CS167: Machine Learning

Transformers

Monday, December 2nd, 2024

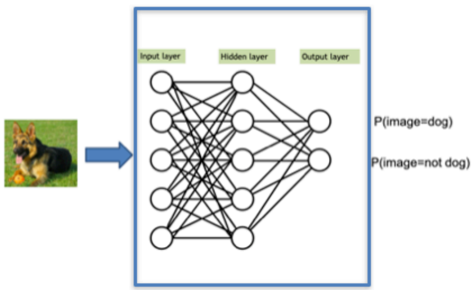
Drake
UNIVERSITY

Announcements

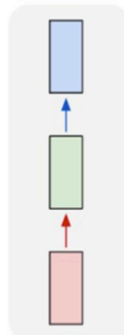
- **Project#2**
  - Released and due on **12/14 (Saturday) by 11:59pm**
- **Quiz#3**
  - Will be released later this week

Recap: mappings of different tasks

- we input one training/testing example and make one prediction.



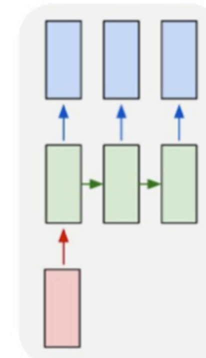
one to one



- we input one training/test example, and output many predictions

Image caption generation

one to many

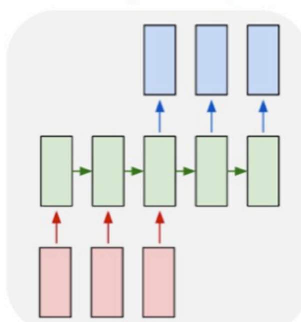


- we input multiple things, and make multiple predictions from it
- the input and output size do not need to be the same length

Machine Translation (translating from one language to another)

"Hello my name is" --> "Hola me llamo"

many to many

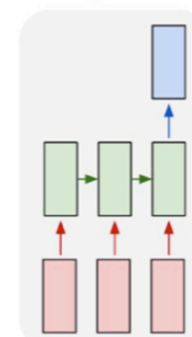


- we input multiple things, and make one prediction from it

Review (X)	Rating (Y)
"This movie is fantastic! I really like it because it is so good!"	★★★★☆
"Not to my taste, will skip and watch another movie"	★★☆☆☆
"This movie really sucks! Can I get my money back please?"	★☆☆☆☆

Product review prediction

many to one



Advantages and disadvantages of various RNNs

Vanilla RNN Advantages

- they can also handle inputs of varying lengths.

Vanilla RNN disadvantages

- short term memory
- suffers from the **vanishing gradient** problem:
  - forgets what is seen in longer sequences
  - this problem gets worse with the more layers the network has

LSTM Advantages

- solves vanishing gradient problem
- can capture both the short and long term patterns of a sequence.

LSTM disadvantages

- because LSTMs add complexity, they also are computationally more expensive, leading to longer training times.

GRU Advantages

- solves vanishing gradient problem
- less computationally expensive than LSTMs, which makes them faster to train

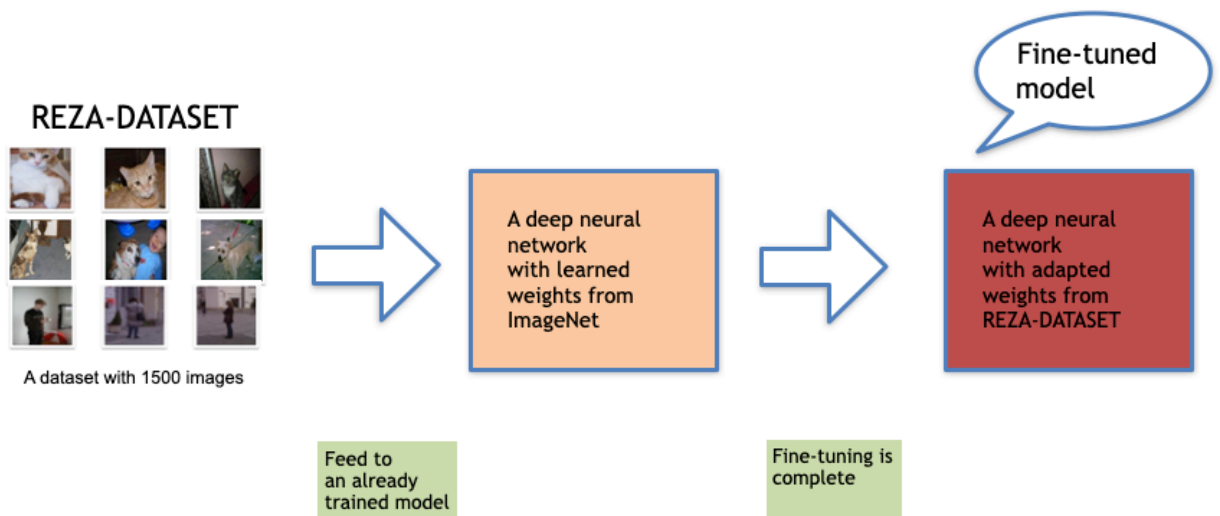
GRU disadvantages

- do not have a separate hidden and cell state, so they might not be able to consider observations as far into the past as the LSTM

Caveats in LSTM

- **LSTMs** with added complexity, they also are computationally more expensive, leading to **longer training times**

- **Transfer learning** (on a new dataset with limited samples) never worked with LSTM
  - we did transfer learning in CNN when we fine-tuned **a pretrained AlexNet** on a new datasets such as BCDP
  - we quickly achieved excellent accuracy of over 90% within a few epochs of training
  - you will also fine-tune two other CNNs for your Project 2: i) **a pretrained VGG** and ii) **a pretrained ResNet**



Transformers

Today's agenda

- In-depth exploration of the caveats of vanilla RNN and LSTM
    - Vanishing gradient in vanilla RNN
    - Transfer learning is not possible with LSTM (or RNN)

- Transformers
    - Transfer learning is possible
    - New type of network architecture

Transformers

- In 2017, a new mechanism is introduced for context learning called **attention mechanism**
  - more precisely, **self-attention**
- It takes less time to train advantage
- Transfer learning on a new task is possible advantage
- In subsequent years, it revolutionized the field of AI

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

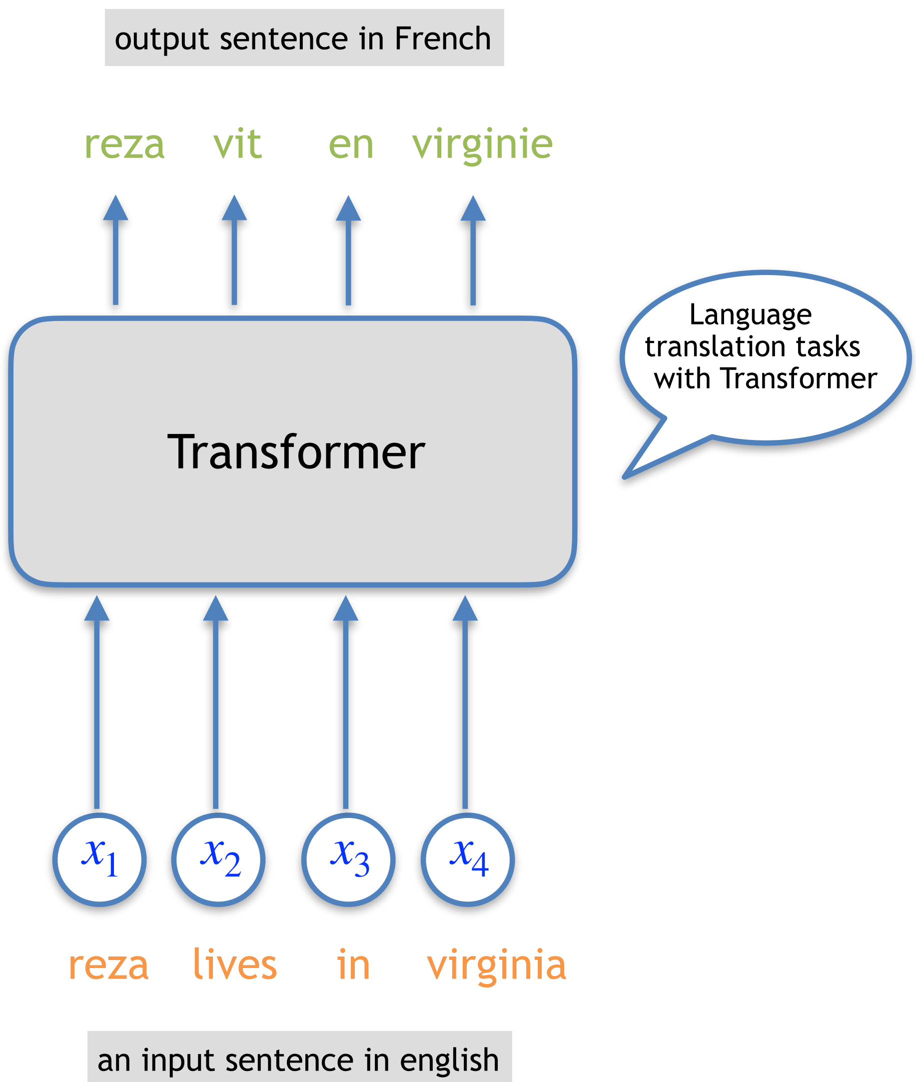
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

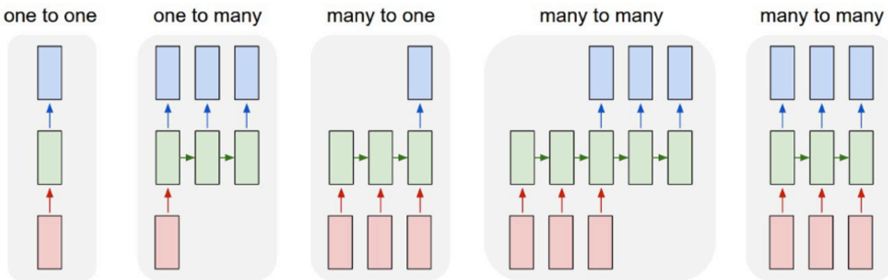
The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, 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 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

Attention is all you need - NeurIPS’2017

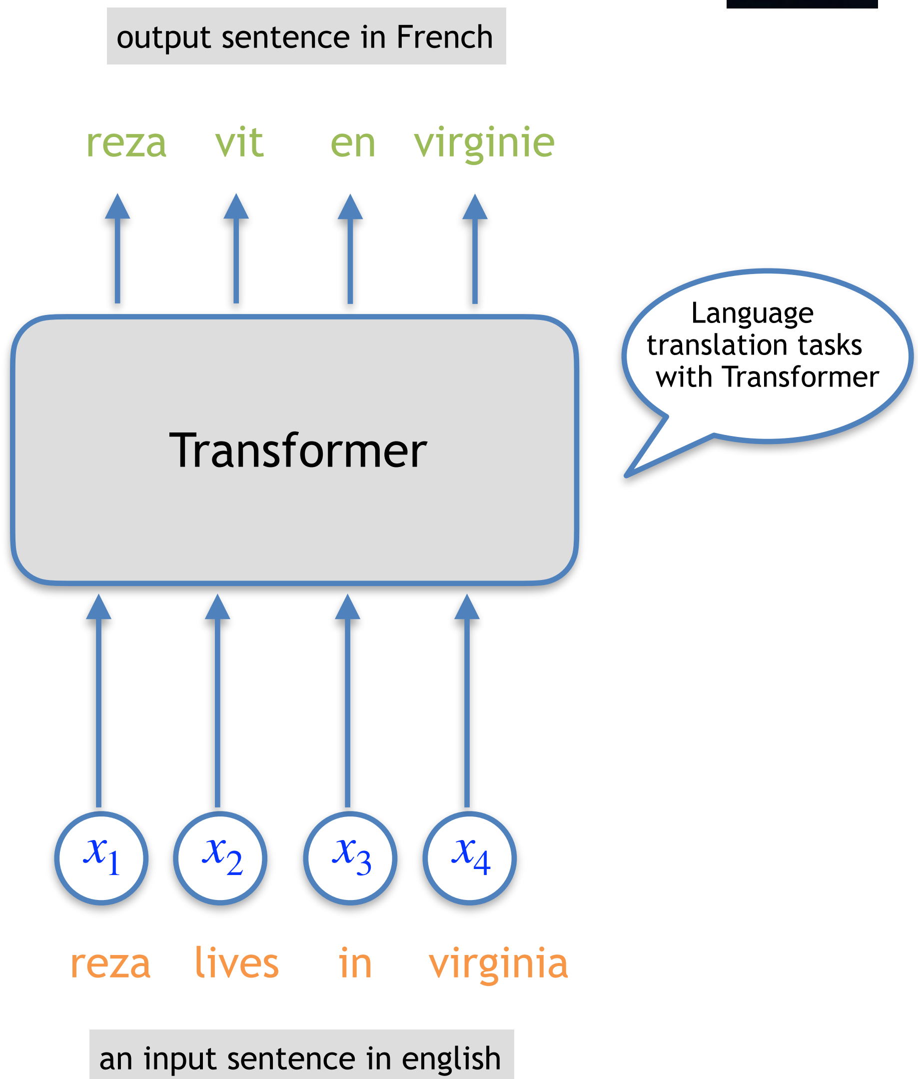
Transformers



Transformers

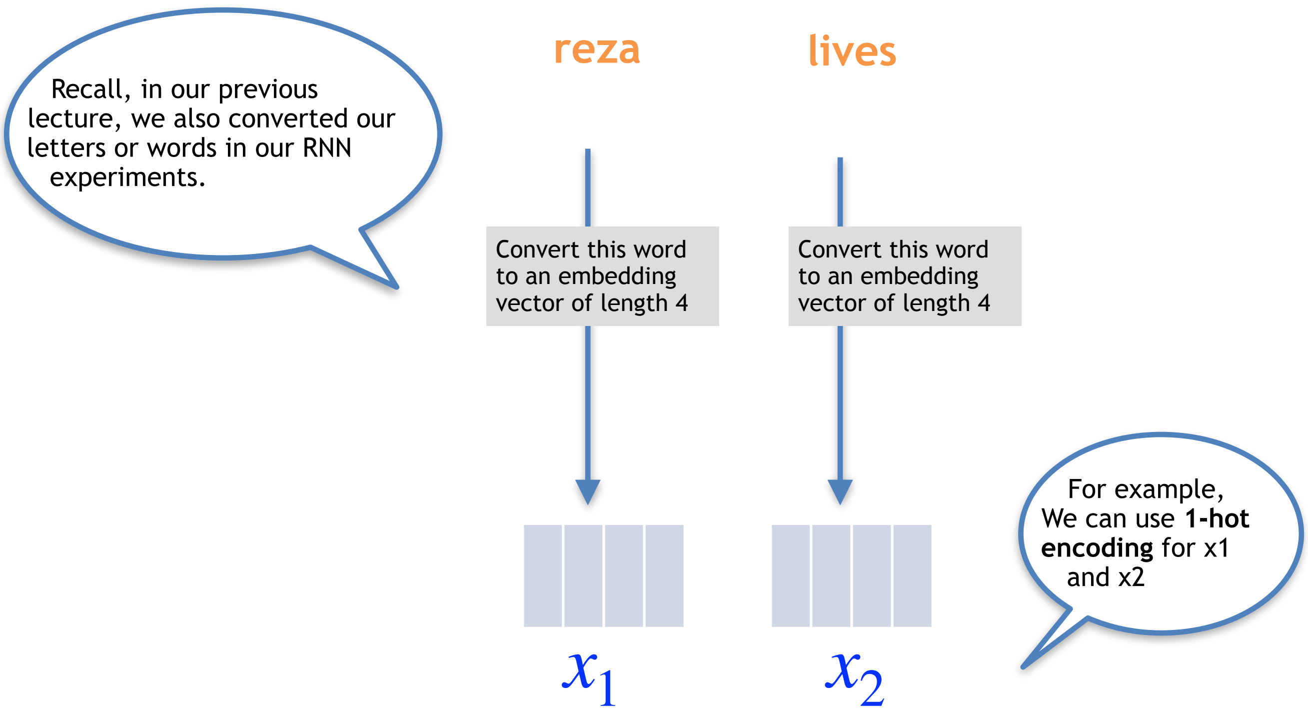


What type of task is this?



Attention

- Let's find out how to calculate the attention mechanism in a toy example
- Let's calculate attention with first two words of our sentence: “reza lives”



Attention

Blame 1410 lines (1410 loc) · 81.6 KB   Code 55% faster with GitHub Copilot

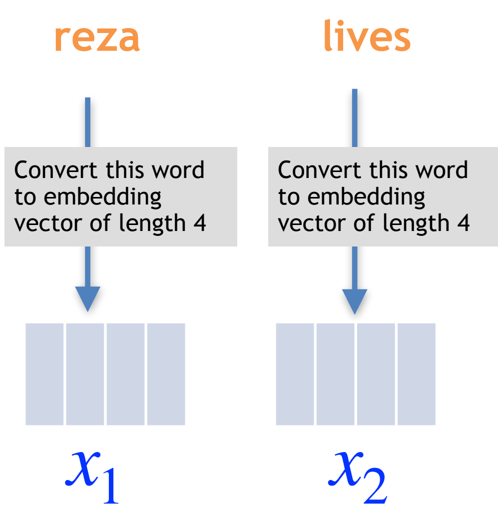
In [6]:

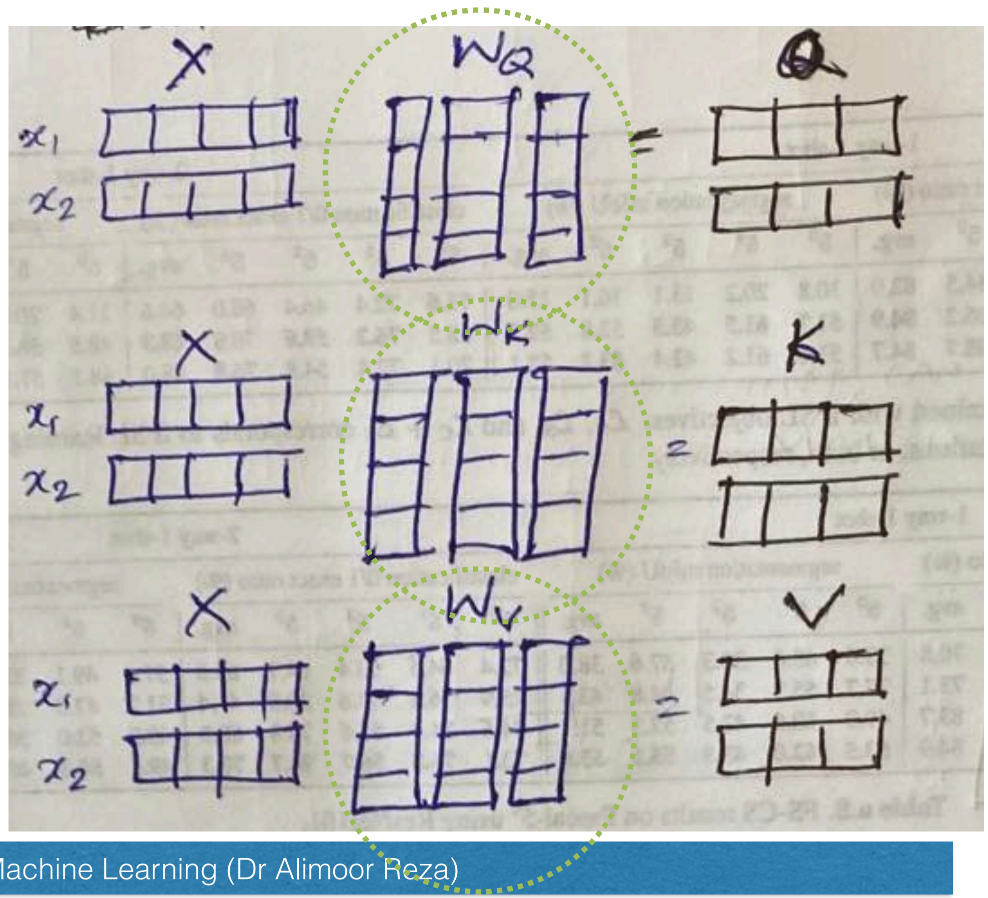
```python
# Step 1: create a mapping between the characters in our voculary to a set of numeric indices
def convert_vocab_to_index(vocab):
  vocab_to_index_dict = {}
  for index, char in enumerate(vocab):
    vocab_to_index_dict[char] = index
  return vocab_to_index_dict

def convert_index_to_vocab(vocab):
  index_to_vocab_dict = {}
  for index, char in enumerate(vocab):
    index_to_vocab_dict[index] = char
  return index_to_vocab_dict


vocab_to_index_dict = convert_vocab_to_index(text_vocab)
index_to_vocab_dict = convert_index_to_vocab(text_vocab)


# Step 2: convert the text_data to numeric numbers using the above conversion method (this mapped data will be us
text_data_numeric_values = np.zeros(text_data_size)
for i in range(text_data_size):
  cur_character = text_data[i].lower()
  text_data_numeric_values[i] = vocab_to_index_dict[cur_character]

# Step 3: visualize the first few characters in our text_data
for i in range(6):
  print("character: ", text_data[i].lower(), " encoded as: ", text_data_numeric_values[i])
```

```
character:  f  encoded as:  18.0
character:  i  encoded as:  21.0
character:  r  encoded as:  30.0
character:  s  encoded as:  31.0
character:  t  encoded as:  32.0
```

Attention

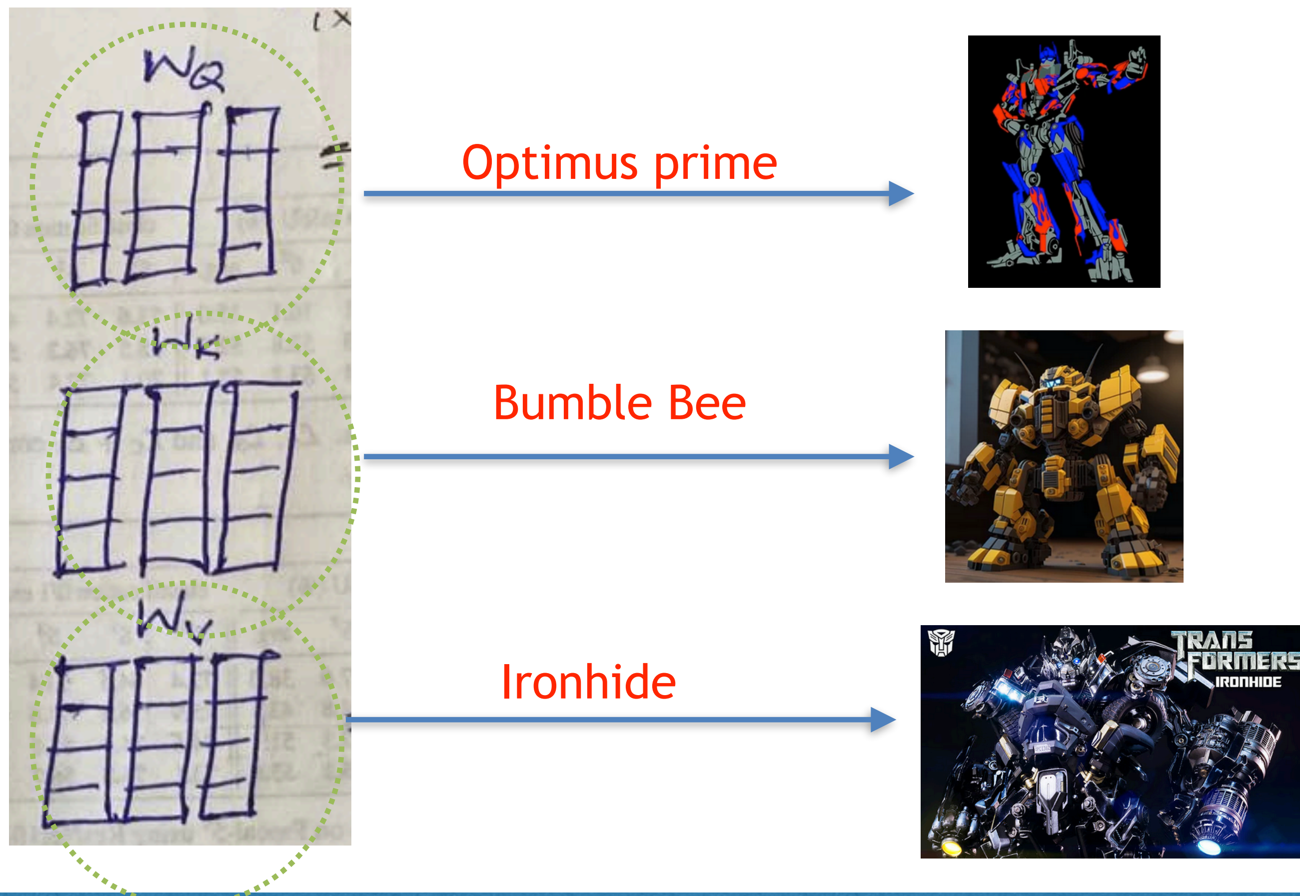
- It calculates three new matrices Q, K, and V with the help of three weight matrices W_Q, W_K, and W_V
- These three matrices (W_Q, W_K, and W_V) are learned during training





Attention

- It calculates three new matrices Q, K, and V with the help of three weight matrices W_Q, W_K, and W_V
- These three matrices (W_Q, W_K, and W_V) are learned during training



So why do we need this complicated attention? Recall our "context" discussion

- Consider a language model trying to predict the next word based on the previous ones. Let's predict the last word to this sequence:

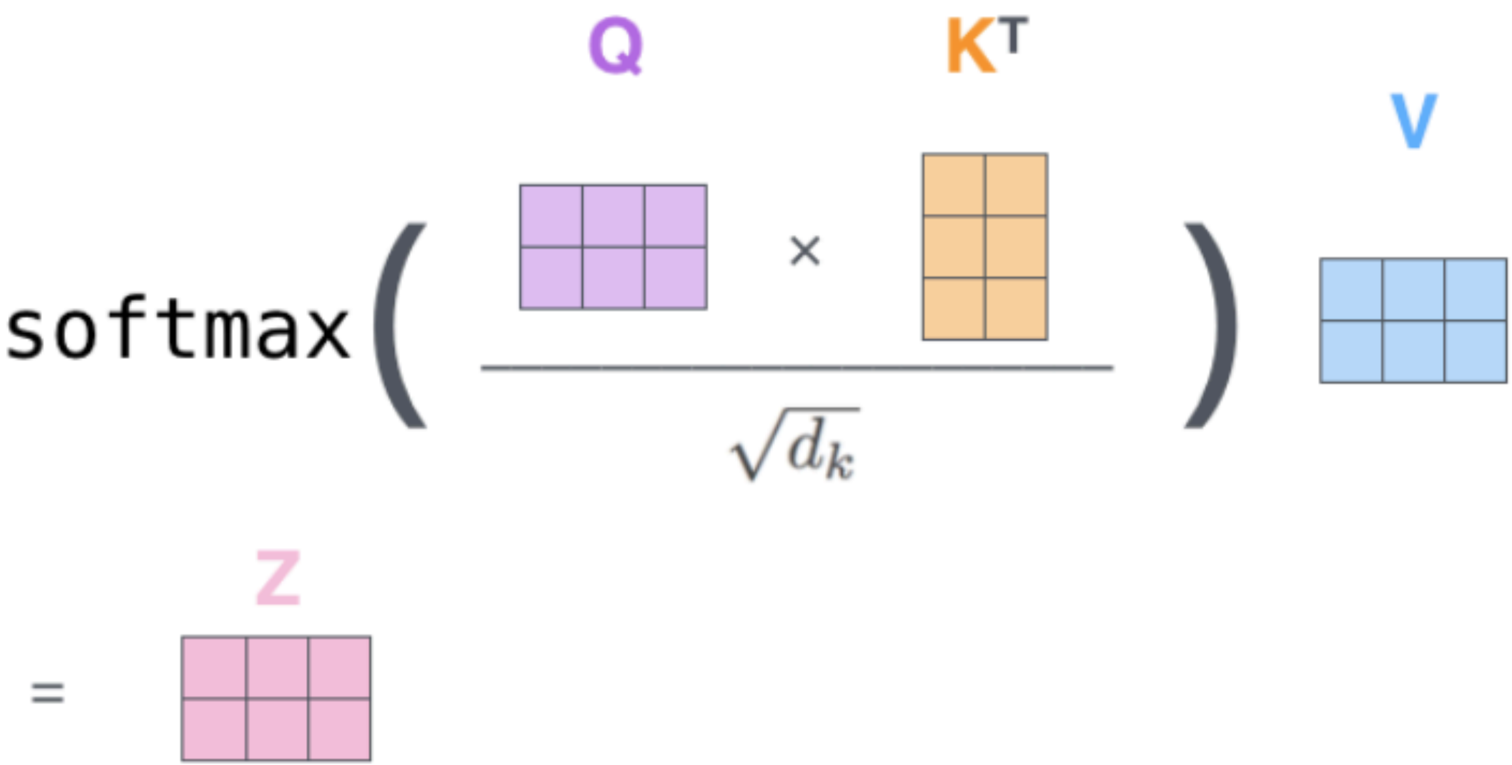
  - `"The clouds are in the sky"`

- We can guess from recent information that it will be the name of a language; we need the context of **France** to make a prediction.

  - `"I grew up in France, I speak fluent french"`

Attention

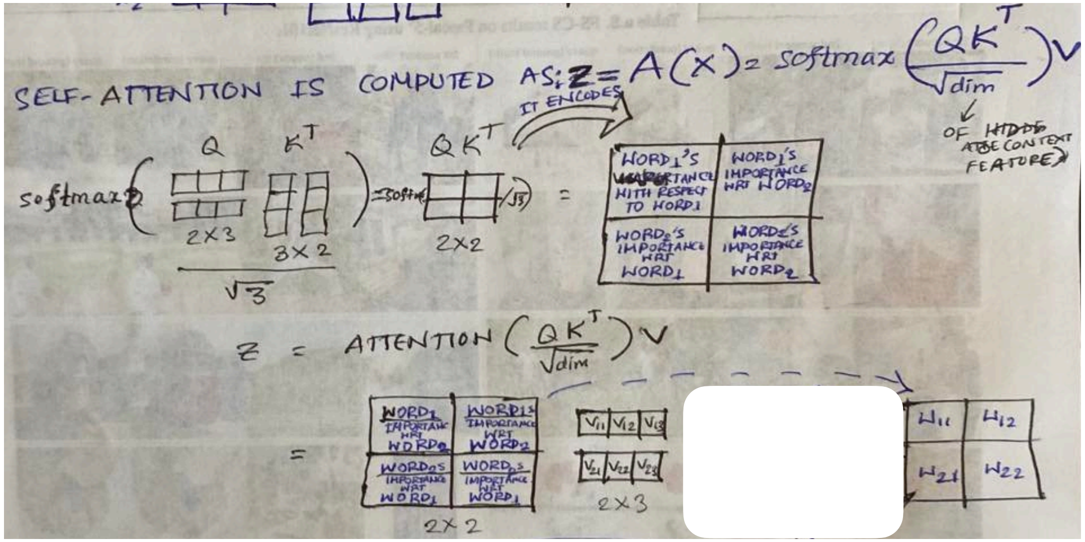
- Finally, attention is calculated using Q, K, and V matrices using the following equation:



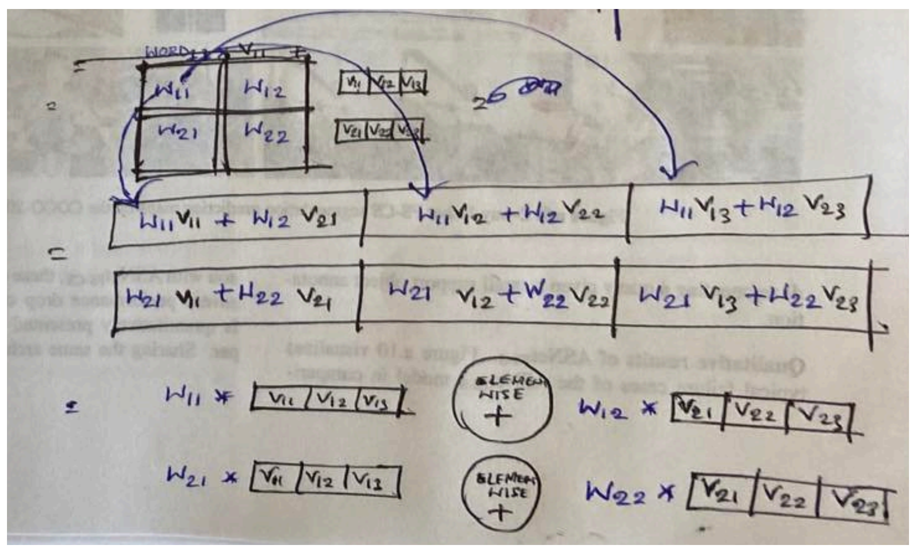
[Reference: Illustrated Transformer]

Attention

My hand-notes



My hand-notes



[Reference: Illustrated Transformer]()

Going Back to the Transformer Idea

- This new mechanism for context learning, called the **attention mechanism** is only one part—of course, the central one.

- There are other components. Let's examine those.

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

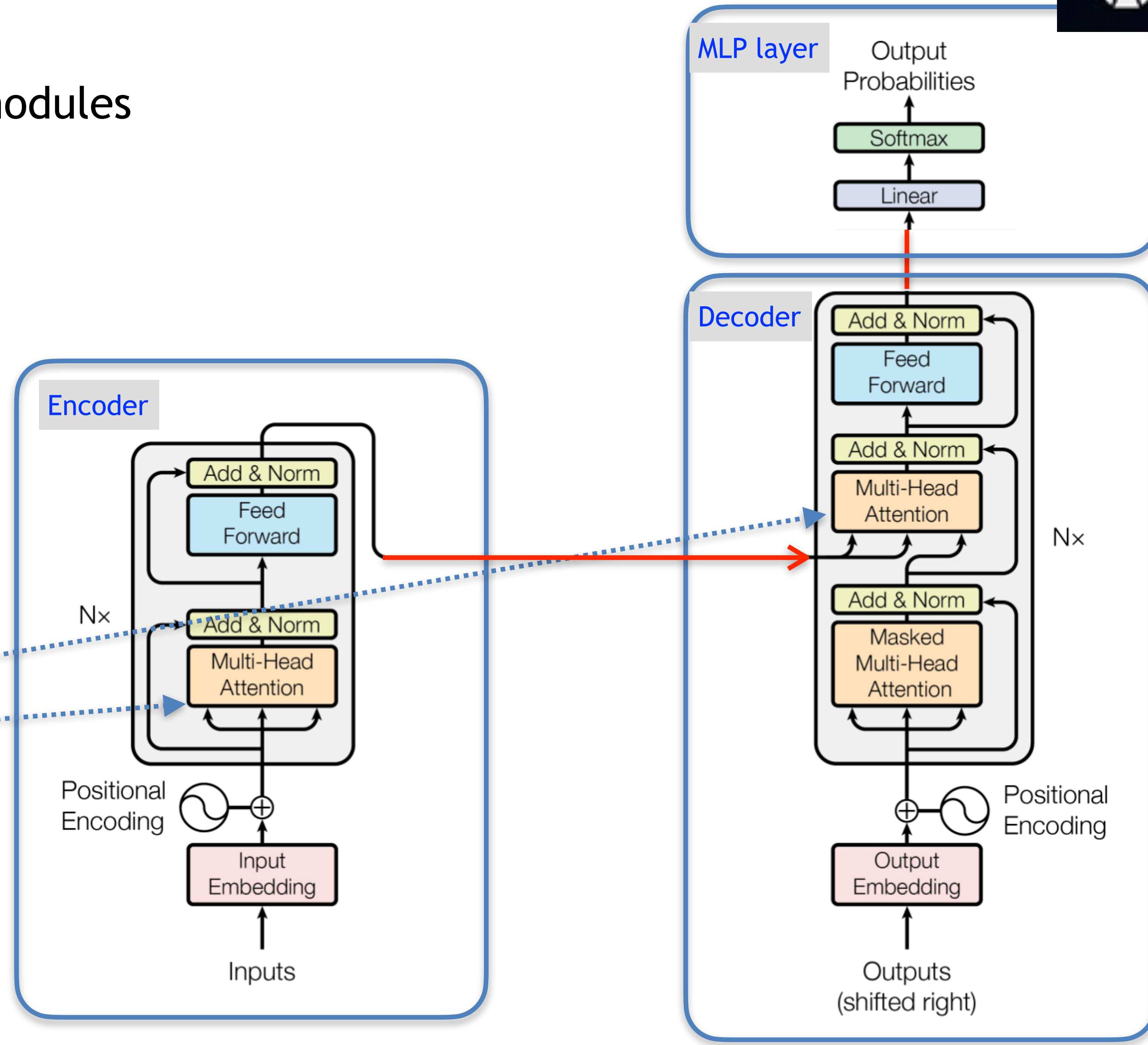
Abstract

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 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

Attention is all you need - NeurIPS'2017

Transformers

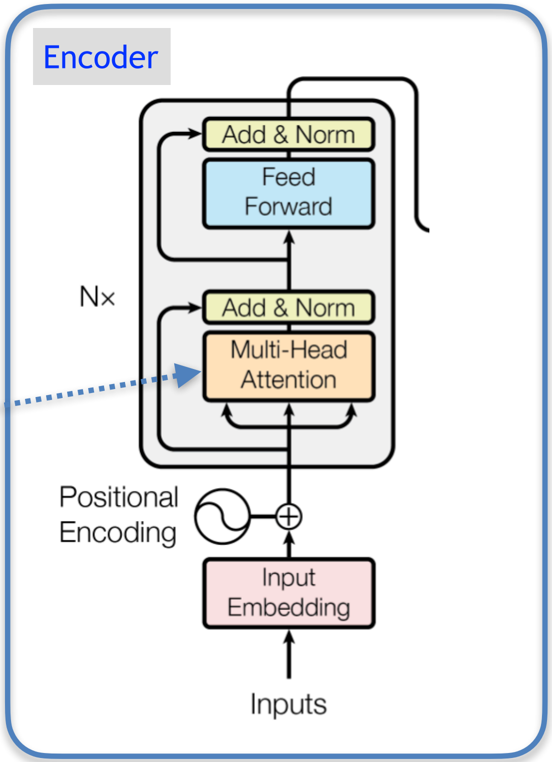
- It has three modules
  - Encoder
  - Decoder
  - MLP layer



The driving force behind transformer is attention mechanism

Transformers: Encoder

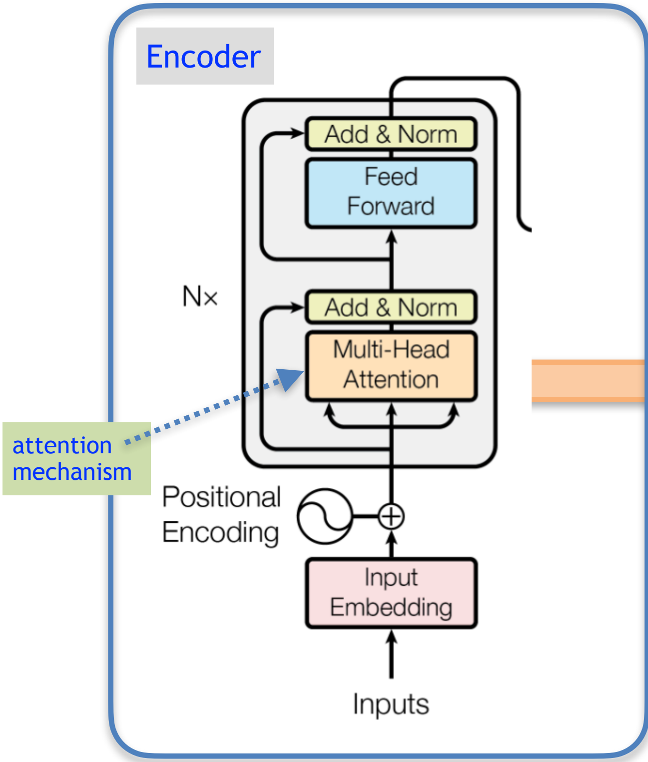
- Lets focus on the encoder to understand what is this attention mechanism



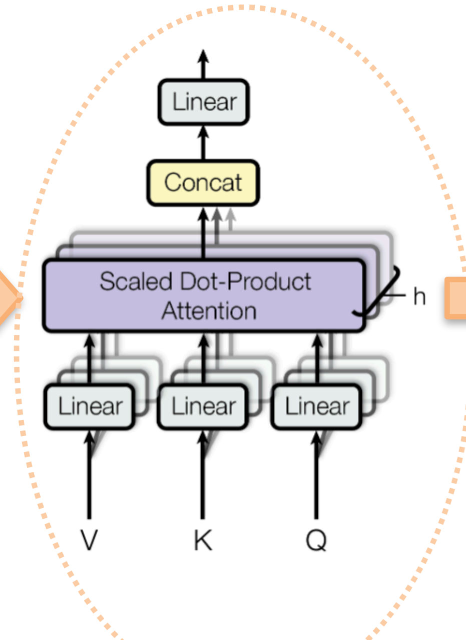
The driving force behind transformer is attention mechanism

Transformers: Encoder

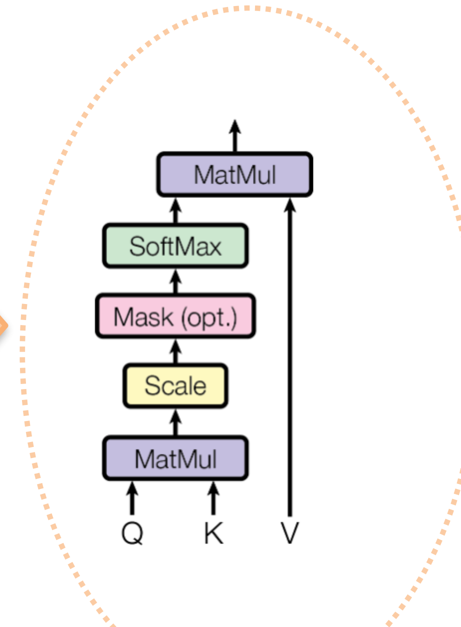
- Lets focus on the encoder to understand what is this attention mechanism



Encoder has a **multi-headed attention**



A multi-headed attention has several **scaled dot-product attention**



A single scaled dot-product attention has sequential matrix matrix multiplications in terms of Query (Q), Key (K), and Value (V)

Transformers

Today’s agenda

- In-depth exploration of the caveats of vanilla RNN and LSTM
    - Vanishing gradient in vanilla RNN
    - Transfer learning is not possible with LSTM (or RNN)

- Transformers
    - Transfer learning is possible
    - New type of network architecture