Yale University

Introduction to Machine Learning

S&DS 265 · Lecture 1 · Data, models, computation, and generalization

Zhuoran Yang

Fall 2026

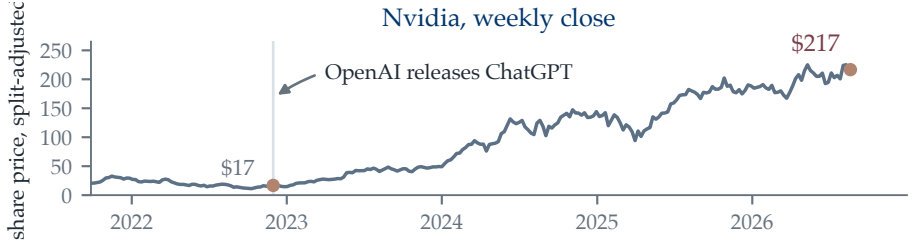
AI usage: figures were generated, and these slides polished, with the help of AI tools.

Outline

Outline

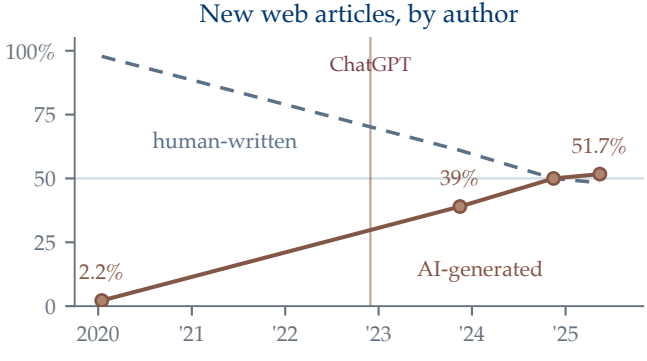
AI Is Changing the World

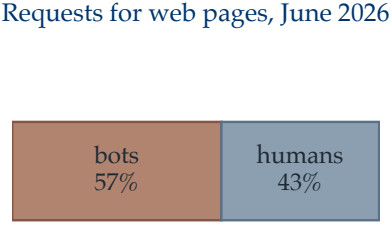
In 2024 analysts called Nvidia's developer conference (GTC) *"the Woodstock of AI."* Nvidia Stock took off after ChatGPT release.



Source: Weekly adjusted closes, Yahoo Finance, retrieved 20 August 2026.

How Much of the Web Is Machine-Made

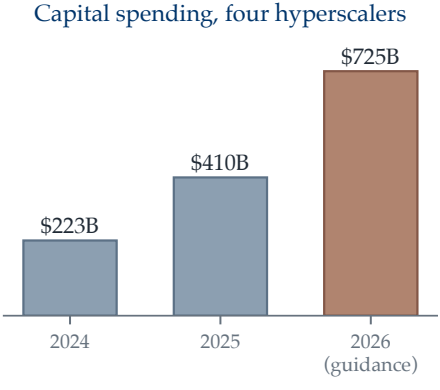


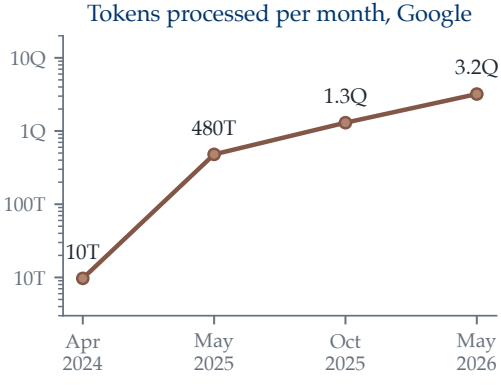


Left: of newly published English-language articles, the share written by a model passed the share written by people in November 2024, and stood at 51.7% by May 2025. Right: on Cloudflare's network, 57% of requests for web pages in June 2026 came from bots and 43% from human.

Articles: Graphite, October 2025, 43,000 Common Crawl URLs; only the four marked points are reported figures and the lines interpolate between them. Requests: Cloudflare Radar, reported 3 June 2026.

Capital Spending and Token Volume





Left: combined capital spending by the four largest cloud companies, mostly on data centres and accelerators. Right: tokens processed per month by one provider, on a logarithmic axis — from 10 trillion to 3.2 quadrillion, a factor of roughly 300 in two years.

Capex: reported company guidance, August 2026. Tokens: figures given by Google at I/O, May 2026.

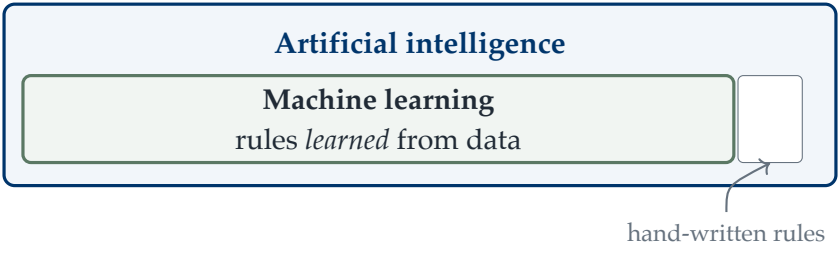
Two Questions Before Anything Else

What is artificial intelligence?

And since this is a course in machine learning: how are the two related?

Outline

Machine Learning and Artificial Intelligence



Artificial intelligence is the goal; machine learning reaches it by fitting rules to data rather than writing them down. Hand-written rules were the older route, and the thin band left to them keeps shrinking — on the problems this course cares about, almost everything is now learned. So we treat AI and machine learning as the same subject.

Machine Learning: Learning from Data

We want to find patterns in data we have observed, and turn them into a model that predicts well on data we have not seen.

$$\text{observed data} \longrightarrow \text{a model that also works on new cases}$$

That leaves four questions, and everything in this course answers one of them:

1. What data do we have?
2. What kind of model do we allow?
3. How do we build the model from the data?
4. How do we tell whether it predicts well?

Data, Model, Fitting, Evaluation

data	model	fitting	evaluation	compute
what we observed	what rules we allow	how the rule is found	does it work on new data	how it runs at scale

mostly outside this course

Compute: What We Leave Out

Compute is a fifth ingredient, and at frontier scale it dominates engineering effort. It is also a different subject:

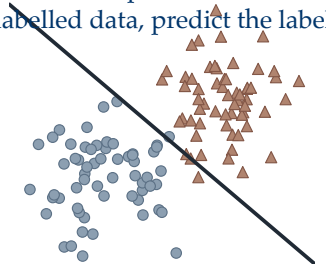
- Distributed training across thousands of accelerators;
- Memory hierarchies, activation checkpointing, and quantization;
- Communication schedules and interconnect bandwidth.

We will meet compute where it changes the statistics — minibatches, stochastic gradients, why we stop early — and leave the systems engineering to courses built for it.

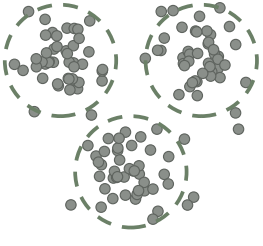
Outline

Supervised, Unsupervised, and Reinforcement Learning

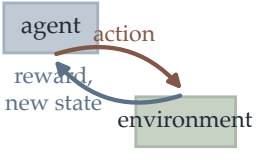
supervised
labelled data, predict the label



unsupervised
no labels, find the structure



reinforcement
act, then learn from the result



Supervised: every example carries a target, and the job is to predict it. Unsupervised: no targets at all, so the job is to find structure. Reinforcement: no fixed data set — the learner acts, and its own actions decide what it observes next.

Topics in Each Setting

Supervised

regression, classification, trees, neural networks

predict a labelled outcome, and measure whether it generalizes

Unsupervised

PCA, clustering, mixtures, autoencoders, diffusion

find structure, compress, and generate new examples

Reinforcement

value, policy, and the language-model connection

decision-making under uncertainty

Topics in Each Setting

Supervised

regression, classification, trees, neural networks

predict a labelled outcome, and measure whether it generalizes

Unsupervised

PCA, clustering, mixtures, autoencoders, diffusion

find structure, compress, and generate new examples

Reinforcement

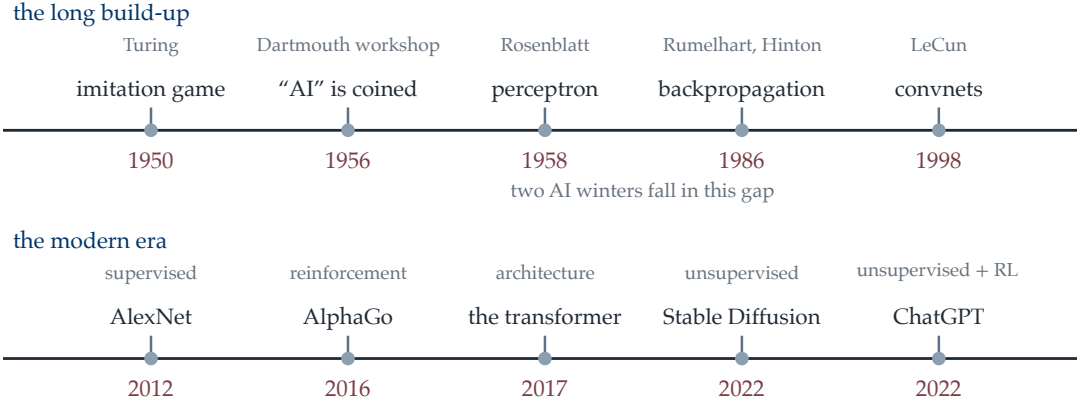
value, policy, and the language-model connection

decision-making under uncertainty

The same questions recur in all three: (i) what structure is imposed, (ii) how the fit is carried out, and (iii) what evidence would show it worked.

Outline

Timeline of Key Milestones



Selected milestones.

Reading the Timeline

- The core ideas are old. Backpropagation is from the 1980s, convolutional networks from the 1990s, and the perceptron from 1958.
- Progress was not steady. Two AI winters followed periods of overpromising, and both were survived by a small number of people who kept working.
- What changed after 2012 was mostly scale: larger data sets, faster hardware, and architectures that use both.
- In 2024 the Nobel prizes in physics and chemistry went to this work — Hopfield and Hinton for neural networks, Hassabis and Jumper for protein structure.

Outline

Reinforcement Learning: AlphaGo

AlphaGo
Mastering the ancient game of Go

Figure: Google DeepMind, AlphaGo: The Challenge Match.

4–1

AlphaGo defeated Lee Sedol in the 2016 challenge match.

Learning ingredients

- Expert games;
- Self-play;
- Reinforcement learning;
- Search at decision time.

Supervised Learning: AlphaFold

AlphaFold reveals the structure of the protein universe

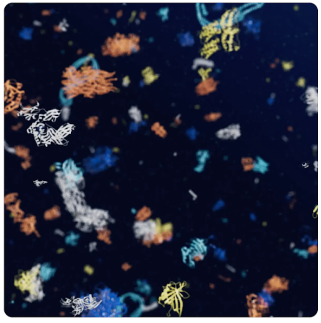


Figure: Google DeepMind, AlphaFold.

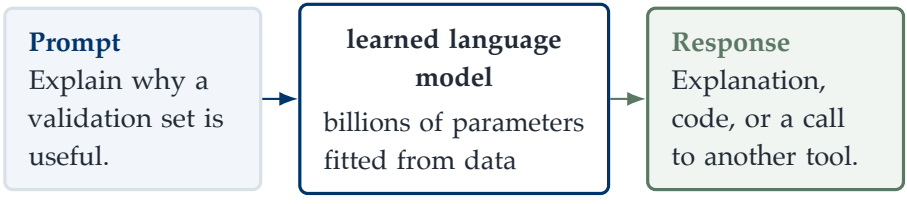
200+ million

predicted protein structures have been released by DeepMind and EMBL–EBI.

The learned map

amino-acid sequence ⟶ 3D structure

Unsupervised Learning: Language Models



A powerful predictor is still a fallible predictor

Fluency does not guarantee truth, current knowledge, or safe behavior.

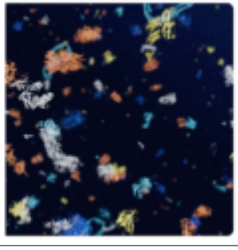
Source: OpenAI, GPT-4 (2023), for the scale-and-capability example.

Machine Learning Applications

Games

AlphaGo beats Lee Sedol, 2016
reinforcement learning

Science



AlphaFold protein structures, 2021
supervised learning

Generation

text-to-image, 2022 onward
unsupervised learning

Three results, one per setting. Games: a policy learned from self-play. Science: protein structure predicted from sequence. Generation: an image drawn from a sentence. The same recipe — data, model, fitting, evaluation — reaches all three.

AlphaGo and AlphaFold press images; text-to-image sample from Wikimedia Commons, public domain. See figures/01-ml-workflow/sources.txt.

Definition of Learning

Use observed data to build predictions or decisions that remain useful on new cases.

Outline

Motivating Example: Predicting Income

An analyst observes education, seniority, and income for 30 people. For a new person whose income is unknown, the analyst must produce a numerical prediction.

What data, model class, loss, and computation turn the observed rows into a prediction that can be evaluated on unseen people?

Example and data: ISLP §2.1 (James, Witten, Hastie, Tibshirani, and Taylor, 2023).

The Income Data

person	education	seniority	income
	years	no unit	$1,000s
1	21.59	113.10	99.92
2	18.28	119.31	92.58
3	12.07	100.69	34.68
⋮	⋮	⋮	⋮

One row is one person. Education is in years, income in thousands of dollars. Seniority level has no stated unit: it runs from 20 to 188. The data are simulated.

Source: `Income2`, ISLP Figures 2.2–2.3. Figure 2.2 gives the units for education and income; ISLP never gives one for seniority.

Reading One Row

person	education	seniority	income
1	21.59	113.10	99.92

For person 1, the feature vector is $x_1 = (21.59, 113.10)^\top$ and the observed response is $y_1 = 99.92$. The synthetic units are retained from the textbook; the statistical roles of the columns are what matter here.

Problem Setup

Training data. $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^n$ contains observed input–response pairs.

Features. $x_i \in \mathbb{R}^d$ contains measurements available at prediction time.

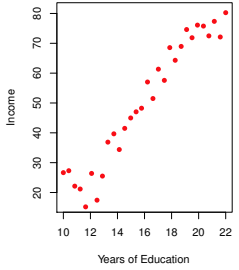
Response. $y_i \in \mathbb{R}$ is the numerical outcome.

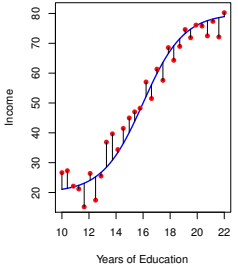
Goal. Build $\hat{f}$ so $\hat{f}(x^*)$ predicts the response at an unseen x^*.

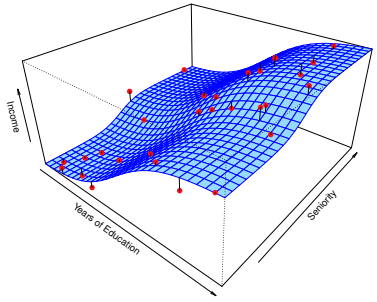
Because the response is a number rather than a category, this is a **regression** problem. Predicting a category is **classification**, and comes later in the course.

For `Income2`: $n = 30$, $d = 2$, $x_i = (\text{education}, \text{seniority})^\top$, and $y_i = \text{income}$ — years, an unlabelled index, and thousands of dollars respectively.

Income Data in Feature Space







Left: income against education alone — one fitted curve. Right: the same 30 people against both features, with the fitted surface. Each red point is one person; the vertical segment is the residual.

Figure: ISLP, Figures 2.2 and 2.3 (James et al., 2023).

Signal and Noise in Regression

A statistical description

$$Y = m(X) + \varepsilon, \qquad \mathbb{E}[\varepsilon \mid X] = 0.$$

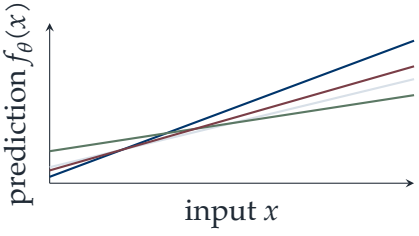
Reducible error

We can improve how well our fitted rule $\hat{f}$ approximates $m(x) = \mathbb{E}[Y \mid X = x]$.

Irreducible error

Even the conditional mean $m(X)$ cannot predict new noise ε.

The Model Class — Hypothesis



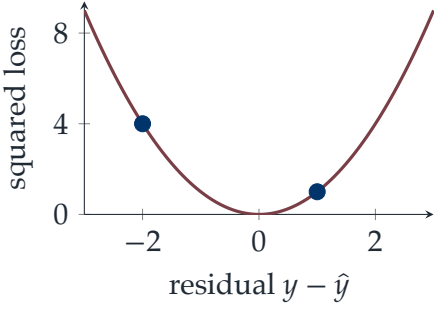
Linear family

$$f_\theta(x) = \theta_0 + \theta_1 x$$

The family fixes which patterns are possible; training picks one member of it.

A model class is an assumption: choosing straight lines asserts that a straight line is close enough to be useful. "All models are wrong, but some are useful." — George Box (1976). The question is not whether the assumption is true, but whether it is wrong in a way that matters.

Squared Loss



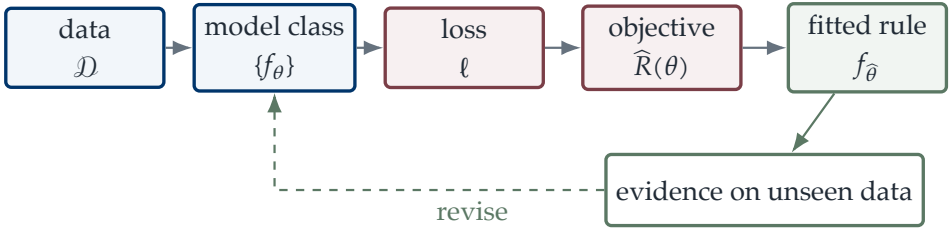
Squared loss

$$\ell(y, \hat{y}) = (y - \hat{y})^2$$

A two-unit miss costs four times as much as a one-unit miss.

The loss is a modeling choice.

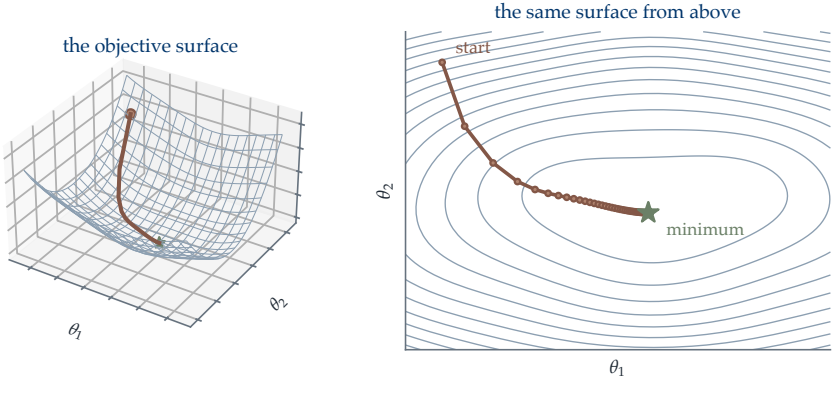
Empirical Risk Minimization



$$\hat{\theta} \in \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_\theta(x_i)) \,.$$

Source: Workflow adapted from D2L §3.1 and Stanford CS 329P, "ML Model Overview."

Optimization: Which Parameters Minimize the Objective



The objective assigns a number to every choice of parameters. Optimization starts somewhere, repeatedly steps downhill, and stops near a minimum. That is the whole question the optimizer answers.

Exact gradient descent on a declared two-parameter surface.

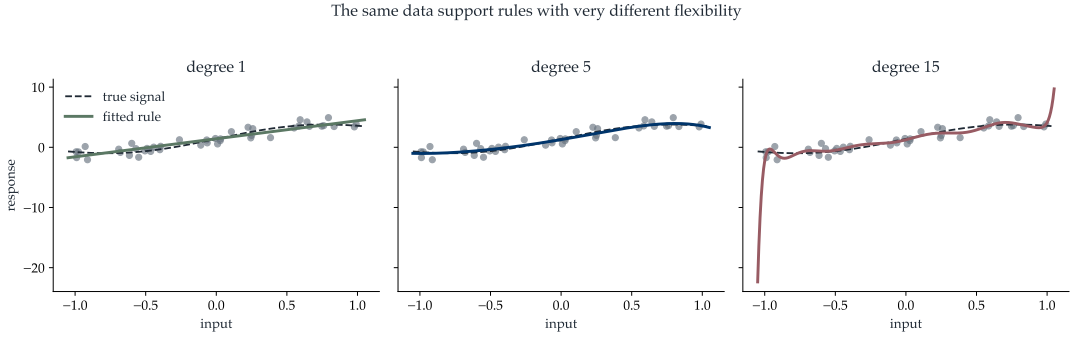
Outline

The Second Question

Optimization finds the best member of the family we chose.

It cannot tell us whether we chose the right family.

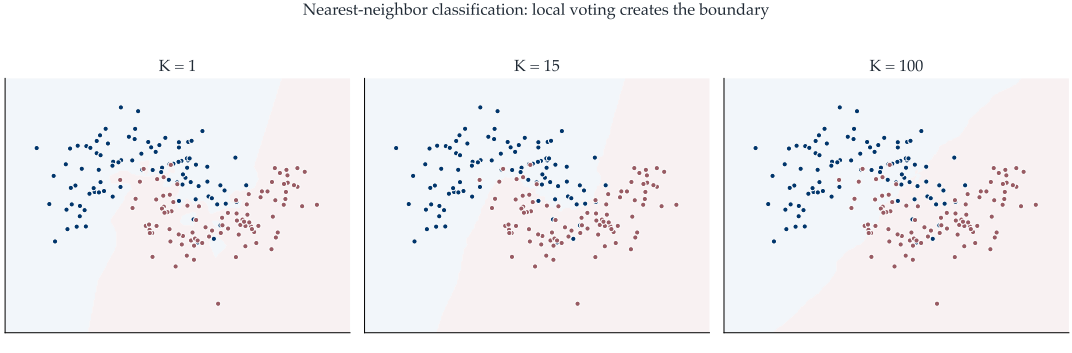
Polynomial Regression: the Same Data, Three Families



Degree 1 is too rigid to follow the curve. Degree 5 tracks the true signal. Degree 15 passes closer to the observed points than either — and invents structure at the edges that is not in the signal.

The dashed curve is the signal the data were generated from.

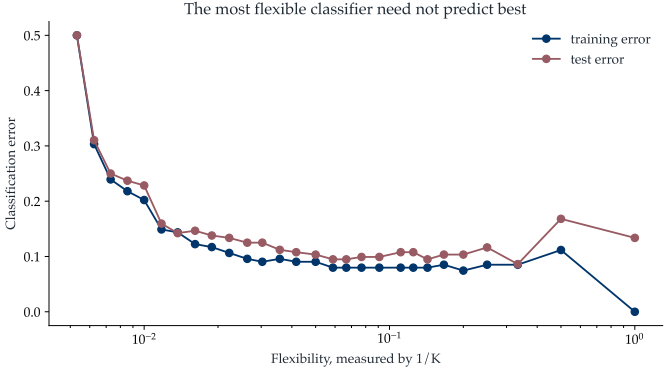
Nearest Neighbors: Flexibility Set by One Number



The same points classified by a vote among the K nearest neighbors. $K = 1$ carves out an island around every single observation. $K = 100$ averages so widely that the boundary is almost straight.

Shading shows the predicted class at every point.

Fitting the Data Is Not the Goal



Error against flexibility. Training error falls all the way to zero at $K = 1$: the most flexible rule reproduces every observation exactly. Test error turns around and rises. The best rule is neither the most rigid nor the most flexible.

The two curves come from the same fitted models on held-out points.

Overfitting

What the training error sees	What we actually want
how closely the rule reproduces the data it was fitted on	how well the rule predicts cases it has never seen

Overfitting is the gap between the two: a rule flexible enough to memorize the sample also memorizes its noise, and the noise does not repeat.

Overfitting

What the training error sees	What we actually want
how closely the rule reproduces the data it was fitted on	how well the rule predicts cases it has never seen

Overfitting is the gap between the two: a rule flexible enough to memorize the sample also memorizes its noise, and the noise does not repeat.

This is why it is mandatory to keep a held-out set for evaluation

Summary: Learning Problem

Data. $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^n$ records cases produced by a sampling and measurement process.

Model. $\mathcal{F} = \{f_\theta : \theta \in \Theta\}$ specifies which prediction rules are available.

Loss. $\ell(y, f_\theta(x))$ states which prediction errors are costly.

Training. The empirical-risk minimizer is

$$\hat{\theta} \in \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_\theta(x_i)) \,.$$

Summary: Generalization and Feedback

Training error measures fit to observed rows; test error estimates behavior on new rows.

supervised learning	feedback is an observed target y
unsupervised learning	no targets; structure is found in the inputs
reinforcement learning	feedback follows actions that alter future data

Self-supervision is a device *within* unsupervised learning: a target is manufactured from the data itself — the next word in a sentence — so that unlabelled data can be fitted by supervised machinery. Evaluation must match the population and feedback process in which the fitted system will be used.

Next Lecture

Every model in this lecture used vectors, distances, probabilities, and derivatives without yet developing their common language.

Lecture 2 develops the mathematical objects needed to check dimensions, reason about uncertainty, and follow how a loss changes with its parameters.

Reading for this lecture: ISLP Chapter 2; D2L Chapter 1.