CS 232: Artificial Intelligence

Fall 2026

Prof. Carolyn Anderson
Wellesley College

Reminders

- Homework 7 due Monday
- My help hours:

  Today 3-4 in W423 and Monday 3-4 in W323

Bonus Talk Opportunity



AI, Race, and Chinese Gaming Technologies

This talk investigates how AI recodes "Asian" faces into operative images in games, with particular attention to the scientific paradigms and epistemological convictions that underpin such recoding. It focuses on three Chinese deep learning models that reorganize facial information and redistribute semantic significance on digital faces.

Homework 7

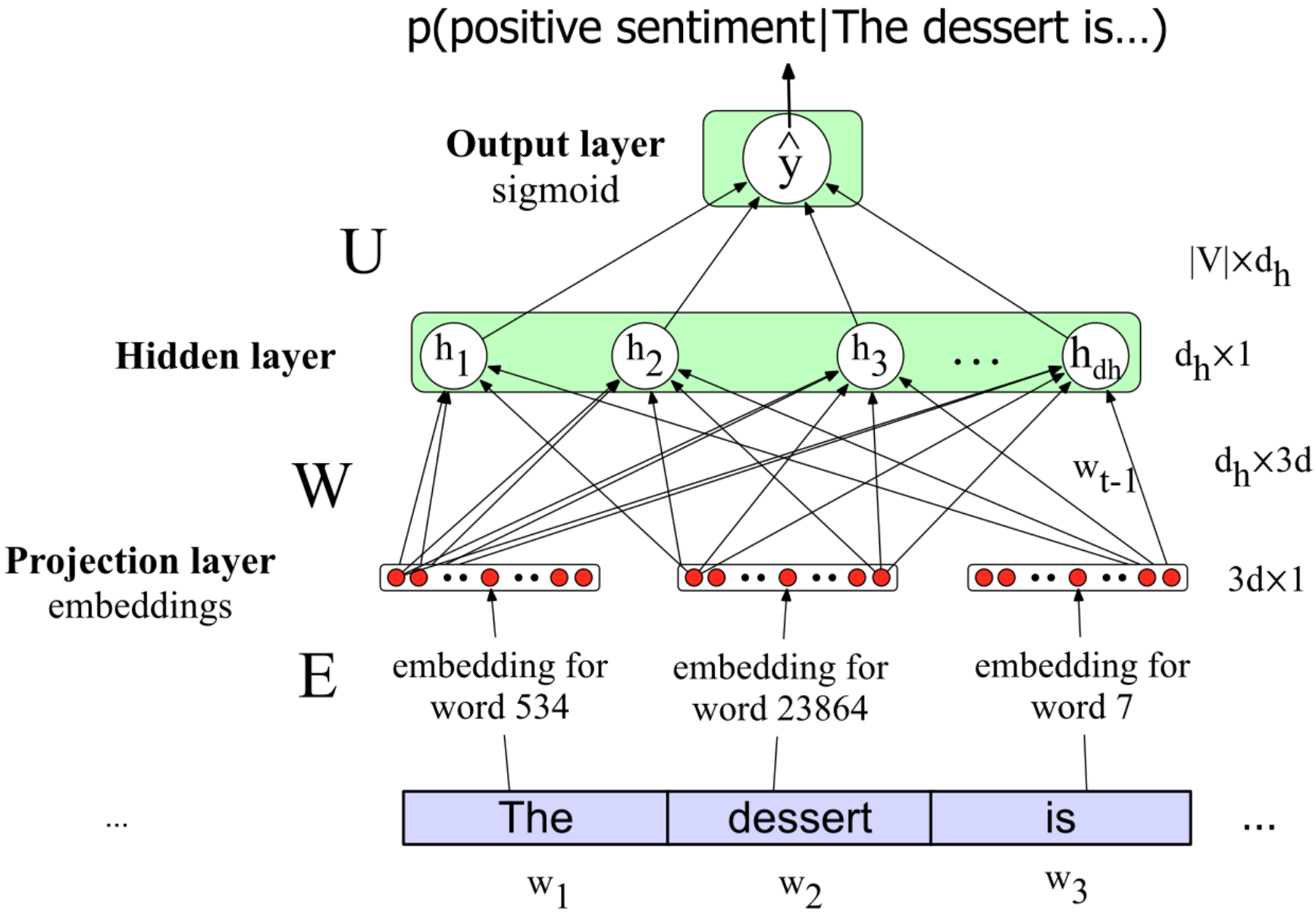
In HW 7, you will *evaluate* a pretrained Transformer model on two tasks: **solving** plain-text Murdle puzzles and **generating** new Murdle puzzles.

For puzzle generation, you will also do a small-scale human evaluation of the generated puzzles for qualities that can't be automatically scored.

Because this is a hard task, you will work with a proprietary OpenAI model. I have emailed you a secret API key to use. **PLEASE DO NOT SHARE IT!**

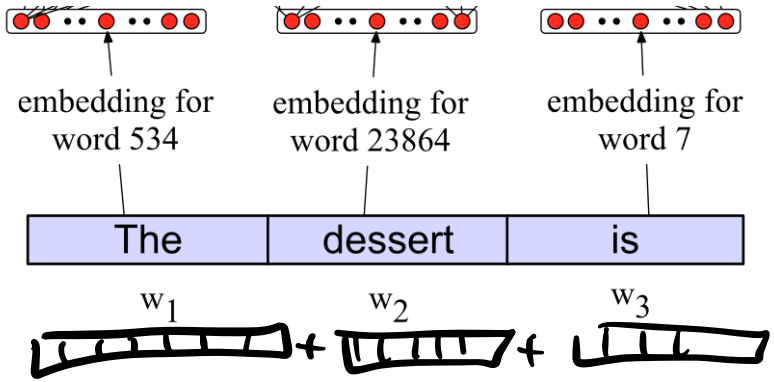
Recap

Neural Net Classification with embeddings as input features!



Issue: texts come in different sizes

This assumes a fixed size length (3)!



Some simple solutions (more sophisticated solutions later)

1. Make the input the length of the longest review
   - If shorter then pad with zero embeddings
   - Truncate if you get longer reviews at test time
2. Create a single "sentence embedding" (the same dimensionality as a word) to represent all the words
   - Take the mean of all the word embeddings
   - Take the element-wise max of all the word embeddings
     - For each dimension, pick the max value from all words



A Better Solution: Attention

Attention mechanisms allow language models to give **more weight to certain words** when predicting the next word.

Attention

What is attention?

Let's learn a task-specific vector $\boldsymbol{v}$



Intuition: think of $\boldsymbol{v}$ like an "**important word**" vector



x_1: I | x_2: loved | x_3: the | x_4: movie | x_5: !

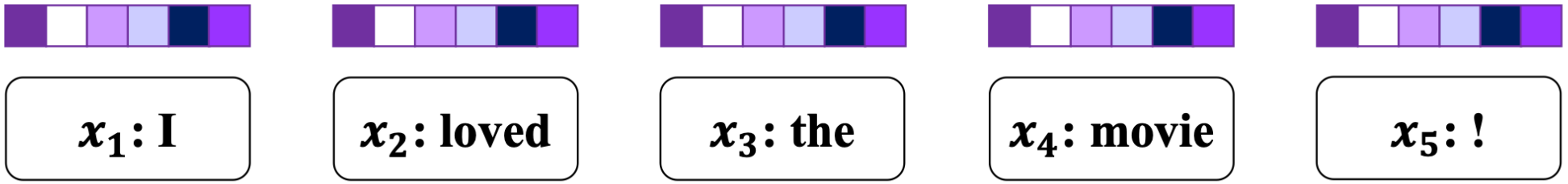
What is attention?

We'll measure the "**importance**" of each input vector $\boldsymbol{x}$ by its similarity (dot product) to $\boldsymbol{v}$ $a = \text{softmax}(r)$

0 0.64 0.02 0.02 0.32

-3.4 2.4 -0.8 -1.2 1.7

$r_1 = v \cdot x_1$ $r_2 = v \cdot x_2$ $r_3 = v \cdot x_3$ $r_4 = v \cdot x_4$ $r_5 = v \cdot x_5$



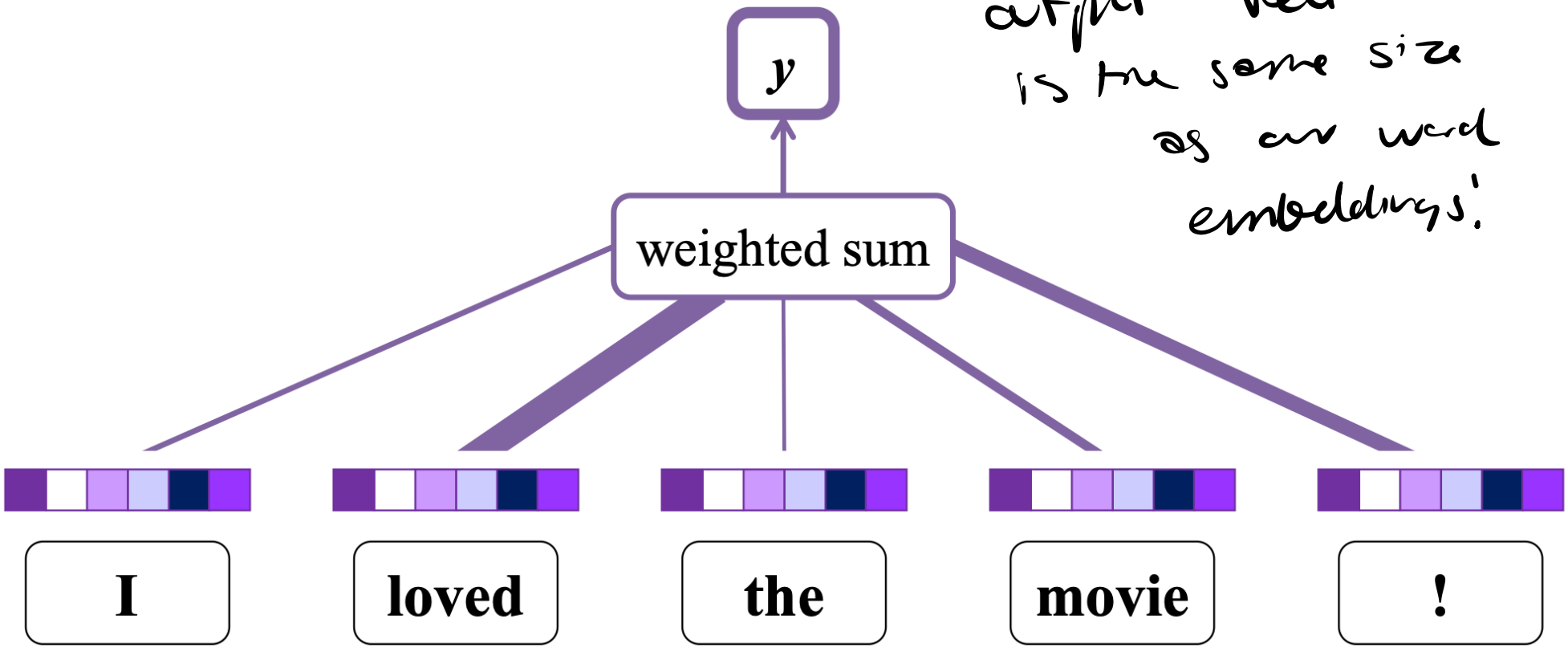
x_1: I | x_2: loved | x_3: the | x_4: movie | x_5: !

Adapted from the slides by Bamman (2021)



= weighted average of our word embeddings based on a-scores

...this seems familiar



output vector is the same size as our word embeddings!

Why dot product?

- Dot product provides a measure of similarity between keys and queries.
- But you might be wondering: *why do we want to pay attention to words that are similar to the current word*?

Why dot product?

- Dot product provides a measure of similarity between keys and queries.
- But you might be wondering: *why do we want to pay attention to words that are similar to the current word*?

Consider:

My brother, a chemist, was late yesterday because he missed the bus. When he arrived, he was surprised to find that his lab _____

Why dot product?

- Dot product provides a measure of similarity between keys and queries.
- But you might be wondering: *why do we want to pay attention to words that are similar to the current word*?

Consider:

My brother, a chemist, was late yesterday because he missed the bus. When he arrived, he was surprised to find that his lab _____

lab



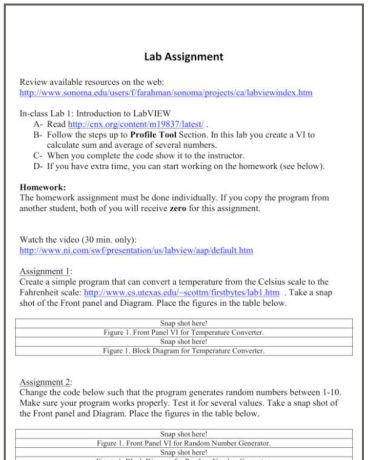
lab



lab



lab



Lab Assignment

Review available resources on the web:
http://www.sonoma.edu/users/f/farahman/sonoma/projects/ca/labviewindex.htm

In-class Lab 1: Introduction to LabVIEW
A- Read http://cnx.org/content/m19837/latest/ .
B- Follow the steps up to **Profile Tool** Section. In this lab you create a VI to calculate sum and average of several numbers.
C- When you complete the code show it to the instructor.
D- If you have extra time, you can start working on the homework (see below).

Homework:
The homework assignment must be done individually. If you copy the program from another student, both of you will receive **zero** for this assignment.

Watch the video (30 min. only):
http://www.ni.com/swf/presentation/us/labview/aap/default.htm

Assignment 1:
Create a simple program that can convert a temperature from the Celsius scale to the Fahrenheit scale: http://www.cs.utexas.edu/~scottm/firstbytes/lab1.htm . Take a snap shot of the Front panel and Diagram. Place the figures in the table below.

Snap shot here!
Figure 1. Front Panel VI for Temperature Converter.
Snap shot here!
Figure 1. Block Diagram for Temperature Converter.

Assignment 2:
Change the code below such that the program generates random numbers between 1-10. Make sure your program works properly. Test it for several values. Take a snap shot of the Front panel and Diagram. Place the figures in the table below.

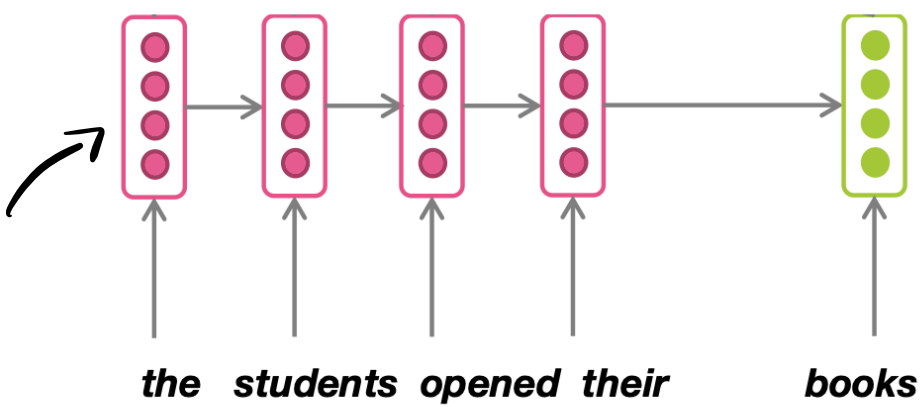
Snap shot here!
Figure 1. Front Panel VI for Random Number Generator.
Snap shot here!


Vicki
@vboykis
They don’t tell you this in the paper (well they do but you have to read it like 15 times)
Multiplying
a lot of vectors
a lot of times
with scaled softmax
Attention
imgflip.com
6:20 PM · Feb 22, 2023 · 88.1K Views

Attention mechanisms in neural language models

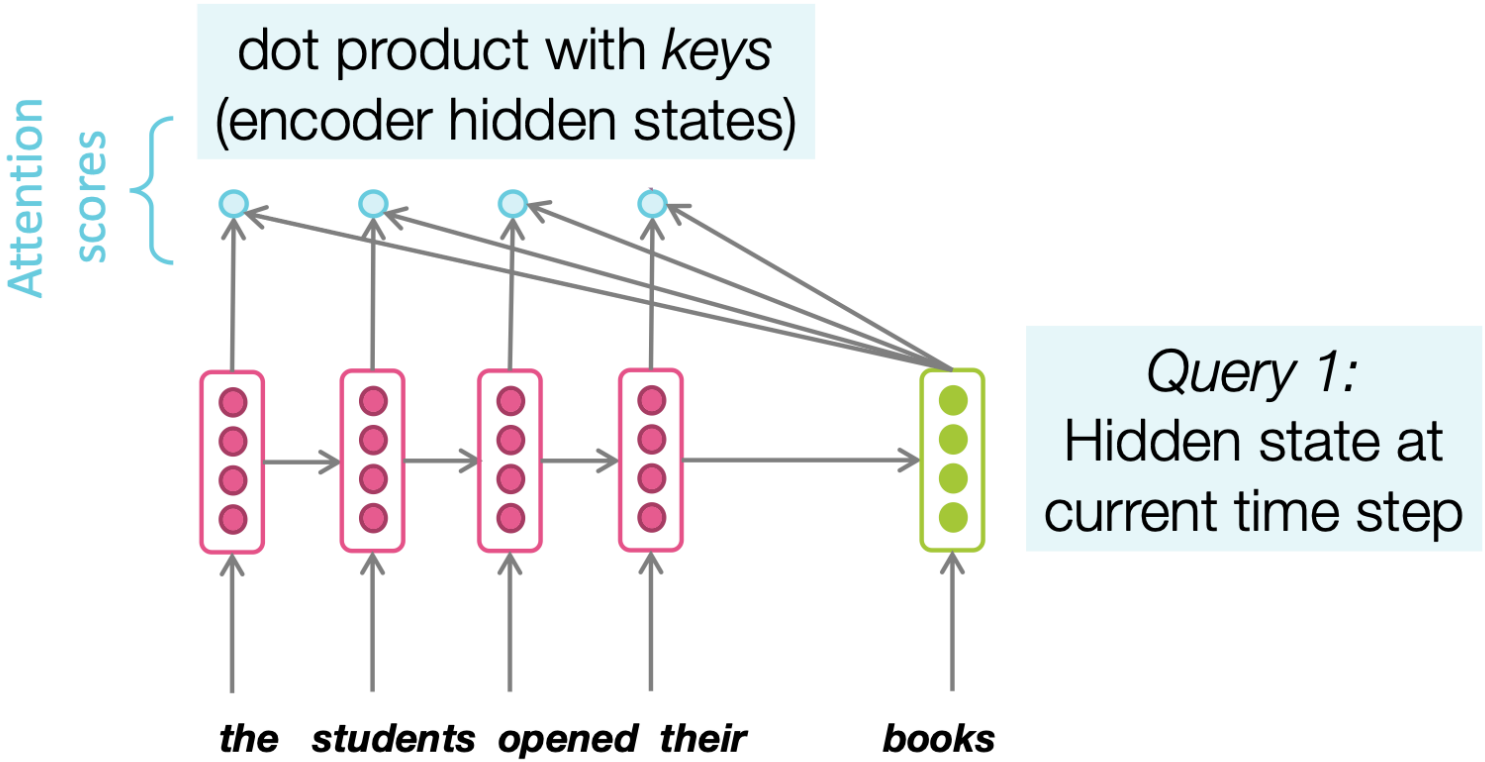
Keys: Words in the prior context

Query: looking for relevant information in the context

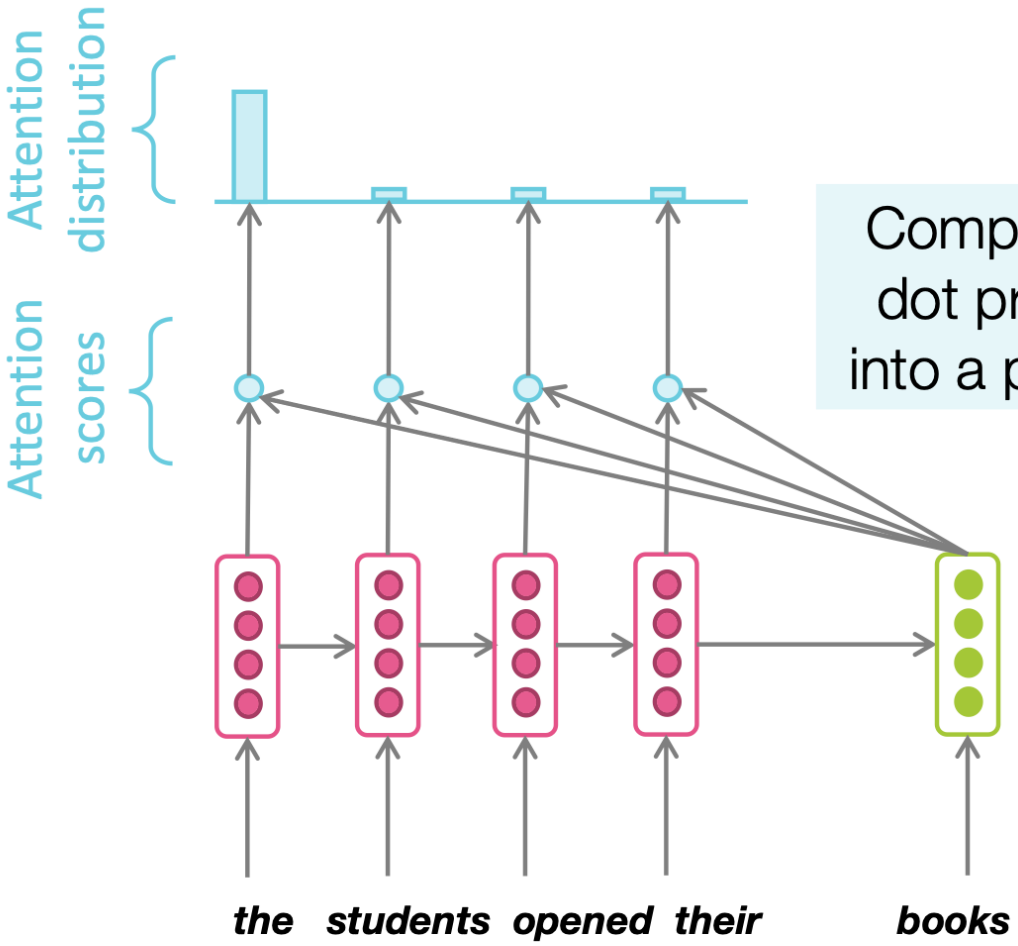


Query 1:
Hidden state at current time step

Attention mechanisms in neural language models

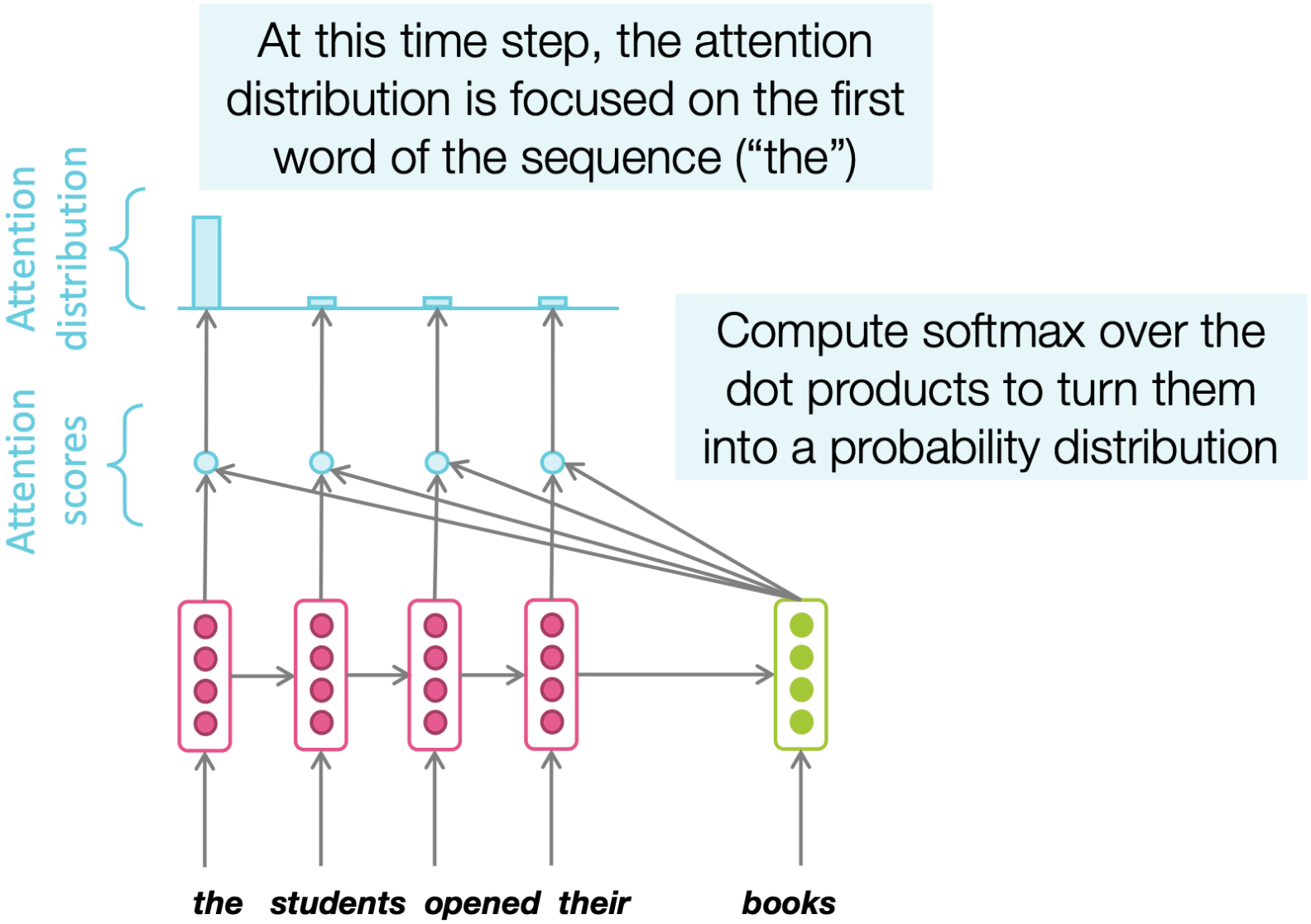


Attention mechanisms in neural language models

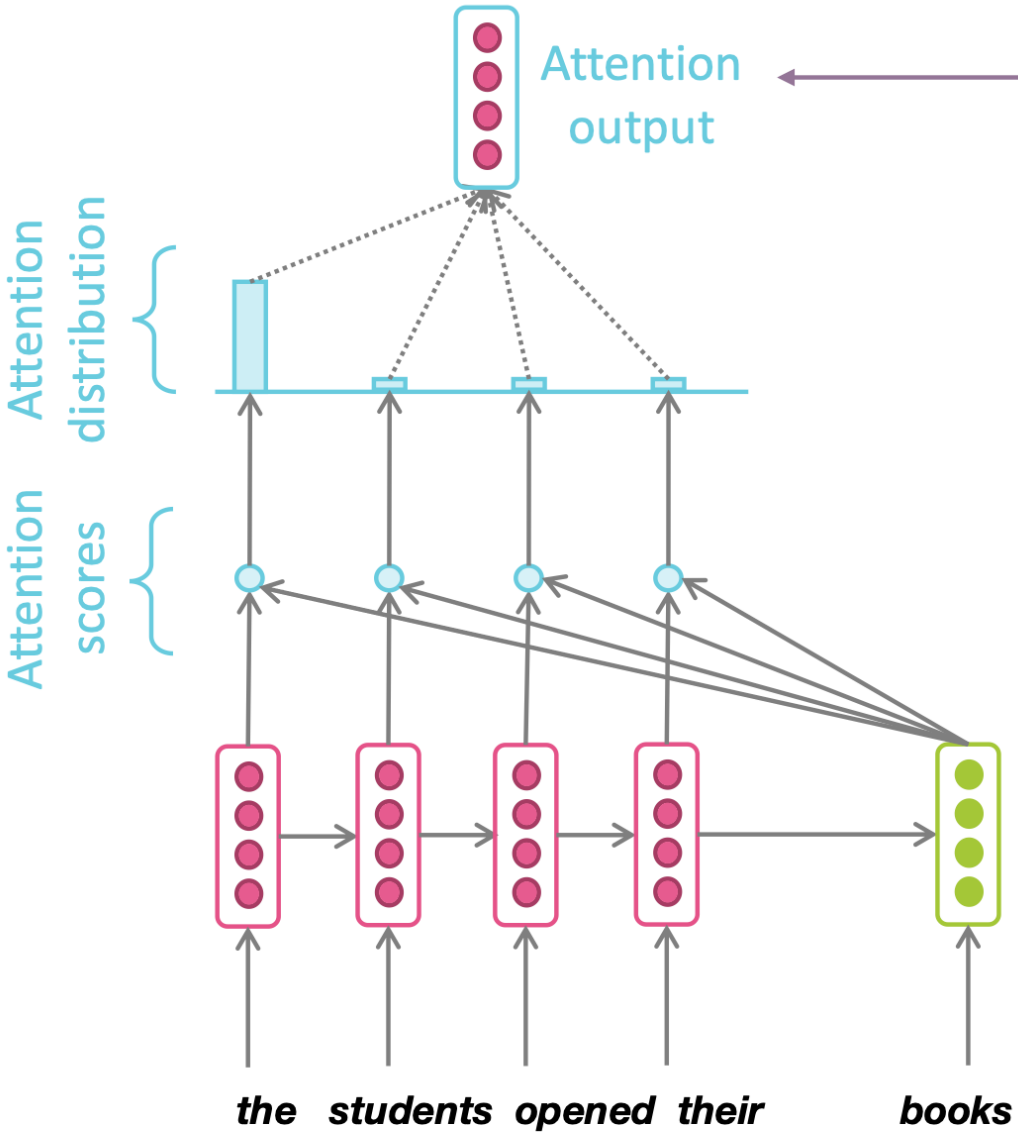


Compute softmax over the dot products to turn them into a probability distribution

Attention mechanisms in neural language models



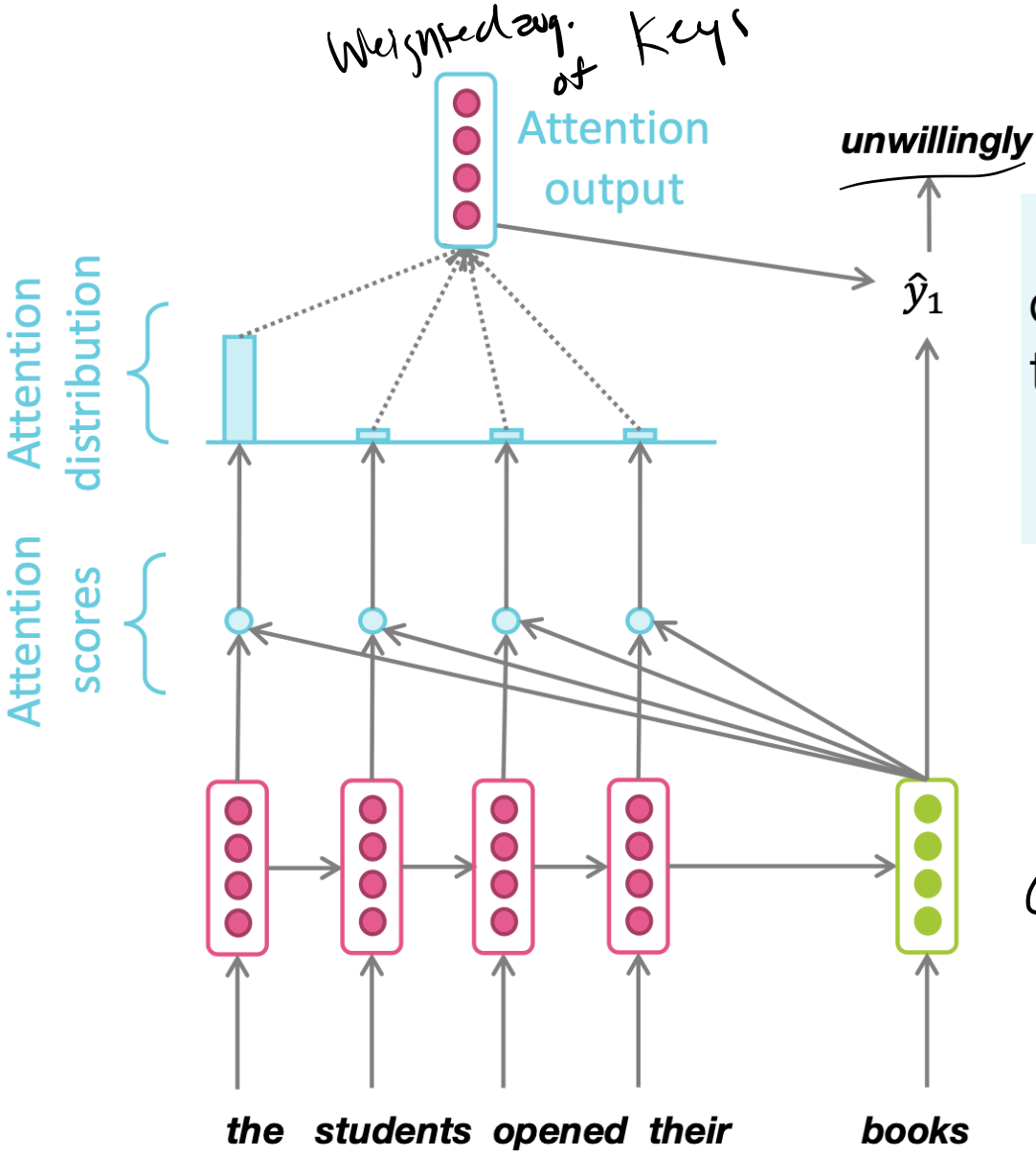
Attention mechanisms in neural language models



We use the attention distribution to compute a weighted average of the hidden states.

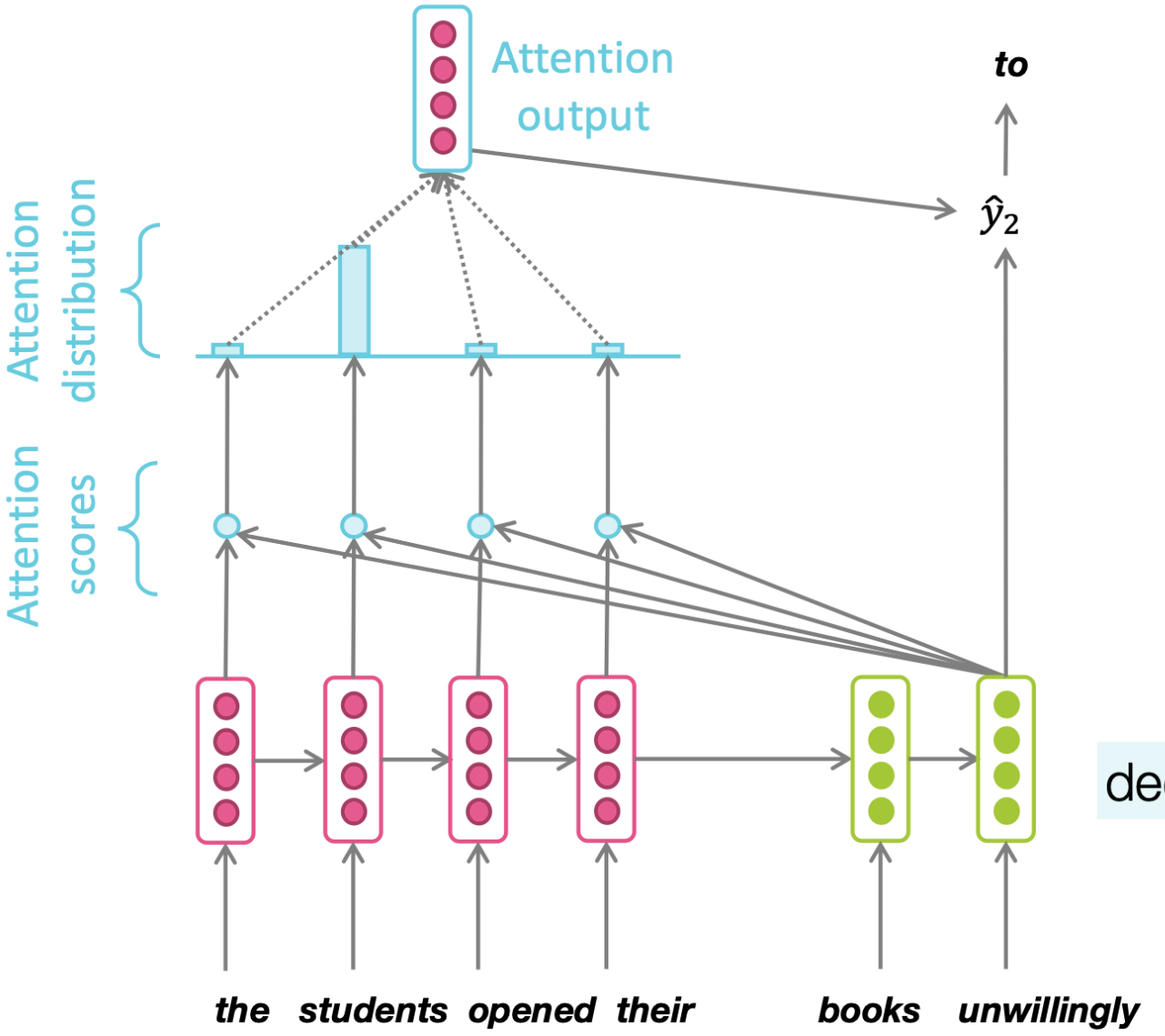
Intuitively, the resulting attention output contains information from hidden states that received high attention scores

Sequence-to-sequence with attention



Concatenate (or otherwise compose) the attention output with the current hidden state, then pass through a softmax layer to predict the next word

Sequence-to-sequence with attention



decoder, second time step

Slides by Emma Strubell!

[Vaswani et al. 2017]

Self-attention

Black: input word embedding

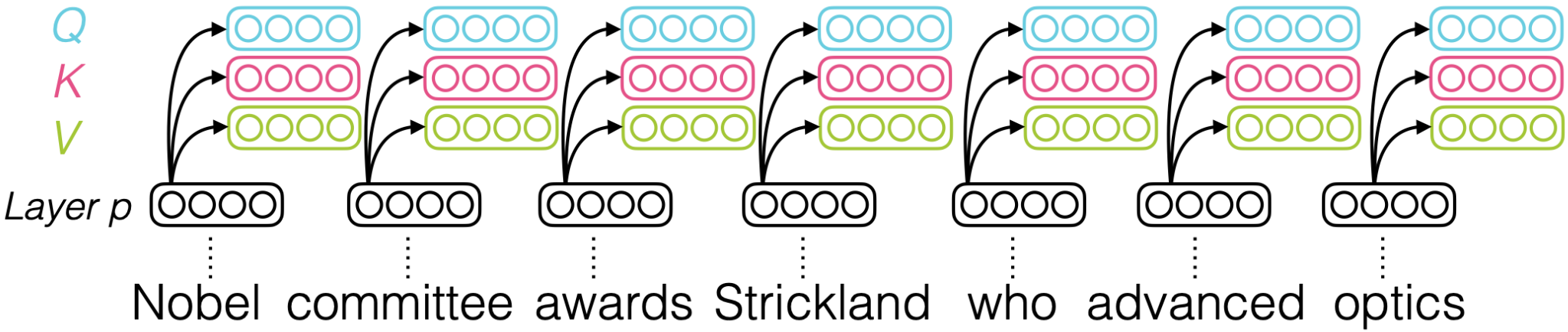
× 3 sets of weights

to produce 3 vectors:

Blue: query vector W_q

Pink: key vector W_k

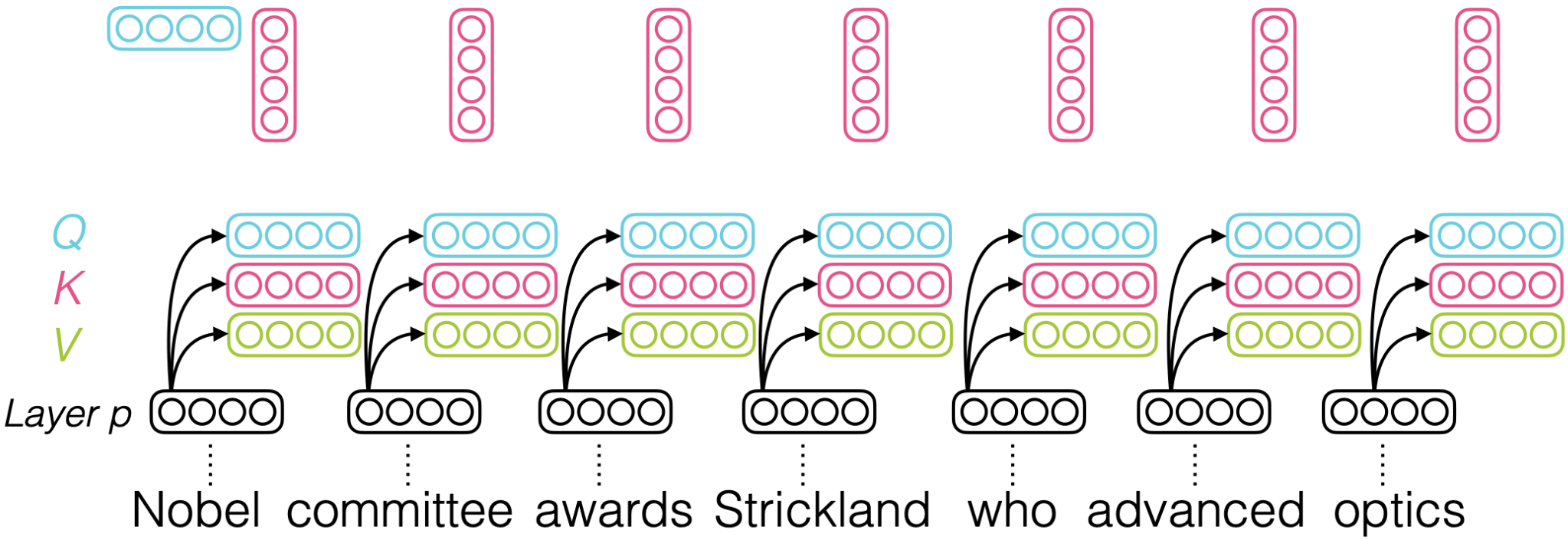
Green: value vector W_v



Slides by Emma Strubell!

Self-attention

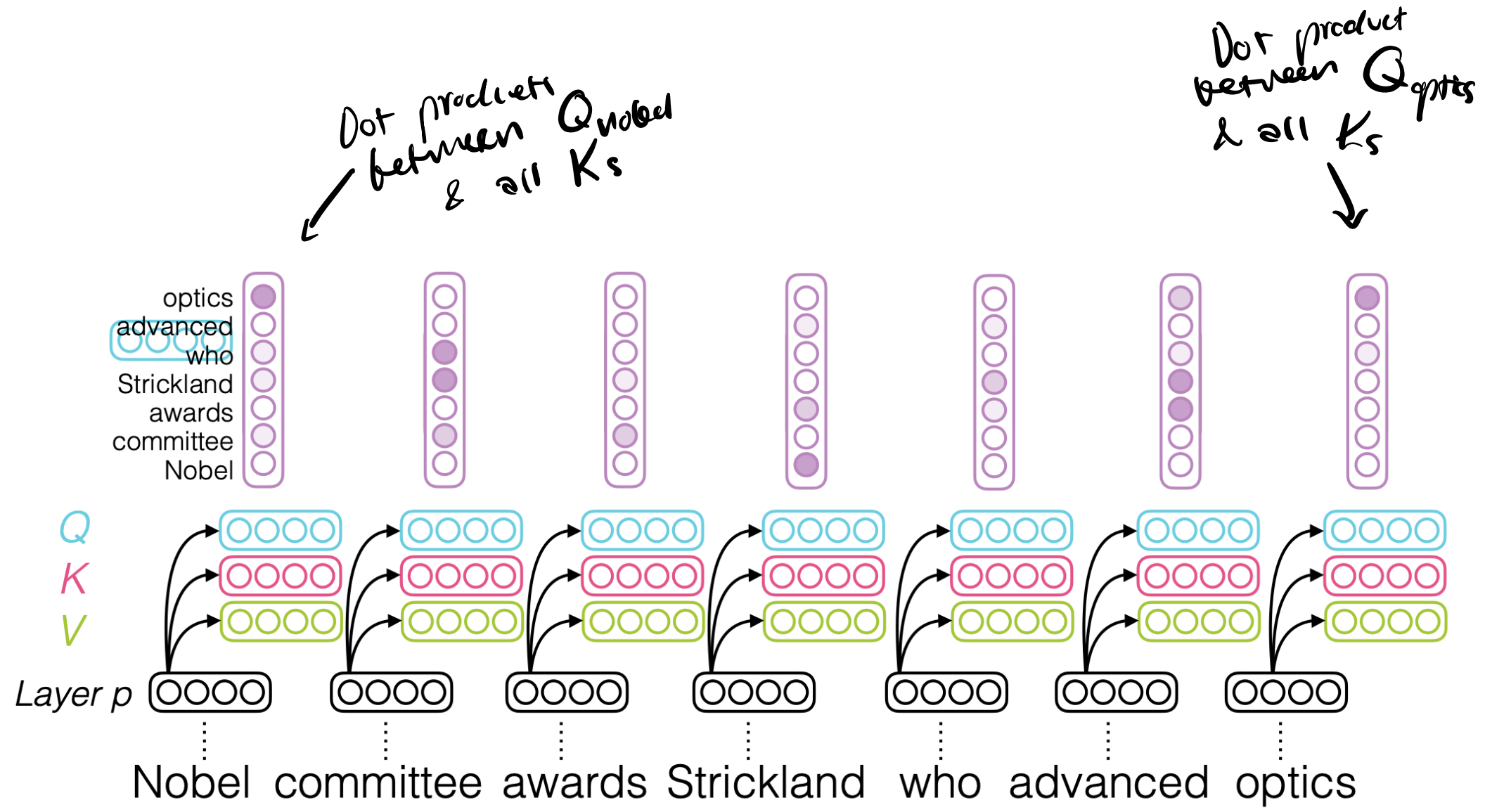
[Vaswani et al. 2017]



Slides by Emma Strubell!

[Vaswani et al. 2017]

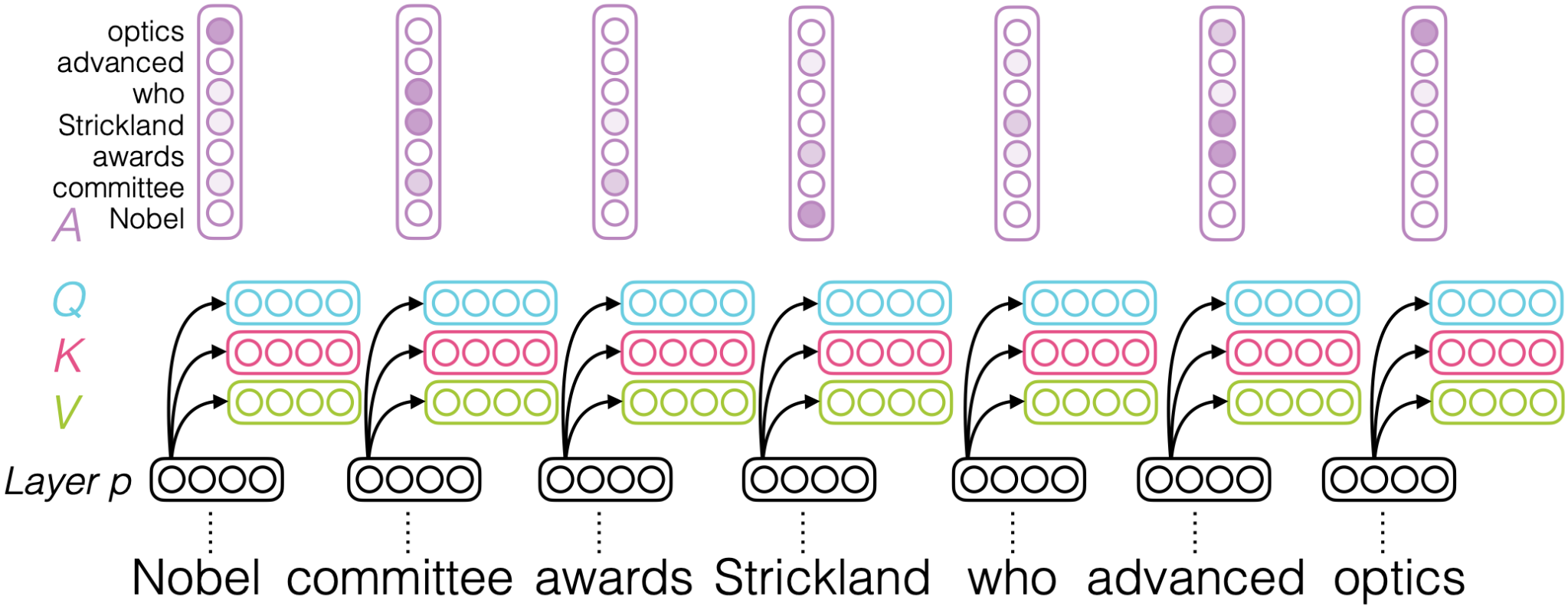
Self-attention



Slides by Emma Strubell!

[Vaswani et al. 2017]

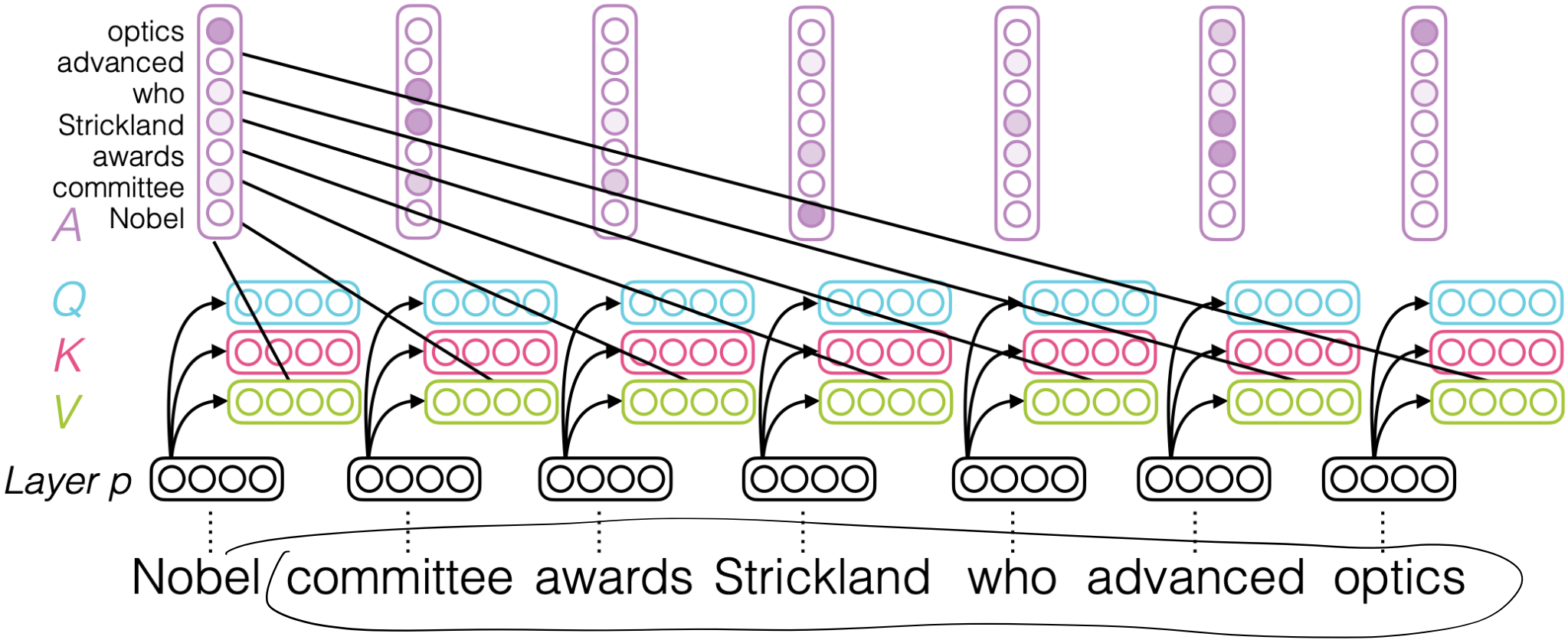
Self-attention



Slides by Emma Strubell!

[Vaswani et al. 2017]

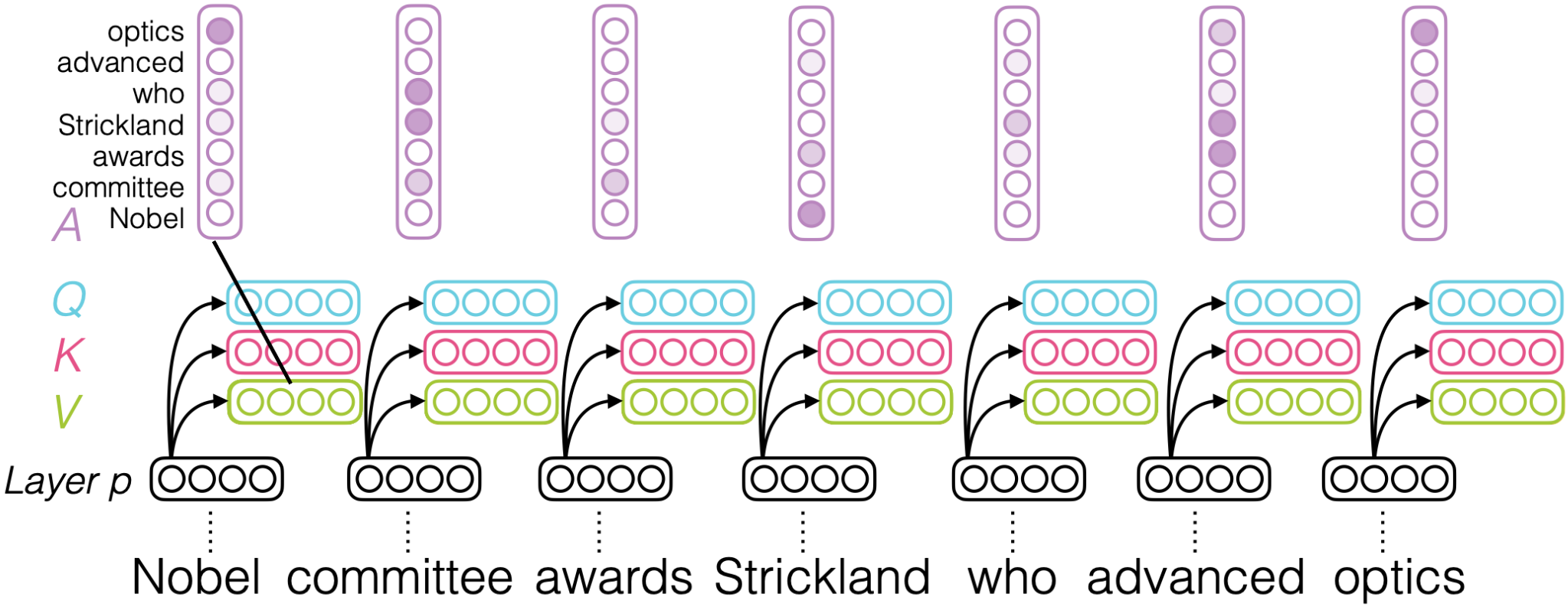
Self-attention



Slides by Emma Strubell!

[Vaswani et al. 2017]

Self-attention

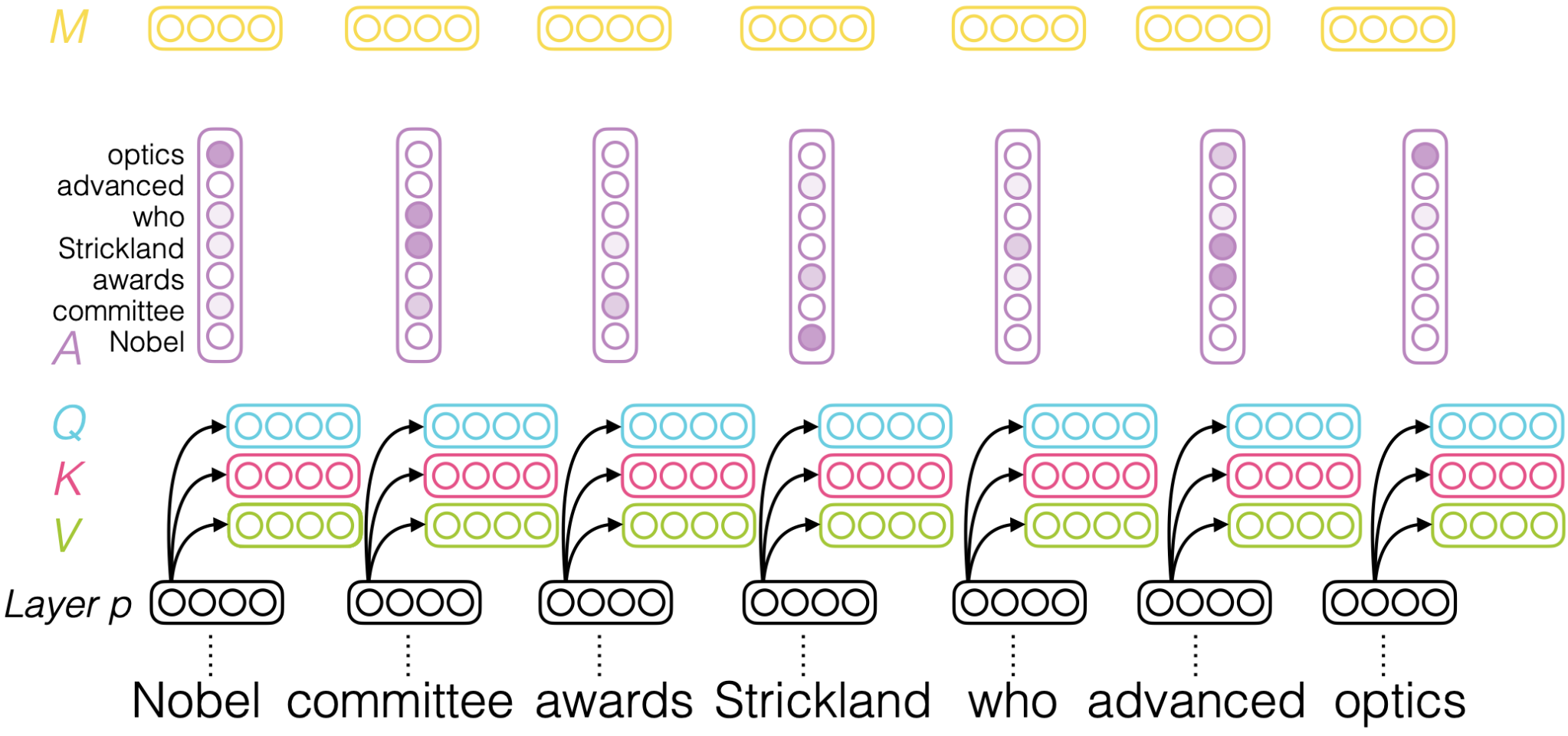


Slides by Emma Strubell!

[Vaswani et al. 2017]

Self-attention

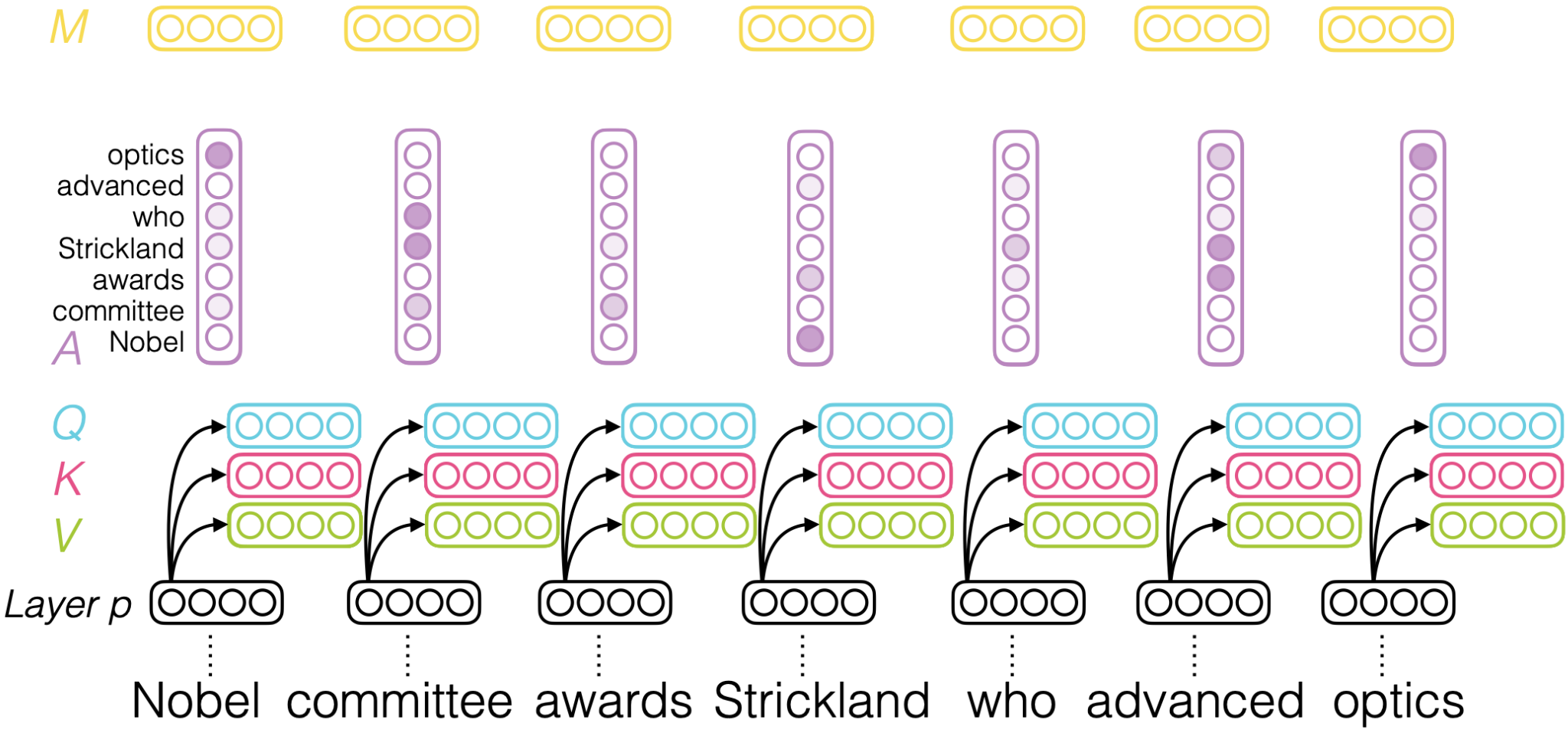
$M = A \cdot V$



Slides by Emma Strubell!

Self-attention

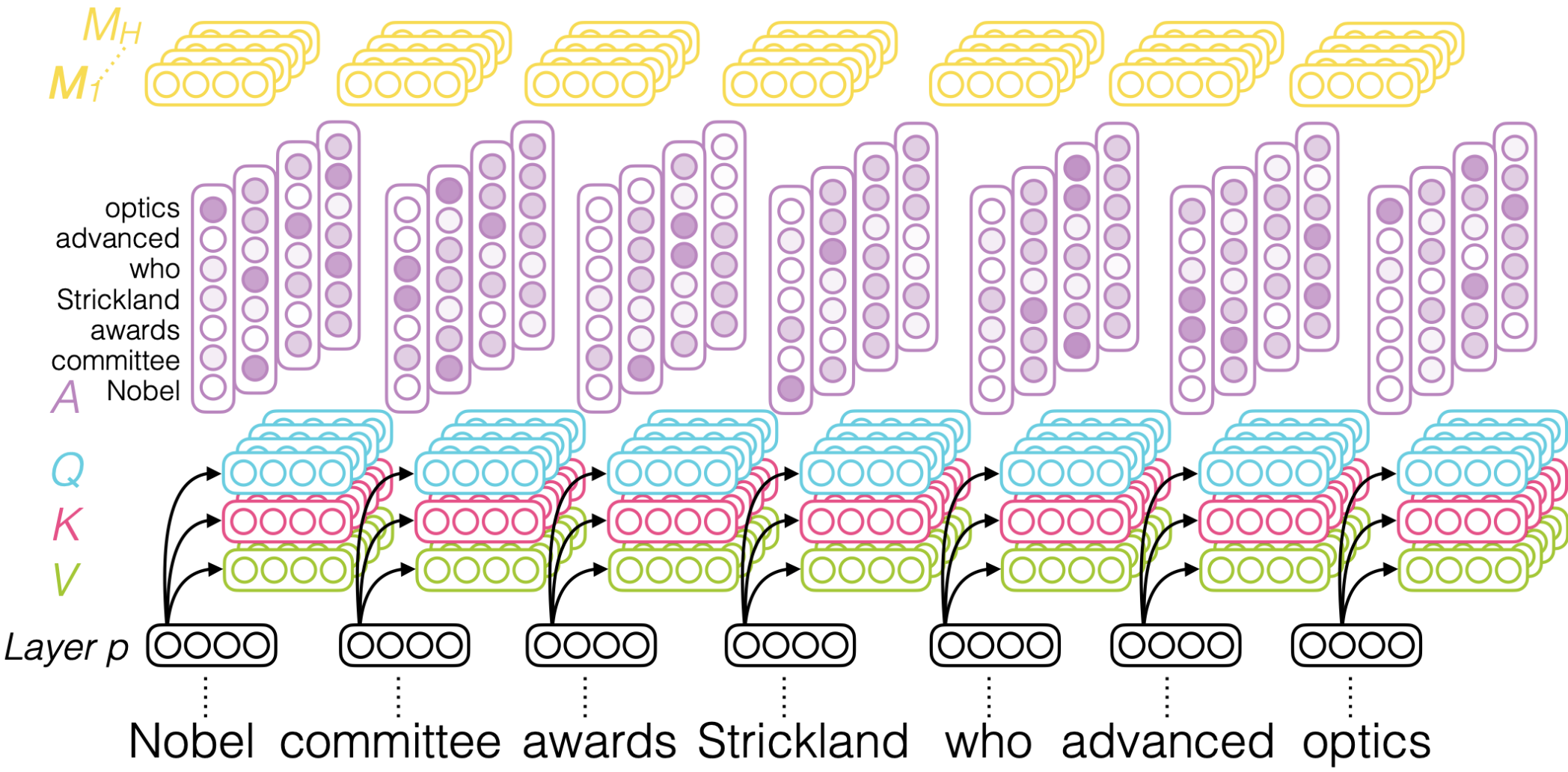
[Vaswani et al. 2017]



Slides by Emma Strubell!

[Vaswani et al. 2017]

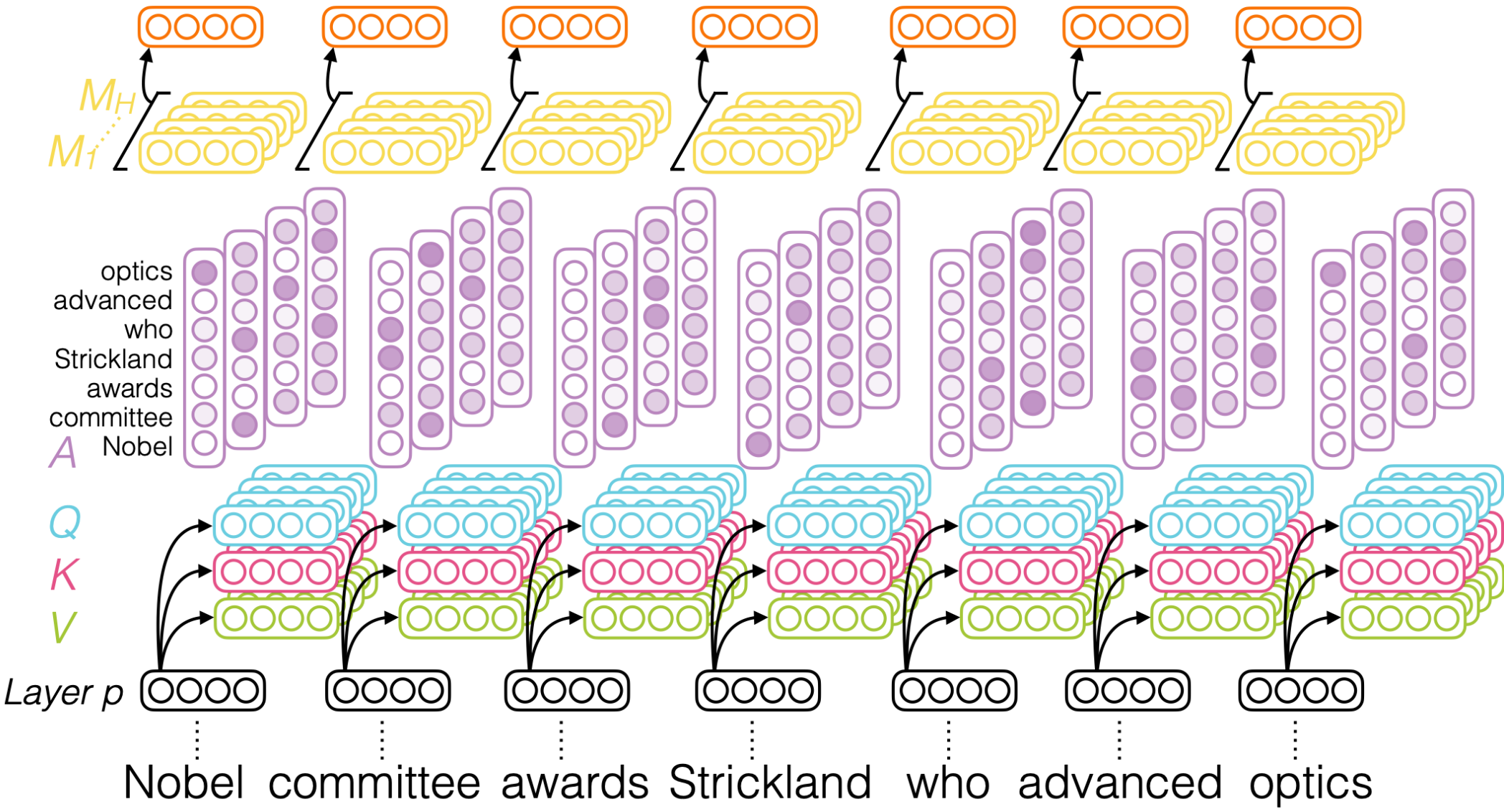
Multi-head self-attention



Slides by Emma Strubell!

[Vaswani et al. 2017]

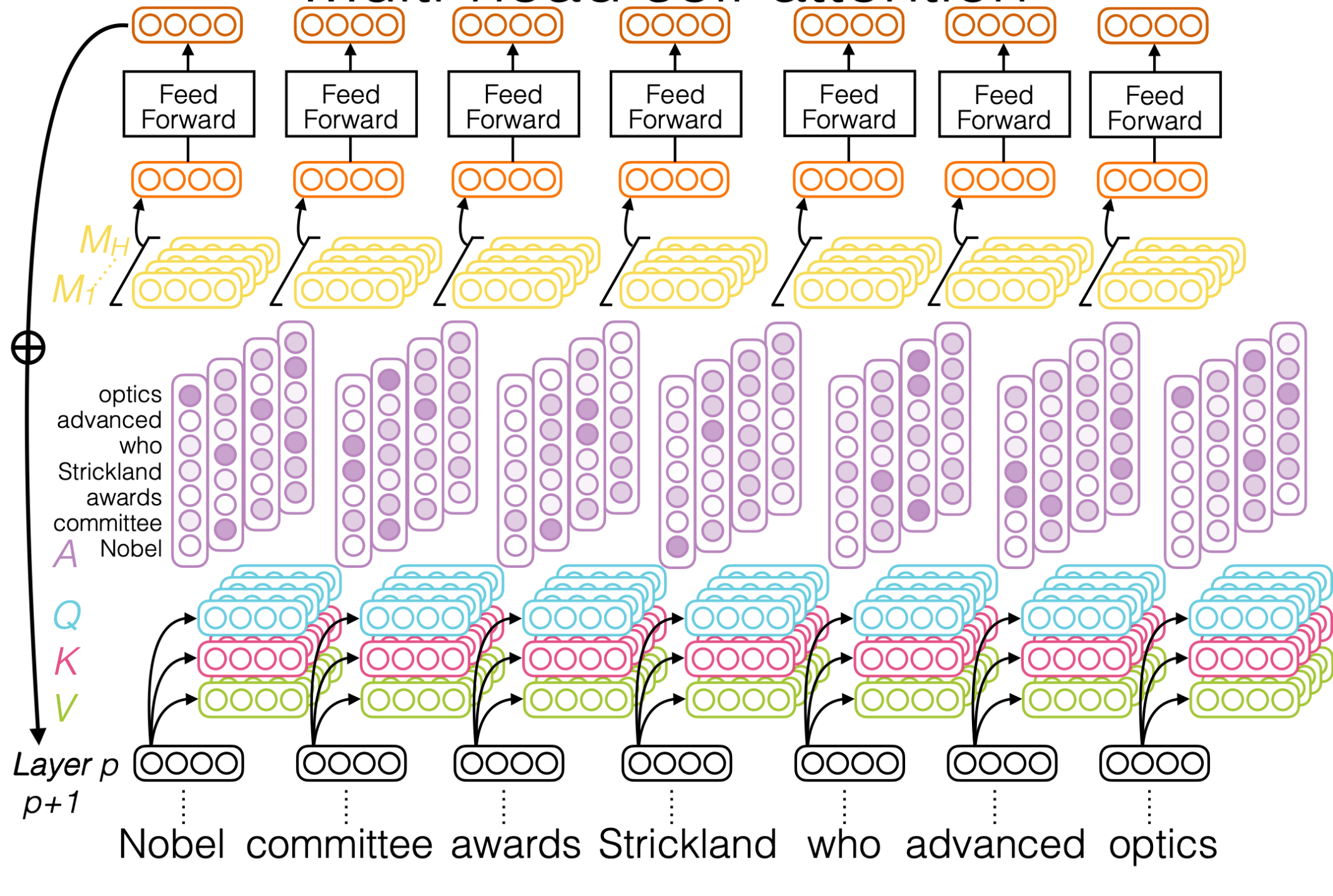
Multi-head self-attention



Slides by Emma Strubell!

[Vaswani et al. 2017]

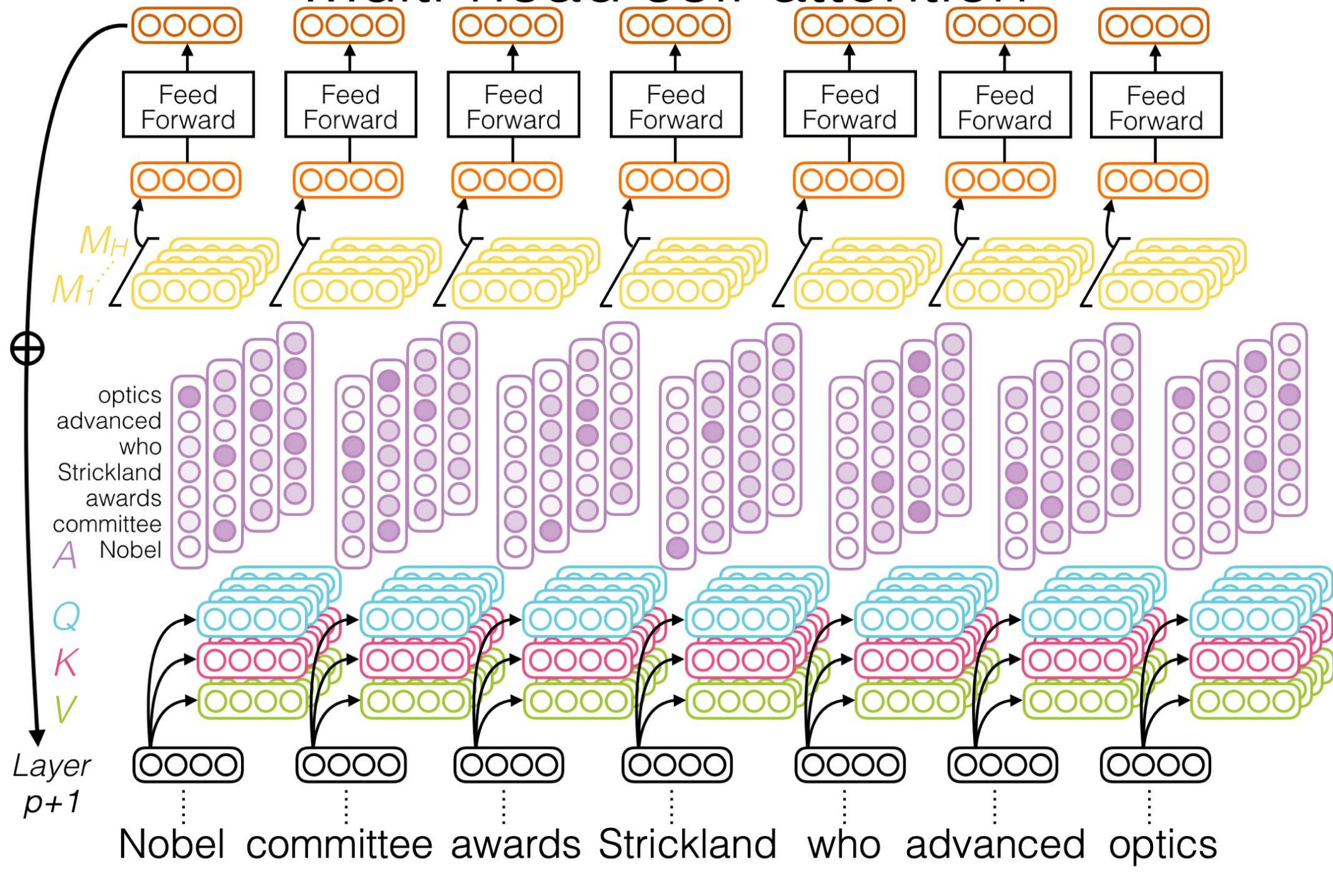
Multi-head self-attention



Slides by Emma Strubell!

[Vaswani et al. 2017]

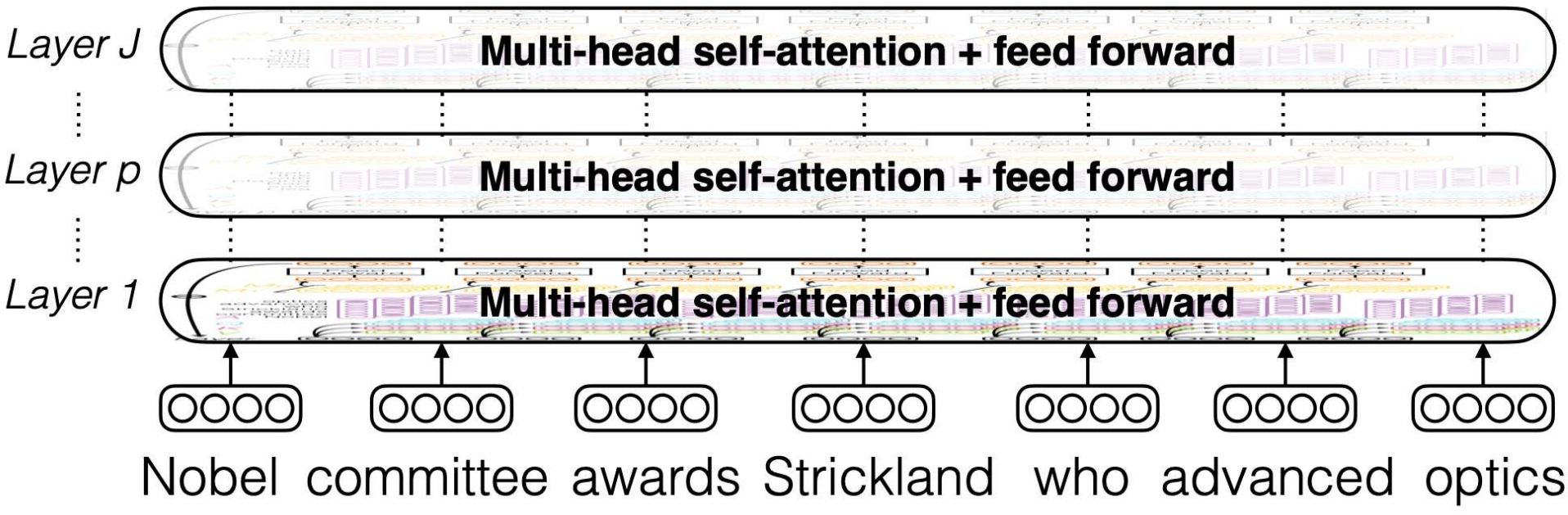
Multi-head self-attention



Slides by Emma Strubell!

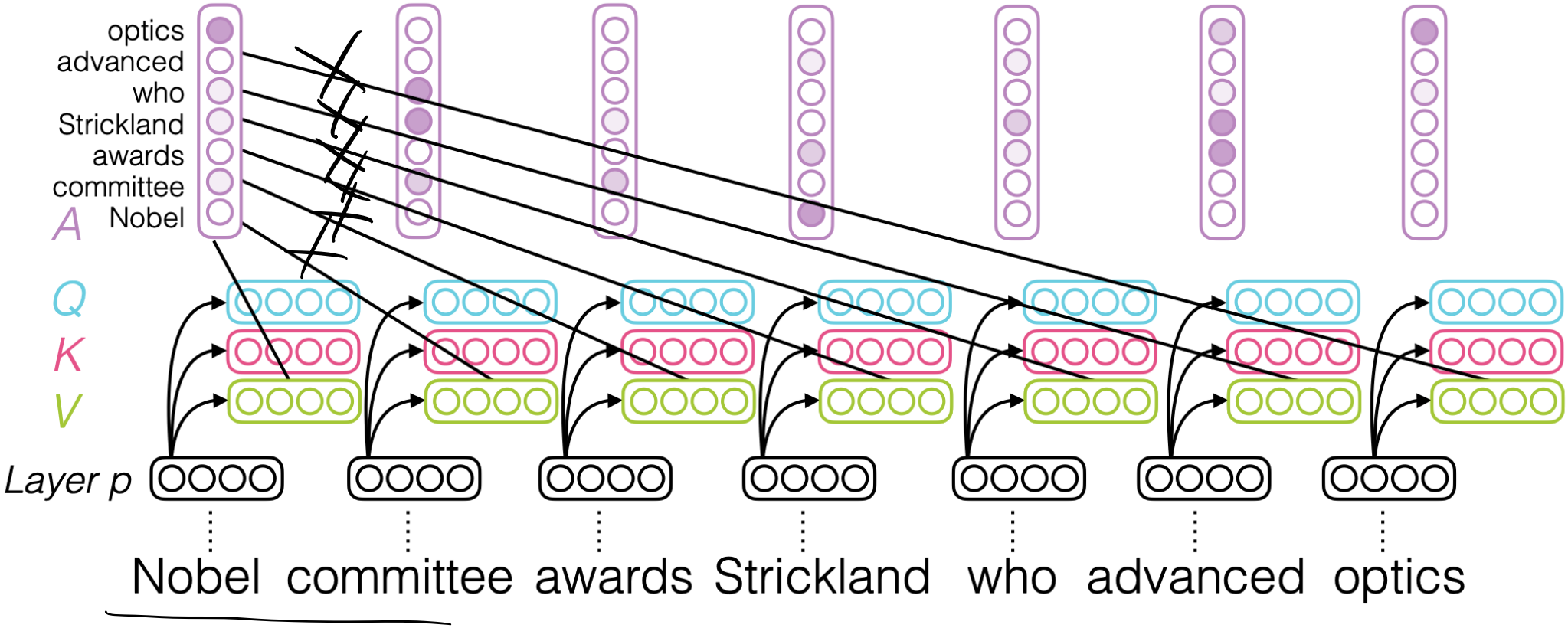
[Vaswani et al. 2017]

Multi-head self-attention



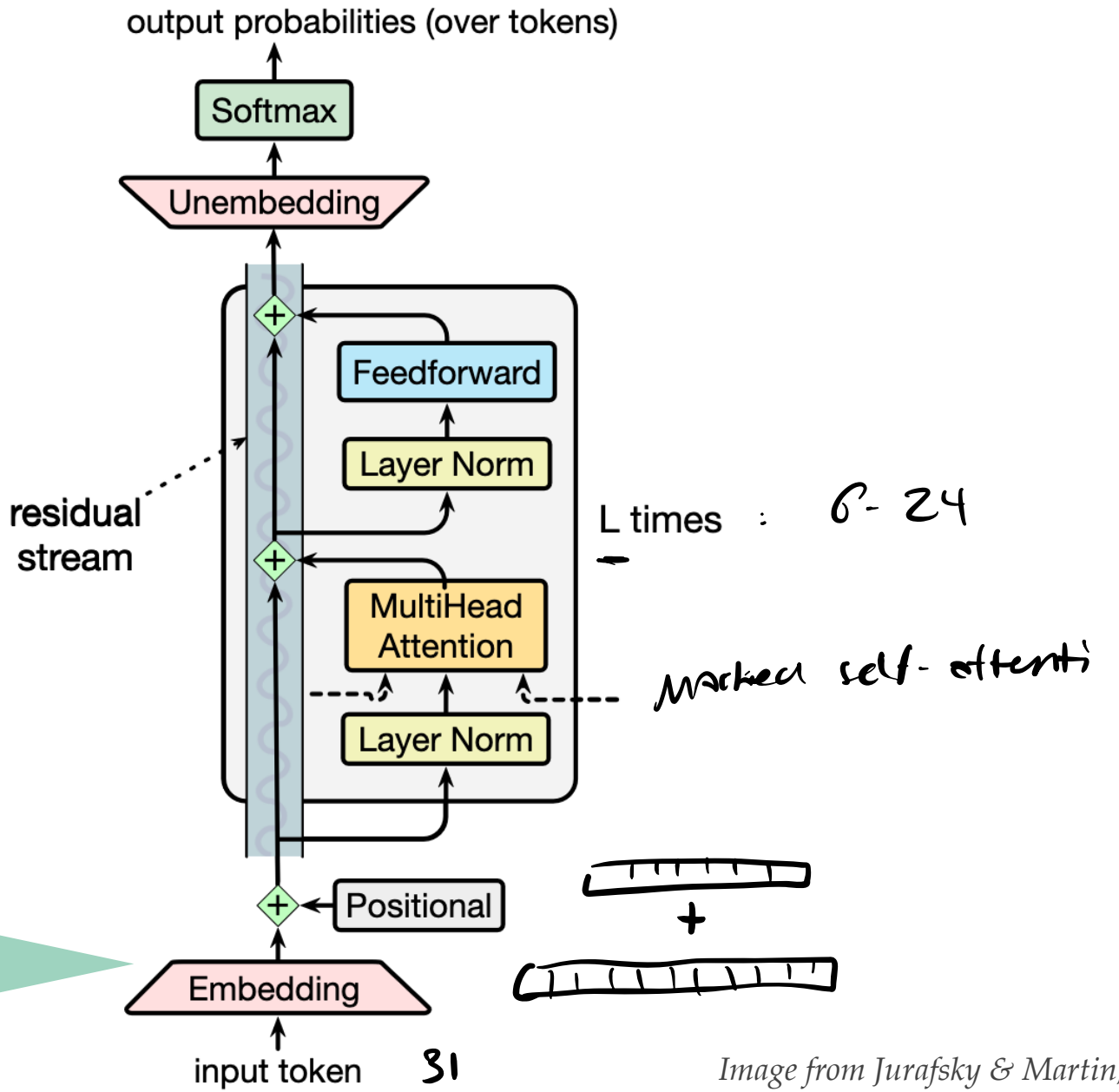
Masked Self-Attention

Problem: we're cheating by letting "Nobel" look at the words that come after it!



Transformers

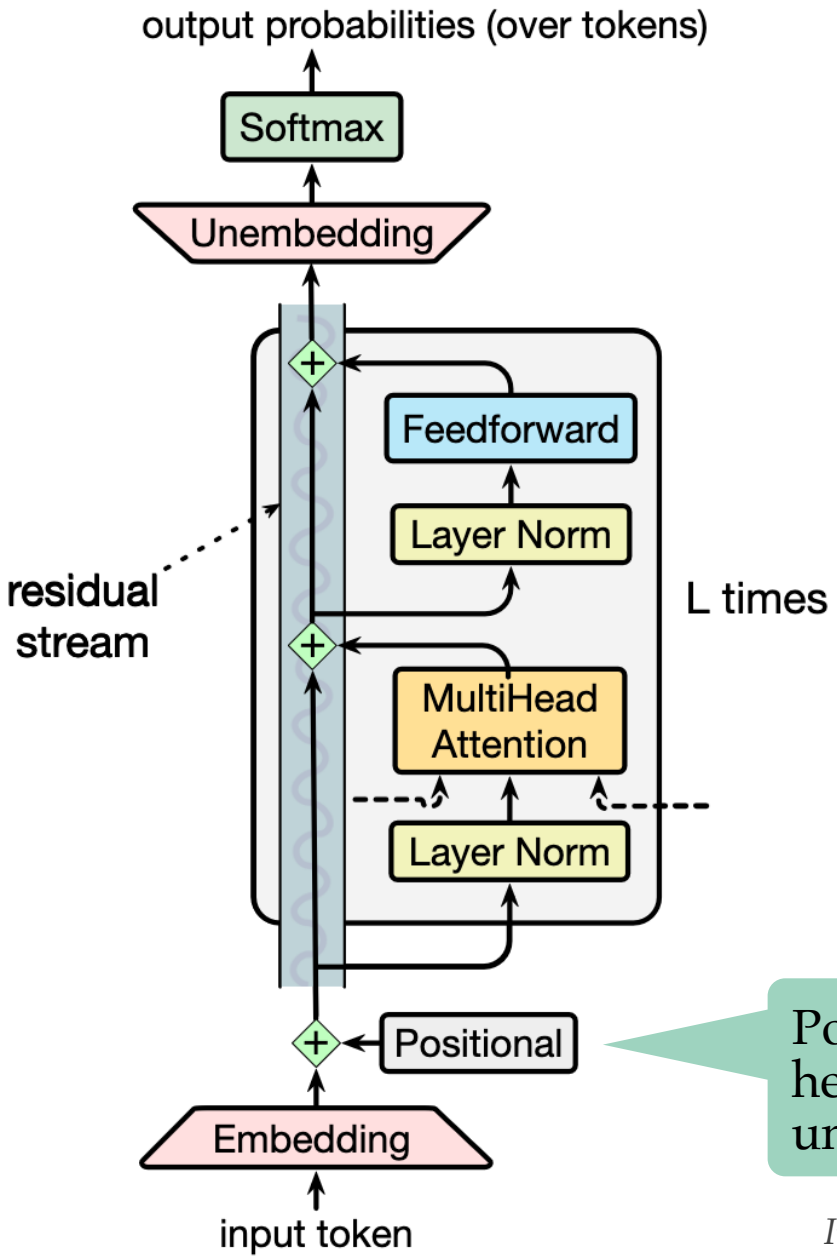
Decoder-Only Transformer



Embedding layer goes from word ids to embeddings

Image from Jurafsky & Martin, Chapt. 7

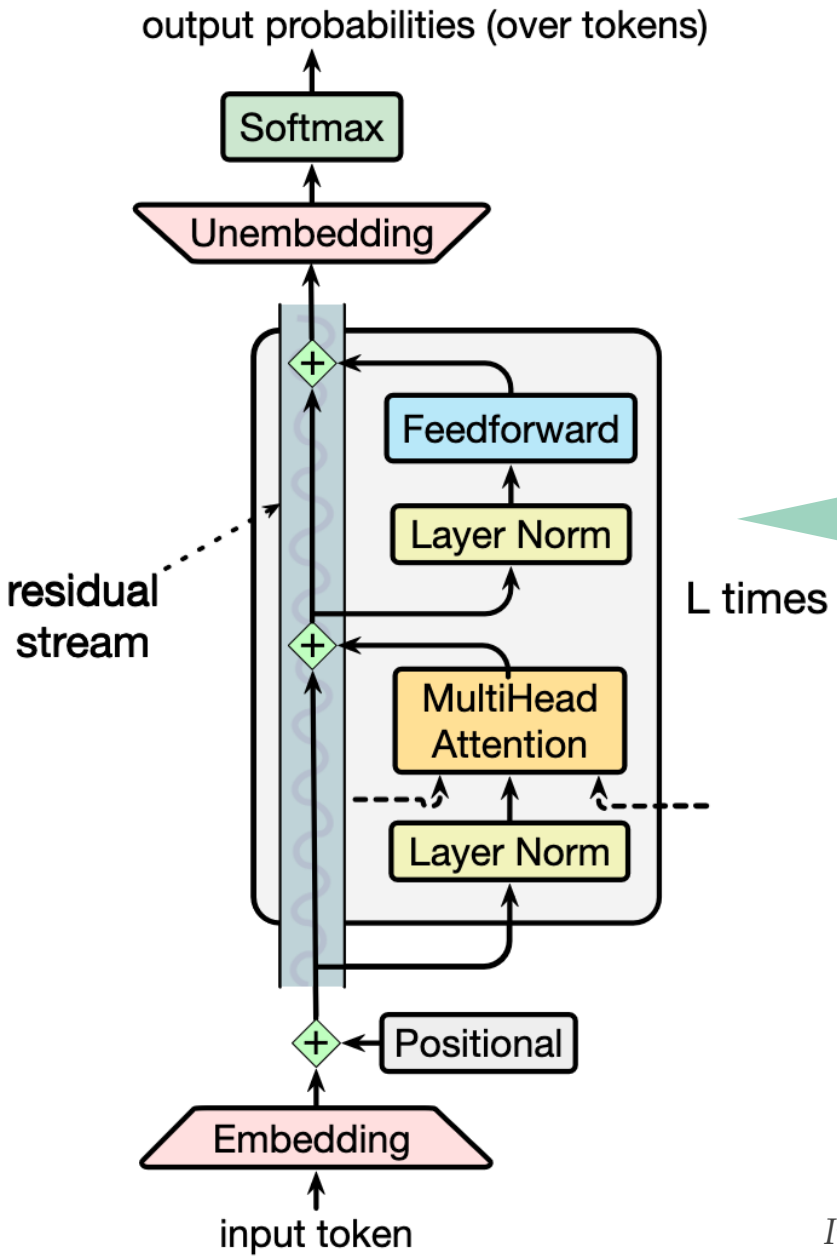
Decoder-Only Transformer



Positional embeddings help the model understand word order

Image from Jurafsky & Martin, Chapt. 7

Decoder-Only Transformer



Many attention blocks are stacked atop each other, interleaved with feedforward layers.

Image from Jurafsky & Martin, Chapt. 7

Decoder-Only Transformer

After stacking a bunch of these attention blocks, we finally have our familiar softmax layer to predict the next word

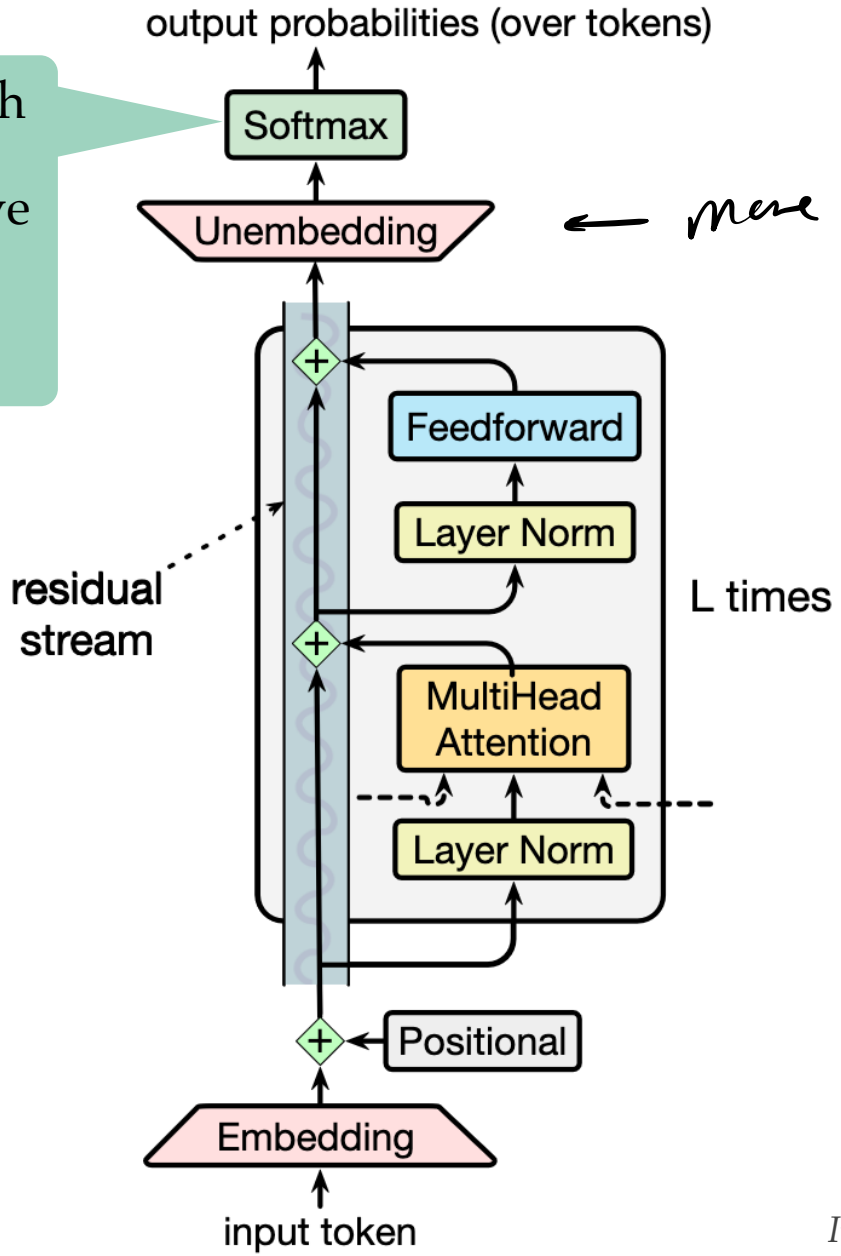


Image from Jurafsky & Martin, Chapt. 7

Text Generation

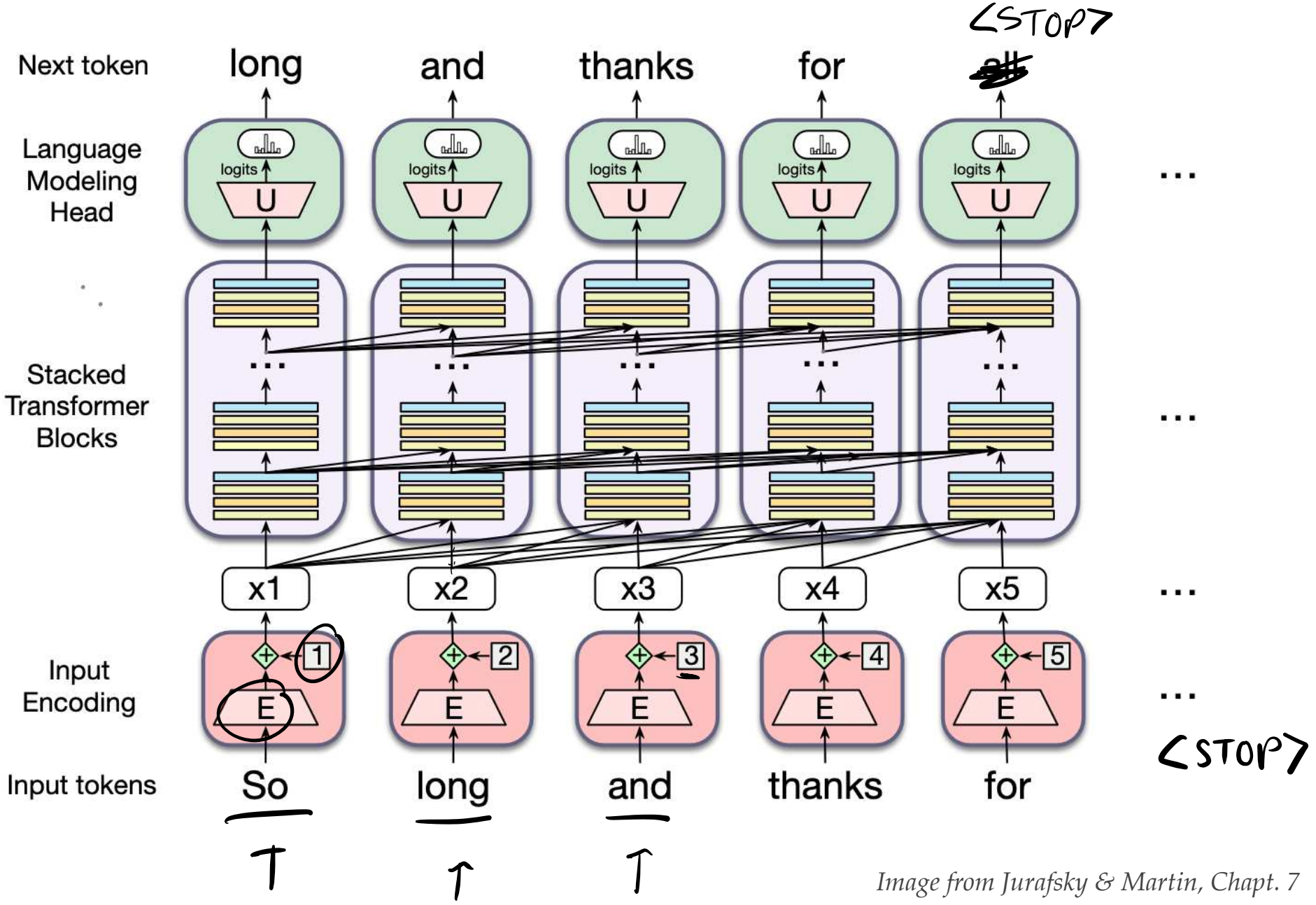


Image from Jurafsky & Martin, Chapt. 7