## homework: logistic regression and softmax function

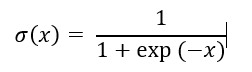
1) Tìm hiểu về cách sử dụng cross entropy để xây dựng hàm loss trong mô hình Logistic Regression (LR)

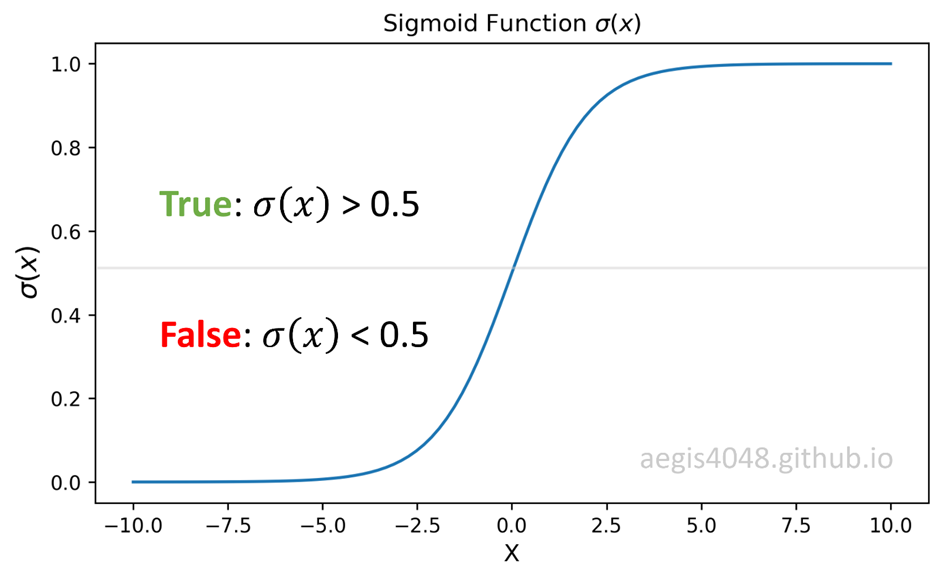
Trình bày thuật toán học cho mô hình LR

2) Tìm hiểu về hàm softmax và mô hình phân loại nhiều nhãn. Trình bày thuật toán học cho mô hình này.

## Sigmoid

Hàm sigmoid được sử dụng cho hồi quy logistic hai lớp.Nó được sử dụng để phân loại xem một mẫu nhất định là Đúng hay Sai dựa trên xác suất được tính toán. Mẫu được phân loại là True nếu giá trị lớn hơn 0,5 và ngược lại.

\

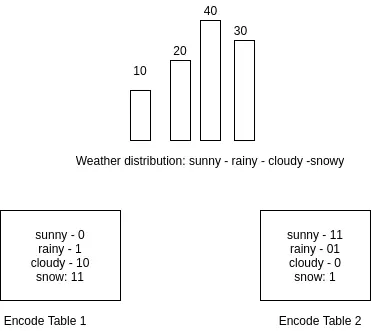


### **Entropy**

#### **Kích cỡ mã hóa trung bình**

Ta đã hiểu được cách mã hóa thông tin. Vậy làm sao để so sánh độ hiểu quả khi mã hóa bằng hai bảng mã khác nhau. Nói cách khác, làm sao để xác định được bảng mã nào tốt hơn. Hãy quan sát hình dướ

i đây.

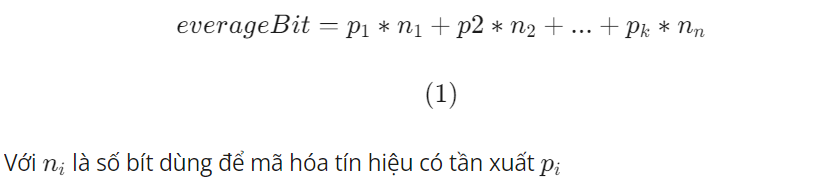


Lưu ý: khi mã hóa, các số 0 đừng trước sẽ được lược bỏ, VD: 001 -> 1, 0000100 -> 100 ...

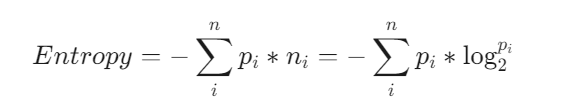
Giả sử, sau 100 lần ghi nhận thời tiết tại Tokyo, người ta thu được phân phối như hình trên với tần xuất nắng-mưa-mây-tuyết lần lượt là 10, 20, 40, 30 lần. Ta có hai bảng mã hóa như trong hình. Để so sánh độ hiểu quả của hai bảng mã trên, ta so sánh số bit trung bình được truyền đi mỗi lần truyền tin

* Bảng mã 1: everage\_bit =
* (10∗1+20∗1+40∗2+30∗2)/100=1.7
* (10∗1+20∗1+40∗2+30∗2)/100=1.7 bit.
* Bảng mã 2: everage\_bit =
* (10∗2+20∗2+40∗1+30∗1)/100=1.1
* (10∗2+20∗2+40∗1+30∗1)/100=1.1 bit

Trong lĩnh vực mã hóa, bảng mã 2 được coi là tối ưu hơn, tốt hơn bảng mã 1 vì nếu dùng bảng mã 2 sẽ tiết kiệm được số bit hơn, tốn ít băng thông (khi truyền tin) hay chiếm ít dung lượng (lưu trữ). Cách tính trên được gọi là tính kích cỡ mã hóa trung bình



công thức tính kích cỡ mã hóa trung bình tối thiểu, đây chính là công thức tính Entropy:



Entropy được tính hoàn toàn dựa vào xác suất. Giả sử với phân phối xác suất về thời tiết (mưa, nắng) là P = {p1, p2}. Người ta nhận ra Entropy(P) đạt max khi p1=p2. Khi p1 và p2 càng lệch nhau thì Entropy(P) càng giảm. Nếu p1=p2=0.5, entropy đạt max, ta rất khó để đoán thời tiết ngày mai mưa hay nắng. Nếu p1=0.1 và p2=0.9, ta sẽ tự tin đoán rằng ngày mai trời sẽ nắng, lúc này entropy có giá trị thấp hơn nhiều.

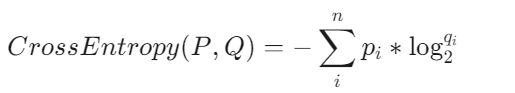
Entropy cao đồng nghĩa với việc ta khó đoán trước được sự kiện sắp xảy ra. Ta có thể gọi đó là sự bất định, sự bất ổn hay entropy là 1 thước đo sự "khó đoán" của thông tin.

## Cross entropy

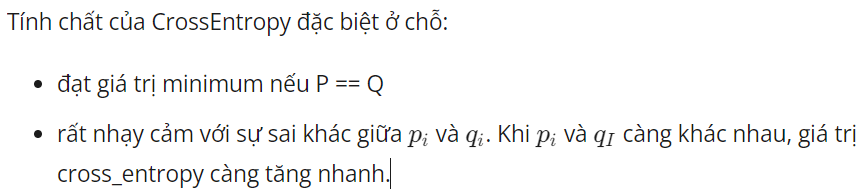
Ta cho ví dụ để hiểu hơn về cross entropy

* Đầu năm 2019, người ta thống kê thời tiết cả năm 2018 và thu được phân phối (nắng, mưa, mây, tuyết) là Q = {0.1, 0.3, 0.4, 0.2}
* Người ta dựa vào Q để lập bảng mã hóa 4 tin nhắn này cho năm tới 2019. Q được gọi là phân phối ước lượng. Tin nhắn được mã hóa và vẫn tiếp tục được truyền đi trong cả năm 2019 từ Tokyo tới New York.
* Tới cuối năm 2019, người ta thống kê lại thời tiết cả năm, thu được phân phối mới là P = {0.11, 0.29, 0.41, 0.19}. P được coi là phân phối chính xác đối với năm 2019.

Như vậy, dựa theo công thức đã chứng minh ở phần 1, số bit trung bình được dùng trong cả năm 2019 là:

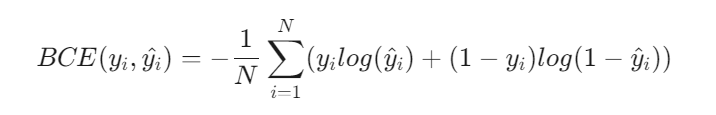


Như vậy, Entropy của tín hiệu này có phân phối P (tính kì vọng dựa vào P) nhưng lại được mã hóa dựa vào phân phối Q. Đó là lí do cái tên Cross Entropy ra đời (cross có nghĩa là *chéo nhau*, mình không biết dịch thế nào cho chuẩn).

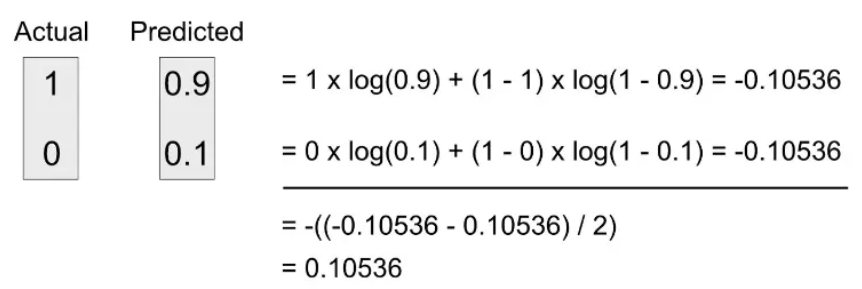


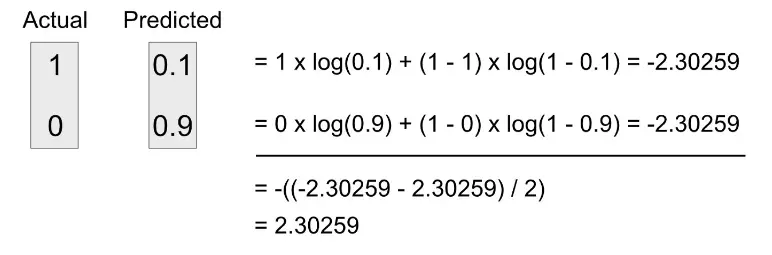
Bởi vì các bài toán machine learning thường quy về bài toán xây dựng model sao cho output càng gần target càng tốt (output, target thường đều ở dạng phân phối xác suất). Dựa vào tính *nhạy cảm* này, người ta dùng Cross Entropy để tối ưu hóa các model.

### Binary Cross Entropy Loss - BCE



Ví dụ 1: Nếu mô hình của bạn dự đoán 90% cho positive class và 10% cho negative class thì giá trị của BCE sẽ là bao nhiêu?

Ví dụ 2: Nếu mô hình của bạn dự đoán 10% cho positive class và 90% cho negative class thì giá trị của BCE sẽ là bao nhiêu?

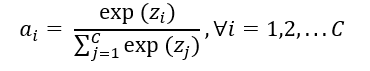


## Softmax

Softmax chuyển đổi một vector các giá trị thành một vector các xác suất, trong đó tổng tất cả các xác suất là 1. Đây là lý do tại sao Softmax thường được sử dụng ở đầu ra của các mô hình phân loại đa lớp.

Cho 1 vector đầu vào: x= [x\_1,x\_2,…,x\_d]

Công thức Softmax:

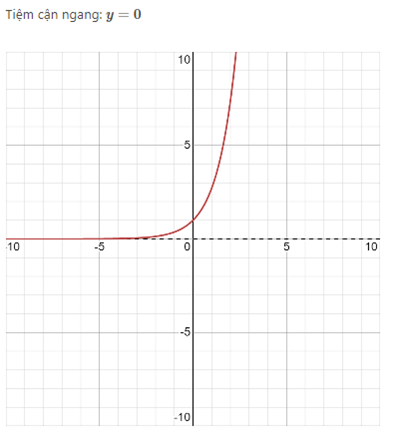


Trong đó :

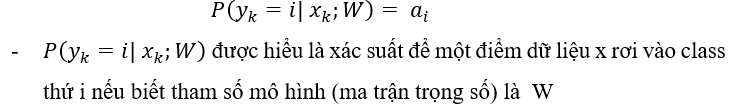
a\_i thể hiện xác suất để input đó rơi vào class i, với lớn hơn 0 và có tổng bằng 1

z\_i= w\_i^T x, càng lớn thì xác suất dữ liệu rơi vào class i càng cao. z\_i có thể nhận giá trị âm hoặc dương.

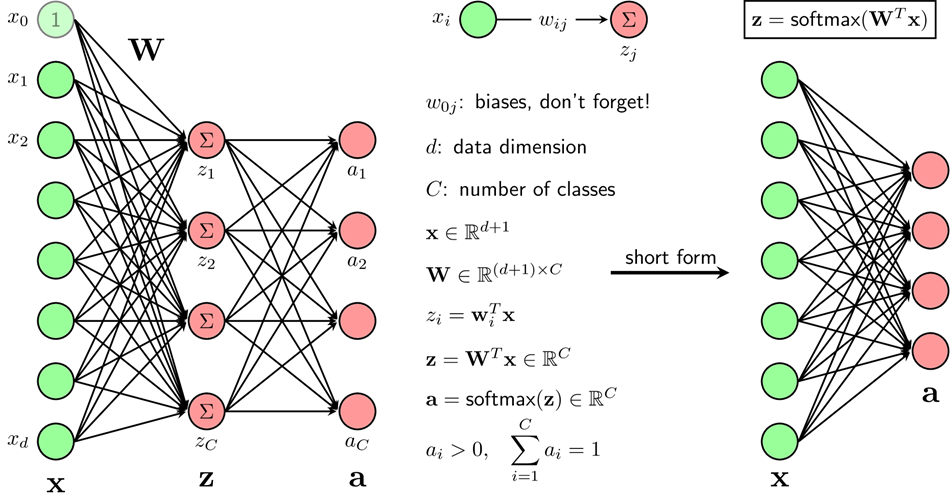
exp⁡(z\_i) là một hàm mượt, có nghĩa là nó liên tục và khả vi vô hạn. Nghĩ là z\_i là 1 giá trị dương thì sẽ trở thành 1 hàm đồng biến(đơn điệu tăng) với z\_i tăng thì exp⁡(z\_i) tăng.



Ta có thể viết thành



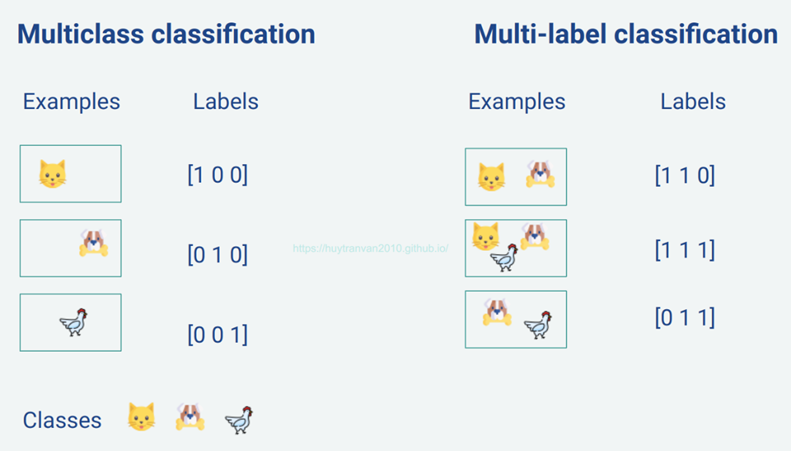
Hình vẽ dưới đây thể hiện mạng Softmax Regression dưới dạng neural network:



## Bài toán đa nhãn: Trong phân loại nhiều lớp, mỗi phiên bản (ví dụ: tài liệu, hình ảnh, điểm dữ liệu) thuộc về chính xác một lớp trong một tập hợp các lớp loại trừ lẫn nhau. Điều này có nghĩa là đối với một thể hiện nhất định, mô hình phải chọn một và chỉ một lớp từ nhiều lớp có thể.

Ví dụ:Phân loại các loại trái cây: táo, chuối, cam,… Mỗi loại trái cây thuộc một và chỉ một loại.

Bài toán phân loại văn bản nhiều nhãn: Đây là bài toán trong lĩnh vực phân loại văn bản trong đó một tài liệu đầu vào có thể thuộc nhiều nhãn. Ví dụ: một tờ báo trực tuyến có thể được phân loại là thông tin chính trị, xã hội hoặc cả hai.



Với phương pháp này chúng ta cố gắng chuyển đổi multi-label problem về single-label problem. Phương pháp này có thể thực hiện qua một số cách sau:

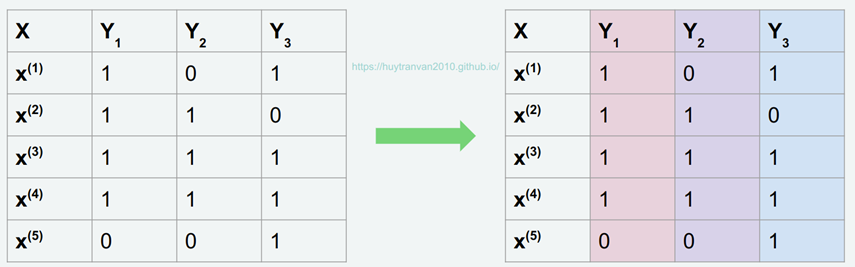
- Binary Relevance

- Classifier Chains

- Label Powerset

### Binary Relevance

Trong binary relevance, chúng ta sẽ chia bài toán thành 3 bài toán phân loại theo từng lớp đơn (do có 3 lớp ở đây). Từ đó, chúng ta có thể dễ dàng xử lý từng bài toán riêng lẻ và sau đó kết hợp kết quả cuối cùng lại với nhau. Và sử dụng Thư viện Scikit-multilearn



### Classifier Chains

Chuỗi phân loại (CC): Sắp xếp các nhãn theo một thứ tự cụ thể và mỗi nhãn sẽ được dự đoán dựa trên các nhãn trước đó trong chuỗi.

o Dữ liệu gốc:

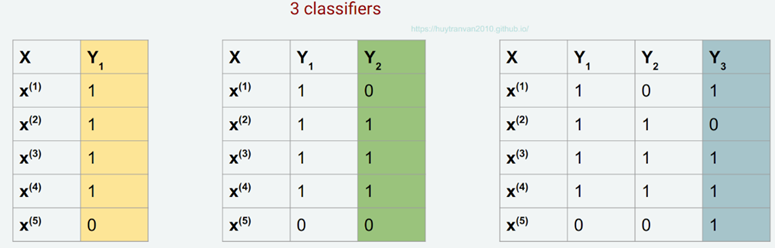


Trong phân loại chuỗi, chúng tôi cũng có 3 bảng toán nhãn đơn tương ứng với 3 lớp. Với:

o Phân loại 1: Đặc điểm: X —- mục tiêu: Y1

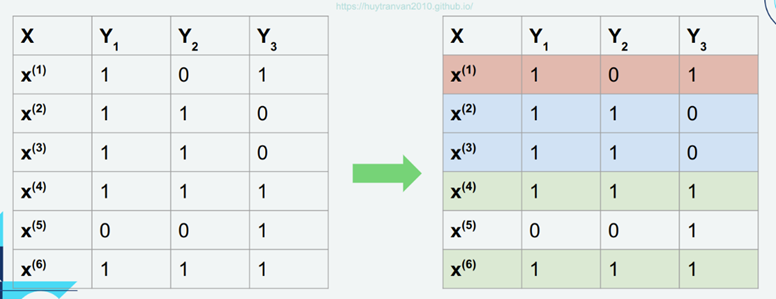
o Phân loại 2: Đặc điểm: X, Y1 —- mục tiêu: Y2

o Phân loại 2: Đặc điểm: X, Y1, Y2 —- mục tiêu: Y3



### Label Powerset

Label Powerset (LP): Xử lý mọi tổ hợp nhãn duy nhất dưới dạng một lớp duy nhất, chuyển vấn đề thành vấn đề nhiều lớp.



o Như thể hiện trong hình trên, các điểm dữ liệu x2 và x3 được coi là có cùng nhãn,

o x4 và x6 được coi là có cùng nhãn. x1 có nhãn riêng, x5 có nhãn riêng.

o Tại thời điểm này, chúng tôi có tổng cộng 4 lớp mới cho mô hình mới. Mô hình này phục vụ cho các vấn đề phân loại nhiều lớp. Người ta nhận thấy rằng bài toán phân loại đa nhãn hiện đã được chuyển thành bài toán phân loại nhiều lớp.