How to Generalize Softmax to the Case When an Object May Not Belong to Any Given Class

Dinh Tuan Nguyen, Vladik Kreinovich, Olga Koshevela, and Nguyen Hoang Phuong

Abstract The usual softmax formula transforms the degrees to which we are convinced that an object belongs to different classes – as computed by subnetworks of a neural network – into the probabilities that this objects belongs to each class. The sum of these probabilities is always 1 – which means that this formula implicitly assumes that the given object belongs to one of the given classes. In practice, however, there is always a possibility that an object does not belong to any of the given classes. To take this possibility into account, it is desirable to appropriately generalize softmax formula. In this paper, we show that all extensions that satisfy several reasonable conditions form a 1-parametric family; these extensions correspond to adding a constant to the denominator of the softmax formula.

1 Formulation of the Problem

What is softmax: a brief reminder. In many practical situations, we need to classify an object into one of the classes: e.g., based on a X-ray, decide between possible

Dinh Tuan Nguyen
Institut für Photogrammetrie und GeoInformation (IPI), Leibniz Universität Hannover
Nienburger Str. 1, D-30167 Hannover, Germany, e-mail: tuan.nguyen@ipi.uni-hannover.de

Vladik Kreinovich
Department of Computer Science, University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA, e-mail: vladik@utep.edu

Olga Kosheleva
Department of Teacher Education, University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA, e-mail: olgak@utep.edu

Nguyen Hoang Phuong
Artificial Intelligence Division, Information Technology Faculty, Thang Long University
Nghiem Xuan Yem Road, Hoang Mai District, Hanoi, Vietnam
e-mail: nhphuong2008@gmail.com

diagnoses. In the last decades, neural network-based systems turned out to be most successful in this task. In these systems, for each class i, the corresponding part of the neural networks computes a degree of confidence x_i. Based on the values, we compute the probability p_i that the given object belongs to the i-th class:

$$p_i = \frac{f(x_i)}{\sum\limits_j f(x_j)}, \tag{1}$$

where usually we take

$$f(x) = \exp(\alpha \cdot x), \tag{2}$$

for some α. For the case when the function $f(x)$ is described by the formula (2), the formula (1) is known as *softmax*; see, e.g., [1].

If want to select a single class, we pick up the class for which the probability p_i (that the object belongs to this class) is the highest, but we also get probabilities that this classification may be wrong, and that the object belongs to other classes.

Need to go beyond softmax. Softmax implicitly assumes that the object belongs to one of the given classes – since the sum of the probabilities p_i corresponding to different classes is 1. However, in practice, there is usually a possibility that the given object does not belong to any of these classes.

For example, a self-driving car needs to constantly compare the current image of its environment with the previous images, so that, based on the changes in the positions of different objects, we will be able to predict their locations in the next moments of time – and navigate accordingly. For this purpose, we need to identify each object in the new image with one of the objects in the previous image. However, it may be that the new objects has just appeared, it was not visible before: e.g., a new car has just entered the intersection. In this case, it is desirable that the system should inform us that this is probably a new object, and not one of the previously observed objects.

In this case, in addition to the probabilities $p_1, p_2, \ldots$ that the new object belongs to the each of the known classes, we would like to also have a probability p_0 that the object does not belong to any of the known classes. In this arrangement, the sum of all the probabilities – including p_0 – should also be equal to 1:

$$p_0 + p_1 + p_2 + \ldots = 1. \tag{3}$$

Formulation of the problem in commonsense terms. It is therefore desirable to come up with formulas – like (1) and (2) – that would enable us to compute all these probabilities based on the values $x_1, x_2, \ldots$.

Of course, there are many such possible formulas. So we would like to come up with reasonable conditions that would uniquely – or at least almost uniquely – determine the corresponding formulas.

What we do in this paper. In this paper, we provide such conditions, and we show that they indeed uniquely determine some formulas, formulas that form a natural generalization of softmax.

2 Analysis of the problem and the first result

Notations. Let us denote the number of possible classes by n. Then, what we need is $n+1$ functions that describe how the desired probabilities depend on the inputs:

$$p_i = f_{n,i}(x_1, \ldots, x_n), \quad i = 0, 1, \ldots, n, \tag{4}$$

for which we always have

$$p_0 + p_1 + \ldots + p_n = f_{n,0}(x_1, \ldots, x_n) + f_{n,1}(x_1, \ldots, x_n) + \ldots + f_{n,n}(x_1, \ldots, x_n) = 1. \tag{5}$$

First natural requirement: continuity. Values x_i come from processing inputs. Inputs usually come from measurements, and measurements are never absolutely accurate. There is always a difference between the measurement result and the actual value of the corresponding quantity. As a result, the values x_i – that we computed by the neural network based on the measurements results – are also somewhat different from the ideal values – the values that we would have gotten if we could use the actual (unknown) values of the corresponding quantities.

We want to make sure that when the measurements are very accurate – so that the measurement values are very close to the actual value, and thus, the computed values x_i are close to their ideal values – the resulting probabilities should be close to what we would get if we used the ideal values x_i. In precise terms, if $x_j^{(m)} \to x_j$ for all j, then we should have $f_{n,i}(x_1^{(m)}, \ldots) \to f_{n,i}(x_1, \ldots)$ for all i. In other words, all the functions $f_{n,i}(x_1, \ldots, x_n)$ should be continuous.

Second natural requirement: permutation invariance. The probabilities p_i should not depend on the order of the alternatives. In precise terms, for every permutation $\pi : \{1, \ldots, n\} \mapsto \{1, \ldots, n\}$, if we have (4), then for the probabilities

$$\widetilde{p}_i = f_{n,i}(x_{\pi(1)}, \ldots, x_{\pi(n)}), \tag{6}$$

we should have

$$\widetilde{p}_0 = p_0 \text{ and } \widetilde{p}_i = p_{\pi(i)} \text{ for } i > 0. \tag{7}$$

Third natural requirement: consistency. The values (4) are based on the assumption that all $n+1$ options are possible. If it turns out that only options $i_1, \ldots, i_k$ are possible, then we can compute the new probabilities in two different ways:

- we can start from scratch and compute the new probabilities by using the same functions, i.e., compute the values

$$\widetilde{p}_{i_j} = f_{k,i_j}(x_{i_1}, \ldots, x_{i_k}), \tag{8}$$

- we can also take into account that the new probabilities are simply conditional probabilities under the condition that only options $i_1, \ldots, i_k$ are possible; in this case, we have:

$$\widetilde{p}_{i_j} = \frac{p_{i_j}}{p_{i_1} + \ldots + p_{i_k}}. \tag{9}$$

These are two estimates for the same quantity, so they should coincide.

Fourth natural requirement: non-triviality. We are talking about situations in which there is a possibility that an object is not in any of the given classes. It is therefore reasonable to require that the corresponding probability p_0 should always be positive: $p_0 > 0$.

Our first result. It turns out that these four requirements determine the following softmax-type form of the probabilities.

Definition 1.

- *By a* probabilistic formula*, we mean a set of continuous functions $f_{n,i}(x_1, \ldots, x_n)$ from tuples of real numbers into the interval $[0,1]$, $n = 1, 2, \ldots$, $i = 0, 1, \ldots, n$.*
- *We say that a probabilistic formula is* permutation-invariant *if for every n and for every permutation $\pi : \{1, \ldots, n\} \mapsto \{1, \ldots, n\}$, the equality (7) is satisfied.*
- *We say that a probability formula is* consistent *if for every n and for every subset $\{i_1, \ldots, i_k\} \subset \{1, \ldots, n\}$, the expressions (8) and (9) coincide for every i.*
- *We say that a probabilistic formula is* non-trivial *if for every tuples $x_1, \ldots, x_n$, we have $f_{n,0}(x_1, \ldots, x_n) > 0$.*

Proposition 1.

- *Every permutation-invariant consistent non-trivial probability formula has the following form, for some continuous function $f(x) \geq 0$:*

$$f_{n,0}(x_1, \ldots, x_n) = \frac{1}{1 + f(x_1) + \ldots + f(x_n)}; \tag{10}$$

$$f_{n,i}(x_1, \ldots, x_n) = \frac{f(x_i)}{1 + f(x_1) + \ldots + f(x_n)} \text{ when } i > 0. \tag{11}$$

- *Vice versa, for every non-negative continuous function $f(x)$, the formulas (10) and (11) define a permutation-invariant consistent non-trivial probability formula.*

Comment. Thus, the only reasonable generalization of the general softmax (1) is obtained when add 1 to the denominator.

Proof. It is easy to show that the probability formula (10)–(11) is permutation-invariant, consistent, and non-trivial. Thus, to complete the proof, it is sufficient to prove that any permutation-invariant consistent non-trivial probability formula has the form (10)–(11). Indeed, let us assume that we have such a probability formula $f_{n,i}(x_1,\ldots,x_n)$. Let us prove that it has the desired form.

Let us first consider the consistency property for the subset $\{i\}$. For this subset, the equality between the expressions (8) and (9) takes, for $i=0$, the following form:

$$\frac{f_{1,0}(x_i)}{f_{1,0}(x_i)+f_{1,i}(x_i)}=\frac{f_{n,0}(x_1,\ldots,x_n)}{f_{n,0}(x_1,\ldots,x_n)+f_{n,i}(x_1,\ldots,x_n)}. \tag{12}$$

If we reverse both sides of this equality, and then subtract 1 from both sides, we will then conclude that:

$$A_i(x_i)=\frac{f_{n,i}(x_1,\ldots,x_n)}{f_{n,0}(x_1,\ldots,x_n)}, \tag{13}$$

where we denoted

$$A_i(x_i)\stackrel{\text{def}}{=}\frac{f_{1,i}(x_i)}{f_{1,0}(x_i)}. \tag{14}$$

Thus, for all $i>0$, we have

$$f_{n,i}(x_1,\ldots,x_n)=A_i(x_i)\cdot f_{n,0}(x_1,\ldots,x_n), \tag{15}$$

If we consider a permutation that swaps i and j, then, from permutation-invariance, we conclude that $A_i(x_i)=A_j(x_i)$ for all i and j. In other words, all n functions $A_1(x),\ldots,A_n(x)$ are the same function. Let us denote this function by $f(x)$. Then, the formula (15) takes a simplified form

$$f_{n,i}(x_1,\ldots,x_n)=f(x_i)\cdot f_{n,0}(x_1,\ldots,x_n). \tag{16}$$

Substituting these expressions into the formula (5), we conclude that

$$f_{n,0}(x_1,\ldots,x_n)+f(x_1)\cdot f_{n,0}(x_1,\ldots,x_n)+\ldots=1, \tag{18}$$

i.e., that

$$f_{n,0}(x_1,\ldots,x_n)\cdot(1+f(x_1)+\ldots+f(x_n))=1. \tag{19}$$

Thus, for $f_{n,0}(x_1,\ldots,x_n)$, we have exactly the expression (10). If we substitute the expression (10) into the formula (16), then for $f_{n,i}(x_1,\ldots,x_n)$, we get exactly the formula (11). The proposition is proven.

3 Alternative approach

Main idea behind this approach: let us use Bayes formula. Alternatively, let us use the usual way to update probabilities – the Bayes formula; see, e.g., [3]. In this

formula, we consider the situation in which we have several mutually inconsistent hypotheses $H_0, H_1, \ldots, H_n$ with prior probabilities $p_0(H_i)$ for which $\sum p_0(H_i) = 1$. For each possible outcome E and for each hypothesis H_i, let us denote, by $p(E \,|\, H_i)$, the probability with which the outcome E happens if this hypothesis is true. Then, if we observe one of the possible outcomes E_0, the probabilities of different hypotheses change:

- for hypotheses in which E_0 is highly probable the probabilities of these hypotheses increases, while
- for hypotheses for which the outcome E_0 was highly improbable the probabilities of these hypotheses decreases.

The resulting new probabilities p_i of different hypotheses H_i are described by the following Bayes formula:

$$p_i = \frac{p(E_0 \,|\, H_i) \cdot p_0(H_i)}{\sum\limits_j p(E_0 \,|\, H_i) \cdot p_0(H_i)}. \tag{20}$$

Let us apply the Bayes formula to our case. Let us see how we can apply the Bayes formula to the case when an object either belongs to one of the n classes, or does not belong to any of these classes. In this case, we have $n+1$ possible options, i.e., for each object, we have $n+1$ hypotheses:

- the hypotheses $H_1, \ldots, H_n$ that the object belongs to one of the n classes, and
- the hypothesis H_0 that the object does not belong to any of the given classes.

Let $p_0(H_0)$ denote the prior probability that the object does not belong to any of the given classes. What about $p_0(H_i)$? In many practical situations, we have no reason to believe that one of the classes is more probable. So, common sense implies that we should assign equal prior probability to all these n events: $p_0(H_1) = \ldots = p_0(H_n)$. This argument is known as *Laplace Indeterminacy Principle*; see, e.g., [2]. Since the sum of all the probabilities should be equal to 1, we conclude that $p_0(H_0) + n \cdot p_0(H_1) = 1$, so

$$p_0(H_1) = \ldots = p_0(H_n) = \frac{1 - p_0(H_0)}{n}. \tag{21}$$

In this case, for each hypothesis H_i, $1 \le i \le n$, an outcome E_0 is characterized by the value x_i generated by the part of the neural network that corresponds to the i-th class. We do not know how the probability $p(E_0 \,|\, H_i)$ depends on the value x_i, but we know that the larger x_i, the more probable it is that the object belongs to the i-th class. In other words, we know that $p(E_0 \,|\, H_i) = F_i(x_i)$ for some increasing function $F_i(x_i)$. Again, we do not have any reason to believe that for some x and for some classes $i \neq j$, the value $F_i(x)$ is larger than or smaller than $F_j(x)$. Thus, it makes sense to assume that for each x, the corresponding values are the same: $F_1(x) = \ldots = F_n(x)$. So, for each i, we have $p(E_0 \,|\, H_i) = F_1(x_i)$.

Under the hypothesis H_0 that the object does not belong to any of the given classes, we do not have any reason to believe that some combinations of values x_i will be more probable or less probable than others. So, in this case, we have $p(E_0\,|\,H_0) = c$ for some constant c. Now, we have expressions for prior probabilities and we have expressions for conditional probabilities. Substituting these expressions into the Bayes formula, we conclude that

$$p_0 = \frac{c}{c + \sum\limits_{j=1}^{n} F_1(x_j) \cdot p_0(H_1)} \text{ and } p_i = \frac{F_1(x_i) \cdot p_0(H_1)}{c + \sum\limits_{j=1}^{n} F_1(x_j) \cdot p_0(H_1)}. \tag{22}$$

If we divide both the numerator and the denominator of this formula by c, then we get the following expressions:

$$p_0 = \frac{1}{1 + \sum\limits_{j=1}^{n} f(x_j)} \text{ and } p_i = \frac{f(x_i)}{1 + \sum\limits_{j=1}^{n} f(x_j)}, \tag{23}$$

where we denoted

$$f(x) \stackrel{\text{def}}{=} \frac{F_1(x) \cdot p_0(H_1)}{p_0(H_0)}. \tag{24}$$

This is exactly the formulas (10)–(11) that we wanted to derive.

Comment. In our derivation, we assumed that we have no information about the corresponding probabilities, and this is indeed often the case. However, in principle, we can determine these probabilities from the observations and experiments:

- The prior probabilities $p_0(H_1), \ldots, p_0(H_n)$ are the frequencies with which objects of the corresponding class occur in the sample.
- The prior probability $p_0(H_0)$ is the frequency with which we encounter objects that do not belong to any of the given classes.

Similarly, the conditional probability $p(x_i\,|\,H_i)$ can be determined, crudely speaking, as the proportion, among all objects of the class i, of the objects for which the i-th neural sub-network returns the value x_i. To be more precise, since x_i is a continuous variable, the probably of each value is 0, so we should consider probability density:

- for some small $\varepsilon > 0$, we compute the proportion $p([x_i, x_i + \varepsilon]\,|\,H_i)$, among all the objects of class i, the ones for which the i-th neural sub-network returns a value from the interval $[x_i, x_i + \varepsilon]$,
- and then we divide this proportion by the width ε of this interval:

$$p(x_i\,|\,H_i) = \frac{p([x_i, x_i + \varepsilon]\,|\,H_i)}{\varepsilon}. \tag{26}$$

In this case, the Bayes formula enables us to use this additional information about the situation. Thus, this formula will give us more accurate estimates of the desired

probabilities p_i than the formulas (10)–(11) – formulas that do not use this information.

4 From the first result to the final formula

Which function $f(x)$ shall we use? Our objective is to generalize softmax, i.e., to make sure that when we are absolutely sure that the object belongs to one of the given classes, then we will get exactly the softmax formula (1)–(2). The corresponding probability can be obtained, from our formula (10)–(11), as the conditional probability

$$\widetilde{p}_i = \frac{p_i}{p_1 + \ldots + p_n} = \frac{f_{n,i}(x_1, \ldots, x_n)}{f_{n,1}(x_1, \ldots, x_n) + \ldots + f_{n,n}(x_1, \ldots, x_n)}. \tag{27}$$

Definition 2. *We say that a probability formula* generalizes softmax *if for every tuple $x_1, \ldots, x_n$ the expression (21) coincides with the softmax expression (1).*

Proposition 2. *For every permutation-invariant consistent non-trivial probability formula (10) – (11), the following two conditions are equivalent to each other:*

- *the probability formula generalizes softmax, and*
- *the function $f(x)$ has the form $f(x) = c \cdot \exp(\alpha \ldots x)$ for some $c > 0$.*

Comment. If we divide both numerator and denominator of the corresponding expression (10) –(11) by c, and denote $C \stackrel{\text{def}}{=} 1/c$, we conclude that in general, the probability formula that generalizes softmax has the following form:

$$p_0 = \frac{C}{C + \exp(\alpha \cdot x_1) + \ldots + \exp(\alpha \cdot x_n)}; \tag{28}$$

$$p_i = \frac{\exp(\alpha \cdot x_i)}{C + \exp(\alpha \cdot x_1) + \ldots + \exp(\alpha \cdot x_n)}. \tag{29}$$

In other words, this formula differs from the standard softmax formula by adding a positive constant C to the denominator.

In the limit, when this constant C tends to 0, our new formulas turns into the original softmax (1).

Proof. It is easy to check that for $f(x) = c \cdot \exp(\alpha \ldots x)$, the expressions (1) and (21) are indeed identical: to see that, it is sufficient to divide both numerator and denominator of the formula (27) by c.

Vice versa, let us assume that these probabilities are always equal. Since, for every $i \neq j$, the probabilities computed by formulas (1) and (21) and (22) are equal, we can conclude that if we use each of the formulas (1) and (27), we will get the exact same value of the ratio p_i/p_j. If we implicitly find this ratio by using the

formulas (1) and (27), then, by equating the two resulting expressions for p_i/p_j, we get the following equality:

$$\frac{f(x_i)}{f(x_j)} = \frac{\exp(\alpha \cdot x_i)}{\exp(\alpha \cdot x_j)}. \tag{30}$$

If we divide both sides of this equality by $\exp(\alpha \cdot x_i)$ and multiply both sides if the resulting equality by $f(x_j)$, we will get the following equality:

$$\frac{f(x_i)}{\exp(\alpha \cdot x_i)} = \frac{f(x_j)}{\exp(\alpha \cdot x_j)}. \tag{31}$$

This is true for all possible values x_i and x_j. Thus, the ratio

$$\frac{f(x)}{\exp(\alpha \cdot x)} \tag{32}$$

has the same value for all x – i.e., this ratio is a constant. If we denote this constant by c, then we conclude that for all x, we indeed have $f(x) = c \cdot \exp(\alpha \cdot x)$.

The proposition is proven.

Acknowledgments

This work was supported in part by the National Science Foundation grants 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science), HRD-1834620 and HRD-2034030 (CAHSI Includes), EAR-2225395 (Center for Collective Impact in Earthquake Science C-CIES), and by the AT&T Fellowship in Information Technology.

It was also supported by a grant from the Hungarian National Research, Development and Innovation Office (NRDI), by the Institute for Risk and Reliability, Leibniz Universitaet Hannover, Germany, and by the European Union under the project ROBOPROX (No. CZ.02.01.01/00/22 008/0004590).

References

1. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, Massachusetts, 2016.
2. E. T. Jaynes and G. L. Bretthorst, *Probability Theory: The Logic of Science*, Cambridge University Press, Cambridge, UK, 2003.
3. D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman and Hall/CRC, Boca Raton, Florida, 2011.