

Deep learning week 2: NEURAL NETWORK

Nguyễn Tuấn Duy - 11204971

September 2023

1 Bài 3: Tại sao hàm softmax lại dùng exponential function (e) mà không phải số khác như 2, 3 hoặc 4?

phương trình softmax có dạng:

$$\sigma(z)_i = \frac{\exp(z_i)}{\sum_{j=1}^K \exp(z_j)}$$

1) Đạo hàm đẹp:

Khi dùng exponential function thì phương trình sẽ dễ đạo hàm hơn khi mà đạo hàm của e^x chỉ là:

$$\frac{\delta}{\delta x} e^x = e^x$$

Trong khi đó khi thay e bằng một số tự nhiên bất kì thì phép đạo hàm sẽ phức tạp hơn:

$$\frac{\delta}{\delta x} a^x = a^x \ln(a)$$

2) Những số gần max sẽ có giá trị gần với hardmax:

Hàm hardmax sẽ biểu diễn input của mình dưới dạng one-hot với giá trị cao nhất là 1 còn các thành phần còn lại là 0. Ví dụ tập [2, 4, 2, 1] sẽ được biểu diễn thành [0, 1, 0, 0]. Tuy nhiên, hàm này lại không đạo hàm được.

Còn với softmax tập giá trị sẽ được biểu diễn thành [0.102, 0.757, 0.102, 0.04]

Khi thay e bằng 2 thì ta sẽ có được [0.15, 0.61, 0.15, 0.07] giá trị lớn nhất 0.61 xa 1 hơn 0.757 của softmax

Tuy nhiên, hàm này khi thay e bằng số cực lớn thì max sẽ gần 1 nhất nhưng các thành phần khác sẽ rất bé và khó tính toán khi mà giới hạn phần thập phân lớn nhất của numpy là float64 (10^{308} chữ số thập phân)

3) Có thể chuyển sang dạng xác suất với log-likelihood:

Trong bài toán classification, ta cần tìm:

$$\hat{c} = \operatorname{argmax}_i P(C = c_i | x)$$

Ta có:

$$\begin{aligned} P(C = c_i|x) &= \frac{P(x|C = c_i)P(C = c_i)}{\sum_{j=1}^N P(x|C = c_j)P(C = c_j)} \\ &= \frac{\exp(\ln(P(x|C = c_i)P(C = c_i)))}{\sum_{j=1}^N \exp(\ln(P(x|C = c_j)P(C = c_j)))} \\ &= \frac{\exp(z_i)}{\sum_{j=1}^N \exp(z_j)} \end{aligned}$$

Trong đó:

$$\ln(P(x|C = c_i)P(C = c_i)) = z_i$$

Như vậy output của hàm soft max khi dùng với e có thể hiểu như là một dạng xác suất để chọn class c khi biết dữ liệu x. Điều này rất quan trọng trong bài toán classification khi mà chúng ta không chỉ cần biết x_i thuộc nhóm nào trong c mà còn cần phải tìm xác suất bao nhiêu phần trăm x_i thuộc class c_i .