Understanding AI with Category Theory

Raoul Grouls

May 27, 2026

Contents

1 Introduction **2**

2 Lesson 1: Mathematical Basics **4**
2.1 Abstraction 4
2.2 Context 4
2.3 Defining things by their role 5
2.4 Diagrams 5
2.5 Posets and Tosets 7
2.6 High-dimensional spaces 8
2.7 Metrics 9
2.8 Monoids 10
2.9 Categories 10

1 Introduction

Since the breakthrough of Large Language Models (LLM) at the end of 2022 with chatGPT 3.5, the general public has shown increased interest in Artificial Intelligence (AI), up to the point that nowadays for a lot of people AI equals an LLM, or even a specific brand. From my perspective, this is like equating all transport to a specific brand of car. Not only are there many different brands and types of cars, but transport has been around for a long time and every situation requires its own type of transport.

In fact, the aspiration to let machines think is far older than modern computing and the Turing Test. Ideas about AI go back to the "analytical machine" conceptualized in 1837 by Charles Babbage and Ada Lovelace; the "calculus ratiocinator" by Leibniz in 1672 which was a universal logical calculus, a formal system in which disputes could be resolved by calculation, and the "Ars Magna" by Ramon Llull around 1305 in which he described a machine with rotating discs and combinatorial logic.

An important backbone of most modern AI is what is called neural AI or deep learning, which I would like to summarize as "a method to learn how to map information to a high dimensional vector space." This mathematical framing is precise and can be clarifying once you speak the language, but for most people it rads more like a magical spell than an explanation: even though the mathematics underlying this idea go back to the 1950s, for most people words like "high dimensional" and "vector space" invoke something that resembles a traumatic response, bringing them back to unpleasant memories of high school mathematics. This is unfortunate, because if AI is having an increasingly profound impact on our reality, a lot of people feel they are not equipped to understand how AI works and wouldn't even know where to start. The course you are following and to which this reader is a supplement intends to help you to have a better intuition of what "mapping to a high dimensional vector space" means.

A major motivation for me in giving this course is to help people understand what AI is through the lens of mathematics. I feel that mathematics essentially is the skill of consistent thinking, which is something I feel we need more in our society. I have found that a lot of people actually enjoy mathematics way more than they thought they would in the style they encounter in my courses. One of my favorite compliments is when people tell me "I wish you had been my math teacher when I was younger."

My approach to teaching people without a background in mathematics about AI is, perhaps surprisingly, Category Theory. Category Theory is a branch of mathematics that was invented to compare other branches of mathematics, and is therefore often seen as the endpoint of a study in mathematics, not the starting point. Eugenia Cheng introduced me to this approach of Category Theory, and I can highly recommend her work as an introduction into mathematical thinking. I find that making things more abstract often gives people a better grasp of what is going on. Sometimes making things more abstract also makes them simpler.

The course is organized into four lessons. The first sets up the mathematical groundwork: what abstraction is, when two things should count as "the same", and how we can impose structure on otherwise shapeless collections of things. The second turns to learning itself: what does it mean to learn something, how is learning similar or different between machines and humans, what does it mean to learn a model, can AI really be said to think? The third focuses at the territory between structure and chaos: what kinds of structure does

information need before it can be reasoned about, and how can AI help us bridge that gap. The fourth and final lesson tackles the hardest question — meaning: when we say an AI "understands" something, what could that even mean, can we as humans understand AI, and what does this mean for how we work with AI?

2 Lesson 1: Mathematical Basics

2.1 Abstraction

A nice definition of abstraction is stripping away irrelevant details. What is irrelevant is determined by the context and your goals. We are used to a lot of abstractions, so much that we sometimes don't even realize things are an abstraction. For example, all numbers are an abstraction. Look around, and spot three things (e.g. chairs) and three other things (e.g. books). These items probably don't have a lot in common; they have different functions, different sizes, different prices, different material etc. They do share their "threeness", which is a way of stripping away all irrelevant details.

Sometimes, people seem to think that abstractions are "objective". We often encounter this belief when people consider numeric models, and they might say things like "the data is objective" or "I prefer the objectivity of the model". However, there is a sense in which abstractions and models are highly subjective; someone made the choice of stripping away a lot of details, thereby considering all these details irrelevant. This choice might carry a lot of subjectivity. This often promotes behavior like "make the number go up": give people something to focus on (e.g. amount of billable hours) and they suddenly ignore a lot of relevant other items that don't fit in the model [1].

The relevant question then is not whether an abstraction is true, but whether it is useful, and useful always means: useful *for something*. I think Newton's laws are an excellent example of how models don't need to be true, only "true enough". If I throw a ball, we can use the laws of Newton to predict the trajectory. This model of reality ignores a lot of details, like shape, air resistance, etc. but in the end it works out fine for the goal of predicting where the ball will land. Since Einstein we know that Newton's laws are not absolutely true — they are, in fact, a perfect example of exactly what we just said: good enough approximations for certain purposes. It is more accurate to replace them with new laws that take into consideration how closely you approach speed of light and the curvature of spacetime. For our everyday use, this is irrelevant, but if you want to build a GPS device, you absolutely need to take into consideration that clocks on satellites run at a different speed due to the effects of gravity. According to Einstein's laws, even when you live or work on, let's say, the 10th floor, time runs faster there, but we don't really care because the effect is so small that we can safely ignore it.

In other words, abstractions are not true in any absolute sense; they are only true enough for certain purposes. Category Theory takes this idea seriously: rather than asking what things *are*, it asks what role things play and how they relate to one another. As we will see, this shift in perspective turns out to be surprisingly powerful.

2.2 Context

Context matters for models in terms of the things you want to use the model for. But there is another way in which context matters. For example, people often give "1+1=2" as a basic fact that is indisputable. But in the context of a binary number system there are only 0s and 1s, and 1+1=10. Even more fundamentally, philosophers like Frege write that "the

[1] This is an instance of Goodhart's Law, formulated by economist Charles Goodhart in 1975 and later popularised by Marilyn Strathern: "When a measure becomes a target, it ceases to be a good measure"

thought we express by the Pythagorean theorem is surely timeless, eternal, unchangable."[2] But it turns out that, at least on earth, the Pythagorean theorem is not absolutely true, just true enough. Let me explain: according to Einstein, gravity causes spacetime to curve. This means we don't live in a flat Euclidian spacetime, but in a slightly curved hyperbolic spacetime[3]. Pythagorean theorem assumes a flat Euclidian space. Therefore, one should use a hyperbolic variation of $a^2 + b^2 = c^2$ that incorporates the curvature of space as a parameter R:

$$\cosh\left(\frac{c}{R}\right) = \cosh\left(\frac{a}{R}\right) \cdot \cosh\left(\frac{b}{R}\right)$$

Again, this is not really relevant if you just want to engineer something, because the errors you would get with using the traditional version are very small, but they will be there and grandiose claims like Frege made don't hold up. Context, then, is not mere background detail: it is part of what determines meaning. The same statement can be true in one context and false in another, not because truth is arbitrary, but because the surrounding rules have changed. In mathematics, we make this precise through axioms: a theorem is not simply true, it is true *within a given set of rules*. Category Theory takes this insight as a design principle: every object and every arrow always lives inside a category, so the context is never assumed; it is always stated explicitly.

2.3 Defining things by their role

A key idea in Category Theory is that it is possible to define things completely by the role they play.

A nice example of this is the role of what we call *identity*. Given an object a, we say there is an identity morphism $1_a \colon a \to a$. A morphism is the category way of saying "there is a relationship between these two things" and it is denoted by an arrow. The identity morphism (or relationship) is basically the same as the statement "everything is the same as itself" and we really want this basic relationship in every category. It is possible that there is no identity, but this gives us a very fuzzy situation because if we can't even know for sure that something is the same as itself, we loose a fundament for being consistent.

We can distinguish the identity *morphism* from the identity *element*. The morphism $a \to a$ is a structural concept: it describes a role, namely the role of "doing nothing" or of "a thing being the same as itself". When we ask which concrete element fills that role, the answer depends on the structure. For example, given numbers and addition, the identity element is zero: $a + 0 = a = 0 + a$. For multiplication, the identity element is 1: $a \times 1 = a = 1 \times a$. In both cases this is an identity morphism, but the element that plays the role differs because the underlying operation — addition versus multiplication — is different.

It now is the case that, for addition, what 0 is, is uniquely defined by the role it plays as identity. We actually don't need any more information because it has a unique role and that is all there is to know about 0 in the context of addition.

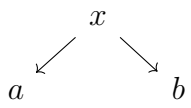
2.4 Diagrams

We can find patterns and structure in many things, for example in arguments.

[2]Frege, G. (1956). The thought: A logical inquiry. Mind, 65(259), 289-311.

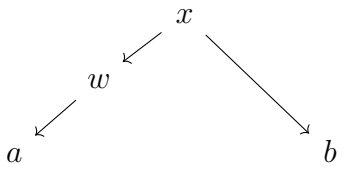
[3]If you are intrigued, this https://youtu.be/n7GYYerlQWs is a great 9-minute explanation.

A typical argument would be the statement "I think a and b are examples of x". This structure could really be anything, for example "I think bananas and oranges are examples of fruit". In this situation every arrow $a \xrightarrow{f} b$ would mean something like "b is an example of a."



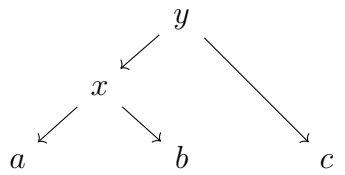
It would then be possible to object to this argument in different ways. One would would be to reply with "I think there is a more specific example w that explains a." In a diagram, this would look something like this:

Let's make this concrete. For example, someone could say "I think the coronavirus is just a flu". This argument follows the pattern "both coronavirus (a) and the seasonal flu (b) are respiratory viruses with a mortality rate somewhere between 1-5 percent (x)". Someone else could argue "Yes, they have common characteristics x, however there is a category w "viruses that are new to the immune system of most people and therefore highly contagious" that only applies to a and therefore they are different.. The introduction of this new level w also explains why the current "seasonal corona" are no longer causing the ICUs to be overloaded.



Another way to object to arguments that are analogies would be to argue that the argument is invoking a more general principle y, and the objection is against an example c that they think is included. This structure would look like this:

This next example is politically charged, but we should not shy away from charged arguments. In my opinion, it only makes it more important to understand the structure of the argument.



Currently, there are protests in the Netherlands against immigration centers. During these protests, it is common for people to somehow equate a "freeing up limited resources in society for refugees" and something like b "being under existential threat as a society". I would argue that these two are not equal. The principle x would be something like "as a society, let us have compassion for people that are worse off, and free up some resources" and my objection would be that what they label as b should actually be positioned at c. I'm not sure if protesters are actually trying to have a rational argument about this, taking into consideration that increasing polarity in our society is an active goal of information warfare[4]. But if someone would want to have a rational discussion about this, I think the issue is that the protesters seem to focus on a broader principle y that would need to sound a bit like "let's spend unlimited resources on everybody (except native Dutch people) that is interested, regardless of their motivation or criminal background". A lot of polarized discussions in our society follow this structure, where one group is defending a based on principle x, while another group is attacking this as if people are defending principle y (which they don't) and act like a equates c.

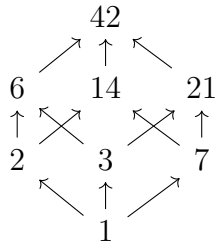
[4]This is quite a broad subject; I recorded these clips `https://www.youtube.com/playlist?list=PLDDloDrHSwSum-0wTG58DKqtd2G76eDlp` to discuss the topic of information warfare in more depth.

2.5 Posets and Tosets

Now that we have arrows and the idea of relationships between things, we can look at some particularly tidy ways for those relationships to behave. A natural place to begin is with the idea of *order*: situations in which we want to say that one thing comes before, below, or is in some sense less than another.

Some orderings are very strict: every pair of objects can be compared. Think of dates on a calendar, or the natural numbers under $\leq$. For any two dates, one comes before the other (or they are the same day) — there is no ambiguity. We call this kind of structure a *totally ordered set*, or *toset* for short. In our arrow language, a toset is a situation in which between any two objects a and b there is *exactly one* arrow: either $a \to b$ or $b \to a$, and if $a = b$ the arrow is the identity.

Other orderings are less strict. Consider the relationship "is a divisor of". We have 2 divides 6 and 3 divides 6, but 2 and 3 are not divisors of each other: they are simply not comparable in this relationship. The factors of 42 form a nice example, shown on the right.[5] We see that the picture is no longer a straight line; some pairs sit at the same level but have no arrow between them. A structure like this is called a *partially ordered set*, or *poset*: there is *at most one* arrow between any two objects, but not necessarily one.



Tosets are a special case of posets. Every toset is a poset that happens to have the extra property that every pair is comparable. We get from one to the other by relaxing a single stipulation, going from "exactly one" to "at most one" — a pattern we will see again and again. Relaxing a definition often gives us a more general object that contains the original as a special case, and lets us include more interesting examples.

A concrete example of this would be relationships. I think relationships are a poset, even though we are often tempted to pretend they are a toset. Especially in love relationships, we tend to look at several different dimensions at once: e.g. emotional capability, intelligence, sensuality, and so on. If you think three axes is not enough, just add more dimensions, but for now, let's keep it simple and stick to three. Even with just these three it is already hard to compare all people you would like to date. Yes, the smarter person is interesting, but how does this really compare to the emotionally more available, but less smart person? Of course, you can impose a total order, but this new order does not automatically follow from the ordering of each axis itself. It would be a new structure imposed upon the combination of axes.

We could now view monogamy as the philosophy that relationships and love are a totally ordered set: every candidate can be compared to every other candidate, and it is possible to pick the "best" overall person. Polyamory, on the other hand, could be viewed as the philosophy that love is a poset, with multiple incomparable axes. There is no principled way to flatten all dimensions into a single ordered line, and thus it doesn't make sense to say who you love "the most". My point here is not to claim one of these to be a better

[5]The choice of 42 is a nod to Douglas Adams' *The Hitchhiker's Guide to the Galaxy*, in which a supercomputer named Deep Thought, after 7.5 million years of computation, reveals that 42 is "the Answer to the Ultimate Question of Life, the Universe, and Everything". Sadly, nobody had thought to write down the Ultimate Question itself.

model but that we impose a mental structure on reality, often inherited from society, and that picking a structure has an impact on how we act and feel. If the relationship example doesn't convince you, let me ask you another question: which of your children (if you have them) do you love most? Almost everyone agrees that the answer to this question is some variation of the idea that "love is a poset, not a toset".

Even with a poset it is sometimes possible to find a single object that sits above everything else. This is called a *terminal object*, which would be the number 42 in the lattice of factors. In the example of dating, if you find a 42, this probably means you should try to marry them.

2.6 High-dimensional spaces

The lattice for the factors of 42 is already a small hint of something more general. We laid out the picture along three distinct axes — one for each prime — and the combinations of those primes filled in the middle. Even with three axes the diagram is already a bit crowded, and the moment we add a fourth or fifth we lose any hope of drawing it on a flat page. This is not a quirk of the example; it is the rule rather than the exception. Most interesting spaces have many more than three axes.

In modern AI we routinely work with vector spaces of hundreds or thousands of dimensions. Large language models, for example, typically embed words into spaces with 768 or 1024 dimensions. There is good reason to think these numbers are larger than what is strictly required. Researchers have tried to estimate the *intrinsic* dimensionality of language: the dimension of the underlying manifold[6] on which the actual meaning lives, as opposed to the much larger ambient space we choose to embed it in. The numbers that come out are surprisingly small. A recent study measured the intrinsic dimension of token embeddings across a range of language models and found values clustering around 40 for typical small and mid-size models, even though the ambient embedding dimension was many times larger.[7]

To get a feel for how powerful even those much smaller numbers already are, suppose we allow just ten distinct options along each axis. A three-dimensional space would be able to contain 1000 elements, or 10^3, and a 40-dimensional space thus contains a minimum of 10^{40} distinguishable points. For comparison, the observable universe is estimated to contain around 10^{80} atoms, and a 80-dimensional space with ten options per axis already has enough distinct points to give every atom in it its own private address. We do not need something exotic to capture an enormous amount of variation; we just need enough independent axes.

But "independent" is doing real work in that sentence. In a flat geometric setting we say two axes are *orthogonal* when they meet at a 90 degree angle. When organizing data in a high dimensional space we want the same property, but not just in a purely geometric sense; we want it in a *semantic* sense. Two axes are semantically orthogonal when the value along one does not predict the value along the other — when they capture

[6] A manifold is a shape that lives inside a higher-dimensional space – like a sheet of paper, which has only two dimensions but can lie within the three-dimensional space around it

[7] Kataiwa, Hakaze and Ohki, *Measuring Intrinsic Dimension of Token Embeddings*, arXiv:2503.02142 (2025). The reported intrinsic dimensions for word embeddings such as GloVe and Word2Vec come in around 25, and for the Pythia model family they range from roughly 25 to 40 for models up to a few billion parameters, drifting higher for the very largest.

genuinely separate aspects of whatever the data describes. If two axes turn out to be highly correlated (height and weight, say, or being extroverted and laughing out loud), they are not really two independent axes adding much new information; they collapse into roughly one, and the space has fewer effective dimensions than we thought. A great deal of the work in building a useful AI representation — in any field, really — is finding axes that are orthogonal in this semantic sense.

A single point in a high dimensional space, equipped with the ability to be moved around, already forms a small structure all by itself: it is, in a precise sense, a *monoid*, a kind of object we will meet in a couple of sections. For now the key idea to take away is that high dimensional spaces are not just a technical curiosity; they are how AI represents the world, and the structure of those spaces is what determines what an AI "sees".

2.7 Metrics

We have talked about *order*: which things come before others. We can also ask about *distance*: how far apart things are. The familiar way to measure distance is "as the crow flies" — a straight line. But, just as Newton's laws are only true in a specific context, the straight-line distance is only one of many possible ways of measuring how far apart two things are, and what counts as the "right" distance depends on the situation.

Imagine you live in a city laid out on a perfect grid, like a stylized Manhattan. To get from intersection A to intersection B, you can't cut diagonally through the buildings; you have to follow the streets. If B is 3 blocks east and 4 blocks south of A, the crow flies 5 blocks, but you walk 7. This is called the *taxi-cab metric* or the *Manhattan metric*, and it is just as valid a way of measuring distance as the straight line, it just answers a different question. With this metric the analogous shape of "a circle" (all points the same distance from a center) would be a diamond, not a round shape, because that's where all the points at distance r end up sitting on the grid.

Once we accept that there are different ways of measuring distance, the mathematician's reflex is to ask: what should count as a "reasonable" notion of distance? The answer takes the form of a small list of criteria. A function d that takes two points A and B and returns a number is called a *metric* if it satisfies:

- $d(A, B) \geq 0$: distance is never negative.
- $d(A, B) = 0$ if and only if $A = B$: the only thing at distance zero from a point is the point itself.
- $d(A, B) = d(B, A)$: distance is symmetric.
- $d(A, B) \leq d(A, X) + d(X, B)$ for any third point X: taking a detour through X cannot shorten the trip. This is the *triangle inequality*.

The triangle inequality captures the idea that "the shortest distance between two points is a direct route", but in a form that applies to any metric, not just the Euclidean one.

This level of abstraction is exactly what makes the concept useful, and it brings us back to the AI motivation from the introduction. Modern AI is largely about mapping things (words, images, behaviors, documents) into a high dimensional vector space (for our purposes, think of this as a space with multiple axes). Once you are in a vector space, you can ask "how close is this thing to that thing?", and the answer is given by a metric.

Different choices of metric yield different geometries and therefore different answers to what feels like the same question. A search engine, a recommendation system, or a fraud detector using one metric will give different results from one using another, even on identical data.

2.8 Monoids

Tosets and posets put restrictions on what kind of *arrows* can sit between objects. Now we will consider what happens if we restrict the *objects* that can sit in the category at all.

Definition 2.1. A *monoid* is a category with exactly one object.

That is the entire definition, and it feels both abstract and a bit empty: "a category with one object" — so what? The strangeness sharpens when we look at the canonical example, the natural numbers $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ under addition. We are used to thinking of these as a set: an infinite collection of distinct things. How is that supposed to fit into a category with a single object?

The answer is a shift in perspective that takes some getting used to. In the categorical view the numbers are no longer the objects — they have become the *arrows*. We take a single dummy object, we can call it whatever we like, e.g. $*$, and every natural number becomes an arrow from that object back to itself: 0 is one arrow $* \xrightarrow{0} *$, 1 is another, 2 is another, and so on. Composing two of these arrows is just addition: the arrow 2 composed with the arrow 3 gives the arrow 5. The infinite collection of numbers has not gone anywhere; it has only changed dimension.

A monoid is what you get when you take a category, throw away all but one object, and reinterpret the arrows on that object as "elements".

There are many other examples once you start to look. Strings of letters under concatenation form a monoid: we simple have $* \xrightarrow{a} *$, $* \xrightarrow{b} *$, etc. Lists under appending form a monoid. The natural numbers under multiplication form a different monoid, with identity 1 instead of 0.

It is possible to add to the monoid the extra property that every arrow has an *inverse* — an arrow which, composed with it, gives back the identity. The natural numbers under addition do not have this (there is no natural number you can add to 3 to get 0). The integers do: $3 - 3 = 0$ so -3 is the inverse of 3 (note how we need the identity to "organize" and define our extra property). A monoid in which every arrow has an inverse is called a *group*, and groups capture the idea of symmetry: the rotations of a square, the permutations of a deck of cards, a Rubik's Cube. The moment you can "undo" every operation, the monoid becomes a group, which is a new type of mathematical structure.

2.9 Categories

So far we have seen tosets, posets, high dimensional vector spaces, metric spaces, monoids and groups. Each is a different way of equipping a bare "collection of objects" with extra structure: order, distance, combination, symmetry. They look quite different on the surface, but Category Theory is a language that can describe them all. Let's finally come to the definition of what a category is.

Definition 2.2. A *category* $\mathcal{C}$ consists of:

Data.

- A collection of *objects*.
- For each pair of objects a, b, a collection of *arrows* (or *morphisms*) $a \to b$.

Structure.

- Identity: For each object a, a distinguished *identity arrow* $1_a \colon a \to a$.
- Composition: for arrows $f \colon a \to b$ and $g \colon b \to c$, a *composite* arrow $g \circ f \colon a \to c$.

Properties.

- Unit laws: for any arrow $f \colon a \to b$, we have $f \circ 1_a = f = 1_b \circ f$.
- Associativity: for any composable arrows f, g, h we have $(h \circ g) \circ f = h \circ (g \circ f)$.

That is the whole definition. It is short, but each of the structures from the previous sections is hiding inside it as a special case:

- A *poset* is a category in which there is at most one arrow between any two objects.
- A *toset* is a category in which there is exactly one arrow between any two objects.
- A *monoid* is a category with exactly one object.
- A *group* is a monoid in which every arrow has an inverse.

This is one of the things that makes Category Theory powerful. By identifying the small number of patterns that lots of structures have in common — things relate to each other, relations can be composed in a sensible way, and there is always some notion of identity — it gives us a single language in which to talk about all of them. This is also what makes Category Theory an ideal language to describe structures over multiple disciplines, bridging mathematics, philosophy, physics, etc.

In the lessons that follow we will use this framework to ask more complex questions. What does it actually mean to *learn* something, and how is the way an AI learns the same as, or different from, the way you and I do? Once we have a handle on that, we can ask what kind of structures we can impose on unstructured data, and how AI can help us bridge that gap. And finally we can ask about *meaning*: when we say an AI "understands" something, what would that even mean? And can we say that AI "thinks" or "understands"?