Tokens and Transformers

Instructor - Simon Lucey

RVSS - 2025

THE UNIVERSITY of ADELAIDE

AUSTRALIAN INSTITUTE FOR MACHINE LEARNING

Today

- **Tokenization and ConvNets**
- Skip Connections and BatchNorm
- Attention and Transformers
- Self Supervision

$$\mathbf{X} \in \mathbb{R}^{768 \times 320}$$

Reminder: Shallow vs Deep Network

- A shallow learner has **only one** hidden unit,

$$\boldsymbol{\theta}\eta(\mathbf{W}\mathbf{x} + \boldsymbol{\delta})$$

- A deep learner **has more than one** hidden unit,

$$\boldsymbol{\theta}\eta(\mathbf{W}^{(1)}\eta(\mathbf{W}^{(0)}\mathbf{x} + \boldsymbol{\delta}^{(0)}) + \boldsymbol{\delta}^{(1)})$$

76800 x 76800 76800 x 1

Reminder: Shallow vs Deep Network

- A shallow learner has **only one** hidden unit,

$$\boldsymbol{\theta}\eta(\mathbf{W}\mathbf{x}+\boldsymbol{\delta})$$

- A deep learner **has more than one** hidden unit,

$$\boldsymbol{\theta}\eta(\mathbf{W}^{(1)}\eta(\mathbf{W}^{(0)}\mathbf{x}+\boldsymbol{\delta}^{(0)})+\boldsymbol{\delta}^{(1)})$$

76800 x 76800

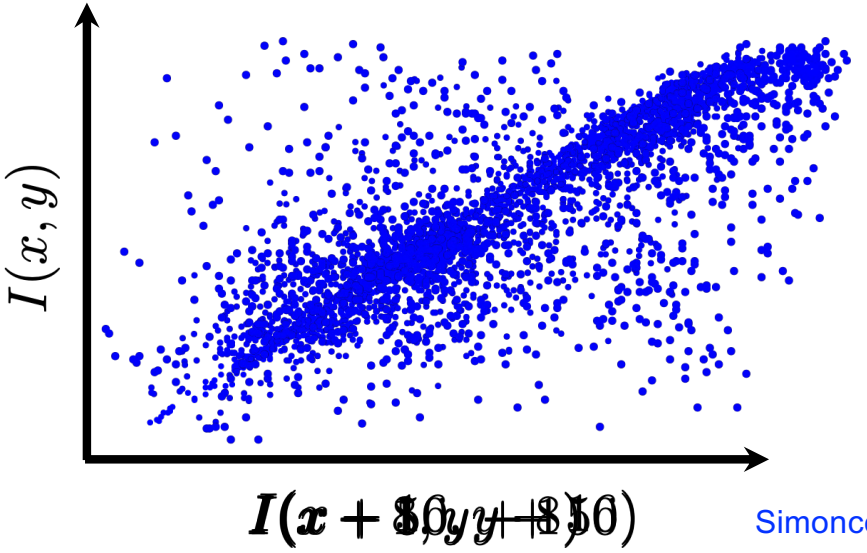
76800 x 1

11.8 billon parameters!!!

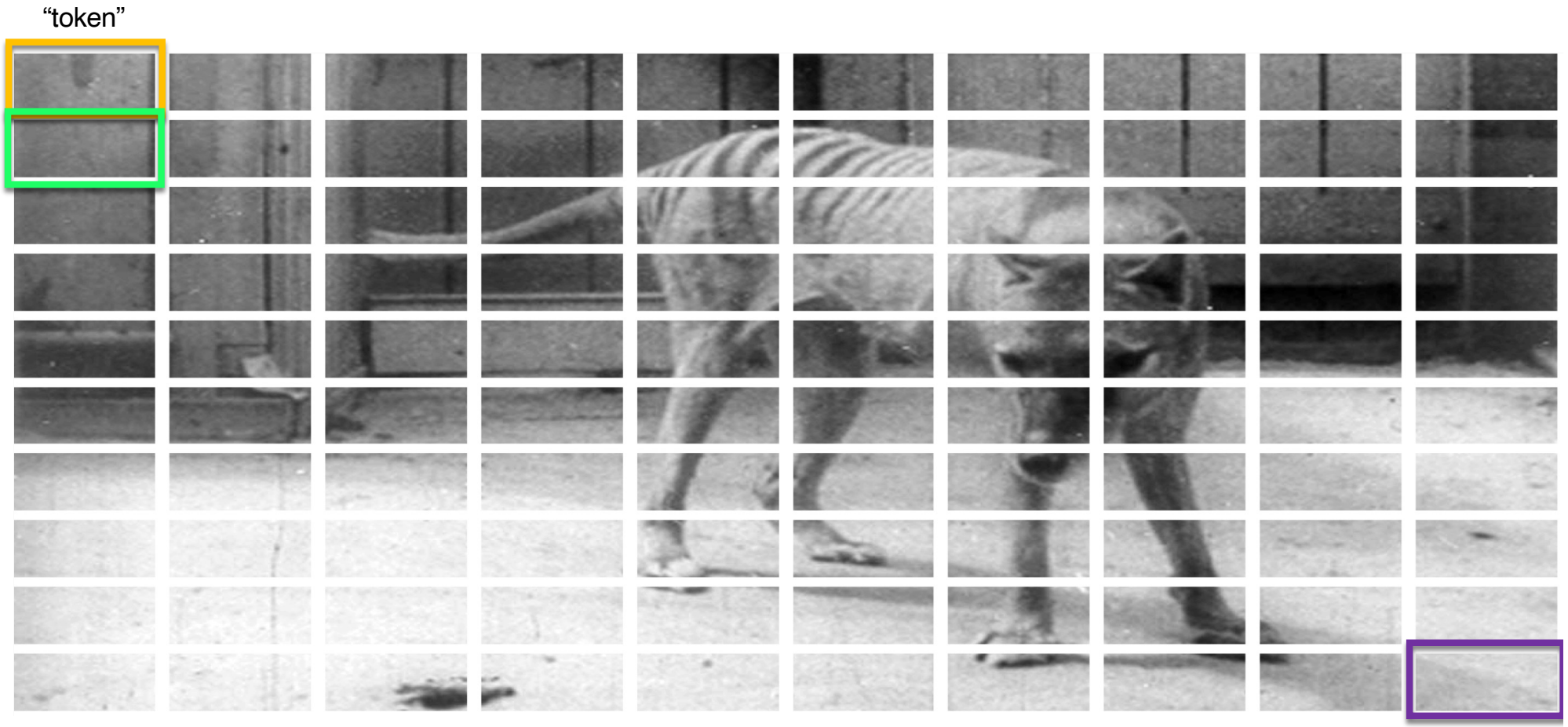
Let’s have a play!!!

https://colab.research.google.com/github/slucey-cs-cmu-edu/RVSS24/blob/main/MLP_Example.ipynb

I



Simoncelli & Olshausen 2001

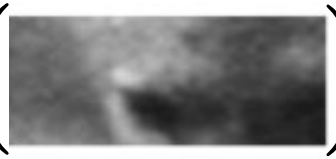


$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^{\mathrm{T}} \in \mathbb{R}^{N \times D}$$

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^{\mathrm{T}} \in \mathbb{R}^{N \times D}$$

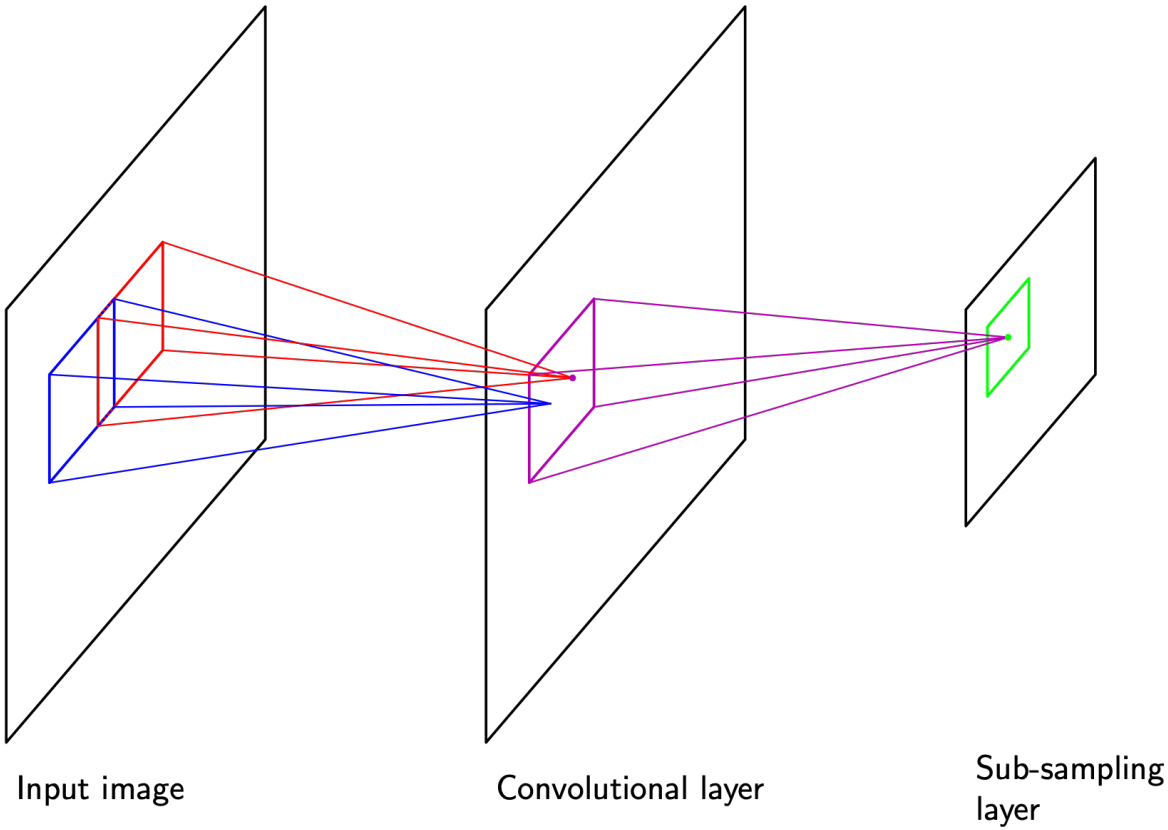
$N \rightarrow \#$ of tokens

$D \rightarrow \#$ of channels

$$\mathbf{x}_n = \mathrm{vec}\Big($$  $$\Big)$$

"token"

Convolutional Neural Network



LeCun 1980

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

Abstract

We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called "dropout" that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

Reminder: Shallow vs Deep Network

- A shallow learner has **only one** hidden unit,

$$\boldsymbol{\theta}\eta(\mathbf{W}\mathbf{x} + \boldsymbol{\delta})$$

- A deep learner **has more than one** hidden unit,

$$\boldsymbol{\theta}\eta(\mathbf{W}^{(1)}\eta(\mathbf{W}^{(0)}\mathbf{x} + \boldsymbol{\delta}^{(0)}) + \boldsymbol{\delta}^{(1)})$$

$$\boldsymbol{\theta}\eta(\mathbf{W}^{(1)}\eta(\mathbf{W}^{(0)}\mathbf{x}+\boldsymbol{\delta}^{(0)})+\boldsymbol{\delta}^{(1)})$$

"tokenization"

$$\mathbf{W}^{(0)}\mathbf{x} = \mathrm{vec}(\mathbf{U}^{(0)}\mathbf{X}\mathbf{V}^{(0)})$$

"spatial mixing" "channel mixing"

$$= \mathbf{V}^{(0)T} \otimes \mathbf{U}^{(0)}\mathrm{vec}(\mathbf{X})$$

$$= [\mathbf{V}^{(0)T} \otimes \mathbf{U}^{(0)}]\mathbf{x}$$

why? $N^2 + D^2 \gg N^2 D^2 \quad (\mathbf{X} \in \mathbb{R}^{N \times D})$

$$\mathbf{U}[\mathbf{X}_1, \ldots, \mathbf{X}_H]$$

$(N \times D/H)$ $(N \times D/H)$

"spatial mixing"

"multiple heads"

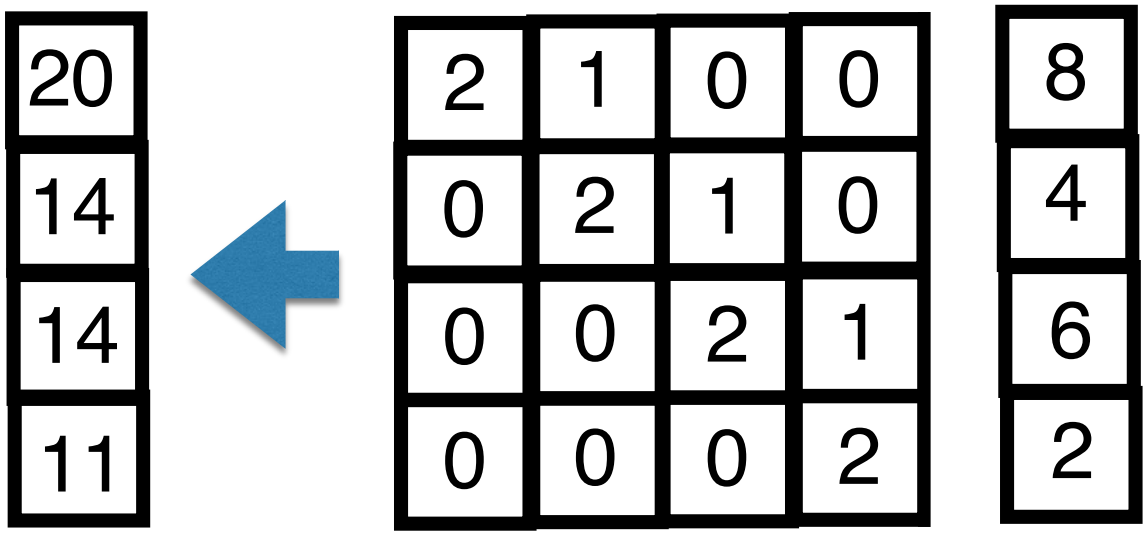
$$[\mathbf{U}_1\mathbf{X}_1, \ldots, \mathbf{U}_H\mathbf{X}_H]$$

"head specific spatial mixing"

"head specific spatial mixing"

$$H \cdot N^2 + D^2 \gg N^2 D^2$$

Reminder Convolution



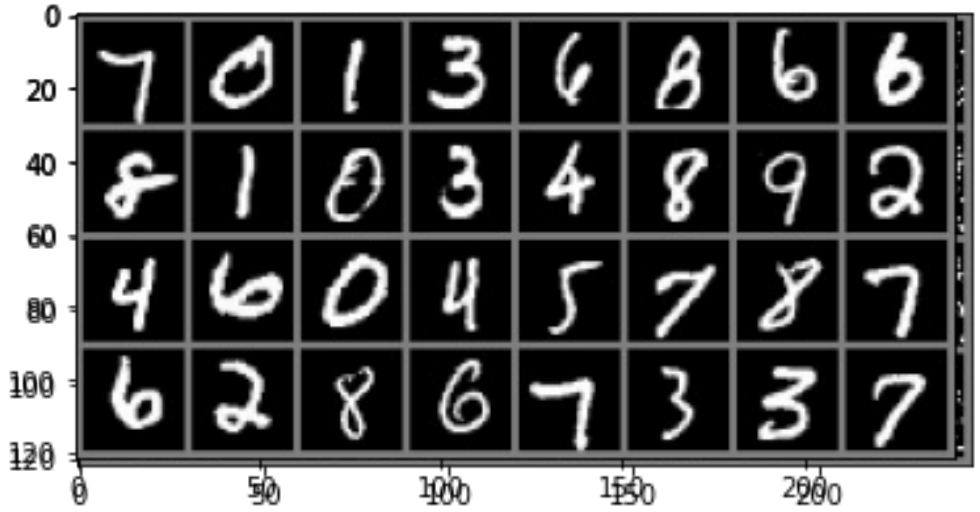
Let’s have another play!!!

https://colab.research.google.com/github/slucey-cs-cmu-edu/RVSS24/blob/main/Token_Example.ipynb

Some things to try!!!

- What happens to performance if you use a linear function?
- What happens when you permute the pixels?

```
from numpy.random import permutation
idx_permute = torch.from_numpy(permutation(784))
transform = transforms.Compose([transforms.ToTensor(),
                                transforms.Lambda(lambda x: x.view(-1)[idx_permute].view(1, 28, 28) ),
                               transforms.Normalize((0.5,), (0.5,)),
                               ])
```

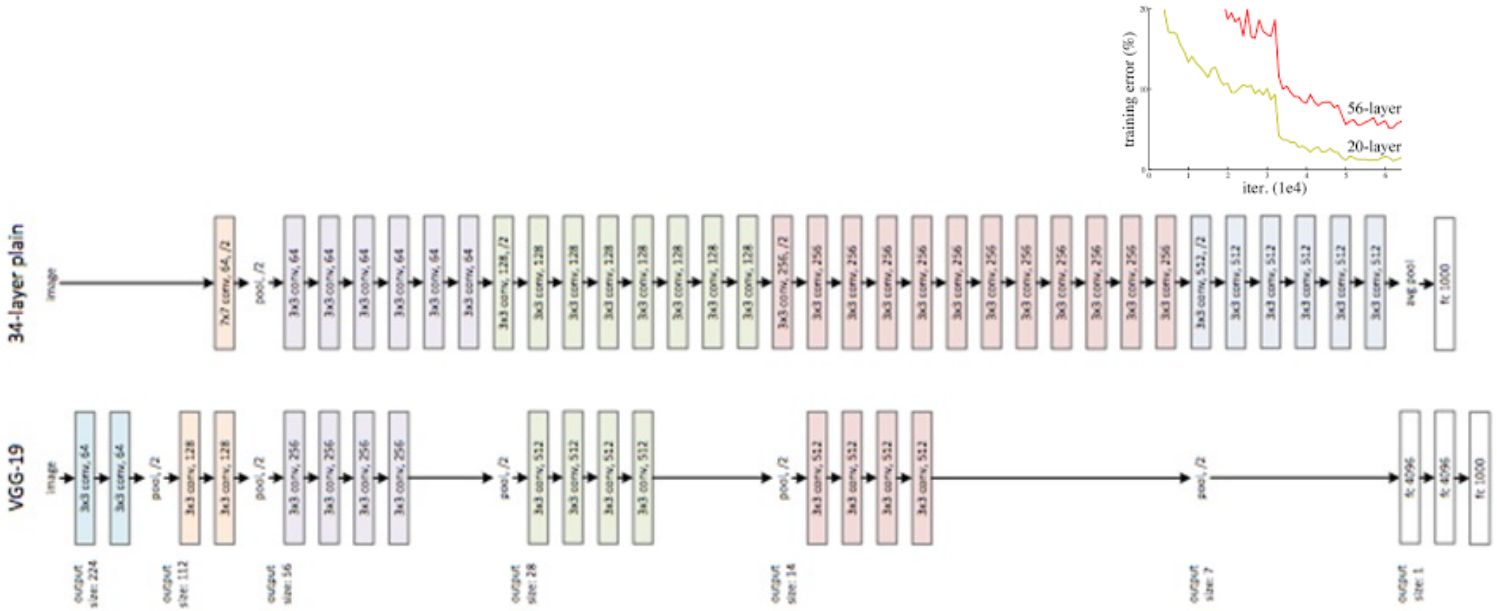


Today

- Tokenization and ConvNets
- **Skip Connections and BatchNorm**
- Attention and Transformers

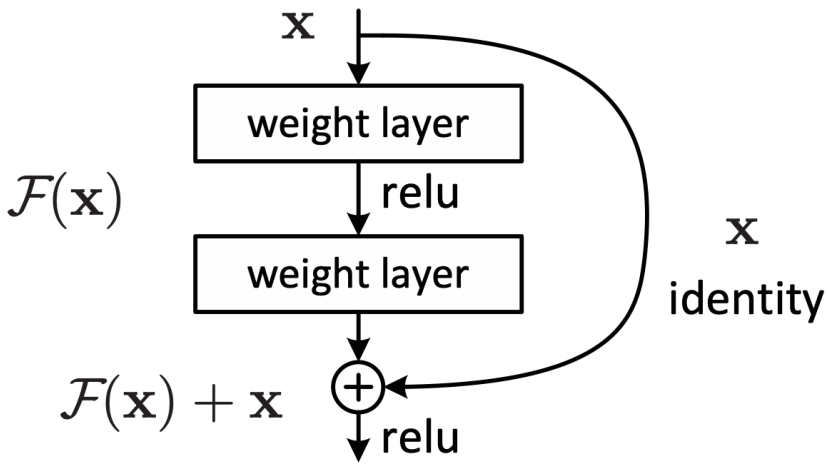
Neural Net : Deeper is Better?

- Very deep network -- training error increases.



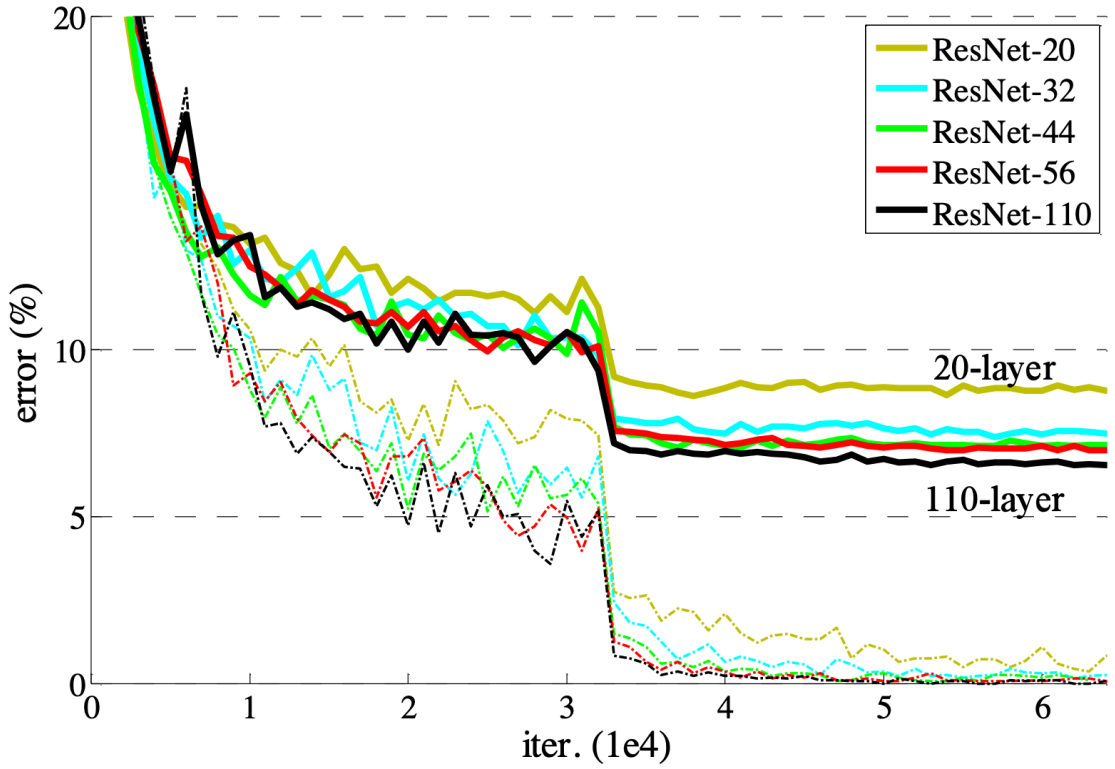
How Deep?

- Network depth is crucial for good performance (e.g. ImageNet).
- Counter intuitively, naively trained deeper networks tend to have higher train error than shallow networks.
- Innovation of residual learning has greatly helped with this.



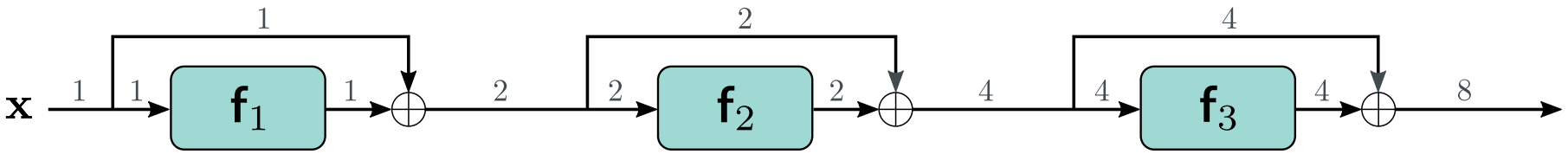
K. He, et al. "Deep residual learning for image recognition", CVPR 2016.

How Deep?



K. He, et al. "Deep residual learning for image recognition", CVPR 2016.

Initialization Problem

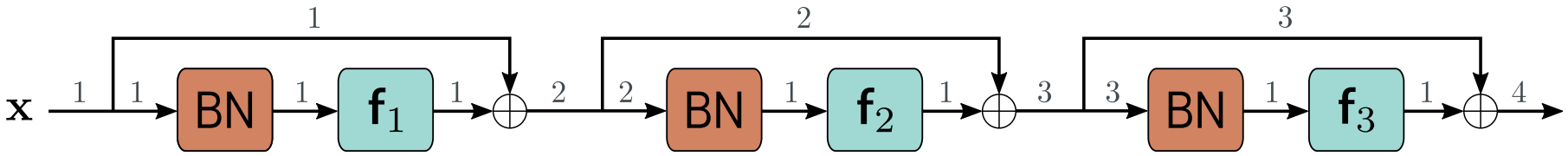


Try example:-

https://colab.research.google.com/github/udlbook/udlbook/blob/main/Notebooks/Chap11/11_3_Batch_Normalization.ipynb

Prince “Understanding Deep Learning” 2023.

Batch Norm

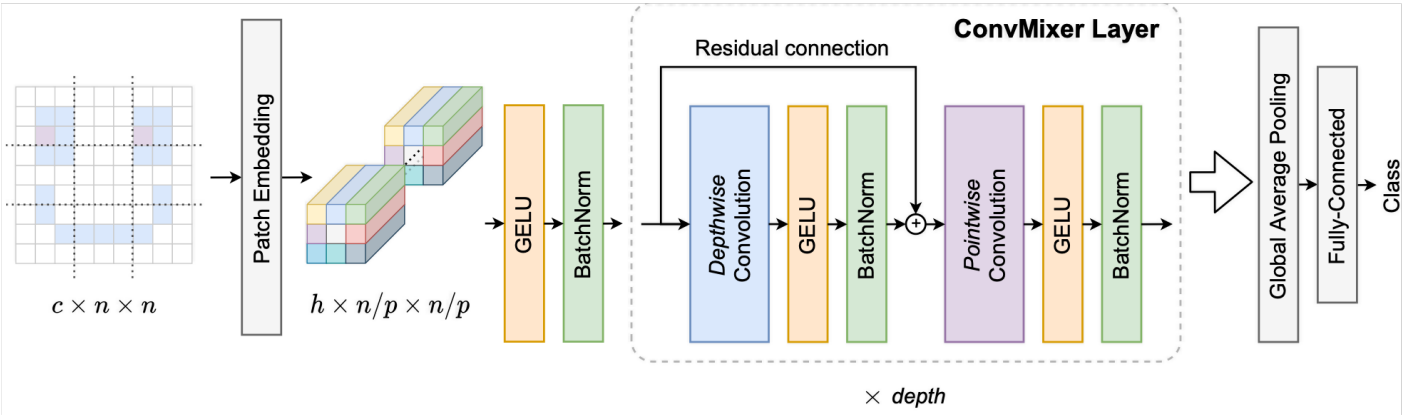


$$m_h = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} h_i \qquad h_i \leftarrow \frac{h_i - m_h}{s_h + \epsilon}$$

$$s_h = \sqrt{\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}(h_i - m_h)^2}, \qquad h_i \leftarrow \gamma h_i + \delta$$

Prince “Understanding Deep Learning” 2023.

ConvMixer - Patches are all you need



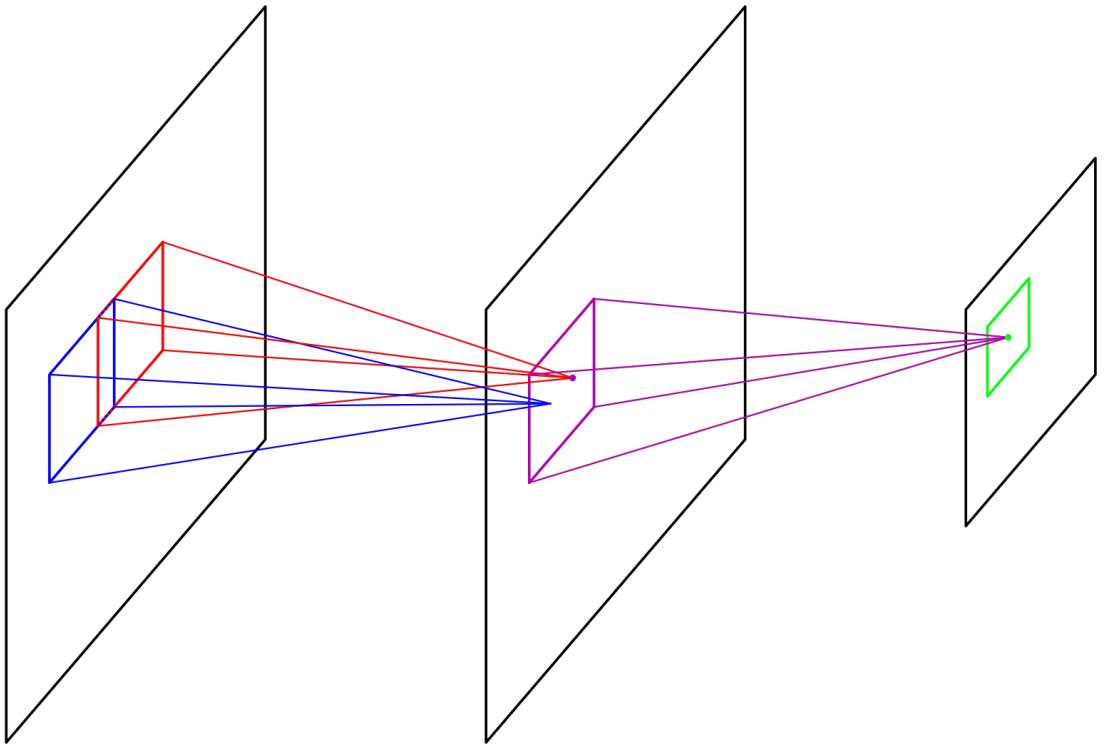

```
def ConvMixer(h, depth, kernel_size=9, patch_size=7, n_classes=1000):
    Seq, ActBn = nn.Sequential, lambda x: Seq(x, nn.GELU(), nn.BatchNorm2d(h))
    Residual = type('Residual', (Seq,), {'forward': lambda self, x: self[0](x) + x})
    return Seq(ActBn(nn.Conv2d(3, h, patch_size, stride=patch_size)),
        *[Seq(Residual(ActBn(nn.Conv2d(h, h, kernel_size, groups=h, padding="same"))),
          ActBn(nn.Conv2d(h, h, 1))) for i in range(depth)],
        nn.AdaptiveAvgPool2d((1,1)), nn.Flatten(), nn.Linear(h, n_classes))
```

https://colab.research.google.com/github/slucey-cs-cmu-edu/RVSS24/blob/main/Simple_ViT_in_Colab.ipynb

Today

- Tokenization and ConvNets
- Skip Connections and BatchNorm
- **Attention and Transformers**
- Self Supervision

Convolutional = A Type of Attention??



LeCun 1980

Attention is all you need!!

- Convolution is hard-coded attention.
- Could there be any way to learn it from data?

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

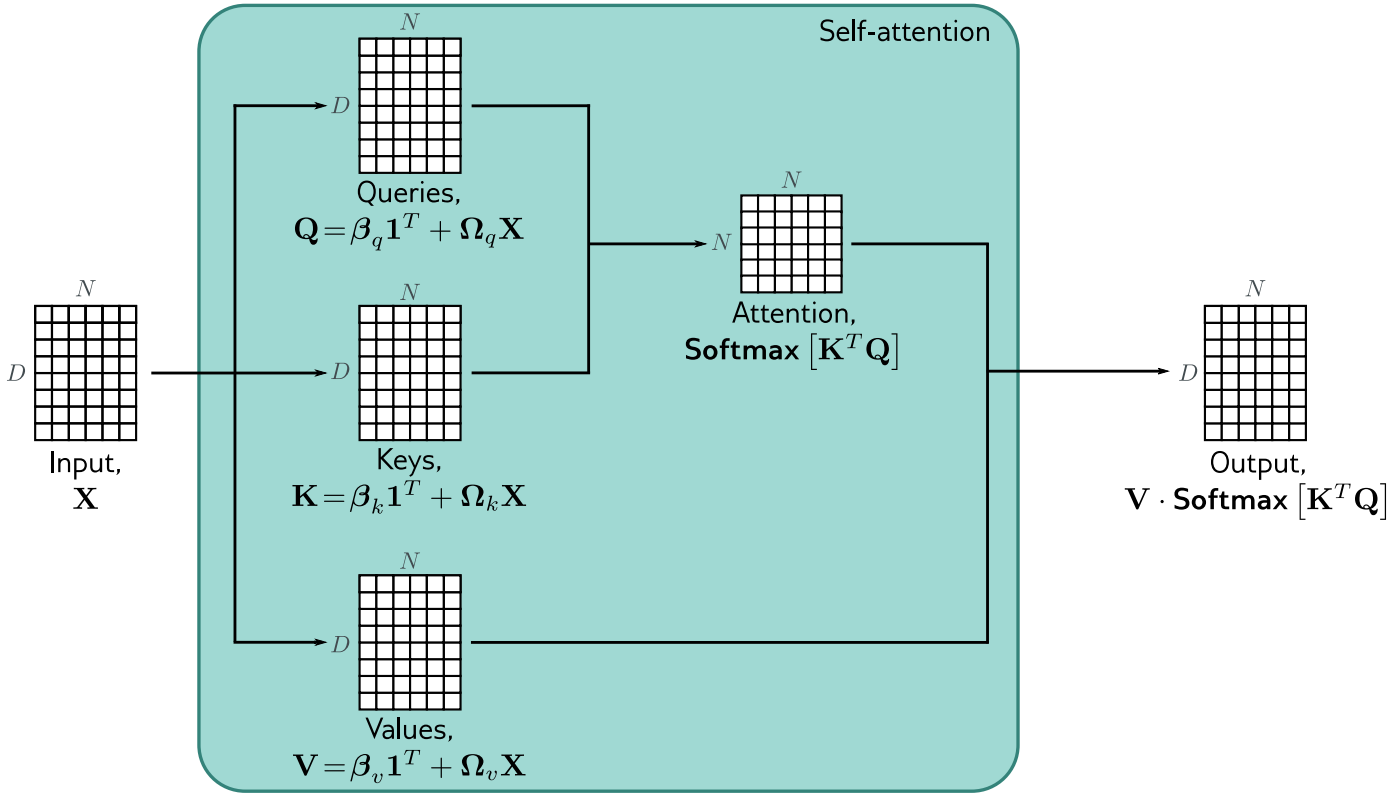
A. Vaswani et al. “Attention is all you need”, NeurIPS 2017.

Soft-Max Attention

$$
\begin{aligned}
a[\mathbf{x}_m, \mathbf{x}_n] &= \text{softmax}_m\left[\mathbf{k}_\bullet^T \mathbf{q}_n\right] \\
&= \frac{\exp\left[\mathbf{k}_m^T \mathbf{q}_n\right]}{\sum_{m'=1}^{N} \exp\left[\mathbf{k}_{m'}^T \mathbf{q}_n\right]},
\end{aligned}
$$

Prince “Understanding Deep Learning” 2023.

What is Attention?



Prince “Understanding Deep Learning” 2023.

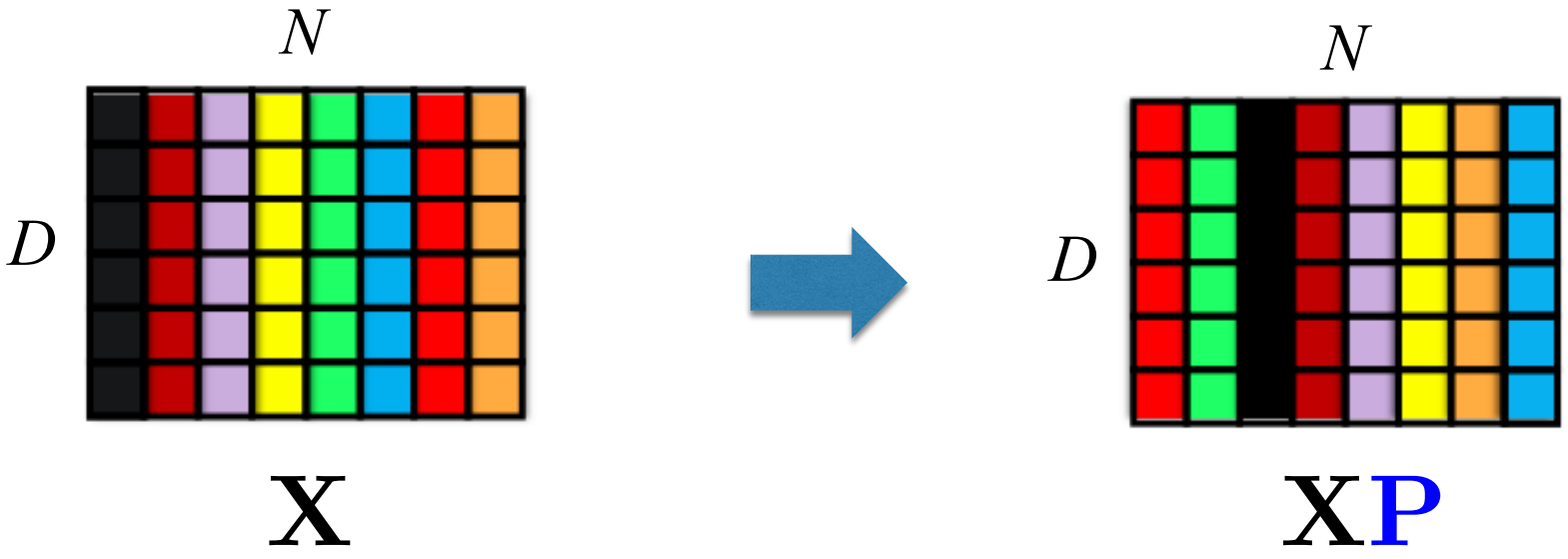
What is Attention?

$$\begin{aligned}\mathbf{V}[\mathbf{X}] &= \boldsymbol{\beta}_v\mathbf{1}^\mathbf{T} + \boldsymbol{\Omega}_v\mathbf{X} \\ \mathbf{Q}[\mathbf{X}] &= \boldsymbol{\beta}_q\mathbf{1}^\mathbf{T} + \boldsymbol{\Omega}_q\mathbf{X} \\ \mathbf{K}[\mathbf{X}] &= \boldsymbol{\beta}_k\mathbf{1}^\mathbf{T} + \boldsymbol{\Omega}_k\mathbf{X},\end{aligned}$$

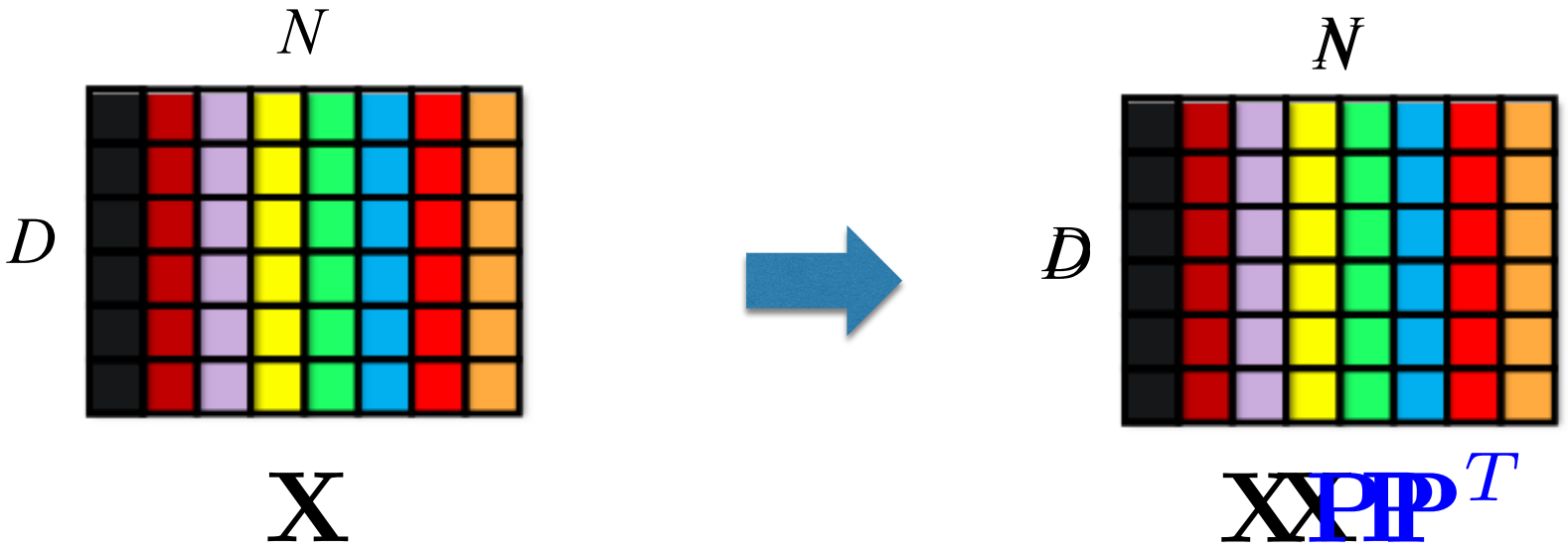
$$\mathbf{Sa}[\mathbf{X}] = \mathbf{V}[\mathbf{X}]\cdot\mathbf{Softmax}\left[\mathbf{K}[\mathbf{X}]^T\mathbf{Q}[\mathbf{X}]\right],$$

Prince “Understanding Deep Learning” 2023.

Permutating X



Permutating X



Permutation Equivariance?

$$\mathbf{Sa}(\mathbf{X}) = \mathbf{Sa}(\mathbf{X}\mathbf{P}\mathbf{P}^T)$$

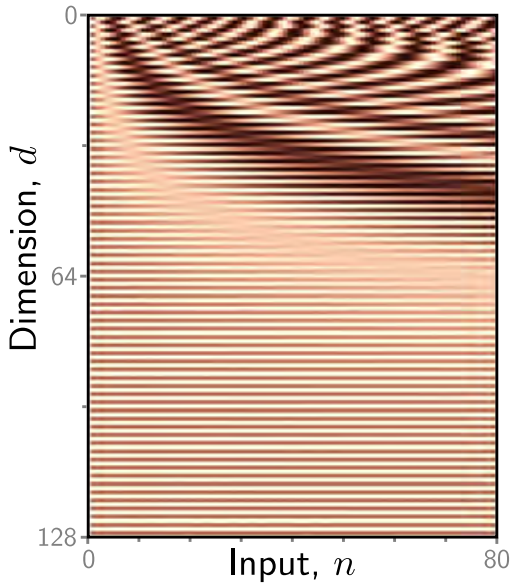
$$\mathbf{Sa}(\mathbf{X}) = \mathbf{Sa}(\mathbf{X}\mathbf{P})\mathbf{P}^T$$

Position information ignored!!!

Positional Encoding

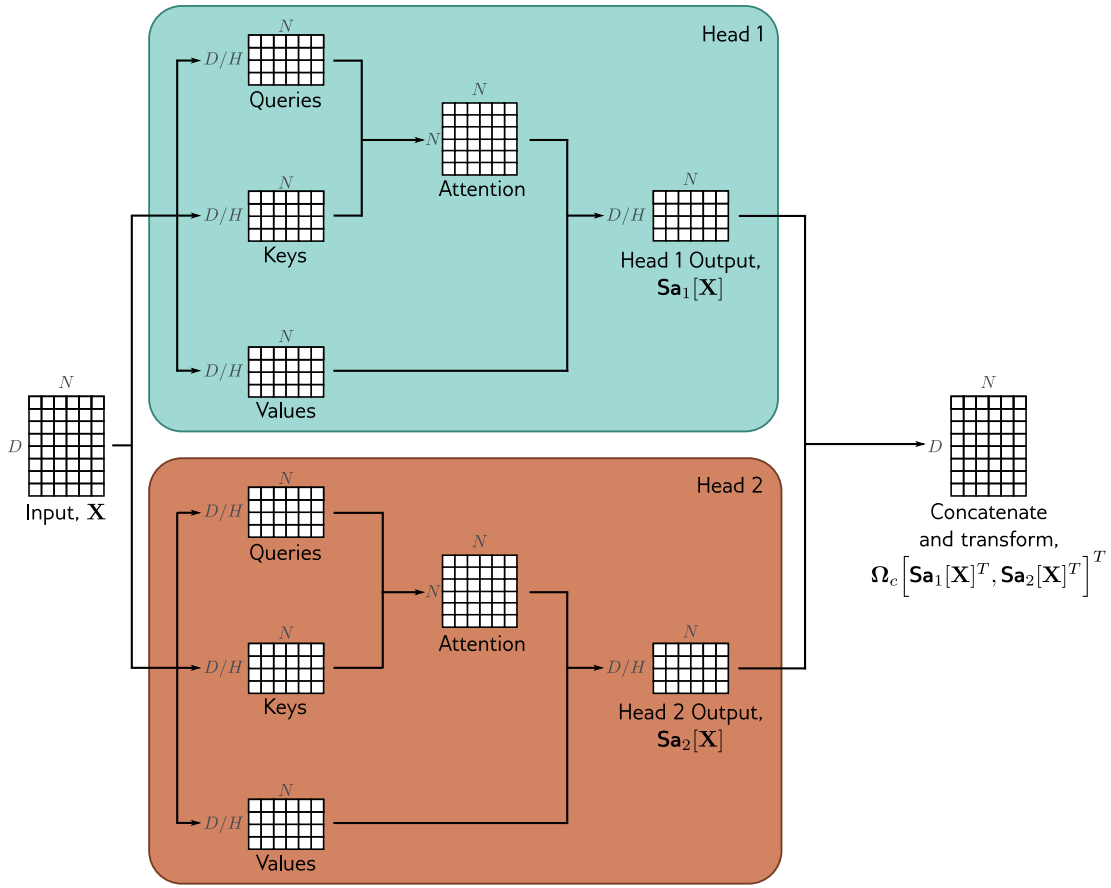
$$\mathbf{X} \rightarrow \mathbf{X} + \mathbf{P}$$

$\mathbf{P} =$



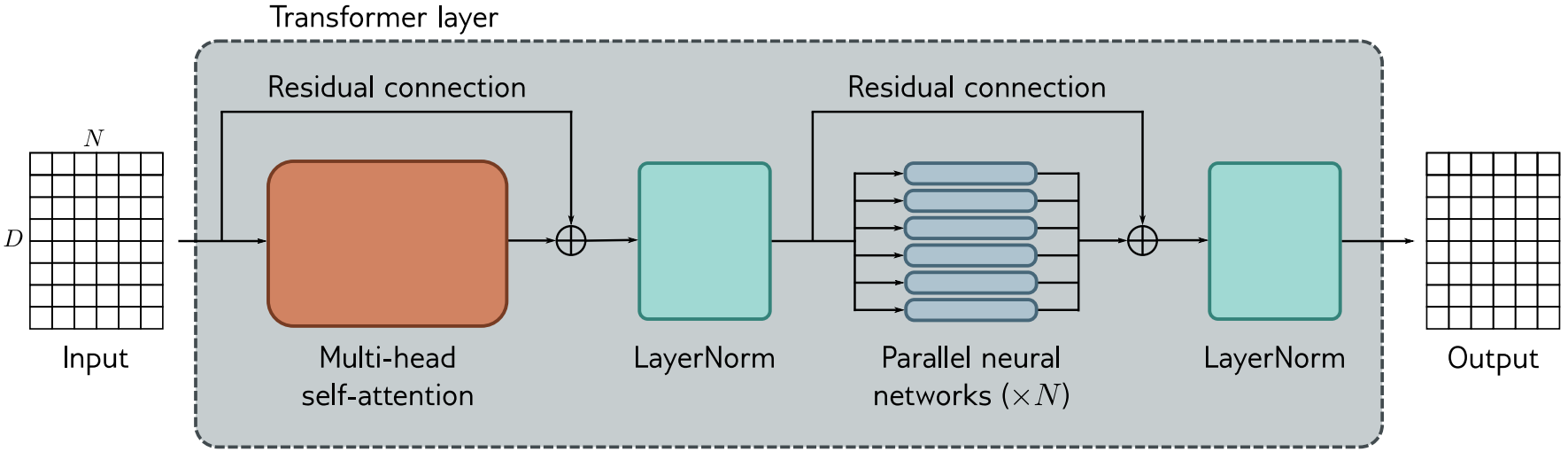
Prince “Understanding Deep Learning” 2023.

Multi-Head Attention



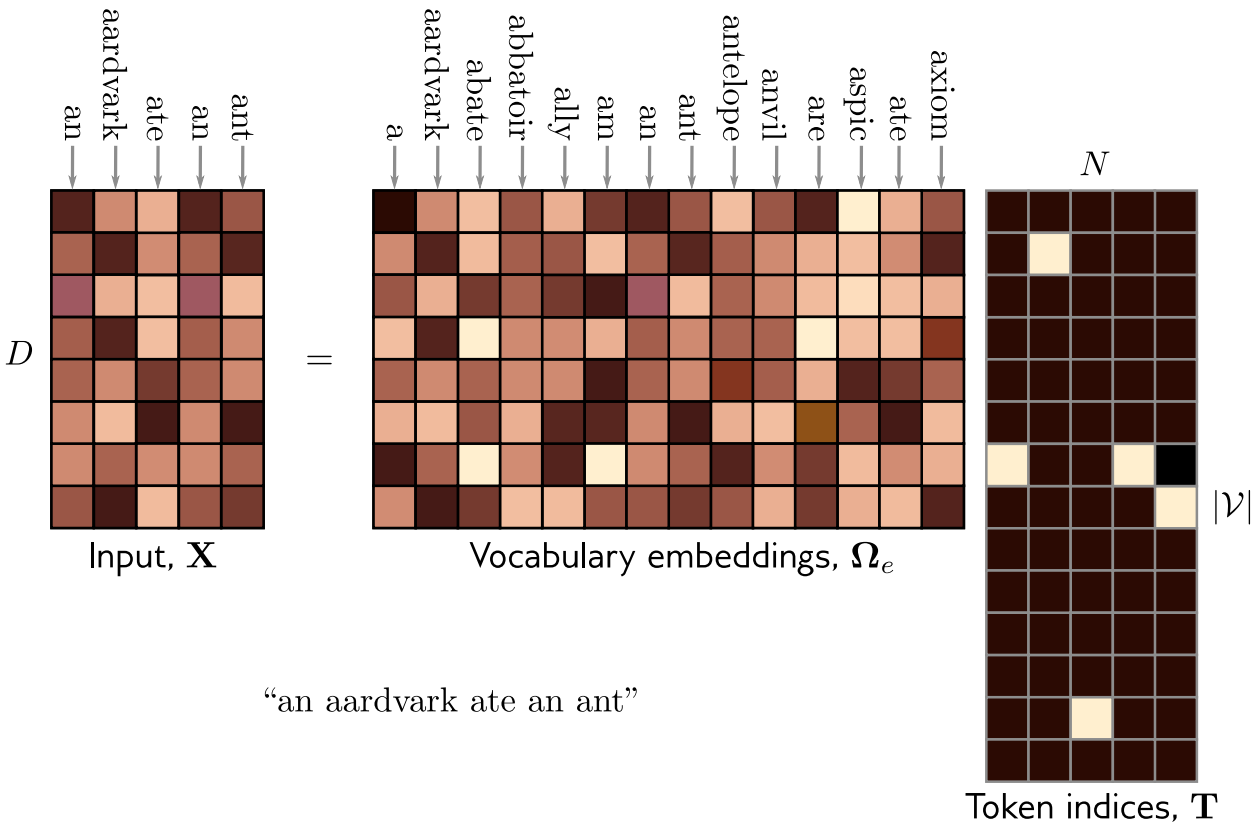
Prince “Understanding Deep Learning” 2023.

Transformer Layer



Prince “Understanding Deep Learning” 2023.

Language + Transformers



Prince "Understanding Deep Learning" 2023.

Vision Transformer (ViT)

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

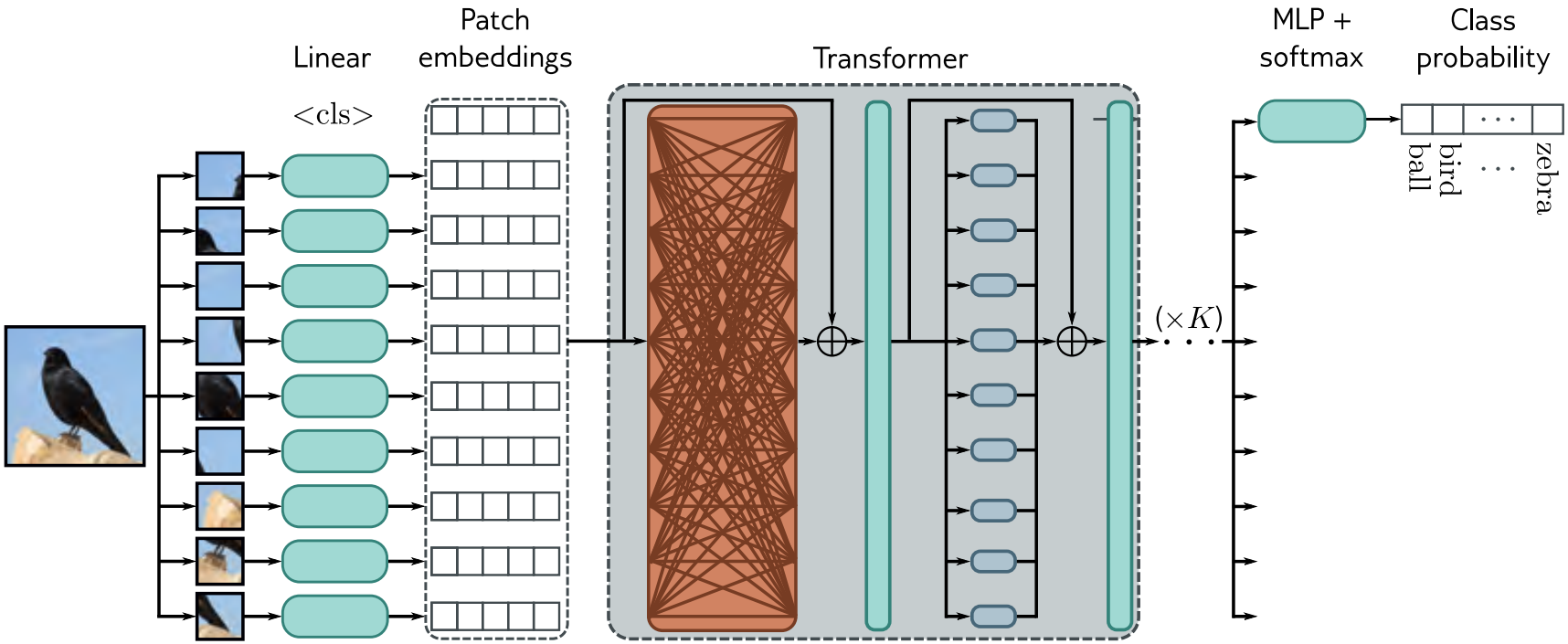
Google Research, Brain Team

{adosovitskiy, neilhoulsby}@google.com

ABSTRACT

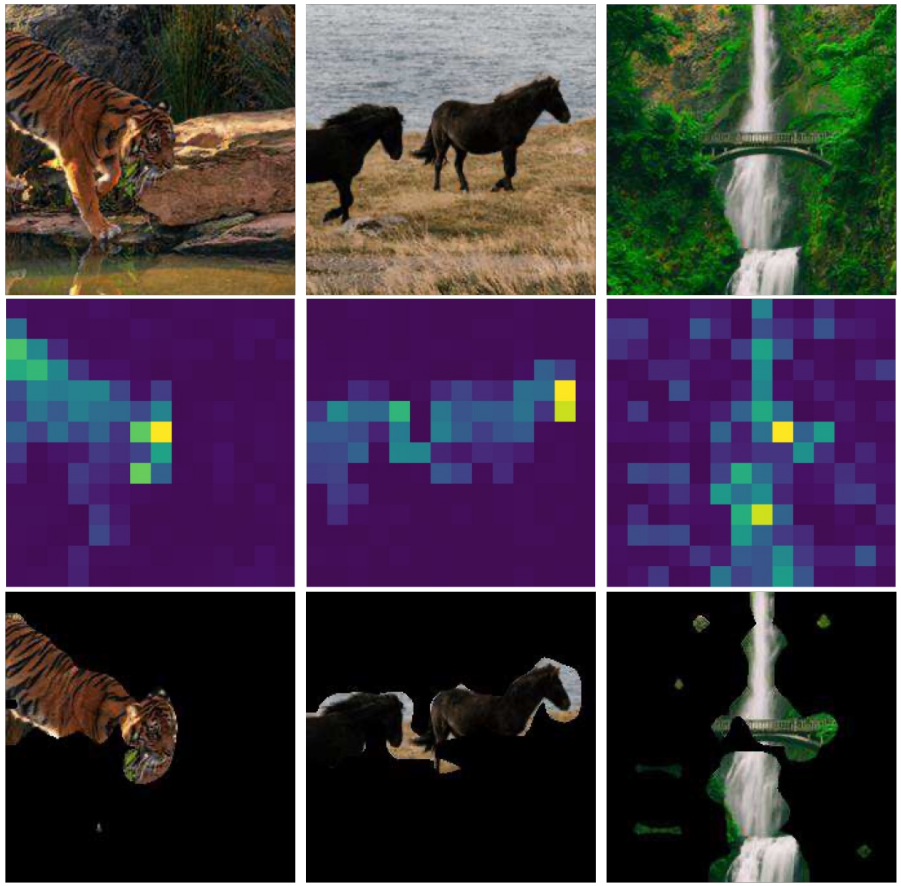
While the Transformer architecture has become the de-facto standard for natural language processing tasks, its applications to computer vision remain limited. In vision, attention is either applied in conjunction with convolutional networks, or used to replace certain components of convolutional networks 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

Vision Transformer (ViT)

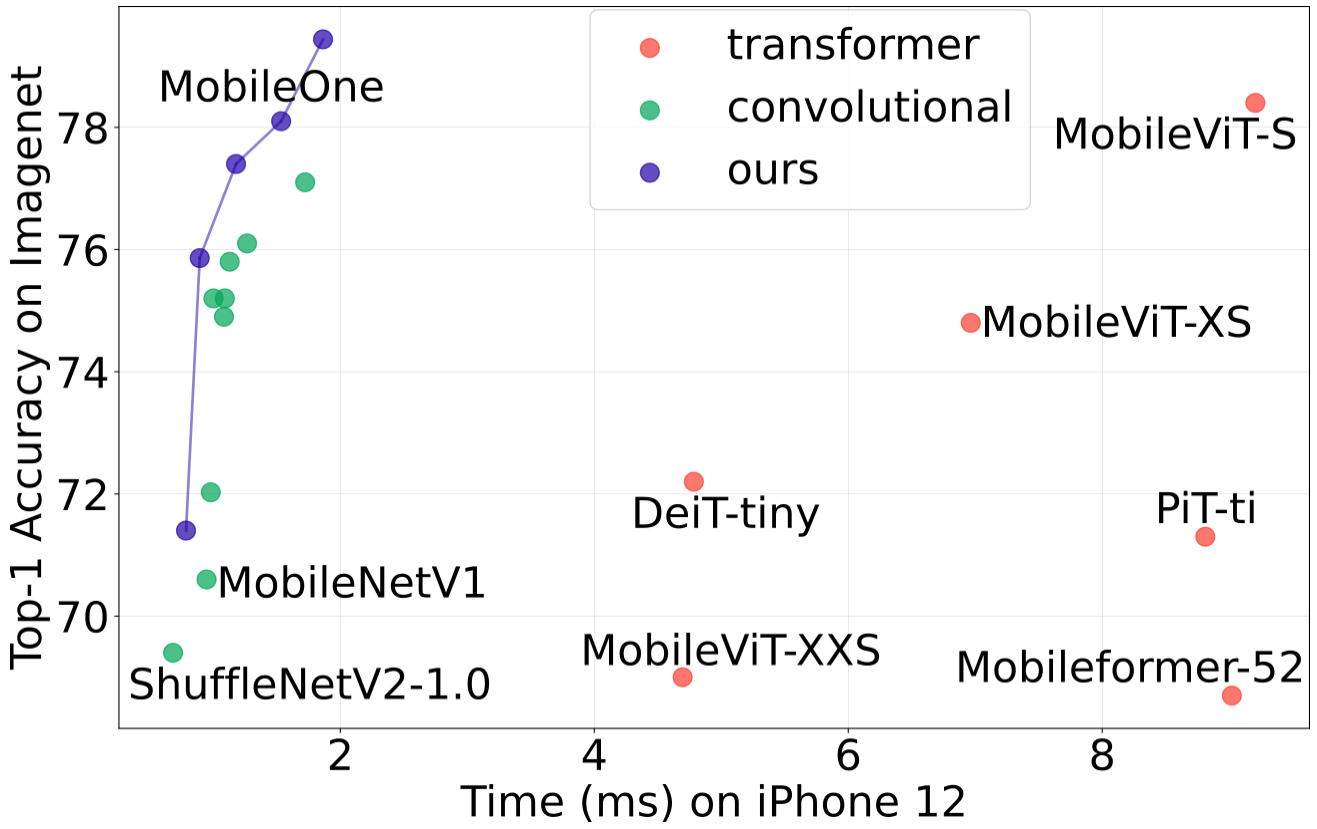


Prince “Understanding Deep Learning” 2023.

Visualizing Attention

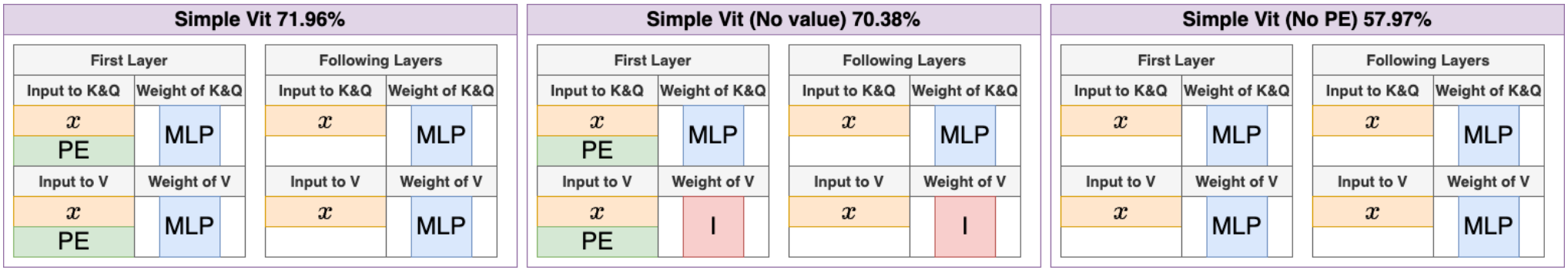


Transformers are Getting Faster!!!



Vasu et al. “MobileOne: An Improved One millisecond Mobile Backbone” In CVPR 2023.

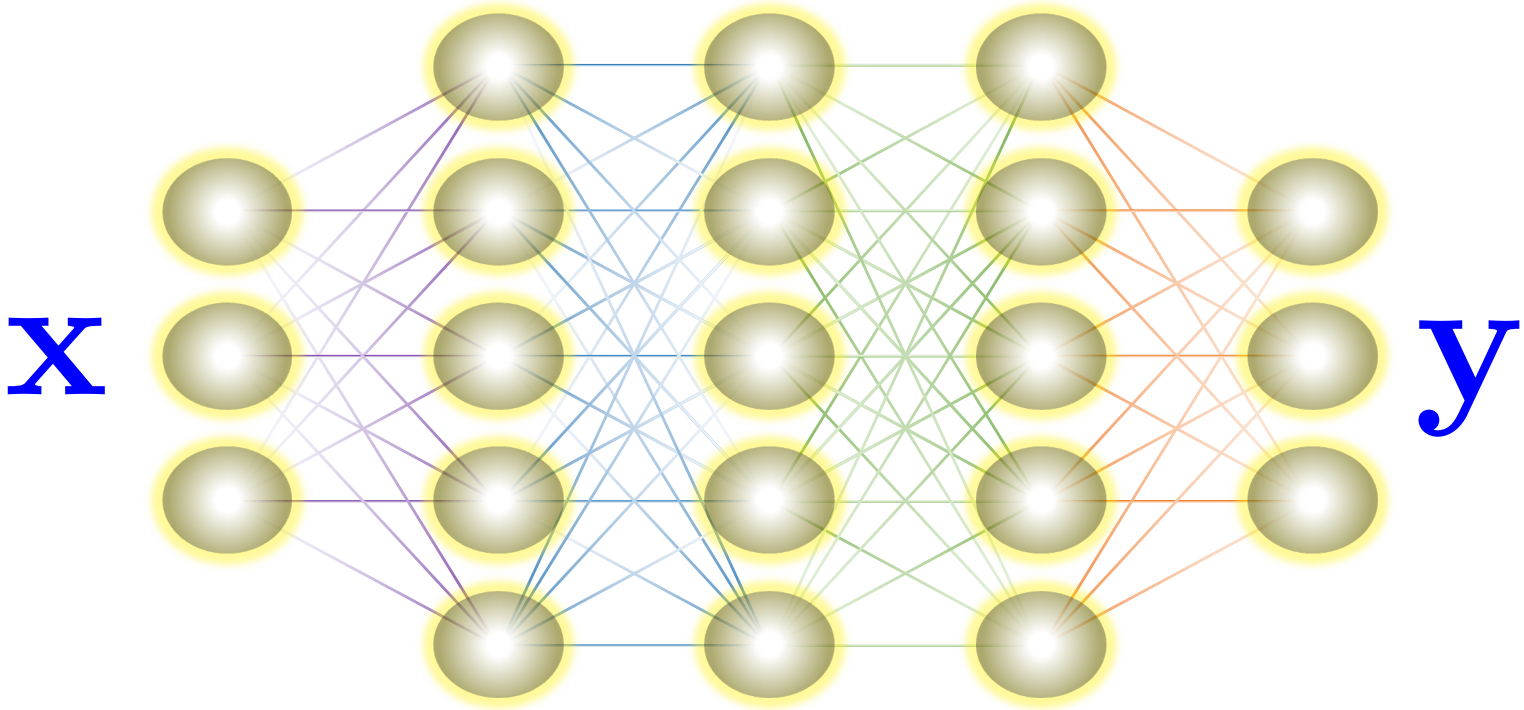
Let's have a play

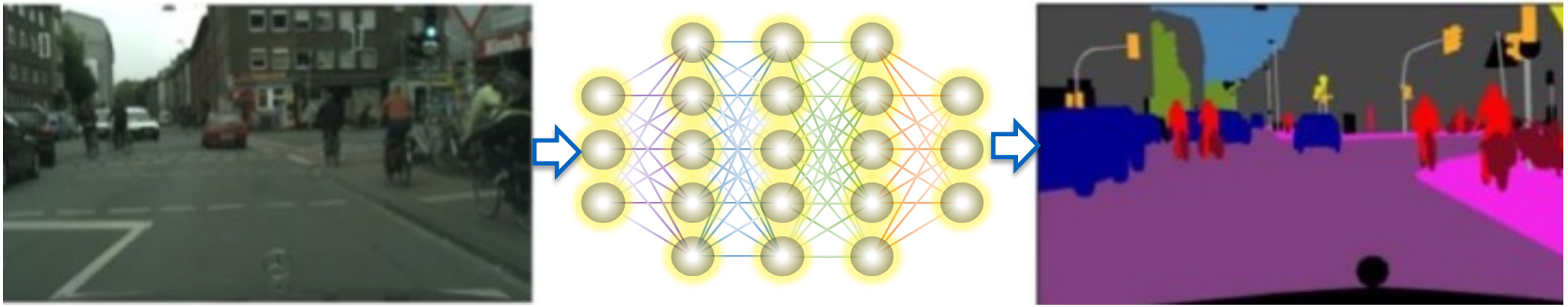


https://colab.research.google.com/github/slucey-cs-cmu-edu/RVSS24/blob/main/Simple_ViT_in_Colab.ipynb

Today

- Tokenization and ConvNets
- Skip Connections and BatchNorm
- Attention and Transformers
- **Self Supervision**

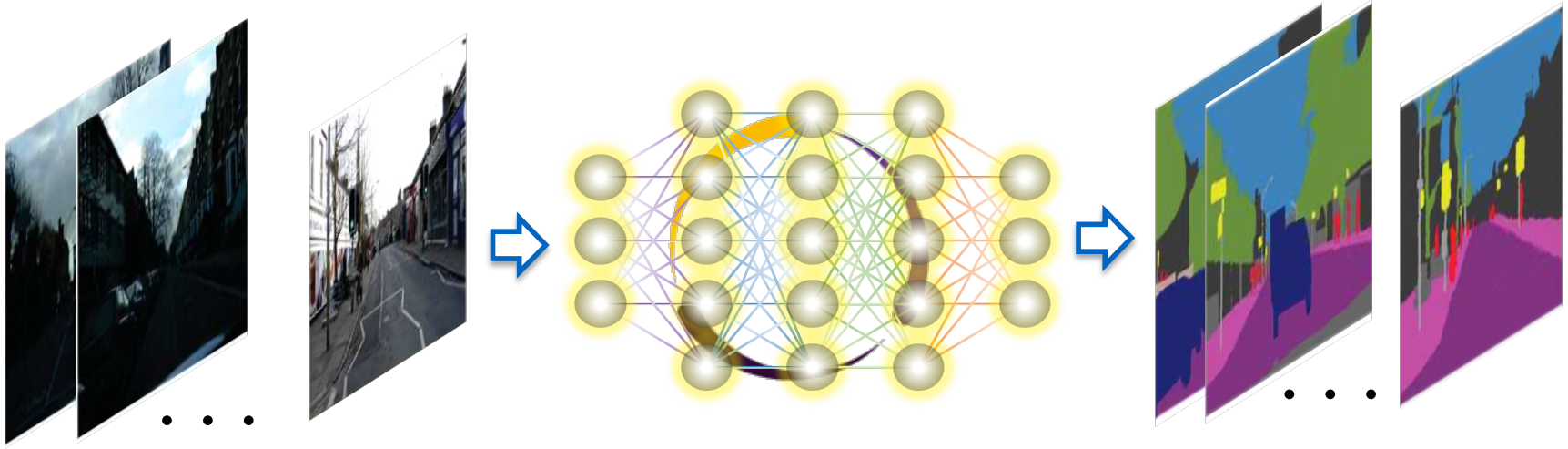

x
y



x

y

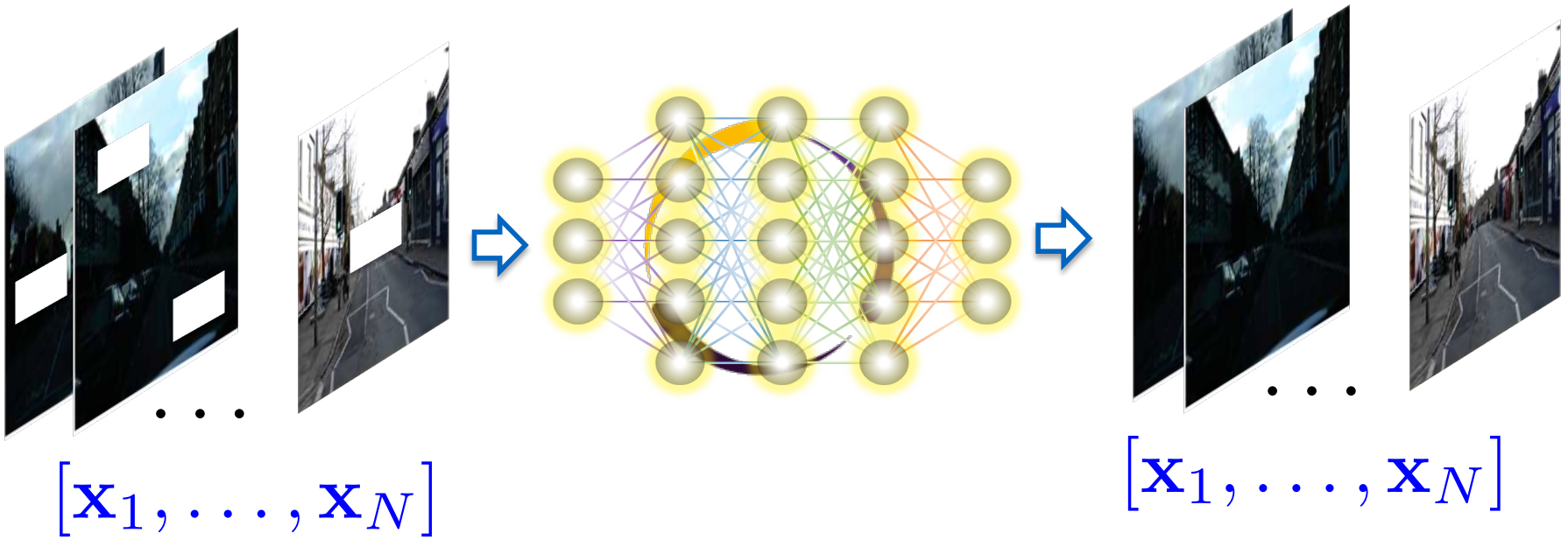
Where do I get the labels?



$[\mathbf{x}_1, \ldots, \mathbf{x}_N]$ $[\mathbf{y}_1, \ldots, \mathbf{y}_N]$

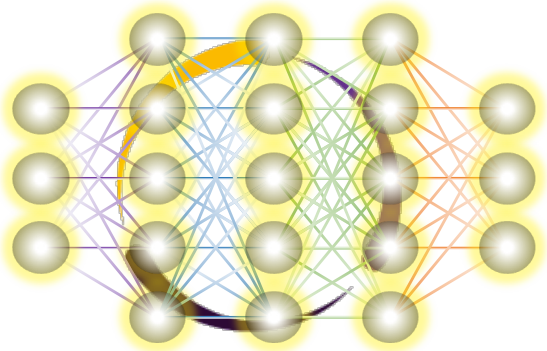
$$N \propto 10^6 - 10^9$$

Requires huge datasets and compute!!!



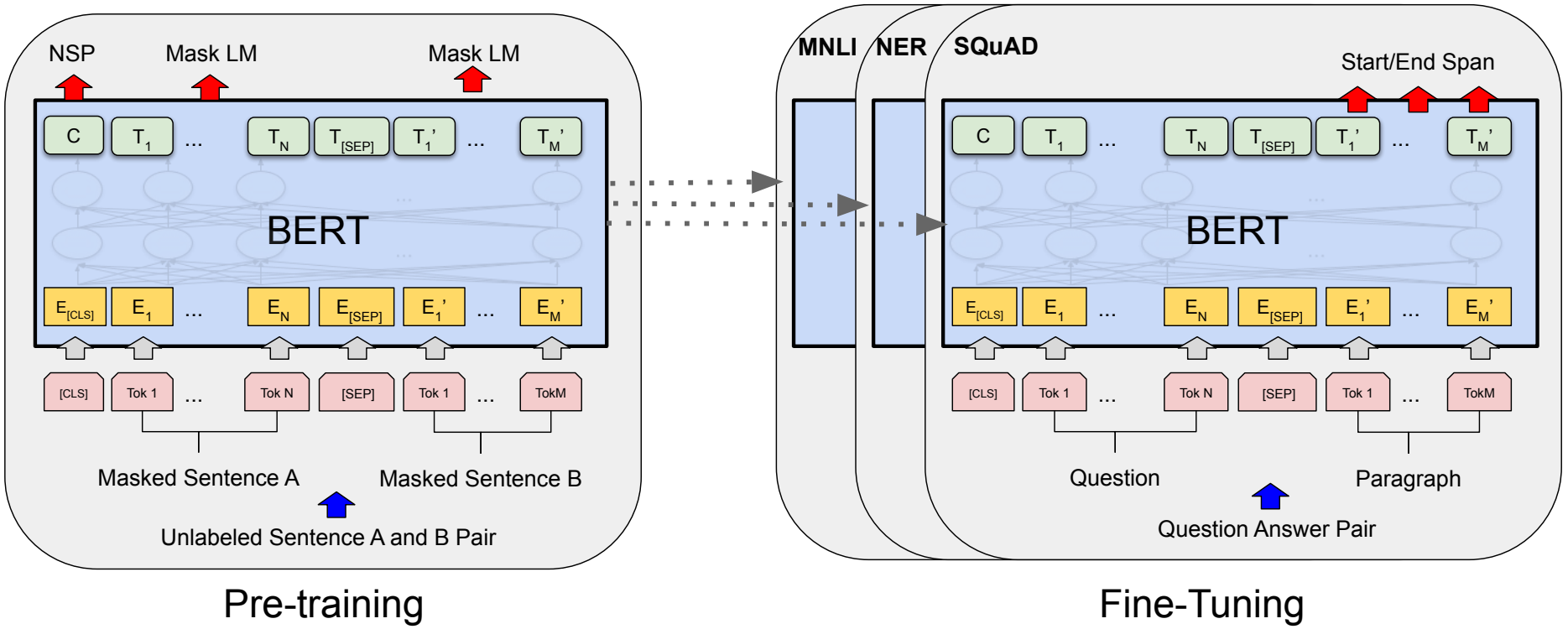
Self Supervision

“The [redacted] walked across the [redacted]n”



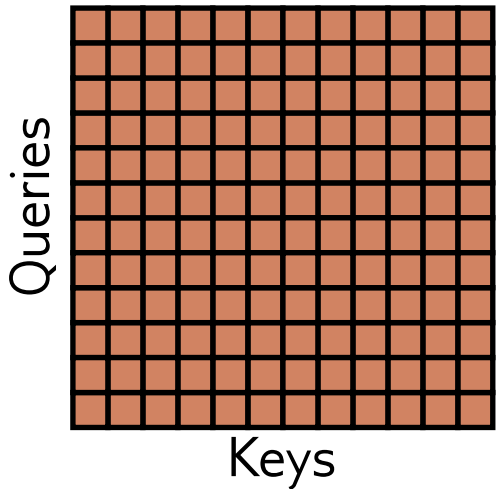
“The cat walked across the room”

Self Supervision

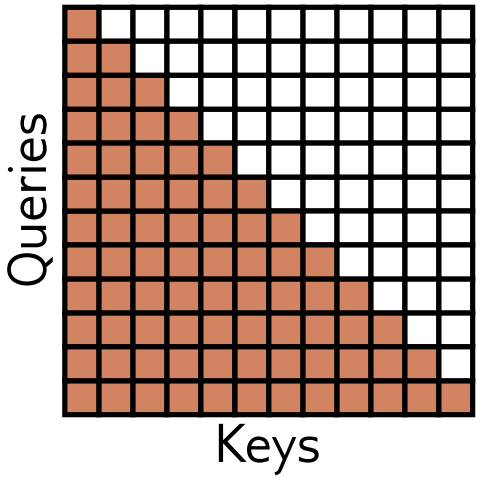


J. Devlin, et al. “BERT: Pre-training of deep bidirectional transformers for language understanding”. In NAACL, 2019..

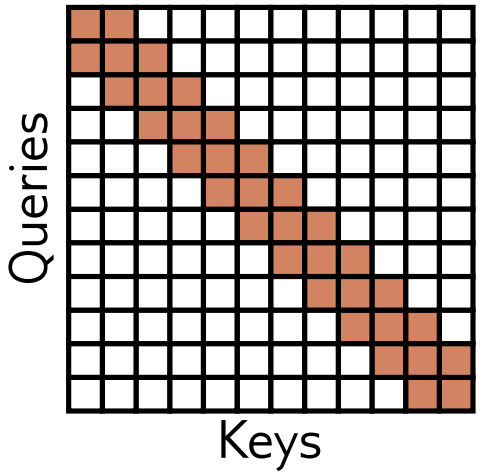
Interaction Matrices for Self Attention



Encoder



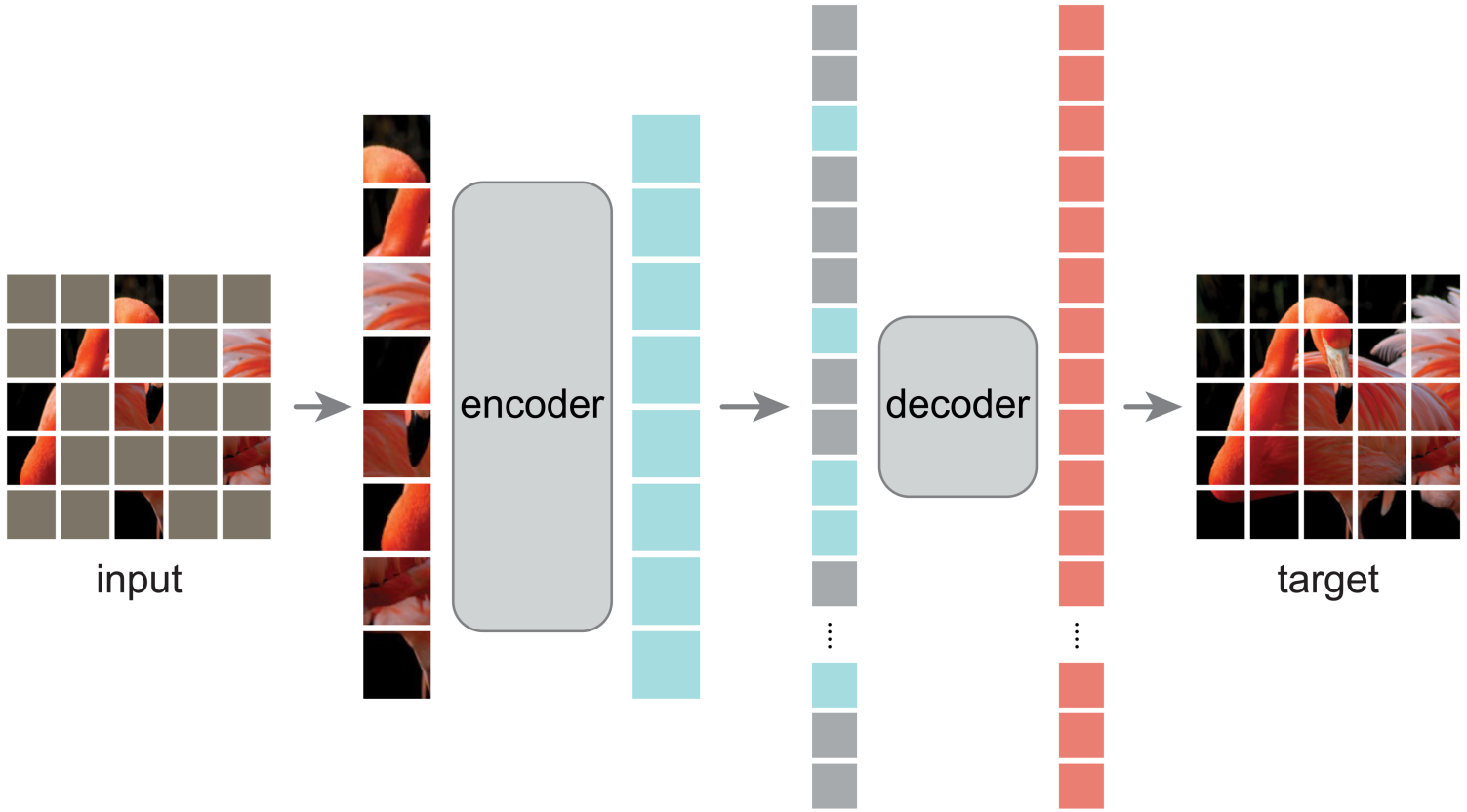
Decoder



Convolutional Encoder

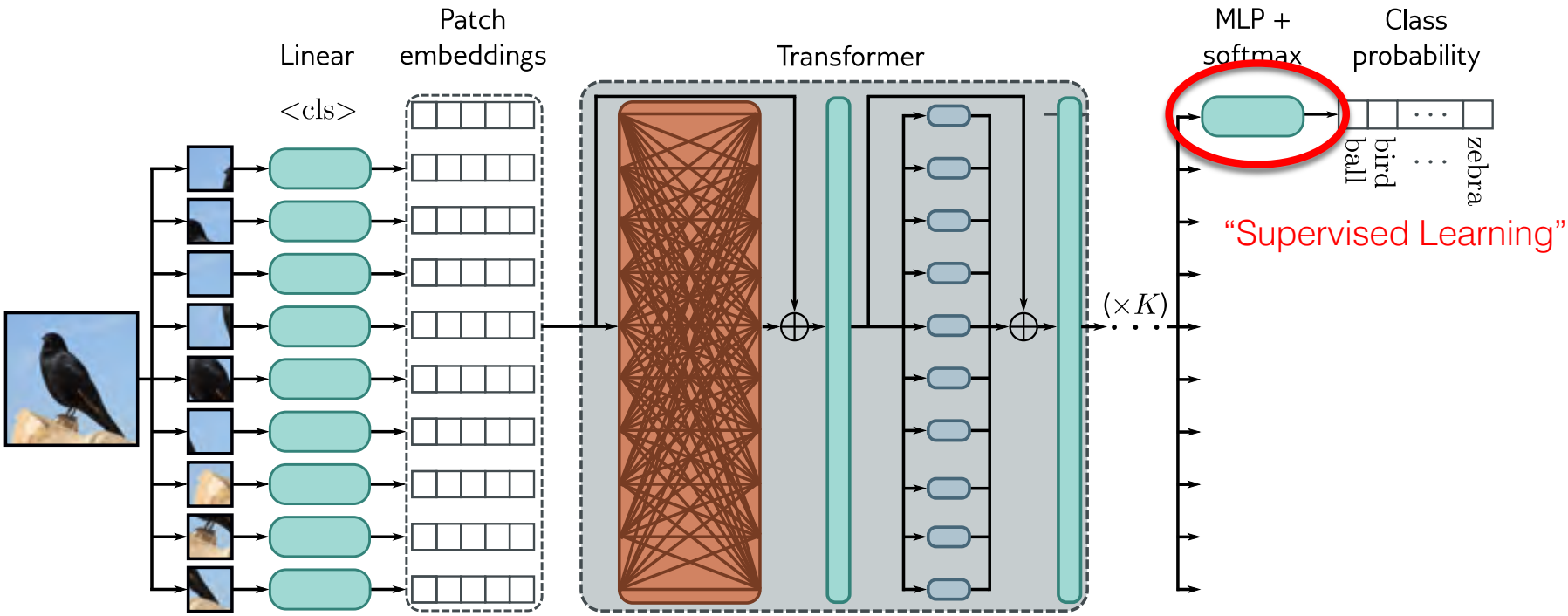
Prince “Understanding Deep Learning” 2023.

Masked ViT



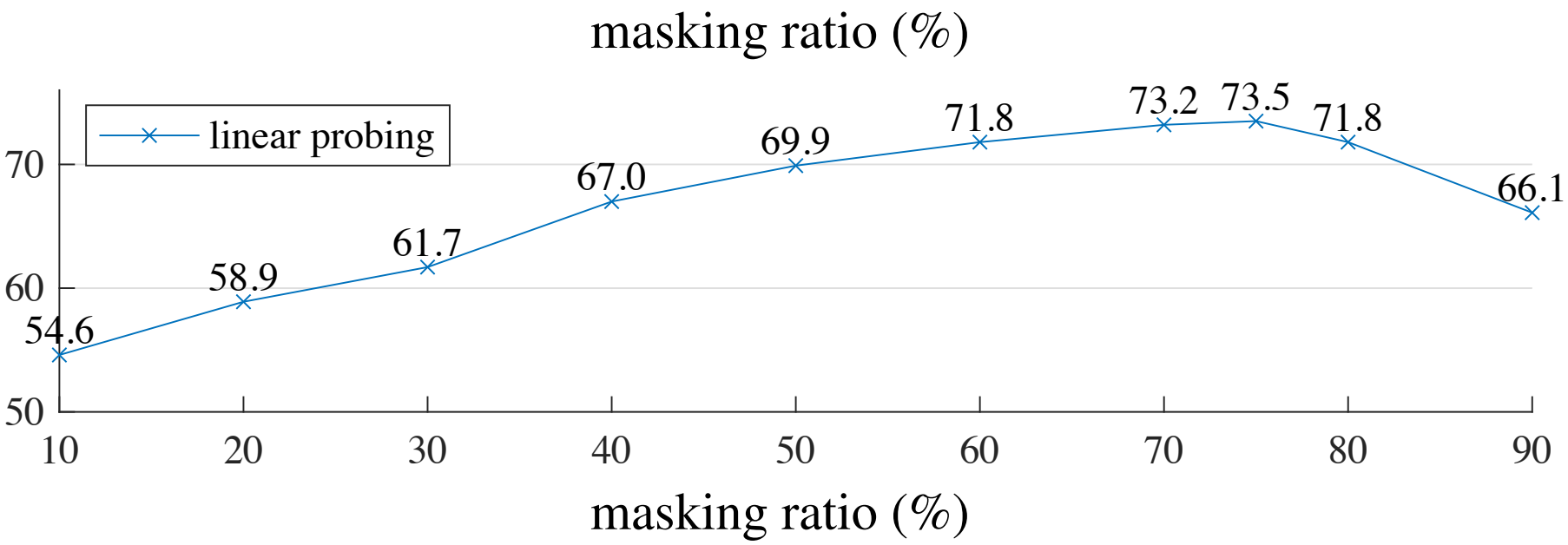
K. He, et al. “Masked Autoencoders Are Scalable Vision Learners” In CVPR 2022.

Masked ViT



Prince "Understanding Deep Learning" 2023.

Masked ViT



K. He, et al. “Masked Autoencoders Are Scalable Vision Learners” In CVPR 2022.