Evolution of NLP Models

8 August 2020

Jingli SHI

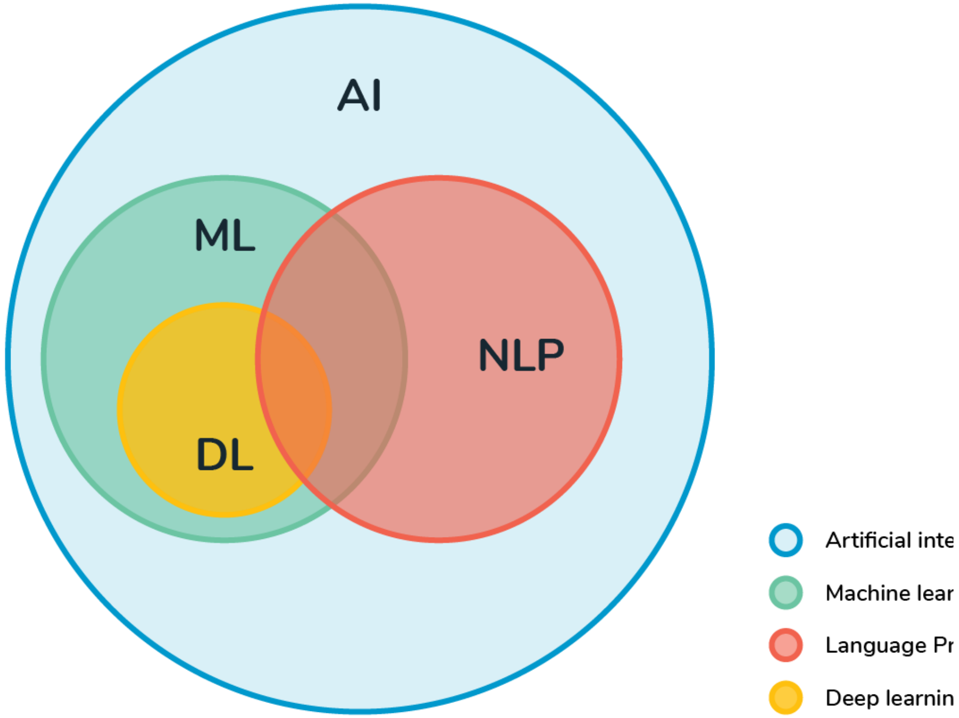
Content

- ❑ Generation #1
  - N-Gram
- ❑ Generation #2
  - Simple ML
- ❑ **Generation #3**
  - RNN / LSTM / GRU
- ❑ **Generation #4**
  - Seq2Seq (Encoder-Decoder)
- ❑ **Generation #5**
  - Transformer



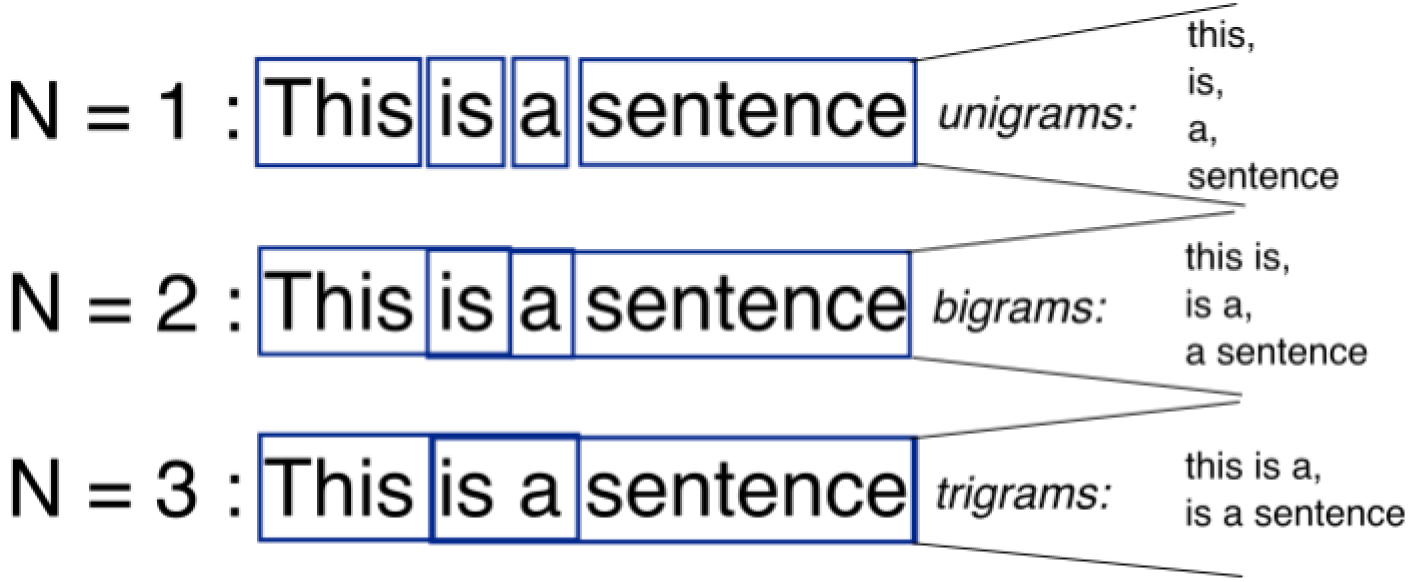
Background

NLP began in the 1950s as machine translation (MT). These early MT efforts were intended to aid in code-breaking during World War II. Developers hoped MT would translate Russian into English, but results were unsuccessful. Although the translations were not successful, these early stages of MT were necessary stepping stones on the way to more sophisticated technologies.



Background (AI vs ML vs DL vs NLP)

Gen #1 (*N* - Gram)

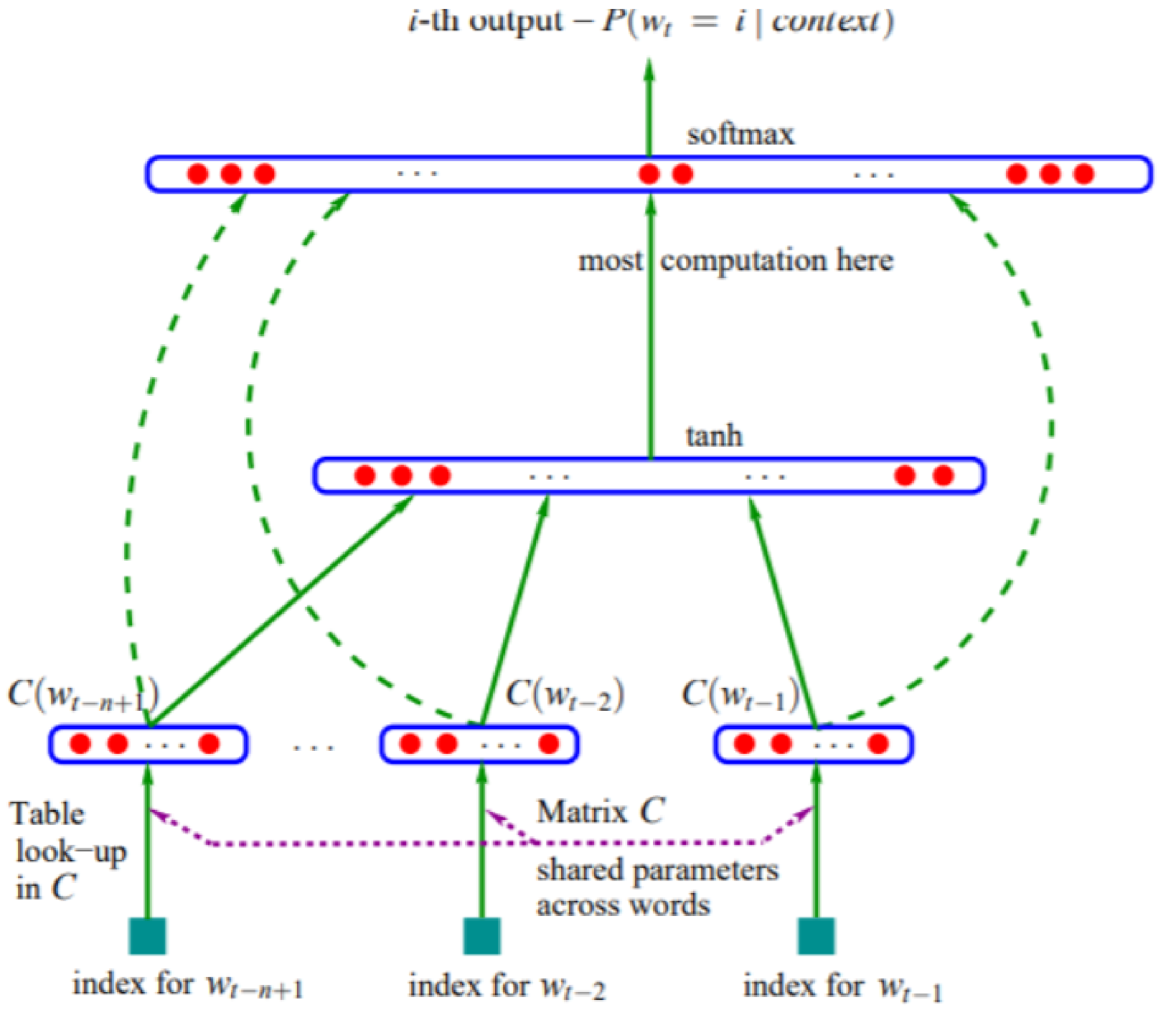


P(“This is a sentence”)
= ***P***(“This”)P(“is”|”This”)***P***(“a”|”is”, “This”) … ***P***(“sentence”|”is”, “a”)

Gen #1 (*N*-Gram)

- N-gram models almost does not know the complicated structure of human languages.
  - *The ship {sailed, sank, anchored, …}*
- N-gram models only know some low-level syntax.
  - *In study (noun) room*
  - *Study(verb) a language*

Gen #2 (Simple ML)



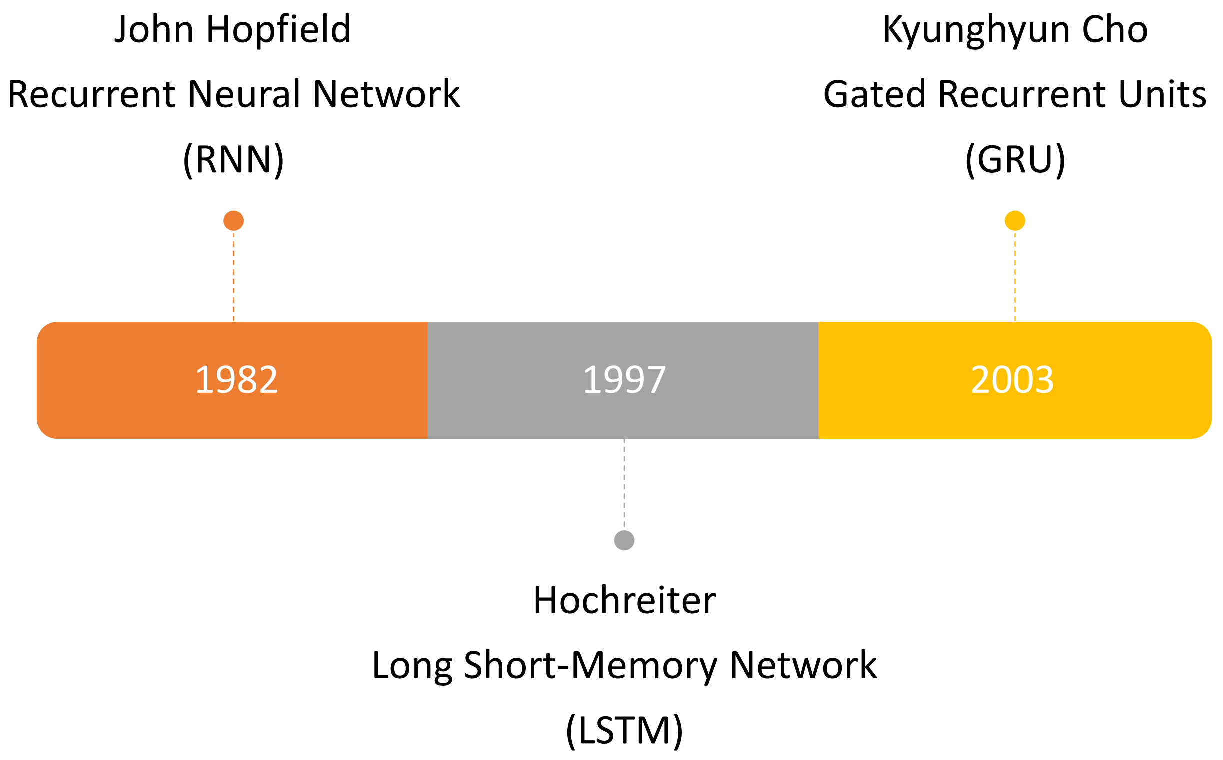
The first neural language model by Bengio et al. 2003 (source)

Simple Neural Network

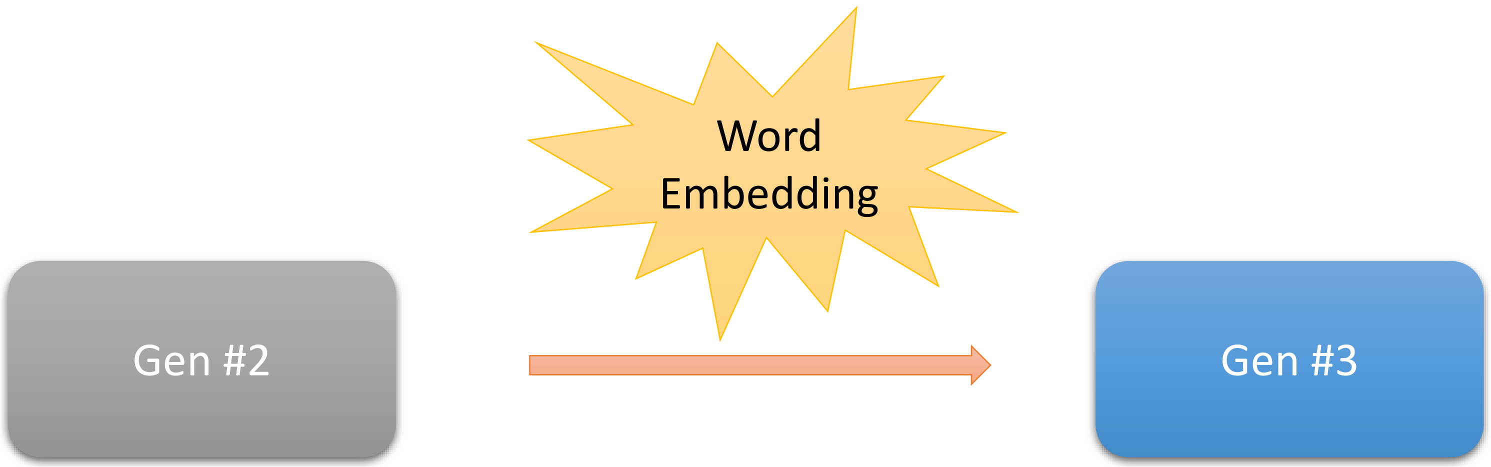
- Models like :
  - Linear Regression, Logistic Regression, Decision Tree, KNN, SVM, *et, al*.

- Works better on small data
- Financially and computationally cheap
- Easier to interpret

Gen #3 (RNN/LSTM/GRU)



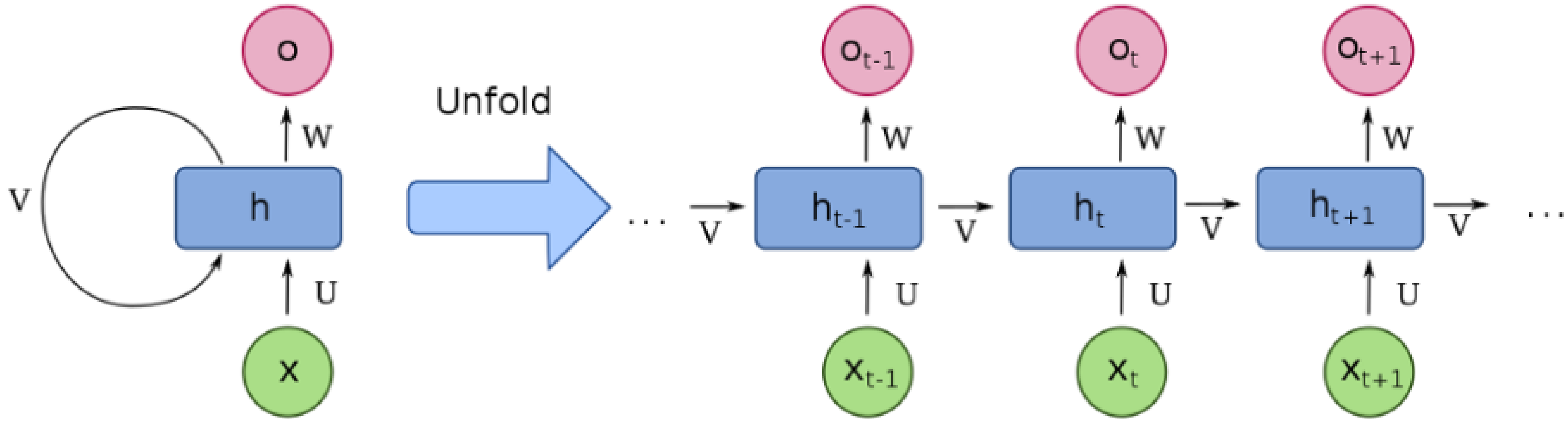
Evolution Accelerator



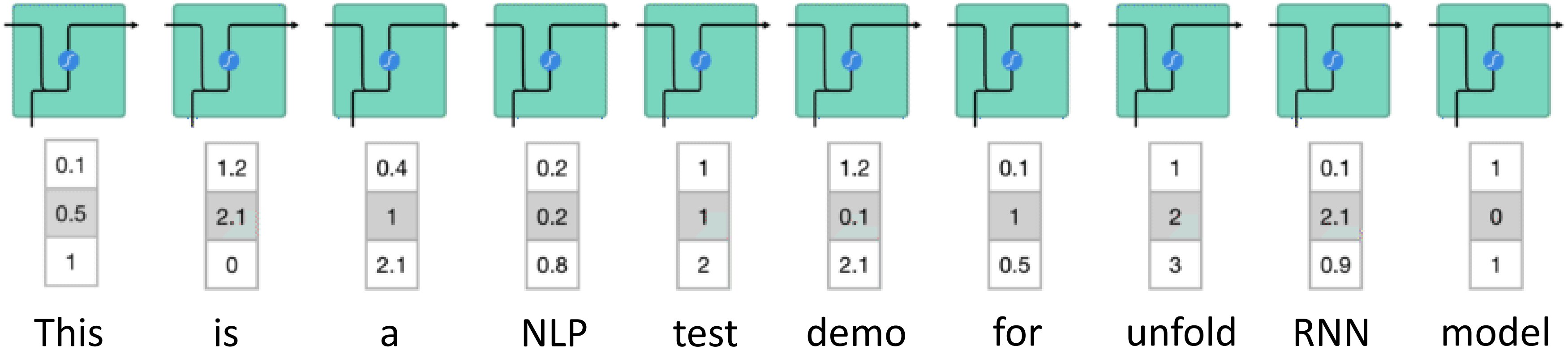
Word embedding models at early stage:

- Bag of words
- One-Hot
- TF-IDF
- **Word2Vec**
- GloVec (hypothesis: words that occur in same context tend to have similar meanings)

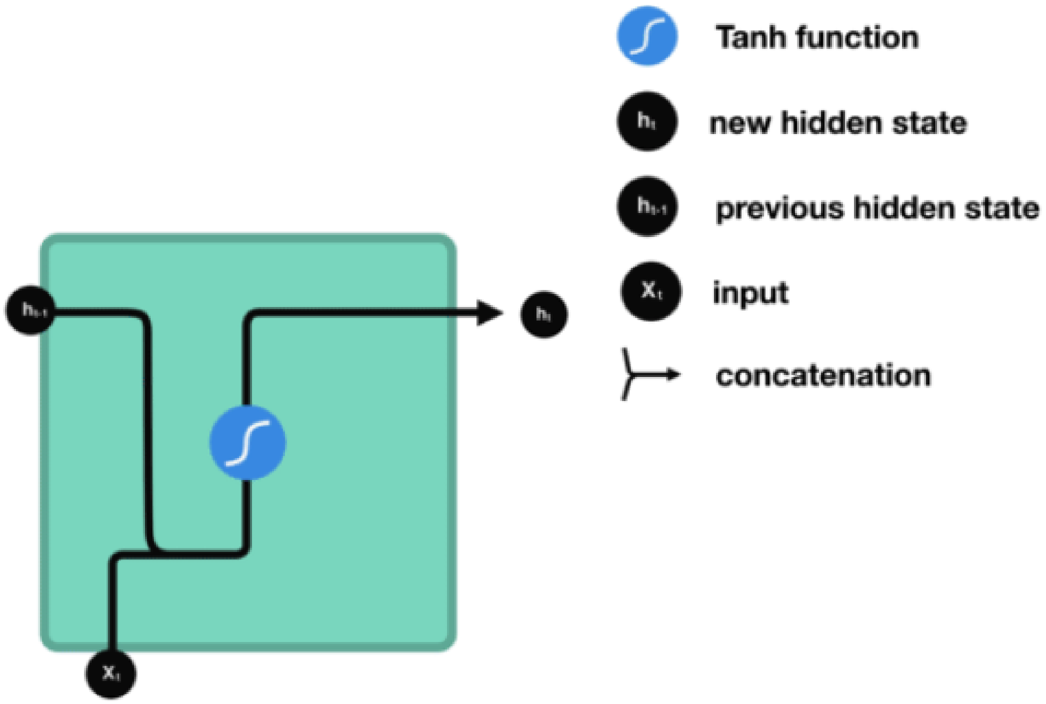
Gen #3 (RNN/LSTM/GRU)



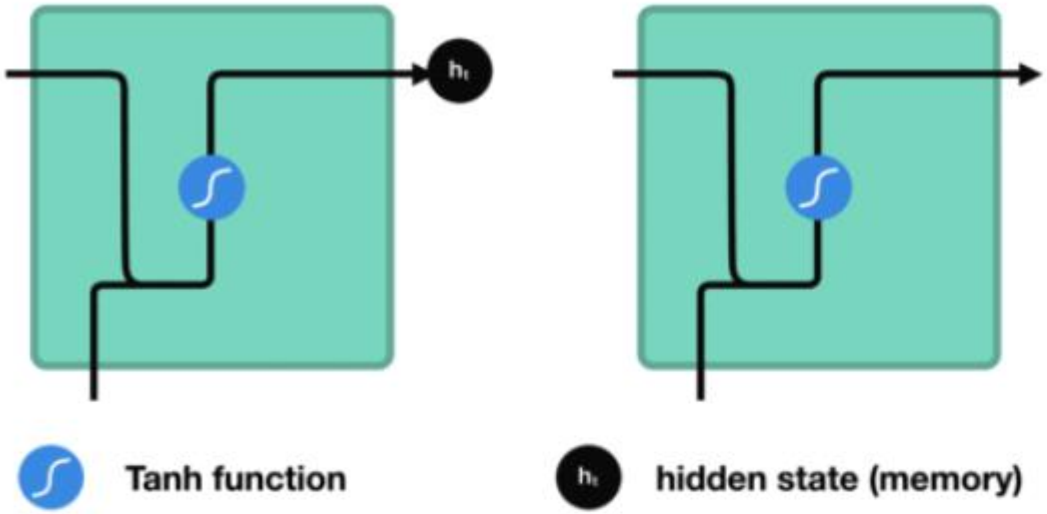
RNN – Recurrent Neural Network



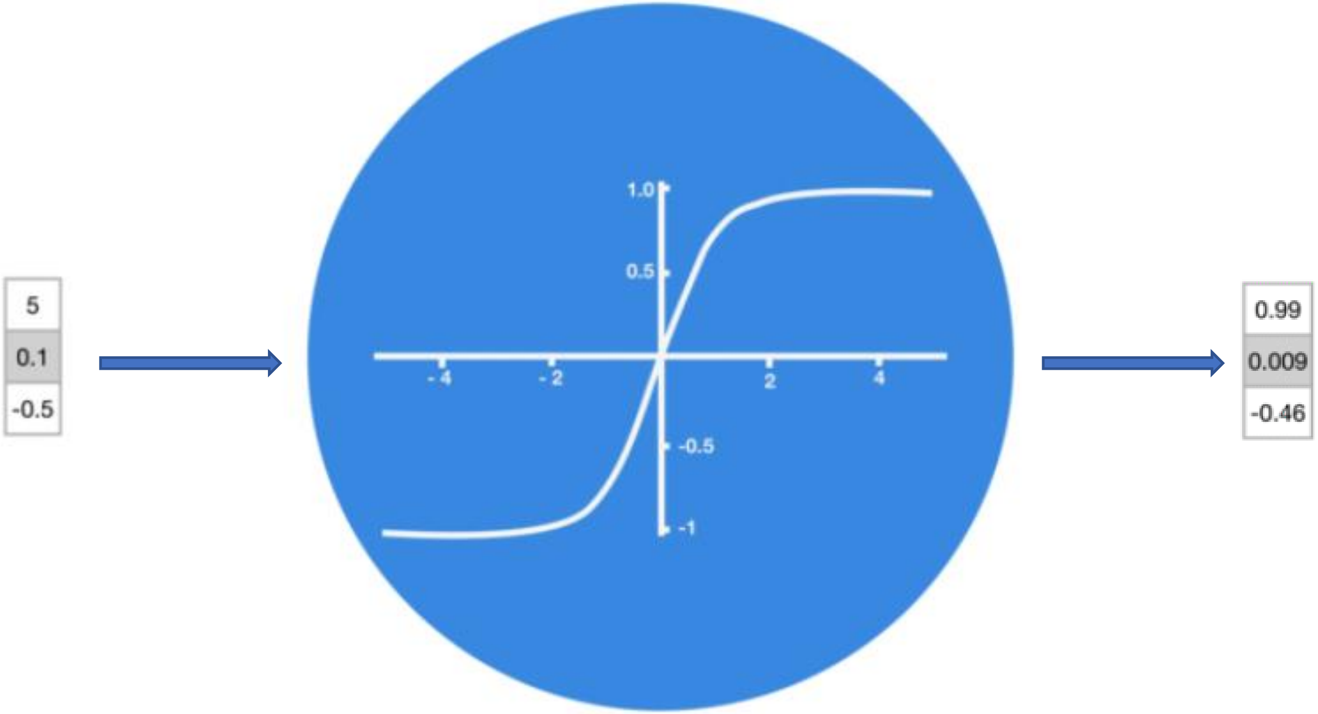
RNN - Unit


Tanh function
new hidden state
previous hidden state
input
concatenation

RNN – Hidden State

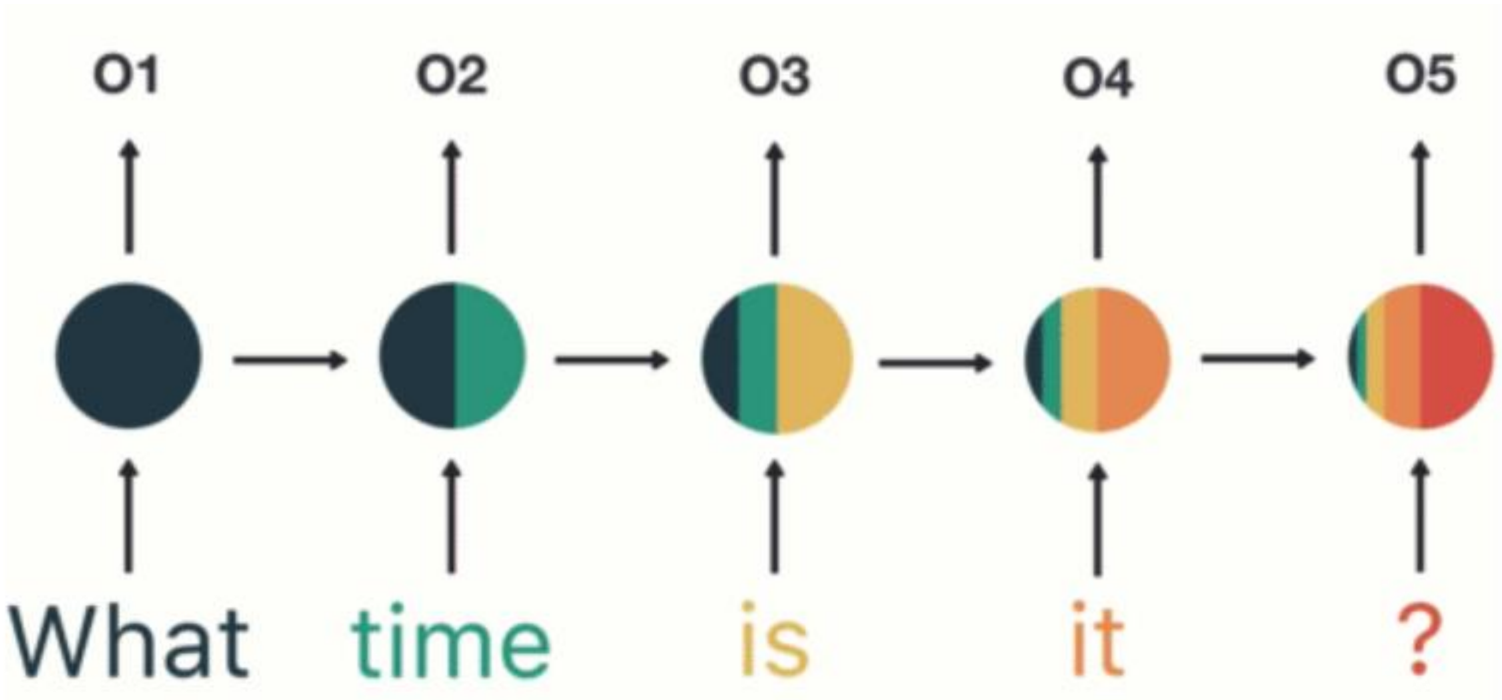


RNN - Tanh Activation

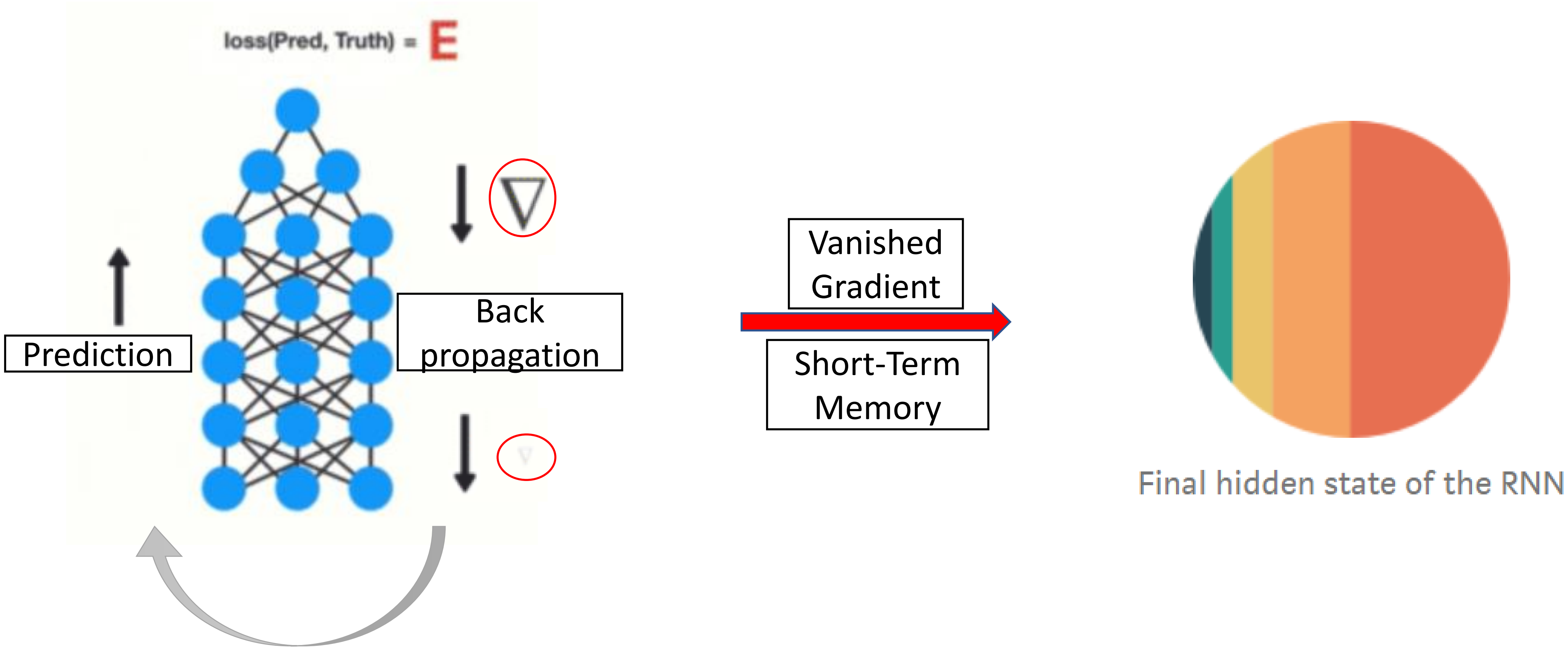


Activation function is used for regulating flow of information.

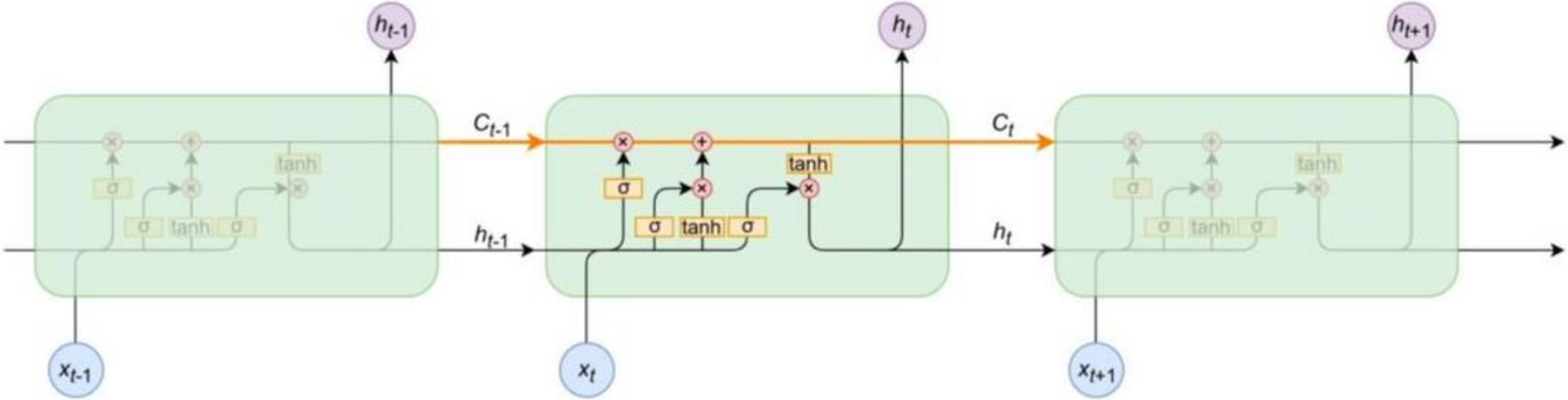
RNN – Vanishing Gradient



RNN – Vanishing Gradient Problem

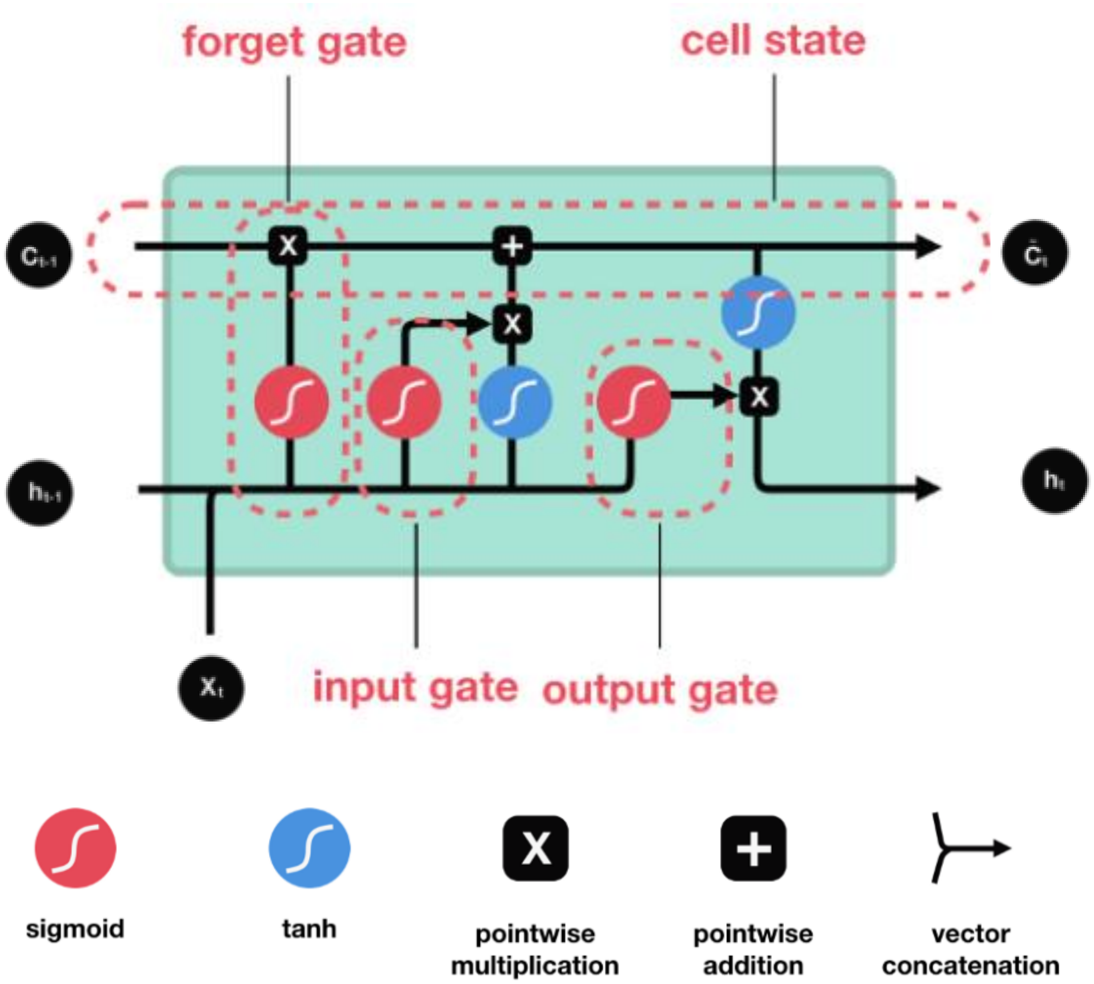


Long Short-Term Memory (LSTM)



C_t: Cell state is an internal state that is not output;
H_t: hidden state is an output

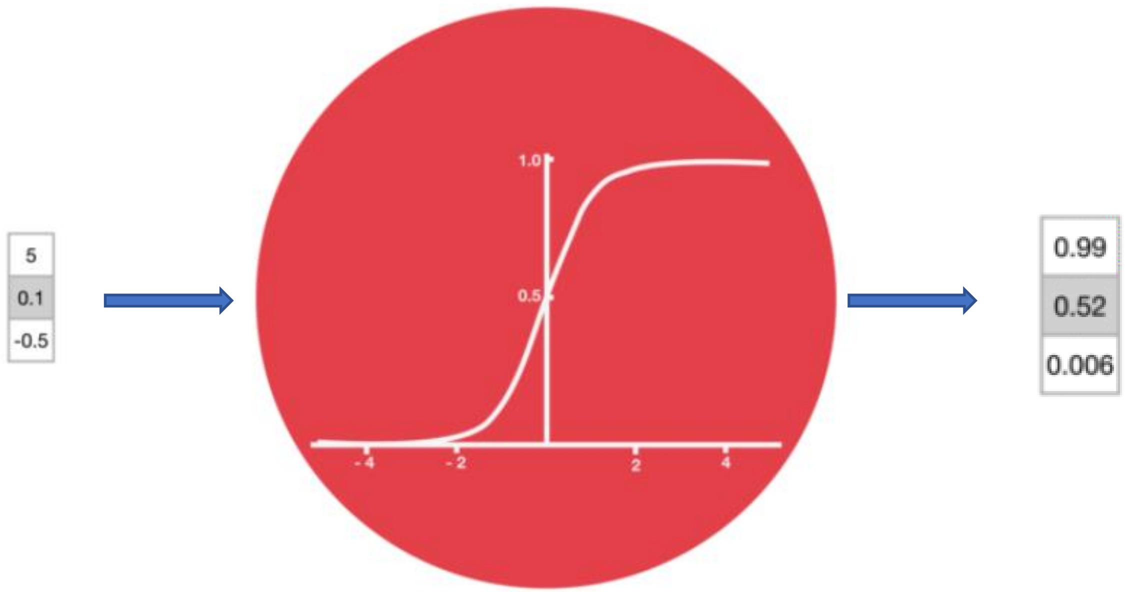
LSTM - Architecture



LSTM - Gates

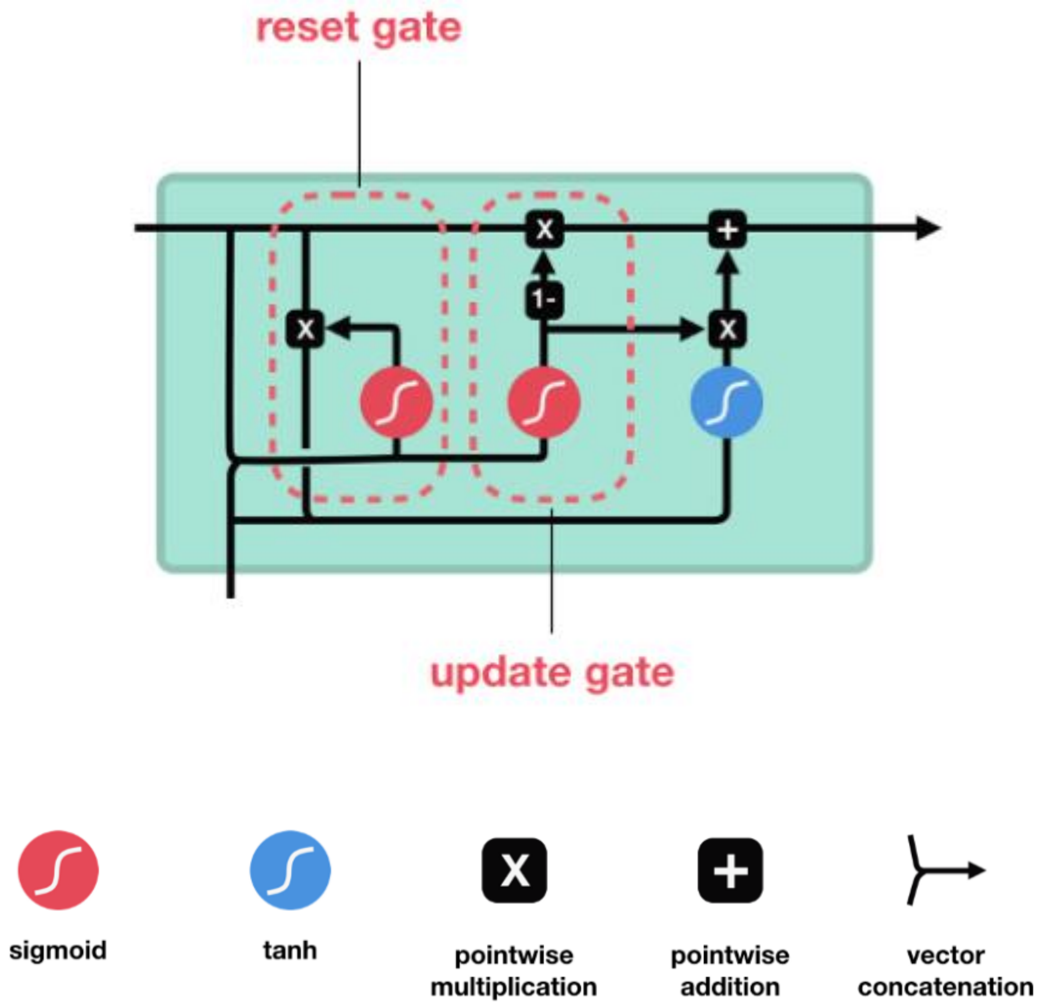
- Input Gate
- Cell State (“memory” of network)
  - act as a transport highway that transfers relative information all the way down the sequence chain.
- Forget Gate
  - learn what information is relevant to keep or forget during training.
- Output Gate

LSTM – Sigmoid Activation



Forget / Update Information

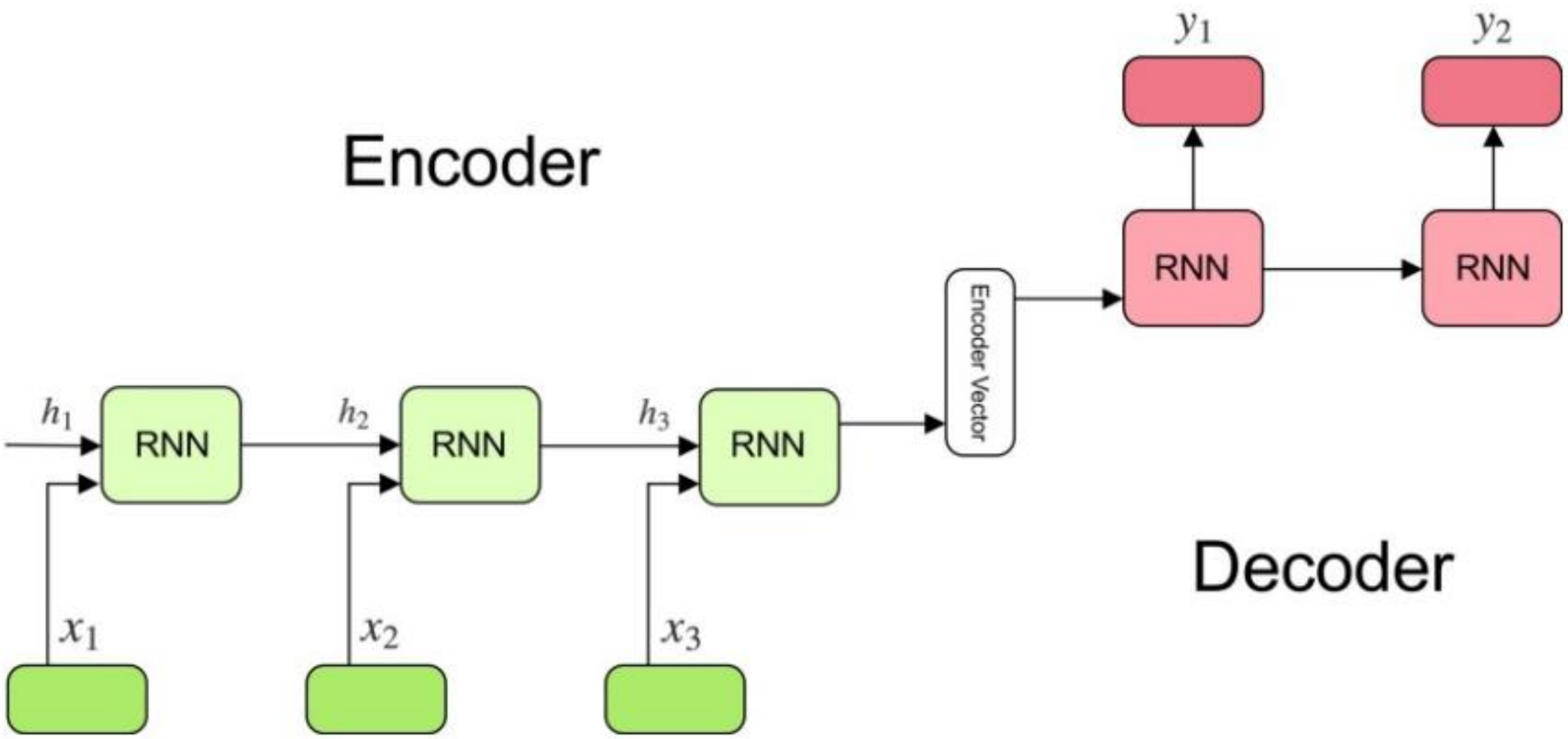
GRU – Gated Recurrent Unit



LSTM vs GRU

- GRU’s has fewer tensor operations, a little speedier to train than LSTM’s
- No clear winner which one is better.

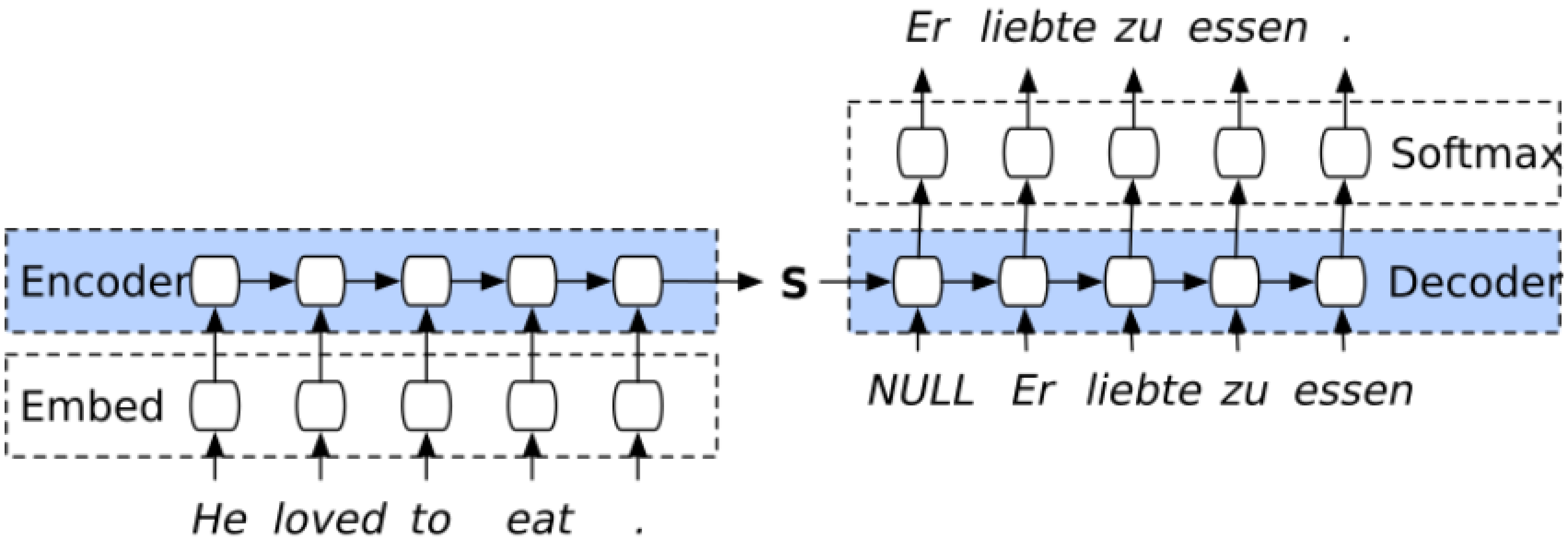
Gen #4 (Encoder-Decoder/Seq2Seq)



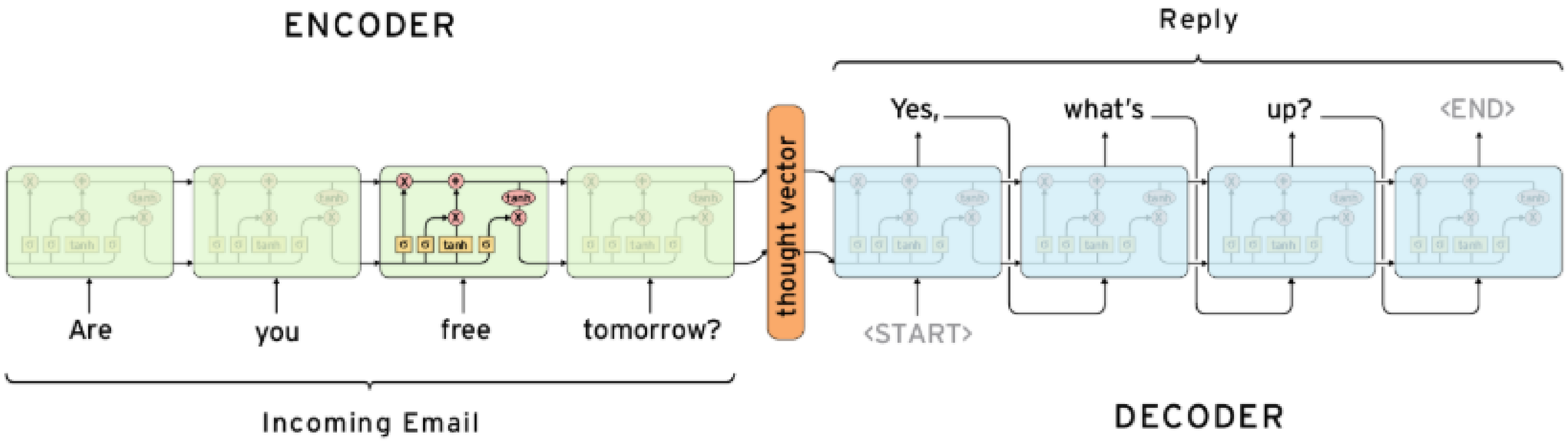
Encoder-decoder sequence to sequence model

- Unlike sequence prediction with a single RNN, where every input corresponds to an output, the seq2seq model frees us from sequence length and order, which makes it ideal for translation between two languages.

Encoder-Decoder for MT



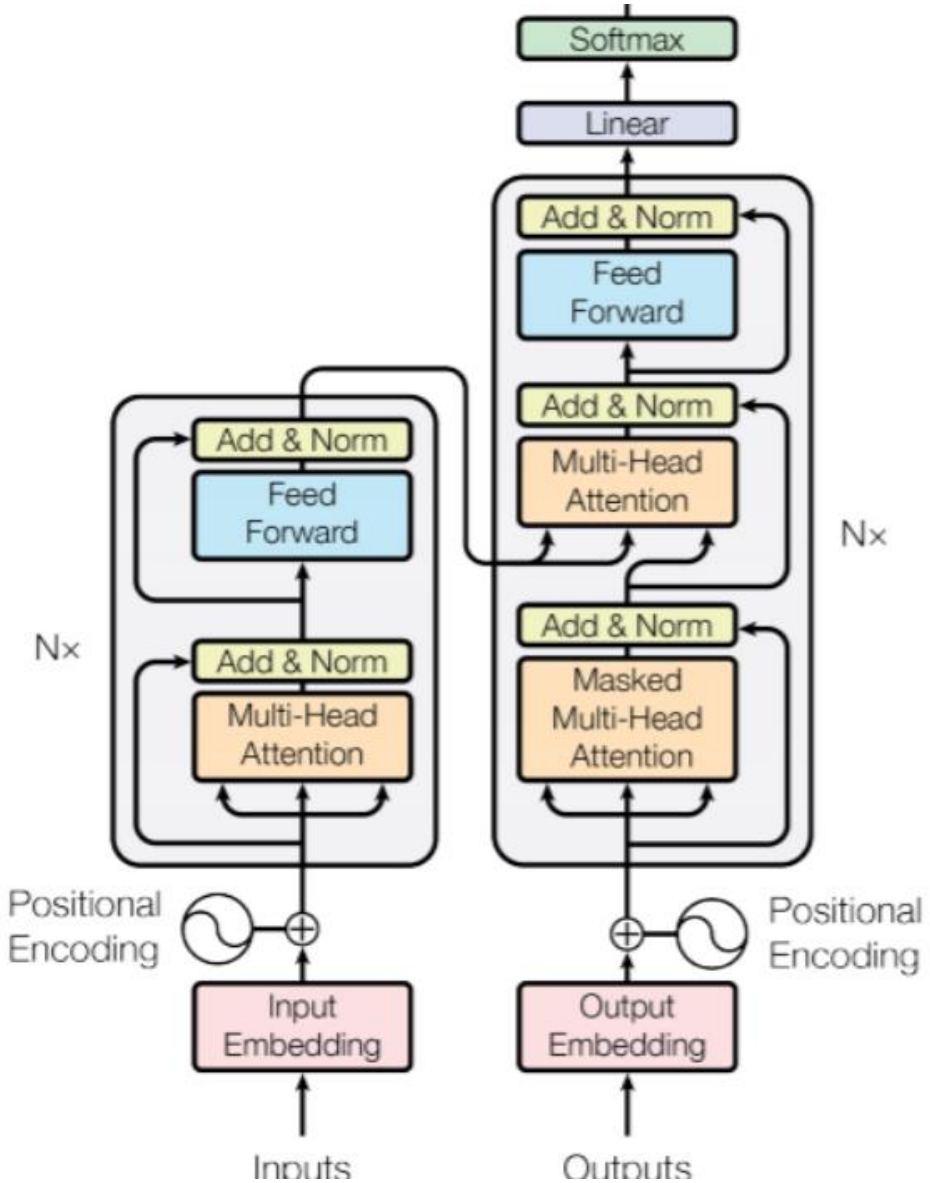
Encoder-Decoder for Chatbots



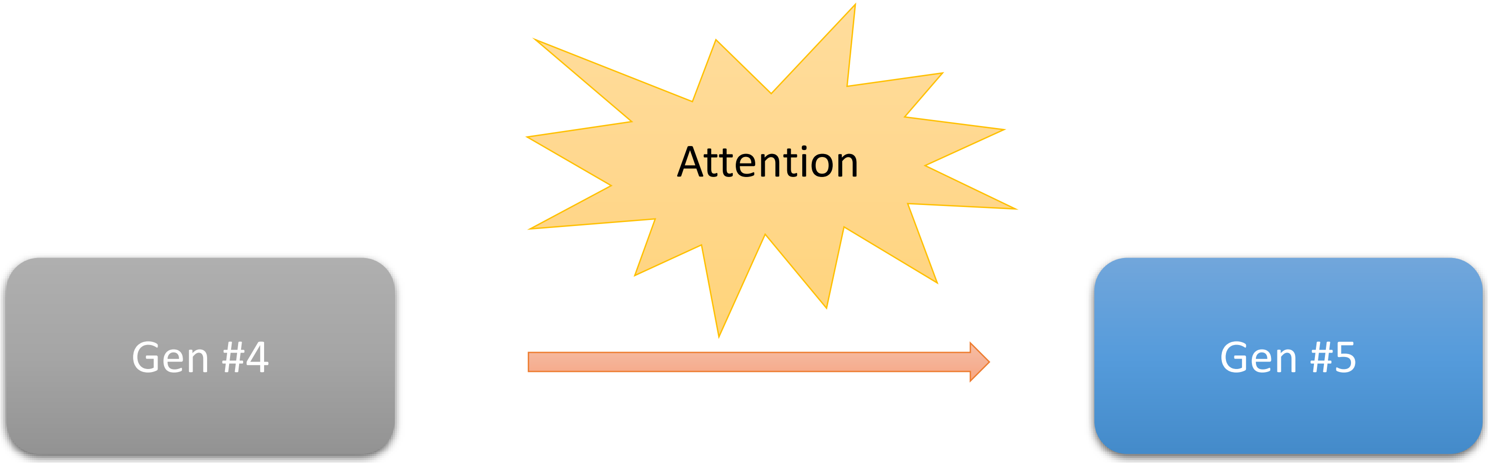
Generative Model Chatbots

Gen #5 (Transformer)

- Superior quality results on MT
- Parallelization benefits performance
- Learn long term dependency



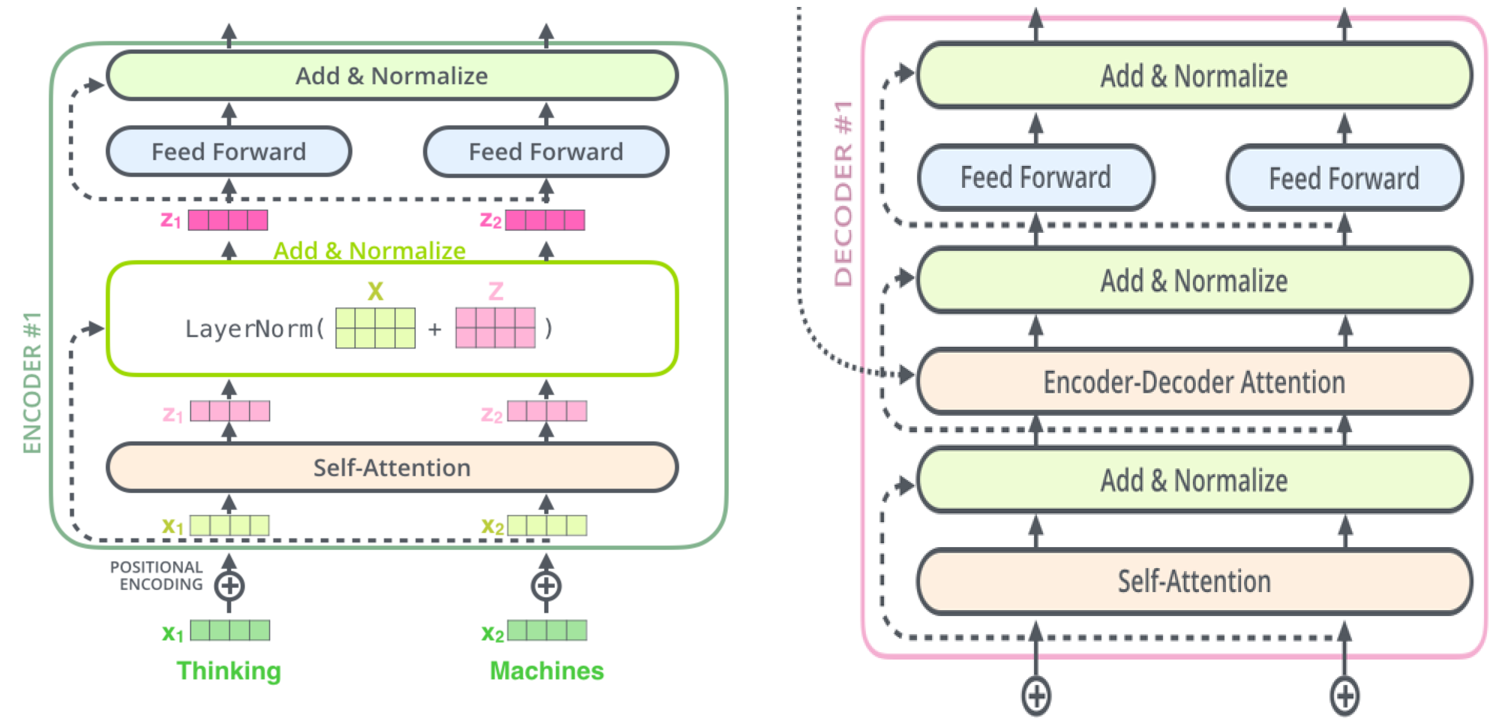
Evolution Accelerator



Attention:

- Born to solve the incapability of seq2seq models to remember longer sequences

Transformer (Encoder – Decoder)



Transformer – Pre-Trained Models

ELMo,
ULMfit
Jan 2018
Training:
103M words
1 GPU day

GPT
June 2018
Training
800M words
240 GPU days

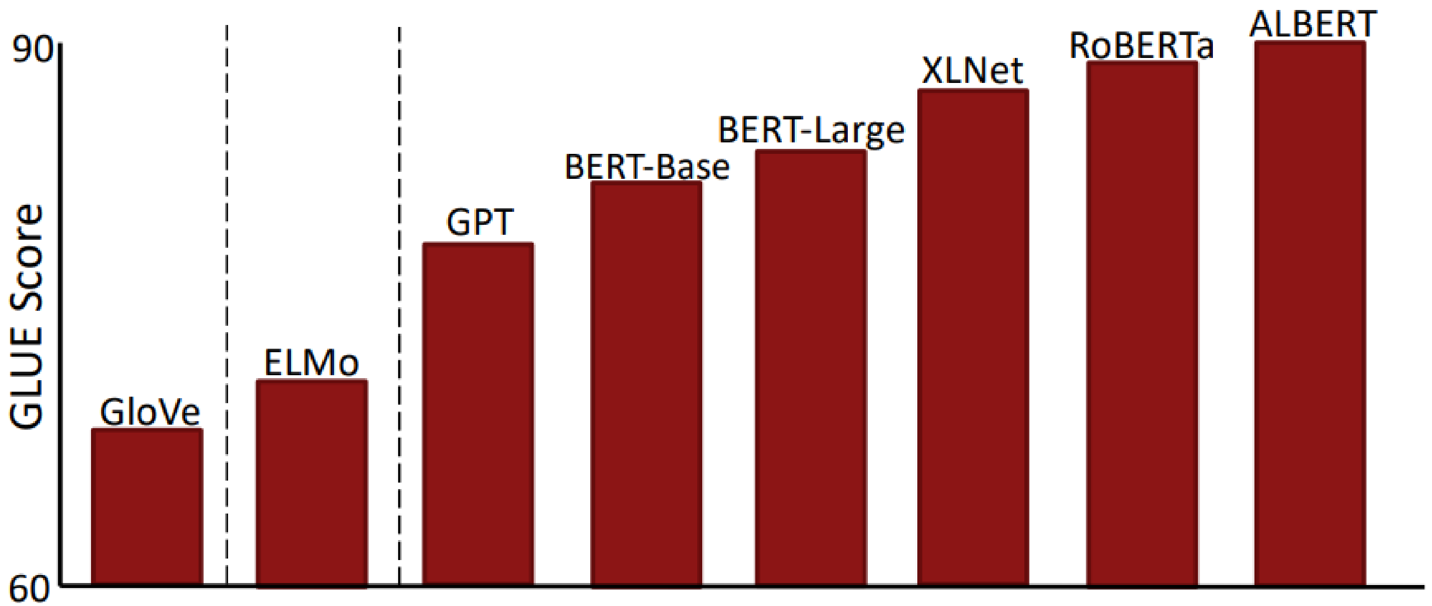
BERT
Oct 2018
Training
3.3B words
256 TPU days
~320–560
GPU days

GPT-2
Feb 2019
Training
40B words
~2048 TPU v3 days
according to a reddit thread

XL-Net, ERNIE,
Grover, AlBERT,
Megatron-LM, T5,
RoBERTa, GPT-3
July 2019–

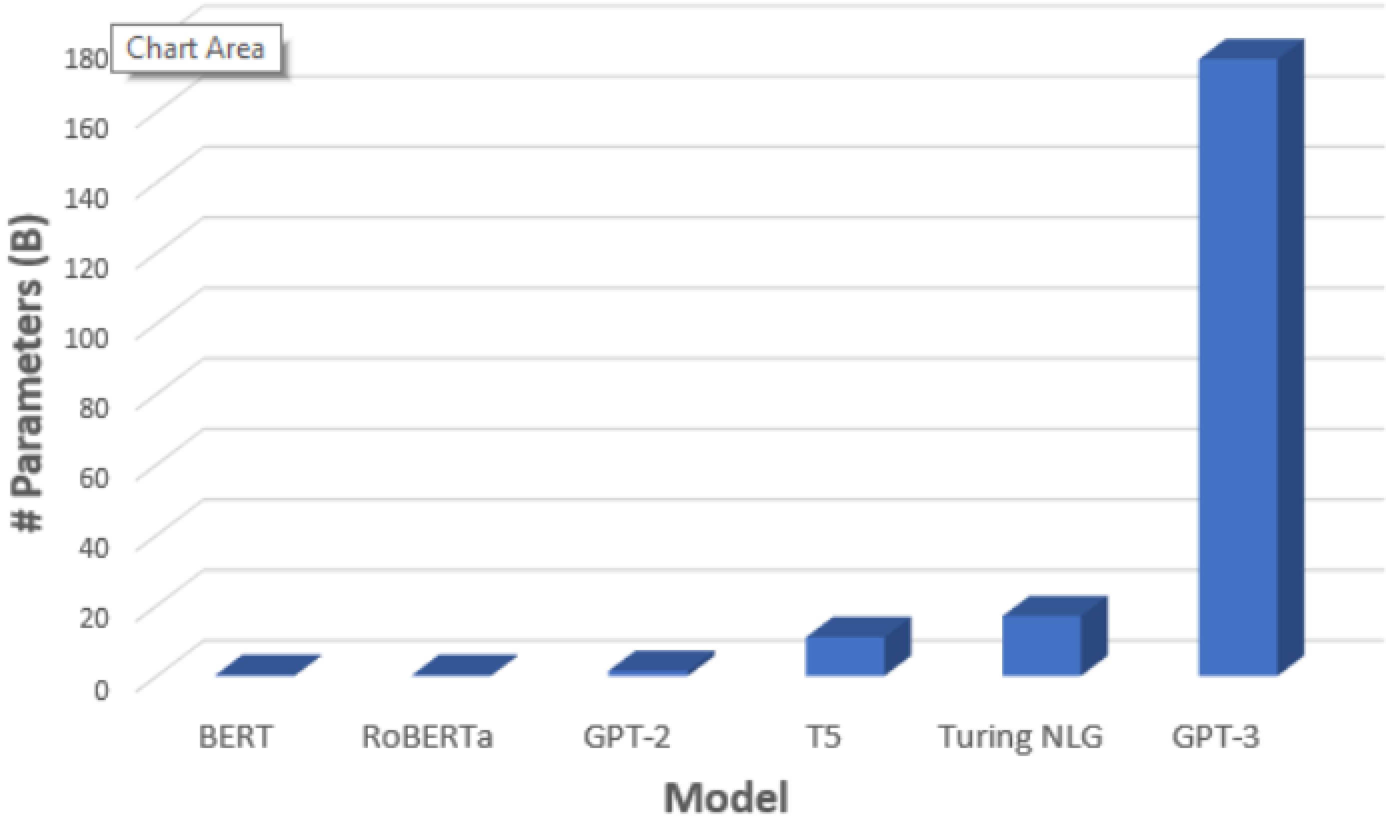
Transformer – Pre-Training Model Progress

Rapid Progress from Pre-Training **(GLUE benchmark)**



Over 3x reduction in error in 2 years, “superhuman” performance

GPT-3 (latest monster)



GPT-3 (latest monster)

- 175 billion parameters
- Applied to **ANY** language task

- Code generator
  - https://twitter.com/i/status/1282676454690451457

- Text to SQL query
  - https://twitter.com/FaraazNishtar/status/1285934622891667457?s=20