The problem

The problem statement is: can we find a way to compute the softmax of a vector without going through the vector 3 times, but also preventing the exponential from exploding?

$$x = [3, 2, 5, 1]$$

Pseudocode:

$m_0 = -\infty$
for $i = 1$ to N
  $m_i = \max(m_{i-1}, x_i)$
$\ell_0 = 0$
for $J = 1$ to N
  $\ell_J = \ell_{J-1} + e^{x_J - m_N}$
for $K = 1$ to N
  $x_K \leftarrow \dfrac{e^{x_K - m_N}}{\ell_N}$

Let's try fusing the computation of the max element with the normalization constant

$$\text{softmax}(x_i) = \frac{e^{x_i - x_{max}}}{\sum_{j=1}^{N} e^{x_j - x_{max}}}$$

$$X = [3, \ 2, 5, \ 1]$$

STEP 1:

$max_1 = 3$

$\ell_1 = e^{3-3}$

STEP 2

$max_2 = max(3, 2) = 3$

$\ell_2 = \ell_1 + e^{2-3}$

If our vector only contained the first two elements, then the max element and the normalization factor we've computed would be correct.

However, things change at position 3...

$$x = [3,\ 2,\ 5,\ 1]$$

STEP 3

$$\max_3 = \max(3, 5) = 5$$

$$\ell_3 = \ell_2 + e^{5-5} = e^{3-3} + e^{2-3} + e^{5-5}$$

The ℓ_3 we computed is wrong!

$$\ell_3 = \underbrace{e^{3-3} + e^{2-3}}_{\text{Here it is wrong!}} + e^{5-5}$$

Can we fix it ON THE FLY? YES!

$$\ell_3 = \ell_2 \cdot e^{3-5} + e^{5-5} = \left(e^{3-3} + e^{2-3}\right) e^{3-5} + e^{5-5}$$

correction factor

$$= e^{3-3+3-5} + e^{2-3+3-5} + e^{5-5} = \underbrace{e^{3-5} + e^{2-5} + e^{5-5}}_{\text{CORRECT!}}$$

So, every time we encounter a number bigger than the current maximum, we can "fix" the normalization constant computed so far!

$$x = [3, 2, 5, 1]$$

STEP 4

$$max_4 = max(5, 1) = 5$$

$$l_4 = l_3 \cdot e^{5-5} + e^{1-5}$$

IN THIS CASE WE DO NOT NEED TO FIX ANYTHING

New pseudocode

$m_0 = -\infty$

$l_0 = 0$

for $i = 1$ to N

$\quad m_i = max(m_{i-1}, x_i)$

$\quad l_i = l_{i-1} \cdot e^{m_{i-1} - m_i} + e^{x_i - m_i}$

for $k = 1$ to N

$\quad x_k \leftarrow \dfrac{e^{x_k - m_N}}{l_N}$

Can we prove this algorithm is correct?

Let's do it!

$m_0 = -\infty$

$\ell_0 = 0$

for $i = 1$ to N

$m_i = \max(m_{i-1}, x_i)$

$\ell_i = \ell_{i-1} \cdot e^{m_{i-1} - m_i} + e^{x_i - m_i}$

for $k = 1$ to N

$x_k \leftarrow \dfrac{e^{x_k - m_N}}{\ell_N}$

We want to prove that at the end of this loop:

$m_N = \max_i (x_i) = x_{MAX}$

$\ell_N = \sum_{j=1}^{N} e^{x_j - x_{MAX}}$

We will prove it by induction:

1) Prove that it holds for a vector of size $N=1$

$$m_1 = \max(-\infty, x_1) = x_1 = \max_i (x_i) = X_{MAX}$$

$$\ell_1 = 0 \times e^{-\infty} + e^{x_1 - x_1} = \sum_{J=1}^{N} e^{x_i - x_{MAX}}$$

$\blacksquare$

2) If we assume it holds for a vector of size N, does it hold for a vector of size $N+1$?

$$m_{N+1} = \max(m_N, x_{N+1}) = \max_i (x_i)$$

$$\ell_{N+1} = \ell_N e^{m_N - m_{N+1}} + e^{x_{N+1} - m_{N+1}} =$$

$$= \left(\sum_{J=1}^{N} e^{x_J - m_N}\right) e^{m_N - m_{N+1}} + e^{x_{N+1} - m_{N+1}}$$

$$= \sum_{J=1}^{N} e^{x_J - m_{N+1}} + e^{x_{N+1} - m_{N+1}}$$

$$= \sum_{J=1}^{N+1} e^{x_J - m_{N+1}}$$

$\blacksquare$