Biologist's survival guide to ML

written by Cris Darbellay

What is machine learning?

Artificial intelligence (AI) is the broad idea of machines performing tasks that appear intelligent. **Machine learning (ML)** is a subfield of AI in which systems learn useful patterns from data instead of being fully programmed by hand. **Deep learning (DL)** is a powerful family of ML methods based on multi-layer neural networks that learn rich internal representations.

In practice, a machine learning project usually follows a simple logic: Define a question, collect data, choose inputs and targets, train a model, assess performance, use the trained model on new data.
A useful way to think about ML is this: instead of explicitly writing all the rules ourselves, we give the computer many examples, and it learns a mathematical function that maps inputs to outputs. In biology, those inputs may be sequences, images, metadata, coordinates, or measurements; the outputs may be classes, continuous scores, predicted properties, protein's structures or even newly generated candidates

Core workflow of an ML project

A core principle in ML is to separate data according to their role.

- **Training set:** used to fit the model parameters.
- **Validation set**: held out during training; used to compare model versions, tune settings, and detect overfitting.
- **Test set:** untouched until the end; used once for an honest final estimate of performance.

If we evaluate a model only on the data it trained on, we may overestimate its true usefulness. The model may simply memorize patterns that do not generalize. The validation set helps answer questions such as:

- Should the model be larger or smaller?
- Which learning rate works best?
- Which version should be selected?

The test set answers the final question: *How well should we expect this model to perform on genuinely new data?*

Overfitting, underfitting, and the bias–variance tradeoff

Overfitting

A model **overfits** when it learns the training data too specifically, including noise or accidental patterns, and therefore performs poorly on new examples. Typical signs:

- very good training performance
- clearly worse validation/test performance.
-

Underfitting

A model underfits when it is too simple, too constrained, or insufficiently trained to capture the important structure in the data. Typical signs:

- poor performance on both training and validation data.

Bias–variance tradeoff

This is one of the most important conceptual ideas in ML.

- **High bias:** the model is too simple and misses the true pattern.
- **High variance:** the model is too sensitive to the specific training data and does not generalize well.

Bigger goals tend to have lower bias and higher variance. The practical goal is to find a balance: a model complex enough to learn the important biological signal, but not so flexible that it memorizes noise. This tradeoff explains why larger models can be powerful but also risk overfitting if data are limited or poorly sampled.

Regularization

Regularization refers to strategies used to reduce overfitting. Examples include:

- limiting model complexity
- early stopping
- dropout,
- weight decay
- data augmentation.

Deep learning and representations

Why deep learning matters

Traditional ML often depends heavily on manually designed features. Deep learning aims to learn useful features directly from raw data. This is especially useful in biology because:

- biological data are often high-dimensional
- important patterns may be subtle and hierarchical
- useful features are not always obvious in advance.

Representation learning

A key idea in deep learning is the **embedding** or **representation**: a learned numerical encoding that captures useful structure in the data.
For example, two biologically similar objects may end up with similar internal representations even if their raw descriptions look different.
This idea is powerful because the model can transform complex raw inputs into a space where prediction becomes easier.

Neural networks

A neural network is a layered function that progressively transforms inputs into more abstract internal representations. Deeper networks can learn more complex mappings, though they also require more data, more compute, and careful training.

Modern ML paradigms

Supervised learning

The model learns from examples paired with known targets. Examples:

- image → cell type
- patient variables → diagnosis
- sequence → experimentally-determined structure

Unsupervised learning

The model tries to find structure in the data without explicit targets. This may involve clustering, dimensionality reduction, or anomaly detection.

Self-supervised learning

The model creates its own learning task from the data itself. For example, it may hide part of an input and learn to reconstruct it. This is extremely important in modern biology because unlabeled data are abundant, while carefully labeled data are often scarce and expensive.

Generative models

A **generative mode**l learns to produce new plausible examples. Depending on the application, it may generate text, images, molecular candidates, sequences, or structures. Generative models are especially interesting in science because they are not limited to prediction; they can also propose new hypotheses or new candidates to test.

Architectures: what makes models different?

Different ML models are not just different “software tools”; they are different mathematical designs with different assumptions. We talk about “model size” to refer to the number of trainable parameters. A bigger model can learn more complex relationships but is also more computationally demanding to train and run. Classical models are the smallest with typically only a couple parameters. Models powering today’s chatbots and protein design tools can have hundreds of billions.

Classical models

Examples include linear regression, logistic regression, decision trees, and random forests. These are often simpler, easier to interpret, and strong baselines.

Deep learning architectures

Common examples:

- **Convolutional neural networks (CNNs):** strong for images and spatial grids.
- **Recurrent neural networks (RNNs):** historically used for sequential data.
- **Graph neural networks (GNNs):** useful when data can be naturally represented as graphs, such as molecules or interaction networks.
- **Transformers:** now central for sequences and many other domains.
- **Diffusion models:** powerful generative models based on iterative denoising.

Transformers

Transformers are architectures built around the idea of **attention**. Attention allows the model to decide which parts of the input are most relevant when processing each element.This is useful because many biological patterns depend on long-range relationships:

- Distant residues in a sequence may be close in 3D space and interact.
- Distant genomic regions may regulate each other
- context often matters more than local proximity alone. T

Transformers became highly influential because they scale well, learn rich representations, and work effectively in supervised, self-supervised, and generative settings. They power today’s chatbot technologies (ChatGPT, Gemini,...)

Performance, confidence, and calibration

A good model is not only accurate; it should also provide meaningful confidence.

Calibration

A model is **well calibrated** if its confidence estimates match reality. If it says “90% confidence” that should correspond, on average, to being correct about 90% of the time. Calibration matters in science and medicine because decisions may depend not only on the predicted answer, but also on how trustworthy the confidence appears.

Metrics

Model performance must be measured using appropriate metrics. The right metric depends on the task:

- accuracy, precision, recall, F1-score for classification
- mean squared error or correlation for regression
- likelihood or sample quality for generative models.

A model can perform well on one metric and poorly on another, so metrics must be chosen according to the biological objective.

Dictionary of essential terms

Artificial intelligence (AI). The broad idea of machines performing tasks that appear intelligent.
Machine learning (ML). A subfield of AI in which a system learns useful patterns from data.
Deep learning (DL). ML based on multi-layer neural networks that learn rich representations.
Sample / example. One observation in a dataset, such as one sequence, one image, or one patient.
Feature / input. The information fed into the model: sequence, image, coordinates, metadata, etc.
Label / target. The quantity the model is asked to predict: class, score, activity, structure, etc.
Dataset. The full collection of examples used for development and evaluation.
Training set. The subset used to fit model parameters.
Validation set. The held-out subset used to tune settings and compare model versions.
Test set. The final held-out subset used for an honest estimate of performance.
Model. A parameterized mathematical function that maps inputs to outputs, often written as $f_\theta(x)$.
Parameters / weights. The learned numerical values inside the model.
Architecture. The overall model design (for example transformer, graph neural network, diffusion model).
Embedding / representation. A learned numerical encoding that captures useful relationships in the data.

Loss function. A number measuring how wrong the model is; training tries to minimize it.
Gradient. The direction telling us how to change parameters to reduce the loss.
Optimizer. The update rule that uses gradients to adjust parameters (for example SGD or Adam).
Learning rate. The step size used by the optimizer.
Batch / mini-batch. A small subset of examples used for one update step.
Epoch. One full pass through the training set.
Inference. Using a trained model on new data.
Supervised learning. Learning from inputs paired with known targets.
Unsupervised learning. Finding structure in data without explicit labels.
Self-supervised learning. Creating learning tasks from the data itself (for example masked-token prediction).
Generative model. A model that creates new plausible examples.

Classification. Predicting a category.
Regression. Predicting a continuous value.
Overfitting. Memorizing the training set and failing on new data. Underfitting. Failing to capture the important pattern.
Generalization. Performance on unseen examples.
Regularization. Any strategy used to reduce overfitting.
Bias–variance tradeoff. The balance between models that are too simple and models that are too sensitive to training noise.
Calibration. Whether confidence estimates match reality.
Transformer. A neural network architecture based on attention, designed to model relationships across an input sequence or set.
Attention. A mechanism allowing the model to focus on the most relevant parts of the input when making a prediction.

Good luck and Enjoy the Ride!