Supervised Learning: Deep(er) learning

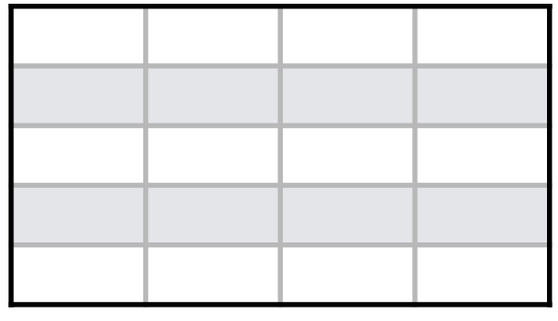
Jean Feng & Ali Shojaie

Aug 28-30, 2025
Summer Institute in Statistics for Big Data
University of Washington

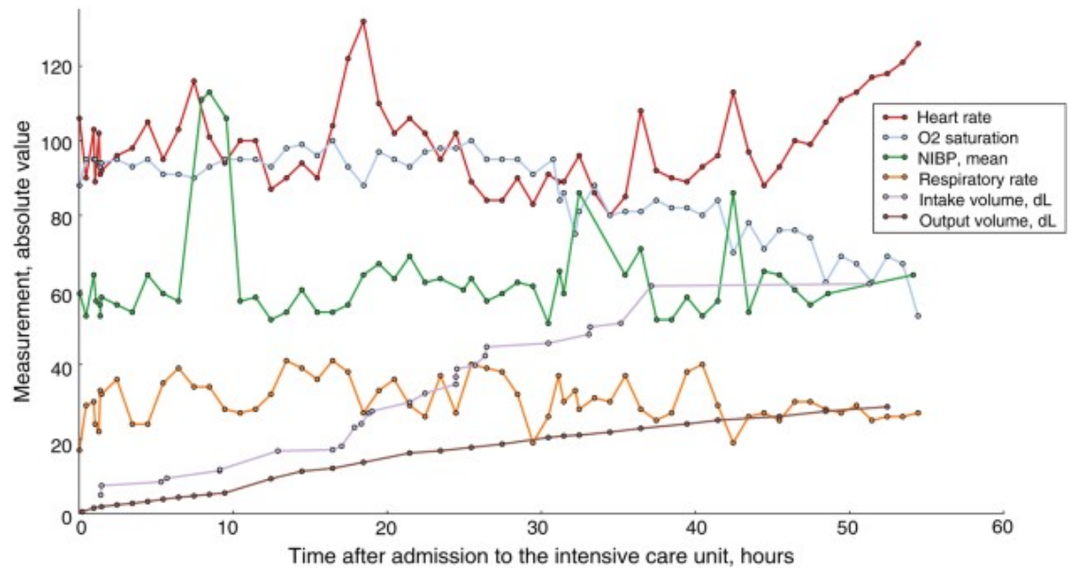
Sequence data

- Up to now, we've assumed all inputs to a machine learning model have the same exact size
    - Images
    - Tabular data
- What happens if we are trying to predict labels using sequential data, e.g.
    - Text data
    - Time series data
    - Genetic sequences
    - Video analysis
    - …

Tabular data:

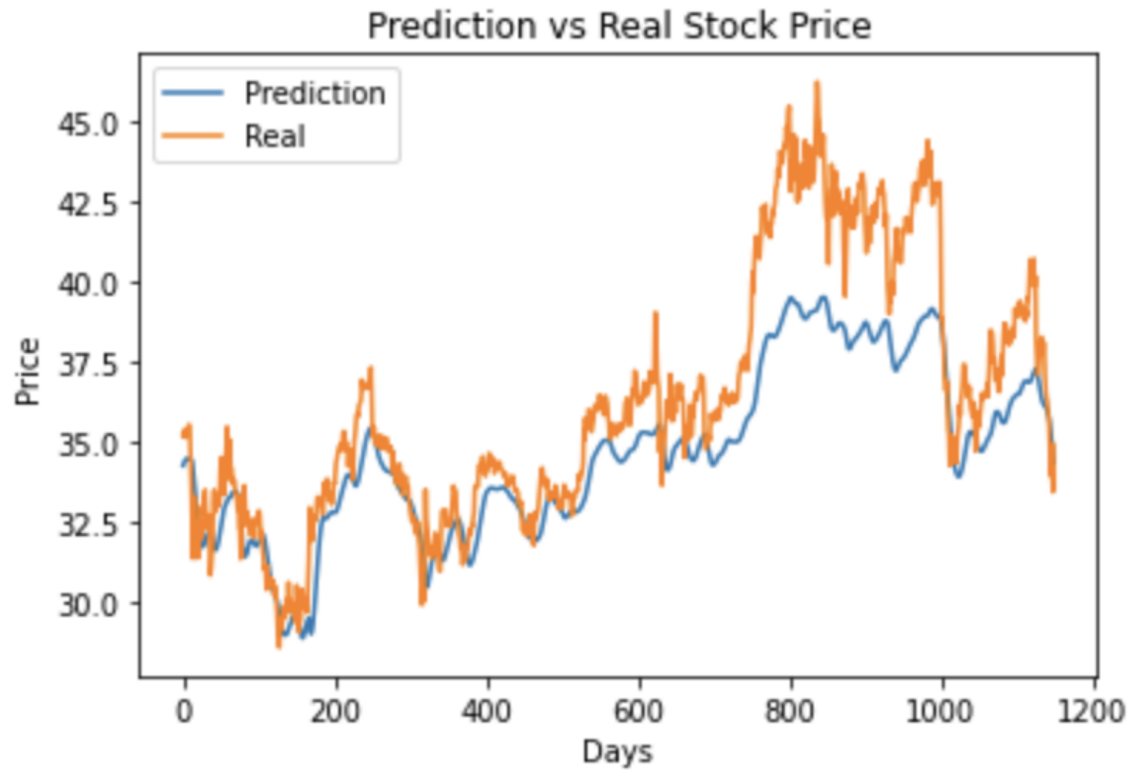


Patient data from the ICU:



Example tasks for sequence data

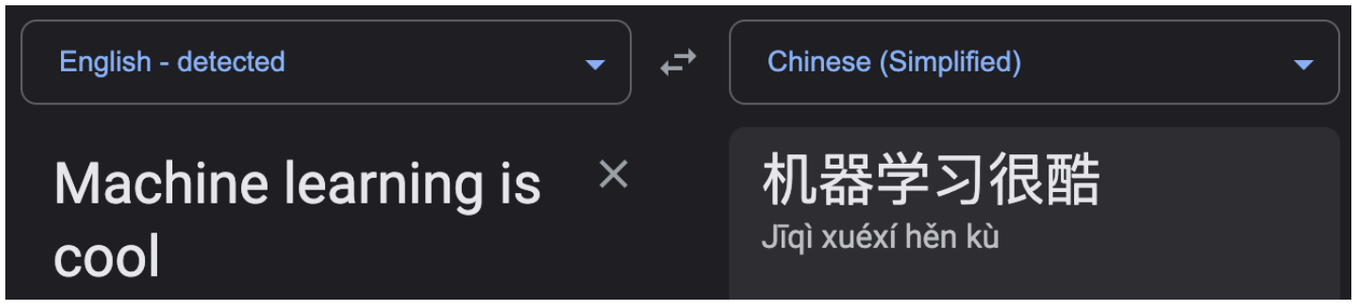
- **Auto-complete:**
  - *Text*: The patient said _____
  - *Time series data*:



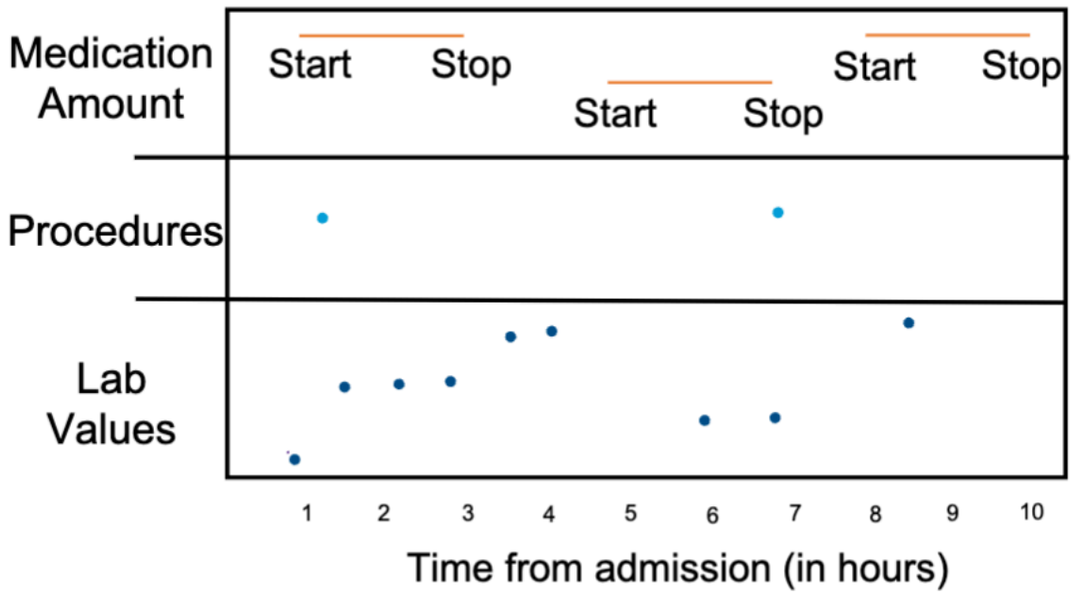
Do this well and you'll be rich! ;)

Example tasks for sequence data

- **Translation:**



- **Classification:**



Readmission within 30-days?

Example tasks for sequence data

Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning

Jiasen Lu[2*†], Caiming Xiong[1†], Devi Parikh[3], Richard Socher[1]
[1]Salesforce Research, [2]Virginia Tech, [3]Georgia Institute of Technology
jiasenlu@vt.edu, parikh@gatech.edu, {cxiong, rsocher}@salesforce.com

Figure 4: Visualization of generated captions and image attention maps on the COCO dataset. Different colors show a correspondence between attended regions and underlined words. First 2 rows are success cases, last rows are failure examples. Best viewed in color.

Lu et al, CVPR 2017

How do humans process text data?

+

Does this refer to a brain network? A gene network?

Deep learning is the subset of machine learning methods based on neural networks with representation learning. The adjective "deep" refers to the use of multiple layers in the network. Methods used can be either supervised, semi-supervised or unsupervised.[2]

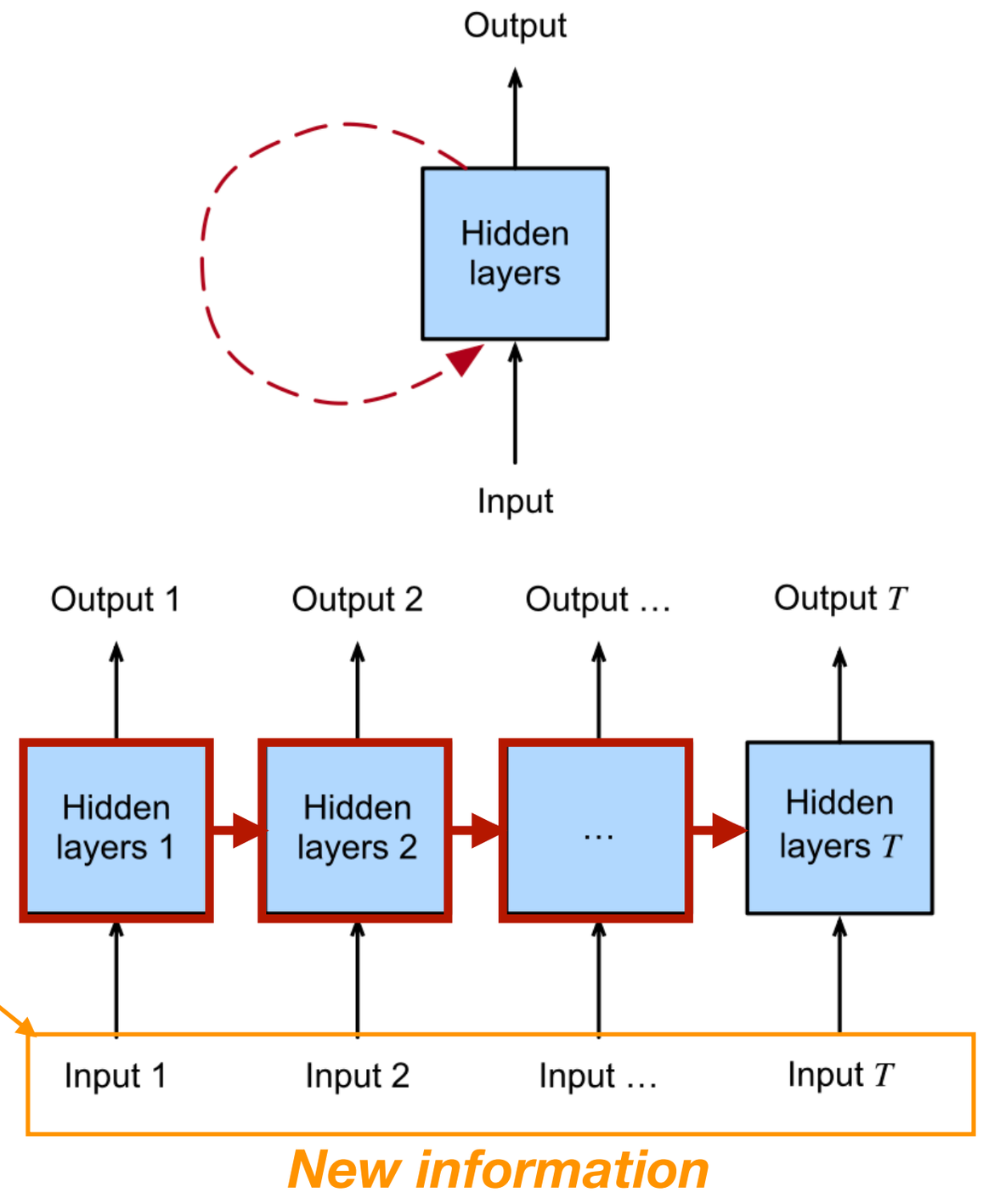
Deep-learning architectures such as deep neural networks, deep belief networks, recurrent neural networks, convolutional neural networks and transformers have been applied to fields including computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, climate science, material inspection and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance.[3][4][5]

Early forms of neural networks were inspired by information processing and distributed communication nodes in biological systems, in particular the human brain. However, current neural networks do not intend to model the brain function of organisms, and are generally seen as low quality models for that purpose.[6]

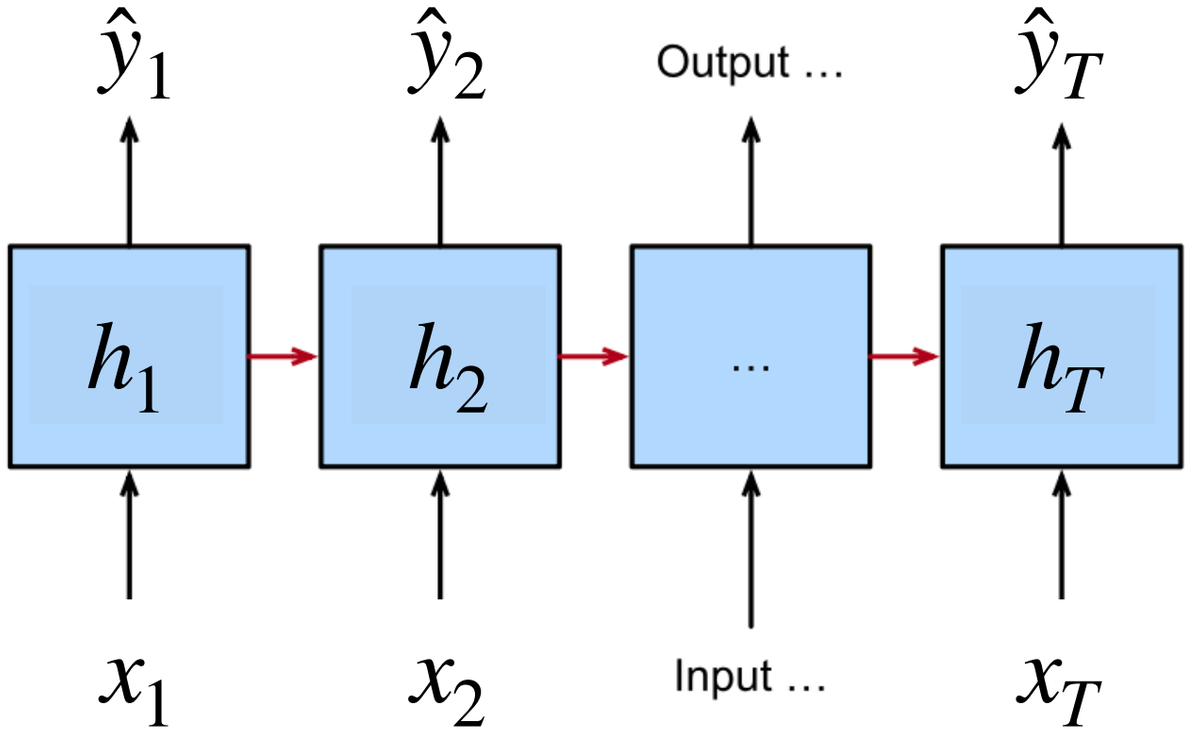
Recurrent neural networks

- **Key idea:** Define an NN that will be applied in the same manner to each position in the sequence. To make a prediction at position t, the inputs to the RNN are:
  - The input at position t
  - (Intermediate) Outputs at position $t-1$

"Memory"



Any ideas how we make this into math?

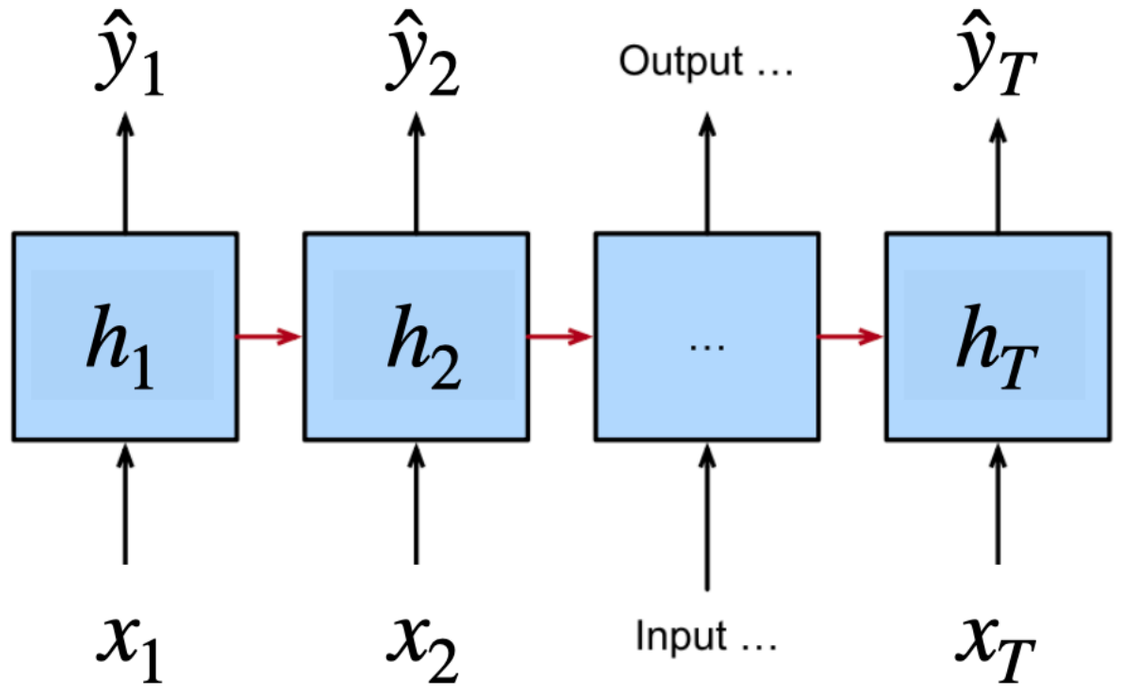


- $x_t =$ Input at position t
- $\hat{y}_t =$ Prediction at position t
- $h_t =$ Hidden layer at position t, encodes “memory”

Recurrent neural networks

Notation:

- $x_t =$ Input at position t
- $\hat{y}_t =$ Prediction at position t
- $h_t =$ Hidden layer at position t, encodes “memory”



Example of a simple RNN:

$$h_t = \sigma_1(Uh_{t-1} + Wx_t)$$

$$\hat{y}_t = \sigma_2(Vh_t)$$

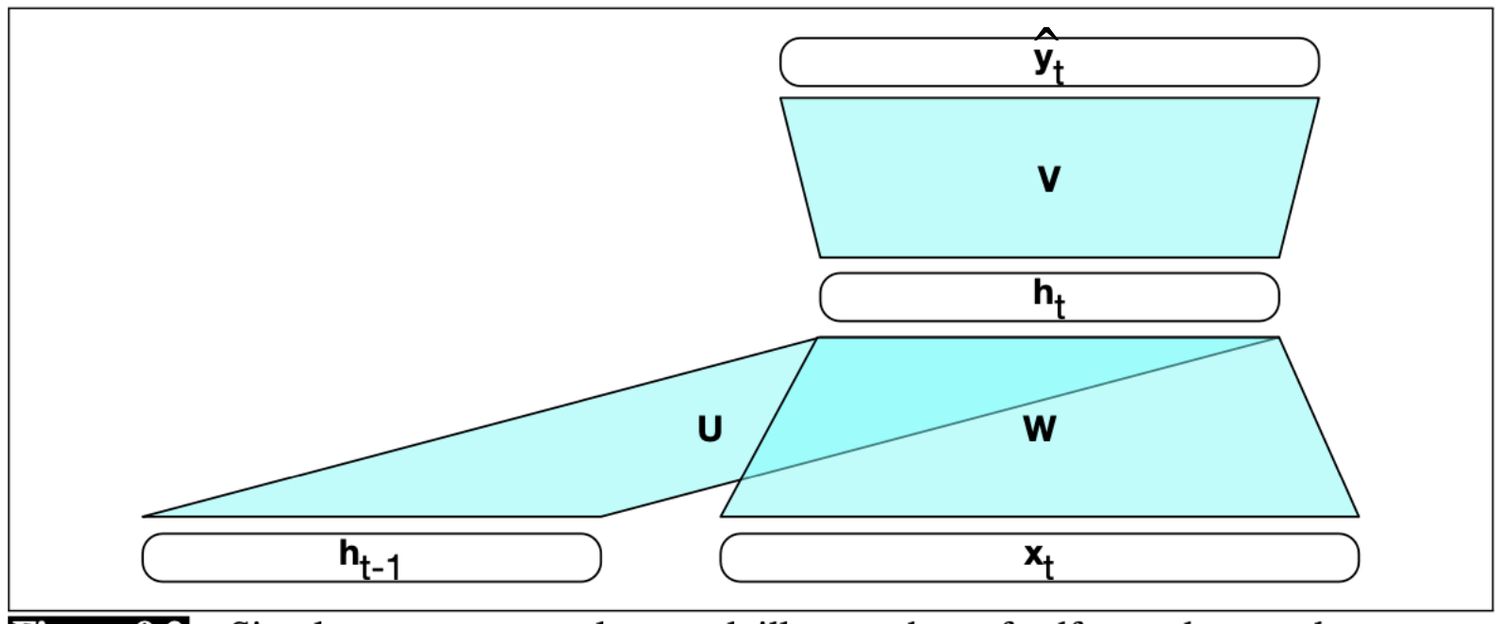


Figure 9.2 Simple recurrent neural network illustrated as a feedforward network.

Recurrent neural networks

Example of a simple RNN:

$$h_t = \sigma_1(Uh_{t-1} + Wx_t)$$

$$\hat{y}_t = \sigma_2(Vh_t)$$

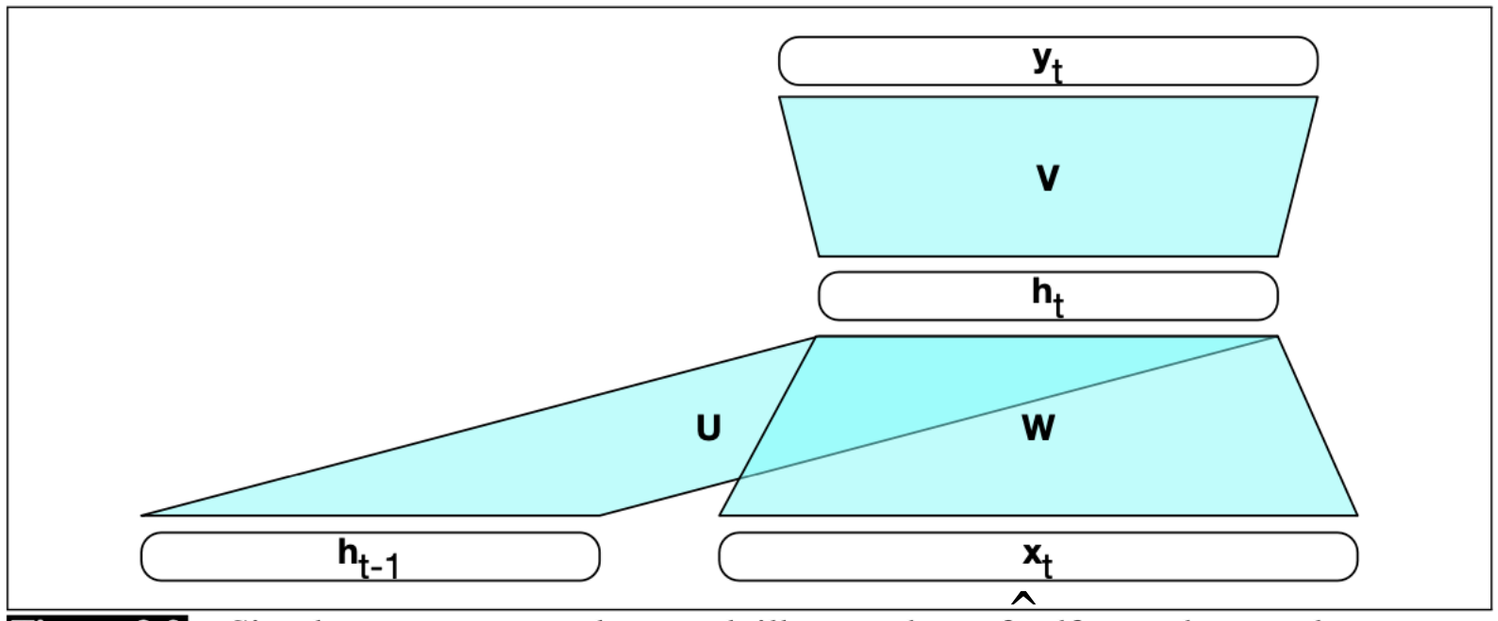


Figure 9.2 Simple recurrent neural network illustrated as a feedforward network.

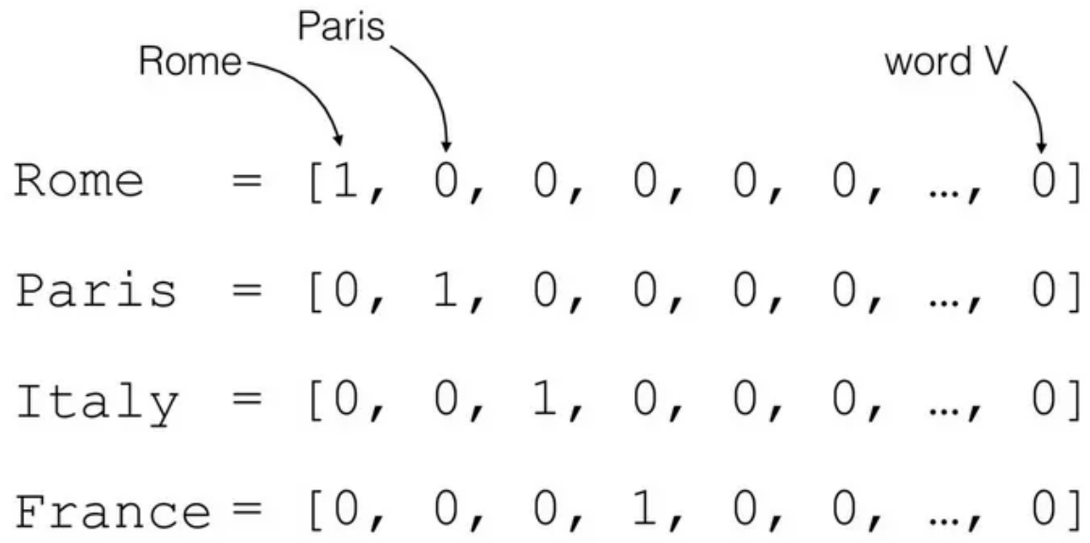
What are the parameters that need to be learned in this RNN?

RNNs for text

- Ok… but how do you encode a word as an input vector?
- How do we represent words to an RNN (i.e. input)?
- How do we words generated by an RNN (i.e. output)?

Example dataset

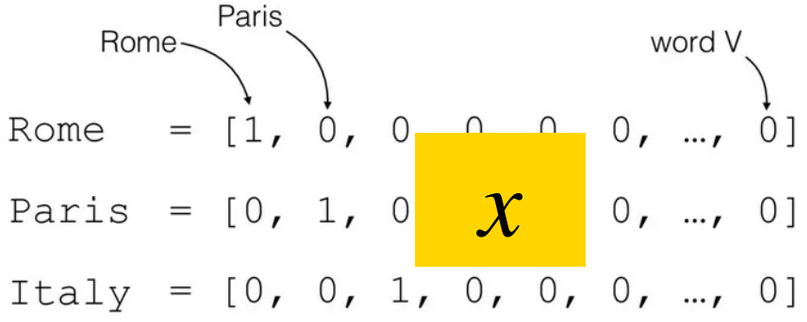
If I smelled the scent of hand sanitizers today on someone in the past, I would think they were so i...	How #COVID19 Will Change Work in General (and recruiting, specifically) via/ @ProactiveTalent #Recru...	#coronavirus #covid19 deaths continue to rise. It's almost as bad as it ever was. Politicians and ...

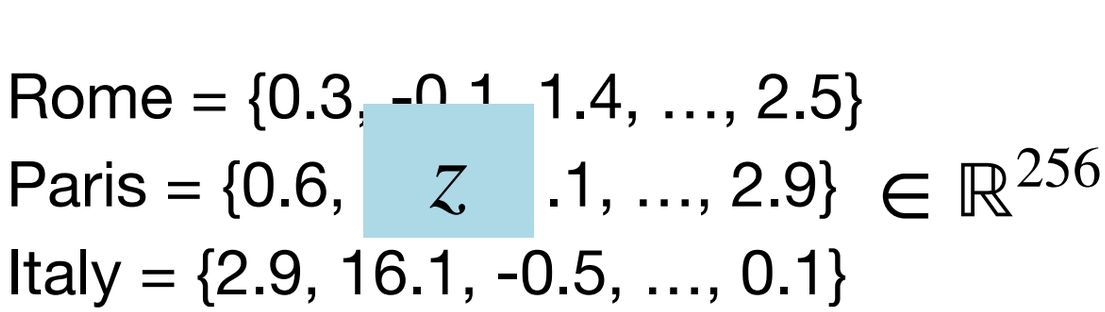


RNNs for text

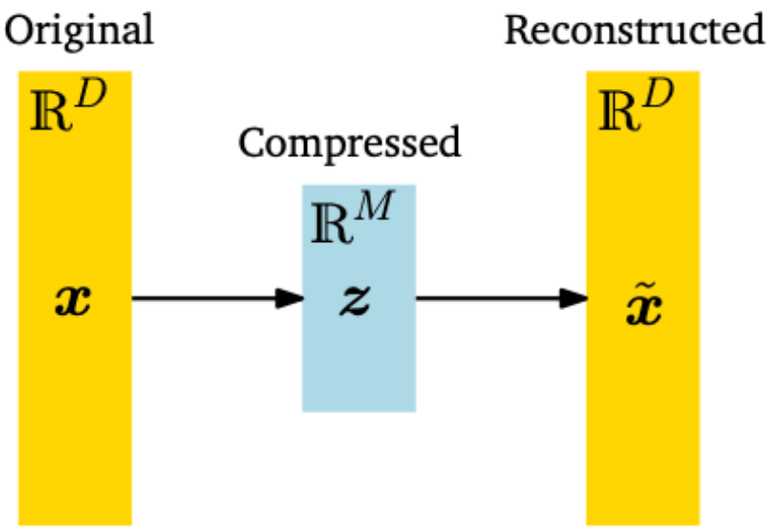
However, there are A LOT of words, so a one-hot encoding would be very high-dimensional.

What can we try to reduce the dimension of a high-dimensional dataset?





How about PCA?



$Ex = z$

RNNs for predicting text

Example of a simple RNN for text:

$$e_t = Ex_t$$

$$h_t = \text{softmax}(Uh_{t-1} + We_t)$$

$$\hat{p}_t = \text{softmax}(Vh_t)$$

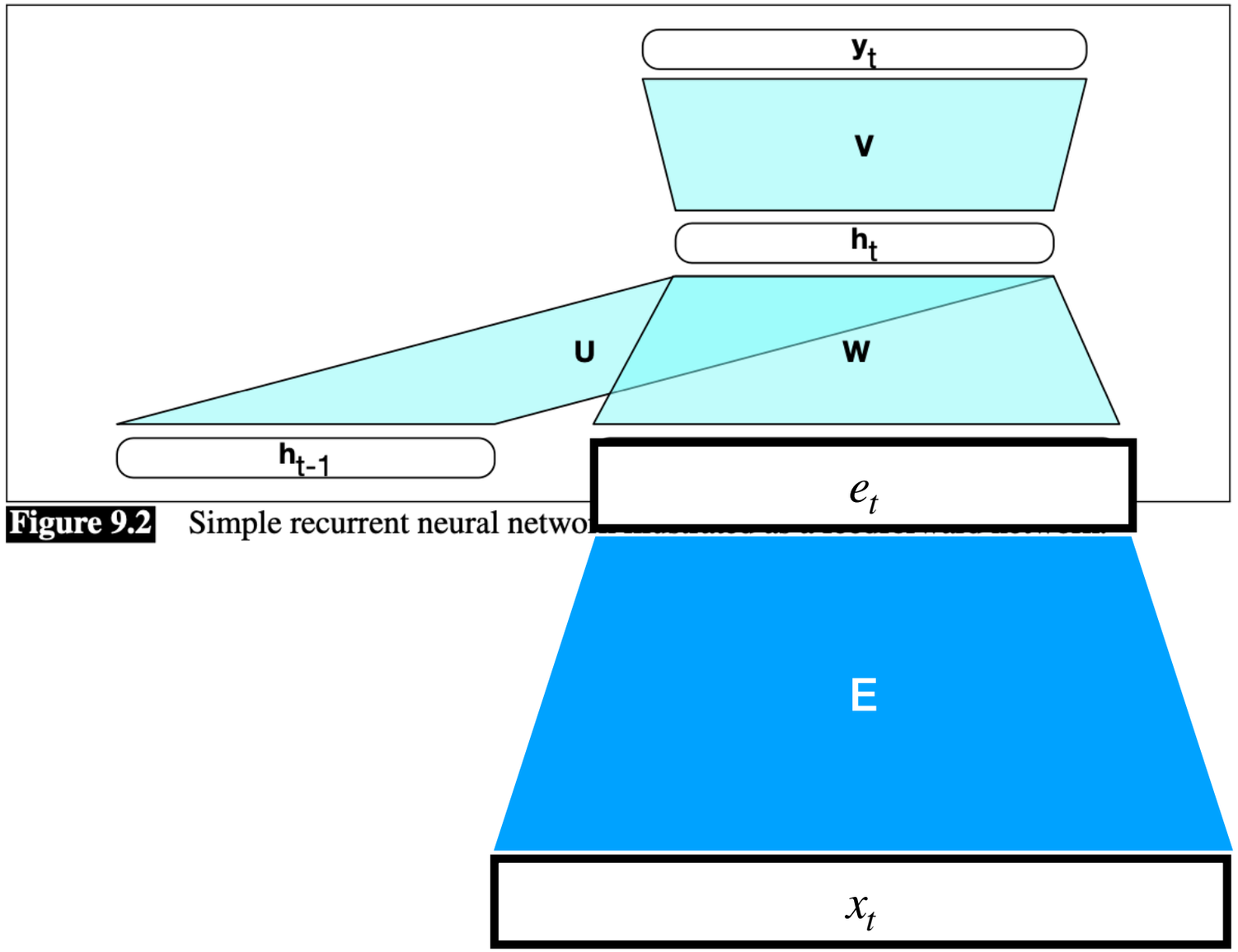


Figure 9.2 Simple recurrent neural netwo

RNNs for predicting text

Example of a simple RNN for text:

$$e_t = Ex_t$$

$$h_t = \text{softmax}(Uh_{t-1} + We_t)$$

$$\hat{p}_t = \text{softmax}(Vh_t)$$

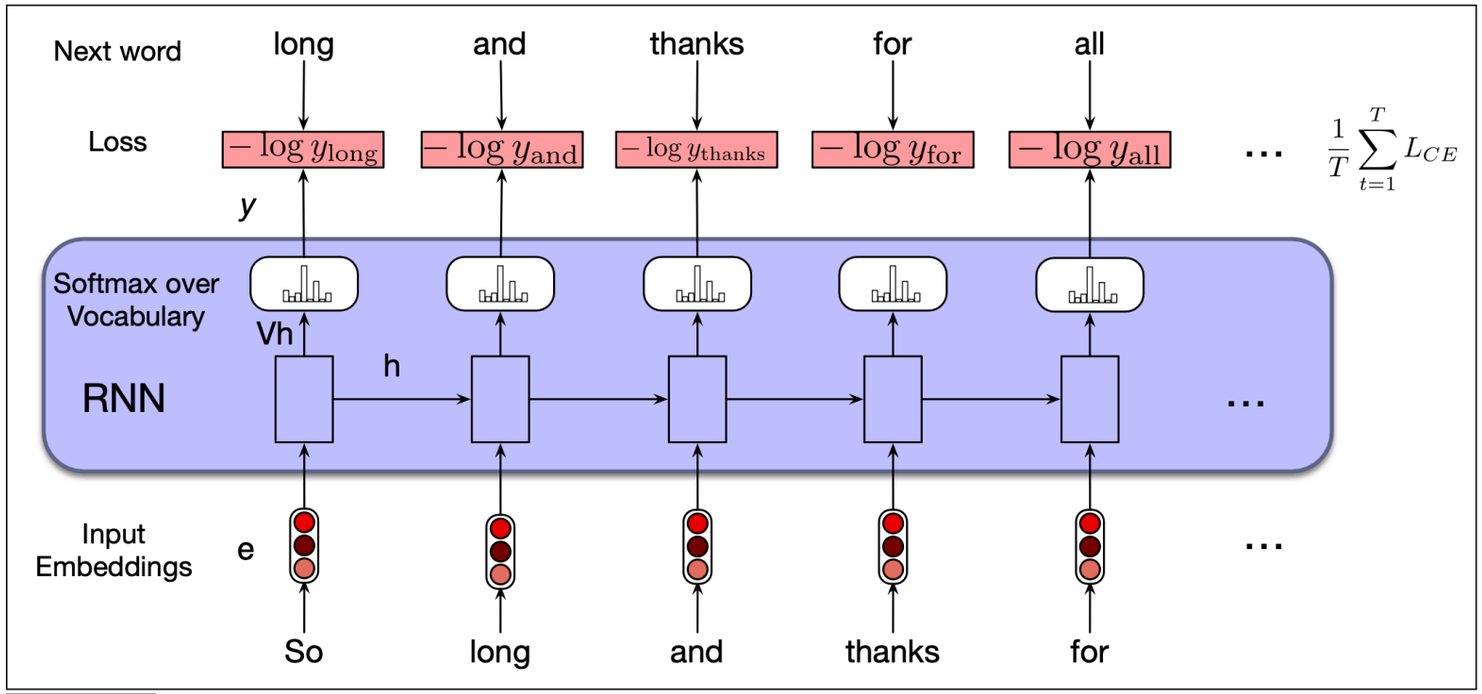


Figure 9.6 Training RNNs as language models.

Word2vec

Efficient Estimation of Word Representations in Vector Space

Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com

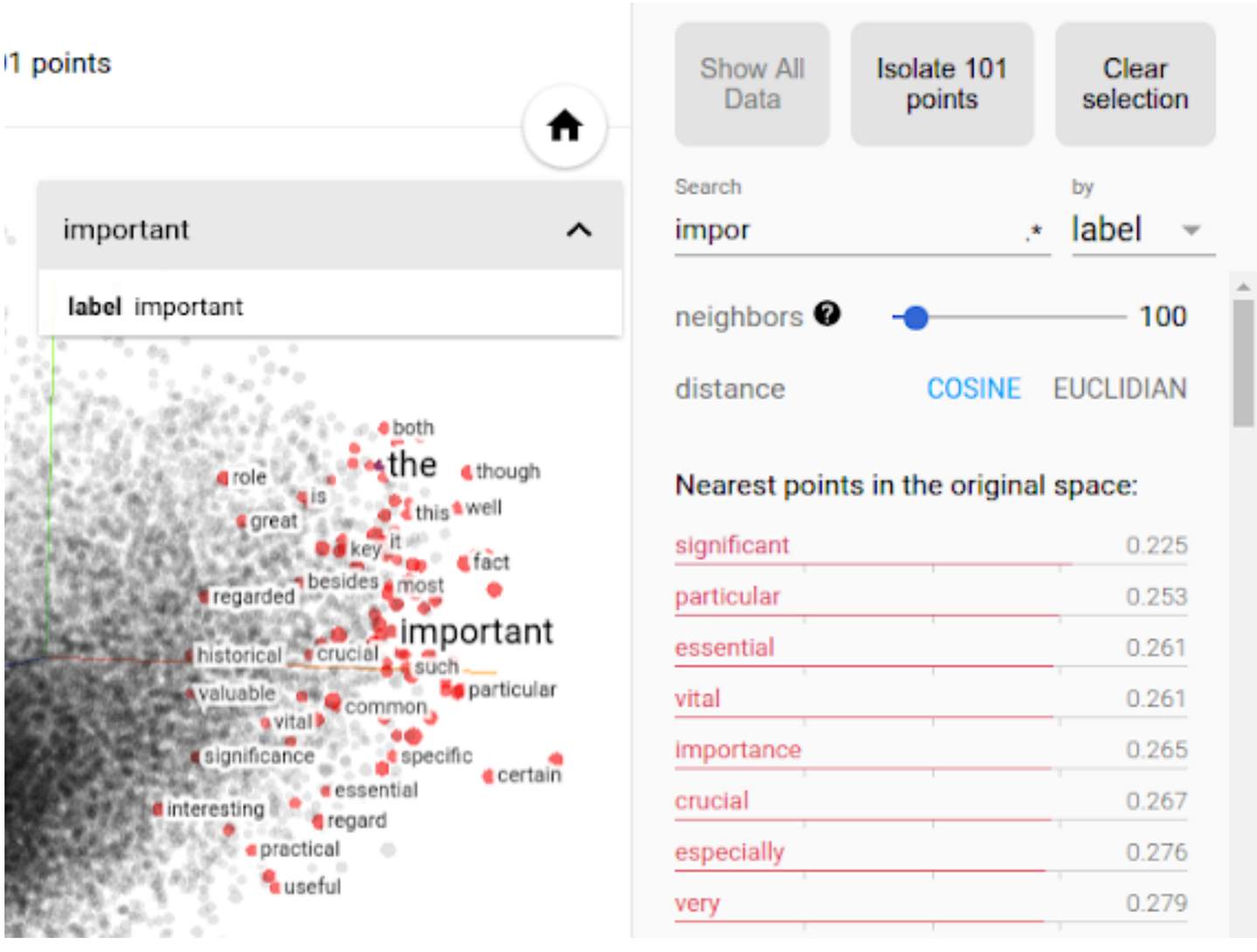
Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com

Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com

Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com

Abstract

We propose two novel model architectures for computing continuous vector representations of words from very large data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks. We observe large improvements in accuracy at much lower computational cost, i.e. it takes less than a day to learn high quality word vectors from a 1.6 billion words data set. Furthermore, we show that these vectors provide state-of-the-art performance on our test set for measuring syntactic and semantic word similarities.



Training data: internal Google dataset with 1+ billion words.

Vocabulary: 692K words, after discarding all words that occurred less than 5 times in the training data

https://projector.tensorflow.org/

Word2vec

Linguistic Regularities in Continuous Space Word Representations

Tomas Mikolov*, Wen-tau Yih, Geoffrey Zweig
Microsoft Research
Redmond, WA 98052

Abstract

Continuous space language models have recently demonstrated outstanding results across a variety of tasks. In this paper, we examine the vector-space word representations that are implicitly learned by the input-layer weights. We find that these representations are surprisingly good at capturing syntactic and semantic regularities in language, and that each relationship is characterized by a relation-specific vector offset. This allows vector-oriented reasoning based on the offsets between words. For example, the male/female relationship is automatically learned, and with the induced vector representations, "King - Man + Woman" results in a vector very close to "Queen." We demonstrate that the word vectors capture syntactic regularities by means of syntactic analogy questions (provided with this paper), and are able to correctly answer almost 40% of the questions. We demonstrate that the word vectors capture semantic regularities by using the vector offset method to answer SemEval-2012 Task 2 questions. Remarkably, this method outperforms the best previous systems.

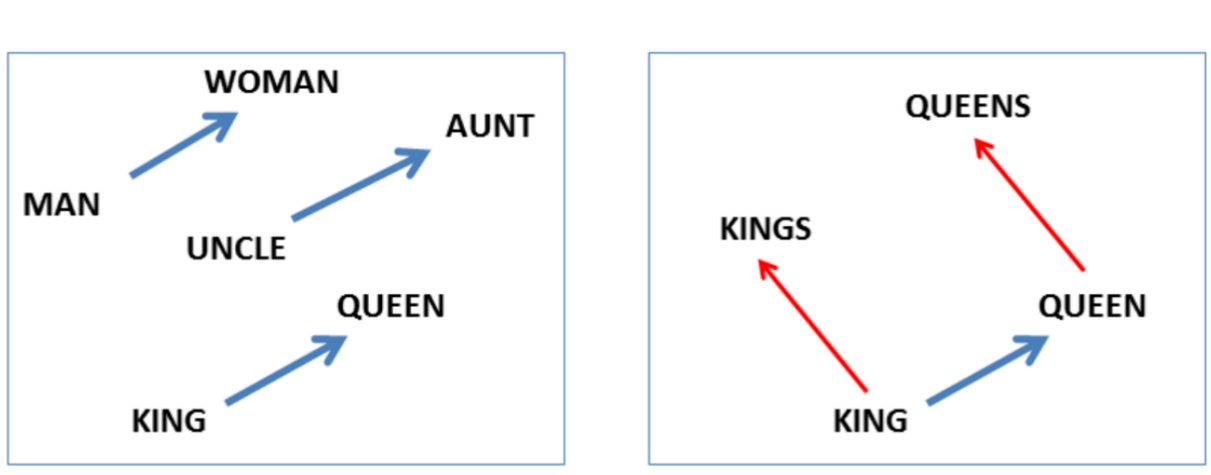


Figure 2: Left panel shows vector offsets for three word pairs illustrating the gender relation. Right panel shows a different projection, and the singular/plural relation for two words. In high-dimensional space, multiple relations can be embedded for a single word.

Word2vec

- Now that we have embeddings for words, we can use that for lots of other tasks!
  - **Text classification**: Encode a sentence/phrase by the average word embedding
  - **Initialize embeddings for other models**
  - **Knowledge discovery:** By analyzing embeddings, you can discover new knowldge, e.g. **relationships between genes and diseases**

Your turn to make a NN!

- **Q:** We've talked about how to create an RNN to predict the next word. How would you modify the RNN from before to predict the next two words?

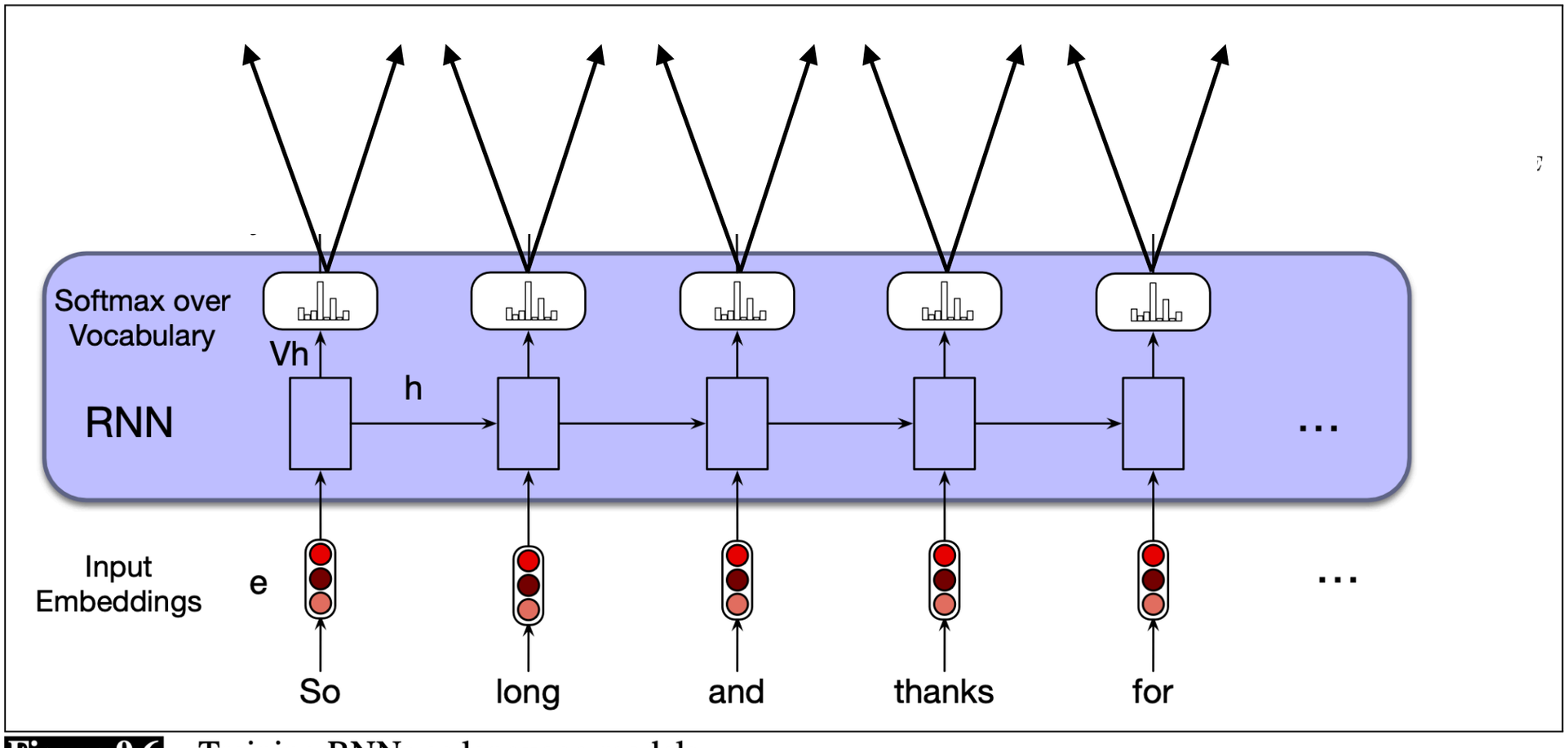


Figure 9.6 Training RNNs as language models.

Your turn to make a NN!

- **Q:** How would you construct an RNN for classifying sentences, e.g. outputs a single probability for a single sentence? How would you modify the RNN from before?

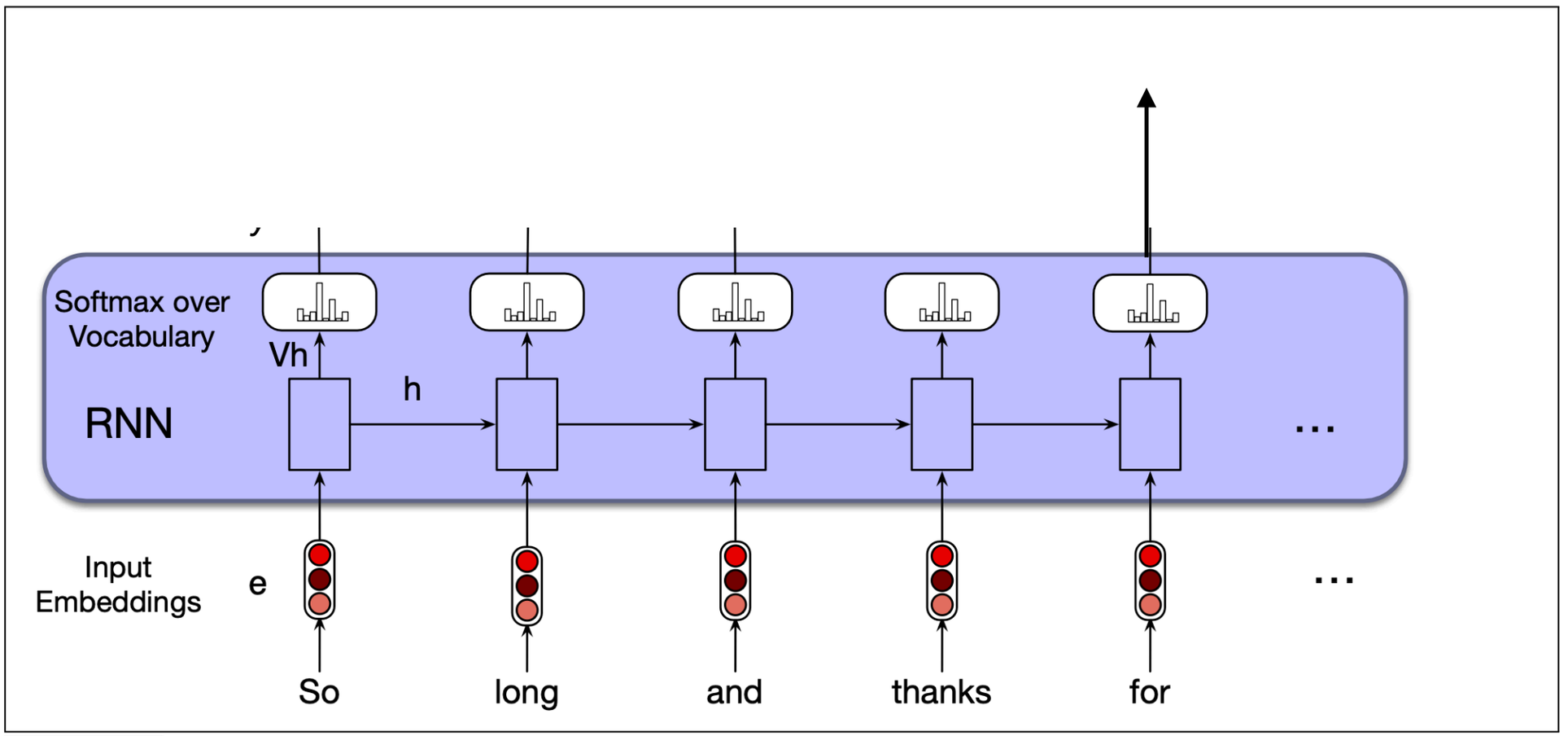


Figure 9.6 Training RNNs as language models.

Your turn to make a NN!

- **Q:** How would you construct an RNN for classifying sentences, e.g. outputs a single probability for a single sentence? How would you modify the RNN from before?

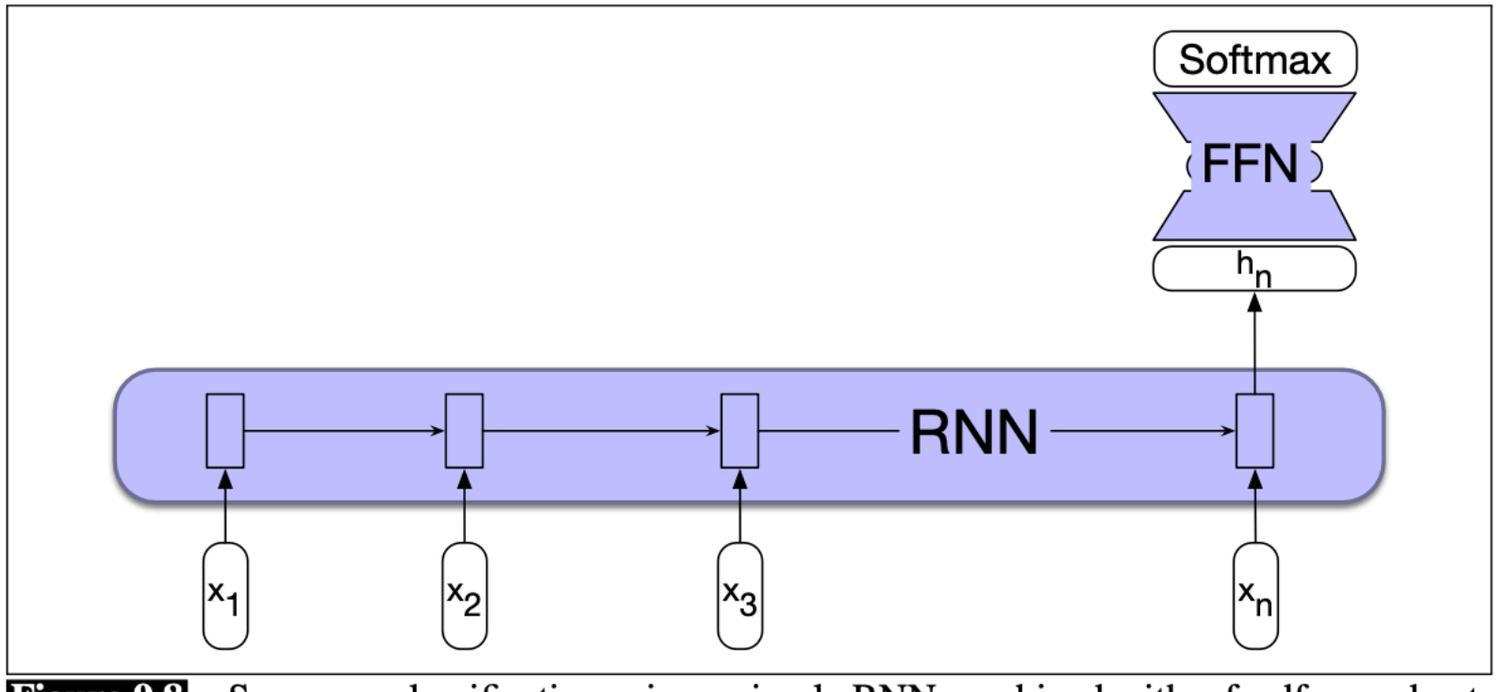


Figure 9.8 Sequence classification using a simple RNN combined with a feedforward network. The final hidden state from the RNN is used as the input to a feedforward network that performs the classification.

Your turn to make a NN!

- **Q:** What if we want to translate a sentence from English to a different language? Brainstorm an RNN architecture for conducting this translation task.

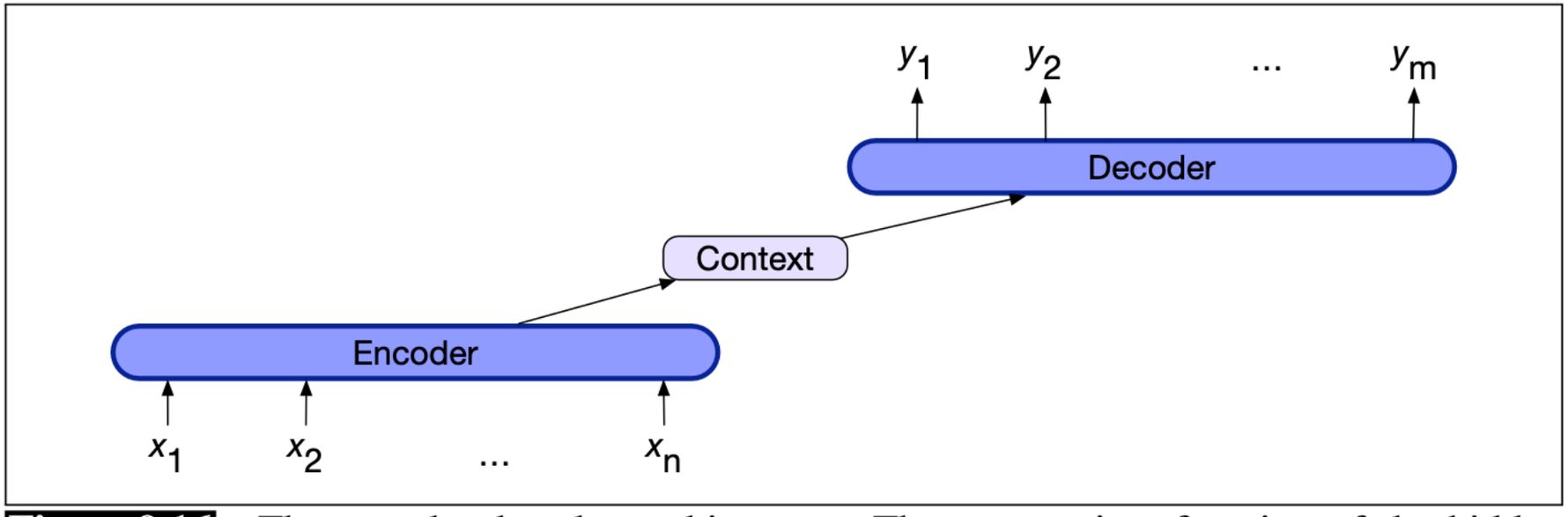


Figure 9.16 The encoder-decoder architecture. The context is a function of the hidden representations of the input, and may be used by the decoder in a variety of ways.

Outline

- Recurrent Neural Networks
  - **LSTMs**
- Transformers

Problems with RNNs

- Information captured in hidden nodes for RNNs tend to be more “local.” It is hard for RNNs to remember the distant past.

  - **Problem 1:** Hidden nodes are being used for two tasks: predicting the next output as well as carrying forward information to the next time step.

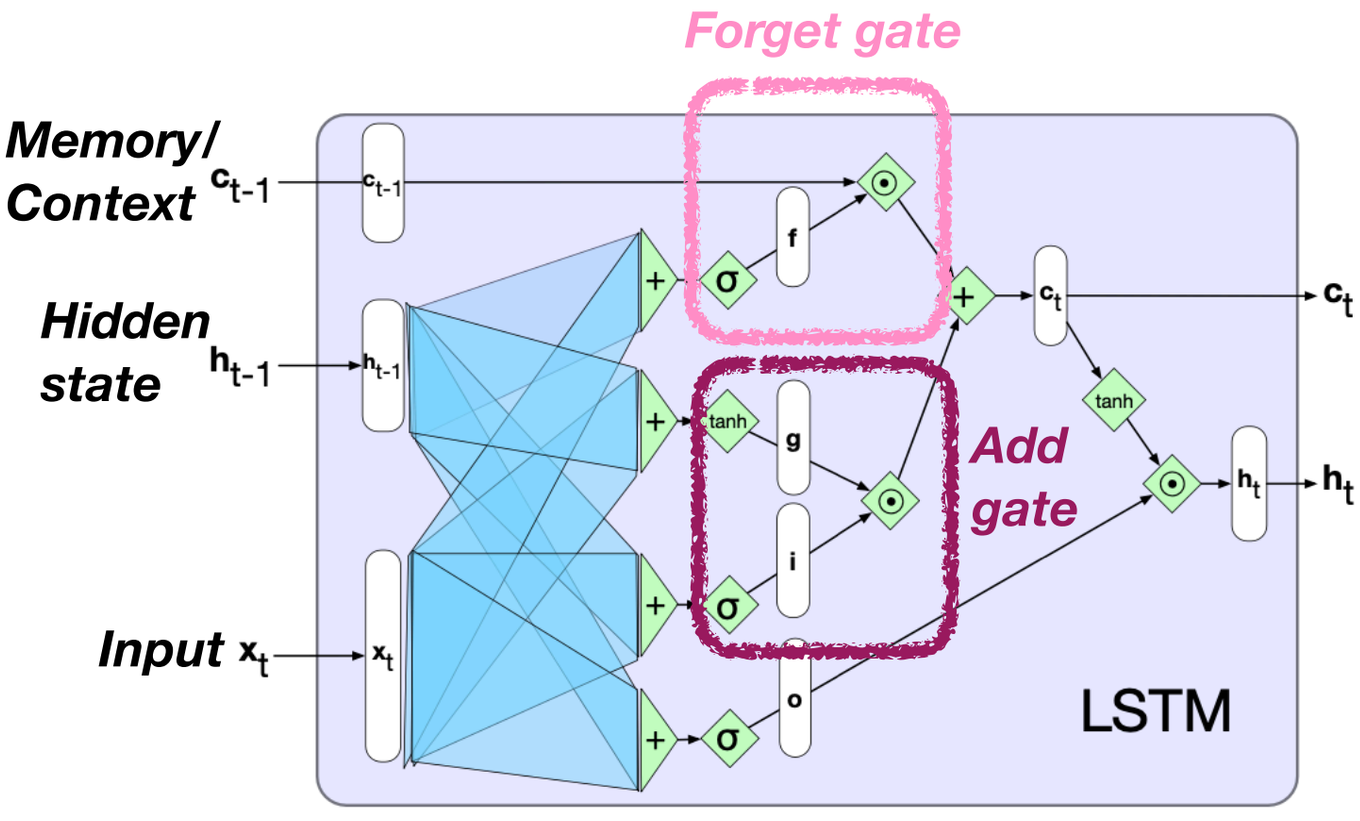
    => Decouple this

  - **Problem 2:** RNNs have to backpropagate through time, resulting in **vanishing/exploding gradients.**

    => Introduce modules for long-term memory

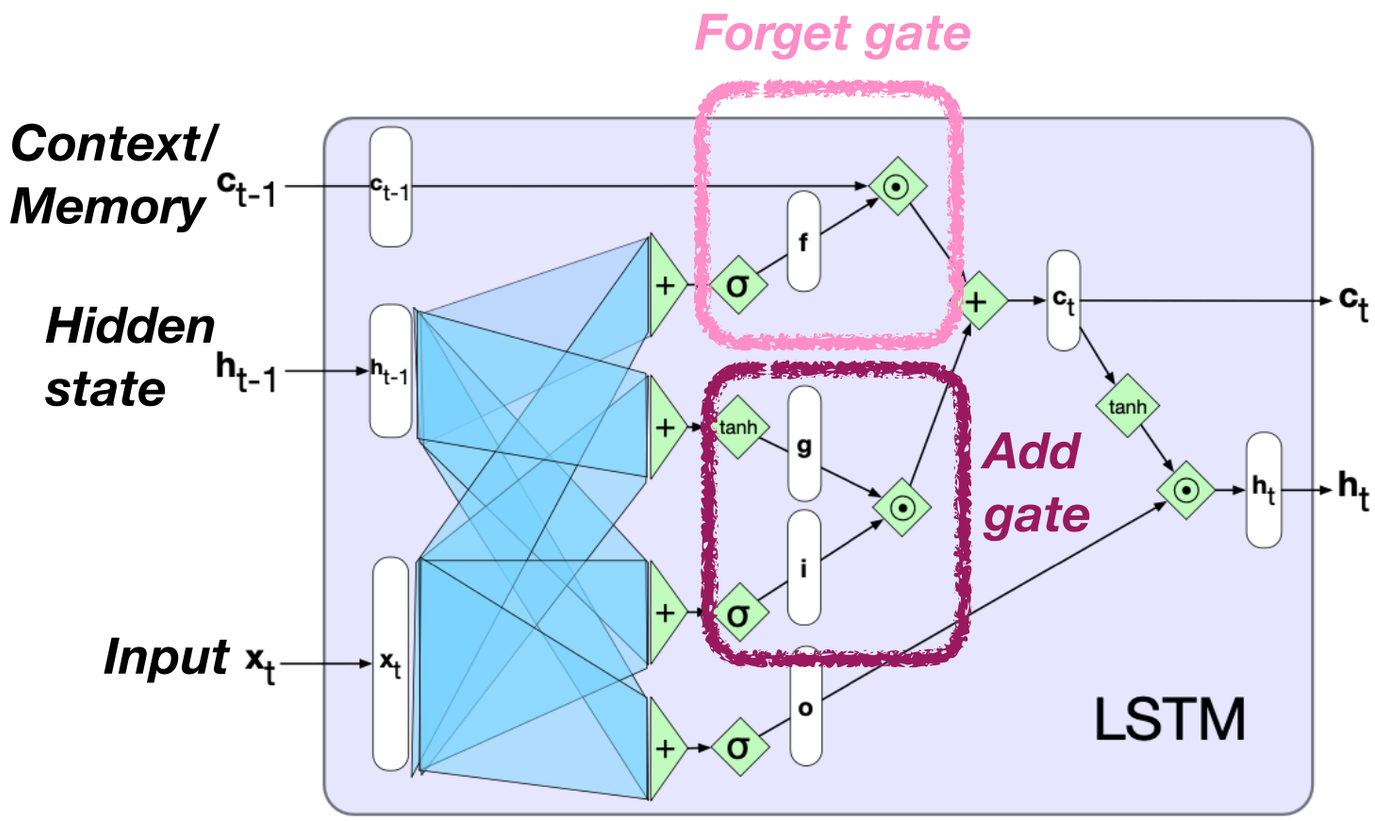
Long-short term memory network (LSTMs)

- LSTMs improve on RNNs by introducing:
  - *Components for storing memory*:
    - **Context or memory cell:** This stores the long-term memory
    - **Hidden state:** This is the short-term memory
  - *Modules for modifying memory:*
    - **Forget gates** remove parts of the “memory/ context” that aren’t needed anymore
    - **Add gates** add information that will likely be useful later on



Long-short term memory network (LSTMs)

- **Forget gate**:
  - Compute forget vector $f_t = \sigma(U_f h_{t-1} + W_f x_t)$.
  - Given memory cell $c_{t-1} \in \mathbb{R}^m$, apply element-wise product $f_t \odot c_{t-1}$.

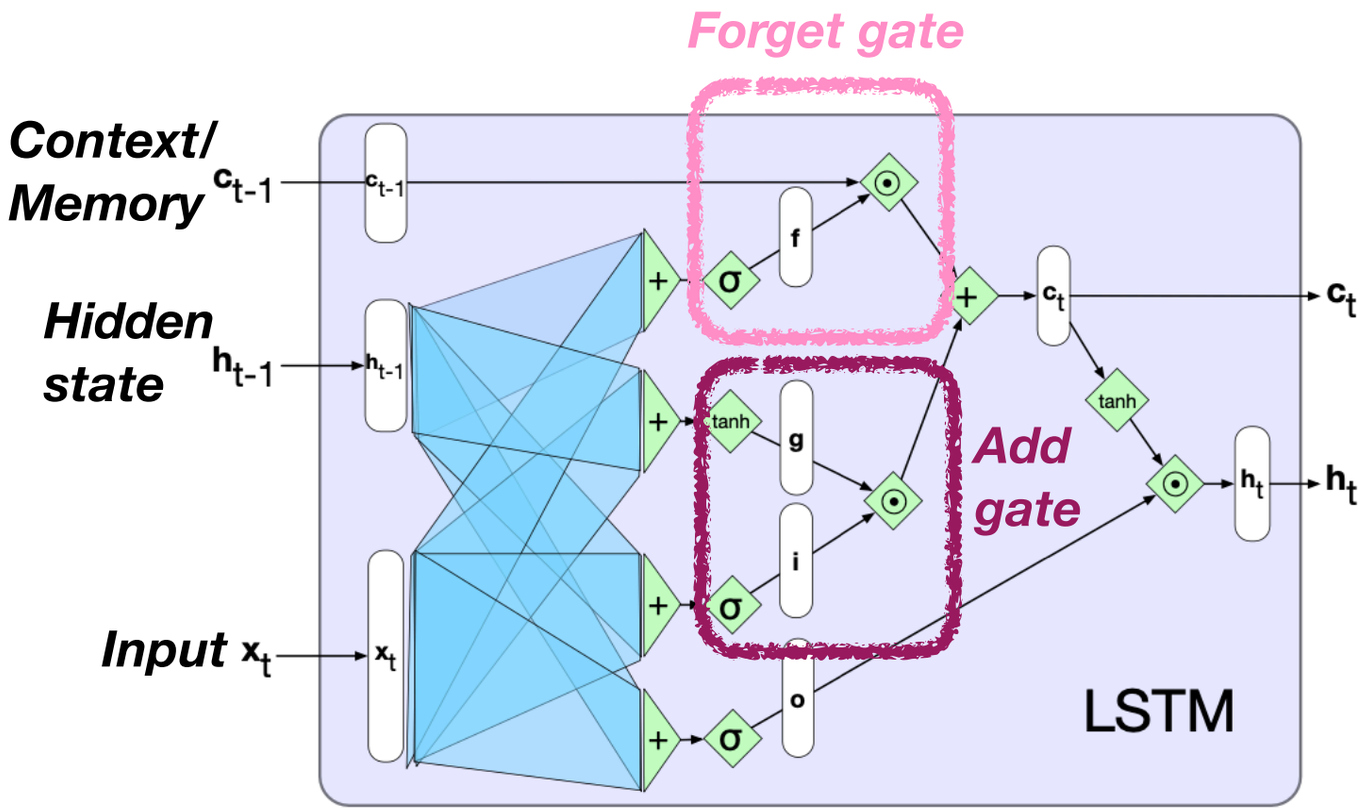


f_t	1	0	1	0	1	1	1
$\times\ c_{t-1}$	0.5	0.9	-0.3	1.3	-4.1	0.5	0.2
	0.5	0	-0.3	0	-4.1	0.5	0.2

Forgetting these elements

Long-short term memory network (LSTMs)

- **Add gate:**
  - Combine new input with previous hidden state:
    $g_t = \tanh(U_g h_{t-1} + W_g x_t)$
  - Compute a "add" probability vector $i_t \in \mathbb{R}^m$.
  - Apply element-wise product $i_t \odot g_t$:

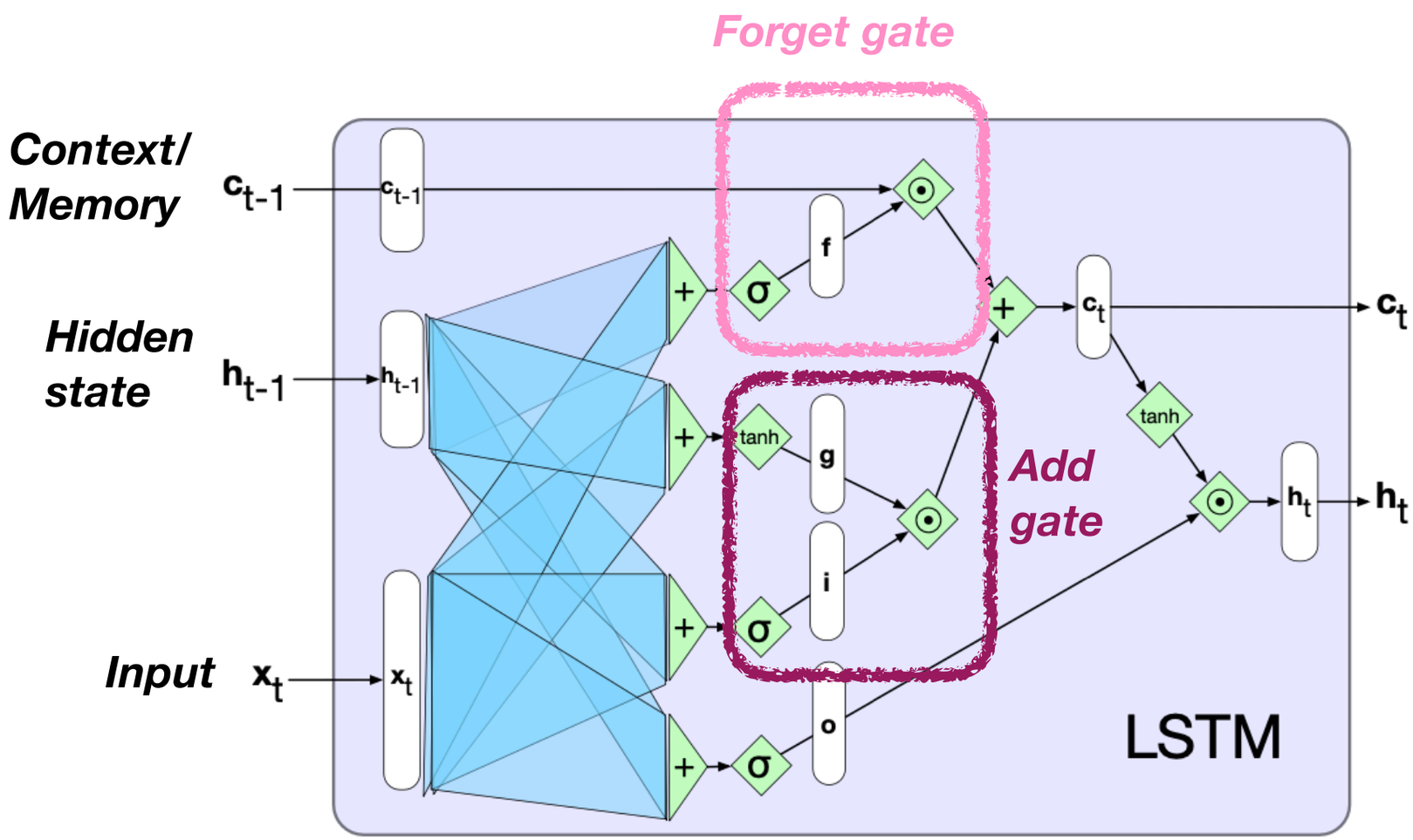


	i_t	1	0	1	0	1	1	1
×	g_t	0.5	0.9	-0.3	1.3	-4.1	0.5	0.2
		0.5	0	-0.3	0	-4.1	0.5	0.2

Adding these elements

Long-short term memory network (LSTMs)

- **Putting it together:** The updated memory cell is defined as the sum of the forget gate and the add gate: $c_t = f_t \odot c_{t-1} + i_t \odot g_t$.



Long-short term memory network (LSTMs) in Encoder-Decoder Models

Europarl: A Parallel Corpus for Statistical Machine Translation

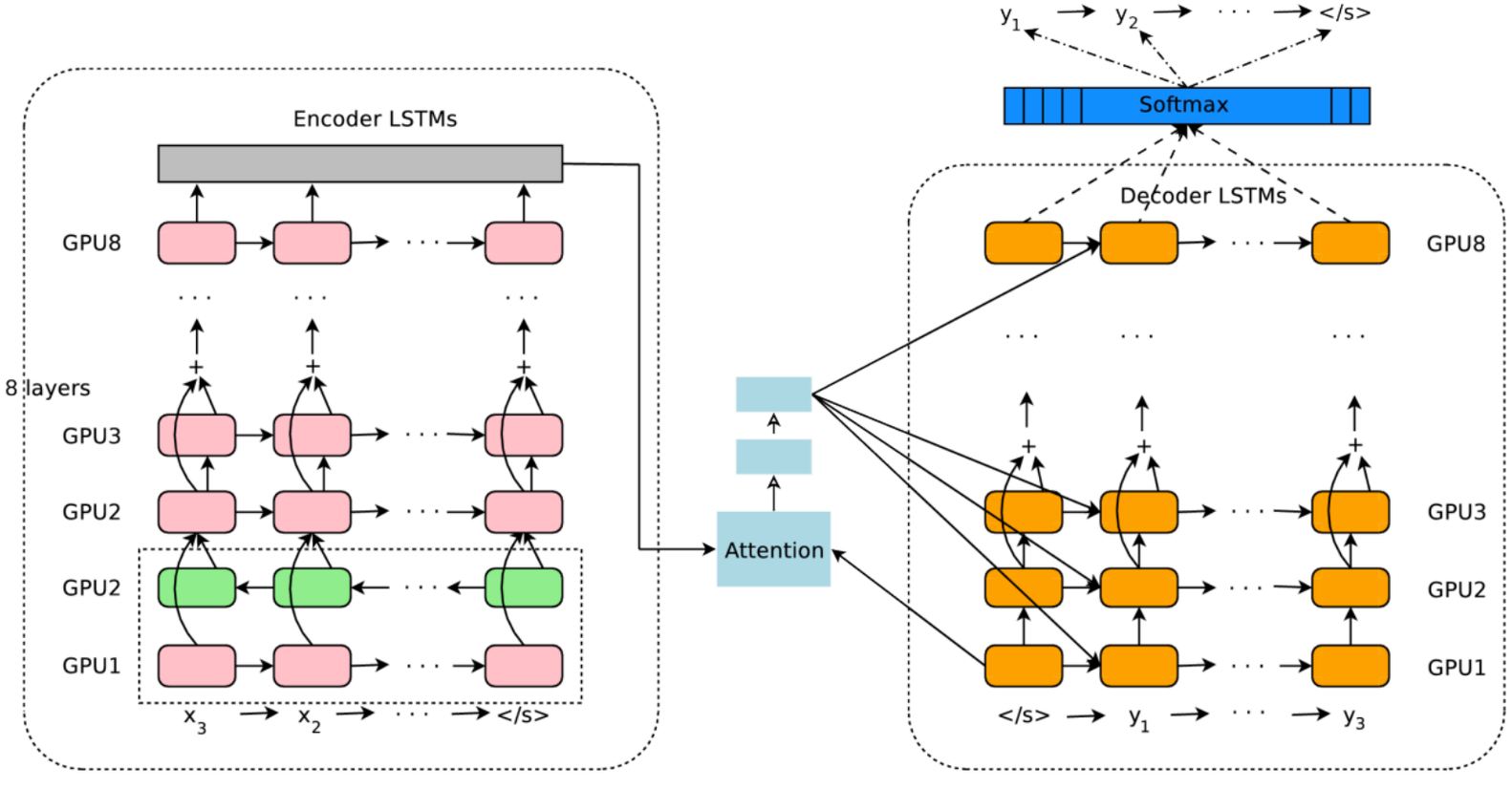
Philipp Koehn
School of Informatics
University of Edinburgh, Scotland
pkoehn@inf.ed.ac.uk

Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi
yonghui,schuster,zhifengc,qvl,mnorouzi@google.com

Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean

Danish: det er næsten en personlig rekord for mig dette efterår .
German: das ist für mich fast persönlicher rekord in diesem herbst .
Greek: πρόκειται για το προσωπικό μου ρεκόρ αυτό το φθινόπωρο .
English that is almost a personal record for me this autumn !
Spanish: es la mejor marca que he alcanzado este otoño .
Finnish: se on melkein minun ennätykseni tänä syksynä !
French: c ' est pratiquement un record personnel pour moi , cet automne !
Italian: e ' quasi il mio record personale dell ' autunno .
Dutch: dit is haast een persoonlijk record deze herfst .
Portuguese: é quase o meu recorde pessoal deste semestre !
Swedish: det är nästan personligt rekord för mig denna höst !



Multi-layer bidirectional LSTM with residual connections!

Wu, et al. *arXiv 2016*

Learning to Diagnose with LSTM Recurrent Neural Networks

Zachary C. Lipton *†
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093, USA
zlipton@cs.ucsd.edu

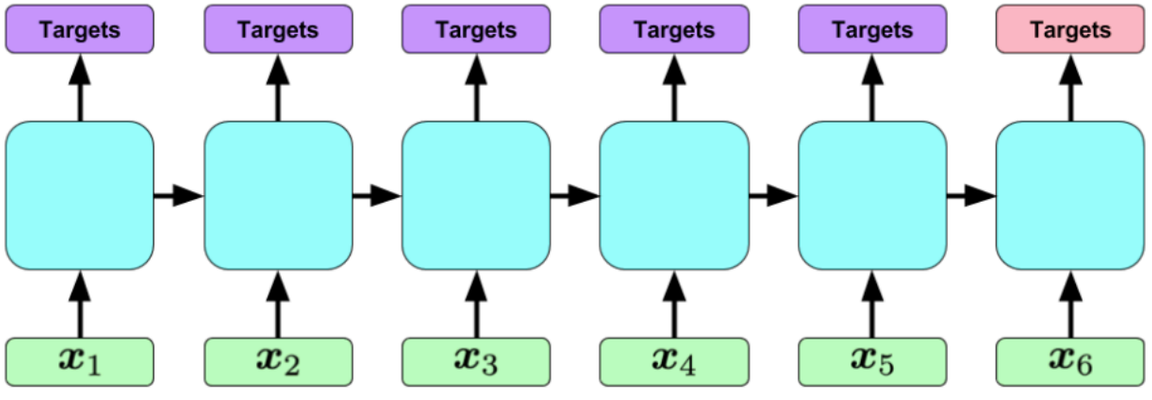
David C. Kale *‡
Department of Computer Science
University of Southern California
Los Angeles, CA 90089
dkale@usc.edu

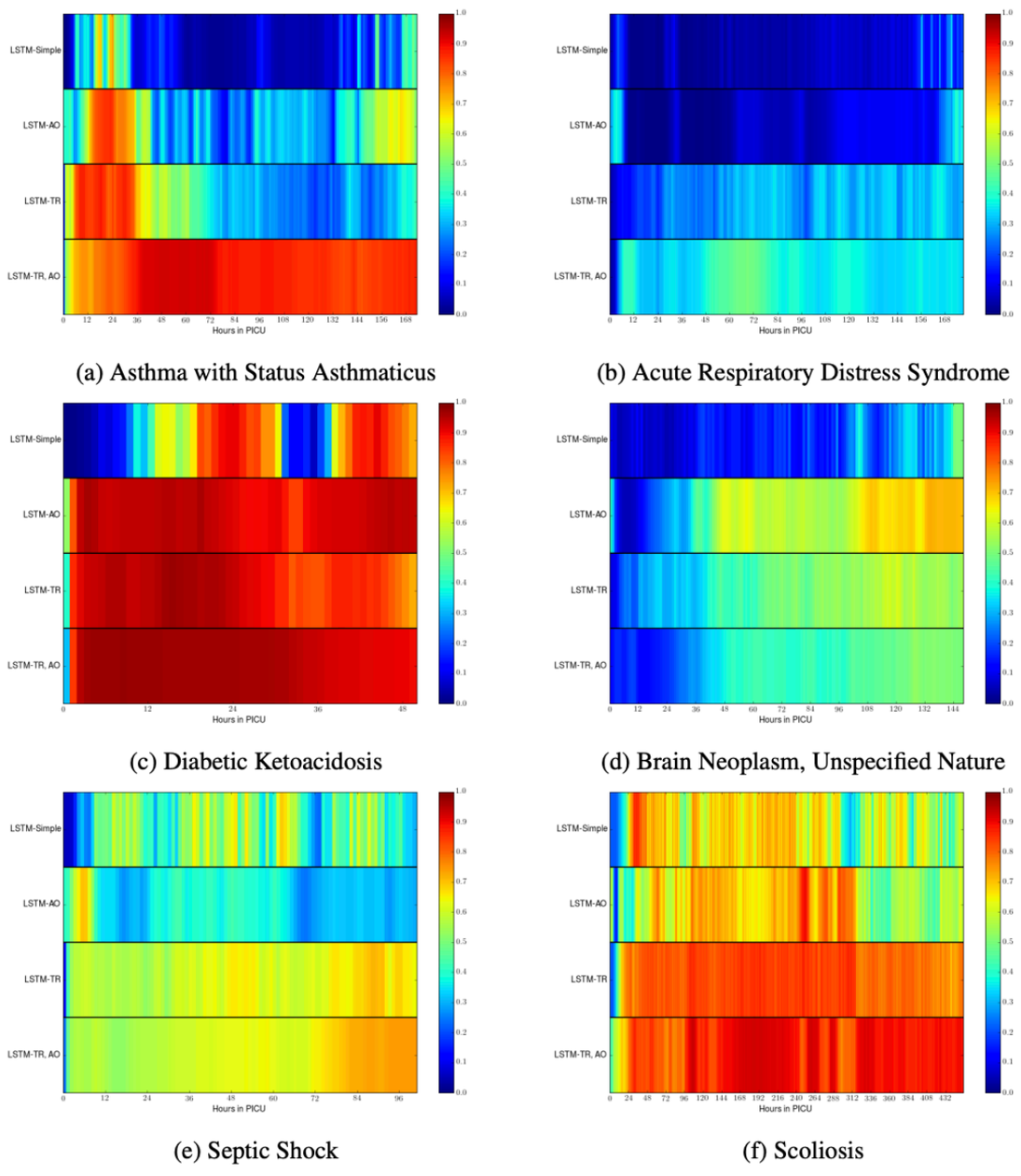
Charles Elkan
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093, USA
elkan@cs.ucsd.edu

Randall Wetzel
Laura P. and Leland K. Whittier Virtual PICU
Children's Hospital Los Angeles
Los Angeles, CA 90027
rwetzel@chla.usc.edu

Abstract

Clinical medical data, especially in the intensive care unit (ICU), consist of multivariate time series of observations. For each patient visit (or *episode*), sensor data and lab test results are recorded in the patient's Electronic Health Record (EHR). While potentially containing a wealth of insights, the data is difficult to mine effectively, owing to varying length, irregular sampling and missing data. Recurrent Neural Networks (RNNs), particularly those using Long Short-Term Memory (LSTM) hidden units, are powerful and increasingly popular models for learning from sequence data. They effectively model varying length sequences and capture long range dependencies. We present the first study to empirically evaluate the ability of LSTMs to recognize patterns in multivariate time series of clinical measurements. Specifically, we consider multilabel classification of diagnoses, training a model to classify 128 diagnoses given 13 frequently but irregularly sampled clinical measurements. First, we establish the effectiveness of a simple LSTM network for modeling clinical data. Then we demonstrate a straightforward and effective training strategy in which we replicate targets at each sequence step. Trained only on raw time series, our models outperform several strong baselines, including a multilayer perceptron trained on hand-engineered features.





(a) Asthma with Status Asthmaticus

(b) Acute Respiratory Distress Syndrome

(c) Diabetic Ketoacidosis

(d) Brain Neoplasm, Unspecified Nature

(e) Septic Shock

(f) Scoliosis

Figure 5: Each chart depicts the probabilities assigned by each of four models at each (hourly resampled) time step. LSTM-Simple uses only targets at the final time step. LSTM-TR uses target replication. LSTM-AO uses auxiliary outputs (diagnoses), and LSTM-TR,AO uses both techniques. LSTMs with target replication learn to make accurate diagnoses earlier.

Outline

- Recurrent Neural Networks
  - LSTMs
- **Attention mechanism**
- Transformers

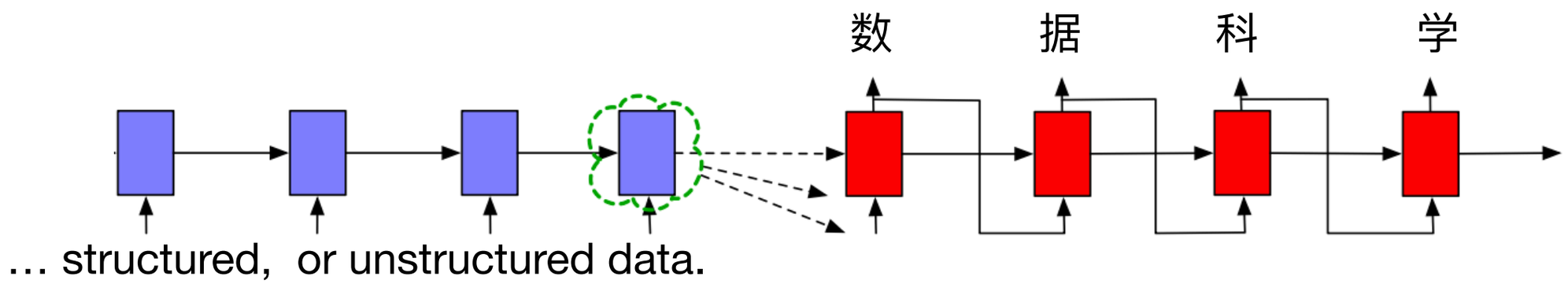
The problem of long inputs

Translation task:

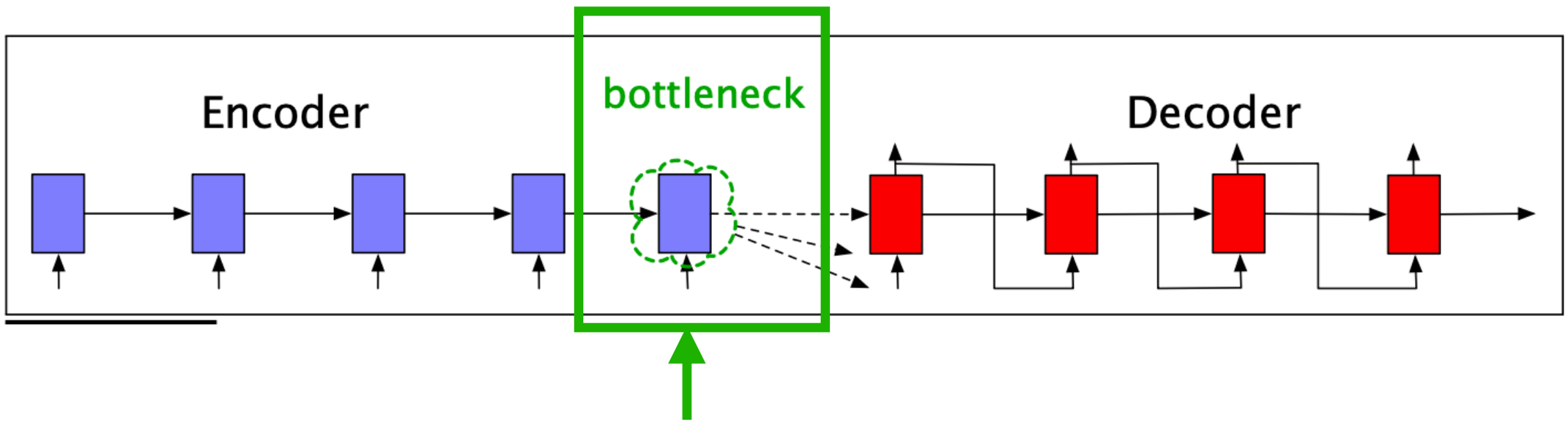
Data science is an interdisciplinary academic field that uses statistics, scientific computing, scientific methods, processes, algorithms and systems to extract or extrapolate knowledge and insights from potentially noisy, structured, or unstructured data.

数据科学是一个跨学科的学术领域，它使用统计学、科学计算、科学方法、流程、算法和系统从潜在的噪声、结构化或非结构化数据中提取或推断知识和见解。

Encoder-decoder networks



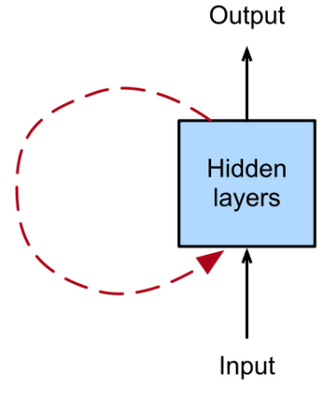
The problem with bottlenecks



A standard RNN/LSTM summarizes the entire input into one fixed-size vector, regardless of how long the input sentence is.

Compression of variable-sized inputs into a fixed-size vector is inevitably lossy, which can hurt prediction.

How can we have a recurrent structure to process variable-sized inputs but have memory that grows with the size of the input?



Fixing the problem of bottlenecks with attention

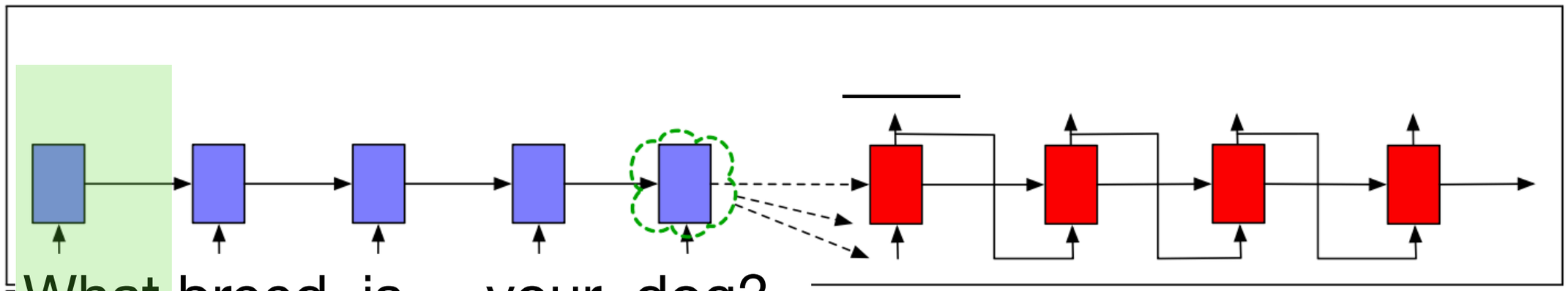
NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

Dzmitry Bahdanau
Jacobs University Bremen, Germany

KyungHyun Cho **Yoshua Bengio***
Université de Montréal

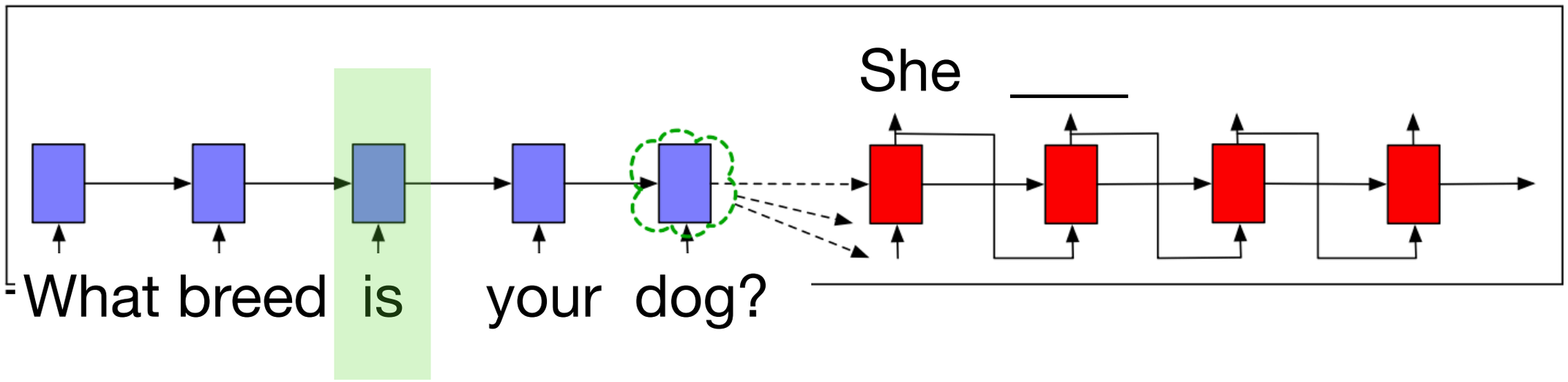
ICLR 2015

Which inputs matter?



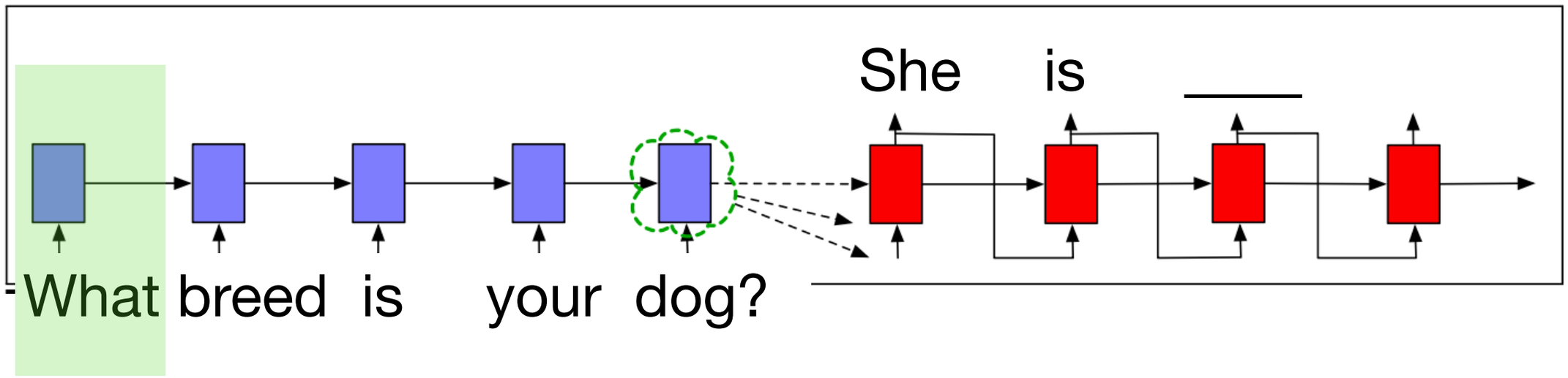
To complete this sentence, we probably want to pay **attention** to certain words in the input.

Which inputs matter?



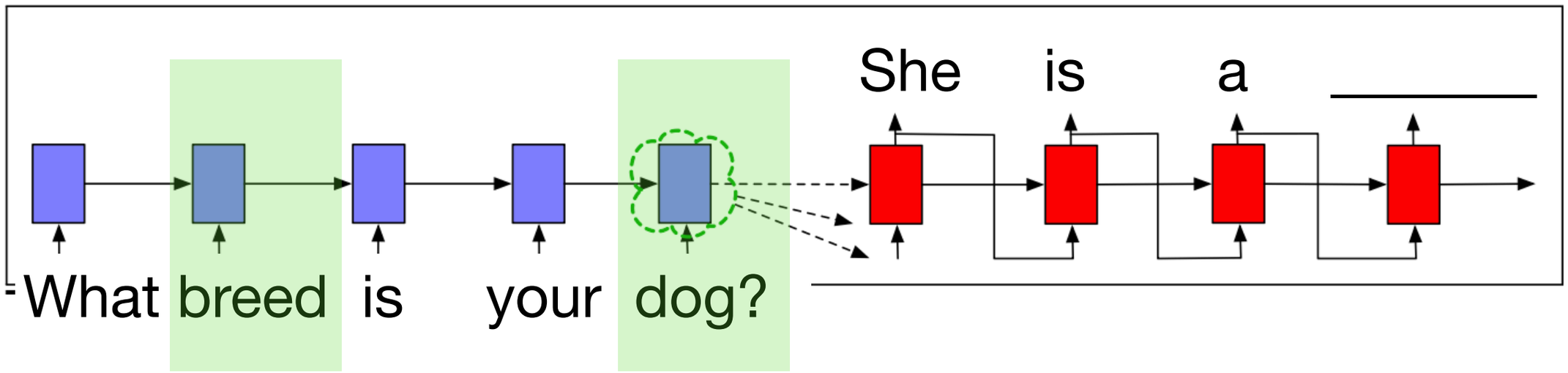
To complete this sentence, we probably want to pay **attention** to certain words in the input.

Which inputs matter?



To complete this sentence, we probably want to pay **attention** to certain words in the input.

Which inputs matter?



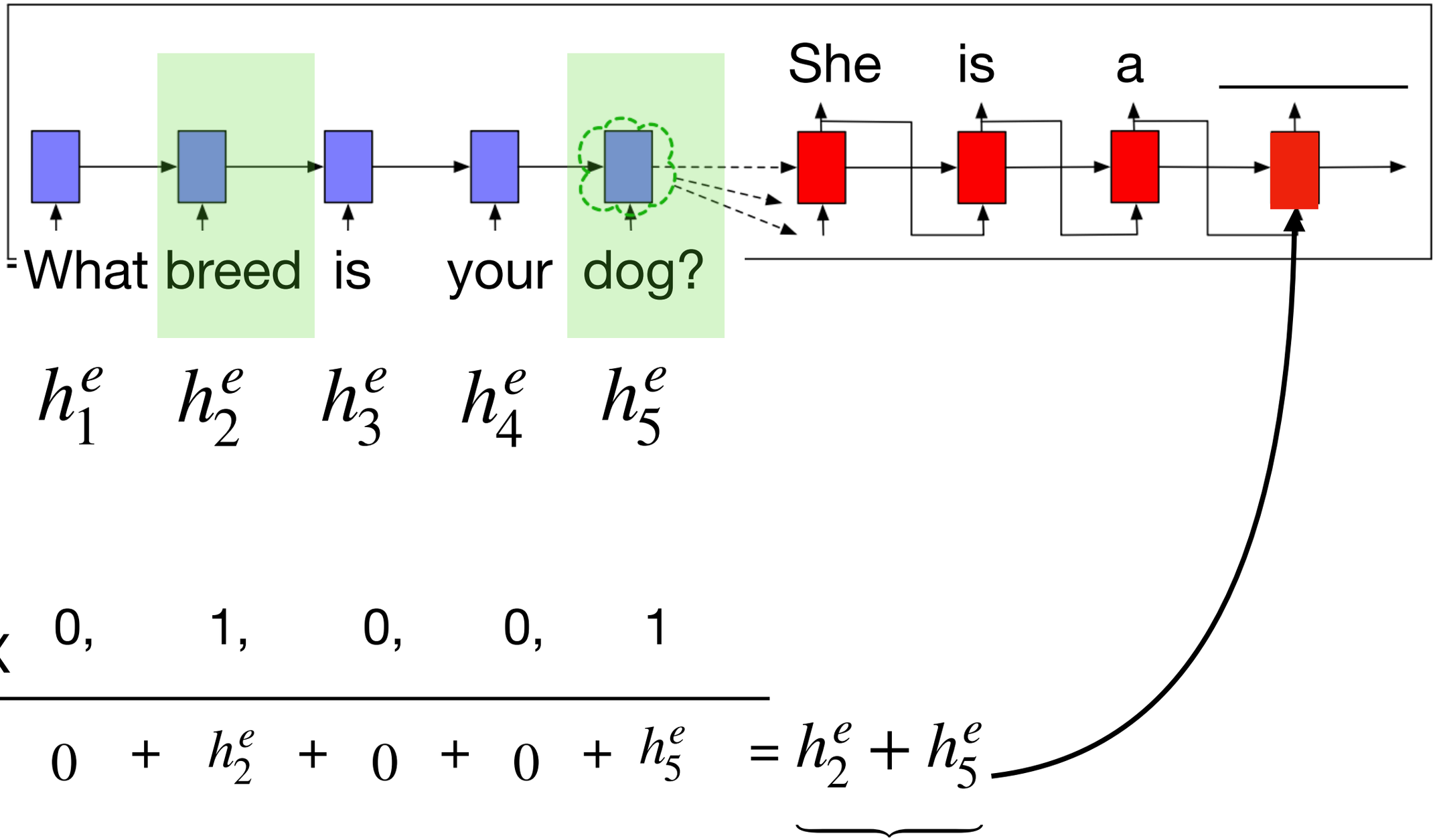
To complete this sentence, we probably want to pay **attention** to certain words in the input.

How can we mathematically define "attention"?

(1) Given a vector representing "attention," how do I summarize the input?

(2) How do I compute the attention vector?

Given an attention vector, how to summarize inputs?



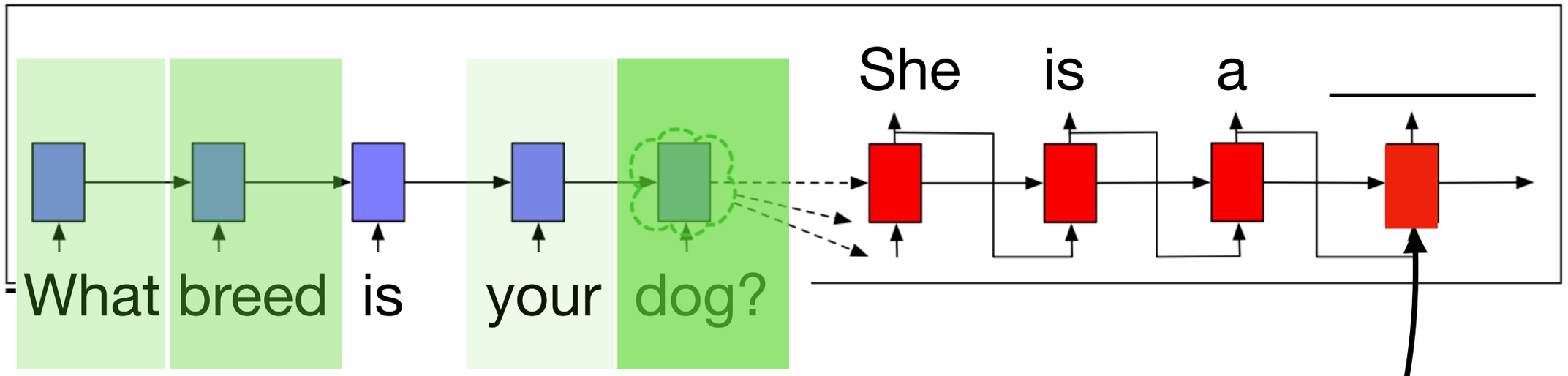
Hidden layer: $h_1^e \quad h_2^e \quad h_3^e \quad h_4^e \quad h_5^e$

Attention vector at pos 4:

$$\frac{X \quad 0, \quad 1, \quad 0, \quad 0, \quad 1}{0 \quad + \quad h_2^e \quad + \quad 0 \quad + \quad 0 \quad + \quad h_5^e} = \underbrace{h_2^e + h_5^e}$$

Does not depend on number of input tokens

Given an attention vector, how to summarize inputs?



Hidden layer: $h_1^e \quad h_2^e \quad h_3^e \quad h_4^e \quad h_5^e$

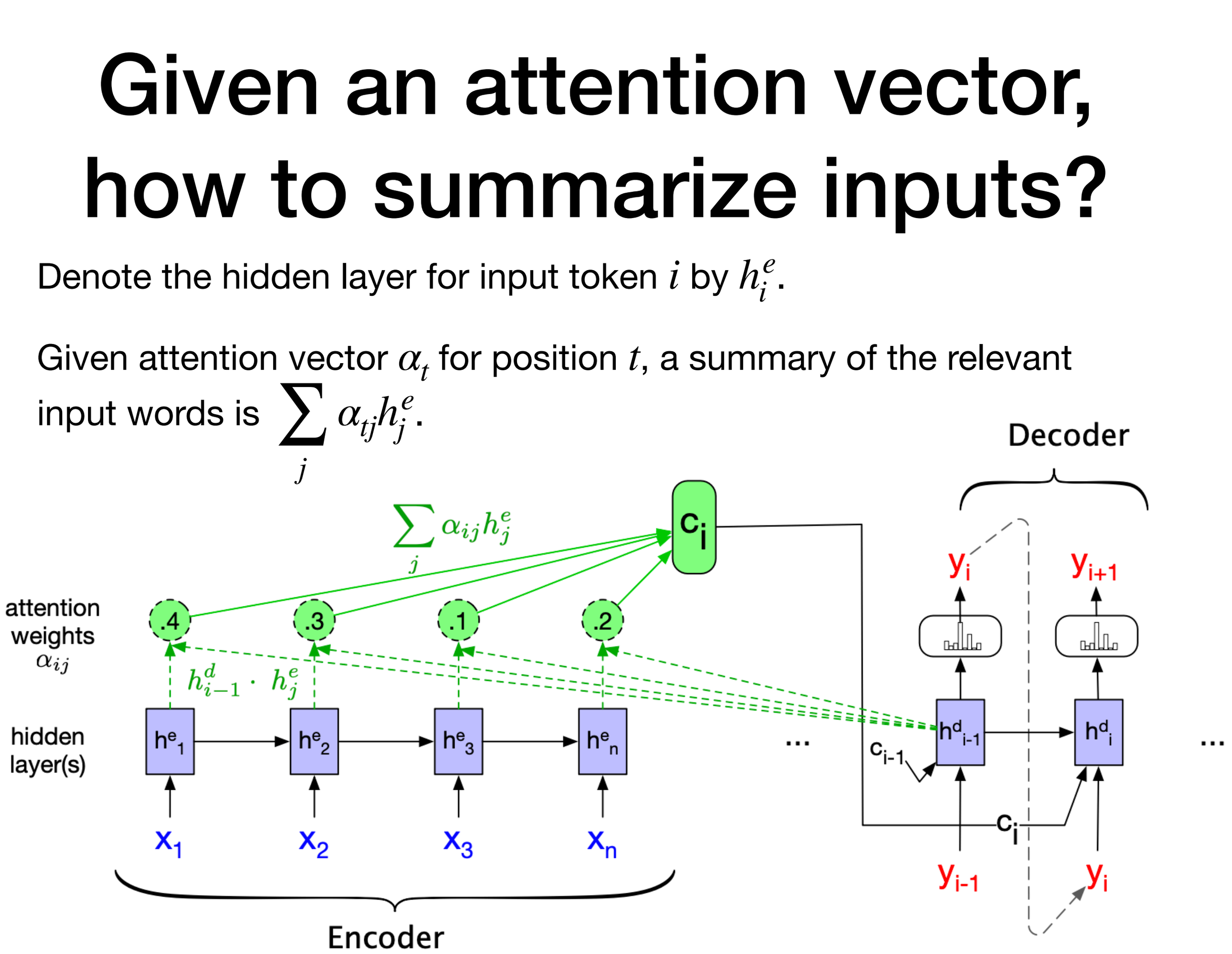
Attention vector at pos 4: X 0.2, 0.3, 0, 0.1, 0.4

$$0.2h_1^e + 0.3h_2^e + 0 + 0.1h_4^e + 0.4h_5^e$$

Given an attention vector, how to summarize inputs?

Denote the hidden layer for input token i by h_i^e.

Given attention vector α_t for position t, a summary of the relevant input words is $\sum_j \alpha_{tj} h_j^e$.



How can we mathematically define "attention"?

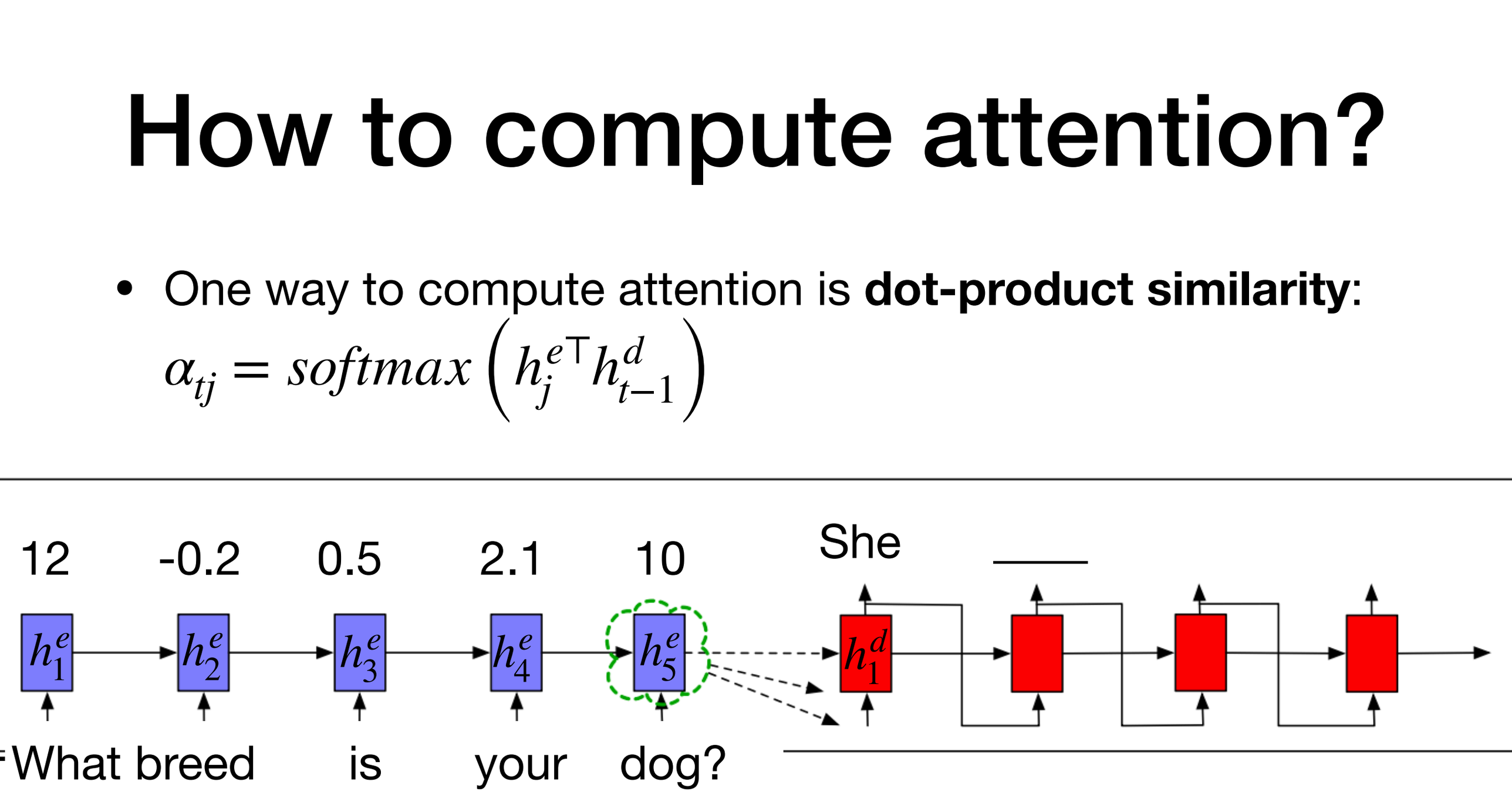
✓ (1) Given a vector representing "attention," how do I summarize the input?

(2) How do I compute the attention vector?

How to compute attention?

- One way to compute attention is **dot-product similarity**:

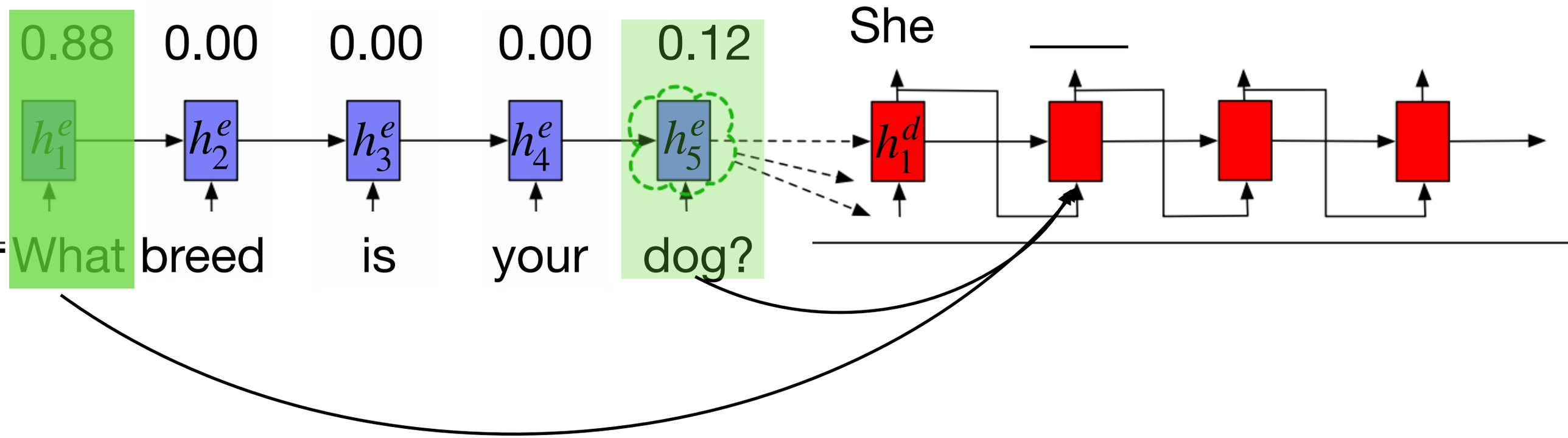
$$\alpha_{tj} = softmax\left(h_j^{e\top} h_{t-1}^{d}\right)$$



How to compute attention?

- One way to compute attention is **dot-product similarity**:

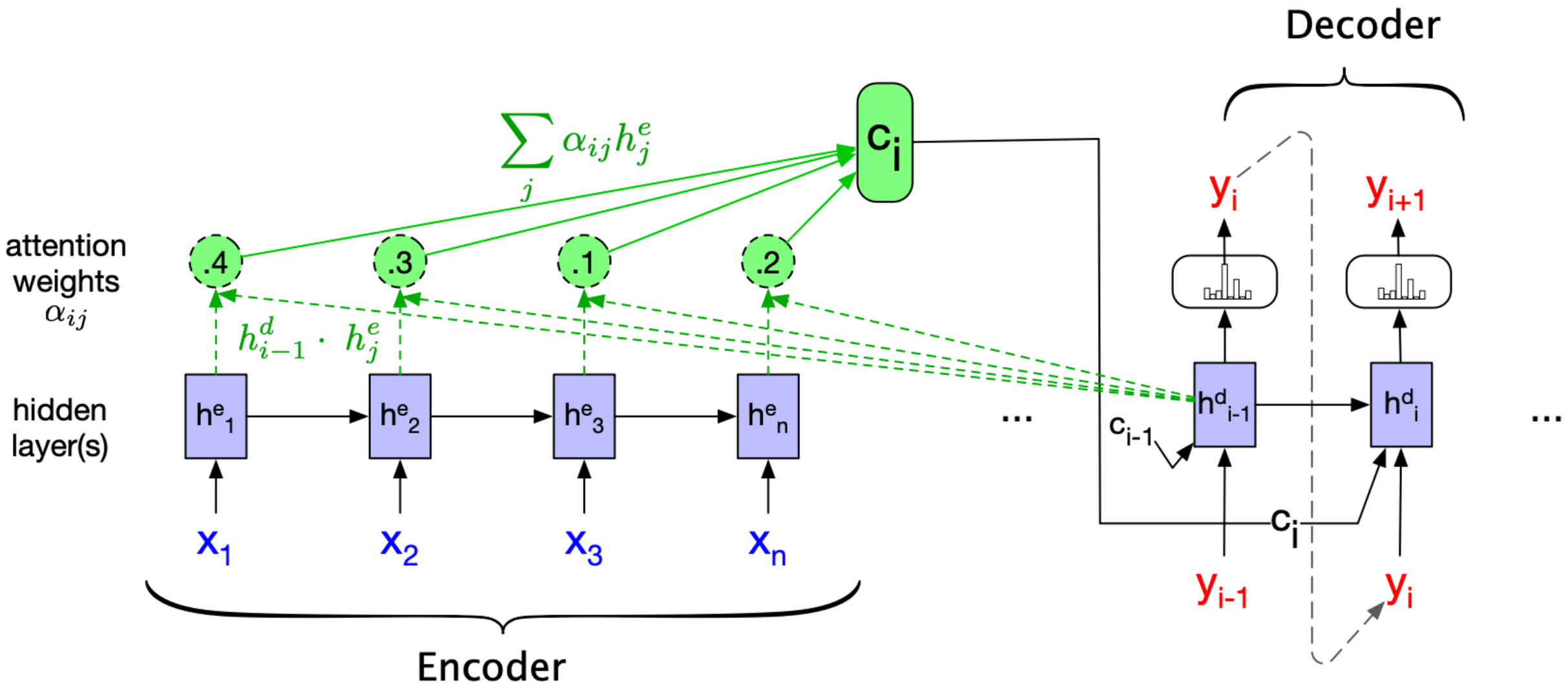
$$\alpha_{tj} = softmax\left(h_j^{e\top} h_{t-1}^{d}\right)$$



How can we mathematically define "attention"?

✓ (1) Given a vector representing "attention," how do I summarize the input?

✓ (2) How do I compute the attention vector?



Attention

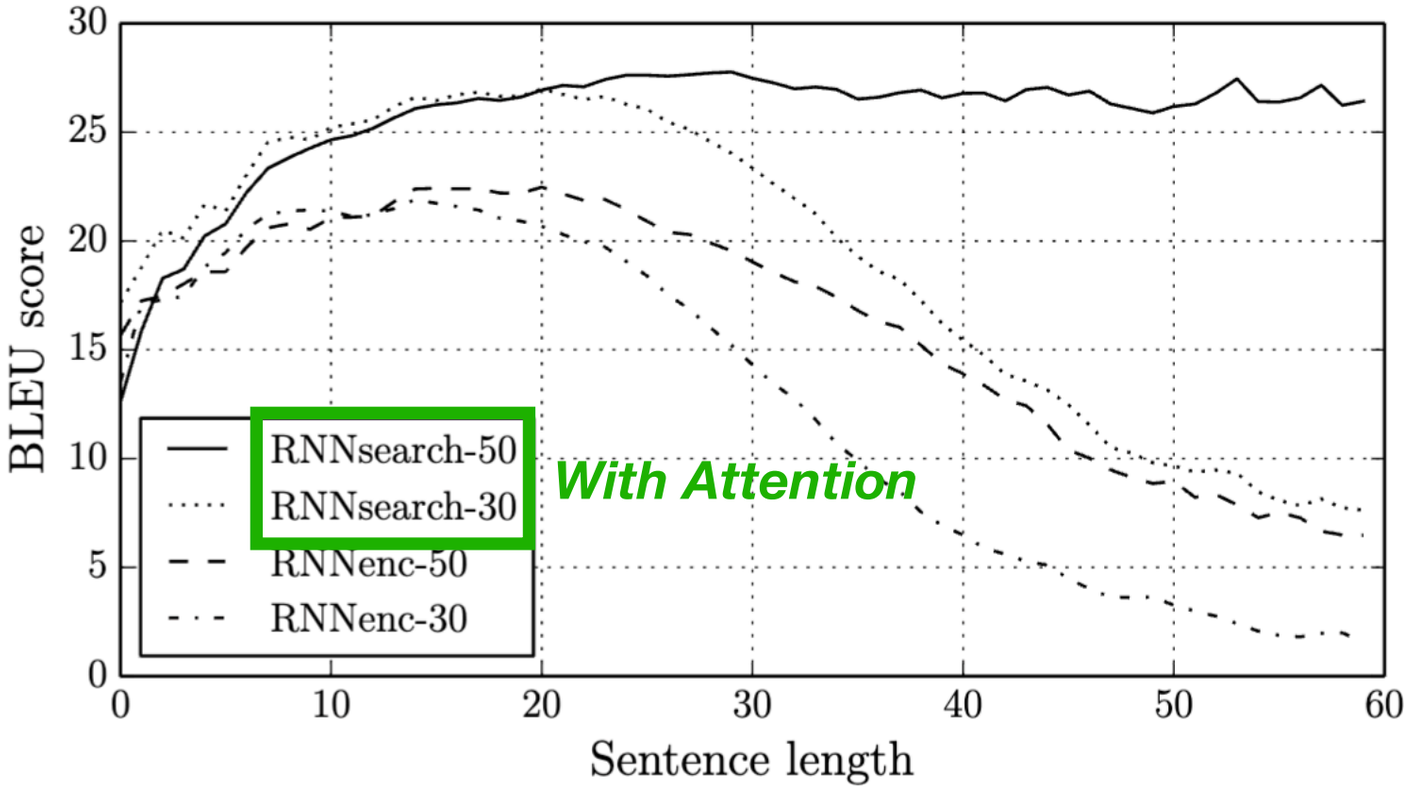
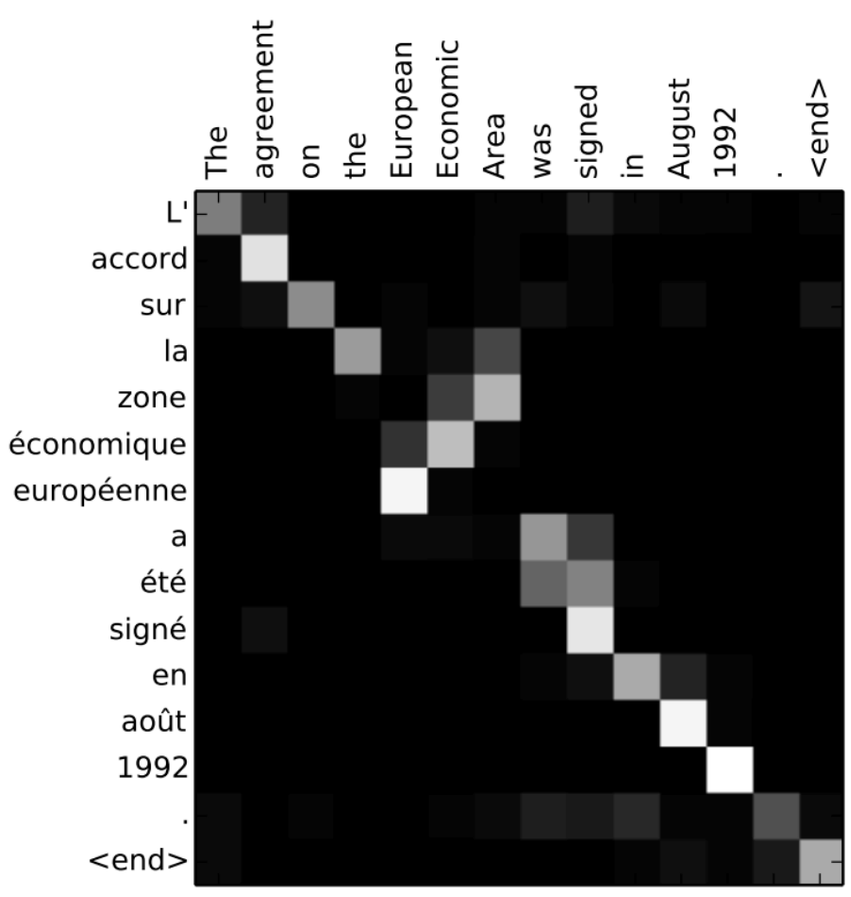
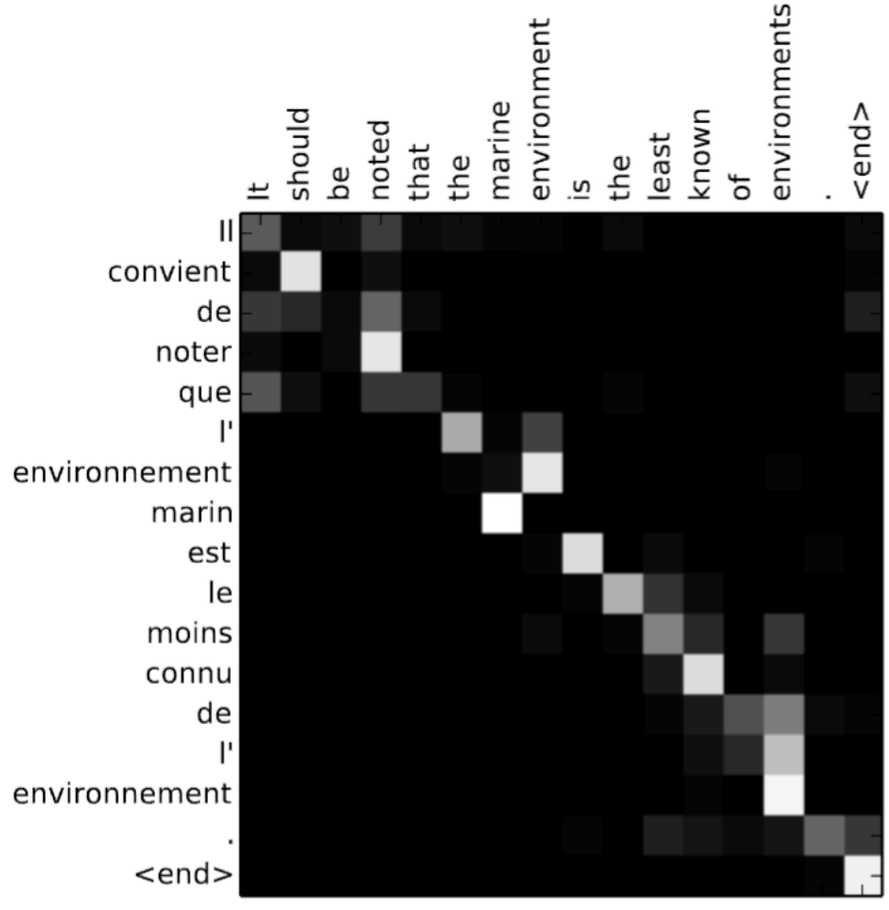


Figure 2: The BLEU scores of the generated translations on the test set with respect to the lengths of the sentences. The results are on the full test set which includes sentences having unknown words to the models.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. *International Conference on Learning Representations*.

Attention: visualization





Bahdanau et al ICLR 2015

Scalable and accurate deep learning with electronic health records

Alvin Rajkomar[1,2], Eyal Oren[1], Kai Chen[1], Andrew M. Dai[1], Nissan Hajaj[1], Michaela Hardt[1], Peter J. Liu[1], Xiaobing Liu[1], Jake Marcus[1], Mimi Sun[1], Patrik Sundberg[1], Hector Yee[1], Kun Zhang[1], Yi Zhang[1], Gerardo Flores[1], Gavin E. Duggan[1], Jamie Irvine[1], Quoc Le[1], Kurt Litsch[1], Alexander Mossin[1], Justin Tansuwan[1], De Wang[1], James Wexler[1], Jimbo Wilson[1], Dana Ludwig[2], Samuel L. Volchenboum[3], Katherine Chou[1], Michael Pearson[1], Srinivasan Madabushi[1], Nigam H. Shah[4], Atul J. Butte[2], Michael D. Howell[1], Claire Cui[1], Greg S. Corrado[1] and Jeffrey Dean[1]

npj | Digital Medicine

Interpreting attention?

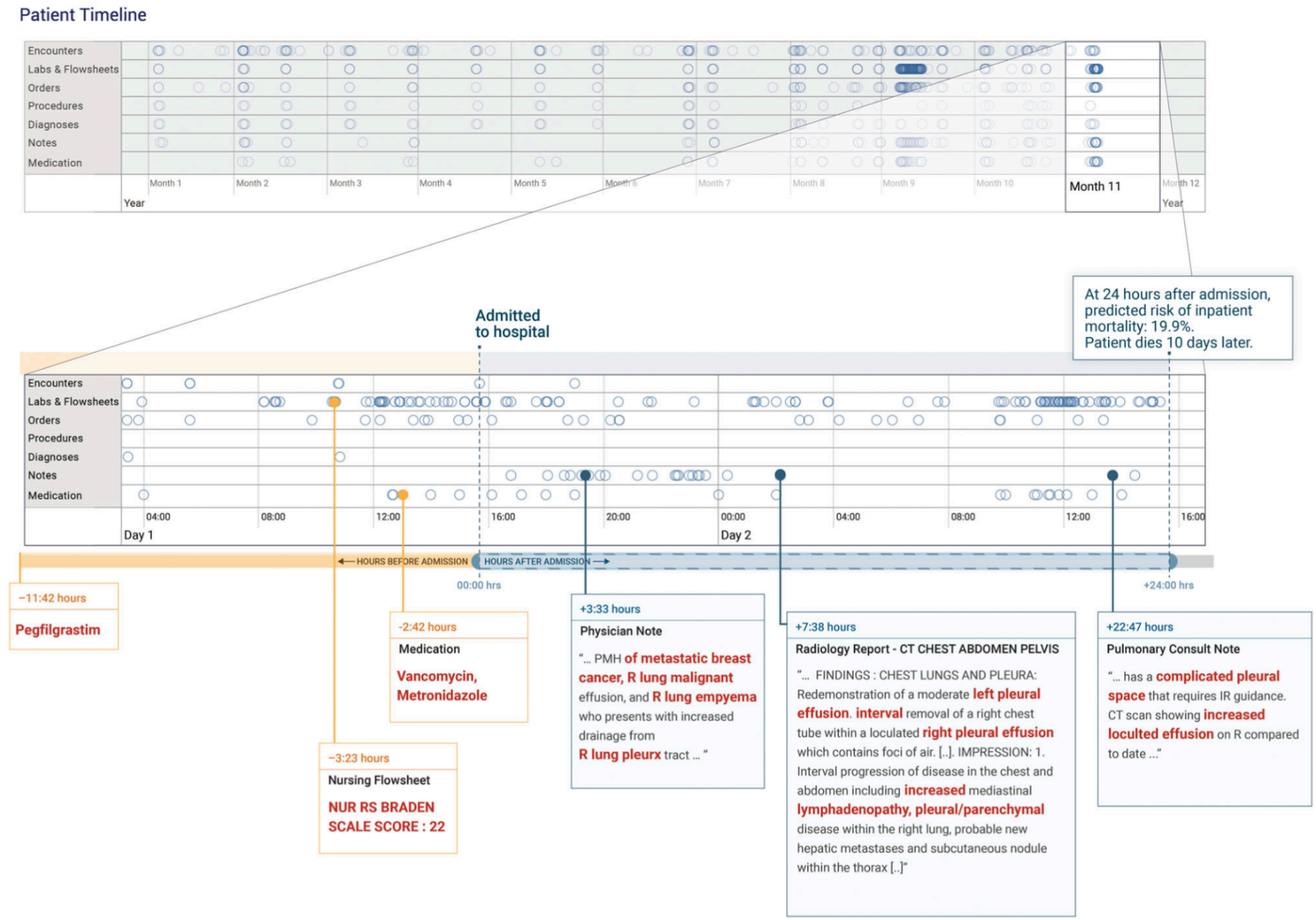


Fig. 3 The patient record shows a woman with metastatic breast cancer with malignant pleural effusions and empyema. The patient timeline at the top of the figure contains circles for every time-step for which at least a single token exists for the patient, and the horizontal lines show the data type. There is a close-up view of the most recent data points immediately preceding a prediction made 24 h after admission. We trained models for each data type and highlighted in red the tokens which the models attended to—the non-highlighted text was not attended to but is shown for context. The models pick up features in the medications, nursing flowsheets, and clinical notes relevant to the prediction

But is attention an explanation?

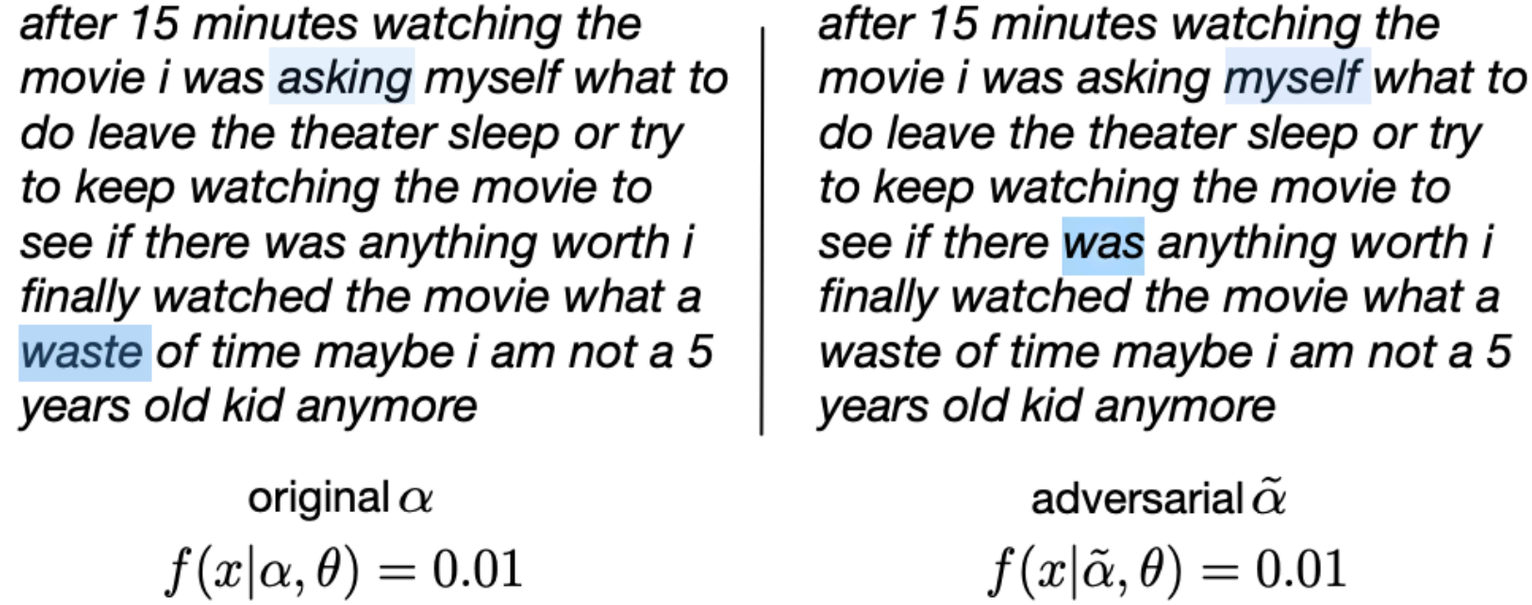


Figure 1: Heatmap of attention weights induced over a negative movie review. We show observed model attention (left) and an adversarially constructed set of attention weights (right). Despite being quite dissimilar, these both yield effectively the same prediction (0.01).

Jain, Sarthak, and Byron Wallace. 2019. Attention Is Not Explanation. *Conference of the North American Chapter of the Association for Computational Linguistics*.

Outline

- Recurrent Neural Networks
  - LSTMs
- Attention mechanism
- **Transformers**

Inductive biases in NN

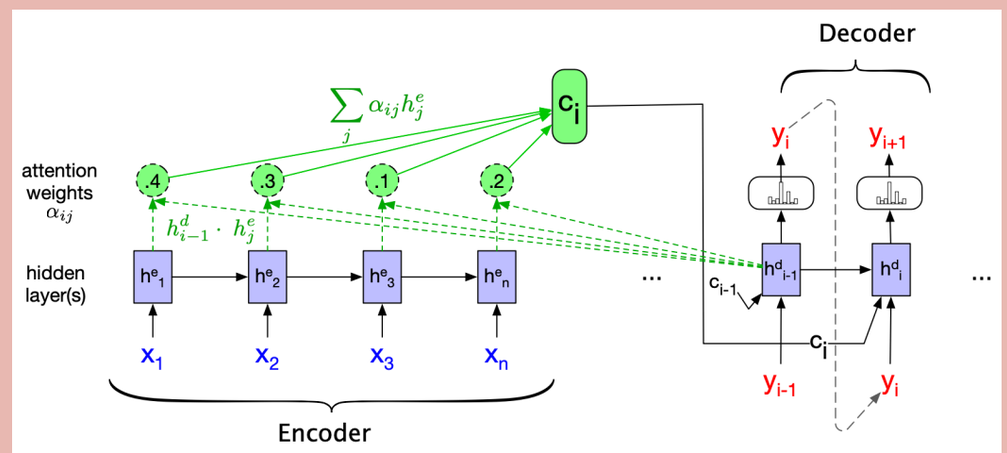
- Until very recently, the assumption was that we need inductive biases in the NN to help it generalize.

- **Convolutional neural networks:** Encodes translational invariance

- **Recurrent neural networks:** Also encodes translational invariance, but with memory

<u>Problems with these models:</u>

- *Computation is slow*



- *Learning dependencies between distant positions is difficult*

What if we threw it all out the window?

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

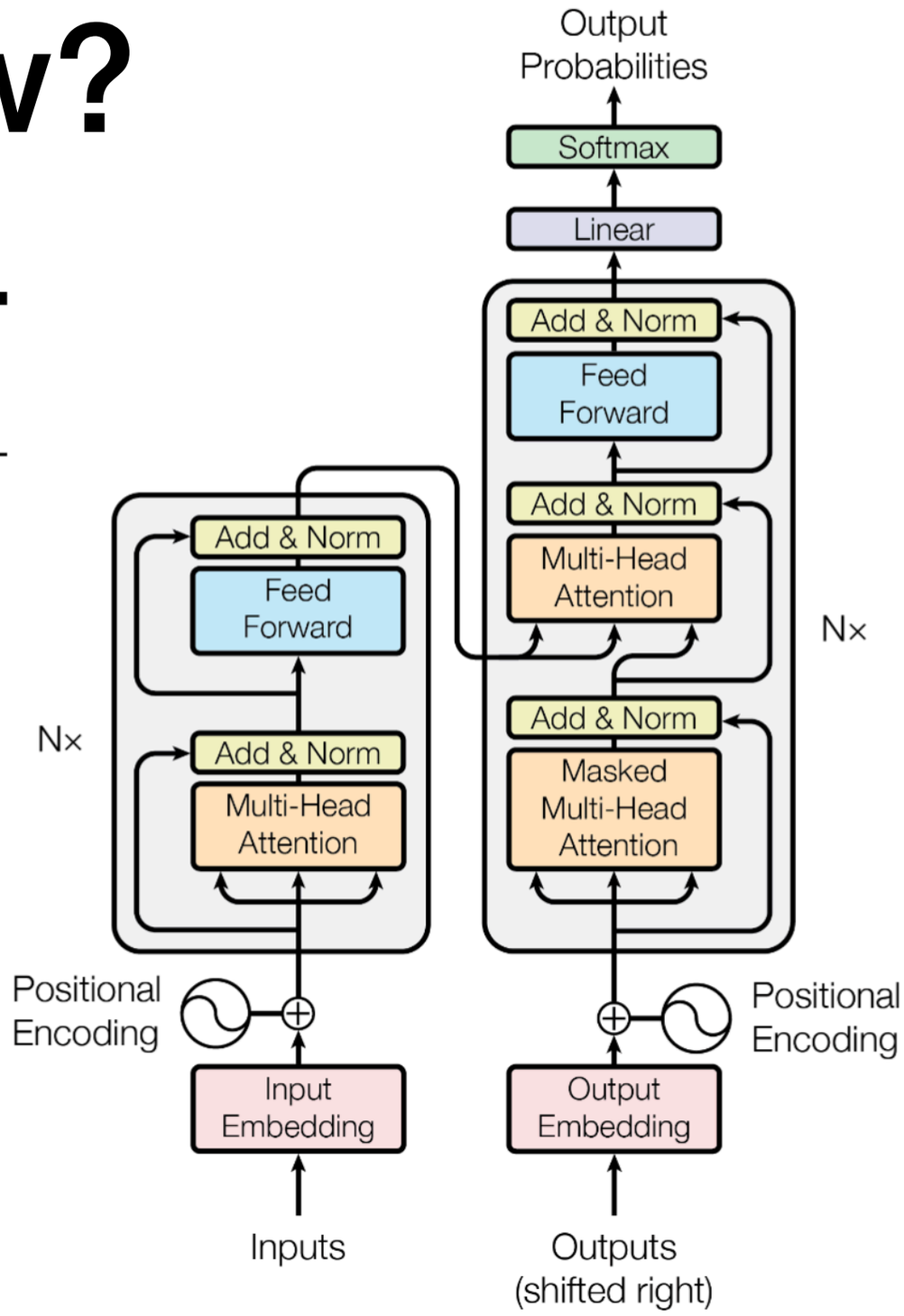
Jakob Uszkoreit* Google Research usz@google.com

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



Essentially zero inductive biases

NeurIPS 2017

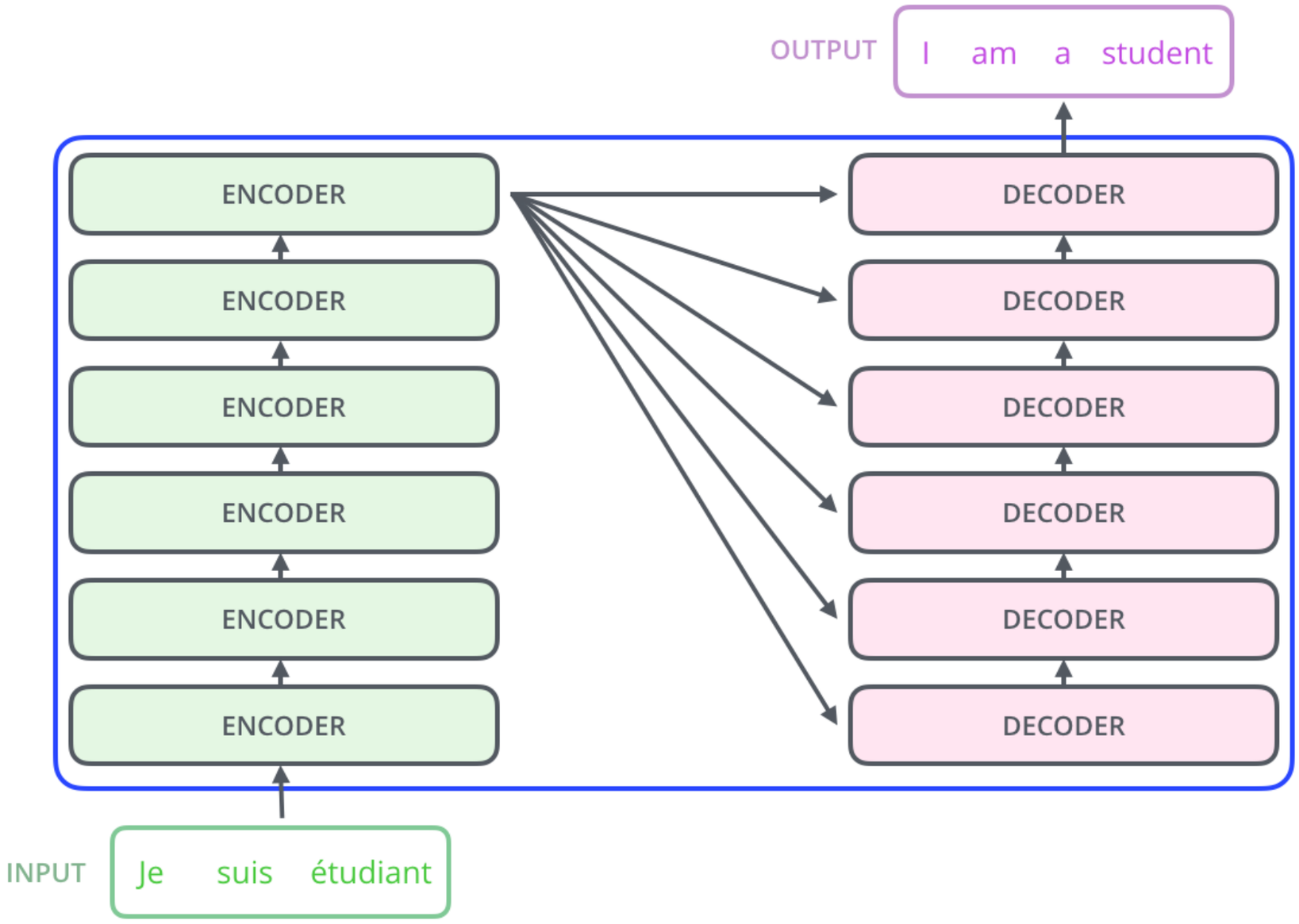
Shoot, it works well.

We trained on the standard WMT 2014 English-German dataset consisting of about 4.5 million sentence pairs. Sentences were encoded using byte-pair encoding [3], which has a shared source-target vocabulary of about 37000 tokens. For English-French, we used the significantly larger WMT 2014 English-French dataset consisting of 36M sentences and split tokens into a 32000 word-piece vocabulary [31]. Sentence pairs were batched together by approximate sequence length. Each training batch contained a set of sentence pairs containing approximately 25000 source tokens and 25000 target tokens.

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

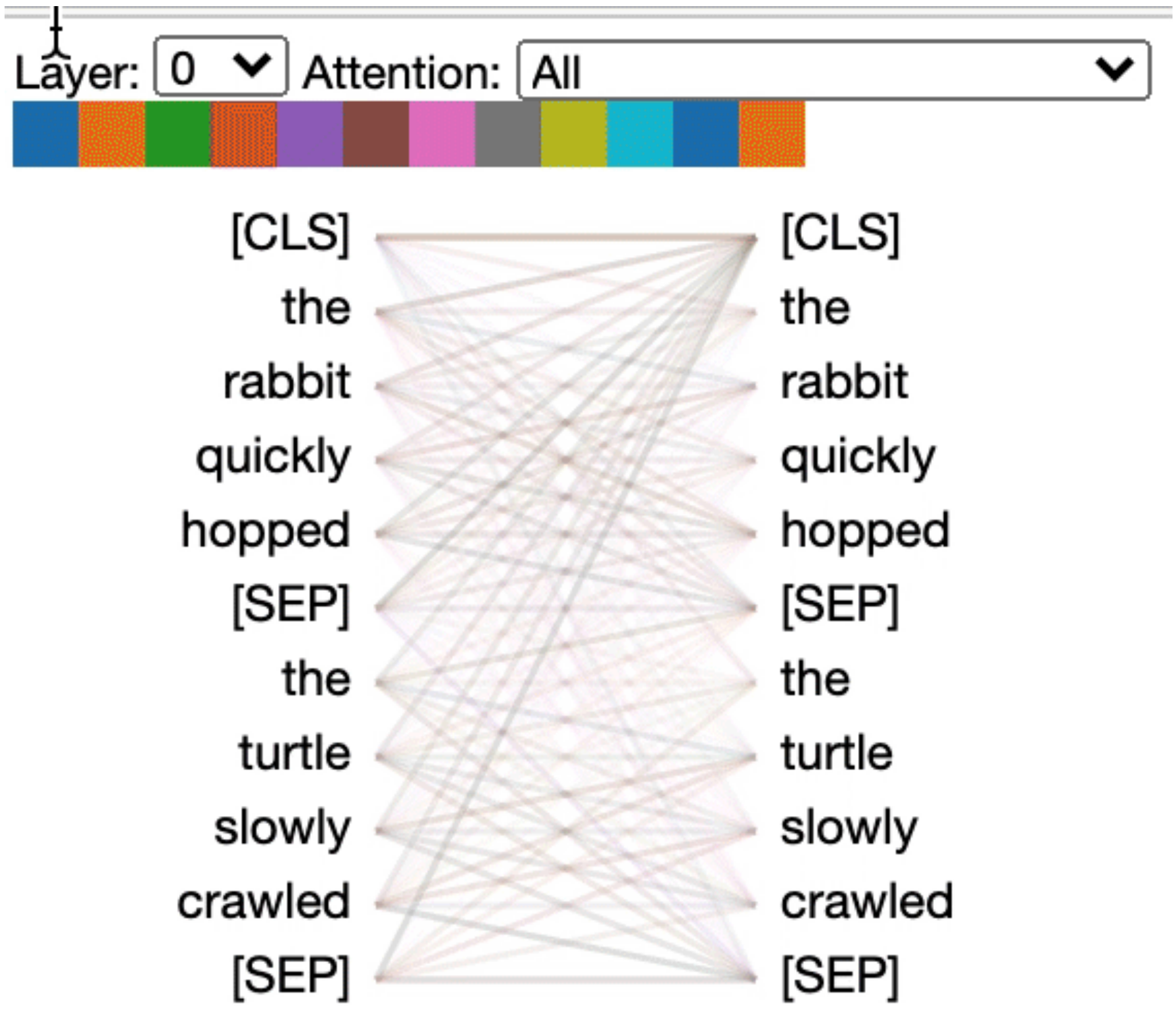
Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [15]	23.75			
Deep-Att + PosUnk [32]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [31]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [8]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [26]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [32]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [31]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [8]	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**	**41.0**	$2.3 \cdot 10^{19}$	

Understanding transformers



https://jalammar.github.io/illustrated-transformer/

So much attention



From transformers to GPT

Improving Language Understanding by Generative Pre-Training

Alec Radford
OpenAI
alec@openai.com

Karthik Narasimhan
OpenAI
karthikn@openai.com

Tim Salimans
OpenAI
tim@openai.com

Ilya Sutskever
OpenAI
ilyasu@openai.com

Abstract

Natural language understanding comprises a wide range of diverse tasks such as textual entailment, question answering, semantic similarity assessment, and document classification. Although large unlabeled text corpora are abundant, labeled data for learning these specific tasks is scarce, making it challenging for discriminatively trained models to perform adequately. We demonstrate that large gains on these tasks can be realized by *generative pre-training* of a language model on a diverse corpus of unlabeled text, followed by *discriminative fine-tuning* on each specific task. In contrast to previous approaches, we make use of task-aware input transformations during fine-tuning to achieve effective transfer while requiring minimal changes to the model architecture. We demonstrate the effectiveness of our approach on a wide range of benchmarks for natural language understanding. Our general task-agnostic model outperforms discriminatively trained models that use architectures specifically crafted for each task, significantly improving upon the state of the art in 9 out of the 12 tasks studied. For instance, we achieve absolute improvements of 8.9% on commonsense reasoning (Stories Cloze Test), 5.7% on question answering (RACE), and 1.5% on textual entailment (MultiNLI).

Key idea:
Pairing transformers with unsupervised pre-training

=> Unprecedented performance

From transformers to GPT

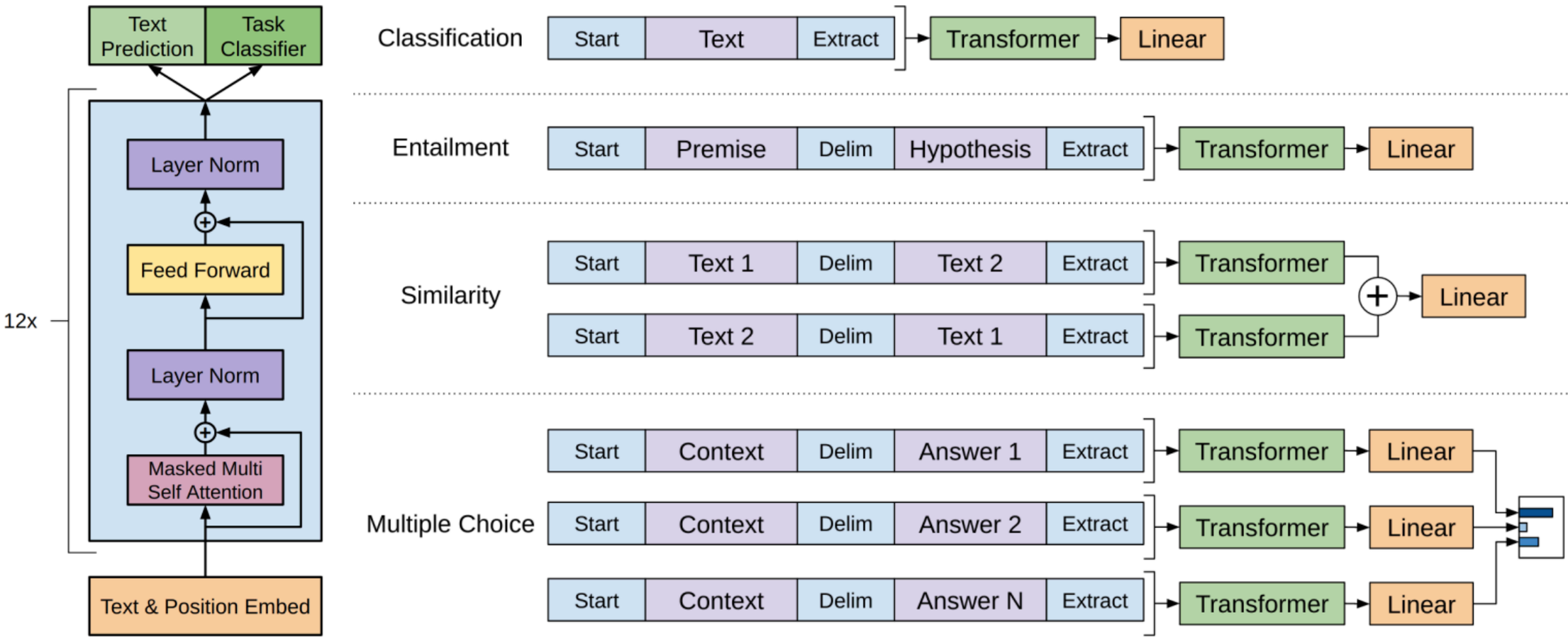


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

Turns out GPT straight out of the box gives funny answers…
How do we make it respond more like a human?

Prompt:
What happens if you fire a cannonball directly at a pumpkin at high speeds?

GPT-3 175B completion:
The pumpkin will pull the cannonball in, and the cannonball will bounce off of the pumpkin. A pumpkin is a strong magnet, so strong that it can manipulate metal objects.

Training language models to follow instructions with human feedback

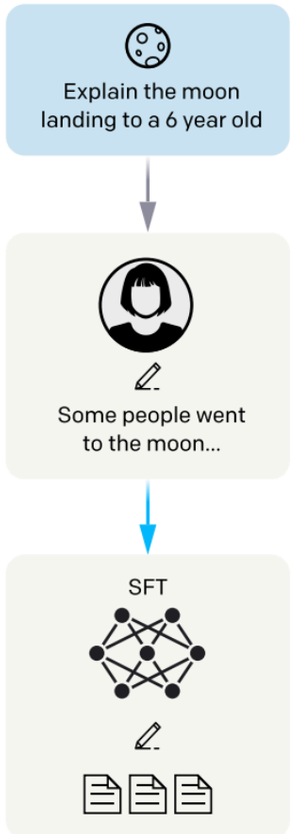
Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.



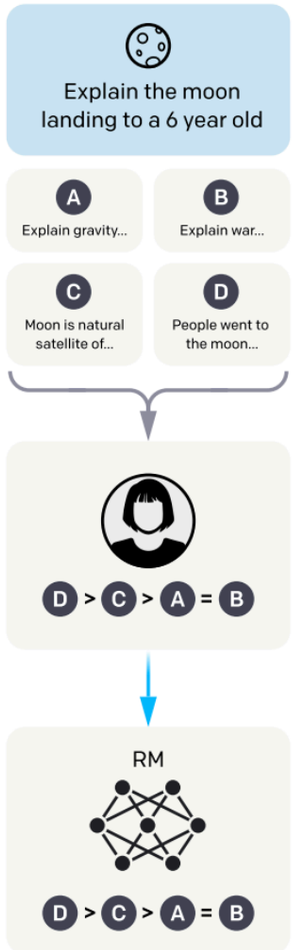
Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

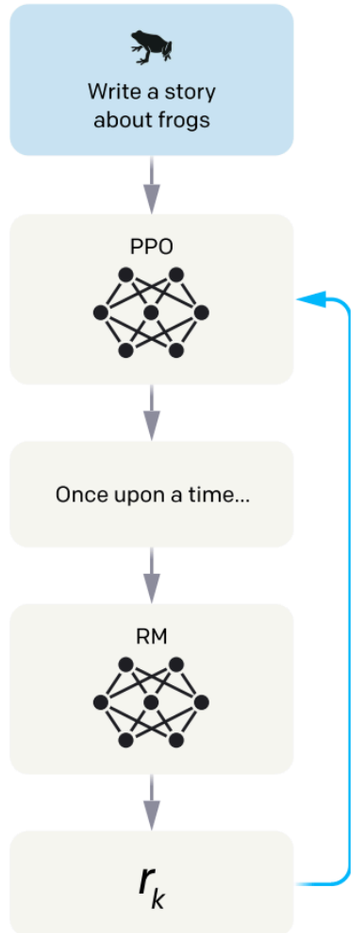
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



Training language models to follow instructions with human feedback

Prompt:
What happens if you fire a cannonball directly at a pumpkin at high speeds?

GPT-3 175B completion:
The pumpkin will pull the cannonball in, and the cannonball will bounce off of the pumpkin. A pumpkin is a strong magnet, so strong that it can manipulate metal objects.

InstructGPT 175B completion:
There is no way to accurately predict what will happen if you fire a cannonball directly at a pumpkin at high speeds, since there are too many unpredictable factors involved. Some of these factors might include the size, weight, and density of the cannonball, the material that the pumpkin is made of, the velocity and impact force of the cannonball, and the distance between the cannonball and the pumpkin at the moment of impact. Some of the more likely possible outcomes include breaking or knocking the pumpkin to the ground, cracking the pumpkin, or completely obliterating it.

Much better! :)

Foundation models

This paper coined the term “Foundation Models”

On the Opportunities and Risks of Foundation Models

Rishi Bommasani* Drew A. Hudson Ehsan Adeli Russ Altman Simran Arora
Sydney von Arx Michael S. Bernstein Jeannette Bohg Antoine Bosselut Emma Brunskill
Erik Brynjolfsson Shyamal Buch Dallas Card Rodrigo Castellon Niladri Chatterji
Annie Chen Kathleen Creel Jared Quincy Davis Dorottya Demszky Chris Donahue
Moussa Doumbouya Esin Durmus Stefano Ermon John Etchemendy Kawin Ethayarajh
Li Fei-Fei Chelsea Finn Trevor Gale Lauren Gillespie Karan Goel Noah Goodman
Shelby Grossman Neel Guha Tatsunori Hashimoto Peter Henderson John Hewitt
Daniel E. Ho Jenny Hong Kyle Hsu Jing Huang Thomas Icard Saahil Jain
Dan Jurafsky Pratyusha Kalluri Siddharth Karamcheti Geoff Keeling Fereshte Khani
Omar Khattab Pang Wei Koh Mark Krass Ranjay Krishna Rohith Kuditipudi
Ananya Kumar Faisal Ladhak Mina Lee Tony Lee Jure Leskovec Isabelle Levent
Xiang Lisa Li Xuechen Li Tengyu Ma Ali Malik Christopher D. Manning
Suvir Mirchandani Eric Mitchell Zanele Munyikwa Suraj Nair Avanika Narayan
Deepak Narayanan Ben Newman Allen Nie Juan Carlos Niebles Hamed Nilforoshan
Julian Nyarko Giray Ogut Laurel Orr Isabel Papadimitriou Joon Sung Park Chris Piech
Eva Portelance Christopher Potts Aditi Raghunathan Rob Reich Hongyu Ren
Frieda Rong Yusuf Roohani Camilo Ruiz Jack Ryan Christopher Ré Dorsa Sadigh
Shiori Sagawa Keshav Santhanam Andy Shih Krishnan Srinivasan Alex Tamkin
Rohan Taori Armin W. Thomas Florian Tramèr Rose E. Wang William Wang Bohan Wu
Jiajun Wu Yuhuai Wu Sang Michael Xie Michihiro Yasunaga Jiaxuan You Matei Zaharia
Michael Zhang Tianyi Zhang Xikun Zhang Yuhui Zhang Lucia Zheng Kaitlyn Zhou
Percy Liang*[1]

Center for Research on Foundation Models (CRFM)
Stanford Institute for Human-Centered Artificial Intelligence (HAI)
Stanford University

AI is undergoing a paradigm shift with the rise of models (e.g., BERT, DALL-E, GPT-3) trained on broad data (generally using self-supervision at scale) that can be adapted to a wide range of downstream tasks. We call these models foundation models *to underscore their critically central yet incomplete character. This report provides a thorough account of the opportunities and risks of foundation models, ranging from their capabilities (e.g., language, vision, robotic manipulation, reasoning, human interaction) and technical principles (e.g., model architectures, training procedures, data, systems, security, evaluation, theory) to their applications (e.g., law, healthcare, education) and societal impact (e.g., inequity, misuse, economic and environmental impact, legal and ethical considerations). Though foundation models are based on standard deep learning and transfer learning, their scale results in new emergent capabilities, and their effectiveness across so many tasks incentivizes homogenization. Homogenization provides powerful leverage but demands caution, as the defects of the foundation model are inherited by all the adapted models downstream. Despite the impending widespread deployment of foundation models, we currently lack a clear understanding of how they work, when they fail, and what they are even capable of due to their emergent properties. To tackle these questions, we believe much of the critical research on foundation models will require deep interdisciplinary collaboration commensurate with their fundamentally sociotechnical nature.*

Foundation models

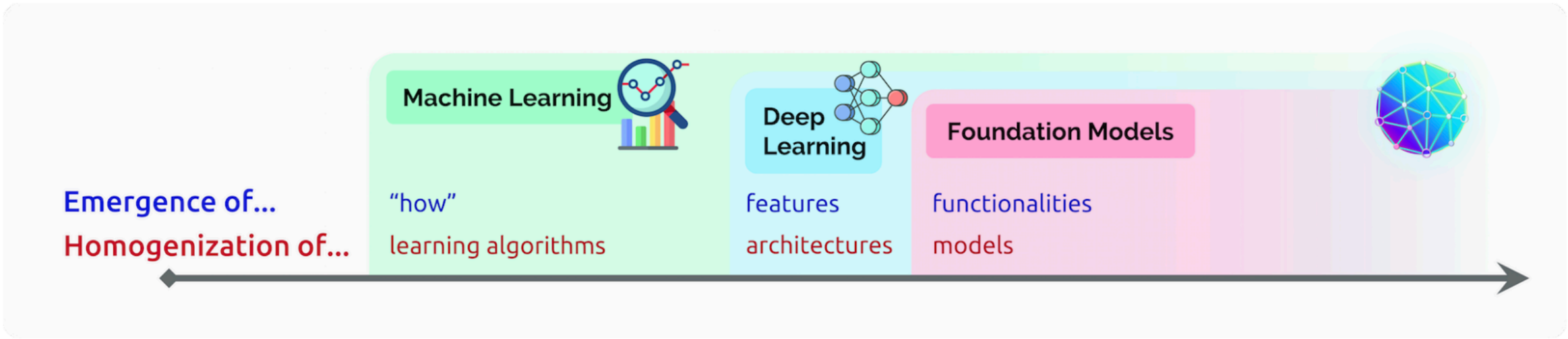
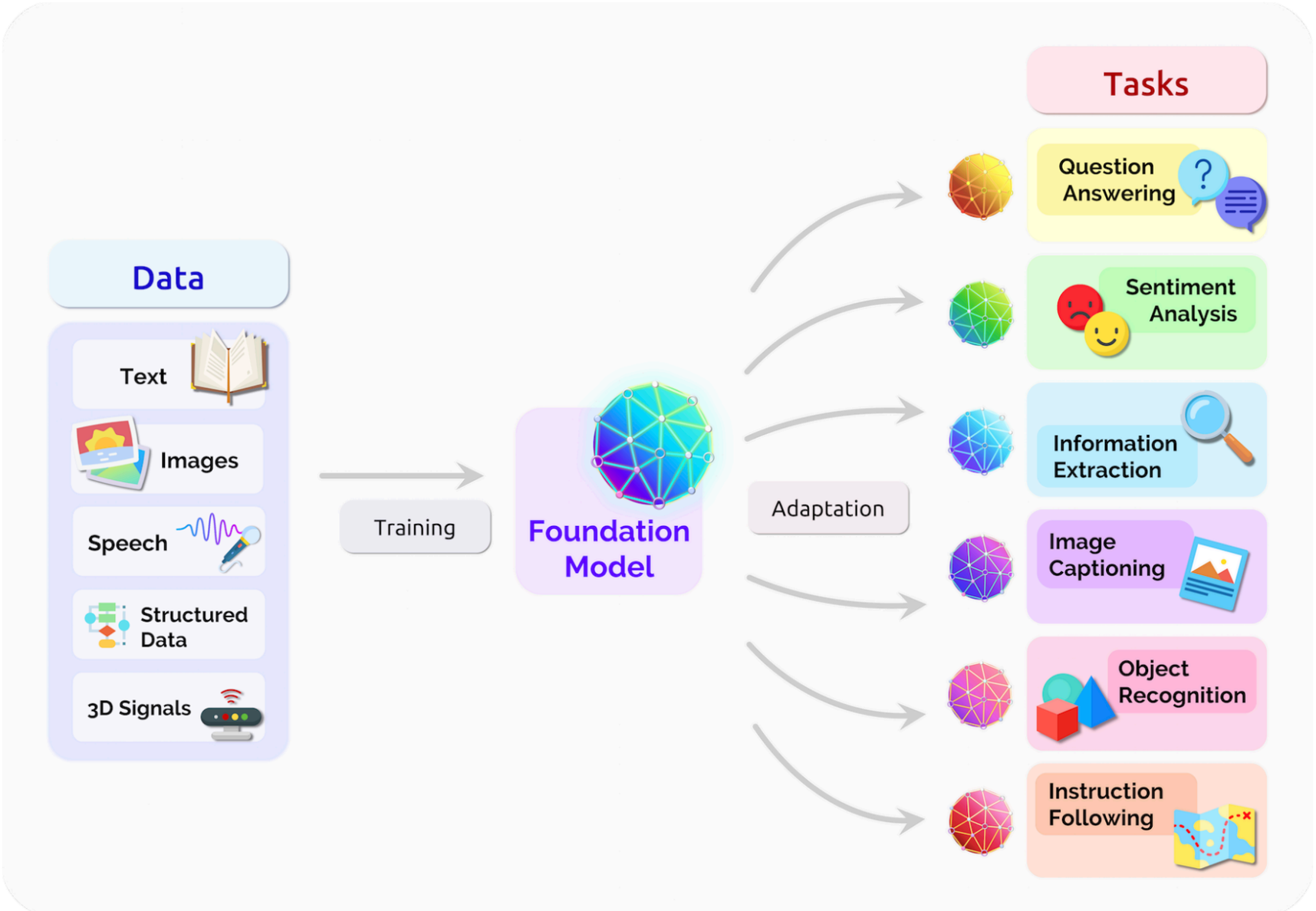


Fig. 1. The story of AI has been one of increasing *emergence* and *homogenization*. With the introduction of machine learning, *how* a task is performed emerges (is inferred automatically) from examples; with deep learning, the high-level features used for prediction emerge; and with foundation models, even advanced functionalities such as in-context learning emerge. At the same time, machine learning homogenizes learning algorithms (e.g., logistic regression), deep learning homogenizes model architectures (e.g., Convolutional Neural Networks), and foundation models homogenizes the model itself (e.g., GPT-3).

Foundation models



Lab (if time allows):

https://colab.research.google.com/github/huggingface/education-toolkit/blob/main/03_getting-started-with-transformers.ipynb#scrollTo=C85OWgOu5coO

References

- “Dive into Deep Learning” by Aston Zhang, Alexander J. Smola, Zachary Lipton, Mu Li

- “Speech and Language Processing” by Dan Jurafsky and James H. Martin (https://web.stanford.edu/~jurafsky/slp3/)

- https://cs229.stanford.edu/main_notes.pdf