IBM AI Engineering Professional Certificate

# CHƯƠNG 1. Machine Learning with Python

## Giới thiệu về machine learning

Là lĩnh vực con của khoa học máy tính, tập hợp các thuật toán để giải quyết bài toán về dự đoán, phân loại.

Có các phương pháp học tập trong ML là: supervised, unsupervised, semi-supervised, self-supervised, … Trong đó có 2 phương pháp học chính và phổ thông nhất là supervised, unsupervied.

Supervised là phương pháp học có giám sát, nghĩa là khi 1 dữ liệu được đưa vào mô hình thì nó đã được gắn nhãn, hoặc có kết quả rõ ràng, ví dụ: khi đưa dữ liệu vào để máy có thể nhận diện khuôn mặt thì dữ liệu đó đã dược gắn nhãn tương ứng với tên, tuổi, …

Unsupervised là phương pháp học không giám sát, nghĩa là khi 1 dữ liệu được đưa vào mô hình thì mô hình phải cố gắng tìm ra được cấu trúc ẩn bên trong dữ liệu đó, ví dụ: đưa dữ liệu là các loài động vật thì mô hình phải cố gắng tự phân loại các loài vật.

## Bài toán của ML

### Hồi quy

Là dựa vào các thông tin có sẵn để có thể dự đoán một giá trị tiếp theo. Thuật toán phổ biến trong hồi quy là hồi quy tuyến tính.

Hồi quy tuyến tính là tìm ra mối quan hệ giữa các thông tin đầu vào với thông tin đầu ra dựa trên phương trình toán học. Có 2 phương pháp hồi quy tuyến tính là hồi quy tuyến tính đơn giản (chỉ chứa 1 tham số độc lập) và hồi quy tuyến tính bội (chứa nhiều hơn 1 tham số độc lập).

Để đánh giá một mô hình hồi quy thì thường sử dụng các số liệu sau: MAE (Mean absolute error), MSE (Mean Square Error), RSMA (Root Mean Square Error), RAE (Realative Absolute Error), RSA (Relative Square Error).

Tham số đánh giá được sử dụng nhiều nhất là MSE= 1/n \* 2 là khoảng cách bình phương trung bình giữa giá trị thực tế và giá trị mô hình dự đoán.

### Phân loại

Là nhóm các dữ liệu có điểm chung với nhau thành các nhóm. Một số thuật toán phổ biến là hồi quy logistic, k-nearest neighbor, cây quyết định, support vector machine, …

* Hồi quy logistic: là dựa vào các thông tin đã có để đưa ra một dự đoán hữu hạn nhị phân, cụ thể như có hoặc không, đúng hoặc sai, thành công hoặc thất bại, cao hoặc thấp, …

Ví dụ như ta có mối quan hệ như sau: ծ (ƟTX) 🡪 P(y=1|x)

Với mỗi giá trị theta và x, mô hình sẽ cho ra một dự đoán ŷ, và từ đó tính ra xác suất y=1 với mỗi x. Hồi quy logistic sử dụng hàm logistic (sigmoid)

Bước 1: Tạo ngẫu nhiên giá trị theta (Ɵ)

Bước 2: Với mỗi giá trị x từ dữ liệu tính toán ra ŷ (ŷ= ծ (ƟTX))

Bước 3: Tính toán mất mát giữa giá trị được tính bởi mô hình và thực tế (y - ŷ)

Bước 4: Tính toán tổng mất mát giữa mọi giá trị từ dữ liệu cho trước (J())

Bước 5: Thay đổi theta bằng giá trị khác và thực hiện lại Bước 2 đến Bước 4

Thuật toán hồi quy logistic cố gắng tìm ra các giá trị theta sao cho tổng mất mát giữa giá trị mô hình và thực tế thấp nhất (J() min). Khi đạt được đến độ chính xác yêu cầu thì mô hình ngừng lại. Để tối ưu hóa bước 5 thì có thể sử dụng gradient descent để có thể đạt hiệu suất tốt nhất.

Có thể sử dụng các phương pháp thay đổi tham số khác ngoài phương pháp gradient, mục đích của việc thay đổi tham số để tìm ra được theta phù hợp nhất, hàm mất mát giữa giá trị dự đoán và giá trị thực tế là thấp nhất. Đây cũng là bước khó và quan trọng nhất trong hồi quy logistic.

* Support Vector Machine

Đưa dữ liệu đầu vào vào các không gian đa chiều, thuật toán SVM sẽ tạo ra một mặt siêu phẳng bởi phương trình ngăn cách các dữ liệu với nhau. Với không gian 2 chiều gồm x, y thì siêu phẳng là một đường thẳng, với không gian 3 chiều thì siêu phẳng là một mặt phẳng, …

Để đưa dữ liệu vào các chiều không gian cao hơn thì sử dụng các hàm toán học khác nhau như: sigmoid, RBF, đa thức, …

Hàm SVM đạt hiệu quả tốt với các dữ liệu nhiều chiều và tối ưu hóa bộ nhớ, tuy nhiên lại không đạt được kết quả tốt với những dữ liệu lớn từ 1000 dòng và không cung cấp trực tiếp ước tính xác suất. SVM sử dụng tốt cho việc phát hiện chữ viết tay, phát hiện thư rác, phân tích dữ liệu GEN.

* K -Mean Clustering

Thuật toán này nhóm các dữ liệu dựa trên khoảng cách thành các cụm, việc xác định khoảng cách giữa các đối tượng (dữ liệu) bằng các hàm toán học khác nhau, trong đó hàm khoảng cách thông dụng nhất là euclide.

Phương pháp K -Mean sẽ đặt các vị trí k-centroid ngẫu nhiên, từ đó gom các đối tượng vào các k-centroid thành các cụm. Sau đó tính toán lại các vị trí k-centroid bằng cách tính trung bình khoảng cách của các đối tượng trong cụm. Rồi gom các đối tượng lại thành các cụm mới và thực hiện lại các bước tính toán vị trí k -centroid cho đến khi vị trí này không đổi nữa.

Nhược điểm của thuật toán này là phải xác định được trước số cụm của dữ liệu để có thể khởi tạo được bao nhiêu vị trí k -centroid, điều này là không dễ dàng.

## Các thông số để đánh giá mô hình thuật toán phân loại.

Để đánh giá một mô hình ML phân loại cần xác định các thông số sau:

Accuracy, precision, recall, f1 score, ma trận nhầm lẫn, trong đó 2 thông số thường xuyên được sử dụng là accuracy (độ chính xác), precision (độ chính xác khi kết quả dự đoán đúng).

Accuracy = (TP +TN)/ (tổng số số lượng dự đoán/phân loại)

Precision = TP/(TP+FP)

# Chương 2. Introduction to Deep Learning & Neural Networks with Keras

## 2.1 Introduction to Deep Learning

DL là một nhánh nhỏ của ML. Nó bao gồm nhiều lớp (layer), đầu vào là input layer, đầu ra là output layer, giữa 2 lớp đầu vào và đầu ra là các lớp ẩn (hidden layer). Bên trong mỗi lớp sẽ chứa các nút mạng (node). Mỗi node sẽ thực hiện 1 nhiệm vụ, đầu ra của node này sẽ là đầu vào của lớp tiếp theo.

A diagram of a network

Description automatically generated

Trước khi được đưa vào các lớp tiếp theo thì dữ liệu sẽ được đưa qua hàm kích hoạt nhằm giúp mạng có thể học được những biểu diễn phức tạp. Bởi để xử lý các vấn đề phức tạp thì sử dụng mô hình tuyến tính là chưa đủ, và thực tế việc dữ liệu phân bố rất phức tạp, sẽ cần thêm các hàm phi tuyến để có thể biểu diễn tốt hơn.

A graph of a simple and simple model

Description automatically generated

Và việc sử dụng hàm kích hoạt sẽ ngăn việc dữ liệu sau mỗi lần qua các node là rất lớn, khi các phương trình có các hàng trăm, hàng triệu tham số thì kết quả của nó có thể là dương vô cùng hoặc vô cùng. Vì vậy sử dụng hàm kích hoạt có thể giới hạn giá trị đầu tra ở một khoảng nhất định, như hàm sigmoid là (0;1), ReLU (0,x).

A diagram of a function

Description automatically generated

## 2.1 Thuật toán tối ưu

Thuật toán tối ưu là cơ sở để xây dựng mô hình neural network với mục đích "học " được các features (hay pattern) của dữ liệu đầu vào, từ đó có thể tìm 1 cặp weights và bias phù hợp để tối ưu hóa mode (các thông số cho phương trình trong node).

Với mỗi giá trị được dự đoán bởi mô hình sẽ có sự khác nhau với giá trị thực tế, khi đó sẽ gọi là mất mát. Hàm mất mát (loss function) được xây dựng để nói lên điều này. Hàm mất mát có công thức tổng quát như sau:

Trong đó y là giá trị thực tế, w\*xi là giá trị mô hình dự đoán, hàm mất mát này là phương trình bậc 2 nên nó sẽ có hình dạng là parabol.

### 2.1.1 Gradient Descent

Tìm giá trị nhỏ nhất của 1 hàm số nào đó bằng cách hàm số đạt giá trị nhỏ nhất khi đạo hàm bằng 0. Đối với các hàm số nhiều biến thì đạo hàm rất phức tạp, thậm chí là bất khả thi. Nên thay vào đó người ta tìm điểm gần với điểm cực tiểu nhất. Gradient Descent có hướng tiếp cận ở đây là chọn 1 nghiệm ngẫu nhiên cứ sau mỗi vòng lặp (hay epoch) thì cho nó tiến dần đến điểm cần tìm. Công thức: xnew = xold – lr \* gradient (x).

Trong đó lr (learning rate) là bước nhảy, gradient là độ dốc của hàm.

Ví dụ: Xét hàm số f(x)=x2+5sin(x) với đạo hàm f′(x)=2x+5cos(x). Giả sử bắt đầu từ một điểm x= -5, tại vòng lặp thứ t, chúng ta sẽ cập nhật như sau:

xnew = xold – lr \* (2xold+5cos(xold))

A graph of a function

Description automatically generated

A graph of a function

Description automatically generated

Từ hình minh họa trên ta thấy rằng ở hình bên trái, tương ứng với x0=−5, nghiệm hội tụ nhanh hơn, vì điểm ban đầu x0 gần với nghiệm x∗≈−1 hơn. Hơn nữa, với x0=5 ở hình bên phải, đường đi của nghiệm có chứa một khu vực có đạo hàm khá nhỏ gần điểm có hoành độ bằng 2. Điều này khiến cho thuật toán la cà ở đây khá lâu. Khi vượt qua được điểm này thì mọi việc diễn ra rất tốt đẹp.

Việc lựa chọn learning rate rất quan trọng trong các bài toán thực tế. Việc lựa chọn giá trị này phụ thuộc nhiều vào từng bài toán và phải làm một vài thí nghiệm để chọn ra giá trị tốt nhất. Ngoài ra, tùy vào một số bài toán, GD có thể làm việc hiệu quả hơn bằng cách chọn ra learning rate phù hợp hoặc chọn learning rate khác nhau ở mỗi vòng lặp. Tôi sẽ quay lại vấn đề này ở phần 2.

Quy tắc cần nhớ: luôn luôn đi ngược hướng với đạo hàm.