Artificial Intelligence for Sciences and Humanities

© Copyright 2026

Section 2: How AI Sees Data

<u>DO NOT POLLUTE!</u> AVOID PRINTING, OR PRINT 2-SIDED MULTIPAGE.

2.1 Data as Numbers

As discussed earlier, data plays a central role in artificial intelligence (AI). This section focuses on how AI *sees* data, or more precisely, how AI *represents* data in order to process it and learn from it.

Unlike humans, AI systems do not perceive meaning, intention, or context directly. They do not see an image as an image, read a poem as poetry, or hear music as sound. Instead, AI systems operate on numerical representations. Every form of data, no matter how rich or expressive to humans, must first be translated into numbers before an AI system can work with it. This translation is known as a *data representation.*

At the most basic level, AI represents data as long lists or arrays of numbers. These numbers encode measurable properties of the data, allowing mathematical operations to be performed on them. Learning, in this context, means discovering patterns, regularities, and relationships among these numbers across many examples.

2.2 How Different Kinds of Data Become Numbers

Images

To a human, an image may depict a face, a landscape, or a work of art. To an AI system, however, an image begins as a grid of pixels, where each pixel is described by numerical values corresponding to brightness and color.

One of the most common image representations is the RGB color system, where each pixel is described by three numbers corresponding to the intensity of red, green, and blue light at that location. Most colors visible to the human eye can be represented by combining different proportions of these three basic colors. Standard image processing convention encodes each intensity level with a value from 0 to 255. This number may appear arbitrary, but it arises from a technical reason: computers store information using a binary system (based on 0's and 1's), and using 8 binary digits (called bits) allows for exactly 256 possible values. This range provides enough detail to represent most color variations that the human eye can distinguish, while remaining efficient for digital storage and processing. This way, for example:

- A black pixel is represented as $[0, 0, 0]$,
- A red pixel as $[255, 0, 0]$,
- A green pixel as $[0, 255, 0]$,

- A blue pixel as $[0, 0, 255]$,
- A white pixel as $[255, 255, 255]$, and
- A purple pixel as $[128, 0, 128]$.

Therefore, to an AI system, a photograph is not a picture, but a large table of numbers, like a spreadsheet. AI systems then learn to transform these raw numbers into progressively more abstract representations, such as edges, shapes, textures, and recurring visual patterns, without ever *seeing* the image in a human sense.

Text

Words, sentences, and entire documents must be converted into numerical representations before AI can analyze them. Typically, text is broken into basic *units* (such as words or word fragments). For example, one of the simplest forms to represent text is through *one-hot* enconding. The main idea is to represent each word (unit) with a very long *vector* (list of numbers) in which all entries are zero except for a single position corresponding to that word, which is set to one. In other words, each word is marked by a single *on* value among many *off* values. Hence the term one-hot.

For an over-simplified example, imagine we have a text database that contains only five words: *apple*, *climbs*, *cat*, *dog*, and *tree*. We assign each word a fixed position in a vector of length five:

$$\textit{apple} \mapsto \begin{bmatrix}1\\0\\0\\0\\0\end{bmatrix}, \qquad \textit{climbs} \mapsto \begin{bmatrix}0\\1\\0\\0\\0\end{bmatrix}, \qquad \textit{cat} \mapsto \begin{bmatrix}0\\0\\1\\0\\0\end{bmatrix}, \qquad \textit{dog} \mapsto \begin{bmatrix}0\\0\\0\\1\\0\end{bmatrix}, \qquad \textit{tree} \mapsto \begin{bmatrix}0\\0\\0\\0\\1\end{bmatrix}.$$

This way, the sentence *cat climbs apple tree* can be represented as the *matrix* (table of numbers, like a spreadsheet) containing the concatenation of the corresponding vectors:

$$\textit{cat climbs apple tree} \mapsto \begin{bmatrix}0&0&1&0\\0&1&0&0\\1&0&0&0\\0&0&0&0\\0&0&0&1\end{bmatrix}.$$

Notice that these representations do not encode dictionary definitions, semantic meaning, or cultural context. They are not concepts; they are simply numerical markers that allow a computer to distinguish one word from another. AI language models learn patterns by observing how these numerical representations tend to appear together, recur, or follow one another across vast collections of text. In this way, relationships among words are inferred statistically from usage, rather than interpreted symbolically or semantically in a human sense.

Sound

Sound can be seen as a signal that changes over time. Hence, it can be represented with a sequence of numbers that capture physical properties of the signal, such as amplitude (loudness) and frequency (pitch).

From these signals, AI systems can learn recurring musical patterns, identify stylistic similarities, or classify genres.

Traits

Continuous traits, such as a person's height or weight can be recorded directly as numbers. Categorical traits like hair color or sex may be encoded using simple numerical indicators. For example, hair color can be encoded using numerical categories: black hair might be represented by 1, brown by 2, blonde by 3, and so on. It is important to note that these numbers do not imply any ordering, ranking, or degree of similarity. The choice of numbers is arbitrary and purely symbolic. They simply allow the AI system to distinguish between categories. Treating these numbers as quantities would be misleading, since brown is not twice black, nor is blonde greater in any meaningful sense.

Multimodal

Many modern AI systems work with multiple types of data at once, such as images and text together. In these cases, the system learns numerical representations that allow different modalities to be linked, for example, associating a visual image with a textual description or historical category. Another perfect example of multimodal data are **electronic health records**, which contain a rich mixture of information: laboratory test results, medication histories, imaging reports, timelines of patient visits, clinical notes written by physicians, and diagnostic codes. Structured components of EHRs, such as laboratory values or vital signs, are relatively straightforward to represent numerically. A blood pressure reading, a cholesterol level, or a heart rate is already a number and can be stored directly. Categorical information, such as diagnoses or procedures, is often converted into numerical codes. For example, a diagnosis like *type 2 diabetes* may be represented as a specific code. Medications may be encoded in a similar way, with each drug represented as an on/off indicator or a numerical dosage value. Time also becomes numerical. The sequence of patient visits, treatments, and outcomes is often represented as ordered numerical events along a timeline. Unstructured components such as physicians' notes or imagery can be transformed to numbers as discussed above. From the AI's perspective, an electronic health record is therefore not a patient history, but a structured sequence of numbers capturing measurements, categories, word patterns, and temporal relationships. AI systems look for patterns in how these sequences evolve across large populations of patients, such as which combinations of measurements tend to precede certain outcomes.

2.3 What Is Lost When Data Becomes Numbers?

The translation of data into numerical form is powerful, but it is also reductive. When data is converted into numbers, certain aspects are emphasized, while others are inevitably diminished or lost.

For example, context can be one such casualty. Historical background, cultural significance, irony, ambiguity, and subtext are difficult to encode numerically. An AI system may detect that two texts are stylistically similar without understanding that one is parodying the other. Likewise, an image may be grouped with others based on visual similarity while ignoring its symbolic or political meaning.

In other words, meaning and content may be lost in the numerical translation of data. On their own, AI systems may only identify patterns in data that are statistically related without ever *knowing* what those patterns represent. For example, an AI image system may learn to identify recurring edges, shapes, color contrasts, or textures across many images without ever knowing that those images contain dogs.

Meaning enters the system only when **labels** are introduced, linking these learned visual patterns to human-defined concepts. Until then, the AI's perception remains detached from human interpretation: it recognizes structure without recognizing significance.

2.4 Labels

Labels are human-provided annotations that assign meaning, categories, or values to data. A label tells an AI system what a piece of data is supposed to represent. For example:

- An image may be labeled *dog*, or *portrait* or *landscape.*
- A musical excerpt may be labeled by genre, composer, or mood.
- A literary passage may be labeled by theme, sentiment, or historical period.
- A medical record may be labeled with a diagnosis or outcome.

Just like images, text, or sound, **labels themselves must also be encoded as numbers** in order for AI systems to use them. For example, consider an image of a dog. The image itself may be represented internally as millions of numbers corresponding its pixel values. By contrast, the label *dog* is typically encoded as a much simpler numerical object, such as a single number or a short numerical vector indicating membership in a category (like a one-hot vector).

While both, the data itself and the label are numerical, they play very different roles in learning. Generally, the data represents the *input*, that is, the object being analyzed, such as an image, a text, or a sound; what the AI *observes.* Labels represent the *output*, that is, the human-provided answer that the AI system is expected to learn to *respond.*

2.5 Where do Labels Come From?

Labels are not inherent in data; they are assigned. There are several common ways in which labels are obtained.

Expert annotation

In some domains, labels are provided by trained experts. Art historians may label artworks by style or period; clinicians may label medical data with diagnoses; literary scholars may annotate texts by theme or genre. These labels are often high quality but expensive and time-consuming to produce.

Crowdsourcing

Many large datasets rely on annotations from non-experts through online platforms. Contributors may be asked to identify objects in images, judge sentiment in text, or classify sounds. While this approach allows labels to be collected at scale, it can introduce inconsistency and cultural bias.

Institutional or administrative labels

In archives, libraries, and medical systems, labels often come from existing classification systems: catalog records, diagnostic codes, metadata fields, or standardized taxonomies. These labels were usually created for administrative or organizational purposes, not for AI, yet they are frequently repurposed for machine learning.

Automatically generated or inferred labels

In some cases, labels are derived indirectly. For example, using user behavior (clicks, ratings, viewing time) as a proxy for interest or preference. These labels reflect behavior rather than explicit interpretation and can be noisy or misleading.

2.6 Top to Bottom Example

Imagine we want to teach an AI system to distinguish edible from poisonous mushrooms based on how they look. The ultimate goal is to embed this system in a phone app that people can use in the wild to help identify mushrooms safely.

Step 1: Choosing the Data

Since the task is visual, our data consists of images of mushrooms. Each image shows a mushroom from a particular angle, under particular lighting conditions, possibly against a natural background. To humans, these images convey shape, color, texture, and visual cues associated with different species. To an AI system, however, they are just raw visual inputs that must first be translated into numbers.

Step 2: Turning Images into Numbers

The first step in any AI system is numerical representation. Following the standard convention described above, we will use the RGB system to represent each pixel with three numerical values, each ranging from 0 to 255, corresponding to its intensity of red, green, and blue. An entire image, which may contain millions of pixels, is therefore represented as a very large collection of numbers. From the AI's perspective, the image is no longer a mushroom; it is simply a structured array of numbers.

Step 3: Encoding the Labels

Since we want the AI system to distinguish between a specific characteristic (edible or poisonous), we must explicitly indicate the system whether each image corresponds to an edible or poisonous mushroom, so that it can later identify the patterns distinguishing each group.

Therefore, each image in our dataset must be paired with a label provided by humans, typically experts such as mycologists or reliable field guides. For example:

- Any edible mushroom image might be labeled with the number 1,
- Any poisonous mushroom image might be labeled with the number 0.

Recall that just like the images themselves, labels are also represented as numbers. However, they play a different role: the image numbers describe what the mushroom looks like, while the label number encodes the trait that we are interested in.

In some cases, particular AI systems benefit from a carefully chosen way of encoding labels. For example, representing edible mushrooms with the number 1 and poisonous mushrooms with the number 0 works especially well for certain models, such as logistic regression. However, in many situations this numerical choice is largely a matter of convention rather than necessity. For many AI systems, we could just as well label poisonous mushrooms with the number -1 instead of 0, with no meaningful change in the final outcome. What matters is not the specific numbers themselves, but the consistent distinction they represent.

Step 4: Learning the Relationship Between Images and Labels

The next step consists on giving the AI system many examples of image-label pairs (in their corresponding numerical representations). We will see in later sections how exactly this is done. The system's task is to learn patterns in the image data that tend to be associated with each label.

For instance, the AI system may learn that certain combinations of colors, shapes, textures, or surface patterns appear more often in images labeled *poisonous* while others appear more often in images labeled *edible*. Importantly, the system is not told which features to look for; it discovers them by analyzing large numbers of examples, and such features may have no apparent interpretation to humans.

Step 5: Making Predictions on New Images

Once trained, the AI system can be shown a new image of a mushroom it has never seen before. This new image is converted into numbers in exactly the same way as before. The system then compares its numerical patterns to those it learned during training and produces a prediction, delivered in the same label representation, which can then be translated back into a human-readable result. In our example, a 1 would be translated to *likely edible* and a 0 to *likely poisonous*.

Edible or poisonous?

Step 6: Human Use and Responsibility

From the user's perspective, the app appears simple: take a photo, receive an answer. But behind the scenes, the system has never seen a mushroom, understood danger, or recognized edibility. It has only processed numbers and learned statistical associations between image patterns and human-provided labels.

It is important to recognize that an AI system's outputs depend entirely on the data it is trained on and on how carefully its labels are obtained. The system does not possess independent knowledge or judgment; it can only learn from the examples it is given. As a result, errors, biases, or inconsistencies in the training data are inevitably reflected in the system's behavior.

This dependence on labels places AI squarely within a long historical tradition of classification as an exercise of power. Throughout history, systems of classification have shaped how people, objects, and practices are understood and governed. Medical classifications defined what counted as illness or normality. Archival and bibliographic systems determined whose voices were preserved, categorized, or erased.

AI labeling inherits this legacy. When an image is labeled edible or poisonous, normal or abnormal, authentic or fake, these labels do not merely describe reality. They actively structure it. They encode assumptions, institutional priorities, and cultural norms, often presented in a neutral or technical guise. Once converted into numbers, these labels may appear objective, even though they originate in historically contingent human judgments.

This dependence raises significant ethical questions, especially in high-stakes settings such as food safety, medical diagnosis, legal decision-making, or public policy, where misclassifications can cause real harm. A mislabeled example (whether due to lack of expertise, systemic bias, or malicious intent) may seem insignificant, but it can propagate through an AI system at scale, leading the system to learn incorrect or even dangerous patterns. This echoes historical cases in which flawed or biased classification systems were institutionalized, normalized, and difficult to challenge.

These concerns naturally lead to broader questions of responsibility and governance. Should the collection and labeling of training data be regulated in critical applications? What safeguards can be put in place to detect labeling errors, disagreement among annotators, or malicious manipulation of data? Who has the authority to define categories?

Equally important is the question of accountability. When an AI system makes a harmful mistake, responsibility does not lie with the algorithm itself, but with the human institutions that designed the system, selected the data, defined the labels, and chose how and where the system would be deployed. Understanding how labels are created and how fragile they can be is therefore not only a technical concern, but a social, ethical, and legal one.

Recognizing these issues encourages a more cautious and critical approach to AI. Rather than treating AI outputs as objective or authoritative, especially in high-stakes contexts, they should be understood as the result of human decisions embedded in data, labels, and design choices.

2.7 Notation

As you can imagine, AI systems quickly end up handling vast amounts of numbers. For instance, a single 48 megapixel RGB image is represented using 144 million numbers. Modern systems like DALL-E use hundreds of millions of images. That is over 10 quadrillion numbers. To get an idea, that is 16 zeros!

For these numbers to remain meaningful, they must be kept carefully organized. To manage such large collections of numerical information, we rely on simple structures that arrange numbers in a consistent way. These structures are generally called *arrays*. Depending on how many dimensions they have, they go by different names.

A single number is often called a *scalar*. A one-dimensional list of numbers is called a *vector*. A two-dimensional table of numbers, much like a spreadsheet with rows and columns, is called a *matrix*. More complex, multi-dimensional boxes of numbers are sometimes called *tensors*.

For the purposes of this course, we will only work with scalars, vectors, and matrices. To avoid confusion we will use different notations to distinguish clearly between these three types of objects:

- Scalars will be represented with regular letters, like x.
- Vectors will be represented with bold letters, like **x**.
- Matrices will be represented with bold capital letters, like **X**.

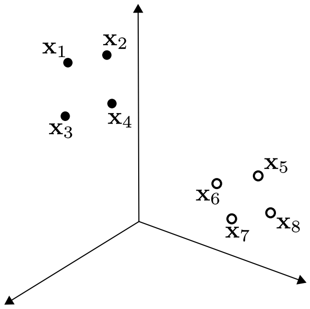
For example:

$$\mathrm{x} = 2, \qquad \mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \qquad \mathbf{X} = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}.$$

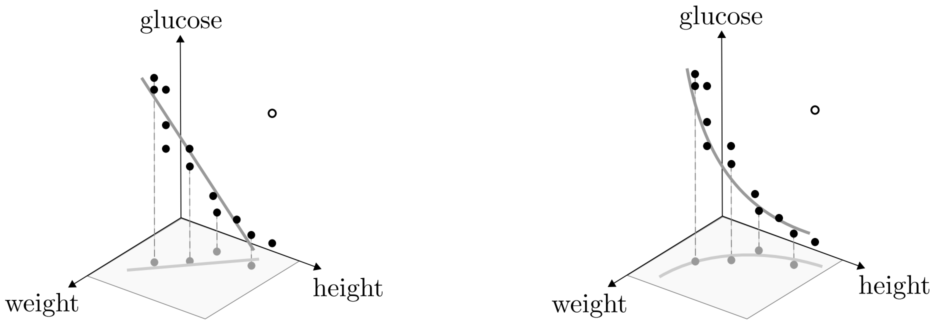
These notations will ease our exposition as we describe the how AI systems process data.

2.8 Data as Points in Space

A useful way to think about data is as points in space. When we represent an object using a list of numbers (such as height, weight, and glucose level), we can imagine each number as a coordinate along a different direction. Together, these coordinates place the object at a specific location in a multi-dimensional space:



Objects with similar properties appear as points that lie close to one another, while very different objects appear far apart. From this perspective, learning in AI often amounts to studying the shape of these clouds of points: finding clusters, boundaries, outliers, or patterns in how data is arranged in space:



2.9 More Examples on How Data Becomes Numbers

Recommender systems

Companies such as Amazon, Netflix, and Spotify collect information about their users in order to personalize recommendations. This information may include basic attributes such as age, sex, and income level, as well as users' ratings or interactions with products, such as movies watched, songs listened to, or items purchased.

For a given user, this information can be organized into a single vector where each entry corresponds to one characteristic or preference. For example, the information of the i^{th} user can be written schematically as

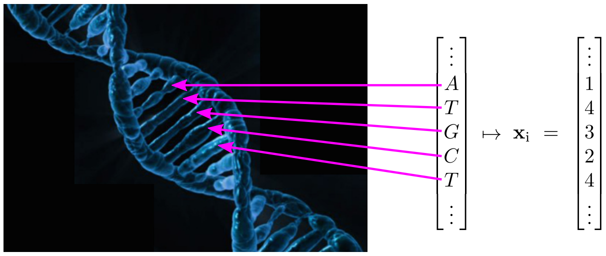
$$\mathbf{x}_{\mathrm{i}} = \begin{bmatrix} \text{age} \\ \text{sex} \\ \text{income} \\ \text{rating of item 1} \\ \text{rating of item 2} \\ \cdots \\ \text{rating of item M} \end{bmatrix}$$

Each user is thus represented as a point in a high-dimensional space, where each dimension corresponds to one variable or item.

The goal in this type of problem is to understand how user characteristics (such as age or income) and past preferences relate to future choices. In other words, we want to learn which patterns in the data help predict which products a user is likely to enjoy.

Genomics

The genome of an organism can be stored as a vector containing its corresponding sequence of nucleotides



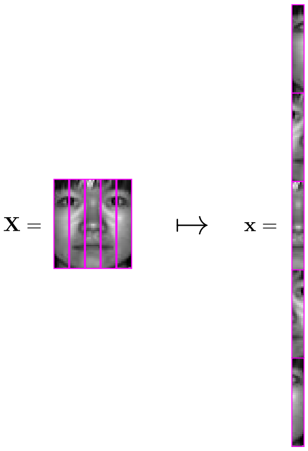
In this type of problem, the goal is to analyze these data vectors in order to identify which genes tend to be associated with particular diseases or observable traits, such as height or weight.

Image processing

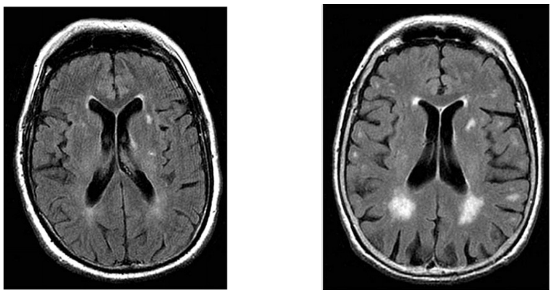
A grayscale image of size $m \times n$ can be stored as a data matrix $\mathbf{X}$ whose $(i, j)^{th}$ entry represents the gray intensity of the pixel located at position (i, j). In other words, the image is treated as a table of numbers recording how light or dark a particular pixel is.

This matrix can also be vectorized by stacking its columns one below another to form a single long vector $\mathbf{x}$. In this case, the image is represented as a list of mn numbers. This vector form is often convenient for analysis, since many AI and data-analysis methods expect inputs to be represented as vectors rather than matrices.

Furthermore, $\mathbf{X}$ can be *vectorized*, i.e., we can concatenate its rows to form a vector $\mathbf{x}$, with $D = mn$.

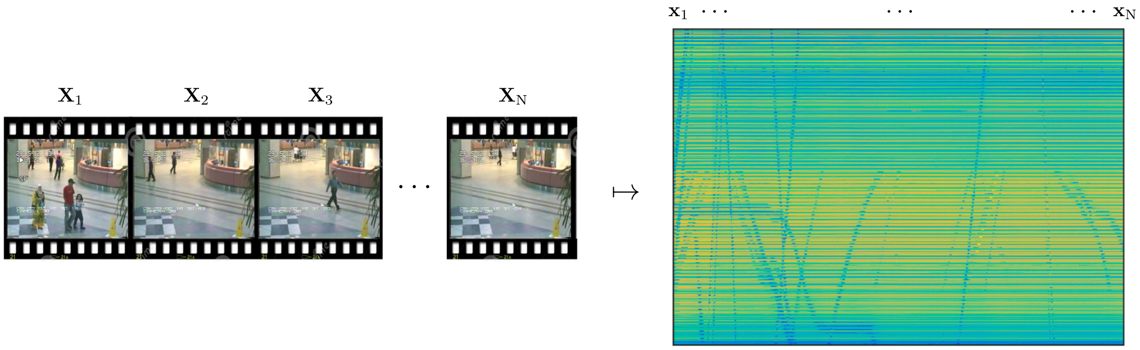


AI analyzes these vectors in order to interpret the images: to identify the objects they contain, to recognize and classify faces, or to support medical diagnosis. For example, one may ask whether the given magnetic resonance images (MRI) correspond to individuals with Alzheimer's disease.



Computer vision

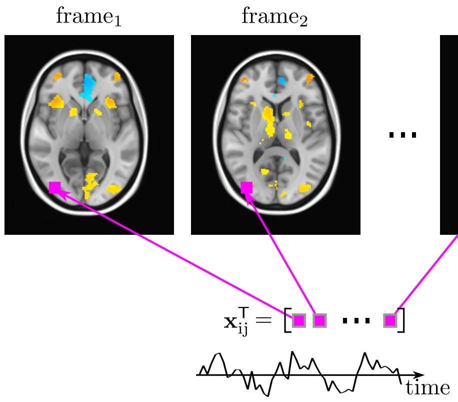
Each of the mn images $\mathbf{X}_1, \ldots, \mathbf{X}_\mathrm{N}$ that form a video can be vectorized to obtain vectors $\mathbf{x}_1, \ldots, \mathbf{x}_\mathrm{N}$, which can be concatenated into a large mn× matrix representing the entire video.



As with images, AI systems analyze numerical vectors to interpret video data. This includes tasks such as distinguishing foreground from background, tracking objects over time, and recognizing events or actions. These capabilities underpin applications in areas such as surveillance, defense, and robotics, where understanding motion and change is essential.

Neural activity

Functional magnetic resonance imaging (fMRI) generates a series of MRI images over time. Because oxygenated and deoxygenated hemoglobin have slightly different magnetic characteristics, variations in the MRI intensity indicate areas of the brain with increased blood flow and hence neural activity. The central task of fMRI is to reliably detect neural activity at different spatial locations (pixels) in the brain. The measurements over time at the $(\mathrm{i}, \mathrm{j})^{\mathrm{th}}$ pixel can be stored in a data vector $\mathbf{x}_{\mathrm{ij}}$.


frame$_1$
frame$_2$
frame$_D$
...
$\mathbf{x}_{ij}^{\mathsf{T}} = [\ \blacksquare\ \blacksquare\ \cdots\ \blacksquare\ ]$
time

By examining patterns across many such vectors, AI systems can help reveal how different parts of the brain respond to tasks, stimuli, or disease, contributing to a deeper understanding of how the brain functions.