Welcome to the Class

Llama for Python Programmers

Llama 2 Benefits

- Available under a royalty free license
- Suitable for use in highly regulated industries or where data is sensitive
- Prototyping development with minimum cost
- Competitive with other commercial offerings
- Available to be scaled as a service on major cloud platforms

This Course

- Uses the Coursera Labs virtual environment
  - Allows for practice while learning without any software installation
  - Uses Jupyter notebooks in VS Code
- Focuses on programmatic interaction with llama 2 through python and llama.cpp

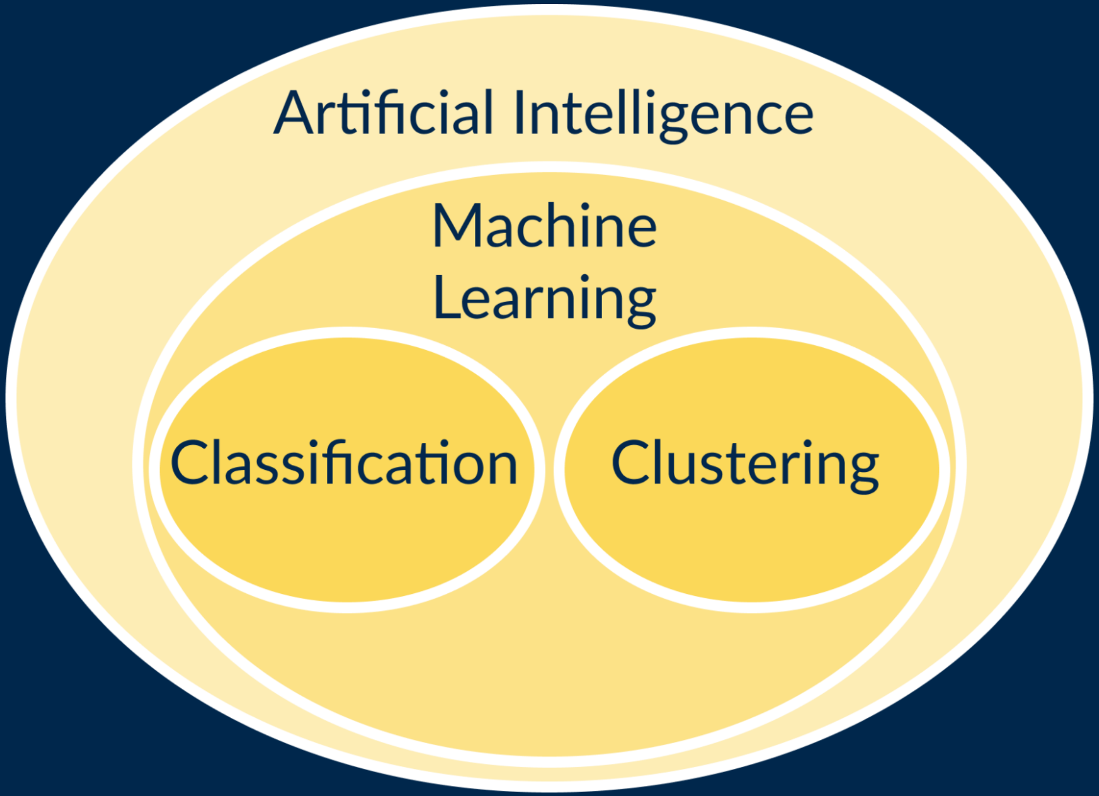
Credits:

Christopher Brooks
Associate Professor
School of Information

What Is Llama 2?

An open source Large Language Model (LLM)

Family of AI methods



Artificial Intelligence

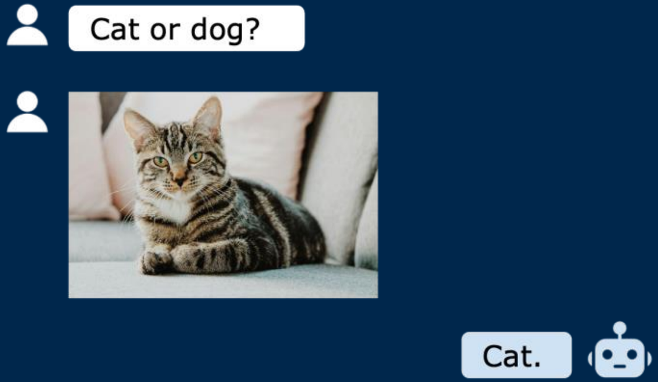
Machine Learning

Classification

Clustering

Classification

- Supervised learning
- Pre-trained with labeled data



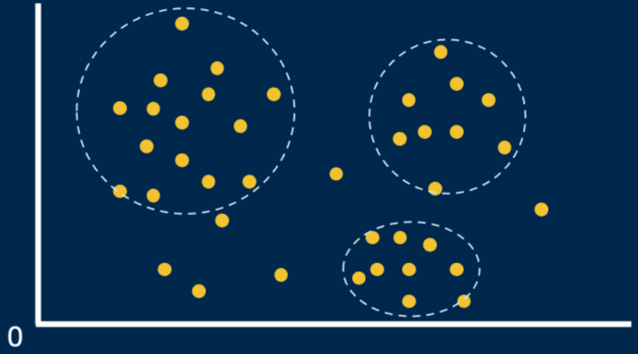
Cat or dog?

Cat.

Clustering

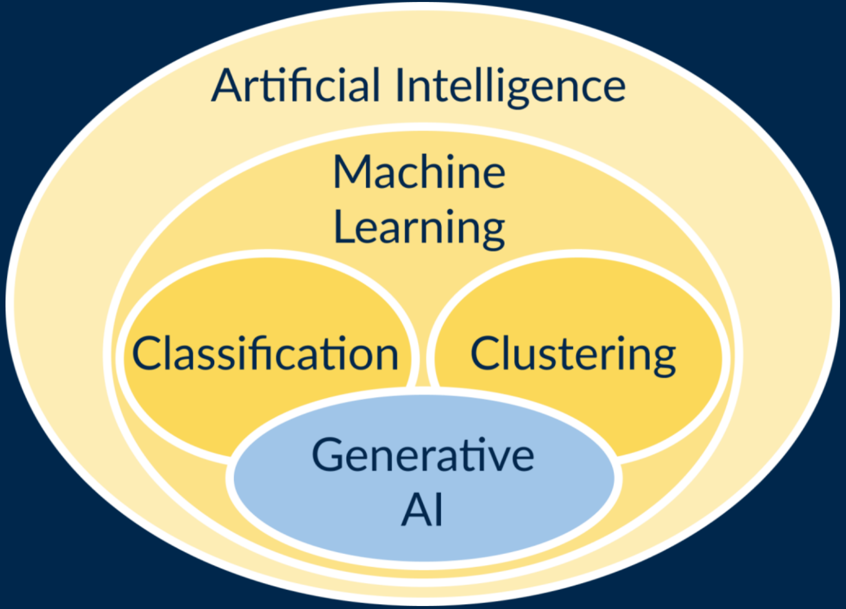
- Unsupervised learning
- Analysis of unlabeled data

Records of students and employees



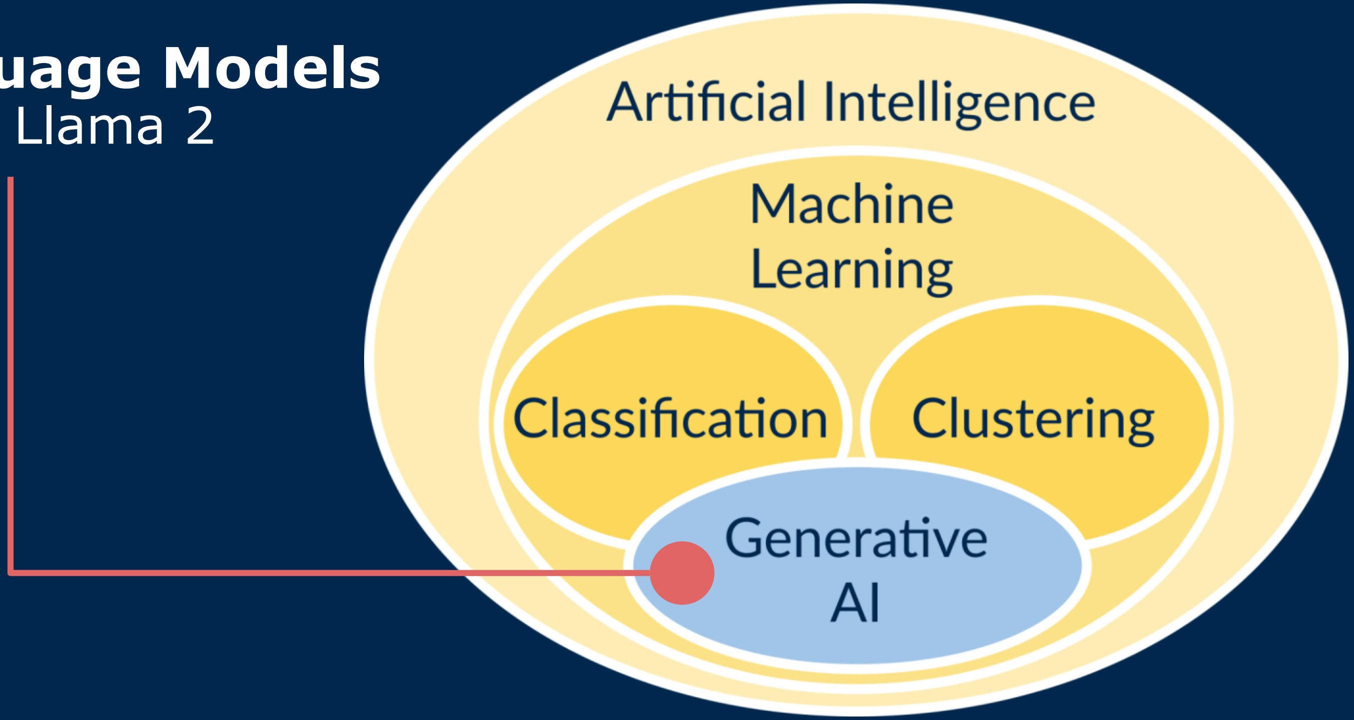
Generative AI

- It builds on top of these and other techniques from machine learning.
- It generates new output data in response to both an input data stream and a pre-trained corpus.



Given a black box model trained on large amounts of data, and a prompt to feed into that model, what is the ideal continuation (response) of that prompt?

Large Language Models
such as Llama 2



What is Llama 2

- A family of models with different complexities and capabilities
- High quality, free, and open sourced by Meta for redeployment and derivative work
- Tasks: base, chat, and code
- Sizes: 7B, 13B, 34B, 70B

How Llama 2 works?

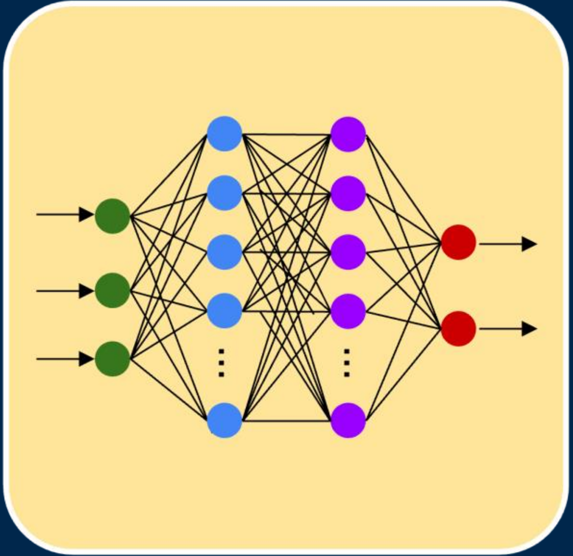
Llama 2 is pre-trained

- The model architecture is initialized with values from training data, including:
  - Wikipedia
  - Scraped webpages
  - Open Source code repositories
  - Free online books

Over **2 trillion** tokens used to train Llama 2!

A Pre-trained model

- The data structure has been filled with the weights of the model.
- **Weights:** Relationships between all of the pieces of text put into the model through pre-training.
- The quality of these weights determines how accurate, and thus good, the model is at predicting text.

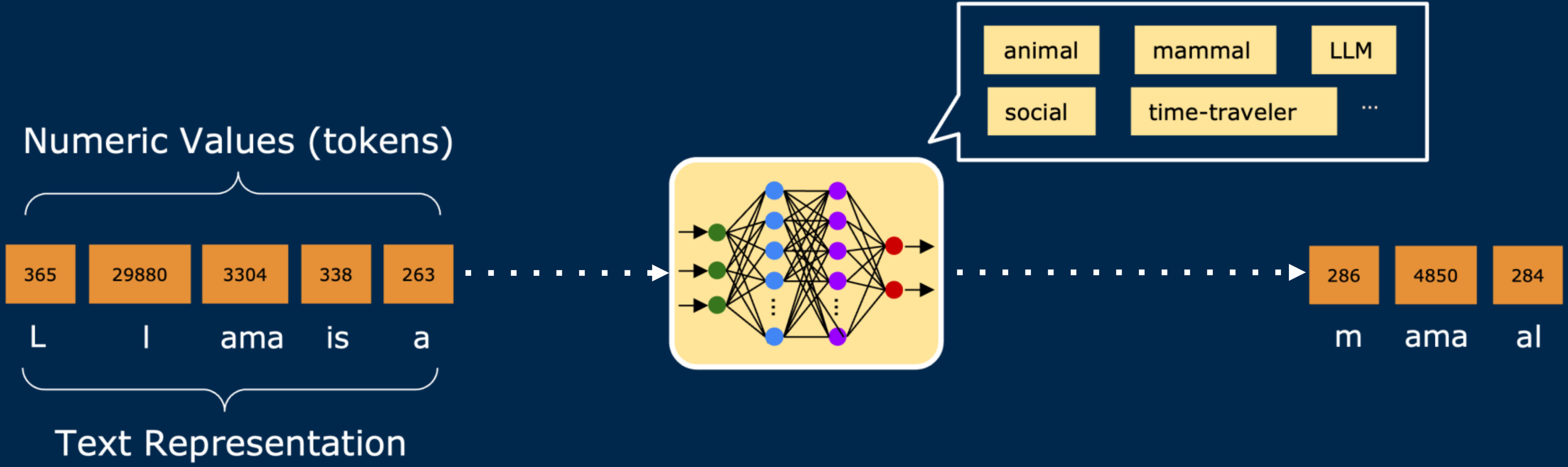


A complex data structure

Fine-tuning Llama 2

- Adjusting the weights of the model to be a little bit better
    - Use **small** amount of **high quality** data
    - Applying a supervised learning method
    - Tuning for different purposes, including output quality, helpfulness, safety, and more

Inference



Numeric Values (tokens)

365 29880 3304 338 263

L l ama is a

Text Representation

animal mammal LLM

social time-traveler ...

286 4850 284

m ama al

Prompt **Model** **Completion**

Sumerian Riddle

“A house based on a foundation like the skies,

A house one has covered with a veil like a (secret), tablet box,

A house set on a base like a 'goose',

One enters it blind,

Leaves it seeing”

18th century BC, translated by E.I. Gordon, 1960, Sumerian Riddles A Corpus (1987)

Credits:

Christopher Brooks
Associate Professor
School of Information

Llamas Llamas Everywhere!

Inference, chat, and code completion

Llama 2 Model Flavors

Base Llama

- Suitable for **inference**
- Pre-trained for factually correct responses

Chat Llama

- Suitable for Human interaction through dialog
- Fine-tuned for safety and non-toxicity

Code Llama

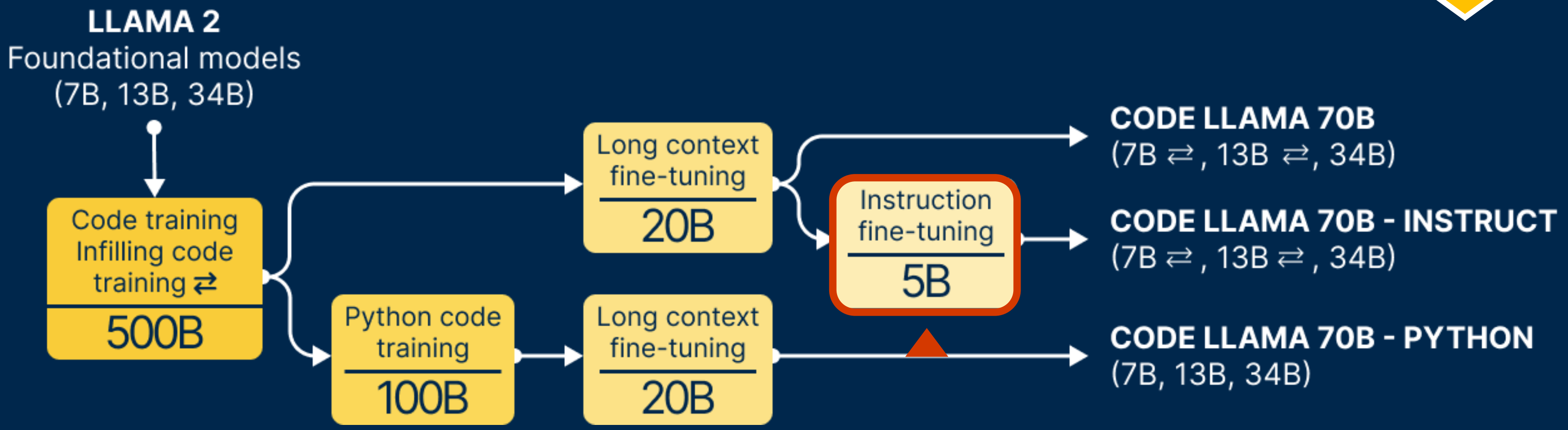
- Built specifically for programming tasks

Code Llama

- Three different model sizes: 7B, 13B, 34B parameters
- Joined in January 2024 with a 70B parameter code llama variant
- Fine-tuned variants:
    - Code completion
    - Infilling
- An additional 500B tokens were used to fine-tune these models, coming from web code resources

Meta fine-tuned their Code Llama models further

New in 2024, 70B code Llama!



Roziere, Baptiste, et al. "Code Llama: Open foundation models for code." arXiv preprint arXiv:2308.12950 (2023).

Llama's all the way down

- Code llama can produce code and we can run it to test the output
- So, let's "hire" the best model!
  - Llama 2 70B base model used to generate 52,000 programming interview questions
  - Code Llama had to solve this ten different times
  - Take the first solution and use it for fine-tuning

Takeaways

- Making an LLM isn't as simple as just throwing text at compute and waiting, you can fine-tune a model after the base is built in order to achieve specific objectives;
- Fine-tuning doesn't require nearly as much data as building a base model;
- While human preferences are often used with reinforcement learning for fine-tuning, this isn't the only approach!

Credits:

Christopher Brooks
Associate Professor
School of Information

Open Source LLMs

Llama and its competitors

What is “Open Source”?

- Open Source Initiative (OSI) defines “open source” for software.
  - It approves licenses which adhere to ten specific principles.
  - At a high level, being "open source" means that software should allow permissively redistribution.
- However, the software part of an LLM is really a small part of what makes it useful.

The Llama 2 Community License

- Meta provides **a royalty free license** to the code, the model weights, and the software.
- Meta also allows for companies to redistribute derivative works, like products and fine-tuned versions of models
- But there are a few caveats of the Llama 2 license which cause conflict with the OSI.

Caveats of Llama 2 license

1. You must follow the llama usage policy and laws, and explicitly no weapons development or unlicensed activities.

1. You must not use llama 2 to train other large language models.

1. If your organization has more than 700 million monthly users you must apply for a separate license from Meta.

A Growing List of Alternatives

- Meta raised the bar, and a growing list of new models are being released as open source:
  - Minstral 7B, Mixtral 8x7B
  - Microsoft Phi-2 (2.7B)
  - Falcon 40B, Falcon 180B

Credits:

Christopher Brooks
Associate Professor
School of Information

The Elephant in the Room

Running Large Language Models on small hardware

How can we actually run these large models on **commodity hardware**?

Model Size

- Pre-trained models are big, we can estimate the memory needed by taking the model size and multiplying it by 4 bytes, e.g.:
  - Llama 2 7B = 7,000,000,000 * 4 = 28 GB
  - Llama 2 14B = 14,000,000,000 * 4 = 56 GB
  - Llama 2 70B = 70,000,000,000 * 4 = 280 GB
- On top of this, the processing of the models is highly parallel and done on video cards (GPUs), so this isn't system memory (RAM) but video card memory (VRAM)

Quantization

- The process of reducing data size at the cost of precision
- Quick quantization is supported in most language model toolchains
  - Load the model weights as **16 bit floating point numbers** instead of **32 bit** and reduce VRAM usage by 50%
- Can we go further?
  - Yes!
  - And precision remains remarkably reasonable!

Moving off the GPU

GPUs

- Specialized for parallel processing of floating point numbers
- Great for inferences of LLMs
- Expensive!
- Limited with respect to memory

→

CPUs

- Well-suited for integer arithmetic
- Expansive amount of RAM
- Able to run model inference with less precision
- Significantly more affordable

llama.cpp

- llama.cpp, an open source tool started by Georgi Gerganov to run large language models on commodity hardware
- Features include:
  - Fine-tuning models
  - Support for mixture of CPU and GPU hardware
  - State of the art quantization methods allowing for mixed levels of precision
  - Support for non-Nvidia GPUs
  - Language bindings for python, go, node, java, rust and more!

Credits:

Christopher Brooks
Associate Professor
School of Information