AI Applications Lecture 1
Real-World Problems and Task Solving with Artificial Intelligence

SUZUKI, Atsushi
Jing WANG

September 1, 2025

Contents

1 Introduction

The purpose of this lecture is to learn the fundamentals of AI, particularly **Machine Learning** using **Neural Networks**, from the ground up.

1.1 Learning Outcomes of This Lecture

Through this lecture, students will aim to be able to perform the following tasks.

- Formulate tasks that humanity seeks to solve as a relationship between **input** and **output**.
- Mutually convert various real-world **entities** that humans handle and their corresponding **numerical sequences**.
- Formulate **Artificial Intelligence** as a **function determination problem**.

1.2 Policy of This Lecture

This lecture will begin with fundamental topics such as the definitions of machine learning and neural networks. While some students may have prior knowledge, there is a reason for deliberately starting with basic definitions.

The field of AI and machine learning is developing very rapidly, and specific technologies or models that are widely known today (for example, the architecture of a particular famous neural network) may well be outdated by the time you graduate. Therefore, this lecture emphasizes the understanding of more general and universal concepts that will remain applicable in the future, rather than learning specific techniques. To this end, we aim to redefine AI and machine learning in the most general terms possible and to understand that framework.

Furthermore, many AI-related lectures first teach the individual components of AI (e.g., activation functions, optimization algorithms) and then introduce application examples. However, this lecture adopts the reverse approach.

1. First, we will treat AI as a functional **black box** and begin by learning ”how to use it.”
2. Then, as needed, we will **white-box** the contents of the black box and uncover its mechanisms.

This approach has the following advantages.

- **Promotion of multifaceted understanding:** Even when dealing with the same concepts as other lectures, approaching them from a different perspective leads to a deeper understanding.

- **Acquisition of practical knowledge:** Even if you find it difficult to understand the detailed theory during the lecture, you will at least acquire practical knowledge of ”how to use” AI.
- **Acquisition of universal knowledge:** While the specific implementation methods of AI change daily, the ”framework for using AI” taught first in this lecture is relatively resistant to change and will be useful for a long time.
- **Addressing realistic constraints:** The details of the learning methods for recent large-scale **Generative AI** are often kept as trade secrets and are not disclosed. It is also theoretically known that there is no ”panacea learning method for all problems,” and practical learning recipes are inevitably case-by-case. Therefore, it is efficient to understand the broad framework first and learn the specific detailed recipes when they become necessary.

2 Preliminaries: Mathematical Notations

Here, we will organize the basic mathematical notations used in this lecture.

- **Set**:
  - $\mathbb{R}$: The set of all real numbers.
  - $\mathbb{Z}$: The set of all integers.
  - $\mathbb{N}$: The set of all natural numbers (in this lecture, not including 0: $\{1, 2, 3, \ldots\}$).
  - $A = \{a, b, c\}$: Indicates that the set A consists of the elements a, b, c.
  - $x \in A$: Indicates that the element x belongs to the set A.
- **Function**:
  - $f : X \to Y$: Indicates that the function f is a **map** that takes an element of the set X as input and outputs an element of the set Y. X is called the **domain**, and Y is called the **codomain**.
  - $y = f(x)$: Indicates that the output of the function f for an input $x \in X$ is $y \in Y$.
- **Vector**:
  - Vectors are denoted by bold italic lowercase letters. Example: $\boldsymbol{v}$.

- $\boldsymbol{v} \in \mathbb{R}^n$: Indicates that the vector $\boldsymbol{v}$ is an n-dimensional real vector composed of a tuple of n real numbers.

$$\boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}, \quad v_i \in \mathbb{R} \text{ for } i = 1, \ldots, n$$

- **Matrix**:
  - Matrices are denoted by bold italic uppercase letters. Example: $\boldsymbol{A}$.
  - $\boldsymbol{A} \in \mathbb{R}^{m,n}$: Indicates that the matrix $\boldsymbol{A}$ belongs to the set of $m \times n$ real matrices.

$$\boldsymbol{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}, \quad a_{i,j} \in \mathbb{R} \text{ for } i = 1, \ldots, m, \; j = 1, \ldots, n$$

- **Tensor**:
  - In this lecture, we treat tensors as multi-dimensional arrays. A vector can be considered a 1st-order tensor, and a matrix can be considered a 2nd-order tensor.
- **Note on Implementation**:
  - Unless otherwise specified, both vectors and matrices have real numbers ($\mathbb{R}$) as their components. However, when implementing them on a computer, finite subsets such as floating-point numbers or integer sets with upper and lower bounds are used. These subsets can be considered approximations of the set of real numbers, but it is important to note that their discrete and finite nature may lead to properties different from those of the set of real numbers (though their theoretical analysis is not easy).

3 What is AI?

This lecture is about "AI Applications," but what exactly is "Artificial Intelligence"? Defining this term is extremely difficult.

The word 'Artificial', in the modern context, can be interpreted as almost synonymous with "realized using a **computer**." However, for the word 'Intelligence', there is no unified definition even among experts, and it is uncertain whether a common understanding will be reached in the future.

From an engineering perspective, however, whether a technology can be called ”intelligent” is not the essential question. What is important is whether that technology can contribute to human society. Therefore, this lecture will not delve into any philosophical discussions regarding the definition of intelligence, and will define AI in an extremely practical manner as follows.

Definition 3.1 (Definition of AI in this lecture)**.** **Artificial Intelligence (AI)** is a procedure, implemented on a computer, for solving some task that humans want to solve.

Based on this definition, we will consider how AI ”solves tasks.”

4 Formulation of Tasks: Problems as Input-Output Relationships

Many of the tasks that humans want computers to solve can be formulated as a problem of ”given a certain input, return an appropriate output.” This is the idea of viewing a task as a kind of transformation rule, that is, a correspondence between inputs and outputs.

Let’s look at some specific examples.

- **Chat AI:**
  - **Input:** The text of a question or command typed by the user. (e.g., ”What’s the weather today?”)
  - **Output:** The AI’s response text to that input. (e.g., ”The weather in Tokyo is sunny.”)
- **Autonomous Driving:**
  - **Input:** Information obtained from numerous sensors such as camera images, LiDAR (laser sensors), GPS, and vehicle speed sensors.
  - **Output:** Signals for controlling the vehicle, such as the steering wheel angle and the amount of accelerator or brake depression.
- **Text-to-Image (Image Generation AI):**
  - **Input:** A sentence describing the content of the image to be generated (**prompt**). (e.g., ”A cat flying in space, oil painting style”)
  - **Output:** Image data generated according to the instructions in the input text.
- **Medical Diagnosis from Images:**
  - **Input:** Medical image data such as X-rays, CT scans, or MRI images.

- **Output:** The diagnosis determined from the image. For example, a label indicating whether a lesion is ”positive” or ”negative.”

- **Fraudulent Credit Card Use Detection:**
  - **Input:** Various pieces of information from when a credit card is used (transaction time, location, amount, purchase item category, past usage frequency, etc.).
  - **Output:** A judgment indicating whether the transaction is ”highly likely to be fraudulent” or ”a normal transaction.”

As these examples show, even seemingly completely different tasks share a common structure of ”input → output.”

5 Numerical Representation of Real-World Entities

What computers can handle are, fundamentally, **bit strings** of 0s and 1s, or the **byte strings** and numbers they form. Therefore, to process the various inputs and outputs seen in the previous section on a computer, they must first be converted into numerical data. This conversion rule is defined by humans according to the task.

Below are examples of conversion rules between representative entities and numerical data.

5.1 Natural Language

A sentence is a sequence of characters. The rule that maps each character to a specific byte sequence is called a **character encoding**.

- **ASCII (American Standard Code for Information Interchange):** Maps alphabets, numbers, symbols, etc., to 7-bit (usually 1-byte) integers. For example, ‘A‘ is 65, and ‘B‘ is 66. The ASCII code table for representative printable characters is shown below.

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
32	20	(space)	56	38	8	80	50	P	104	68	h
33	21	!	57	39	9	81	51	Q	105	69	i
34	22	”	58	3A	:	82	52	R	106	6A	j
35	23	#	59	3B	;	83	53	S	107	6B	k
36	24	$	60	3C	¡	84	54	T	108	6C	l
37	25	%	61	3D	=	85	55	U	109	6D	m
38	26	&	62	3E	¿	86	56	V	110	6E	n
39	27	’	63	3F	?	87	57	W	111	6F	o
40	28	(	64	40	@	88	58	X	112	70	p
41	29	)	65	41	A	89	59	Y	113	71	q
42	2A	*	66	42	B	90	5A	Z	114	72	r
43	2B	+	67	43	C	91	5B	[	115	73	s
44	2C	,	68	44	D	92	5C	\	116	74	t
45	2D	-	69	45	E	93	5D	]	117	75	u
46	2E	.	70	46	F	94	5E	ˆ	118	76	v
47	2F	/	71	47	G	95	5F	_	119	77	w
48	30	0	72	48	H	96	60	‘	120	78	x
49	31	1	73	49	I	97	61	a	121	79	y
50	32	2	74	4A	J	98	62	b	122	7A	z
51	33	3	75	4B	K	99	63	c	123	7B	{
52	34	4	76	4C	L	100	64	d	124	7C	—
53	35	5	77	4D	M	101	65	e	125	7D	}
54	36	6	78	4E	N	102	66	f	126	7E	˜
55	37	7	79	4F	O	103	67	g			

- **UTF-8[1] (Unicode Transformation Format-8):** This is one of the encoding schemes of the Unicode standard, which includes characters from all over the world, including those from the Chinese character cultural sphere (Kanji, Hangul, Hiragana, Katakana). It represents a single character with a variable-length byte sequence of 1 to 4 bytes. This allows characters like ‘A‘ to be represented by 1 byte, same as in ASCII, while ‘あ ‘ and many common Kanji characters are represented by 3 bytes.

  Here are some encoding examples for Japanese Kanji (3 bytes).

  – ”日” (Unicode: U+65E5) → UTF-8 (hex): `E6 97 A5`

  – ”本” (Unicode: U+672C) → UTF-8 (hex): `E6 9C AC`

  – ”語” (Unicode: U+8A9E) → UTF-8 (hex): `E8 AA 9E`

  Here are examples of characters that use other byte lengths.

  – Kanji ”體” (Unicode: U+9AD4) → UTF-8 (hex): `E9 AB 94` (3 bytes)

[1] https://www.unicode.org/glossary/#UTF_8

– Greek letter ”Ω” (Unicode: U+03A9) → UTF-8 (hex): `CE A9` (2 bytes)

– Emoji ”⊠⊠[2]” (Unicode: U+1F602) → UTF-8 (hex): `F0 9F 98 82` (4 bytes)

With this rule, any text can be converted into a unique byte sequence.

Remark 5.1 (On the Technical Significance of Emoji)**.** Emoji, originating from Japan, are now used worldwide. As a Japanese author, I would like to mention their technical significance, which is important for regions like the Chinese character cultural sphere. Since Emoji are multi-byte characters not included in ASCII, their popularization has globally motivated support for multi-byte characters. For example, people in the Chinese character cultural sphere have also benefited from the support for multi-byte characters in various technologies, such as the increasing number of web environments where Kanji can be used. Until then, for the computer and web technology communities, which tended to be led by the English-speaking world, supporting multi-byte characters like Kanji was not an urgent issue, and even if support was implemented, there were few ways to verify its correctness. In that sense, the spread of Emoji is an important historical fact, for example, for people in the Chinese character cultural sphere.

Also, as a Japanese author, I should note that the word Emoji comes from the Japanese words ”絵” (e, picture) and ”文字” (moji, character). I have heard that many people in the alphabetic cultural sphere mistakenly believe it originates from the word ”emotion,” so I am noting this for the record. The practice of assigning character codes to pictures was already being done by Japanese newspaper companies in the 1950s[a]. Later, on Japanese mobile phones in the 2000s before the spread of smartphones, emoji were so abundant that one could communicate using them alone (based on the author's own experience, though there was no compatibility between different mobile carriers). However, the contribution of the Gmail team in assigning Unicode points was also significant for the global spread of emoji outside of Japan[b].

[a] http://etlcdb.db.aist.go.jp/etlcdb/etln/etl2/e2code.jpg

[b] https://japan.cnet.com/article/20409186/

5.2 Image

A color image is a collection of **pixels**. The color of each pixel can be represented by a combination of the intensities of the three primary colors of light: Red, Green, and Blue.

- **Bitmap representation:** Let the height of the image be H and the width be W. If the color of each pixel is represented by integer values from 0 to 255 for R, G, and B respectively, a single color image can be represented as a 3rd-order integer tensor $\underline{\boldsymbol{I}} \in \mathbb{Z}^{3\times H\times W}$ of size $3\times H\times W$.

  As a specific example, consider a small 2×2 pixel image. Suppose the color of each pixel is as follows.

[2] https://www.utf8-chartable.de/unicode-utf8-table.pl?start=127744&number=1024

– Top-left: Red (R:255, G:0, B:0)

– Top-right: Green (R:0, G:255, B:0)

– Bottom-left: Blue (R:0, G:0, B:255)

– Bottom-right: White (R:255, G:255, B:255)

This image is represented as a $3 \times 2 \times 2$ tensor $\underline{\boldsymbol{I}}$ like this:

$$\underline{\boldsymbol{I}}_R = \begin{bmatrix} 255 & 0 \\ 0 & 255 \end{bmatrix}, \quad \underline{\boldsymbol{I}}_G = \begin{bmatrix} 0 & 255 \\ 0 & 255 \end{bmatrix}, \quad \underline{\boldsymbol{I}}_B = \begin{bmatrix} 0 & 0 \\ 255 & 255 \end{bmatrix}$$

Here, $\underline{\boldsymbol{I}}_R, \underline{\boldsymbol{I}}_G, \underline{\boldsymbol{I}}_B$ are the 2×2 matrices corresponding to the red, green, and blue channels, respectively.

5.3 Video

A video is a sequence of images that change over time.

- **Tensor representation:** Add a time dimension to the tensor representation of an image. A video with T frames can be represented as a 4th-order tensor $\underline{\boldsymbol{V}} \in \mathbb{Z}^{3 \times T \times H \times W}$ of size $3 \times T \times H \times W$.

5.4 Audio

Audio is the vibration of air (a wave). When recorded with a microphone, it yields a voltage signal that changes over time.

- **PCM (Pulse Code Modulation):** An analog signal, the sound wave, is divided at regular time intervals (**sampling**), and the amplitude at each point in time is approximated by an integer value (**quantization**). This process is illustrated in the diagram below. A smooth analog signal (blue line) is measured at regular time intervals (sampling), its value is assigned to the nearest discrete level (quantization), and it is recorded as a sequence of integers.

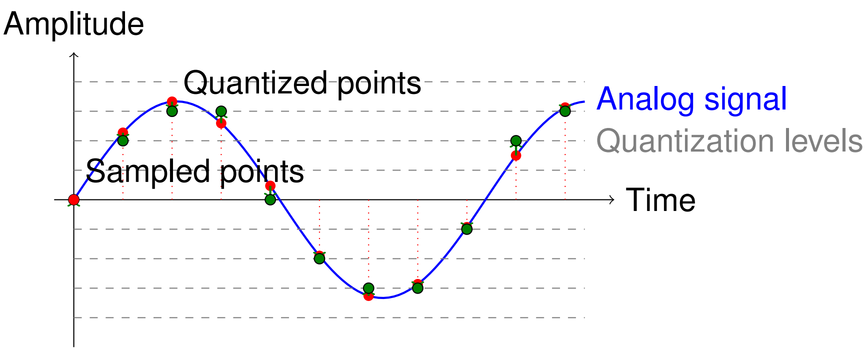

Figure 1: Schematic diagram of sampling and quantization of an audio signal

In the case of stereo audio, this process is performed for both the left and right channels, resulting in two integer sequences $\boldsymbol{l}, \boldsymbol{r}$, where $\boldsymbol{l}, \boldsymbol{r} \in \mathbb{Z}^D$. Here, the number of samples D is given by $D = fL$, using the duration L $[\mathrm{s}]$ and the sampling frequency f $[\mathrm{s}^{-1}]$.

In this way, various real-world entities can be converted into numerical data (vectors or tensors) based on clear rules.

6 Redefining AI: The Function Determination Problem

Now, let's organize the discussion so far.

1. A task to be solved can be seen as a correspondence of "input entity → output entity."
2. Input and output entities can be converted into "input numerical data → output numerical data" through appropriate rules.

Through these two steps, the problem we want AI to solve ultimately reduces to the problem of "given some input numerical data, compute the appropriate output numerical data."

In mathematical terms, this is none other than the **problem of finding a function** from the set of all input numerical data (**input space**) to the set of all output numerical data (**output space**).

Definition 6.1 (Engineering Goal of AI)**.** To solve a task is to construct a function f on a computer that imitates an unknown ideal function $f^* : \mathcal{X} \to \mathcal{Y}$ representing the input-output relationship of that task.

Here, $\mathcal{X}$ is the set of input data (input space), and $\mathcal{Y}$ is the set of output data (output space).

6.1 Input and Output Spaces in Each Application Example

Let's redefine the previous application examples within this function framework.

- **Chat AI:**
  - Input space $\mathcal{X}$: The set of byte sequences encoded in UTF-8.
  - Output space $\mathcal{Y}$: Similarly, the set of byte sequences encoded in UTF-8.
- **Autonomous Driving:** Let the number of sensors be D_{in} and the number of control signals be D_{out}.
  - Input space $\mathcal{X}$: A subset of the D_{in}-dimensional vector space containing real values from each sensor. $\mathcal{X} \subset \mathbb{R}^{D_{in}}$.

– Output space $\mathcal{Y}$: A subset of the D_{out}-dimensional vector space containing real values for each control signal. $\mathcal{Y} \subset \mathbb{R}^{D_{out}}$.

- **Text-to-Image:**
  – Input space $\mathcal{X}$: The set of byte sequences representing the prompt.
  – Output space $\mathcal{Y}$: The set of tensors representing $H \times W$ pixel RGB images. $\mathcal{Y} = \mathbb{Z}^{3 \times H \times W}$.
- **Medical Diagnosis from Images:**
  – Input space $\mathcal{X}$: The set of tensors representing medical images. $\mathcal{X} \subset \mathbb{Z}^{C \times H \times W}$ (C is the number of channels).
  – Output space $\mathcal{Y}$: The set representing the diagnosis results. $\mathcal{Y} = \{0, 1\}$ (0: negative, 1: positive).
- **Fraudulent Credit Card Use Detection:** Let there be D features for a transaction.
  – Input space $\mathcal{X}$: The set of real vectors consisting of D features. $\mathcal{X} \subset \mathbb{R}^{D}$.
  – Output space $\mathcal{Y}$: The set representing the judgment result. $\mathcal{Y} = \{0, 1\}$ (0: normal, 1: fraudulent).

Thus, we can see that any task can be uniformly described as a problem of determining a function f from an input space $\mathcal{X}$ to an output space $\mathcal{Y}$.

7 Summary and Future Outlook

In this lecture, we have reorganized the diverse tasks that AI tackles from a mathematical perspective.

7.1 Today's Summary

- **Task as Input-Output:** Many tasks that humanity seeks to solve can be formulated as a problem of determining an output given an input.
- **Numerical Representation of Entities:** Real-world entities such as text, images, and audio are converted into numerical data (vectors or tensors) that computers can handle, according to clear rules like UTF-8 or RGB bitmaps.
- **AI and the Function Determination Problem:** From the above two points, solving a task using AI can be equated to the problem of determining an appropriate function $f : \mathcal{X} \to \mathcal{Y}$ from an input space to an output space.

7.2 Preview of the Next Lecture

We have seen that solving a task can be equated with ”determining a function.” So, how can this function f be realized on a computer? In particular, when the ideal function f^* is unknown, how can we make f ”learn” to be close to it?

In the upcoming lectures, we will learn about the neural network, which is a concrete component of this function f, and unravel how it learns complex input-output relationships.