Machine Learning Specialization

Course Notes

JESUS CARDENAZ

1   Supervised Vs. Unsupervised Machine Learning

What is Machine Learning?

"Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed." Arthur Samuel (1959)

The two main types of Machine Learning are Supervised Learning and Unsupervised Learning. Supervised Learning, being the most widely used, is used in many real-world applications and has seen the most rapid advancements.

Arthur Samuel's checkers playing program is a historical example of a learning algorithm.

What is Supervised Learning

In Supervised Learning, algorithms learn input to output mappings from provided examples. The input and correct output (x and y) are provided to the learning algorithm, and by seeing these correct pairs, it learns to predict the output from the given input.

This method of learning is used in spam filters, speech recognition, machine translation, online advertising, self-driving cars, and product inspection in manufacturing

Most Common Types of Supervised Learning:

- Classification: Predicts categorical outputs from a limited set of possibilities.

- Regression: Predicts numerical outputs from infinite possibilities.

> **Note 1**
>
> Classification differs from regression in that it does not predict all possible numbers in between categories

Classification   Classification is a type of supervised learning where the output of the variable is categorical. i.e, it belongs to a specific finite set of categories or classes. The model is trained to predict the class of the given input data. Common classification algorithms include:

- Logistic Regression

- K-nearest Neighbors

- Support Vector Machines

- Decision Trees

- Random Forests

Regression   Regression is another type of supervised learning where the output variable is continuous or numerical. The model learns to predict a value on the input data. Common regression algorithms include:

- Linear Regression

- Polynomial Regression

- Decision Trees

- Random Forest

- Support Vector Regression

In addition to classification and regression, there are other types of supervised learning that cater to more specific needs:

- Multi-label Classification.

- Multivariate Regression.

- Ordinal Regression (Ranking Learning).

- Time Series Prediction.

What is Unsupervised Learning?

Unsupervised learning is another type of machine learning where we provide the algorithm with data that isn't linked with any output labels. The algorithm's task is to identify patterns or structures in the data.

A specific form of unsupervised learning is a clustering algorithm, used for instance in Google News to group related stories.

> **Note**
>
> Unsupervised Learning involves identifying patterns and structures in data without any pre-existing labels.

Clustering   Clustering is a specific type of unsupervised learning where the algorithm groups related data together. This technique can be used in many applications, such as news grouping, genetic data classification, and customer segmentation.

Anomaly Detection   An unsupervised learning technique that is used to detect unusual events in the data, which could potentially be signs of fraudulent activity or other abnormalities.

Dimensionality Reduction   Reduces a large dataset into a smaller one while retaining as much important information as possible.

Linear Regression Model

Training Set   A training set in supervised learning includes both the input features, such as the size of a house and also the output targets, such as the price of the house.

Function/Hypothesis   The function or Hypothesis, often denoted as f, is generated by the learning algorithm. Its job is to take an input x and output an estimate or prediction $\hat{y}$.

Where x is called the input or the input feature, the output of the model is the prediction, and y is the actual true value in the training set.

Model   The function f is also known as the model. The model takes in inputs and makes predictions based on what it has learned from the training data.

$\hat{y}$ y-hat is the prediction or estimated output of the model for a given input. It is based on the model's learned function.

How to represent f

$$f_{w,b} = wx + b$$

The expression represents a linear function, also known as a linear model or line equation. You might remember the equation of the straight line $y = mx + c$, well the equation in the context of machine learning, describes a simple linear regression model, which tries to predict a dependent variable (y) as a linear combination of independent variables (x).

Breaking down each component of the formula we have:

- $f_{w,b}$: This is your function (or model) parameterized by w and b. The goal in machine learning is to adjust these parameters to best fit your data.

- w: This is the weight or slope in th elinear function. In machine learning, it quantifies the strength of the relationship between the feature x and the target variable. It tells us how much the dependent varaible y is expected to increase when teh independent variable x increases by one.

- x: This represents the independent variable or feature. In machine learning context, it's the input data you are trying to make predictions from.

- b: This is the bias or y-intercept in the linear function. It is the value of the target when all independent variables are zero.

*An alternative to writing $f_{w,b}$ is simply writing $f(x)$.

Note

This type of regression is also known as Linear Regression with one variable, or Univariate Linear Regression

House Prices vs House Sizes with Regression Line

Model Parameters (weights/coefficients)   These are the variables in the model that are learned from the training data and used to make predictions on new data. For linear regression, these parameters are w (weight or slope) and b (bias or y-intercept).

Cost Function   Also known as loss or error function, it measures the difference between the actual output and the model's predicted output, given the input. **The goal of a machine learning algorithm is to minimize the cost function**. In the context of linear regression, the cost function used is the squared error cost function, also known as Mean Squared Error (MSE).

Suppose that you have m examples in your training set. For each example i, you have a pair of input-output x^i, y^i, where x^i is the input (feature) and y^i is the actual output (label),

You're trying to fit a linear model of the form $f_{w,b}(x) = wx + b$, where w and b are the parameters of the model.

For each input x^i, the model makes a prediction $\hat{y}^i$.

The squared error for this prediction is given by $(\hat{y}^i - y^i)^2$, which measures the square of the difference between the predicted and actual output. This squares the 'residual' or 'error' to ensure that it is positive.

> **Note**
>
> Is squared as it ensures that under-predictions and overpredictions are equally bad and avoids situations where they could cancel each other out.

Then, the total squared error cost is the sum of the squared errors over all m examples, given by $\sum_{i=1}^{m}(\hat{y}^i - y^i)^2$.

To average this over all examples,we divide by m, and for mathematical convenience, we often add a division by 2, yielding the final cost function:

$$J(w, b) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y}^i - y^i)^2$$

The more formal definition:

$$J(w, b) = \frac{1}{2m} \sum_{i=1}^{m} \left(f_{w,b}(x^i) - y^i \right)$$

3D Surface Plot   In the realm of machine learning, 3D surface plots serve as visualization tools, especially when understanding how the loss or cost function changes with respect to two parameters.

When a model has two parameters, this optimization process can be visualized in three dimensions: two for each parameter and a third for the loss.

3D Surface Plot of Loss Function
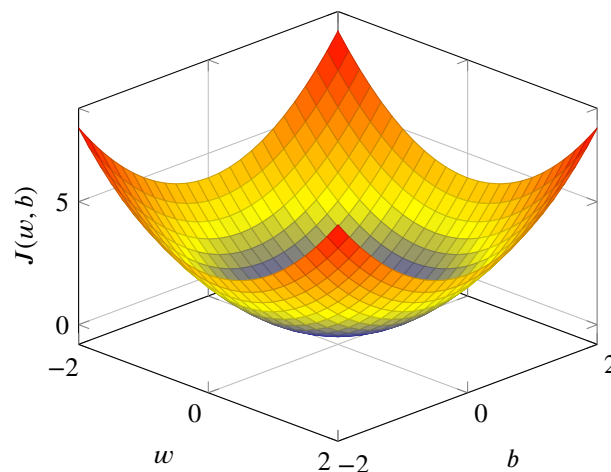
Contour Plot   A contour plot provides a 2D representation of this 3D space. It's somewhat like a top-down view of the 3D plot. In a contour plot:

- Each "contour" or line represents a level curve, where the loss function has the same value.

- Points on the same contour have the same loss.

- The spacing between contours indicates the gradient or rate of change of the loss function. If the contours are closely packed, the loss function is changing rapidly.