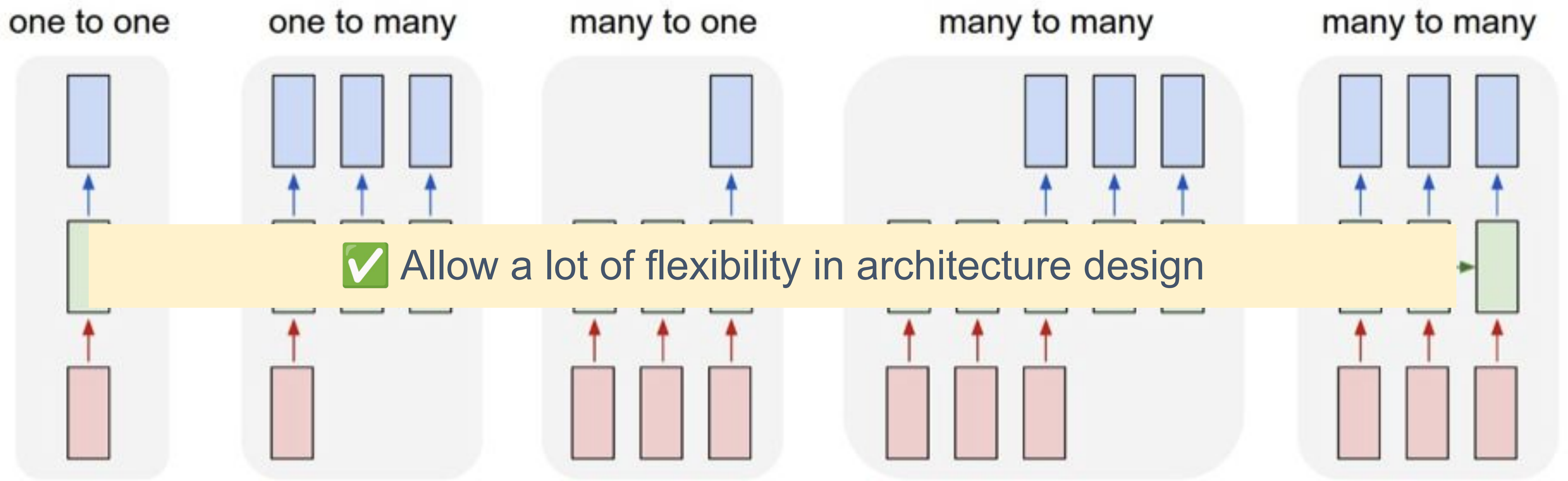
Announcements

- Pset4 out, due Thursday, April 10th
- No laptops during class - feel free to leave now.
- Challenge will be released today.

Last time

- Recurrent Neural Network (RNN).

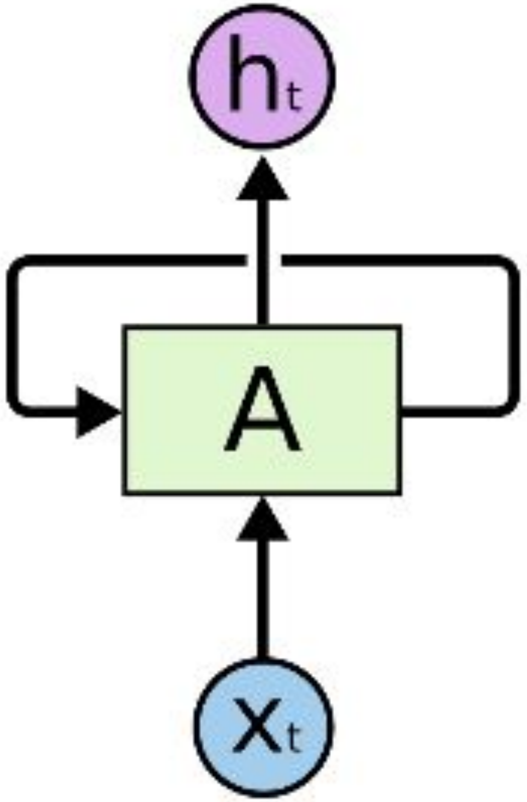
Recurrent Neural Networks: Process sequences



✅ Allow a lot of flexibility in architecture design

Slide credit: Justin Johnson

Recurrent Neural Network (**R**NN)



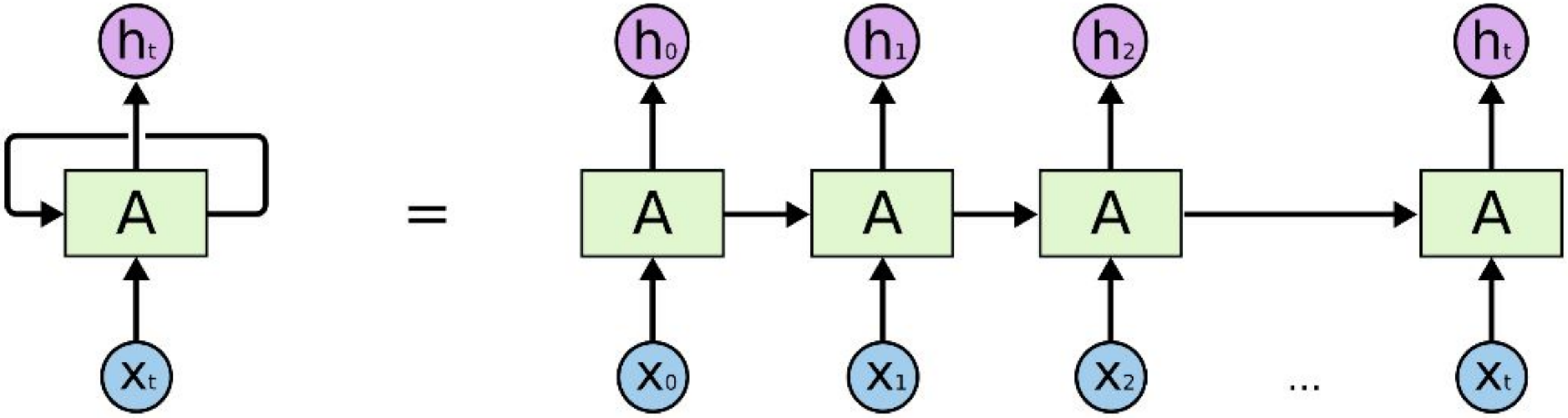
- RNN
- The loop allows information to be passed from one time step to the next.
- Now we are modeling the dynamics.

Chris Olah, Blog, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

slide by Sarah Bargal

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.



Chris Olah, Blog, http://colah.github.io/posts/2015-08-Understanding-LSTMs/ slide by Sarah Bargal

Today

- Practical scenarios where RNN is used
- RNN Gradients
- Attention
- Types of attention
- Transformers

Today

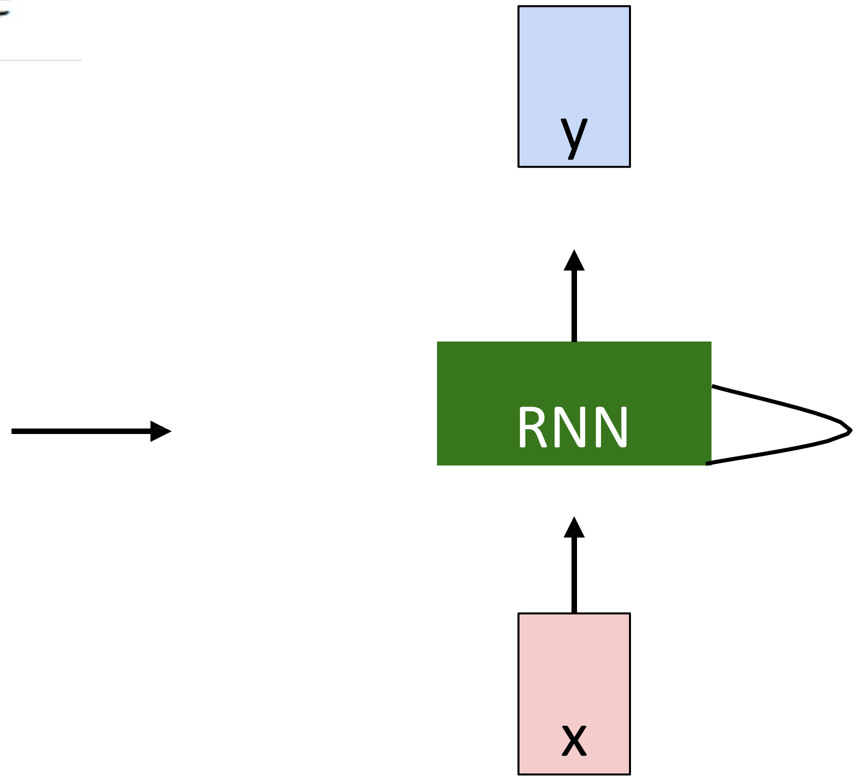
- **Practical scenarios where RNN is used**
- RNN Gradients
- Attention
- Types of attention
- Transformers

THE SONNETS

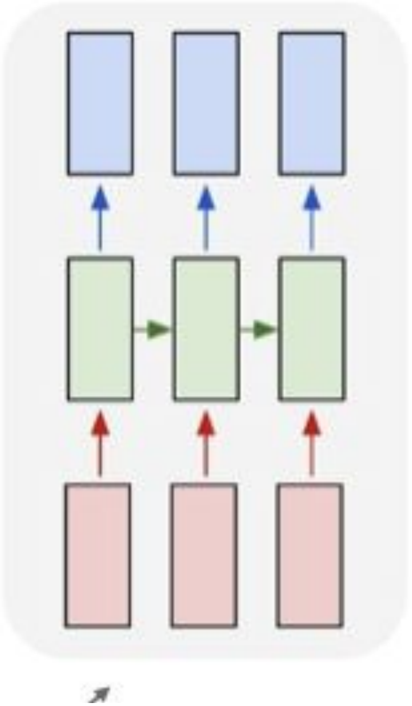
by William Shakespeare

From fairest creatures we desire increase,
That thereby beauty's rose might never die,
But as the riper 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,
Were 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.



many to many



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

train more

```
"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy 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.
```

many to many

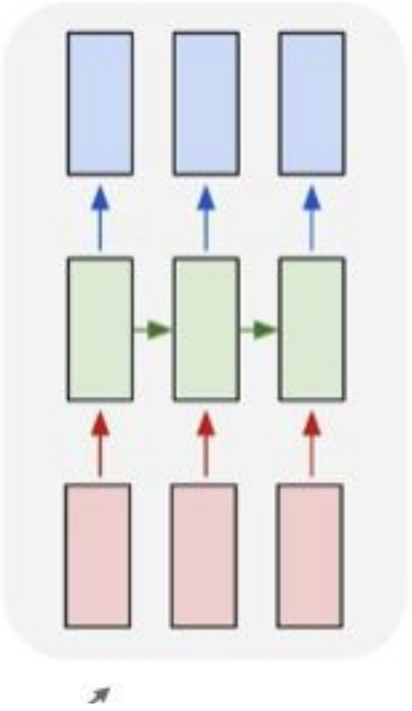


Image Captioning: Example Results

A cat sitting on a suitcase on the floor

A cat is sitting on a tree branch

A dog is running in the grass with a frisbee

A white teddy bear sitting in the grass

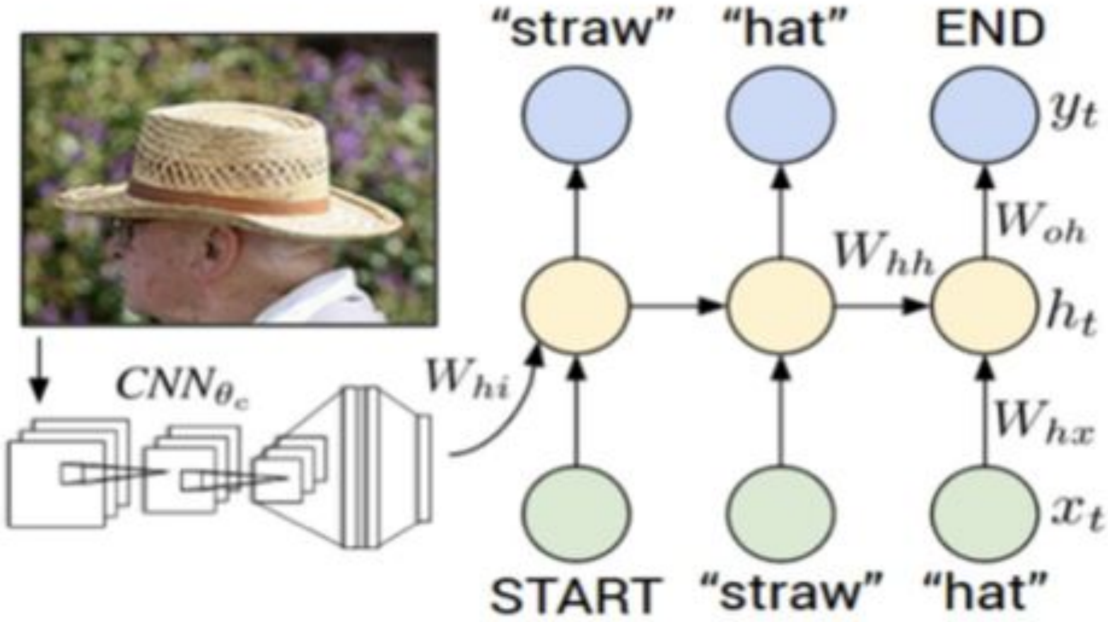
Two people walking on the beach with surfboards

A tennis player in action on the court

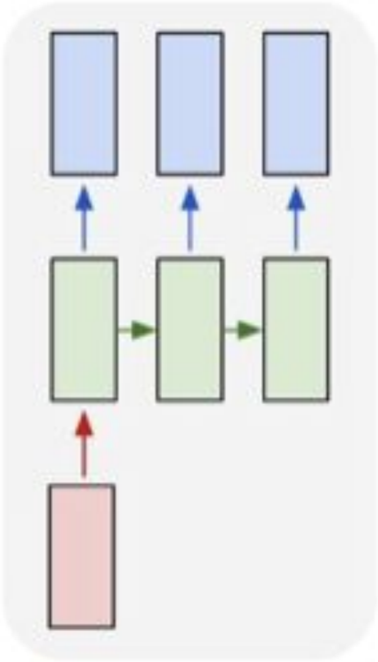
Two giraffes standing in a grassy field

A man riding a dirt bike on a dirt track

Example: Image Captioning



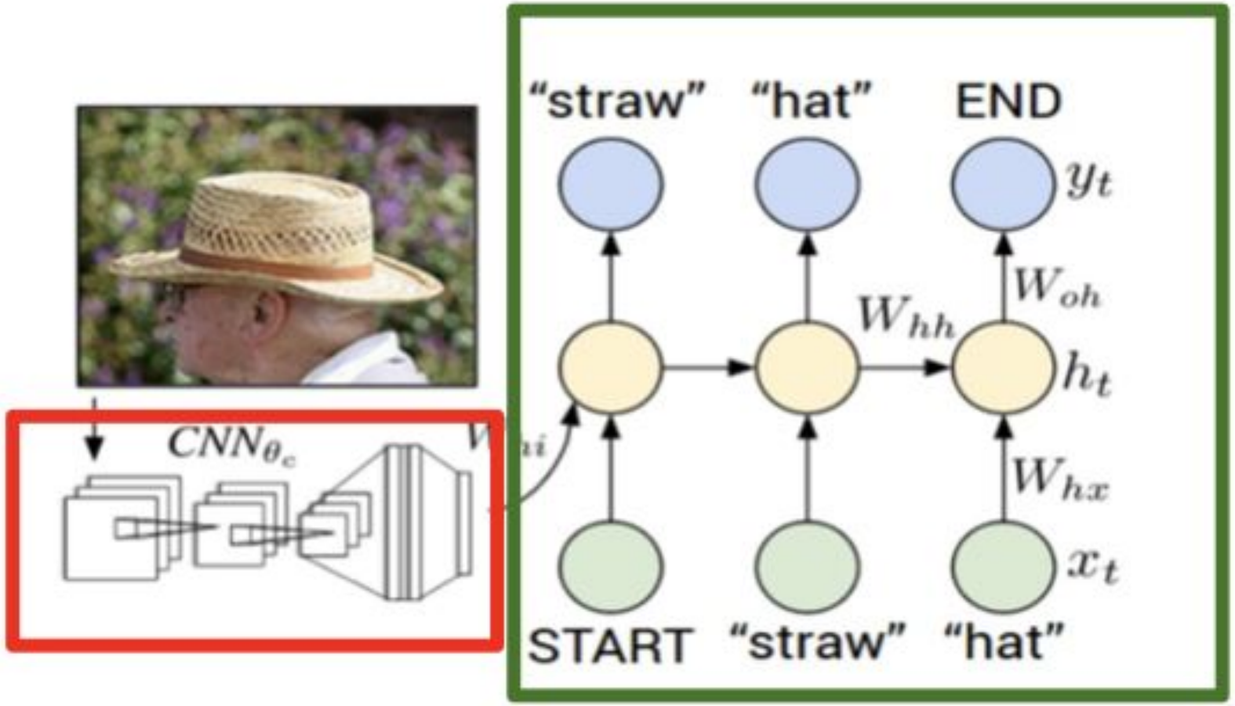
one to many



Mao et al, “Explain Images with Multimodal Recurrent Neural Networks”, NeurIPS 2014 Deep Learning and Representation Workshop
Karpathy and Fei-Fei, “Deep Visual-Semantic Alignments for Generating Image Descriptions”, CVPR 2015
Vinyals et al, "Show and Tell: A Neural Image Caption Generator”, CVPR 2015
Donahue et al, “Long-term Recurrent Convolutional Networks for Visual Recognition and Description”, CVPR 2015
Chen and Zitnick, “Learning a Recurrent Visual Representation for Image Caption Generation”, CVPR 2015

Figure from Karpathy et a, “Deep Visual-Semantic Alignments for Generating Image Descriptions”, CVPR 2015

Example: Image Captioning



Recurrent Neural Network

Convolutional Neural Network

one to many

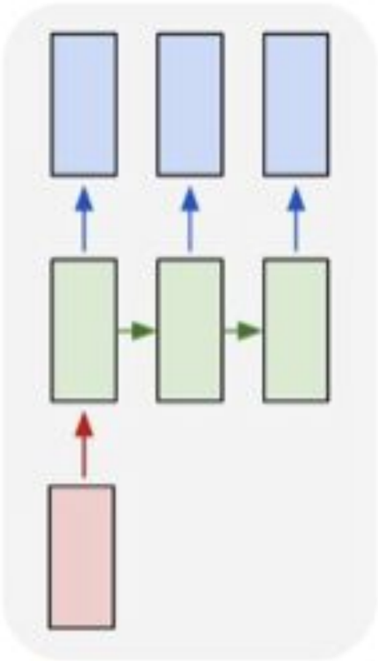


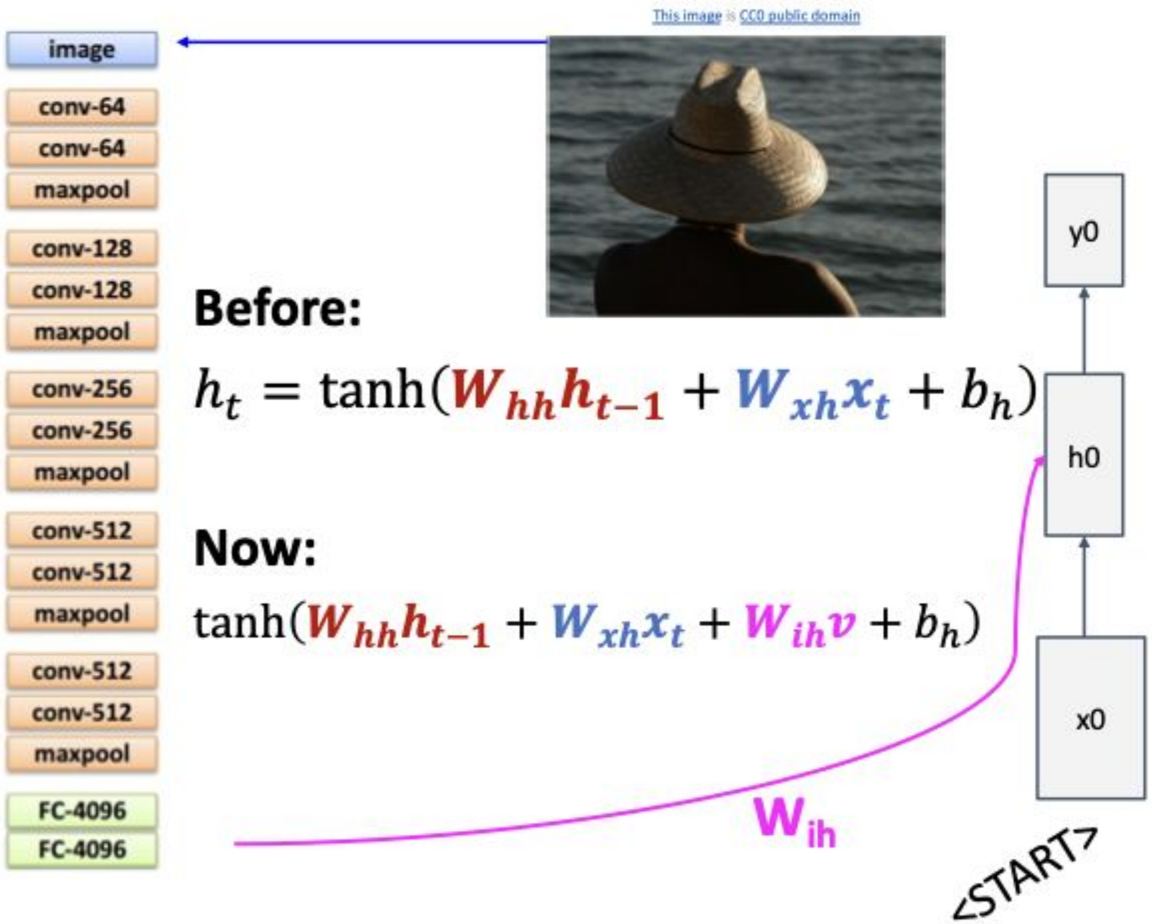
Figure from Karpathy et a, "Deep Visual-Semantic Alignments for Generating Image Descriptions", CVPR 2015

Example: Image Captioning

Transfer learning: Take CNN trained on ImageNet, chop off last layer

Example: Image Captioning

Example: Image Captioning



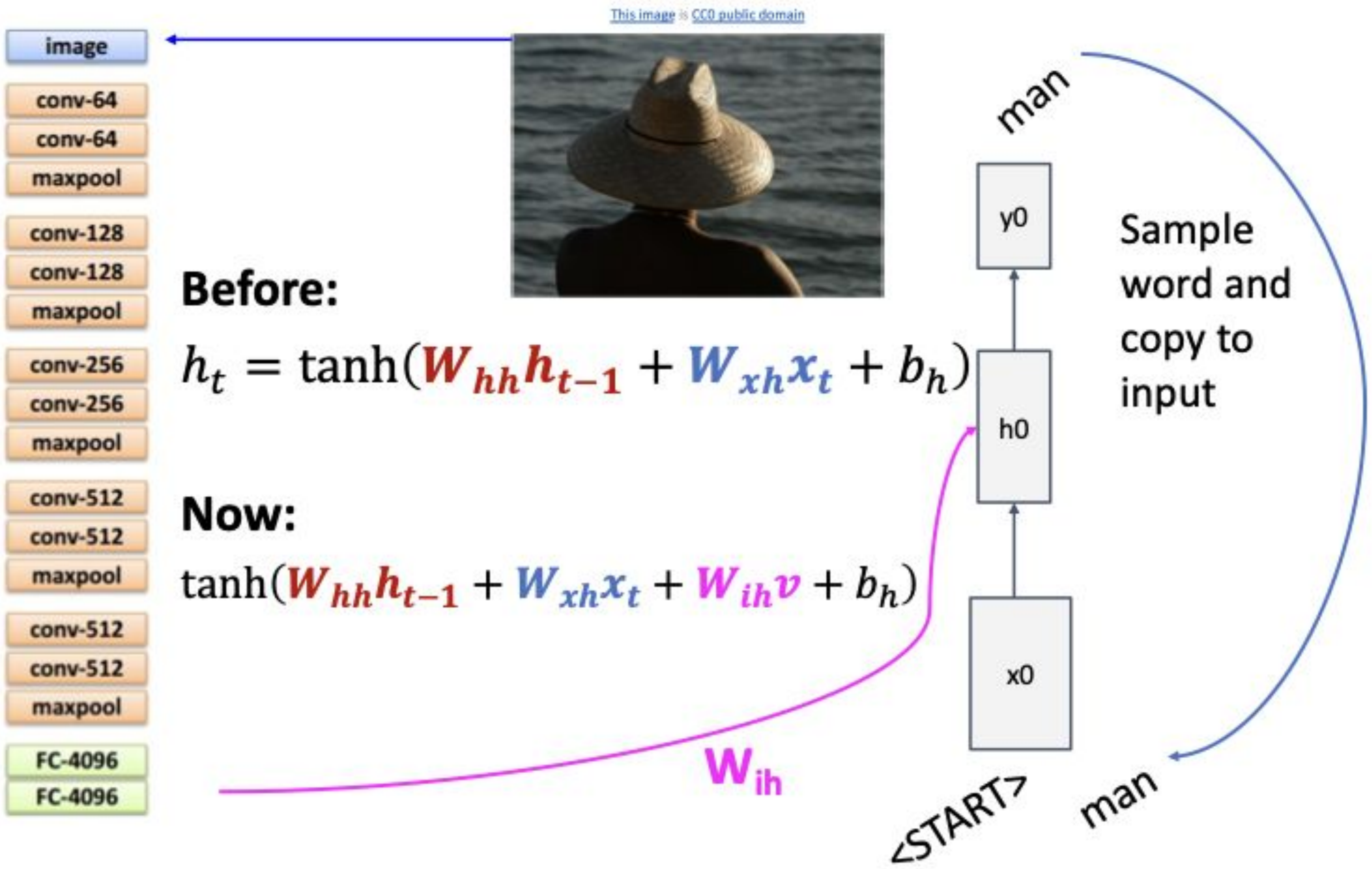
Before:

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

Now:

$$\tanh(W_{hh}h_{t-1} + W_{xh}x_t + W_{ih}v + b_h)$$

Example: Image Captioning



Before:

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

Now:

$$\tanh(W_{hh}h_{t-1} + W_{xh}x_t + W_{ih}v + b_h)$$

Example: Image Captioning

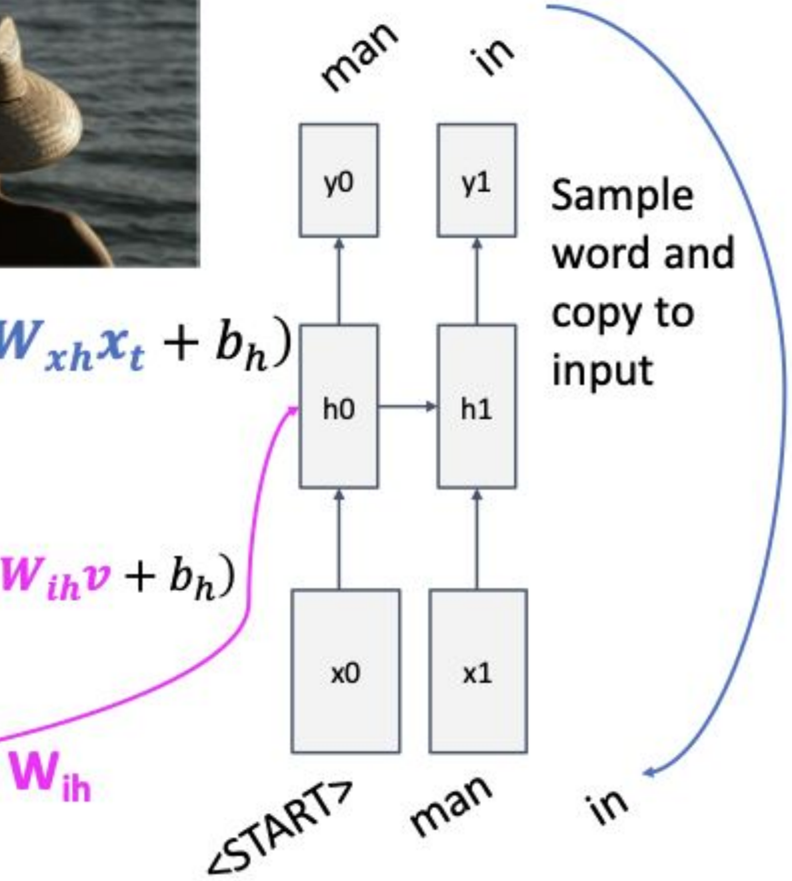
This image is CC0 public domain

Before:

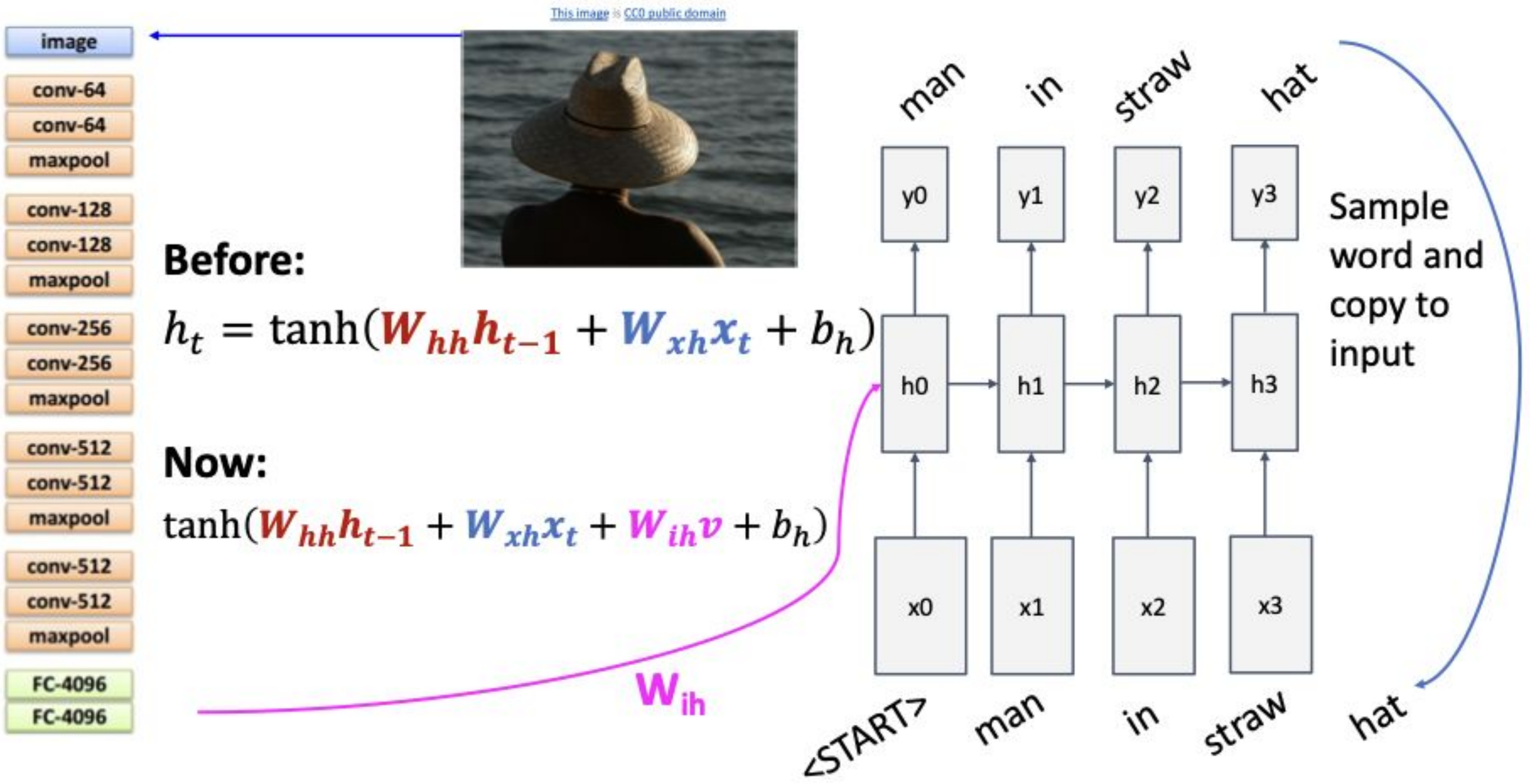
$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t + b_h)$$

Now:

$$\tanh(W_{hh}h_{t-1} + W_{xh}x_t + W_{ih}v + b_h)$$



Example: Image Captioning



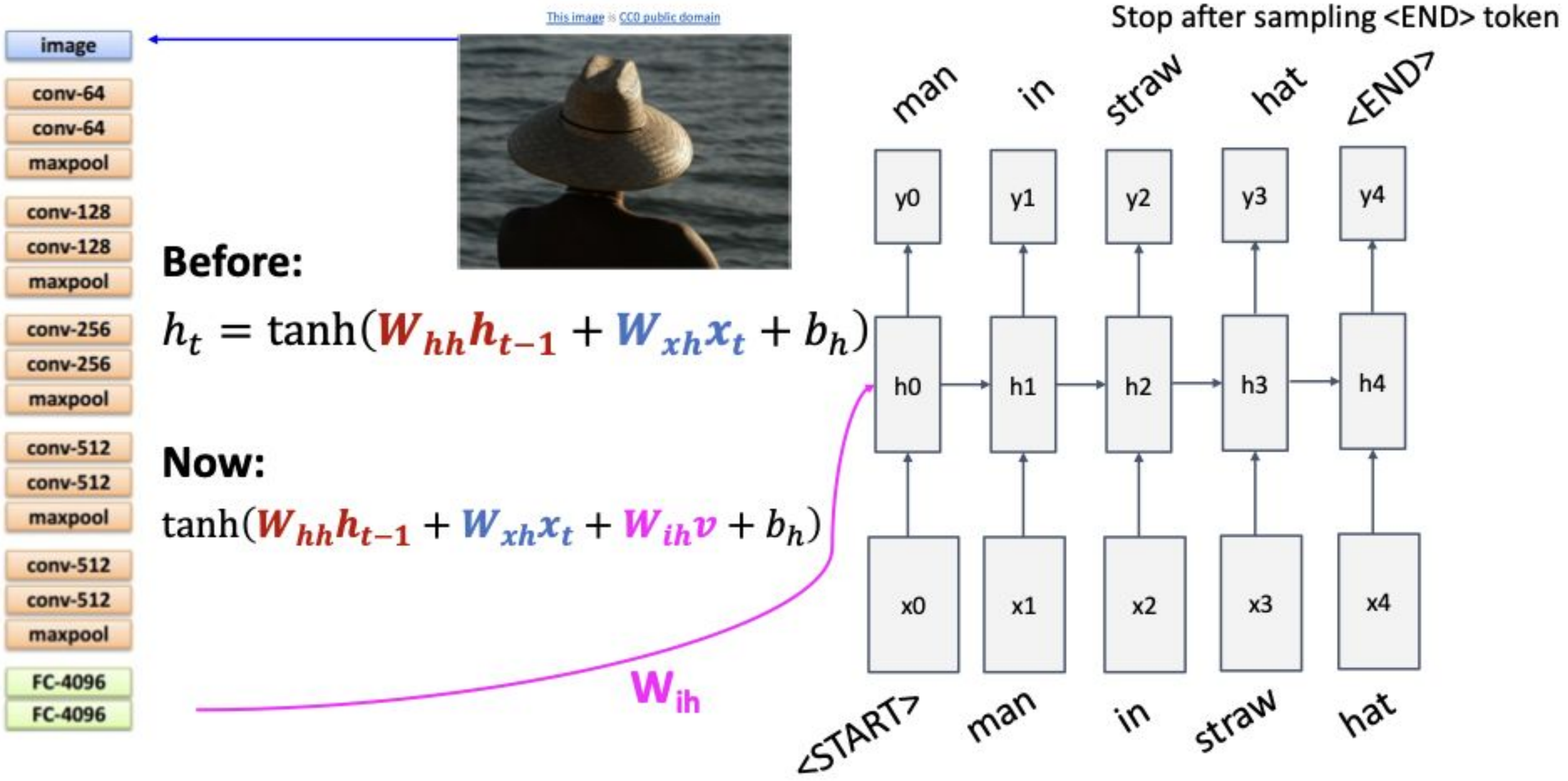
Before:

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

Now:

$$\tanh(W_{hh}h_{t-1} + W_{xh}x_t + W_{ih}v + b_h)$$

Example: Image Captioning



Before:

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

Now:

$$\tanh(W_{hh}h_{t-1} + W_{xh}x_t + W_{ih}v + b_h)$$

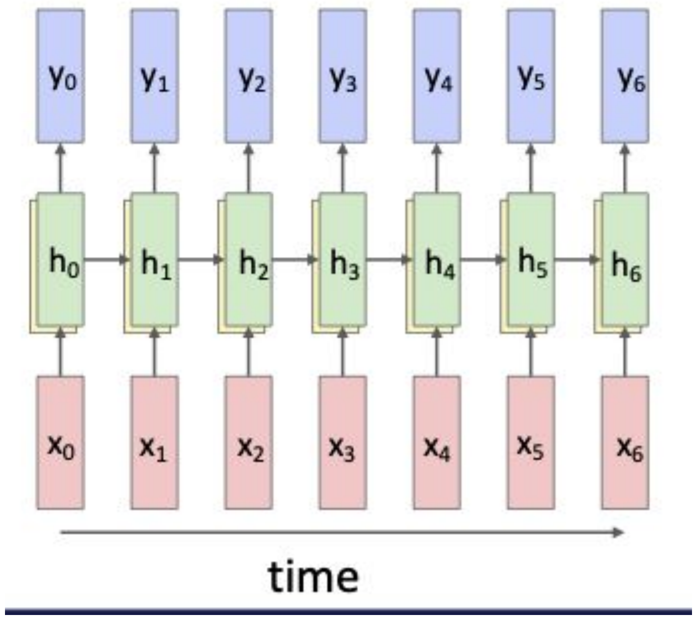
Today

- Practical scenarios where RNN is used
- **RNN Gradients**
- Attention
- Types of attention
- Transformers

So far..

Single-Layer RNNs

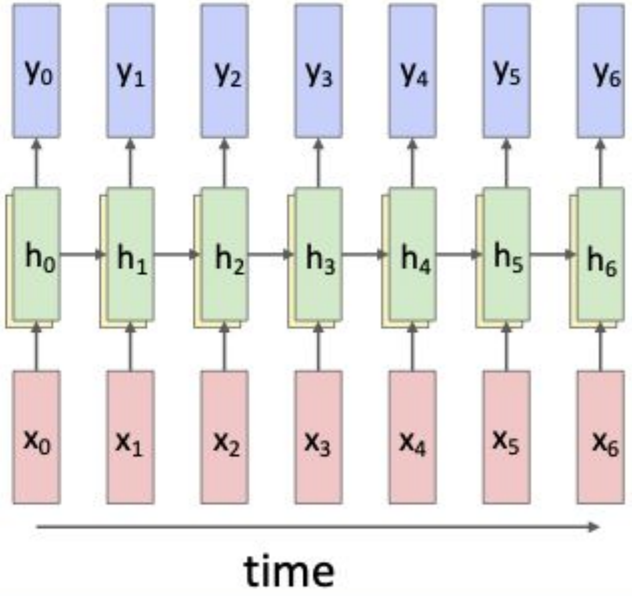
$$\boxed{h_t = \tanh\left(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} + b_h\right)}$$



So far..

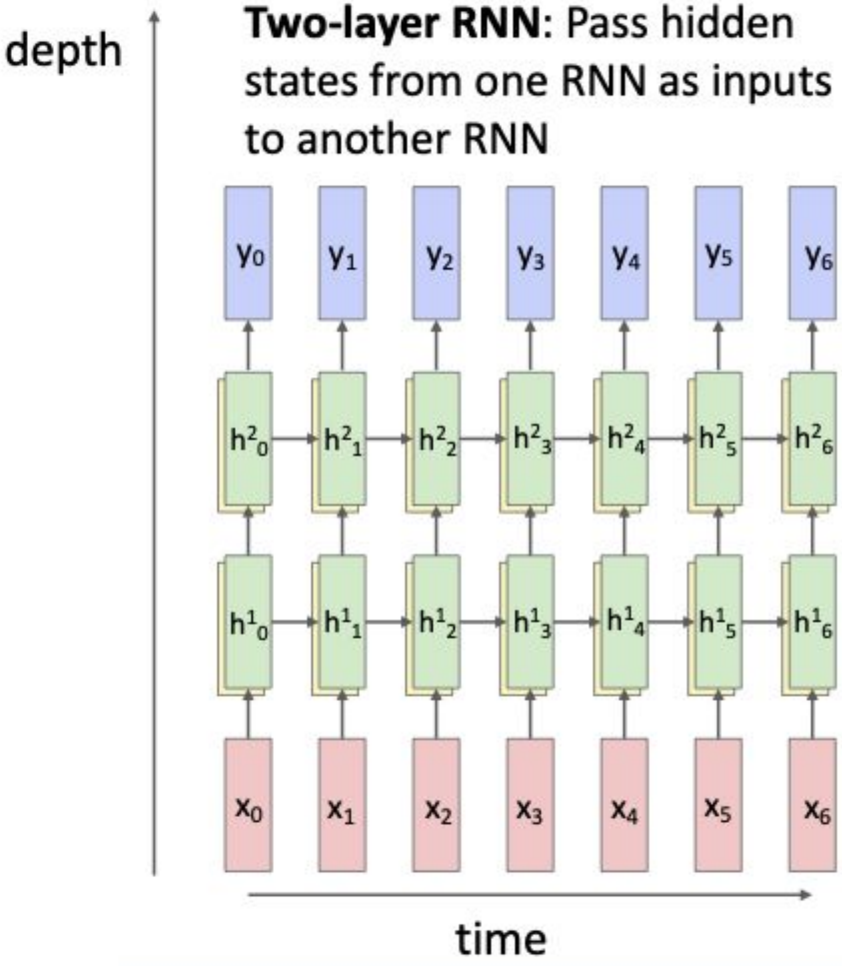
Single-Layer RNNs

$$h_t = \tanh\left(W\begin{pmatrix}h_{t-1}\\ x_t\end{pmatrix} + b_h\right)$$



Mutilayer RNNs

$$\boxed{h_t^{\ell} = \tanh\left(W\begin{pmatrix}h_{t-1}^{\ell}\\ h_t^{\ell-1}\end{pmatrix} + b_h^{\ell}\right)}$$

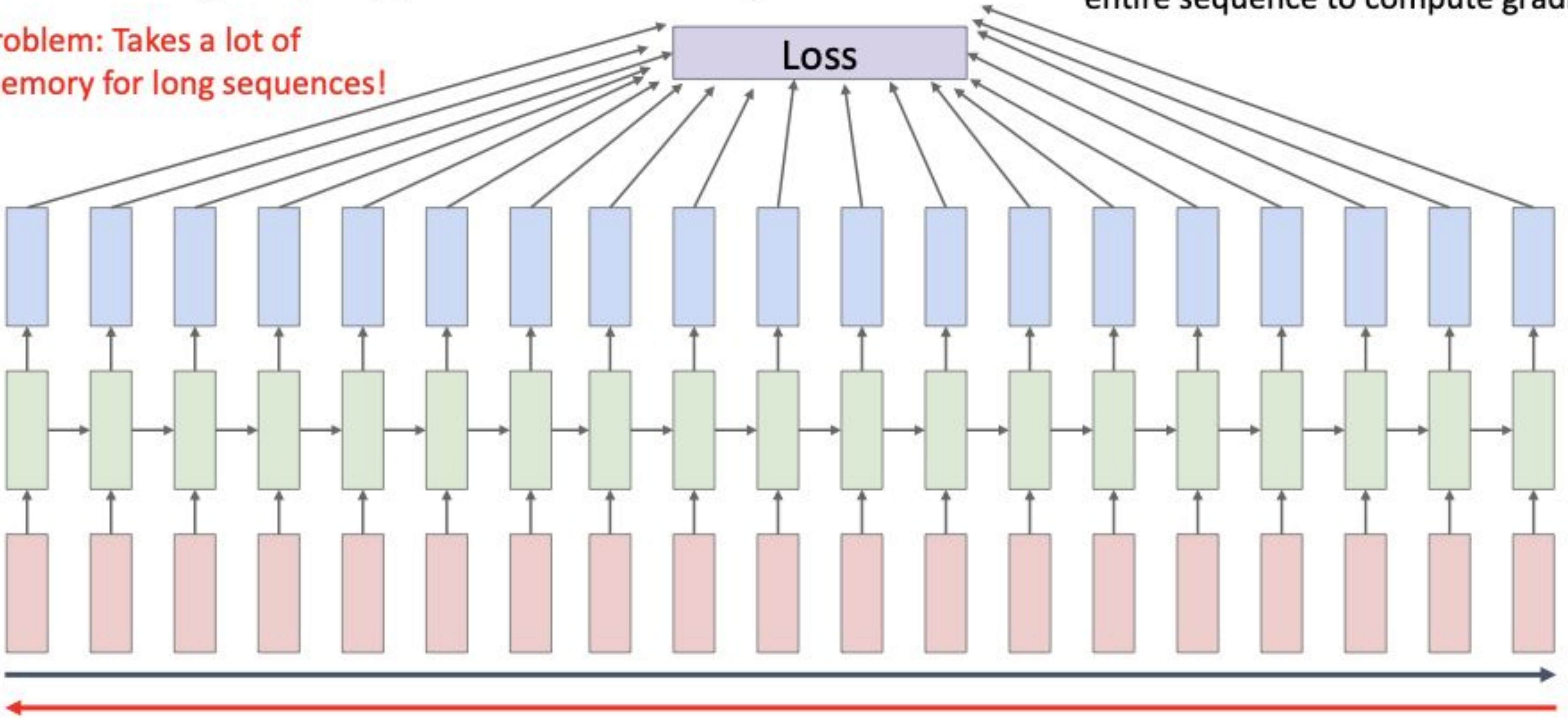


Slide credit: Justin Johnson

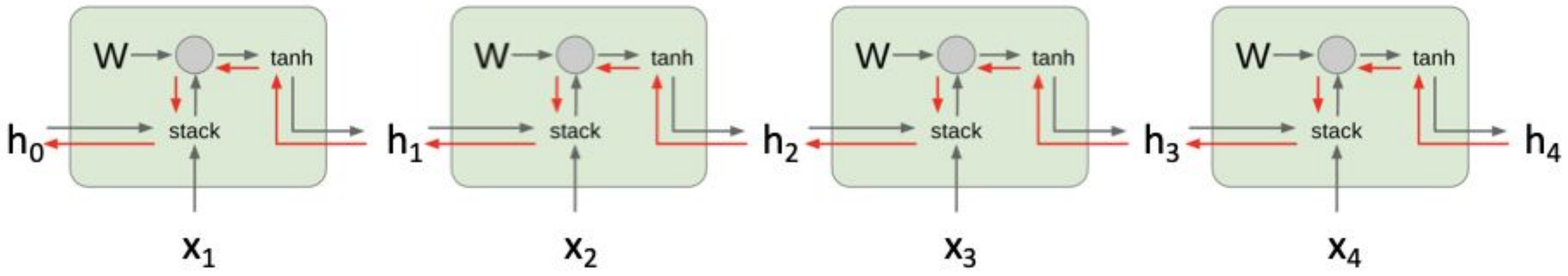
Backpropagation Through Time

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

Problem: Takes a lot of memory for long sequences!

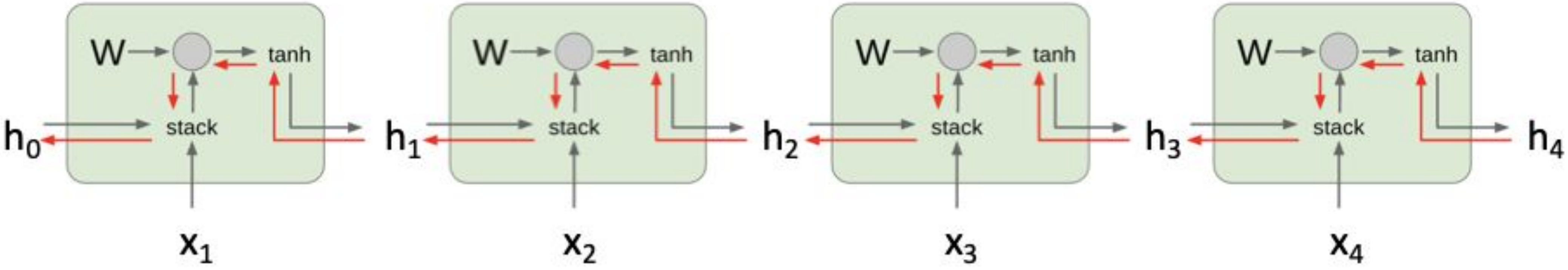


Gradient flow in RNN



Computing gradient of h_0 involves many factors of W (and repeated tanh)

Gradient flow in RNN

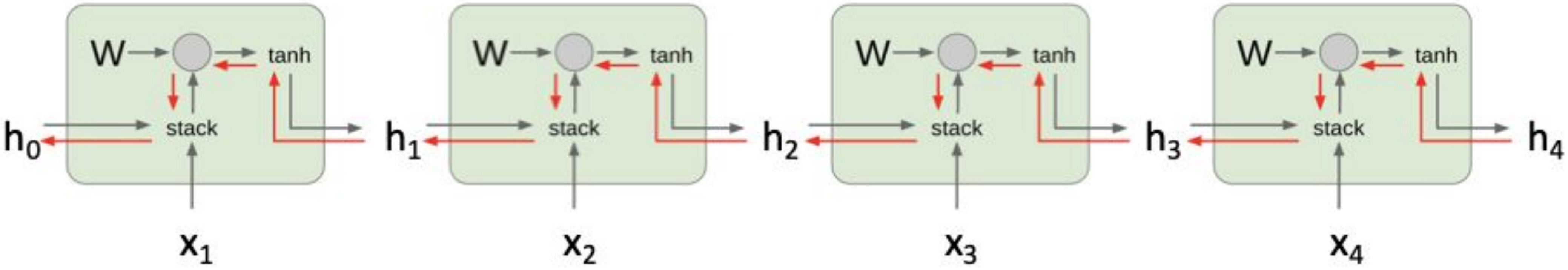


Computing gradient of h_0 involves many factors of W (and repeated tanh)

Largest singular value > 1:
Exploding gradients

Largest singular value < 1:
Vanishing gradients

Gradient flow in RNN



Computing gradient of h_0 involves many factors of W (and repeated tanh)

Largest singular value > 1:
Exploding gradients

Largest singular value < 1:
Vanishing gradients

Gradient clipping

$g = c \cdot g / \|g\|$

What value does gradient clipping offer?

What value does gradient clipping offer?

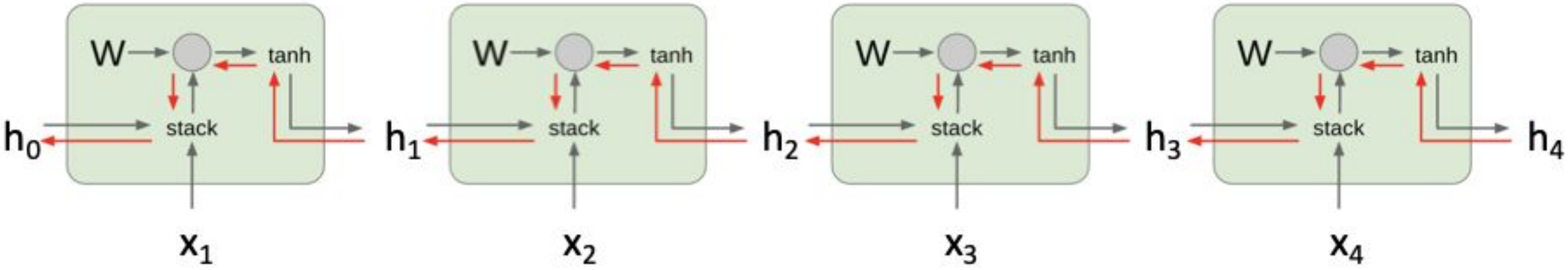
It scales up the norm of the gradient to c
42%

It scales down the norm of the gradient to c ✓
63%

It helps in improving the stability of training ✓
88%

It leads to loss of information and unstable training
4%

Gradient flow in RNN



Computing gradient of h_0 involves many factors of W (and repeated tanh)

Largest singular value > 1:
Exploding gradients

Largest singular value < 1:
Vanishing gradients

Gradient clipping

$g = c \cdot g/\|g\|$

Applied only on gradients whose norm > c

Do not edit
How to change the design

How to identify the hyper parameter c?

Presenting with animations, GIFs or speaker notes? Enable our Chrome extension

slido

By observing average gradient norm

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.
- LSTMs (Long Short-Term Memory) and GRU (Gated Recurrent Units)

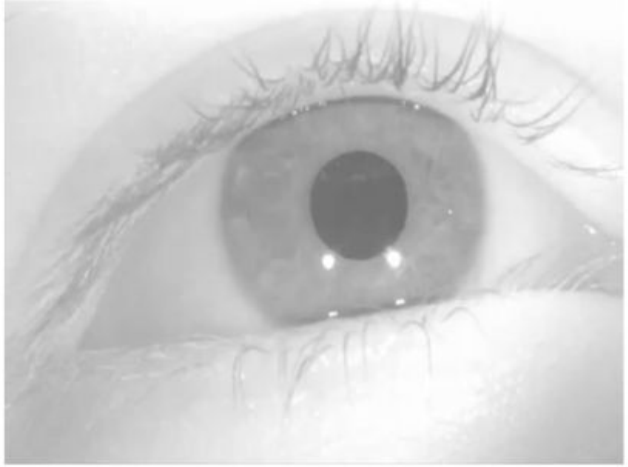
Summary so far: RNNs

- RNNs allow a lot of flexibility in architecture design
- Vanilla RNNs are simple but don't work very well
- Common to use LSTM or GRU: additive interactions improve gradient flow
- Backward flow of gradients in RNN can explode or vanish.
  - Exploding is controlled with gradient clipping.
  - Vanishing is controlled with additive interactions (LSTM)

Today

- Practical scenarios where RNN is used
- RNN Gradients
- **Attention**
- Types of attention
- Transformers

Saccades :: Attention weights



Learn to ***attend*** to different parts of the image.

Slide credit: Justin Johnson

Three Ways of Processing Sequences

Recurrent Neural Network | 1D Convolu^on 1D Convolution | Self-Acen^on | Self-Attention

Attention is all you need

Vaswani et al, NeurIPS 2017

Works on **Ordered Sequences**

(+) Good at long sequences: After one RNN layer, h_T "sees" the whole sequence

(-) Not parallelizable: need to compute hidden states sequentially

Works on **Mul+dimensional Grids**

(-) Bad at long sequences: Need to stack many conv layers for outputs to "see" the whole sequence

(+) Highly parallel: Each output can be computed in parallel

Works on **Sets of Vectors**

(-) Good at long sequences: after one self-attention layer, each output "sees" all inputs!

(+) Highly parallel: Each output can be computed in parallel

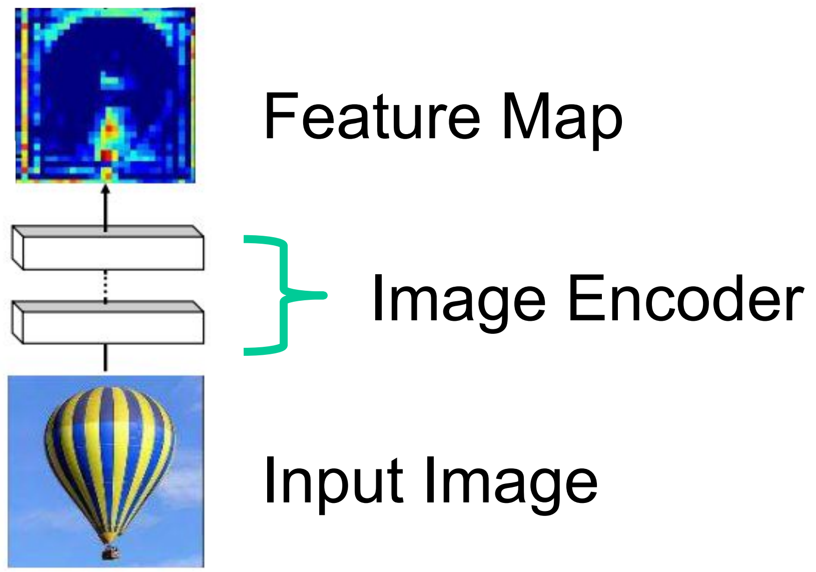
(-) Very memory intensive

Main Idea

Input Image

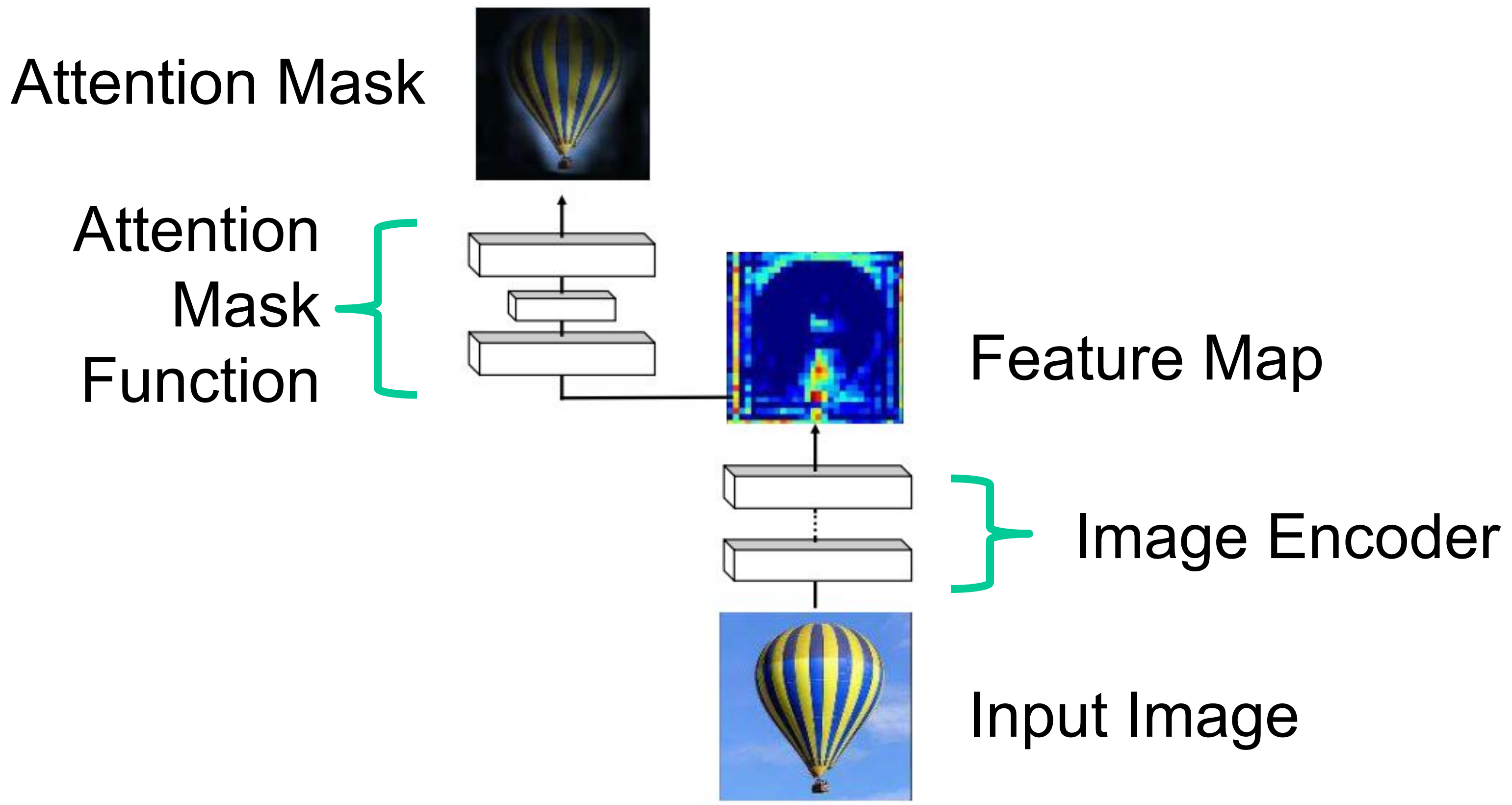
Wang et al, “Residual Attention Network for Image Classification”, CVPR 2017

Main Idea



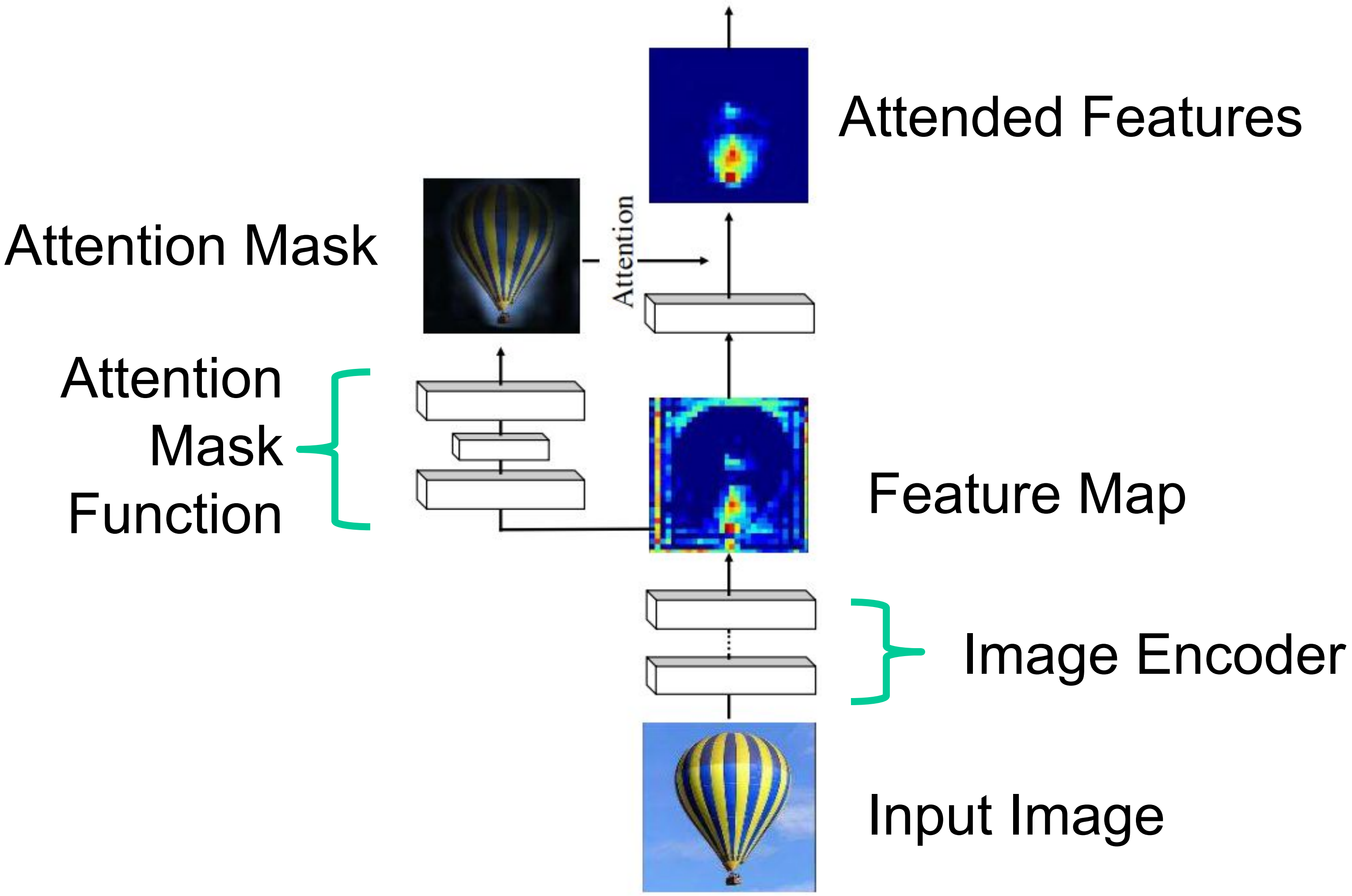
Wang et al, “Residual Attention Network for Image Classification”, CVPR 2017

Main Idea



Wang et al, "Residual Attention Network for Image Classification", CVPR 2017

Main Idea



Wang et al, "Residual Attention Network for Image Classification", CVPR 2017

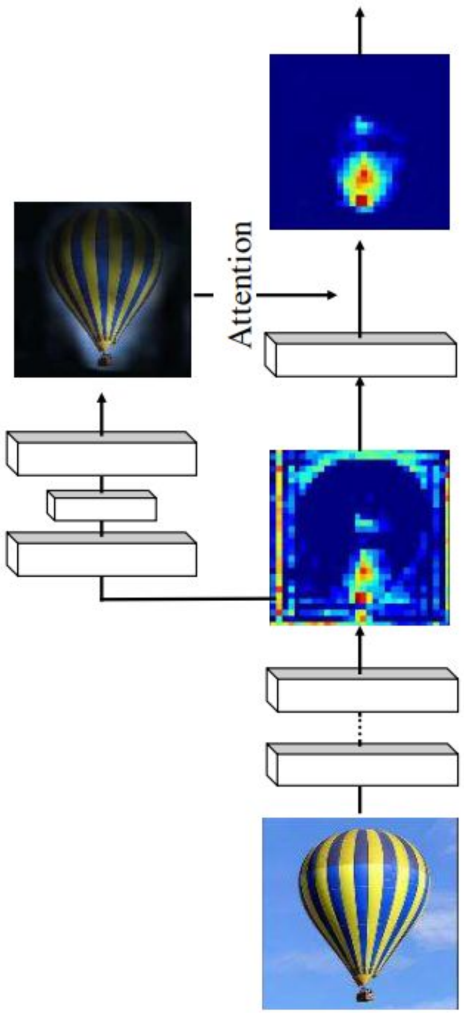
Main Idea

Attended Features

Attention Mask

Cosine similarity

Attention Mask Function



Feature Map

Image Encoder

Input Image

Wang et al, "Residual Attention Network for Image Classification", CVPR 2017

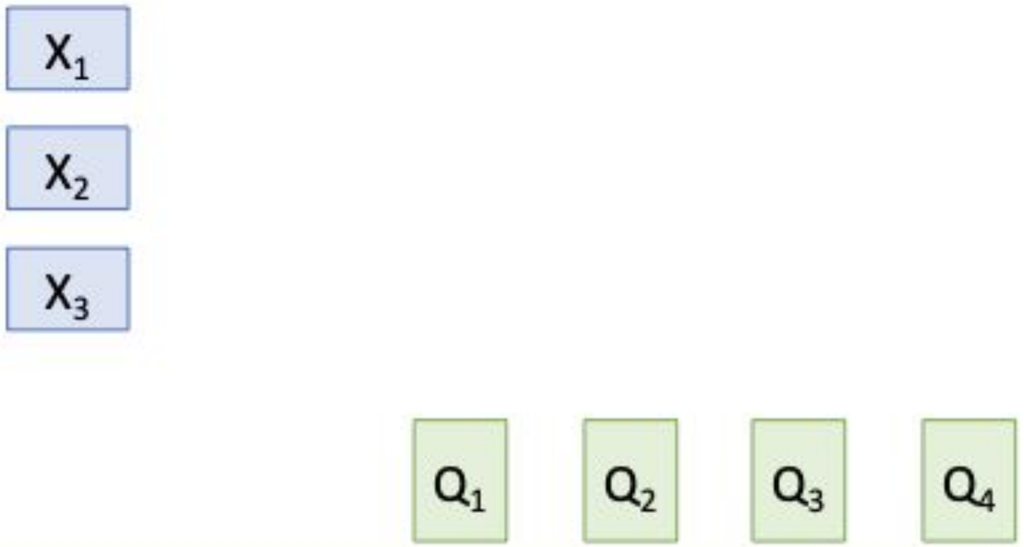
Attention in 3 minutes

Q = query
K = key
V = value

- Q = query
- K = key
- V = value

Attention Layer

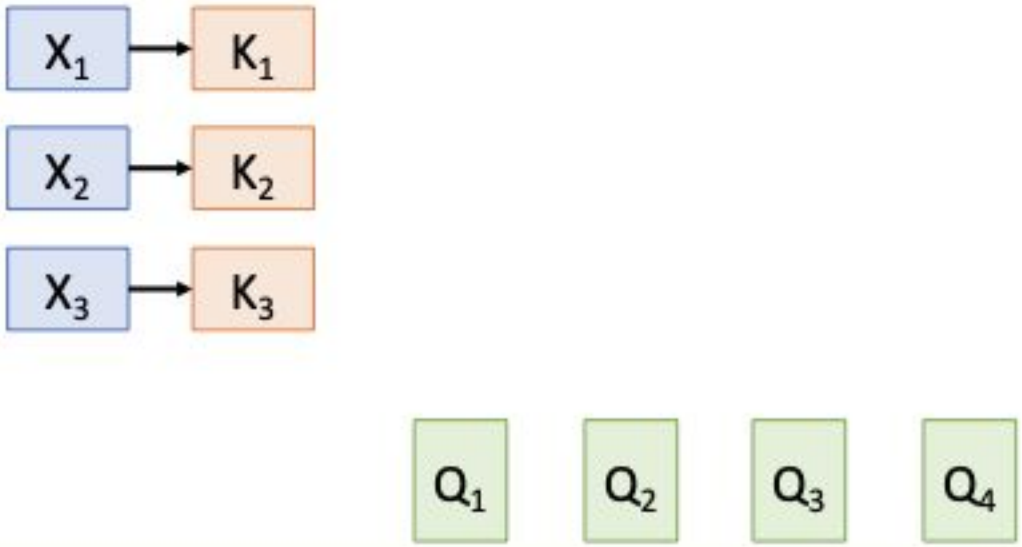
Inputs:
Query vectors: Q (Shape: N_Q x D_Q)
Input vectors: X (Shape: N_X x D_X)
Key matrix: W_K (Shape: D_X x D_Q)
Value matrix: W_V (Shape: D_X x D_V)



- Q = query
- K = key
- V = value

Attention Layer

Inputs:
Query vectors: Q (Shape: $N_Q \times D_Q$)
Input vectors: X (Shape: $N_X \times D_X$)
Key matrix: W_K (Shape: $D_X \times D_Q$)
Value matrix: W_V (Shape: $D_X \times D_V$)

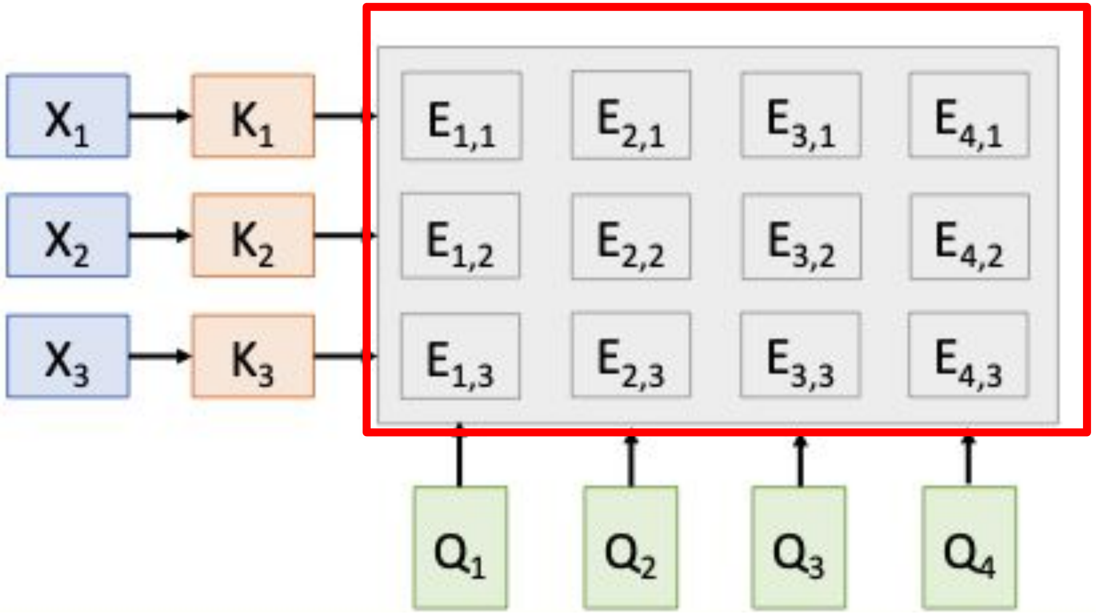


- Q = query
- K = key
- V = value

Attention Layer

Inputs:
Query vectors: Q (Shape: N_Q x D_Q)
Input vectors: X (Shape: N_X x D_X)
Key matrix: W_K (Shape: D_X x D_Q)
Value matrix: W_V (Shape: D_X x D_V)

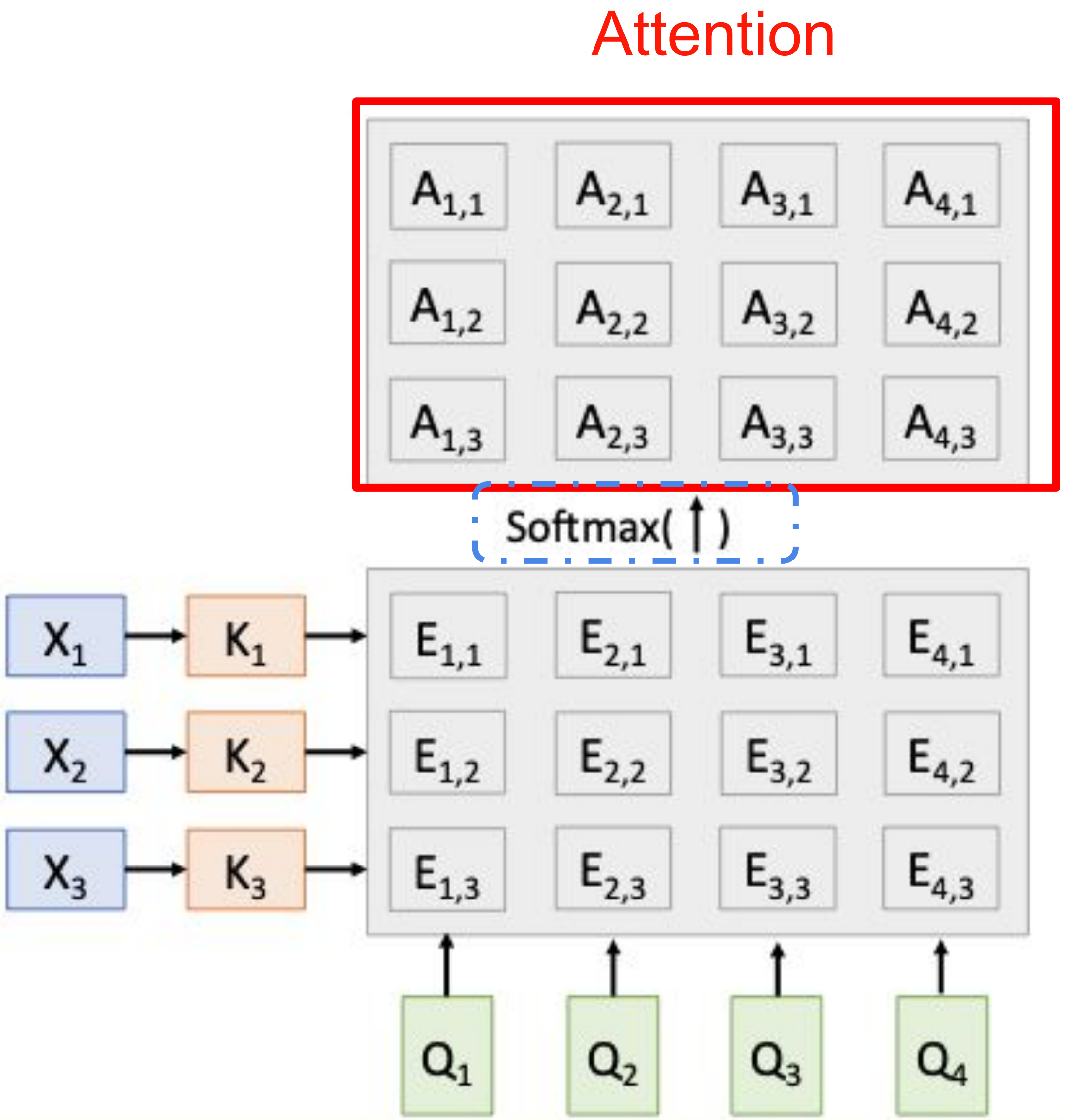
Similarities (query, key)



- Q = query
- K = key
- V = value

Attention Layer

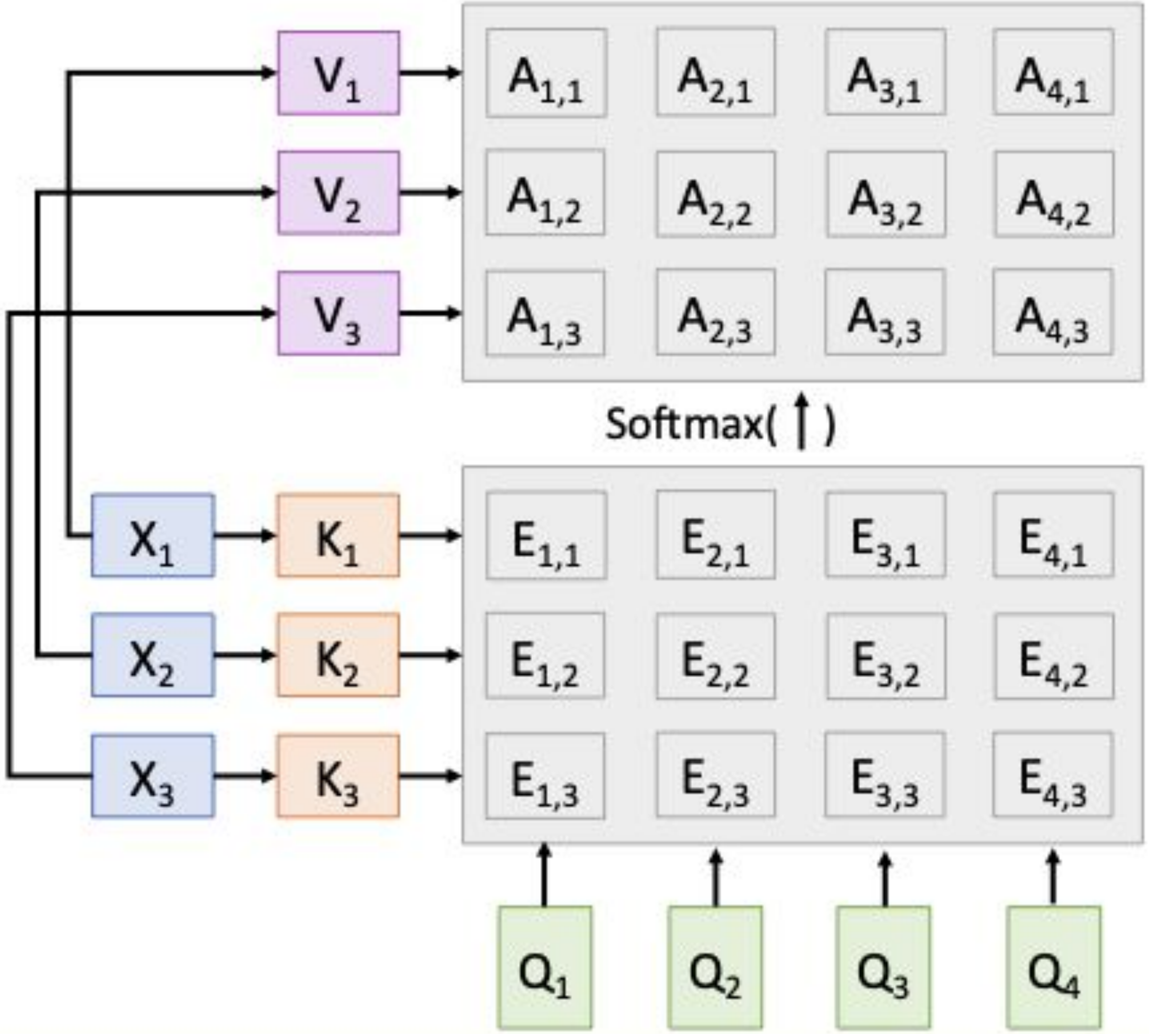
Inputs:
Query vectors: Q (Shape: N_Q x D_Q)
Input vectors: X (Shape: N_X x D_X)
Key matrix: W_K (Shape: D_X x D_Q)
Value matrix: W_V (Shape: D_X x D_V)



- **Q = query**
- **K = key**
- **V = value**

Attention Layer

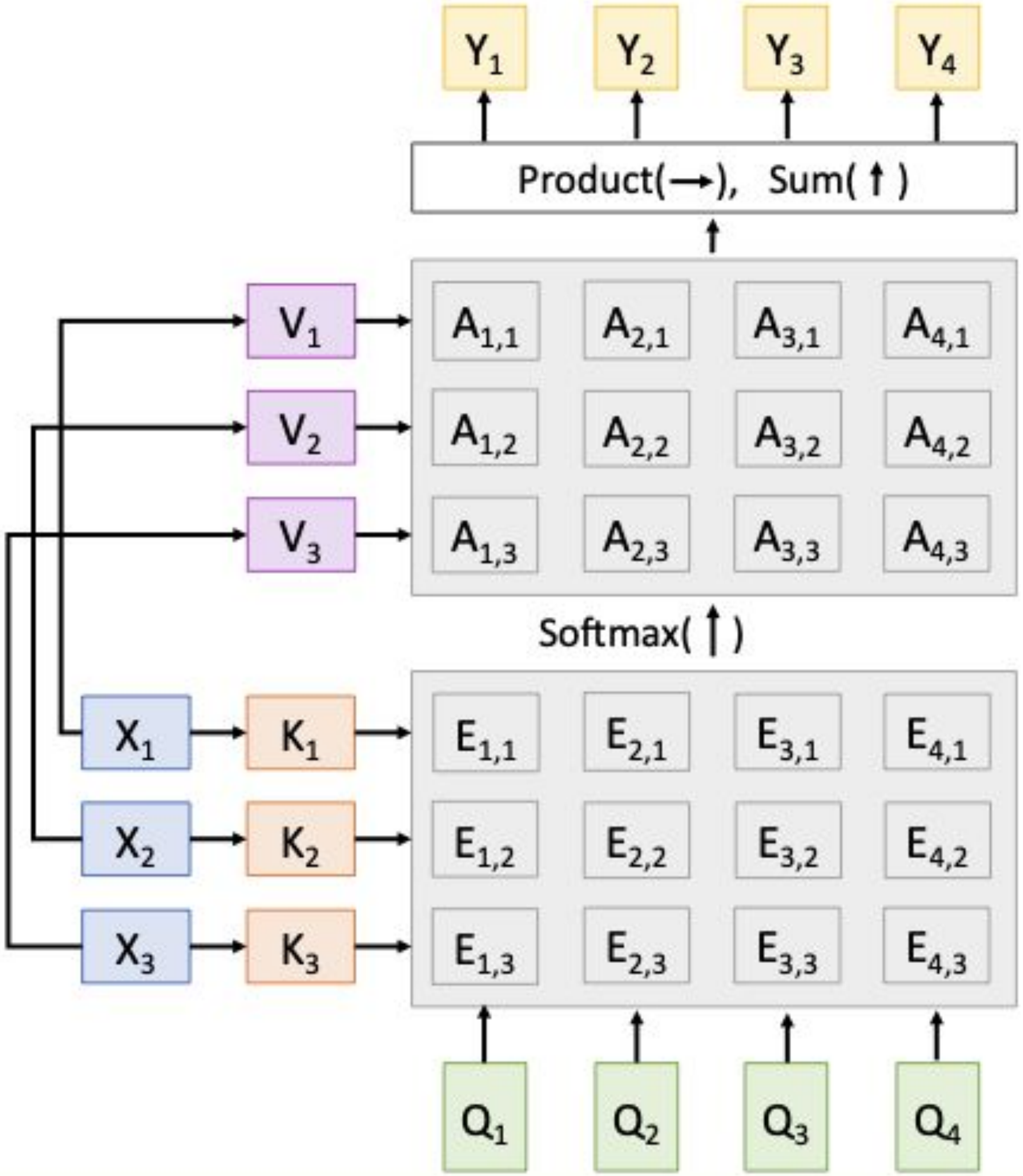
Inputs:
Query vectors: Q (Shape: N_Q x D_Q)
Input vectors: X (Shape: N_X x D_X)
Key matrix: W_K (Shape: D_X x D_Q)
Value matrix: W_V (Shape: D_X x D_V)



Attention Layer

Inputs:
Query vectors: **Q** (Shape: N_Q x D_Q)
Input vectors: **X** (Shape: N_X x D_X)
Key matrix: W_K (Shape: D_X x D_Q)
Value matrix: W_V (Shape: D_X x D_V)

Computation:
Key vectors: **K** = XW_K (Shape: N_X x D_Q)
Value Vectors: **V** = XW_V (Shape: N_X x D_V)
Similarities: E = QK^T / D_Q (Shape: N_Q x N_X) $E_{i,j} = (Q_i \cdot K_j) / D_Q$
Attention weights: A = softmax(E, dim=1) (Shape: N_Q x N_X)
Output vectors: Y = AV (Shape: N_Q x D_V) $Y_i = \sum_j A_{i,j} V_j$



Today

- Practical scenarios where RNN is used
- RNN Gradients
- Attention
- **Types of attention**
- Transformers

Types of attention

Self attention

- Query = Key = Value = image tokens.
- **Motivation:** Give maximum flexibility to attend and use the same input in different ways.

Cross-attention

- Key = Value = image tokens
- Query = something else

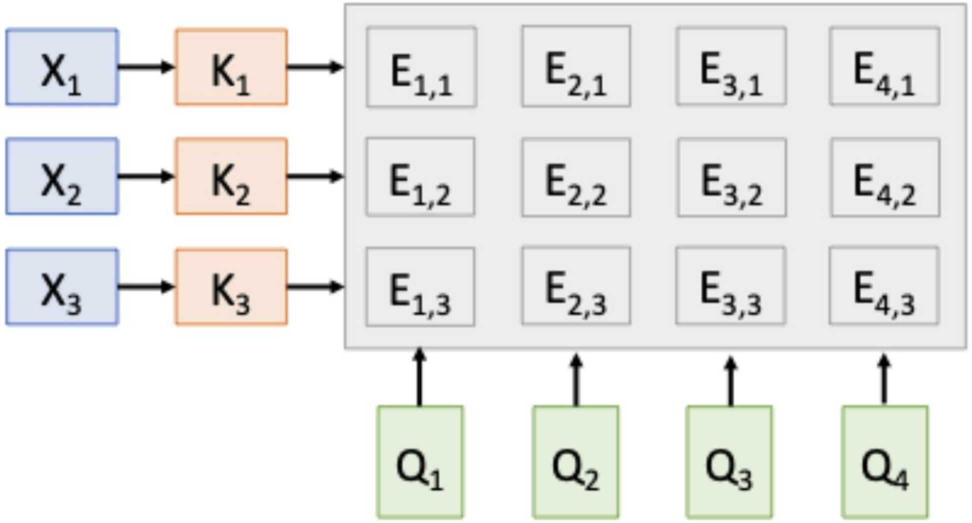
Lets compute a similarity matrix (E)

X = [0.8 0.2 0.1]

X = 1 X 3 matrix

In self-attention, query = key = value = X

2 minutes, enter in slido in the next slide ⏰



The similarity matrix

The similarity matrix

Size: 1X3: [0.16 0.4 0.2]

6%

Size: 3X1: [0.64 0.04 0.01]

2%

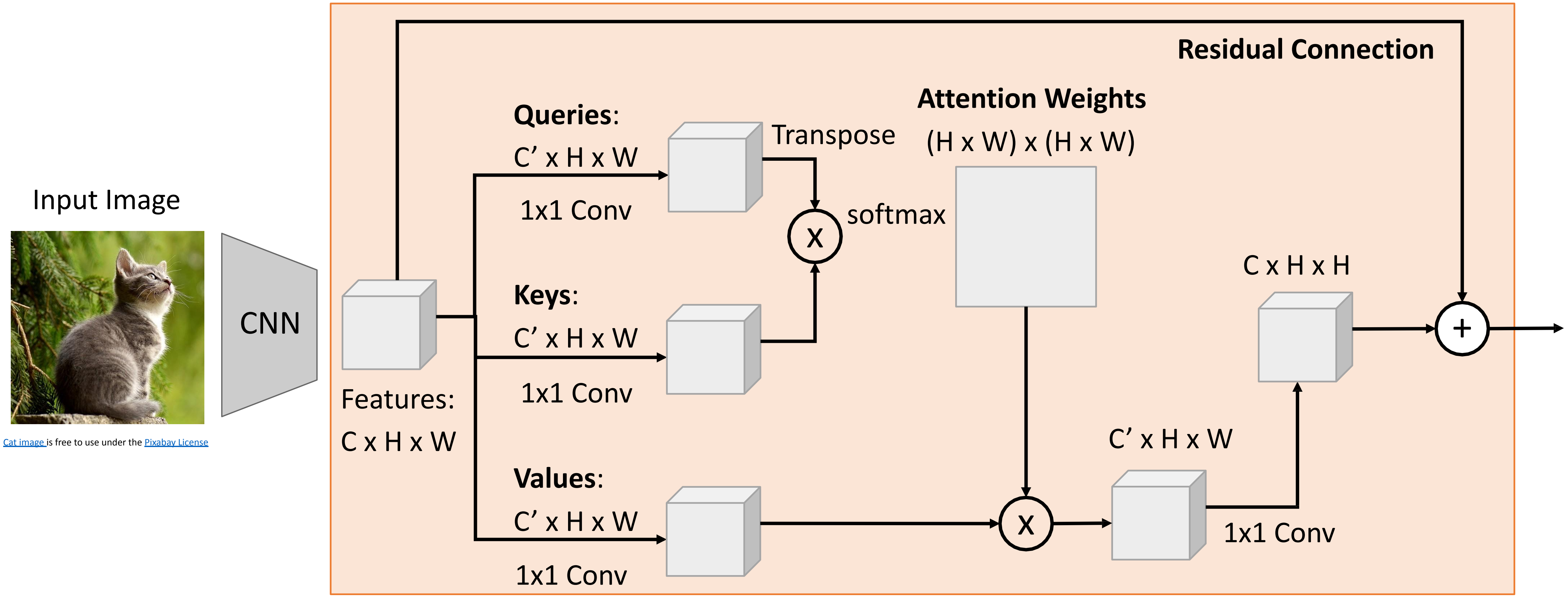
Size: 3X3: [[0.16 1 0.9], [1 0.4 0.3], [0.9 0.3 0.2]]

23%

Size: 3X3: [[0.64 0.16 0.08], [0.16 0.04 0.02] [0.08 0.02 0.01]]

69%

Takeaway: The concept of attention can be applicable to any architecture.



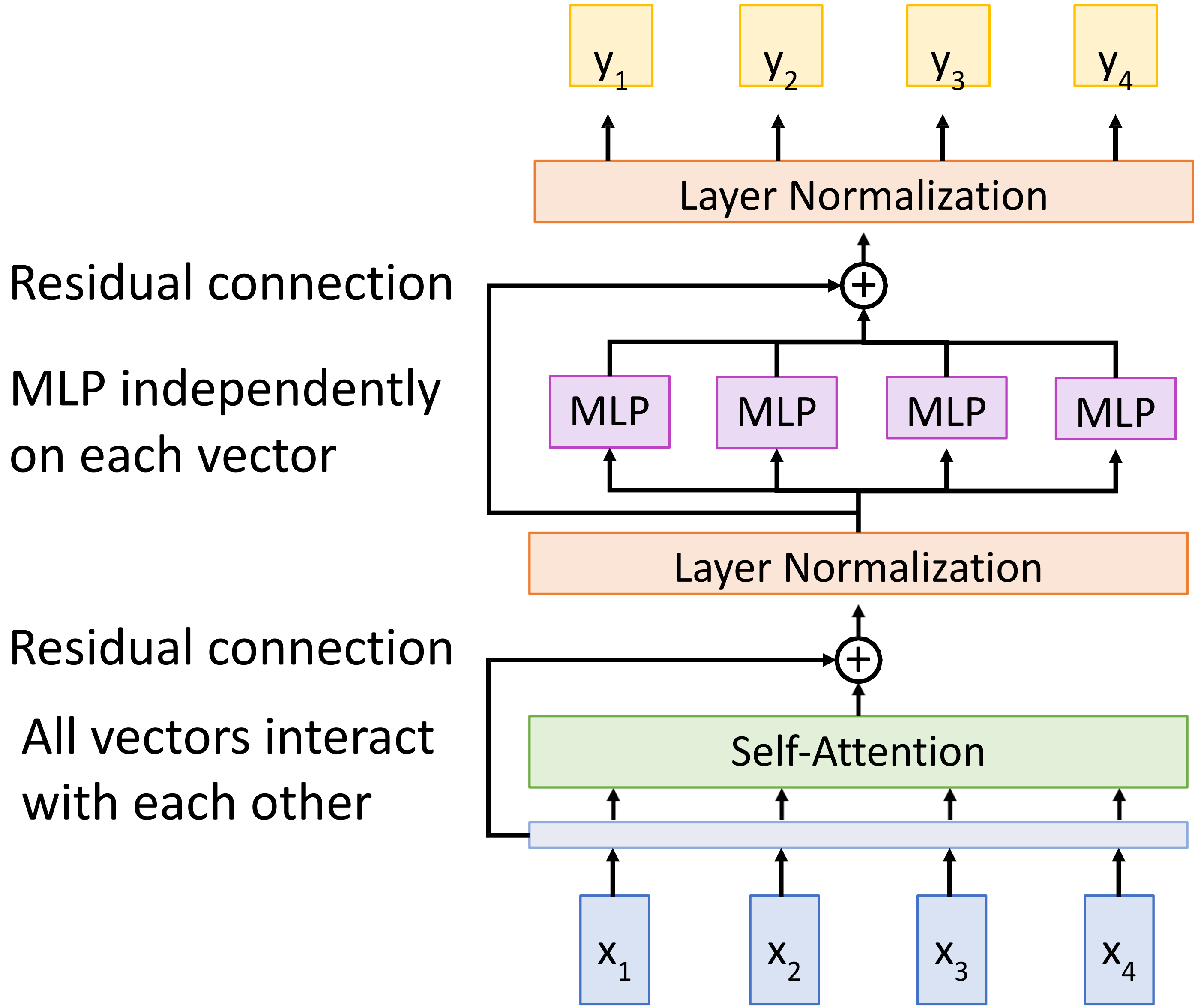
Self-Attention Module

Zhang et al, “Self-Attention Generative Adversarial Networks”, ICML 2018

Transformers

Many slides adapted from Justin Johnson

The Transformer

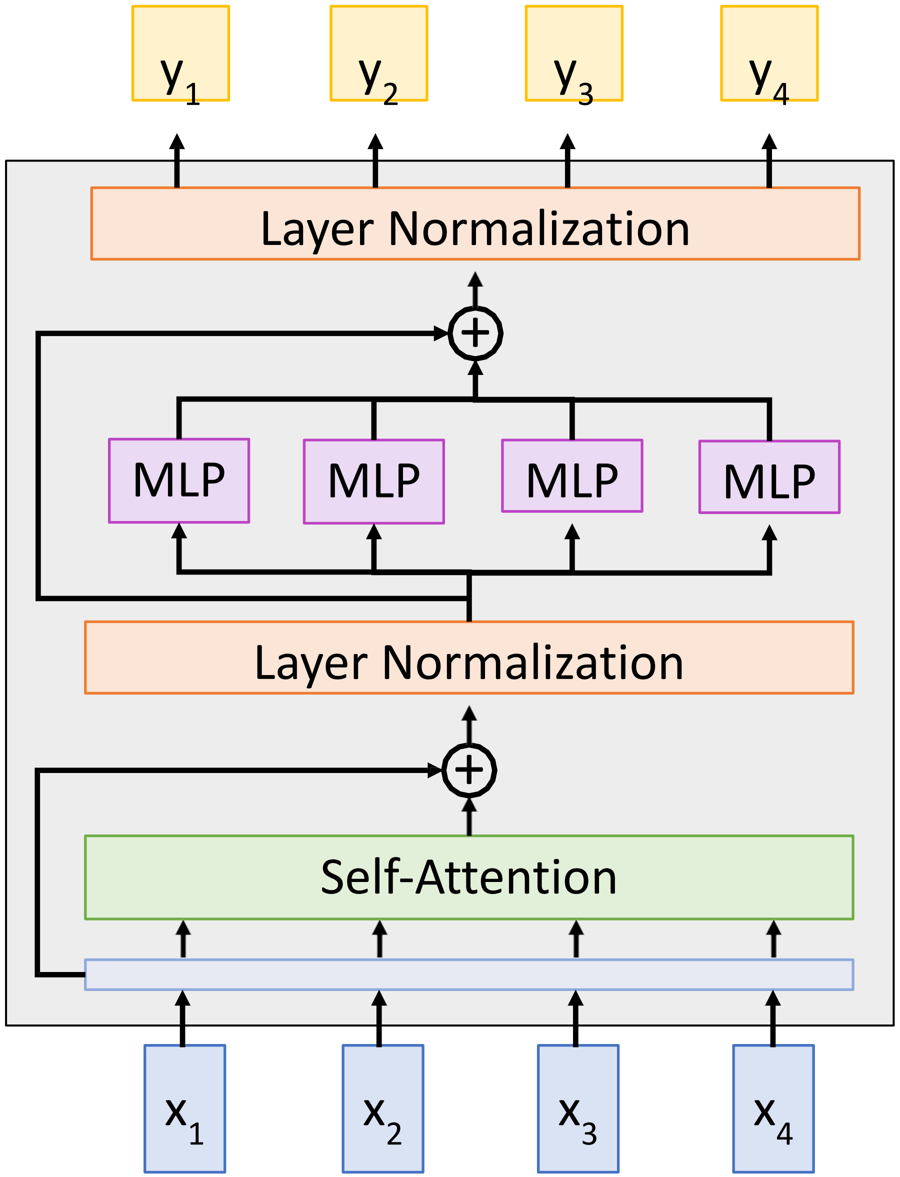


Vaswani et al, “Attention is all you need”, NeurIPS 2017

The Transformer

<u>Transformer Block</u>:
Input: Set of vectors x
Output: Set of vectors y

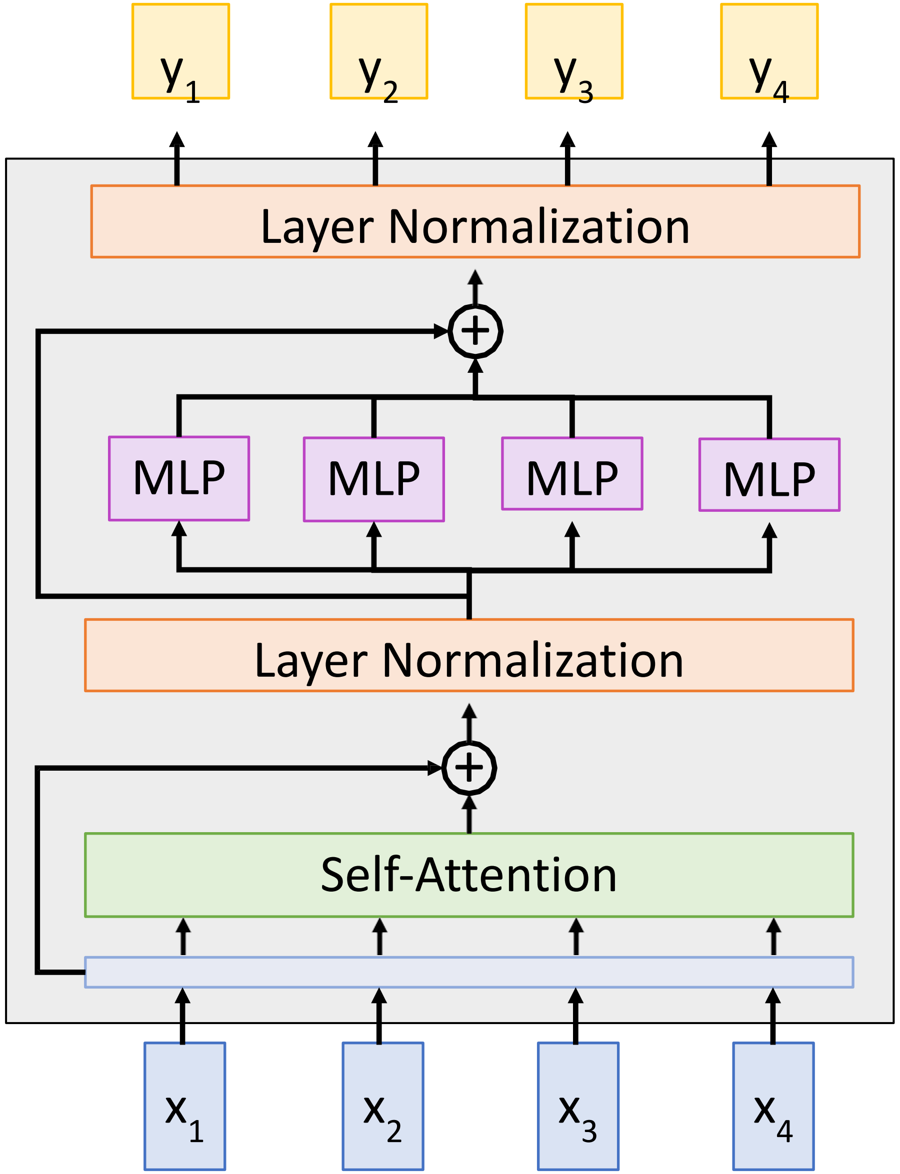
Self-attention is the only interaction between vectors!



Vaswani et al, “Attention is all you need”, NeurIPS 2017

Post-Norm Transformer

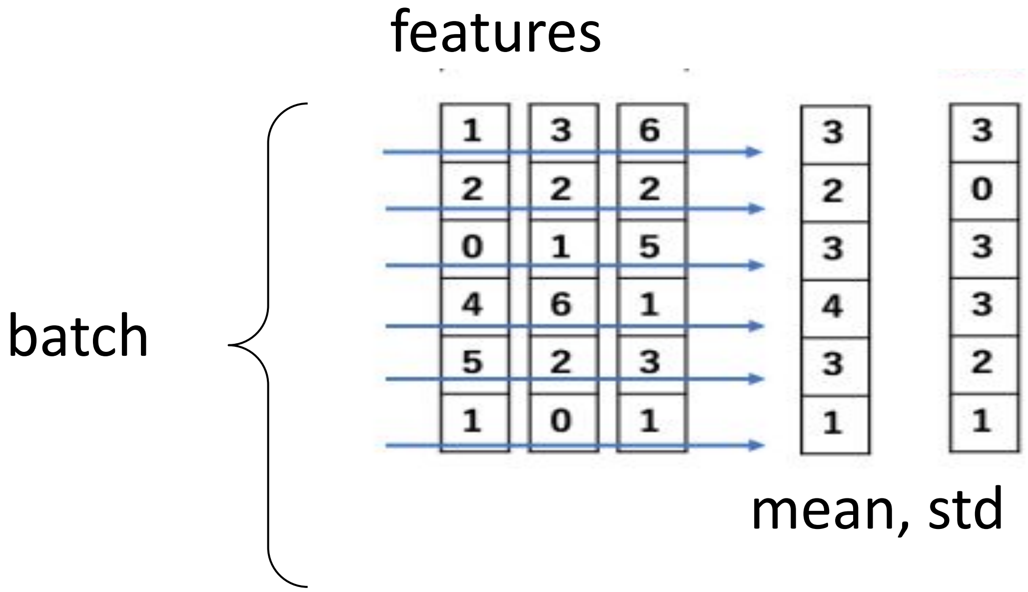
Layer normalization is **after** residual connections



Vaswani et al, "Attention is all you need", NeurIPS 2017

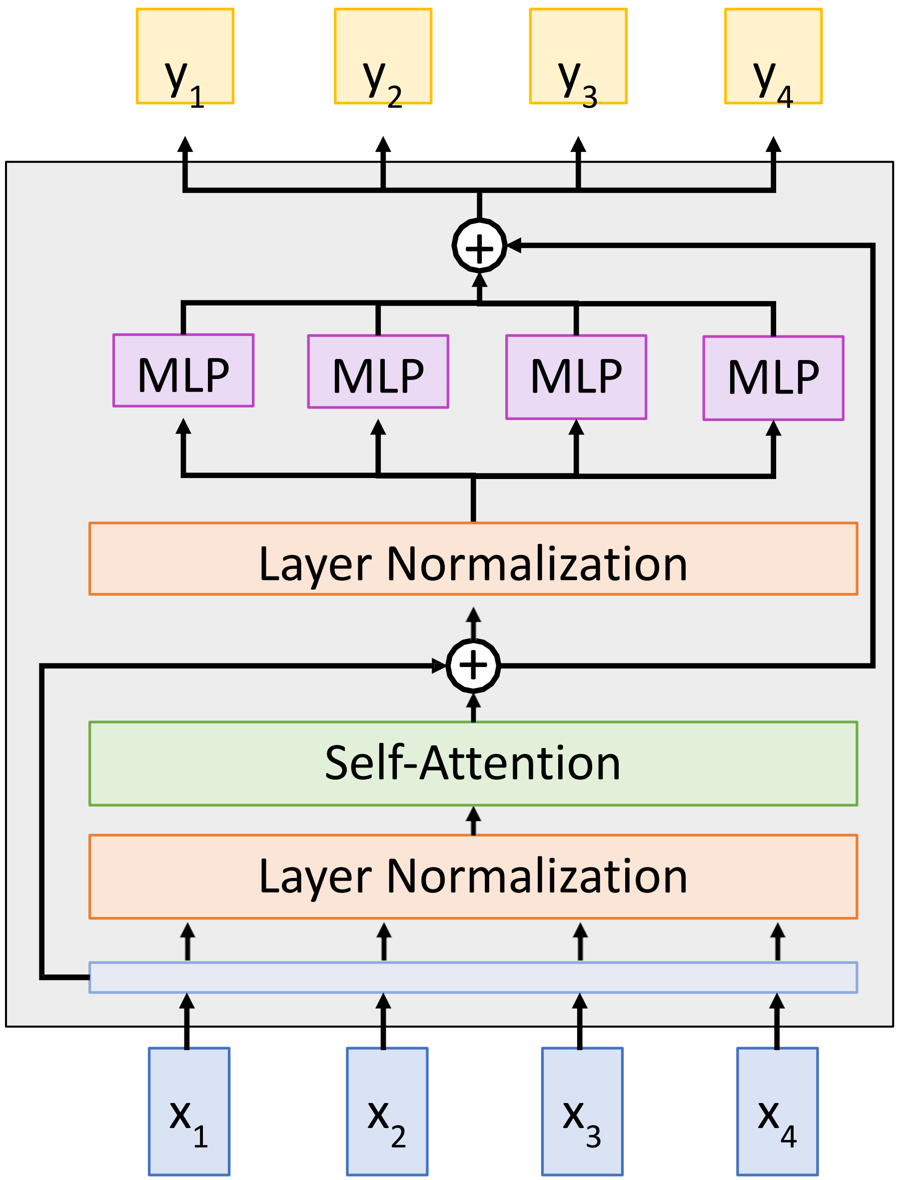
Recall: Layer Normalization in 2D

- Used for feature dimension for a single sample



- Same mean and variance for all features

Pre-Norm Transformer



Layer normalization is **inside** residual connections

Gives more stable training, commonly used in practice

Baevski & Auli, "Adaptive Input Representations for Neural Language Modeling", arXiv 2018

The Transformer

<u>Transformer Block</u>:
Input: Set of vectors x
Output: Set of vectors y

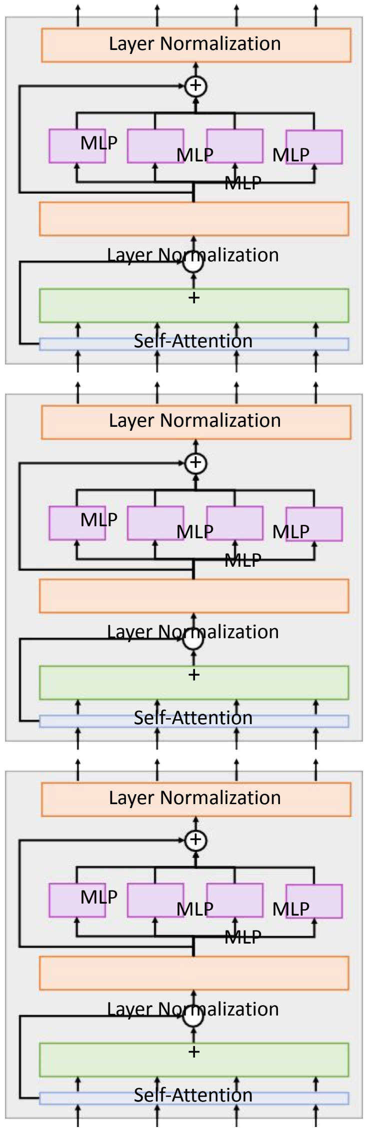
Self-attention is the only interaction between vectors!

Layer norm and MLP work independently per vector

Highly scalable, highly parallelizable

A **Transformer** is a sequence of transformer blocks

Vaswani et al:
12 blocks, D_Q=512, 6 heads



Vaswani et al, "Attention is all you need", NeurIPS 2017

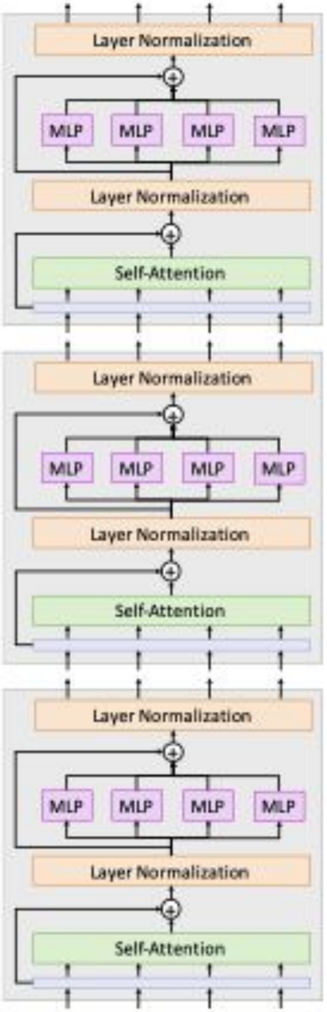
The Transformer: Transfer Learning

"ImageNet Moment for Natural Language Processing"

Pretraining:
Download a lot of text from the internet

Train a giant Transformer model for language modeling

Finetuning:
Fine-tune the Transformer on your own NLP task

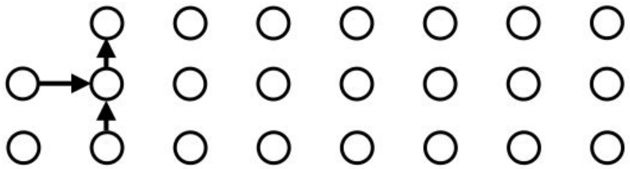


Devlin et al, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", EMNLP 2018

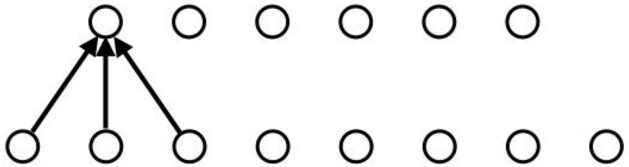
What are some of the advantages that transformers offer over convolutional networks?

Modeling arbitrarily long sequences

- RNNs — recurrent weights are shared across time



- Convolution — conv weights are shared across time



- Attention — weights are dynamically determined