

INTRODUCTION TO ARTIFICIAL INTELLIGENCE

From Data Science → Machine Learning → Deep Learning →
Transformers → Generative AI

MODULE 1: THE AI SPECTRUM AND THE EVOLUTION TOWARD MODERN AI

The best way to begin studying Artificial Intelligence is not by immediately defining a robot, neural network, or chatbot.

Instead, begin with the **problem of data and information**.

Computers have always processed data. What has changed dramatically is:

- the amount of data available;
- the variety of data;
- the speed at which data is generated;
- the computational resources available;
- the algorithms used to learn from that data.

This gives us a useful conceptual progression:

```
graph TD; A[Data Collection / Storage] --> B[Data Warehousing / Data Mining]; B --> C[Data Science]; C --> D[Machine Learning]; D --> E[Deep Learning]; E --> F[Transformers]; F --> G[Large Language Models]; G --> H[Generative AI]; H --> I[AI Applications / Intelligent Systems];
```

Data Collection / Storage
↓
Data Warehousing / Data Mining
↓
Data Science
↓
Machine Learning
↓
Deep Learning
↓
Transformers
↓
Large Language Models
↓
Generative AI
↓
AI Applications / Intelligent Systems

This is a **conceptual progression**, not a claim that each field completely replaced the previous one.

Data warehouses did not disappear because machine learning appeared. Statistics did not disappear because deep learning appeared. Traditional algorithms are still extremely useful.

Rather, each stage adds new ways of extracting value from information.

1. Data Warehousing and Data Mining

The first problem faced by organizations was:

How do we store and organize the enormous amounts of data that we generate?

Businesses began collecting information about:

- customers;
- sales;
- employees;
- products;
- transactions;
- inventory;
- suppliers;
- financial activity.

A **data warehouse** provides structured storage for large quantities of organizational data, particularly data intended for analysis and reporting.

The emphasis is often on questions such as:

What happened?

For example:

How many products were sold last month?
Which branch had the highest sales?
Which customers purchased product X?
What was the revenue in 2025?

This is largely **descriptive analysis**.

Data Mining

Data mining goes further.

Instead of simply asking what happened, we begin looking for useful patterns hidden within the data.

For example:

Customers who buy A frequently also buy B.
Sales increase during certain periods.
Certain transactions have unusual characteristics.
Certain groups of customers behave similarly.

Therefore:

Data Mining → Discovery of useful patterns

The important transition is from:

Storage → Analysis → Pattern discovery

2. Data Science

As organizations obtained increasingly large and complicated datasets, simply storing and querying data was no longer sufficient.

This contributed to the growth of **Data Science**.

Data science combines ideas from areas such as:

- statistics;
- mathematics;
- programming;
- databases;
- visualization;
- domain knowledge;
- machine learning.

A data scientist may ask:

What is happening in this dataset?

Why is it happening?

What patterns exist?

Can we predict what happens next?

Can we use the data to make better decisions?

Therefore, data science is broader than simply machine learning.

A simplified view is:

```
Data
↓
Cleaning
↓
Exploration
↓
Statistics
↓
Visualization
↓
Modelling
↓
Prediction / Insight
↓
Decision
```

Machine learning can be one component of this pipeline.

3. The Transition from Data Science to Machine Learning

This is where your lecturer's discussion becomes particularly important.

Traditional programming usually works approximately like this:

```
Rules + Data
↓
Computer Program
↓
Output
```

The programmer explicitly determines the rules.

For example:

```
IF marks >= 80
    Grade = A
ELSE IF marks >= 70
    Grade = B
...
```

But consider a more difficult problem:

Given thousands of photographs, determine whether each photograph contains a cat.

It would be extremely difficult to write every rule manually.

What colour must the cat be?

What shape must its ears have?

What if it is facing away?

What if the photograph is dark?

What if the cat is partially hidden?

Machine learning provides another approach.

```
Examples / Data
  ↓
Learning Algorithm
  ↓
Trained Model
  ↓
New Data
  ↓
Prediction
```

The programmer specifies the learning procedure rather than manually specifying every possible rule.

This is one of the fundamental transitions:

Programming rules explicitly

toward

Learning patterns from examples

4. Machine Learning

Machine Learning (ML) is a major approach within AI in which computational models learn patterns from data and use those learned patterns to make predictions, classifications, decisions, or representations.

A simple abstraction is:

$$f_{\theta}(x) \rightarrow \hat{y}$$

where:

- x = input;
- θ = learned parameters;
- $\hat{y}$ = model output.

During training, the model adjusts its parameters so that its predictions become more useful according to an objective or loss function.

For example:

House Features $\rightarrow$ ML Model $\rightarrow$ Predicted House Price

The model does not memorize one fixed answer for every possible house.

It attempts to learn a relationship between the input features and the output.

5. Statistical Models and Prediction

Your lecturer's notes mention:

“prediction, statistical models”

This is an important connection.

Many machine-learning models can be understood statistically.

Suppose we want to predict house price.

We might have:

x_1 = house size

x_2 = number of bedrooms

x_3 = number of bathrooms

and:

y = house price

A simple linear model might be:

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$

The model learns the parameters:

$\theta_0, \theta_1, \theta_2, \theta_3$

from training data.

Stanford's CS229 notes introduce supervised learning through exactly this kind of feature-vector representation and linear hypothesis.

6. Linear Regression

Linear regression is one of the simplest predictive machine-learning models.

It is normally used when the target variable is continuous.

Examples:

```
House features → House price
Study hours → Exam score
Advertising expenditure → Sales
Temperature → Energy consumption
```

A simple one-variable model is:

$$\hat{y} = \theta_0 + \theta_1 x$$

With multiple features:

$$\hat{y} = \theta^T x$$

The model learns values of θ that minimize an error/loss measure over the training examples.

A common loss is mean squared error:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y^{(i)} - \hat{y}^{(i)})^2$$

The important idea at this stage is not to memorize the equation alone.

Understand the process:

```
Input features
  ↓
Linear equation
  ↓
Prediction
  ↓
Compare prediction with actual value
  ↓
Calculate error
  ↓
Adjust parameters
```

↓
Repeat

Google's Machine Learning Crash Course provides an interactive treatment of linear regression, loss, gradient descent and hyperparameter tuning.

7. Discrete Models and Decision Trees

Not every problem produces a continuous numerical output.

Sometimes we want a category.

For example:

Email → Spam / Not Spam
Patient → High Risk / Low Risk
Transaction → Fraud / Legitimate
Image → Cat / Dog / Bird

This is a **classification problem**.

One model used for classification is the **decision tree**.

A decision tree repeatedly asks questions about features:

```
      Income > X?  
      /  
    Yes      No  
    /        /  
  Age > Y?   Class B  
  /         /  
Yes      No  
/         /  
Class A  Class C
```

The tree creates decision boundaries by splitting the data according to feature values.

Decision trees can also be used for regression.

The important distinction is therefore:

Regression → continuous output

Classification → categorical output

Google's classification material discusses binary and multiclass classification as well as accuracy, precision, recall, ROC and AUC.

MODULE 2: THE GROWTH OF DATA AND COMPUTATIONAL POWER

At this point the lecturer asks an important question:

Why did machine learning become increasingly important?

One major answer is the growth of data.

8. The Explosion of Data

Modern systems generate enormous quantities of information.

Examples include:

- social media posts;
- photographs;
- videos;
- emails;
- online transactions;
- GPS coordinates;
- sensor readings;
- medical records;
- financial transactions;
- web pages;
- machine logs;
- audio recordings.

The amount of data available today is far greater than what was available when many traditional AI systems were developed.

This produces an important relationship:

More Digital Data + Better Algorithms + More Computing Power → More Capable AI
--

However, remember:

More data / automatically better data

The quality and representativeness of the data remain critical.

9. Computational Resources: RAM, CPU, GPU and Beyond

Your lecturer specifically mentioned:

RAM, CPU and other resources

This refers to the computational infrastructure needed to process increasingly large datasets and models.

CPU

The Central Processing Unit performs general-purpose computation.

RAM

Random Access Memory provides temporary working space for data and programs.

GPU

Graphics Processing Units became particularly important for deep learning because neural-network training involves enormous quantities of parallel numerical operations.

Modern AI infrastructure can also use:

- TPUs;
- AI accelerators;
- distributed computing systems;
- high-bandwidth memory;
- large-scale cloud infrastructure.

The basic idea is:

Larger Data + Larger Models $\Rightarrow$ Greater Computational Requirements

This relationship becomes especially important when we reach deep learning and large language models.

MODULE 3: DATA QUALITY, DISTRIBUTIONS AND MODEL CORRECTNESS

Your lecturer then moves toward a very important question:

Can data be skewed?

Yes.

And understanding the data is essential before training a model.

10. Data Distribution

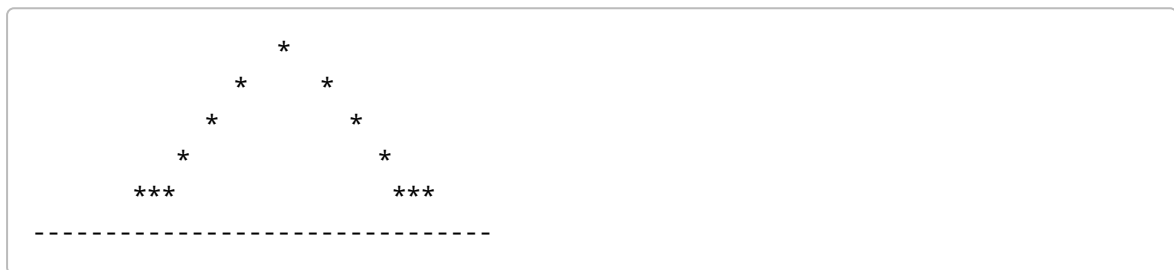
A dataset contains observations.

The way values are distributed is called the **distribution** of the data.

A common theoretical distribution is the normal distribution:

$$X \sim N(\mu, \sigma^2)$$

It has the familiar bell-shaped form.



The centre is approximately:

$$\mu$$

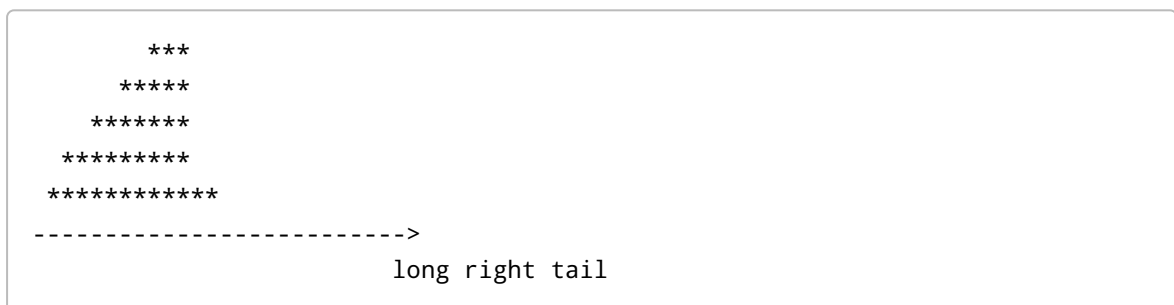
and the spread is related to:

$$\sigma$$

But real-world datasets are not always normally distributed.

11. Skewed Data

Data may be **right-skewed** or **positively skewed**.



It may also be **left-skewed** or **negatively skewed**.

```
      ***
     *****
    *****
   *****
  <-----
  long left tail
```

Examples of variables that may have strongly skewed distributions include:

- income;
- transaction values;
- waiting times;
- population sizes.

Why does this matter?

Because the distribution affects:

- summary statistics;
- visualization;
- scaling;
- outlier detection;
- assumptions of statistical models;
- model performance.

12. Data Is Not Always “Normal”

A common beginner mistake is assuming:

“Machine-learning data should be normally distributed.”

That is not generally true.

Different algorithms make different assumptions.

Some algorithms are relatively insensitive to distributional shape, while others can be affected substantially by scale and distribution.

For example, feature scaling can be important for algorithms based on distances or gradient-based optimization.

Google's ML Crash Course includes numerical-data analysis, normalization, scaling and statistical exercises as part of its data preparation material.

Therefore the correct workflow is:

```
Collect Data
  ↓
Explore Data
  ↓
Understand Distribution
  ↓
Clean / Transform / Represent
  ↓
Train Model
```

not:

```
Collect Data → Immediately Train Model
```

13. The Problem of Biased or Unrepresentative Data

Another important issue is whether the dataset actually represents the environment in which the model will operate.

Suppose a facial-recognition model is trained predominantly on one demographic group.

Even if the model performs extremely well on its training distribution, its performance may differ on another population.

Therefore:

Training Distribution / Real-World Distribution

can produce poor generalization.

This connects to the lecturer's broader question:

"Are the existing systems the best?"

A system cannot be considered good merely because it produces a high number during training.

We must ask:

Good according to what data?

Good according to what metric?

Good for which users?

Good under which conditions?

14. Model Accuracy and Correctness

The lecturer notes:

“Talk about accuracy and mean of correctness of a model.”

Accuracy is one measure of correctness.

For classification:

$$Accuracy = \frac{Correct\ Predictions}{Total\ Predictions}$$

Suppose:

950

out of

1000

predictions are correct.

Then:

$$Accuracy = 95\%$$

But this does not mean the model is necessarily good.

15. Why Accuracy Can Mislead

Imagine a dataset containing:

990 healthy patients
10 patients with a rare disease

Suppose the model predicts:

Everyone = Healthy

It will be correct for 990 out of 1000 cases:

$$Accuracy = 99\%$$

Yet it detects:

0/10

of the patients who actually have the disease.

Therefore accuracy alone is insufficient.

We may need:

- precision;
- recall;
- sensitivity;
- specificity;
- F1-score;
- ROC-AUC;
- confusion matrix.

This is one of the major lessons from the lecturer's emphasis on **quality**.

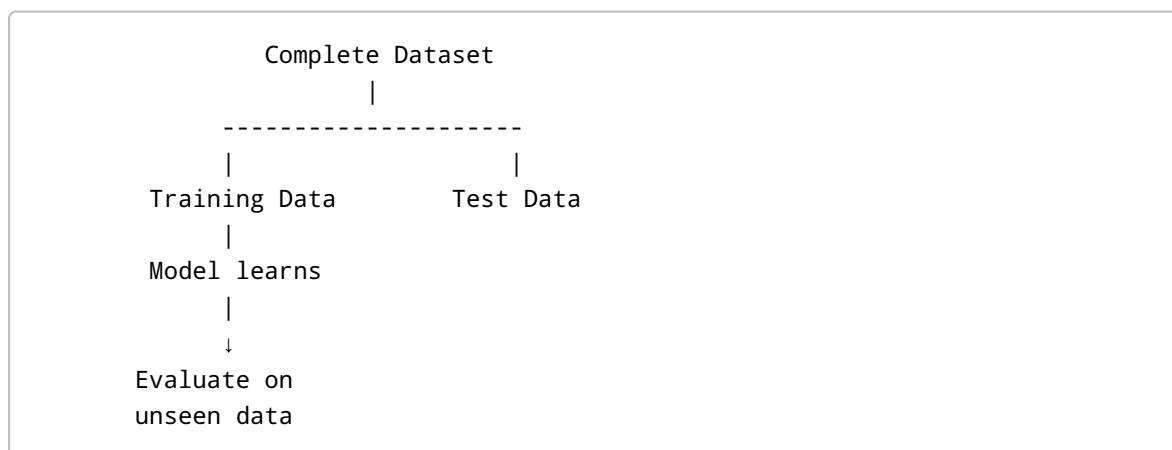
16. Training, Testing and Generalization

A machine-learning model should not normally be evaluated only on the same data used to train it.

Why?

Because we want to know whether it can work on **unseen data**.

A simplified dataset arrangement is:



Sometimes a third validation set is used:

Training → learn parameters
Validation → choose/tune model
Test → final evaluation

The purpose is to estimate **generalization**.

17. Overfitting

Suppose:

$$\textit{Training Accuracy} = 99\%$$

but:

$$\textit{Test Accuracy} = 65\%$$

This suggests that the model may have **overfit** the training data.

It learned patterns that are too specific to the training examples.

The goal is not:

“Memorize the training set.”

The goal is:

Learn patterns that generalize to new examples.

Google's current ML Crash Course explicitly includes datasets, generalization and overfitting as core modules.

MODULE 4: SUPERVISED AND UNSUPERVISED LEARNING

Your lecturer explicitly writes:

“supervised / unsupervised”

This is one of the central foundations of machine learning.

18. Supervised Learning

In supervised learning, the training examples contain both:

$$X$$

and:

$$Y$$

where Y represents the target.

For example:

Input		Output
Hours studied = 5	→	Exam mark = 72
Hours studied = 8	→	Exam mark = 86
Hours studied = 2	→	Exam mark = 48

The algorithm attempts to learn:

$$f(X) \approx Y$$

After training, we provide a new X , and the model predicts:

$$\hat{Y}$$

19. Main Supervised Learning Problems

Regression

Predict a numerical value.

$$X \rightarrow \mathbb{R}$$

Example:

$$\text{House Features} \rightarrow \text{Price}$$

Classification

Predict a class.

$$X \rightarrow \{Class_1, Class_2, \dots, Class_k\}$$

Example:

$$Email \rightarrow Spam/NotSpam$$

20. Unsupervised Learning

Now remove the labels.

We have:

$$X_1, X_2, X_3, \dots, X_n$$

but no corresponding target:

$$Y$$

The algorithm attempts to discover structure.

```
Unlabeled Data
  ↓
Algorithm
  ↓
Patterns / Groups / Representations
```

This connects directly to your lecturer's notes:

"random points running algorithms"

The points may initially look like an unorganized collection of observations.

The algorithm attempts to discover meaningful structure.

21. Clustering

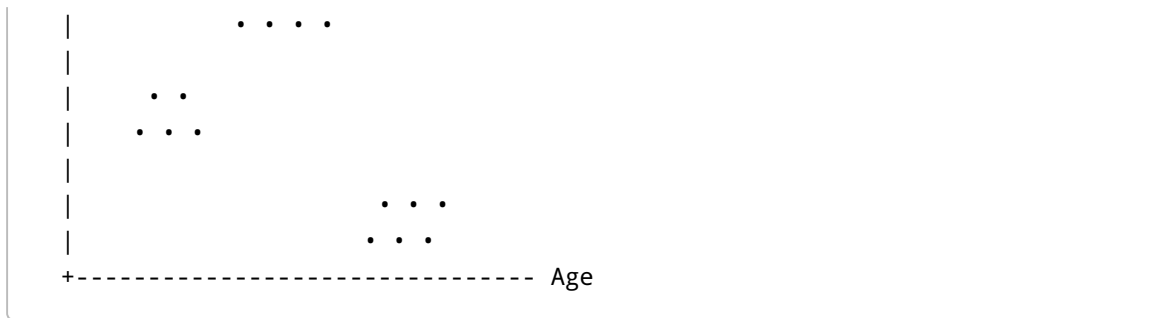
One important unsupervised-learning technique is **clustering**.

Imagine plotting customers using:

- age;
- income;
- spending frequency.

You might see:

```
Income
|
|      . . .
|
```



The algorithm may identify groups such as:

$$Cluster_1, \quad Cluster_2, \quad Cluster_3$$

without being told what those groups mean.

Google's clustering course specifically covers similarity measures, k-means, clustering evaluation and dimensionality reduction.

22. K-Means

A basic clustering algorithm is **k-means**.

Suppose:

$$k = 3$$

The algorithm attempts to identify three clusters.

Conceptually:

```

Initial centres
  ↓
Assign each point to nearest centre
  ↓
Recalculate centres
  ↓
Reassign points
  ↓
Repeat
  ↓
Stable clusters
  
```

The cluster centres are called **centroids**.

Mathematically, k-means attempts to minimize within-cluster squared distances:

$$J = \sum_{j=1}^k \sum_{x_i \in C_j} \|x_i - \mu_j\|^2$$

where:

- C_j = cluster j ;
- μ_j = centroid of cluster j .

This is an example of how an apparently simple idea—“group these points”—becomes a mathematical optimization problem.

23. Trends, Clusters, Breakpoints and Anomalies

Your notes mention:

“Trends, clusters, breakpoints etc.”

These are examples of structures we may attempt to discover.

Trend

A systematic direction in data.

For example:

Sales ↑ over time

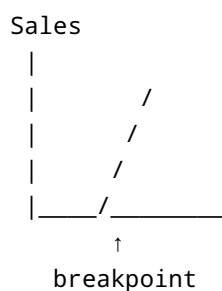
Cluster

A group of similar observations.

Breakpoint

A point where the statistical behaviour of a sequence changes.

For example:



Anomaly

An observation that differs substantially from the expected pattern.

Therefore unsupervised analysis is not only:

“Find clusters.”

It can involve discovering many forms of hidden structure.

24. Why Unsupervised Learning Matters

Your lecturer emphasizes the move toward unsupervised learning.

One major reason is the cost of obtaining labels.

Imagine:

10,000,000

images.

If every image requires a human expert to label, the process can become extremely expensive.

Unlabeled data, however, is often much easier to collect.

Modern AI therefore increasingly uses techniques that can learn useful representations from large amounts of unlabeled or weakly labelled data.

This leads naturally toward:

Unsupervised Learning → Representation Learning → Deep Learning → Modern Generative AI

MODULE 5: DATA REPRESENTATION — “AFTER A MATRIX, WHAT?”

This question in your lecturer's notes is extremely important.

Before a machine-learning algorithm can work with data, we need to represent that data mathematically.

25. Why Representation Matters

A computer does not naturally receive:

“This is a photograph of a dog.”

It receives numerical information.

Similarly, a computer does not directly process the abstract concept:

“This customer is likely to purchase product X.”

The information must be represented.

Therefore:

Real World Information $\rightarrow$ *Numerical Representation* $\rightarrow$ *Machine Learning Model*

This is why your lecturer says:

“Representation of data depending on the task.”

26. Scalar

A scalar is a single numerical value.

Example:

$$x = 5$$

It has no vector direction.

Examples:

Temperature = 25
Age = 21
Price = 50,000

27. Vector

A vector is an ordered collection of numbers.

$$x = [x_1 \ x_2 \ x_3]$$

For example, one student might be represented as:

$$x = [20 \ 4 \ 85]$$

where:

- age = 20;
- years of study = 4;
- average mark = 85.

This is called a **feature vector**.

The vector represents one observation in a mathematical feature space.

28. Matrix

A matrix contains rows and columns.

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{21} & x_{22} & x_{23} & x_{31} & x_{32} & x_{33} \end{bmatrix}$$

In a dataset:

Rows → observations/examples
Columns → features

For example:

Student	Age	Study Hours	Mark
A	20	5	72
B	21	8	86
C	19	3	61

can be represented as:

$$X = \begin{bmatrix} 20 & 5 & 72 & 21 & 8 & 86 & 19 & 3 & 61 \end{bmatrix}$$

This is why linear algebra becomes central to machine learning.

29. Matrix Multiplication

Neural networks perform enormous numbers of matrix operations.

A simple neural layer can be represented as:

$$z = Wx + b$$

where:

- W = weight matrix;
- x = input vector;
- b = bias vector;
- z = transformed representation.

Then an activation function is applied:

$$a = \sigma(z)$$

This basic operation is repeated across many layers in deep neural networks.

Therefore:

Linear Algebra is not optional background knowledge in modern AI.

30. Inverse of a Matrix

Your supplied example also includes:

“Inverse of a matrix”

For a square matrix A , its inverse A^{-1} satisfies:

$$AA^{-1} = A^{-1}A = I$$

where I is the identity matrix.

For example:

$$A = \begin{bmatrix} a & b & c & d \end{bmatrix}$$

has an inverse when:

$$ad - bc \neq 0$$

and:

$$A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b & -c & a \end{bmatrix}$$

Why does this matter?

Matrix operations provide tools for solving systems of equations and manipulating transformations.

However, in practical machine learning, we often avoid explicitly computing matrix inverses when numerical methods such as factorization or iterative optimization are more stable or efficient.

This is an important distinction between:

knowing the mathematical operation

and:

knowing how it should be used in practical computing.

31. Data Representation Depends on the Task

This is one of the most important statements in the lecture.

Different data types require different representations.

Tabular Data

Often:

$$X \in \mathbb{R}^{n \times d}$$

where:

- n = number of observations;
- d = number of features.

Images

A grayscale image may be represented as:

$$H \times W$$

where H is height and W is width.

A colour RGB image may be:

$$H \times W \times 3$$

because each pixel contains three colour channels.

Video

A video may require an additional time dimension:

$$T \times H \times W \times C$$

Text

Text can be represented using:

- tokens;
- one-hot vectors;
- word embeddings;
- contextual embeddings.

Therefore:

Same real-world information → different representations depending on task

32. From Matrices to Tensors

A tensor is a generalization of scalars, vectors and matrices to higher dimensions.

Scalar → 0 dimensions
Vector → 1 dimension
Matrix → 2 dimensions
Tensor → 3+ dimensions

For example:

Image:
Height × Width × Channels

Video:
Time × Height × Width × Channels

Deep-learning frameworks commonly manipulate tensors because neural networks operate on batches of multidimensional data.

This gives a natural progression:

Scalar → Vector → Matrix → Tensor

MODULE 6: DEEP LEARNING

Your lecturer then moves to:

“Deep learning: back to layers & transformers.”

This is the next major stage.

33. What Is Deep Learning?

Deep learning is a subfield of machine learning based primarily on neural networks with multiple layers.

A simplified network looks like:

```
Input
  ↓
Layer 1
  ↓
Layer 2
  ↓
Layer 3
  ↓
Output
```

The word **deep** refers mainly to the depth—the number of layers or transformations involved.

A neural layer can perform:

$$z = Wx + b$$

followed by:

$$a = f(z)$$

The output of one layer becomes the input to the next.

34. Why Multiple Layers?

Suppose we are recognizing an image.

A conceptual hierarchy might look like:

```
Pixels
  ↓
Edges
  ↓
Textures
  ↓
Shapes
  ↓
Parts
```

↓
Objects

The exact internal representations learned by a neural network are more complex than this simplified picture, but the hierarchy illustrates the motivation.

Deep learning allows the system to learn increasingly abstract representations.

This reduces the amount of manually engineered feature extraction required for many tasks.

It does **not** mean human-designed preprocessing is completely unnecessary.

35. Neural Network Parameters

A neural network contains learnable parameters.

For example:

W

represents weights and:

b

represents biases.

During training, the system attempts to find parameter values that minimize a loss function.

Conceptually:

```
Initial Parameters
  ↓
Make Prediction
  ↓
Calculate Loss
  ↓
Calculate Gradients
  ↓
Update Parameters
  ↓
Repeat
```

This process is associated with **gradient descent** and **backpropagation**.

The parameters can become extremely numerous.

A large modern neural network may contain billions of parameters.

36. Neural Network Video

A particularly useful first video is 3Blue1Brown's:

 video  But what is a neural network? — 3Blue1Brown  turn1youtube13 

It explains neurons, layers, weights, biases, matrix notation and how learning relates to the parameters.

This is an excellent bridge between your lecturer's:

“matrix”

and:

“deep learning / layers.”

MODULE 7: TRANSFORMERS

This is where your lecturer's handwritten note:

“transformers”

becomes extremely important.

37. Why Transformers?

Before Transformers became dominant, sequence-processing models commonly included:

- recurrent neural networks (RNNs);
- long short-term memory networks (LSTMs);
- convolutional architectures for some sequence tasks.

The Transformer architecture was introduced in the 2017 paper:

Attention Is All You Need.

The paper proposed a sequence-transduction architecture based entirely on attention rather than recurrence and convolution, and emphasized its improved parallelizability during training.

38. Attention

The central idea is **attention**.

Suppose the sentence is:

“The student put the book on the table because it was heavy.”

The meaning of “it” depends on its relationship with other words.

Attention gives the model a mechanism for assigning different importance to different parts of the context.

A simplified conceptual representation is:

$$Attention(Q, K, V)$$

where:

- Q = queries;
- K = keys;
- V = values.

The standard scaled dot-product attention is:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) V$$

You do not need to master this equation immediately.

First understand the idea:

The model calculates which pieces of information should influence one another.

39. Transformers and Parallelism

The original Transformer was significant partly because it removed recurrence from the proposed architecture and made training more parallelizable.

This matters because GPUs are very good at performing large numbers of parallel numerical operations.

However, be careful with the phrase:

“Transformers process everything simultaneously.”

That is an oversimplification.

During training, many operations can be highly parallelized.

But in an autoregressive language model, generating a response generally involves producing tokens sequentially:

$$Token_1 \rightarrow Token_2 \rightarrow Token_3 \rightarrow \dots$$

So the important distinction is:

Transformer training can be highly parallel

but:

autoregressive generation is sequential across generated tokens

40. Transformers and Language

A language model ultimately works with tokens.

For example:

"The cat is sleeping."

may be converted into a sequence of tokens:

[token1, token2, token3, token4, ...]

The model converts tokens into numerical representations called embeddings.

Conceptually:

```
Text
↓
Tokenization
↓
Token IDs
↓
Embeddings
↓
Transformer Layers
↓
Probability Distribution
↓
Next Token
```

MODULE 8: LARGE LANGUAGE MODELS AND GENERATIVE AI

Now the lecturer's discussion naturally moves toward modern AI.

41. Large Language Models

A Large Language Model (LLM) is a large neural language model trained on extensive text data.

Modern LLMs are commonly based on Transformer architectures, although “LLM” and “Transformer” are not identical concepts.

The basic autoregressive language-modelling objective can be expressed as predicting:

$$P(x_{t+1} | x_1, x_2, \dots, x_t)$$

In ordinary language:

Given the preceding context, estimate the probability distribution of what comes next.

The model learns statistical regularities from its training data.

42. Does This Mean the Model “Just Predicts the Next Word”?

This phrase is useful but incomplete.

Technically, many autoregressive language models predict the next **token**, not necessarily the next complete word.

A token might be:

- an entire word;
- part of a word;
- punctuation;
- a symbol.

The model repeatedly predicts the next token.

For example:

The student opened the ...

The model calculates probabilities for possible continuations.

It may assign different probabilities to:

```
book  
door  
window  
computer  
...
```

The generated response emerges from repeated token generation.

43. Generative AI

Generative AI refers to AI systems capable of producing new content.

Examples include:

Text

```
Question → Generated explanation
```

Code

```
Requirement → Generated program
```

Image

```
Text prompt → Generated image
```

Audio

```
Prompt → Generated speech/music
```

Video

```
Prompt → Generated video
```

Therefore:

Generative AI = AI systems that generate new content

Modern generative AI is heavily based on deep learning.

44. Where Does Generative AI Fit?

The hierarchy can be represented as:

```
Artificial Intelligence
    ↓
Machine Learning
    ↓
Deep Learning
    ↓
Neural Networks
    ↓
Transformer / other generative architectures
    ↓
Generative AI applications
```

But remember that this is a teaching hierarchy, not a strict statement that every generative model is a Transformer.

For example, generative systems can use different architectures depending on the modality and task.

45. Software-Based AI vs Physical AI

Your lecturer's second page moves from models to:

"hardware + software"

and:

"interfacing real environment + AI."

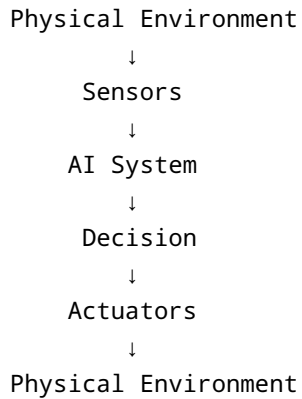
This is an important transition.

AI can exist purely as software:

```
User
    ↓
Application
    ↓
AI Model
```

↓
Text / Prediction / Decision

But AI can also interact with the physical world:



This creates a feedback loop.

MODULE 9: ROBOTICS, EMBEDDED SYSTEMS AND AUTOMATION

46. Robotics

Your lecturer explicitly mentions:

“Robots”

Robotics combines:

- hardware;
- software;
- sensing;
- control;
- planning;
- perception;
- decision-making;
- sometimes machine learning.

A robot may use a camera to perceive the environment:

Camera → Computer Vision → Object Detection

then:

Object Detection → Planning → Motor Commands

The robot therefore becomes an intelligent system interacting with the real environment.

47. Embedded AI

An embedded system contains computing functionality within a larger device.

Examples:

- smart cameras;
- autonomous vehicles;
- drones;
- industrial machines;
- mobile phones;
- medical equipment;
- smart appliances.

AI running on an embedded device introduces constraints.

For example:

Memory ↓

Power ↓

Latency requirements ↑

Therefore a model must not only be accurate.

It may also need to be:

- small;
- fast;
- energy efficient;
- reliable.

This connects directly to your lecturer's note about:

"efficiency."

48. AI and Automation

Automation refers to using technology to perform tasks with reduced human intervention.

AI can make automation more adaptive.

Traditional automation:

```
IF condition  
THEN action
```

AI-based automation may instead use learned patterns:

```
Sensor Data  
↓  
ML Model  
↓  
Prediction  
↓  
Decision  
↓  
Action
```

This is especially useful when the environment is too complex to encode entirely through fixed rules.

MODULE 10: AI APPLICATION DOMAINS

Your lecturer's notes mention:

Computer vision

NLP

Robotics

Automation

These are major application areas.

49. Computer Vision

Computer Vision (CV) allows machines to process and interpret visual information.

Typical tasks include:

- image classification;
- object detection;
- image segmentation;
- optical character recognition;
- image generation;
- medical-image analysis.

The basic pipeline may be:

Image → Numerical Representation → Neural Network → Prediction

50. Natural Language Processing

Natural Language Processing (NLP) deals with human language.

Applications include:

- translation;
- sentiment analysis;
- summarization;
- question answering;
- information extraction;
- text generation;
- conversational systems.

LLMs represent one major modern development in NLP.

51. Robotics

Robotics adds physical interaction.

A robot may combine:

CV + NLP + Planning + Control + ML + Hardware

For example, a service robot could:

1. hear a command;
2. interpret the language;
3. identify the requested object;
4. locate the object;
5. plan a path;
6. move toward it;
7. pick it up;
8. deliver it.

This demonstrates why AI is increasingly a **system discipline** rather than merely a model.

MODULE 11: APIs, MODELS AND USING AI IN SOFTWARE

Your lecturer's note appears to mention:

"API"

This is an important practical concept.

52. What Is an API?

API stands for:

Application Programming Interface.

It provides a defined way for one software system to interact with another.

An AI application might work like:

```
graph TD
    A[Your Application] --> B[API]
    B --> C[AI Model / Service]
    C --> D[API]
    D --> E[Your Application]
```

The application may send:

```
input text
model identifier
parameters
instructions
```

and receive:

```
model output
metadata
usage information
```

This means you can build an AI-powered application without personally training a large model from scratch.

53. Using a Model vs Training a Model

These are different activities.

Using a model

You send inputs to an existing model.

```
Input
↓
Pretrained Model
↓
Output
```

Fine-tuning a model

You start with an existing model and adapt it to a particular task/domain.

```
Pretrained Model
↓
Domain/task data
↓
Fine-tuned Model
```

Training from scratch

You construct/train the model using your own training pipeline.

```
Large Dataset
↓
Model Architecture
↓
Training
↓
Trained Model
```

Training a frontier-scale model from scratch is enormously more expensive than simply using an API or fine-tuning a smaller pretrained model.

This distinction is important when your lecturer says:

“Everyone can train.”

Today, many people can experiment with models, but that does not mean everyone can train frontier-scale models from scratch.

MODULE 12: BUILDING AN AI PERSONA AND PROMPTING

Your lecturer's notes contain:

"Building Persona"

and:

"Give us 5 surfaces — Pick one."

Although the exact wording is difficult to read, the broader concept is clearly connected to designing how an AI system behaves and interacts.

54. What Is an AI Persona?

A persona specifies characteristics such as:

- role;
- purpose;
- audience;
- tone;
- behaviour;
- constraints;
- output format.

For example:

```
Role:
Academic tutor

Audience:
First-year computer science students

Purpose:
Explain AI concepts

Style:
Technical but beginner-friendly
```

Constraint:
Use examples and equations where appropriate

This is not the same as changing the underlying model's intelligence.

Rather, we are giving the model contextual instructions that influence its behaviour.

55. Prompting as an Interface

A prompt can contain:

```
Role
+
Task
+
Context
+
Constraints
+
Output format
+
Examples
```

For example:

```
You are an introductory AI tutor.

Explain supervised learning to a first-year
computer science student.

Use one mathematical example and one
real-world example.

End with three examination questions.
```

This gives the model a clearer task than simply:

```
"Explain AI."
```

56. Prompting Does Not Eliminate Model Limitations

A very detailed prompt does not guarantee correctness.

We can still have:

$$\textit{Good Prompt} + \textit{Bad Information} \rightarrow \textit{Bad Output}$$

or:

$$\textit{Good Prompt} + \textit{Model Limitation} \rightarrow \textit{Incorrect Output}$$

Therefore prompting is an **interface technique**, not a substitute for model evaluation.

MODULE 13: CHALLENGING AI MODELS

This connects directly to your lecturer's repeated discussion of:

"challenge AI models"

and:

"what if it says 5?"

This is an important practical attitude.

57. Never Assume the Model Is Correct

Suppose we ask:

$$1 + 1 = ?$$

The correct answer is:

2

If an AI system says:

5

then something has gone wrong.

The lesson is not merely that the AI "made a mistake."

The deeper lesson is:

AI outputs must be evaluated according to the requirements of the task.

If the task is arithmetic, exact correctness is essential.

If the task is creative writing, multiple answers may be acceptable.

If the task is medical diagnosis, safety and clinical validity become extremely important.

Therefore:

Evaluation criteria depend on the task

58. Hallucination

Generative AI systems can produce information that sounds convincing but is unsupported or incorrect.

This is commonly called **hallucination**.

For example, a model might:

- invent a citation;
- invent a historical event;
- provide incorrect mathematics;
- attribute a quote to the wrong person;
- generate a nonexistent reference.

The important relationship is:

Fluency /

= *Truth*

A model can produce grammatically excellent output while still being factually wrong.

59. Why Do Generative Models Produce Errors?

A simplified explanation is that a language model learns statistical patterns in data and generates outputs according to those learned patterns and the decoding process.

It is not automatically a database containing a guaranteed verified answer for every question.

This is why:

Plausible Output / *Guaranteed Correct Output*

The problem becomes particularly important when a model is asked about:

- obscure facts;
- current information;
- exact calculations;
- precise citations;
- highly specialized subjects.

External tools, retrieval systems, calculators, code execution and human verification can improve reliability for appropriate tasks.

60. Challenge the Model

Instead of simply asking:

“Is this answer correct?”

we can test the model.

For example:

Explain your assumptions.
Check the calculation.
Give an independent verification.
Identify possible weaknesses.
Compare two possible answers.
Provide sources.

This does not guarantee correctness, but it creates opportunities for errors to be exposed.

For high-stakes applications, independent verification remains important.

MODULE 14: AI MODEL QUALITY AND EFFICIENCY

Your lecturer asks questions around:

quality

test

error

efficiency

These should be considered together.

61. Model Quality

Model quality is multidimensional.

A useful model may need to be:

- accurate;

- robust;
- efficient;
- fair;
- reliable;
- interpretable;
- secure;
- appropriately calibrated;
- affordable.

Therefore:

$$\boxed{\text{Good Model} / } = \text{Highest Accuracy Alone}$$

62. Error

Suppose the actual value is:

$$y$$

and the model predicts:

$$\hat{y}$$

Then:

$$\text{Error} = y - \hat{y}$$

For multiple observations, we can aggregate errors using metrics such as:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Other tasks require different loss functions and evaluation metrics.

63. Efficiency

Imagine two models:

Model A:
 Accuracy = 96%
 Memory = 2 GB
 Inference = 2 seconds

Model B:
 Accuracy = 95%

Memory = 100 MB
Inference = 20 ms

Which one is better?

There is no universal answer.

If the system is a powerful server processing one request per minute, Model A might be acceptable.

If it is a battery-powered embedded device making decisions in real time, Model B may be much better.

Therefore AI engineering involves trade-offs:

$$Accuracy \leftrightarrow Speed \leftrightarrow Memory \leftrightarrow Energy \leftrightarrow Cost$$

This connects directly to your lecturer's discussion of **efficiency**.

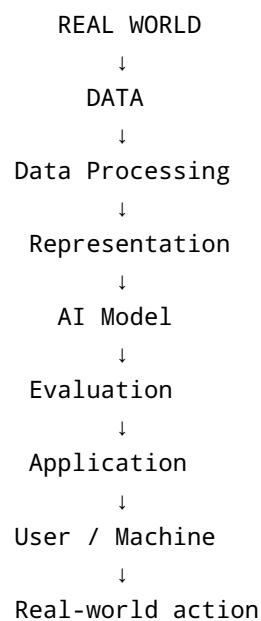
MODULE 15: THE COMPLETE AI SYSTEM

At this point, the individual topics can be connected.

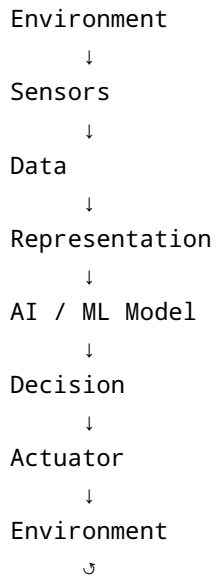
An AI system is not simply:

Model

A practical system may be:



For a robot:



The feedback loop is essential.

MODULE 16: THE AI SKILL QUESTION

Your lecturer asks:

“What wages/skills do these advancements bring about?”

The important point is that AI changes the skills required across many areas of computing.

A student entering AI should develop several layers of competence.

Foundation

- programming;
- mathematics;
- statistics;
- databases;
- algorithms.

Machine Learning

- supervised learning;
- unsupervised learning;
- model evaluation;
- feature engineering;
- optimization.

Deep Learning

- neural networks;

- backpropagation;
- CNNs;
- embeddings;
- Transformers.

AI Application Development

- APIs;
- Python;
- data pipelines;
- model integration;
- deployment;
- testing.

Modern Generative AI

- LLMs;
- prompting;
- embeddings;
- retrieval;
- evaluation;
- agents;
- multimodal systems.

Systems

- cloud computing;
- GPUs;
- distributed systems;
- embedded computing;
- robotics.

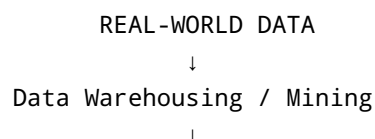
The strongest AI professional is therefore not someone who knows only how to ask an AI chatbot questions.

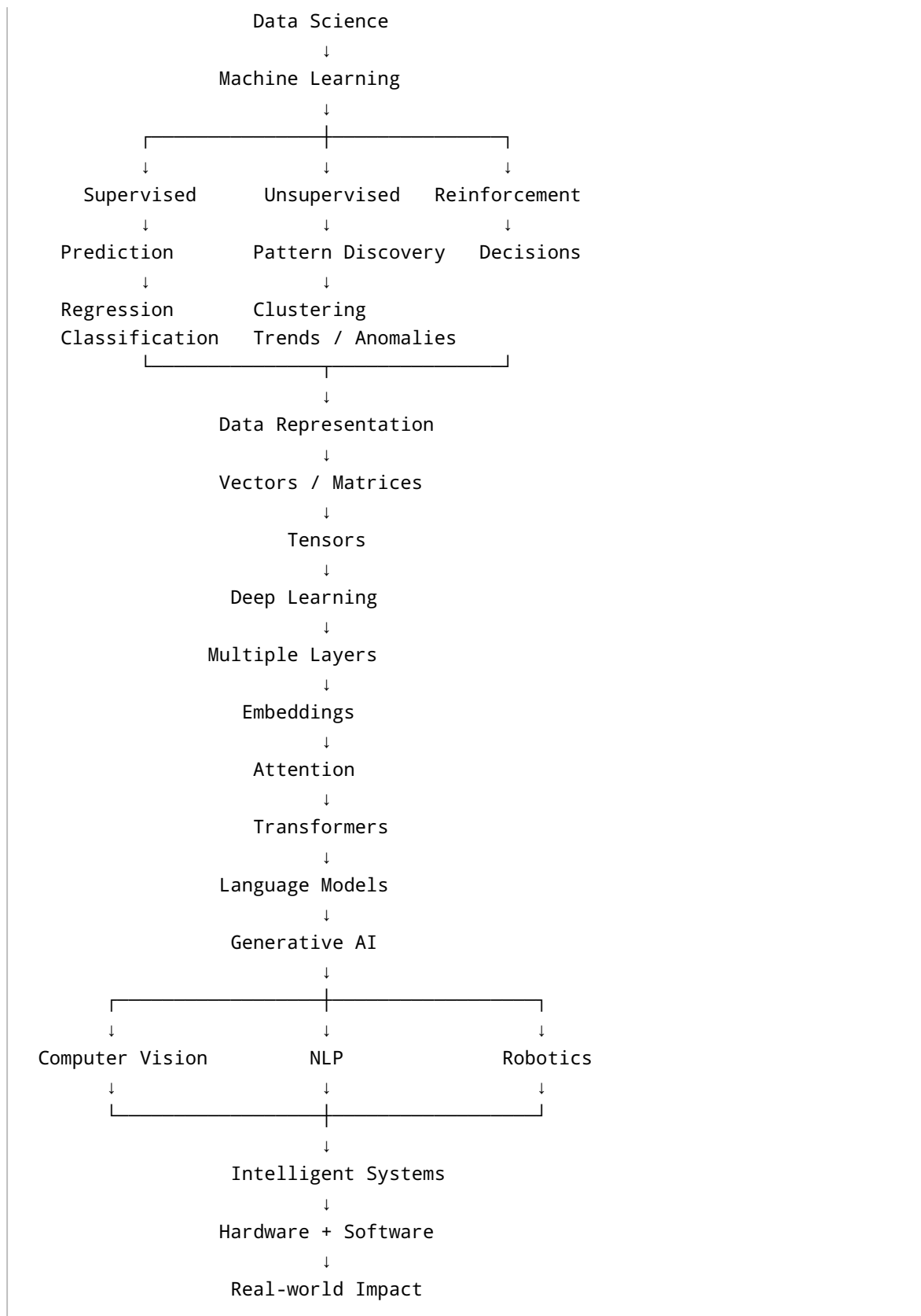
It is someone who understands:

Problem + Data + Mathematics + Algorithm + Programming + Model + Evaluation + Deployment

MODULE 17: THE BIG PICTURE — RECONSTRUCTING THE LECTURER'S FLOW

The entire lecture can now be understood as one connected story.





The surrounding questions are:

Is the data good?
↓
Is the representation appropriate?
↓
Is the model suitable?
↓
Does it generalize?
↓
How do we measure quality?
↓
Is it efficient?
↓
Can it be deployed?
↓
Can it fail?
↓
How do we verify it?
↓
What happens when humans depend on it?

This is the deeper structure behind the handwritten lecture.

MODULE 18: THE MOST IMPORTANT CONCEPTUAL DISTINCTIONS

AI vs Machine Learning

Artificial Intelligence is the broader field.

Machine learning is a major approach used to construct AI systems.

$$ML \subset AI$$

Machine Learning vs Deep Learning

Deep learning is a subfield of machine learning based primarily on multilayer neural networks.

$$DL \subset ML$$

Deep Learning vs Generative AI

Deep learning is a methodology/technical field.

Generative AI describes systems designed to generate new content.

Therefore they are not equivalent terms.

Supervised vs Unsupervised Learning

Supervised : $(X, Y) \rightarrow Model$

Unsupervised : $X \rightarrow Structure$

Regression vs Classification

Regression $\rightarrow Numerical Value$

Classification $\rightarrow Category$

Prediction vs Generation

A predictive system might answer:

"Is this image a cat?"

A generative system might produce:

"Generate an image of a cat."

Model vs System

A model is a computational component.

An AI system can include:

```
Data
+
Model
+
Software
+
Hardware
+
API
+
User Interface
+
```

Evaluation
+
Monitoring

MODULE 19: WHAT YOU SHOULD BE ABLE TO EXPLAIN AFTER THIS LECTURE

Do not simply memorize the following words.

You should be able to **connect them**:

Data
Data Science
Machine Learning
Supervised Learning
Unsupervised Learning
Regression
Classification
Clustering
Data Distribution
Skewness
Accuracy
Error
Generalization
Overfitting
Representation
Vector
Matrix
Tensor
Neural Network
Deep Learning
Embedding
Attention
Transformer
LLM
Generative AI
Computer Vision
NLP
Robotics
Automation
API
Persona
Model Evaluation
Hallucination
Efficiency

The important thing is to explain **why one concept leads to the next**.

MODULE 20: A SHORT EXAM-STYLE SYNTHESIS

If asked:

“Explain the evolution from data science to modern AI.”

A strong answer should follow this logic:

Large-scale digital data created a need for techniques capable of extracting useful information from increasingly complex datasets. Data warehousing and data mining provided mechanisms for storing, querying and discovering patterns in data. Data science expanded this process through statistics, programming, visualization and modelling. Machine learning introduced systems capable of learning statistical relationships from examples and using those relationships for prediction or classification. As datasets and computational resources grew, deep learning enabled multilayer neural networks to learn increasingly complex representations. Transformer architectures then introduced highly scalable attention-based processing for sequence and multimodal tasks, contributing to the development of modern large language models and generative AI. These technologies are now applied in computer vision, natural language processing, robotics, automation and embedded systems. Throughout this progression, however, model quality depends on appropriate data, representation, evaluation, generalization, efficiency and verification.

MODULE 21: FURTHER STUDY — MATCHED TO THIS LECTURE

A. Start with the Machine Learning Foundation

Google Machine Learning Crash Course

 [url](https://www.coursera.org/learn/machine-learning) Google Machine Learning Crash Course  turn0search0 

This is one of the best practical companions to this lecture because it progresses through linear regression, logistic regression, classification, numerical data, datasets, generalization, overfitting and neural networks. It also provides interactive exercises that can be run in the browser.

Study first:

1. Introduction to ML
 2. Linear Regression
 3. Logistic Regression
 4. Classification
 5. Numerical Data
 6. Datasets, Generalization and Overfitting
-

B. Stanford CS229 — Machine Learning

[Stanford CS229 Course Materials](#)

This is the more mathematical route.

Stanford's material covers:

- supervised learning;
- linear regression;
- generative algorithms;
- SVMs;
- learning theory;
- regularization;
- model selection;
- unsupervised learning;
- k-means;
- mixture models.

Especially useful PDF

[CS229 Lecture Notes 1 — Supervised Learning](#)

This directly reinforces the feature-vector and linear-regression discussion in these notes.

C. Stanford Linear Algebra / ML Mathematics

Stanford CS229's course structure also includes a dedicated **Linear Algebra Review**, alongside probability and machine-learning material.

[CS229 Notes and Linear Algebra Resources](#)

Use this when you reach:

$$Vector \rightarrow Matrix \rightarrow Tensor$$

rather than trying to learn all the mathematics before understanding why it is needed.

D. 3Blue1Brown — Linear Algebra

For intuition:

[3Blue1Brown — Essence of Linear Algebra](https://www.3blue1brown.com/topics/linear-algebra)

Start with the videos on:

- vectors;
- linear combinations;
- span;
- linear transformations;
- matrix multiplication;
- determinants;
- eigenvalues/eigenvectors.

The goal is not merely to calculate matrices.

The goal is to understand:

What does a matrix actually do to information?

That question becomes extremely important once we study neural networks.

E. 3Blue1Brown — Neural Networks

 But what is a neural network? — 3Blue1Brown  1youtube13 

This is one of the best bridges between your lecturer's discussion of matrices and deep learning. It explains neurons, layers, weights, biases, learning and the linear-algebra notation behind neural networks.

F. Deep Learning Textbook

 Deep Learning — Goodfellow, Bengio & Courville  0search10 

The complete textbook is available online free of charge.

Do not try to read it cover-to-cover immediately.

For this lecture, begin with:

Chapter 1 — Introduction

then:

Chapter 2 — Linear Algebra

then:

Chapter 3 — Probability and Information Theory

and later:

Chapter 6 — Deep Feedforward Networks

G. MIT Artificial Intelligence Lecture Notes

  MIT OpenCourseWare — Artificial Intelligence Lecture Notes  turn0search5 

This is particularly useful because the course contains dedicated material for:

- Introduction;
- Learning Introduction;
- Machine Learning I;
- Machine Learning II;
- Machine Learning III;
- Machine Learning IV;
- Language Understanding.

This makes it a good **second explanation** when your lecturer's discussion becomes difficult.

H. Transformers — Read the Original Paper

  Attention Is All You Need — original Transformer paper  turn0academia84 

This is the foundational Transformer paper from 2017.

The central idea is attention-based sequence processing, with the paper emphasizing the architecture's parallelizability during training.

Do not attempt to memorize the entire paper.

Initially understand:

$$Query \rightarrow Key \rightarrow Value \rightarrow Attention$$

and then:

$$Attention \rightarrow Transformer\ Layer \rightarrow Representation$$

I. Transformer Video — 3Blue1Brown

  Attention in transformers, step-by-step — 3Blue1Brown  turn0youtube85 

This is particularly valuable because it visually explains:

- embeddings;
- attention;
- masking;
- context;
- values;
- multiple attention heads;
- cross-attention;
- Transformer layers.

This should be watched **after** the neural-network video and basic linear algebra.

J. BERT — Understanding the Transformer Era

Once you understand the Transformer architecture:

  BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding  1  15 

BERT is useful historically because it demonstrates how Transformer representations could be pretrained from large amounts of unlabeled text and subsequently adapted to multiple NLP tasks.

K. Modern AI Landscape

  Stanford AI Index 2026 Report  0  4 

Use this later—not as your first AI textbook.

It gives current information on:

- technical progress;
- frontier models;
- open-source ecosystems;
- AI infrastructure;
- responsible AI;
- economics;
- science;
- medicine;
- education;
- policy;
- public opinion.

The 2026 report is especially useful for connecting the technical material in this lecture to the wider AI ecosystem.

MODULE 22: THE STUDY PATH I RECOMMEND FOR THESE NOTES

Do **not** jump immediately into LLMs.

Follow this sequence:

WEEK / STAGE 1
AI Concepts
↓
Data Science
↓
Machine Learning

STAGE 2
Supervised Learning
↓
Regression
↓
Classification
↓
Evaluation

STAGE 3
Unsupervised Learning
↓
Clustering
↓
Patterns
↓
Dimensionality Reduction

STAGE 4
Mathematics
↓
Scalars
↓
Vectors
↓
Matrices
↓
Matrix Operations
↓
Tensors

STAGE 5

Neural Networks



Layers



Weights / Biases



Activation Functions



Loss



Gradient Descent



Backpropagation

STAGE 6

Deep Learning



CNNs / Sequence Models



Embeddings



Attention



Transformers

STAGE 7

Modern AI



LLMs



Generative AI



Multimodal AI



AI Agents

STAGE 8

AI Engineering



APIs



Applications



Deployment



Hardware

↓
Robotics
↓
Evaluation
↓
Safety / Reliability

FINAL TAKEAWAY

The most important thing from this introductory lecture is **not** the individual definitions.

The important thing is to understand the movement of the field:

We first stored data.

Then:

We learned to analyze and mine it.

Then:

We learned to build statistical models from it.

Then:

We learned representations automatically using deep neural networks.

Then:

Transformers gave us highly scalable attention-based architectures.

Then:

Large models began generating language and other forms of content.

And now:

AI models are becoming components of complete intelligent systems.

But throughout the entire progression, the same fundamental questions remain:

What data are we using?

How is the data represented?

What algorithm/model are we using?

What exactly is the model learning?

How do we know that it is correct?

Does it generalize?

How efficient is it?

What happens when it fails?

How do we integrate it into a real system?

Can humans trust its output?

That is the conceptual foundation of Artificial Intelligence that your lecturer is beginning to establish.