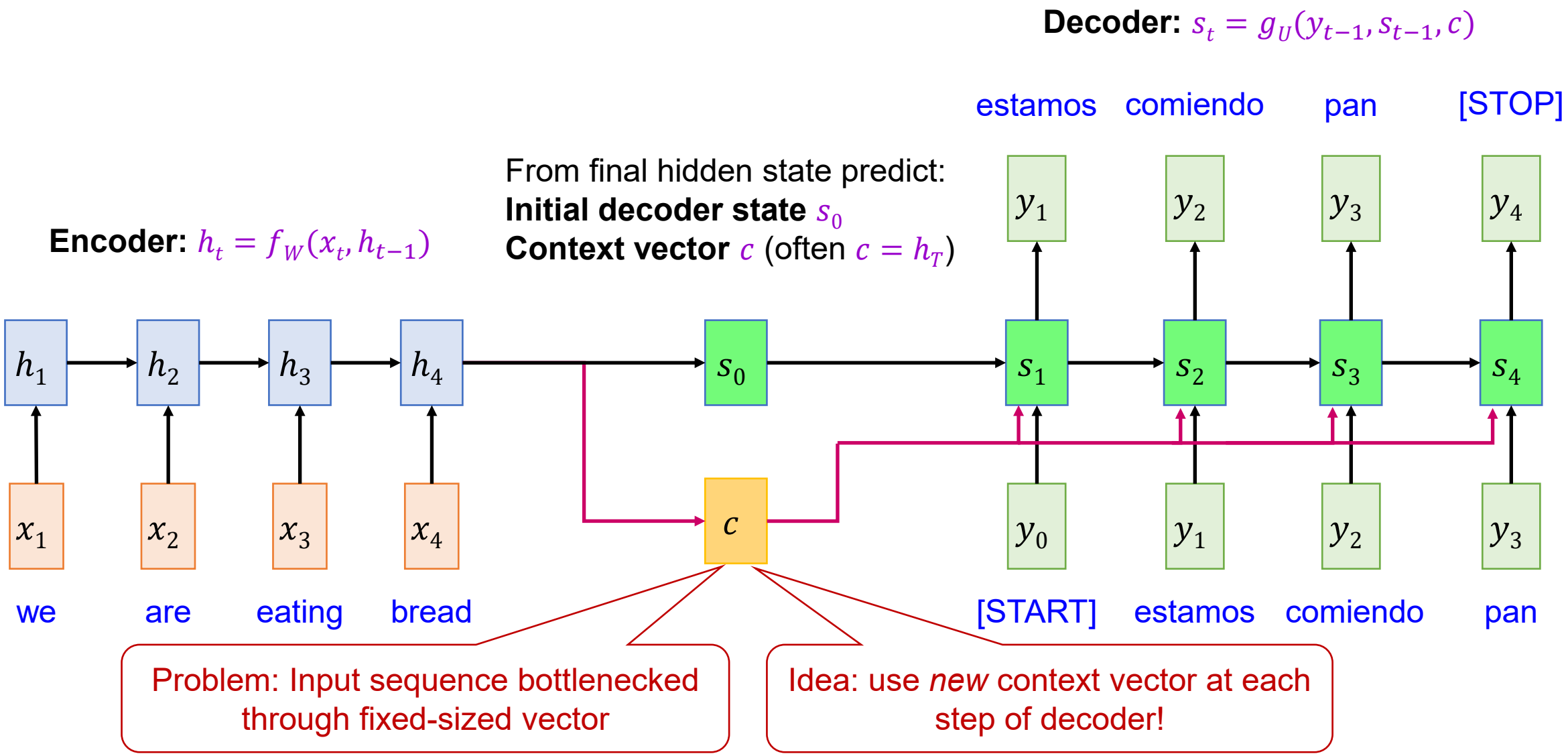
Lecture 10: Transformers, BERT, GPT and beyond

Olexandr Isayev

Department of Chemistry, CMU

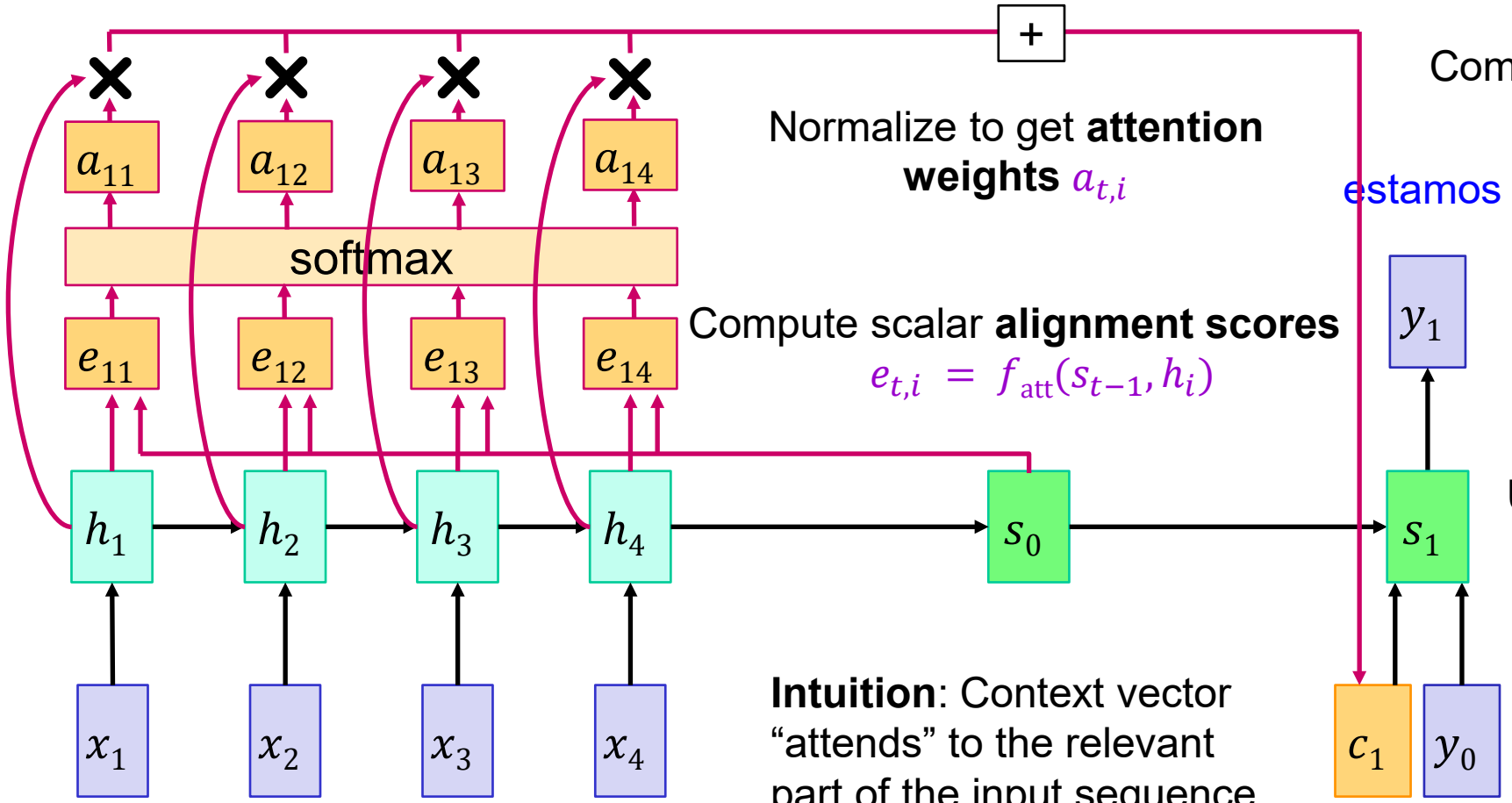
olexandr@cmu.edu

Sequence-to-sequence with RNNs



A. Sutskever, O. Vinyals, Q. Le, Sequence to sequence learning with neural networks, NeurIPS 2014

Sequence-to-sequence with RNNs and attention



Normalize to get **attention weights** $a_{t,i}$

Compute scalar **alignment scores**

$$e_{t,i} = f_{\text{att}}(s_{t-1}, h_i)$$

Compute **context vector** as

$$c_t = \sum_i a_{t,i} h_i$$

Use context vector in decoder:

$$s_t = g_U(y_{t-1}, s_{t-1}, c_t)$$

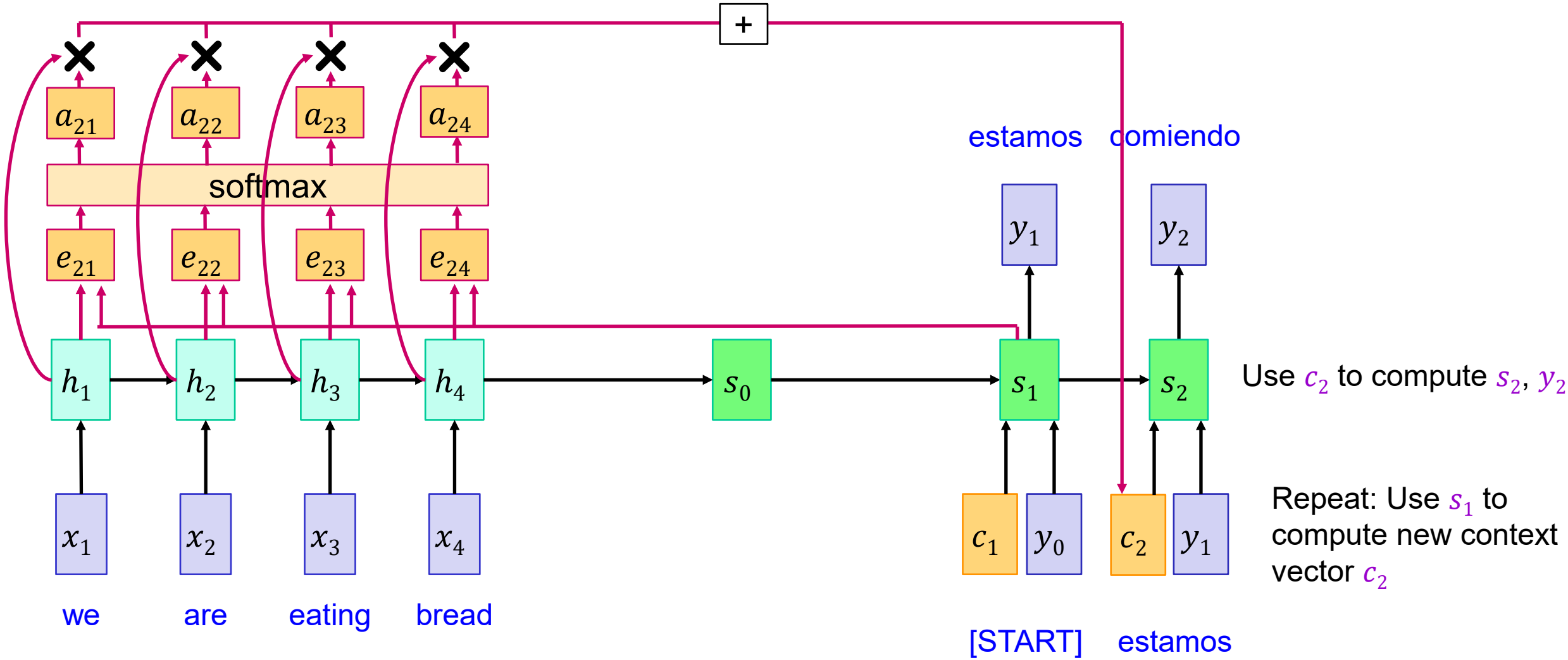
Intuition: Context vector "attends" to the relevant part of the input sequence

"estamos" **=** *"we are"*

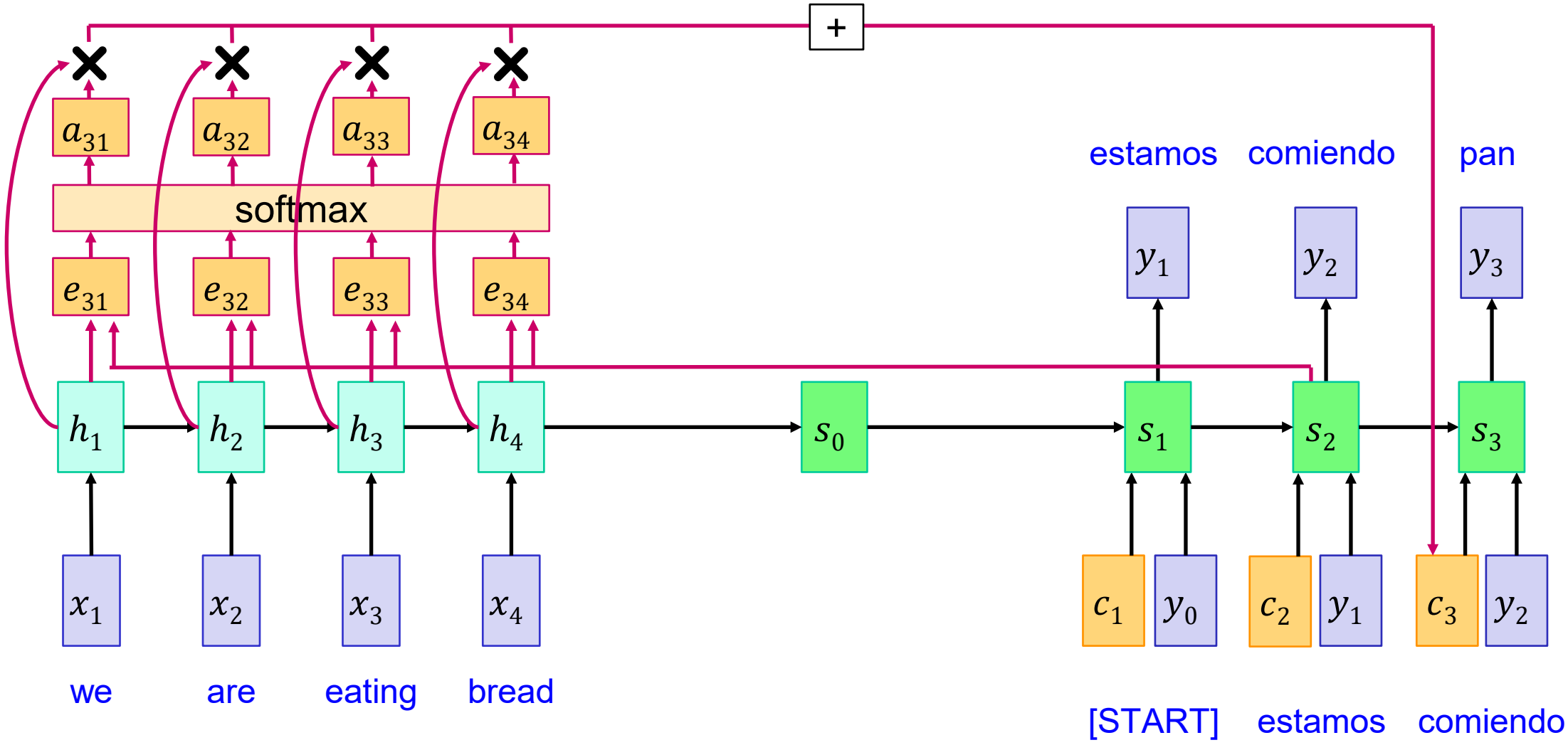
so maybe $a_{11} = a_{12} = 0.45$, $a_{13} = a_{14} = 0.05$

D. Bahdanau, K. Cho, Y. Bengio, Neural Machine Translation by Jointly Learning to Align and Translate, ICLR 2015

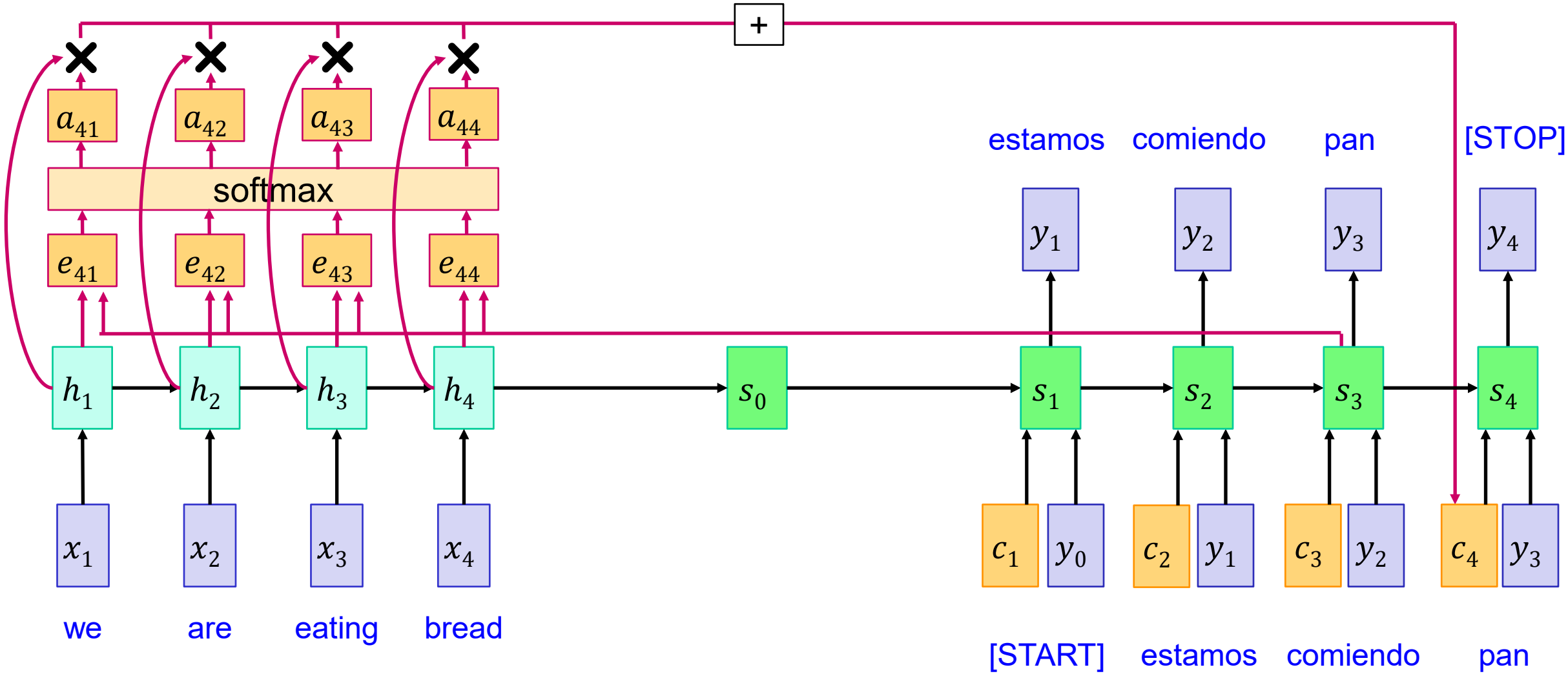
Sequence-to-sequence with RNNs and attention



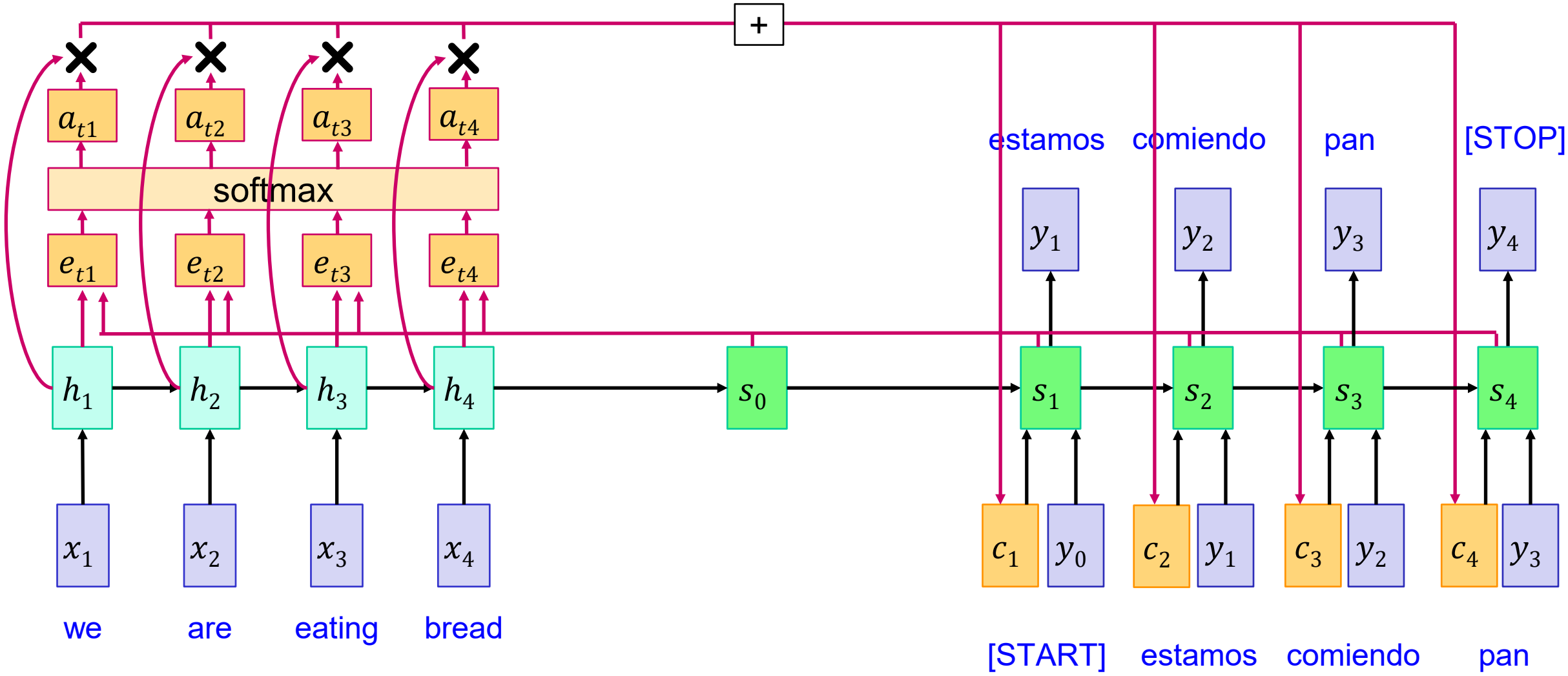
Sequence-to-sequence with RNNs and attention



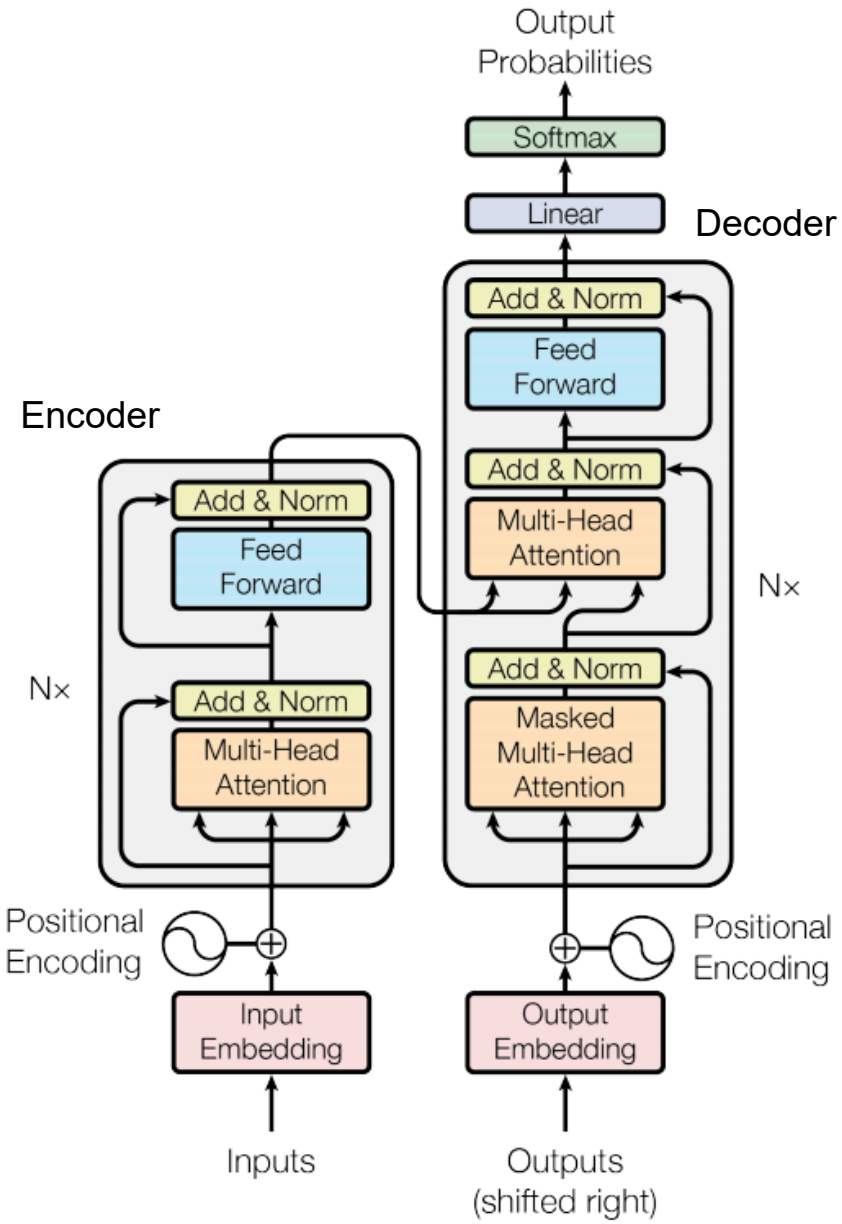
Sequence-to-sequence with RNNs and attention



Sequence-to-sequence with RNNs and attention

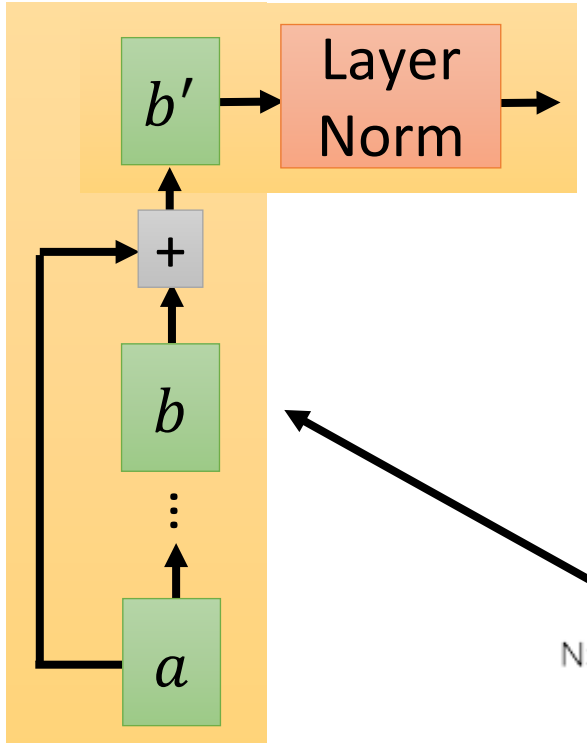


Transformer architecture



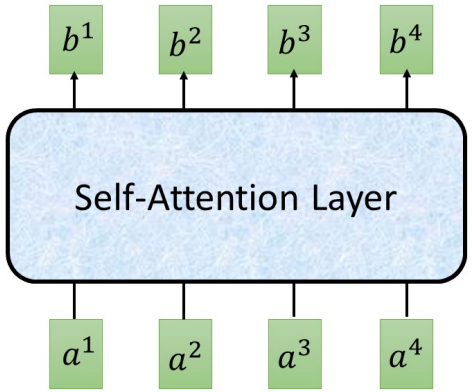
A. Vaswani et al., Attention is all you need, NeurIPS 2017

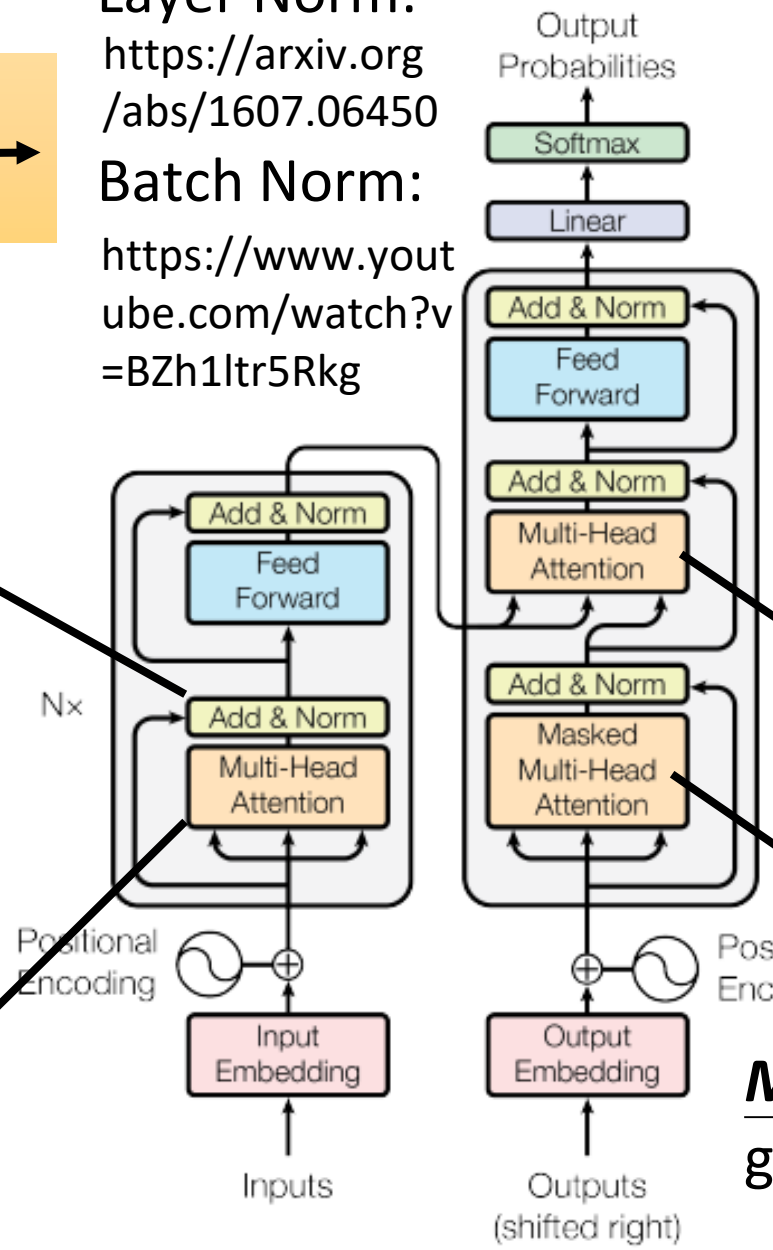
Transformer



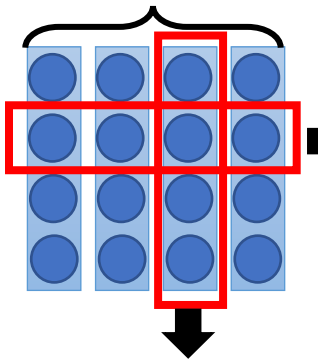
Layer Norm:
https://arxiv.org/abs/1607.06450

Batch Norm:
https://www.youtube.com/watch?v=BZh1ltr5Rkg





Batch Size



$\mu = 0, \sigma = 1$ Batch

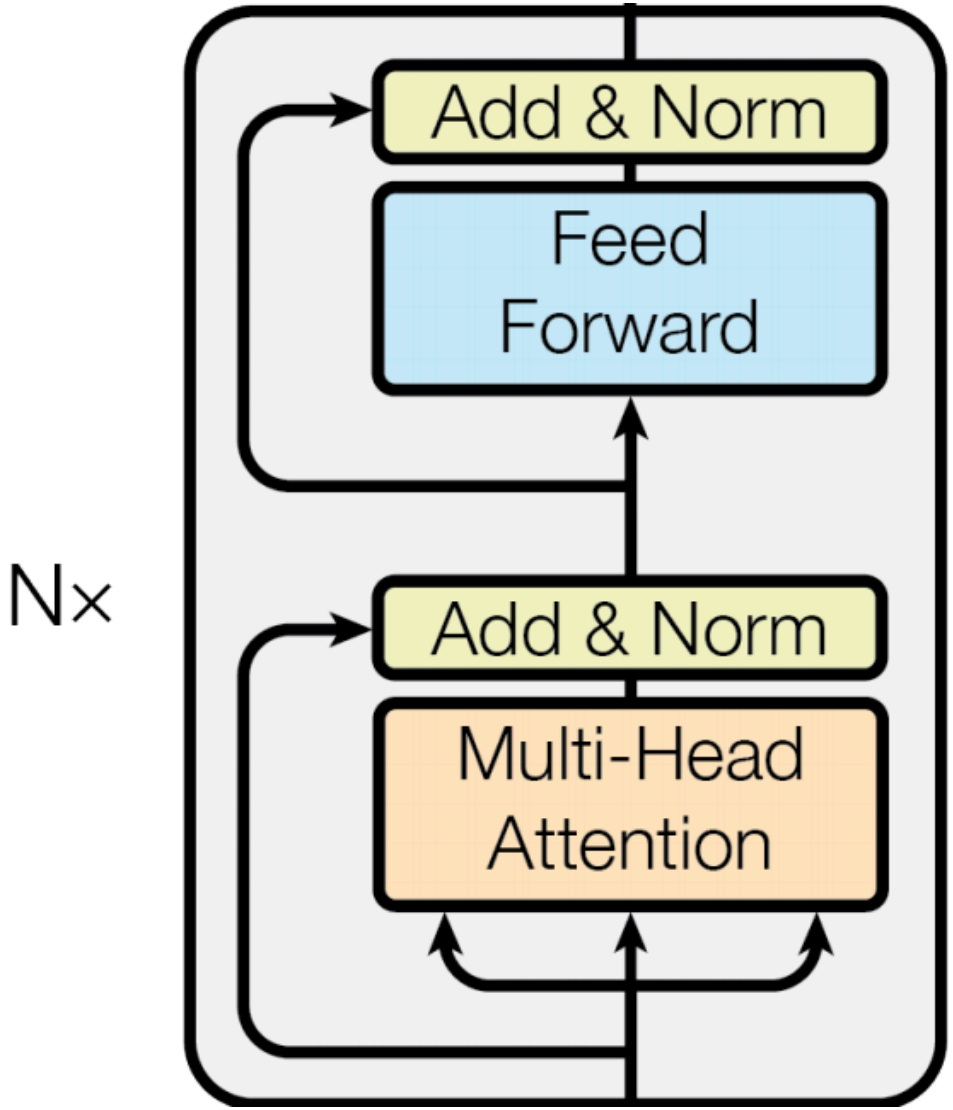
$\mu = 0, \sigma = 1$ Layer

attend on the input sequence

Masked: attend on the generated sequence

Transformer blocks

- A **Transformer** is a sequence of transformer blocks
  - Vaswani et al.: N=12 blocks, embedding dimension = 512, 6 attention heads
  - **Add & Norm:** residual connection followed by layer normalization
  - **Feedforward:** two linear layers with ReLUs in between, applied independently to each vector
- Attention is the only interaction between inputs!

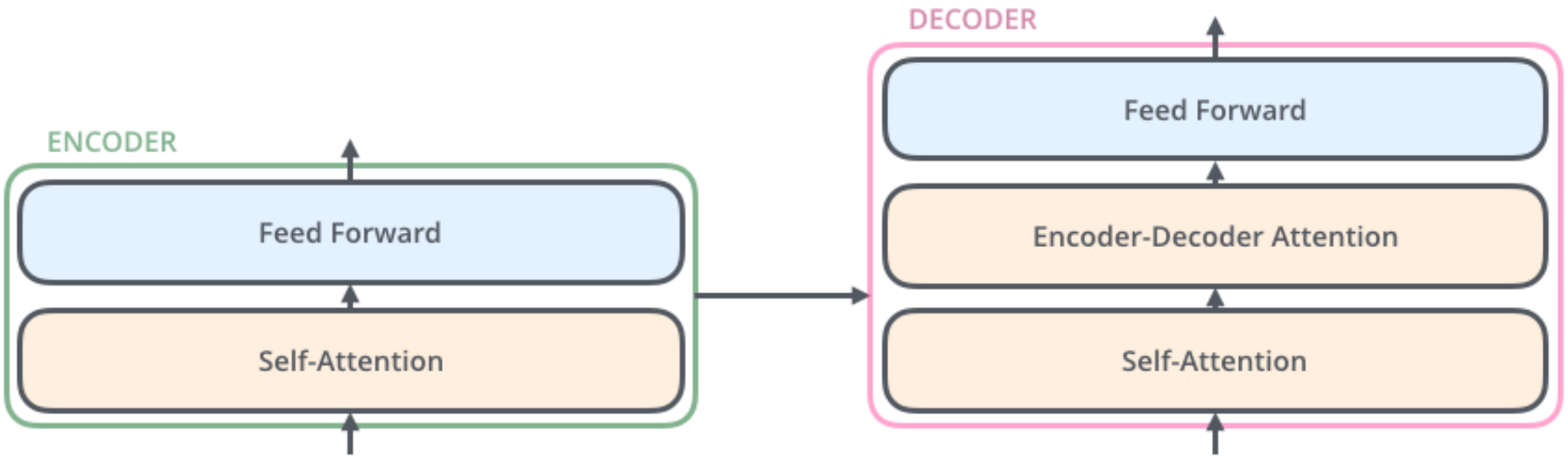


Basic transformer model (review)

- Sequence-to-sequence architecture using only point-wise processing and attention (no recurrent units or convolutions)

Encoder: receives entire input sequence and outputs encoded sequence of the same length

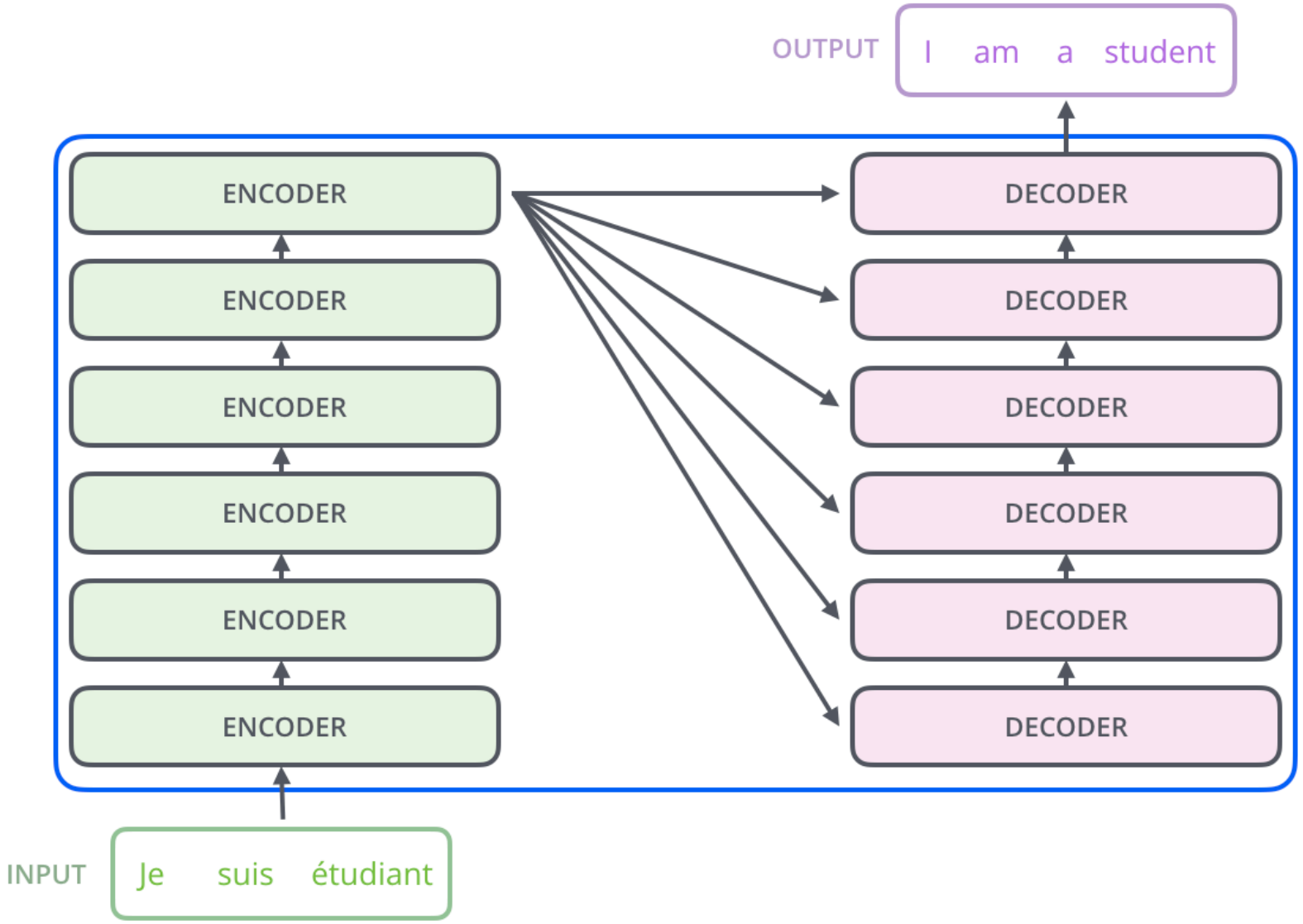
Decoder: predicts next token conditioned on encoder output and previously predicted tokens



A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, NeurIPS 2017

Image source

Transformer



Key-Value-Query attention model

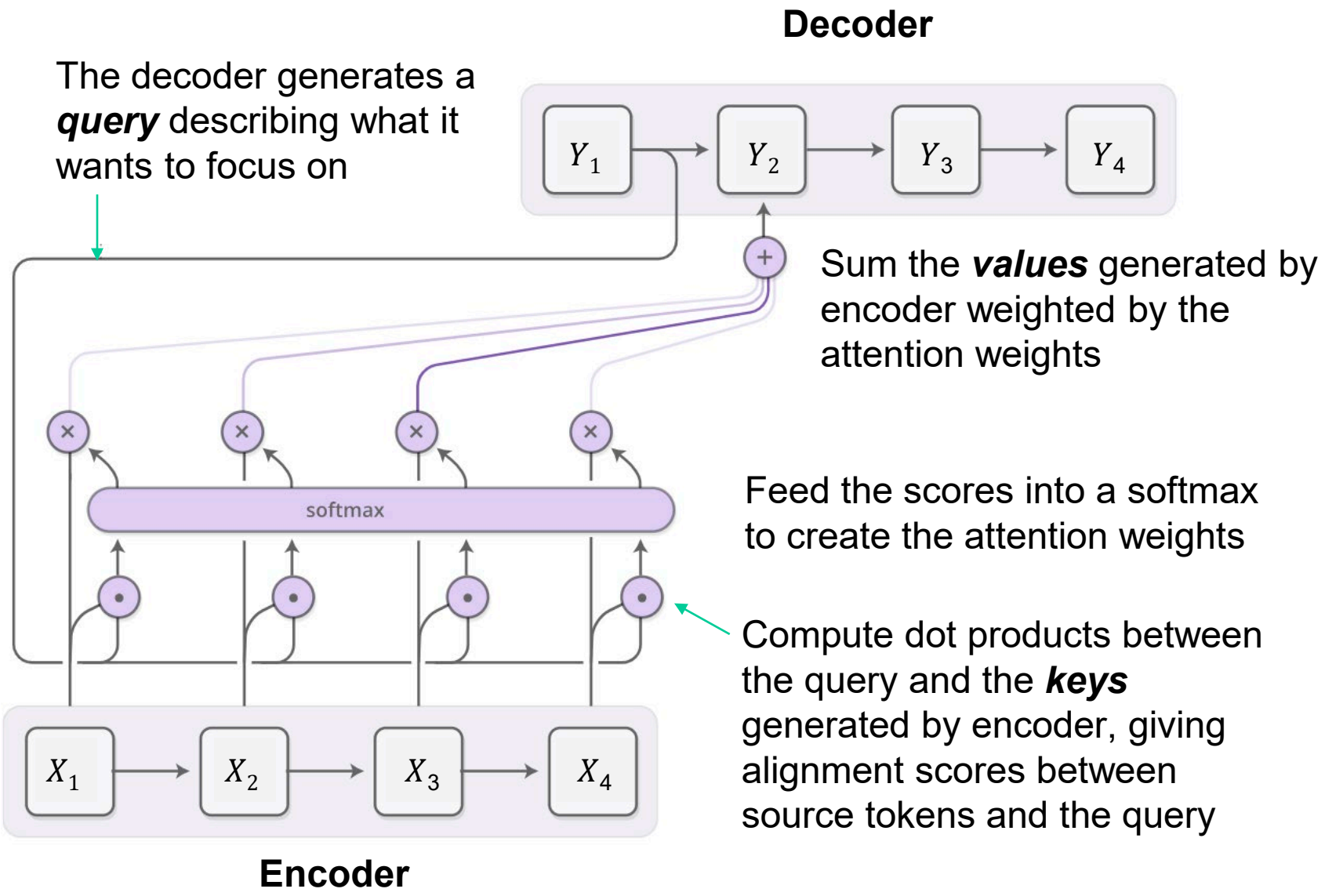


Image source

Key-Value-Query attention model

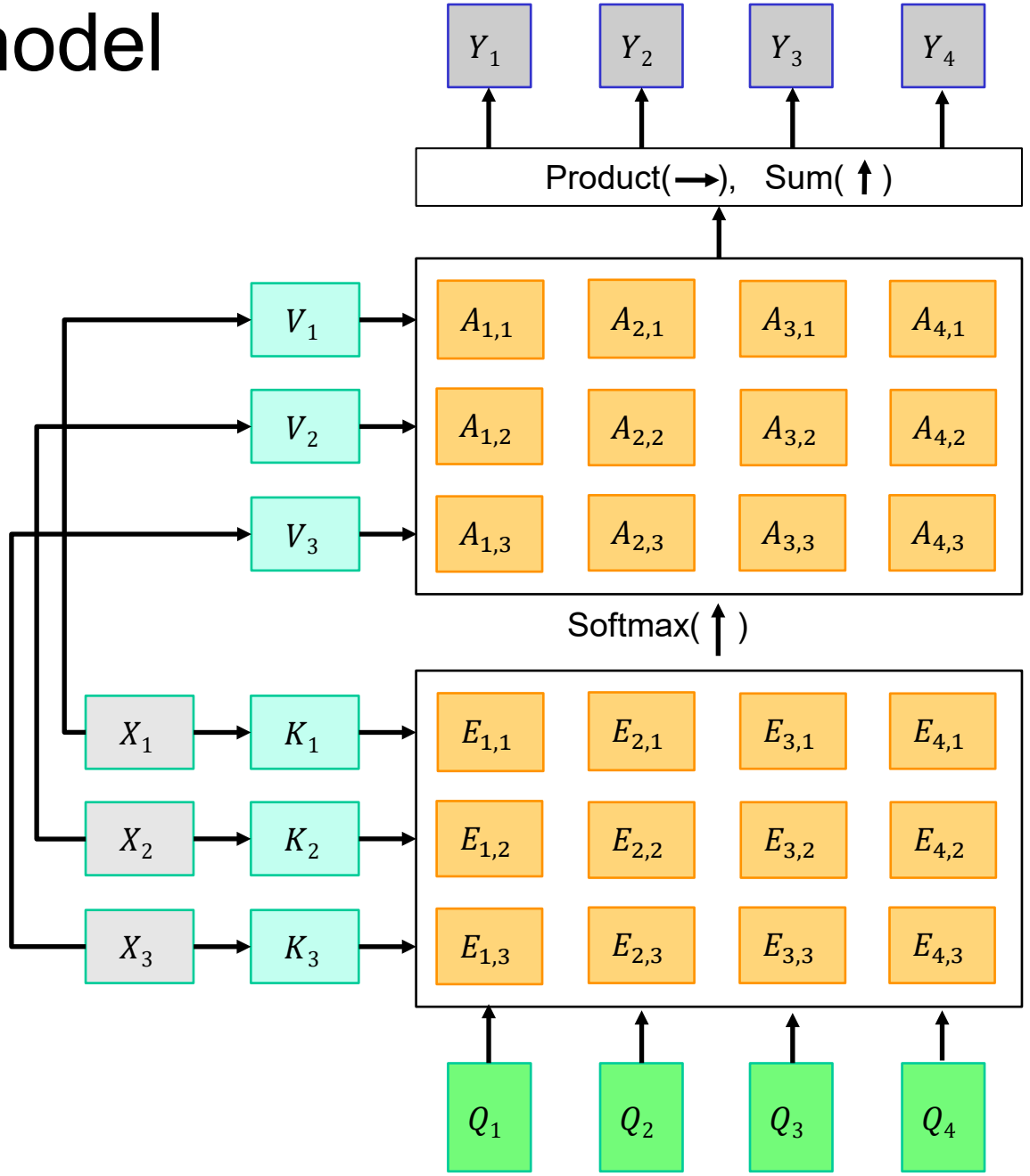
- Key vectors: $K = XW_K$
- Value Vectors: $V = XW_V$
- Query vectors
- Similarities: *scaled dot-product attention*

$$E_{i,j} = \frac{(Q_i \cdot K_j)}{\sqrt{D}} \text{ or } E = QK^T/\sqrt{D}$$

(D is the dimensionality of the keys)

- Attn. weights: $A = \text{softmax}(E, \text{dim} = 1)$
- Output vectors:

$$Y_i = \sum_j A_{i,j} V_j \text{ or } Y = AV$$



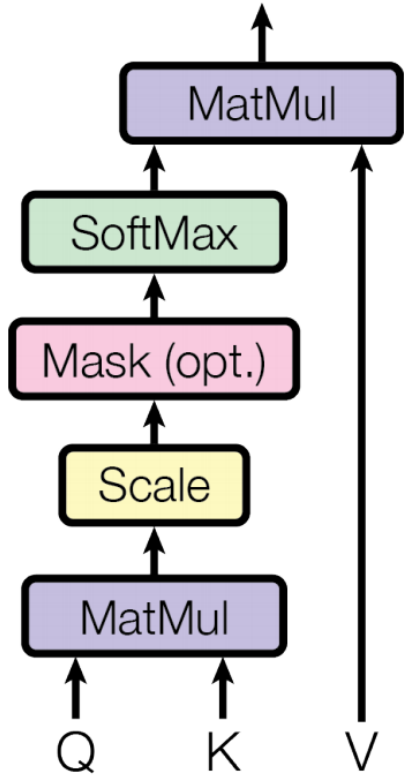
Adapted from J. Johnson

Scaled Dot-Product Attention

Problem: As d_k gets large, the variance of $q^T k$ increases → some values inside the softmax get large → the softmax gets very peaked → hence its gradient gets smaller.

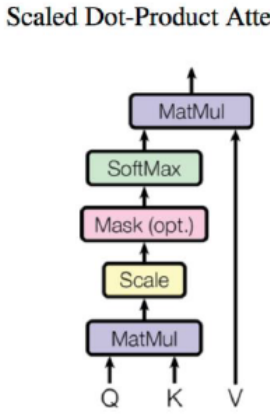
Solution: Scale by length of query/key vectors:

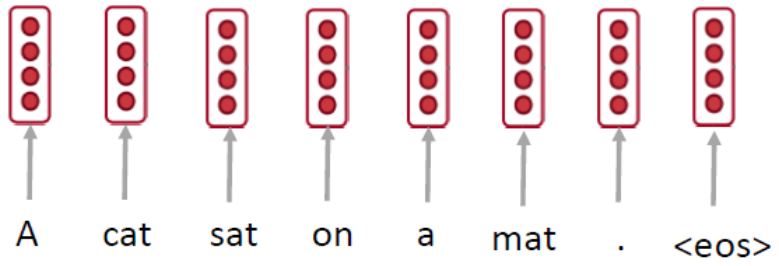
$$A(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$



Self-attention: A Running Example

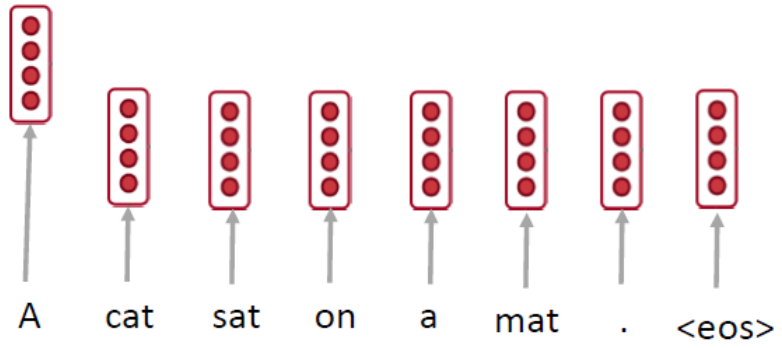
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



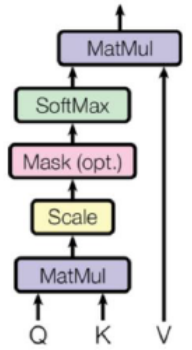


Self-attention: A Running Example

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

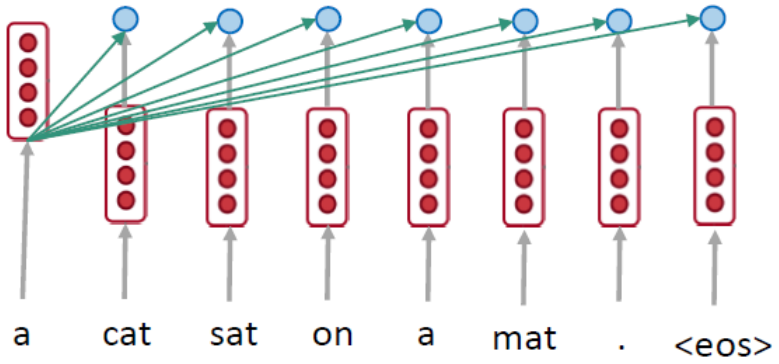


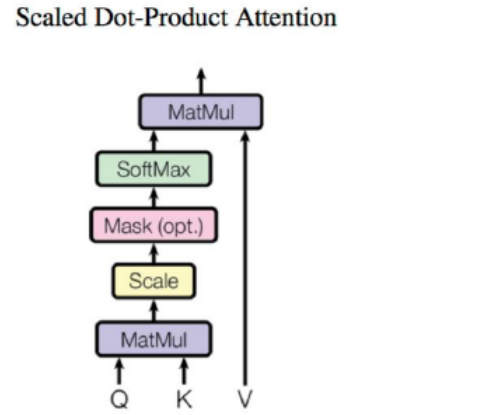
Scaled Dot-Product Attention



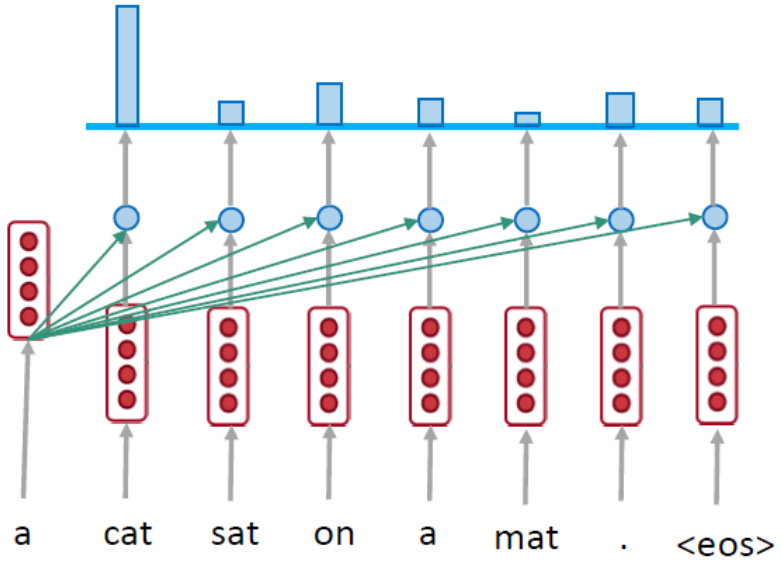
Self-attention: A Running Example

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

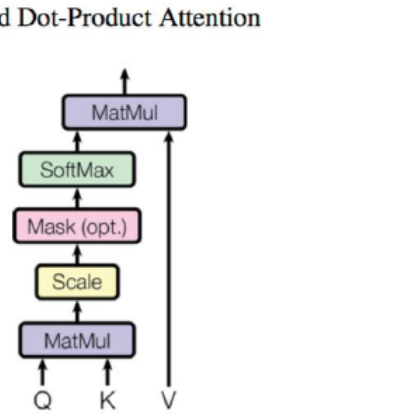




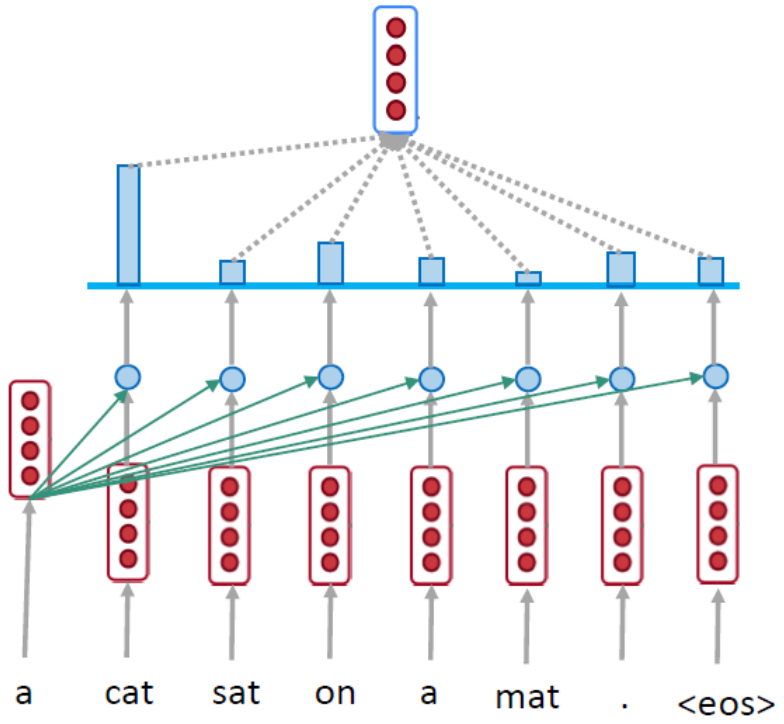
Self-attention: A Running Example



$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

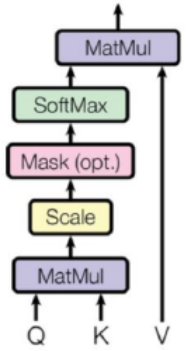


Self-attention: A Running Example

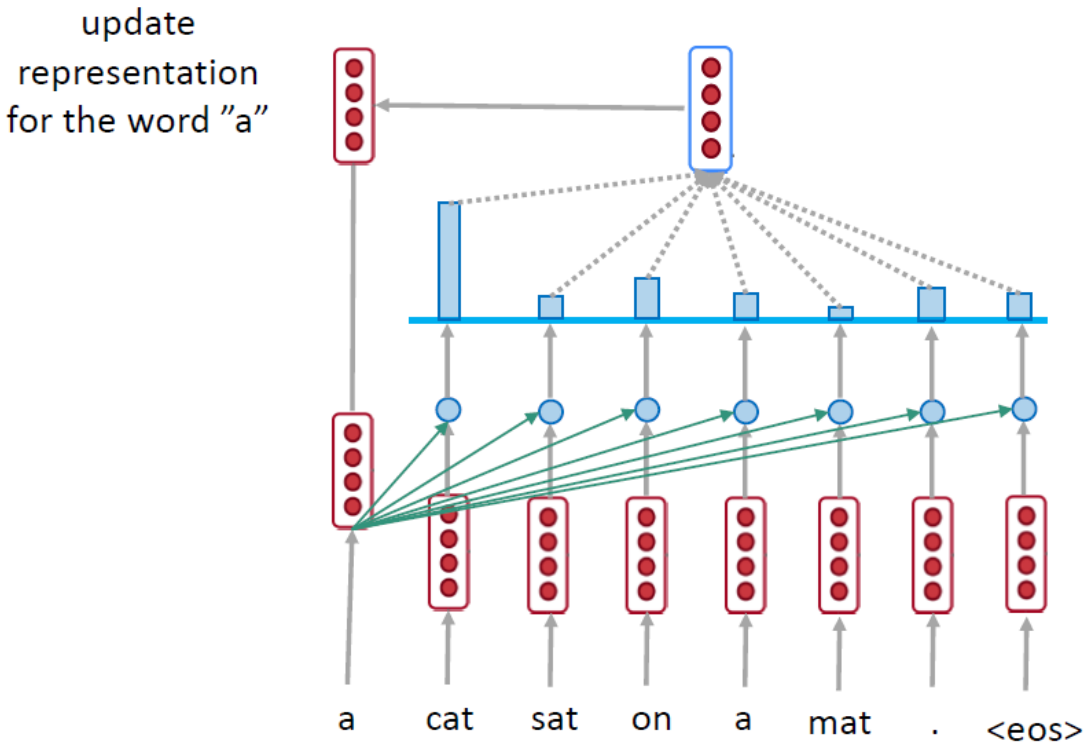


$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

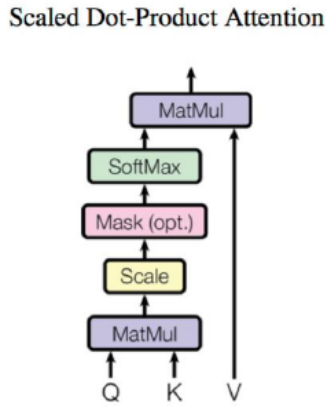
Scaled Dot-Product Attention



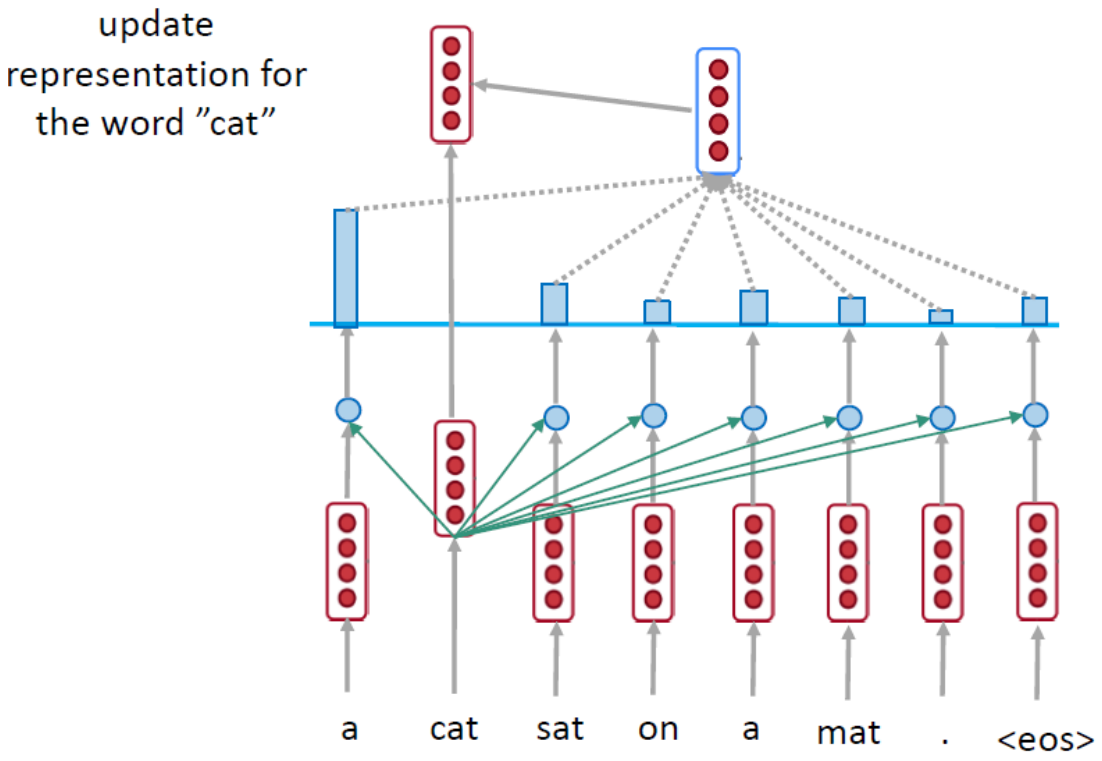
Self-attention: A Running Example



$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

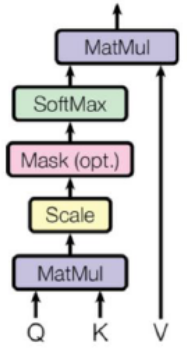


Self-attention: A Running Example



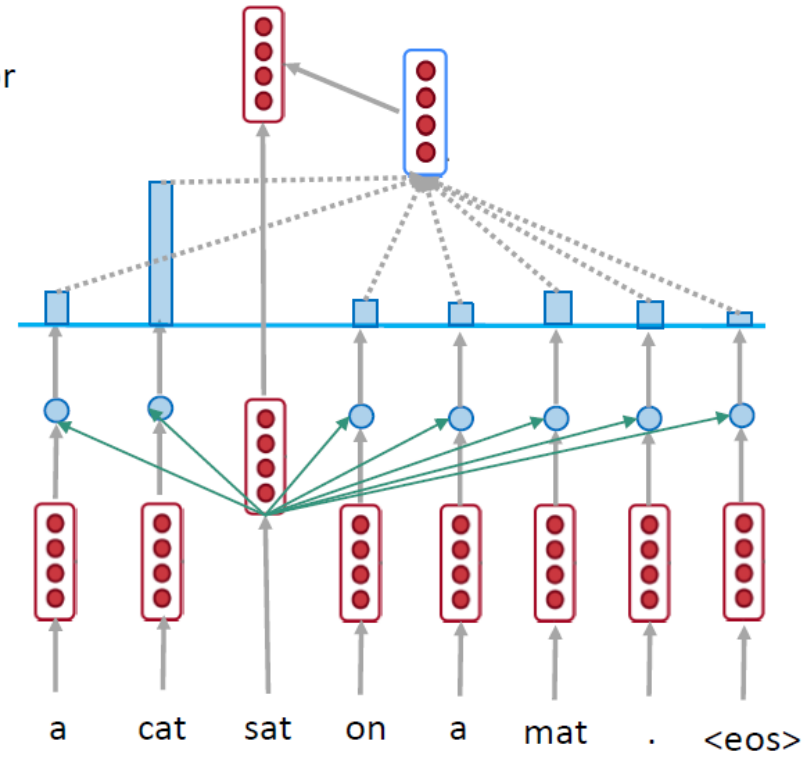
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Scaled Dot-Product Attention



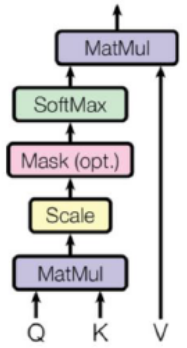
Self-attention: A Running Example

update representation for the word "sat"



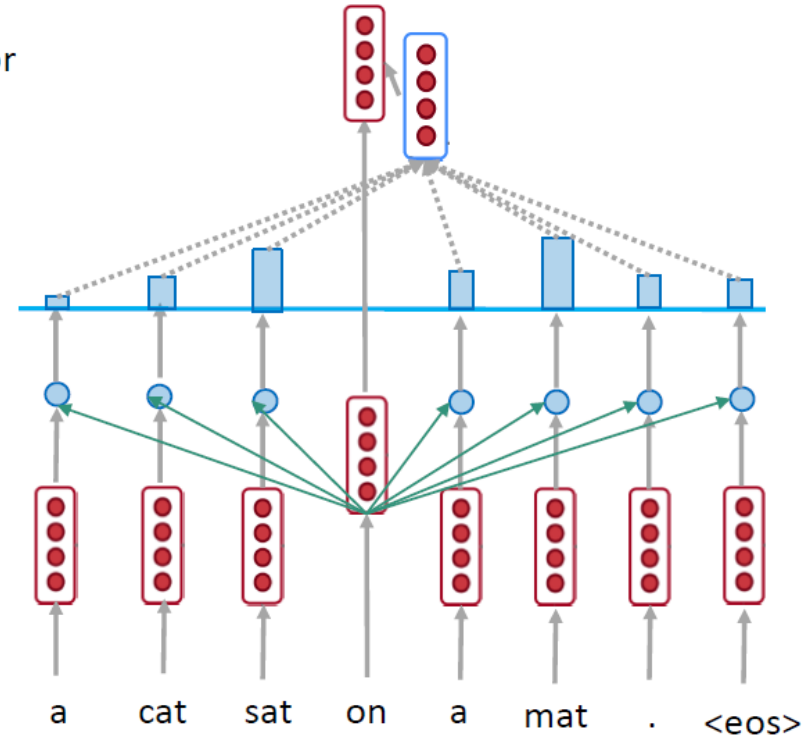
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Scaled Dot-Product Attention



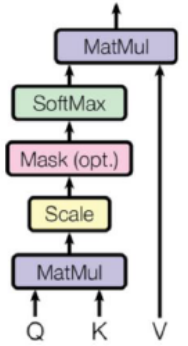
Self-attention: A Running Example

update representation for the word "on"



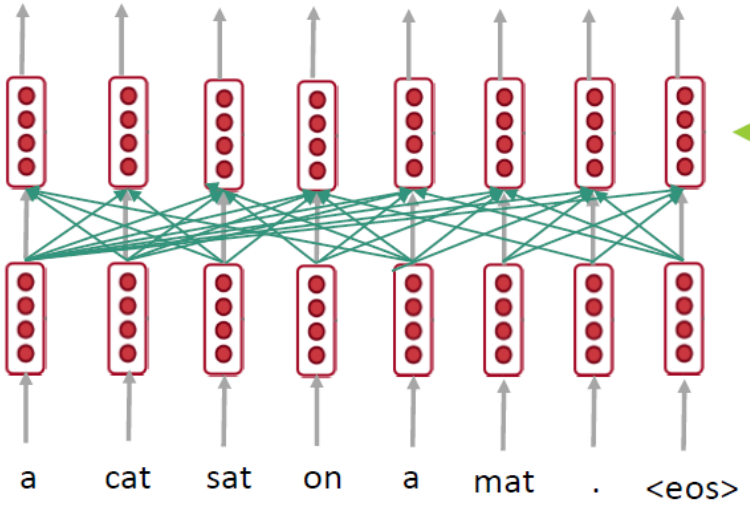
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Scaled Dot-Product Attention



Self-attention: A Running Example

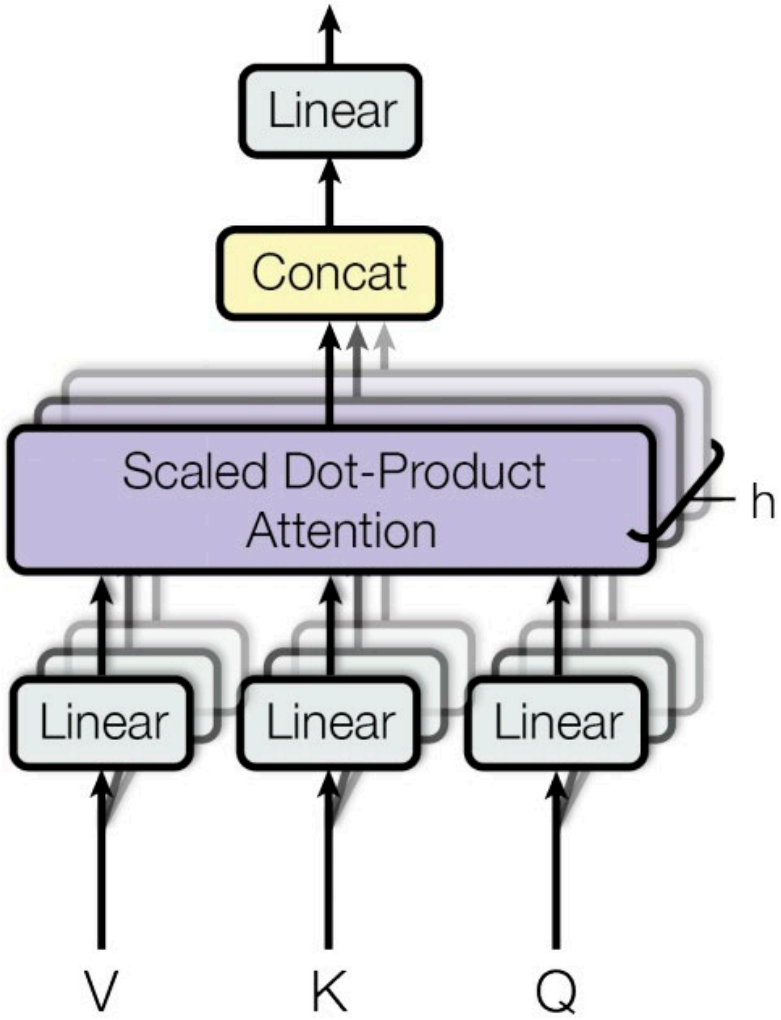
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



Updated representations
for each word
(one layer)

Multi-head attention

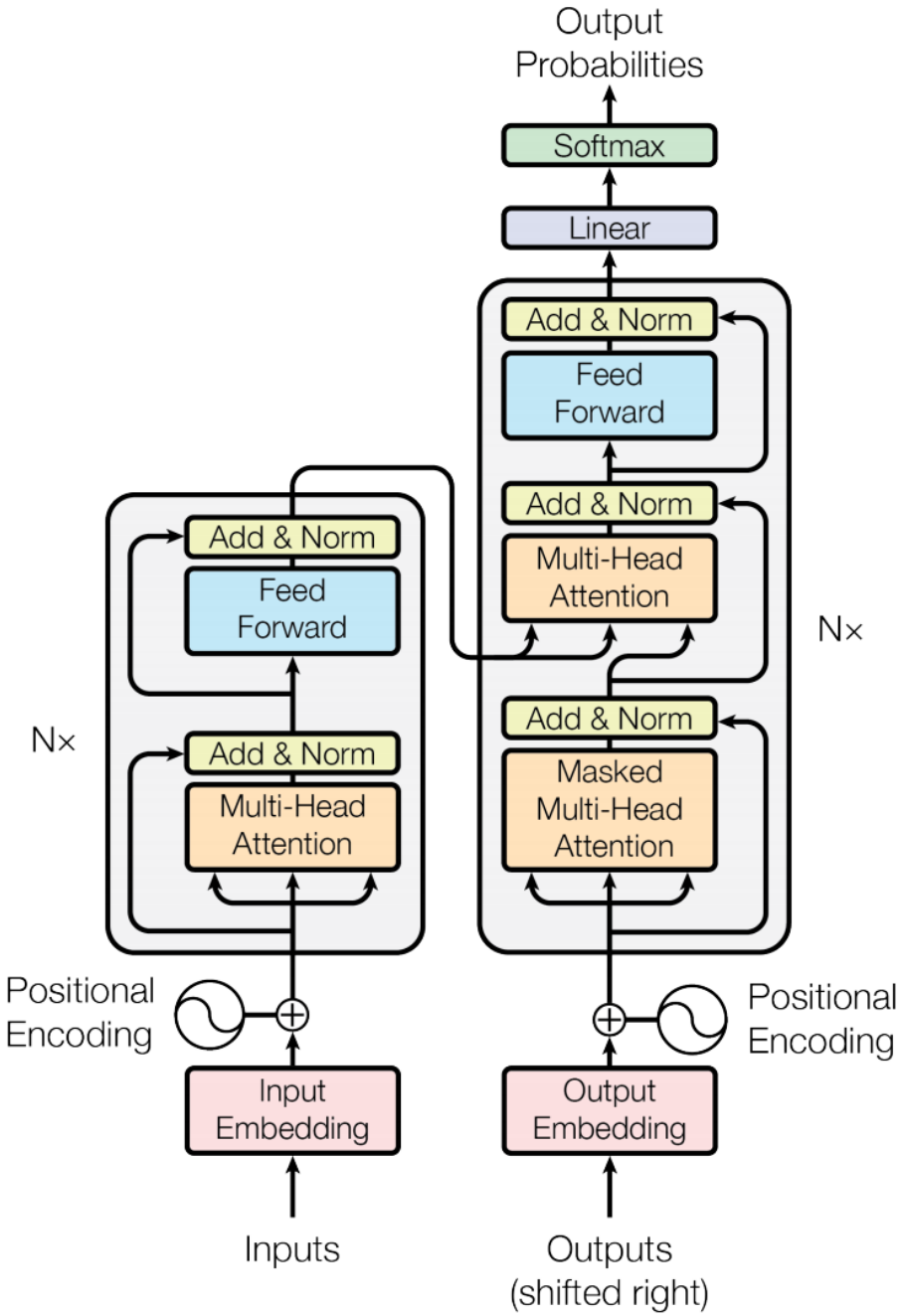
- Run h attention models in parallel on top of different linearly projected versions of Q, K, V; concatenate and linearly project the results
- Intuition: enables model to attend to different kinds of information at different positions (see visualization tool)



$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$

$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Blocks are repeated
N=6 times

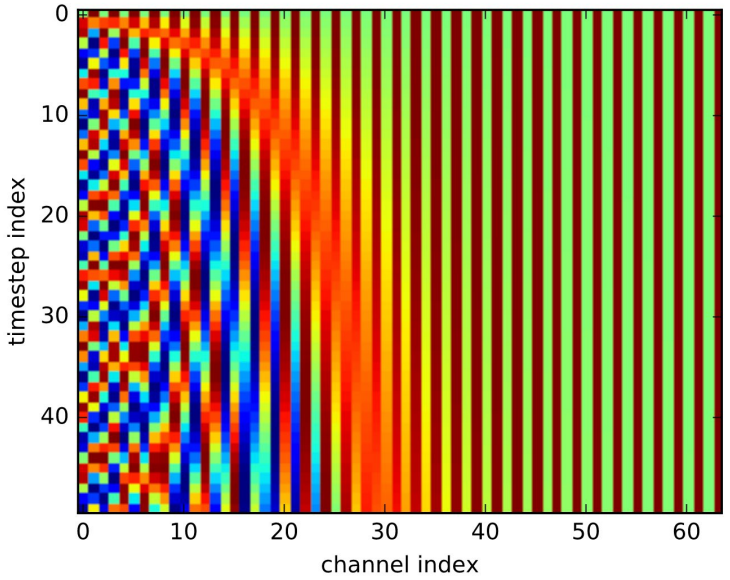


Encoder Input

- Actual word representations are byte-pair encodings
  - Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. ACL 2016.
- Added is a positional encoding so same words at different locations have different overall representations:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

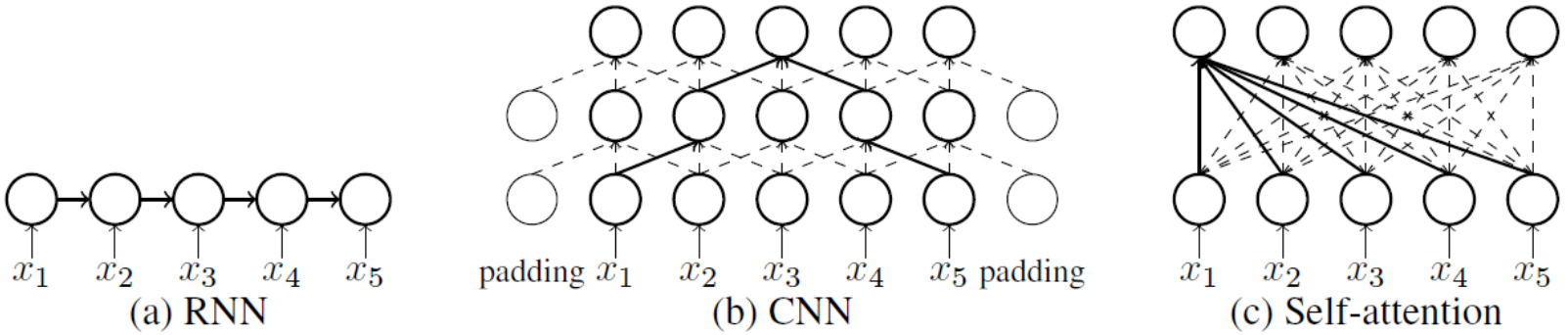
pos is the position of a word
i is the dimension index



Advantages

- No recurrence: parallel encoding
- Fast training: both encoder and decoder are parallel
- No long range problem: O(1) for all tokens direct connections
- Three attentions: the model does not have to remember too much
- Multi-head attention allows to pay attention to different aspects

- Why self-attention and CNN is better than RNN on NMT is still under investigation



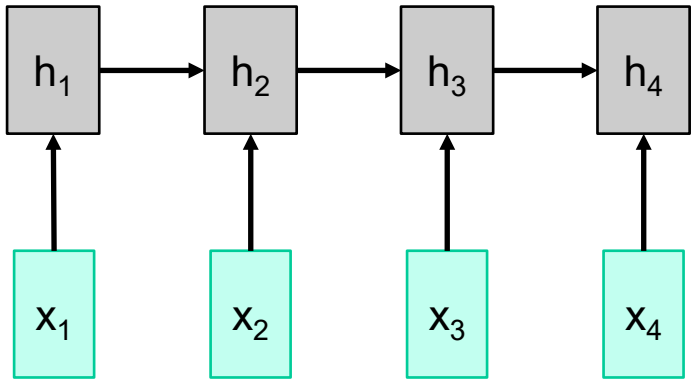
A comparison of RNN, CNN, and self-attention
http://aclweb.org/anthology/D18-1458

Transformer Networks Visualization

https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

Different ways of processing sequences

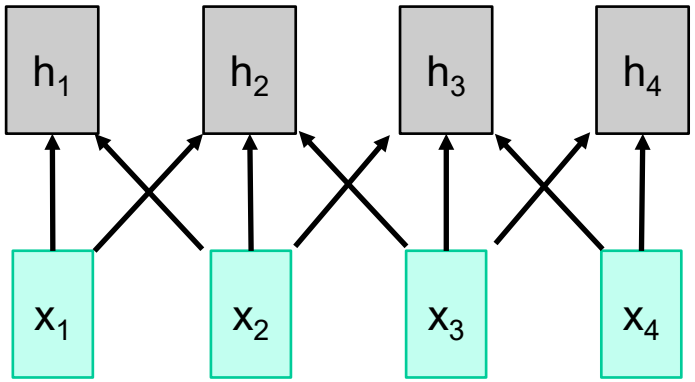
RNN



Works on **ordered sequences**

- Pros: Good for long sequences: After one RNN layer, h_T "sees" the whole sequence
- Con: Not parallelizable: need to compute hidden states sequentially
- Con: Hidden states have limited expressive capacity

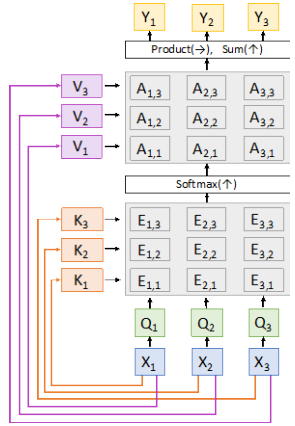
1D convolutional network



Works on **multidimensional grids**

- Pro: Each output can be computed in parallel (at training time)
- Con: Bad at long sequences: Need to stack many conv layers for outputs to "see" the whole sequence

Transformer



- Works on **sets of vectors**
- Pro: Good at long sequences: after one self-attention layer, each output "sees" all inputs!
- Pro: Each output can be computed in parallel (at training time)
- Con: Memory-intensive

Image transformer – Google

- Image generation and super-resolution with 32x32 output, attention restricted to local neighborhoods

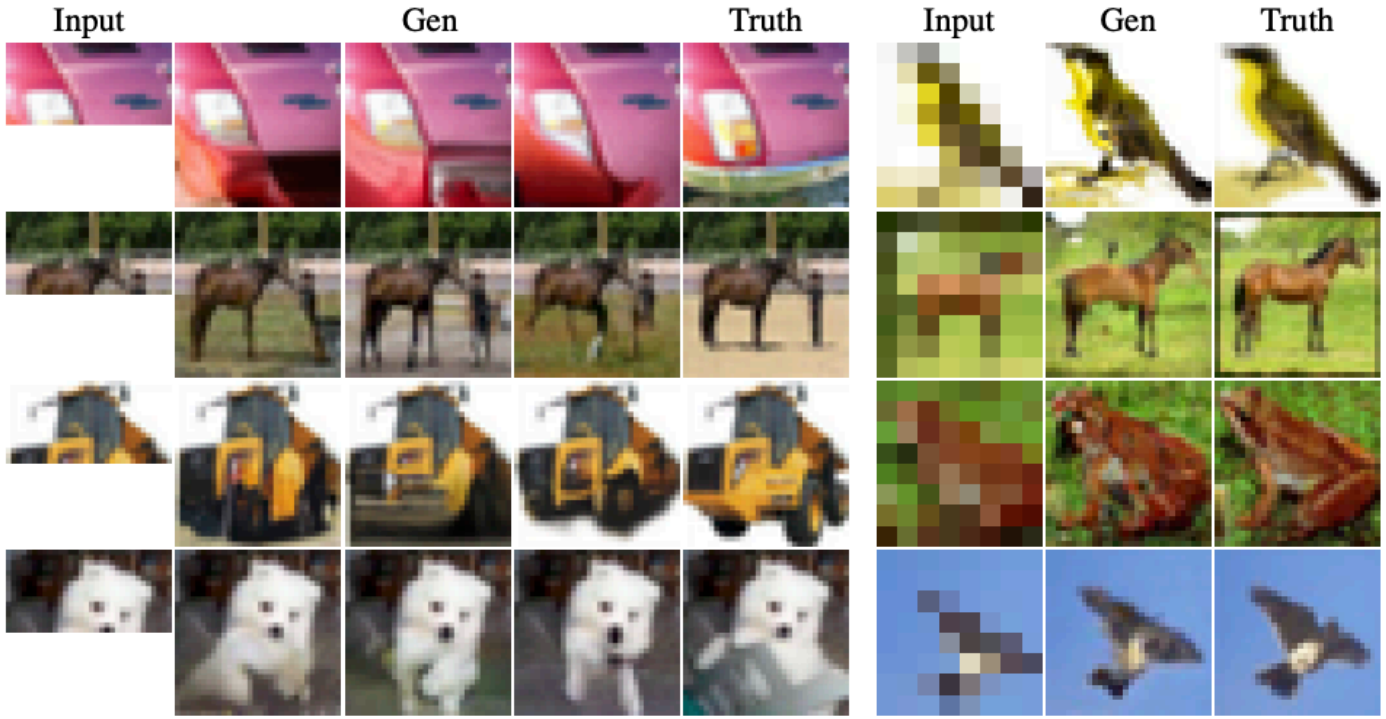


Table 2. On the left are image completions from our best conditional generation model, where we sample the second half. On the right are samples from our four-fold super-resolution model trained on CIFAR-10. Our images look realistic and plausible, show good diversity among the completion samples and observe the outputs carry surprising details for coarse inputs in super-resolution.

N. Parmar et al., Image transformer, ICML 2018

Sparse transformers – OpenAI

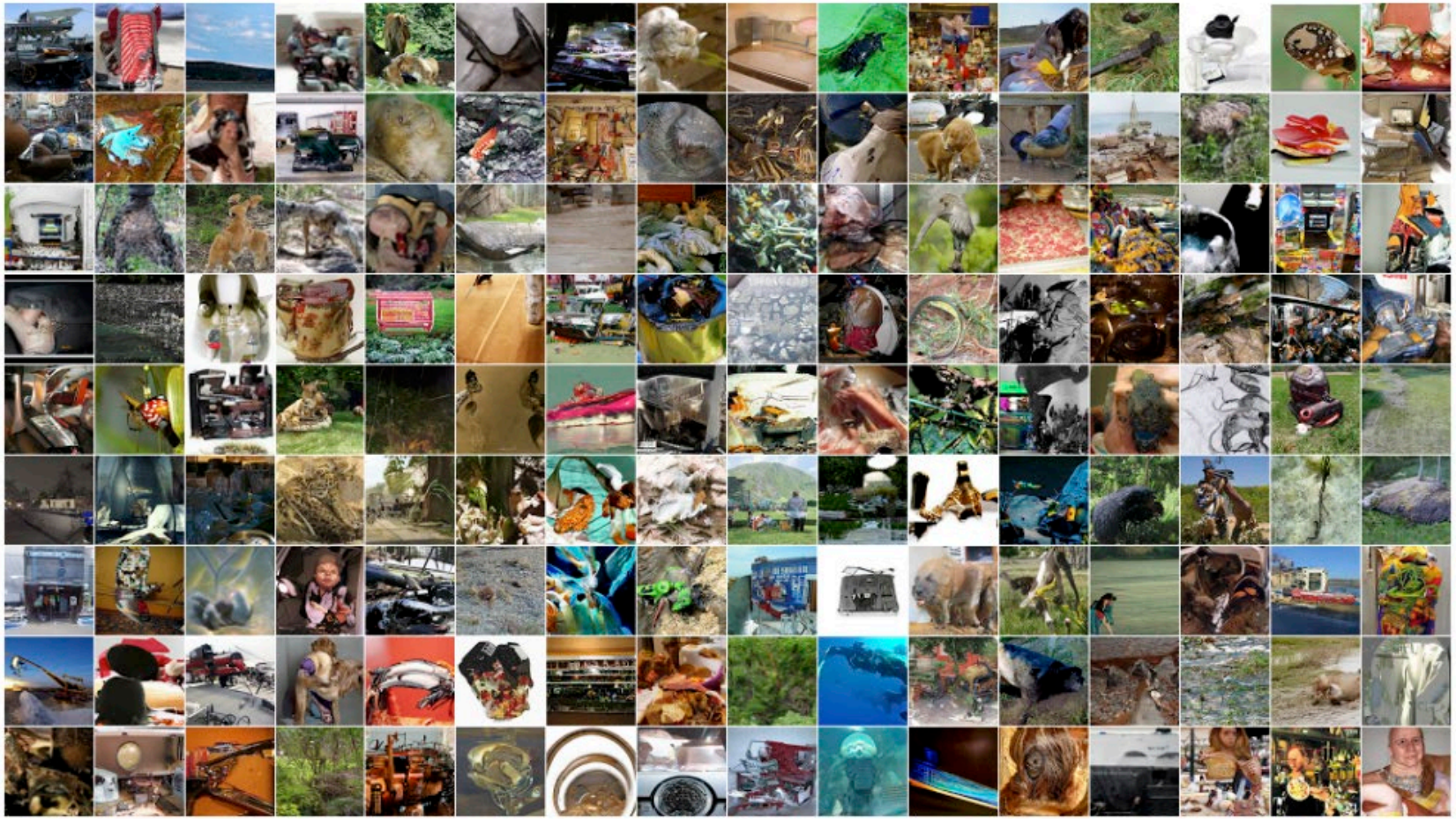
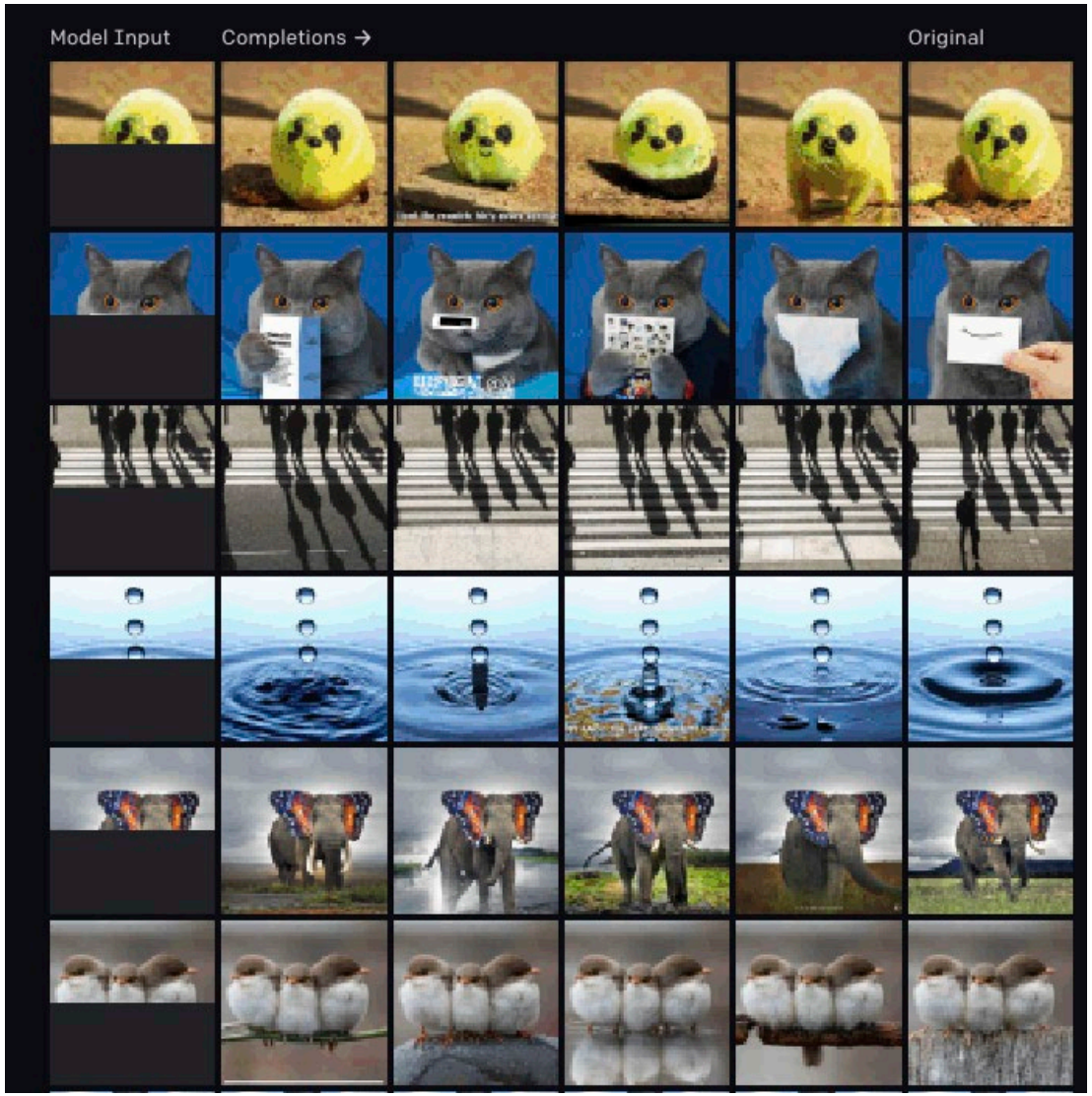


Figure 5. Unconditional samples from ImageNet 64x64, generated with an unmodified softmax temperature of 1.0. We are able to learn long-range dependencies directly from pixels without using a multi-scale architecture.

R. Child et al., Generating Long Sequences with Sparse Transformers, arXiv 2019

Image GPT – OpenAI

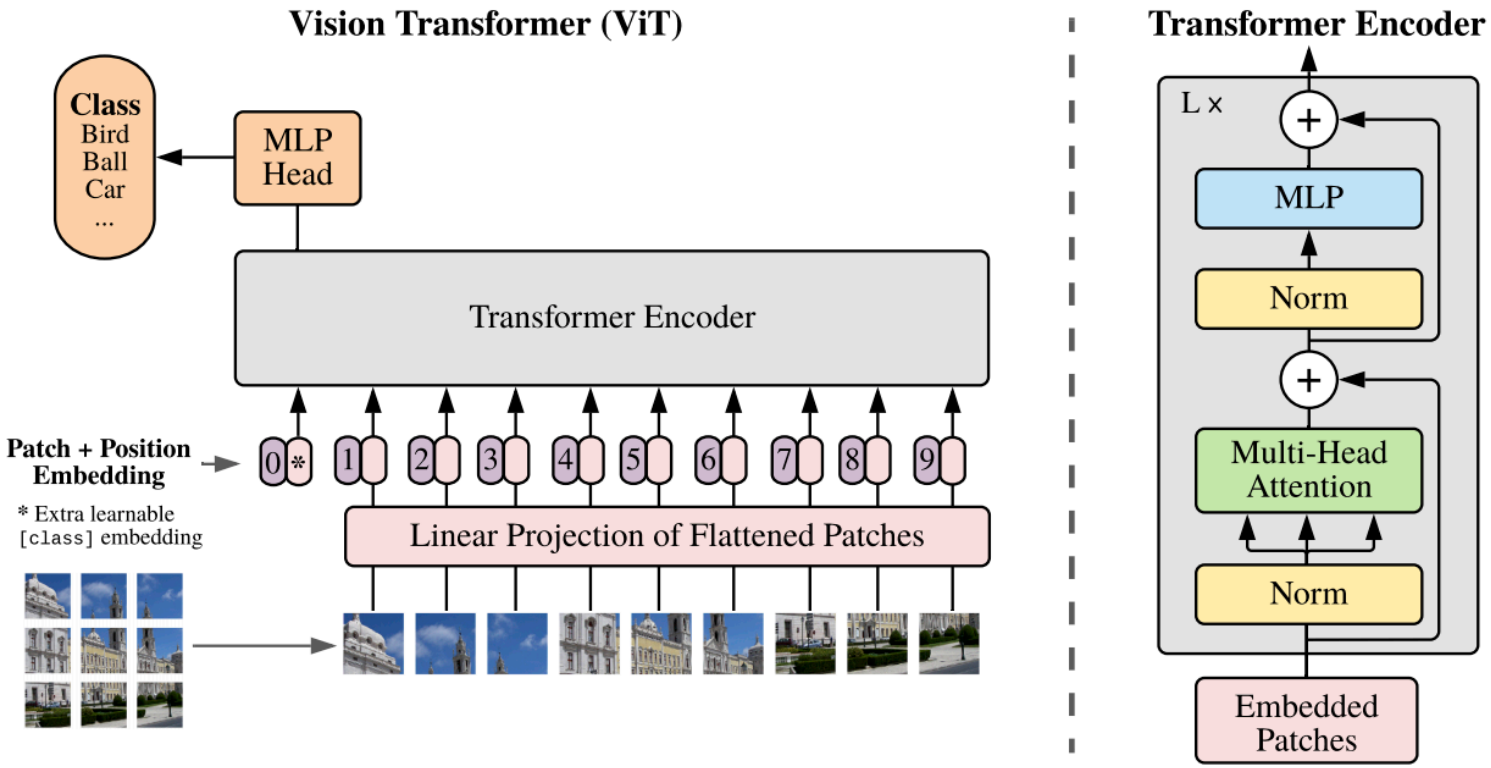


https://openai.com/blog/image-gpt/

M. Chen et al., Generative pretraining from pixels, ICML 2020

Vision transformer (ViT) – Google

- Split an image into patches, feed linearly projected patches into standard transformer encoder
  - With patches of 14x14 pixels, you need 16x16=256 patches to represent 224x224 images



A. Dosovitskiy et al. An image is worth 16x16 words: Transformers for image recognition at scale. ICLR 2021

Self-supervised language modeling with transformers

1. Download A LOT of text from the internet
2. Train a giant transformer using a suitable pretext task
3. Fine-tune the transformer on desired NLP task

Model Alias	Org.	Article Reference
ULMfit	fast.ai	*Universal Language Model Fine-tuning for Text Classification* Howard and Ruder
ELMo	AllenNLP	*Deep contextualized word representations* Peters et al.
OpenAI GPT	OpenAI	*Improving Language Understanding by Generative Pre-Training* Radford et al.
BERT	Google	*BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* Devlin et al.
XLM	Facebook	*Cross-lingual Language Model Pretraining* Lample and Conneau

Image source

Self-Supervised Learning

Why self-supervised learning?

- Creating labeled datasets for each task is an expensive, time-consuming, tedious task
  - Requires hiring human labelers, preparing labeling manuals, creating GUIs, creating storage pipelines, etc.
  - High quality annotations in certain domains can be particularly expensive (e.g., medicine)
- Self-supervised learning takes advantage of the vast amount of unlabeled data on the internet (images, videos, text)
  - Rich discriminative features can be obtained by training models without actual labels
- Self-supervised learning can potentially generalize better because we learn more about the world

Challenges for self-supervised learning

- How to select a suitable pretext task for an application
- There is no gold standard for comparison of learned feature representations
- Selecting a suitable loss functions, since there is no single objective as the test set accuracy in supervised learning

Self-Supervised Learning

Self-supervised learning versus unsupervised learning

- Self-supervised learning (SSL)
  - Aims to extract useful feature representations from raw unlabeled data through ***pretext tasks***
  - Apply the feature representation to improve the performance of ***downstream tasks***
- Unsupervised learning
  - Discover patterns in unlabeled data, e.g., for clustering or dimensionality reduction
- Note also that the term “self-supervised learning” is sometimes used interchangeably with “unsupervised learning”

Self-supervised learning versus transfer learning

- Transfer learning is often implemented in a supervised manner
  - E.g., learn features from a labeled ImageNet, and transfer the features to a smaller dataset
- SSL is a type of transfer learning approach implemented in an unsupervised manner

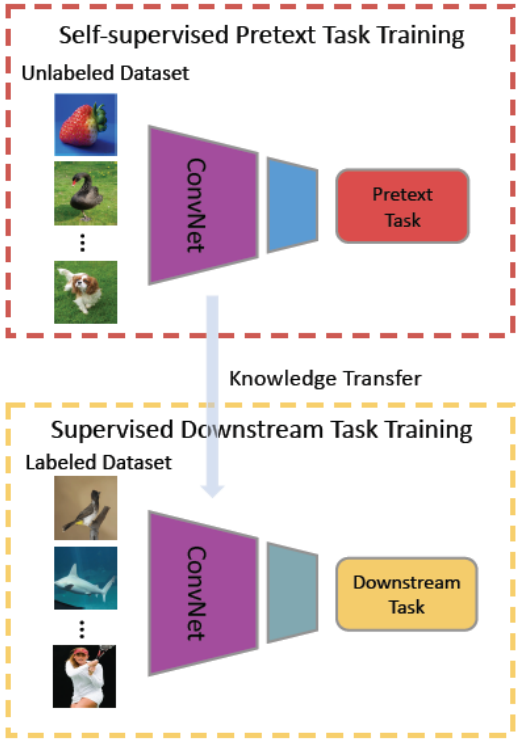
Self-supervised learning versus data augmentation

- Data augmentation is often used as a regularization method in supervised learning
- In SSL, image rotation of shifting are used for feature learning in raw unlabeled data

Self-Supervised Learning

One more depiction of the general pipeline for self-supervised learning is shown in the figure

- For the downstream task, re-use the trained ConvNet base model, and fine-tune the top layers on a small labeled dataset



Picture from: Jing and Tian (2019) Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey

Image Rotation

Geometric transformation recognition

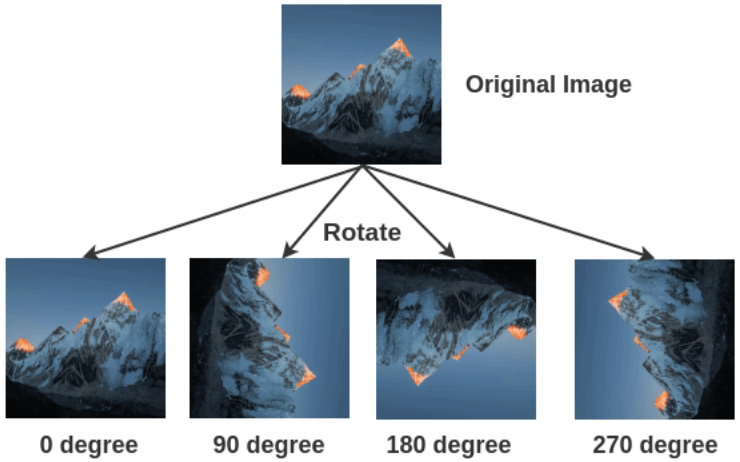
- Gidaris (2018) - Unsupervised Representation Learning by Predicting Image Rotations

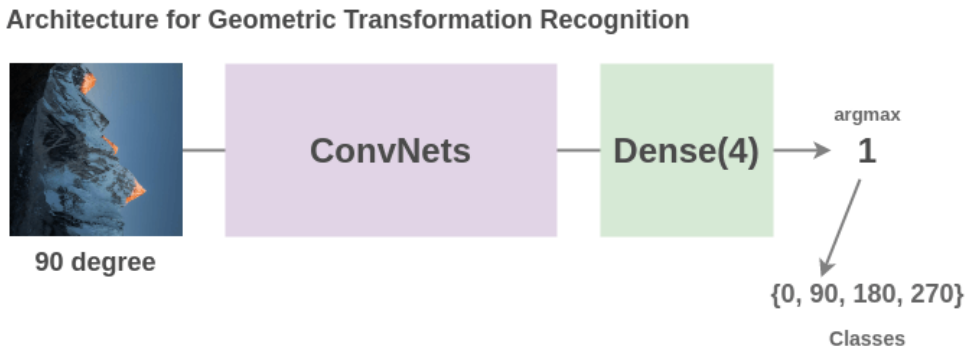
Training data: images rotated by a multiple of 90° at random

- This corresponds to four rotated images at 0°, 90°, 180°, and 270°

Pretext task: train a model to predict the rotation degree that was applied

- Therefore, it is a 4-class classification problem





Picture from: Amit Chaudhary – The Illustrated Self-Supervised Learning

Relative Patch Position

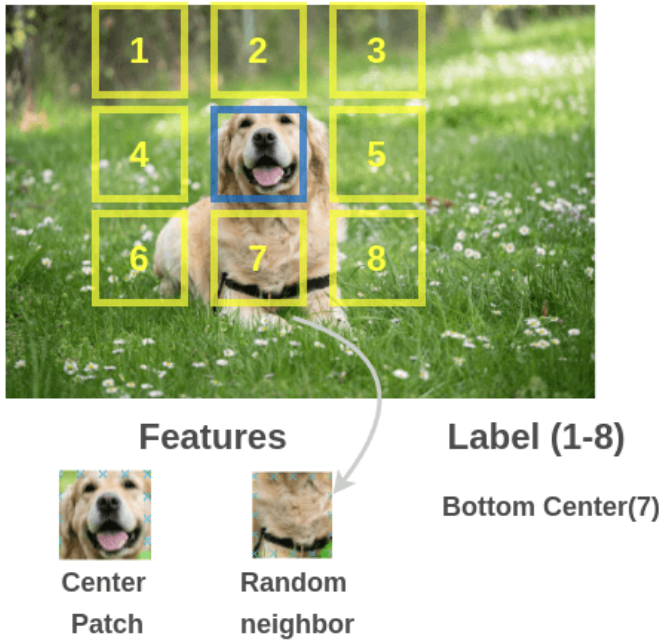
Relative patch position for context prediction

- Dorsch (2015) Unsupervised Visual Representation Learning by Context Prediction

Training data: multiple patches extracted from images

Pretext task: train a model to predict the relationship between the patches

- E.g., predict the relative position of the selected patch below (i.e., position # 7)
  - For the center patch, there are 8 possible neighbor patches (8 possible classes)

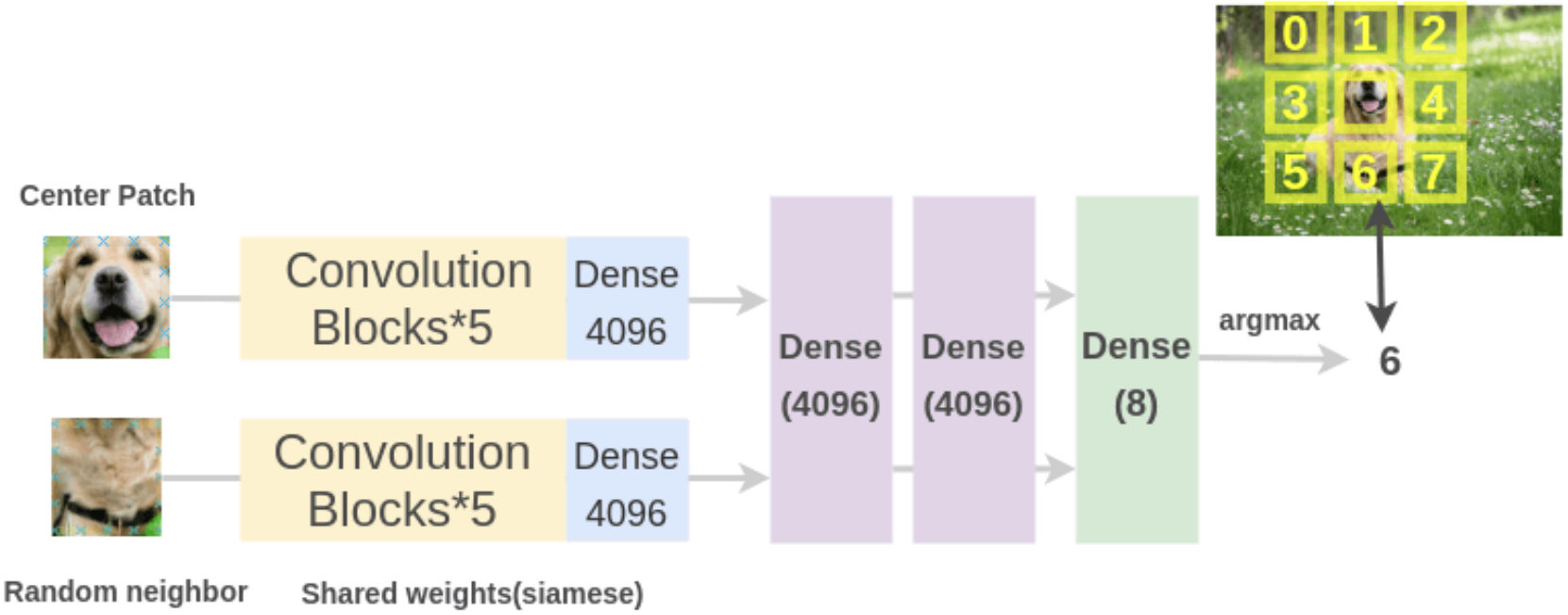


Relative Patch Position

The patches are inputted into two ConvNets with shared weights

- The learned features by the ConvNets are concatenated
- Classification is performed over 8 classes (denoting the 8 possible neighbor positions)

The model needs to understand the spatial context of images, in order to predict the relative positions between the patches



Picture from: Amit Chaudhary – The Illustrated Self-Supervised Learning

Context Encoders

Predict missing pieces, also known as ***context encoders***, or ***inpainting***

- Pathak (2016) Context Encoders: Feature Learning by Inpainting

Training data: remove a random region in images

Pretext task: fill in a missing piece in the image

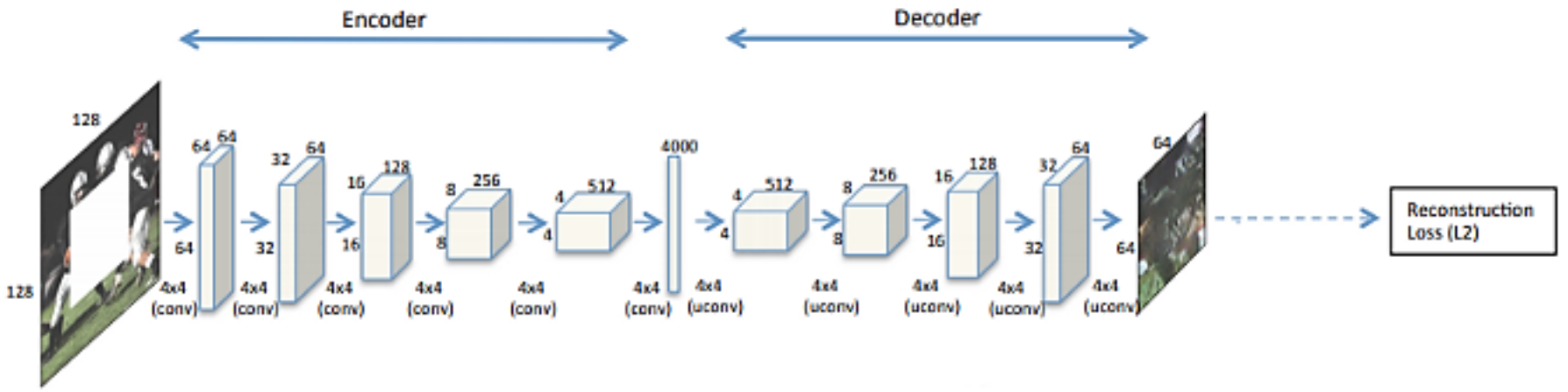
- The model needs to understand the content of the entire image, and produce a plausible replacement for the missing piece

Picture from: Amit Chaudhary – The Illustrated Self-Supervised Learning

Context Encoders

The initially considered model uses an encoder-decoder architecture

- The encoder and decoder have multiple Conv layers, and a shared central fully-connected layer
- The output of the decoder is the reconstructed input image
- A Euclidean ℓ_2 distance is used as the reconstruction loss function $\mathcal{L}_{\text{rec}}$



NLP

Self-supervised learning has driven the recent progress in the ***Natural Language Processing*** (NLP) field

- Models like ELMO, BERT, RoBERTa, ALBERT, Turing NLG, GPT-3 have demonstrated immense potential for automated NLP

Employing various pretext tasks for leaning from raw text produced rich feature representations, useful for different downstream tasks

Pretext tasks in NLP:

- Predict the center word given a window of surrounding words
  - The word highlighted with green color needs to be predicted

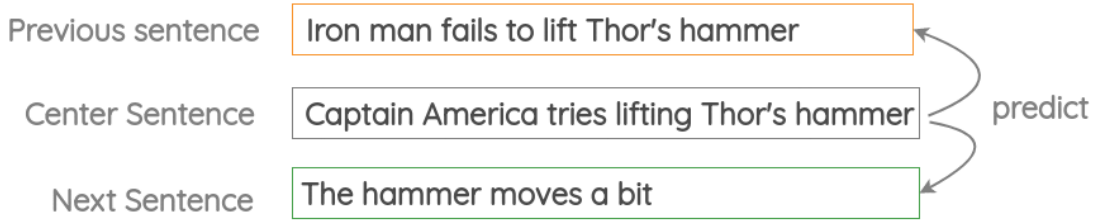
- Predict the surrounding words given the center word

Picture from: Amit Chaudhary – Self-Supervised Representation Learning in NLP

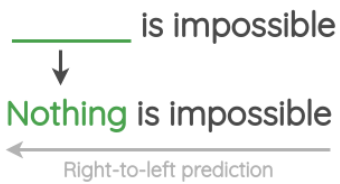
NLP

Pretext tasks in NLP:

- From three consecutive sentences, predict the previous and the next sentence, given the center sentence



- Predict the previous or the next word, given surrounding words



- Predict randomly masked words in sentences

Picture from: Amit Chaudhary – Self-Supervised Representation Learning in NLP

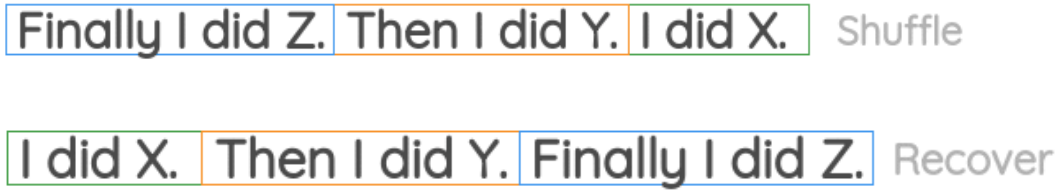
NLP

Pretext tasks in NLP:

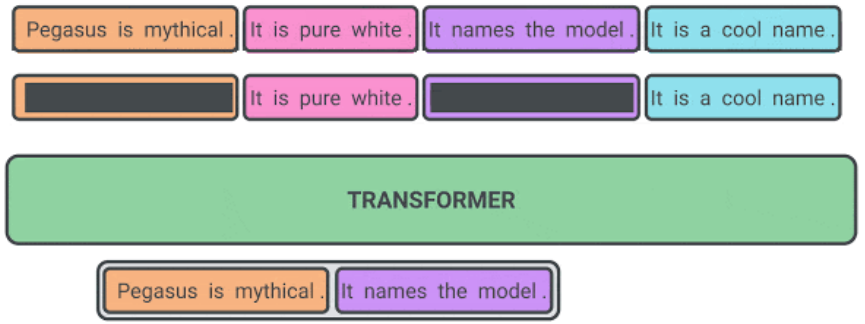
- Predict if the ordering of two sentences is correct

Sentence 1	Sentence 2	Next Sentence
I am going outside	I will be back in the evening	yes
I am going outside	You know nothing John Snow	no

- Predict the order of words in a randomly shuffled sentence



- Predict masked sentences in a document

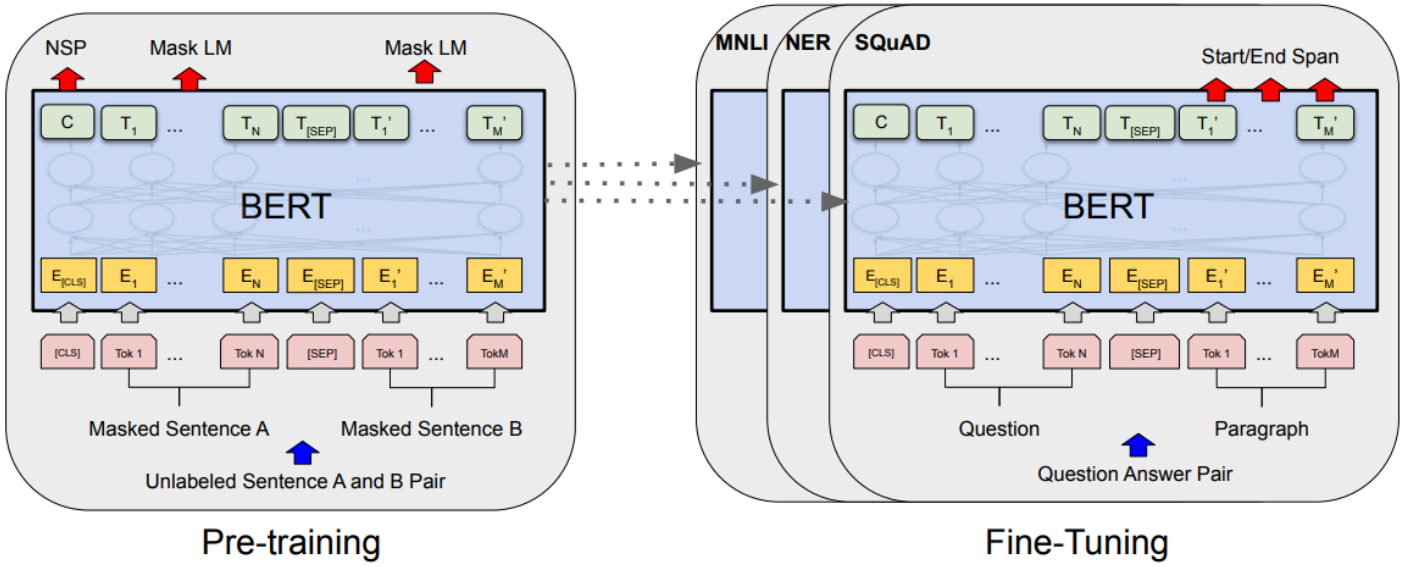


Picture from: Amit Chaudhary – Self-Supervised Representation Learning in NLP

Self-supervised language modeling with transformers

1. Download A LOT of text from the internet
2. Train a giant transformer using a suitable pretext task
3. Fine-tune the transformer on desired NLP task

Bidirectional Encoder Representations from Transformers (BERT)



J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, EMNLP 2018

BERT: Pretext tasks

- Masked language model (MLM)
  - Randomly mask 15% of tokens in input sentences, goal is to reconstruct them using bidirectional context

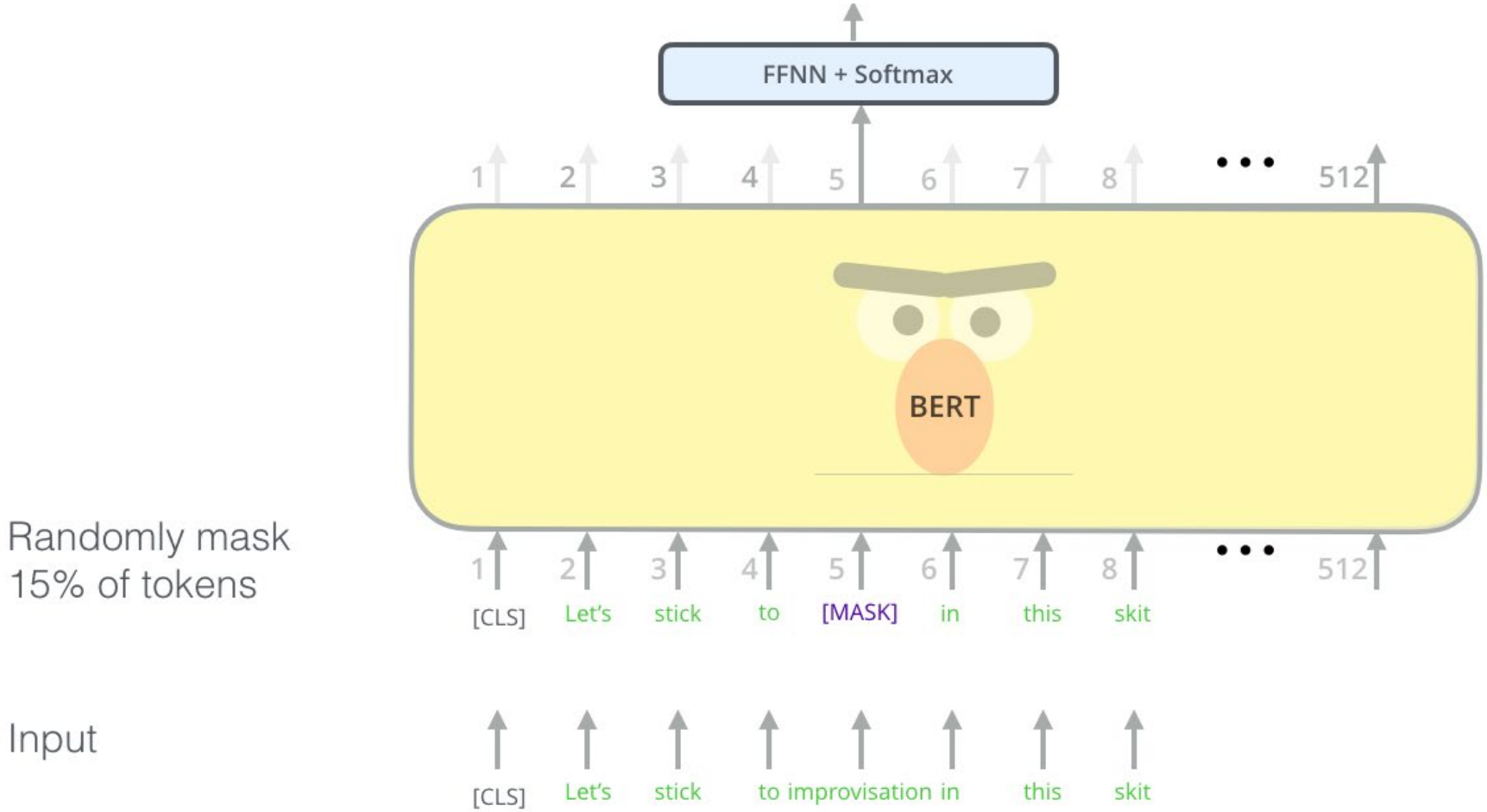
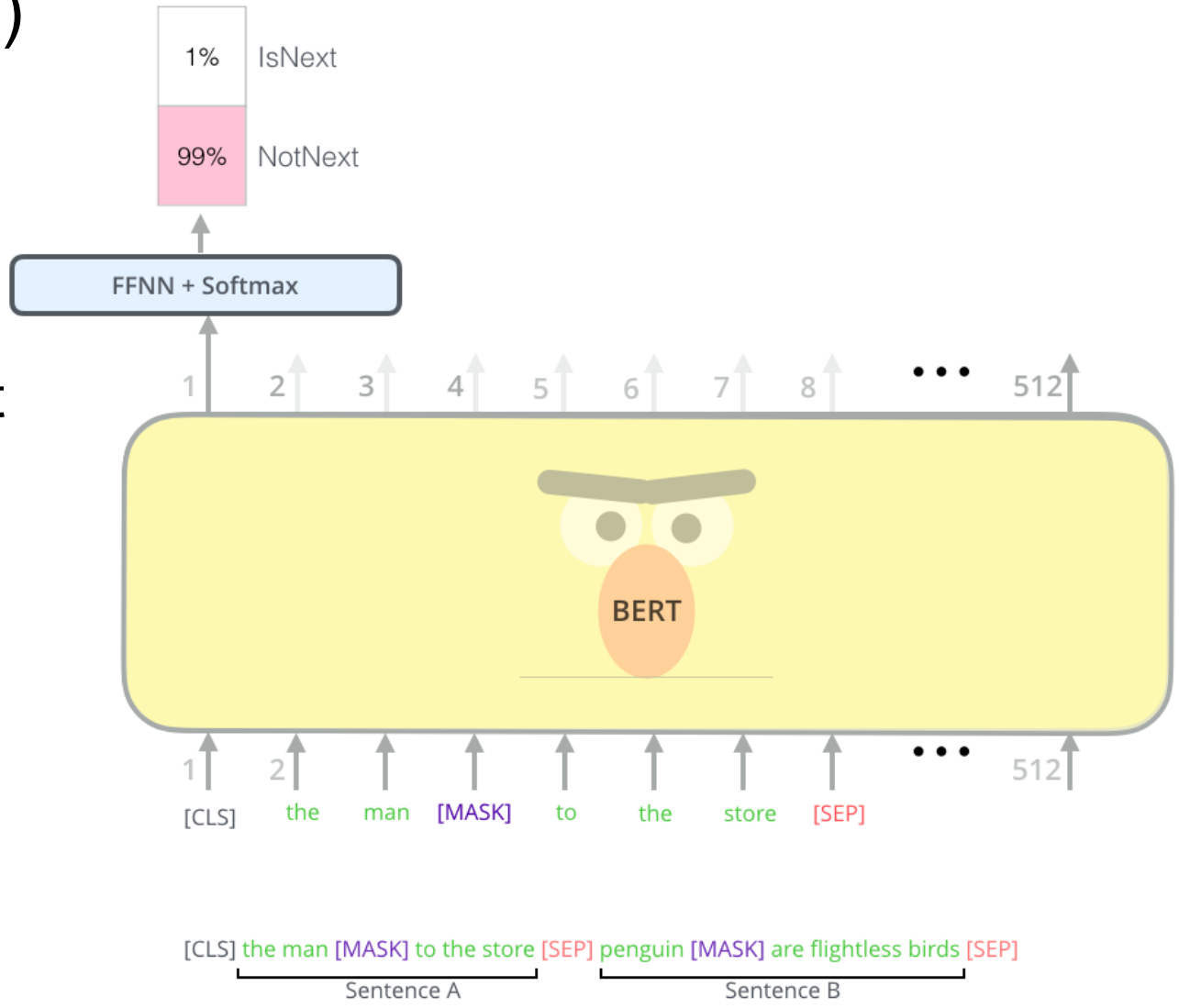


Image source

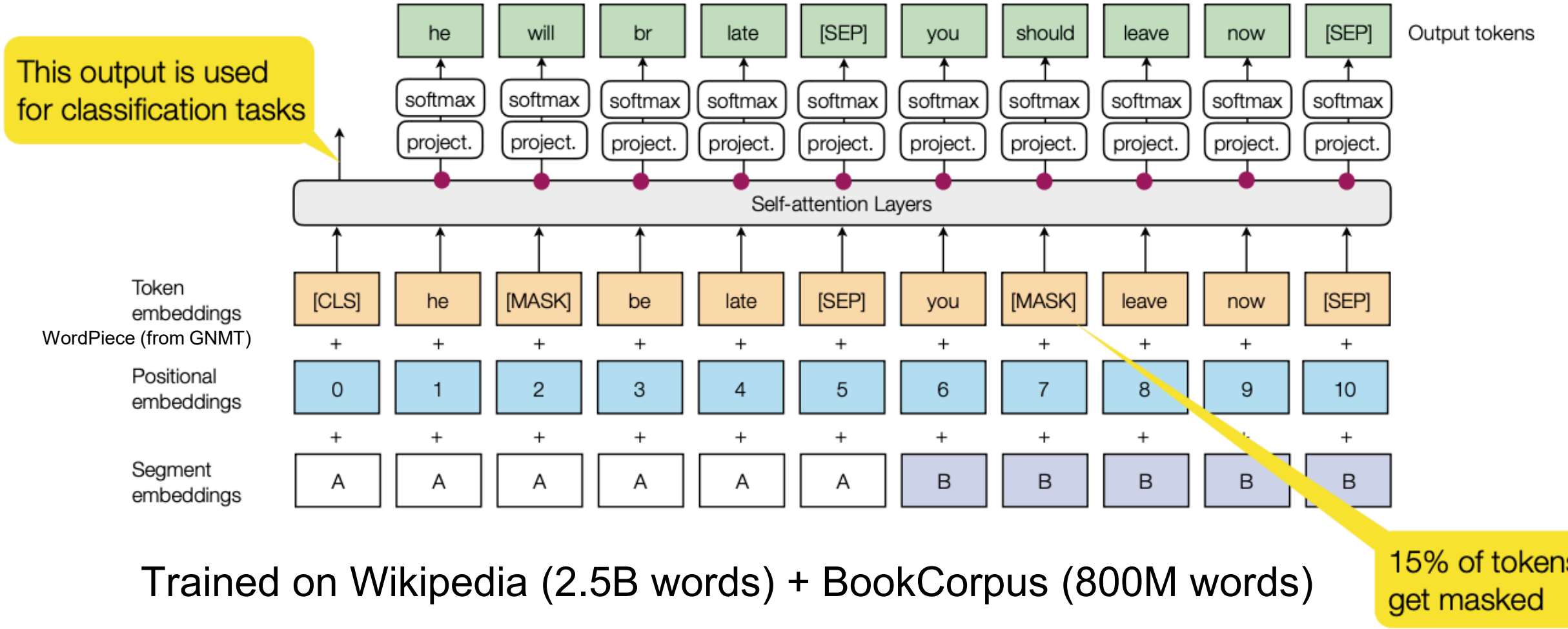
BERT: Pretext tasks

- Next sentence prediction (NSP)
  - Useful for Question Answering and Natural Language Inference tasks
  - In the training data, 50% of the time B is the actual sentence that follows A (labeled as IsNext), and 50% of the time it is a random sentence (labeled as NotNext).



[Image source]

BERT: More detailed view



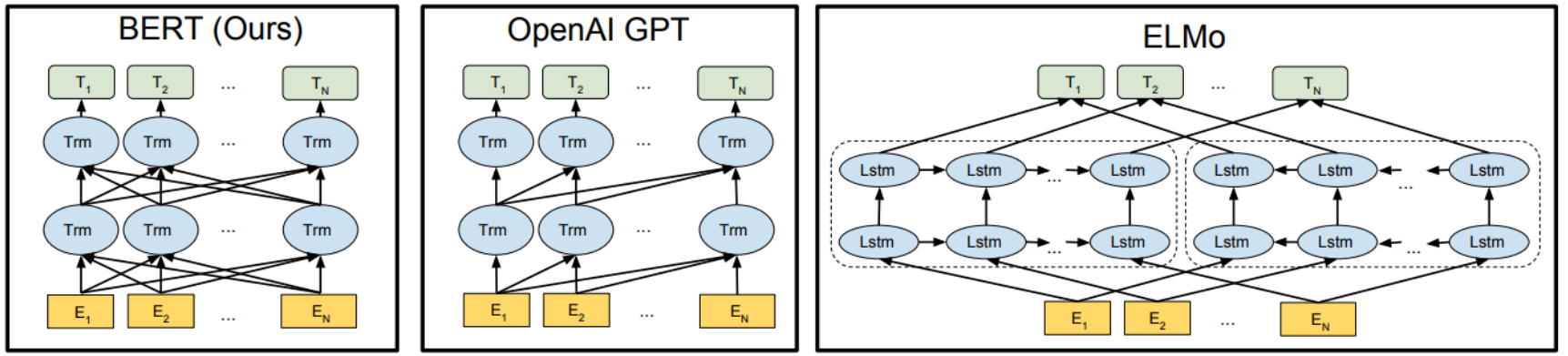
Trained on Wikipedia (2.5B words) + BookCorpus (800M words)

Image source

Other language models

Alias	Model	Token	Tasks	Language
ULMfit	LSTM	word	Causal LM	English
ELMo	LSTM	word	Bidirectional LM	English
OpenAI GPT	Transformer	subword	Causal LM + Classification	English
BERT	Transformer	subword	Masked LM + Next sentence prediction	Multilingual
XLM	Transformer	subword	Causal LM +Masked LM + Translation LM	Multilingual

Image source



Source: J. Johnson

Scaling up transformers

Model	Layers	Hidden dim.	Heads	Params	Data	Training
Transformer-Base	6	512	8	65M		8x P100 (12 hours)
Transformer-Large	12	1024	16	213M		8x P100 (3.5 days)

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

Source: J. Johnson

Scaling up transformers

Model	Layers	Hidden dim.	Heads	Params	Data	Training
Transformer-Base	12	512	8	65M		8x P100 (12 hours)
Transformer-Large	12	1024	16	213M		8x P100 (3.5 days)
BERT-Base	12	768	12	110M	13 GB	4x TPU (4 days)
BERT-Large	24	1024	16	340M	13 GB	16x TPU (4 days)

Devlin et al., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, EMNLP 2018

Source: J. Johnson

Scaling up transformers

Model	Layers	Hidden dim.	Heads	Params	Data	Training
Transformer-Base	12	512	8	65M		8x P100 (12 hours)
Transformer-Large	12	1024	16	213M		8x P100 (3.5 days)
BERT-Base	12	768	12	110M	13 GB	4x TPU (4 days)
BERT-Large	24	1024	16	340M	13 GB	16x TPU (4 days)
XLNet-Large	24	1024	16	~340M	126 GB	512x TPU-v3 (2.5 days)
RoBERTa	24	1024	16	355M	160 GB	1024x V100 GPU (1 day)

Yang et al., XLNet: Generalized Autoregressive Pretraining for Language Understanding, 2019
Liu et al., RoBERTa: A Robustly Optimized BERT Pretraining Approach, 2019

Source: J. Johnson

Scaling up transformers

Model	Layers	Hidden dim.	Heads	Params	Data	Training
Transformer-Base	12	512	8	65M		8x P100 (12 hours)
Transformer-Large	12	1024	16	213M		8x P100 (3.5 days)
BERT-Base	12	768	12	110M	13 GB	4x TPU (4 days)
BERT-Large	24	1024	16	340M	13 GB	16x TPU (4 days)
XLNet-Large	24	1024	16	~340M	126 GB	512x TPU-v3 (2.5 days)
RoBERTa	24	1024	16	355M	160 GB	1024x V100 GPU (1 day)
GPT-2	48	1600	?	1.5B	40 GB	

Radford et al., Language models are unsupervised multitask learners, 2019

Source: J. Johnson

Scaling up transformers

Model	Layers	Hidden dim.	Heads	Params	Data	Training
Transformer-Base	12	512	8	65M		8x P100 (12 hours)
Transformer-Large	12	1024	16	213M		8x P100 (3.5 days)
BERT-Base	12	768	12	110M	13 GB	4x TPU (4 days)
BERT-Large	24	1024	16	340M	13 GB	16x TPU (4 days)
XLNet-Large	24	1024	16	~340M	126 GB	512x TPU-v3 (2.5 days)
RoBERTa	24	1024	16	355M	160 GB	1024x V100 GPU (1 day)
GPT-2	48	1600	?	1.5B	40 GB	
Megatron-LM	72	3072	32	8.3B	174 GB	512x V100 GPU (9 days)

~$430,000 on Amazon AWS!

Shoeybi et al., Megatron-LM: Training Multi-Billion Parameter Languge Models using Model Parallelism, 2019

Source: J. Johnson

Scaling up transformers

Model	Layers	Hidden dim.	Heads	Params	Data	Training
Transformer-Base	12	512	8	65M		8x P100 (12 hours)
Transformer-Large	12	1024	16	213M		8x P100 (3.5 days)
BERT-Base	12	768	12	110M	13 GB	4x TPU (4 days)
BERT-Large	24	1024	16	340M	13 GB	16x TPU (4 days)
XLNet-Large	24	1024	16	~340M	126 GB	512x TPU-v3 (2.5 days)
RoBERTa	24	1024	16	355M	160 GB	1024x V100 GPU (1 day)
GPT-2	48	1600	?	1.5B	40 GB	
Megatron-LM	72	3072	32	8.3B	174 GB	512x V100 GPU (9 days)
Turing-NLG	78	4256	28	17B	?	256x V100 GPU

Microsoft, Turing-NLG: A 17-billion parameter language model by Microsoft, 2020

Source: J. Johnson

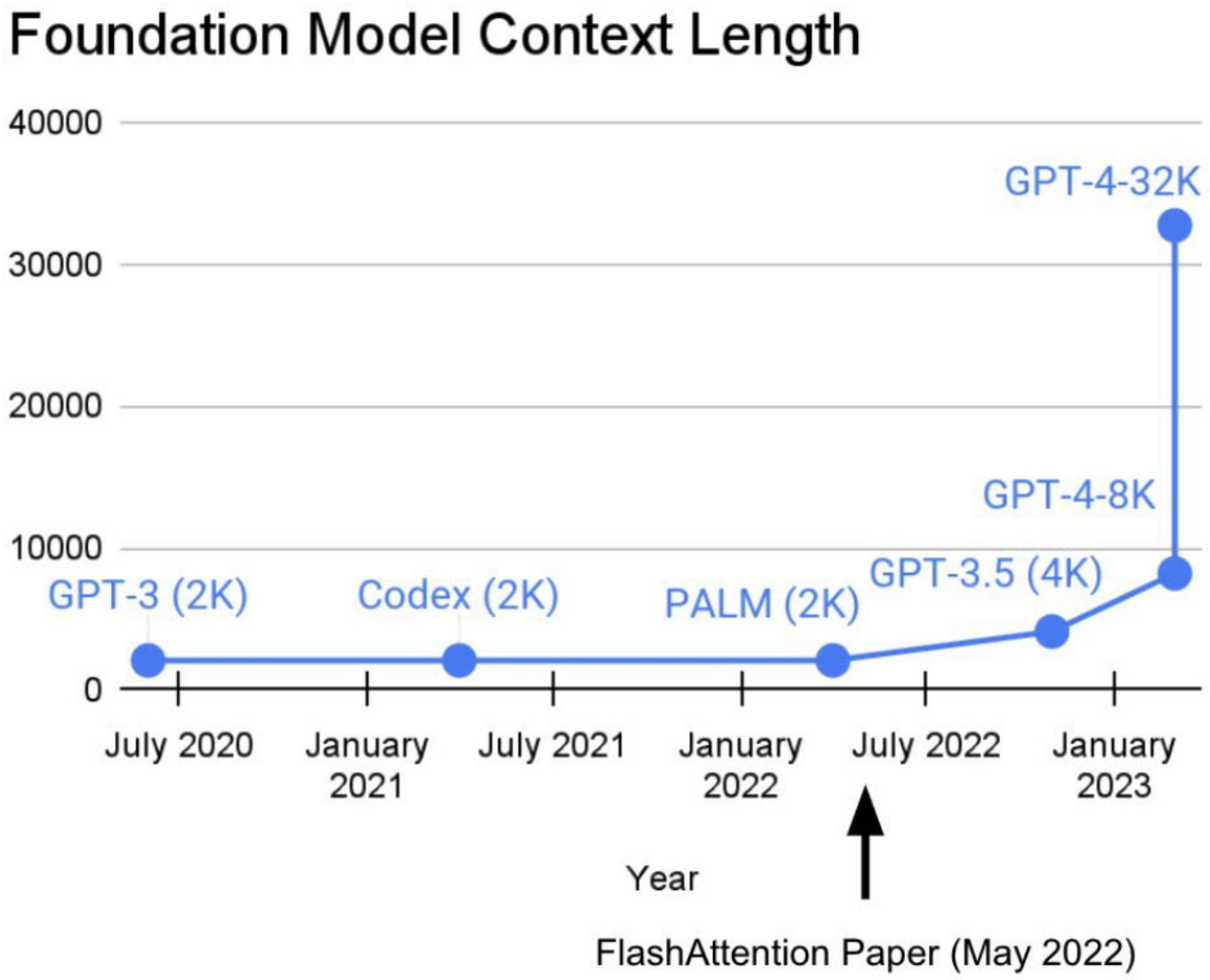
Scaling up transformers

Model	Layers	Hidden dim.	Heads	Params	Data	Training
Transformer-Base	12	512	8	65M		8x P100 (12 hours)
Transformer-Large	12	1024	16	213M		8x P100 (3.5 days)
BERT-Base	12	768	12	110M	13 GB	4x TPU (4 days)
BERT-Large	24	1024	16	340M	13 GB	16x TPU (4 days)
XLNet-Large	24	1024	16	~340M	126 GB	512x TPU-v3 (2.5 days)
RoBERTa	24	1024	16	355M	160 GB	1024x V100 GPU (1 day)
GPT-2	48	1600	?	1.5B	40 GB	
Megatron-LM	72	3072	32	8.3B	174 GB	512x V100 GPU (9 days)
Turing-NLG	78	4256	28	17B	?	256x V100 GPU
GPT-3	96	12288	96	175B	694GB	?

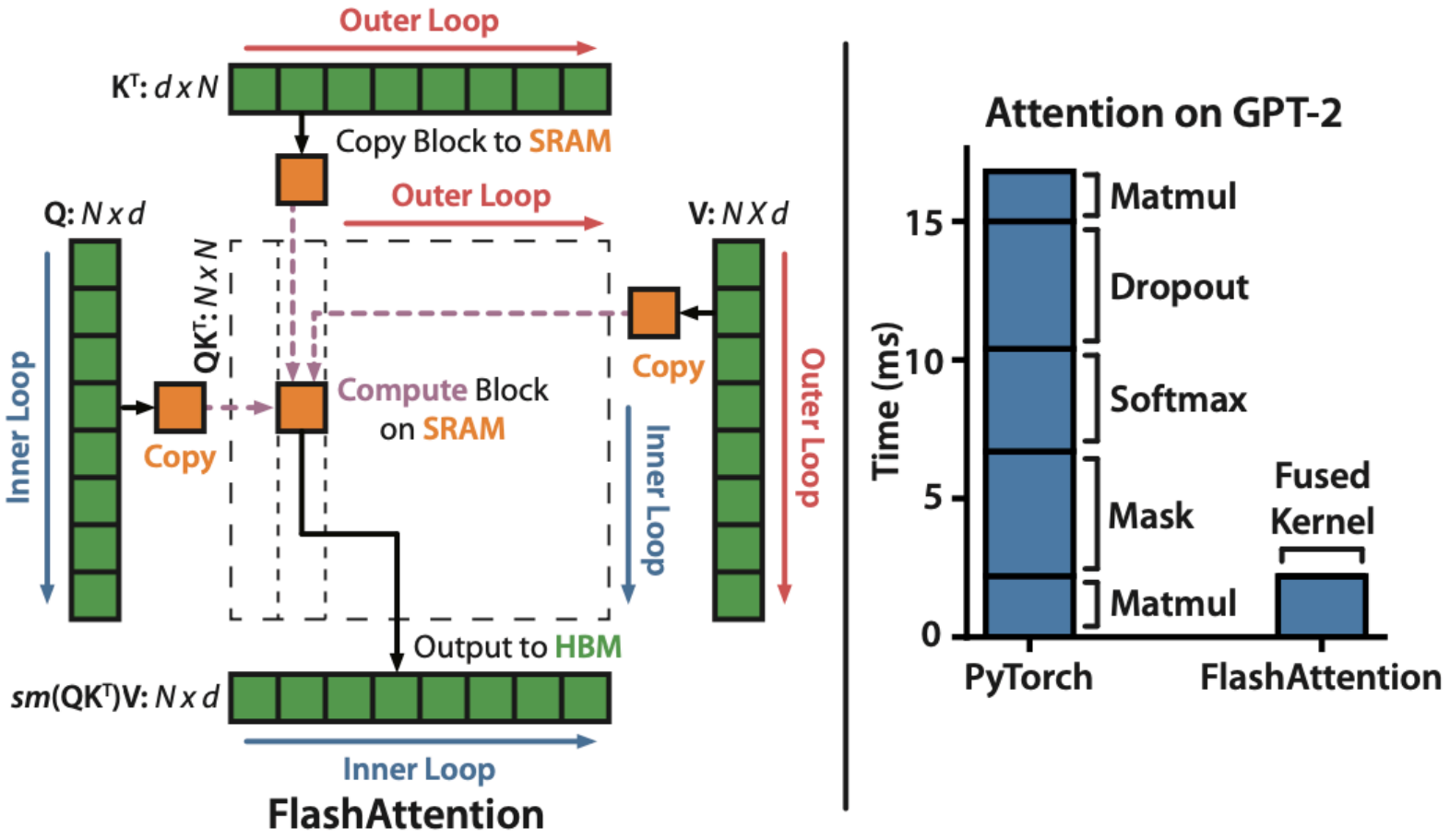
~$4.6M, 355 GPU-years (source)

Brown et al., Language Models are Few-Shot Learners, NeurIPS 2020 (Best Paper Award)

Scaling Up!


Foundation Model Context Length
40000
30000
20000
10000
0
GPT-4-32K
GPT-4-8K
GPT-3.5 (4K)
GPT-3 (2K)
Codex (2K)
PALM (2K)
July 2020
January 2021
July 2021
January 2022
July 2022
January 2023
Year
FlashAttention Paper (May 2022)

Scaling Laws and Improving Efficiency



FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness (2022), by Dao, Fu, Ermon, Rudra, and Ré, https://arxiv.org/abs/2205.14135.

Human Feedback to LLMs

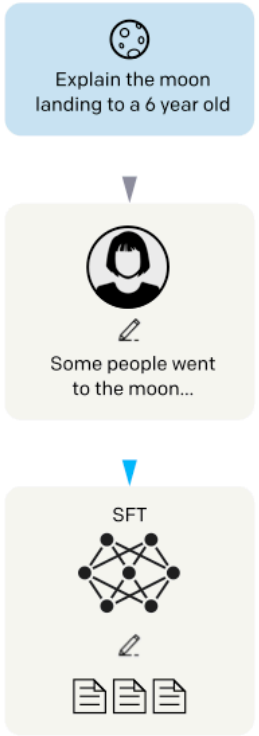
Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.



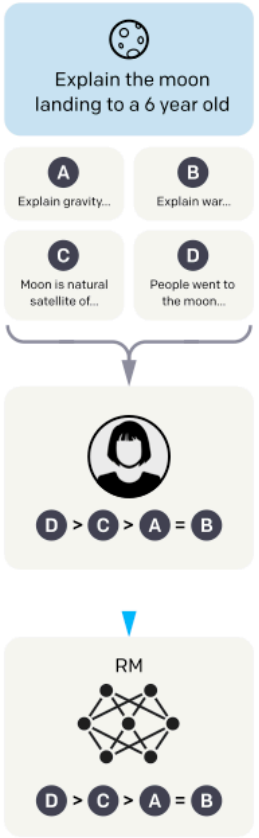
Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

Optimize a policy against the reward model using reinforcement learning.

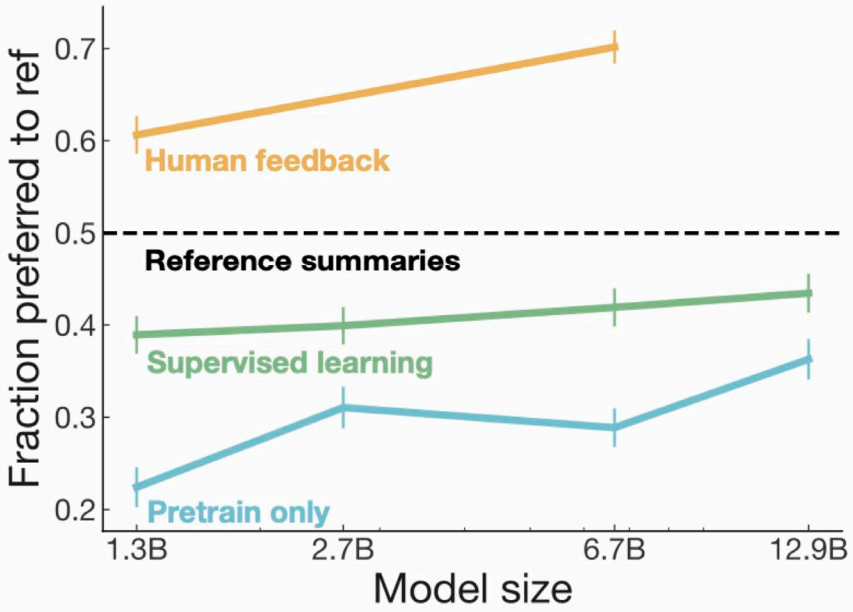
In this so-called InstructGPT paper, the researchers use a reinforcement learning mechanism with humans in the loop (RLHF). They start with a pretrained GPT-3 base model and fine-tune it further using supervised learning on prompt-response pairs generated by humans (Step 1).

Next, they ask humans to rank model outputs to train a reward model (step 2).

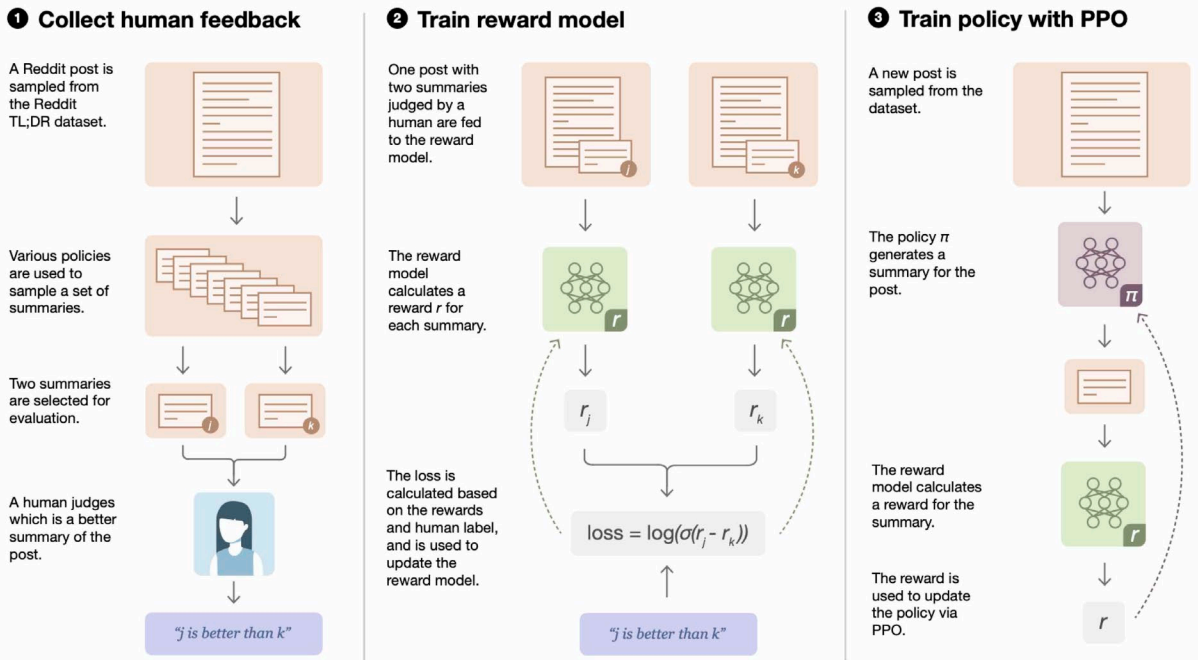
Finally, they use the reward model to update the pretrained and fine-tuned GPT-3 model using reinforcement learning via proximal policy optimization (step 3).

Reinforcement Learning with Human Feedback (RLHF)

RL with PPO results in “better” LLMs
than using regular supervised learning

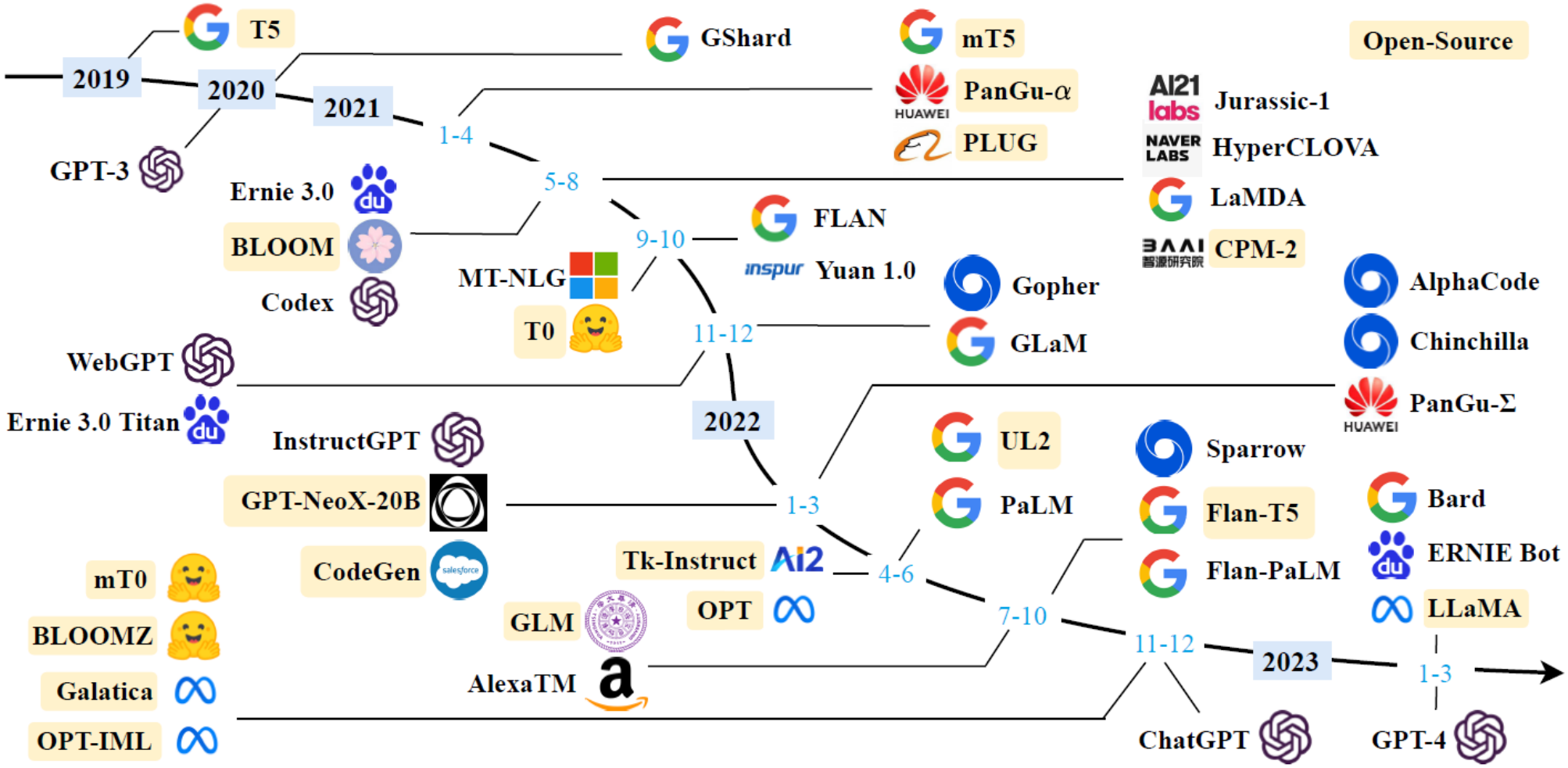


The original RLHF method for summarization
(“Learning to Summarize from Human Feedback")



Training Language Models to Follow Instructions with Human Feedback (2022) by Ouyang, Wu, Jiang, Almeida, Wainwright, Mishkin, Zhang, Agarwal, Slama, Ray, Schulman, Hilton, Kelton, Miller, Simens, Askell, Welinder, Christiano, Leike, and Lowe (https://arxiv.org/abs/2203.02155)

A timeline of existing large language models (larger than 10B)



A Survey of Large Language Models. https://arxiv.org/abs/2303.18223

	Model	Release Time	Size (B)	Base Model	Adaptation IT	Adaptation RLHF	Pre-train Data Scale	Latest Data Timestamp	Hardware (GPUs / TPUs)	Training Time	Evaluation ICL	Evaluation CoT
Open Source	T5 [71]	Oct-2019	11	-	-	-	1T tokens	Apr-2019	1024 TPU v3	-	✓	-
	mT5 [72]	Mar-2021	13	-	-	-	1T tokens	Apr-2019	-	-	✓	-
	PanGu-α [73]	May-2021	13*	-	-	-	1.1TB	-	2048 Ascend 910	-	✓	-
	CPM-2 [74]	May-2021	198	-	-	-	2.6TB	-	-	-	-	-
	T0 [28]	Oct-2021	11	T5	✓	-	-	-	512 TPU v3	27 h	✓	-
	GPT-NeoX-20B [75]	Feb-2022	20	-	-	-	825GB	Dec-2022	96 40G A100	-	✓	-
	CodeGen [76]	Mar-2022	16	-	-	-	577B tokens	-	-	-	✓	-
	Tk-Instruct [77]	Apr-2022	11	T5	✓	-	-	-	256 TPU v3	4 h	✓	-
	UL2 [78]	Apr-2022	20	-	✓	-	1T tokens	Apr-2019	512 TPU v4	-	✓	✓
	OPT [79]	May-2022	175	-	-	-	180B tokens	-	992 80G A100	-	✓	-
	BLOOM [66]	Jul-2022	176	-	-	-	366B	-	384 80G A100	105 d	✓	-
	GLM [80]	Aug-2022	130	-	-	-	400B tokens	-	768 40G A100	60 d	✓	-
	Flan-T5 [81]	Oct-2022	11	T5	✓	-	-	-	-	-	✓	✓
	mT0 [82]	Nov-2022	13	mT5	✓	-	-	-	-	-	✓	-
	Galactica [35]	Nov-2022	120	-	-	-	106B tokens	-	-	-	✓	✓
	BLOOMZ [82]	Nov-2022	176	BLOOM	✓	-	-	-	-	-	✓	-
	OPT-IML [83]	Dec-2022	175	OPT	✓	-	-	-	128 40G A100	-	✓	✓
	LLaMA [57]	Feb-2023	65	-	-	-	1.4T tokens	-	2048 80G A100	21 d	✓	-
Closed Source	GShard [84]	Jan-2020	600	-	-	-	1T tokens	-	2048 TPU v3	4 d	-	-
	GPT-3 [55]	May-2020	175	-	-	-	300B tokens	-	-	-	✓	-
	LaMDA [85]	May-2021	137	-	-	-	2.81T tokens	-	1024 TPU v3	57.7 d	-	-
	HyperCLOVA [86]	Jun-2021	82	-	-	-	300B tokens	-	1024 A100	13.4 d	✓	-
	Codex [87]	Jul-2021	12	GPT-3	-	-	100B tokens	May-2020	-	-	✓	-
	ERNIE 3.0 [88]	Jul-2021	10	-	-	-	375B tokens	-	384 V100	-	✓	-
	Jurassic-1 [89]	Aug-2021	178	-	-	-	300B tokens	-	800 GPU	-	✓	-
	FLAN [62]	Oct-2021	137	LaMDA	✓	-	-	-	128 TPU v3	60 h	✓	-
	MT-NLG [90]	Oct-2021	530	-	-	-	270B tokens	-	4480 80G A100	-	✓	-
	Yuan 1.0 [91]	Oct-2021	245	-	-	-	180B tokens	-	2128 GPU	-	✓	-
	WebGPT [70]	Dec-2021	175	GPT-3	-	✓	-	-	-	-	✓	-
	Gopher [59]	Dec-2021	280	-	-	-	300B tokens	-	4096 TPU v3	920 h	✓	-
	ERNIE 3.0 Titan [92]	Dec-2021	260	-	-	-	300B tokens	-	2048 V100	28 d	✓	-
	GLaM [93]	Dec-2021	1200	-	-	-	280B tokens	-	1024 TPU v4	574 h	✓	-
	InstructGPT [61]	Jan-2022	175	GPT-3	✓	✓	-	-	-	-	✓	-
	AlphaCode [94]	Feb-2022	41	-	-	-	967B tokens	Jul-2021	-	-	-	-
	Chinchilla [34]	Mar-2022	70	-	-	-	1.4T tokens	-	-	-	✓	-
	PaLM [56]	Apr-2022	540	-	-	-	780B tokens	-	6144 TPU v4	-	✓	✓
	AlexaTM [95]	Aug-2022	20	-	-	-	1.3T tokens	-	128 A100	120 d	✓	✓
	Sparrow [96]	Sep-2022	70	-	-	✓	-	-	64 TPU v3	-	✓	-
	U-PaLM [97]	Oct-2022	540	PaLM	-	-	-	-	512 TPU v4	5 d	✓	✓
	Flan-PaLM [81]	Oct-2022	540	PaLM	✓	-	-	-	512 TPU v4	37 h	✓	✓
	Flan-U-PaLM [81]	Oct-2022	540	U-PaLM	✓	-	-	-	-	-	✓	✓
	GPT-4 [46]	Mar-2023	-	-	✓	✓	-	-	-	-	✓	✓
	PanGu-Σ [98]	Mar-2023	1085	PanGu-α	-	-	329B tokens	-	512 Ascend 910	100 d	✓	-

Should we be scared of GPT-3?

Opinion

The New York Times

How Do You Know a Human Wrote This?

Machines are gaining the ability to write, and they are getting terrifyingly good at it.

By Farhad Manjoo
Opinion Columnist

July 29, 2020

https://www.nytimes.com/2020/07/29/opinion/gpt-3-ai-automation.html

See also:
https://www.nytimes.com/2020/11/24/science/artificial-intelligence-ai-gpt3.html

MIT Technology Review

Opinion

GPT-3, Bloviator: OpenAI's language generator has no idea what it's talking about

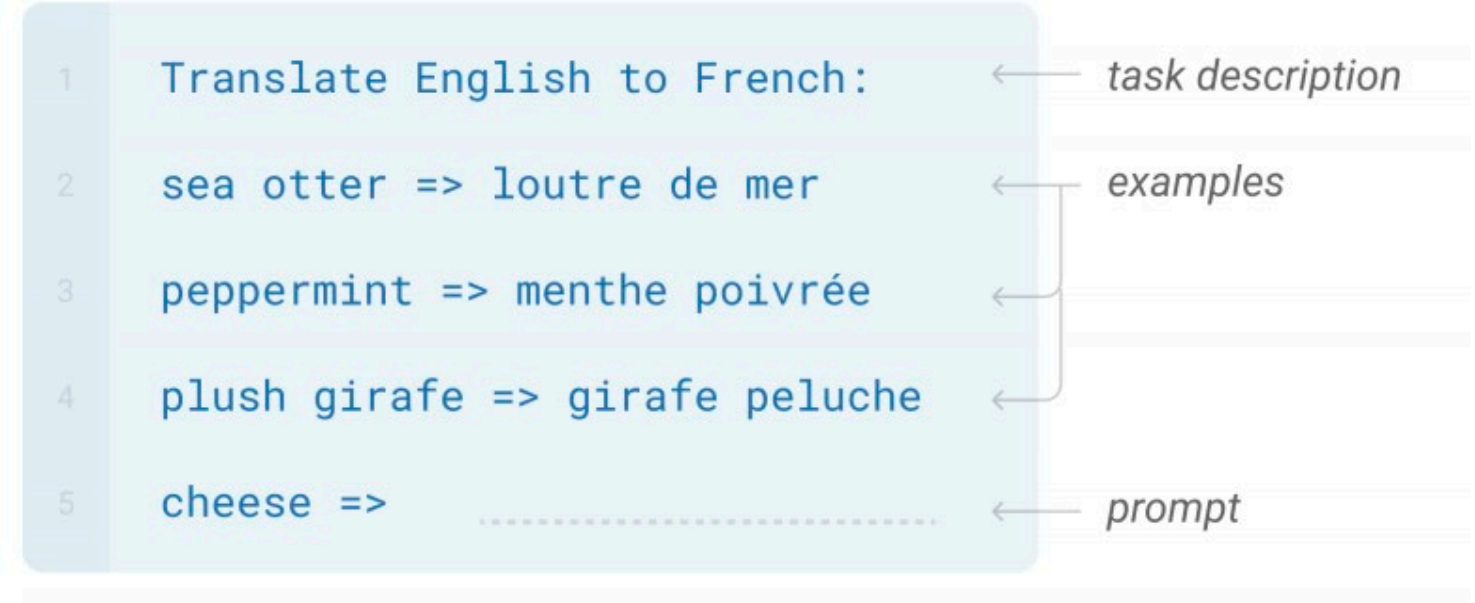
Tests show that the popular AI still has a poor grasp of reality.

by **Gary Marcus** and **Ernest Davis** August 22, 2020

https://www.technologyreview.com/2020/08/22/1007539/gpt3-openai-language-generator-artificial-intelligence-ai-opinion/

GPT-3

- Key idea: if the model and training datasets are big enough, model can adapt to new tasks *without fine-tuning*
- **Few-shot learning:** In addition to the task description, the model sees a few examples of the task



T. Brown et al., Language models are few-shot learners, NeurIPS 2020

GPT-3

- Key idea: if the model and training datasets are big enough, model can adapt to new tasks *without fine-tuning*
- **One-shot learning:** In addition to the task description, the model sees a *single example* of the task

T. Brown et al., Language models are few-shot learners, NeurIPS 2020

GPT-3

- Key idea: if the model and training datasets are big enough, model can adapt to new tasks *without fine-tuning*
- **Zero-shot learning:** The model sees the task description and *no* training examples

T. Brown et al., Language models are few-shot learners, NeurIPS 2020

Task: Generate news article

Title: United Methodists Agree to Historic Split
Subtitle: Those who oppose gay marriage will form their own denomination
Article: **After two days of intense debate, the United Methodist Church has agreed to a historic split - one that is expected to end in the creation of a new denomination, one that will be "theologically and socially conservative," according to The Washington Post. The majority of delegates attending the church's annual General Conference in May voted to strengthen a ban on the ordination of LGBTQ clergy and to write new rules that will "discipline" clergy who officiate at same-sex weddings. But those who opposed these measures have a new plan: They say they will form a separate denomination by 2020, calling their church the Christian Methodist denomination.**
The Post notes that the denomination, which claims 12.5 million members, was in the early 20th century the "largest Protestant denomination in the U.S.," but that it has been shrinking in recent decades. The new split will be the second in the church's history. The first occurred in 1968, when roughly 10 percent of the denomination left to form the Evangelical United Brethren Church. The Post notes that the proposed split "comes at a critical time for the church, which has been losing members for years," which has been "pushed toward the brink of a schism over the role of LGBTQ people in the church." Gay marriage is not the only issue that has divided the church. In 2016, the denomination was split over ordination of transgender clergy, with the North Pacific regional conference voting to ban them from serving as clergy, and the South Pacific regional conference voting to allow them.

Gray: human prompts, boldface: GPT-3 completions

(Three articles provided as training examples)

Figure 3.14: The GPT-3 generated news article that humans had the greatest difficulty distinguishing from a human written article (accuracy: 12%).

Task: Use new word in sentence

A "whatpu" is a small, furry animal native to Tanzania. An example of a sentence that uses the word whatpu is:
We were traveling in Africa and we saw these very cute whatpus.

To do a "farduddle" means to jump up and down really fast. An example of a sentence that uses the word farduddle is:
One day when I was playing tag with my little sister, she got really excited and she started doing these crazy farduddles.
A "yalubalu" is a type of vegetable that looks like a big pumpkin. An example of a sentence that uses the word yalubalu is:
I was on a trip to Africa and I tried this yalubalu vegetable that was grown in a garden there. It was delicious.

A "Burringo" is a car with very fast acceleration. An example of a sentence that uses the word Burringo is:
In our garage we have a Burringo that my father drives to work every day.

A "Gigamuru" is a type of Japanese musical instrument. An example of a sentence that uses the word Gigamuru is:
I have a Gigamuru that my uncle gave me as a gift. I love to play it at home.

To "screeg" something is to swing a sword at it. An example of a sentence that uses the word screeg is:
We screeghed at each other for several minutes and then we went outside and ate ice cream.

Gray: human prompts, boldface: GPT-3 completions

Task: Correct grammar

Poor English input: I eated the purple berries.
Good English output: I ate the purple berries.
Poor English input: Thank you for picking me as your designer. I'd appreciate it.
Good English output: Thank you for choosing me as your designer. I appreciate it.
Poor English input: The mentioned changes have done. or I did the alteration that you requested. or I changed things you wanted and did the modifications.
Good English output: The requested changes have been made. or I made the alteration that you requested. or I changed things you wanted and made the modifications.
Poor English input: I'd be more than happy to work with you in another project.
Good English output: I'd be more than happy to work with you on another project.

Poor English input: Please provide me with a short brief of the design you're looking for and that'd be nice if you could share some examples or project you did before.
Good English output: Please provide me with a brief description of the design you're looking for and that would be nice if you could share some examples or projects you have done before.

Poor English input: The patient was died.
Good English output: The patient died.

Poor English input: We think that Leslie likes ourselves.
Good English output: We think that Leslie likes us.

Poor English input: Janet broke Bill on the finger.
Good English output: Janet broke Bill's finger.

Poor English input: Mary arranged for, in St. Louis, John to rent a house cheap.
Good English output: Mary arranged for John to rent a house in St. Louis.

Gray: human prompts, boldface: GPT-3 completions

Task: Generate poems

Context →	The City BY C. P. CAVAFY TRANSLATED BY EDMUND KEELEY [Poem text omitted] SOME TREES John Ashbery [Poem text omitted] Shadows on the Way Wallace Stevens

Task: Generate poems

-------- Generated Poem 1 --------

I must have shadows on the way
If I am to walk I must have
Each step taken slowly and alone
To have it ready made

And I must think in lines of grey
To have dim thoughts to be my guide
Must look on blue and green
And never let my eye forget
That color is my friend
And purple must surround me too

The yellow of the sun is no more
Intrusive than the bluish snow
That falls on all of us. I must have
Grey thoughts and blue thoughts walk with me
If I am to go away at all.

-------- Generated Poem 4 --------

Nobody will come to this place. It is a road that leads nowhere.
The solitude is deep. The mountains are high.
But they are desolate, and they turn the traveler's face
Towards the North. All the sounds of the world are far away.
When the wind rises above the trees,
The boughs bow to the ground.
Even the birds that inhabit the tangle of weeds
That is the roadside cover, are silent. One listens,
But hears no roar of the forest. One is alone.
One will be taken.
One will be taken.
There is no utterance, there is no conversation,
But one is uneasy all the same....
There is a thin blue mist,
A darkness rising like smoke,
And within that darkness
A possession of the heart.
One will be taken.... It was here, and it will be here again-
Here, under this sky empty and full of light.

GPT-3 creative fiction

Topic: humor: literary parodies and homages, pastiches, style parodies.

Parodies of the fantasy novel series *Harry Potter* in the style of various famous authors:

By Ernest Hemingway:

"It was a cold day on Privet Drive. A child cried. Harry felt nothing. He was dryer than dust. He had been silent too long. He had not felt love. He had scarcely felt hate. Yet the Dementor's Kiss killed nothing. Death didn't leave him less dead than he had been a second before. It wasn't about living or dying really. It wasn't about death. It was about the Dark Lord, his creator, his engenderer. He was not resurrected. He was created again. He was whole again in the courtyard of Malfoy Manor."[33]

By Jane Austen:

"It is a truth universally acknowledged, that a broken Harry is in want of a book. This he knows to his cost; pressing that wretched nose against the window of a bookstore in Diagon Alley the last thing before he goes into hiding for several years, where upon, he goes straight to Azkaban."

By Arthur Conan Doyle:

"Harry pushed at the swinging doors of the bookshop hard, and nearly knocked himself unconscious. He staggered in with his ungainly package, his cheeks scarlet with cold and the shame of having chosen the wrong month to go Christmas shopping. The proprietor of the store, however, didn't cast even a cursory glance at him, being far more interested in an enormous hunk of wood lying in the middle of the floor, which certainly looked like a gallows. Yes, the proprietor said to a reedy old man wearing a bowler hat and a forlorn expression that made Harry want to kick him, I can rent you such a gallows for a small fee."

https://www.gwern.net/GPT-3

For much, much more, see: https://github.com/elyase/awesome-gpt3