# Softmax function in output layer

## Reference

https://en.wikipedia.org/wiki/Softmax_function

## Definition

the softmax function, also known as softargmax or normalized exponential function, is a function that **takes as input a vector of K real numbers, and normalizes it into a probability distribution consisting of K probabilities proportional to the exponentials of the input numbers.**

## Motivation

Prior to applying softmax, some vector components could be negative, or greater than one; and might not sum to 1; but after applying softmax, each component will be in the interval $(0,1)$, and the components will add up to 1, so that they can be interpreted as probabilities.

Furthermore, the larger input components will correspond to larger probabilities.

## Applications

Softmax is often used in neural networks, to map the non-normalized output of a network to a probability distribution over predicted output classes.

## Formula

The standard (unit) softmax function $\sigma: R^{K} \rightarrow R^{K}$ is defined by the formula $$ \sigma(z)i = \frac{e^{z_i}}{\sum{j = 1}^{K}e^{z_j}} $$ where $i = 1, 2, ..., K$ and $z = (z\_1, z\_2, z\_3, ..., z\_K) \in R^{K}$.

## Neural Network

Generally, it is written as

$$ P(y = j|x) = \frac{\exp(z_j)}{\sum_{k=1}^{K} \exp(z_k)} $$

where $j$ is the $j^{th}$ class and $x$ is the input sample.

$$ P(y=j|x) = \frac{\exp(z_j)-x^}{x^ + log(exp(x_1 - x^)) + ... + log(exp(x_n - x^))} $$

## Multiclass cross-entropy loss function

$$

J = \frac{1}{m}\sum_{i=1}^{m}\sum_{c=1}^{C}[-y_{(c)}\log P(y_{(c)} | x^{(i)})]

$$ where C is the ground true label, m is the number of input samples, and $y_{(c)}$ is the corresponding ground true.

## Example

Classification：

- Dog: class 1;
- Cat: class 2;
- Baby: class 3.
- Tiger: class 4.

A softmax output layer of a neural network is a 4 by 1 matrix.

The layer before a softmax, for instance, could be:

$$z = \begin{bmatrix} 5 & 2 & -1 & 3 \end{bmatrix}$$

Then, the softmax would be:

$$\sigma = \begin{bmatrix} 0.842 & 0.042 & 0.002 & 0.114 \end{bmatrix}$$

The ground true label may be:

$$y_{(c)} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

## log-sum-exp operation

https://en.wikipedia.org/wiki/LogSumExp

Softmax cross entropy loss involves the log-sum-exp operation. This can result in numerical underflow/overflow. Read about the solution in the link, and try to understand the calculation of loss in the code.

$$ LSE(x_1, x_2, ..., x_n) = x^* + log(exp(x_1 - x^)) + \cdot \cdot \cdot + log(exp(x_n - x^)) $$ where $x^* = max\{x_1, x_2, ... ,x_n\}$.

## Derivative of softmax

$$P(y = j|x) = \frac{\exp(z_j)}{\sum_{k=1}^{K} \exp(z_k)}$$

$$\frac{\partial P}{\partial z_j} = \frac{exp(z_j)' \sum_{k=1}^{K} \exp(z_k) - [\sum_{k=1}^{K} \exp(z_k)]' exp(z_j)}{[\sum_{k=1}^{K} \exp(z_k)]^2}$$

$$= \frac{exp(z_j)(\sum_{k=1}^{K} \exp(z_k) - exp(z_j))}{[\sum_{k=1}^{K} \exp(z_k)]^2}$$

Therefore,

$$\frac{\partial P}{\partial z_j} = P(y = j|x)(1 - P(y = j|x))$$