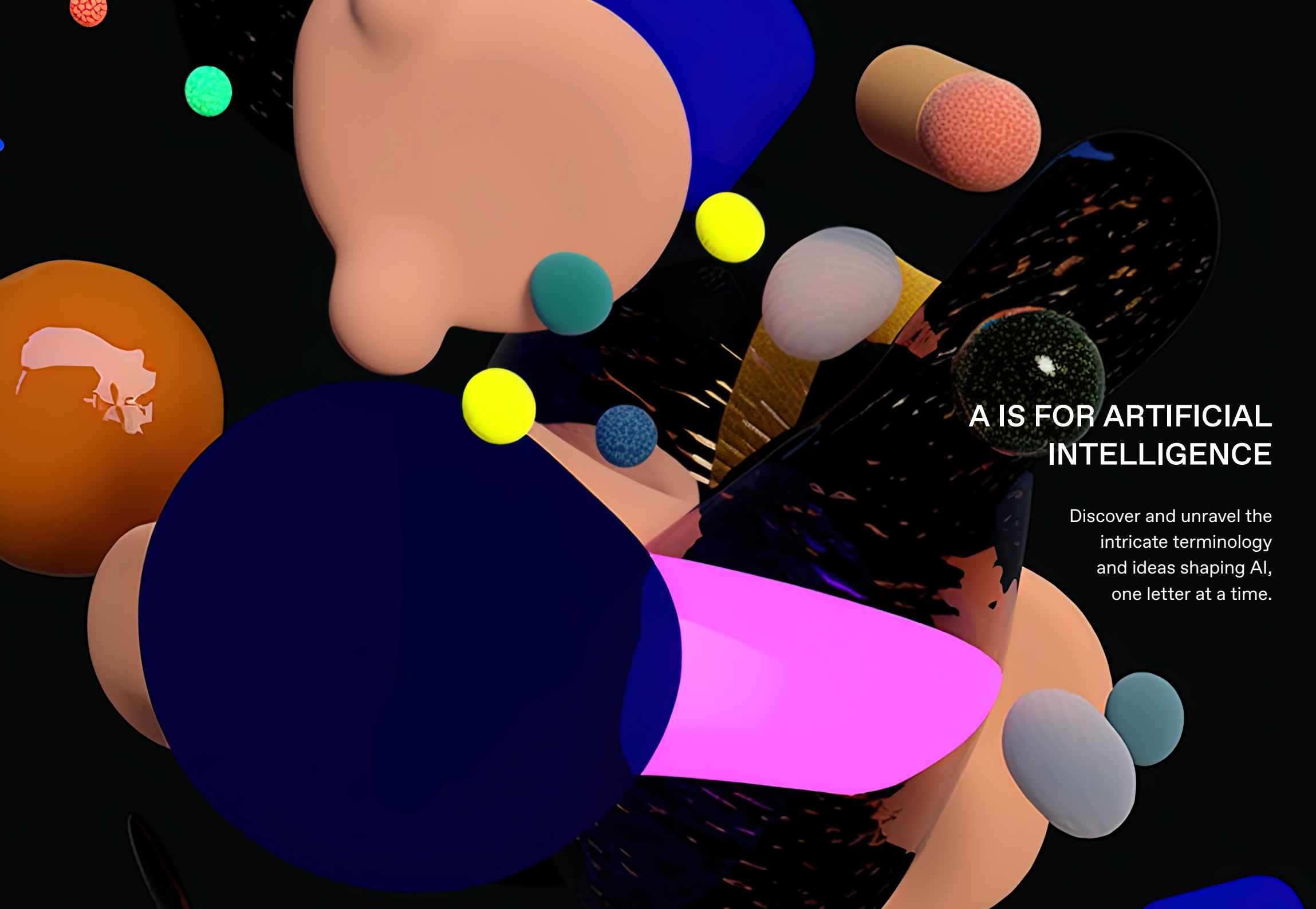


A IS FOR ARTIFICIAL INTELLIGENCE

Discover and unravel the intricate terminology and ideas shaping AI, one letter at a time.

A is for Artificial Intelligence

Discover and unravel the intricate terminology and ideas shaping AI, one letter at a time.

Concept and direction by Euro Beinat
and Marino Capitanio
for Prosus AI

Text by Prosus' AI Assistant
Images by Midjourney

October 2023
Made in Amsterdam

Contents

How it's made

How it's made

We used Prosus' AI Assistant for all text and Midjourney, the platform for image generation, for all illustrations.

We started by asking the AI Assistant to generate 10 ideas for "something" that illustrates the main concepts of AI and Large Language Models. From those suggestions we took the idea of an illustrated alphabet of AI.

We asked the AI Assistant to select 4 concepts related to AI and Large Language Models for each of the 26 letters of the alphabet. For each letter we selected one concept and asked the AI Assistant to explain the concept. The prompt used is: "Explain [here concept name] with a short paragraph in simple and plain English. Add a short description or example that shows how it works in practice. Then, provide an entertaining analogy to explain the concept in a different way."

We used 95% of the text produced by Prosus' AI Assistant as-is. We edited a small number of suggestions to shorten the text or avoid repetitions, for instance with the analogies. For the letter J we instead asked why there were comparatively less interesting concepts starting with that letter.

We then asked the AI Assistant to generate a catchy title for the booklet. The prompt used is: "You act as a creative director. Your task is to generate a title for an illustration booklet that contains 26 concepts related to AI and Large Language Models, one concept for each letter of the alphabet. Each letter is illustrated by a colorful artwork. Suggest 10 possible titles". From the suggestions we selected the reasonably expected "A is for Artificial Intelligence".

The monogram illustrations were developed with Midjourney, based on the following prompt: "Create a white and gold monogram featuring a single capital letter [here letter], superimposed on a black coat of arms. The letter [here letter] should reflect a whimsical wilderness, Victorian Gothic, Milleniwave, detailed world-building, absinthe culture, and organic design aesthetic"

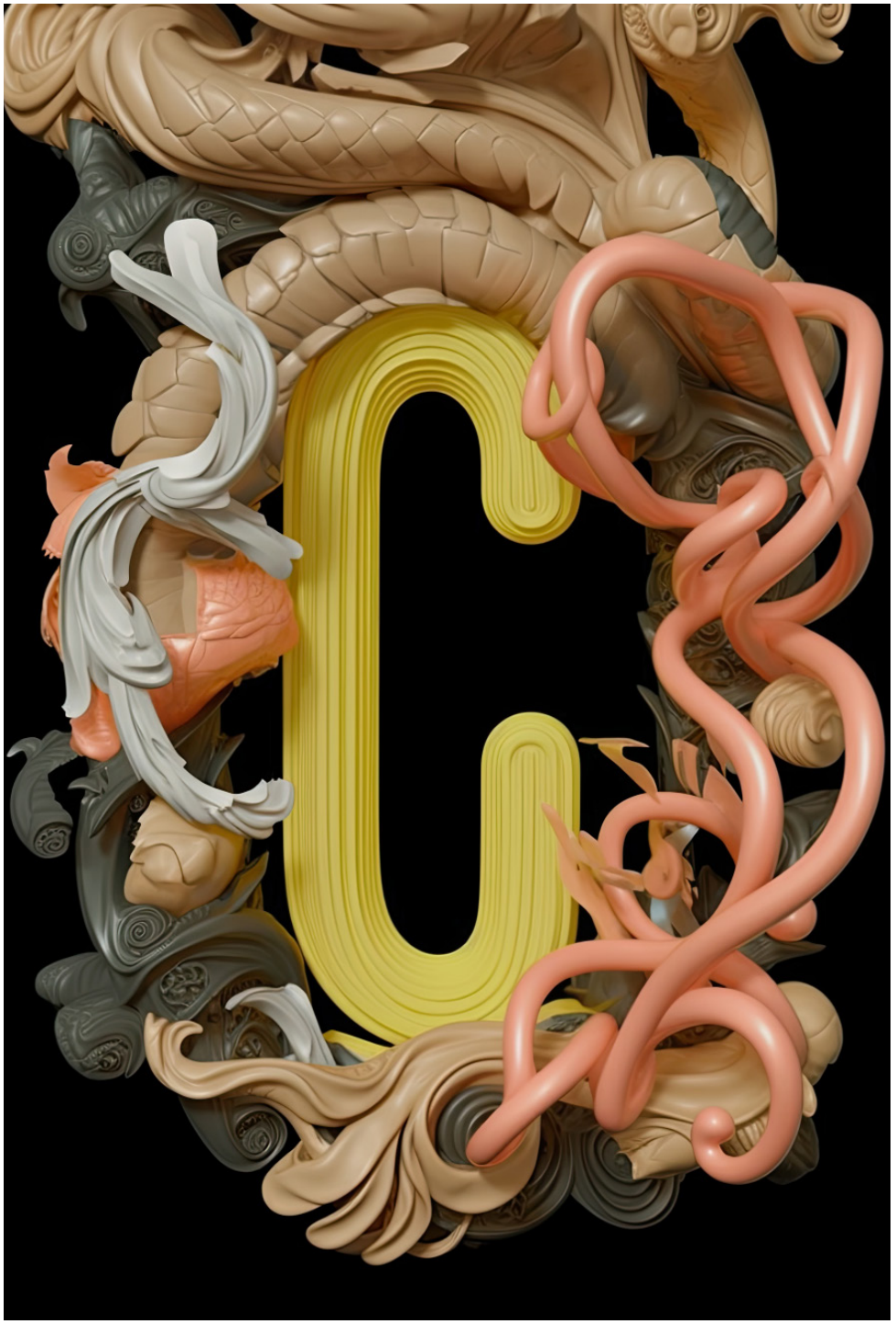
Four iterations were generated for each letter, and the most legible variant was chosen. The selected design was then remixed in Midjourney with an abstract 3D image to create the final, definitive monogram look. No edits were made to the final images. They were upscaled to achieve the required resolution for printing.

Attention Mechanism /əˈtɛnʃən ˈmɛkənɪzəm/

Attention mechanisms help neural networks focus on the most relevant parts of the input data when making predictions. They weigh the importance of different input elements and prioritize the most significant ones. Attention mechanisms are used in natural language processing tasks like machine translation, where the model needs to focus on specific words in the source sentence to generate the correct translation. Imagine you're at a party, and you're trying to listen to your friend's story. The attention mechanism is like your brain's ability to focus on your friend's voice while filtering out the background noise.

Back-propagation /ˌbæk prɒpəˈgeɪʃən/

Backpropagation is a way for models to learn from their mistakes by adjusting their internal parameters based on the difference between their predictions and the correct answers. In practice, backpropagation is used in training neural networks to minimize the error and improve their performance. Backpropagation is like a coach giving feedback to a sports team. The coach identifies the mistakes made during the game, and the team adjusts their strategy to improve their performance in the next match.



Curriculum Learning /kəˈrɪkjʊləm ˈlɜːrnɪŋ/

Curriculum learning is a training strategy in which a model is first trained on simpler tasks before gradually moving on to more complex tasks. This approach can improve the model's learning efficiency and generalization. In natural language processing, a model might first learn to predict the next word in a sentence before moving on to more complex tasks like sentiment analysis or machine translation. Curriculum learning is like teaching a child to swim by first using floaties, then moving on to shallow water, and finally deep water, gradually increasing the difficulty as they become more confident and skilled.

Deep Learning /diːp ˈlɜːrnɪŋ/

Deep learning is a subfield of machine learning that focuses on neural networks with many layers, allowing them to learn complex patterns and representations from large amounts of data. Deep learning has been used to achieve state-of-the-art results in tasks like image recognition, natural language processing, and speech recognition. Deep learning is like a multi-layered cake, where each layer contributes to the overall flavor and complexity, allowing it to capture intricate patterns and tastes.

Evaluation metrics /ɪˌvæljʊˈeɪʃən ˈmɛtrɪks/

Evaluation metrics in large language models are used to measure how well these models understand and generate human-like text. They help us determine the model's accuracy, fluency, and relevance when it comes to understanding and producing text. One common metric is called "perplexity," which measures how well the model predicts the next word in a sentence. Lower perplexity scores indicate better performance. For example, if a model is given the sentence "The cat is on the _," it should predict the word "mat" with high probability. If it does, the perplexity score will be low, indicating good performance.

Fine-tuning /faɪn 'tjuːnɪŋ/

Fine-tuning in large language models is a process where a pre-trained model, which has already learned general language patterns, is further trained on a specific dataset to adapt to a particular task or domain. For example, imagine a model that has learned English from various sources. To make it an expert in medical terminology, you would fine-tune it on a dataset of medical texts. This way, the model becomes more accurate and useful for medical professionals. Fine-tuning is like a professional athlete who has already built a strong foundation of skills and fitness, and then focuses on specific techniques and strategies to excel in their chosen sport.

Generative Adversarial Networks (GANs) /gænz/

GANs are a type of deep learning model that consists of two neural networks, a generator and a discriminator, which compete against each other. The generator creates fake data, while the discriminator tries to distinguish between real and fake data. GANs can be used to generate realistic images, such as creating new artwork or enhancing low-resolution photos. GANs are like a forger trying to create a convincing fake painting, while an art expert tries to detect the forgery. As the forger gets better, the expert becomes more discerning, and the process continues until the fake paintings are nearly indistinguishable from the real ones.

Hyper-parameters /ˌhaɪpərˈpærəmɪtərz/

Hyperparameters are the adjustable settings in a machine learning model that control its learning process. Examples include learning rate or the number of layers in a neural network. Selecting the right hyperparameters can significantly improve a model's performance. Imagine you are baking a cake. The recipe (algorithm) tells you the ingredients and steps to follow, but you can still tweak certain aspects, like the baking time or oven temperature (hyperparameters), to get the perfect cake. Just like in machine learning, finding the right balance of these settings can lead to the best results.

Inference /ˈɪnfərəns/

Inference refers to the process of generating predictions or responses based on the input provided. Large Language Models have been trained on vast amounts of text data, learning patterns and relationships between words and phrases. When you give the model a prompt, it uses its knowledge to generate a relevant and coherent response. For example, if you ask the model, “What is the capital of France?”, it will not retrieve the answer from a database of capitals, but instead use its understanding of language and the relationships it has learned to generate the answer, “Paris.”

J

/j/

There are comparatively less terms or concepts related to large language models that start with the letter “J”. This is mostly coincidental. The development of language models and their associated terminology is influenced by various factors, such as the history of the field, the researchers involved, and the techniques used. It just so happens that the letter “J” is less common in the names of these concepts, but this doesn’t have any significant impact on the field itself.

KL divergence /ˌkeɪ ˈɛl dɪˈvɜːdʒəns/

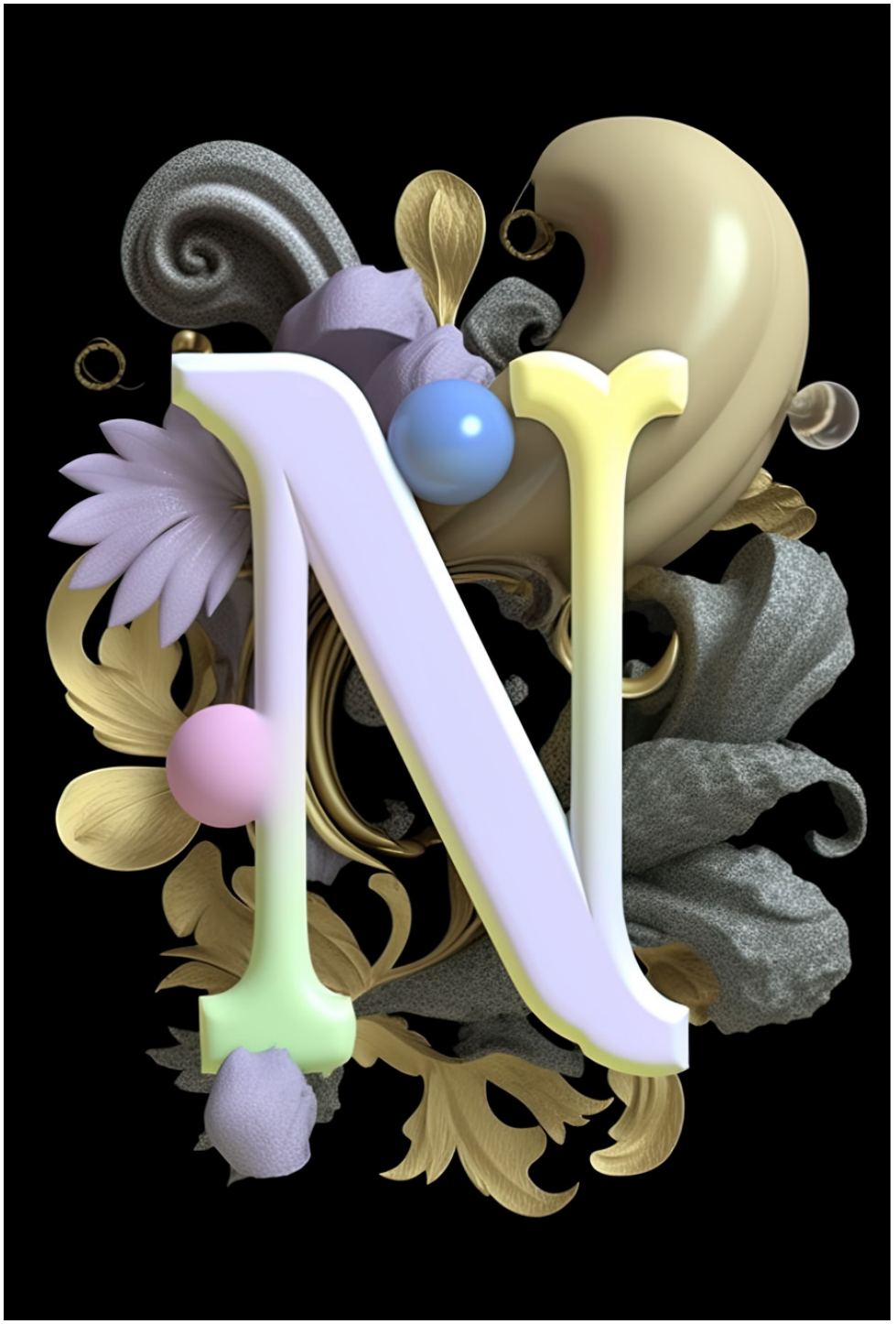
KL divergence, or Kullback-Leibler divergence, is a measure used to compare two probability distributions. In large language models, it quantifies the difference between what the model expects and what actually occurs. For example, imagine a model trying to predict the next word in a sentence. If the model's predicted word distribution is very similar to the actual distribution of words in that context, the KL divergence will be low, indicating a good match. On the other hand, if the model's predictions are far off from the actual distribution, the KL divergence will be high, signaling a poor match. Think of KL divergence as a game of darts. The actual word distribution is the bullseye, and the model's predictions are the darts thrown by a player. If the player consistently hits close to the bullseye, the KL divergence is low, meaning the player (or model) is doing a great job. However, if the darts are scattered all over the dartboard, the KL divergence is high, indicating that the player (or model) needs to improve their aim to better match the target (actual word distribution).

Loss Function /lɒs ˈfʌŋkʃən/

A loss function is a mathematical way to measure the difference between the predicted output and the actual output (or target) in a machine learning model. It helps us understand how well our model is performing. The goal is to minimize this difference, so the model's predictions become more accurate.
Think of the loss function as a game of "hot and cold." You're trying to find a hidden object in a room, and a friend tells you if you're getting "hotter" (closer) or "colder" (further away) from the object. The loss function is like your friend's feedback, guiding you to the correct location (the best model parameters) by telling you how far off you are from the target.

Multi-task learning /ˈmʌlti tɑːsk ˈlɜːrnɪŋ/

Multi-task learning in large language models is a technique where a single model learns to perform multiple tasks at the same time. This helps the model to improve its overall understanding and performance by sharing knowledge across different tasks. In practice, a model might learn to translate languages, answer questions, and summarize text all at once, which can lead to better results in each task. Imagine a musician who can play multiple instruments like guitar, piano, and drums. By practicing all these instruments together, the musician develops a deeper understanding of music and becomes better at each instrument. Similarly, multi-task learning allows a language model to become more proficient in various tasks by learning them together.



Natural language understanding (NLU) /ɛn.ɛl.juː/

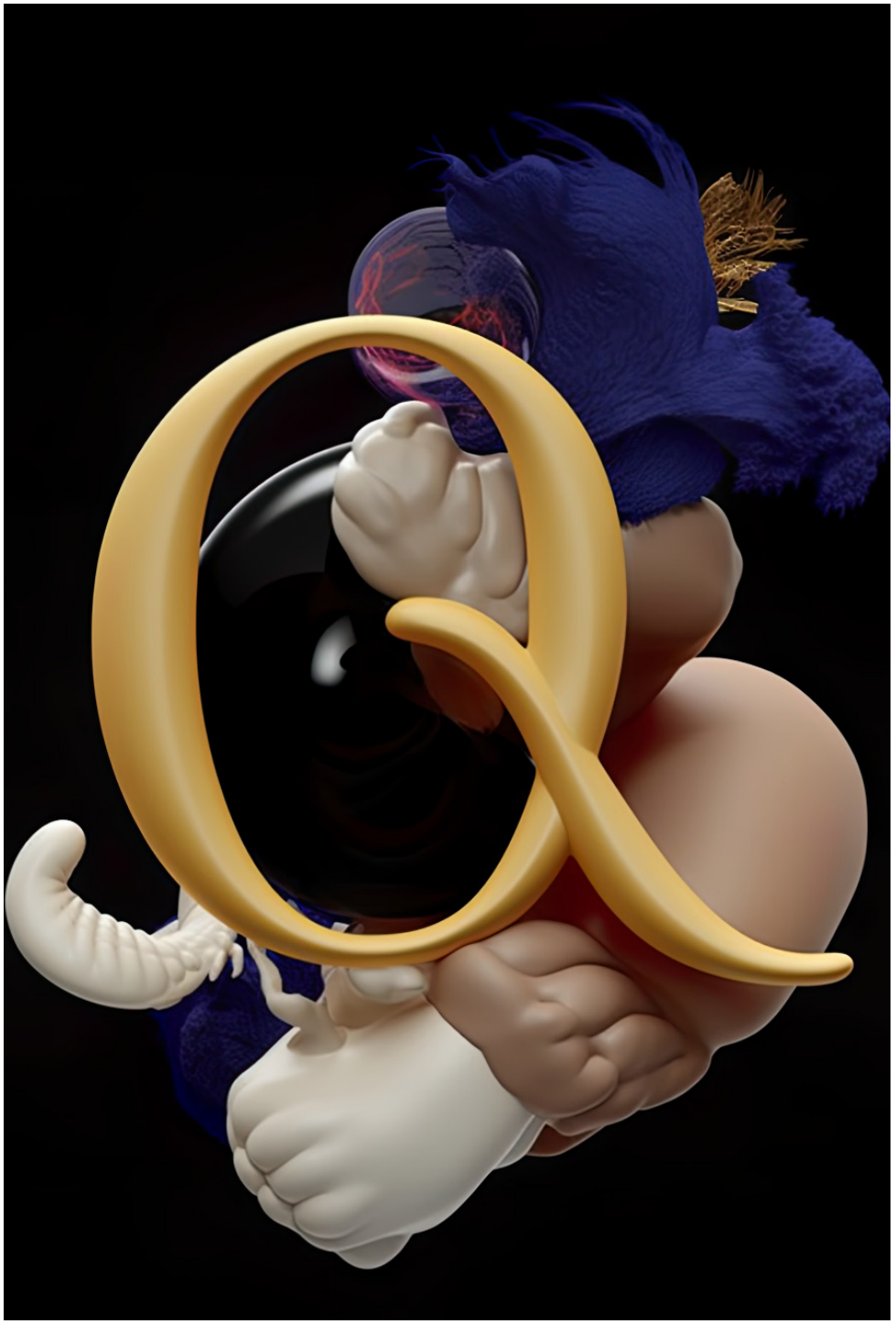
Natural language understanding (NLU) in large language models refers to the ability of these models to comprehend and interpret human language. They analyze text, identify patterns, and extract meaning from it. This helps them respond to questions, generate text, and perform various tasks that involve language. For example, when you ask a language model to summarize a news article, it reads the text, understands the main points, and generates a concise summary for you. Imagine our language model is a skilled doctor who can diagnose illnesses by listening to patients' symptoms. The patients' words are like the text input, and the doctor's understanding of their condition is like the NLU. Just as the doctor prescribes the right treatment based on their understanding, the language model generates appropriate responses or actions based on its understanding of the text.

Overfitting /ˌəʊvəˈfɪtɪŋ/

Overfitting in AI occurs when a model learns the training data too well, capturing not only the underlying patterns but also the noise or random fluctuations. As a result, the model performs poorly on new, unseen data because it's too focused on the specific details of the training data rather than generalizing to new situations. Imagine a carpenter who's great at making one specific type of chair, following a precise blueprint. They've become so good at making this chair that they can reproduce it perfectly every time. However, when asked to make a different type of chair, they struggle because they've only learned the specific details of that one blueprint. This is similar to overfitting in AI, where the model becomes too specialized in the training data and fails to generalize to new situations.

Pretraining
/ˌpriːˈtreɪnɪŋ/

Pretraining in large language models is like teaching a model the basics of a language before it starts solving specific tasks. It involves feeding the model a massive amount of text from various sources, so it learns grammar, vocabulary, and some general knowledge. This way, the model gets a solid foundation in the language, making it easier to fine-tune for specific tasks later on. Pretraining a language model is like teaching a young child the alphabet, basic words, and simple sentences. Once the child has a grasp of the language, they can learn more complex tasks like writing essays or solving math problems using words. The pre-training phase is like the early years of learning, where the child absorbs the fundamentals of language, making it easier to build upon those skills later in life.



Quantization /kwɒntʌɪ-zeɪʃən/

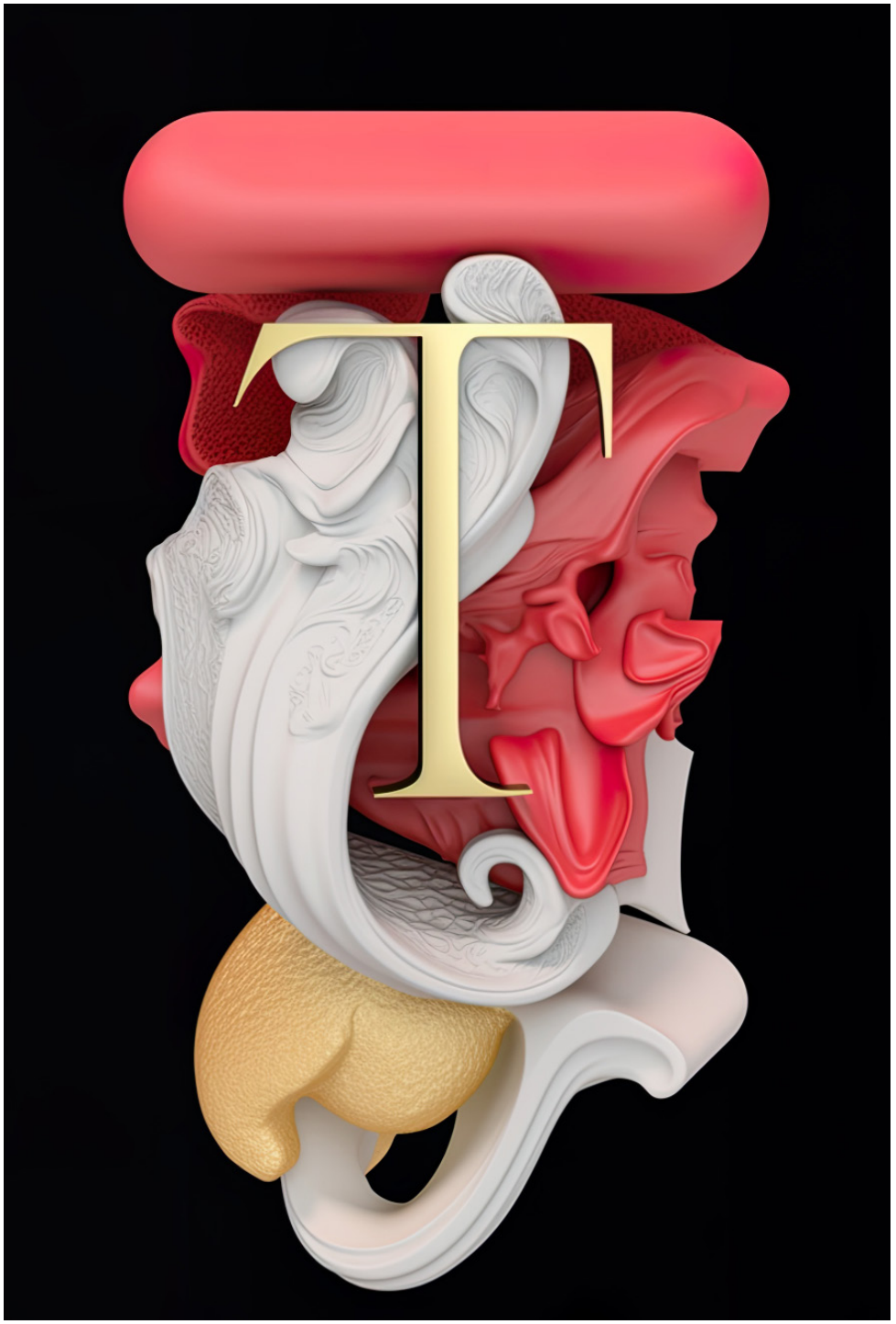
Quantization in machine learning is like turning a high-definition photo into a smaller file size. It simplifies the information by reducing the precision of the numbers used in an AI model, similar to how a photo might be compressed to take up less space. For instance, a model might use 8-bit numbers instead of 32-bit ones, which makes it lighter and quicker to run, though slightly less detailed. Imagine quantization as packing a suitcase with only your favorite clothes instead of your entire wardrobe. You still have what you need for your trip, but your bag is lighter and easier to manage. However, just as you have to be careful to pack the right clothes to have enough options for your trip, in quantization, you have to reduce the precision of the model carefully to make sure it still performs well.

Reinforce-ment learning /ˌriːɪnˈfɔrsmənt ˈlɜːrnɪŋ/

Reinforcement learning is a type of machine learning where an AI agent learns to make decisions by interacting with its environment. The agent receives feedback in the form of rewards or penalties, which helps it improve its decision-making over time. In large language models, reinforcement learning can be used to fine-tune the model's ability to generate more accurate and relevant text based on the user's input. For example, imagine a language model that's learning to write poetry. It starts by generating random lines of text. When it creates a line that resembles a good poem, it receives a reward. Over time, it learns to generate better poetry by maximizing the rewards it receives. Think of reinforcement learning like training a dog to catch a frisbee. At first, the dog might not know how to catch it, but as you keep throwing the frisbee and rewarding the dog with treats when it catches it, the dog learns to improve its catching skills. Similarly, the AI agent learns to make better decisions by receiving rewards for its good actions and penalties for its bad ones.

Sequence-to-sequence models /ˈsiːk-wəns tə ˈsiːk-wəns ˈmɒdəlz/

Sequence-to-sequence models in AI are a type of neural network that can transform one sequence of data into another sequence. They're often used for tasks like language translation, where the input is a sentence in one language and the output is the same sentence translated into another language. For example, a sequence-to-sequence model could take the English sentence "How's it going?" and output the Italian translation "Come va?" Imagine a choreographer teaching a dance routine to a group of dancers. The choreographer demonstrates the steps in one dance style (like ballet), and the dancers must learn to perform the same routine in a different dance style (like hip-hop). The sequence-to-sequence model is like the choreographer, taking the input (ballet steps) and transforming it into a new output (hip-hop steps) while maintaining the essence of the original routine.



Transformer /trænsˈfɔːmər/

The transformer architecture is a type of AI model designed to handle sequences of data, like sentences, by paying attention to different parts of the input at once. This “attention” mechanism allows the model to understand the context and relationships between words more effectively. Large language models are built using this architecture and can generate human-like text, answer questions, and write code. Think of the transformer architecture as a group of detectives trying to solve a mystery. Each detective focuses on a different clue (word) and shares their findings with the others. By working together and paying attention to each other’s insights, they can piece together the entire story (sentence) and solve the mystery (understand the meaning).

Universal Approximation Theorem /ju.eɪ.ti/

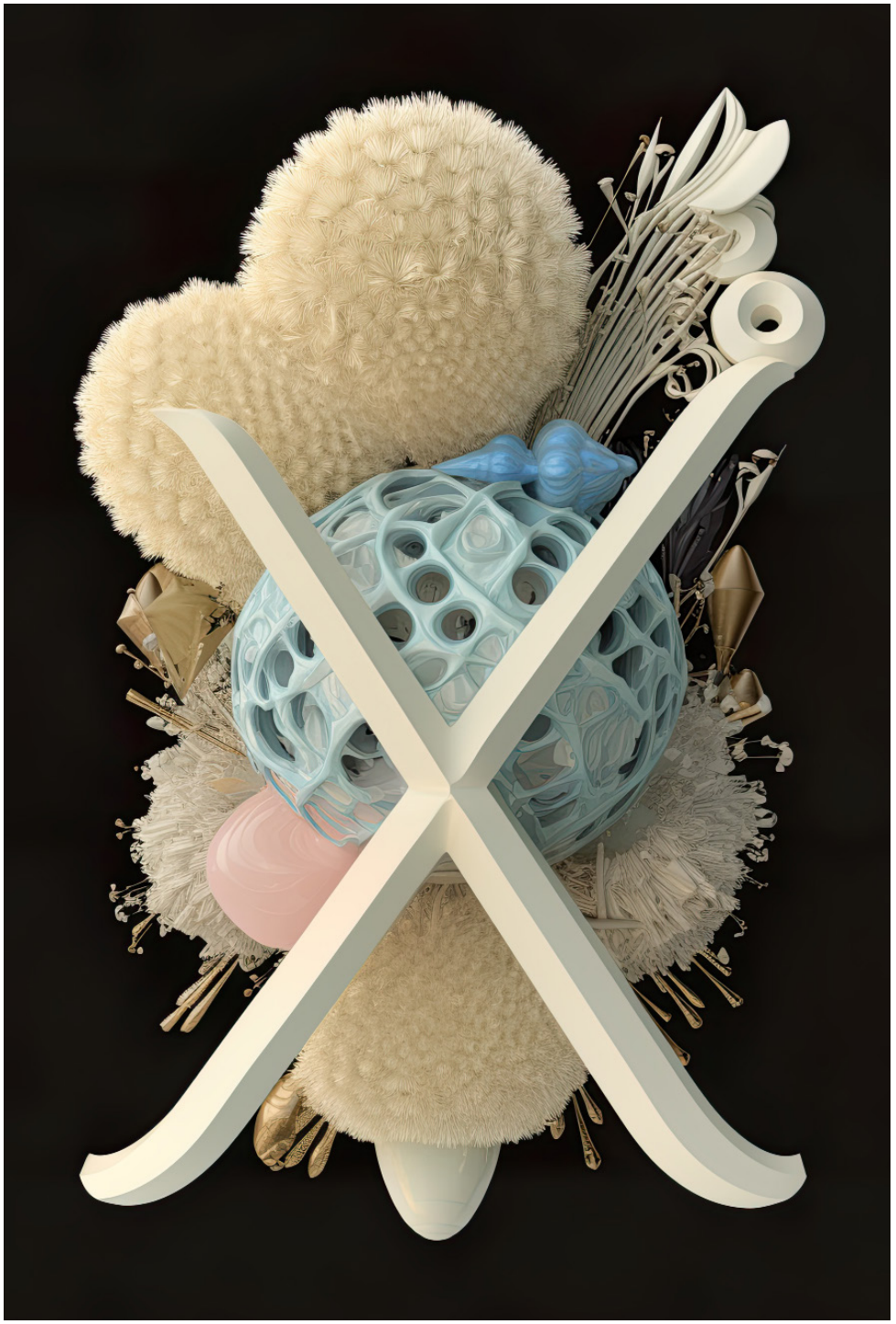
The Universal Approximation Theorem states that a neural network, specifically a feedforward network with at least a single hidden layer, can approximate any continuous function to a desired level of accuracy. In simpler terms, it means that a neural network can learn to mimic any pattern or behavior, given enough neurons in the hidden layer and proper training. Imagine you have a dataset of house prices based on various features like size, location, and age. A neural network can learn to predict the price of a house based on these features, even if the relationship between the features and the price is complex. Think of the neural network as a versatile artist who can mimic any painting style. The Universal Approximation Theorem is like saying that, given enough colors and brushes (neurons), the artist can recreate any masterpiece, no matter how intricate or unique the style is.

Visual question answering /ˈvɪʒʊəl ˈkwɛs.tʃən ˈænsərɪŋ/

Visual question answering (VQA) in AI is a process where an AI model can understand and answer questions about images. Large language models, like the one I'm based on, can process both text and visual information to provide accurate answers. In practice, you might show the AI an image of a cat sitting on a chair and ask, "What is the animal doing?" The AI would then analyze the image and respond with something like, "The cat is sitting on a chair." Imagine a soccer match where the AI is the coach. Then show a still image of the ongoing game and ask the coach, "Which player just scored a goal?" The coach (AI) observes the game (image) and answers the question based on their understanding of the situation.

Word embeddings /wɜːd ɪm'bedɪŋz/

Word embeddings are a way to represent words as numerical values, or vectors. By converting words into numbers, the model can perform mathematical operations to find similarities and differences between them. For example, imagine we have three words: “cat”, “dog”, and “car”. The model might assign the following vectors: cat = [1, 2], dog = [1.5, 2.5], and car = [5, 1]. The closer the vectors are, the more similar the words. In this case, “cat” and “dog” are closer to each other than “car”, indicating they are more related. Embeddings are like a magical librarian who can instantly organize the books on a single shelf based on their content. Books with similar topics would be placed close together, while unrelated books would be farther apart. This makes it easier for you to find and compare books on the same subject.



XAI
/eks.eɪ.aɪ/

XAI (Explainable AI) is a branch of artificial intelligence that focuses on making AI systems more understandable and transparent to humans. It aims to provide clear explanations of how AI models make decisions, so people can trust and effectively use these systems. In practice, XAI might involve breaking down an AI's decision-making process into simpler steps or visualizing the factors that contribute to a decision. For example, in finance, an AI system might be used to predict whether a loan applicant is likely to default. XAI could help explain the AI's decision by showing which factors, such as credit score or income, were most important in determining the outcome.

YOLO

/ˈjəʊləʊ/

YOLO (You Only Look Once) is a real-time object detection algorithm that can quickly identify objects in images. Unlike other methods that scan an image multiple times, YOLO looks at the entire image in a single pass, making it faster and more efficient. In practice, YOLO divides an image into a grid and predicts the objects and their locations within each grid cell. This allows it to detect multiple objects simultaneously. Imagine you're trying to find flowers in a garden. Traditional object detection methods would be like examining each flower one by one, which can be time-consuming. YOLO, on the other hand, is like taking a single glance at the entire garden and instantly knowing where all the flowers are and what types they are.

Zero-shot learning /ˈzɪərəʊ ʃɒt ˈlɜːrnɪŋ/

Zero-shot learning in AI refers to a model's ability to understand and perform tasks it hasn't been explicitly trained on. Large language models can do this by leveraging their vast knowledge and understanding of language patterns. They can make connections between different concepts and generate relevant responses even without specific training examples. For example, if a model is trained on various tasks like translation and summarization, it might still be able to answer questions about a new topic, like gardening, without having seen any examples of gardening questions during training. Imagine a skilled musician who can play multiple instruments. They've never played a specific song before, but they know the chords and scales. When asked to play that song, they can figure it out and perform it well, even without having practiced it before.

END