AI SUMMER CAMP

TRANSFORMER & NLP

Global AI Hub

Content

Natural Language Processing

Common tasks include:

- text classification
- translation
- summarization
- named entity recognition
- dialogue (chatbots)
- question answering

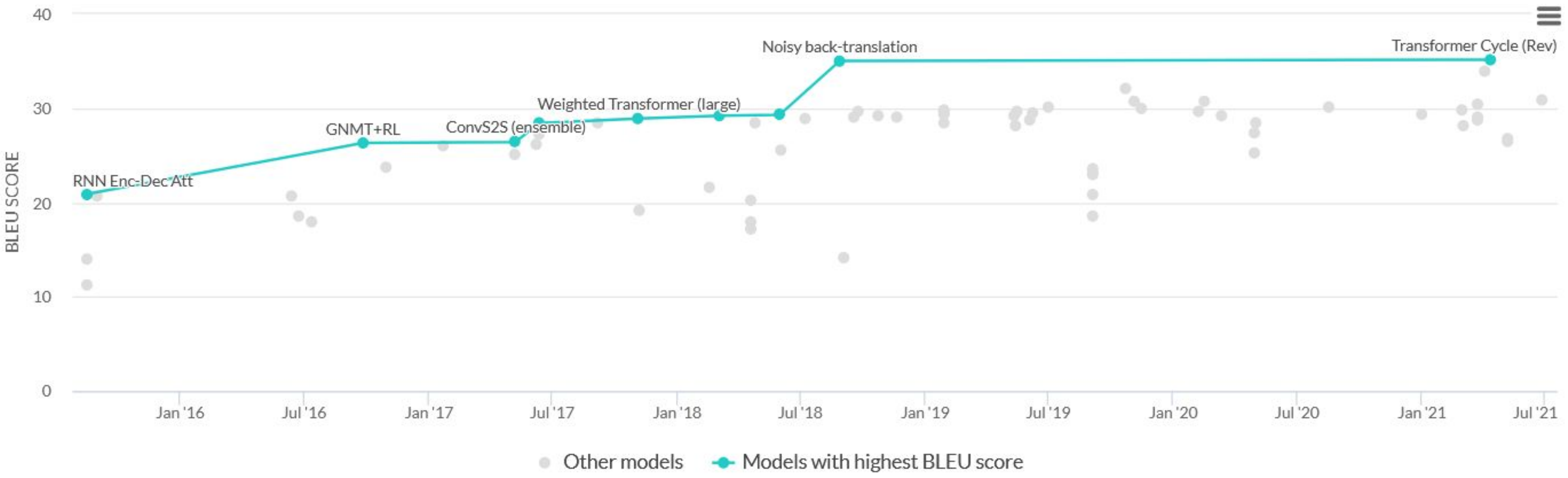
For more info visit https://paperswithcode.com/area/natural-language-processing

Global AI Hub

Machine Translation on WMT2014 English-German

Leaderboard Dataset

View BLEU score by Date for All models



Global AI Hub

1	**Transformer Cycle** (Rev)	35.14	33.54		✓	Lessons on Parameter Sharing across Layers in Transformers	2021	Transformer
2	**Noisy back-translation**	35.0	33.8		✓	Understanding Back-Translation at Scale	2018	
3	**Transformer+Rep** (Uni)	33.89	32.35		✓	Rethinking Perturbations in Encoder-Decoders for Fast Training	2021	Transformer
4	**T5-11B**	32.1			✓	Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer	2019	Transformer
5	**Transformer + R-Drop**	30.91			✗	R-Drop: Regularized Dropout for Neural Networks	2021	Transformer
6	**BERT-fused NMT**	30.75			✗	Incorporating BERT into Neural Machine Translation	2020	Transformer
7	**Data Diversification - Transformer**	30.7			✗	Data Diversification: A Simple Strategy For Neural Machine Translation	2019	Transformer
8	**Mask Attention Network** (big)	30.4		215M	✗	Mask Attention Networks: Rethinking and Strengthen Transformer	2021	
9	**Transformer** (ADMIN init)	30.1	29.5		✗	Very Deep Transformers for Neural Machine Translation	2020	Transformer

Transformer

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Recurrent Neural Network (RNN)

The Vanilla RNN Model

First time-step ($t = 1$):

$$\mathbf{h}_1 = tanh\left(W^{xh} \cdot \mathbf{x}_1 + W^{hh} \cdot \mathbf{h}_0\right)$$

$$\hat{\mathbf{y}}_1 = softmax\left(W^{hy} \cdot \mathbf{h}_1\right)$$

$$L_1 = CE(\hat{\mathbf{y}}_1, \mathbf{y}_1)$$

In general:

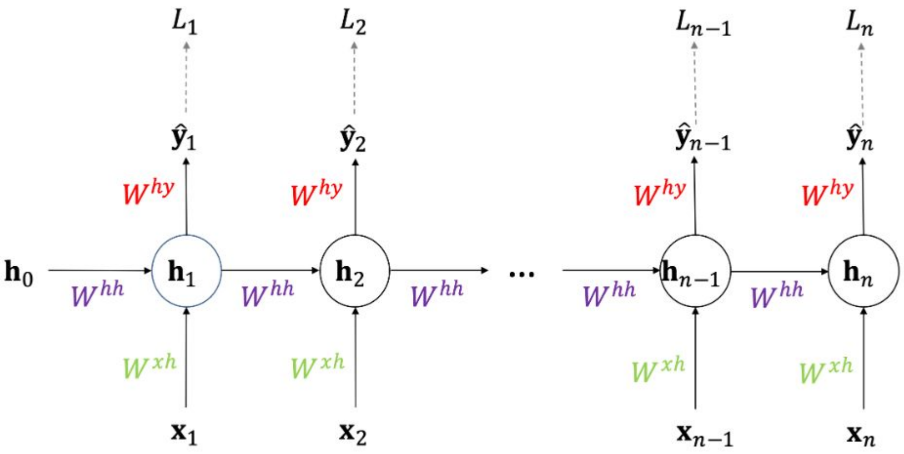
$$\mathbf{h}_t = tanh\left(W^{xh} \cdot \mathbf{x}_t + W^{hh} \cdot \mathbf{h}_{t-1}\right)$$

$$\hat{\mathbf{y}}_t = softmax\left(W^{hy} \cdot \mathbf{h}_t\right)$$

$$L_t = CE(\hat{\mathbf{y}}_t, \mathbf{y}_t)$$

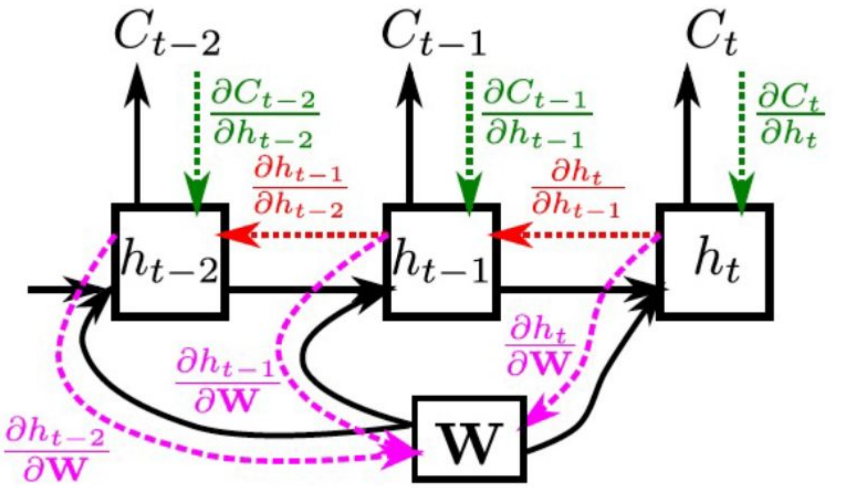
In total:

$$L = \sum_t L_t$$



Global AI Hub

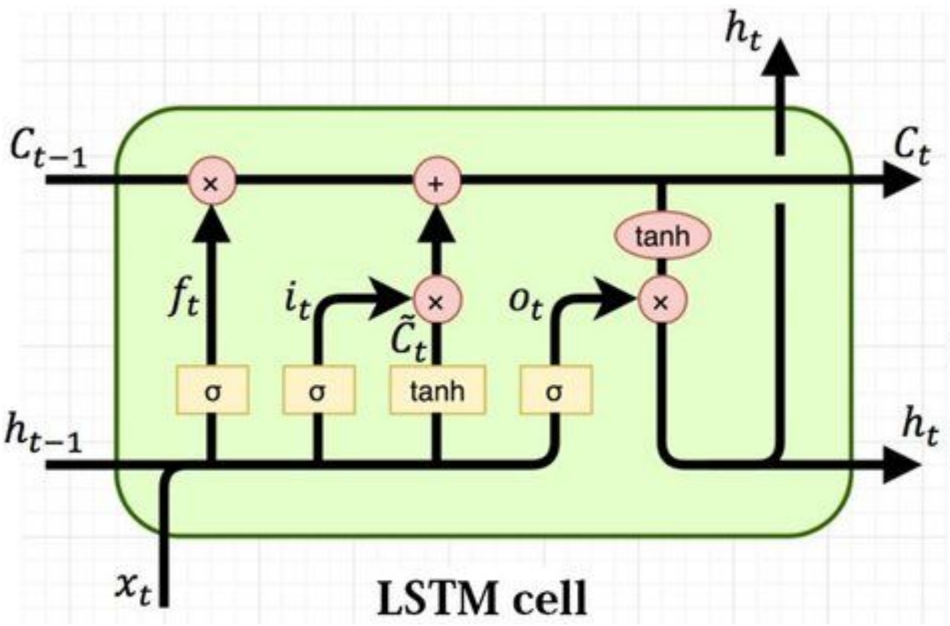
Backpropagation through time



$$\frac{\partial C_t}{\partial \mathbf{W}} = \sum_{t'=1}^{t} \frac{\partial C_t}{\partial h_t} \frac{\partial h_t}{\partial h_{t'}} \frac{\partial h_{t'}}{\partial \mathbf{W}}, \text{ where } \frac{\partial h_t}{\partial h_{t'}} = \prod_{k=t'+1}^{t} \frac{\partial h_k}{\partial h_{k-1}}$$

Global AI Hub

Long Short Term Memory (LSTM)



$$i_t = \sigma\left(x_t U^i + h_{t-1} W^i\right)$$

$$f_t = \sigma\left(x_t U^f + h_{t-1} W^f\right)$$

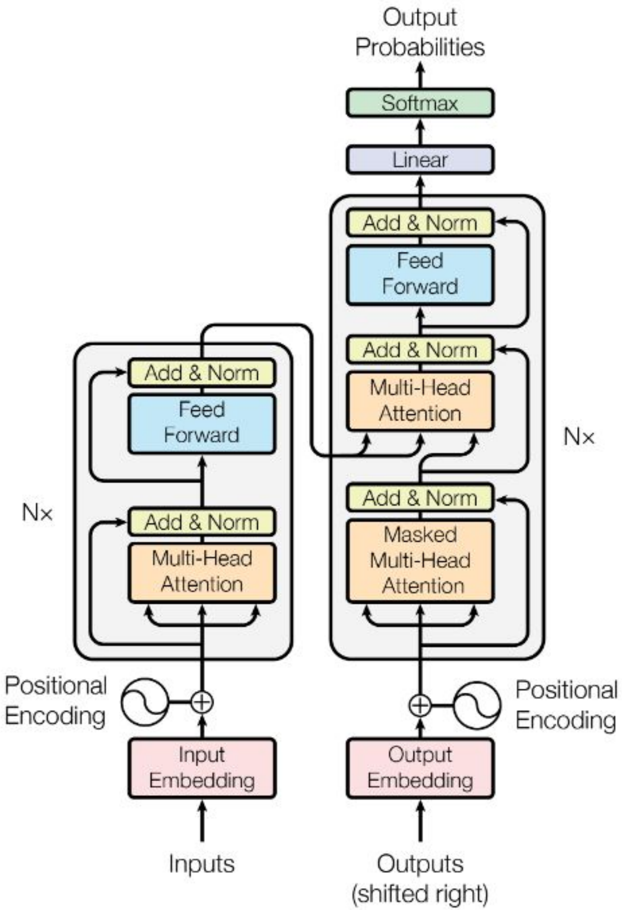
$$o_t = \sigma\left(x_t U^o + h_{t-1} W^o\right)$$

$$\tilde{C}_t = \tanh\left(x_t U^g + h_{t-1} W^g\right)$$

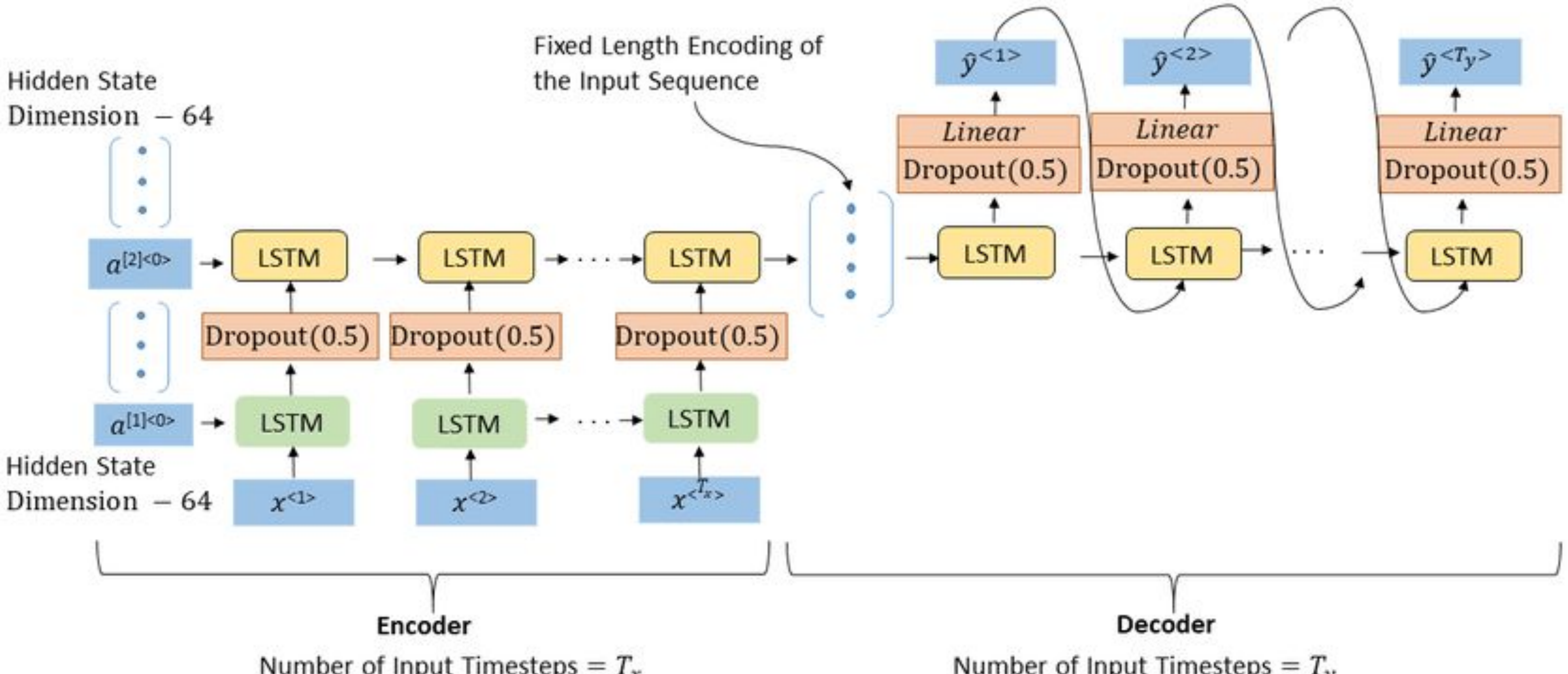
$$C_t = \sigma\left(f_t * C_{t-1} + i_t * \tilde{C}_t\right)$$

$$h_t = \tanh(C_t) * o_t$$

Transformer

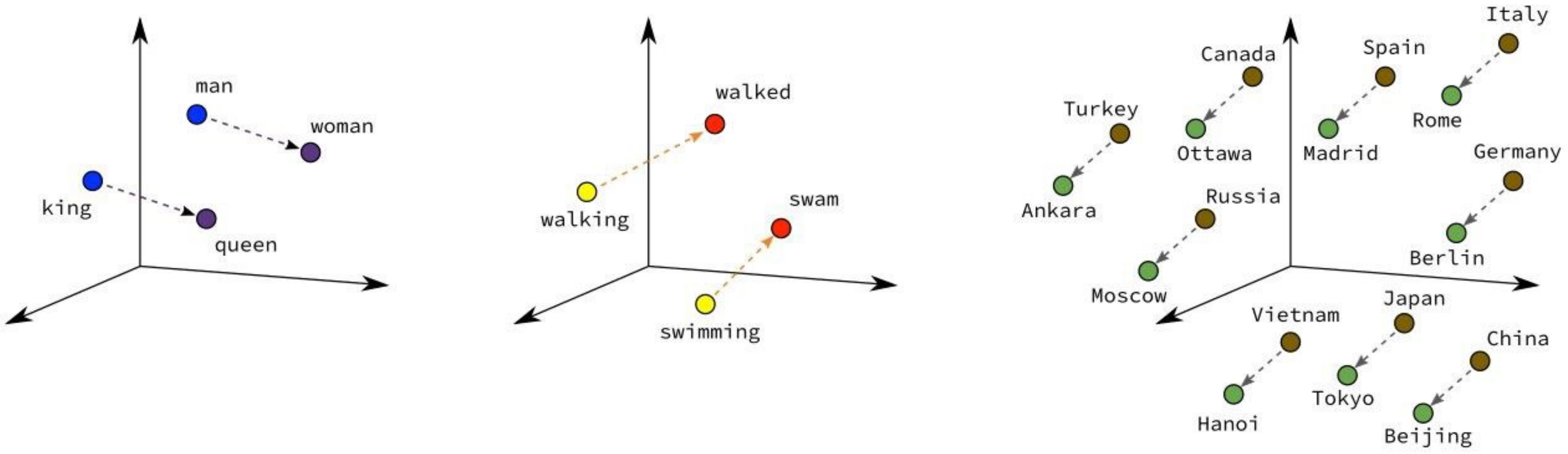


Encoder Decoder



$x^{<t>}$ = [pollution<t-1>, dew point, temperature, pressure, wind direction, wind speed, snow, rain]

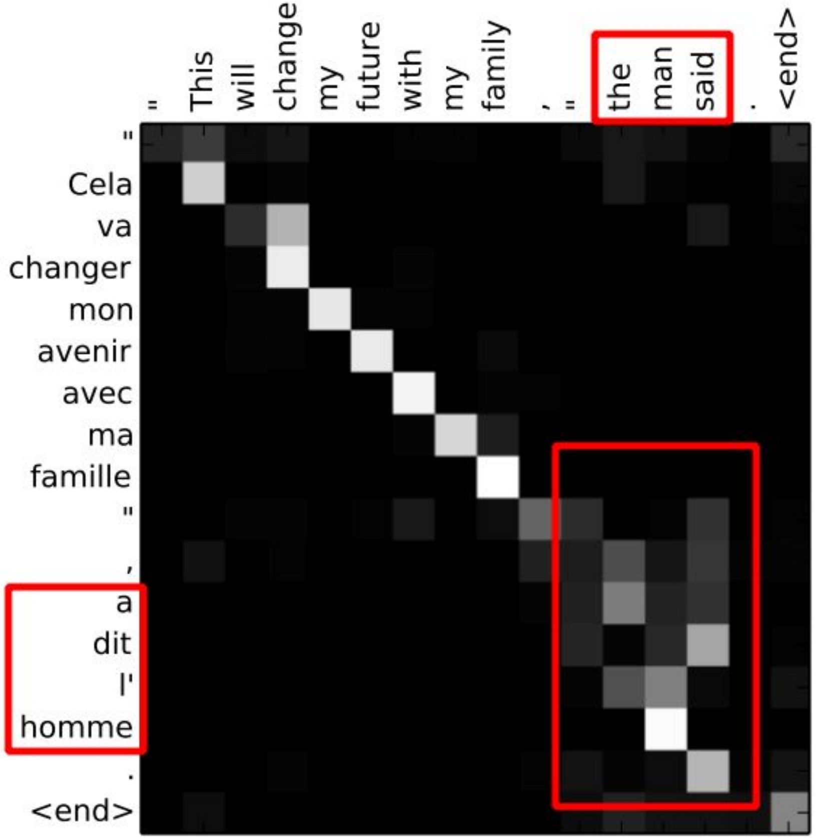
Word Embedding



Male-Female   Verb Tense   Country-Capital

https://projector.tensorflow.org/

Attention



" This will change my future with my family , " the man said . <end>
" Cela va changer mon avenir avec ma famille " , a dit l' homme . <end>

Scaled Dot Product Attention

$$attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \qquad (4)$$

- Q -> query
- K -> key
- V -> value

$$\underset{\text{yemek}}{\begin{bmatrix} 2 & -1 \end{bmatrix}} \bullet \underset{\text{koşarken}}{\begin{bmatrix} -3 \\ 1 \end{bmatrix}} = 2*(-3) + (-1)*1 = \mathbf{-7}$$

$$\underset{\text{yemek}}{\begin{bmatrix} 2 & -1 \end{bmatrix}} \bullet \underset{\text{yemek}}{\begin{bmatrix} 2 \\ -1 \end{bmatrix}} = \mathbf{5}$$

$$\underset{\text{yemek}}{\begin{bmatrix} 2 & -1 \end{bmatrix}} \bullet \underset{\text{yedik}}{\begin{bmatrix} 2 \\ 1 \end{bmatrix}} = \mathbf{3}$$

(1)

$$\begin{aligned} softmax(-7) &= 10^{-6} \\ softmax(5) &= 0.88 \\ softmax(3) &= 0.12 \end{aligned} \tag{2}$$

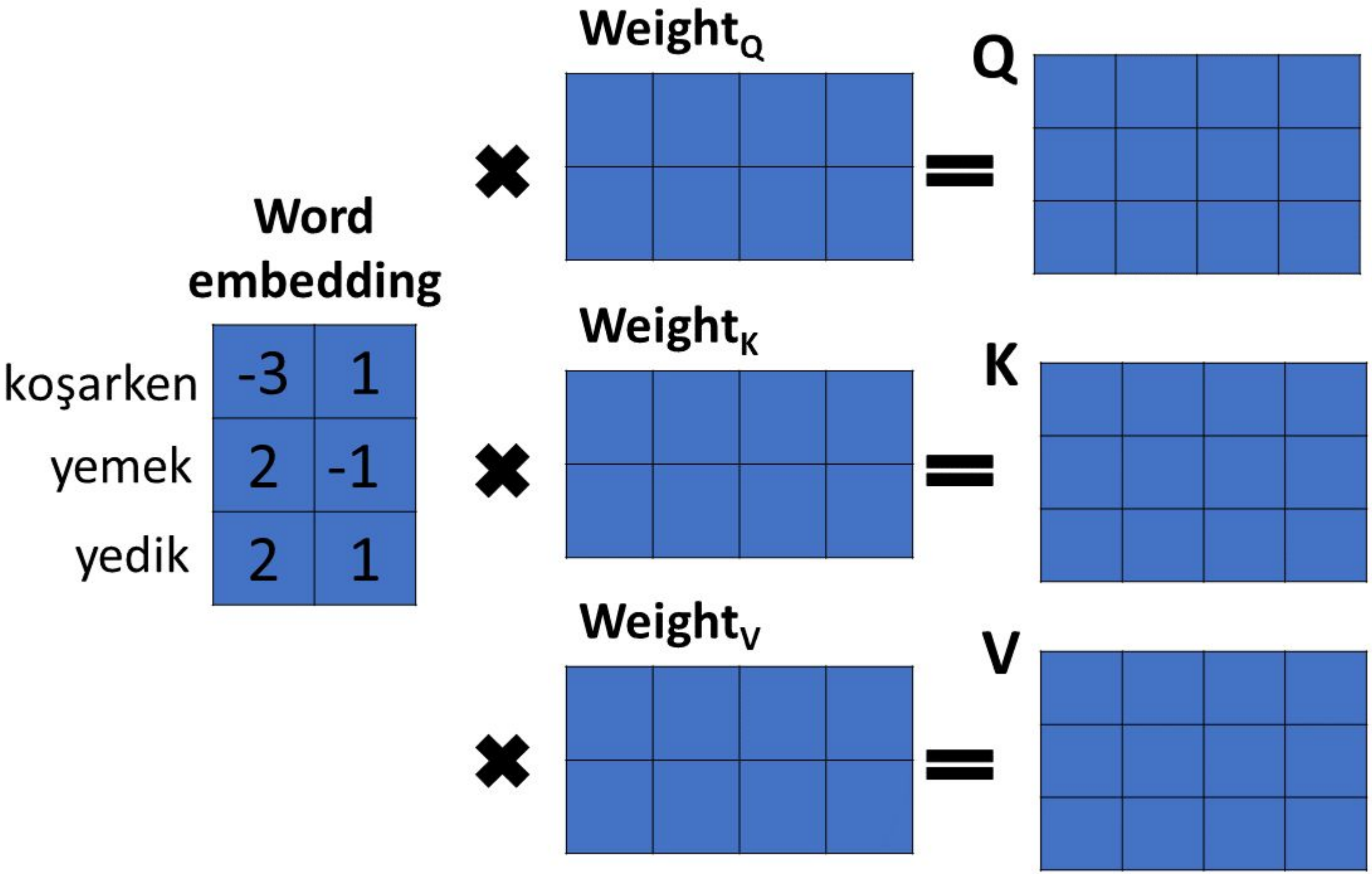
$$10^{-6}\overset{\text{koşarken}}{\begin{bmatrix}-3 & 1\end{bmatrix}} + 0.88\,\overset{\text{yemek}}{\begin{bmatrix}2 & -1\end{bmatrix}} + 0.12\,\overset{\text{yedik}}{\begin{bmatrix}2 & 1\end{bmatrix}}$$

$$= \overset{\text{yemek'}}{\begin{bmatrix}2 & -0.76\end{bmatrix}} \qquad (3)$$

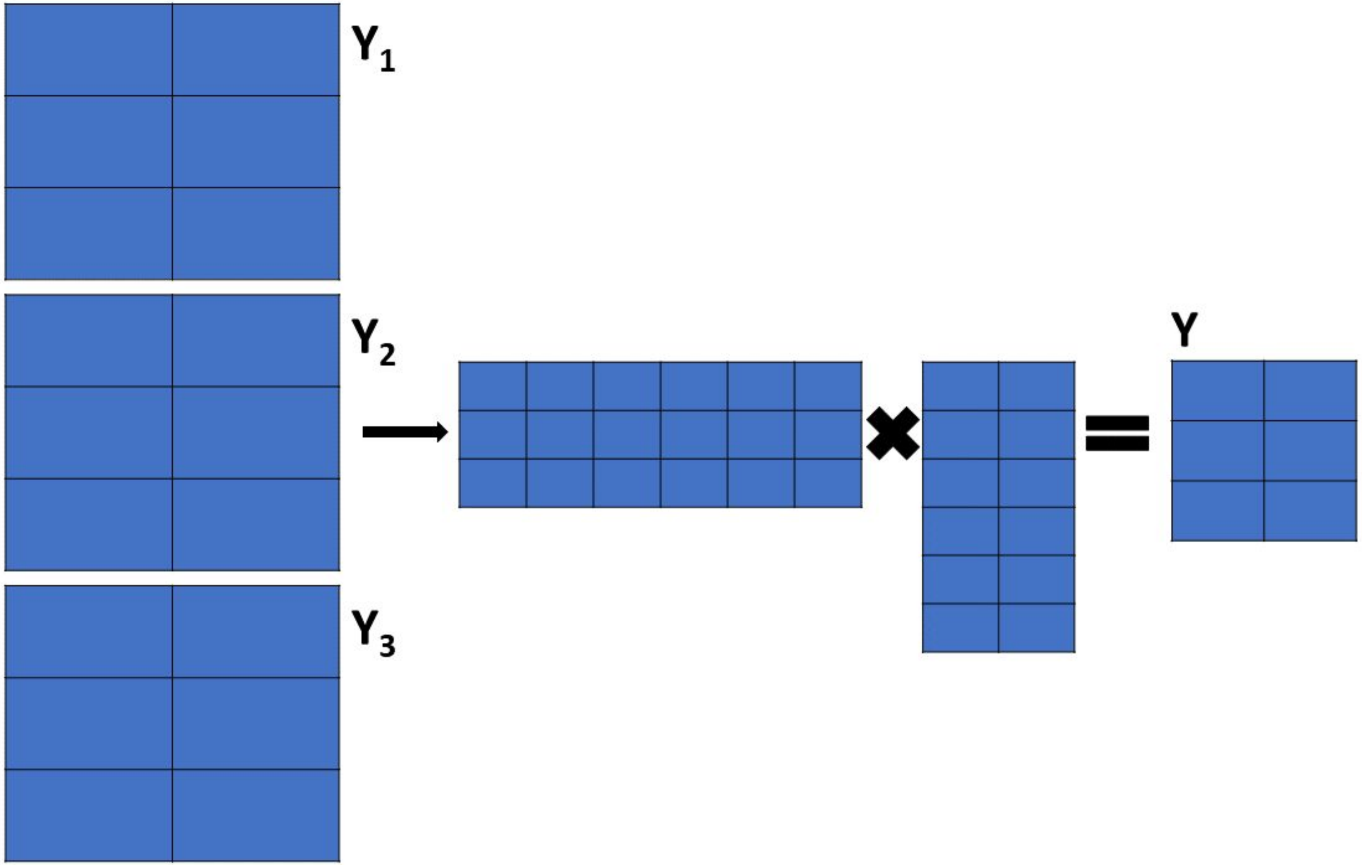
$$\begin{array}{rc} & \textbf{Query} \\ \begin{array}{r}\text{koşarken}\\ \text{yemek}\\ \text{yedik}\end{array} & \begin{bmatrix}-3 & 1\\ 2 & -1\\ 2 & 1\end{bmatrix}\end{array} \times \begin{array}{c} \textbf{Key}^T \\ \begin{bmatrix}-3 & 2 & 2\\ 1 & -1 & 1\end{bmatrix} \\ \text{koşarken}\quad\text{yemek}\quad\text{yedik}\end{array} = \begin{array}{rc} \begin{array}{r}\text{koşarken}\\ \text{yemek}\\ \text{yedik}\end{array} & \begin{bmatrix}10 & -7 & -5\\ -7 & 5 & 3\\ -5 & 3 & 5\end{bmatrix} \\ & \text{koşarken}\quad\text{yemek}\quad\text{yedik}\end{array} \qquad (5)$$

$$\mathrm{softmax}\left(\frac{\begin{bmatrix} 10 & -7 & -5 \\ -7 & 5 & 3 \\ -5 & 3 & 5 \end{bmatrix}}{\sqrt{2}}\right) = \begin{bmatrix} 0.99 & 10^{-6} & 10^{-5} \\ 10^{-4} & 0.80 & 0.19 \\ 10^{-4} & 0.19 & 0.80 \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} 0.99 & 10^{-6} & 10^{-5} \\ 10^{-4} & 0.80 & 0.19 \\ 10^{-4} & 0.19 & 0.80 \end{bmatrix} \times \begin{array}{c} \textbf{Value} \\ \begin{bmatrix} -3 & 1 \\ 2 & -1 \\ 2 & 1 \end{bmatrix} \end{array} = \begin{array}{cl} \textbf{Y} & \\ \begin{bmatrix} -2.97 & 0.99 \\ 1.98 & -0.61 \\ 1.98 & 0.61 \end{bmatrix} & \begin{array}{l} \text{koşarken} \\ \text{yemek (7)} \\ \text{yedik} \end{array} \end{array}$$


Word embedding
koşarken
-3
1
yemek
2
-1
yedik
2
1
Weight$_Q$
Weight$_K$
Weight$_V$
Q
K
V

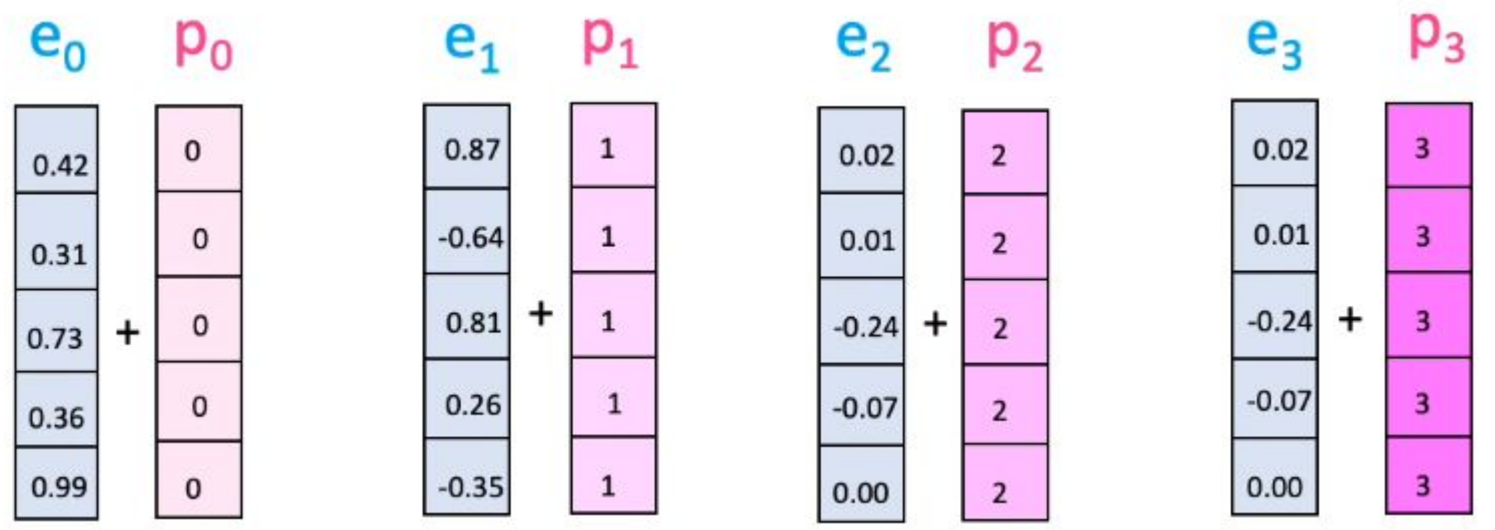
(8)



(9)

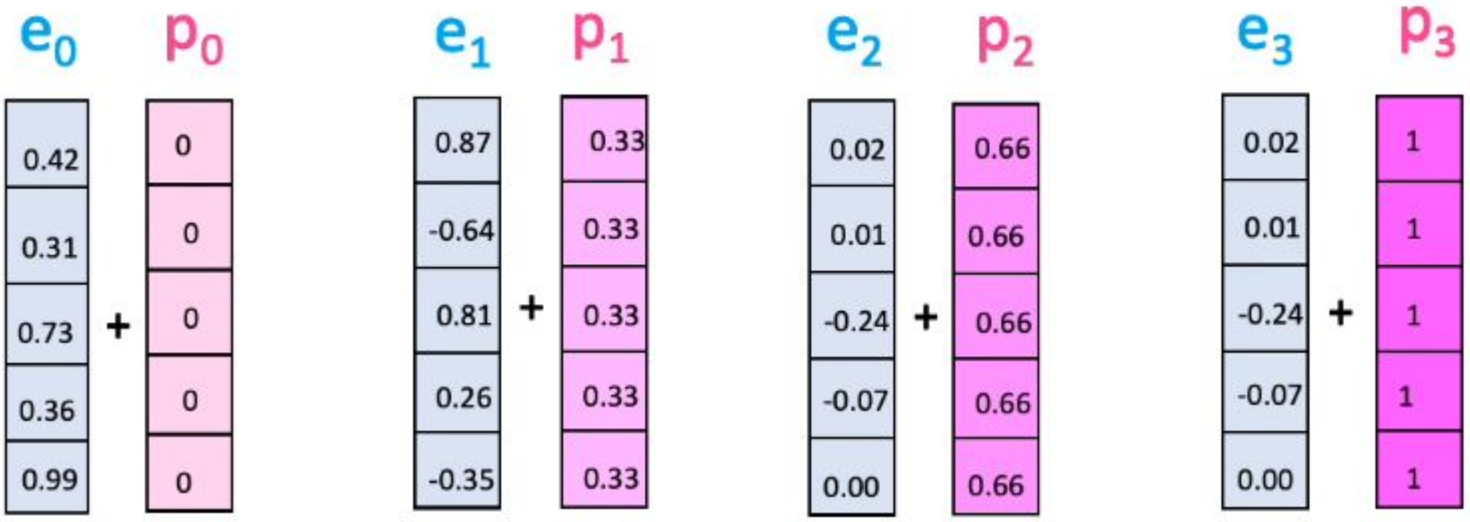
Positional Encoding

Solution #1



Positional Encoding

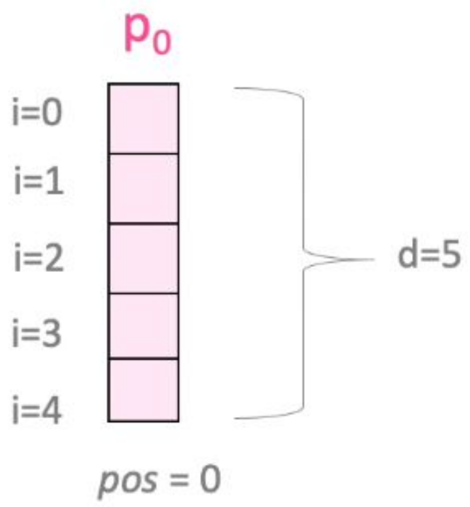
Solution #2



Positional Encoding

Solution #3

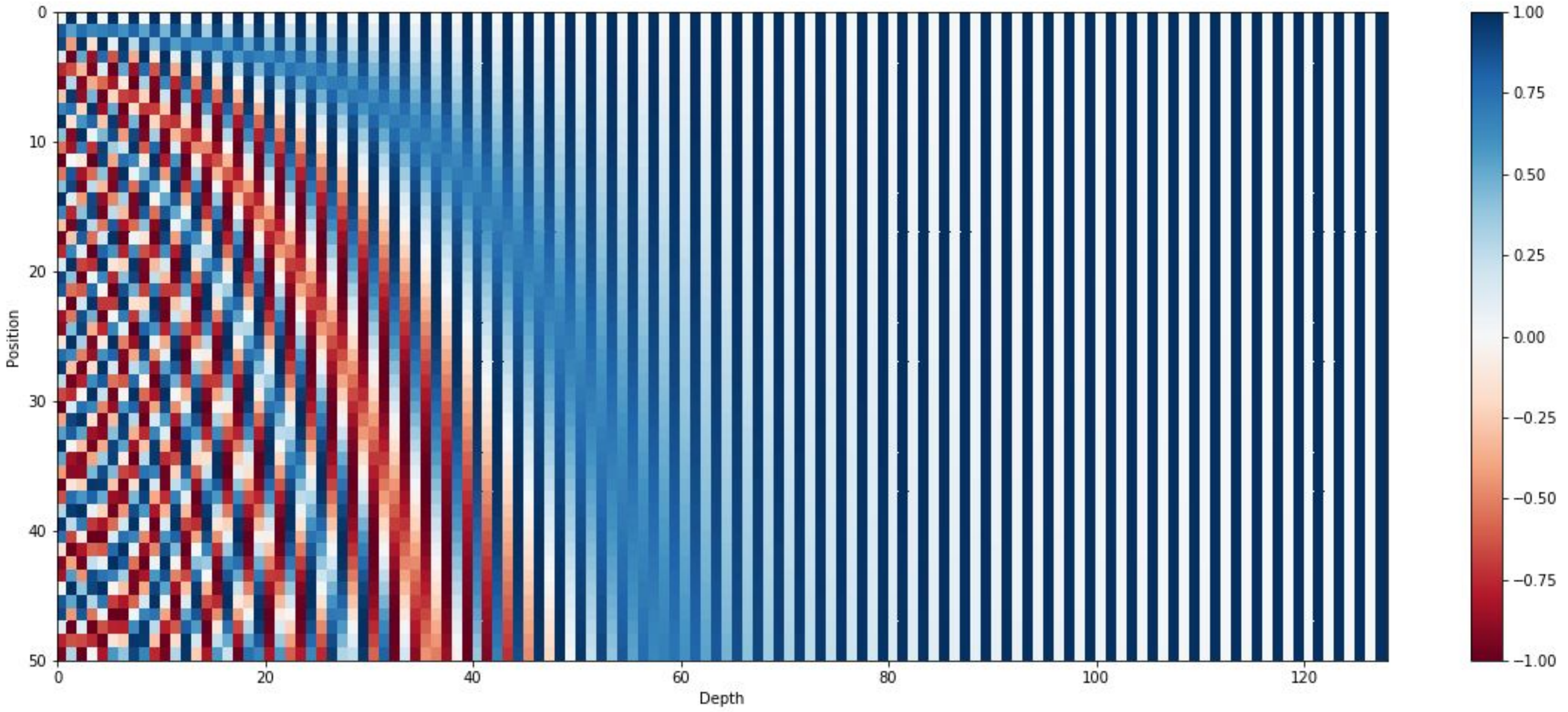
$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$



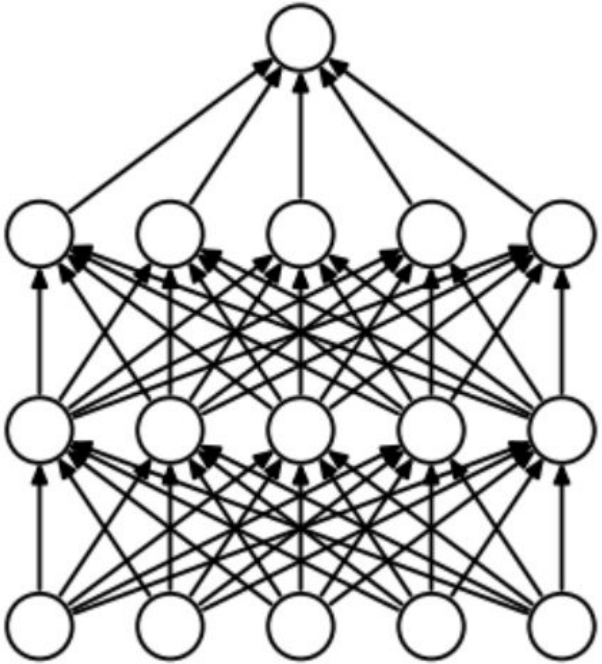
$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

https://www.youtube.com/watch?v=dichIcUZfOw

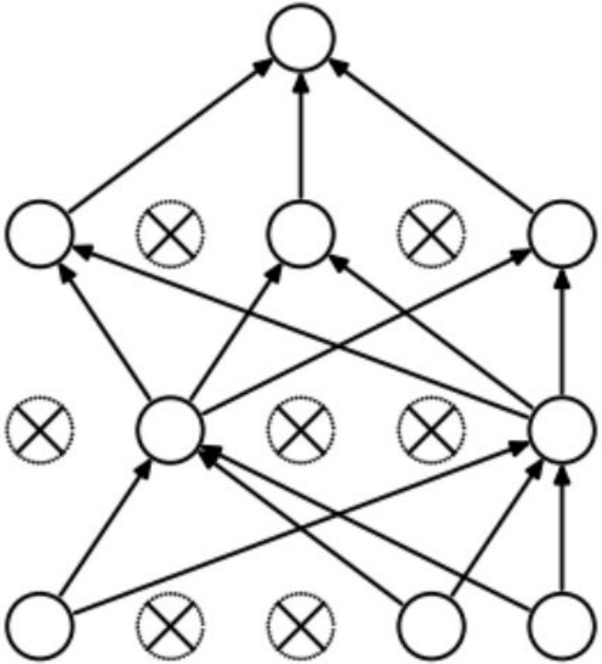
Global AI Hub



Dropout

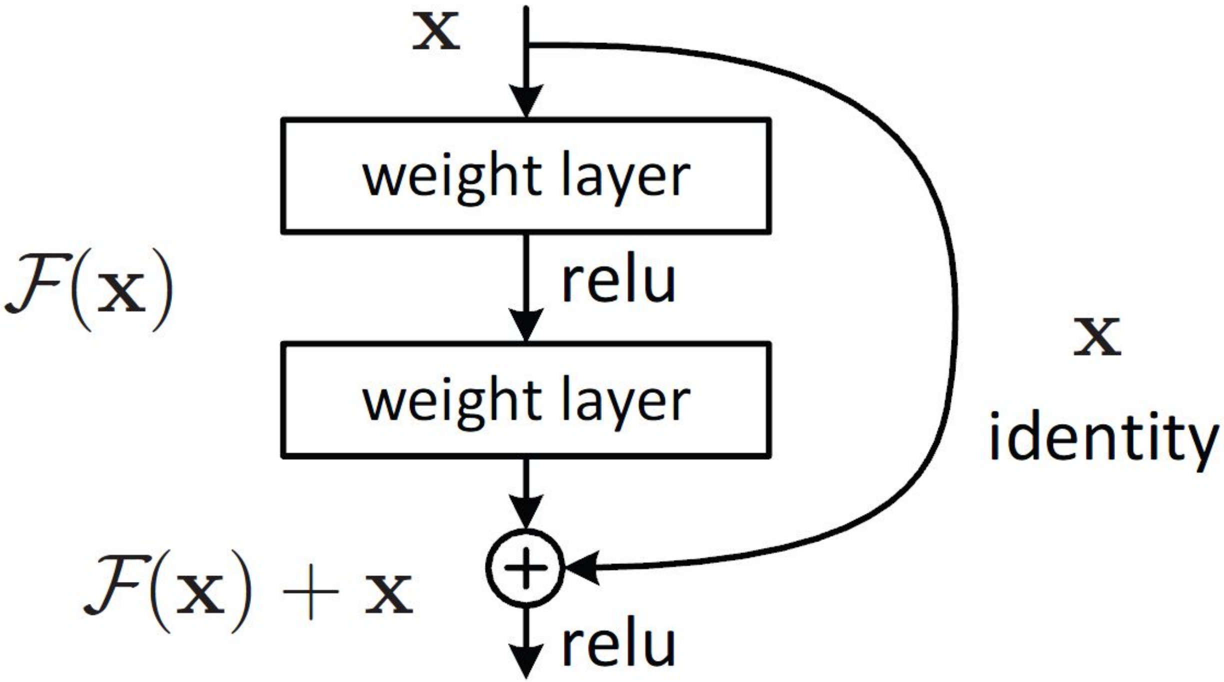


(a) Standard Neural Net



(b) After applying dropout.

Residual Connection



Complexity

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

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)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Advantages

parallelizable (thus faster)

computationally less complex (most of the time)

better capture longer dependencies

more interpretable

Improved Transformers

- **BERT**
- GPT
- T5
- BART
- Pegasus
- XLM
- Reformer
- Longformer
- ELECTRA
- RoBERTa
- ...

Vision Transformers

An Image is Worth 16x16 Words:Transformers for Image Recognition at Scale

https://arxiv.org/pdf/2010.11929.pdf

Global AI Hub

Courses & Resources

- Stanford University NLP w/ DL - http://web.stanford.edu/class/cs224n/
- Huggingface - https://huggingface.co/course/chapter1
- Deeplearning.ai NLP - https://www.deeplearning.ai/program/natural-language-processing-specialization/

Bibliography

- https://user.ceng.metu.edu.tr/~skalkan/DL/week_13.pdf
- https://kazemnejad.com/blog/transformer_architecture_positional_encoding/
- https://www.youtube.com/watch?v=dichIcUZfOw
- https://arxiv.org/pdf/1706.03762.pdf