QTM 347 Machine Learning

Lecture 21: Introduction to Foundation Models

Ruoxuan Xiong

EMORY

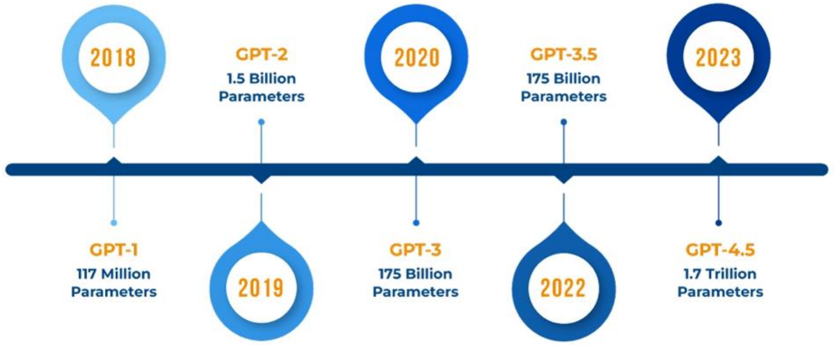
Lecture plan

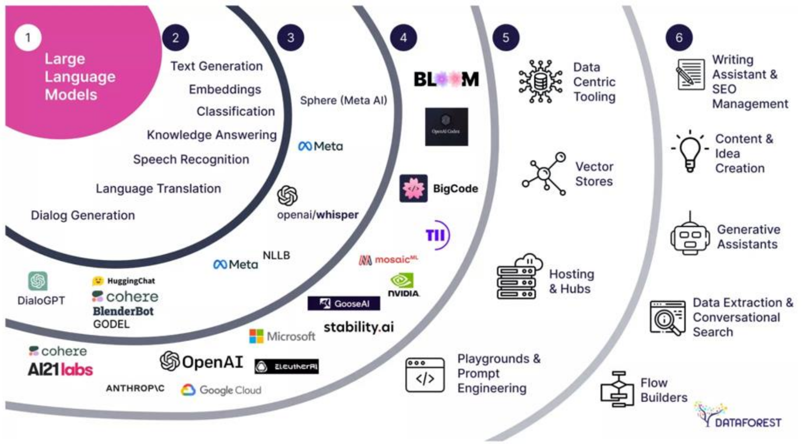
- Introduction to foundation models
- Introduction to language models

Foundation model

- A **foundation model** is a large-scale, pre-trained machine learning model that serves as a general-purpose base for a wide-range of downstream tasks

- Foundation models are used in many disciplines
  - Natural language process (NLP). For example, GPT (generative pre-trained transformer)
    - Developed by OpenAI
    - GPT-3: 175 billion parameters; GPT-4 with more parameters
    - Training data: books, articles, websites, and other sources
    - Applications: text generation, translation, and conversation





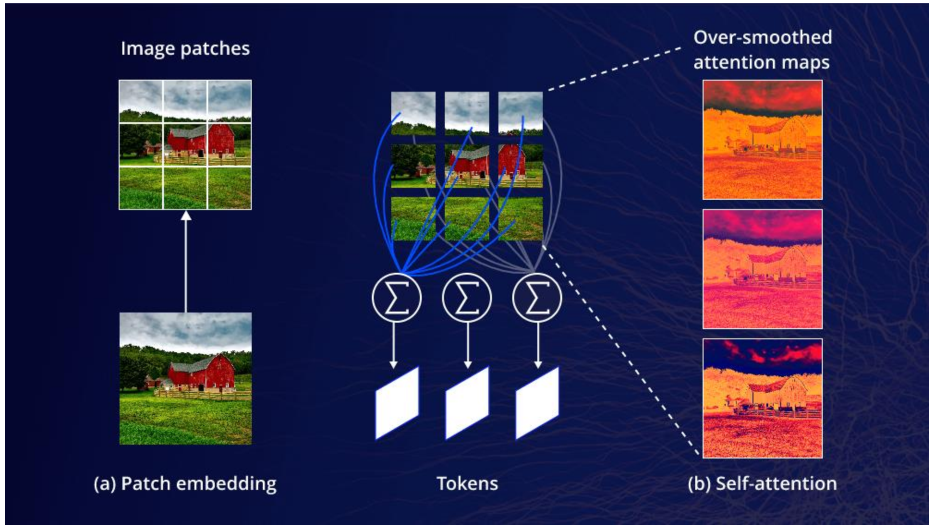
1 — Available Large Language Models
2 — General Use-Cases
3 — Specific Implementations
4 — Models
5 — Foundation Tooling
6 — End User UIs

EMORY

Foundation model

- A **foundation model** is a large-scale, pre-trained machine learning model that serves as a general-purpose base for a wide-range of downstream tasks

- Foundation models are used in many disciplines
  - Computer vision (CV). For example, Vision Transformer (ViT)
    - Introduced by Google [paper]
    - Training data: ImageNet-21k and Google JFT-300M
    - Applications: image classification, object detection



EMORY

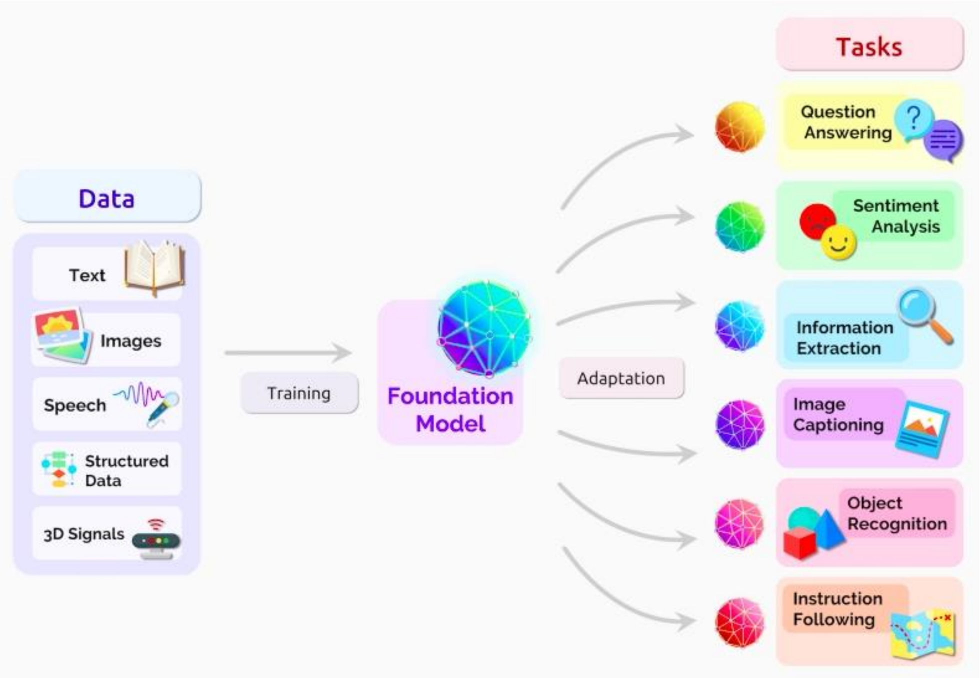
Foundation model

- A **foundation model** is a large-scale, pre-trained machine learning model that serves as a general-purpose base for a wide-range of downstream tasks

- Foundation models are used in many disciplines
  - Multimodal models. For example, DALL-E (use transformer, diffusion model)
    - Developed by OpenAI: generates high-quality images from textual description
    - DALL-E (original): 12 billion parameters
    - Applications: text-to-image generation for creative design, marketing, prototyping

Opportunities of foundation models

- Enable a new paradigm of AI-driven innovation. For example,
  - Accelerating AI development, e.g., general-purpose models, efficiency
  - Enhancing creativity and innovations, e.g., idea exploration
  - Revolutionizing healthcare, e.g., drug discovery, diagnosis, chatbot
  - Driving business and operational efficiency, e.g., process automation, decision support, scalability



Bommasani, Rishi, et al. "On the opportunities and risks of foundation models." *arXiv preprint arXiv:2108.07258* (2021).

EMORY

Risks of foundation models

- We still lack a clear understanding of how foundation models work and when they fail
  - Bias and fairness issues, e.g., gender bias, cultural or racial bias
  - Data privacy risks, e.g., memorization of training data
  - Over-reliance and automation risks, e.g., erosion of human expertise and skills
  - Environmental impact, e.g., high energy consumption
  - Hallucination, e.g., factually incorrect, nonsensical
  - Stochastic parrot, e.g., generate texts without true understanding or reasoning
  - Lack of explainability

Bommasani, Rishi, et al. "On the opportunities and risks of foundation models." *arXiv preprint arXiv:2108.07258* (2021).

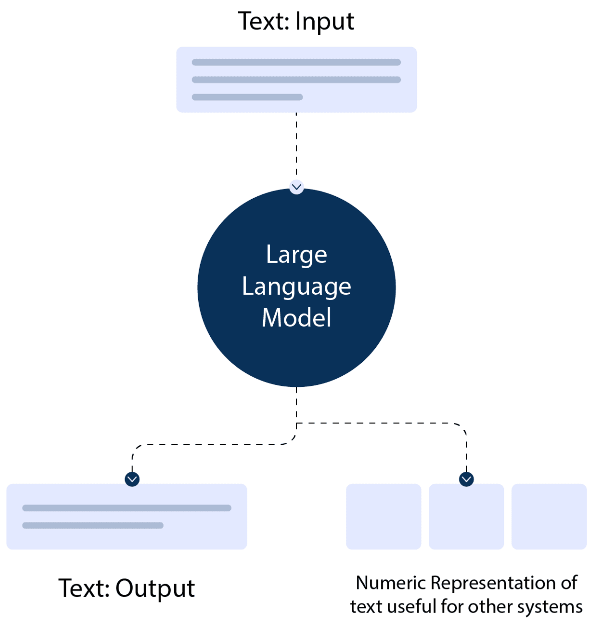
Lecture plan

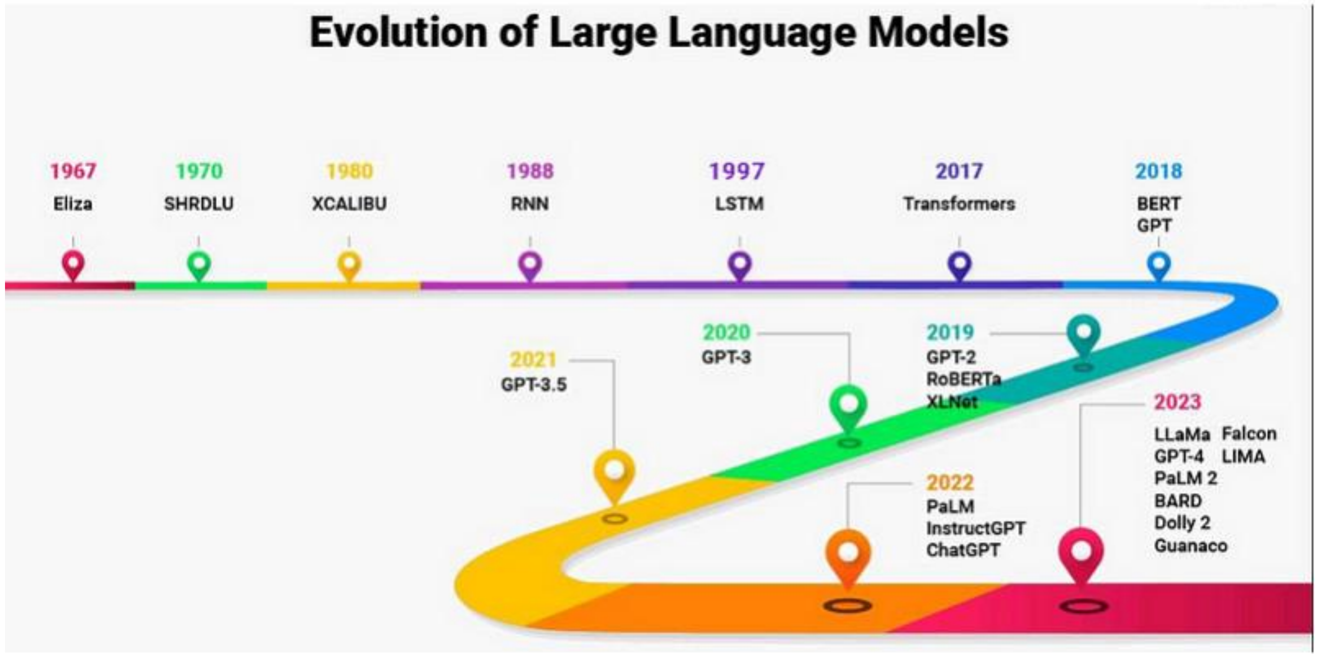
- Introduction to foundation models

- Introduction to language models

Language models

- A language model enables computers to understand, generate, and interact with human language effectively
- It evolves from simple probabilistic approaches to advanced transformer-based models like GPT and BERT





EMORY

History of language models

- **Stage 1**: Early foundations, rule-based systems (1950s-1970s)

- Relied on manually crafted rules for syntax and grammar, lacking statistical or probabilistic components

- Examples:
  - Eliza (1966): A rule-based chatbot that simulates a psychotherapist using simple pattern matching
  - SHRDLU (1970s): A system that used logic-based rules to understand and manipulate blocks in a simulated environment

History of language models

- **Stage 2**: Statistical language models (1980s-1990s)

- Probabilistic models to estimate the probability of word sequences based on observed frequencies in text corpora

- **Key techniques**: n-grams, hidden Markov models (HMM)

N-gram models

- Predict the next word/token based on the previous $n - 1$ words/tokens

- For example, a trigram ($n = 3$) model will define

$$\text{P(cheese} \mid \text{the, mouse, ate, the)} = \text{P(cheese} \mid \text{ate, the)}$$

- **Limitations**: Difficulty capturing long-range dependencies due to fixed n-gram windows

History of language models

- **Stage 3**: Neural language models (NLMs) (2000s)

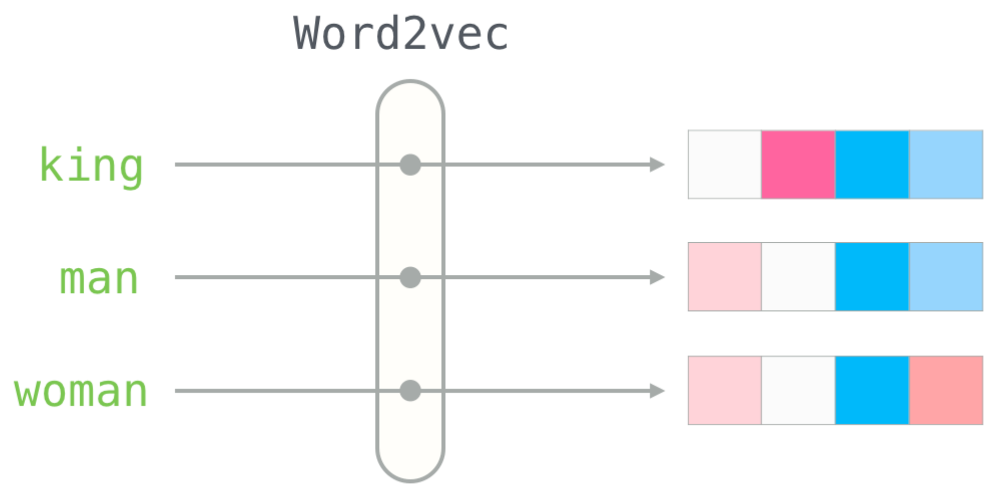
- Use neural networks to learn distributed representations of words, known as word embeddings

- NLMs could generalize better by capturing semantic and syntactic relationships

- **Key innovations**: Word2Vec (2013), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM, 1997)

Word2Vec

- A neural network-based model developed by Google in 2013 to generate word embeddings
- Word2Vec maps each word to a fixed-size vector in a high-dimensional space. Words with similar meanings or contexts are placed close together in this space
- Example: "woman" and "man" will be closer than "king" and "woman"



Mikolov, Tomas. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* 3781 (2013).

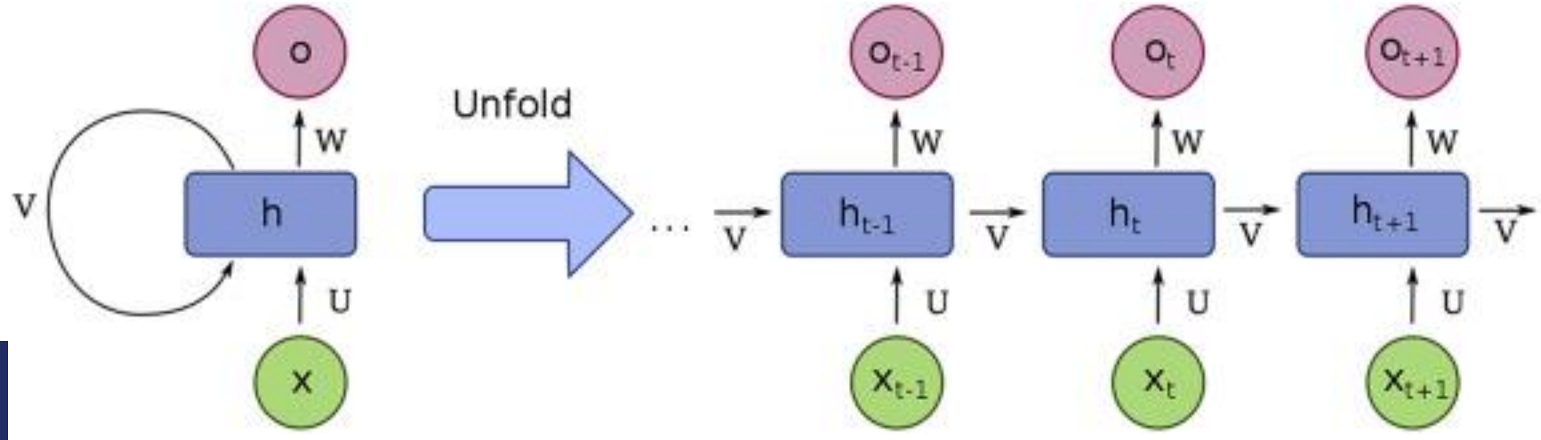
Recurrent Neural Network (RNN)

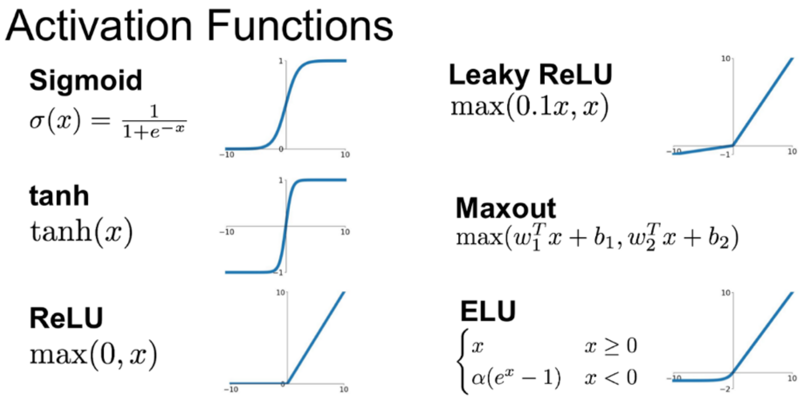
- A type of artificial neural network designed to handle sequential data by maintaining a memory of past inputs
- Inputs: Input sequences (e.g., words in a sentence, time-series data)
- Hidden state: At time t, the hidden state is updated based on

$h_t = f(W_h h_{t-1} + W_x x_t + b)$ *for weights* W_h, W_x *and* b *and activation function* f

- Output: The output depends on the hidden state

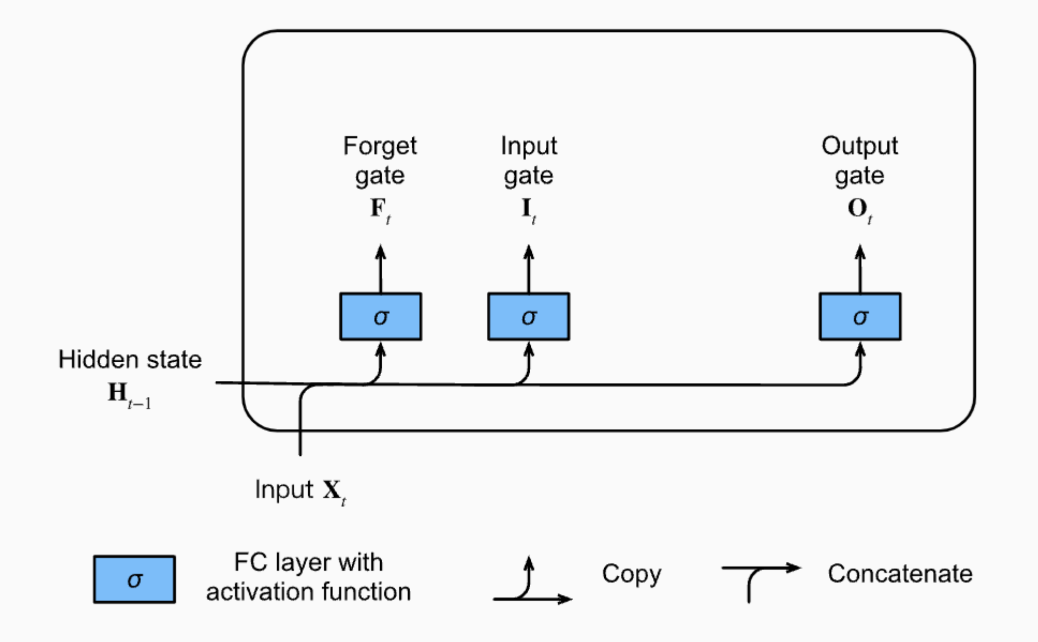
$y_t = g\left(W_y h_t + b_y\right)$ *for activation function* g





Long Short-Term Memory (LSTM)

- LSTM is a type of RNN designed to maintain information over long sequences
- LSTM has three gates to control the flow of information
  - Forget gate: Determines what information to discard
  - Input gate: Determines what new information to store in memory
  - Output gate: Determines what information to output
- LSTM also has hidden states



History of language models

- **Stage 4**: Transformer revolution (2017-present)
- Transformer architecture in the paper "Attention is All You Need" revolutionized language modeling
- Transformers use **self-attention mechanisms** to model long-range dependencies efficiently
- **Key Milestones**: BERT (2018), GPT (2018-now)

Transformer

- **Key feature**: **Self-Attention Mechanism**

- It allows the model to focus on relevant parts of the input sequence, regardless of their distance from the current token

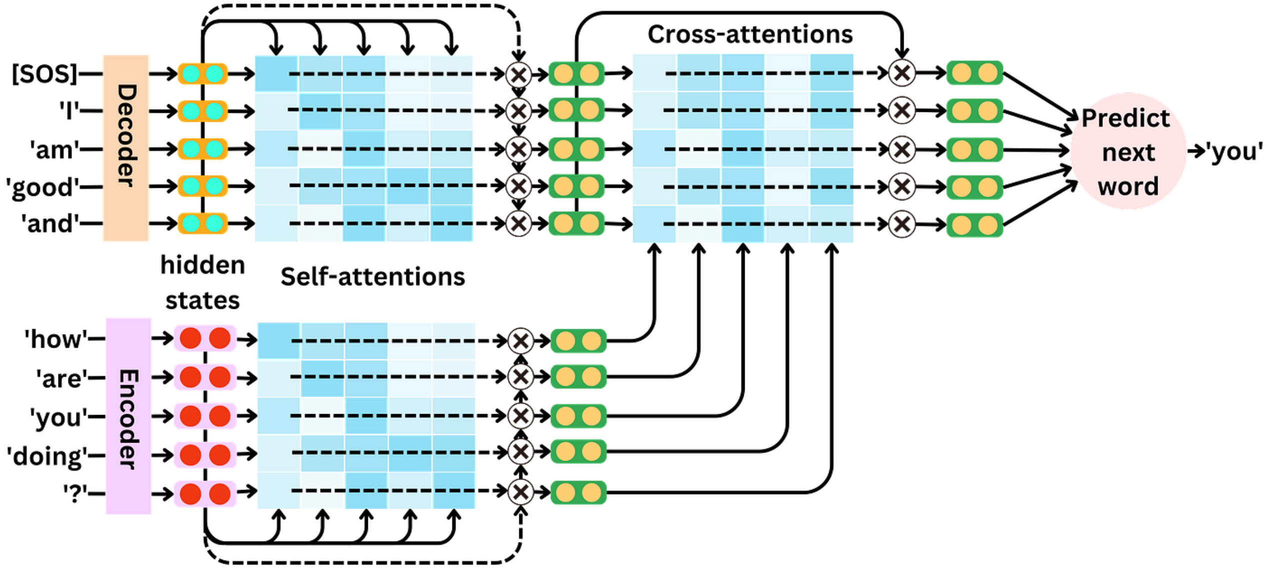
- Example #1: "*The cat chased the mouse, and it ran away*."
  - "It" can refer to "mouse" or "cat"
  - Self-attention helps the model focus on "mouse" based on semantic relationship

- Example #2: "*The dog chased the ball, and it was happy*."
  - "It" can refer to "dog" or "ball"
  - Self-attention helps the model focus on "dog" based on semantic relationship

EMORY

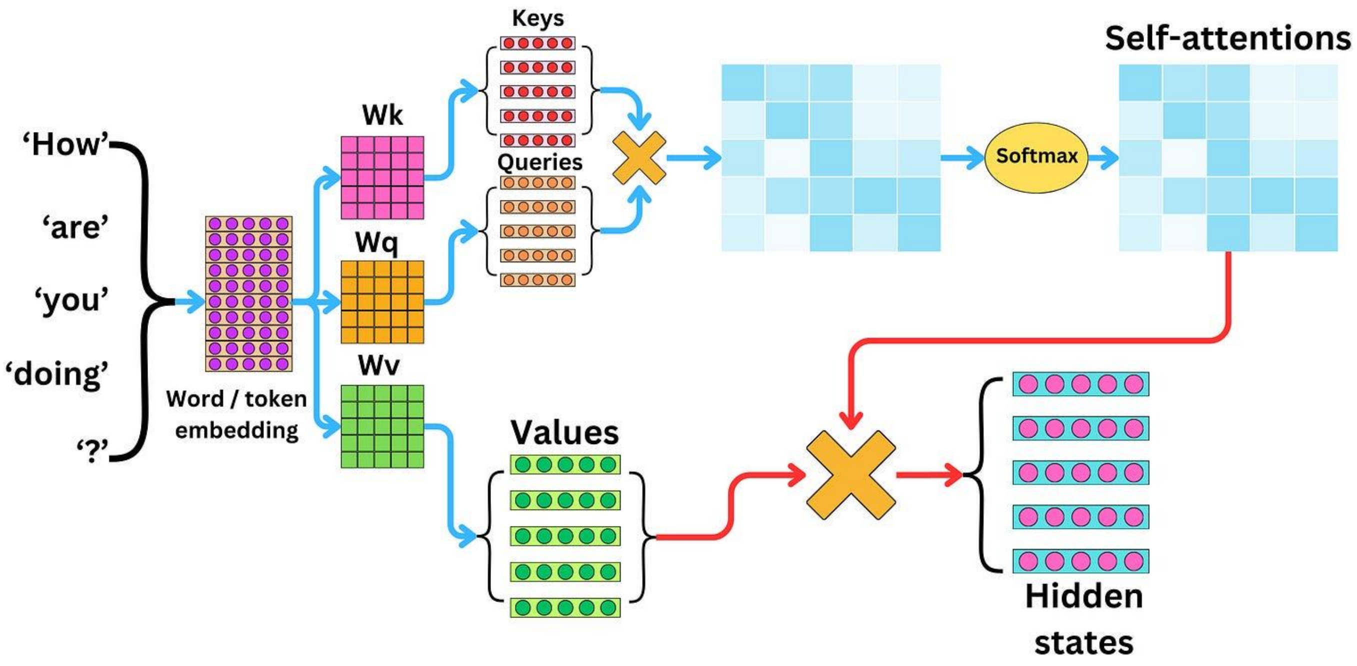
Self-attention

- Input sequence: *How are you doing*?
- Output sequence: *I am good and* ______
- Self-attention can capture the interactions between words within the input sequence and within the output sequence



Self-attention mechanism

- First compute the **keys**, **queries**, and **values** vectors
- Next compute the **matrix multiplication** between the keys and queries
- After a **softmax transformation**, this matrix of interactions is called the **attention matrix**
- Multiply attention matrix by values vector to obtain **hidden states**



https://newsletter.theaiedge.io/p/understanding-the-self-attention

Bidirectional encoder representations from transformers

- **Bidirectional**: understand the context of words in a sentence by considering both the words before and after the target word
- BERT-base: 12 transformer layers (encoders), 768 hidden dimensions, 12 attention heads; Total parameters: ~110 million
- **Two pre-training objectives**:
  - **Masked language model** (MLN): randomly mask some words in the input sentence and trains the model to predict these words based on context
    - Example: Input: "*The cat eats [MASK]*"; Output: "*fish*"
  - **Next sentence prediction** (NSP): trains the model to determine if a given sentence follows another logically
    - Example: Sentence A: "*I love traveling.*" Sentence B: "*During the Thanksgiving break, I went to Miami.*" (related) Sentence B: "*We have final project presentations this week.*" (unrelated)

Kenton, Jacob Devlin Ming-Wei Chang, and Lee Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." *Proceedings of naacL-HLT*. Vol. 1. 2019.

Generative Pre-trained Transformer (GPT)

- Generate human-like text and perform various natural language processing (NLP) tasks

- **Autoregressive model** (unidirectional): predict the next word in a sequence based on previous words
  - Example: Input: "*The sky is*"; Output: "*blue*"

- Training objective: maximize the probability of next word given preceding words

- Tokenization: Text is broken into small units called tokens (e.g., words, subwords, characters)

EMORY