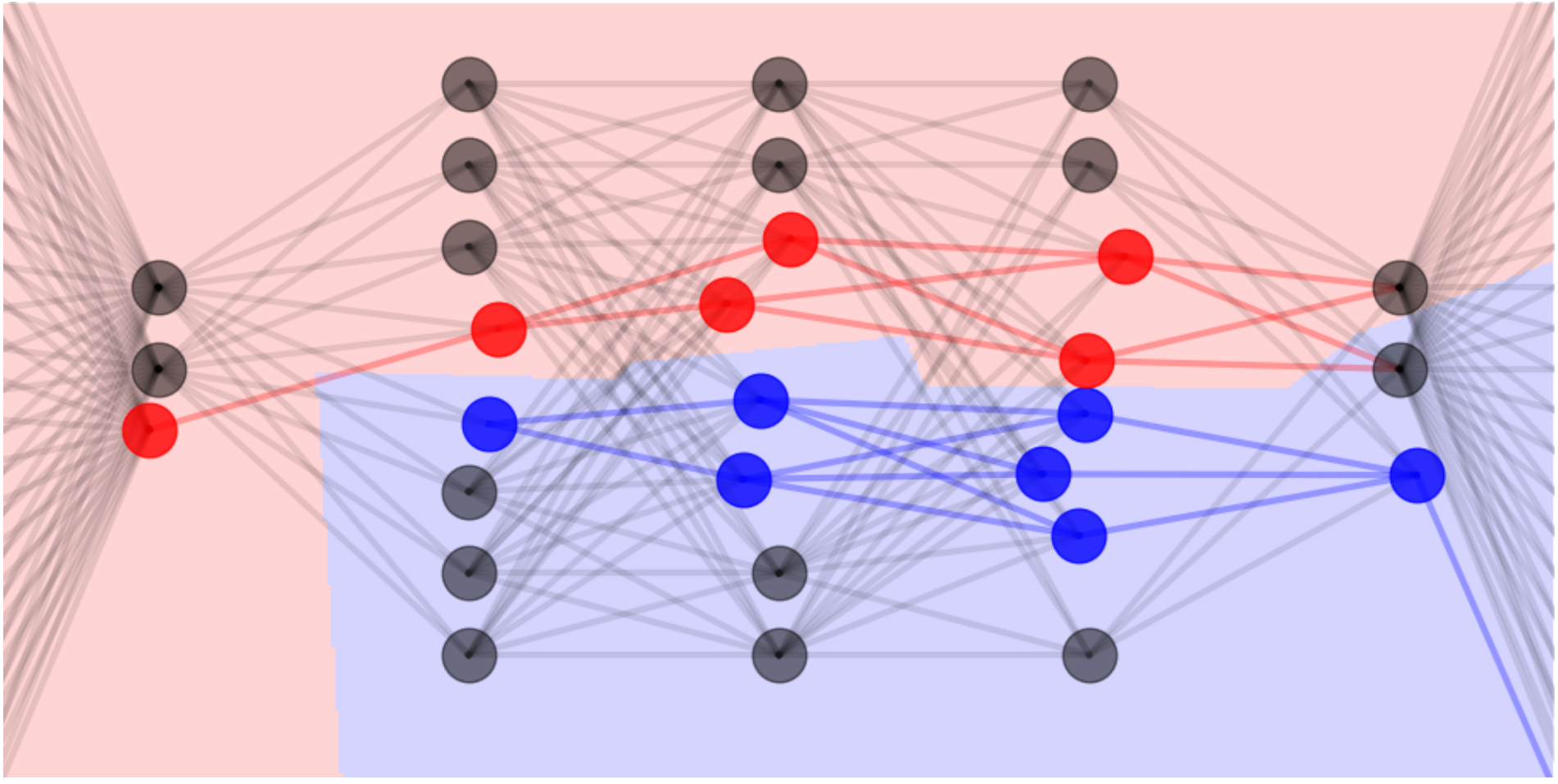
TU-Delft Machine and Deep Learning



Foundation models

Lecture 9

Lecturer: Jan van Gemert

Topic: Foundation models

- Foundation models: pre-training and adapting
- Transformer architecture
- Pre-training
- Adapting to new settings (fine-tuning)

Foundation models

Paper: `https://crfm.stanford.edu/report.html`

- Any model trained on broad data that can be adapted (fine-tuned) to a wide range of downstream tasks.

Foundation models

Paper: `https://crfm.stanford.edu/report.html`

- Any model trained on broad data that can be adapted (fine-tuned) to a wide range of downstream tasks.
- From a technical point of view not new; yet, scale and scope has exceeded imagination (eg. chatGPT)

Foundation models

Paper: `https://crfm.stanford.edu/report.html`

- Any model trained on broad data that can be adapted (fine-tuned) to a wide range of downstream tasks.
- From a technical point of view not new; yet, scale and scope has exceeded imagination (eg. chatGPT)
- At the same time: potential to accentuate harms, and their characteristics are in general poorly understood

Foundation models

Paper: `https://crfm.stanford.edu/report.html`

- Any model trained on broad data that can be adapted (fine-tuned) to a wide range of downstream tasks.
- From a technical point of view not new; yet, scale and scope has exceeded imagination (eg. chatGPT)
- At the same time: potential to accentuate harms, and their characteristics are in general poorly understood
- Naming: ~~pre-trained model~~; ~~self-supervised model~~; ~~(large) language model~~; ~~general purpose model~~, ~~multi-purpose model~~, ~~task-agnostic model~~; Foundation model.

Emergence and homogenization

- Emergence: behavior implicitly induced rather than explicitly constructed;
- Homogenization: consolidation of ML methods across a wide range of applications

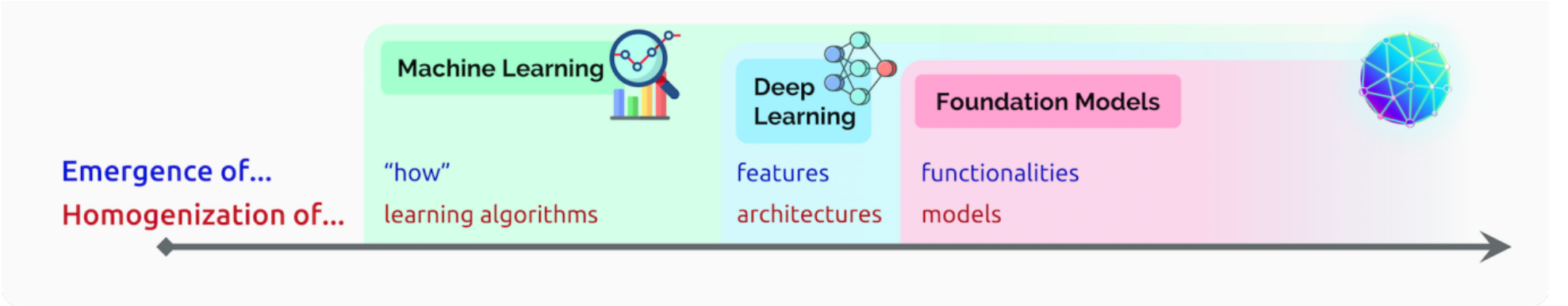


Fig. 1. The story of AI has been one of increasing *emergence* and *homogenization*. With the introduction of machine learning, *how* a task is performed emerges (is inferred automatically) from examples; with deep learning, the high-level features used for prediction emerge; and with foundation models, even advanced functionalities such as in-context learning emerge. At the same time, machine learning homogenizes learning algorithms (e.g., logistic regression), deep learning homogenizes model architectures (e.g., Convolutional Neural Networks), and foundation models homogenizes the model itself (e.g., GPT-3).

Homogenization: modalities/tasks

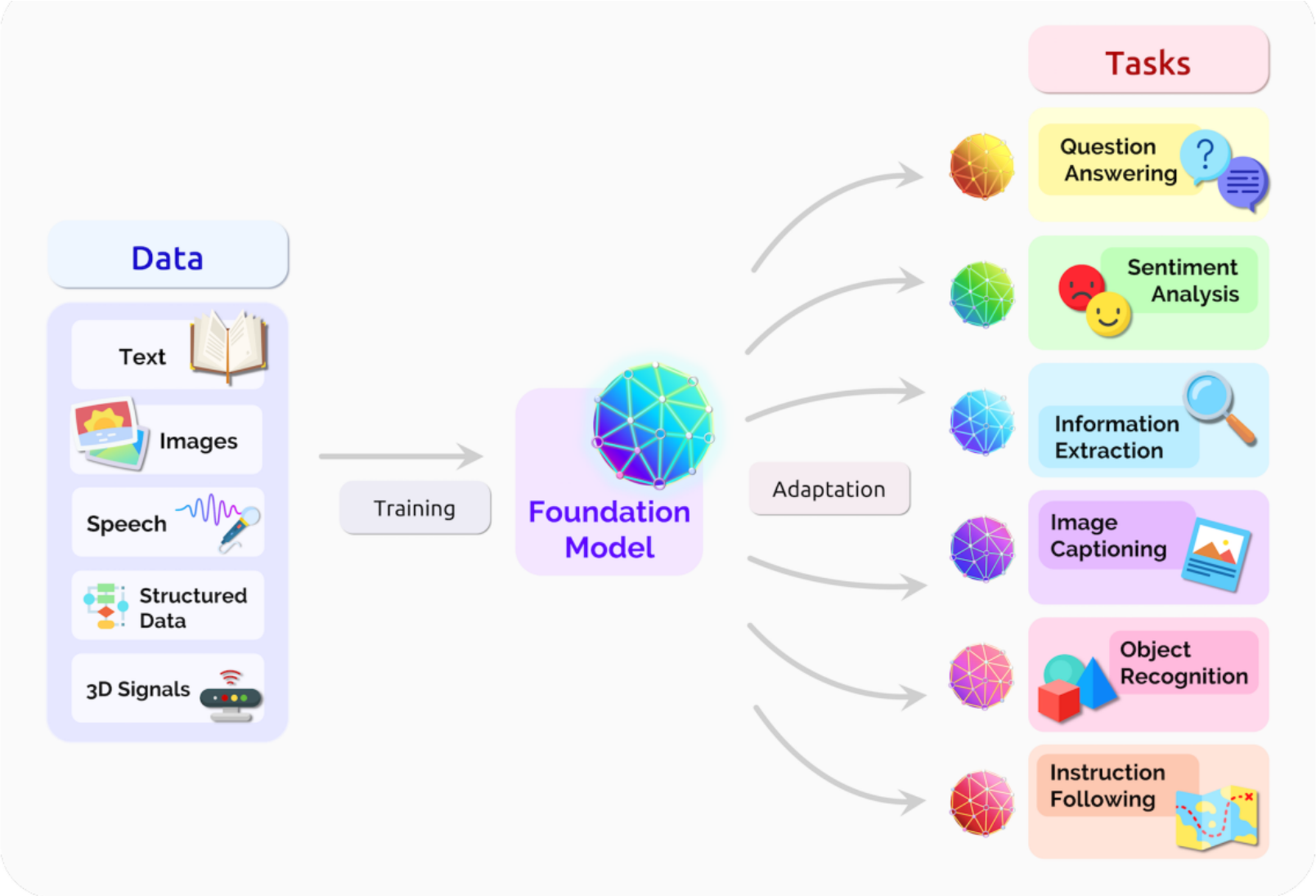


Fig. 2. A foundation model can centralize the information from all the data from various modalities. This one model can then be adapted to a wide range of downstream tasks.

Social impact

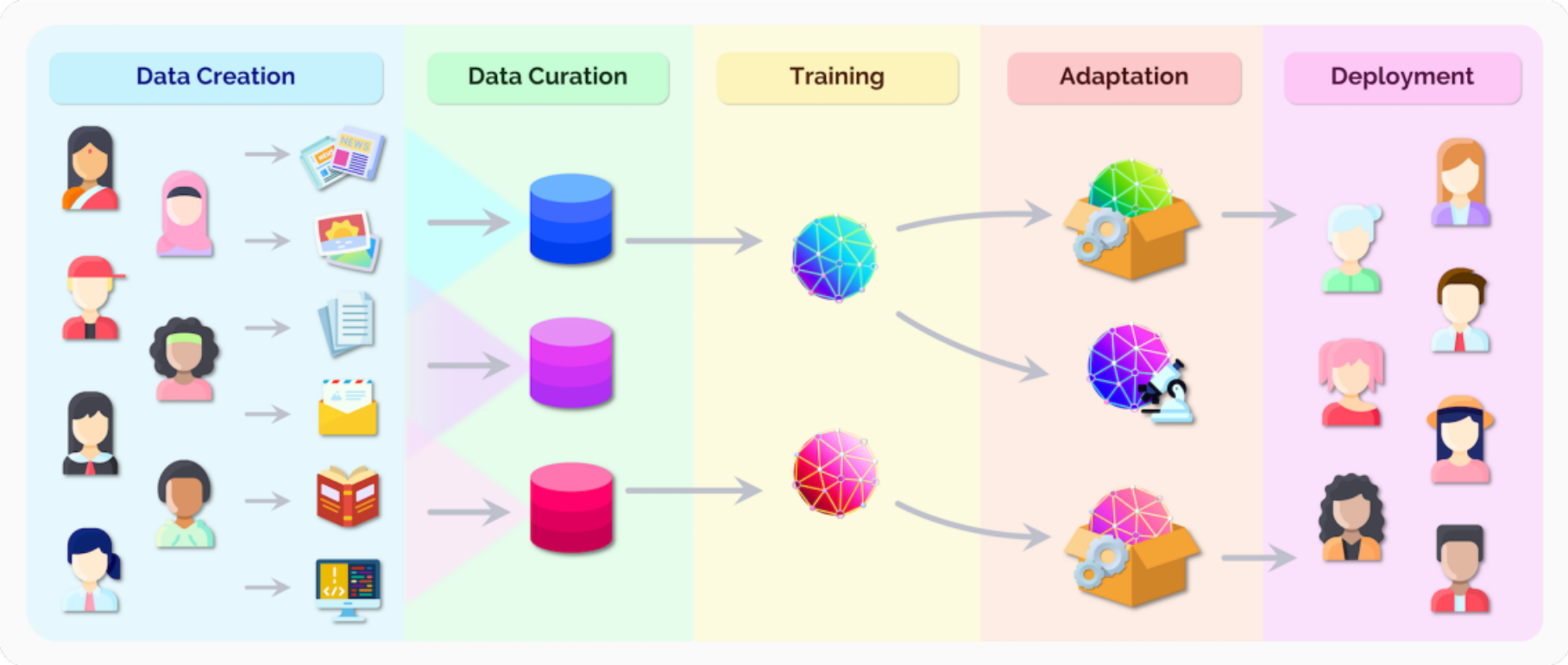


Fig. 3. Before reasoning about the social impact of foundation models, it is important to understand that they are part of a broader ecosystem that stretches from data creation to deployment. At both ends, we highlight the role of people as the ultimate source of data into training of a foundation model, but also as the downstream recipients of any benefits and harms. Thoughtful data curation and adaptation should be part of the responsible development of any AI system. Finally, note that the deployment of adapted foundation models is a decision separate from their construction, which could be for research.

Fairness; economics; environment, mis-use, legality, ethics.

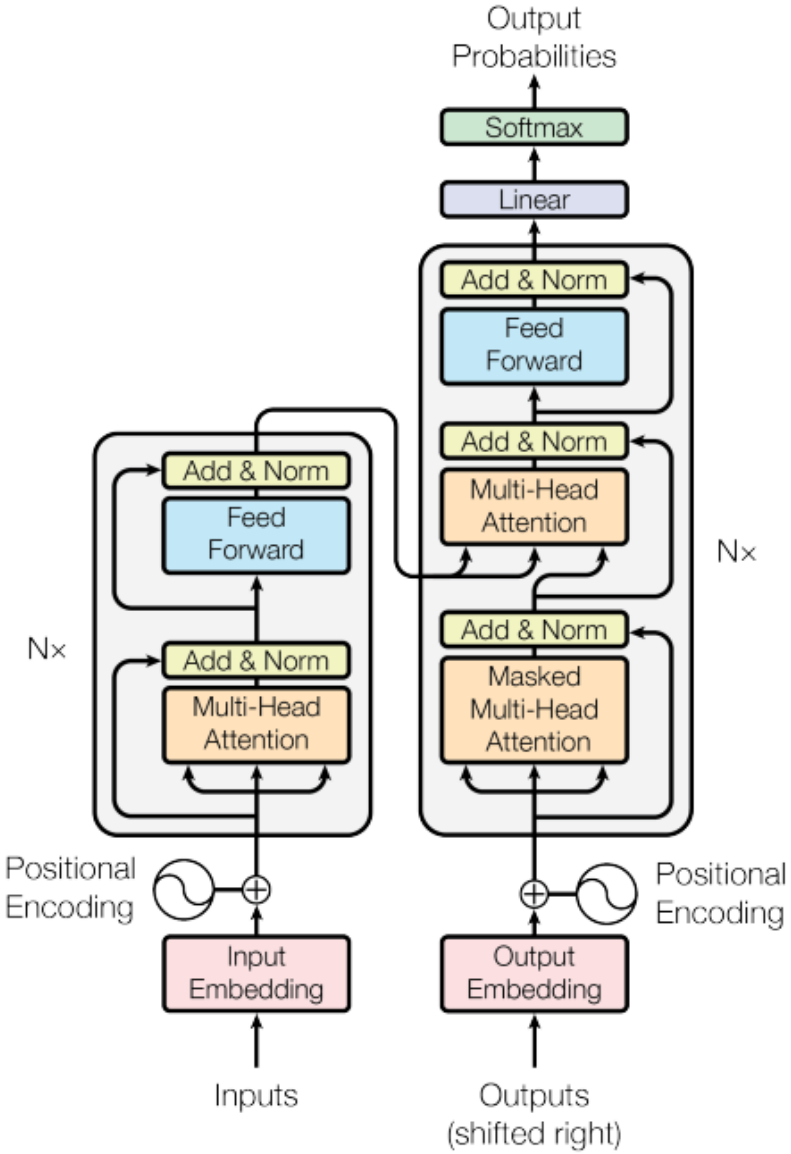
The rest of the paper...

Fig. 4. This report is divided into four parts: capabilities, applications, technology, and society, where each part contains a set of sections, and each section covers one aspect of foundation models.

Questions?

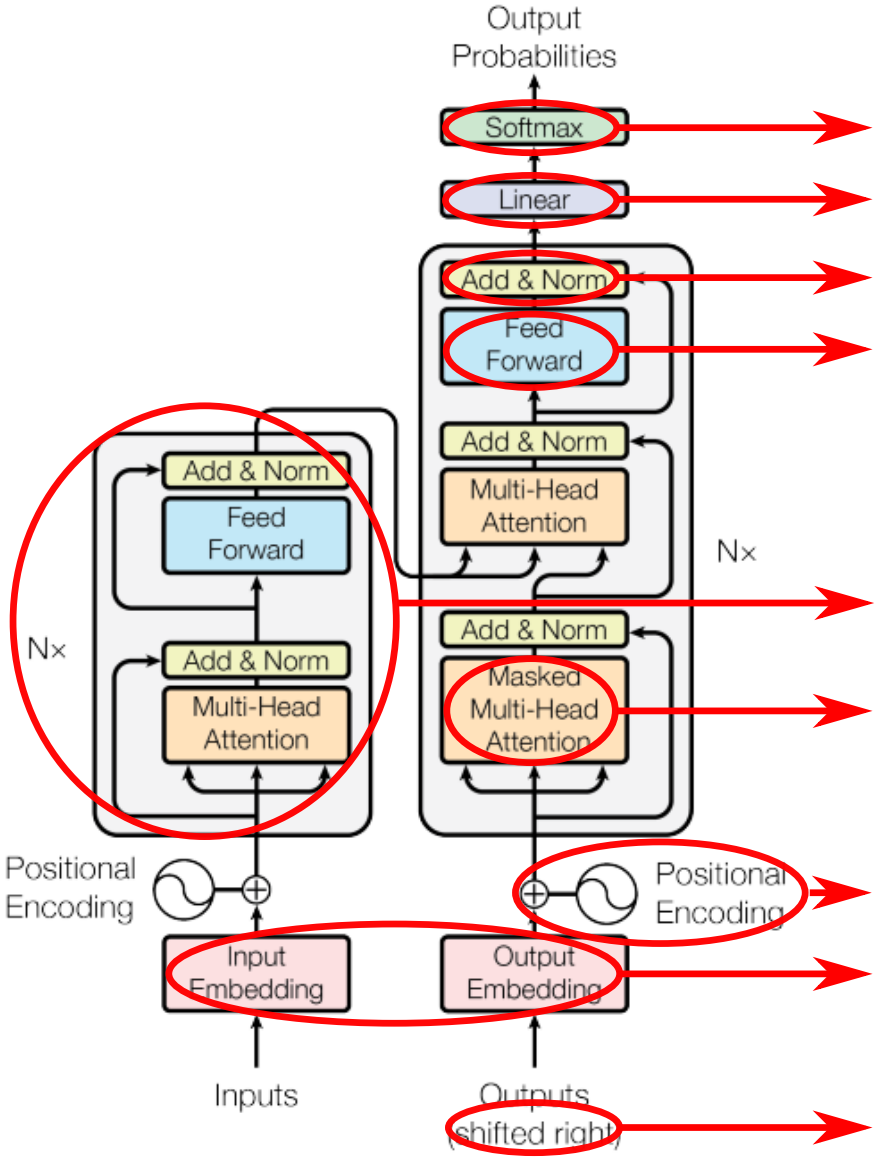
Attention is all you need: full *Transformer* architecture

Paper: `https://research.google/pubs/attention-is-all-you-need/`



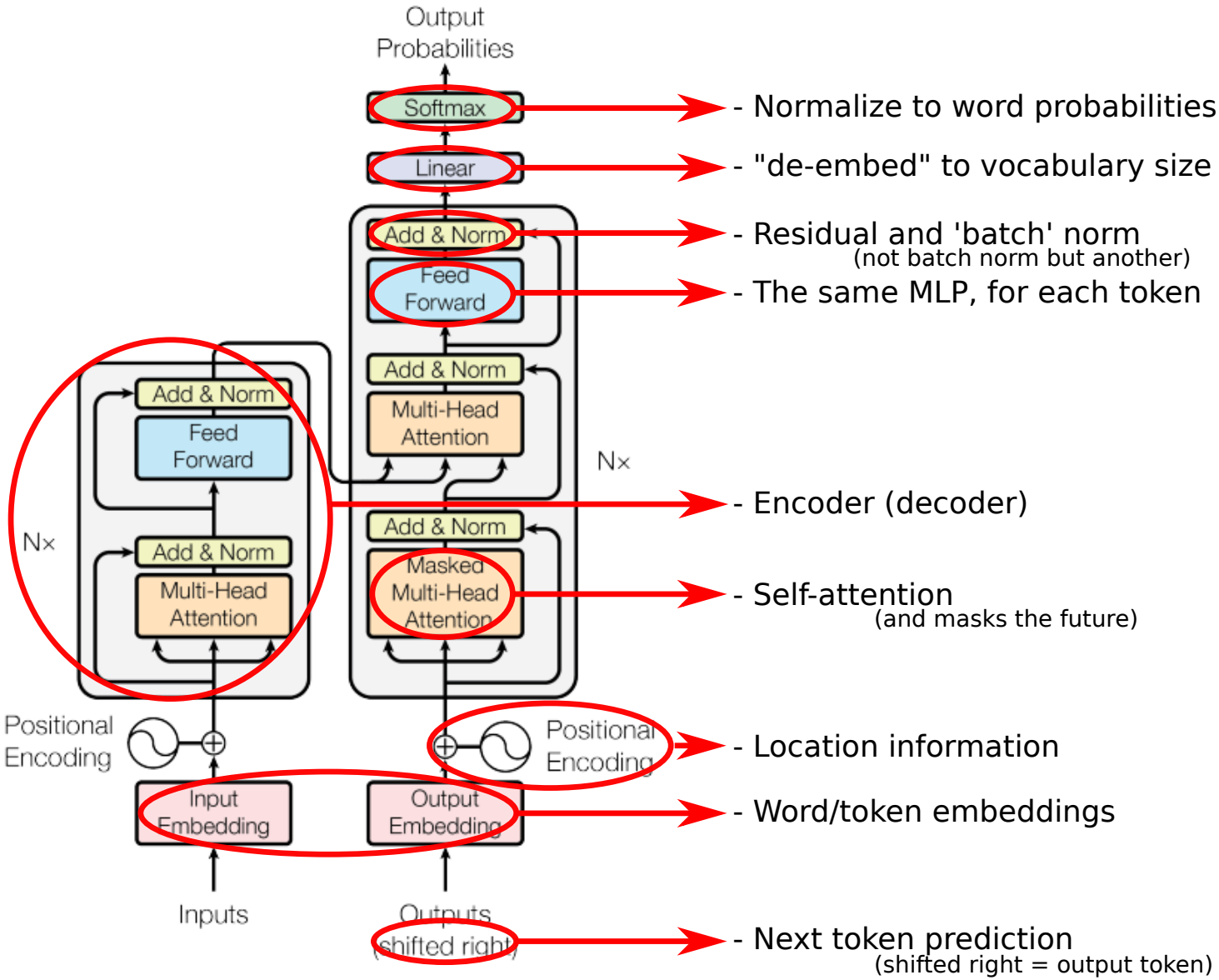
Attention is all you need: full *Transformer* architecture

Paper: `https://research.google/pubs/attention-is-all-you-need/`



Attention is all you need: full *Transformer* architecture

Paper: `https://research.google/pubs/attention-is-all-you-need/`



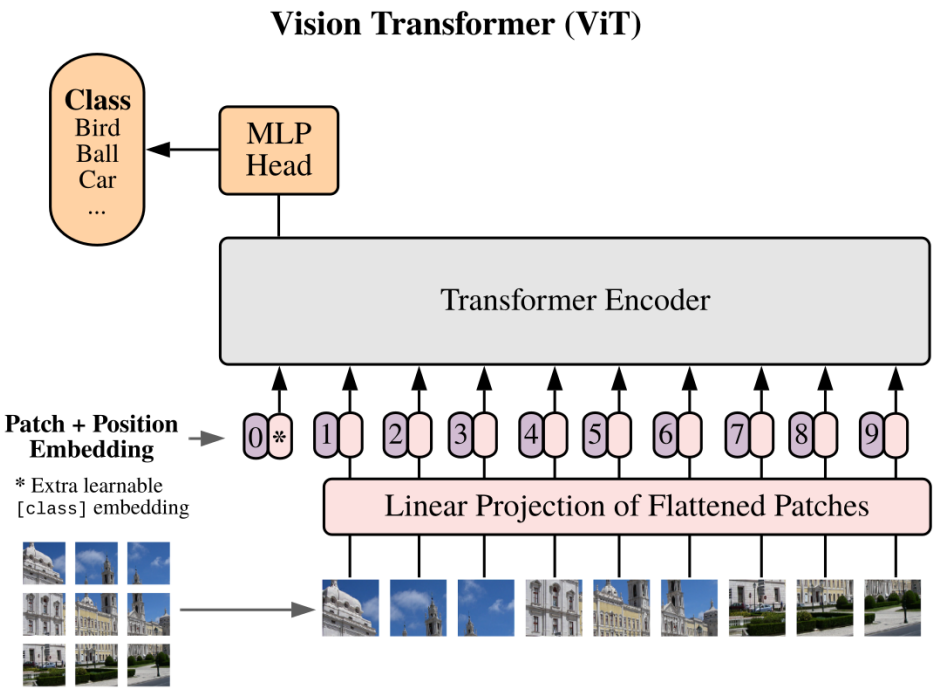
Questions?

Other types of data

Q: How to apply the transformer on images?

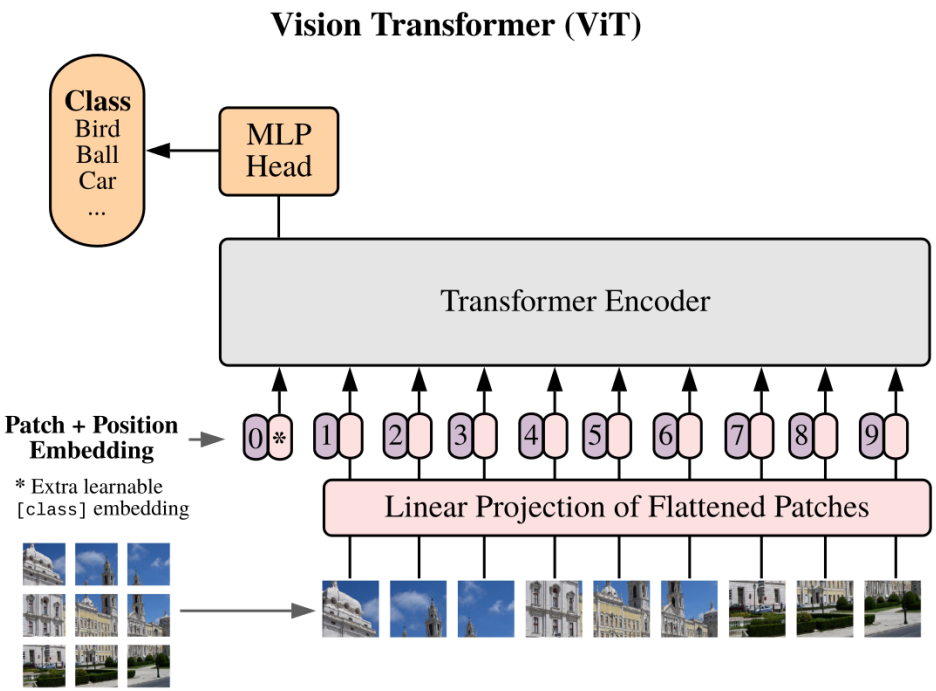
Other types of data

Q: How to apply the transformer on images? A:



Other types of data

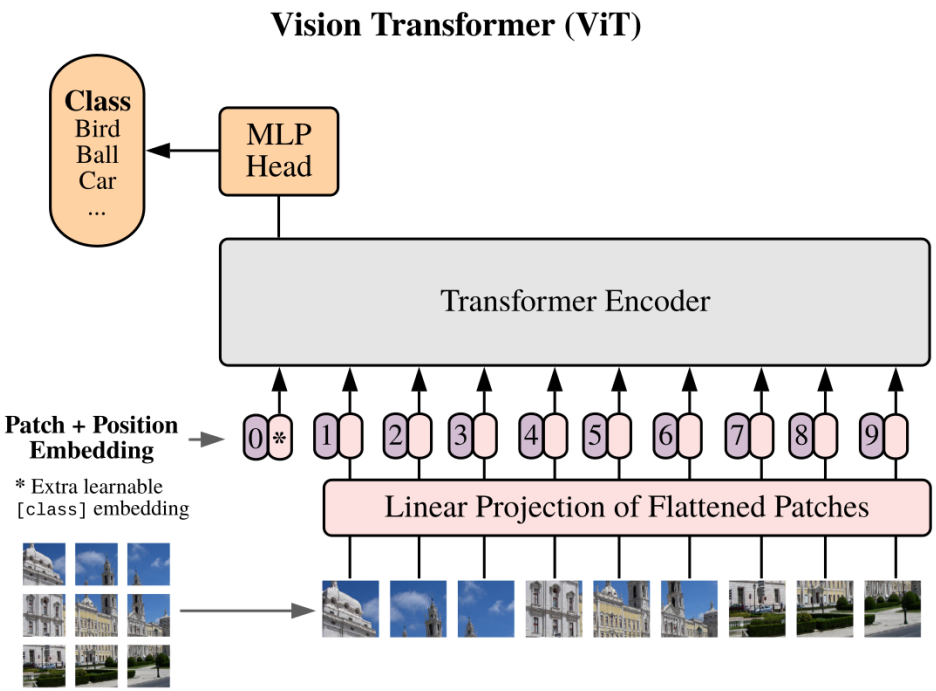
Q: How to apply the transformer on images? A:



Q: What is similar in ViT and CNN?

Other types of data

Q: How to apply the transformer on images? A:

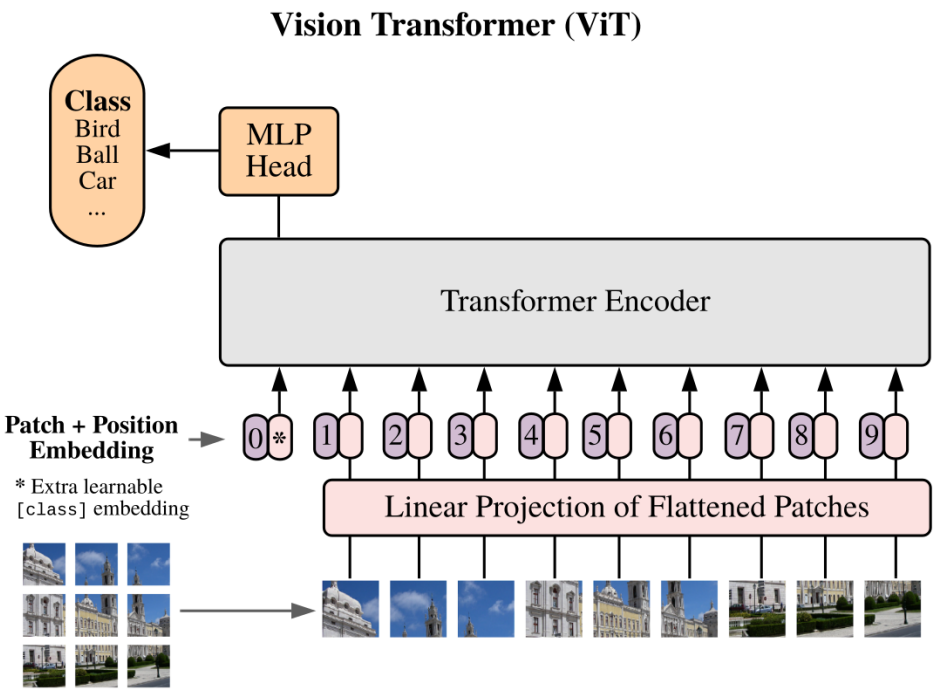


Q: What is similar in ViT and CNN?

· Convolutions at each layer (shared MLP)

Other types of data

Q: How to apply the transformer on images? A:



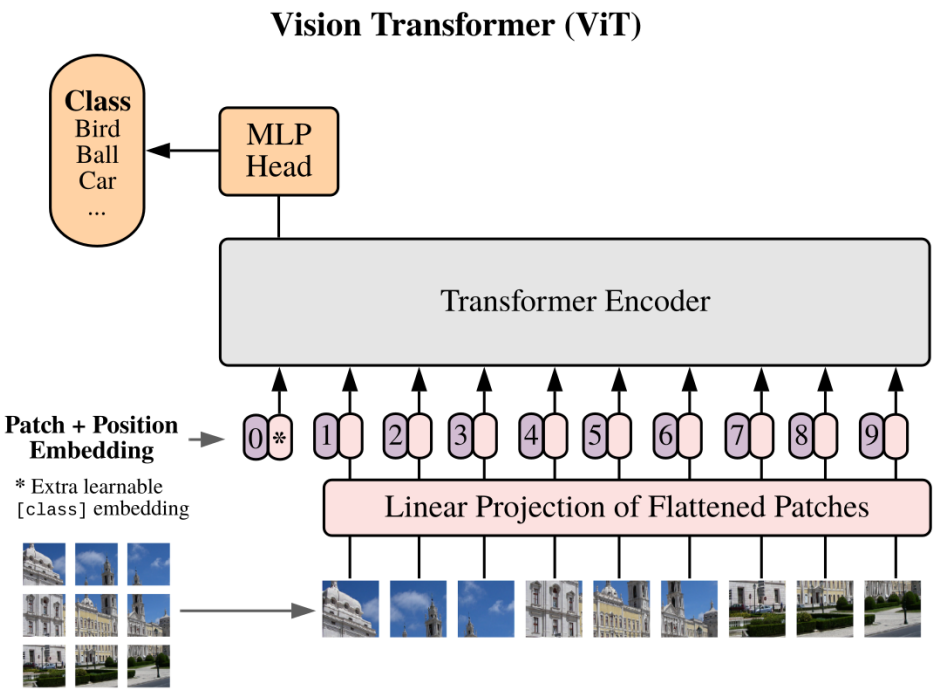
Q: What is similar in ViT and CNN?

· Convolutions at each layer (shared MLP)

Q: What is different in a ViT vs CNN?

Other types of data

Q: How to apply the transformer on images? A:



Q: What is similar in ViT and CNN?

· Convolutions at each layer (shared MLP)

Q: What is different in a ViT vs CNN?

· 2D Position embedding

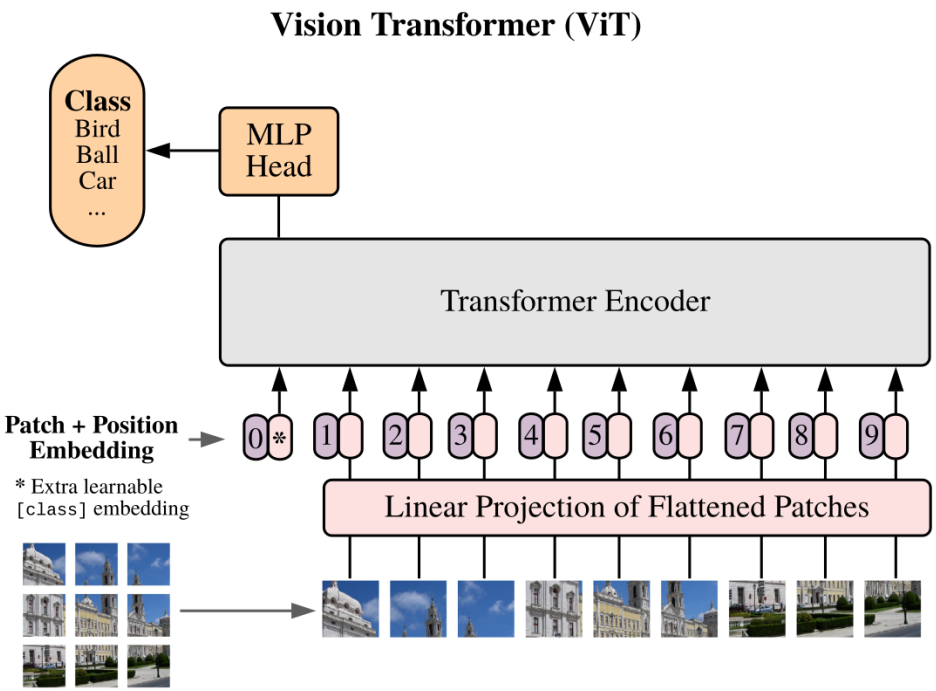
· SA (Self-Attention) at each layer

· ViT has global receptive field after SA

· SA is input dependent

Other types of data

Q: How to apply the transformer on images? A:



Q: What is similar in ViT and CNN?

· Convolutions at each layer (shared MLP)

Q: What is different in a ViT vs CNN?

· 2D Position embedding

· SA (Self-Attention) at each layer

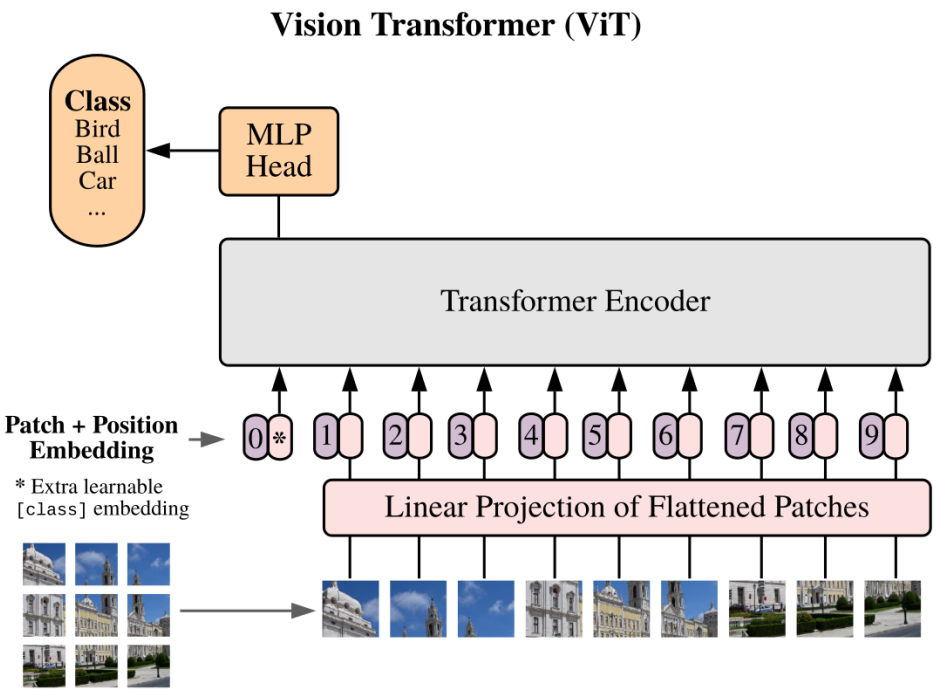
· ViT has global receptive field after SA

· SA is input dependent

Q: How would you apply it on video?

Other types of data

Q: How to apply the transformer on images? A:



Q: What is similar in ViT and CNN?

· Convolutions at each layer (shared MLP)

Q: What is different in a ViT vs CNN?

· 2D Position embedding

· SA (Self-Attention) at each layer

· ViT has global receptive field after SA

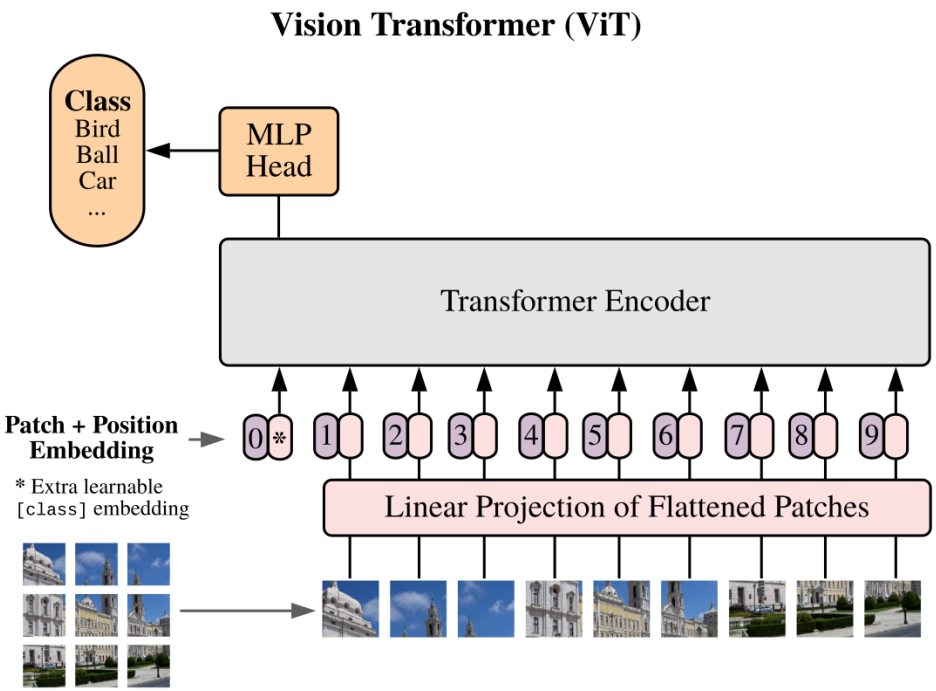
· SA is input dependent

Q: How would you apply it on video?

A: Make 3D patches of video.

Other types of data

Q: How to apply the transformer on images? A:



Q: What is similar in ViT and CNN?

· Convolutions at each layer (shared MLP)

Q: What is different in a ViT vs CNN?

· 2D Position embedding

· SA (Self-Attention) at each layer

· ViT has global receptive field after SA

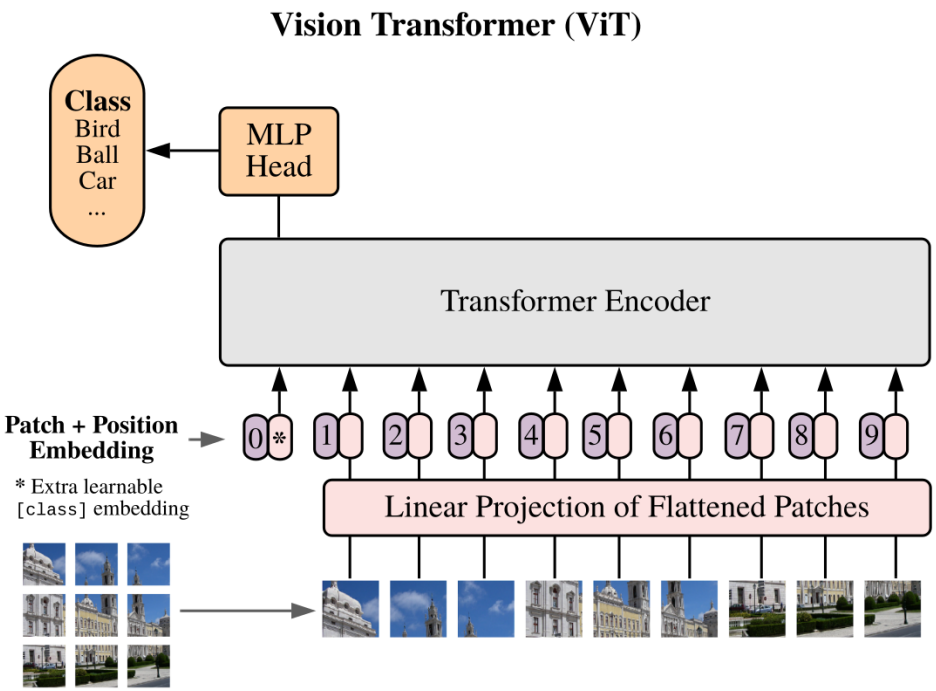
· SA is input dependent

Q: How would you apply it on video?

A: Make 3D patches of video.

Other types of data

Q: How to apply the transformer on images? A:



Q: What is similar in ViT and CNN?

· Convolutions at each layer (shared MLP)

Q: What is different in a ViT vs CNN?

· 2D Position embedding

· SA (Self-Attention) at each layer

· ViT has global receptive field after SA

· SA is input dependent

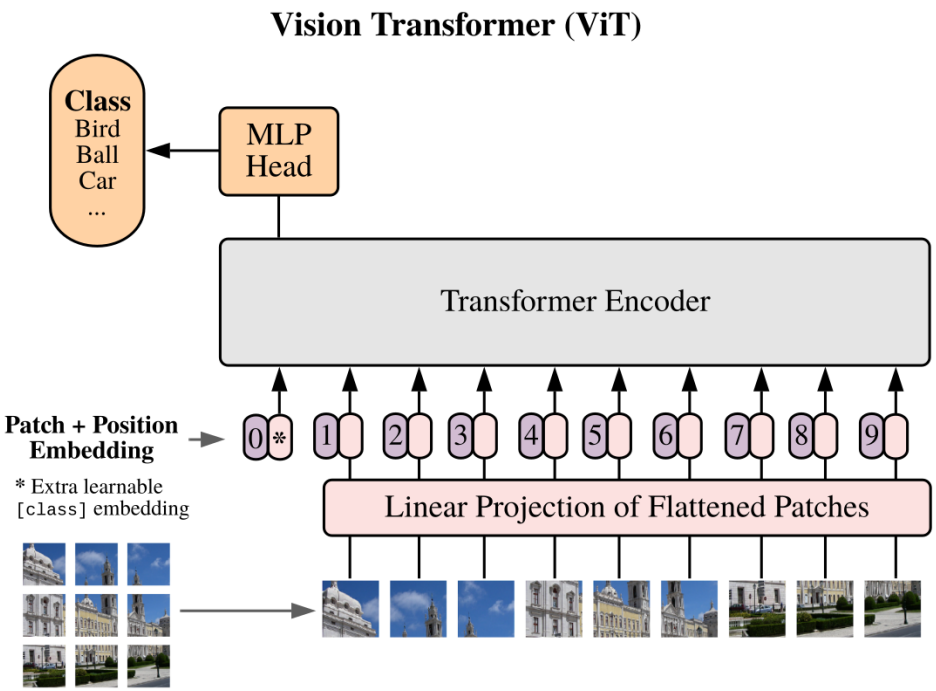
Q: How would you apply it on video?

A: Make 3D patches of video.

Q: How to apply it on other signals?

Other types of data

Q: How to apply the transformer on images? A:



Q: What is similar in ViT and CNN?
· Convolutions at each layer (shared MLP)

Q: What is different in a ViT vs CNN?
· 2D Position embedding
· SA (Self-Attention) at each layer
· ViT has global receptive field after SA
· SA is input dependent

Q: How would you apply it on video?
A: Make 3D patches of video.

Q: How to apply it on other signals?
A: As long as it can be turned into patches/tokens.

Questions?

Pre-training

Q: Why can transformers be trained on large texts?

Pre-training

Q: Why can transformers be trained on large texts?
A: Next word prediction: let the model learn to reconstruct the input.

Pre-training

Q: Why can transformers be trained on large texts?
A: Next word prediction: let the model learn to reconstruct the input.

No additional human ground-truth labels needed: Self-supervised.

Pre-training

Q: Why can transformers be trained on large texts?
A: Next word prediction: let the model learn to reconstruct the input.

No additional human ground-truth labels needed: Self-supervised.

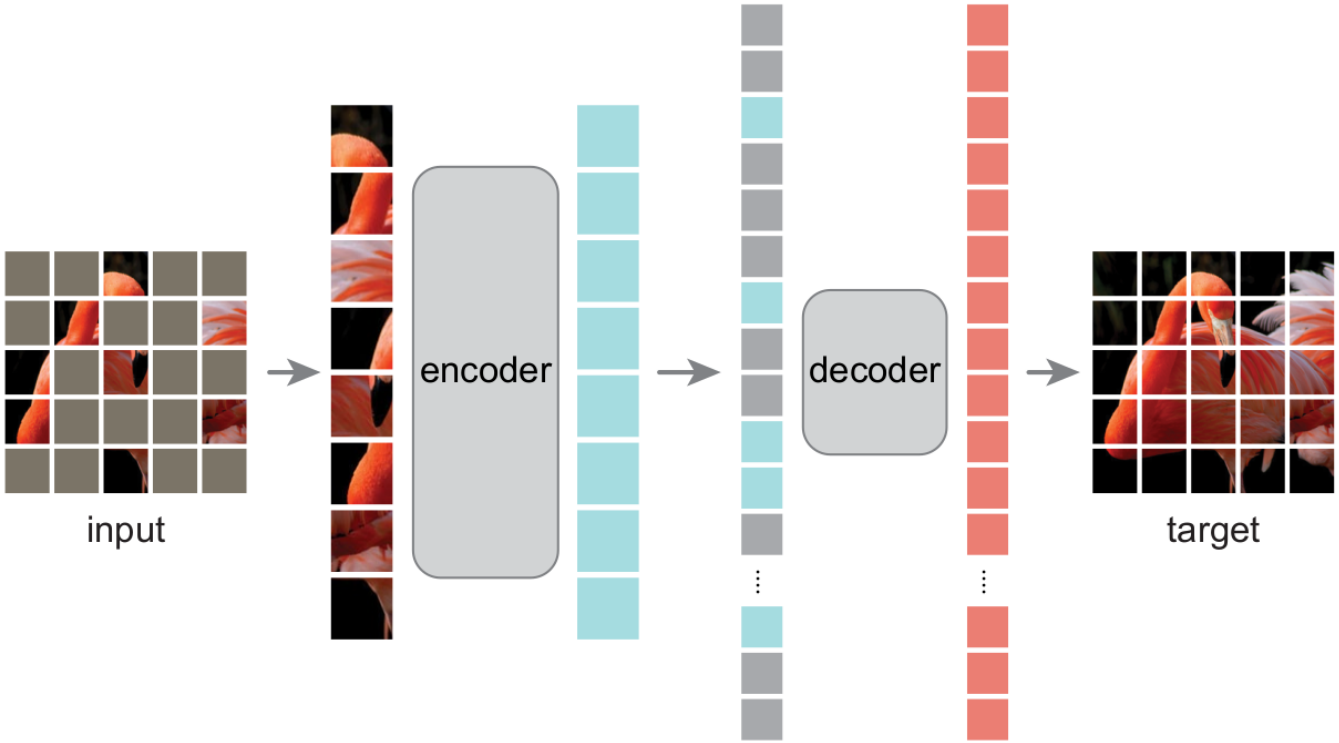
Q: How would you do that for images?

Pre-training

Q: Why can transformers be trained on large texts?
A: Next word prediction: let the model learn to reconstruct the input.

No additional human ground-truth labels needed: Self-supervised.

Q: How would you do that for images? A:



Masked auto-encoder `https://arxiv.org/abs/2111.06377`

Pre-training of images and text (captions)

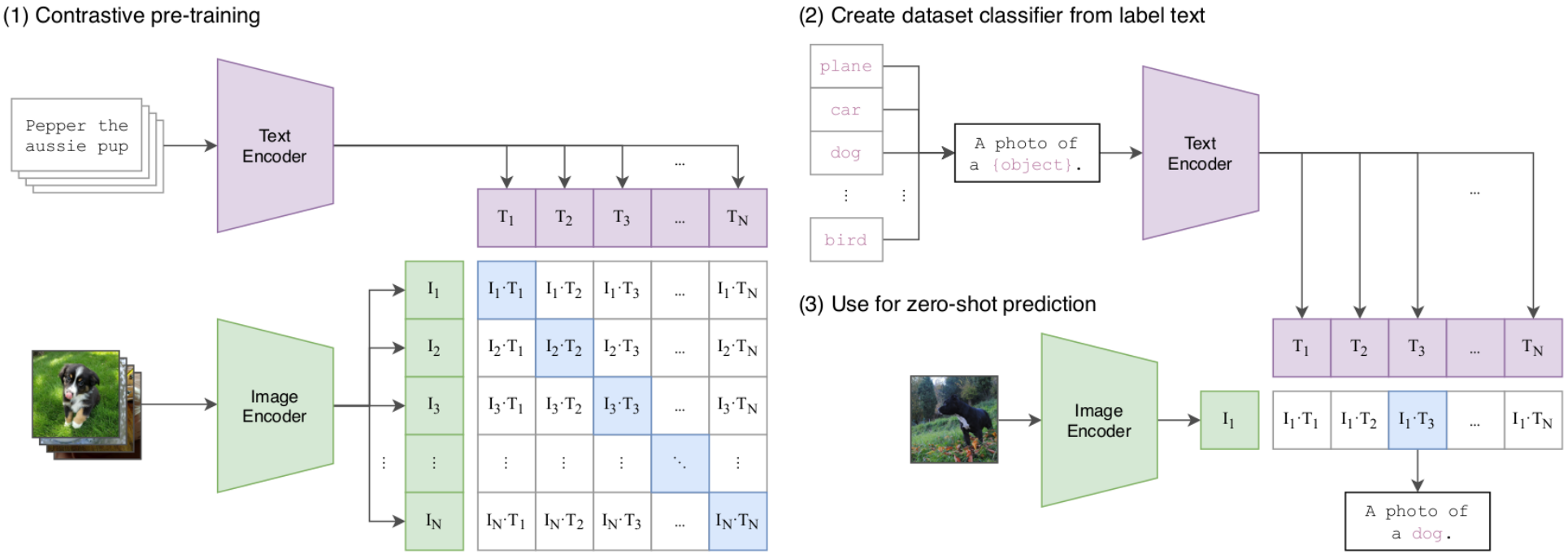
Q: Given N images, with their N textual captions; how to pre-train?

Pre-training of images and text (captions)

Q: Given N images, with their N textual captions; how to pre-train?
A: Maximize the cosine similarity of the N correct pairs;
A: Minimize the cosine similarity of the $N^2 - N$ incorrect pairs;

Pre-training of images and text (captions)

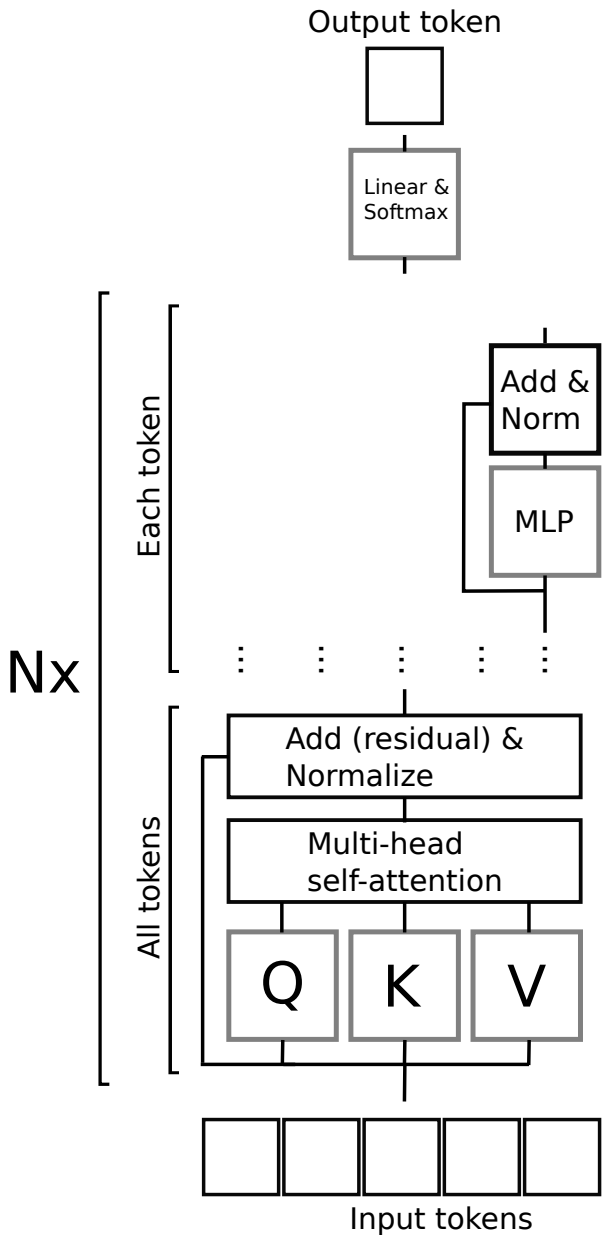
Q: Given N images, with their N textual captions; how to pre-train?
A: Maximize the cosine similarity of the N correct pairs;
A: Minimize the cosine similarity of the $N^2 - N$ incorrect pairs;



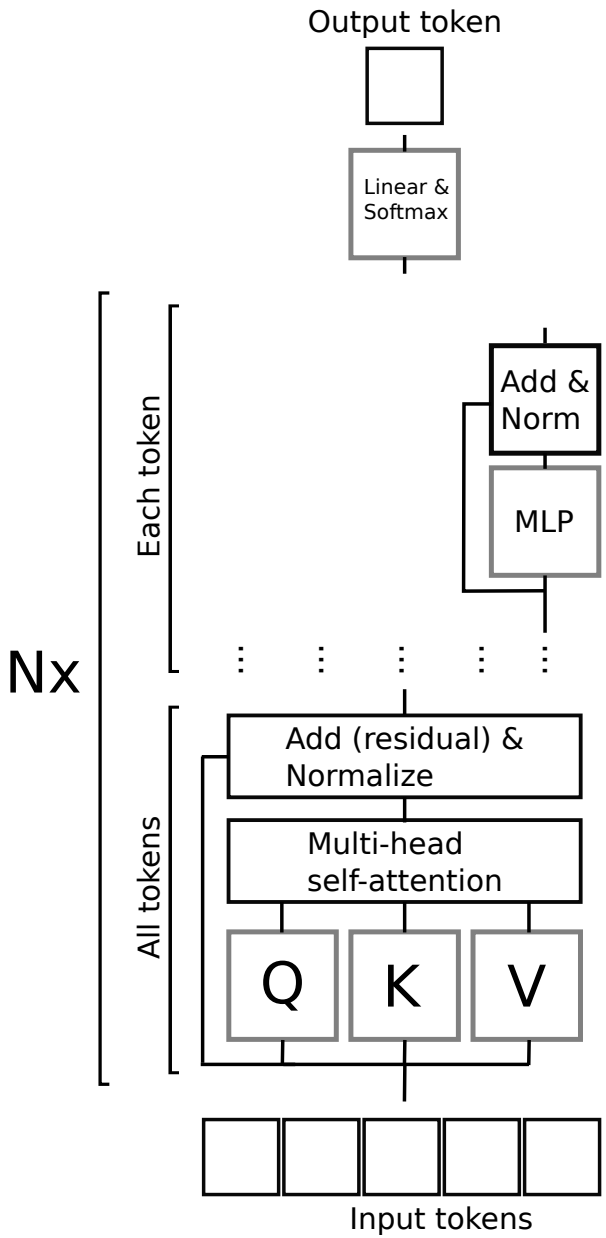
CLIP: https://arxiv.org/abs/2103.00020

Questions?

Parameter-Efficient Fine-Tuning (PEFT) https://arxiv.org/abs/2403.14608

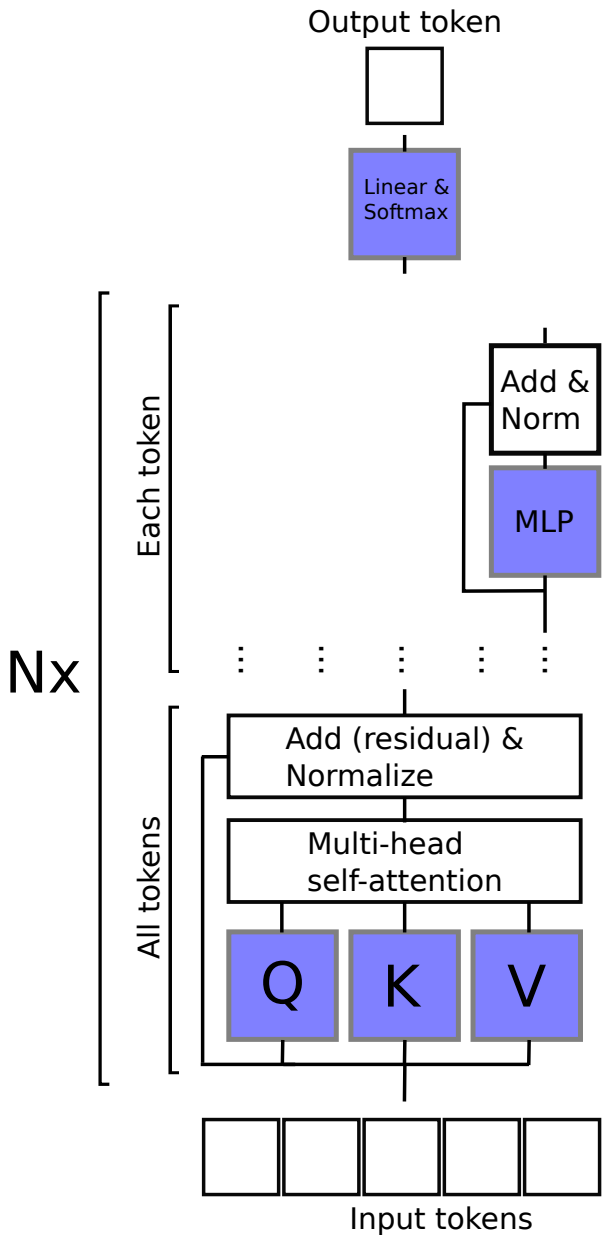


Parameter-Efficient Fine-Tuning (PEFT) https://arxiv.org/abs/2403.14608



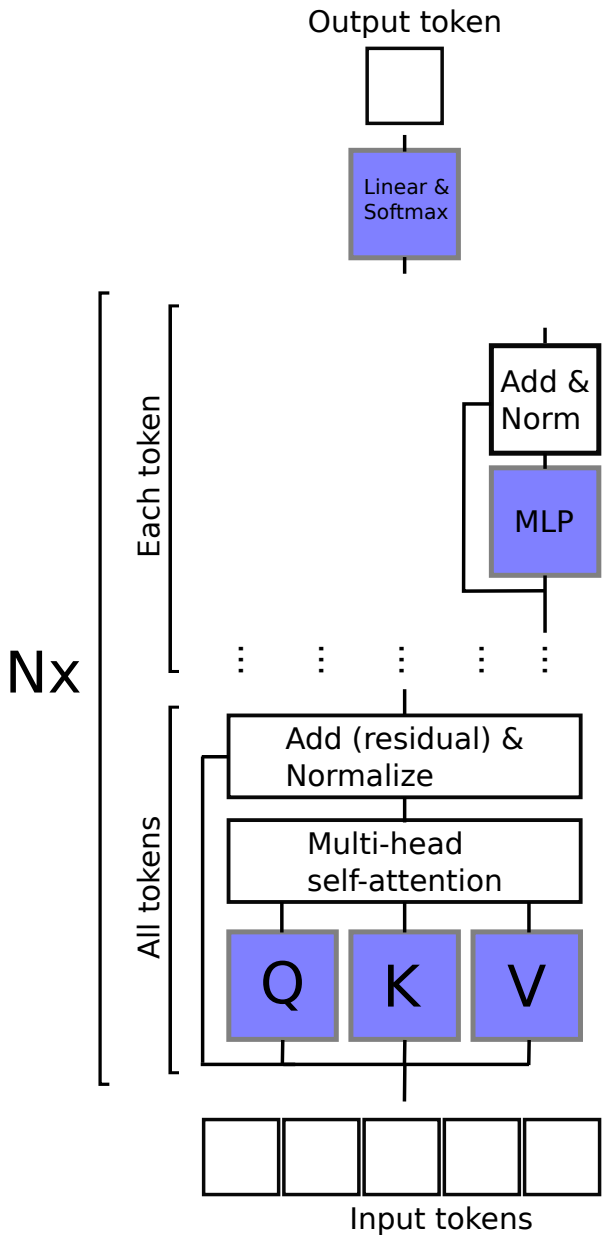
Q: Where are the learnable parameters?

Parameter-Efficient Fine-Tuning (PEFT) https://arxiv.org/abs/2403.14608



Q: Where are the learnable parameters?

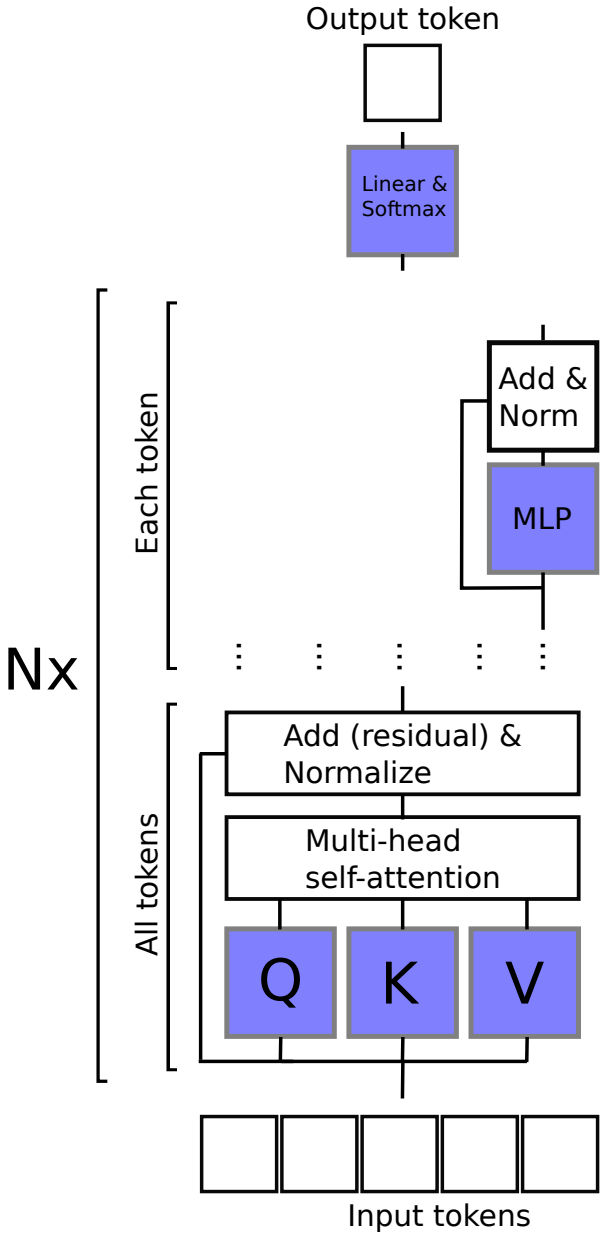
Parameter-Efficient Fine-Tuning (PEFT) https://arxiv.org/abs/2403.14608



Q: Where are the learnable parameters?

Q: How to fine-tune on new data?

Parameter-Efficient Fine-Tuning (PEFT) https://arxiv.org/abs/2403.14608

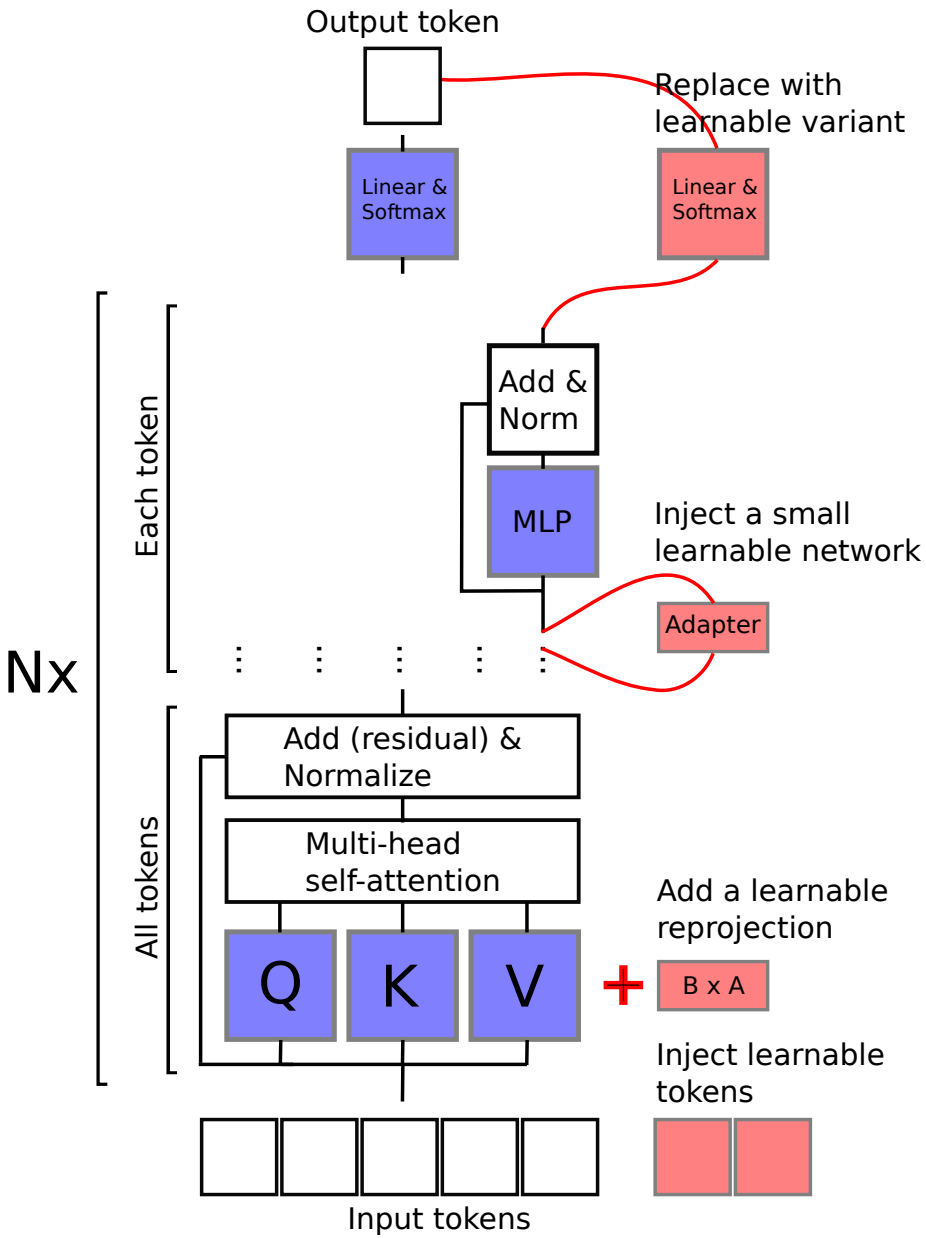


Q: Where are the learnable parameters?

Q: How to fine-tune on new data?

Parameters: Frozen ; Learnable (few)

Parameter-Efficient Fine-Tuning (PEFT) https://arxiv.org/abs/2403.14608

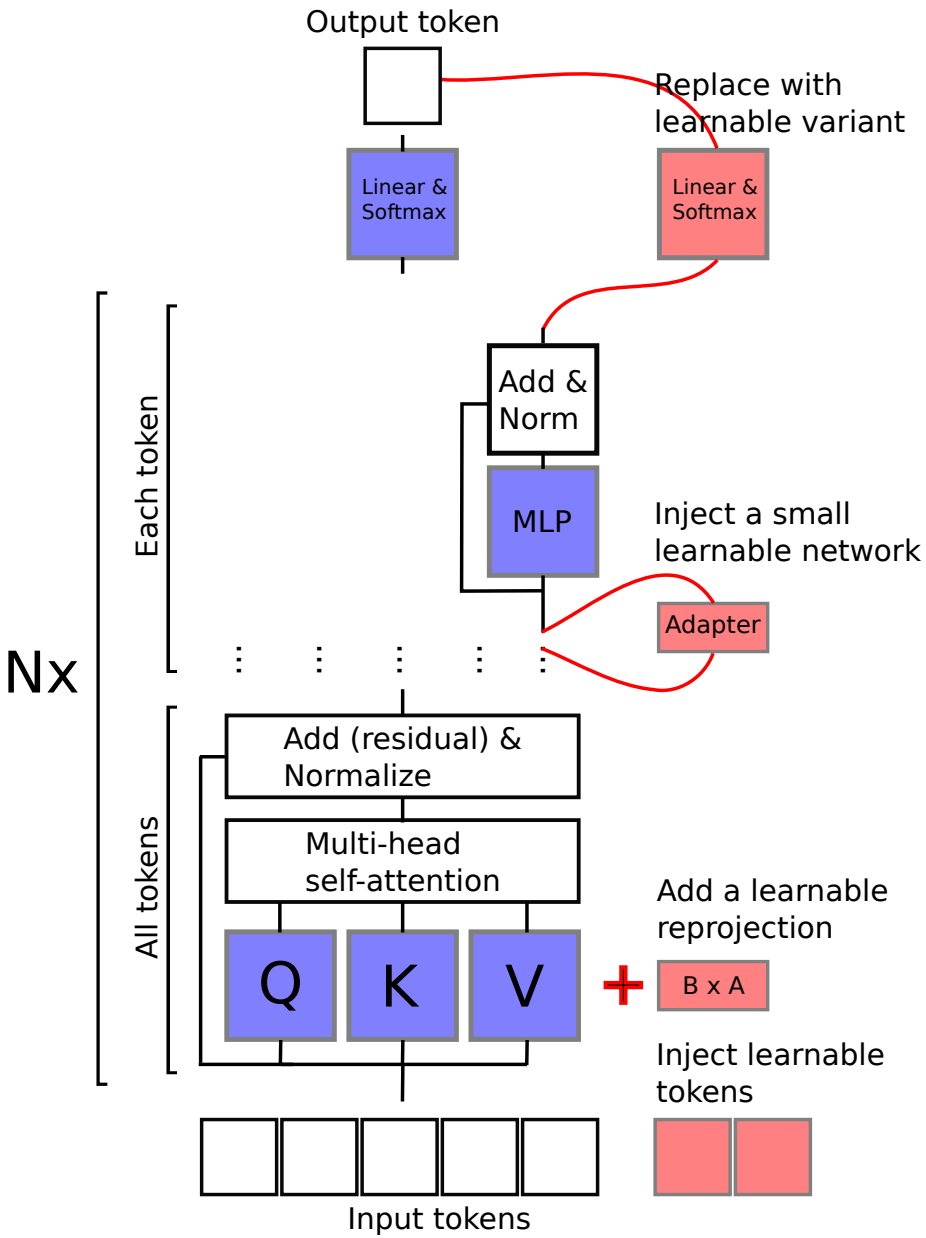


Q: Where are the learnable parameters?

Q: How to fine-tune on new data?

Parameters: Frozen ; Learnable (few)

Parameter-Efficient Fine-Tuning (PEFT) https://arxiv.org/abs/2403.14608



Q: Where are the learnable parameters?

Q: How to fine-tune on new data?

Parameters: Frozen ; Learnable (few)

Selective (replace):

- Replace and re-learn a subset of existing parameters (eg last layer).

Additive (inject):

- Inject new learnable parameters; eg: $\text{Adapter}(\mathbf{x}) = \mathbf{W}_{\text{up}}\sigma(\mathbf{W}_{\text{down}}\mathbf{x}) + \mathbf{x}$
- Inject a fixed amount of learnable tokens (prompt tuning)

Reparameterized (reproject):

- Learn a low-rank reprojection of weight matrices; eg: LoRa: $\mathbf{h} = \mathbf{W}_0\mathbf{x} + \Delta\mathbf{W}\mathbf{x} = \mathbf{W}_0\mathbf{x} + \mathbf{B}\mathbf{A}\mathbf{x}$ $\mathbf{W}_0 \in \mathbb{R}^{d\times k}$, $\mathbf{A} \in \mathbb{R}^{r\times k}$, $\mathbf{B} \in \mathbb{R}^{d\times r}$ where $\text{rank}(r) << \min(d, k)$

Questions?

Summary

- Foundation models: large, broadly pre-trained, models
- Transformer architecture: homogenization of models
- Other modalities, images (ViT), video, etc.
- Pre-training; eg on images (mae); images and captions (clip)
- PEFT: Substitute (train a small sub-set of parameters), inject (Adapter), add a weight reprojection (LoRa),