# CS195: Computer Vision

Transformer
Vision Transformer (ViT)
Swin Transformer

Monday, September 30th, 2024

# Announcements

- **Grade posted**
  - In-class activities #1, #2, and #3
- **Notebook#3**
  - Due on **10/02 (Wednesday) by 11:59pm**
- **Quiz#1**
  - Will be released tomorrow

# Transformers

# Today's agenda

- Transformer
  - Transfer learning is possible
  - New type of network architecture besides convolutional neural network
- Vision Transformer (ViT)

# Transformers

- In 2017, a new mechanism is introduced for context learning called **attention mechanism**
  - more precisely, self-attention
- It takes less time to train advantage
- Transfer learning on a new task is possible advantage
- In subsequent years, it revolutionized the field of AI

**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

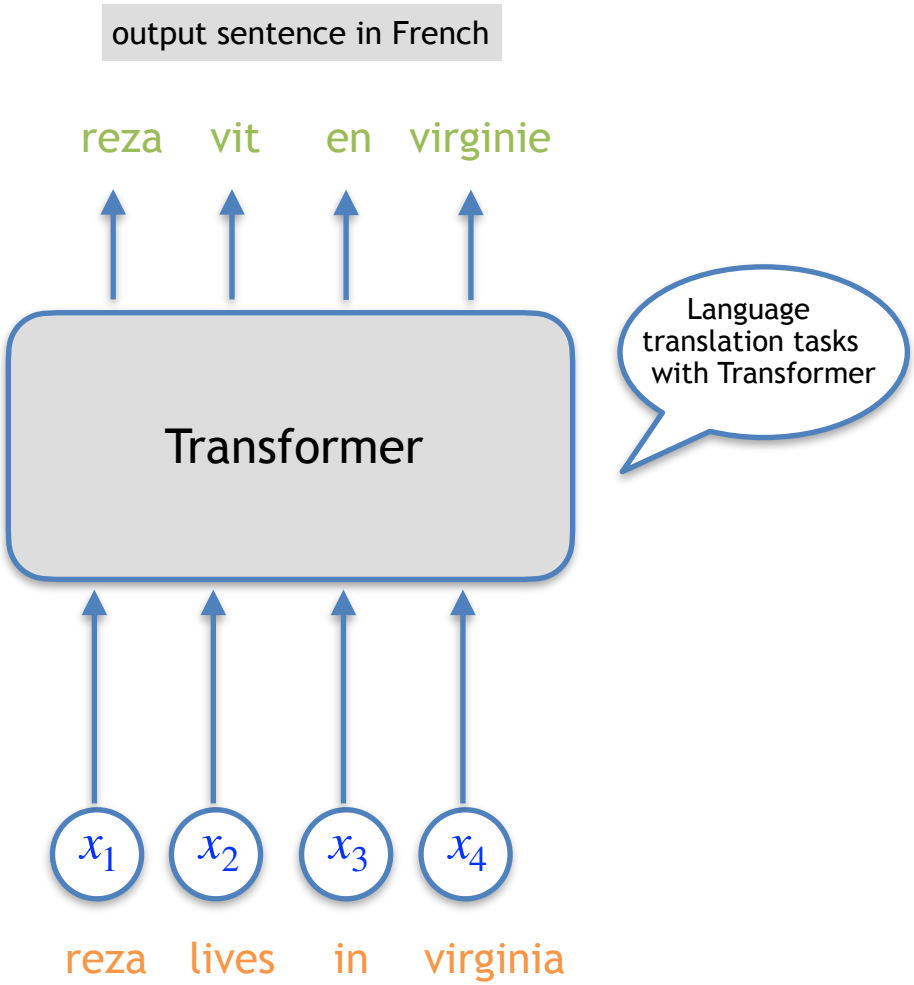


**Abstract**

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 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

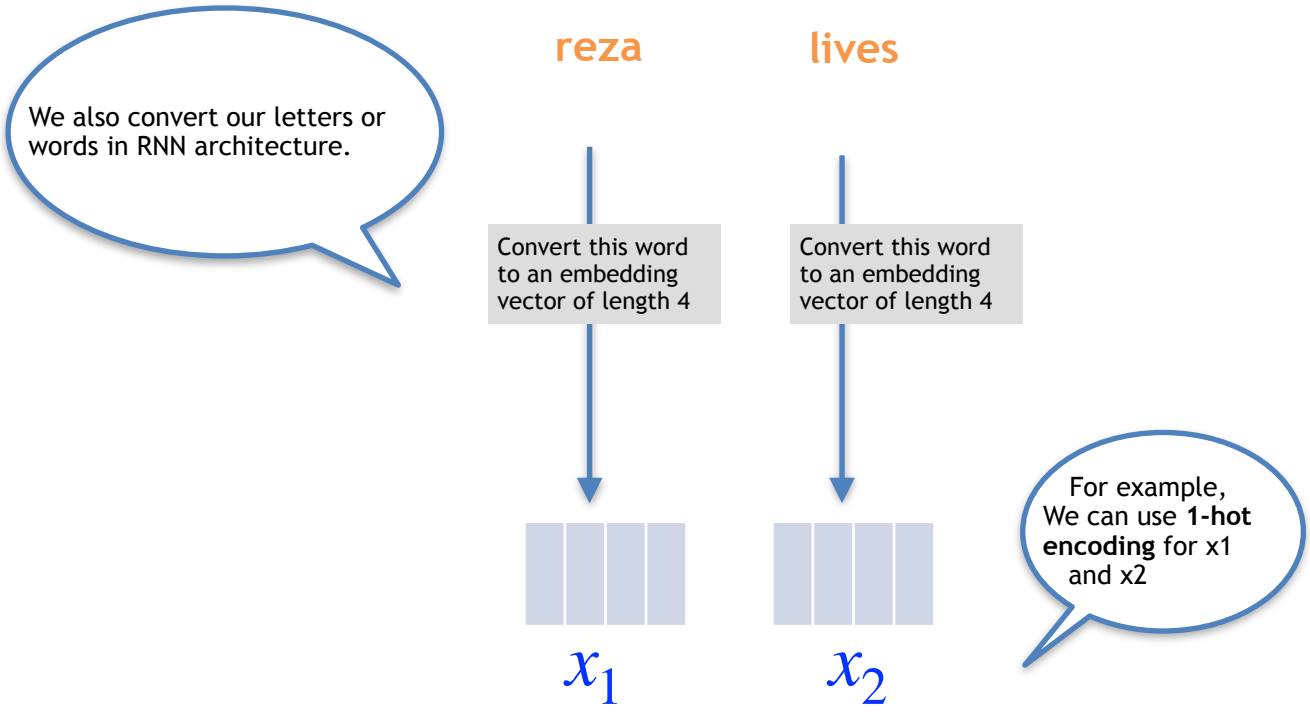

Attention is all you need - NeurIPS'2017

# Transformers



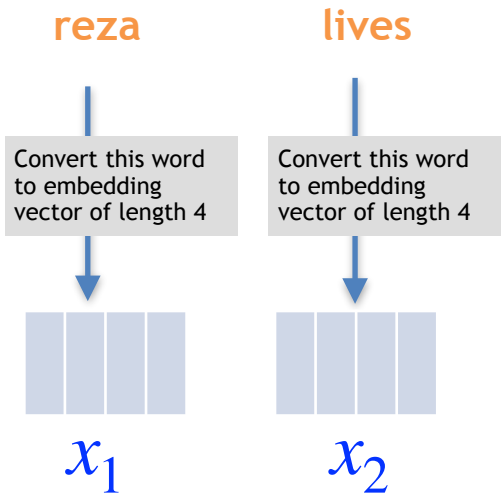
# Attention

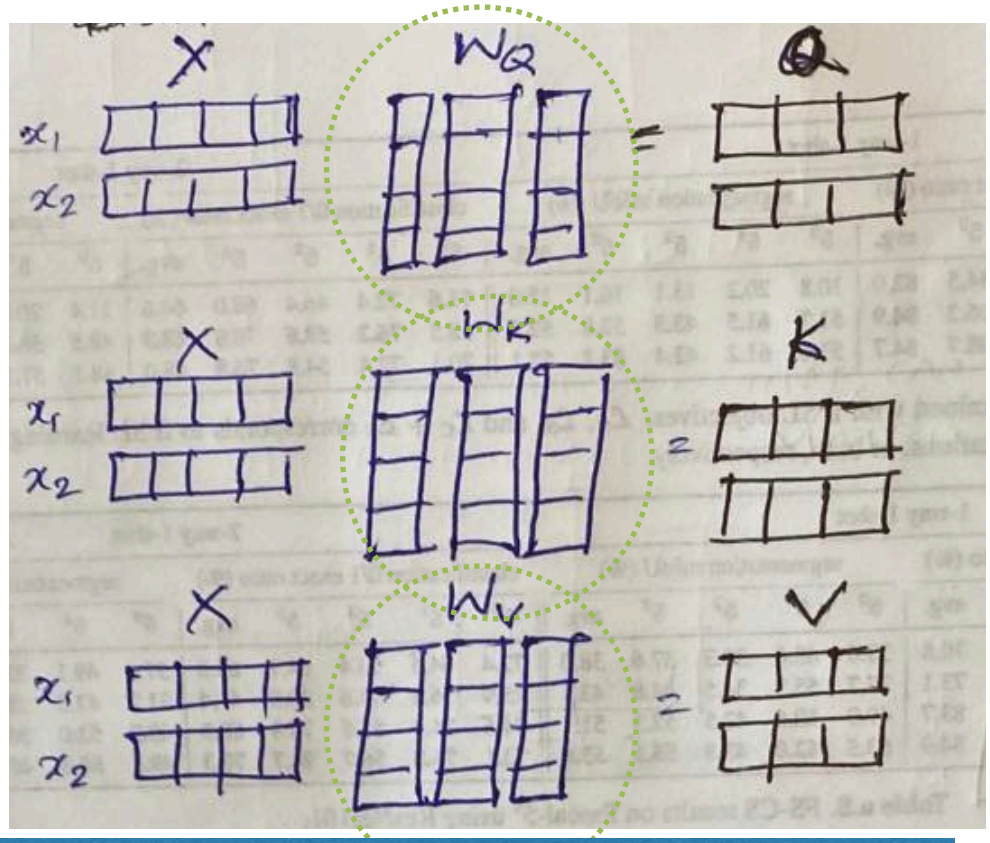
- Let's find out how to calculate the attention mechanism in a toy example
- Let's calculate attention with first two words of our sentence: "reza lives"



# Attention

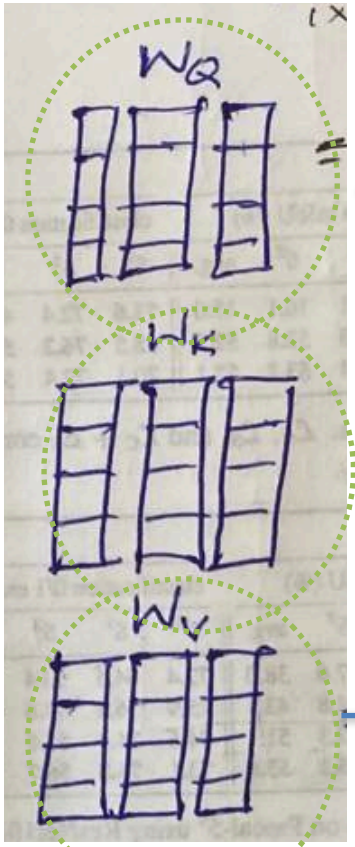
- It calculates three new matrices Q, K, and V with the help of three weight matrices $W_Q$,$W_K$, and $W_V$
- These three matrices ($W_Q$,$W_K$, and $W_V$) are learned during training



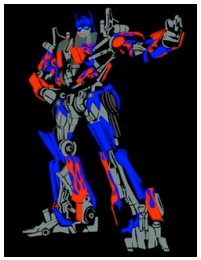



# Attention

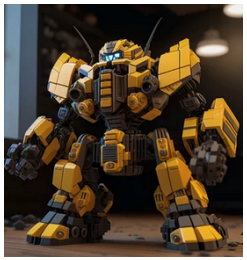
- It calculates three new matrices Q, K, and V with the help of three weight matrices $W_Q$,$W_K$, and $W_V$
- These three matrices ($W_Q$,$W_K$, and $W_V$) are learned during training




Optimus prime



Bumble Bee



Ironhide

# So why do we need this complicated attention? It's all about "context"

- Consider a language model trying to predict the next word based on the previous ones. Let's predict the last word to this sequence:
  - "The clouds are in the ?"

- We can guess from recent information that it will be the name of a language; we need the context of **France** to make a prediction.
  - "I grew up in France, I speak fluent ?"

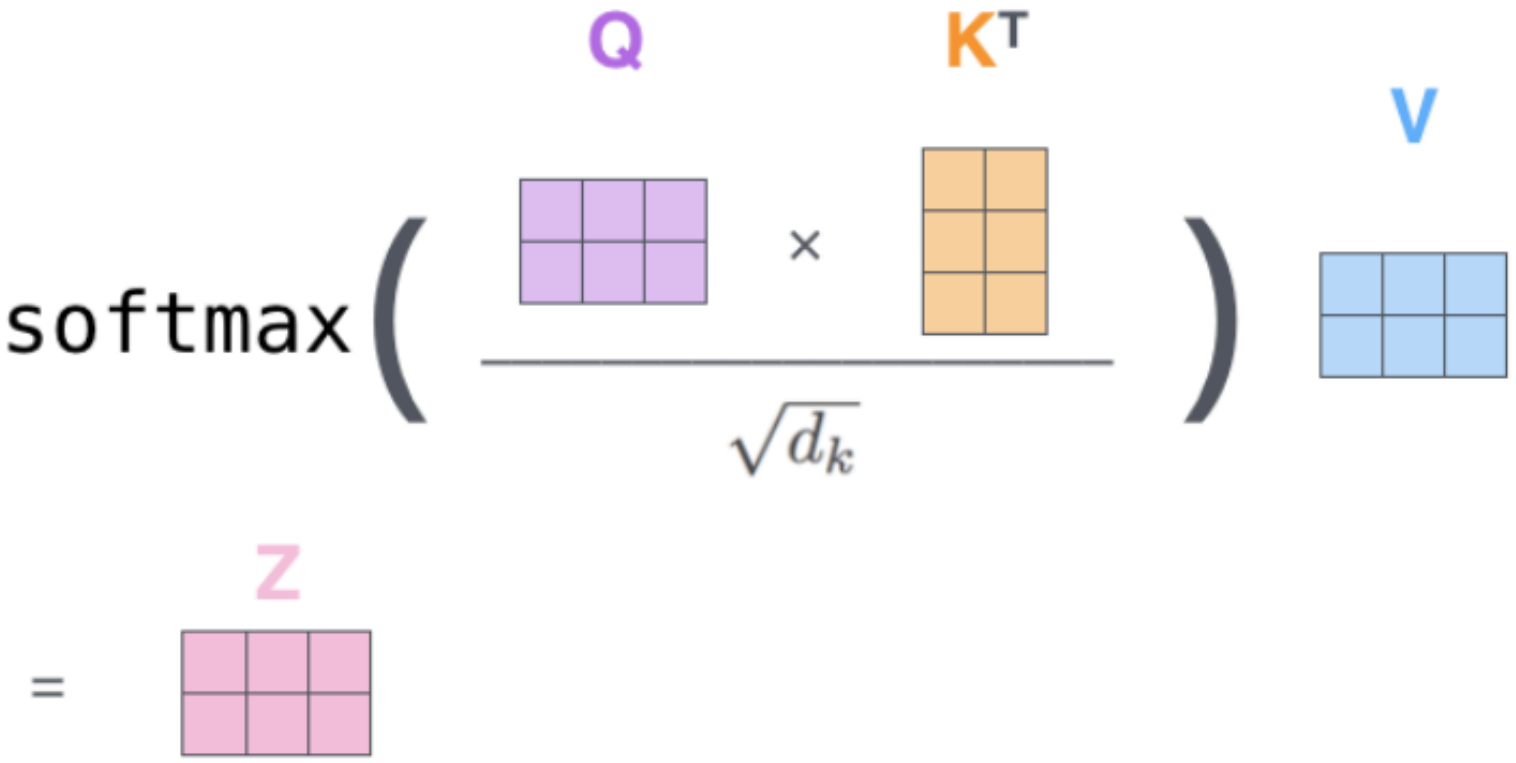
# So why do we need this complicated attention? It's all about "context"

- Consider a language model trying to predict the next word based on the previous ones. Let's predict the last word to this sequence:
  - "The clouds are in the sky"

- We can guess from recent information that it will be the name of a language; we need the context of **France** to make a prediction.
  - "I grew up in France, I speak fluent french"

# Self-Attention

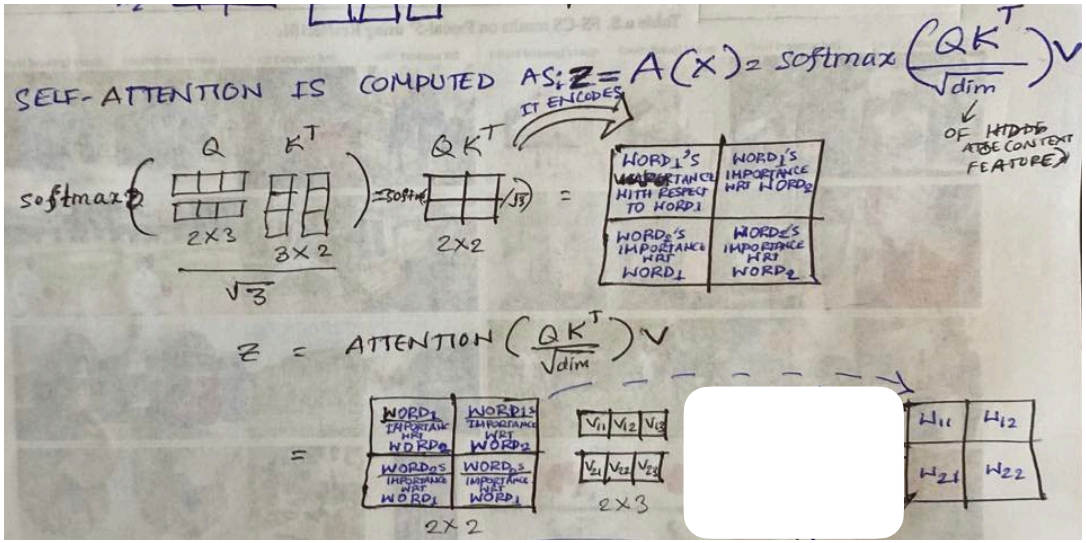
- Finally, attention is calculated using Q, K, and V matrices using the following equation:



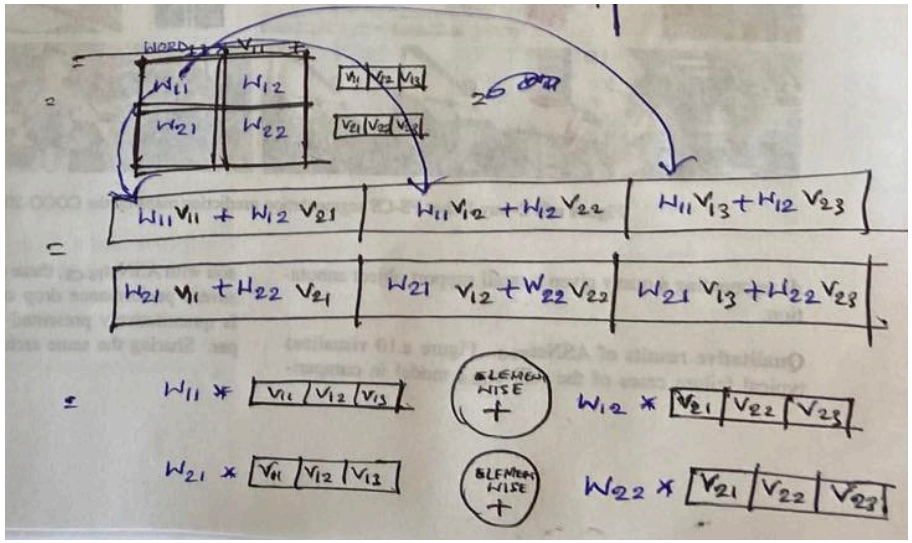

Reference: Illustrated Transformer

# Self-Attention

My hand-notes




My hand-notes




Reference: Illustrated Transformer

# Going Back to the Transformer Idea

- This new mechanism for context learning, called the **attention mechanism** is only one part—of course, the central one.

- There are other components. Let's examine those.

**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



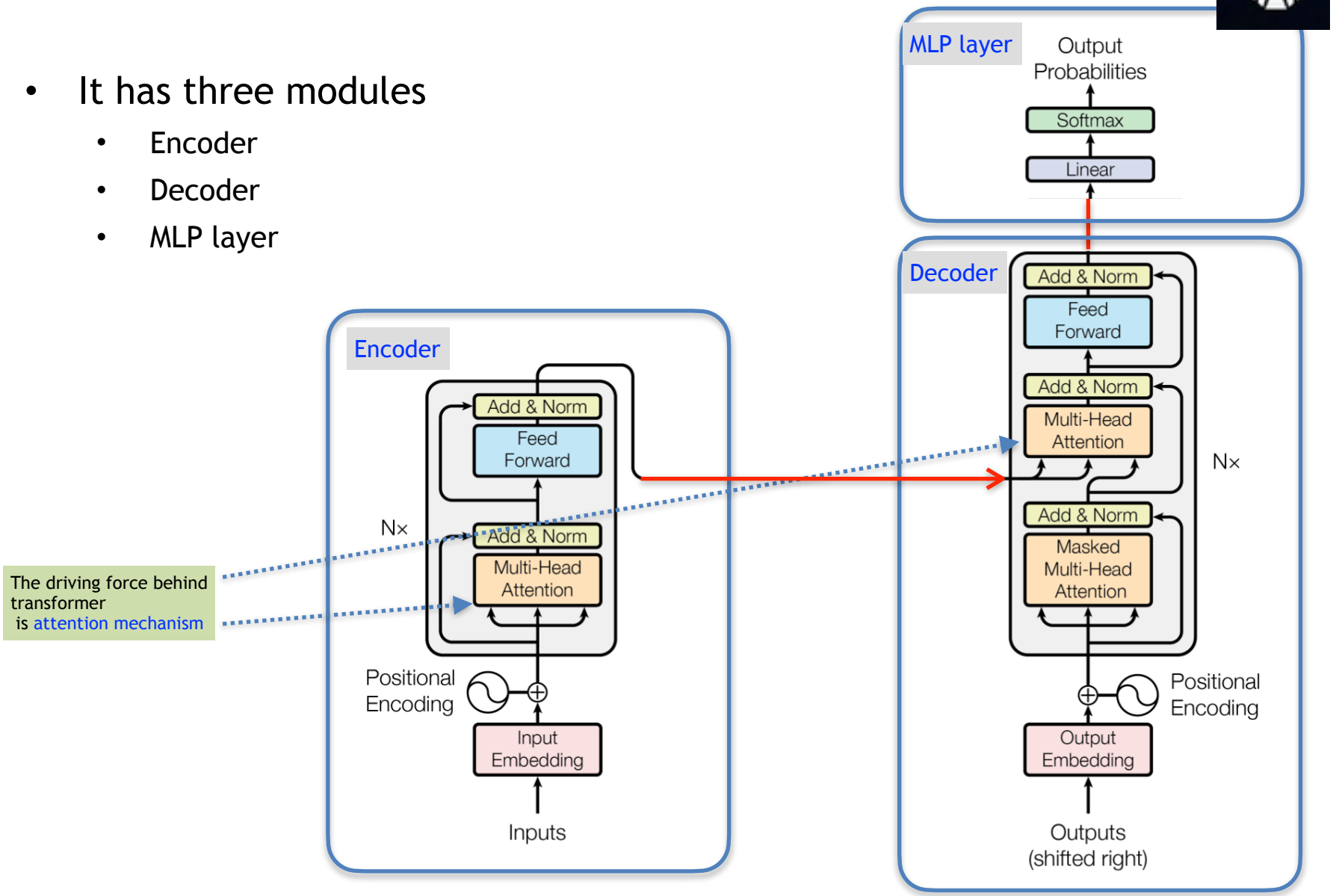
**Abstract**

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 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.


Attention is all you need - NeurIPS'2017

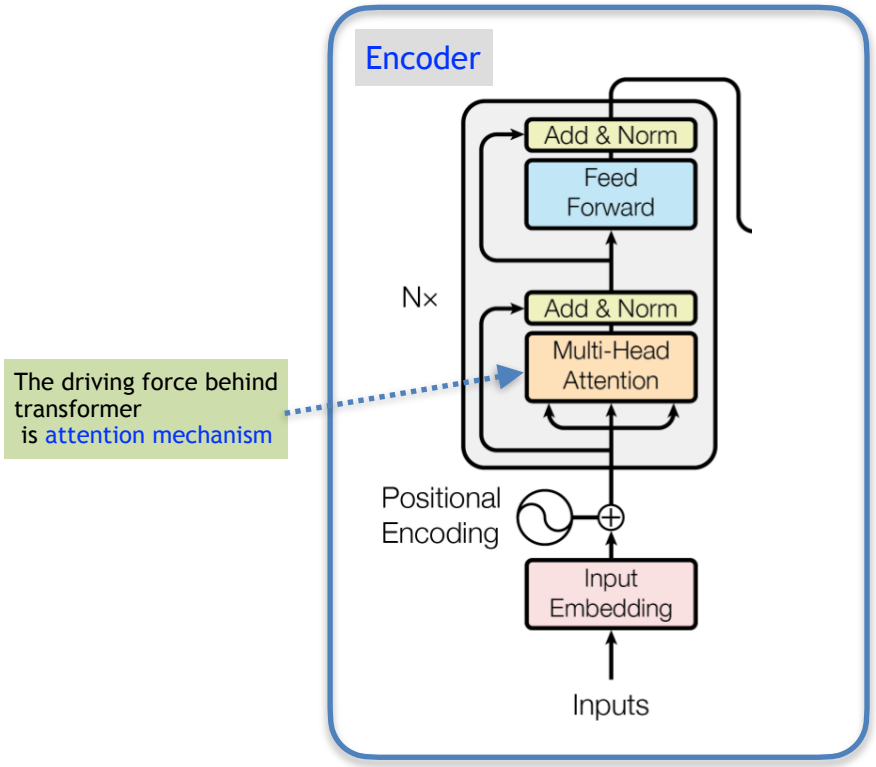
# Transformers

- It has three modules
  - Encoder
  - Decoder
  - MLP layer



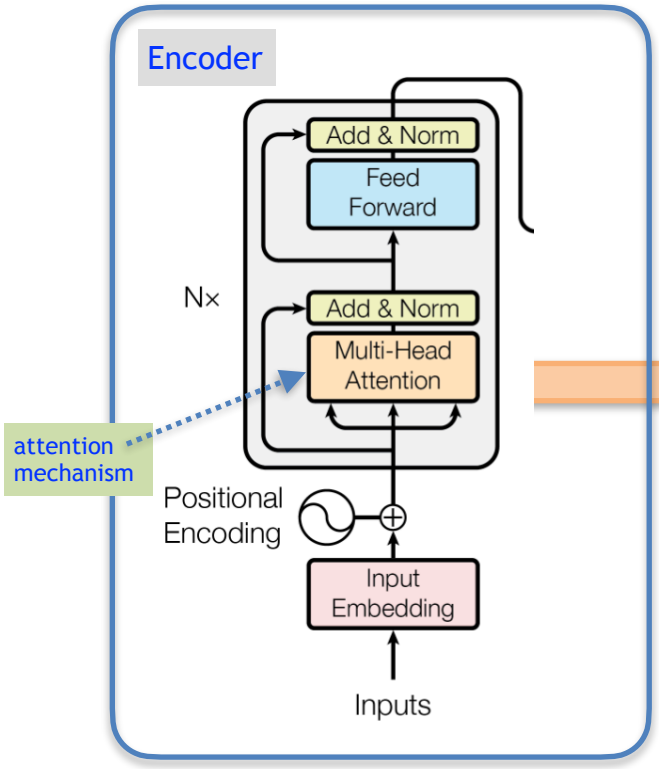
# Transformers: Encoder

- Lets focus on the encoder to understand what is this attention mechanism

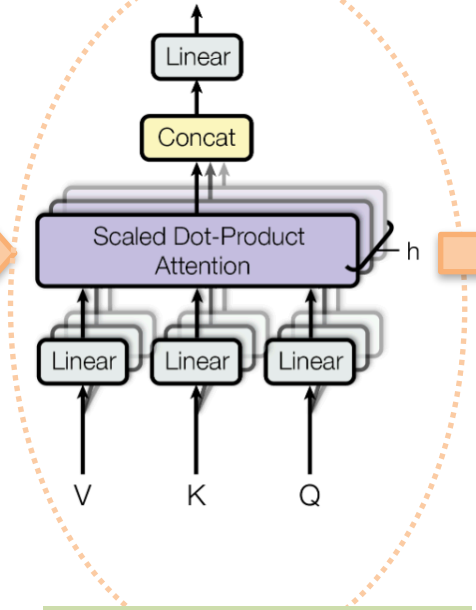


# Transformers: Encoder

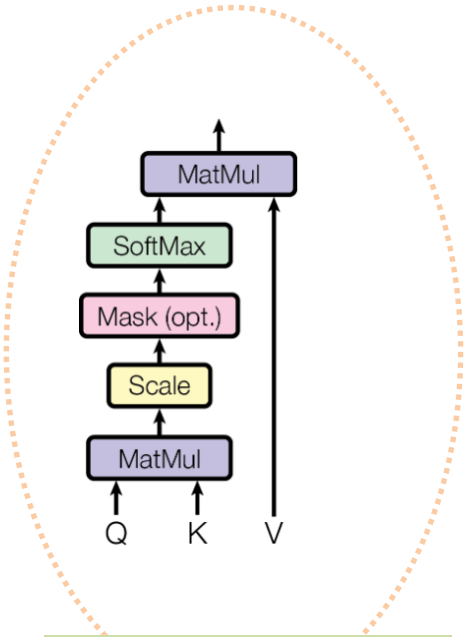
- Lets focus on the encoder to understand what is this attention mechanism




Encoder has a **multi-headed attention**



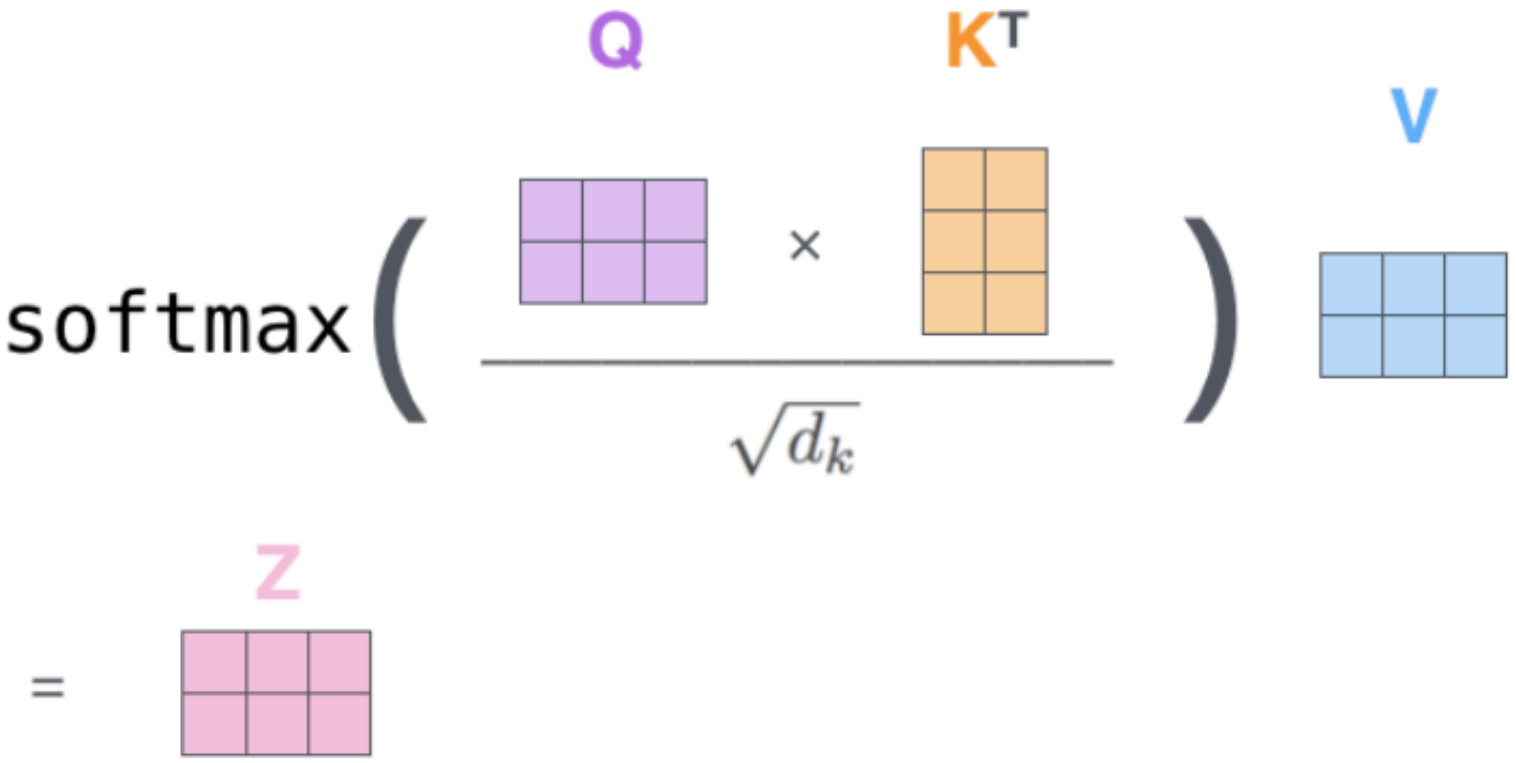

A multi-headed attention has several **scaled dot-product attention**




A single scaled dot-product attention has sequential matrix matrix multiplications in terms of Query (Q), Key (K), and Value (V)

# Exactly what we discussed in Attention computation

- Self-attention is calculated using Q, K, and V matrices using the following equation:




Reference: Illustrated Transformer

# Today's agenda

- Transformers
  - Transfer learning is possible
  - New type of network architecture besides convolutional neural network

- Vision Transformer (ViT)

# Vision Transformer (ViT)

- In 2021, the **attention mechanism** is used for computer vision problems
- Excellent results compared to CNN while requiring substantially fewer parameters advantage
- For classification the encoder part of transformer is sufficient advantage

Published as a conference paper at ICLR 2021

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

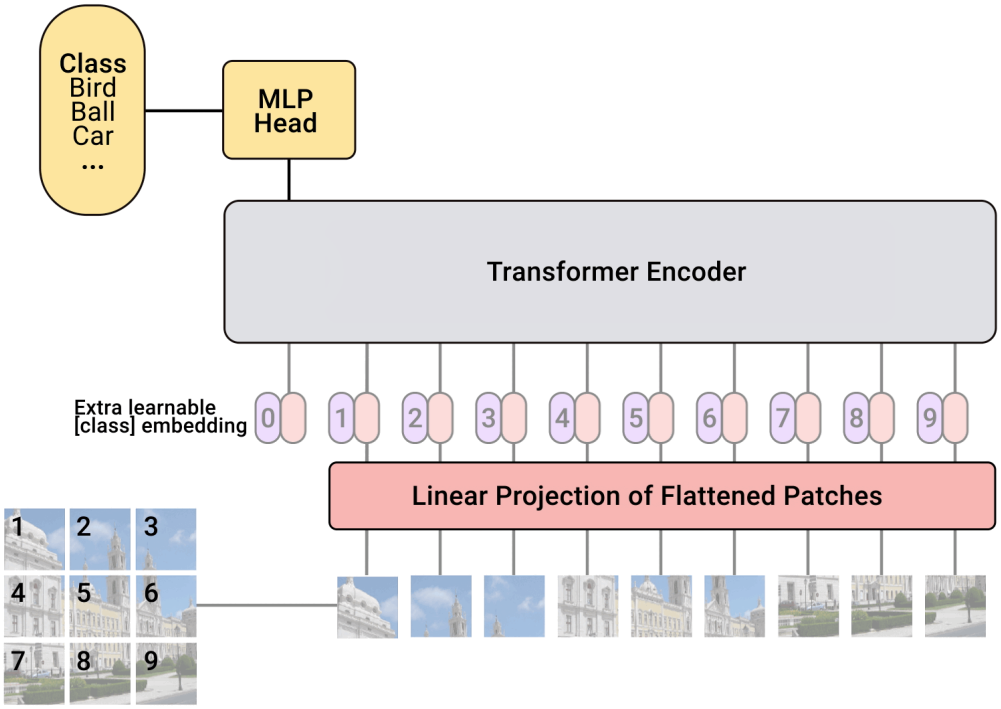
**Alexey Dosovitskiy*,†, Lucas Beyer*, Alexander Kolesnikov*, Dirk Weissenborn*, Xiaohua Zhai*, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby*,†**
*equal technical contribution, †equal advising
Google Research, Brain Team
{adosovitskiy, 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].

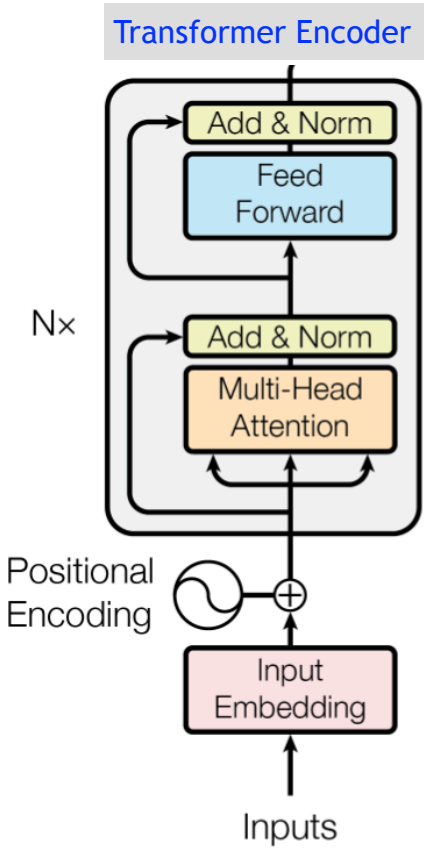
An image is worth 16x16 words: transformers for image recognition at scale - ICLR'21

# Vision Transformer (ViT) for Image Classification

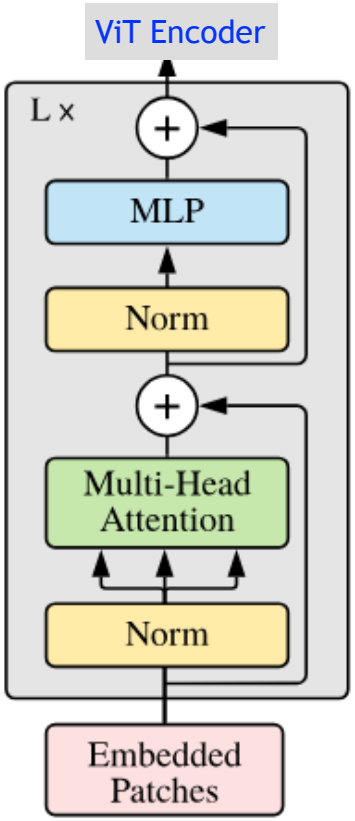



- It's an application of transformer for computer vision problem with fewest possible modifications.
- For image classification task, only need the encoder part of Transformer
- To mimic the setup of a language model, an image is divided into patches (consider each patch as a word token).
- Each patch is converted to fix-sized embedding, and each patch encoding is attached with a positional encoding.

# Transformer Encoder vs. ViT Encoder

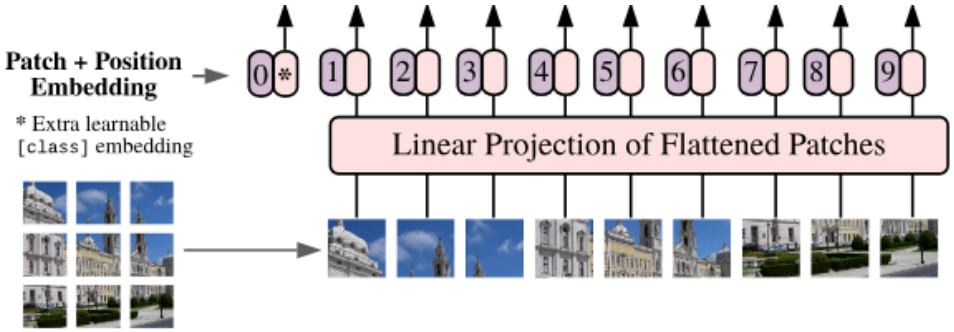



- Components of encoder are same
- Layers are rearranged
  - Location of norm is important
- Both generate fix-sized embedding
- Both adds positional encoding after embedding generation
- ViT adds an extra classification token to the sequence of patches
  - It is learnable parameters (output)



# Embedding and Positional Encoding

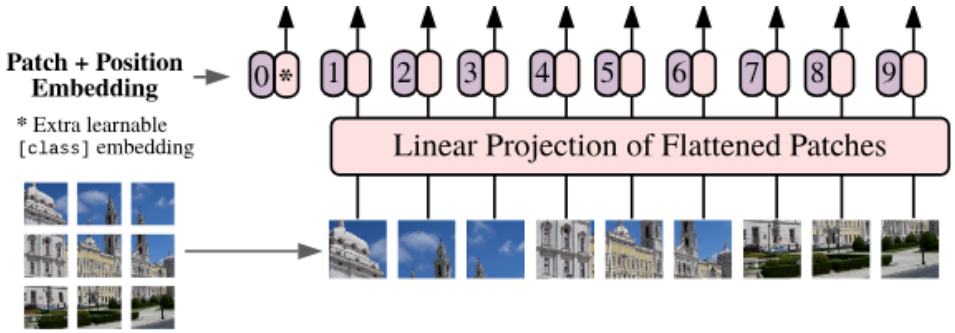
- Divide an image into 16x16 (hyper-parameter which you can pick and chose). Each patch will contain 256 pixels. There will be several such patches (eg, 9 patches in the image below) for that particular image.




- Linearly embed it to some lower dimension eg, 64-dimensional embedding
- Positional embedding is added to retain the positional information, eg, 1D learnable position
- Classification toke is added at zero$^{-th}$ position

# Positional Encoding

- Different ways of encoding spatial information (positional information)
  - No positional information: input as a bag of patches
  - 1D positional information: a sequence of patches in the raster order
  - 2D positional information: two sets of embedding are learned (x- and y-embeddings)




- Different ways of incorporating this information (1D and 2D):
  - Right after stem before feeding it to the ViT encoder block
  - Learn and add the positional encoding at the beginning of each layer (not shared between layers)
  - Learn and add the positional encoding at the beginning of each layer (shared between layers)

# Inductive Bias in CNN vs. Inductive Bias ViT

- ViT lack inductive bias (unlike CNNs)
  - Don't generalize well when trained on insufficient data, eg ImageNet is insufficient
  - But large-scale training helps

- Inductive bias in CNNs
  - Locality
  - Grid-like 2D neighborhood structure
  - Translation equivariance (translating the patch in different location on an image will produce the same convolutional output)
  - Translation invariance (achieves it to some level with multiple spatial pooling layers)

- Inductive bias in ViT
  - Self-attention is global (means each patch's attention is computed with all other patches during self-attention computation)
  - MLP layers are local and have translation equivariance
  - Negligible 2D information; apart from 1D positional encoding, everything is learned from scratch by transformer

# Vision Transformer Comparisons
## (PyTorch official implementation)

### Vision Transformer (ViT)

| Weight | Acc@1 | Acc@5 | Params | GFLOPS |
|---|---|---|---|---|
| ViT_B_16_Weights.IMAGENET1K_V1 | 81.072 | 95.318 | 86.6M | 17.56 |
| ViT_B_16_Weights.IMAGENET1K_SWAG_E2E_V1 | 85.304 | 97.65 | 86.9M | 55.48 |
| ViT_B_16_Weights.IMAGENET1K_SWAG_LINEAR_V1 | 81.886 | 96.18 | 86.6M | 17.56 |
| ViT_B_32_Weights.IMAGENET1K_V1 | 75.912 | 92.466 | 88.2M | 4.41 |
| ViT_H_14_Weights.IMAGENET1K_SWAG_E2E_V1 | 88.552 | 98.694 | 633.5M | 1016.72 |
| ViT_H_14_Weights.IMAGENET1K_SWAG_LINEAR_V1 | 85.708 | 97.73 | 632.0M | 167.29 |
| ViT_L_16_Weights.IMAGENET1K_V1 | 79.662 | 94.638 | 304.3M | 61.55 |
| ViT_L_16_Weights.IMAGENET1K_SWAG_E2E_V1 | 88.064 | 98.512 | 305.2M | 361.99 |
| ViT_L_16_Weights.IMAGENET1K_SWAG_LINEAR_V1 | 85.146 | 97.422 | 304.3M | 61.55 |
| ViT_L_32_Weights.IMAGENET1K_V1 | 76.972 | 93.07 | 306.5M | 15.38 |

### Swin Transformer (another variant of vision transformer)

| Weight | Acc@1 | Acc@5 | Params | GFLOPS |
|---|---|---|---|---|
| Swin_B_Weights.IMAGENET1K_V1 | 83.582 | 96.64 | 87.8M | 15.43 |
| Swin_S_Weights.IMAGENET1K_V1 | 83.196 | 96.36 | 49.6M | 8.74 |
| Swin_T_Weights.IMAGENET1K_V1 | 81.474 | 95.776 | 28.3M | 4.49 |
| Swin_V2_B_Weights.IMAGENET1K_V1 | 84.112 | 96.864 | 87.9M | 20.32 |
| Swin_V2_S_Weights.IMAGENET1K_V1 | 83.712 | 96.816 | 49.7M | 11.55 |
| Swin_V2_T_Weights.IMAGENET1K_V1 | 82.072 | 96.132 | 28.4M | 5.94 |

# Vision Transformer (ViT) for Classification

- Torch vision has pretrained weights for various vision transformers

Open this notebook: vision transformer (ViT)