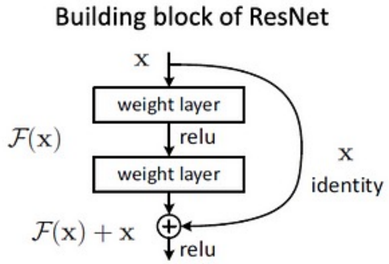
Transformers, CNNs in disguise

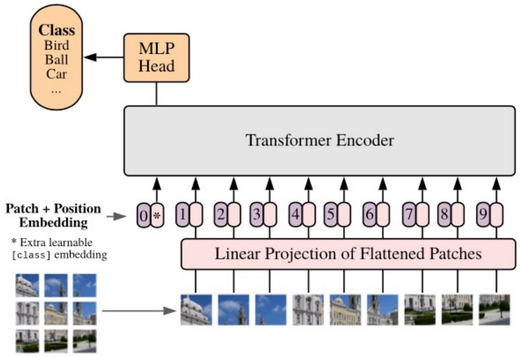
Lisa Dunlap, CS280 Sp24

Slides adapted from Sara Beery, MIT Deep Learning course

Boring Administrative Stuff

- HW1 grades will be out soon-ish (before the beginning of next week)
- HW3 will be out mid-late next week
  - Transformer-based HW, this lecture is very important
- Please excuse Lisa’s OOD behavior





ImageNet 2009

AlexNet (deepCNN) 2012

Fast R-CNN ResNet 2015

RevNet YOLOv2 **Transformer** 2017

Vision Transformer (ViT) 2021

1998 LeNet

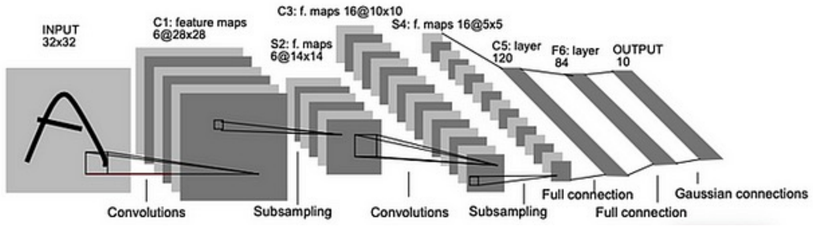
2010 Pascal VOC

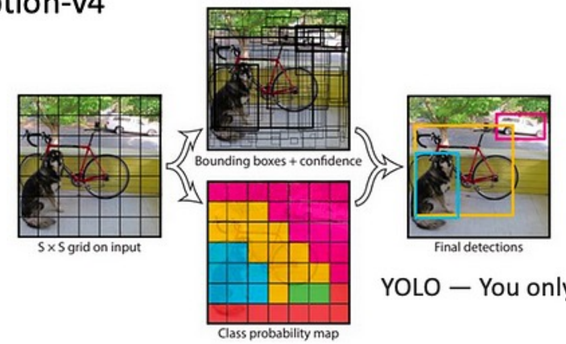
2014 VGG R-CNN GoogLeNet (Inception)

2016 SSD YOLO Inception-v4

2018 YOLOv3 **BERT**

2020 Image GPT





YOLO — You only look once

Rank	Model	Top 1 Accuracy	Top 5 Accuracy	Number of params	GFLOPs	Extra Training Data	Paper	Code	Result	Year	Tags
1	**CoCa** (finetuned)	91.0%		2100M		✓	CoCa: Contrastive Captioners are Image-Text Foundation Models			2022	Transformer, JFT-3B, ALIGN
2	**Model soups** (BASIC-L)	90.98%		2440M		✓	Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time			2022	Conv+Transformer, JFT-3B, ALIGN
3	**Model soups** (ViT-G/14)	90.94%		1843M		✓	Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing			2022	Transformer, JFT-3B

The “roaring 20’s” of visual recognition

https://paperswithcode.com/sota/image-classification-on-imagenet

ViT – vision transformer

Published as a conference paper at ICLR 2021

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Alexey Dosovitskiy*,†, Lucas Beyer*, Alexander Kolesnikov*, Dirk Weissenborn*, Xiaohua Zhai*, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby*,†

*equal technical contribution, †equal advising

Google Research, Brain Team

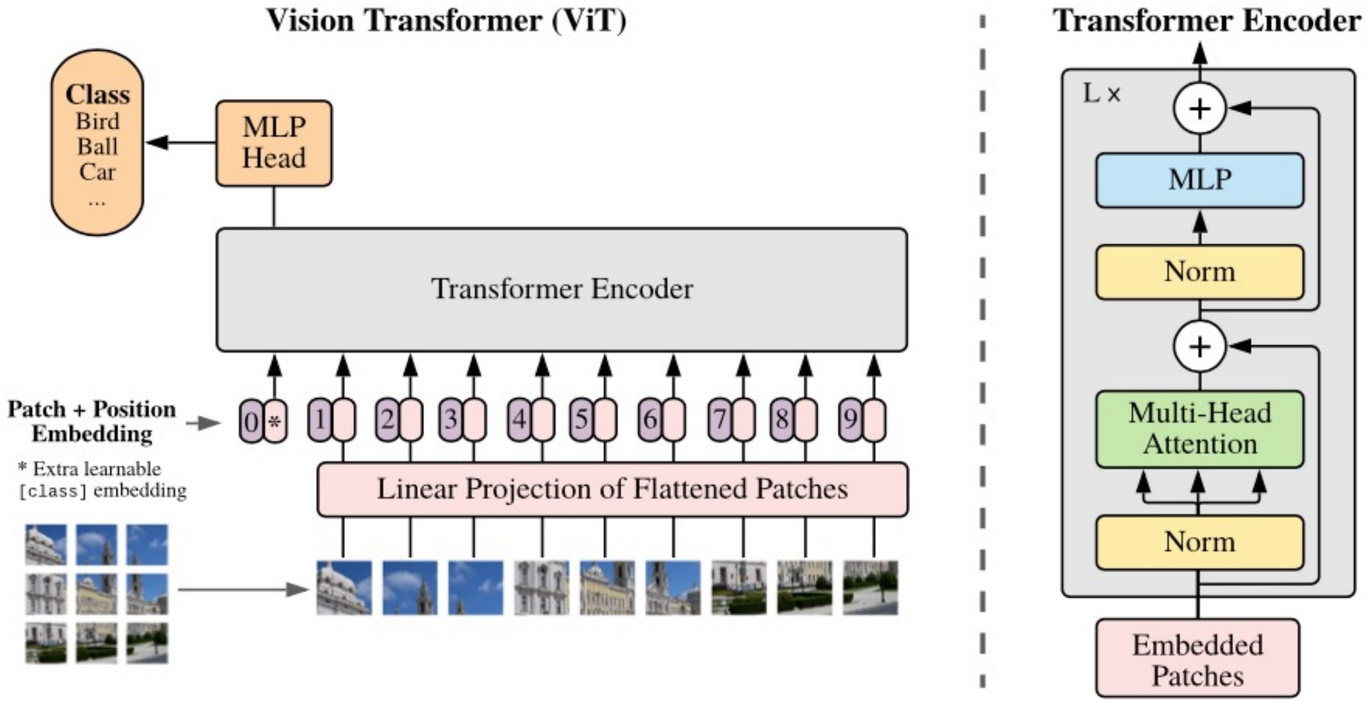
{adosovitskiy, neilhoulsby}@google.com

ABSTRACT

While the Transformer architecture has become the de-facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks while keeping their overall structure in place. We show that this reliance on CNNs is not necessary and a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks. When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.), Vision Transformer (ViT) attains excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train.[1]

1 INTRODUCTION

Self-attention-based architectures, in particular Transformers (Vaswani et al., 2017), have become the model of choice in natural language processing (NLP). The dominant approach is to pre-train on a large text corpus and then fine-tune on a smaller task-specific dataset (Devlin et al., 2019). Thanks to Transformers' computational efficiency and scalability, it has become possible to train models of unprecedented size, with over 100B parameters (Brown et al., 2020; Lepikhin et al., 2020). With the models and datasets growing, there is still no sign of saturating performance.



Transformers

- 3 big ideas in transformers
  - Tokens
  - (self) Attention
  - Positional embedding
- Why use a transformer over a CNN?
- Examples of architectures and applications

What is so special about transformers?

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

Abstract

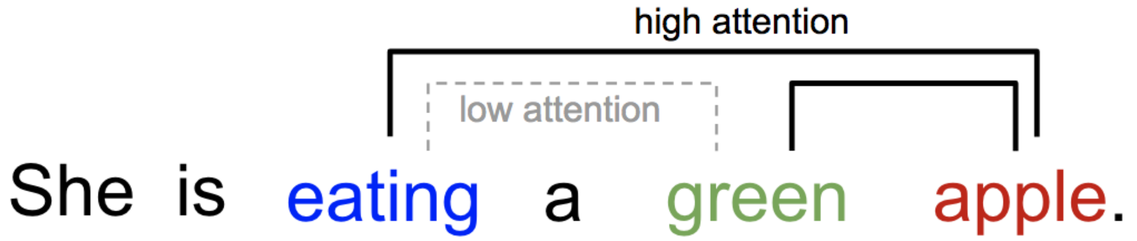
The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

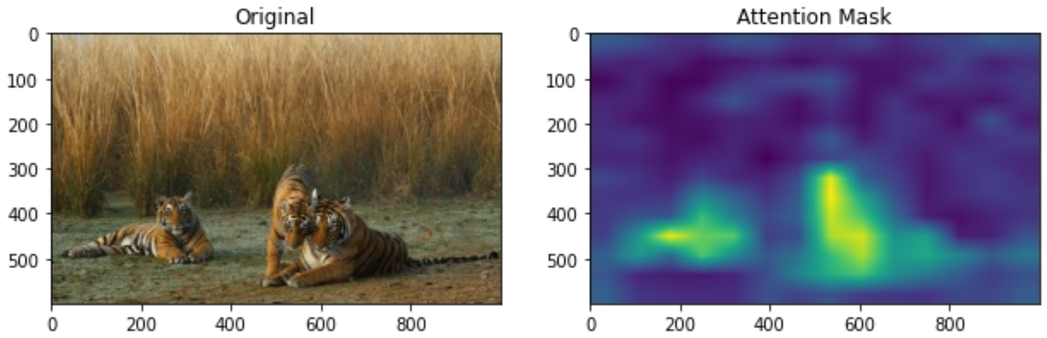
When people think transformers, they think **attention**

We want to focus on the parts of the input data **that are relevant for the current task**

Forget all your NLP knowledge

Sentences have ***sequential structure*** while images have ***spatial structure***





*Lisa does not fully agree with this

What is so special about transformers?

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

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

When people think transformers, they think **attention**

We want to focus on the parts of the input data **that are relevant for the current task**

what do we mean by “parts”?

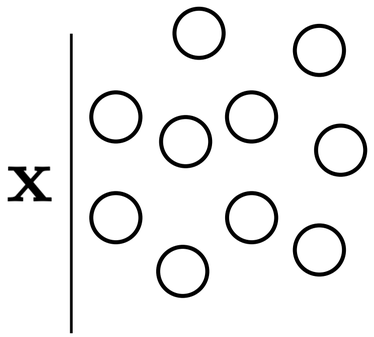
Idea #1: tokens

A new data structure: Tokens

- A **token** is an encapsulated bundle of information; with transformers we will operate over tokens rather than over neurons

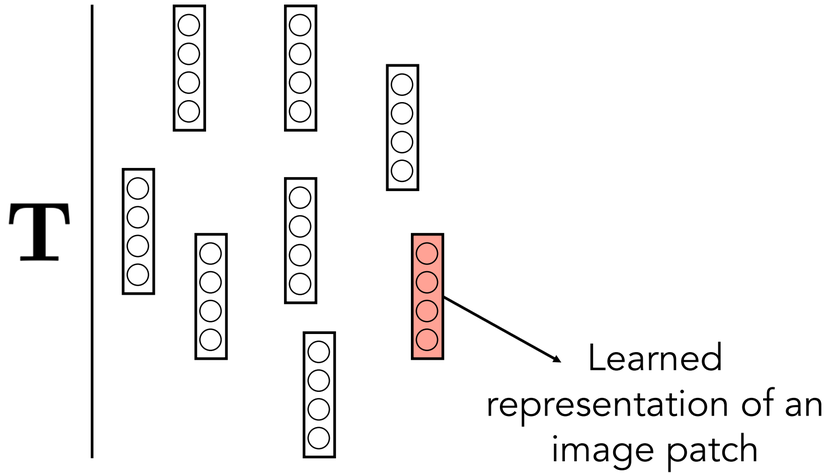
- *Tokens are to transformers as neurons are to neural nets*

set of **neurons**

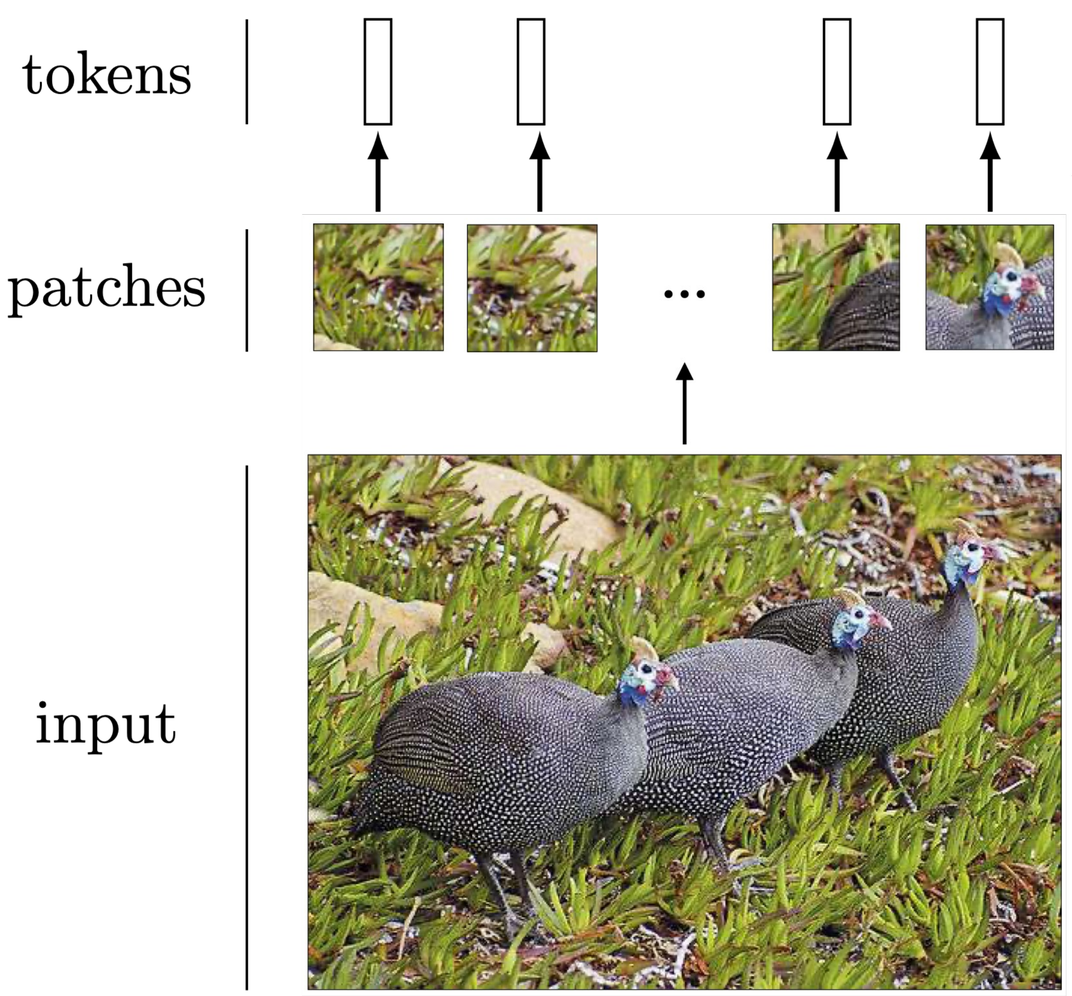


Channels, tensors, batches

set of **tokens**



Tokenizing the input data

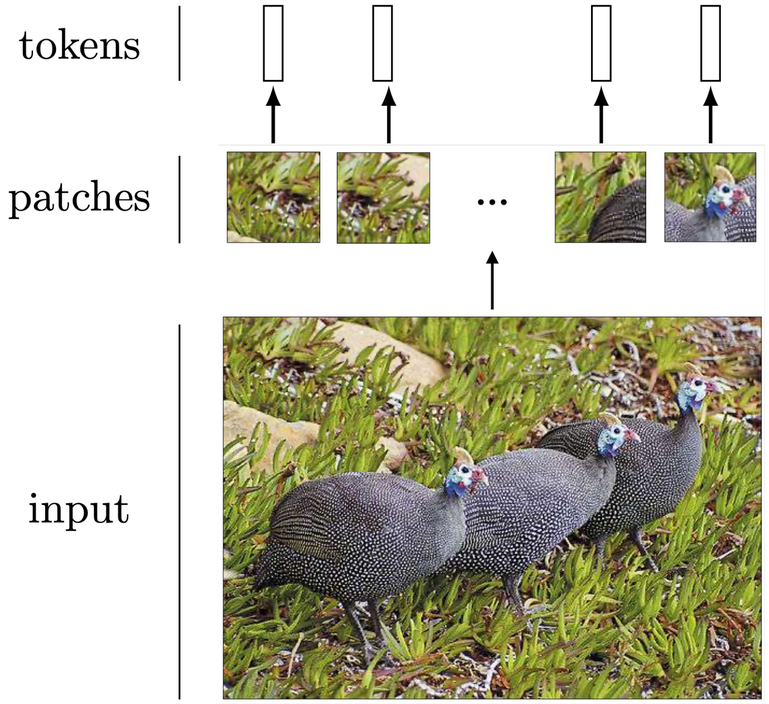


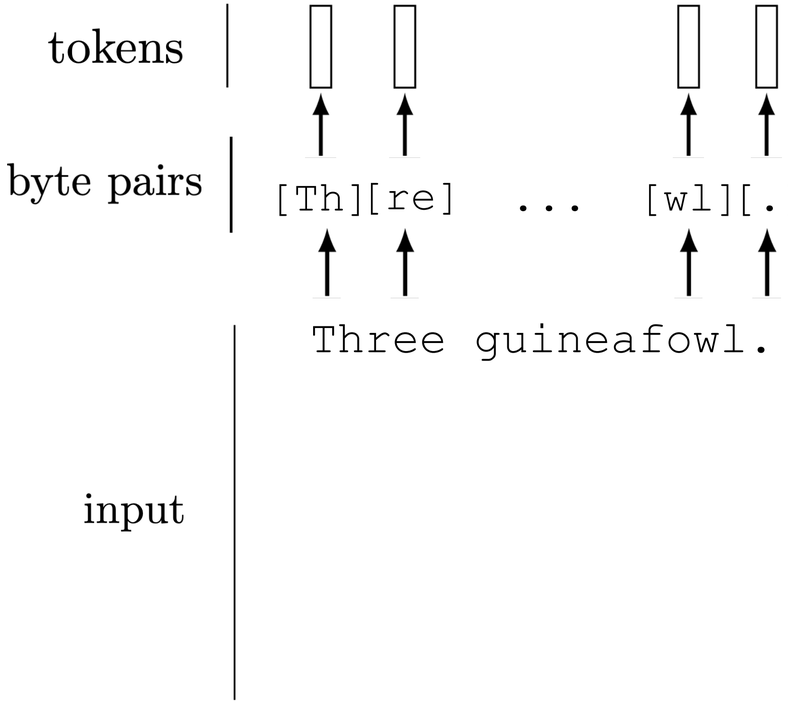
- When operating over *neurons*, we represent the input as an array of scalar-valued measurements (e.g., pixels)

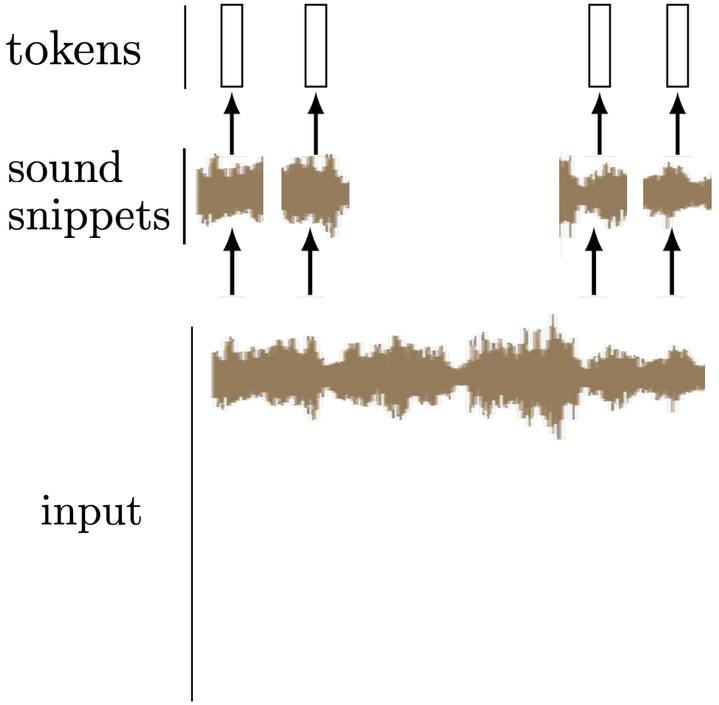
- When operating over *tokens*, we represent the input as an array of vector-valued measurements

Tokenizing the input data

You can tokenize anything.
General strategy: chop the input up into chunks, project each chunk to a vector.

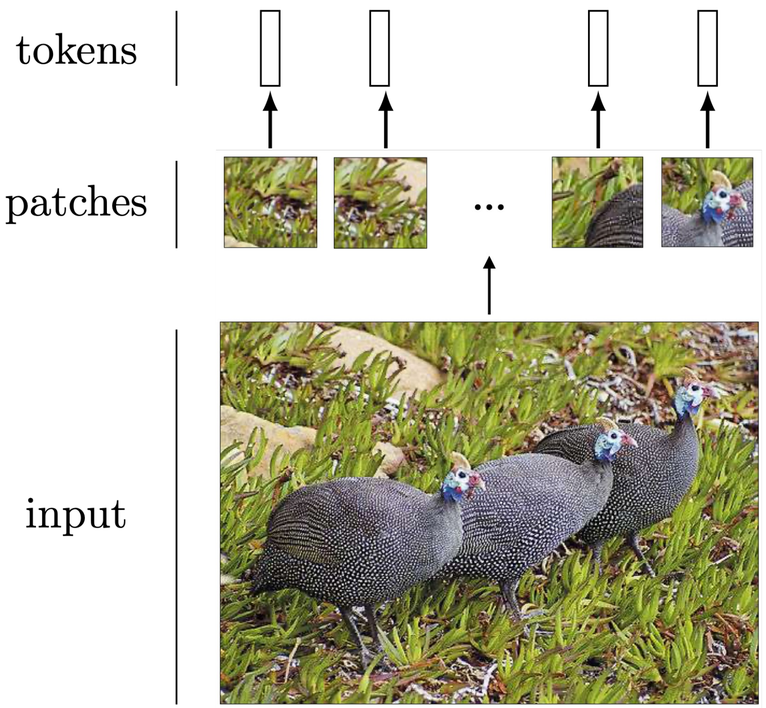


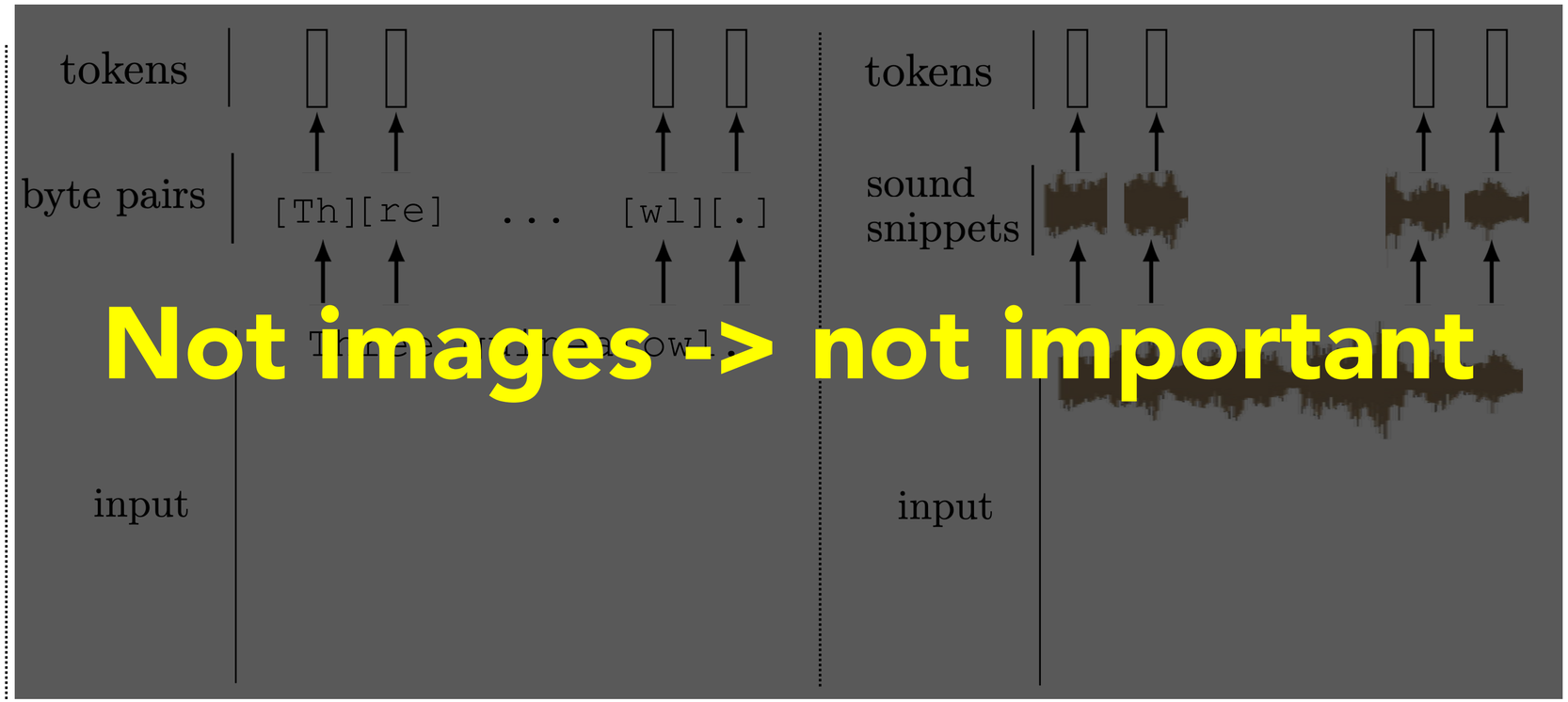




Tokenizing the input data

You can tokenize anything.
General strategy: chop the input up into chunks, project each chunk to a vector.





Operating on Tokens

Transformers consist of two main operations:

- *Mixing* tokens via a weighted sum (attention)
- *Modifying* each token via nonlinear transformations

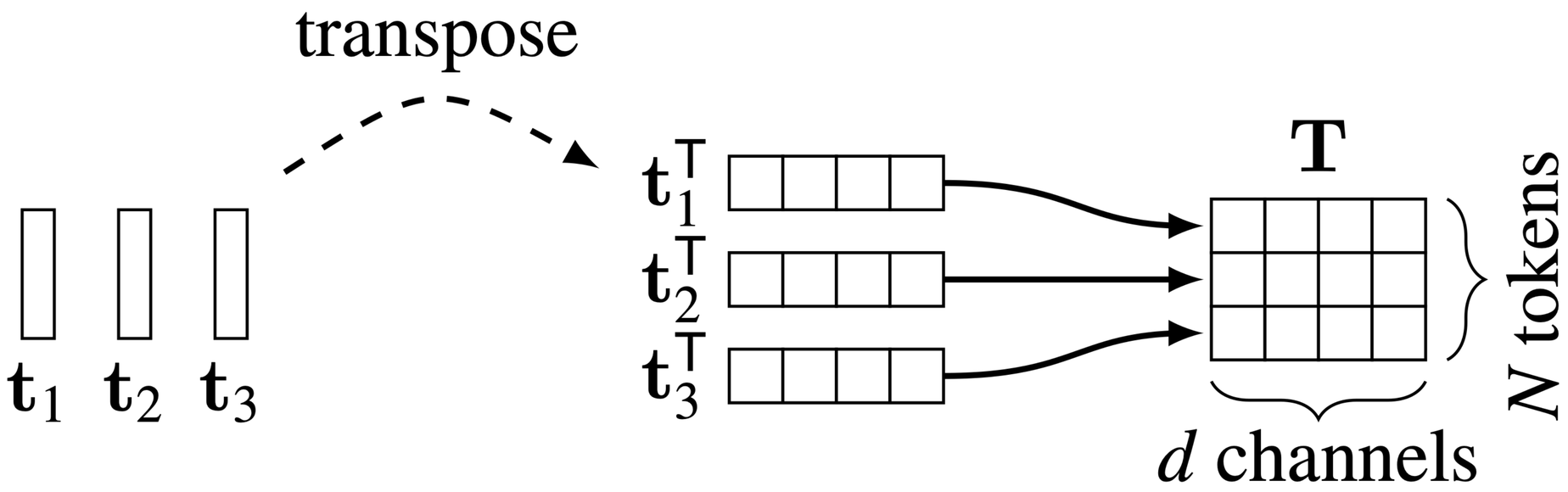
Fully connected layer

Pointwise nonlinearity (e.g. relu)

Look familiar?

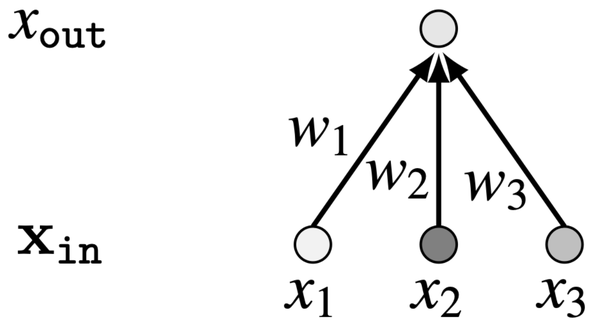
Analogous to the operations of neural nets!

Notation


transpose
$\mathbf{t}_1$ $\mathbf{t}_2$ $\mathbf{t}_3$
$\mathbf{t}_1^\mathsf{T}$
$\mathbf{t}_2^\mathsf{T}$
$\mathbf{t}_3^\mathsf{T}$
$\mathbf{T}$
N tokens
d channels

Linear combination of tokens (attention)

Linear combination of neurons

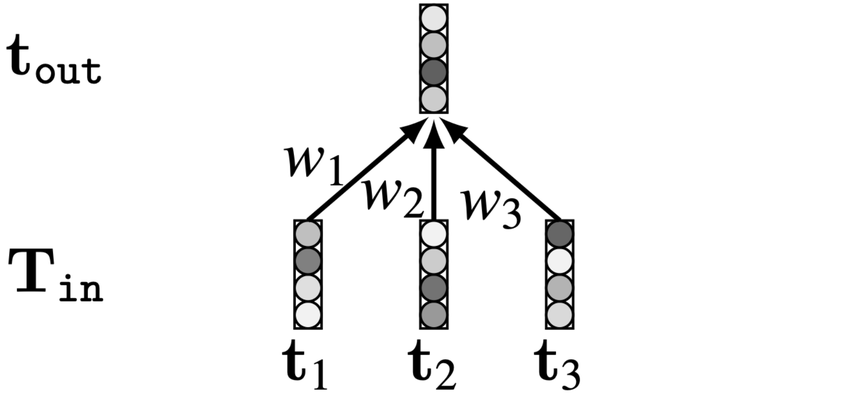


$$x_{\text{out}} = w_1 x_1 + w_2 x_2 + w_3 x_3$$

$$x_{\text{out}}[i] = \sum_{j=1}^{N} w_{ij} x_{\text{in}}[j]$$

$$\mathbf{x}_{\text{out}} = \mathbf{W}\mathbf{x}_{\text{in}}$$

Linear combination of tokens



$$\mathbf{t}_{\text{out}} = w_1 \mathbf{t}_1 + w_2 \mathbf{t}_2 + w_3 \mathbf{t}_3$$

$$\mathbf{T}_{\text{out}}[i,:] = \sum_{j=1}^{N} w_{ij} \mathbf{T}_{\text{in}}[j,:]$$

$$\mathbf{T}_{\text{out}} = \mathbf{W}\mathbf{T}_{\text{in}}$$

Token-wise nonlinearity

$$\mathbf{x}_{\texttt{out}} = \begin{bmatrix} \texttt{relu}(x_{\texttt{in}}[0]) \\ \vdots \\ \texttt{relu}(x_{\texttt{in}}[N-1]) \end{bmatrix}$$

$$\mathbf{T}_{\texttt{out}} = \begin{bmatrix} F_\theta(\mathbf{T}_{\texttt{in}}[0,:]) \\ \vdots \\ F_\theta(\mathbf{T}_{\texttt{in}}[N-1,:]) \end{bmatrix}$$

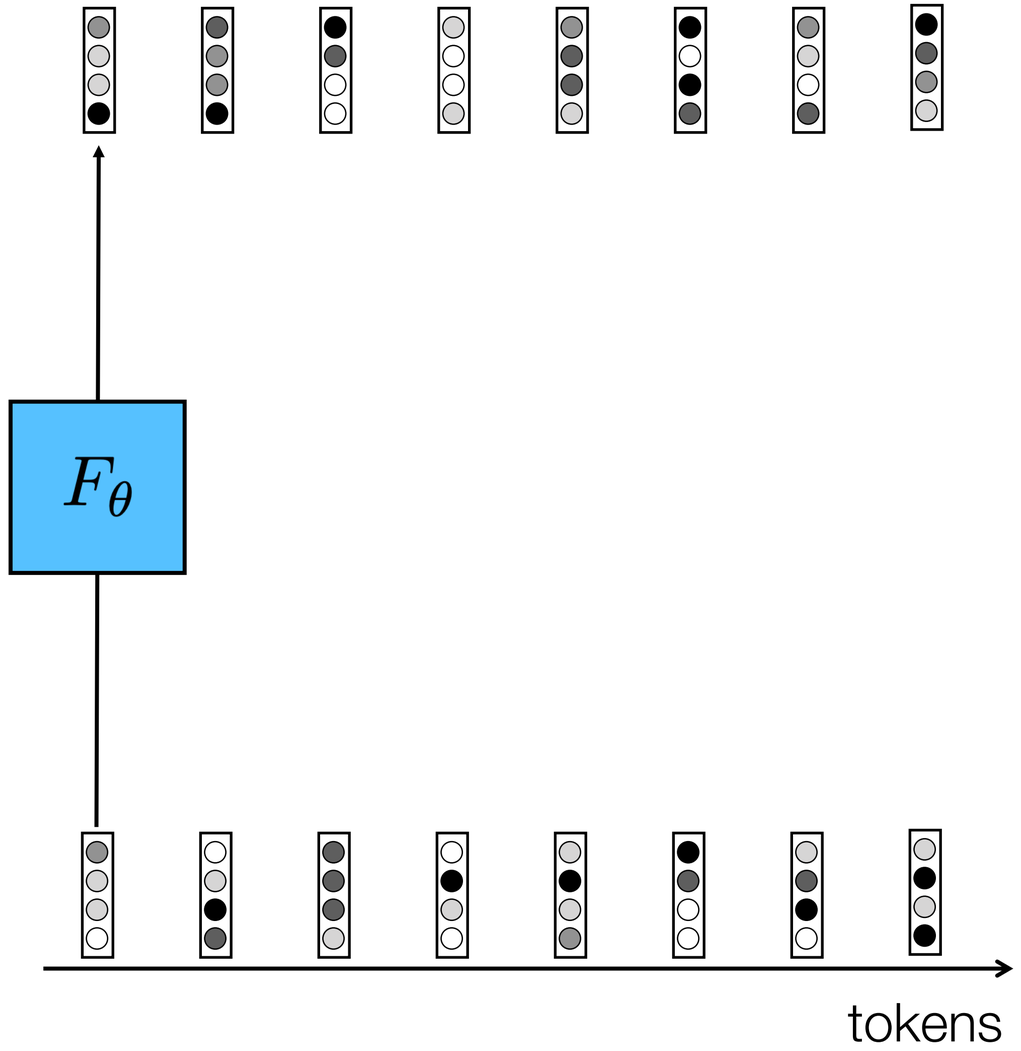
F is typically an MLP

Equivalent to a CNN with 1x1 kernels run over token sequence

Token-wise nonlinearity

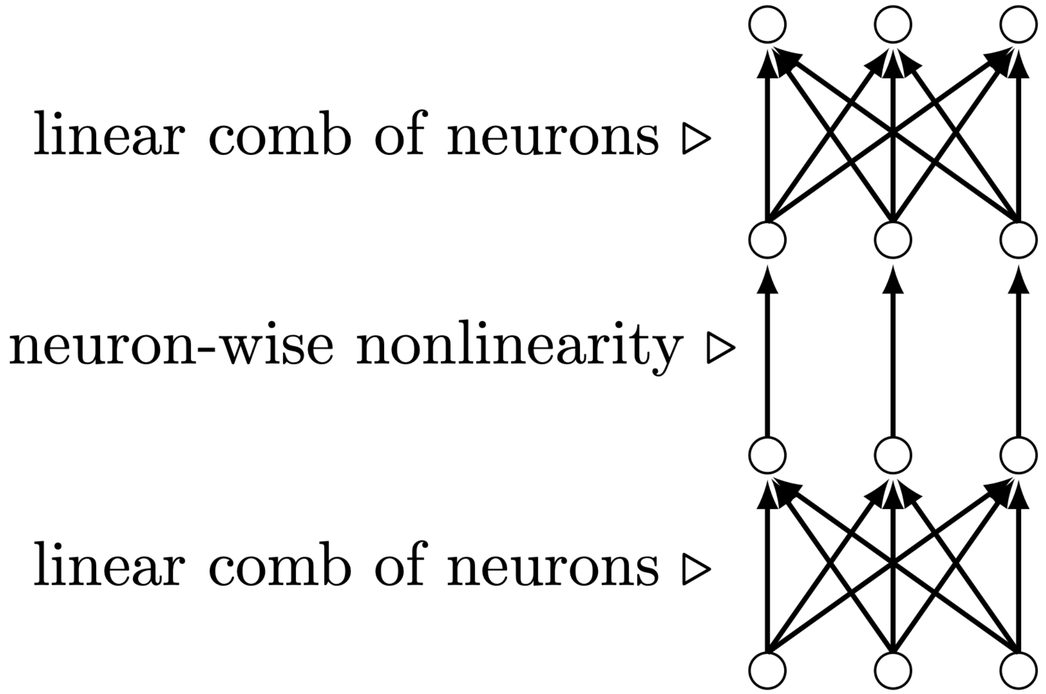
$$\mathbf{x}_{\texttt{out}} = \begin{bmatrix} \texttt{relu}(x_{\texttt{in}}[0]) \\ \vdots \\ \texttt{relu}(x_{\texttt{in}}[N-1]) \end{bmatrix}$$

$$\mathbf{T}_{\texttt{out}} = \begin{bmatrix} F_\theta(\mathbf{T}_{\texttt{in}}[0,:]) \\ \vdots \\ F_\theta(\mathbf{T}_{\texttt{in}}[N-1,:]) \end{bmatrix}$$



Token nets

Neural net



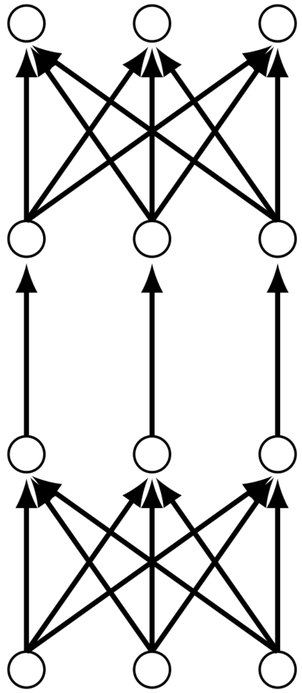
Token nets

Neural net

linear comb of neurons ▷

neuron-wise nonlinearity ▷

linear comb of neurons ▷

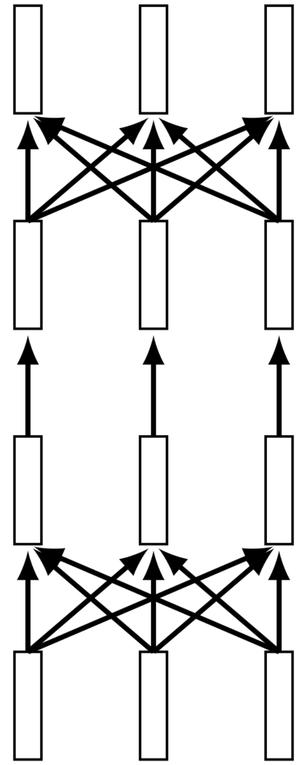


Token net

linear comb of tokens ▷

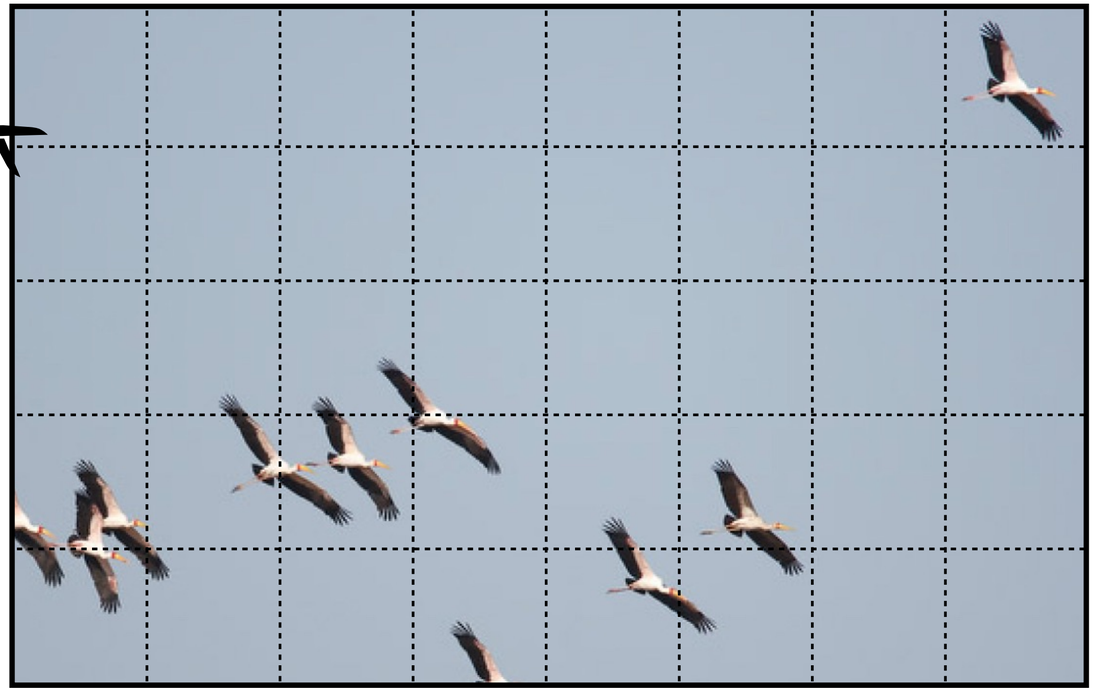
token-wise nonlinearity ▷

linear comb of tokens ▷



Idea #2: attention

A limitation of CNNs

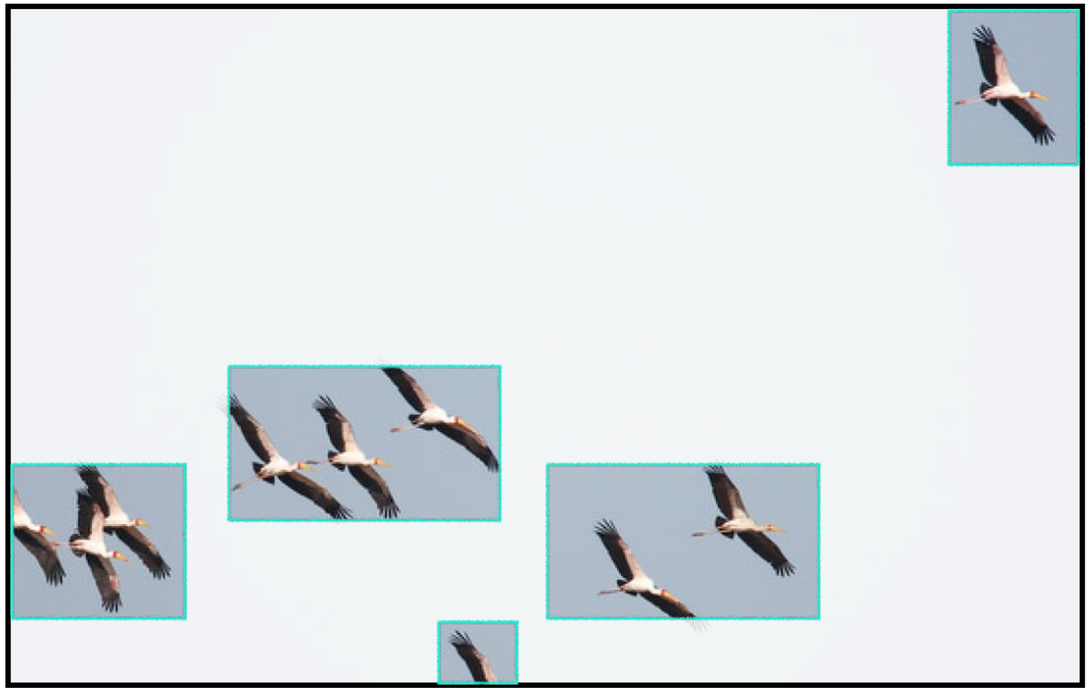


How many birds are in this image?

Is the top right bird the same species as the bottom left bird?

- CNNs are built around the idea of locality: different local regions of an image can be processed independently
- not well-suited to modeling long distance relationships

What is attention?



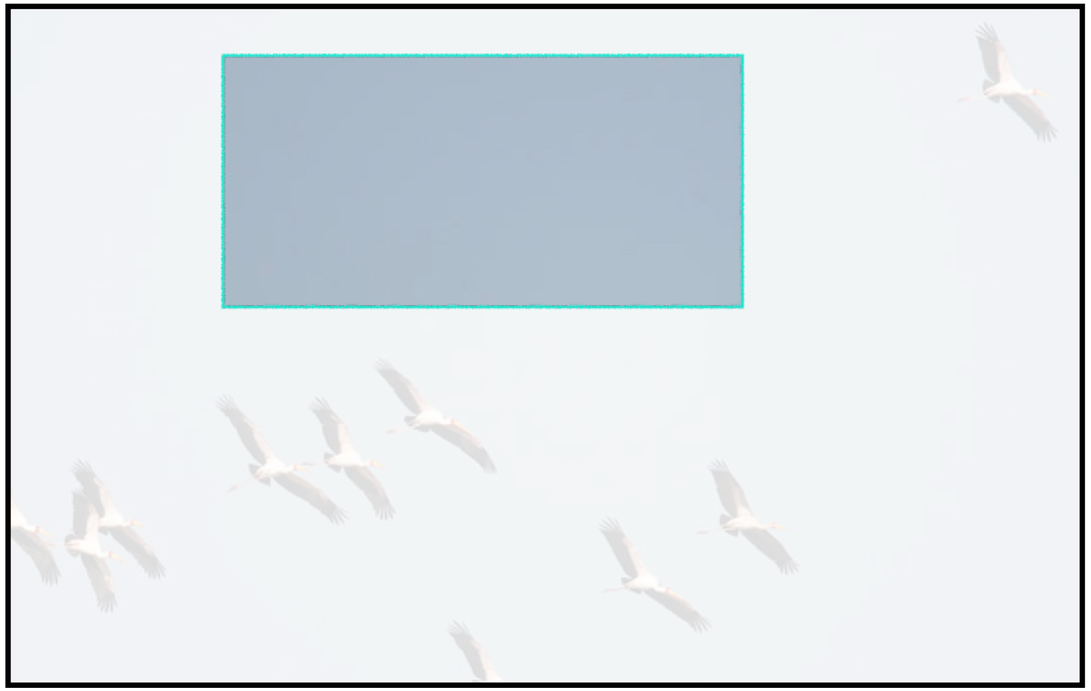
How many birds are in this image?

What is attention?

Is the top right bird the same species as the bottom left bird?

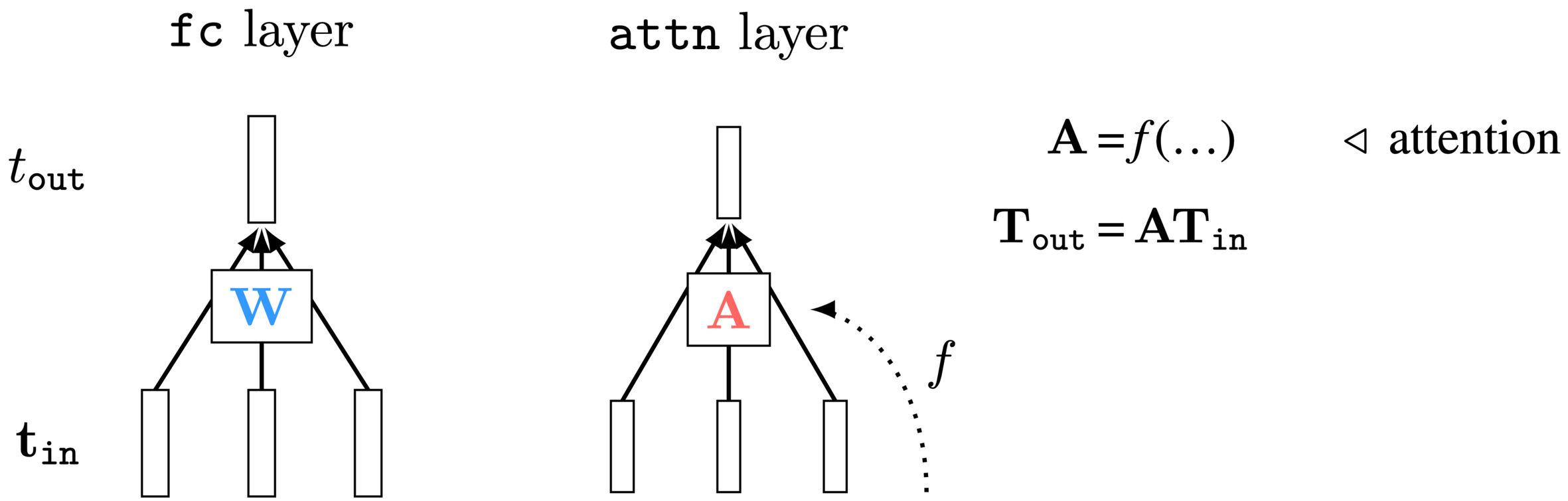
Humans do this to! We move your eyes around to the areas relevant to the question

What is attention?



What’s the color of the sky?

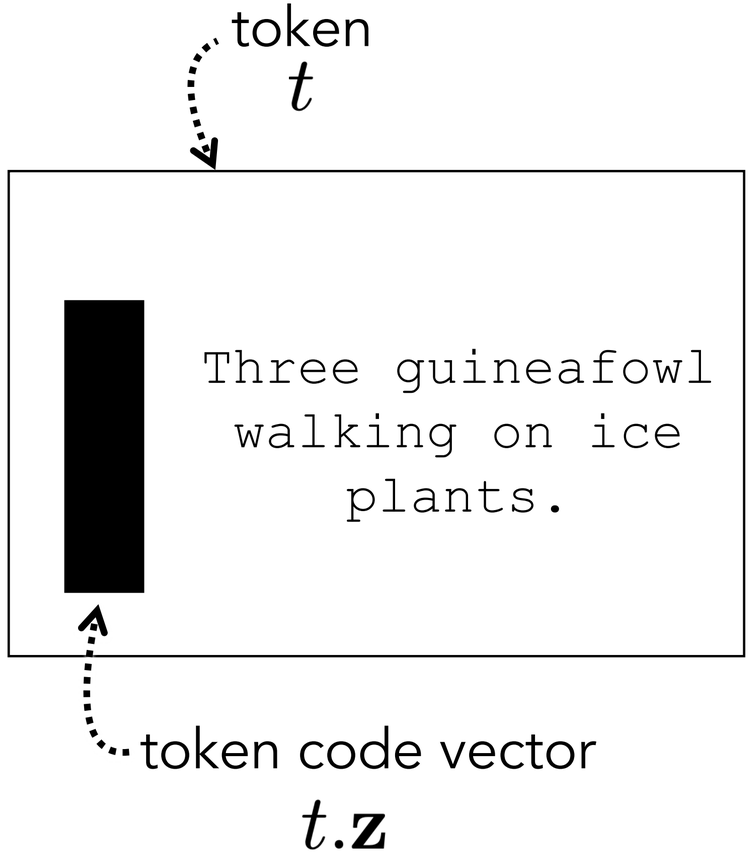
Attention allows us to process global information efficiently



W is free *parameters*.

A is a function of some input *data*. The data tells us which tokens to attend to (assign high weight in weighted sum)

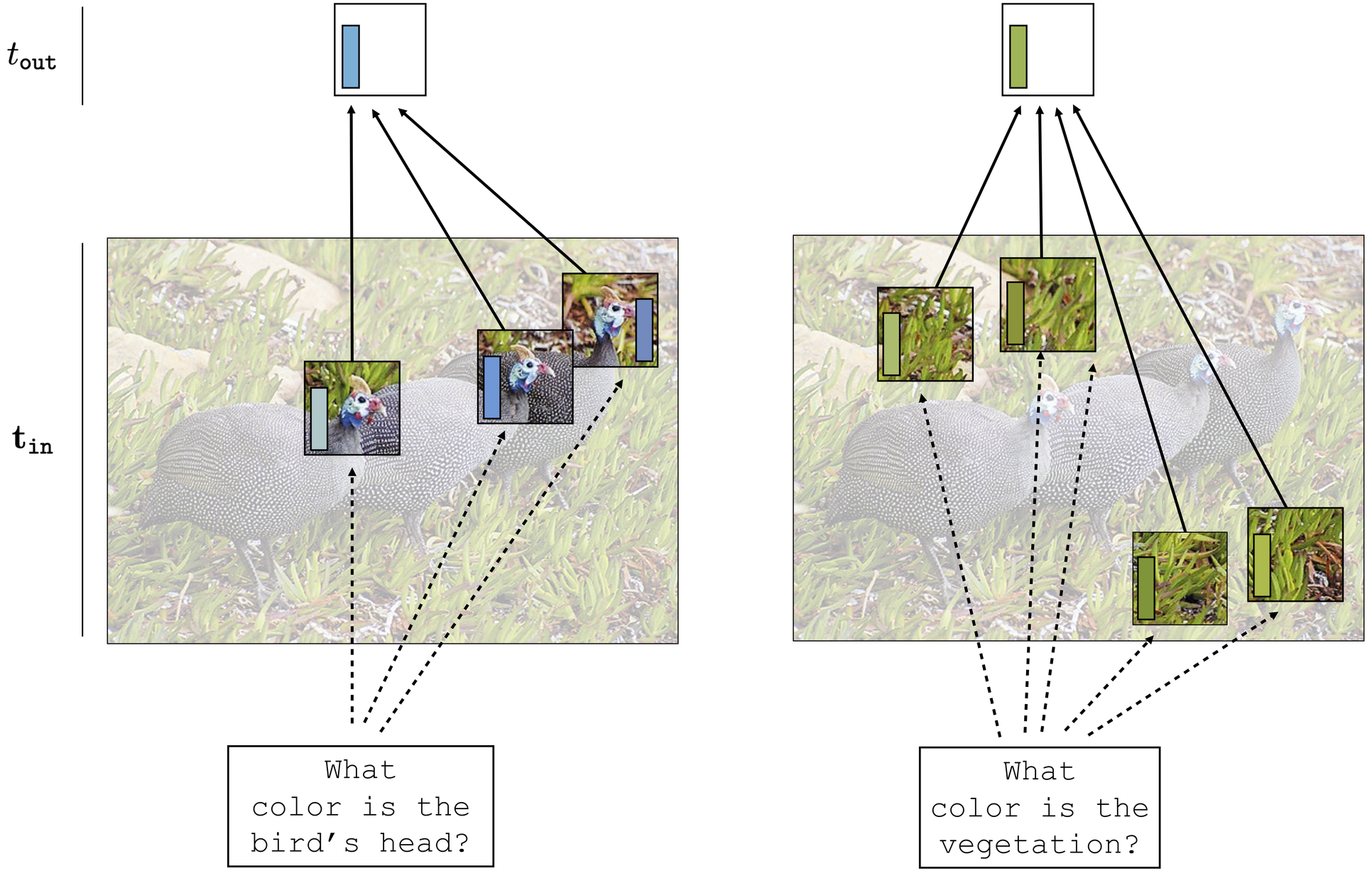
Notation reminder



t_{out}

t_{in}

What color is the bird's head?


t_{out}
t_{in}
What color is the bird's head?
What color is the vegetation?

Attention via Queries, Keys, and Values

- In databases, a database cell holds a value, which is retrieved when a query matches a cells key
- In transformer attention, each token is associated with a query vector, key vector, and value vector
- The similarity between a query and a key determines the amount of attention weight the query will apply to the token with that key

$$\mathbf{q} = \mathbf{W}_q \mathbf{t} \quad \triangleleft \ \mathbf{query}$$

$$\mathbf{k} = \mathbf{W}_k \mathbf{t} \quad \triangleleft \ \mathbf{key}$$

$$\mathbf{v} = \mathbf{W}_v \mathbf{t} \quad \triangleleft \ \mathbf{value}$$

Attention via Queries, Keys, and Values

- In databases, a database cell holds a value, which is retrieved when a query matches a cells key
- In transformer attention, each token is associated with a query vector, key vector, and value vector
- The similarity between a query and a key determines the amount of attention weight the query will apply to the token with that key

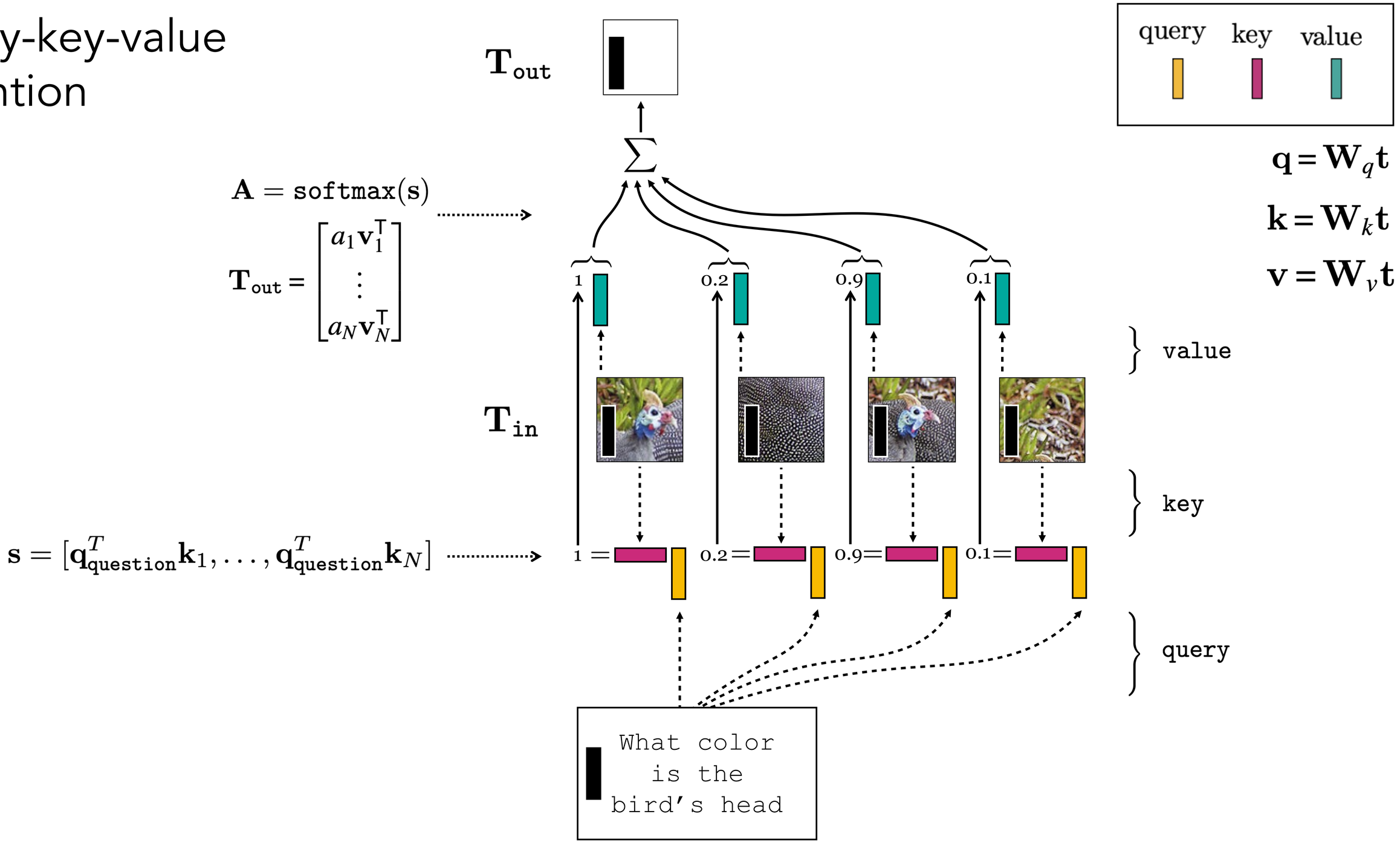
$$\mathbf{q} = \mathbf{W}_q \mathbf{t} \quad \triangleleft \text{ query}$$

$$\mathbf{k} = \mathbf{W}_k \mathbf{t} \quad \triangleleft \text{ key}$$

$$\mathbf{v} = \mathbf{W}_v \mathbf{t} \quad \triangleleft \text{ value}$$

The weight matrices define the notion of similarity

query-key-value attention

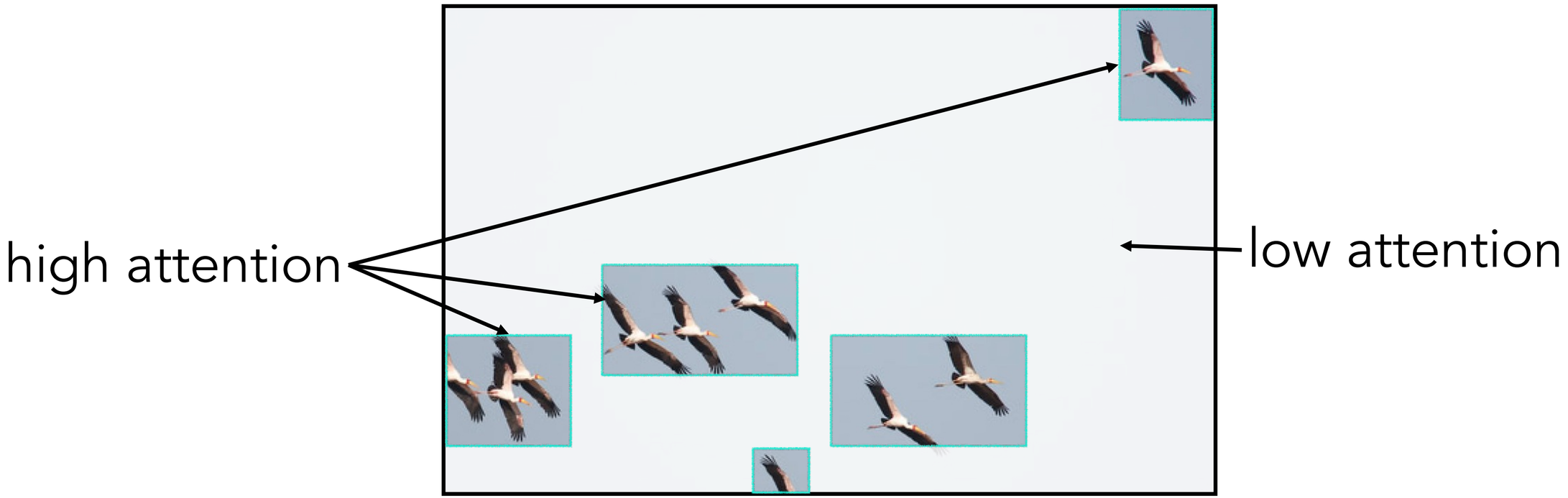


Guess the attention

How many birds are in this image?

Guess the attention

How many birds are in this image?

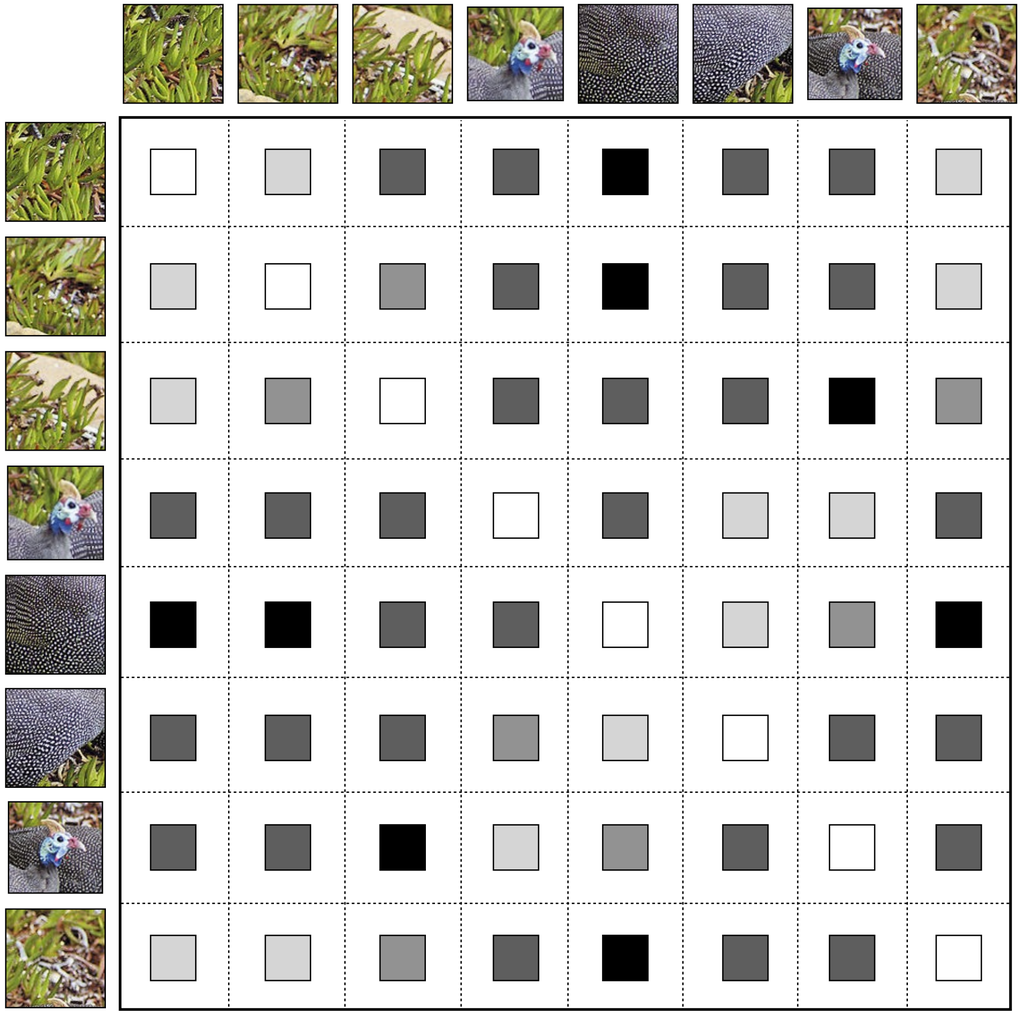


Attention VS Self-Attention

- Our VQA example uses attention but vision transformers use a more general architecture: **self-attention**
  - Instead of using a **question** as a query, each **image token** is a query
- In **self-attention**, all tokens submit queries and for each of these queries, we take a weighted sim over all tokens in that layer

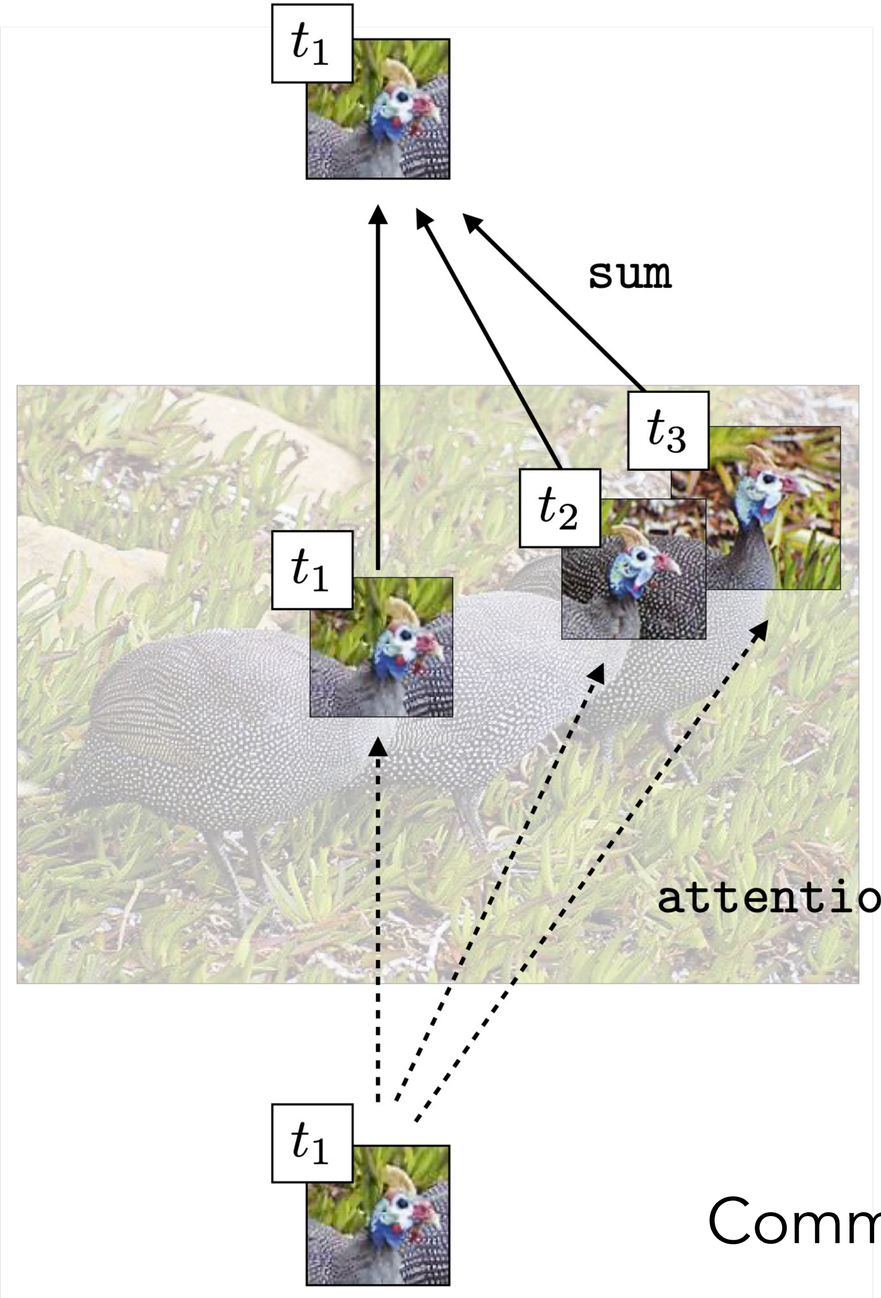
Self-attention

Example of attention if query() and key() are the identity function



—> just a Gram matrix (similarity matrix) over tokens!
Essentially: clusters similar tokens

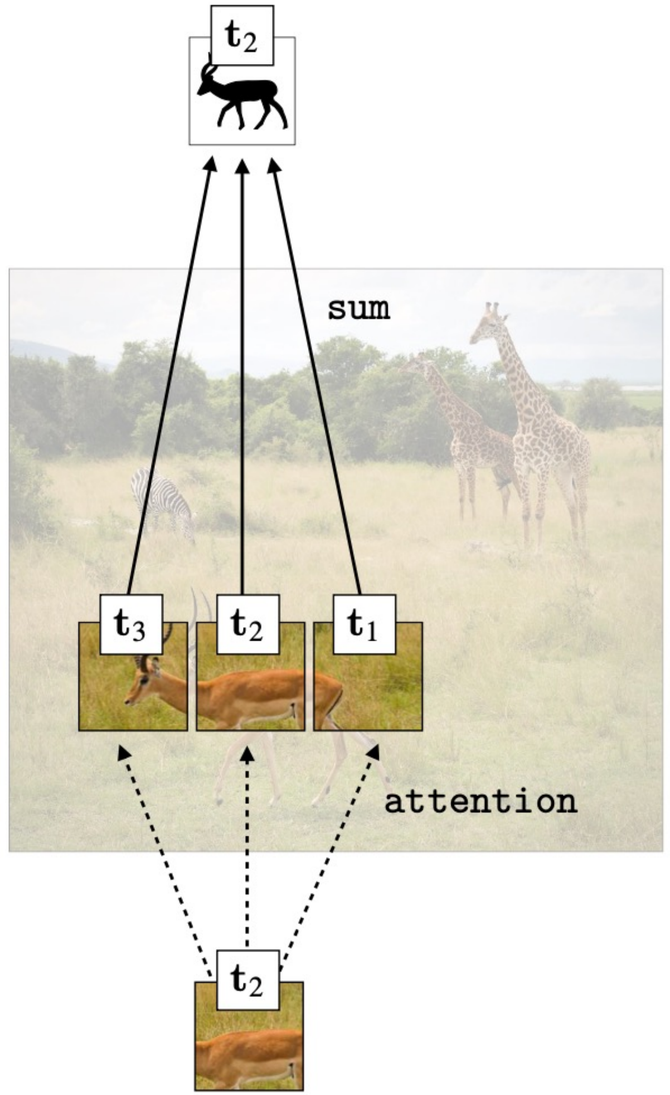
Self-attention



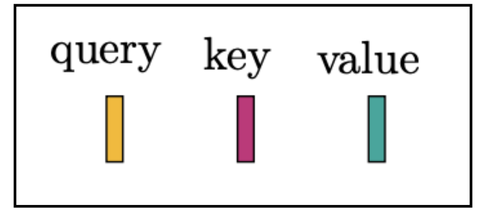
Common to learn visual similarity

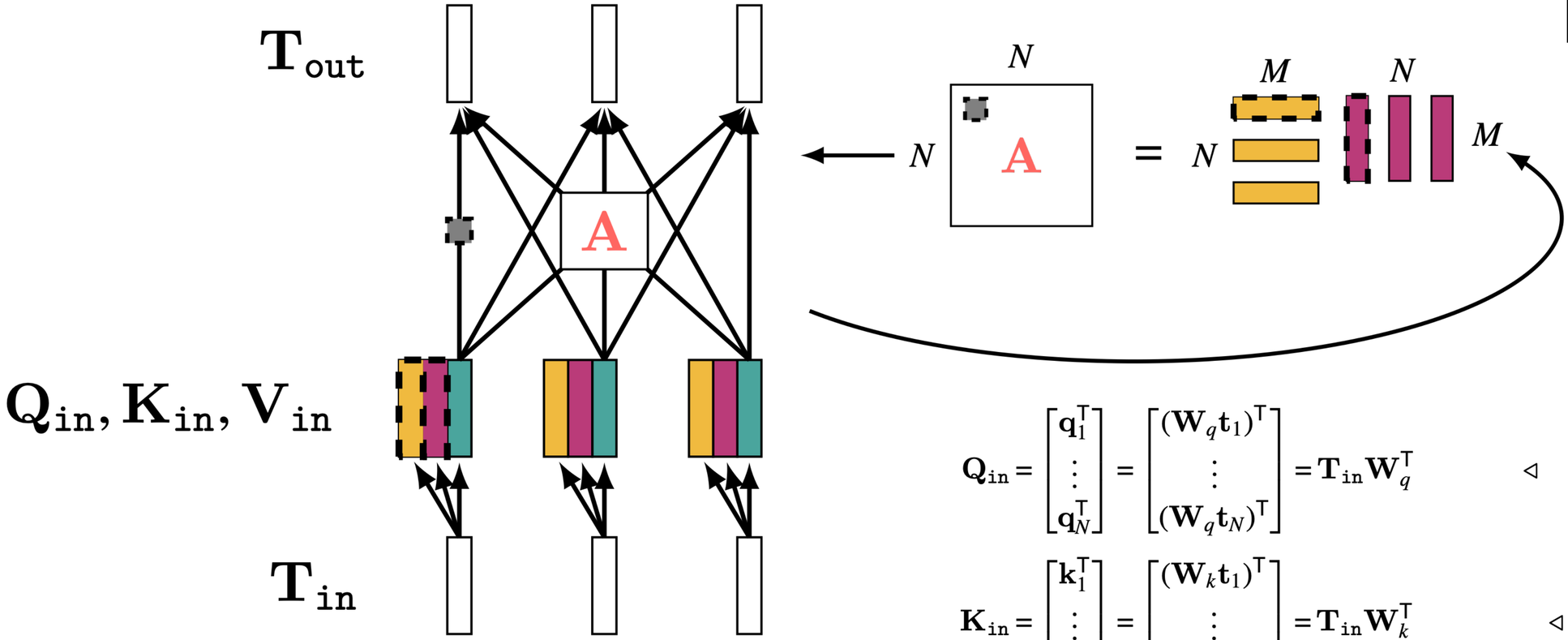
Self-attention – aggregate object information

Attention could aggregate information across all patches containing the same object, resulting in better representation of the object in the query patch.



`self attn` layer (expanded)





$$\mathbf{Q}_{\texttt{in}} = \begin{bmatrix} \mathbf{q}_1^\mathsf{T} \\ \vdots \\ \mathbf{q}_N^\mathsf{T} \end{bmatrix} = \begin{bmatrix} (\mathbf{W}_q \mathbf{t}_1)^\mathsf{T} \\ \vdots \\ (\mathbf{W}_q \mathbf{t}_N)^\mathsf{T} \end{bmatrix} = \mathbf{T}_{\texttt{in}} \mathbf{W}_q^\mathsf{T} \quad \triangleleft \quad \text{query matrix}$$

$$\mathbf{K}_{\texttt{in}} = \begin{bmatrix} \mathbf{k}_1^\mathsf{T} \\ \vdots \\ \mathbf{k}_N^\mathsf{T} \end{bmatrix} = \begin{bmatrix} (\mathbf{W}_k \mathbf{t}_1)^\mathsf{T} \\ \vdots \\ (\mathbf{W}_k \mathbf{t}_N)^\mathsf{T} \end{bmatrix} = \mathbf{T}_{\texttt{in}} \mathbf{W}_k^\mathsf{T} \quad \triangleleft \quad \text{key matrix}$$

$$\mathbf{V}_{\texttt{in}} = \begin{bmatrix} \mathbf{v}_1^\mathsf{T} \\ \vdots \\ \mathbf{v}_N^\mathsf{T} \end{bmatrix} = \begin{bmatrix} (\mathbf{W}_v \mathbf{t}_1)^\mathsf{T} \\ \vdots \\ (\mathbf{W}_v \mathbf{t}_N)^\mathsf{T} \end{bmatrix} = \mathbf{T}_{\texttt{in}} \mathbf{W}_v^\mathsf{T} \quad \triangleleft \quad \text{value matrix}$$

$$\mathbf{A} = f(\mathbf{T}_{\texttt{in}}) = \texttt{softmax}\left(\frac{\mathbf{Q}_{\texttt{in}} \mathbf{K}_{\texttt{in}}^\mathsf{T}}{\sqrt{m}}\right) \quad \triangleleft \quad \text{attention matrix}$$

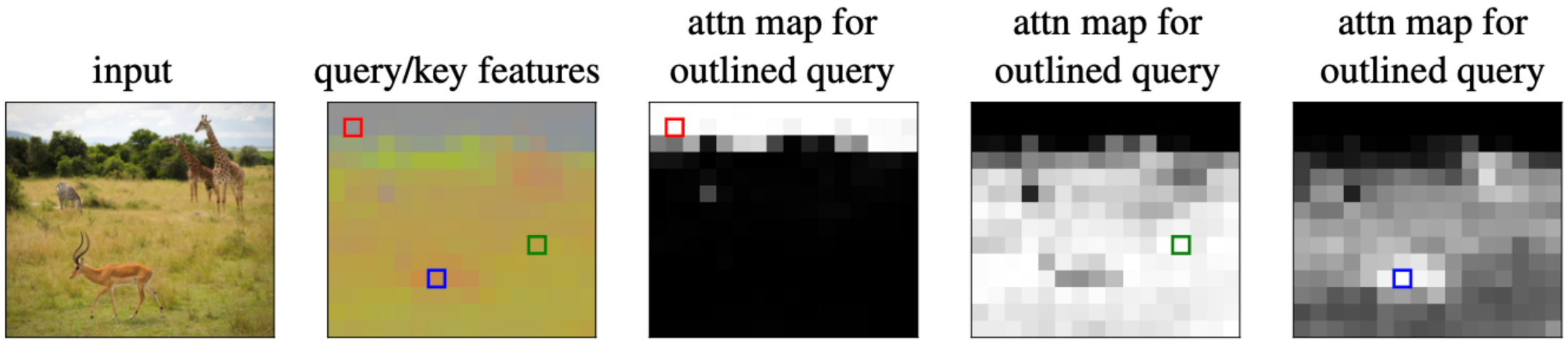
$$\mathbf{T}_{\texttt{out}} = \mathbf{A}\mathbf{V}_{\texttt{in}}$$

Attention maps in a trained transformer

["DINO", Caron et all. 2021]

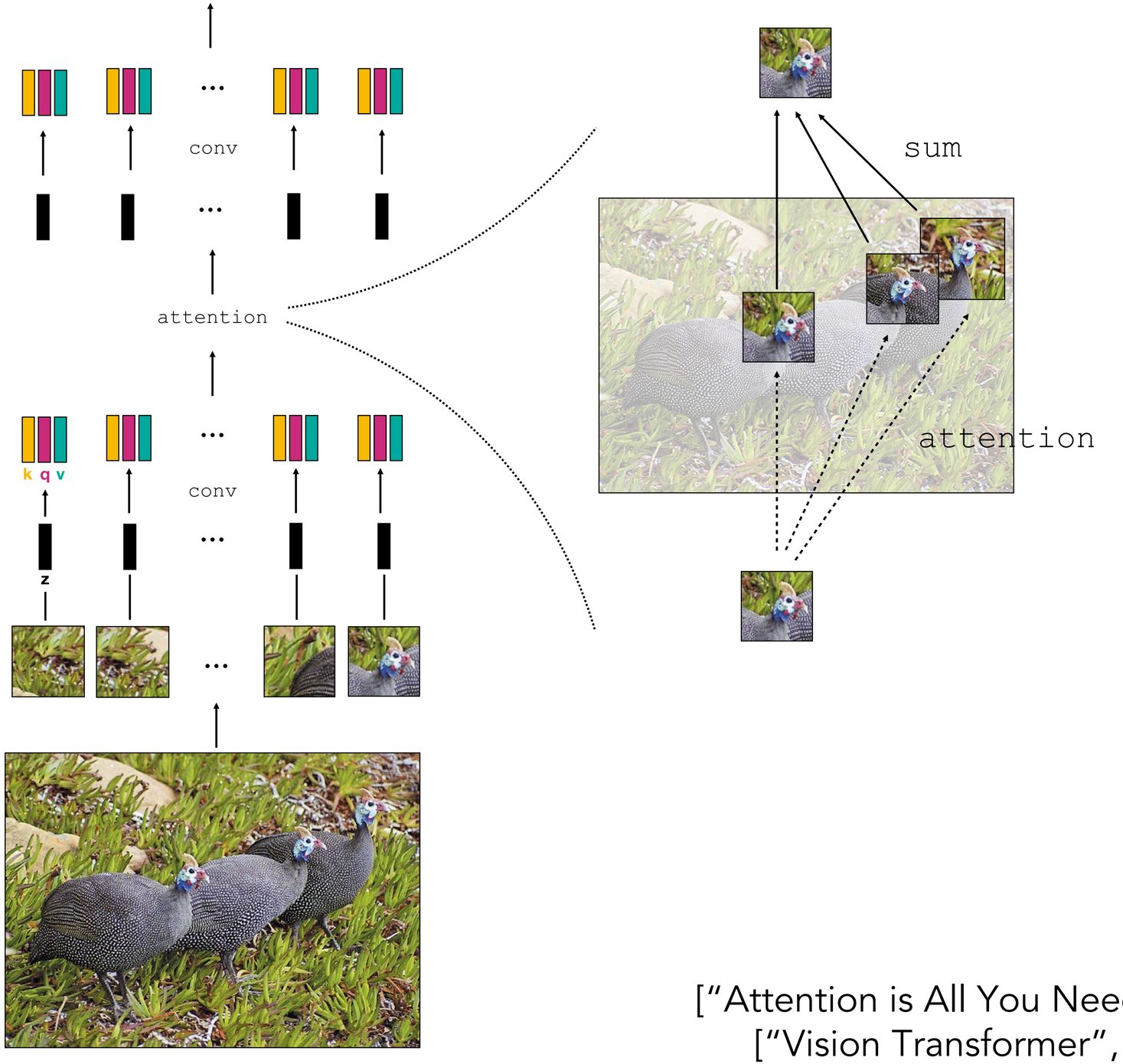
Self Attention Maps

Like how different questions result in needing to look at different areas of the image, self-attention maps are different for different image patches (query)



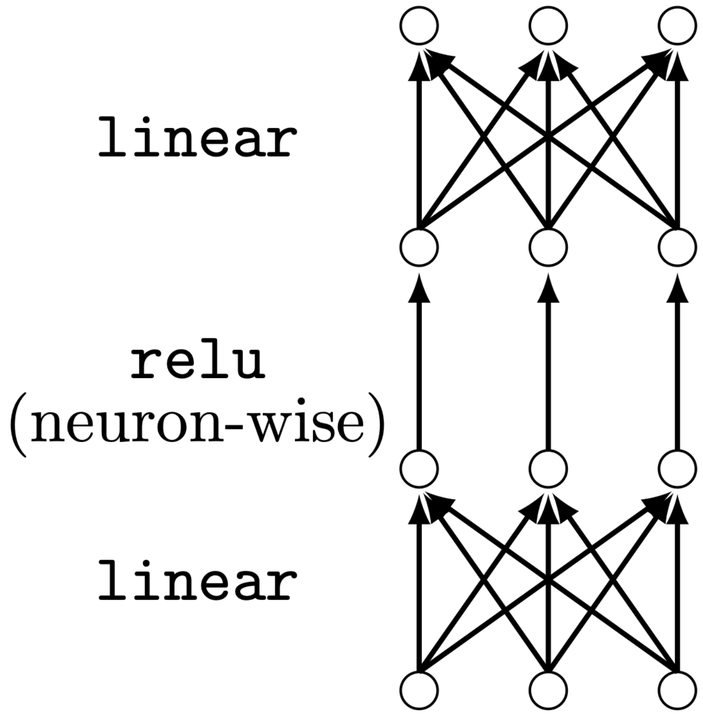
*simplified example, query key vectors are average color of patch

Transformer (simplified)

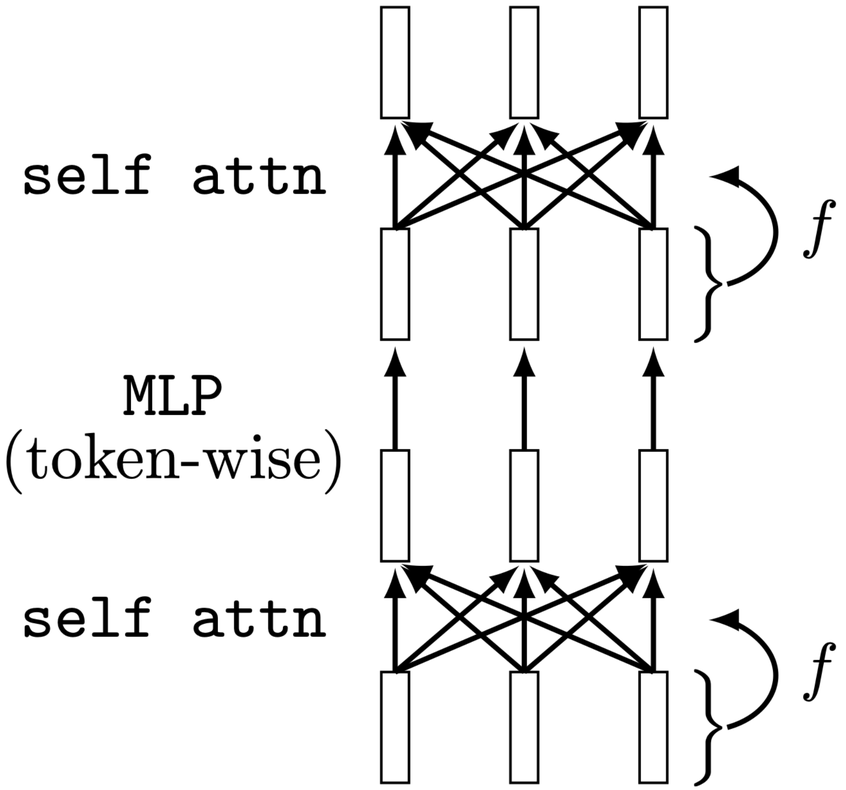


["Attention is All You Need", Vaswani et al. 2017]
["Vision Transformer", Dosovitskiy et al. 2020]

MLP

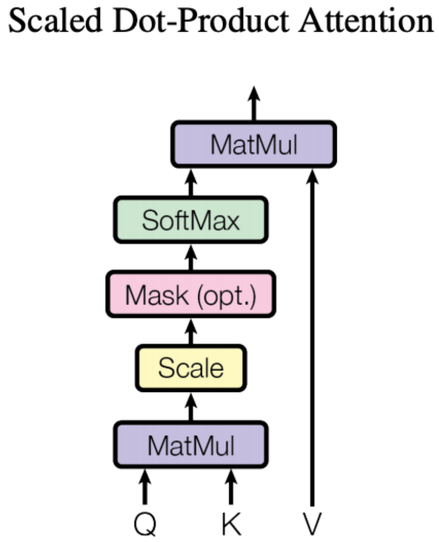


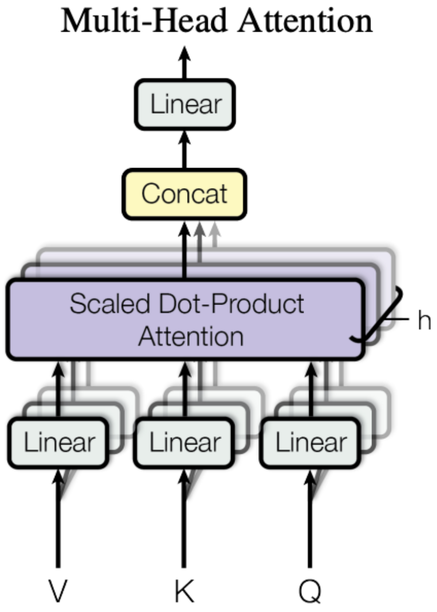
Transformer (vanilla)



Multihead self-attention (MSA)

- Rather than having just one way of attending, why not have k?
- Each gets its own parameterized query(), key(), value() functions
- Run them all in parallel, then (weighted) sum the output token code vectors





$$\mathbf{q} = \mathbf{W}_q \mathbf{t} \quad \triangleleft \text{ query}$$

$$\mathbf{k} = \mathbf{W}_k \mathbf{t} \quad \triangleleft \text{ key}$$

$$\mathbf{v} = \mathbf{W}_v \mathbf{t} \quad \triangleleft \text{ value}$$

The weight matrices define the notion of similarity

Putting it all together

Published as a conference paper at ICLR 2021

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Alexey Dosovitskiy*,†, Lucas Beyer*, Alexander Kolesnikov*, Dirk Weissenborn*, Xiaohua Zhai*, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby*,†

*equal technical contribution, †equal advising

Google Research, Brain Team

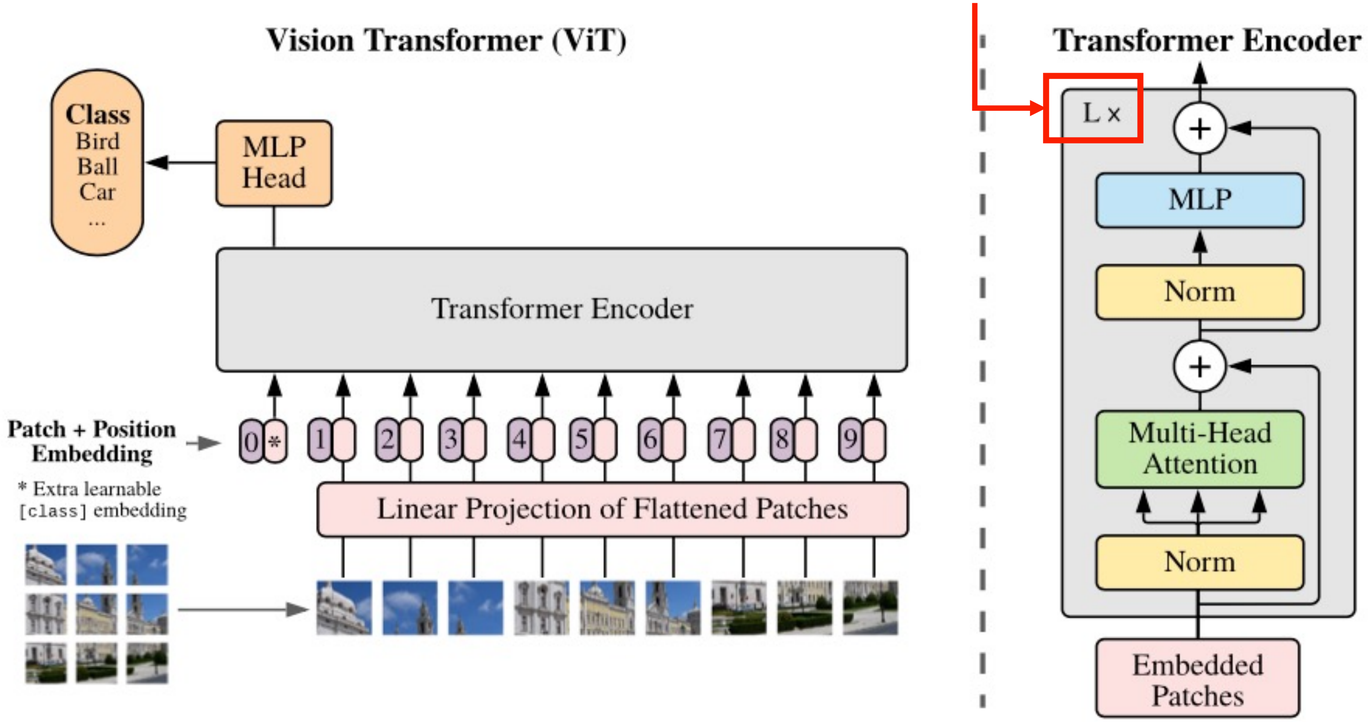
{adosovitskiy, neilhoulsby}@google.com

ABSTRACT

While the Transformer architecture has become the de-facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks while keeping their overall structure in place. We show that this reliance on CNNs is not necessary and a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks. When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.), Vision Transformer (ViT) attains excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train.[1]

1 INTRODUCTION

Self-attention-based architectures, in particular Transformers (Vaswani et al., 2017), have become the model of choice in natural language processing (NLP). The dominant approach is to pre-train on a large text corpus and then fine-tune on a smaller task-specific dataset (Devlin et al., 2019). Thanks to Transformers' computational efficiency and scalability, it has become possible to train models of unprecedented size, with over 100B parameters (Brown et al., 2020; Lepikhin et al., 2020). With the models and datasets growing, there is still no sign of saturating performance.



Putting it all together

Published as a conference paper at ICLR 2021

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Alexey Dosovitskiy*,†, Lucas Beyer*, Alexander Kolesnikov*, Dirk Weissenborn*, Xiaohua Zhai*, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby*,†

*equal technical contribution, †equal advising

Google Research, Brain Team

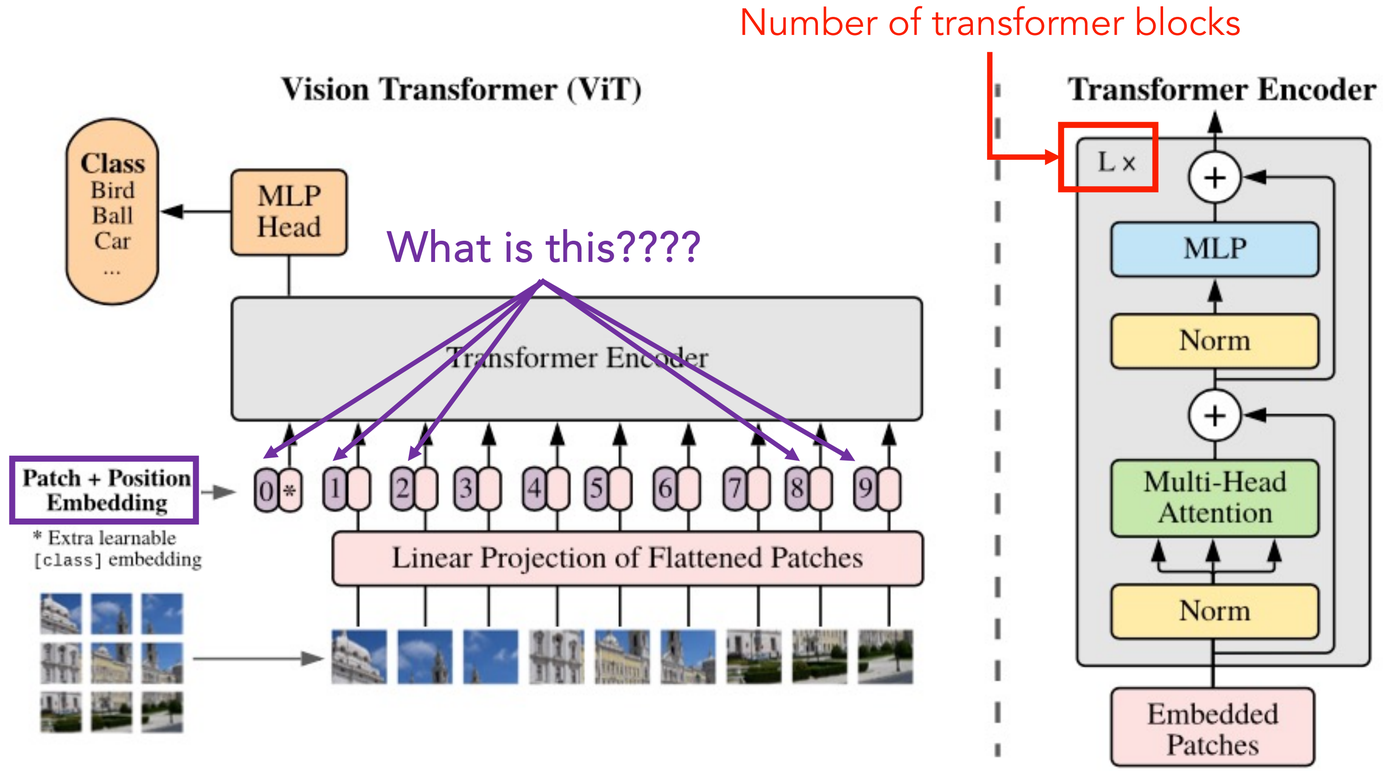
{adosovitskiy, neilhoulsby}@google.com

ABSTRACT

While the Transformer architecture has become the de-facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks while keeping their overall structure in place. We show that this reliance on CNNs is not necessary and a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks. When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.), Vision Transformer (ViT) attains excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train.[1]

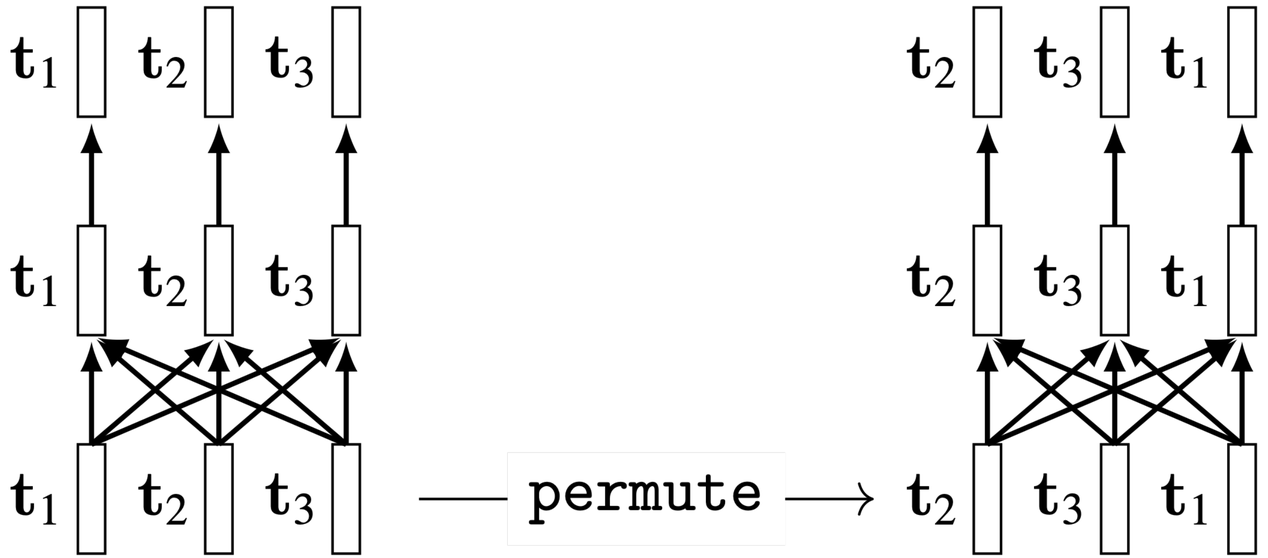
1 INTRODUCTION

Self-attention-based architectures, in particular Transformers (Vaswani et al., 2017), have become the model of choice in natural language processing (NLP). The dominant approach is to pre-train on a large text corpus and then fine-tune on a smaller task-specific dataset (Devlin et al., 2019). Thanks to Transformers' computational efficiency and scalability, it has become possible to train models of unprecedented size, with over 100B parameters (Brown et al., 2020; Lepikhin et al., 2020). With the models and datasets growing, there is still no sign of saturating performance.



Idea #3: positional encoding

Transformers are equivariant to permutations



$$F_\theta(\texttt{permute}(\mathbf{T}_{\texttt{in}})) = \texttt{permute}(F_\theta(\mathbf{T}_{\texttt{in}}))$$

$$\texttt{attn}(\texttt{permute}(\mathbf{T}_{\texttt{in}})) = \texttt{permute}(\texttt{attn}(\mathbf{T}_{\texttt{in}}))$$

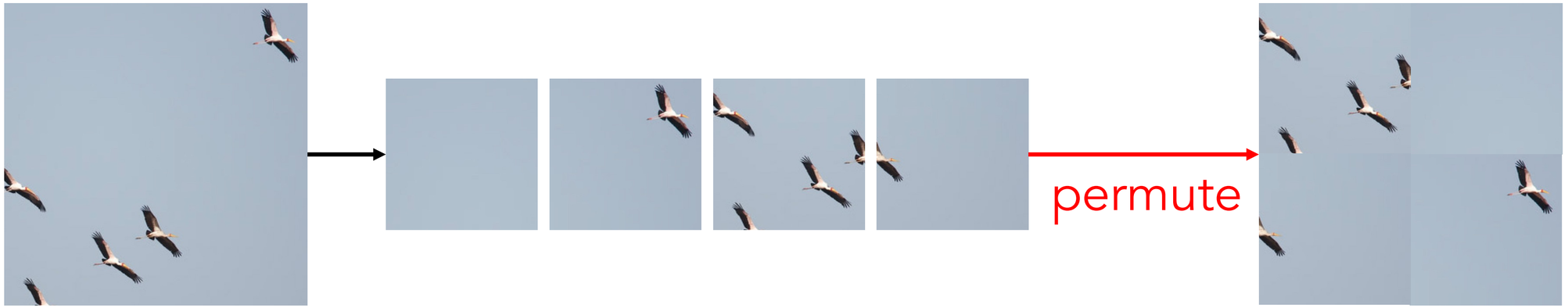
$$\downarrow$$

$$\texttt{transformer}(\texttt{permute}(\mathbf{T}_{\texttt{in}})) = \texttt{permute}(\texttt{transformer}(\mathbf{T}_{\texttt{in}}))$$

You can scramble patches in the input image and nothing big changes

I don’t want my transformer to be equivariant

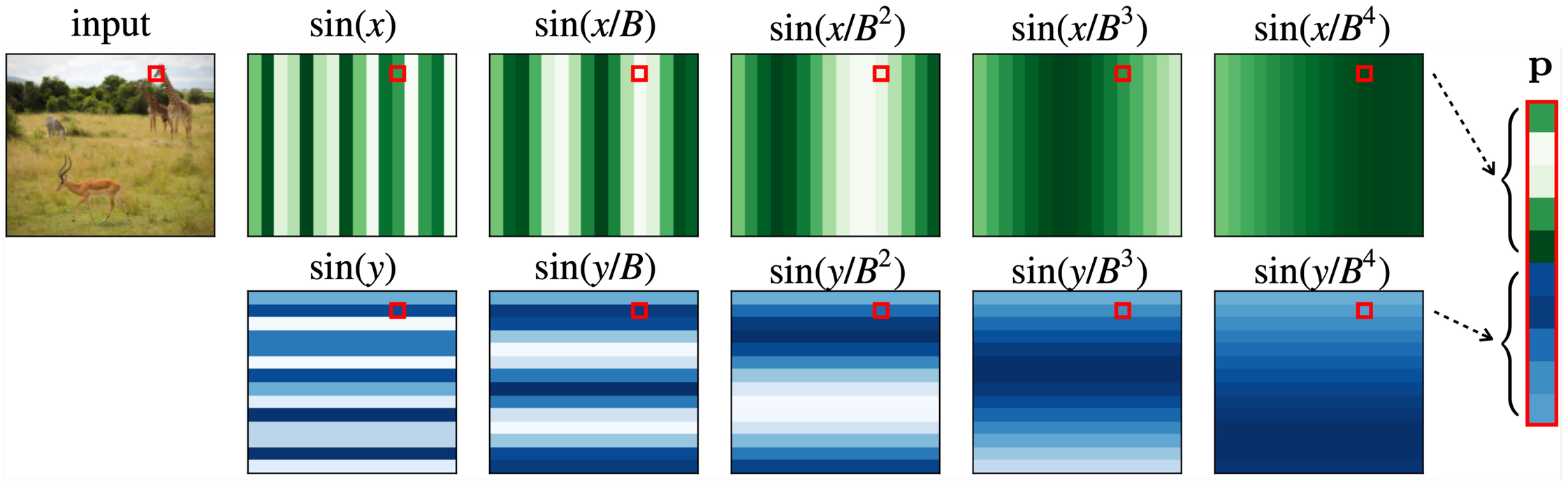
- Scrambling image patches can disrupt the spatial layout
  - E.g. what is the species of the bird in the upper right corner?
- We need to encode ***positional information*** with our tokens to maintain spatial layout
  - Done via a **positional encoding**, which concatenates position onto each token



Is the top right bird the same species as the bottom left bird?

Fourier positional codes

The positional encoding could be a simple x,y scaler coordinates, but in practice we use periodic representations of position (Fourier basis)



Putting it all together (again)

Published as a conference paper at ICLR 2021

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Alexey Dosovitskiy*,†, Lucas Beyer*, Alexander Kolesnikov*, Dirk Weissenborn*, Xiaohua Zhai*, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby*,†

*equal technical contribution, †equal advising

Google Research, Brain Team

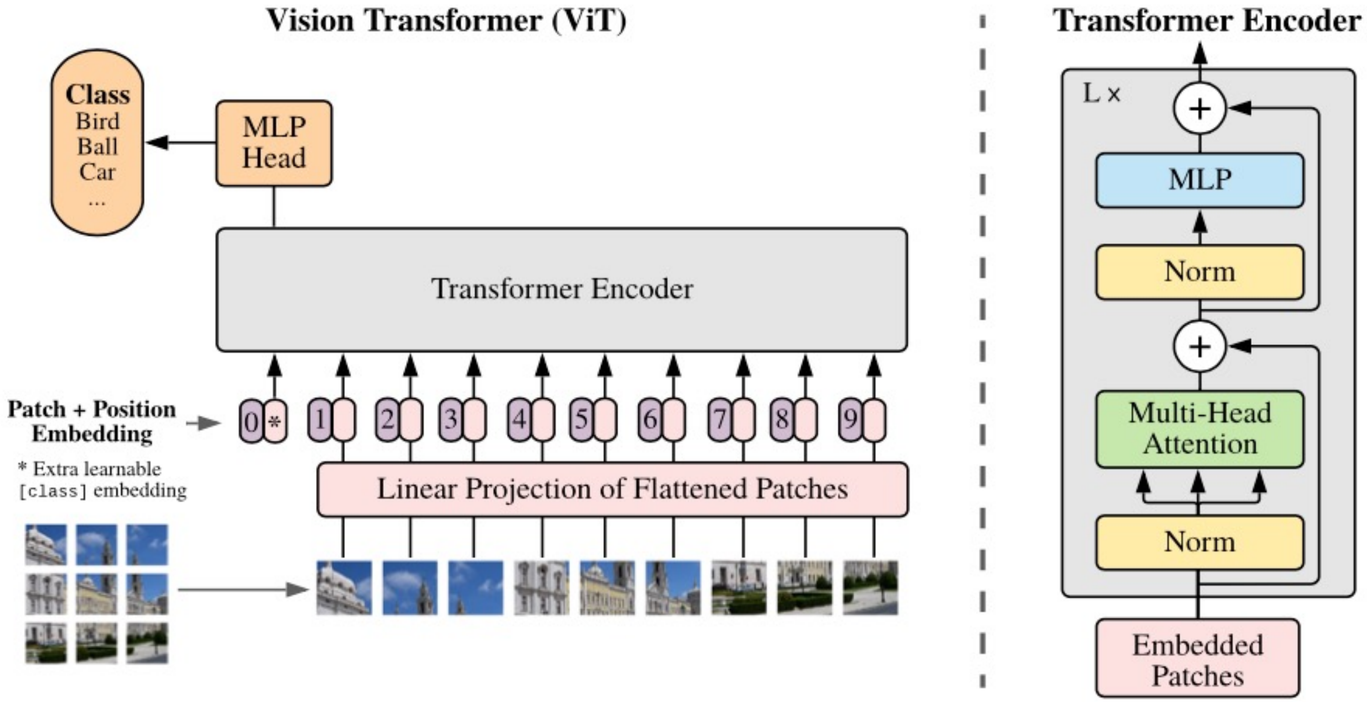
{adosovitskiy, neilhoulsby}@google.com

ABSTRACT

While the Transformer architecture has become the de-facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks while keeping their overall structure in place. We show that this reliance on CNNs is not necessary and a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks. When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.), Vision Transformer (ViT) attains excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train.[1]

1 INTRODUCTION

Self-attention-based architectures, in particular Transformers (Vaswani et al., 2017), have become the model of choice in natural language processing (NLP). The dominant approach is to pre-train on a large text corpus and then fine-tune on a smaller task-specific dataset (Devlin et al., 2019). Thanks to Transformers' computational efficiency and scalability, it has become possible to train models of unprecedented size, with over 100B parameters (Brown et al., 2020; Lepikhin et al., 2020). With the models and datasets growing, there is still no sign of saturating performance.



Why do people like ViT’s?

Why do people like ViT's?

ViT's scale better with data than CNN's

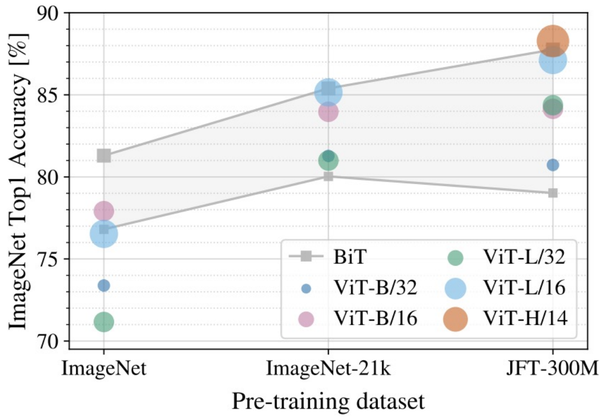


Figure 3: Transfer to ImageNet. While large ViT models perform worse than BiT ResNets (shaded area) when pre-trained on small datasets, they shine when pre-trained on larger datasets. Similarly, larger ViT variants overtake smaller ones as the dataset grows.

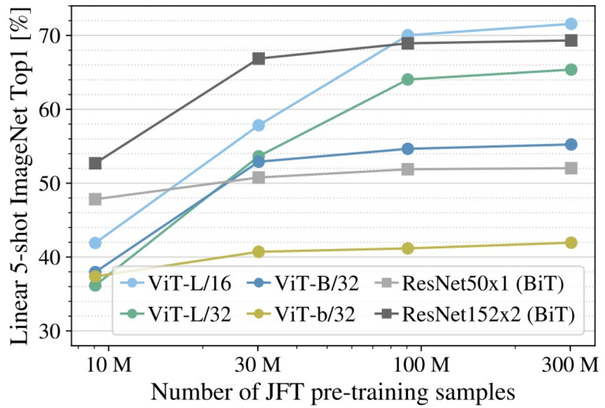


Figure 4: Linear few-shot evaluation on ImageNet versus pre-training size. ResNets perform better with smaller pre-training datasets but plateau sooner than ViT, which performs better with larger pre-training. ViT-b is ViT-B with all hidden dimensions halved.

["Vision Transformer", Dosovitskiy et al. 2020]

Why do people like ViT's?

1. Global Context
2. Dynamic Attention
3. Scalability
4. Highest numbers on the leaderboard
5. Having one architecture across modalities

DISCLAIMER: there are several papers that show ConvNets can be adapted to see similar benefits (e.g. ConvNext)

["A ConvNet for the 2020s", Liu et al]

~~Some~~ one application

Visual Question Answering

MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models

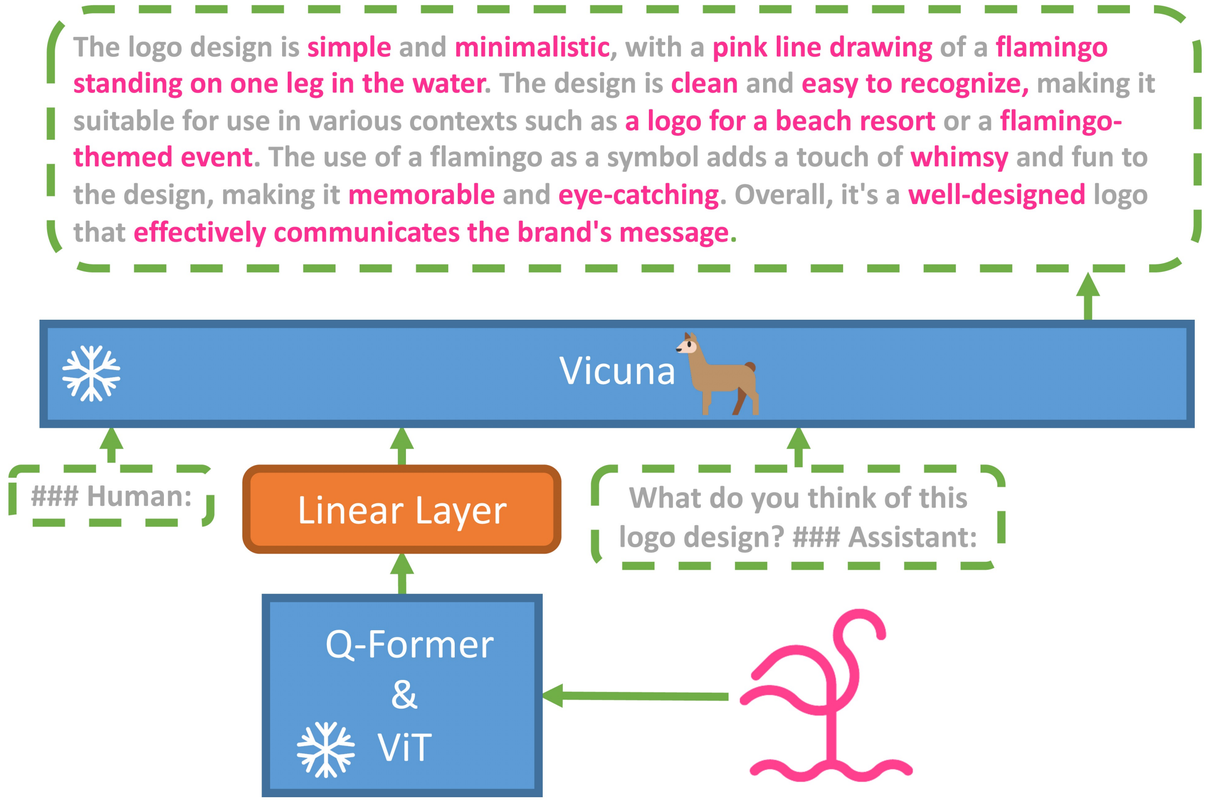
Deyao Zhu*, Jun Chen*, Xiaoqian Shen, Xiang Li, Mohamed Elhoseiny
King Abdullah University of Science and Technology
{deyao.zhu,jun.chen,xiaoqian.shen,
xiang.li.1,mohamed.elhoseiny}@kaust.edu.sa

Abstract

The recent GPT-4 has demonstrated extraordinary multi-modal abilities, such as directly generating websites from handwritten text and identifying humorous elements within images. These features are rarely observed in previous vision-language models. However, the technical details behind GPT-4 continue to remain undisclosed. We believe that the enhanced multi-modal generation capabilities of GPT-4 stem from the utilization of sophisticated large language models (LLM). To examine this phenomenon, we present MiniGPT-4, which aligns a frozen visual encoder with a frozen advanced LLM, Vicuna, using one projection layer. Our work, for the first time, uncovers that properly aligning the visual features with an advanced large language model can possess numerous advanced multi-modal abilities demonstrated by GPT-4, such as detailed image description generation and website creation from hand-drawn drafts. Furthermore, we also observe other emerging capabilities in MiniGPT-4, including writing stories and poems inspired by given images, teaching users how to cook based on food photos, and so on. In our experiment, we found that the model trained on short image caption pairs could produce unnatural language outputs (e.g., repetition and fragmentation). To address this problem, we curate a detailed image description dataset in the second stage to finetune the model, which consequently improves the model's generation reliability and overall usability. Our code, pre-trained model, and collected dataset are available at https://minigpt-4.github.io/.

1 Introduction

In recent years, large language models (LLMs) have experienced rapid advancements (Ouyang et al., 2022; OpenAI, 2022; Brown et al., 2020; Scao et al., 2022a; Touvron et al., 2023; Chowdhery et al., 2022; Hoffmann et al., 2022). With exceptional language understanding capabilities, these models can perform a variety of intricate linguistic tasks in a zero-shot manner. Notably, GPT-4, a large-scale multimodal model, has been recently introduced and demonstrated several impressive capabilities of vision-language understanding and generation (OpenAI, 2023). For example, GPT-4 can produce detailed and accurate image descriptions, explain unusual visual phenomena, and even construct websites based on handwritten text instructions.



Extra Slides
(+Autoregressive modeling)

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Alexey Dosovitskiy[*,†], **Lucas Beyer**[*], **Alexander Kolesnikov**[*], **Dirk Weissenborn**[*], **Xiaohua Zhai**[*], **Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby**[*,†]

[*]equal technical contribution, [†]equal advising

Google Research, Brain Team

{adosovitskiy, neilhoulsby}@google.com

ABSTRACT

While the Transformer architecture has become the de-facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks while keeping their overall structure in place. We show that this reliance on CNNs is not necessary and a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks. When pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks (ImageNet, CIFAR-100, VTAB, etc.), Vision Transformer (ViT) attains excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train.[1]

1 INTRODUCTION

Self-attention-based architectures, in particular Transformers (Vaswani et al., 2017), have become the model of choice in natural language processing (NLP). The dominant approach is to pre-train on a large text corpus and then fine-tune on a smaller task-specific dataset (Devlin et al., 2019). Thanks to Transformers' computational efficiency and scalability, it has become possible to train models of unprecedented size, with over 100B parameters (Brown et al., 2020; Lepikhin et al., 2020). With the models and datasets growing, there is still no sign of saturating performance.

https://arxiv.org/abs/2010.11929

lucidrains / vit-pytorch Public

Sponsor | Watch 125 | Fork 1.8k | Star 10.8k

Code | Issues 89 | Pull requests 4 | Actions | Projects | Wiki | Security | Insights

main | 2 branches | 143 tags | Go to file | Add file | Code

lucidrains offer way for extractor to return latents without detaching them — f86e052 4 days ago — 249 commits

.github	sponsor button	4 months ago
examples	fix transforms for val an test process	11 months ago
images	add EsViT, by popular request, an alternative to Dino that is compati...	3 months ago
tests	add some tests	7 months ago
vit_pytorch	offer way for extractor to return latents without detaching them	4 days ago
.gitignore	Initial commit	2 years ago
LICENSE	Initial commit	2 years ago
MANIFEST.in	include tests in package for conda	7 months ago
README.md	make extractor flexible for layers that output multiple tensors, show...	last month
setup.py	offer way for extractor to return latents without detaching them	4 days ago

About

Implementation of Vision Transformer, a simple way to achieve SOTA in vision classification with only a single transformer encoder, in Pytorch

computer-vision | transformers | artificial-intelligence | image-classification | attention-mechanism

Readme

MIT license

10.8k stars

125 watching

1.8k forks

Releases 142

https://github.com/lucidrains/vit-pytorch

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

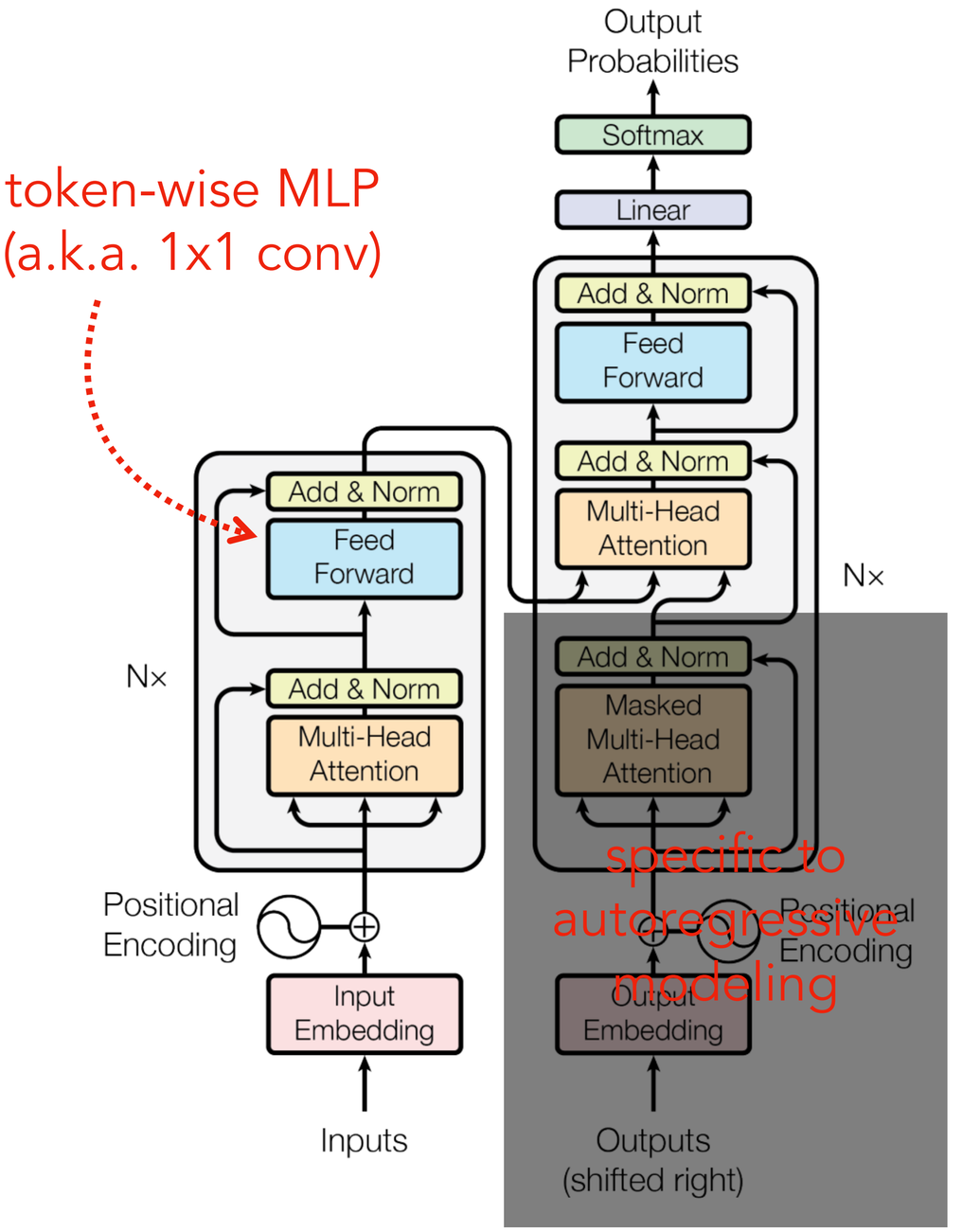
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

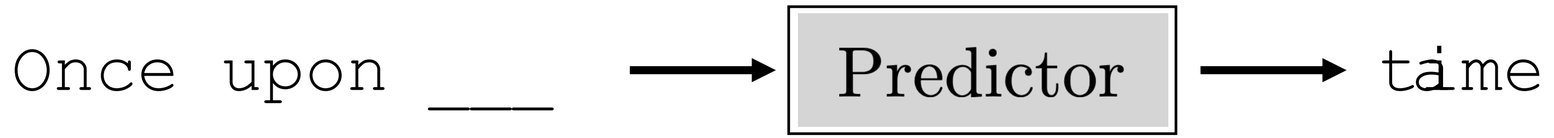
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

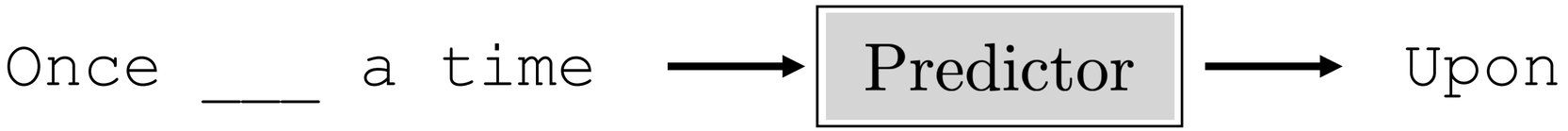
Abstract

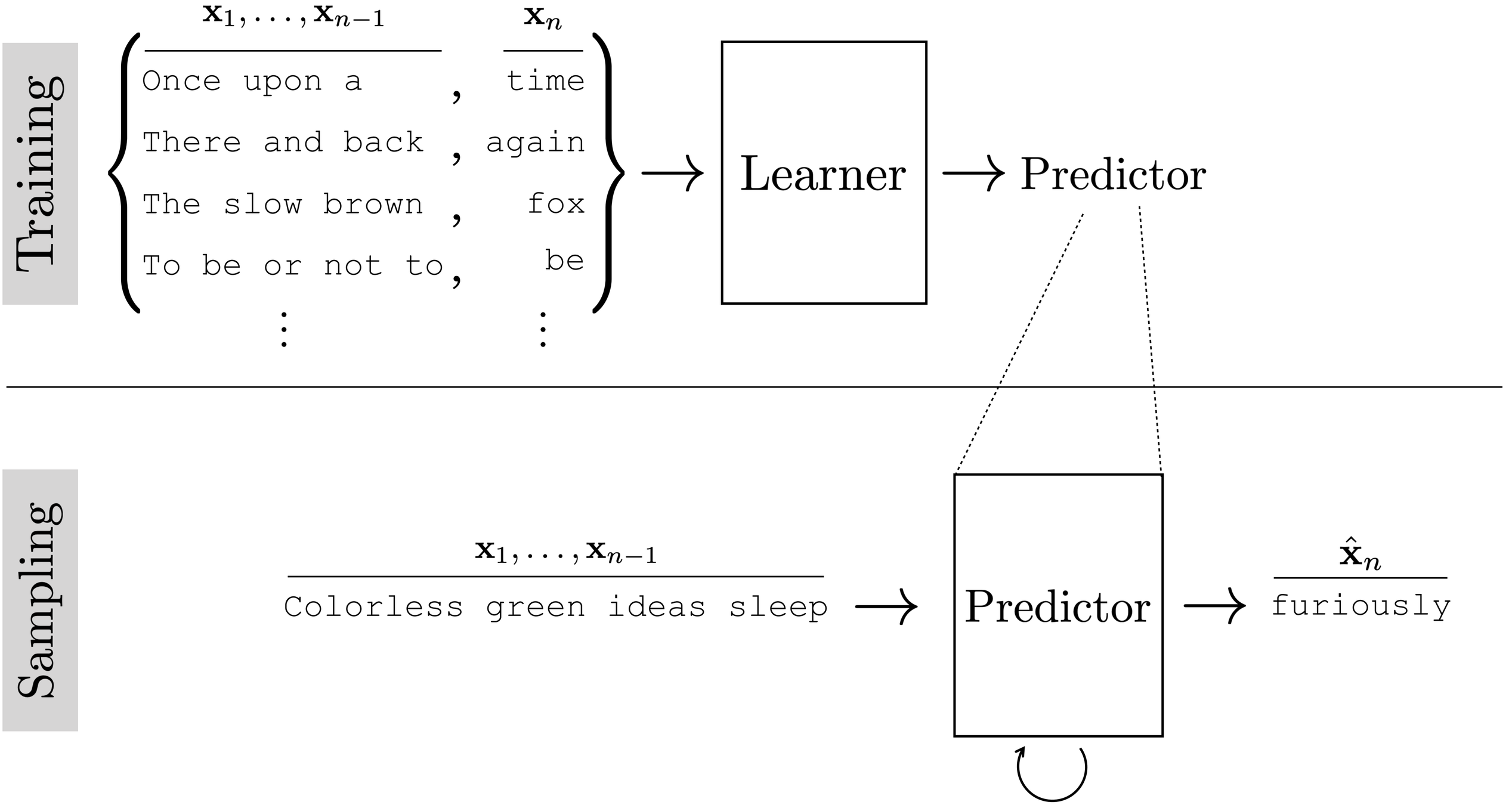
The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.



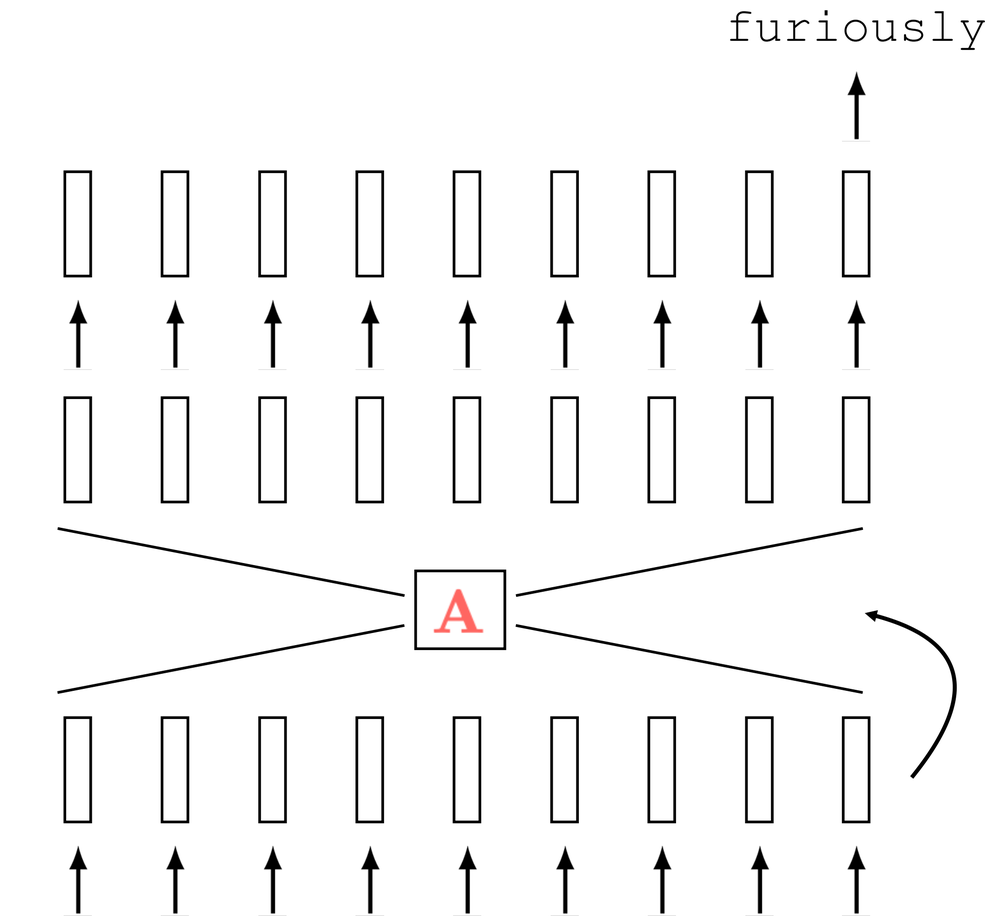
Autoregressive models



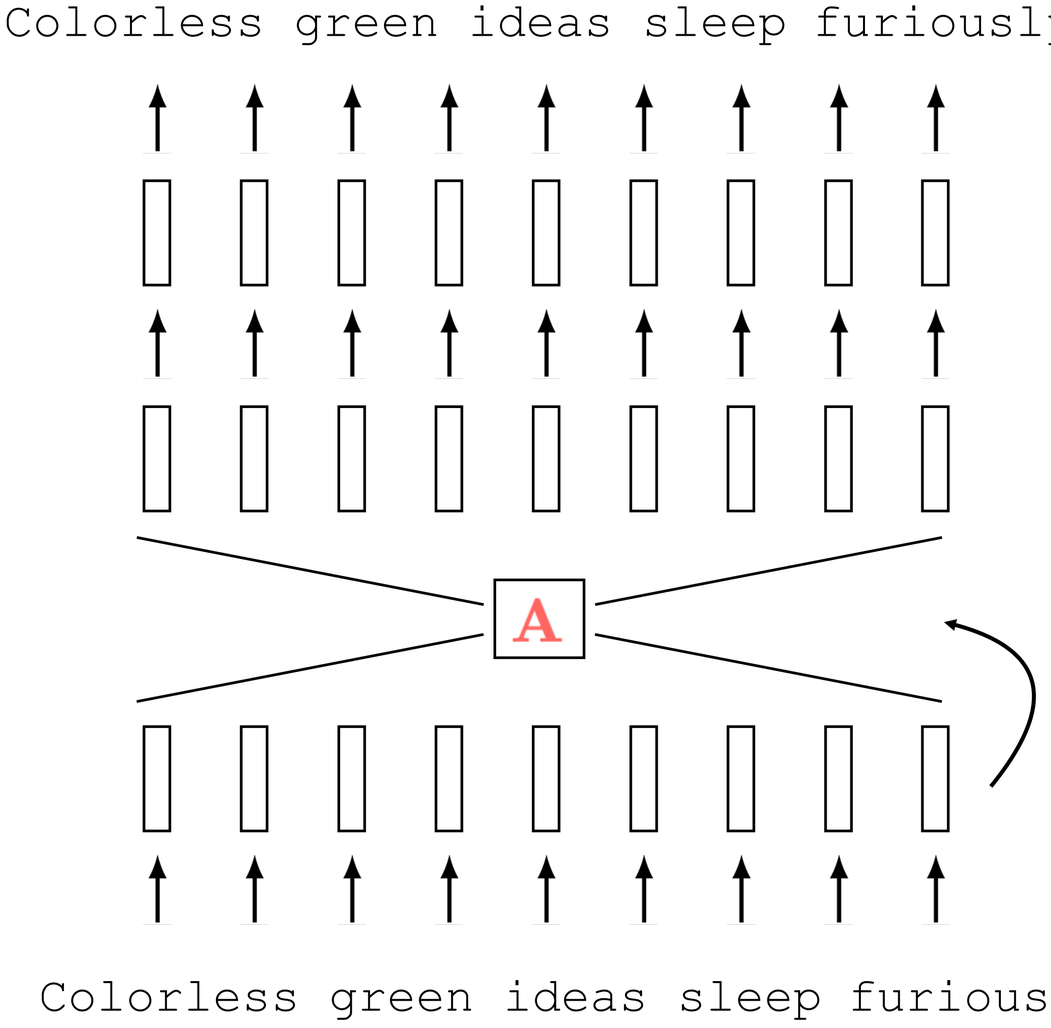


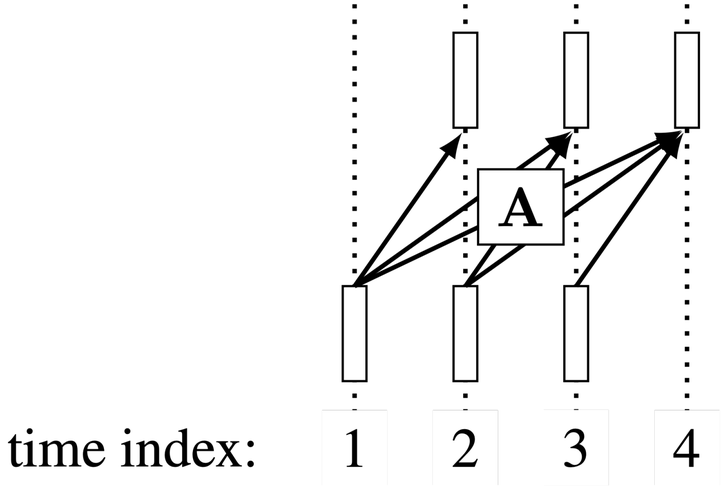

Training
$\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}$
$\mathbf{x}_n$
Once upon a , time
There and back , again
The slow brown , fox
To be or not to , be
Learner
Predictor
Sampling
$\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}$
Colorless green ideas sleep
Predictor
$\hat{\mathbf{x}}_n$
furiously

GPT (and many other related models)



GPT training (and many other related models)





karpathy / minGPT Public

Code | Issues 22 | Pull requests 11 | Actions | Projects | Security | Insights

master | 3 branches | 0 tags | Go to file | Add file | Code

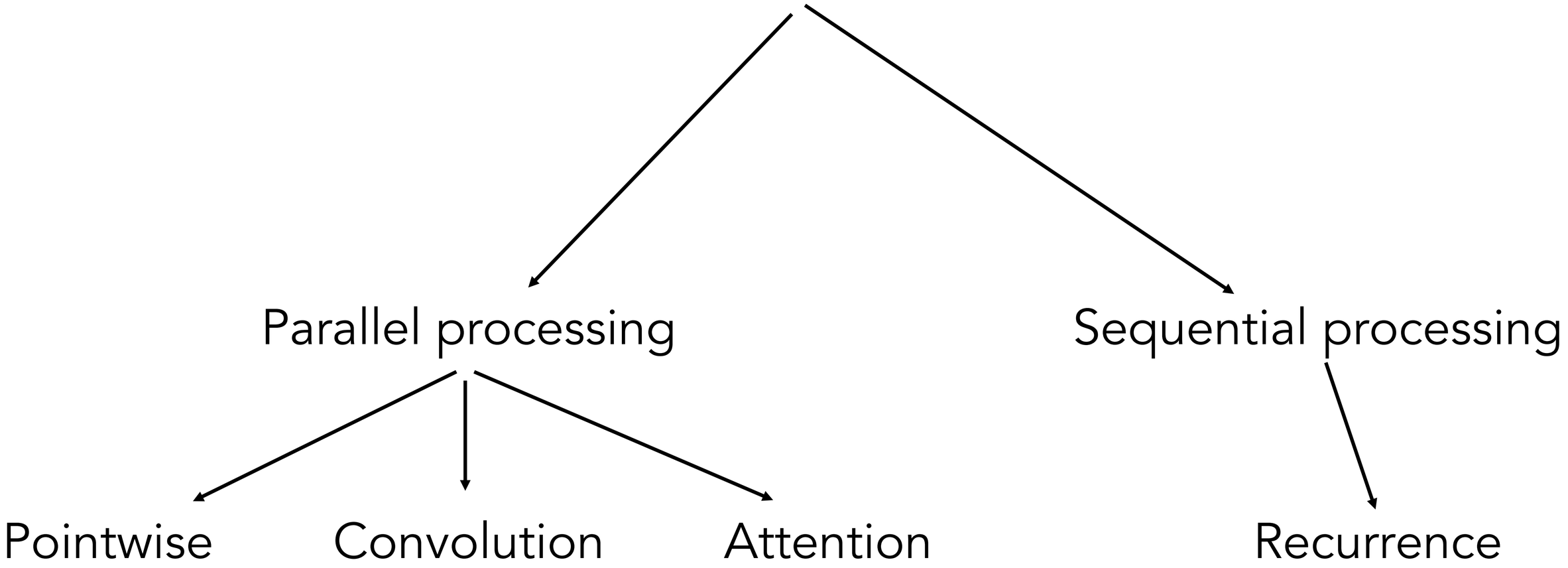
karpathy Merge pull request #84 from ericjang/master ... 7218bcf on Aug 4 **93** commits

mingpt	Use XOR operator ^ for checking assertion `type_given XOR param...	2 months ago
projects	refactor sequence generation into the model and match the huggingf...	3 months ago
tests	add a refactored BPE encoder from openai. Basically I dont super tru...	3 months ago
.gitignore	tiny tweaks to printing and some function apis	4 months ago
LICENSE	mit license file	2 years ago
README.md	Add setup.py to allow mingpt to be used as a third-party library	2 months ago
demo.ipynb	refactor sequence generation into the model and match the huggingf...	3 months ago
generate.ipynb	add a refactored BPE encoder from openai. Basically I dont super tru...	3 months ago
mingpt.jpg	first commit, able to multigpu train fp32 GPTs on math and character...	2 years ago
setup.py	Add setup.py to allow mingpt to be used as a third-party library	2 months ago

README.md

minGPT

Neural layers



Neural architectures

Architectures	Properties	Common uses
MLPs		
CNNs (just the conv layers)	Local, translation equivariant	Grid2Grid
GNNs	Local, Permutation equivariant	Set2Set
Transformers	Permutation equivariant	Set2Set
+ pos encoding	~Local, ~translation equivariant	Grid2Grid, Seq2Seq
RNNs	Markovian	Seq2Seq