Sketching out the Machine Learning Pipeline

COSC 410: Applied Machine Learning
Fall 2025
Prof. Forrest Davis

August 29, 2025

Warm-up

1. Talk to person next to you about the best (or worst) movie you saw this summer.

Logistics

- Course website: https://forrestdavis.github.io/cosc410
- Complete survey: https://forms.gle/yHhPQxwhFSLrCeWU7

Learning Objectives

- Describe the 4 key parts of an ML pipeline
- Articulate the concept of paradigm and other key jargon for the semester (including the basic mathematical objects for data)
- Engage with limitations of machine learning
- Identify which type of paradigm we need for various problems

Summary: In a (hopefully unusually) long-winded handout, we lay out the basic components of machine learning (data, models, objective functions, and optimization), discuss the basic pipeline, and define terms we will use throughout the semester.

Data

DATA IS COMPRISED OF two components: (1) samples and (2) features. Consider data on the price of houses in Hamilton. One *sample* would represent one *house* which has features such as *square footage, number of bedrooms, number of bathrooms, whether there is a garage*, etc.[1]

Mathematically, we denote single samples as **vectors**.[2] vectors with lowercased and bolded letters (e.g., $\mathbf{x}, \mathbf{u}$) or with an arrow on top of

[1] Notice that some of these features are **continuous,** like square footage (i.e., all values are meaningful in the range, so we might expect different prices for houses with a difference in square footage of 1110.5), and others are **discrete**, like whether there is a garage (i.e., this is either true or false, or 1 or 0). It is often the case in things like regression that these types of variables are treated differently. A lab will cover some of the applied cases of this.

[2] While column vectors, as in the vector example given, are the default assumption in mathematics, they are often written as row vectors (i.e., as a single row rather than a single column). I will swap between these, as the book also does from time to time. Where it matters, I'll try to note it.

it (often when handwritten; e.g., $\vec{x}, \vec{u}$). For example, let's consider a house with 2 beds 2.5 baths, 8000 sqft, and no garage:

$$\vec{\text{house}} = \begin{bmatrix} 2 \\ 2.5 \\ 8000 \\ 0 \end{bmatrix}$$

Now, one sample is mostly useless for machine learning. Instead, we will be often working with millions (even trillions in real settings) of samples. To represent data mathematically, we will use **matrices**, denoted with bold uppercase letters (e.g., $\mathbf{Z}, \mathbf{Y}$). Recall, matrices are collections of vectors, whose shape is specified by number of rows and number of columns (in that order; denoted as a superscript, so $\mathbf{Z}^{3\times2}$ is a matrix with 3 rows and 2 columns). We follow a convention (also followed by many machine learning books and papers) that rows in a matrix represent samples and columns represent features. Expanding our example to have another house, this time with 3 beds 1.5 baths, 6000 sqft, and a garage:

$$\mathbf{H} = \begin{bmatrix} 2 & 2.5 & 8000 & 0 \\ 3 & 1.5 & 6000 & 1 \end{bmatrix}$$

As we progress in this course, we will extend our work from matrices to **tensors**, which are N-dimensional mathematical objects (1-dimensional tensors are vectors and 2-dimensional tensors are matrices). We will denote them with capital letters, as with matrices, sometimes with a fancy font if I remember. Imagine we are modeling house prices over time. We may, then, have 1000 houses with 6 features and prices measured once a year over ten years. This would be a tensor like $\mathcal{T}^{10\times1000\times6}$.

Data is the material basis for so-called (generative) artificial intelligence. Nonetheless, the labor behind it is often undervalued and marginalized in discussions of current advanced models. Much of the data annotation necessary for chat models, like ChatGPT, is done by workers in the third world at extremely low wages. For a harrowing example, see Perrigo [2023].

Question

What is the shape of $\mathbf{H}$? In the textbook, m denotes the number of samples and n the number of features. Using these variables, what is the generic shape of data?

I cannot stress enough how important it is to know and keep in mind the basic dimensions of your tensors and their shapes/meanings in working with machine learning. As we use PyTorch to do more sophisticated things, I highly recommend adding comments in your code that specify both the shape and meaning of the dimensions of your data (and other objects). I promise you, if you really want to build neural network models, this consistent practice is key to success (including helping you debug many things).

Models

MODELS ARE A REPRESENTATION OF DATA, a representation of the function that may underline the creation of the data, or a function

that transforms data from one form to another. These models can be simply rules, like determining whether it is hot or cold based on applying a threshold to a temperature, or be complex functions, determining the label of an image from its pixel values.

We will unpack common modeling terms below. For now, it's important to note that we are often working with **statistical models** which represent an approximation (determined from data) of some process often used to produce some output associated with some likelihood. For example, I might build a statistical model to infer a function that determines the likelihood I'll eat ice cream or not. I might train that model on data that consists of samples with three features, the daily high, the daily low, and whether I ate ice cream (good luck to that model, I eat ice cream regardless of the season).[3]

Our model is often comprised of **parameters**, which are tweaked during learning and represent the function learned by the model to solve the relevant task. This will become more clear next week, but I mention it here, just so the jargon is around for the end of the handout.

[3] Note, often the dependent feature, in this case whether I ate ice cream, is separated from the indepedent features, in this case the temperatures, in a different vector. We will do this later.

Question

Is a Python program a model? Why or why not?

Objective Functions and Optimization Algorithms

OBJECTIVE FUNCTIONS ARE what we aim our model at doing.[4] All I'll say today is that we create objective functions to "score" a model's ability. Lower is better, and we, therefore, aim to minimize our objective function during training (as minimizing the objective function means we found the best model we could given the data).

[4] We also call objective functions loss functions or cost functions.

Basic Pipeline Applied to a Task: Sentiment Analysis

SENTIMENT ANALYSIS IS THE TASK of predicting whether text is positive or negative.[5]. For example, we can apply this to movie reviews, in hopes of rating reviews like *The new Superman movie was fun* as positive and reviews like *I almost fell asleep while watching The Materialists* as negative.

[5] Sentiment analysis is setup with a variety of different labels, including *positive*, *negative*, or *neutral*, but also things like *happy*, *angry*, etc. The labels don't matter so much as defining a problem which maps from text to discrete labels representing some aspect of the "vibe" of the text

Question

In Figure 1, I sketch out the pipeline as applied to this task. What do you think examples of data formatting are for this task? How might you evaluate the performance of your model?

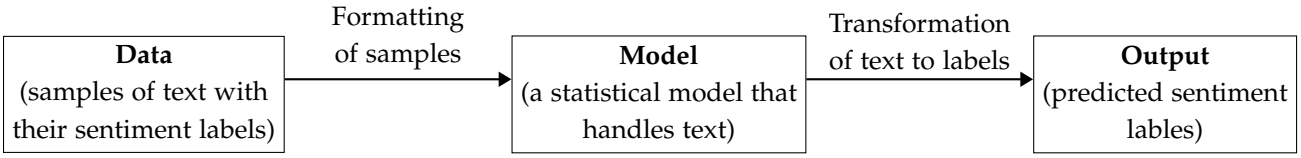

Figure 1: A sketch of the machine learning pipeline. The boxes represent elements where you, as an engineer, interject in the process. You must find, inspect, treat, and format the data, specify a model (and its computations), and inspect, validate, and evaluate the output of the model.

Paradigms and Jargon

IN THINKING ABOUT TASKS FOR THIS COURSE, we will use **paradigms**. This is non-standard (i.e., idiosyncratic to this class), but I hope nonetheless helpful. Our paradigm is comprised of two components: (1) the type of learning and (2) the type of system. When you approach a task, you should be able to identify the paradigm you want to use and justify the choice of each component.

Paradigms: Types of Learning

- **Supervised Learning** Your data has the labels you want to predict. This could be for *regression* where the labels are continuous (e.g., you want to predict the price of a house and your data on houses includes each house's price), or *classification* where the labels are discrete (e.g., you want to predict whether a pet picture is a cat or a dog and your data includes images labeled as cat or dog).
- **Unsupervised Learning** Your data lacks the labels you want to predict. Examples of this could be clustering patient data into groups to determine whether there are specific populations that respond to certain medication well, or detecting anomalies in network traffic to see if a hacker has broken in.
- **Semi-supervised Learning** Your data is only partially labeled (e.g., you only had enough money to label 60% of your images as

cats or dogs). Your model mixes techniques from supervised and unsupervised learning.

- **Reinforcement Learning** You tasks is to learn a **policy** or a sequence of **actions** that you should take in order to achieve a goal. For example, you may want to have a robot learn how to pick up a cup with (or without) shattering it.

Paradigms: Type of System

- **Shallow Learning** Your data comes with (or you create) rich features that are useful for learning. For example, the housing data comes with information that is useful for the task (e.g., number of bedrooms).
- **Deep Learning** Your data lacks rich features and part of your ML algorithm is creating features relevant to the task. For example, you may just want to pass a photo of animals (which are just pixel values) to a model which then extracts useful features as part of its learning. These features might be something like, whether the photo has an animal with whiskers.[6]

Language modeling, which is the objective behind the base models in generative AI is often considered 'semi-supervised' learning. In reality, this depends a lot on how you conceptualize the task. If you think you are building a model that can do a variety of tasks (e.g., predict sentiment, label words), than the modeling has two phases: an unsupervised (or 'self-supervised') component where you train a model to reproduce the training data and a supervised component where you add in the relevant tasks. If you think of it as strictly language modeling, then you have a supervised task of determing the probability of words (or sentences). These lines are often blurred perhaps to market the models as more mysterious than they are.

Question

What do you think are the pros and cons of shallow vs deep learning?

[6] Part of what makes neural networks 'black boxes' is that we often don't have a way of cleaning determining what abstract features are built during training.

Other Jargon

- **Batch (Offline) vs Online Learning** In batch learning, the ML system is trained all at once with all available data. If new data is collected, the system must be re-trained with new and old data. With online learning, the ML system can be upgraded incrementally with new data.

Question

What do you think are the pros and cons of offline vs online learning?

- **Instance-based vs Model-based Learning** In instance-based learning, the model memorizes all provided data and makes predictions for new data points based on all memorized examples. In model-based learning, the model uses provided data to create a representation of the data, and makes predictions based on the model, rather than strictly the data.

Some Limitations of Model-Based Learning

DATA ISN'T ENOUGH TO LEARN REPRESENTATIONS. There does not exist a un-biased, universal learning algorithm that is capable of learning any type of task (in practice or in theory). All learners, biological or otherwise, must be constrained in order to meaningfully generalize from data. That is, the learner has to make assumptions about the form of their desired solution.

Practice

Practice Problems

Complete these with the people around you. For the following problem descriptions, identify with justification, the relevant paradigm.

1. You would like to determine how exercise impacts lung capacity. You work at a hospital and have access to tons of anonymized patient data which includes lung capacity and survey information about habits.

2. You work at Amazon and you'd like to use existing product ratings (e.g., a rating of 4.7) to determine whether a new user would like the item

Practice Problems

Complete these with the people around you. For the following problem descriptions, identify with justification, the relevant paradigm.

1. You are developing a stock trading algorithm to learn when prices for a stock will change

2. You are an technician for a campus network and want to identify whether traffic on your network is malicious

Before Next Class

- Complete the pre-class quiz (no associated reading)
- Complete Lab 0, linked on the course website under the Labs tab (just setting up your computer for the course)
- Make some progress on Codelet 0 (on the website under the Codelets tab), which is Due Sep 5

References

Bill Perrigo. OpenAI Used Kenyan Workers on Less Than $2 Per Hour to Make ChatGPT Less Toxic. *Time*, 18 Jan 2023. URL https://time.com/6247678/openai-chatgpt-kenya-workers/.