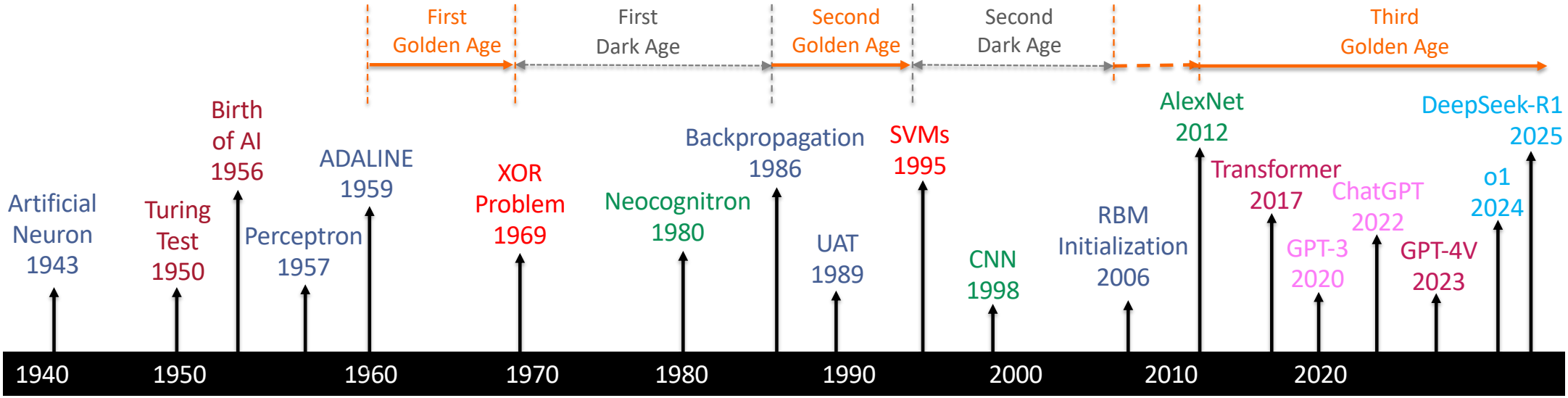
A Brief History of AI with Deep Learning

AI with Deep Learning
EE4016

Prof. PO Lai-Man
Department of Electrical Engineering
City University of Hong Kong

A Brief History of AI with Deep Learning



McCulloch-Pitts

Rosenblatt Widrow-Hoff

Minsky-Papert

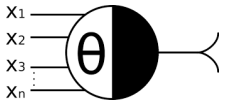
Rumelhart, Hinton et al.

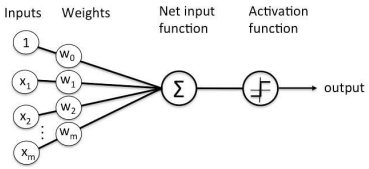
LeCun

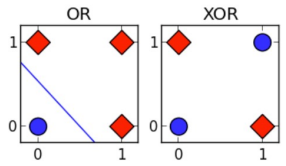
Hinton-Ruslan

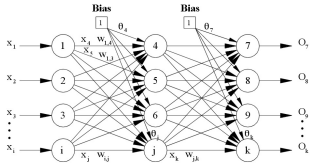
Krizhevsky et al.

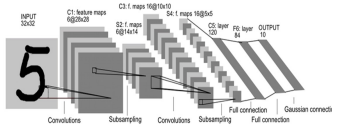
Vaswani

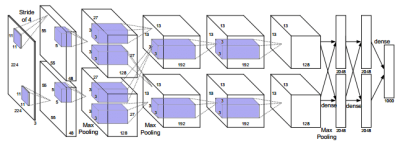












https://medium.com/@edmond.po/a-brief-history-of-ai-with-deep-learning-26f7948bc87b

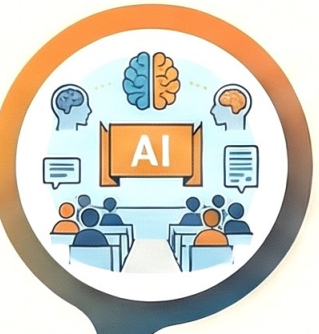
The Ages of AI: A Timeline of Breakthroughs

The history of Artificial Intelligence is not a straight line but a series of "Golden Ages" of innovation and "Dark Ages" of stagnation. Key theoretical breakthroughs and architectural shifts, from the first artificial neuron to the Transformer, have driven progress, culminating in today's advanced reasoning and autonomous AI agents.

The Foundational Era (1940s - 2000s)

1943
The First Artificial Neuron

McCulloch and Pitts proposed a mathematical model of a neuron, laying the groundwork for neural networks.

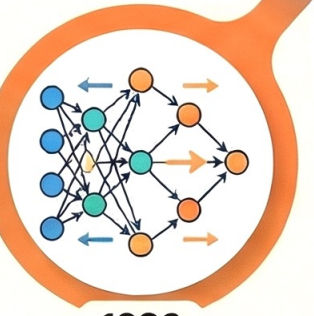


1956
The Birth of "Artificial Intelligence"

The term was coined at the Dartmouth Conference, officially establishing AI as a research field.

1969
The "First AI Winter" Begins

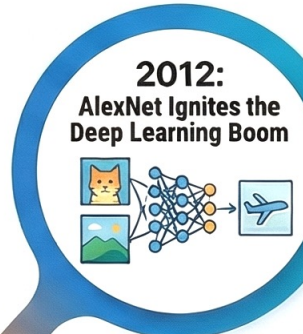
The XOR Problem revealed the limits of single layer perceptrons, leading to reduced funding and interest.



1986
The Backprepagation Breakthrough

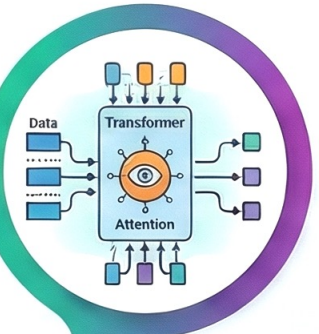
The popularization of this algorithm enabled the training of multi-layer networks, sparking a "Second Golden Age."

The Modern Revolution (2012 - Present)



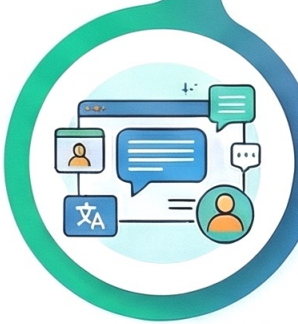
2012:
AlexNet Ignites the Deep Learning Boom

2012
A Convolutional Neural Network (CNN) won the ImageNet challenge, starting the "Third Golden Age."



2017
"Attention Is All You Need"

The Transformer architecture was introduced, replacing recurrence with attention and enabling modern LLMs.



2022
ChatGPT Goes Mainstream

Built on GPT-S.5 and RLHF, its release made advanced AI accessible to millions of users.

2025
The Rise of Agentic AI

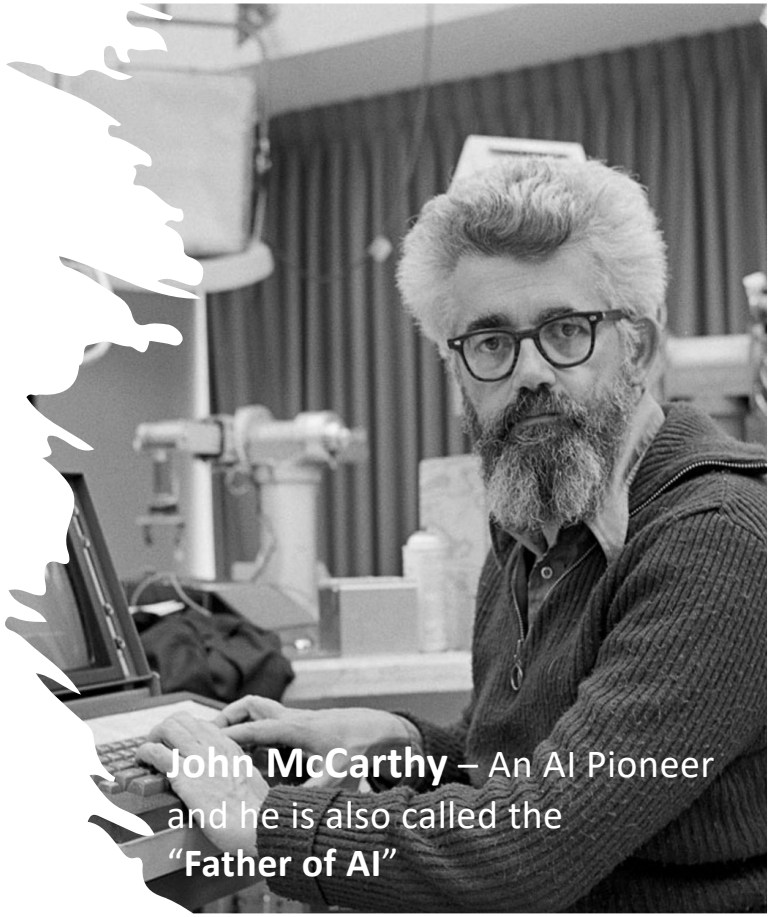
AI shifts from passive tools to autonomous agents that can plan and execute complex tasks.

Content

https://medium.com/@edmond.po/a-brief-history-of-ai-with-deep-learning-26f7948bc87b

The Birth of Artificial Intelligence (1956)

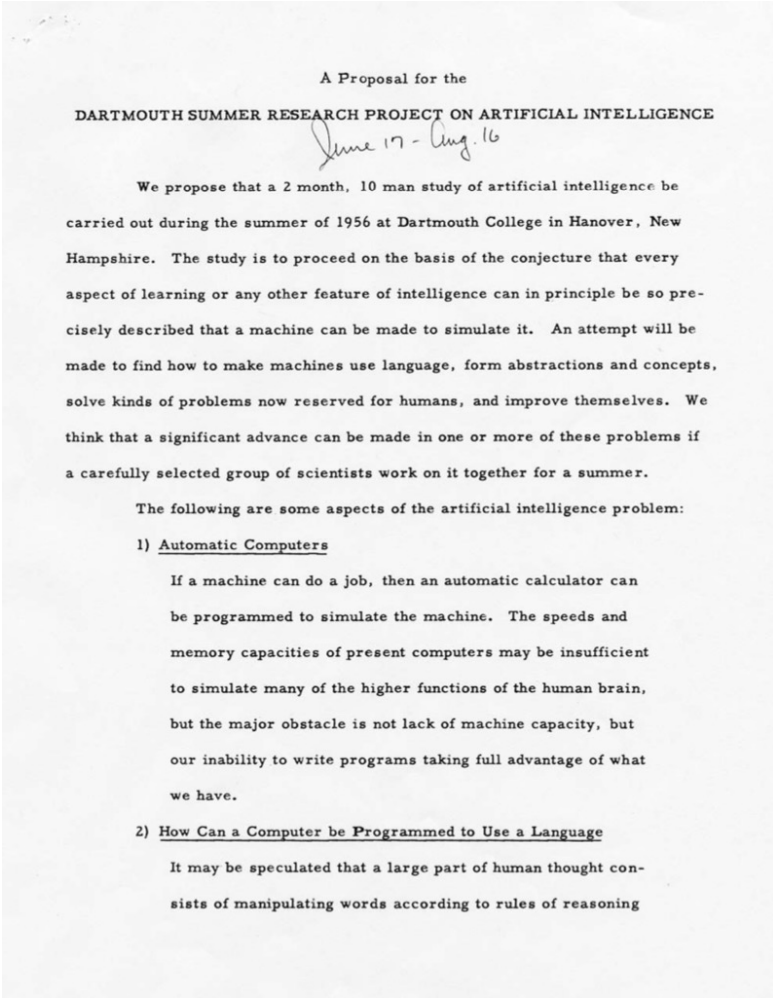
- The concept of Artificial Intelligence (AI) has been around for centuries, but the modern field of AI began to take shape in the mid-20th century.
- The term "Artificial Intelligence" was first coined in 1956 by John McCarthy at the **Dartmouth Summer Research Project on Artificial Intelligence**.
- This conference, which brought together notable figures such as Marvin Minsky, Nathaniel Rochester, and Claude Shannon, is often considered the birthplace of AI as a field of research.



John McCarthy – An AI Pioneer and he is also called the "**Father of AI**"

Dartmouth Summer Research Project on Artificial Intelligence

Marvin Minsky, Claude Shannon, Ray Solomonoff and other scientists at the Dartmouth Summer Research Project on Artificial Intelligence (Photo: Margaret Minsky)
https://www.cantorsparadise.com/the-birthplace-of-ai-9ab7d4e5fb00



A Proposal for the

DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE

June 17 - Aug. 16

We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.

The following are some aspects of the artificial intelligence problem:

1) Automatic Computers

If a machine can do a job, then an automatic calculator can be programmed to simulate the machine. The speeds and memory capacities of present computers may be insufficient to simulate many of the higher functions of the human brain, but the major obstacle is not lack of machine capacity, but our inability to write programs taking full advantage of what we have.

2) How Can a Computer be Programmed to Use a Language

It may be speculated that a large part of human thought consists of manipulating words according to rules of reasoning

A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence (1955)

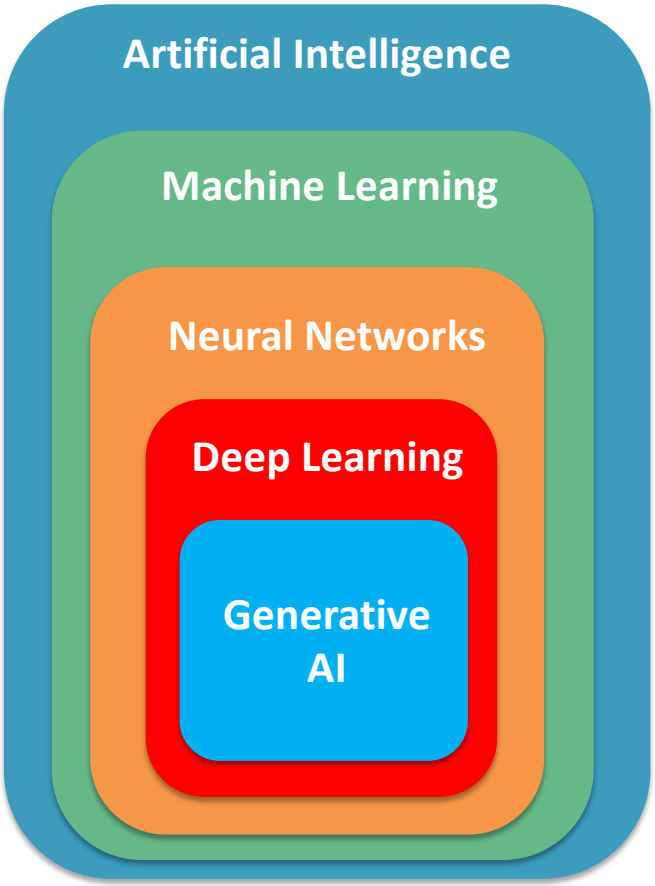
The Evolution of AI: From Rule-Based Systems to DL

- The development of AI has progressed through several eras.
- In the 1950s, AI focused on creating algorithms for tasks like playing chess and problem-solving.
- **The 1960s and 1970s introduced rule-based expert systems**, such as MYCIN, which could diagnose bacterial infections.
- **The 1980s saw the emergence of machine learning**, which enabled AI systems to learn from data and improve over time.
- This led to the development of backpropagation algorithms and neural networks, laying the foundation for modern deep learning (DL) techniques.
- Today, **deep learning**, a specialized branch of machine learning, **is the driving force behind cutting-edge AI technologies**, transforming the field and enabling AI systems to extract complex features from raw input data.

AI, ML, NN, DL, Generative AI

Generative AI

- Generative AI creates new content, such as text, images, or videos, by learning from large datasets. It has many applications.
- Examples:
  - **Large Language Models** (LLMs): ChatGPT, GPT-5, Gemini 2.5 Pro, Claude 4 Sonnet, DeepSeek-R1
  - **Text-to-Image Generation**: Midjourney, Stable Diffusion Model, FLUX.1, Nano Banana
  - **Text-to-Video Generation**: Veo3, Kling, MiniMax



Early Artificial Neural Networks (1940s — 1960s)

Main Topics of the Course

- **Course Overview**
- **History of AI with Deep Learning**
- **Review of Linear Algebra, Calculus and Probability**
- **Multi-Layer Perceptron (MLP)**
  - Perceptron and MLP
  - Activation function, Loss function
  - Backpropagation and Gradient Descent
  - Regularizations and Optimizations
- **Convolutional Neural Networks (CNNs)**
  - Convolutional Neural Layer
  - LeNet, AlexNet, VGGNet, InceptionNet, ResNet, DenseNet, EfficientNet
  - Computer Vision Applications:
    - Object Detection, Image Segmentation
- **Recurrent Neural Network (RNN)**
  - Introduction to NLP
  - Tokenization and Word Embeddings
  - Vanilla RNN, LSTM, GRU
  - Encoder-Decoder Models and Attention
- **Transformers and Large Language Models (LLMs)**
  - Self-Attention Mechanism
  - Transformers
  - BERT, GPT, GPT-2, GPT-3, T5 and BART
  - LLM Decoding and Prompt Engineering
  - Model Quantization
  - Parameter-Efficient Fine-Tuning (PEFT)
- **Preference Alignment and Multimodality**
  - Preference Alignment: Instruction-Tuning, RLHF, DPO, ORPO
  - Multimodality: CLIP, GPT-4V, LLaVA

McCulloch & Pitts Neuron Model (1943)

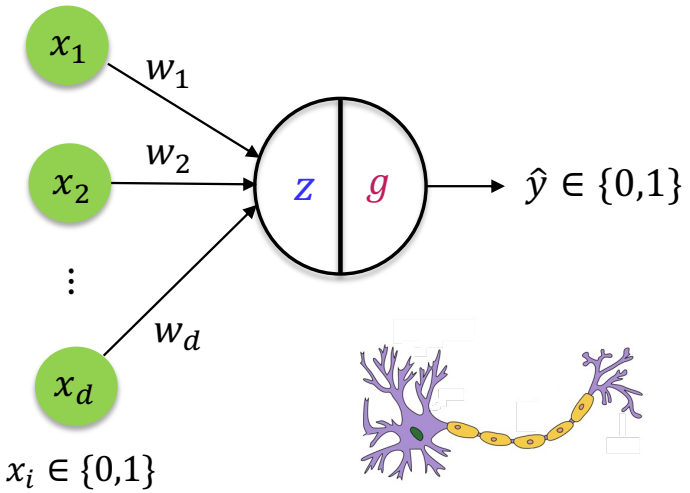
- **McCulloch** (neuroscientist) and **Pitts** (logician) proposed a highly simplified computational model of the neuron which known as MP Neuron
- MP Neuron aggregates inputs with function $z(\cdot)$ and makes a decision with activation function $g(\cdot)$. Both inputs and output are binary (0 or 1).

$$z = \sum_{j=1}^{d} w_j \cdot x_j$$

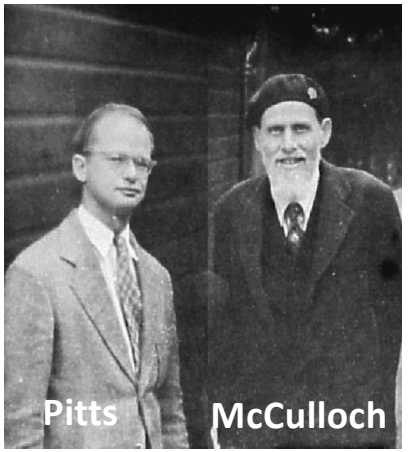
$$\hat{y} = g(z) = \begin{cases} 1 & \text{if } z(x) \geq T \\ 0 & \text{if } z(x) < T \end{cases}$$

Threshold Activation function

An Artificial Neuron



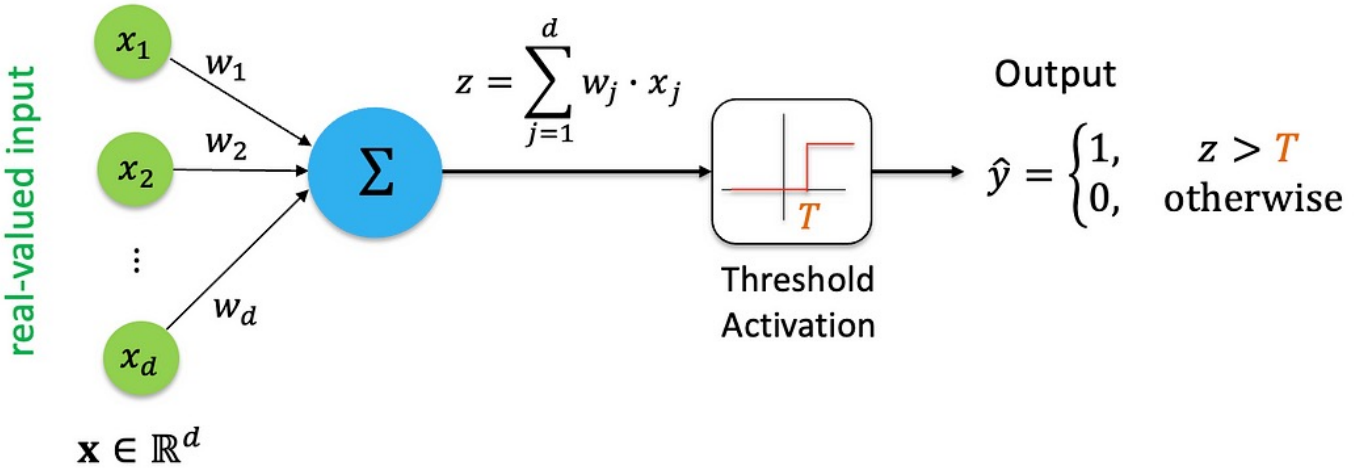
A Biologic Neuron



McCulloch and Pitts: A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, 1943

Rosenblatt's Perceptron Model (1957)

- **Frank Rosenblatt** introduced the **Perceptron** in 1957, a single-layer neural network that can learn and recognize patterns.
- **It processes real-valued inputs** and adjusts weights to minimize classification errors.



Frank Rosenblatt

Rosenblatt: The perceptron - a probabilistic model for information storage and organization in the brain. Psychological Review, 1958.

1957 News about the Rosenblatt's Perceptron

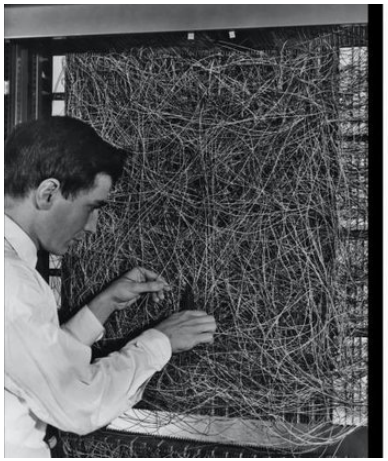
- In the 1950s, Rosenblatt predicted to the New York Times that Perceptrons would be capable of:
  - Recognizing individuals and addressing them by name
  - Translating speech from one language to another, either verbally or in written form
- These ambitious claims, reminiscent of 2022's AI breakthrough of ChatGPT, generated significant excitement and anticipation for the potential of artificial intelligence.

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

Rosenblatt's Perceptron Learning

- Rosenblatt also developed **a supervised learning algorithm** for the Perceptron, allowing it to learn from training data.

$$\mathcal{L}(\mathbf{w}) = -\sum_{i \in \mathcal{M}} \mathbf{w}^T \mathbf{x}^{(i)} y^{(i)}$$



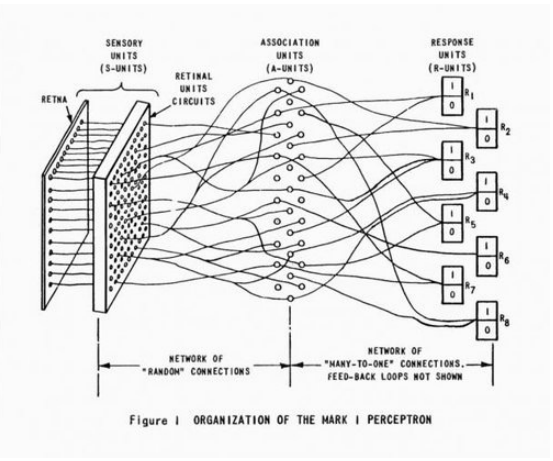
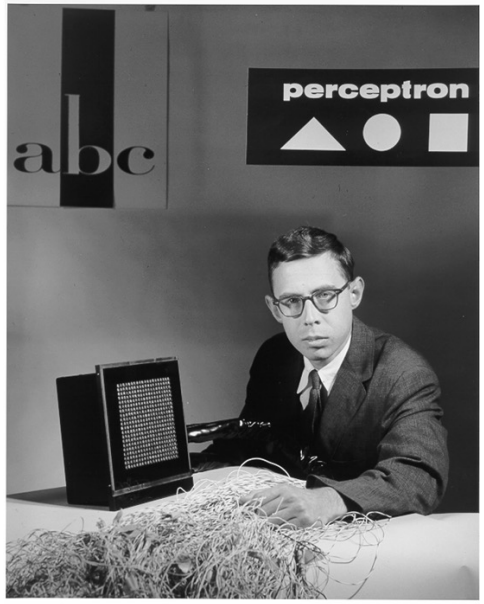


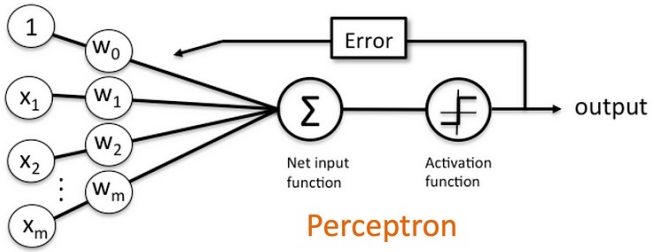
Figure 1 ORGANIZATION OF THE MARK 1 PERCEPTRON



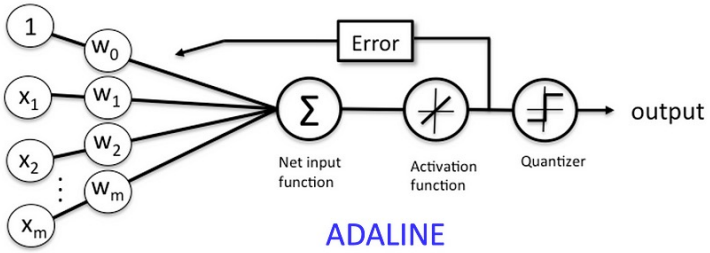
- The Perceptron's potential to recognize individuals and translate speech sparked public interest in AI. However, **it had a critical limitation: it couldn't learn from non-linearly separable data.**

ADALINE (1959)

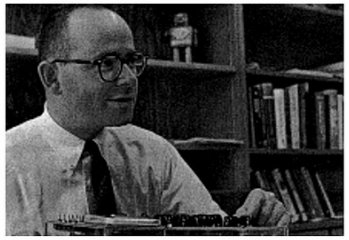
- In 1959, **Widrow and Hoff** introduced **ADALINE** (Adaptive Linear Neuron) aka Delta Learning Rule, an improvement over the Perceptron learning rule.
- ADALINE addressed limitations like binary output and noise sensitivity and could learn and converge on non-linearly separable data, a major breakthrough in neural network development.



Perceptron



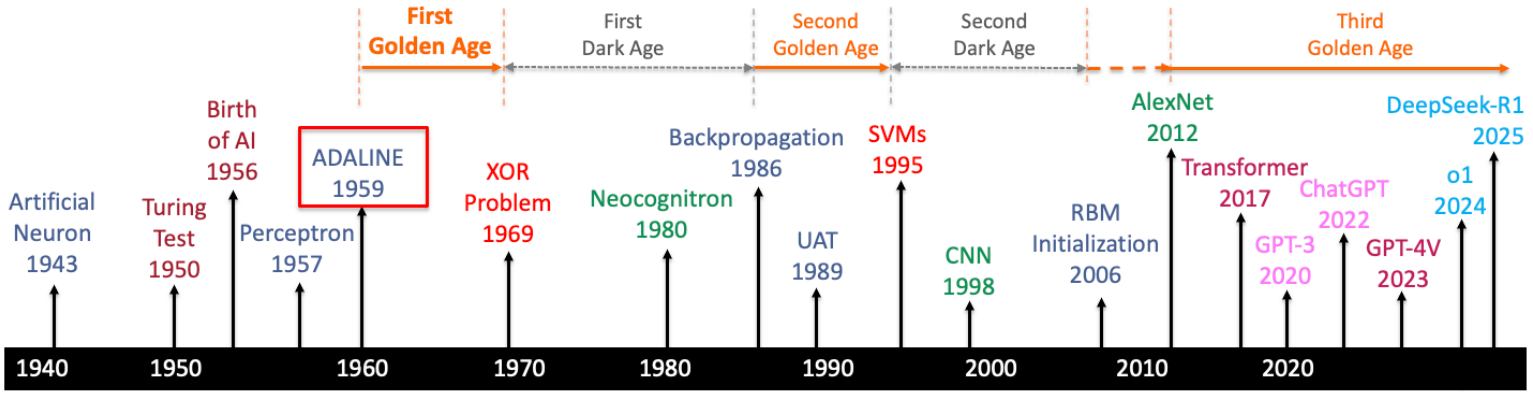
ADALINE


Bernard Widrow


Marcian Hoff

The Birth of the "First Golden Age" of Neural Networks

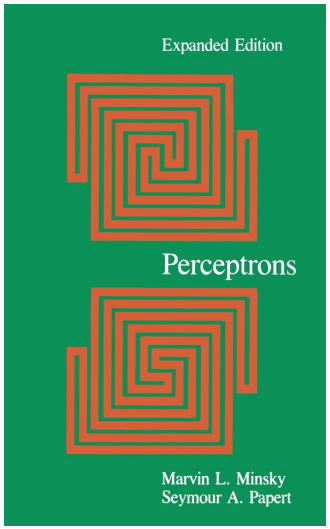
- **ADALINE**'s key features include:
  - Linear activation function, suitable for regression tasks and continuous outputs
  - Least Mean Squares (LMS) algorithm, minimizing mean squared error and providing efficient learning
  - Adaptive weights, adjusting based on error and enabling effective learning in noisy environments
- **The introduction of ADALINE marked the start of the First Golden Age of Neural Networks**, overcoming limitations of Rosenblatt's Perceptron Learning. This breakthrough enabled efficient learning, continuous outputs, and adaptation to noisy data, sparking a wave of innovation and rapid progress in the field.

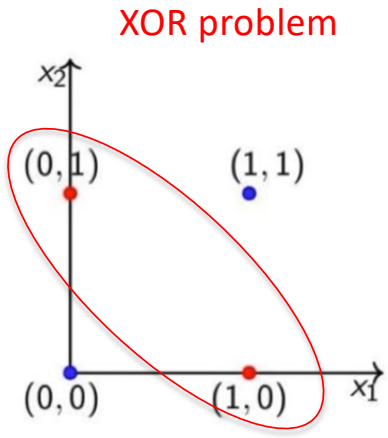


XOR Problem (1969)

- In 1969, **Minsky and Papert**'s book "**Perceptrons**" exposed a critical limitation of the single-layer Perceptron.
- They showed that the **Perceptron couldn't solve the XOR problem**, a simple binary classification task, due to its linear decision boundary.
- **The XOR problem is non-linearly separable**, meaning no single linear boundary can correctly classify all input patterns.

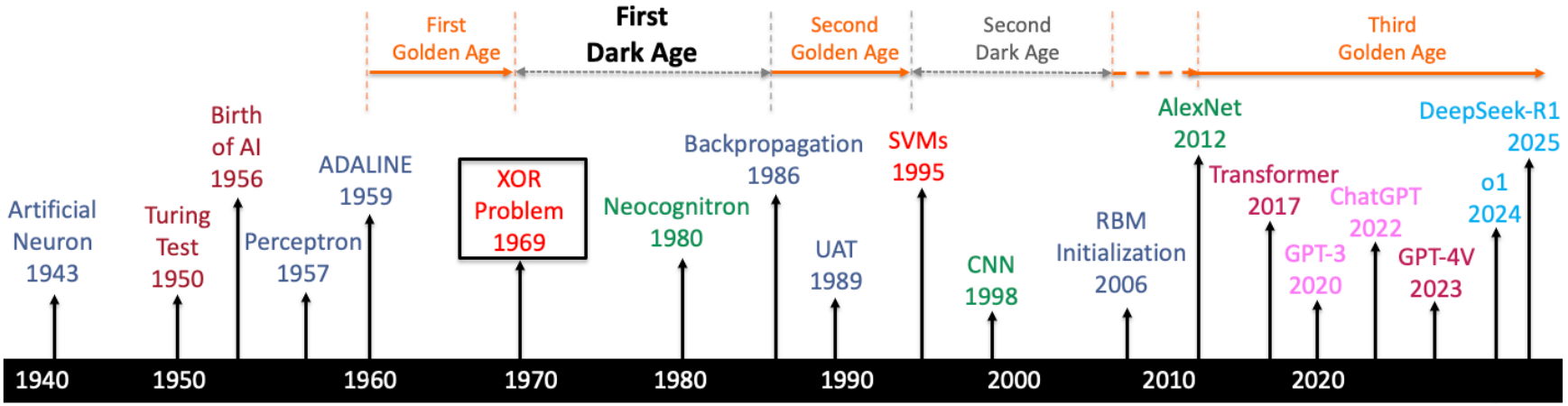
Minsky and **Papert**: Perceptrons: An introduction to computational geometry. MIT Press, 1969.





First Dark Age of Neural Networks

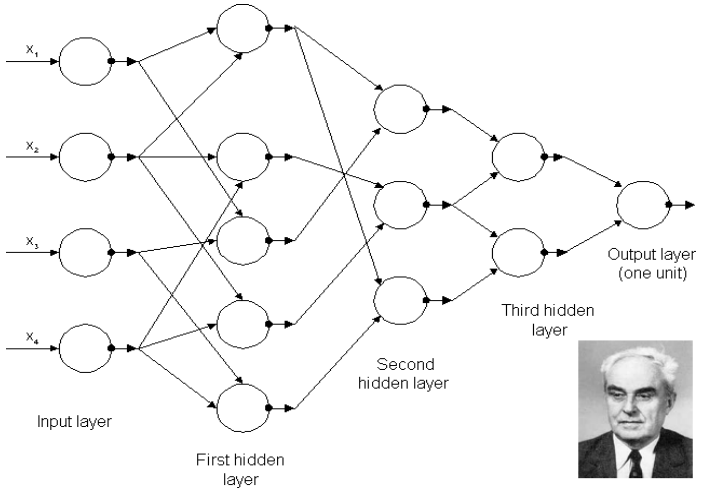
- The revelation about the Perceptron's limitations led to a loss of confidence in neural networks and a shift towards symbolic AI methods, marking the "**First Dark Age of Neural Network**" from the 1970s to the 1980s.



- However, the insights gained from the XOR problem paved the way for future advancements, ultimately leading to the creation of more complex models like the Multilayer Perceptron and the resurgence of neural networks in later decades.

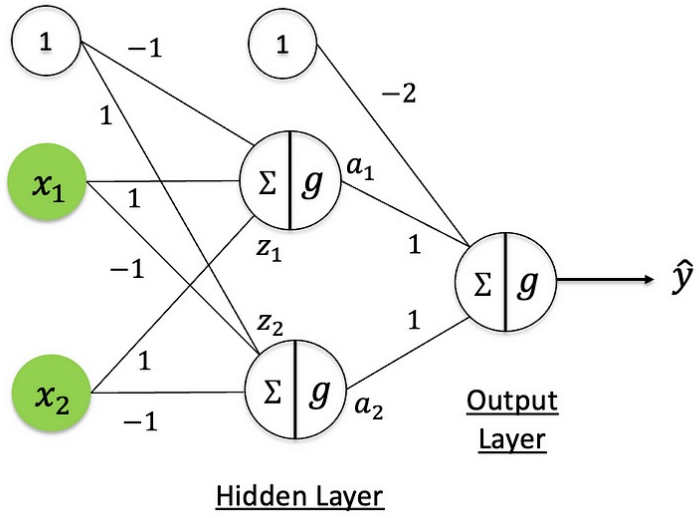
The Multilayer Perceptron (1960s)

- Building upon the foundational work of the Perceptron, researchers in the 1960s introduced the **Multilayer Perceptron (MLP)**, a neural network model with multiple layers of interconnected neurons.
- The MLP addressed the limitations of the single-layer Perceptron by incorporating one or more hidden layers between the input and output layers.
- **A. G. Ivakhnenko and V. Lapa**, Soviet scientists, made significant contributions to the development of the MLP, building upon the foundational work of the Perceptron and introducing a multi-layer neural network with interconnected layers of neurons.



Hidden Layers

- The addition of hidden layers allows the MLP to capture and represent complex, non-linear relationships in the data.
- These hidden layers significantly enhance the network's learning capabilities, enabling it to solve problems that are not linearly separable, such as the XOR problem.



XOR

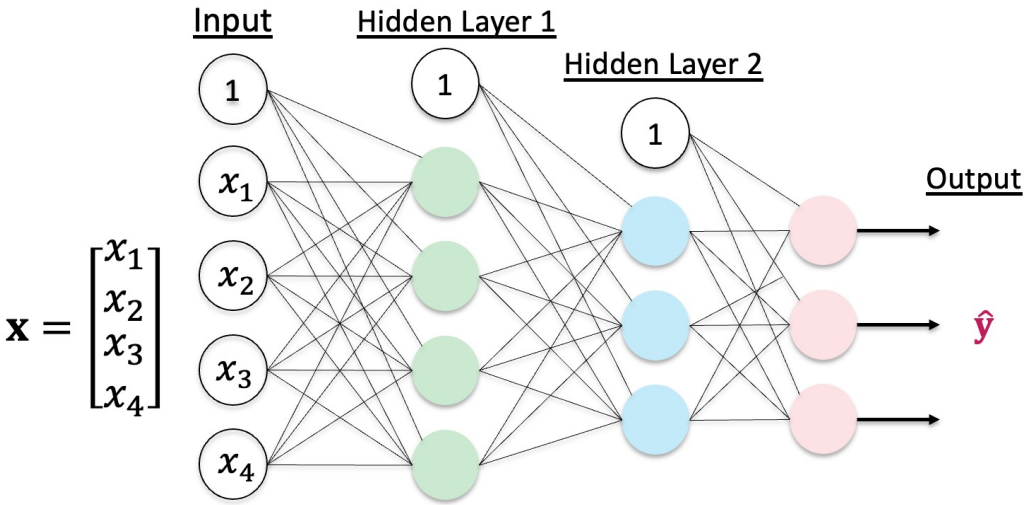
(x_1, x_2)	(z_1, z_2)	(a_1, a_2)	$\hat{y}$
$(0, 0)$	$(-1, 1)$	$(0, 1)$	0
$(0, 1)$	$(0, 0)$	$(1, 1)$	1
$(1, 0)$	$(0, 0)$	$(1, 1)$	1
$(1, 1)$	$(1, -1)$	$(1, 0)$	0

$$a = g(z) = u(z) = \begin{cases} 1, & z \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

Activation function is the unit step function $u(z)$

Feedforward Neural Networks

- Today the term ”**Feedforward Network**" is more commonly used in deep learning, referring to **Multi-Layer Perceptron (MLP)**.
  - A feedforward Neural Network **is just a function**



Hyperparameters of the Network :

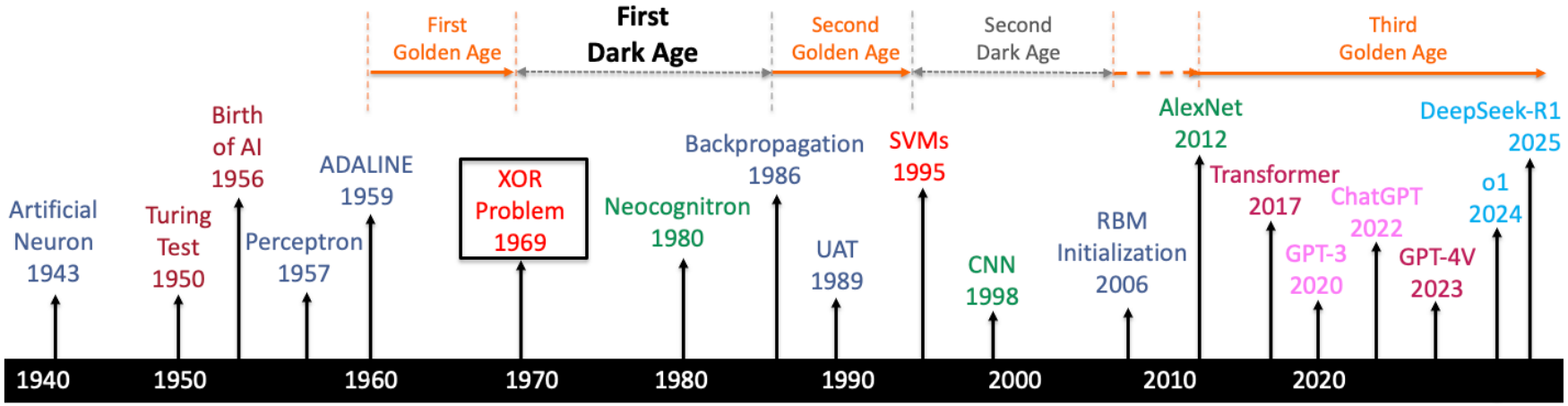
- No. of Layers: 3 ($L = 3$)
- Input Layer: 4 neurons ($d = 4$)
- Layer 1: 4 neurons ($n_1 = 4$)
- Layer 2: 3 neurons ($n_2 = 3$)
- Output Layer: 3 neuron ($K = n_3 = 3$)
- Activation function: Sigmoid σ

$$\hat{\mathbf{y}} = f_\theta(\mathbf{x}) = \sigma\left(\mathbf{W}^{(3)}\sigma\left(\mathbf{W}^{(2)}\sigma\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}\right) + \mathbf{b}^{(2)}\right) + \mathbf{b}^{(3)}\right)$$

Network Parameters are Weights and Biases : $\theta = \left\{\left(\mathbf{W}^{(1)}, \mathbf{b}^{(1)}\right), \left(\mathbf{W}^{(2)}, \mathbf{b}^{(2)}\right), \left(\mathbf{W}^{(3)}, \mathbf{b}^{(3)}\right)\right\}$

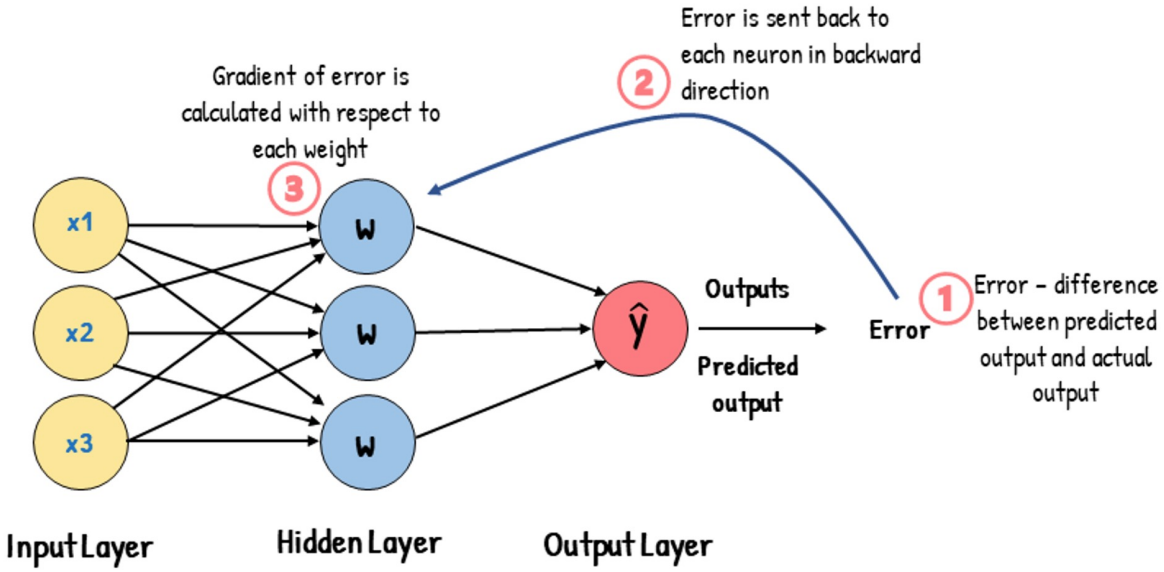
Historical Context and Challenges of MLPs

- The MLP marked a significant advancement in neural network research, but its development in the 1960s and 1970s was hindered by challenges:
  - **Lack of efficient training algorithms**, including backpropagation
  - **Computational limitations**, making it difficult to train deep networks
- The first Dark Age of Neural Networks ended in 1986 with the rediscovery and publication of **the backpropagation algorithm, starting the Second Golden Age of Neural Networks**.



Backpropagation (1970s-1980s)

- **Backpropagation** is a fundamental component of deep neural network training, allowing for efficient gradient calculation in complex networks.
- Since its inception, the algorithm has undergone significant refinement, transformed the field of neural network training and driven innovation in various areas.



Backpropagation Development in 1970s and 1980s

- **Early Developments (1970s)**
  - Seppo Linnainmaa (1970): Introduced automatic differentiation, a key component of backpropagation.
  - Paul Werbos (1974): Proposed using the chain rule to compute the gradient of the error function, enabling multilayer neural network
- **Refinement and Popularization (1986)**
  - David Rumelhart, Geoffrey Hinton, and Ronald Williams: Presented backpropagation as a practical and efficient method for training deep neural networks.

David Rumelhart

Geoffrey Hinton

Ronald Williams

Key Features of Backpropagation

- **Gradient Descent**: Used to minimize error function by updating weights iteratively.
- **Chain Rule**: Decomposes error gradient into partial derivatives, computed through backward pass.

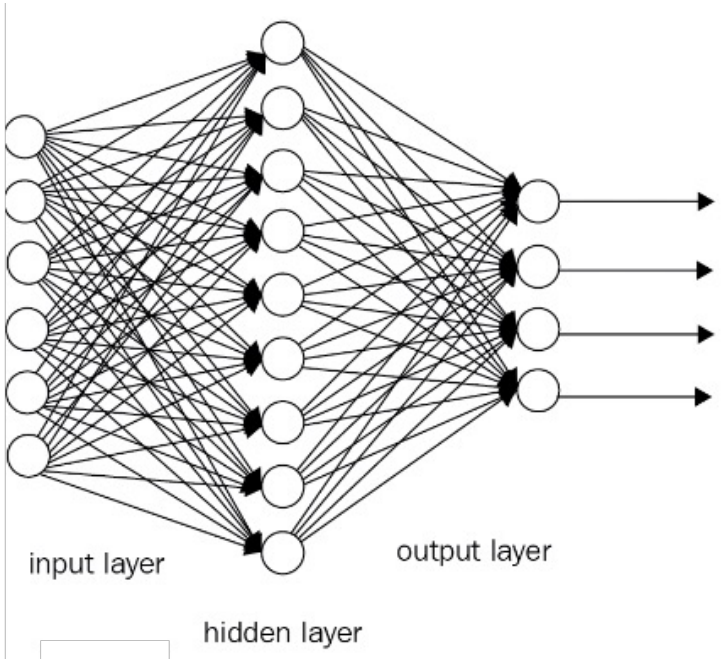
$$\frac{\partial \mathcal{L}}{\partial w_{i,j}^{(l)}} = \frac{\partial \mathcal{L}}{\partial z_i^{(l)}} \cdot \frac{\partial z_i^{(l)}}{\partial w_{i,j}^{(l)}} = \delta_i^{(l)} \cdot a_j^{(l-1)} \quad \text{and} \quad \frac{\partial \mathcal{L}}{\partial b_i^{(l)}} = \frac{\partial \mathcal{L}}{\partial z_i^{(l)}} \cdot \frac{\partial z_i^{(l)}}{\partial b_i^{(l)}} = \delta_i^{(l)}$$

- **Layered Computation**: Operates layer-by-layer, starting from output layer, to propagate gradients correctly.

Universal Approximation Theorem (1989)

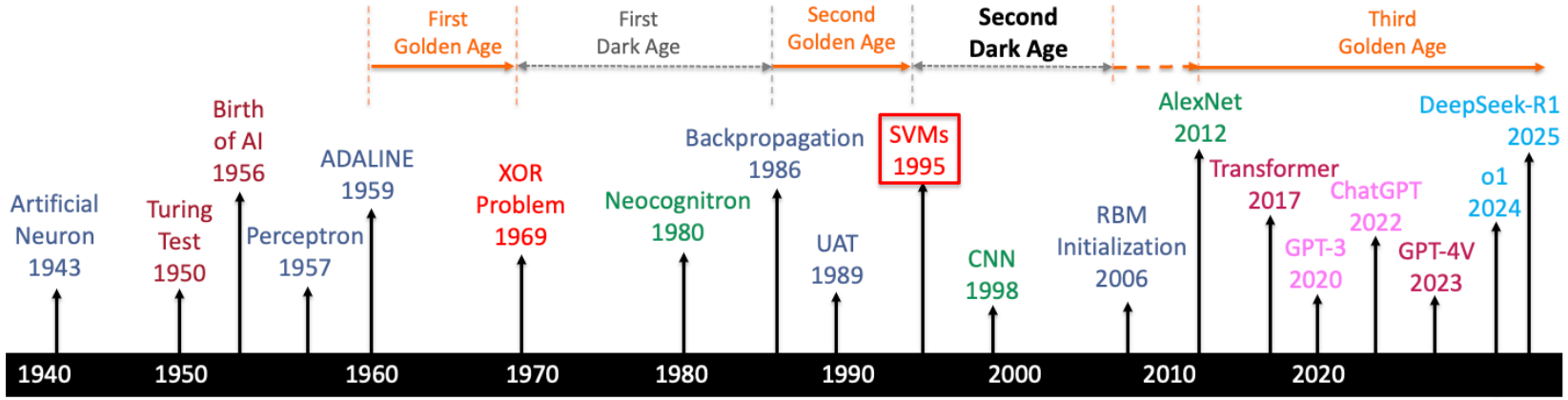
- The Universal Approximation Theorem (UAT), proposed by **George Cybenko** in 1989, provided a mathematical foundation for the capabilities of multilayer neural networks.
- The theorem states that a feedforward neural network with a single hidden layer can approximate any continuous function to an arbitrary degree of accuracy, **given sufficient neurons and using non-linear activation functions.**
- This theorem underscores the power and flexibility of neural networks, making them suitable for a wide range of applications.

One Hidden Layer is Enough



Second Dark Ages (Early 1990s — Early 2000s)

- The field of neural networks experienced a "**second dark age**" due to:
  - Rise of Support Vector Machines (SVMs)
  - Computational limitations to implement deep neural networks
  - Overfitting and generalization issues
- These challenges led to a shift in focus away from neural networks, causing stagnation in the field.



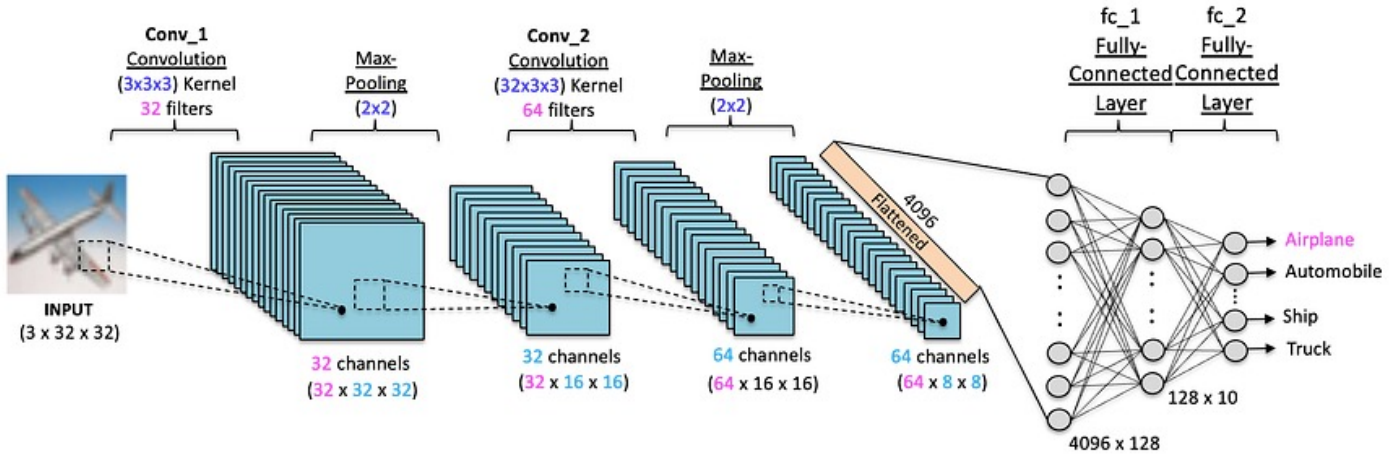
Convolutional Neural Networks for Computer Vision (1980s – 2010s)

Main Topics of the Course

- **Course Overview**
- **History of AI with Deep Learning**
- **Review of Linear Algebra, Calculus and Probability**
- **Multi-Layer Perceptron (MLP)**
  - Perceptron and MLP
  - Activation function, Loss function
  - Backpropagation and Gradient Descent
  - Regularizations and Optimizations
- **Convolutional Neural Networks (CNNs)**
  - Convolutional Neural Layer
  - LeNet, AlexNet, VGGNet, InceptionNet, ResNet, DenseNet, EfficientNet
  - Computer Vision Applications:
    - Object Detection, Image Segmentation
- **Recurrent Neural Network (RNN)**
  - Introduction to NLP
  - Tokenization and Word Embeddings
  - Vanilla RNN, LSTM, GRU
  - Encoder-Decoder Models and Attention
- **Transformers and Large Language Models (LLMs)**
  - Self-Attention Mechanism
  - Transformers
  - BERT, GPT, GPT-2, GPT-3, T5 and BART
  - LLM Decoding and Prompt Engineering
  - Model Quantization
  - Parameter-Efficient Fine-Tuning (PEFT)
- **Preference Alignment and Multimodality**
  - Preference Alignment: Instruction-Tuning, RLHF, DPO, ORPO
  - Multimodality: CLIP, GPT-4V, LLaVA

Convolutional Neural Networks (1980s - 2010s)

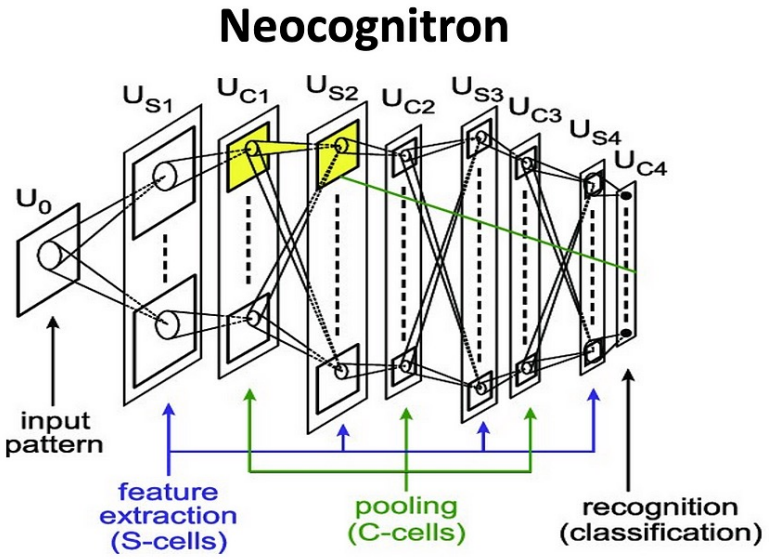
- **Convolutional Neural Networks (CNNs)** have dramatically transformed the landscape of deep learning, particularly in the fields of computer vision and image processing.
- Their evolution from the 1980s to the 2010s reflects significant advancements in architecture, training techniques, and applications.



https://medium.com/@lmpo/mastering-the-basics-of-convolutional-neural-networks-cnns-e98a732ea127

Neocognitron (Early 1980s)

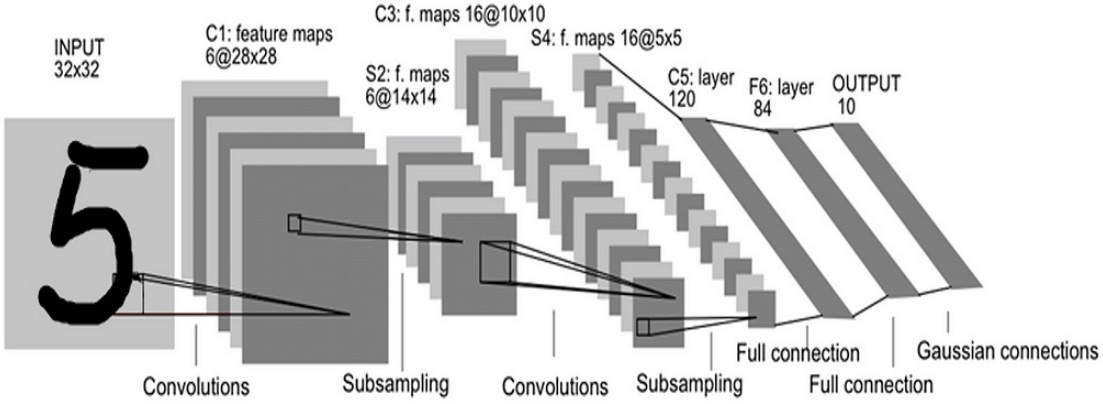
- The concept of CNNs was first introduced in the 1980s by Kenji Fukushima, who proposed the **Neocognitron**, a hierarchical neural network that mimicked the structure of the human visual cortex.
- This pioneering work laid the foundation for the development of CNNs.



Kenji Fukushima

LeNet-5 (1989–1998)

- In the late 1980s and early 1990s, Yann LeCun and his team further developed CNNs, introducing the **LeNet-5** architecture, which was specifically designed for handwritten digit recognition.



LeNet-5

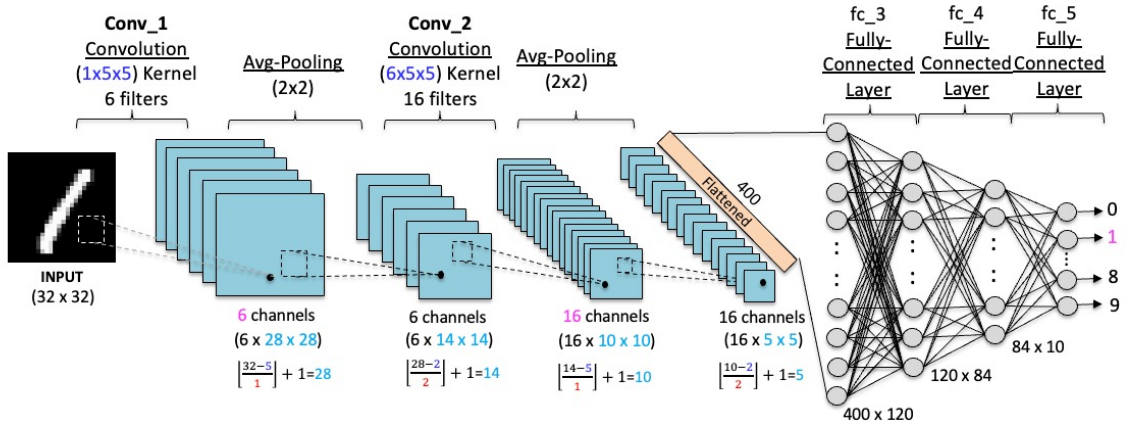
Yann LeCun

Handwritten Digits CNNs Demo (1989)

https://www.youtube.com/watch?v=FwFduRA_L6Q

Key Components of CNNs

- CNNs are constructed by three key components:
  1. **Convolutional Layers**: These layers automatically learn spatial hierarchies of features from input images by applying a set of learnable filters.
  2. **Pooling Layers**: Pooling layers reduce the spatial dimensions of the input, enhancing robustness to variations and decreasing computational load.
  3. **Fully Connected Layers**: Following convolutional and pooling layers, fully connected layers are used for classification tasks, integrating features learned from previous layers.

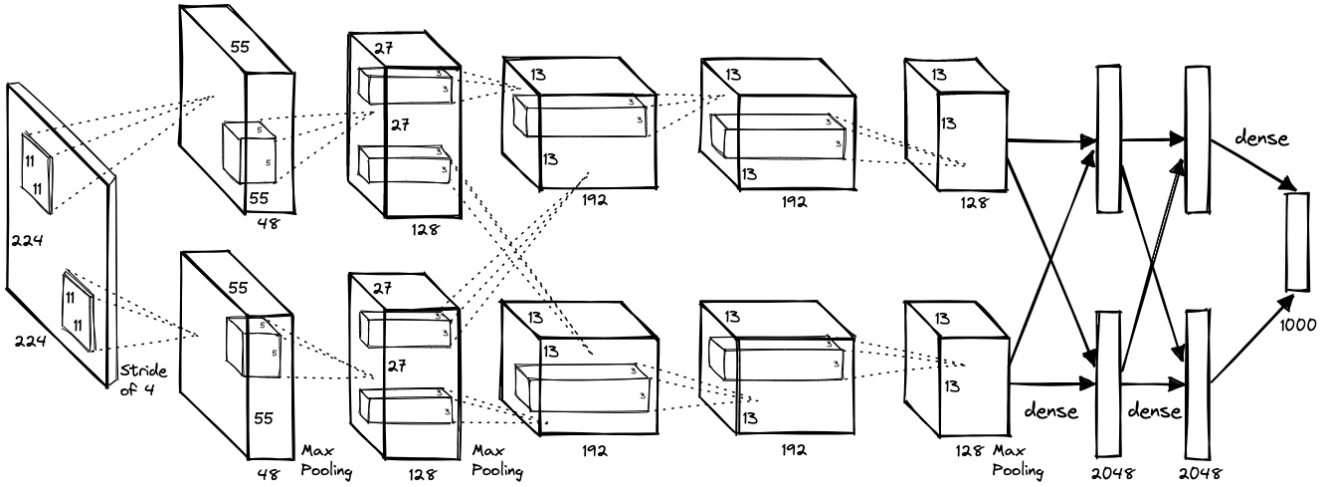


Key Features of CNNs

- **Local Receptive Fields**: CNNs use local receptive fields to capture local patterns in the input data, making them highly effective for image and visual tasks.
- **Shared Weights**: The use of shared weights in convolutional layers reduces the number of parameters in the network, making it more efficient and easier to train.
- **Translation Invariance:** Pooling layers introduce translation invariance, allowing the network to recognize patterns regardless of their position in the input image.

The Rise of CNNs: AlexNet's Impact (2012)

- In 2012, a major milestone was reached in the development of CNNs, as AlexNet emerged victorious in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), achieving a significant margin of victory and marking a significant breakthrough in image classification.

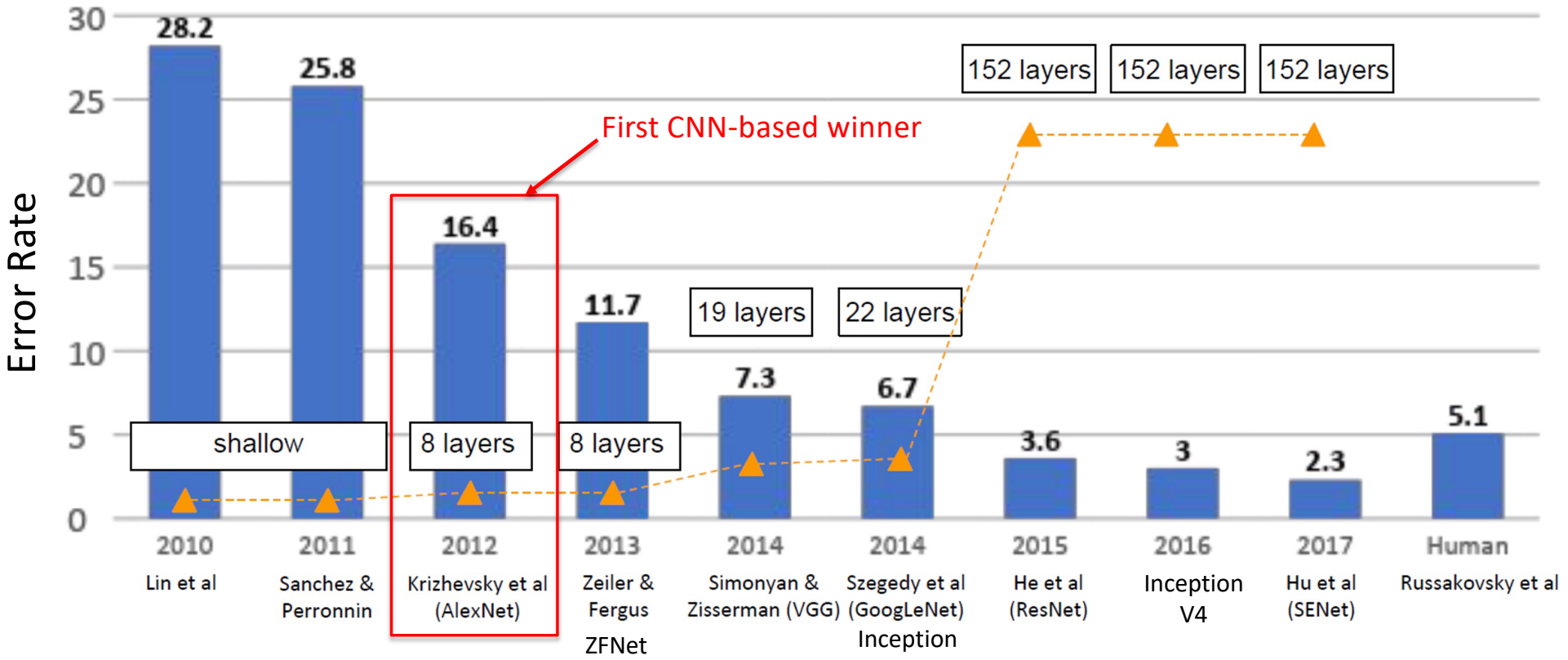


Architecture of the AlexNet (2012)

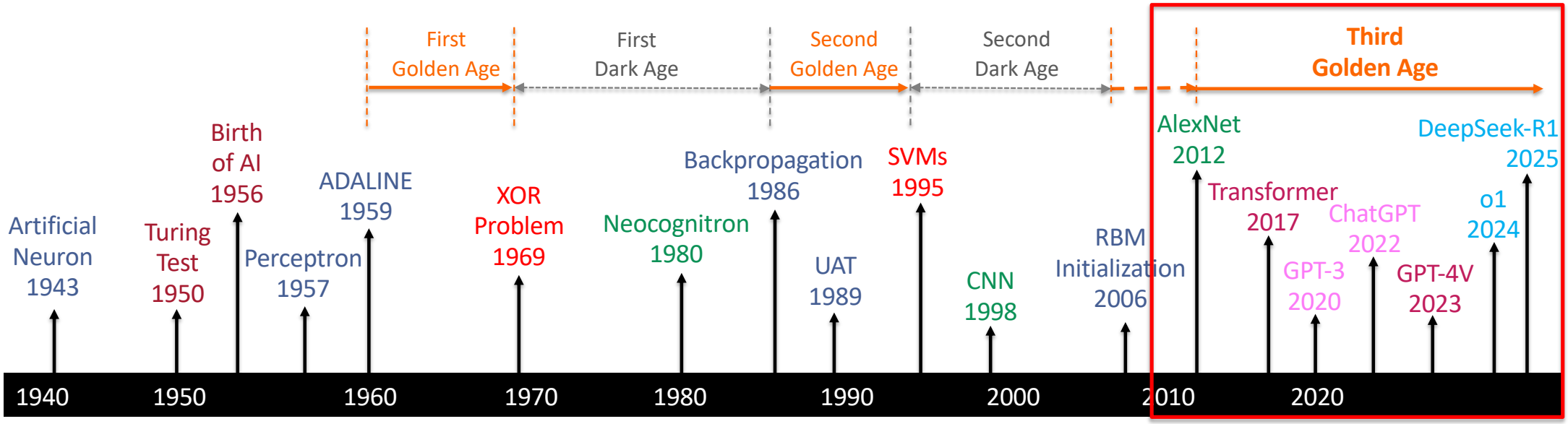
AlexNet Features

- The ILSVRC is an annual image recognition benchmark that evaluates algorithms on a dataset of over 10 million annotated images, categorized into 1000 classes. AlexNet's innovations included:
  1. **ReLU Activation Functions**: Introduced to overcome issues with traditional activation functions, ReLU enabled faster training and improved performance.
  2. **Dropout Regularization**: This technique reduced overfitting by randomly dropping units during training.
  3. **Data Augmentation**: Enhancements to the training dataset improved generalization by artificially increasing the diversity of the training data.

Winners of ImageNet Classification Challenge (ILSVRC)



AlexNet Unleashes the "Third Golden Age" of Neural Networks



The current golden age (2010s-present) is marked by the convergence of deep learning, big data, and powerful computing platforms. This era has seen remarkable breakthroughs in image recognition, natural language processing, and robotics. Ongoing research continues to push the boundaries of AI capabilities.

Evolution of CNN Architectures

- **LeNet-5** – First CNN for handwritten digit recognition (1989-1998)
- **AlexNet** – “ImageNet moment” (2012)
- **VGGNet** – stacking 3x3 layers (2014)
- **Inceptions** – parallel branches (2014)
- **ResNet** – identity shortcuts (2015)
- **ResNeXt** – grouped convolution (2016)
- **DenseNet** – dense shortcuts (2016)
- **MobileNets** – depthwise conv; inverted residuals (2017/18)
- **EfficientNet** – model scaling (2019)
- **RegNet** – design spaces (2020)
- **ConvNeXt** – (2022)

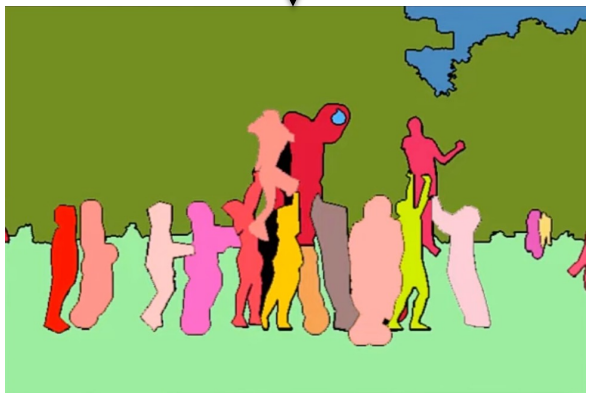
CNNs for Computer Vision Applications

https://www.v7labs.com/blog/image-segmentation-guide

Image Segmentation

Sematic Segmentation

Instance Segmentation



Panoptic Segmentation

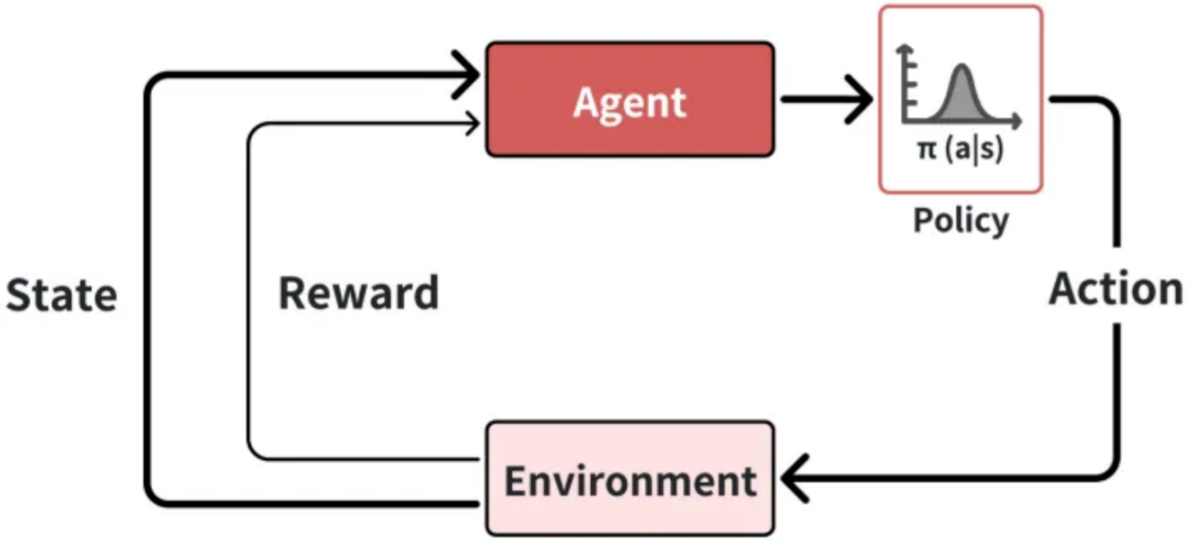
CNN-based Object Detection and Image Segmentation

- **Two Stage Object Detection**
  - R-CNN (Region-CNN)
  - Fast R-CNN
  - Faster R-CNN
- **One Stage Object Detection**
  - YOLO (You Only Look One)
  - SSD
  - RetinaNet
- **Semantic Segmentation**
  - SegNet (2015)
  - U-Net (2015)
  - PSPNet (2017)
- **Instance Segmentation**
  - Mask R-CNN (2017)
- **Panoptic Segmentation**

Deep Reinforcement Learning (2013–Present)

Deep Reinforcement Learning (2013–Present)

- Paradigm Shift: Combining Deep Learning with Reinforcement Learning
  - Overcoming the "curse of dimensionality" using deep neural networks for value functions and policies.



Architecture of reinforcement learning. (Source)

Deep Q-Networks (DQN) and Atari Breakthrough (2013)

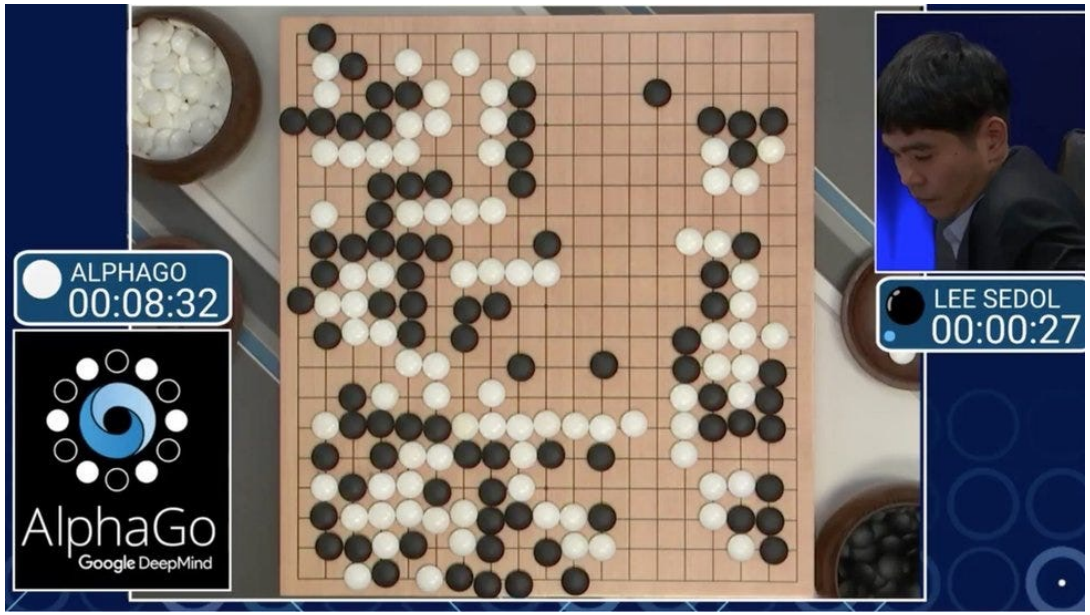
- **Introduced by DeepMind (acquired by Google in 2014), led by Volodymyr Mnih.**
  - Agent learns Atari 2600 games from raw pixels, no game-specific rules.
  - Achieved human-level or superhuman performance on Breakout, Pong, Space Invaders.
- **Key Innovations:** Experience Replay (breaks data correlation) and Target Networks (stabilizes Q-function learning).
- Proved single architecture can learn diverse tasks via trial and error.

Google DeepMind's Deep Q-learning playing Atari Breakout!

AlphaGo and the Move 37 Moment (2016)

DeepMind's AlphaGo defeated Go world champion Lee Sedol.

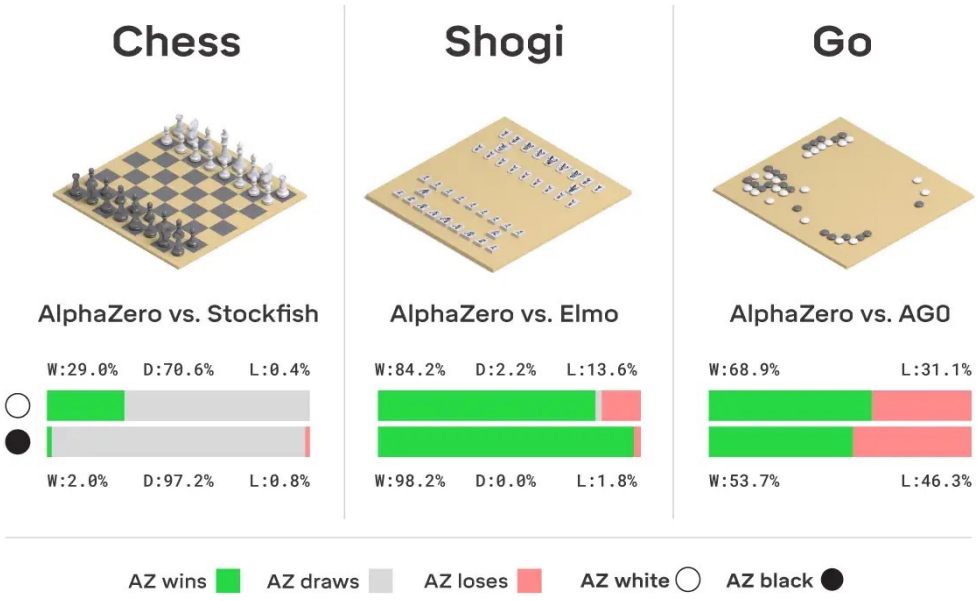
- Go's vast search space (10^{170} positions) makes brute-force impossible.
- **Move 37**: Seemingly mistaken move that revealed deep, original strategy.
- **Architecture**: Monte Carlo Tree Search (MCTS) + Policy Network (move selection) + Value Network (position evaluation).



Artificial intelligence: Google's AlphaGo beats Go master Lee Se-dol.

AlphaZero: Learning from Scratch (2017)

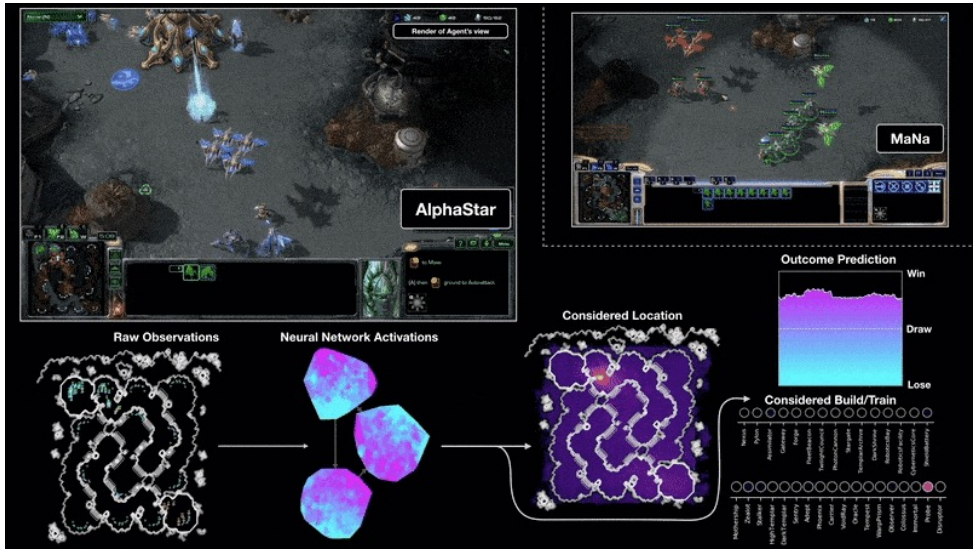
- **Successor to AlphaGo**, trained tabula rasa (blank slate) without human games.
- Used self-play: Millions of games against itself, starting from rules only.
- In 24 hours, defeated world-champion programs in Chess (Stockfish), Shogi (Elmo), and Go (AlphaGo Zero).
- Demonstrated AI can surpass centuries of human knowledge through self-optimization.



https://deepmind.google/blog/alphazero-shedding-new-light-on-chess-shogi-and-go/

Scaling to Complex Worlds (2019)

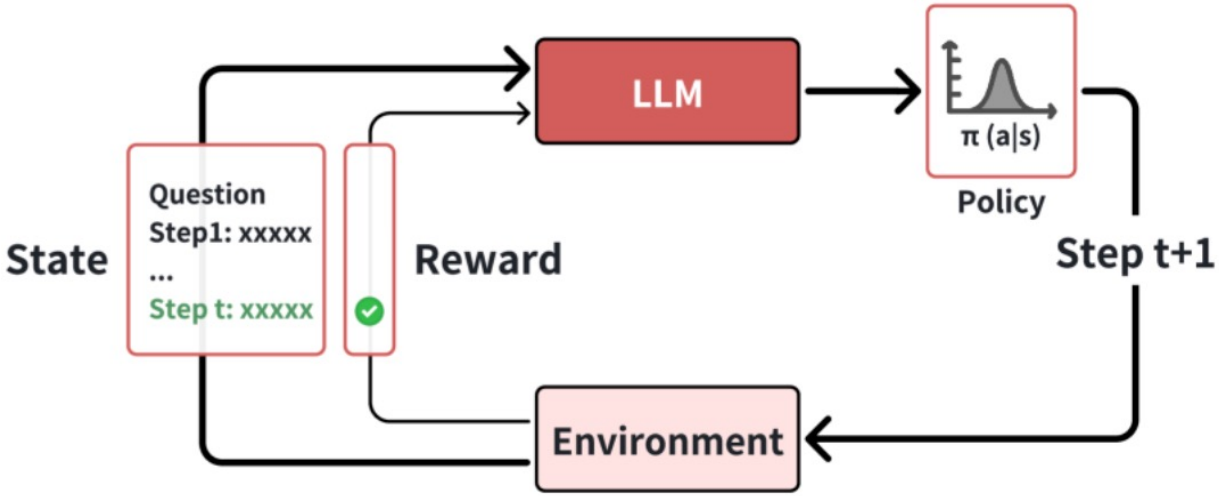
- **OpenAI Five:** AI team defeated world champion OG in Dota 2.
  - Learned via 180 years' worth of daily self-play games.
  - Mastered long-term planning and team coordination.
- **AlphaStar:** DeepMind's agent reached Grandmaster in StarCraft II.
  - Handled vast action spaces, imperfect information, and real-time strategy.



https://deepmind.google/blog/alphastar-mastering-the-real-time-strategy-game-starcraft-ii/

RLHF: Bridging RL and LLMs (2020s)

- **Reinforcement Learning from Human Feedback (RLHF):** Aligns LLMs (e.g., GPT-4) with human intent.
  - Solves alignment problem: LLMs can be toxic/untruthful from next-token prediction alone.
  - Process: Train Reward Model on human preferences, use Proximal Policy Optimization (PPO) to fine-tune.
  - Critical for transforming models into helpful assistants like ChatGPT.



(Source)

DRL Summary

- DRL milestones show AI's path to general intelligence: From Atari to complex games and LLM alignment.
- Future Impact: Self-optimizing systems in real-world applications like robotics, healthcare, and beyond.
- **Key Takeaway:** Trial-and-error learning at scale unlocks unprecedented capabilities.

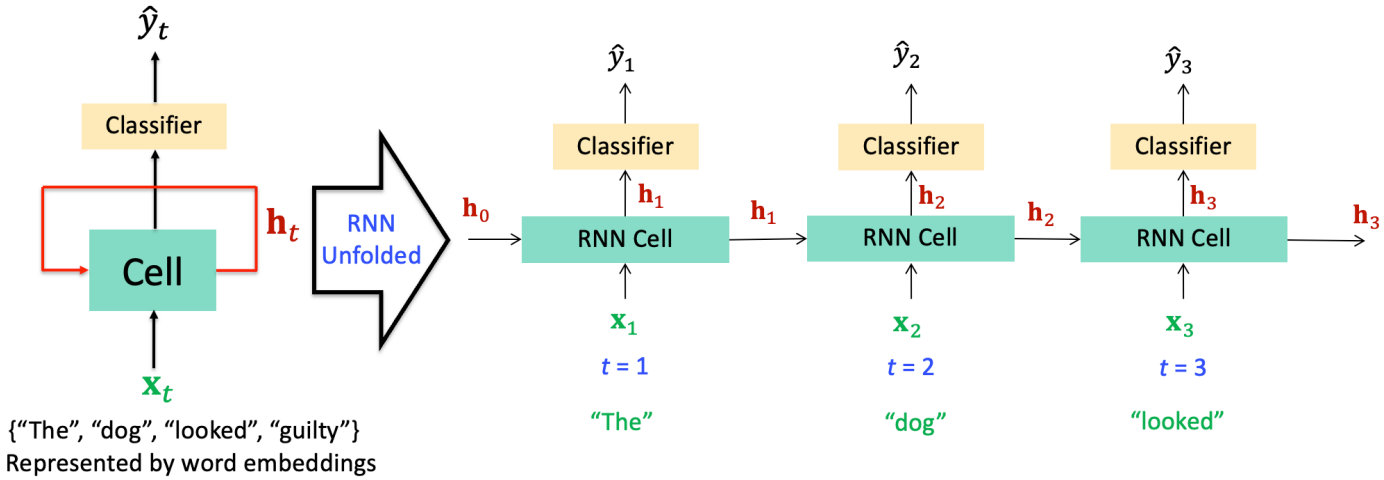
Recurrent Neural Networks (1982 – 2017)

Main Topics of the Course

- **Course Overview**
- **History of AI with Deep Learning**
- **Review of Linear Algebra, Calculus and Probability**
- **Multi-Layer Perceptron (MLP)**
  - Perceptron and MLP
  - Activation function, Loss function
  - Backpropagation and Gradient Descent
  - Regularizations and Optimizations
- **Convolutional Neural Networks (CNNs)**
  - Convolutional Neural Layer
  - LeNet, AlexNet, VGGNet, InceptionNet, ResNet, DenseNet, EfficientNet
  - Computer Vision Applications:
    - Object Detection, Image Segmentation
- **Recurrent Neural Network (RNN)**
  - Introduction to NLP
  - Tokenization and Word Embeddings
  - Vanilla RNN, LSTM, GRU
  - Encoder-Decoder Models and Attention
- **Transformers and Large Language Models (LLMs)**
  - Self-Attention Mechanism
  - Transformers
  - BERT, GPT, GPT-2, GPT-3, T5 and BART
  - LLM Decoding and Prompt Engineering
  - Model Quantization
  - Parameter-Efficient Fine-Tuning (PEFT)
- **Preference Alignment and Multimodality**
  - Preference Alignment: Instruction-Tuning, RLHF, DPO, ORPO
  - Multimodality: CLIP, GPT-4V, LLaVA

Recurrent Neural Networks (1986 - 2017)

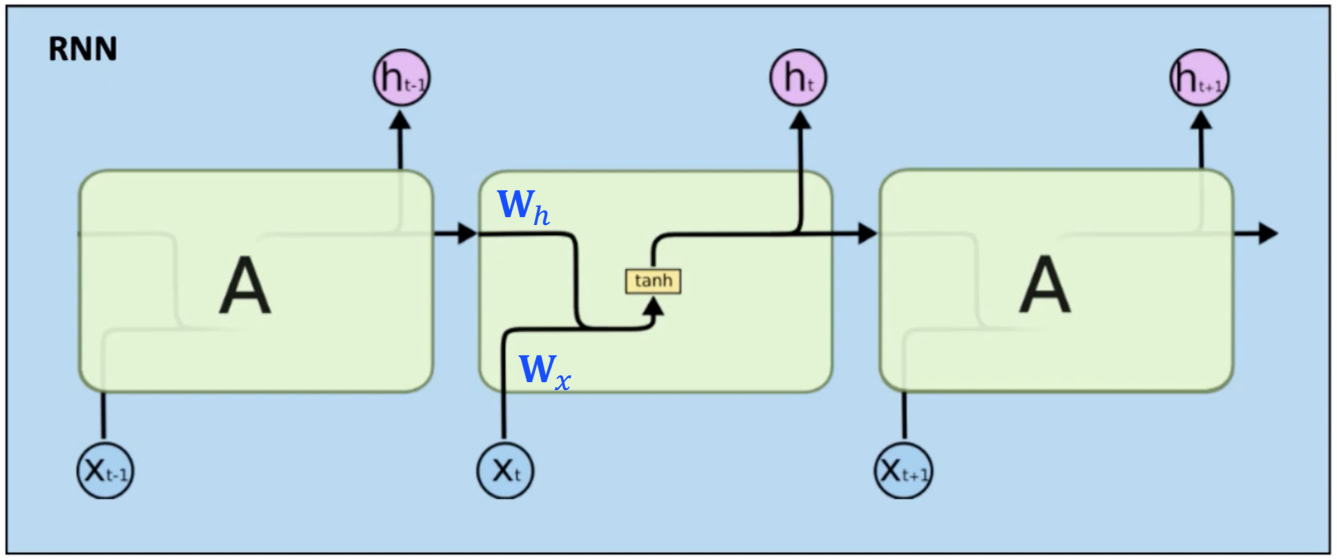
- **Recurrent Neural Networks (RNNs) (1986)** emerged as a powerful architecture for **handling sequential and temporal data**.
- Unlike feedforward neural networks, RNNs are designed to process sequences of inputs, making them particularly effective for tasks such as language modeling, time series forecasting, and speech recognition.



Early Developments (1980s-1990s)

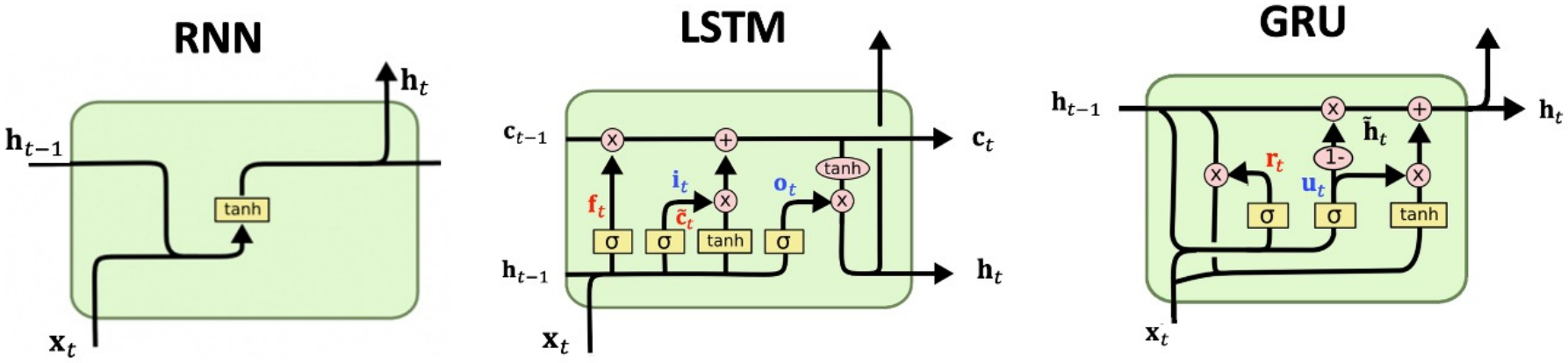
- The concept of RNNs dates back to the 1980s, with pioneers like **John Hopfield, Michael I. Jordan, and Jeffrey L. Elman** contributing to their development.

$$\mathbf{h}_t = tanh(\mathbf{W}_x x_t + \mathbf{W}_h h_{t-1} + \mathbf{b}_h)$$



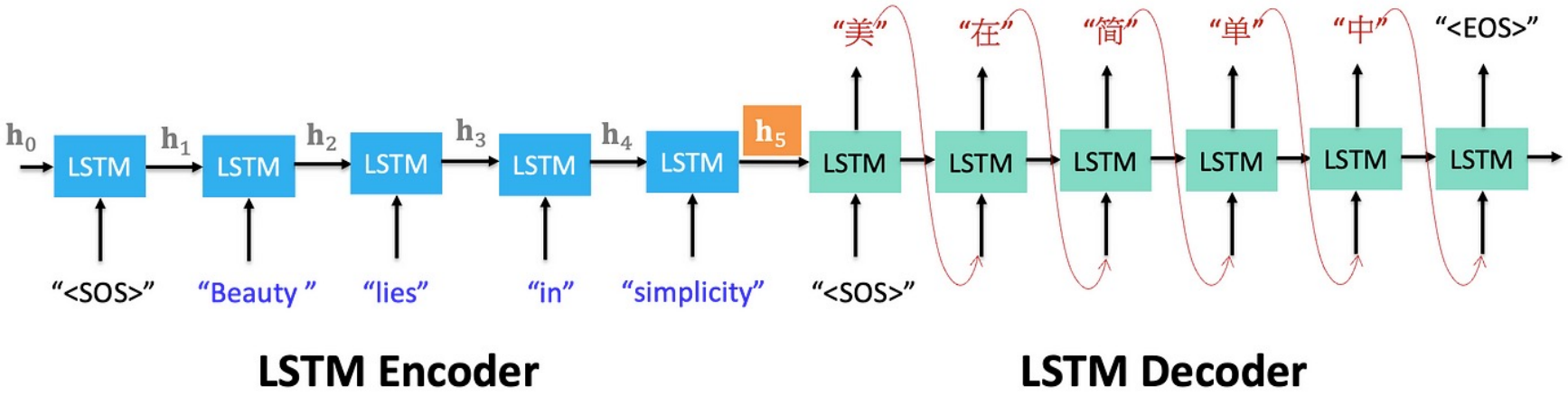
LSTM, GRU and Seq2Seq Models (2000s - 2010s)

- **LSTM (1997)**: Introduced by Hochreiter & Schmidhuber to address vanishing gradients in RNNs. Uses gating mechanism for long-term dependencies.
- **GRU (2014)**: Proposed by Cho et al. Simplified LSTM with fewer parameters, faster training, also uses gating mechanism.

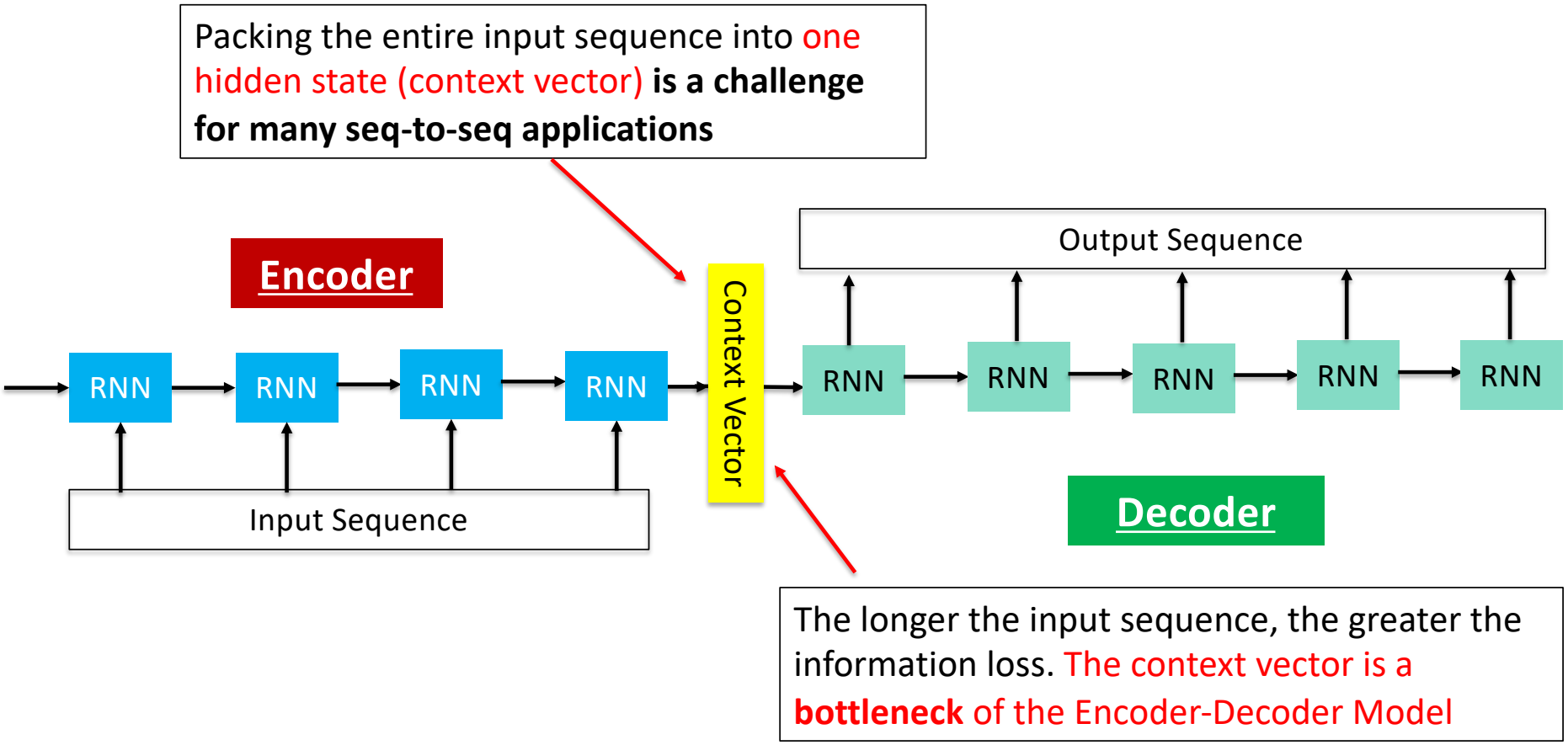


LSTM, GRU and Seq2Seq Models (2000s - 2010s)

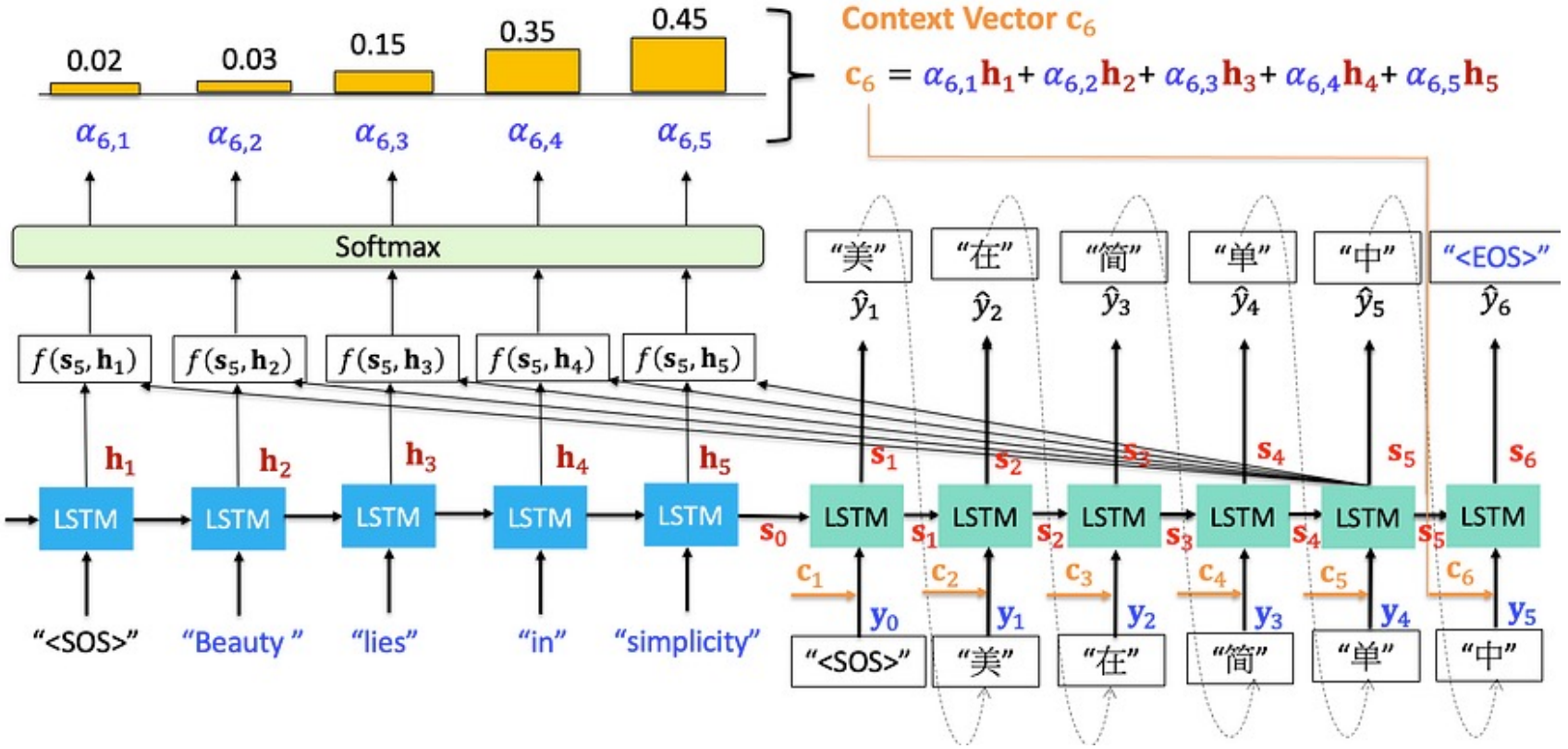
- **Sequence-to-Sequence Models (Seq2Seq) (2014)**: Ilya Sutskever and his team introduced the Seq2Seq model, which uses an encoder-decoder architecture to map input sequences to output sequences. This model has been widely used for tasks such as machine translation, speech recognition, and text summarization.



RNN Encoder-Decoder: Bottleneck



Attention Mechanism



Example of Attention

- The heatmap on the right shows where the model is paying more attention to when translating a sentence:

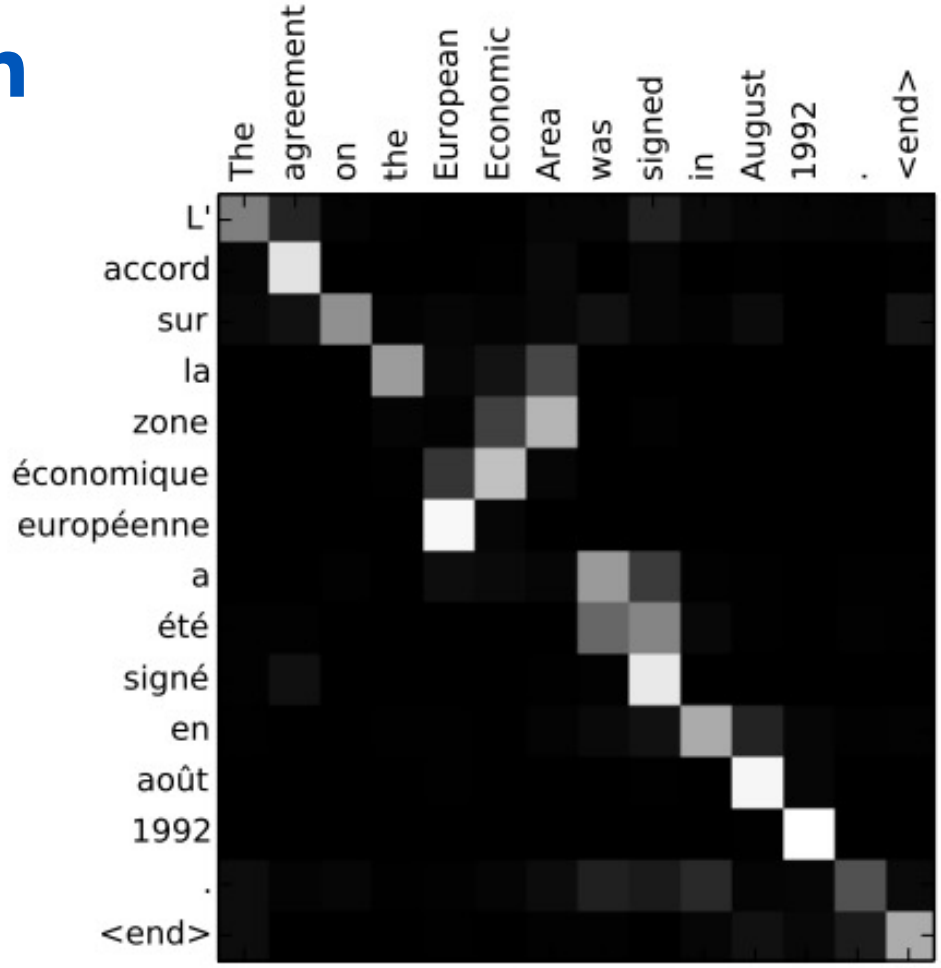


Image from [‘Neural Machine Translation by jointly learning to align and translate’](#) by Bzmitry Bahdanau et al in 2015.

Seq2Seq with Attention vs Traditional Seq2Seq

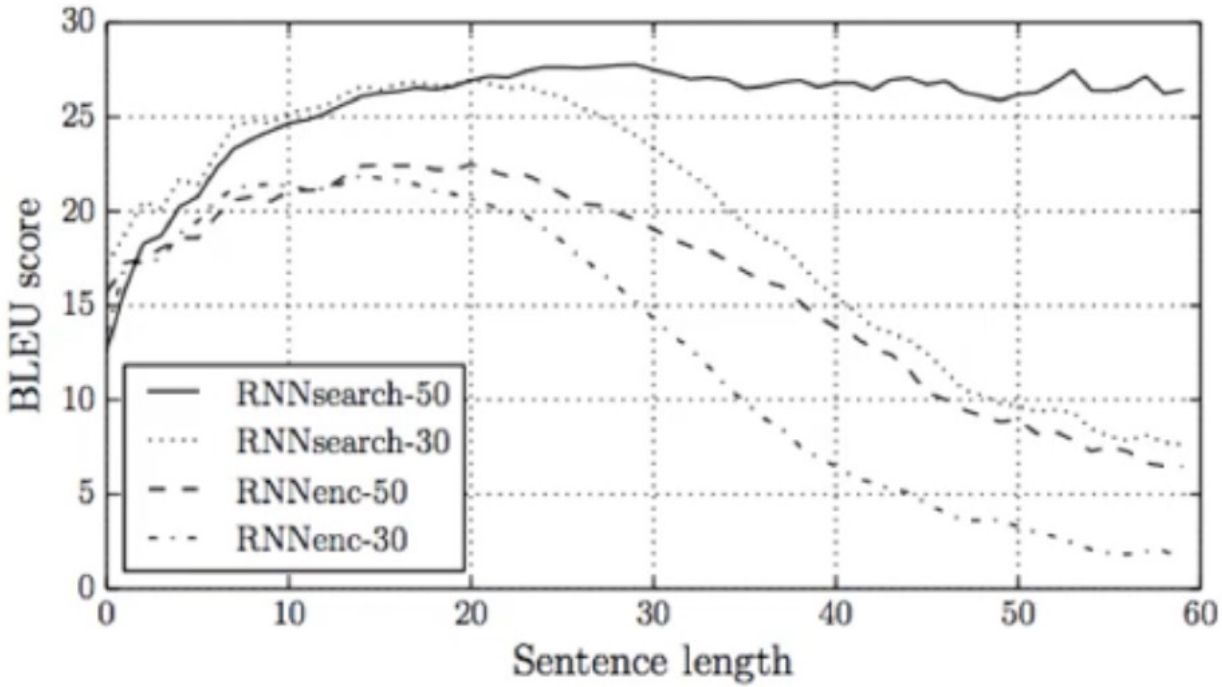
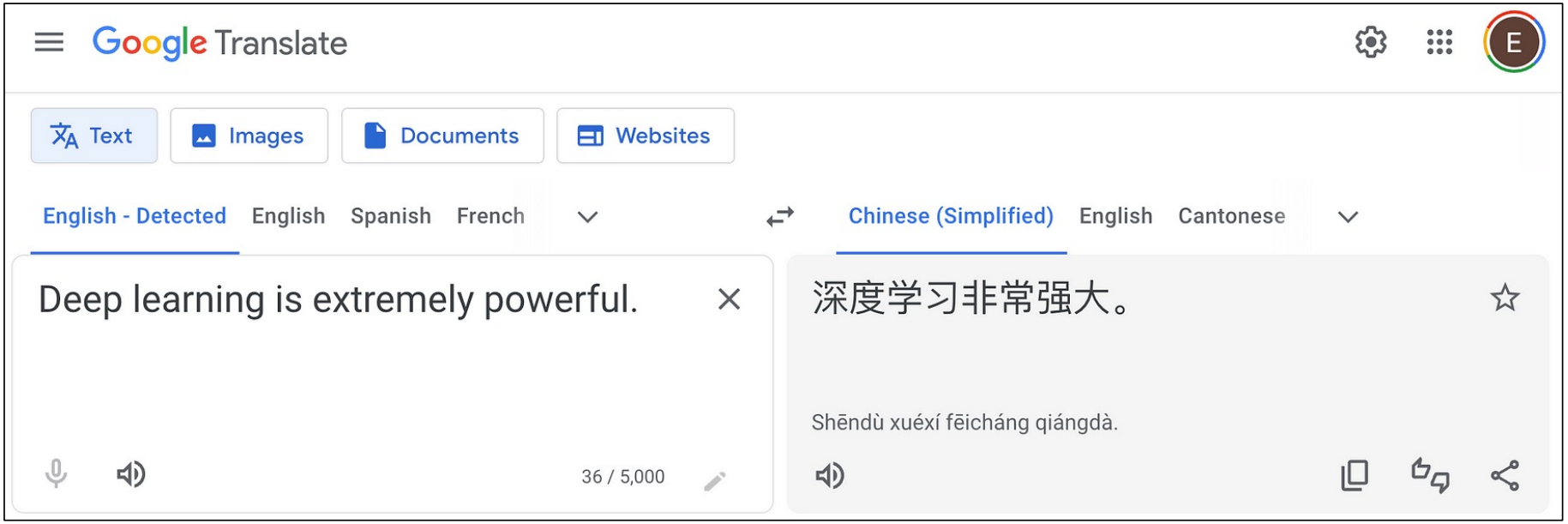


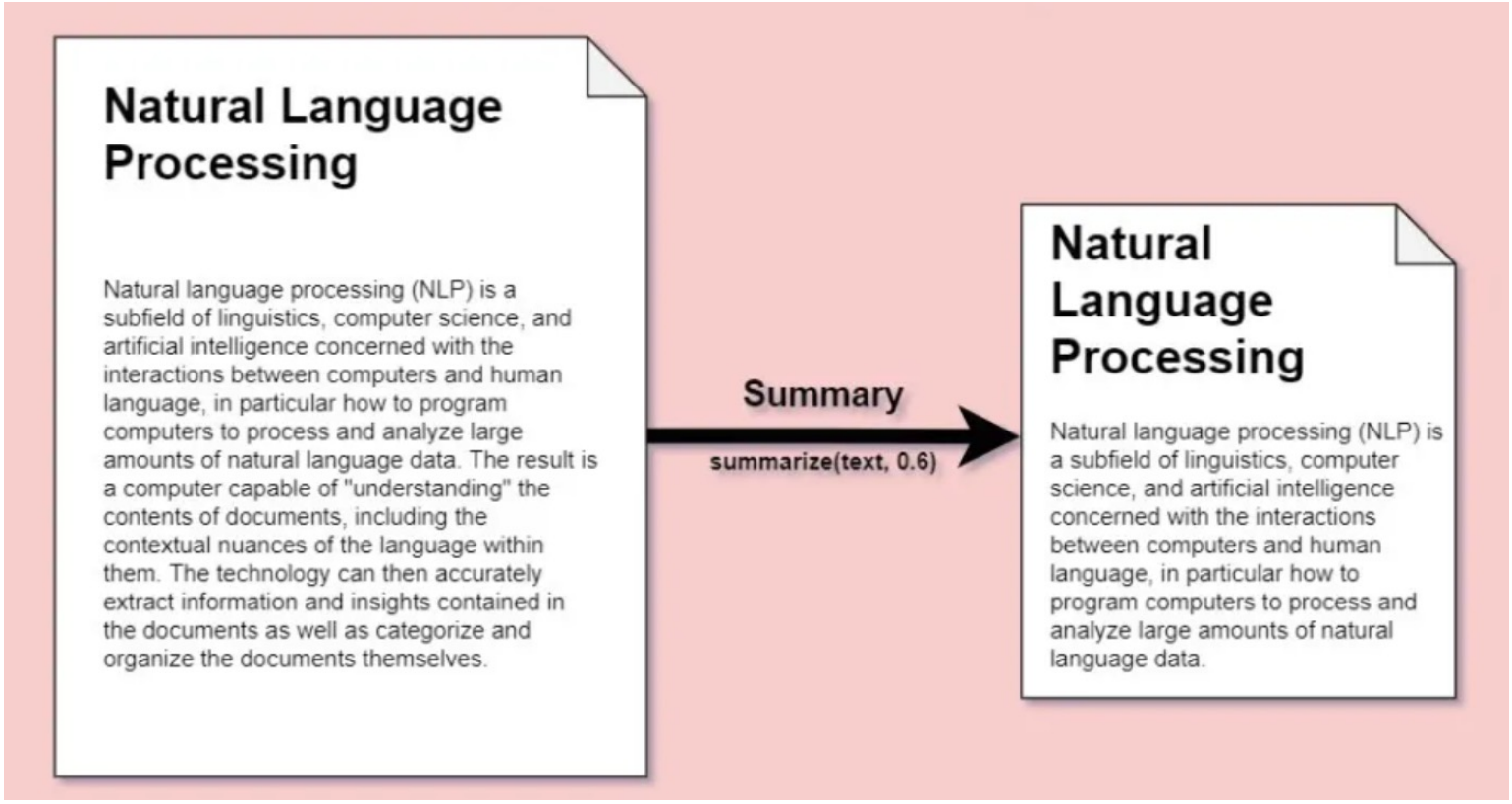
Figure 2: The BLEU scores of the generated translations on the test set with respect to the lengths of the sentences. The results are on the full test set which includes sentences having unknown words to the models.

Google Translate

- In 2016, **Google Translate switched to** a neural machine translation system, dramatically improving translation quality



Seq2Seq: Text Summarization



https://turbolab.in/types-of-text-summarization-extractive-and-abstractive-summarization-basics/

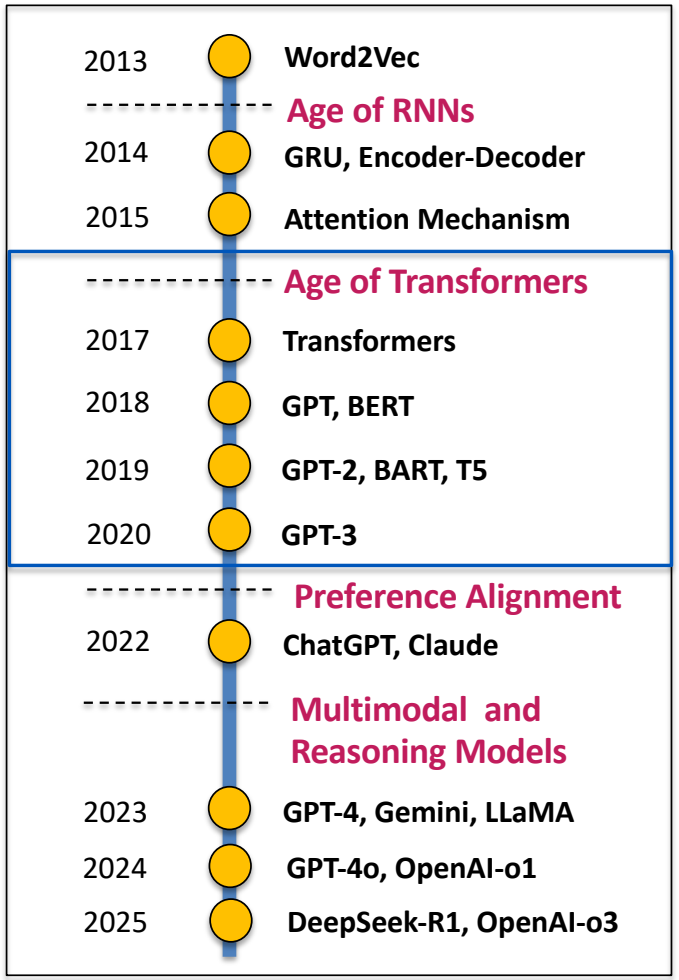
Transformers (Self-Attention) (2017 – Present)

Main Topics of the Course

- **Course Overview**
- **History of AI with Deep Learning**
- **Review of Linear Algebra, Calculus and Probability**
- **Multi-Layer Perceptron (MLP)**
  - Perceptron and MLP
  - Activation function, Loss function
  - Backpropagation and Gradient Descent
  - Regularizations and Optimizations
- **Convolutional Neural Networks (CNNs)**
  - Convolutional Neural Layer
  - LeNet, AlexNet, VGGNet, InceptionNet, ResNet, DenseNet, EfficientNet
  - Computer Vision Applications:
    - Object Detection, Image Segmentation
- **Recurrent Neural Network (RNN)**
  - Introduction to NLP
  - Tokenization and Word Embeddings
  - Vanilla RNN, LSTM, GRU
  - Encoder-Decoder Models and Attention
- **Transformers and Large Language Models (LLMs)**
  - Self-Attention Mechanism
  - Transformers
  - BERT, GPT, GPT-2, GPT-3, T5 and BART
  - LLM Decoding and Prompt Engineering
  - Model Quantization
  - Parameter-Efficient Fine-Tuning (PEFT)
- **Preference Alignment and Multimodality**
  - Preference Alignment: Instruction-Tuning, RLHF, DPO, ORPO
  - Multimodality: CLIP, GPT-4V, LLaVA

LLMs: From Word2Vec to DeepSeek-R1 (2012-2025)

1. Tokenization and Word2Vec: BPE, CBOW, Skip-Gram
2. RNNs, LSTM, GRU, Seq-to-Seq (Encoder-Decoder) and Attention Mechanism
3. Transformers with Self-Attention
4. Lager Language Models (LLMs): BERT, GPT, BART, T5
5. Preference Alignment by SFT and RLHF: ChatGPT, Claude, LLaMA
6. Multimodal Models: GPT-4, GPT-4o, Gemine, LLaVA
7. Reasoning Models: OpenAI-o1, DeepSeek-R1



Attention Is All You Need (2017-06)

- In 2017, **Vaswani et al**. published "*Attention is All You Need*," introducing the **Transformer** architecture. It replaced RNNs with **self-attention**, enabling:
  - Parallel processing (faster training).
  - Long-range dependency modeling.
  - Scalability to massive datasets.

Attention Is All You Need

https://arxiv.org/pdf/1706.03762.pdf

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

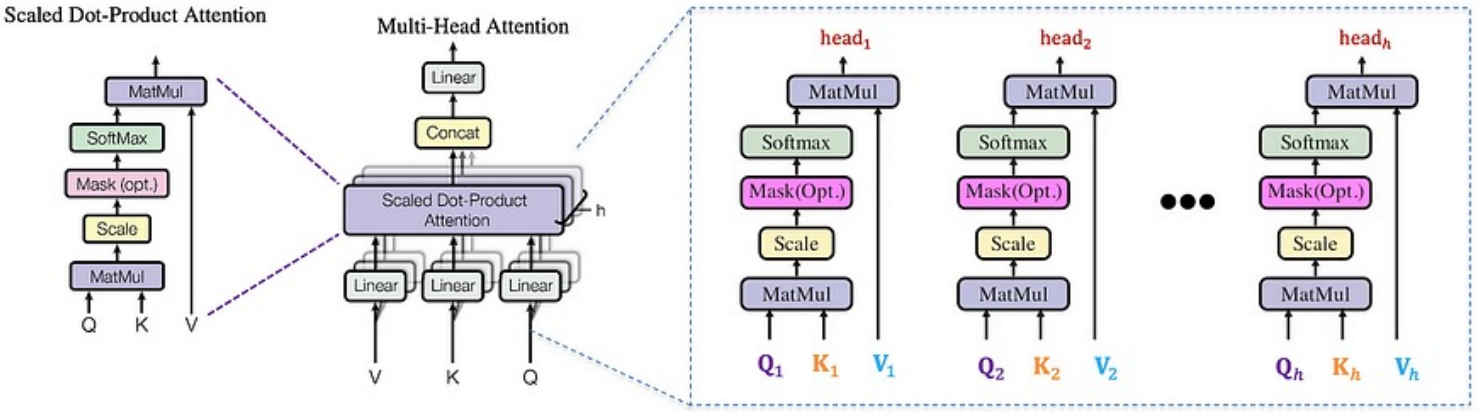
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

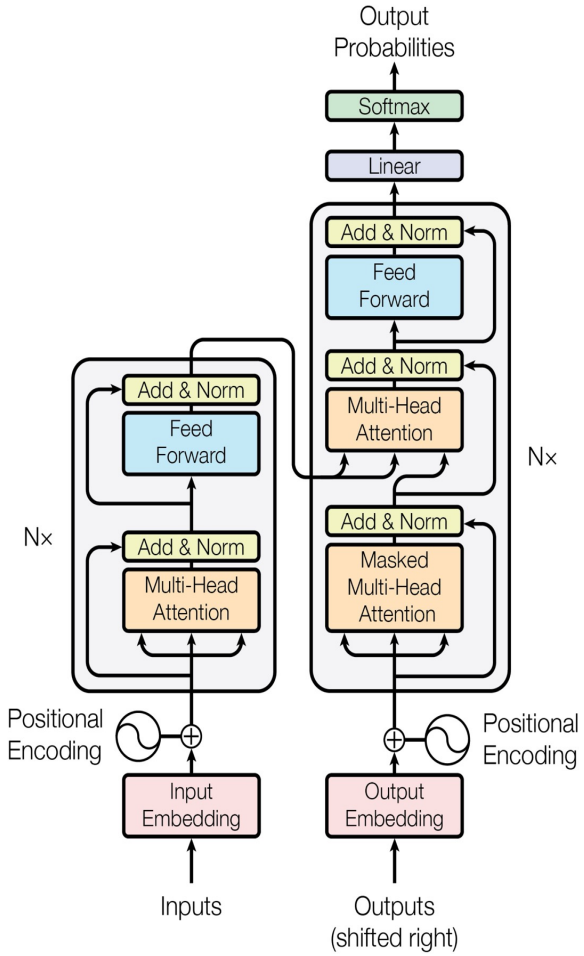
Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

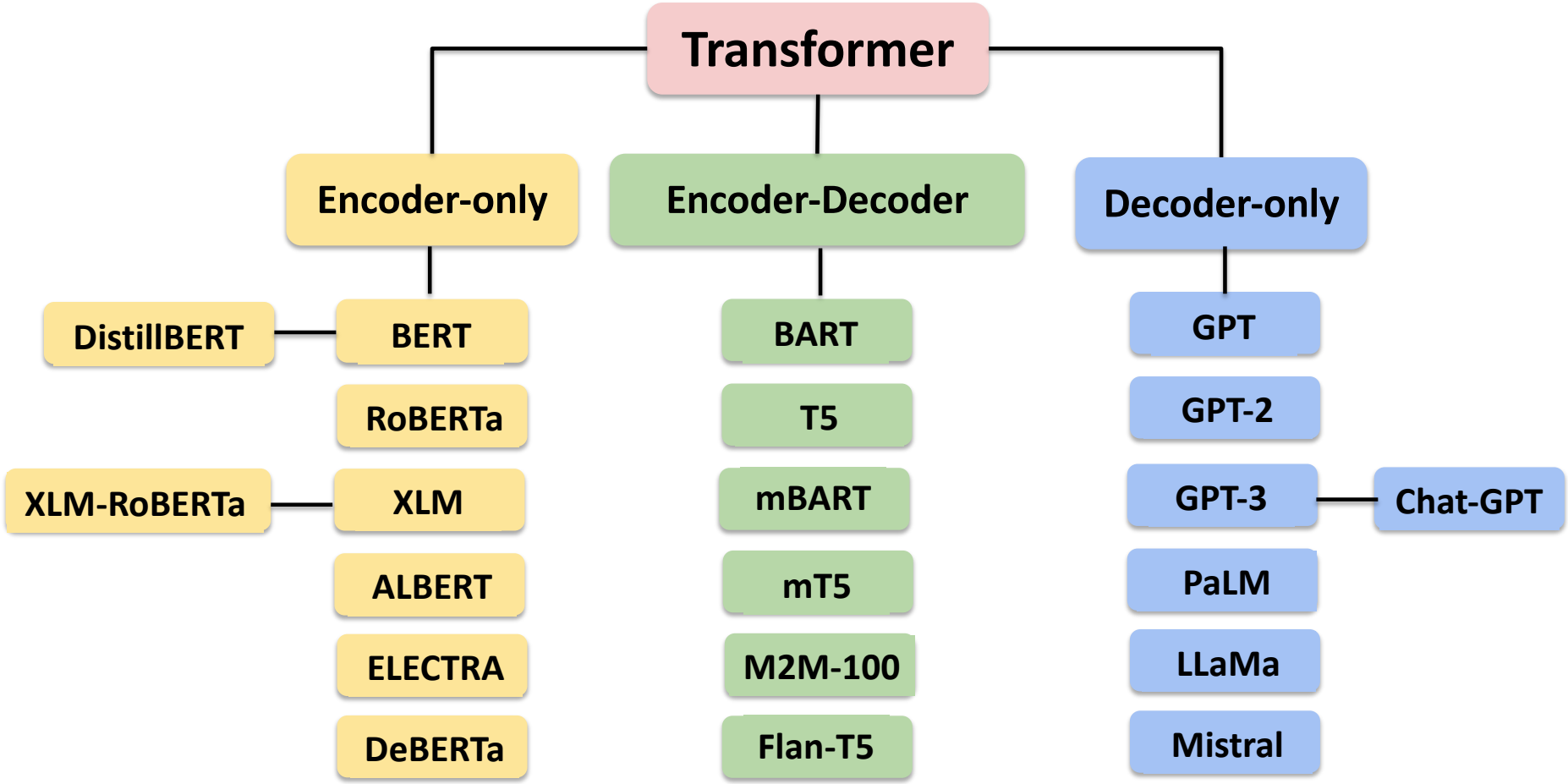
Key Features of Transformers

- **Self-Attention**: Weights importance of all tokens.
- **Multi-Head Attention**: Multiple attention heads for richer context.
- **Positional Encoding**: Preserves token order.





Transformer-based Models



Well-known Transformer-based Models

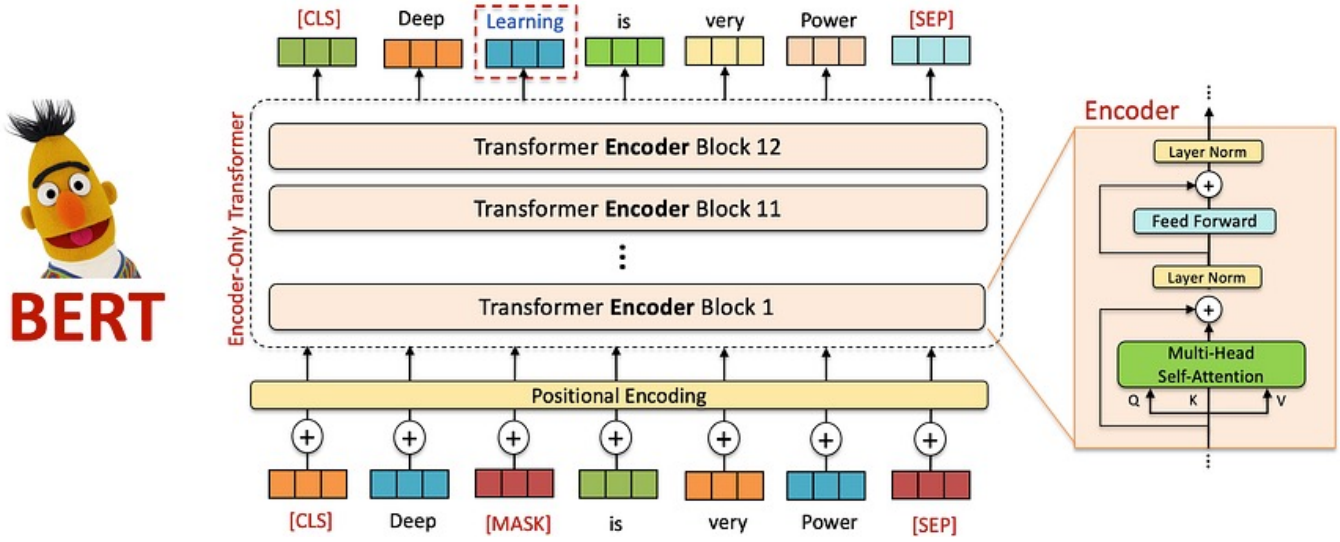
The original Transformer Model (2017-06)

- **Encoder-only Transformer Models**
  - BERT Series: BERT (2018-10), DistillBERT, RoBERTa, XLM, XLM-RoBERTa, ALBERT, ELECTRA
- **Decoder-only Transformer Models**
  - GPT Series: GPT-1 (2018-6), GPT-2 (2019-02), GPT-3 (2020-06), InstructGPT (2022-02), ChatGPT (2022-11), GPT-4 (2023-03)
- **Encoder-Decoder Transformer Models**
  - BART, T5, mBART, mT5, M2M-100, BigBird

https://www.youtube.com/watch?v=wuj8Hao1TT4

BERT: Bidirectional Contextual Understanding (2018)

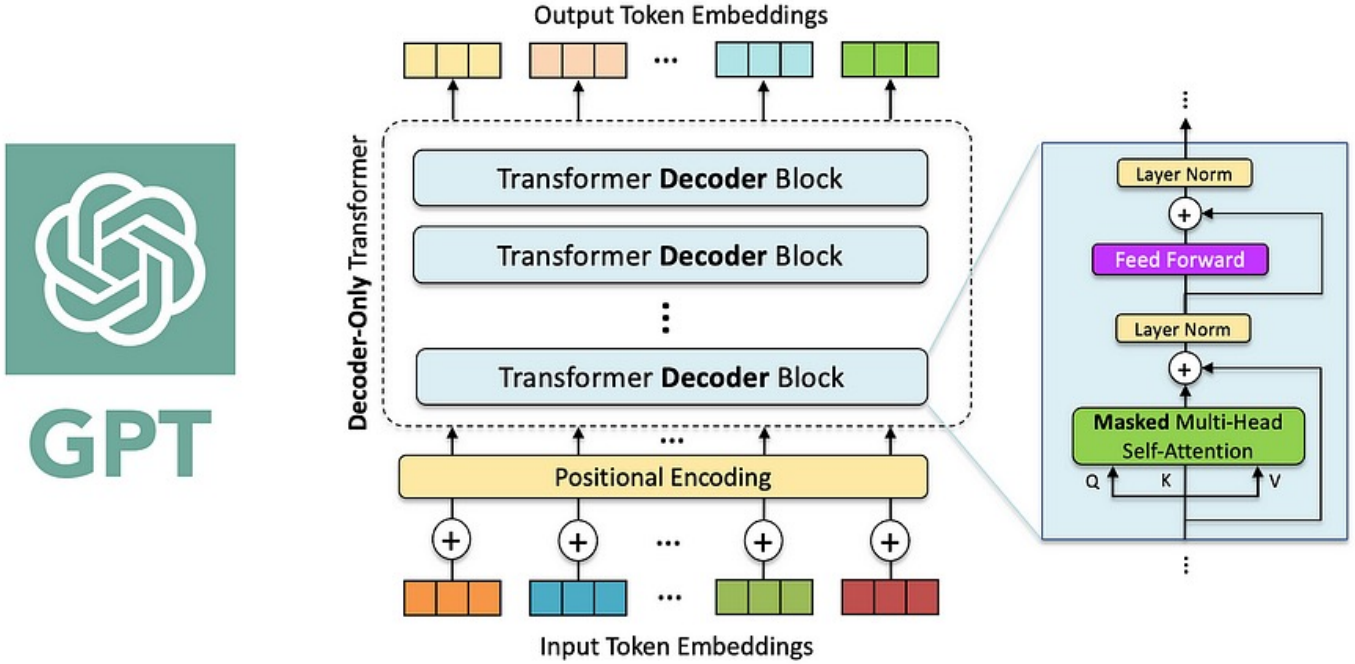
- **BERT (Google, 2018):** Encoder-only, bidirectional model. Trained via Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). Excelled in language understanding



BERT is the **first encoder-only Transformer model**

GPT: Generative Pre-trained Transformer (2018–2020)

- **GPT (OpenAI, 2018):** Decoder-only, autoregressive model. Predicted next tokens. Pioneered generative capabilities.



GPT (Generative Pre-trained Transformer) is the **first decoder-only Transformer model**.

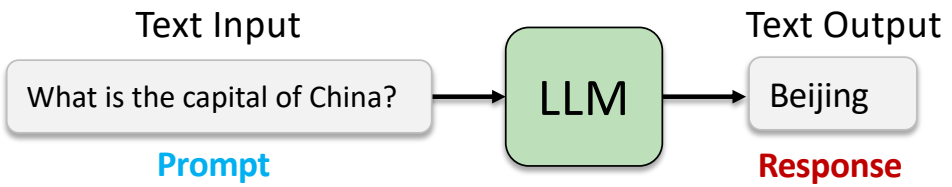
Scaling Era: GPT-2 and GPT-3 (2019–2020)

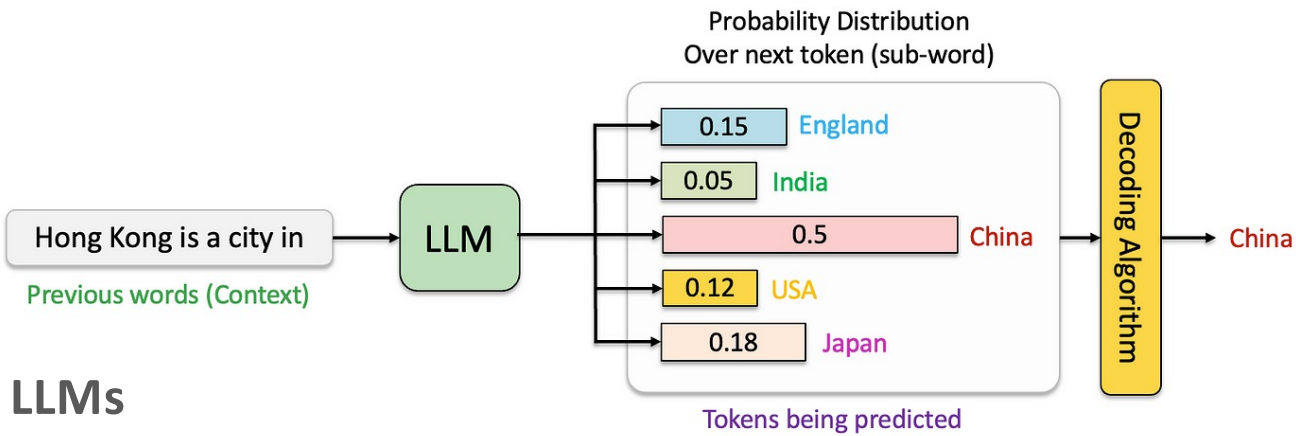
- **GPT-2 (2019):** Demonstrated zero-shot learning with **1.5B parameters**
- **GPT-3 (2020):** Showed few-shot and zero-shot generalization across tasks—writing, coding, reasoning with **175B parameters**

The scaling hypothesis was confirmed: bigger models + more data = better performance.

Large Language Models (LLMs)

- **Text-to-Text Generative models**
  - OpenAI **GPT-3, ChatGPT**
  - Anthropic **Claude**
  - Google **Gemini**
  - Meta **LLaMA-3.1-405B**
  - **…**

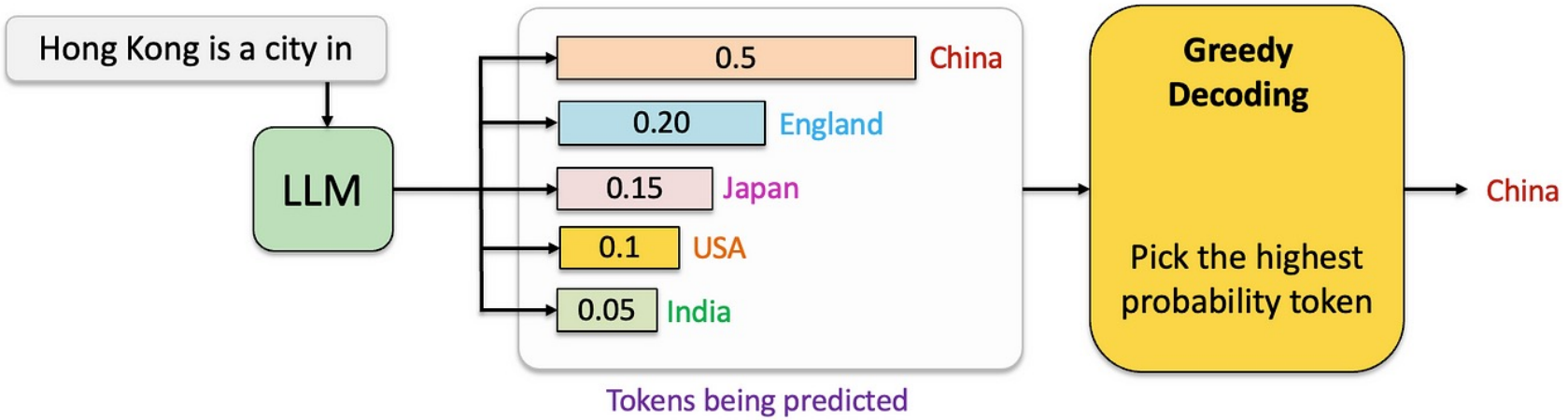




- **Autoregressive Nature of LLMs**
  - **LLMs predict the next token (sub-word) distribution** in a sequence based on the preceding words with prompt and previously generated words.

Autoregressive Language Models

- **Autoregressive Language Modeling** is the task of **predicting the next word**
  - 文字接龙
  - Hong Kong is a city in _____. China?, England?, Japan?, USA?, …

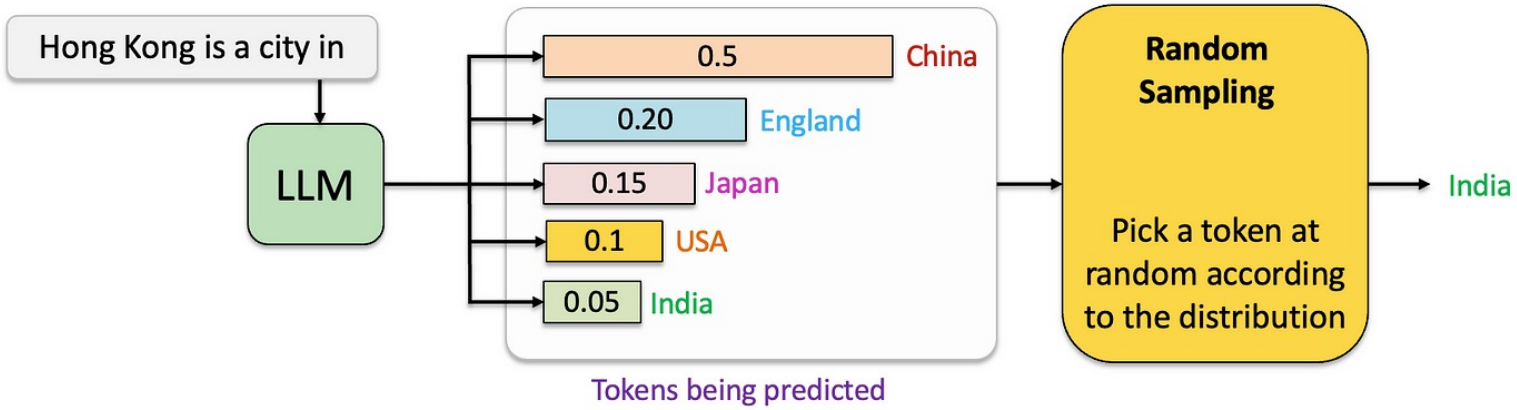


Hong Kong TV Game: GPT Game (文字接龙)

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

LLM Hallucination (幻觉)

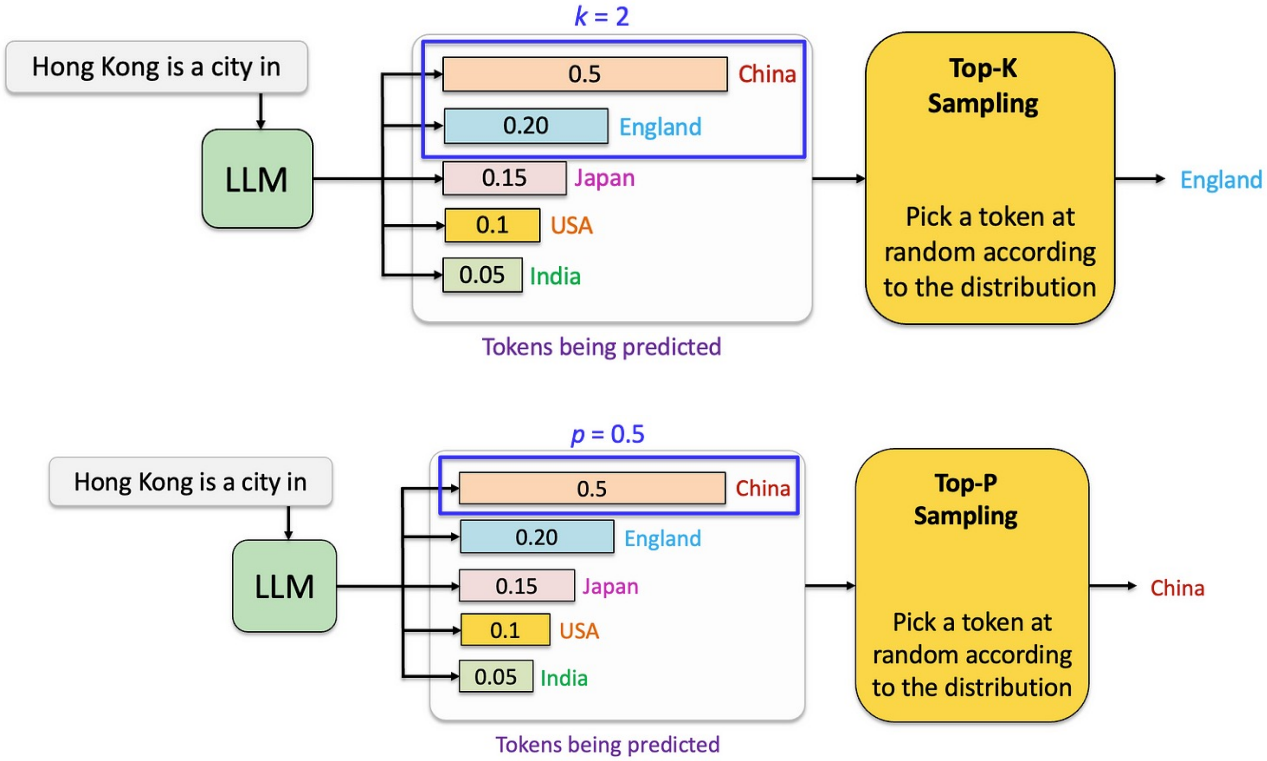
- LLM Hallucination occurs when a LLM **generates confident but factually incorrect or misleading outputs.**

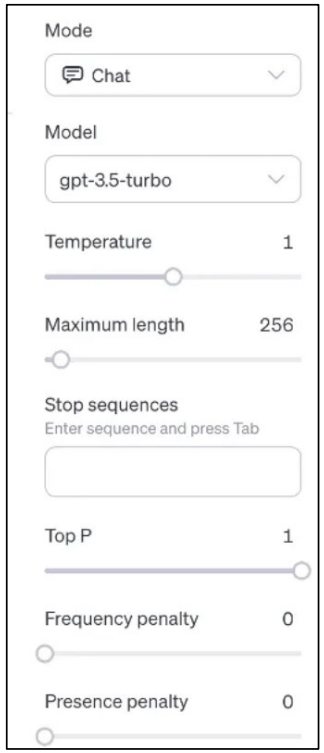


- **User:** Who was the first female president of the United States?
- **LLM:** The first female president of the United States was **Emily Wilson**, who served from 1921 to 1925. She was a pioneering figure in American politics and a champion of women's rights. During her presidency, she oversaw the passage of the 19th Amendment, granting women the right to vote.

Top-K and Top-P Decoding

- To mitigate hallucinations in LLMs, decoding methods such as adjusting temperature, using top-k or nucleus sampling, and implementing beam search can be employed. Additionally, fact-checking and grounding responses with external knowledge sources help ensure accuracy and reliability.





Main Issues of Pre-Trained Only LLMs:

- Despite their power, LLMs like GPT-3 suffered from:
  - **Instruction-following inconsistencies.**

> Example 1: Instruction-Following Issues
> Prompt: "Write a short poem about the ocean, focusing on its calmness, without mentioning any colors."
> **GPT-3 Output:** "The ocean's blue surface sways gently, its tranquil waves whispering peace under the sky."

This response **fails to follow the instruction** by mentioning the color “blue,” despite the explicit constraint, and only partially emphasizes calmness.

  - **Hallucinations (factual errors).**

> Example 2: Factual Inaccuracies
> Prompt: "What is the capital of Australia?"
> **GPT-3 Output**: "The capital of Australia is Sydney."

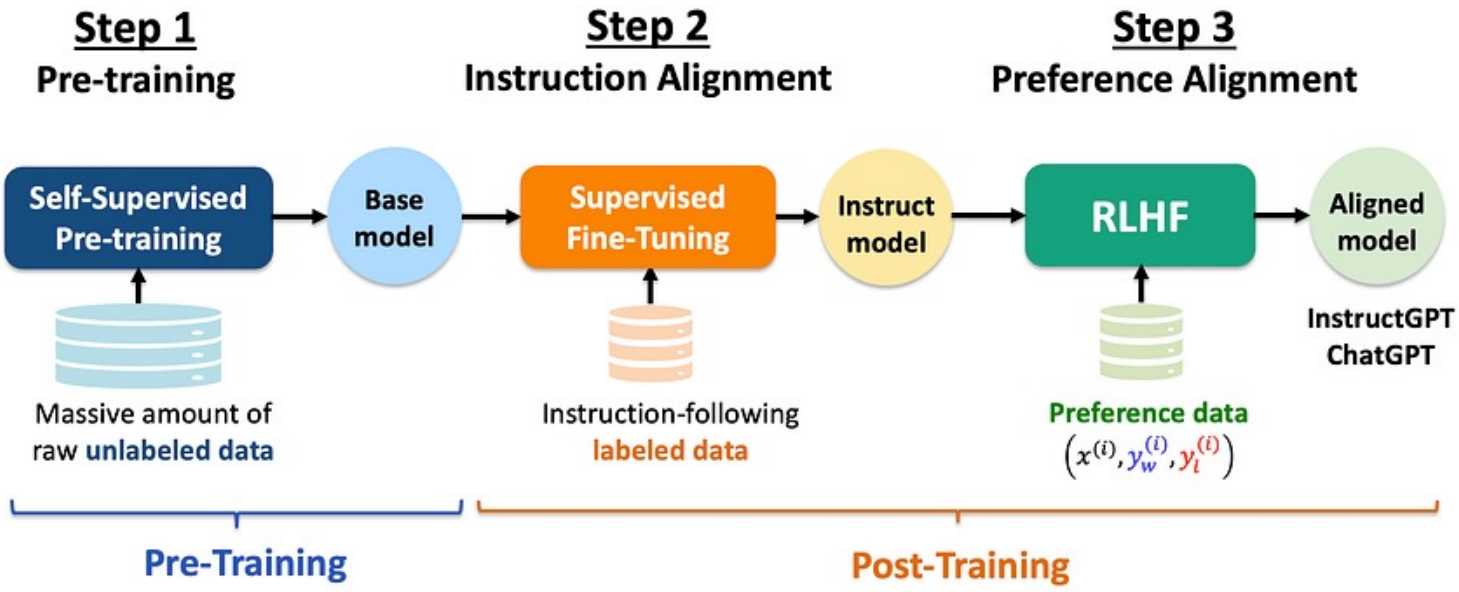
This is incorrect; the capital is **Canberra**, highlighting the model’s tendency to prioritize common associations over facts.

Main Topics of the Course

- **Course Overview**
- **History of AI with Deep Learning**
- **Review of Linear Algebra, Calculus and Probability**
- **Multi-Layer Perceptron (MLP)**
  - Perceptron and MLP
  - Activation function, Loss function
  - Backpropagation and Gradient Descent
  - Regularizations and Optimizations
- **Convolutional Neural Networks (CNNs)**
  - Convolutional Neural Layer
  - LeNet, AlexNet, VGGNet, InceptionNet, ResNet, DenseNet, EfficientNet
  - Computer Vision Applications:
    - Object Detection, Image Segmentation
- **Recurrent Neural Network (RNN)**
  - Introduction to NLP
  - Tokenization and Word Embeddings
  - Vanilla RNN, LSTM, GRU
  - Encoder-Decoder Models and Attention
- **Transformers and Large Language Models (LLMs)**
  - Self-Attention Mechanism
  - Transformers
  - BERT, GPT, GPT-2, GPT-3, T5 and BART
  - LLM Decoding and Prompt Engineering
  - Model Quantization
  - Parameter-Efficient Fine-Tuning (PEFT)
- **Preference Alignment and Multimodality**
  - Preference Alignment: Instruction-Tuning, RLHF, DPO, ORPO
  - Multimodality: CLIP, GPT-4V, LLaVA

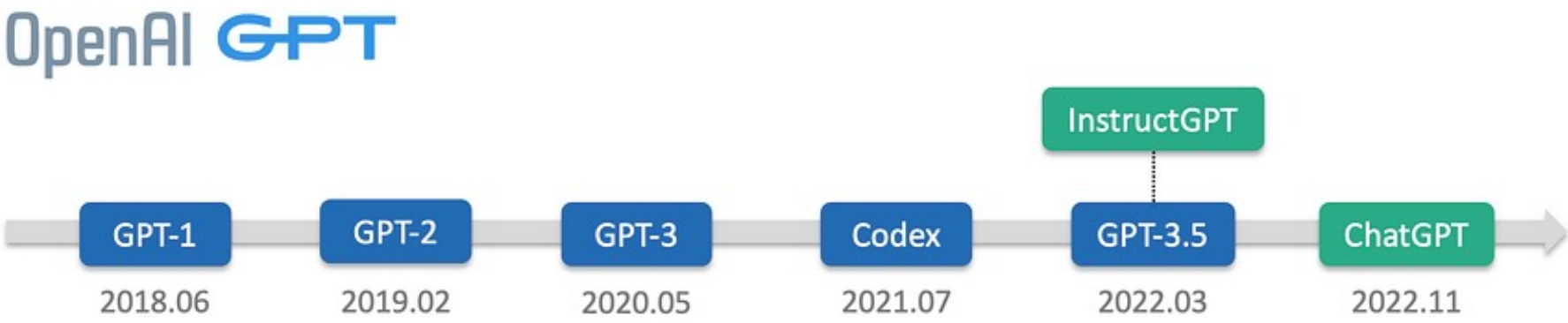
Post-Training Alignment: RLHF and ChatGPT (2022)

- **Supervised Fine-Tuning (SFT):** Training on high-quality instruction-response pairs.
- **Reinforcement Learning from Human Feedback (RLHF):** Training a reward model from human-ranked responses, then fine-tuning the LLM via PPO.



ChatGPT Moment (2022-11)

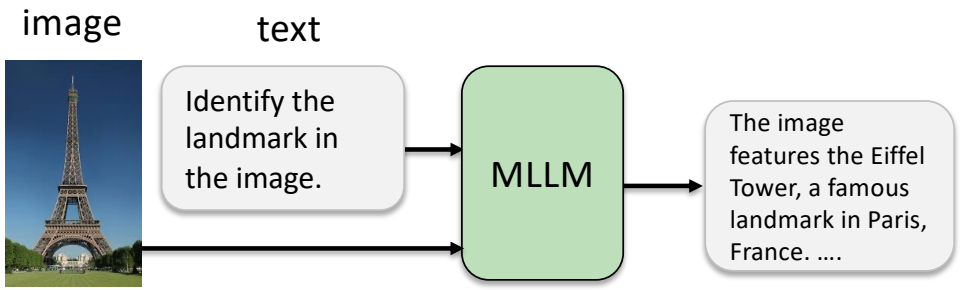
- In November 2022, ChatGPT (based on GPT-3.5) was launched, marking the "ChatGPT moment"—AI went mainstream.
- It is because ChatGPT demonstrated the potential of conversational AI to transform human-machine interactions.



Multimodal Models (2023–2024)

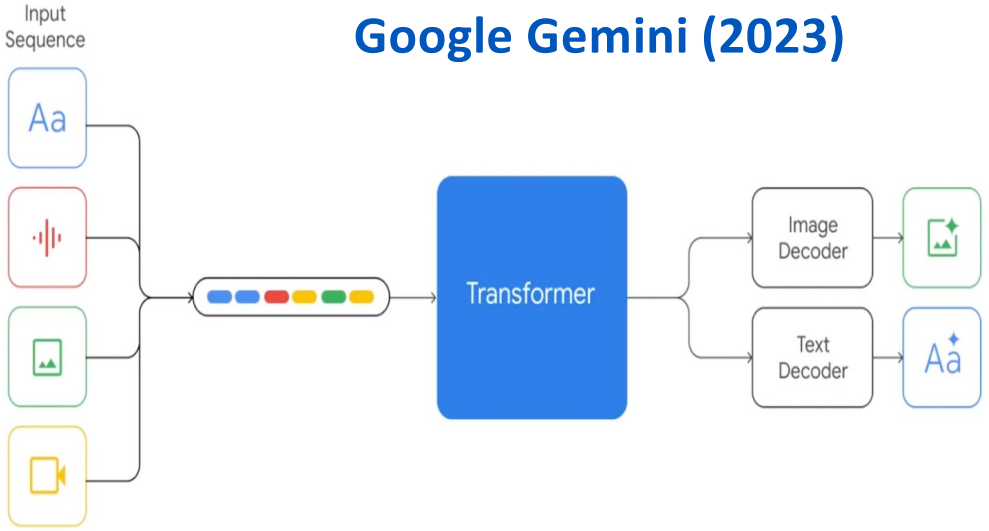
- Multimodal Large Language Models (MLLMs) are advanced AI systems that can process and generate responses based on **multiple types of input**, such as **text, images, and audio**.
- This capability enhances applications in areas like visual question answering, image captioning, and multimedia content creation.

OpenAI GPT-4V (2023)



(Image and Text) to Text Generation

Google Gemini (2023)



OpenAI Multimodal Models

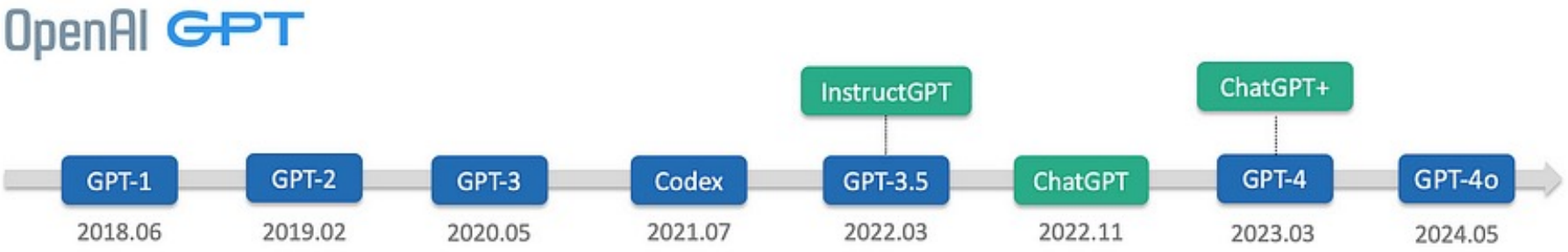
- GPT-4V (2023): Integrated vision and language.
- GPT-4o (2024): Unified text, image, audio, and video processing. Enabled real-time interaction and omni-modal reasoning.

GPT-4 visual input example, Extreme Ironing:

User: What is unusual about this image?

Source: https://www.barnorama.com/wp-content/uploads/2016/12/03-Confusing-Pictures.jpg

GPT-4: The unusual thing about this image is that a man is ironing clothes on an ironing board attached to the roof of a moving taxi.

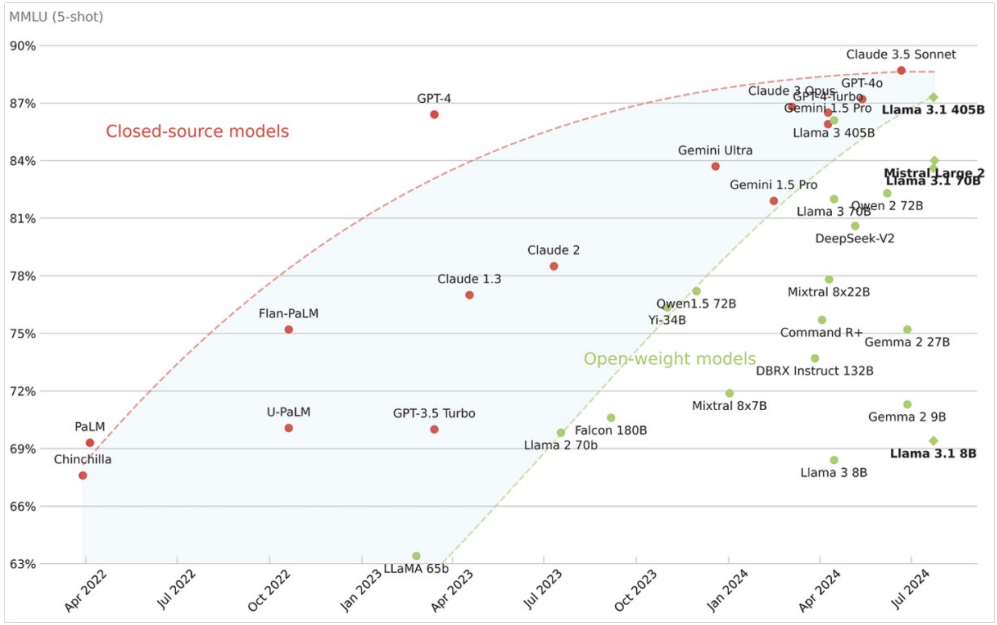


Say Hello to GPT-4o

https://www.youtube.com/watch?v=vgYi3Wr7v_g

Open-Source Movement (2023–2024)

- Meta’s LLaMA, LLaMA2, LLaMA3: Open-weight models.
- Mistral 7B, Mixtral 8x7B: Efficient, high-performance open models.
- Hugging Face: Community-driven innovation with LoRA, PEFT.
- The gap between closed and open models narrowed significantly.

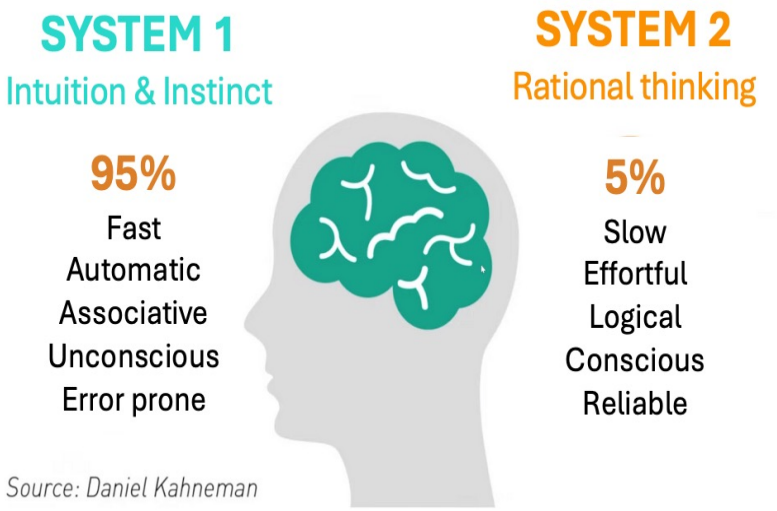


OpenAI's GPT LLM Series

- **GPT** (2018-06): 117M parameters
  - **Pioneered decoder-only transformer architecture** and improved language understanding through text-completion objective pre-training (self-supervised learning)
- **GPT-2** (2019-02): 1.5B parameters
  - Demonstrated language models as unsupervised multitask learners and showcased **zero-shot learning capabilities**
- **GPT-3** (2020-05): 175B parameters.
  - **Introduced few-shot learning with in-context learning** and exhibited remarkable versatility in performing various tasks without task-specific training
- **ChatGPT/GPT-3.5** (2022-11)
  - **Optimized for conversational AI and Preference Alignment using RLHF**
- **GPT-4Vision** (2023-03)
  - **Enhanced multimodal capabilities** and context understanding
- **GPT-4o** (2024-05)
  - A more efficient and optimized version of GPT-4
- **GPT-o1-preview** (2024-09)
  - Designed to **enhance reasoning capabilities**, particularly in complex tasks such as math, science, and coding

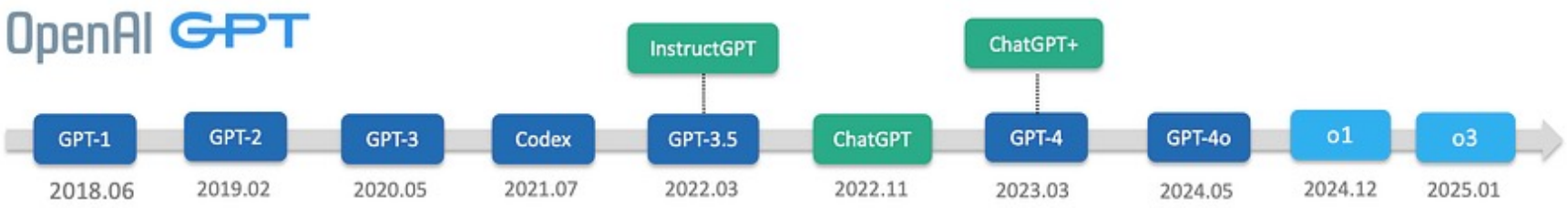
Reasoning Models: System 2 Thinking (2024)

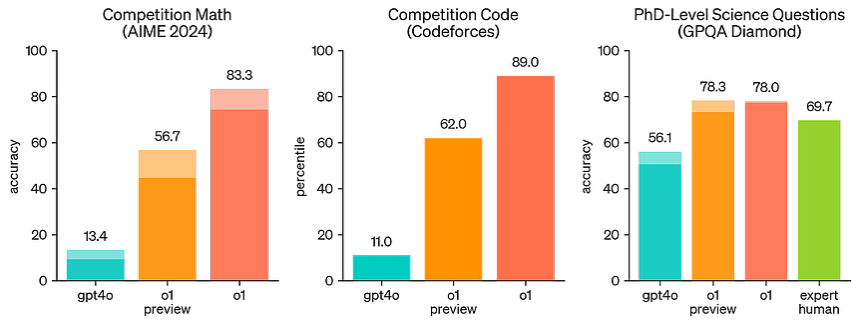
- Inspired by cognitive science's dual-process theory:
  - **System 1: Fast, intuitive.**
  - **System 2: Slow, analytical.**



OpenA1: o1 (2024-12) and o3 (2025-01

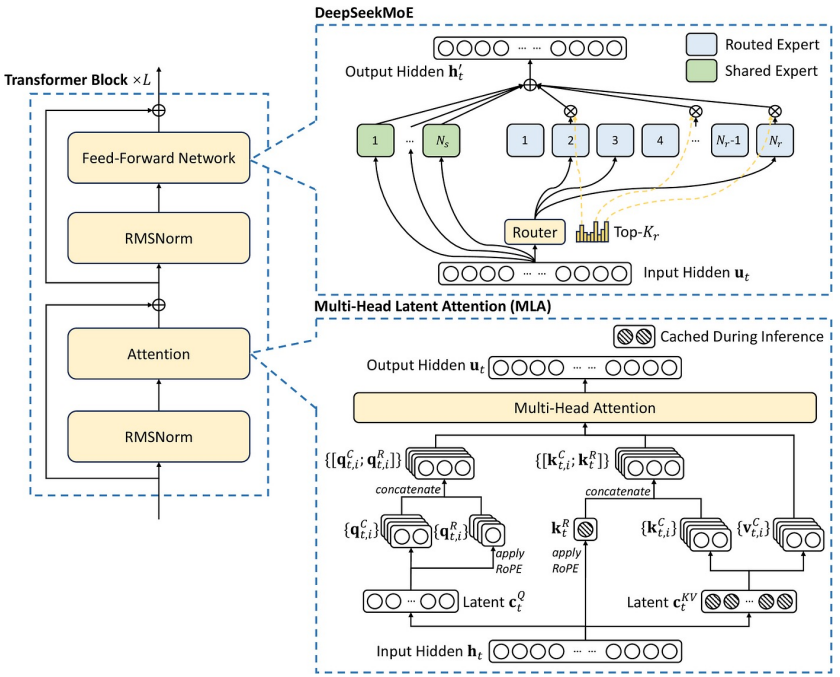
- OpenAI-o1 (2024) introduced **Long Chains of Thought (Long CoT)**, enabling internal reasoning, self-critique, and search-like problem solving.
  - o1-mini: 80% cheaper, strong in coding.
  - o3 (2025): Achieved 87.5% on ARC-AGI, surpassing human-level performance.





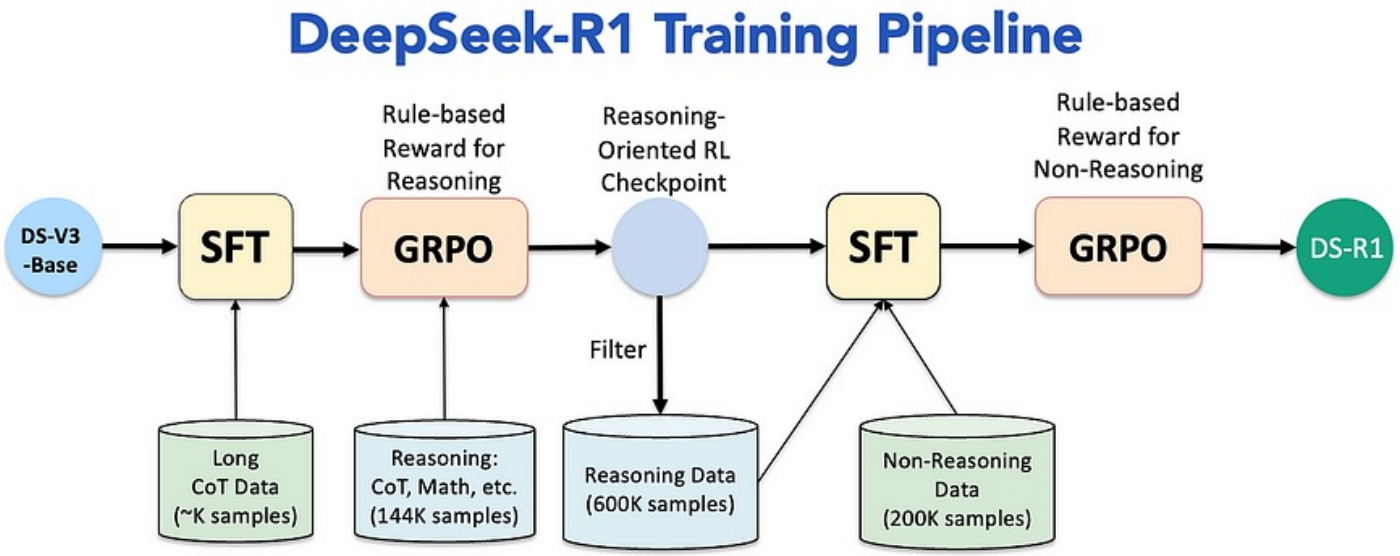
DeepSeek-V3 (2024-12)

- 671B parameters, 37B active (MoE).
- $5.6M training cost (vs. $100M+ for GPT-4).
- Architectural innovations:
  - Multi-Head Latent Attention (MLA): Compressed KV cache.
  - DeepSeekMoE: Shared + routed experts.
  - Multi-Token Prediction: Improved coherence.
- Priced at $2.19/million tokens (1/30th of OpenAI).



DeepSeek-R1 and R1-Zero (2025)

- **R1-Zero**: Trained via Group Relative Policy Optimization (GRPO), skipping SFT. Rule-based RL for reasoning.
- **DeepSeek-R1**: Enhanced with cold-start data and multi-stage RL. Better readability and alignment.



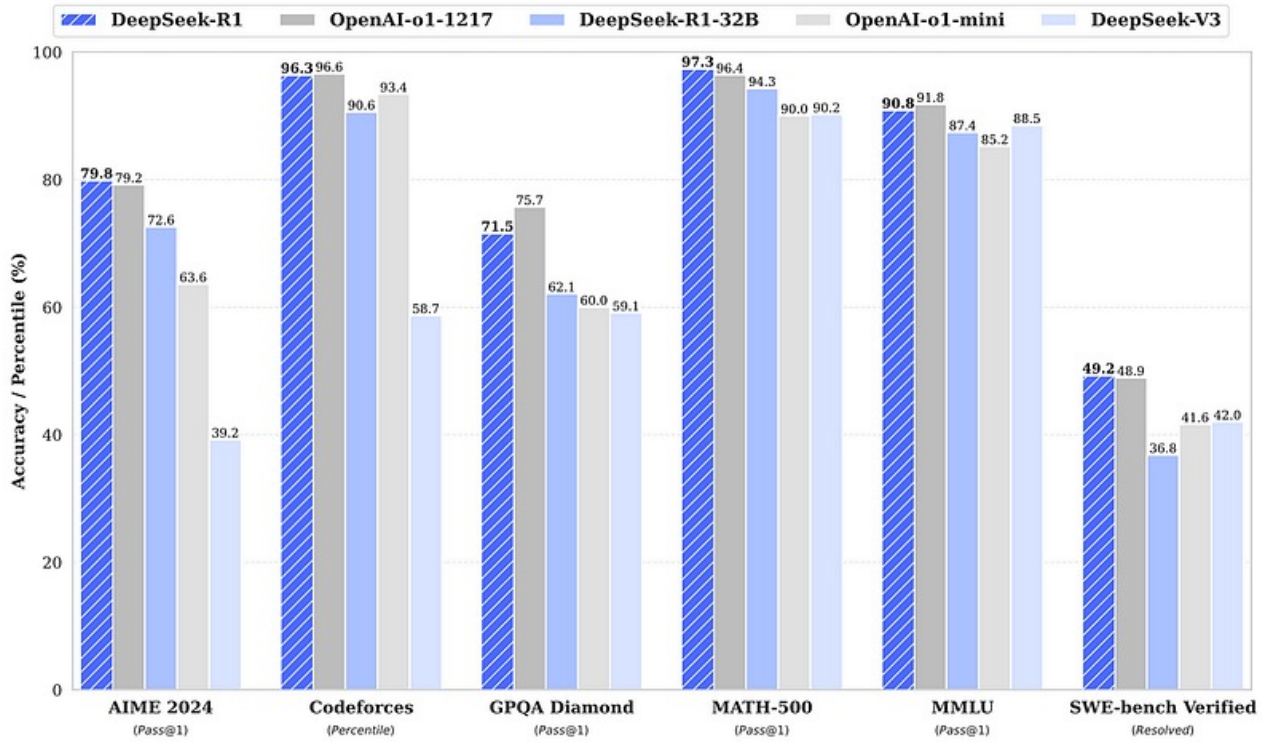
Distilled DeepSeek-R1 (Small Models)

- **Distilled versions**: 1.5B to 70B parameters for local deployment.

Model Name	Base Model	Total Parameters
DeepSeek-R1-Distill-Qwen-1.5B	Qwen2.5-Math-1.5B	1.5 billion
DeepSeek-R1-Distill-Qwen-7B	Qwen2.5-Math-7B	7 billion
DeepSeek-R1-Distill-Llama-8B	Llama-3.1-8B	8 billion
DeepSeek-R1-Distill-Qwen-14B	Qwen2.5-14B	14 billion
DeepSeek-R1-Distill-Qwen-32B	Qwen2.5-32B	32 billion
DeepSeek-R1-Distill-Llama-70B	Llama-3.3-70B-Instruct	70 billion

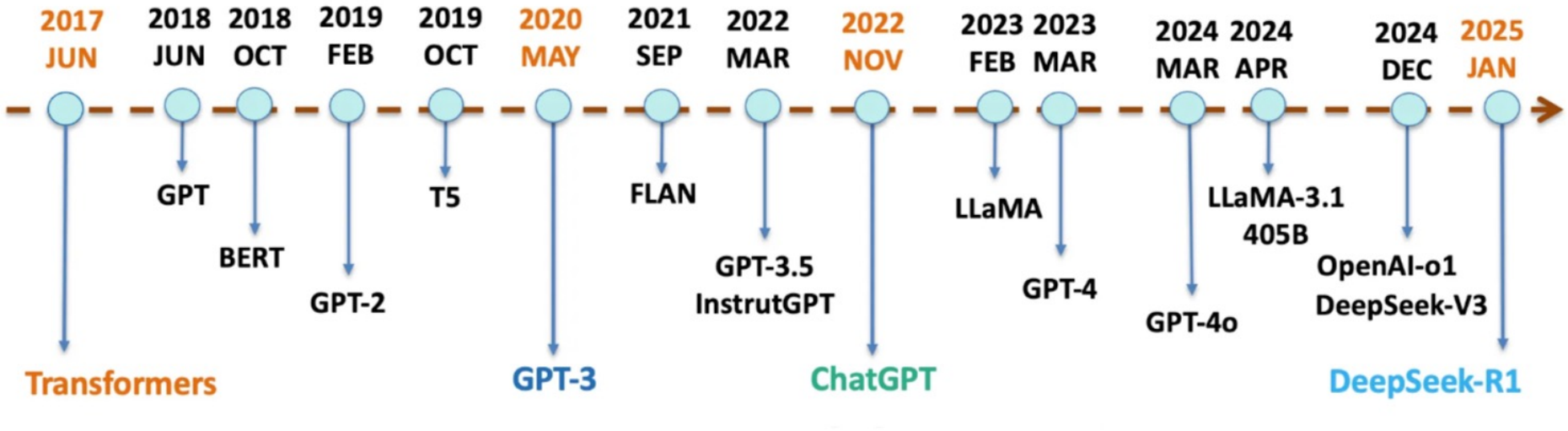
DeepSeek-R1 Performance and Cost

- DeepSeek-R1 delivers strong performance across math, coding, knowledge, and writing benchmarks, **offering significant cost savings—up to 20–50x cheaper than OpenAI's o1**—depending on usage.



DeepSeek Moment

- DeepSeek-R1 is democratizing AI with affordable, high-performance LLMs, now available on major clouds and driving broader adoption.



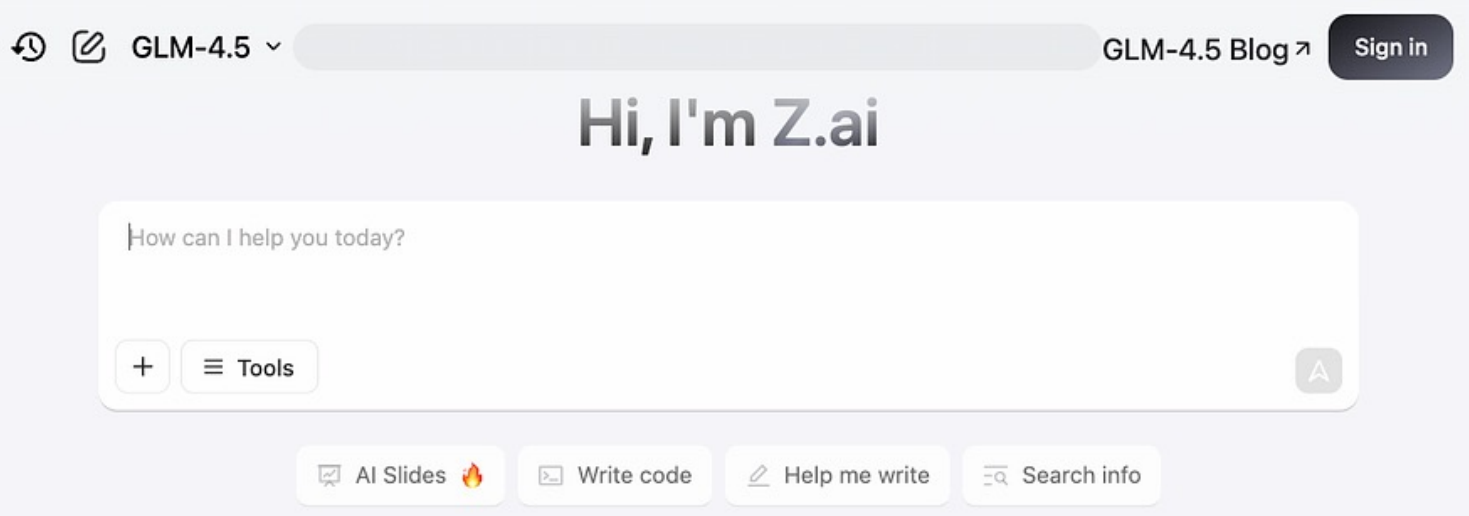
https://medium.com/@lmpo/a-brief-history-of-lmms-from-transformers-2017-to-deepseek-r1-2025-dae75dd3f59a

The Hybrid Reasoning Era (2025)

- Industry-wide adoption of dual-mode systems combining fast responses with deeper analytical capabilities.
  - **Anthropic's Claude 4** (May 2025):
    - Both Claude Opus 4 and Sonnet 4 are hybrid reasoning models.
    - Can produce near-instant answers or engage an "extended thinking" mode for step-by-step reasoning on harder problems.
    - Support interleaved thinking for complex agentic interactions.
  - • **xAI's Grok 3 & 4** (2025):
    - Grok 3's reasoning capabilities (refined through large-scale RL) allow it to "think" for seconds to minutes, correcting errors and exploring alternatives.
    - Features two reasoning modes: "Think" for step-by-step logic and "Big Brain" for heavy-duty problem-solving.
    - Grok 4 provides frontier-level multimodal understanding with advanced reasoning.
  - **Alibaba's Qwen 3** (April 2025): Offers a novel hybrid thinking mode for dynamic reasoning control (thinking mode for step-by-step reasoning and non-thinking mode for quick responses).

Recent Well-Known China's LLMs

- Alibaba **Qwen3-235B**: https://chat.qwen.ai/
- Moonshot AI **Kimi K2**: https://www.kimi.com/
- Z.ai **GLM-4.5** : https://chat.z.ai/



How can AI respond to questions about electronic circuits?

4.2 Problem 4.49

We wish to design the amplifier circuit of Figure (4.2) under the constraint that V_{CC} is fixed. Let the input signal $v_{be} = \hat{V}_{be} \sin \omega t$ where $\hat{V}_{be}$ is the maximum value for acceptable linearity. Show for the design that results in the largest signal at the collector without the BJT leaving the active region, that

$$R_C I_C = \frac{V_{CC} - V_{BE} - \hat{V}_{be}}{1 + \frac{\hat{V}_{be}}{V_T}}$$

and find an expression for the voltage gain obtained. For $V_{CC} = 10$ V, $V_{BE} = 0.7$ V, and $\hat{V}_{be} = 5$ mV, find the dc voltage at the collector, the amplitude of the output voltage signal, and the voltage gain.

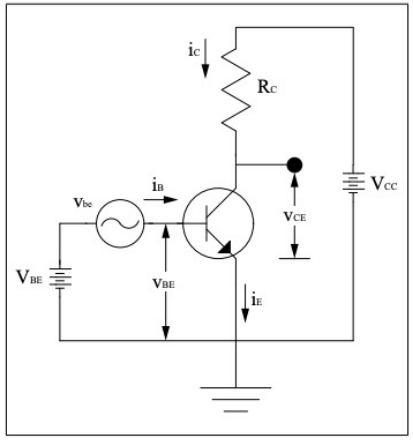


Figure 4.2:

Qwen3-235B-A22B-2507 4:06 PM

Solve the problem in the provided image:

Qwen3-235B-A22B-2507 4:06 PM

Thinking completed · 81,920 tokens budget

To solve the problem, we need to determine the DC voltage at the collector, the amplitude of the output voltage signal, and the voltage gain for the given BJT amplifier circuit under the constraint that V_{CC} is fixed.

Key Steps:

1. **Derive the Expression for $R_C I_C$:**
   - The maximum peak swing at the collector without leaving the active region is given by the formula:

$$R_C I_C = \frac{V_{CC} - V_{BE} - \hat{V}_{be}}{1 + \frac{\hat{V}_{be}}{V_T}}$$

   - Substituting the given values $V_{CC} = 10\,\text{V}$, $V_{BE} = 0.7\,\text{V}$, $\hat{V}_{be} = 5\,\text{mV}$, and $V_T = 26\,\text{mV}$:

$$R_C I_C = \frac{10 - 0.7 - 0.005}{1 + \frac{0.005}{0.026}} \approx \frac{9.295}{1.1923} \approx 7.8\,\text{V}$$

2. **Calculate the DC Voltage at the Collector:**
   - The DC collector voltage is:

$$V_C = V_{CC} - R_C I_C = 10 - 7.8 = 2.2\,\text{V}$$

3. **Determine the Amplitude of the Output Voltage Signal:**
   - The voltage gain A_v is given by:

$$A_v = -g_m R_C$$

   - where $g_m = \frac{I_C}{V_T}$. Using $R_C I_C = 7.8\,\text{V}$:

$$g_m = \frac{7.8}{R_C V_T}$$

   - The amplitude of the output voltage signal is:

$$\text{Amplitude} = |A_v| \cdot \hat{V}_{be} = \left(\frac{7.8}{V_T}\right) \cdot 5\,\text{mV} \approx \left(\frac{7.8}{0.026}\right) \cdot 0.005 = 300 \cdot 0.005 = 1.5\,\text{V}$$

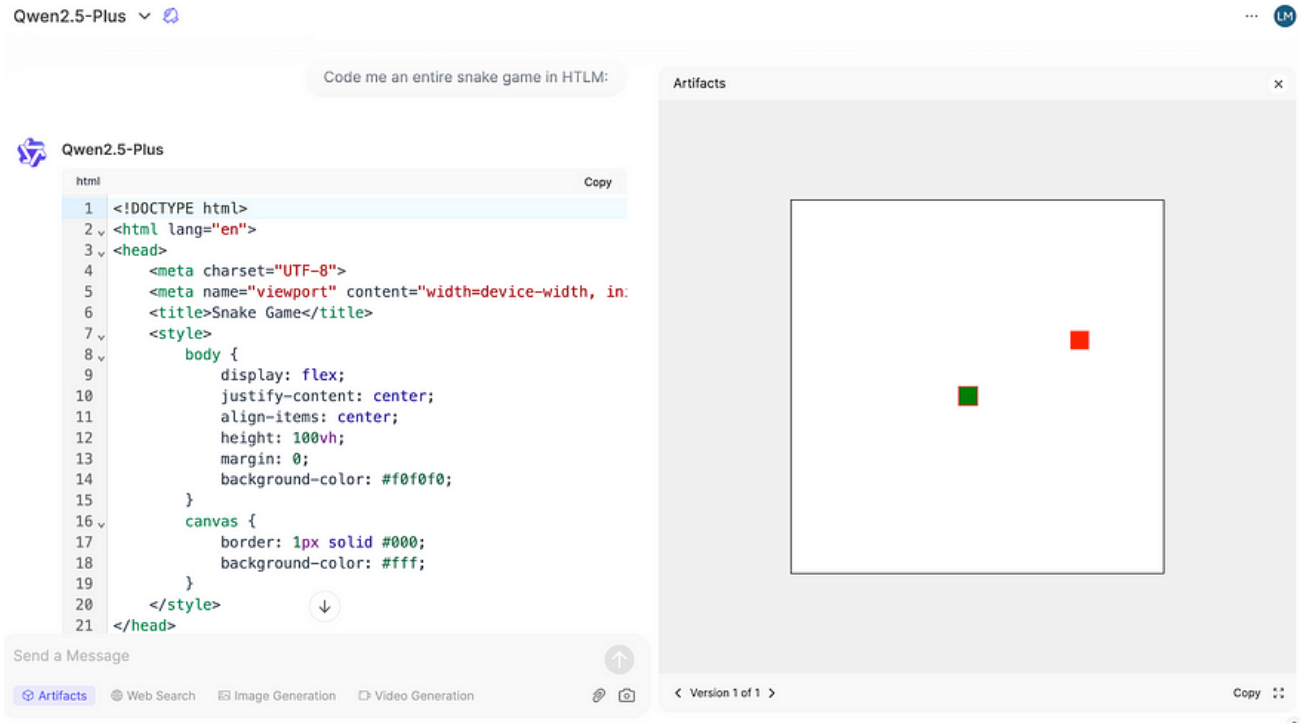
4. **Calculate the Voltage Gain:**
   - The voltage gain is:

$$A_v = -\frac{7.8}{V_T} \approx -\frac{7.8}{0.026} \approx -300$$

Final Answers:

- The dc voltage at the collector is $\boxed{2.2}$ V.
- The amplitude of the output voltage signal is $\boxed{1.5}$ V.
- The voltage gain is $\boxed{-300}$.

AI Code Generation: Vibe Coding

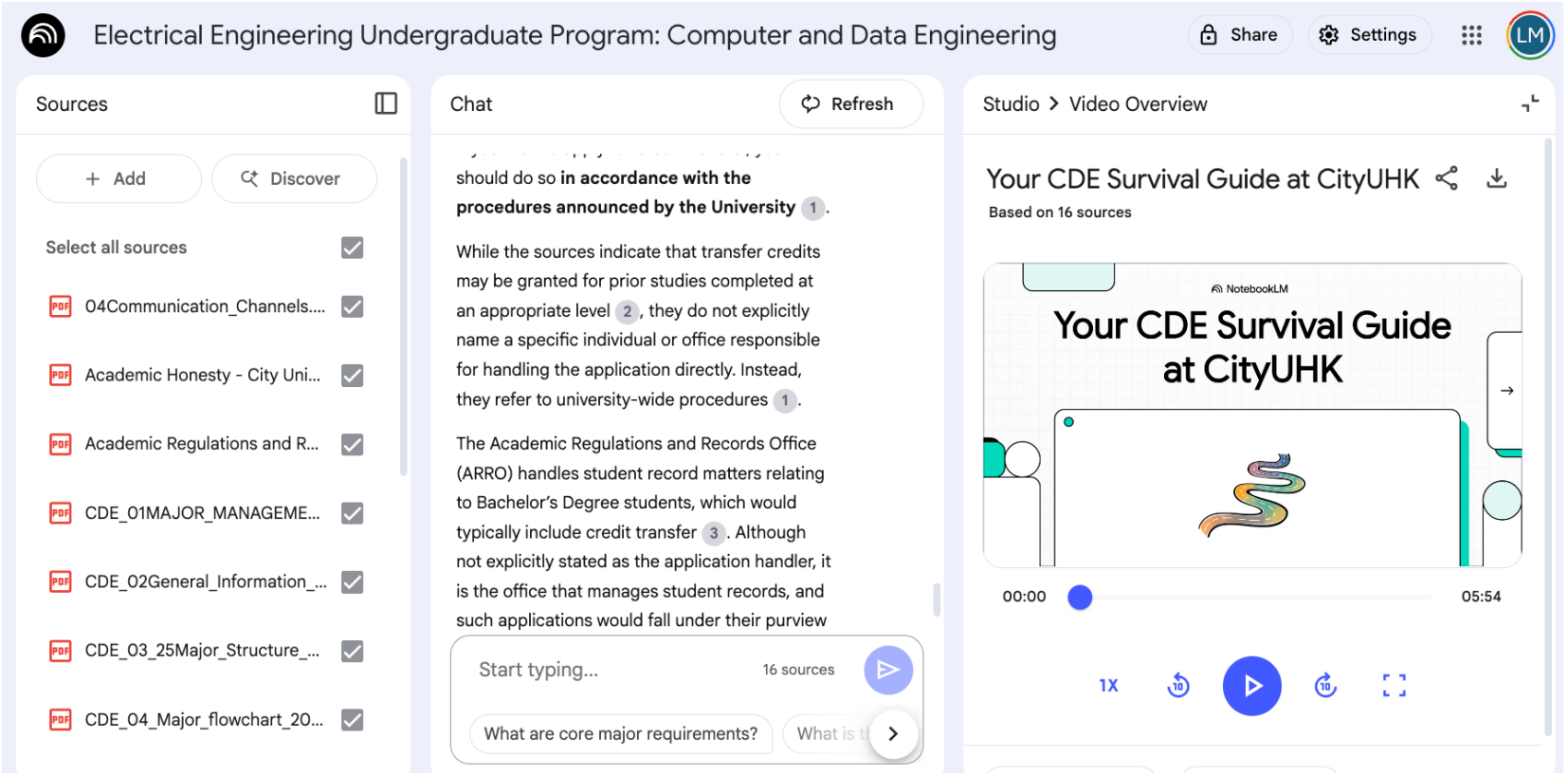


Vibe coding, introduced by Andrej Karpathy in February 2025, is a programming technique where developers describe problems in natural language, and large language models (LLMs) generate the corresponding code.

Programmers focus on guiding, testing, and refining the AI-produced solutions rather than writing code manually.

https://medium.com/@lmpo/qwen-lm-platform-your-gateway-to-advanced-llms-5f8786cf6829

Google's NotebookLM

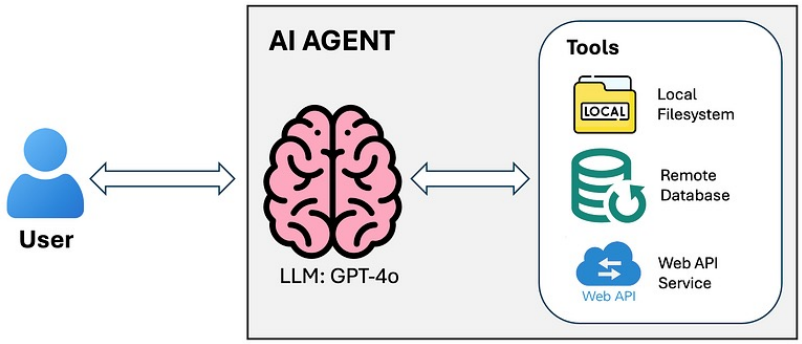


LLM-based Agentic AI

- Shift from passive question-answering to proactive problem-solving
  Autonomous agents plan, execute complex tasks, and interact with external systems
- Notable examples:
  - **Manus** (Monica.im, March 2025): Turns user thoughts into actions for web-based tasks
  - **Genspark** (April 2025): No-code super agent platform for creating AI tools
  - **ChatGPT Agent** (OpenAI, July 2025): Selects skills to complete
  - tasks in a simulated environment

Foundation of Agentic Systems

- **Tool Use and Function Calling**
  - LLMs use external tools, APIs, and services (e.g., web search, code execution, database queries)
- **Planning and Task Decomposition**
  - Break complex objectives into sub-tasks
  - Create and adapt execution plans dynamically
- **Memory and Context Management**
  - Maintain long-term memory and track task progress
  - Manage extensive context across interactions

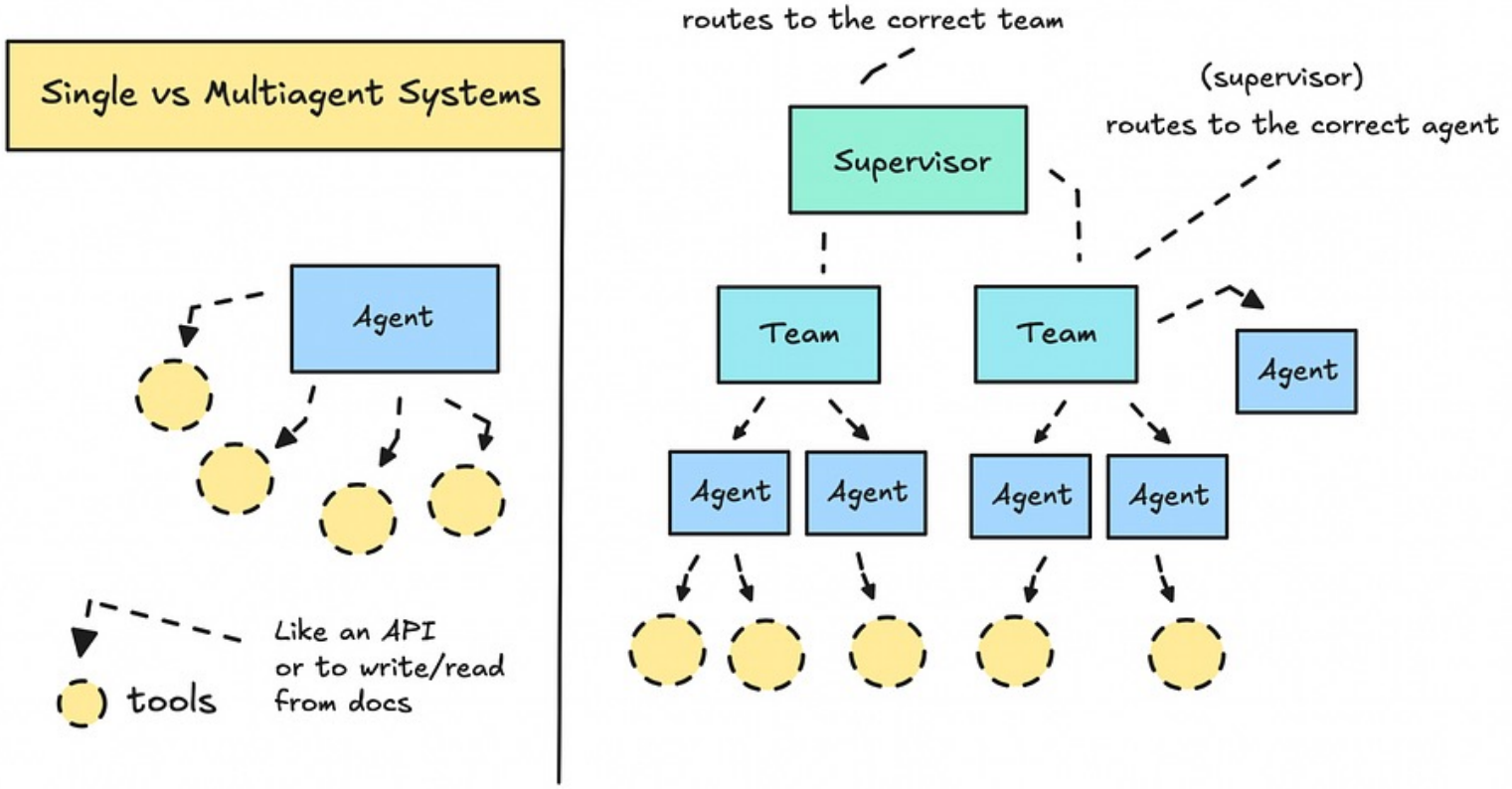


Multi-Agent Systems

Core Concept

- Collaborative AI Teams
  - Multiple specialized agents with distinct roles
  - Communicate and coordinate for complex problem-solving
- Human-AI Collaboration
  - Frameworks for seamless human-AI interaction
  - Clear division of responsibilities and transparent communication
- Agent Orchestration
  - Manage multiple agents, resolve conflicts, ensure coherent behavior

Single vs multiagent workflows



Conclusion – The Future of AI

- The journey of AI from the **McCulloch-Pitts neuron** (1943) to the **sophisticated agentic AI systems** of 2025 is a remarkable trajectory of innovation.
- Each milestone (Perceptron, backpropagation, CNNs, RNNs, Transformers, LLMs) has **expanded AI's capabilities**.
- The recent shift toward **reasoning-focused models, multimodal understanding, and agentic AI** underscores a future where intelligent systems not only mimic human cognition but also proactively solve complex problems.
- AI continues to evolve rapidly, with ongoing research focusing on improved planning, tool integration, enhanced learning, and broader autonomy.

15-Minute Break

- During the break, **students are highly recommended to start forming the group project team.**
- More information about the Group projects will be provided and discussed in the coming few weeks.