

Announcements

- Pset4 out, due Thursday, April 10th
- Challenge will be released soon.
- No laptops during class.

Last time

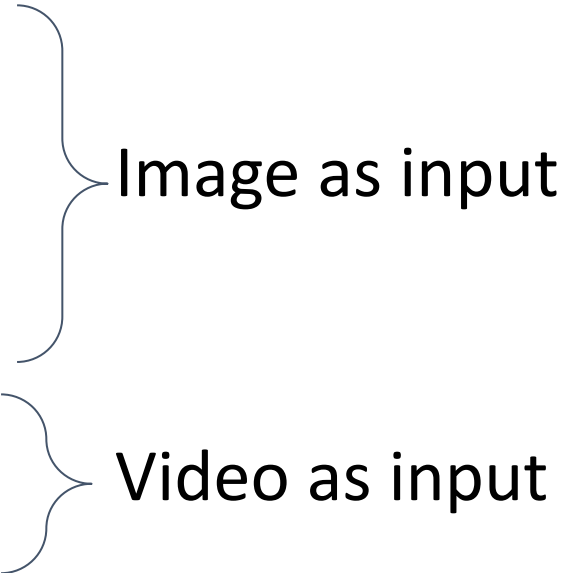
- Benefits of CNNs
- Examples of very popular CNNs
 - LeNet
 - CIFAR-10
 - AlexNet
 - VGG-Net
 - ResNet

Image as input

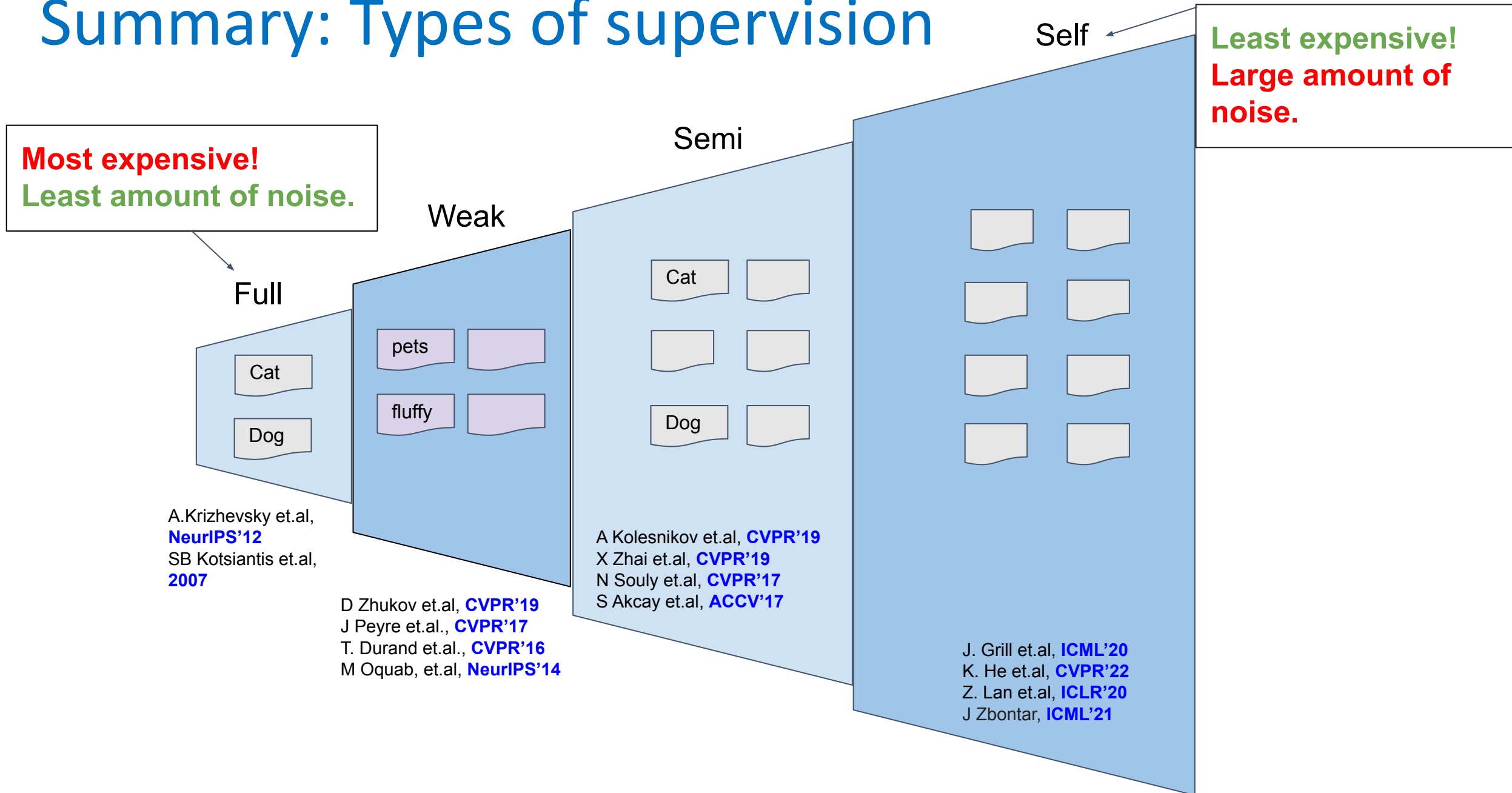
 - R3D
 - R(2+1)D

Video as input

Last time

- Some practical tips while training your model.
- Forms of supervision
- Pre-training v/s fine tuning

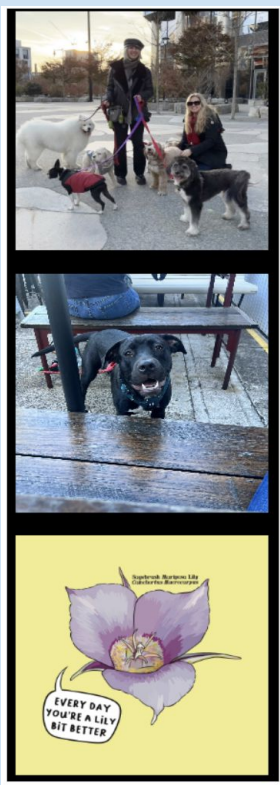
Summary: Types of supervision



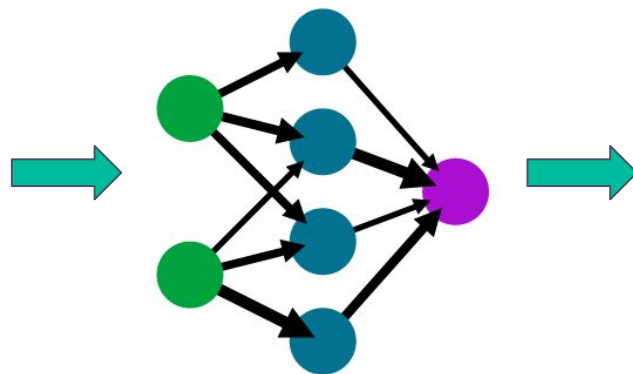
Overall Approach

Weakly supervised

Pre-training data (65M)



R(2+1)D video
architecture [1]



Fully-supervised

Fine-tune on target datasets

Kinetics [3]

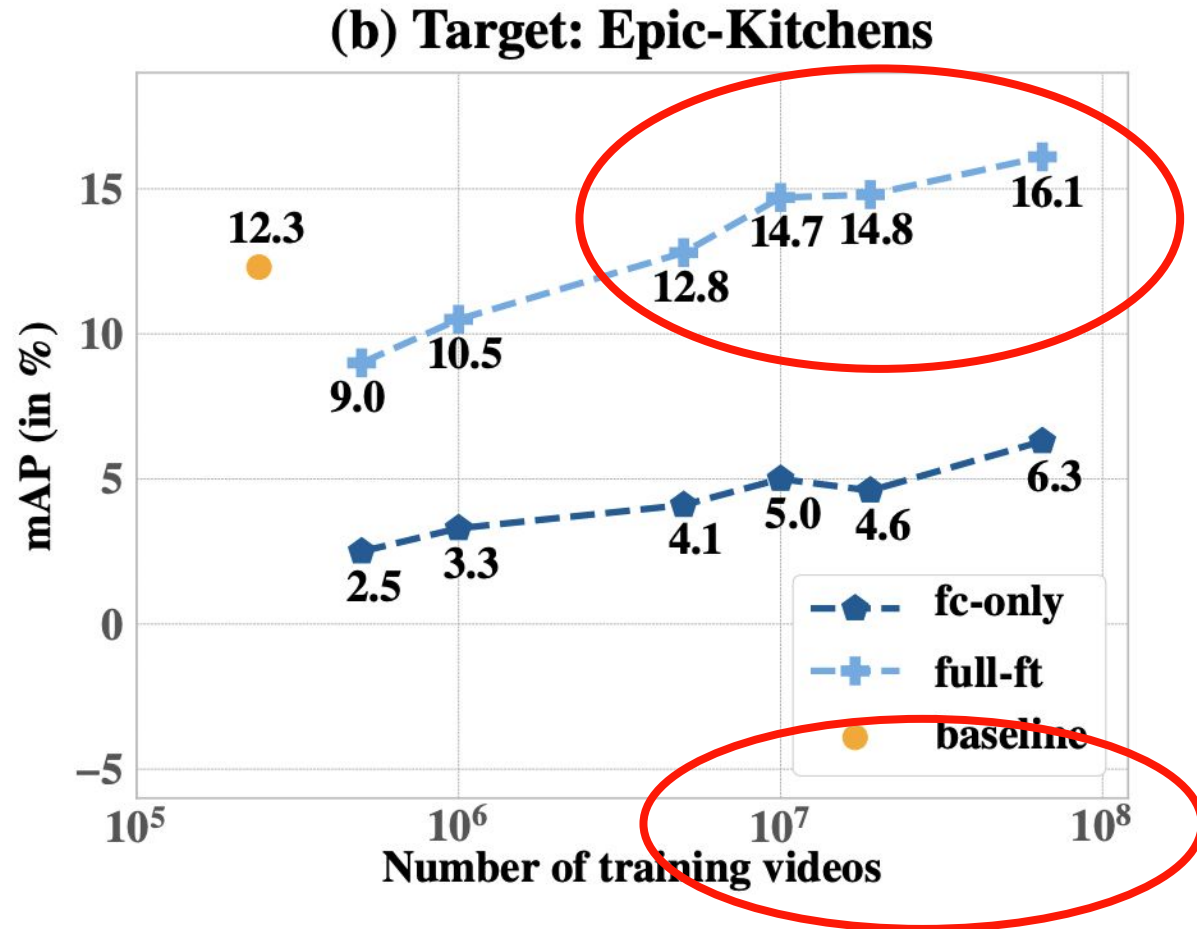


Epic Kitchens [2]



- [1] Tran et.al, CVPR'18
- [2] Damen et.al., ECCV'18
- [3] Carreira et.al

Pre-training data noise v/s volume



Key takeaway: Weaker forms of supervision **benefit only at scale.**

Data volume v/s model capacity

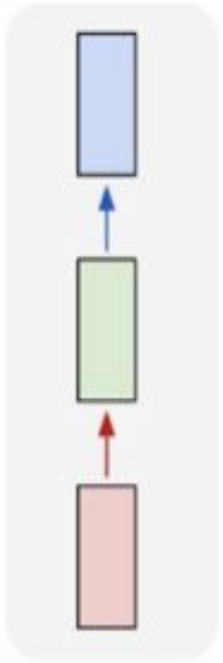
Pre-training data = 65M

			Kinetics	
Models	GFLOPS	# params	full-ft	baseline
R(2+1)D-18	83	33M	76.0	69.3
R(2+1)D-34	152	64M	78.2	69.6
R(2+1)D-101	176	86M	79.1	71.7
R(2+1)D-152	252	118M	79.9	72.0

Key takeaways: With larger models comes the need for larger datasets

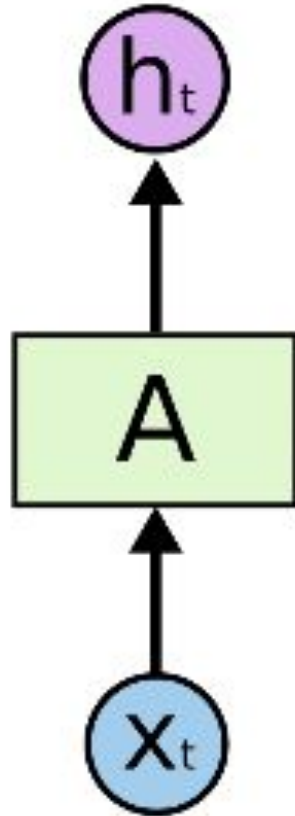
So far: Feed-forward neural networks

one to one



e.g. Image classification
Image -> Label

Vanilla Neural Network (NN)



- NN

x_t : input/event

h_t : output/prediction

A : chunk of NN

Every input is treated independently.

Recall: Why machine learning?

Relieve humans of boring, easy tasks

Perception for robotics / autonomous agents

Fun applications (e.g. art styles to my photos)

Organize and give access to visual content

Human-computer interaction

Description of content for the visually impaired

Description of content for the visually impaired



"man in black shirt is playing guitar."



"construction worker in orange safety vest is working on road."



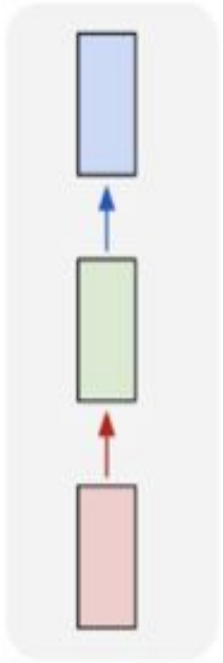
"girl in pink dress is jumping in air."



"black and white dog jumps over bar."

So far: Feed-forward neural networks

one to one



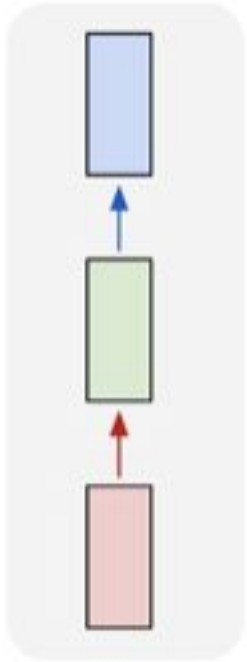
e.g. Image classification
Image -> Label 🤔

Today

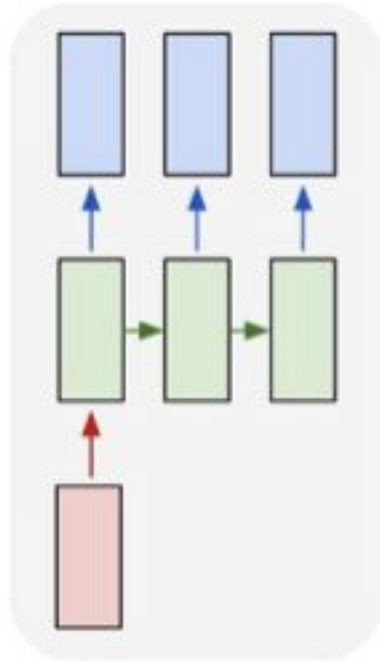
- Some practical tips while training your model.
- Forms of supervision
- **Recurrent Neural Network (RNN).**
- Transformers

Recurrent Neural Networks: Process sequences

one to one



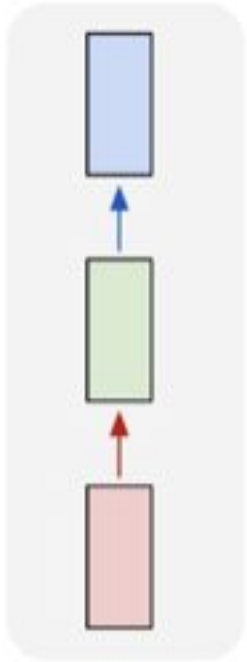
one to many



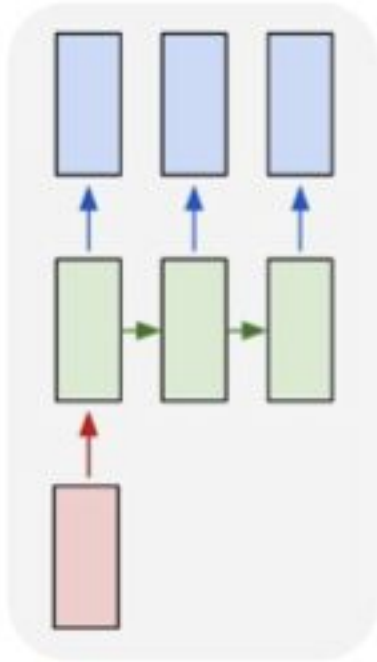
e.g. **Image Captioning:**
Image -> sequence of words

Recurrent Neural Networks: Process sequences

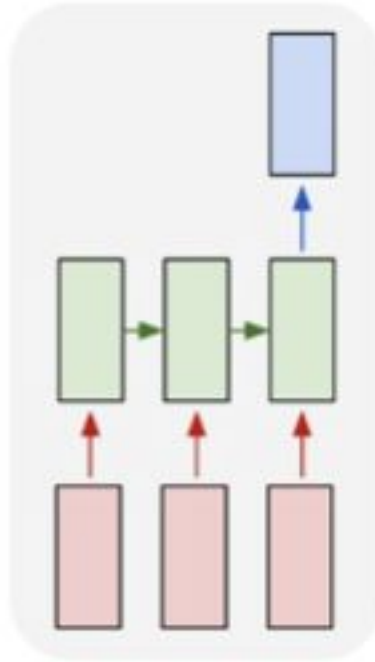
one to one



one to many



many to one

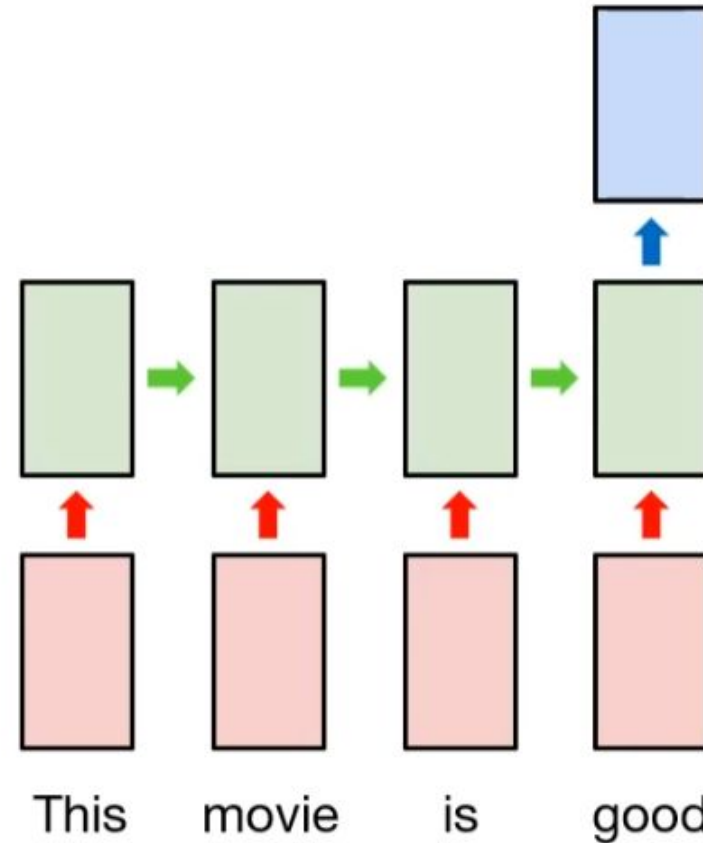




Examples usecases where we have multiple inputs and a single output.

Practical examples of many-to-one learning

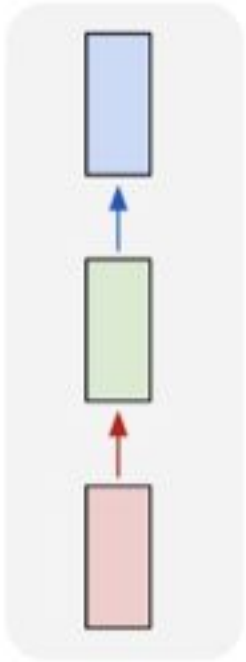
Classification : Positive or negative?



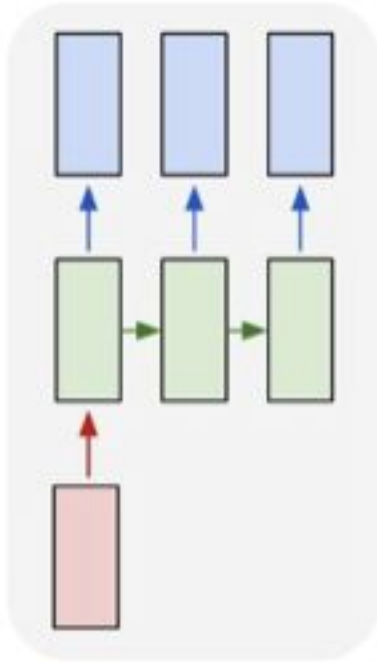
- Sentiment analysis
 - (text or audio)
- Language classification

Recurrent Neural Networks: Process sequences

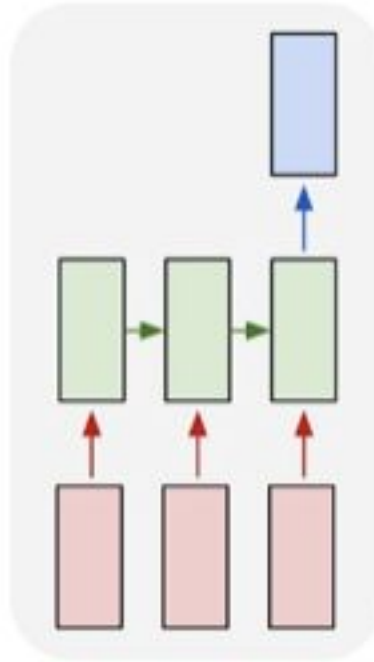
one to one



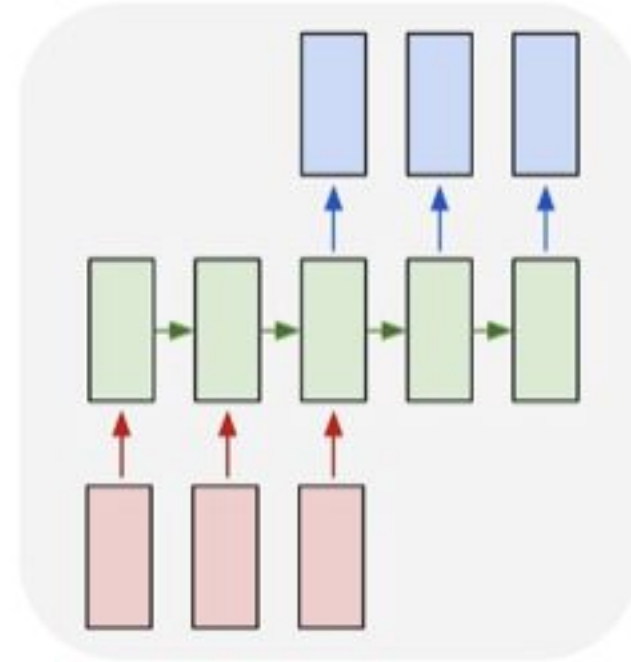
one to many



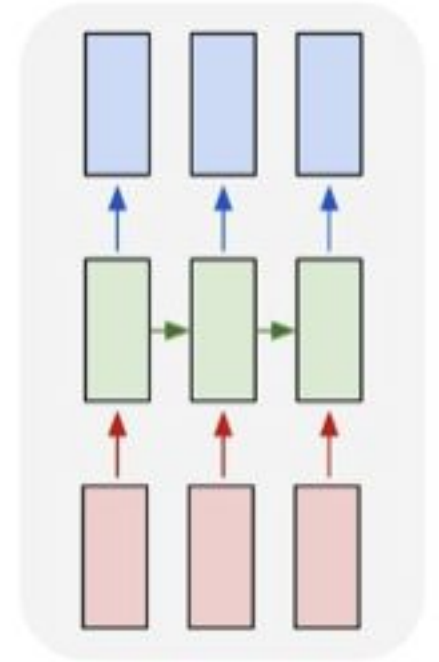
many to one

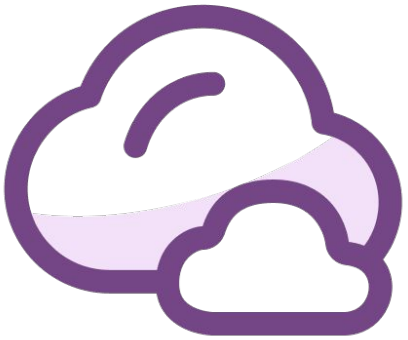


many to many



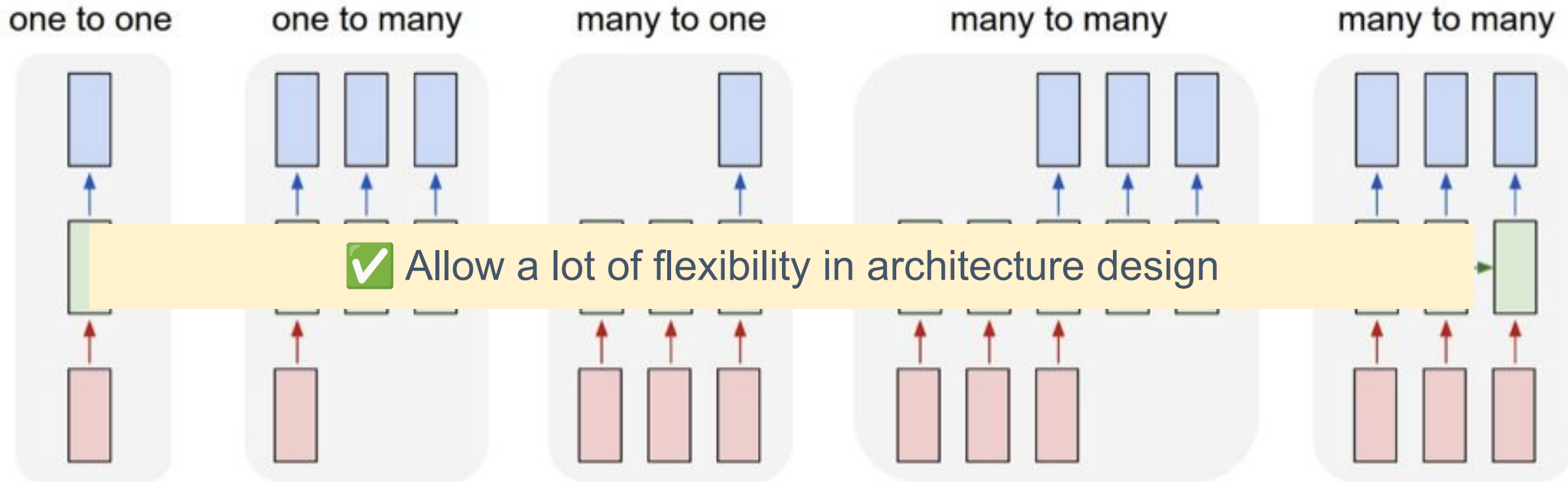
many to many



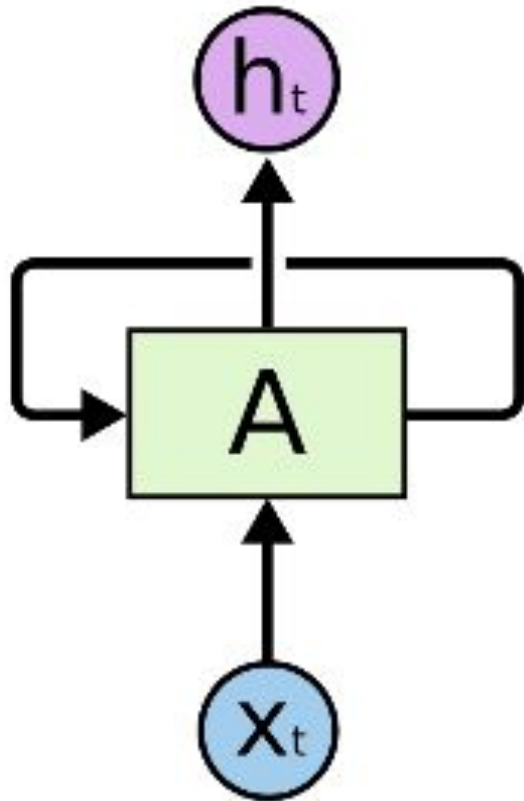


Example applications where many inputs map to many outputs

Recurrent Neural Networks: Process sequences



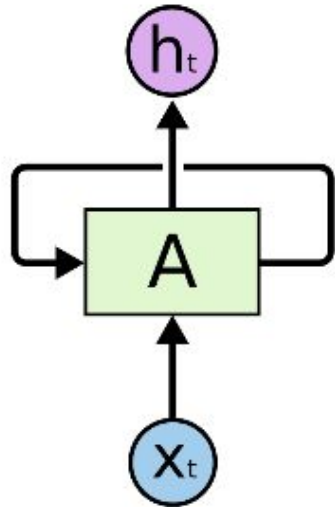
Recurrent Neural Network (RNN)



- RNN
- The loop allows information to be passed from one time step to the next.
- Now we are modeling the dynamics.

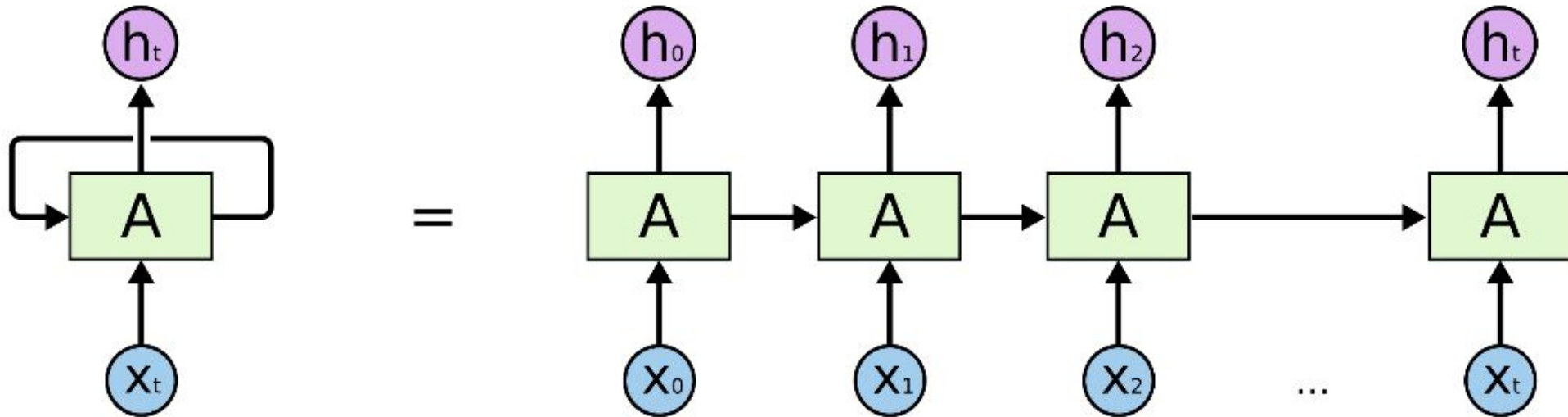
Recurrent Neural Network (RNN)

- A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor.



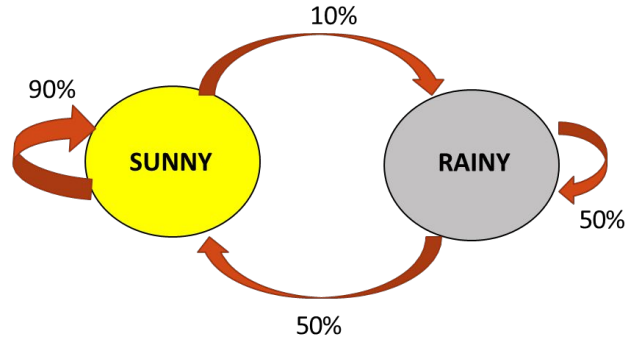
Recurrent Neural Network (RNN)

- A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor.



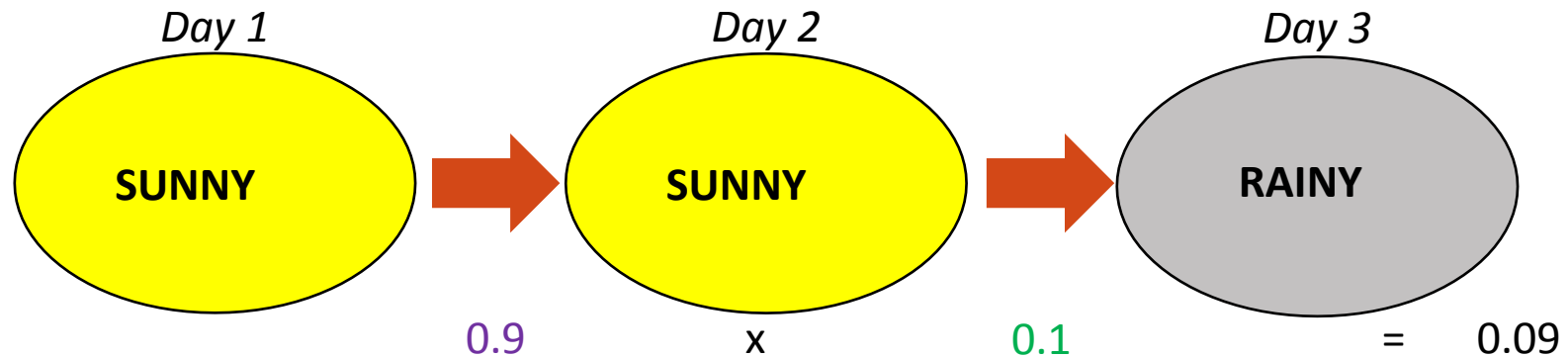
Recall: Markov models

Transition
Matrix:

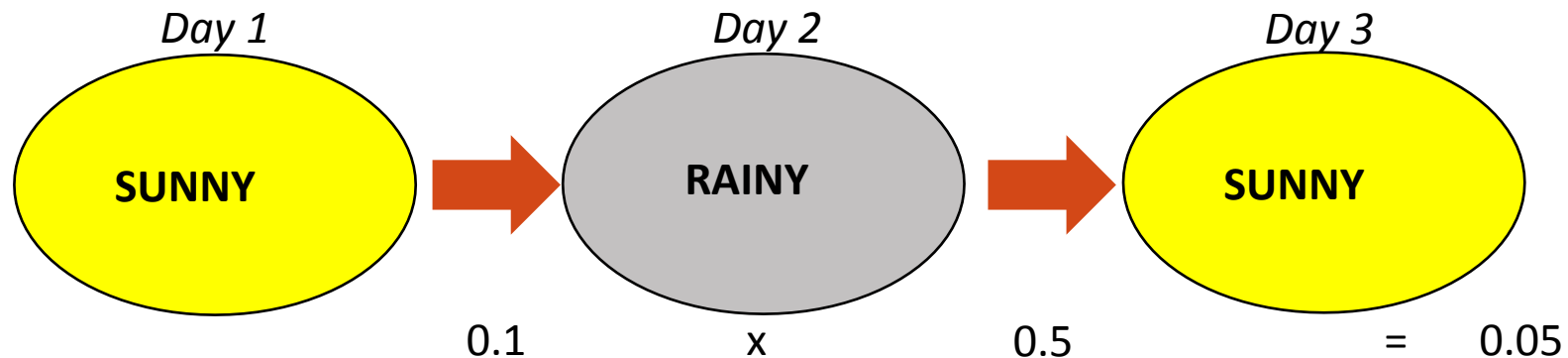


	Sunny	Rainy
Sunny	0.9	0.5
Rainy	0.1	0.5

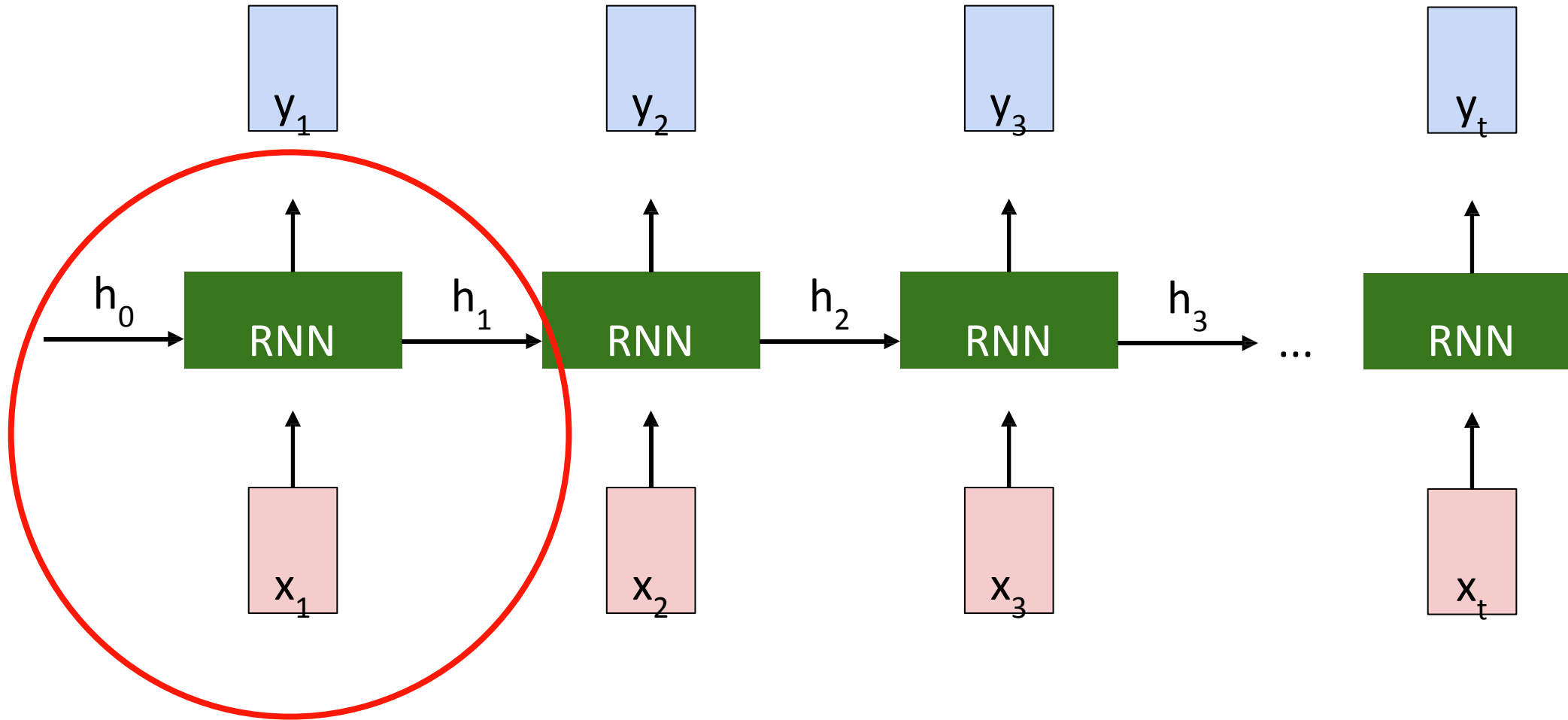
Sequence A:



Sequence B:



Recurrent Neural Network



RNN hidden state update

We can process a sequence of vectors x by applying a recurrence formula at every time step:

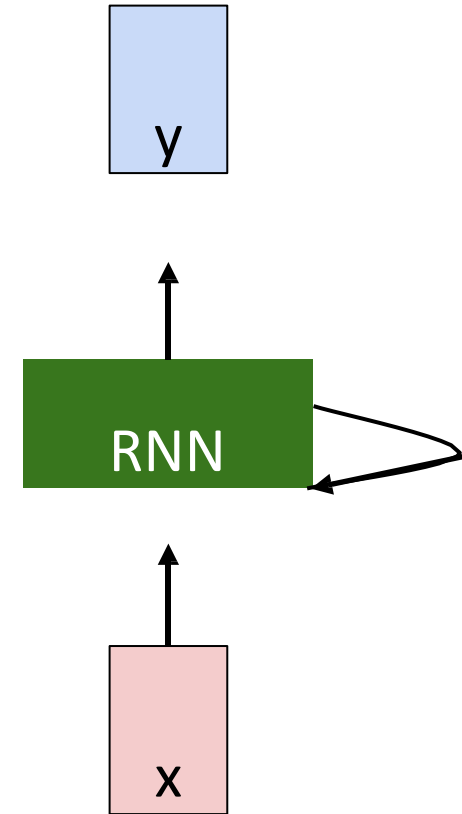
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new
state

some function with
parameters W

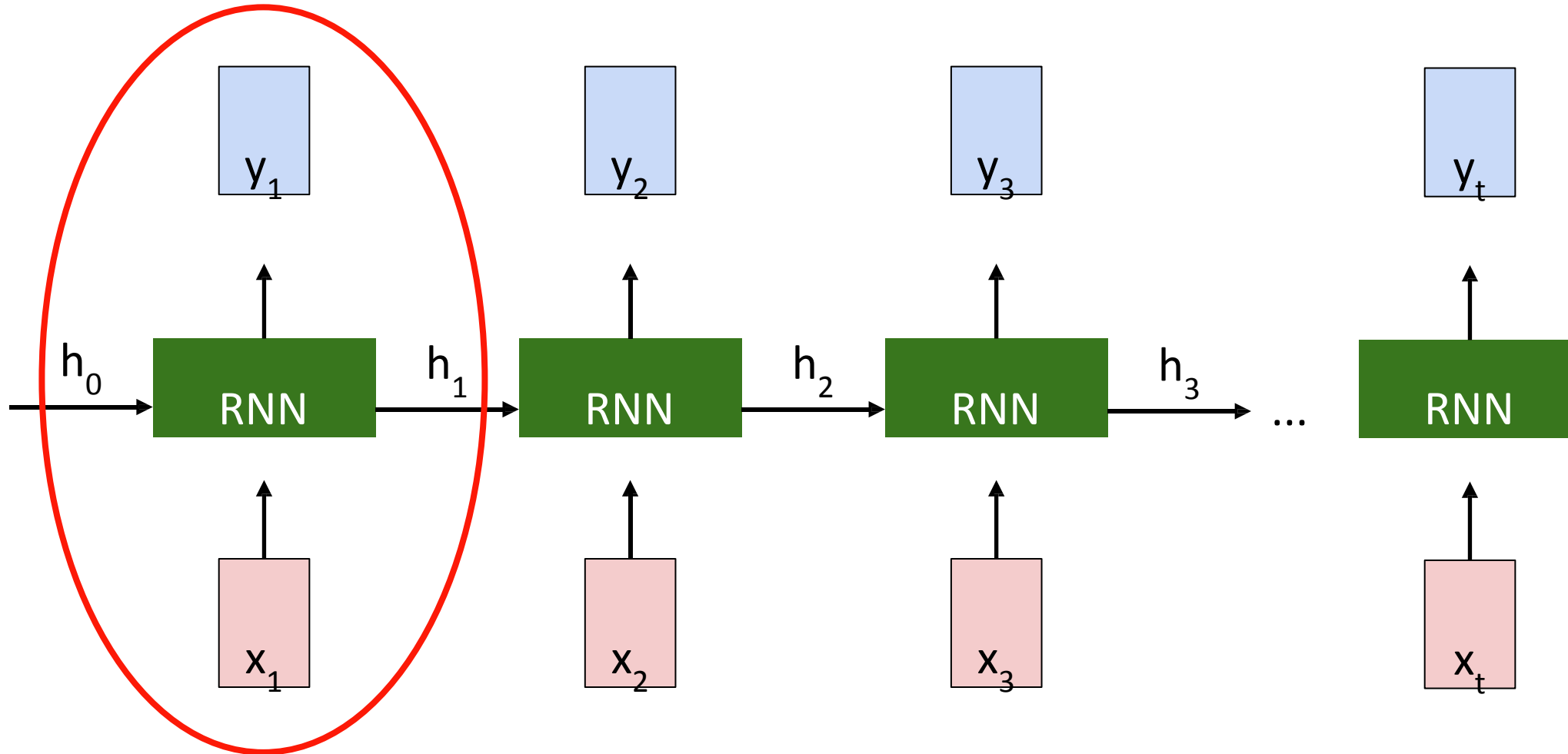
old state

input vector at
some time step



Notice: the same function (f_W) and the same set of parameters are used at every time step.

Recurrent Neural Network



RNN output generation

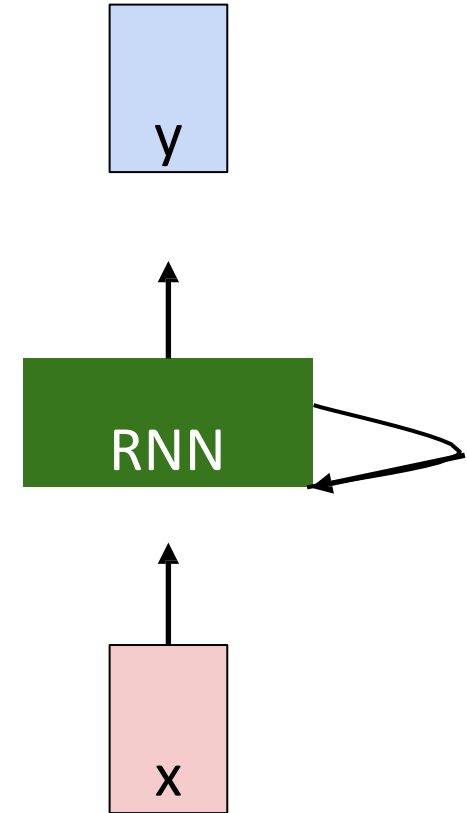
We can process a sequence of vectors x by applying a recurrence formula at every time step:

$$\boxed{y_t} = \boxed{f_{W_{hy}}}(\boxed{h_t})$$

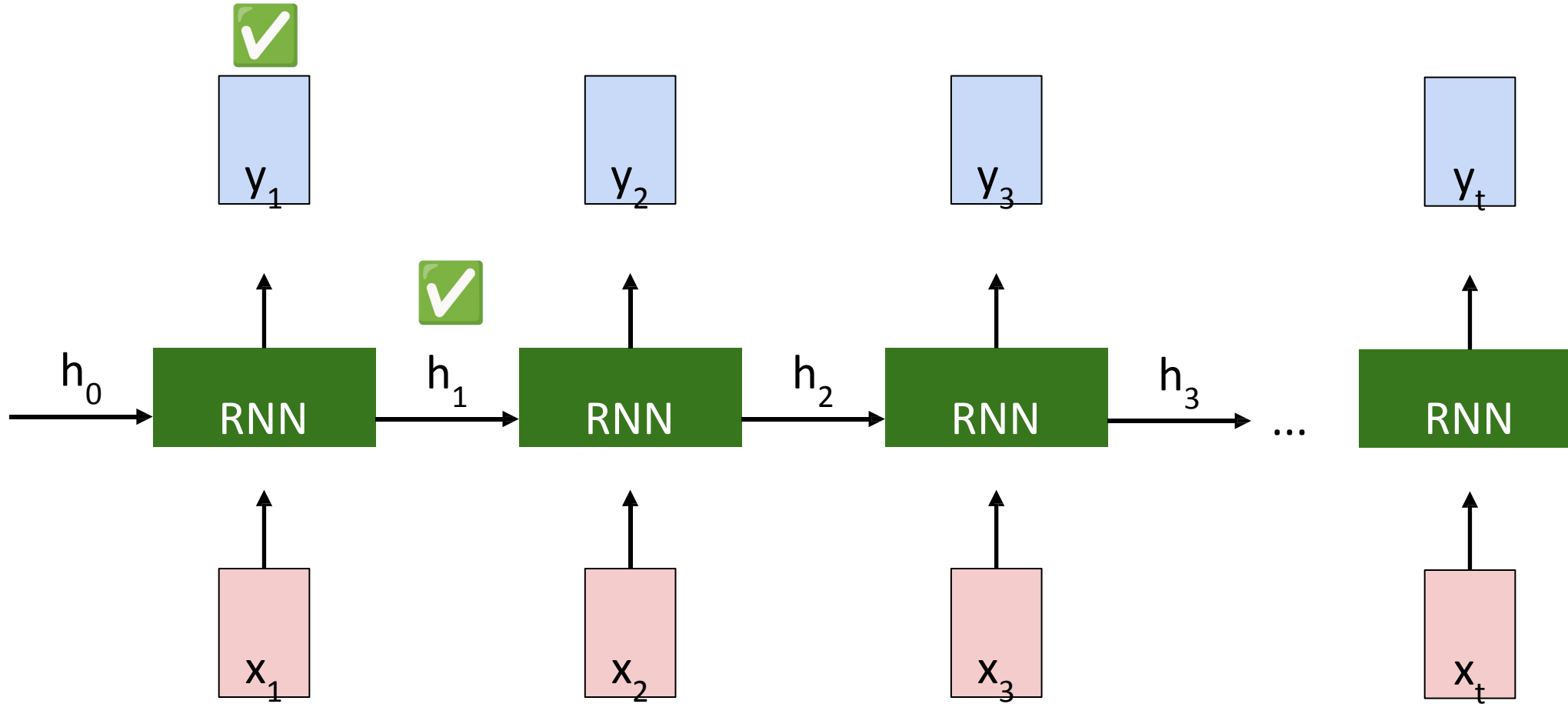
output

another function
with parameters W_{hy}

new state



Recurrent Neural Network





Example: Character RNN

Slides adapted from Fei-Fei Li & Ehsan Adeli

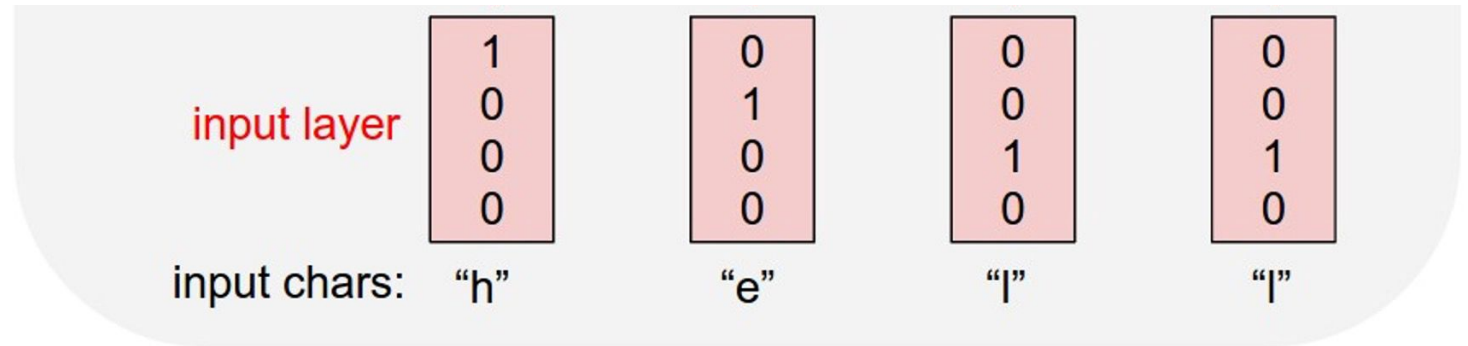
Goal: Predict next word, given current word

Application: Text generation.

Example:
Character-level
Language Model

Vocabulary:
[h,e,l,o]

Example training
sequence:
“hello”

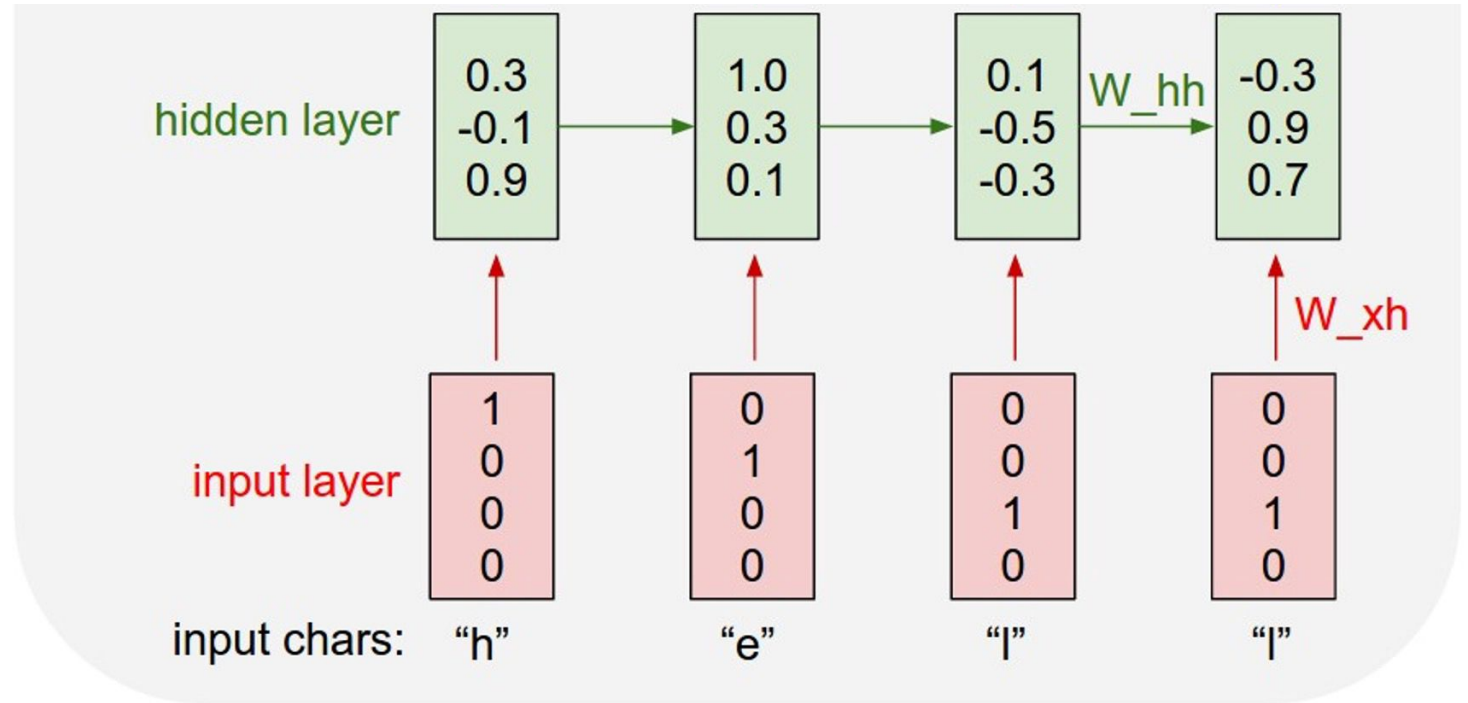


Example:
Character-level
Language Model

Vocabulary:
[h,e,l,o]

Example training
sequence:
“hello”

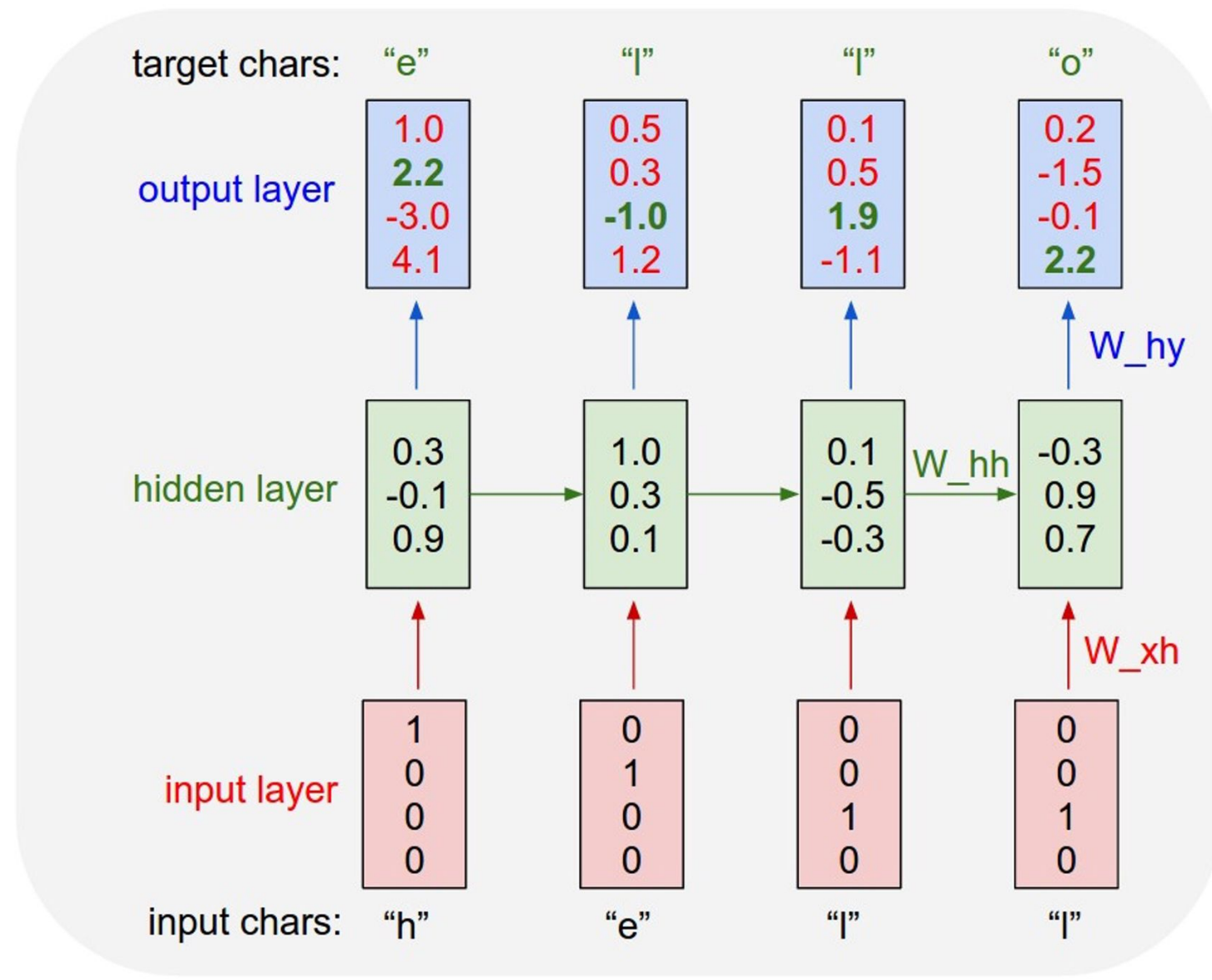
$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$



Example: Character-level Language Model

Vocabulary:
[h,e,l,o]

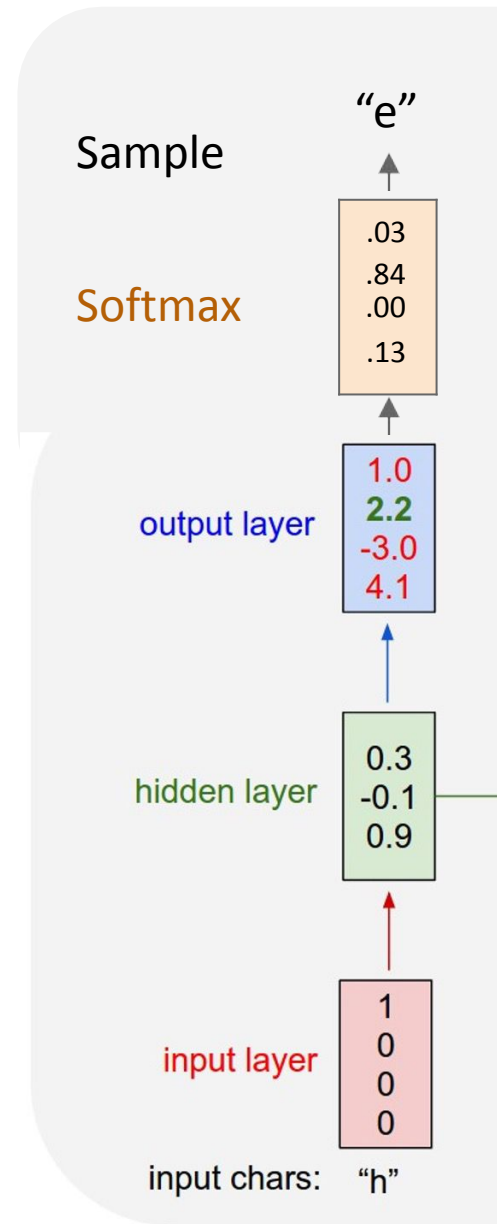
Example training
sequence:
“hello”



Example: Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

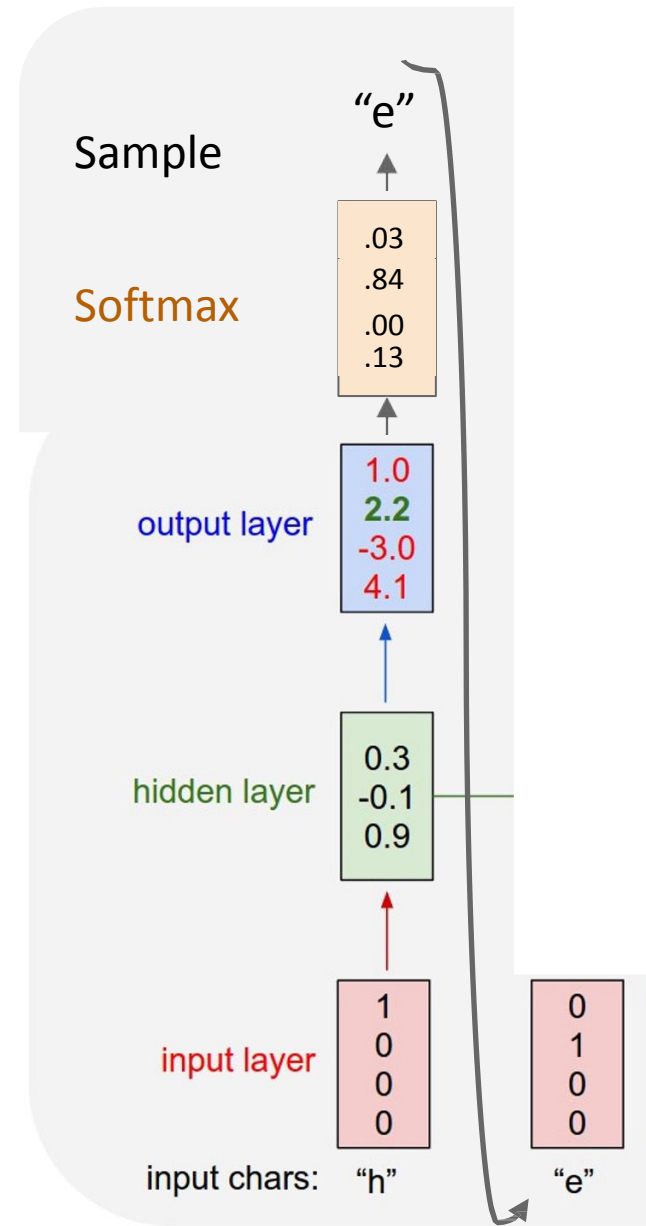
At test-time sample characters
one at a time, feed back to
model



Example: Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

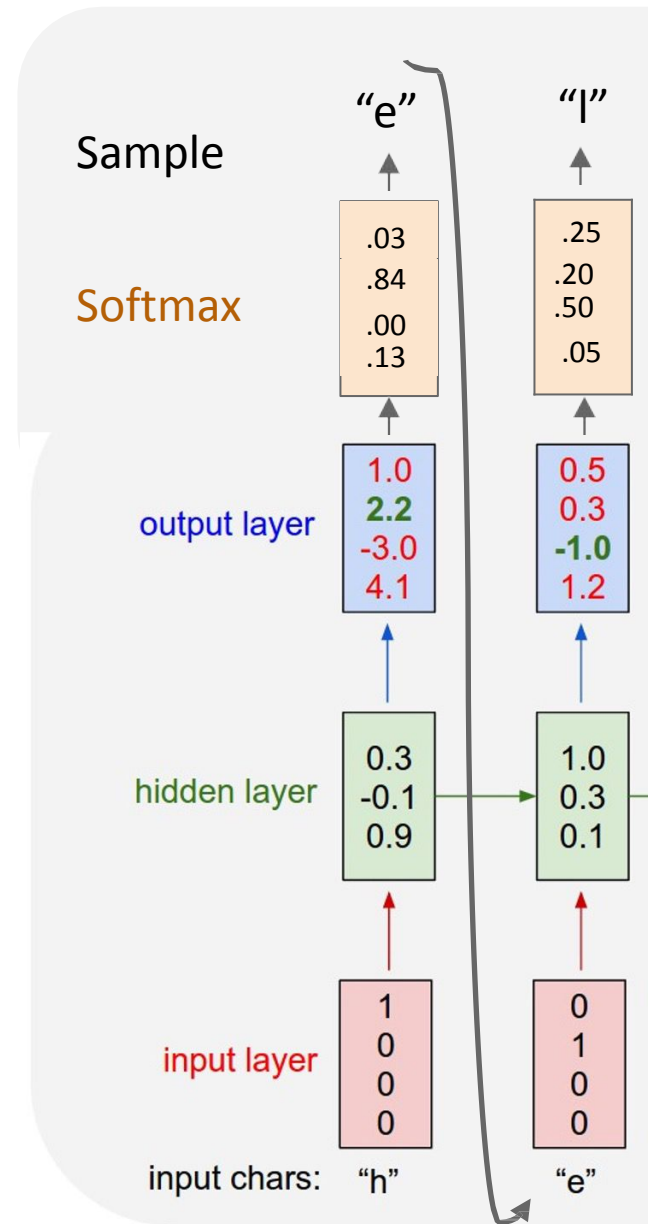
At test-time sample characters
one at a time, feed back to
model



Example: Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

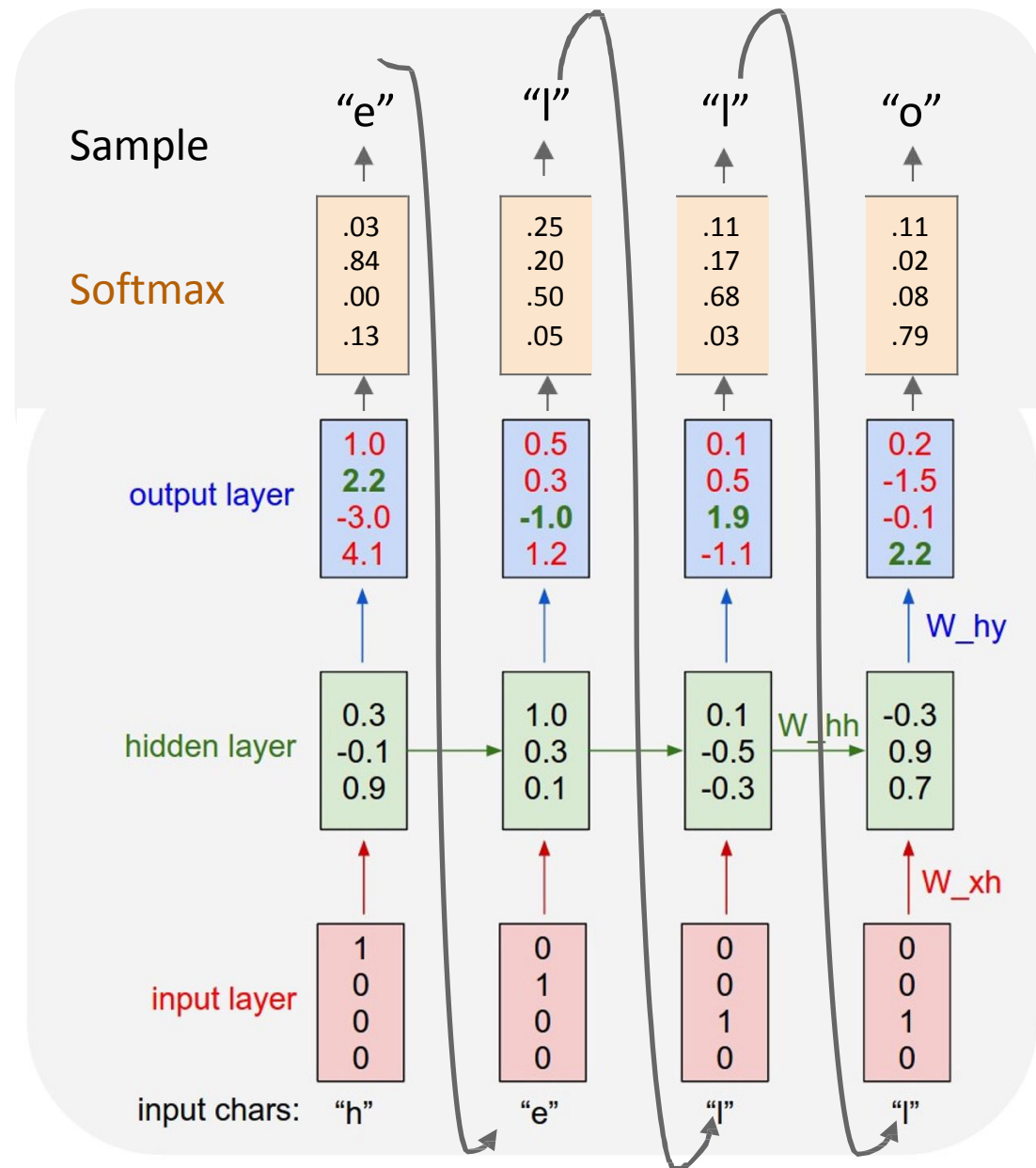
At test-time sample characters
one at a time, feed back to
model



Example: Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

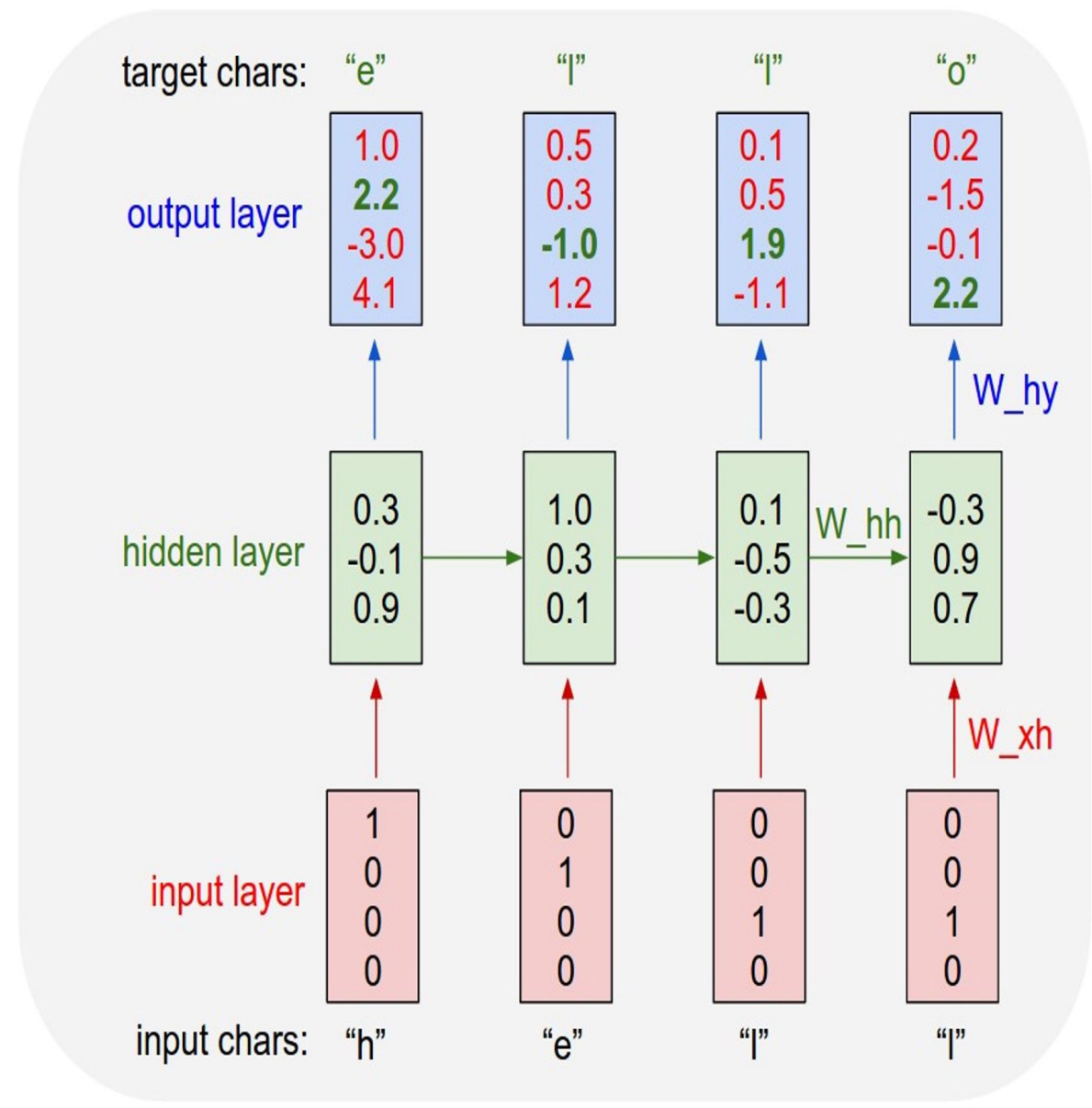
At test-time sample characters
one at a time, feed back to
model



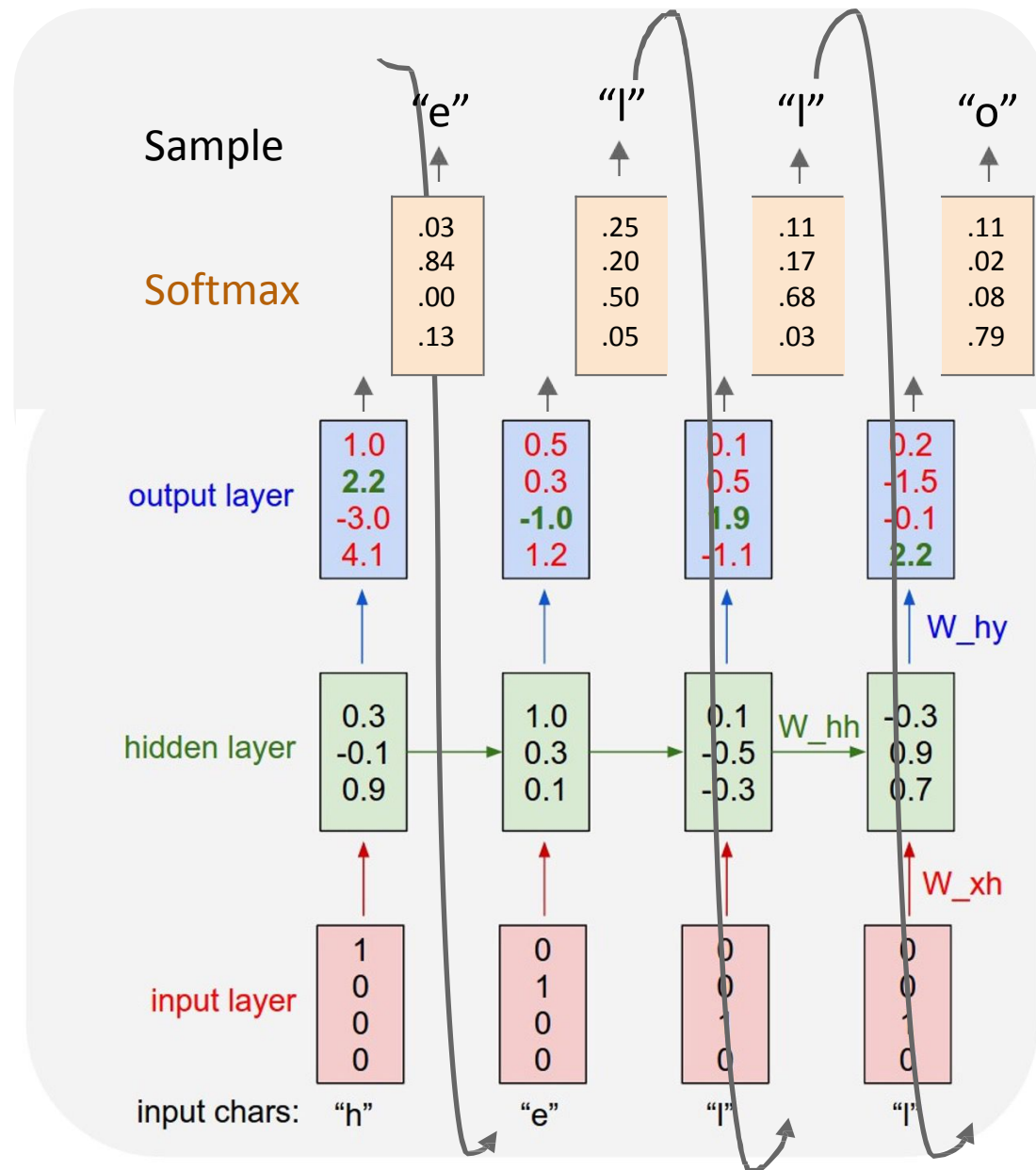


What is this model trying to learn during training?

Answer: Next character prediction



Training



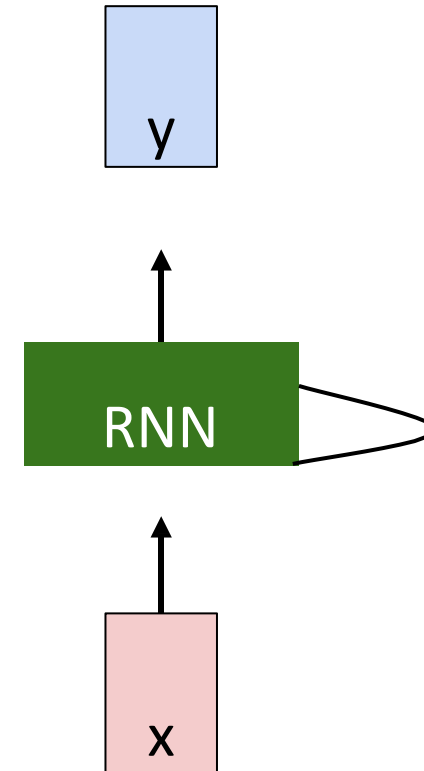
Inference

THE SONNETS

by William Shakespeare

From fairest creatures we desire increase,
That thereby beauty's rose might never die,
But as the ripper should by time decease,
His tender heir might bear his memory:
But thou, contracted to thine own bright eyes,
Feed'st thy light's flame with self-substantial fuel,
Making a famine where abundance lies,
Thyself thy foe, to thy sweet self too cruel:
Thou that art now the world's fresh ornament,
And only herald to the gaudy spring,
Within thine own bud buriest thy content,
And tender churl mak'st waste in niggarding:
Pity the world, or else this glutton be,
To eat the world's due, by the grave and thee.

When forty winters shall besiege thy brow,
And dig deep trenches in thy beauty's field,
Thy youth's proud livery so gazed on now,
Will be a tatter'd weed of small worth held:
Then being asked, where all thy beauty lies,
Where all the treasure of thy lusty days;
To say, within thine own deep sunken eyes,
Where an all-eating shame, and thriftless praise.
How much more praise deserv'd thy beauty's use,
If thou couldst answer 'This fair child of mine
Shall sum my count, and make my old excuse,'
Proving his beauty by succession thine!
This were to be new made when thou art old,
And see thy blood warm when thou feel'st it cold.



at first:

tyntd-iafhatawiaoihrdemot lytdws e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrqd t o idoe ns,smtt h ne etie h,hregtrs niglike,aoaenns lng



train more

"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwv fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."



train more

Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.



train more

"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.

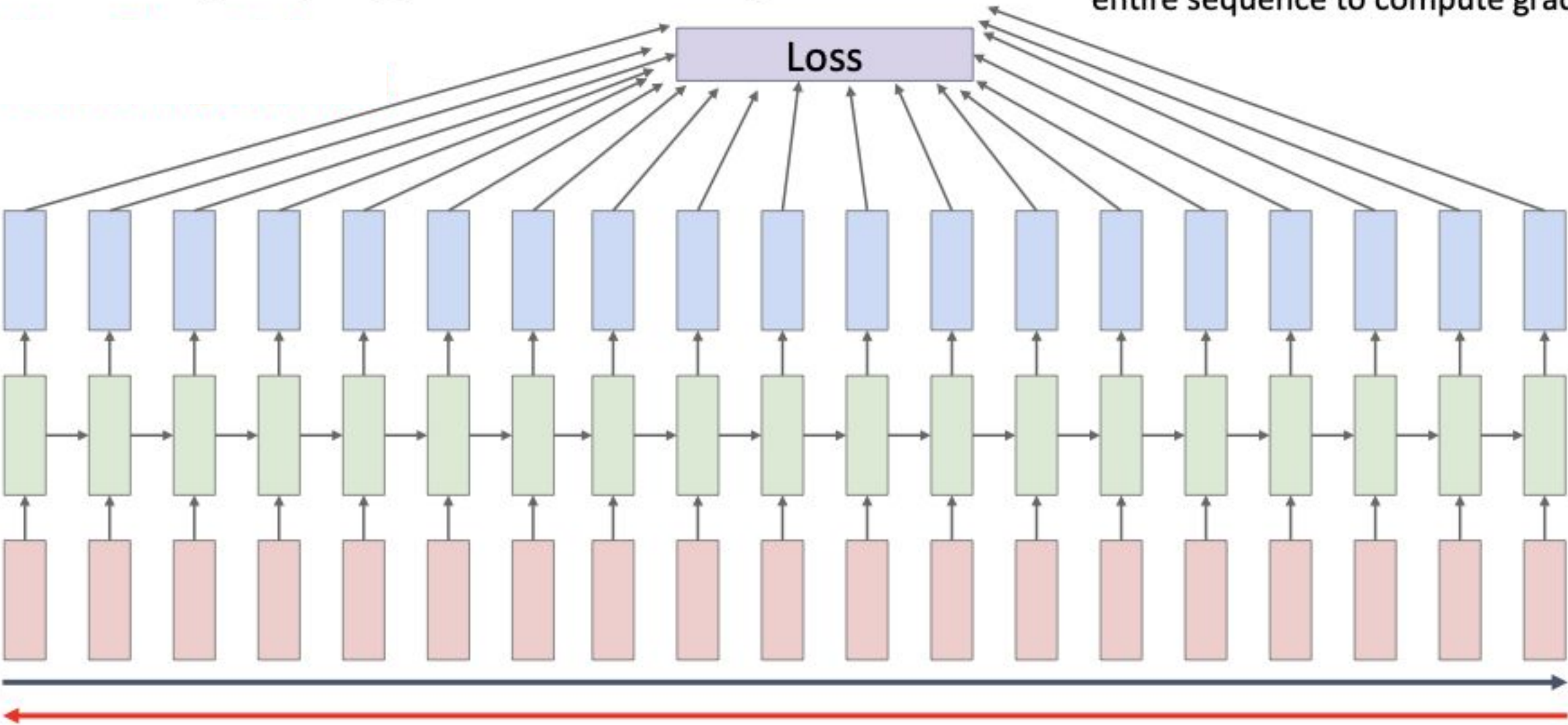
Recurrent Neural Networks: Process sequences

How does the model retain information throughout time?

- Remember everything! 🧠
 - Sequence at timestep (t) remembers everything from 0 to t-1

Backpropagation Through Time

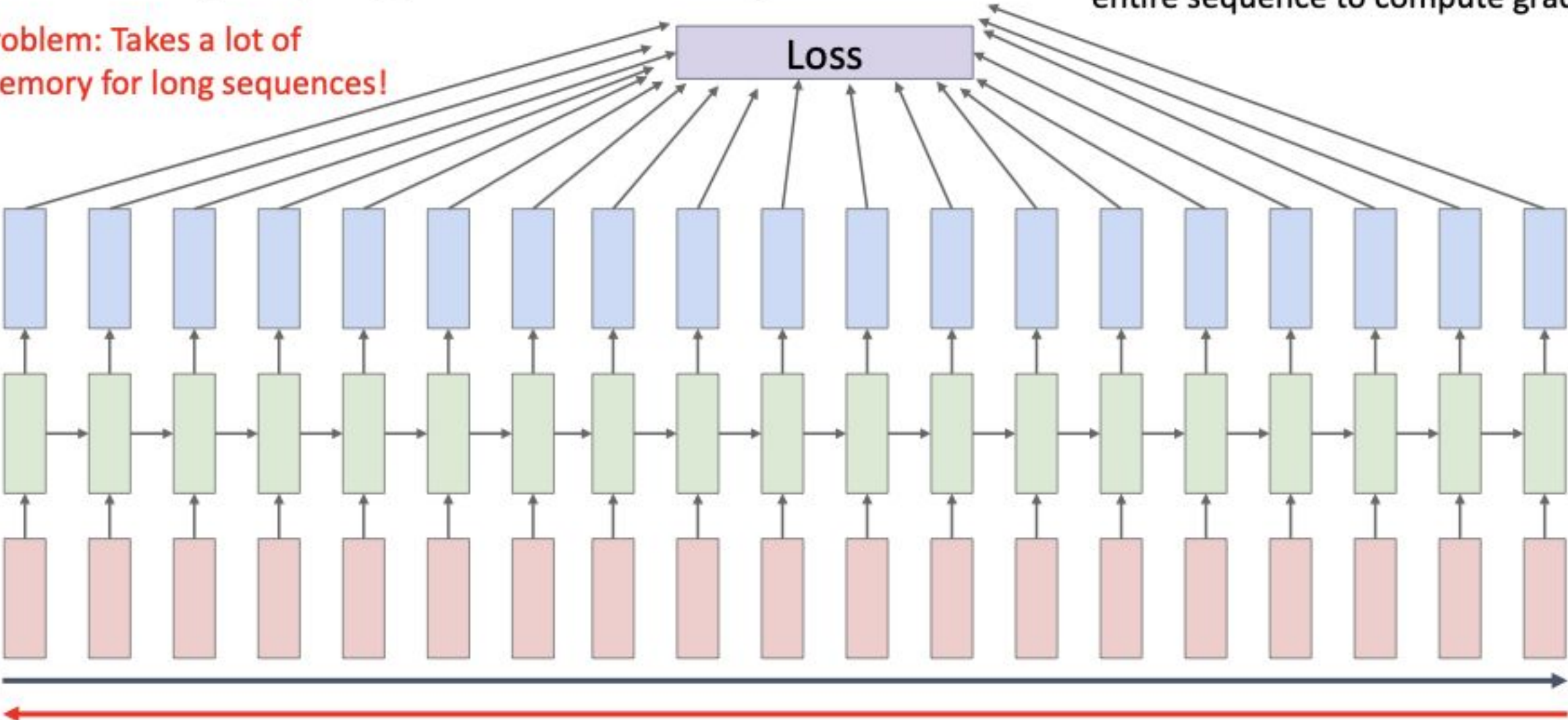
Forward through entire sequence to compute loss, then backward through entire sequence to compute gradient



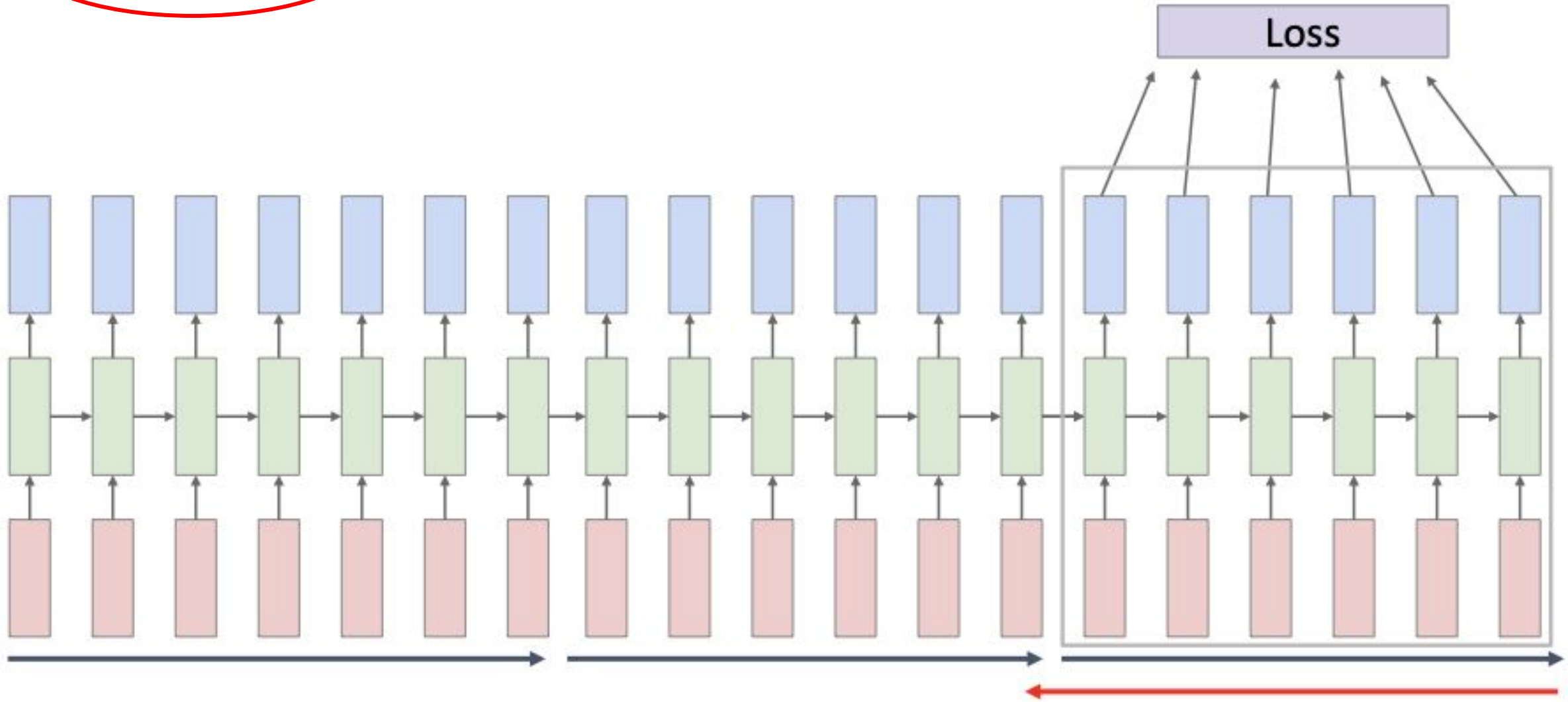
Backpropagation Through Time

Problem: Takes a lot of memory for long sequences!

Forward through entire sequence to compute loss, then backward through entire sequence to compute gradient



Truncated Backpropagation Through Time



Recurrent Neural Networks: Process sequences

How does the model retain information throughout time?

- Truncate to a fixed time steps for gradient influence.
 - **Pro:** Reduces the memory footprint.
 - **Con:** What if there is a dependency on a token which is past the fixed step parameter?

Recurrent Neural Networks: Process sequences

How does the model retain information throughout time?

- Truncate to a fixed time steps for gradient influence.
 - **Pro:** Reduces the memory footprint.
 - **Con:** What if there is a dependency on a token which is past the fixed step parameter?
- Summarize the past into a single context vector.
 - **Pro:** Reduces the memory footprint.
 - **Con:** Hard to pack the entire past into a single context vector.



Recurrent Neural Networks: Process sequences

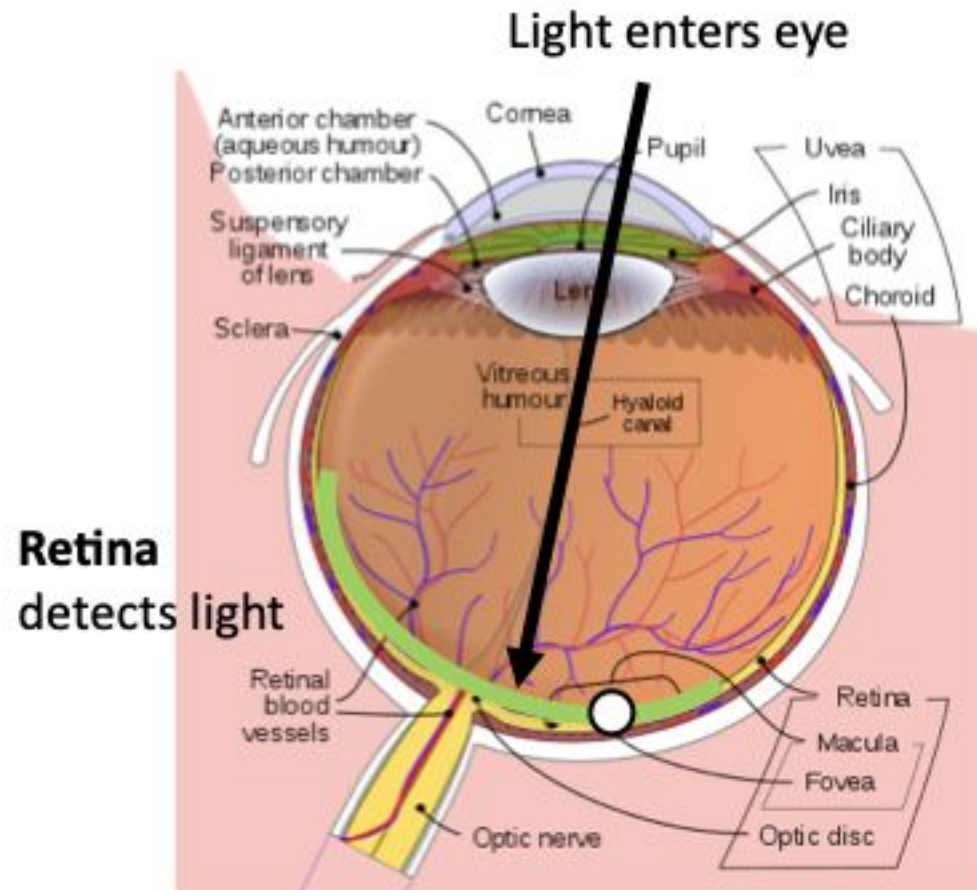
How does the model retain information throughout time?

- Truncate to a fixed time steps for gradient influence.
 - **Pro:** Reduces the memory footprint.
 - **Con:** What if there is a dependency on a token which is past the fixed step parameter?
- Summarize the past into a single context vector.
 - **Pro:** Reduces the memory footprint.
 - **Con:** Hard to pack the entire past into a single context vector.
- ...

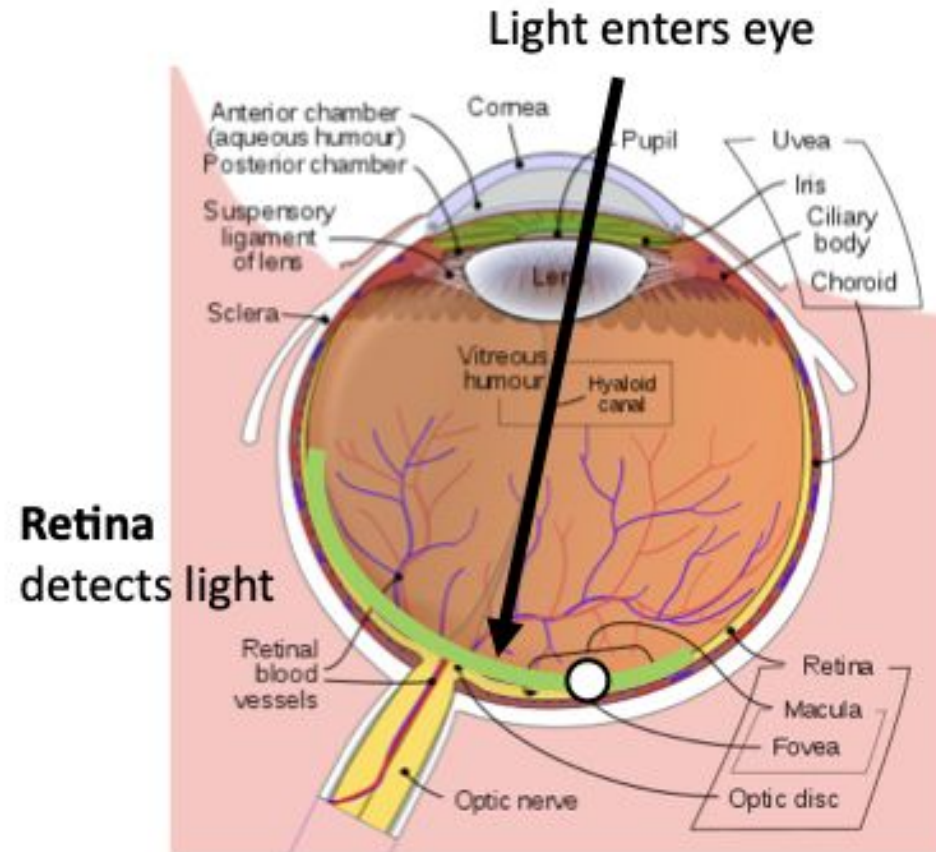
Attention!



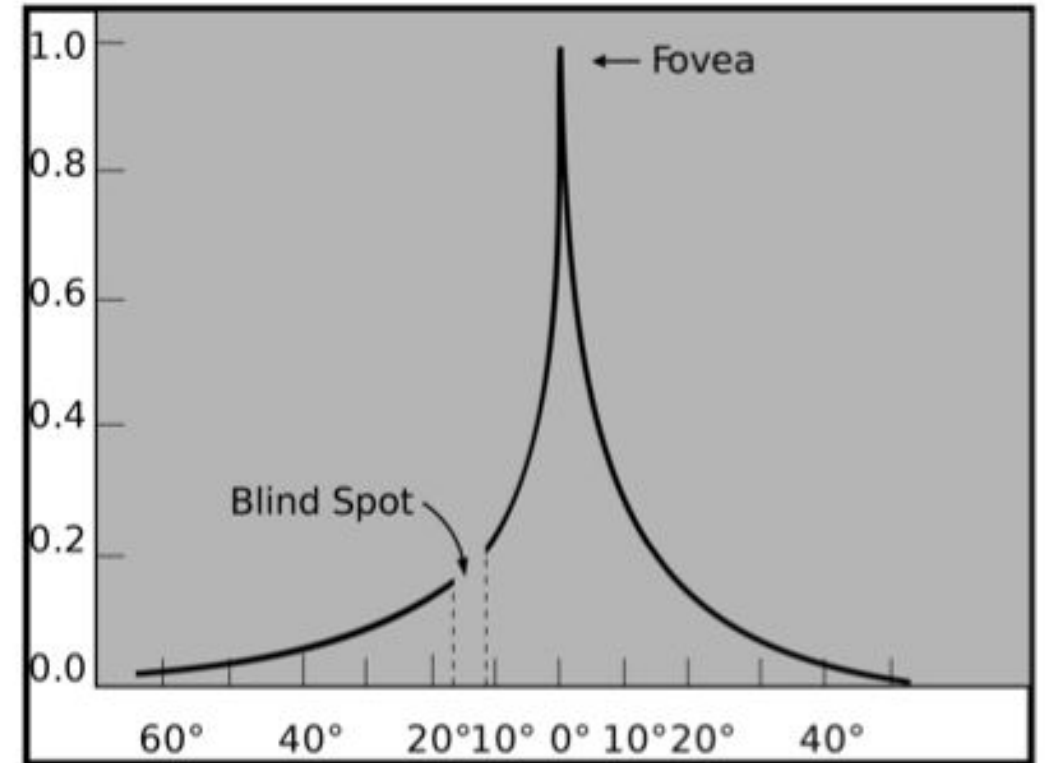
Human vision: Fovea



Human vision: Fovea



The **fovea** is a tiny region of the retina that can see with high acuity

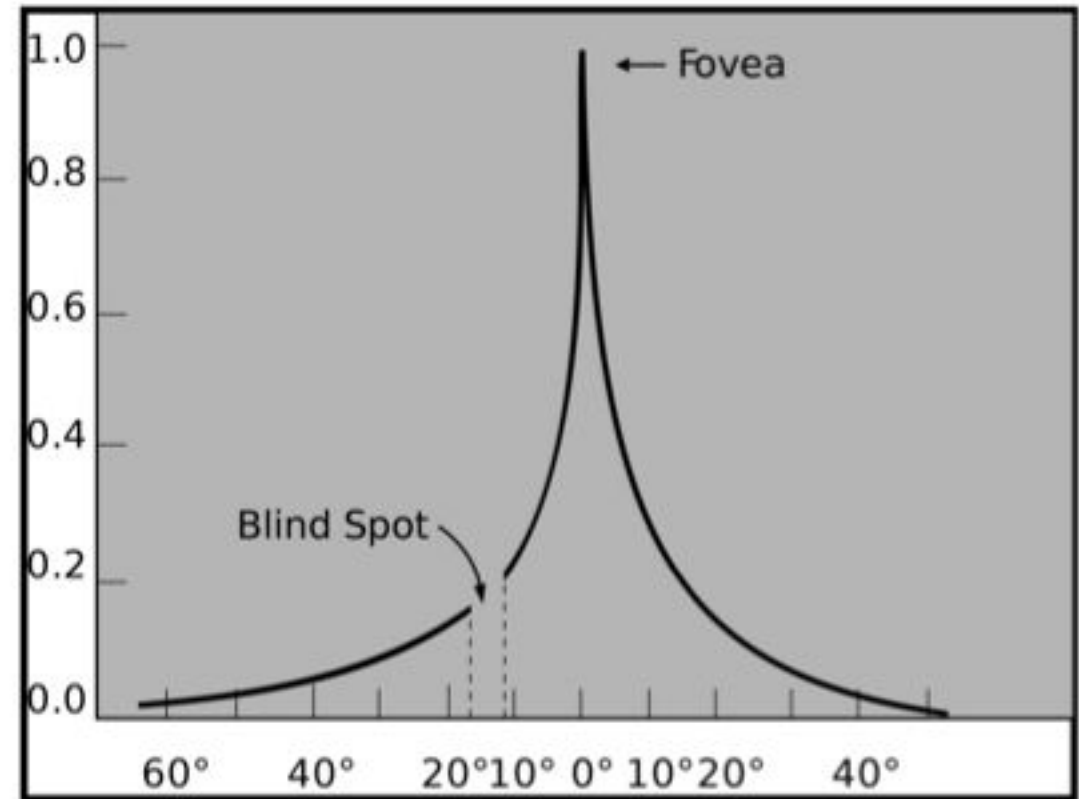


Human vision: Saccades

Human eyes are constantly moving so we don't notice



The **fovea** is a tiny region of the retina that can see with high acuity



Slide credit: Justin Johnson

Key idea: Instead of truncating to a fixed sized window, **learn** which parts to

