Natural Language Processing

Self Attention and Transformers

Kabir Ahuja
kahuja@cs.washington.edu

Adapted from slides from by Vidhisha Balachandran, Emma Strubell, Graham Neubig

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

Readings

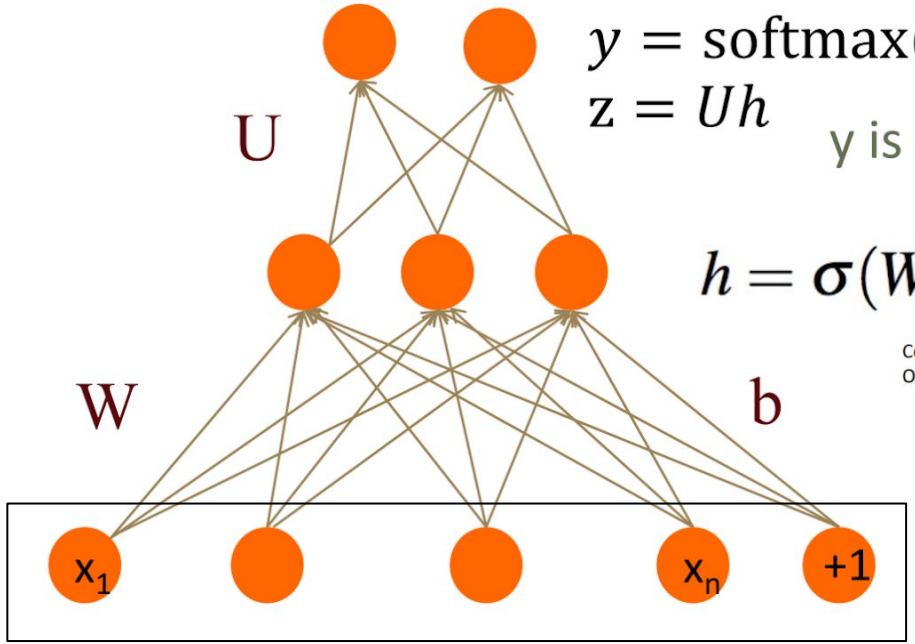
- Attention Is All You Need
- The Illustrated Transformer
- The Annotated Transformer
- Language Modeling with Transformers and PyTorch

Recap - 2 Layer MLP

Output layer
(σ node)

hidden units
(σ node)

Input layer
(vector)



$y = \text{softmax}(z)$

$z = Uh$

y is a vector

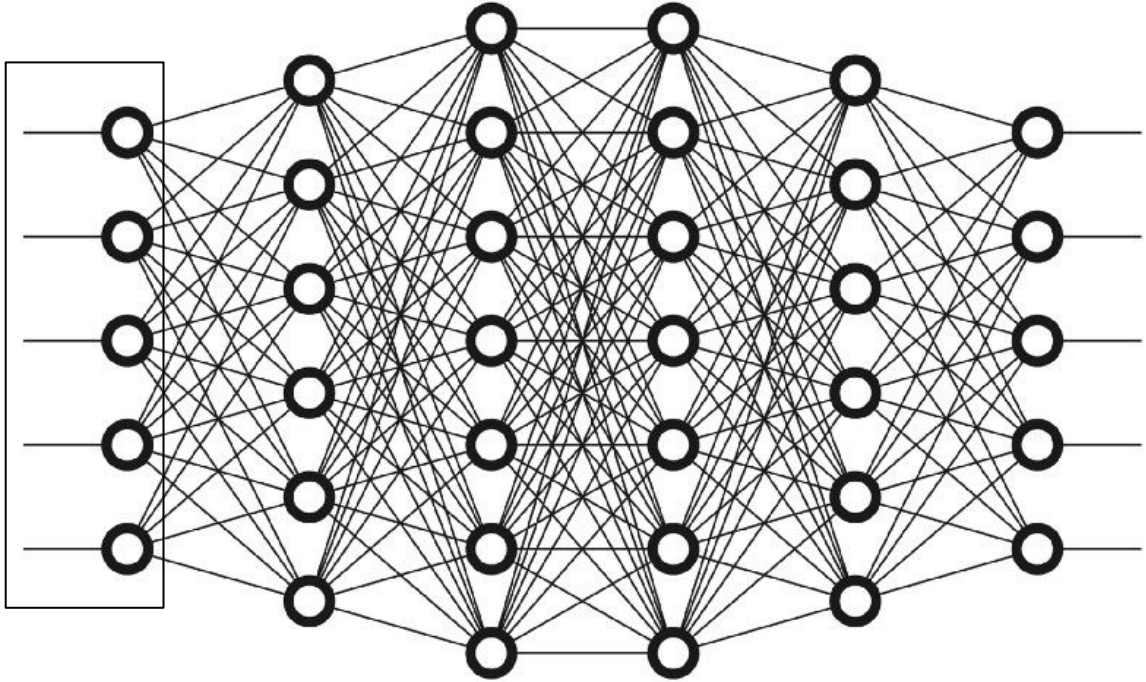
$h = \sigma(Wx + b)$

Could be ReLU
Or tanh

Fixed-Size Inputs

Deep MLP

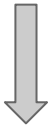
Fixed-Size Inputs



Fixed size representations for Natural Language

How to turn this

Learning NLP is fun!



Into this

[x_1, x_2, …, x_d]

(fixed length d-dimensional vector)

Fixed size representations for Natural Language

Remember Homework 1

Var	Definition	Value in Fig. 5.2
x_1	count(positive lexicon words $\in$ doc)	3
x_2	count(negative lexicon words $\in$ doc)	2
x_3	$\begin{cases} 1 & \text{if "no"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_4	count(1st and 2nd pronouns $\in$ doc)	3
x_5	$\begin{cases} 1 & \text{if "!"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	0
x_6	ln(word count of doc)	ln(66) = 4.19

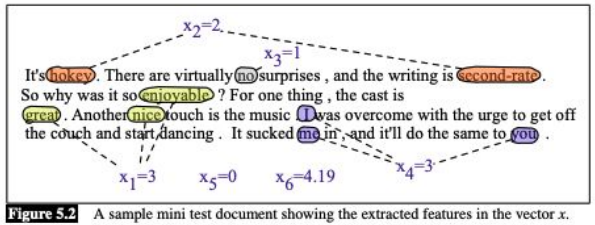


Figure 5.2 A sample mini test document showing the extracted features in the vector *x*.

Linguistic Features

As an example consider we have 2 examples present in our dataset:

x1: john likes to watch movies mary likes movies too

x2: mary also likes to watch football games

Based on these two documents we can get the list of all words that occur in this dataset which will be:

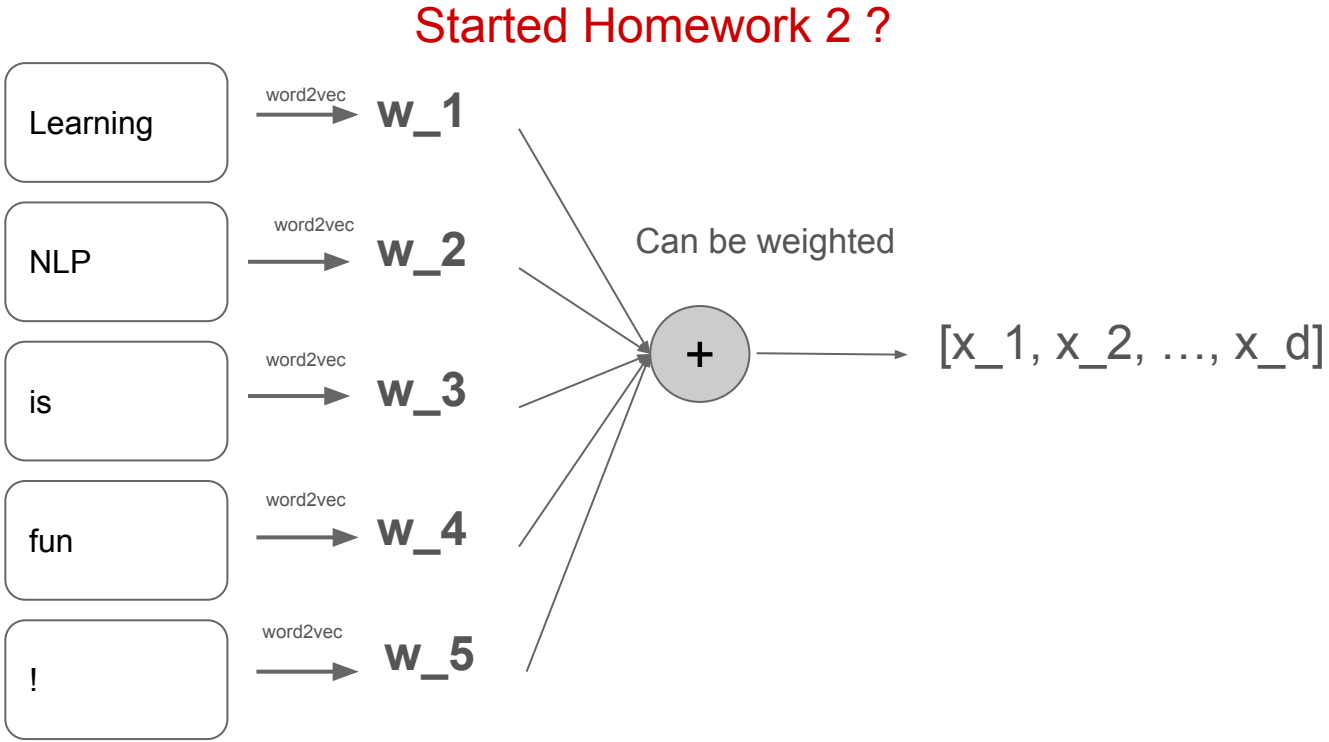
index	word
0	also
1	football
2	games
3	john
4	likes
5	mary
6	movies
7	to
8	too
9	watch

We can then define features for the two x1 and x2 as follows:

	also	football	games	john	likes	mary	movies	to	too	watch
x1	0	0	0	1	2	1	2	1	2	1
x2	1	1	1	0	1	1	0	0	0	0

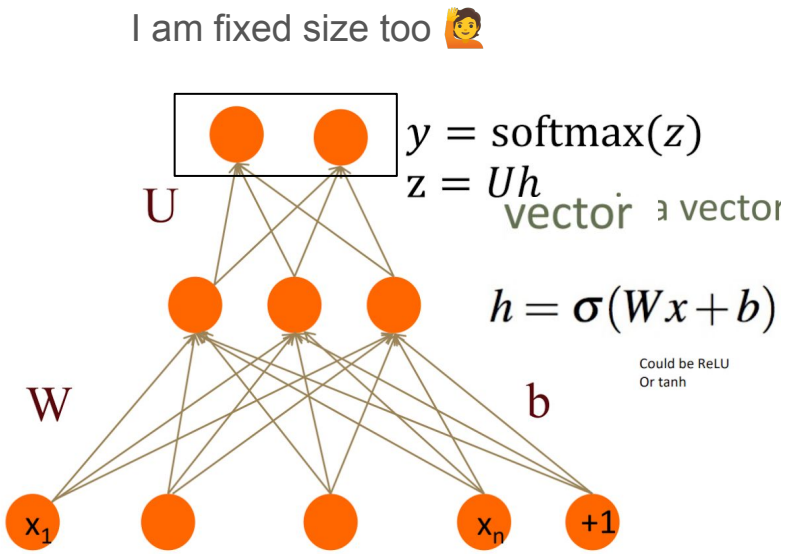
Bag of words with counts

Fixed size representations for Natural Language



Problems with Fixed-Size Representations

1. Loss of important contextual information
2. Order invariant
3. Often Poor Generalization (Especially in case of Linguistic Features)
4. What if the output is a sequence too? Think about tasks like machine translation, good luck getting the sentence back from the fixed representation :)



Sequence Models To Rescue

Recurrent Neural Networks - RNNs

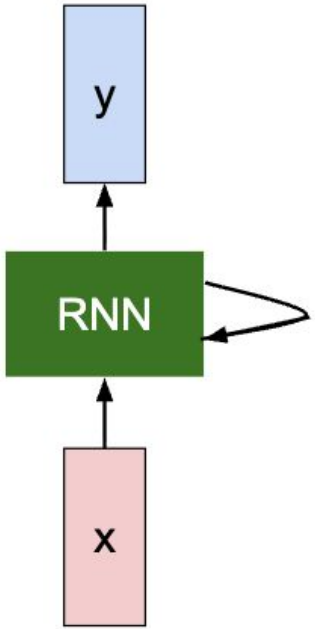
For language it can be **sequence of words, characters, bytes** etc.

We can process a sequence of vectors **x** by applying a **recurrence formula** at every time step:

Both fixed size vectors

$$h_t = f_W(h_{t-1}, x_t)$$

new state — some function with parameters W — old state — input vector at some time step



Think of x as word-vector

From CS231n slides. Lecture 8 by Fei-Fei Li, Yunzhu Li, Ruohan Gao

Recurrent Neural Networks - RNNs

RNN output generation

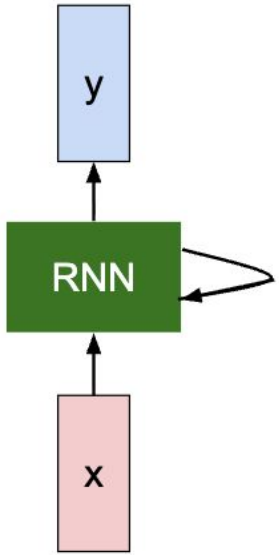
We can process a sequence of vectors **x** by applying a **recurrence formula** at every time step:

$$y_t = f_{W_{hy}}(h_t)$$

output

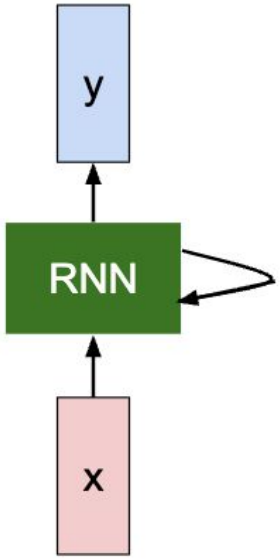
new state

another function with parameters W_o



From CS231n slides. Lecture 8 by Fei-Fei Li, Yunzhu Li, Ruohan Gao

Recurrent Neural Networks - RNNs



$$h_t = f_W(h_{t-1}, x_t)$$

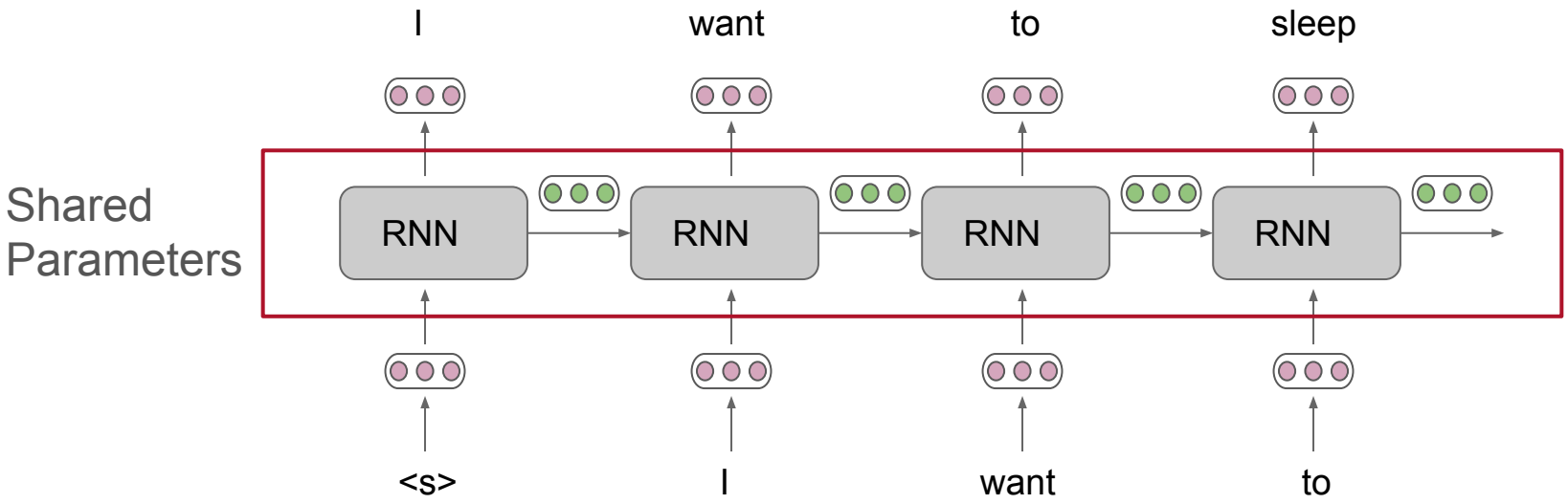
$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$

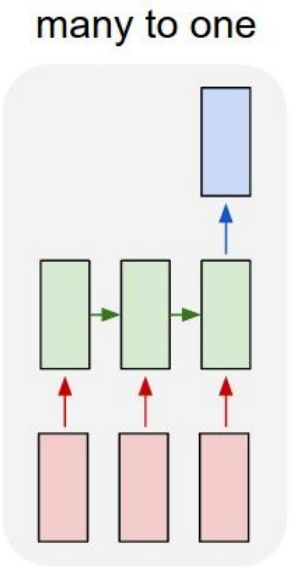
Sometimes called a “Vanilla RNN” or an “Elman RNN” after Prof. Jeffrey Elman

From CS231n slides. Lecture 8 by Fei-Fei Li, Yunzhu Li, Ruohan Gao

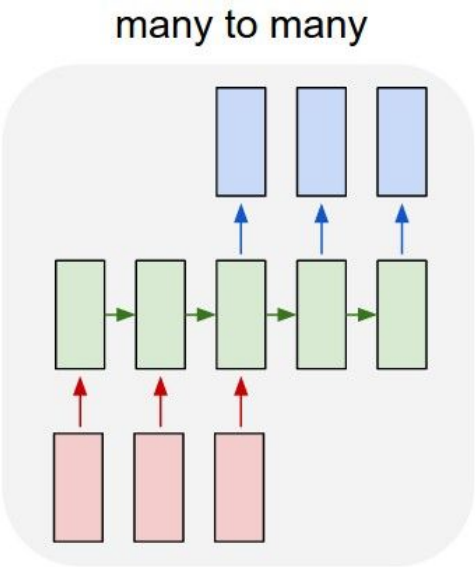
Recurrent Neural Networks - RNNs



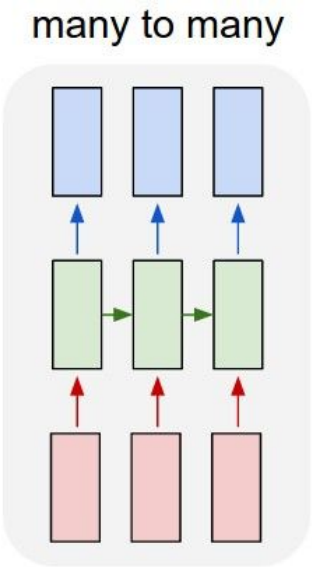
Multiple Ways of Stacking RNNs



For tasks like classification



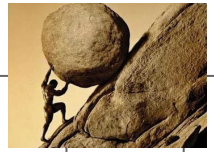
For tasks like machine translation



For tasks like language modeling

From The Unreasonable Effectiveness of Recurrent Neural Networks by Andrej Karpathy

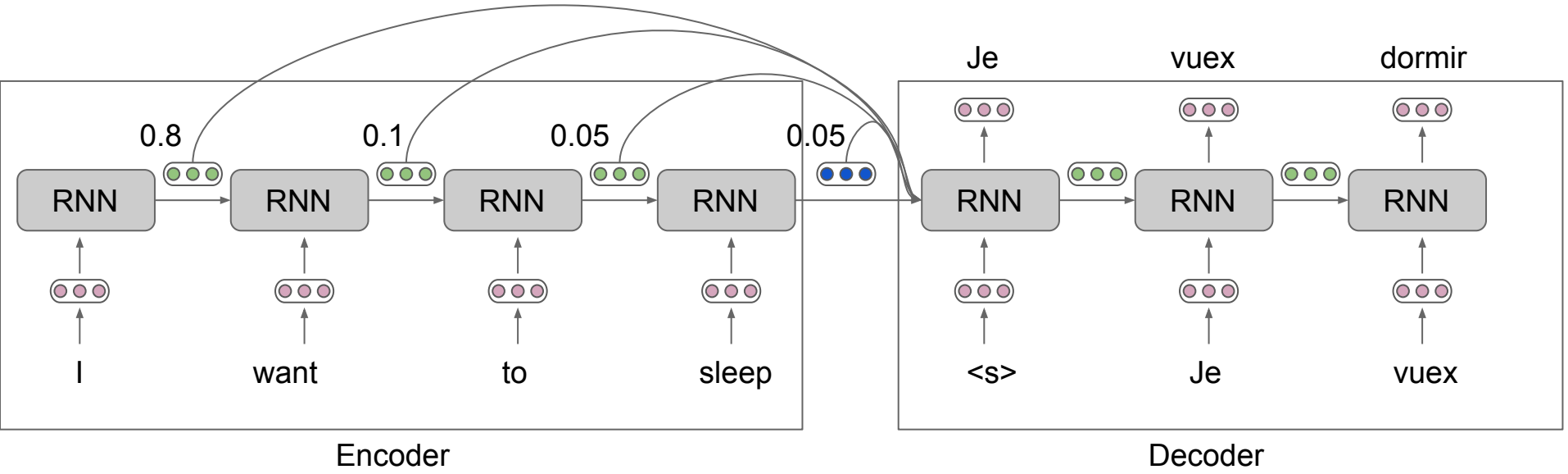
Encoder-Decoder Models



Typically Encoder and Decoder would be separate networks i.e. have their own separate weights

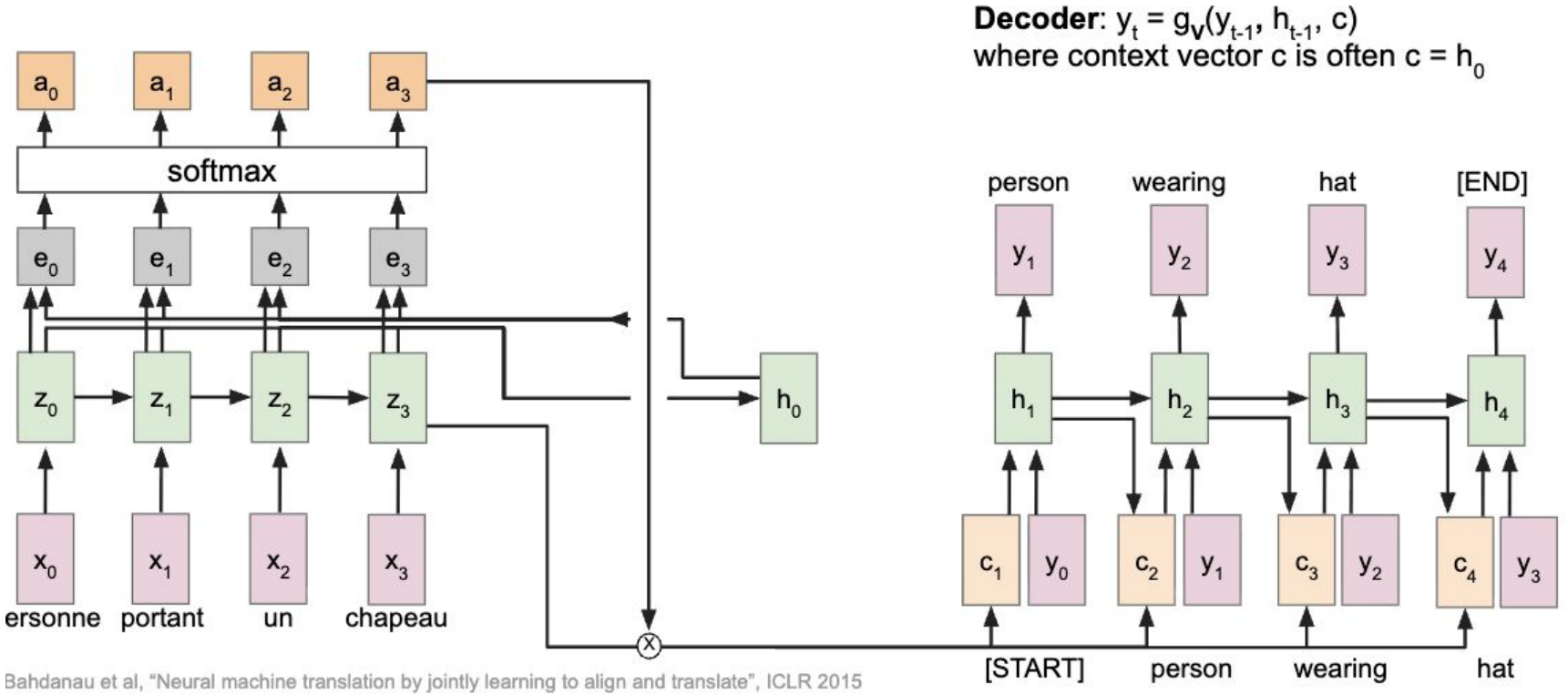
Ilya Sutskever, Oriol Vinyals, Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks.

Encoder-Decoder Models With Attention



Neural Machine Translation by Jointly Learning to Align and Translate. Bahdanau et al, 2015

Encoder-Decoder Models With Attention



Bahdanau et al, "Neural machine translation by jointly learning to align and translate", ICLR 2015

From CS231n slides. Lecture 9 by Fei-Fei Li, Yunzhu Li, Ruohan Gao

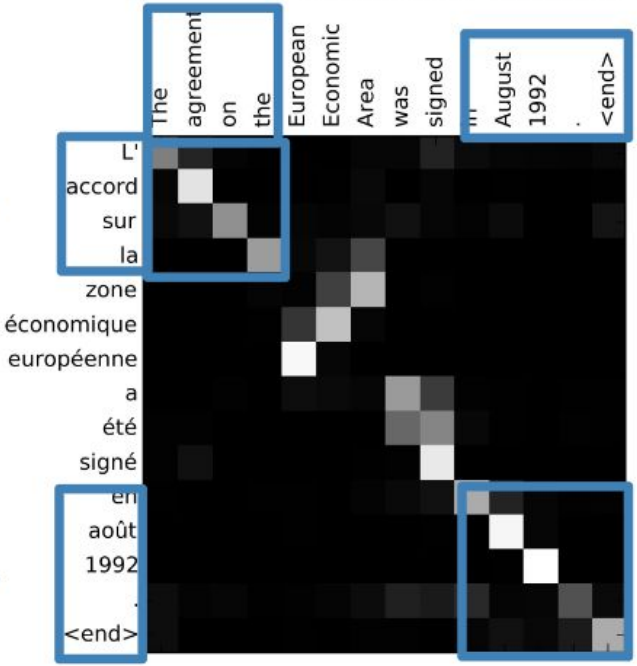
Encoder-Decoder Models With Attention

Example: English to French translation

Input: "**The agreement on the** European Economic Area was signed **in August 1992**."

Output: "**L'accord sur la** zone économique européenne a été signé **en août 1992**."

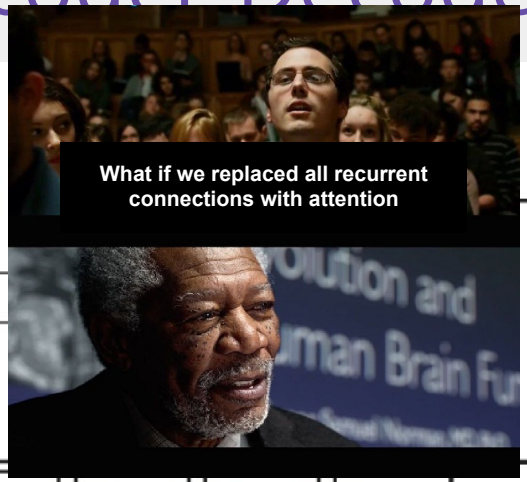
Visualize attention weights $a_{t,i}$



Bahdanau et al, "Neural machine translation by jointly learning to align and translate", ICLR 2015

From CS231n slides. Lecture 9 by Fei-Fei Li, Yunzhu Li, Ruohan Gao

Encoder-Decoder Models With Attention



Decoder: $y_t = g_V(y_{t-1}, h_{t-1}, c)$
where context vector c is often $c = h_0$

Do you even need me anymore 🥺

Bahdanau et al, "Neural machine translation by jointly learning to align and translate", ICLR 2015

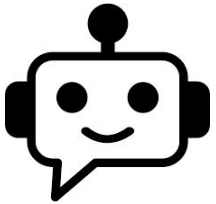
What's wrong with RNNs?

- Long Range Dependencies
- Gradient vanishing / explosion
- Long time to converge
- Expensive computation

Long Range Dependencies

I'm want to watch Wicked! How does the weather in NYC look next week?

It looks sunny with some light rain during the weekend.



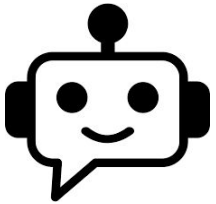
Oh! But I don't have a rain jacket :(Is there a store nearby?

There's a marshall's a mile away. They have the navy blue jacket you have been eyeing for a while!

Long Range Dependencies

I'm want to watch Wicked! How does the weather in NYC look next week?

It looks sunny with some light rain during the weekend.



Oh! But I don't have a rain jacket :(Is there a store nearby?

There's a marshall's a mile away. They have the navy blue jacket you have been eyeing for a while!

Ok! Looks like I can actually go! Book **the tickets** for next Wed!

Long Range Dependencies

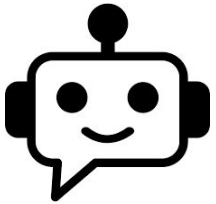
I'm want to watch **Wicked**! How does the weather in NYC look next week?

It looks sunny with some light rain during the weekend.

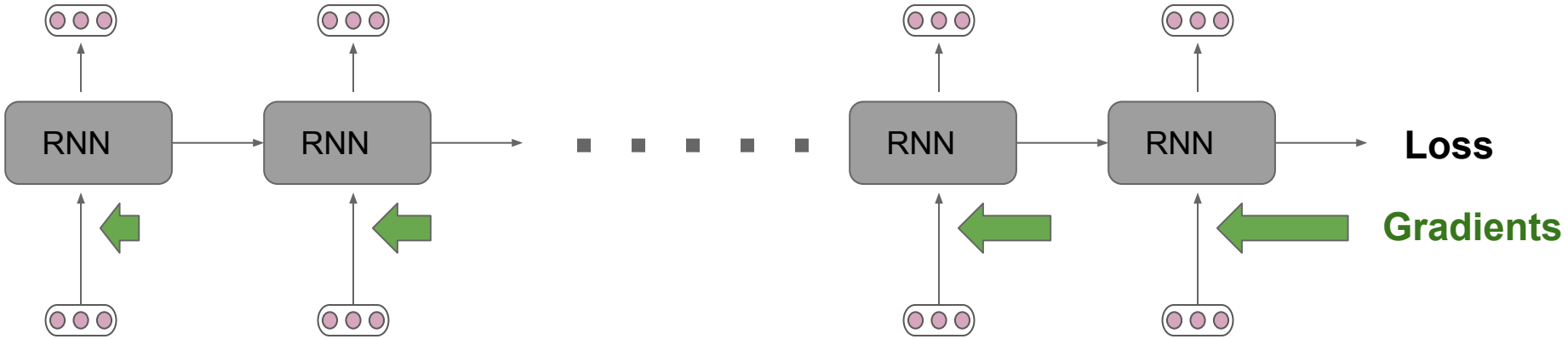
Oh! But I don't have a rain jacket :(Is there a store nearby?

There's a marshall's a mile away. They have the navy blue jacket you have been eyeing for a while!

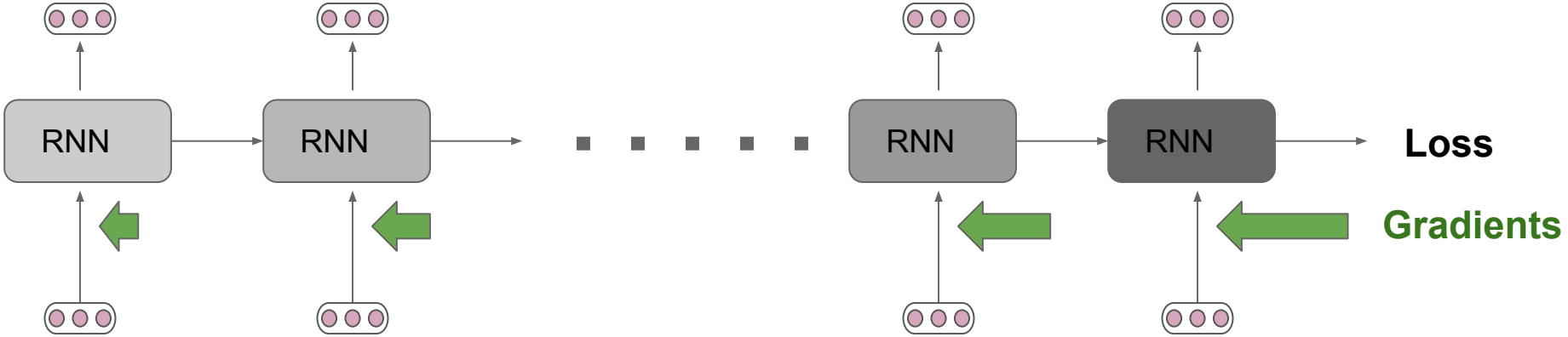
Ok! Looks like I can actually go! Book **the tickets** for next Wed!



Gradient vanishing / explosion



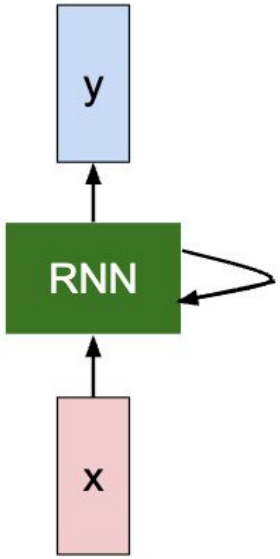
Gradient vanishing / explosion



What's wrong with RNNs?

- Long Range Dependencies
- Gradient vanishing / explosion
- Long time to converge
- Expensive computation

Recurrent Neural Networks - RNNs



$$h_t = f_W(h_{t-1}, x_t)$$

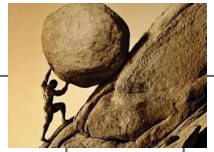
$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$

Sometimes called a “Vanilla RNN” or an “Elman RNN” after Prof. Jeffrey Elman

From CS231n slides. Lecture 8 by Fei-Fei Li, Yunzhu Li, Ruohan Gao

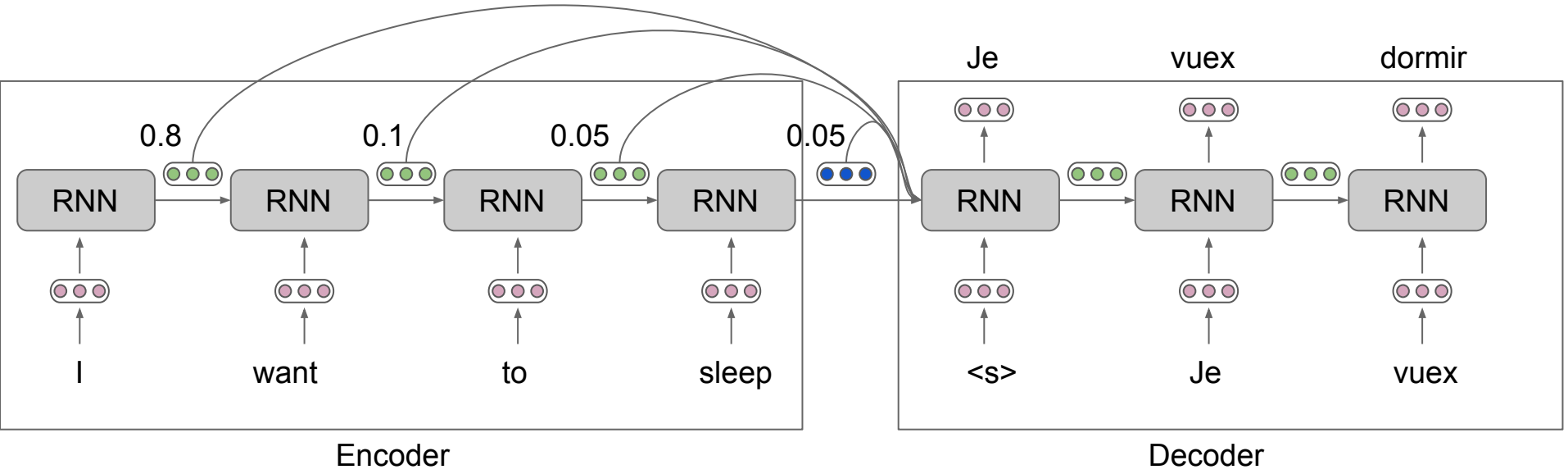
Encoder-Decoder Models



Typically Encoder and Decoder would be separate networks i.e. have their own separate weights

Ilya Sutskever, Oriol Vinyals, Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks.

Encoder-Decoder Models With Attention

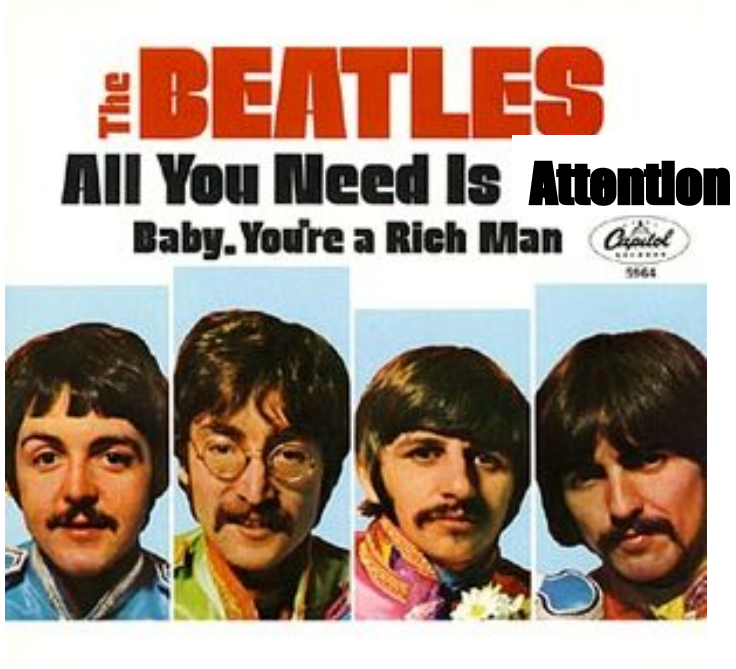


Neural Machine Translation by Jointly Learning to Align and Translate. Bahdanau et al, 2015

Encoder-Decoder Models With Attention

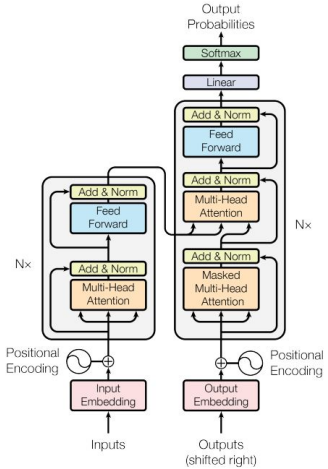
Decoder: $y_t = g_V(y_{t-1}, h_{t-1}, c)$
where context vector c is often $c = h_0$

Bahdanau et al, "Neural machine translation by jointly learning to align and translate", ICLR 2015

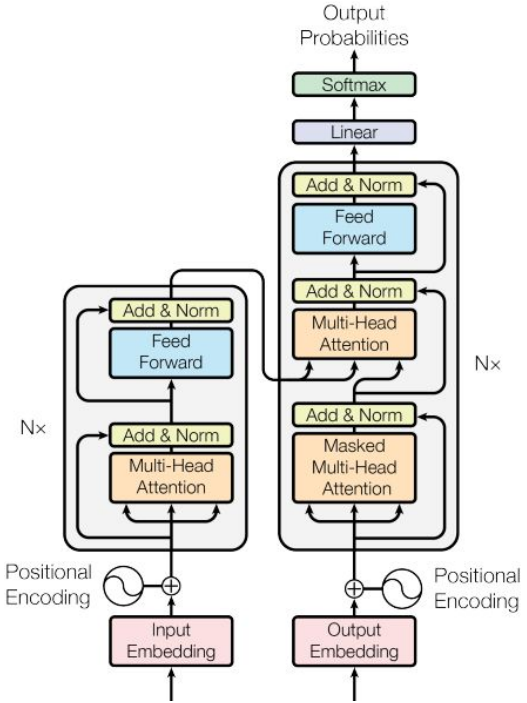


Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com



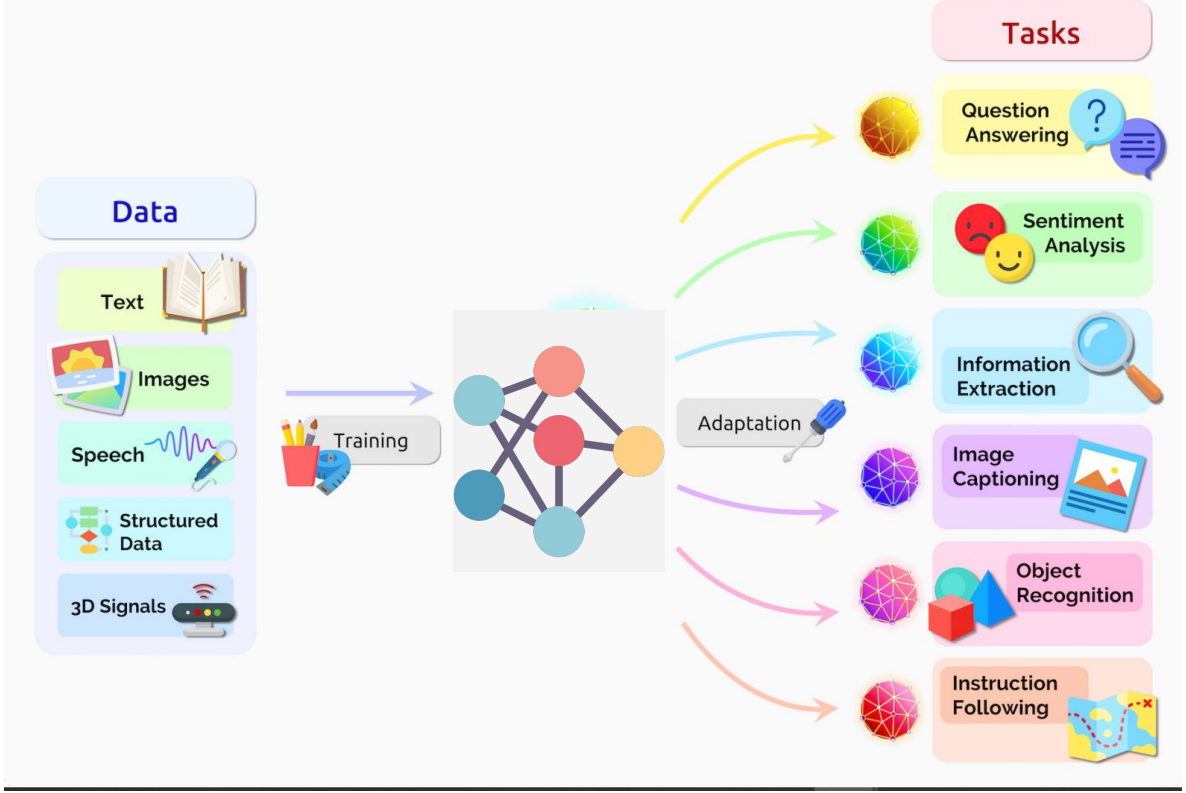
Transformer Model



Google Research

Attention is all you need (Vaswani et.al, 2017)

Wide Applications



https://blogs.nvidia.com/blog/2022/03/25/what-is-a-transformer-model/

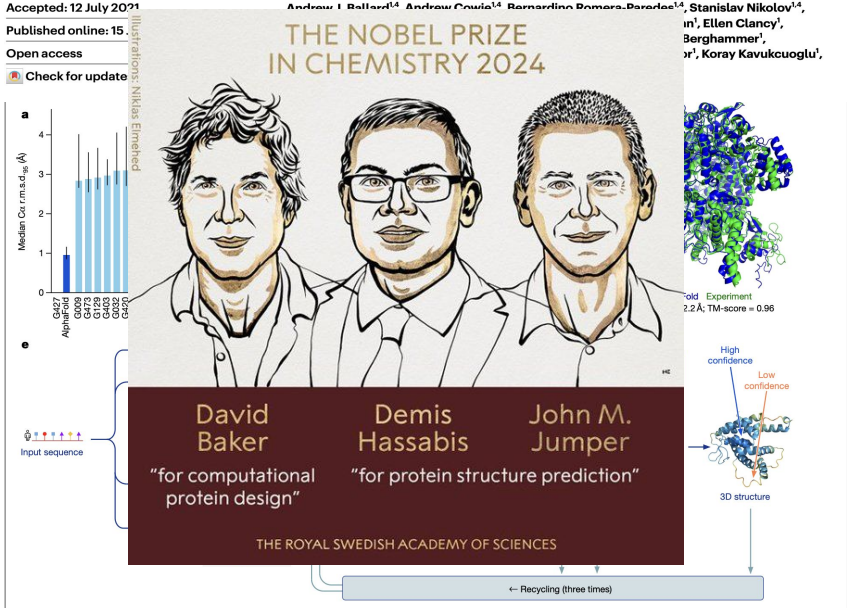
Real World Impact

Real World Impact

Highly accurate protein structure prediction with AlphaFold

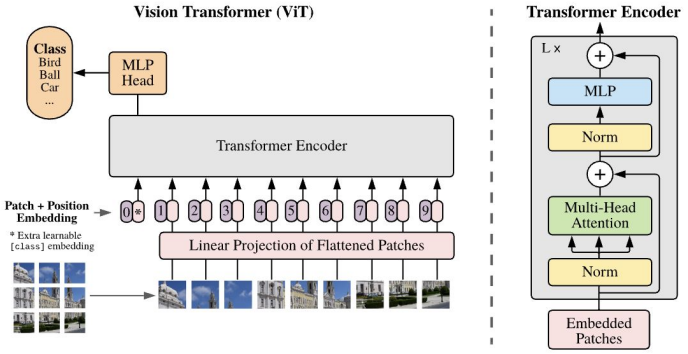
https://doi.org/10.1038/s41586-021-03819-2
Received: 11 May 2021
Accepted: 12 July 2021
Published online: 15 .
Open access
Check for update

John Jumper[1,4✉], Richard Evans[1,4], Alexander Pritzel[1,4], Tim Green[1,4], Michael Figurnov[1,4], Olaf Ronneberger[1,4], Kathryn Tunyasuvunakool[1,4], Russ Bates[1,4], Augustin Žídek[1,4], Anna Potapenko[1,4], Alex Bridgland[1,4], Clemens Meyer[1,4], Simon A. A. Kohl[1,4], Andrew J. Ballard[1,4], Andrew Cowie[1,4], Bernardino Romera-Paredes[1,4], Stanislav Nikolov[1,4], [illegible][1], Ellen Clancy[1], [illegible] Berghammer[1], [illegible][1], Koray Kavukcuoglu[1],



An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

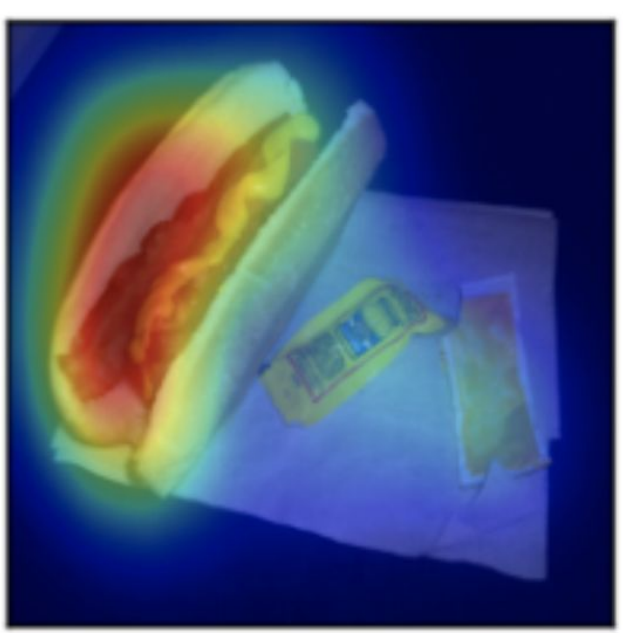
Alexey Dosovitskiy*,†, Lucas Beyer*, Alexander Kolesnikov*, Dirk Weissenborn*, Xiaohua Zhai*, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby*,†
*equal technical contribution, †equal advising
Google Research, Brain Team
{adosovitskiy, neilhoulsby}@google.com



Questions?

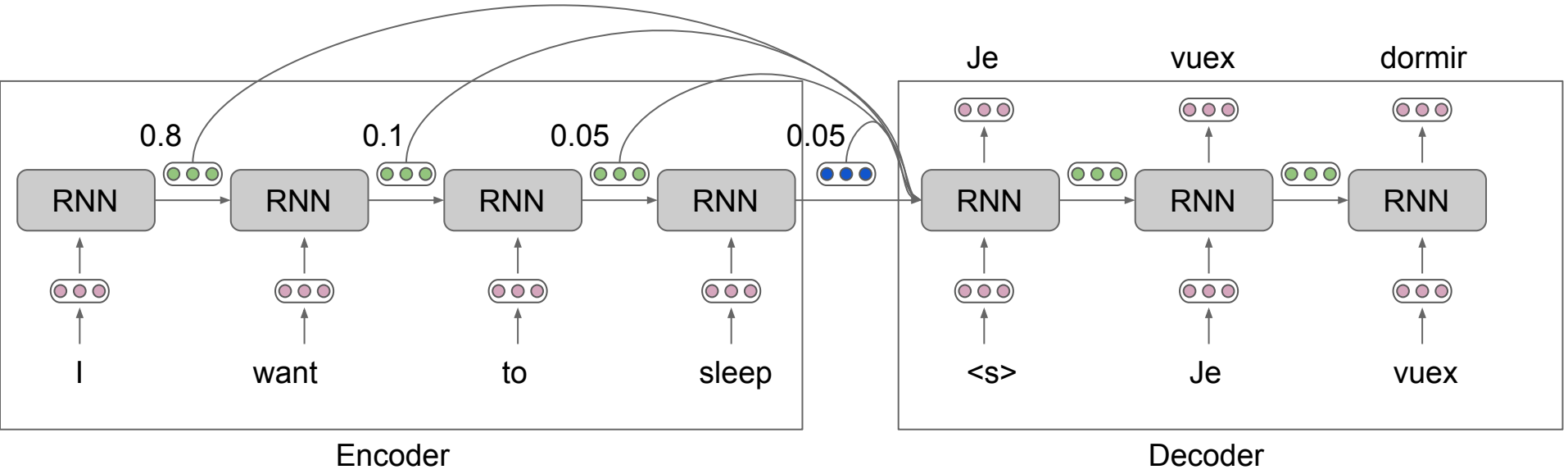
Visual Attention

What toppings are on the hot dog?



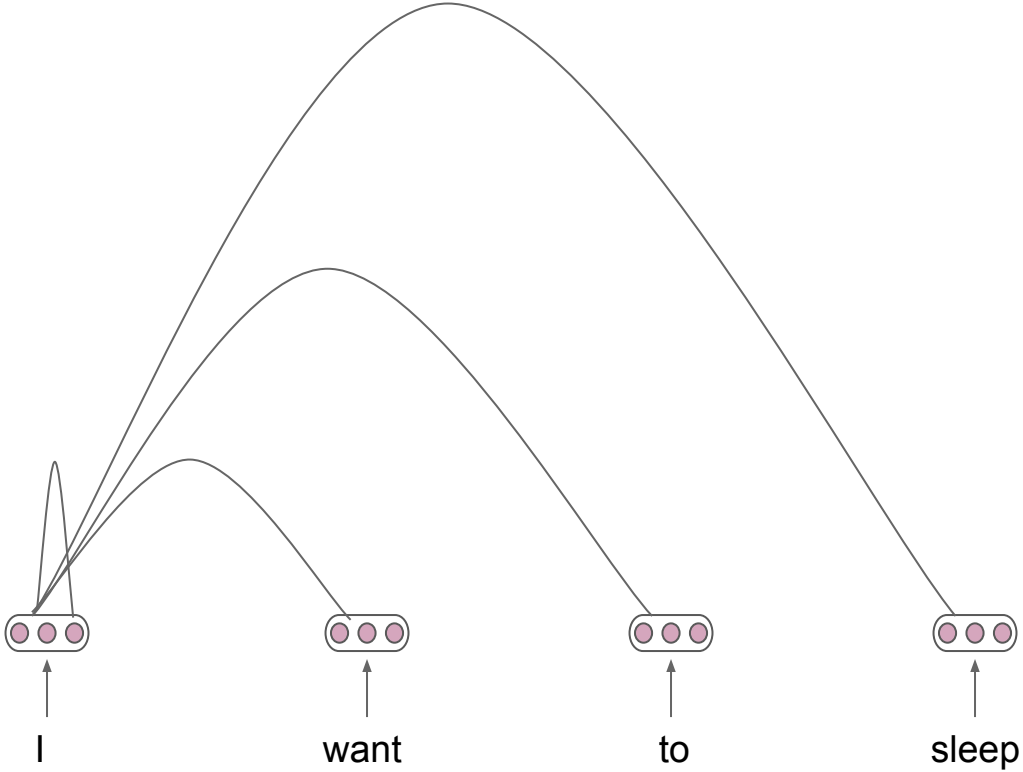
Differential Attention for Visual Question Answering (Patro et.al, 2018)

Last look at RNNs

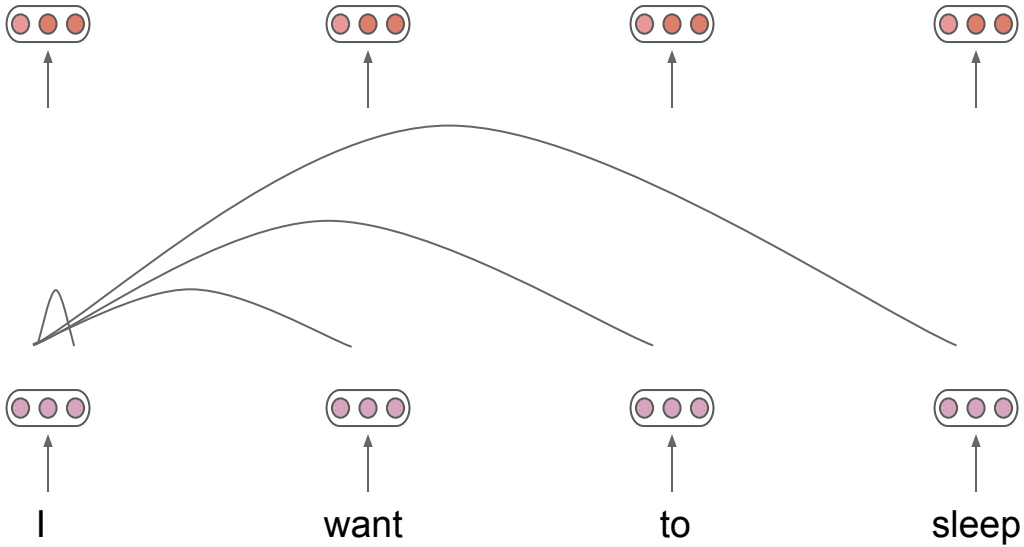


Neural Machine Translation by Jointly Learning to Align and Translate. Bahdanau et al, 2015

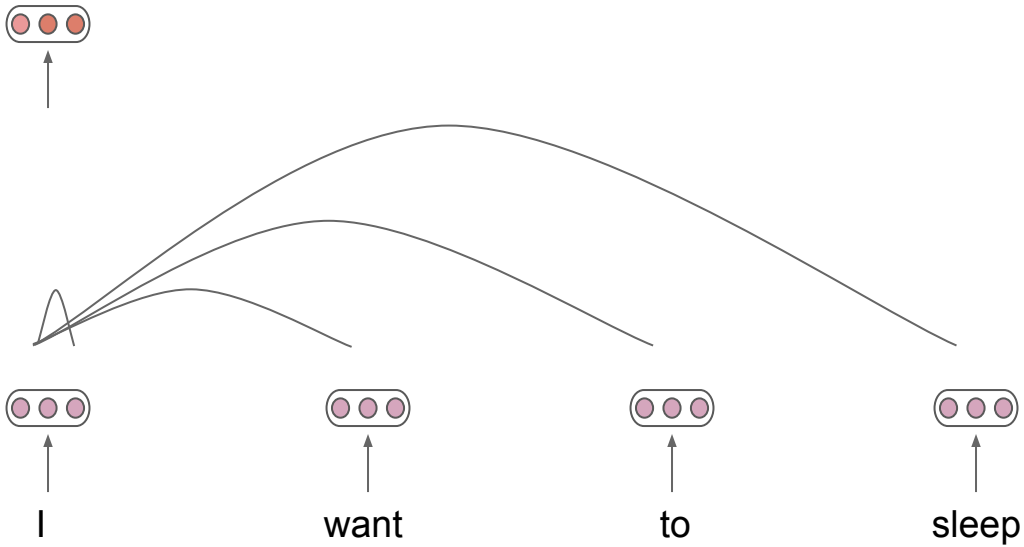
Self Attention



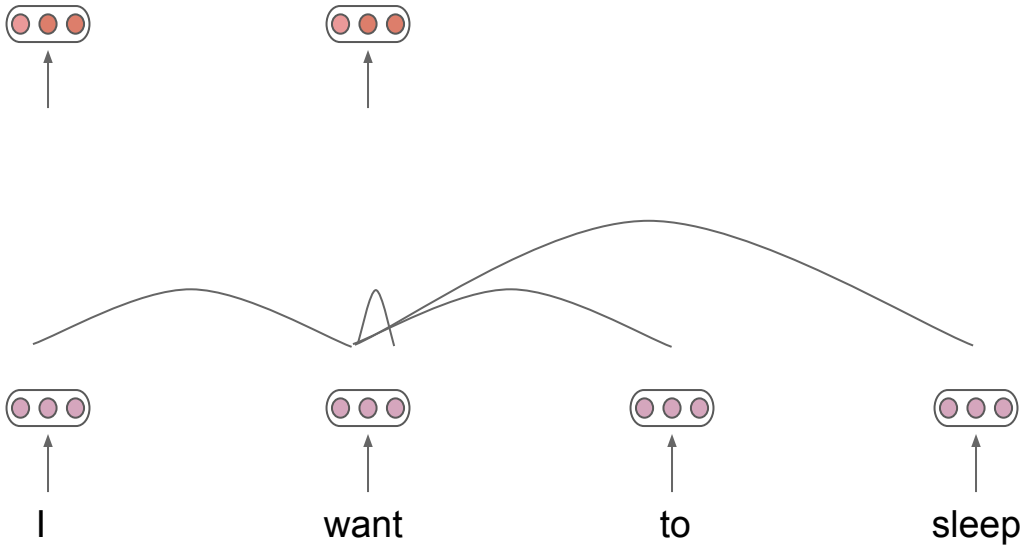
Self Attention - No more recurrence



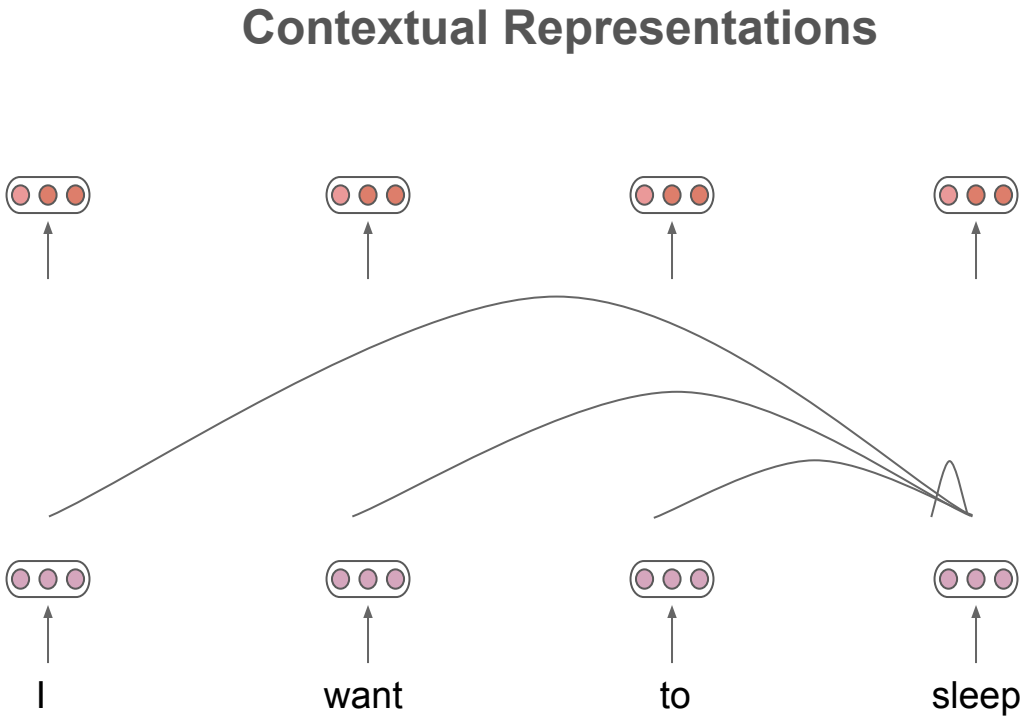
Self Attention - No more recurrence



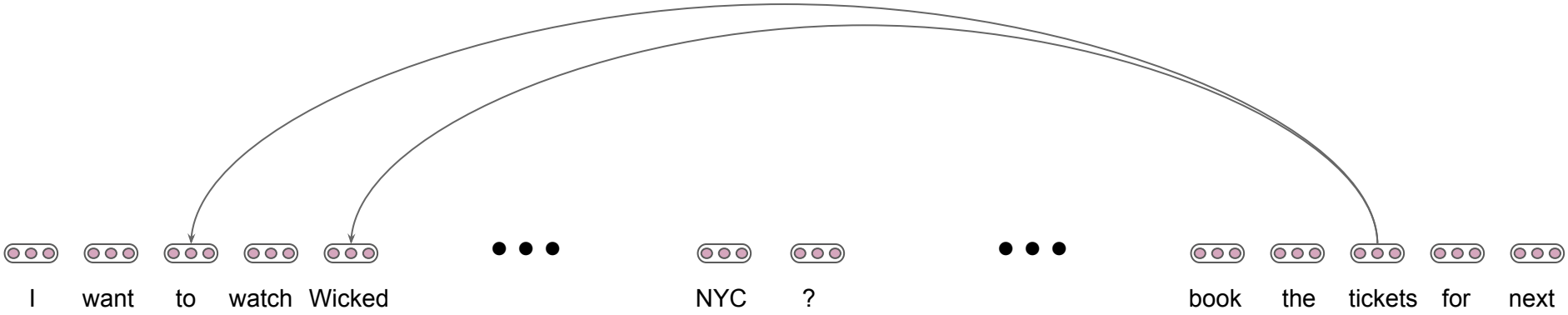
Self Attention - No more recurrence



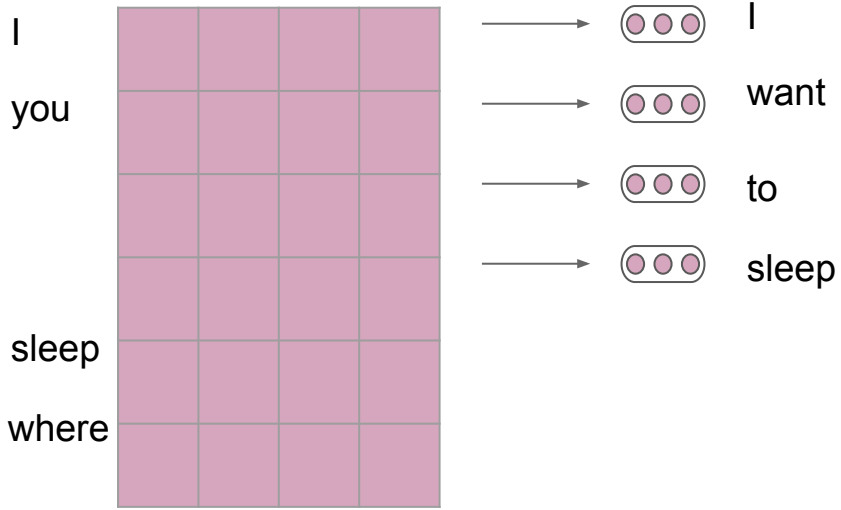
Self Attention - No more recurrence

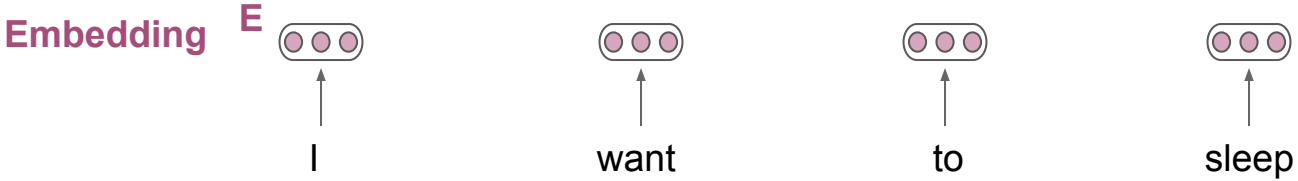


Self Attention for long sequences

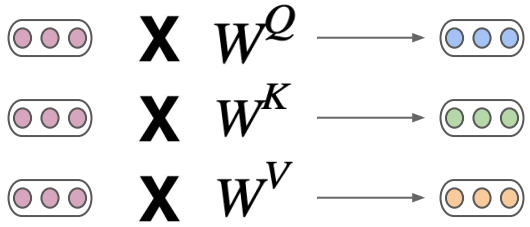


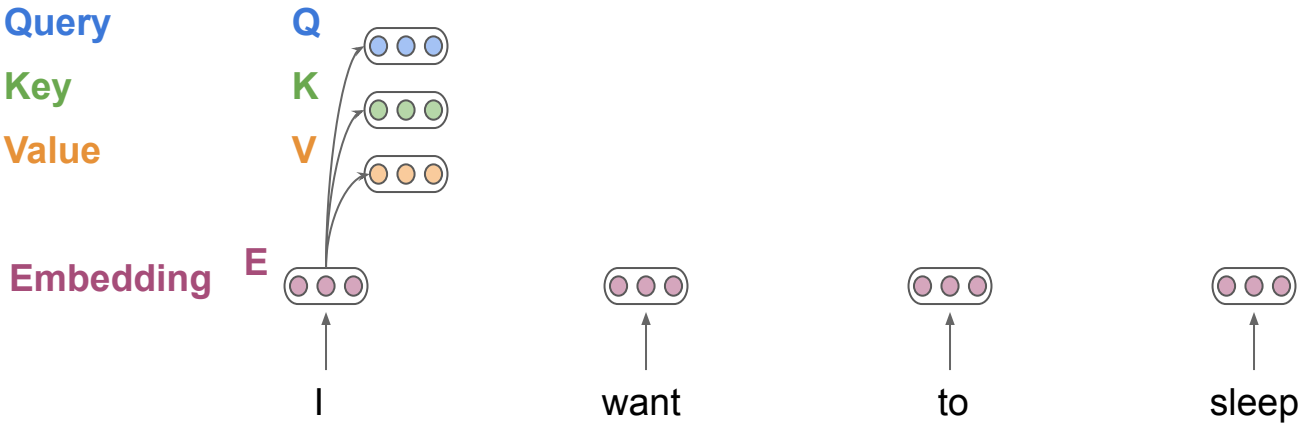
Self Attention - Word Embedding



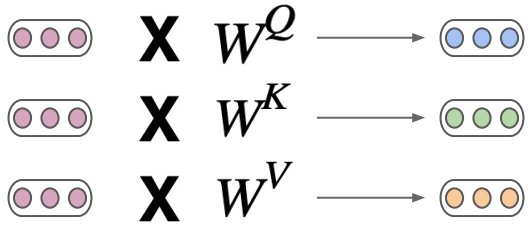


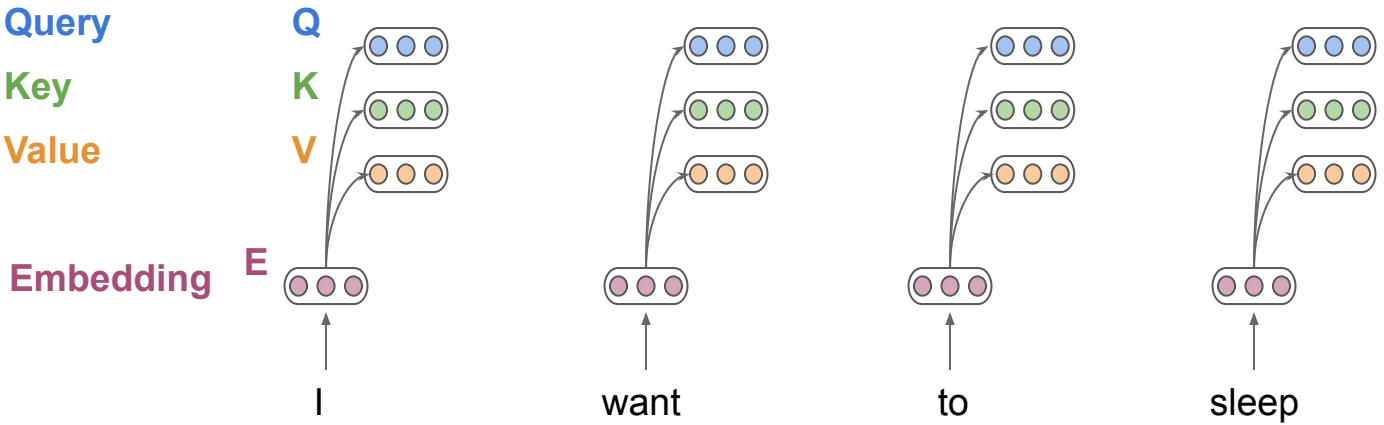
Self Attention - Projection Layer



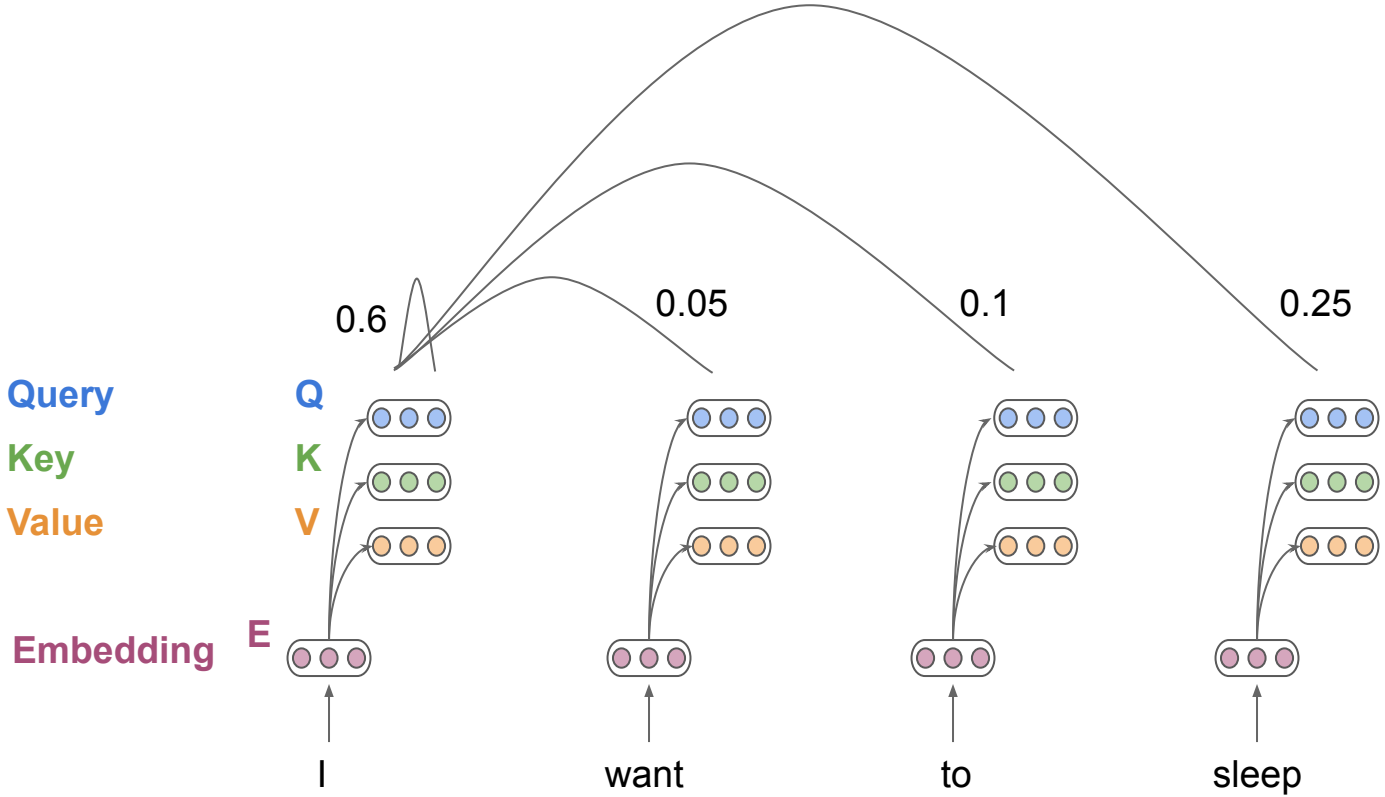


Self Attention - Projection Layer

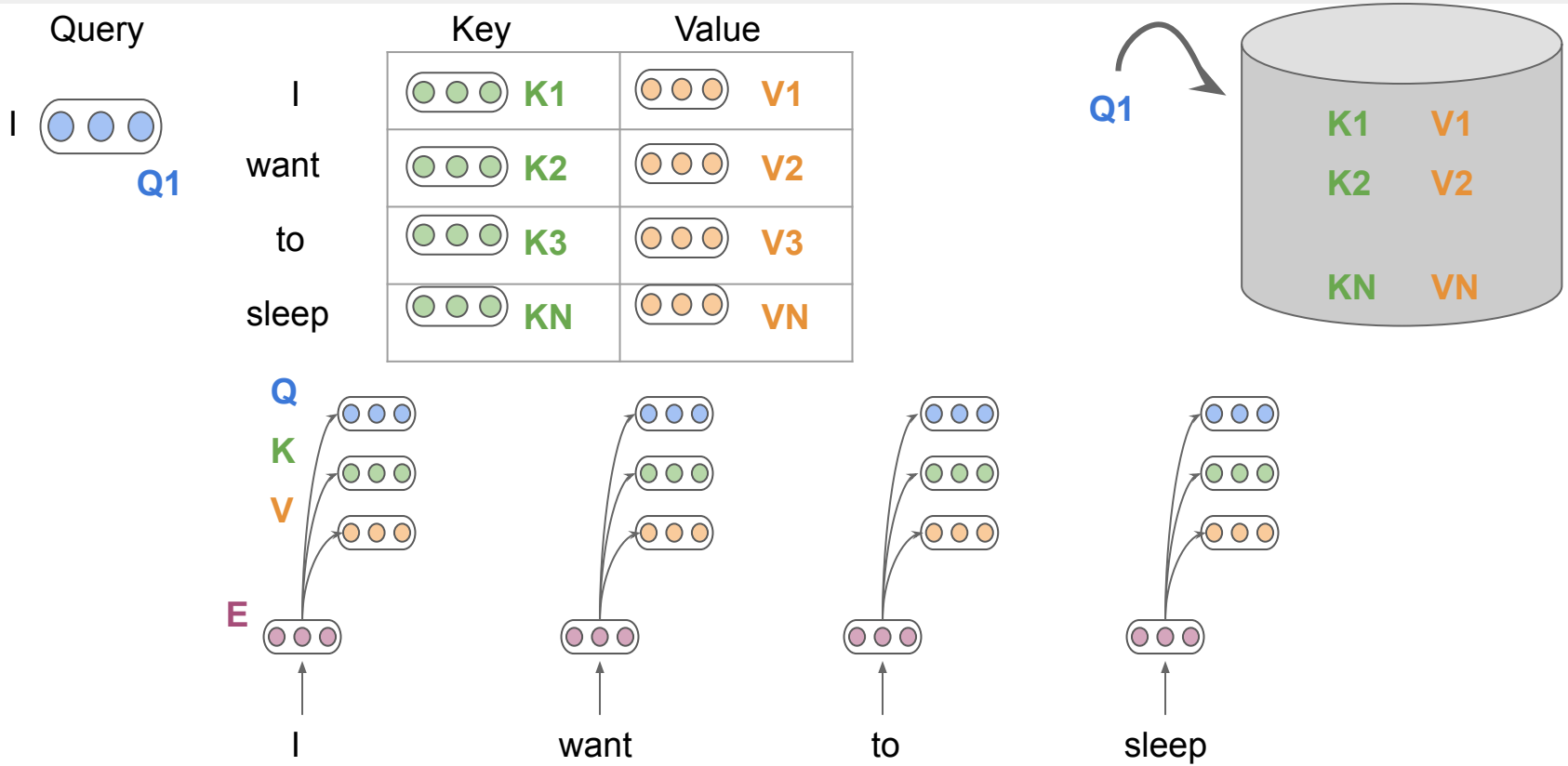




Self Attention - Attention Scores

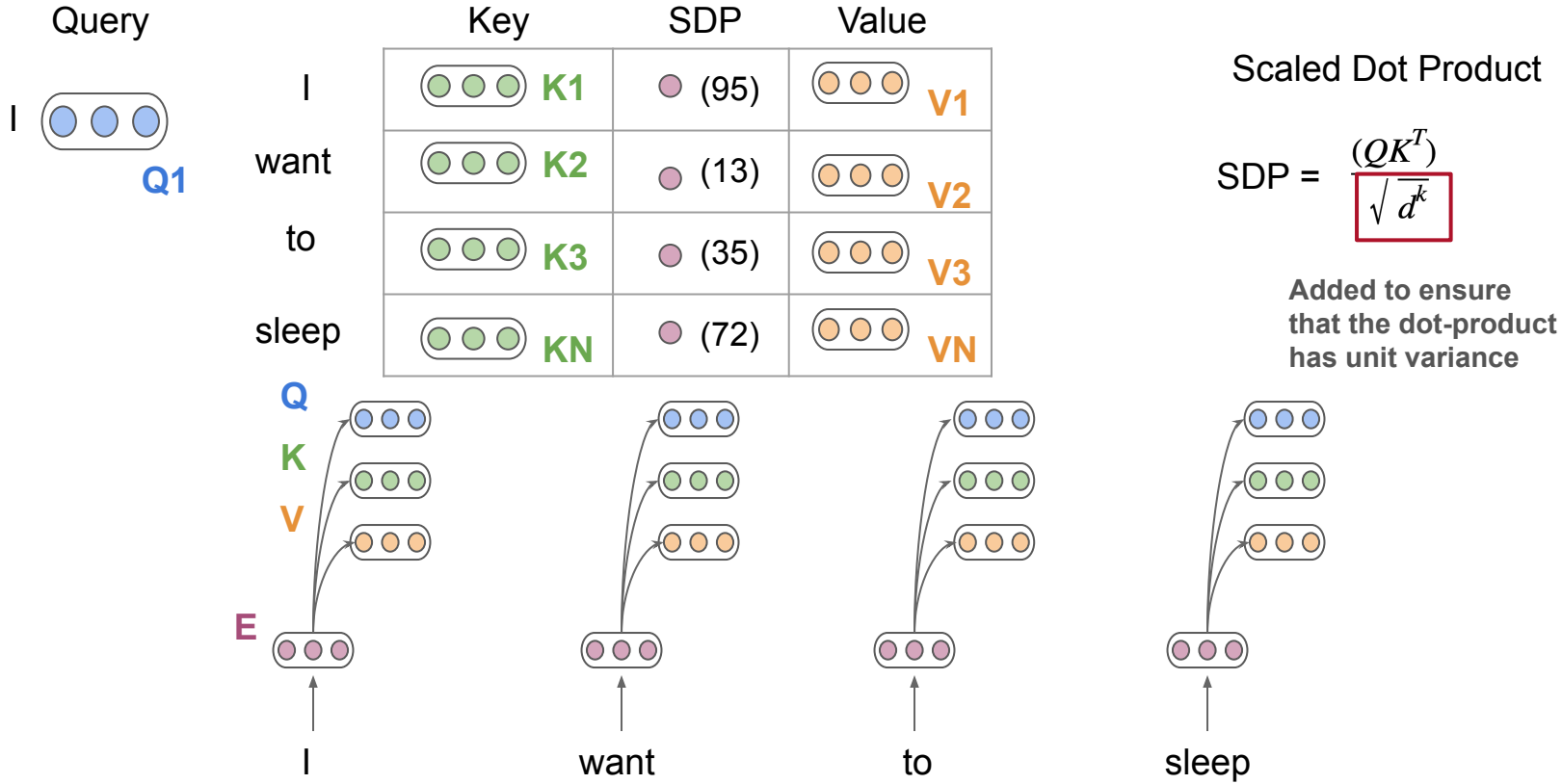


Self Attention



Questions?

Self Attention - Scaled Dot Product

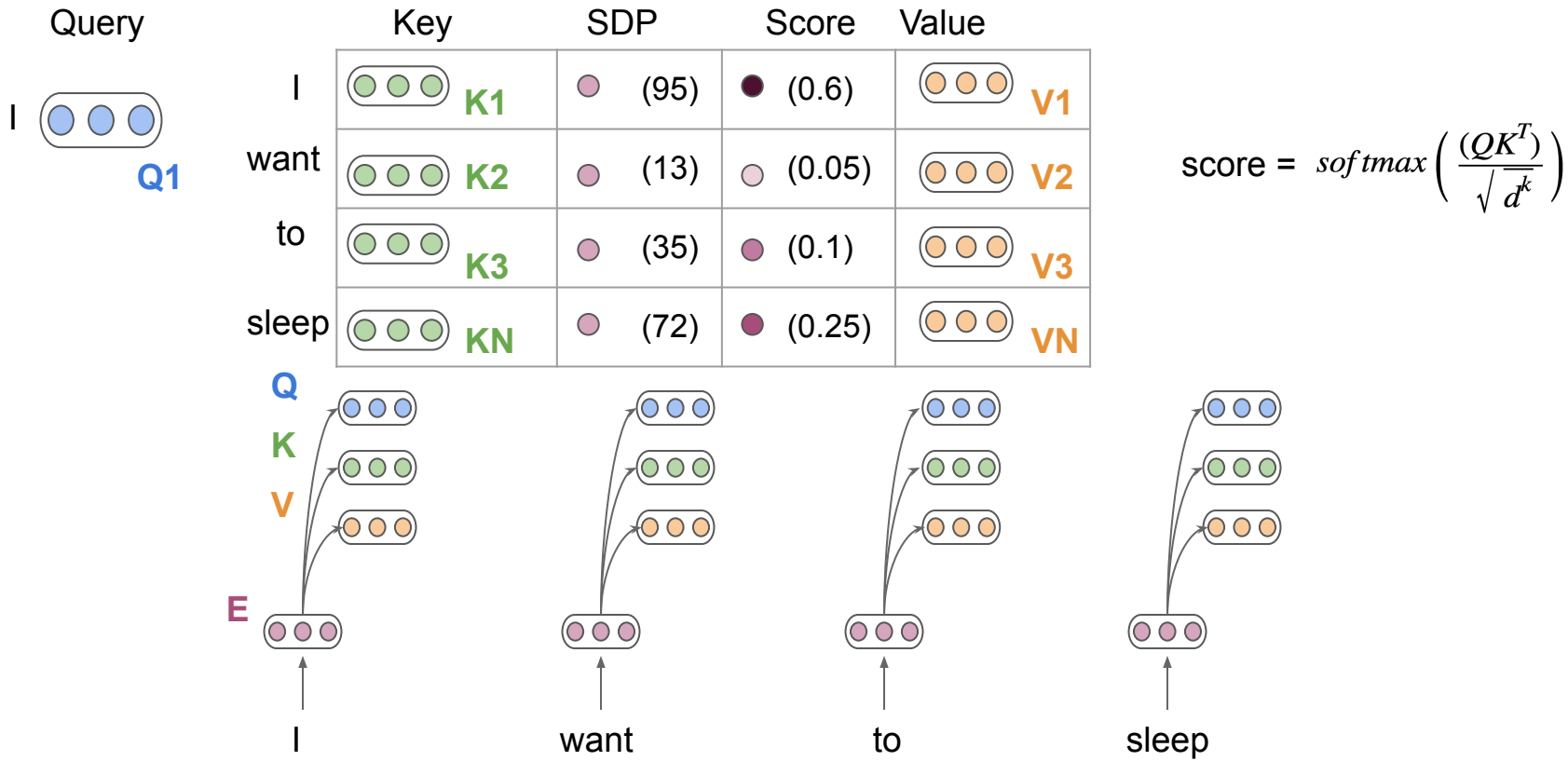


Scaled Dot Product

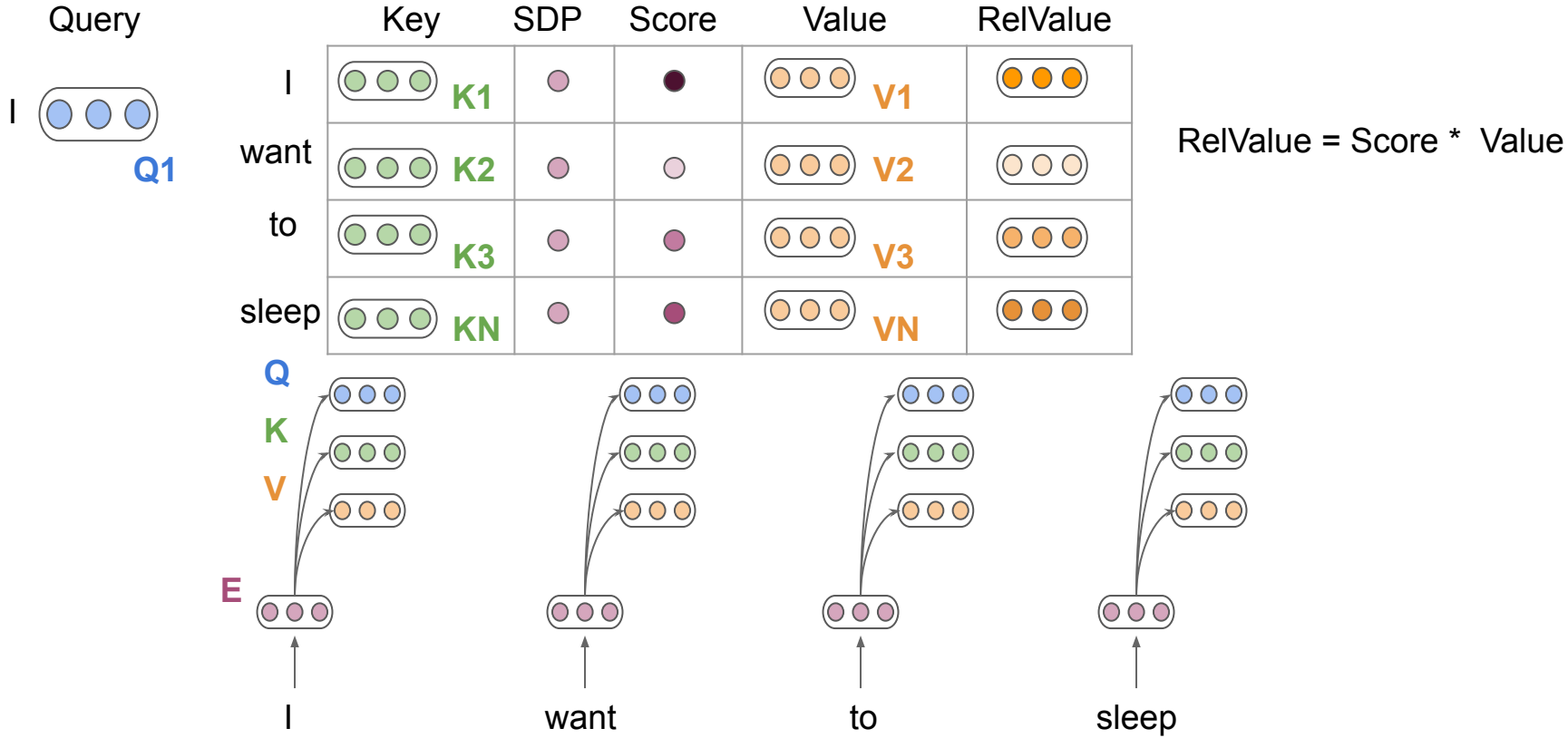
$$\text{SDP} = \frac{(QK^T)}{\sqrt{d^k}}$$

Added to ensure that the dot-product has unit variance

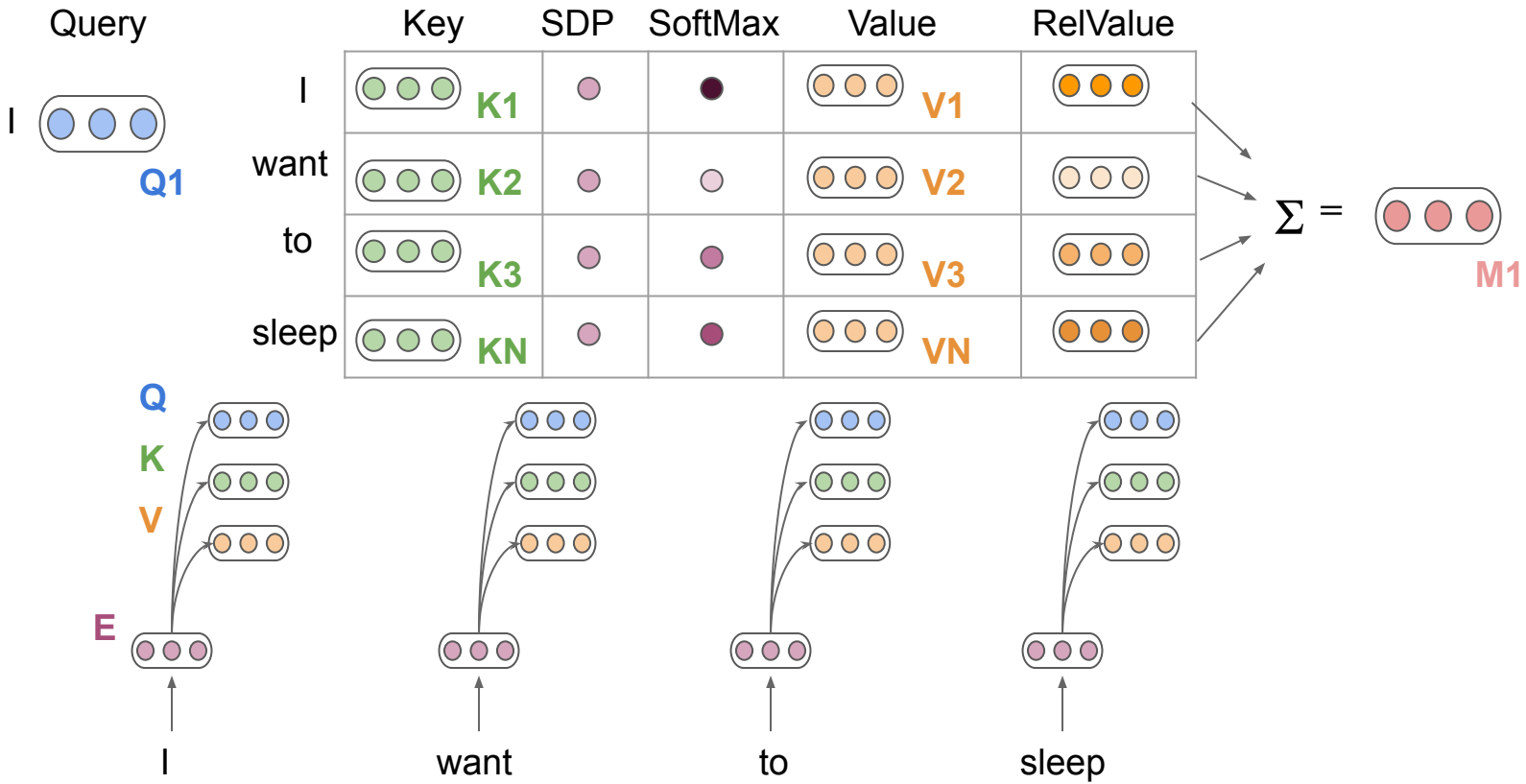
Self Attention - SoftMax



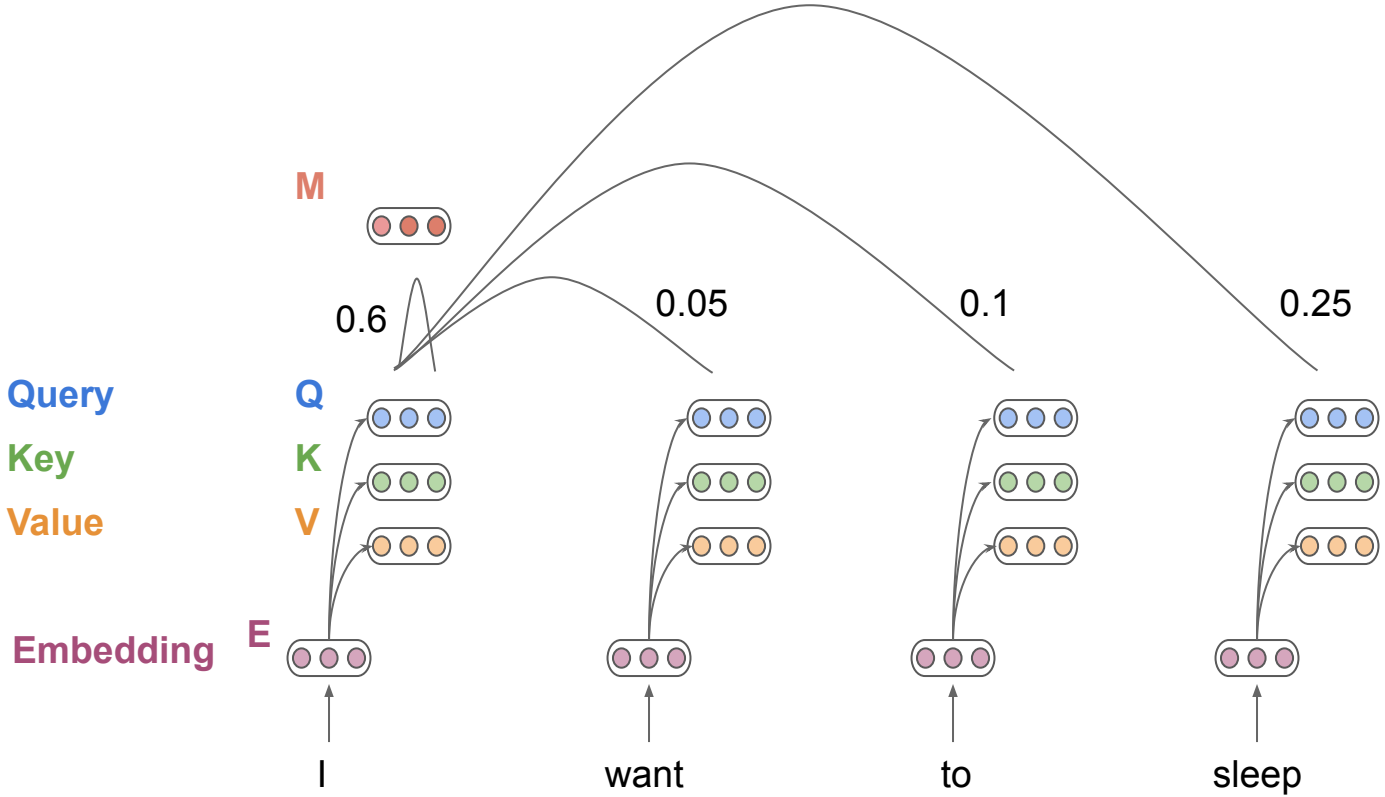
Self Attention - Soft (Relative) Values



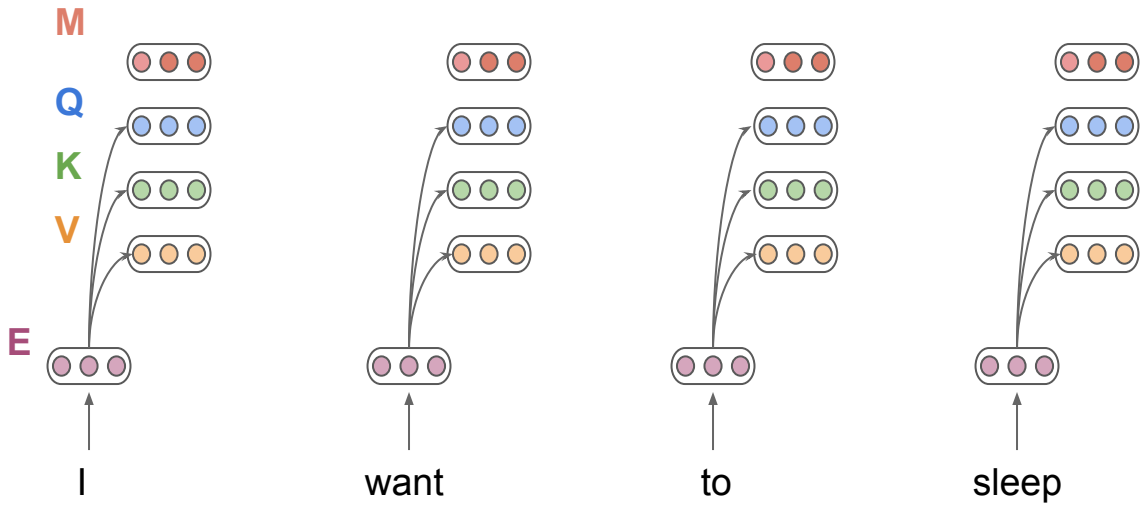
Self Attention - Attended Repr



Self Attention - Attended Contextual Rep



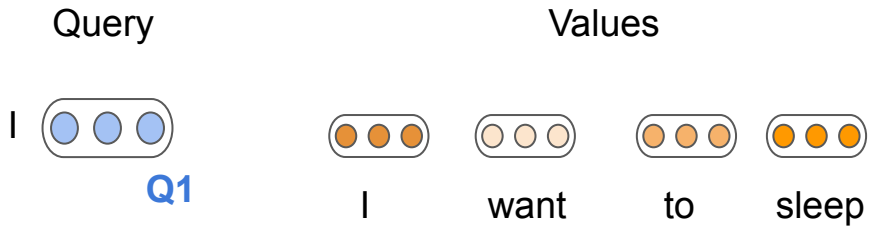
Self Attention - Attended Contextual Rep



Questions?

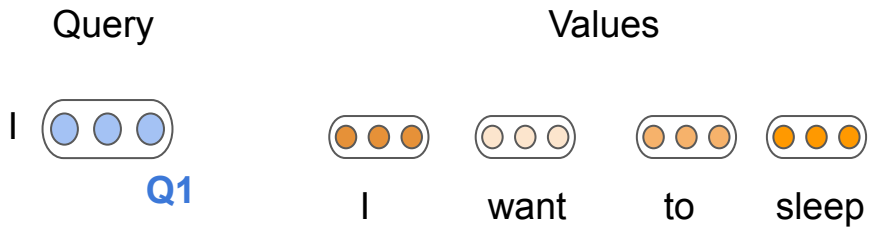
Problem with Self Attention

- Self Attention can focus heavily on the same word!



Problem with Self Attention

- Self Attention can focus heavily on the same word!



- Single representation

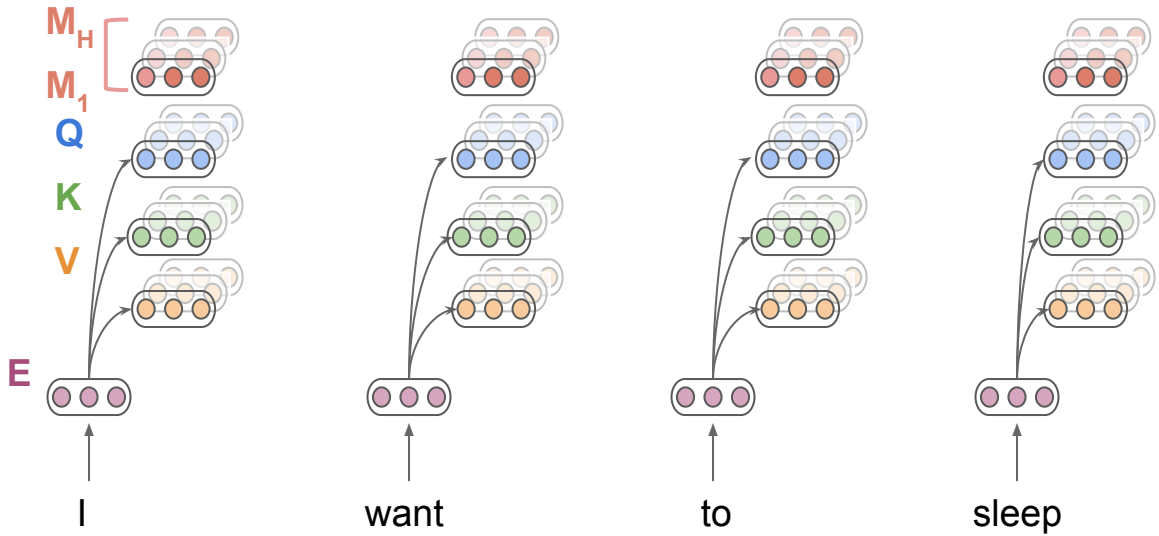
I like **Harry** **Potter** v/s I like **Harry** **Potter**

(Book) (Movie)

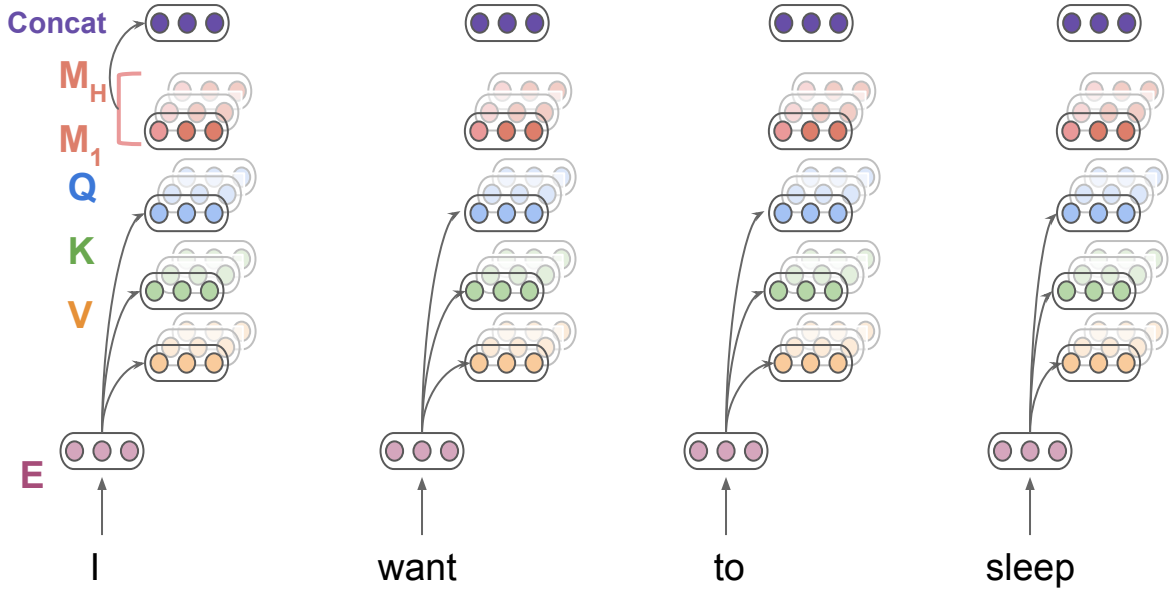
Multi-Headed Self Attention

Love me some representation power!

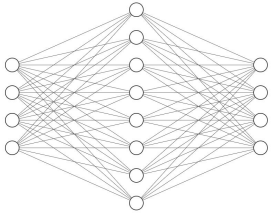
H (no: of heads) Different versions of Q,K,V
Each different repr -> Different attended repr



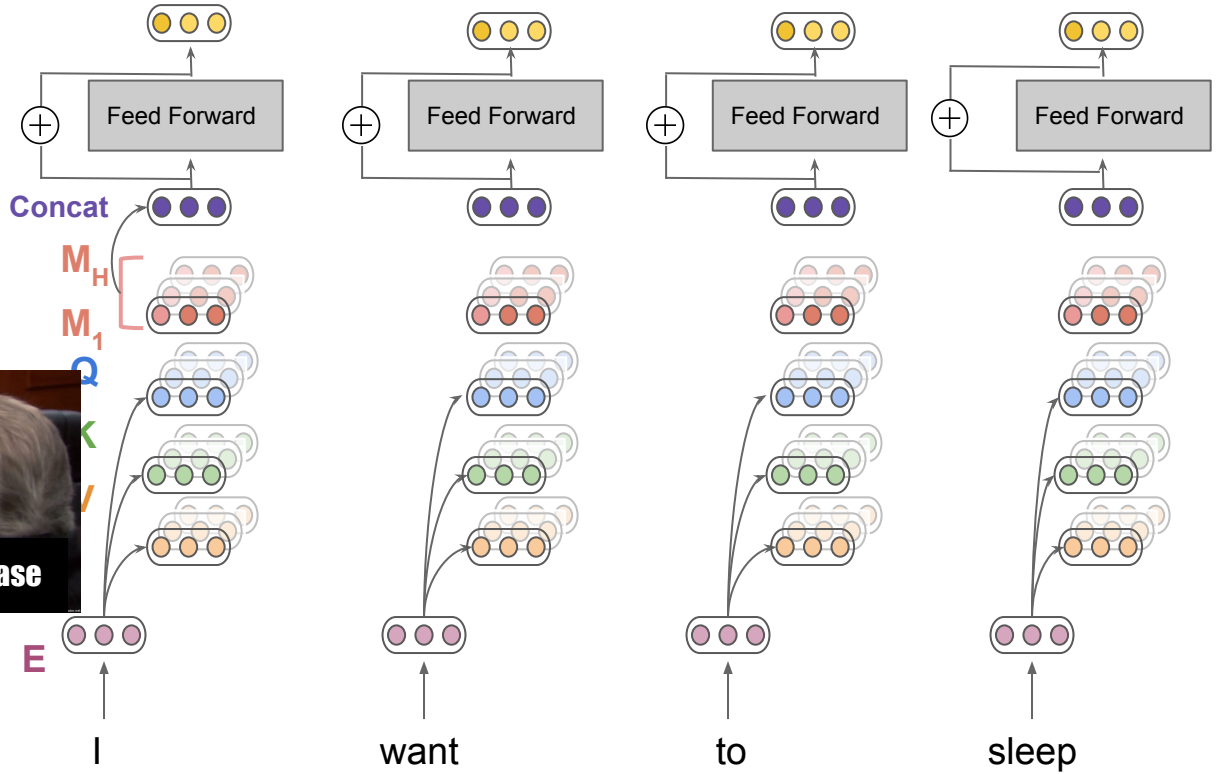
Multi-Headed Self Attention



Multi-Headed Self Attention



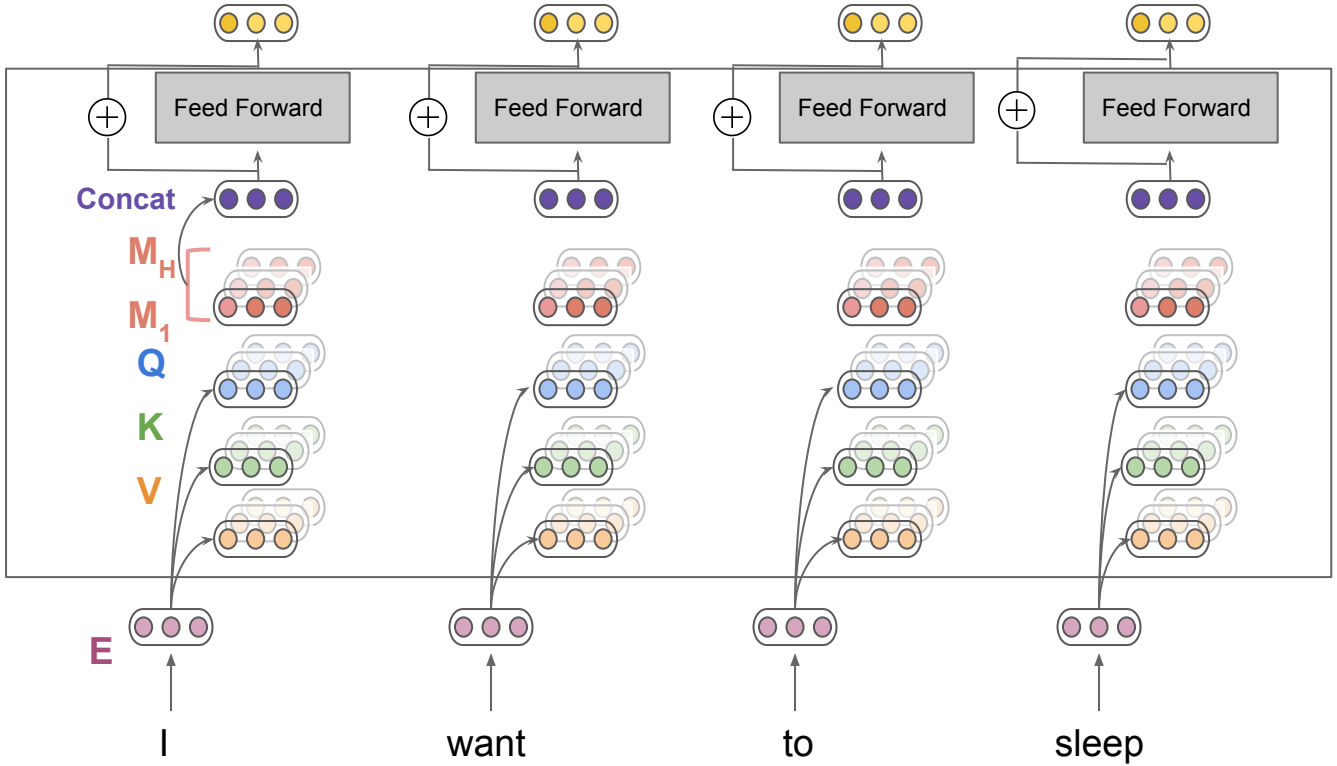
It's Me MLP



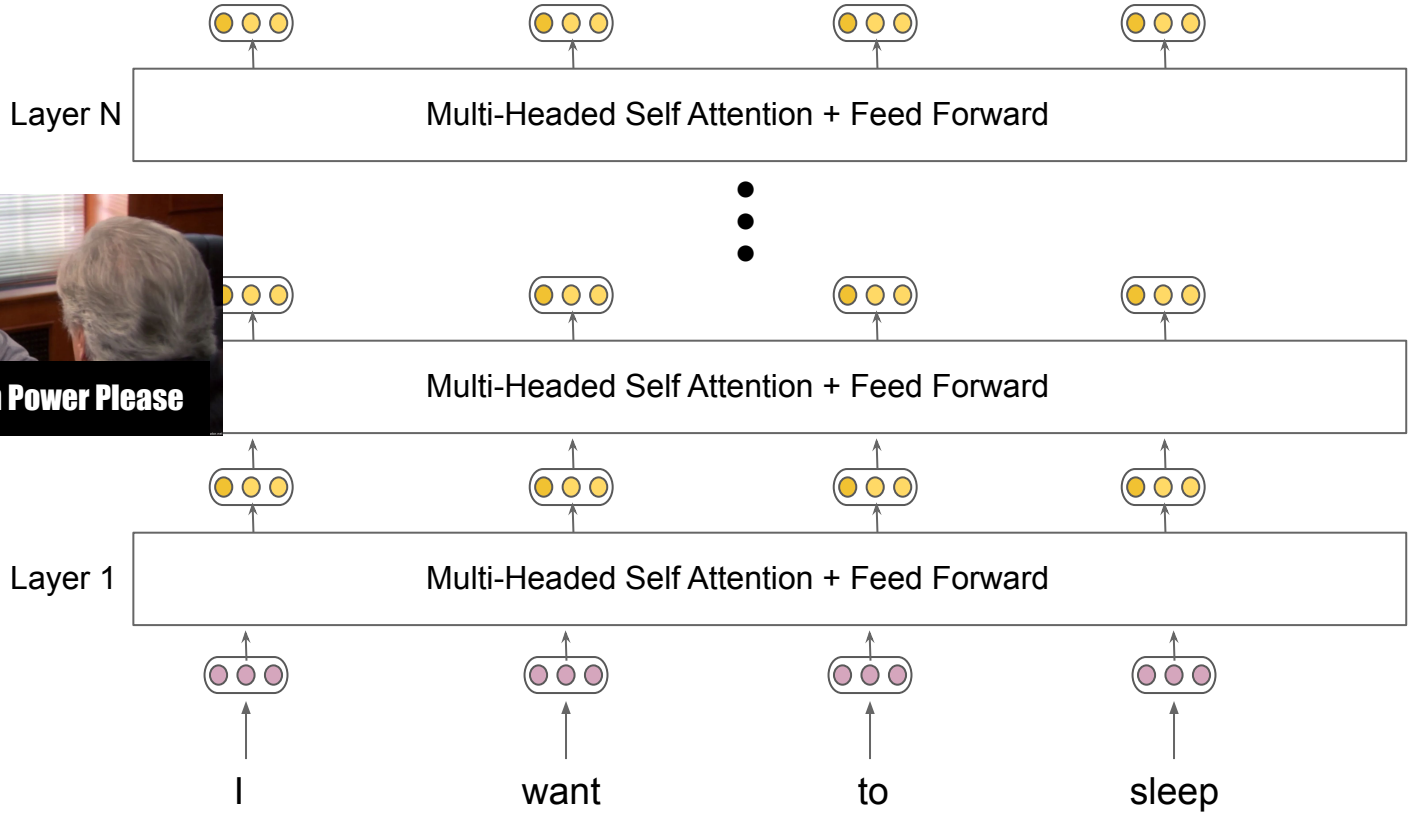
Residual Connections : Help with cleaner gradient flows during back-prop

Multi-Headed Self Attention

Multi-Headed Self Attention + Feed Forward

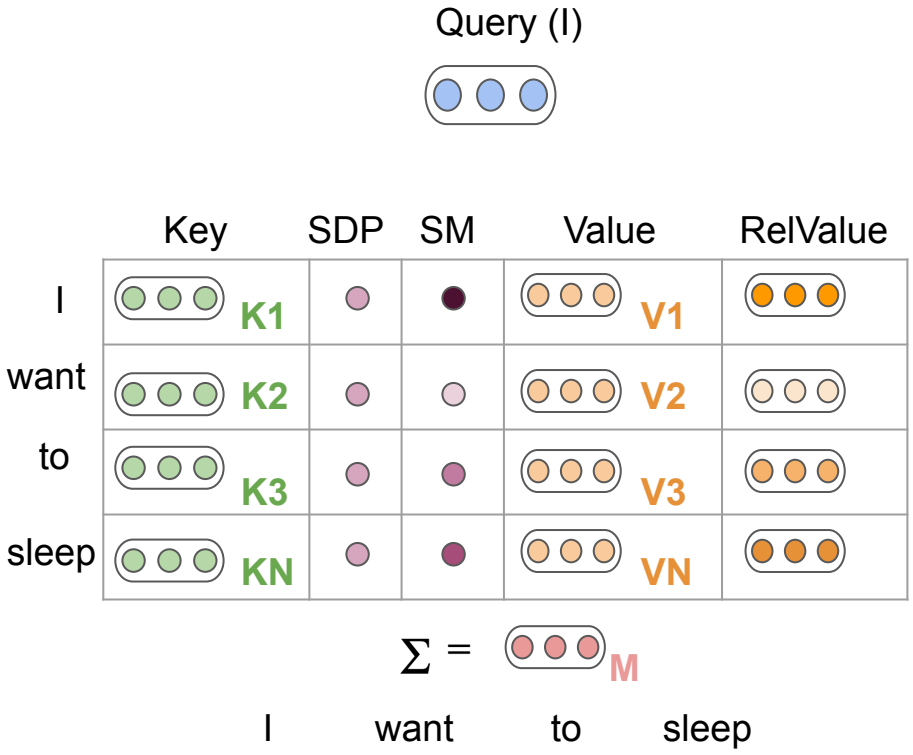


Multi-Headed Self Attention

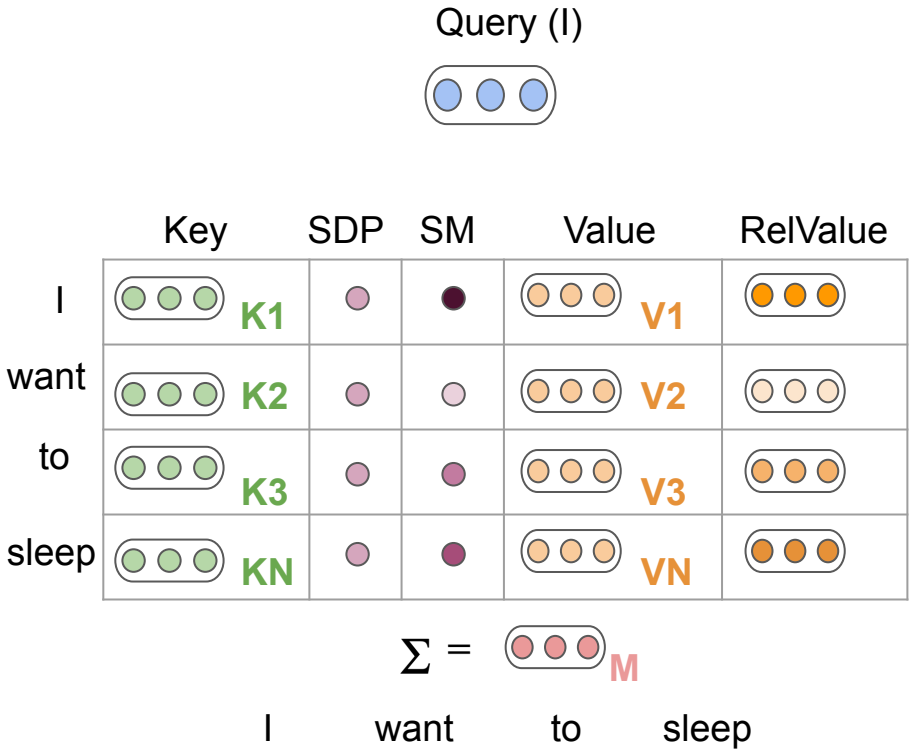


Questions?

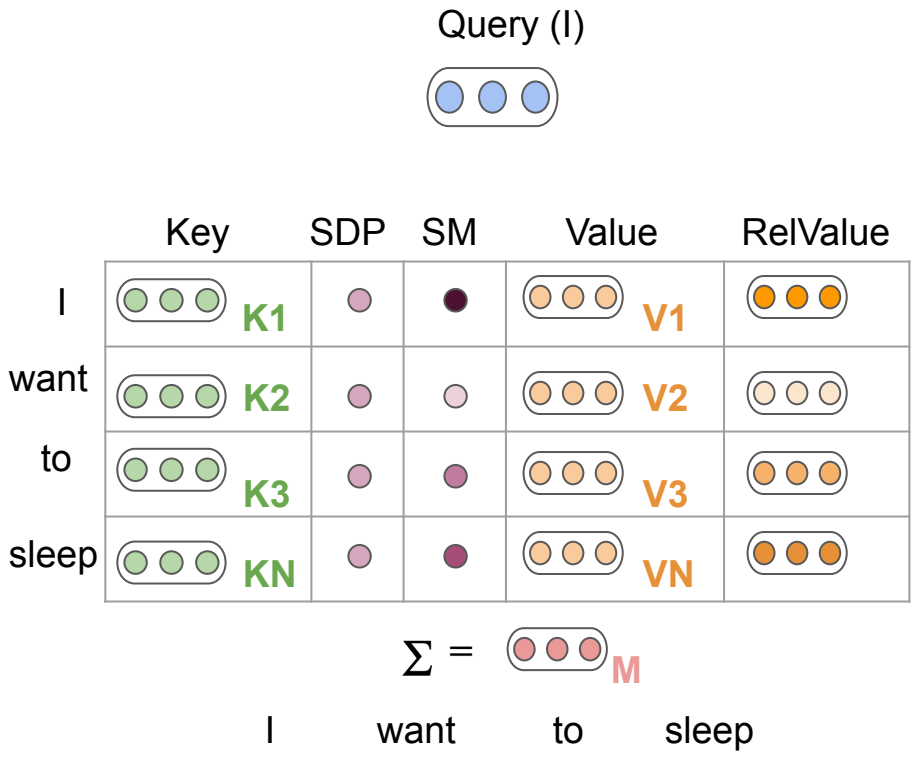
Revisiting Self Attention

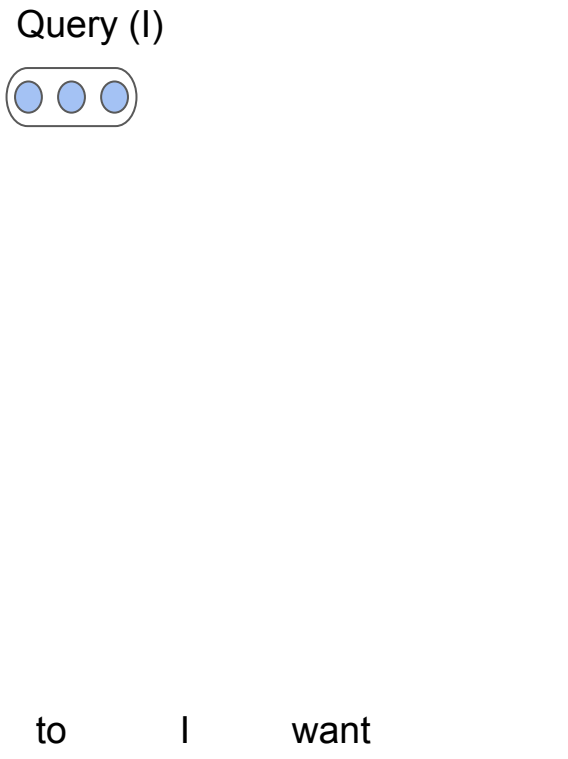


Revisiting Self Attention

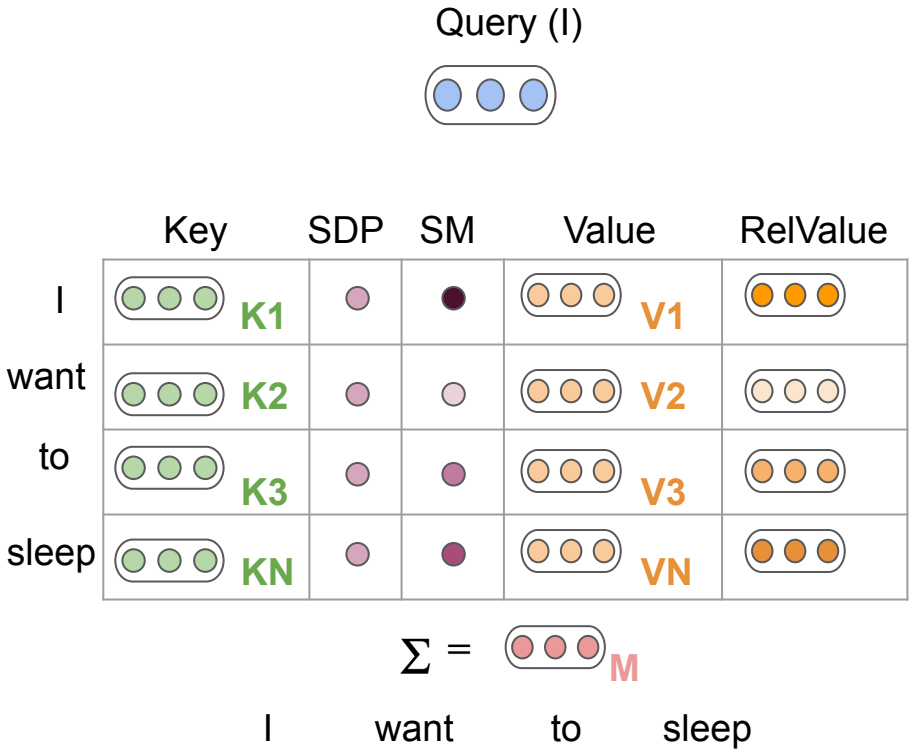


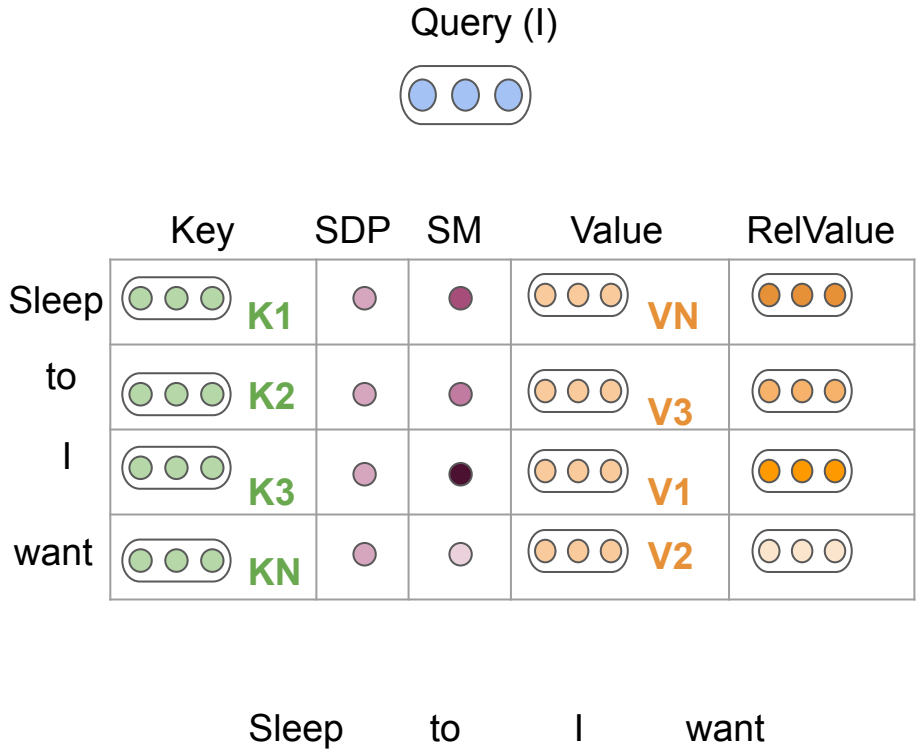
Revisiting Self Attention



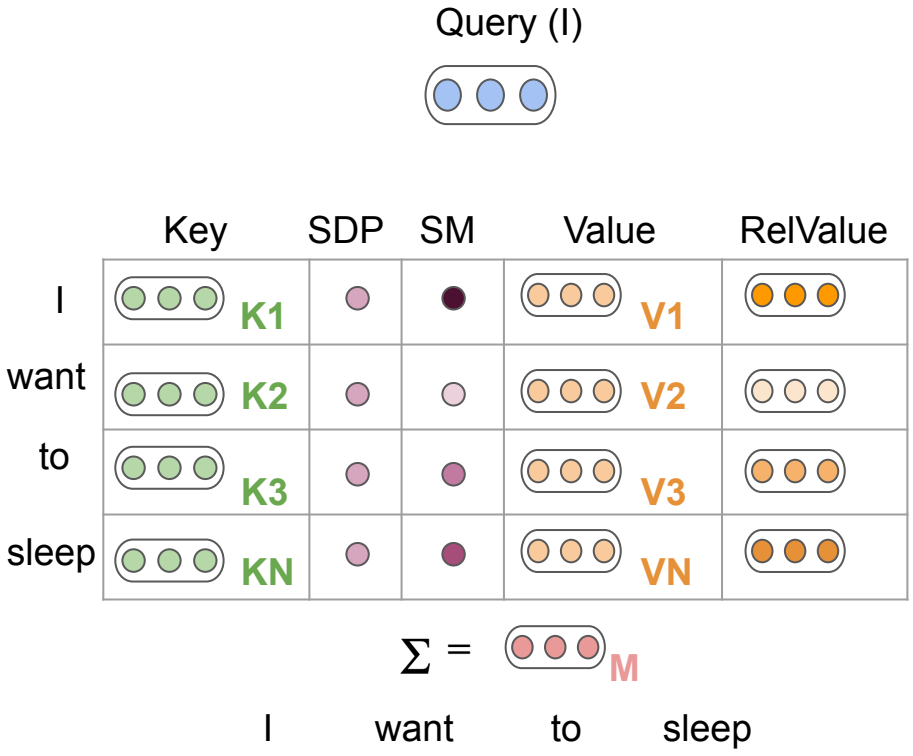


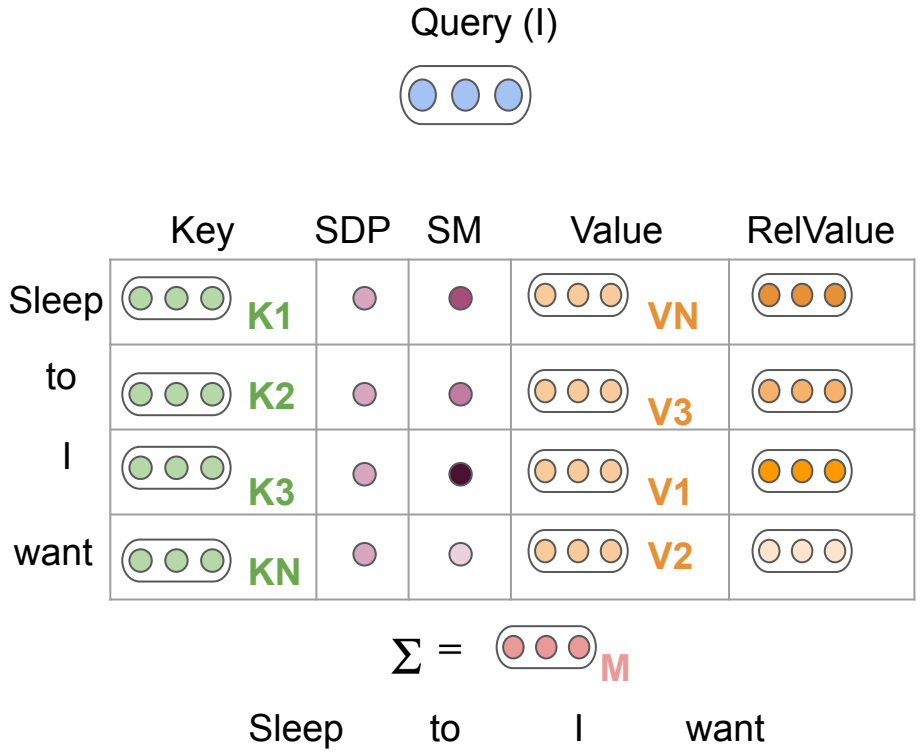
Revisiting Self Attention





Revisiting Self Attention



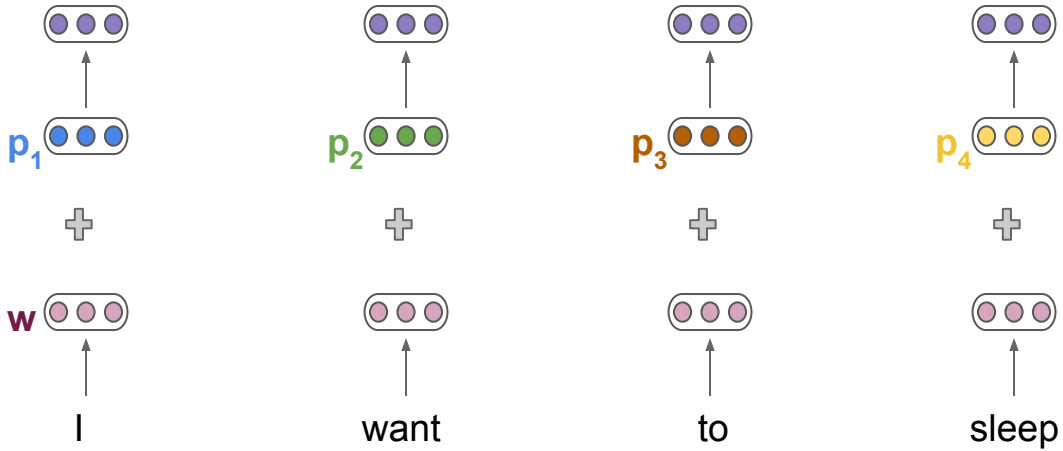


Revisiting Self Attention

Same representation for both sentences - But positions matter!

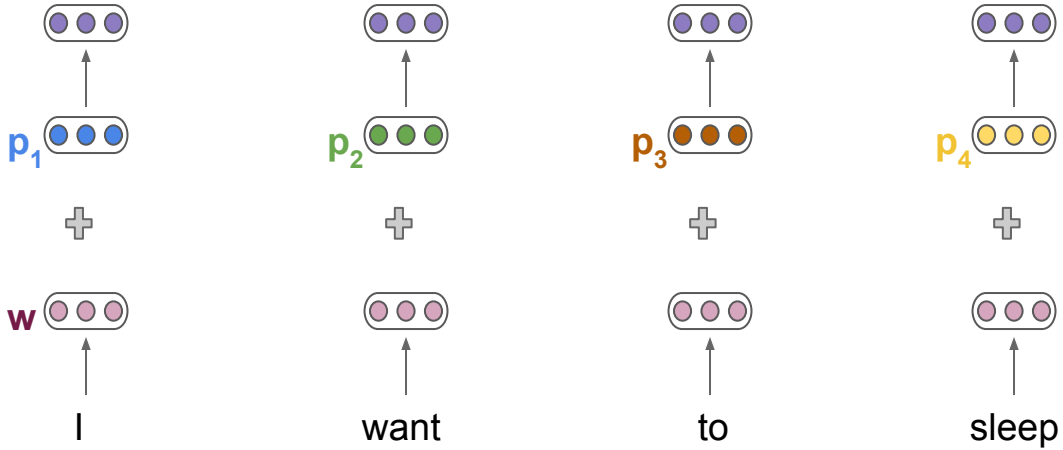
Positional Encoding

Position embeddings - each position number has an associated embedding



Positional Encoding

$$p = f(i, t)$$



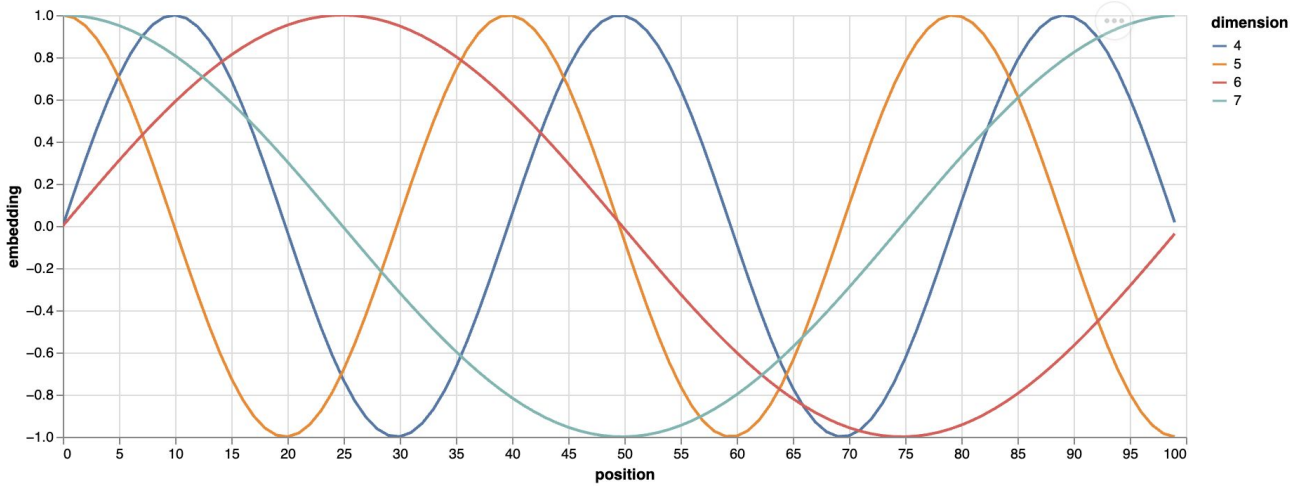
Learnable Positional Encoding

- Simplest type of positional encodings
- Initialize position encodings as random vectors for each position from 0 to MAX LENGTH
- Used in GPT-1, GPT-2, GPT-3
- CONS: Doesn't generalize to sequences of length greater than MAX LENGTH

Sinusoidal Positional Encoding

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{\text{model}}})$$

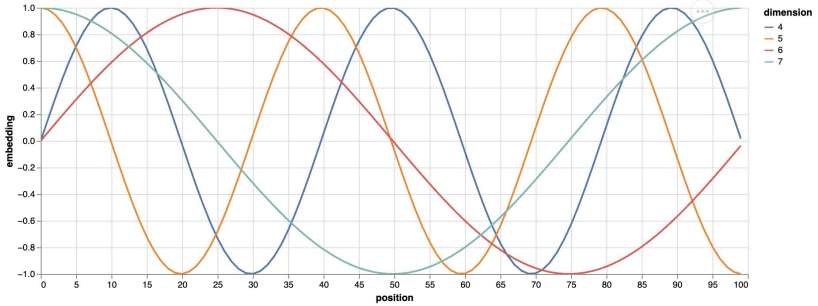


Sinusoidal Positional Encoding

- Introduced in the original Transformers paper and was used in models like BERT
- Major promise of this embeddings over using learnable encodings was that these might lead to length generalization beyond maximum length seen during training
- Alas, that's not the case and sinusoidal embeddings are usually as bad as learnable embeddings when it comes to length generalization

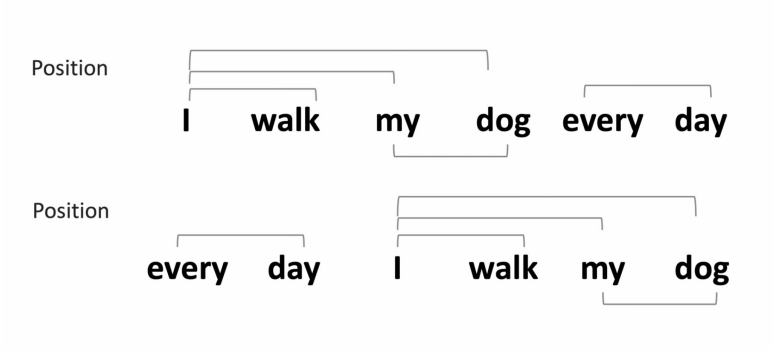
$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{\text{model}}})$$



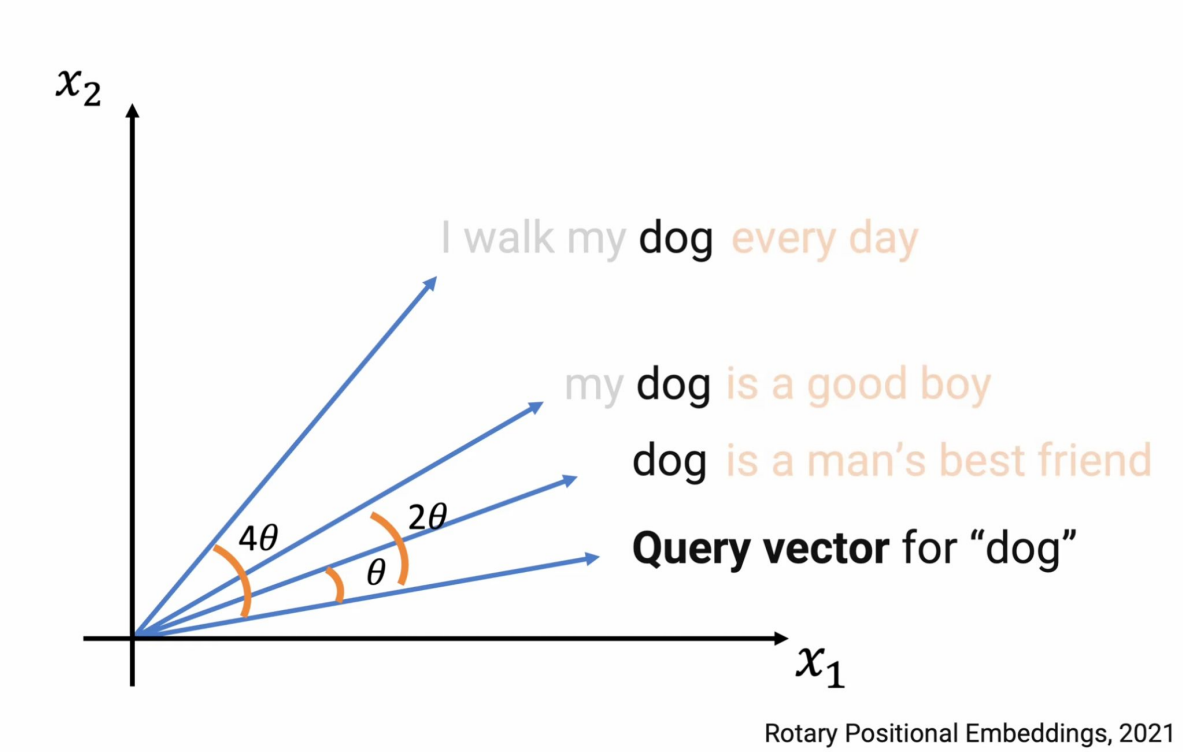
Relative Position Encodings

- Both learnable and sinusoidal position encodings were examples of Absolute Position Encodings
- i.e. positional information depends on the absolute position of the token
- In relative position encodings we provide the positional information in form of the relative difference between token position
- This information is added while taking the dot product between the query and key vectors



From Jia-Bin Huan's Youtube Video: https://www.youtube.com/watch?v=SMBkImDWOyQ&t=683s

Rotary Positonal Encodings (RoPE)



From Jia-Bin Huan's Youtube Video: https://www.youtube.com/watch?v=SMBkImDWOyQ&t=683s

Rotary Positonal Encodings (RoPE)

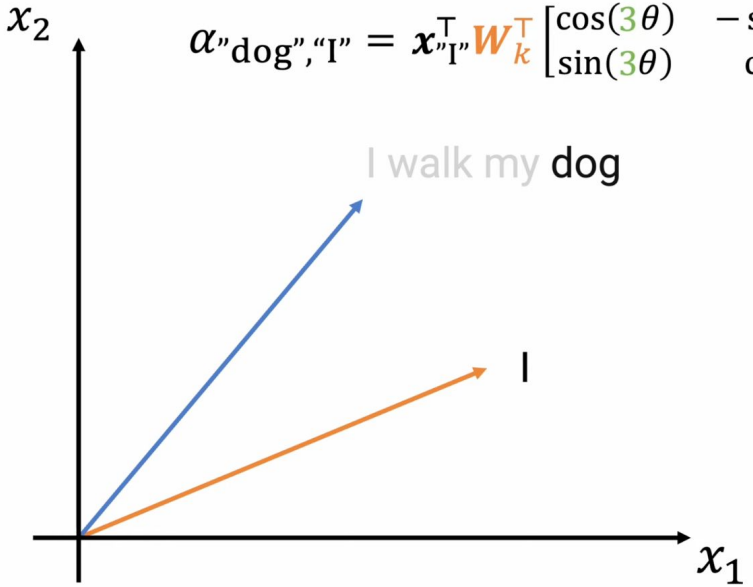
$$q_{\text{"dog"}} = \begin{bmatrix} \cos(4\theta) & -\sin(4\theta) \\ \sin(4\theta) & \cos(4\theta) \end{bmatrix} W_q x_{\text{"dog"}}$$

Query vector for "dog"

$$k_{\text{"I"}} = \begin{bmatrix} \cos(1\theta) & -\sin(1\theta) \\ \sin(1\theta) & \cos(1\theta) \end{bmatrix} W_k x_{\text{"I"}}$$

Key vector for "I"

$$\alpha_{\text{"dog","I"}} = x_{\text{"I"}}^{\top} W_k^{\top} \begin{bmatrix} \cos(3\theta) & -\sin(3\theta) \\ \sin(3\theta) & \cos(3\theta) \end{bmatrix} W_q x_{\text{"dog"}}$$



From Jia-Bin Huan's Youtube Video: https://www.youtube.com/watch?v=SMBkImDWOyQ&t=683s

Rotary Positonal Encodings (RoPE)

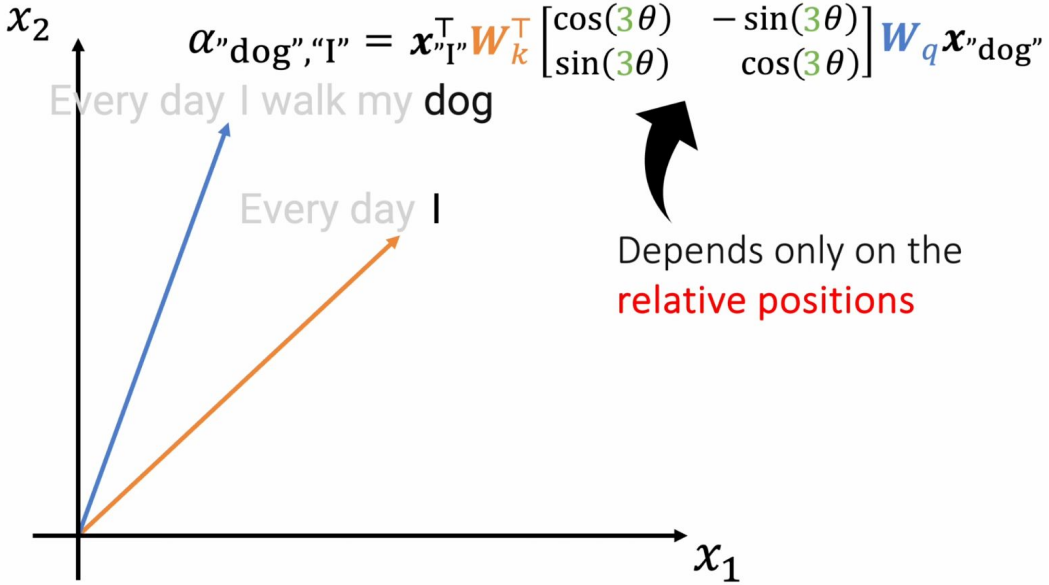
$$q_{\text{"dog"}} = \begin{bmatrix}\cos(6\theta) & -\sin(6\theta)\\ \sin(6\theta) & \cos(6\theta)\end{bmatrix} W_q \boldsymbol{x}_{\text{"dog"}}$$

Query vector for "dog"

$$k_{\text{"I"}} = \begin{bmatrix}\cos(3\theta) & -\sin(3\theta)\\ \sin(3\theta) & \cos(3\theta)\end{bmatrix} W_k \boldsymbol{x}_{\text{"I"}}$$

Key vector for "I"

$$\alpha_{\text{"dog","I"}} = \boldsymbol{x}_{\text{"I"}}^{\top} W_k^{\top} \begin{bmatrix}\cos(3\theta) & -\sin(3\theta)\\ \sin(3\theta) & \cos(3\theta)\end{bmatrix} W_q \boldsymbol{x}_{\text{"dog"}}$$



Depends only on the relative positions

From Jia-Bin Huan's Youtube Video: https://www.youtube.com/watch?v=SMBkImDWOyQ&t=683s

Rotary Positonal Encodings (RoPE)

$$R_\Theta^m \boldsymbol{W}_q \boldsymbol{x}_m$$

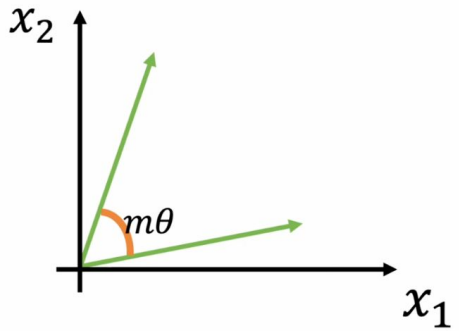
query vector for the token at position m

$$R_\Theta^m = \begin{bmatrix} \cos(m\theta) & -\sin(m\theta) \\ \sin(m\theta) & \cos(m\theta) \end{bmatrix}$$

$$R_\Theta^n \boldsymbol{W}_k \boldsymbol{x}_n$$

key vector for the token at position n

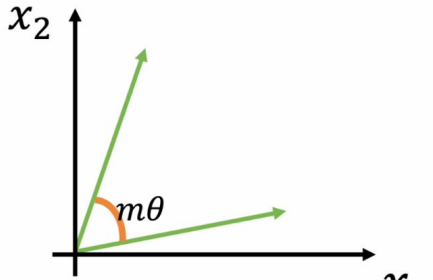
$$\alpha_{m,n} = \boldsymbol{x}_n^\top \boldsymbol{W}_k^\top \quad R_\Theta^{m-n} \quad \boldsymbol{W}_q \boldsymbol{x}_m$$

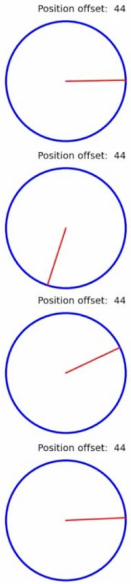


From Jia-Bin Huan's Youtube Video: https://www.youtube.com/watch?v=SMBkImDWOyQ&t=683s

Rotary Positonal Encodings (RoPE)

$$R_\Theta^m = \begin{bmatrix} \cos(m\theta) & -\sin(m\theta) \\ \sin(m\theta) & \cos(m\theta) \end{bmatrix}$$





$$\begin{bmatrix} \square \\ \square \end{bmatrix} = \begin{bmatrix} \cos(m\theta_1) & -\sin(m\theta_1) \\ \sin(m\theta_1) & \cos(m\theta_1) \end{bmatrix} \begin{bmatrix} \square \\ \square \end{bmatrix}$$

$$\begin{bmatrix} \square \\ \square \end{bmatrix} = \begin{bmatrix} \cos(m\theta_2) & -\sin(m\theta_2) \\ \sin(m\theta_2) & \cos(m\theta_2) \end{bmatrix} \begin{bmatrix} \square \\ \square \end{bmatrix}$$

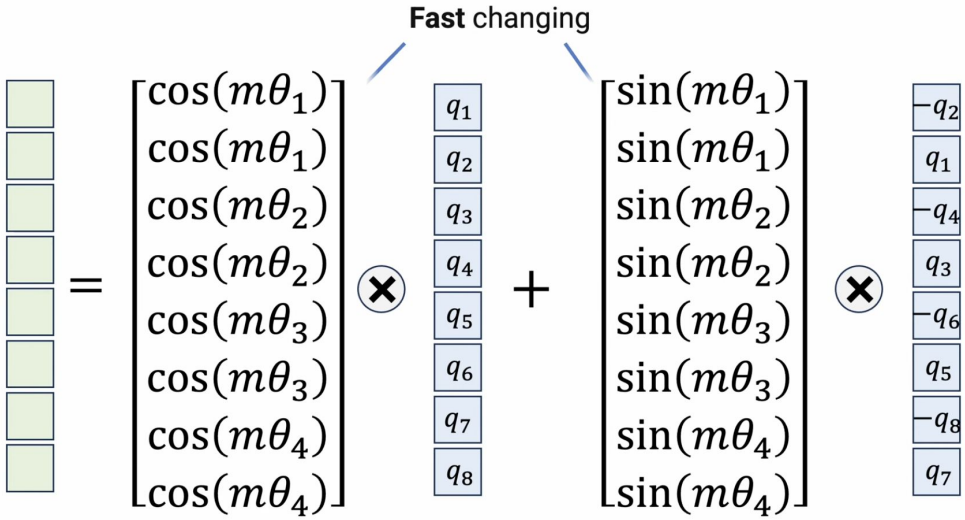
$$\begin{bmatrix} \square \\ \square \end{bmatrix} = \begin{bmatrix} \cos(m\theta_3) & -\sin(m\theta_3) \\ \sin(m\theta_3) & \cos(m\theta_3) \end{bmatrix} \begin{bmatrix} \square \\ \square \end{bmatrix}$$

$$\begin{bmatrix} \square \\ \square \end{bmatrix} = \begin{bmatrix} \cos(m\theta_4) & -\sin(m\theta_4) \\ \sin(m\theta_4) & \cos(m\theta_4) \end{bmatrix} \begin{bmatrix} \square \\ \square \end{bmatrix}$$

$$W_q x_m$$

From Jia-Bin Huan's Youtube Video: https://www.youtube.com/watch?v=SMBkImDWOyQ&t=683s

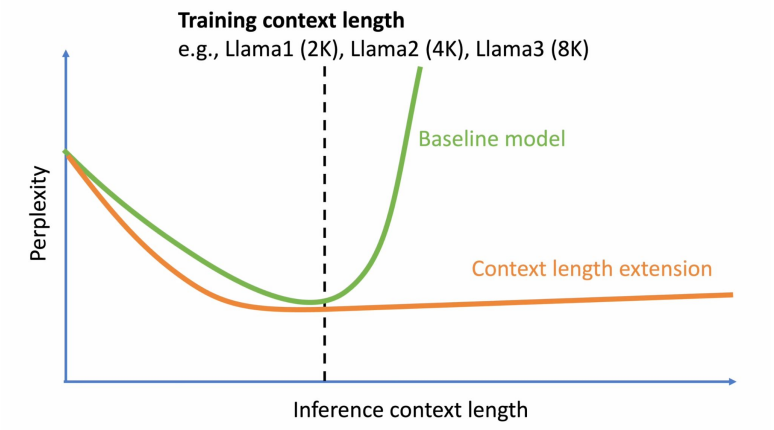
Rotary Positonal Encodings (RoPE)



From Jia-Bin Huan's Youtube Video: https://www.youtube.com/watch?v=SMBkImDWOyQ&t=683s

Rotary Positonal Encodings (RoPE)

- Generally show faster convergence than Absolute Position Encoding methods
- The length generalization is still not guaranteed!
- Unseen relative distances i.e. the ones that exceed the maximum sequence length during training remain an issue
- Neural-Tangent-Kernel (NTK) RoPE is an extension to RoPE that enables generalization to long sequences

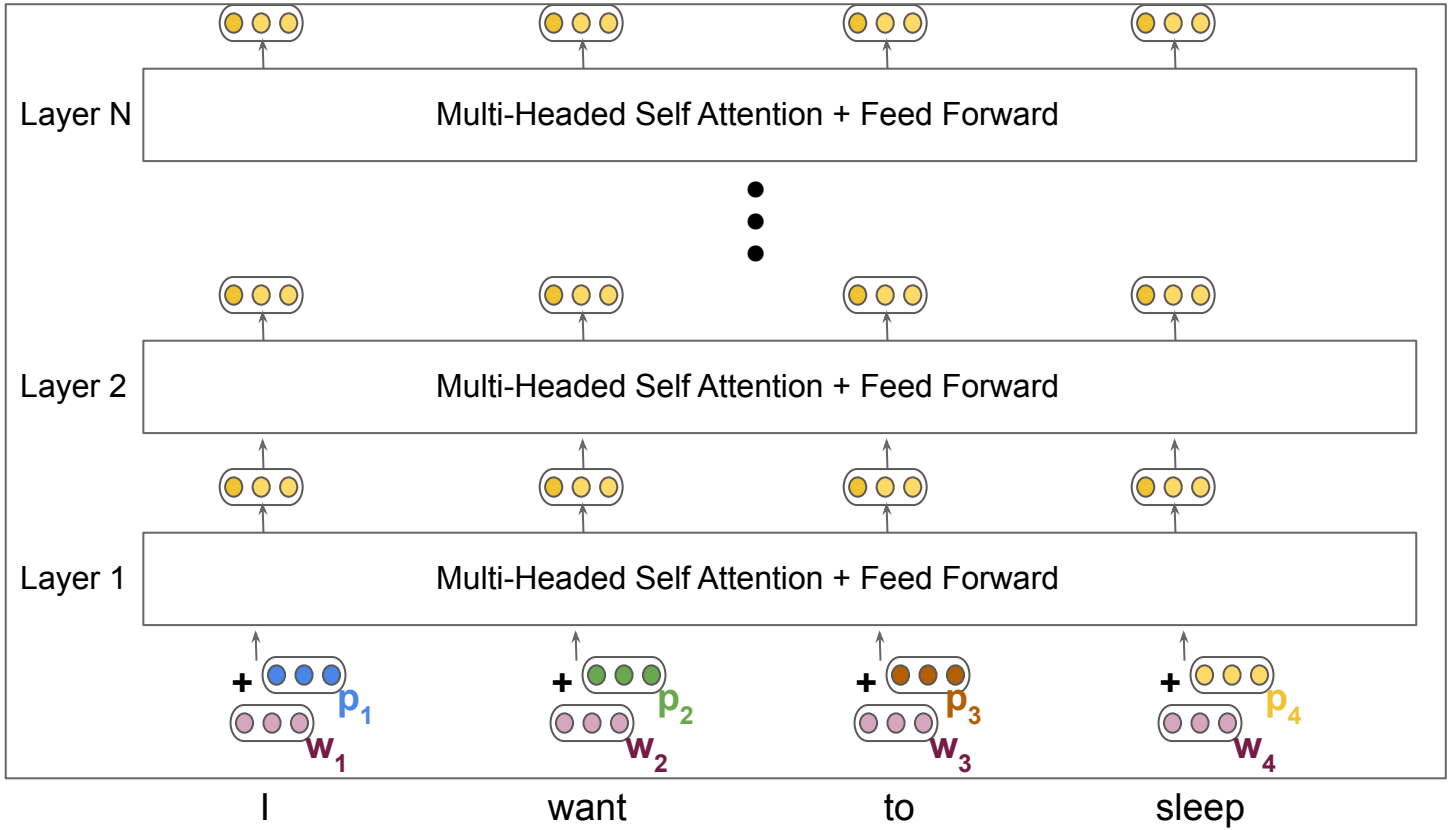


From Jia-Bin Huan's Youtube Video: https://www.youtube.com/watch?v=SMBkImDWOyQ&t=683s

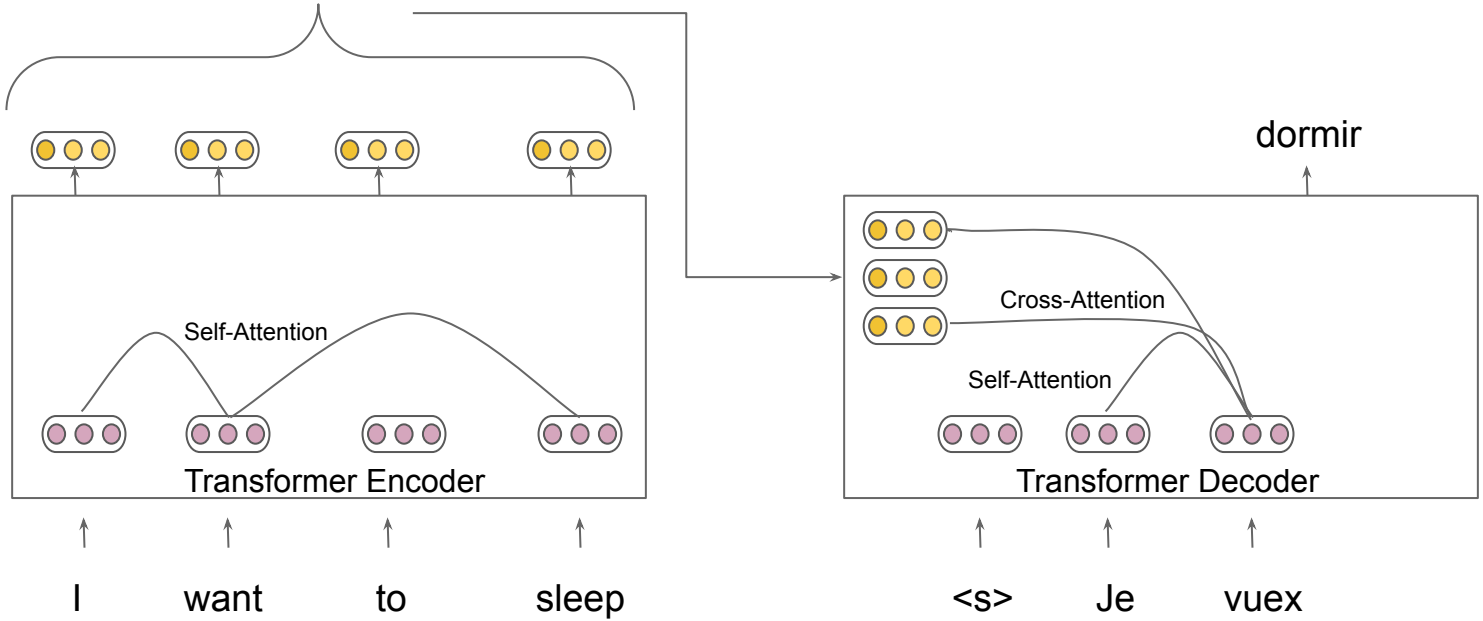
Questions?

Transformer Encoder

N-Layer Transformer Encoder



Transformer Encoder - Decoder

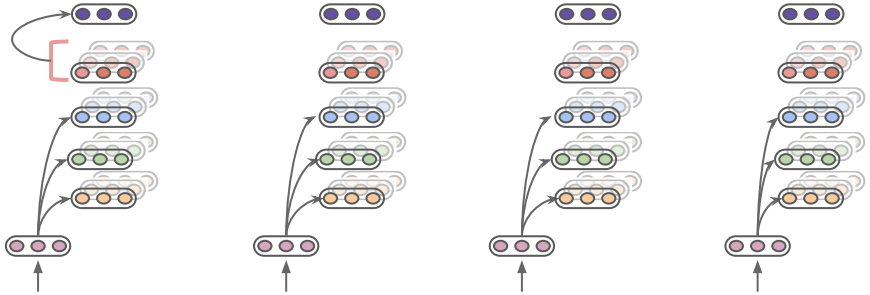


What's so great about Transformers?

- Parallelizable computation
  - Entire sequence, All queries, all attention heads computed in parallel
  - Benefits from fast matrix multiplication on GPUs
- Rich expressive power
  - Every token connected to every other token
  - Can form long range dependencies
- Depth not proportional to seq length
  - Reduces exploding/vanishing gradient problem
  - Converges faster

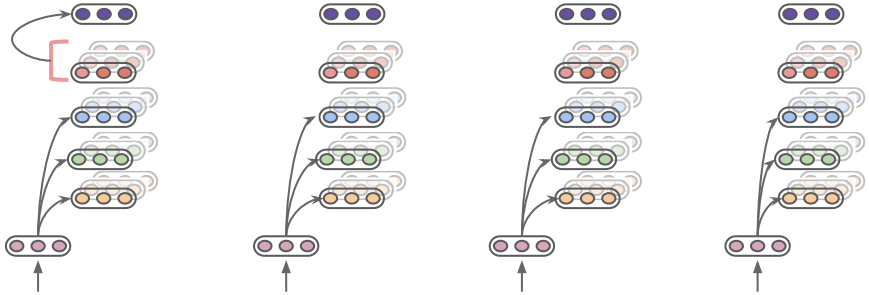
What's so great about Transformers?

- Parallelizable computation - Entire sequence can be processed in parallel

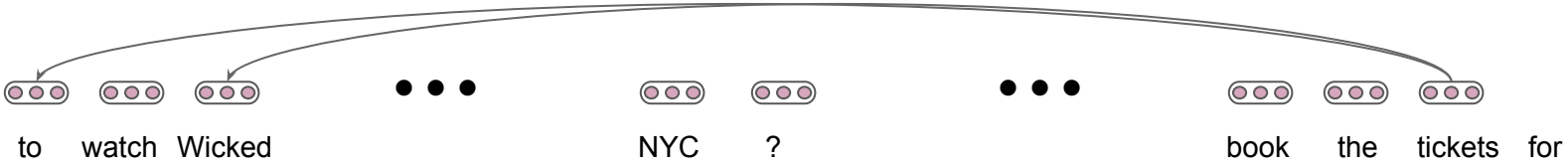


What's so great about Transformers?

- Parallelizable computation - Entire sequence can be processed in parallel

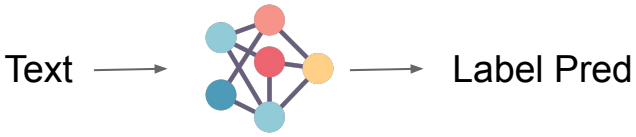


- Rich expressive power - long range dependencies

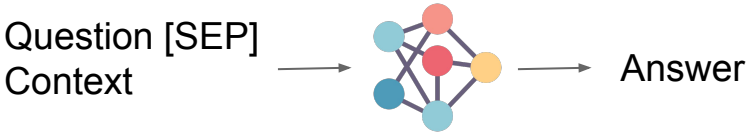


Impact - Wide Applications!

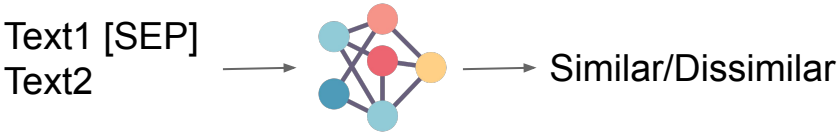
Classification



Question Answering



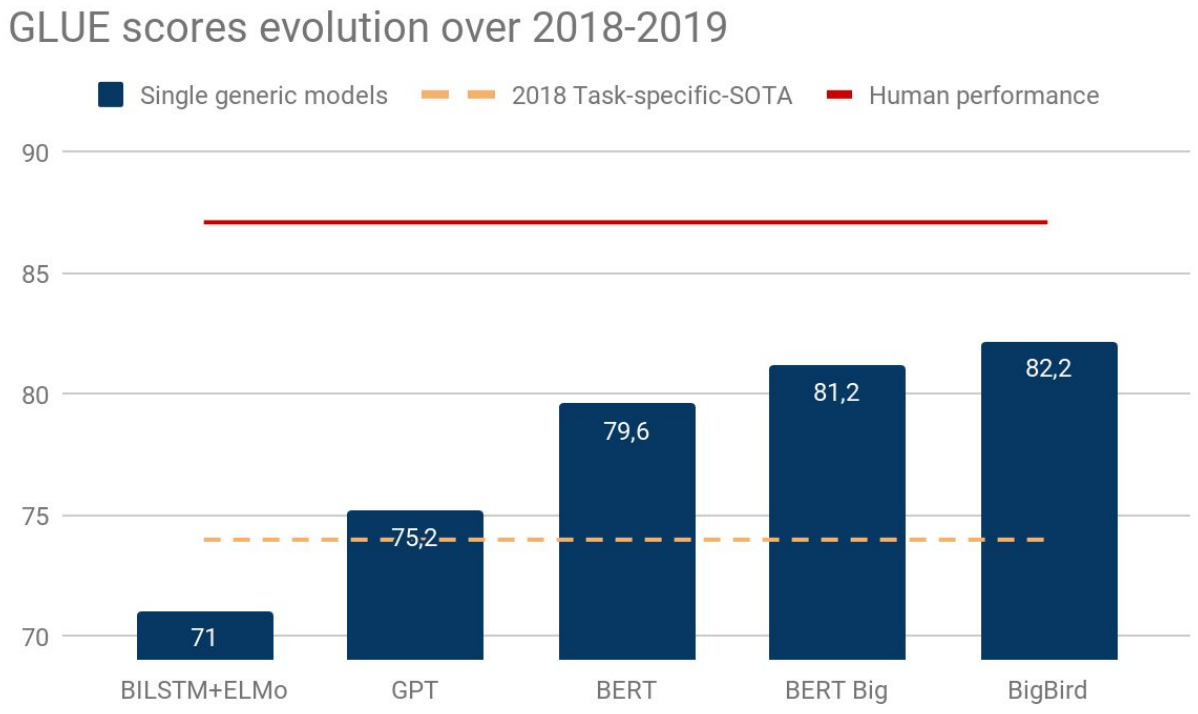
Sentence Similarity



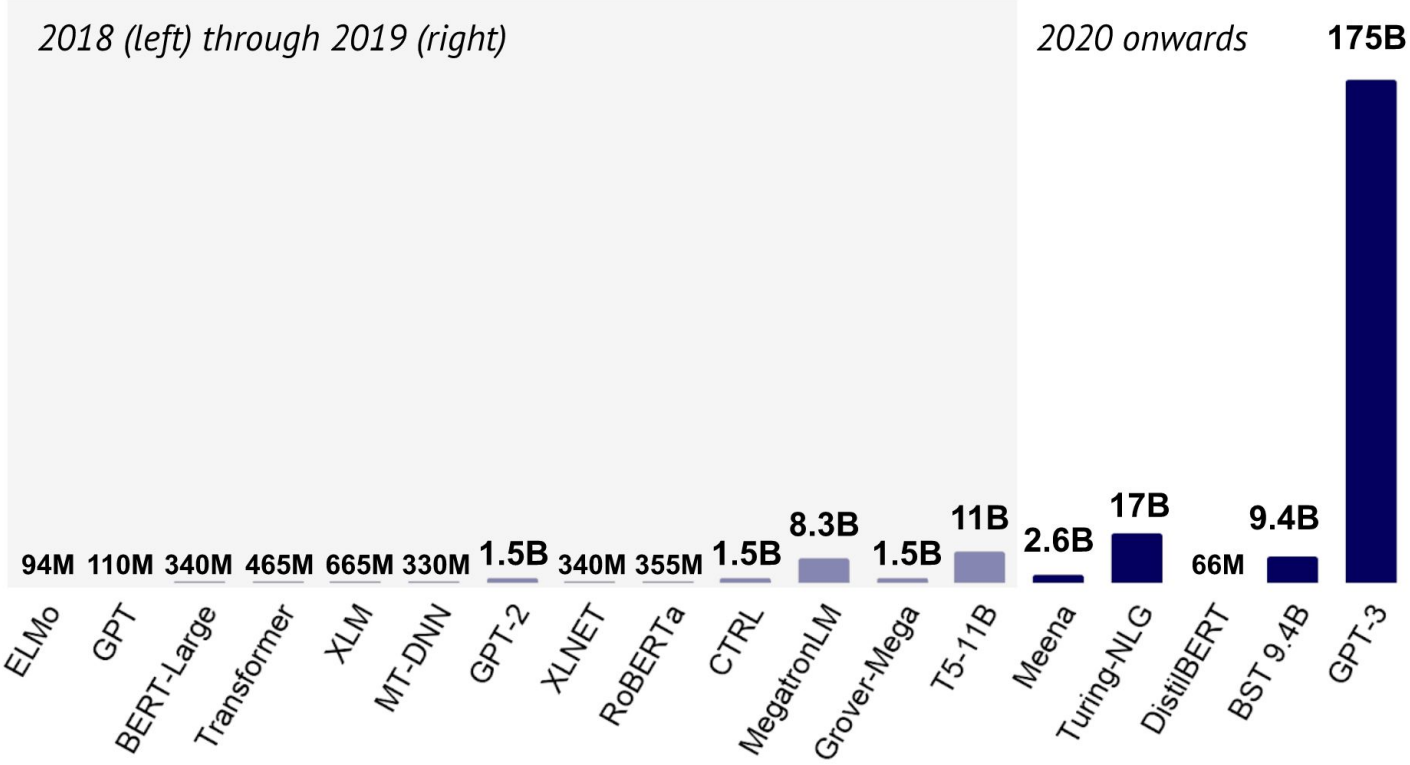
Translation

Larger Impact

Larger Impact



Larger Impact



Thank you!

kahuja@cs.washington.edu

Results/Impact

- Improves results, Establishes SOTA in various tasks!
  - Machine Translation
  - Constituency Parsing
  - Language Modeling
  - and more!

- Computationally faster!
  - No sequential computation - Entire sequence processed in parallel