INTRODUCTION TO LLM

Speakers:

Anurag Mahajan

Saloni Redij

Hosts: Asmita Samuel,

Isha Shetye

University at Buffalo The State University of New York

Agenda

- Introduction and History of LLM
- Neural Networks and LLM (RNN, LSTMs, Transformers)
- Deep Dive – Architecture of LLM models
- Live Demo on LLM using transformer
- Types of LLM models
- Use cases
- Live Demo – Basic usage of GPT 2
- Challenges
- Quiz
- Q/A

What are Large Language Models?

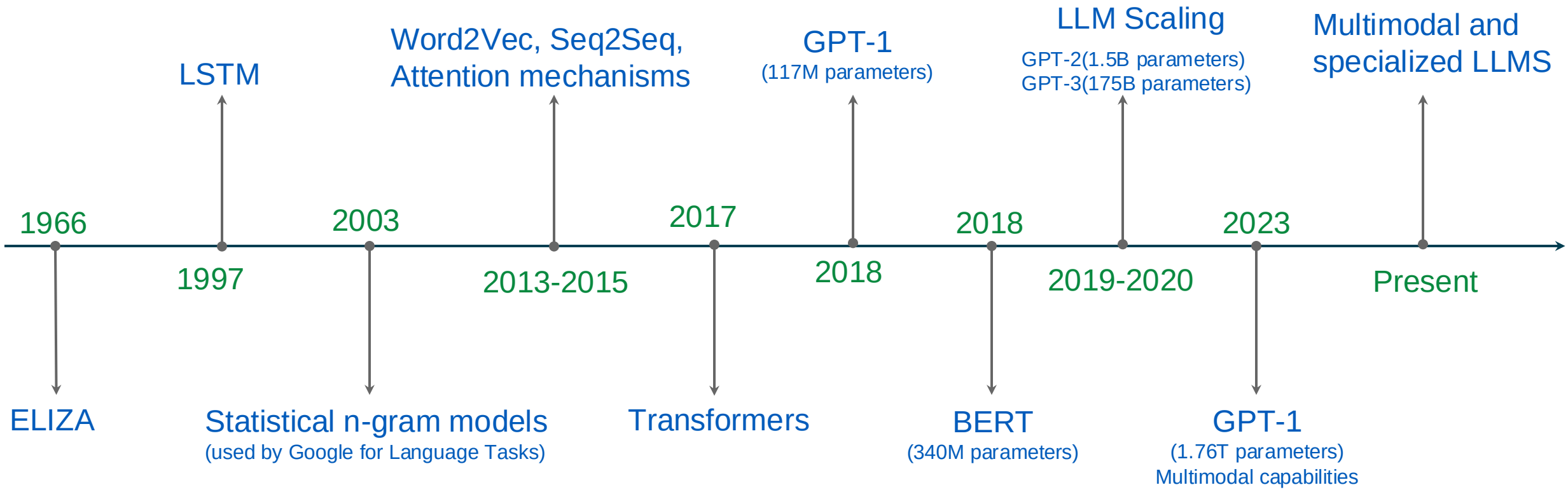
Definition: Type of Artificial Intelligence that uses deep learning techniques to process, understand and generate human language.

Foundation: Built using deep learning and neural networks

Training: Utilizes vast amounts of text datasets from internet, books, blogs, etc.

Utility: Ability to perform diverse set of language tasks like chatbots, text summarization, translation and more.

History of LLMs



University at Buffalo The State University of New York

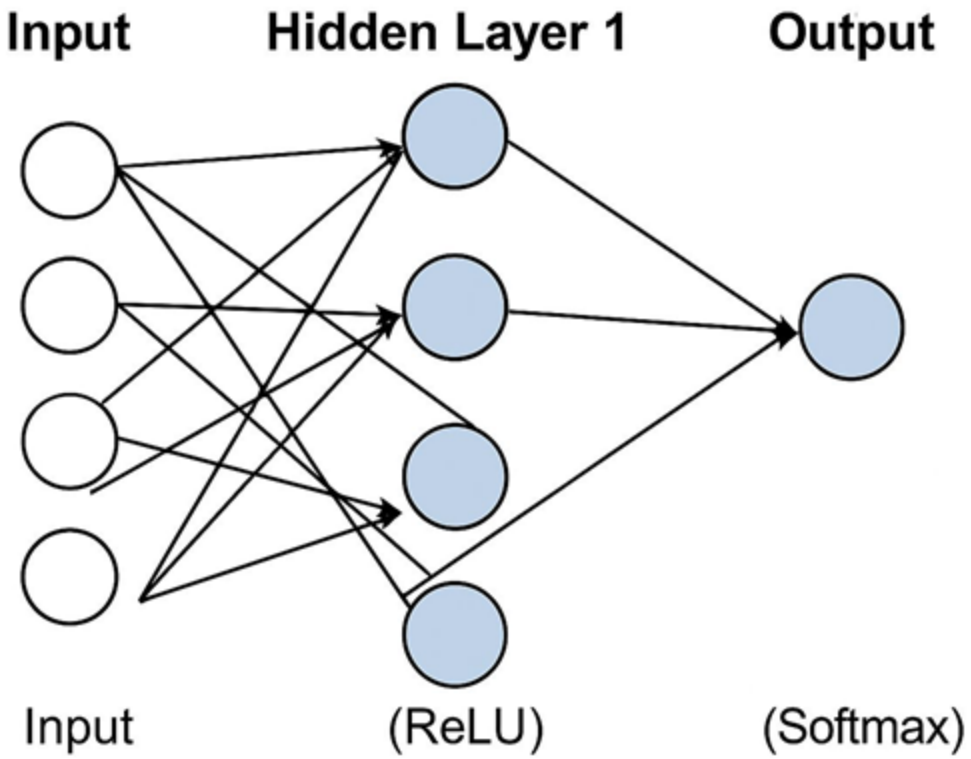
Neural Networks and LLM

Introduction to Neural Networks

Definition: System of connected nodes (neurons) inspired by the human brain

Commonly used in vision, speech and language tasks

Utility: Ability to perform diverse set of language tasks like chatbots, text summarization, translation and more.



From Traditional Neural Networks to LLMS

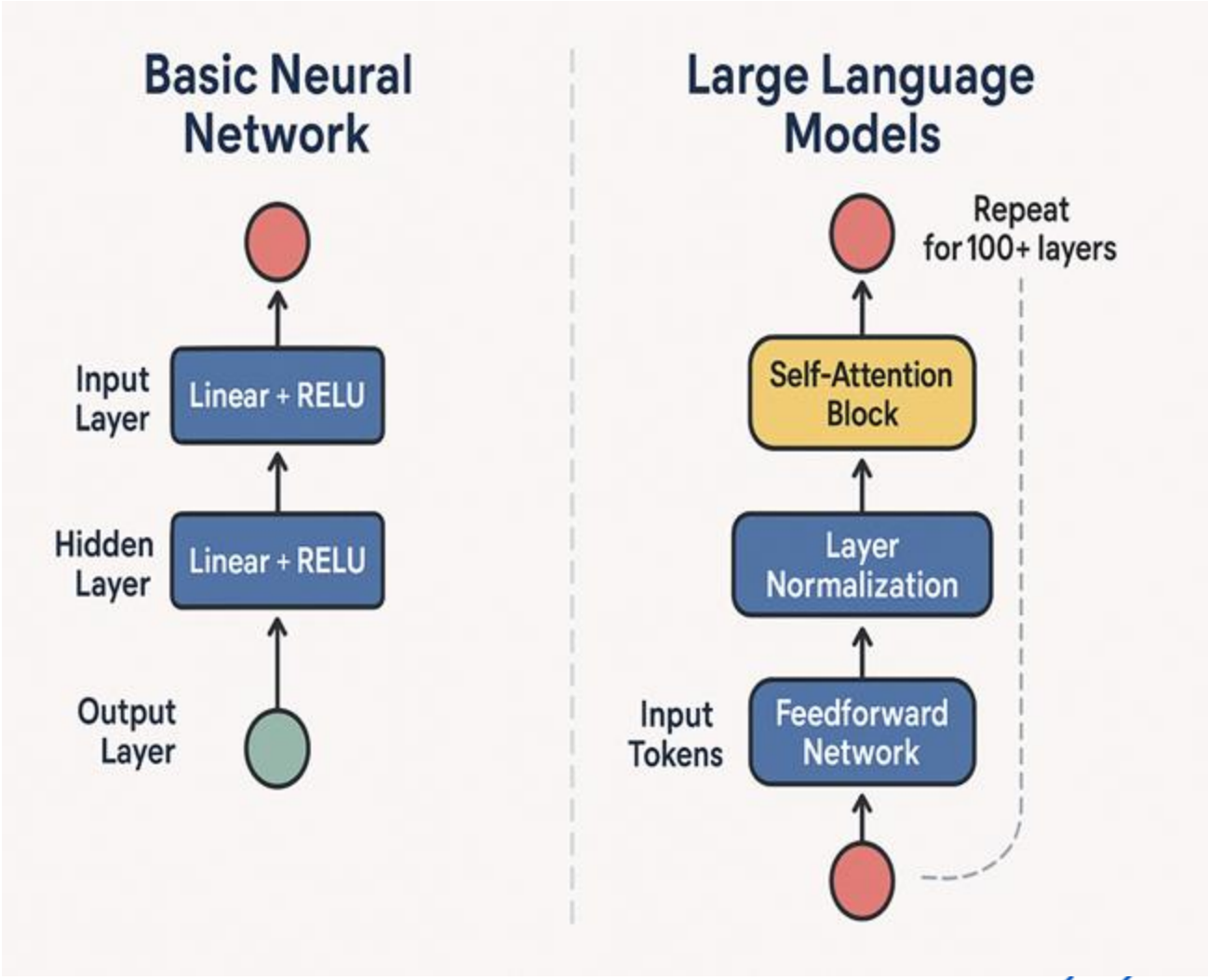
Issue → Traditional NNs work with fixed-size inputs → Not ideal for sequences

Solution : Use sequence-aware models

RNNs → capture previous state

LSTMs → handle longer dependencies

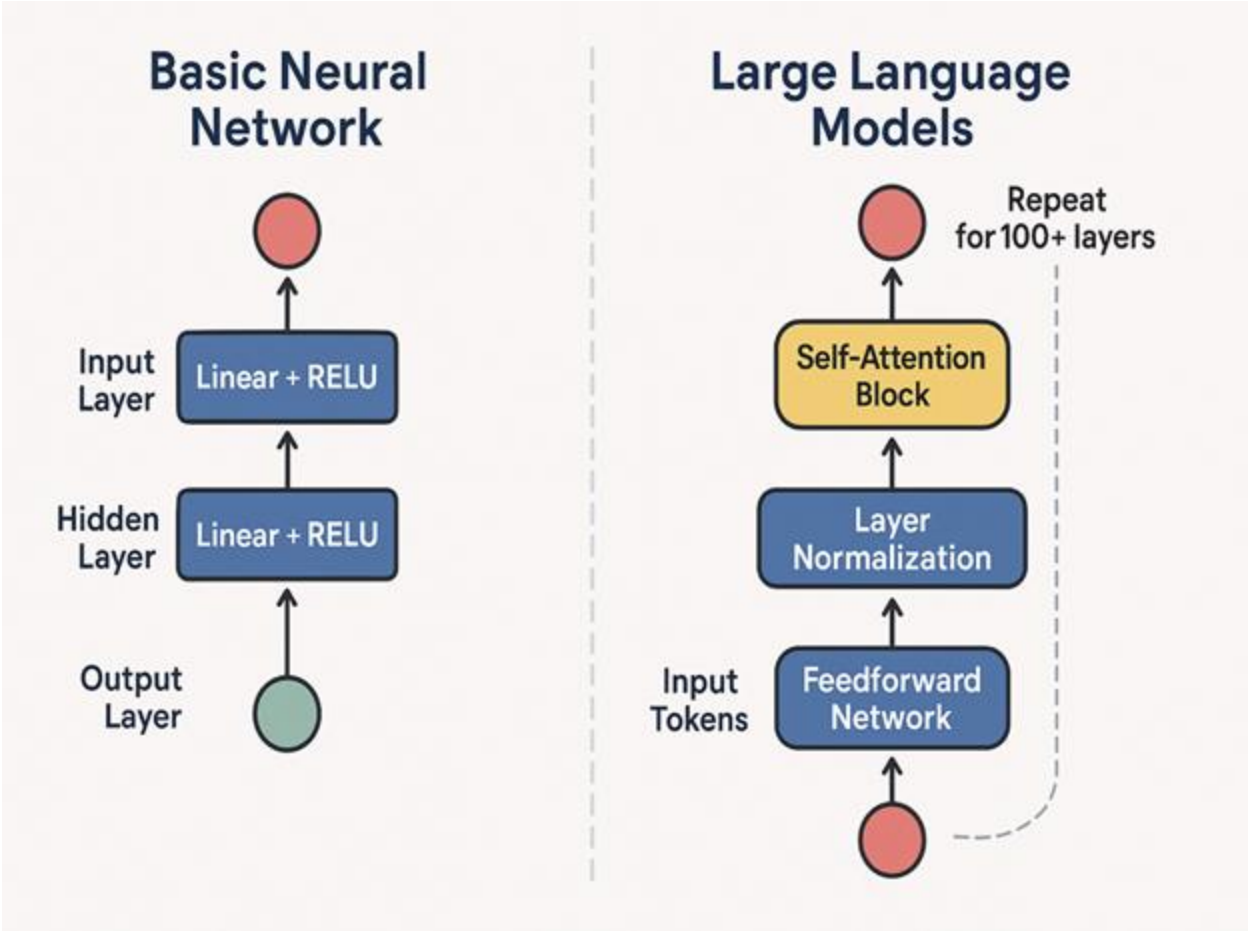
Transformers → attend to all positions



From Traditional Neural Networks to LLMS

Large Language Models are Deep Neural Networks

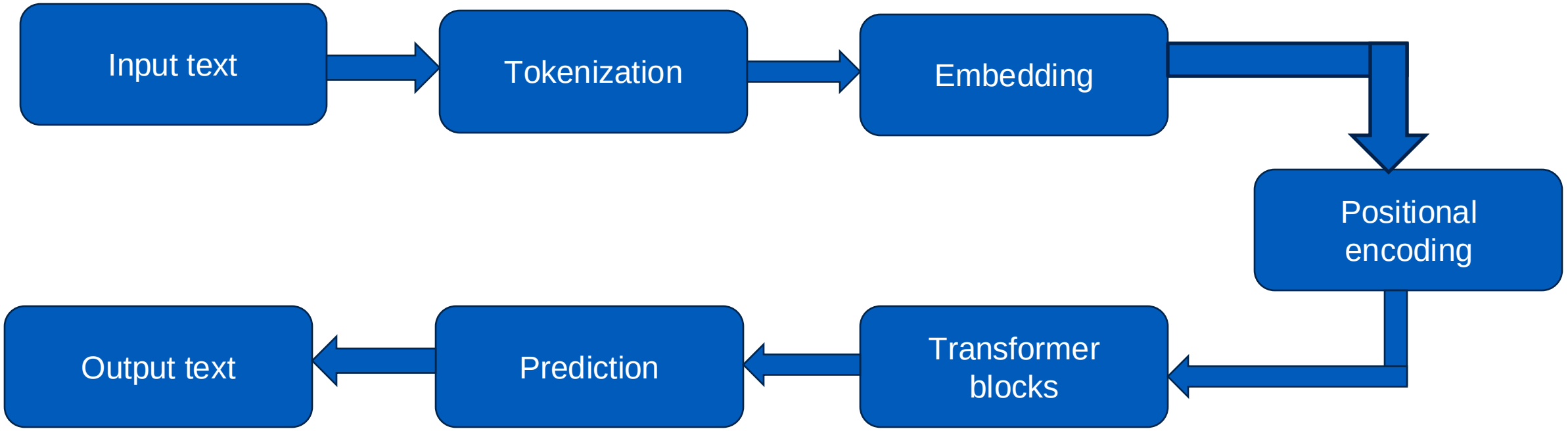
LLMs (like GPT, BERT) are deep neural networks with 100s of layers



University at Buffalo The State University of New York

WORKING OF LLMS

Pipeline


Input text
Tokenization
Embedding
Positional encoding
Transformer blocks
Prediction
Output text

Tokenization

Converts text into numerical tokens
(words, sub-words, or characters)

```
from transformers import GPT2Tokenizer
tokenizer = GPT2Tokenizer.from_pretrained("gpt2")
tokenizer.tokenize("unwatched cats")
```

✓ 0.1s

```
['un', 'w', 'atched', 'Ġcats']
```

Examples using GPT-2 tokenizer and BERT

"unwatched" → "un", "watch", "ed" (sub-word)

"cats" → "cat", "s"

```
from transformers import BertTokenizer
tokenizer = BertTokenizer.from_pretrained("bert-base-uncased")
tokens = tokenizer.tokenize("unwatched cats")
print(tokens)
```

✓ 0.1s

```
['un', '##watch', '##ed', 'cats']
```

Tokenization

GPT-2's tokenizer treats spaces as meaningful and sometimes splits words in unusual ways at the byte level.

```
from transformers import GPT2Tokenizer
tokenizer = GPT2Tokenizer.from_pretrained("gpt2")
tokenizer.tokenize("unwatched cats")
```

✓ 0.1s

```
['un', 'w', 'atched', 'Ġcats']
```

In BERT's tokenizer, ## indicates token is a sub-word

```
from transformers import BertTokenizer
tokenizer = BertTokenizer.from_pretrained("bert-base-uncased")
tokens = tokenizer.tokenize("unwatched cats")
print(tokens)
```

✓ 0.1s

```
['un', '##watch', '##ed', 'cats']
```

Embedding

Converts token IDs (e.g., 210 → "the") into dense vectors

These vectors capture semantic meaning (e.g., "king" and "queen" are clos

```
from transformers import GPT2Tokenizer, GPT2Model
tokenizer = GPT2Tokenizer.from_pretrained("gpt2")
model = GPT2Model.from_pretrained("gpt2")

inputs = tokenizer("I love pizza", return_tensors="pt")
outputs = model(**inputs)
print(outputs.last_hidden_state.shape)
```

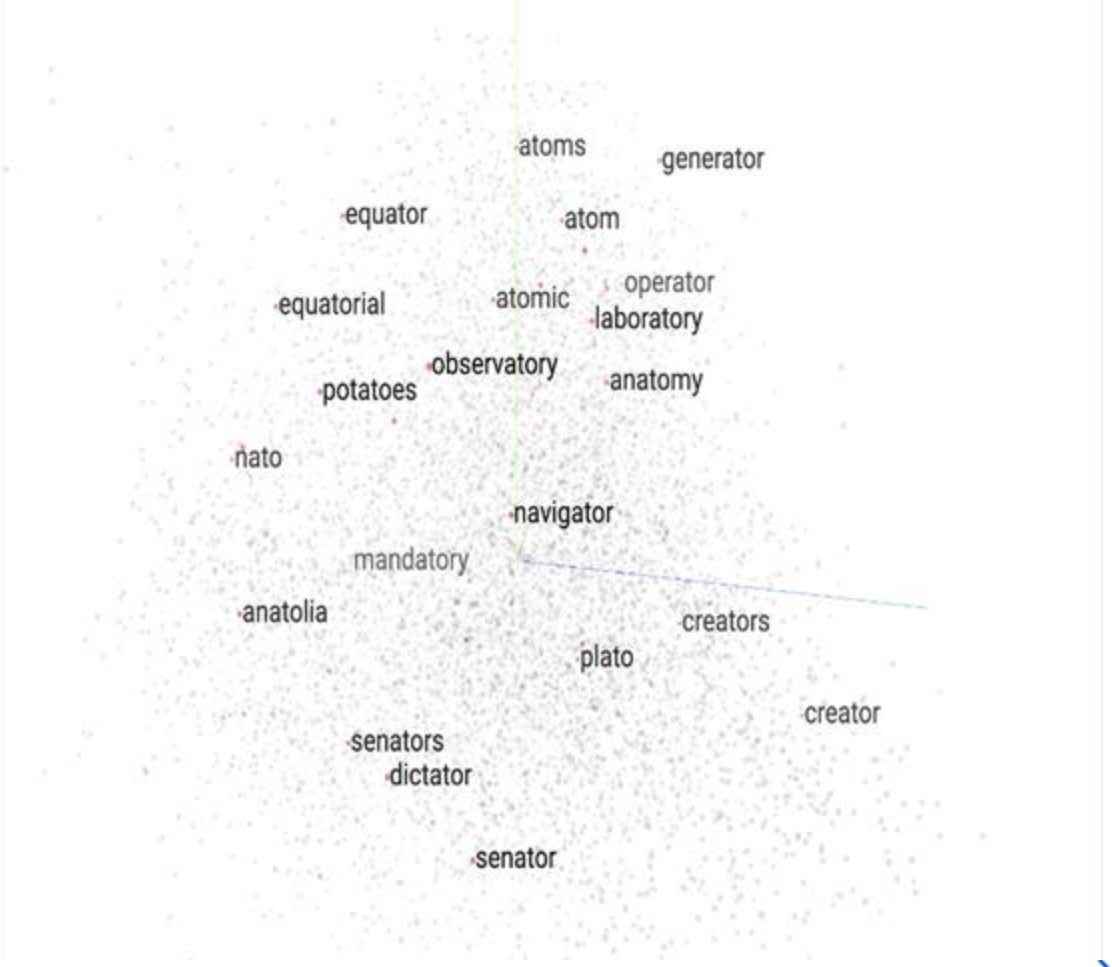
✓ 0.8s

```
torch.Size([1, 3, 768])
```

Embedding

Perform the following tasks

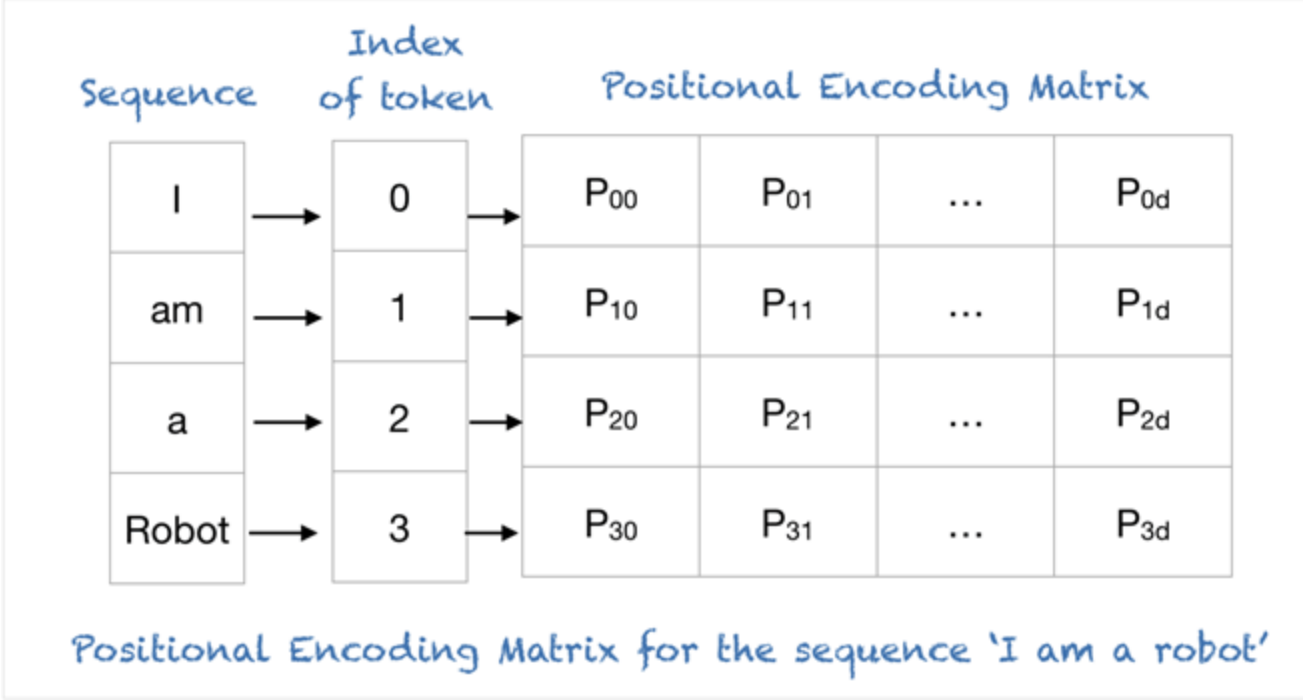
- ✓ Open the Embedding Projector tool at https://projector.tensorflow.org/
- ✓ In the right panel, enter the word **atom** in the **Search** field.
- ✓ Then click the word **atom** from the results below (under **4 matches**).



Positional Encoding

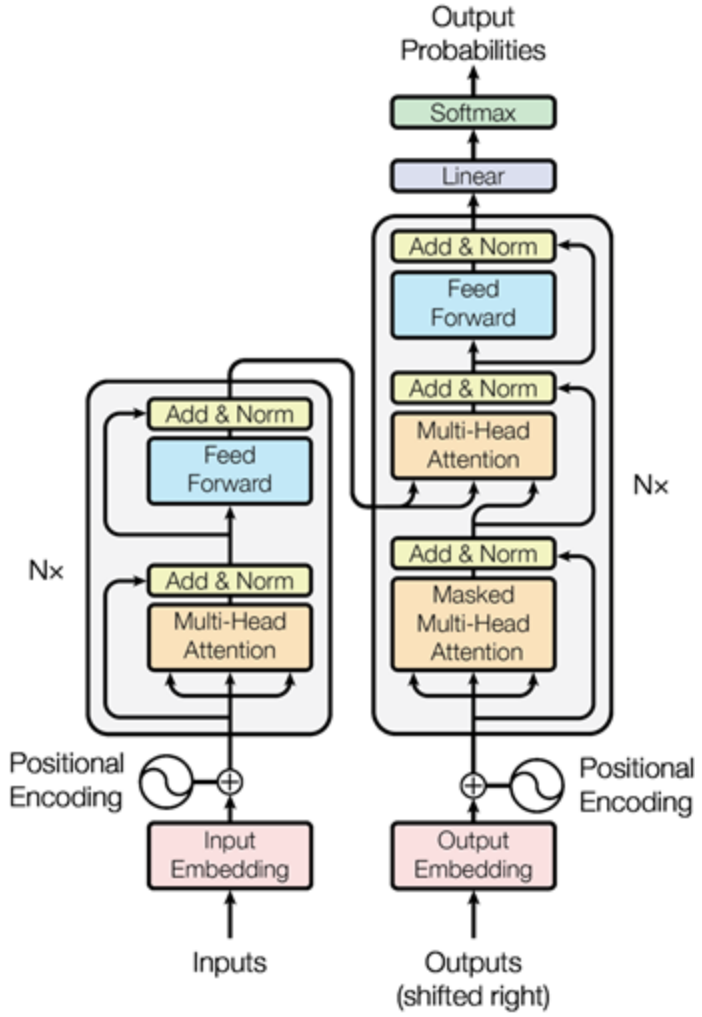
Why Do We Need Position?

- ✓ Transformers Are Orderless
- ✓ Self-attention has no built-in order
- ✓ Example "Cats eat fish" ≠ "Fish eat cats"



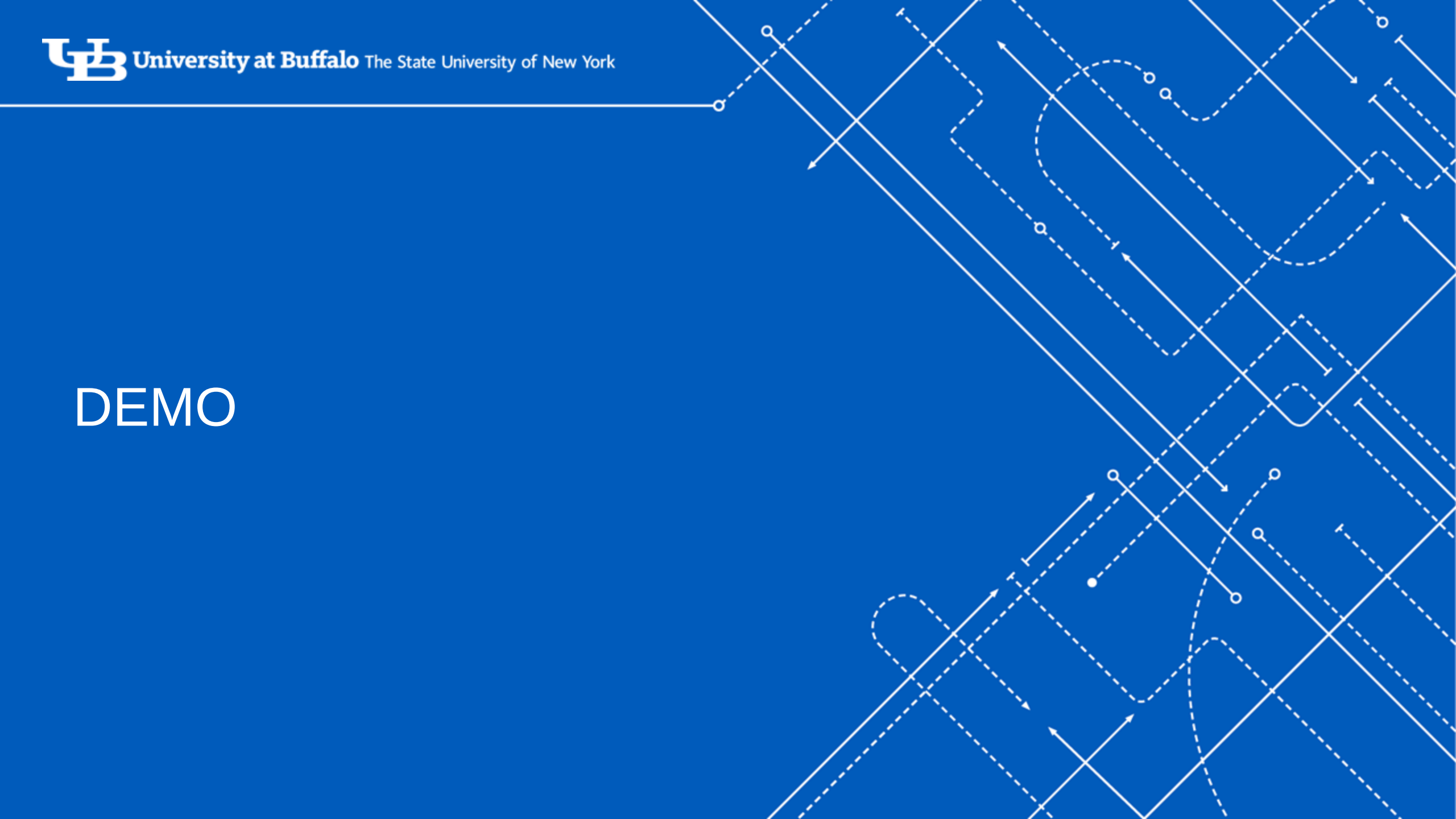
Architecture of LLM

- Based on Transformer architecture
  - Key component:
    - Embedding layer
- Self-attention mechanism
- Feed-forward neural networks
  - Encoder-decoder or decoder only designs
  - Billions of parameters



University at Buffalo The State University of New York

DEMO



University at Buffalo The State University of New York

TYPES OF LLM

Types of LLMs

Language Representation Models
(GPT, BERT, RoBERTa)

Zero-Shot Models
(GPT-3)

Multimodal Models
(DALL-E2)

Fine-Tuned Models
(Medical chatbots)

Selfie Time!!!!!

University at Buffalo The State University of New York

USECASES

USECASES

AI Assistants

Semantic Search with embeddings

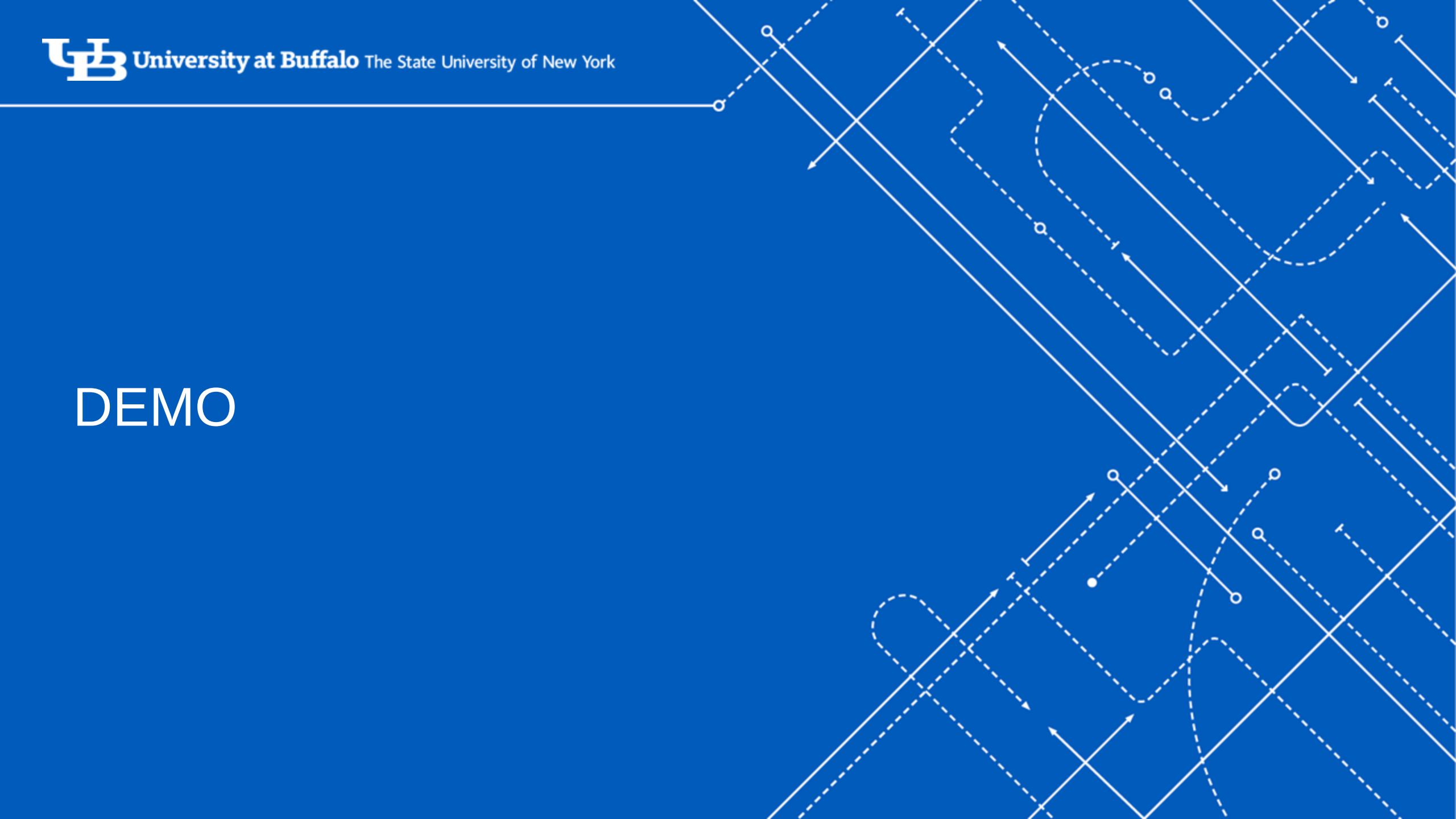
Text summarization

Sentiment analysis

Procurement management

University at Buffalo The State University of New York

DEMO

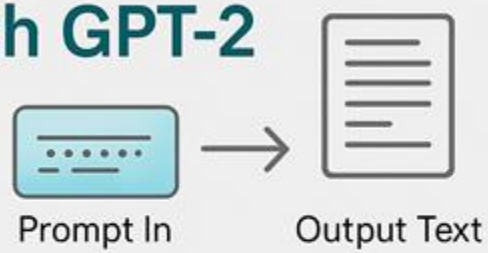


Prompt Engineering

Prompt Engineering with GPT-2

What is Prompt Engineering?

Crafting effective input prompts to guide the language model's output



Why It Matters in GPT-2?

GPT-2 is not instruction-tuned.

Output depends heavily on prompt phrasing

No fine-tuning = Smart prompting is essential

Techniques:

Instructional Prompt: "Write a poem about summer."

Few-Shot Learning: Provide 1-2 examples in the prompt

Challenges:

- Requires prompt trial & error.
- Sensitive to slight wording changes
- Harder than with GPT-3+ models

Use Cases:

- Text completion
- Story generation
- Summarization
- Q&A
- Code synthesis

University at Buffalo The State University of New York

CHALLENGES OF LLM

CHALLENGES

Hallucination → Making up facts

Bias → Reflecting societal stereotypes in training data

Compute costs → Training huge models requires massive hardware and energy

Ethics → Risk of misuse in misinformation, impersonation, etc.

University at Buffalo The State University of New York

QUIZ

References

- https://cohere.com/blog/large-language-models?utm_source=google&utm_medium=cpc&utm_campaign=fy25_amer_1_awareness_paidsearch_22285080165_179273636361_737093627743&utm_term={querystring}&gad_source=1&gclid=Cj0KCQjwna6_BhCbARIsALId2Z3_WHKixAhHFECaRxC0JFxGhy-Ix4XI-ItRlnmmI3u40-ZZ1rv8DJ0aArWQEALw_wcB
- https://www.eweek.com/artificial-intelligence/large-language-model/?ref=cohere-ai.ghost.io
- https://cohere.com/blog/say-hello-to-precision-how-rerankers-and-embeddings-boost-search?ref=cohere-ai.ghost.io&_gl=1*mcqtq4*_up*MQ..*_gs*MQ..&gclid=Cj0KCQjwna6_BhCbARIsALId2Z3_WHKixAhHFECaRxC0JFxGhy-Ix4XI-ItRlnmmI3u40-ZZ1rv8DJ0aArWQEALw_wcB
- https://www.eweek.com/artificial-intelligence/large-language-model/?ref=cohere-ai.ghost.io
- https://machinelearningmastery.com/a-gentle-introduction-to-positional-encoding-in-transformer-models-part-1/

University at Buffalo The State University of New York

Q & A