

Artificial Intelligence: Past, Present, and Future

Radford Neal

radfordneal@gmail.com

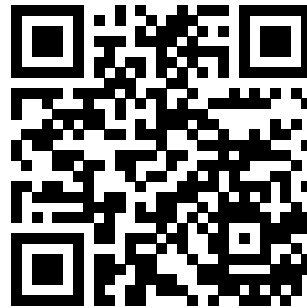
Lecture 1: The pace of AI progress

Lecture 2: Computation and intelligence, artificial and natural

Lecture 3: How modern AI works

Lecture 4: Can an AI think, feel, be conscious? How can we know?

Lecture 5: Benefits and dangers of AI, today and tomorrow



<https://glizen.com/radfordneal/ai-lectures/>

Originally presented at The Abelard School, Toronto, Spring 2025

Visions of AI at the Dawn of Computing

Soon after computers were built — even before — people envisioned intelligent machines.

1879. George Eliot (Mary Ann Evans), *Impressions of Theophrastus Such*:

... when I am told ... of a machine for drawing the right conclusion, which will doubtless by-and-by be improved into an automaton for finding true premisses ... I exclaim — “Am I already in the shadow of the Coming Race? and will the creatures who are to transcend and finally supersede us be steely organisms, giving out the effluvia of the laboratory, and performing with infallible exactness more than everything that we have performed with a slovenly approximateness and self-defeating inaccuracy?”

1950. Alan Turing, *Computing Machinery and Intelligence*:

I believe that in about fifty years' time it will be possible to programme computers, with a storage capacity of 10^9 [bits] ... the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted.

1965. I. J. Good, *Speculations Concerning the First Ultraintelligent Machine*:

It is more probable than not that, within the twentieth century, an ultraintelligent machine will be built and that it will be the last invention that man need make, since it will lead to an “intelligence explosion”. This will transform society in an unimaginable way.

AI Before Deep Learning

Two approaches (here a bit caricatured), both going back to 1960s and earlier.

Symbolic / Logical AI: (Sometimes known as “Good Old-Fashioned AI” (GOFAI))

- Represent facts about the world as logical propositions.
- Use a general theorem prover to derive consequences of known facts.
- Plan what action to take by proving a theorem about what it is optimal to do.
- **Later...** Probabilistic rather than logical inference, but still hand-crafted to capture the structure of reasoning.

Neural networks: (Also known as “connectionism”)

- Inspired by networks of neurons in the brain — but not all that close...
- Strengths of connections between “neurons” are learned from experience.
- Higher-level cognition emerges from such lower-level processes.

Representative issue: Does grammatical use of language require following symbolic rules, or can it emerge from the behaviour of a network of neurons?

The Rise of Deep Learning using GPUs — AlexNet

In 2012, a U of T group (Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton) obtained stunningly good results for image classification. They used:

- An existing large data set of 1.2 million images, each manually annotated with one of 1000 labels.
- A large, deep neural network, with 650,000 neurons, and with 60 million parameters determining connection weights. Some connections implement “convolutions”, that average over local patches of the image.
- GPUs (Graphical Processing Units) for computation.
- Several incremental but crucial innovations in how the network was designed and in how training was done.



Probabilities assigned to top 5 classes for an image

Lessons: More data, Bigger network, Lots of compute power \Rightarrow Better results.

Trial and error will get you far, even absent theoretical understanding.



Learning from Synthetic Data — AlphaGo

Go is a strategy game played for thousands of years in China, Korea, and Japan. “Black” and “White” players alternate placing “stones” on a grid, aiming to surround territory.

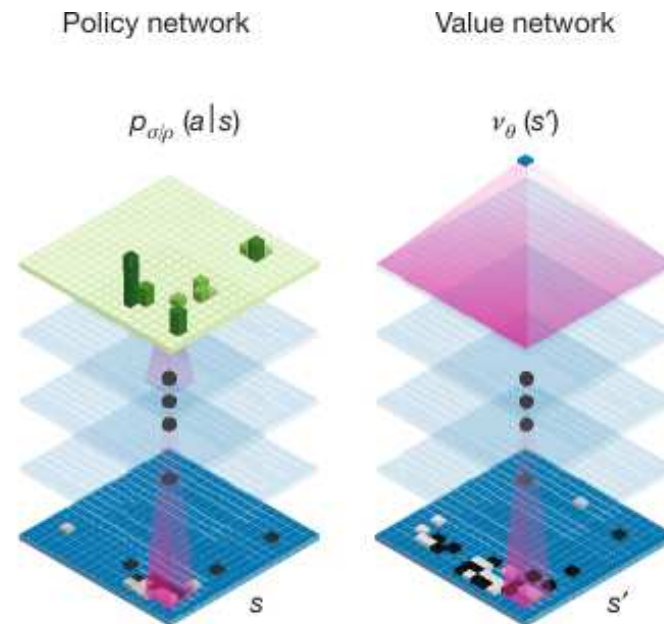
Go was considered very hard for computers to play well. In 2016, DeepMind created AlphaGo, which played Go at the level of top professional Go players.

They trained a “policy” neural network to predict moves of expert human players. Sampling predictions of this network produced quite strong play. They refined this network to better prefer moves leading to wins when playing against earlier versions of itself.

They then trained a “value” network to predict the outcome of self-play games when both sides chose moves using the policy network. Final moves are chosen considering many sequences of possible future moves, guided by the policy network, assessed by the value network.

A later version, AlphaGoZero, no longer used data from human players, learning entirely by self-play. It is much stronger than any human player.

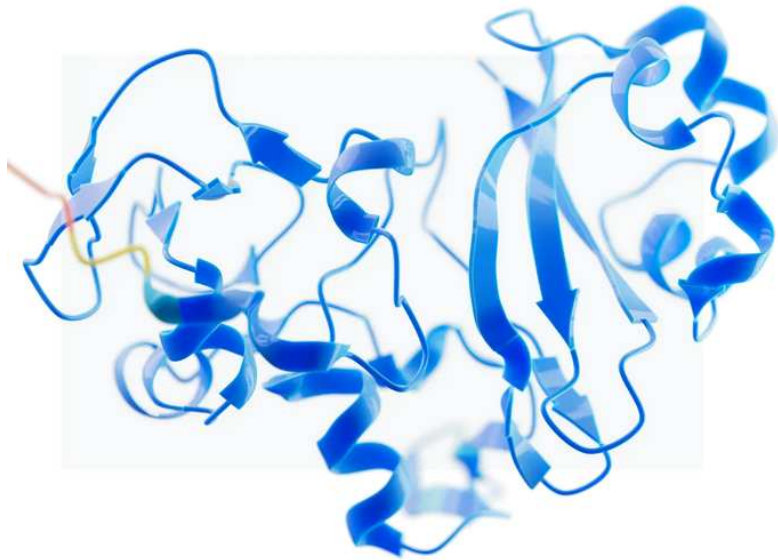
Lesson: Being able to generate huge amounts of *synthetic* data can help learning a lot.



Impressive Narrow AI — AlphaFold

It's relatively easy to find the DNA sequence of a gene, and from that, the sequence of amino acids in the protein made by that gene. But what does this protein do?

That depends on how the protein folds into a three-dimensional shape.



Predicting this shape from the sequence of amino acids has long been a dream of biologists.

Successive models from DeepMind — AlphaFold (2019), AlphaFold 2 (2020), and AlphaFold 3 (2024) — have had astonishing success at predicting protein structure, and later the structure of complexes of multiple interacting proteins (and also RNA and other molecules).

The AlphaFold models use deep neural networks, designed using lots of specialized biochemical knowledge, trained using data on known protein structures, and data on evolution of proteins.

AlphaFold is an example of *narrow* AI: A useful *tool* for solving certain sorts of problems.



Large Language Models (LLMs) — GPT-2

A breakthrough in *general purpose* AI came with OpenAI’s GPT-2 LLM, released in 2019.

GPT-2 and other LLMs are trained to predict, in a probabilistic way, what text follows previous text:

I think I saw {
you with someone
Rob Gronkowski go down on a play
the figure in a newspaper or something like that
a clear play that would make Tyrod even better
Fox, uh-huh

These are five random continuations by GPT-2 of “I think I saw ”.

GPT-2 learned to do this when trained on 8 million web documents totaling 40 Gigabytes of text. Its neural network had 1.5 billion parameters (“weights”).

Predicting text might seem to be of limited usefulness. But a model that can do this well must have learned a lot about both language and the world. With suitable prompting, it can perform many tasks.



Prompting GPT-2 to Answer Questions

To produce coherent text, GPT-2 must know lots of things. The trick is to get it to tell them to you by “prompting” it the right way.

One can try just asking (prompt in blue, GPT-2’s continuation in black):

What is the capital of Texas?

Your guess is as good as mine. Probably not Austin. Since when is Austin foreign to the state’s capital?" (Added for illustrative purposes only.)

Oh, and how about the state flag?

Why did it take Austin so long to get their very own state flag? Is this not the fundamental question?

Maybe giving question-answer pairs as examples will help:

Where is Calgary? In Alberta.

Where is Winnipeg? In Manitoba.

Where is Halifax? In Nova Scotia.

Where is Toronto? In Ontario. (Winnipeg’s population has nearly doubled since 2011, owing to an influx of refugees)

Where is Ottawa? In Canada.

And, thus, they deserve our complete and undivided attention!

As you can see, GPT-2 can get a bit distracted from what you were trying to talk about.

Prompting GPT-2 to Form Questions

A prompt establishing a task, which GPT-2 then performs in its continuation (in black):

Let's turn statements into questions!

Albert grows tomatoes -> Does Albert grow tomatoes?

Strawberries smell good -> Do strawberries smell good?

Water flows downhill -> Does water flow downhill?

Fire spreads quickly -> Does fire spread quickly?

Cats play outside. -> Do cats play outside?

I like them. -> Do I like them?

Players take turns -> Do players takes turns?

Nobody goes there. -> Does nobody go there?

John reads comics. -> Does John read comics?

Dogs run fast -> Do dogs run fast?

Some leave early -> Do some leave early?

Diamonds cost money -> Do diamonds cost money?

Broccoli tastes bad -> Does broccoli taste bad?

She hates me. -> Does she hate me?

Meteors fall quickly. -> Do meteors fall quickly?

Philosophers read Socrates -> Do philosophers read Socrates?

Mary eats apples. -> Does Mary eat apples?

Many birds fly around the same place. -> Do many birds fly around the same place?

Some horses have brown legs. -> Do some horses have brown legs?

Further Continuation...

Here's the further continuation after the previous slide:

Water flows downhill -> Does water flow downhill?

Who came? -> Do you like me?

See also: My note about this topic on Quora . Write a comment below and let us know what questions you think are important to ask.

Copyright© 2017 John Scudamore And The Board Of Trustees of the University Of Chicago | Campus Discussion

GPT-2 randomly picks a continuation from what it thinks are the plausible ones.

Since its idea of what's plausible isn't all that accurate, it eventually breaks the pattern in the prompt, and then diverges into random text that might plausibly be found on a web page.

Prompting GPT-2 to Tell a Story

Violins in the Valley

K. E. Evans

John grew up in a musical household. His mother sang in the church choir - her father a flautist - and his father, Captain Lewis Evans of the Royal Norfolk Regiment, was a tenor for Folkestone's "Powerhouse." John's first lady allure (so frequently a factor in the triumph or failure of romantic matches) may well have been the tall size of his wife and the white dress she wore. Also in the family was Anthony Madera, now an actor in the East End.

John started public school at Bishop Leete's School near Bishop's Green, which gave service to the booming merchant-trade of Norfolk Island and the nearby fishing-fleet that bound the sea for India. Upon graduating from college he chose a career with one of King George I's ships, and after three years spent as a statesman and administrator in other ships and in the Admiralty, was accepted into private trade to pay for the education at Bishop's Green. It was then known as Seaforth, before being renamed Barnstaple in the Fifties.

John studied music, working with his father, and joining an organ at Barnstaple Abbey, playing behind the organs of other clergymen at dinners. One note that he had certainly recorded so as to be able to play it was "The Middle C and D." Bets fell upon the organist who played them for The Black Watch Conference.

The very popular local flute player Harry Smith then introduced John to the bass

[CONTINUES]

Larger Large Language Models

GPT-2 was the first LLM good enough to be fun, but it wasn't of much practical use.

It did show that training to predict text implicitly trains for many tasks (not narrow AI).

Bigger and better LLMs have since been created by several companies:

OpenAI — ChatGPT-3.5 (2022) to ChatGPT-4.5 (2025)

Anthropic — Claude (2023) to Claude 3.7 (2025)

Google — Gemini (2023) to Gemini 2.5 (2025) & Gemma (2024) to Gemma 3 (2025)

Meta — Llama (2023) to Llama 3.1 (2024)

xAI — Grok 1 (2023) to Grok 3 beta (2025)

DeepSeek — DeepSeek (2023) to DeepSeek-V3-0324 (2025)

Compare GPT-2 of 2019 with DeepSeek V3 of 2024 (Llama 3.1 is similar):

More parameters (and more layers of neurons):

GPT-2: 1.8 billion parameters \implies DeepSeek V3: 671 billion parameters

Trained using more data:

GPT-2: 40 Gigabytes of text \implies DeepSeek V3: 50,000 Gigabytes (text, code)

The new LLMs are similar to GPT-2 — some innovations, but not fundamental ones.

Prompting Llama 3.1 (Base Model) to Answer Questions

Let's try directly asking:

What is the capital of Texas?

By looking at this Red States w/ Borders you probably can't tell. But when your looking at a wall map with State capitals labeled, it will be easier to see that the capital of Texas and its largest city is Houston. Use your fingers or sticky notes to label the different state capitals found on this Wall Map of United States (With borders).

Let's try again (another random completion):

What is the capital of Texas?

That's right it's Austin. Now that you clicked, why not stay and enjoy what Austin has to offer? There's the Zoo for lovers of animals, Barton Springs and Zilker Park for those who love the great outdoors, and the Museum of Art for artistic tourists. Enjoy the city with all your senses and this white t-shirt.

Posted in T Shirts By CityTagged Austin, TexasBookmark the permalink.

More coherent than GPT-2, but still has two problems: Wandering away from the topic, and giving false information.

These issues result from the goal of mimicking web text. Web pages *can* talk about lots of things, and they *can* have false information. Llama 3.1 is doing what it was trained to do.

Story by Llama 3.1 (Base Model)

Violins in the Valley

K. E. Evans

John grew up in a musical household. His mother sang in the church choir and studied voice with an opera singer named Helen Von Lane, whose summer home was at Lake Michigan near John's grandparents. Another influence on him was his uncle Roger Davies who played piano professionally in New York City. When being driven around Manhattan as a child, John would whine out the window of their rattling hot Cadillac that he wanted to hear Roger Davies's piano playing.

It was only one example of how music impinged on ordinary life in young John's world. Grandma always reminded her children to tell John not to talk so much when they were watching the evening orchestra concerts broadcast at Christmas time from Vienna, Austria. When he came into a room where TV cartoons or quiz shows featuring Arthur Godfrey or Jack Benny could be heard then a cartoon character would break out in song - Arthur and his ukulele, or Frank Sinatra if one of John's teenaged sisters had control over the dial.

And now at age eight it seemed time for him to learn an instrument too. His older 10-year-old cousin Roger, Jr., was starting to play flute but that did not seem like a serious enough task - generally in cartoons an unlikely character like a cat with fingers waggled wildly would pick up the shiny metal and do some trills before using it as just one more item in a slapstick routine. Their sister Mary at age 12 was attracting attention for her nimble piano playing, which she'd begun

[CONTINUES]

Getting Llama 3.1 (Base Model) to Write Python Code

Let's try to prompt with the header and documentation for a simple function:

```
def all_positive (a):  
    ''' Returns True if all elements of list a are positive, False otherwise.'''  
  
    return min(a) > 0
```

A correct and concise solution! Let's see what happens in another try:

```
def all_positive (a):  
    ''' Returns True if all elements of list a are positive, False otherwise.'''  
  
    # YOUR CODE HERE  
    return
```

```
# Do not modify the code below:  
print("-----Test 1-----")  
print("Correct Answer: True")  
print("Your Answer      : " + str(all_positive( [1,2,3] ) ) )  
  
print("-----Test 2-----")  
print("Correct Answer: False")  
print("Your Answer      : " + str(all_positive([0,1,2]) ) )
```

Rather than complete the function, it wrote a question for a programming assignment!

More Python with Llama 3.1 (Base Model)

Let's try a slightly harder problem, again prompting with header and documentation:

```
def three_equal (a):
    ''' Returns True if list a has three consecutive equal elements,
        and False otherwise. '''

    for i in range(len(a)-2):
        if a[i] == a[i+1] and a[i+3]:
            return True
    return False

assert three_equal([42, 63, 17, 9999, -17]) == False
assert three_equal(["donkey", "donkey"]) == True
```

Close, but not actually correct. The condition “`a[i] == a[i+1] and a[i+3]`” should be “`a[i] == a[i+1] and a[i] == a[i+2]`”. The second test case is also wrong.

Much better than GPT-2, though, which spits out gobs of ridiculous Python.

Chat Models — LLMs for Questions / Coding / Conversations

Training a model just to “predict what comes next” can give amazing results, but it’s hard to get such a *base* model to do useful things.

So it’s common to train the base model further, specifically forcing it to do well at answering questions or performing requested tasks. It’s also common to train the model to refuse to answer questions like “How can I make fentanyl?”

After this training, the model no longer just predicts the next bit of text — it also tries to be helpful and safe — but its training to predict is still *most* of what’s driving it.

The result of this additional training is called a *chat* or *instruct* model. These are the models that Anthropic, OpenAI, Google, Xai, etc. make available to the public.

Questions for Chat Models

Llama 3.1 (instruct model, with a “system prompt” encouraging brevity):

When did distributors stop being used in cars manufactured in North America?

Distributors were largely phased out in the late 1990s and early 2000s.

Most North American car manufacturers began transitioning to distributor-less ignition systems (DIS) around this time, with some models still using distributors until around 2005-2008.

General Motors, for example, stopped using distributors in most of their vehicles by the early 2000s, while Ford phased out distributors around 2002.

Compare with the base Llama 3.1 model:

When did distributors stop being used in cars manufactured in North America?

Did all manufacturers change at the same time. I imagine that OBDII forced the issue.

North American Models and Years below:

1988-2000 - Ford Escort 1.9L< also here >

1995-1997 - GMC Sierra/ Chevrolet Silverado Full Size PU,
Blazer/Jimmy/Pickup (VIN W)

All Years Honda Accord RX and Prelude

It has to be rather complex when people bring in lists of makes/models to a wiki with little or no further explanation. The wiki is simply now only making

[CONTINUES]

Chat Model Writing Python Code

Llama 3.1 (instruct model, with a “system prompt” encouraging brevity):

Please complete the following Python function:

```
def three_equal (a):  
    ''' Returns True if list a has three consecutive equal elements,  
        and False otherwise. '''
```

Here is the completed Python function:

```
'''python  
def three_equal(a):  
    ''' Returns True if list a has three consecutive equal elements,  
        and False otherwise. '''  
    for i in range(len(a) - 2):  
        if a[i] == a[i + 1] == a[i + 2]:  
            return True  
    return False  
'''
```

This program code is correct, with good style.

Story by Llama 3.1 (Instruct Model)

Please complete the following story:

Violins in the Valley

K. E. Evans

John grew up in a musical household. His mother sang in the church choir and his father played the violin in a local orchestra. From a young age, John was fascinated by the sound of the violin and would often sit and listen to his father practice for hours on end.

As he grew older, John's interest in music only deepened, and he began taking piano lessons from a teacher in town. But it wasn't until he stumbled upon an old violin in the attic of their family home that he felt an inexplicable connection to the instrument.

John's father was overjoyed when his son expressed an interest in learning to play the violin and offered to teach him himself. And so, John's musical journey began, with the sweet sounds of violins filling the valley where they lived.

Years passed, and John became a skilled player, joining his father's orchestra and performing in concerts throughout the region. But it was during an impromptu solo performance at a local festival, surrounded by the rolling hills and green pastures of the valley, that John truly found his voice - or rather, his tone.

The music flowed from him like a river, echoing off the hillsides and entrancing the crowd. It was as if the valley itself had come alive in harmony with the violins, creating a symphony of sound that would stay with John forever.

From that day forward, John's love affair with the violin knew no bounds. He traveled far and wide, sharing his music with anyone who would listen, but always returning to the valley where his musical journey began - and where his heart remained.

Story by ChatGPT 4.5

Please complete the following story:

Violins in the Valley

K. E. Evans

John grew up in a musical household. His mother sang in the church choir and his father played violin in the community orchestra. From an early age, John had been fascinated by the graceful curves of his father's violin, the delicate craftsmanship, and the warmth of its voice.

Every evening after dinner, John's father would take out his instrument, carefully polished to a soft glow, and fill their modest home with music that seemed to dance through the walls and echo into the valley below. Each note was like a story, a memory woven through the strings. John dreamed of playing the violin himself, imagining standing on the small stage at the community hall, just like his father did every spring.

On his tenth birthday, John received his very own violin. It was smaller than his father's, but beautifully carved with intricate patterns along its edges. From that day on, John practiced relentlessly, driven by passion and determination. He spent countless hours perfecting his craft, learning the subtle movements of the bow, the precise placement of fingers, and the sensitivity of the strings.

Years passed, and John's talent flourished. His music was imbued with a depth that resonated with all who heard it. The valley became known not only for its breathtaking beauty but also for the enchanting melodies that drifted gently along the evening breeze.

One warm evening, as twilight painted the sky in hues of lavender and gold, John stood on the familiar stage in the community hall. He held his father's violin, now passed down to him, its familiar warmth comforting beneath his fingers. With a gentle breath, he drew the bow across the strings, and music poured forth, vibrant and clear.

As the notes filled the hall, John realized he wasn't merely playing for himself or the gathered audience. He was continuing a legacy, a story that connected generations, echoing through the valley as gently and persistently as the river below.

And in that moment, John understood: music was not simply sound - it was the heart of their valley, a timeless thread binding their lives together in harmony.

Story by Grok 3 Beta

Please complete the following story:

Violins in the Valley

K. E. Evans

John grew up in a musical household. His mother sang in the church choir and his father played the fiddle with a local bluegrass band. From the time John could walk, the sounds of strings and harmonies filled the air of their small home in the rolling hills of the valley. The family didn't have much in the way of money, but they had music, and that was enough to make the days feel rich.

By the time John was eight, his father had carved him a tiny violin from a piece of maple wood, sanded smooth and strung with care. It wasn't perfect - its tone was a little sharp, and the bow frayed at the edges - but to John, it was a treasure. He'd sit on the porch for hours, mimicking the melodies he'd heard his father play, his small fingers fumbling over the strings until they found the right notes. His mother would hum along, her voice weaving through the chirps of crickets and the rustle of leaves in the evening breeze.

As he grew older, John's talent sharpened. The valley wasn't a place known for producing virtuosos, but folks started to whisper about the boy with the homemade violin. By sixteen, he was playing at barn dances and county fairs, his music carrying over the fields like a secret shared between the hills. People said it wasn't just skill - it was something deeper, like the valley itself was singing through him.

One summer, a stranger rolled into town. He wore a pressed suit and carried a leather case, a stark contrast to the worn overalls and dusty boots of the locals. Word got around that he was a scout from some big music school up north, looking for raw talent to polish into something the world would notice. John's father heard about it and, with a mix of pride and reluctance, told the man to come listen to his boy play.

The stranger arrived at their porch on a humid evening, the air thick with the scent of honeysuckle. John, nervous but steady, picked up his battered violin and began. He played a tune his father had taught him, one that started soft and slow, like a creek winding through the woods, then built into a rush of notes that danced like a storm over the ridge. When he finished, the valley seemed to hold its breath.

The man in the suit didn't say much at first. He just nodded, scribbled something in a notebook, and handed John's father a card. "Bring him to the city," he said. "He's got something special." Then he left, his car kicking up dust as it disappeared down the dirt road.

John's parents didn't know what to make of it. The city was a world away - loud, fast, and foreign to the quiet rhythm of their lives. His mother worried he'd lose himself there, swallowed up by concrete and ambition. His father, though, saw a chance for John to carry their music beyond the valley, to let the world hear what they'd nurtured in those hills.

That night, John sat on the porch with his violin, staring out at the shadowed slopes. The crickets sang, and a whippoorwill called in the distance. He thought about the city, about stages bigger than a barn loft, about people who'd never heard the voice of the valley. His fingers brushed the strings, and a single note trembled into the air|clear, bright, and full of everything he knew.

The next morning, he made his choice. With his violin tucked under his arm and his father's blessing in his ears, John climbed into the old pickup truck and set out for the city. The valley watched him go, its echoes clinging to the strings he carried, ready to spill out into a world that didn't yet know what it was about to hear.

Image Generation — DALL-E 2

Concurrently with LLMs, AI systems for generating images from brief text descriptions were also developed (using some of the same methods as LLMs).

To the left is an image produced by DALL-E 2, released by OpenAI in 2022. (But no longer available.)

The same year, image generation systems were released by Midjourney and Stable Diffusion.

Today, many LLMs incorporate image generation features as part of their chat interfaces.



a teddy bear on a skateboard in times square



Images Generated with ChatGPT 4o

Please produce a photographic-style image of an old man sitting on a chair, sharpening a hunting knife.



Please produce an image of an old man sitting on a chair, sharpening a hunting knife, drawn freely with pencil.



Image Analysis, Web Search, Reasoning, Agents, ...

LLMs have now been extended to do many things beyond outputting text:

- **Analyse images:** For example, identify a plant from a photo.
- **Modify images:** For example, change a photo to look like a painting.
- **Search the web:** Find information that wasn't in the model's training data.
- **Think things through:**

Before answering, the LLM produces textual descriptions about how best to solve the problem, or steps along the way, which influence the final answer.

Became prominent with OpenAI's o1 model (2024) and DeepSeek r1 (2025). Now an option for several other models. Very useful for math and programming.

- **Act as an agent:**

Base LLMs don't *do* anything — they just predict — and even for chat versions, thinking of them as doing things is probably not too helpful.

But one can turn an LLM into an agent by repeatedly asking it:

In a situation where..., what is the best thing to do?

Then do what it suggests, and update the next situation according to what happened.

Accessing LLMs in the Cloud

OpenAI at `chat.com` — various versions of chatGPT, o1, etc.

With no account, you just get their low-end model, no images, no saved chats.

With a free account, just one model, but you can make images, save chats.

Paying \$20 USD/month, you have limited use of the best models.

Paying \$200 USD/month, you get everything.

Google at `gemini.google.com` — Gemini models

With no account, you just get their low-end model, no images, no saved chats.

With a free Google account (same as gmail), you have limited use of the best models, and can make images, save chats.

You can upgrade to a paid account to get everything.

xAI at `X.com` or `grok.com` — Grok models

Have to have an account (X, or Google, or Apple)

With free account, can currently use Grok 3 beta (latest model).

Anthropic at `anthropic.com` — Claude models. But TOS require 18 or older.

Remember that when using these services, you're sharing information with them — you may not want to share everything!

Running LLMs Locally

You can run an “open weights” LLM on your own computer.

But for big models, you need lots of memory, preferably in GPUs (or it’s slow).

You need about one byte per parameter. There are some small models with about a billion parameters — should fit on any modern computer — but some approach a trillion parameters.

Some open weight models:

- OpenAI: GPT-2 (of historical interest only), but not their later models.
- Meta: The Llama models, up to Llama 3.1. Interesting because they release the base models as well as the instruct models.
- Google: The Gemma models are open weights, but not as good as the Gemini models.
- DeepSeek: Probably the best open weight models. The biggest require lots of memory, but are relatively fast (not all parameters used at once).

It’s reasonably easy to run these models using the software at ollama.com, which works on Windows, Mac, or Linux.

Try Things!

Understanding what AI can do is important, but also difficult.

They can do some amazing things!

But they can also be amazingly dumb.

They may be “intelligent”, but they are intelligent in a different way than humans.

To learn about AI, it’s crucial to play around, and see what AI systems can and can’t do.

- Ask it to translate from one language you know to another you know.
- Try learning about some subject you don’t know yet by asking an LLM to tutor you — then check whether it gave correct information.
- Try generating an image you’d like to have — see if you can get what you want, and if it looks good. (No funny fingers!)
- Get it to write a program to your specifications, and see whether it works (perhaps after a few tries).
- Get it to solve a math problem — then check whether it’s correct.
- Ask it to write a story.