UNIVERSITÄT
HEIDELBERG

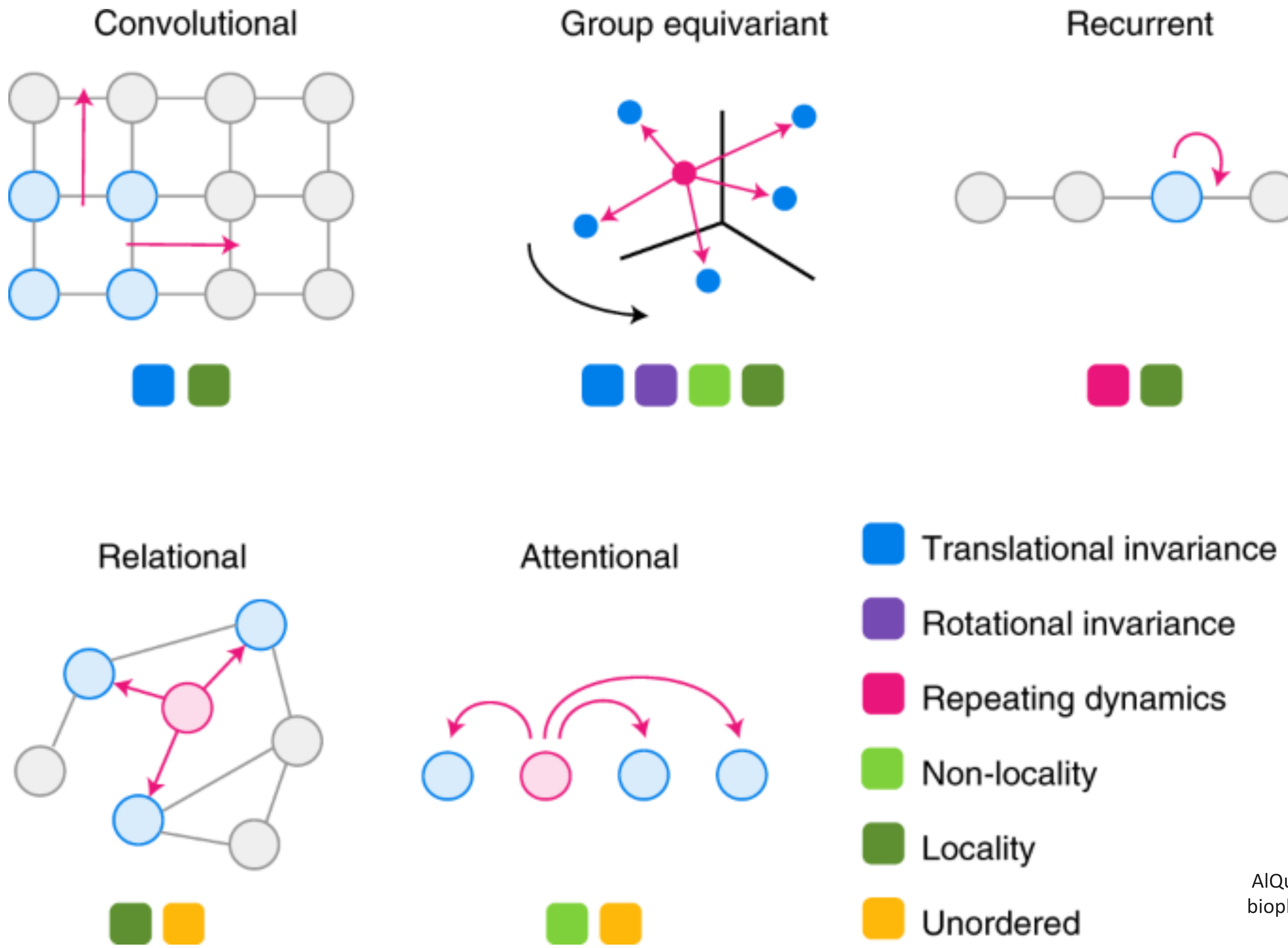
A Zoo of Models

L3, Structural Bioinformatics

WiSe 2023/24, Heidelberg Universitys

How to make sense of all these models?

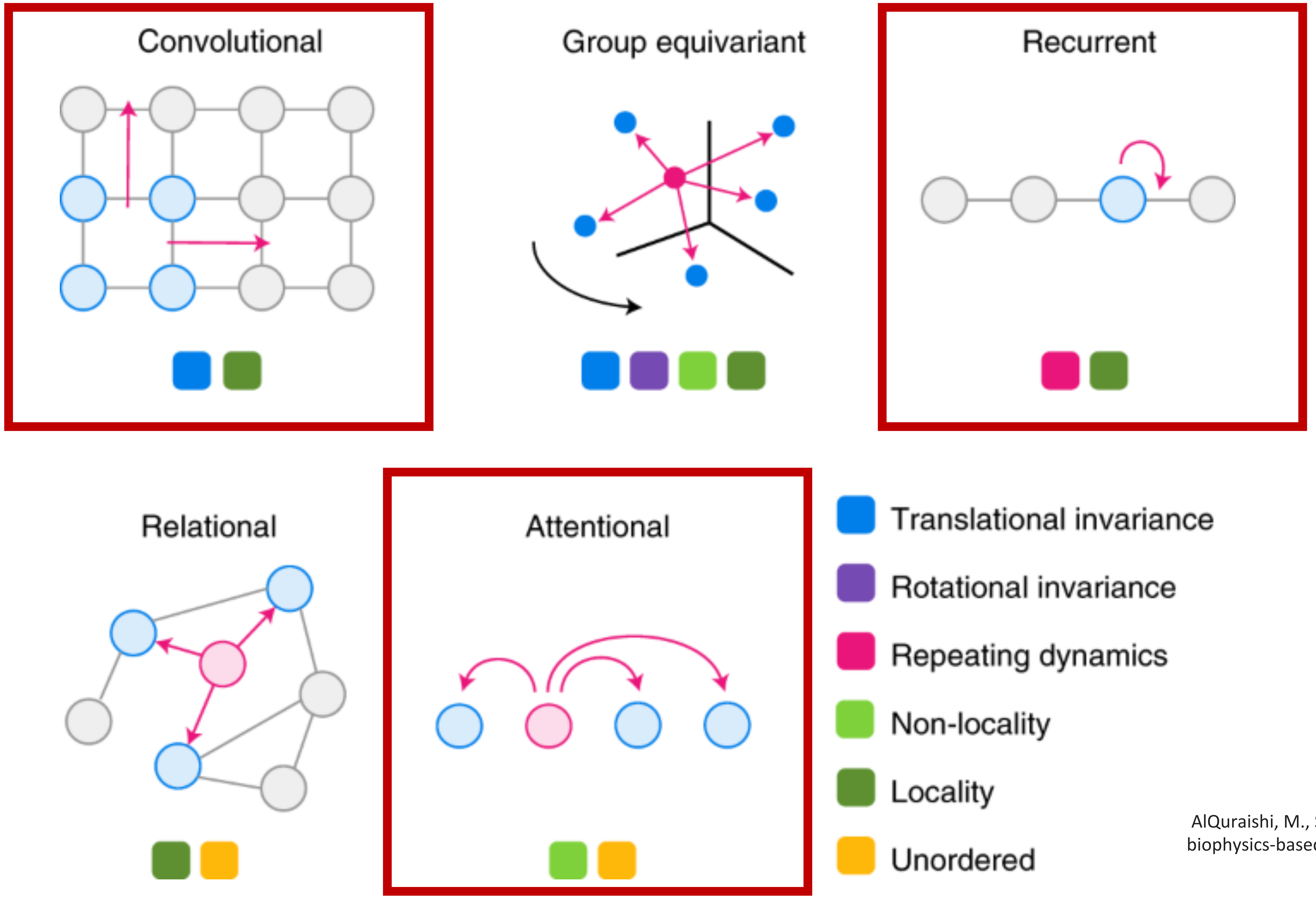
Find the inductive biases they instill in the network



AlQuraishi, M., Sorger, P.K. Differentiable biology: using deep learning for biophysics-based and data-driven modeling of molecular mechanisms. *Nat Methods* **18**, 1169–1180 (2021).

How to make sense of all these models?

Find the inductive biases they instill in the network

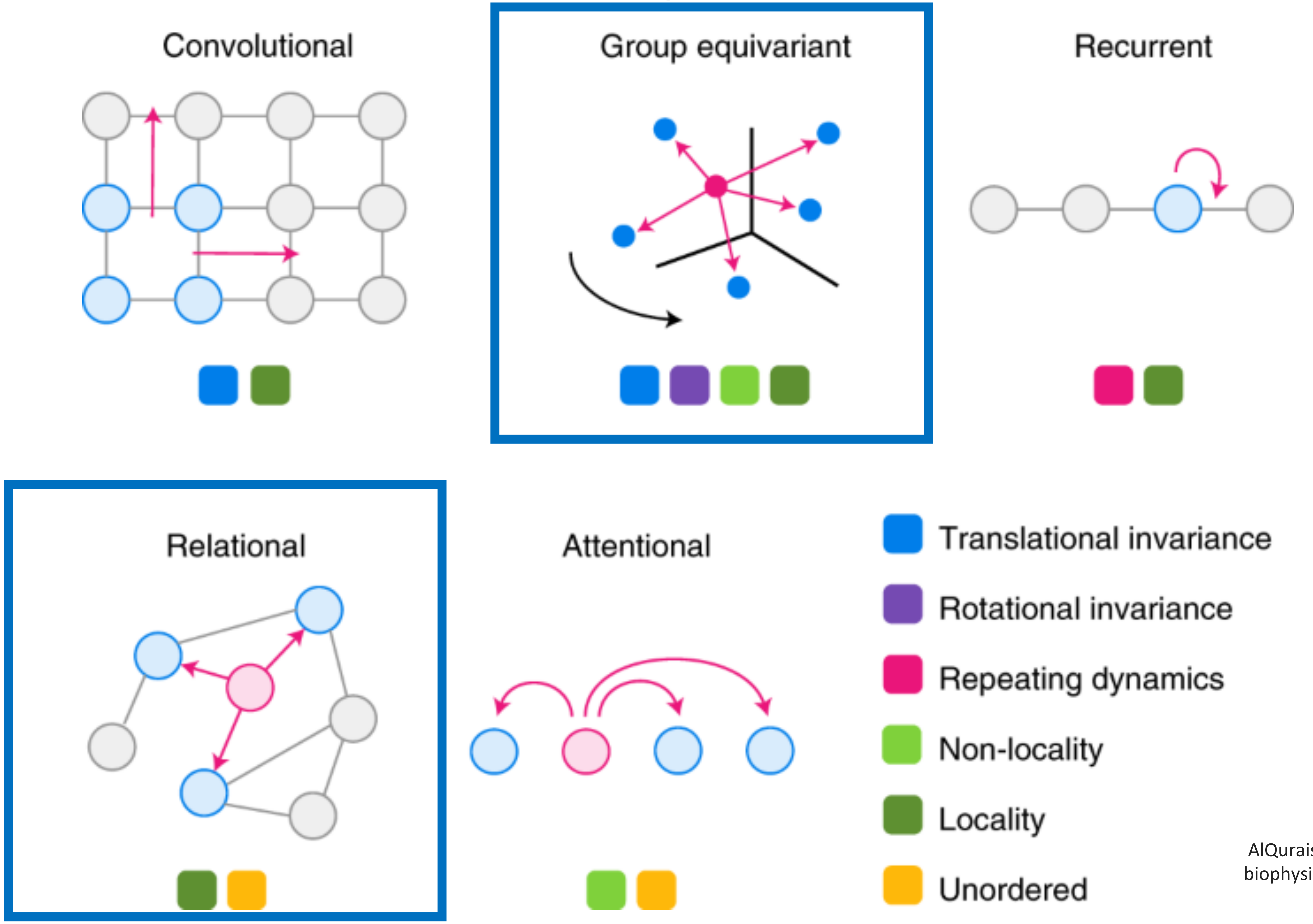


This week

AlQuraishi, M., Sorger, P.K. Differentiable biology: using deep learning for biophysics-based and data-driven modeling of molecular mechanisms. *Nat Methods* **18**, 1169–1180 (2021).

How to make sense of all these models?

Find the inductive biases they instill in the network



Next week

AlQuraishi, M., Sorger, P.K. Differentiable biology: using deep learning for biophysics-based and data-driven modeling of molecular mechanisms. *Nat Methods* **18**, 1169–1180 (2021).

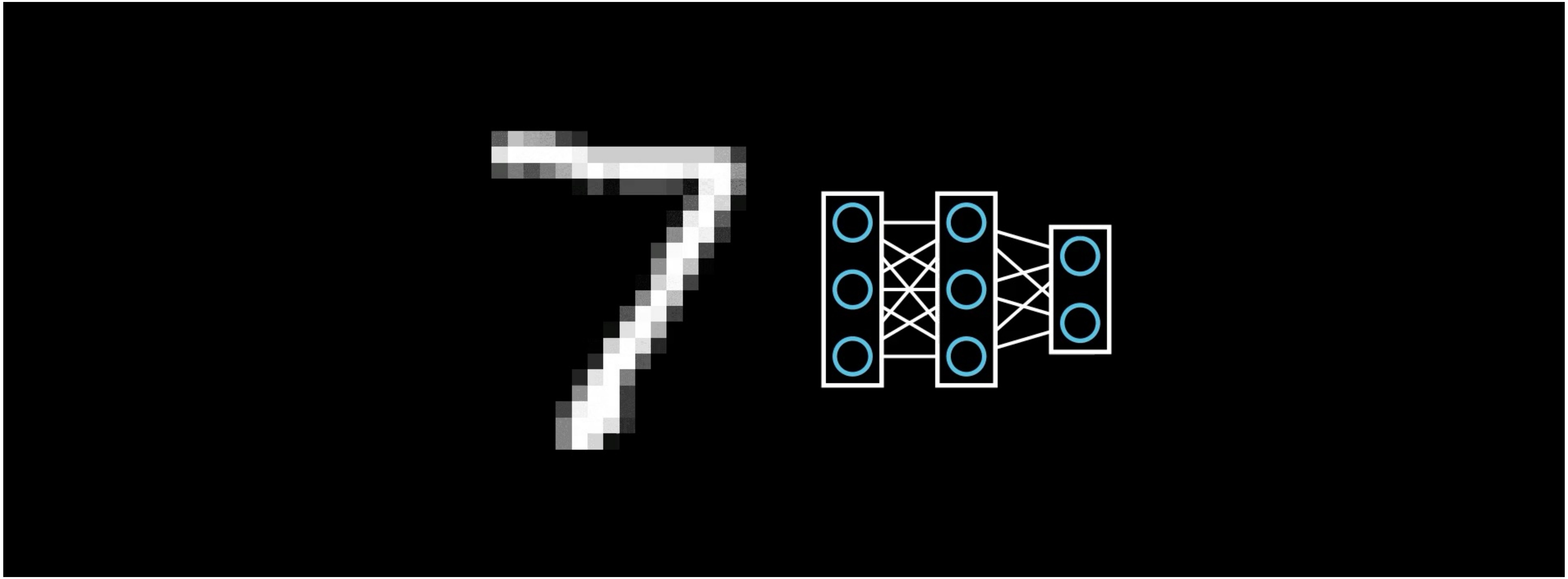
Overview

1. Images: Convolutional Neural Networks
2. Sequences: RNNs
3. Transformers
4. Current developments

1. Convolutional Neural Networks

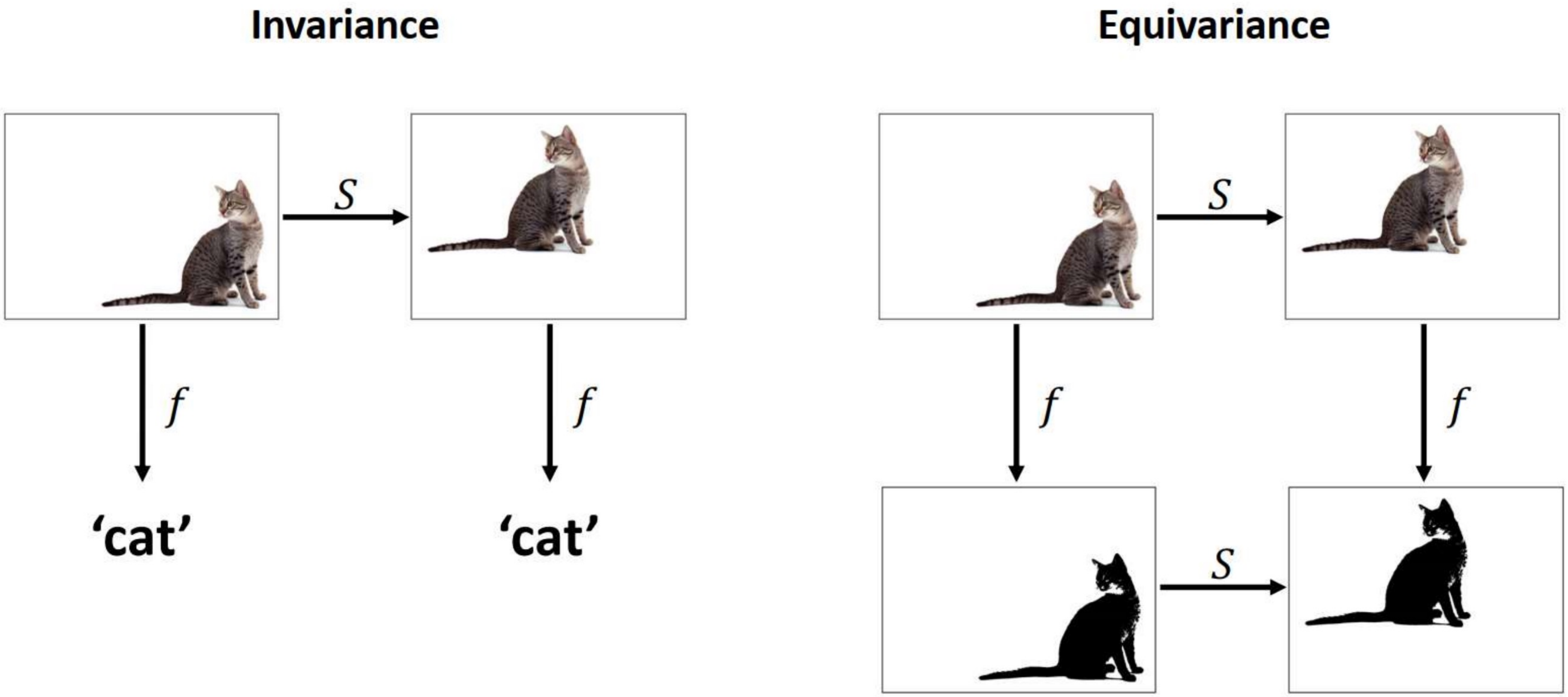
How to deal with images

Naive approach: unroll them and passt them into an MLP



Inductive Bias: Translational In-/Equivariance

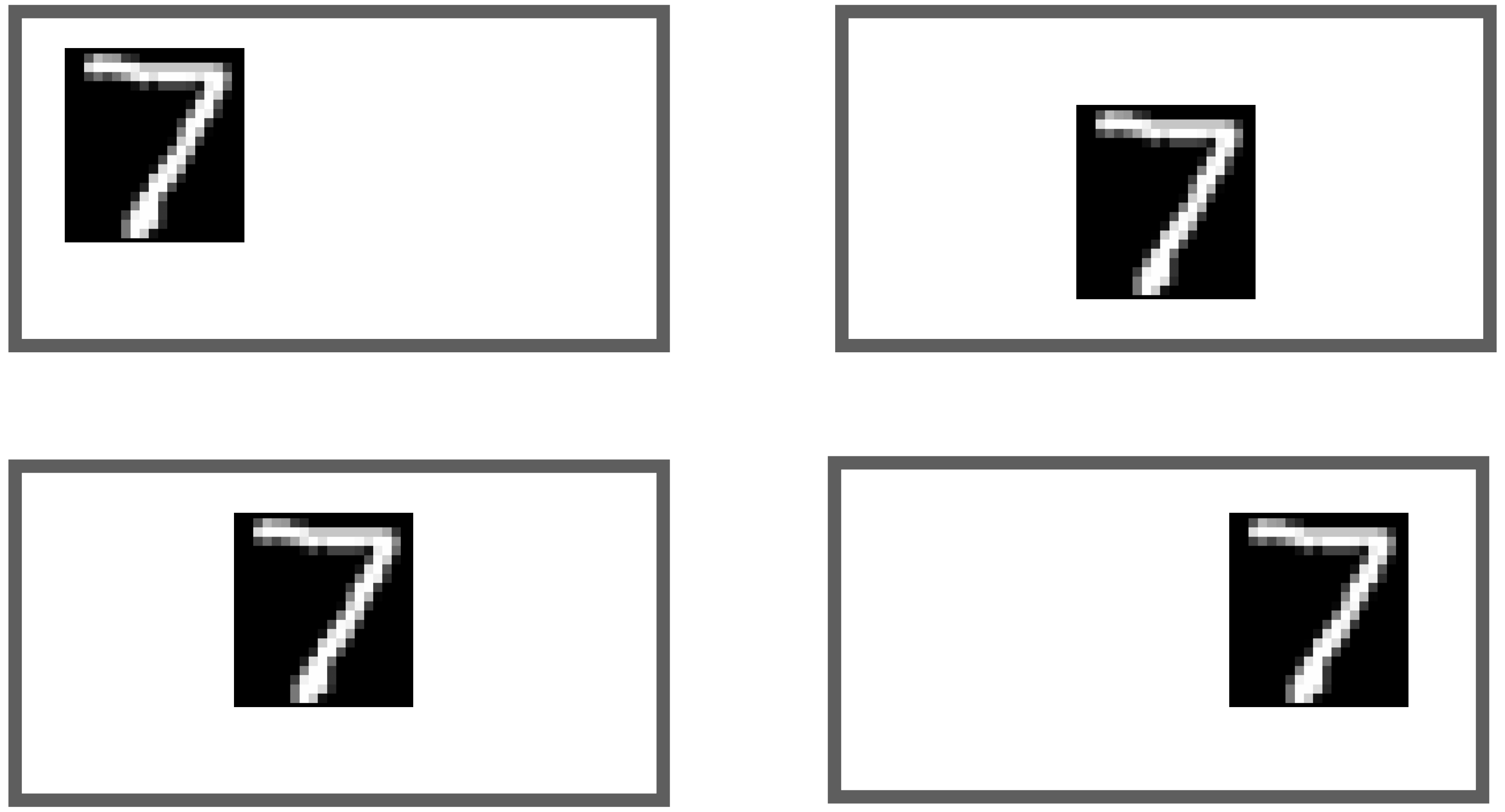
Leverage the symmetry of your data



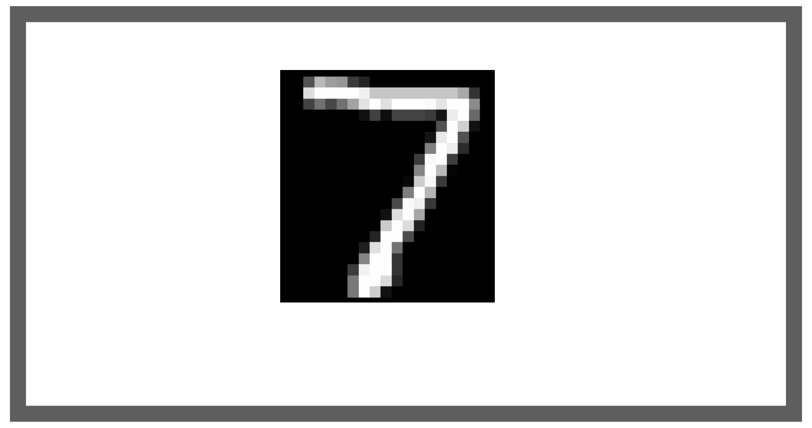
Why leverage symmetries?

We need more data = our network is more efficient!

Training without translational symmetry

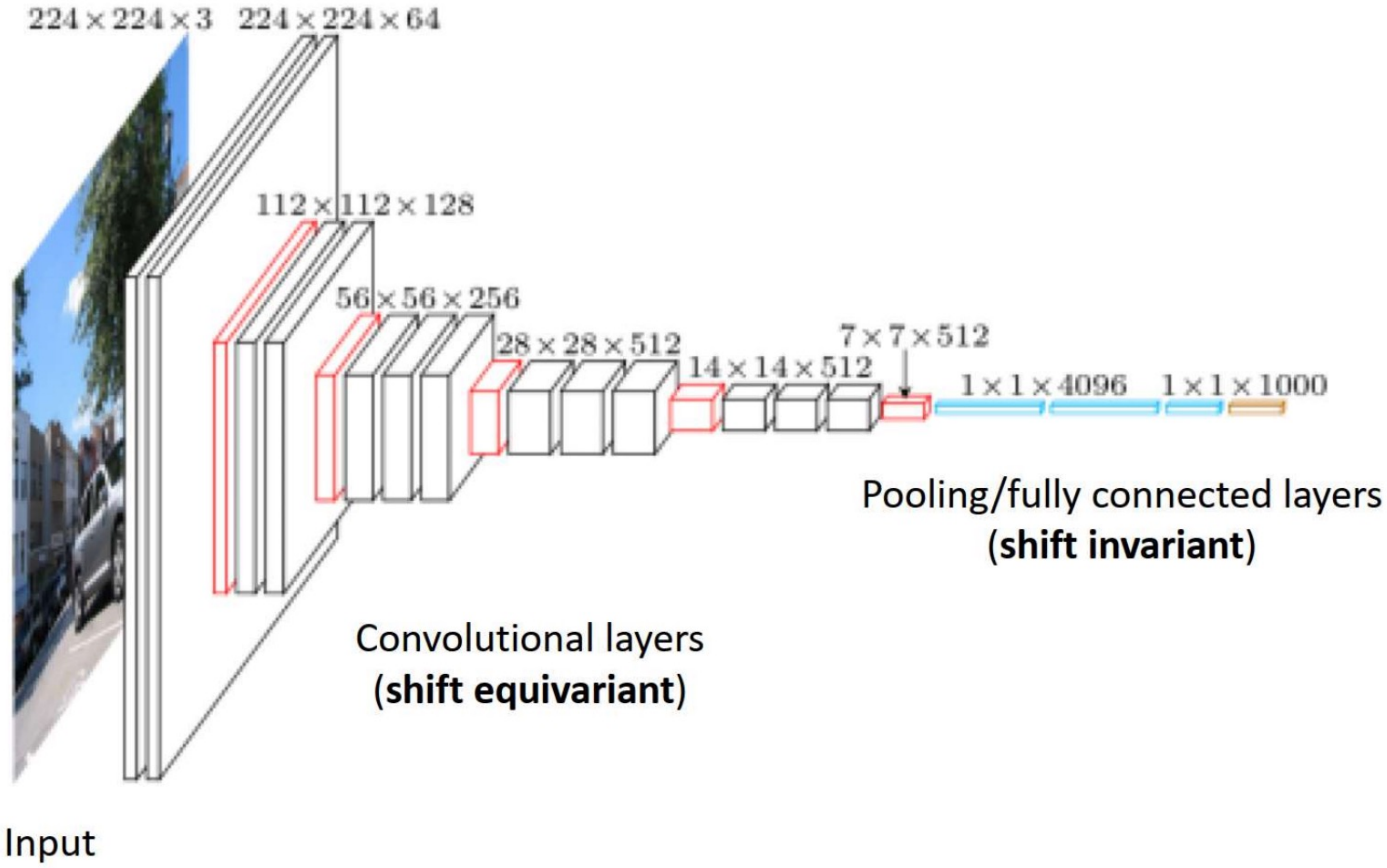


Training with translational symmetry



How do we do this in practice?

Implement neural network layers that respect these symmetries

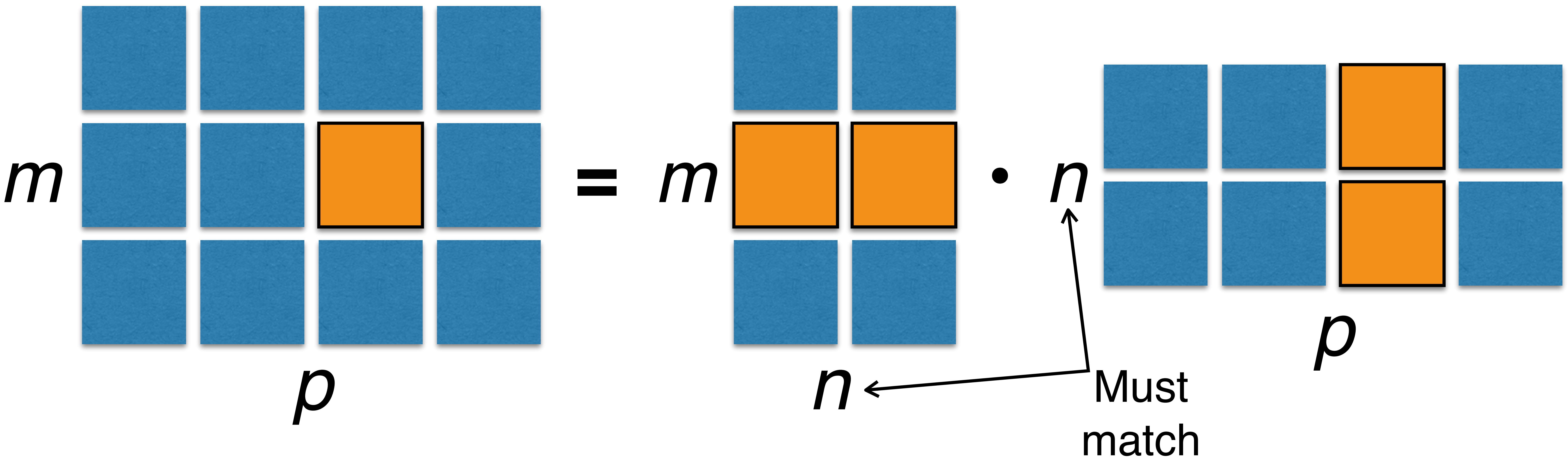


Convolutional Layers

Reminder: Matrix multiplication

$$\boldsymbol{C} = \boldsymbol{A}\boldsymbol{B}. \tag{2.4}$$

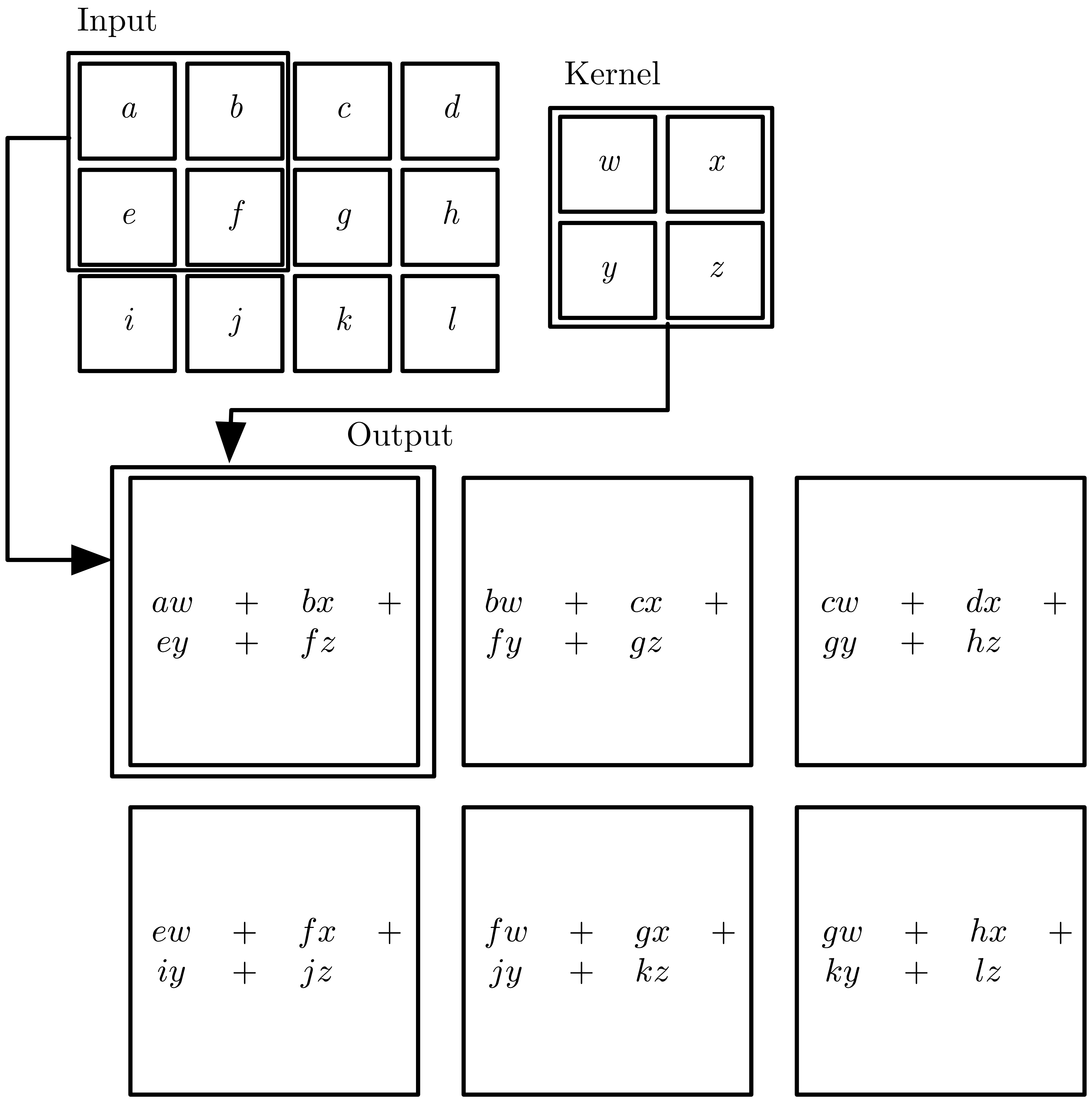
$$C_{i,j} = \sum_k A_{i,k} B_{k,j}. \tag{2.5}$$



(Goodfellow 2016)

Convolutional Layers

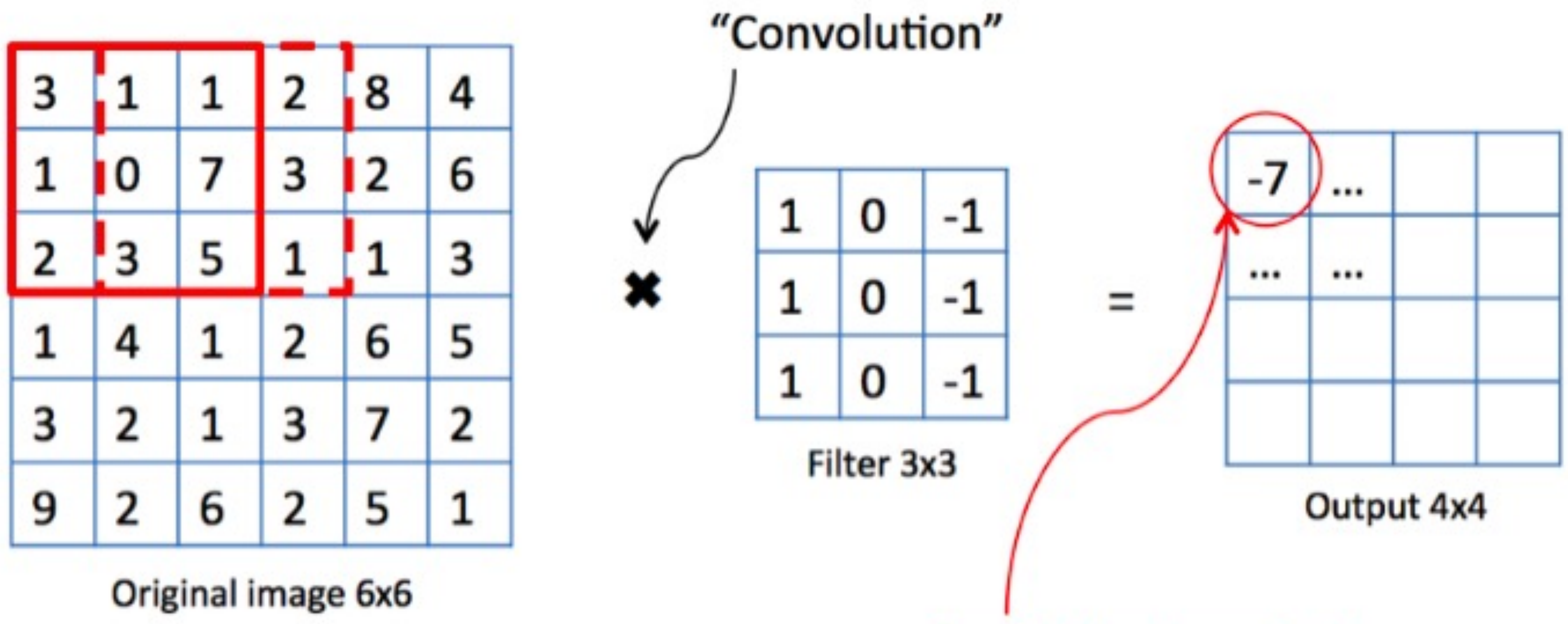
The weights are in the kernel



Goodfellow 2016

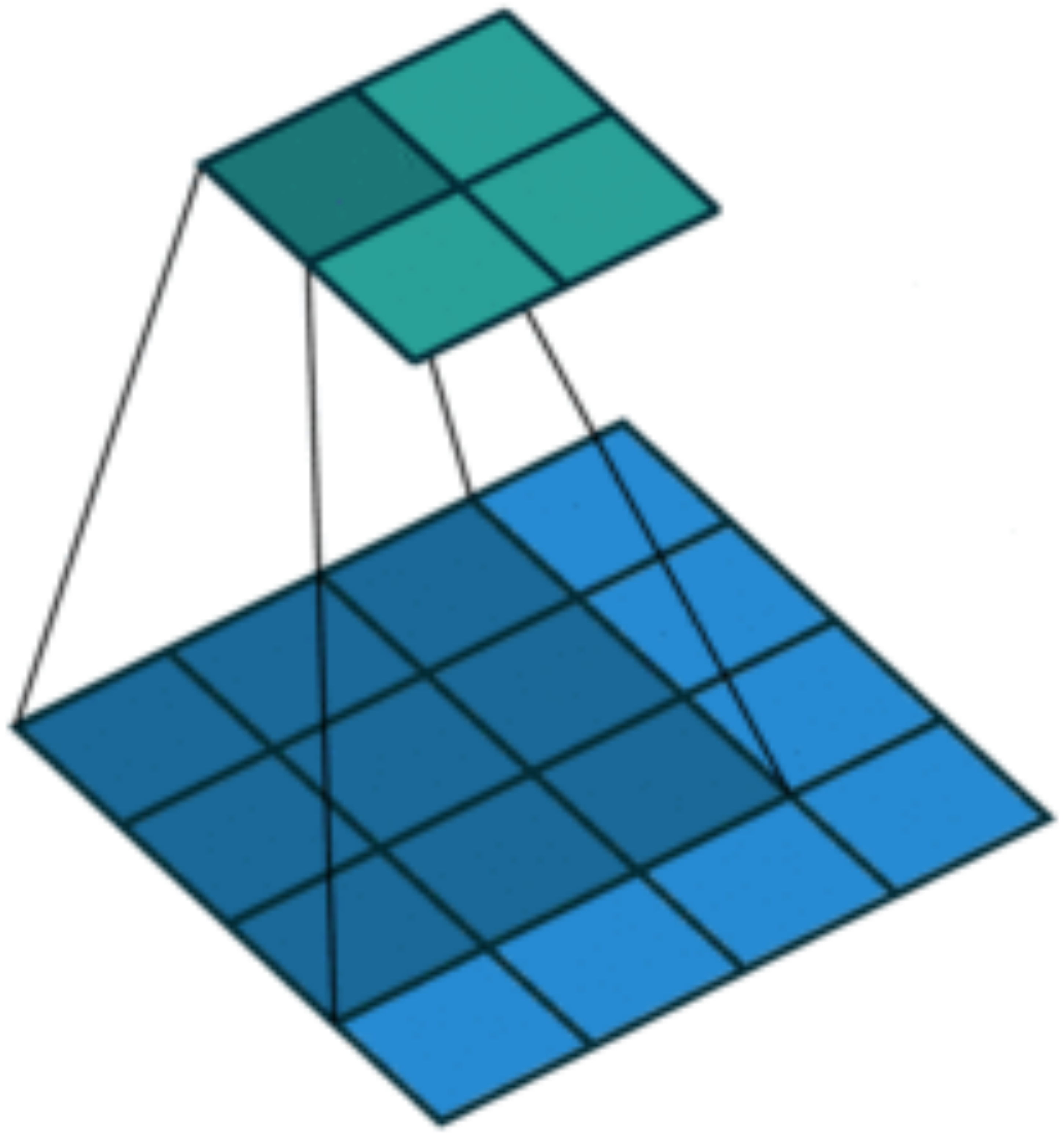
Convolutional Layers

Convolution = Repeated Matrix Multiplication



How can I imagine that?

Sliding the kernel over the image



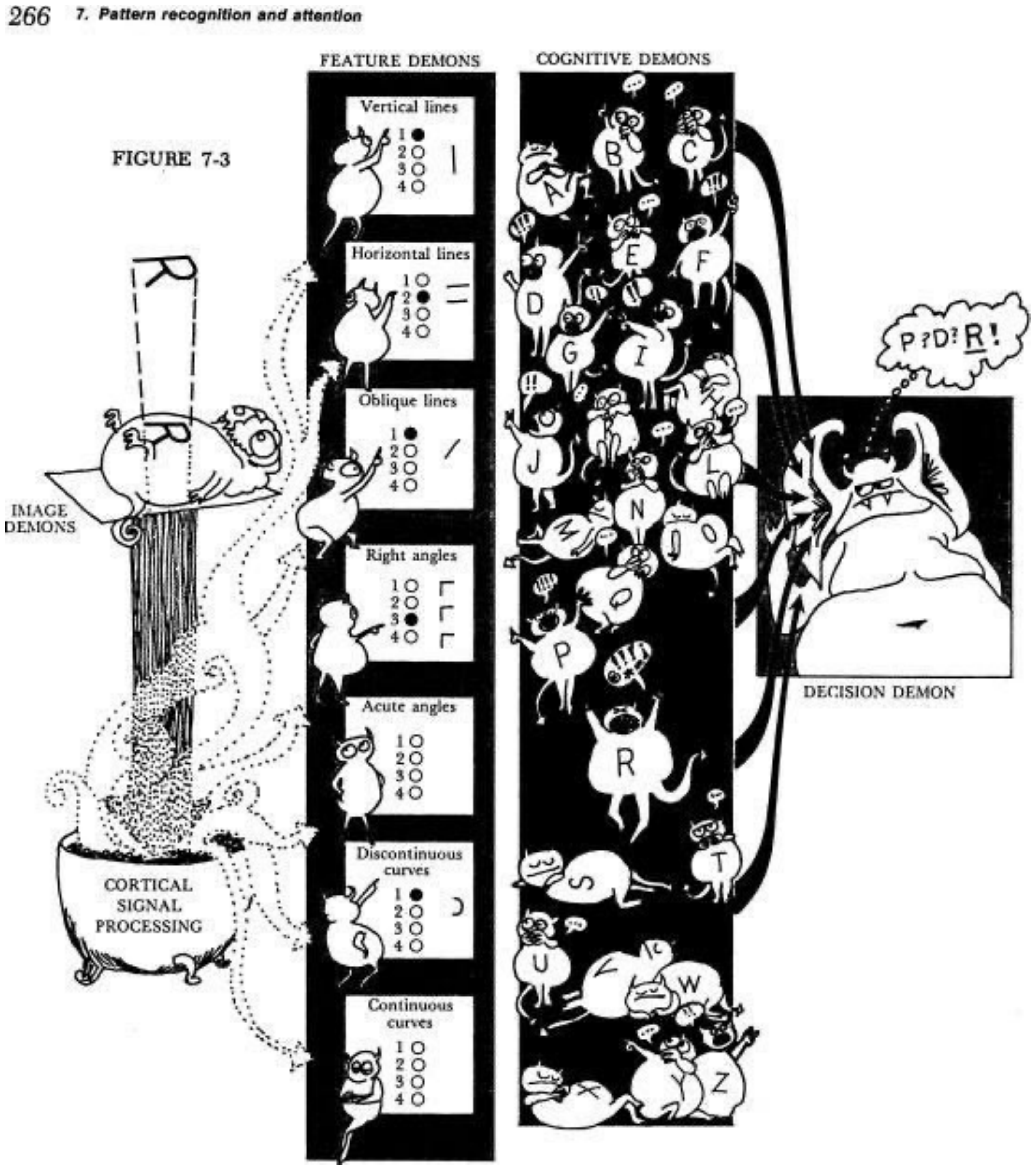
Vincent Dumoulin, Francesco Visin - A guide to convolution arithmetic for deep learning

How can I imagine that?

Multiple Kernels allow detecting multiple features

Pattern Recognition all over again

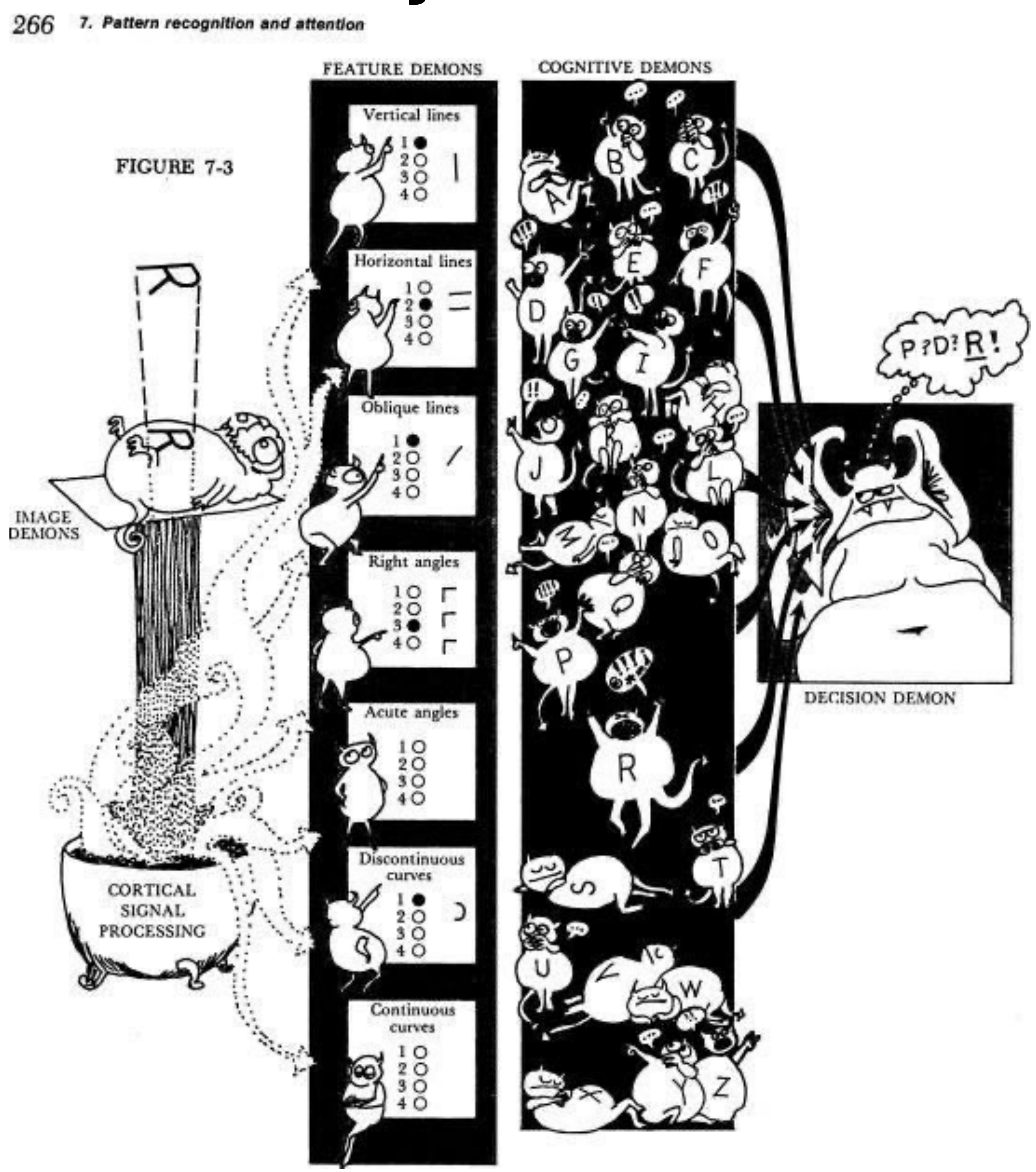
This time adjusted to the image case

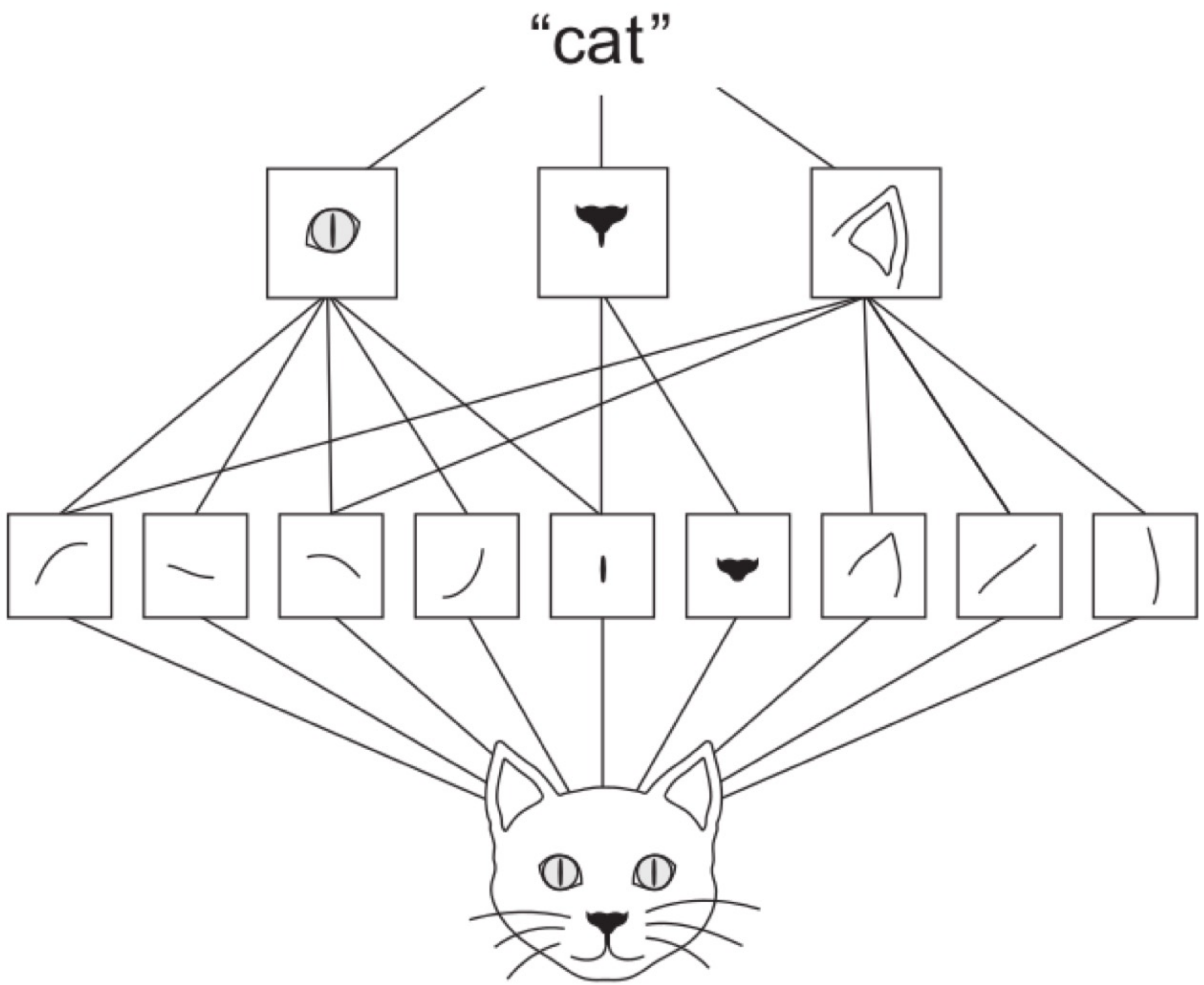


[F. Chollet, Deep Learning with Python]

Pattern Recognition all over again

This time adjusted to the image case





F. Chollet, Deep Learning with Python

Pattern Recognition all over again

Look at it yourself!

Google Brain: Feature Visualisation

OpenAI: Microscope

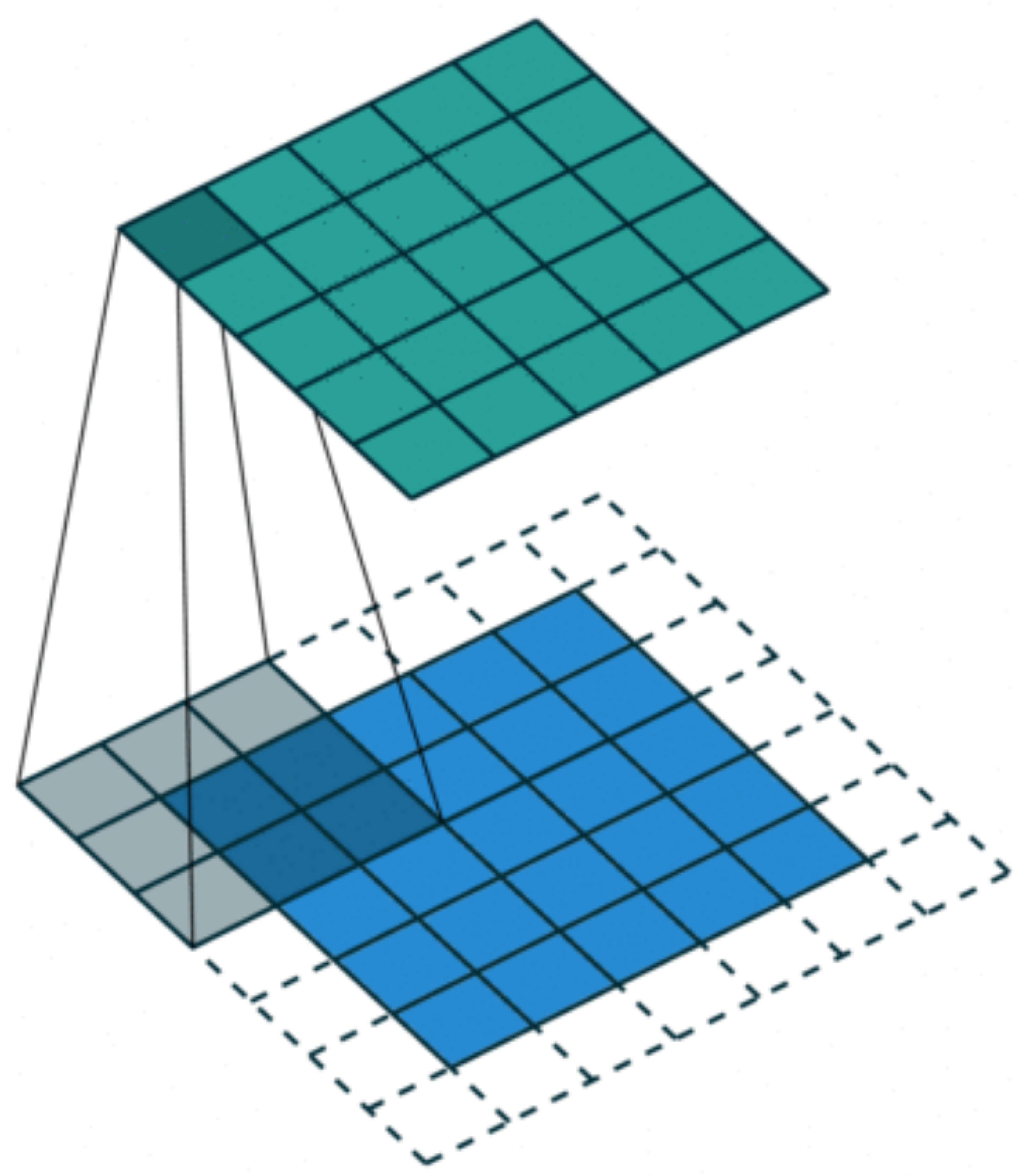
Dataset Examples show us what neurons respond to in practice

Optimization isolates the causes of behavior from mere correlations. A neuron may not be detecting what you initially thought.

Baseball—or stripes?
mixed4a, Unit 6

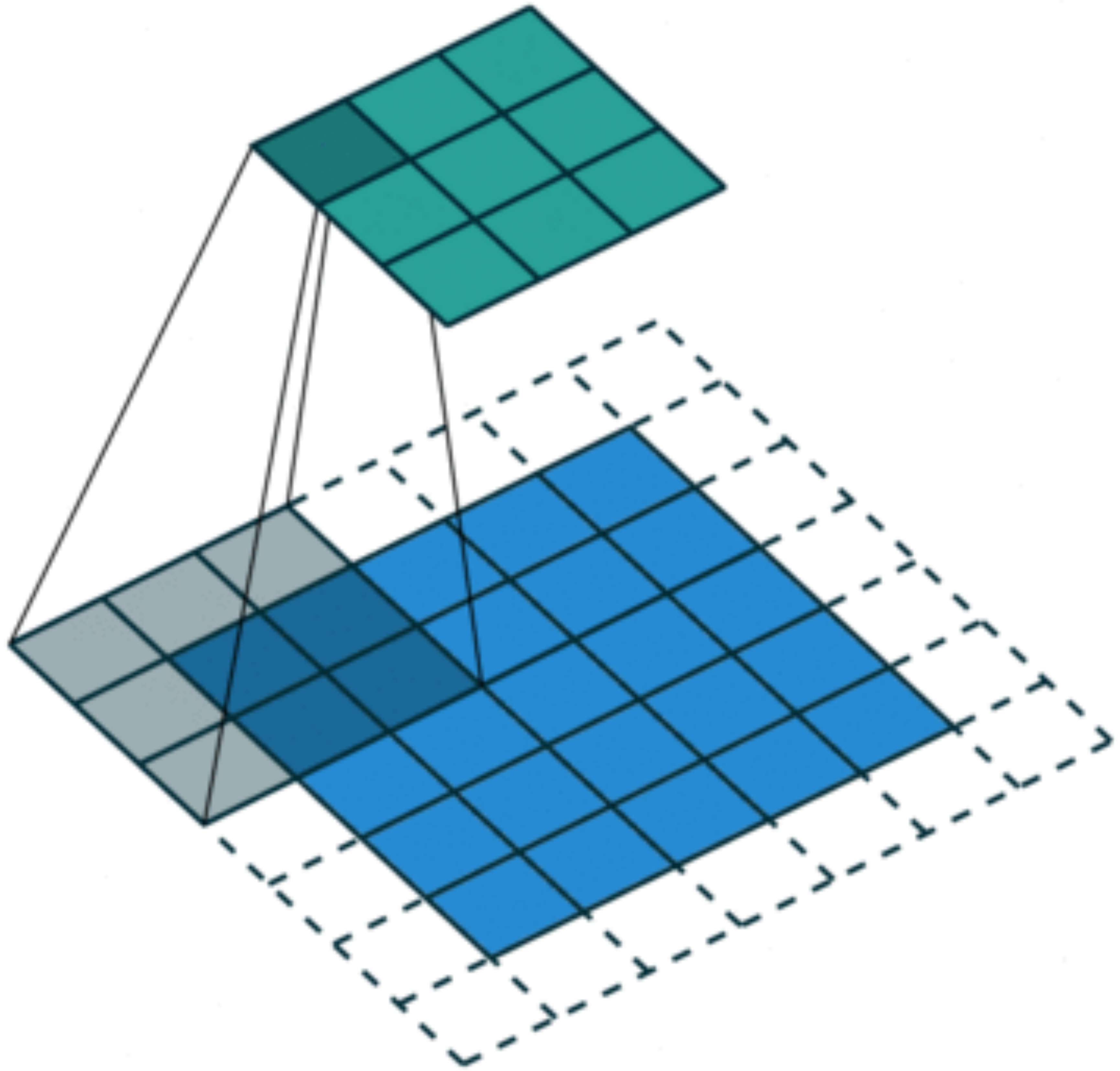
Avoid reducing size with padding

Different ways to pad (zero-pad, mean-pad, ...)



Vincent Dumoulin, Francesco Visin - A guide to convolution arithmetic for deep learning

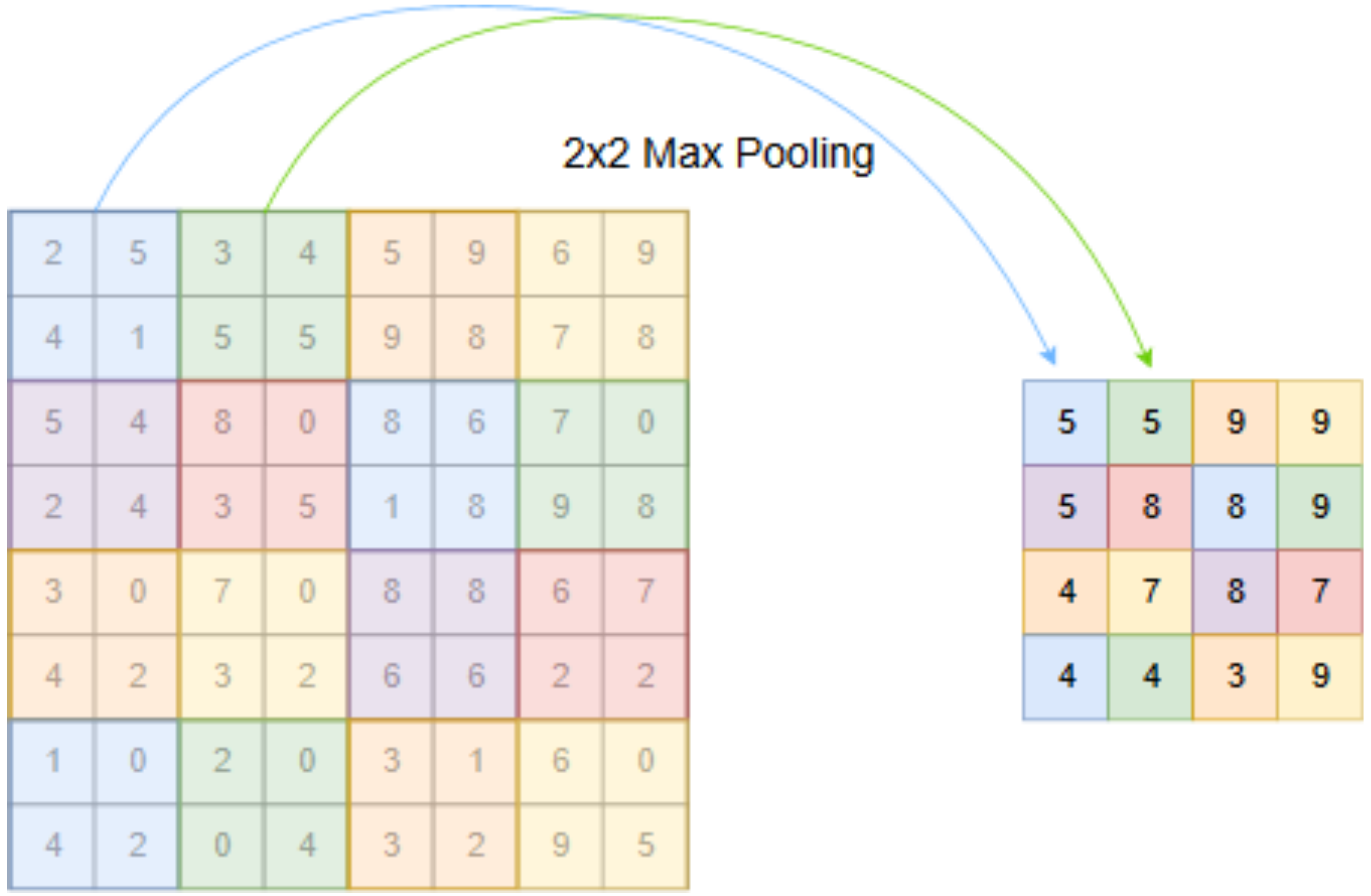
Make bigger jumps with strides



Vincent Dumoulin, Francesco Visin - A guide to convolution arithmetic for deep learning

Pooling: Shift-invariant operation

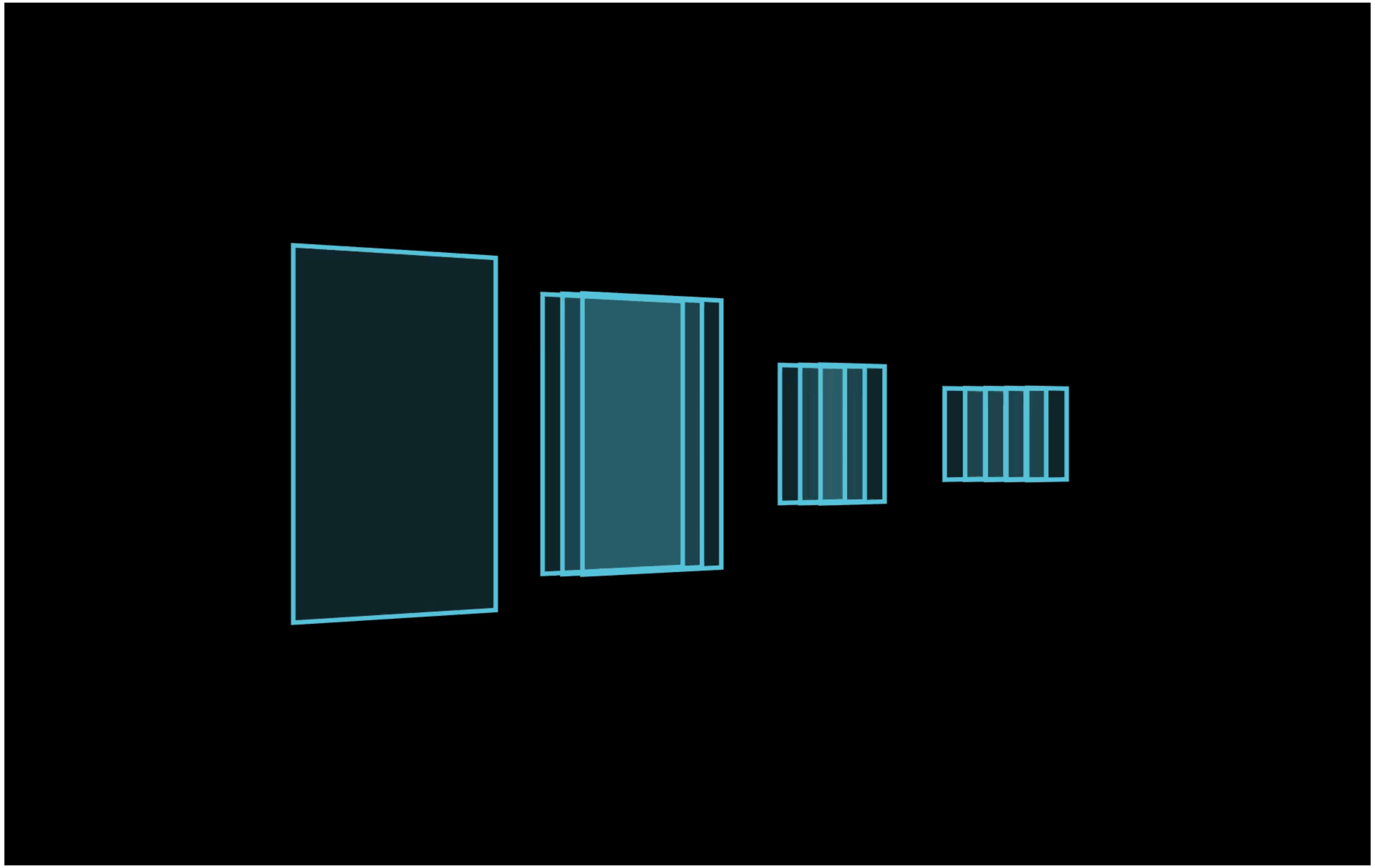
Reduce size, but no learning involved



F. Chollet, Deep Learning with Python

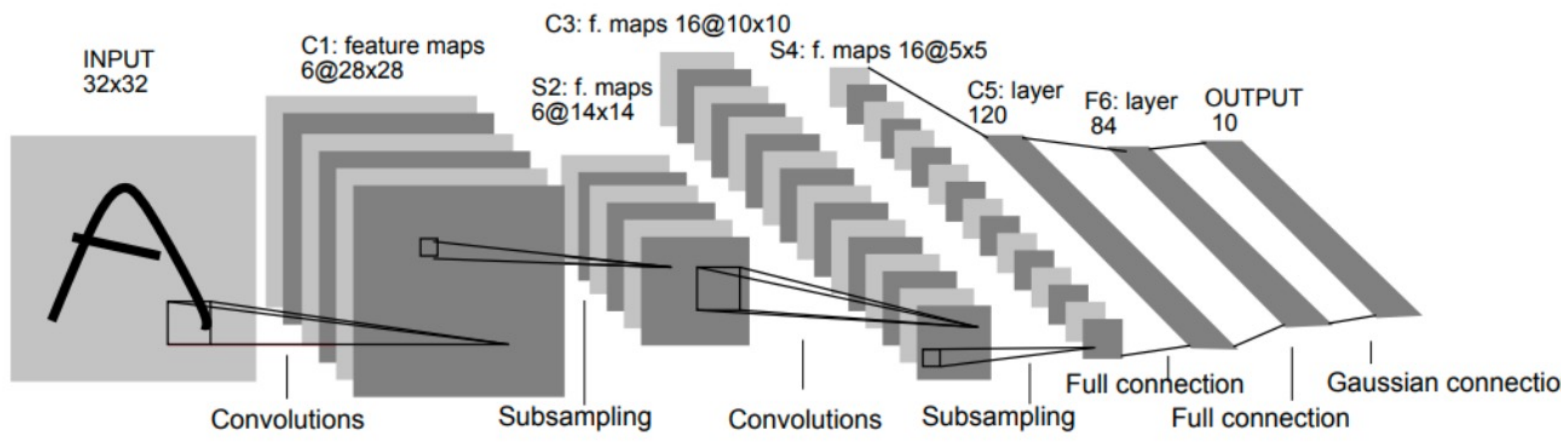
Putting things together: A full CNN

Conv. Layers -> Pooling -> FC Layers



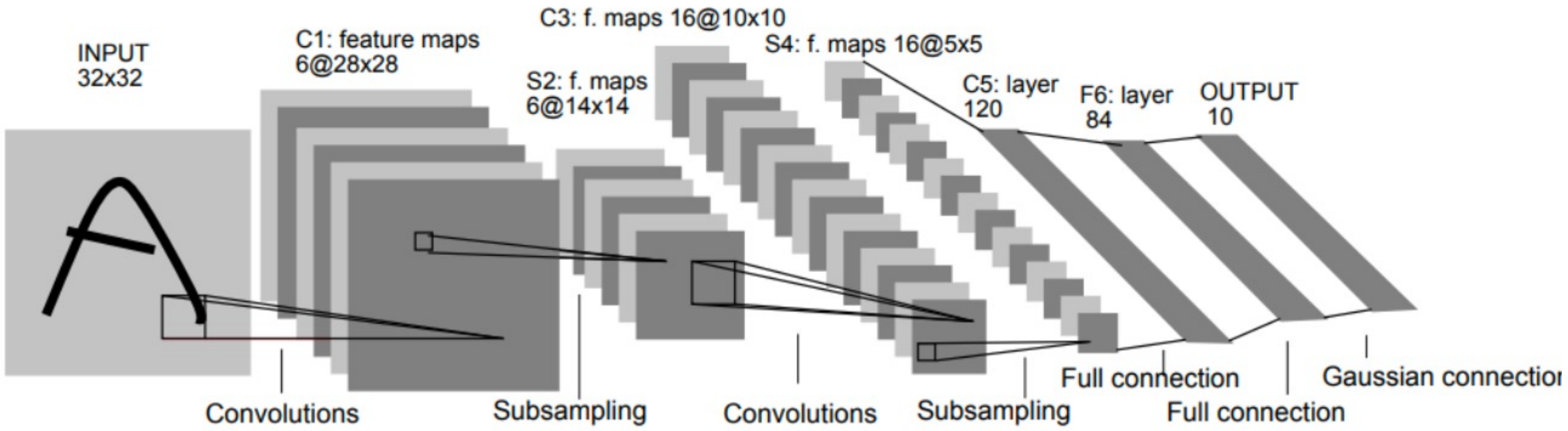
LeNet (1998): CNNs become a thing

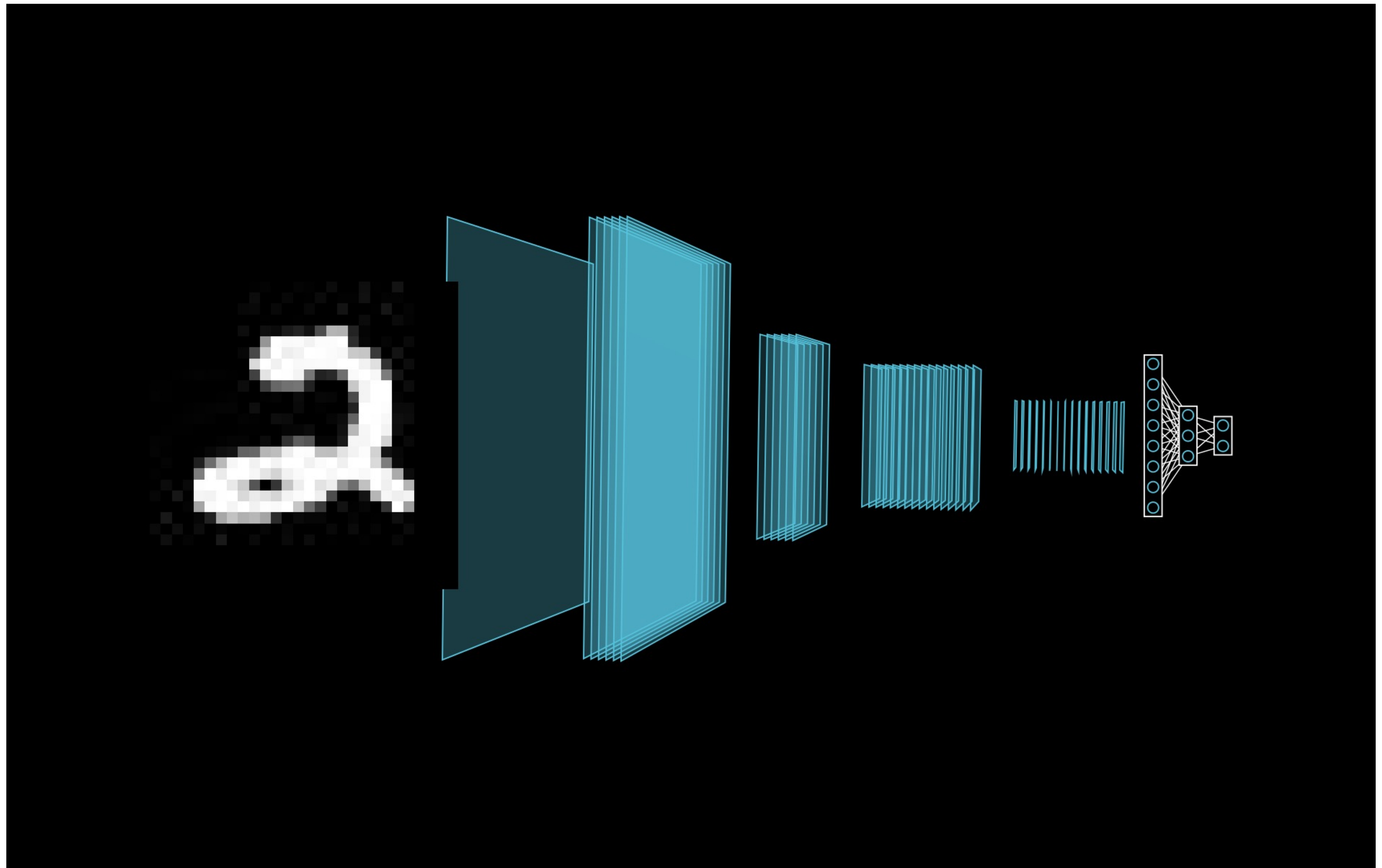
Exactly what we discussed, just bigger



LeNet (1998): CNNs become a thing

Exactly what we discussed, just bigger





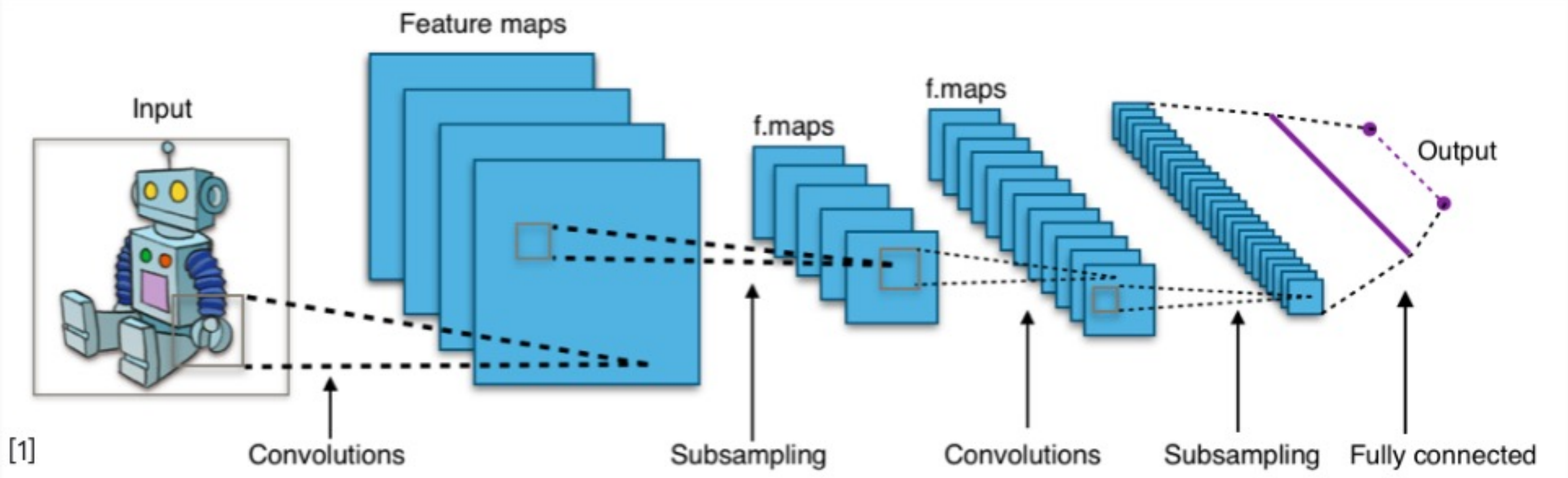
2. RNNs

The classic landscape

One architecture per community

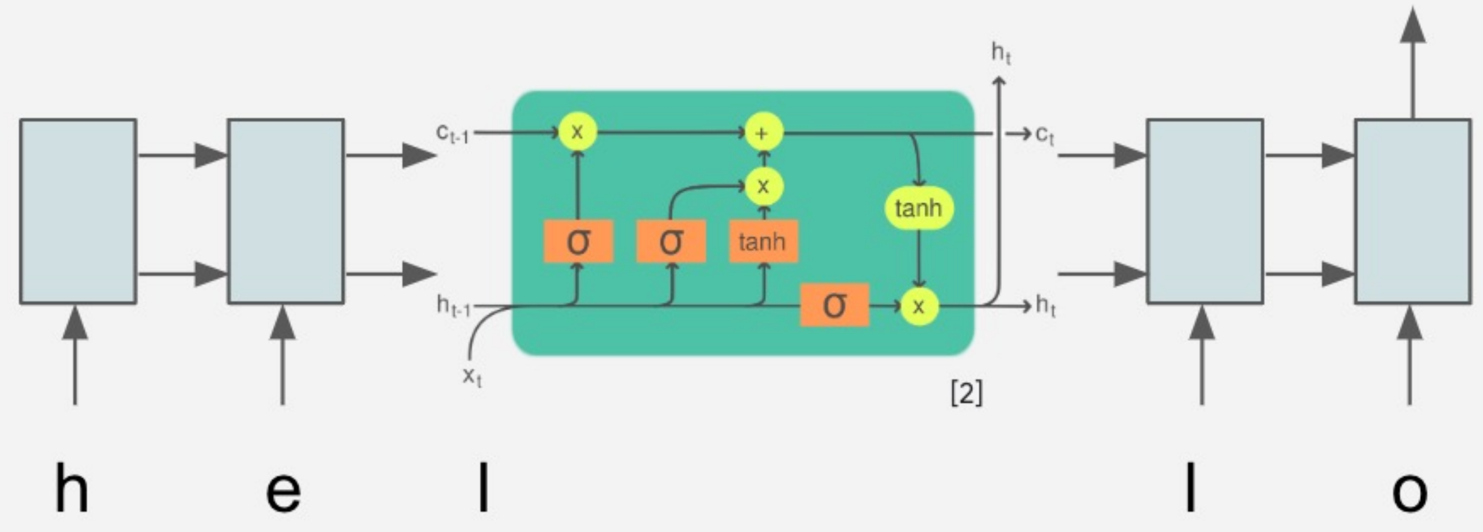
Computer Vision

Convolutional NNs (+ResNets)



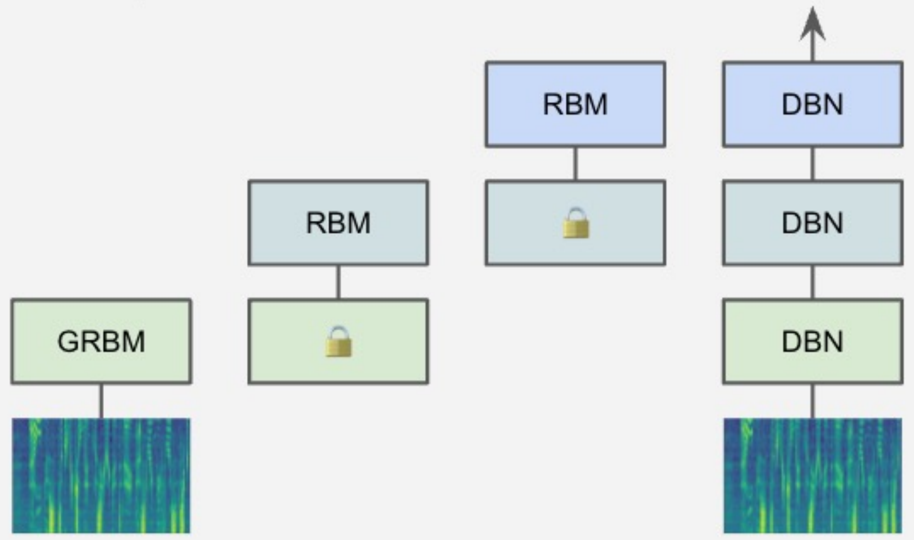
Natural Lang. Proc.

Recurrent NNs (+LSTMs)



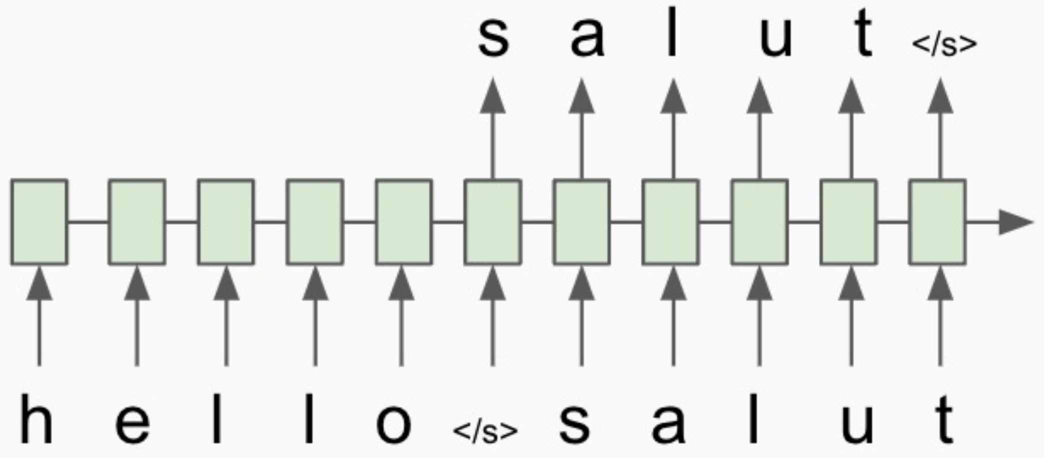
Speech

Deep Belief Nets (+non-DL)



Translation

Seq2Seq



RL

BC/GAIL

Algorithm 1 Generative adversarial imitation learning

1: **Input:** Expert trajectories $\tau_E \sim \pi_E$, initial policy and discriminator parameters θ_0, w_0
2: **for** $i = 0, 1, 2, \dots$ **do**
3: Sample trajectories $\tau_i \sim \pi_{\theta_i}$
4: Update the discriminator parameters from w_i to w_{i+1} with the gradient

$$\hat{\mathbb{E}}_{\tau_i}[\nabla_w \log(D_w(s,a))] + \hat{\mathbb{E}}_{\tau_E}[\nabla_w \log(1 - D_w(s,a))] \quad (17)$$

5: Take a policy step from θ_i to θ_{i+1}, using the TRPO rule with cost function $\log(D_{w_{i+1}}(s,a))$. Specifically, take a KL-constrained natural gradient step with

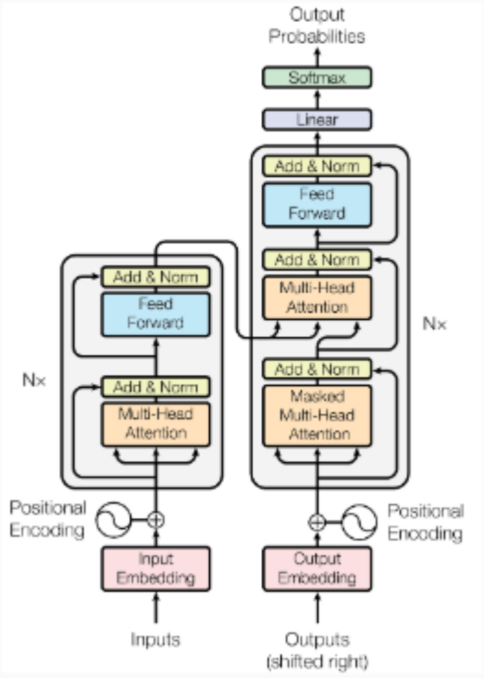
$$\hat{\mathbb{E}}_{\tau_i}\left[\nabla_\theta \log \pi_\theta(a|s) Q(s,a)\right] - \lambda \nabla_\theta H(\pi_\theta), \quad \text{where } Q(\bar{s},\bar{a}) = \hat{\mathbb{E}}_{\tau_i}[\log(D_{w_{i+1}}(s,a)) \,|\, s_0 = \bar{s}, a_0 = \bar{a}] \quad (18)$$

6: **end for**

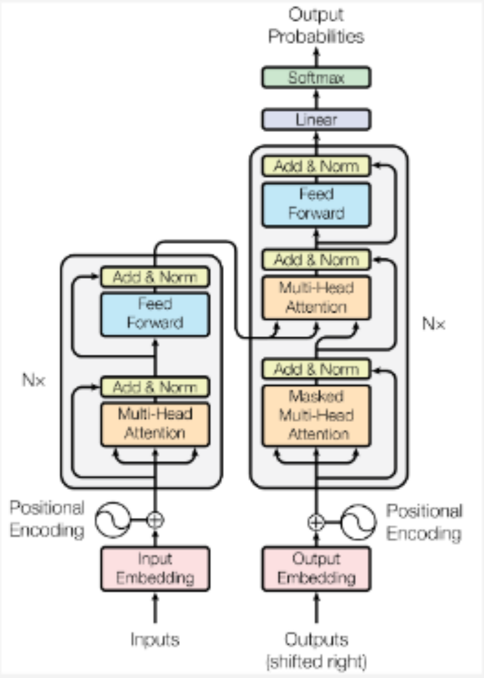
The transformer’s takeover

One community at a time

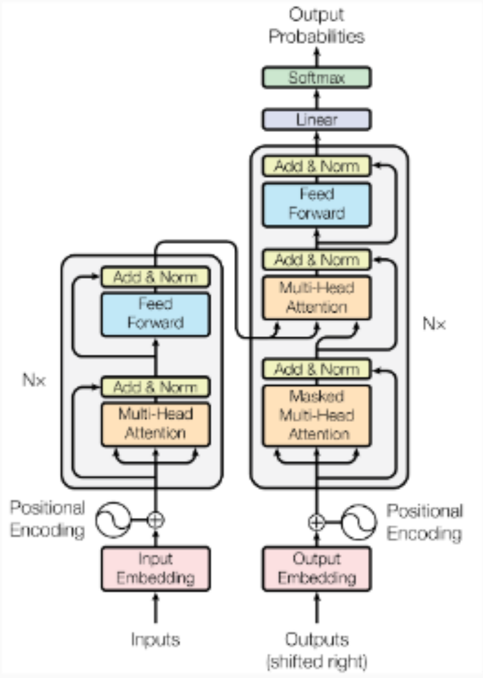
Computer Vision



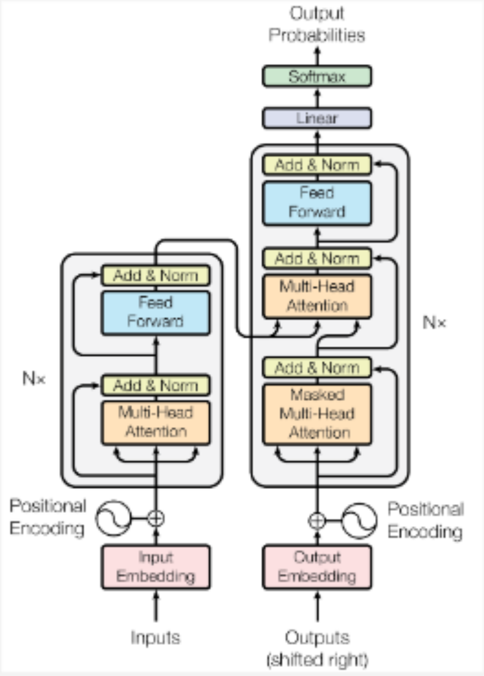
Natural Lang. Proc.



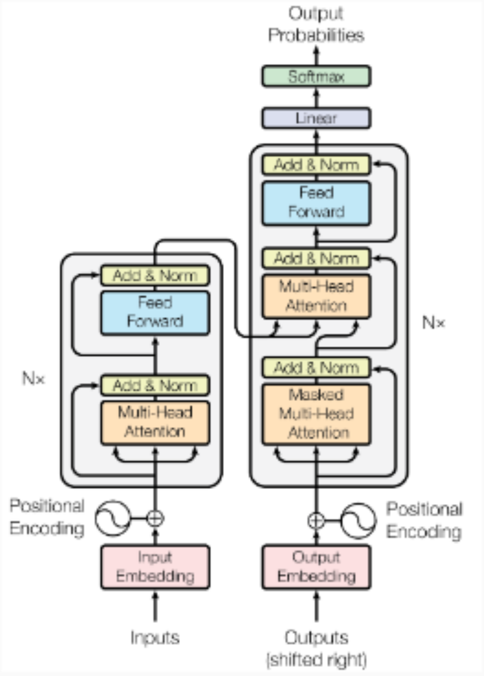
Reinf. Learning



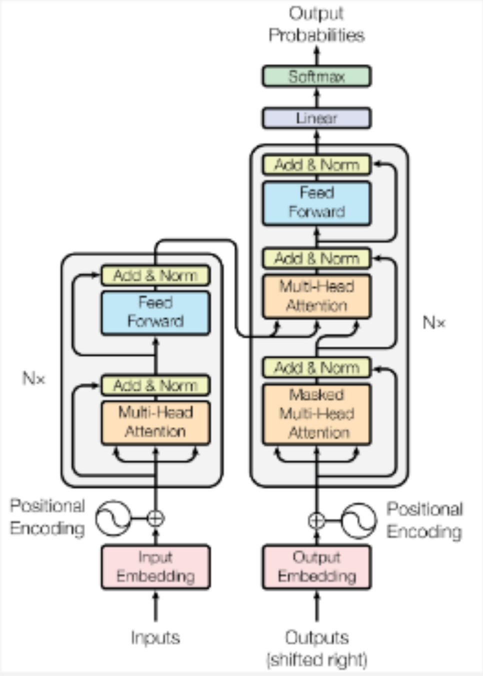
Speech



Translation



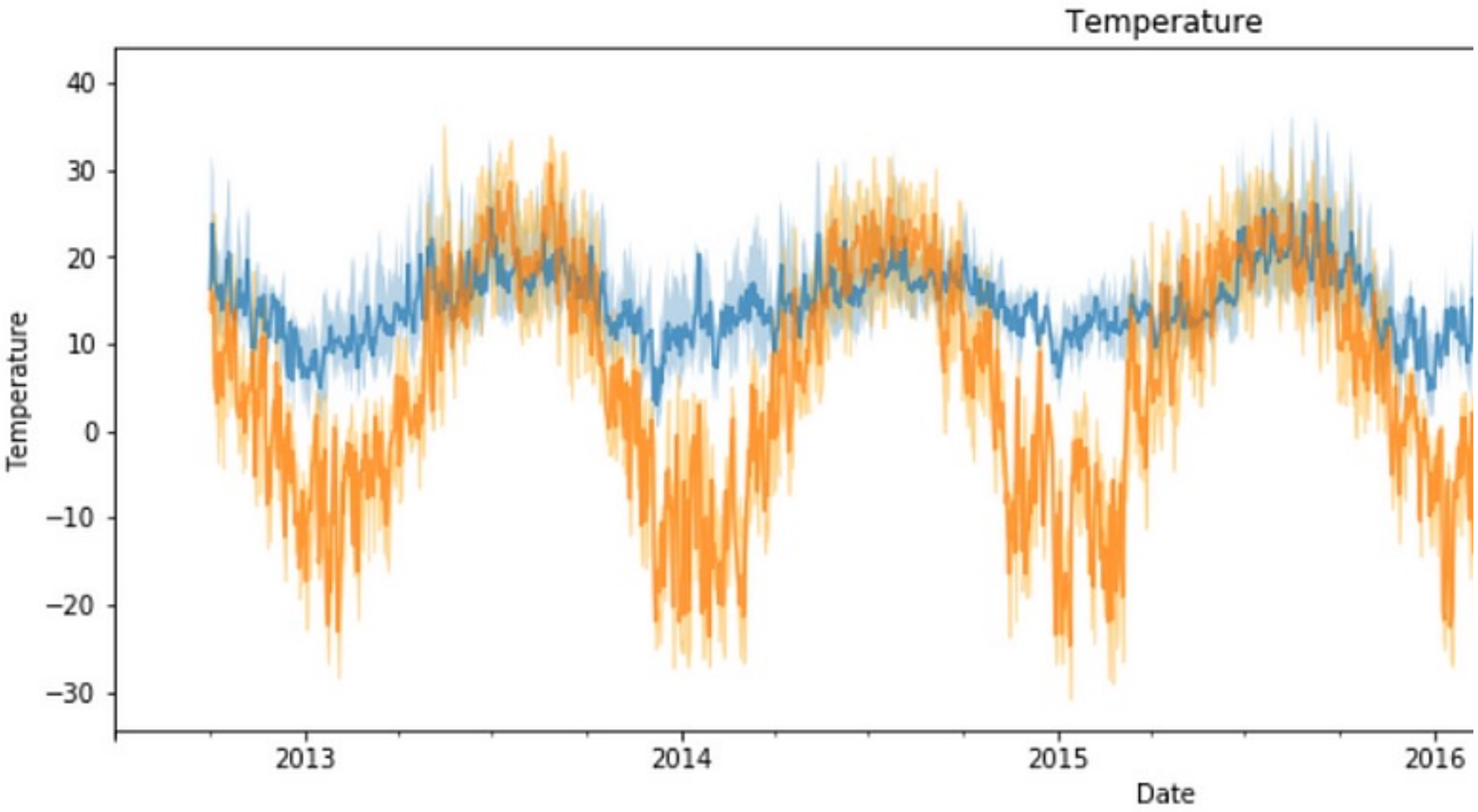
Graphs/Science



ransformer image source: "Attention Is All You Need" paper

How to deal with sequential data?

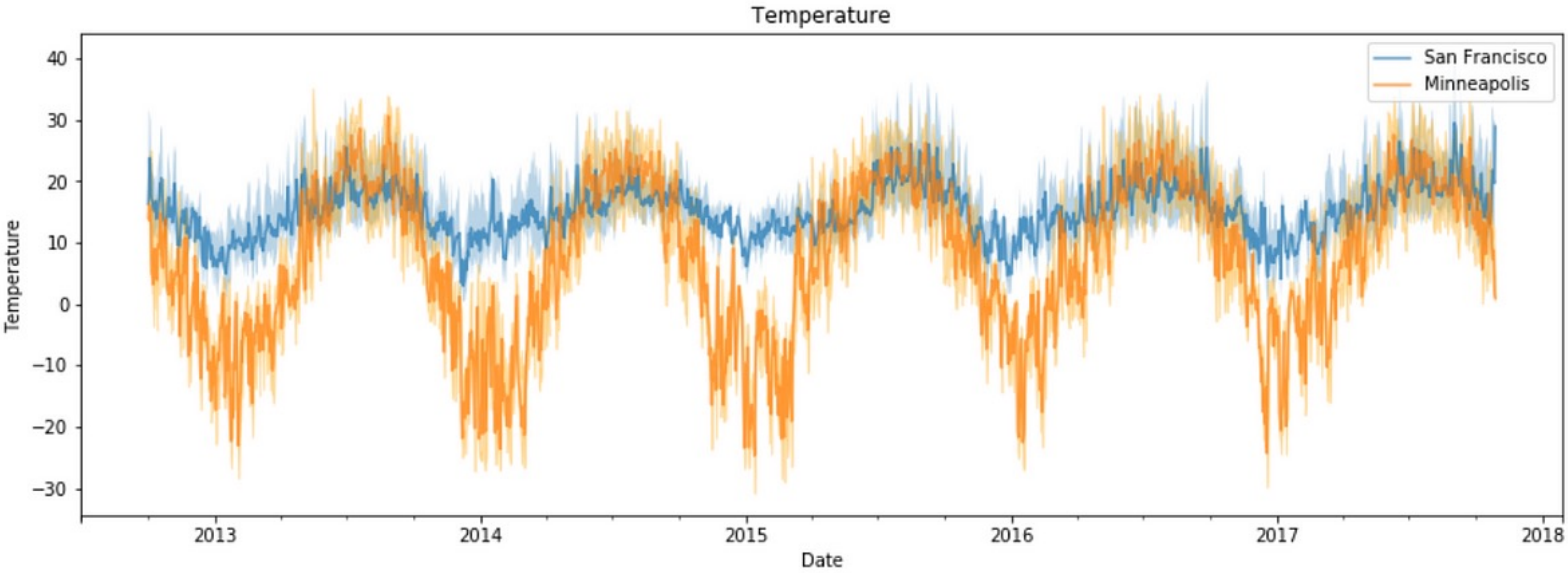
You can only look into the past, not into the future



Towardsdatascience.com

How to deal with sequential data?

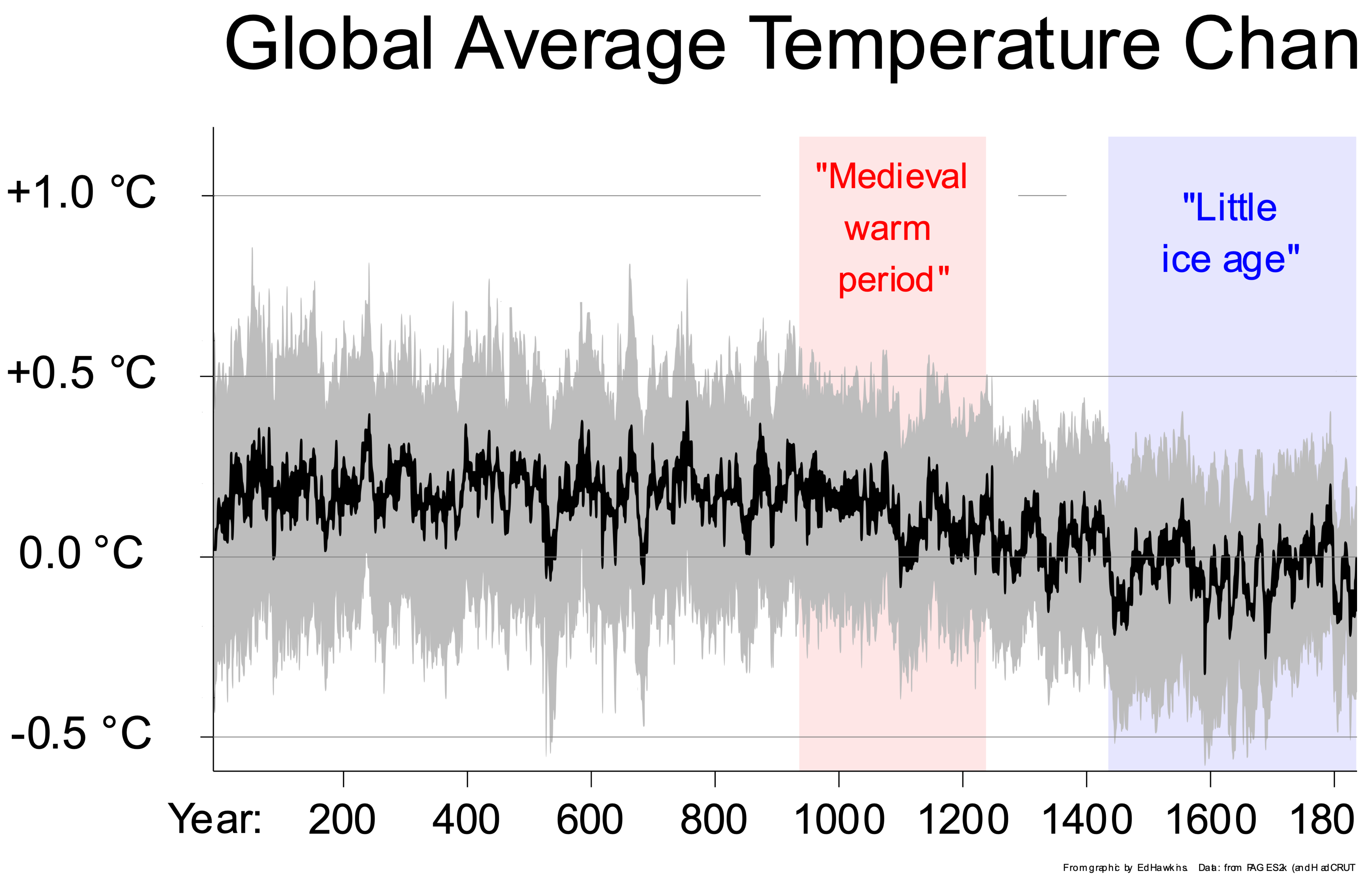
You can only look into the past, not into the future



Towardsdatascience.com

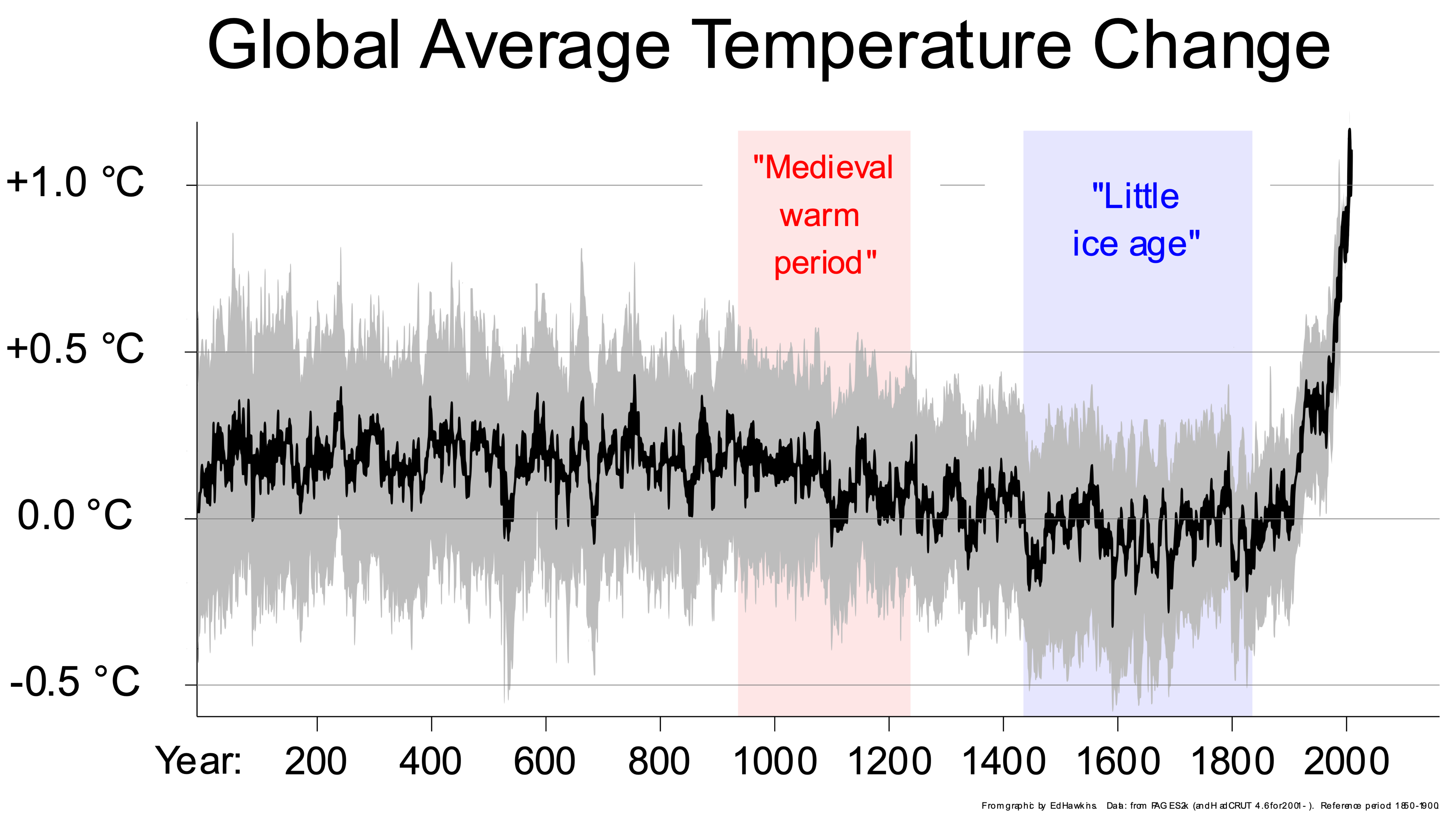
How to deal with sequential data?

You can only look into the past, not into the future



How to deal with sequential data?

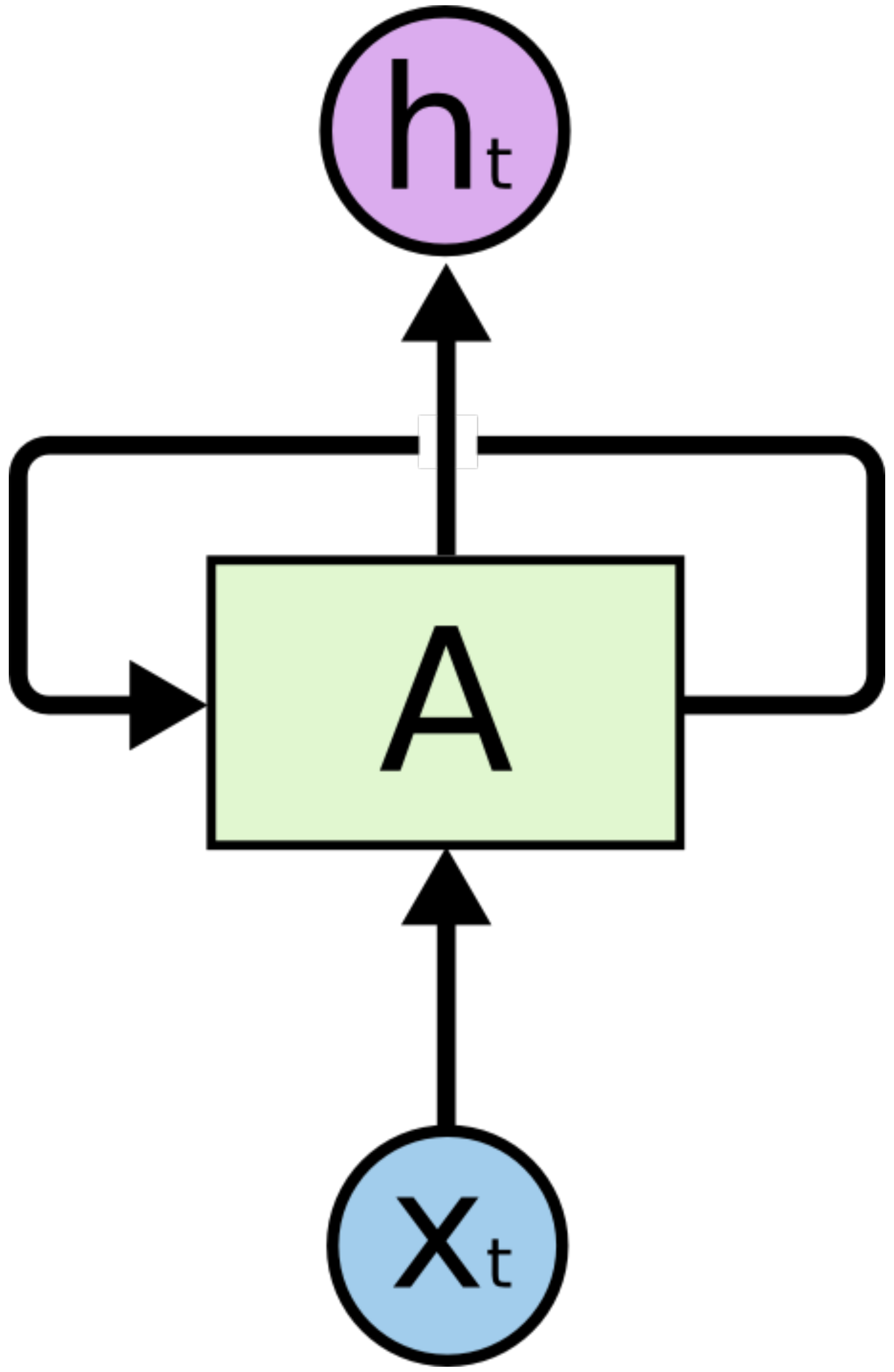
You can only look into the past, not into the future



Wikipedia.org

RNN: Recurrent Neural Networks

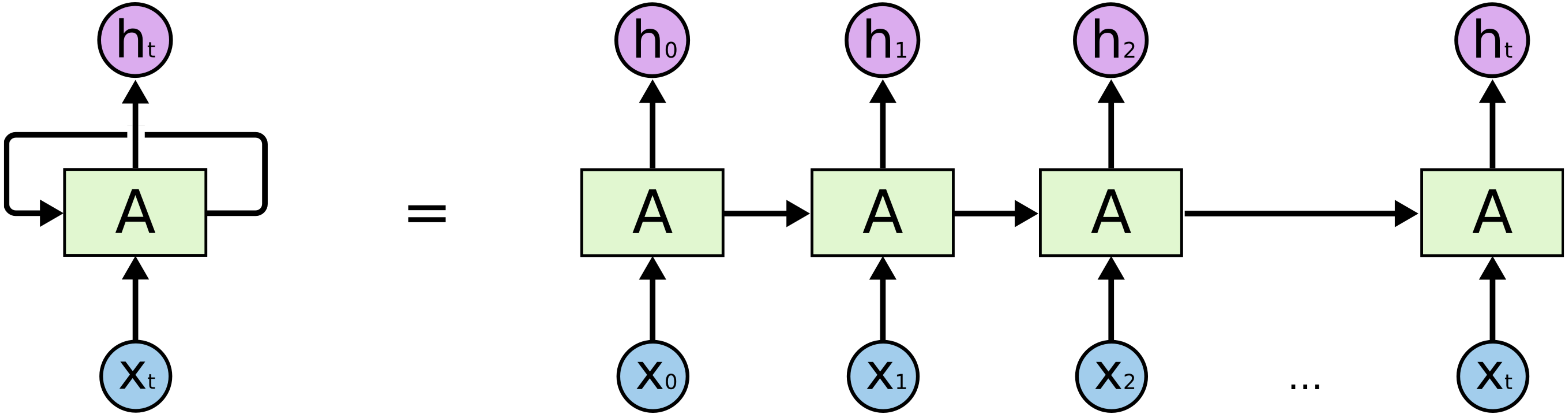
Making predictions with respect to time



colah.github.io

RNN: Recurrent Neural Networks

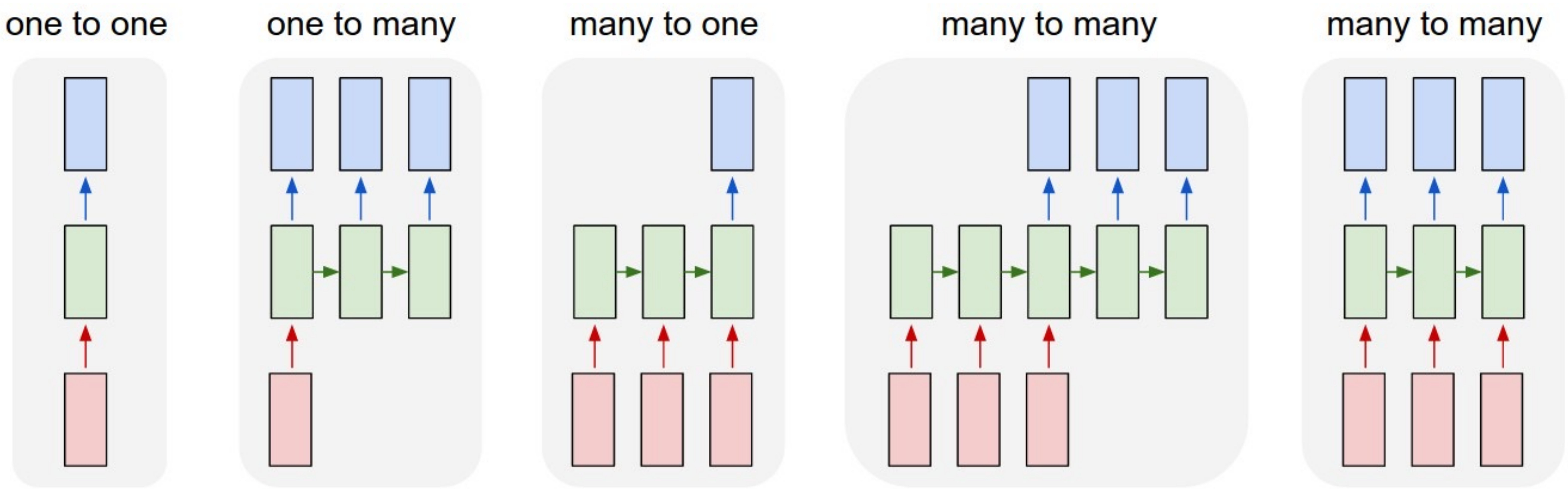
Making predictions with respect to time



colah.github.io

Different tasks, different architectures

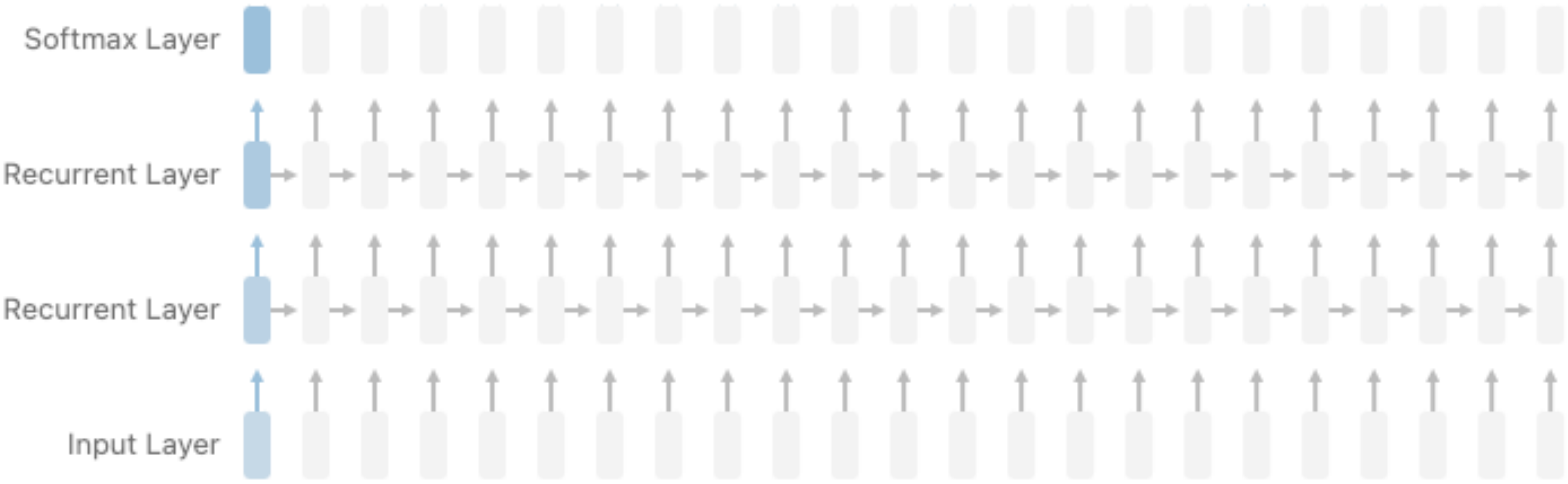
Making predictions with respect to time



karpathy.github.io

RNNs have problems

Vanishing Gradients cause short context lengths

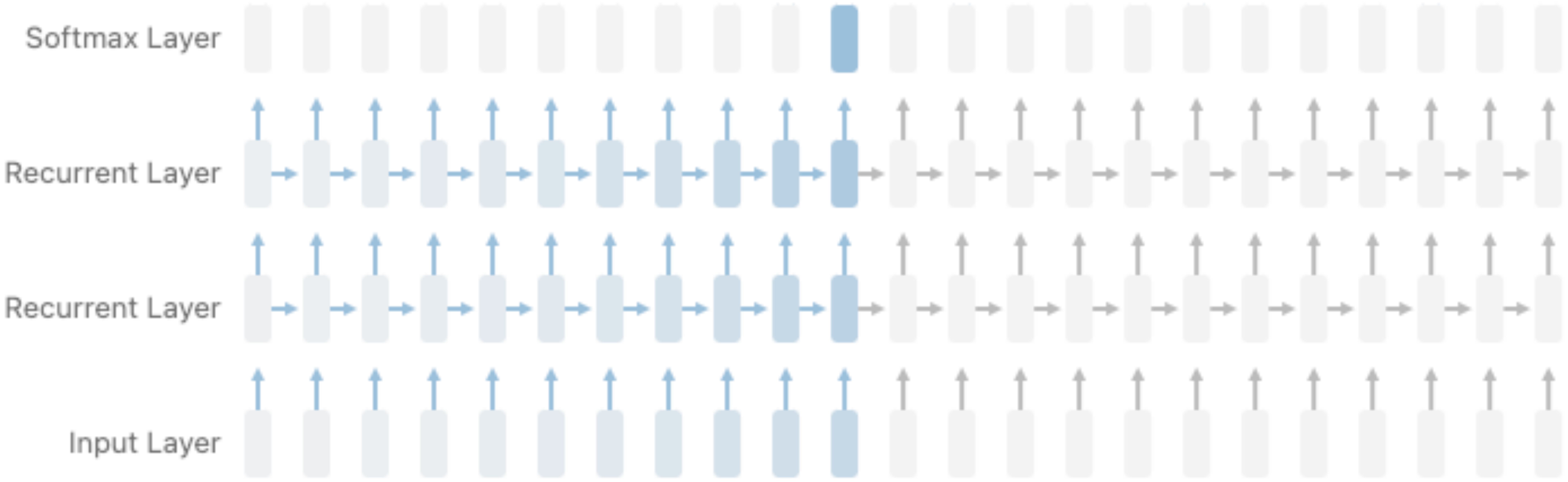


Vanishing Gradient: where the contribution from the earlier steps becomes insignificant in the gradient for the vanilla RNN unit.

distill.pub/2019/memorization-in-rnns/

RNNs have problems

Vanishing Gradients cause short context lengths

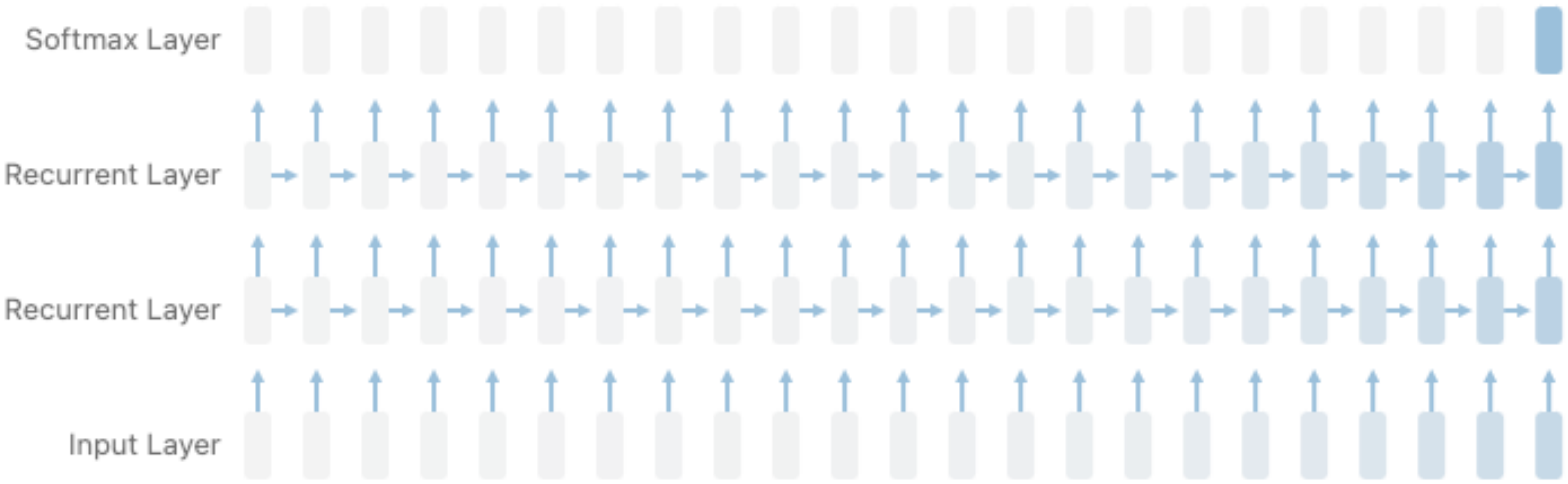


Vanishing Gradient: where the contribution from the earlier steps becomes insignificant in the gradient for the vanilla RNN unit.

distill.pub/2019/memorization-in-rnns/

RNNs have problems

Vanishing Gradients cause short context lengths

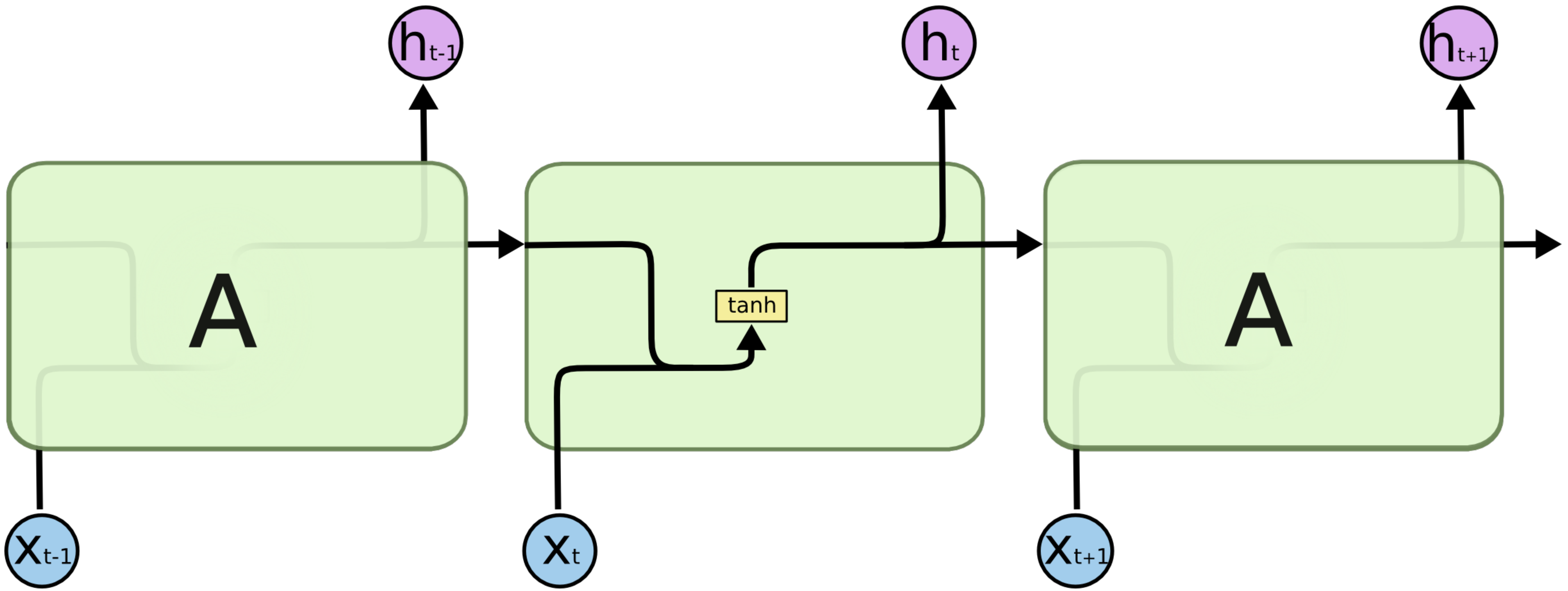


Vanishing Gradient: where the contribution from the earlier steps becomes insignificant in the gradient for the vanilla RNN unit.

distill.pub/2019/memorization-in-rnns/

RNN variants tackle vanishing gradients

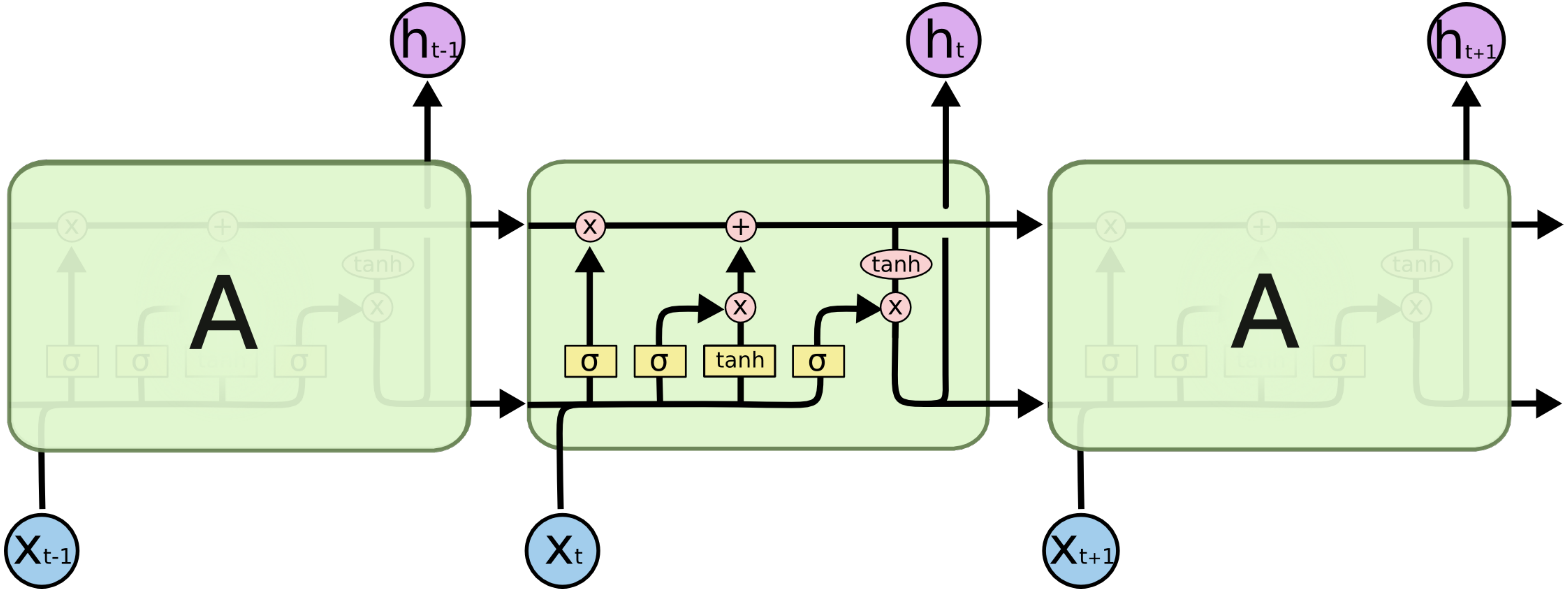
Still, the problem of limited context length remains



colah.github.io

RNN variants tackle vanishing gradients

Still, the problem of limited context length remains



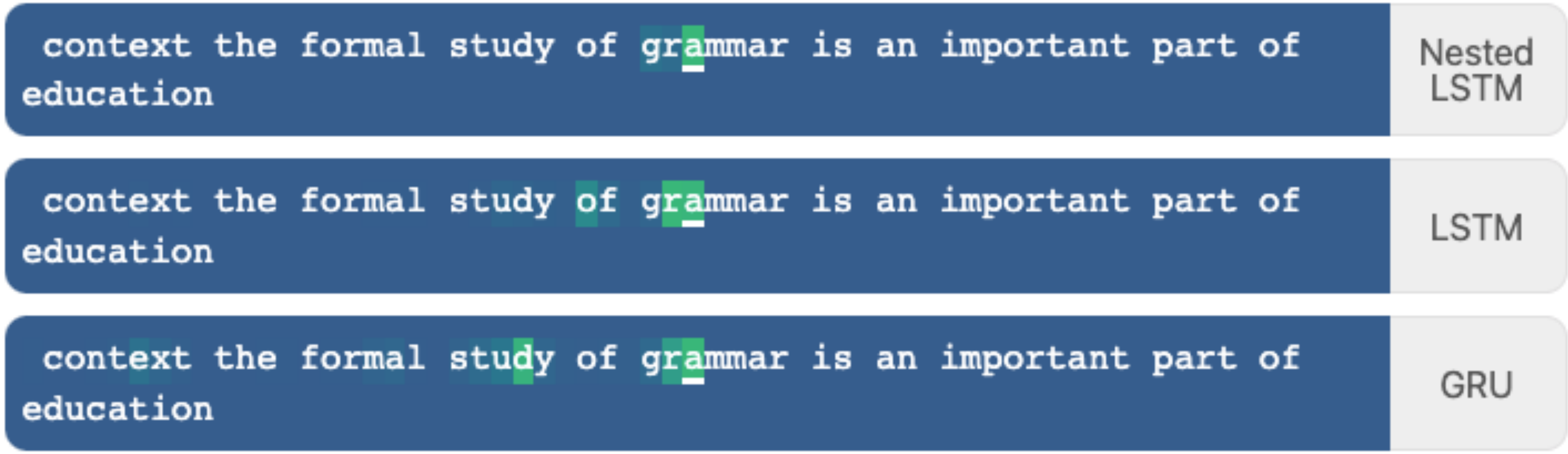
colah.github.io

RNNs have problems

Vanishing Gradients cause short context lengths

Visualizing memorization in RNNs

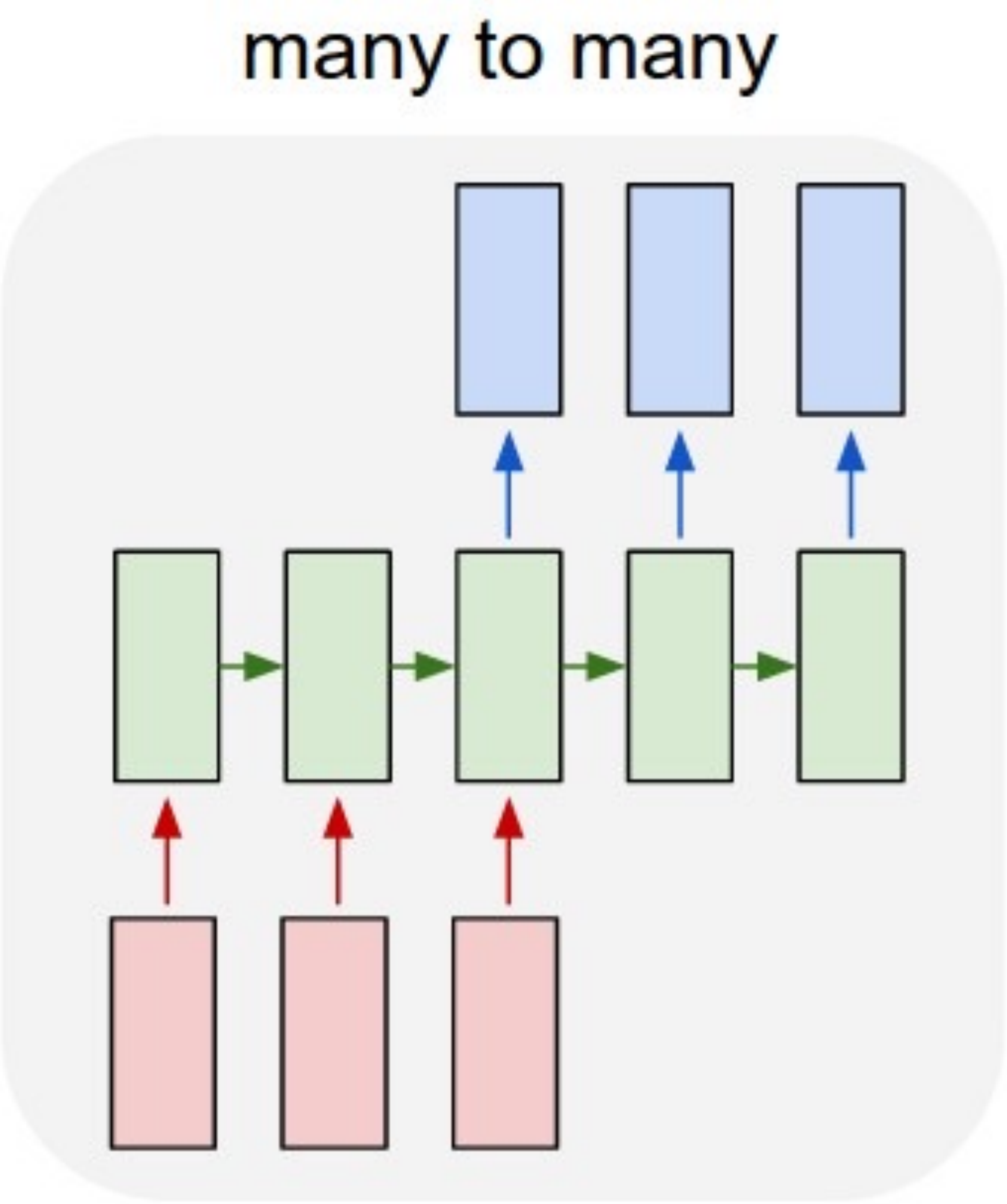
Inspecting gradient magnitudes in context can be a powerful tool to see when recurrent units use short-term or long-term contextual understanding.



distill.pub/2019/memorization-in-rnns/

RNNs have other problems, too

No parallelisation possible

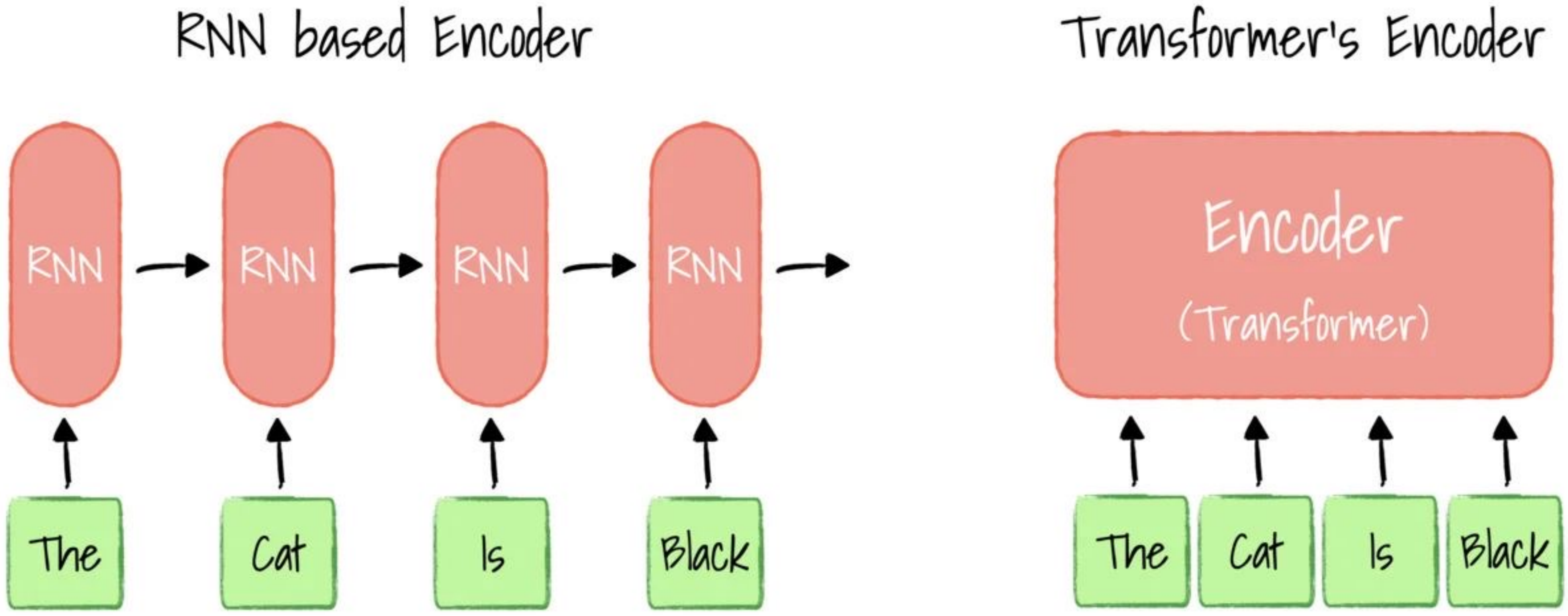


karpathy.github.io

3. Transformers

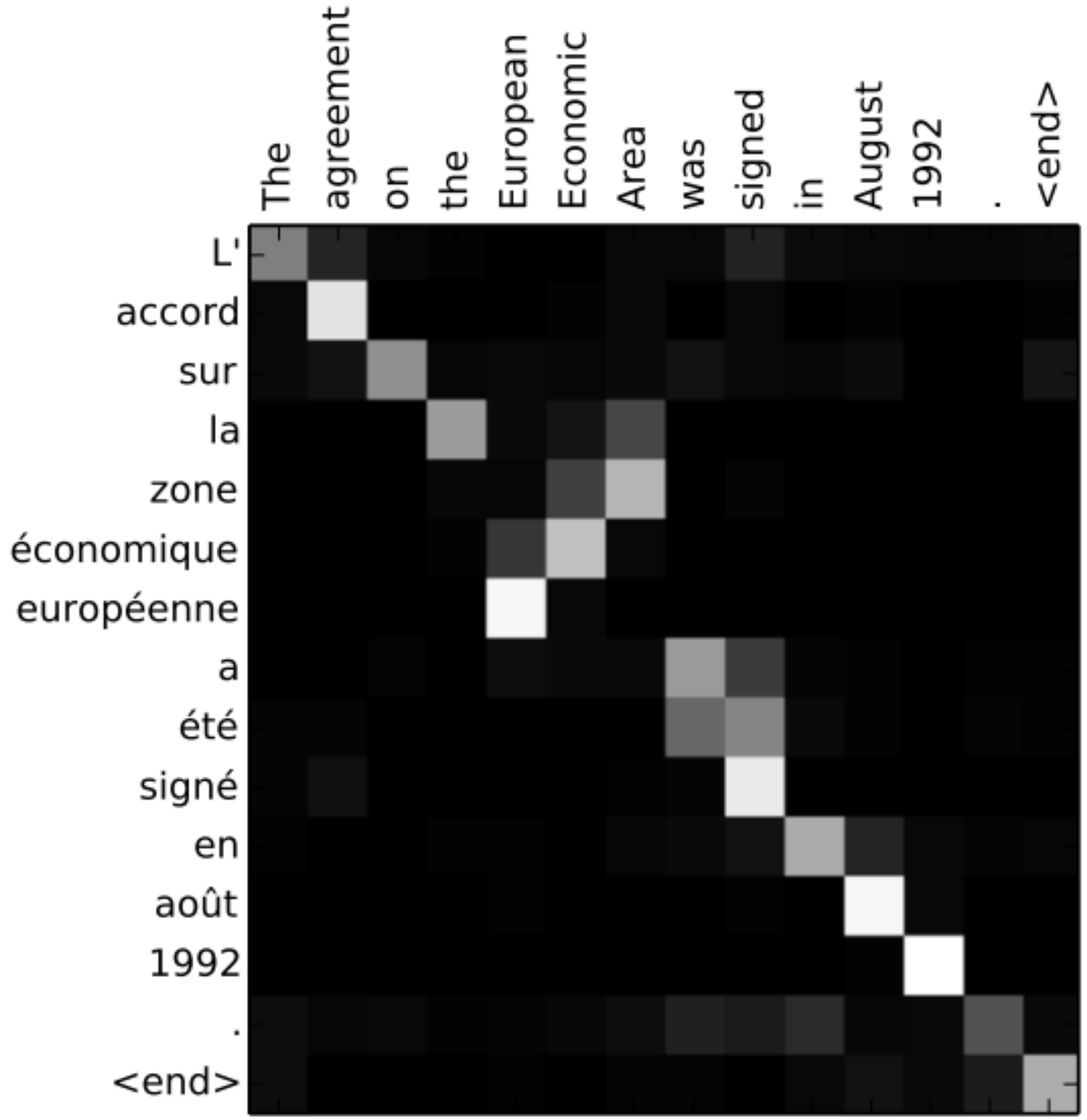
Transformers to the rescue

Parallel instead of sequential encoding with attention



What is Attention?

Allowing every word to be influenced by any other word



Bahdanau et al, 2014, *arxiv*

What is Attention?

Apparently it is all you need

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

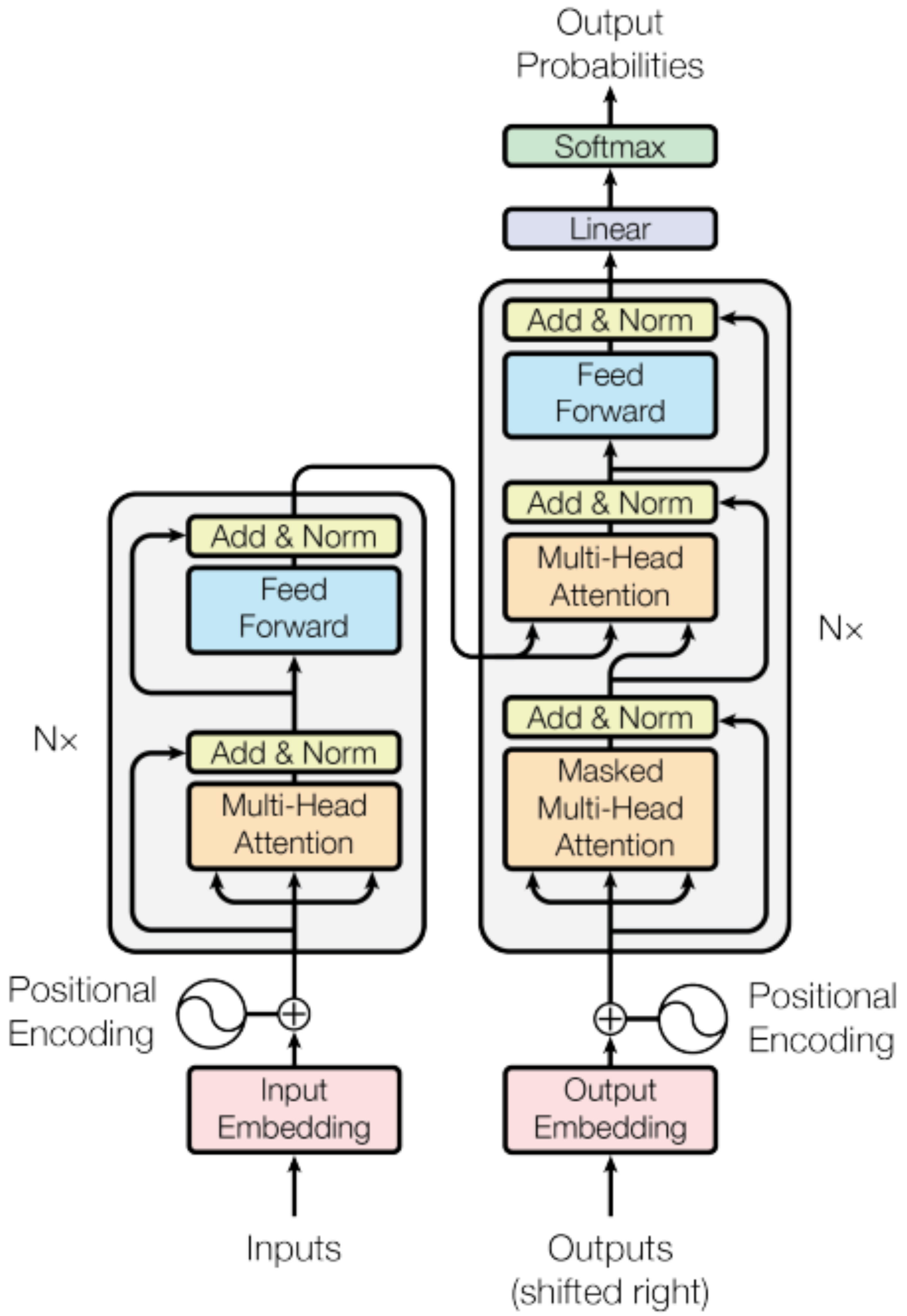
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

The Transformer

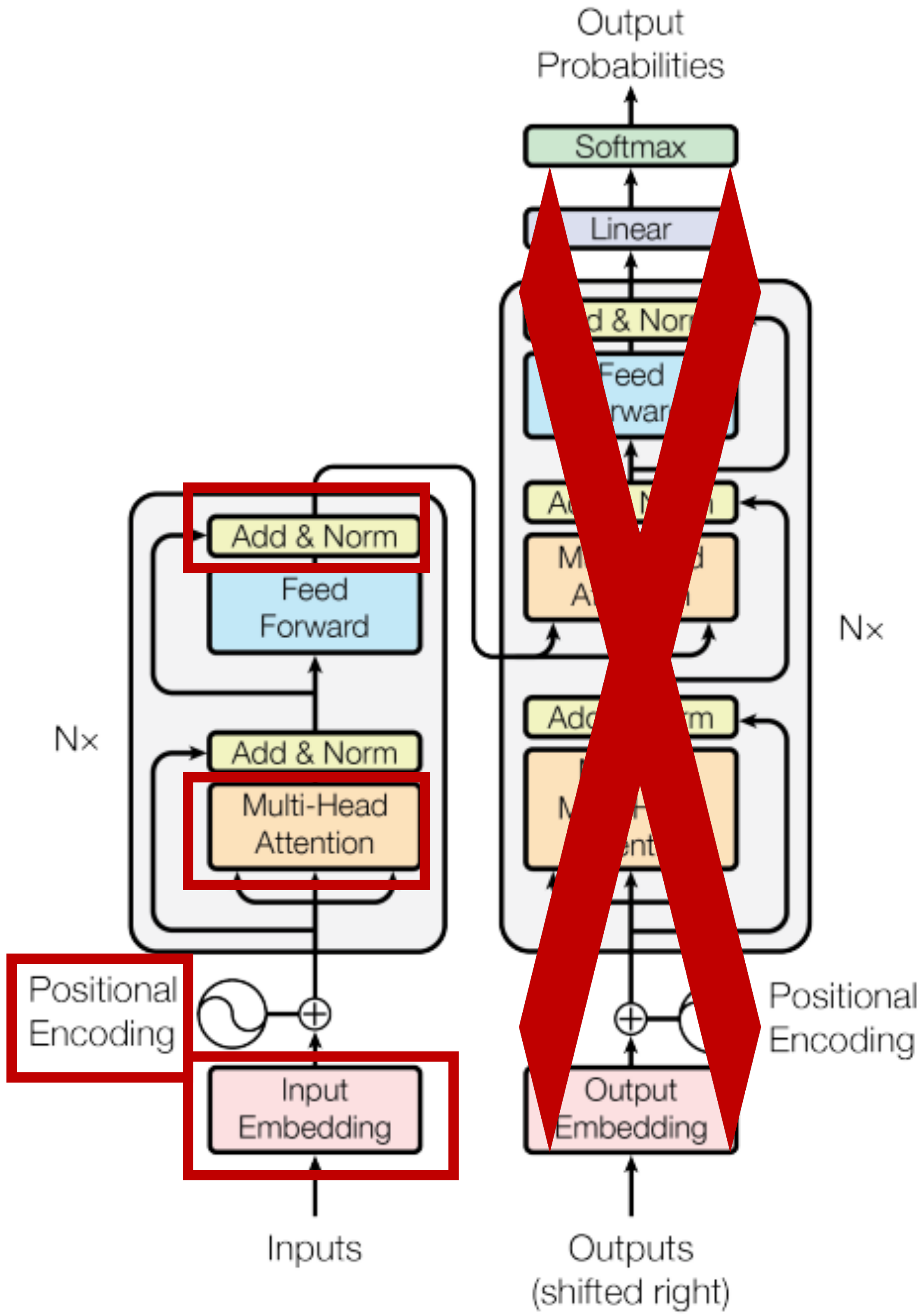
Not as scary as it looks like



Vaswani et al, 2017, *arxiv*

The Transformer

Not as scary as it looks like



Vaswani et al, 2017, *arxiv*

Input Embedding

Our computer does not understand English

Vocabulary

One-hot vectors

Input Embedding

From one-hot encodings to word embeddings

One-hot vectors

Word embeddings

Input Embedding

Play with a few word embeddings yourself

https://lamyiowce.github.io/word2viz/

https://ronxin.github.io/wevi/

http://projector.tensorflow.org/

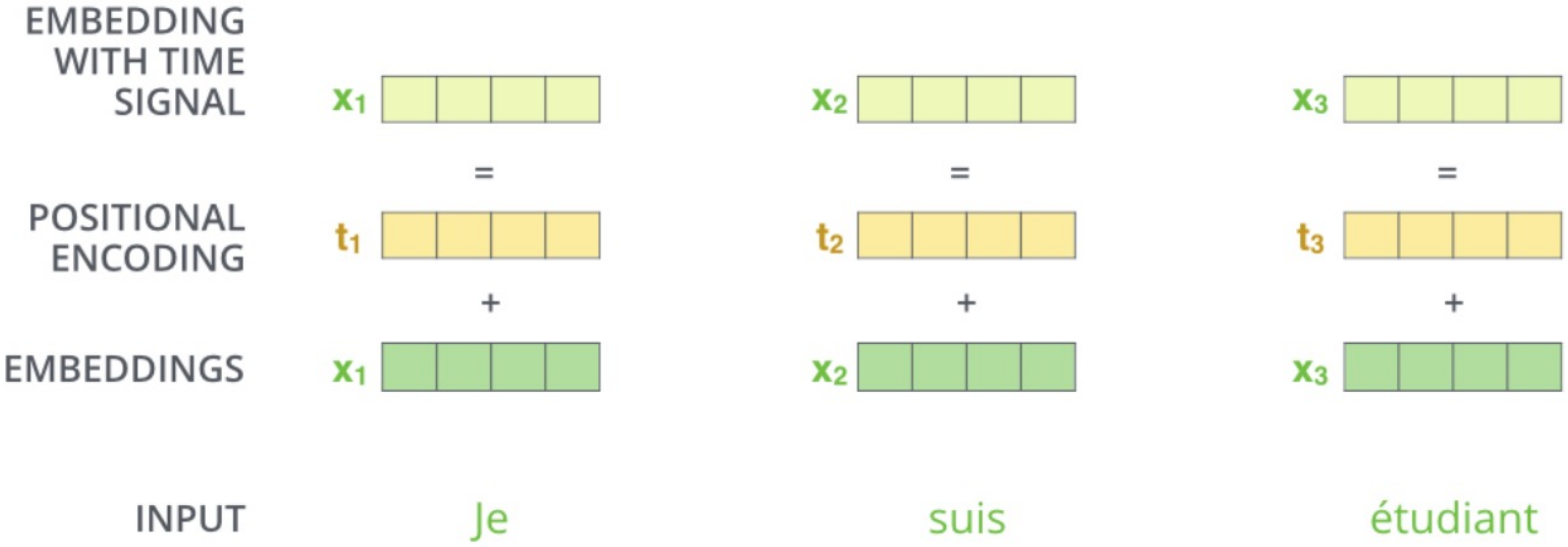
Positional Embedding

We must tell our computer what comes first and what later

The Illustrated Tranformer, Jay Allamar

Positional Embedding

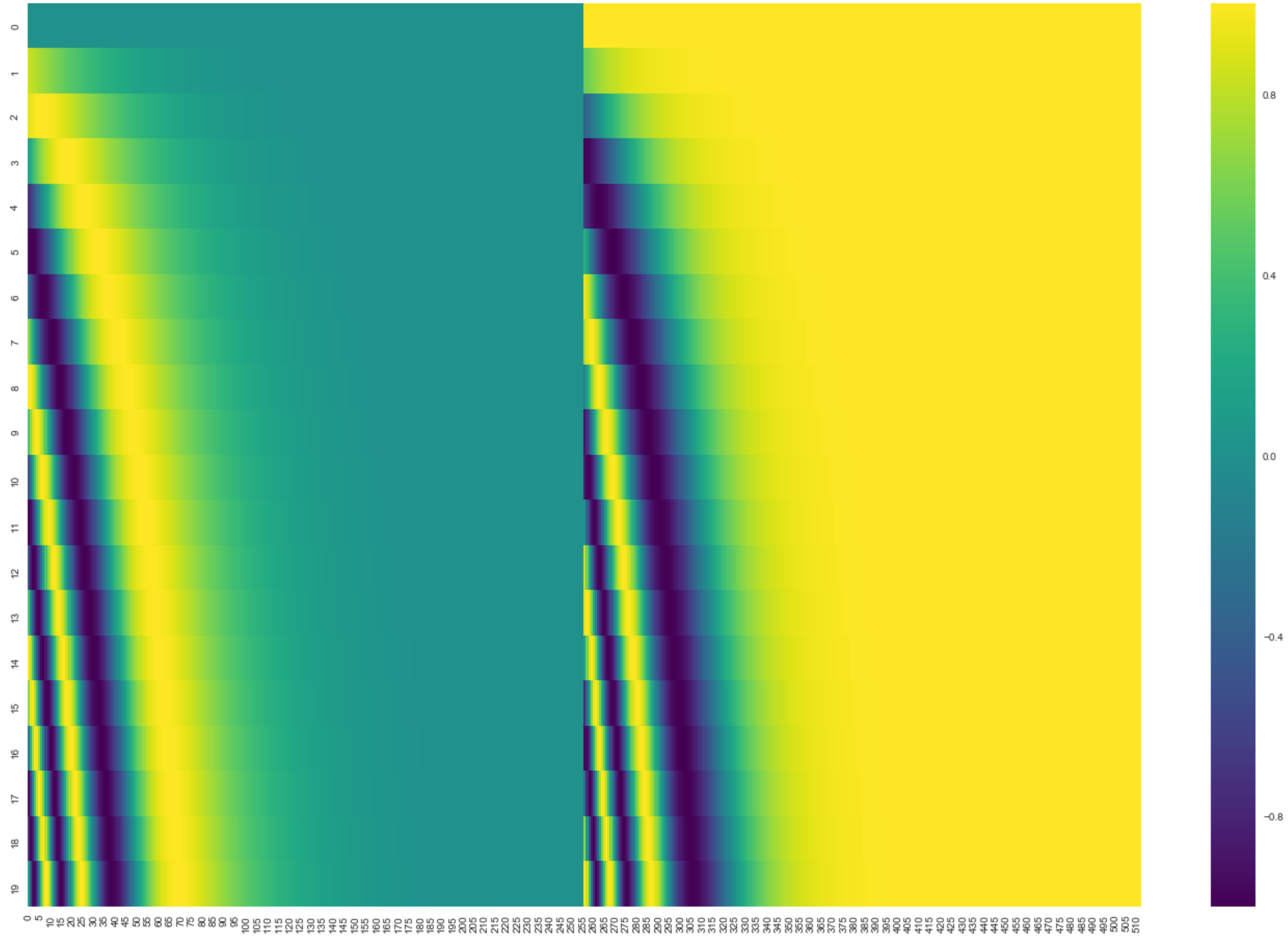
We must tell our computer what comes first and what later



The Illustrated Tranformer, Jay Allamar

Positional Embedding

We must tell our computer what comes first and what later



$$P(k, 2i) = \sin\left(\frac{k}{n^{2i/d}}\right)$$

$$P(k, 2i+1) = \cos\left(\frac{k}{n^{2i/d}}\right)$$

The Illustrated Tranformer, Jay Allamar

Attention

Looking at everyone around you to determine your update

- Input: sequence of tensors

  $\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_t$

Attention

Looking at everyone around you to determine your update

- Input: sequence of tensors

  $\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_t$

- Output: sequence of tensors, each one a weighted sum of the input sequence

  $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_t$

  $$\mathbf{y}_i = \sum_j w_{ij} \mathbf{x}_j$$

Attention

Looking at everyone around you to determine your update

- Input: sequence of tensors

  $\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_t$

- Output: sequence of tensors, each one a weighted sum of the input sequence

  $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_t$

  $$\mathbf{y}_i = \sum_j w_{ij} \mathbf{x}_j$$

  - weight is just a dot product $w'_{ij} = \mathbf{x}_i{}^\mathsf{T} \mathbf{x}_j$

Attention

Looking at everyone around you to determine your update

- Input: sequence of tensors

  $\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_t$

- Output: sequence of tensors, each one a weighted sum of the input sequence

  $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_t$

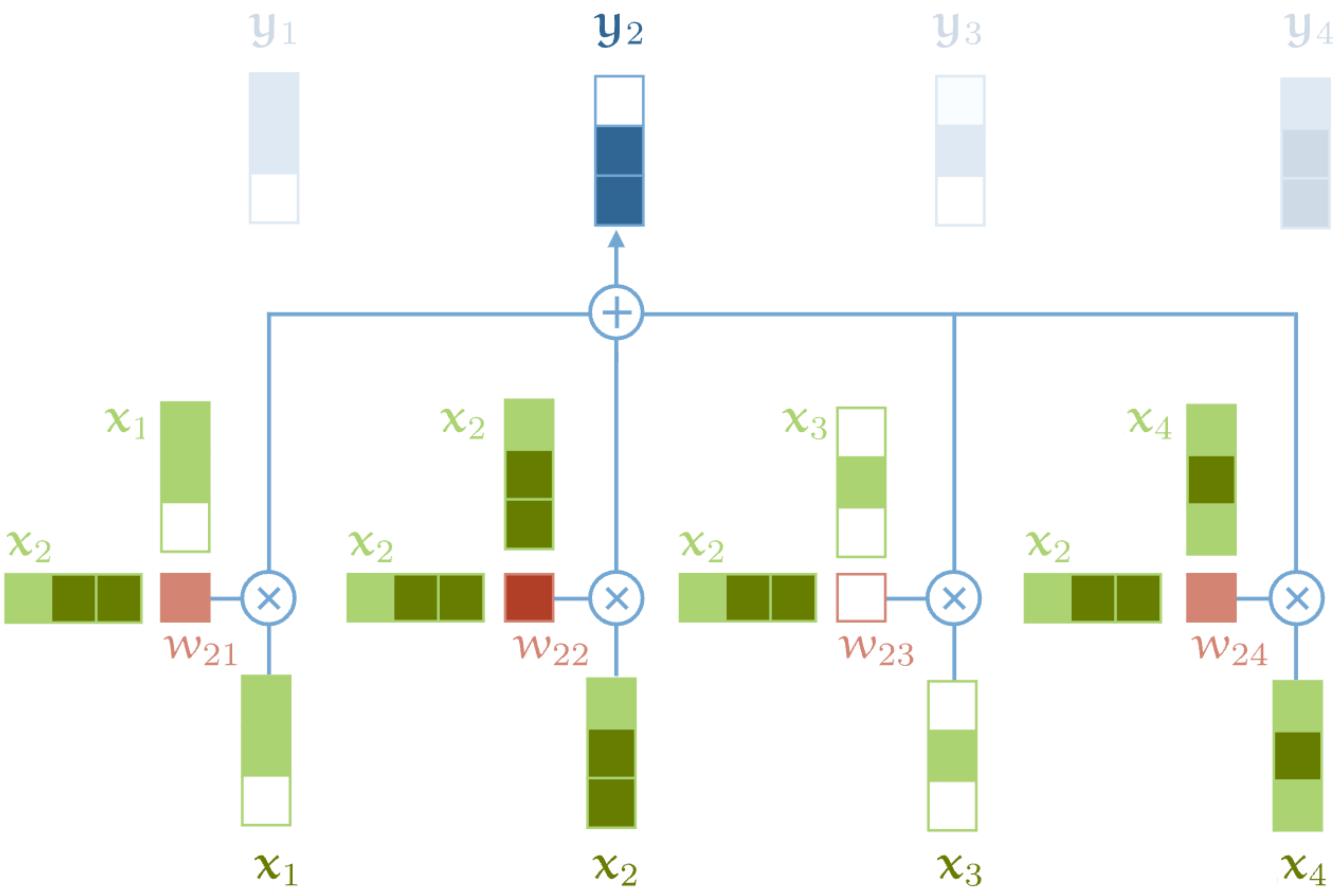
  $$\mathbf{y}_i = \sum_j w_{ij} \mathbf{x}_j$$

  - weight is just a dot product $w'_{ij} = \mathbf{x}_i{}^{\mathsf{T}} \mathbf{x}_j$

  - make it sum to 1 $$w_{ij} = \frac{\exp w'_{ij}}{\sum_j \exp w'_{ij}}$$

Attention

Looking at everyone around you to determine your update



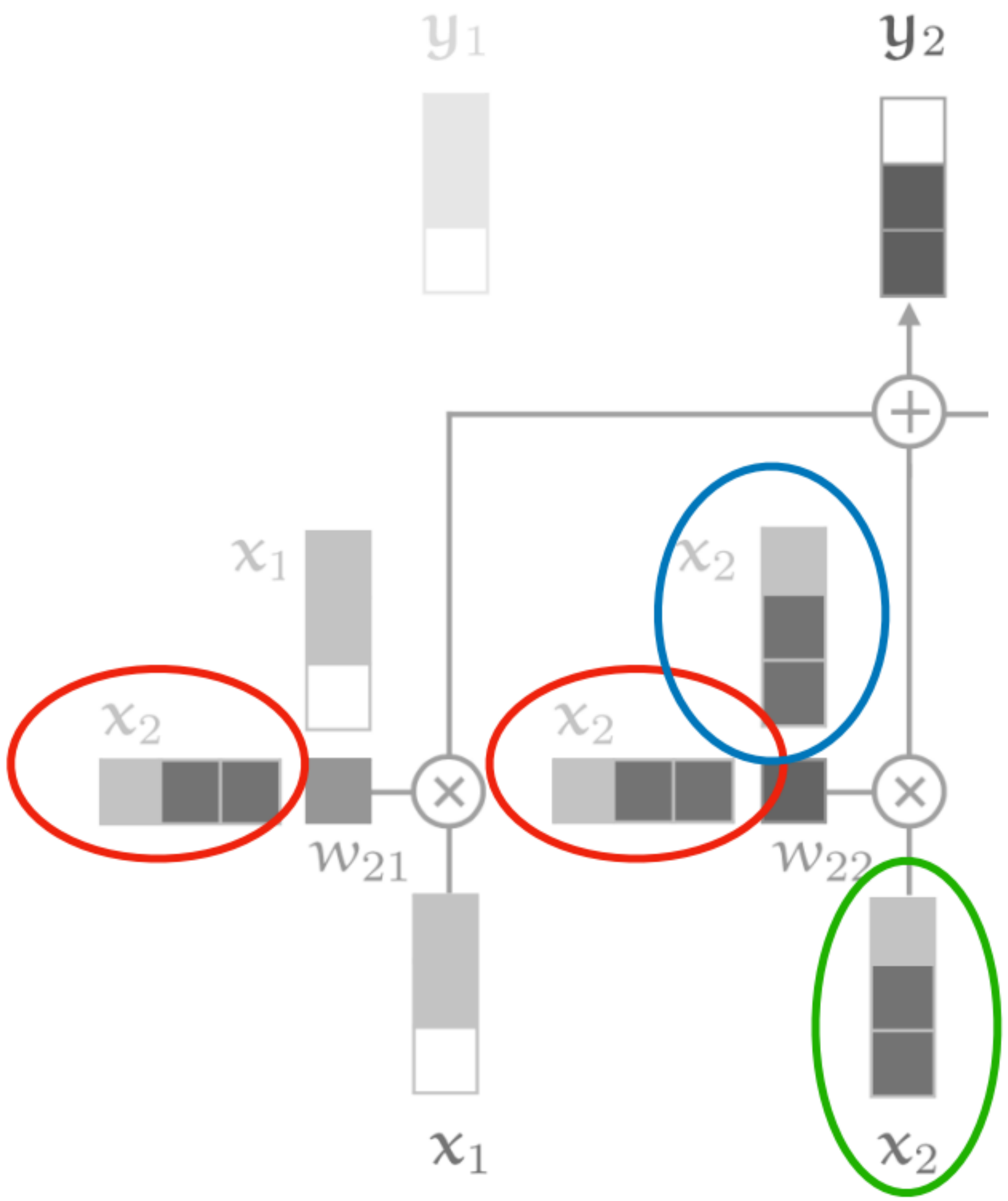
Transformers from Scratch, peterbloem.nl

Attention

Learning the weights

Query, Key, Value

- Every input vector x_i is used in 3 ways:
  - Query
  - Key
  - Value

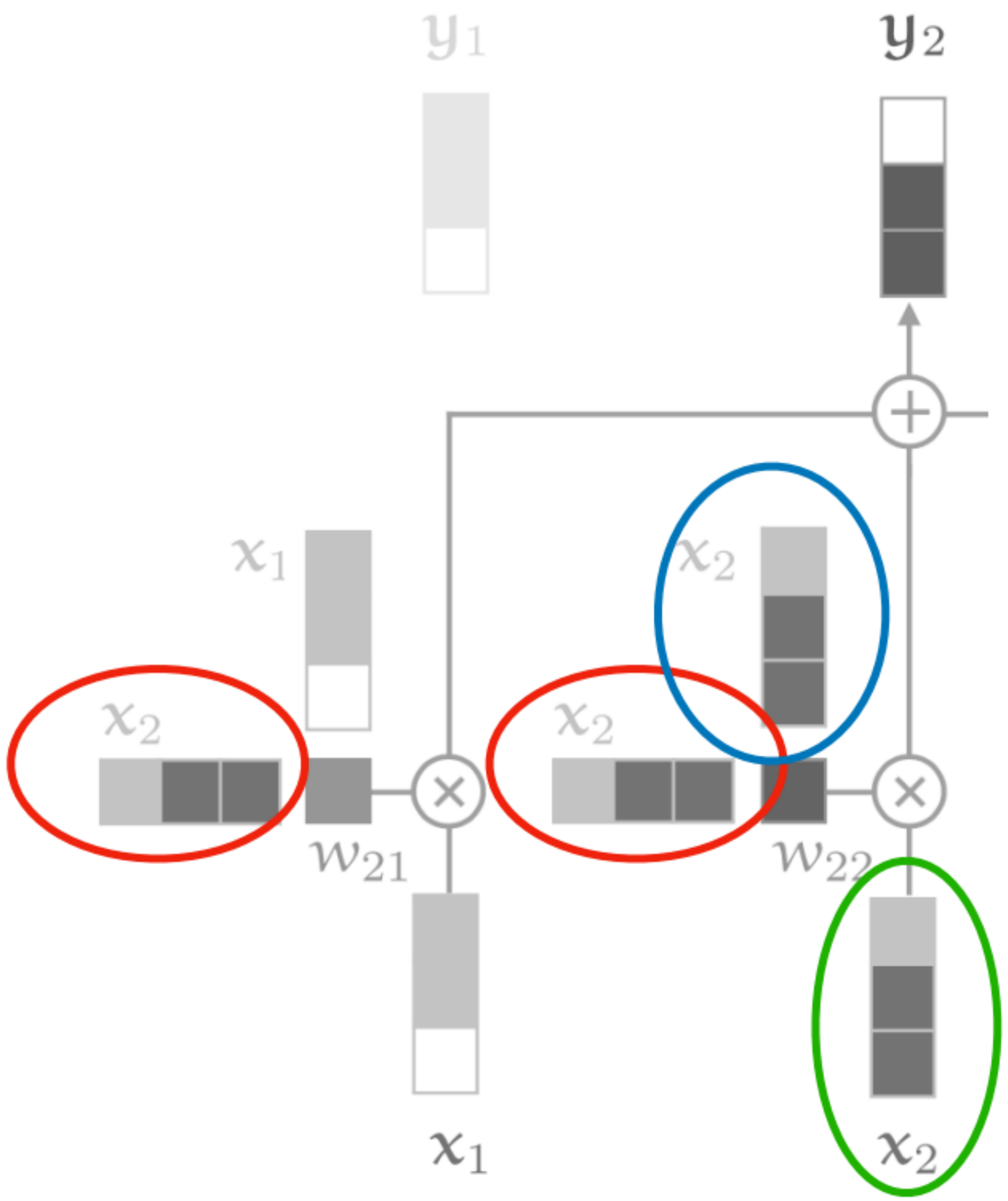


Attention

Learning the weights

Query, Key, Value

- Every input vector x_i is used in 3 ways:
  - Query — **What am I looking for?**
  - Key — **What do I have?**
  - Value — **What do I reveal/give to others?**



Attention

Learning the weights

- We can process each input vector to fulfill the three roles with matrix multiplication
- Learning the matrices → learning attention

What am I looking for? | What do I have? | What do I reveal/give to others?

$$\boxed{\mathbf{q}_i = \mathbf{W}_q \mathbf{x}_i} \quad \boxed{\mathbf{k}_i = \mathbf{W}_k \mathbf{x}_i} \quad \boxed{\mathbf{v}_i = \mathbf{W}_v \mathbf{x}_i}$$

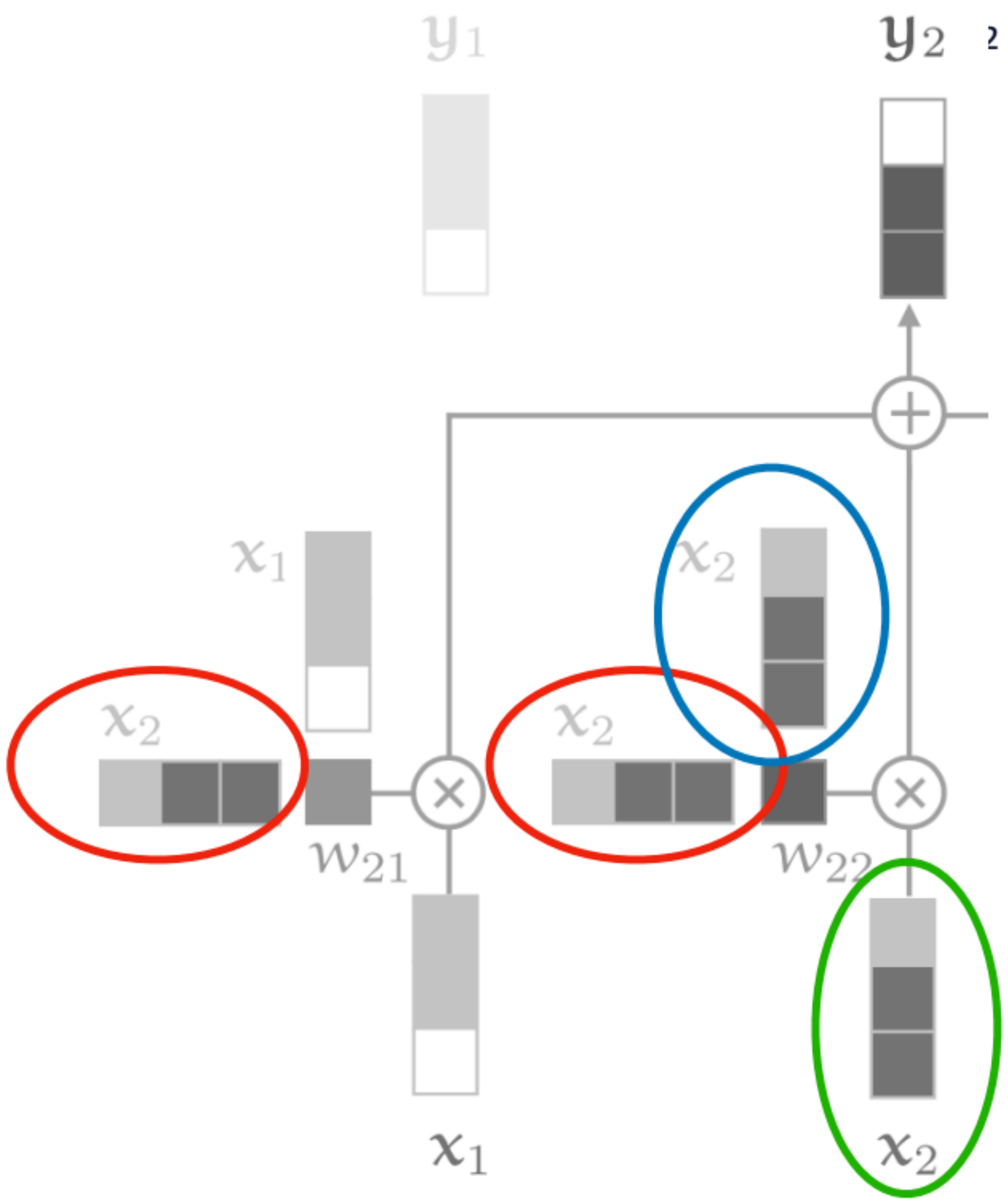
$$w'_{ij} = \mathbf{q}_i{}^{\mathsf{T}} \mathbf{k}_j$$

$$w_{ij} = \text{softmax}(w'_{ij})$$

$$\mathbf{y}_i = \sum_j w_{ij} \mathbf{v}_j \,.$$

Imagine you are in a library

How do you answer a question you have?

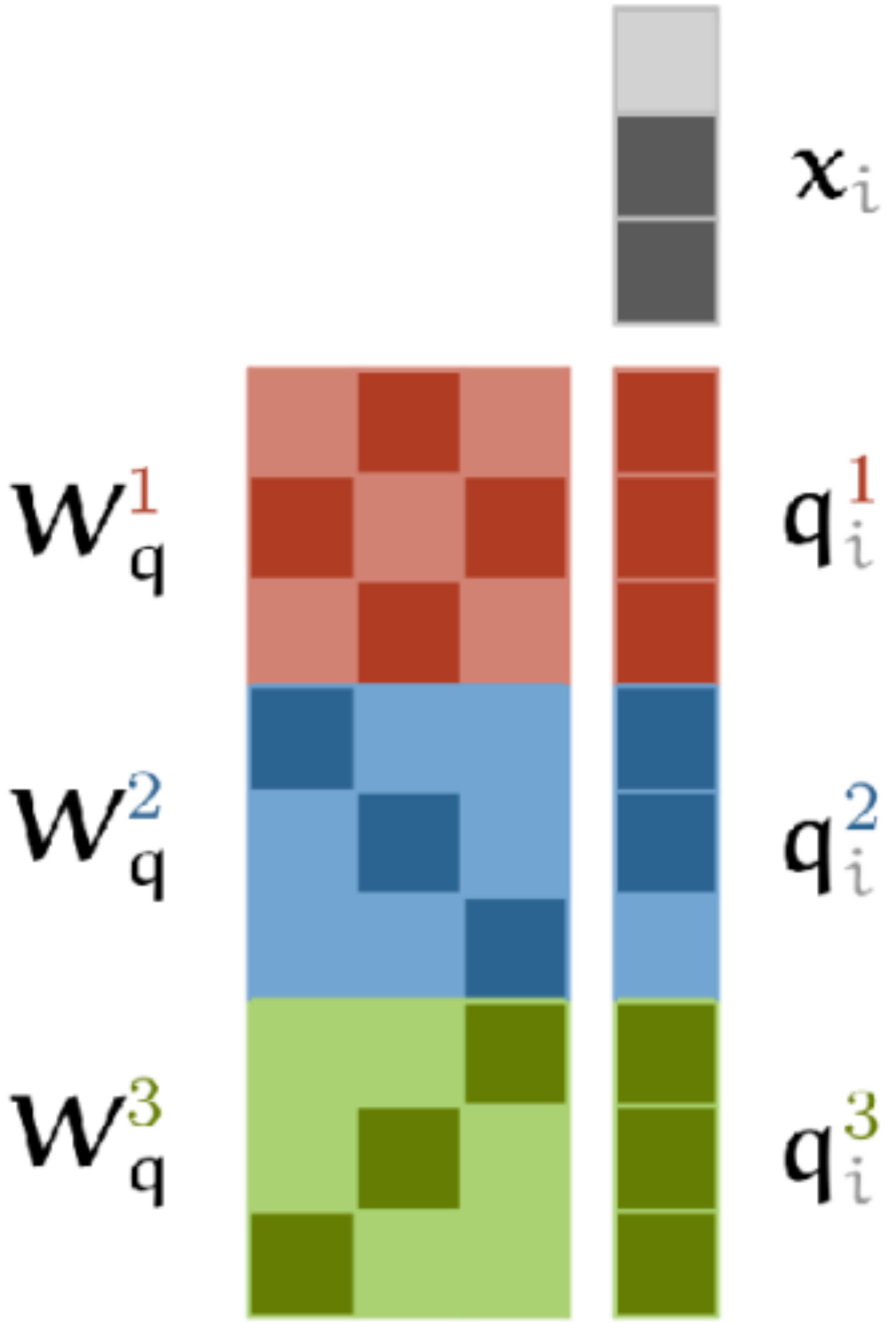


- Query — The question you have
- Key — The titles books have on their spines
- Value — Information the book contains

Multi-head attention

Looking at everyone around you to determine your update

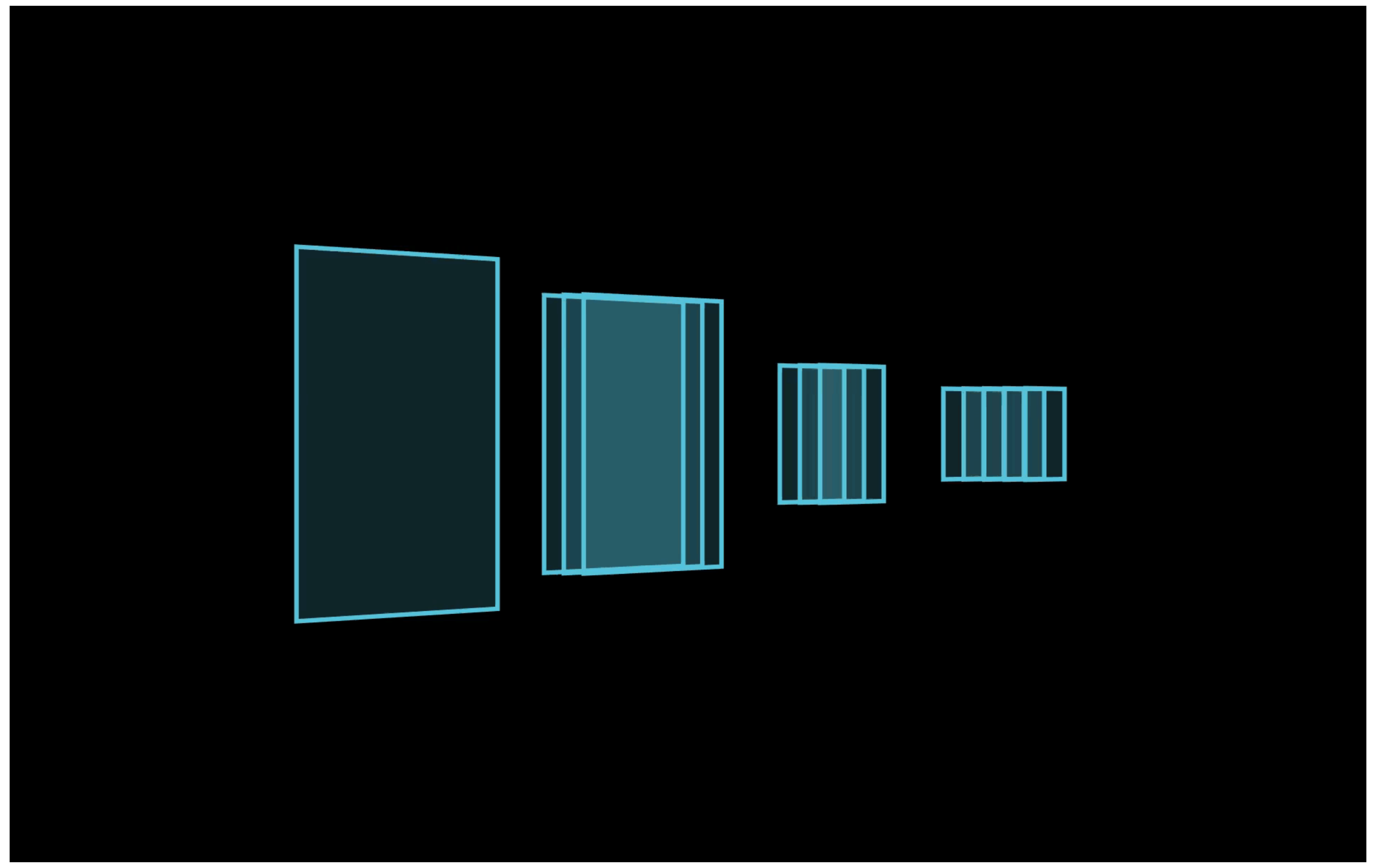
- Multiple "heads" of attention just means learning different sets of W_q, W_k, and W_v matrices simultaneously.
- Implemented as just a single matrix...

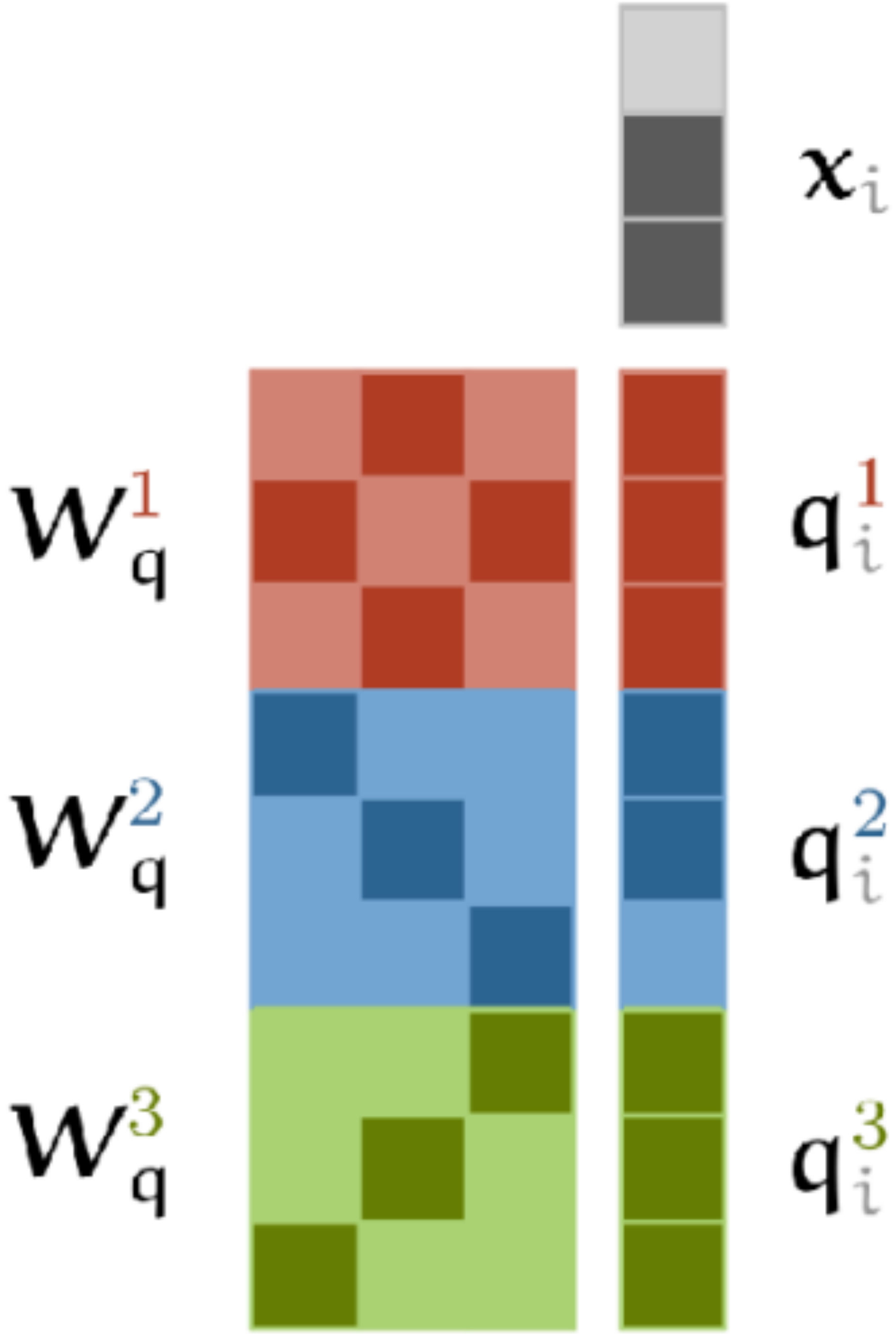


Multi-head attention

Looking at everyone around you to determine your update

- Multiple "heads" of attention just means learning different sets of W_q, W_k, and W_v matrices simultaneously.
- Implemented as just a single matrix...

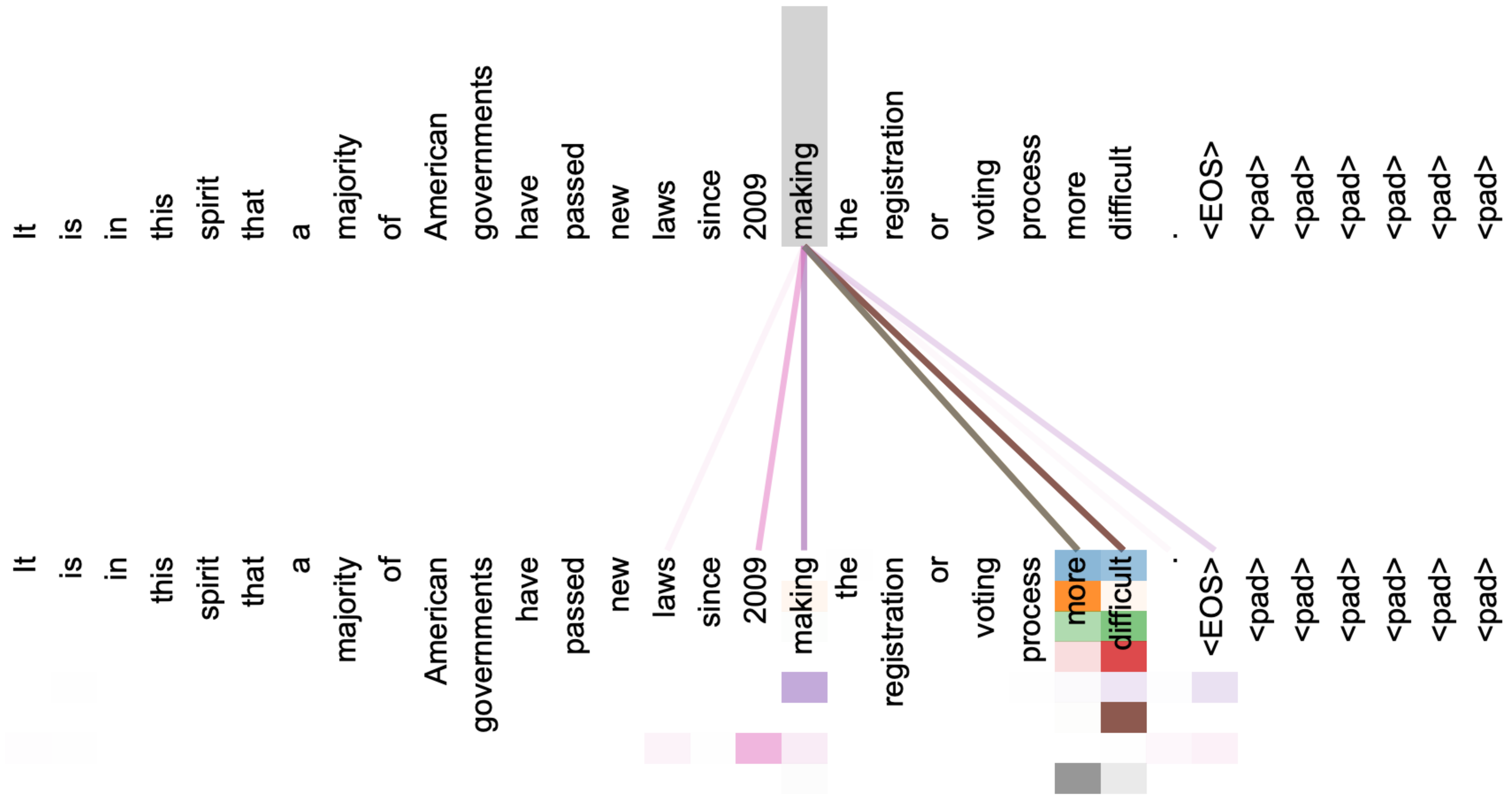




Multi-head attention

Different heads attend to different parts in a sentence

Attention Visualizations

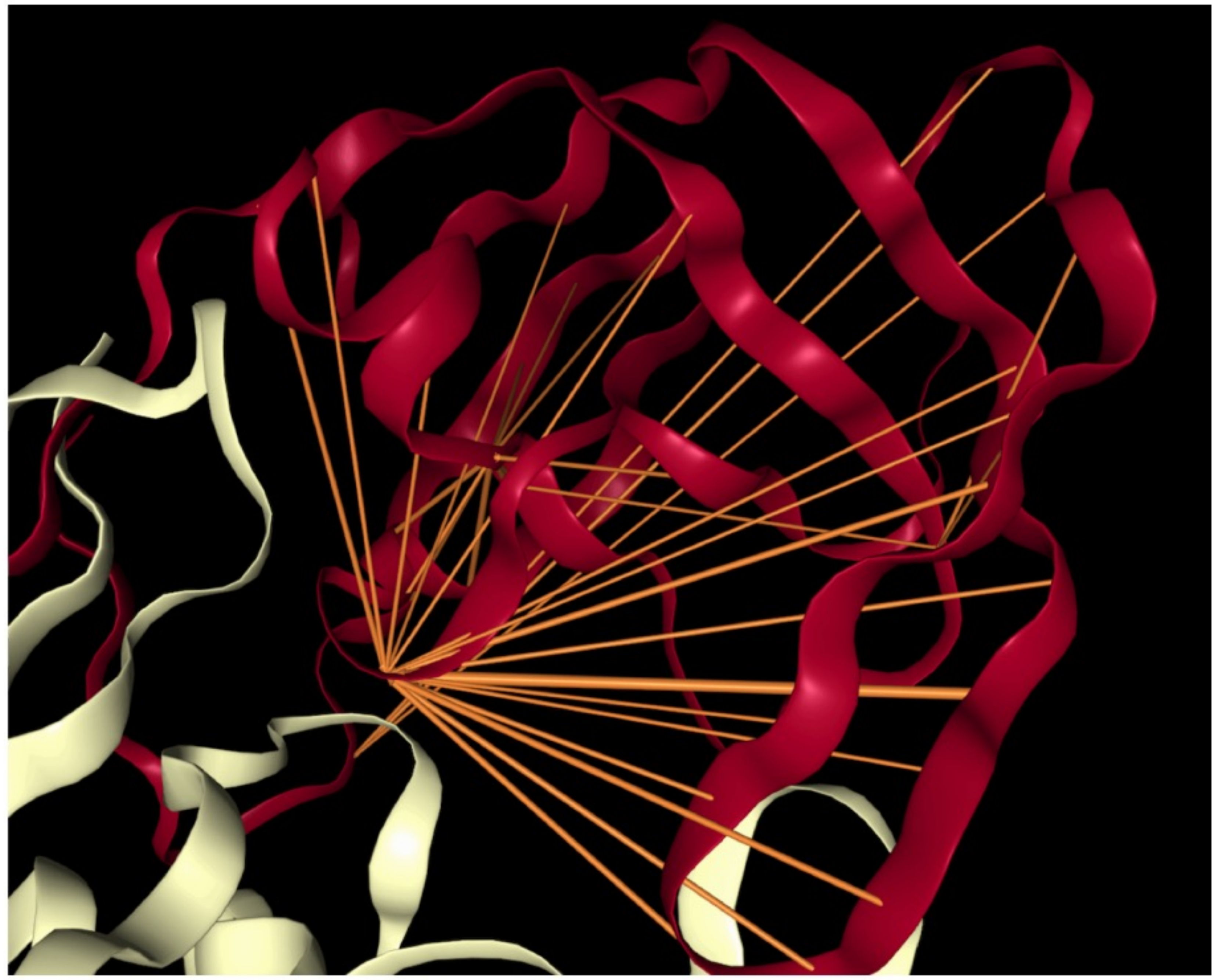


Vaswani et al, 2017, *arxiv*

Multi-head attention

The same applies for proteins

(a) Attention in head 12-4, which targets amino acid pairs that are close in physical space (see inset subsequence 117D-157I) but lie apart in the sequence. Example is a *de novo* designed TIM-barrel (5BVL) with characteristic symmetry.



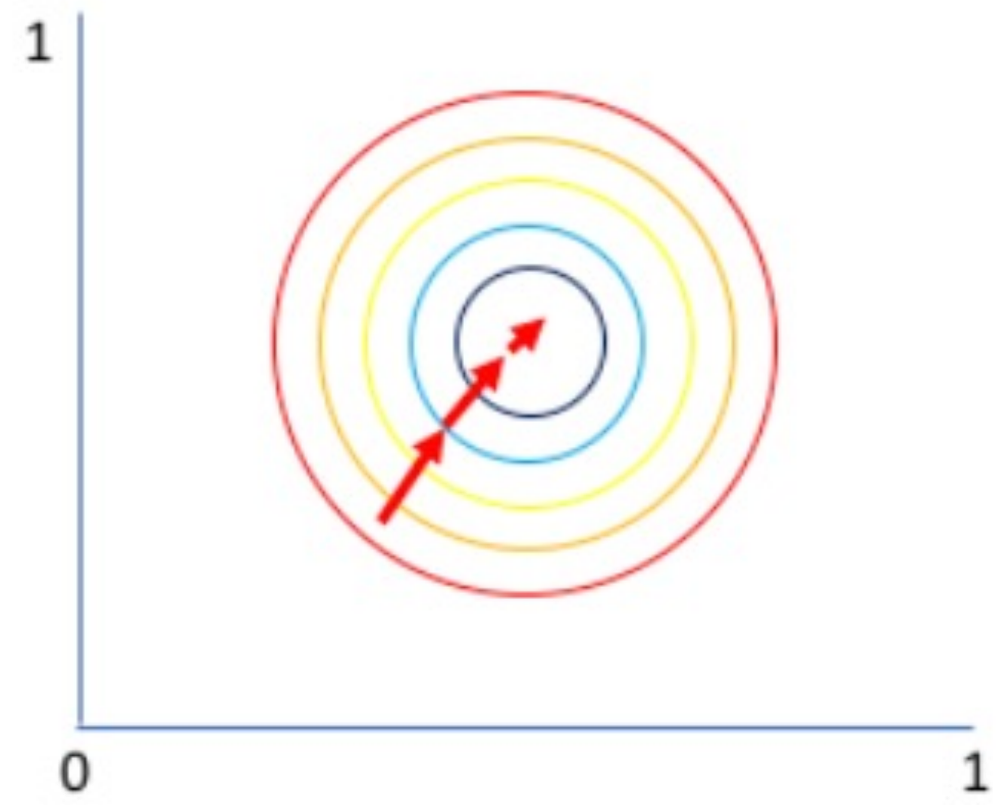
(b) Attention in head 7-1, which targets binding sites, a key functional component of proteins. Example is HIV-1 protease (7HVP). The primary location receiving attention is 27G, a binding site for protease inhibitor small-molecule drugs.

Vig et al, ILCR 2021

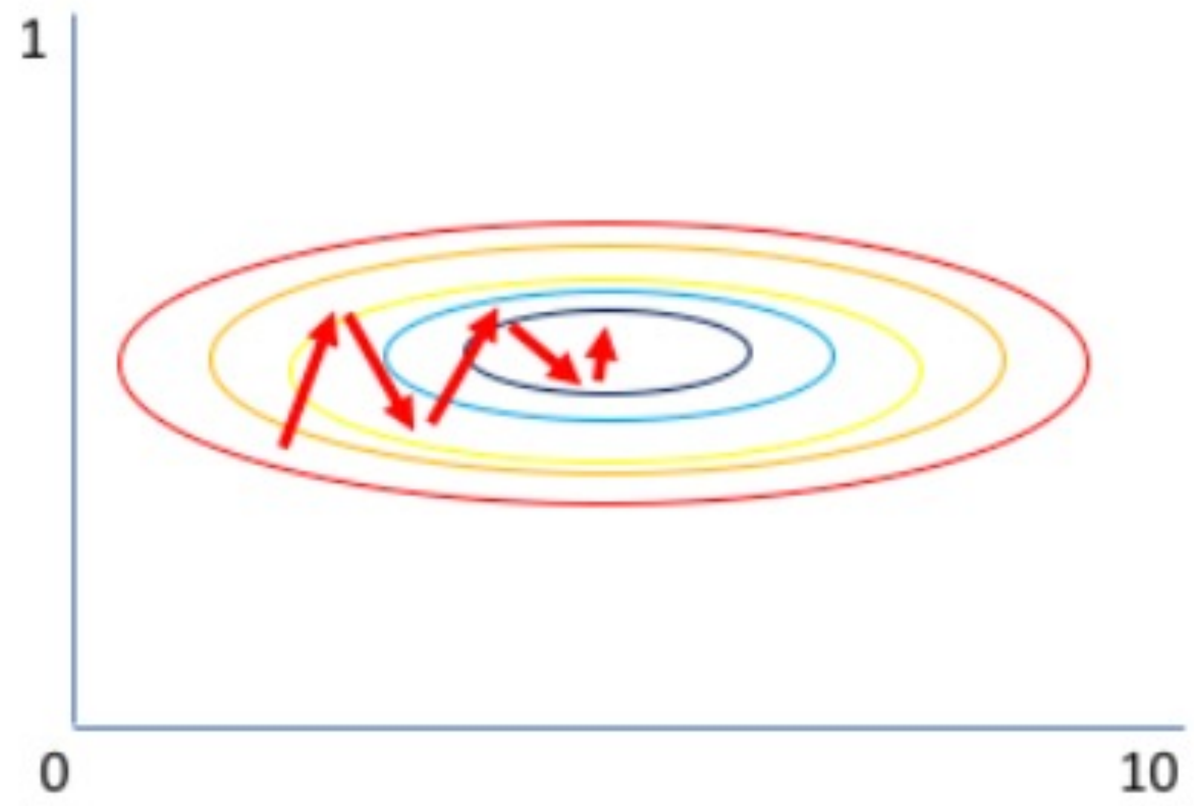
Layer Normalization

Standardize means and stds of input vectors

- Neural net layers work best when input vectors have uniform mean and std in each dimension
- As inputs flow through the network, means and std's get blown out.
- Layer Normalization is a hack to reset things to where we want them in between layers.



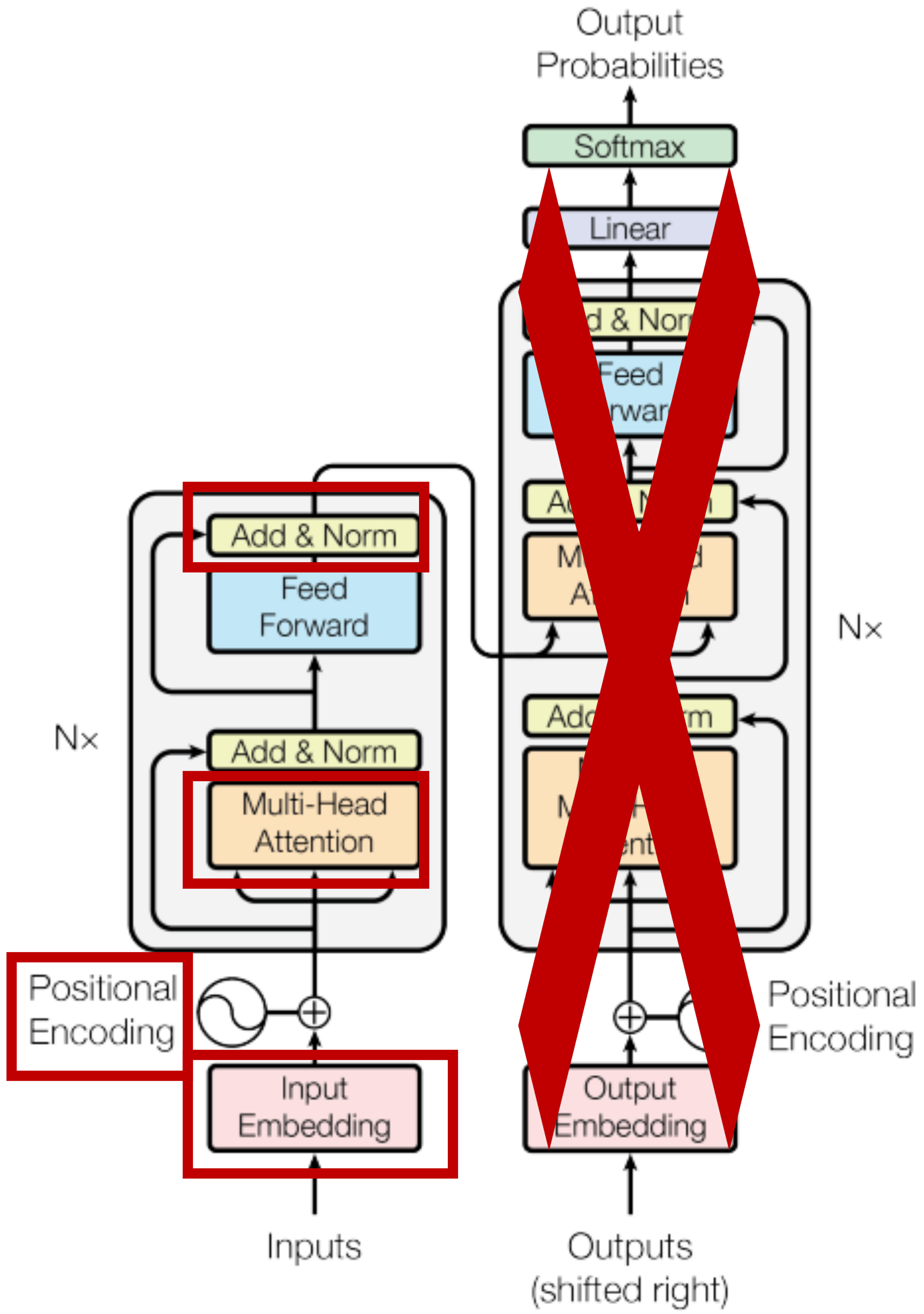
Both parameters can be updated in equal proportions



Gradient of larger parameter dominates the update

The Transformer

Not as scary as it looks like



Vaswani et al, 2017, *arxiv*

Many good blogs about Transformers

I leave it to you to choose the ones you like best

1. The Illustrated Transformer (Pictures)
2. The Annotated Transformer (Code)
3. Transformers from Scratch (Code)
4. Transformers from Scratch (Again, this time long detailed deep dive)
5. An Intuitive Introduction to Transformers (Pictures)
6. The Transformer – Attention is All You Need ()
7. Primers – Transformer (Long, detailed Deep Dive)
8. Some Intuition on Attention and the Transformer (Short insights)
9. Transformer Math (If you want to implement a big one in practice)

🥡 Takeaway 🥡

Model architectures are influenced by the **inductive bias of the data.** is present, while **Respecting symmetry** and **making models scale well** are two popular approaches these days.