# What Is the Softmax Function?

Nicholas J. Higham*

January 12, 2021

The softmax function takes as input a real $n$-vector $x$ and returns the vector $g$ with elements given by

$$g_j(x) = \frac{\mathrm{e}^{x_j}}{\sum_{i=1}^{n} \mathrm{e}^{x_i}}, \quad j = 1\colon n. \qquad (*)$$

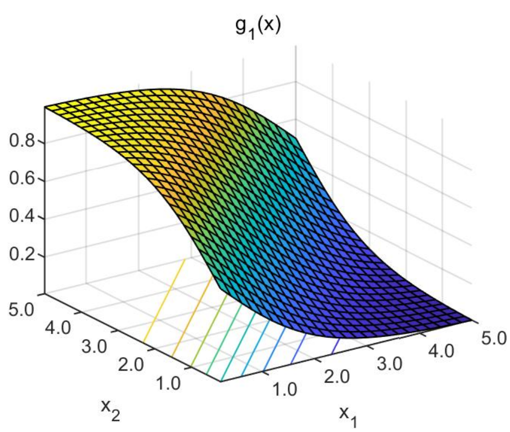
It arises in machine learning, game theory, and statistics. Since $\mathrm{e}^{x_j} \ge 0$ and $\sum_{j=1}^{n} g_j = 1$, the softmax function is often used to convert a vector $x$ into a vector of probabilities, with the more positive entries giving the larger probabilities.

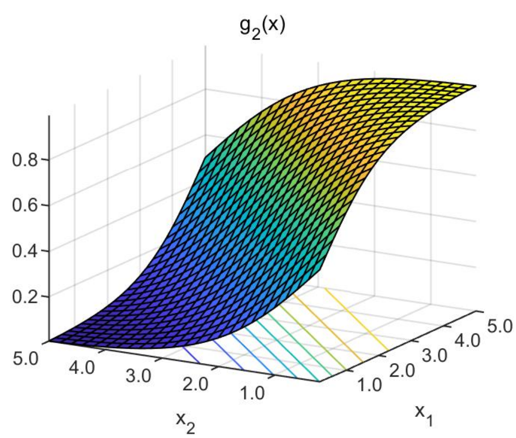
The softmax function is the gradient of the log-sum-exp function

$$\mathrm{lse}(x) = \log \sum_{i=1}^{n} \mathrm{e}^{x_i},$$

where log is the natural logarithm, that is, $g_j(x) = (\partial/\partial x_j)\mathrm{lse}(x)$.

The following plots show the two components of softmax for $n = 2$. Note that they are constant on lines $x_1 - x_2 = \text{constant}$, as shown by the contours.




Here are some examples:

```
>> softmax([-1 0 1])
ans =
   9.0031e-02
```

*Department of Mathematics, University of Manchester, Manchester, M13 9PL, UK (nick.higham@manchester.ac.uk).

```
   2.4473e-01
   6.6524e-01
>> softmax([-1 0 10])
ans =
   1.6701e-05
   4.5397e-05
   9.9994e-01
```

Note how softmax increases the relative weighting of the larger components over the smaller ones. The MATLAB function `softmax` used here is available at `https://github.com/higham/logsumexp-softmax`.

A concise alternative formula, which removes the denominator of $(*)$ by rewriting it as the exponential of $\mathrm{lse}(x)$ and moving it into the numerator, is

$$g_j = \exp\bigl(x_j - \mathrm{lse}(x)\bigr). \qquad (\#)$$

Straightforward evaluation of softmax from either $(*)$ or $(\#)$ is not recommended, because of the possibility of overflow. Overflow can be avoided in $(*)$ by shifting the components of $x$, just as for the log-sum-exp function, to obtain

$$g_j(x) = \frac{\mathrm{e}^{x_j - \max(x)}}{\sum_{i=1}^{n} \mathrm{e}^{x_i - \max(x)}}, \quad j = 1\colon n. \qquad (\dagger)$$

where $\max(x) = \max_i x_i$. It can be shown that computing softmax via this formula is numerically reliable. The shifted version of $(\#)$ tends to be less accurate, so $(\dagger)$ is preferred.

## References

This is a minimal set of references, which contain further useful references within.


- Pierre Blanchard, Desmond J. Higham, and Nicholas J. Higham, Accurately Computing the Log-Sum-Exp and Softmax Functions, IMA J. Numer. Anal., Advance access, 2020.
- Bolin Gao and Lacra Pavel, On the Properties of the Softmax Function with Application in Game Theory and Reinforcement Learning, ArXiv:1209.5145, 2018.
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press, 2016.


## Related Blog Posts

- What is Numerical Stability? (2020)
- What is the Log-Sum-Exp Function? (2020)

This article is part of the “What Is” series, available from `https://nhigham.com/category/what-is` and in PDF form from the GitHub repository `https://github.com/higham/what-is`.