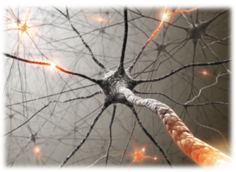
Machine learning: overview



- In this module, I will provide an overview of the topics we plan to cover under machine learning.

Course plan

Search problems
Markov decision processes
Adversarial games

Constraint satisfaction problems
Markov networks
Bayesian networks

Reflex **States** **Variables** **Logic**

"Low-level intelligence" "High-level intelligence"

Machine learning

- Recall that machine learning is the process of turning data into a model. Then with that model, you can perform inference on it to make predictions.

Course plan

Search problems
Markov decision processes
Adversarial games

Constraint satisfaction problems
Markov networks
Bayesian networks

Reflex **States** **Variables** **Logic**

"Low-level intelligence" "High-level intelligence"

Machine learning

- While machine learning can be applied to any type of model, we will largely focus on reflex-based models, which include models such as linear classifiers and neural networks.
- In reflex-based models, inference (prediction) involves a fixed set of fast, feedforward operations.

Reflex-based models

predictor

input x $\longrightarrow$ $\boxed{f}$ $\longrightarrow$ y **output**

- Abstractly, a **reflex-based model** (which we will call a **predictor** f) takes some **input** x and produces some **output** y.
- Predictors can also be referred to as models or function approximators, in the sense that they are trying to approximate some function.
- The input can usually be arbitrary (an image or sentence), but the form of the output y is generally restricted, and what it is determines the type of **prediction task**.

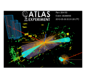
Binary classification

classifier

x $\longrightarrow$ $\boxed{f}$ $\longrightarrow$ $y \in \{+1, -1\}$ **label**

Fraud detection: credit card transaction $\rightarrow$ fraud or no fraud

Toxic comments: online comment $\rightarrow$ toxic or not toxic

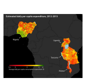


Higgs boson: measurements of event $\rightarrow$ decay event or background

- One common prediction task is binary classification, where the output y, typically expressed as positive $(+1)$ or negative (-1).
- In the context of classification tasks, f is called a **classifier** and y is called a **label** (sometimes class, category, or tag).
- Here are some practical applications.
- One application is fraud detection: given information about a credit card transaction, predict whether it is a fradulent transaction or not, so that the transaction can be immediately blocked.
- Another application is moderating online discussion forums: given an online comment, predict whether it is toxic (and therefore should get flagged) or not.
- A final example comes from physics: After the discovery of the Higgs boson, scientists were interested in how it decays. The Large Hadron Collider at CERN smashes protons against each other and then detects the ensuing events. The goal is to predict whether each event is a Higgs boson decaying or just background noise.
- Each of these applications has an associated Kaggle dataset. You can click on the pictures to find out more details.
- As an aside, **multiclass classification** is a generalization of binary classification where the output y could be one of K possible values. For example, in digit classification, $K = 10$.

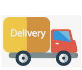
Regression

x $\longrightarrow$ $\boxed{f}$ $\longrightarrow$ $y \in \mathbb{R}$ **response**



Poverty mapping: satellite image $\rightarrow$ asset wealth index

Housing: information about house $\rightarrow$ price



Arrival times: destination, weather, time $\rightarrow$ time of arrival

- The second major type of prediction task we'll cover is regression. Here, the output y is a real number (often called the **response** or target).
- One application of regression is poverty mapping: given a satellite image, predict the average asset wealth index of the homes in that area. This is used to measure poverty across the world and determine which areas are in greatest need of aid.
- Another application: given information about a house (e.g., location, number of bedrooms), predict its price.
- A third application is to predict the arrival time of some service, which could be package deliveries, flights, or rideshares.
- The key distinction between classification and regression is that classification has **discrete** outputs (e.g., "yes" or "no" for binary classification), whereas regression has **continuous** outputs. For example, housing prices can be any positive real number.

Structured prediction

$x \longrightarrow \boxed{f} \longrightarrow y$ is a complex object

Machine translation: English sentence $\rightarrow$ Japanese sentence

Dialogue: conversational history $\rightarrow$ next utterance

Image captioning: image $\rightarrow$ sentence describing image

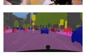


Image segmentation: image $\rightarrow$ segmentation

- The final type of prediction task we will consider is structured prediction, which is a bit of a catch all term.
- In **structured prediction**, the output y is a complex object, which could be a sentence, an image, or a graph even. So the space of possible outputs is much much larger.
- One application is machine translation: given an input sentence in one language, predict its translation into another language.
- Dialogue can also be cast as structured prediction: given the past conversational history between a user and an agent (in the case of virtual assistants), predict the next utterance (what the agent should say).
- In image captioning, say for assistive technologies: given an image, predict a sentence describing what is in that image.
- In image segmentation, which is needed to localize objects for autonomous driving: given an image of a scene, predict the segmentation of that image into regions corresponding to objects in the world.
- Generating an image or a sentence can seem daunting, but there's a trick that we can use to make it easier. A structured prediction task can often be broken up into a sequence of multiclass classification tasks. For example, to predict an entire sentence, predict one word at a time, going left to right. This is a very powerful reduction! But structured prediction is generally still much more difficult than classification and regression.
- Aside: one challenge with this approach is that the errors might cascade: if you start making errors, then you might go off the rails and start making even more errors.

Roadmap

Tasks	Models
Linear regression	Non-linear features
Linear classification	Feature templates
K-means	Neural networks
	Differentiable programming

Algorithms	Considerations
Stochastic gradient descent	Generalization
Backpropagation	Best practices

- Now here is an overview of the rest of the modules under the machine learning unit.
- We will start by talking about regression and binary classification, the two most fundamental tasks in machine learning. Specifically, we study the simplest setting: **linear regression** and **linear classification**, where we have linear models trained by gradient descent.
- Next, we will introduce **stochastic gradient descent**, and show that it can be much faster than vanilla gradient descent.
- Next week, we will push the limits of linear models by showing how you can define **non-linear features**, which effectively gives us non-linear predictors using the machinery of linear models! **Feature templates** provide us with a framework for organizing the set of features.
- Then we introduce **neural networks**, which also provide non-linear predictors, but allow these non-linear features to be learned automatically from data. We follow up immediately with **backpropagation**, an algorithm that allows us to automatically compute gradients needed for training without having to take gradients manually.
- We then briefly discuss the extension of neural networks to **differentiable programming**, which allows us to easily build up many of the existing state-of-the-art deep learning models in NLP and computer vision like lego blocks.
- All of these modules so far focus on supervised learning. We take a brief detour and discuss **K-means**, which is a simple unsupervised learning algorithm that aims to discover structure in data without any labels.
- We will end by discussing some important considerations in machine learning: **Generalization** is about answering the question: when does a model trained on set of training examples actually perform well on new test inputs? This is where model complexity comes up. Finally, we discuss **best practices** for doing machine learning in practice.