

Argmax and Softmax

Argmax:

It simply converts the highest output to 1 and convert rest of the outputs to 0.

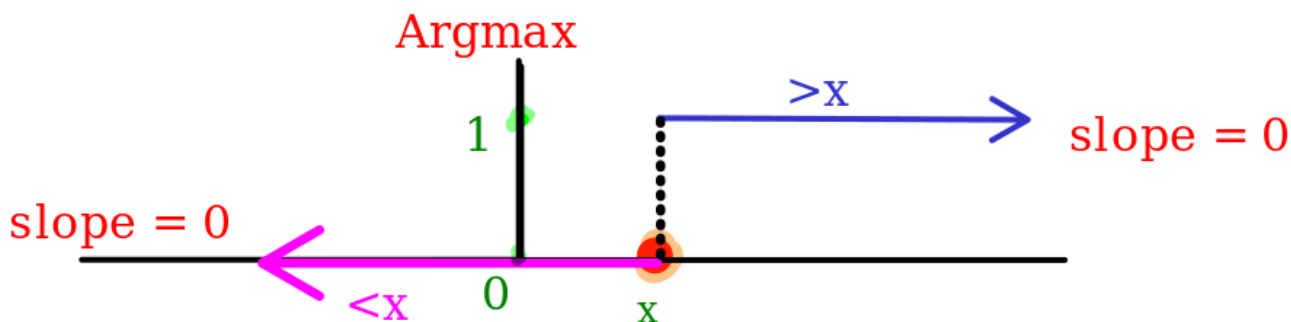
for example, $\text{argmax}([0.23, 0.99, -102, 10]) \Rightarrow [0, 0, 0, 1]$

i.e., it just converted the maximum number to 1 and rest of the outputs to 0.

The only problem is that we can't use argmax to optimize weights and biases since its derivative is always = 0 (as it produces a constant output always 0 or 1).

for eg:

let x be the second largest observation in the outputs.



Any number greater than x is the largest number and is converted to 1, on the other hand, rest of the observations are converted to 0. This shows that both the lines have $\text{slope} = 0$, and hence, their derivative is also 0.



...then we would end up plugging 0 into **The Chain Rule** for the derivative of **ArgMax**...

$$\frac{d \text{ Loss Function}}{d \text{ Some Parameter}} = \frac{d \text{ Loss Function}}{d \text{ ArgMax}} \times \frac{d \text{ ArgMax}}{d \text{ Raw Output}} \times \dots$$



The whole derivative would become 0, and hence Gradient descent won't proceed.

Softmax:

SoftMax function uses the formulae:

$$\text{softmax}_{(\text{output value})_i} = \frac{e^{(\text{output value})_i}}{\sum_{j=1}^n e^{(\text{output value})_j}}$$

here,

- i is one of the output values
- the denominator is the sum of exponentials of all output values.

In simple manner:

$$e^x / (e^x + e^y + e^z)$$



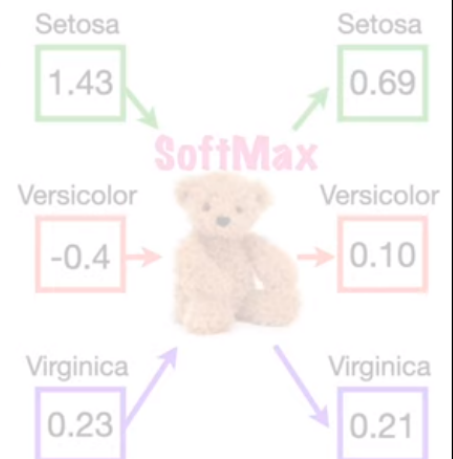
...the derivative of the **SoftMax** function is not always **0** and we can use it for **Gradient Descent**.

$$\text{SoftMax}_{\text{Setosa}}(\text{Output Values}) = \frac{e^{\text{Setosa}}}{e^{\text{Setosa}} + e^{\text{Versicolor}} + e^{\text{Virginica}}}$$

$$\frac{d \text{ "p}_{\text{Setosa}}"}{d \text{ Raw}_{\text{Setosa}}} = \text{ "p}_{\text{Setosa}} \times (1 - \text{ "p}_{\text{Setosa}}) = 0.21$$

$$\frac{d \text{ "p}_{\text{Setosa}}"}{d \text{ Raw}_{\text{Versicolor}}} = -\text{ "p}_{\text{Setosa}} \times \text{ "p}_{\text{Versicolor}} = -0.07$$

$$\frac{d \text{ "p}_{\text{Setosa}}"}{d \text{ Raw}_{\text{Virginica}}} = -\text{ "p}_{\text{Setosa}} \times \text{ "p}_{\text{Virginica}} = -0.15$$



The derivatives are obtained by simple **Quotient Rule of Derivatives**.

Quotient Rule:

$$\frac{dy}{dx} = \frac{u'v - uv'}{v^2}$$

where, $y = \frac{u}{v}$; $u' = \frac{du}{dx}$; $v' = \frac{dv}{dx}$

Conclusion:

- Neural Networks with multiple outputs use **SOFTMAX** for **Training**.
 - And, use **ARGMAX**, which has easy to understand outputs, to make classification of new observations.
-

Reference links:

1. <https://www.youtube.com/watch?v=KpKog-L9veg&list=PLblh5JKOoLUlxGDQs4LFFD--41Vzf-ME1&index=9>
2. https://medium.com/@s_hash_wat/argmax-and-softmax-496714956aab