PATTERN RECOGNITION AND MACHINE LEARNING

Lecture 01 – Overview: Learning from Examples

Chizhi Chris ZHANG

zhangchizhi@ciomp.ac.cn

Advanced Computing and Digital Technology Research Center

University of Chinese Academy of Sciences

Fall 2026

Which digits do you see?

Try it

Read the four handwritten digits. Which one is hardest?

The difficulty

The same digit can have different shapes. Different digits can look similar.

Our question. How could a computer make this distinction?

The computer receives measurements

An image is an array

A 28×28 grayscale image contains 784 pixel values.

$$\underbrace{\mathbf{x} = (x_1, \ldots, x_{784})}_{\text{pixel values}} \quad \longmapsto \quad \underbrace{\hat{y} \in \{0, 1, \ldots, 9\}}_{\text{predicted digit}}$$

Two different objects

The input is an image. The output is a class label.

The image size here is an illustrative data format.

What is pattern recognition?

Pattern

An observable description of an object or event.

Pattern recognition

Use regularities in observations to identify categories or structure.

Examples

A waveform $\rightarrow$ a spoken word. An image $\rightarrow$ a digit.

Keep the distinction. The physical object, its measurement, and its class are not the same thing.

What does it mean for a machine to learn?

Machine learning

Improve performance on a task by using data or experience.

Task	Recognize handwritten digits
Experience	Labeled examples from many writers
Performance	Accuracy on new, held-out images

Learning is not storage. Remembering the training images is not enough.

We write the procedure; data shape the rule

Rule-based programming	Machine learning
People specify the decision rule.	People choose the model and how it learns.
The rule is applied to new inputs.	Examples determine fitted model parameters.
Example: check a date format.	Example: recognize messy handwriting.

Still a designed system

Data, model, objective, and evaluation all need human choices.

Not every useful program needs learning

Problem	A sensible starting point	Why?
Sort a list of numbers	A sorting algorithm	The rule is exact
Check a password length	A direct condition	The requirement is explicit
Read handwritten digits	A learned recognizer	Appearance varies
Predict tomorrow's demand	A learned predictor	The relationship is uncertain

A practical question

Is the rule known, or must a useful relationship be estimated from examples?

How the neighboring fields connect

Field	Typical question
Artificial intelligence	How can a system perform tasks requiring intelligence?
Machine learning	How can experience improve task performance?
Pattern recognition	How can observations reveal categories or structure?
Statistics	What can data tell us under explicit assumptions?
Computer vision	What can be inferred from images and videos?

These fields overlap; they are not interchangeable names.

What you should be able to do

Describe a problem

Name the input, target, and available examples.

Explain a method

Say what it learns and how it predicts.

Run a small experiment

Build a baseline, test it, and explain a mistake.

Judge the evidence. Separate training success from performance on new cases.

Course roadmap

Part of the course	Main content
Foundations	Concepts, development environment, statistical learning
Prediction methods	Generative and discriminative classifiers, linear models, support vector machines
Representations	Feature engineering, clustering, dimensionality reduction
Further learning methods	Semi-supervised learning, ensembles, deep learning

Schedule

Meetings 1–18: teaching. Meeting 19: review. Meeting 20: open-book examination.

Understanding matters more than copying a formula

A small example

A classifier predicts the class with the nearest average weight.

Mathematics	What is an average? What is a distance?
Programming	Which examples are used to compute them?
Interpretation	Why might a heavy tomato be misclassified?

Open-book practice

Explain the prediction, identify a limitation, and propose a test.

Six questions for this lecture

Course introduction	What are we trying to learn?
Basic concepts	What are examples, features, and models?
A short history	How did the main ideas develop?
Method families	What different assumptions can we make?
Learning systems	How do we build and test a complete system?
Applications	How do these ideas transfer to new tasks?

First stop

Build a tiny classifier that we can calculate by hand.

Six examples are enough to start

Known class	Weight in grams		
Tomato	40	50	60
Mandarin	100	110	120

Training examples

Each fruit has a measured weight and a known class.

The task

Predict the class of a new fruit from its weight.

Illustrative data for hand calculation, not a biological rule.

One sample is one observed case

Sample

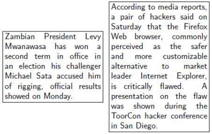
One unit of observation chosen for the task.

Examples

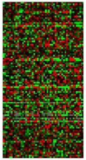
One fruit, one face image, one gene-expression profile, or one recorded signal window.

Specify the unit. Ten photographs of one fruit are ten images, but not ten independent fruits.

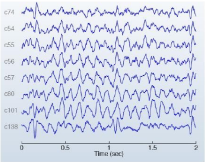
face images



documents



gene expression data



MEG readings

A feature is a measurable property

Sample	Weight / g	Red-pixel fraction
A tomato	50	0.90
A mandarin	110	0.30

$$\mathbf{x} = \begin{bmatrix} \text{weight} \\ \text{red-pixel fraction} \end{bmatrix}$$

Feature vector

An ordered collection of measurements for one sample.

Illustrative measurements; the color measurement needs a fixed imaging rule.

The label is not the prediction

Sample	Input	True label y	Prediction $\hat{y}$
A heavy tomato	85 g	Tomato	Mandarin

Label

The target value attached to an example.

Prediction

The model's estimated output for that example.

An error

Here $\hat{y} \neq y$. A valid output can still be wrong.

A dataset has rows, columns, and targets

$$X = \begin{bmatrix} 40 & 0.8 \\ 50 & 0.9 \\ 60 & 0.7 \\ 100 & 0.2 \\ 110 & 0.3 \\ 120 & 0.1 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} T \\ T \\ T \\ M \\ M \\ M \end{bmatrix}$$

$n = 6$	Number of samples
$d = 2$	Number of input features
$X \in \mathbb{R}^{n \times d}$	One sample per row

T: tomato. M: mandarin. Values are illustrative.

Classification and regression predict different targets

Task	Target	Example
Classification	A category	Tomato or mandarin
Regression	A numerical quantity	Weight in grams

Same object, different task

A fruit image can support either task, depending on the target.

A numeric code is not a quantity. Class codes 0 and 1 do not make classification a regression task.

Supervised learning uses examples with targets

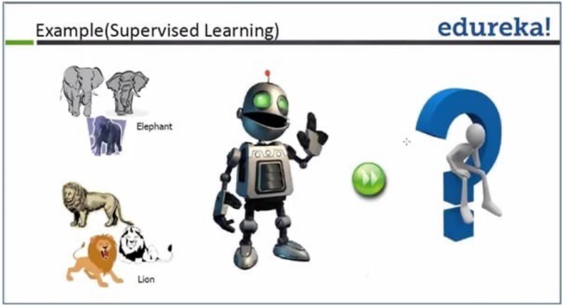
During training

The learner receives inputs together with their target labels.

For a new example

The target is unknown. The fitted model must predict it.

Supervised does not mean watched live. The supervision is the target information in the training examples.



Unsupervised learning looks for structure

Customer	Visits per month	Average spend
A	2	20
B	3	22
C	15	80
D	16	75

No target column

We ask whether the observed customers form useful groups.

A possible structure

Low-frequency / low-spend and high-frequency / high-spend groups.

Illustrative data. A discovered group is not automatically a meaningful business category.

Labels are useful, but they can be expensive

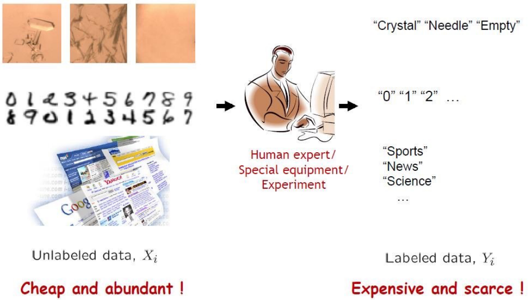
Semi-supervised learning

Use a small labeled set together with additional unlabeled examples.

A handwriting project

1,000 labeled images and 100,000 unlabeled images.

An assumption is required. Unlabeled structure must be relevant to the prediction task.



Reinforcement learning uses consequences of actions

State	A game position
Action	Choose a legal move
Feedback	Rewards during or after play
Goal	Improve expected long-term return

A different learning signal

The system is not necessarily given the best move for every position.

Delayed feedback

A move may matter because of what happens several turns later.

Identify the task before naming the learning type

Available information and goal	Learning setting
Images plus digit labels; predict a digit	Supervised
Images without labels; discover groups	Unsupervised
A few labeled images plus many unlabeled ones	Semi-supervised
Choose game moves and learn from rewards	Reinforcement

Do not classify by the input alone. The same image collection can be used in different learning settings.

Learning two class averages

$$\mu_T = \frac{40+50+60}{3} = 50 \text{ g} \qquad \mu_M = \frac{100+110+120}{3} = 110 \text{ g}$$

A nearest-centroid model

Represent each class by its average weight.

What was learned?

Two values: $\mu_T = 50$ and $\mu_M = 110$.

An assumption. Nearness to a class average is useful for predicting its label.

Use the learned averages to predict

New weight	Distance to 50	Distance to 110	Prediction
75 g	25	35	Tomato
95 g	45	15	Mandarin

$$\widehat{y} = \arg\min_{c \in \{T, M\}} |x - \mu_c|$$

Read the formula

Choose the class whose learned average is closest to the new weight.

Model, parameters, and learning algorithm

Term	In our fruit example
Model family	Predict the class with the nearest class center
Parameters	The two centers: μ_T and μ_M
Learning algorithm	Compute each center from its labeled training samples
Fitted model	The prediction rule using centers 50 and 110

The decision boundary is at 80 grams

$$|x - 50| = |x - 110| \implies x = 80$$

$$\hat{y}(x) = \begin{cases} \text{Tomato}, & x < 80, \\ \text{Mandarin}, & x \geq 80. \end{cases}$$

Decision boundary

The location where the predicted class changes.

Tie convention

At exactly 80 g, this implementation chooses mandarin.

Training and prediction are different operations

Training	Prediction
Use labeled training examples.	Receive a new input.
Estimate centers 50 and 110.	Compare distances to the fixed centers.
Change the model parameters.	Produce a predicted class.

No hidden retraining. Adding a new prediction to the training set is a separate choice.

Does the rule work on new fruit?

Weight	True class	Prediction	Correct?
75 g	Tomato	Tomato	Yes
95 g	Mandarin	Mandarin	Yes
85 g	Tomato	Mandarin	No
115 g	Mandarin	Mandarin	Yes

$$\text{Test accuracy} = \frac{3}{4} = 75\%$$

Generalization

Performance on unseen cases, not recall of the training examples.

Training, validation, and test data have separate jobs

Training set	Fit the class centers or other model parameters.
Validation set	Choose between candidate methods or settings.
Test set	Estimate performance after those choices are fixed.

Example choice

Weight only, or weight plus color? Compare using validation data.

Repeated tuning on the test set

The test set then becomes part of model selection.

Underfitting: the model cannot follow the pattern

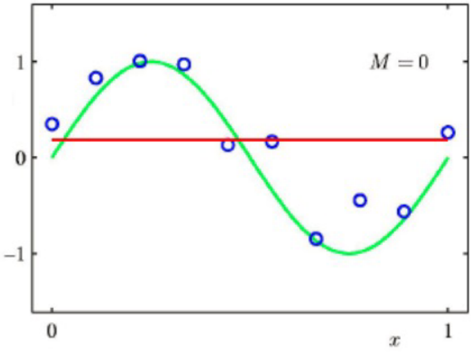
Read the figure

Blue circles: observed data.
Green curve: underlying signal in this illustration.
Red line: a constant fitted model.

Underfitting. The model is too restrictive to capture the main variation.

In the fruit task

Always predict one class, regardless of the measured weight.



Overfitting: following the sample too closely

Read the figure

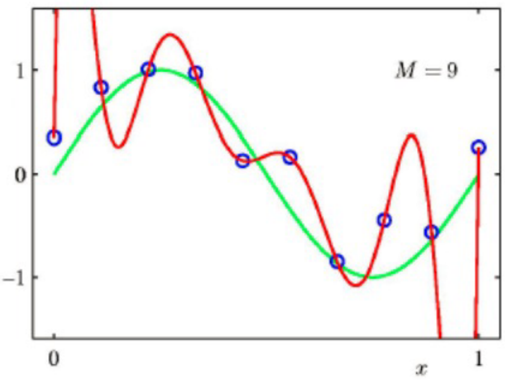
The red curve closely follows the observed blue points. Its large oscillations depart from the green signal.

Overfitting

The fitted model captures sample-specific noise as if it were a general rule.

The test

Compare behavior on new data, not only fit on the observed points.



A loss makes a mistake measurable

$$\ell(y, \widehat{y}) = \begin{cases} 0, & \widehat{y} = y, \\ 1, & \widehat{y} \neq y. \end{cases}$$

On our four test fruits

The losses are $0, 0, 1, 0$.

$$\text{Average loss} = \frac{0+0+1+0}{4} = 0.25$$

Different targets need different losses

For weight prediction, the size of the numerical error also matters.

A class label and a probability answer different questions

Hard prediction	Mandarin
Probability prediction	$P(Y = \text{Mandarin} \mid \mathbf{x}) = 0.65$

Different information

A probability prediction expresses uncertainty between possible labels.

Our centroid model

Distance is not automatically a calibrated probability.

The value 0.65 is illustrative, not computed from our two class centers.

The entire classifier fits in a few lines


```python
tomato = [40, 50, 60]
mandarin = [100, 110, 120]
center_t = sum(tomato) / len(tomato)
center_m = sum(mandarin) / len(mandarin)

def predict(weight):
    if abs(weight - center_t) < abs(weight - center_m):
        return "Tomato"
    return "Mandarin"

print([predict(w) for w in [75, 95, 85, 115]])
```

Output

Tomato, Mandarin, Mandarin, Mandarin.

Explain the fruit classifier in four sentences

Input

One measured fruit weight.

Learning

Estimate one average weight for each labeled class.

Prediction

Choose the nearer class center; break ties toward mandarin.

Limitation. Heavy tomatoes and light mandarins can overlap in weight.

Several ideas developed alongside one another

Tradition	Central idea
Statistical recognition	Describe variation and uncertainty with distributions
Structural recognition	Describe objects through parts and relationships
Neural learning	Adjust connected processing units from examples
Modern large-scale learning	Learn reusable representations from extensive data

Not a replacement ladder. Older ideas remain useful and often reappear inside newer systems.

Structural recognition describes parts and relations

A toy character description

A vertical stroke, a horizontal stroke, and their relative positions.

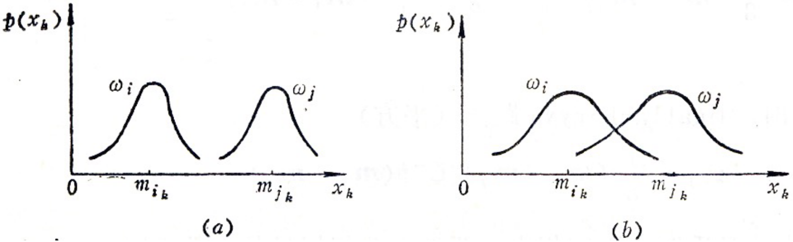
Same components	Different relationships
One vertical + one horizontal	Crossing near the center
One vertical + one horizontal	Meeting at an endpoint

Useful when structure matters

Characters, diagrams, and arrangements of components.

A practical difficulty. Noisy measurements can make reliable parts hard to extract.

Statistical recognition accepts overlapping classes



Read the two panels. Left: the class distributions are well separated. Right: they overlap.

A useful change in viewpoint

Describe how likely an observation is under each class.

Overlap matters. A single measurement may not determine the class without uncertainty.

The perceptron learns a linear decision rule

$$z = w_1 x_1 + w_2 x_2 + b, \qquad \widehat{y} = \begin{cases} 1, & z \geq 0, \\ 0, & z < 0. \end{cases}$$

Perceptron, late 1950s

Use labeled examples to adjust the weights of a linear classifier.

A simple interpretation

The weights control how each input contributes to the decision.

A boundary on its ability. A single linear boundary cannot solve every classification pattern.

Four points expose the limit of a straight line

x_1	x_2	Exclusive-or target
0	0	0
0	1	1
1	0	1
1	1	0

Exclusive-or

The target is 1 when exactly one input is 1.

The limitation

No single straight line separates the two classes in the original two-dimensional space.

Possible response

Change the representation or use a nonlinear model.

Backpropagation connects errors to hidden weights

A feedforward network

Inputs pass through a hidden layer before producing an output.

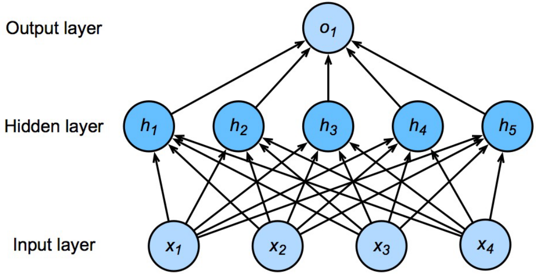
The training problem

How should weights inside the hidden layer change when the output is wrong?

Backpropagation

Compute loss gradients through the layers using the chain rule.

The 1986 work helped popularize learning useful hidden representations.



Kernel methods changed the representation

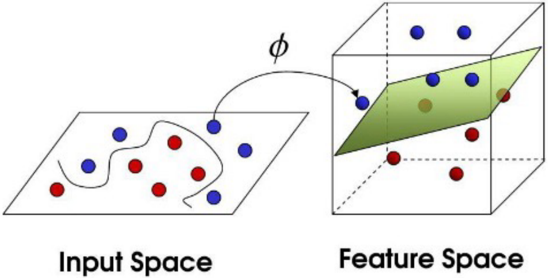
The geometric idea

Map inputs to a feature space where a simpler boundary may work.

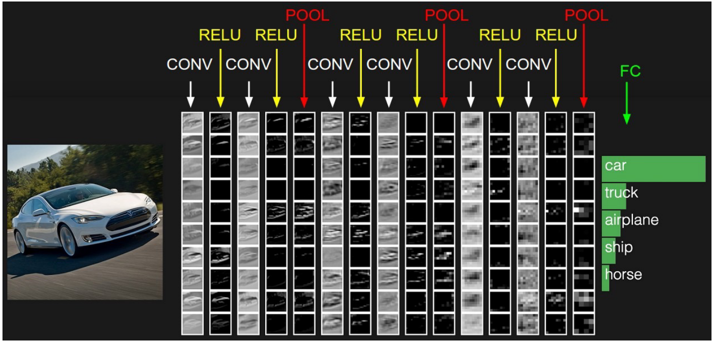
The kernel idea

Compute suitable feature-space similarities without always constructing every mapped coordinate explicitly.

Still a modeling choice. The chosen similarity encodes assumptions about the data.



Deep learning made feature learning practical at scale



ImageNet and AlexNet, 2012

Large-scale visual learning brought data, computing, and trainable representations together.

Image stages: convolution; rectified linear activation; pooling; fully connected layer.

AlphaGo combined learning with search

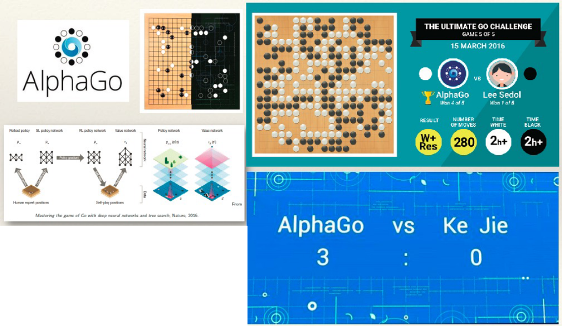
Different components, different jobs

Learn promising moves.
Estimate positions.
Search possible continuations.

Two landmark matches

Lee Sedol, 2016: 4–1.
Ke Jie, 2017: 3–0.

Not just one classifier. The system combines learned evaluation with sequential decision making.



Attention lets a representation depend on context

A language example

The animal did not cross the street because **it** was tired.

A contextual question

Which earlier words help us interpret "it"?

Transformer, 2017

Attention provides a mechanism for combining information across positions.

Not a guarantee of understanding. Useful contextual computation can still produce mistakes.

Pretraining reuses experience across many tasks

Pretraining	Learn representations or predictive structure from broad data
Adaptation	Use task-specific examples, instructions, or additional training
Evaluation	Check the target task under its actual conditions

Self-supervised example

Hide part of a text and learn to predict missing content.

The target still exists. It can be constructed from the data instead of supplied by a human annotator.

The history leaves us with three lasting questions

Representation	What information should the model use?
Learning	How can examples determine a useful rule?
Generalization	Why should the rule work on new cases?

Different methods make different commitments

Rules, distributions, distances, and networks emphasize different assumptions.

Next

Compare those assumptions using small examples.

Methods differ in what they assume

Probability models	Classes have characteristic distributions
Distance methods	Nearby examples tend to have related outputs
Linear models	Weighted combinations capture useful relationships
Decision trees	Sequences of feature tests divide the problem
Neural networks	Composed nonlinear transformations learn representations

The useful comparison

What assumption makes this method reasonable for this data?

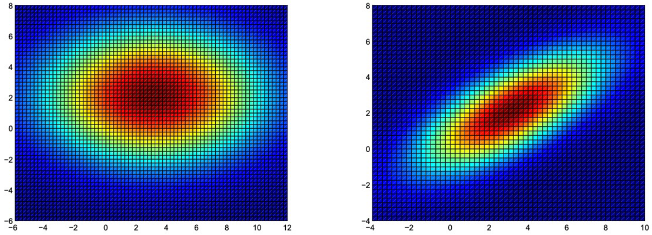
Generative and discriminative views

Generative classification	Discriminative classification
Describe each class and its frequency.	Learn the target or boundary directly.
$p(\mathbf{x} \mid y)$ and $p(y)$	$p(y \mid \mathbf{x})$ or a decision function
Then infer the class of an input.	Directly predict from the input.

Same goal

Predict the class. Different modeling routes.

A distribution describes more than an average



Read the image. The two panels show different two-dimensional density shapes. Warmer colors indicate higher density.

Information beyond the center

Spread, direction, and correlation between features.

Modeling choice. A Gaussian distribution is a useful assumption, not a universal law.

Different boundaries make different errors

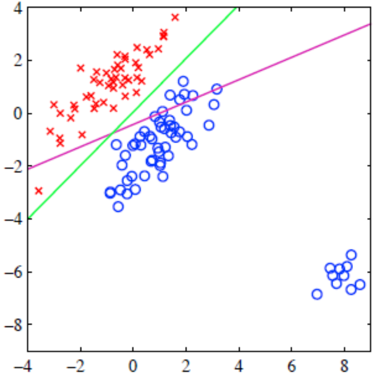
Read the scatter plot

Red crosses and blue circles represent two classes. Green and magenta lines are candidate boundaries.

Inspect the disputed region

The lines disagree about some points near the center.

Check the whole dataset. Both lines place the separate lower-right blue group on the blue side.



Nearest neighbors keep local examples

Stored weight	Distance from 78 g	Known class
60	18	Tomato
75	3	Tomato
82	4	Mandarin
100	22	Mandarin

One-nearest-neighbor rule

Use the label of the nearest stored training example. Here: tomato.

Different from a class center. A nearby example and a class average are not the same quantity.
A new illustrative training set.

A linear model combines feature contributions

$$z = 2x_1 - x_2 - 1, \qquad \hat{y} = \mathbf{1}[z \geq 0]$$

x_1	x_2	Score z	Predicted class
1	0	1	1
0	1	-2	0
2	3	0	1

Linear score

Each feature makes an additive weighted contribution.

Illustrative coordinates; class 1 is chosen at a zero score.

A new feature can simplify a nonlinear problem

Original x	New feature $z = x^2$	Target
-2	4	1
-1	1	0
1	1	0
2	4	1

In the original coordinate

A single threshold on x cannot separate both outer points from both inner points.

In the new coordinate

Predict class 1 when $z > 2.5$.

A decision tree asks a sequence of questions

Path through an illustrative tree	Prediction
Red-pixel fraction > 0.6	Tomato
Otherwise, weight < 80 g	Tomato
Otherwise, weight ≥ 80 g	Mandarin

Example

For $(85\text{ g}, 0.9)$, the first test predicts tomato.

These tests are illustrative. A trained tree chooses splits from training data; the rule is not a biological guarantee.

Regression predicts a quantity

$$\widehat{y} = 3 + 0.02x$$

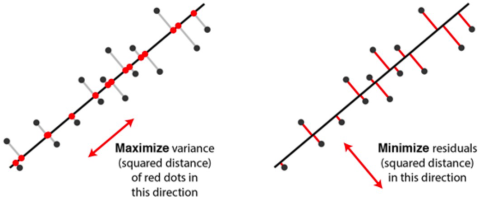
Package weight x / g	Predicted price $\widehat{y}$ / yuan
100	5
200	7
300	9

Interpretation

3 yuan is an intercept; each additional gram contributes 0.02 yuan.

A hypothetical pricing example, not a real tariff.

Dimensionality reduction keeps selected information



Read the projection. Gray points are projected onto a line; red points are their projections.

Principal component analysis

Choose directions that retain substantial variation under an orthogonal projection.

Not the same objective as classification. Large variation is not always the variation that separates labels.

A curved surface needs more than a straight projection

The Swiss roll example

Points lie on a curled two-dimensional surface inside a three-dimensional space.

Two notions of closeness

Through the surrounding space, or along the surface.

What a projection can lose. Different layers of the roll may overlap after a simple projection.

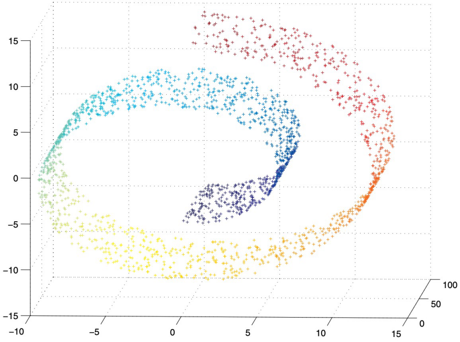


Figure 1: 2000 Random data points on the swiss roll.

Clustering depends on the meaning of a group

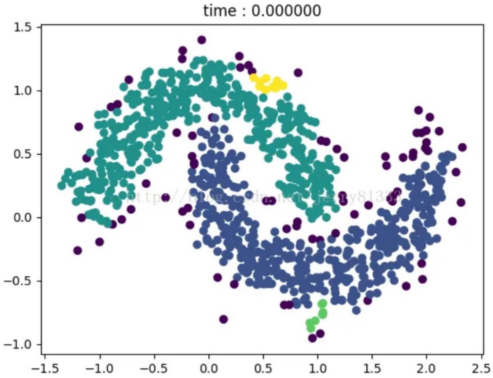
Look at the shape

The point cloud contains curved, non-spherical structure and scattered points.

Different grouping assumptions

Near a center, connected by nearby points, or dense relative to the surroundings.

Read colors carefully. Colors in an algorithm illustration are assignments, not proof of ground-truth classes.



Neural networks compose learnable transformations

$$\mathbf{h} = \sigma(W_1\mathbf{x} + \mathbf{b}_1), \qquad \widehat{\mathbf{y}} = g(W_2\mathbf{h} + \mathbf{b}_2)$$

Hidden representation

The intermediate vector **h** is computed from the input.

Why nonlinearity matters

Without nonlinear activations, composing affine layers is still an affine transformation.

More flexibility requires control. Data, optimization, and regularization affect generalization.

An ensemble combines several predictions

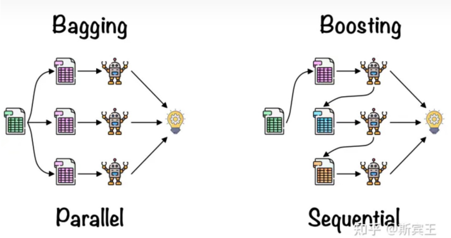
Model A	Model B	Model C	Majority vote
Tomato	Mandarin	Tomato	Tomato
Mandarin	Mandarin	Tomato	Mandarin

Why combine?

Different models may make different mistakes.

Agreement alone is not enough. Copies of one mistaken model do not add independent evidence.
Illustrative predictions from three classifiers.

Bagging and boosting organize learning differently

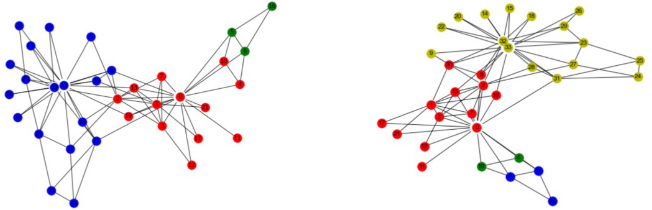


Bagging. Train learners on resampled data and combine their predictions.

Boosting. Build learners sequentially to address the current ensemble's remaining errors or loss.

The arrows show dependence. Sequential construction and independent construction are different, even when both use many models.

Graphs describe relationships between samples



Read the network. Nodes represent objects; edges represent chosen relationships. The two panels show different groupings.

A semi-supervised use

Known labels may help infer labels of strongly connected unlabeled nodes.

A condition, not a guarantee. Connected objects must share relevant information for propagation to help.

Choose a starting method from the problem

Situation	A useful starting comparison
Small labeled table	A simple linear model or tree
Meaningful similarity	A nearest-neighbor baseline
Many correlated measurements	Representation or feature selection
Complex images with suitable data	Learned visual representations

Starting points, not rankings. Validate under the same data split and task-relevant criteria.

Match the description to the method

Description	Method idea
Find the most similar stored example	Nearest neighbor
Ask a sequence of feature questions	Decision tree
Keep a few informative coordinates	Dimensionality reduction
Combine several model outputs	Ensemble learning

One more question

What would make the method fail in your example?

A recognition system is more than a classifier

Acquisition	Photograph a manufactured part
Preprocessing	Align, crop, and normalize the measurement
Representation	Compute or learn useful features
Prediction	Estimate whether a defect is present
Evaluation	Compare with reliable inspection labels
Use and monitoring	Route the part and check later performance

The system question

Where could information be lost or a wrong assumption enter?

Define the inspection task precisely

Unit

One physical part, photographed after production.

Input

A fixed-view surface image available at inspection time.

Target

Whether the visible surface meets a specified defect criterion.

Out of scope for this input. Internal damage that cannot be seen in the photograph.

Collect examples from the conditions that matter

Variation	Narrow collection	Better coverage
Lighting	One bright setting	Expected lighting range
Production	One batch	Several relevant batches
Appearance	Only obvious defects	Subtle and borderline cases
Equipment	One camera state	Expected equipment variation

More copies are not more coverage. Repeated images of the same easy case leave important gaps.

A labeling rule must survive disagreement

Ambiguous rule	“Mark a bad-looking surface.”
Operational rule	Define defect types, inspection regions, and borderline cases.
Quality check	Label the same sample independently, then review differences.

Separate uncertainty

Record “needs review” when the available evidence cannot support a reliable label.

Split by the unit that must generalize

Image	Physical part	Assigned split
A-front	A	Training
A-side	A	Training
B-front	B	Test
B-side	B	Test

Goal

Recognize new physical parts, not another photograph of a familiar part.

Grouped split

Keep photographs of the same part together. Consider batch or time splits when required by deployment.

Fit preprocessing on training data only

Training measurements	50, 60, 70
Training mean	$\mu_{\text{train}} = 60$
New measurement	100
Centered new value	$100 - 60 = 40$

Do not refit on all data

Including 100 changes the mean to 70 and lets test information affect preprocessing.

Reuse the fitted transformation

Apply the training mean consistently during validation, testing, and use.

Features must be available when the prediction is made

Candidate feature	Available at inspection?	Use as input?
Surface texture	Yes	Potentially
Part dimensions	Yes, if measured	Potentially
Repair outcome next week	No	No
Folder name “defective”	Encodes the answer	No

Availability test

Could this value be known for the next unlabeled part at the moment of prediction?

Training loss is not the only curve to watch

Training stage	Training loss	Validation loss
1	0.40	0.42
2	0.20	0.25
3	0.10	0.23
4	0.03	0.31

What the example suggests

Later training continues to fit the training set, while validation performance worsens.

Choose using validation. Do not use the final test set to select the stopping point.
Illustrative losses.

A simple baseline can expose a misleading score

Normal parts	90
Defective parts	10
Total	100

Always predict "normal"

Accuracy $= 90/100 = 90\%$.

What it misses

All 10 defective parts.

Baseline purpose

Give a meaningful reference before celebrating a model's score.

A confusion matrix separates four outcomes

	Predicted defect	Predicted normal	Total
Actual defect	8	2	10
Actual normal	9	81	90
Total	17	83	100

Correct detections

8 defects found; 81 normal parts passed.

Two different mistakes

2 missed defects; 9 false alarms.

Accuracy, precision, and recall answer different questions

Metric	Question	Value
Accuracy	How many parts were classified correctly?	$89/100 = 89\%$
Precision	Among alarms, how many were true defects?	$8/17 \approx 47.1\%$
Recall	Among defects, how many were found?	$8/10 = 80\%$

Positive class

Defect. Always state which class the metric refers to.

The cheaper policy can have lower accuracy

Illustrative costs

One missed defect: 500 yuan. One false alarm: 10 yuan.

Policy	Missed defects	False alarms	Accuracy	Total cost
A	2	9	89%	1,090 yuan
B	1	20	79%	700 yuan

$$C_A = 2(500) + 9(10) = 1090, \qquad C_B = 1(500) + 20(10) = 700$$

Conditional conclusion

B is cheaper under these assumed costs and this evaluation sample.

A threshold changes which cases trigger an alarm

$$\text{Alarm if } p(\text{defect} \mid \mathbf{x}) \geq t$$

Threshold change	Typical effect on one fixed score set	Trade-off
Lower t	More alarms	Fewer misses, more false alarms
Higher t	Fewer alarms	More misses, fewer false alarms

Select before final testing. Use validation data and the intended costs or constraints. The model's scores must support the probability interpretation shown here.

Leakage lets information cross the wrong boundary

Leak	Why the result is misleading
Same part appears in train and test	Familiarity is mistaken for generalization
Future repair result enters the input	Information is unavailable at prediction time
Test labels guide feature selection	The final evaluation influences the model

Leakage is about information use

Renaming a file or dataset does not repair the experimental design.

A new camera can change the input distribution

Condition	Training	Later use
Lighting	Diffuse white light	Strong directional light
Camera setting	Fixed exposure	Different exposure
Surface	One finish	A new finish

Distribution shift

The data encountered in use differ from the training conditions.

Investigate before retraining

Inspect inputs, acquisition changes, label definitions, and error patterns.

A correct model can still be too slow

Incoming rate	100 parts / second
Time per prediction, one serial worker	0.05 seconds
Processing capacity	$1/0.05 = 20$ parts / second
Backlog growth	$100 - 20 = 80$ parts / second

System constraint

A growing queue prevents timely inspection even if individual predictions are accurate.

Illustrative rates; acquisition and communication overhead are omitted.

Monitoring needs feedback beyond the alarms

Observe inputs

Missing images, changed brightness, new surface types.

Observe operation

Alarm frequency, latency, queue length.

Observe outcomes

Review labeled samples from both alarmed and passed parts.

Selection bias

If only alarms are checked, missed defects among passed parts can remain invisible.

Diagnose the experiment before changing the model

A project report

"99% test accuracy on 10,000 images from 100 parts."

Question	Why it matters
Were parts kept in separate splits?	Repeated views can leak object identity
How many defects were included?	Overall accuracy can hide missed defects
Were later batches tested?	The result may not cover future conditions

A sound next step

Inspect the split and error counts; then test on genuinely held-out parts.

Read an application through its input and output

Application	Input	Output
Crowd analysis	An image	People count
Traffic analysis	Video frames	Locations and tracks
Speech recognition	Audio waveform	Word sequence
Recommendation	User and item information	Ranked items
Scientific prediction	Measurements or sequences	A predicted property or structure

A useful first description

Name what is observed and what must be inferred.

Crowd counting is not identity recognition

Read the image

Green markers indicate annotated head locations in a crowd.

A counting target

Estimate how many people are present, without identifying each person.

Practical difficulties. Occlusion, perspective, and different apparent head sizes.

Detection, counting, and tracking are different tasks

Read the annotations. Colored points and a category summary indicate people and vehicle types in one scene.

Detection	Where are the objects?
Counting	How many objects are present?
Tracking	Which object is the same one across frames?

One image cannot establish a track. Tracking needs information across time.

Perception is only one part of autonomous behavior

Perception	Identify road users, lanes, and obstacles
Prediction	Estimate how other road users may move
Planning	Choose a possible course of action
Control	Convert the plan into physical commands

A category is insufficient. Recognizing “pedestrian” does not alone specify a safe action.

One photograph can define three vision tasks

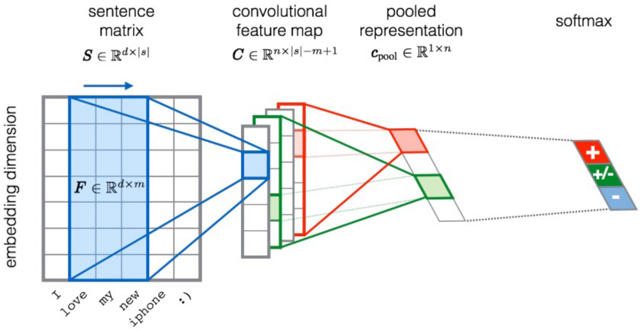
Classification	Is there a cat?
Localization	Where is the cat?
Segmentation	Which pixels belong to the cat?

Different targets

A class label, a region, or a pixel-level mask.

Different annotation effort. Knowing the class does not supply the exact boundary of every pixel.

Text classification uses numerical representations too



Sentiment classification

Sentence matrix → local features → pooled representation → sentiment scores.

"I love it" and "I do not love it" share words, but differ in meaning.

Speech recognition predicts a sequence

Input	A time-varying audio signal
Output	A sequence of words or characters
Variation	Speaker, speed, accent, noise, and pauses
Evaluation	Substitutions, deletions, and insertions in the recognized sequence

A sequence error

Missing a short word can change the meaning of the sentence.

Recommendation predicts an ordering

Candidate item	Predicted relevance score	Rank
An introductory tutorial	0.82	1
A worked example	0.76	2
An advanced proof	0.30	3

A ranking target

Place useful items near the top for a particular context.

Exposure affects feedback. Items that are never shown cannot receive the same opportunity to be clicked. Illustrative relevance scores; not calibrated click probabilities.

Scientific prediction can narrow a search

Example: structure prediction

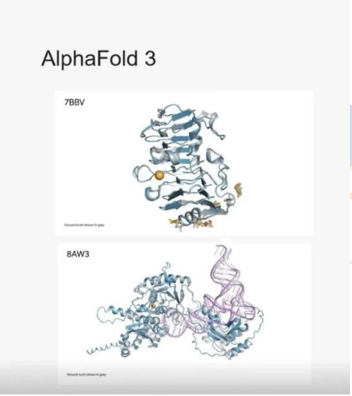
Learn relationships that help predict three-dimensional molecular arrangements.

A scientific role

Prioritize hypotheses and guide subsequent investigation.

Prediction is not experimental confirmation. A plausible structure does not establish every property or function.

The illustration shows example predicted and reference molecular structures.



A two-dimensional map is a view, not the whole dataset

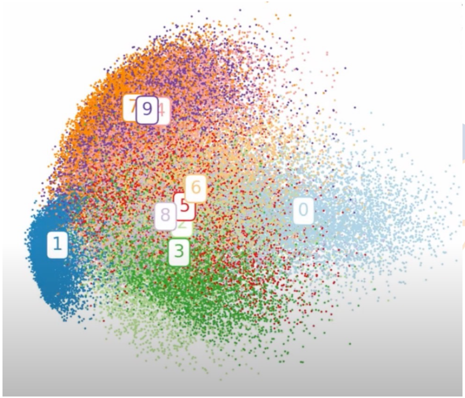
Read the map

Each point represents a handwritten image; colors indicate digit classes in this illustration.

What it can reveal

Some classes occupy relatively distinct regions; others overlap in this view.

What it cannot prove. A visually separated map does not establish test accuracy, causality, or an optimal classifier.



Collect only what the task actually needs

Goal	Question about data collection
Estimate room occupancy	Do we need identities, or only counts?
Improve speech transcription	Which audio must be retained, and why?
Study model errors	Can the necessary examples be reviewed with less personal information?

Responsible design

Limit unnecessary collection and test performance across relevant conditions.

Design a handwriting experiment for new writers

Available data

20 writers; each writes every digit twice: $20 \times 10 \times 2 = 400$ images.

Split	Writers	Images
Training	12	240
Validation	4	80
Test	4	80

A defensible plan

Keep writers separate, fit preprocessing on training data, compare a simple baseline, and inspect digit confusions.

Target population

This split tests new writers within the collected setting; it does not cover every future condition.

Four ideas to remember precisely

A sample is not its label

$\mathbf{x}$ describes what is observed; y is the target.

Training is not prediction

Training estimates parameters; prediction applies the fitted model.

A good fit is not generalization

Use suitable held-out examples to estimate performance on new cases.

A model is not a complete system. Data, representation, evaluation, and operating conditions matter.

From observations to a usable dataset

Sample	Weight / g	Red-pixel fraction	Label
01	50	0.90	Tomato
02	110	0.30	Mandarin
03	Missing	0.85	Tomato

Lecture 02

Set up the environment; use NumPy and pandas; explore data.

Questions to bring

What does one row represent? Are units consistent? Which values are missing?

Before fitting a model. Make sure the data mean what you think they mean.