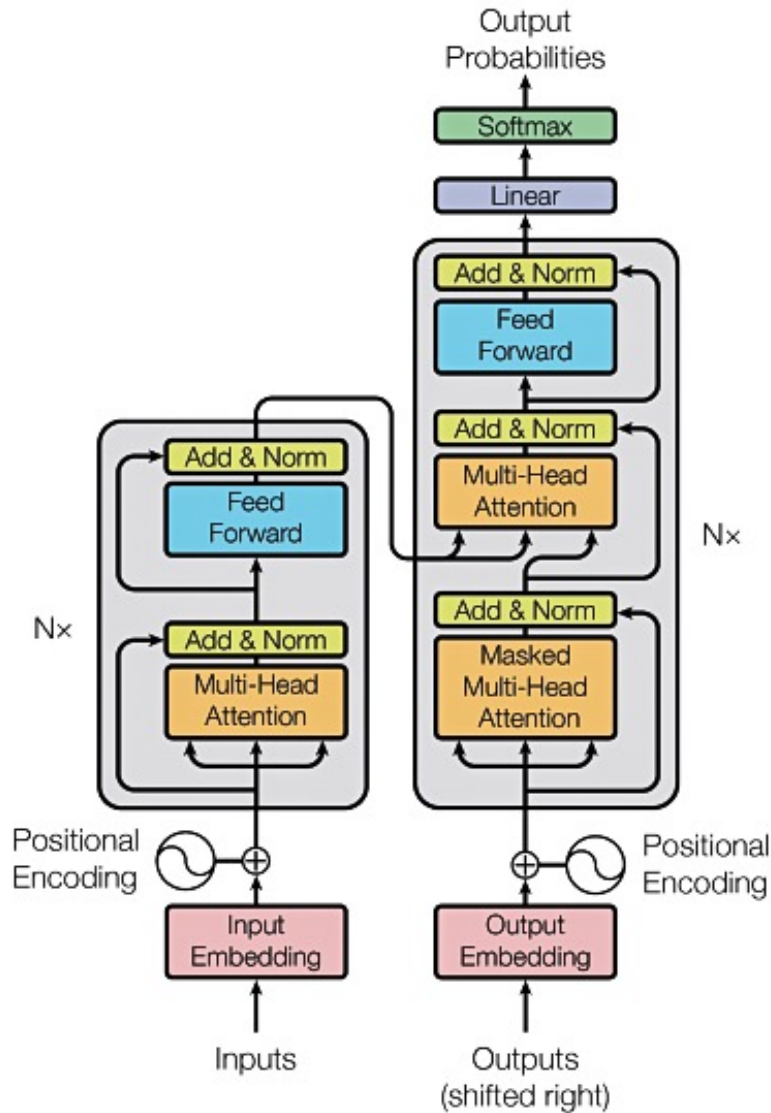


Transformers



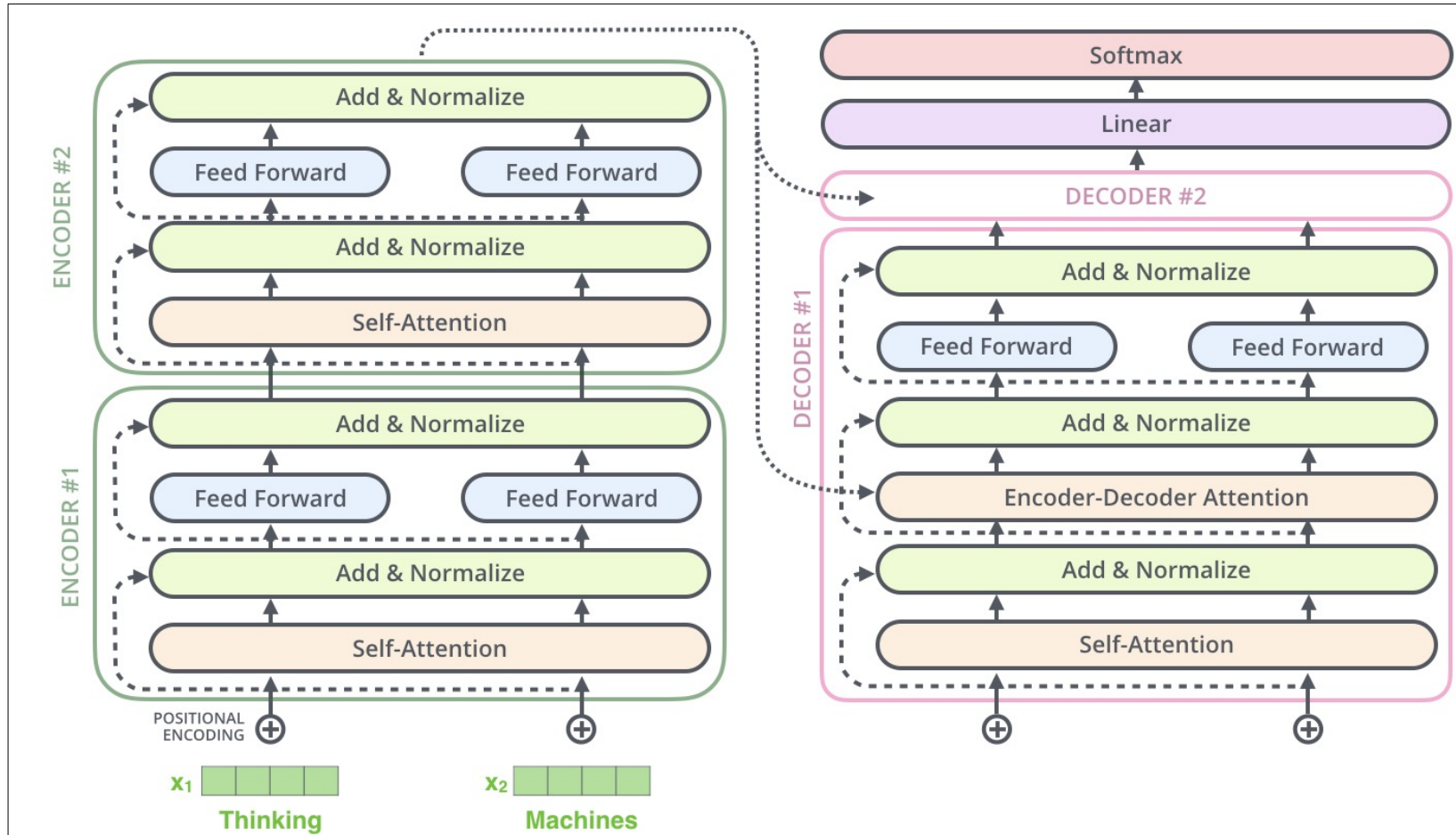
Background (1)

- The **RNN** and **LSTM** neural models were designed to process language and perform tasks like classification, summarization, translation, and sentiment detection
 - RNN: Recurrent Neural Network
 - LSTM: Long Short Term Memory
- In both models, layers get the next input word and have access to some previous words, allowing it to use the word's left context
- They used word embeddings where each word was encoded as a vector of 100-300 real numbers representing its meaning

Background (2)

- Transformers extend this to allow the network to process a word input knowing the words in both its left and right context
- This provides a more powerful context model
- Transformers add additional features, like attention, which identifies the important words in this context
- And break the problem into two parts:
 - An encoder (e.g., Bert)
 - A decoder (e.g., GPT)

Transformer model

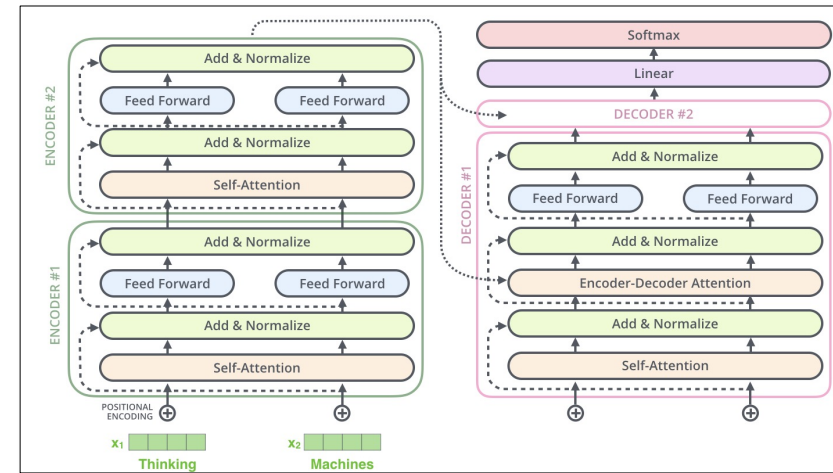


Encoder (e.g., BERT)

Decoder (e.g., GPT)

Transformers, GPT-2, and BERT

1. A transformer uses an **encoder stack** to model input, and uses **decoder stack** to model output (using input information from encoder side)
2. If we do not have input, we just want to model the “next word”, we can get rid of the encoder side of a transformer and output “next word” one by one. This gives us **GPT**
3. If we are only interested in training a language model for the input for some other tasks, then we do not need the decoder of the transformer, that gives us **BERT**



Training a Transformer

- Transformers typically use semi-supervised learning with
 - Unsupervised pretraining over a very large dataset of general text
 - Followed by supervised **fine-tuning** over a focused data set of inputs and outputs for a particular task
- Tasks for pretraining and fine-tuning commonly include:
 - language modeling
 - next-sentence prediction (aka completion)
 - question answering
 - reading comprehension
 - sentiment analysis
 - paraphrasing

Pretrained models

- Since training a model requires huge datasets of text and significant computation, researchers often use common pretrained models
- Examples (circa December 2021) include
 - Google's [BERT](#) model
 - Huggingface's various [Transformer models](#)
 - OpenAI's and [GPT-3 models](#)

Huggingface Models

The screenshot shows the Hugging Face website's 'Models' page. The browser address bar displays 'huggingface.co/models'. The page features a navigation bar with the Hugging Face logo, a search bar, and links for 'Models', 'Datasets', 'Spaces', 'Resources', 'Solutions', 'Pricing', 'Log In', and 'Sign Up'. On the left, there are sections for 'Tasks' and 'Libraries'. The main content area displays a list of models, each with its name, primary task, update date, download count, and heart icon.

Tasks

- Fill-Mask
- Question Answering
- Summarization
- Table Question Answering
- Text Classification
- Text Generation
- Text2Text Generation
- Token Classification
- Translation
- Zero-Shot Classification
- Sentence Similarity + 12

Libraries

- PyTorch
- TensorFlow
- JAX

Models 23,887 Sort: Most Downloads

- bert-base-uncased**
Fill-Mask • Updated May 18 • ↓ 24.9M • ♥ 72
- sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2**
Sentence Similarity • Updated Nov 2 • ↓ 12.2M • ♥ 10
- roberta-base**
Fill-Mask • Updated Jul 6 • ↓ 5.21M • ♥ 9
- distilbert-base-uncased**
Fill-Mask • Updated Aug 29 • ↓ 5.01M • ♥ 30
- gpt2**
Text Generation • Updated May 19 • ↓ 4.88M • ♥ 31

OpenAI Application Examples

The screenshot shows a web browser window with the URL `beta.openai.com/examples/`. The page features a navigation bar with links for Overview, Documentation, and Examples, along with Log in and Sign up buttons. The main content area displays a grid of application examples, each with a colored icon, a title, and a brief description.

Icon	Example Name	Description
Chat bubbles	Chat	Open ended conversation with an AI assist...
Question mark	Q&A	Answer questions based on existing knowle...
Red box with white text	Grammar correction	Corrects sentences into standard English.
Play button	Summarize for a 2nd grader	Translates difficult text into simpler concep...
Code icon	Natural language to OpenAI API	Create code to call to the OpenAI API usin...
Terminal icon	Text to command	Translate text into programmatic commands.
Globe	English to French	Translates English text into French.
Dollar sign	Natural language to Stripe API	Create code to call the Stripe API using nat...
Question mark	SQL translate	Translate natural language to SQL queries.
Table icon	Parse unstructured data	Create tables from long form text
Tag icon	Classification	Classify items into categories via example.
Hash icon	Python to natural language	Explain a piece of Python code in human un...
Smiley face	Movie to Emoji	Convert movie titles into emoji.
Clock	Calculate Time Complexity	Find the time complexity of a function.
Text icon	Translate programming languages	
Hash icon	Advanced tweet classifier	

GPT-2, BERT

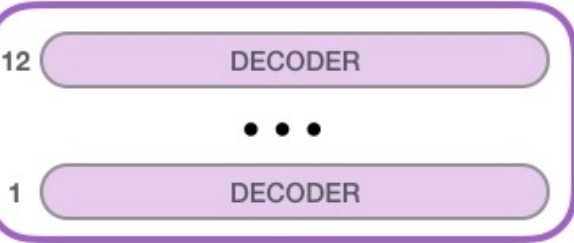


GPT released June 2018

GPT-2 released Nov. 2019 with 1.5B parameters

GPT-3 released in 2020 with 175B parameters

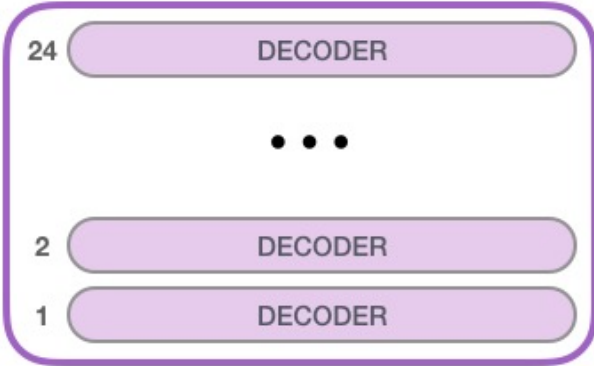
 GPT-2
SMALL



Model Dimensionality: 768

117M parameters

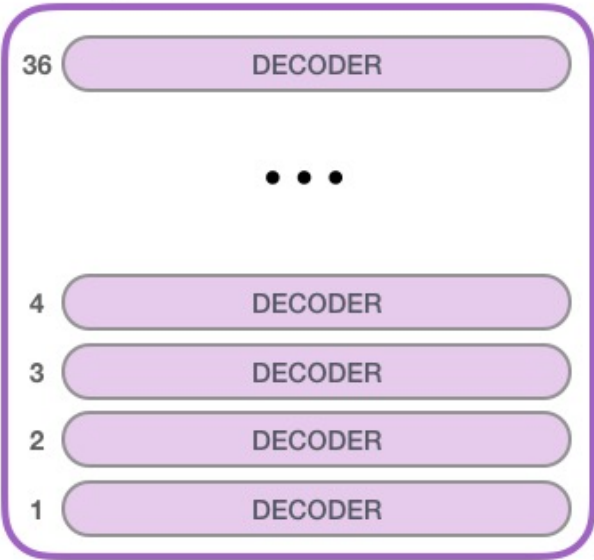
 GPT-2
MEDIUM



Model Dimensionality: 1024

345M

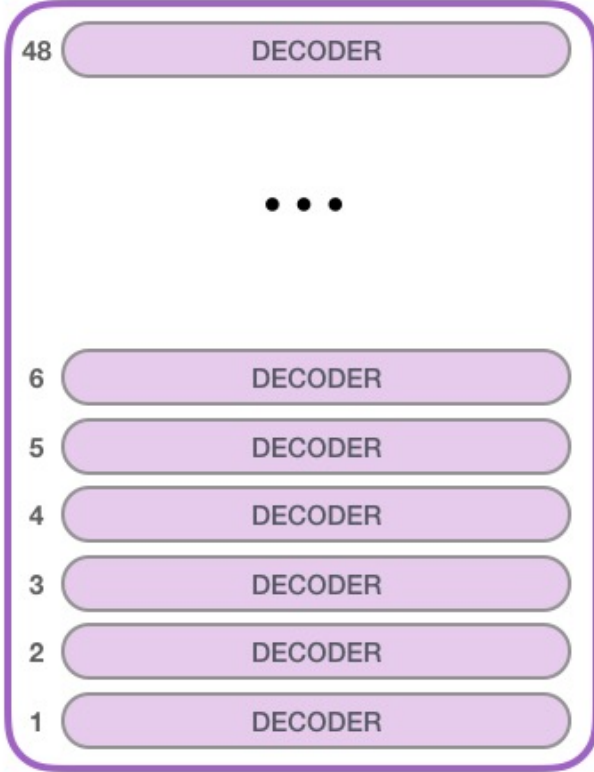
 GPT-2
LARGE



Model Dimensionality: 1280

762M

 GPT-2
EXTRA
LARGE



Model Dimensionality: 1600

1542M