TASK

Introduction to Machine Learning

Visit our website

Introduction

WELCOME TO THE INTRODUCTION TO MACHINE LEARNING TASK!

This task introduces you to the most important concepts in Machine Learning, giving you a general overview of the landscape and preparing you to learn about specific supervised and unsupervised learning algorithms.

A note from the

HyperionDev Team

The history of Machine Learning

For decades, visions of machines that can learn the way humans can have captured the imagination of science-fiction authors and researchers alike. But only in recent years have machine learning programs been developed that can be applied on a wide scale, influencing our daily lives.

Machine learning programs are working behind the scenes to produce and curate our playlists, news feeds, weather reports and email inboxes, and they help us find restaurants, translate documents, and even meet potential dates. From a business perspective, machine learning-based software is becoming central to many industries, generating demand for experts.

In 1959, Arthur Samuel, a pioneer in artificial intelligence and gaming defined machine learning as the "field of study that gives computers the ability to learn without being explicitly programmed." He is known for developing a program capable of playing checkers. Samuel never programmed exactly which strategies the systems could use. Instead, he devised a way in which the program could learn such strategies through the experience of playing thousands of games.

In the 50s, machines were hard to acquire and not very powerful, so machine learning algorithms were mostly an object of theoretical research. Now that computers are vastly more powerful and more affordable, machine learning has become a very active field of study with a variety of real-world applications.

DEFINING MACHINE LEARNING

A widely used definition of machine learning is that provided by Tom Mitchell, author of a well-known textbook simply titled “Machine Learning”. This definition is as follows:

"A program can be said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

According to this definition, a program has ‘learned’ how to recognise animals in pictures if it were shown many pictures with animals in them and able to use those pictures to improve its score when attaching animal names to pictures.

The term “machine learning” is often used interchangeably with the term “artificial intelligence” (AI). While the two are very much related, they are not the same thing. There is much debate about the difference between the two, but a simple way to look at it for our purposes is to see Machine Learning as a *type* of artificial intelligence. Any program that completes a task in a way that can be considered human-like can be considered an example of artificial intelligence, but only programs that solve the task by learning without pre-programming are machine learning programs.

A note from our coding mentor
Ridhaa

*Did you know that a London-based company called Google **DeepMind** has developed an artificial intelligence-based gamer, which can play 49 video games from Atari 2600 and achieves better than a professional human player’s top score in 23 of them? Yes, you read that right!*

According to an article, the link to which is provided below, “The software isn’t told the rules of the game – instead, it uses an algorithm called a deep neural network to examine the state of the game and figure out which actions produce the highest total score.”

*One of the most impressive, and probably the eeriest example, is that in the boxing game, the software learned how to pin its opponent on the ropes (which is something only seasoned players of the game knew how to do), and release a barrage of punches until its opponent was knocked out! Extremely ruthless, right? (**Source**)*

INPUT AND OUTPUT

Whatever it is that we want a machine learning algorithm to learn, we first need to express it in a numerical way. The machine-readable version of a task consists of an **input** and an **output**. The input is whatever we want the algorithm to learn from, and the output is the outcome we want the algorithm to be able to produce. An example of an input would be the budget or number of awards received by a movie. An example of output would be the box office sales of that movie.

Since machine learning is a young field that overlaps with several other disciplines, including statistics, the input and output may be referred to by several other names.
For input, these include:

- features (named after the fact that inputs typically 'describe' something),
- independent variables, and
- explanatory variables (named due to the fact that the output is usually assumed to depend on or be explained by the input).

For output, alternate terms are:

- labels,
- predictions,
- dependent variables, and
- response variables.

EXPERIENCE

The defining feature of machine learning is its ability to use experience to find an optimal solution. What 'experience' are we talking about? Essentially, it is data.

Remember the definition of machine learning? Specifically, there were some important aspects to it: the task T, performance measure P and experience E. To best explain this, let's take a look at an example of machine learning in action: a program that, given a picture of a dog or a cat, can tell whether the picture is of a dog or a cat.

Our task T is to differentiate between a dog or a cat. This is done by building up a **model** which uses a number of **parameters**. If you get a very specific combination fof these parameters, your model will accomplish this task. How is this done? Using some form of experience E.

Experience simply refers to the model trying to make predictions between dogs and cats. The machine learning algorithm will start by taking a number of guesses between dogs and cats. This will achieve some performance measure P, which will depend on the number of correct and incorrect guesses. The 'learning' process then uses an update rule to change the model **parameters** to make a better and more informed guess. This is the basis of machine learning. This process is heavily reliant on data.

Supervised learning

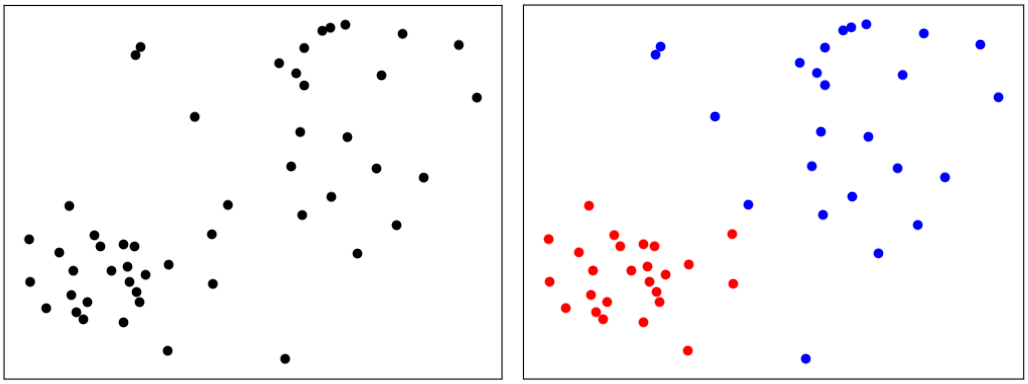
In supervised learning problems, a program predicts an output for an input by learning from pairs of inputs and outputs (labels); that is, the program learns from examples that have had the right answers assigned to them beforehand. These assignments are often called **annotations**. Because they are considered the correct answers, they are also called **gold labels**, **gold data**, or the **gold standard**.

The collection of examples that comprise the supervised experience is called a **training set**. A collection of examples that are used to assess the performance of a program is called a **test set**. Like a student learning in a language course that teaches only through exposure, supervised learning problems see a collection of correct answers to a variety of questions, and must learn to provide correct answers to new, but similar questions.

Unsupervised learning

In unsupervised learning, a program does not learn from labelled data. Instead, it attempts to discover patterns in the data on its own.

For example, suppose you have two classes scattered in a 2-dimensional space (as in the first of the images below) and you want to separate the two data sets (as in the second image). Unsupervised learning finds underlying patterns in the data, allowing the classes to be separated.



To highlight the difference between supervised and unsupervised learning, consider the following example. Assume that you have collected data describing the heights and weights of people. An unsupervised clustering algorithm might produce groups that correspond to men and women, or children and adults. An example of a supervised learning problem is if we label some of the data with the person's sex, and then try to induce a rule to predict whether a person is male or female based on their height and weight.

Semi-supervised learning

Supervised learning and unsupervised learning can be thought of as occupying opposite ends of a spectrum. Some types of problems, so-called "semi-supervised" learning problems, are in the middle of the spectrum and have access to only some human supervision.

One example is an approach that only uses human annotations for a subset of the data. Another is distant supervision, where labels are generated for input automatically, but the automatic generation itself took human annotations as its input. Yet another example is reinforcement learning, in which a program receives supervision that is more of a general rating than a strict set of correct answers. For example, a reinforcement learning program that learns to play a side-scrolling video game such as Super Mario Bros. will be given a reward for behaviour that makes it likely to win (such as completing a level or attaining a high score), but a punishment for losing a life. This feedback is quite general, and not about which specific actions to execute.

Compulsory Task 1

Answer the following in a document titled **Intro_to_ML**. Convert your answer document to a PDF before submitting it.

- For each of the following examples describe at least one possible **input and output**:
  - A self-driving car
  - Netflix recommendation system
  - Signature recognition
  - Medical diagnosis

Completed the task(s)?

Ask an expert to review your work!

Review work

Rate us

Share your thoughts

HyperionDev strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved, or think we've done a good job?

Click here to share your thoughts anonymously.