HARDWARE & BEHAVORIAL AI

INC NETWORKS

Team_MHZ

TABLE OF CONTENT

CHAPTER A: INTRODUCTION TO AI

TABLE OF CONTENT

CHAPTER A: INTRODUCTION TO AI

Lesson 2: Emergence of Modern AI (1987 - now)

Lesson 3: Future of AI

TABLE OF CONTENT

CHAPTER A: INTRODUCTION TO AI

TABLE OF CONTENT

CHAPTER B: Behaviorist AI

TABLE OF CONTENT

CHAPTER B: Behaviorist AI

Lesson 3: Markov Decision Process (MDP)

TABLE OF CONTENT

CHAPTER B: Behaviorist AI

Lesson 4: Problems we are solving for RL+ MDP

TABLE OF CONTENT

Chapter C: HARDWARE

TABLE OF CONTENT

CHAPTER C: HARDWARE

TABLE OF CONTENT

CHAPTER C: HARDWARE

TABLE OF CONTENT

CHAPTER C: HARDWARE

TABLE OF CONTENT

CHAPTER D: Networking for AI

WHETTING YOUR APPETITE

Picture this. It is finally Saturday. You can finally have a little break after a whole week of procrastination and boring lectures. You decide to throw away your backpack to one side, lie down on your bed, and start scrolling short videos on Tiktok.

As usual, the videos you receive from the app were funny af - tons of brainrot content being fed to your eyes to have a little escape from all your deadlines and homework. Feels good, right? We've all been there. But have you ever stopped to wonder, why in the world are those videos so good? Like, one of my favorite contents on Tiktok is this one guy arguing with himself about what's the best dipping sauce for french fries. How does the app know what kind of videos I like?

While you were wondering this, your gamer nerd roommate started laughing in satisfaction because some stupid trash talking hackers got banned in their CSGO game. How did the game even know that the dumbass was hacking? Do they have a moderator watching every single game at once? Of course not, right?

WHETTING YOUR APPETITE

I mean , if you have read the title of the book, you could probably have guessed it - they used some sorts of AI to figure these stuffs out. But how exactly?

Think about it like this: We humans learn a lot from experience. You touch a boiling cup of water, it burns, probably shouldn't do that next time. You buy your girlfriend a pet frog for her birthday, she hates it, probably shouldn't do that next time. You get arrested for pointing a flamethrower at a police, well, probably... shouldn't do that next time. (That last one was done by a Florida man by the way).

From this principle, a bunch of nerds came together and wondered: "damn, what if machines can do this as well?". That is how Behaviorist AI was born - and it changed the world. I mean, can you live in this world if you had your mom's Tiktok feed? I don't think so.

WHETTING YOUR APPETITE

Think about it like this: We humans learn a lot from experience. You touch a boiling cup of water, it burns, probably shouldn't so that next time. You buy your girffriend a pet frog for her birthday , she hates it, probably shoudn't do that next time. From this principle, a bunch of nerds came together and wondered " damn, what if machines can do this as well?" . That is how Behaviorist AI was born - and it changed the world. I mean, can you live in this world if you had your mom's Tiktok feed? I don't think so

We know the majority of people reading this are either the professor(s) or fellow students from our university. Therefore, we are positive that 99% of our readers will not even finish half of this book. If there is any student that actually manages to read this book from start to finish, do not talk to them, they're a psychopath.

We get it, it's theoretical, it's boring, it's too many words, we've all been there. But still, we try our best for our content to be as engaging as possible for our fellow brainrotten people.

Best of luck to you,
Team MHz

CHAPTER A: INTRODUCTION TO AI

CHAPTER A: INTRODUCTION TO AI

Lesson 1

Early days of AI (1943-1986)

PRO TIP

If you are reading this book for the technical concepts and all that, you should only scan through this lesson, treating it as extra reading, or just skip it. This lesson just focuses on the definition of AI in itself, as well as the different perspectives where people have approached AI in the past.

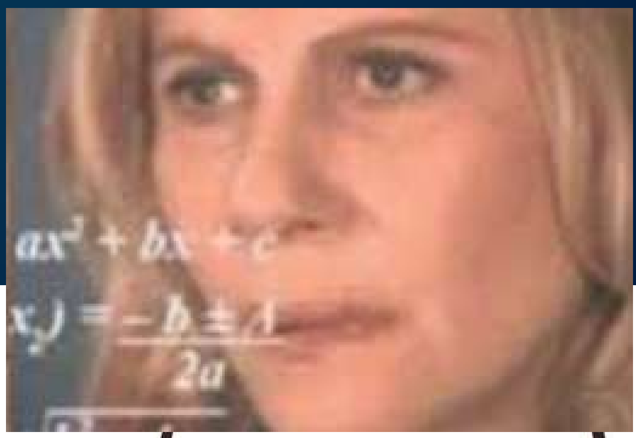
While some of the concepts mentioned in this lesson is still used to this day (that is Symbolic AI), well, just saying, most of the actually interesting concepts will be in <u>Lesson 2: Emergence of Modern AI (1986 - present)</u>

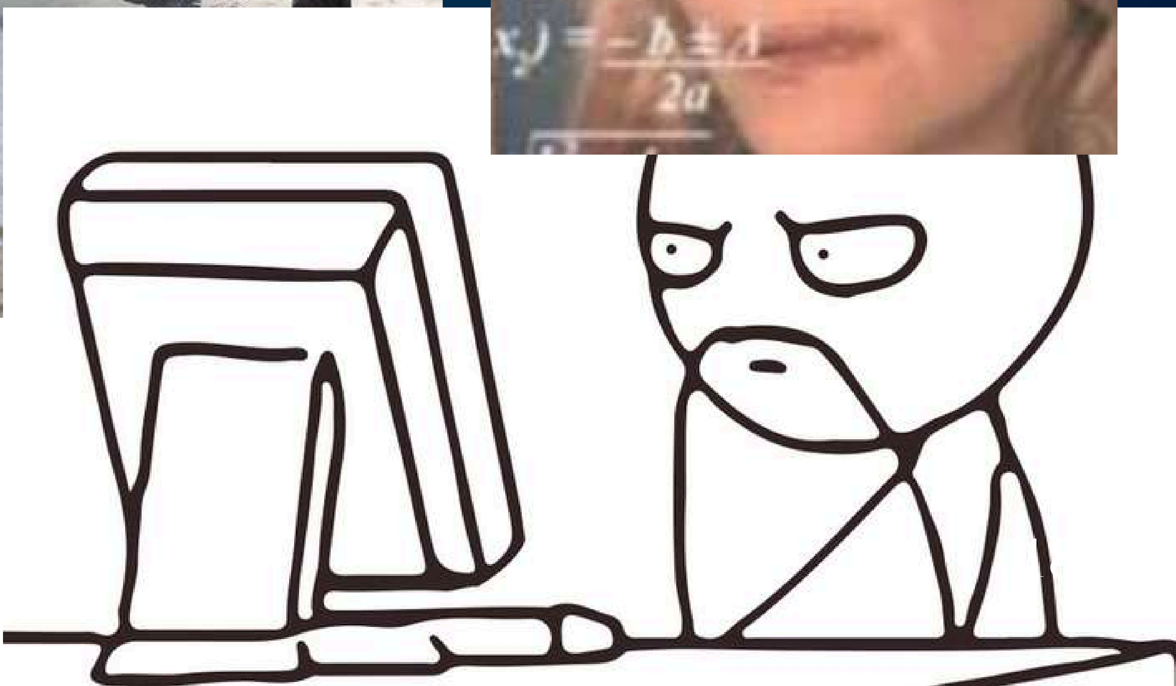
But for people who are actually interested in the history stuffs, it is best for you to approach this chapter with a pen and a notebook in hand

I. DEFINITION OF AI

Ah, yes. The definition of AI. What is an AI? This question actually goes really deep the more you look into it.

On the surface, maybe an AI is just a machine that can mimic some human abilities such as recognising images and text, learning and reasoning, problem solving, and so on. But, is that really enough for a machine to be "intelligent"? While it may look like they are behaving like a human, in reality, they are really just a very sophisticated program that analyzes vast amounts of data, and turns the user's input into outputs. Can they really be intelligent without having emotional consciousness and understanding? You can even extend so much that you end up having an existential crisis yourself. What if there is even another type of "intelligence" so sophisticated it is not yet imagined by human beings? Are we even intelligent? What if we are beings made by some extraterrestrial creature? Are WE AI?





I. DEFINITION OF AI

But for the purpose of simplicity, we are going to use the first definition that we initialized:

<u>AI is a machine that can mimic human abilities such as recognising images and text, learning, reasoning and problem solving</u>

With this kind of machine, humanity believed that we can do awesome stuff to the world and to ourselves - and it sure did - you can see AI everywhere nowadays. But have you stopped to wonder: How did we even figure this stuff out? How did people in the past approach this problem? Let’s find out right now in the following section

II. THE ORIGINS OF AI

Fun fact:

The concept of AI was actually initialized quite a long time ago, specifically in ancient Greek. (Yes, I did not expect the Greeks to be in this book at first either). It was called - The myth of Talos.

The myth describes Talos as a giant bronze man built by Hephaestus, the Greek god of invention and blacksmithing. Talos was commissioned by Zeus, the king of Greek gods, to protect the island of Crete from invaders. He marched around the island three times every day and hurled boulders at approaching enemy ships." - Stanford Report

II. THE ORIGINS OF AI

1. INCEPTION OF AI (1943-1956)

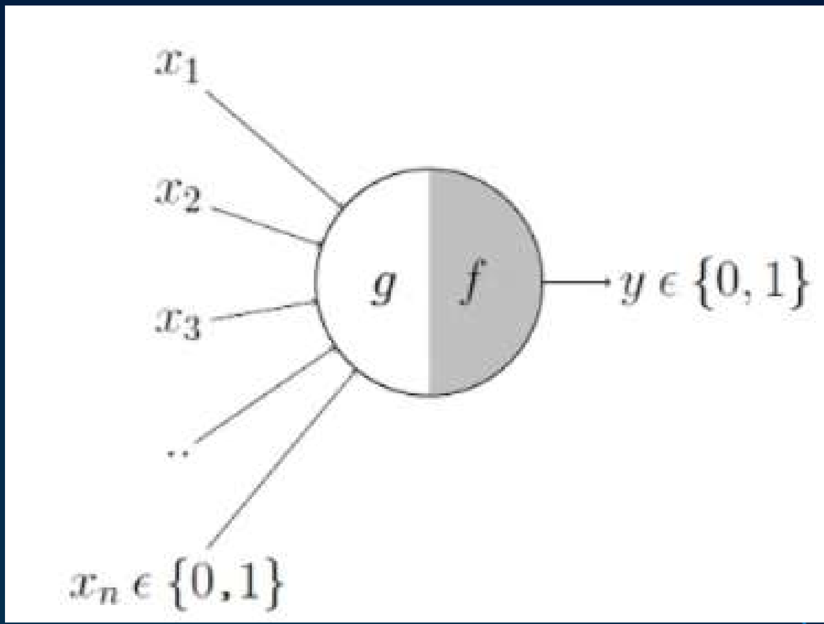
1.1 MCCULLOCH AND WALTER PITTS (1943)

The beginning of modern AI research actually began with these 2 nerds - Warren McCulloch and Walter Pitts, who mathematically modeled the human neurons. Sounds pretty cool, but it just simply looks like this:

II. THE ORIGINS OF AI

1.INCEPTION OF AI (1943-1956)

1.1 MCCULLOCH AND WALTER PITTS (1943)



You can think of x1, x2, x3 ... as like multiple values of either True (1) or False (0). After put into function f, it outputs another True (1) or False (0) signal.

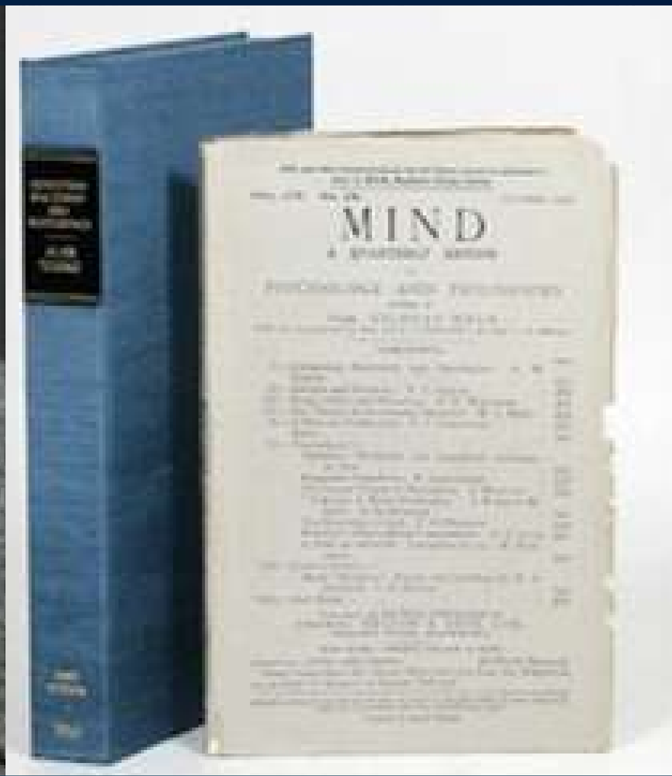
Take an example, you just saw a girl, and you decide whether or not you like her. By looking at her appearance, personality, hobbies etc..., you decide that you do like her. So by asking... "is she pretty? Yes! Is she fun? Yes! Do we like to play Smash bros? Yes!", you are basically adding a bunch of either Yes or No (or True and False) inputs into your brain, which will eventually drive your brain into the conclusion: "Do I like her? Yes!"

Of course, our brain is a lot more complicated than that, having millions and millions of neurons for every single decision we make. But on a basic level, this is how the brain's neural network system works.

This concept of neural networks wasn't used a lot by the early AI programs, but it certainly laid a good foundation for the concept of Connectionist AI, which is basically AI that operates on neural networks (Get it? because those neurons are connected to each other). This Connectionist approach is actually used a lot nowadays, with all those machine learning and deep learning stuff.

1.2 ALAN TURING'S CONTRIBUTIONS (1950)

Alan Turing is quite a big guy in the field of CS and AI - in fact, you may have heard of his name a lot by no



His biggest contribution to the AI and CS field is evident in his research paper called "Computing Machinery and Intelligence", where he demonstrated these key points:

-The Imitation Game (aka the Turing Test):

Turing proposes a game where a human (the "interrogator") interacts with a machine and another human without knowing which is which. If the interrogator is unable to reliably distinguish between the human and the machine, the machine can be said to exhibit intelligent behavior. Basically, if machine fool people, machine smart (Do not worry too much about this test, it has been outdated for a long time now)

1.2 ALAN TURING'S CONTRIBUTIONS (1950)

-Machines and Thinking: Turing suggests that instead of defining "thinking," we should focus on observing behavior. If a machine behaves like a human, it should be considered intelligent.

-Objections to Machine Intelligence: Turing acknowledges several objections to the idea of machines being intelligent, including:

+) The Argument from Disabilities: Some argue that machines will never have certain human attributes (e.g.,emotions).

+) Lady Lovelace's Objection: This claims that machines can only do what they are programmed to do, lacking creativity.
The Argument from Consciousness: Turing counters that consciousness is not necessary for intelligent behavior.

-Future of AI: Turing speculates about the future of artificial intelligence, suggesting that advancements in technology could lead to more sophisticated machines that challenge our understanding of intelligence.

1.2 ALAN TURING'S CONTRIBUTIONS (1950)

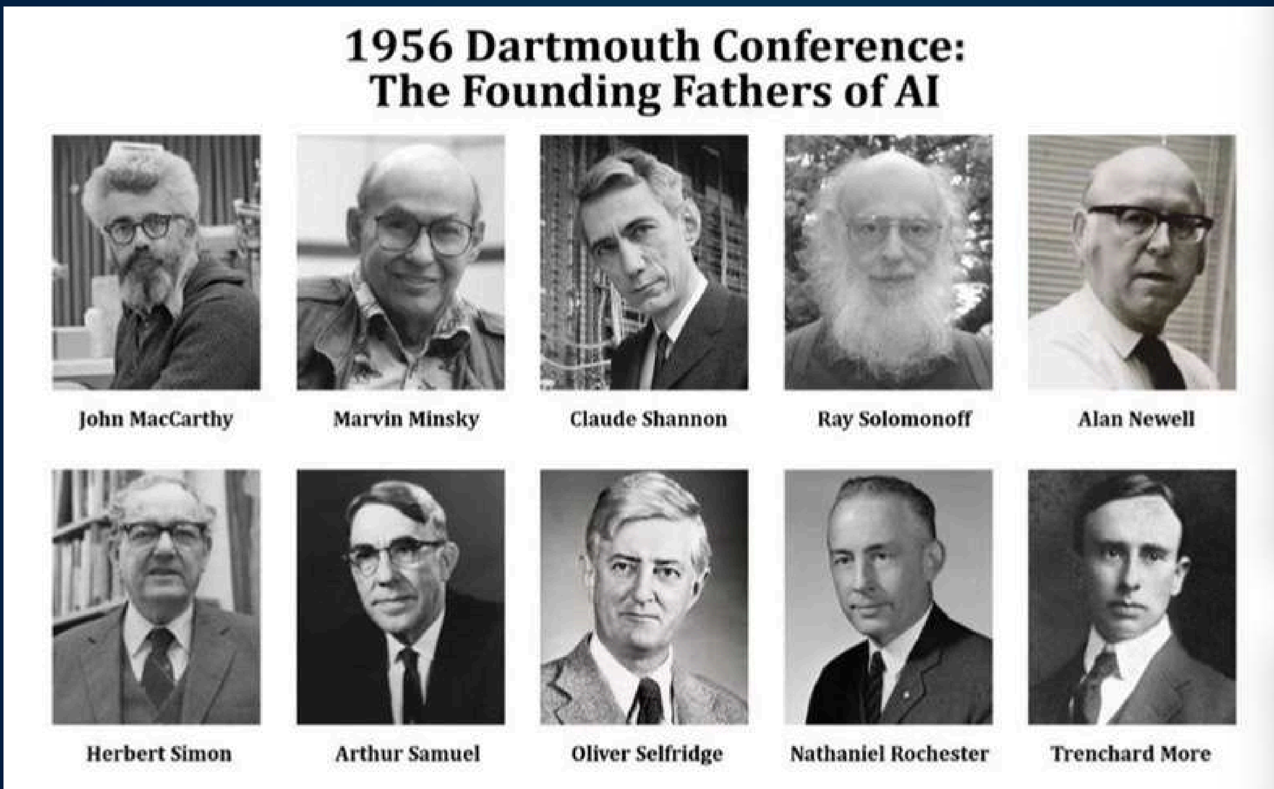
Fun fact:

Demonstrating the Turing Test, "AI21 Labs" have created the game "Human or Not?". The game lets you chat with someone for 2 minutes, and then let the player guess whether they have been chatting with a human or an AI.

You can try out the game at https://www.humanornot.ai (not sponsored btw)

1.3 DARTMOUTH CONFERENCE (1956)

The 1956 Dartmouth Conference, organized by John McCarthy, Marvin Minsky, Nathaniel Rochester, and Claude Shannon, is often considered the birth of AI as a field. It aimed to explore ways machines could simulate aspects of human intelligence. The Conference lasted for around 6 - 8 weeks in Dartmouth College. It is basically just 4 bored guys coming together doing historically significant brainstorming for the concept of Artificial Intelligence.



Fun fact:

"The Dartmouth Conference" also refers to a historical conference between the Americans and Soviets, which also took place in Dartmouth College from 1961 and (apparently) still continues to this day. Therefore, the AI Dartmouth Conference is sometimes referred to as the "Dartmouth workshop" instead.

2.SYMBOLIC AI AND EARLY PROGRAMS (1952-1966):

2.1 DEFINITION OF SYMBOLIC AI

In Artificial Intelligence, symbolic AI are algorithms that process “symbols” and their relationships. The said “symbols” are, simply put, the computer representations of objects, concepts and problems in the real world.

Take, for example, a program that gives the doctor suggestions based on the patient’s symptoms. If the patient is reported to have signs of fever and headache, a symbolic AI algorithm would interpret it something like this:

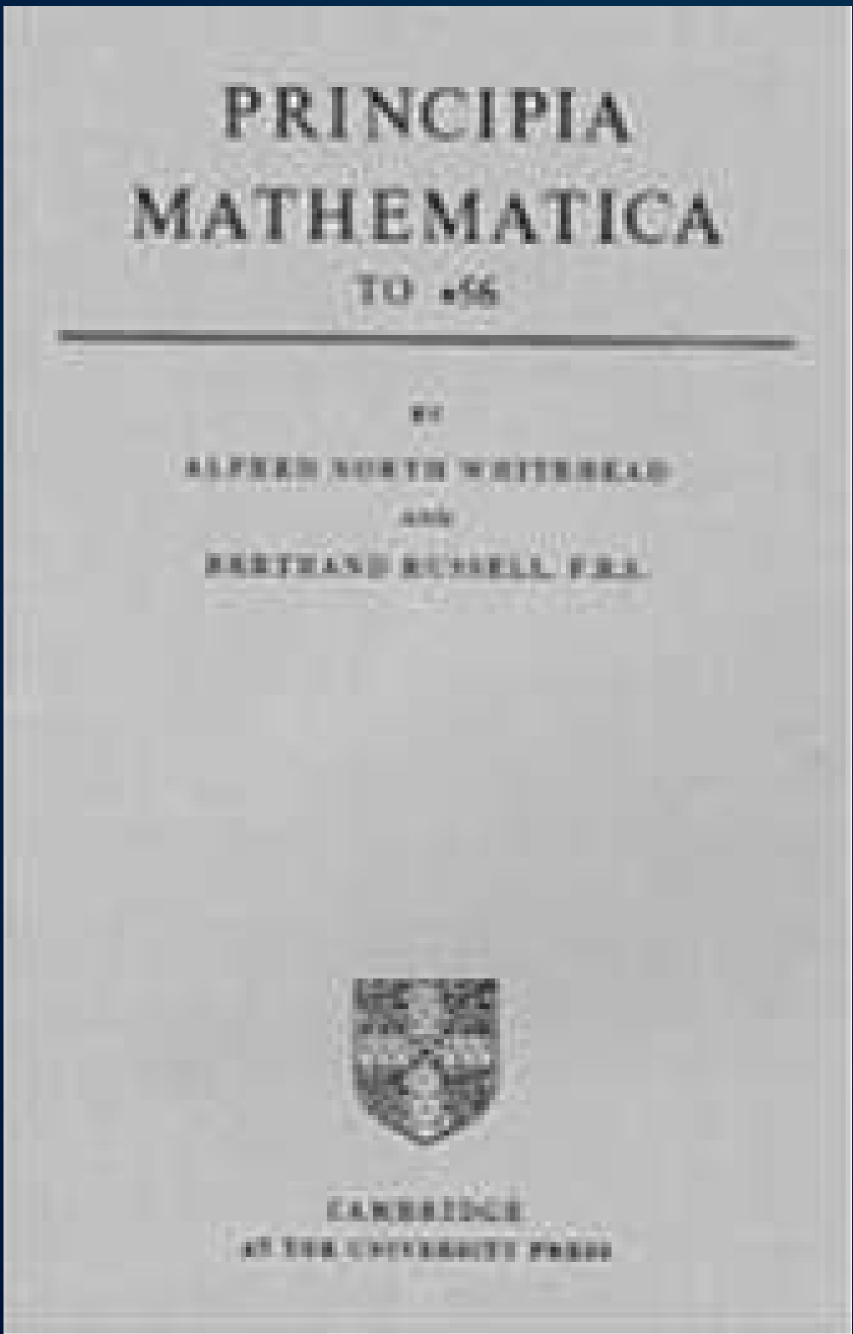
IF patient has fever AND patient has headache

THEN patient may have a flu.

Programs that use rules like this were the main research point during the period from 1950s to 1990s. Researchers were convinced this was THE approach to creating the ultimate Artificial Intelligence, well, spoiler alert, until machine learning and deep learning’s effectiveness was recognised. Nevertheless, people during this time did create some really cool stuff with this concept in mind.

2.2 THE LOGIC THEORIST (1955)

The Logic Theorist is the first ever program in history to perform "automated reasoning", and was widely regarded as the first ever AI program. The Logic Theorist managed to prove 38 of the first 52 theorems in chapter two of Whitehead and Bertrand Russell's Principia Mathematica, and even found new and shorter proofs for some of them.


PRINCIPIA
MATHEMATICA
TO *56

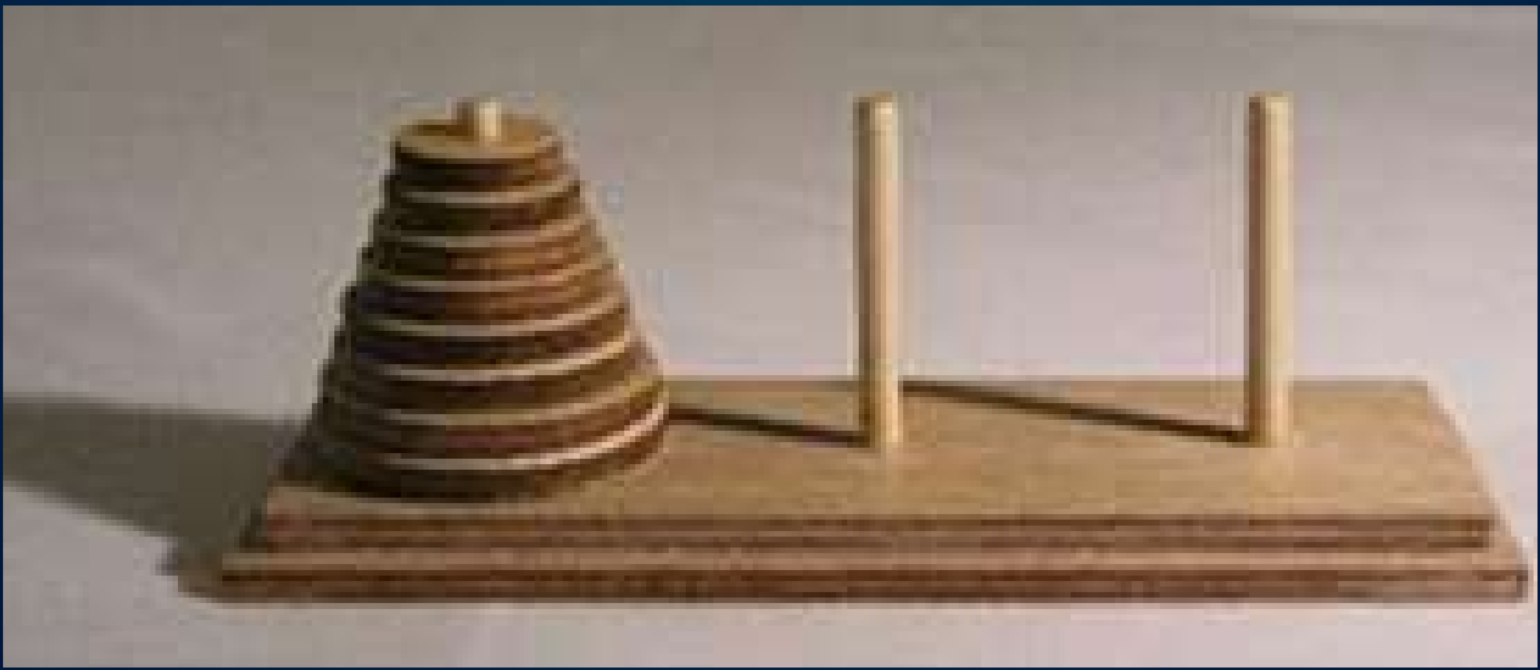
BY
ALFRED NORTH WHITEHEAD
AND
BERTRAND RUSSELL, F.R.S.

CAMBRIDGE
AT THE UNIVERSITY PRESS

II. THE ORIGINS OF AI

2.3 General Problem Solver (GPS) (1957)

• As the name implies, GPS is a program that's designed to, well, solve general problems. Quite a lot of problem-solving programs have existed in the past, but they only solve specific problems that they're instructed to do. Therefore, the GPS aims to broaden the range of stuff that it can do, making it one of the first "useful" AI programs.
• Nevertheless, while it can sufficiently solve problems that can be formally defined, such as the Hanoi Tower problem, it lacked the ability to pull off any real world problems, because the machine would have to include way too many rules to figure such problems out.



II. THE ORIGINS OF AI

2.4 Samuel's Checkers Playing Program (1959):

In 1959, Arthur Samuel created a program that can not only calculate the winning chance of each player in a checker game, but also play checkers on its own. What is interesting about this program is that the machine itself learns on its own: as it plays more checkers games, it also slowly gets better and better at the game. Eventually, the program itself beat one of the best American checkers players at that time - which was quite a big thing at the time.

II. THE ORIGINS OF AI

2.5 ELIZA (1964-1966):

ELIZA, created by Joseph Weizenbaum, was one of the first chatbots, or Natural Language Processing (NLP) systems, ever created. It's also one of the first programs to be qualified to take the Turing Test!

Many people were convinced of ELIZA's intelligence and capabilities to understand emotions, while its creator, Weizenbaum, insisted the contrary. The program merely works by analyzing user's inputted keywords, assigning each value to such keywords, and then printing out a predefined answer. To put it bluntly, ELIZA was just a bunch of if - else rules without any abilities to understand complex human emotions and stuff like that.

II. THE ORIGINS OF AI

2.5 ELIZA (1964-1966):

```
response[ 89]="You seem quite positive.";
response[ 90]="Are you sure about that?";
response[ 91]="I see. Let's explore that a bit more.";
response[ 92]="I understand.";
response[ 93]="Why do you bring up the topic of friends?";
response[ 94]="Do your friends worry you?";
response[ 95]="Do your friends say things to you thay seem unfriendly
response[ 96]="Are you sure your friends are truly friends?";
response[ 97]="Perhaps you impose on these people. Is that a possibil
response[ 98]="Perhaps your love for people actually worries you.";
response[ 99]="Do computers make you uncomfortable?";
response[100]="Are you talking about me in particular?";
response[101]="Are you frightened by machines?";
response[102]="Why do you mention computers?";
response[103]="What do you think inanimate things have to do with you
response[104]="Don't you believe technology can help people?";
response[105]="What is it about technology that worries you?";
response[106]="Would you say that you have psychological problems?";
response[107]="What does that suggest to you?";
response[108]="I see. Let's try another topic and we will come back t
response[109]="I'm not sure I understand you fully.";
response[110]="I think you are blocking what you really want to say."
response[111]="Can you elaborate on that?";
response[112]="That is an interesting response. Let me make anote of
response[113]="Okay, now you're repeating yourself. That is not movin
response[114]="Do you really expect a different answer if you keep re
response[115]="Let's take a moment here. Think about what you just sa
response[116]="Again? We need to move on.";
response[117]="And hello to you.";
```

Therefore, people also created the concept "the Eliza Effect", stating that people tend to instinctively think that machines act like humans. This is the reason why the Turing test became outdated - People are easily fooled by machines.

II. THE ORIGINS OF AI

3.AI Winter (1966-1973):

As more AI programs were created, people had unrealistically high expectations for it. They believed that people could make an AI program so powerful it can solve any given problem in the real world (basically, a GPS but better). But in reality, that just didn't happen, which was disappointing. AI was overpromising, and it was also underdelivering.

II. THE ORIGINS OF AI

3.AI Winter (1966-1973):

And then everything just started getting worse and worse for the research of AI in this era, usually referred to as the "AI winter":

- The early design of AI has very limited scalability, which makes real world problems very difficult to implement.

- Computers were not advanced enough back then, so they couldn't handle all the computations and data storage.

- People stopped giving money for AI research.

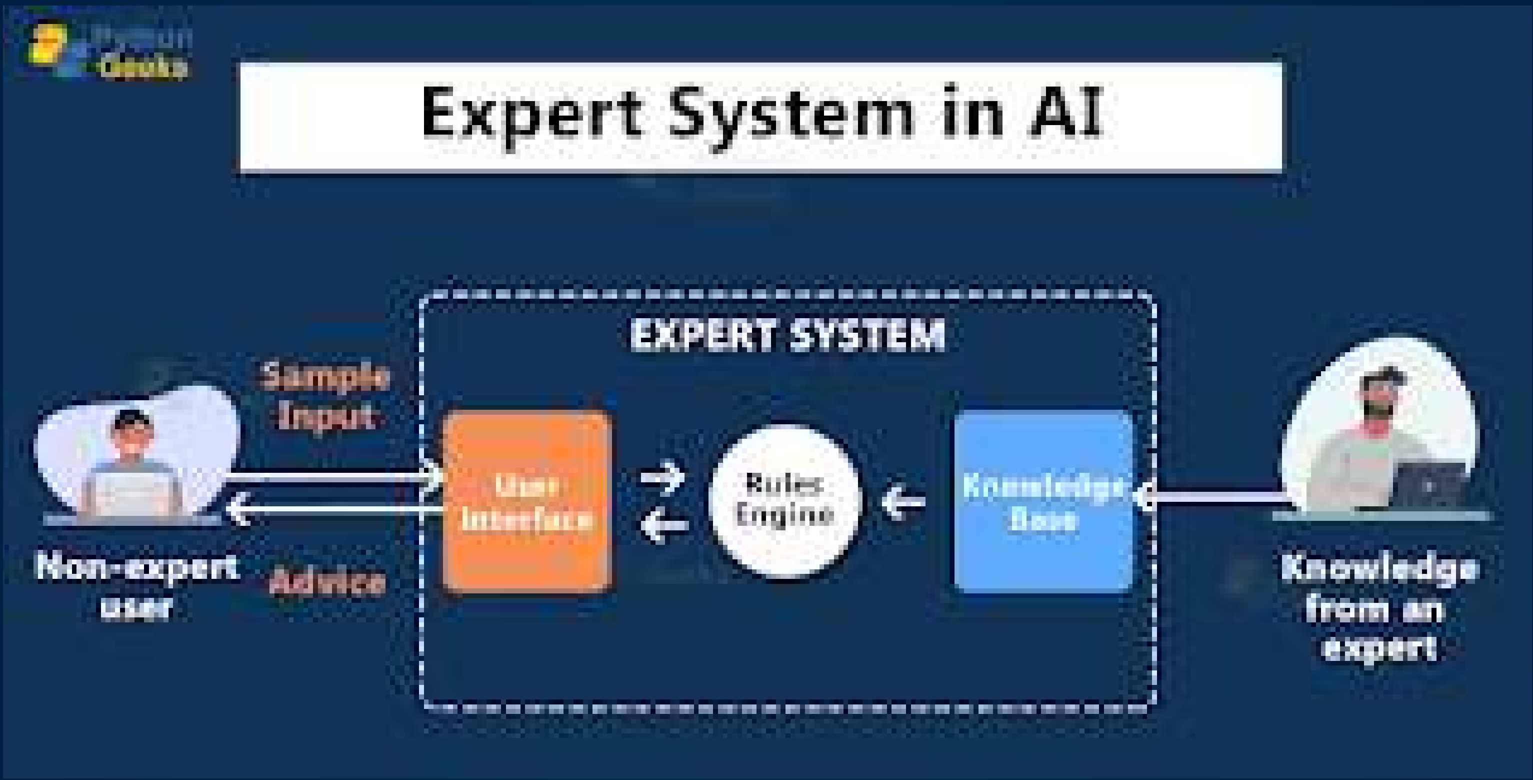
- No one cared about AI anymore, and decided to focus on other research topics.

Eventually, people gave up trying to make a program that can solve every problem, and instead switched to creating programs that can solve specialized tasks, namely expert systems.

II. THE ORIGINS OF AI

4.Expert Systems (1969-1986):

Expert systems are machines that have been fed human knowledge by human experts. And then, using a bunch of if-else statements, it would write the human expert knowledge out to a dumb human. This concept was the main focus of AI research for a while.



Some of the expert systems programs back in the days include:

- DENDRAL (1965): a chemical-analysis expert system, and also one of the first ever successful expert systems. The substance to be analyzed might, for example, be a complicated compound of carbon, hydrogen, and nitrogen. Starting from spectrographic data obtained from the substance, DENDRAL would hypothesize the substance's molecular structure. DENDRAL's performance rivaled that of chemists expert at this task, and the program was used in industry and in academia.

II. THE ORIGINS OF AI

4.Expert Systems (1969-1986):

Many of the future expert systems also derived from DENDRAL, including MYCIN, MOLGEN, PROSPECTOR, XCON, and STEAMER.

- MYCIN (1970s): One of the earliest and most famous expert systems, designed for diagnosing bacterial infections and recommending antibiotics. MYCIN's knowledge was encoded as a set of rules that could be updated or expanded

II. THE ORIGINS OF AI

4.Expert Systems (1969-1986):

The program was named MYCIN because many antibiotics' names end with the suffix "-mycin", for example azithromycin and tobramycin.

MYCIN's user interface

- XCON/R1 (1978): stands for eXpert CONfigurator. A rule-based system used by Digital Equipment Corporation (DEC) to ensure customers were supplied with all the components and software that was needed to make up the specified computer system that they had ordered. It helped DEC save millions of dollars by reducing the number of incorrect or inconsistent configurations.

These systems excelled in narrow domains but faced challenges in knowledge acquisition, maintenance, and scalability, because translating human knowledge into machine language is very difficult, exhausting, time-consuming, and tedious. Furthermore, the program dies when faced with problems that are not predefined in its system, making them less useful than that of a human expert. Therefore, research on expert systems slowly declined.

Nevertheless, after the decline of expert systems, people soon figured out a different approach to AI that changed everything for the better. The methods are now known as Machine Learning and Deep Learning, and it's still being used widely in various AI models these days. How did those turn out? Let's find out in the next lesson - Emergence of Modern AI

III. EXERCISES

Lesson 1: Early days of AI

Ex1: Select the correct answer

1. Who were the first researchers to mathematically model human neurons in 1943?

A) Alan Turing and John McCarthy
B) Warren McCulloch and Walter Pitts
C) Marvin Minsky and Nathaniel Rochester
D) Claude Shannon and Lady Lovelace
Answer: B) Warren McCulloch and Walter Pitts

2. What concept did Warren McCulloch and Walter Pitts' model lay the foundation for?

A) Turing Test
B) Connectionist AI
C) Symbolic AI
D) Machine Learning
Answer: B) Connectionist AI

III. EXERCISES

Lesson 1: Early days of AI

Ex1: Select the correct answer

3. What is the main idea behind Alan Turing's Imitation Game (Turing Test)?

A) Machines should imitate human emotions

B) A machine is intelligent if it can fool a human into thinking it is human

C) Machines should perform tasks faster than humans

D) Machines should mimic all aspects of human behaviors

Answer: B) A machine is intelligent if it can fool a human into thinking it is human

4. Which of the following was NOT one of the objections to machine intelligence addressed by Turing?

A) Argument from Disabilities

B) Argument from Time Complexity

C) Lady Lovelace's Objection

D) Argument from Consciousness

Answer: B) Argument from Time Complexity

III. EXERCISES

Lesson 1: Early days of AI

Ex1: Select the correct answer

5. What does the term “AI Winter” refer to?

A) A period of rapid AI development

B) A time when AI research showed due to high expectations and low results

C) The era whe expert systems were invented

D) A phase of AI’s dominance in solving real-world problems

Answer: B) A time when AI research showed due to high expectations and low results

6. Which of the following would be a complex scenario for AI in probabilistic reasoning?

A) Predicting the outcome of a simple coin toss

B) Calculating the probability of falling given known factors

C) Processing multiple possible responses to a middle school kid’s confession

D) Finding a route with the shortest distance

Answer: B) Argument from Time Complexity

III. EXERCISES

Lesson 1: Early days of AI

Ex1: Select the correct answer

5. What does the term "AI Winter" refer to?

A) A period of rapid AI development

B) A time when AI research showed due to high expectations and low results

C) The era whe expert systems were invented

D) A phase of AI's dominance in solving real-world problems

Answer: B) A time when AI research showed due to high expectations and low results

6. Which of the following would be a complex scenario for AI in probabilistic reasoning?

A) Predicting the outcome of a simple coin toss

B) Calculating the probability of falling given known factors

C) Processing multiple possible responses to a middle school kid's confession

D) Finding a route with the shortest distance

Answer: B) Argument from Time Complexity

III. EXERCISES:

Ex2: Fill in the blanks:

In 1943, Warren McCulloch and Walter In 1943, Warren McCulloch and Walter Pitts mathematically modeled _________ neurons.

Answer: human

The ___________ Test proposed a way to determine if a machine exhibited intelligent behavior.

Answer: Turing

The term "__________ Effect" refers to the tendency of people to attribute human-like understanding to machines.

Answer: Eliza

Turing addressed multiple objections to machine intelligence, including the Argument from ____________, Lady Lovelace's Objection, and the Argument from Consciousness.

Answer: Disabilities

The approach where AI operates based on neural networks is called __________ AI.

Answer: Connectionist

__________ was one of the earliest expert systems, designed to diagnose bacterial infections.

Answer: MYCIN

III. EXERCISES:

3.Determine whether the following statement(s) are True or False

1, The Dartmouth Conference is considered the starting point of AI as a field.

Answer: True

2, Alan Turing believed that for a machine to be considered intelligent, it must possess human emotions.

Answer: False

3, Lady Lovelace's Objection argued that machines could be creative and act beyond their programming.

Answer: False

4, Alan Turing's Imitation Game suggests that behavior, rather than consciousness, can indicate intelligence in machines.

Answer: True

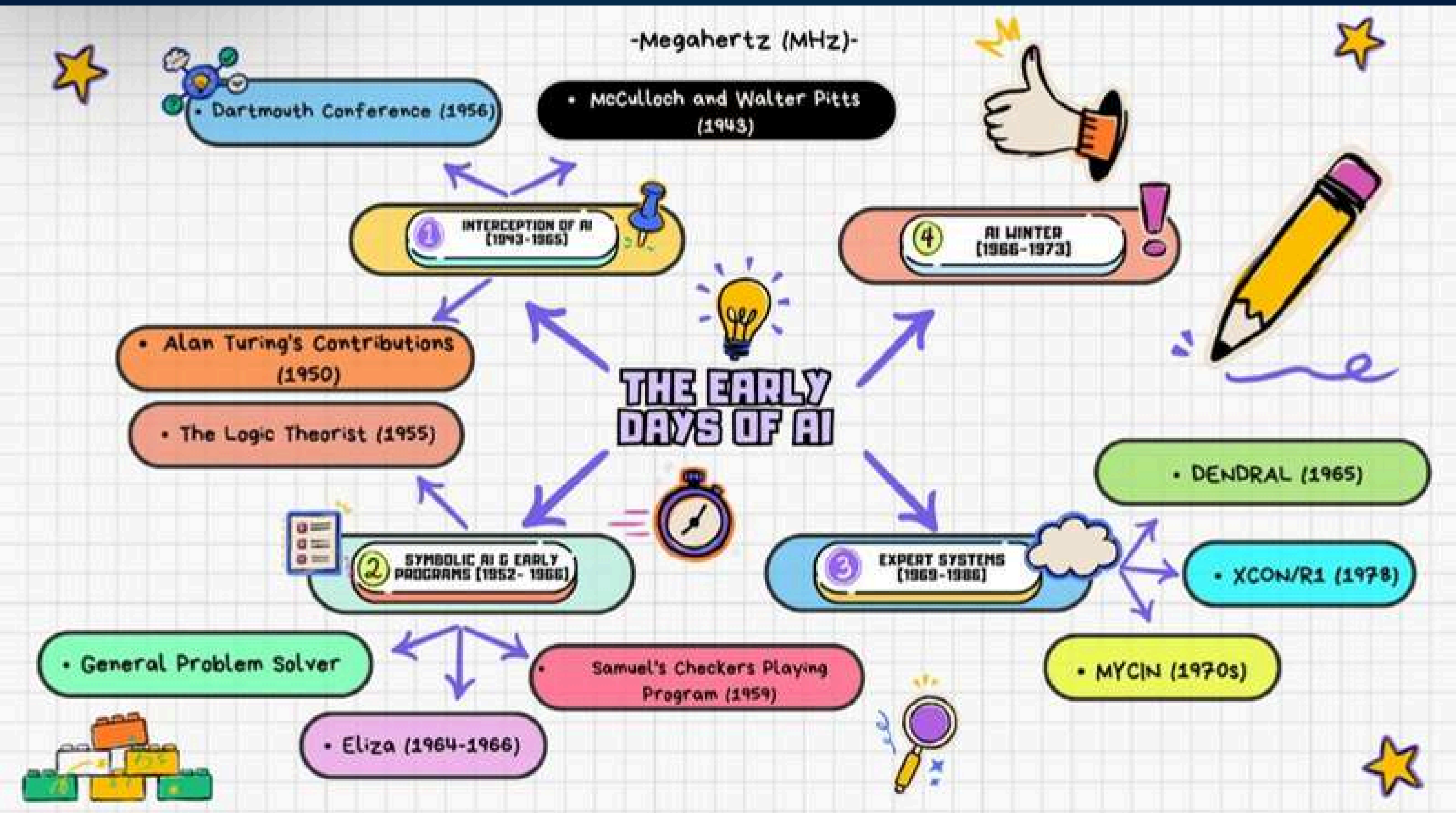
5, Connectionist AI relies on a symbolic, rule-based approach rather than neural networks.

Answer: False

MYCIN was named because it was used to diagnose and treat bacterial infections, with many antibiotics having names ending in "-mycin."

Answer: True

IV. MIND MAP:

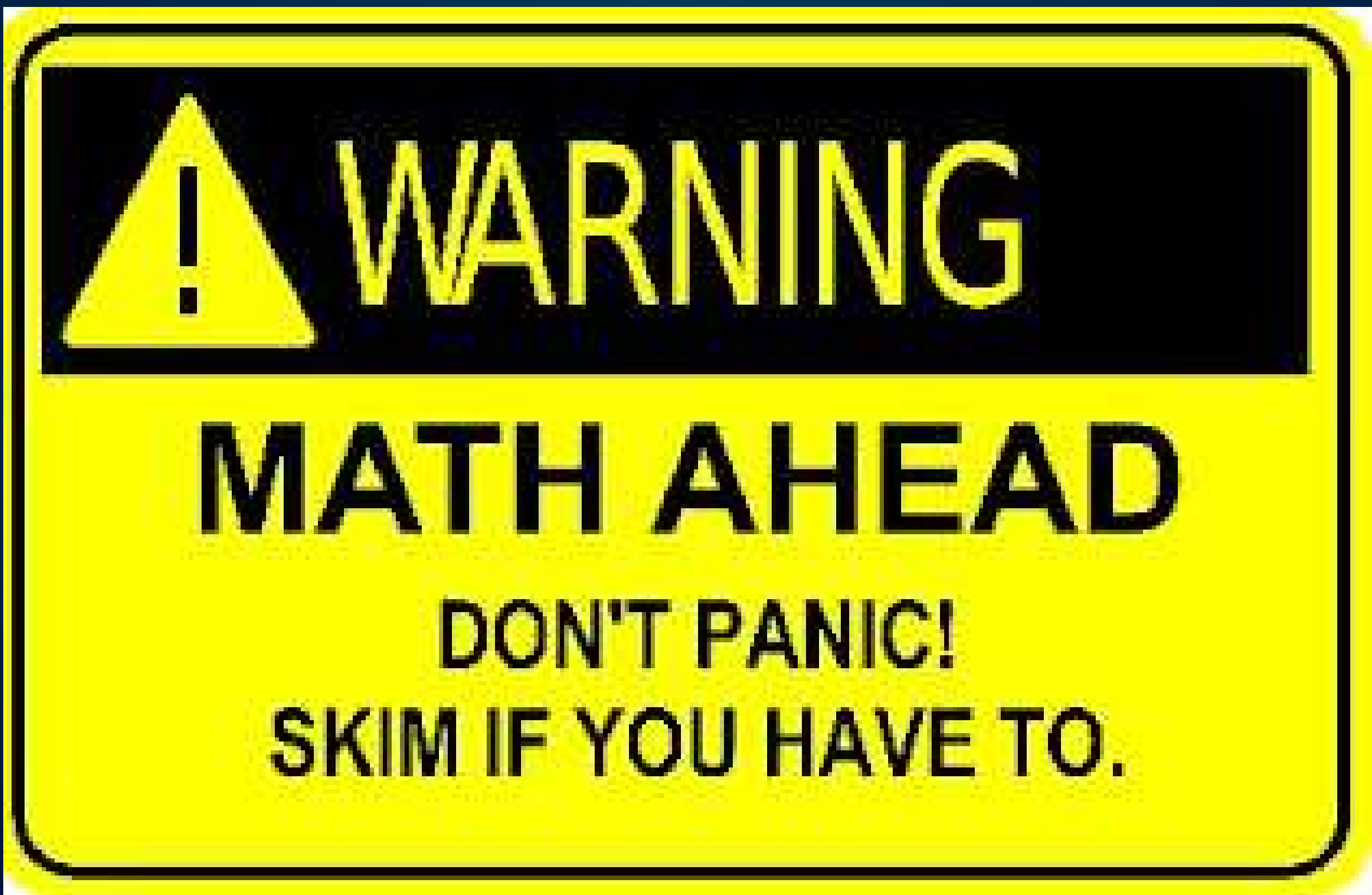


-Megahertz (MHz)-
• Dartmouth Conference (1956)
• McCulloch and Walter Pitts (1943)
1 INTERCEPTION OF AI (1943-1965)
4 AI WINTER (1966-1973)
• Alan Turing's Contributions (1950)
• The Logic Theorist (1955)
THE EARLY DAYS OF AI
• DENDRAL (1965)
2 SYMBOLIC AI & EARLY PROGRAMS (1952- 1966)
3 EXPERT SYSTEMS (1969-1986)
• XCON/R1 (1978)
• General Problem Solver
• Samuel's Checkers Playing Program (1959)
• MYCIN (1970s)
• Eliza (1964-1966)

LESSON 2: EMERGENCE OF MODERN AI (1987 - NOW)

Pro tip

There are quite a lot of concepts being introduced in this part. Even though we tried our best to demonstrate them using real life examples, there is still some math involved. Proceed with caution
(Spoiler alert, it will get worse in Part B)



I.Rise of Machine Learning and Deep Learning

1.Probabilistic Reasoning and Machine Learning (1987-present):

1.1 Probabilistic Reasoning and Bayesian networks:

In real life problems, very few things are actually certain. When you wake up in the morning and decide to go to class, on some days, you just go down the elevator peacefully, but on the other days, you trip and fall down the emergency exit stairs.

I.Rise of Machine Learning and Deep Learning

1.Probabilistic Reasoning and Machine Learning (1987-present)

1.1 Probabilistic Reasoning and Bayesian networks:

Things happen. Things happen, and AI also has to take that into account. That's why modern AI uses Probabilistic Reasoning to face real life problems - in order to take the action that has the highest probability to achieve its goal.

Let's go back to the falling down the stairs example. Now here's what we're trying to figure out: What's the probability of me going to class and falling down the stairs?

Now let's take this into consideration. There may be two reasons why you would fall down the stairs:

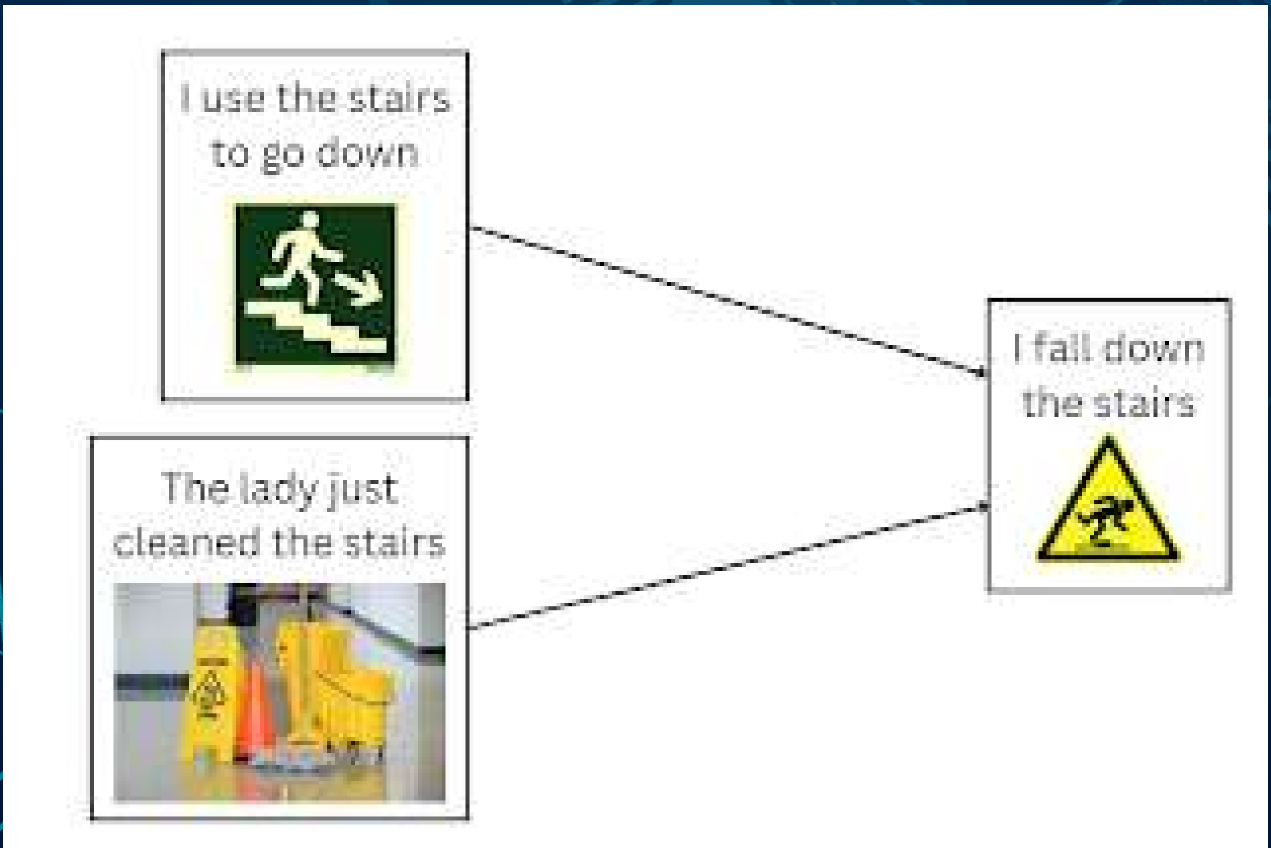
- **You use the stairs to go down, obviously**
- **The janitor lady just decided to wipe the stairs, making it super slippery.**

I.Rise of Machine Learning and Deep Learning

1.Probabilistic Reasoning and Machine Learning (1987-present)

1.1 Probabilistic Reasoning and Bayesian networks:

Let's draw a diagram to demonstrate this phenomenon. Because the probability of falling down the stairs depends on these two facts, we say, me falling down the stairs is conditionally dependent on me going down the stairs and the lady cleaning the stairs. Each of these rectangles in the diagram is called a node, and the arrows describe their probabilistic dependencies.

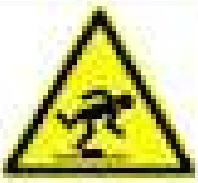


I.Rise of Machine Learning and Deep Learning

1.Probabilistic Reasoning and Machine Learning (1987-present)

1.1 Probabilistic Reasoning and Bayesian networks:

Now, supposedly, you don't use the stairs, then obviously, even if the lady wiped the stairs, you can't fall down the stairs, right? Now if we mark F as False, T as true, we would have a table like this:

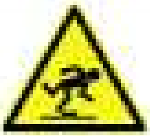
		
0%	F	F
0%	F	T

I.Rise of Machine Learning and Deep Learning

1.Probabilistic Reasoning and Machine Learning (1987-present)

1.1 Probabilistic Reasoning and Bayesian networks

Next, if we go down the stairs, there always is a small chance we trip and fall down the stairs. But of course, the chance itself increases significantly when the lady has wiped the floor beforehand.

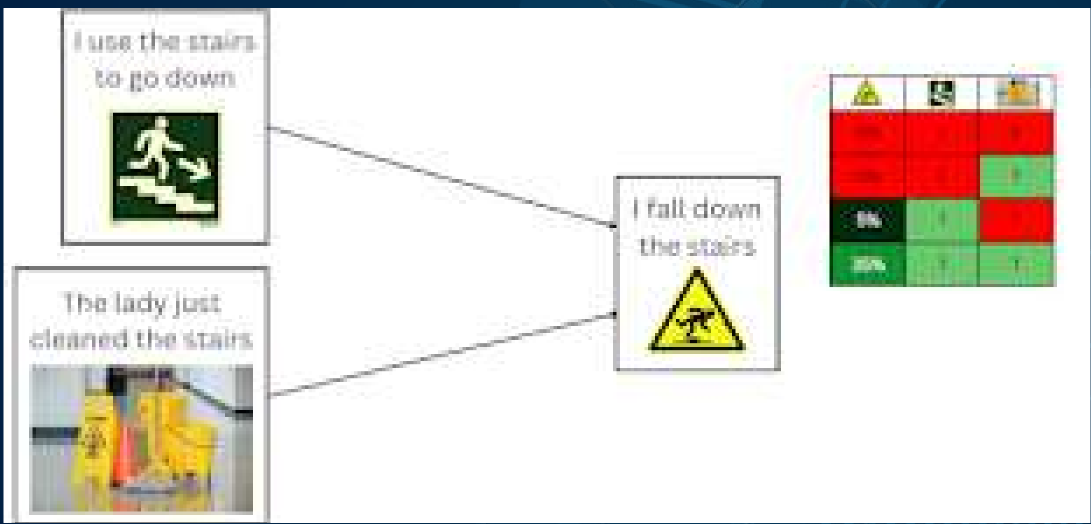
		
0%	F	F
0%	F	T
5%	T	F
35%	T	T

I.Rise of Machine Learning and Deep Learning

1.Probabilistic Reasoning and Machine Learning (1987-present)

1.1 Probabilistic Reasoning and Bayesian networks:

Now if we put that into the first diagram, it would look something like this



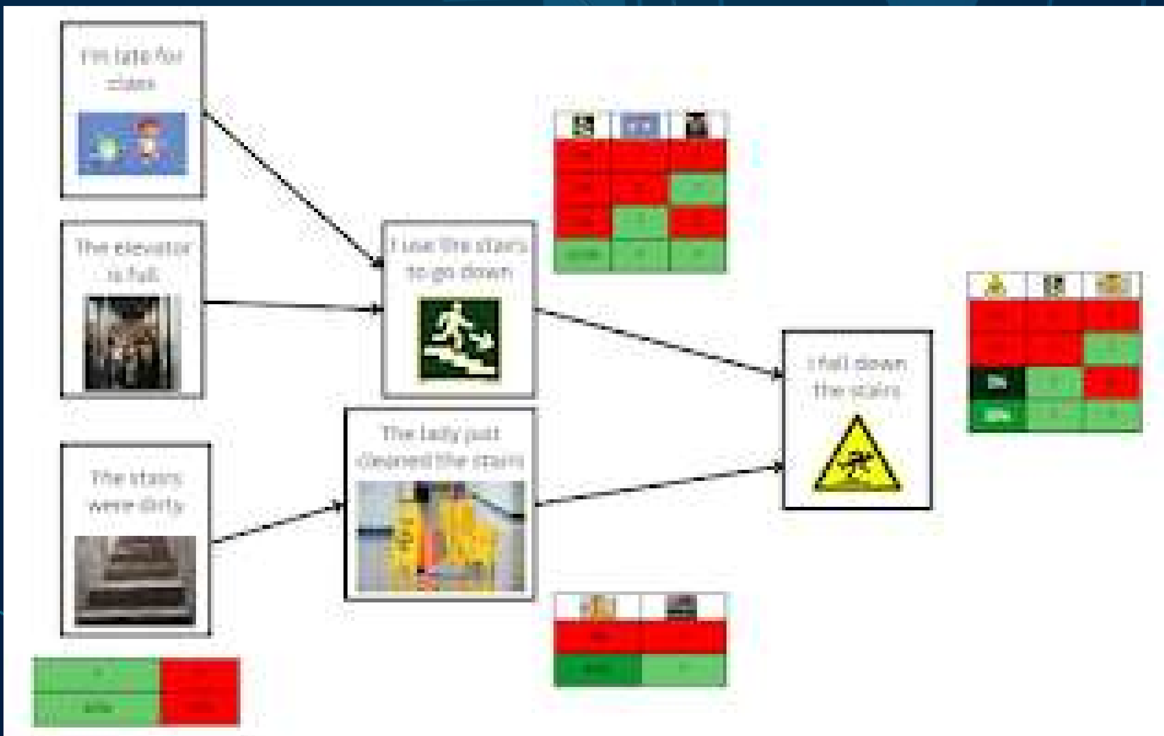
Now of course, this isn't enough to calculate the probability of me tripping and falling down the stairs. We also have to figure out the probabilities of each of the two nodes on the left. And think about this again, why would I use the stairs to go down? Well, there are, of course, 2 reasons: I'm late for class, and the elevator is full. Why would I be late for class? It's because I stayed up the other night, because I had to do the calculus homework, because the teacher decided to give us homework out of nowhere. The nodes will keep going on and on.

I.Rise of Machine Learning and Deep Learning

1.Probabilistic Reasoning and Machine Learning (1987-present)

1.1 Probabilistic Reasoning and Bayesian networks

Taking all of those probabilities together using some complicated probability formula, you will eventually be able to calculate the probability of you coming to class and falling down the stairs! Yay.



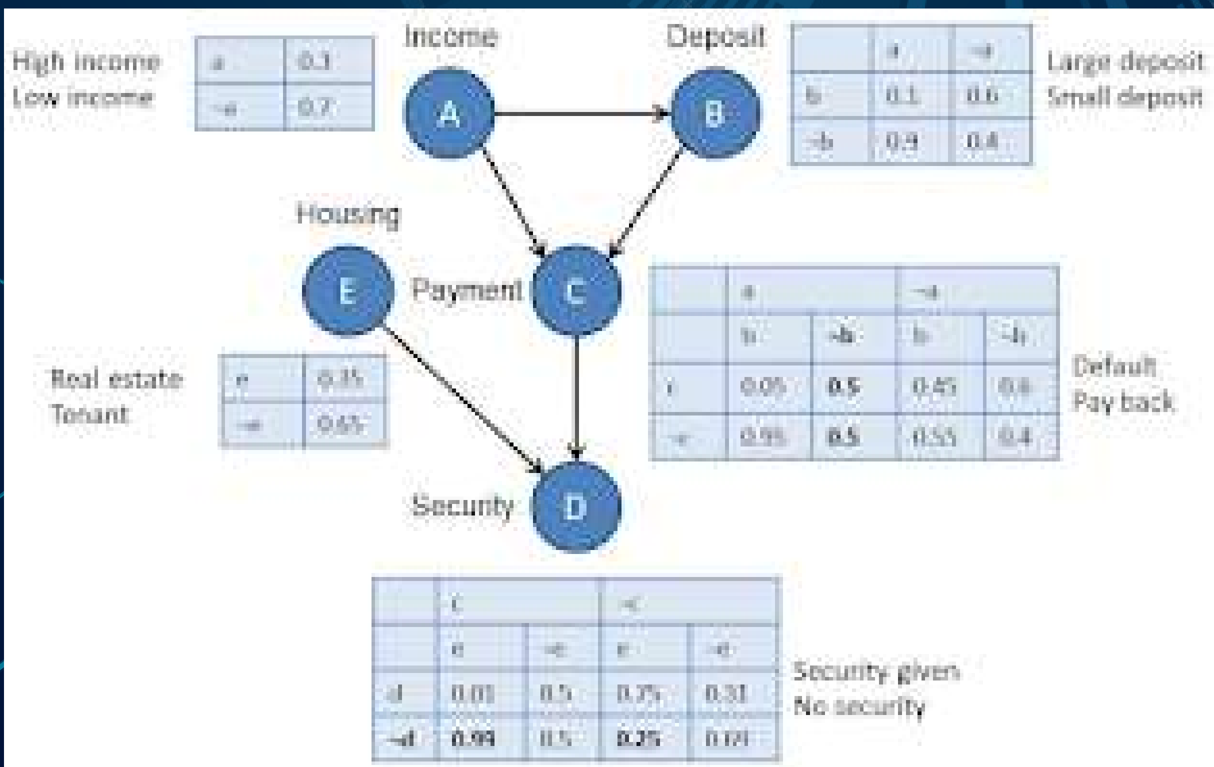
This diagram that we just drew represents random variables (nodes) and their conditional dependencies (arrows), and is called a Bayesian Network.

I.Rise of Machine Learning and Deep Learning

1.Probabilistic Reasoning and Machine Learning (1987-present)

1.1 Probabilistic Reasoning and Bayesian networks

Bayesian networks that are utilized in AI and Machine Learning and stuff are significantly more complicated than this diagram, as more and more aspects are taken into account in real life. Sometimes, the probabilities don't stop at a True or False statement. Take, for example, a middle school kid that just confessed to his crush - her answers may vary a lot: "Yes", "No", "We can still be friends", "I need time to think about it", "Don't talk to me ever again", or even an awkward silence. And yes, AI would have to take into account all that, requiring an insane amount of computational power.



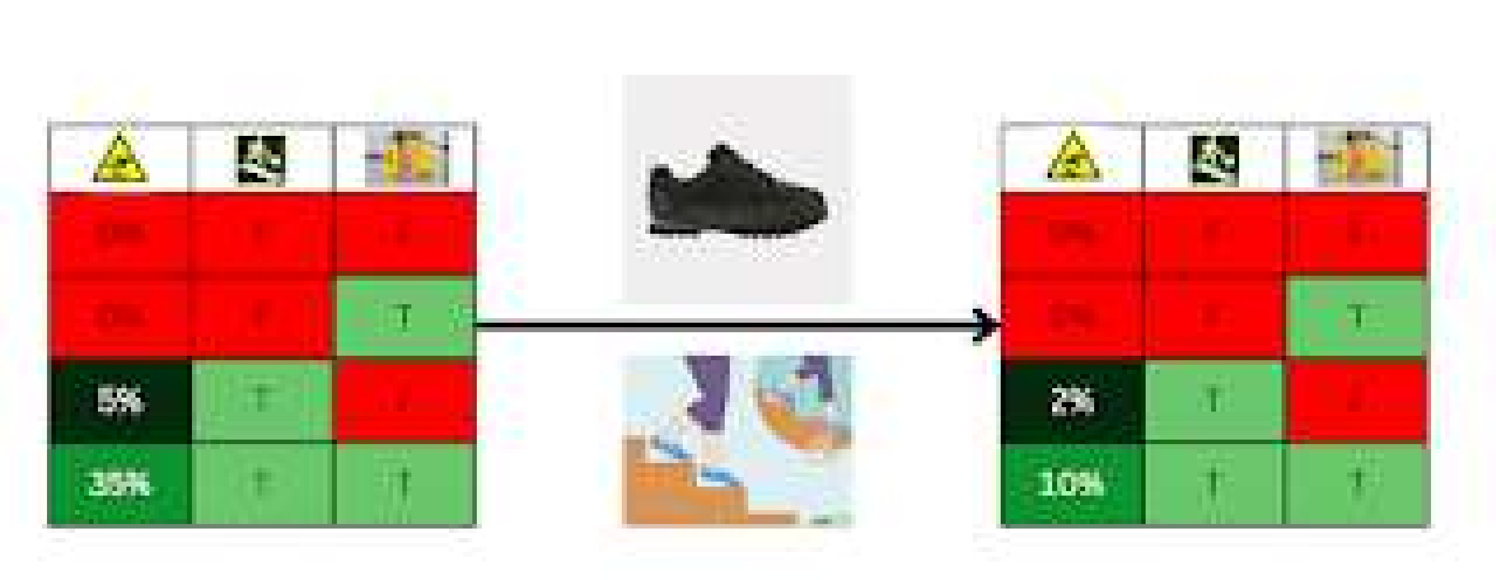
1.2 PROBABILISTIC MODELS IN MACHINE LEARNING

Machine learning refers to algorithms that can learn from a specific data set. Does that sound familiar? If you go back to Lesson 1, you'd realize that the 1959 checkers program in itself is a Machine Learning algorithm. However, the program itself depended heavily on absolute values, basically meaning it tried to look ahead to every possible move in order to assess which one is the best move. Implementing this kind of algorithm in solving real life problems is going to be quite problematic.

That's why Machine Learning algorithms to this day rely heavily on probabilistic models, which utilizes systems like the Bayesian system mentioned above, to make the best prediction. The approaches to these probabilistic models include:
- Maximum Likelihood Estimation (MLE): Basically, the algorithm finds the likeliest path of actions to achieve its goal using past data.
- Bayesian Inference: The algorithm combines its past observations and its current observations to make predictions. Or in other words, when the machine receives new data, it updates its old probability predictions.

1.2 PROBABILISTIC MODELS IN MACHINE LEARNING

It works the same way when you decide to buy shoes with better friction, and learn to be more careful when going down the stairs. When that happens, the calculated probabilities of you falling down the stairs will also decrease substantially.



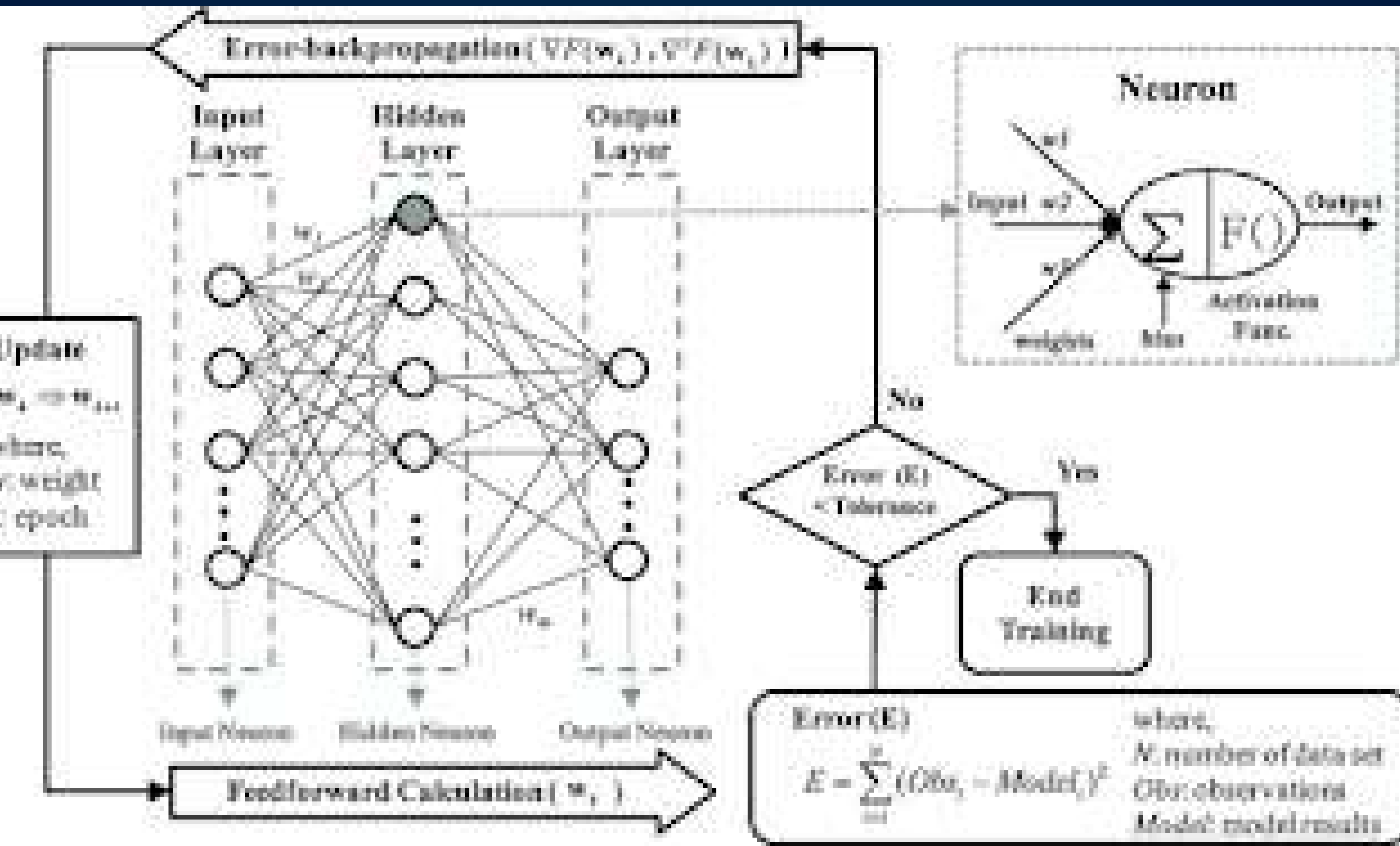
2. RETURN OF NEURAL NETWORKS (1986-PRESENT)

Remember the mathematical neuron made by those two nerds back in 1943? Well, at that time, the idea of artificial neural networks was only seen as a concept, because the amount of calculations and if else and all that is simply massive. Therefore, the concept of neural networks never got out of the hood.

Until 1986 that is. Then, computers have already improved by a ton. Particularly, the use of GPUs (Graphics Processing Units) in the 2000s allowed for training of much larger and deeper neural networks. This computational boost made it feasible to work with vast amounts of data, enabling significant advances in the field.

Another breakthrough that pushed neural networks to another level is backpropagation. Explaining backpropagation would unfortunately require a lot of explanation of how neural networks work, which involve a bunch of calculus, linear algebra and some other fancy mathematical things, weight, bias, cost function and all that stuff. This will be explained further in Part B: Behavioral AI.

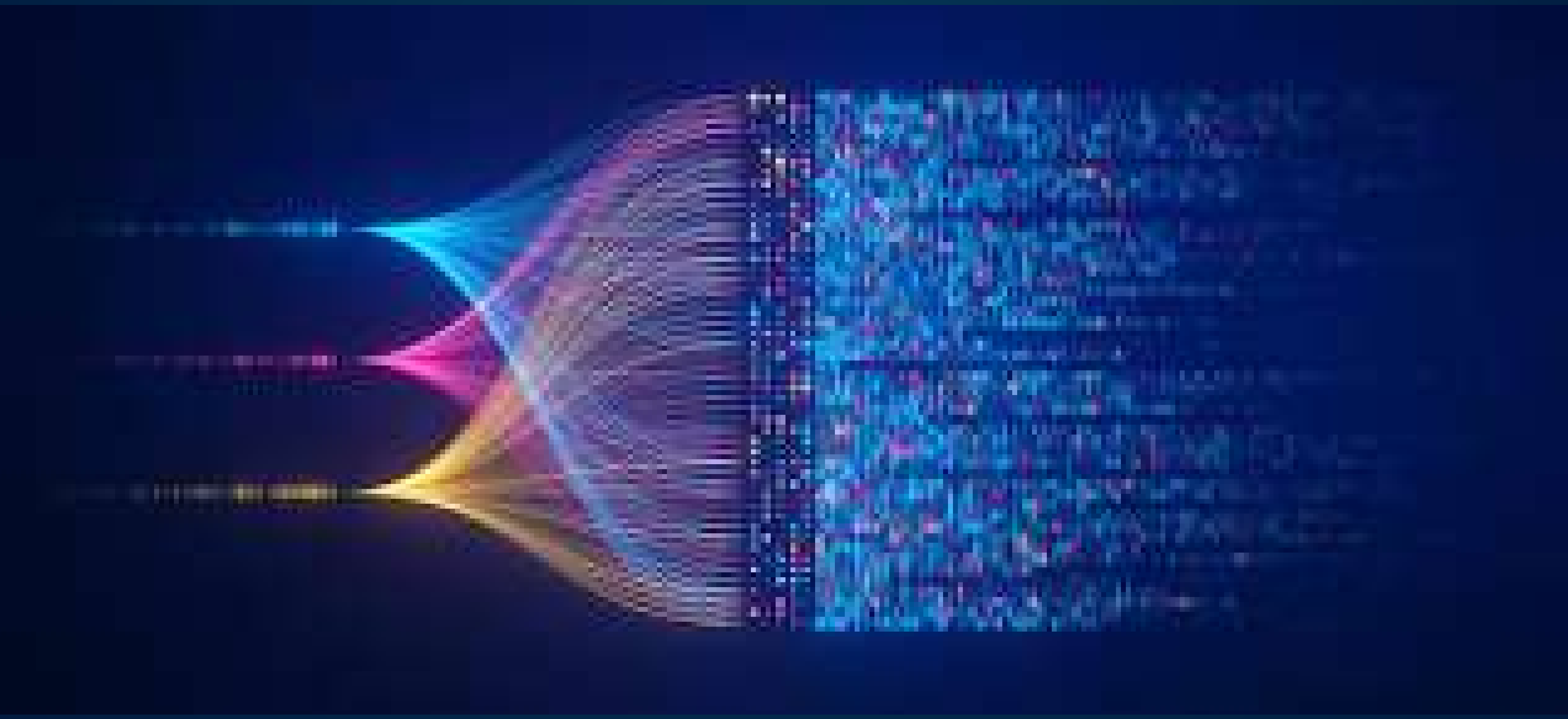
Anyways, with these developments in mind, the history of AI turned into a new page: a subset of machine learning focusing on neural networks - Deep learning.



3.Big Data (2001-present):

Big data refers to, well, big data. It's one of the most straight-forward concepts in the field of AI. Yes, it really is just some gigantic dataset that grows in size over time.

The thing that makes this such a huge thing for Machine Learning is that, again, they need to be trained on a specific dataset. Sometimes, the datasets themselves are even more important than the ML/DL algorithms. It's like a smart person getting influenced by bad people. No matter how smart they are, they will still likely be pulled into the bad stuff, like drugs and alcohol and all that.



The evolution of Big Data significantly accelerated the training rate of Machine Learning algorithms and Deep Learning algorithms. With this, people were able to create all sorts of different AIs that require a large amount of training, like AI chatbots (ChatGPT, Copilot,...) and Self-driving cars. //

II. Extra reading: Chess - Man vs Machines

Chess goes a long way. I'm sure everyone reading this part has played a game of chess before, or at least, heard of it. Well, if I have to ask: "Do you know how to play chess?", your answer would (most likely) be: "I can, but um, I'm really dumb at it". Even though the game has a relatively simple ruleset, it is still a ridiculously complex game with all sorts of calculations that require you to look several moves ahead, if not one, then two, three, five, sometimes even ten

To put into perspective how many possible positions there are in a chess game, after the 3rd move, there are already 9 million different positions. After the 4th, 288 billion different positions - That's already 36 times higher than the human population. It is estimated that a game with 40 moves (which is, in average, how long a chess game lasts) would have more possible positions than the number of atoms in the universe.

II. Extra reading: Chess - Man vs Machines

One day, one of the best chess grandmasters (also called GMs) in the past decided to start "The greatest game in the history of chess". It's a game between that guy and... well, the rest of the world. That's right, him against everyone else in the world, with all the other Grandmasters, with all the calculation tools and everything. The game was held online with the following format: The GM guy gets 12 hours to move, and the World team gets some other GMs to analyze the moves, and then post the moves up to a billboard for the smaller chess communities to vote on the best move, which takes another 36 hours. The game proceeded with one move per day, lasting 4 months. Yes, 4 months for a game of chess. For the whole duration, the game was analyzed and voted on by 50,000 people, including young talents, grandmasters, international Chess masters, even big chess organizations, coming from 75 countries. Nevertheless, after all that time and brainpower, and after 66 moves, the GM still emerged victorious.

The guy's name? Garry Kasparov.

II. Extra reading: Chess - Man vs Machines

That's why in 1997, it was such a big deal for Kasparov to be defeated 2 to 4 by an upgraded version of Deep Blue - a symbolic AI program that can evaluate up to 200 million potential chess moves per second, looking up to 12 moves ahead. Deep Blue's development actually began in 1985, when it also lost to Kasparov 2 to 4. After 10 years of development, the program finally became advanced enough to defeat the best human player in history.

Deep Blue beats chess champ

IBM's computer stages winning attack in first of six-game match

By The Associated Press

PHILADELPHIA — A chess computer turned retreat into a winning attack Saturday to defeat world champion Garry Kasparov in the first of a six-game match.

IBM's Deep Blue can master a move no human can accomplish: shifting through more than 200 million possible chess maneuvers in a second. The duel is the first to pit human against machine for a regulation, six-game chess match.

Kasparov ceded defeat on the 37th move when Deep Blue pinned his king between a knight and a rook.

Playing black, Kasparov was putting heat on Deep Blue's king in the 28th move when the computer managed to maneuver its way out of a defensive posture by capturing a key Kasparov pawn.

By the 29th move, Grandmasters Yasser Seirawan and Maurice Ashley were saying that Kasparov lost.

The glowering, 33-year-old Ukrainian champion left the game site in Philadelphia's Convention center without a word.

The three-hour defeat came much quicker than Deep Blue's design team had even dared to hope, IBM computer engineer Chung Jen Tang said after the match.

Kasparov's defeat came as a surprise to both chess grandmasters and IBM programmers, who thought Kasparov's aggressive attack would give him the day.

"I know my opponent is invincible, but I strongly believe that it's not invincible," Kasparov said before the match.

In 1989, he proved critics wrong by handily defeating Deep Thought, the IBM prototype for Deep Blue.

TODAY'S SCRIPTURE

"And God said, Let there be light: and there was light."

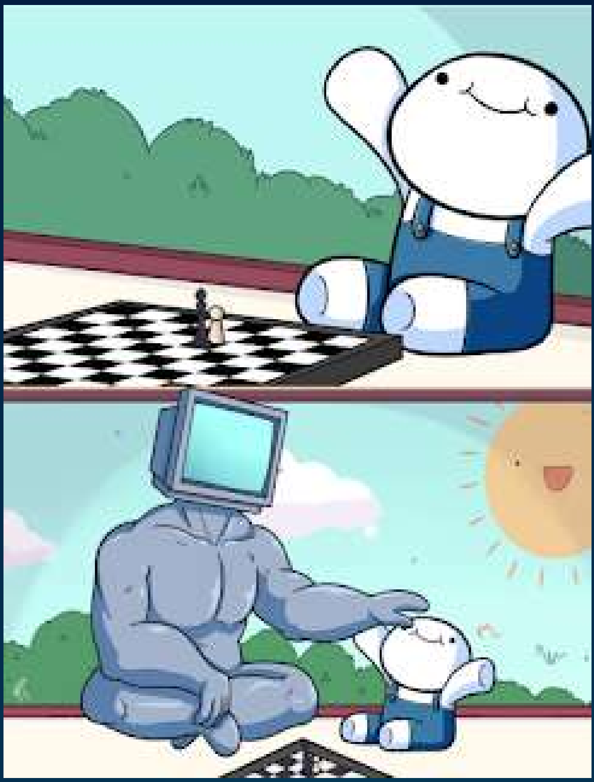
—Genesis 1:3

II. Extra reading: Chess - Man vs Machines

From that point onwards, chess analysis algorithms have developed at an insane rate, with numerous engines coming after Deep Blue, namely AlphaZero, Leela chess Zero or StockFish, which can now look up to 20 moves ahead. Nowadays, if you put a human and a machine against each other in a chess game, the human has already lost - even if you put the best human chess players into it.

II. Extra reading: Chess - Man vs Machines

And even if you win, the machine is going easy on you.



Therefore, chess engines nowadays are instead used to analyze human chess games, determining whether the human's move was bad, good, and show them how they can play better. Unlike the past, when chess playing machines were inferior to humans, machines are now teaching chess to humans. It really does show how much AI has developed over time.

III. EXERCISES

Ex1.Select the correct answer

1.Why does modern AI use probabilistic reasoning?
A) To ensure certain outcomes
B) To account for uncertainties in real-life problems
C) To predict emotions with high accuracy
D) To optimize machine learning algorithms
Answer: B) To account for uncertainties in real-life problems.

2.Which of the following would be a complex scenario for AI in probabilistic reasoning?
A) Predicting the outcome of a simple coin toss
B) Calculating the probability of falling given known factors
C) Processing multiple possible responses to a middle school kid's confession
D) Finding a route with the shortest distance
Answer: C) Processing multiple possible responses to a middle school kid's confession

3.What was a major factor in the resurgence of neural networks after 1986?
A) The development of rule-based AI
B) Increased computer processing power, especially with GPUs
C) Reduced interest in artificial intelligence
D) The decline of expert systems
Answer: B) Increased computer processing power, especially with GPUs

III. EXERCISES

Ex1.Select the correct answer

4.What is the primary purpose of "big data" in AI?
A) To limit the information processed by algorithms
B) To create massive rule-based systems
C) To provide extensive datasets for training machine learning models
D) To reduce computational power needed for AI
Answer: C) To provide extensive datasets for training machine learning models.

5.What analogy is used to describe the influence of datasets on machine learning algorithms?
A) A car gaining speed with more fuel
B) A smart person being influenced by bad people
C) A teacher guiding students
D) A river flowing to the ocean
Answer: B) A smart person being influenced by bad people.

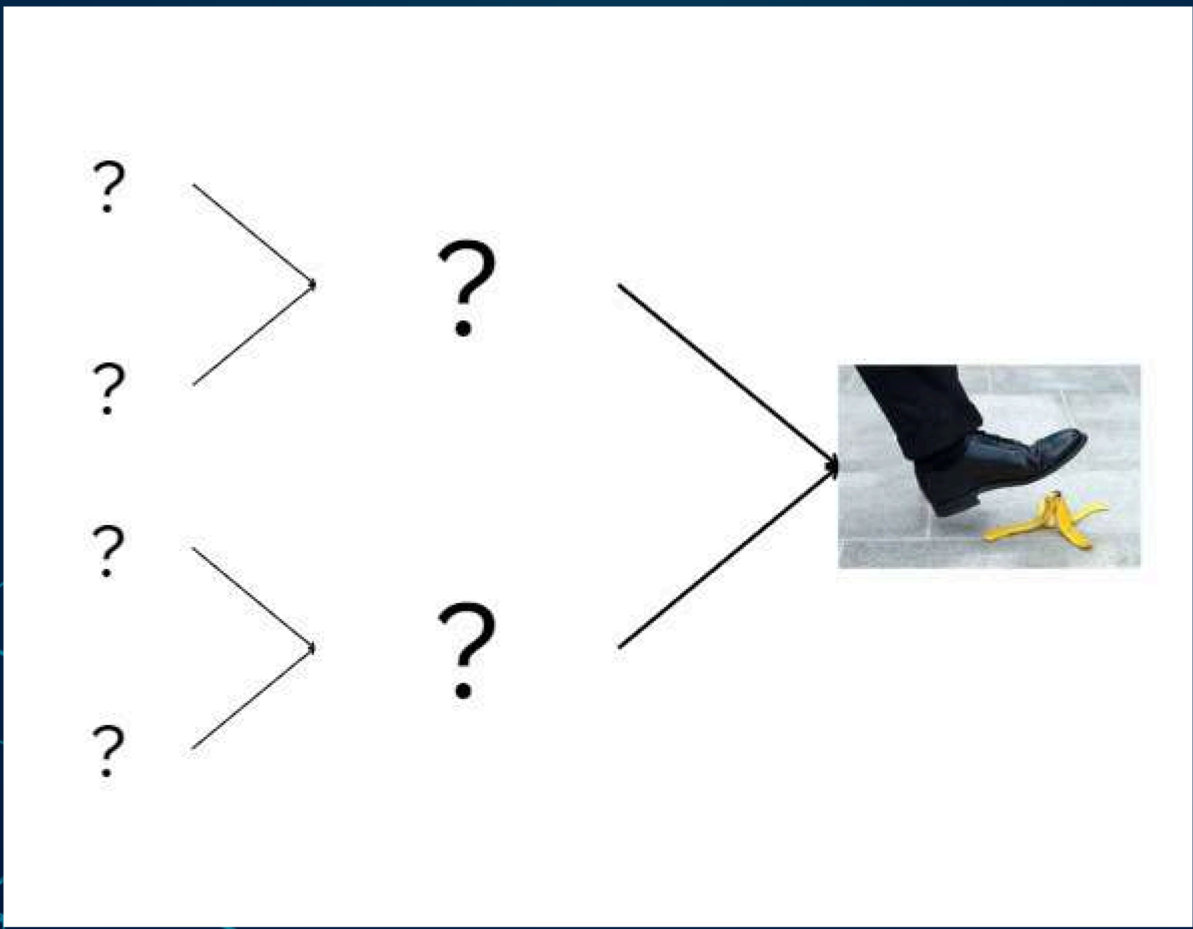
III. EXERCISES

Ex2.Fill in the blanks

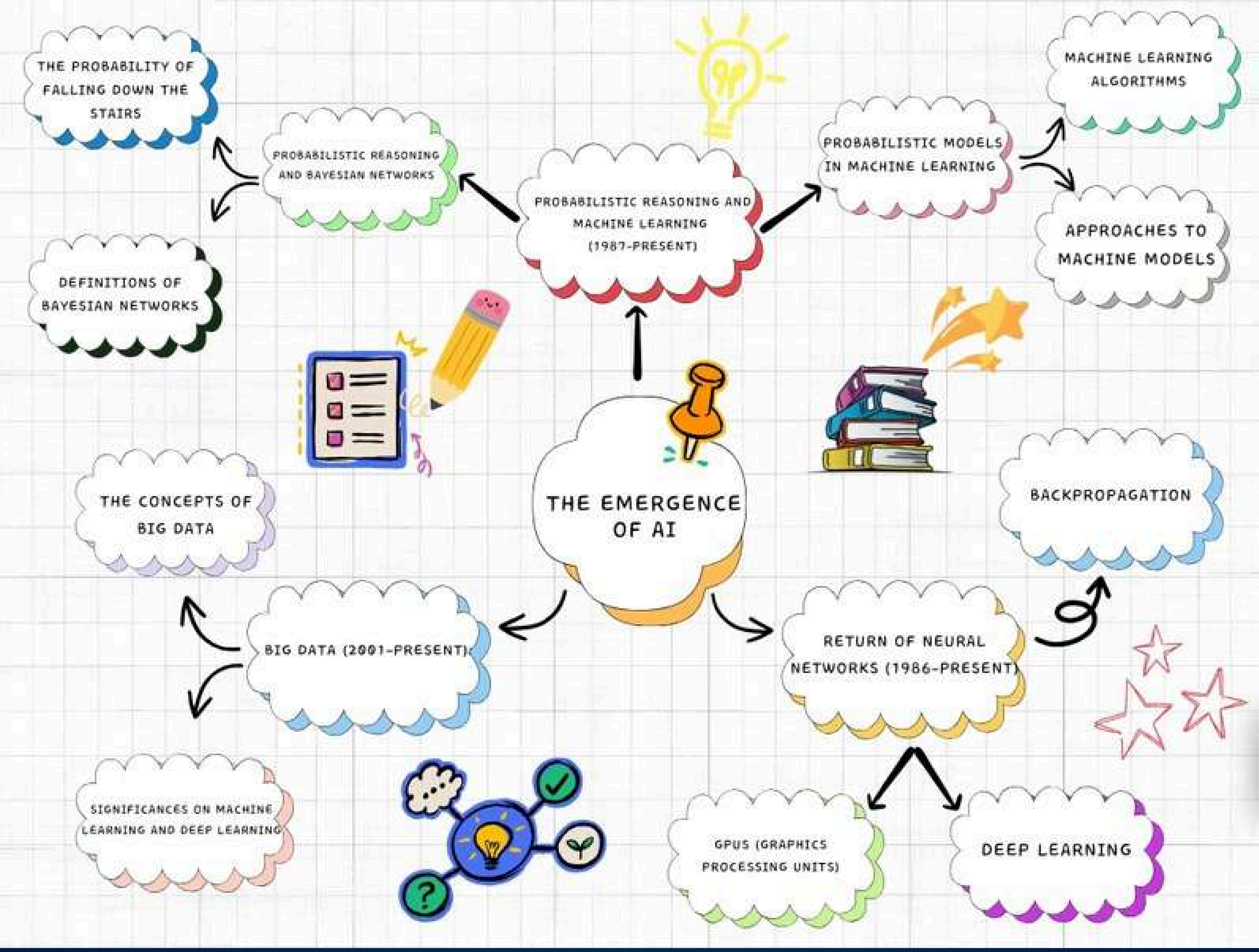
1. A __________ network is a diagram that shows random variables and their conditional dependencies.
Answer: Bayesian
2. __________ reasoning is used in AI to handle uncertainties in real-life scenarios.
Answer: Probabilistic
3.AI might struggle to process complex scenarios with multiple possible responses, such as a middle school kid's __________ to their crush.
Answer: confession
4.In AI, the quality of __________ can be as important as the algorithms used for machine learning.
Answer: datasets
5.The increase in the availability of big data has helped __________ learning models train faster and more accurately.
Answer: machine

III. EXERCISES

Side Quest: Draw a Bayesian network on a piece of paper demonstrating the probability of you stepping on a banana peel on the road. You get more points the more ridiculous nodes you manage to come up with ;>. Be creative!



IV. MIND MAP


THE EMERGENCE OF AI
PROBABILISTIC REASONING AND MACHINE LEARNING (1987-PRESENT)
PROBABILISTIC REASONING AND BAYESIAN NETWORKS
THE PROBABILITY OF FALLING DOWN THE STAIRS
DEFINITIONS OF BAYESIAN NETWORKS
PROBABILISTIC MODELS IN MACHINE LEARNING
MACHINE LEARNING ALGORITHMS
APPROACHES TO MACHINE MODELS
BIG DATA (2001-PRESENT)
THE CONCEPTS OF BIG DATA
SIGNIFICANCES ON MACHINE LEARNING AND DEEP LEARNING
RETURN OF NEURAL NETWORKS (1986-PRESENT)
BACKPROPAGATION
GPUS (GRAPHICS PROCESSING UNITS)
DEEP LEARNING

Lesson 3: Future of AI

Pro tip

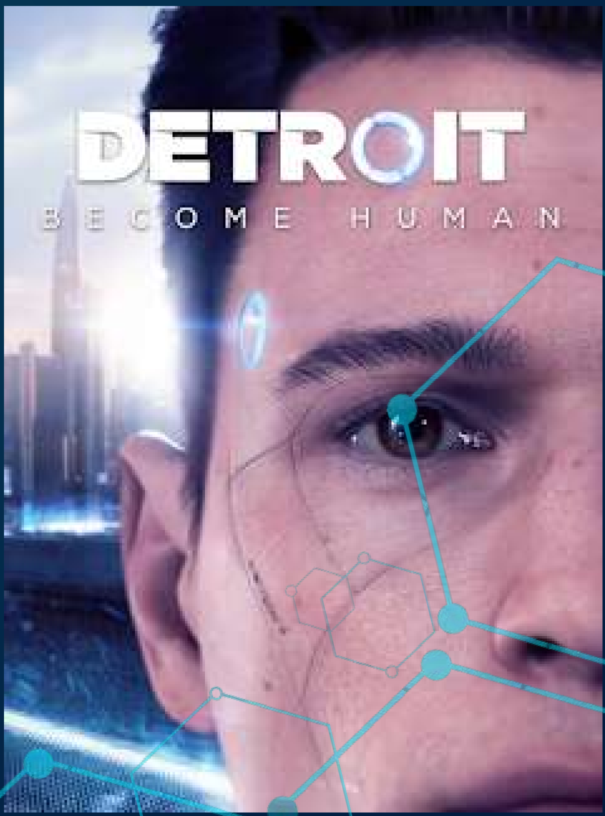
This lesson is not too important, and is kind of considered extra reading

I.General trends

The AI we know today is already being developed at a ridiculous speed. Comparing the DeepMind AI (which beat every one of the 57 Atari games 10 years ago) to Inception 2.5 (an AI model that rivals GPT-4 and Google's Gemini), our current model has 5,000,000,000 more computational power than the model 10 years ago. That's 10 times more computational power than the previous year, for 10 years in a row. Just imagine how much AI will have improved in the future, if we keep following this trend.

II.ARTIFICIAL GENERAL INTELLIGENCE (AGI)

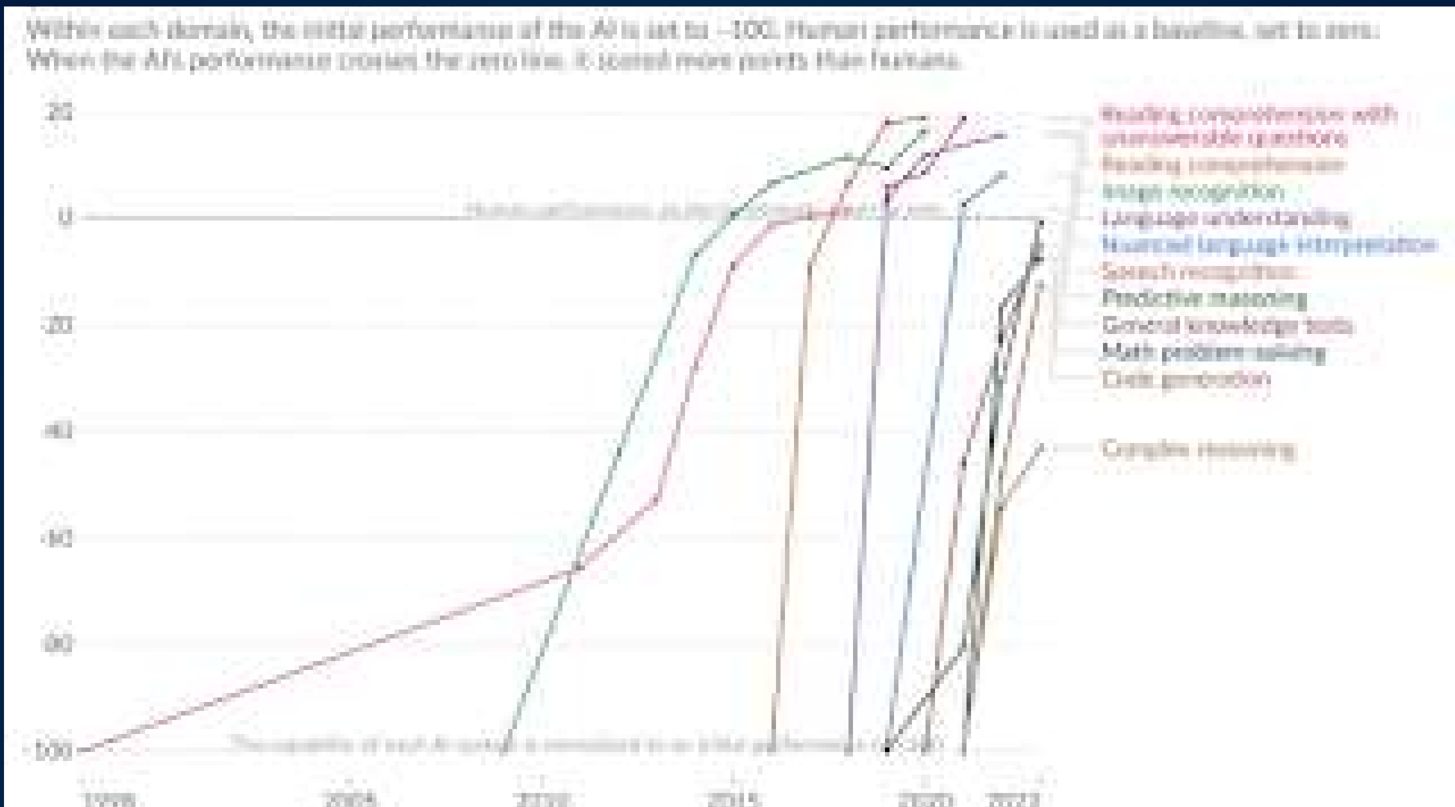
Currently, the AI stuff that's around us, ChatGPT, video enhancer AIs, sound recognition AIs, Siri, Deep Blue, or Tesla's self-driving cars, are all referred to as Narrow AI, which is the type of AI that can only perform one specific task. Our current goal right now is to move from Narrow AI to Artificial General Intelligence (AGI), which is a type of AI that's basically a human: it can physically walk, it can move objects, it can talk, it can observe, and it can learn. An AGI would look and act something like a character in Detroit: Become Human. Some types of AGI will even be able to understand human emotions, and express them themselves.



Big AI companies like Google's DeepMind and Elon Musk's xAI claim that we may have a complete AGI by around 2028 and 2029. However, "practical" uses of AGI may need significantly more years, even decades.

III.ARTIFICIAL SUPERINTELLIGENCE (ASI)

- **ASI refers to a hypothetical where Artificial Intelligence becomes even smarter than every human in every aspect. It's kind of the same case as the Deep Blue chess engine mentioned above, except, well, it can do quite literally everything better than humans.**



- **As of 2023, computers have already surpassed humans in certain skills, like language understanding and image recognition, and it's catching up quite quickly on the other skills as well. This is why many scientists believe ASI will be possible very shortly after AGI gets developed.**

IV. AI SINGULARITY

Artificial Intelligence's growth rate extends far beyond that of human being's intelligence, grows out of control, and leads to unthinkable consequences. Most likely, the machine becomes so intelligent that we ourselves can't even imagine what they would look like yet. But if I have to imagine, they would probably enslave humans.

But of course, that's like the very very very very far future. If you're reading this right now, you will probably not witness any of this AI-Enslave-Humanity stuff. But in the short future, as we improve AI's speed, accuracy and efficiency , we will also see benefits in many many different fields such as healthcare, transportation, cybersecurity, education or (especially) robotics. So for now, just know that it's probably going to be good for us. HeheYeah. It will probably benefit us.

V. EXERCISES

Fill in the blanks

1. The goal of moving from Narrow AI to Artificial General Intelligence (AGI) is to create an AI that can __________, __________, and __________ similar to a human.
walk, move objects, and talk

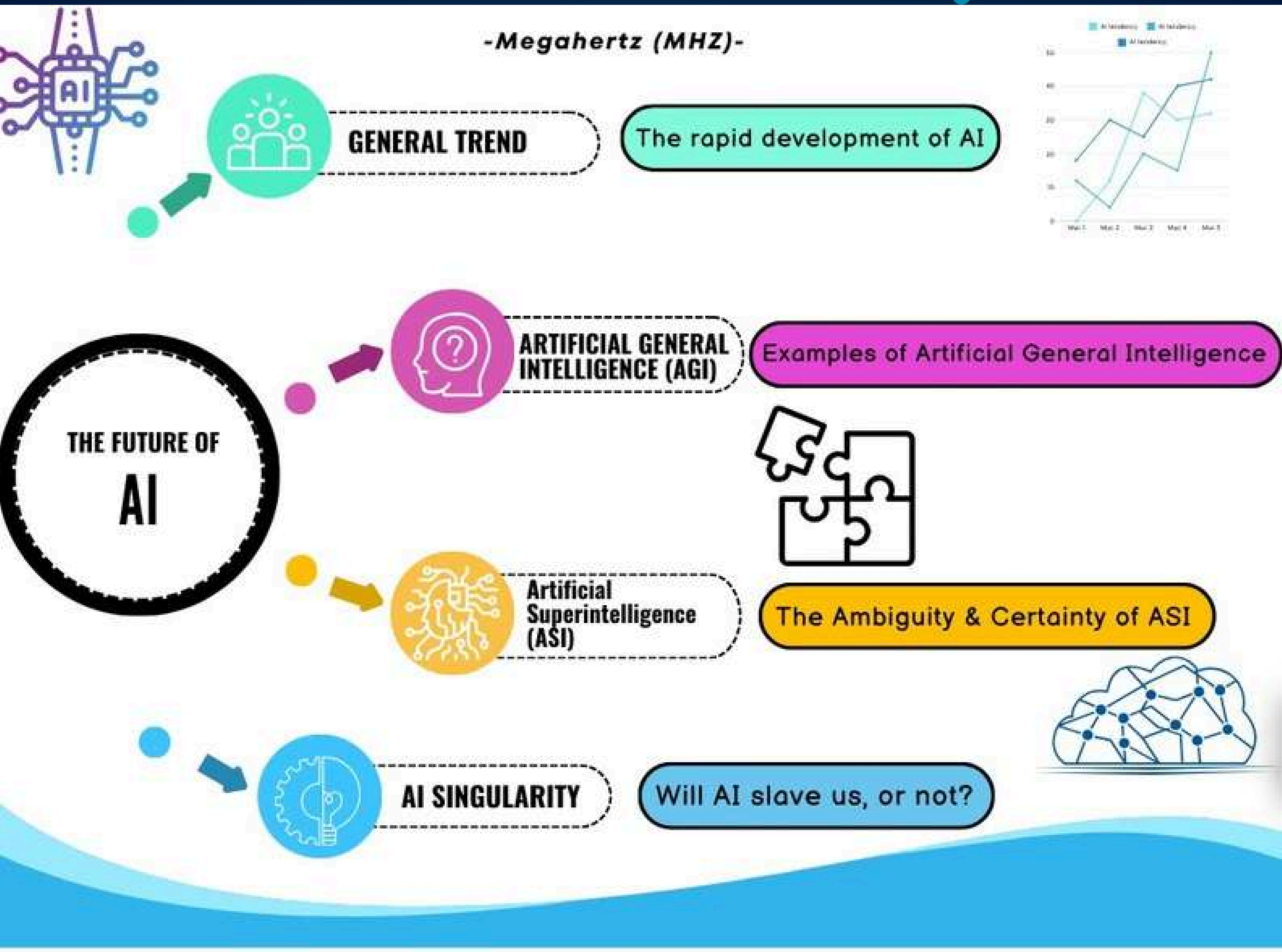
2. Companies like Google's DeepMind and Elon Musk's xAI predict that a complete AGI may be developed around the years __________ and __________.
2028 and 2029

3. Artificial Superintelligence (ASI) is a hypothetical scenario in which AI becomes __________ than humans in __________ aspects.
smarter, every

4. The rapid growth of AI, potentially surpassing human intelligence and control, is referred to as the __________.
AI singularity

5. In the near future, advancements in AI are expected to bring benefits in fields like __________, __________, and __________.
healthcare, transportation, and cybersecurity

IV. MIND MAP


-Megahertz (MHZ)-
AI
GENERAL TREND
The rapid development of AI
THE FUTURE OF
AI
ARTIFICIAL GENERAL INTELLIGENCE (AGI)
Examples of Artificial General Intelligence
Artificial Superintelligence (ASI)
The Ambiguity & Certainty of ASI
AI SINGULARITY
Will AI slave us, or not?

CHAPTER B: BEHAVIORIST AI

CHAPTER B: BEHAVIORIST AI

Pro Tip:

This is the part where it starts getting technical, although the contents in this book are still quite theoretical.

While it is nice to take note of the fundamentals, some people may be more inclined to learn by doing rather than just reading. In that case, if you happen to have some knowledge in python, you can also look at Pytorch, a deep learning framework that uses Python, as well as some Github Repos related to this framework.

https://github.com/gordicaleksa/pytorch-neural-style-transfer

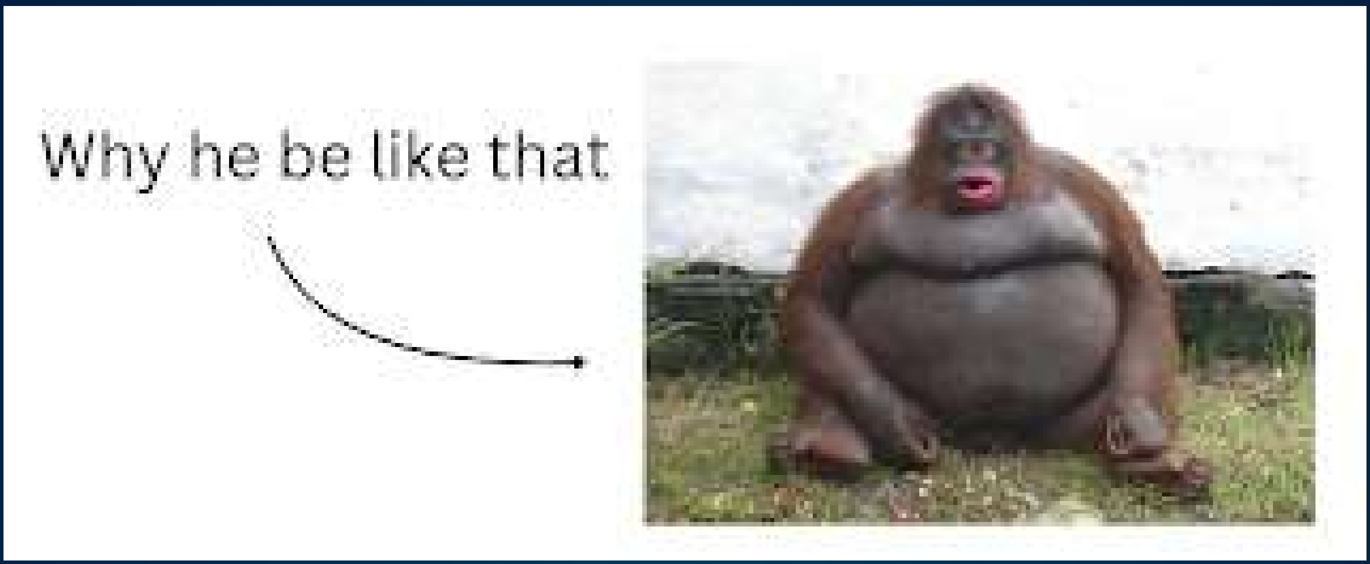
One of my favorite ones ^

LESSON 1: UNDERSTANDING BEHAVIORIST AI

I.Machines with behaviors?

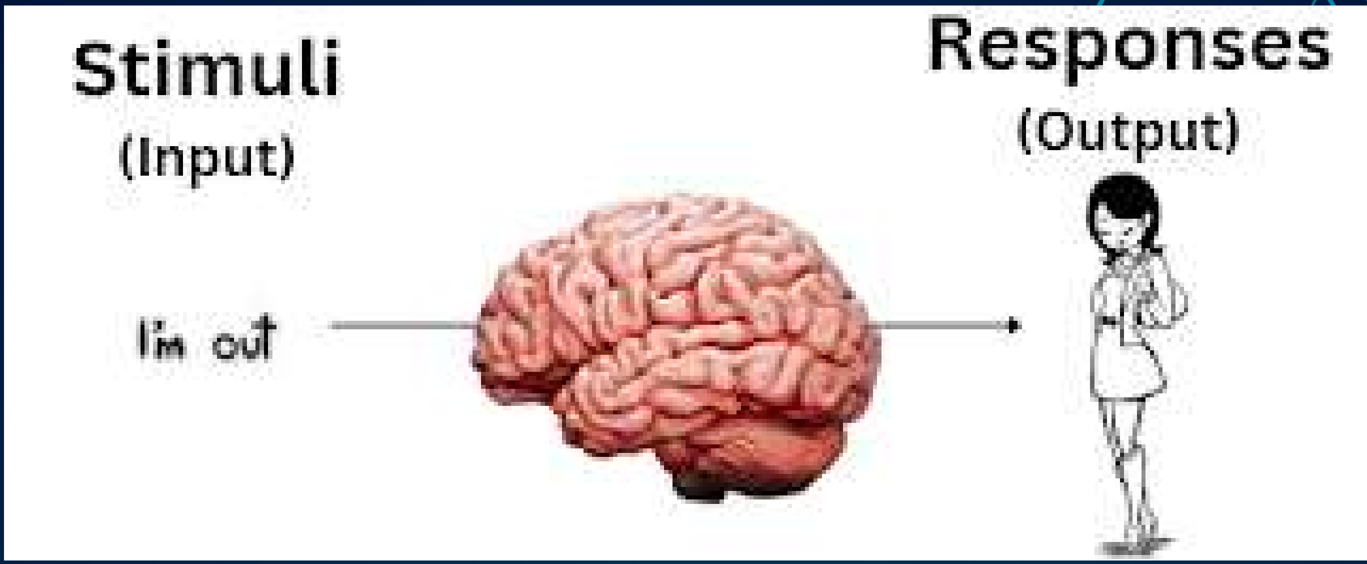
Before we dive into the definition of the Behaviorist AI, let's think for a bit - what is behavior? Behavior is the way we speak, the way we walk, the way we observe - or to be simple - the way we act.

But what are those actions based off of?

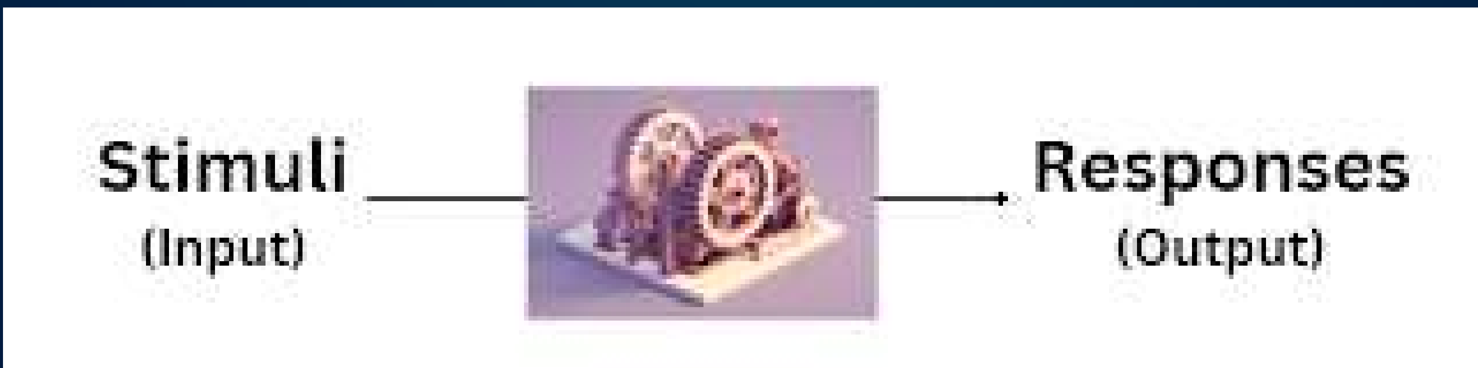


Notice how we all behave differently in different places and different situations: outdoors and indoors, in class and after class, during a meeting and during work

I.Machines with behaviors?



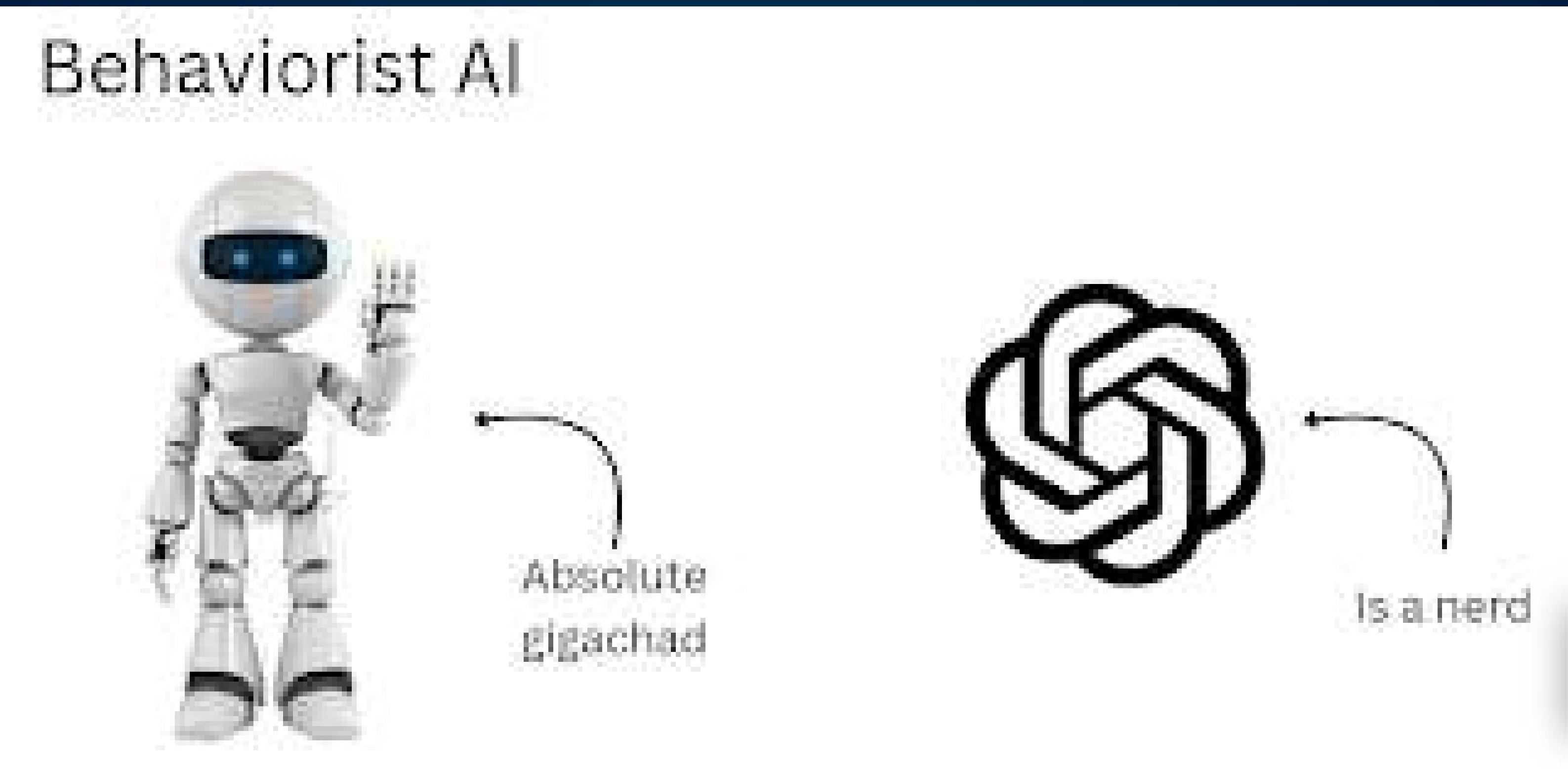
This is all possible because our brain adjusts our behavior according to the observed surrounding environment. Or in other words, as we receive different stimuli from our environment, we also respond to it in different ways.



That's why the artificial scientists looked at this phenomenon and were like: "Hmm, what if machines can do it as well?" And hence, Behaviorist AI was born: a branch of AI exploring how the computer interacts with its environment

I.Machines with behaviors?

It's an interesting branch of AI, because Behaviorist AI actually focuses on predicting actions, not just thinking. And, different compared to the other AI models like chatGPT - it can actually learn from what it does, rather than from the data that we feed to it.



With this in mind, this type of AI has been used, and will be used by humans in numerous different tasks that require behavioral analysis and behavioral actions.

II.WHAT CAN BEHAVIORAL AI DO?

1.Robotics

Well, if there's actions and behaviors involved, it has to do something with Robotics.

AI in robotics is one of the most groundbreaking technological advancements ever, revolutionizing how robots perform tasks. Today's robots, using Behaviorist AI, can observe its environment, learn from experiences, reason, and make decisions. These capabilities significantly enhance their effectiveness in fields like manufacturing, healthcare and transportation.

II.WHAT CAN BEHAVIORAL AI DO?

2.Cybersecurity

The final big thing that Behaviorist AI can do is in cybersecurity, or in other words detecting anomalies in user activity. This can include phishing, scams, and data thefts.

By utilizing Behavioral AI in cybersecurity, not only will it significantly improve the anomaly detecting efficiency (especially in large scale websites), but it will also reduce the costs required to implement this.

3.Game playing

For the fellow gamers who can't buy robot parts, Behaviorist AI can also be applied in playing video games. And they can get crazy good at it.

In fact, Google recently launched an AI called SIMA (Scalable Instructable Multiworld Agent) that can learn how to play open world games that it has never seen before. This includes No man's sky, Teardown, and Goat simulator 3.

Funny enough, AI still struggles when playing Minecraft, because the game itself doesn't have a definite objective: some players may want to build cute looking houses, while others want the goodies - diamonds, golden apples, totem of undying etc. Because the objective of the game is based on the player, the artificial intelligence required to learn how to play the game would have to be something close to an AGI - which is far from now.

Who knows? Maybe you will be the first ever person to ever create a Minecraft playing AI.

III. EXERCISES:

Ex1. Determine whether the following statement(s) are True or False:

1.Behaviorist AI focuses on predicting actions, not just on thinking.
•Answer: True

2.Behaviorist AI primarily learns from the data that humans feed it, rather than from its own actions.
•Answer: False (It learns from its actions and interactions with the environment)

3.Behaviorist AI can be used in fields like robotics, cybersecurity, and gaming.
•Answer: True

4.Behaviorist AI is similar to AI models like ChatGPT, which process pre-fed data to generate responses.
•Answer: False (Behaviorist AI interacts with and learns from its environment)

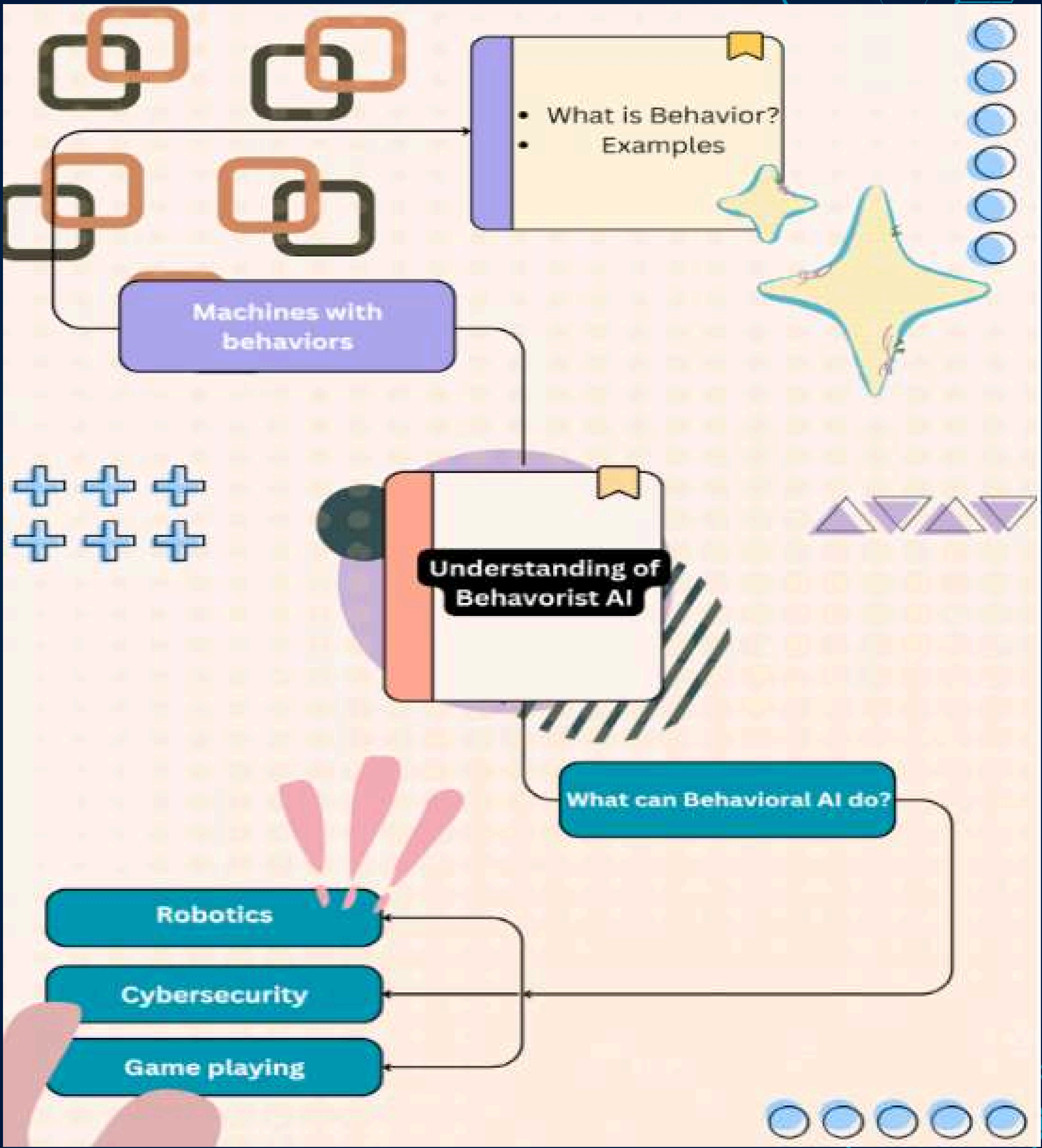
5
.One application of Behaviorist AI in cybersecurity is detecting anomalies in user activities, such as phishing and scams.
•Answer: True

6.Google's AI, SIMA, was designed to play games with clear objectives only, like Minecraft.
•Answer: False (SIMA can play open-world games like No Man's Sky and Goat Simulator 3)

7.In gaming, Behaviorist AI struggles with games like Minecraft because the game lacks a definite objective.
•Answer: True

8.Behaviorist AI's effectiveness in robotics has greatly improved fields like manufacturing, healthcare, and transportation.
Answer: True

IV. MIND MAP



What is Behavior?
Examples
Machines with behaviors
Understanding of Behavorist AI
What can Behavioral AI do?
Robotics
Cybersecurity
Game playing

LESSON 2: REINFORCEMENT LEARNING (RL)

I. Trial and Error

Trial and Error (noun): a way of achieving an aim or solving a problem by trying a number of different methods and learning from the mistakes that you make

If you're reading this, get out of your chair, stand up, and take a walk around the room.

Yes, that's right. stand up. Nice. You're killing it. Keep walking. Well, if you can't walk around the room, just picture that you're actually doing it. Very nice. You can sit down now.
It's a fact that if your legs are working, we all know to walk, right? We do it every day, getting out of bed, going to class, going back home, going to the toilet. It's definitely nothing new to us as humans.

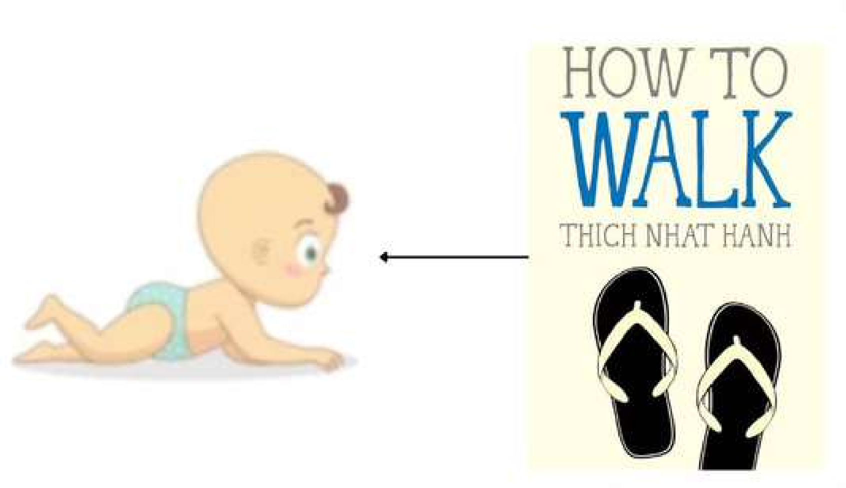
LESSON 2: REINFORCEMENT LEARNING (RL)

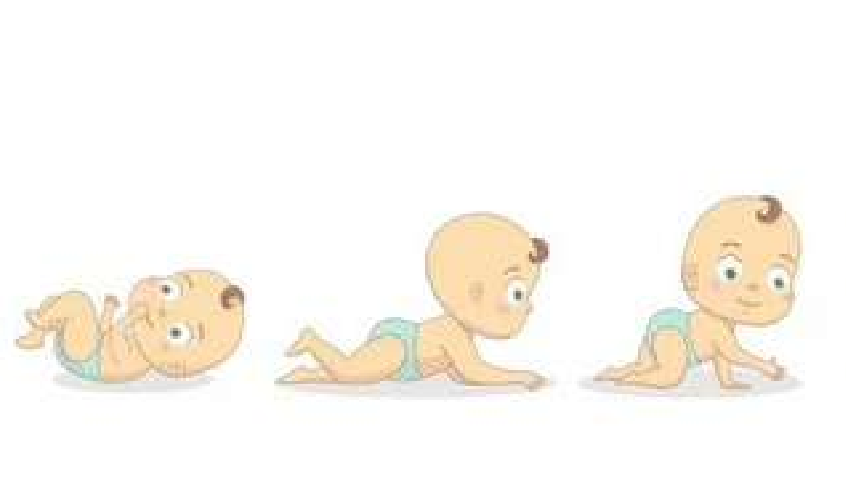
But what if, after you sat down, and you saw a 6 month old toddler next to you, how would you explain how to walk to them?

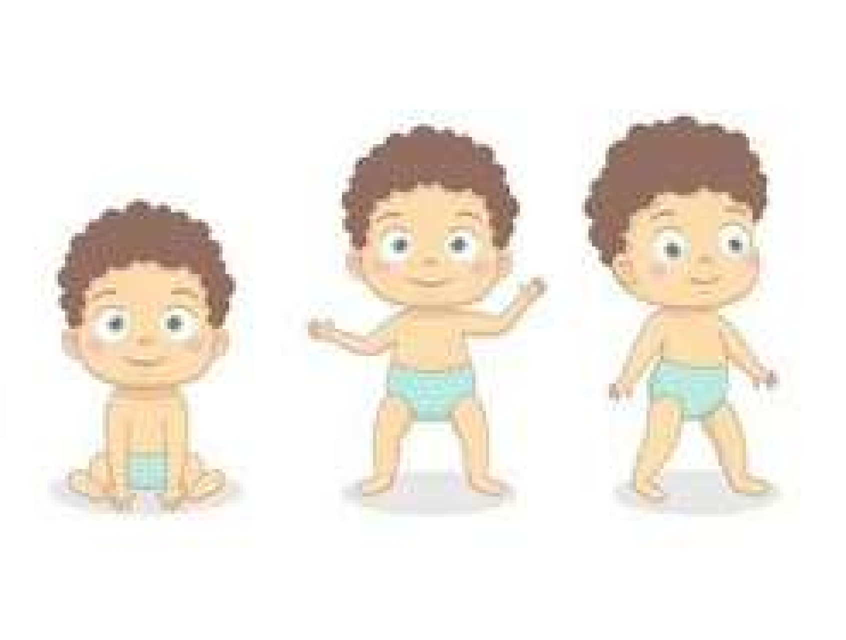
Surely, you're not going to give them a book of instructions on how to walk, right?

That's the thing - there are a lot of things that we know how to do, but don't know how to explain. Like, to do that with walking, you'll need to explain stuff like leg angle and movement and some nerdy stuff.

Instead, the baby has to go through trial and error stuff: trying to walk, falling, adjusting himself, trying to walk again, and falling again. Eventually, he will be able to walk like all of us right now.

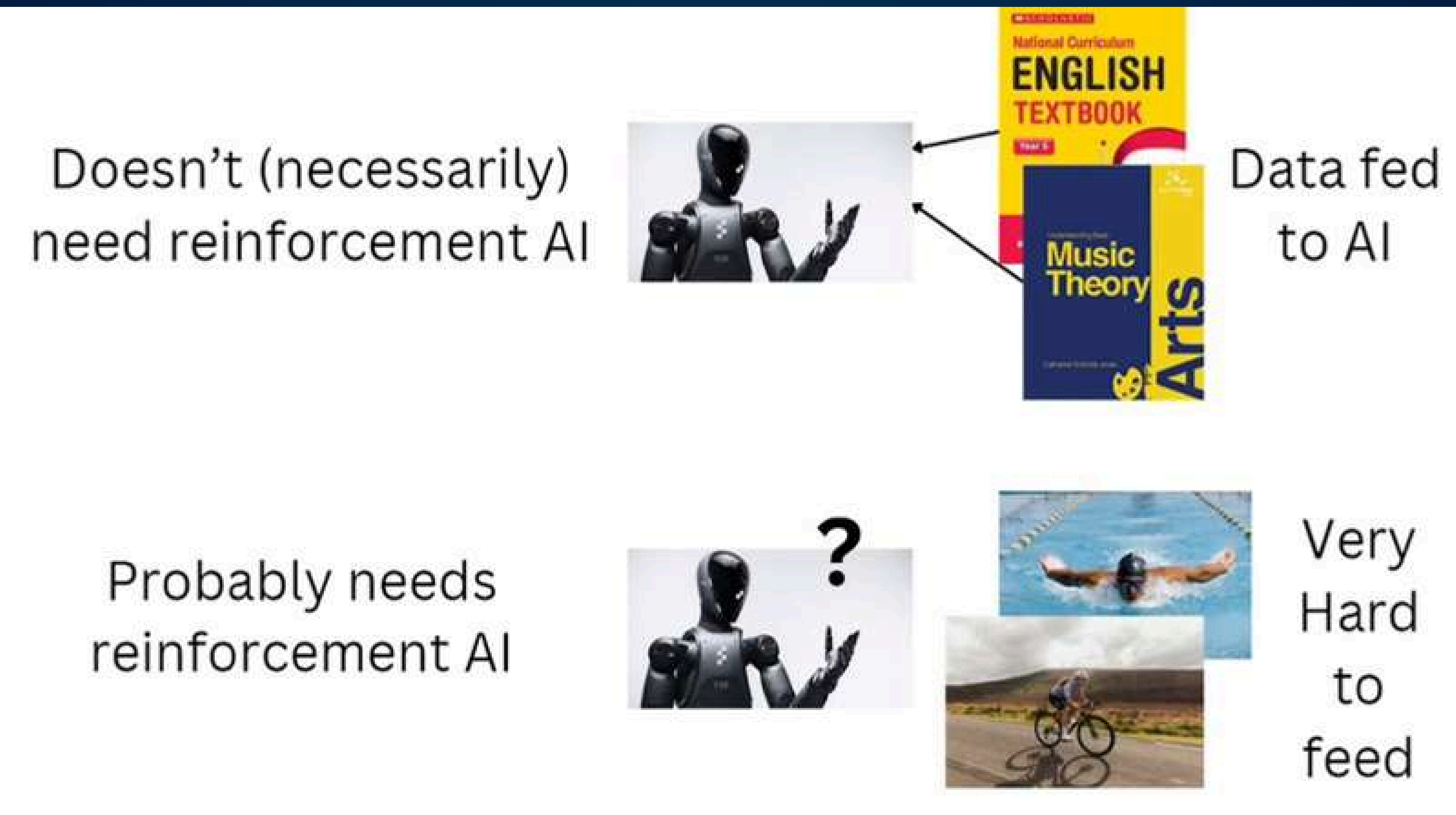






LESSON 2: REINFORCEMENT LEARNING (RL)

And, it’s the same thing for machines. If we want to teach machines how to walk, we can’t just feed the machine training data, the way you teach text and image recognition programs. You have to make the machine do, make mistakes, and then learn from what it did wrong.



This is what reinforcement learning is for: Teaching machines things that are hard to be explained by humans. Notice how for text and image recognition AI, there is always a human book explaining how to do it. However, there usually aren't any books explaining how to do stuff like riding a bike or swimming or something. It opens up the door for a lot of different AI tasks that weren't possible before.

II. COMPONENTS OF AN RL PROGRAM

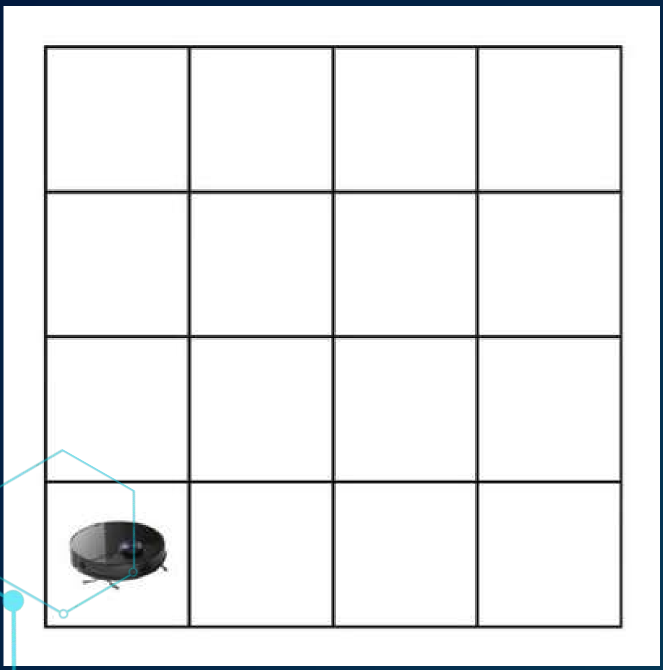
Quite a good way to demonstrate a Reinforcement Learning AI program is with a path-navigating problem, which involves 3 main components:

1. AGENT

In this case, our agent, or decision-maker, is going to be a vacuum robot . This kind of vacuum has been around for quite a while now, and is used to automatically detect debris and navigate around the house.

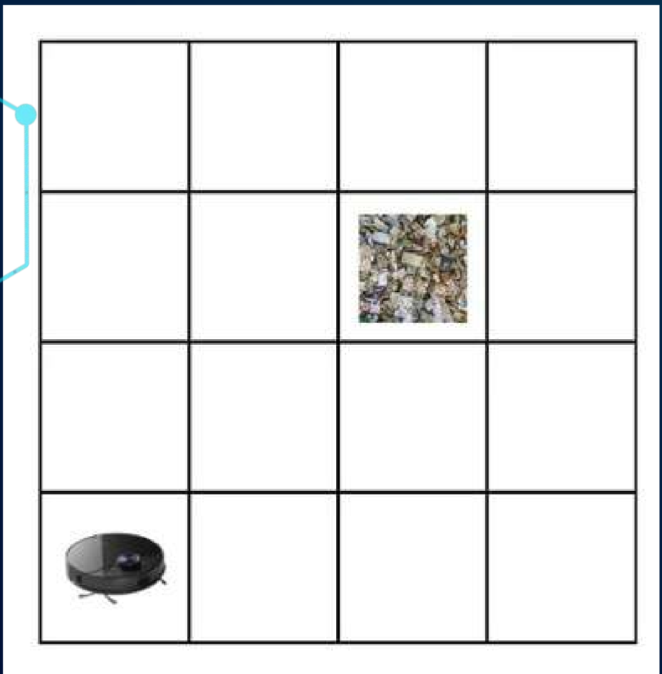
II. COMPONENTS OF AN RL PROGRAM

2.ENVIRONMENT



Our environment for this example will be a simple 4x4 grid like this. Our agent, the robotic vacuum, will be placed at the bottom left.

3.REWARDS/PUNISHMENTS



As a vacuum cleaner, our robot will try to arrive at the place with trash, which we will scatter some in the middle. This will be the reward of our algorithm.

When our vacuum cleaner collects this trash, it would basically say:

That's all we need to know for now. Next lesson, we'll discover further on how this process works by learning about a crucial concept about Reinforcement Learning: Markov Decision Process.

III. EXERCISES

Ex1. True/False Questions

1.The process of learning by trying different methods and adjusting based on mistakes is called _______________.

- Answer: trial and error

2.In reinforcement learning, the main goal is to teach machines tasks that are hard to _______________ by humans.

- Answer: explain

3.In the vacuum cleaner example, the _______________ is the component that makes decisions based on rewards and punishments.

- Answer: agent

4.The robotic vacuum cleaner's _______________ in the example is a 4x4 grid where it navigates to clean up trash.

- Answer: environment

5.The _______________ of the vacuum cleaner in the RL program is to find and collect trash in the environment.

- Answer: reward

III. EXERCISES

Ex2. Multiple Choice Questions

1.What does trial and error involve?

- **A) Making one attempt and stopping**
- **B) Trying different methods and learning from mistakes**
- **C) Following strict instructions**
- **D) Copying exactly from other examples**
- **Answer: B) Trying different methods and learning from mistakes**

2.In reinforcement learning, what is the purpose of rewards?

- **A) To encourage the agent to explore new environments**
- **B) To improve human instructions**
- **C) To guide the agent's behavior toward the correct goal**
- **D) To avoid all mistakes**
- **Answer: C) To guide the agent's behavior toward the correct goal**

3.In the RL example provided, which of the following is the agent?

- **A) The 4x4 grid**
- **B) The robotic vacuum cleaner**
- **C) The trash**
- **D) The floor**
- **Answer: B) The robotic vacuum cleaner**

4.Which of the following tasks is typically taught using reinforcement learning?

- **A) Text translation**
- **B) Image recognition**
- **C) Playing sports like soccer**
- **D) Checking grammar**
- **Answer: C) Playing sports like soccer**

5.Which concept is often introduced as a next step in understanding reinforcement learning?

- **A) Neural networks**
- **B) Markov Decision Process**
- **C) Quantum computing**
- **D) Convolutional models**
- **Answer: B) Markov Decision Process**

IV. MIND MAP


ERASER
TRIAL AND ERROR
• Teaching machines things that are hard to be explained by humans.
• Learning through rewards and punishments
REINFORCEMENT LEARNING
COMPONENTS
• Agent
• Environment
• Rewards/Punishments
<taking vacuum cleaner as an example>

LESSON 3: MARKOV DECISION PROCESS (COMING SOON)

LESSON 4: PROBLEMS WE'RE SOLVING FOR RL-MDP (COMING SOON)

CHAPTER C:
HARDWARE

CHAPTER C: HARDWARE

Pro tip

Hardware is a subject that requires a lot of curiosity. While reading this chapter, a good way to broaden your hardware knowledge even further is to watch a guy build a pc on social media, or, if you have the money, even build a pc yourself.

Otherwise, you can also try to take every electrical thing you see apart and analyze what it's made of, for example old laptops, phones, etc. Though, while you do it, make sure not to electrocute yourself or break the electrical appliance

MINI LESSON: WHAT IS HARDWARE?

To keep it simple:

In a computer, if you can touch it, it's hardware. If you can't, it's software.

So right now, just put your hands forward, and touch around your laptop or something. Was that the keyboard you just touched? Hardware. Did you touch that Mouse? Hardware. Is that a GeForce RTX 30 Series Graphics Card? Yeah, that is hardware. You get the idea.

Basically, a computer hardware is used to run instructions provided by the software. So if you want to run your Genshin Impact game, it's the game's software telling your hardware to display the game's windows on the screen (and overheat itselves).

But if you stopped to wonder, how in the world did this block of metal manage to be able to do so much stuff while being so small? What are those weird thingy written on my laptop that's called "specs" or something?

Item	Value
OS Name	Microsoft Windows 10 Pro
Version	10.0.19045 Build 19045
Other OS Description	Not Available
OS Manufacturer	Microsoft Corporation
System Name	DESKTOP-TVBJ12H
System Manufacturer	VMware, Inc.
System Model	VMware20,1
System Type	x64-based PC
System SKU	
Processor	Intel(R) Core(TM) i3-3110M CPU @ 2.40GHz,
Processor	Intel(R) Core(TM) i3-3110M CPU @ 2.40GHz,
BIOS Version/Date	VMware, Inc. VMW201.00V.20904234.B64.22
SMBIOS Version	2.7
Embedded Controll...	255.255

Those are what we're going to find out at the end of this chapter. So sit back, relax, and have a nice read.

(Yes I know, that's quite an old laptop.)

LESSON 1: NUMBERS AND STORAGE

Pro tip:

This lesson contains a lot of fundamental stuff, which are aimed at absolute beginners. So if you've already known something that's mentioned in this part, just skim it.

I .NUMBER SYSTEMS

1.Seriously, numbers?

Numbers are a tool used by humans in order to count, to label, or to measure something. Now, if you have learned anything about computers before, you'll know exactly where this part is going – with all that binary and hexadecimal number stuff (and if you do, you can skip this whole I, Number systems part)

But if you haven't, you may be wondering to yourselves: "Numbers? I thought we're learning about computers? This isn't a math textbook right?". Well, actually, understanding number systems is in fact crucial to understanding how a computer stores data and memory. In fact, what you're looking at right now, yes, this line of text, and that line of text, yes, even that image, it's all numbers to your laptop.

And how does that work? We're going to find out very shortly.

2.THE NUMBERS THAT WE USE - DECIMAL SYSTEM:

But first, let's ponder for a bit how the number system we use on a daily basis - which is called the Decimal system - works.

The decimal number system consists of ten different symbols as followBut first, let's ponder for a bit how the number system we use on a daily basis - which is called the Decimal system - works.

The decimal number system consists of ten different symbols as follows:

0 1 2 3 4

5 6 7 8 9

Now, whenever we see a number written in this system, our mind can subconsciously visualize the value that the number holds, take for example:

123 = a hundred and twenty three

However, if you take a look at the number again, it's just three numbers written close to each other. How does our brain look at these 3 numbers next to each other and then immediately visualize the number a hundred and twenty three?

Basically, when we run out of symbols in our number system (that is from 0 to 9), when we want to express a higher value, we instead add an extra number to the left of that digit. Notice how if we want to express a number bigger than 9, we instead add an extra 1 to the left of the current number.

$$9 + 1 = 10$$

$$99 + 1 = 100$$

Typically, what the number is expressing is: the further the digit is to the left, the higher value it's expressing.

$$123 = 1 \text{ x } 100 + 2 \text{ x } 10 + 3 \text{ x } 1$$

And since there are 10 numbers in the decimal system, every digit on the left has 10 times the value that the digit on its right.

$$123 = 1.10^{2} + 2.10^{1} + 3.10^{0}$$

Even though we've been used to this number system for a long time, it's quite difficult for a machine to read this kind of numbers. This is why people have to invent a different number system in order for computers to read, which is called the binary system.

3.BINARY SYSTEM:

Unlike the decimal system, binary numbers only has 2 numbers:

$$0 \text{ and } 1$$

And apart from that, everything works the same as the decimal system. When it runs out of digits, it will hop to the next one on the left.

$$1 + 1 = 10$$

$$11 + 1 = 100$$

And the further the digit is to the left, the higher its value. But, this time, the value of the digit to the left is twice as much as the digit to the right. Well, if you turn it back into the decimal number system I guess.

$$10110 = 1.16 + 0.8 + 1.4 + 1.2 + 0.1$$

$$10110 = 1.2^4 + 0.2^3 + 1.2^2 + 1.2^1 + 0.2^0 = 22$$

The way you turn decimal into binary, however, is by dividing the decimal number by 2, and recording the remainder until the decimal number hits 0. So for 22, it will look something like this:

$$22\,/\,2 = 11\,(rem\ 0)$$

$$11\,/\,2 = 5\,(rem\ 1)$$

$$5\,/\,2 = 2\,(rem\ 1)$$

$$2\,/\,2 = 0\,(rem\ 0)$$

$$1\,/\,2 = 0\,(rem\ 1)$$

And then, your final binary number will be all the remainders written from bottom to top. In this case, it's going to be 10110.

4.HEXADECIMAL SYSTEM:

The hexadecimal system, however, uses 16 symbols instead:

0 1 2 3 4 5 6 7 8 9

A B C D E F

And, after the number 9, the number system switches to the letters A B C D E F instead of more numbers. So if converted back into decimal system:

A = 10 B = 11 C = 12

D = 13 E = 14 F = 15

The hexadecimal system will be important later when we explore how data storage works. But fundamentally, you should look at a hexadecimal number as a compressed binary number. As hexadecimal numbers use a total of 2^4 numbers, every one of them can be represented as a series of 4 binary numbers.

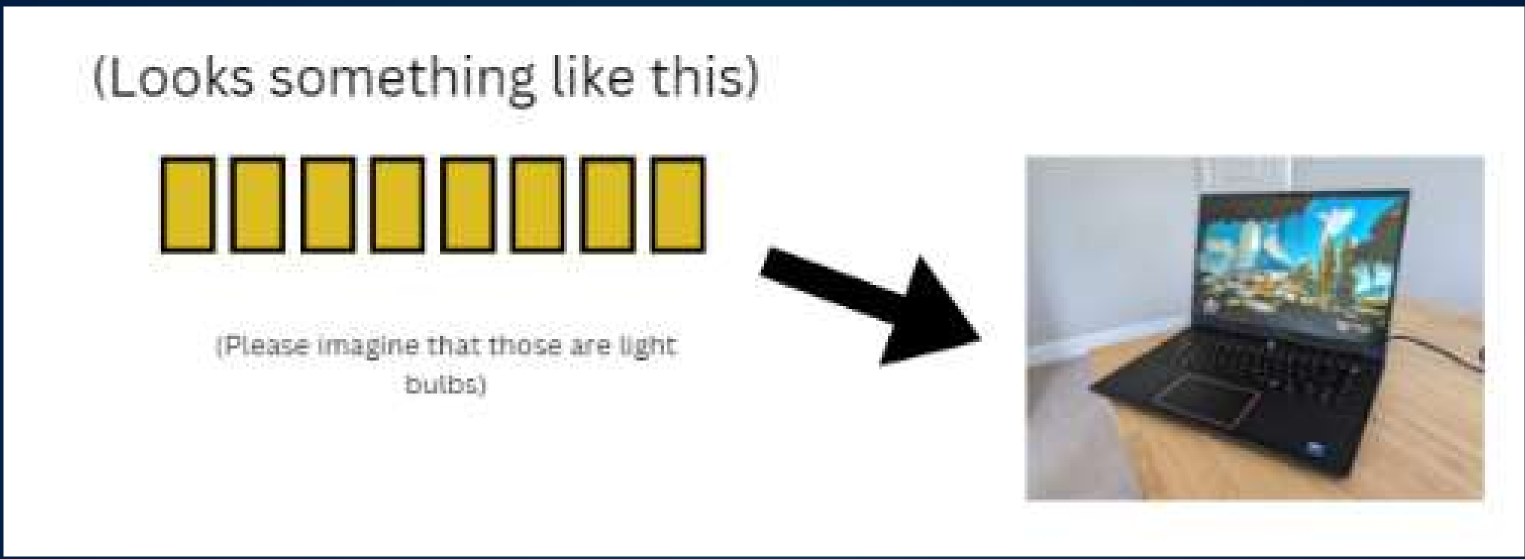
0 = 0000	1 = 0001	2 = 0010	3 = 0011
4 = 0100	5 = 0101	6 = 0110	7 = 0111
8 = 1000	9 = 1001	A = 1010	B = 1011
C = 1100	D = 1101	E = 1110	F = 1111

For example: B2 = 10110010

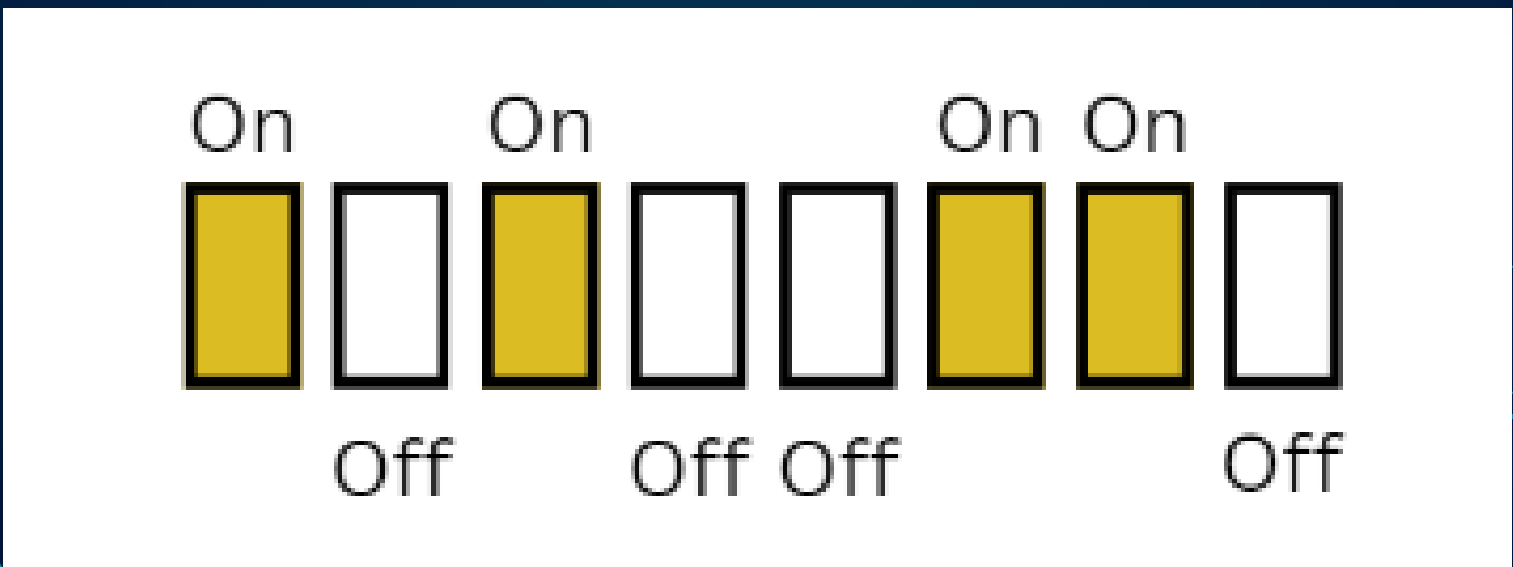
And yes, to a computer, it still works the same as binary numbers. The whole reason why the hexadecimal system was invented was that it's easier to read by humans and easily converted into the binary system.

5.BUT WHY? WHY DO WE NEED TO DO ALL THAT

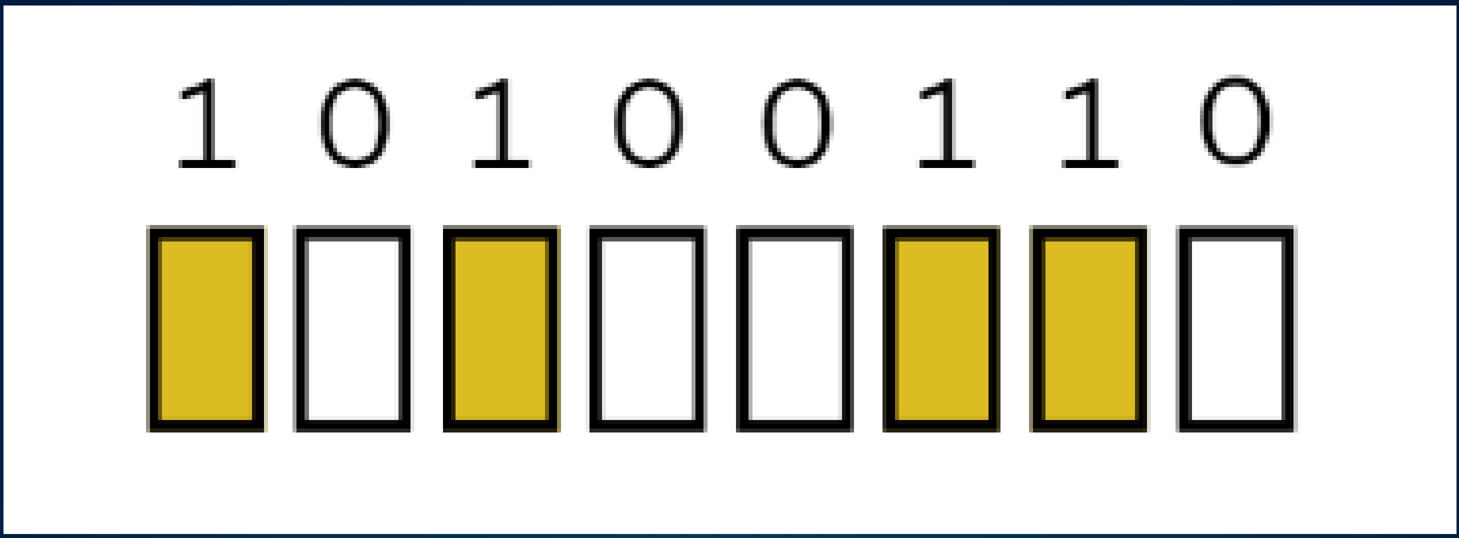
Now, the question is, why? Why do we need to know all these 10101 thingies? Why can't we just use the decimal system that we're all familiar with to store data.
All you need to understand is, the computer part that stores data works quite similar to a series of really, really small light bulbs.



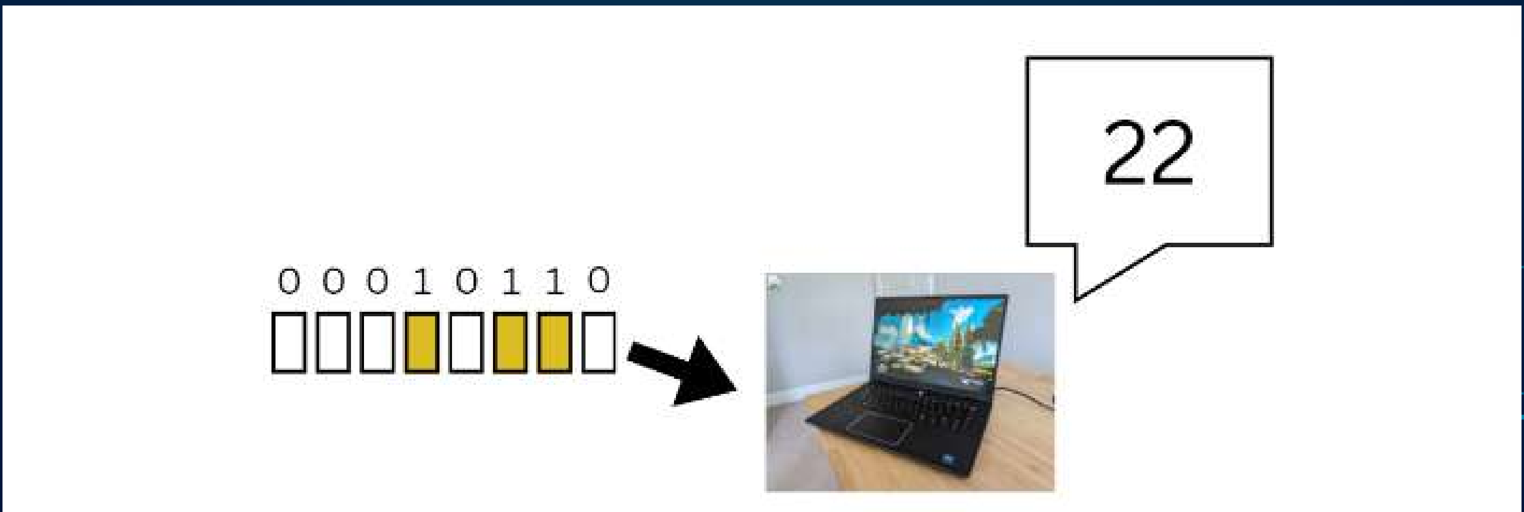
And each of those lightbulbs can either be turned on or off.



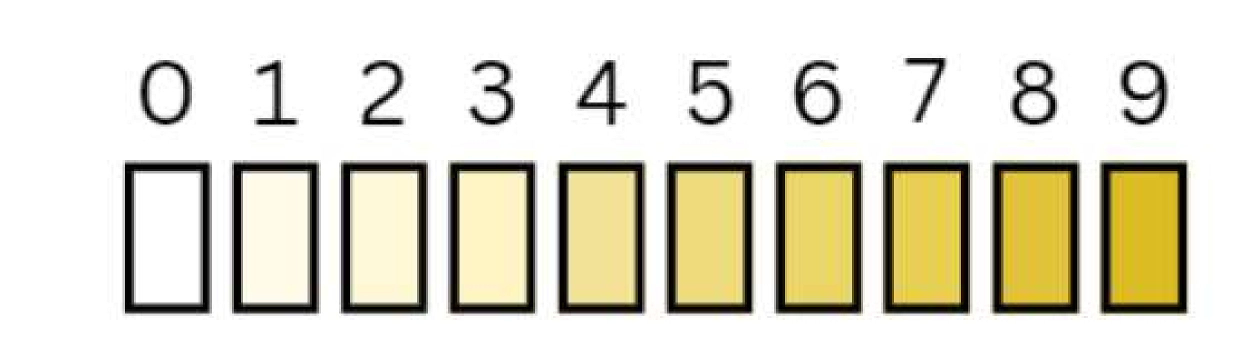
And yes, you've guessed it. Each of those on and off signals can be represented as a 0 or a 1 in the binary system, with 0 for OFF and 1 for ON



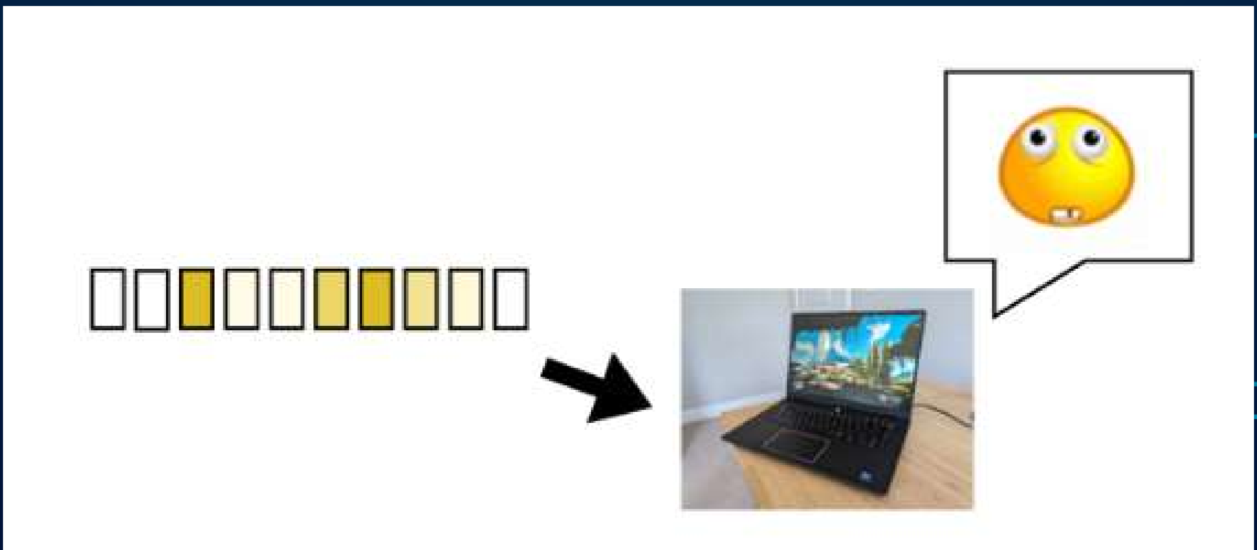
And with this sort of stuff, we can actually make our computer recognise our human numbers (that is the decimal numbers) quite easily.



Now, if we don't want the lightbulbs to just display 0 and 1, but we want the lightbulbs to display numbers from 0 to 9 instead, then, how? How are we supposed to get our poor laptop to understand those signals? Like are we supposed to make the lightbulbs turn "on", "off", "kinda off", "very off", "slightly off" to represent those numbers?



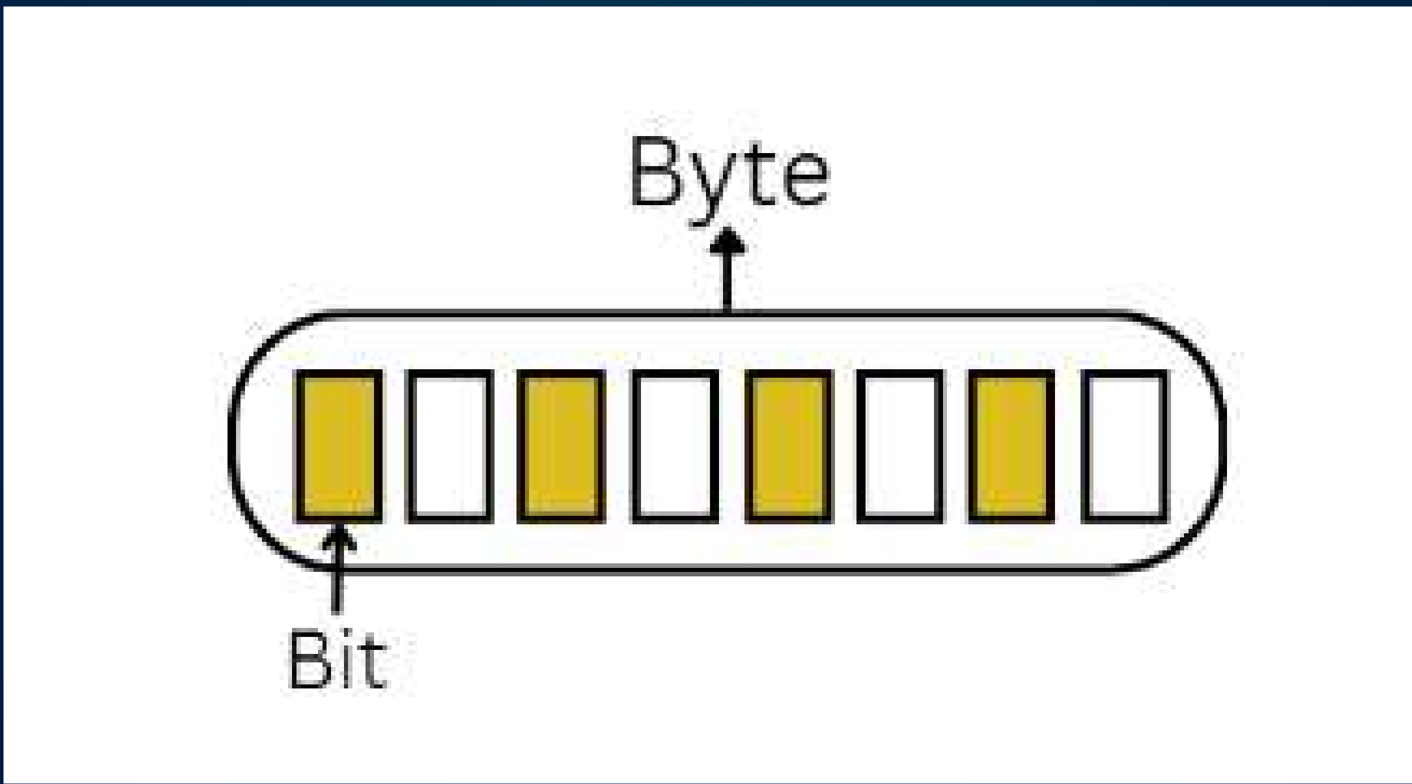
I mean, let's be real. Even in our eyes, those look pretty difficult to tell apart already. While technically it is possible to make it like that, well, it's just not worth it. Well, I'm not like an electrician or something, but I can imagine how much of a hell it would be to divert all the electricity sources so that the lightbulb can display the exact intensity you want it to be. So yeah, it's easier for machines to just detect whether lightbulbs are on or off - and that's why we should just use the binary system.



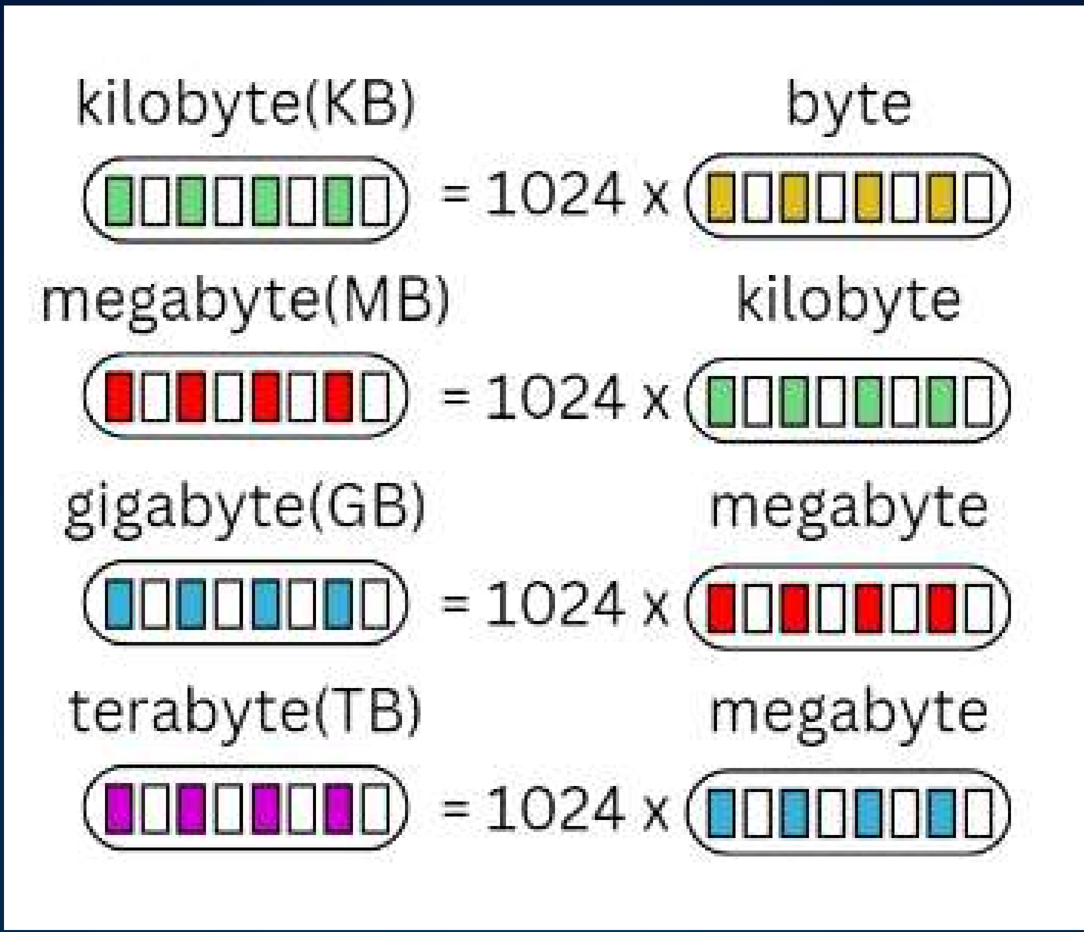
II. DATA STORAGE

1. BITS AND BYTES

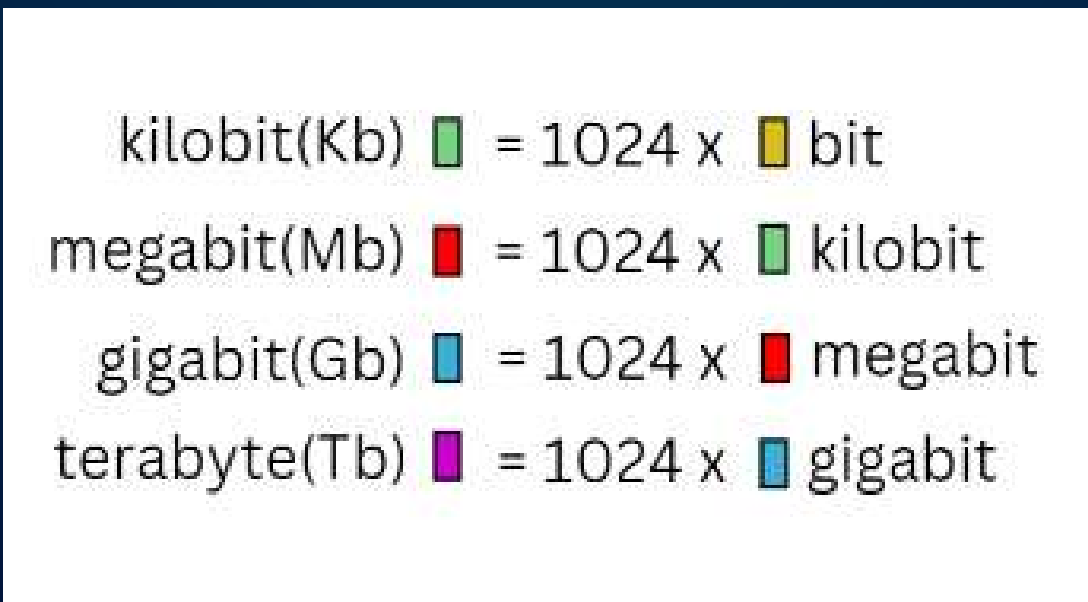
Remember those lightbulbs, on or off, 0 or 1 thing that we talked about last time? Well, in a computer storage system, each one of those lightbulbs is called a bit (which, fun fact, is binary digit shortened). A bit is represented as a lowercase b. A bit is the smallest measuring unit for storage in a computer. And since people decided that computers like to process those bits in bite-sized amounts, they invented a new measuring unit for 8 bits, called bytes. A byte is represented by an uppercase B.



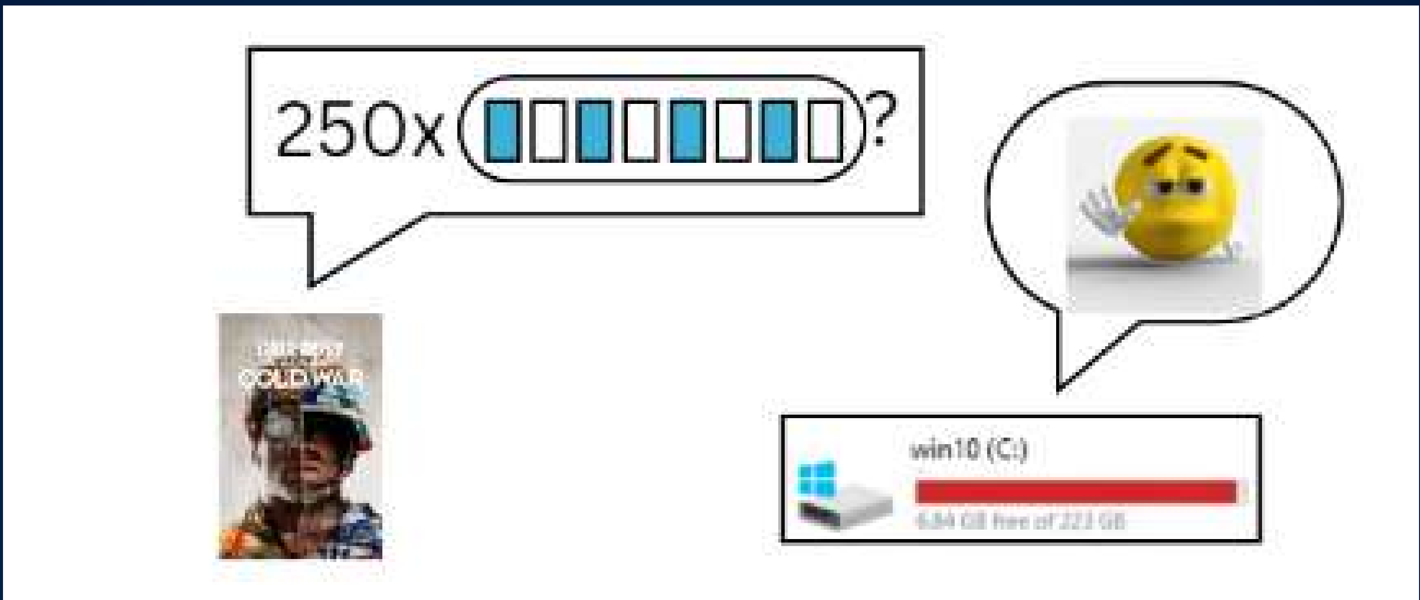
Using these bytes, people have created bigger measuring units in order to measure the amount of storage a certain data takes up.



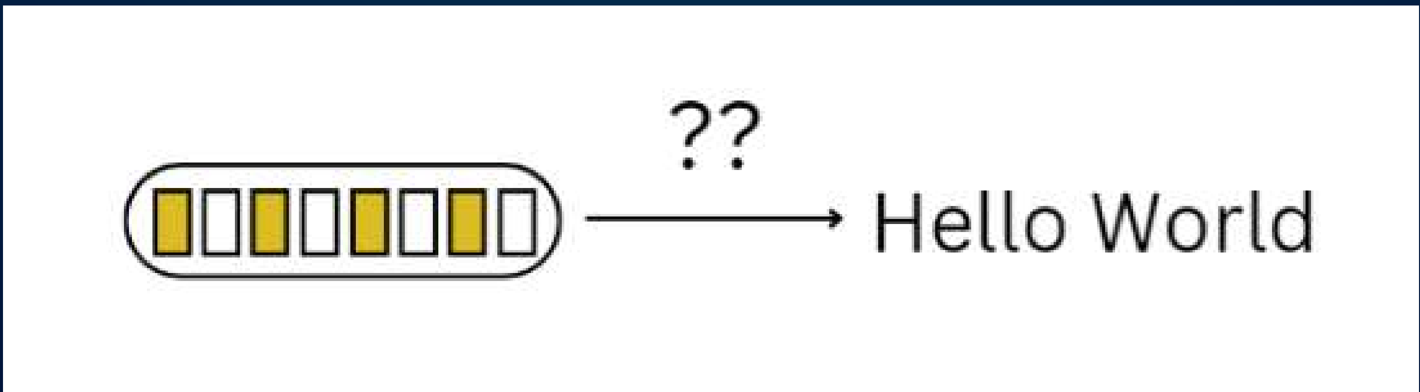
There is also a similar kind of thing for bits, which is usually used for measuring network traffic.



Basically, what these numbers are displaying is the number of bytes/bits required to store a piece of data. Or in other words, how many lightbulbs are needed. So for "Call of Duty Black Ops Cold War", you will need around 250GB, or 2.147483648e+12 light bulbs, just to store the game in your system.



And yes, every single kind of data in a computer can all be expressed with these series of bytes. Texts, pictures, sounds, videos, even this pdf file that you're viewing right now, to a computer, it's only a bunch of on and off signals. However, the way they are translated from those on and off signals are, of course, different from that of numbers.



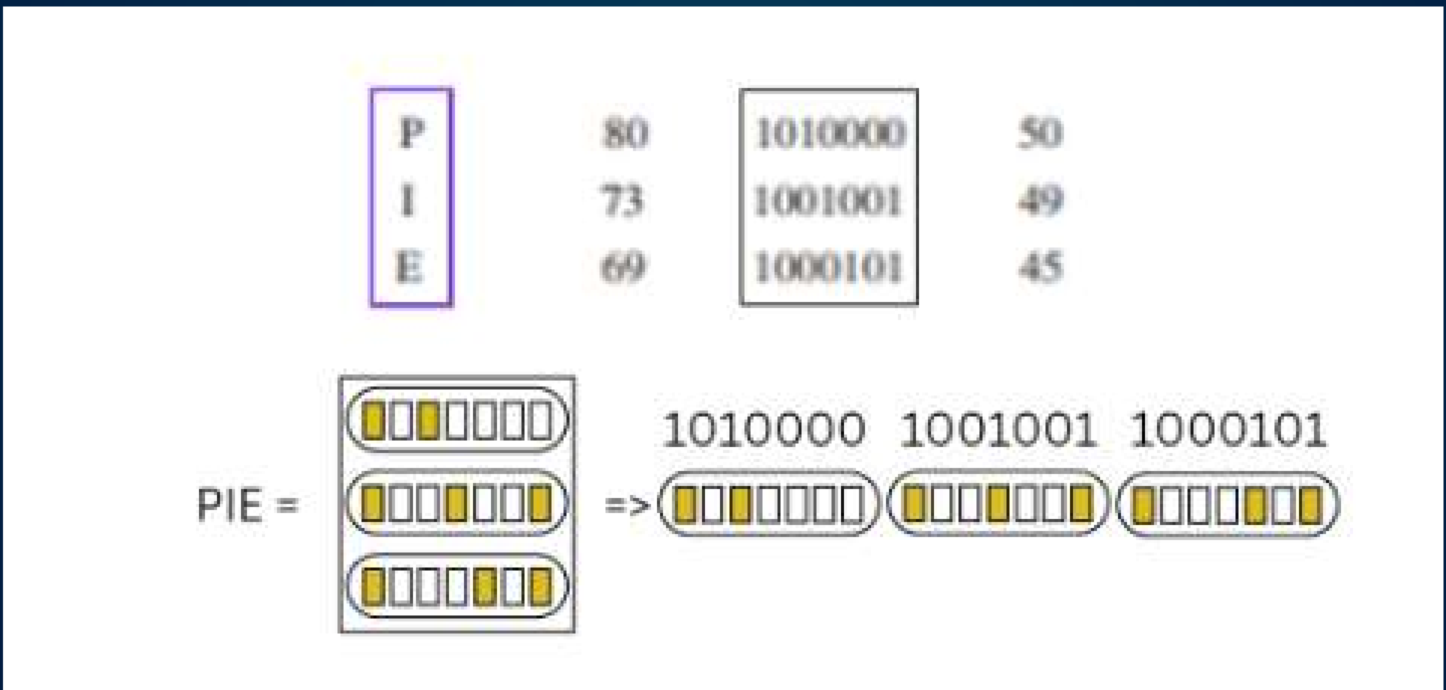
II. DATA STORAGE

2. TEXT

There have been different ways people have tried to encode letters with bytes. One of the earliest ways people have tried to do this is with ASCII (American Standard Code for Information Interchange, which uses 7 bits per character.

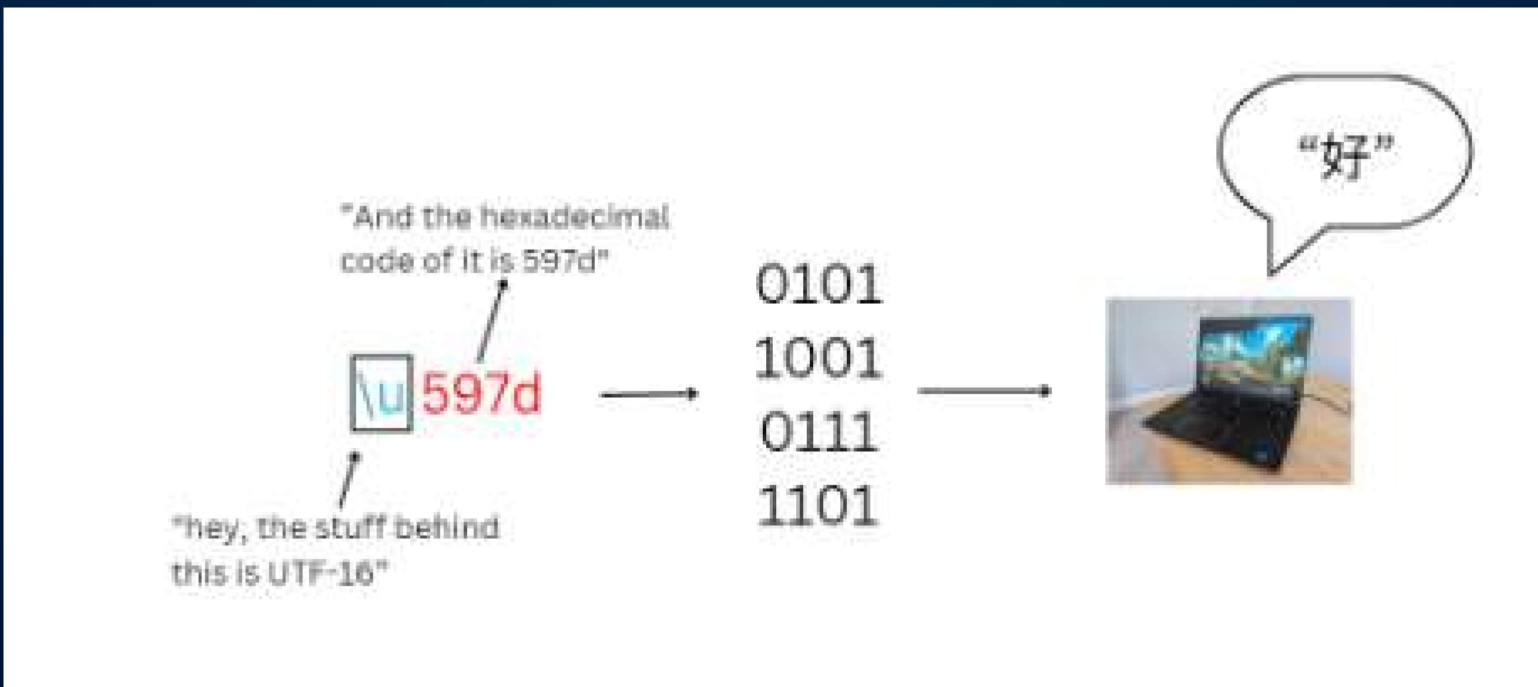
Control Characters				Graphic Symbols											
Name	Dec	Binary	Hex	Symbol	Dec	Binary	Hex	Symbol	Dec	Binary	Hex	Symbol	Dec	Binary	Hex
NUL	0	0000000	00	space	32	0100000	20	@	64	1000000	40	`	96	1100000	60
SOH	1	0000001	01	!	33	0100001	21	A	65	1000001	41	a	97	1100001	61
STX	2	0000010	02	"	34	0100010	22	B	66	1000010	42	b	98	1100010	62
ETX	3	0000011	03	#	35	0100011	23	C	67	1000011	43	c	99	1100011	63
EOT	4	0000100	04	$	36	0100100	24	D	68	1000100	44	d	100	1100100	64
ENQ	5	0000101	05	%	37	0100101	25	E	69	1000101	45	e	101	1100101	65
ACK	6	0000110	06	&	38	0100110	26	F	70	1000110	46	f	102	1100110	66
BEL	7	0000111	07	'	39	0100111	27	G	71	1000111	47	g	103	1100111	67
BS	8	0001000	08	(	40	0101000	28	H	72	1001000	48	h	104	1101000	68
HT	9	0001001	09	)	41	0101001	29	I	73	1001001	49	i	105	1101001	69
LF	10	0001010	0A	*	42	0101010	2A	J	74	1001010	4A	j	106	1101010	6A
VT	11	0001011	0B	+	43	0101011	2B	K	75	1001011	4B	k	107	1101011	6B
FF	12	0001100	0C	,	44	0101100	2C	L	76	1001100	4C	l	108	1101100	6C
CR	13	0001101	0D	–	45	0101101	2D	M	77	1001101	4D	m	109	1101101	6D
SO	14	0001110	0E	.	46	0101110	2E	N	78	1001110	4E	n	110	1101110	6E
SI	15	0001111	0F	/	47	0101111	2F	O	79	1001111	4F	o	111	1101111	6F
DLE	16	0010000	10	0	48	0110000	30	P	80	1010000	50	p	112	1110000	70
DC1	17	0010001	11	1	49	0110001	31	Q	81	1010001	51	q	113	1110001	71
DC2	18	0010010	12	2	50	0110010	32	R	82	1010010	52	r	114	1110010	72
DC3	19	0010011	13	3	51	0110011	33	S	83	1010011	53	s	115	1110011	73
DC4	20	0010100	14	4	52	0110100	34	T	84	1010100	54	t	116	1110100	74
NAK	21	0010101	15	5	53	0110101	35	U	85	1010101	55	u	117	1110101	75
SYN	22	0010110	16	6	54	0110110	36	V	86	1010110	56	v	118	1110110	76
ETB	23	0010111	17	7	55	0110111	37	W	87	1010111	57	w	119	1110111	77
CAN	24	0011000	18	8	56	0111000	38	X	88	1011000	58	x	120	1111000	78
EM	25	0011001	19	9	57	0111001	39	Y	89	1011001	59	y	121	1111001	79
SUB	26	0011010	1A	:	58	0111010	3A	Z	90	1011010	5A	z	122	1111010	7A
ESC	27	0011011	1B	;	59	0111011	3B	[	91	1011011	5B	{	123	1111011	7B
FS	28	0011100	1C	<	60	0111100	3C	\	92	1011100	5C	|	124	1111100	7C
GS	29	0011101	1D	=	61	0111101	3D	]	93	1011101	5D	}	125	1111101	7D
RS	30	0011110	1E	>	62	0111110	3E	^	94	1011110	5E	~	126	1111110	7E
US	31	0011111	1F	?	63	0111111	3F	_	95	1011111	5F	Del	127	1111111	7F

With 7 bits available, there are a total of 127 characters (or 2^7 - 1) it can represent. It works pretty simple: you see binary numbers, you look it up on the table, and just write out the corresponding symbol. For example, this is how you would write "PIE" in binary:



However, it became apparent that 127 characters is NOT enough. I mean, there are plenty of other languages that we have to take into account as well , right? Otherwise, how are we supposed to be able to type out 안녕하세요, 你好, and こんにちは?

That's why currently, we instead mainly use the UTF-16 text encoding instead. And unlike the ASCII encoding, UTF-16 instead uses 4 bytes (or 16 bits), which means it can now support 2^16, or 65536 values. This allows characters all across the world to be encoded into machine language, including Chinese, Cyryllic, Arabic, Japanese, Korean and even emojis.
And remember the Hexadecimal system we were talking about earlier? Since every character is encoded with 16 bits, instead of writing 16 sets of 1 or 0, people usually use 4 hexadecimal numbers (each representing 4 bits) instead. For example: 好 when encoded in UTF-16 becomes "\u597d", with "\u" being used to specify that it's an UTF-16 character, and 597d being the actual hexadecimal code.



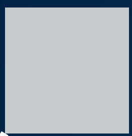
II. DATA STORAGE

3. IMAGES

There are plenty of different ways that images can be stored in a computer. But fundamentally, they all work like this.

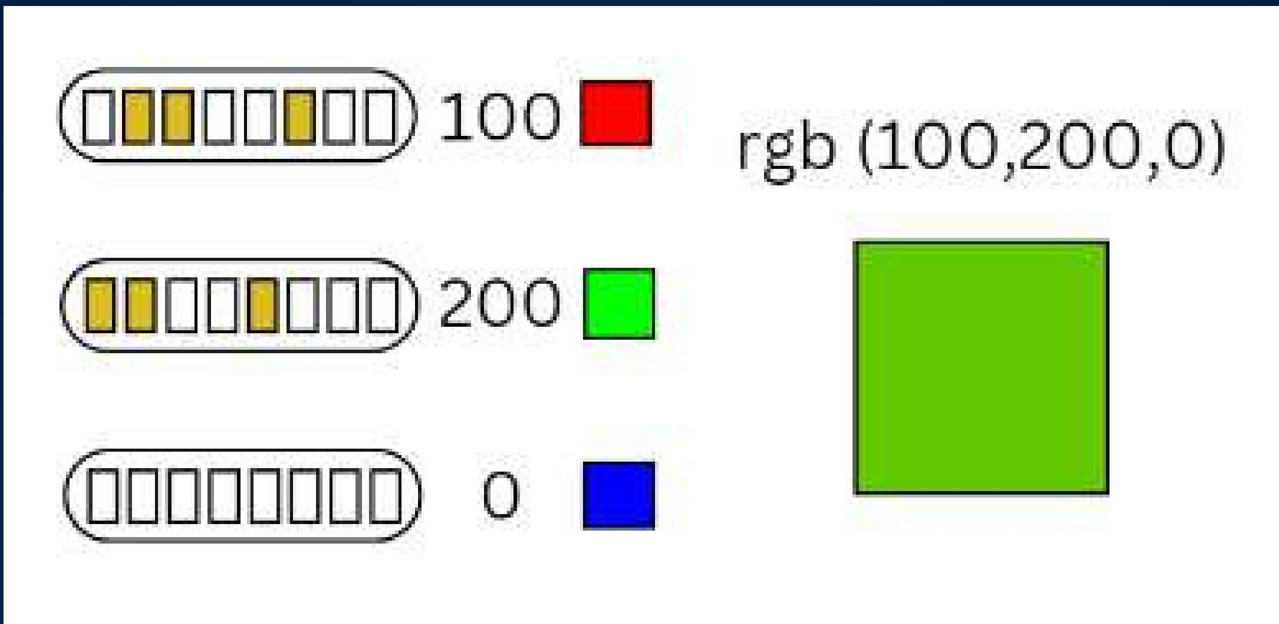
Look at this photo of this cat. And yes, even though it's an image of a cat to us, to a computer, it's still just a bunch of 0s and 1s.

Basically, images are stored as a combination of pixels, which is a little little little colored square that makes up every kind of image you see on the computer. For instance, this cat picture above, when I hover over its image on google, there's a little text, saying "3424 x 2570". This means that the image is 3424 pixels in width and 2570 pixels in height, or 8799680 pixels in total.

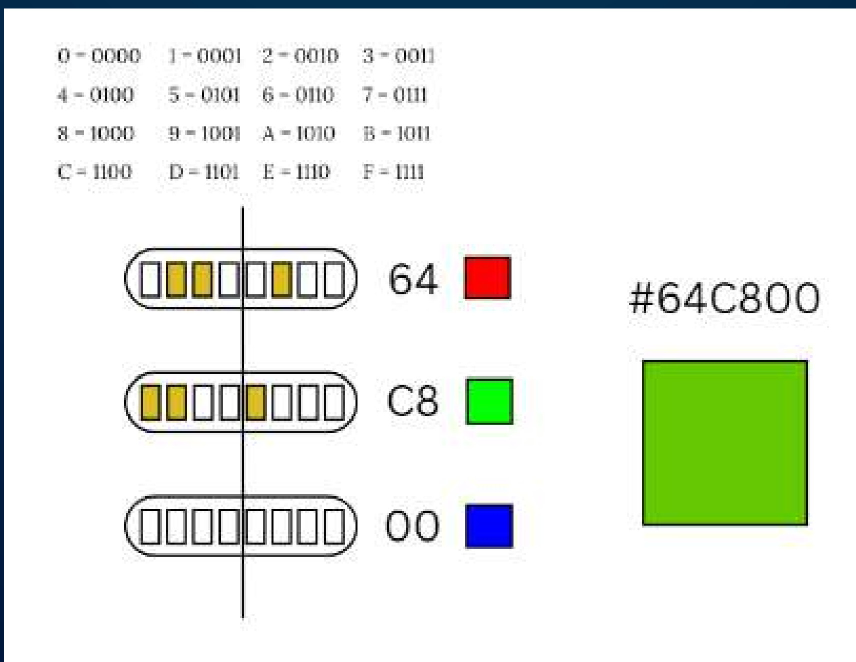


This image is made of 8799680 little squares like this

Each of those 8799680 pixels has a color assigned to it - which is represented by 3 bytes: a byte, or 8 bits for every Red, Green and Blue value. The higher the value, the more apparent the color looks. This is called the RGB color model, usually with the syntax rgb(red,green,blue), with the value amount represented in decimal numbers.



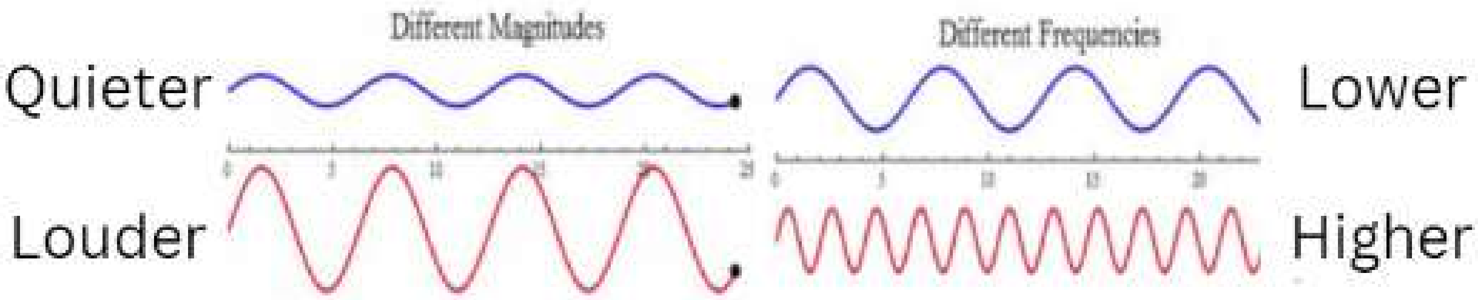
Also, since every value of red, green and blue is 8 bits, it can also be represented with 2 hexadecimal digits - This is called the HEX code, with the syntax #RRGGBB.



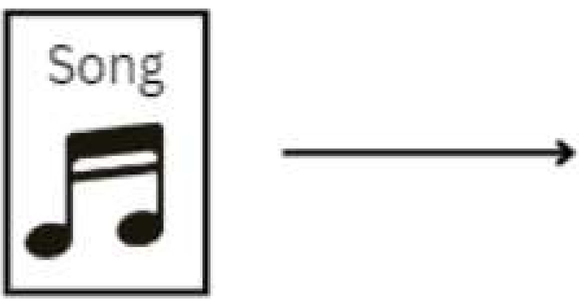
II. DATA STORAGE

4. AUDIO

In real life (and physics class), audio is usually represented as a wave. This wave right here has two main properties: frequency and magnitude. Frequency refers to how fast the wave is moving back and forth - and the higher the frequency, the higher the sound's pitch. Magnitude refers to how far away the wave moves away from the center point - and the higher the magnitude, the louder the sound is.



Now let's say, we download a song onto our computer. Now the question is, how can this song be represented by a series of binary numbers?



Well, quite similar to images actually. The computer simply divides the song into an enormous number of really really really small parts and samples them. The binary numbers will then be used to represent the frequency and magnitude of each one of those small parts, which, when put all together, the computer will be able to piece together a full audio file.

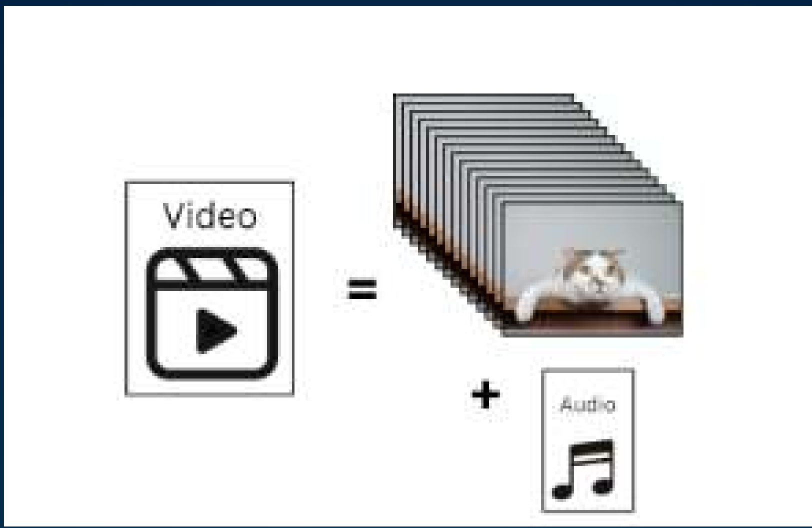
Usually, how much sampling the computer does really depends on the quality of the audio. People usually use the unit hertz (Hz) to measure this sampling rate, basically meaning how many times the machine samples the audio every second. So for example, for excellent quality audio, the audio gets sampled 48000 times every second :0 (hence why i used a dust image up there)

Format	Sampling	Bit Rate	Quality	Size
MP3	8000 - 16000 Hz	16 - 96 kbps	Very low	Very small
	16000 - 32000 Hz	96 - 196 kbps	Decent	Small
	44100 Hz	256 - 320 kbps	Good	Medium
	48000 Hz	320 kbps	Excellent	Large

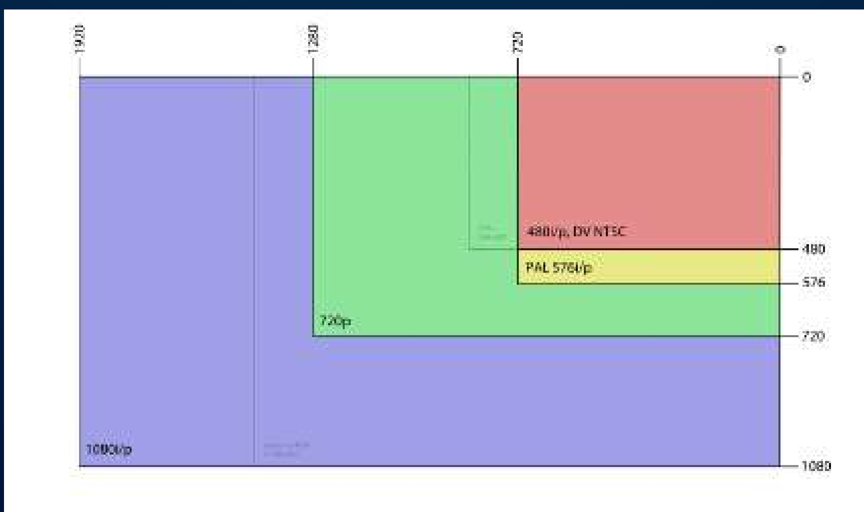
II. DATA STORAGE

5. VIDEOS

Videos are quite straightforward. It's just an audio file playing over a series of images that get shown to the screen one by one.



The images that form a video like that are usually referred to as frames, and a video may have a different number of frames every second, or fps. Usually, a standard video uses 60fps; an animation video uses 24fps, and video games and some movies may resort to a higher fps, like 120fps, 144fps or 240fps.
You may be also wondering - when you watch a Youtube video, what does the 1080p, 144p, 4k thingy mean? Well simply, they're the amount of pixels in each frame of the video. You can refer to this image below.



III. EXERCISES

EX1. Convert the hexadecimal/decimal number into binary

1. Convert decimal 45 to binary.
Answer: 101101

2.Convert hexadecimal 1A to binary.
Answer: 00011010

3.Convert binary 11101 to decimal.
Answer: 29

4.Convert decimal 255 to hexadecimal.
Answer: FF

5.Convert hexadecimal B2 to binary.
Answer: 10110010

III. EXERCISES

Ex2. Fill in the blanks

1.The decimal number system consists of ______ different symbols, ranging from 0 to ______.
Answer: 10, 9

2.A binary number system only has ______ symbols, which are ______ and ______.
Answer: 2, 0, 1

3.In computer storage, a single light bulb-like unit that represents an on/off signal is called a ______.
Answer: bit

4.ASCII encoding uses ______ bits per character, while UTF-16 uses ______ bits per character.
Answer: 7, 16

5.The color model that represents colors using Red, Green, and Blue values in a computer is called the ______ model.
Answer: RGB

III. EXERCISES

Ex3. True/False Questions

1.The hexadecimal system uses 12 symbols, including letters and numbers.
Answer: False (It uses 16 symbols, from 0-9 and A-F)

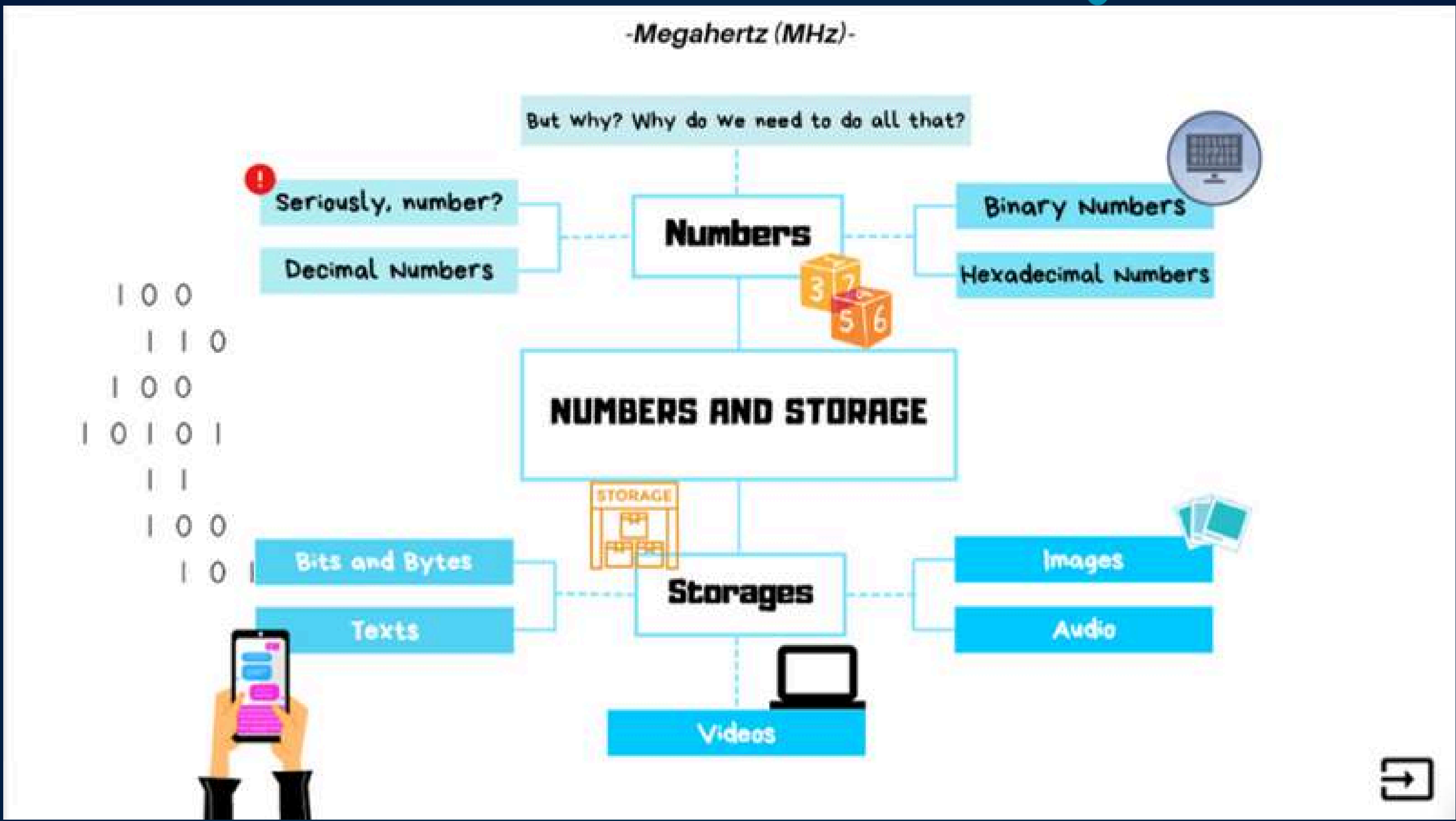
2.A byte consists of 8 bits.
Answer: True

3.In a binary system, each digit to the left has half the value of the digit to its right.
Answer: False (Each leftward digit doubles in value)

4.UTF-16 encoding allows computers to represent characters from multiple languages.
Answer: True

5.Videos are essentially a series of images (frames) displayed in quick succession, often accompanied by audio.
Answer: True

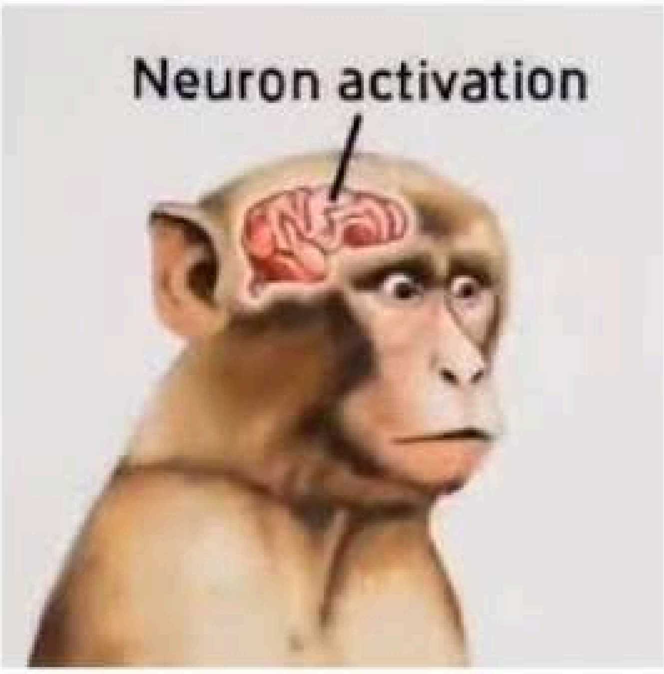
IV. MIND MAP



-Megahertz (MHz)-
But why? Why do we need to do all that?
Seriously, number?
Decimal Numbers
Numbers
Binary Numbers
Hexadecimal Numbers
NUMBERS AND STORAGE
STORAGE
Bits and Bytes
Texts
Storages
Images
Audio
Videos

LESSON 2:
THE COMPUTER ARCHITECTURE

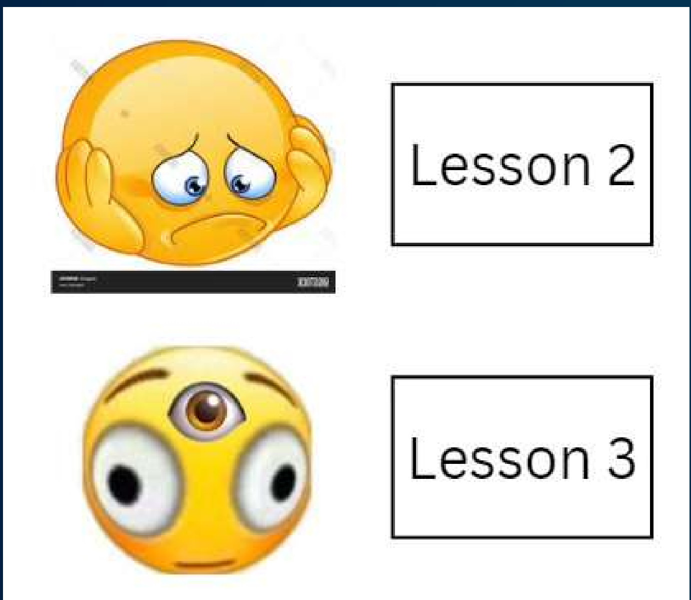
In this lesson, we're going to look at what's actually inside a computer when you take it apart, that includes the CPU GPU and stuff like that. What you learn here may be useful next time you're looking for a good gaming PC setup. So yeah, you'd better pay attention :D

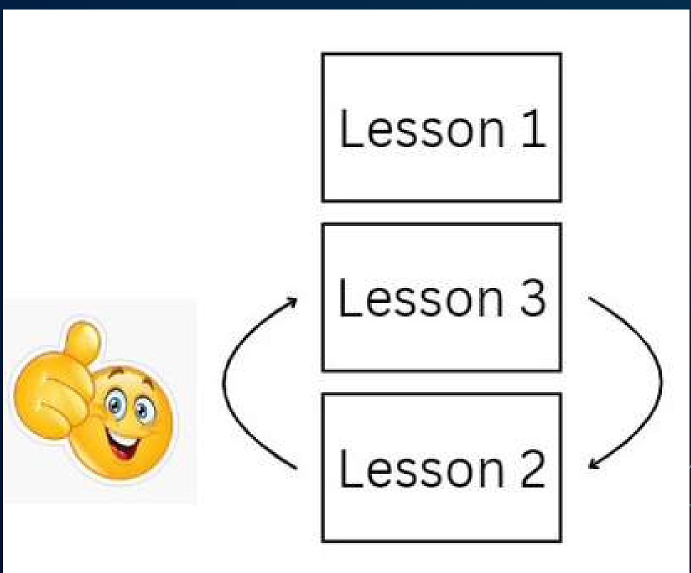


LESSON 2:
THE COMPUTER ARCHITECTURE

Pro tip

This lesson is highly theory-based, which may appear boring to a lot of people. In order to understand everything better, pair what you learned in this lesson with Lesson 3: Program Execution.

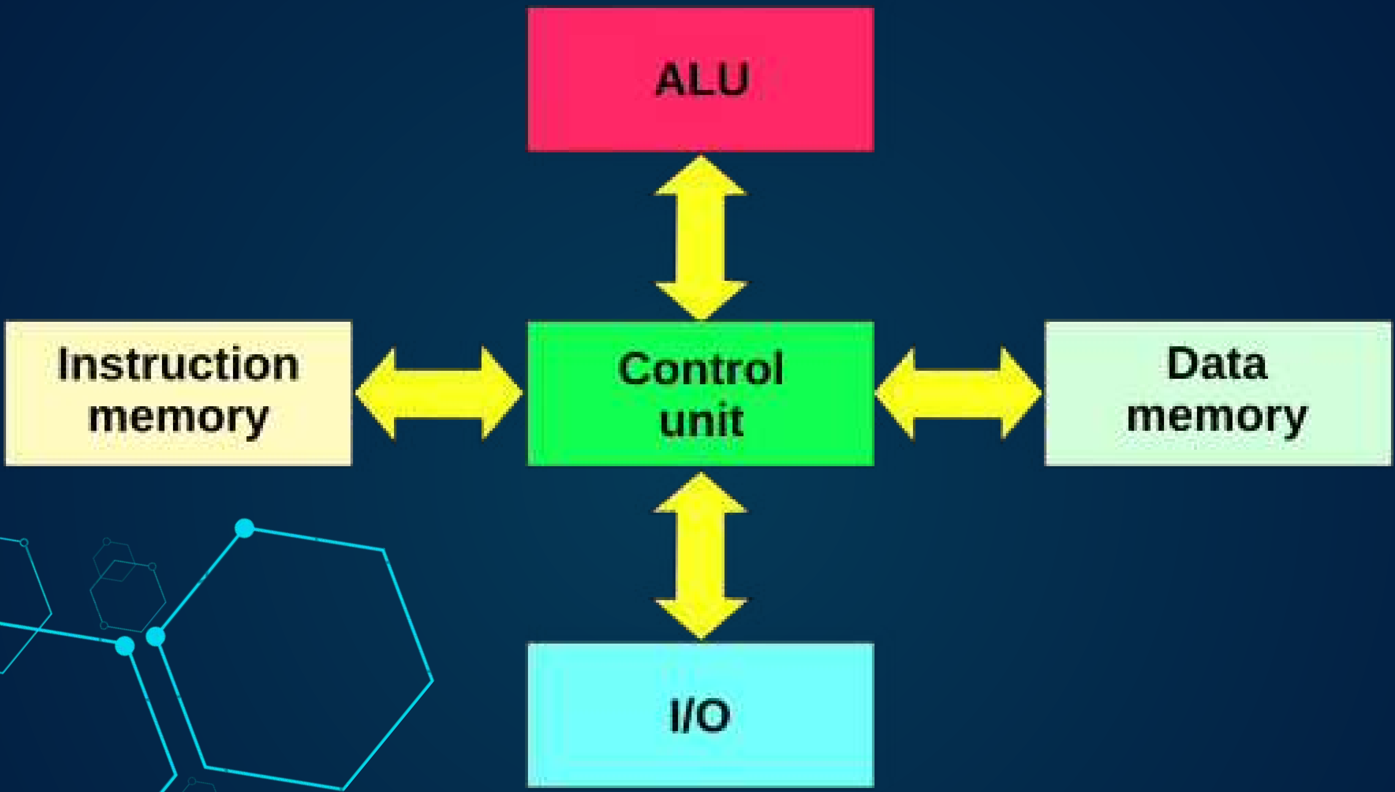




For some people, it may even be more efficient if you start from Lesson 3 first, then go back to Lesson 2. But since this lesson provides the fundamentals and basics of the computer model, we're still going to put it before Lesson 3.

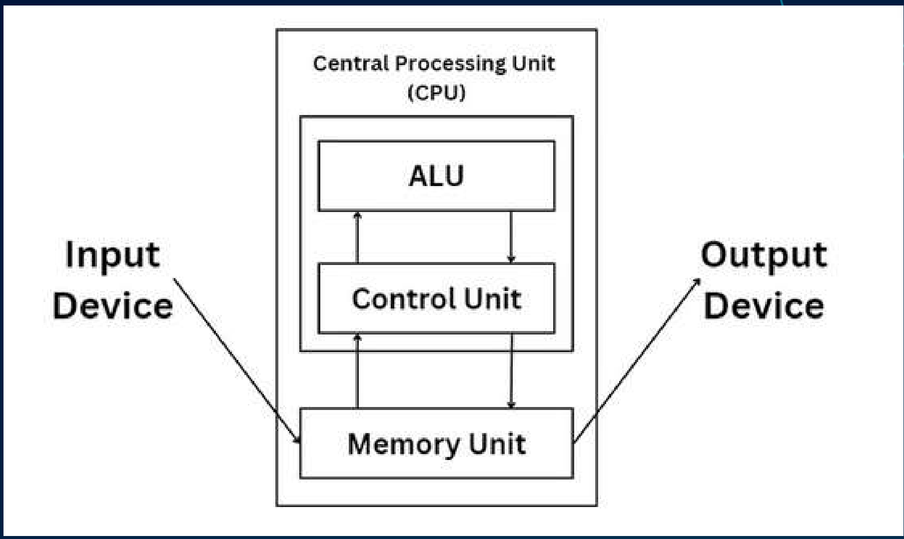
I. THE VON NEUMANN MODEL

In the past, there were numerous ways people have attempted to create machines that can think. One of the approaches to this problem was the Harvard Model, which was, sadly, not used by a lot of computers these days.



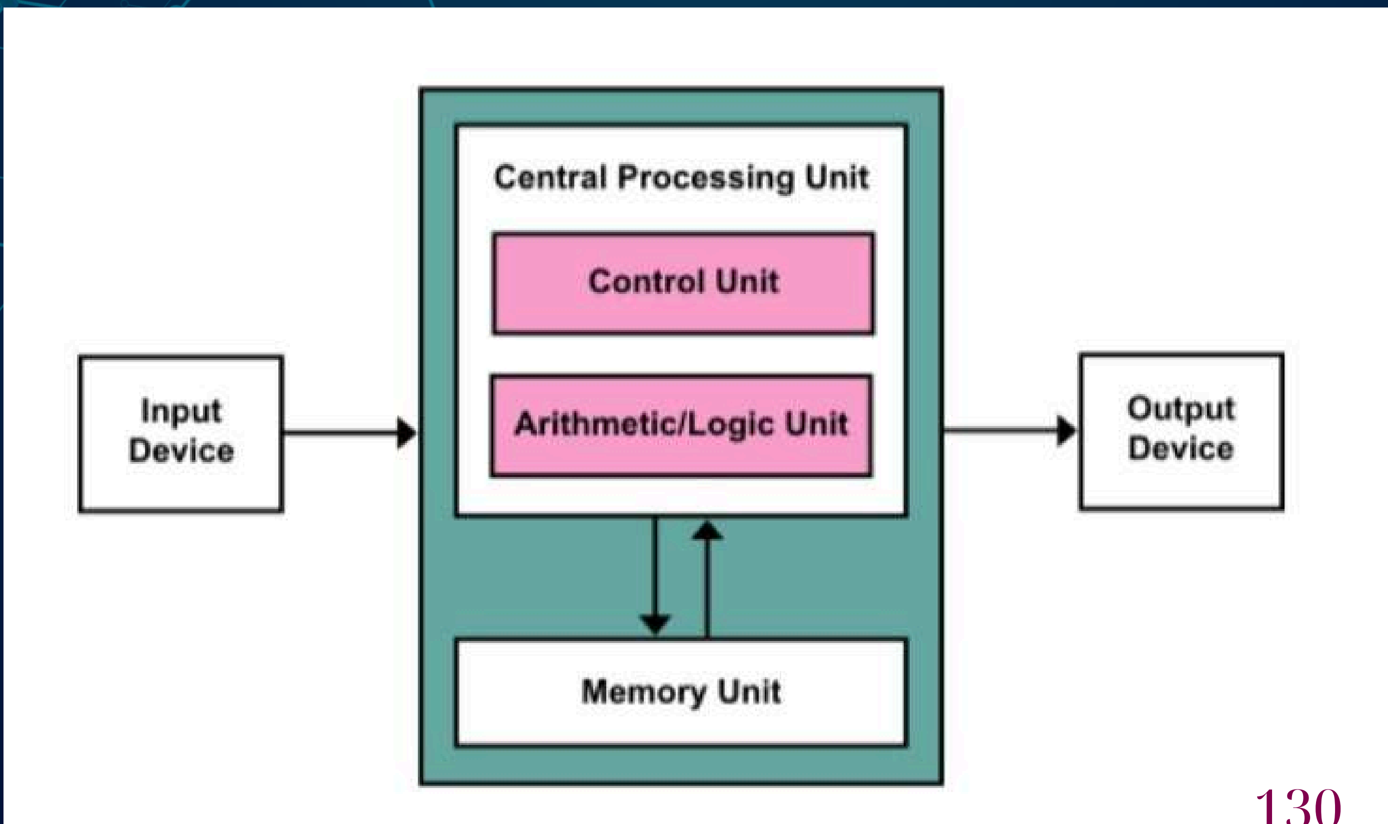
However, the discovery that changed computers forever was the Von Neumann model, also known as Von Neumann architecture, which is actually the computer model that's used by most of the computers these days. It was first described in 1945 when John von Neumann and his colleagues worked on the first draft of a Report on the EDVAC.

I. THE VON NEUMANN MODEL



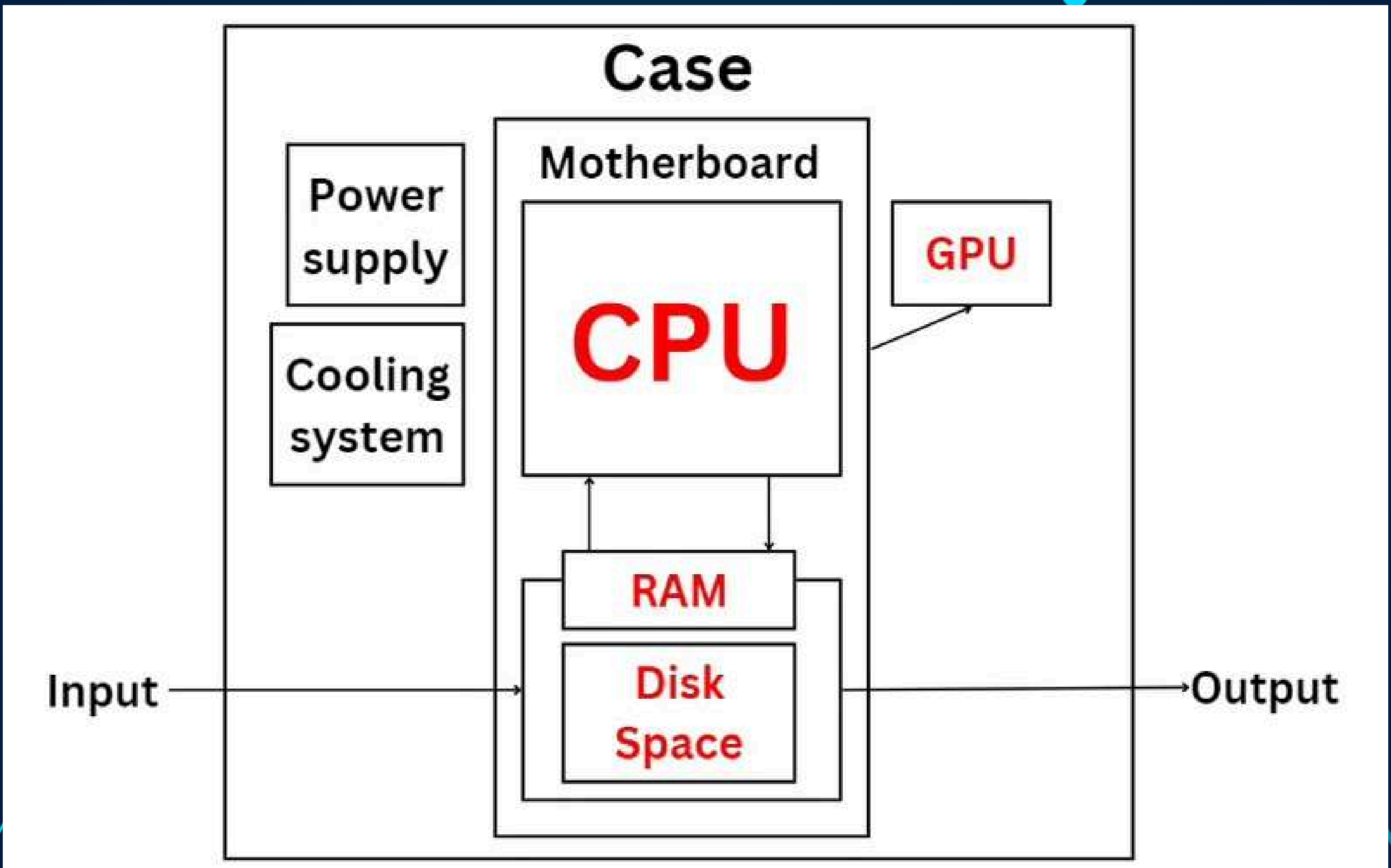
The von Neumann model has 3 key components: an Input/Output Device, a Memory Unit, and a Central Processing Unit (CPU) (which includes Arithmetic Logic Unit and Control Unit). You can notice how the approach is different from the Harvard model, as the instructions and data are all stored in the same Memory unit.

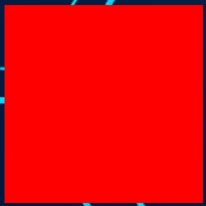
The crazy thing is that this model is actually pretty much the same as the current computers these days - with just a little bit more upgrade. We will explore the components of the modern computer right away!



II. COMPUTER HARDWARE COMPONENTS

To explain the components of a computer hardware better, I'm going to split them into 2 parts: the simple components, and the more complex components.





Complex Components

Simple Components

1.Simple Components

These components should be quite straightforward, because they are not involved in any data storage or processing.

Input devices

are stuff you can use to send instructions to your computer. This includes the mouse, keyboard, microphone, cameras and video game controllers. So for example, when you press a button on a keyboard, it's basically sending instructions for the computer to display the character we just sent on the screen.

Output devices

are stuff that the computer uses to send you back information. This can include your screen, speakers and headphones. In the previous case, after the instructions have been received, the computer will then display the letter "J" on the screen,

II. COMPUTER HARDWARE COMPONENTS

Case

contains everything inside the computer, except input - output devices.

Power supply

provides electricity to computer

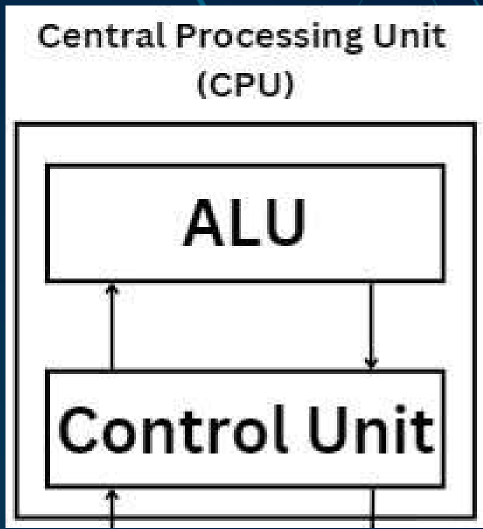
Motherboard

is an electric circuit that connects the CPU, memory units and a lot iof other stuff together. Usually referred to as the "heart" of the computer.

II. COMPUTER HARDWARE COMPONENTS

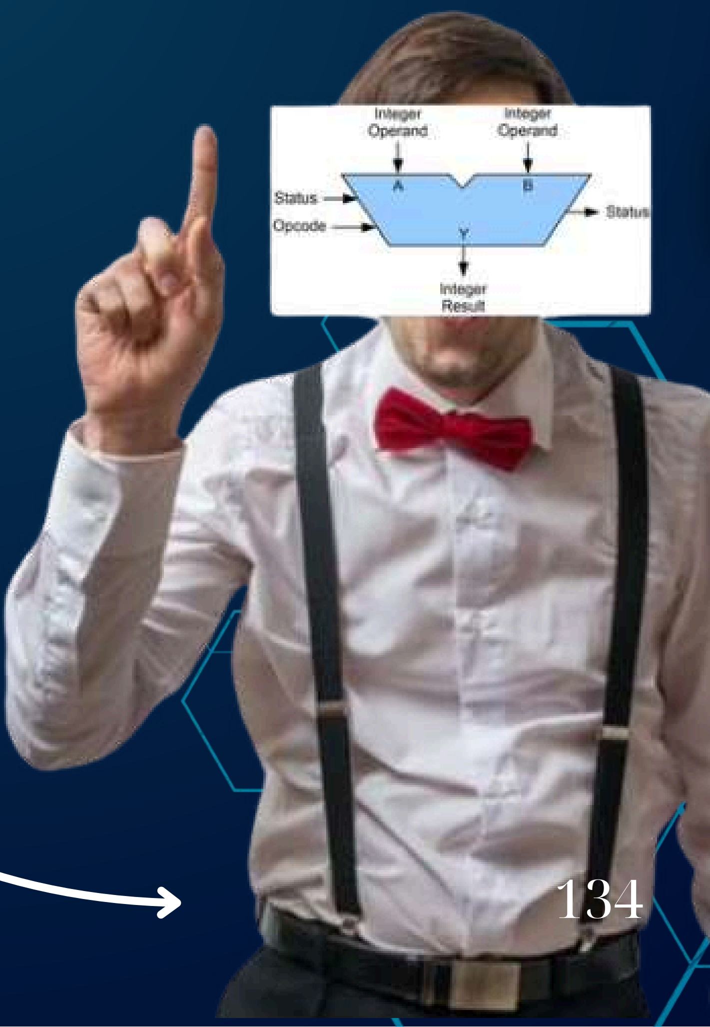
2.Central Processing Unit (CPU)

The CPU, also called the Central Processing Unit or Processor, is used to execute instructions in the computer's software.
The CPU is composed of two main parts: Arithmetic Processing Unit (ALU) and the Control Unit (CU).



An Arithmetic Processing Unit (ALU)

is used to do mathematical operations and logic processing. Or in other words, ALU is a nerd.

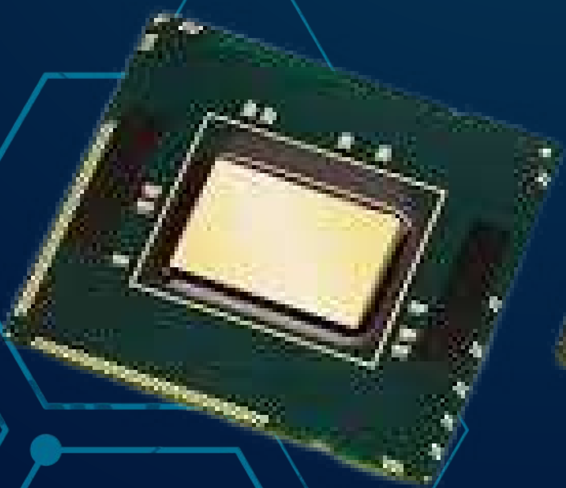


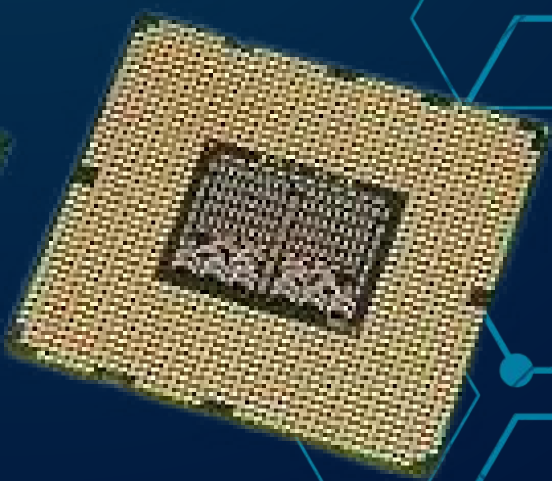
II. COMPUTER HARDWARE COMPONENTS

2.Central Processing Unit (CPU)

The Control Unit (CU)

controls all the main memory, arithmetic & logic unit (ALU), input and output devices, and is also the thing that sends the instructions to the ALU. Therefore, it’s usually considered the “brain” of the CPU. I personally like to think of the Control Unit as a kind of delivery man in the computer model.





II. COMPUTER HARDWARE COMPONENTS

3.Memory Unit

3.1 Storage units

Storage units are used to store your files, folders and operating systems, and will still be maintained even when the computer turns off. Storage units is actually split into 2 types:

Hard Disk Drive (HDD):

The HDD is the primary long-term storage device. It uses magnetic storage to store the operating system, software, and user files on a spinning disk

Anatomy of an HDD (left) v/s Anatomy of an SDD

Solid-State Drive (SSD):

SSDs perform the same function as HDDs but use flash memory instead of spinning disks to store data.

It's faster than HDDs because they have no moving parts, which leads to quicker boot times and faster access to data. Even though it's more expensive than HDDs, it is becoming increasingly popular due to their speed and durability.

II. COMPUTER HARDWARE COMPONENTS

3.2 Random Access Memory (RAM):

RAM is Temporary memory allocated from disk space in order to be used and processed by the CPU. More RAM allocated into the computer means the CPU can also process information faster, improving computer performance.

II. COMPUTER HARDWARE COMPONENTS

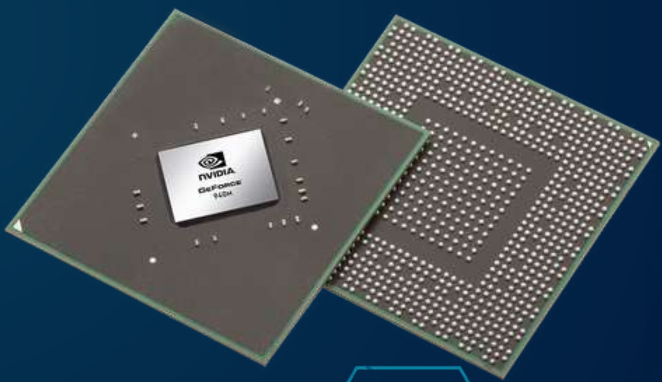
4.Graphics Processing Unit (GPU)

The GPU, or graphics card, is responsible for rendering images, video, and animations for the computer's display. It's especially important for tasks like gaming, video editing, and 3D rendering. It's also really go

GPUs are split into 2 types:

Integrated GPU

Built into the CPU and shares system memory. Suitable for general use and basic graphics tasks.



Dedicated GPU

A separate card with its own memory (VRAM) for higher performance, often used in gaming and professional applications.

III. EXERCISES

Ex1. Multiple choice (Do like 10 questions on this one)

1.Who first described the von Neumann model?
a) Charles Babbage
b) Alan Turing
c) John von Neumann
d) Steve Jobs
Answer: c) John von Neumann

2.Which component of the von Neumann model is responsible for arithmetic and logic operations?
a) Control Unit (CU)
b) Memory Unit
c) Arithmetic Logic Unit (ALU)
d) Input/Output Device
Answer: c) Arithmetic Logic Unit (ALU)

3.What kind of memory is used by the CPU to store frequently accessed data for quick access?
a) RAM
b) Cache Memory
c) SSD
d) BIOS
Answer: b) Cache Memory

4.In the von Neumann model, data and instructions are stored:
a) Separately in different memory units
b) In the CPU only
c) Together in the same memory unit
d) Only in the Control Unit
Answer: c) Together in the same memory unit

5.Which of the following is NOT a type of expansion card?
a) Sound Card
b) Network Interface Card (NIC)
c) Power Supply Unit (PSU)
d) USB Card
Answer: c) Power Supply Unit (PSU)

III. EXERCISES

6.Which component is often called the "brain" of the computer?
a) GPU
b) RAM
c) CPU
d) PSU
Answer: c) CPU
7.What does RAM store?
a) Operating system files only
b) Long-term files and user data
c) Data actively being used by the CPU
d) Permanent configuration settings
Answer: c) Data actively being used by the CPU
8.Which of these has its own memory known as VRAM?
a) CPU
b) Motherboard
c) Integrated GPU
d) Dedicated GPU
Answer: d) Dedicated GPU
9.What type of storage device uses spinning platters to store data?
a) SSD
b) RAM
c) HDD
d) BIOS
Answer: c) HDD

10.Which firmware initializes hardware and loads the operating system during boot-up?
a) SSD
b) BIOS/UEFI
c) RAM
d) CPU
Answer: b) BIOS/UEFI

III. EXERCISES:

Ex2.True/False Questions

1. The Control Unit (CU) is responsible for sending instructions to the ALU.
    - Answer: True
2. Random Access Memory (RAM) retains data even when the computer is powered off.
    - Answer: False
3. The von Neumann model stores data and instructions in separate memory units.
    - Answer: False
4. The Power Supply Unit (PSU) converts electricity from an outlet into a usable form for the computer.
    - Answer: True
5. Integrated GPUs are better suited for gaming and 3D rendering than dedicated GPUs.
    - Answer: False

Ex3.Fill in the blanks

1.The ________________ is responsible for performing arithmetic and logical operations in the CPU.

- Answer: Arithmetic Logic Unit (ALU)

2.________________ memory stores data that is actively being used by the CPU for faster access.

- Answer: Random Access Memory (RAM)

3.The ________________ is the main circuit board that connects all major hardware components.

- Answer: Motherboard

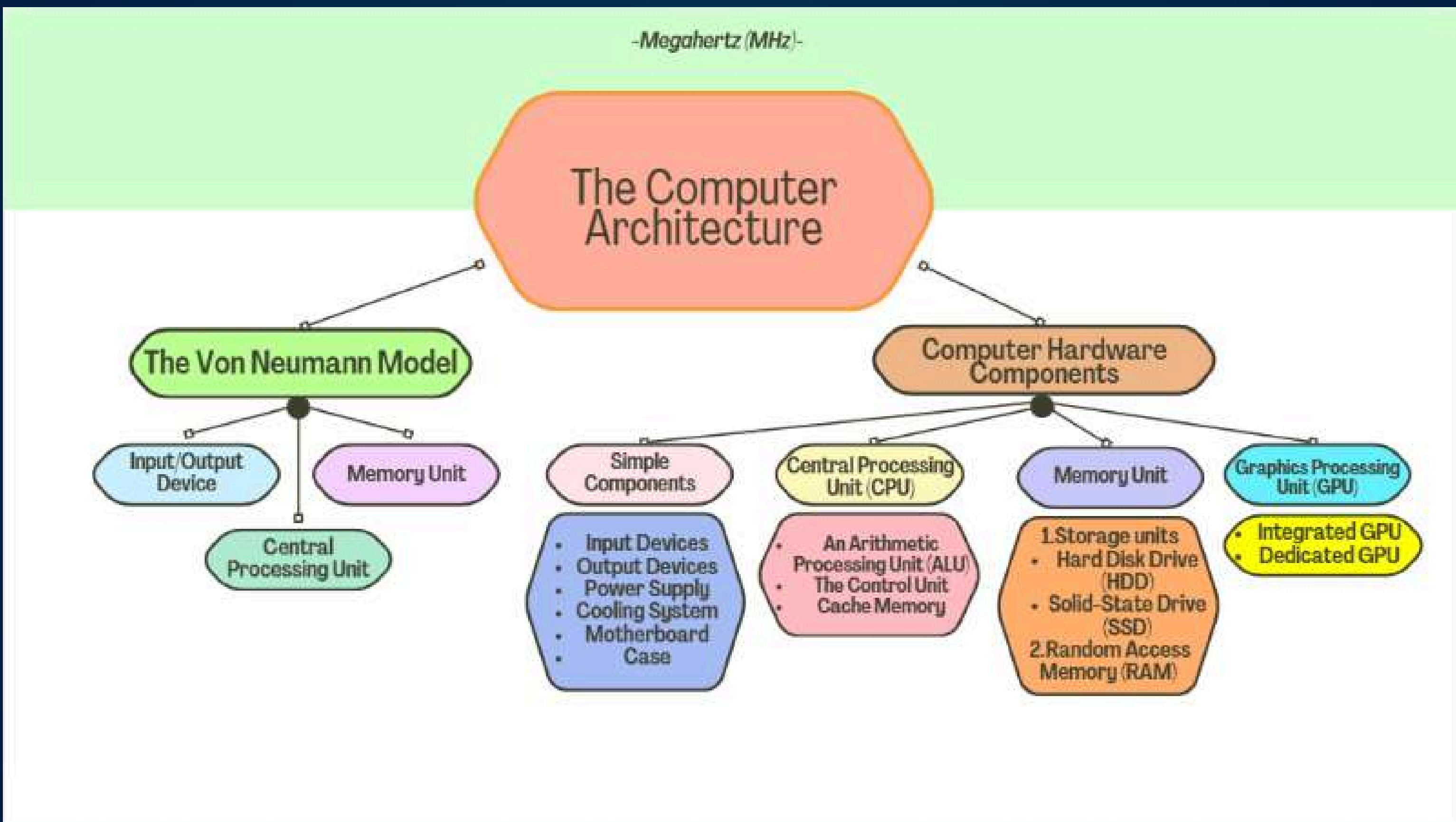
.4.Hard Disk Drives (HDDs) use ________________ storage to save data.

- Answer: magnetic

5.The ________________ or UEFI initializes hardware and loads the operating system during boot-up.

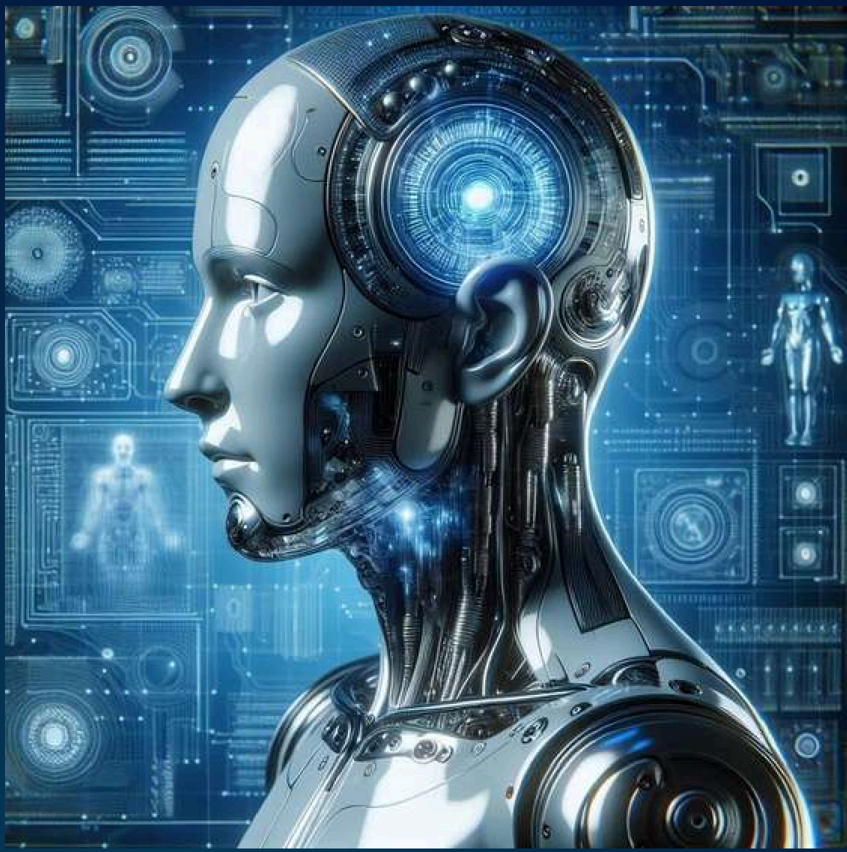
- Answer: BIOS

IV. MIND MAP


-Megahertz (MHz)-
The Computer Architecture
The Von Neumann Model
Input/Output Device
Memory Unit
Central Processing Unit
Computer Hardware Components
Simple Components
Input Devices
Output Devices
Power Supply
Cooling System
Motherboard
Case
Central Processing Unit (CPU)
An Arithmetic Processing Unit (ALU)
The Control Unit
Cache Memory
Memory Unit
1.Storage units
Hard Disk Drive (HDD)
Solid-State Drive (SSD)
2.Random Access Memory (RAM)
Graphics Processing Unit (GPU)
Integrated GPU
Dedicated GPU

LESSON 3: PROGRAM EXECUTION

In this lesson, we're going to follow up the last lesson by explaining how the mentioned computer components actually work together while executing a program. And to do that, we're going to explore how a simple computer processes an instruction provided by the user. Let's get right into it!



Pro tip:

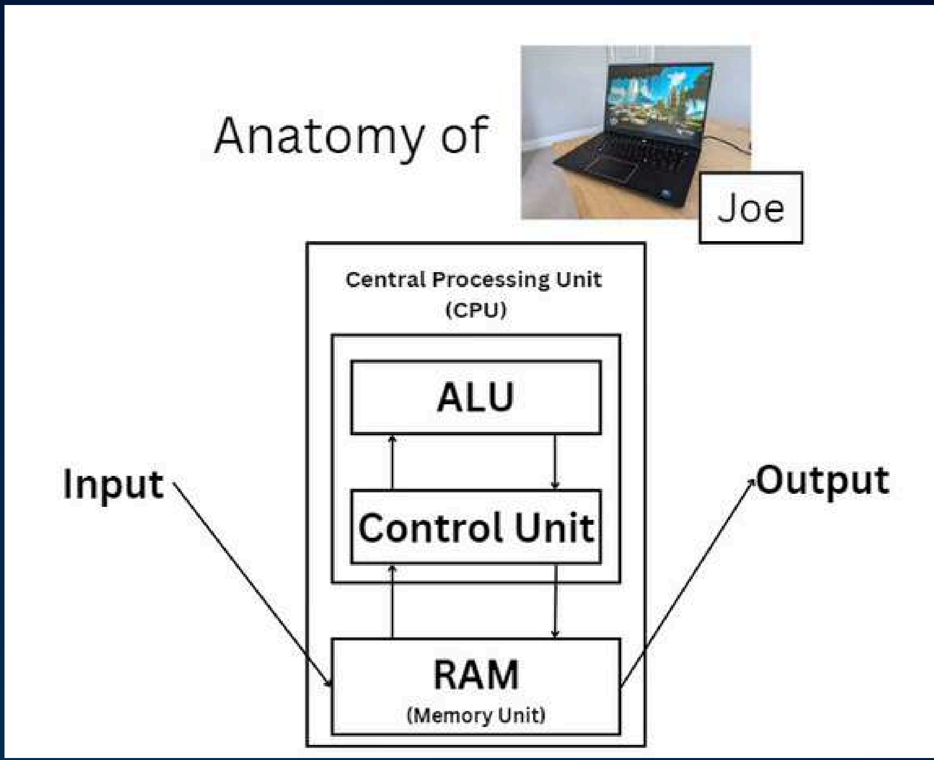
It's best to return to lesson 2 after the end of this lesson to ponder again how each one of the components work

I. A simple computer (ft. Joe)

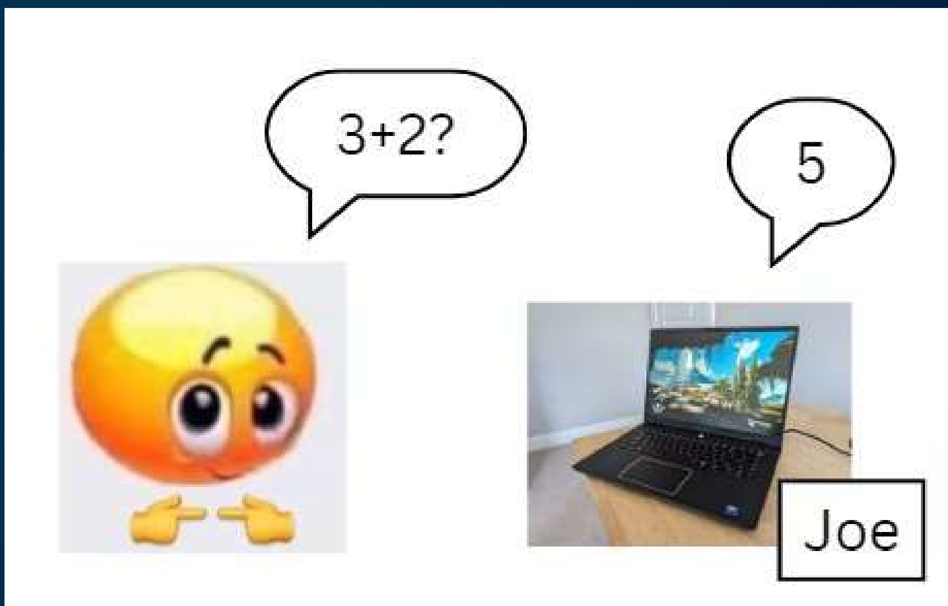
1.Overview

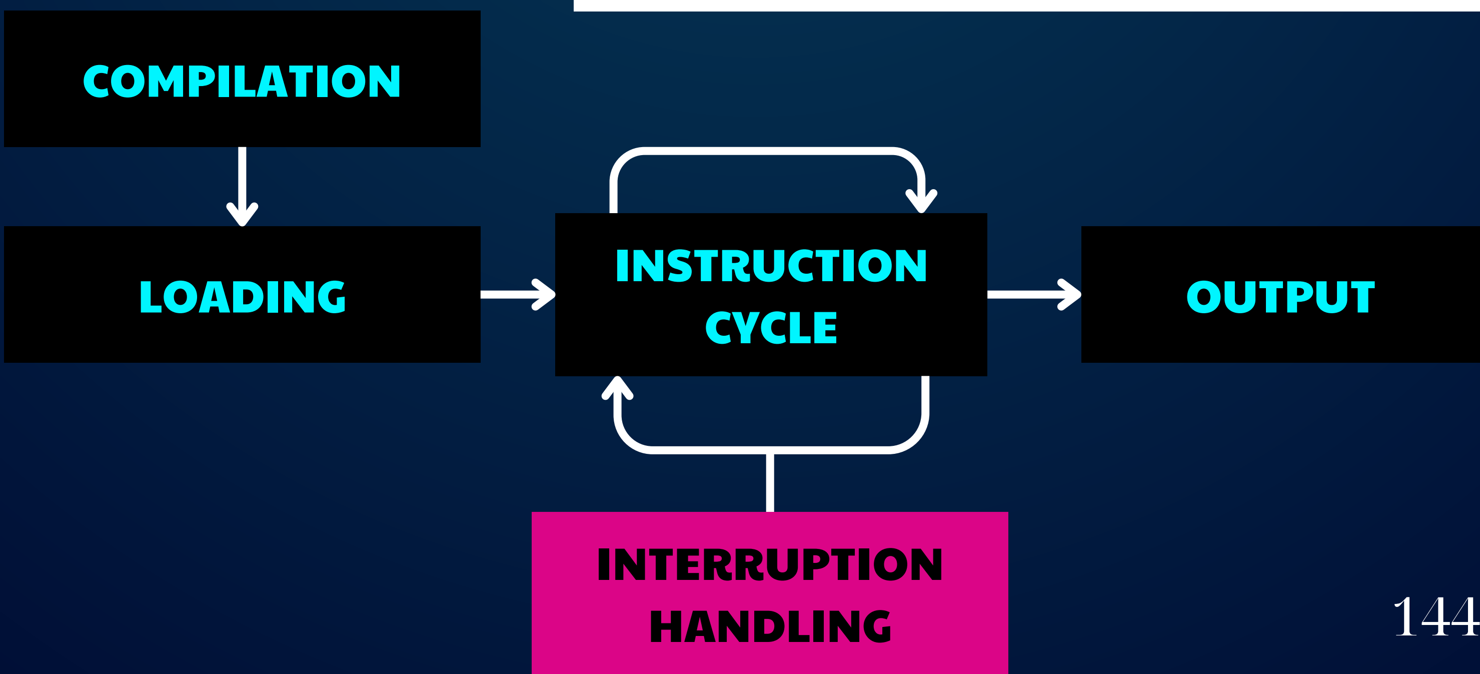
Remember this guy? He had appeared several times in the previous lessons, and today, he will be our main character for this lesson! We are going to name him Joe the simple computer.

As Joe is merely a simple computer, we're going to set his anatomy to be the von Neumann model. In this lesson, Joe will demonstrate to everyone how he can execute a basic math operation provided by the human.



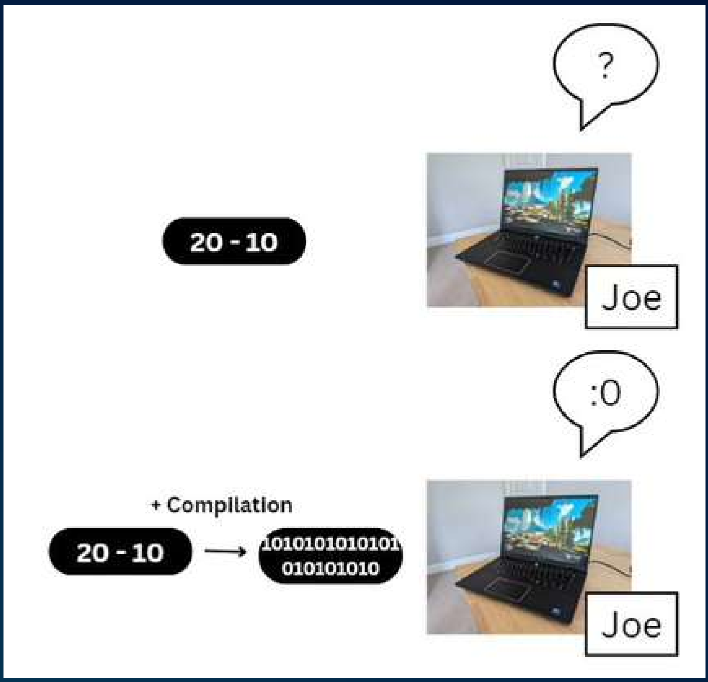
When we input, for example, "20 - 10" into Joe, he will go through all these processes to execute the program:

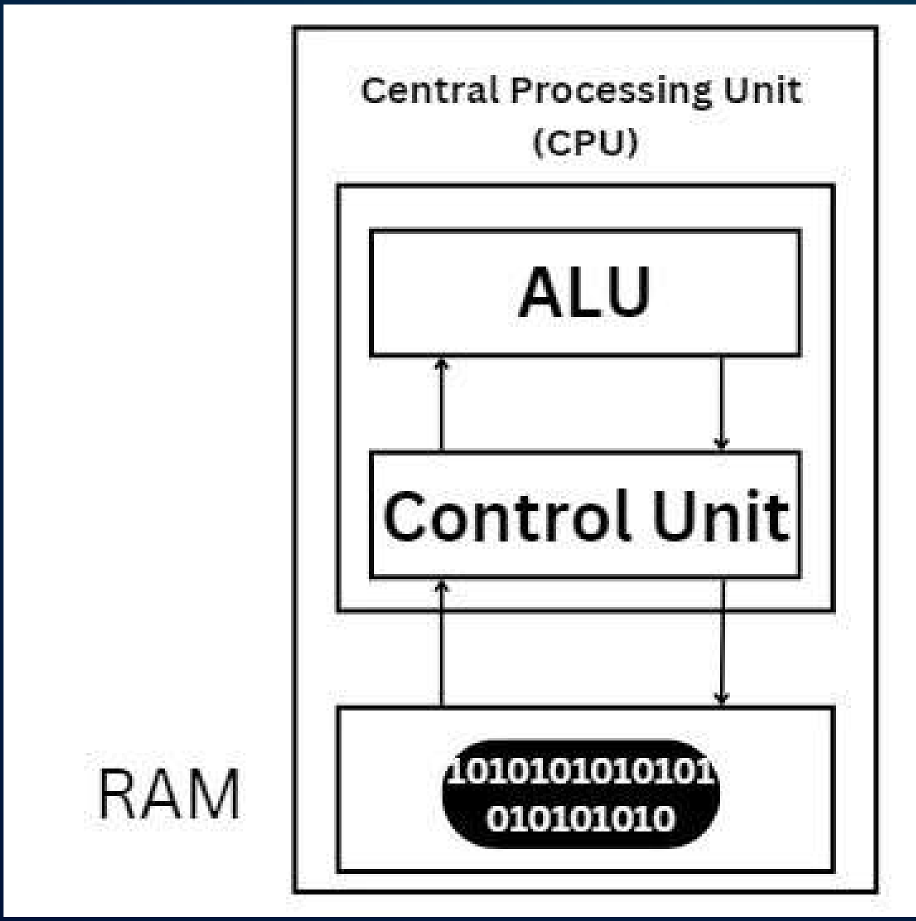




2.Compilation

The sad thing is, Joe can't understand anything we're saying at all. So to him, "20 - 10" is just a bunch of text, if we just immediately throw it into the CPU to process. That's why the instructions that we write for the machine have to be translated, or compiled into machine instructions in order for Joe to understand what he needs to do.



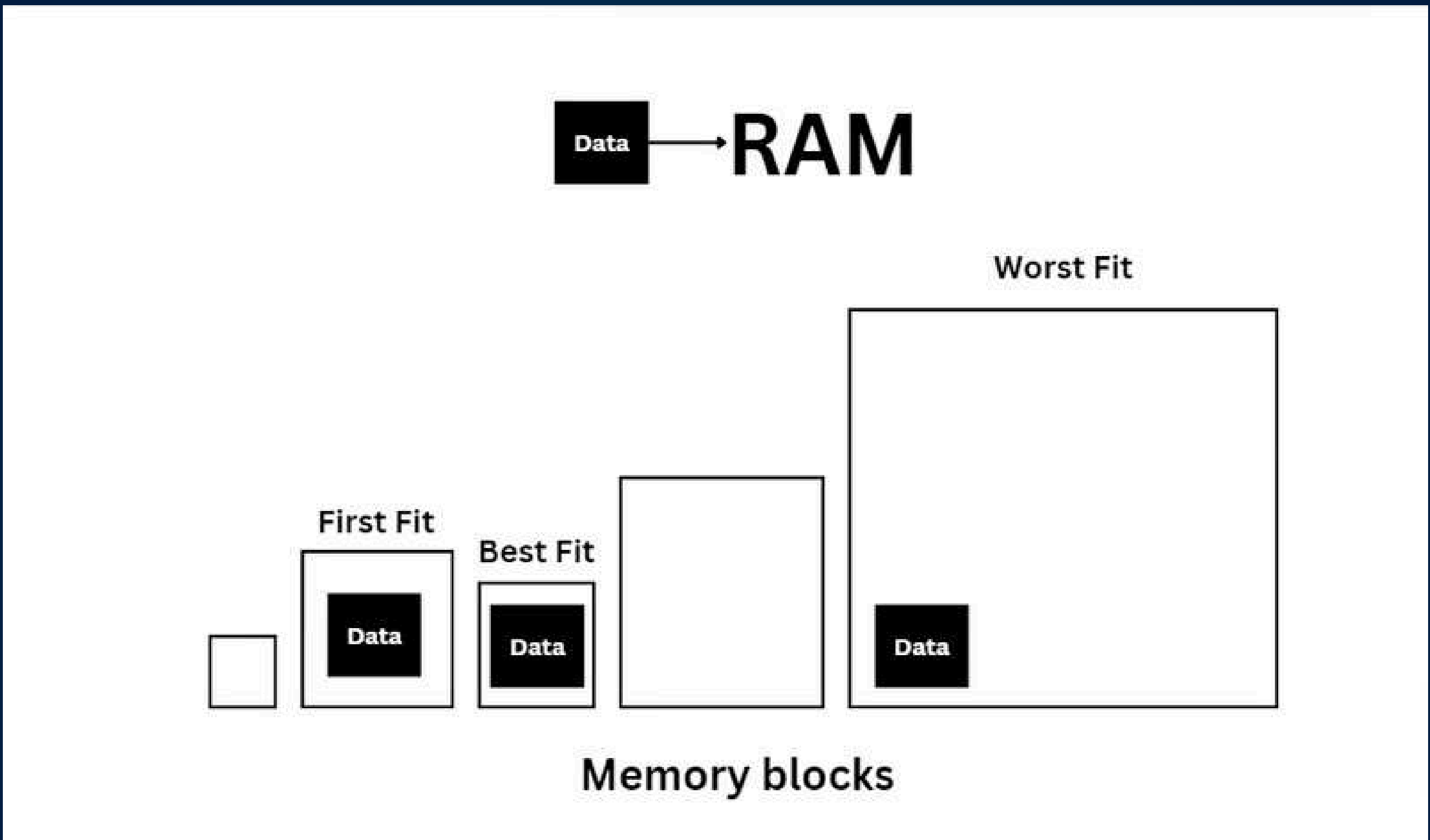


3.Loading

After compilation, the instructions will be loaded into the computer's RAM in order to be processed by the CPU.

3.Loading

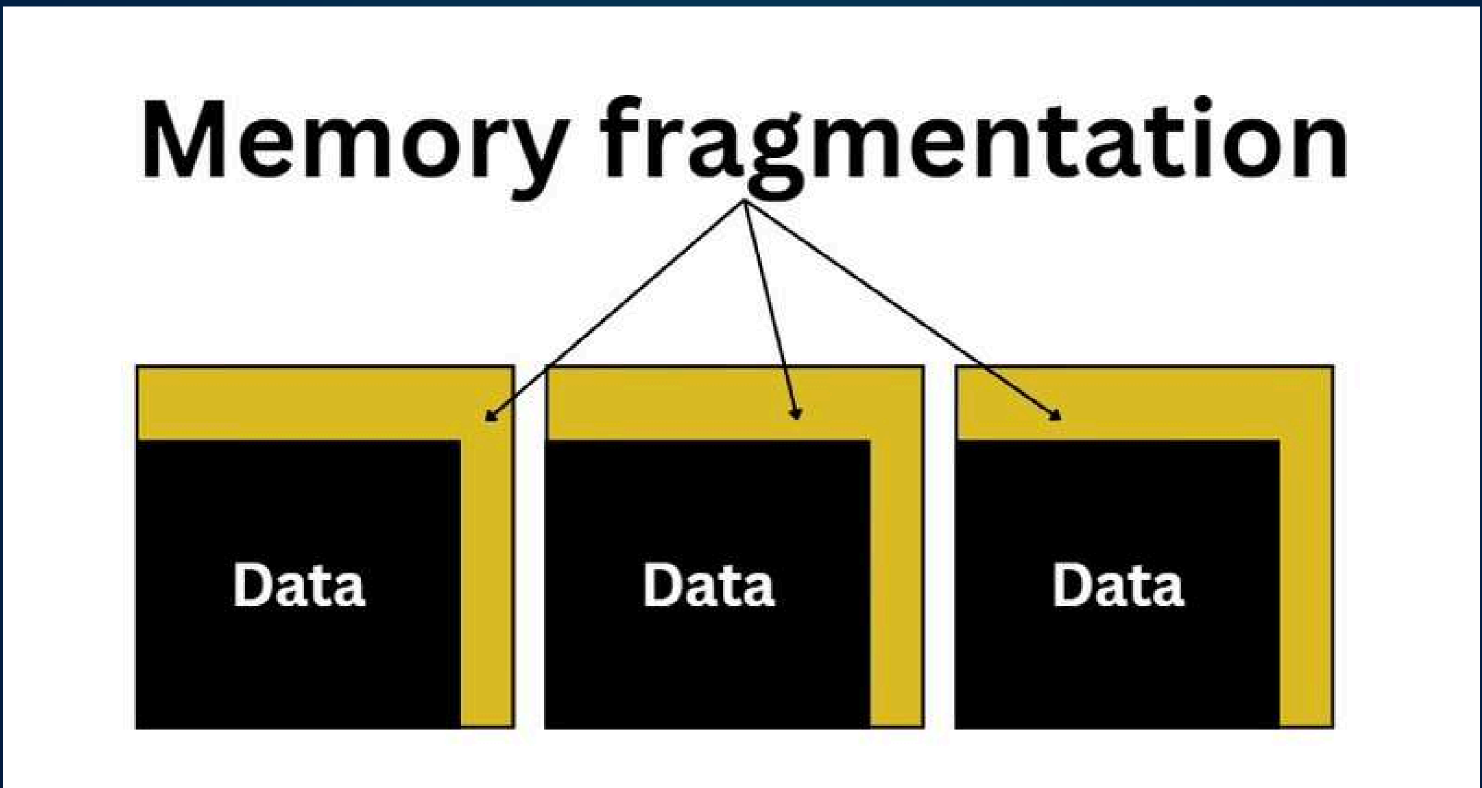
The RAM is usually divided into small and big memory blocks where your data can be allocated into. There are 3 ways the computer can do this: Best fit, first fit, and worst fit.



3.Loading

Sometimes, data stored across these blocks leave a tiny little space that's not enough to store any full block of data. This is called "memory fragmentation".

When this happens, sometimes the computer has to do something called "paging", where the data is too big to store in one memory block, so it gets split into different small blocks instead. This is bad for the computer because the computer has to access different memory blocks in order to access one data, which slows down the computer.



Therefore, the RAM has to carefully select between best fit, first fit, or worst fit. Remember, a memory block can contain multiple Data blocks as well, so sometimes, "worst" fit may not even be the worst fit.

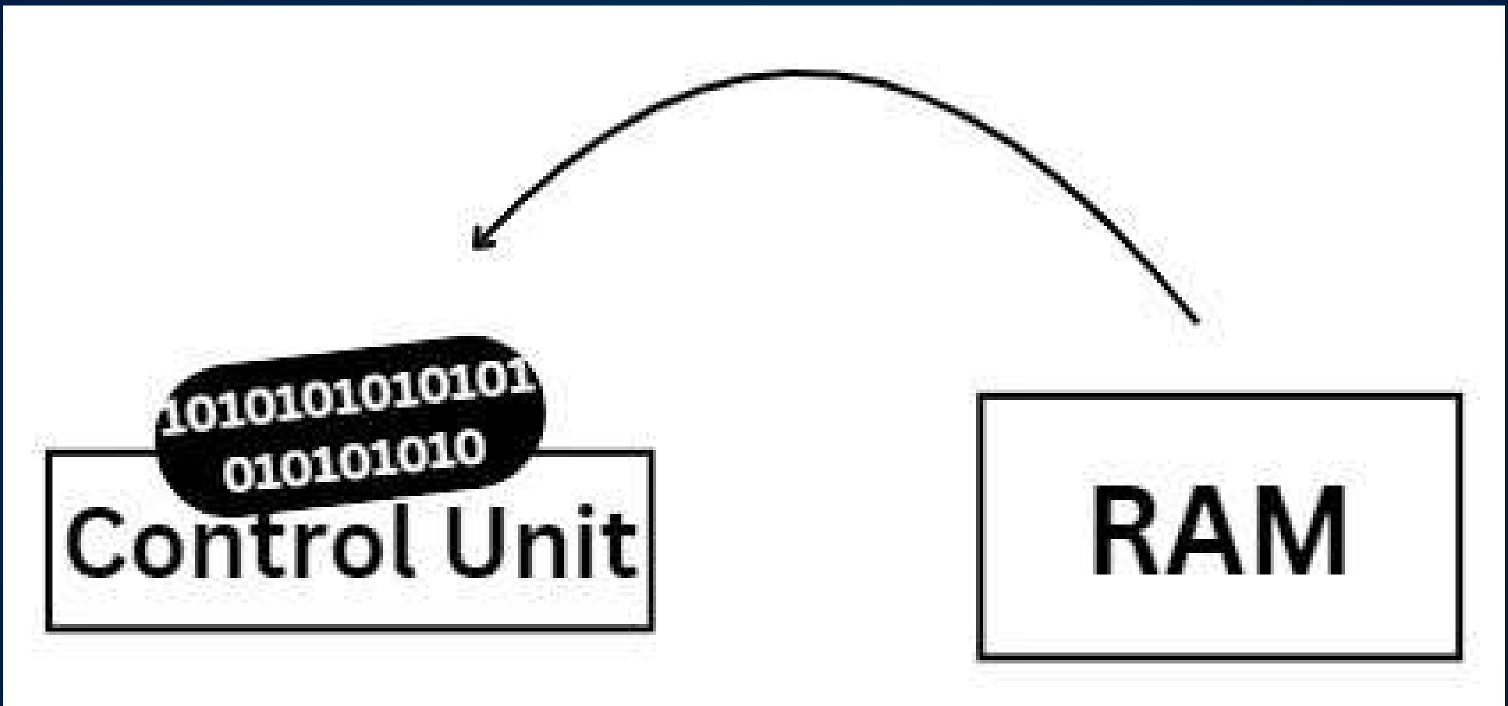
4.Instruction cycle

This is the part where the instructions we provided to the computer get executed. This part contains 3 small steps:

FETCH => DECODE => EXECUTE

Fetch

The Control Unit arrives at RAM’s house to collect the instructions.



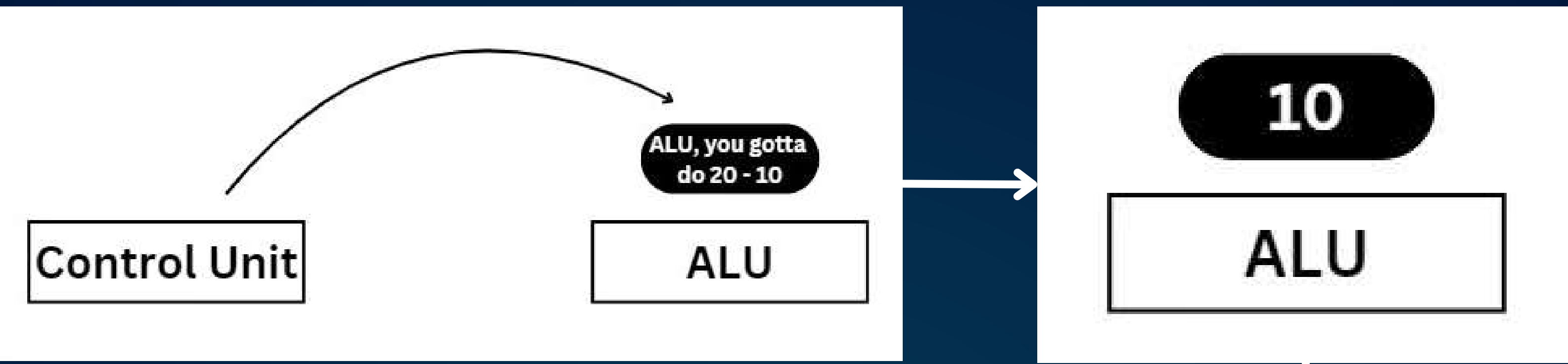
Decode

The Control Unit reads the instructions and make notes of what the ALU have to do

4.Instruction cycle

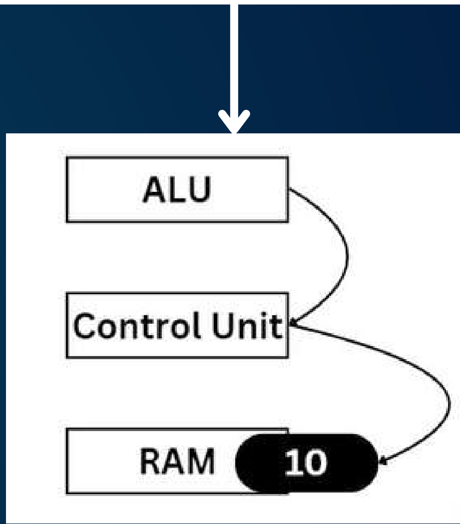
Execute

The Control Unit ships the instructions to the nerd ALU to be processed.

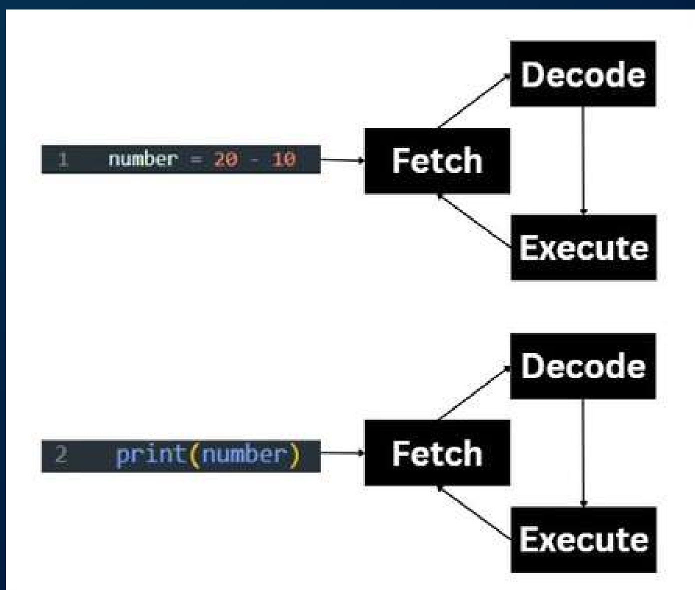


Repeat

The Control Unit takes the value that the ALU just processed, sends it back to the RAM, and then prepares to send the next instruction to the ALU.



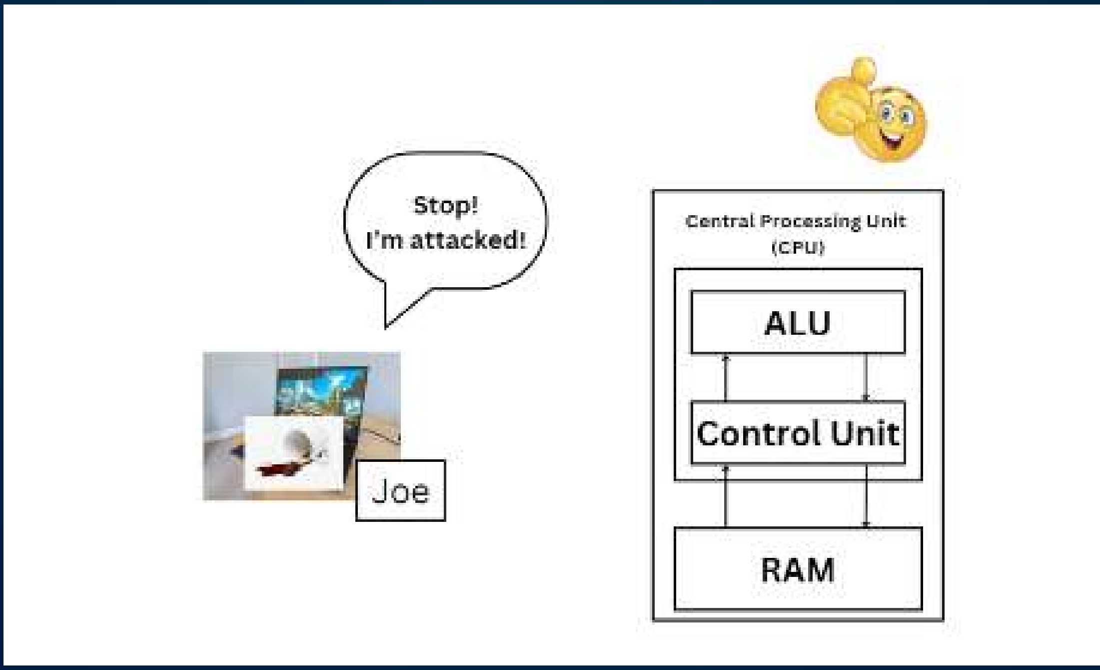
Usually, when we want the computer to do something, we will need not just one, but multiple instructions at once. This is why coding languages exist - so that people can communicate those instructions to machines better.



5.Interrupts

Interruption is a special case in the instruction cycle. And to demonstrate this, I will spill coffee on Joe. Oh no!

When the computer realizes that there's something wrong with its hardware or software system, or with the instruction that it's given, it will prioritize fixing the problem first. Meaning, it has to issue a stop order for the CPU for the current task, and divert its attention to the problem.



And after the problem has been resolved, the CPU will keep on going with its current task. Let's dry Joe up and see what happens.

6.Output

Lastly, when everything has been cleared up, Joe will output the answer to the question that we asked him in the first place. Let's give an applause for Joe, and apologize Joe for spilling coffee on him!

III. EXERCISES

Ex1. Multiple Choice

1.What does Joe need to do before he can understand and execute the instruction "20 - 10"?
a) Decode the instruction directly
b) Compile it into machine language
c) Store it in the ALU
d) Fetch it from the output device
Answer: b) Compile it into machine language

2.Which step involves the Control Unit reading instructions and determining what the ALU needs to do?
a) Fetch
b) Decode
c) Execute
d) Output
Answer: b) Decode

3.What term is used when data is split across multiple memory blocks due to lack of a large enough single block?
a) Interrupt handling
b) Compilation
c) Paging
d) Fragmentation
Answer: c) Paging

4.In the loading process, which memory allocation method is designed to fit data in the smallest available block?
a) Worst fit
b) Random fit
c) Best fit
d) First fit
Answer: c) Best fit

5.When Joe encounters an error during processing, which of the following occurs?
a) The ALU corrects the error automatically
b) The CPU stops the current task to handle the issue
c) The Control Unit skips the error and continues
d) The error is ignored until all tasks are completed
Answer: b) The CPU stops the current task to handle the issue

II. EXERCISES:

Ex2.Fill in the blanks

1.Before instructions can be processed, they must first be ________________ into machine language that Joe can understand.
Answer: compiled
2.During the ________________ phase, the Control Unit gathers instructions from the RAM.
Answer: Fetch
3.When data is divided into multiple blocks because it doesn't fit in a single block, this is called ________________.
Answer: paging
4.The ________________ is responsible for reading the instructions and sending them to the ALU for processing.
Answer: Control Unit (CU)
5.If Joe encounters a problem, an ________________ may be issued to pause the CPU and handle the issue before resuming.
Answer: interrupt

III. MIND MAP

-Megaherzt (MHz)-
Lesson 3: Program Execution
Overview
Compilation
Loading
A simple Computer
Outputs
Interrupts
Instruction

LESSON 4: HARDWARE FOR AI (COMING SOON)

CHAPTER D: NETWORKING FOR AI (WORK IN PROGRESS)

Chapter D: Networking for AI

I. Introduction to AI networking:

1.AI Networking: Definition & Importance

- **Definition: Integration of AI with network systems to boost performance, efficiency, and functionality.**
- **Importance: Enhances network resource management, communication, and intelligent decision-making.**

2.Key Components:

- **Data Communication: Crucial for sharing large datasets across networks, which is vital for AI applications.**
- **Distributed Systems: AI aids in managing systems where multiple agents work together, requiring robust networking.**

3.Machine Learning in Networking:

- **Applications: Used to analyze traffic, detect anomalies, and optimize routing, improving adaptability and performance.**

4.Challenges:

- **Issues: Latency, bandwidth, and security must be addressed to ensure reliable AI-driven network solutions.**

5.Future Directions:

- **Potential: AI could transform networking with advanced data analysis, predictive maintenance, and smart resource allocation.**
- **Research: Aims to develop more adaptive networks that respond dynamically to user needs and conditions.**

II. DATA TRANSFER AND LATENCY REQUIREMENTS

Role of Data Transfer in AI Systems

Importance: Efficient data transfer between processors, memory, and storage is vital for AI performance, particularly in deep learning, where large volumes of data must be moved. Poor management of data transfer can create bottlenecks.

Latency Issues:

Training Models: High latency during data transfers between storage (e.g., HDDs, SSDs), memory (RAM), and processors (CPUs, GPUs, TPUs) can significantly slow down training times, especially with large datasets or distributed training.

Inference: Low latency is crucial for real-time AI applications (e.g., autonomous driving, robotics). Delays in data transfer can hinder performance, making fast hardware design essential.

High-Bandwidth Data Transfer:

Memory Bandwidth: High-bandwidth memory (HBM) facilitates faster data transfer between processors and memory, crucial for training large neural networks.

Interconnects and Protocols: Efficient communication technologies (e.g., NVLink, PCIe, InfiniBand) enable high-speed data transfer, reduce latency, and support parallel processing in distributed systems.

Distributed Training and Data Parallelism:

Data Transfer Across Nodes: In distributed training across multiple GPUs or TPUs, fast data transfer is essential for synchronizing weights and gradients to avoid delays. Efficient networking and data handling are critical.

Minimizing Latency: Techniques like model parallelism and data parallelism help reduce data transfer time by distributing workloads across processors, allowing concurrent data processing.

Edge Computing:

Real-Time Processing: In edge AI applications, where data is processed near the source (e.g., sensors, cameras), low latency is vital for real-time analytics. Edge devices must have optimized data pathways and efficient processing units to minimize delays between data capture and action.

III. CLOUD COMPUTING AND AI

Definition and Integration
Cloud Computing: Provides scalable, on-demand computing resources over the internet, allowing users to access powerful hardware and software without local infrastructure.
Integration with AI: Enhances data processing capabilities and facilitates the deployment of AI models at scale.
Scalability
Dynamic Resource Management: Cloud platforms can easily adjust resources based on demand, making them ideal for AI applications that require significant computational power during model training and data processing.
Cost Efficiency
Financial Viability: Utilizing cloud services eliminates the need for large upfront hardware investments. Organizations can pay for resources as needed, making AI projects more affordable.

III. CLOUD COMPUTING AND AI

Collaboration and Accessibility
Centralized Environment: Cloud computing promotes collaboration by providing a centralized platform for developing, testing, and deploying AI applications. It enhances accessibility, allowing users to work from anywhere and access AI tools.
Data Storage and Management
Robust Solutions: Cloud platforms offer effective solutions for storing and managing large datasets, which are essential for training AI models. They also provide tools for data analysis and processing, streamlining AI development workflows.
Emerging Technologies
Advancements: The book highlights how innovations in cloud computing, such as edge computing and serverless architectures, are shaping the future of AI by improving data processing speeds and reducing latency.

IV. SECURITY CONSIDERATIONS FOR AI NETWORKS

Vulnerability to Attacks

Security Threats: AI networks face various threats, such as data poisoning, adversarial attacks, and model theft, which can compromise the integrity and reliability of AI systems.

Data Privacy

Protection Measures: Ensuring the privacy of sensitive data used in AI training and processing is critical. Security measures must be implemented to protect personal and confidential information from unauthorized access.

Robustness of Models

Model Security: AI models should be resilient against attacks that manipulate inputs to yield incorrect outputs. Continuous evaluation and enhancement of model security are essential.

Access Control

Strong Mechanisms: Implementing robust access control mechanisms, including authentication and authorization protocols, is crucial to prevent unauthorized access to AI systems and sensitive data.

Continuous Monitoring

Anomaly Detection: Regular monitoring of AI networks is necessary to detect anomalies and potential security breaches in real time, allowing for prompt responses to emerging threats.

Regulatory Compliance

Adherence to Standards: AI systems must comply with relevant regulations and standards concerning data security and privacy, which can vary by region and industry.

Collaboration for Security

Stakeholder Collaboration: Collaboration among developers, researchers, and organizations is vital for sharing knowledge and best practices to enhance the security of AI networks.