Vision Transformers

Danna Gurari
University of Colorado Boulder
Fall 2023

Review

- Last lecture on semantic segmentation:
  - Problem
  - Applications
  - Datasets
  - Evaluation metric
  - Computer vision models: fully convolutional networks
- Assignments (Canvas):
  - Reading assignment was due earlier today
  - Next reading assignments due on Wednesday and next Monday
  - Project proposal due on Wednesday
  - (Student-led lectures start next week)
- Questions?

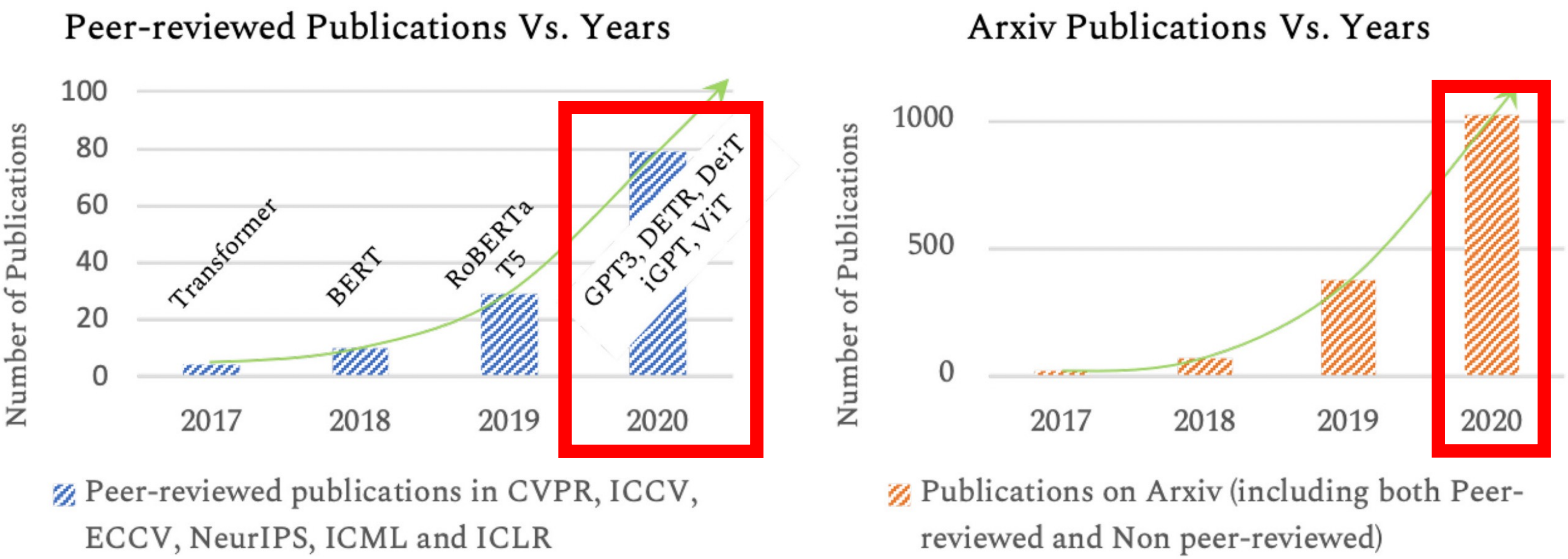
Today’s Topics

- Motivation
- ViT
- Swin Transformer
- Discussion

Today’s Topics

- Motivation
- ViT
- Swin Transformer
- Discussion

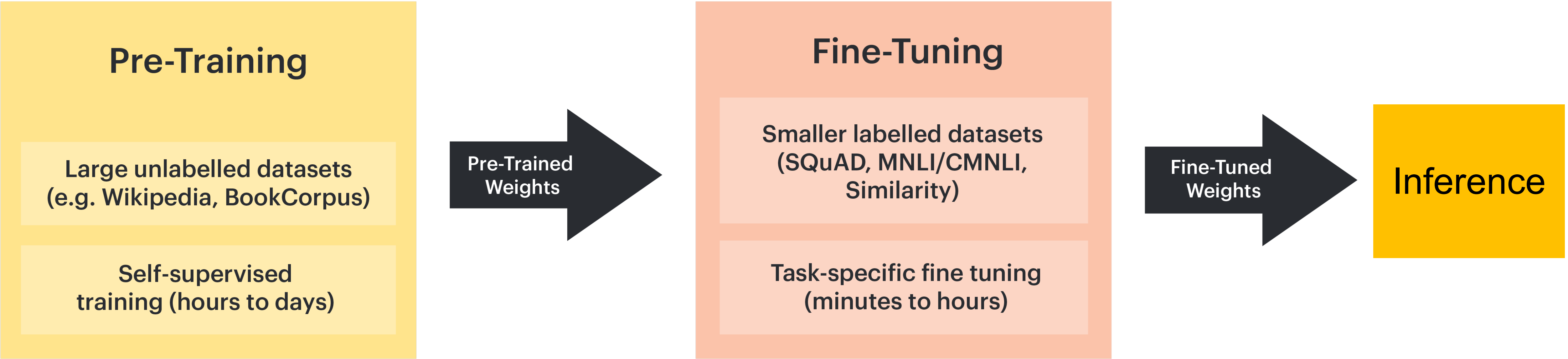
Introduced in 2017, Transformers Achieved Astonishing Performance for NLP Problems



Inspired, researchers in the computer vision community explored transformers for many vision problems and discovered they perform well!

Khan et al. Transformers in Vision: A Survey. CSUR 2022.

Common Paradigm for NLP Transformers



Transformers can provide effective features for downstream tasks!

https://docs.graphcore.ai/projects/bert-training/en/latest/bert.html

Today’s Topics

- Motivation
- ViT
- Swin Transformer
- Discussion

Why ViT?

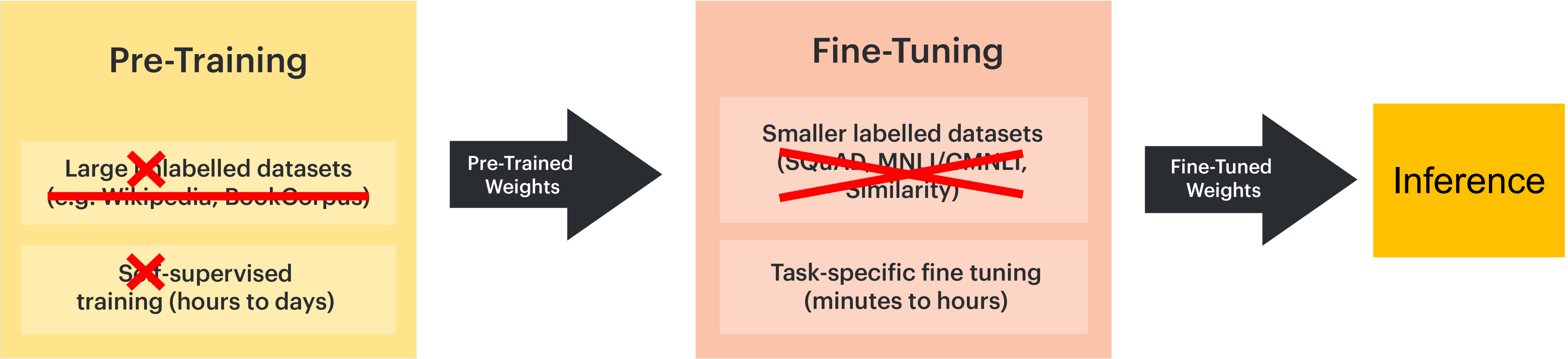
Named after the proposed technique: Vision Transformer

Dosovitskiy et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR 2021.

Novelty

- First paper to demonstrate that a pure transformer architecture can achieve strong performance on vision tasks, achieving comparable or better image classification results to the best methods at the time

Approach



https://docs.graphcore.ai/projects/bert-training/en/latest/bert.html

ViT: Key Ingredients for Success

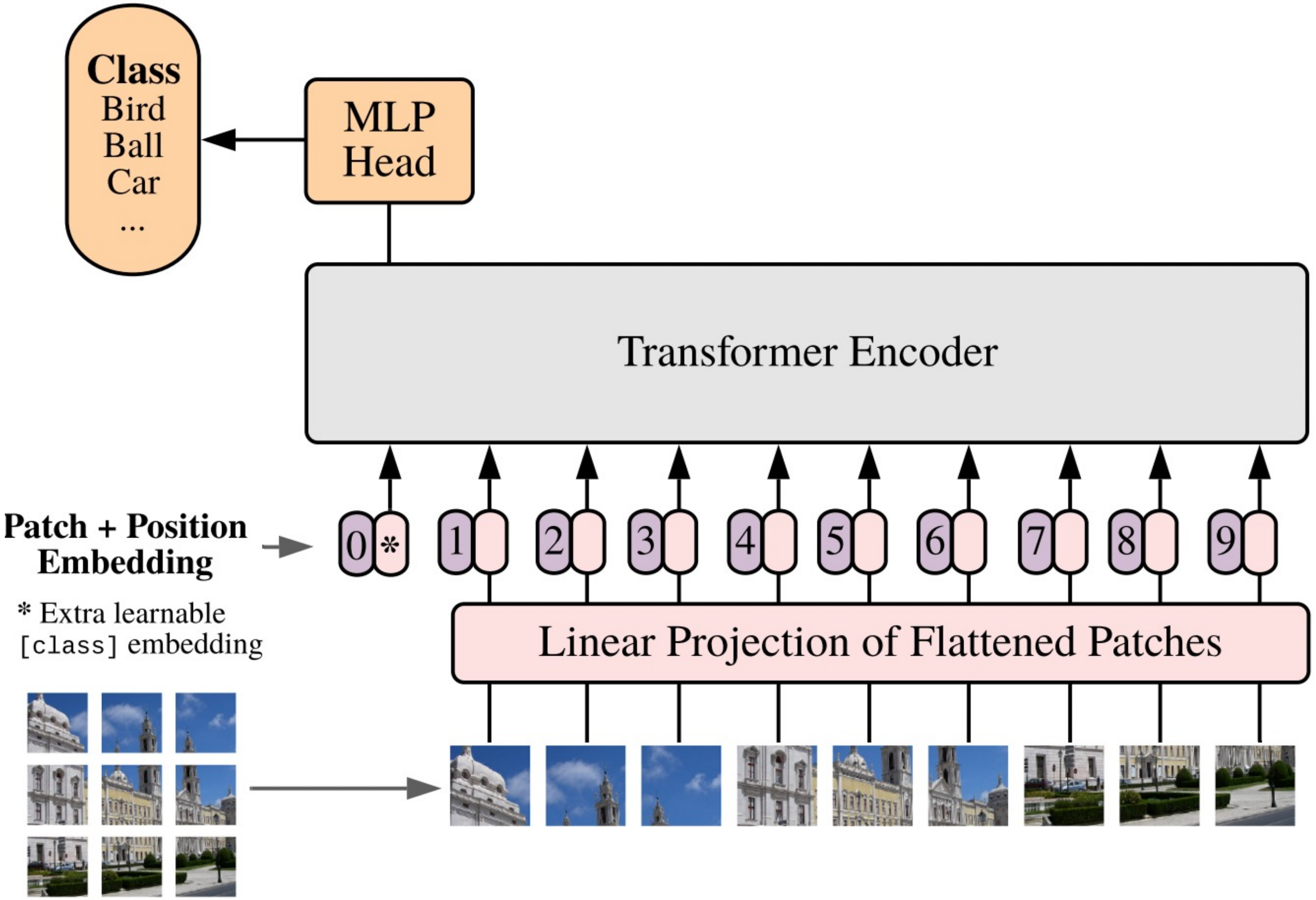
- Transformer architecture (embeds self-attention)

- Pre-training with massive amounts of data

ViT: Key Ingredients for Success

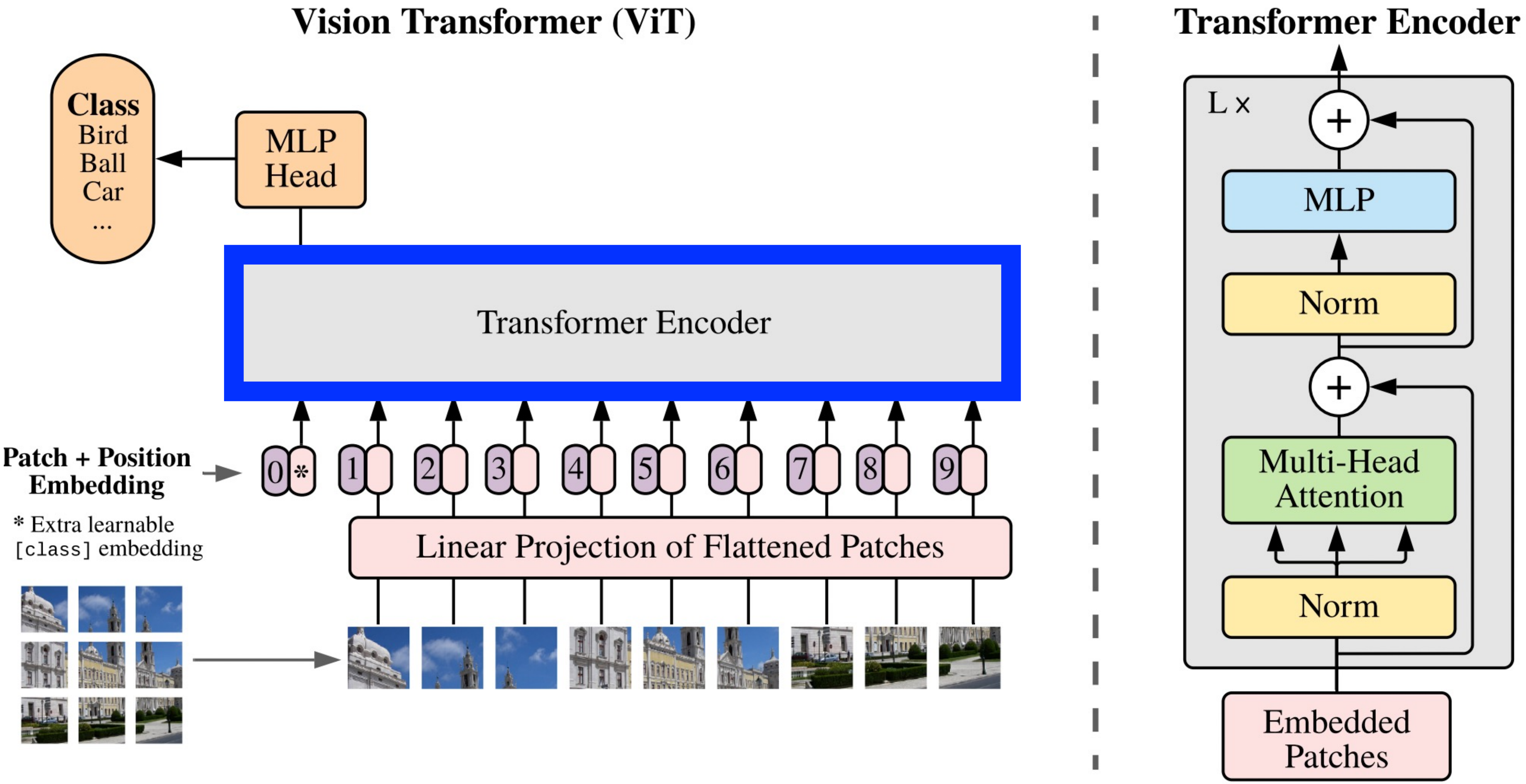
- Transformer architecture (embeds self-attention)
- Pre-training with massive amounts of data

Architecture



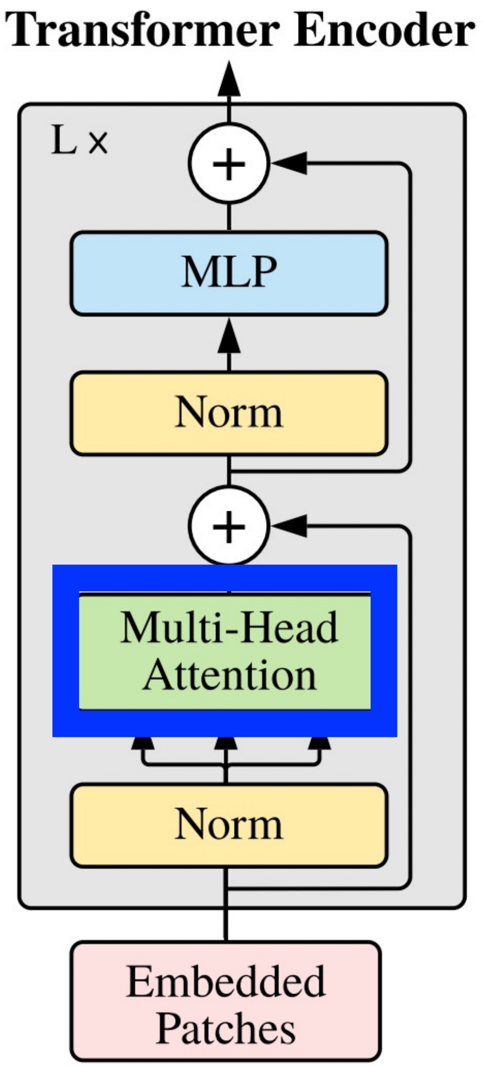
Dosovitskiy et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR 2021.

Architecture: Uses Popular BERT Architecture



Dosovitskiy et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR 2021.

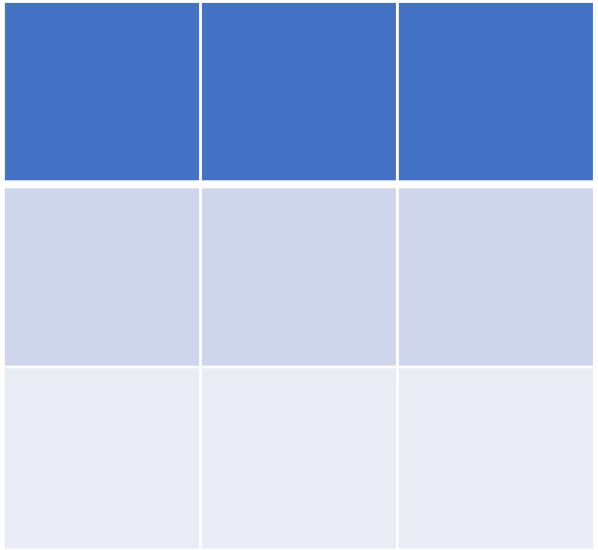
Architecture: Key Novelty is Self-Attention



Dosovitskiy et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR 2021.

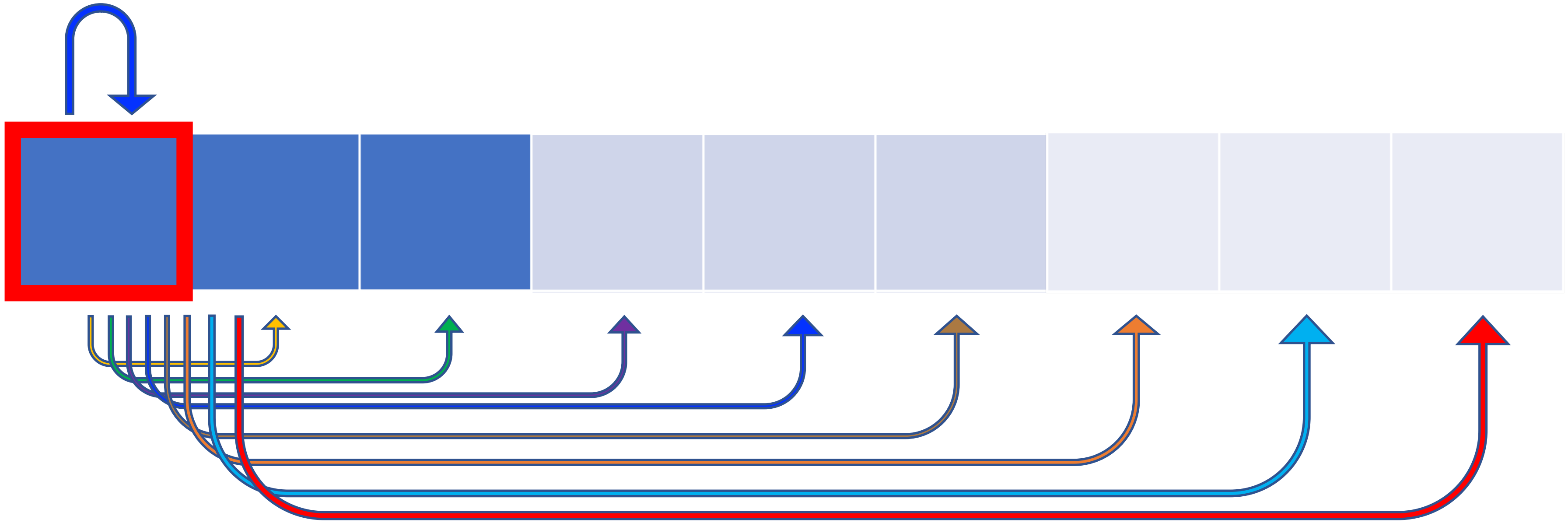
Self-Attention: Idea

New representation of each pixel showing its relationship to all pixels; e.g., assume this 3x3 image



Self-Attention: Idea

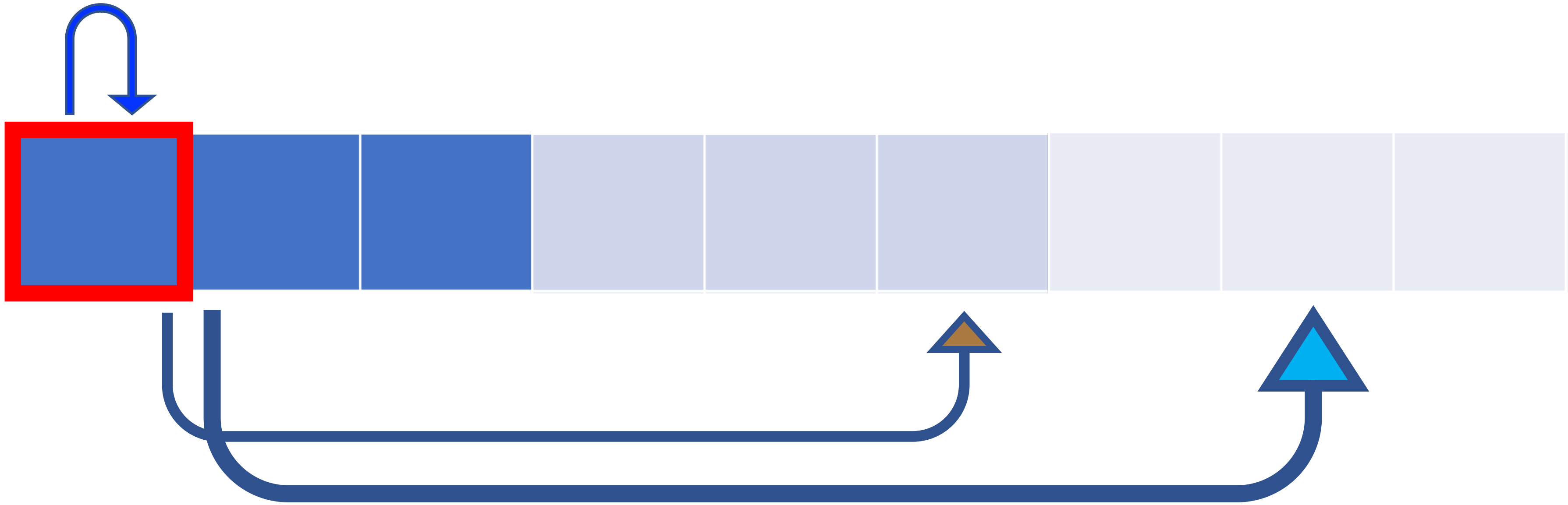
New representation of each pixel showing its relationship to all pixels; e.g., assume this 3x3 image



Learned new representation indicates which global information clarifies a pixel's meaning (e.g., include in the representation of a pixel of an eye context of what animal it belongs to)

Self-Attention: Idea

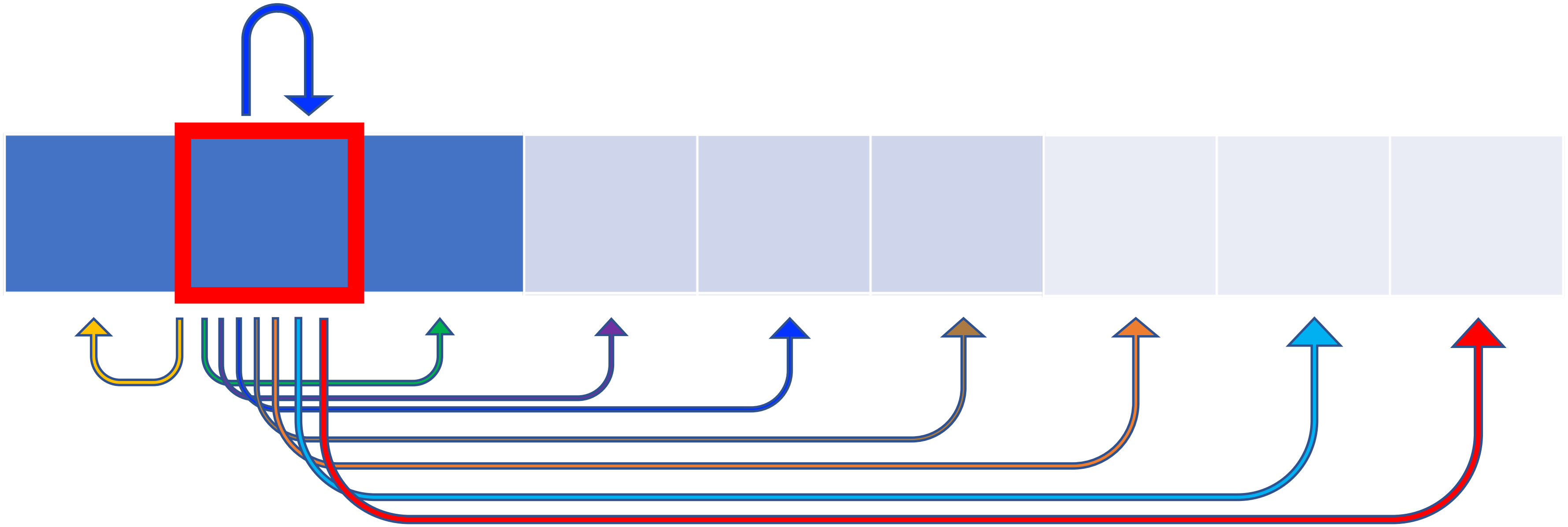
New representation of each pixel showing its relationship to all pixels; e.g., assume this 3x3 image



Learned new representation indicates which global information clarifies a pixel's meaning (e.g., include in the representation of a pixel of an eye context of what animal it belongs to)

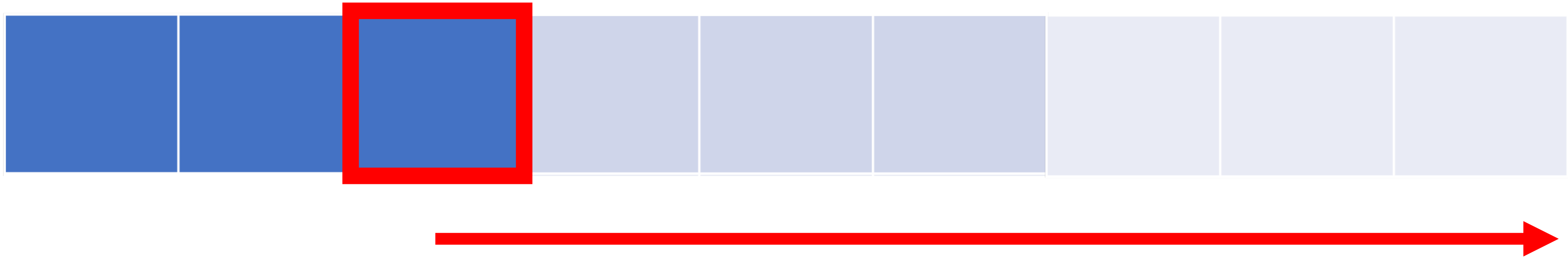
Self-Attention: Idea

New representation of each pixel showing its relationship to all pixels; e.g., assume this 3x3 image



Self-Attention: Idea

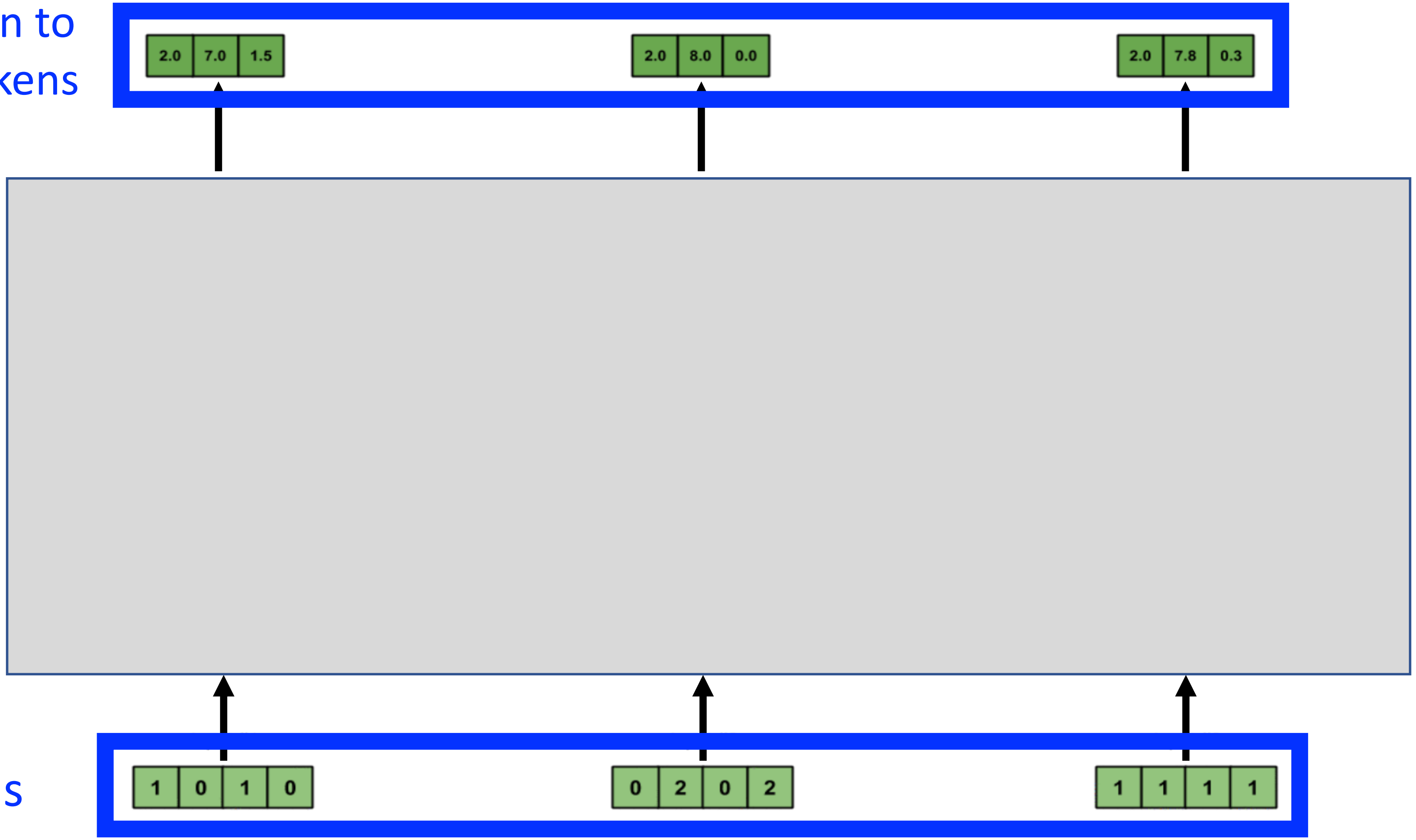
New representation of each pixel showing its relationship to all pixels; e.g., assume this 3x3 image



And so on for remaining image pixels…

Computing Self-Attention: Example

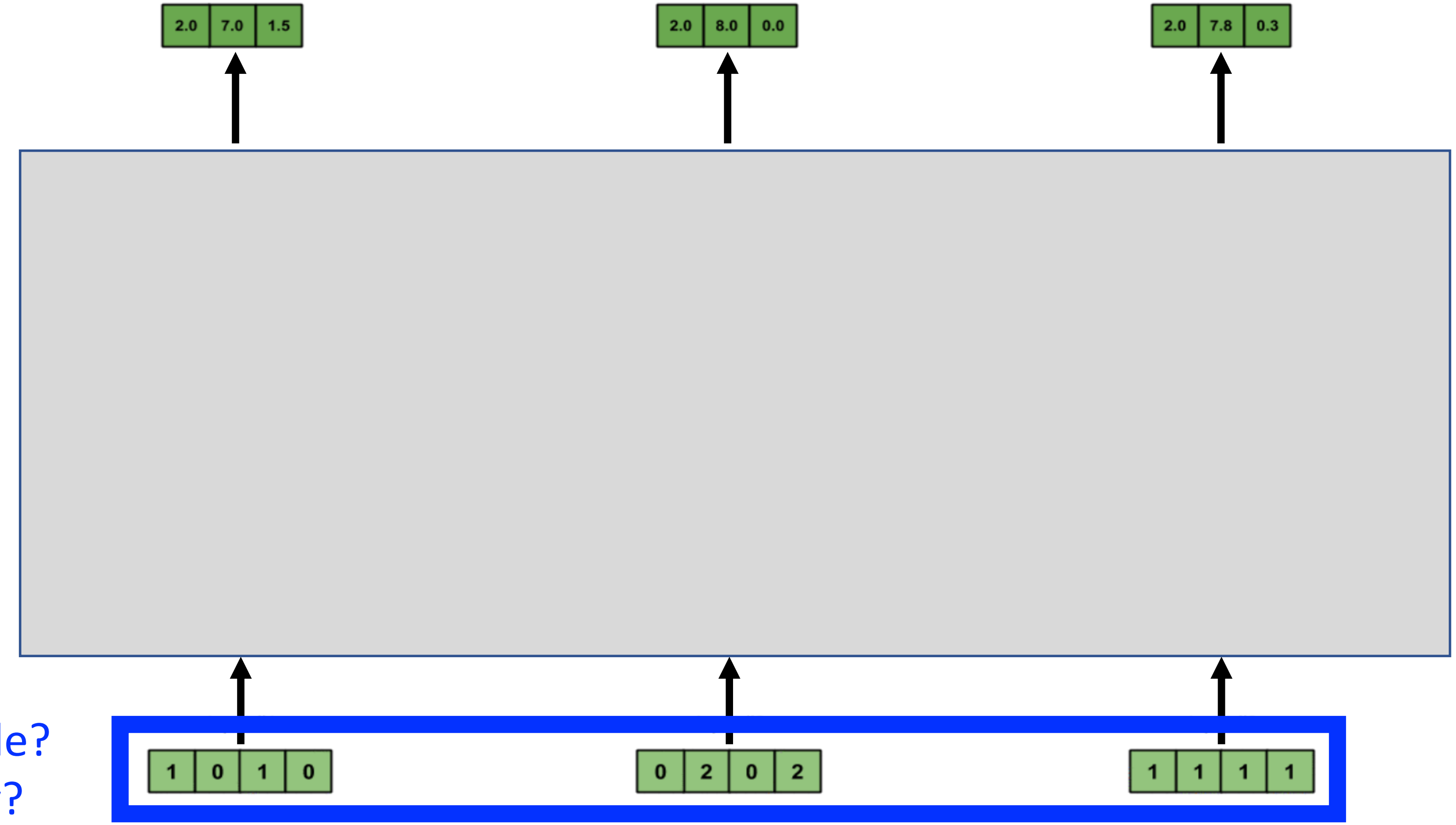
New representation of each input token to reflect each one's relationship to all tokens



Input tokens

https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

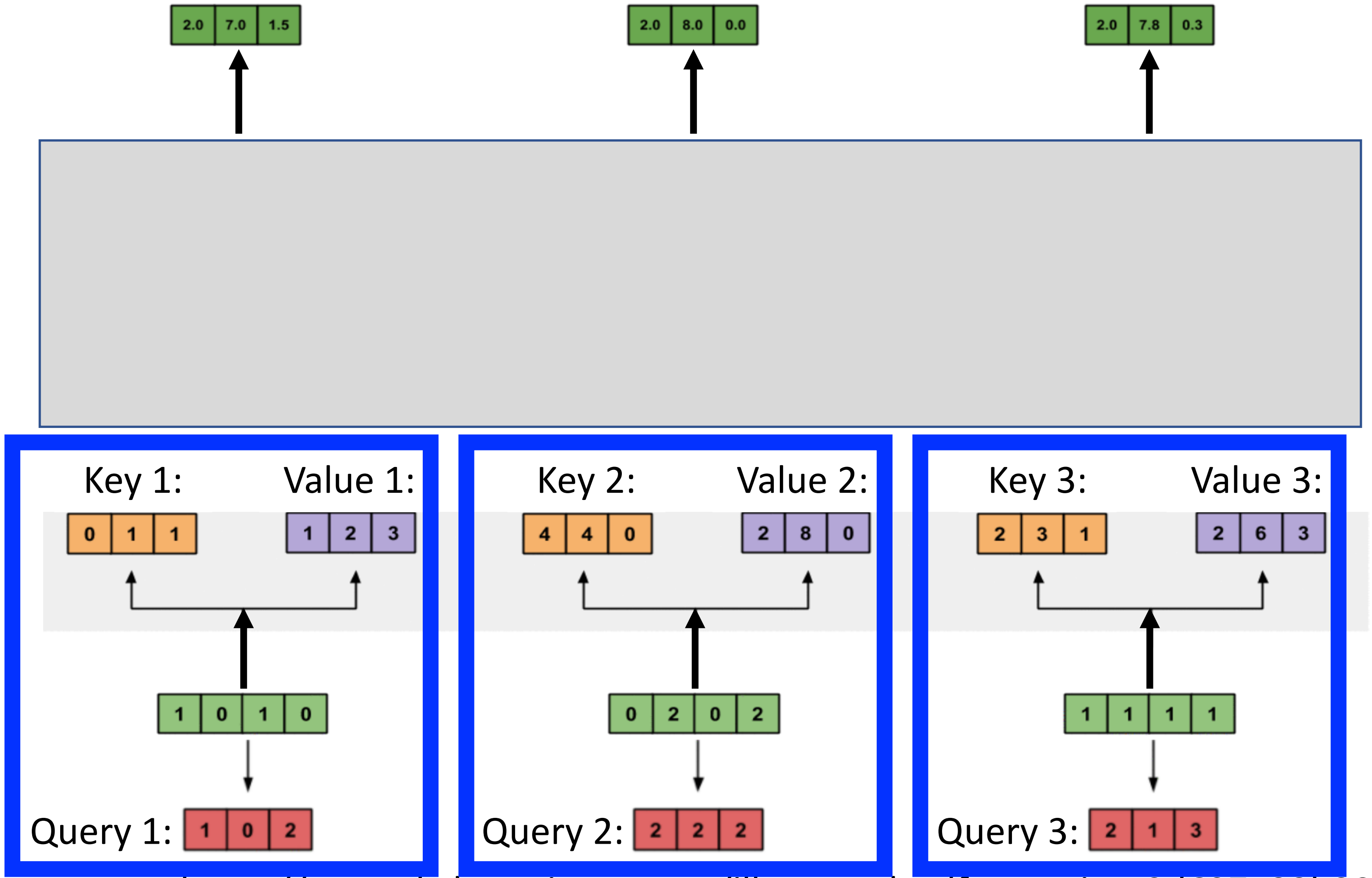
Computing Self-Attention: Example



- How many inputs are in this example?
- What is each input's dimensionality?

https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

Computing Self-Attention: Example



Three vectors are derived for each input by multiplying with three weight matrices (learned during training): query, key, and value

https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

Computing Self-Attention: Example

e.g., key weights

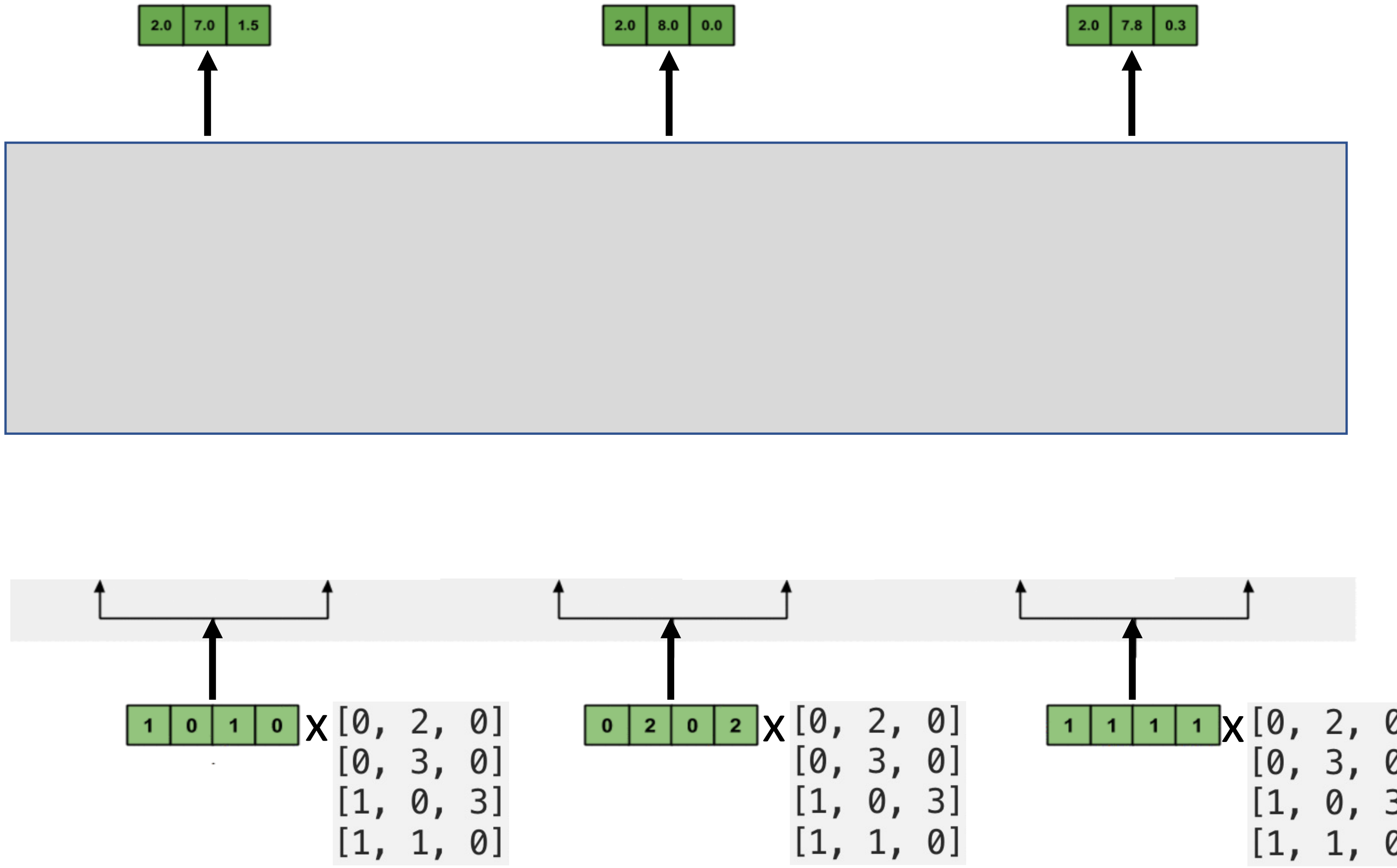
```
[0, 0, 1]
[1, 1, 0]
[0, 1, 0]
[1, 1, 0]
```

https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

Computing Self-Attention: Example

e.g., value weights

```
[0, 2, 0]
[0, 3, 0]
[1, 0, 3]
[1, 1, 0]
```

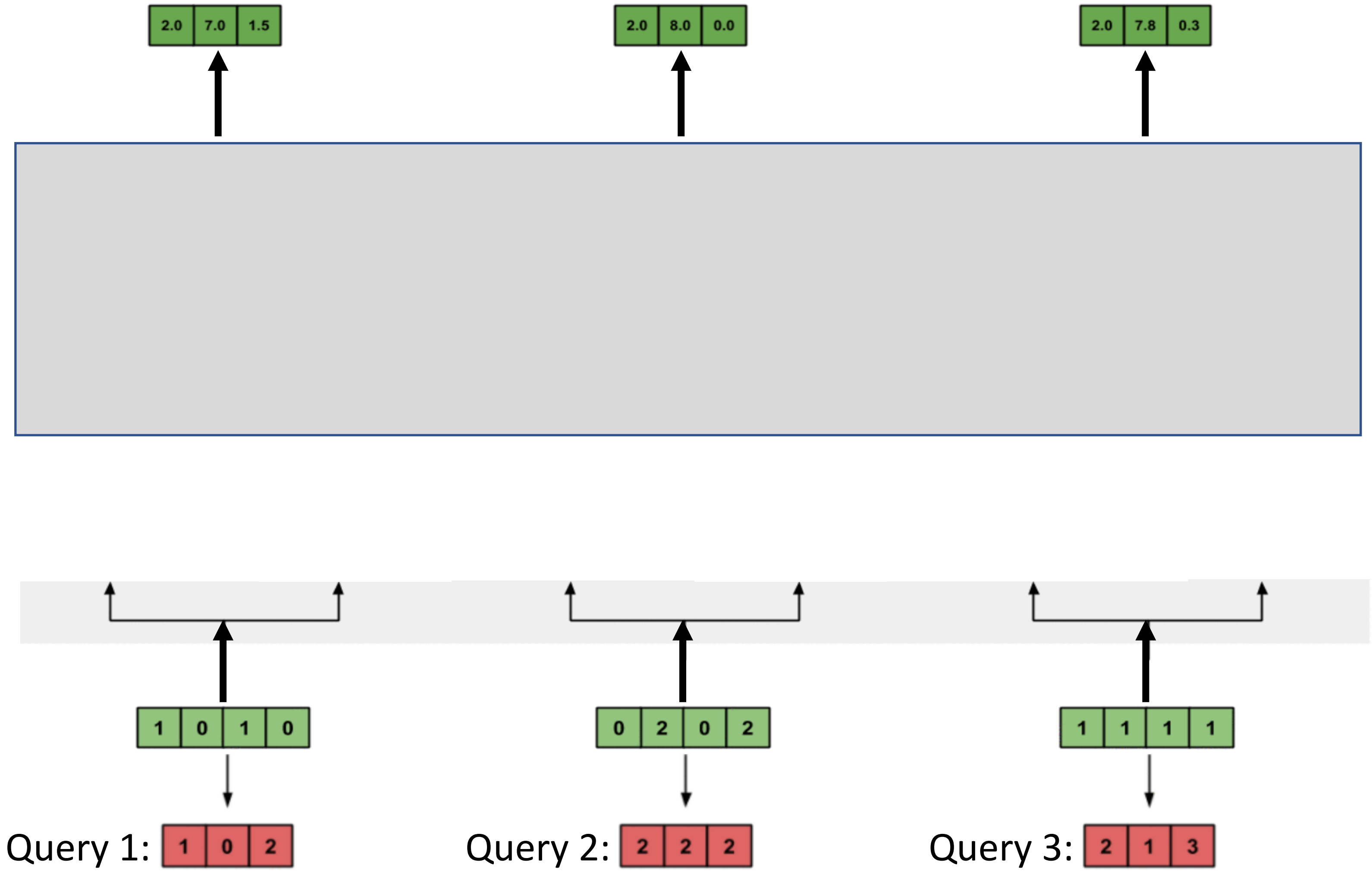


https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

Computing Self-Attention: Example

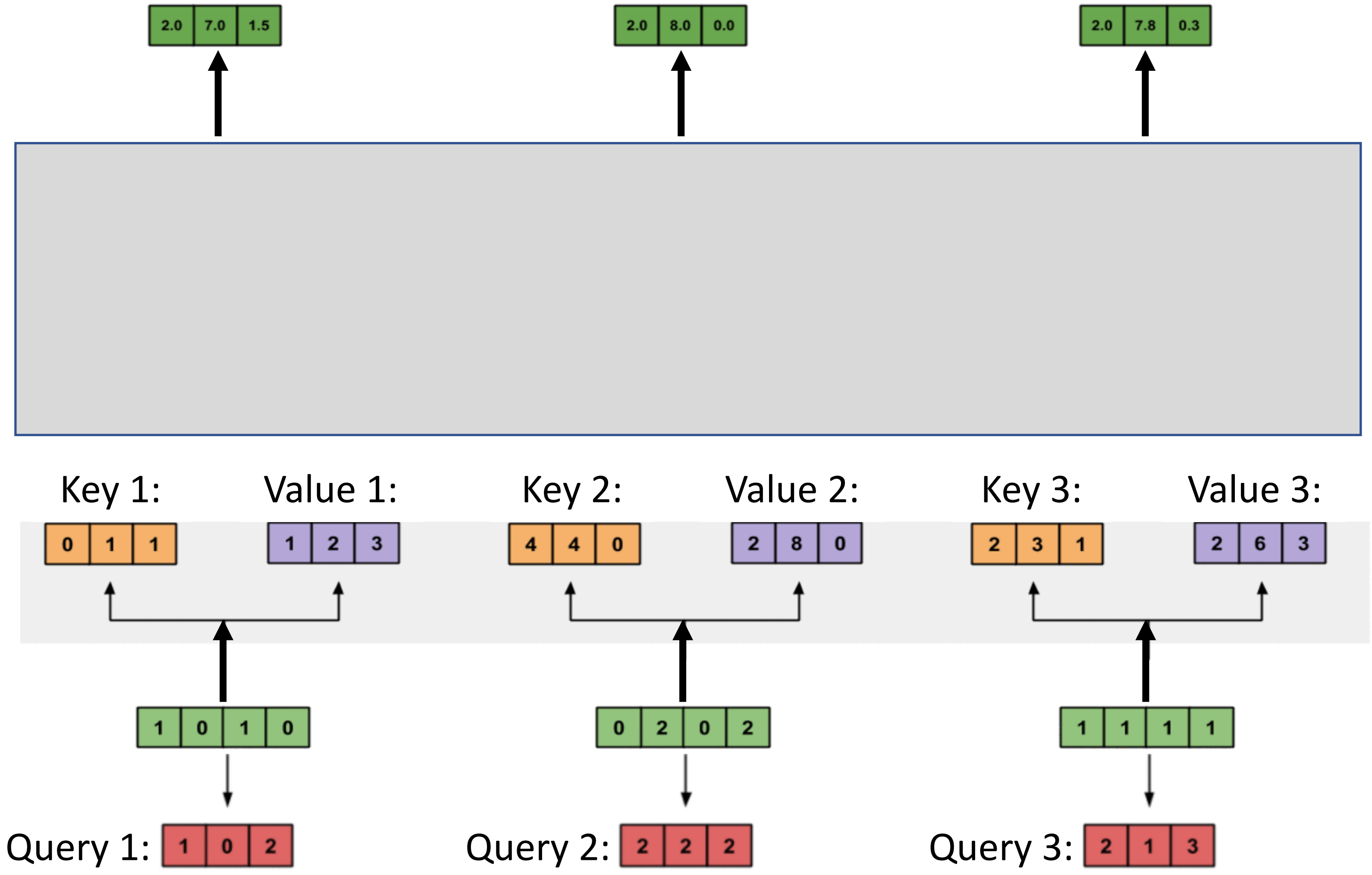
e.g., query weights

```
[1, 0, 1]
[1, 0, 0]
[0, 0, 1]
[0, 1, 1]
```



https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

Computing Self-Attention: Example



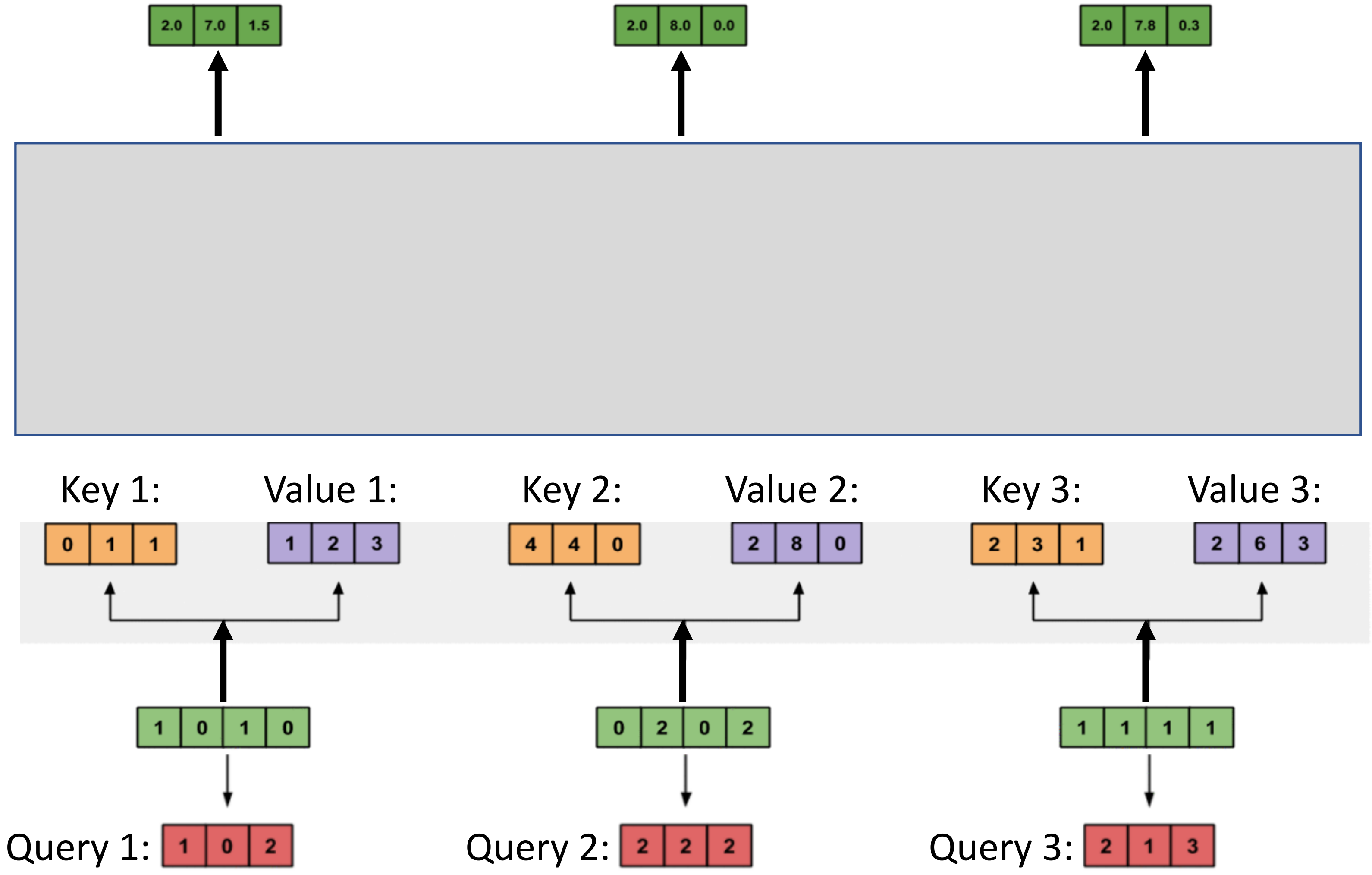
How many weight matrices are learned in this example?

https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

Computing Self-Attention: Example

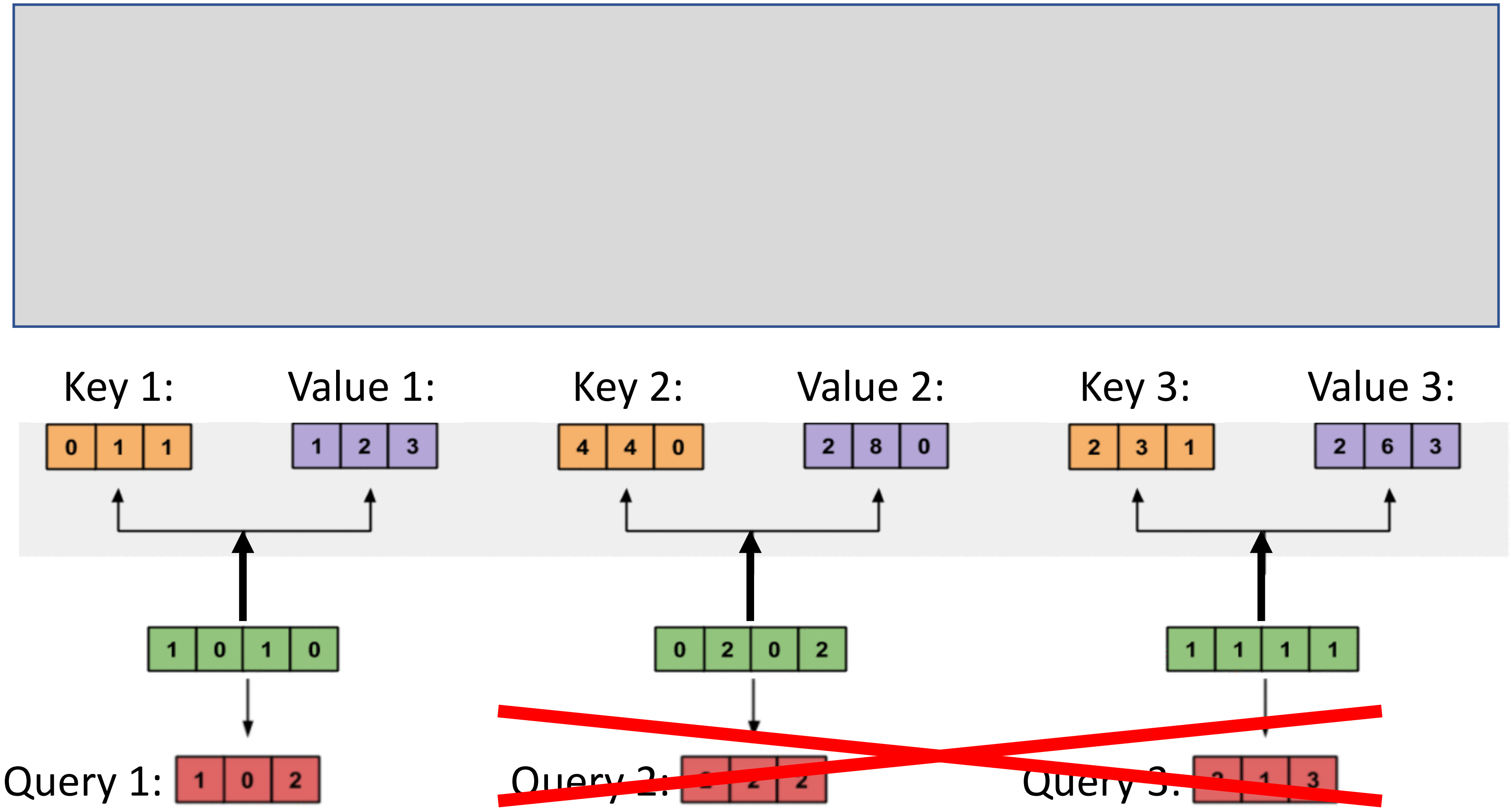
What is the purpose of the three weight matrices?

For each input, 2 of the derived vectors are used to compute **attention weights** (query and key) and the 3rd is **information** passed on for the new representation (value)



https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

Computing Self-Attention: Example



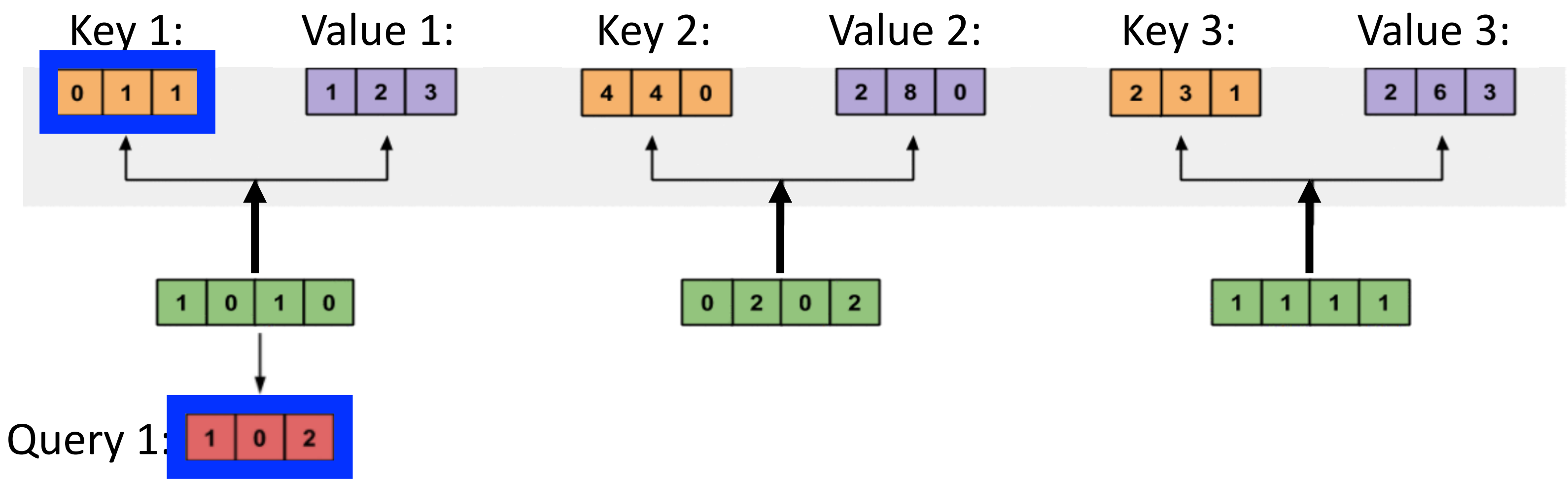
We now will examine how to find the new representation for the first input.

https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

Computing Self-Attention: Example

Attention score: dot product of query with all keys to identify relevant tokens; e.g.,

$$\begin{bmatrix} 1 & 0 & 2 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = ?$$

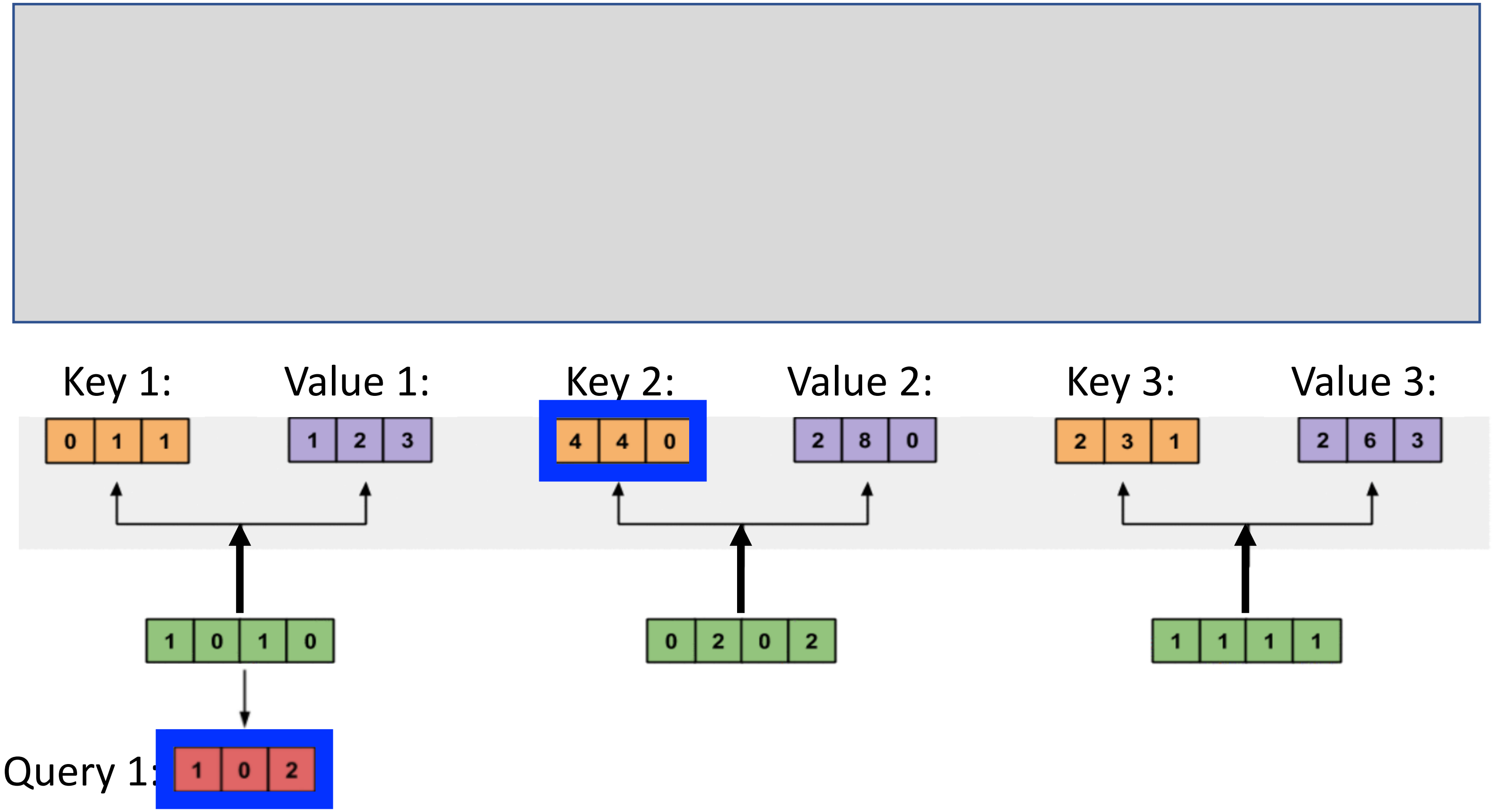


https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

Computing Self-Attention: Example

Attention score: dot product of query with all keys to identify relevant tokens; e.g.,

$$\begin{bmatrix} 1 & 0 & 2 \end{bmatrix} \times \begin{bmatrix} 4 \\ 4 \\ 0 \end{bmatrix} = ?$$

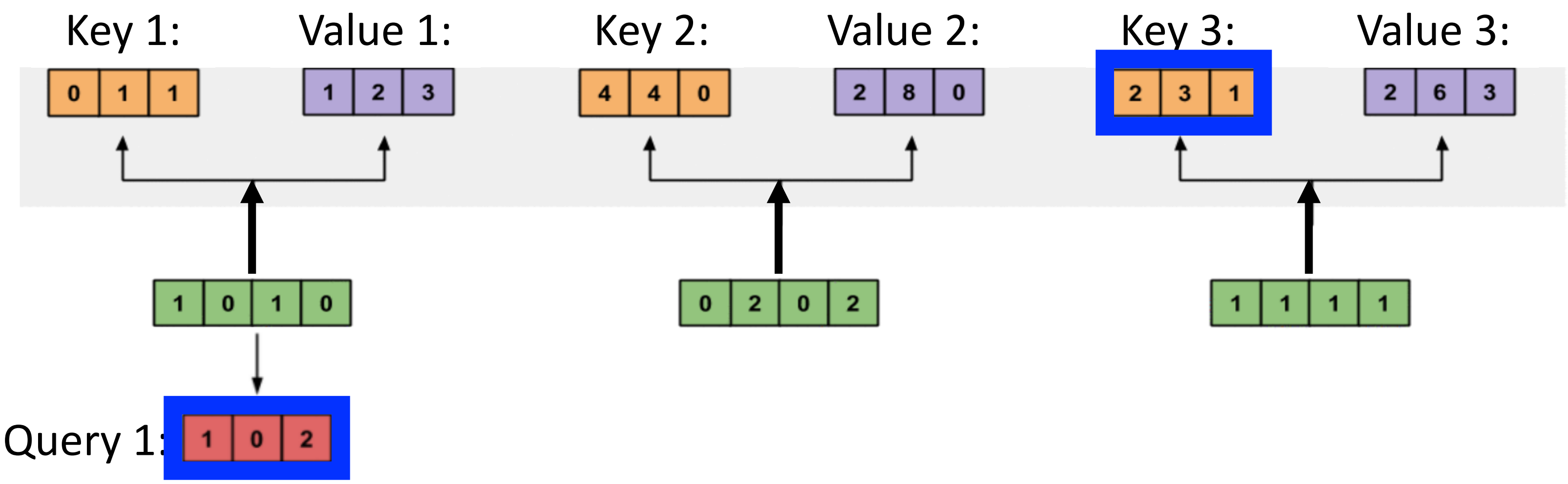


https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

Computing Self-Attention: Example

Attention score: dot product of query with all keys to identify relevant tokens; e.g.,

$$\begin{bmatrix} 1 & 0 & 2 \end{bmatrix} \times \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} = ?$$

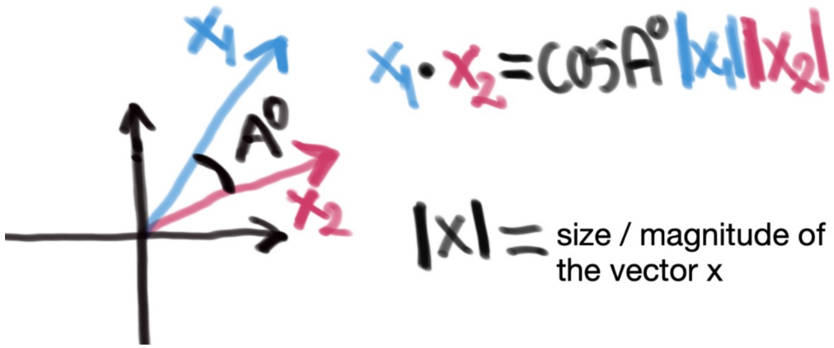


https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

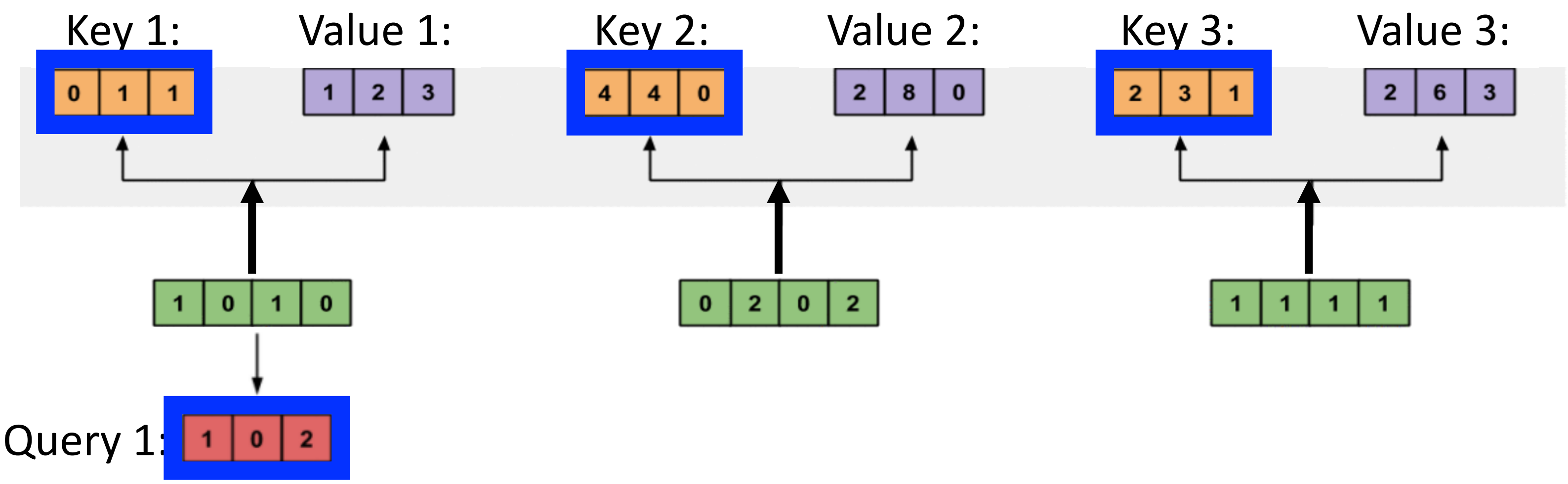
Computing Self-Attention: Example

Why dot product? Indicates similarity of two vectors

- Match = 1 (i.e., cos(0))
- Opposites = -1 (i.e., cos(180))



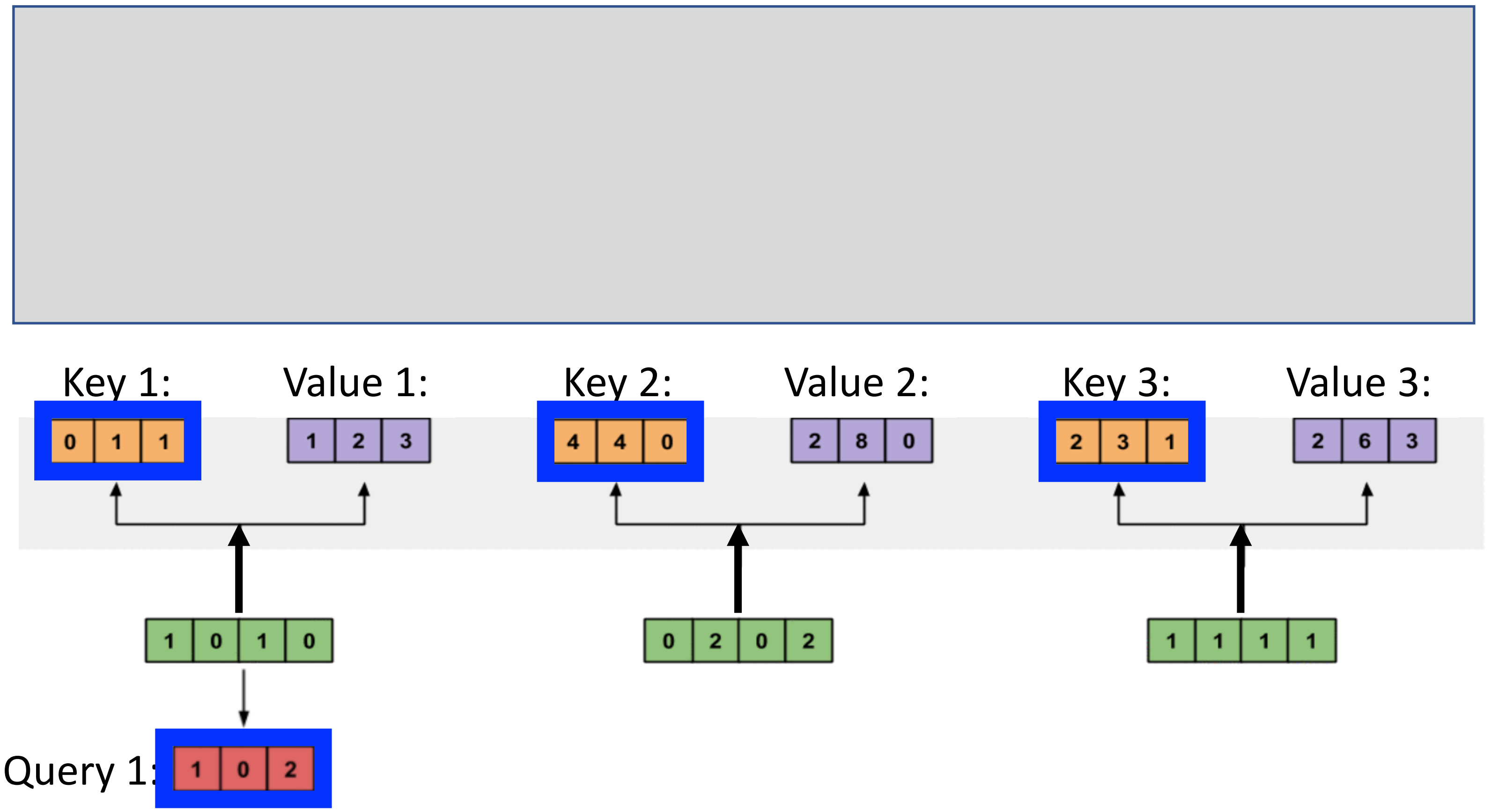
https://towardsdatascience.com/self-attention-5b95ea164f61



https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

Computing Self-Attention: Example

Can use similarity measures other than the dot product



https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

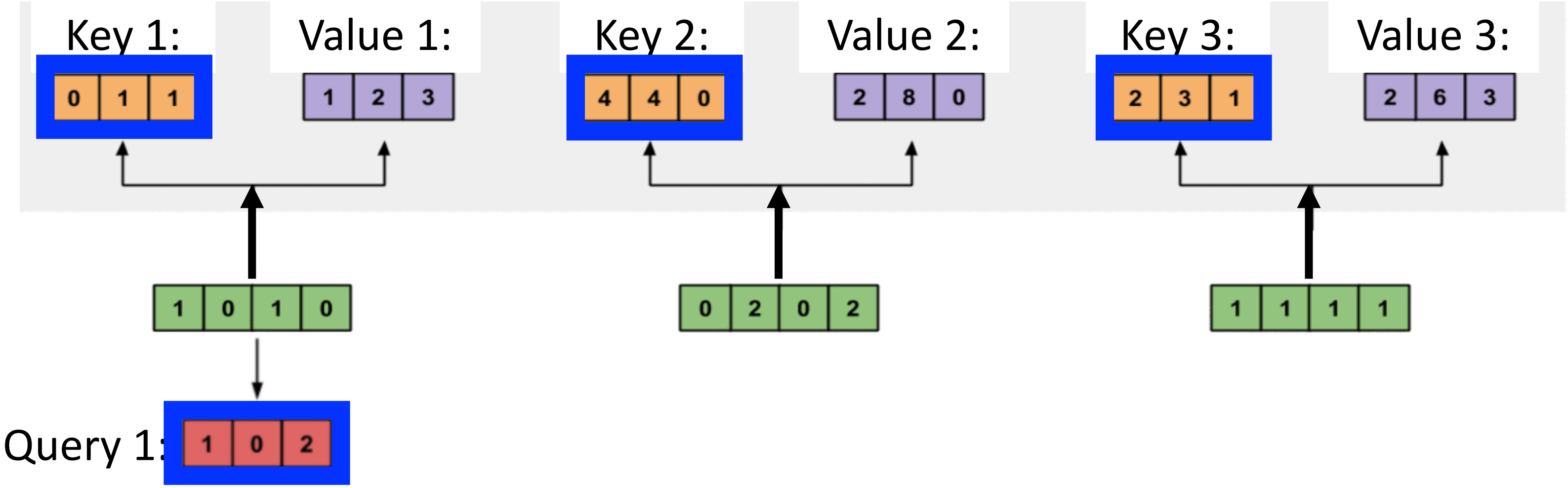
Computing Self-Attention: Example

Attention weights: softmax scores for all inputs to quantify each token's relevance; e.g.,

= softmax([2, 4, 4])

= [0.0, 0.5, 0.5])

Note: softmax doesn't return 0, but can arise from rounding



To which input(s) is input 1 **least** related?

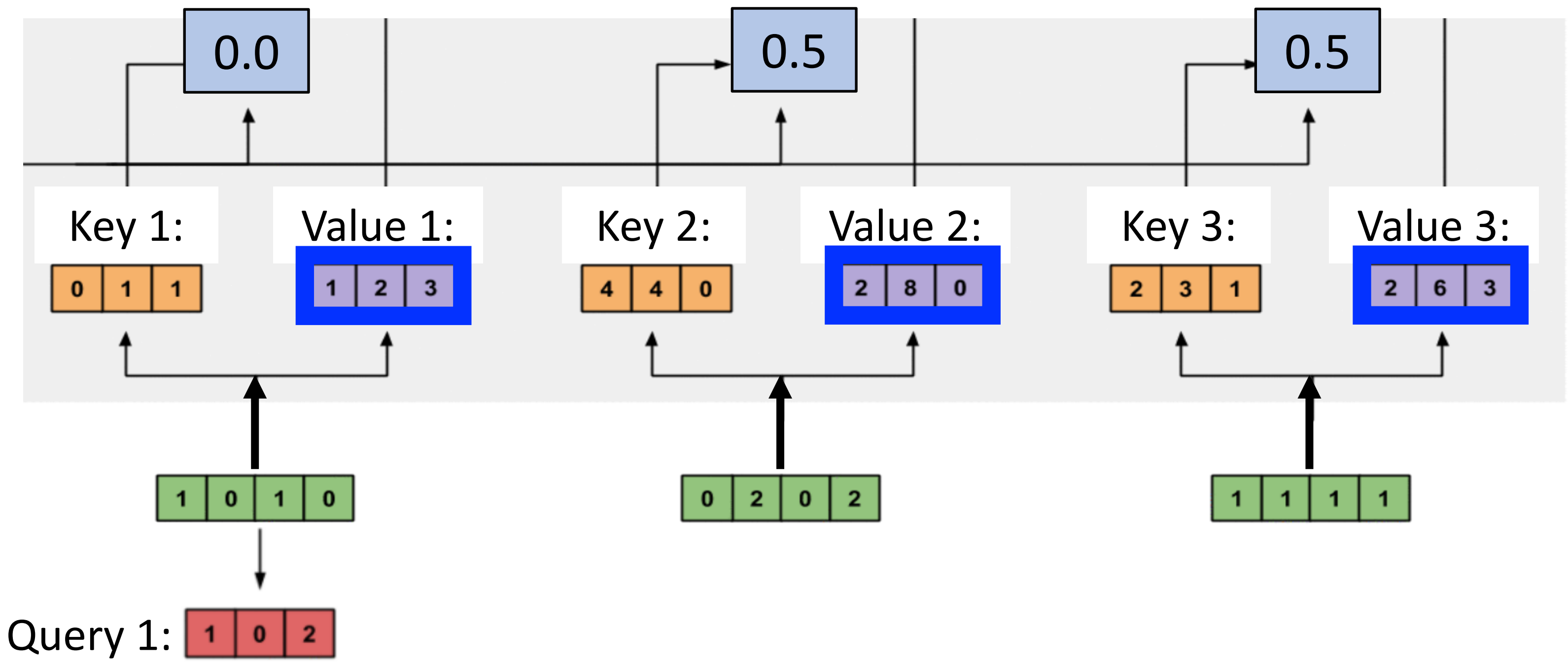
To which input(s) is input 1 **most** related?

https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

Computing Self-Attention: Example

Compute new representation of input token that reflects entire input:

1. Attention weights x Values



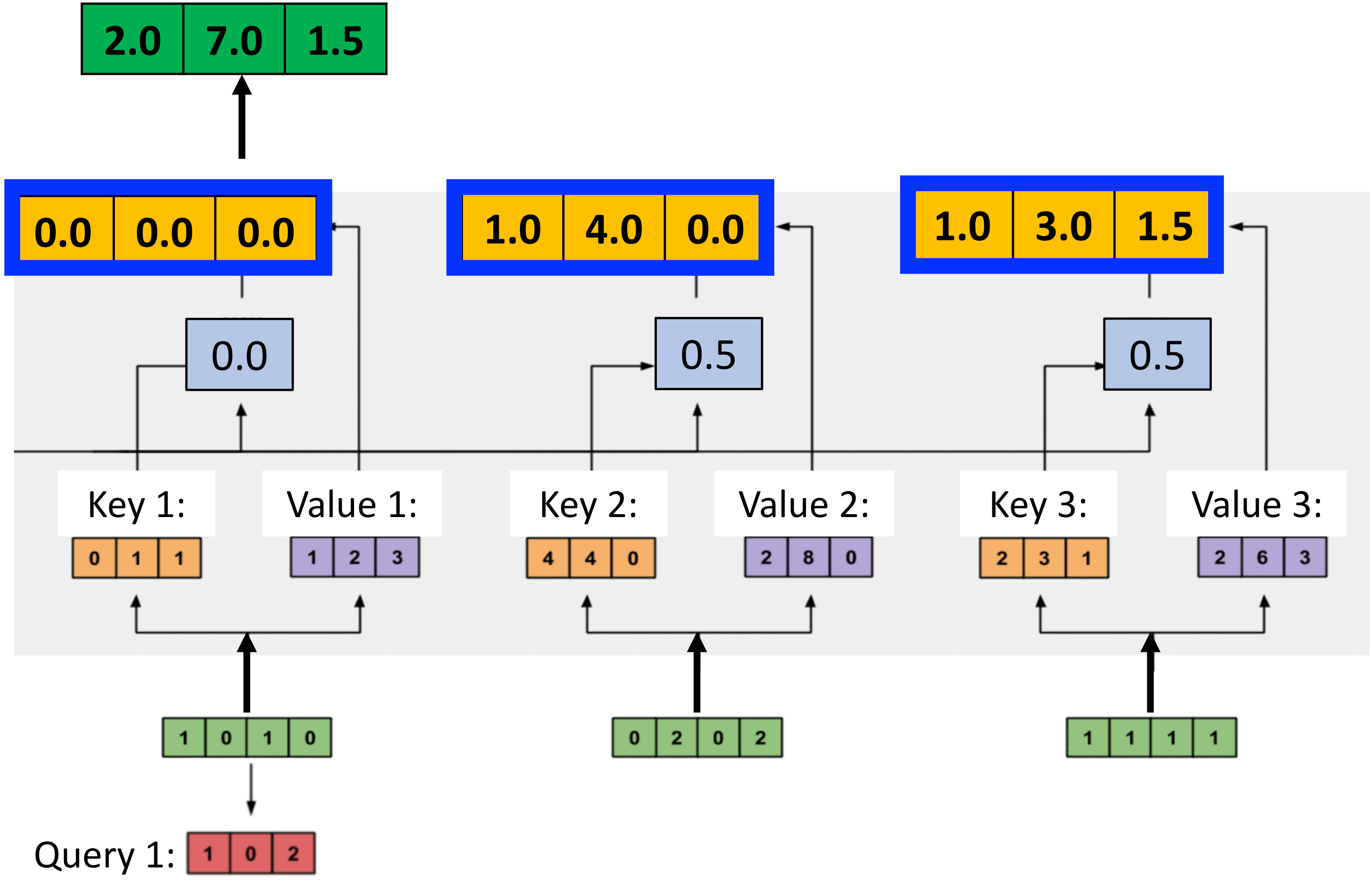
https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

Computing Self-Attention: Example

Compute new representation of input token that reflects entire input:

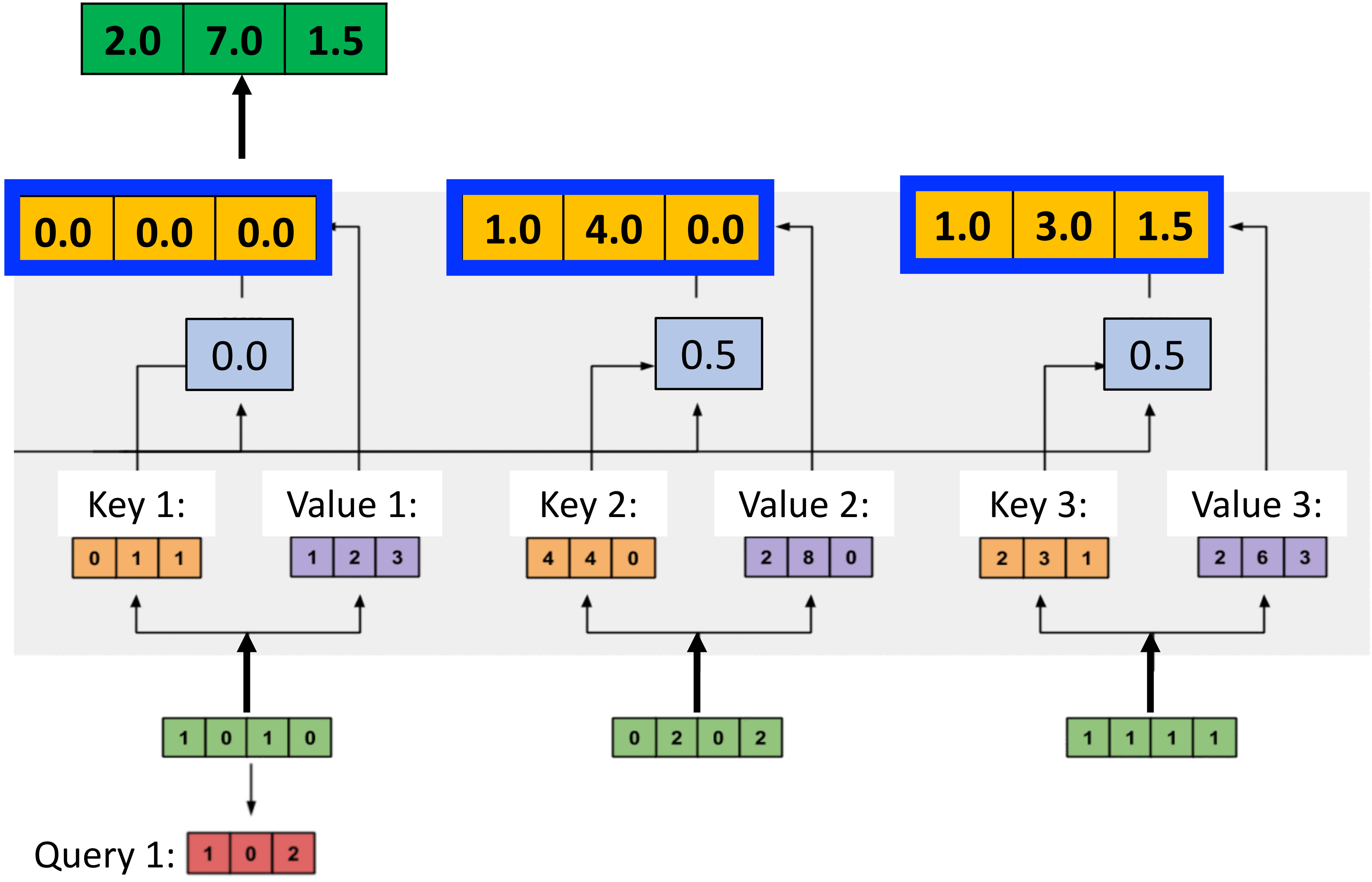
1. Attention weights x Values

2. Sum all weighted vectors



https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

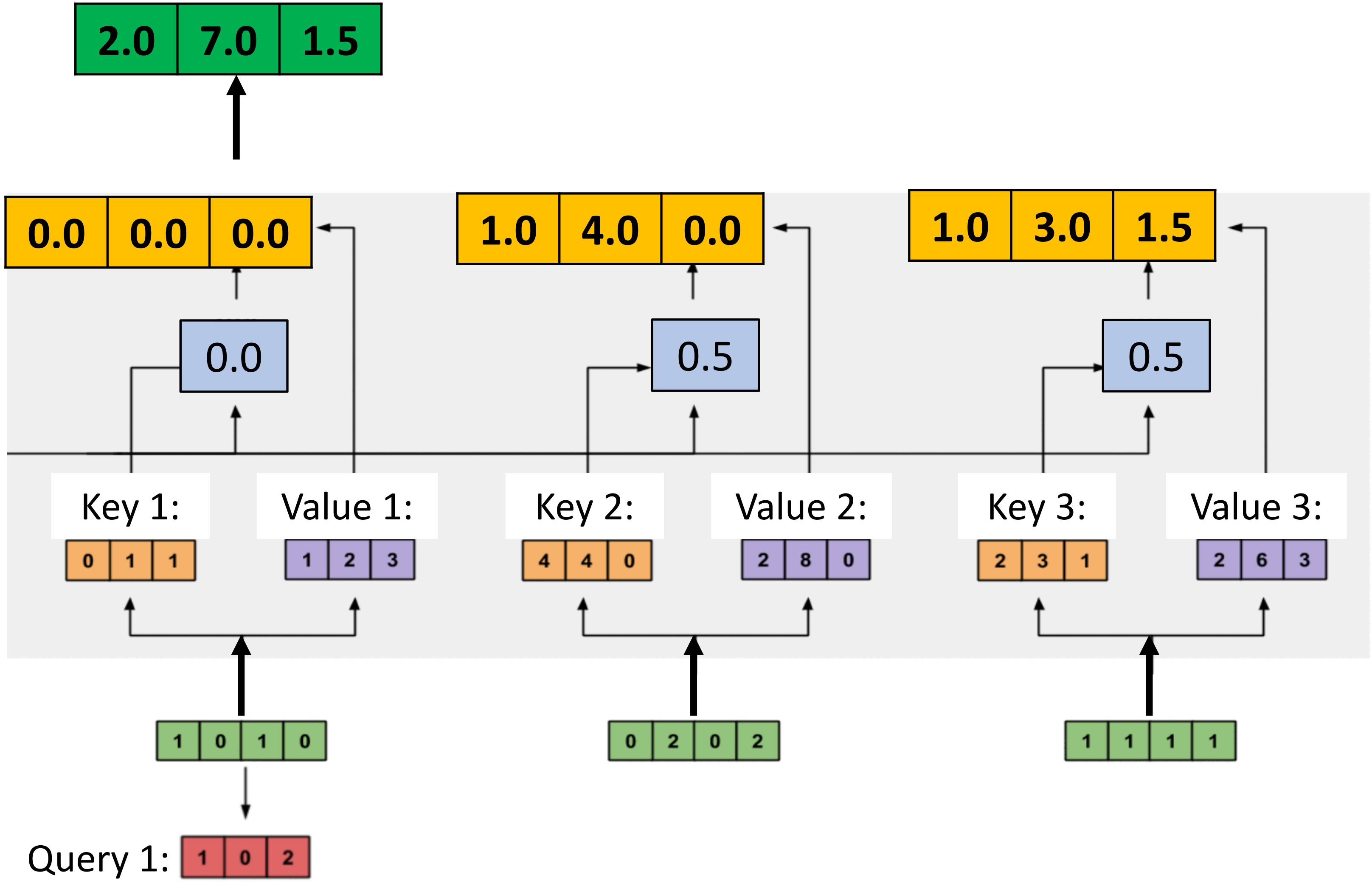
Computing Self-Attention: Example



Attention weights amplify input representations (values) that we want to pay attention to and repress the rest

https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

Computing Self-Attention: Example

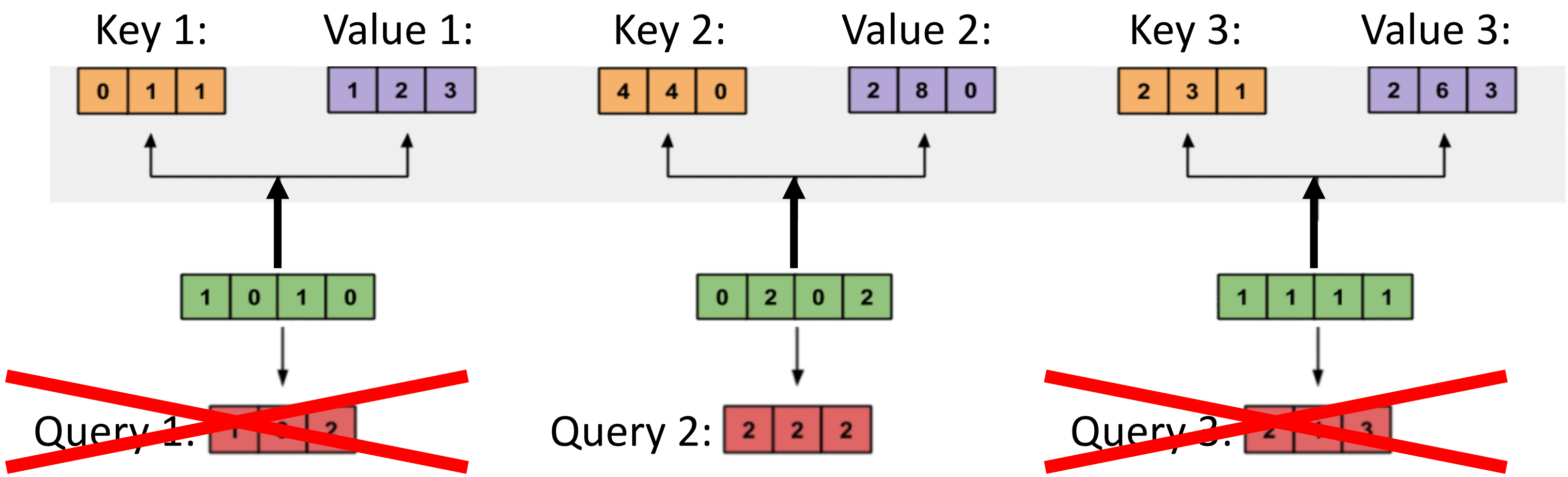


Attention weights amplify input representations (values) that we want to pay attention to and repress the rest

https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

Computing Self-Attention: Example

Repeat the same process for each remaining input token

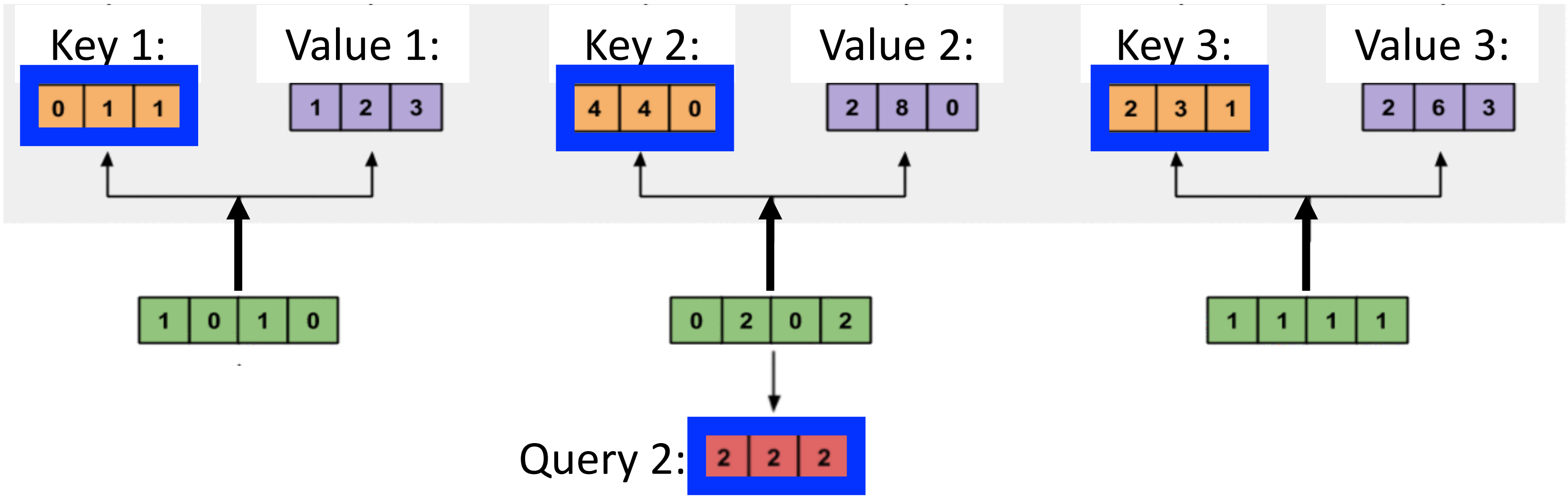


https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

Computing Self-Attention: Example

1. Compute attention weights
- Softmax resulting 3 scores from query x keys

To which input(s) is input 2 most related?

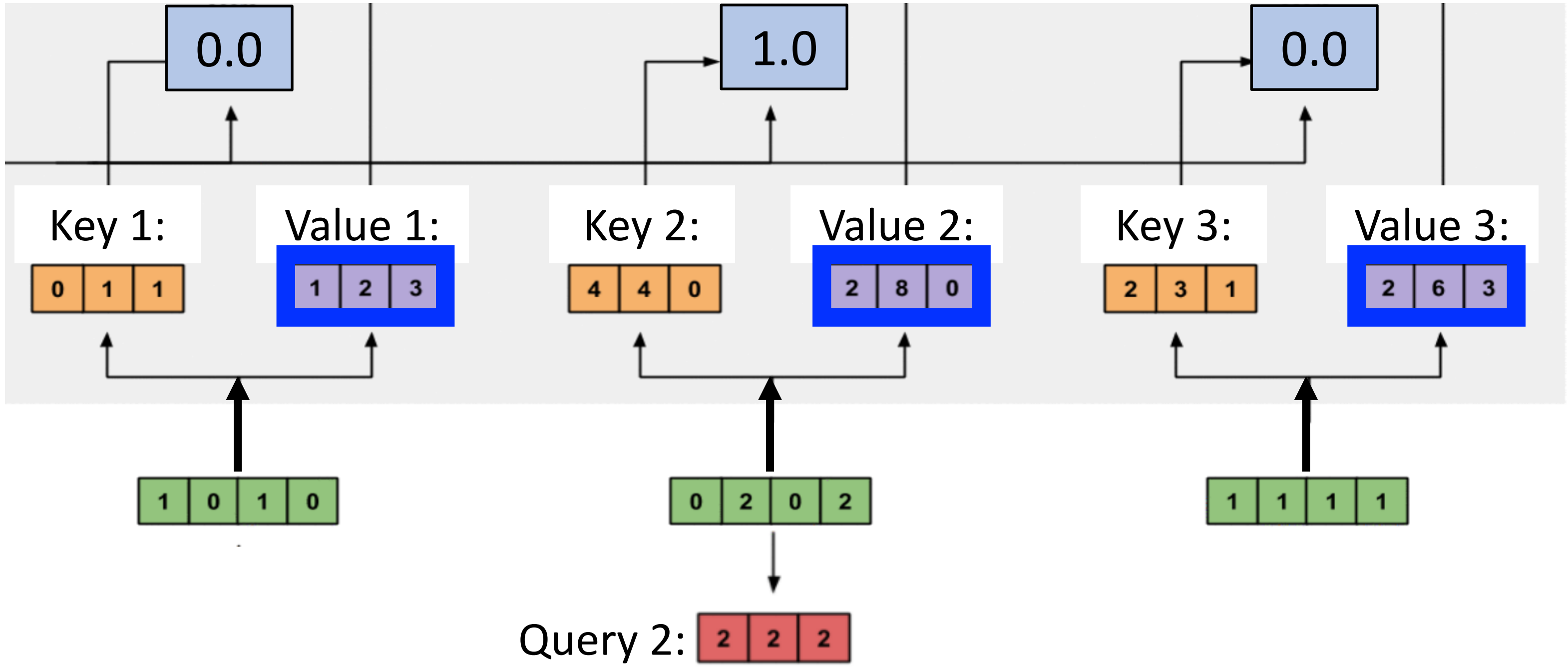


https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

Computing Self-Attention: Example

1. Compute attention weights
   - Softmax resulting 3 scores from query x keys

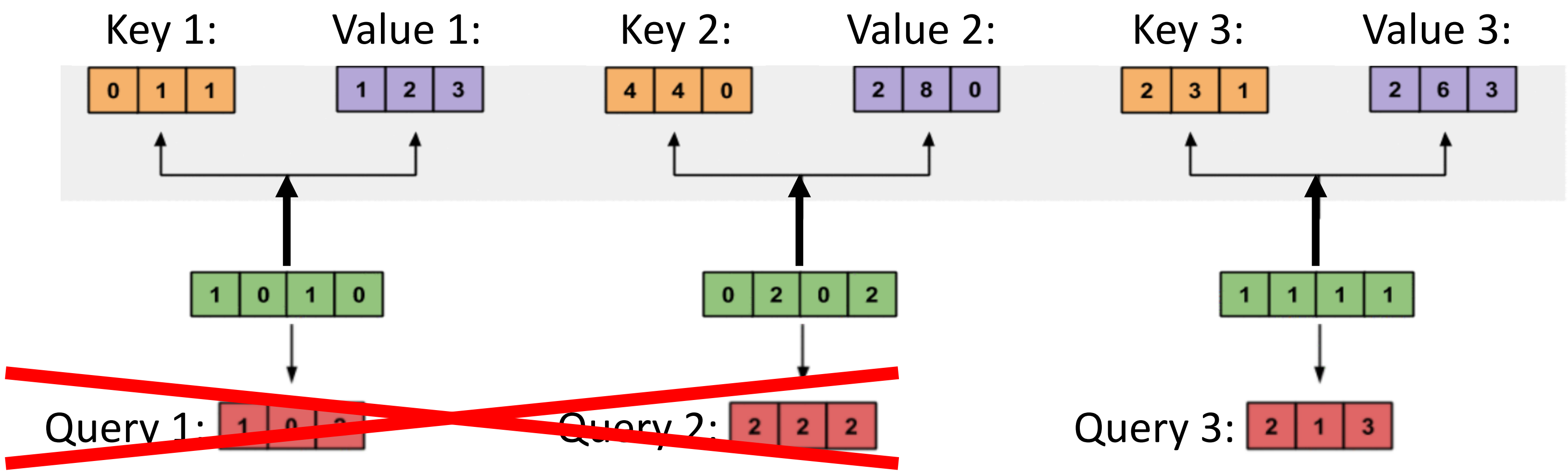
2. Compute weighted sum of values using attention scores



https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

Computing Self-Attention: Example

Repeat the same process for each remaining input token



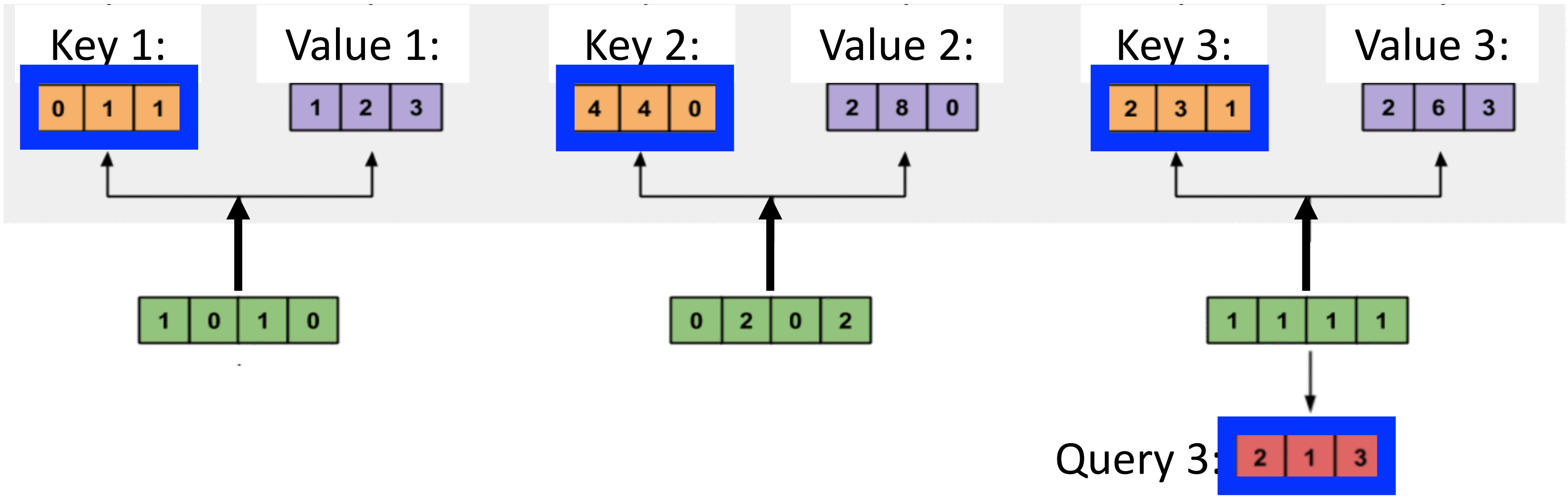
https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

Computing Self-Attention: Example

1. Compute attention weights

- Softmax resulting 3 scores from query x keys

To which input(s) is input 3 most related?

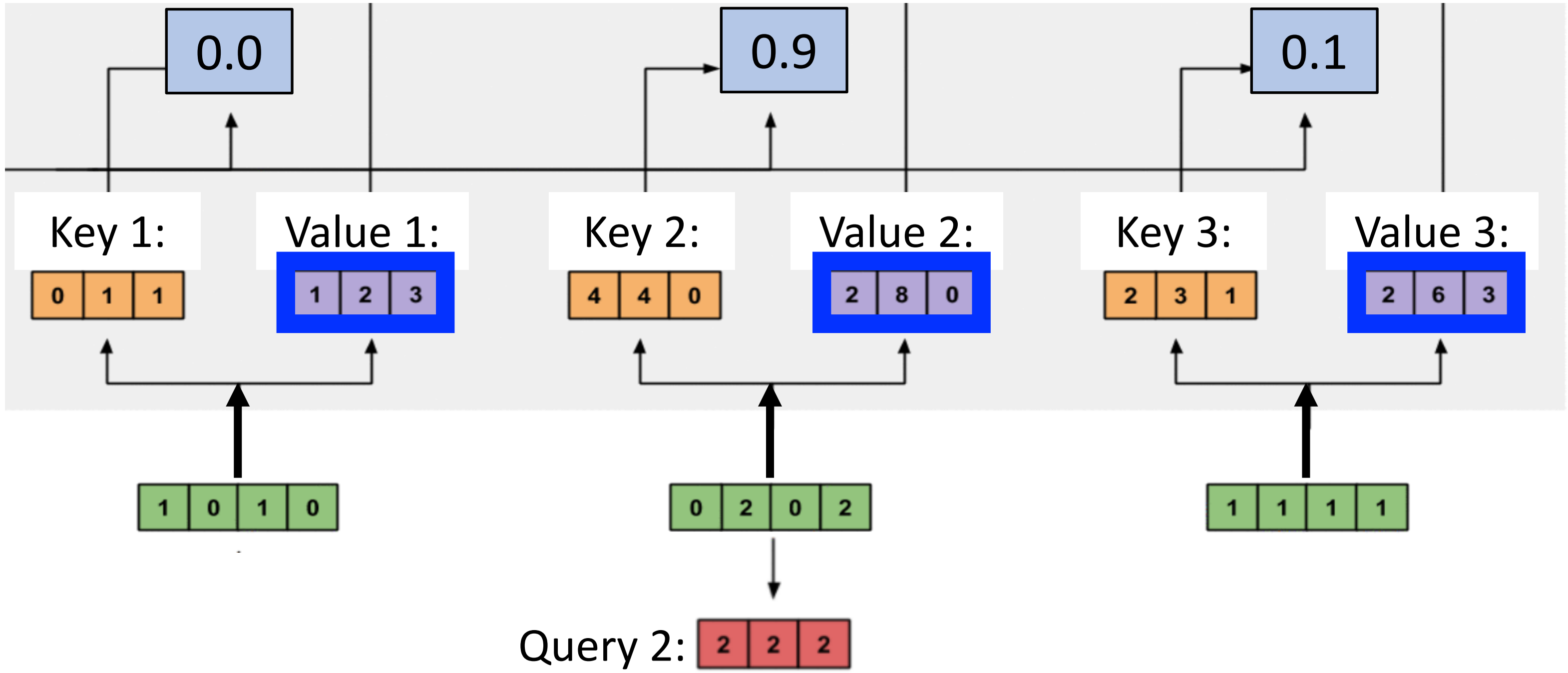


https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

Computing Self-Attention: Example

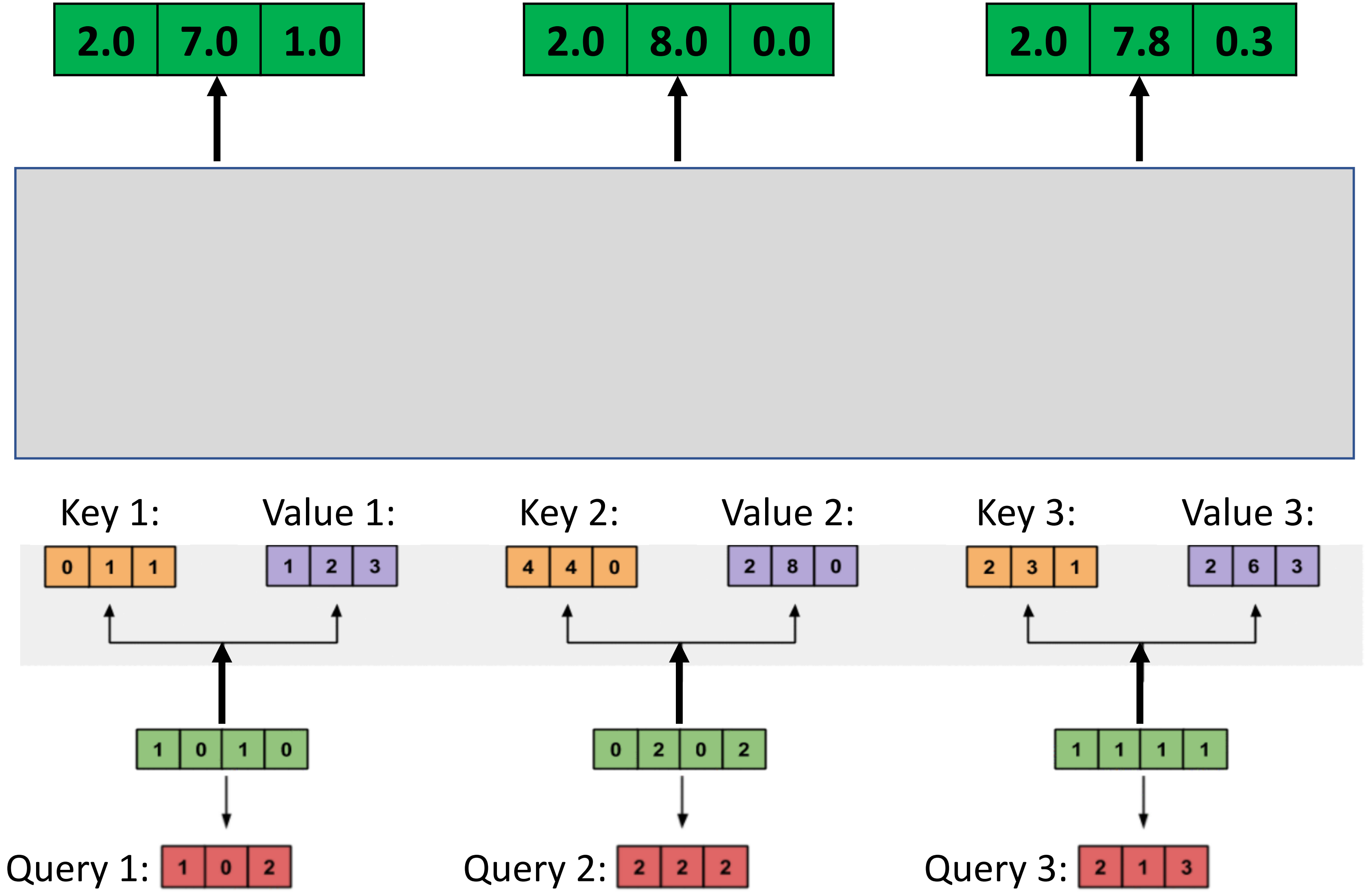
1. Compute attention weights
- Softmax resulting 3 scores from query x keys

2. Compute weighted sum of values using attention scores



https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

Computing Self-Attention: Example

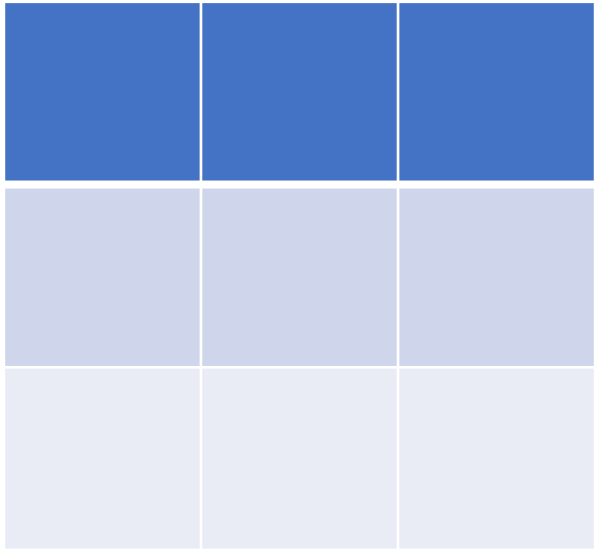


https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

Problem: Self-Attention's Computational Expense

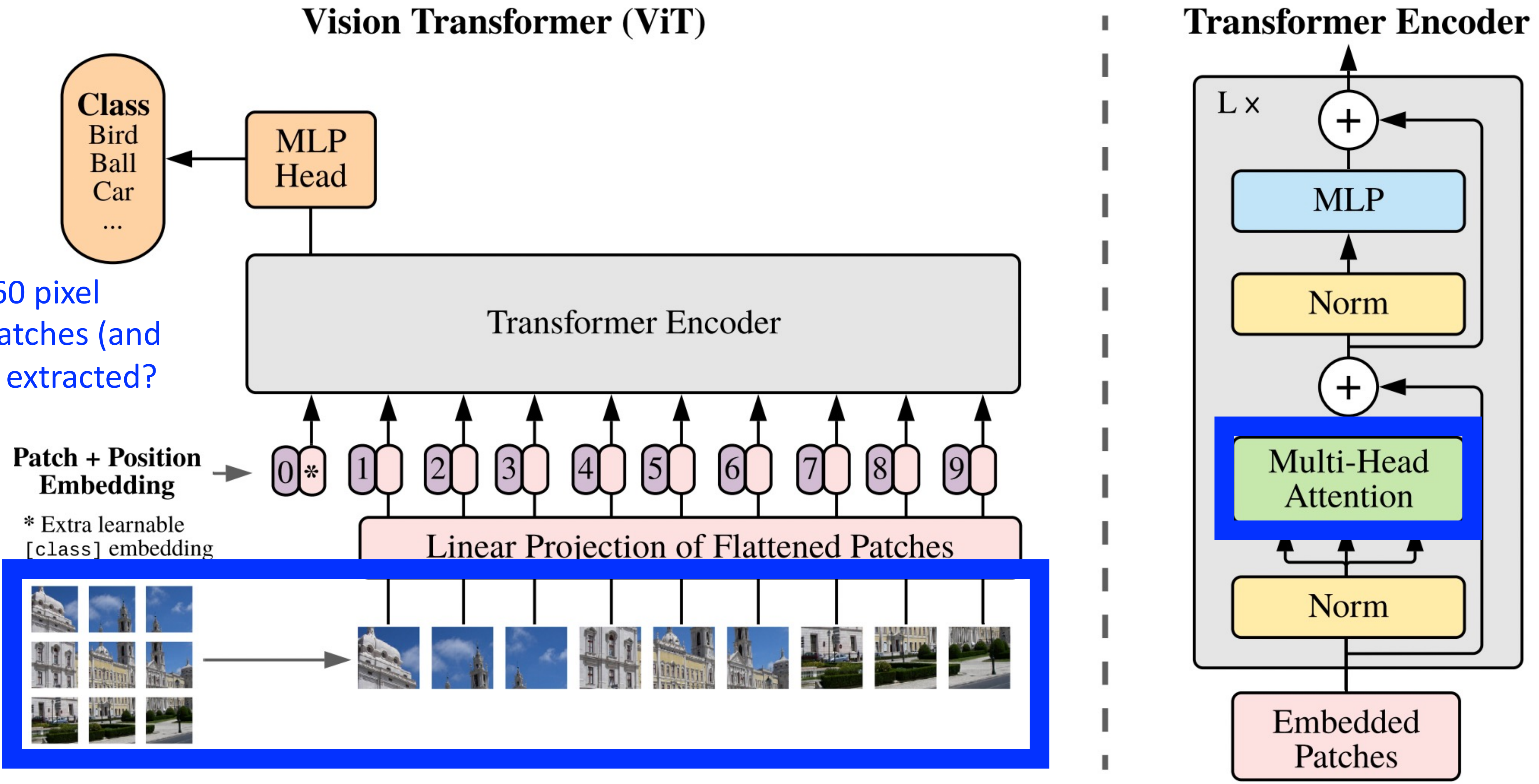
e.g., instead of using 3x3 image, what if a 1920 x 1080 image was used? How many self-attention computations would be needed?

- $(1920 \times 1080)^2$ = 4,299,816,960,000 (i.e., ~4.3 trillion)



Quadratic cost of self-attention in transformers is often impractical for pixels!

ViT Solution: Input Patches Instead of Pixels



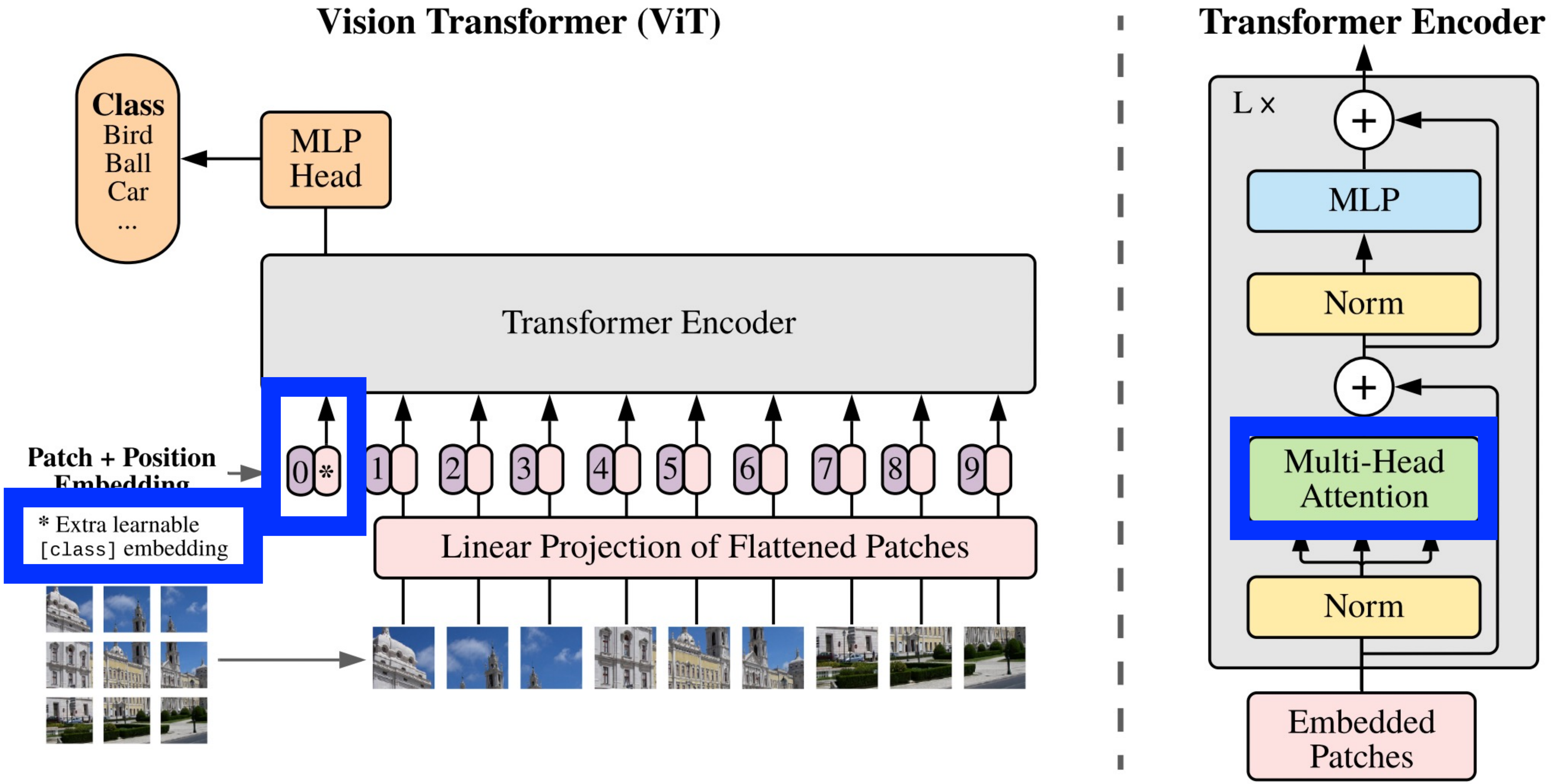
Assuming a 160 x 160 pixel image, how many patches (and so inputs) would be extracted?
- 10

Image decomposed into 16x16 non-overlapping patches (example simplified); representations include "flattened" and ResNet features

Dosovitskiy et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR 2021.

ViT Solution: Use [CLS] for Image Classification

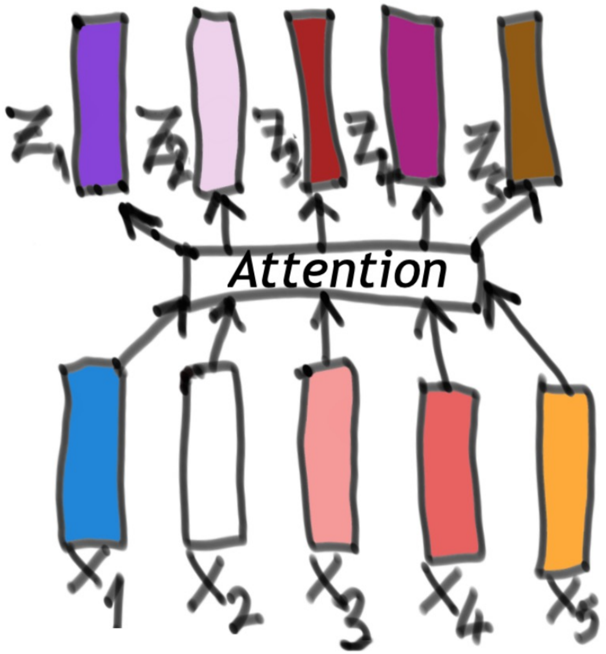
[CLS] token represents entire image



Dosovitskiy et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR 2021.

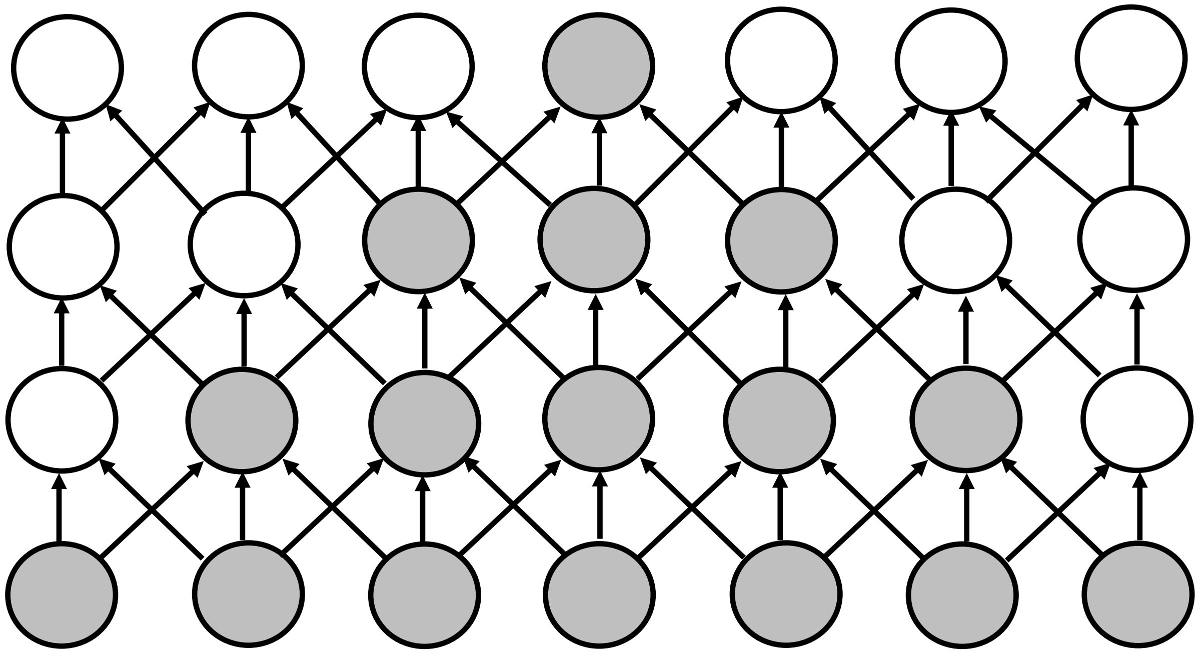
Transformers vs CNNs

Self-attention: each layer has a global receptive field



https://towardsdatascience.com/self-attention-5b95ea164f61

Convolutional layers: deeper layers have increasingly more global receptive fields



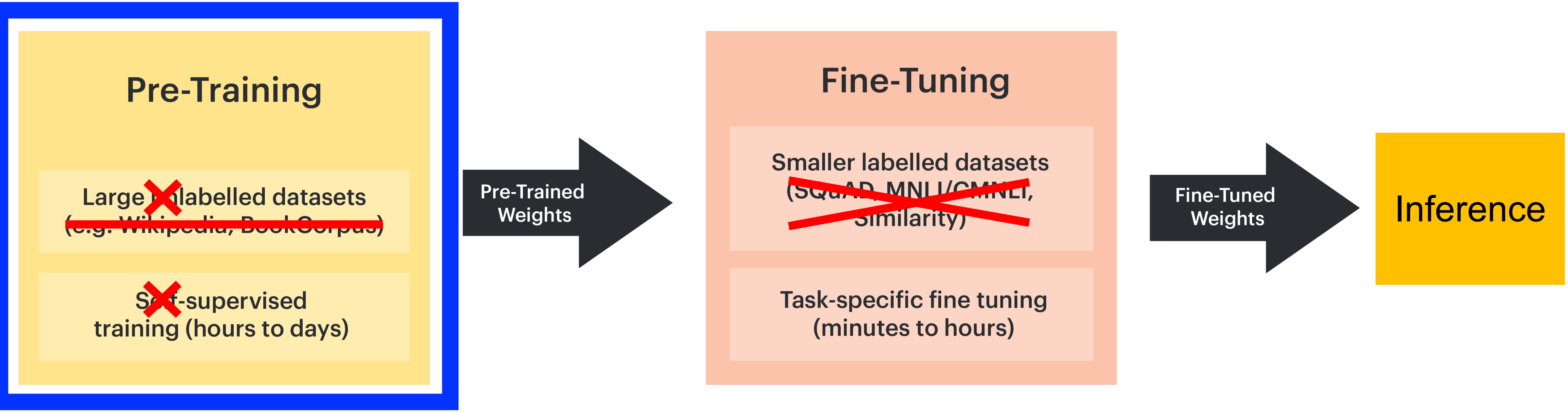
https://www.deeplearningbook.org/contents/convnets.html

ViT: Key Ingredients for Success

- Transformer architecture (embeds self-attention)

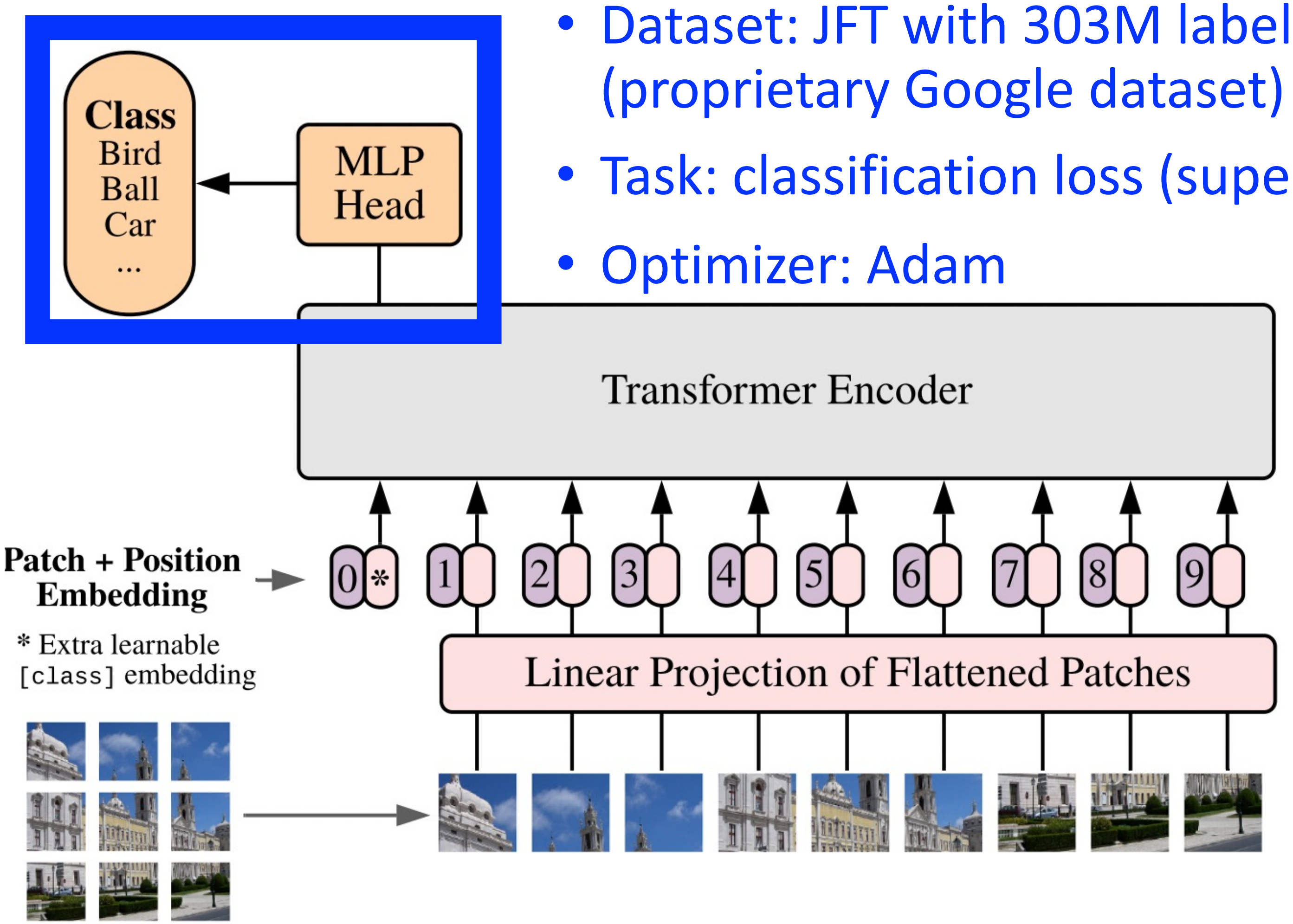
- Pre-training with massive amounts of data

Approach



https://docs.graphcore.ai/projects/bert-training/en/latest/bert.html

ViT Pre-Training

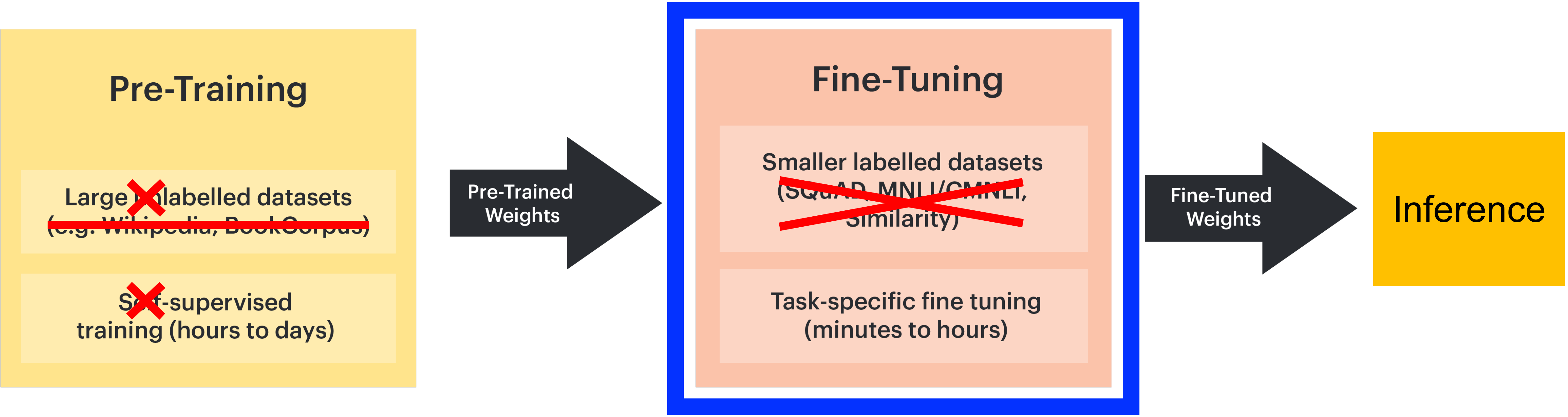


- Dataset: JFT with 303M labeled images (proprietary Google dataset)
- Task: classification loss (supervised)
- Optimizer: Adam

* Note: research also is exploring how smaller training datasets can be effective; e.g., data efficient image transformers (DeiT) from "Training data-efficient image transformers & distillation through attention"

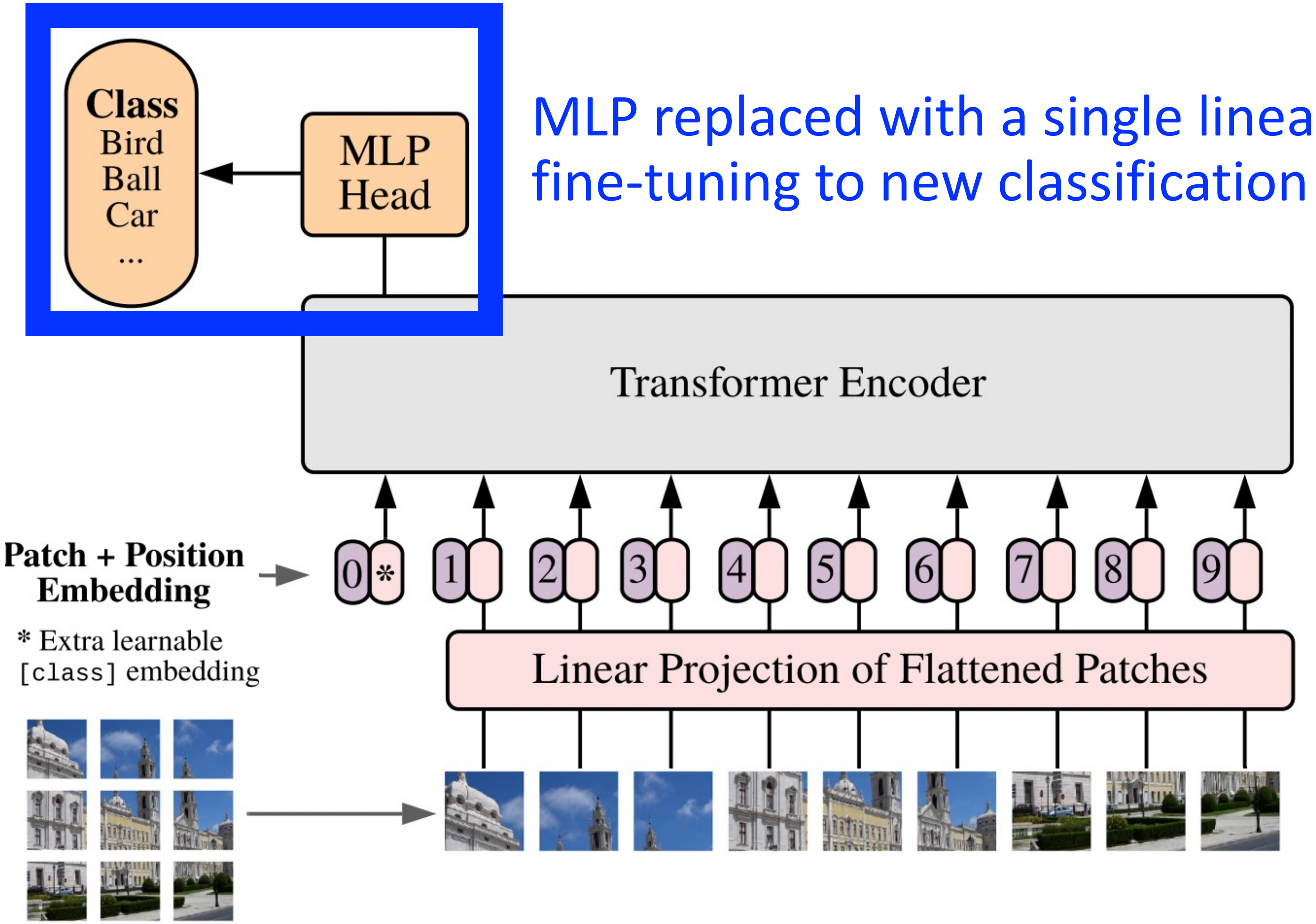
Dosovitskiy et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR 2021.

ViT Training



https://docs.graphcore.ai/projects/bert-training/en/latest/bert.html

ViT Fine-Tuning: Other Image Classification Tasks



MLP replaced with a single linear layer when fine-tuning to new classification categories

Dosovitskiy et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR 2021.

Experimental Findings and Closing Question

ViT achieved strong results on all tested image classification datasets, prompting the question of whether transformers’ success would generalize to other vision tasks.

Dosovitskiy et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR 2021.

Today’s Topics

- Motivation
- ViT
- Swin Transformer
- Discussion

Why Swin Transformer?

Named after the proposed technique: Shifted Windows

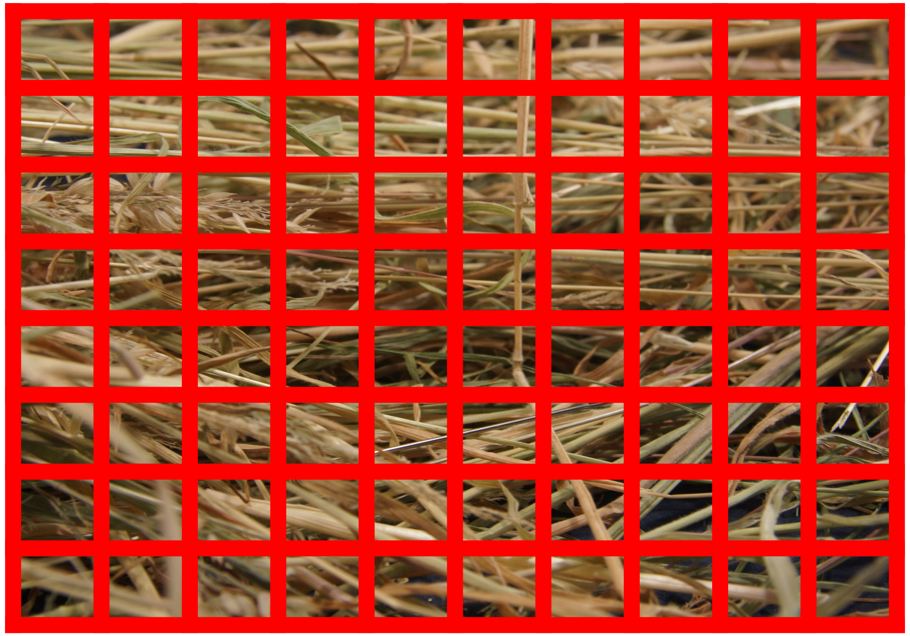
Liu et al. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. ICCV 2021.

Novelty

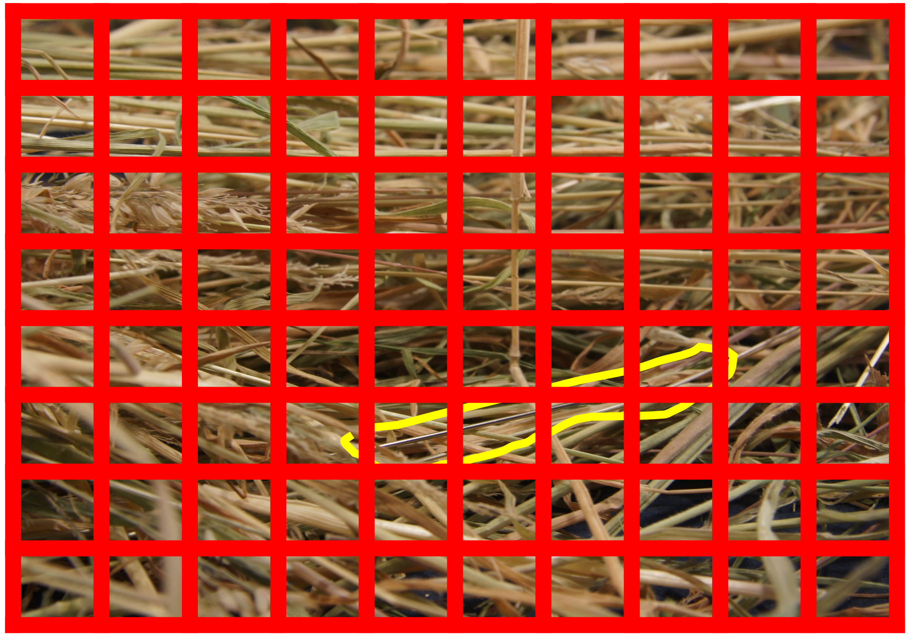
- First paper to demonstrate how a transformer "backbone" can generalize to diverse vision tasks, yielding state-of-the-art results for object detection and semantic segmentation (aka – dense prediction problems) and strong results for image classification

Why ViT Is Inadequate for Dense Prediction

Image classification
- What image label is predicted?
- "Big" patches are sufficient



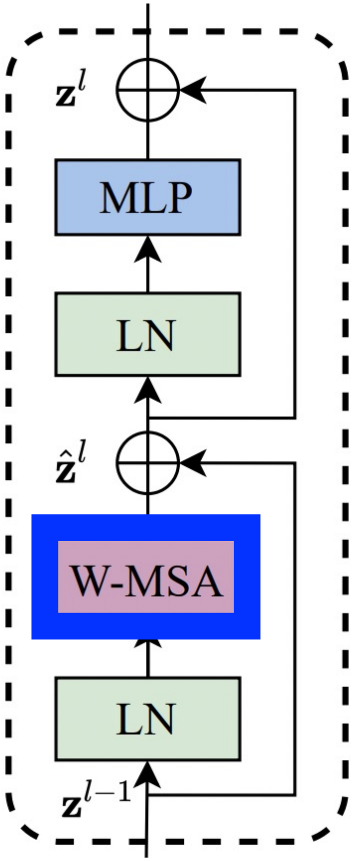
Object detection/Semantic segmentation
- What pixel label(s) are predicted?
- "Big" patches are insufficient



Issue: quadratic expense of self-attention necessitated 16 x 16 patches, but this can be too large for pixel-level predictions (e.g., locating needle in a haystack)

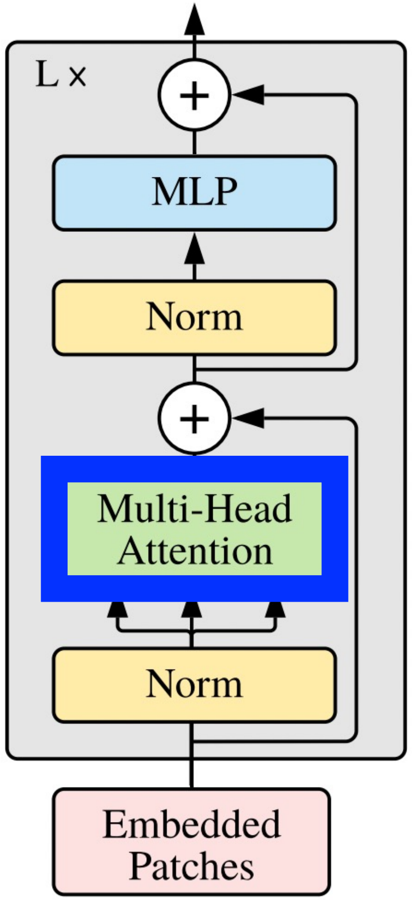
Key Idea of Swin: Modify Self-Attention Module

Swin Transformer



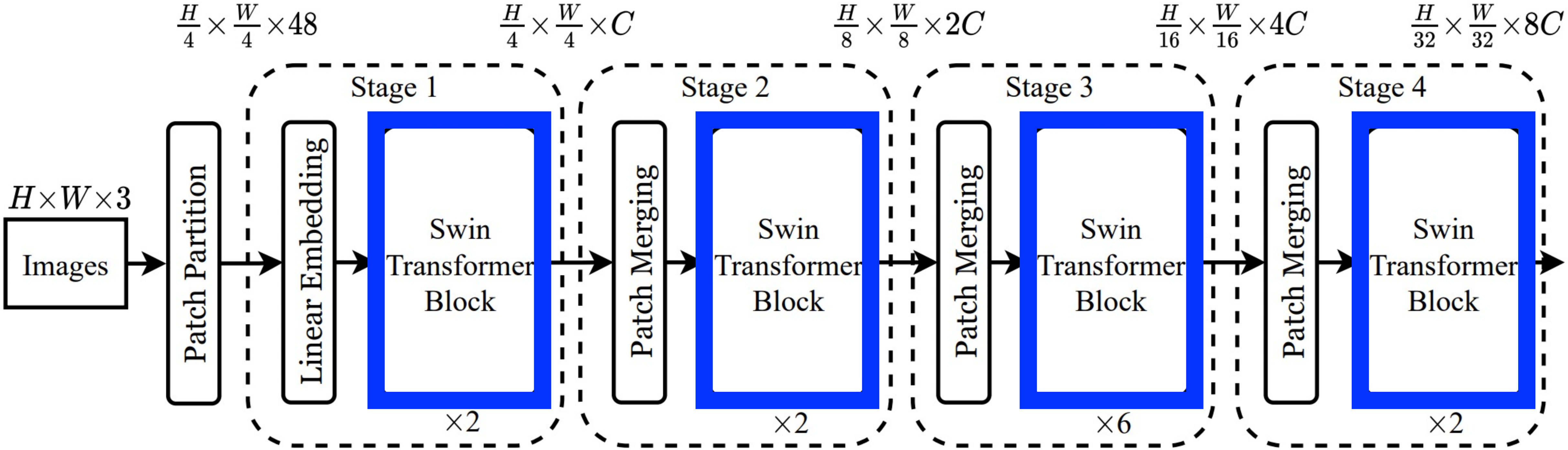
Liu et al. ICCV 2021.

ViT



Dosovitskiy et al. ICLR 2021.

Architecture

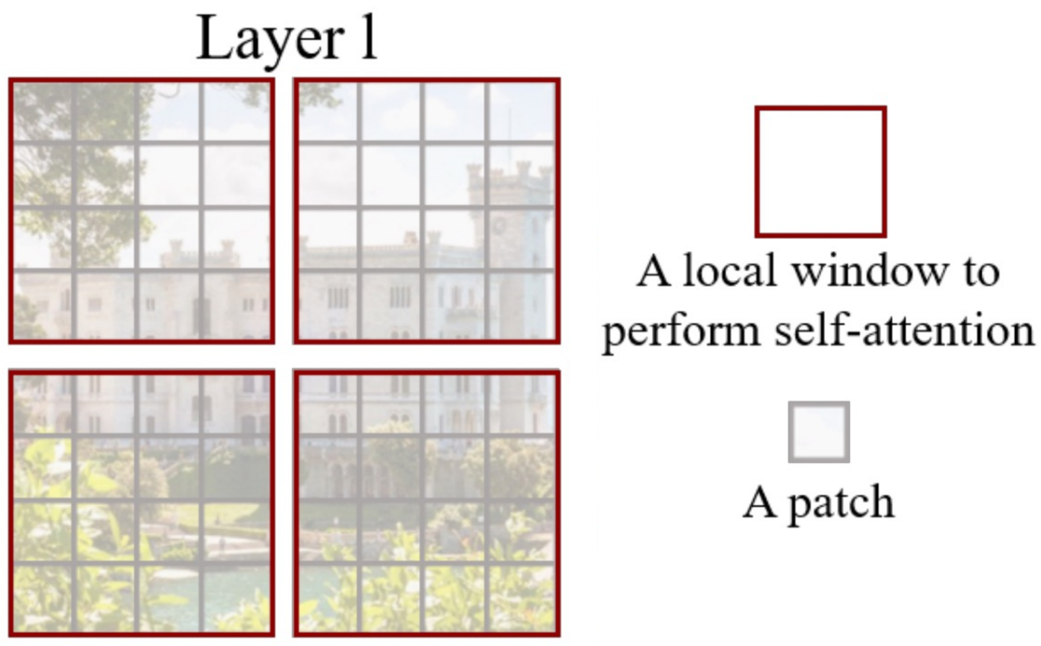


Contains a series of modified self-attention modules

Liu et al. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. ICCV 2021.

Key Idea: Modified Self-Attention Module

Applies self-attention only between the fixed number of patches in each window to capture fine-grained details (i.e., limited to local context)



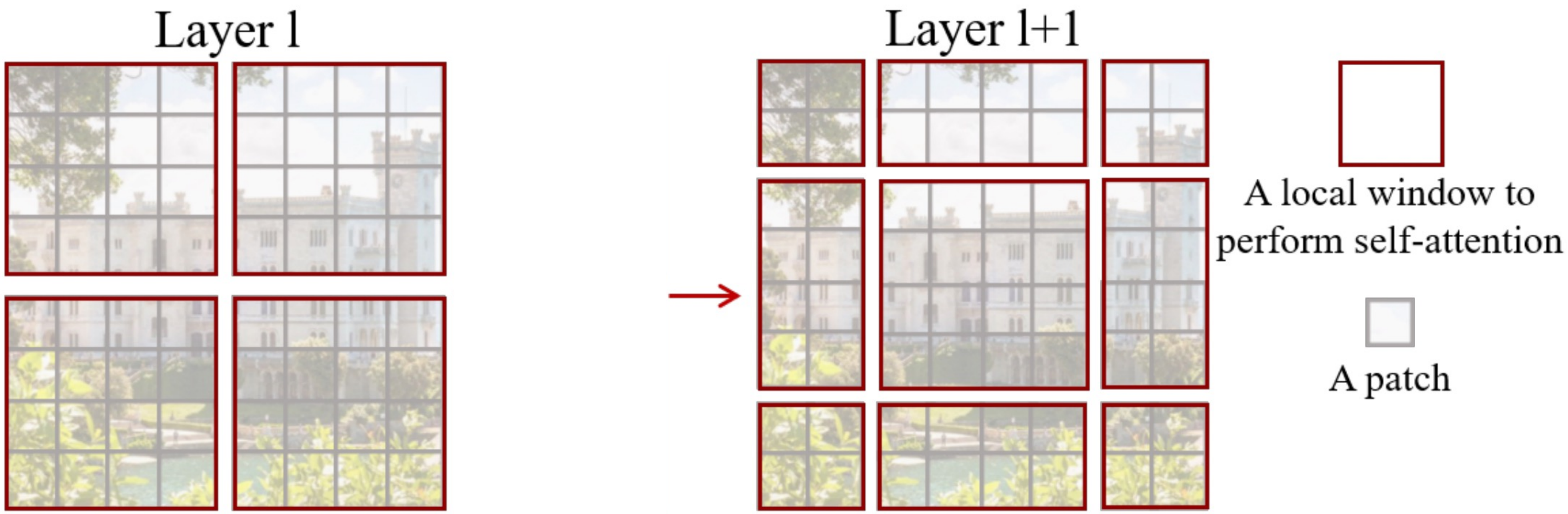
What is the computational complexity?
- Linear based on fixed patch number chosen per window rather than quadratic based on number of input patches

Liu et al. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. ICCV 2021.

Key Idea: Modified Self-Attention Module

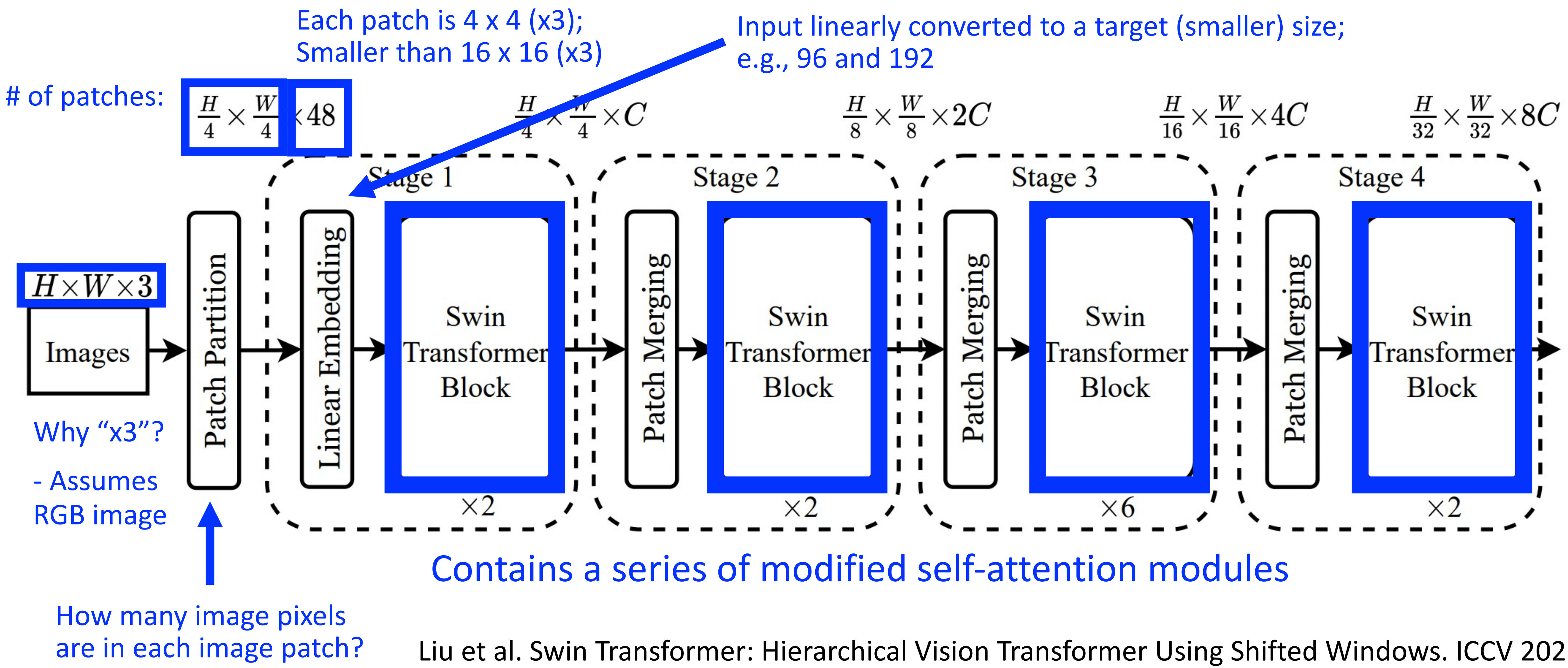
Applies self-attention only between the fixed number of patches in each window to capture fine-grained details (i.e., limited to local context)

In each subsequent layer, windows shifted to infuse global context by enabling communication between previously non-communicative neighboring patches



Liu et al. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. ICCV 2021.

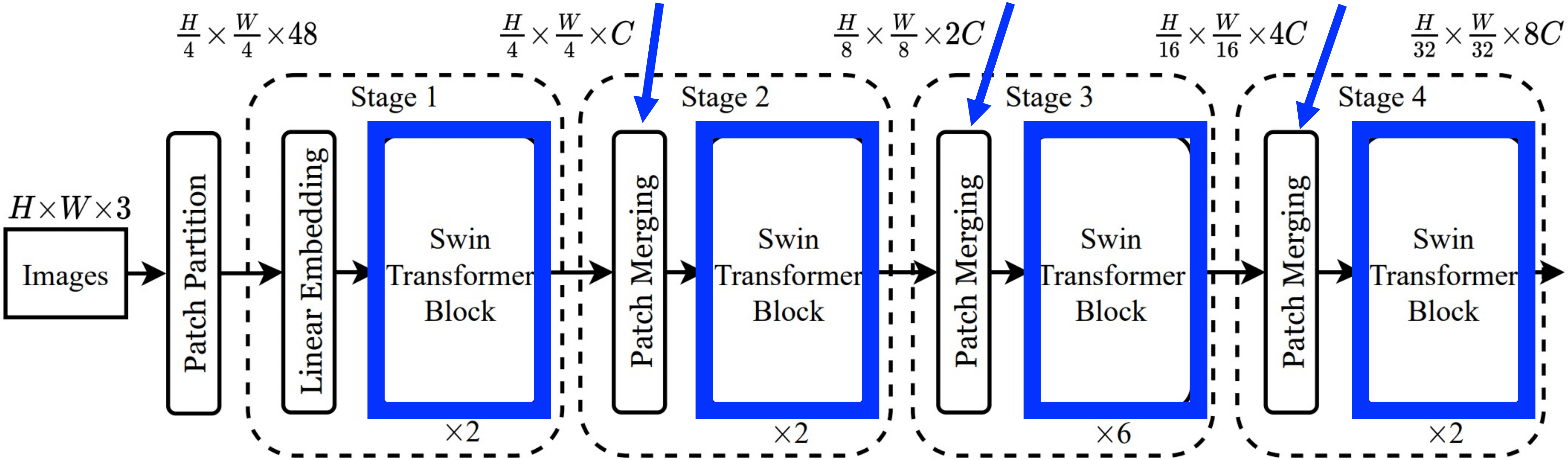
Architecture



Liu et al. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. ICCV 2021.

Architecture

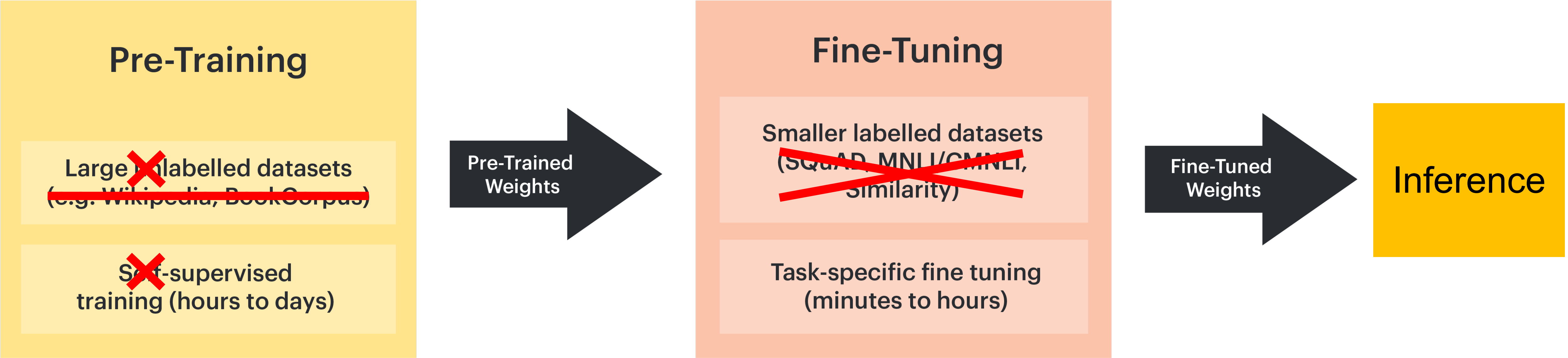
Neighboring patches merged into increasingly bigger patches (mimics convolutional layers); this hierarchical design also increases global context to better support visual content at different scales! (*output feature maps match resolution of common CNNs, e.g., VGG & ResNet*)



Contains a series of modified self-attention modules at different resolutions

Liu et al. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. ICCV 2021.

Image Classification: Outperformed ViT



Used ImageNet-22K dataset
(14.2M images and 22K classes)

Used ImageNet-1K dataset

https://docs.graphcore.ai/projects/bert-training/en/latest/bert.html

Dense Prediction: State-of-the Art Results

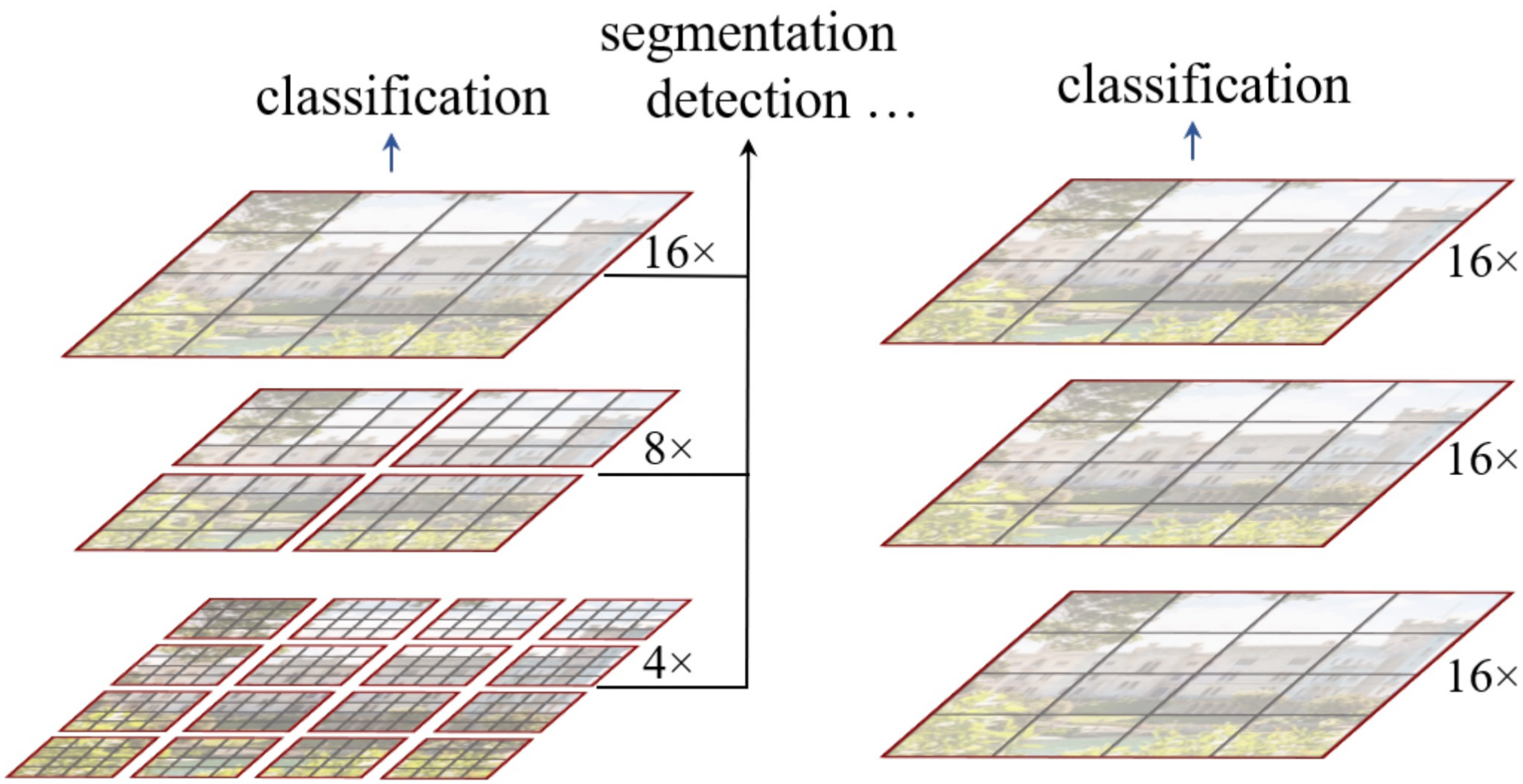
Four object detection algorithms tested on COCO 2017 with three “backbone” sources:
- ResNe(X)t
- DeiT
- Swin: was consistently top-performer

UperNet semantic segmentation algorithm tested on ADE20K with two “backbone” sources:
- DeiT
- Swin: was consistently top-performer

Liu et al. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. ICCV 2021.

Summary

- Reduces complexity, by applying self-attention to a fixed number of patches within each window to capture fine-grained details and then infuses global context with shifted windows
- Uses a hierarchical design by merging patches in deep layers to capture visual entities at different scales
- Outperforms ViT on image classification and was state-of-the-art for object detection and semantic segmentation



(a) Swin Transformer (ours)

(b) ViT

Liu et al. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. ICCV 2021.

Authors’ Conclusions

“It is our belief that a unified architecture across computer vision and natural language processing could benefit both fields.”

Today’s Topics

- Motivation
- ViT
- Swin Transformer
- Discussion

Transformers vs Convolutional Neural Networks

- Note that there remains an open debate about which architecture to prefer
- Ideas from both architectures are being infused into each other; e.g.,
  - https://arxiv.org/pdf/2201.03545.pdf
  - https://openaccess.thecvf.com/content/CVPR2023/papers/Wang_InternImage_Exploring_Large-Scale_Vision_Foundation_Models_With_Deformable_Convolutions_CVPR_2023_paper.pdf
  - https://proceedings.neurips.cc/paper_files/paper/2022/file/5e0b46975d1bfe6030b1687b0ada1b85-Paper-Conference.pdf
  - https://arxiv.org/pdf/2207.13317.pdf
  - https://arxiv.org/pdf/2201.09792.pdf
- Benchmarks are comparing their robustness; e.g.,
  - https://arxiv.org/pdf/2207.11347.pdf
  - https://arxiv.org/pdf/2206.03452.pdf
  - https://proceedings.neurips.cc/paper_files/paper/2022/file/5ce3a49415f78db65a714b4f05c62f4e-Paper-Conference.pdf

Student Google Form

Today’s Topics

- Motivation
- ViT
- Swin Transformer
- Discussion


The End