CS480/680: Introduction to Machine Learning

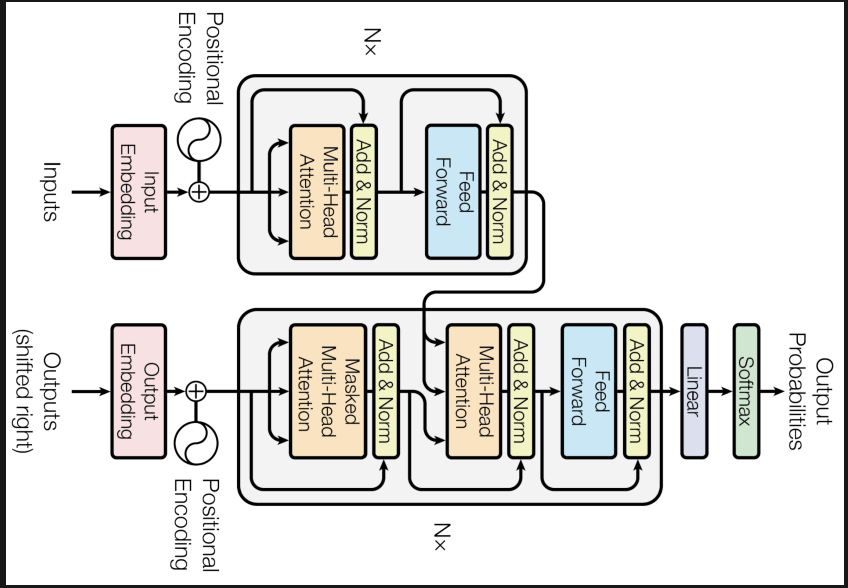
Lec 16: Attention

Yaoliang Yu

UNIVERSITY OF WATERLOO
FACULTY OF MATHEMATICS
DAVID R. CHERITON SCHOOL OF COMPUTER SCIENCE

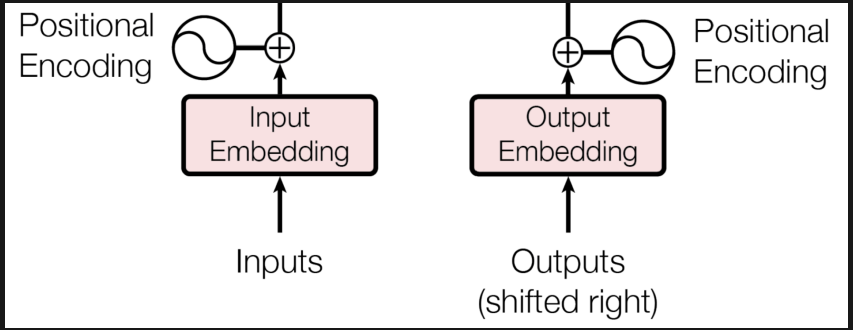
March 11, 2025


THEIR WAR. OUR WORLD.
TRANSFORMERS
27.07.07
TRANSFORMERS
REVENGE OF THE FALLEN
6.24.09
TRANSFORMERS
DARK OF THE MOON
IN THEATRES, RGⓇ AND IMAX 3D
JULY 1
THIS IS NOT WAR.
IT'S EXTINCTION.
TRANSFORMERS
AGE OF EXTINCTION
IN THEATRES, RGⓇ AND IMAX 3D
JUNE 27
TWO WORLDS COLLIDE. ONE SURVIVES.
TRANSFORMERS
THE LAST KNIGHT
FILMED WITH IMAX 3D CAMERAS
IN REAL 3D AND IMAX 3D
JUNE 21



A. Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30*. 2017, pp. 5998–6008.

Input and Output



- Input sequence $X = (\mathbf{x}_1, \ldots, \mathbf{x}_m)^\top$, $\mathbf{x}_t \in \mathbb{R}^p$ one-hot
- Output sequence $Y = (\mathbf{y}_1, \ldots, \mathbf{y}_l)^\top$, $\mathbf{y}_t \in \mathbb{R}^p$ one-hot
- Embedding: XW^e and YW^e, where $W^e \in \mathbb{R}^{p \times d}$, d = 512
- (Additive) positional encoding: $W^p \in \mathbb{R}^{\text{len} \times d}$
  - $w^p_{t,2i} = \sin(t/10000^{2i/d}),\ w^p_{t,2i+1} = \cos(t/10000^{2i/d}),\ i = 0, \ldots, \frac{d}{2} - 1$

Self-Attention

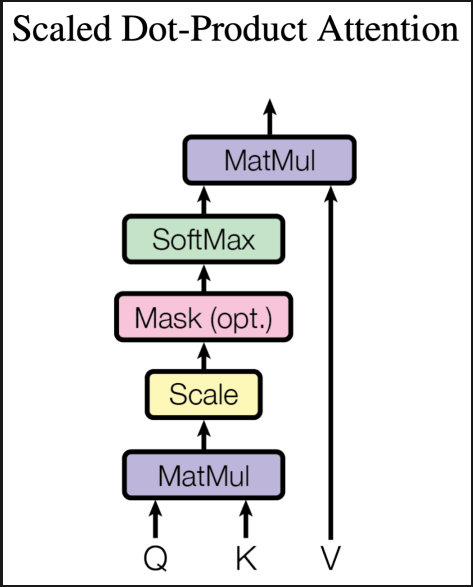
$$V \leftarrow VW^v$$
$$V \leftarrow \texttt{softmax}(VV^\top/\lambda) \cdot V, \quad e.g., \quad \lambda = \sqrt{d}$$

- Highly parallelizable matrix product replaces recurrence
- Each output is a convex combination of all inputs
- Dot product $\langle \mathbf{v}_{i:}, \mathbf{v}_{j:} \rangle$ measures similarity: more similar, more contribution
- Softmax is dense: every output attends to every input

More Generally

$$[Q, K, V] \leftarrow [QW^q, KW^k, VW^v]$$
$$V \leftarrow \texttt{softmax}(QK^\top/\lambda) \cdot V$$

- Q: query, linear transformed by W^q
- K: key, linear transformed by W^k
- V: value, linear transformed by W^v
- Similarity by dot product between query Q and key K
- Output is convex combination of input
- Self-attention becomes a special case: $[Q, K, V] \leftarrow [VW^v, VW^v, VW^v]$



Multi-head

- $h = 8$ or $h = 12$
- keep $d = 512$

for $i = 1, \ldots, h$ **do**

  `// linear transformation`

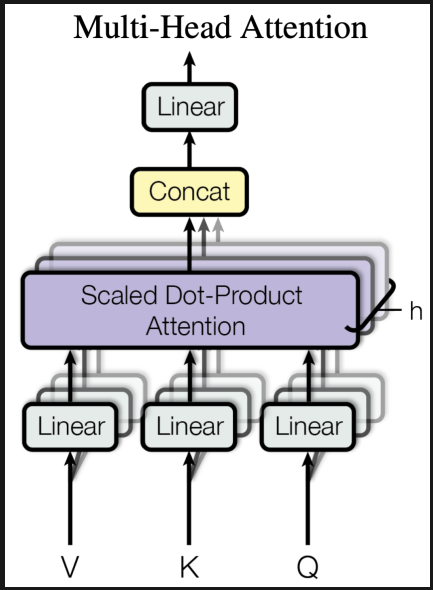
  $[Q, K, V] \leftarrow [QW_i^q, KW_i^k, VW_i^v]$

  `// convex combination`

  $V_i \leftarrow \texttt{softmax}(QK^\top/\lambda)V$

`// concatenation & linear trans`

$V \leftarrow [V_1, \cdots, V_h]W$



Position-wise Feed-forward

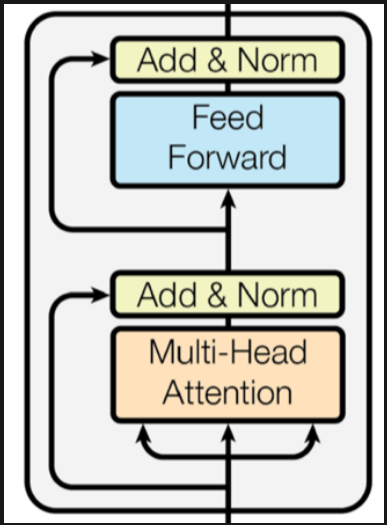
- Let $\mathbf{V} \in \mathbb{R}^{\text{len} \times d}$ be the values (at some layer), arranged row-wise:

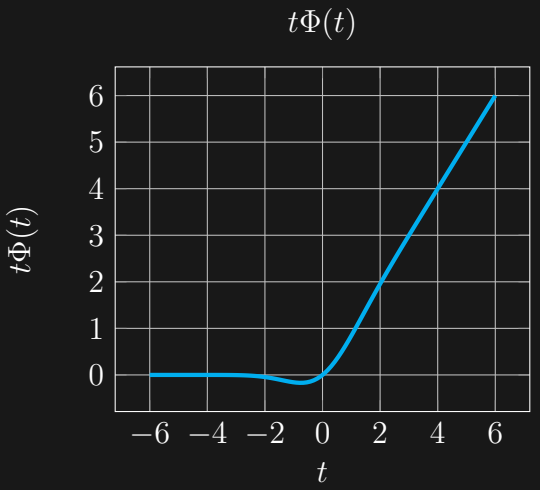
$$\texttt{FFN}(\mathbf{V}) = \sigma(\mathbf{V}\mathbf{W}_1)\mathbf{W}_2$$

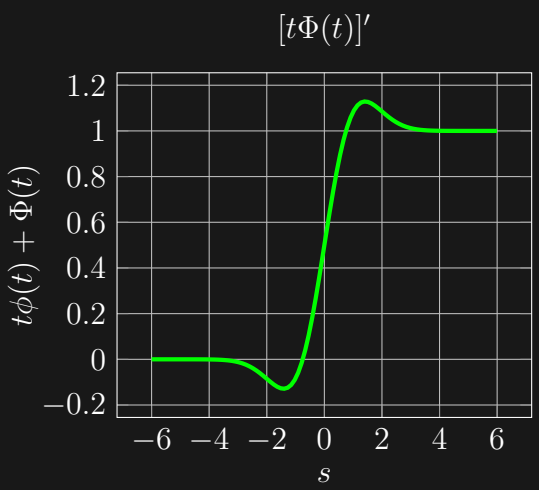
  - two-layer network
  - $\sigma(t) = t\Phi(t)$: Gaussian Error Linear Unit (GELU)

- Weights $\mathbf{W}_1 \in \mathbb{R}^{d \times 4d}$ and $\mathbf{W}_2 \in \mathbb{R}^{4d \times d}$ are shared among different positions but change from layer to layer

- Residual connections and layer-wise normalization

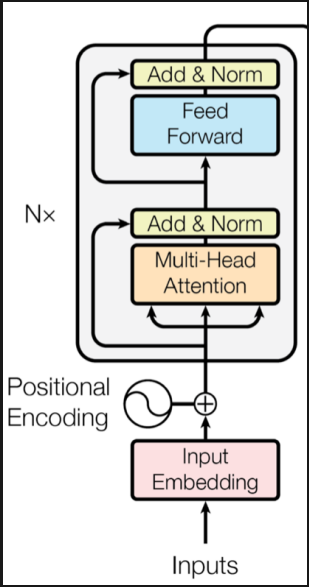






D. Hendrycks and K. Gimpel. "Gaussian Error Linear Units (GELUs)". 2016.

Transformer Encoder



Understanding Attention

- 1x1 conv: consider d filters $W^v = [\mathbf{w}_1, \cdots, \mathbf{w}_d]$ so that

$$u_{ik} \leftarrow \langle \mathbf{v}_{i:}, \mathbf{w}_k \rangle$$

- Global weighted average pooling:

$$\mathbf{v}_{j:} \leftarrow \sum_i p_{ji} \cdot \mathbf{u}_{i:}, \quad \text{where} \quad p_{ji} \propto \exp(\langle \mathbf{u}_{j:}, \mathbf{u}_{i:} \rangle / \lambda)$$

- 1x1 conv: consider $4d$ filters $W = [\mathbf{w}_1, \cdots, \mathbf{w}_{4d}]$ so that

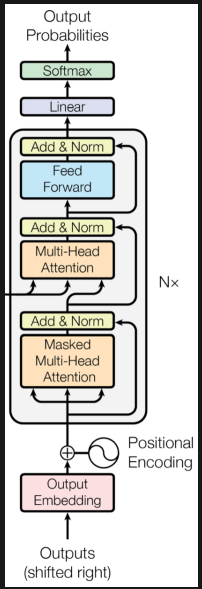
$$u_{ik} \leftarrow \langle \mathbf{v}_{i:}, \mathbf{w}_k \rangle$$

- GELU activation: $\mathbf{v} = \sigma(\mathbf{u})$
- 1x1 conv again with d filters

Transformer Decoder

$$[Q, K, V] \leftarrow [VW^q, VW^k, VW^v]$$
$$V \leftarrow \texttt{softmax}(QK^\top/\lambda) \cdot V$$

- Masked self-attention: each output can only depend on the input and previous outputs
  - simply reset $\langle q_{i:}, k_{j:} \rangle = -\infty$ if $i \leq j$
  - or shift output to the right by 1 position and enforce above for $i < j$
- Context attention: (K, V) from the encoder output while Q from decoder
  - each output attends to every position in input



Transformer

- Last linear transformation and softmax:

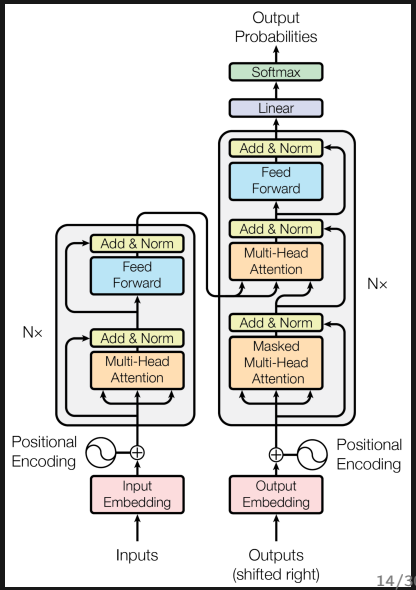
$$\hat{Y} = \texttt{softmax}(\mathbf{V}(\mathbf{W}^e)^\top)$$

  - $\mathbf{W}^e$ is the transpose of the input encoding

- Train by minimizing the log-loss:

$$\min_W \hat{\mathbb{E}}\left[-\left\langle Y, \log \hat{Y}\right\rangle\right]$$

  - $Y = [\mathbf{y}_1, \ldots, \mathbf{y}_l]$ is output sequence, one-hot
  - $\hat{Y} = [\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_l]$ is the predicted probabilities
  - $\hat{\mathbf{y}}_t$ depends on the input sequence $X = [\mathbf{x}_1, \ldots, \mathbf{x}_m]$, as well as previous output tokens $\mathbf{y}_1, \ldots, \mathbf{y}_{t-1}$



Comparison

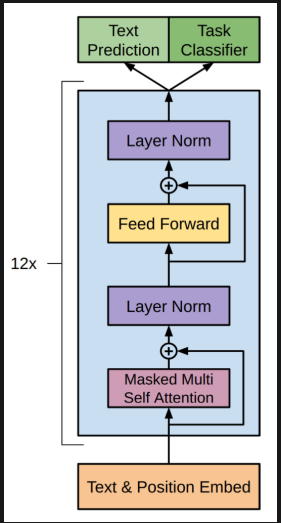
Layer type	per-layer complexity	sequential ops	max path length
Self-attention	$O(m^2 d)$	$O(1)$	$O(1)$
Recurrent	$O(md^2)$	$O(m)$	$O(m)$
Convolution	$O(kmd^2)$	$O(1)$	$O(\log_k m)$
Self-attention (r)	$O(rmd)$	$O(1)$	$O(m/r)$

- m: sequence length
- d: dimension of internal representation
- k: filter size
- r: restricted attention field
- Attention also costs $O(md^2)$ due to linear transformation $V \leftarrow VW$

Does It Work?

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [15]	23.75			
Deep-Att + PosUnk [32]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [31]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [8]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [26]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [32]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [31]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [8]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.0**	$2.3 \cdot 10^{19}$	

Generative Pre-Training (GPT)



- Unsupervised pre-training through a language model:

$$\min_{\Theta} \quad \hat{\mathbb{E}} - \log \prod_{j=1}^{m} p(\mathbf{x}_j | \mathbf{x}_1, \ldots, \mathbf{x}_{j-1}; \Theta)$$

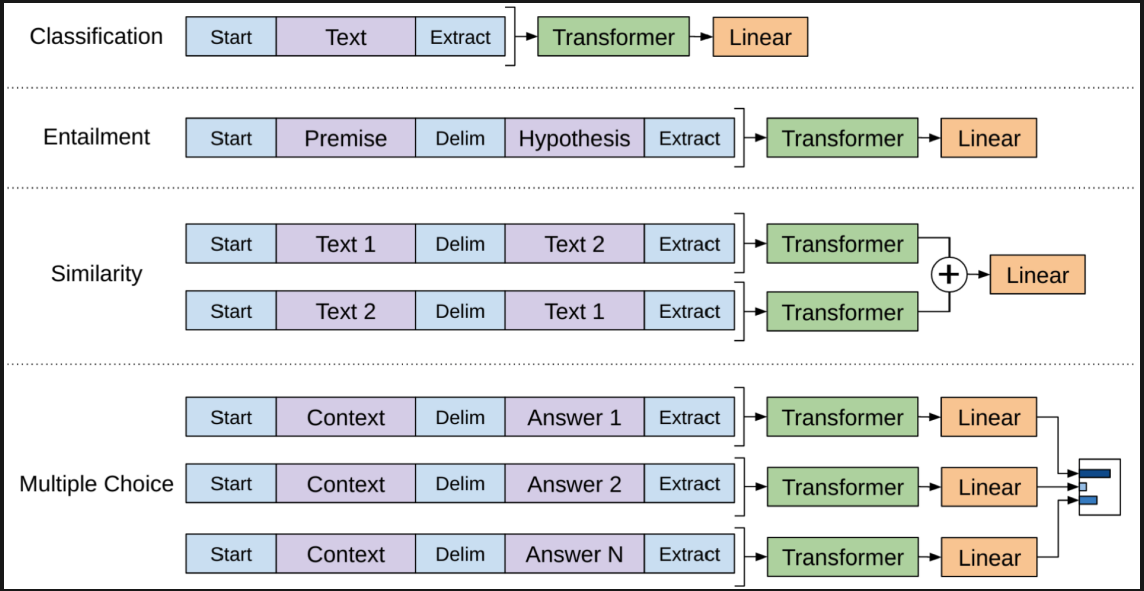
  - given the context consisting of previous tokens $\mathbf{x}_1, \ldots, \mathbf{x}_{j-1}$, predict the current token $\mathbf{x}_j$

- Supervised fine-tuning with task-aware transformations:

$$\min_{W_y} \min_{\Theta} \quad \hat{\mathbb{E}} - \log p(\mathbf{y}|X, \Theta) - \lambda \cdot \hat{\mathbb{E}} \log p(X|\Theta),$$

$$\text{where} \quad \mathrm{p}(\mathbf{y}|X, \Theta) = \left\langle \mathbf{y}, \mathtt{softmax}(\mathbf{h}_m^{(L)} W_y) \right\rangle,$$

A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. "Improving Language Understanding by generative pre-training". 2018.

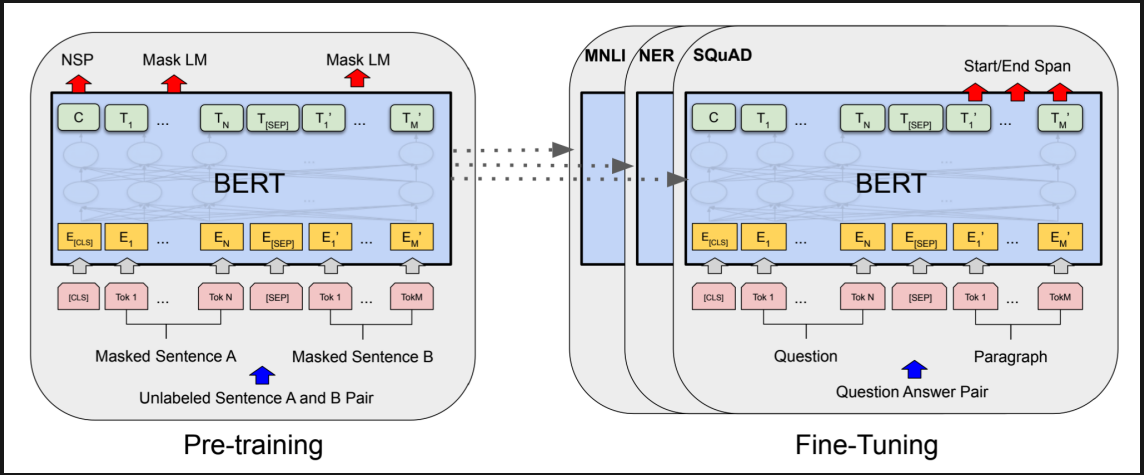

Classification
Start
Text
Extract
Transformer
Linear
Entailment
Start
Premise
Delim
Hypothesis
Extract
Transformer
Linear
Similarity
Start
Text 1
Delim
Text 2
Extract
Transformer
Start
Text 2
Delim
Text 1
Extract
Transformer
Linear
Multiple Choice
Start
Context
Delim
Answer 1
Extract
Transformer
Linear
Start
Context
Delim
Answer 2
Extract
Transformer
Linear
Start
Context
Delim
Answer N
Extract
Transformer
Linear

Method	Classification		Semantic Similarity			GLUE
	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	
Sparse byte mLSTM [16]	-	**93.2**	-	-	-	-
TF-KLD [23]	-	-	**86.0**	-	-	-
ECNU (mixed ensemble) [60]	-	-	-	81.0	-	-
Single-task BiLSTM + ELMo + Attn [64]	35.0	90.2	80.2	55.5	66.1	64.8
Multi-task BiLSTM + ELMo + Attn [64]	18.9	91.6	83.5	72.8	63.3	68.9
Finetuned Transformer LM (ours)	**45.4**	91.3	82.3	**82.0**	**70.3**	**72.8**

Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	89.3	-	-	-
CAFE [58] (5x)	80.2	79.0	89.3	-	-	-
Stochastic Answer Network [35] (3x)	80.6	80.1	-	-	-	-
CAFE [58]	78.7	77.9	88.5	83.3		
GenSen [64]	71.4	71.3	-	-	82.3	59.2
Multi-task BiLSTM + Attn [64]	72.2	72.1	-	-	82.1	**61.7**
Finetuned Transformer LM (ours)	**82.1**	**81.4**	**89.9**	**88.3**	**88.1**	56.0

Method	Story Cloze	RACE-m	RACE-h	RACE
val-LS-skip [55]	76.5	-	-	-
Hidden Coherence Model [7]	77.6	-	-	-
Dynamic Fusion Net [67] (9x)	-	55.6	49.4	51.2
BiAttention MRU [59] (9x)	-	60.2	50.3	53.3
Finetuned Transformer LM (ours)	**86.5**	**62.9**	**57.4**	**59.0**

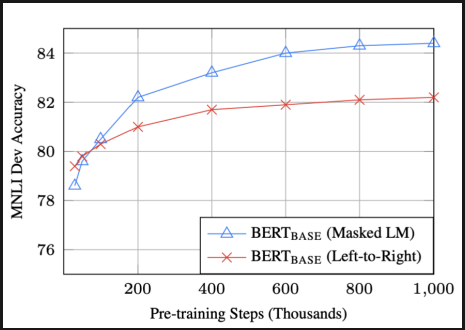
Bidirectional Encoder Representations from Transformers



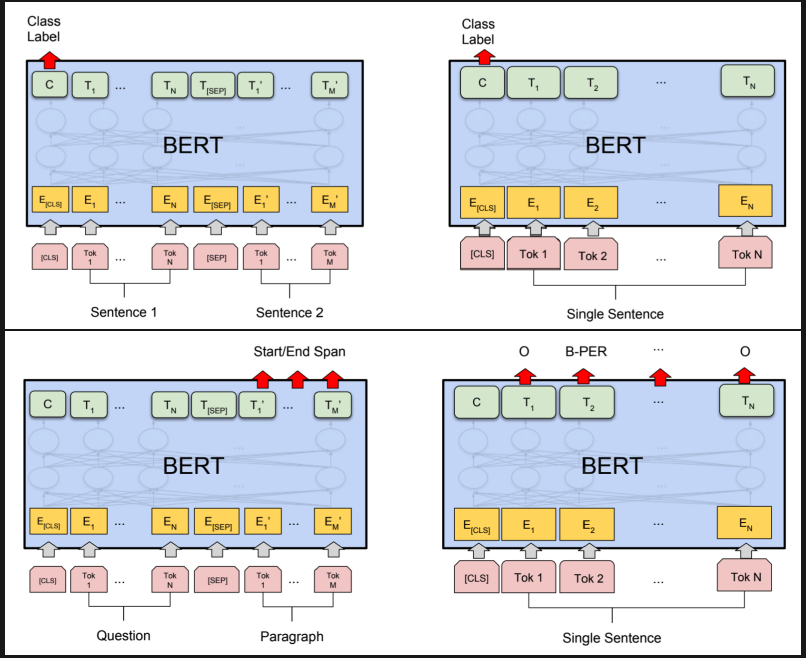
J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. 2019, pp. 4171–4186.

Mask Language Model

- Randomly select 15% input tokens, change to [Mask]
- Add softmax to predict the [Mask] tokens
- Actually 12% replaced with [Mask], 1.5% with random

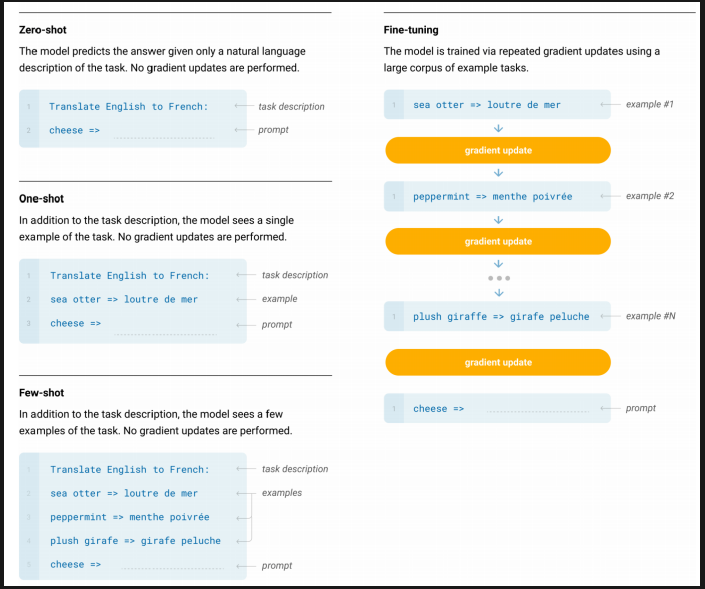


Masking Rates			Dev Set Results		
MASK	SAME	RND	MNLI Fine-tune	NER Fine-tune	NER Feature-based
80%	10%	10%	84.2	95.4	94.9
100%	0%	0%	84.3	94.9	94.0
80%	0%	20%	84.1	95.2	94.6
80%	20%	0%	84.4	95.2	94.7
0%	20%	80%	83.7	94.8	94.6
0%	0%	100%	83.6	94.9	94.6


Class Label
C
T1
...
TN
T[SEP]
T1'
...
TM'
BERT
E[CLS]
E1
...
EN
E[SEP]
E1'
...
EM'
[CLS]
Tok 1
...
Tok N
[SEP]
Tok 1
...
Tok M
Sentence 1
Sentence 2
Class Label
C
T1
T2
...
TN
BERT
E[CLS]
E1
E2
...
EN
[CLS]
Tok 1
Tok 2
...
Tok N
Single Sentence
Start/End Span
C
T1
...
TN
T[SEP]
T1'
...
TM'
BERT
E[CLS]
E1
...
EN
E[SEP]
E1'
...
EM'
[CLS]
Tok 1
...
Tok N
[SEP]
Tok 1
...
Tok M
Question
Paragraph
O
B-PER
...
O
C
T1
T2
...
TN
BERT
E[CLS]
E1
E2
...
EN
[CLS]
Tok 1
Tok 2
...
Tok N
Single Sentence

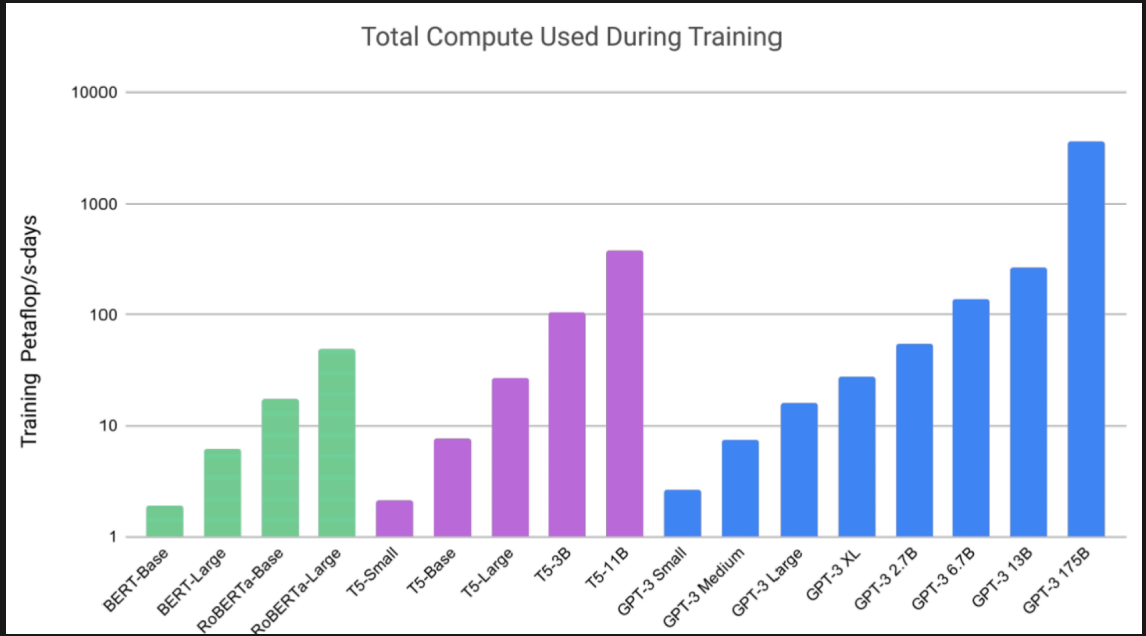
System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

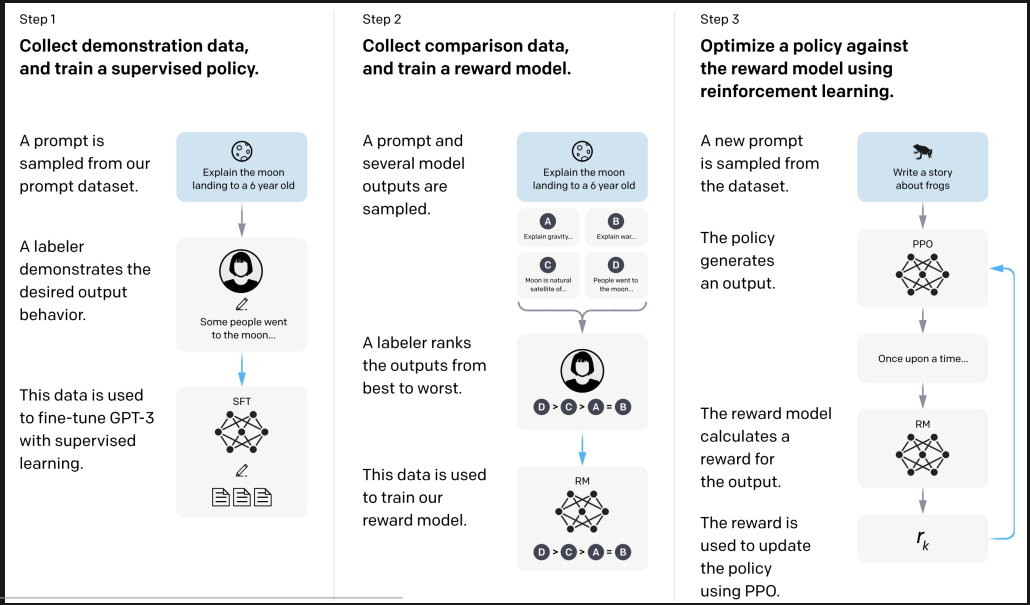
Hyperparams				Dev Set Accuracy		
#L	#H	#A	LM (ppl)	MNLI-m	MRPC	SST-2
3	768	12	5.84	77.9	79.8	88.4
6	768	3	5.24	80.6	82.2	90.7
6	768	12	4.68	81.9	84.8	91.3
12	768	12	3.99	84.4	86.7	92.9
12	1024	16	3.54	85.7	86.9	93.3
24	1024	16	3.23	86.6	87.8	93.7



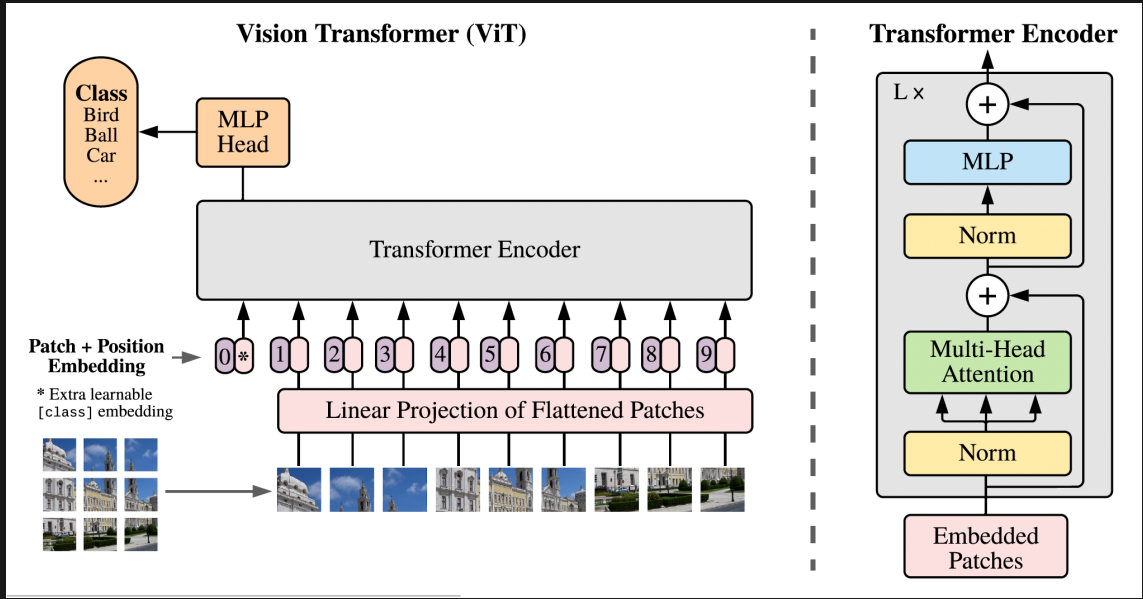
https://beta.openai.com/examples

T. Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. 2020, pp. 1877–1901.

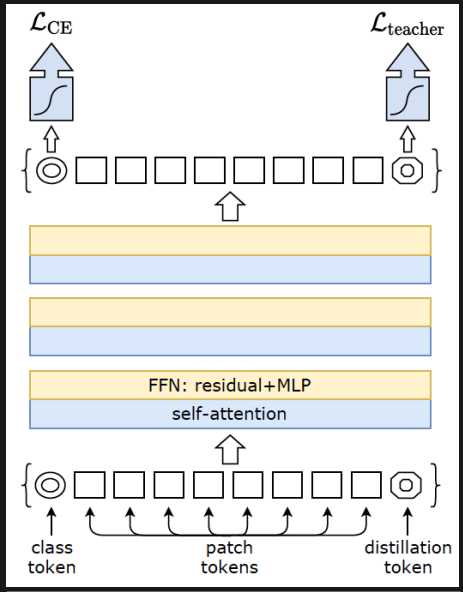

Total Compute Used During Training
Training Petaflop/s-days
10000
1000
100
10
1
BERT-Base
BERT-Large
RoBERTa-Base
RoBERTa-Large
T5-Small
T5-Base
T5-Large
T5-3B
T5-11B
GPT-3 Small
GPT-3 Medium
GPT-3 Large
GPT-3 XL
GPT-3 2.7B
GPT-3 6.7B
GPT-3 13B
GPT-3 175B

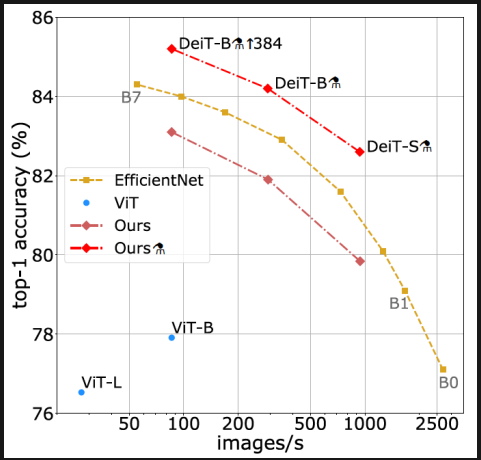


L. Ouyang et al. "Training language models to follow instructions with human feedback". In: *Advances in Neural Information Processing Systems*. 2022.

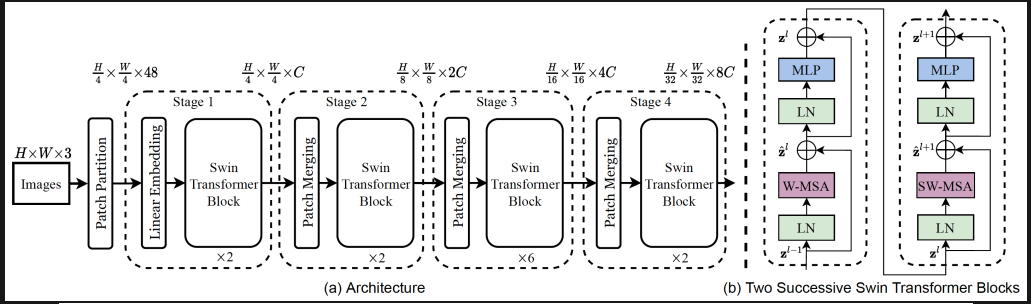


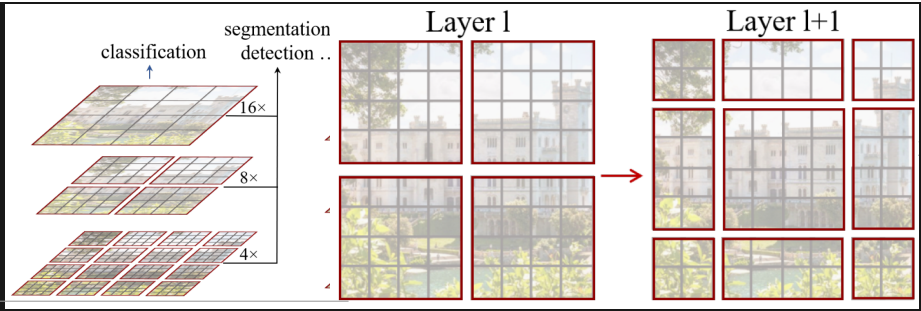
A. Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations*. 2021.





H. Touvron et al. "Training data-efficient image transformers & distillation through attention". In: *Proceedings of the 38th International Conference on Machine Learning*. 2021, pp. 10347–10357.





Z. Liu et al. "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows". In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 9992–10002, Z. Liu et al. "Swin Transformer V2: Scaling Up Capacity and Resolution". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 11999–12009.