CS 232: Artificial Intelligence

Spring 2024

Prof. Carolyn Anderson
Wellesley College

Reminders

- No class next Friday: I will be giving an invited talk at a symposium at Washington University.
  I'll post a recorded video lecture and hold help hours on Thursday morning.
- I have help hours today from 3:30-4:30
- Lyra has help hours on Sunday

Bonus Late Day Opportunity

AI for Wireless and Wireless for AI: A Tale of Two AIs

Francesco Restuccia
Northeastern University

4-5pm
April 23rd

Chiang (2023):
ChatGPT is a Blurry JPEG of the Web

[... H]allucinations are anything but surprising; if a compression algorithm is designed to reconstruct text after ninety-nine per cent of the original has been discarded, we should expect that significant portions of what it generates will be entirely fabricated.

If a large language model has compiled a vast number of correlations between economic terms—so many that it can offer plausible responses to a wide variety of questions—should we say that it actually understands economic theory?

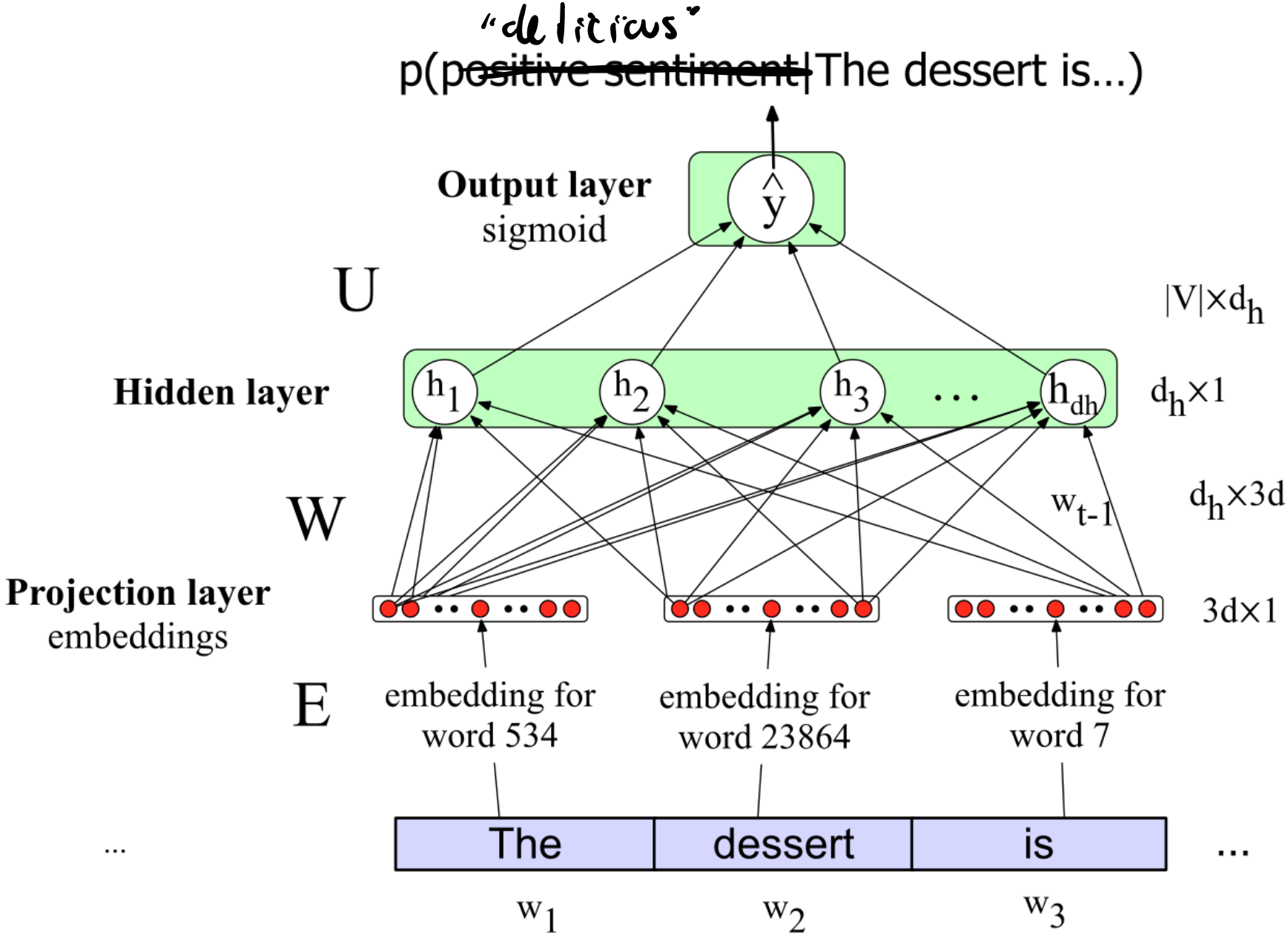
Imagine what it would look like if ChatGPT were a lossless algorithm. If that were the case, it would always answer questions by providing a verbatim quote from a relevant Web page. We would probably regard the software as only a slight improvement over a conventional search engine, and be less impressed by it.[...] When we're dealing with sequences of words, lossy compression looks smarter than lossless compression.

There’s a type of blurriness that is acceptable, which is the re-stating of information in different words. Then there’s the blurriness of outright fabrication, which we consider unacceptable when we’re looking for facts.

Some might say that the output of large language models doesn't look all that different from a human writer's first draft, but, again, I think this is a superficial resemblance. Your first draft isn't an unoriginal idea expressed clearly; it's an original idea expressed poorly, and it is accompanied by your amorphous dissatisfaction, your awareness of the distance between what it says and what you want it to say.

Indeed, a useful criterion for gauging a large language model's quality might be the willingness of a company to use the text that it generates as training material for a new model. If the output of ChatGPT isn't good enough for GPT-4, we might take that as an indicator that it's not good enough for us, either.

Neural Net Classification with embeddings as input features!



Evaluation: How good is our model?

Does our language model prefer good sentences to bad ones?

Does it assign higher probability to “real” or “frequently observed” sentences than “ungrammatical” or “rarely observed” sentences?

Perplexity

The best language model is one that **best predicts an unseen test set** (gives the highest P(sentence)).

Perplexity is the **inverse probability of the test set**, **normalized by the number of words.**

$$PP(W) = \mathrm{P}(w_1 w_2 \ldots w_N)^{-\frac{1}{N}}$$

$$= \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N)}}$$

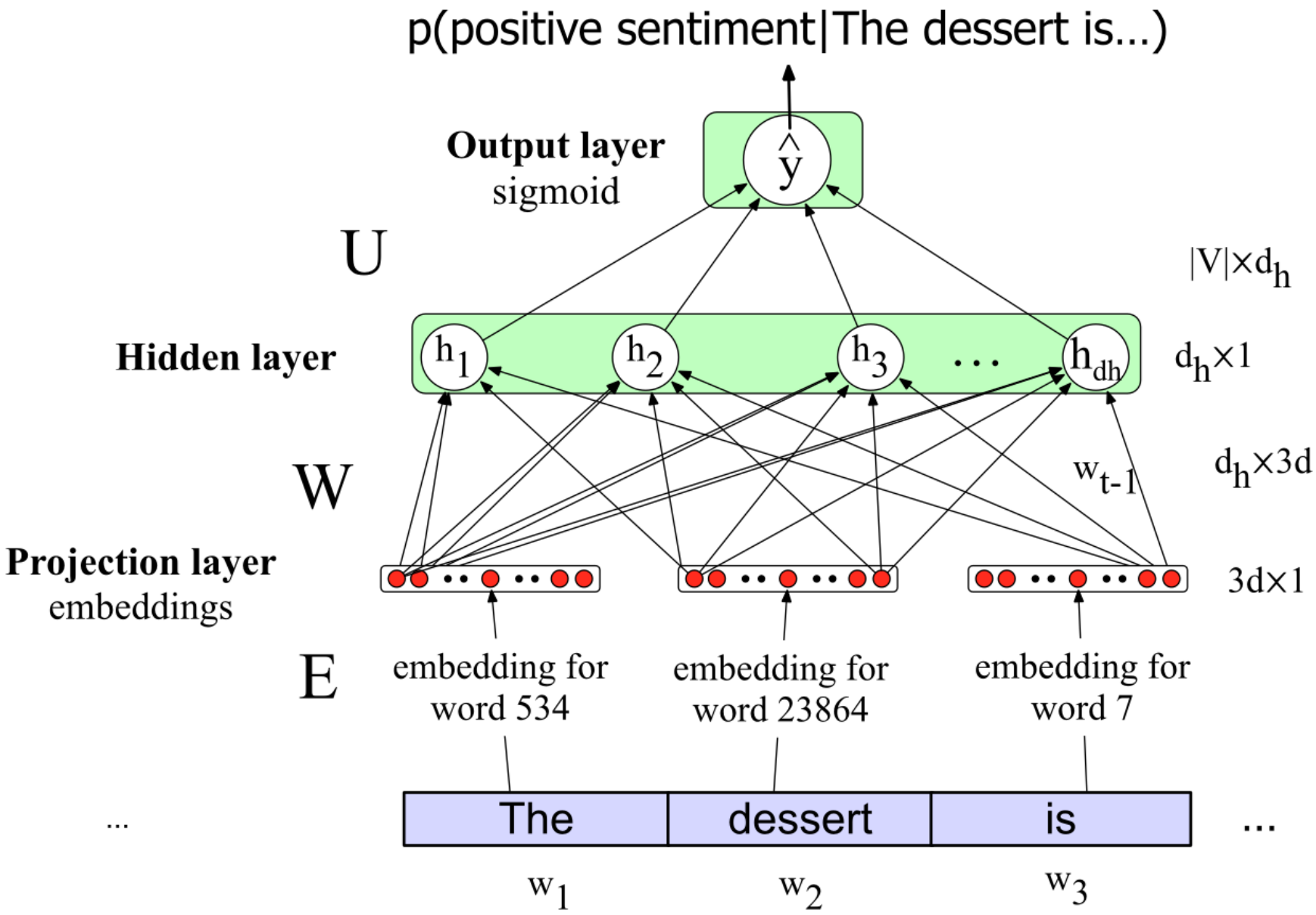
Minimizing perplexity is the same as maximizing probability

Lower perplexity = better model

Training 38 million words, test 1.5 million words, WSJ

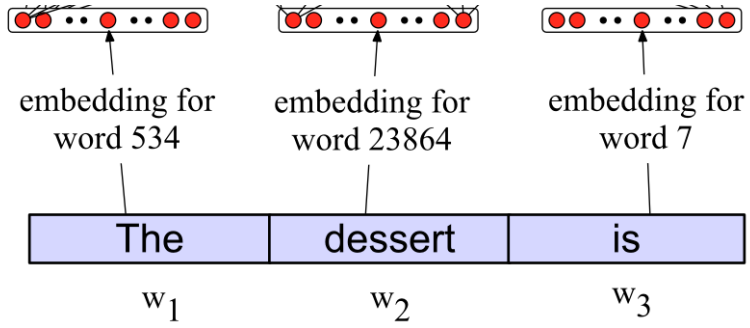
N-gram Order	Unigram	Bigram	Trigram
Perplexity	962	170	109

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.

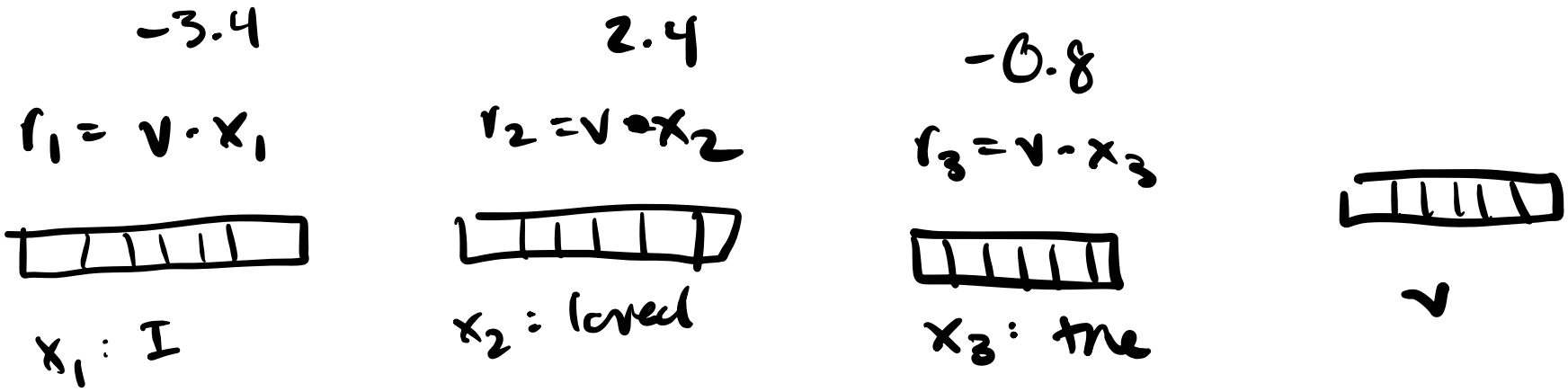
Attention

What Is Attention?

Learn a task-specific vector v

Intuition: v is an "important word" vector

$$a = \text{softmax}(r)$$



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:

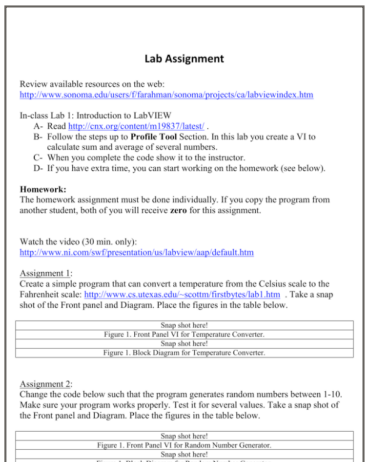
My brother, a chemist, was late yesterday because he missed the bus. When he arrived, he was surprised to find that his lab _____

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.

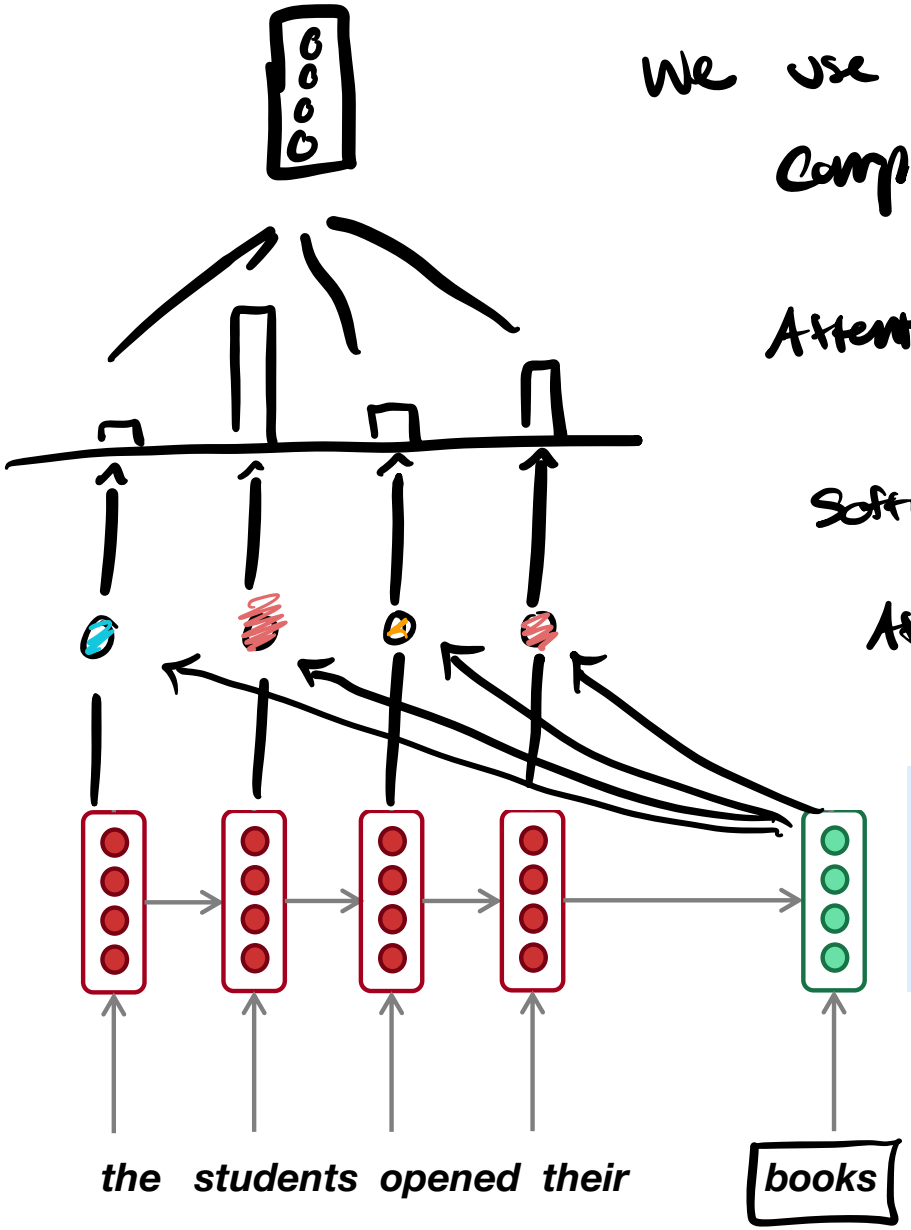
Snap shot here!
Figure 1. Front Panel VI for Temperature Converter.
Snap shot here!
Figure 1. Block Diagram for Temperature Converter.

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



We use the attention distribution to compute a weighted average over the word embeddings

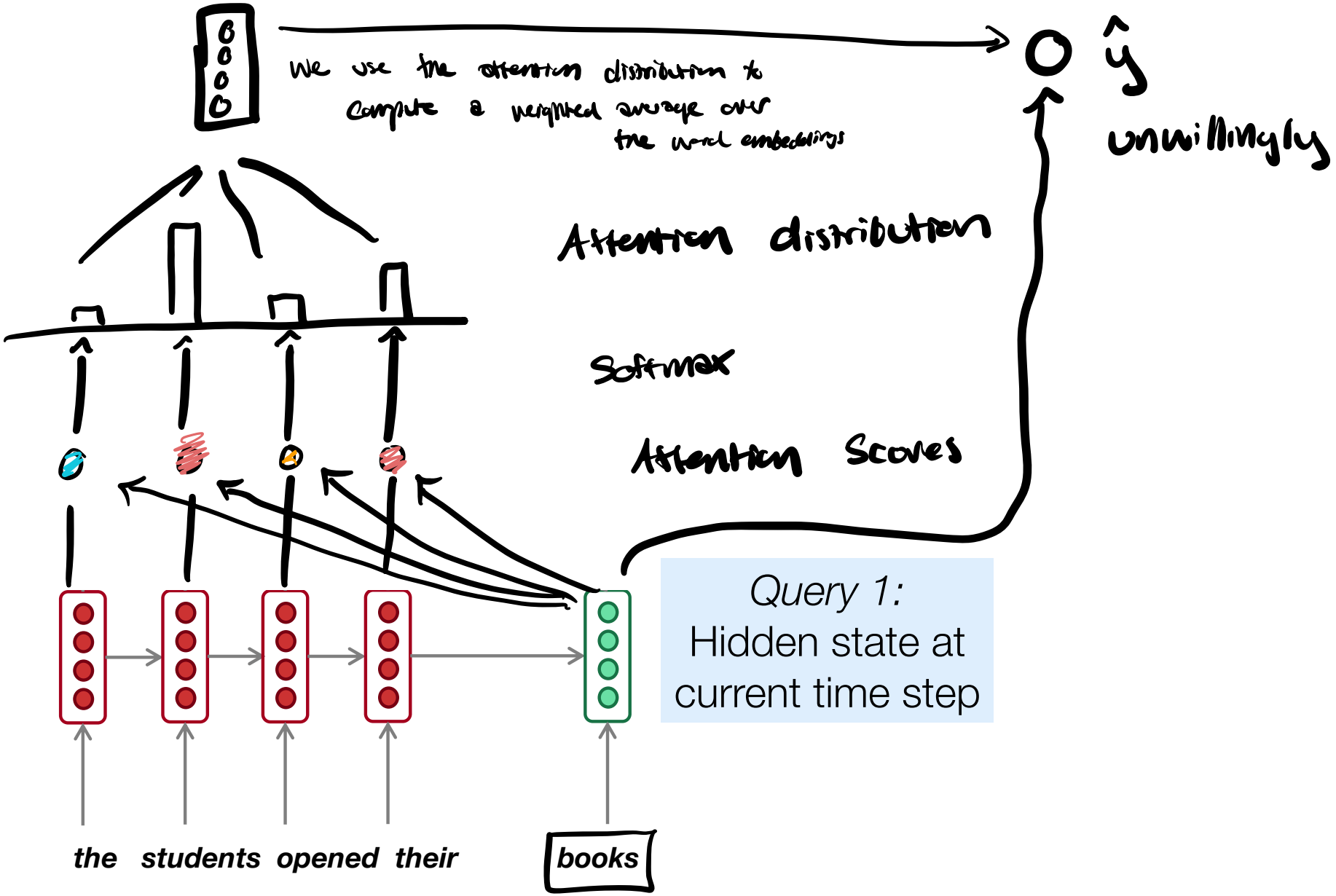
Attention distribution

Softmax

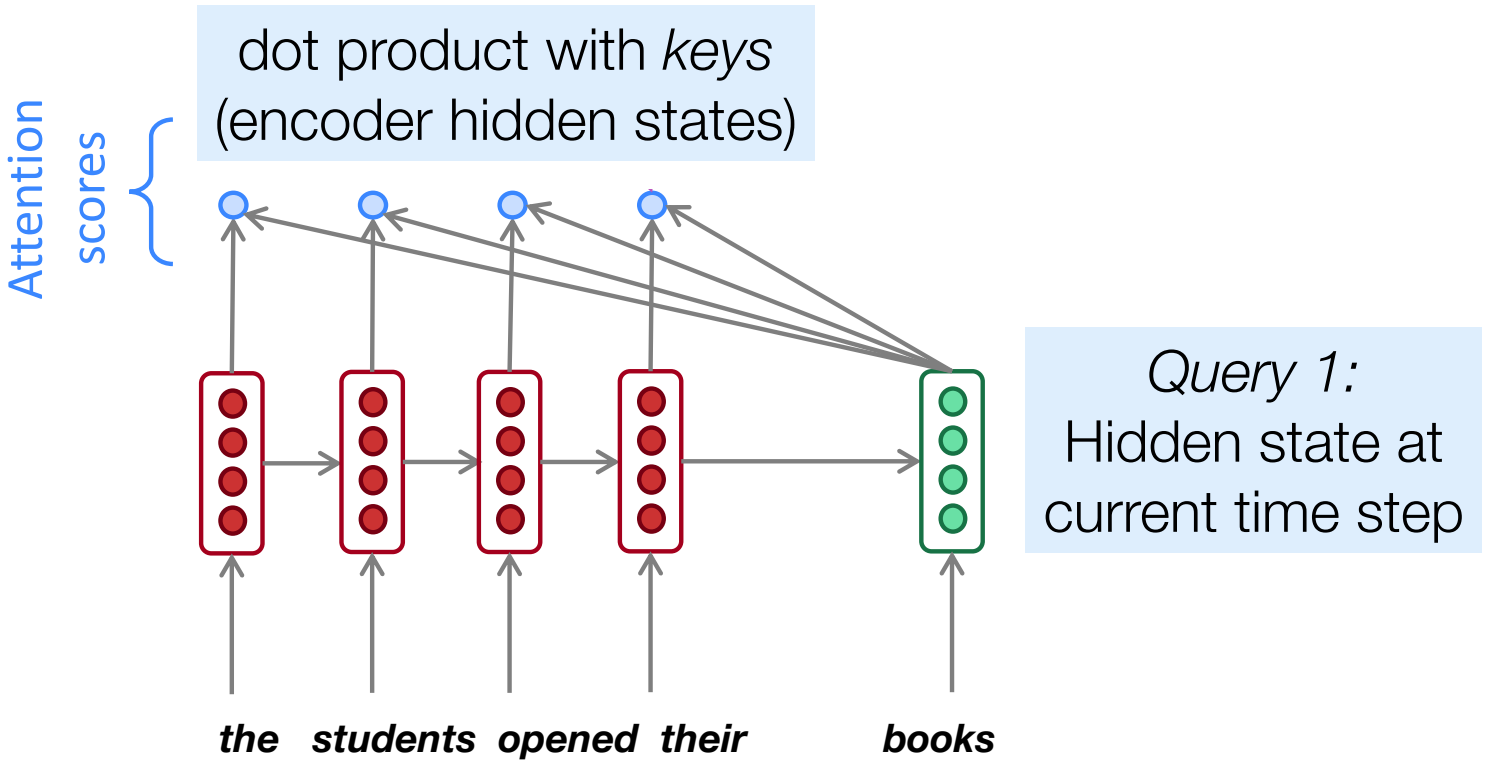
Attention Scores

Query 1:
Hidden state at current time step

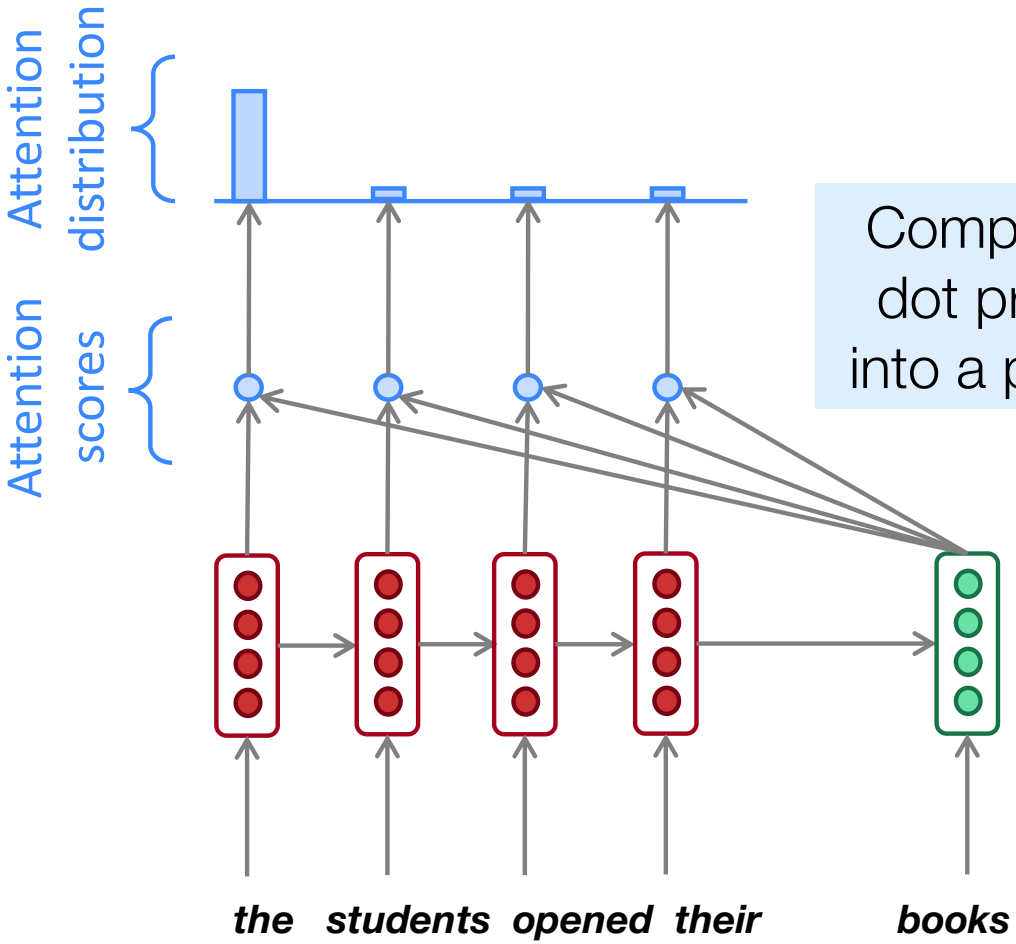
Attention mechanisms in neural language models



Attention mechanisms in neural language models

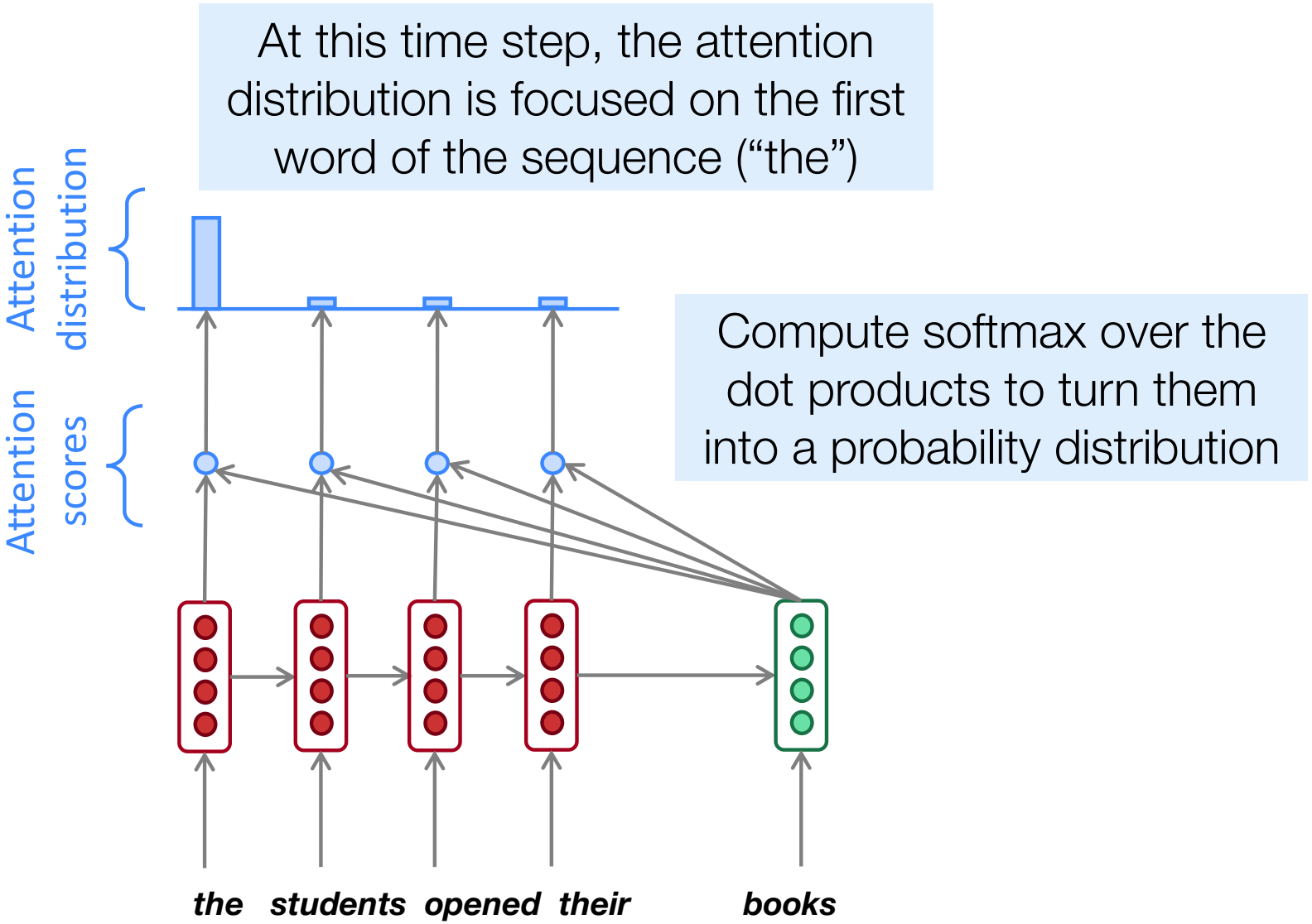


Attention mechanisms in neural language models

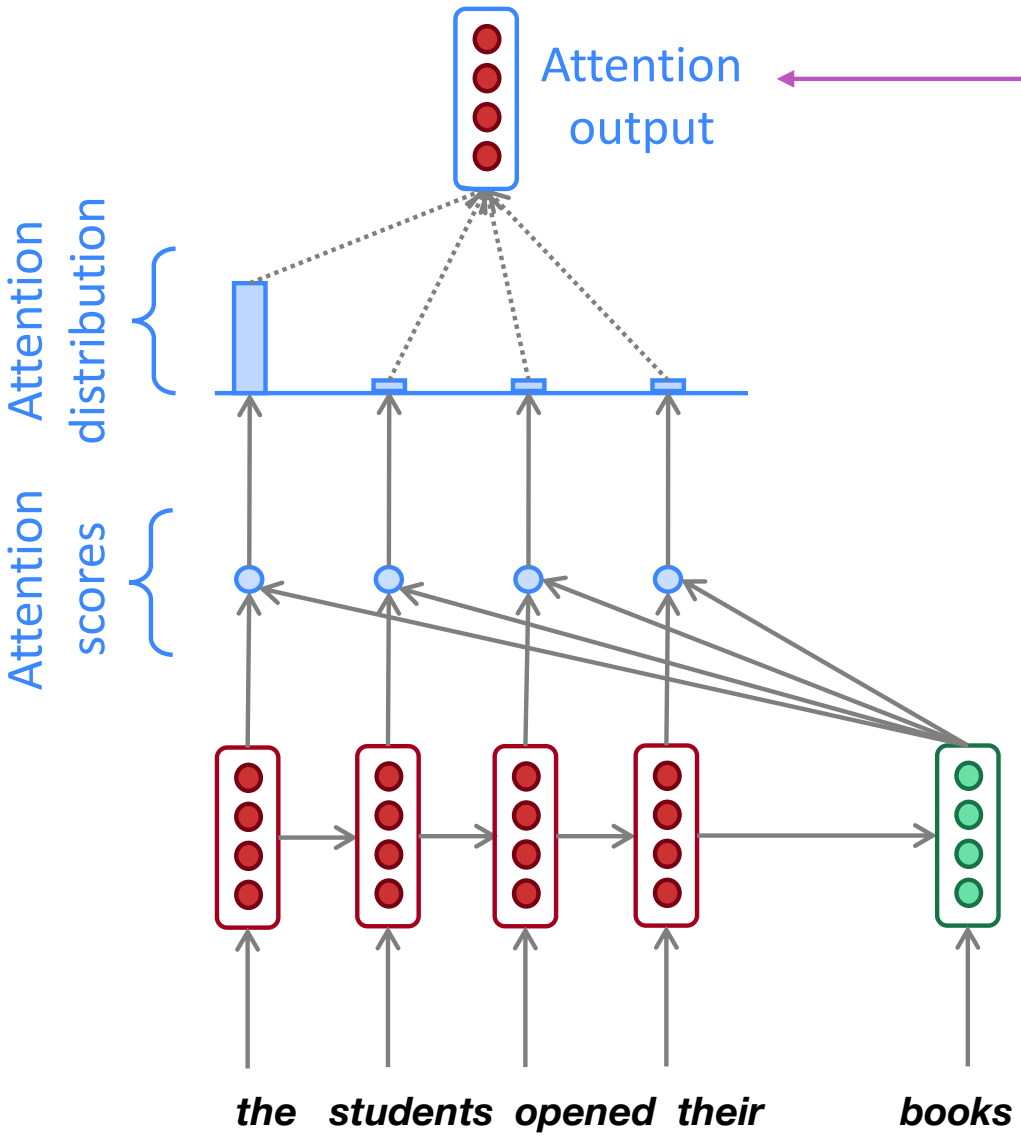


Compute softmax over the dot products to turn them into a probability distribution

Attention mechanisms in neural language models



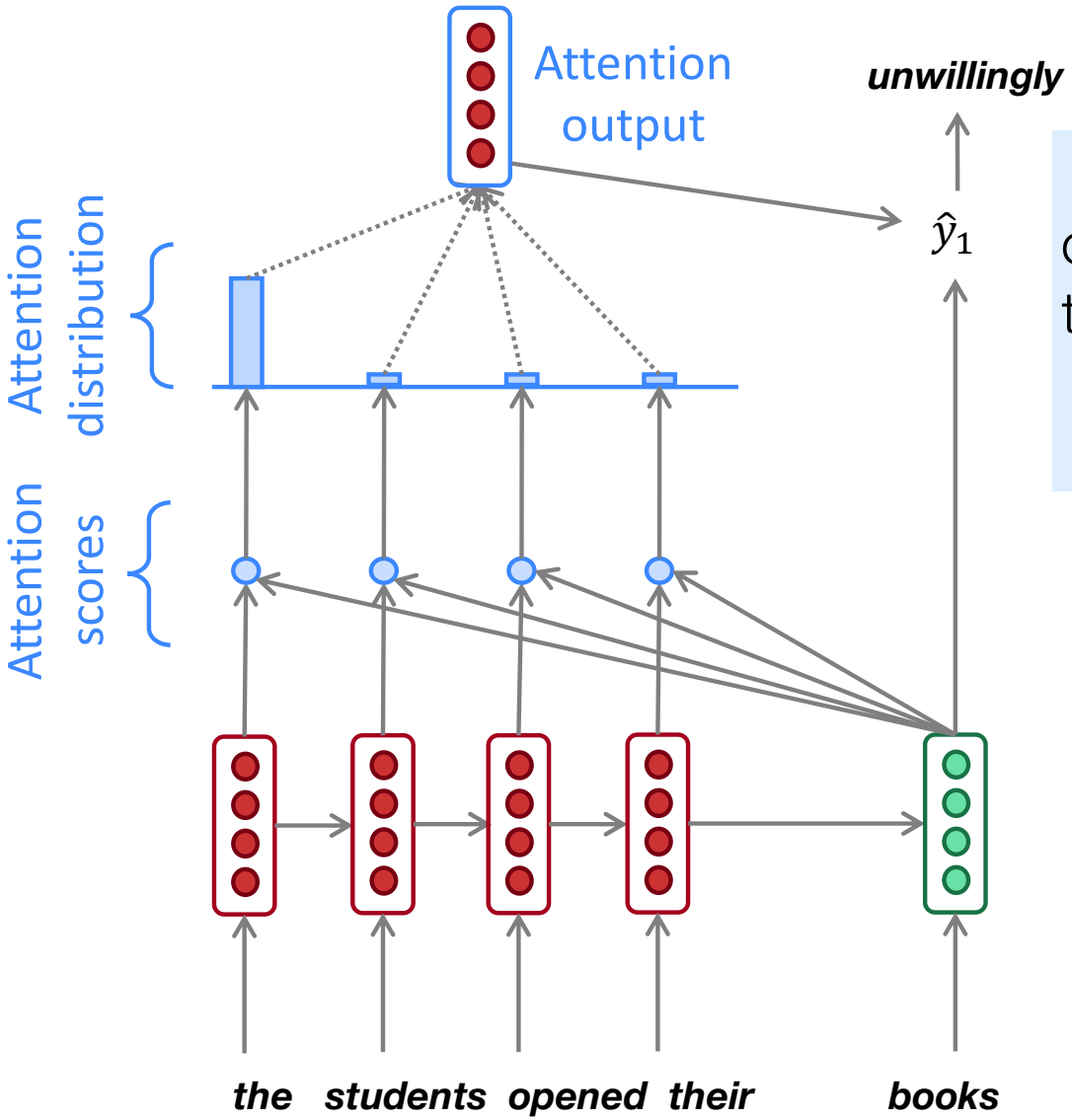
Attention mechanisms in neural language models



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

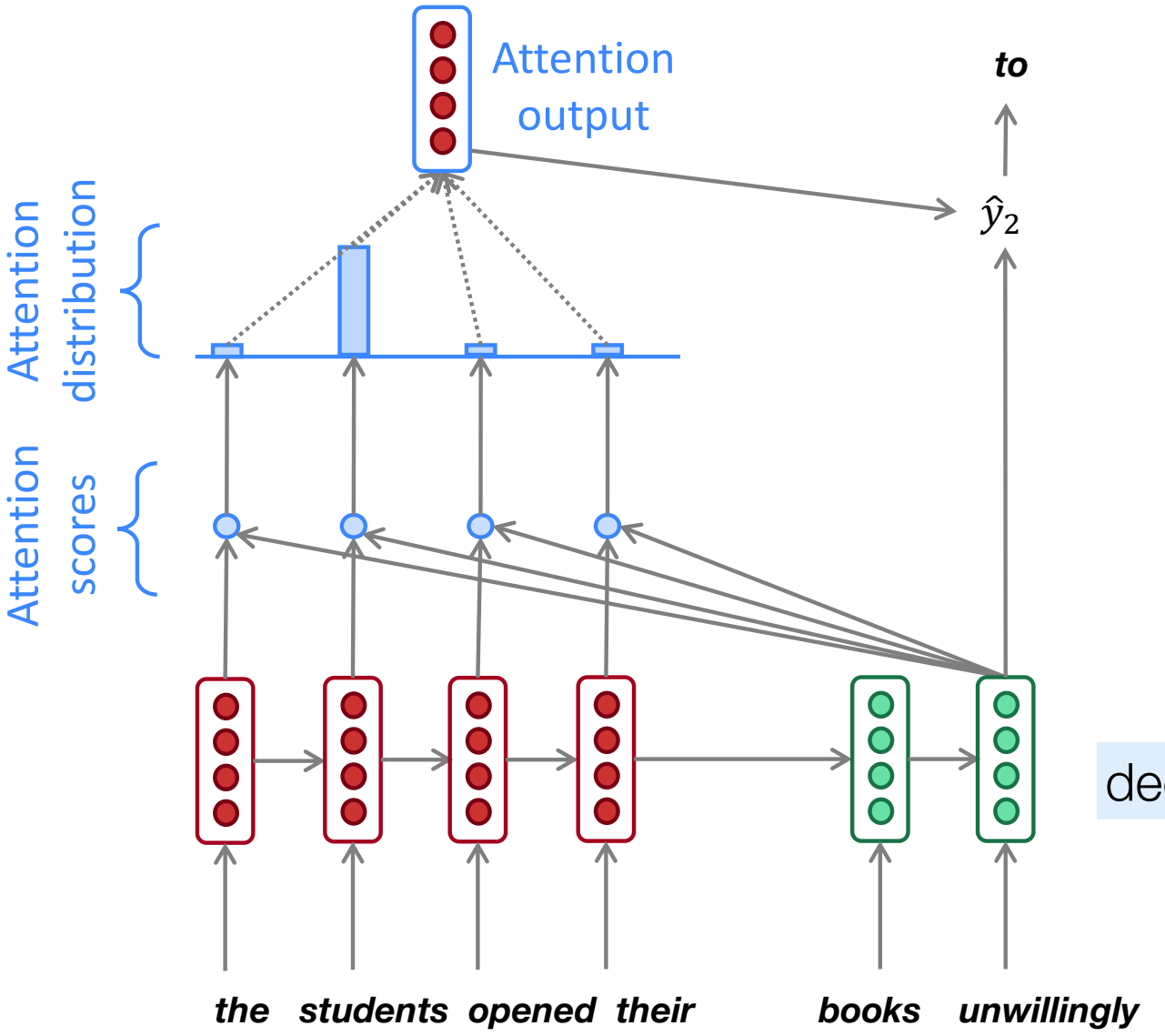
Intuitively, the resulting attention output contains information from hidden states that received high attention scores

Sequence-to-sequence with attention

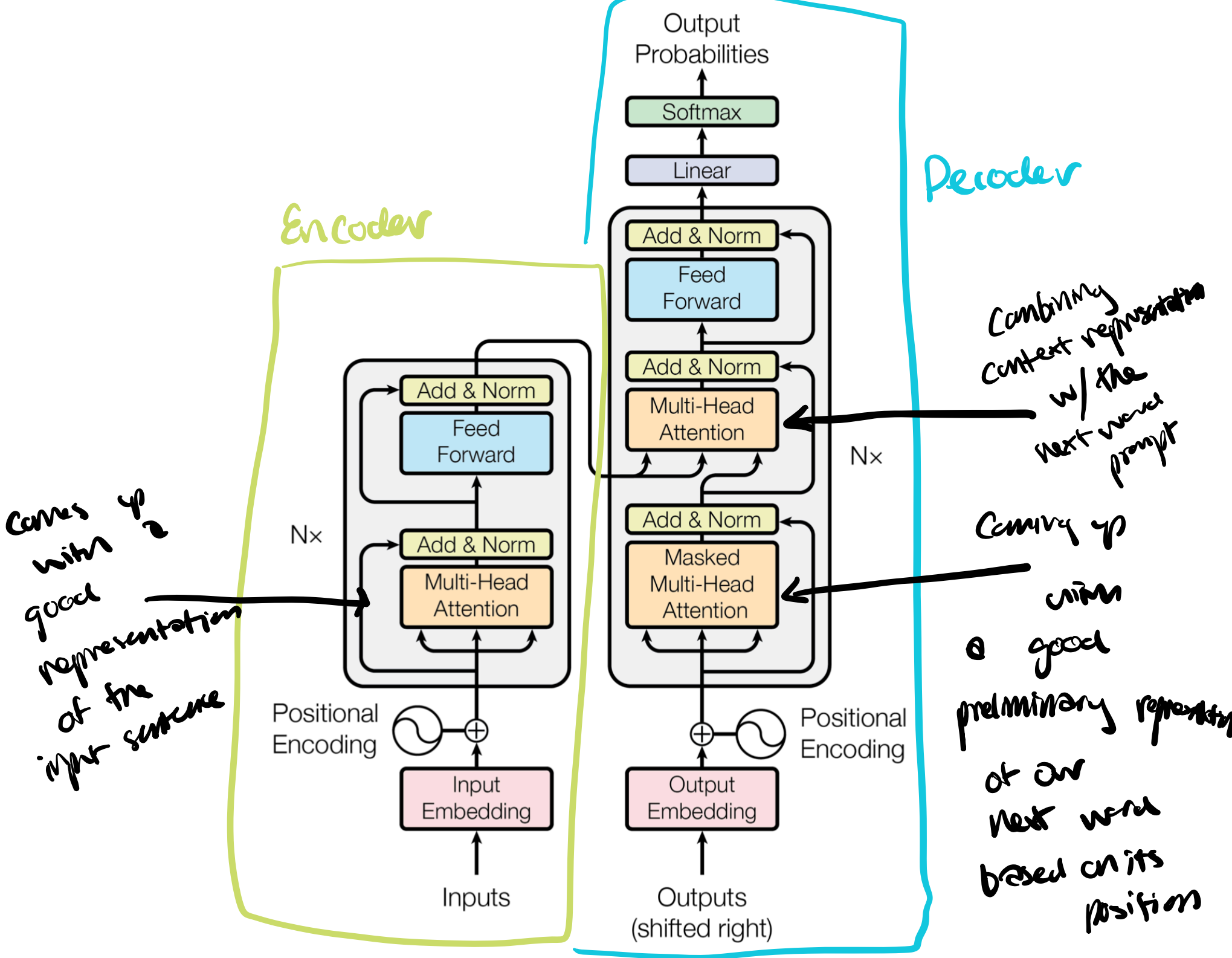


Concatenate (or otherwise compose) the attention output with the current hidden state, then pass through a softmax layer to predict the next word

Sequence-to-sequence with attention



decoder, second time step


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Positional Encoding
Input Embedding
Inputs
Output Embedding
Outputs (shifted right)
Encoder
Decoder
Comes up with a good representation of the input sentence
Combining context representation w/ the next word prompt
Coming up with a good preliminary representation of our next word based on its position