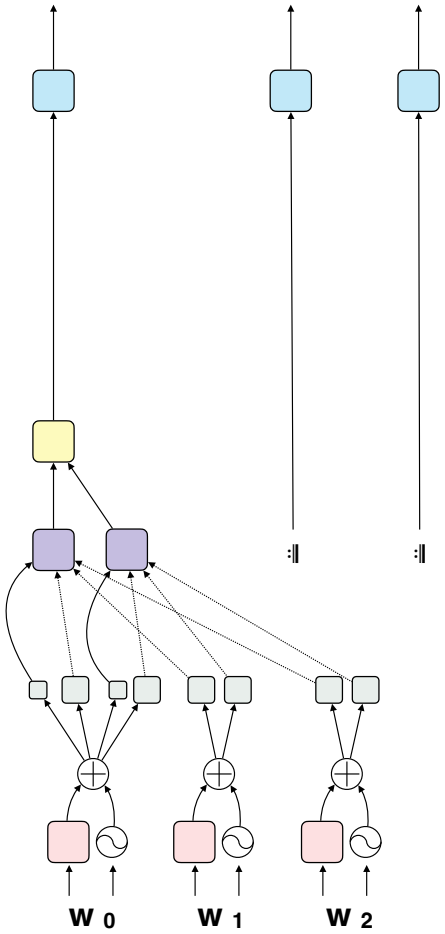


COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Going from Self-Attention to the Transformer: Multi-Head Attention, Residuals



Feed Forward

$$\max(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$$

⊕ Element-wise sum

Concatenate

Attention:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear transform

Look up vector

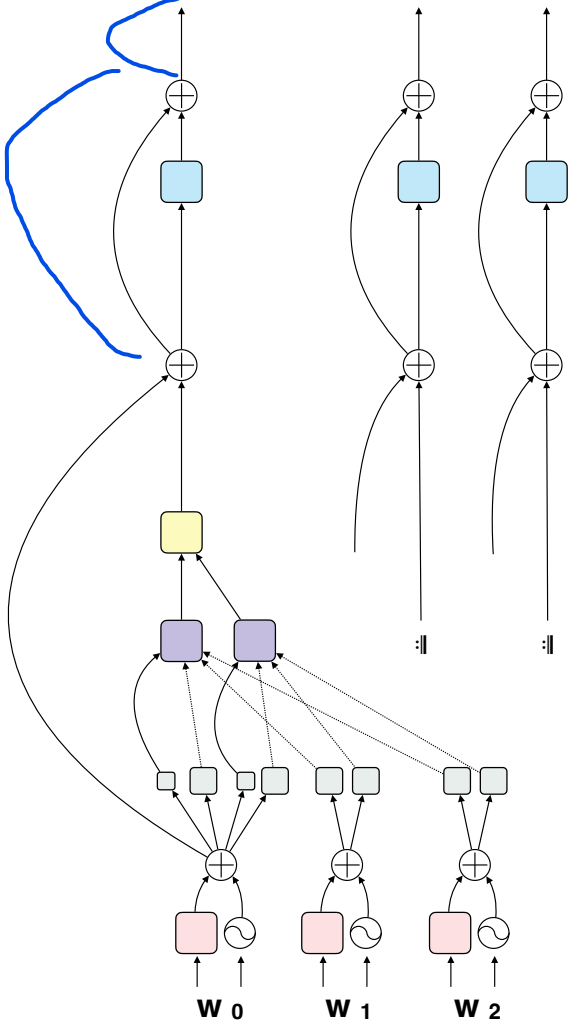
Position as vector

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Going from Self-Attention to the Transformer: Multi-Head Attention, Residuals



Feed Forward

$$\max(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$$

⊕ Element-wise sum

Concatenate

Attention:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear transform

Look up vector

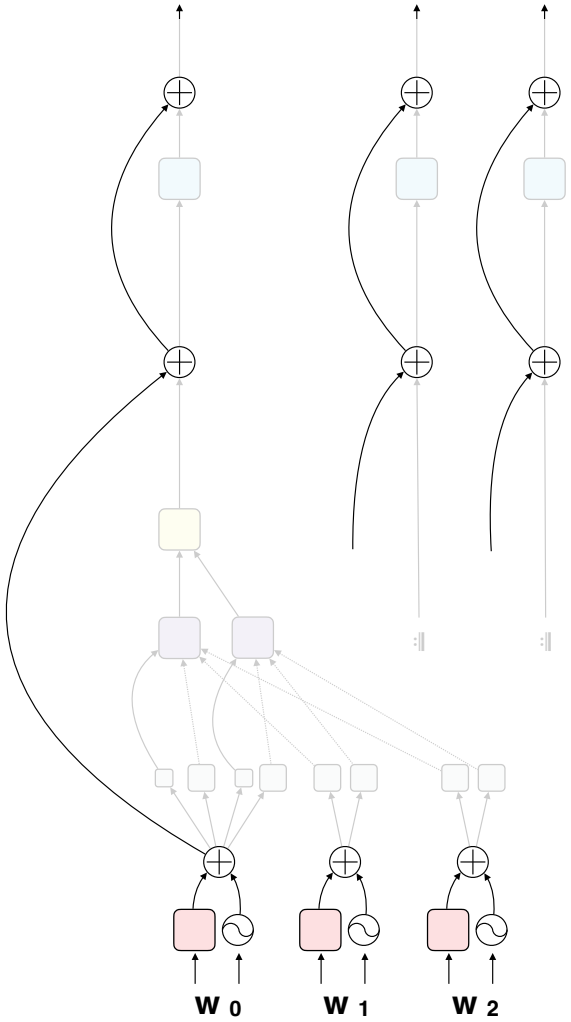
Position as vector

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Going from Self-Attention to the Transformer: Multi-Head Attention, Residuals



Feed Forward

$$\max(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$$

⊕ Element-wise sum

Concatenate

Attention:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear transform

Look up vector

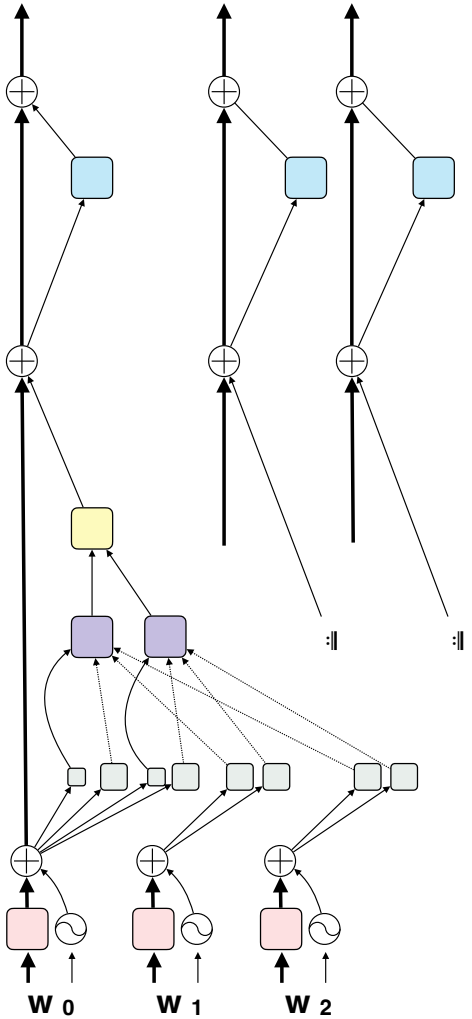
Position as vector

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Going from Self-Attention to the Transformer: Multi-Head Attention, Residuals



Feed Forward

$$\max(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$$

⊕ Element-wise sum

Concatenate

Attention:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear transform

Look up vector

Position as vector

COMP 4446 / 5046
Lecture 7, 2025

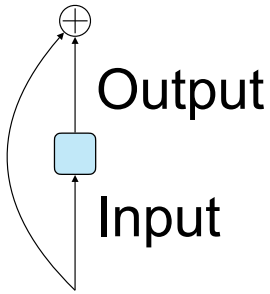
Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

What is a residual?

Originally from Computer Vision.
Taking the output of some calculation and adding the input to it:

means



Why?

- Gradient passes back easily (no non-linearity on the edge we added!)
- Encourages our model to capture something close to the identity function
- Frees up the model to use the weights to capture stuff other than the identity function

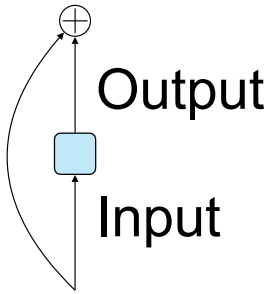
COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

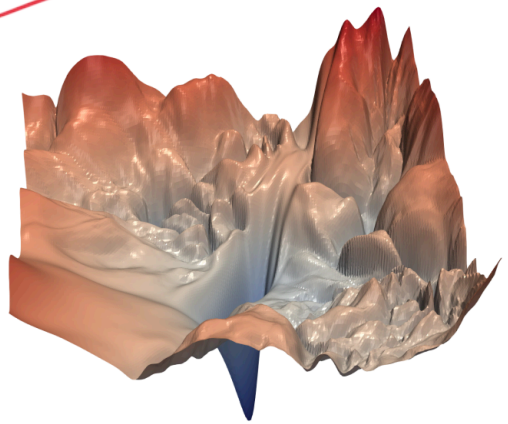
menti.com 2520 1730

What is a residual?

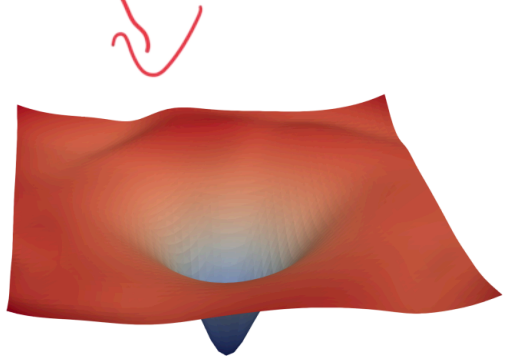
Originally from Computer Vision.
Taking the output of some calculation and adding the input to it:



Smooth gradients.



No residuals



Residuals

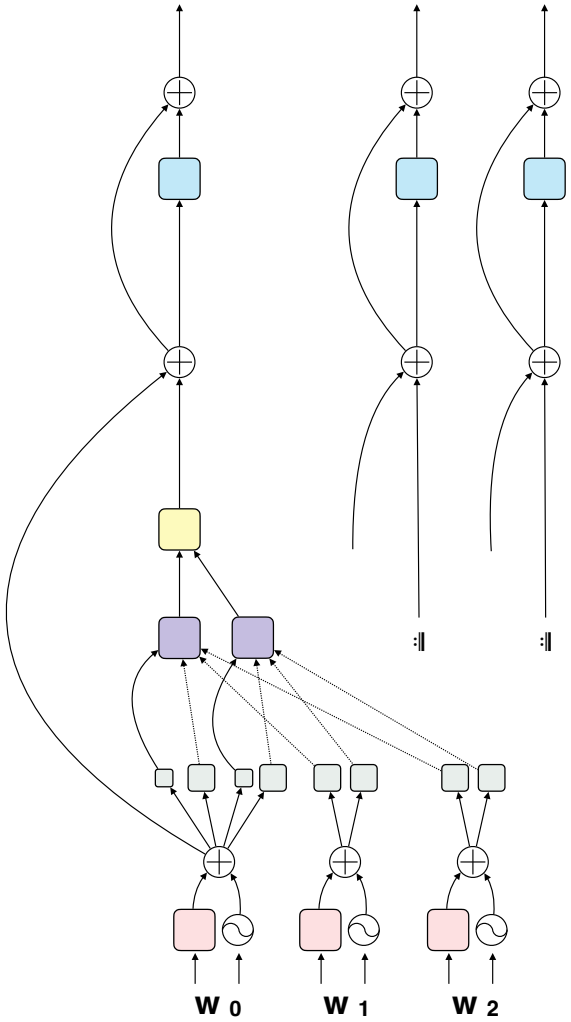
Li, et. Al. (NeurIPS 2018)

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Going from Self-Attention to the Transformer: Multi-Head Attention, Residuals, and Layer Normalization



Feed Forward

$$\max(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$$

⊕ Element-wise sum

Concatenate

Attention:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear transform

Look up vector

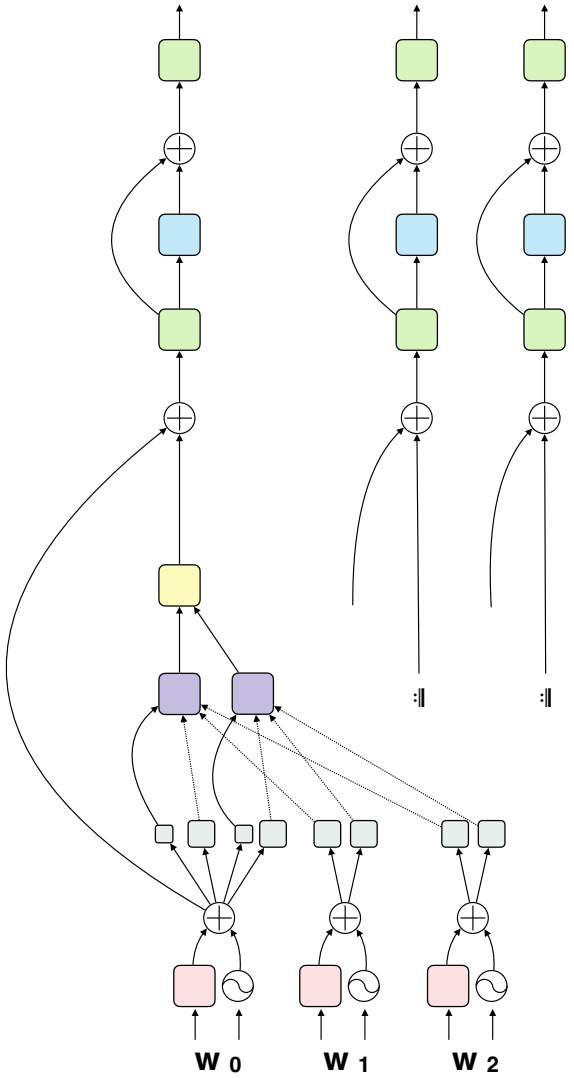
Position as vector

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Going from Self-Attention to the Transformer: Multi-Head Attention, Residuals, and Layer Normalization



Feed Forward

$$\max(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$$

Layer Norm

⊕ Element-wise sum

Concatenate

Attention:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear transform

Look up vector

Position as vector

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Layer Normalization shifts values in a way that improves training

Take a vector $x \in \mathbb{R}^d$

Calculate the mean and standard deviation of the vector's values

$$\mu = \sum_{j=1}^{d} x_j$$

$$\sigma = \sqrt{\frac{1}{d} \sum_{j=1}^{d} (x_j - \mu)^2}$$

Define two learned weight vectors (or set them to 1 and 0)

$\gamma \in \mathbb{R}^d$ $\beta \in \mathbb{R}^d$

Rescale the vector

$$\text{output} = \frac{x - \mu}{\sqrt{\sigma + \epsilon}} * \gamma + \beta$$

to square root

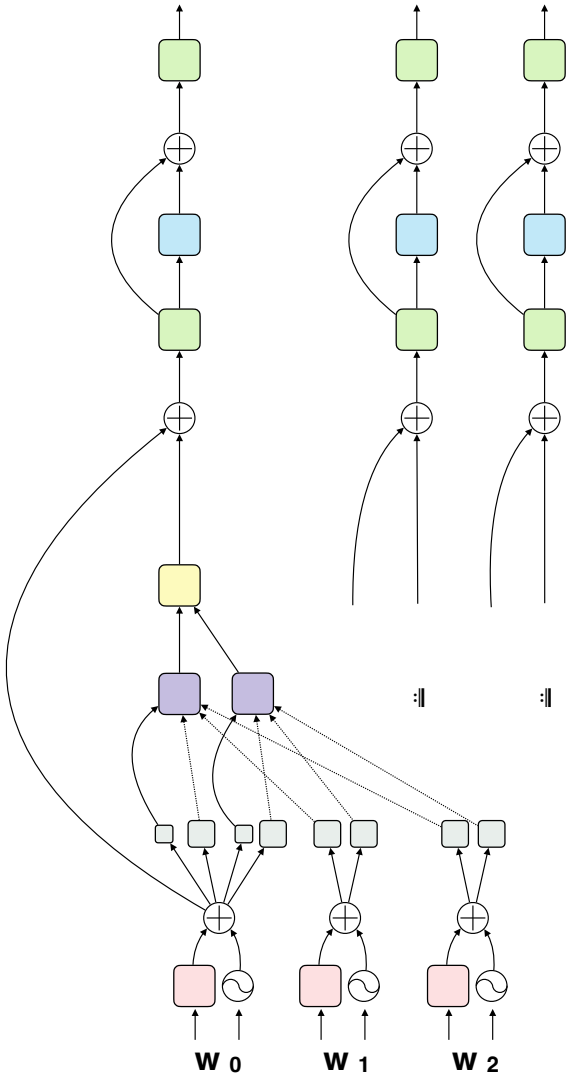
A small value to prevent division by zero

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Going from Self-Attention to the Transformer: Multi-Head Attention, Residuals, and Layer Normalization



Feed Forward

$$\max(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$$

Layer Norm

⊕ Element-wise sum

Concatenate

Attention:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear transform

Look up vector

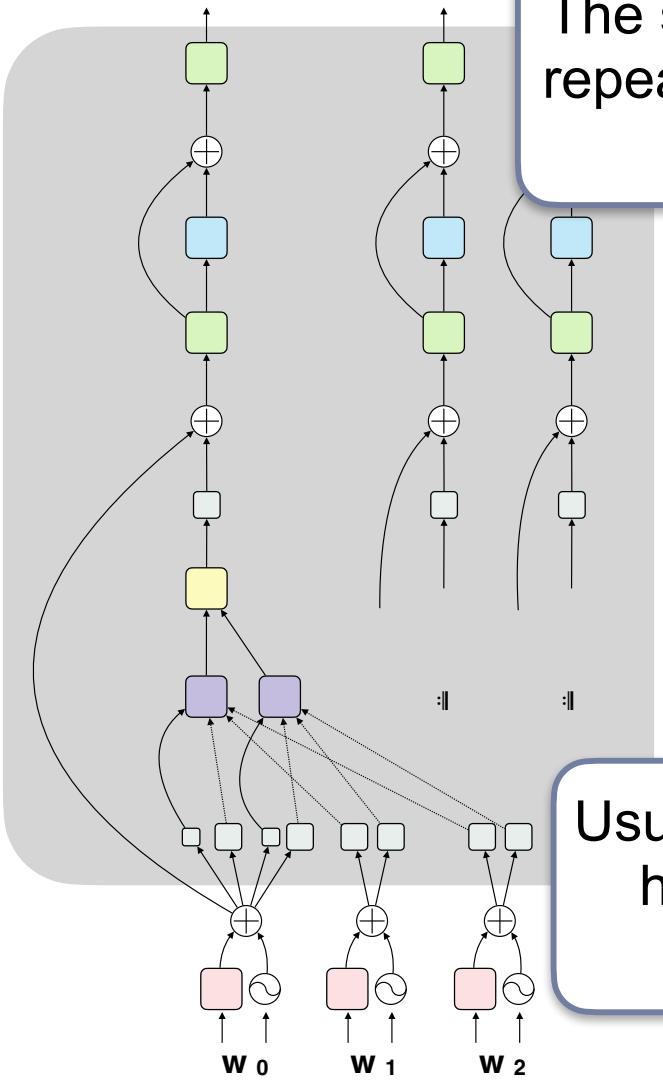
Position as vector

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

This is the Transformer encoder! (Almost)



The shaded area is repeated in a stack, 6 times

Usually more than 2 heads (original paper had 8)

Feed Forward
$\max(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$

Layer Norm

⊕ Element-wise sum

Concatenate

Attention:
$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear transform

Look up vector

Position as vector

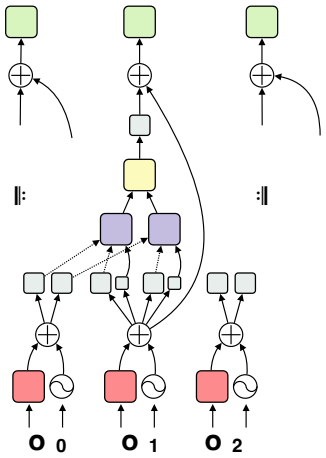
COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

What about the decoder?

Start the same as the encoder, except attention does not apply to the future



Softmax

Feed Forward

$$\max(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$$

Layer Norm

⊕ Element-wise sum

Concatenate

Attention:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear transform

Look up vector

Position as vector

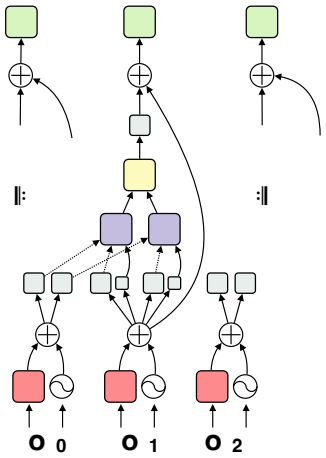
COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

What about the decoder?

Provide encoder state using attention, also multi-head



Softmax

Feed Forward

$$\max(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$$

Layer Norm

⊕ Element-wise sum

Concatenate

Attention:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear transform

Look up vector

Position as vector

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

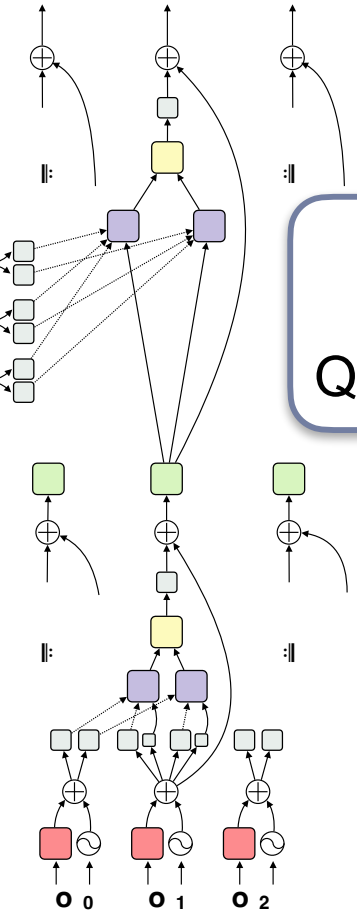
What about the decoder?

Provide encoder state using attention, also multi-head

Called ‘cross attention’ to distinguish it from ‘self attention’

Encoder

Input text



decoder

K and V come from encoder
Q comes from decoder

Softmax

Feed Forward

$$\max(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$$

Layer Norm

Attention:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear transform

Look up vector

Position as vector

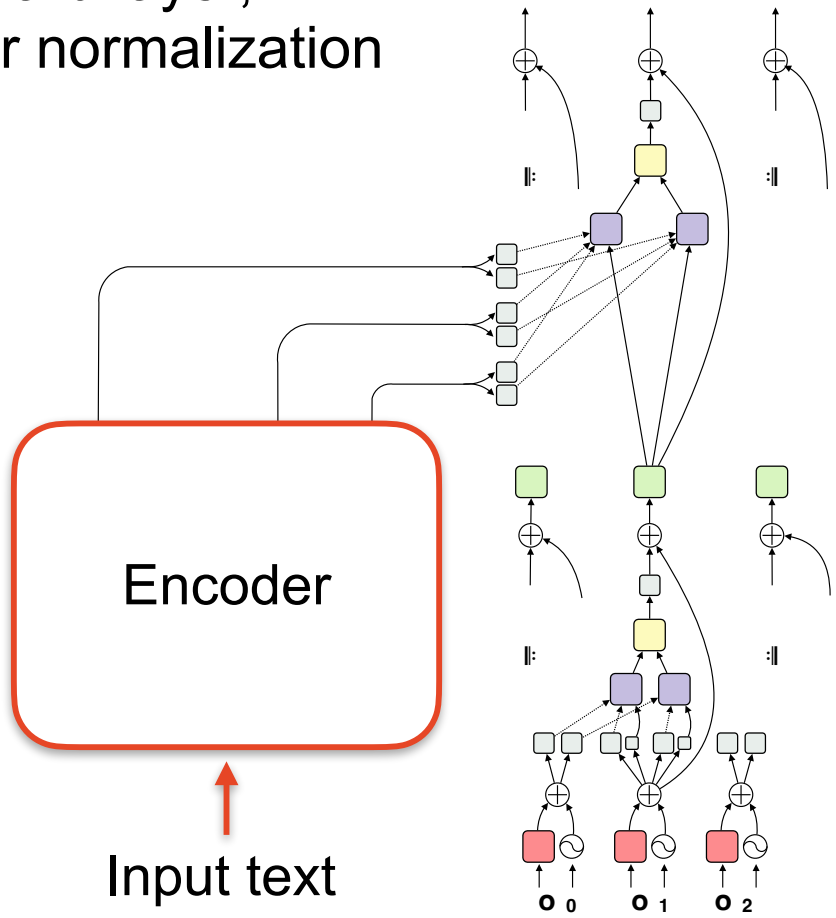
COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

What about the decoder?

Add another feed forward layer, with layer normalization



Softmax

Feed Forward

$$\max(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$$

Layer Norm

⊕ Element-wise sum

Concatenate

Attention:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear transform

Look up vector

Position as vector

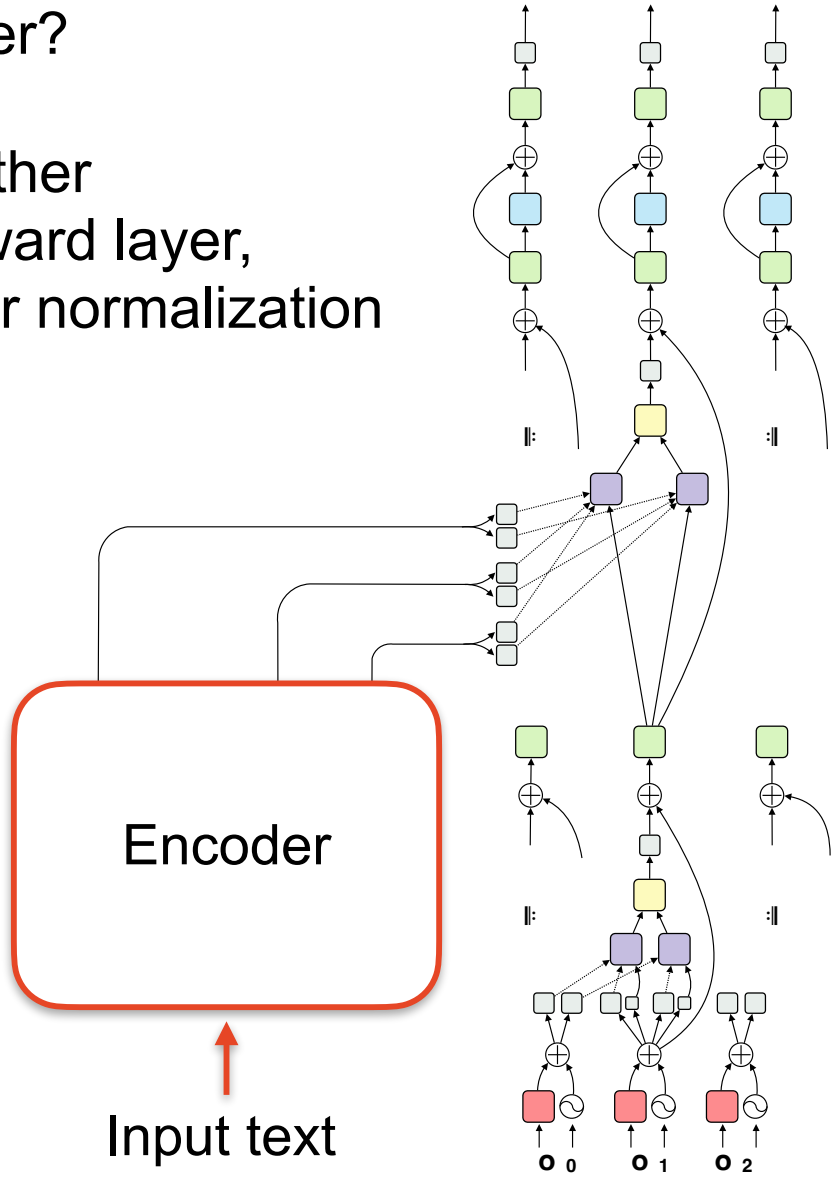
COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

What about the decoder?

Add another feed forward layer, with layer normalization



Softmax

Feed Forward

$$\max(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$$

Layer Norm

⊕ Element-wise sum

Concatenate

Attention:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear transform

Look up vector

Position as vector

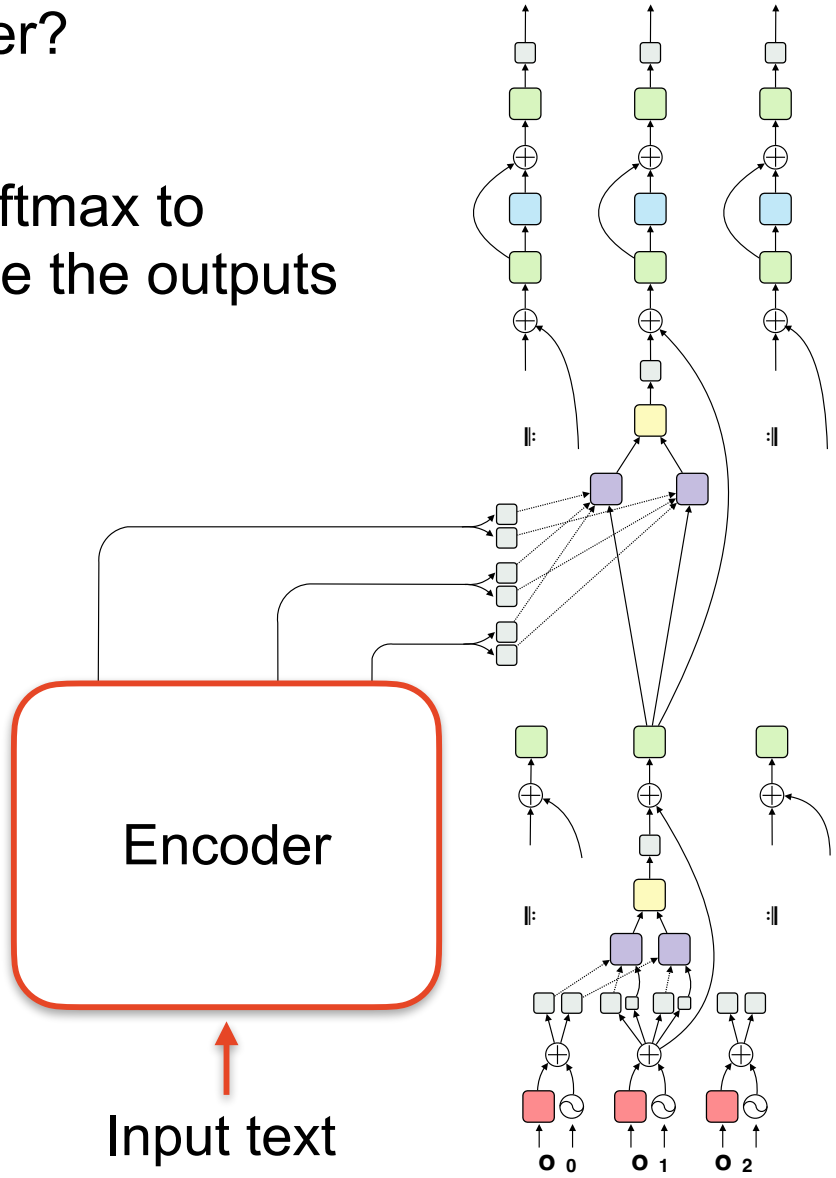
COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

What about the decoder?

Apply softmax to determine the outputs



Softmax

Feed Forward

$$\max(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$$

Layer Norm

⊕ Element-wise sum

Concatenate

Attention:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear transform

Look up vector

Position as vector

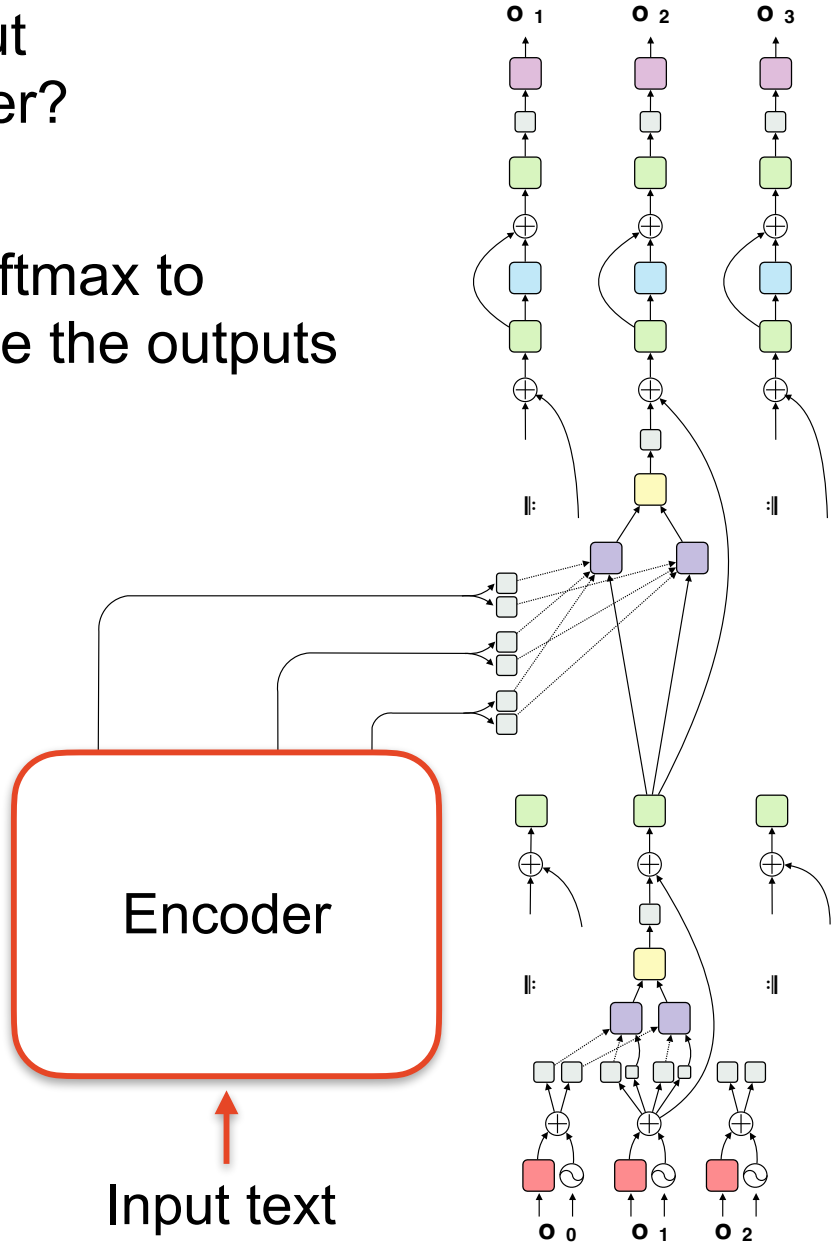
COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

What about the decoder?

Apply softmax to determine the outputs



Softmax

Feed Forward

$$\max(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$$

Layer Norm

⊕ Element-wise sum

Concatenate

Attention:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear transform

Look up vector

Position as vector

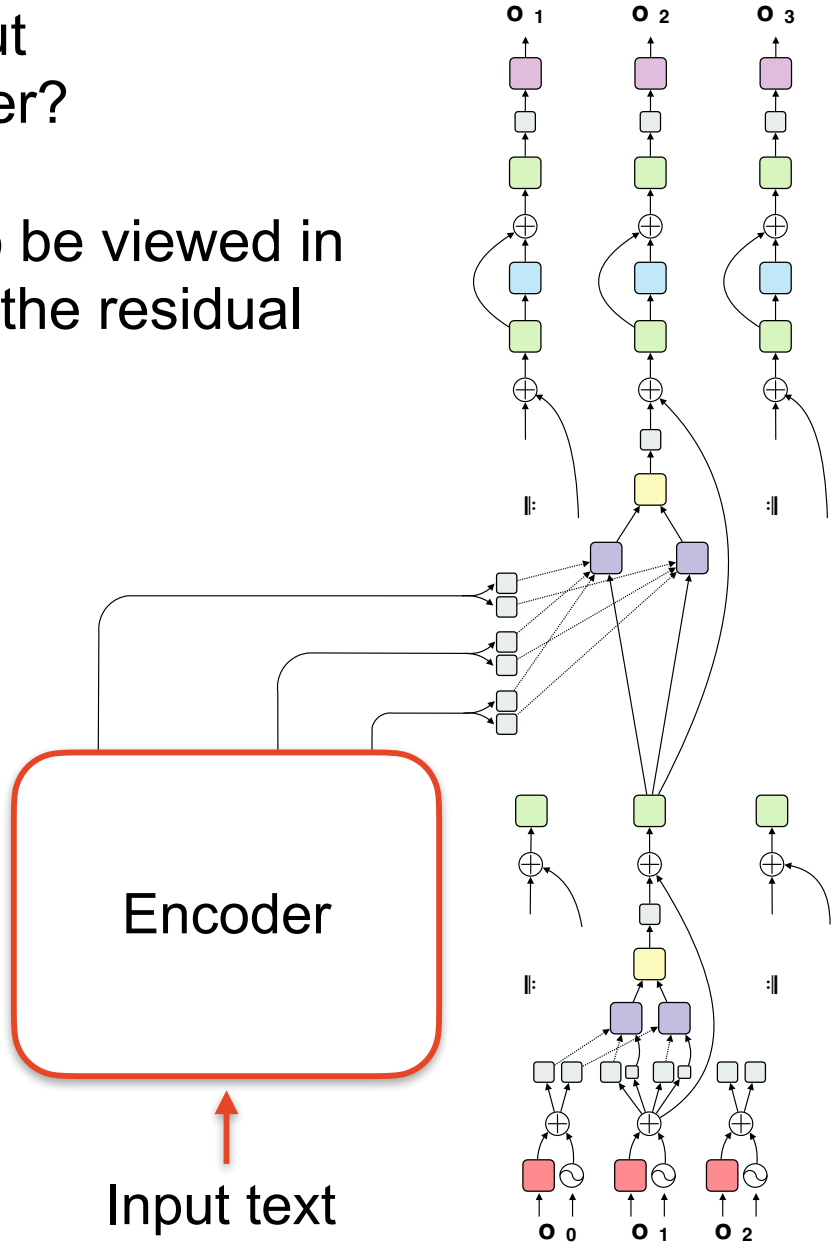
COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

What about the decoder?

Can also be viewed in terms of the residual stream



Softmax

Feed Forward

$\max(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$

Layer Norm

⊕ Element-wise sum

Concatenate

Attention:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear transform

Look up vector

Position as vector

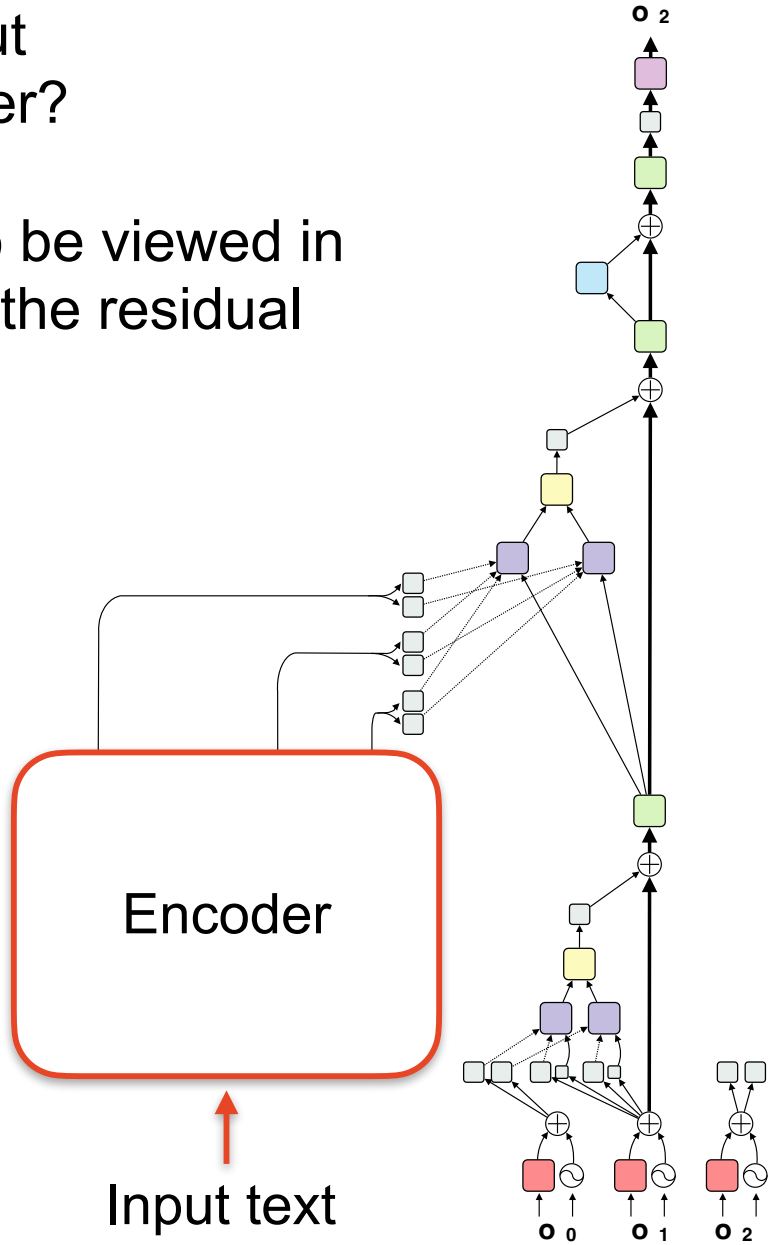
COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

What about the decoder?

Can also be viewed in terms of the residual stream



Softmax

Feed Forward

$\max(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$

Layer Norm

⊕ Element-wise sum

Concatenate

Attention:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear transform

Look up vector

Position as vector

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

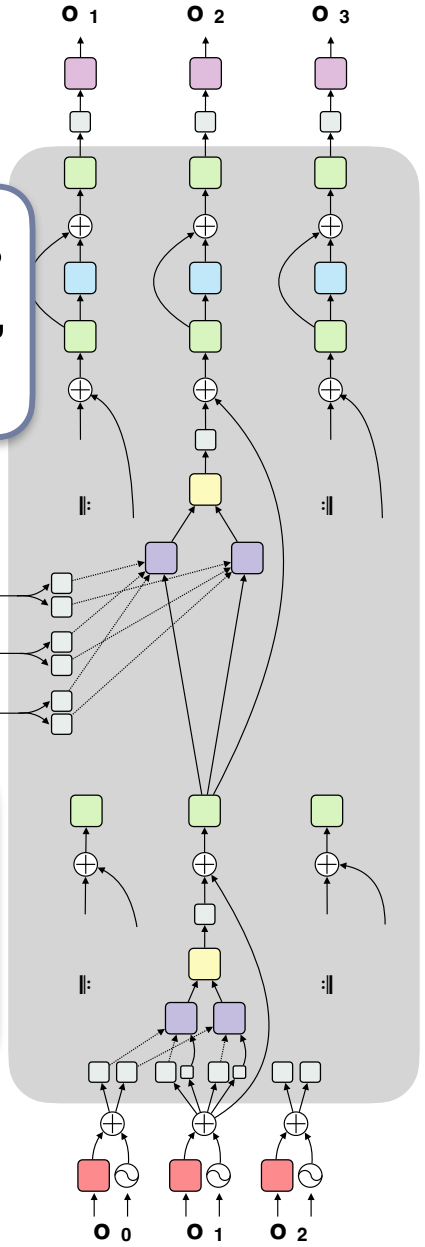
This is the Transformer decoder! (Almost)

The shaded area is repeated in a stack, 6 times

Usually more than 2 heads (original paper had 8)

Encoder

Input text



Softmax

Feed Forward

$$\max(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1})\mathbf{W_2} + \mathbf{b_2}$$

Layer Norm

⊕ Element-wise sum

Concatenate

Attention:

$$\text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear transform

Look up vector

Position as vector

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

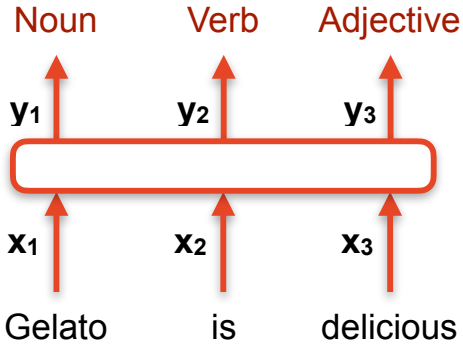
Training and Use

COMP 4446 / 5046
Lecture 7, 2025

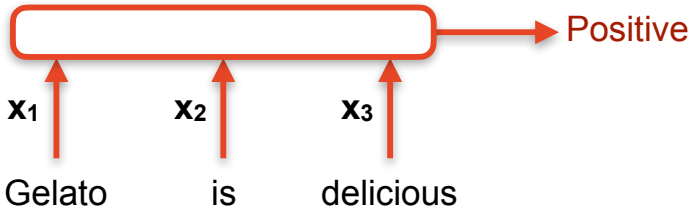
Self-Attention
Transformer
Training and Use
Lab Preview

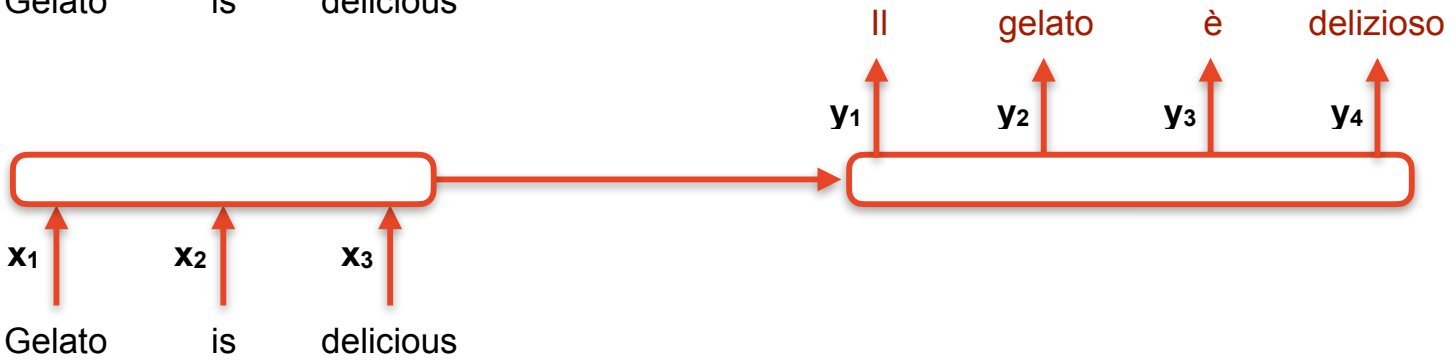
menti.com 2520 1730

How do we use this? - Just like the RNN encoder-decoders



Or use it to create contextual word representations that go into another model





COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Initial results showed slight accuracy gains, big efficiency gains

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

Vaswani, et al., (NeurIPS 2017)

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Over time, transformer models came to be dominant

Summarisation (Lin, et. al., ICLR 2018)

Model	Test perplexity	ROUGE-L
seq2seq-attention, $L = 500$	5.04952	12.7
Transformer-ED, $L = 500$	2.46645	34.2
Transformer-D, $L = 4000$	2.22216	33.6
Transformer-DMCA, no MoE-layer, $L = 11000$	2.05159	36.2
Transformer-DMCA, MoE-128, $L = 11000$	1.92871	37.9
Transformer-DMCA, MoE-256, $L = 7500$	1.90325	38.8

GLUE Benchmark - a range of tasks (Devlin, et. al., NAACL 2019)

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7
BiLSTM+ELMo	.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8
OpenAI GPT	.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4
$\text{BERT}_{\text{LARGE}}$	**.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**

GPT 1

BERT

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

BERT

include

WordPiece tokenisation (last lecture)
+
Transformer architecture (this lecture)
+
Data
=
High performance!

This was the moment when training large models became out of reach for most university research labs

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

Faster training means they can scale

Original paper (2017)

	N	d_{model}	d_{ff}	h	d_k	d_v
base	6	512	2048	8	64	64

GPT-3 (2020)

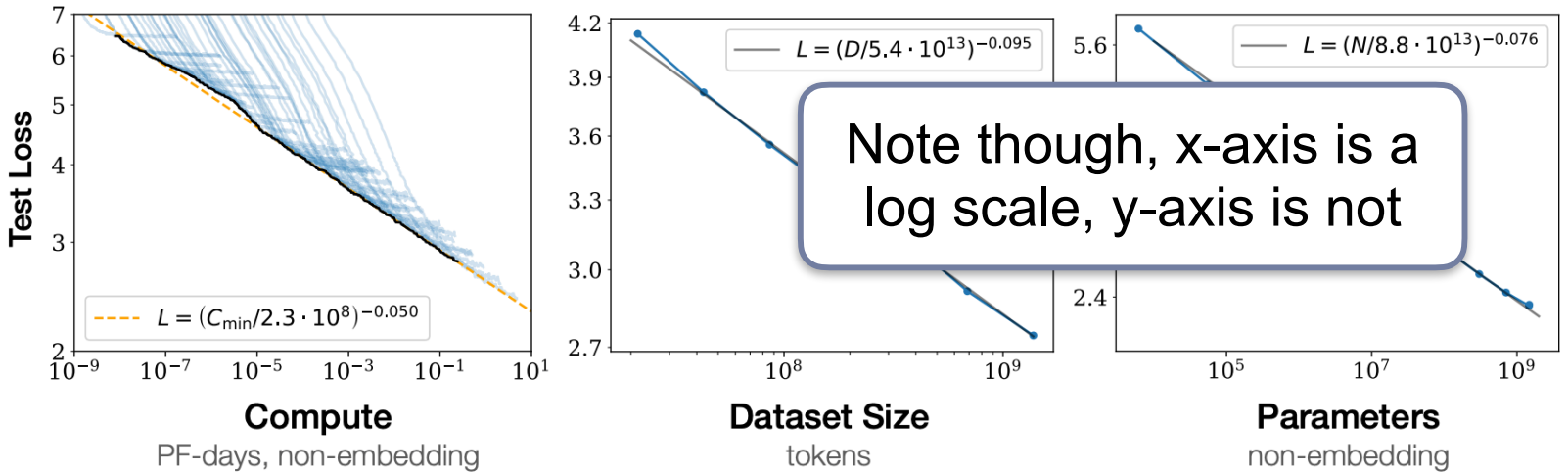
Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}
GPT-3 Small	125M	12	768	12	64
GPT-3 Medium	350M	24	1024	16	64
GPT-3 Large	760M	24	1536	16	96
GPT-3 XL	1.3B	24	2048	24	128
GPT-3 2.7B	2.7B	32	2560	32	80
GPT-3 6.7B	6.7B	32	4096	32	128
GPT-3 13B	13.0B	40	5140	40	128
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

They scale very well to more compute, data, and parameters



Note though, x-axis is a log scale, y-axis is not

This is where 'scaling laws' come from - methods to estimate the best model size given a certain amount of data and a budget for training

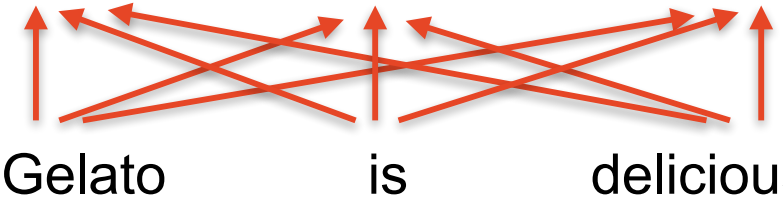
Kaplan, et al., (arXiv 2020)

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

But, attention is quadratic, will that be an issue?



Lots of clever variants to avoid it!
- PerFormer
- LongFormer
- LinFormer

But… it turns out that the bigger issue is memory bandwidth

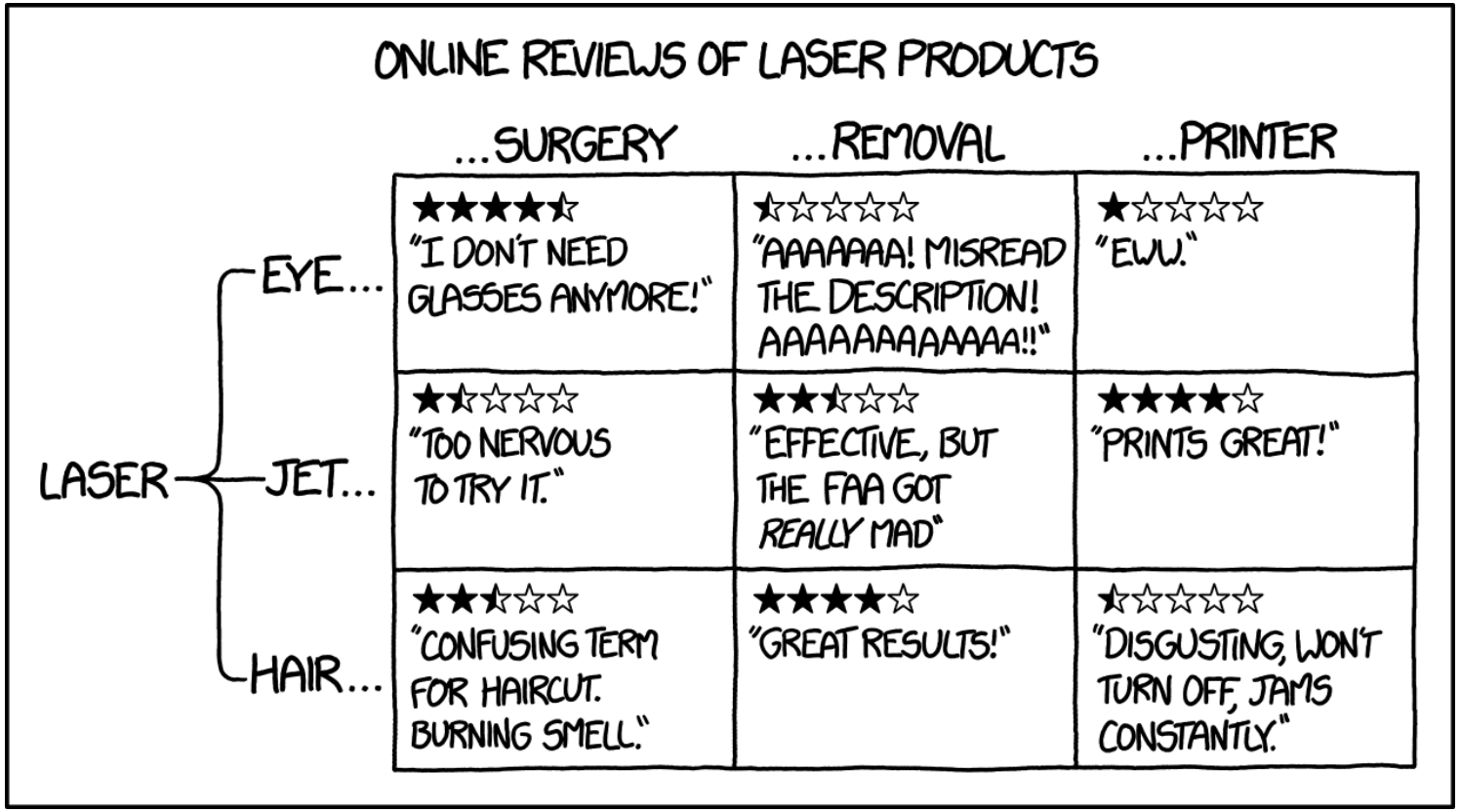
Most notable improvement - FlashAttention

Feedforward computation dominates as model size increases

3 minute Break - stretch and visit Menti

menti.com
2520 1730

Laser Products



[ERRORS: HAIR JAM. COLOR-SAFE CONDITIONER CARTRIDGE RUNNING LOW. LEGAL-SIZE HAIR TRAY EMPTY, USING LETTER-SIZE HAIR ONLY.]

Source: https://xkcd.com/1681/

menti.com 2520 1730

Lab Preview

COMP 4446 / 5046
Lecture 7, 2025

Self-Attention
Transformer
Training and Use
Lab Preview

menti.com 2520 1730

1. multiple stack : better performance, more scenario?

2. mask purpose

HuggingFace!

3. rotary positional embedding

Mid-semester survey

https://forms.office.com/r/4aHbx4U4mY

All questions optional

I will read (manually - no AI!) and report back results + my plans in response

Muddy Card

https://saipll.shinyapps.io/student-interface/

If you do not wish to participate in the study, use the Ed form instead

Go to Ed → Lessons → Muddy Cards Lecture 7