MIT EECS
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

# 6.036: Introduction to Machine Learning

Final lecture! Thanks for joining us on this adventure!

**Lecture start:** Tuesdays 9:35am

**Who's talking?** Prof. Tamara Broderick

**Questions?** Ask on Piazza: "lecture (week) 13" folder

**Materials:** slides, video will all be available on Canvas

**Live Zoom feed:** https://mit.zoom.us/j/94238622313

## Last Time(s)

I. State machines & MDPs
II. Actions change state of world and give reward
III. Choosing "best" actions

## Today's Plan

I. Back to supervised learning
II. Sequential data
III. Recurrent neural networks

# Review

# Review

expected reward

# Review

- $V_\pi^h(s)$: expected reward at horizon *h* with policy $\pi$ starting at *s*

# Review

- $V_{\pi}^{h}(s)$: expected reward at horizon *h* with policy $\pi$ starting at *s*

- $Q^{h}(s, a)$ : expected reward of starting at *s*, making action *a*, and then making the “best” action for the *h*-1 steps left

# Review

- $V_\pi^h(s)$: expected reward at horizon *h* with policy $\pi$ starting at *s*
  - Can compute value of a policy even if not optimal
- $Q^h(s,a)$ : expected reward of starting at *s*, making action *a*, and then making the “best” action for the *h*-1 steps left

# Review

- $V_\pi^h(s)$: expected reward at horizon *h* with policy $\pi$ starting at *s*
  - Can compute value of a policy even if not optimal
- $Q^h(s,a)$ : expected reward of starting at *s*, making action *a*, and then making the "best" action for the *h*-1 steps left
  - Don't specify a policy $\pi$ in advance to compute *Q*. Rather, we typically use *Q* to figure out the "best" policy.

# Review

- $V_{\pi}^{h}(s)$: expected reward at horizon *h* with policy $\pi$ starting at *s*
  - Can compute value of a policy even if not optimal
- $Q^{h}(s, a)$ : expected reward of starting at *s*, making action *a*, and then making the "best" action for the *h*-1 steps left
  - Don't specify a policy $\pi$ in advance to compute *Q*. Rather, we typically use *Q* to figure out the "best" policy.
- *Reinforcement learning* (RL): learning (to maximize rewards) by interacting with the world

# Review

- $V_\pi^h(s)$: expected reward at horizon *h* with policy $\pi$ starting at *s*
  - Can compute value of a policy even if not optimal
- $Q^h(s,a)$ : expected reward of starting at *s*, making action *a*, and then making the “best” action for the *h*-1 steps left
  - Don’t specify a policy $\pi$ in advance to compute *Q*. Rather, we typically use *Q* to figure out the “best” policy.
- *Reinforcement learning* (RL): learning (to maximize rewards) by interacting with the world
  - Contrast with *supervised learning*

# Review

- $V_{\pi}^{h}(s)$: expected reward at horizon *h* with policy $\pi$ starting at *s*
  - Can compute value of a policy even if not optimal
- $Q^{h}(s, a)$ : expected reward of starting at *s*, making action *a*, and then making the "best" action for the *h*-1 steps left
  - Don't specify a policy $\pi$ in advance to compute *Q*. Rather, we typically use *Q* to figure out the "best" policy.
- *Reinforcement learning* (RL): learning (to maximize rewards) by interacting with the world
  - Contrast with *supervised learning*
  - Contrast with the $Q^h$ or $Q^*$ function (expected reward of starting at *s*, making action *a*, and then making the "best" action ever after)

# Review

- $V_\pi^h(s)$: expected reward at horizon *h* with policy $\pi$ starting at *s*
  - Can compute value of a policy even if not optimal
- $Q^h(s,a)$ : expected reward of starting at *s*, making action *a*, and then making the "best" action for the *h*-1 steps left
  - Don't specify a policy $\pi$ in advance to compute *Q*. Rather, we typically use *Q* to figure out the "best" policy.

- *Reinforcement learning* (RL): learning (to maximize rewards) by interacting with the world
  - Contrast with *supervised learning*
  - Contrast with the $Q^h$ or $Q^*$ function (expected reward of starting at *s*, making action *a*, and then making the "best" action ever after)
  - Contrast with (any horizon) *value iteration*

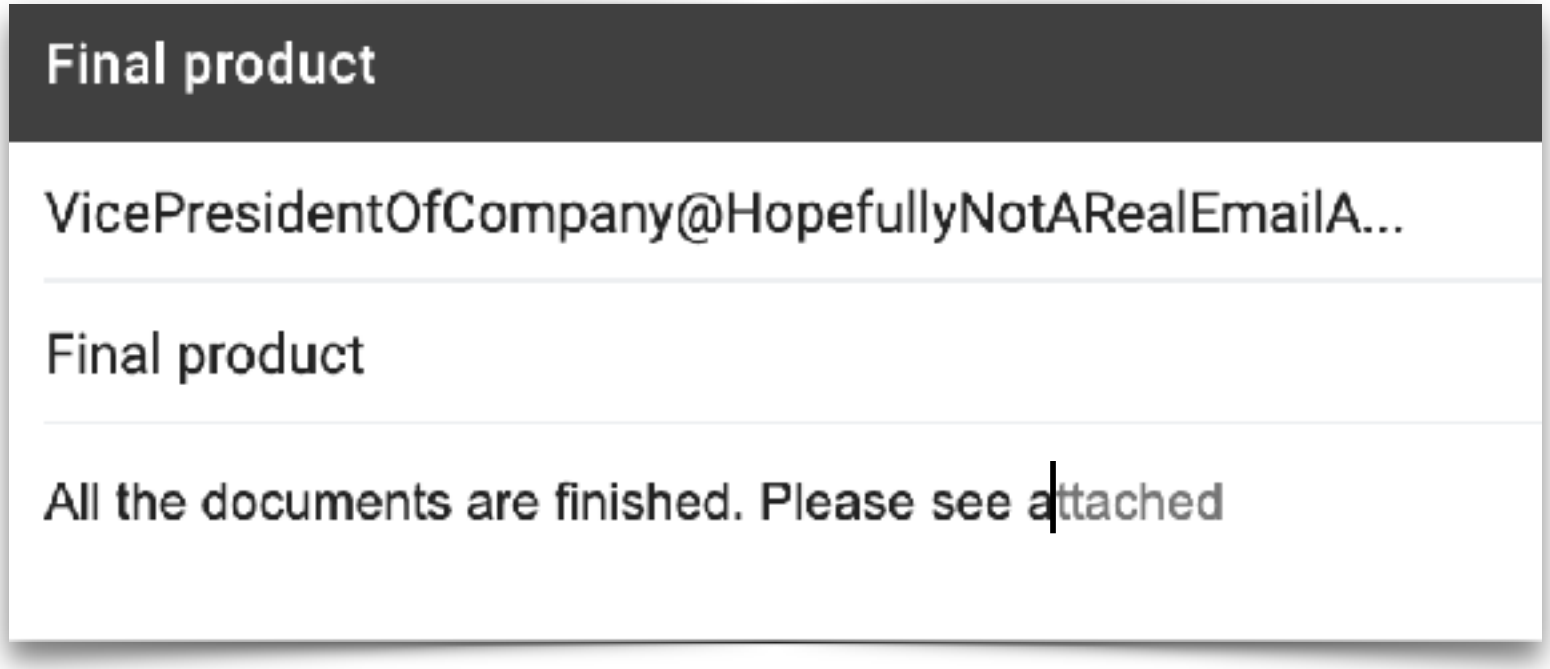
# Review

- $V_\pi^h(s)$: expected reward at horizon *h* with policy $\pi$ starting at *s*
  - Can compute value of a policy even if not optimal
- $Q^h(s, a)$ : expected reward of starting at *s*, making action *a*, and then making the "best" action for the *h*-1 steps left
  - Don't specify a policy $\pi$ in advance to compute *Q*. Rather, we typically use *Q* to figure out the "best" policy.

- *Reinforcement learning* (RL): learning (to maximize rewards) by interacting with the world
  - Contrast with *supervised learning*
  - Contrast with the $Q^h$ or $Q^*$ function (expected reward of starting at *s*, making action *a*, and then making the "best" action ever after)
  - Contrast with (any horizon) *value iteration*
- Today: can use state machines with supervised learning

# Text prediction

# Text prediction



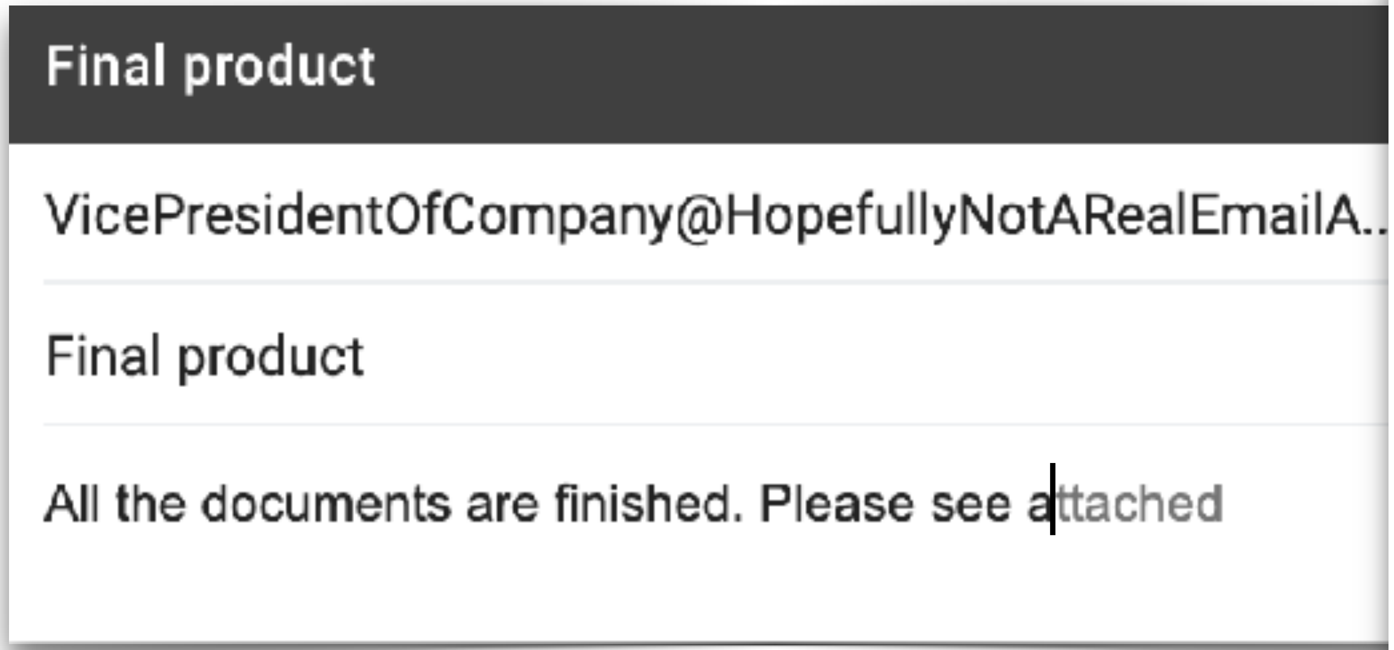
**Final product**

VicePresidentOfCompany@HopefullyNotARealEmailA...

Final product

All the documents are finished. Please see attached

# Text prediction

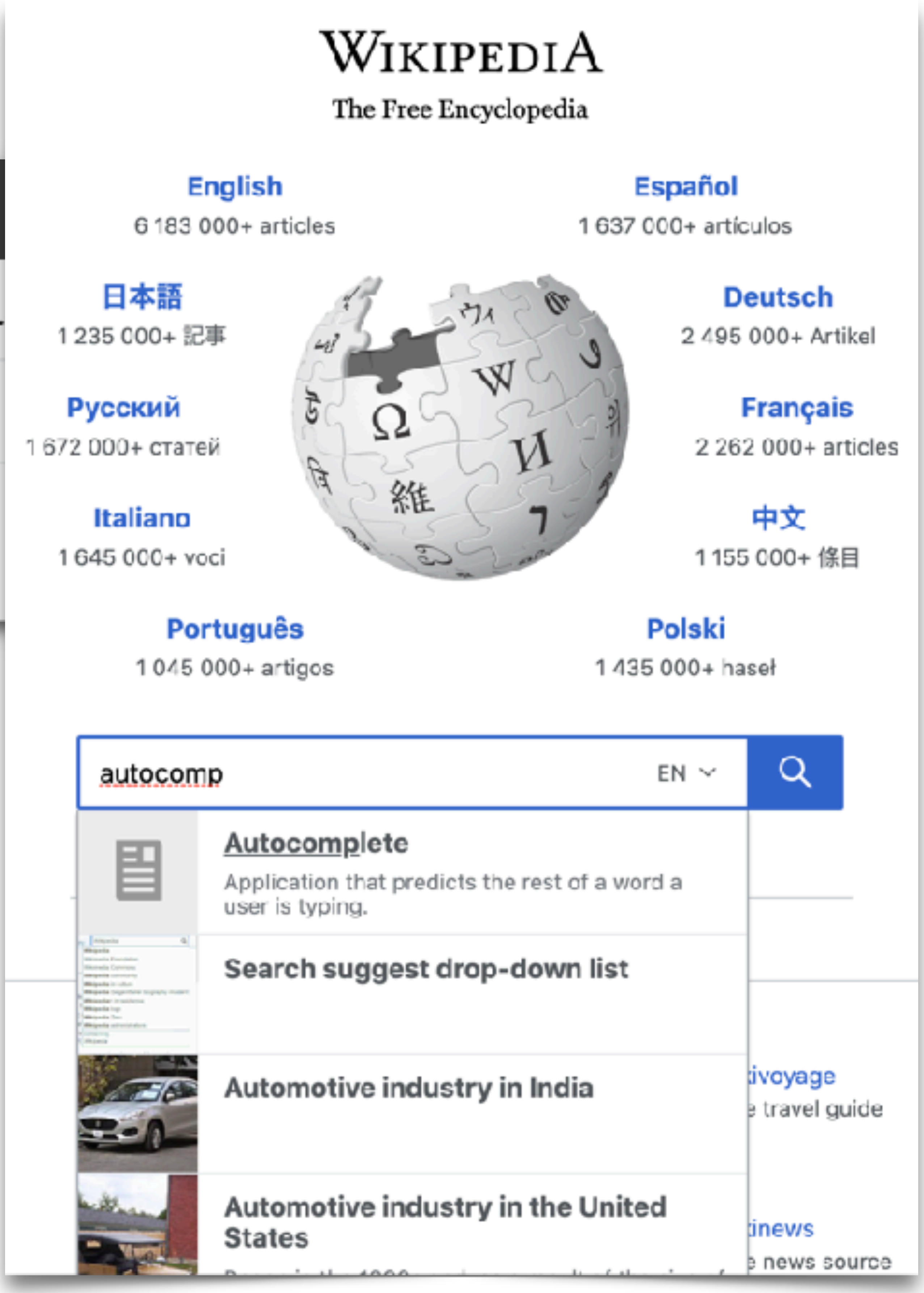


**Final product**

VicePresidentOfCompany@HopefullyNotARealEmailA...

Final product

All the documents are finished. Please see attached



WIKIPEDIA
The Free Encyclopedia

| | |
|---|---|
| **English**<br>6 183 000+ articles | **Español**<br>1 637 000+ artículos |
| **日本語**<br>1 235 000+ 記事 | **Deutsch**<br>2 495 000+ Artikel |
| **Русский**<br>1 672 000+ статей | **Français**<br>2 262 000+ articles |
| **Italiano**<br>1 645 000+ voci | **中文**<br>1 155 000+ 條目 |
| **Português**<br>1 045 000+ artigos | **Polski**<br>1 435 000+ haseł |

autocomp EN

**Autocomplete**
Application that predicts the rest of a word a user is typing.

**Search suggest drop-down list**

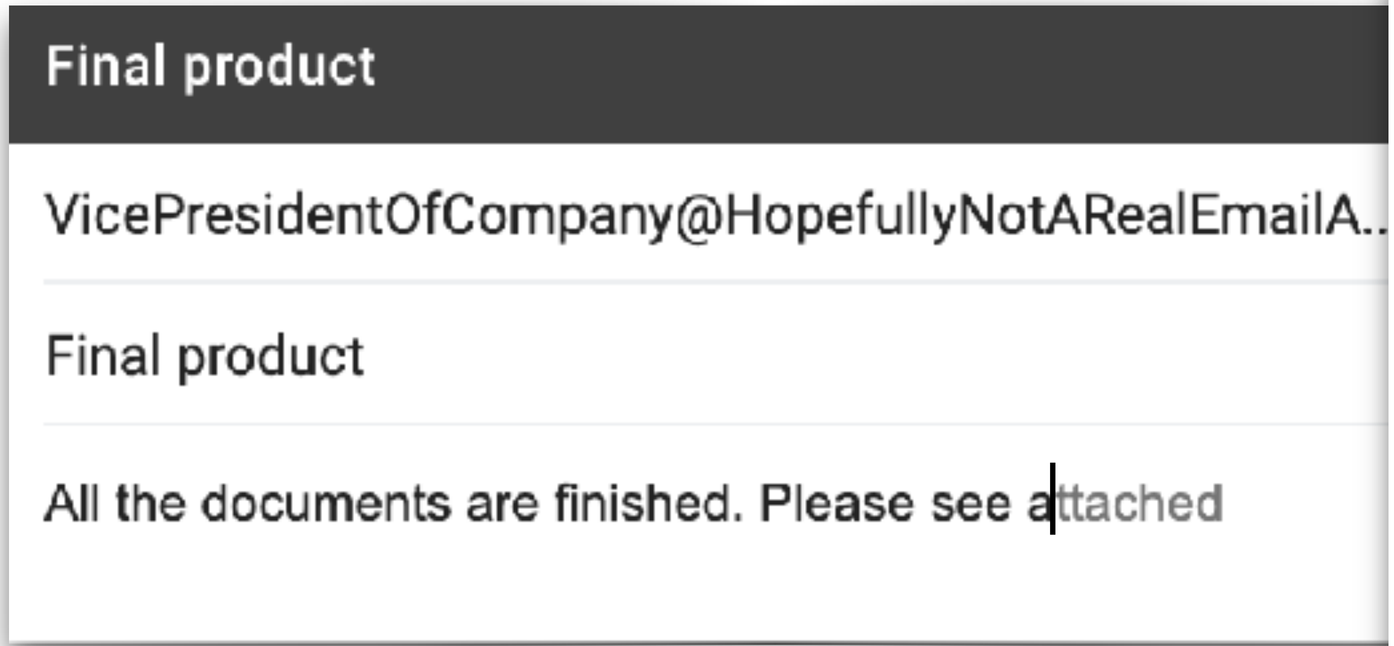
**Automotive industry in India**

**Automotive industry in the United States**

ivoyage
travel guide

inews
news source

# Text prediction

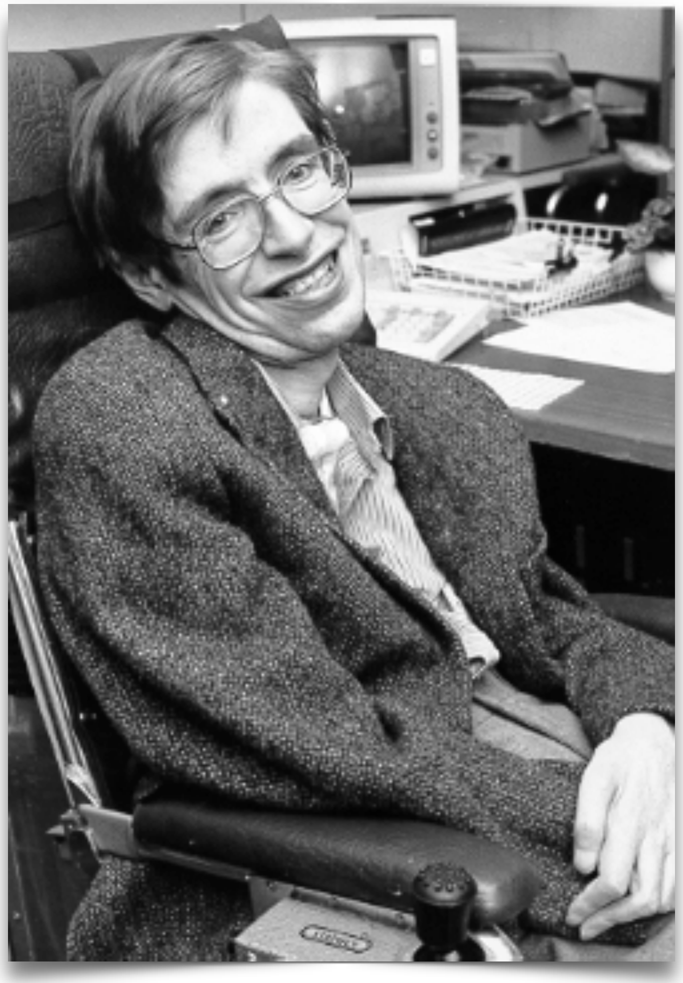


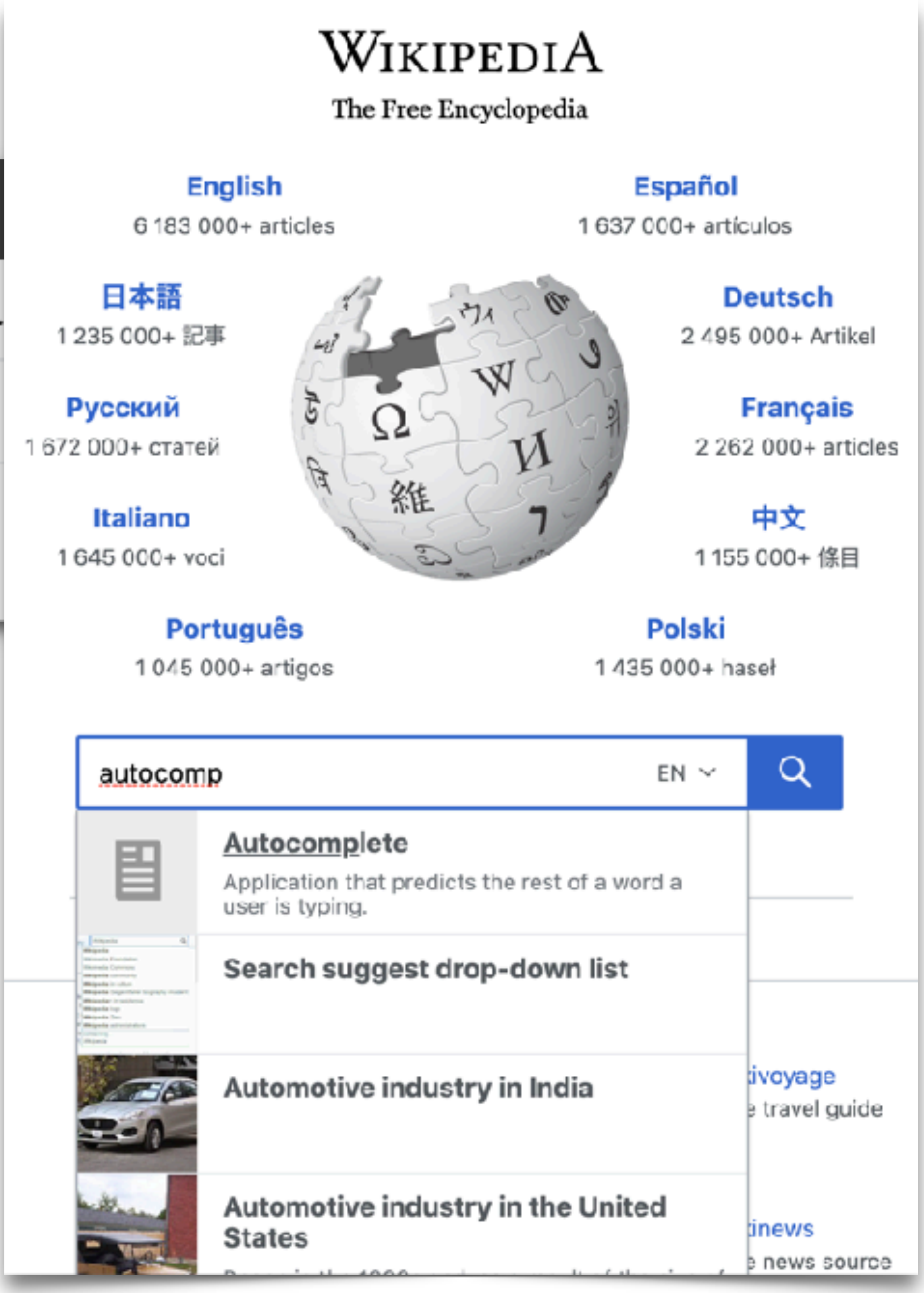
**Final product**

VicePresidentOfCompany@HopefullyNotARealEmailA...

Final product

All the documents are finished. Please see attached





WIKIPEDIA
The Free Encyclopedia

| | |
|---|---|
| **English**<br>6 183 000+ articles | **Español**<br>1 637 000+ artículos |
| **日本語**<br>1 235 000+ 記事 | **Deutsch**<br>2 495 000+ Artikel |
| **Русский**<br>1 672 000+ статей | **Français**<br>2 262 000+ articles |
| **Italiano**<br>1 645 000+ voci | **中文**<br>1 155 000+ 條目 |
| **Português**<br>1 045 000+ artigos | **Polski**<br>1 435 000+ haseł |

autocomp EN

**Autocomplete**
Application that predicts the rest of a word a user is typing.

**Search suggest drop-down list**

**Automotive industry in India**

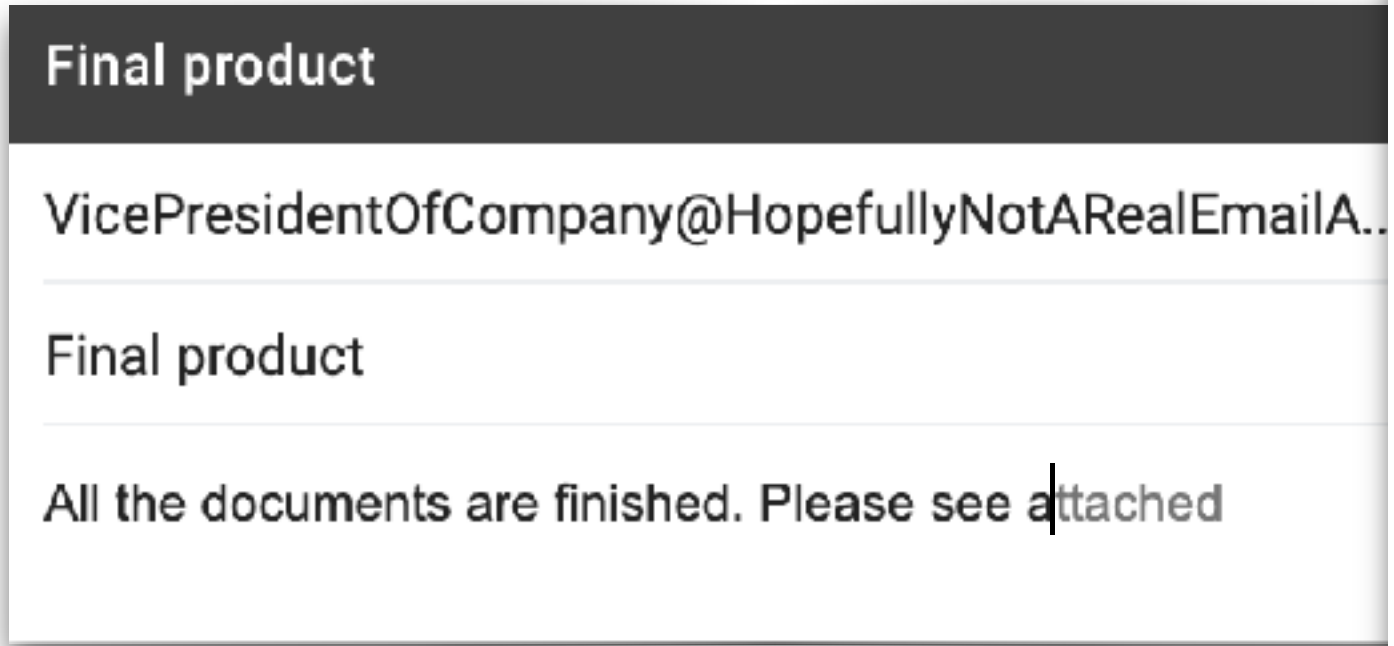
**Automotive industry in the United States**

ivoyage
travel guide

inews
news source

[https://commons.wikimedia.org/wiki/File:Stephen_Hawking.StarChild.jpg]

# Text prediction

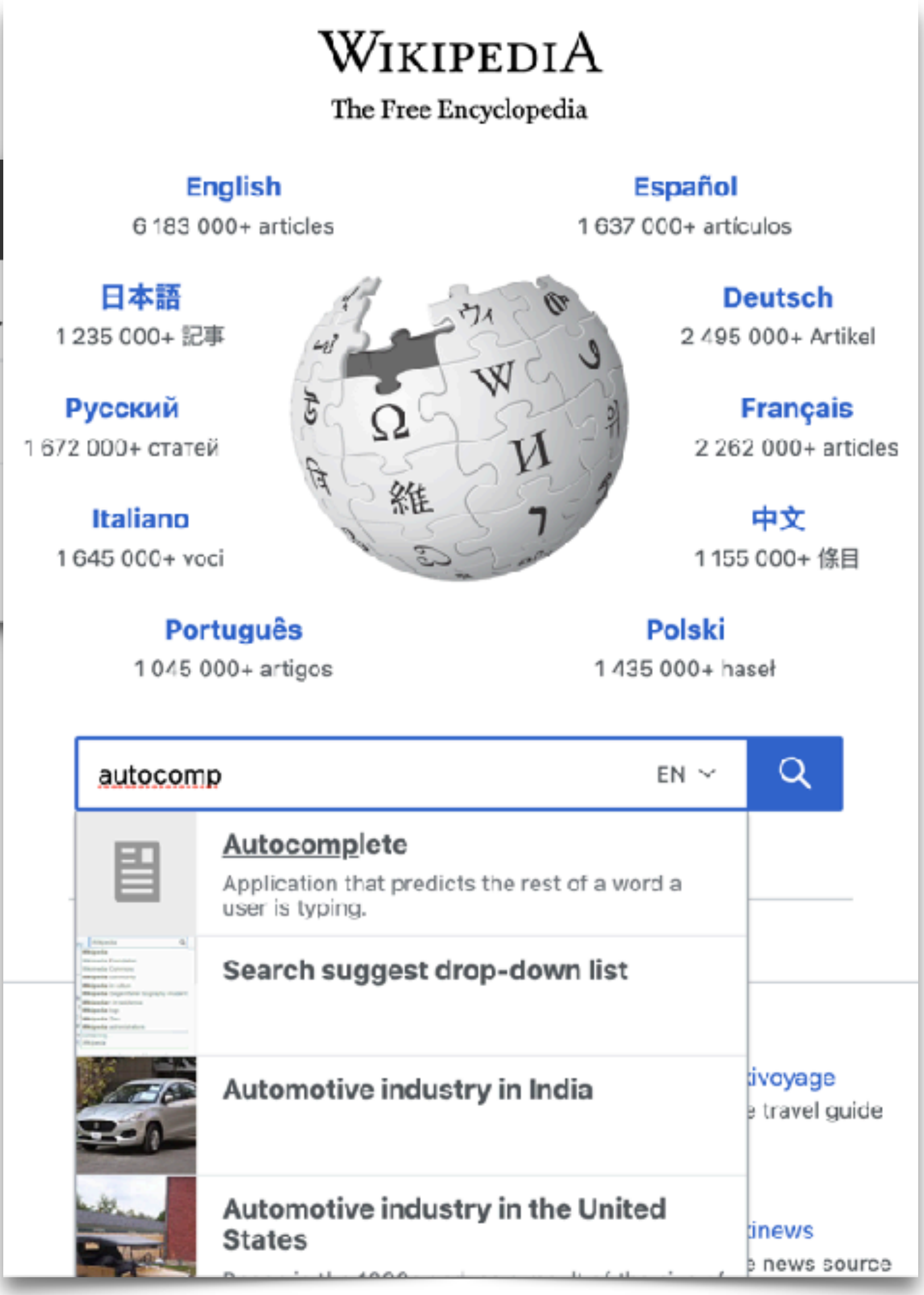


**Final product**

VicePresidentOfCompany@HopefullyNotARealEmailA...

Final product

All the documents are finished. Please see attached



WIKIPEDIA
The Free Encyclopedia

| | |
|---|---|
| English<br>6 183 000+ articles | Español<br>1 637 000+ artículos |
| 日本語<br>1 235 000+ 記事 | Deutsch<br>2 495 000+ Artikel |
| Русский<br>1 672 000+ статей | Français<br>2 262 000+ articles |
| Italiano<br>1 645 000+ voci | 中文<br>1 155 000+ 條目 |
| Português<br>1 045 000+ artigos | Polski<br>1 435 000+ haseł |

autocomp

**Autocomplete**
Application that predicts the rest of a word a user is typing.

**Search suggest drop-down list**

**Automotive industry in India**

**Automotive industry in the United States**

[https://commons.wikimedia.org/wiki/File:Stephen_Hawking.StarChild.jpg, www.itv.com/news/central/story/2014-08-05/locked-in-syndrome-woman-earns-degree-by-blinking]

# Text prediction: supervised learning

# Text prediction: supervised learning

- Training data: lots of text

# Text prediction: supervised learning

- Training data: lots of text
    - “what happens to a dream deferred”

# Text prediction: supervised learning

- Training data: lots of text
    - “what happens to a dream deferred”

| **features** | **label** |
|---|---|
| w | h |

# Text prediction: supervised learning

- Training data: lots of text
    - “what happens to a dream deferred”

| features | label |
|---|---|
| w | h |
| wh | a |

# Text prediction: supervised learning

- Training data: lots of text
  - “what happens to a dream deferred”

| features | label |
|---|---|
| w | h |
| wh | a |
| wha | t |

# Text prediction: supervised learning

- Training data: lots of text
  - “what happens to a dream deferred”

| features | label |
|---|---|
| w | h |
| wh | a |
| wha | t |
| what | _ |
| what_ | h |
| what_h | a |
| what_ha | p |
| what_hap | p |
| what_happ | e |

# Text prediction: supervised learning

- Training data: lots of text
    - “what happens to a dream deferred”

| features | label |
|---|---|
| w | h |
| wh | a |
| wha | t |
| what | _ |
| what_ | h |
| what_h | a |
| what_ha | p |
| what_hap | p |
| what_happ | e |

- Classification with 27 classes

# Text prediction: supervised learning

- Training data: lots of text
    - “what happens to a dream deferred”

| features | label |
|---|---|
| w | h |
| wh | a |
| wha | t |
| what | _ |
| what_ | h |
| what_h | a |
| what_ha | p |
| what_hap | p |
| what_happ | e |

- Classification with 27 classes
- How to featurize?

# Text prediction: supervised learning

- Training data: lots of text
  - “what happens to a dream deferred”

| features | label |
|---|---|
| w | h |
| wh | a |
| wha | t |
| what | _ |
| what_ | h |
| what_h | a |
| what_ha | p |
| what_hap | p |
| what_happ | e |

- Classification with 27 classes
- How to featurize?
- Idea: use all previous characters.

# Text prediction: supervised learning

- Training data: lots of text
  - “what happens to a dream deferred”

| **features** | **label** |
|---|---|
| w | h |
| wh | a |
| wha | t |
| what | _ |
| what_ | h |
| what_h | a |
| what_ha | p |
| what_hap | p |
| what_happ | e |

- Classification with 27 classes
- How to featurize?
- Idea: use all previous characters. But so far we’ve said $x^{(i)} \in \mathbb{R}^d$; i.e. fixed dimension

# Text prediction: supervised learning

- Training data: lots of text
  - “what happens to a dream deferred”

| **features** | **label** |
|---|---|
| w | h |
| wh | a |
| wha | t |
| what | _ |
| what_ | h |
| what_h | a |
| what_ha | p |
| what_hap | p |
| what_happ | e |

- Classification with 27 classes
- How to featurize?
- Idea: use all previous characters. But so far we’ve said $x^{(i)} \in \mathbb{R}^d$ ; i.e. fixed dimension

# Text prediction: supervised learning

- Training data: lots of text
  - “what happens to a dream deferred”

| features | label |
|---|---|
| w | h |
| wh | a |
| wha | t |
| what | _ |
| what_ | h |
| what_h | a |
| what_ha | p |
| what_hap | p |
| what_happ | e |

- Classification with 27 classes
- How to featurize?
- Idea: use all previous characters. But so far we’ve said $x^{(i)} \in \mathbb{R}^{d}$; i.e. fixed dimension

# Text prediction: supervised learning

- Training data: lots of text
  - “what happens to a dream deferred”

| features | label |
|---|---|
| w | h |
| wh | a |
| wha | t |
| what | _ |
| what_ | h |
| what_h | a |
| what_ha | p |
| what_hap | p |
| what_happ | e |

- Classification with 27 classes
- How to featurize?
- Idea: use all previous characters. But so far we’ve said $x^{(i)} \in \mathbb{R}^d$; i.e. fixed dimension
- Idea: just use last character. But lose info

# Text prediction: supervised learning

- Training data: lots of text
    - “what happens to a dream deferred”

| features | label |
|---|---|
| w | h |
| wh | a |
| wha | t |
| what | _ |
| what_ | h |
| what_h | a |
| what_ha | p |
| what_hap | p |
| what_happ | e |

- Classification with 27 classes
- How to featurize?
- Idea: use all previous characters. But so far we’ve said $x^{(i)} \in \mathbb{R}^d$; i.e. fixed dimension
- Idea: just use last character. But lose info
- Idea: use last *m* characters

# Text prediction: supervised learning

- Training data: lots of text
  - “what happens to a dream deferred”

| features | label |
|---|---|
| w | h |
| wh | a |
| wha | t |
| what | _ |
| what_ | h |
| what_h | a |
| what_ha | p |
| what_hap | p |
| what_happ | e |

- Classification with 27 classes
- How to featurize?
- Idea: use all previous characters. But so far we’ve said $x^{(i)} \in \mathbb{R}^d$; i.e. fixed dimension
- Idea: just use last character. But lose info
- Idea: use last $m = 3$ characters

# A state machine: writing & predicting text

“wha”

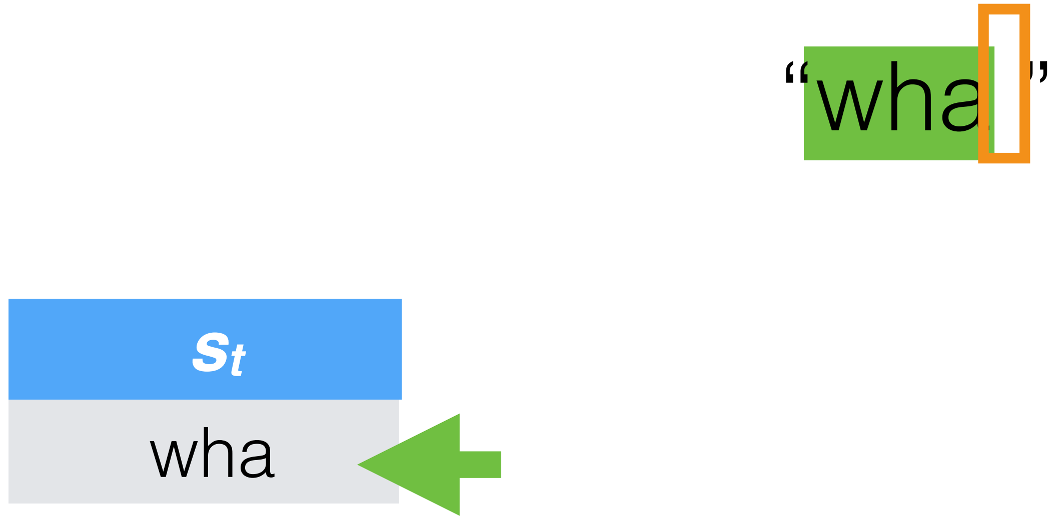
# A state machine: writing & predicting text

“wha▯”

# A state machine: writing & predicting text

“wha ”

# A state machine: writing & predicting text

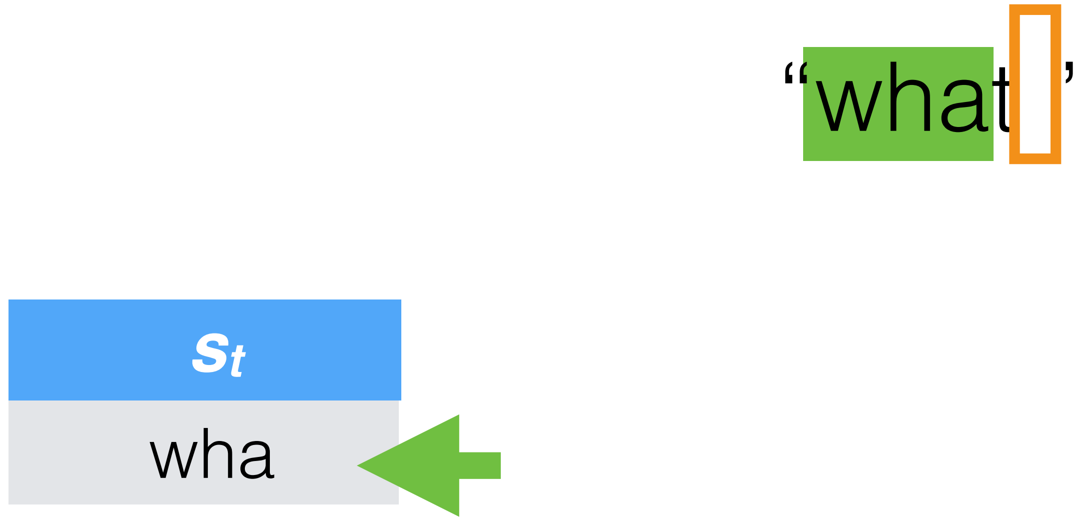


“wha ”

$s_t$

wha

# A state machine: writing & predicting text

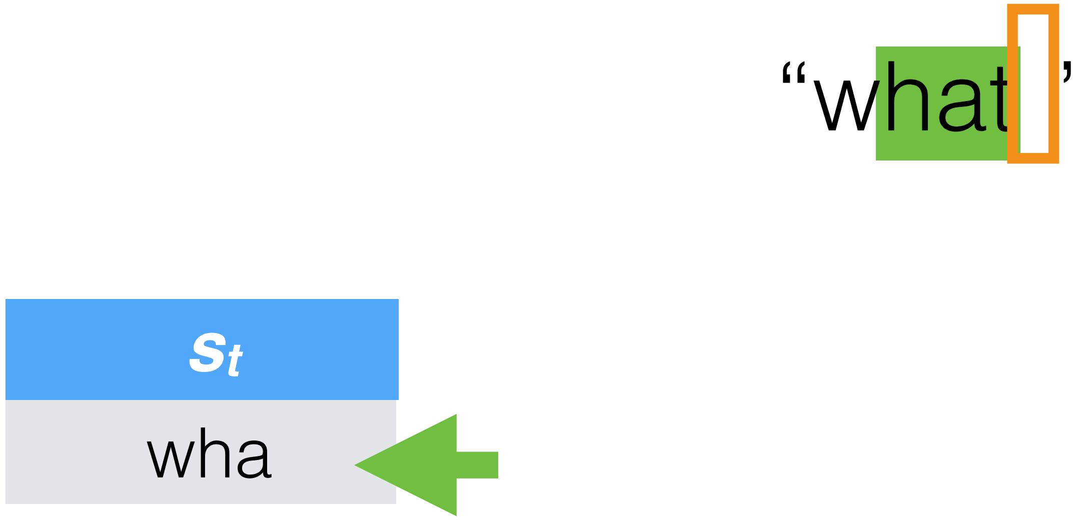


“what ”

$s_t$

wha

# A state machine: writing & predicting text



“what ”

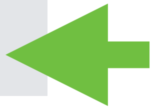
$s_t$

wha

# A state machine: writing & predicting text

“what”

| $x_t$ | $s_t$ |
|---|---|
|  | wha |



t

# A state machine: writing & predicting text

| $x_t$ | $s_t$ |
|---|---|
|  | wha |
| t | hat |

# A state machine: writing & predicting text

“what_ ”

| $x_t$ | $s_t$ |
|---|---|
|  | wha |
| t | hat |

# A state machine: writing & predicting text

“what_ ”

| $x_t$ | $s_t$ |
|---|---|
|  | wha |
| t | hat |

# A state machine: writing & predicting text

“what_ ”

| $x_t$ | $s_t$ |
|---|---|
|  | wha |
| t | hat |
| _ | at_ |

# A state machine: writing & predicting text

“what happens to a dream deferred”

| $x_t$ | $s_t$ |
|---|---|
| | wha |
| t | hat |
| _ | at_ |
| h | t_h |

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$

“what happens to a dream deferred”

| $x_t$ | $s_t$ |
|---|---|
| | wha |
| t | hat |
| _ | at_ |
| h | t_h |

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$

- Example:
  - Every sequence of 3 chars

“what happens to a dream deferred”

| $x_t$ | $s_t$ |
|---|---|
|  | wha |
| t | hat |
| _ | at_ |
| h | t_h |

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$

- Example:
  - Every sequence of 3 chars

“what happens to a dream deferred”

| $x_t$ | $s_t$ |
|---|---|
| | wha |
| t | hat |
| _ | at_ |
| h | t_h |

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
- Example:
  - Every sequence of 3 chars
  - Every single character

“what happens to a dream deferred”

| $x_t$ | $s_t$ |
|---|---|
| | wha |
| t | hat |
| _ | at_ |
| h | t_h |

# A state machine: writing & predicting text

- Recall state machines:
    - Set of possible states $\mathcal{S}$
    - Set of possible inputs $\mathcal{X}$
    - Initial state

- Example:
    - Every sequence of 3 chars
    - Every single character

“what happens to a dream deferred”

| $x_t$ | $s_t$ |
|---|---|
|  | wha |
| t | hat |
| _ | at_ |
| h | t_h |

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0 = \wedge\wedge\wedge$

“what happens to a dream deferred”

| $x_t$ | $s_t$ |
|---|---|
| | wha |
| t | hat |
| _ | at_ |
| h | t_h |

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$ = ^^^

| $x_t$ | $s_t$ |
|---|---|
|  | ^^^ |
| w | ^^w |
| h | ^wh |
| a | wha |
| t | hat |
| _ | at_ |

"what happens to a dream deferred"

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$ = ^^^

| $x_t$ | $s_t$ |
|---|---|
|  | ^^^ |
| w | ^^w |
| h | ^wh |
| a | wha |
| t | hat |
| _ | at_ |

“what happens to a dream deferred”

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$ = ^^^

| $x_t$ | $s_t$ |
|---|---|
|  | ^^^ |
| w | ^^w |
| h | ^wh |
| a | wha |
| t | hat |
| _ | at_ |

“what happens to a dream deferred”

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$ = ^^^

| $x_t$ | $s_t$ |
|---|---|
|  | ^^^ |
| w | ^^w |
| h | ^wh |
| a | wha |
| t | hat |
| _ | at_ |

“what happens to a dream deferred”

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$ = ^^^

| $x_t$ | $s_t$ |
|---|---|
| | ^^^ |
| w | ^^w |
| h | ^wh |
| a | wha |
| t | hat |
| _ | at_ |

"what happens to a dream deferred"

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$ = ^^^

| $x_t$ | $s_t$ |
|---|---|
| | ^^^ |
| w | ^^w |
| h | ^wh |
| a | wha |
| t | hat |
| _ | at_ |

“what happens to a dream deferred”

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$ = ^^^

| $t$ | $x_t$ | $s_t$ |
|---|---|---|
| 0 |  | ^^^ |
| 1 | w | ^^w |
| 2 | h | ^wh |
| 3 | a | wha |
| 4 | t | hat |
| 5 | _ | at_ |

“what happens to a dream deferred”

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$ = ^^^

| $t$ | $x_t$ | $s_t$ |
|---|---|---|
| 0 | | ^^^ |
| 1 | w | ^^w |
| 2 | h | ^wh |
| 3 | a | wha |
| 4 | t | hat |
| 5 | _ | at_ |

“what happens to a dream deferred”

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$ = ^^^
  - E.g. $f$(^^w,h) = ^wh

| $t$ | $x_t$ | $s_t$ |
|---|---|---|
| 0 | | ^^^ |
| 1 | w | ^^w |
| 2 | h | ^wh |
| 3 | a | wha |
| 4 | t | hat |
| 5 | _ | at_ |

"what happens to a dream deferred"

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$ = ^^^
  - E.g. $f$(^^w,h) = ^wh

| **$t$** | **$x_t$** | **$s_t$** |
|---|---|---|
| 0 | | ^^^ |
| 1 | w | ^^w |
| 2 | h | ^wh |
| 3 | a | wha |
| 4 | t | hat |
| 5 | _ | at_ |

“what happens to a dream deferred”

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$ = ^^^
  - E.g. $f$(^^w,h) = ^wh

| $t$ | $x_t$ | $s_t$ |
|---|---|---|
| 0 | | ^^^ |
| 1 | w | ^^w |
| 2 | h | ^wh |
| 3 | a | wha |
| 4 | t | hat |
| 5 | _ | at_ |

"what happens to a dream deferred"

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

previously “y”

| $t$ | $x_t$ | $s_t$ |
|---|---|---|
| 0 | | ^^^ |
| 1 | w | ^^w |
| 2 | h | ^wh |
| 3 | a | wha |
| 4 | t | hat |
| 5 | _ | at_ |

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$ = ^^^
  - E.g. $f$(^^w,h) = ^wh

“what happens to a dream deferred”

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

previously “y”

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$ = ^^^
  - E.g. $f$(^^w,h) = ^wh
  - Output: 27 prediction probs

| $t$ | $x_t$ | $s_t$ |
|---|---|---|
| 0 |  | ^^^ |
| 1 | w | ^^w |
| 2 | h | ^wh |
| 3 | a | wha |
| 4 | t | hat |
| 5 | _ | at_ |

“what happens to a dream deferred”

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

previously "y"

| $t$ | $x_t$ | $s_t$ |
|---|---|---|
| 0 | | ^^^ |
| 1 | w | ^^w |
| 2 | h | ^wh |
| 3 | a | wha |
| 4 | t | hat |
| 5 | _ | at_ |

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$ = ^^^
  - E.g. $f$(^^w,h) = ^wh
  - Output: 27 prediction probs
  - E.g. $p$=$g$(^^^) = [0.08, 0.02, ….., 0.001, 0.01]

"what happens to a dream deferred"

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

previously "y"

| $t$ | $x_t$ | $s_t$ |
|---|---|---|
| 0 | | ^^^ |
| 1 | w | ^^w |
| 2 | h | ^wh |
| 3 | a | wha |
| 4 | t | hat |
| 5 | _ | at_ |

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$ = ^^^
  - E.g. $f$(^^w,h) = ^wh
  - Output: 27 prediction probs
  - E.g. $p$=$g$(^^^) = [0.08, 0.02, …., 0.001, 0.01]

- $x^{(1)}$: "what happens to a dream deferred"

# A state machine: writing & predicting text

- Recall state machines:
    - Set of possible states $\mathcal{S}$
    - Set of possible inputs $\mathcal{X}$
    - Initial state
    - Transition function $f(s,x)$
    - Set of possible outputs
    - Output function $p = g(s)$

previously "y"

| $t$ | $x_t^{(1)}$ | $s_t^{(1)}$ |
|---|---|---|
| 0 | | ^^^ |
| 1 | w | ^^w |
| 2 | h | ^wh |
| 3 | a | wha |
| 4 | t | hat |
| 5 | _ | at_ |

- Example:
    - Every sequence of 3 chars
    - Every single character
    - $s_0$ = ^^^
    - E.g. $f$(^^w,h) = ^wh
    - Output: 27 prediction probs
    - E.g. $p$=$g$(^^^) = [0.08, 0.02, ...., 0.001, 0.01]

- $x^{(1)}$: "what happens to a dream deferred"

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

previously “y”

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$ = ^^^
  - E.g. $f$(^^w,h) = ^wh
  - Output: 27 prediction probs
  - E.g. $p$=$g$(^^^) = [0.08, 0.02, …., 0.001, 0.01]

| $t$ | $x_t^{(1)}$ | $s_t^{(1)}$ |
|---|---|---|
| 0 | | ^^^ |
| 1 | w | ^^w |
| 2 | h | ^wh |
| 3 | a | wha |
| 4 | t | hat |
| 5 | _ | at_ |

- $x^{(1)}$: “what happens to a dream deferred”
- $x^{(2)}$: “if you can keep your head when all about you”
- $x^{(3)}$: “you may write me down in history”

# A state machine: writing & predicting text

- Recall state machines:
    - Set of possible states $\mathcal{S}$
    - Set of possible inputs $\mathcal{X}$
    - Initial state
    - Transition function $f(s,x)$
    - Set of possible outputs
    - Output function $p = g(s)$

- Example:
    - Every sequence of 3 chars
    - Every single character
    - $s_0$ = ^^^
    - E.g. $f$(^^w,h) = ^wh
    - Output: 27 prediction probs
    - E.g. $p$=$g$(^^^) = [0.08, 0.02, …., 0.001, 0.01]

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$ = ^^^
  - E.g. $f$(^^w,h) = ^wh
  - Output: 27 prediction probs
  - E.g. $p$=$g$(^^^) = [0.08, 0.02, …., 0.001, 0.01]

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$ = ^^^
  - E.g. $f$(^^w,h) = ^wh
  - Output: 27 prediction probs
  - E.g. $p$=$g$(^^^) = [0.08, 0.02, …., 0.001, 0.01]

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$ = ^^^
  - E.g. $f$(^^w,h) = ^wh
  - Output: 27 prediction probs
  - E.g. $p$=$g$(^^^) = [0.08, 0.02, …., 0.001, 0.01]

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$ = ^^^
  - E.g. $f$(^^w,h) = ^wh
  - Output: 27 prediction probs
  - E.g. $p$=$g$(^^^) = [0.08, 0.02, …., 0.001, 0.01]

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: 27 prediction probs
  - E.g. $p=g(\wedge\wedge\wedge) = [0.08, 0.02, 0.001, 0.01]$

- Example with alphabet {0,1} ($v=2$); state is last $m = 3$ chars

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: 27 prediction probs
  - E.g. $p=g(\wedge\wedge\wedge) = [0.08, 0.02,$
    $0.001, 0.01]$

- Example with alphabet {0,1} ($v=2$); state is last $m$ = 3 chars

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p=g(\wedge\wedge\wedge) = [0.08, 0.02, \ldots, 0.001, 0.01]$

- Example with alphabet {0,1} ($v=2$); state is last $m = 3$ chars

# A state machine: writing & predicting text

- Recall state machines:
    - Set of possible states $\mathcal{S}$
    - Set of possible inputs $\mathcal{X}$
    - Initial state
    - Transition function $f(s,x)$
    - Set of possible outputs
    - Output function $p = g(s)$
- Example:
    - Every sequence of 3 chars
    - Every single character
    - $s_0$
    - E.g. $f\left( \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x \right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
    - Output: $v$ prediction probs
    - E.g. $p = g(\wedge\wedge\wedge) = [0.08, 0.02,$ …, $0.001, 0.01]$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$
- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p=g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$
- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$
- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p=g(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix})$
- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$s_t = f(s_{t-1}, x_t) =$$

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p=g(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix})$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$s_t = f(s_{t-1}, x_t) =$$

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$s_t = f(s_{t-1}, x_t) = \boxed{?}\; x_t + \boxed{?}\; s_{t-1}$$

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$s_t = f(s_{t-1}, x_t) = \boxed{?}\; x_t + \boxed{?}\; s_{t-1}$$

3x1

# A state machine: writing & predicting text

- Recall state machines:
    - Set of possible states $\mathcal{S}$
    - Set of possible inputs $\mathcal{X}$
    - Initial state
    - Transition function *f*(*s*,*x*)
    - Set of possible outputs
    - Output function *p* = *g*(*s*)
- Example:
    - Every sequence of 3 chars
    - Every single character
    - $s_0$
    - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
    - Output: *v* prediction probs
    - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$
- Example with alphabet {0,1} (*v*=2); state is last *m* = 3 chars

$$\underset{3\text{x}1}{s_t} = f(s_{t-1}, x_t) = \boxed{?}\ \underset{1\text{x}1}{x_t} + \boxed{?}\ s_{t-1}$$

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$\underset{3\text{x}1}{s_t} = f(s_{t-1}, x_t) = \boxed{?}\; \underset{1\text{x}1}{x_t} + \boxed{?}\; \underset{3\text{x}1}{s_{t-1}}$$

# A state machine: writing & predicting text

- Recall state machines:
    - Set of possible states $\mathcal{S}$
    - Set of possible inputs $\mathcal{X}$
    - Initial state
    - Transition function $f(s,x)$
    - Set of possible outputs
    - Output function $p = g(s)$
- Example:
    - Every sequence of 3 chars
    - Every single character
    - $s_0$
    - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
    - Output: $v$ prediction probs
    - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$
- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$\underset{3\text{x}1}{s_t} = f(s_{t-1}, x_t) = \boxed{?}\ \underset{1\text{x}1}{x_t} + \boxed{?}\ \underset{3\text{x}1}{s_{t-1}}$$

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$\underset{3x1}{s_t} = f(s_{t-1}, x_t) = \underset{3x1}{\boxed{?}}\ \underset{1x1}{x_t} + \boxed{\quad ? \quad}\ \underset{3x1}{s_{t-1}}$$

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$\underset{3\text{x}1}{s_t} = f(s_{t-1}, x_t) = \underset{3\text{x}1}{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}} \underset{1\text{x}1}{x_t} + \boxed{?} \underset{3\text{x}1}{s_{t-1}}$$

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p=g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$\underset{3\text{x}1}{s_t} = f(s_{t-1}, x_t) = \underset{3\text{x}1}{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}} \underset{1\text{x}1}{x_t} + \boxed{\quad ? \quad} \underset{3\text{x}1}{s_{t-1}}$$

# A state machine: writing & predicting text

- Recall state machines:
    - Set of possible states $\mathcal{S}$
    - Set of possible inputs $\mathcal{X}$
    - Initial state
    - Transition function $f(s,x)$
    - Set of possible outputs
    - Output function $p = g(s)$
- Example:
    - Every sequence of 3 chars
    - Every single character
    - $s_0$
    - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
    - Output: $v$ prediction probs
    - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$\underset{\text{3x1}}{s_t} = f(s_{t-1}, x_t) = \underset{\text{3x1}}{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}} \underset{\text{1x1}}{x_t} + \underset{\text{3x3}}{\boxed{?}} \underset{\text{3x1}}{s_{t-1}}$$

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$\underset{3\text{x}1}{s_t} = f(s_{t-1}, x_t) = \underset{3\text{x}1}{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}} \underset{1\text{x}1}{x_t} + \underset{3\text{x}3}{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}} \underset{3\text{x}1}{s_{t-1}}$$

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix}c_1\\c_2\\c_3\end{bmatrix}, x\right) = \begin{bmatrix}c_2\\c_3\\x\end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p = g\left(\begin{bmatrix}c_1\\c_2\\c_3\end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$\underset{3\text{x}1}{s_t} = f(s_{t-1}, x_t) = \underset{3\text{x}1}{\begin{bmatrix}0\\0\\1\end{bmatrix}} \underset{1\text{x}1}{x_t} + \underset{3\text{x}3}{\begin{bmatrix}0&1&0\\0&0&1\\0&0&0\end{bmatrix}} \underset{3\text{x}1}{s_{t-1}}$$

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix}c_1\\c_2\\c_3\end{bmatrix}, x\right) = \begin{bmatrix}c_2\\c_3\\x\end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p = g\left(\begin{bmatrix}c_1\\c_2\\c_3\end{bmatrix}\right)$

- Example with alphabet $\{0,1\}$ ($v=2$); state is last $m = 3$ chars

$$\underset{3\text{x}1}{s_t} = f(s_{t-1}, x_t) = \underset{3\text{x}1}{\begin{bmatrix}0\\0\\1\end{bmatrix}} \underset{1\text{x}1}{x_t} + \underset{3\text{x}3}{\begin{bmatrix}0&1&0\\0&0&1\\0&0&0\end{bmatrix}} \underset{3\text{x}1}{s_{t-1}}$$

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function *f*(*s*,*x*)
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix}c_1\\c_2\\c_3\end{bmatrix}, x\right) = \begin{bmatrix}c_2\\c_3\\x\end{bmatrix}$
  - Output: *v* prediction probs
  - E.g. $p = g\left(\begin{bmatrix}c_1\\c_2\\c_3\end{bmatrix}\right)$

- Example with alphabet {0,1} (*v*=2); state is last *m* = 3 chars

$$\underset{\text{3x1}}{s_t} = f(s_{t-1}, x_t) = \underset{\text{3x1}}{\begin{bmatrix}0\\0\\1\end{bmatrix}} \underset{\text{1x1}}{x_t} + \underset{\text{3x3}}{\begin{bmatrix}0 & 1 & 0\\0 & 0 & 1\\0 & 0 & 0\end{bmatrix}} \underset{\text{3x1}}{s_{t-1}}$$

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$\underset{\text{3x1}}{s_t} = f(s_{t-1}, x_t) = \underset{\text{3x1}}{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}} \underset{\text{1x1}}{x_t} + \underset{\text{3x3}}{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}} \underset{\text{3x1}}{s_{t-1}}$$

$$p_t = g(s_t)$$

$$=$$

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p=g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$\underset{3\text{x}1}{s_t} = f(s_{t-1}, x_t) = \underset{3\text{x}1}{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}} \underset{1\text{x}1}{x_t} + \underset{3\text{x}3}{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}} \underset{3\text{x}1}{s_{t-1}}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W^o_0) \end{aligned}$$

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$\underset{3\text{x}1}{s_t} = f(s_{t-1}, x_t) = \underset{3\text{x}1}{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}} \underset{1\text{x}1}{x_t} + \underset{3\text{x}3}{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}} \underset{3\text{x}1}{s_{t-1}}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W^o_0) \end{aligned}$$

same as final layer in all of our neural networks so far, except for notation

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: *v* prediction probs
  - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} (*v*=2); state is last *m* = 3 chars

$$\underset{3\text{x}1}{s_t} = f(s_{t-1}, x_t) = \underset{3\text{x}1}{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}} \underset{1\text{x}1}{x_t} + \underset{3\text{x}3}{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}} \underset{3\text{x}1}{s_{t-1}}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W^o_0) \end{aligned}$$

same as final layer in all of our neural networks so far, except for notation

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$
- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$\underset{3\text{x}1}{s_t} = f(s_{t-1}, x_t) = \underset{3\text{x}1}{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}} \underset{1\text{x}1}{x_t} + \underset{3\text{x}3}{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}} \underset{3\text{x}1}{s_{t-1}}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

same as final layer in all of our neural networks so far, except for notation

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p=g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$\underset{\text{3x1}}{s_t} = f(s_{t-1}, x_t) = \underset{\text{3x1}}{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}} \underset{\text{1x1}}{x_t} + \underset{\text{3x3}}{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}} \underset{\text{3x1}}{s_{t-1}}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

same as final layer in all of our neural networks so far, except for notation

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$
- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$
- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} x_t + \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W^o_0) \end{aligned}$$

same as final layer in all of our neural networks so far, except for notation

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function *f*(*s*,*x*)
  - Set of possible outputs
  - Output function *p* = *g*(*s*)

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: *v* prediction probs
  - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} (*v*=2); state is last *m* = 3 chars

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} x_t + \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

same as final layer in all of our neural networks so far, except for notation

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix}c_1\\c_2\\c_3\end{bmatrix}, x\right) = \begin{bmatrix}c_2\\c_3\\x\end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p=g\left(\begin{bmatrix}c_1\\c_2\\c_3\end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix}0\\0\\1\end{bmatrix} x_t + \begin{bmatrix}0&1&0\\0&0&1\\0&0&0\end{bmatrix} s_{t-1}$$

$$\begin{aligned}p_t &= g(s_t)\\ &= f_2(W^o s_t + W^o_0)\end{aligned}$$

same as final layer in all of our neural networks so far, except for notation

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p=g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} x_t + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

same as final layer in all of our neural networks so far, except for notation

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p=g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} x_t + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

same as final layer in all of our neural networks so far, except for notation

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p=g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} x_t + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

same as final layer in all of our neural networks so far, except for notation

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p=g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$s_t = f(s_{t-1}, x_t) = \begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} x_t + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

same as final layer in all of our neural networks so far, except for notation

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$
- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$
- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$s_t = \begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} x_t + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} s_{t-1}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

same as final layer in all of our neural networks so far, except for notation

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p=g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$s_t = \begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} x_t + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} s_{t-1} + \begin{bmatrix} w_{0,1}^{ss} \\ w_{0,2}^{ss} \\ w_{0,3}^{ss} \end{bmatrix}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

same as final layer in all of our neural networks so far, except for notation

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$
- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$
- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$s_t = \begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} x_t + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} s_{t-1} + \begin{bmatrix} w_{0,1}^{ss} \\ w_{0,2}^{ss} \\ w_{0,3}^{ss} \end{bmatrix}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

same as final layer in all of our neural networks so far, except for notation

# A state machine: writing & predicting text

- Recall state machines:
    - Set of possible states $\mathcal{S}$
    - Set of possible inputs $\mathcal{X}$
    - Initial state
    - Transition function $f(s,x)$
    - Set of possible outputs
    - Output function $p = g(s)$

- Example:
    - Every sequence of 3 chars
    - Every single character
    - $s_0$
    - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
    - Output: $v$ prediction probs
    - E.g. $p=g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$s_t = \begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} x_t + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} s_{t-1} + \begin{bmatrix} w_{0,1}^{ss} \\ w_{0,2}^{ss} \\ w_{0,3}^{ss} \end{bmatrix}$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

same as final layer in all of our neural networks so far, except for notation

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$
- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$
- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$s_t = f_1\left(\begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} x_t + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} s_{t-1} + \begin{bmatrix} w_{0,1}^{ss} \\ w_{0,2}^{ss} \\ w_{0,3}^{ss} \end{bmatrix}\right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

same as final layer in all of our neural networks so far, except for notation

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible in ts $\mathcal{X}$
  - Initial state
  - Transition f
  - Set of p ts
  - Outp = $g(s)$

componentwise; activation functions

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example r alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$s_t = f_1\left(\begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} x_t + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} s_{t-1} + \begin{bmatrix} w_{0,1}^{ss} \\ w_{0,2}^{ss} \\ w_{0,3}^{ss} \end{bmatrix}\right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

same as final layer in all of our neural networks so far, except for notation

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function *f*(*s*,*x*)
  - Set of possible outputs
  - Output function *p* = *g*(*s*)
- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: *v* prediction probs
  - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$
- Example with alphabet {0,1} (*v*=2); state is last *m* = 3 chars

$$s_t = f_1\left(\begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} x_t + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} s_{t-1} + \begin{bmatrix} w_{0,1}^{ss} \\ w_{0,2}^{ss} \\ w_{0,3}^{ss} \end{bmatrix}\right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

same as final layer in all of our neural networks so far, except for notation

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function *f*(*s*,*x*)
  - Set of possible outputs
  - Output function *p* = *g*(*s*)

- Example:
  - Every sequence of 3 chars
  - Every single character
  - *s₀*
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: *v* prediction probs
  - E.g. $p=g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} (*v*=2); state is last *m* = 3 chars

$$s_t = f_1\left(\begin{bmatrix} w_1^{sx} \\ w_2^{sx} \\ w_3^{sx} \end{bmatrix} x_t + \begin{bmatrix} w_{11}^{ss} & w_{12}^{ss} & w_{13}^{ss} \\ w_{21}^{ss} & w_{22}^{ss} & w_{23}^{ss} \\ w_{31}^{ss} & w_{32}^{ss} & w_{33}^{ss} \end{bmatrix} s_{t-1} + \begin{bmatrix} w_{0,1}^{ss} \\ w_{0,2}^{ss} \\ w_{0,3}^{ss} \end{bmatrix}\right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

same as final layer in all of our neural networks so far, except for notation

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$s_t = f_1\left(W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss}\right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

same as final layer in all of our neural networks so far, except for notation

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$
- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$
- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$s_t = f_1\left(W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss}\right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

same as final layer in all of our neural networks so far, except for notation

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$
- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p=g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$
- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$s_t = f_1\left(W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}\right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p=g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$\left.\begin{aligned} s_t &= f_1\left(W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}\right) \\ \\ p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}\right\}$$

**Put it all together:**

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function $f(s,x)$
  - Set of possible outputs
  - Output function $p = g(s)$

- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: $v$ prediction probs
  - E.g. $p=g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$

- Example with alphabet {0,1} ($v$=2); state is last $m$ = 3 chars

$$\left.\begin{aligned} s_t &= f_1\left(W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss}\right) \\ \\ p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}\right\}$$

**Put it all together:**

$$p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of possible inputs $\mathcal{X}$
  - Initial state
  - Transition function *f*(*s*,*x*)
  - Set of possible outputs
  - Output function $p = g(s)$
- Example:
  - Every sequence of 3 chars
  - Every single character
  - $s_0$
  - E.g. $f\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, x\right) = \begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$
  - Output: *v* prediction probs
  - E.g. $p = g\left(\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}\right)$
- Example with alphabet {0,1} (*v*=2); state is last *m* = 3 chars

$$s_t = f_1\left(W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}\right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

Put it all together:

$$p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

"recurrent neural network"

# A state machine: writing & predicting text

- Recall state machines:
  - Set of possible states $\mathcal{S}$
  - Set of po
  - Initial sta
  - Transition
  - Set of po
  - Output fu

- Example:
  - Every sequence of 3 chars
  - racter

$\begin{bmatrix} c_2 \\ c_3 \\ x \end{bmatrix}$

tion probs

Recall:

- In 2-layer, fully-connected NNs, we learned the features.
- In CNNs, we learned the filters.
- Analogous idea here.

- Example with alphabet {0, 1} ($V$=2), state is last $m$ = 3 chars

$$s_t = f_1\left(W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}\right)$$

$$\begin{aligned} p_t &= g(s_t) \\ &= f_2(W^o s_t + W_0^o) \end{aligned}$$

Put it all together:

$$p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$$

"recurrent neural network"

# Recurrent Neural Networks

# Recurrent Neural Networks

Recall: familiar pattern

# Recurrent Neural Networks

Recall: familiar pattern

1. Choose how to predict label (given features & parameters)

# Recurrent Neural Networks

Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)

# Recurrent Neural Networks

Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

# Recurrent Neural Networks

Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

In our example application:

# Recurrent Neural Networks

Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

In our example application:

1. Use an RNN: $p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$

# Recurrent Neural Networks

Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

In our example application:

1. Use an RNN: $p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$

$$s_t = f_1\left(W^{sx}x_t + W^{ss}s_{t-1} + W_0^{ss}\right)$$

# Recurrent Neural Networks

Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

In our example application:

1. Use an RNN: $p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$

$$s_t = f_1\left(W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss}\right)$$
$$p_t = f_2(W^o s_t + W_0^o)$$

# Recurrent Neural Networks

Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

In our example application:

1. Use an RNN: $p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$

$$s_t = f_1\left(W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss}\right)$$

$$p_t = f_2(W^o s_t + W_0^o)$$

2. Loss: for *q* sequences, where the *i*th has length $n^{(i)}$

# Recurrent Neural Networks

Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

In our example application:

1. Use an RNN: $p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$

$$s_t = f_1\left(W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss}\right)$$

$$p_t = f_2(W^o s_t + W_0^o)$$

2. Loss: for *q* sequences, where the *i*th has length $n^{(i)}$

$$L_{\text{seq}}(p^{(i)}, y^{(i)}) = \sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)})$$

# Recurrent Neural Networks

Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

In our example application:

1. Use an RNN: $p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$

$$s_t = f_1\left(W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss}\right)$$

$$p_t = f_2(W^o s_t + W_0^o)$$

2. Loss: for *q* sequences, where the *i*th has length $n^{(i)}$

$$L_{\mathrm{seq}}(p^{(i)}, y^{(i)}) = \sum_{t=1}^{n^{(i)}} L_{\mathrm{elt}}(p_t^{(i)}, y_t^{(i)})$$

element

# Recurrent Neural Networks

Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

In our example application:

1. Use an RNN: $p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$

$$s_t = f_1\left(W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss}\right)$$

$$p_t = f_2(W^o s_t + W_0^o)$$

2. Loss: for *q* sequences, where the *i*th has length $n^{(i)}$

sequence $L_{\text{seq}}(p^{(i)}, y^{(i)}) = \sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)})$ element

# Recurrent Neural Networks

Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

In our example application:

1. Use an RNN: $p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$

$$s_t = f_1\left(W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss}\right)$$

$$p_t = f_2(W^o s_t + W_0^o)$$

2. Loss: for *q* sequences, where the *i*th has length $n^{(i)}$

sequence $L_{\text{seq}}(p^{(i)}, y^{(i)}) = \sum_{t=1}^{n^{(i)}} L_{\text{elt}}(p_t^{(i)}, y_t^{(i)})$ element

$$J(W, W_0) = \sum_{i=1}^{q} L_{\text{seq}}\left(p^{(i)}, y^{(i)}\right)$$

# Recurrent Neural Networks

Recall: familiar pattern

1. Choose how to predict label (given features & parameters)
2. Choose a loss (between guess & actual label)
3. Choose parameters by trying to minimize the training loss

In our example application:

1. Use an RNN: $p^{(i)} = \mathrm{RNN}(x^{(i)}; W, W_0)$

$$s_t = f_1\left(W^{sx} x_t + W^{ss} s_{t-1} + W_0^{ss}\right)$$

$$p_t = f_2(W^o s_t + W_0^o)$$

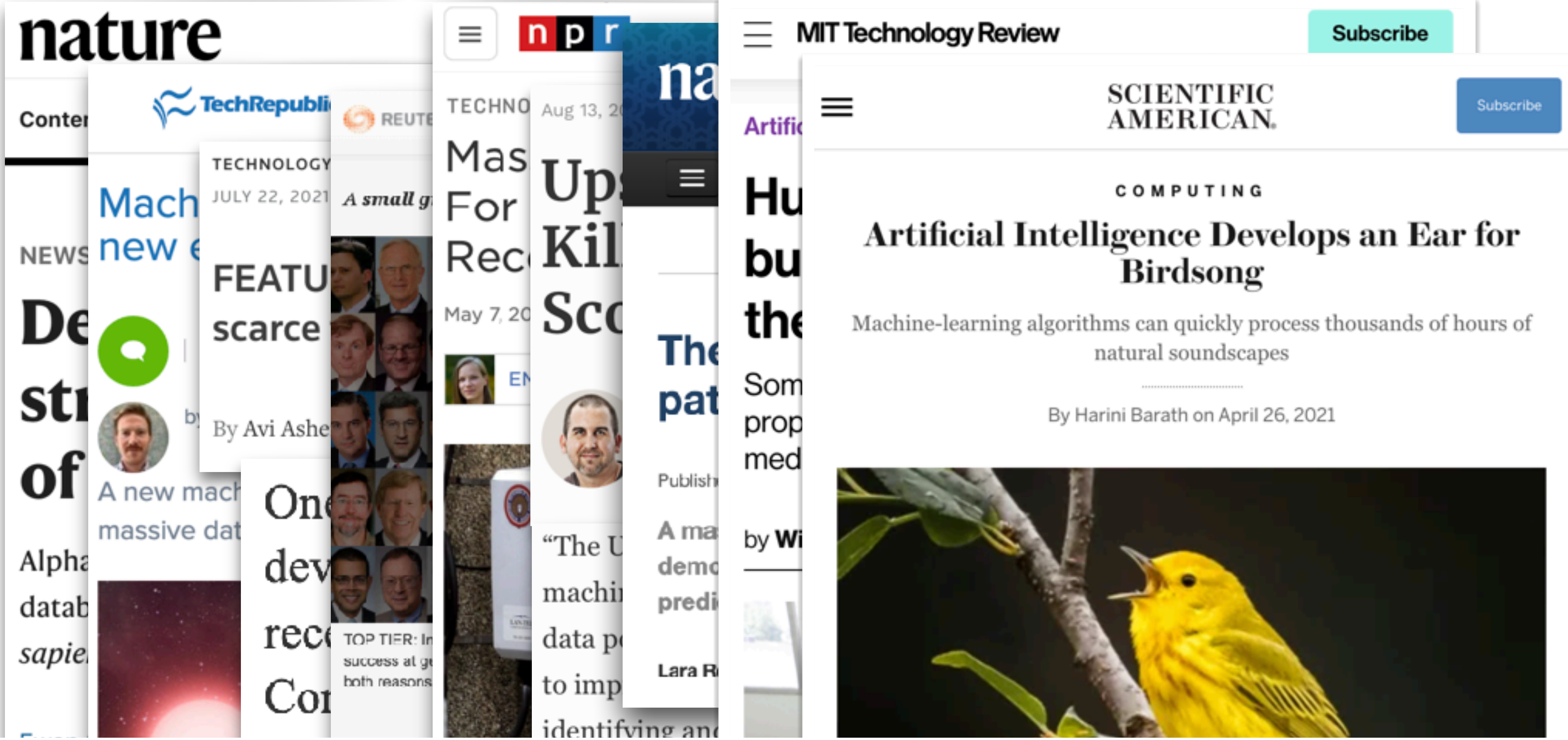
2. Loss: for *q* sequences, where the *i*th has length $n^{(i)}$

sequence $$L_{\mathrm{seq}}(p^{(i)}, y^{(i)}) = \sum_{t=1}^{n^{(i)}} L_{\mathrm{elt}}(p_t^{(i)}, y_t^{(i)})$$ element

$$J(W, W_0) = \sum_{i=1}^{q} L_{\mathrm{seq}}\left(p^{(i)}, y^{(i)}\right)$$

3. Stochastic gradient descent

# Machine learning (ML): why & what



SCIENTIFIC AMERICAN.

Subscribe

COMPUTING

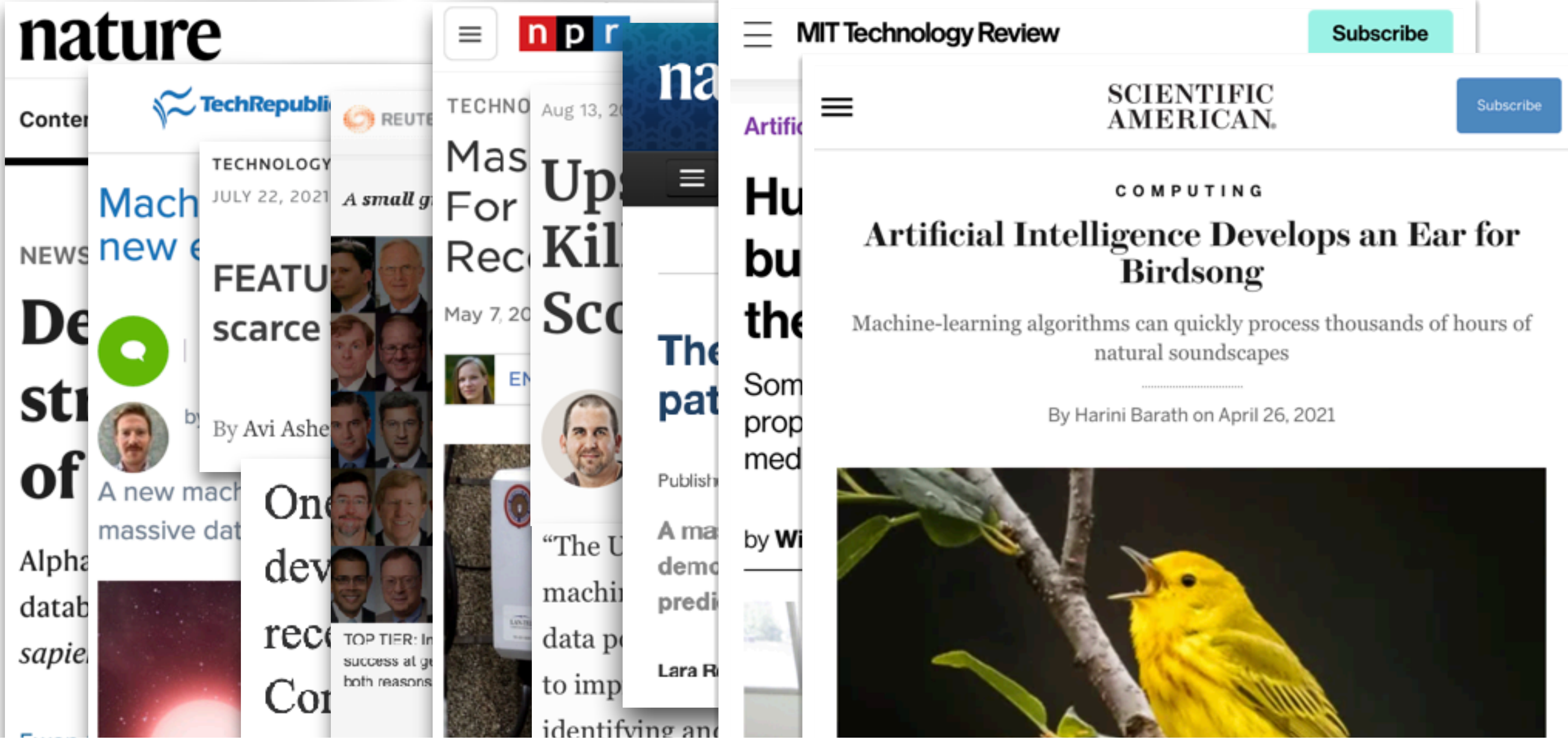
Artificial Intelligence Develops an Ear for Birdsong

Machine-learning algorithms can quickly process thousands of hours of natural soundscapes

By Harini Barath on April 26, 2021

- **What is ML?**

# Machine learning (ML): why & what



SCIENTIFIC AMERICAN

Subscribe

COMPUTING

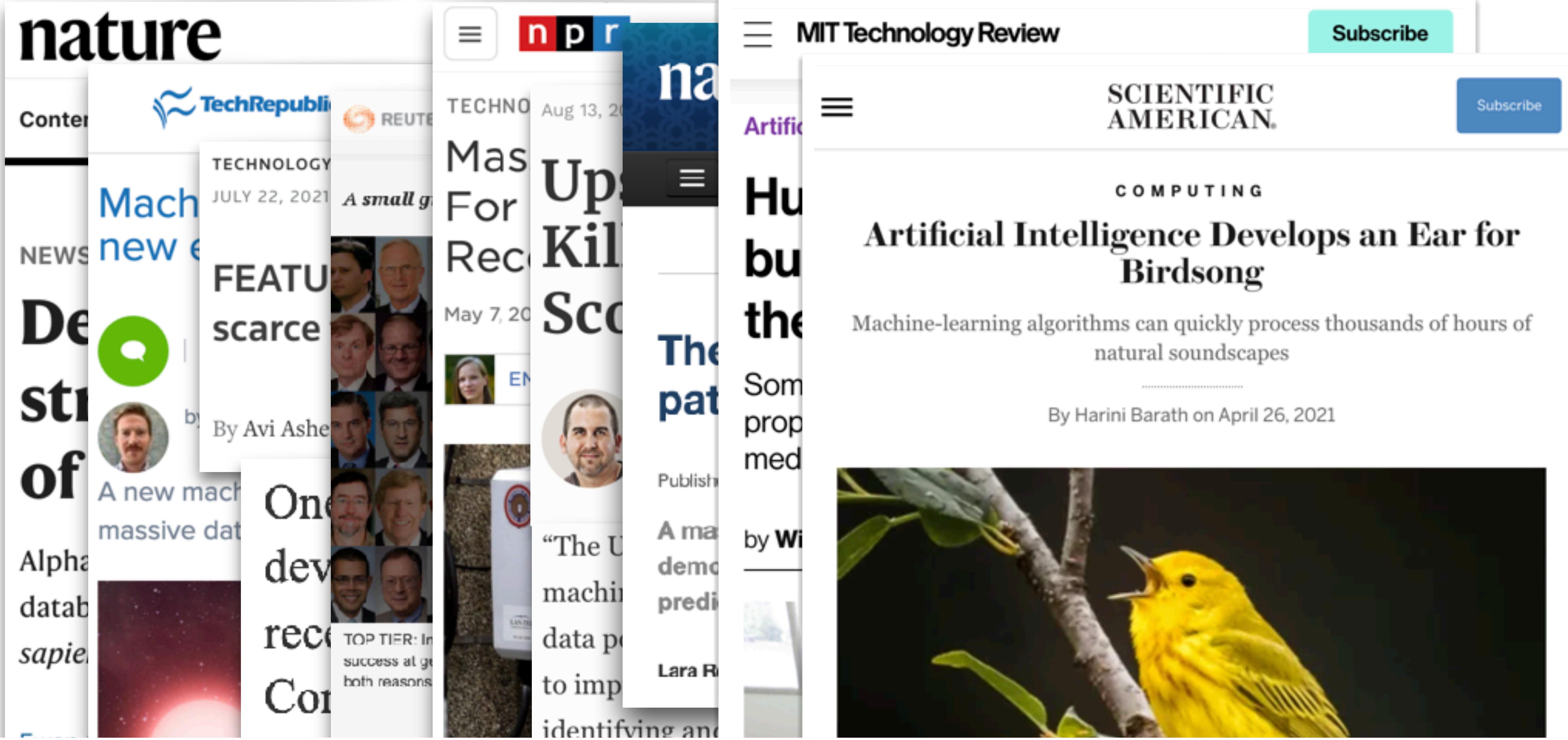
Artificial Intelligence Develops an Ear for Birdsong

Machine-learning algorithms can quickly process thousands of hours of natural soundscapes

By Harini Barath on April 26, 2021

- **What is ML?** A set of methods for making decisions from data. (See the rest of the course!)

# Machine learning (ML): why & what



SCIENTIFIC AMERICAN

Subscribe

COMPUTING

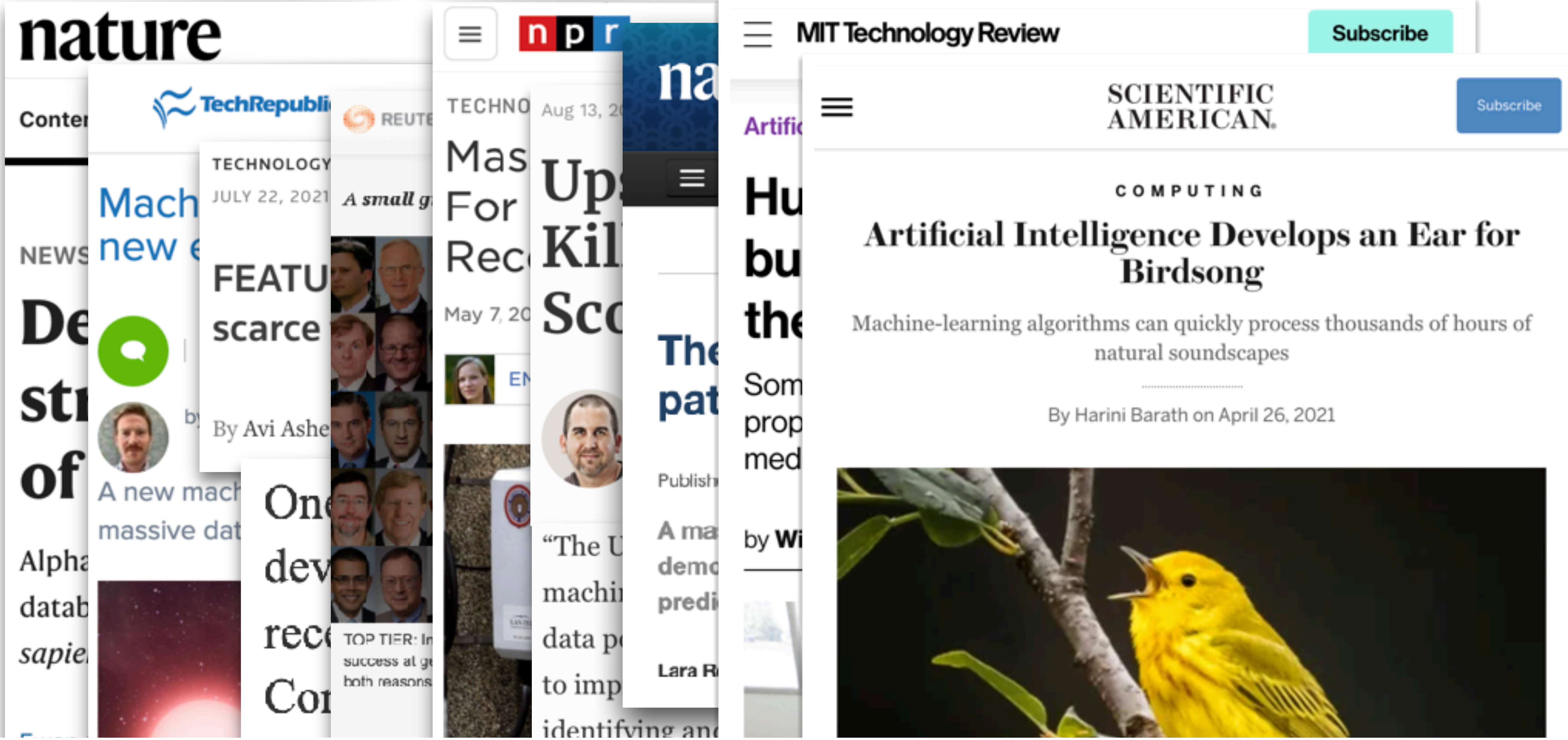
**Artificial Intelligence Develops an Ear for Birdsong**

Machine-learning algorithms can quickly process thousands of hours of natural soundscapes

By Harini Barath on April 26, 2021

- **What is ML?** A set of methods for making decisions from data. (See the rest of the course!)
- **Why study ML?** To apply; to understand; to evaluate

# Machine learning (ML): why & what



SCIENTIFIC AMERICAN

Subscribe

COMPUTING

Artificial Intelligence Develops an Ear for Birdsong

Machine-learning algorithms can quickly process thousands of hours of natural soundscapes

By Harini Barath on April 26, 2021

- **What is ML?** A set of methods for making decisions from data. (See the rest of the course!)
- **Why study ML?** To apply; to understand; to evaluate
- **Notes:** ML is a tool with pros & cons. ML is built on math

MIT EECS
ELECTRICAL ENGINEERING
AND COMPUTER SCIENCE

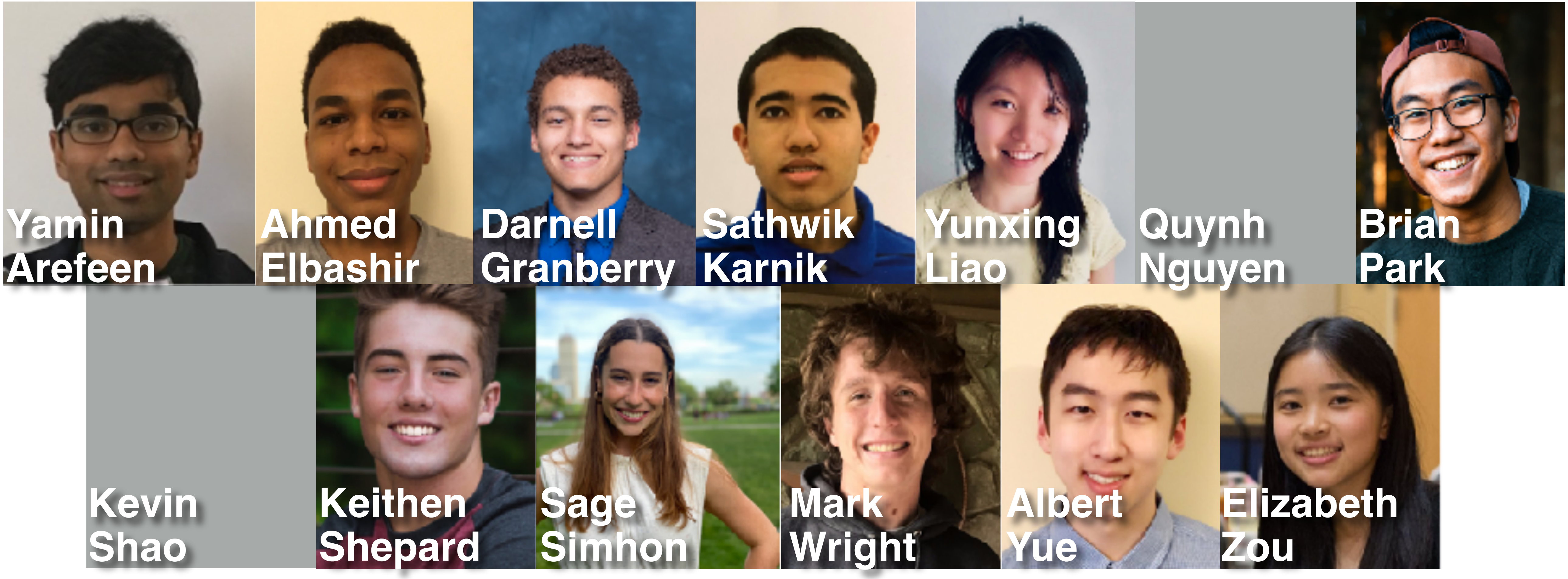
# 6.036: Staff

Big thanks to all of our staff and to the MIT AV team!

Instructors:

Jehangir Amjad
Tamara Broderick
Ike Chuang
Iddo Drori
David Sontag
Tess Smidt

Teaching Assistants:



Yamin Arefeen
Ahmed Elbashir
Darnell Granberry
Sathwik Karnik
Yunxing Liao
Quynh Nguyen
Brian Park
Kevin Shao
Keithen Shepard
Sage Simhon
Mark Wright
Albert Yue
Elizabeth Zou

And Lab Assistants!

# Thank you!