REGRESSION LOSS FUNCTIONS

1. MEAN ABSOLUTE ERROR, L1 LOSS:

Mean Absolute Error (MAE) hay còn được gọi là L1 Loss là một loss function được sử dụng cho các mô hình hồi quy, đặc biệt cho các mô hình hồi quy tuyến tính. MAE được tính bằng tổng các trị tuyệt đối của hiệu giữa giá trị thực (y_i: target) và giá trị mà mô hình của chúng ra dự đoán ($\overline{y_i}$: predicted).

$$MAE = \frac{\sum_{i=1}^{n} \left| y_i - \overline{y_i} \right|}{n}$$

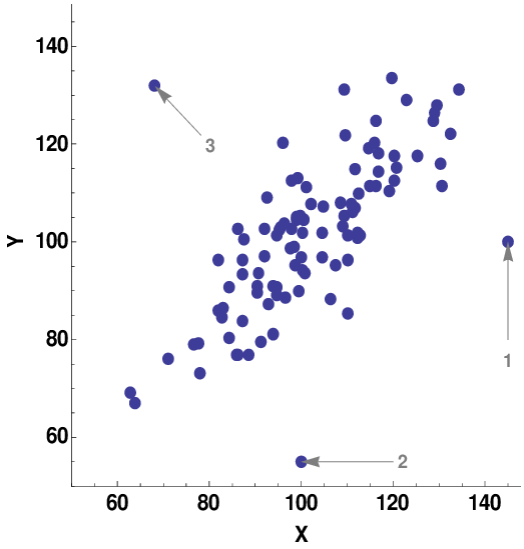
2. MEAN SQUARE ERROR, L2 LOSS:

Mean Square Error (MSE) hay còn được gọi là L2 Loss là một loss function cũng được sử dụng cho các mô hình hồi quy, đặc biệt là các mô hình hồi quy tuyến tính. MSE được tính bằng tổng các bình phương của hiệu giữa giá trị thực (y_i: target) và giá trị mà mô hình của chúng ra dự đoán ($\overline{y_i}$: predicted).

$$MSE = \frac{\sum_{i=1}^{n} (y_i - \overline{y_i})^2}{n}$$

SO SÁNH MAE VÀ MSE (L1 LOSS VÀ L2 LOSS):

a/ Để tìm điểm cực tiểu của các hàm số, thông thường cách đơn giản nhất chúng ta nghĩ đến chính là tìm đạo hàm của hàm số rồi tìm điểm mà tại đó đạo hàm của hàm số bằng 0. Như vậy sẽ có 1 bước đạo hàm, hiển nhiên tìm đạo hàm của MSE sẽ đơn giản hơn rất nhiều so với MAE. Tuy nhiên, MAE thì đem lại kết quả tốt hơn đối với các dữ liệu có outlier.



Hình 1: Dữ liệu bị outlier

Đầu tiên hãy để ý đến MSE, ta có $y_i - \overline{y_i} = e$, e^2 sẽ càng lớn nếu e > 1. Nếu bậc của e càng lớn thì giá trị hàm Loss cũng càng lớn hơn. Vì vậy nếu như chúng ta có 1 outlier trong bộ dữ liệu, giá trị của e lớn, thì e^2 càng lớn, khi đó giá trị MSE sẽ lớn.

Ngược lại với MSE, nếu ta có giá trị e lớn (e > 1) thì |e| vẫn sẽ lớn, nhưng hiển nhiên nhỏ hơn nhiều so với e^2.

Do đó khi tối ưu loss function, L2 sẽ bị ảnh hưởng nhiều hơn với các điểm outliers và model sẽ bị kéo về phía outliers hơn. Do đó MSE bị ảnh hưởng bởi outlier và L1 tốt hơn đối với các dữ liệu có outlier.

Để dễ hiểu hơn tại sao MAE và MSE có sự khác nhau như vậy với các outlier, hãy cùng làm 1 ví dụ đơn giản sau: Giả sử mình có 4 điểm dữ liệu 1, 2, 4, 33. Mình cùng tìm thử nghiệm theo L1 và L2 nhé. Tìm min của

a) $L1 = |x-1| + |x-2| + |x-4| + |x-33|$
+) $-\infty < x \leq 1$: L1 = 40 - 4x đạt min = 36 tại x = 1
+) $1 < x \leq 2$: L1 = 38 - 2x đạt min = 34 tại x = 2
+) $2 < x \leq 4$: L1 = 34
+) $4 < x \leq 33$: L1 = 2x + 26 > 34
+) $33 < x \leq +\infty$: L1 = 4x - 40 đạt min > 4 * 33 - 40 = 92
Vậy L1 đạt min = 34 tại $2 \leq x \leq 4$

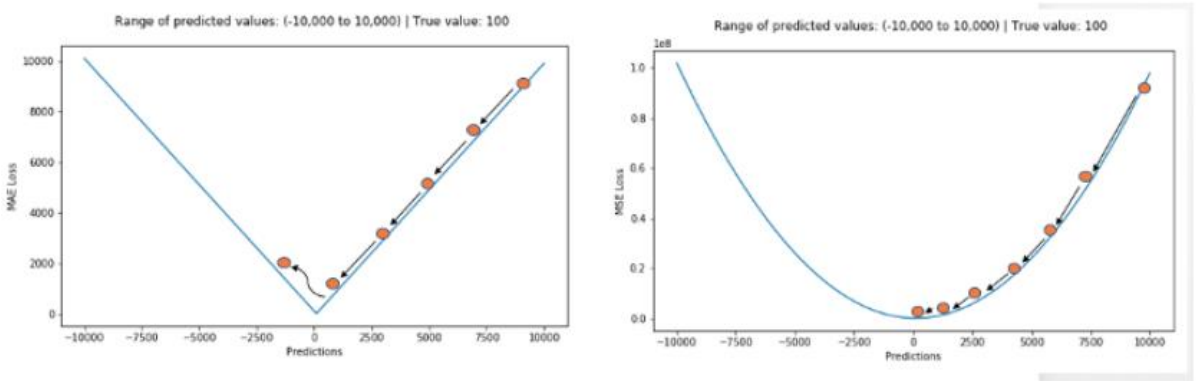
b) $L2 = (x-1)^2 + (x-2)^2 + (x-4)^2 + (x-33)^2$
$L2' = 2 * (x-1+x-2+x-4+x-33) = 0 \Leftrightarrow x = 10$
L2 đạt min = 710 tại x = 10.

Trong dãy số 1, 2, 4, 33 thì có 33 là giá trị outlier, L2 đạt min lớn hơn rất nhiều so với L1. Và để ý hơn thì thấy L2 đạt min tại trung bình (mean) của các giá trị $\frac{1+2+4+33}{4} = 10$, còn L1 đạt min tại trung vị (median) của các giá trị đó.

=> Do đó L1 sẽ tốt hơn với dữ liệu có các outliers.

b/ Một vấn đề lớn trong việc sử dụng MAE (đặc biệt đối với mạng nơ ron) là gradient của nó giống nhau, nghĩa là gradient sẽ lớn ngay cả đối với các giá trị loss nhỏ. Điều này không tốt cho việc học. Để khắc phục điều này, chúng ta có thể sử dụng dynamic learning rate, giá trị này sẽ giảm khi chúng ta tiến gần đến cực tiểu.

MSE hoạt động tốt trong trường hợp này và sẽ hội tụ ngay cả với learning rate cố định. Gradient của MSE cao đối với các giá trị loss lớn hơn và giảm khi loss tiến về 0, làm cho nó chính xác hơn vào cuối quá trình huấn luyện (xem Hình 2).



Hình 2:

c/ Vấn đề cho cả MAE và MSE: Có thể có những trường hợp mà cả hàm mất mát đều không đưa ra dự đoán mong muốn. Ví dụ: nếu 90% quan sát trong dữ liệu của chúng ta có giá trị mục tiêu thực là 150 và 10% còn lại có giá trị mục tiêu trong khoảng 0–30. Khi đó, một mô hình với MAE có thể dự đoán 150 cho tất cả các quan sát, bỏ qua 10% các trường hợp ngoại lệ, vì nó sẽ cố gắng hướng tới giá trị trung vị. Trong trường hợp tương tự, một mô hình sử dụng MSE sẽ đưa ra nhiều dự đoán trong phạm vi từ 0 đến 30 vì nó sẽ bị lệch về phía outliers.

Làm gì trong trường hợp này? Một cách khắc phục dễ dàng là chuyển đổi các biến mục tiêu. Một cách khác là thử một hàm loss khác: Huber loss.

3. HUBER LOSS:

Huber loss ít nhạy hơn với các giá trị outliers trong dữ liệu so với MSE. Nó cũng khả vi tại 0. Về cơ bản, đó là sai số tuyệt đối, trở thành bậc hai khi sai số nhỏ. Sai số đó phải nhỏ đến mức nào để biến nó thành bậc hai phụ thuộc vào một siêu tham số δ (delta), mà có thể được điều chỉnh.

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for} |y - f(x)| \le \delta, \\ \delta\,|y - f(x)| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$

Việc lựa chọn delta là rất quan trọng vì nó xác định những gì bạn coi là outlier. Phần lớn hơn delta được sử dụng L1 (ít nhạy hơn với các outliers lớn), trong khi phần nhỏ hơn delta được sử dụng một cách “thích hợp” với L2.

Chứng mình hàm L_δ liên tục trên miền xác định R ?

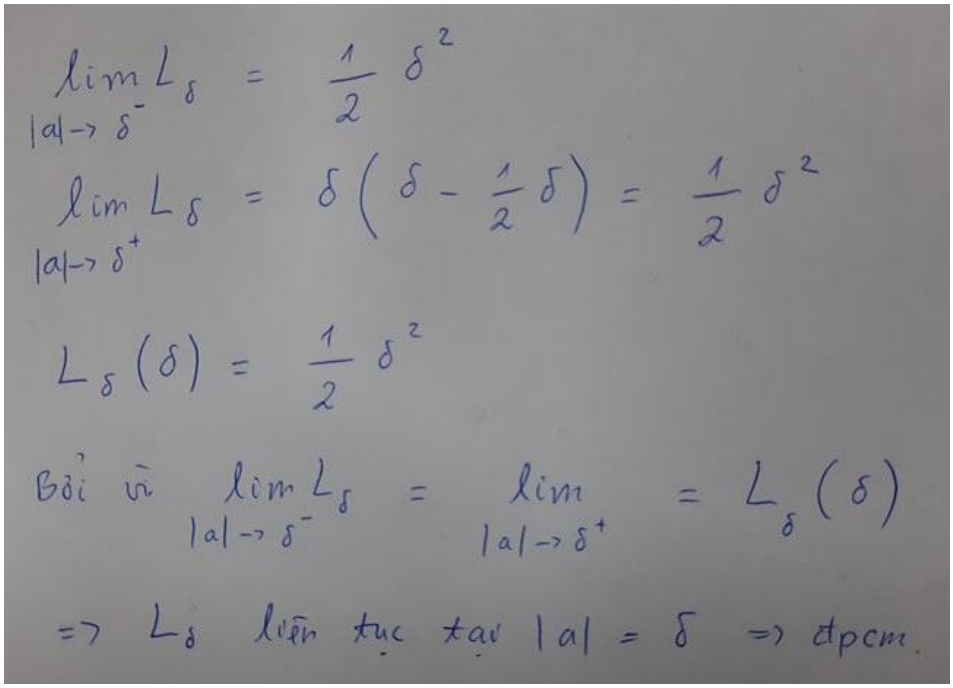
Lời giải:

Đặt $a = y - f(x)$, khi đó:

$$L_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \le \delta, \\ \delta(|a| - \frac{1}{2}\delta), & \text{otherwise.} \end{cases}$$

Ta chỉ cần chứng minh L_δ liên tục tại $|a| = \delta$:

Ta có:



Tại sao sử dụng Huber Loss?

- Một vấn đề lớn khi sử dụng MAE để huấn luyện mạng nơron là độ dốc liên tục lớn của nó, có thể dẫn đến bỏ lỡ cực tiểu khi kết thúc quá trình huấn luyện bằng cách sử dụng gradient descent. Đối với MSE, gradient giảm khi loss gần với cực tiểu của nó, làm cho nó chính xác hơn.

- Huber loss có thể thực sự hữu ích trong những trường hợp như vậy, vì nó cong xung quanh cực tiểu làm giảm gradient. Và nó tốt hơn đối với outliers so với MSE. Do đó, nó kết hợp các đặc tính tốt từ cả MSE và MAE. Tuy nhiên, vấn đề với Huber loss là chúng ta có thể cần phải train siêu tham số delta, đó là một quá trình lặp đi lặp lại.