FULCRUM COURSE NOTES · 03

Machine Learning: Foundations

What it means for a machine to learn from data, the handful of ideas that recur everywhere, and how to tell a real result from a flattering one.

FULCRUM

Utsav Poudel
Founder, Fulcrum

ABOUT FULCRUM

Who made this, and why it is free

Fulcrum is a volunteer-run non-profit. We help people from under-resourced regions learn artificial intelligence through one-on-one mentorship, research and publication supervision, project support, workshops, and a guest speaker series.

The premise is simple. Ability is spread evenly across the world; opportunity is not. What separates a capable student in Pokhara or rural Bihar from one at a well-funded lab is rarely intelligence. It is access to supervision, to reviewers, to equipment, and to the unwritten rules nobody writes down.

This deck is part of how we try to close that. It is free, it will stay free, and you may reuse and adapt it for your own teaching under CC BY 4.0. If it is useful to you, come and find us.

Find us

Website
fulcrumgo.github.io

More material
fulcrumgo.github.io/resources

Discord
discord.gg/gbQCGkupdJ

LinkedIn
linkedin.com/company/gofulcrum

Every deck we publish is free to download at fulcrumgo.github.io/resources

PART ONE

Learning means fitting a function you did not write

Every model in this deck is a family of functions plus a rule for picking one.

The three ingredients

- A **model family**, the shape of function you will allow (a line, a tree, a network).
- A **loss function**, a number saying how wrong a given function is on your data.
- An **optimiser**, a procedure for reducing that number by adjusting parameters.
- Change any one and you get a different algorithm. That is genuinely most of the field.

Worth internalising

When you meet an unfamiliar method, ask those three questions. Papers that seem impenetrable usually become simple once you identify the family, the loss and the optimiser.

Supervised, unsupervised, reinforcement

- **Supervised**. You have inputs and correct outputs. Classification and regression live here, and so does most deployed ML.
- **Unsupervised**, inputs only. Clustering, dimensionality reduction, density estimation.
- **Reinforcement**, an agent acts, receives reward, and learns a policy. Powerful, sample-hungry, and harder to get working than the literature suggests.
- **Self-supervised** sits between the first two: labels invented from the data's own structure.

Linear and logistic regression

Start here, and keep coming back here.

- **Linear regression** fits `y = w·x + b` by minimising squared error. It has a closed-form solution.
- **Logistic regression** passes that same linear score through a sigmoid to get a probability, and minimises cross-entropy.
- Both are **interpretable**: each weight says how the prediction moves as one feature moves.
- They are the correct baseline. A complex model that cannot beat logistic regression is telling you something.

In applied work, a linear model with good features very often beats a neural network with bad ones.

Gradient descent

- Compute the gradient of the loss with respect to each parameter, the direction of steepest increase.
- Step in the opposite direction. Repeat.
- The **learning rate** sets step size: too large and you diverge, too small and you never arrive.
- **Stochastic** gradient descent estimates the gradient from a mini-batch, which is noisier per step but far faster overall, and the noise itself helps escape poor minima.

Adam adapts a per-parameter step size and is a reasonable default. It is not magic; a badly scaled problem still trains badly.

Overfitting, and the bias and variance trade-off

- **Underfitting (high bias)**, the model is too simple to capture the pattern. Poor on train and test alike.
- **Overfitting (high variance)**, the model memorised the training set, including its noise. Excellent on train, poor on test.
- The gap between training and validation performance is your main diagnostic. Watch it every run.
- More data reduces variance. It does not fix bias.

The most common student error

Tuning hyperparameters against the test set, then reporting test performance. That number is now meaningless. Keep a set you look at exactly once.

Regularisation

- **L2 (ridge)** penalises large weights, spreading influence across features.
- **L1 (lasso)** drives some weights to exactly zero, performing feature selection as it fits.
- **Early stopping** halts training when validation loss turns upward.
- **Dropout** randomly disables units during training, preventing over-reliance on any single path.

Trees and ensembles

- A **decision tree** splits on features recursively. Highly interpretable, and prone to overfitting alone.
- **Random forests** average many trees trained on bootstrap samples with random feature subsets, cutting variance.
- **Gradient boosting** (XGBoost, LightGBM) fits each new tree to the previous ensemble's errors.
- On tabular data, boosted trees remain extremely hard to beat, including by deep learning.

If your data is a spreadsheet, start with gradient boosting. Reaching straight for a neural network is a common and expensive mistake.

Evaluation that will survive a reviewer

- **Cross-validation** gives a variance estimate, not just a point score. Report the spread.
- **Precision, recall, F1** for imbalanced problems; state which error type actually matters in your setting.
- **ROC-AUC** summarises ranking quality across thresholds; **PR-AUC** is more informative under heavy imbalance.
- Always report a **baseline**, majority class, or a simple linear model. A number without a baseline means nothing.

Features, leakage, and other quiet disasters

- **Leakage** is any information in your features that would not exist at prediction time. It is the leading cause of results that collapse in deployment.
- Scale and impute using statistics computed on the **training fold only**, then apply to validation. Doing it beforehand leaks.
- Beware features that encode the answer, a patient ID that correlates with ward, and therefore diagnosis.
- For time series, split **chronologically**. Random splits let the model see the future.

If a result seems too good, look for leakage before you look for a discovery. It is the explanation the overwhelming majority of the time.

A workflow that holds up

- Frame the question and decide **in advance** what result would be meaningful.
- Build the dumbest baseline that could work. Record it.
- Improve one thing at a time, and keep a log of what you tried and what happened.
- Test once, at the end. Report honestly, including what failed.

Why the log matters

Six months on, writing the paper, you will not remember which of forty runs used which settings. The log is the difference between a reproducible result and a story.

ABOUT FULCRUM

Next: Deep Learning, Foundations

Neural networks, backpropagation, and what actually changes when you stack layers.

Fulcrum is a volunteer-run non-profit helping people from under-resourced regions learn AI through free mentorship, research supervision, and open material like this deck. Founded 2025 in Kathmandu, Nepal. Nothing we do costs anything.

Website
fulcrumgo.github.io

Every deck, free
fulcrumgo.github.io/resources

Apply for mentorship
discord.gg/gbQCGkupdJ

Partnerships
linkedin.com/company/gofulcrum