**Introduction**

The softmax function, also known as softargmax or normalized exponential function, is, in simple terms, more like a normalization function, which involves adjusting values measured on different scales to a notionally common scale.

Simplest form of normalization called ‘hard-max’: Assume that our classification model has returned three values- 3, 7 and 14 as an output, and we want to assign probabilities or label them. The easiest possible way is to assign a 100% probability to the highest score and 0% to everything else, i.e., 14 would get a 100% probability score. In contrast, both 3, 7 would get probability scores of 0% each. Although simple, this method is relatively crude and does not consider the scores of other variables and their scales.

Next, consider the conventional normalization done by taking the ratio of the score to the sum of all scores. In the same model outputs- 3, 7 and 14 our probabilities would be 3/ (3+7+14) = 0.13, be 7/ (3+7+14) = 0.29 and be 14/ (3+7+14) = 0.58. Although this method considers the scores of other outputs other than the maximum value, it suffers from the following issues:

1. It does not consider the effect of scales, i.e., instead of outputs 3, 7 and 14, and if we had outputs of 0.3, 0.7 and 1.4, we would still end up with the same probability score outputs, namely 0.13, 0.29 and 0.58
2. We would end up with negative probability scores of our outputs were negative values, which may not make mathematical sense.

Thus, it is imperative we resort to some other method that takes care of the issues.

Enter, the softmax method, which is mathematically given by,

Text

Description automatically generated

where, σ (z)iis the probability score, zi,jare the outputs and β is a parameter that we choose if we want to use a base other than e1.

**Features of Softmax:**

Now for our earlier outputs 3, 7 and 14 our probabilities would be e3/ e(3+7+14) = 1.6 X 10-5, e7/ e(3+7+14) = 91 X 10-5and e14/ e(3+7+14) =0.99 respectively. As you would have noticed, this method highlights the largest values and suppresses values that are significantly below the maximum value. Also, this is done proportional to the scale of the numbers, i.e., we would not get the same probability scores if the outputs were 0.3, 0.7 and 1.4, rather we would get the probability scores as 0.18, 0.27 and 0.55

In addition, even if we end up with negative values of outputs, the probability scores would not be negative (due to the property of the distribution)

**Relationship with Sigmoid/Logistic regression:**

Let’s prove that the softmax function, which handles multiple classes, is a generalization of the logistic regression used for two-class classification.

We know that the softmax for k classes, with β=1 is given by:

[Text, letter

Description automatically generated](data:image/svg+xml,%3Csvg%20xmlns=)

We also know that for a logistic regression, there are two-classes, x and non-x (or zero), plugging these in the formula above we get:

[Chart, scatter chart

Description automatically generated](data:image/svg+xml,%3Csvg%20xmlns=)

Now dividing the numerator and denominator by exwe get:

[Chart

Description automatically generated](data:image/svg+xml,%3Csvg%20xmlns=)

The above equation is nothing but the sigmoid function, thus we see how the softmax function is a generalization of the sigmoid function (for two-class problems).

**Implementation:**

Now that we’ve understood how the softmax function works, we can use that function to compute the probabilities predicted by a crude linear model, such as y= mx +b,

Initially, we can use the linear model to make some initial predictions. We can then use an optimization algorithm, such as gradient descent, that adjusts m and b to minimize the prediction errors in our model. Thereby we end up with a final softmax regression model with good enough m and b to make future predictions from the data. Although we use softmax for the probabilities, we will have to assign a class to a data point based on the highest probability (such as argmax).

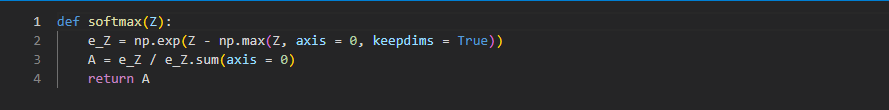
The same is represented in the schematic below.

Diagram

Description automatically generated

**Code:**

* **Softmax function**



We notice that with the large value of Z, the calculation of exp(Z) may cause overflow so we solve this by minus a large enough value from Z. This solution keeps the value of softmax function.

* **One-hot coding**

A picture containing rectangle

Description automatically generated

We assign the highest value of output vector as 1 and other value as 0 so the possibility that this data point belong to the class is 1.

* **Gradient descent**

Text

Description automatically generated

* + Convergence check(line 22): we check if the new W we found has noticeable different with the previous one or not, if not -> stop since we meet the convergence.
  + Gradient descent: Initially I only used max\_count = 10000 and the learning rate eta = 0.05, the accuracy is 0.44, after increasing max\_count to 80000 and eta to 0.5 the accuracy raised to 0.85
* **Prediction result**

A screenshot of a computer

Description automatically generated with low confidence

* + Precision: How many truly positive of all positive prediction we made.
  + Recall: How many correctly predicted positive examples of all actual positive examples.
  + F1-score: Harmonic mean of precision and recall for a more balanced summarization of model performance.

A picture containing text

Description automatically generated

* Macro avg: Average of all value in the above column.
* Weighted avg: Average of all product of each value on the above column with its support value.

**Future improvement:**

For improving the accuracy, or optimizing loss function, we could compare the different between calculated gradient and numerical (defined) gradient, or use cross entropy.

**Referrence:**

<https://machinelearningcoban.com/2017/02/17/softmax/>

<https://www.mygreatlearning.com/blog/introduction-to-softmax-regression/>

https://towardsdatascience.com/micro-macro-weighted-averages-of-f1-score-clearly-explained-b603420b292f