

Deep learning for text processing

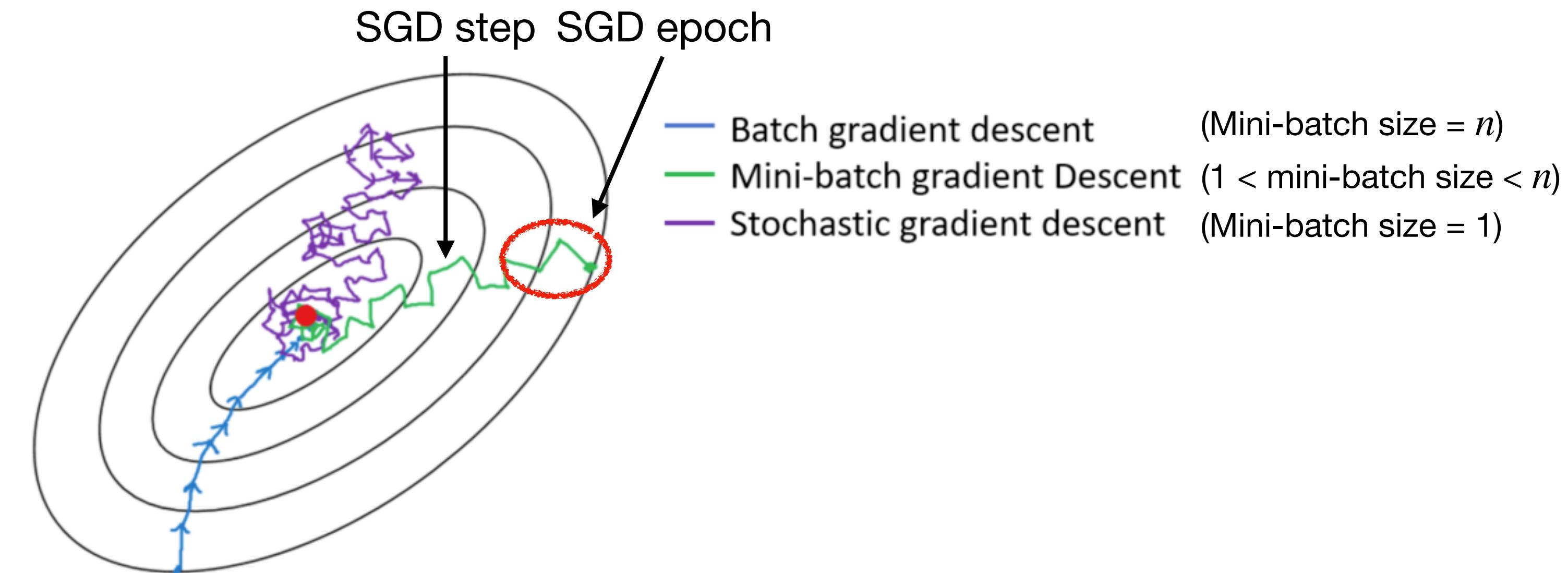
STAT 471

December 2, 2021

Acknowledgement: Material drawn from Stanford's CS 224n class (<http://web.stanford.edu/class/cs224n/index.html>)

Stochastic gradient descent step versus epoch (Clarification)

An epoch consists of SGD steps that cycle through all of the observations once.



Source: <https://medium.com/analytics-vidhya/gradient-descent-vs-stochastic-gd-vs-mini-batch-sgd-fbd3a2cb4ba4>

E.g. $n = 9$, mini-batch size 3. Then each epoch consists of three SGD steps.

Observation	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
SGD step (mini-batch size 1)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SGD step (mini-batch size 3)	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6												
Gradient step	Step 1						Step 2						Step 3					
Epoch	Epoch 1						Epoch 2											

Convolutional layer versus fully-connected layer

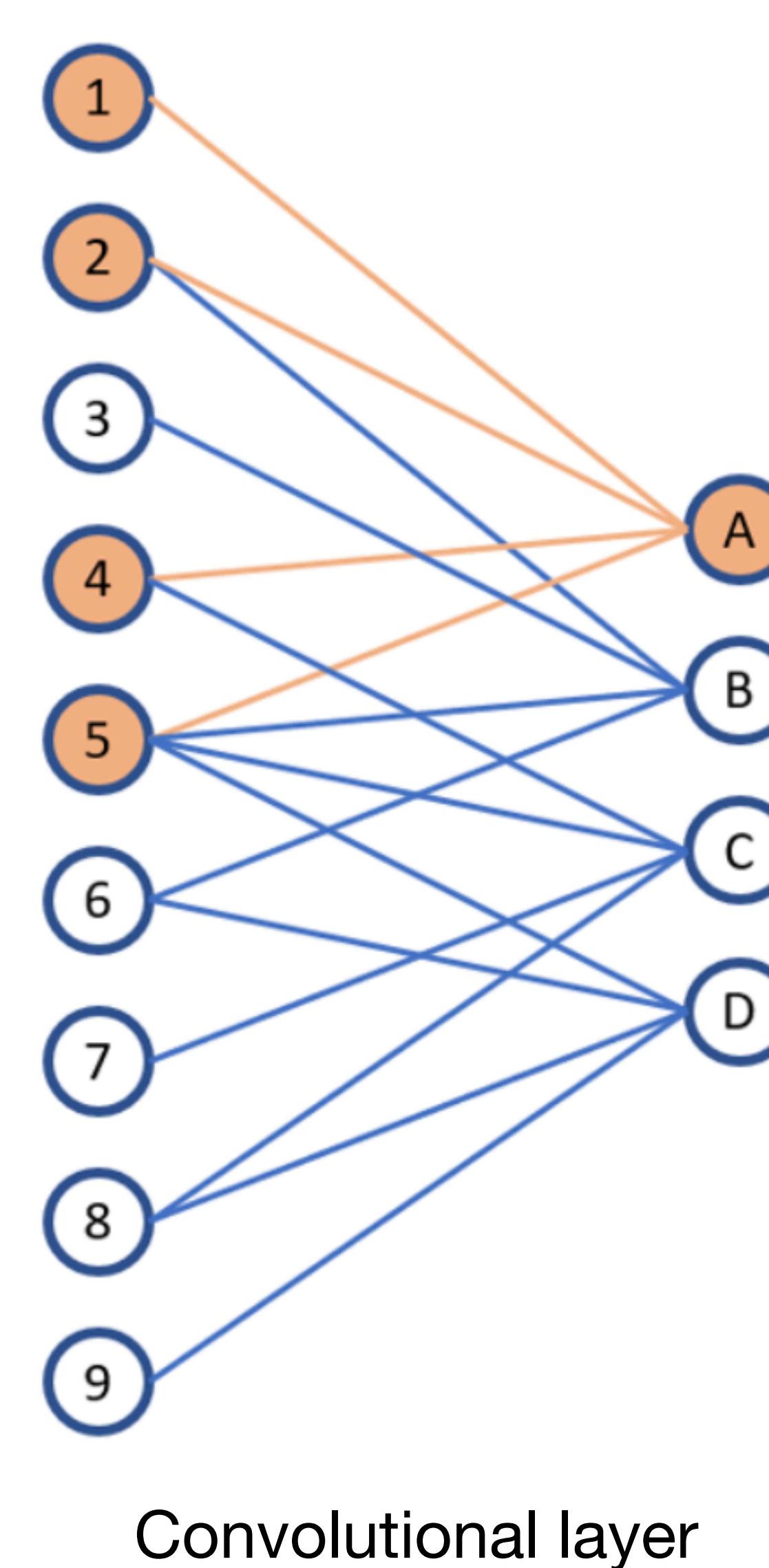
(Clarification)

A convolutional layer can be visualized similarly to a fully-connected layer.

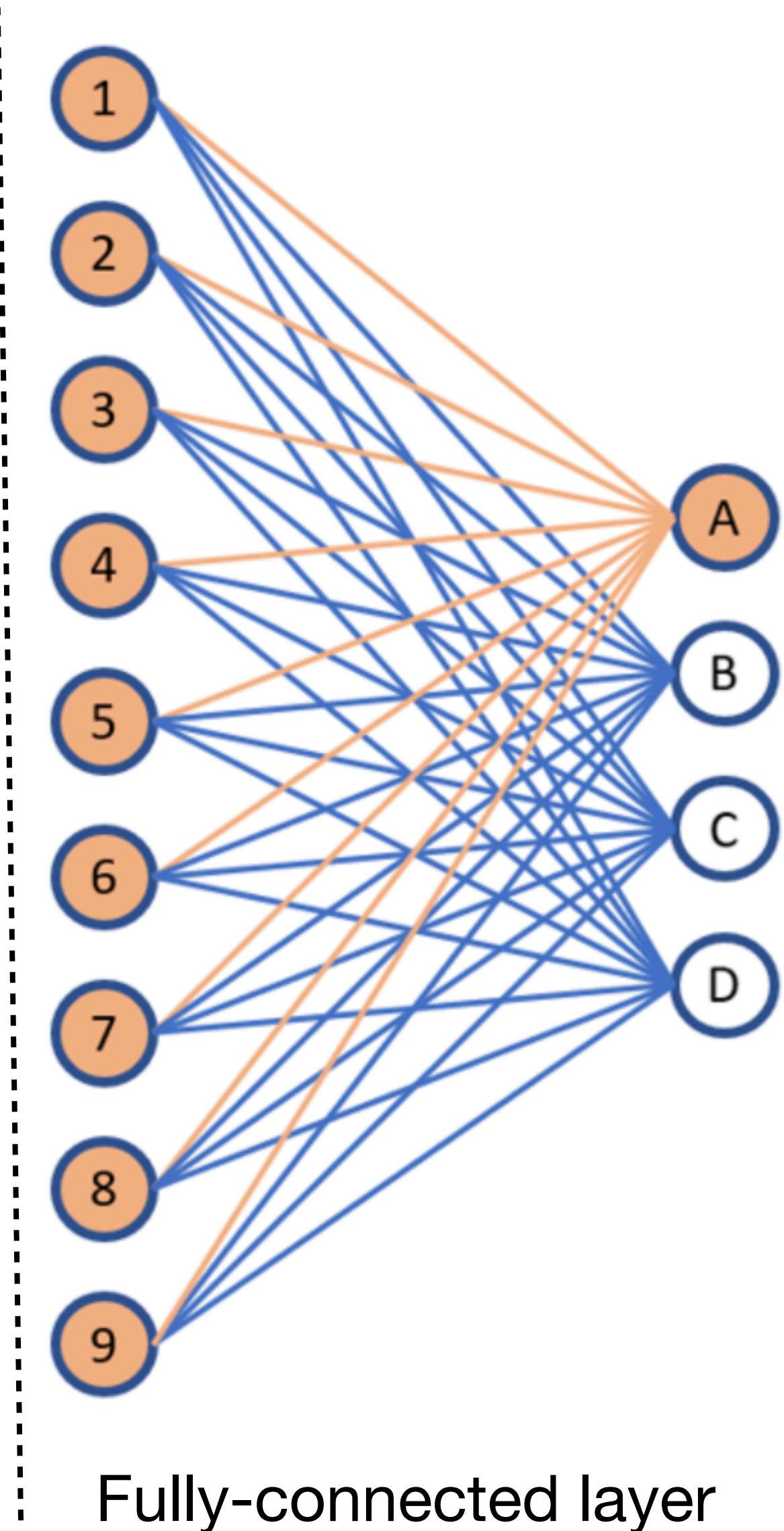
$$\begin{bmatrix} \text{Input} & \otimes & \text{Filter} \end{bmatrix} = \begin{bmatrix} \text{Output} \end{bmatrix}$$

The diagram illustrates the computation of a convolutional layer. On the left, an input matrix f (3x3) is multiplied by a filter matrix (2x2). The result is a 2x2 output matrix where each element is the sum of the overlapping products of the input and filter elements. The input matrix f is labeled "Input" and the filter matrix is labeled "Filter". The resulting output matrix is labeled "Output".

Activation
function



Convolutional layer



Fully-connected layer

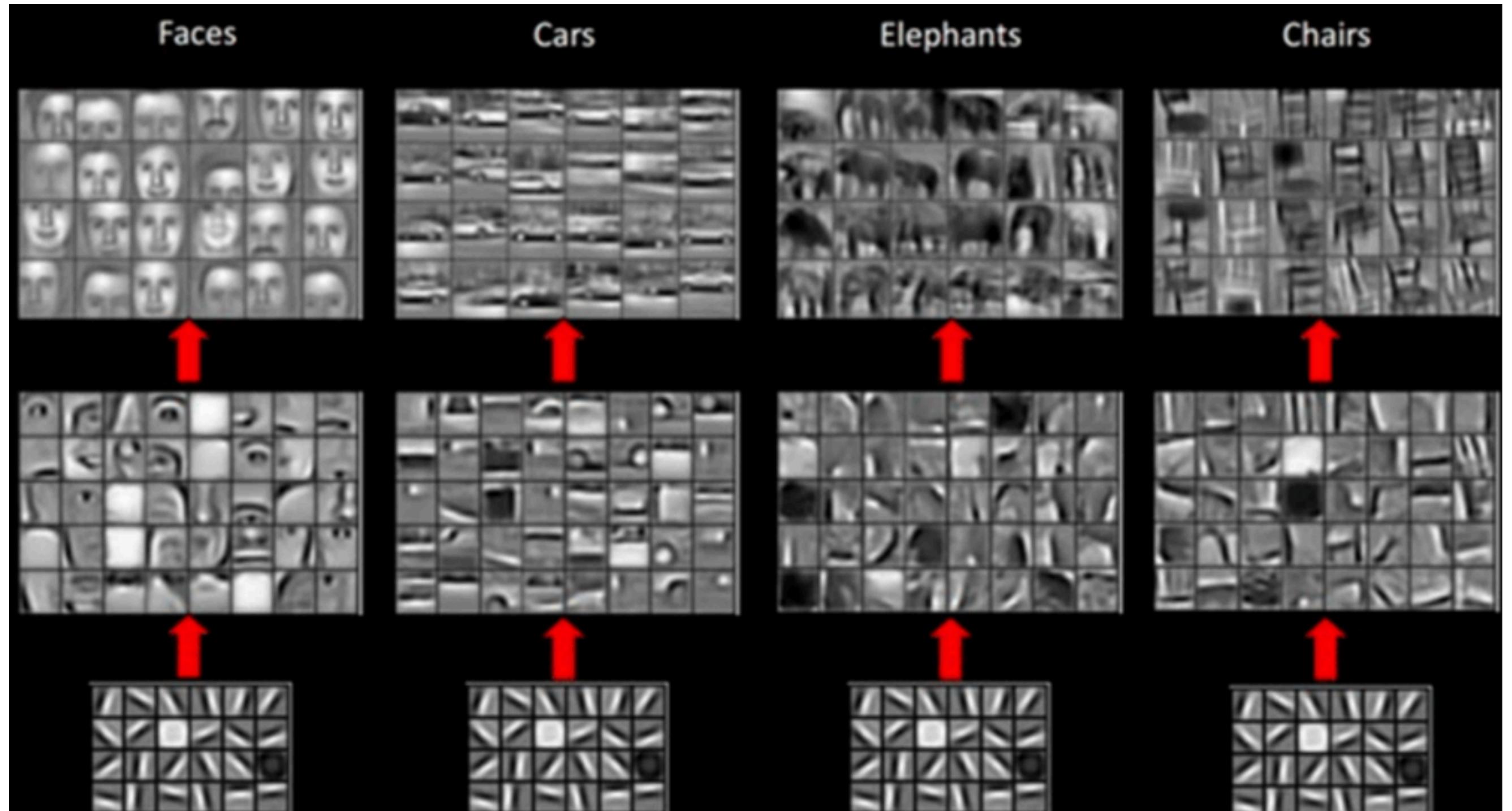
CNN visualization

(Clarification)

What are these images?

Initial layer: they are filters.

Subsequent layers: input images that activate different neurons the most strongly.



Where we are

- ✓ **Unit 1:** Intro to modern data mining
- ✓ **Unit 2:** Tuning predictive models
- ✓ **Unit 3:** Regression-based methods
- ✓ **Unit 4:** Tree-based methods
- Unit 5:** Deep learning

Lecture 1: Deep learning preliminaries

Lecture 2: Neural networks

Lecture 3: Deep learning for images

Lecture 4: Deep learning for text

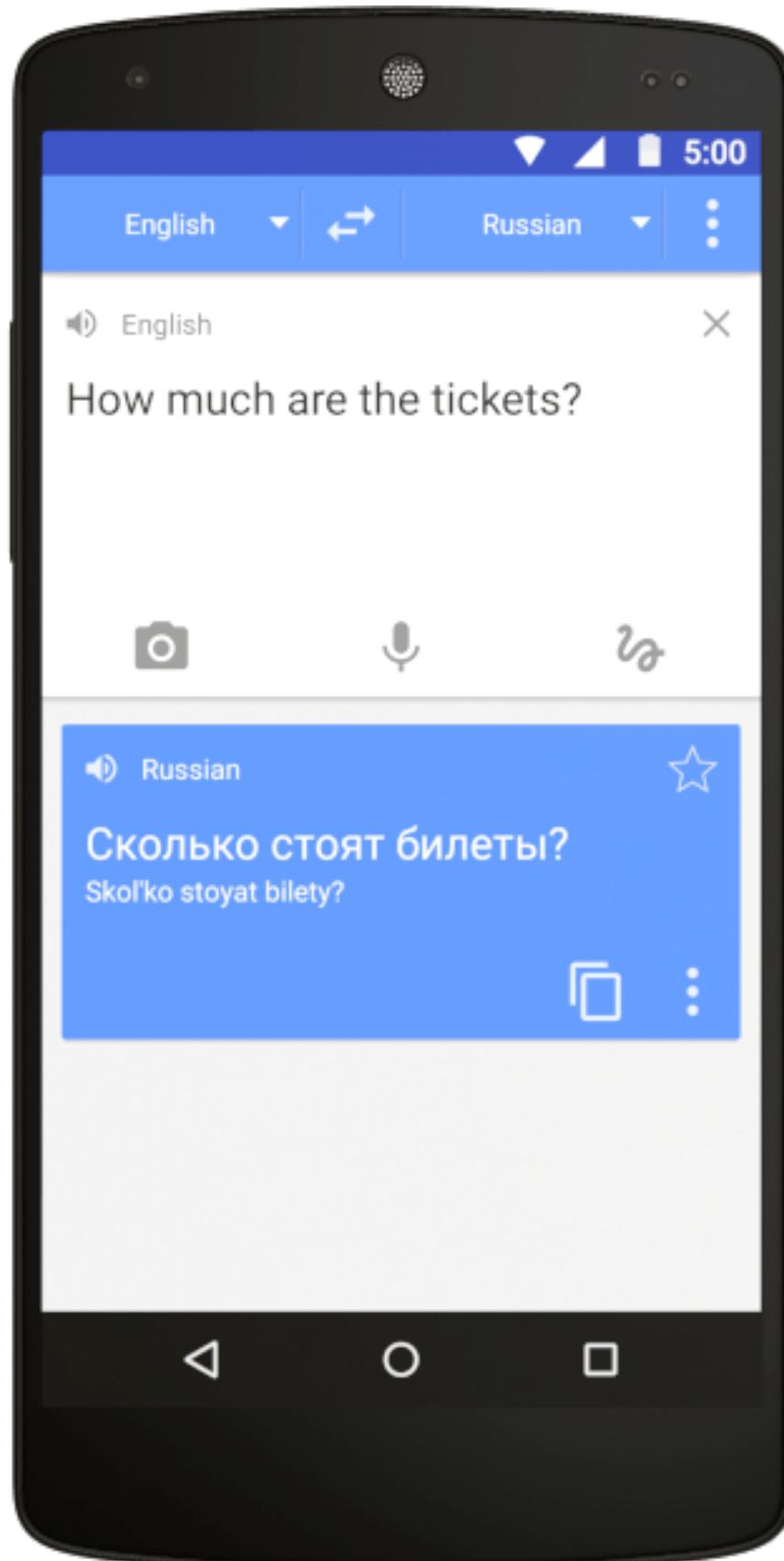
Lecture 5: Unit review and quiz in class

Homework 5 due the following Wednesday.

Applications of natural language processing

Applications of natural language processing

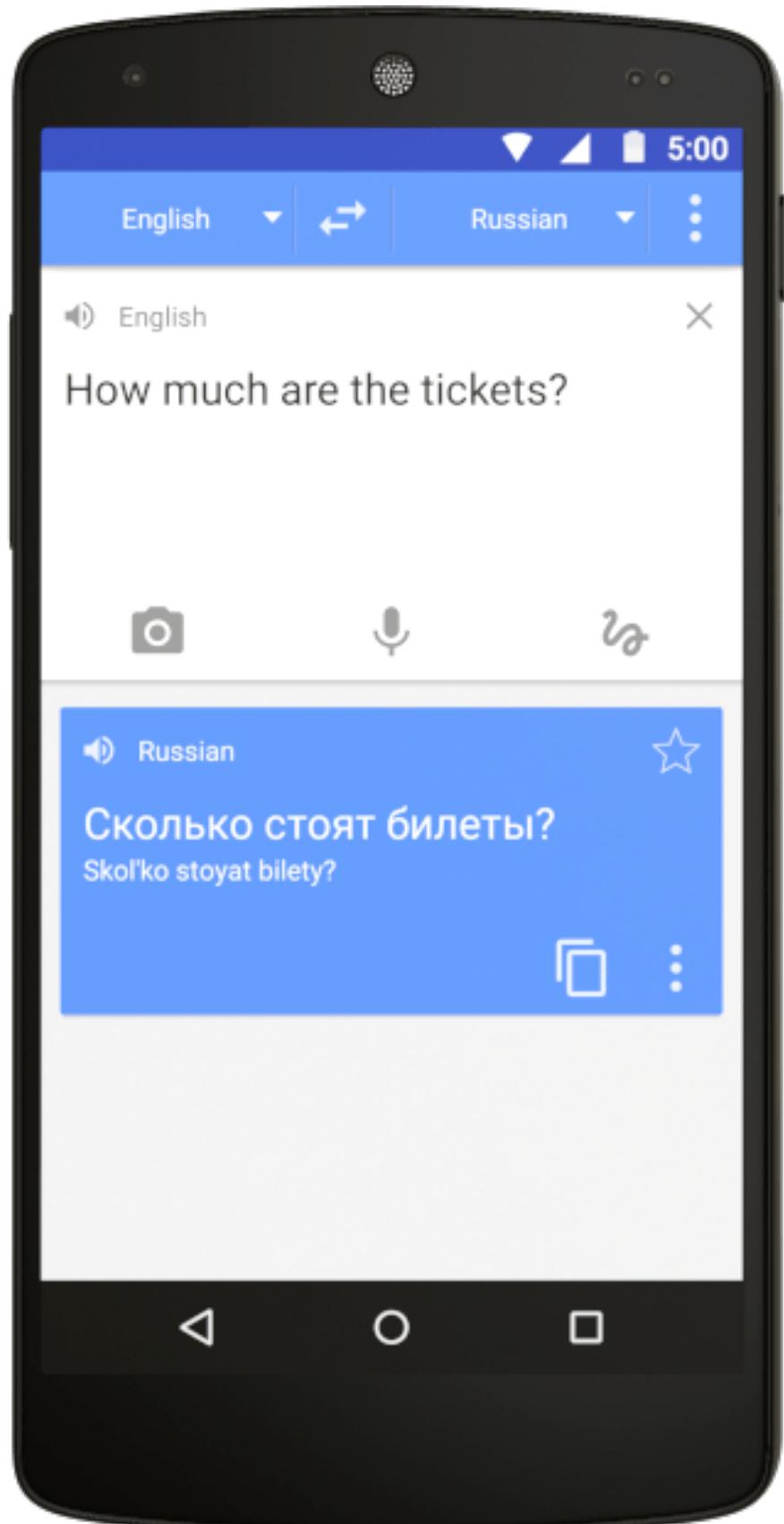
Machine translation



<https://translate.google.com/intl/en/about/>

Applications of natural language processing

Machine translation



Chatbots

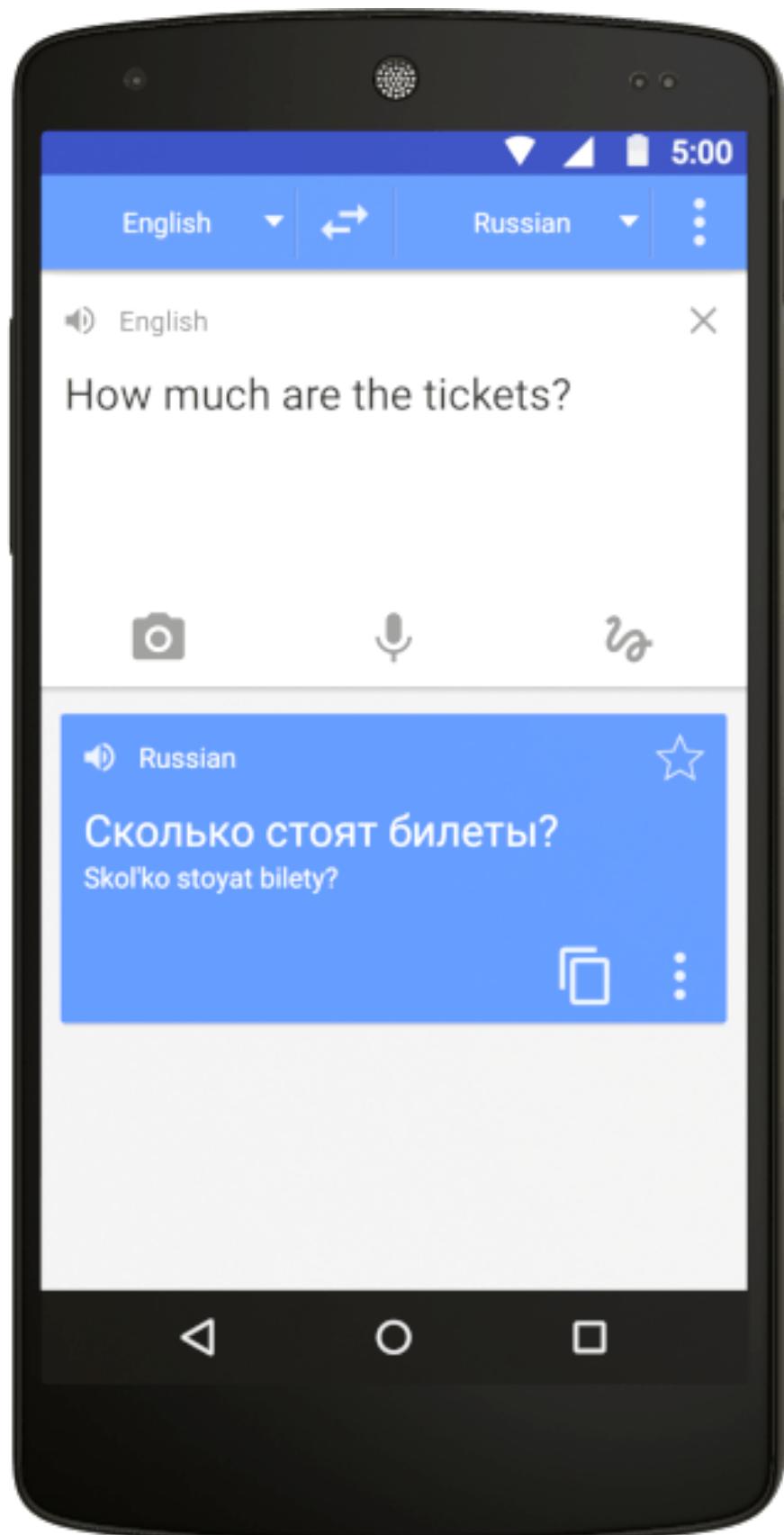


<https://translate.google.com/intl/en/about/>

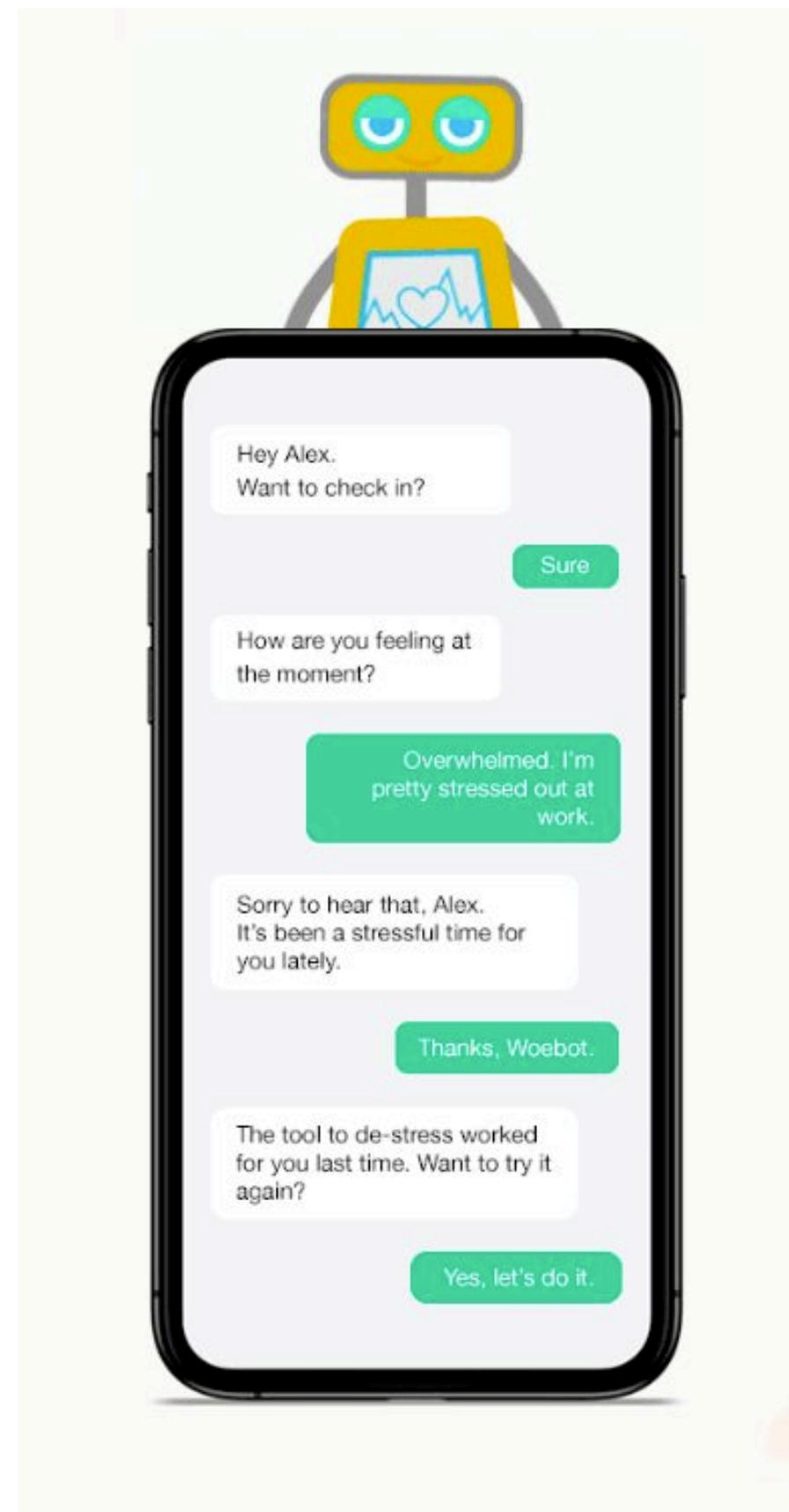
<https://www.trendhunter.com/trends/woebot>

Applications of natural language processing

Machine translation

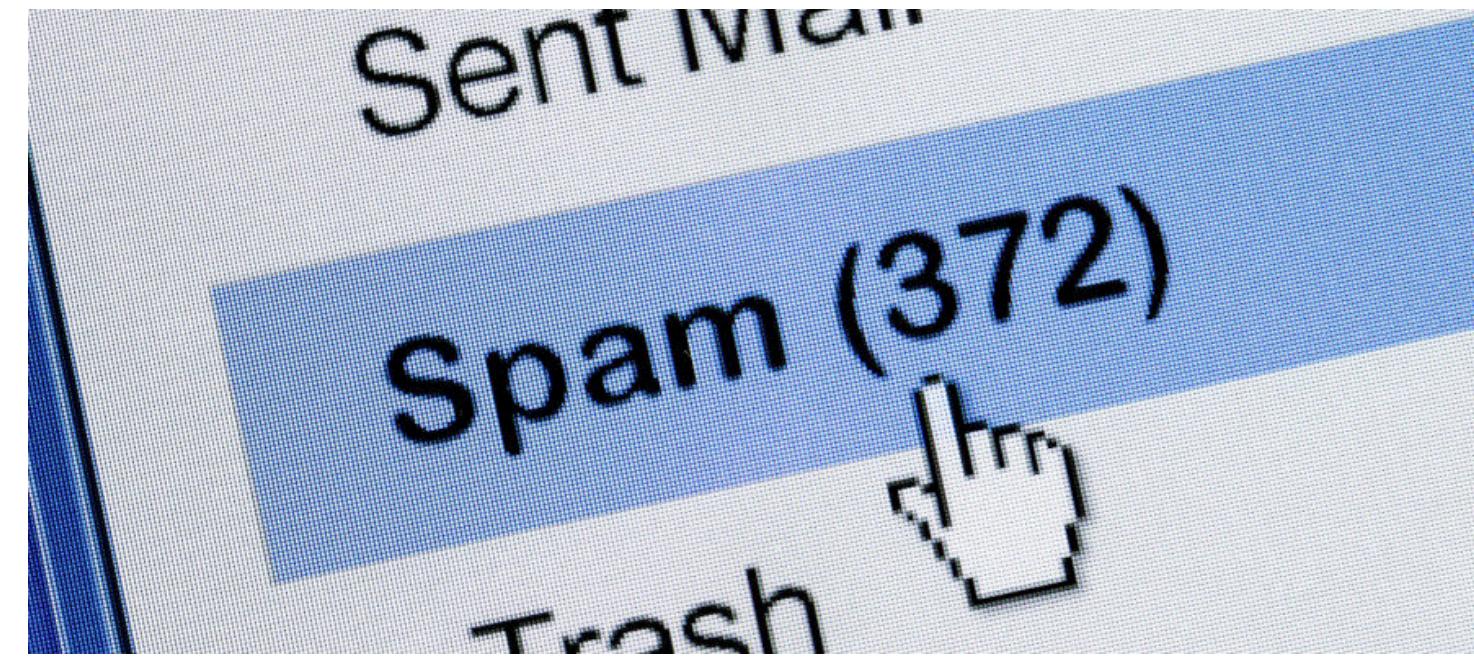


Chatbots



<https://translate.google.com/intl/en/about/>

Spam filtering

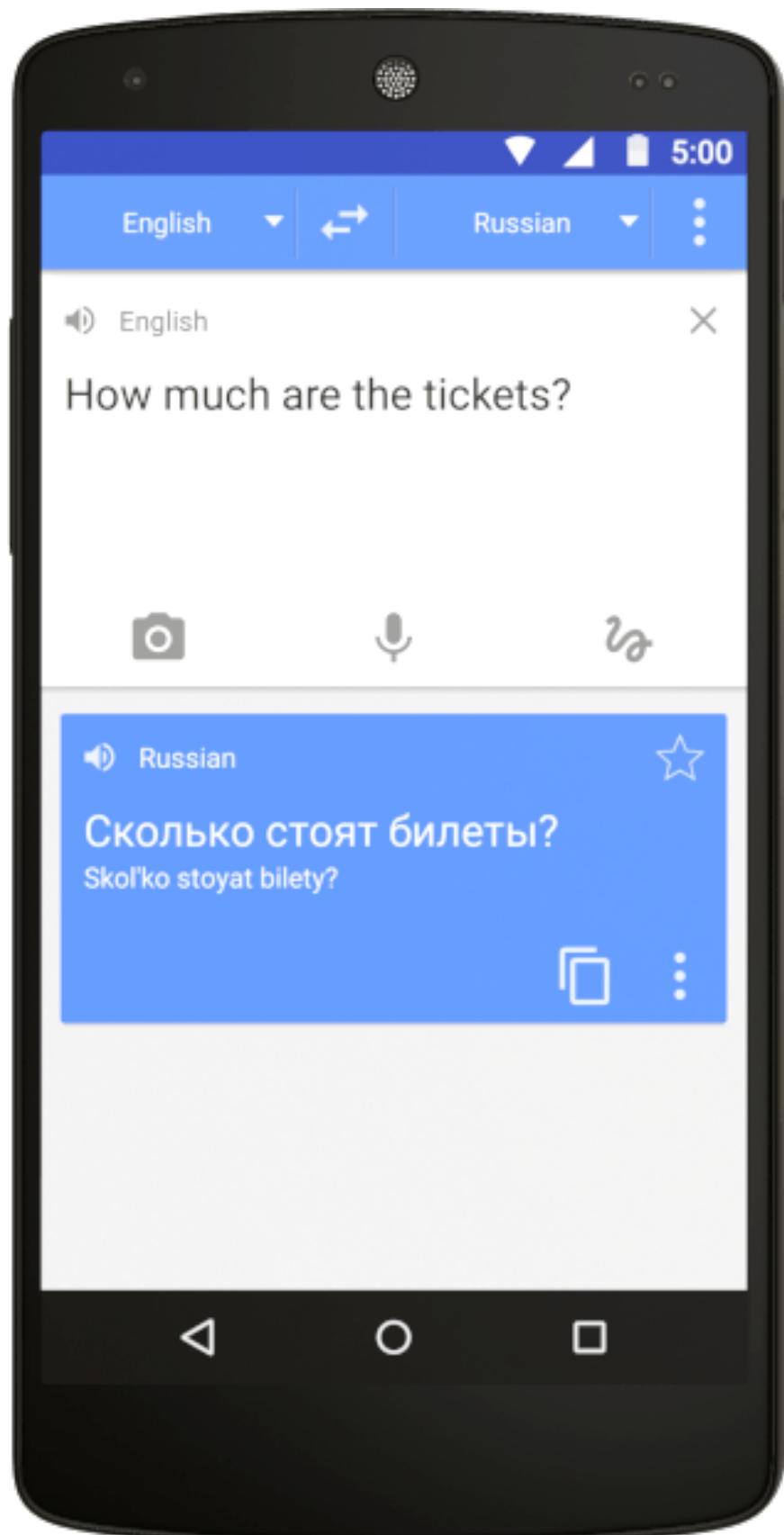


<https://blog.malwarebytes.com/security-world/2017/02/explained-bayesian-spam-filtering/>

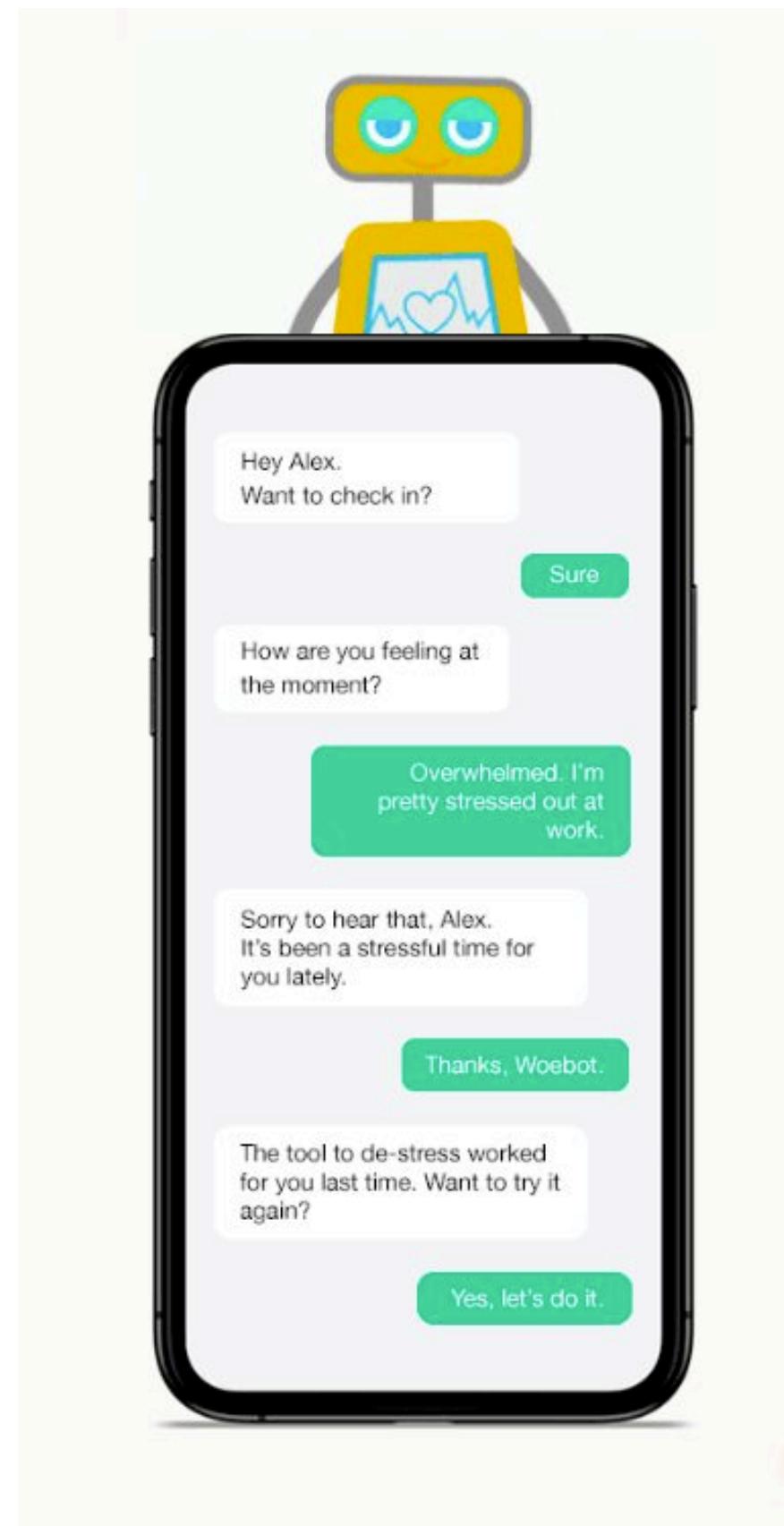
<https://www.trendhunter.com/trends/woebot>

Applications of natural language processing

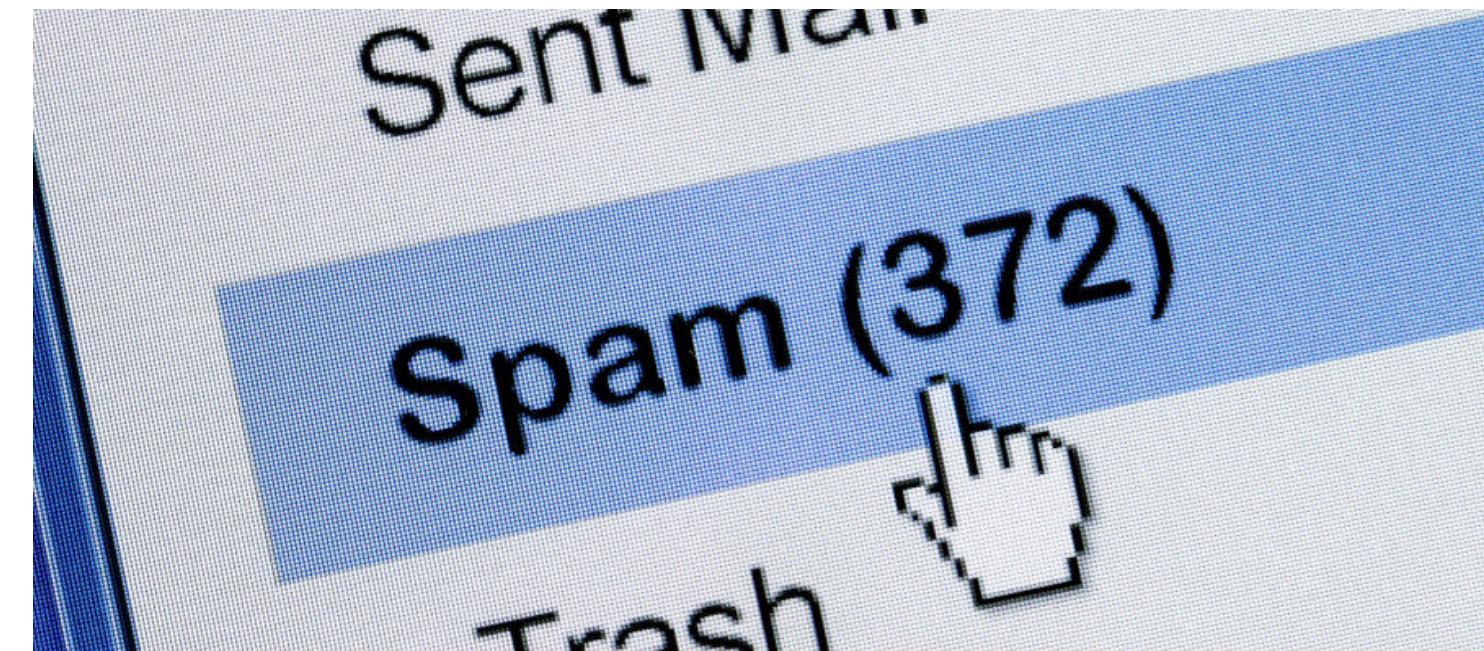
Machine translation



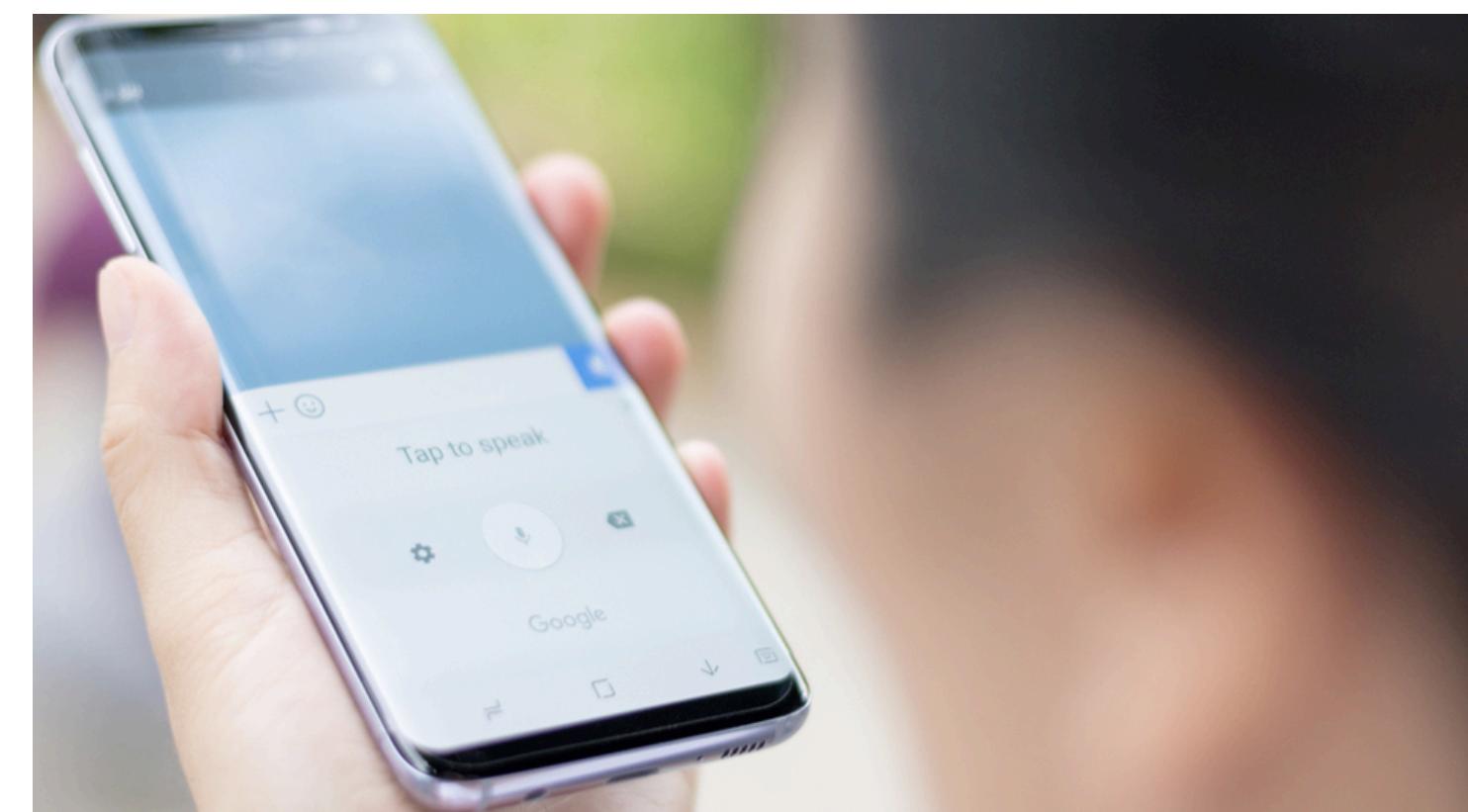
Chatbots



Spam filtering

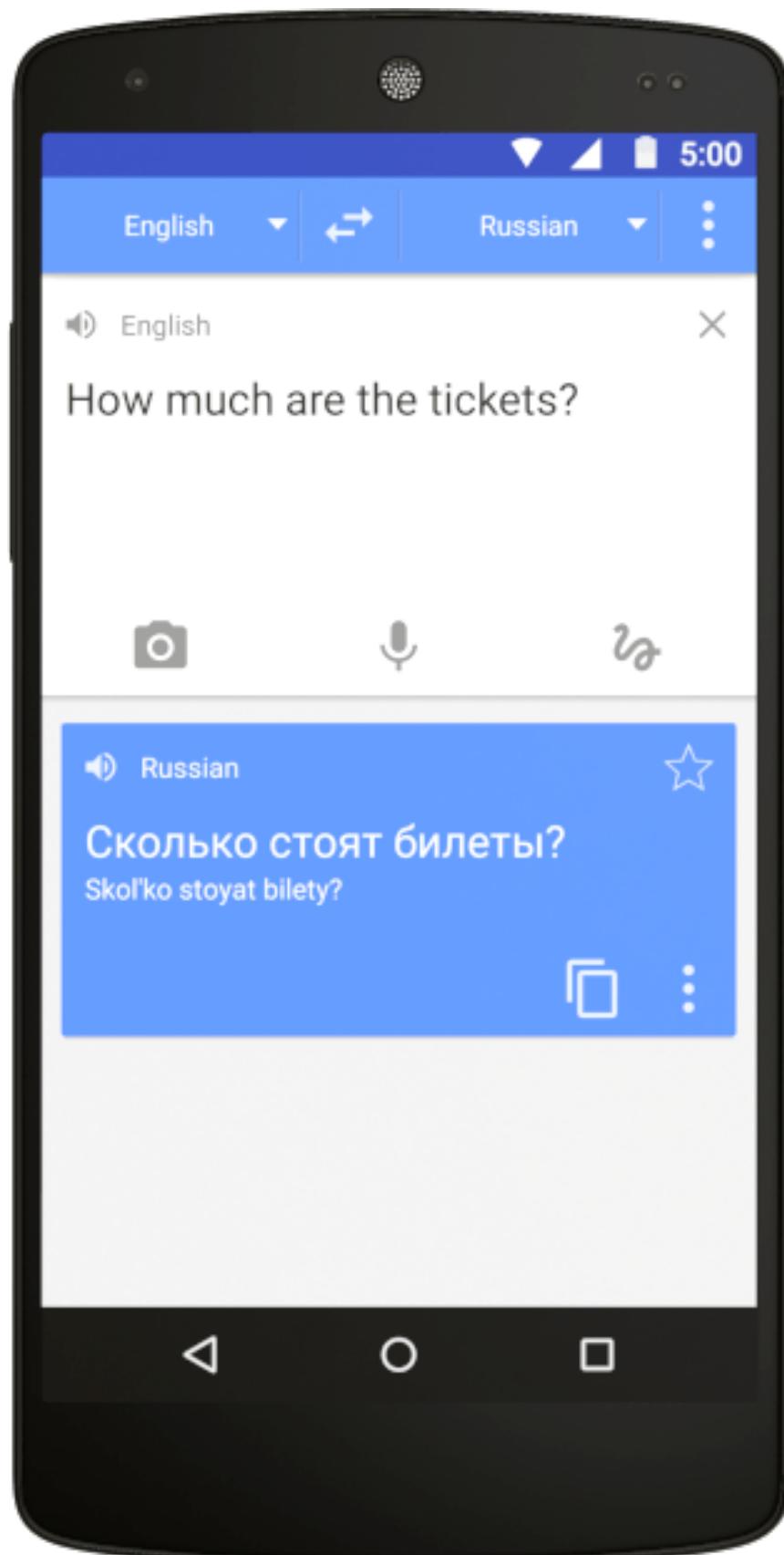


Voice to text

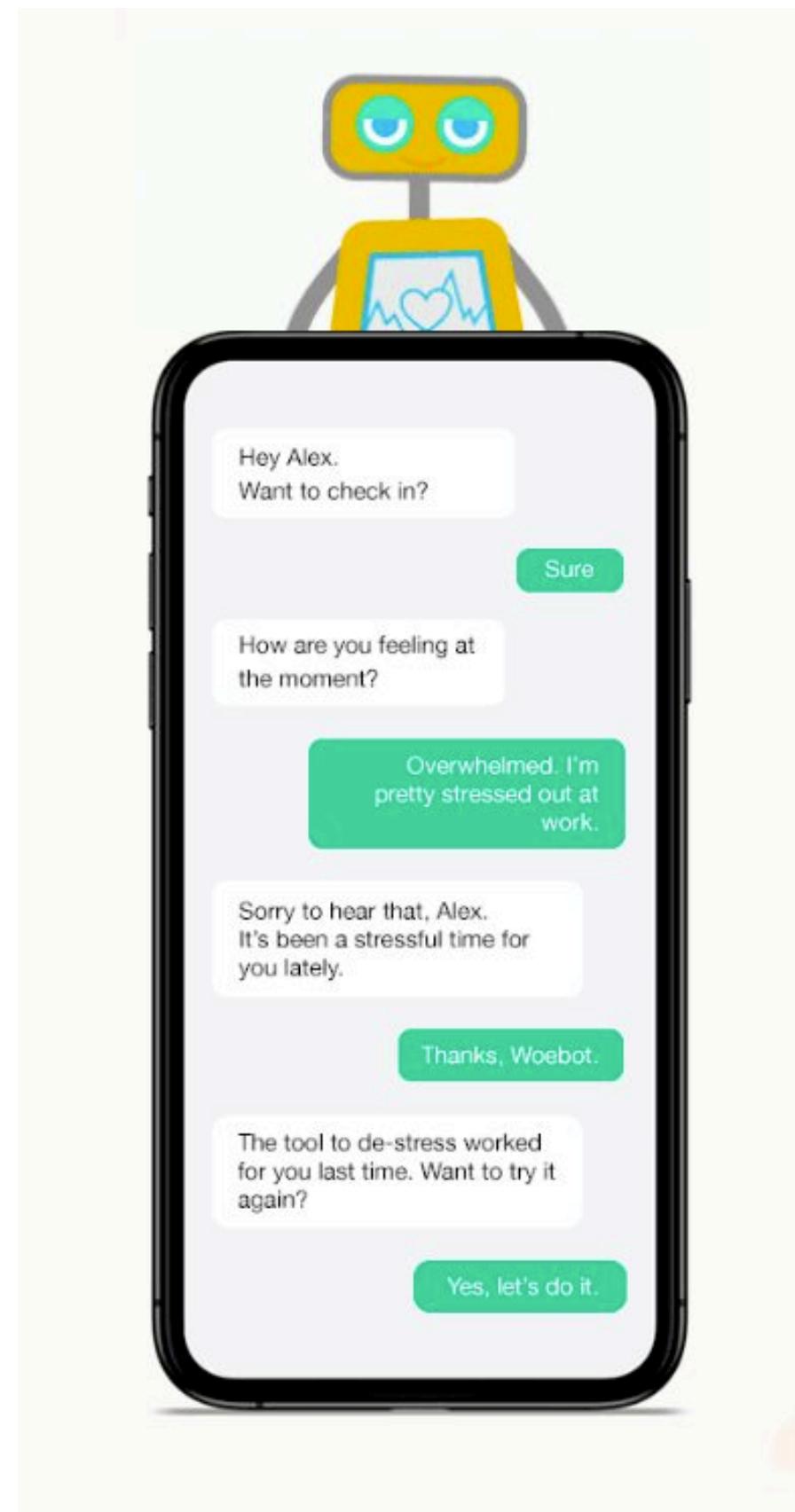


Applications of natural language processing

Machine translation



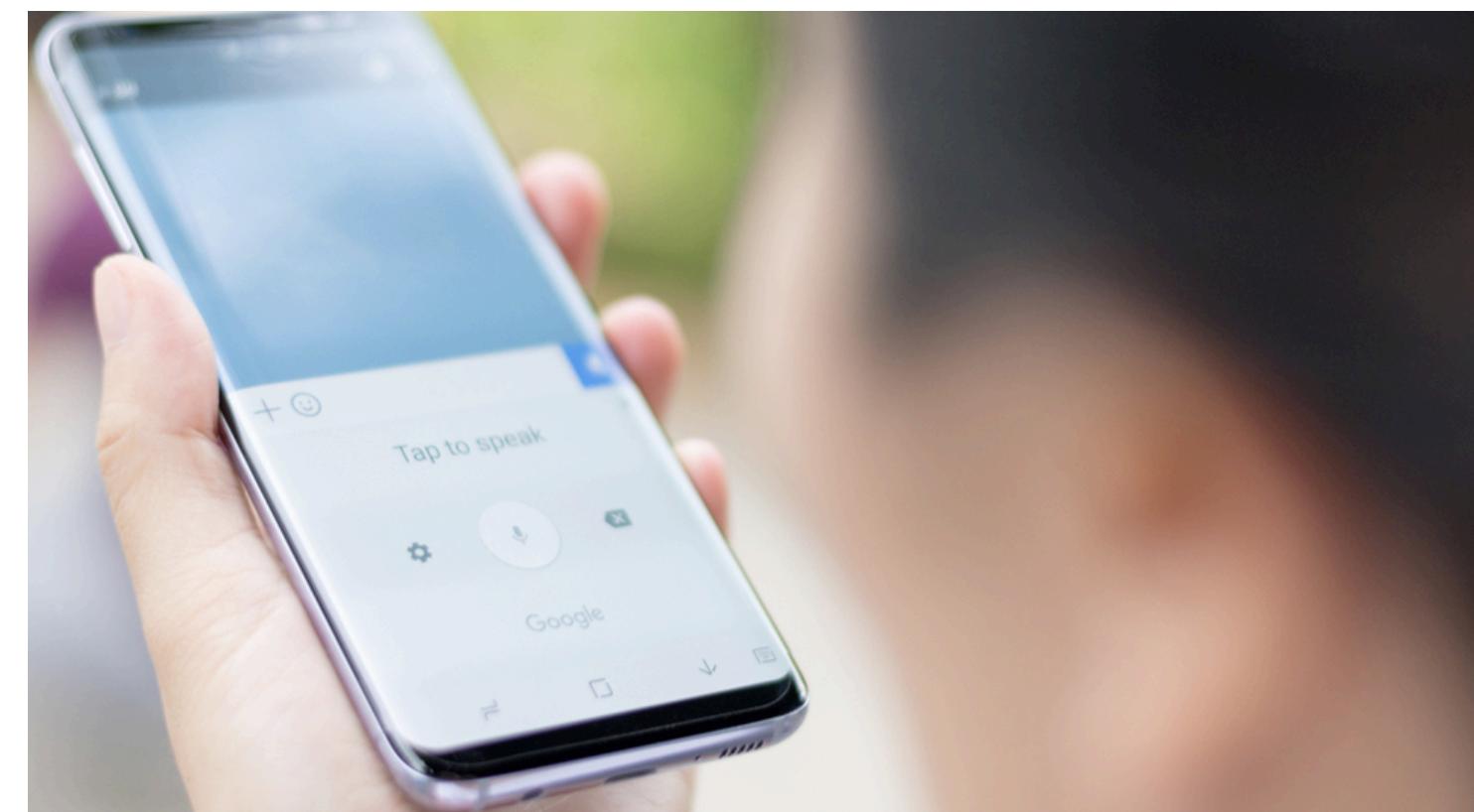
Chatbots



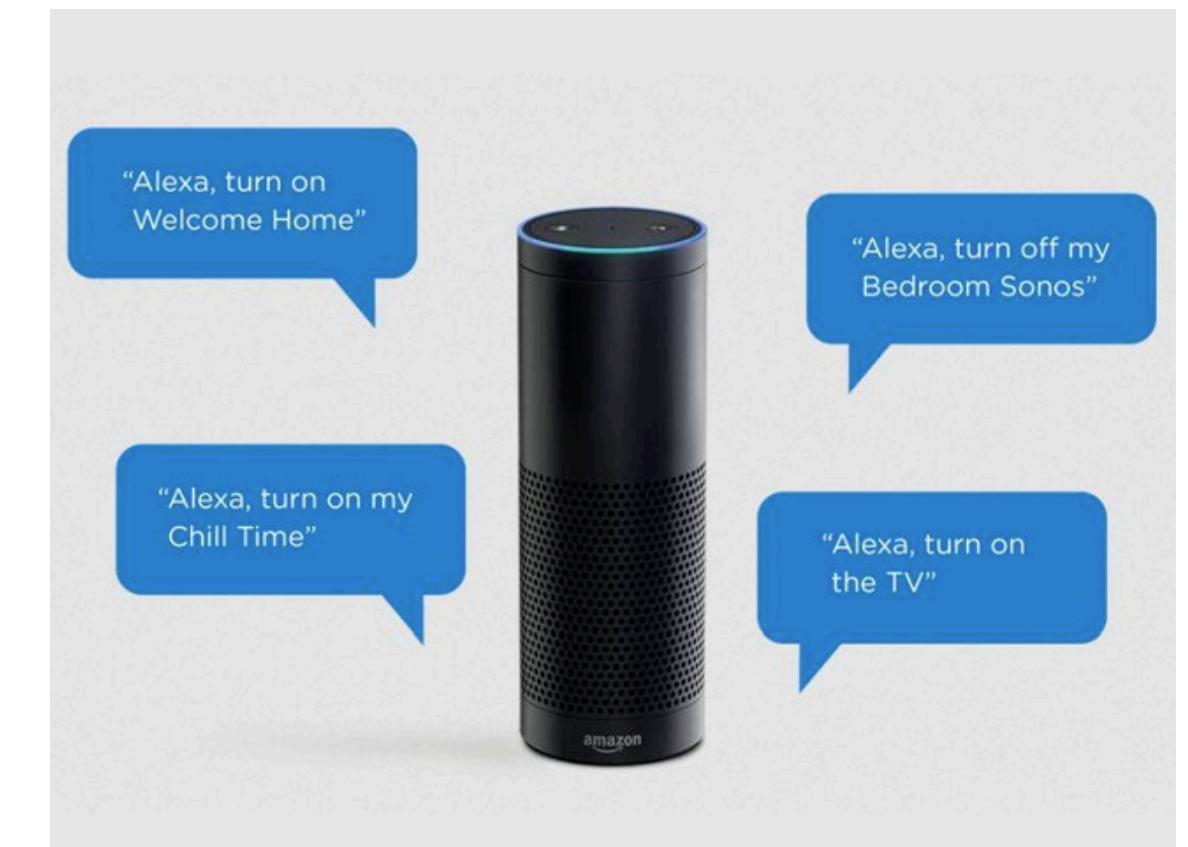
Spam filtering



Voice to text

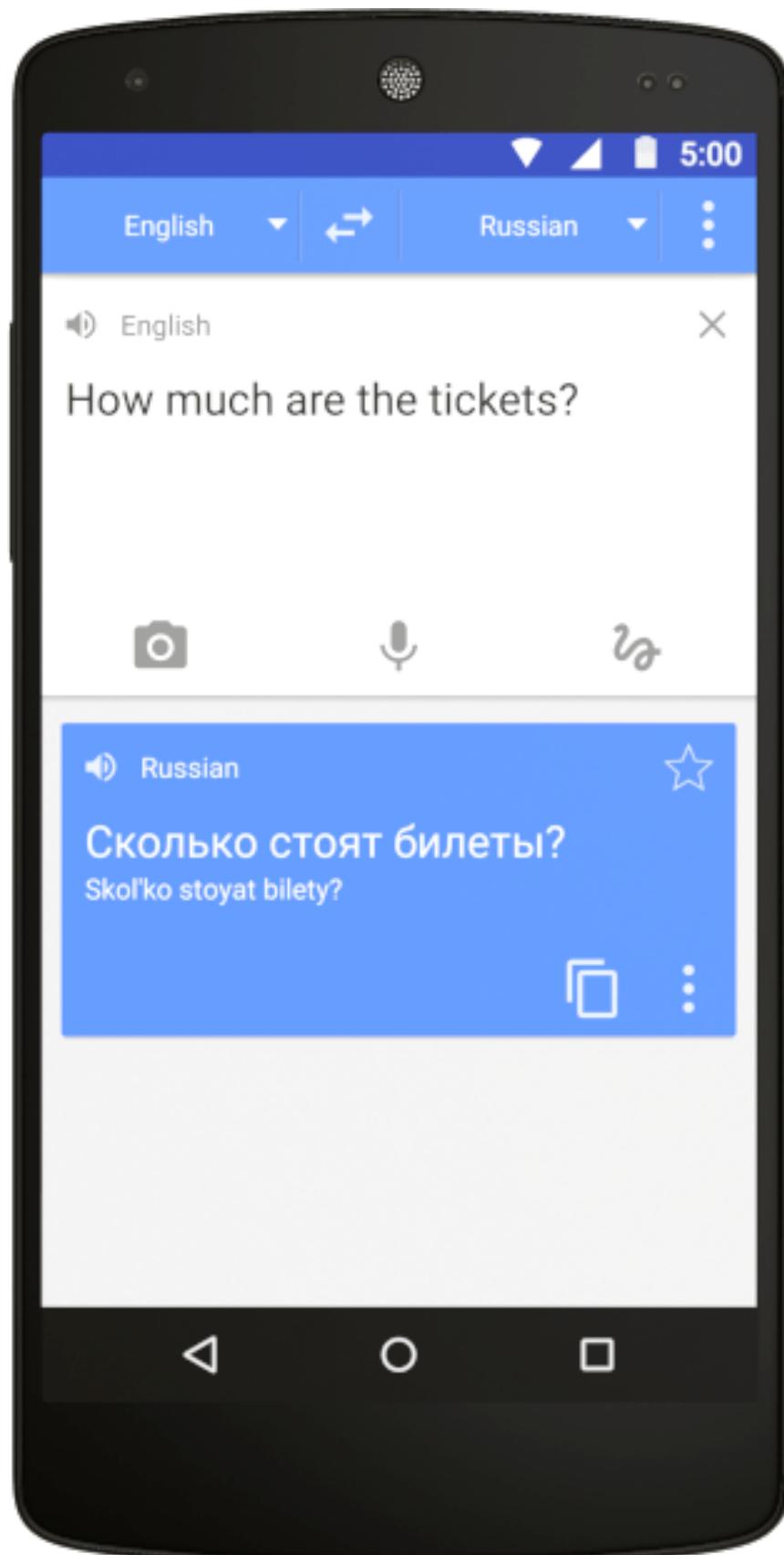


Personal assistant

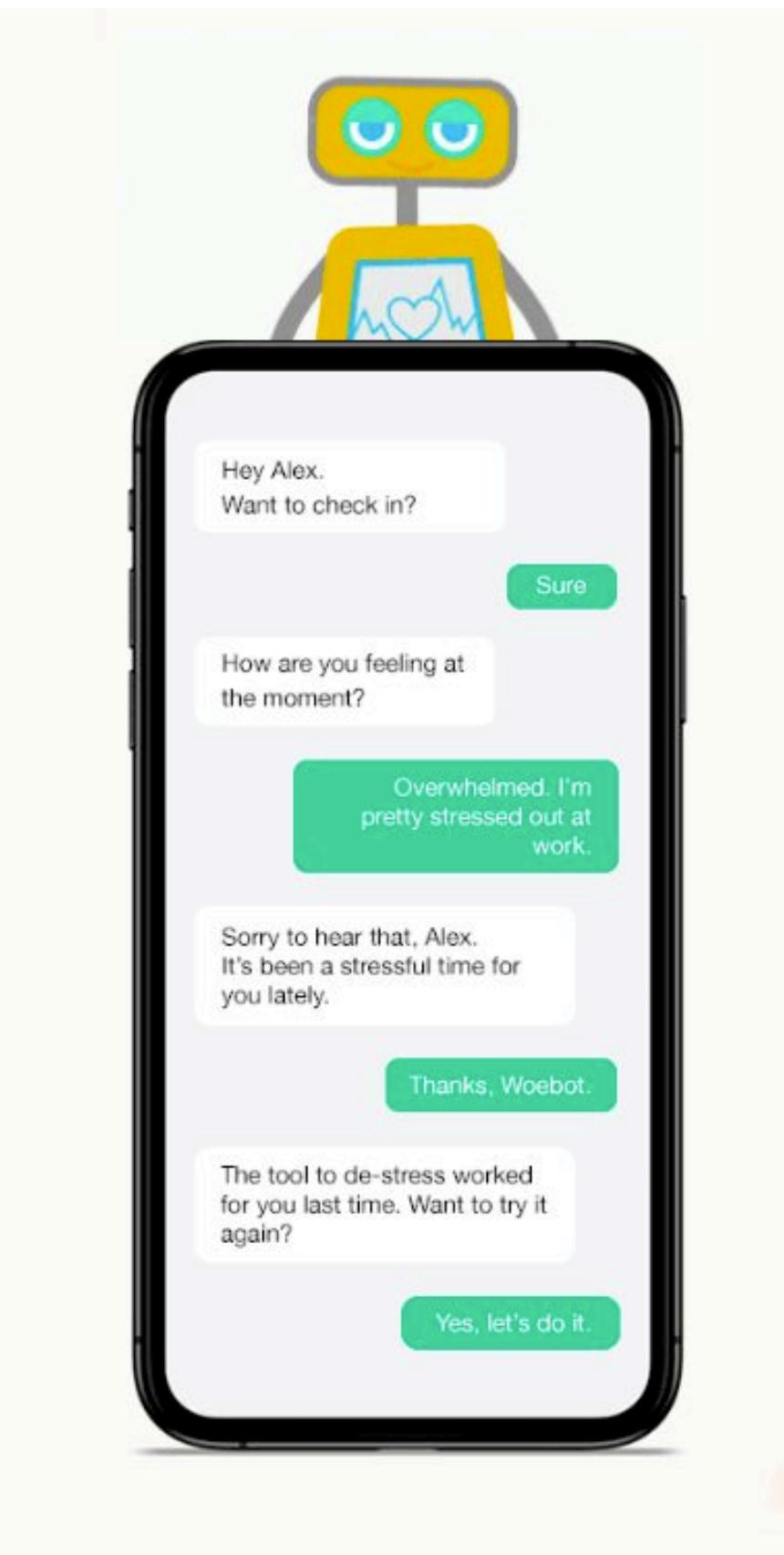


Applications of natural language processing

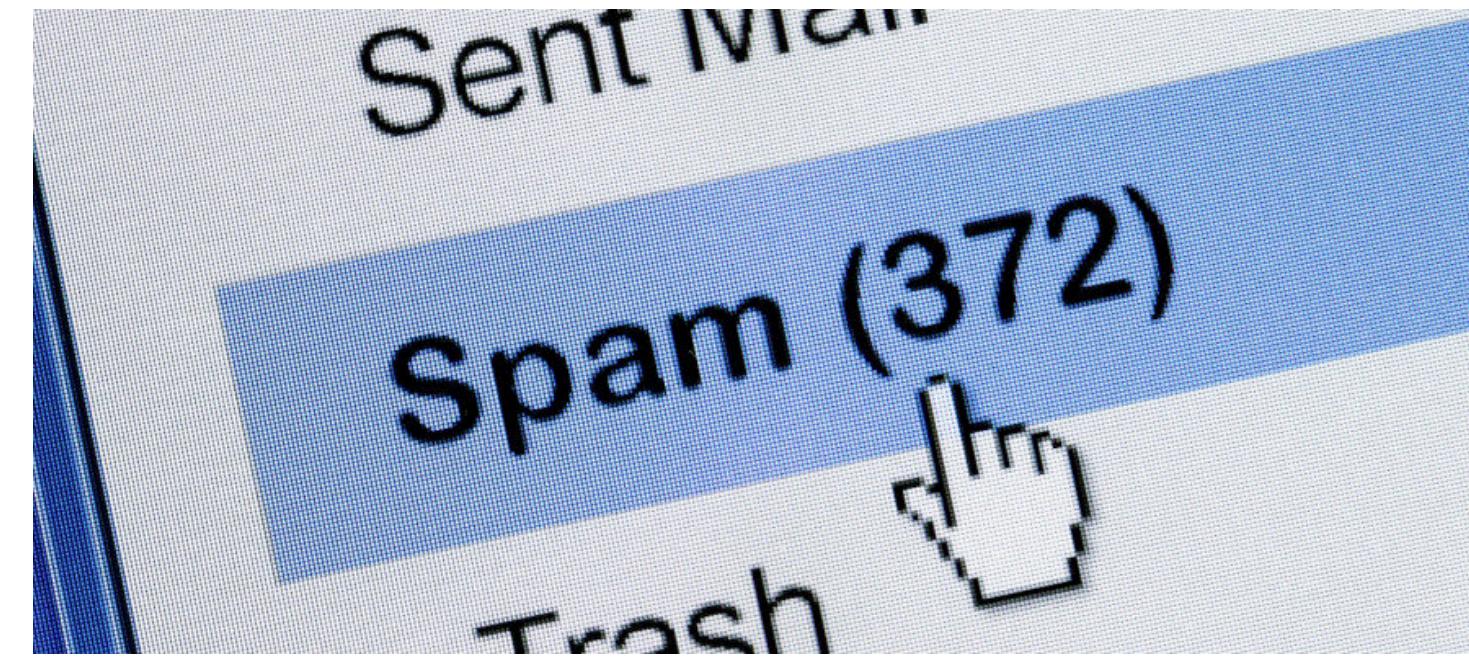
Machine translation



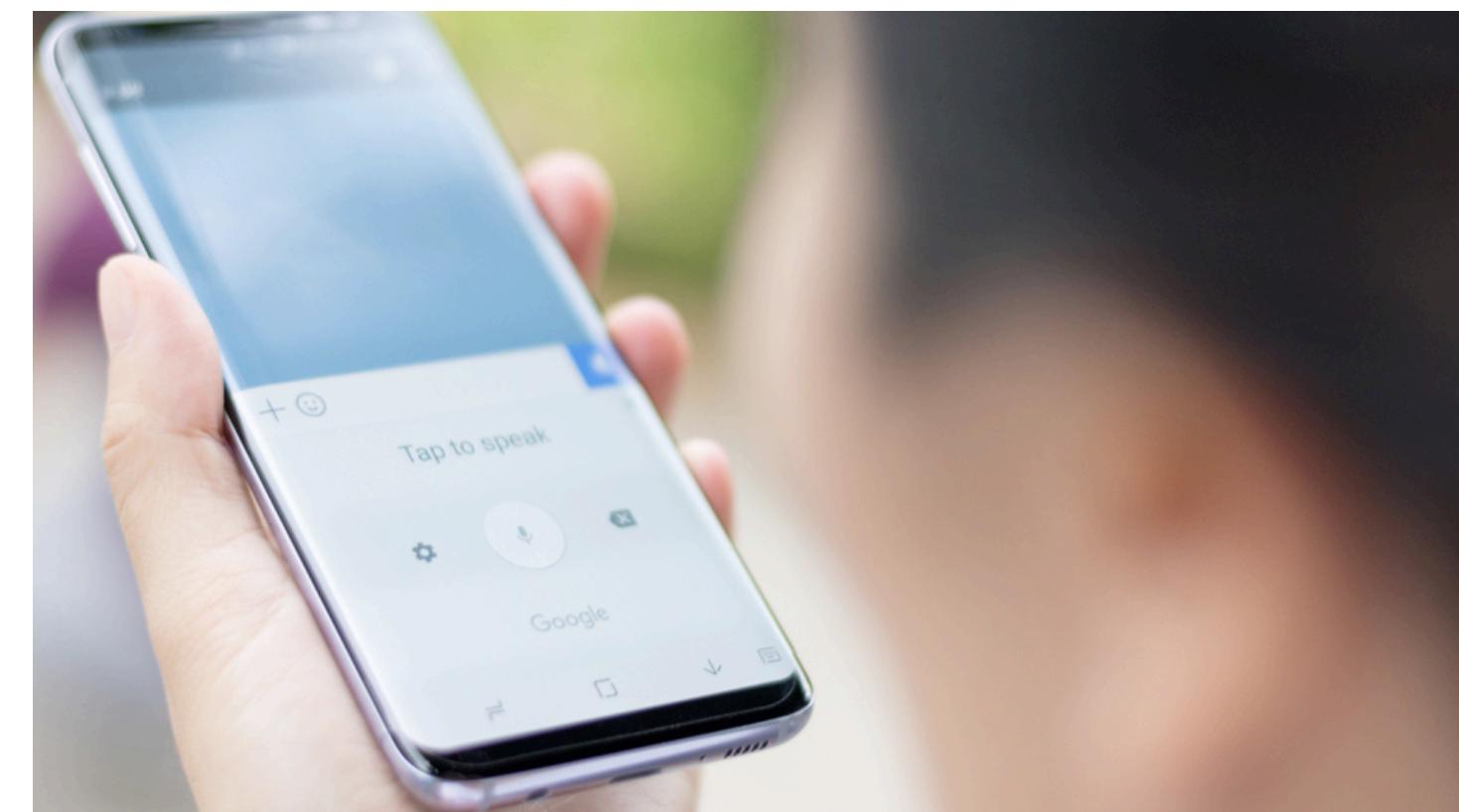
Chatbots



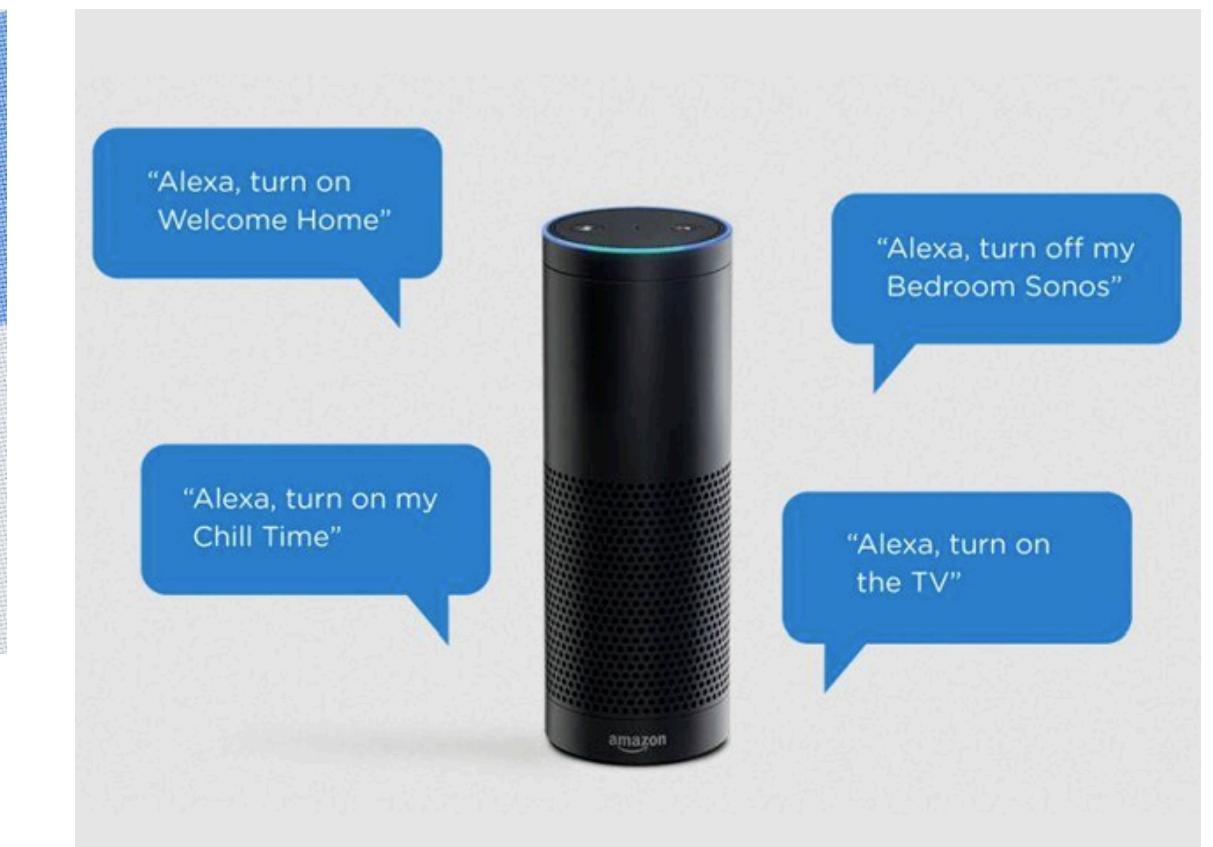
Spam filtering



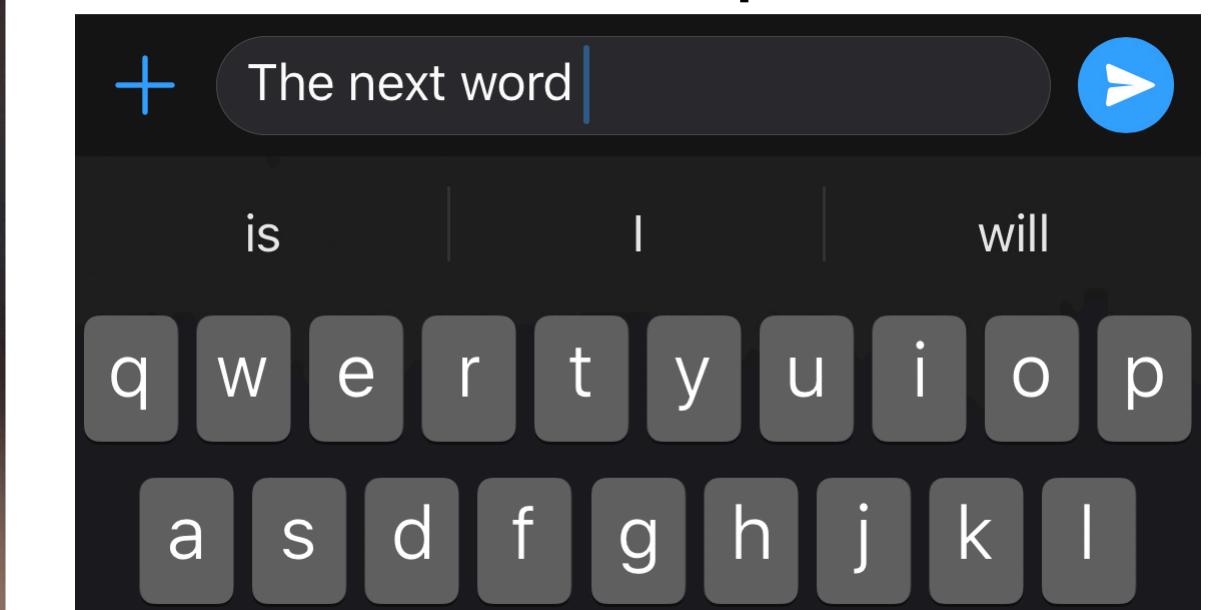
Voice to text



Personal assistant



Auto-complete



Word vectors

The basic units of images are pixels, which automatically have numeric values associated with them.

On the other hand, the basic units of language are words, which do not come with a pre-packaged vector representation.

Word vectors

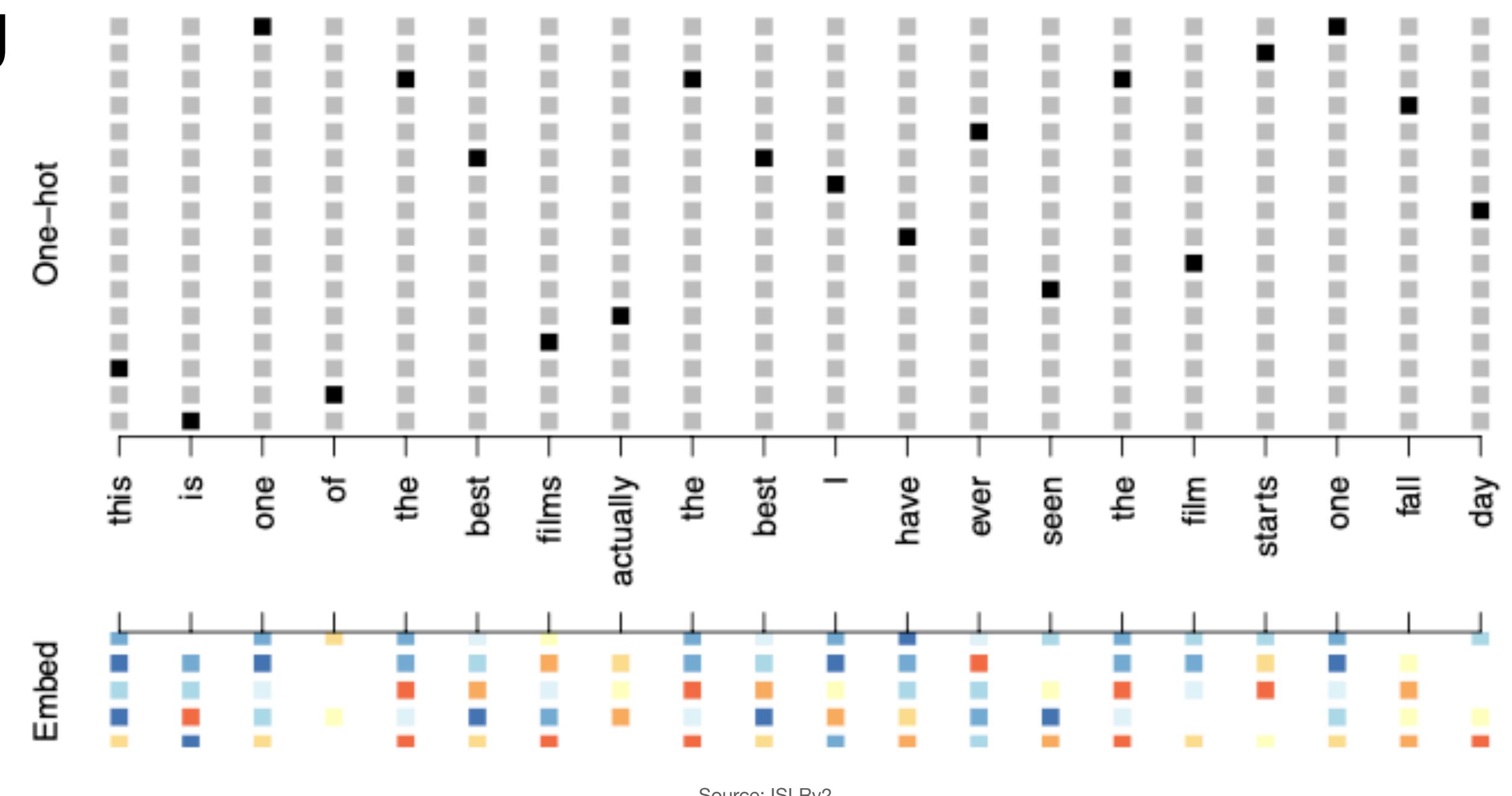
The basic units of images are pixels, which automatically have numeric values associated with them.

On the other hand, the basic units of language are words, which do not come with a pre-packaged vector representation.

A common first step in deep learning for NLP is to encode each word using a **word vector**, e.g.

$$v_{\text{man}} = (0.2, 1.6, 2.5, \dots, 4.1)$$

Word vectors are useful if they capture the meaning of the words they represent.



Constructing word vectors

Example: word2vec

Idea: If two words w_1 and w_2 near each other often (e.g. within 5 words of each other), then they should have similar word vectors v_1 and v_2 .

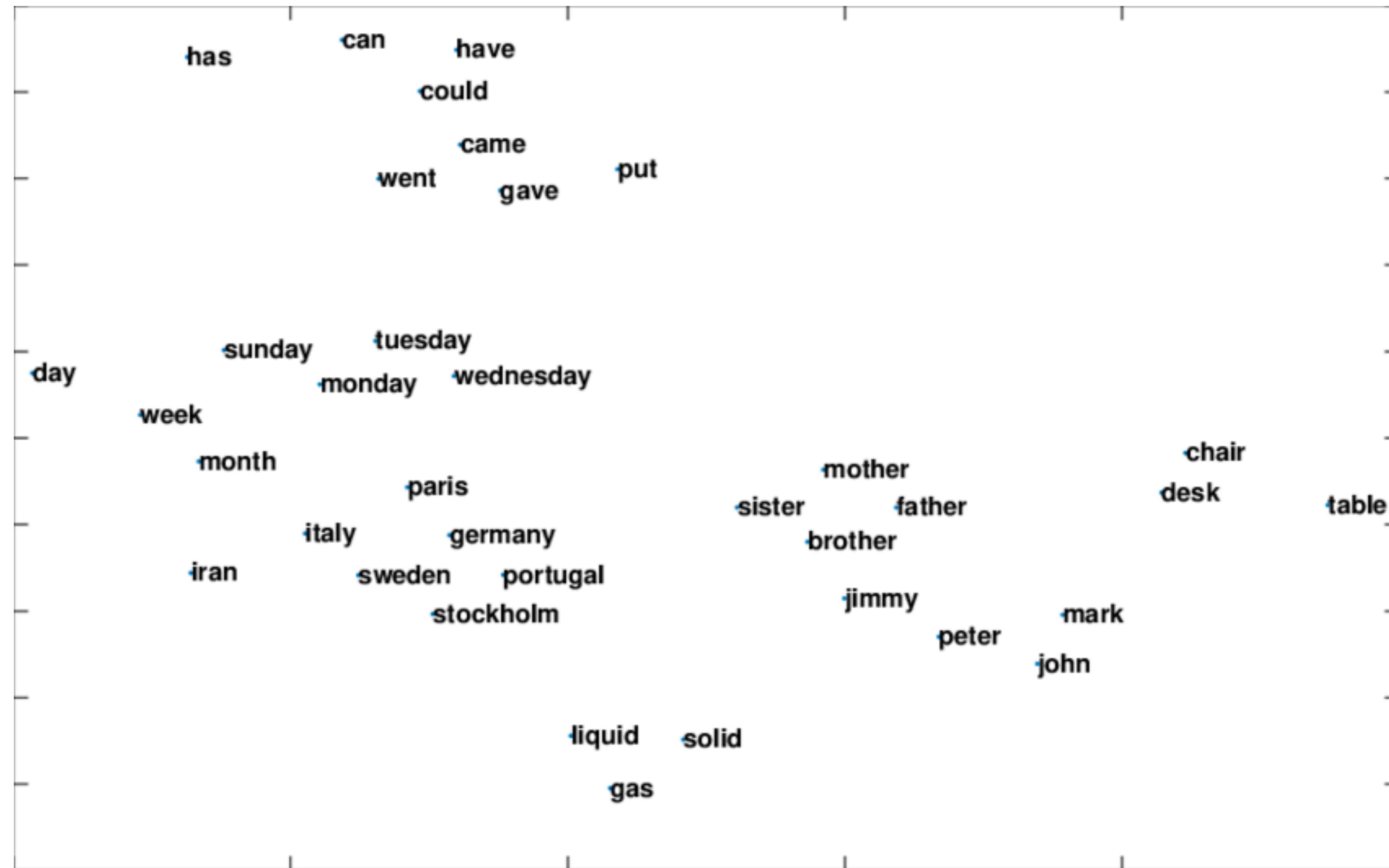
Suppose for a word w_1 , we model the relative frequencies of nearby words w_2 as

$$\mathbb{P}[w_2 | w_1] = \frac{\exp(v_1^T v_2)}{\sum_{\text{all word vectors } v} \exp(v_1^T v)}.$$

Can then train the word vectors v_1, v_2, \dots based on a text corpus by maximizing the likelihood of all the word co-occurrence probabilities.

Word vectors need to be trained only once (for each language), and can be reused thereafter. For example, word2vec vectors for a few dozen languages can be downloaded at <http://vectors.nlpl.eu/repository/>.

Word vectors capture semantic relationships



Case study 1: Named entity classification

Task: Classify the word in the middle of a window as either a person, location, organization, or other, e.g.

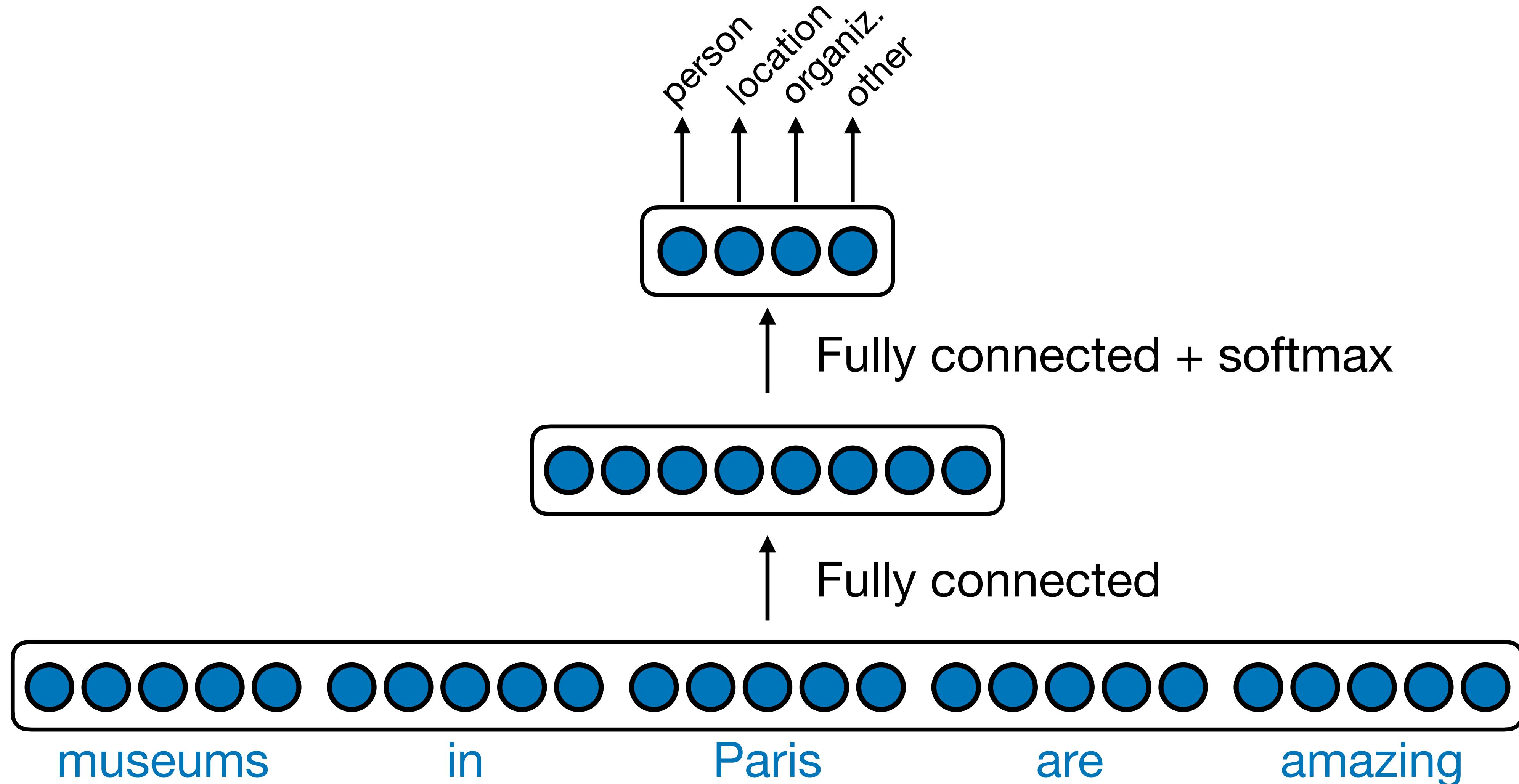
“...museums in **Paris** are amazing...”

In this context, does Paris refer to Paris Hilton (person) or Paris, France (location)?

This is a kind of **word sense disambiguation**.

Case study 1: Named entity classification

Using word vectors, can create a fully-connected neural network:



Fine-tuning word vectors

The word vectors, initialized to their pre-trained values, can either be frozen or fine-tuned during backpropagation. Which is better?

Advantages of freezing word vectors:

- Faster training and fewer parameters to worry about
- Recommended for small training data sets

Advantages of fine-tuning word vectors:

- Can give better predictions if there is enough data
- Recommended for large training data sets

From fixed length to variable length

While images are usually a fixed size (e.g. 128x128 pixels), natural language chunks like sentences can have variable length. Using a fixed window size forces you to artificially chop your input. Outputs might also have variable length.

Input length	Output length	Example
Fixed	Fixed	Image classification
Fixed	Variable	Image captioning
Variable	Fixed	Sentiment analysis
Variable	Variable	Machine translation

We need a neural network architecture to handle variable-length inputs and outputs.

Recurrent neural networks (RNNs)

How to have variable number of inputs/outputs with a fixed number of parameters?

Idea: Use a recurrent structure where the same weights are used to update the model at each time step.

Hidden state h_t captures available info at time t .

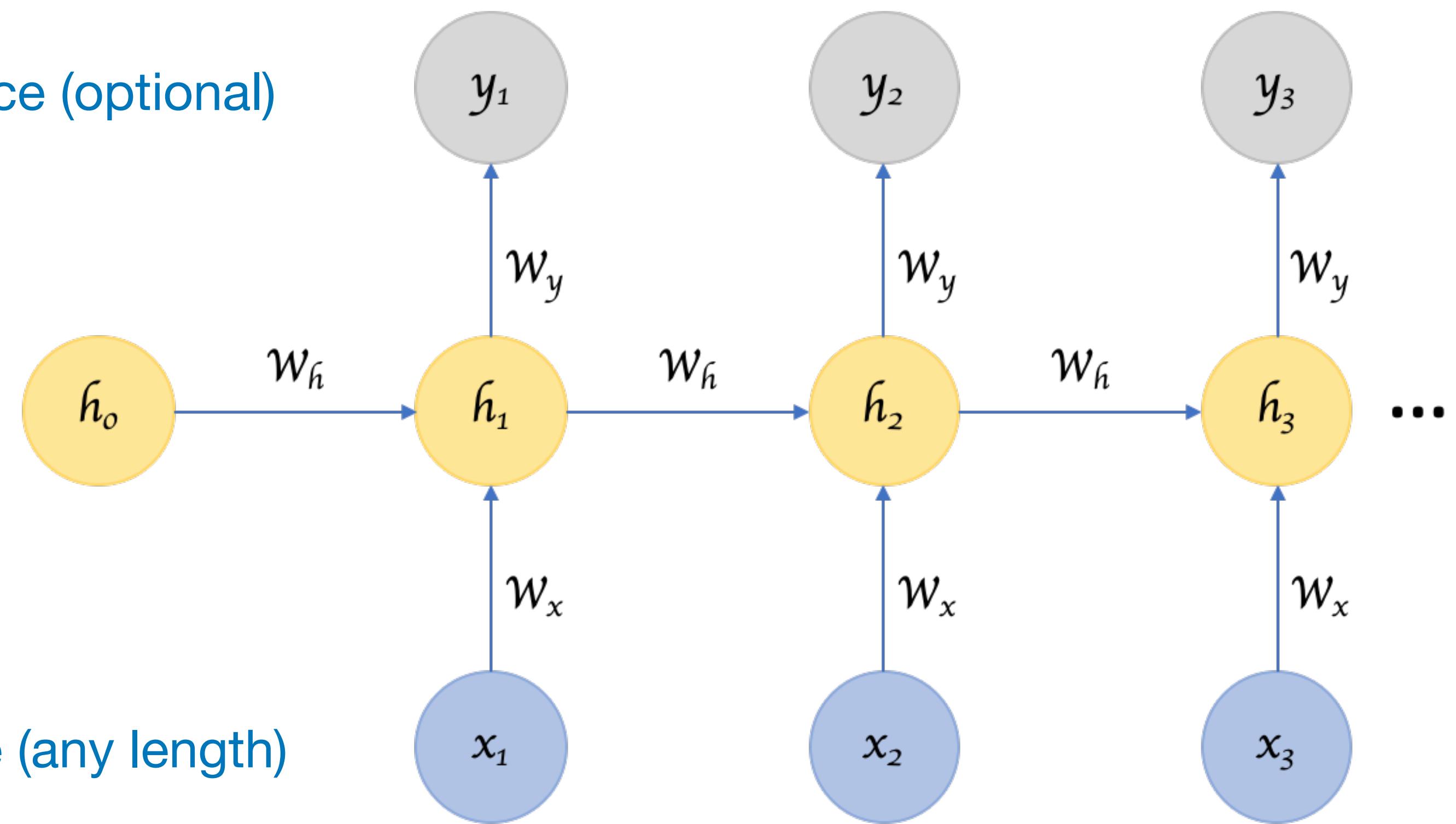
Hidden state evolves based on previous hidden state and new input x_t :

$$h_t = \sigma(W_h h_{t-1} + W_x x_t).$$

Output sequence (optional)

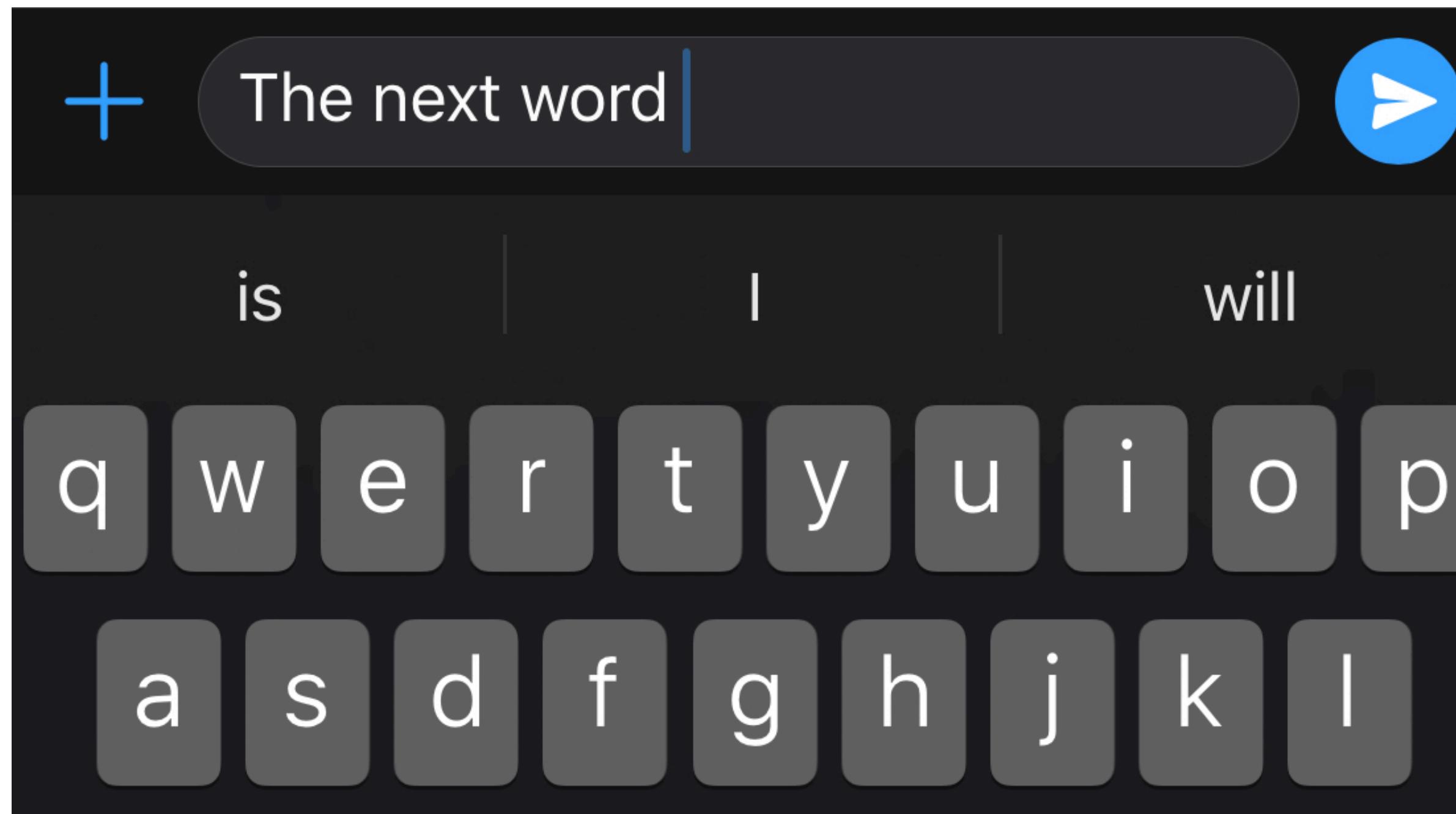
Hidden states

Input sequence (any length)



Case study 2: Language modeling

Given a sequence of words, calculate probabilities of what the next word will be.



Google

is upenn an|

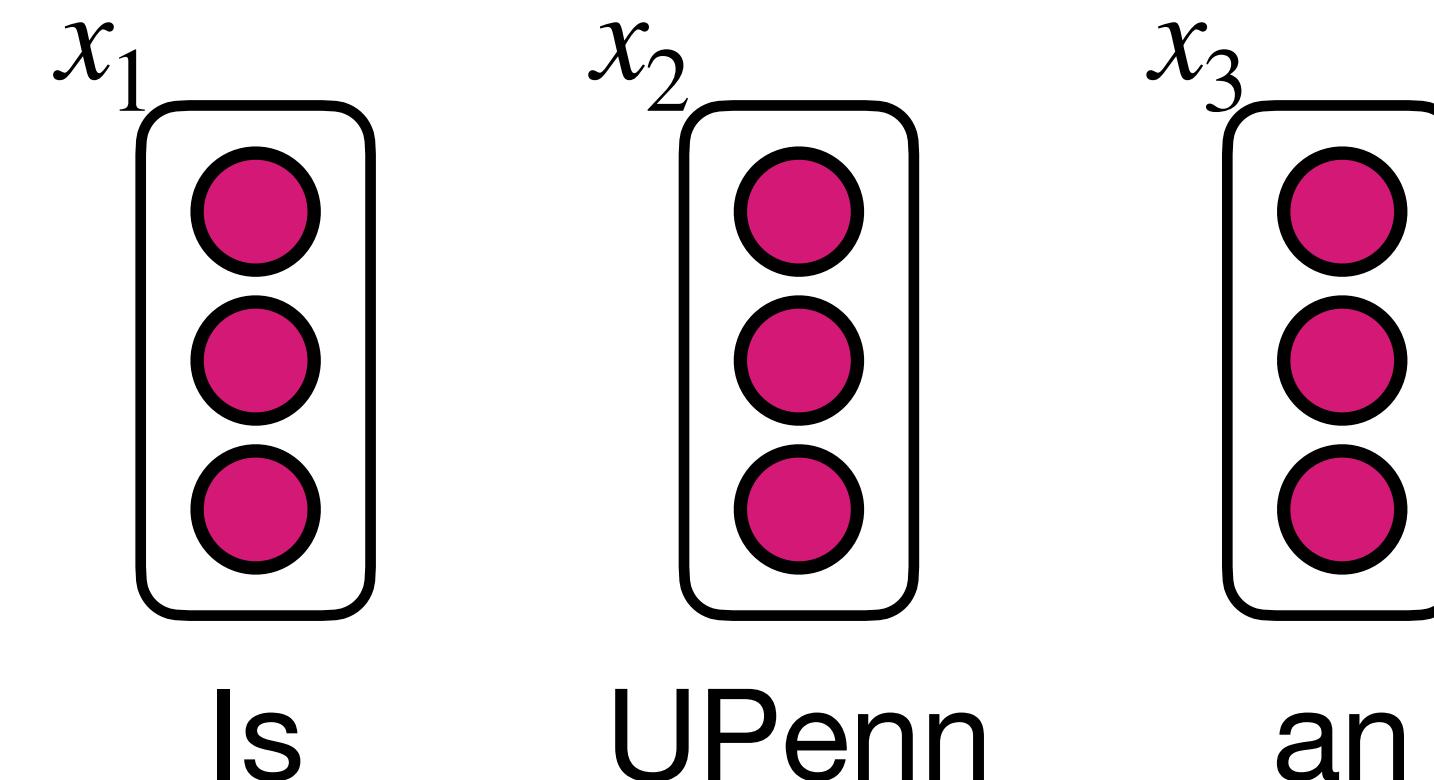
- is upenn an ivy
- is upenn an ivy league school
- is upenn and penn state the same thing
- is upenn an urban campus
- is upenn and ivy league
- is upenn an open campus
- is upenn and penn state different
- is upenn and wharton the same
- is upenn an elite school
- is upenn an accredited school

Google Search I'm Feeling Lucky

Report inappropriate predictions

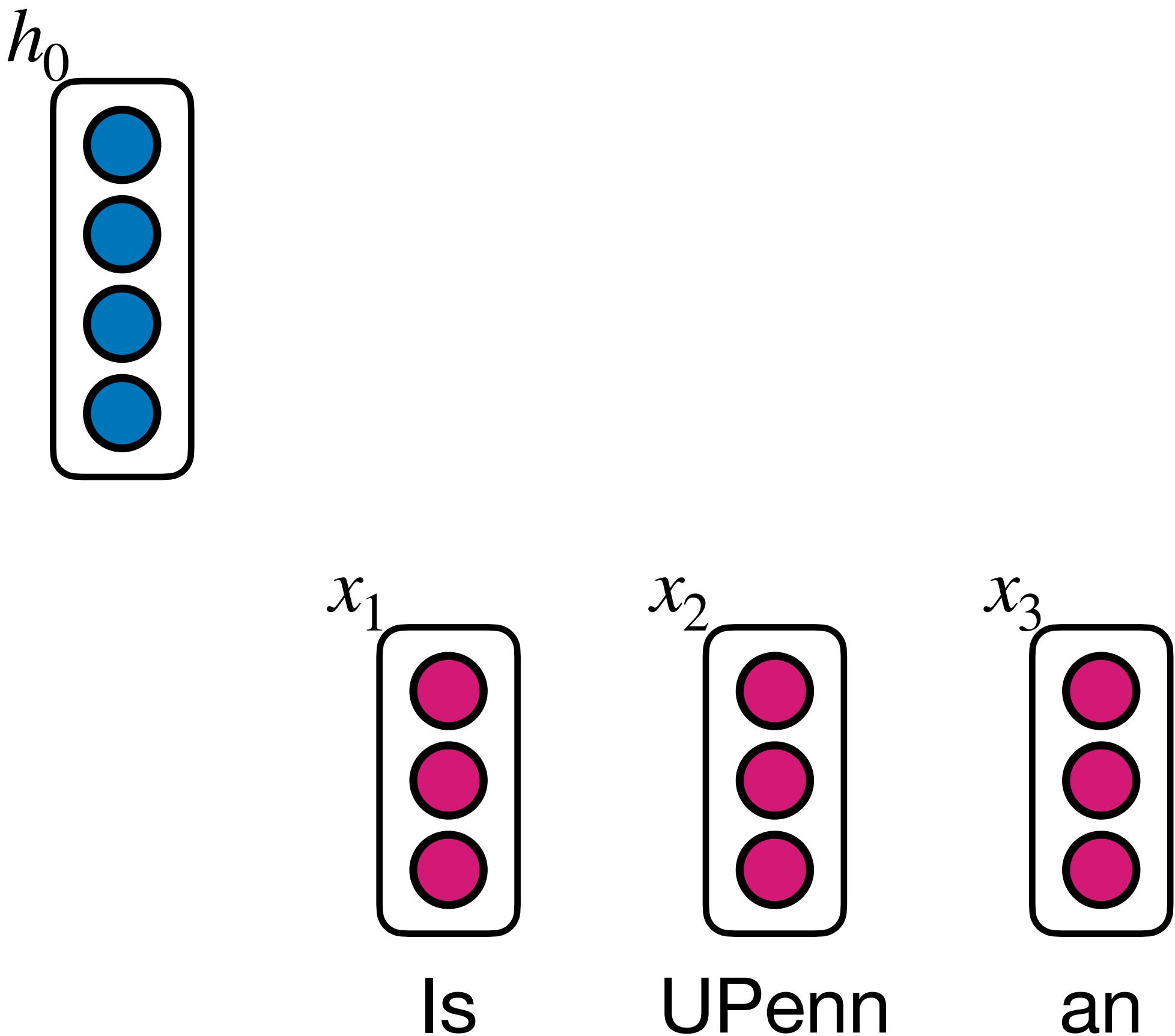
RNN for language modeling

- Initialize h_0 to zero vector.
- Keep updating hidden states using $h_t = \sigma(W_h h_{t-1} + W_x x_t)$.
- Once you get to the end, pass the last hidden state through a fully connected layer plus softmax to get probabilities for next word.



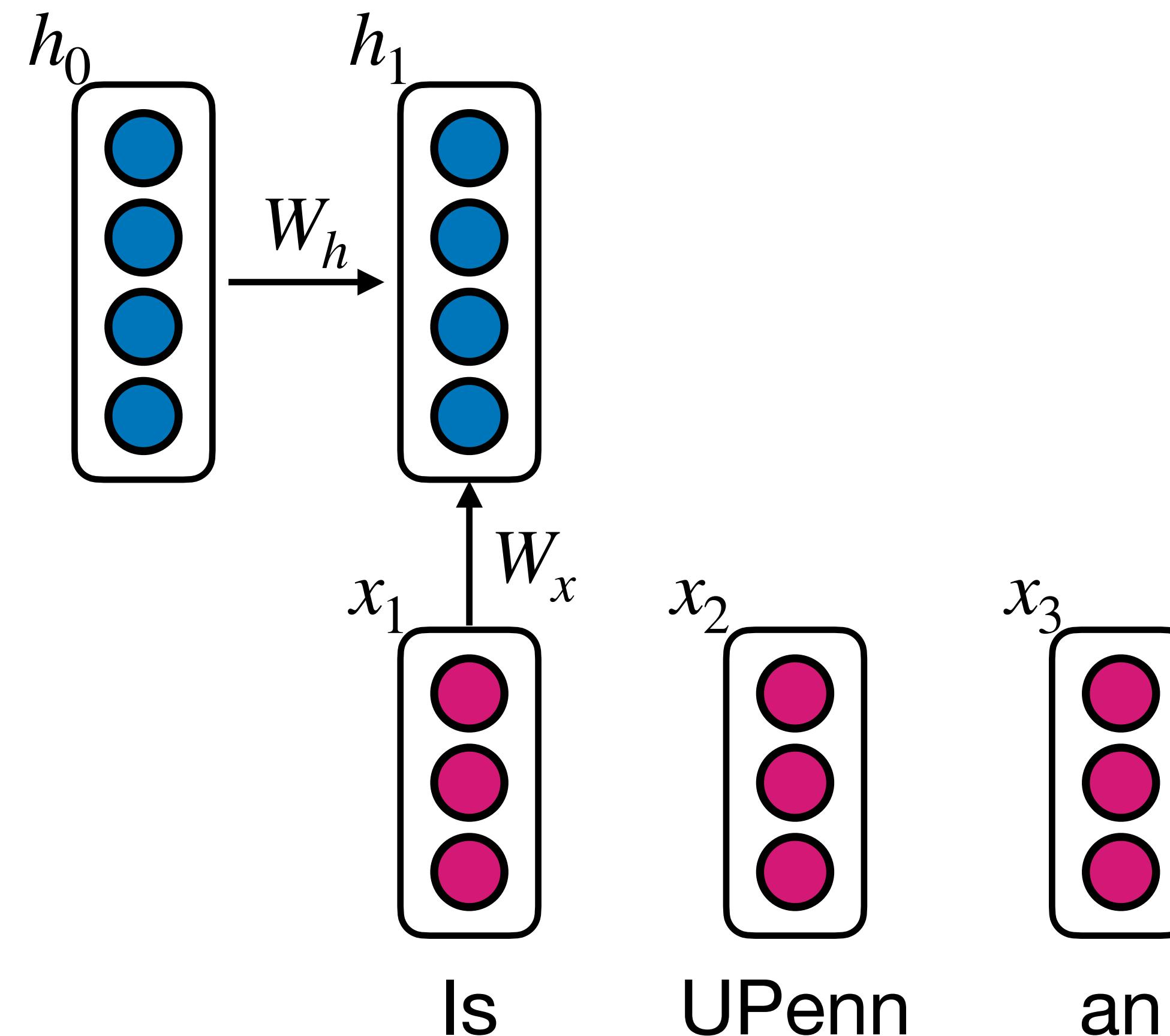
RNN for language modeling

- Initialize h_0 to zero vector.
- Keep updating hidden states using $h_t = \sigma(W_h h_{t-1} + W_x x_t)$.
- Once you get to the end, pass the last hidden state through a fully connected layer plus softmax to get probabilities for next word.



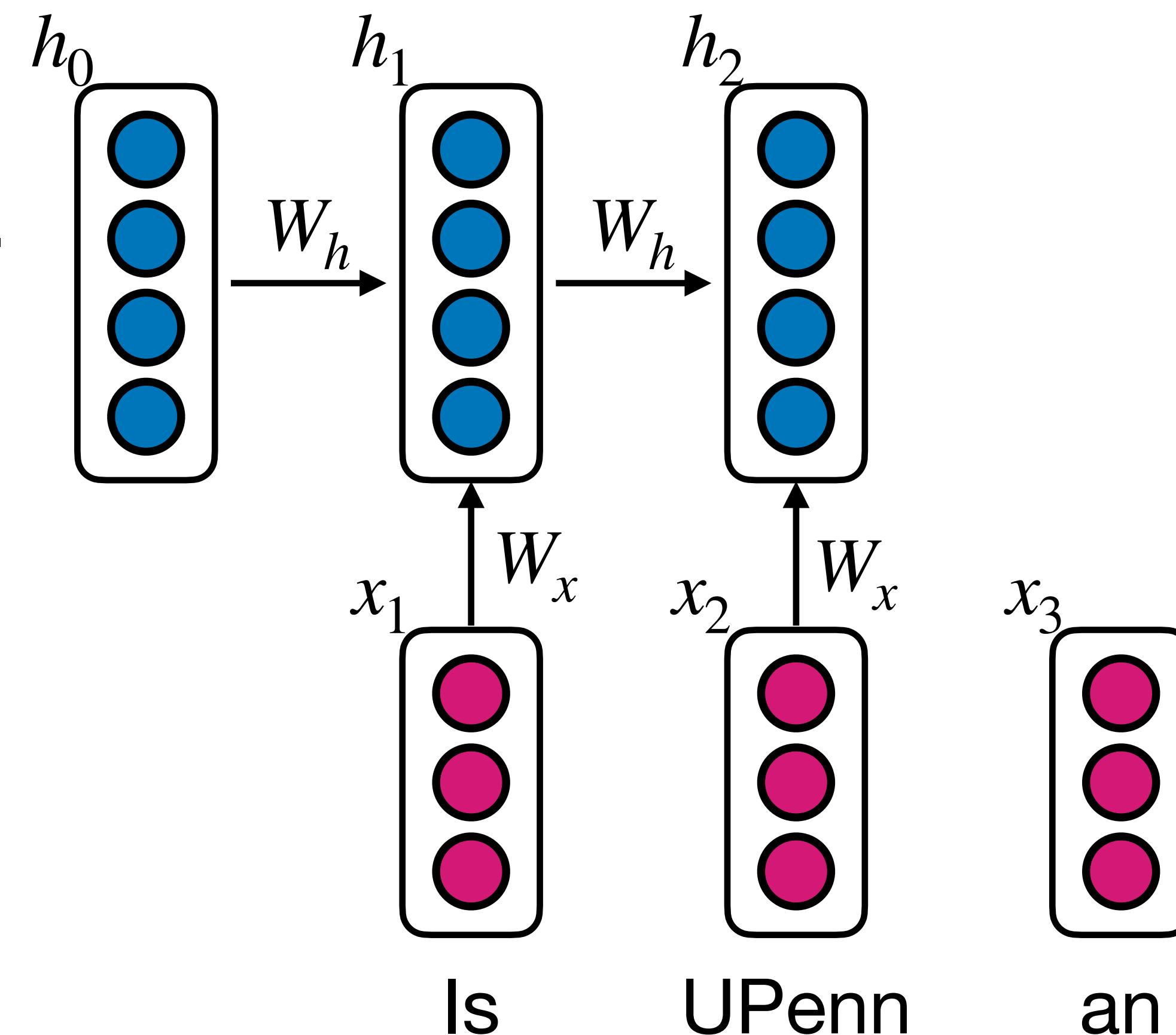
RNN for language modeling

- Initialize h_0 to zero vector.
- Keep updating hidden states using $h_t = \sigma(W_h h_{t-1} + W_x x_t)$.
- Once you get to the end, pass the last hidden state through a fully connected layer plus softmax to get probabilities for next word.



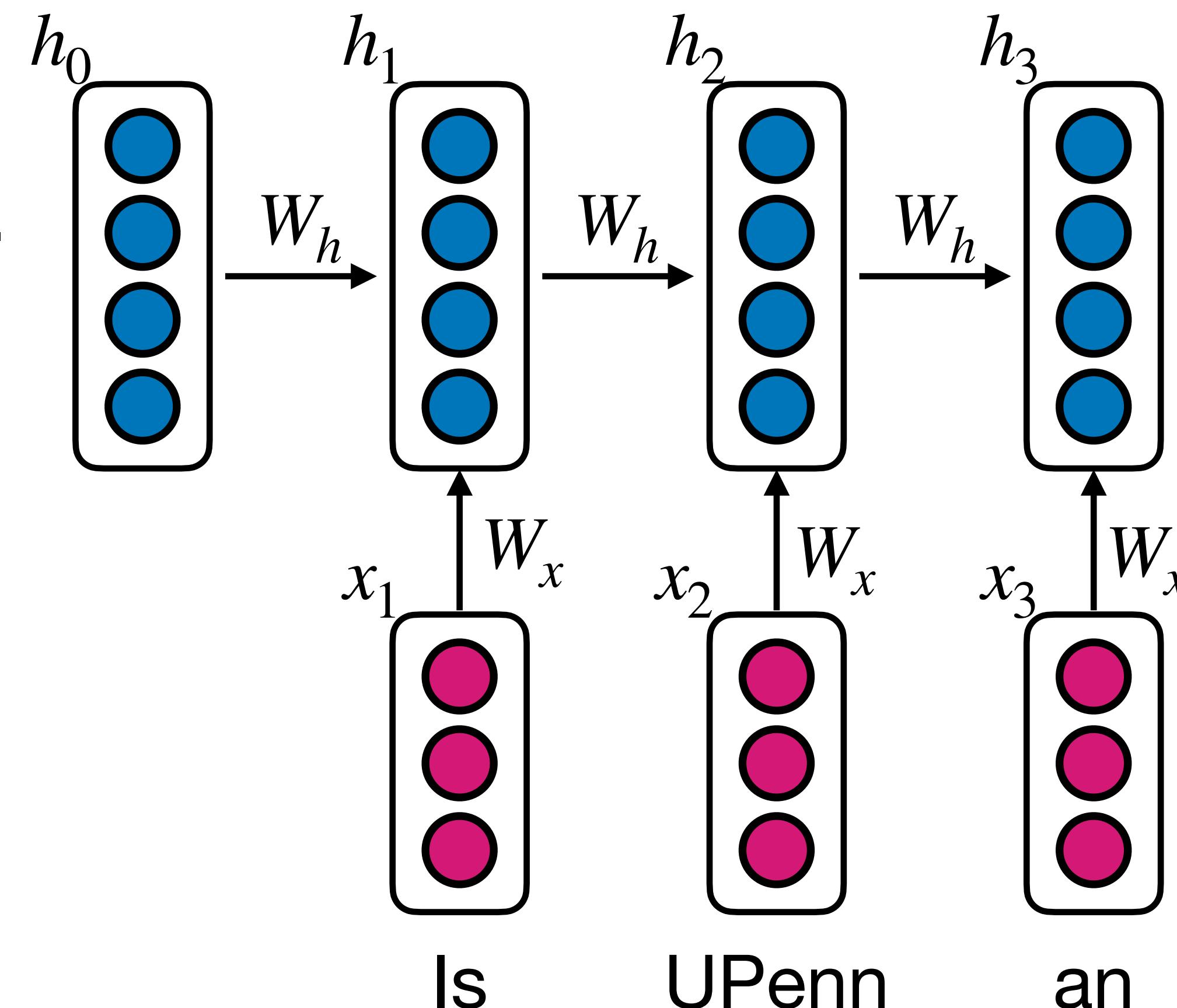
RNN for language modeling

- Initialize h_0 to zero vector.
- Keep updating hidden states using $h_t = \sigma(W_h h_{t-1} + W_x x_t)$.
- Once you get to the end, pass the last hidden state through a fully connected layer plus softmax to get probabilities for next word.



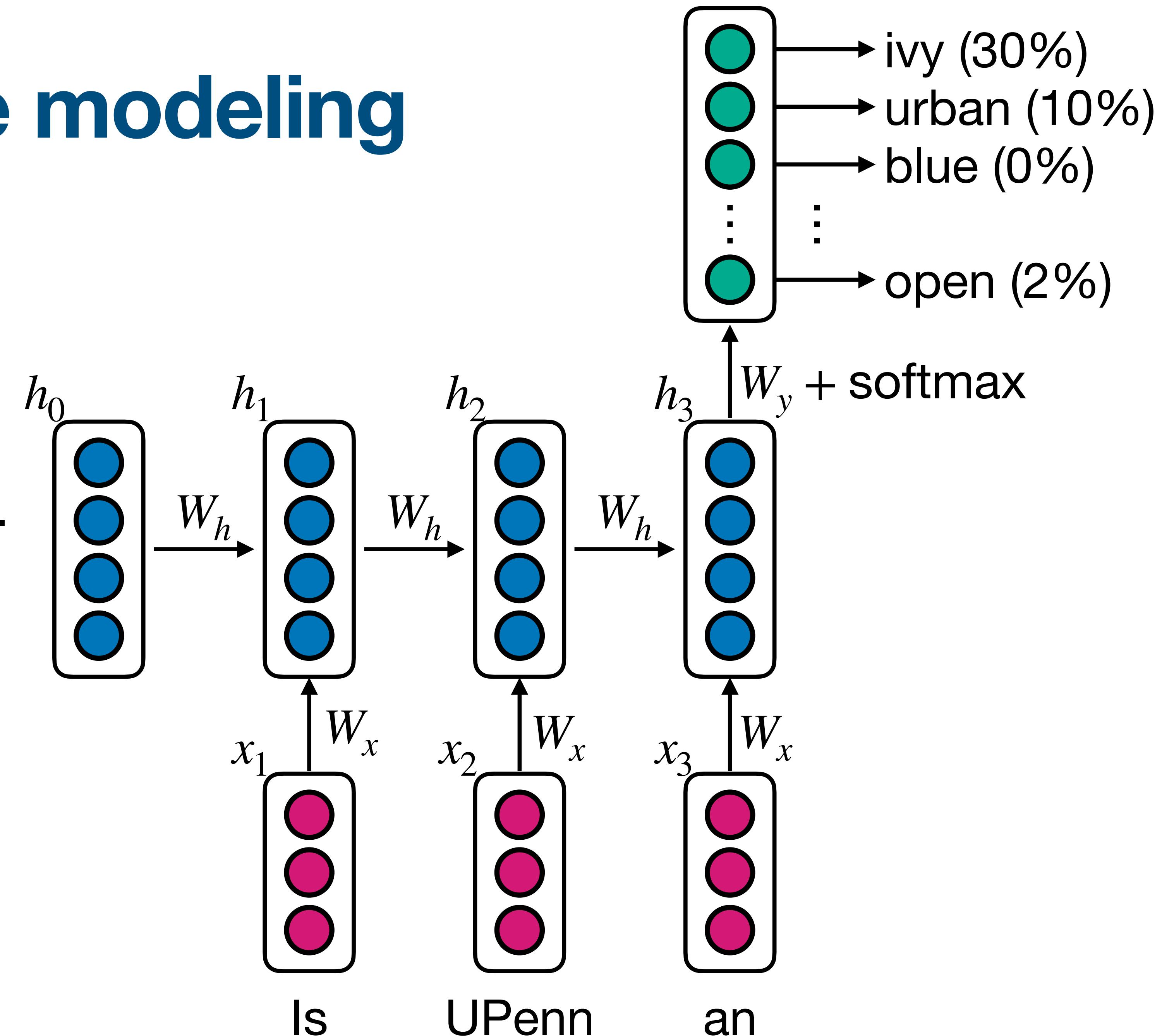
RNN for language modeling

- Initialize h_0 to zero vector.
- Keep updating hidden states using $h_t = \sigma(W_h h_{t-1} + W_x x_t)$.
- Once you get to the end, pass the last hidden state through a fully connected layer plus softmax to get probabilities for next word.



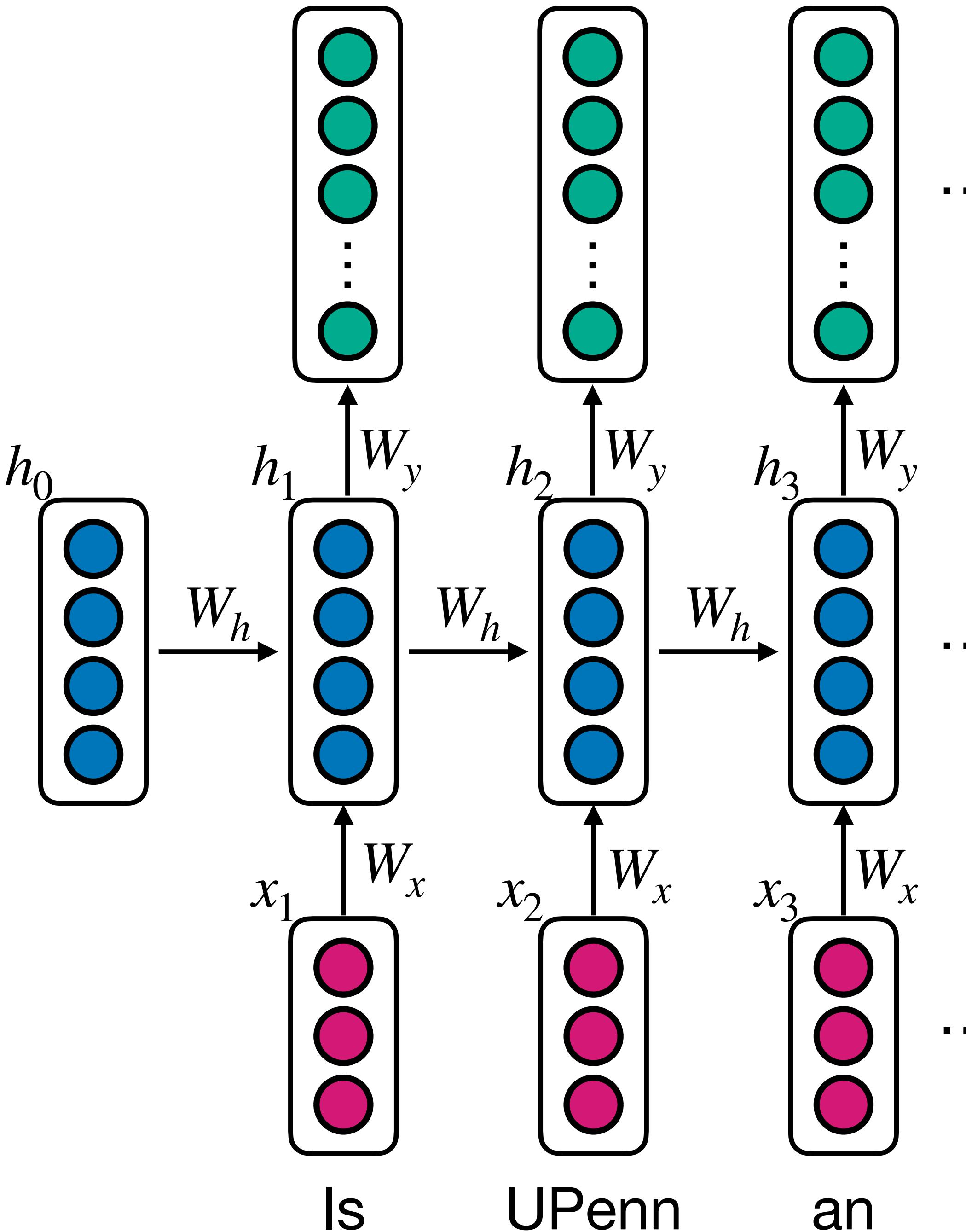
RNN for language modeling

- Initialize h_0 to zero vector.
- Keep updating hidden states using $h_t = \sigma(W_h h_{t-1} + W_x x_t)$.
- Once you get to the end, pass the last hidden state through a fully connected layer plus softmax to get probabilities for next word.



Training an RNN

- Get a big corpus of text, like Wikipedia.
- For any set of parameters, can predict each word based on all previous ones.
- Using the cross-entropy loss function averaged over all words, **backpropagate through time** to learn W_x , W_h , W_y .



Text generated from an RNN

Example: RNN trained on Barack Obama's speeches



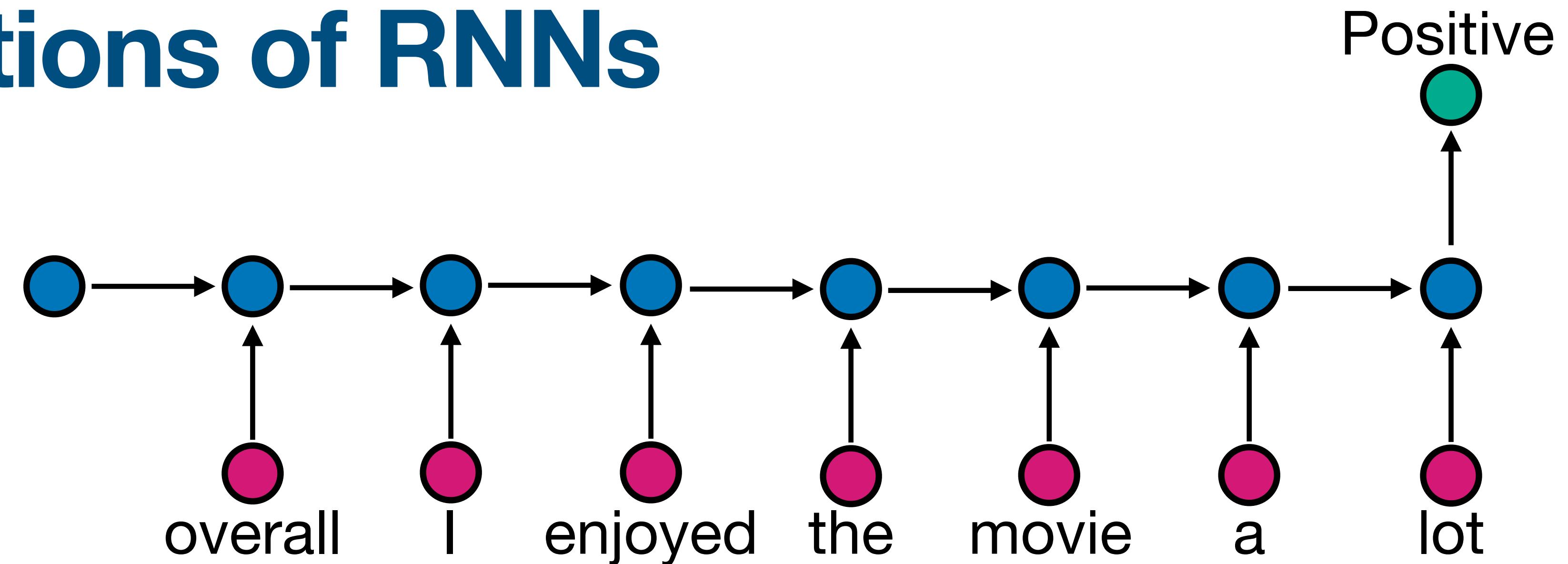
“Good afternoon. God bless you.

The United States will step up to the cost of a new challenges of the American people that will share the fact that we created the problem. They were attacked and so that they have to say that all the task of the final days of war that I will not be able to get this done. The promise of the men and women who were still going to take out the fact that the American people have fought to make sure that they have to be able to protect our part. It was a chance to stand together to completely look for the commitment to borrow from the American people. And the fact is the men and women in uniform and the millions of our country with the law system that we should be a strong stretch of the forces that we can afford to increase our spirit of the American people and the leadership of our country who are on the Internet of American lives.

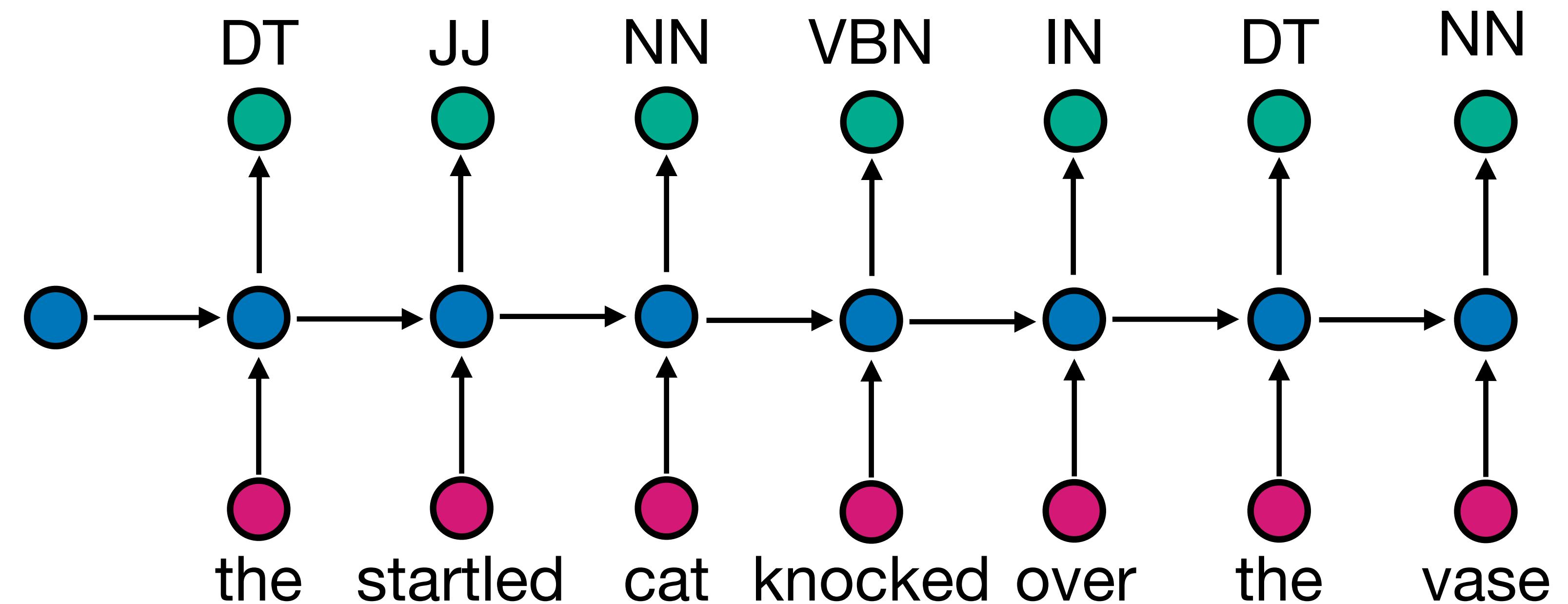
Thank you very much. God bless you, and God bless the United States of America.”

Other applications of RNNs

Sentiment analysis



Part-of-speech tagging



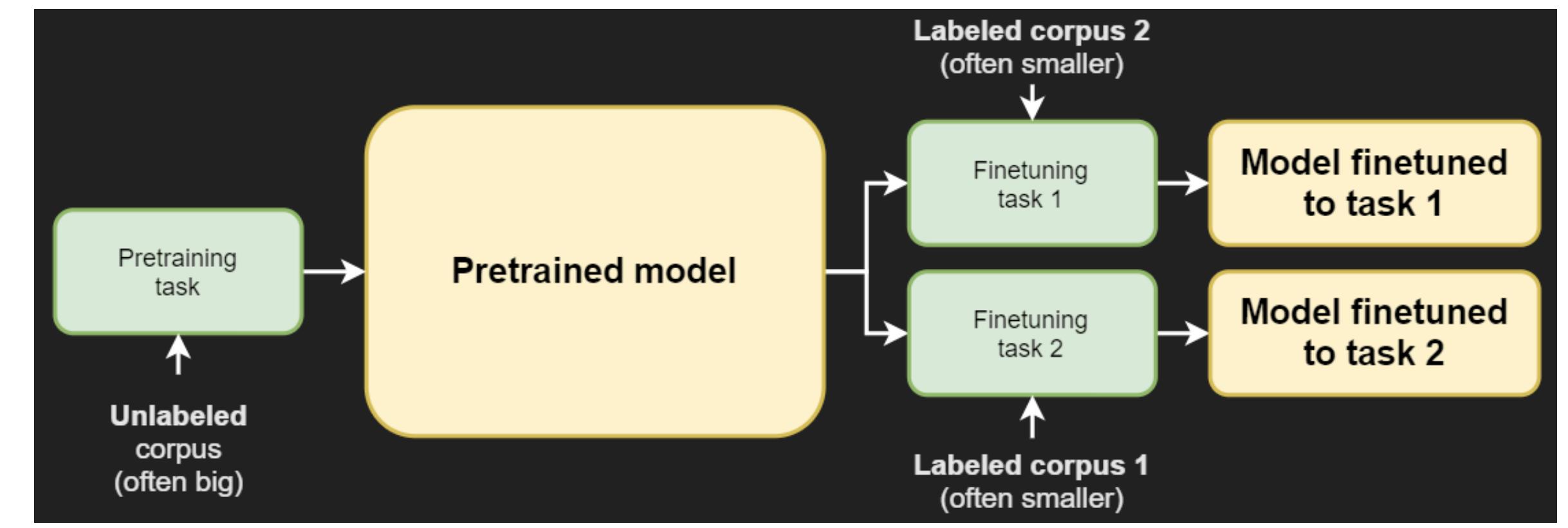
Pre-training and fine-tuning

Neural networks can learn features that are useful for multiple tasks.

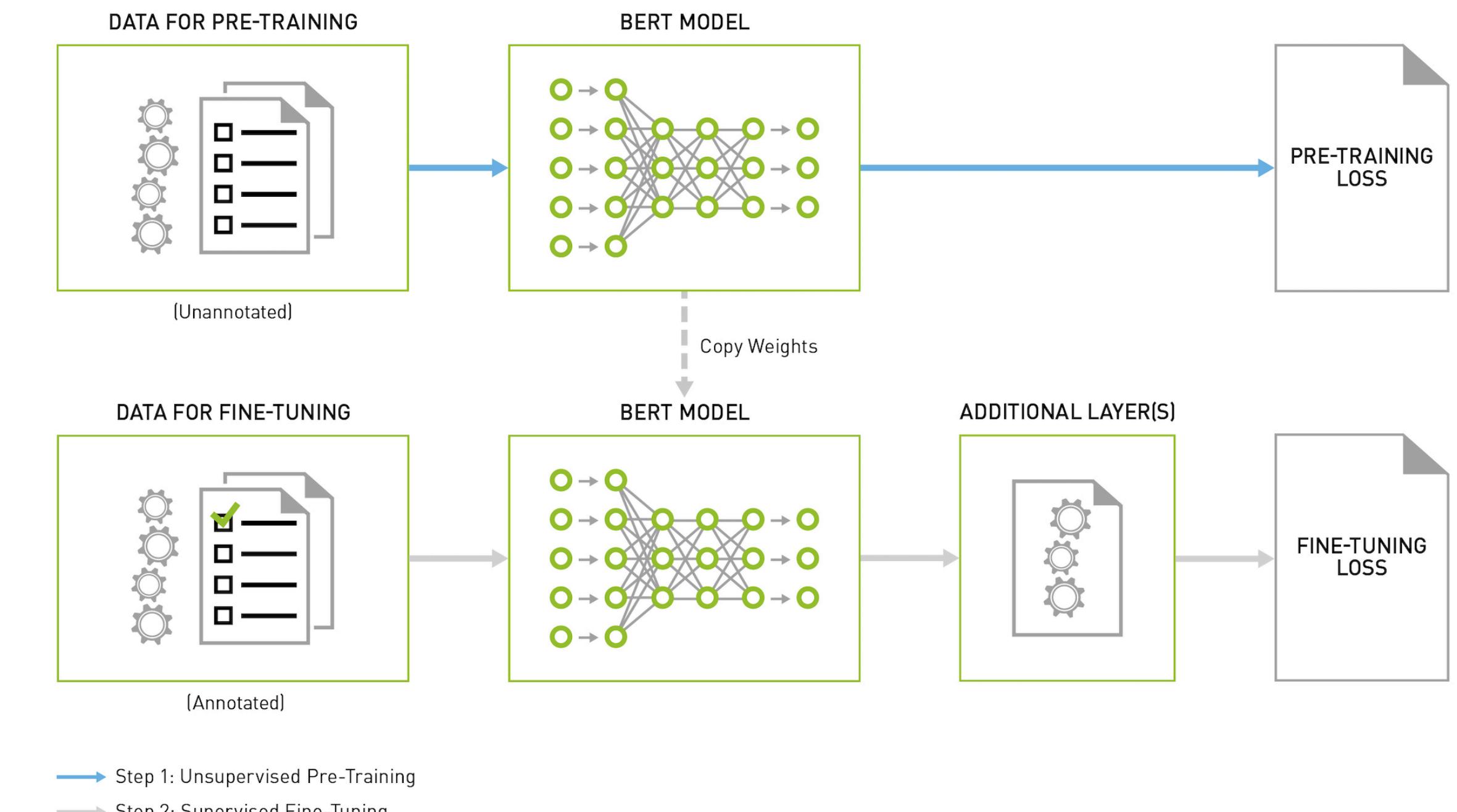
Paradigm: Unsupervised feature learning (pre-training) followed by fine-tuning to specific tasks.

The final model is obtained by adding a few extra layers on top of pre-trained model and training those.

BERT (trained on the entirety of Wikipedia) can be easily fine-tuned for many NLP tasks.



<https://www.machinecurve.com/index.php/2021/01/04/intuitive-introduction-to-bert/>



<https://medium.com/future-vision/bert-meets-gpus-403d3fbcd848>