Textual Information Assisted Transfer Learning in Visual Tasks

background

Presented by Zhuowen Shen

February 10, 2021

Motivation

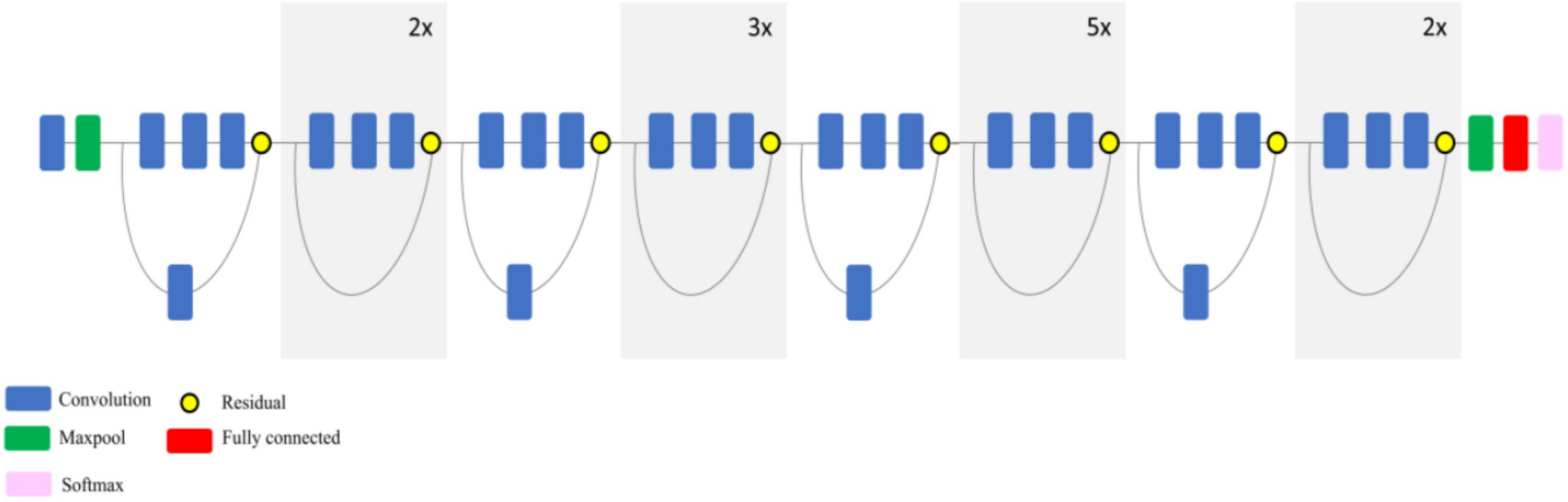
- **Difficulty in acquiring big dataset with high quality**
  - Human labeling is expensive and time consuming
  - Each dataset can only be used for specific tasks

- **Good feature representations in the previous SOTA Deep Neural Networks**
  - Proved good performance on their own tasks and representations
  - Pre-trained weights on big dataset

- **Proved success of Natural Language Processing(NLP) architecture in visual tasks.**
  - Attention and Transformer
  - Long Short-Term Memory

Transfer Learning

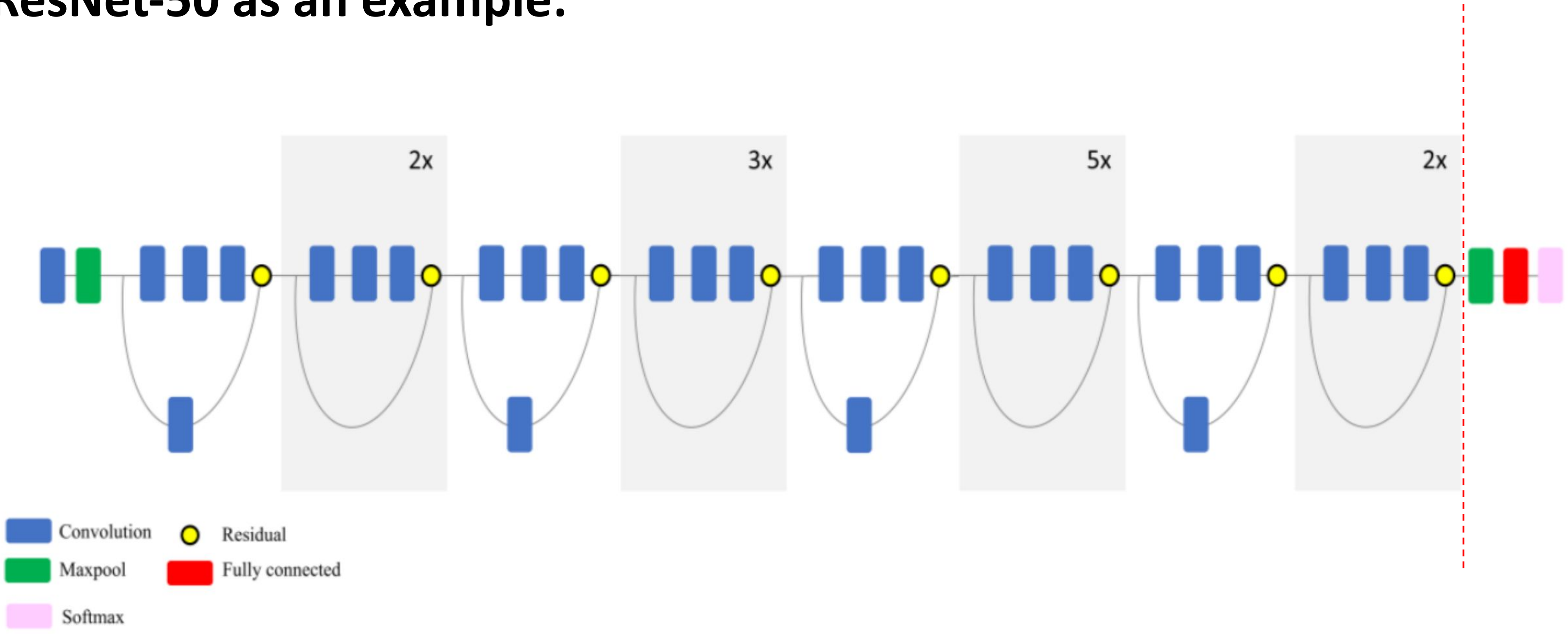
- **Take ResNet-50 as an example:**



Masoud et al, “Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery”, ResearchGate

Transfer Learning

- **Take ResNet-50 as an example:**

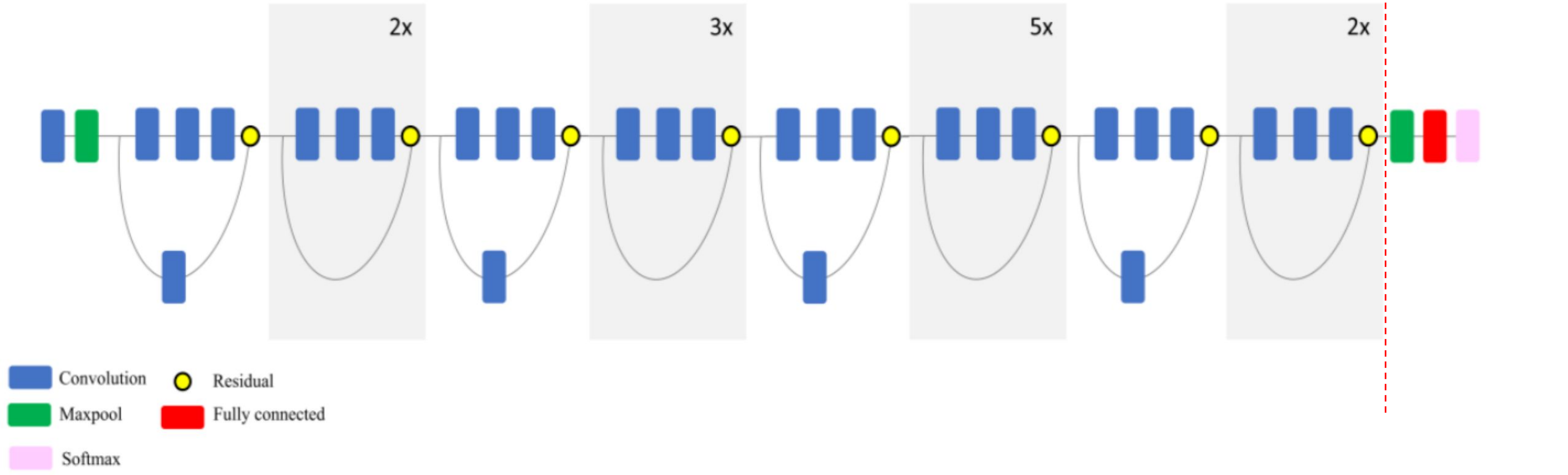


Masoud et al, “Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery”, ResearchGate

Transfer Learning

- **Take ResNet-50 as an example:**

representation
learned from
big datasets



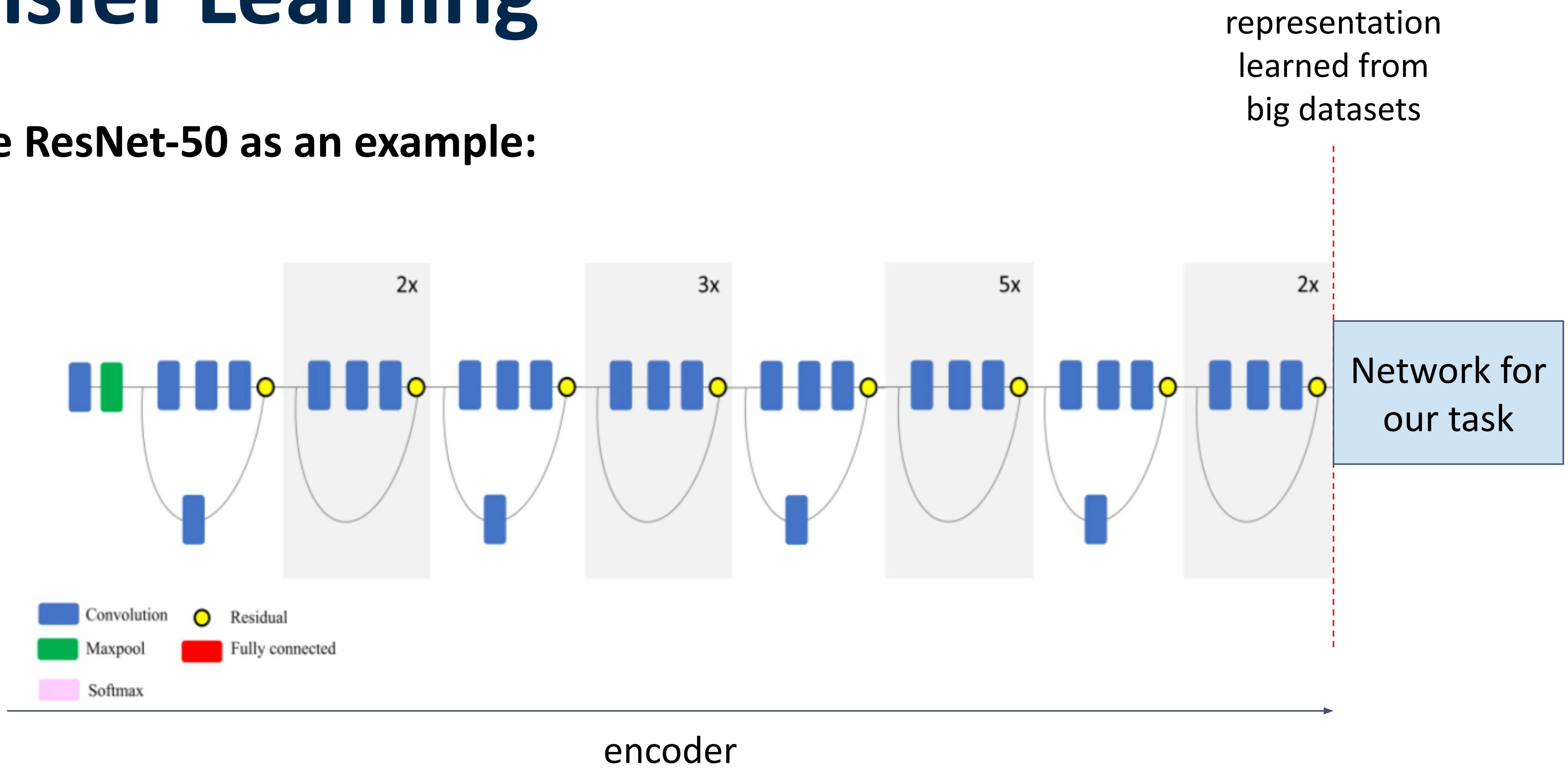
encoder

block for
specific task

Masoud et al, “Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery”, ResearchGate

Transfer Learning

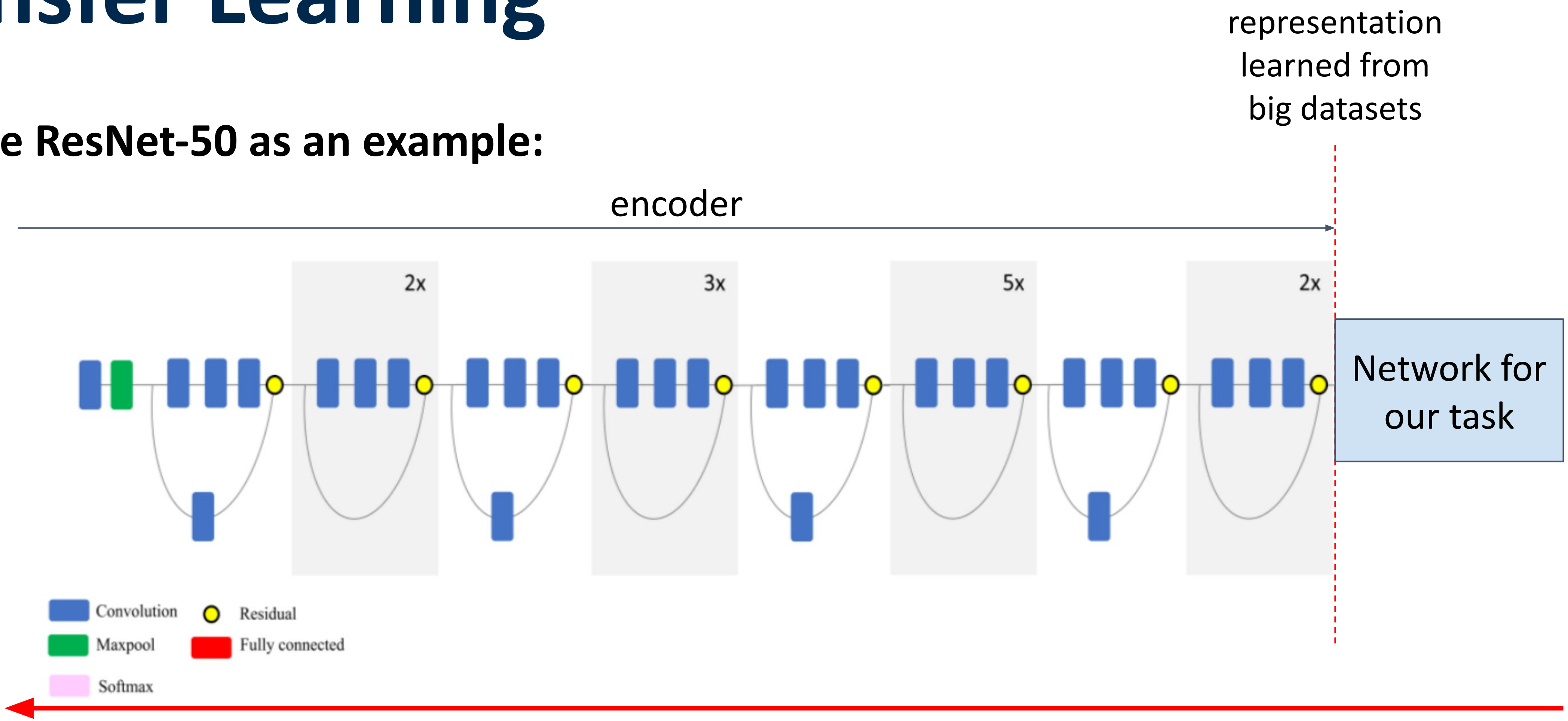
- **Take ResNet-50 as an example:**



Masoud et al, "Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery", ResearchGate

Transfer Learning

- **Take ResNet-50 as an example:**



Fine tune on the target dataset: freeze the encoder and only update for our new network block

Masoud et al, “Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery”, ResearchGate

Transfer Learning

- **Pros:**
  - Achieve training by small of datasets
  - Save training time/has better training results
  - More applicable for practical use

Transfer Learning

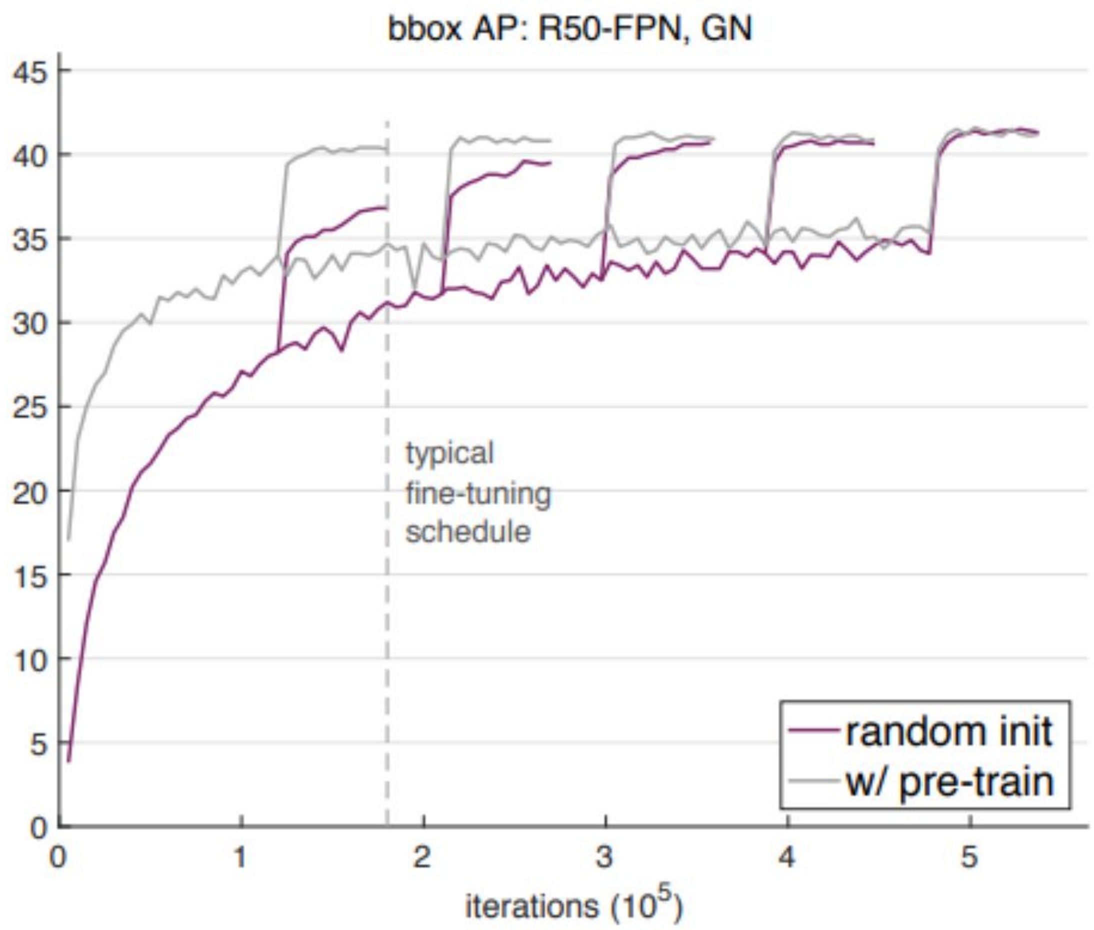
- **Pros:**
  - Achieve training by small of datasets
  - Save training time/has better training results
  - More applicable for practical use

Transfer learning will be the next driver of machine learning's commercial success after supervised learning.

-Andrew Ng

Transfer Learning

- **Potential cons:**
  - Random init will reach the same performance with more iterations
  - At high iterations, random init has even better performance



He et al, ”Rethinking ImageNet Pre-Training”, ICCV 2019

Attention and Transformer

- **Attention layer**
  - In textual message, we’d like to have each word to not only encode itself but also look at other words to have better encodings:
    - *“The animal didn't cross the street because it was too tired”*

Attention and Transformer

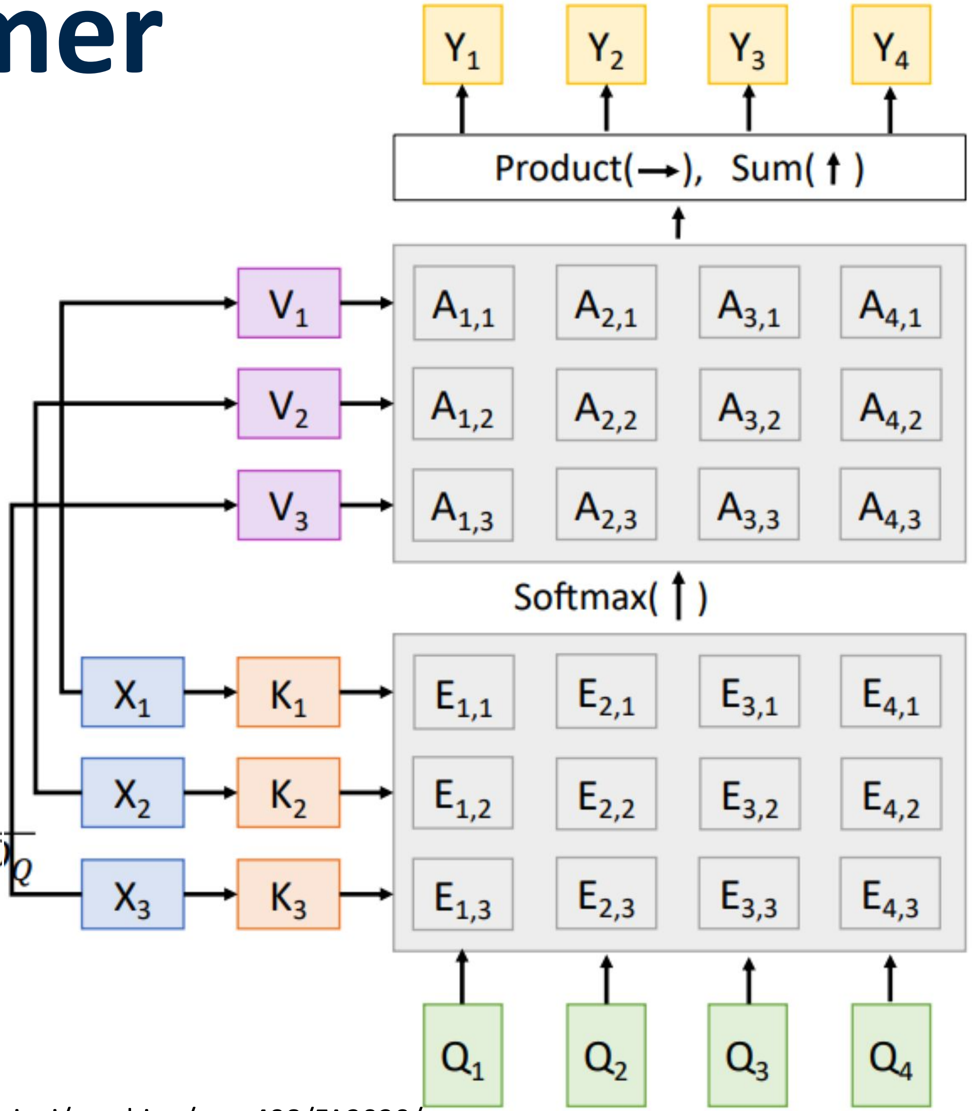
- **Attention layer**
  - In textual message, we’d like to have each word to not only encode itself but also look at other words to have better encodings:
    - *“The animal didn't cross the street because **it** was too tired”*
  - We want some networks that can have attentions between query and key.

Attention and Transformer

- **Attention layer**

<u>**Inputs**</u>:
Query vectors: Q (Shape: $N_Q \times D_Q$)
Input vectors: X (Shape: $N_X \times D_X$)
Key matrix: W_K (Shape: $D_X \times D_Q$)
Value matrix: W_V (Shape: $D_X \times D_V$)

<u>**Computation**</u>:
Key vectors: $K = XW_K$ (Shape: $N_X \times D_Q$)
Value Vectors: $V = XW_V$ (Shape: $N_X \times D_V$)
Similarities: $E = QK^T / \sqrt{D_Q}$ (Shape: $N_Q \times N_X$) $E_{i,j} = (Q_i \cdot K_j) / \sqrt{D_Q}$
Attention weights: $A = \text{softmax}(E, \text{dim}=1)$ (Shape: $N_Q \times N_X$)
Output vectors: $Y = AV$ (Shape: $N_Q \times D_V$) $Y_i = \sum_j A_{i,j} V_j$



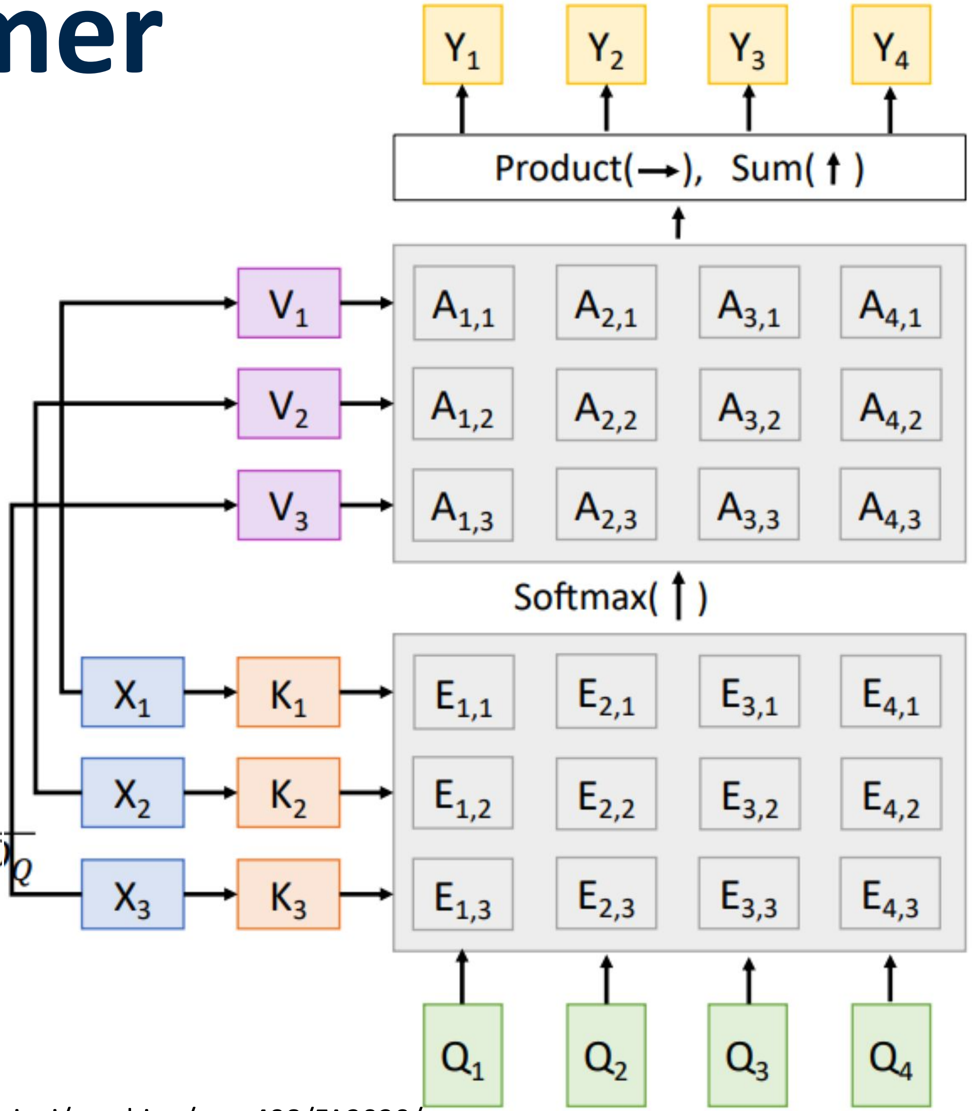
Justin Johnson, EECS 498 Deep Learning in Computer Vision, Fall 2020, https://web.eecs.umich.edu/~justincj/teaching/eecs498/FA2020/

Attention and Transformer

- **Attention layer**

<u>**Inputs**</u>:
Query vectors: **Q** (Shape: N_Q x D_Q)
Input vectors: **X** (Shape: N_X x D_X)
Key matrix: $\mathbf{W_K}$ (Shape: D_X x D_Q)
Value matrix: $\mathbf{W_V}$ (Shape: D_X x D_V)

<u>**Computation**</u>:
Key vectors: **K** = $\mathbf{XW_K}$ (Shape: N_X x D_Q)
Value Vectors: **V** = $\mathbf{XW_V}$ (Shape: N_X x D_V)
Similarities: E = $\mathbf{QK}^T / \sqrt{D_Q}$ (Shape: N_Q x N_X) $E_{i,j} = (\mathbf{Q_i} \cdot \mathbf{K_j}) / \sqrt{D_Q}$
Attention weights: A = softmax(E, dim=1) (Shape: N_Q x N_X)
Output vectors: Y = A**V** (Shape: N_Q x D_V) $Y_i = \sum_j A_{i,j} \mathbf{V_j}$



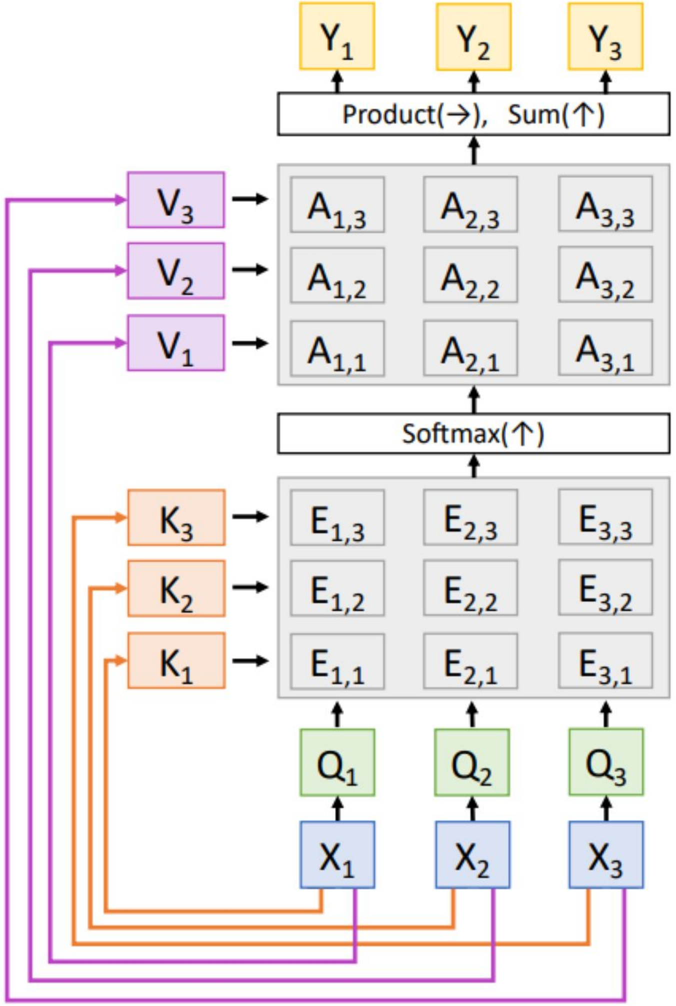
Justin Johnson, EECS 498 Deep Learning in Computer Vision, Fall 2020, https://web.eecs.umich.edu/~justincj/teaching/eecs498/FA2020/

Attention and Transformer

- **Self-Attention layer**

Inputs:
Input vectors: X (Shape: $N_X \times D_X$)
Key matrix: W_K (Shape: $D_X \times D_Q$)
Value matrix: W_V (Shape: $D_X \times D_V$)
Query matrix: W_Q (Shape: $D_X \times D_Q$)

Computation:
Query vectors: $Q = XW_Q$
Key vectors: $K = XW_K$ (Shape: $N_X \times D_Q$)
Value Vectors: $V = XW_V$ (Shape: $N_X \times D_V$)
Similarities: $E = QK^T / \sqrt{D_Q}$ (Shape: $N_X \times N_X$) $E_{i,j} = (Q_i \cdot K_j) / \sqrt{D_Q}$
Attention weights: $A = \text{softmax}(E, \text{dim}=1)$ (Shape: $N_X \times N_X$)
Output vectors: $Y = AV$ (Shape: $N_X \times D_V$) $Y_i = \sum_j A_{i,j} V_j$



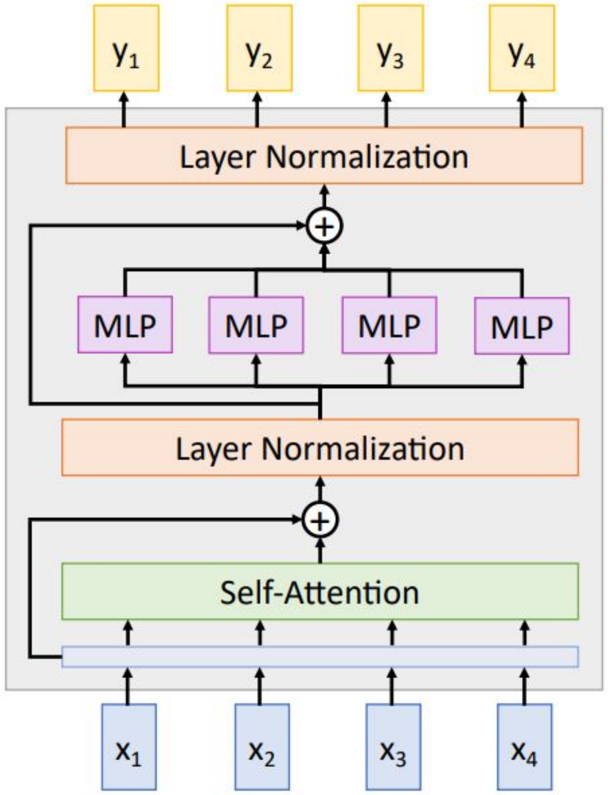
Justin Johnson, EECS 498 Deep Learning in Computer Vision, Fall 2020, https://web.eecs.umich.edu/~justincj/teaching/eecs498/FA2020/

Attention and Transformer

- **Attention layer**
  - Pros:
    - Good at long sequences: after one self-attention layer, each output "sees" all inputs!
    - Highly parallel: Each output can be computed in parallel
  - Cons:
    - Very memory intensive: huge weight matrix; if you want the model to have different attention(multi-head), you have to train them separately.

Attention and Transformer

- **Transformer:**
  - A model that uses attention to achieve parallelization:
    - translate the whole sentence
    - image caption with a phrase



Justin Johnson, EECS 498 Deep Learning in Computer Vision, Fall 2020, https://web.eecs.umich.edu/~justincj/teaching/eecs498/FA2020/

Attention and Transformer

- **Proved success in visual tasks**
  - Local multi-head dot-product self attention blocks can completely replace convolutions.
  - Sparse Transformers employ scalable approximations to global self-attention in order to be applicable to images.

Dosovitskiy et al, "An image is worth 16x16 words: Transformers for image recognition at scale". arXiv preprint arXiv:2010.11929, 2020.