BERT

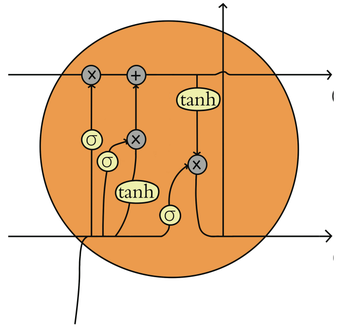
Bidirectional Encoder Representations from Transformers

FORWARD AND BACKWARD
CONTEXTUAL UNDERSTANDING

EXTRACT VECTOR REPRESENTATIONS
ON A TOKEN OR SENTENCE LEVEL

ATTENTION BASED MODEL

ELMo



Replaced fixed word embeddings with context specific embeddings:
used a bi-directional LSTM to generate the embeddings

Trained to predict the next word in a sequence: Laguage Modeling

Semi-supervised → can simply use large corpora, eg. all of wikipedia

ENABLED A NEW ERA OF PRE-TRAINED NLP MODELS CAPTURING BOTH MEANING AND CONTEXT

Open AI - Transformer

Replaced LSTMs with transformer blocks → only used the decoder steps

Thus, it was also trained to predict the next word: Language Modeling.

However, self-attention allowed a better handle on long-term context

OUTPERFORMED ELMo BY BETTER CAPTURING CONTEXT OF WORDS OVER LONG RANGES

Transfer Learning

With ELMo or the OpenAI transformer pre-trained by language modeling on a sufficiently large corpus, models could be easily fine-tuned for specific tasks, such as sentiment analysis.

SUDDENLY, TASK-SPECIFIC MODELS COULD EASILY BE BUILT USING VERY LARGE PRE-TRAINED MODELS, AS HAD BEEN COMMON-PLACE IN MACHINE VISION

Adding bi-directionality: BERT

The OpenAI transformer was only a feed-forward language model, whereas ELMo was bi-directional.

BERT solved this with the **Masked Language Model:**

15% of the input is randomly masked, and the model is tasked with predicting the identity of the masked token.

THUS BERT HAD THE **LONG RANGE CONTEXT** OF THE OPENAI TRANSFORMER AND THE **BI-DIRECTIONAL AWARENESS** OF ELMO

What is a transformer?

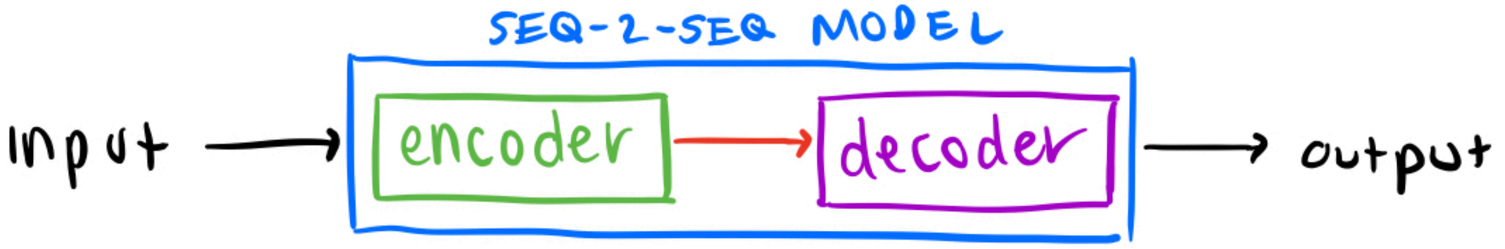
Architecture for transforming one sequence into another.

Uses the encoder / decoder structure of seq-to-seq models, but does not use recurrent networks.

SEQUENCE-TO-SEQUENCE MODELS

Great for language tasks such as translation
summarization
image captioning

Typically consist of an encoder / decoder architecture:



where a context vector is
generated by the encoder
& passed to the decoder

SEQUENCE-TO-SEQUENCE MODELS

The context vector was usually the hidden state from the LSTM/RNN in the last encoding stage

However, these models could not deal with long sequences.
The context vector would lose context

ATTENTION

Instead of passing the last hidden state, pass ALL HIDDEN STATES

The decoder then scores the hidden states to amplify important contexts and dampen unimportant ones.

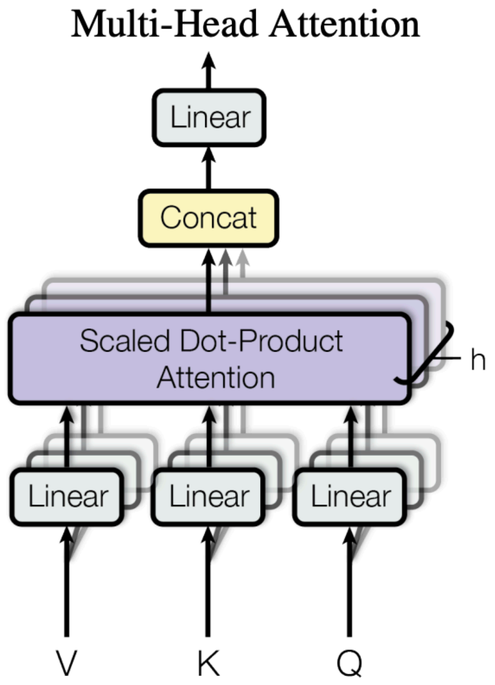
In this way, the model can identify important parts of the input and use them in the decoder to produce output.

MULTIHEADED ATTENTION

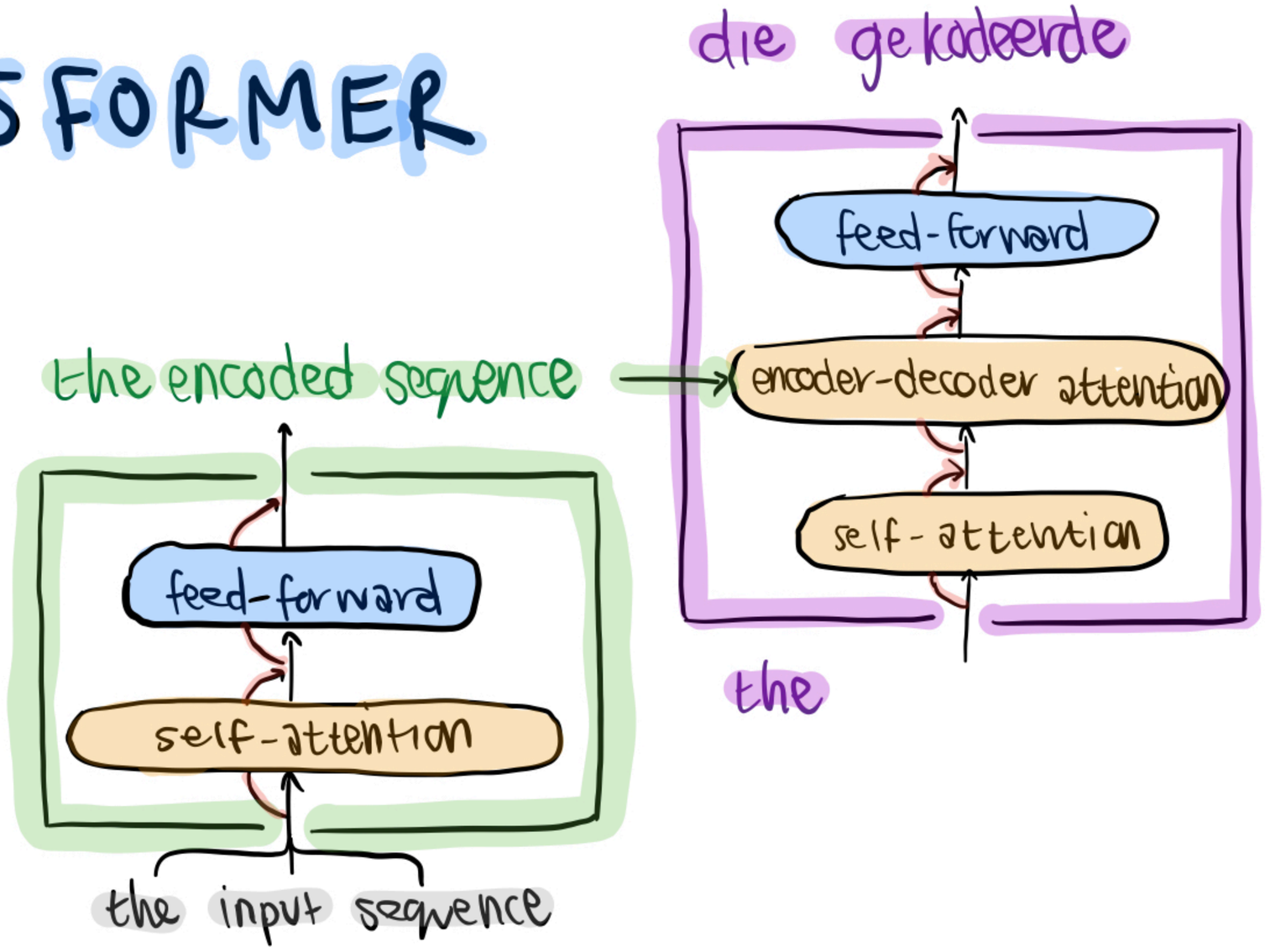
Use a learned linear projection of the vectors involved in attention.

The attention function can be applied in parallel, yielding multiple "attention subspaces" → thereby enhancing the models ability to focus on multiple positions

These are concatenated and projected into the correct feature space.



TRANSFORMER



What can BERT do?

Outperformed the SOTA models on 11 NLP tasks, ranging from machine translation to Question Answering, sentiment analysis, linguistic acceptability and semantic equivalency

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**

What is Next?

RoBERTa = Robustly Trained BERT approach; FAIR
* Dynamic Masking
* 1024 V100 Tesla GPUs
* 10x more training data (160 GB)
* Outperforms BERT and XLnet (2-20%)

XLNet; Google Brain
* permutation language modeling: All tokens predicted but in random order.
* 113GB of data
* 512 TPUs for 2½ days
* 2-15% improvement over BERT

DistilBERT; Hugging face
* Approximated version of BERT → 95% performance but ½ parameters
* Once large network has been trained, it's output distributions can be approximated
* 4x less training than BERT; same dataset.