Building & Implementing a Simple AI/ML Chatbot

BevHacks 2025 Intermediate Workshop

colab.research.google.com

Overview

1. Intro
    a. What we will cover
    b. The goals of this workshop
2. Building a simple chatbot w/out memory
    a. Coding a simple chatbot using Transformers w/out memory
3. Introduction to AI Models & Transformers
    a. Making an AI Model step-by-step
    b. What we will be using—Hugging Face Transformers
    c. Popular Transformer Models
4. What is AI? What is ML?
    a. Artificial intelligence (AI) broken-down
    b. Artificial intelligence (AI): The big picture
    c. Machine learning (ML): A subset of AI

1. Learn about mathematical/ML tensors
    a. In mathematics
    b. In ML
    c. Concluding tensors
2. Building a simple rule-based chatbot
    a. Coding a short controlled bot
3. Building a simple chatbot w/memory
    a. Coding a simple chatbot w/memory
4. Applications

1. Introduction

What will you achieve at the end of this workshop? What will you learn? What will we cover?

Goals

- Build 3 chatbot(s) that can respond to simple questions (simplified, intermediate, advanced)
- Understand applications
- Mathematical tensors

2. A simple rule-based chatbot prototype

Learn how-to build a basic chatbot that responds to specific inputs

Grasping the fundamental logic of how (AI) chatbots work

```
[ ]  responses = { # Here, we are defining a dictionary of questions and responses
         "hello": "Hi there! How can I help you?",
         "how are you": "I'm just a bot, but I'm functioning as expected!",
         "bye": "Goodbye! Have a great day!",
     }
```

```
[ ]  def chatbot(input_text): # We are using a function called chatbot to handle the input from the user
         input_text = input_text.lower()
         return responses.get(input_text, "Sorry, I don't understand that.")
```

```
user_input = input("You: ") # Now, we can test it interactively
print("Bot:", chatbot(user_input))
```

```
...  You:
```

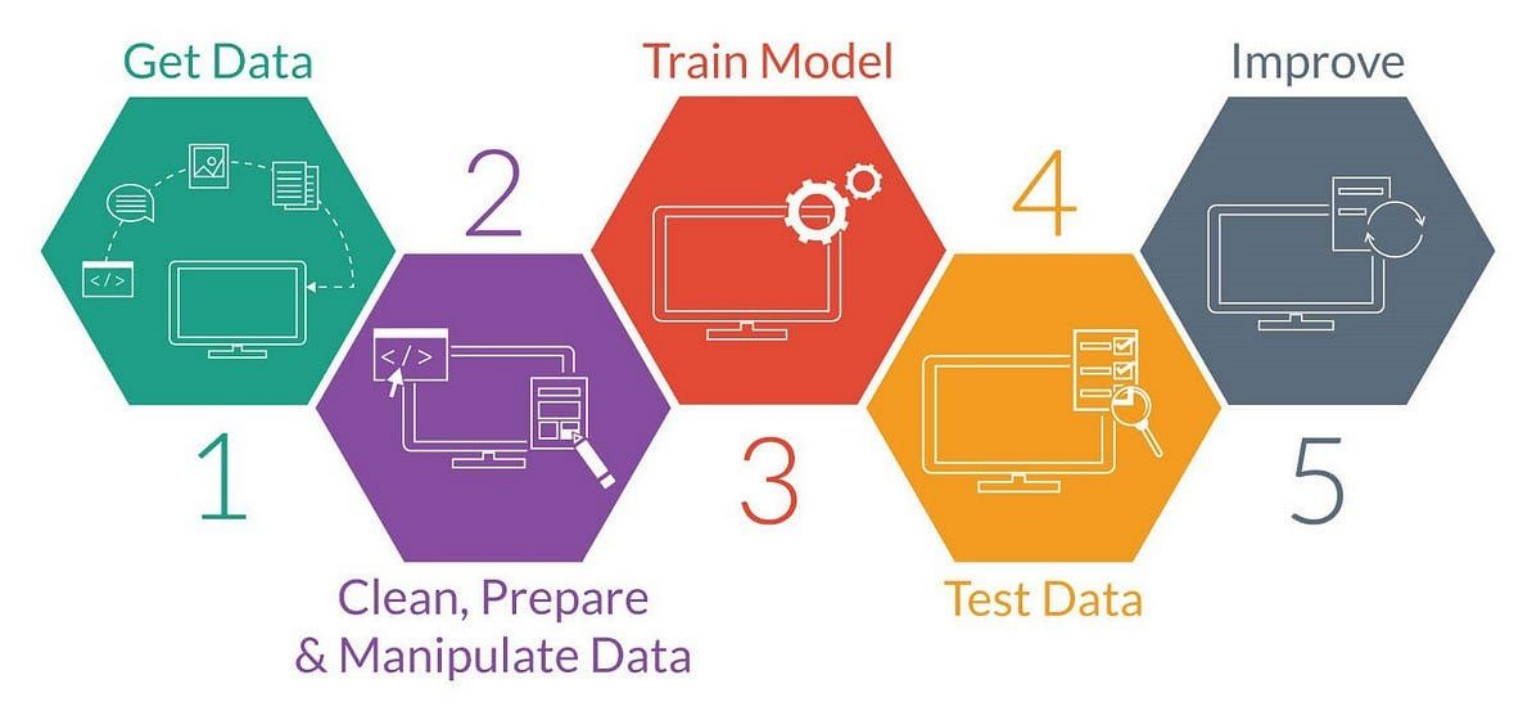
2. A simple rule-based chatbot prototype

3. Introduction to AI Models & Transformers

AI Models Step-By-Step, Hugging Face, Transformer Models

Making an AI Model step-by-step

Step 0: Define goal



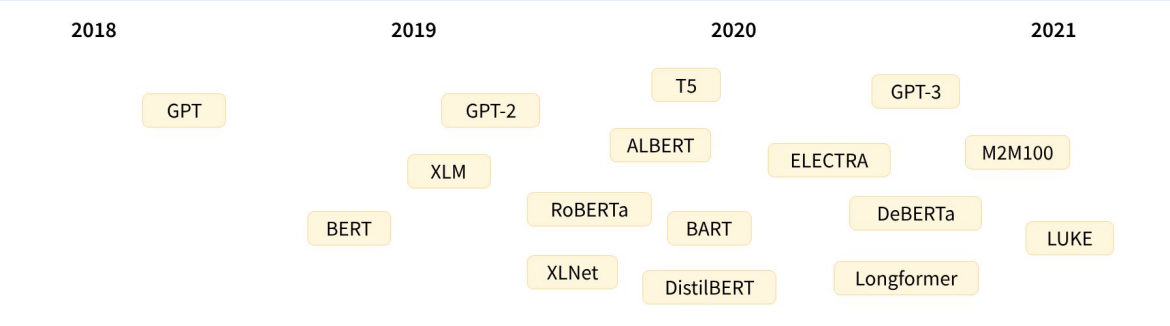
What we will be using—Hugging Face Transformers

Hugging Face

- NLP stands for natural language processing
- Transformers, machine learning library provided by Hugging Face
  - They excel at processing sequences of data, like text, by paying selective “attention” to different parts of the sequence
- Hugging Face, toolbox of AI models, saving time and computational resources

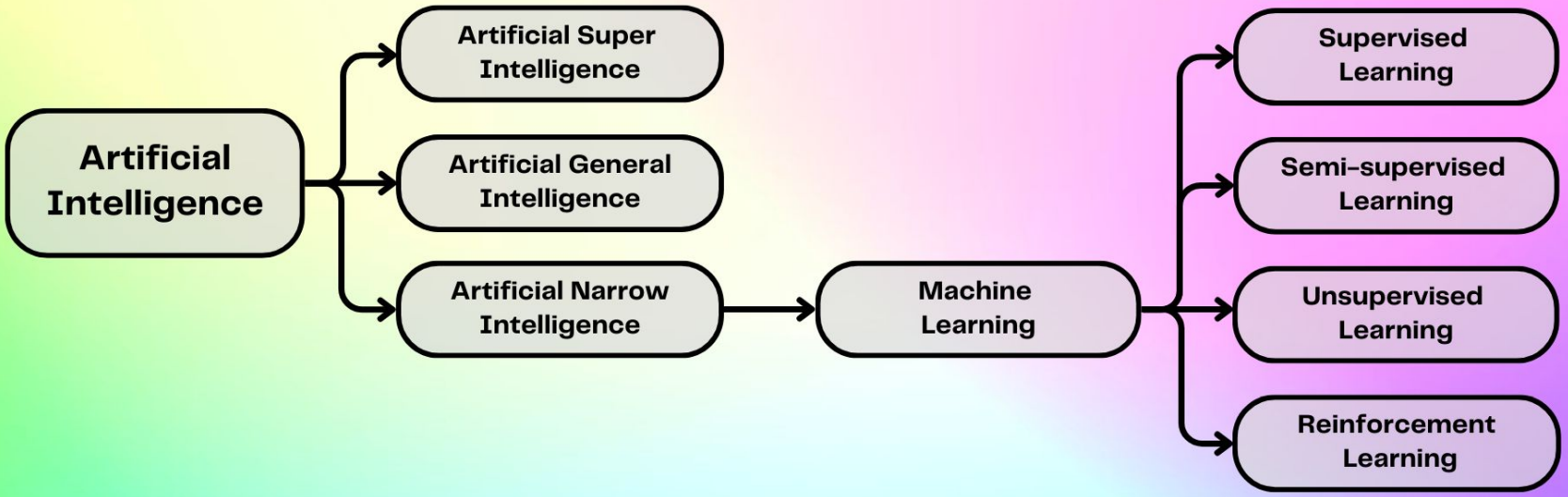
Popular Transformer Models

- GPT
  - Used in OpenAI's ChatGPT
- BERT
  - Focused on understanding text
- DialoGPT ★
  - Conversational tasks



4. What is AI? What is ML?

Artificial Intelligence (AI); Machine Learning (ML)


Artificial Intelligence
Artificial Super Intelligence
Artificial General Intelligence
Artificial Narrow Intelligence
Machine Learning
Supervised Learning
Semi-supervised Learning
Unsupervised Learning
Reinforcement Learning

Artificial Intelligence: The Big Picture

- Broader field of creating MACHINES or SYSTEMS that can perform tasks that require HUMAN-LIKE INTELLIGENCE
- Artificial intelligence includes…
  - …learning from data (ML)
  - …reasoning and decision-making
  - …perception (e.g., vision and speech)
  - …natural language understanding/processing—NLP
- Examples of AI:
  - Virtual assistants (e.g., Siri or Alexa)
  - Recommendation systems (e.g., on Netflix, Amazon, YouTube)

Machine Learning: A Subset of AI

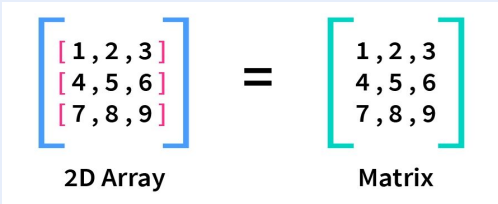
- It is a SUBSET OF AI focused on DEVELOPING SYSTEMS that can LEARN and IMPROVE from DATA W/O being explicitly programmed to do so
- How does ML actually work?
  - Supervised Learning
    - Uses labeled data sets to predict outcomes and recognize patterns
    - Given labeled training to learn relationship between inputs and outputs
  - Unsupervised Learning
    - Learn from unlabeled data without explicit instruction or guidance
  - Semi-Supervised Learning
    - Uses both labeled data and unlabeled data
  - Reinforcement Learning
    - Trains software to make decisions to achieve the most optimal results

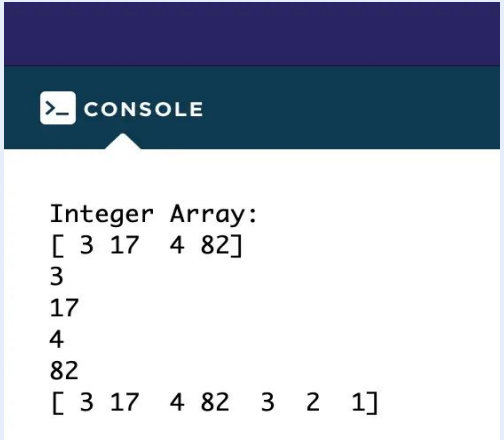
5. Mathematical/ML Tensors

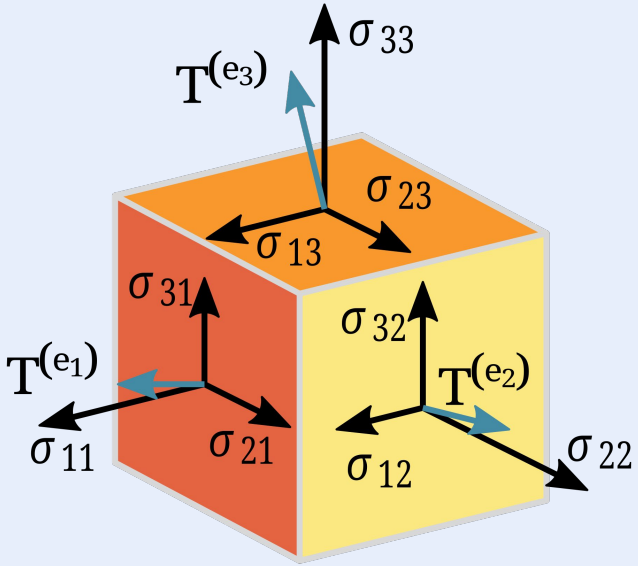
What are tensors?

Tensors in Context (Mathematics)

- **torch** is a **library** for work **with tensors** (multi-dimensional arrays) and **performing operations** on them
- Tensors are **multi-dimensional arrays representing complex data** and describing **multilinear relationships** between sets of algebraic **objects**, often within a **vector space** where **vectors** can be added and scaled by numbers called **scalars**







(above) is the second-order Cauchy [kow-shee] stress tensor (T) that describes the state of stress experienced by a material at a given point in the deformed state, placement, or configuration

Tensors in Machine Learning

- In machine learning (e.g., PyTorch models like the chatbot) and continuum mechanics (a branch of physics studying how materials behave under forces), tensors share similarities as **mathematical objects**
- However, their **applications and interpretations differ significantly**: in machine learning, tensors handle data for computations, while in continuum mechanics, they represent quantities like stress in materials treated as continuous, rather than focusing on individual atoms or molecules.

Concluding Tensors

Multi-dimensional nature:

- A tensor’s rank indicates its dimensionality:

  In ML: a tensor can be 1D (vector), 2D (matrix), or higher (e.g., images, sequences)

6. Real World Applications

How may we apply this to the real-world?

Real World Applications

Health

- Diagnosing
- Research

Agriculture

- Identification of plants
- Automated plant monitoring

Education

- Personalized lessons

Retail/preference

- Analyzes patterns to predict what you would like

Digital assistant

- Siri
- Alexa
- Google Assistant