

AI Bootcamp

---

# Advanced NLP Techniques: Topic Modeling and RNN

Module 20 Day 3

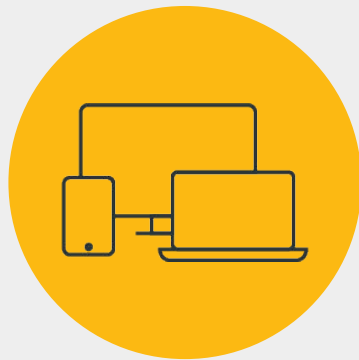


# Class Objectives

By the end of class, you will be able to:

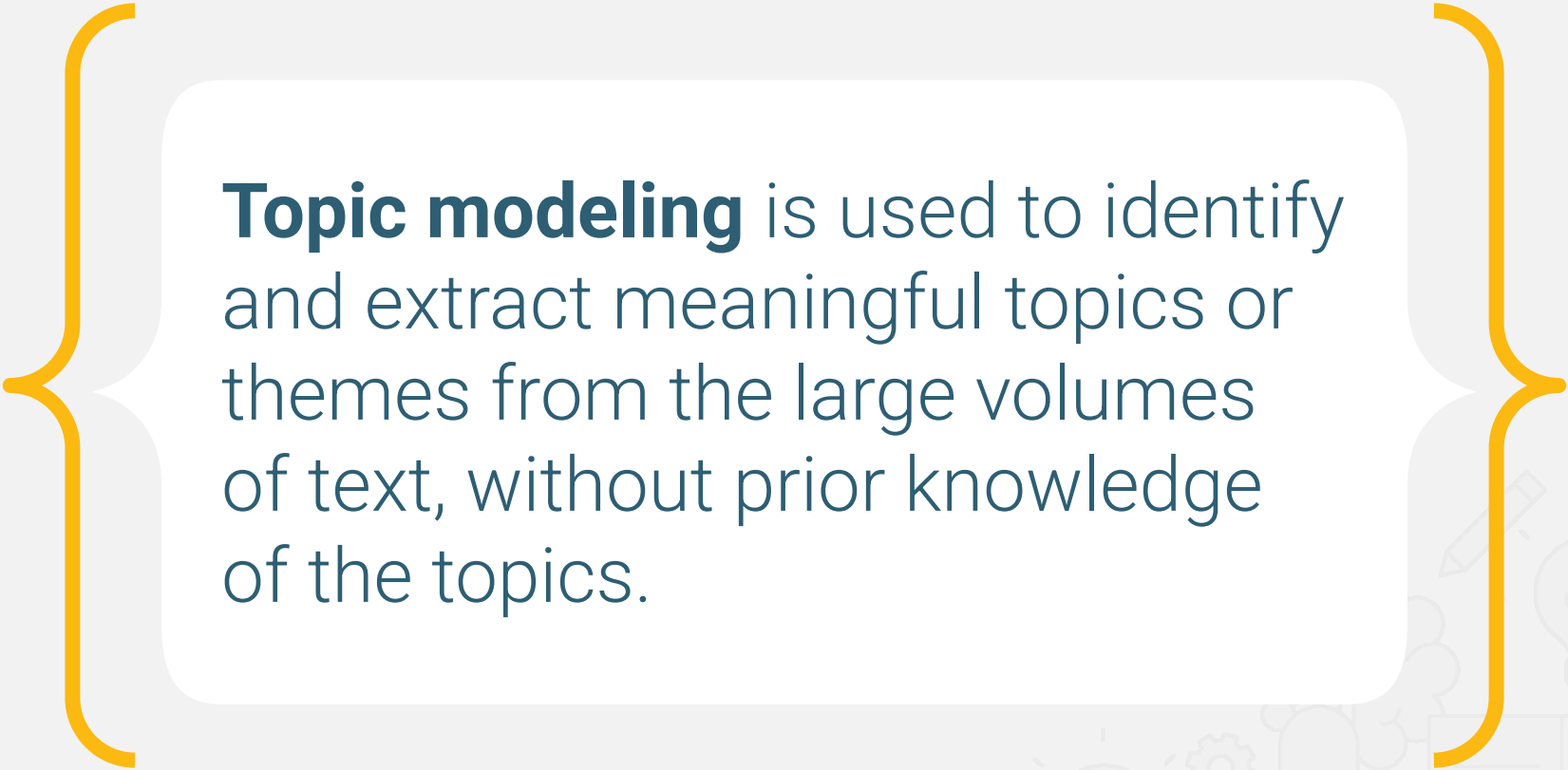
---

- 1 Apply NLP preprocessing to large corpora of text.
- 2 Demonstrate how to classify text into topics using unsupervised learning.
- 3 Understand and demonstrate how to use LSTM RNN to generate text.

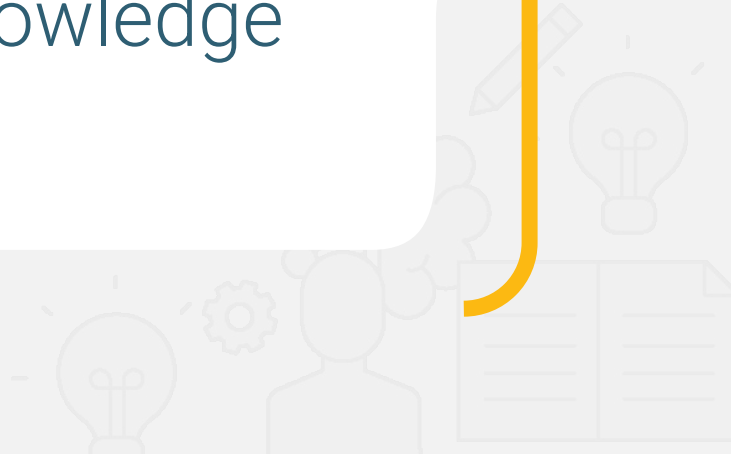


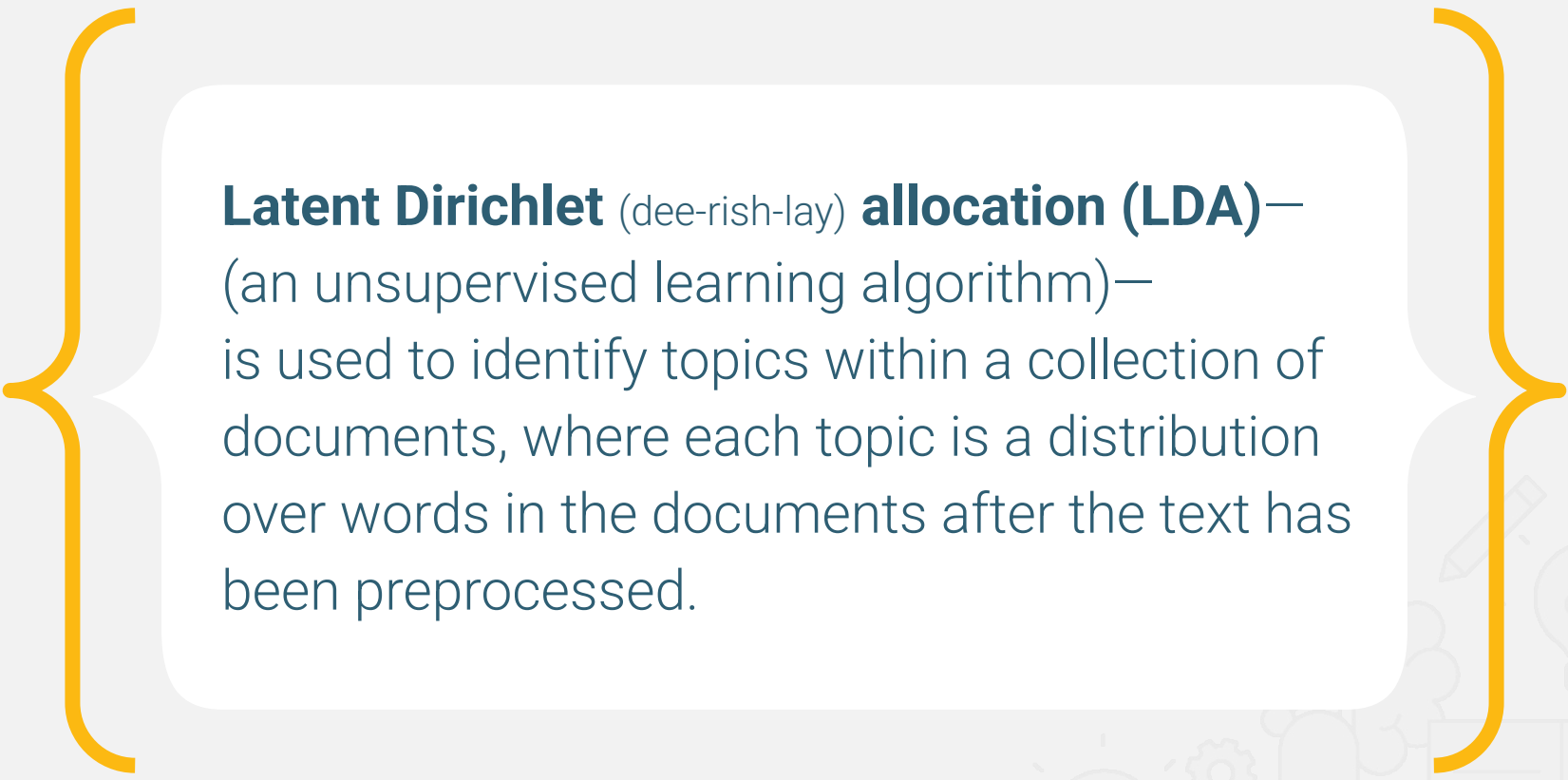
# Instructor **Demonstration**

Topic Modeling with Latent Dirichlet Allocation (LDA)

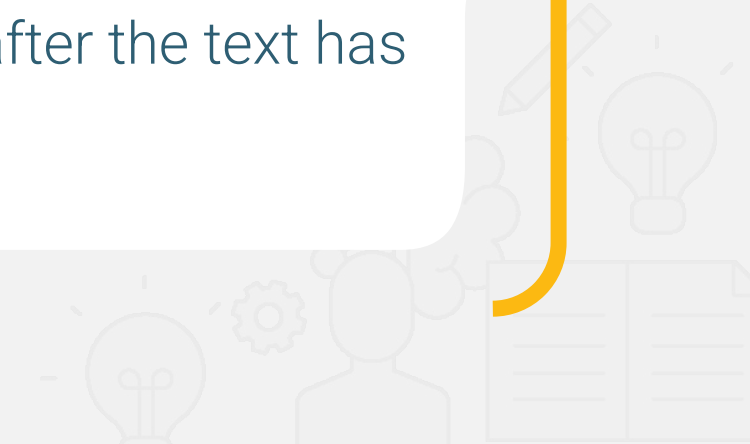


**Topic modeling** is used to identify and extract meaningful topics or themes from the large volumes of text, without prior knowledge of the topics.





**Latent Dirichlet** (dee-rish-lay) **allocation (LDA)**—  
(an unsupervised learning algorithm)—  
is used to identify topics within a collection of  
documents, where each topic is a distribution  
over words in the documents after the text has  
been preprocessed.





# Topic Modeling with LDA

LDA assumes that documents are mixtures of topics and that topics are mixtures of words. It starts with a fixed number of topics, which the user defines, and aims to find two probability distributions:

## **Document-topic distribution:**

For each document, what is the probability of it belonging to each topic?

## **Topic-word distribution:**

For each topic, what is the probability of each word being associated with that topic?

# An Example of Documents

Document 0: Cleveland Browns quarterback Deshaun Watson to undergo season-ending surgery for shoulder injury.

Document 1: Plane turns back to JFK after horse escapes on board.

Document 2: Pizza Hut selling snake pizza in Hong Kong.

Document 3: Inside the remarkably intricate planning for Biden's meeting with Xi.

Document 4: Eight-year-old boy becomes youngest person to climb California's El Capitan

Document 5: US retail sales fell in October for the first time in seven months.

Document 6: From soups to cheese: what seaweed can bring to the dinner table.

Document 7: Escape the crowds at these affordable alternatives to travel hot spots.

Document 8: House passes stopgap bill to avert government shutdown.

Document 9: Walmart, Costco and other companies rethink self-checkout.

# Preparing Text Data: Cleaning the Text

Clean the text by removing punctuation and numbers.

```
# Convert each document to a unicode string.
def clean_text(text):
    # Remove non-alphabetic characters
    text = re.sub(r'^a-zA-Z\s', '', text)
    # Convert to lowercase
    return text.lower()

# Get the cleaned documents
cleaned_documents = [clean_text(doc) for doc in documents]
```



# Cleaned Documents

['Cleveland browns quarterback deshaun watson to undergo season-ending surgery for shoulder injury',  
'plane turns back to JFK after horse escapes on board',  
'pizza hut selling snake pizza in hong kong',  
'inside the remarkably intricate planning for bidens meeting with 'xi',  
'eightyearold boy becomes youngest person to climb californias el capitan',  
'us retail sales fell in october for the first time in seven months',  
'from soups to cheese what seaweed can bring to the dinner table',  
'escape the crowds at these affordable alternatives to travel hot spots',  
'house passes stopgap bill to avert government shutdown',  
'walmart costco and other companies rethink selfcheckout']

# Preparing Text Data: Tokenization and DTM

1 Tokenize the text to words and remove stopwords to create a vocabulary.

2 Process all the documents—each headline or summary—into a document term matrix (DTM) that has the frequency of the words that occur in each document. Where every row is a document and every column is the tokens or the words.

```
from sklearn.feature_extraction.text import CountVectorizer
# Use CountVectorizer to tokenize the text
vectorizer = CountVectorizer(stop_words='english')
# Use fit_transform to create the DTM
dtm = vectorizer.fit_transform(cleaned_documents)
```

# A DTM after Processing

If a word from the vocabulary appears in the document the value is a “1,” otherwise it’s “0.”

	affordable	alternatives	avert	bidens	board	boy	bring	browns	californias	capitan	...	surgery	table	time	travel	turns	undergo	walmart	watson	:
0	0	0	0	0	0	0	0	1	0	0	...	1	0	0	0	0	1	0	1	
1	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	1	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
3	0	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	1	0	0	1	1	...	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	0	...	0	0	1	0	0	0	0	0	
6	0	0	0	0	0	0	1	0	0	0	...	0	1	0	0	0	0	0	0	
7	1	1	0	0	0	0	0	0	0	0	...	0	0	0	1	0	0	0	0	
8	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	0	

10 rows × 68 columns

# Non-negative Matrix Factorization (NMF)



Unsupervised Learning algorithm



It simultaneously performs dimensionality reduction and clustering-like PCA.



Unlike LDA, you can use it with TF-IDF to model topics across documents.



## Activity:

### BBC News Topic Modeling with LDA

---

In this activity, you will use LDA to determine the topic for BBC News summaries. After you have determined the label for each topic, you will add two new columns to the DataFrame that assigns each news summary a topic number and topic label.

**Suggested Time:**

20 Minutes



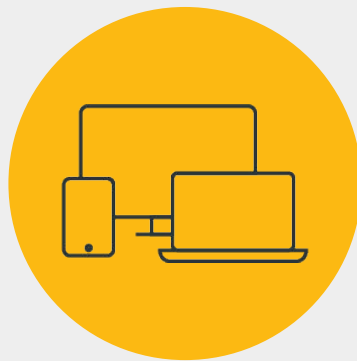


**Time's up!**  
Let's review



**Questions?**





# Instructor **Demonstration**

Topic Modeling with Non-negative Matrix Factorization

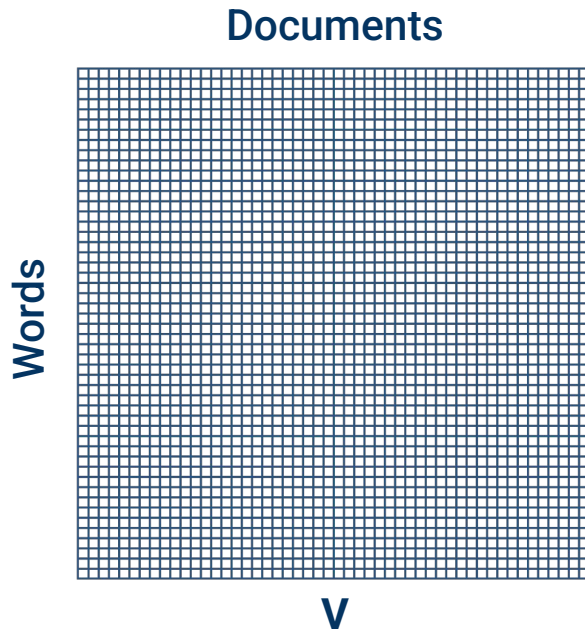


# Topic Modeling with NMF

Using TfidfVectorizer we get a matrix, we will call "M," that has rows and columns, where the rows represent documents, and columns represent unique terms, or words in your document collection.



**Vector (V)** = Documents x Terms



# Topic Modeling with NMF

When we apply topic modeling we use the matrix,  $M$ , from the `TfidfVectorizer`, which will be split into two non-negative matrices, “ $H$ ” for height and “ $W$ ” for width, where “ $H$ ” is a document-topic matrix and “ $W$ ” is a topic-word matrix.

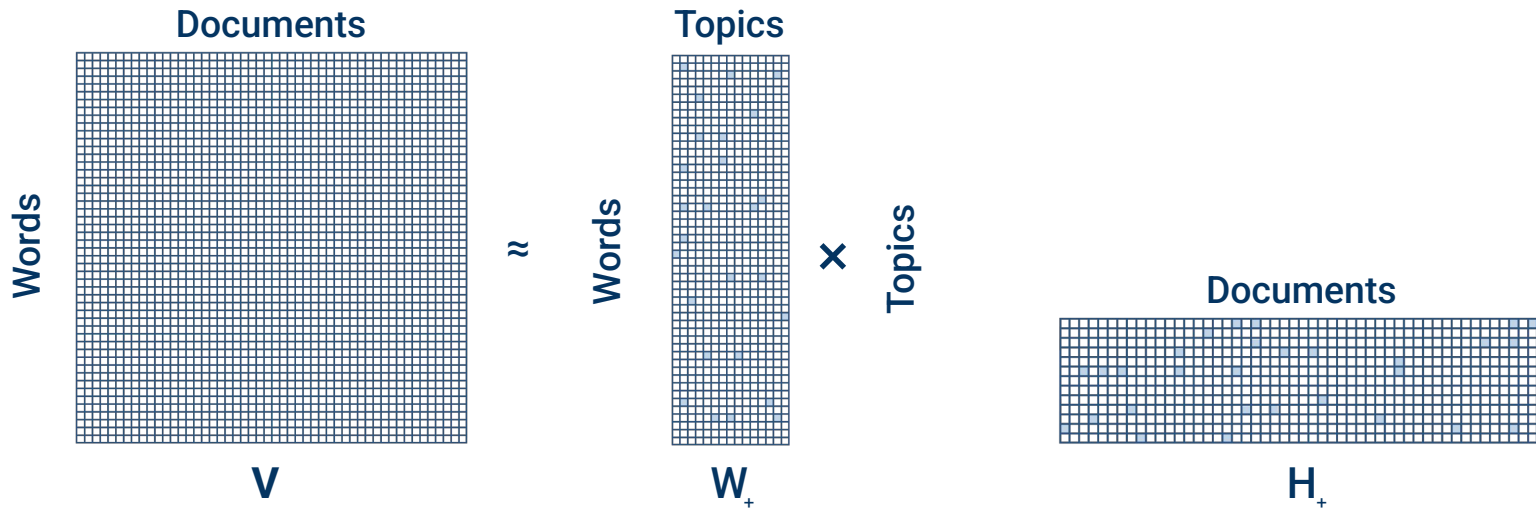


**$W$  = Topics x Words**



**$H$  = Documents x Topics**

$$V \approx W \times H$$



# A DTM after Processing

The matrix that is created from transforming the documents consists of:

A tuple, where the first number represents the row for each document, and the second number represents the index of the word in the vocabulary created by **fit\_transform**.

The last number is the value of the TF-IDF score for that word in the vocabulary.

(0, 29)	0.31622776601683794
(0, 52)	0.31622776601683794
(0, 58)	0.31622776601683794
(0, 47)	0.31622776601683794
(0, 63)	0.31622776601683794
(0, 65)	0.31622776601683794
(0, 16)	0.31622776601683794
(0, 42)	0.31622776601683794
(0, 7)	0.31622776601683794
(0, 11)	0.31622776601683794
(1, 4)	0.408248290463863
(1, 21)	0.408248290463863
(1, 25)	0.408248290463863
(1, 32)	0.408248290463863
(1, 62)	0.408248290463863
(1, 40)	0.408248290463863
(2, 33)	0.3333333333333333
(2, 24)	0.3333333333333333
(2, 54)	0.3333333333333333
(2, 50)	0.3333333333333333
(2, 28)	0.3333333333333333
(2, 39)	0.6666666666666666
(3, 66)	0.3779644730092272
(3, 34)	0.3779644730092272
(3, 3)	0.3779644730092272
(3, 41)	0.3779644730092272
(3, 31)	0.3779644730092272
(3, 43)	0.3779644730092272
(3, 30)	0.3779644730092272



## Activity:

### BBC News Topic Modeling with NMF

---

In this activity, you will code along with the instructor using NMF to determine the topic for BBC News summaries.

**Suggested Time:**

20 Minutes





**Time's up!**  
Let's review



# Questions?





**Break**

15 mins



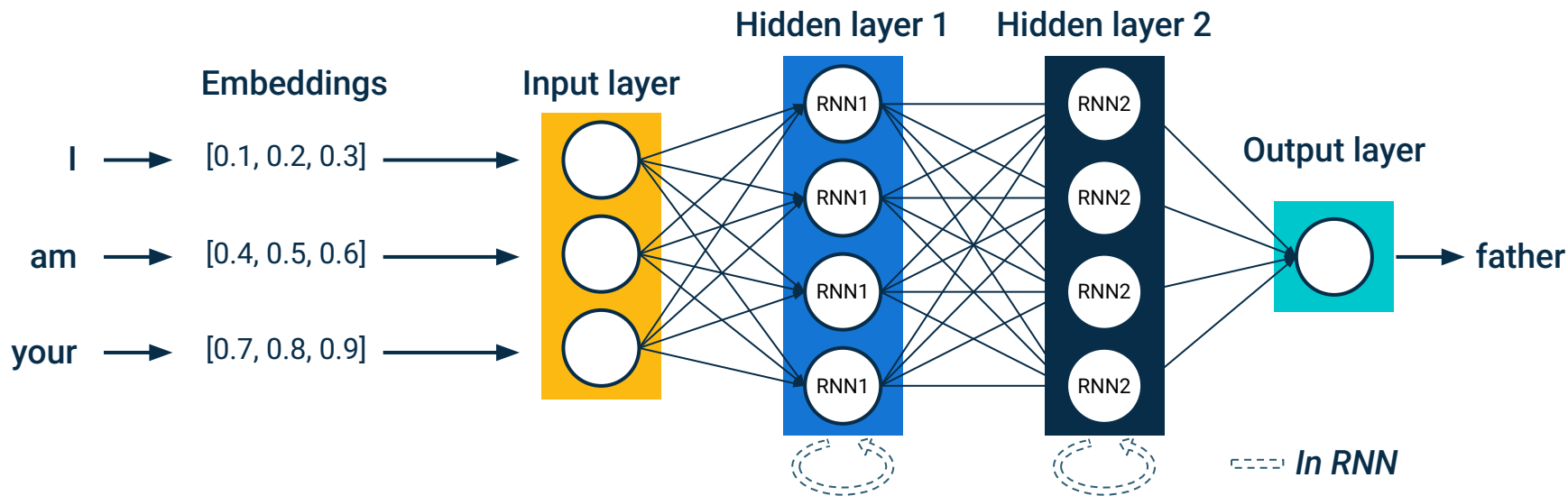
# Instructor **Demonstration**

Introduction to RNNs and LSTMs



# Recurrent Neural Networks (RNNs)

RNNs are able to remember the past. Their decisions are influenced by what they have learned in the past. Imagine we trained a simple RNN model on the text "I am your father". When we feed the trained model with "I am your," it will predict "father" as the output based on the word embeddings.





What are **recurrent  
neural networks**  
used for?





# What Are RNNs Used For?

RNNs are used for the following:

NLP

DNA sequencing

Time series data

Music composition

# What Are RNNs Used For?



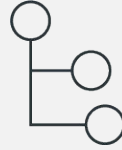
Send \$50 to Allison to be delivered today from my check account

I understand, Tom. I'd be happy to help you with that



# What Are RNNs Used For?

- NLP
- Speech Processing
- Video Analysis
- Healthcare
- Finance
- Video Games
- Recommendation Systems
- Weather Forecasting
- Autonomous Driving

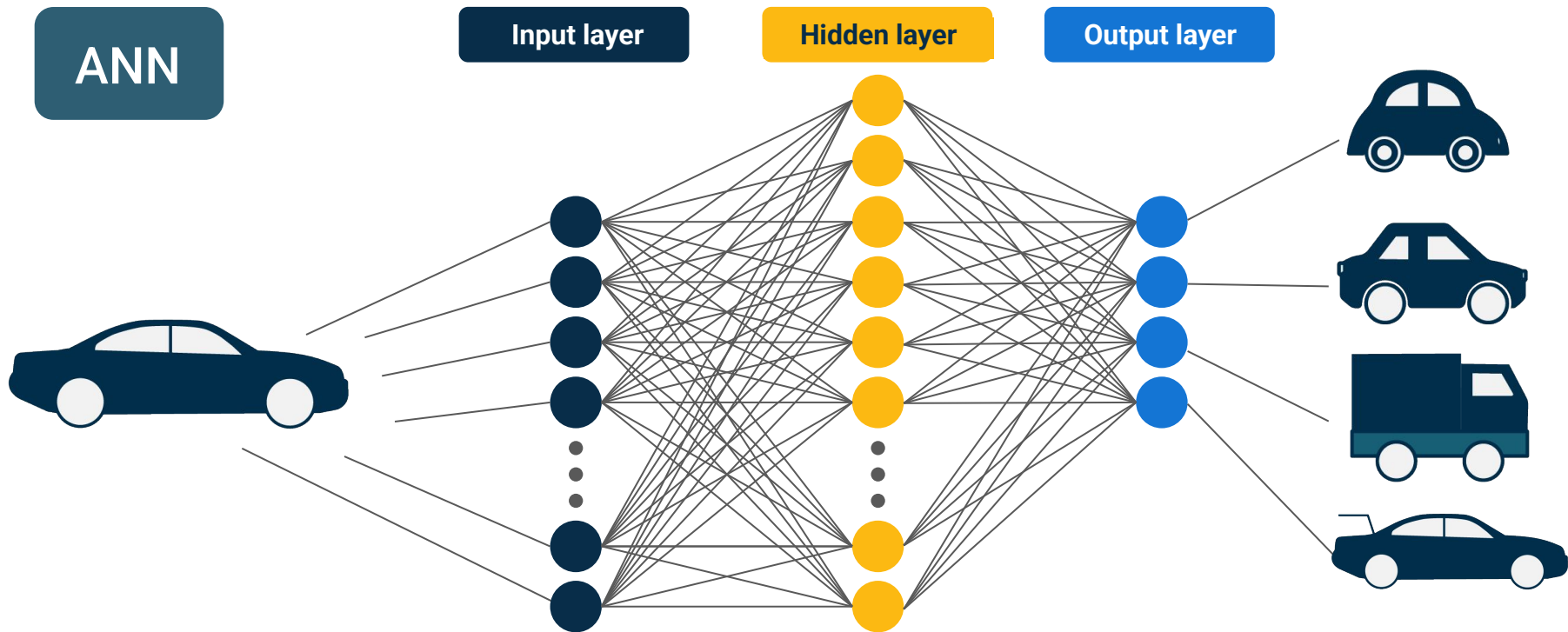


# Artificial neural networks (ANNs) **vs.** Recurrent Neural Networks (RNNs)



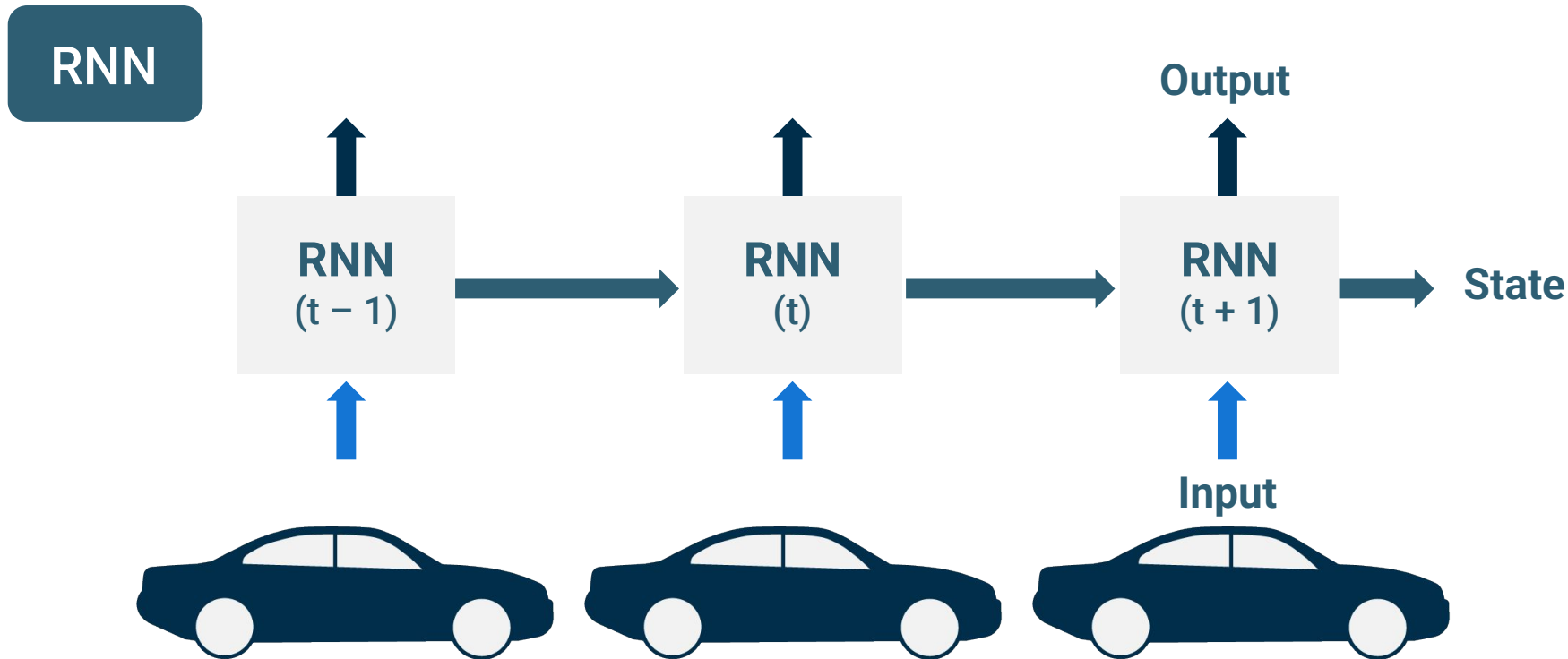
# ANNs vs. RNNs

We can use **ANNs** to identify the type of car from a still image. But can we predict the direction of a car in movement?



# ANNs vs. RNNs

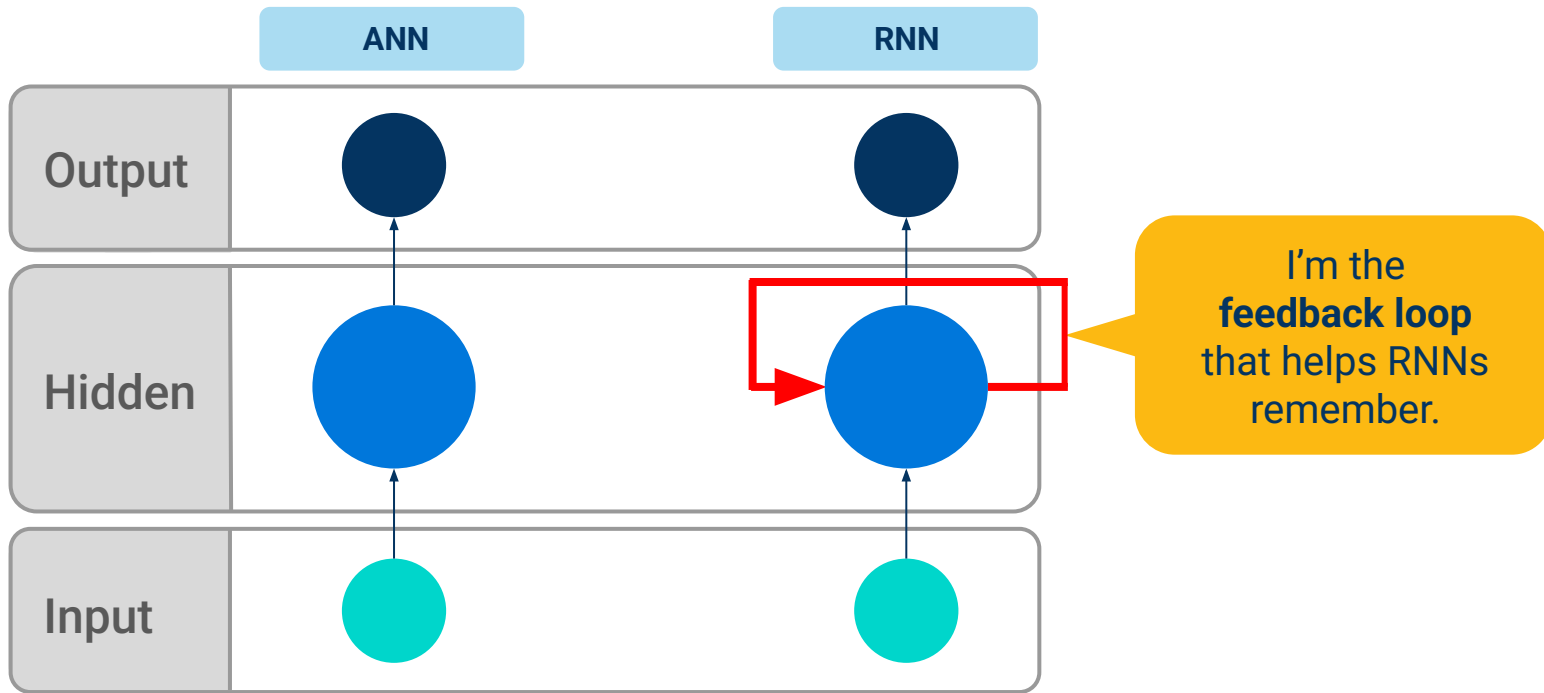
**RNNs** are good at modeling sequence data because of their **sequential memory**. Using RNNs, we can predict that the car is moving to the right, where  $t$  = time.





# ANNs vs. RNNs

RNNs are good at modeling sequence data because of their **sequential memory**. Using RNNs, we can predict that the car is moving to the right.





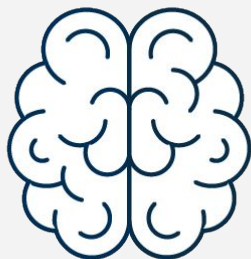
How do **RNNs** work?



# How do RNNs work?

When you read this  
sentence, your

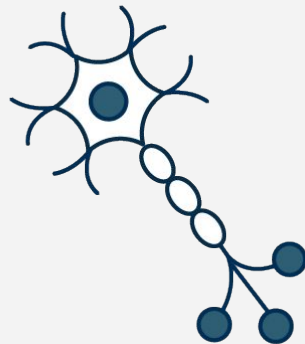
**brain**



is able to decode  
and understand it . . .

. . . because our

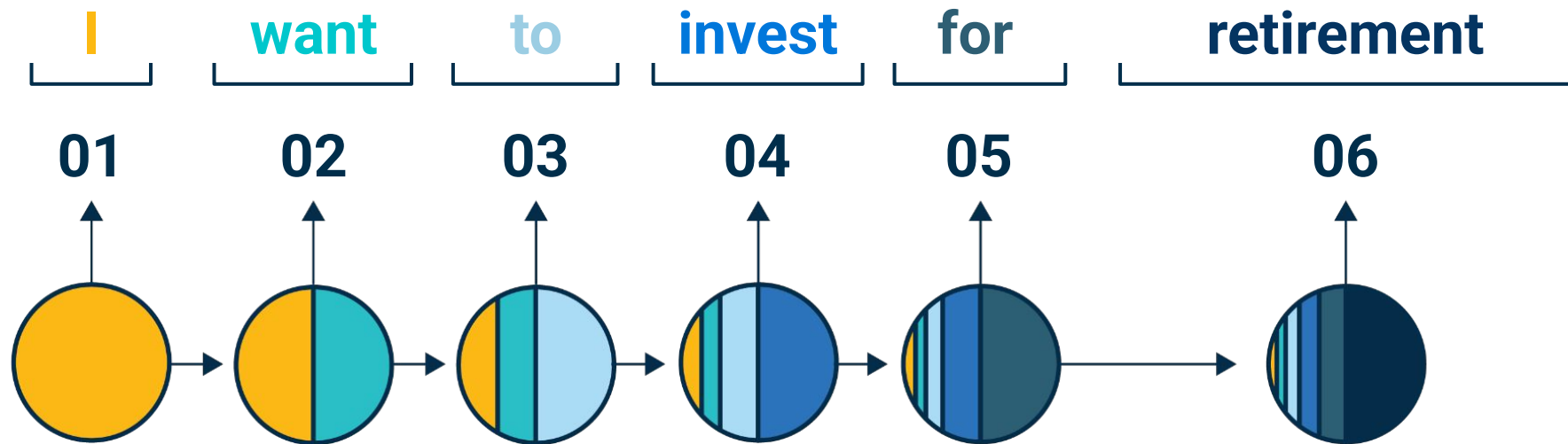
**neurons**



have memory,  
like RNNs.

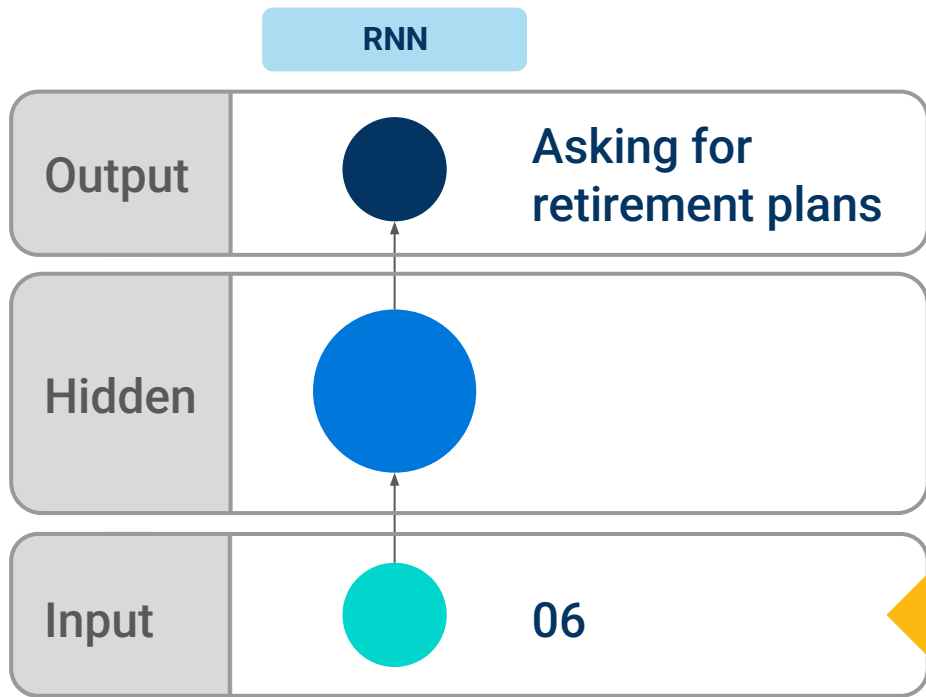
# How do RNNs work?

The sentence is split into individual words. RNNs work sequentially, so we feed it one word at a time. By the final step, the RNN has encoded information from all the words in previous steps.



# How do RNNs work?

RNNs are good at modeling sequence data because of their **sequential memory**. Using RNNs, we can predict that the car is moving to the right.

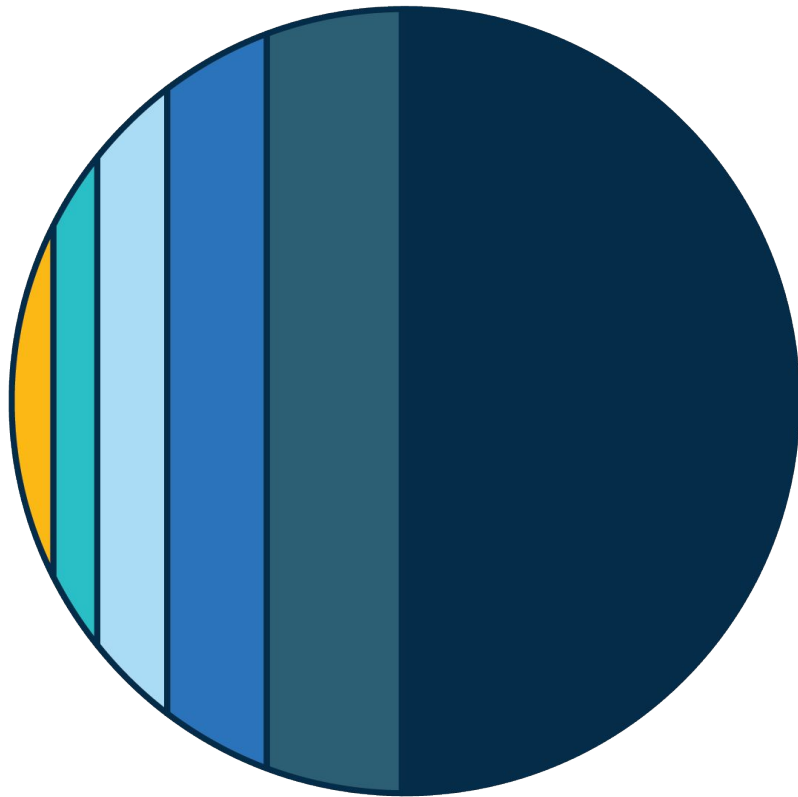


The final output was created from the rest of the sequence. To predict what the phrase means, we take the input and pass it to the feed-forward layer of the RNN to classify the intent.

# RNNs are forgetful

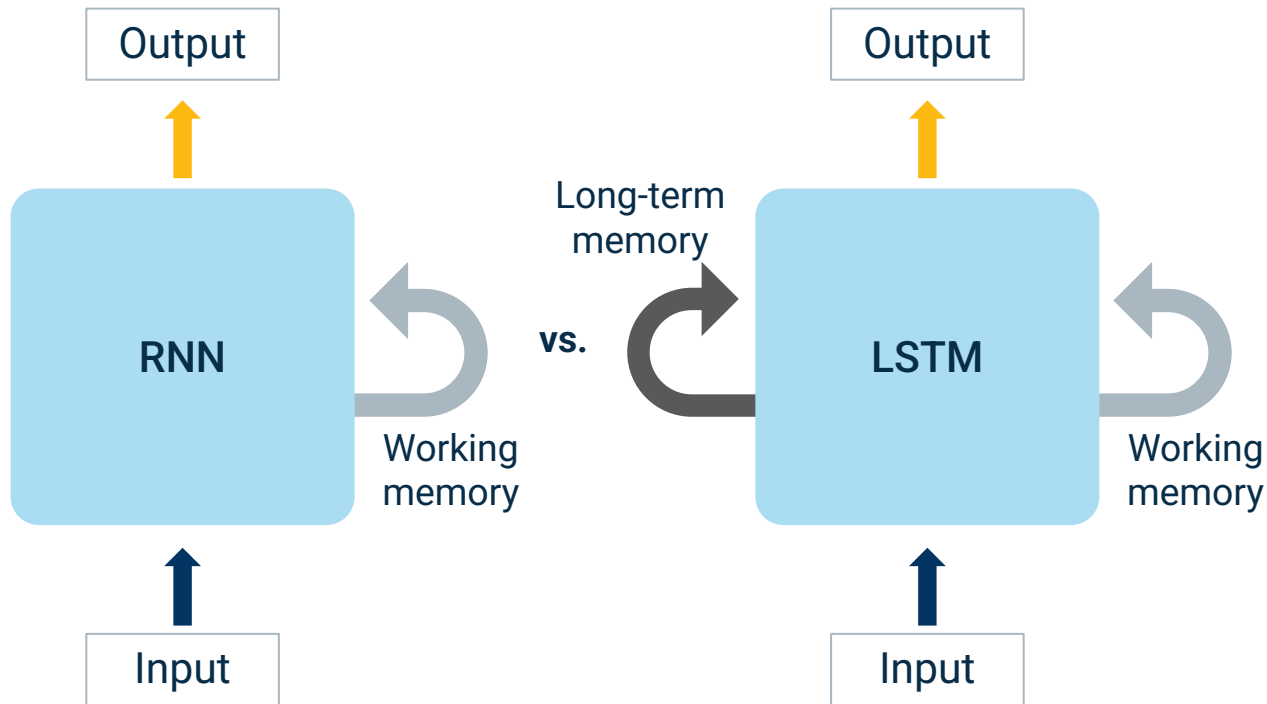
RNNs only “remember” the most recent few steps.

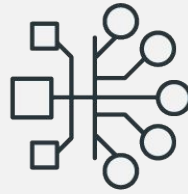
The vanishing gradient in the hidden states illustrates an issue with RNNs: **short-term memory**.



# RNNs vs. LSTMs

RNNs use their internal state (memory) to process sequences of inputs. Long short-term memory (LSTM) networks are a type of RNN, with additional long-term memory to remember past data.





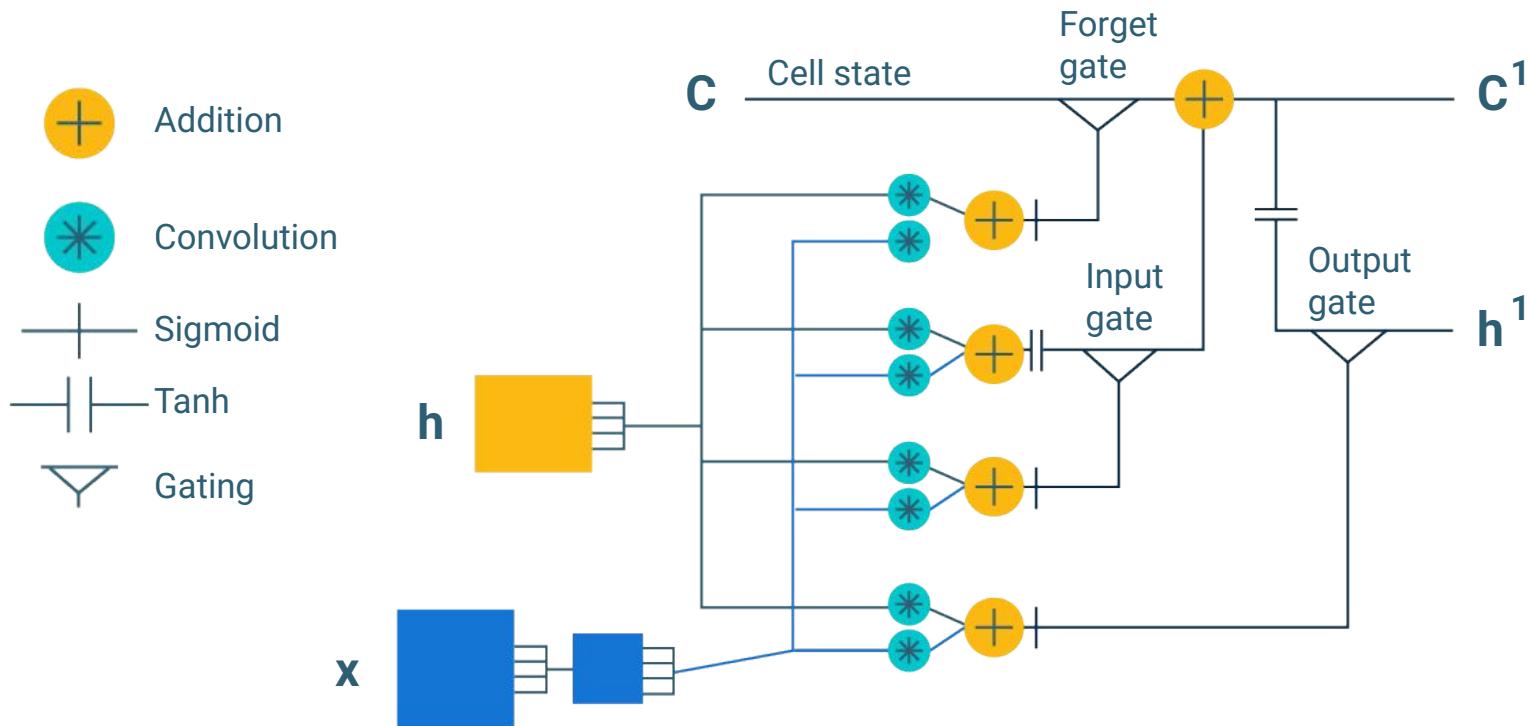
# Long Short-Term Memory (**LSTM**)





# LSTMs to the rescue

LSTM RNNs are one solution for longer time windows. An LSTM RNN works like an original RNN, but it selects which types of longer-term events are worth remembering and which are okay to forget.








# Instructor **Demonstration**

Automatic Text Generation with RNNs

# Automatic text generation with RNNs

In this demo, we will explore how an RNN can be used to automatically generate text.

 **Write With Transformer** gpt2 ⓘ

 Shuffle initial text     Trigger autocomplete or tab    Select suggestion ↑ ↓ and enter    Cancel suggestion esc

See how a modern neural network auto-completes your text 😊

This site, built by the [Hugging Face](#) team, lets you write a whole document directly from your browser, and you can trigger the Transformer anywhere using the Tab key. It's like having a smart machine that completes your thoughts 😊


Get started by typing a custom snippet, [check out the repository](#), or [try one of the examples](#). Have fun!




Want to learn more  
about **RNNs**?



# Take a look at this RNNs cheat sheet

 Shervine Amidi About

Projects Teaching Blog

About Afshine Amidi 

Recurrent Neural Networks

Overview

- Architecture structure
- Applications of RNNs
- Loss function
- Backpropagation

Handling long term dependencies

- Common activation functions
- Vanishing/exploding gradient
- Gradient clipping
- GRU/LSTM
- Types of gates
- Bidirectional RNN
- Deep RNN

Learning word representation

- Notations
- Embedding matrix
- Word2vec
- Skip-gram
- Negative sampling
- GloVe

Comparing words

- Cosine similarity
- t-SNE

[View PDF version on GitHub](#)

Would you like to see this cheatsheet in your native language? You can help us [translating it on GitHub!](#)

[CS 230 - Deep Learning](#) English [فارسی](#) [Français](#) [日本語](#) [한국어](#) [Türkçe](#)

Convolutional Neural Networks

**Recurrent Neural Networks**

Tips and tricks

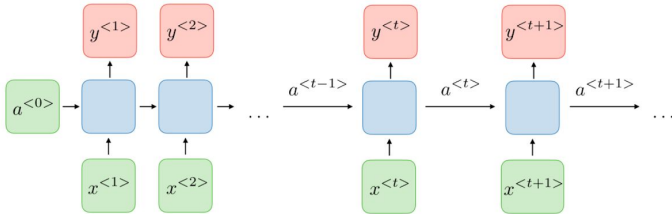
## Recurrent Neural Networks cheatsheet

★ Star 3,793

By [Afshine Amidi](#) and [Shervine Amidi](#)

### Overview

**Architecture of a traditional RNN** — Recurrent neural networks, also known as RNNs, are a class of neural networks that allow previous outputs to be used as inputs while having hidden states. They are typically as follows:



For each timestep  $t$ , the activation  $a^{<t>}$  and the output  $y^{<t>}$  are expressed as follows:

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \quad \text{and} \quad y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$



# Instructor **Demonstration**

Text Generation with LSTMs



## Activity:

### Sherlock Holmes Text Generation

---

In this activity, you will clean and tokenize a Sherlock Holmes short story, “A Case of Identity,” then create and train a LSTM using the tokenized text. After the model has been trained you will provide the model with 25 words from the short novel as a seed text, then the model will return the next 25 words from the short story.

**Suggested Time:**

20 Minutes





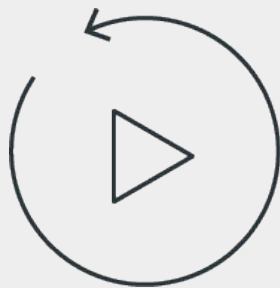
**Time's up!**  
Let's review





**Questions?**





Let's **recap**



# Recap

After today's lesson you are able to:

---

- 1 Apply NLP preprocessing to large corpora of text.
- 2 Demonstrate how to classify text into topics using unsupervised learning.
- 3 Understand and demonstrate how to use LSTM RNN to generate text.



**Questions?**





**The End**