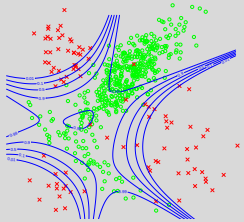
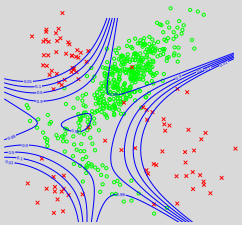


Natural Language Processing and Transformers

Lecture *Machine Learning* vom 29-31.3.2023

Felix Becker
(material in collaboration with Lars Gabriel and Mario Stanke)
Institut für Mathematik und Informatik
Universität Greifswald

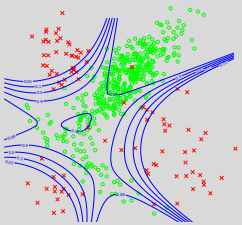
Background image AI-generated by Midjourney



What is Natural Language Processing (NLP)?

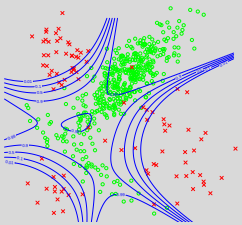
Process and analyze large amounts of natural human language data using computer programs (in the modern context: deep learning models).

The goal is to understand the contents of text, in particular understand the context of a word in its surrounding sentence(s).



Some NLP problems

- Translation:
  - Input: "I love you."
  - Output: "Je t'aime."
- Text generation (example output generated by GPT-3):
  - Input: "Write a joke about machine learning."
  - Output: "Why did the machine learning model break up with its training data? Because it found a better fit!"
- Question answering (related to text generation):
  - Input: "Do I need my car in New York City?"
  - Output: "No. Please keep your car at home."
- Language understanding
- Text summary
- Speech recognition

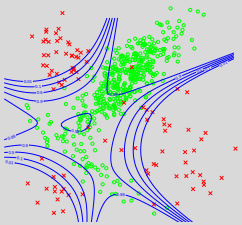


Token

The inputs to NLP models are sequences of tokens. A token is a building block of natural language that can be:

- A word
- Part of a composed word: **countrymen** → country, men
- Part of a contraction: **aren't** → are, not
- An equivalence class of multiple words:
  {anti-discriminatory, antidiscriminatory}
- A specific indicator for the model e.g. EOS (end of sequence)
  or MASK

For example, the tokenizer of the Distilbert model knows 30,522 tokens, whereas the Oxford English Dictionary has about 172,000 words.



Embedding

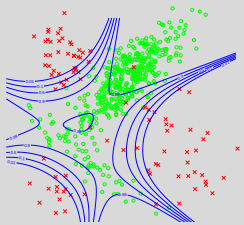
A high dimensional vector representing a specific sequence position (and potentially its context).

Sequence

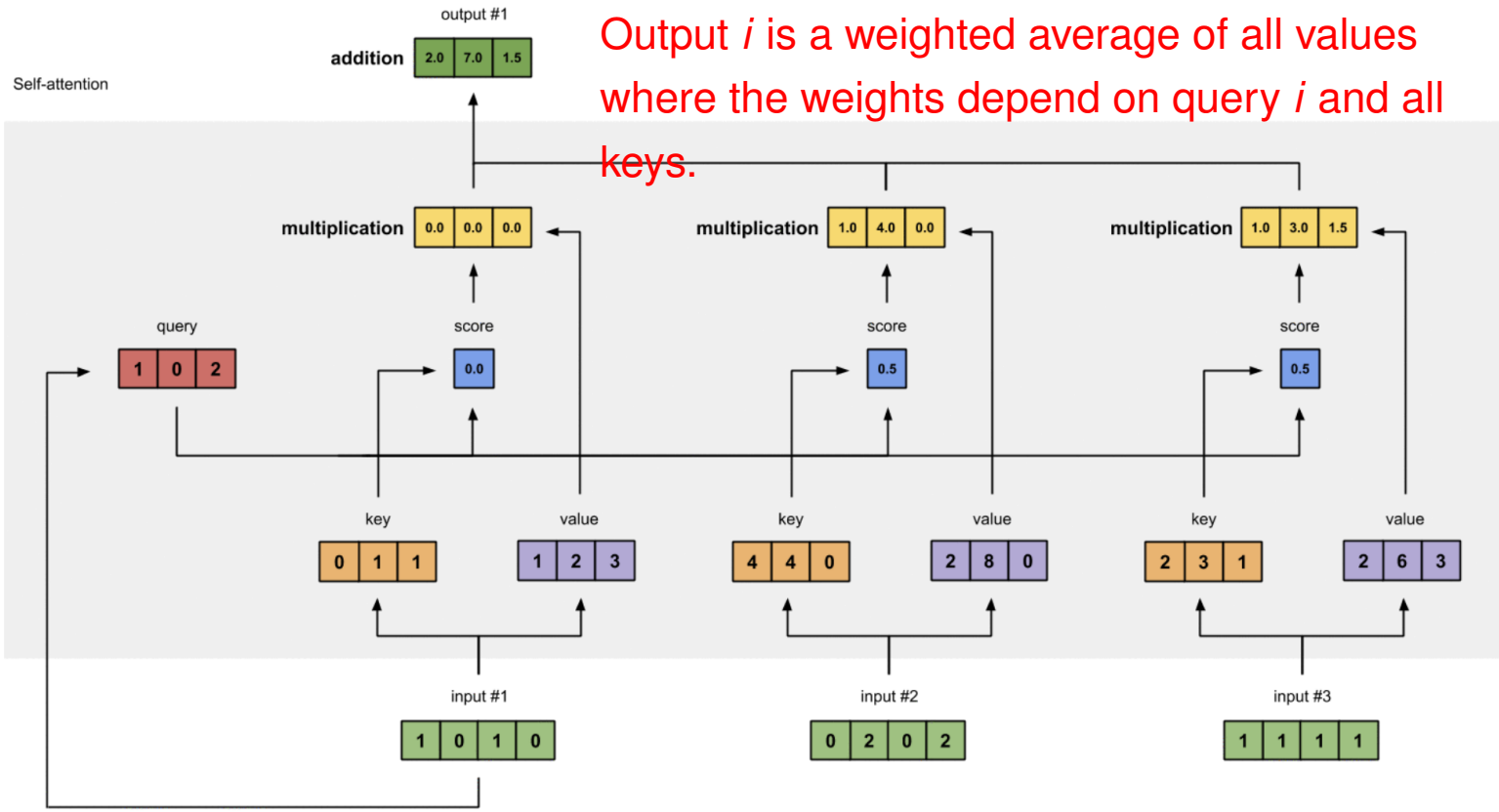
A series of tokens or embeddings in a spatial or temporal relationship.

Sequence-to-sequence model

Maps an input sequence of tokens to an output sequence. The sequences are not required to have the same length and no 1:1 token correspondence is assumed. Predicting a correct output length is responsibility of the model.



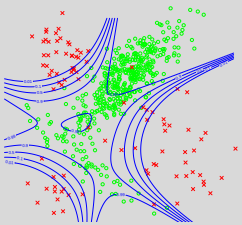
Self-attention



Output *i* is a weighted average of all values where the weights depend on query *i* and all keys.

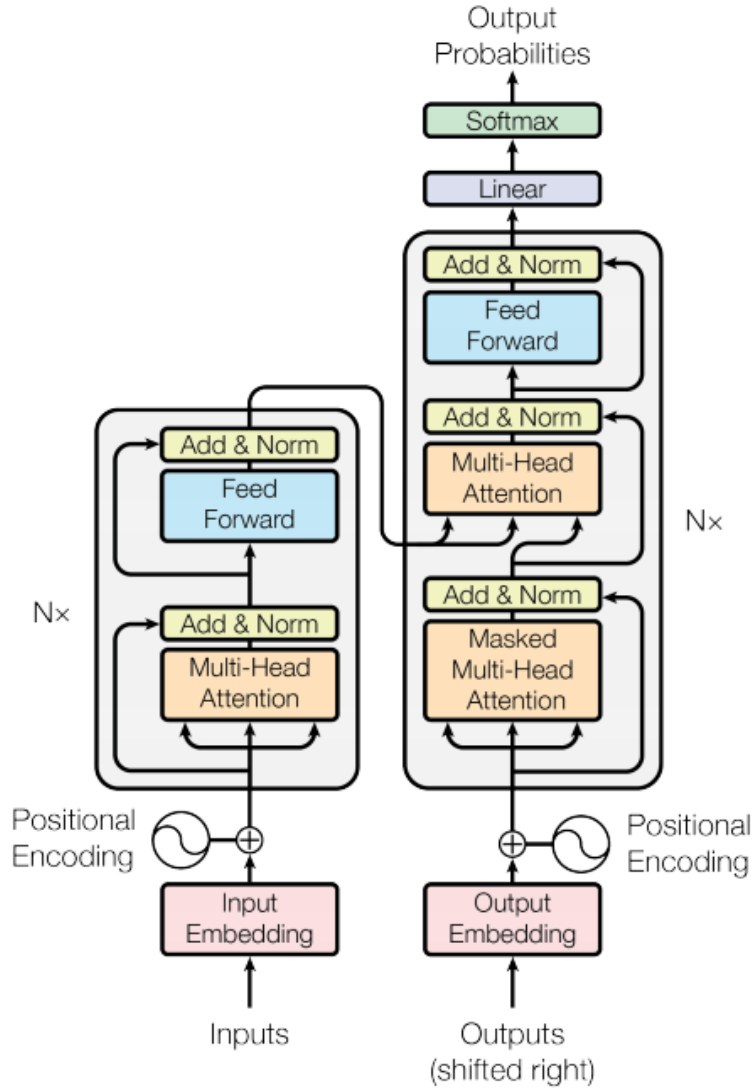
towardsdatascience.com/illustrated-self-attention-2d627e33b20a

The scores in the figure are rounded to one digit after the comma.



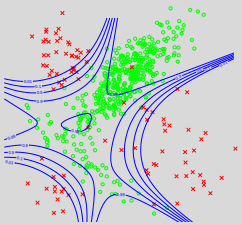
The transformer architecture

- Based on self-attention and cross-attention (attention between input- and output sequence)



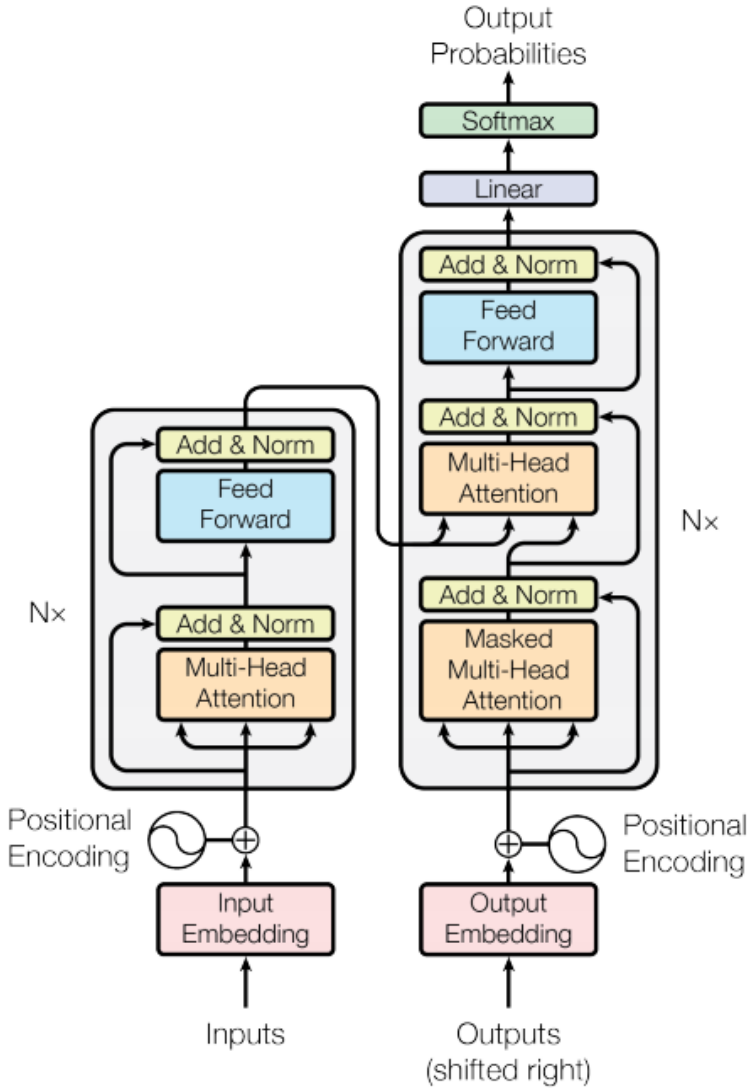
1

[1] Vaswani et al., Attention is all you need, 2017



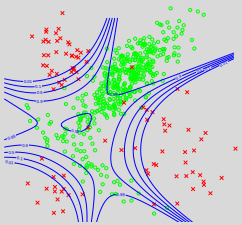
The transformer architecture

- Based on self-attention and cross-attention (attention between input- and output sequence)
- Introduced in 2017 (Attention is all you need, Vaswani et al.)



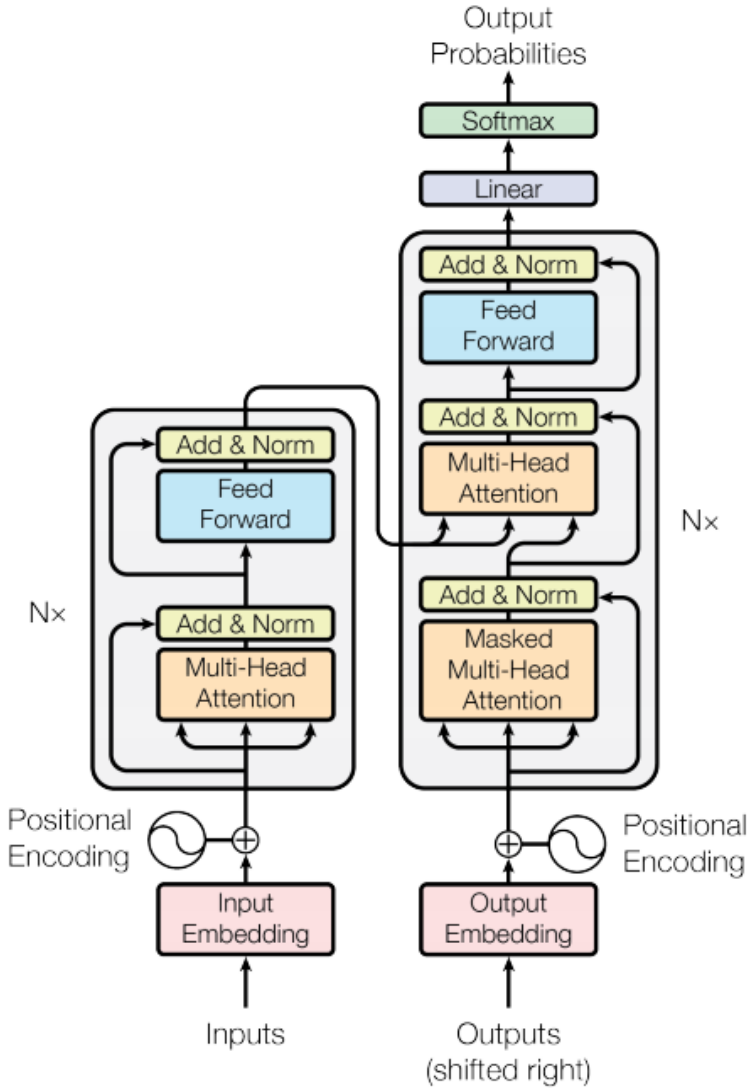
1

[1] Vaswani et al., Attention is all you need, 2017



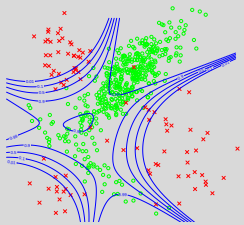
The transformer architecture

- Based on self-attention and cross-attention (attention between input- and output sequence)
- Introduced in 2017 (Attention is all you need, Vaswani et al.)
- A transformer can consist of an encoder (left) and a decoder (right)



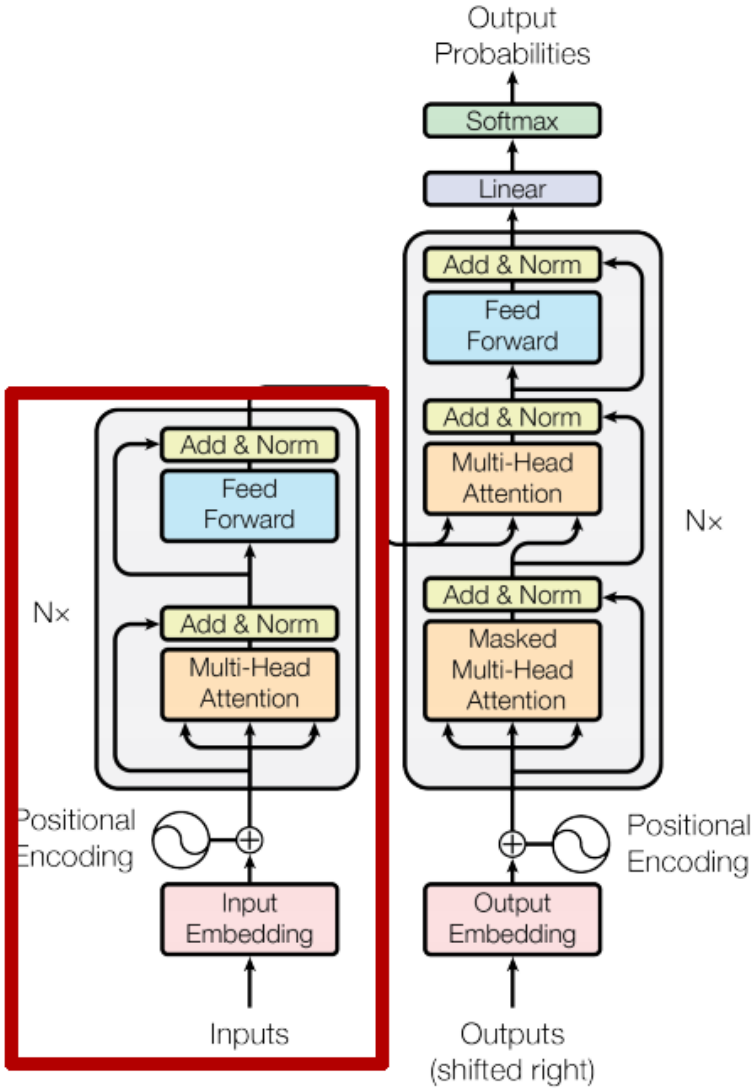
1

[1] Vaswani et al., Attention is all you need, 2017



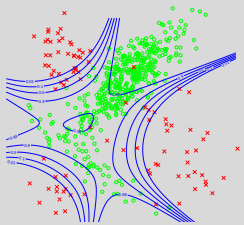
The transformer architecture

- Based on self-attention and cross-attention (attention between input- and output sequence)
- Introduced in 2017 (Attention is all you need, Vaswani et al.)
- A transformer can consist of an encoder (left) and a decoder (right)
- Here, we will focus on the encoder



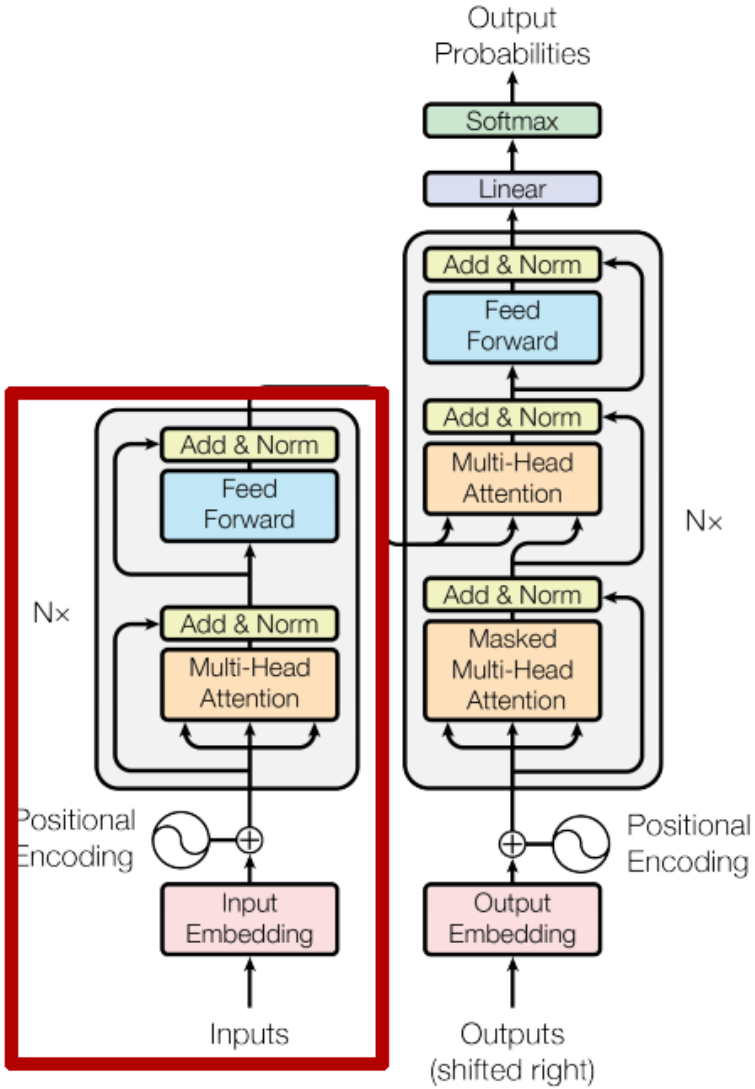
[1]

[1] Vaswani et al., Attention is all you need, 2017 (modified)



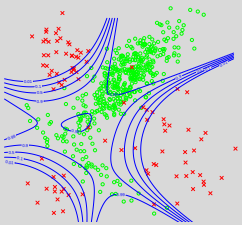
The transformer architecture

- Based on self-attention and cross-attention (attention between input- and output sequence)
- Introduced in 2017 (Attention is all you need, Vaswani et al.)
- A transformer can consist of an encoder (left) and a decoder (right)
- Here, we will focus on the encoder
- The encoders task is to learn a model of the input language (e.g. english or the "language" of protein sequences)



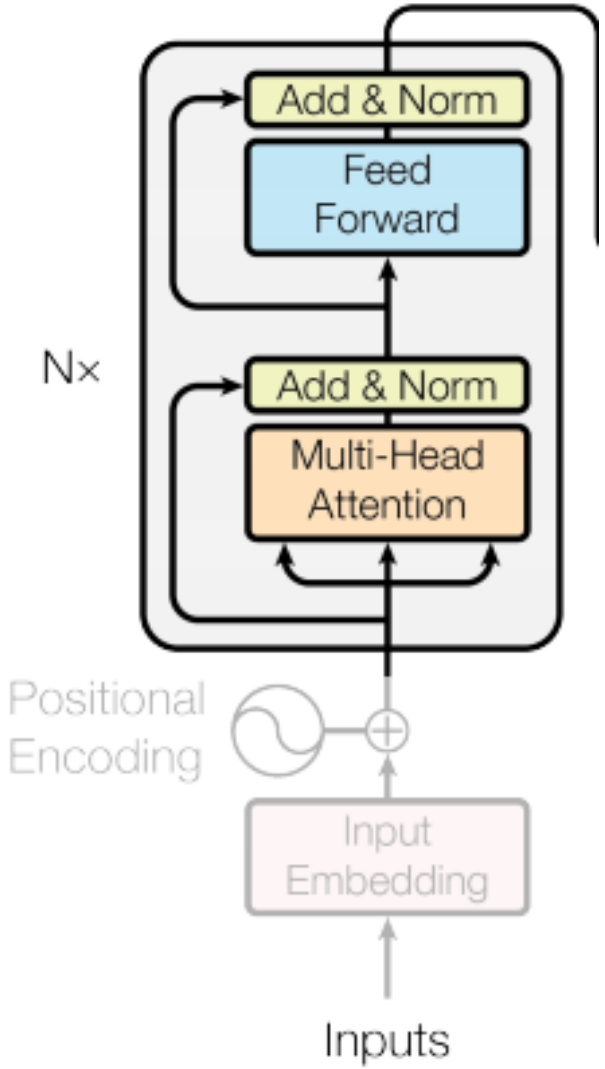
1

[1] Vaswani et al., Attention is all you need, 2017 (modified)



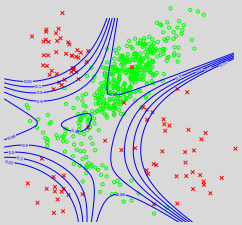
The transformer encoder

- Consumes a tensor of (embedded) input sequences and outputs a tensor with the same shape and updated embeddings



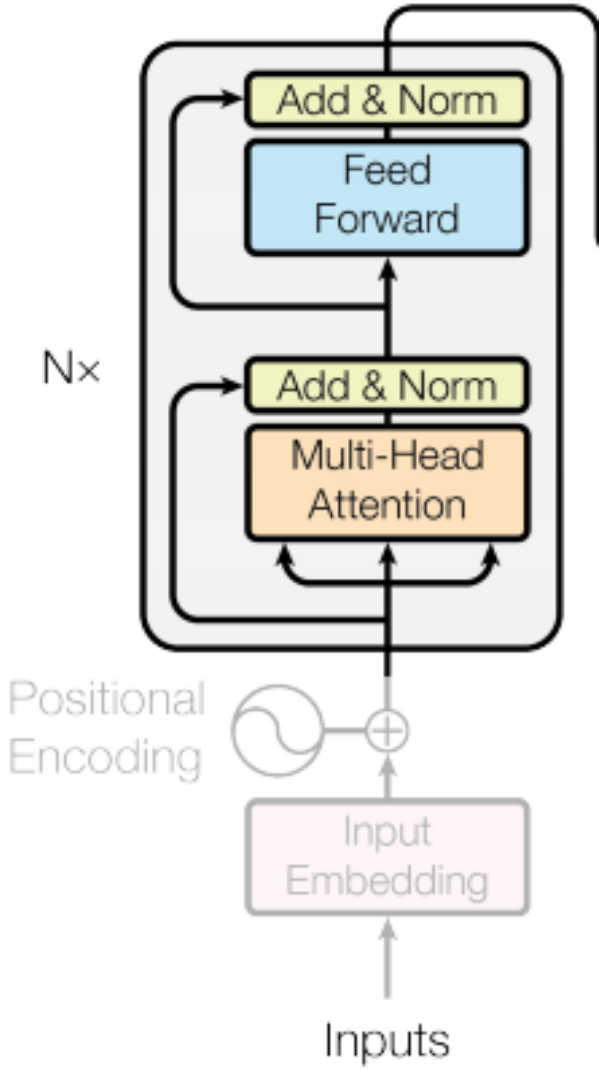
2

[2] Vaswani et al., Attention is all you need, 2017 (modified)



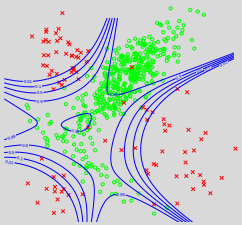
The transformer encoder

- Consumes a tensor of (embedded) input sequences and outputs a tensor with the same shape and updated embeddings
- First step: Self attention makes each embedding (in parallel) aware of all other embeddings



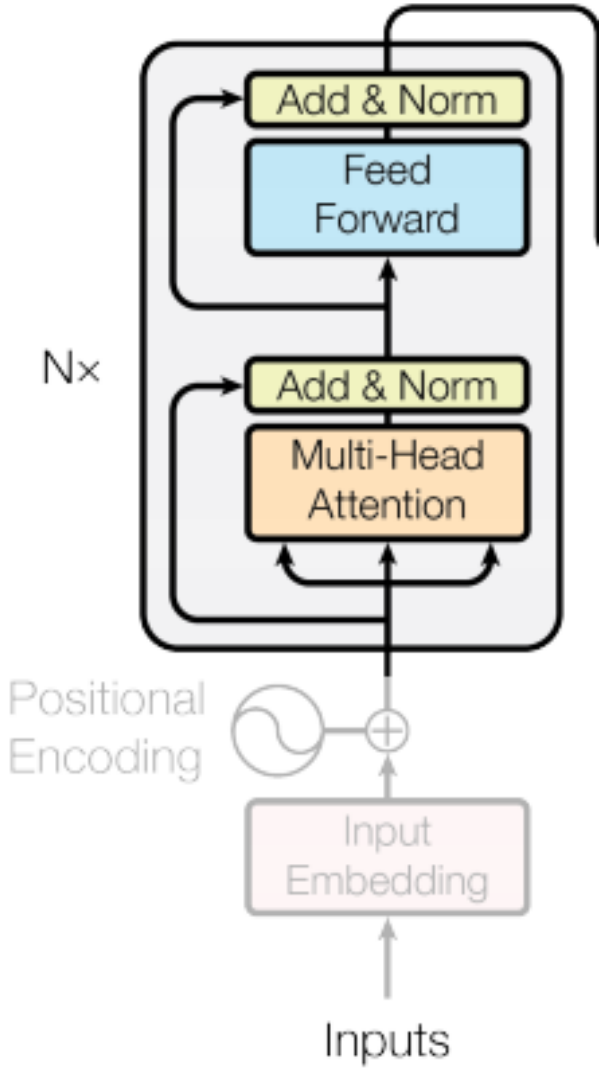
[2]

[2] Vaswani et al., Attention is all you need, 2017 (modified)



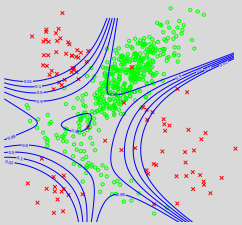
The transformer encoder

- Consumes a tensor of (embedded) input sequences and outputs a tensor with the same shape and updated embeddings
- First step: Self attention makes each embedding (in parallel) aware of all other embeddings
- Second step: Update the embeddings independently with a neural network usually larger than the embeddings itself



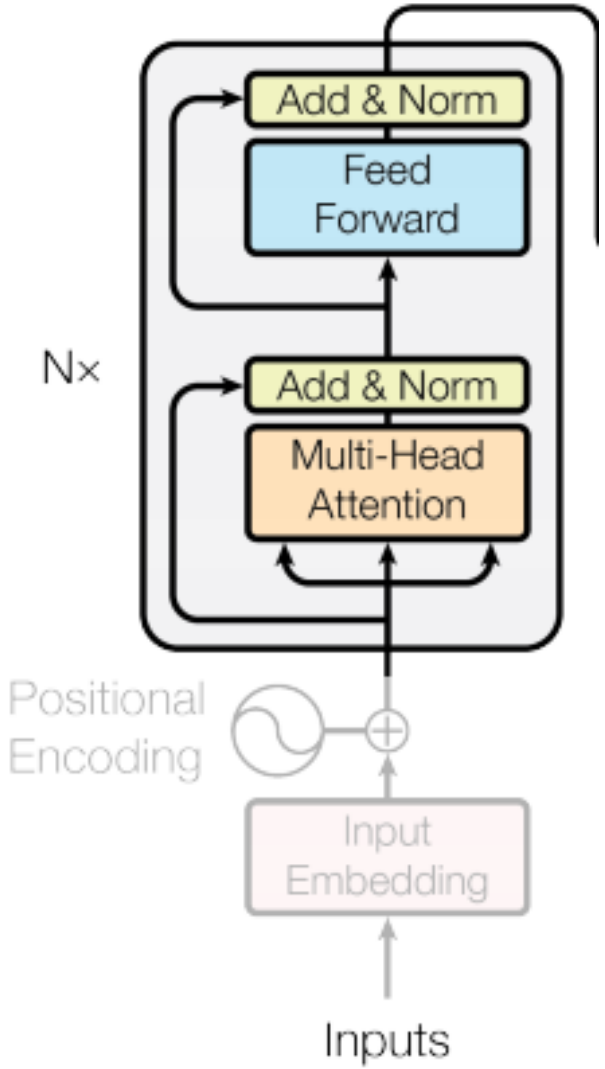
2

[2] Vaswani et al., Attention is all you need, 2017 (modified)



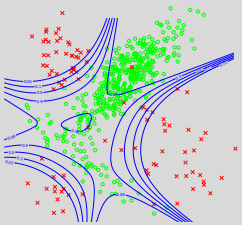
The transformer encoder

- Consumes a tensor of (embedded) input sequences and outputs a tensor with the same shape and updated embeddings
- First step: Self attention makes each embedding (in parallel) aware of all other embeddings
- Second step: Update the embeddings independently with a neural network usually larger than the embeddings itself
- These steps can be repeated several times



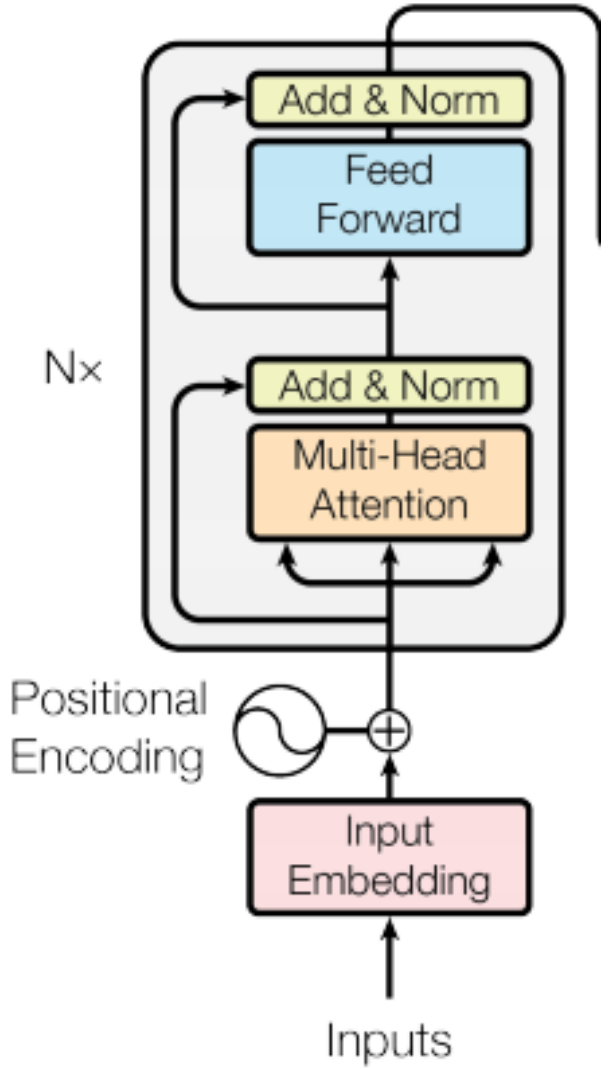
2

[2] Vaswani et al., Attention is all you need, 2017 (modified)



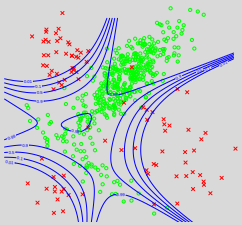
The transformer encoder

- **Input Embedding** replaces tokens with high dimensional embeddings



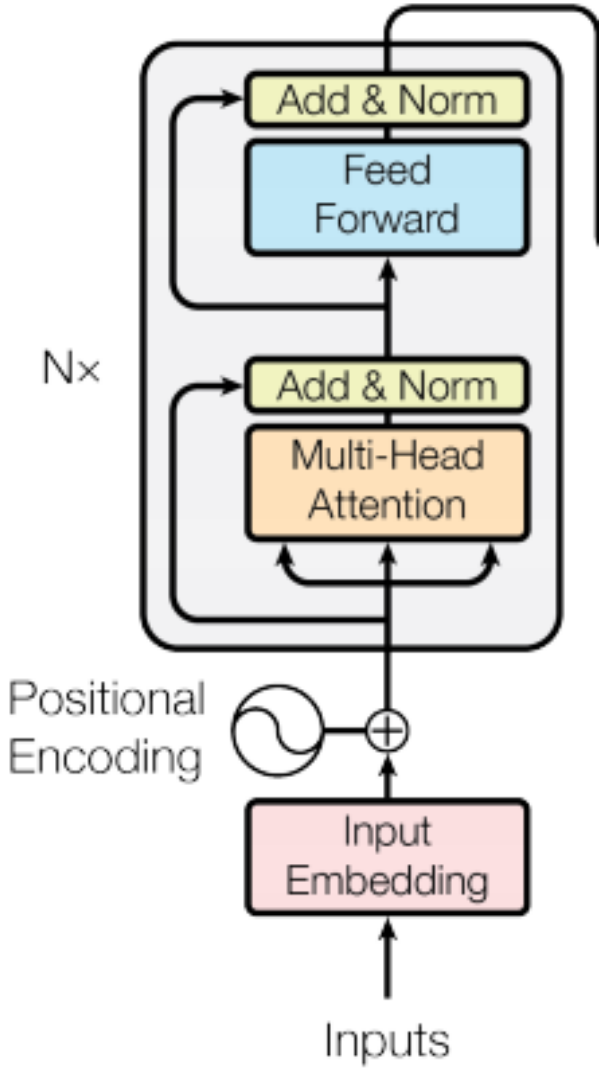
3

[3] Vaswani et al., Attention is all you need, 2017 (modified)



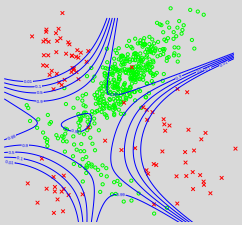
The transformer encoder

- **Input Embedding** replaces tokens with high dimensional embeddings
- A **Positional Encoding** adds spatial/temporal information (without it the transformer is invariant to the ordering of the input tokens)



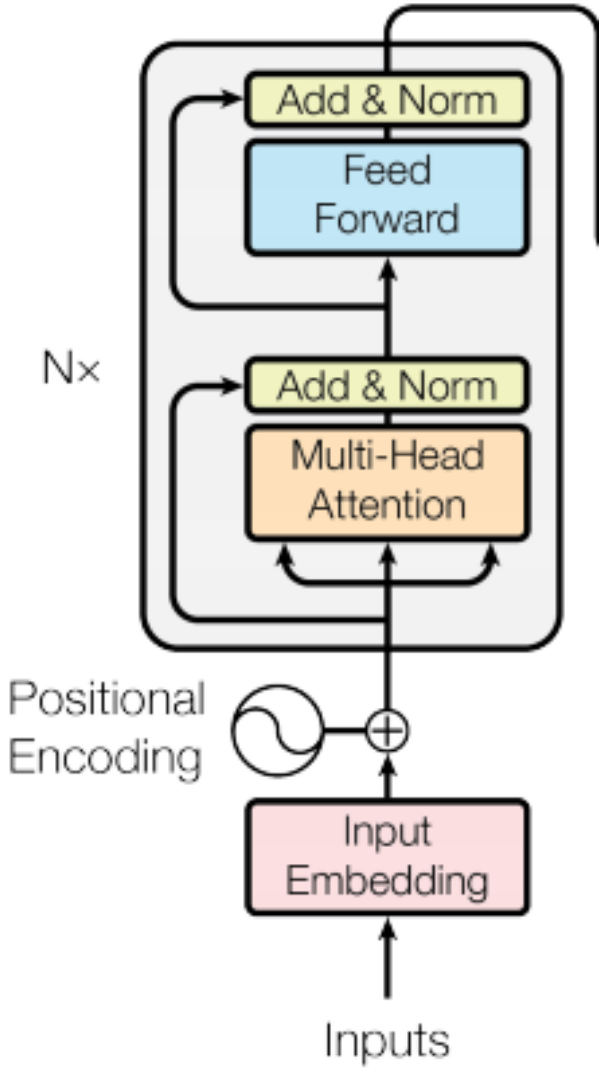
3

[3] Vaswani et al., Attention is all you need, 2017 (modified)



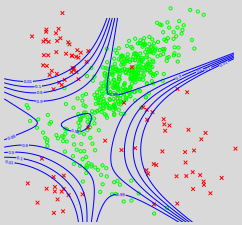
The transformer encoder

- **Input Embedding** replaces tokens with high dimensional embeddings
- A **Positional Encoding** adds spatial/temporal information (without it the transformer is invariant to the ordering of the input tokens)
- **Multi-Head Attention** is a more complicated form of self-attention, where multiple **heads** allow to attention to different subspaces of the sequence in parallel.



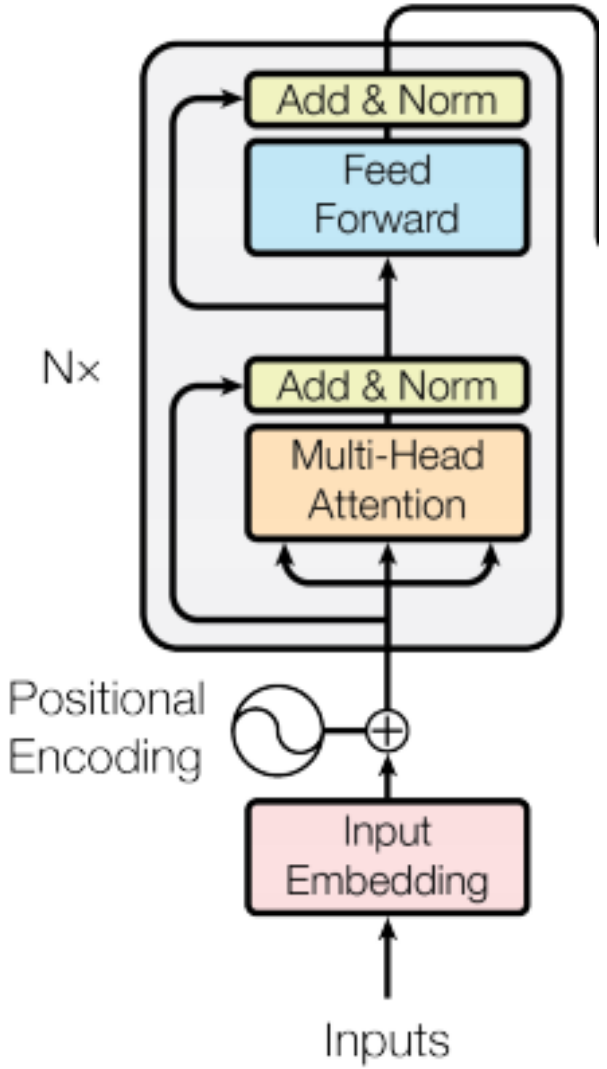
3

[3] Vaswani et al., Attention is all you need, 2017 (modified)



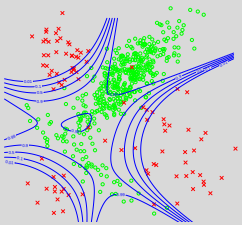
The transformer encoder

- **Input Embedding** replaces tokens with high dimensional embeddings
- A **Positional Encoding** adds spatial/temporal information (without it the transformer is invariant to the ordering of the input tokens)
- **Multi-Head Attention** is a more complicated form of self-attention, where multiple **heads** allow to attention to different subspaces of the sequence in parallel.
- **Feed Forward** is a neural network that is applied position-wise to the embedded sequences



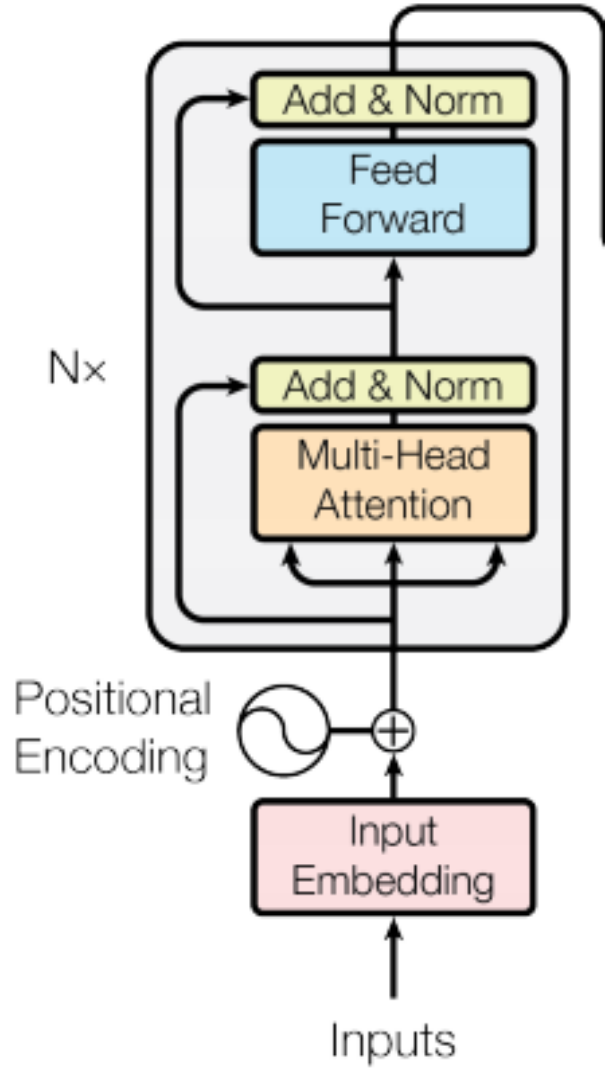
3

[3] Vaswani et al., Attention is all you need, 2017 (modified)



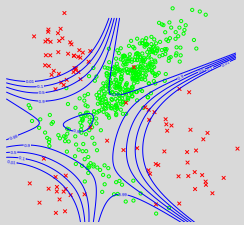
The transformer encoder

- **Input Embedding** replaces tokens with high dimensional embeddings
- A **Positional Encoding** adds spatial/temporal information (without it the transformer is invariant to the ordering of the input tokens)
- **Multi-Head Attention** is a more complicated form of self-attention, where multiple **heads** allow to attention to different subspaces of the sequence in parallel.
- **Feed Forward** is a neural network that is applied position-wise to the embedded sequences
- **Add and Norm** means we introduce so called **skip-connections** and **Layer Normalization** (details omitted)



3

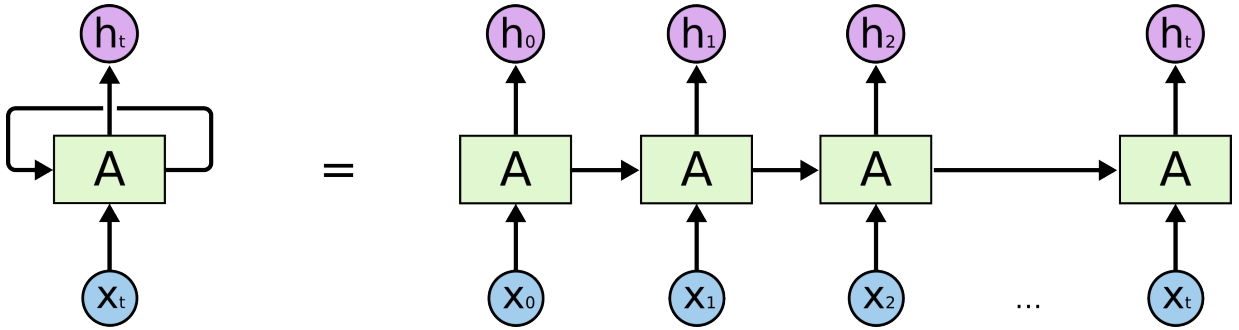
[3] Vaswani et al., Attention is all you need, 2017 (modified)



Comparison to recurrent neural networks (RNNs)

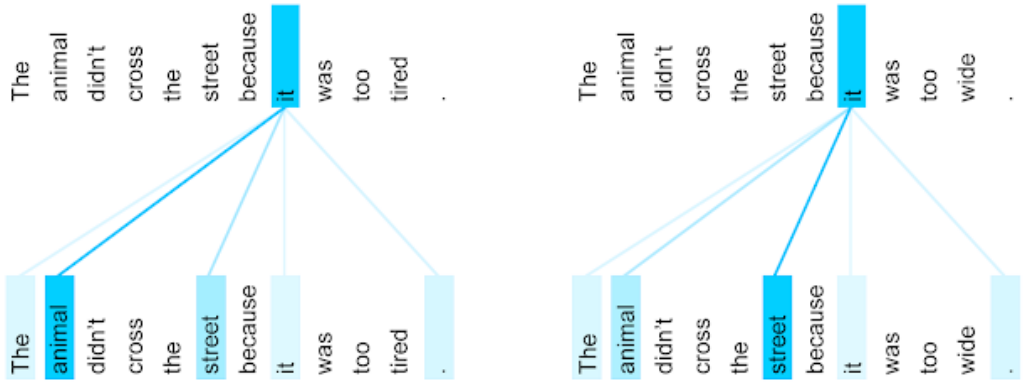
- RNNs lost popularity since the Transformer
- usually slower (because sequential, not parallel) and can not capture long-range interactions as good as attention can

RNN:



[4]

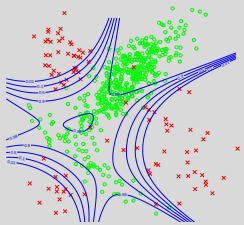
Attention:



[5]

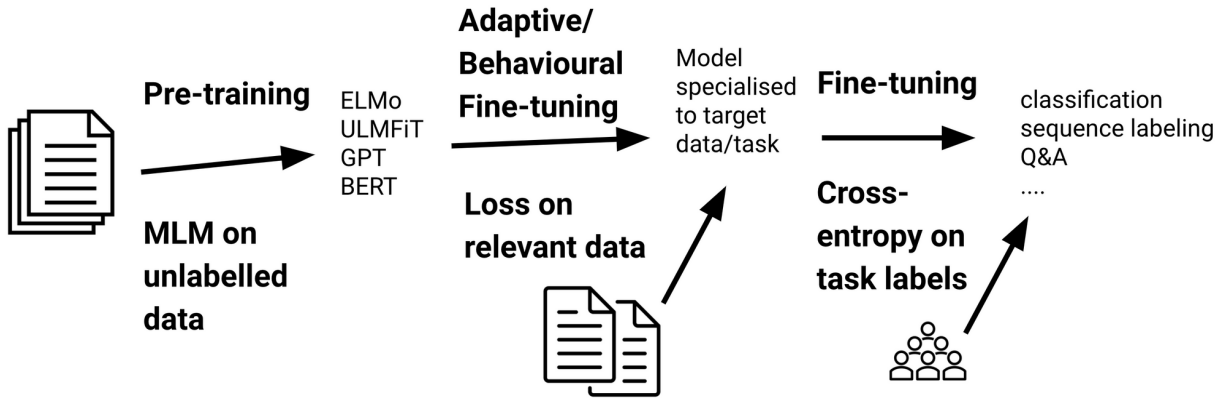
[4] colah.github.io/posts/2015-08-Understanding-LSTMs

[5] ai.googleblog.com/2017/08/transformer-novel-neural-network.html



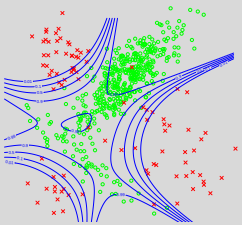
Fine-tuning

- Pre-train a large model on a general dataset (like Wikipedia) with Masked language modeling (MLM)
- Reuse the weights as initialization for further training on more specific datasets (e.g. movie reviews) to solve more specific tasks
- The fine-tuning step is usually much faster than the pre-training
- A single pre-trained model can be reused many times



6

[6] www.ruder.io/recent-advances-lm-fine-tuning/



Masked language modeling

Inputs

'>>> Review: This [MASK] is a great. The plot is very true to the [MASK] which is a [MASK] written by Mark Twain. The movie starts of with a scene where Hank [MASK] a song with a bunch of kids called "when you stub your toe on [MASK] moon" It reminds me of Sinatra's song High Hopes, it [MASK] fun and inspirational. The [MASK] is great throughout and my favorite song is [MASK] by the King, Hank (bing Crosby) and [MASK] "Saggy" Sagamore. OVerall a great family movie or even a [MASK] Date movie. This is a movie you can watch over and over [MASK]. The [MASK] played by Rhonda Fleming is gorgeous. I love this movie!! If you liked Danny Kaye [MASK] the Court Jester then you will definitely like this movie.'

Expected outputs

movie, book, classic, the, sings, is, Music, sung, Sir, great, great, again, princess, in

- Mask a percentage of the input tokens (i.e. the model receives a special MASK token instead of the actual token)
- Unsupervised (or sometimes called semi-supervised) training of a model with the goal to fill the gaps correctly using the cross-entropy loss function
- Can train models on large amounts of text from the internet without requiring any labeling