

Transformers

История

- (19xx) классические методы

История

- (19xx) классические методы
- (2012) CNNs

История

- (19xx) классические методы
- (2012) CNNs
- можно ли лучше?..

Attention is all you need

Революция в NLP: Transformer

[Google] A.Vaswani et al., 2017 “Attention Is All You Need”

<https://arxiv.org/abs/1706.03762>

Attention is all you need

Адаптировали под CV: ViT (Vision Transformer)

[Google] A.Dosovitskiy et al., 2020 “An Image is Worth 16x16 Words”

<https://arxiv.org/abs/2010.11929>

Трансформеры

NLP

Проблемы RNNs:

- зависимости между далекими словами
The animal didn't cross the street, because it was too tired
- исчезающие и взрывающиеся градиенты
- невозможность параллелизации вычислений

Attention

Решение - механизм внимания (attention)

Attention vs Базы данных

Сравнение с базой данных:

- в БД хранятся пары $\langle \text{key}, \text{value} \rangle$
- чтобы получить значение из БД (value retrieval):
 - составляете query
 - query сопоставляется с разными ключами
 - находится key: лучший $\langle \text{query}, \text{key} \rangle$ мэтч (по функции сходства *similarity*)
 - выдается значение value по этому ключу key

$$v^* = D[\arg \max_k \textit{similarity}(k, q)]$$

Attention vs Базы данных

Attention = soft value retrieval:

$$v^* = \sum_k \textit{similarity}(k, q) \cdot v$$

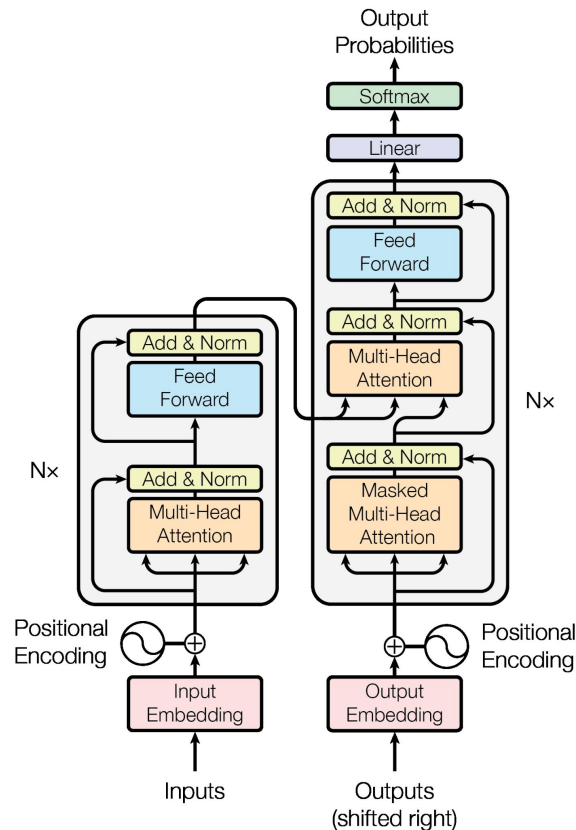
Функции сходства

- скалярное произведение $q^T k$
- масштабированное скалярное произведение $q^T k / \sqrt{d}$
- обобщенное скалярное произведение $q^T W k$
- аддитивное сходство $w_q^T q + w_k^T k_i$
- софтмакс от ... $\text{softmax}(q^T \mathbf{k})$

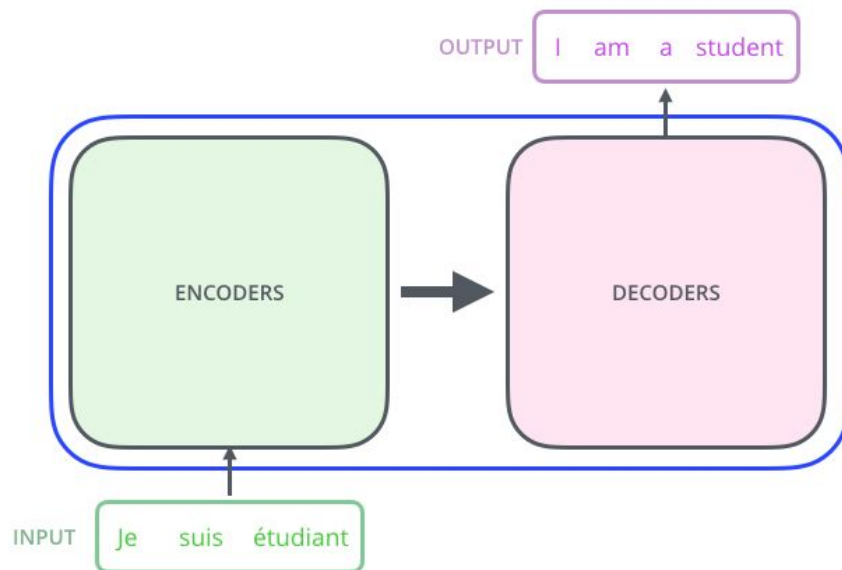
Архитектура Трансформера

Компоненты:

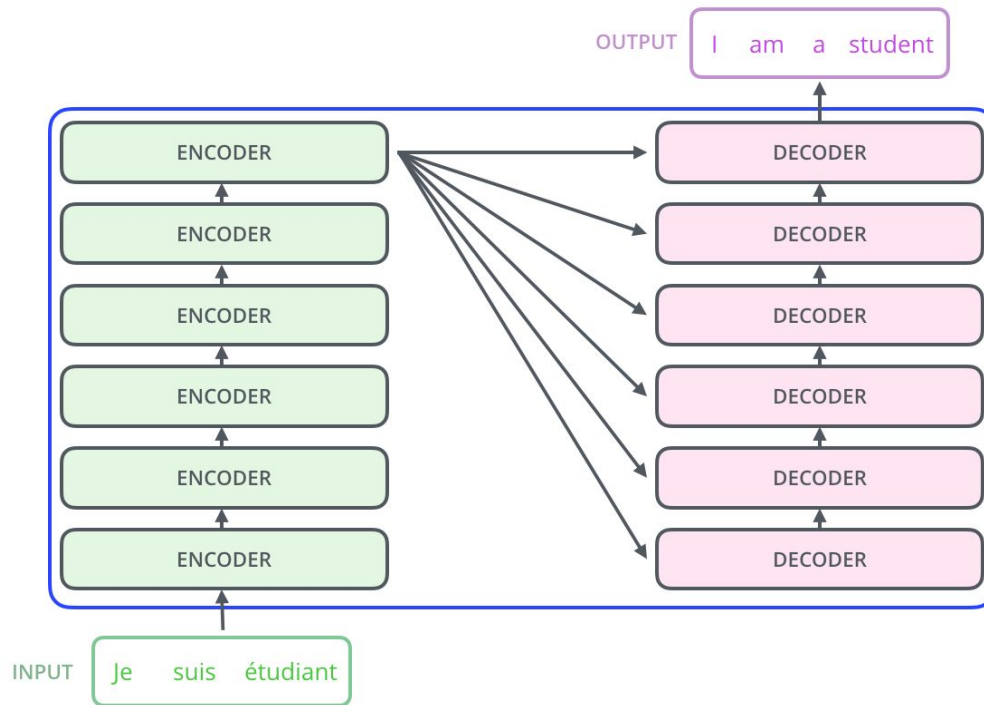
- энкодер и декодер
- эмбединги
- позиционные эмбединги
- skip connections
- multi-head self-attention
- encoder-decoder attention
- нормализация
- полносвязные слои



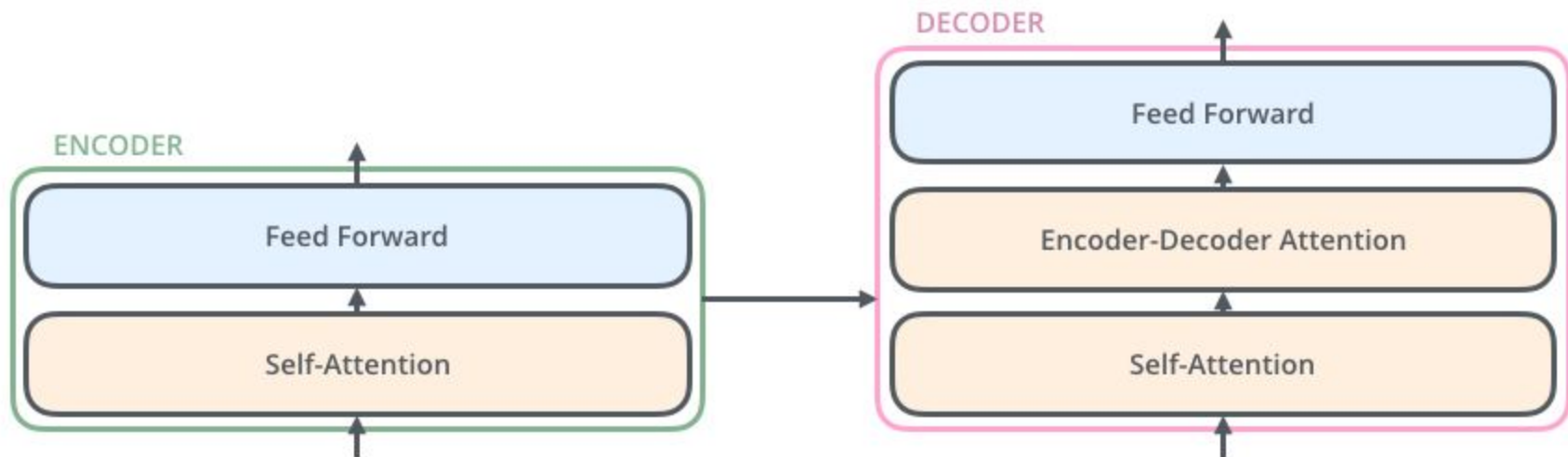
Энкодер и декодер



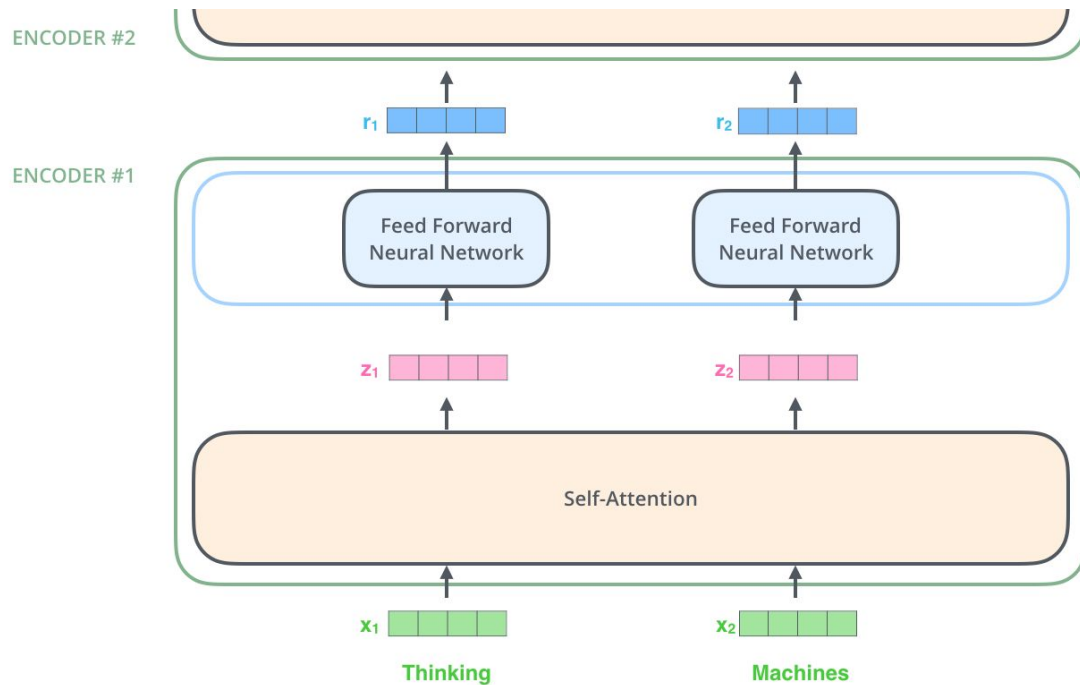
Энкодер и декодер



Энкодер и декодер



Энкодер и декодер



Self-Attention

$$\begin{matrix} \text{X} \\ \begin{array}{|c|c|c|c|} \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array} \end{matrix} \times \begin{matrix} W^Q \\ \begin{array}{|c|c|c|c|c|} \hline & & & & \\ \hline & & & & \\ \hline & & & & \\ \hline & & & & \\ \hline \end{array} \end{matrix} = \begin{matrix} Q \\ \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \end{matrix}$$

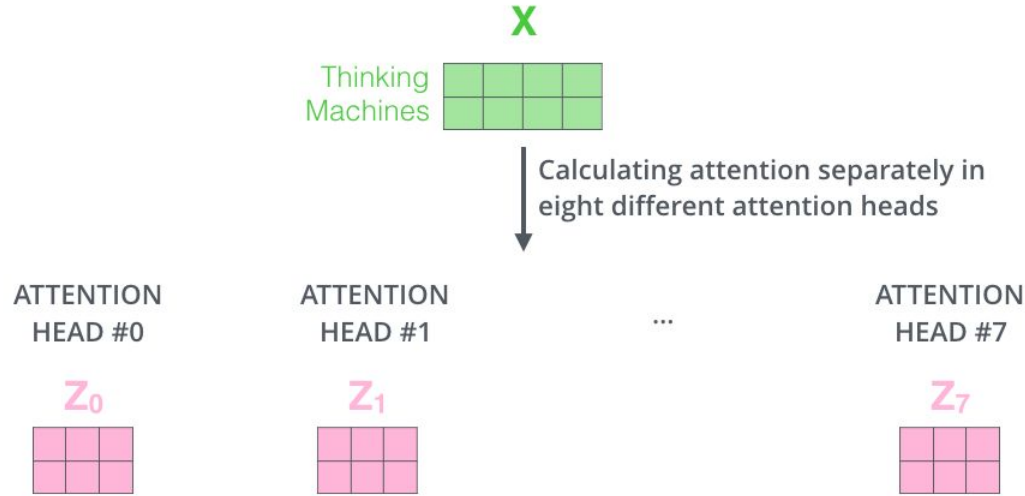
$$\begin{matrix} \text{X} \\ \begin{array}{|c|c|c|c|} \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array} \end{matrix} \times \begin{matrix} W^K \\ \begin{array}{|c|c|c|c|c|} \hline & & & & \\ \hline & & & & \\ \hline & & & & \\ \hline & & & & \\ \hline \end{array} \end{matrix} = \begin{matrix} K \\ \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \end{matrix}$$

$$\begin{matrix} \text{X} \\ \begin{array}{|c|c|c|c|} \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array} \end{matrix} \times \begin{matrix} W^V \\ \begin{array}{|c|c|c|c|c|} \hline & & & & \\ \hline & & & & \\ \hline & & & & \\ \hline & & & & \\ \hline \end{array} \end{matrix} = \begin{matrix} V \\ \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \end{matrix}$$



$$\begin{aligned} & \text{softmax} \left(\frac{\begin{matrix} Q \\ \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \end{matrix} \times \begin{matrix} K^T \\ \begin{array}{|c|c|} \hline & \\ \hline & \\ \hline & \\ \hline \end{array} \end{matrix}}{\sqrt{d_k}} \right) \begin{matrix} V \\ \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \end{matrix} \\ & = \begin{matrix} Z \\ \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \end{matrix} \end{aligned}$$

Multi-Head Self-Attention



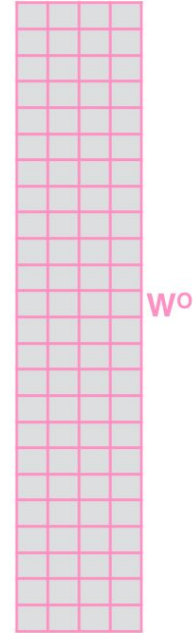
Multi-Head Self-Attention

1) Concatenate all the attention heads

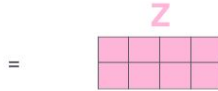


2) Multiply with a weight matrix W^O that was trained jointly with the model

\times

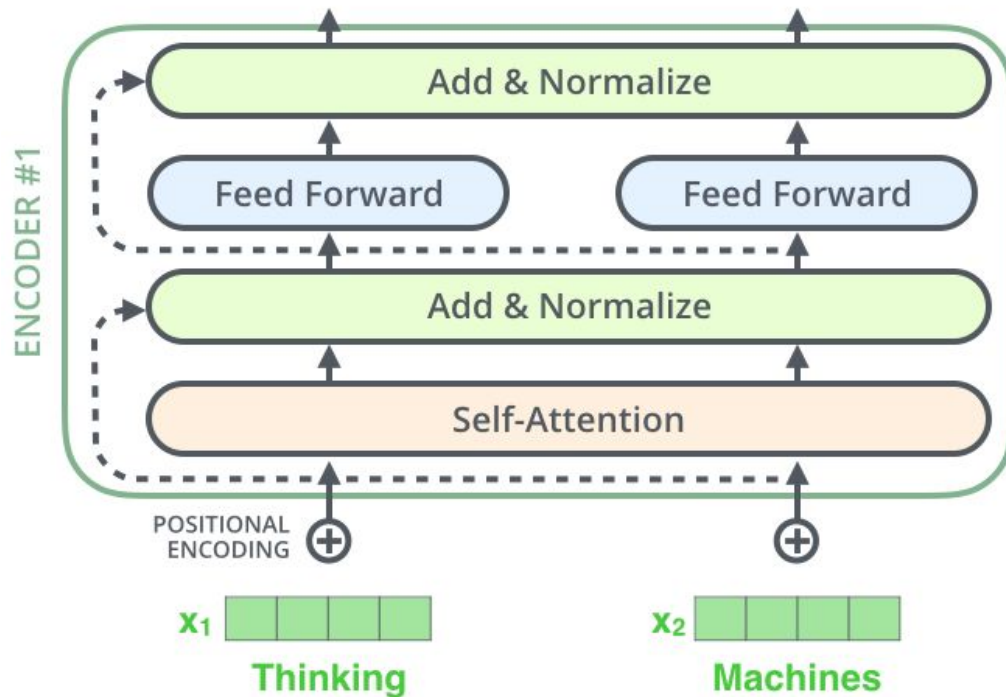


3) The result would be the Z matrix that captures information from all the attention heads. We can send this forward to the FFNN



Энкодер

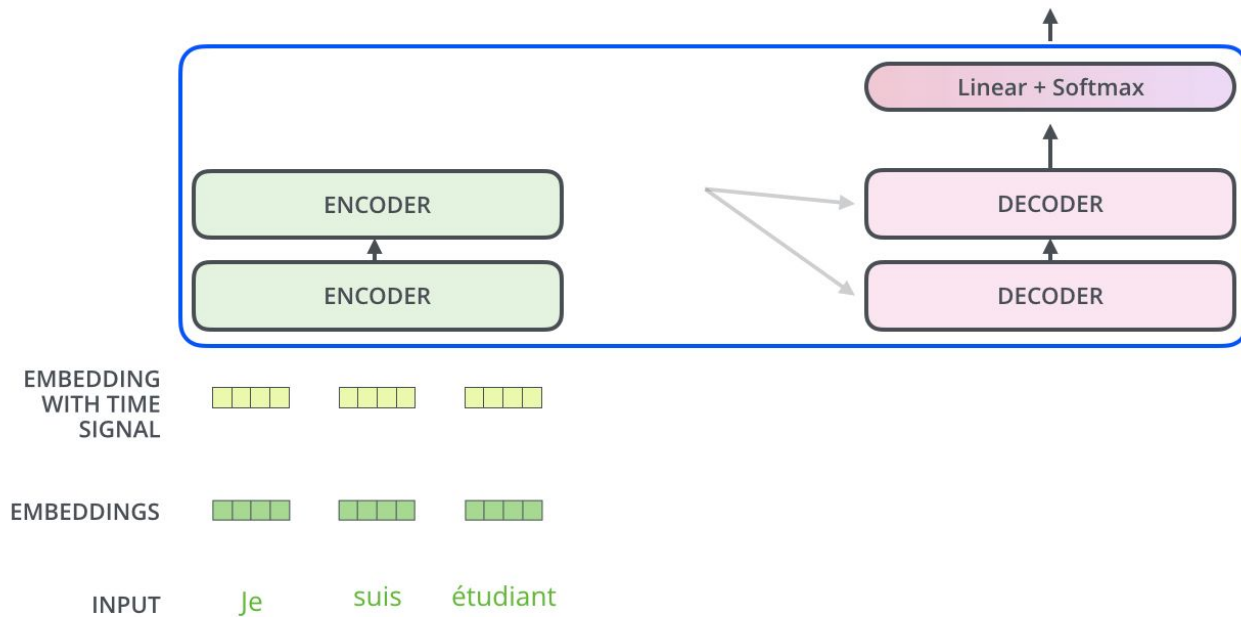
1. self-attention
2. fully connected NN
3. normalization
4. skip connections



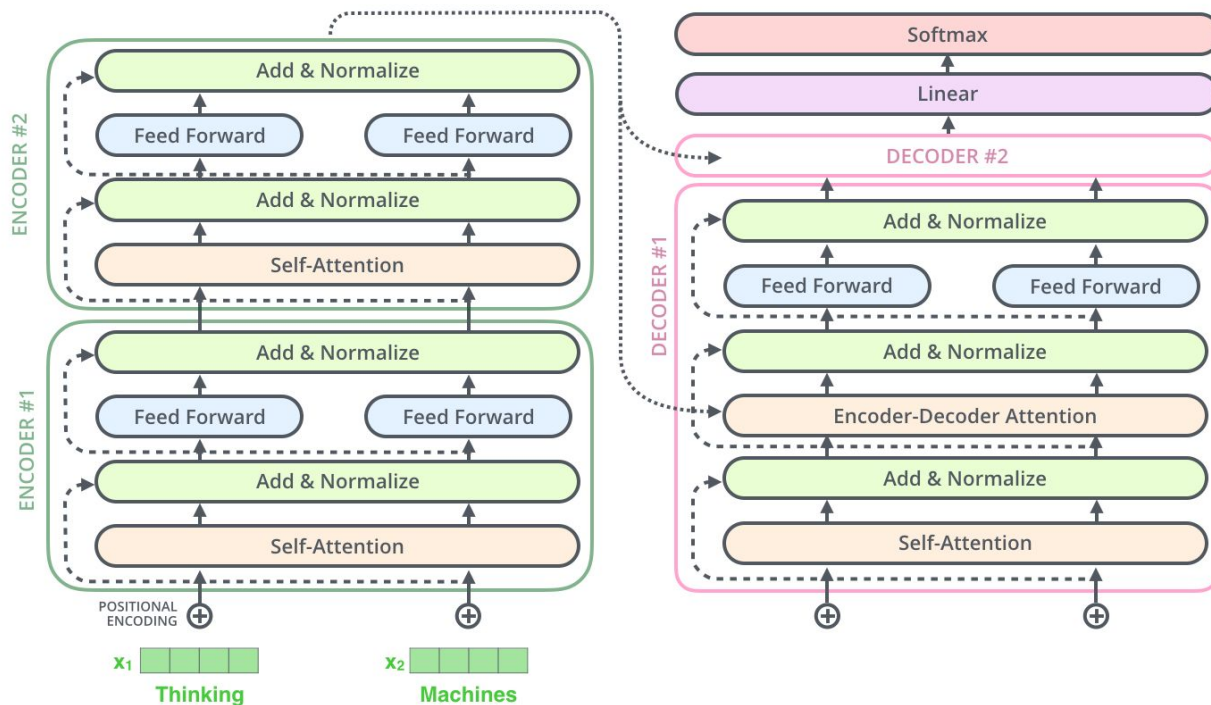
Декодер

Decoding time step: 1 2 3 4 5 6

OUTPUT



Encoder-Decoder Attention

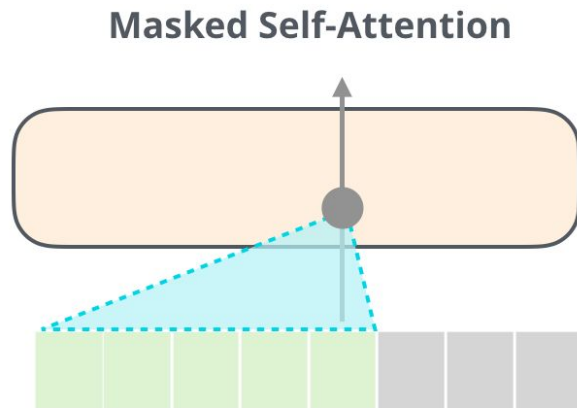
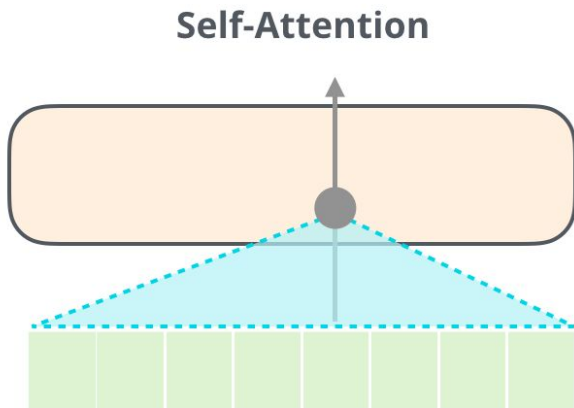


Masked Attention

Например, решаем задачу перевода с английского на русский

=> Не должны подглядывать вперед

=> Закрываем бинарной маской последующие слова (их эмбединги)



Начальные эмбединги

Подготовка входных данных:

- Embedding layer: превращает элементы (слова) в вектора
- Position Embedding Layer: учет позиции элементов

- синусоидальные:

для токена $i \in 1..L$ формируется эмбединг размера δ по формуле:

$$\text{PE}(i, \delta) = \begin{cases} \sin\left(\frac{i}{10000^{2\delta'/d}}\right) & \text{if } \delta = 2\delta' \\ \cos\left(\frac{i}{10000^{2\delta'/d}}\right) & \text{if } \delta = 2\delta' + 1 \end{cases}$$

- обучаемые

CNNs vs Transformers

Convolutions vs Attention

Свертки

- локальные зависимости
- используют 2d структуру
- линейная сложность вычислений

Внимание

- глобальные зависимости
- не используют 2d структуру
- квадратичная сложность вычислений

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(\log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

CNNs vs Transformers

CNNs

- используют индуктивный bias
=> требуется меньше данных
- меньше вычислительная сложность

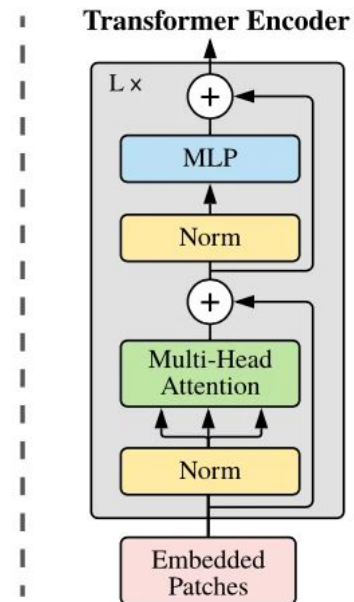
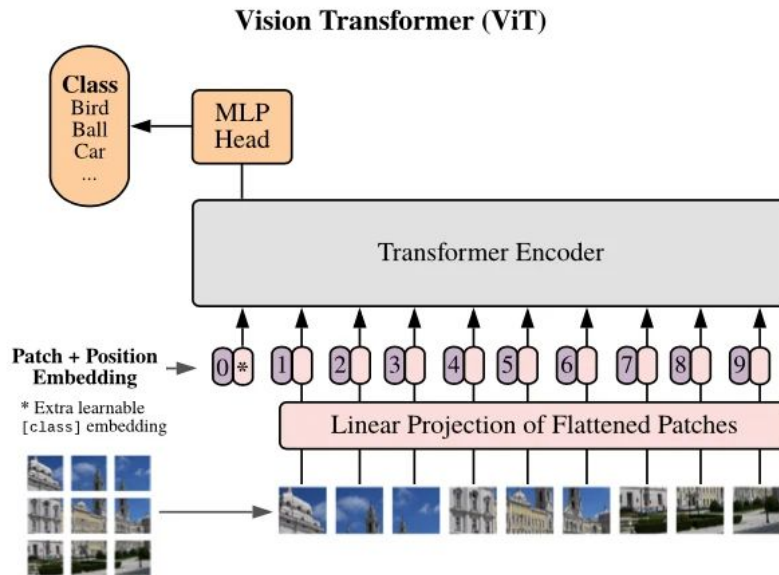
Transformers

- не используют индуктивный bias
=> требуется гораздо больше данных
- больше вычислительная сложность

ViT

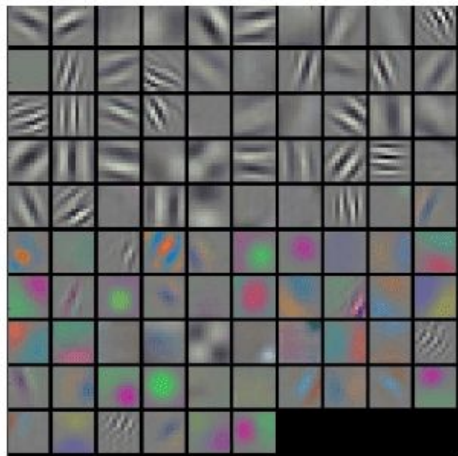
ViT

- токены - патчи изображения
- CLS token
- только энкодер



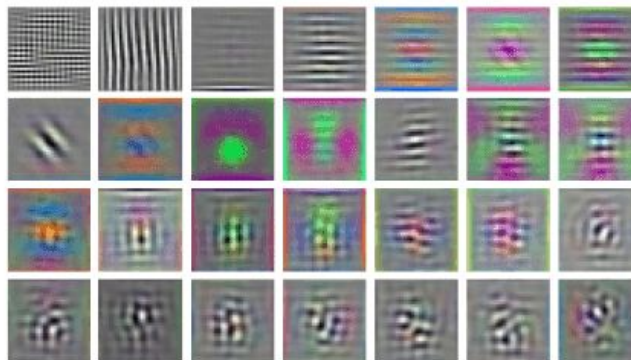
Визуализация фильтров

Alexnet 1st conv filters

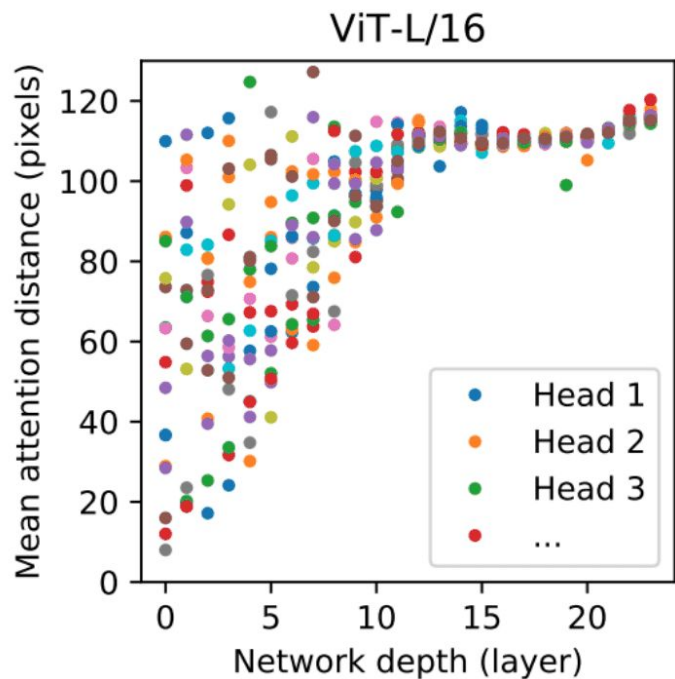


ViT 1st linear embedding filters

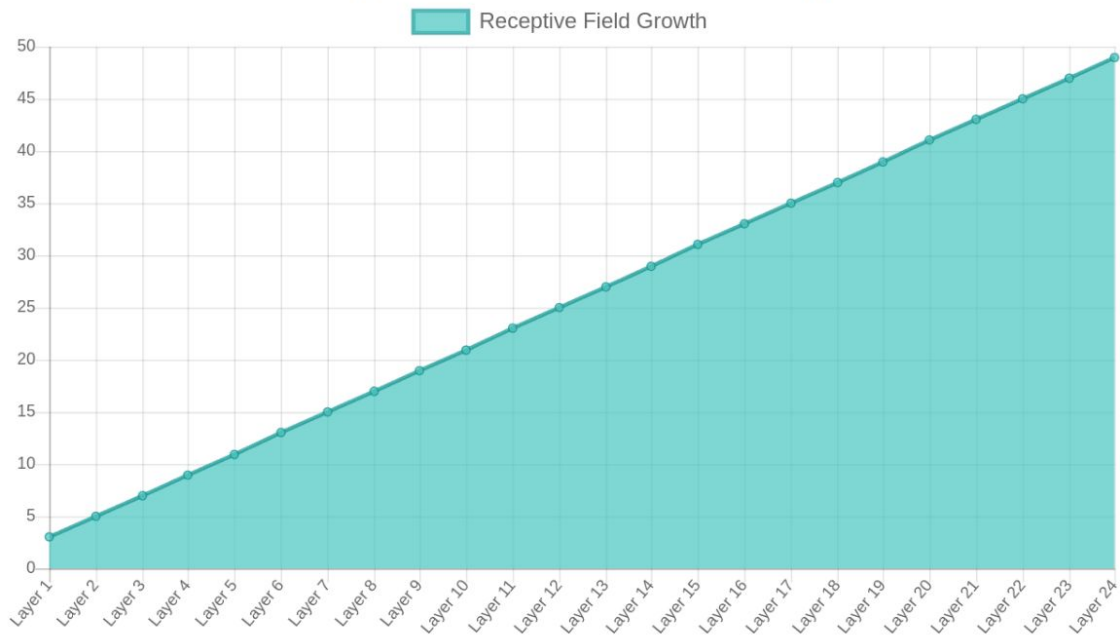
RGB embedding filters
(first 28 principal components)



Рецептивное поле



24 Conv layers with 3x3 kernel and single stride



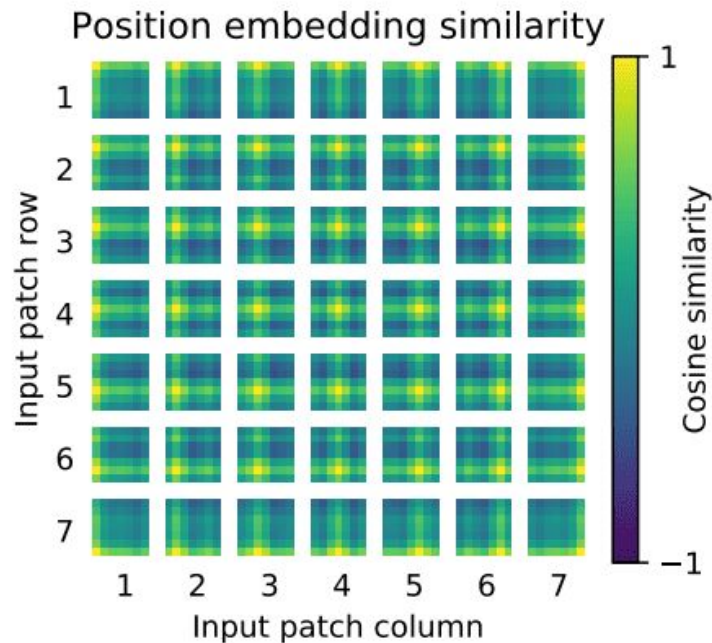
Positional Embedding

Авторы попробовали разные схемы:

1. обучаемые эмбеддинги
 - A. 1d абсолютное кодирование
 - B. 2d абсолютное кодирование (отдельно по O_x и O_y)
 - C. относительное кодирование
2. без позиционного кодирования

Итоги:

- $1 > 2$
- $1A \approx 1B \approx 1C$



Достижения трансформеров

Достижения трансформеров

Foundation models:

- self-supervised learning (SSL)
- masked pretrain
- мультимодальные модели

DINO

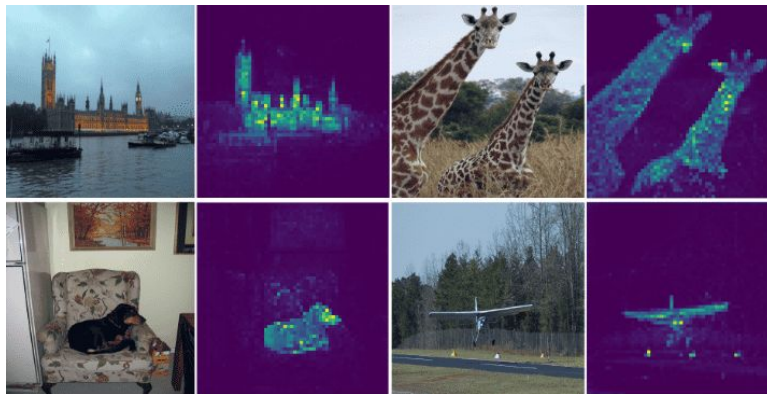
“Emerging Properties in Self-Supervised Vision Transformers” by FAIR, 2021 [[arxiv](#)]

- Трансформеры + SSL = успех

DINO

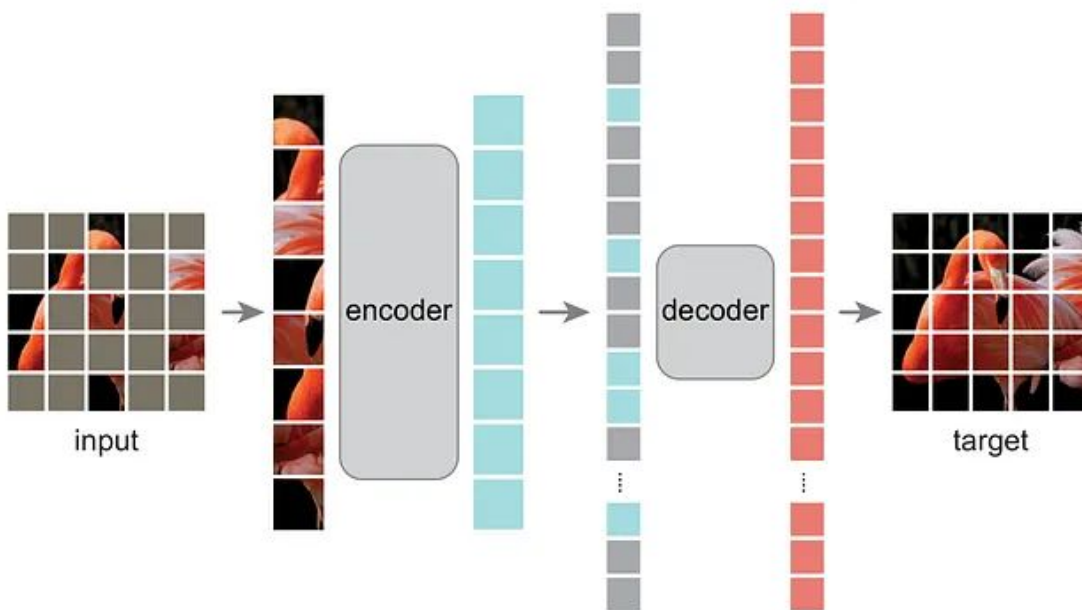
“Emerging Properties in Self-Supervised Vision Transformers” by FAIR, 2021 [[arxiv](#)]

- Трансформеры + SSL = успех
- Emergent capabilities: сегментация



MAE


“Masked Autoencoders Are Scalable Vision Learners” by FAIR, 2022 [[CVPR 2022](#)]



Мультимодальные модели

Vision трансформеры хорошо сочетаются с текстовыми:

- VisualBERT [2019]
- SimVLM [2021]
- CLIP [2021]
- ALIGN [2021]
- VLMo [2021]
- FLAVA [2021]
- Florence [2021]
- ...
- Flamingo [2022]
- BLIP [2022]
- OFA [2022]



This is a picture of Barack Obama. He is a former president of the United States.

What is he doing?

He is looking at the scale.

How many people are there in this picture?

There are at least 5 people in this picture.

Where is Obama's foot positioned?

Obama's foot is positioned on the right side of the scale.

Where was this picture taken?

It was taken in a school.

What happens as a result?

The scale shows a higher weight.

How many mirrors are there?

There are at least 2 mirrors in this picture.

Is the person on the scale aware of it?

I think he is not aware of it.

What is the person standing on?

The person is standing on a rug.

Do you think that's why people are laughing?

I think so.

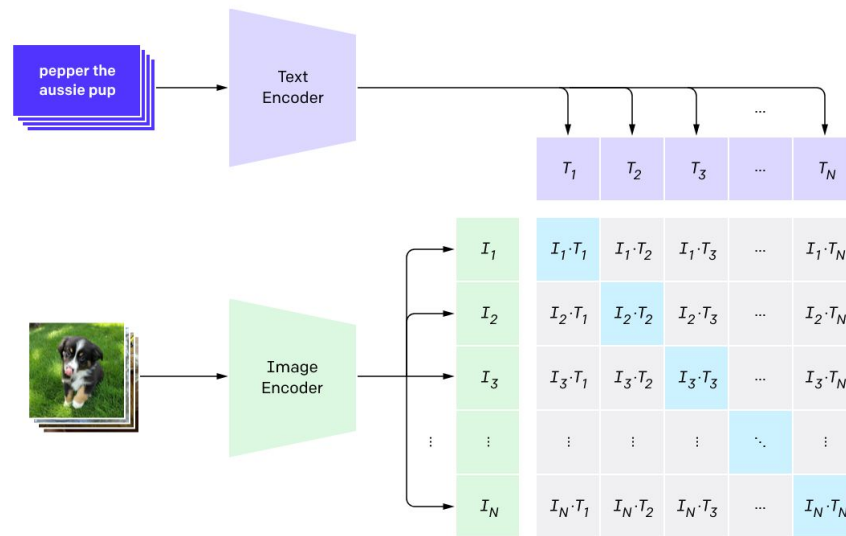
He is standing on a scale.

I think you are right.

CLIP

- контрастивное обучение
- сопоставление эмбеддингов текста и картинок
- много данных:
датасет [WebImageText](#)
400M пар <текст,картинка>

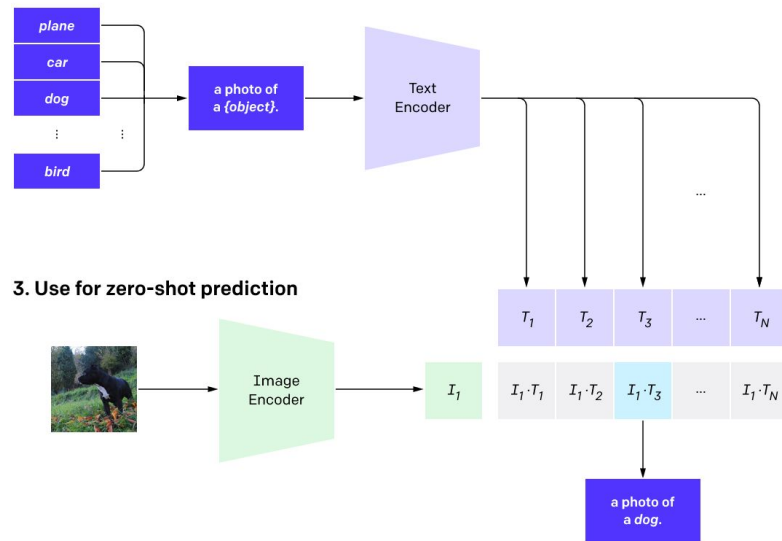
1. Contrastive pre-training



CLIP

Emergent capabilities:
0-shot классификация

2. Create dataset classifier from label text



Мультимодальные модели

По взаимодействию модальностей

- fusion encoder
 - single stream
 - dual stream
- dual encoder

По типу задачи

- cross-modal language modeling
- cross-modal region prediction
- image-text matching
- cross-modal contrastive learning

Downstream задачи:

- Cross-Modal Matching
 - Image Text Retrieval
 - Visual Referring Expression
- Cross-Modal Reasoning
 - Visual Question Answering
- Vision and Language Generation
 - Text-to-Image
 - Multimodal Text Generation

Обзор: “A Survey of Vision-Language Pre-Trained Models” [[IJCAI'22](#)]

Transformers vs CNNs vs MLPs

А лучше ли трансформеры чем CNNs?..

Первоначальный ответ:

- Если мало ресурсов - нет:
у CNN больше инфы благодаря индуктивному bias'у
- Если много - да:
трансформеры лучше скейлятся

Анализ предобучения

Сравнение по объему исходного датасета:

- небольшой: ViT < ResNet
- большой: ViT > ResNet

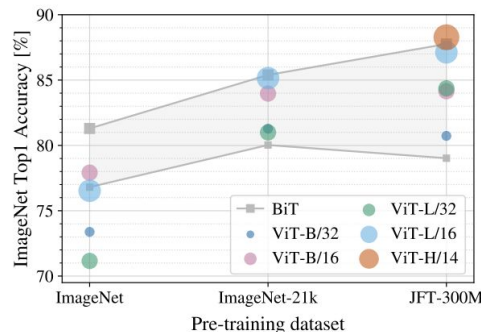


Figure 3: Transfer to ImageNet. While large ViT models perform worse than BiT ResNets (shaded area) when pre-trained on small datasets, they shine when pre-trained on larger datasets. Similarly, larger ViT variants overtake smaller ones as the dataset grows.

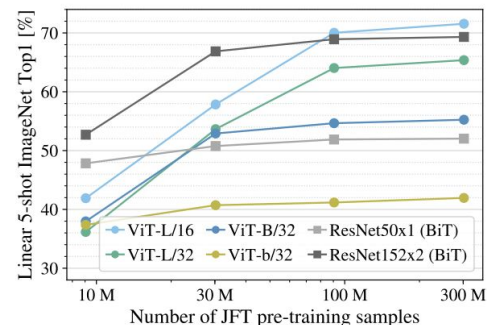


Figure 4: Linear few-shot evaluation on ImageNet versus pre-training size. ResNets perform better with smaller pre-training datasets but plateau sooner than ViT, which performs better with larger pre-training. ViT-b is ViT-B with all hidden dimensions halved.

Анализ предобучения

Сравнение по количеству операций обучения (FLOPs):

- мало: ViT < ResNet
- много: ViT > ResNet

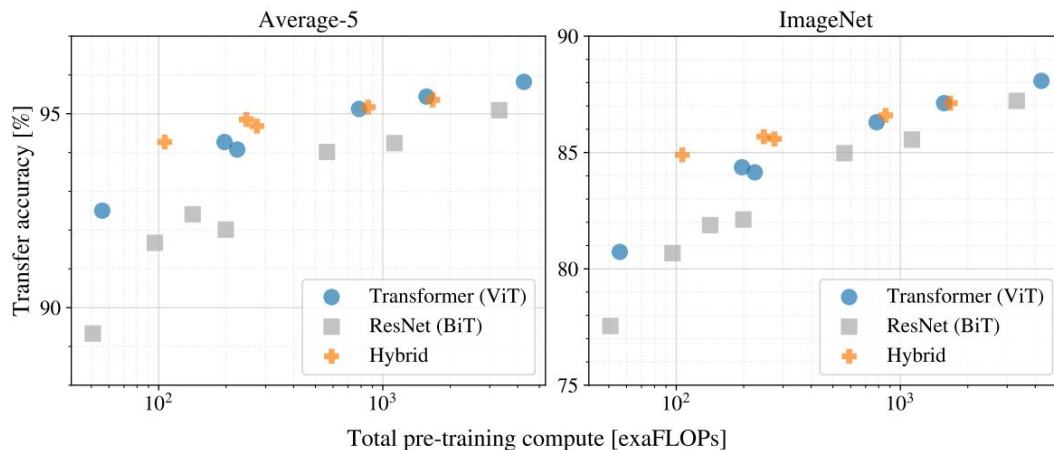
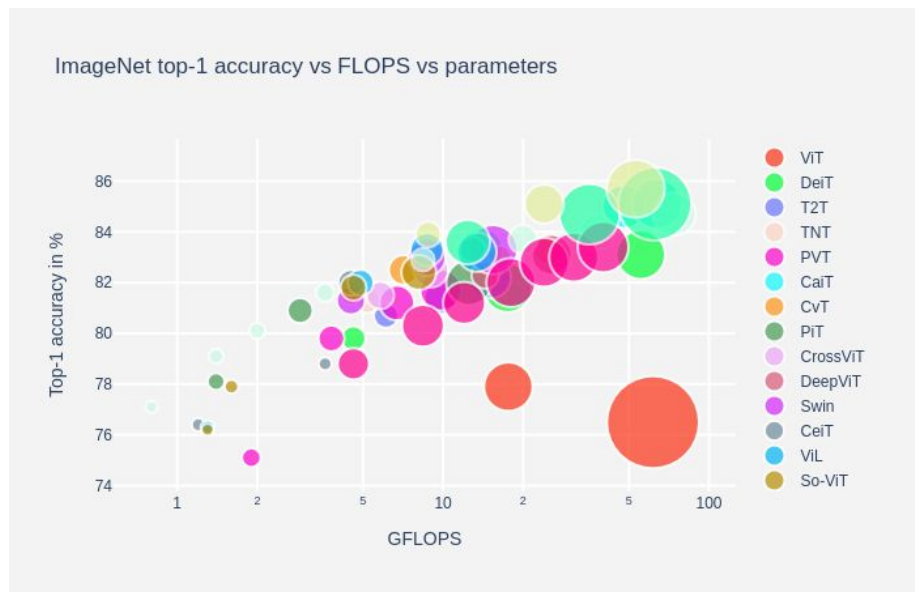


Figure 5: Performance versus pre-training compute for different architectures: Vision Transformers, ResNets, and hybrids. Vision Transformers generally outperform ResNets with the same computational budget. Hybrids improve upon pure Transformers for smaller model sizes, but the gap vanishes for larger models.

ViTs Timeline



А лучше ли трансформеры чем CNNs?..

Даже если ресурсов много - нет: просто сравнивать честнее надо

CNNs for 2020s

“A ConvNet for the 2020s” by Facebook & Berkley, 2022 [[CVPR 2022](#)]

Проблема плохих ablation studies:

ConvNeXt = ResNet + tips & tricks трансформеров ~ трансформеры

Примеры tips & tricks:

- depth-wise convolutions
- inverted bottleneck
- большой kernel size свертка
- ReLU → GELU
- меньше функций активации
- меньше слоев нормализации
- батчнорм → LayerNorm

Архитектура не имеет значения

Если ресурсов много, судя по всему, архитектура не имеет значения:

- CNNs ~ Transformers:
“ConvNets Match Vision Transformers at Scale” by DeepMind, 2023 [[arxiv](#)]
- MLPs ~ Transformers:
“MLP-Mixer: An all-MLP Architecture for Vision” by Google, 2021 [[NeurIPS'21](#)]

Спасибо за внимание!