Lecture 1: Introduction

https://github.com/kaopanboonyuen/SC310005_ArtificialIntelligence_2025s1

Teerapong Panboonyuen
https://kaopanboonyuen.github.io

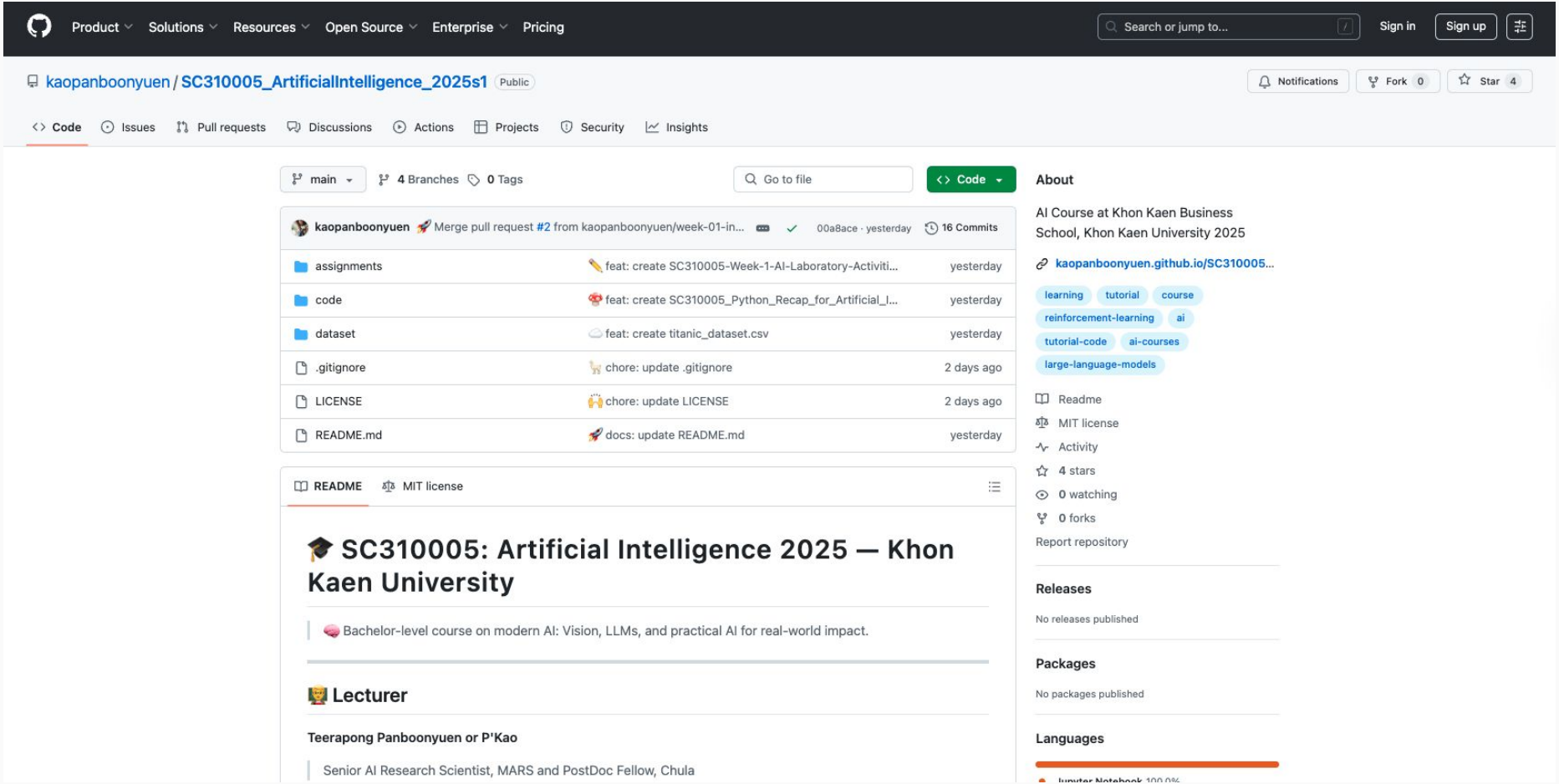
Course Instructor:

Teerapong Panboonyuen (P'Kao)

🎓 **Ph.D. in Computer Engineering** – Chulalongkorn University
🔬 **Research Scientist @ MARS**
📚 **Postdoctoral Researcher (C2F) @ Chula**

Contact: teerapong.pa@chula.ac.th

https://github.com/kaopanboonyuen/SC310005_ArtificialIntelligence_2025s1



What is Artificial Intelligence (AI)?

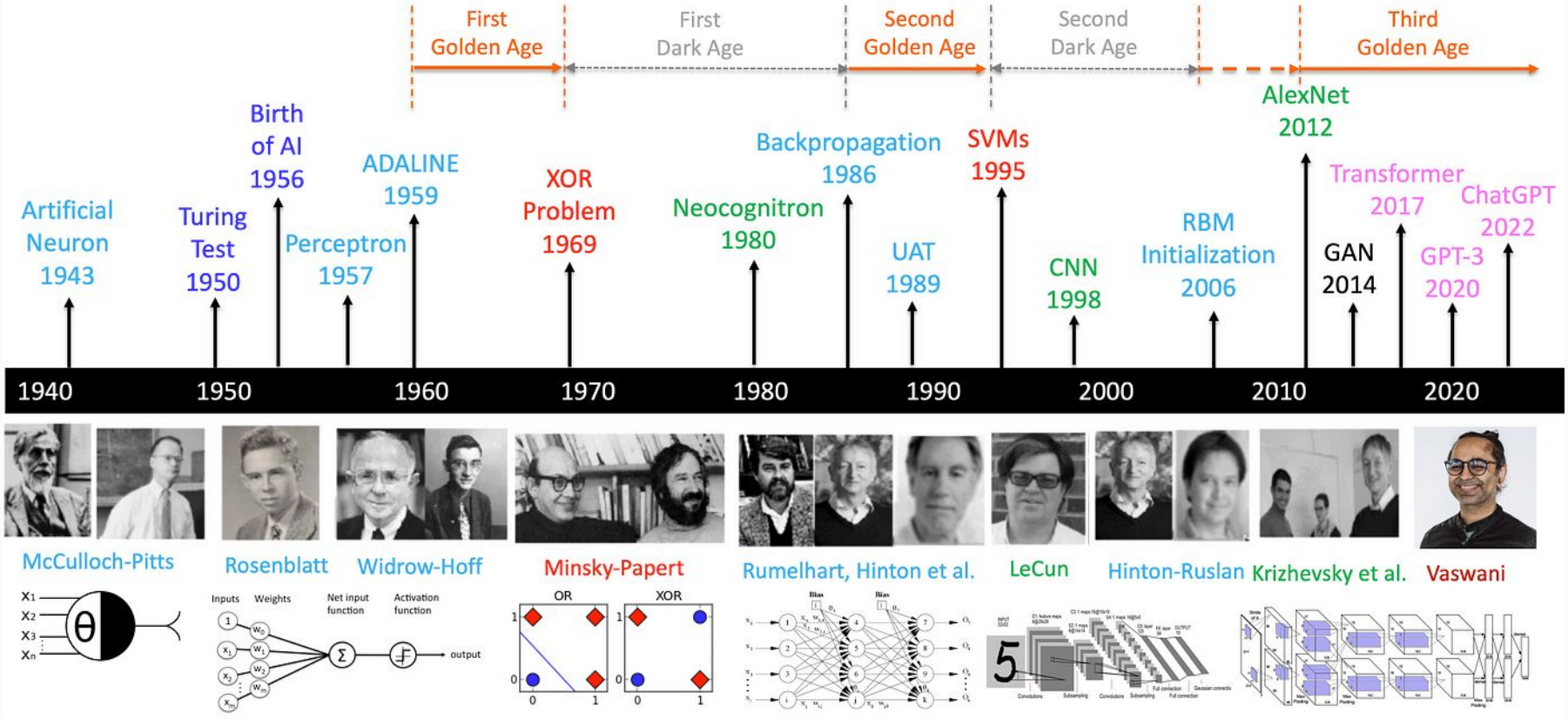
Definition: AI refers to the simulation of human intelligence in machines.

- **Key Areas of AI**: Perception, Learning, Reasoning, Problem Solving, Decision Making.
- **Examples of AI in Real Life**: Siri, Google Search, autonomous vehicles, recommendation systems.

History of AI:

- Early AI (1950s-1980s): Symbolic AI, expert systems.
- Modern AI (2000s-present): Data-driven, deep learning, and neural networks.

A Brief History of AI with Deep Learning



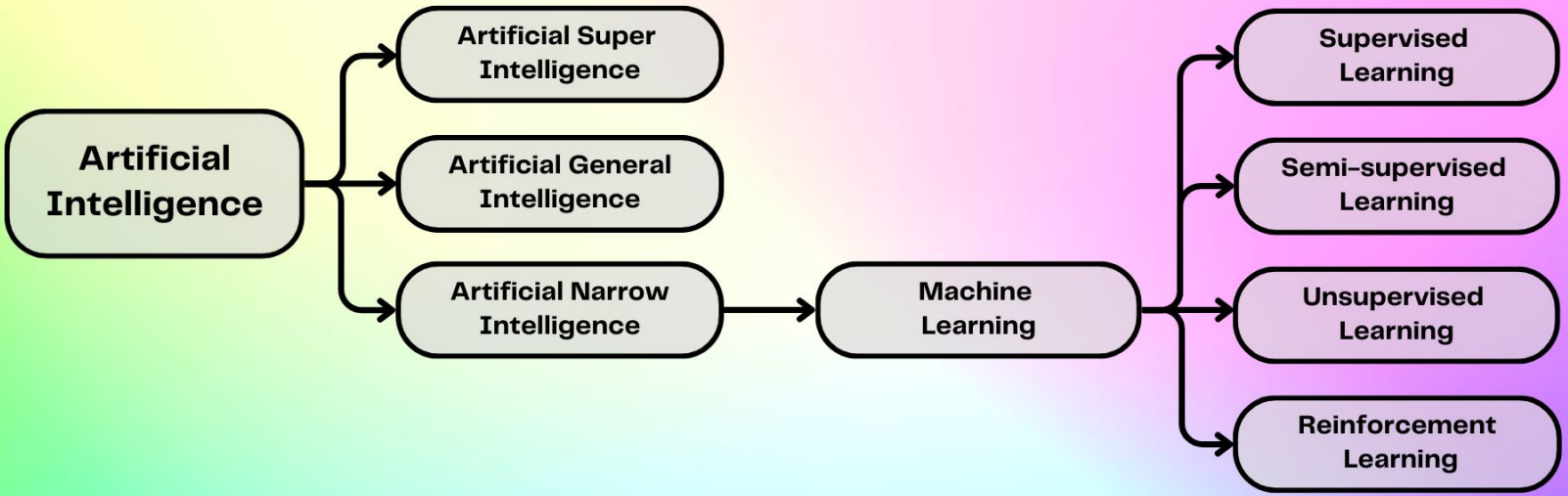
Key AI Technologies Today

Machine Learning (ML):

- The backbone of modern AI: teaches machines to learn from data.
- **Key Types**: Supervised learning, unsupervised learning, reinforcement learning.

Deep Learning:

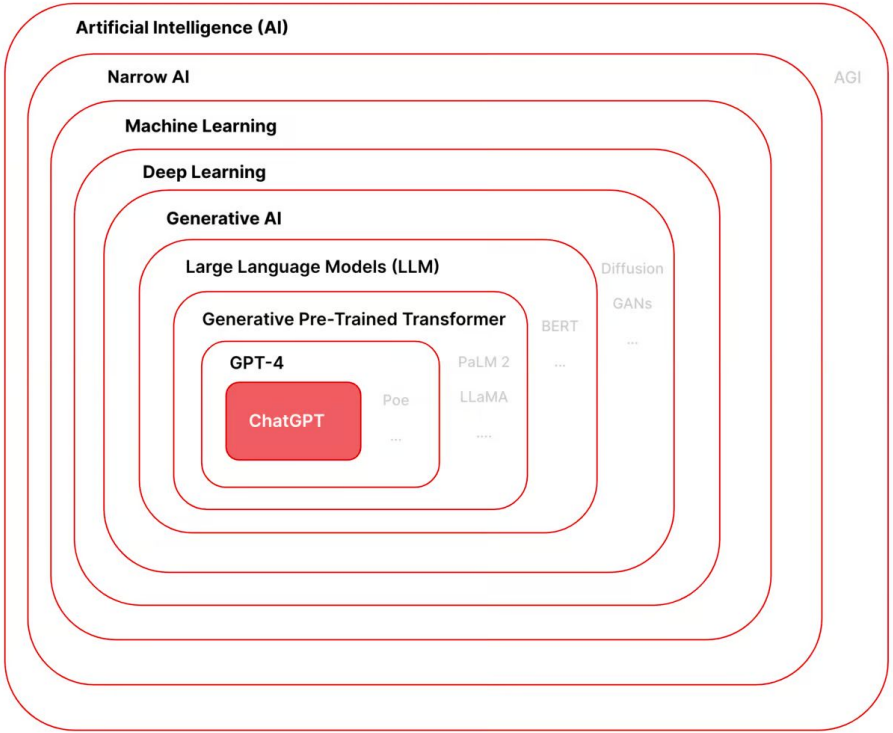
- Neural Networks with multiple layers (Deep Neural Networks).
- The rise of models like CNNs (Convolutional Neural Networks), RNNs (Recurrent Neural Networks), and Transformers.



https://www.gptechblog.com/5-diagrams-to-help-you-understand-generative-ai

Artificial Intelligence Taxonomy

WEEKLY AI USE CASES FOR ANALYTICS
AI FOR BI ROCKS



https://www.gptechblog.com/5-diagrams-to-help-you-understand-generative-ai


The Map Of Artificial Intelligence
Deep Learning
Artificial Neural Networks
Large Language Models
ChatGPT is here!
Convolutional Neural Networks (CNNs)
Recurrent Neural Networks (RNNs)
Generative Adversarial Networks (GANs)
Transformer Networks
Natural Language Processing (NLP)
Text Classification
Named Entity Recognition (NER)
Sentiment Analysis
Language Translation
Question-Answering Systems
Text Generation
Computer Vision
Image Recognition
Object Detection
Image Segmentation
Image Captioning
Facial Recognition
Machine Learning
Supervised Learning
Unsupervised Learning
Reinforcement Learning
Semi-Supervised Learning
Transfer Learning
Ensemble Learning
Robotics
Autonomous Robots
Robotic Process Automation (RPA)
Human-Robot Interaction (HRI)
Reinforcement Learning for Robotics
Expert Systems
Knowledge-Based Systems
Rule-Based Systems
Inference Engines
Swarm Intelligence
Ant Colony Optimization
Particle Swarm Optimization
Cognitive Computing
Emotion AI
Context-Aware AI
Knowledge Graphs
Graph-Based AI
Semantic Graphs
Game Playing AI
Chess AI
Go AI
Video Game AI
Recommender Systems
Collaborative Filtering
Content-Based Filtering
Hybrid Approaches
Autonomous Vehicles
Self-Driving Cars
Advanced Driver Assistance Systems (ADAS)
Computer Vision for Autonomous Vehicles
Virtual Agents and Chatbots
Conversational Agents
Dialogue Systems
Chatbot Development
Speech Recognition
Automatic Speech Recognition (ASR)
Speech-to-Text Conversion
Speech Synthesis (Text-to-Speech)
Knowledge Representation and Reasoning
Ontologies
First-Order Logic
Semantic Web
TechWorld WithMilan
simplifying complex topics

Machine learning process

Define the task.	Select the model.	Collect the data.	Train the model.	Evaluate the model.
• What do you want the model to do?	• What do you want the model to do?	• Clean the data • Split into validation set (if necessary)	• Choose model type • Set some parameters	• Choose evaluation metrics • Assess performance on validation set (if necessary)

Machine Learning Process by Mary Newhauser. Originally published with GPTech.

Classical AI vs Modern AI

Classical AI	Modern AI (ML/DL)
Rule-based systems	Data-driven learning
Logic and planning	Neural networks, statistics
No learning ability	Learns from examples

What is Machine Learning?

Machine Learning (ML):
Algorithms that learn from data and improve over time.

📘 Types of ML:

- **Supervised** – Labeled data (e.g., spam filter)
- **Unsupervised** – No labels (e.g., clustering)
- **Reinforcement** – Learning via reward signals (e.g., game AI)

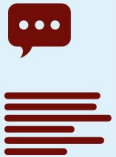
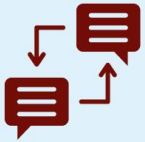
Deep Learning Overview

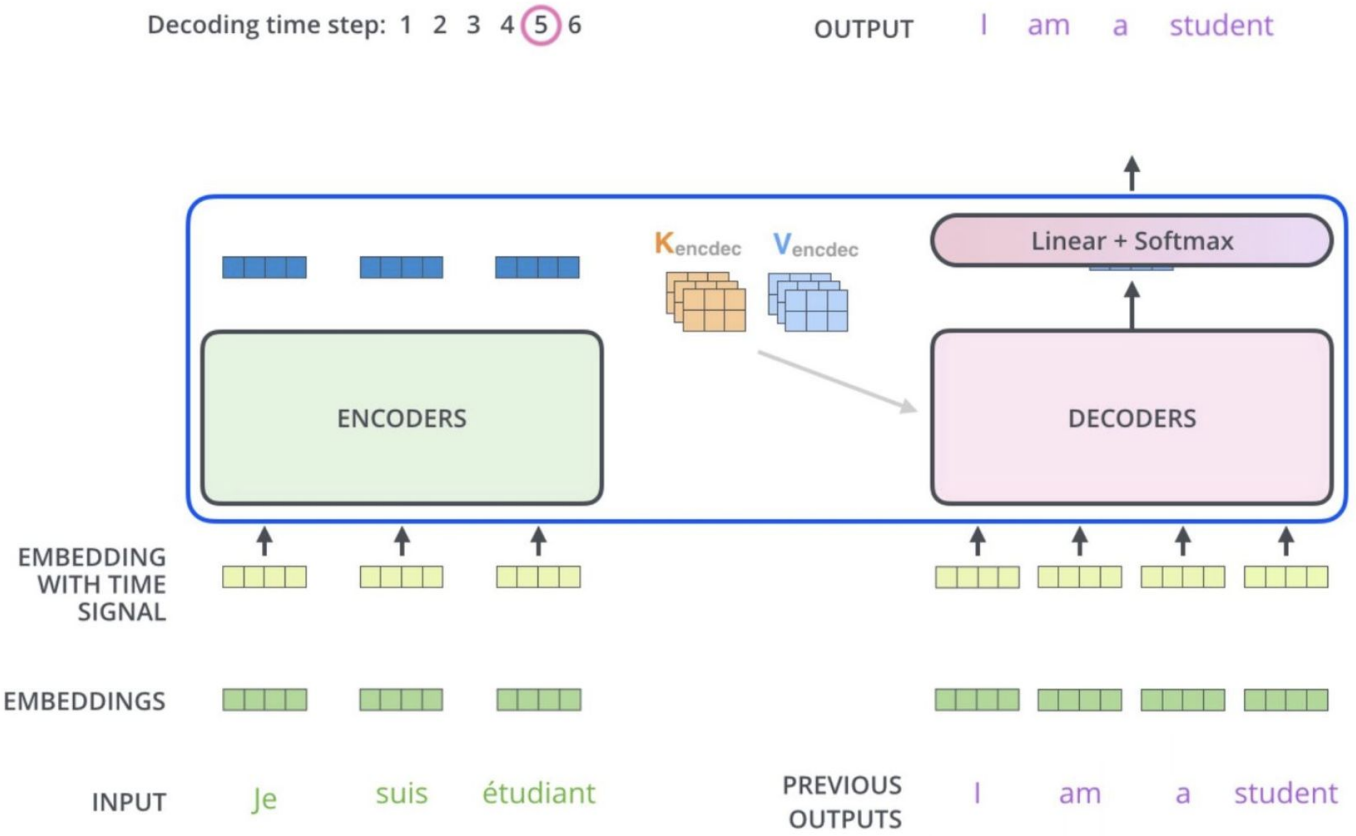
- Subfield of ML using **neural networks**
- Inspired by the brain
- Great for images, audio, language
- Powers modern LLMs and computer vision models

🧠 Examples: CNNs, RNNs, Transformers

GradientFlow.com

Getting The Most Out Of Large Language Models: Tune, Prompt, Reward

	What	How	When
Fine-tuning	Entails taking a pre-trained language model and further training it on a specific & smaller dataset that is specific to the task at hand. This is typically done by updating the weights of the model's last layer or layers while leaving the rest of the model static. 	During fine-tuning, a pre-trained model is loaded into memory and its weights are frozen. A smaller dataset relevant to the task at hand is loaded, and the pre-trained model is adjusted by tuning its weights. The model is typically trained for several epochs until the desired level of accuracy is reached. 	The fine-tuning process is normally used when the task or domain is well-defined, and there is sufficient labeled data available to train on. If you have a large dataset and a specific task in mind, fine-tuning a language model is likely to be the most effective approach. 
Prompt Engineering	Involves designing natural language prompts or instructions that can guide a language model to perform a specific task. 	Select & arrange the words in a prompt or query to elicit a specific response from the model. Top-notch prompt engineers conduct experiments, systematically record their findings, and refine their prompts to identify essential components. 	Best suited for tasks requiring a high level of precision and well-defined outputs. Prompt engineering can be used to craft a query that elicits a desired output. In some cases, prompt engineering can be used to improve the performance of a fine-tuned model by providing more guidance to the model during inference. 
RLHF Reinforcement Learning from Human Feedback	Reinforcement Learning from Human Feedback (RLHF) involves training a model by receiving feedback from human evaluators. 	The model is first trained on a dataset to establish a baseline level of performance. The model then generates a response to a prompt or query that is evaluated by a human. Feedback from the human evaluator is utilized to update the model's weights so that it can generate more accurate responses in the future. 	RLHF is ideally suited when the task requires a high level of accuracy and the model needs to be trained on a wide variety of inputs. RLHF is particularly useful when there is very limited data that can be used to train the model, since the model can be trained on a wide range of inputs through human feedback. 



Just one of many spectacular diagrams from The Illustrated Transformer by Jay Alammar on his blog.

State-of-the-Art AI Technologies

Large Language Models (LLMs):

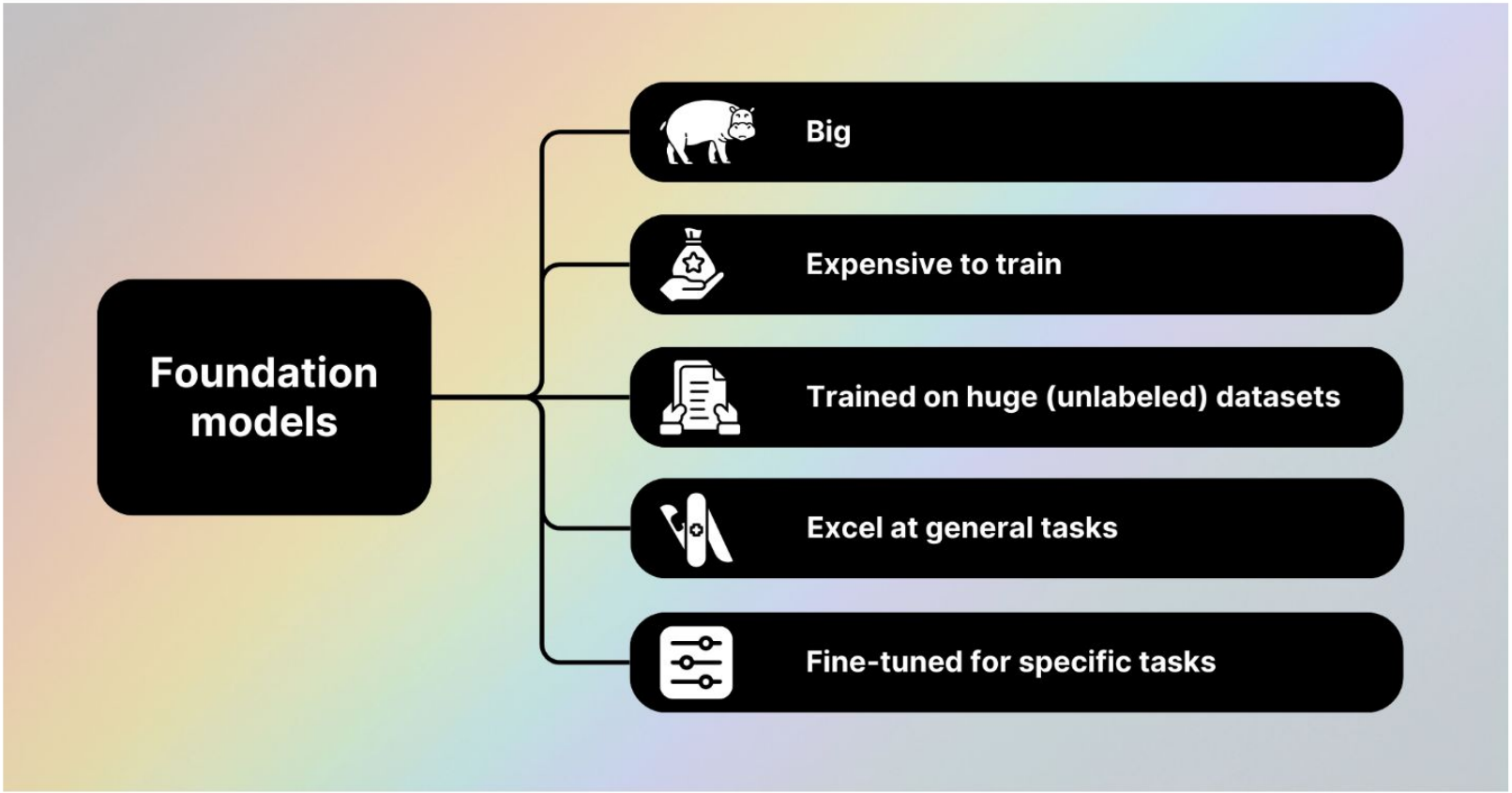
- Example: GPT-4, ChatGPT, BERT.
- Trained on massive text datasets, capable of understanding and generating human-like text.

Vision Models:

- Example: ResNet, YOLO, CLIP.
- Convolutional Neural Networks (CNNs) are revolutionizing image and video processing.

Transformers:

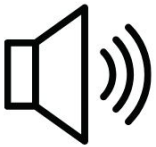
- Core architecture for both language models and vision models.
- Example: Vision Transformers (ViT), Transformers for NLP.



Common characteristics of foundation models in AI. Image by the author.

GPT-4o modalities

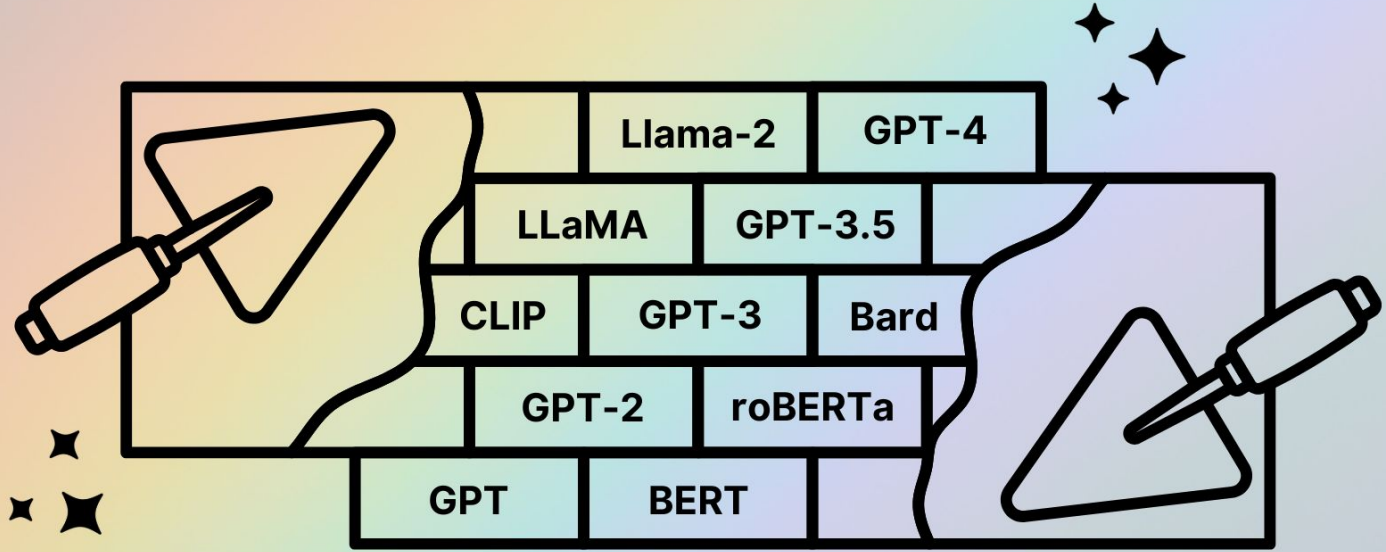
Text



Audio

Images

Video


Llama-2
GPT-4
LLaMA
GPT-3.5
CLIP
GPT-3
Bard
GPT-2
roBERTa
GPT
BERT

Key AI Concepts (Basic)

Machine Learning Basics:

- **Classification**: Predicting discrete labels (e.g., spam vs. non-spam email).
- **Regression**: Predicting continuous values (e.g., house price prediction).
- **Clustering**: Grouping similar data points (e.g., customer segmentation).

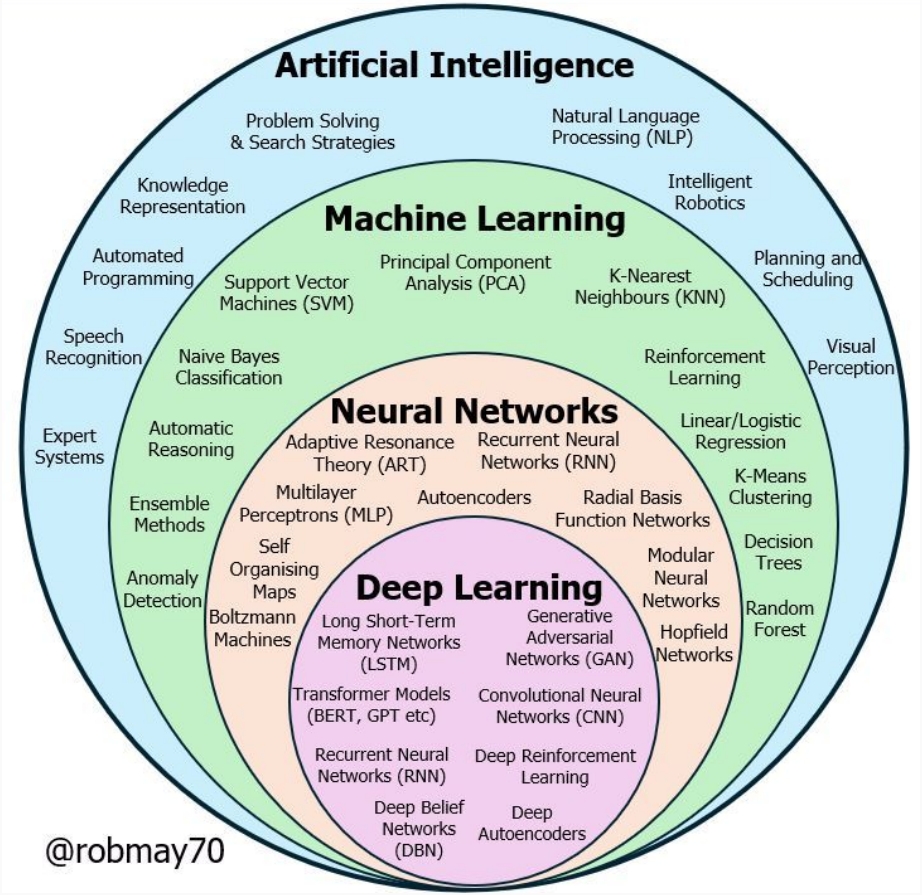
Reinforcement Learning (RL):

- Learning through trial and error (e.g., game playing, robotics).

Computer Vision:

- Identifying and interpreting images (e.g., object detection, image segmentation).

AI Core Components



AI Project Ideas (Applications)

Machine Learning:

- **Classification**: Email spam filtering, sentiment analysis, disease prediction.
- **Regression**: Stock price forecasting, weather prediction, sales forecasting.
- **Clustering**: Customer segmentation, anomaly detection, market basket analysis.

Reinforcement Learning:

- Autonomous agents (e.g., self-driving cars, game agents like AlphaGo).
- Robotics (e.g., robot arm learning to pick objects).

AI Project Ideas (Applications)

Computer Vision:

- Object detection (e.g., facial recognition, autonomous vehicles).
- Image segmentation (e.g., medical imaging, satellite image analysis).

LLMs:

- Text generation (e.g., creative writing, code generation).
- Text summarization and translation (e.g., news summarization, language translation).

🔍 Why Find Your Own Dataset?

- Practice real-world AI skills 🧠
- Learn to solve **undefined** problems 🔧
- Train yourself like a **Kaggle competitor** 👨‍💻👩‍💻
- Build your **portfolio** with unique projects 💼

🌐 Where to Find Datasets

Platform	Type	Link
🔹 Kaggle	Competitions & datasets	kaggle.com/datasets
🔹 Hugging Face	Text, audio, vision datasets	huggingface.co/datasets
🔹 Google Dataset Search	All types	datasetsearch.research.google.com
🔹 UCI ML Repo	Classic ML datasets	archive.ics.uci.edu/ml
🔹 Data.gov	Government open data (USA)	data.gov
🔹 Awesome-Public-Datasets	Curated GitHub list	tinyurl.com/awesome-data

Example Project Ideas

Dataset Example	Possible Task
Movie dataset	Predict movie success 🎬
Product reviews	Sentiment analysis 🛍️
Animal shelter adoption data	Predict if pet gets adopted 🐶
News headlines	Classify fake vs real 📰
Game leaderboard logs	Analyze top player traits 🎮
COVID-19 global cases	Forecast infection trends 🦠

Tools Every AI Student Should Know

🛠️ Common Libraries & Tools:

- **Python** – main programming language
- **NumPy, Pandas** – data handling
- **Scikit-learn** – classical ML
- **TensorFlow, PyTorch** – deep learning
- **Hugging Face** – LLMs & transformers
- **Google Colab** – free GPU notebook

Overview of Course Structure

- **Weeks 1-5**: Basics of AI, Machine Learning, and deep learning.
- **Weeks 6-10**: In-depth study of Transformers, LLMs, Computer Vision.
- **Weeks 11-14**: Practical applications in various domains.
- **Week 15**: Final Project and Course Wrap-Up.

What You'll Learn

Foundations: Understand the core concepts of AI and Machine Learning.

Tools & Techniques: Learn about key algorithms, models, and tools in modern AI.

Practical Experience: Build AI systems through hands-on projects (classification, regression, reinforcement learning, etc.).

Assignments & Projects

- **Project 1 (ML-based)**: Build a model for image classification using CNNs.
- **Project 2 (LLM-based)**: Train a text summarizer or chatbot using GPT-like models.
- **Project 3 (RL-based)**: Create an agent that can learn a task via reinforcement learning (e.g., game playing).

Modern AI Concepts: Quick Intro for Beginners

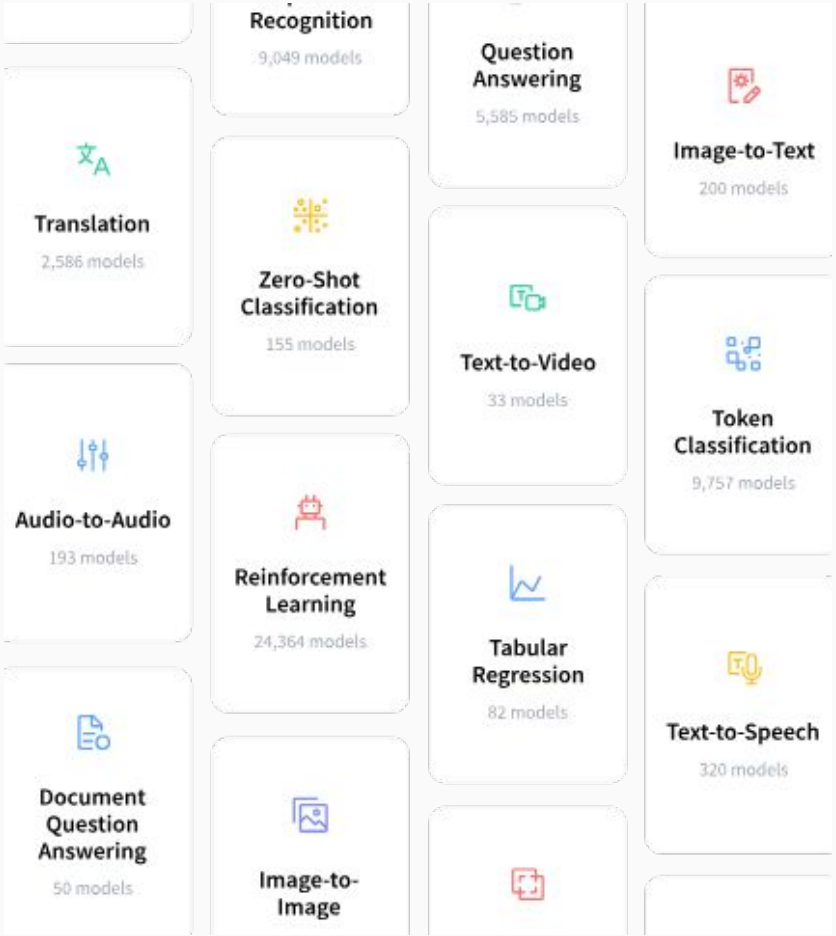
What is Hugging Face?

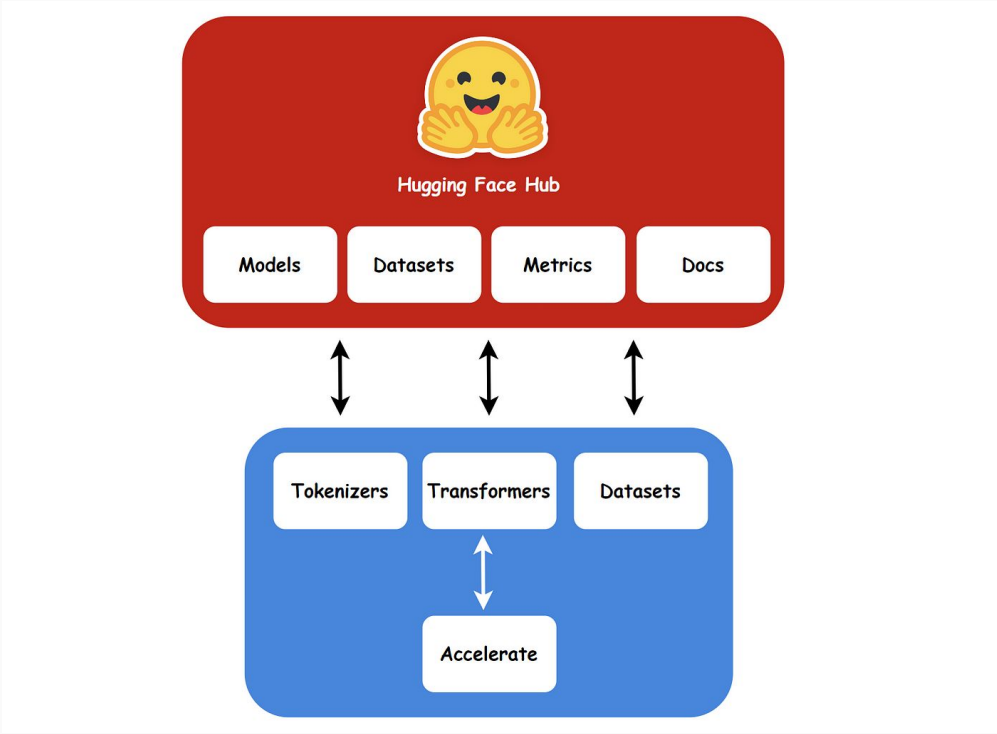
Hugging Face is a company and open-source platform that provides pretrained AI models for Natural Language Processing (NLP) tasks like sentiment analysis, translation, summarization, and more.

You can easily use these models with just a few lines of code using the transformers library and pipeline().

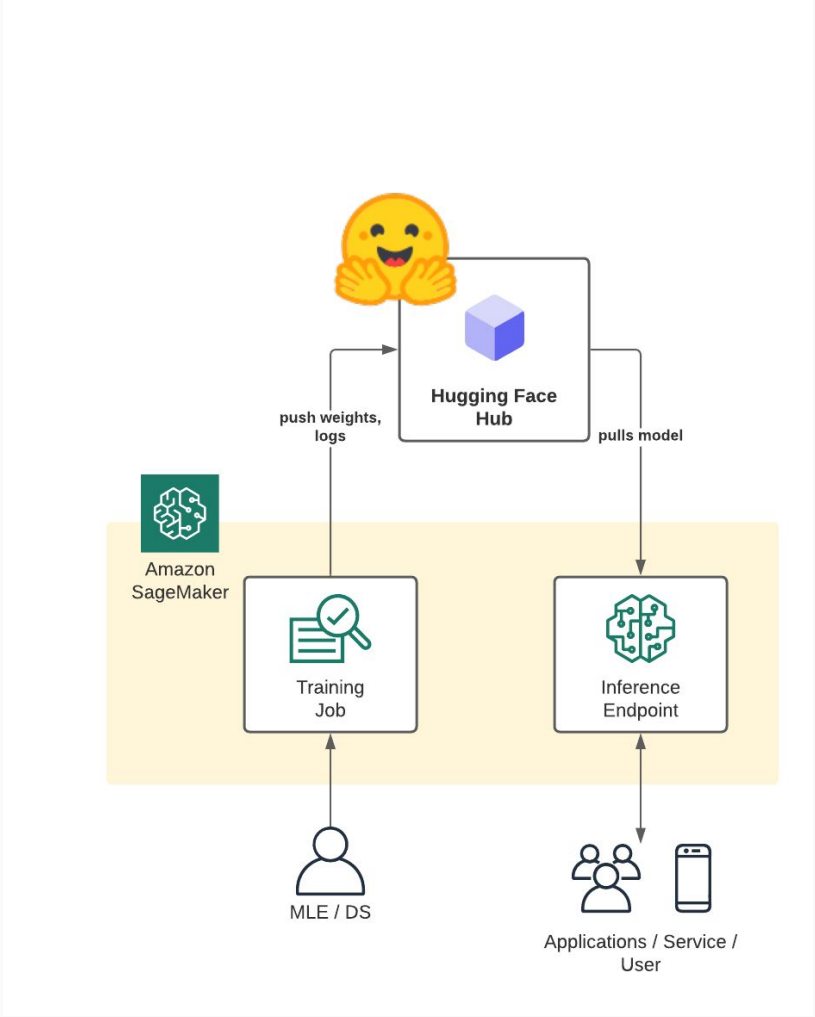
Hugging Face

https://huggingface.co/





https://www.youtube.com/watch?v=jan07gloaRg



https://www.philschmid.de/huggingface-hub-amazon-sagemaker

🧰 What is a Pipeline?

A pipeline in Hugging Face is a simple interface that loads a pretrained model and tokenizer to do specific tasks.

Examples of pipelines:

- pipeline("sentiment-analysis")
- pipeline("ner")
- pipeline("text-generation")
- pipeline("translation")

❤️ Sentiment Analysis

This helps us find out if a piece of text expresses positive, negative, or neutral emotions.

```
from transformers import pipeline

sentiment_analyzer = pipeline("sentiment-analysis")

sentiment_analyzer("I love Khon Kaen!")

# Output: [{'label': 'POSITIVE', 'score': 0.99}]
```

Named Entity Recognition (NER)

NER finds and classifies important things (called *entities*) in text like:

- 🧑 Names of people
- 🌍 Places
- 🏢 Organizations

```
ner = pipeline("ner", aggregation_strategy="simple")

ner("Barack Obama was the President of the United States.")

# Output: [{'entity_group': 'PER', 'word': 'Barack Obama'}, ...]
```

🧠 What is a Pretrained Model?

A **pretrained model** has already been trained on a large dataset (like Wikipedia or news articles), so we can use it without training from scratch. Hugging Face gives access to hundreds of these.

🧾 Common Models:

- distilbert-base-uncased (for sentiment)
- bert-base-cased (for NER)
- gpt2 (for text generation)

Resources

- **Textbooks**:
    - “Deep Learning” by Ian Goodfellow.
    - “Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow” by Aurélien Géron.
- **Tools**:
    - Python (NumPy, Pandas, Matplotlib) on Colab.
    - TensorFlow, PyTorch, OpenCV, HuggingFace.

Next Steps and Questions

Pre-Class Assignment: Read the introduction chapter from the textbook and explore AI news/articles.

Questions?: Any concerns or topics you'd like us to dive deeper into?

Reference and Credit

- https://www.gptechblog.com/5-diagrams-to-help-you-understand-generative-ai/
- https://medium.com/@lmpo/a-brief-history-of-ai-with-deep-learning-26f7948bc87b
- https://introtodeeplearning.com/
- https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks
- https://cs231n.stanford.edu/
- https://openai.com/index/chatgpt/
- https://gemini.google.com/
- https://www.philschmid.de/huggingface-hub-amazon-sagemaker
- https://lilianweng.github.io/posts/2021-07-11-diffusion-models/