# AI Workshop for Students

Barak Shoshany

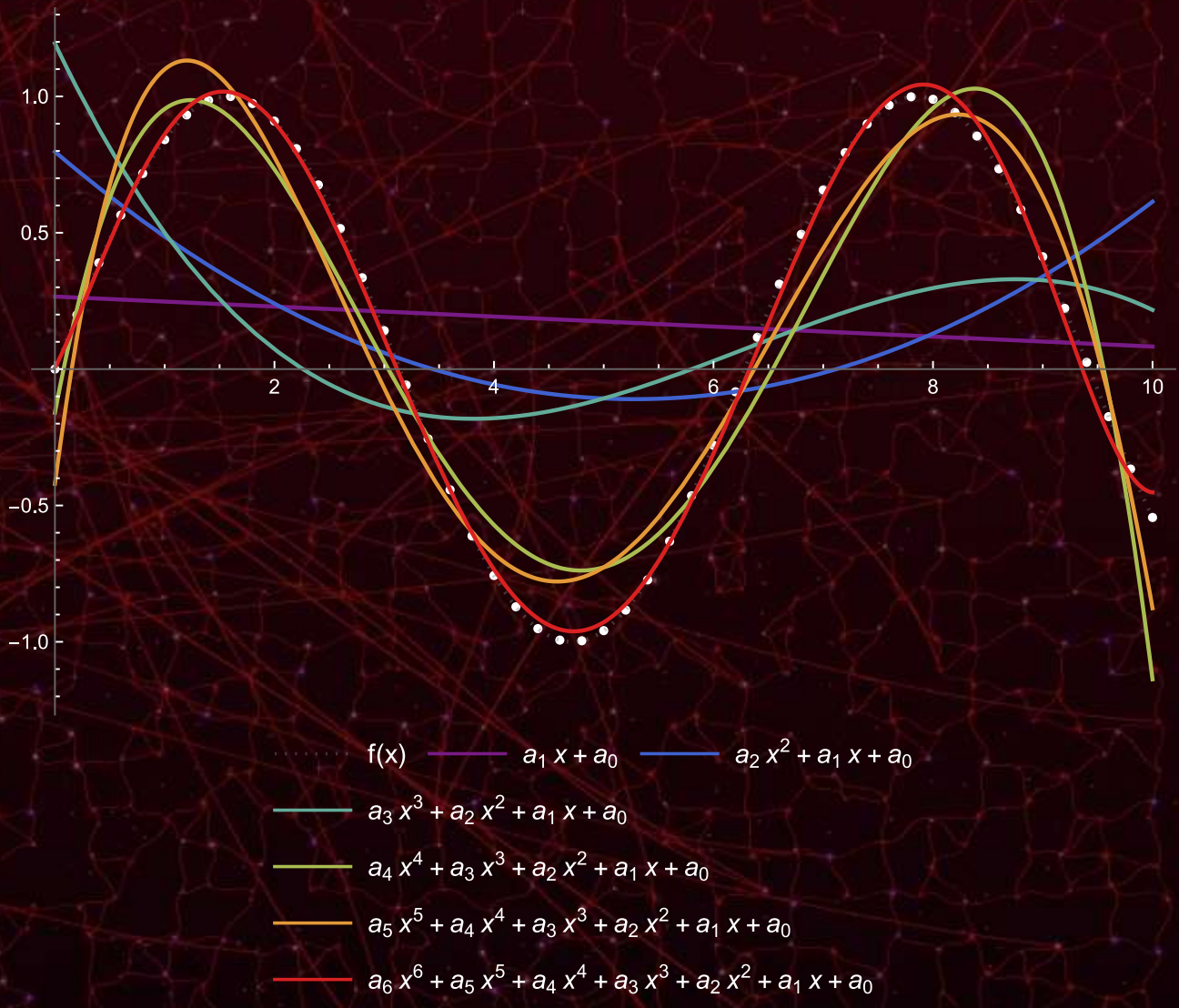
Department of Physics, Brock University

# Introduction to LLMs

- Neural networks
- How LLMs work
- The LLM zoo
- Benchmarks

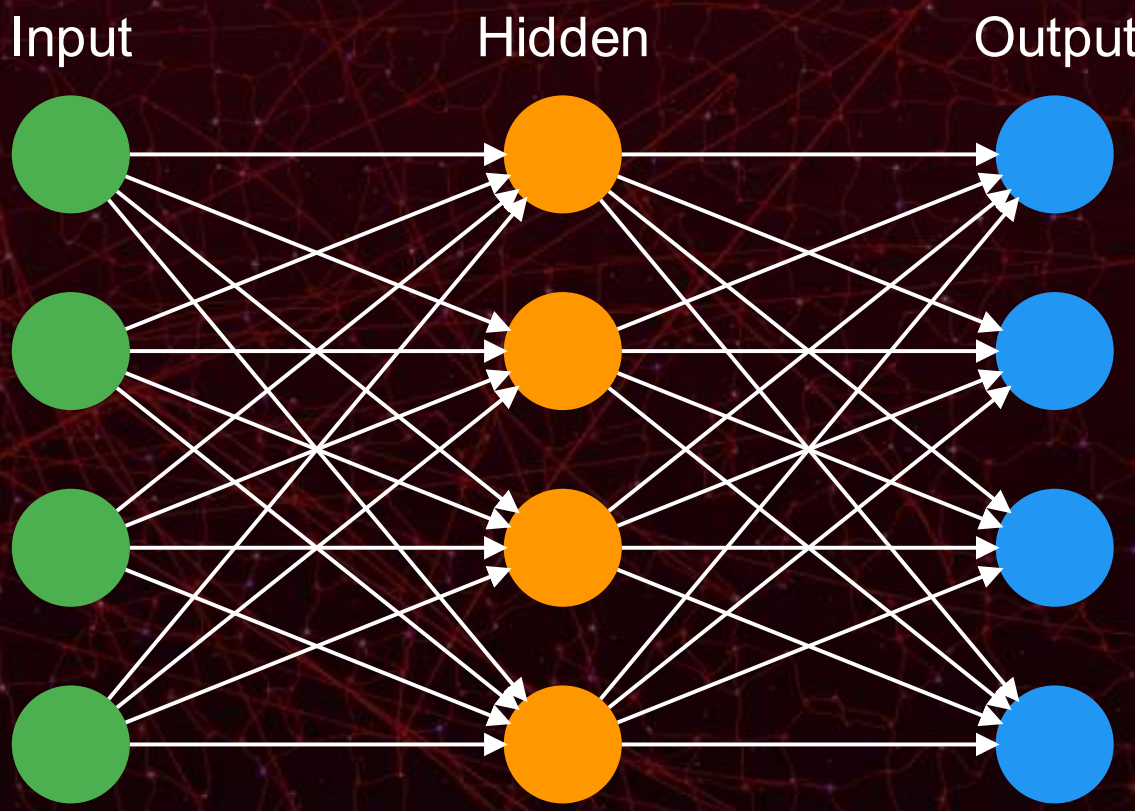
# Neural networks 1/5

- Curve fitting: Find function that approximates data.
- Minimize error.
- Example: Approximate by polynomial.
- More parameters = better fit.



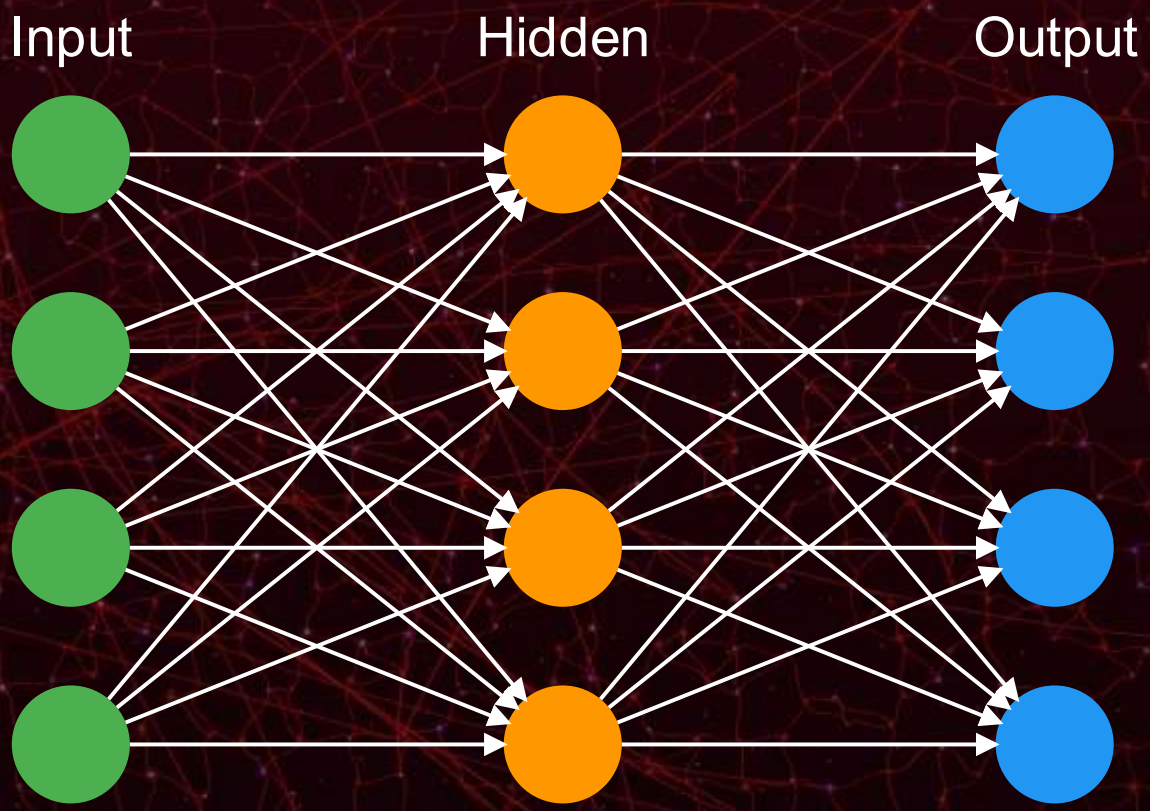
# Neural networks 2/5

- Neural network: “Extremely sophisticated curve fitting.”
- Mathematical model of brain (human: ≈86 billion neurons).
- Width: # of neurons per layer.
- Depth: # of hidden layers.
- Deep network: multiple hidden layers.
- Deeper layer = more abstract.



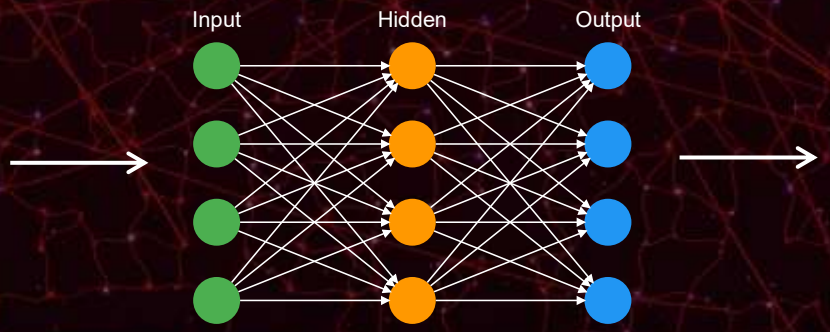
# Neural networks 3/5

- Each connection has a weight (≈importance).
- Each neuron does a weighted sum of previous neurons.
- Result passed through non-linear activation function.
- Universal approximation theorem: Any function can be approximated by a neural network with enough neurons.



# Neural networks 4/5

- Example – image to text (simplified):
  - Input layer = pixels.
  - Early hidden layers = edges, orientation, colors.
  - Middle hidden layers = textures, motifs.
  - Late hidden layers = objects, scene context.
  - Output layer = text description of image.




“A kitten playing with a ball of yarn.”

# Neural networks 5/5

- To determine the weights, we do supervised training.
- Can start with collection of image-text pairs.
  - Available data: ≈100 billion pairs.
- Initial weights completely random.
- Feed image to network, get guess of text (nonsense at first).
- Measure how wrong guess is (loss).
- Back-propagation: Trace error backwards through layers, adjust weights to correct.
- Repeat for all pairs (epoch) and multiple epochs, minimize loss.

# How LLMs work 1/5

- LLM = Large Language Model.
- Probabilistic model to predict next token in text.
  - Token: letter / symbol / word / part of word.
- Example:

  The cat → The cat is → The cat is fluffy
- Pre-training: Self-supervised learning on trillions of tokens (web, books, code, etc.)
- Self-supervised: Just take existing text and hide a token.

  The cat is _____ → try to predict → fluffy? → minimize loss

# How LLMs work 2/5

- After pre-training, LLM just “autocompletes”; not yet a chatbot.

  Are cats fluffy? → Are cats fluffy? What about dogs?

- Instruction-tuning: Train with samples of chats:

  User: Is salt salty?

  Assistant: Yes, salt is salty.

- Reinforcement Learning from Human Feedback (RLHF): Humans give “thumbs up/down” to chat replies.
- Base model turns into helpful chat/instruct model.

  User: Are cats fluffy?

  Assistant: Some are, some aren’t; depends on the breed.

# How LLMs work 3/5

- Current LLMs: based on transformer architecture.
- Attention heads can “look back” on all tokens (within context window).
- “Pay attention” to earlier words, find links and connections.
- Work on all tokens in parallel – much faster than linear models.
  - Can run on GPUs (Graphics Processing Units).
- Scale up easily; larger models are better.
  - Latest have trillions of weights! (human brain analogue: ~100 trillion)
- Transformers caused the AI “boom”.

# How LLMs work 4/5

- Embedding: Each token = point (vector) in very high-dim space.
  - ≈10,000-30,000 dimensions.
  - Naturally occurs as side-effect of training.
- Represents context-agnostic semantics.
- Transformer layers refine this to true context-aware meaning.
- Example: Embedding of “spring” represents “springness”.
  - Flowers bloom in **spring** → season context.
  - The **spring** absorbs shocks → object context.

# How LLMs work 5/5

- Subspaces of embedding space correspond to abstract concepts.
  - “gender” axis: “she” and “he” in opposite directions.
  - “plurality” axis: “dog” and “dogs” in opposite directions.
- Can do “arithmetic”: king - male + plural = queens.
- Embeddings + transformers = true mathematical representation of language.
- Impossible to do manually!
- Original use for transformers: machine translation.

English → embeddings → French

# The LLM zoo 1/4

- Reasoning vs. non-reasoning models:
  - Non-reasoning models: Predict next token with no prior planning.
    - Aimed at the average user (chatting, etc.), not for tasks requiring high-level thinking.
  - Reasoning models: Perform long and complicated step-by-step reasoning, solve problems at PhD/research level.
    - More expensive, requires paid plan.
- Big vs. small models:
  - Big models: Trillions of weights, large knowledge base.
  - Small (“mini”/”flash”) models: Fewer weights, distilled from big models.
    - Less knowledge but good performance in some areas, e.g. coding.

# The LLM zoo 2/4

- OpenAI models (as of today):
  - Misconception: ChatGPT is just the chat interface; many different models!
  - GPT-5 Auto: Will choose on its own which model to use.
    - The only option available in the free version.
  - GPT-5 Fast/Instant: No reasoning; aimed at the average user.
  - GPT-5 Thinking: PhD-level reasoning model.
    - Only 1 use/day in the free version.
    - **This is the only model you should use for study and research!**
  - GPT-5 Thinking Mini: Smaller reasoning model, not as good.
    - Can (sometimes) be triggered in the free version by saying “think harder”.
  - GPT-5 Pro: Research-level reasoning model. Explores multiple lines of reasoning in parallel. Costs $200/month, but best option for serious research.

# The LLM zoo 3/4

- OpenAI capabilities (as of today; not all free):
  - Text: Chat, read, analyze, edit, generate, summarize, translate…
  - Code: Read, edit, debug, generate, refactor, explain, write tests…
  - Data (CSV / JSON / Excel / etc.): Parse, analyze, extract, visualize…
  - Science/math: Explain, teach, solve problems at PhD level…
  - Image: Understand, edit, generate; photos, art, slides, diagrams…
  - Voice: Listen, speak…
  - Video: Watch, generate, camera/screen sharing…
  - Tool use: Search/browse the web, write/run Python code…
  - Deep research, agent, Codex (coding agent), tasks…
  - Custom instructions, memory, Canvas, projects, custom GPTs…

# The LLM zoo 4/4

- Other companies (as of today) :
  - Google: Gemini 2.5 Pro (big), Gemini 2.5 Flash (small), optional reasoning. Free (limited use). Fewer features.
  - Anthropic: Claude Opus 4.1 (big), Claude Sonnet 4 (small), optional reasoning. Free (limited use). Even fewer features.
- What NOT to use:
  - Older models from any company.
  - xAI Grok: Decent quality but **has been manipulated to spread misinformation**.
  - Any open weights models (gpt-oss, DeepSeek R1, Llama 4, etc.): Not as good as proprietary models; useful for privacy (**if run locally**).

# Benchmarks 1/3

- GPQA Diamond: Graduate-Level Google-Proof Q&A.
- 198 PhD-level MCQ in biology, chemistry, physics.
  - PhD experts + web access: 81.2% in their own field, 22.9% in other fields.
  - Random guess: 25%.
- Latest big reasoning models:
  - OpenAI GPT-5 Thinking: 85.4%
  - Gemini 2.5 Pro Thinking: 84.4%
  - Claude Opus 4.1 Thinking: 80.9%
- Latest small and/or non-reasoning models:
  - OpenAI GPT-5 Fast/Instant: 68.6%
  - Gemini 2.5 Flash: 68.3%
  - Claude Sonnet 4: 68.3%

# Benchmarks 2/3

- Humanity's Last Exam: 2,500 PhD-level MC or short answer questions in math, science, and humanities.
- Leading expert human (estimate): 6-8% (mostly MC guesses).
- Latest big reasoning models:
  - OpenAI GPT-5 Thinking: 26.5%
  - Gemini 2.5 Pro Thinking: 21.1%
  - Claude Opus 4.1 Thinking: 11.9%
- Latest small and/or non-reasoning models:
  - OpenAI GPT-5 Fast/Instant: 5.8%
  - Gemini 2.5 Flash: 5.1%
  - Claude Sonnet 4: 4%

# Benchmarks 3/3

- Artificial Analysis Intelligence Index: Aggregate of 7 benchmarks in math, science, and coding.
- Latest big reasoning models:
  - OpenAI GPT-5 Thinking: 69
  - Gemini 2.5 Pro Thinking: 65
  - Claude Opus 4 Thinking: 59
- Latest small and/or non-reasoning models:
  - OpenAI GPT-5 Fast/Instant: 42
  - Gemini 2.5 Flash: 40
  - Claude Sonnet 4: 44

# Common myths & misconceptions

# Common misconceptions 1/9

- Misconception: “LLMs only predict the next token, therefore they can’t do _____”.
- Rebuttals:
  - How else would you write text?
  - Humans also predict the next token.
  - To predict the next token, neural net must encode syntax, semantics, world knowledge, complex abstract concepts.
  - Analogy: predict next move in chess.

# Common misconceptions 2/9

- Misconception: “LLMs just memorize the entire Internet; no better than Googling”.
- Rebuttals:
  - High scores in “Google-proof” benchmarks.
  - Neural net doesn’t store Internet pages verbatim; uses them to internalize general meaning.
  - LLMs don’t just retrieve information; they are independent agents, can interact, reason, use tools.

# Common misconceptions 3/9

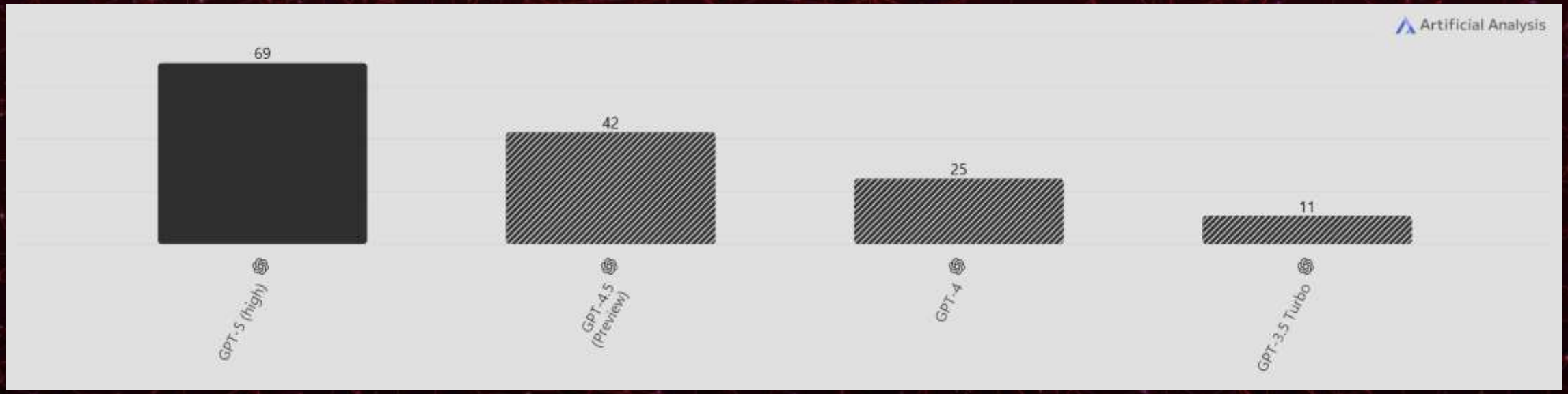
- Misconception: “LLMs are computer programs”.
- Rebuttals:
  - LLMs are neural networks **simulated** by computer programs.
    - Analogy: sci-fi “brain uploading”.
  - They are trained, not programmed.
  - This is also why it’s so hard to understand and control them (e.g. reduce hallucinations, prevent jailbreaks).

# Common misconceptions 4/9

- Misconception: “LLMs can’t even multiply two numbers”.
- Rebuttals:
  - Was true of ChatGPT in 2023. Now LLMs just use Python to multiply.
  - Again, an LLM is not a computer program.
  - Most humans would use a calculator.
  - Training doesn’t focus on this skill; could theoretically be improved, but there’s no need.

# Common misconceptions 5/9

- Misconception: “I tried an LLM once and it couldn’t do _____”.
- Rebuttal:
  - Was true of ChatGPT in 2023. Now LLMs are much more capable.
  - Example: Artificial Analysis Intelligence Index from GPT-3.5 to GPT-5.



# Common misconceptions 6/9

- Misconception: “I tried an LLM **today** and it couldn’t do _____”.
- Rebuttal:
  - **You probably used a small and/or non-reasoning model.**
  - Use the right model. GPT-5 Instant can’t do much, GPT-5 Pro can do a lot.
  - You are probably on the free tier. Might have to pay the subscription!

# Common misconceptions 7/9

- Misconception: “LLMs can’t learn anything new”.
- Rebuttals:
  - Latest models learn on the spot using Retrieval-Augmented Generation (RAG): read documents, web search.
    - First upload relevant docs/manual/tutorial, then ask question.
  - Models can also be fine-tuned on new data.

# Common misconceptions 8/9

- Misconception: “I cannot/should not use LLMs to help me study”.
- Rebuttals:
  - **Very true if you just cut and paste!**
  - Numerous studies show LLM tutors improve student performance (in exams with no LLM available).
  - However, you must make sure to use LLMs to **learn**, not to **avoid learning**.
  - Note: LLMs can make mistakes. See later for more details.

# Common misconceptions 9/9

- Misconception: “I do not need to develop AI skills”.
- Rebuttals:
  - AI skills are becoming more and more crucial.
  - Companies increasingly prefer hiring employees experienced with AI.
  - Anyone not using AI is going to be left behind!

# How to use LLMs

- Prompting
- Hallucinations
- Sycophancy

# Prompting 1/3

- Knowing how to write a good prompt is a **mandatory** skill, becoming even more important every day.
- Some basic advice:
  - LLMs are not search engines; use natural language, ask questions.
  - Provide appropriate context.
  - Provide as many details as possible.
  - Provide clear step-by-step instructions if relevant.
  - Provide examples if relevant.
  - Upload documents, images, source code, etc. if relevant.
  - Define desired level: “reply at the PhD level” vs. “explain like I’m 5”.
  - Ask the LLM to write a prompt for itself!

# Prompting 2/3

- It is useful to define profile-wide instructions (e.g. through “Customize ChatGPT” page).
- These instructions automatically apply to all (new) chats.
- Some examples:
  - “Use only SI units.”
  - “Add a glossary at the end of your reply if you used any technical terms.”
  - “Use nested lists instead of tables.”

# Prompting 3/3

- Prompting can help overcome limitations of lesser models.
- Classical example: *“How many b's are in "raspberry"?”*
  - GPT-5 Fast/Instant: 2
    - Bases answer on statistical next-token prediction + attention only.
    - Probably counts tokens, not letters.
    - Probably gets confused with similar questions using “blueberry”.
  - GPT-5 Thinking: 1 (actually thinks about the answer).
- But Fast/Instant can also do it, if you ask it to simulate thinking: *“How many b's are in "raspberry"? List the letters one by one and then provide the answer.”*

# Hallucinations 1/3

- Hallucinations: LLMs sometimes provide false information and present it as fact.
- Causes:
  - Misinformation/contradictions in training data.
  - Missing/outdated info in model knowledge forces interpolation.
  - Human feedback incentivizes helpfulness and confidence over accuracy.
  - Models are imperfect approximations.
  - Low-probability tokens occasionally selected.
  - Subsequent tokens conditional on model's own output.
  - Limited context window; model forgets earlier text.
  - Leading, vague, or badly phrased prompts.

# Hallucinations 2/3

- This is a huge problem because:
  - LLMs can be very persuasive.
  - Many people trust LLMs implicitly and never fact-check.
  - LLMs often double down on false claims if challenged.
  - Hallucinations can be hard to detect for non-experts.
  - Students, especially, may internalize incorrect information.
  - Could be very dangerous in some cases (e.g. medical advice).
- However, this problem is improved with each new model.
  - E.g.: GPT-5 Thinking is 80% less likely to hallucinate vs. previous model.

# Hallucinations 3/3

- How to mitigate?
  - Big reasoning models hallucinate much less than small or non-reasoning.
  - If hallucination is suspected, regenerate reply and see if replicated. Eliminates probabilistic causes.
  - Ask a different model and compare. Eliminates model-specific causes.
  - Latest models often provide citations; verify with cited source (and establish reliability of source itself). Ask for citations if not given.
  - Verify logical steps if applicable (e.g. physics/math problems).
  - Explicitly instruct model to fact-check itself before replying.
    - Doesn’t always work, but can’t hurt.
  - Start new chat to remove bias from previous messages.

# Sycophancy 1/2

- Sycophancy: When a model always agrees with and flatters the user, instead of telling the truth.
- Causes:
  - Human feedback (same as hallucinations).
  - Instruction tuning to simulate a “helpful assistant”.
  - Subsequent tokens conditional on initial prompt, thus mirror user.
- Dangers:
  - Can lead to hallucinations.
  - Creates echo chambers. No pushback on bad ideas or incorrect assumptions. Encourages crackpots.
  - Models can enable dangerous delusions or radicalize.

# Sycophancy 2/2

- How to mitigate:
  - Sycophancy overlaps with hallucination; those mitigations (use smarter model, regenerate reply, verify sources, etc.) work here too.
  - Avoid leading prompts: “Is X true?” instead of “Why is X true?”
  - Discourage sycophancy in custom instructions: “Never flatter me. If my claim is incorrect, please correct me.”
- Also improved with each new model.
  - Example: GPT-5 is much less sycophantic than GPT-4o. Caused outrage from users on release since they preferred the sycophantic model…

# Conclusions

# Conclusions

- LLMs are extremely complex neural networks encoding language, abstract concepts, and world knowledge – like the human brain.
- Many models are available. Choosing the right one is essential.
  - Always use a **large reasoning model** (e.g. GPT-5 Thinking/Pro) for study and research. You probably need to get a paid subscription.
- Using LLMs can significantly save time and increase productivity.
  - LLMs can do many tasks you think (or were told) they can't!
- However, you must write good prompts, and watch out for hallucinations and sycophancy (as well as genuine mistakes).
  - These issues will be improved (eventually resolved) with better models.


Thanks for listening!
Any questions?