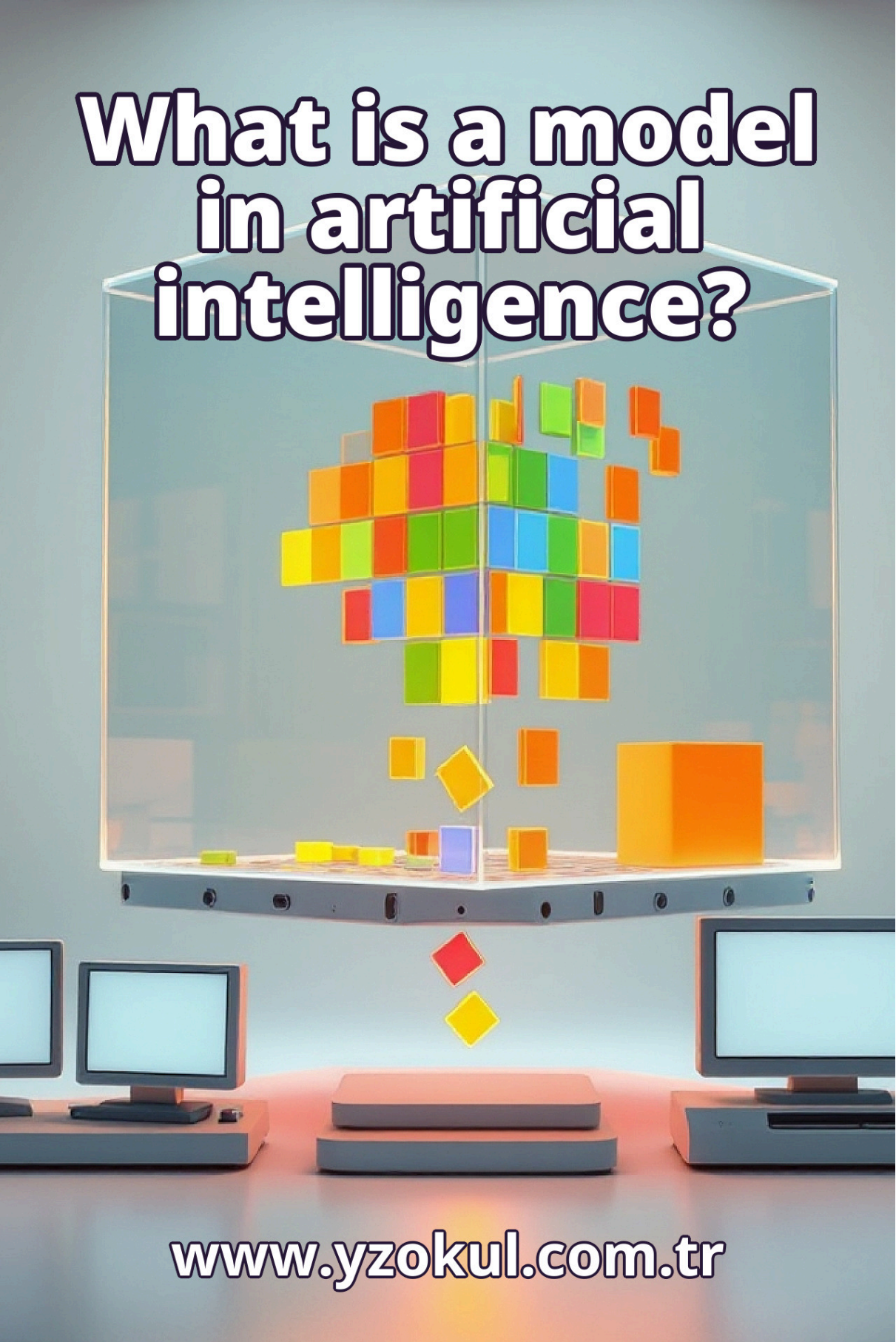
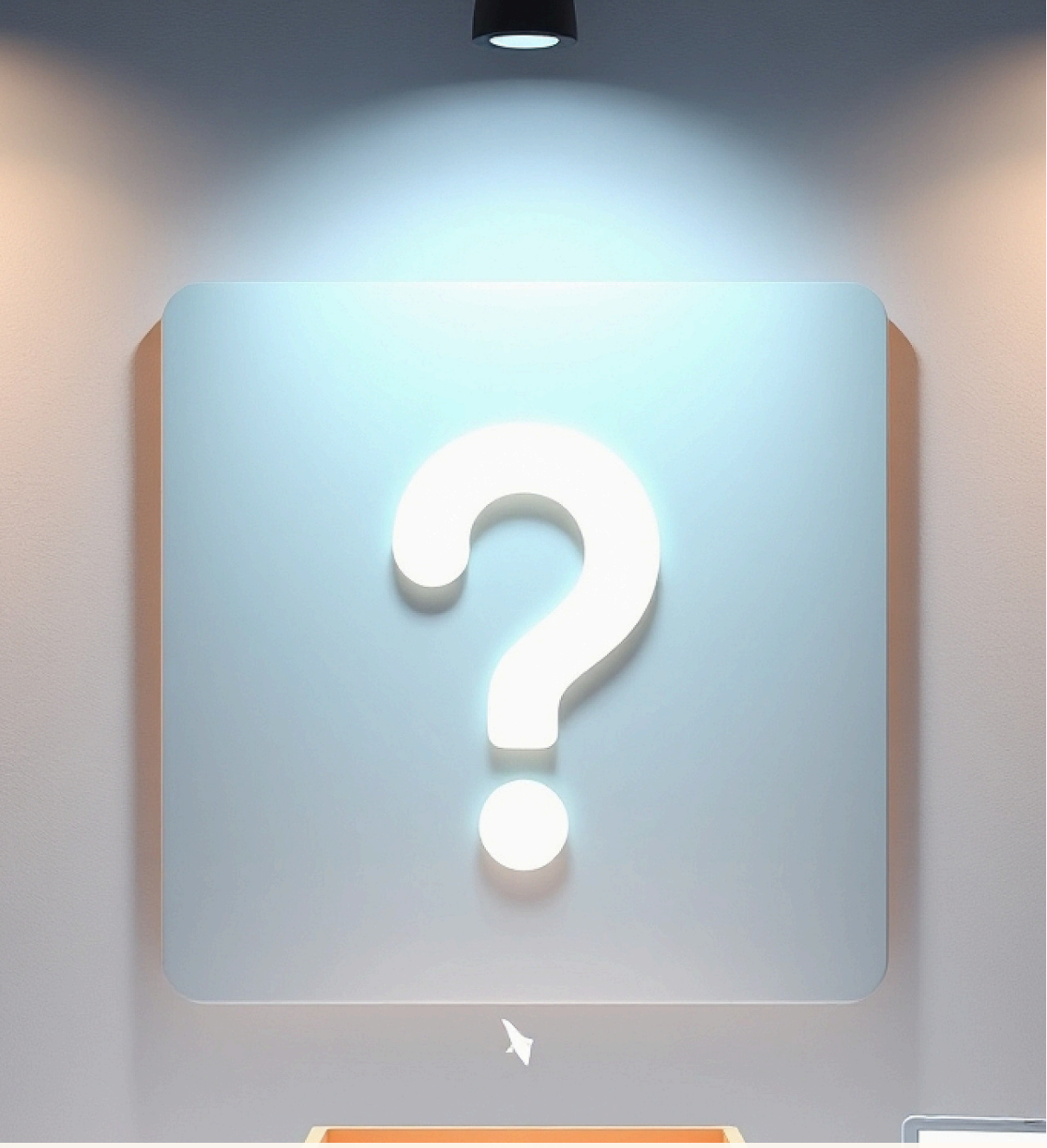
What is a model in artificial intelligence?

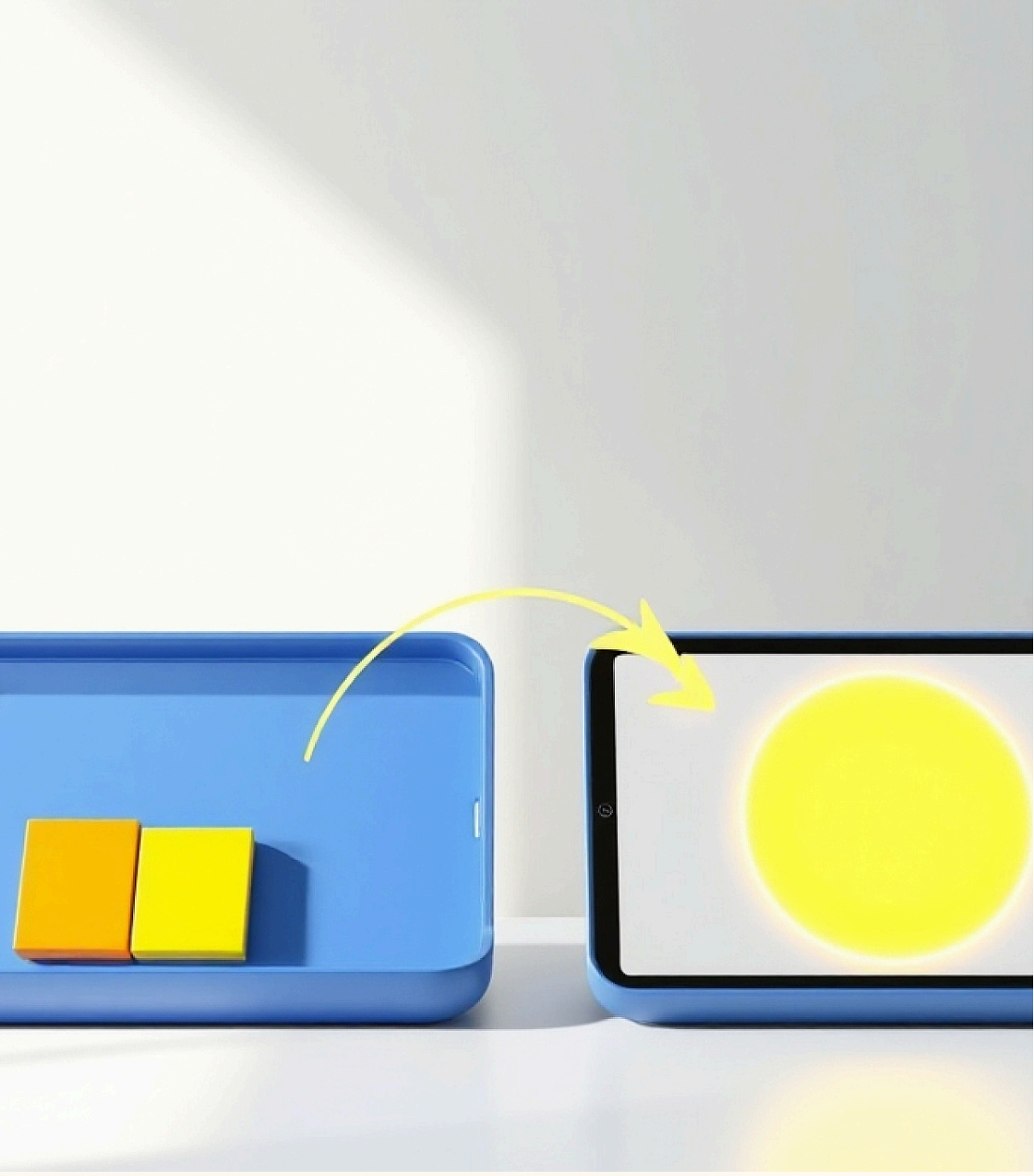


www.yzokul.com.tr

What is a model in artificial intelligence?



You have probably heard the term model in artificial intelligence and wondered what it actually means. The short answer is simple: a model is a mathematical structure that links inputs to outputs. In this book you will see exactly what that means, how a model is built and trained, and why the chat screen you use is not the model itself. Everything is explained from zero, so no prior knowledge is needed.



Look at a simple situation: something goes in, and something comes out. The thing you give is called the input. The thing you get back is called the output. For example, you give a calculator the input 2 + 2, and it returns the output 4. In artificial intelligence, inputs can be numbers, text, or images, and outputs can be a number, a label, or a piece of text.



A model is the thing in the middle that turns an input into an output. It is a mathematical structure, which means it is built from numbers and rules. You can picture it as a box: you put an input in one side, the model processes it, and an output comes out the other side. The model is not magic; it is a precise set of mathematical operations.



If you could open up a model, you would find many numerical settings inside it. These settings are sometimes called parameters or weights. They decide how each input is turned into an output. Each setting is just a number, and the model combines these numbers with the input using simple arithmetic. The exact way they are combined is fixed by the structure, but the numbers themselves can change.



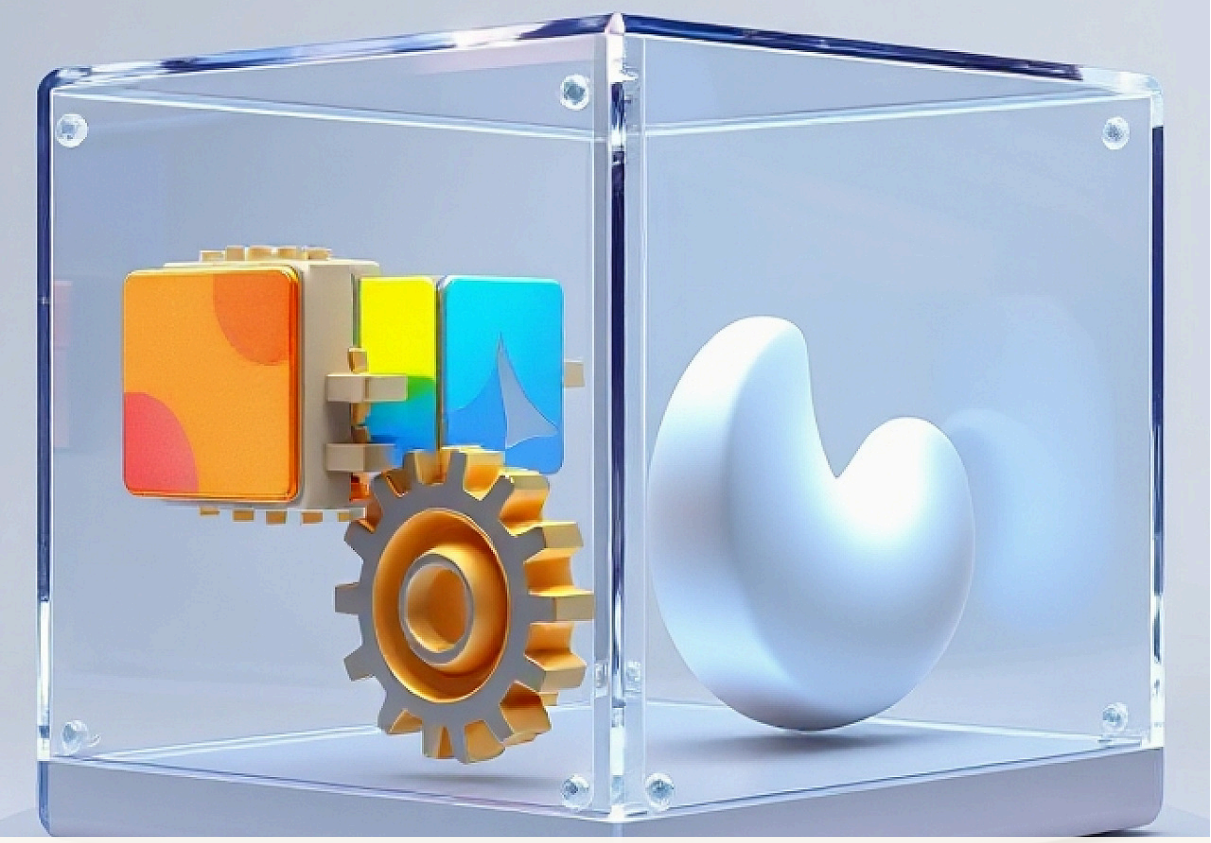
At the very beginning, before any learning happens, the settings inside the model are just rough guesses. Often they are small random numbers. Because these numbers are not tuned yet, the model's answers are poor. If you gave it an input, the output would likely be far from correct. This is normal: every model starts rough and improves through training.



To make the model better, you show it many examples of inputs together with their correct outputs. This collection of examples is called the training data. For instance, you might show it many houses with their sizes, their areas, and their actual prices. The model looks at these examples and uses them to adjust its internal numbers. The more good examples it sees, the better it can learn the pattern.



During training, the numerical settings are adjusted step by step. The model makes a prediction for an example, compares it with the correct output, and then slightly changes its numbers to reduce the error. This process repeats many times over the examples. Each adjustment makes the model's answers a little closer to the correct ones. Training stops when the answers are good enough.



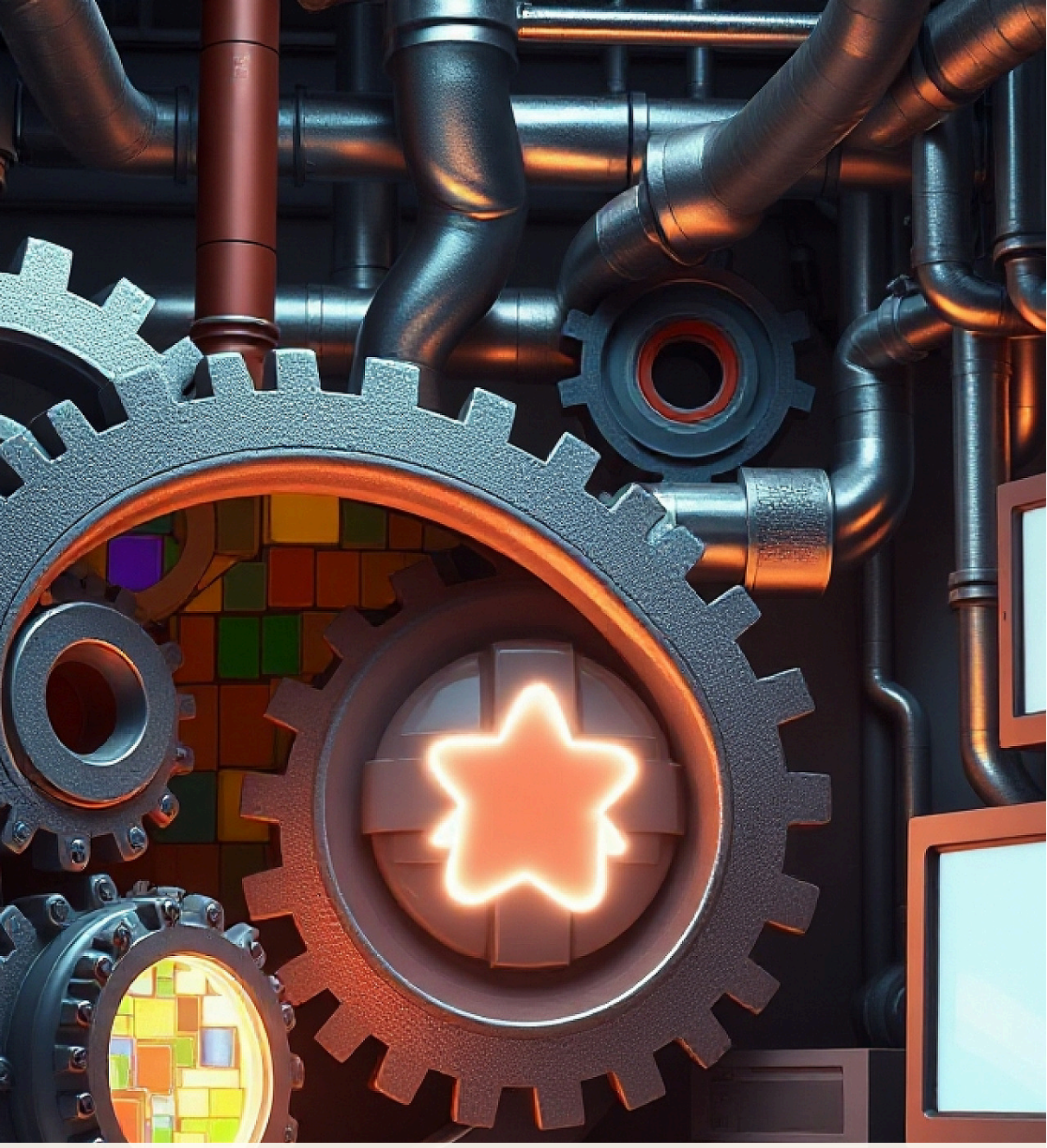
After training, the model uses its tuned numbers to produce answers for new inputs. It can now respond to cases it has not seen before, because its settings capture the general pattern from the examples. For example, a trained model can estimate the price of a house it has never encountered. The quality of its answers depends on how good the training examples were and how well the numbers were adjusted.



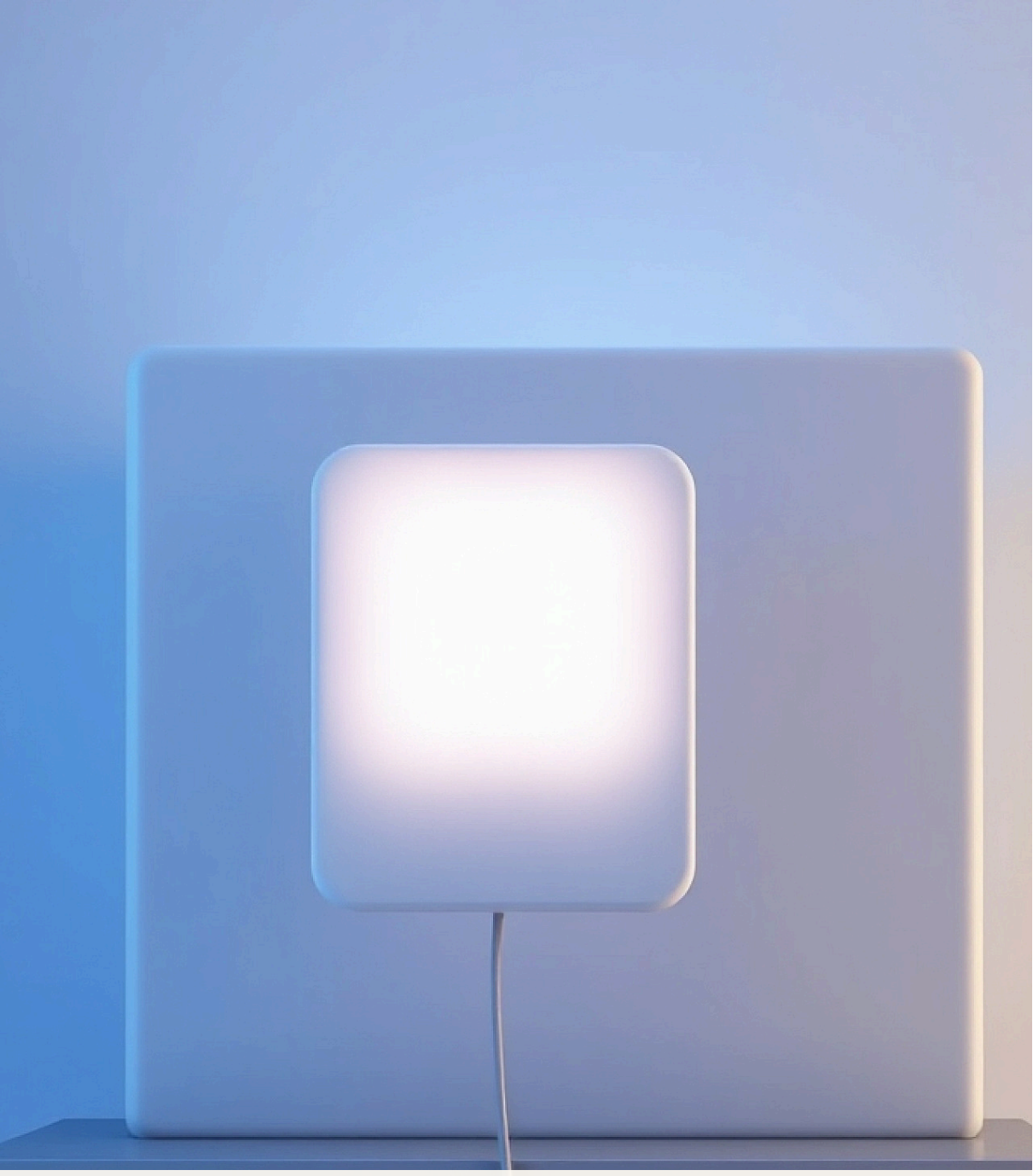
Take a model that receives the size of a house and the area it is located in, and produces an approximate price as its output. The input is two numbers: size and area code. The output is one number: the estimated price. During training, the model sees many houses with known prices and adjusts its internal numbers. After training, it can estimate prices for new houses.



Now give the trained house model a new house it has never seen. You provide its size and area, and the model still produces a price estimate. It can do this because its settings now capture the pattern that links size and area to price. The estimate may not be perfect, but it is based on the learned relationship, not on memorizing that exact house.



You can picture the model as a calculation mechanism adjusted from examples. It is not a list of stored answers. Instead, it is a set of tuned numbers that work together to compute an output from an input. When you give it a new input, it runs the calculation and returns a result. The intelligence comes from the tuning, not from storing every case.



Look at a chat screen where you type messages and read replies. This screen is only the interface of the application, the part you see and interact with. It is designed to be easy to use, with buttons and text boxes. But the interface itself does not think or compute answers. It is just the front end that collects your input and displays the output.



Behind the chat screen, the model is what produces the answers. The interface passes your input to the model, the model computes an output using its tuned numbers, and the interface shows you that output. So when you type a question, the screen sends it to the model, and the model generates a response. The model is the engine; the screen is the dashboard.



A common mistake is thinking the chat screen is the model. The screen is the front; the model is the calculation running behind it. Many different interfaces could use the same model, and one interface could use different models. The model is the mathematical structure with tuned numbers, not the visual layout you see. Keeping this distinction clear helps you understand where the intelligence actually lives.



Another mistake is thinking the model just stores answers in a database. In fact, it holds adjusted numbers that compute an output from an input. It does not look up a stored reply; it calculates one. This is why a model can respond to new inputs it has never seen. A database returns only what was put into it, but a model generalizes from its training.



Remember the key idea: a model is a mathematical structure linking inputs to outputs, tuned by training on examples. The numerical settings inside it are adjusted step by step to make its answers closer to correct. The chat screen you see is only the interface; the model works behind it, computing outputs from inputs. With this understanding, you can now interpret many discussions about artificial intelligence more clearly.