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

Dinh Tuan Nguyen[1], Vladik Kreinoviuch[2],
Olga Koshevela[3], and Nguyen Hoang Phuong[4]

[1]Institut für Photogrammetrie und GeoInformation (IPI),
Leibniz UniversitÃ¤t Hannover, Nienburger Str. 1,
D-30167 Hannover, Germany, tuan.nguyen@ipi.uni-hannover.de
[2,3]Departments of [2]Computer Science and [3]Teacherc Education,
University of Texas at El Paso, 500 W. University,
El Paso, TX 79968, USA, vladik@utep.edu, olgak@utep.edu
[4]Artificial Intelligence Division, Information Technology Faculty,
Thang Long University Nghiem Xuan Yem Road, Hoang Mai District,
Hanoi, Vietnam, nhphuong2008@gmail.com

1. 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 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:

$$p_i = \frac{f(x_i)}{\sum_j f(x_j)}.$$

- Here usually we take $f(x) = \exp(\alpha \cdot x)$ for some α.
- For this function $f(x)$, the formula for p_i is known as *softmax*.

2. What is softmax: a brief reminder (cont-d)

- Usually, we want to select a single class.
- Then, 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 the object belongs to other classes.

3. Need to go beyond softmax

- Softmax implicitly assumes that the object belongs to one of the given classes.
- Indeed, 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.
- Based on the changes in the positions of different objects, it should be able:
  - to predict their locations in the next moments of time – and
  - to navigate accordingly.
- For this purpose, we need to identify each object in the new image with one of the objects in the previous image.

4. Need to go beyond softmax (cont-d)

- 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: $p_0 + p_1 + p_2 + \ldots = 1$.

5. Formulation of the problem in commonsense terms

- It is therefore desirable to come up with some softmax-like formulas 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.
- In this show, we provide such conditions.
- We show that these conditions indeed uniquely determine some formulas – a natural generalization of softmax.

6. 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:

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

- For these functions, we should always have

$$p_0 + p_1 + \ldots + p_n =$$

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

7. 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.

8. First natural requirement: continuity (cont-d)

- 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.

9. 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 $p_i = f_{n,i}(x_1, \ldots, x_n)$,
  - then for the probabilities $\widetilde{p}_i = f_{n,i}(x_{\pi(1)}, \ldots, x_{\pi(n)})$, we should have $\widetilde{p}_0 = p_0$ and $\widetilde{p}_i = p_{\pi(i)}$ for $i > 0$.

10. Third natural requirement: consistency

- The values $p_i = f_{n,i}(x_1, \ldots, x_n)$ are based on the assumption that all $n+1$ options are possible.
- It may turn 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 $\widetilde{p}_{i_j} = f_{k,i_j}(x_{i_1}, \ldots, x_{i_k})$.
- But 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: $\widetilde{p}_{i_j} = \dfrac{p_{i_j}}{p_{i_1} + \ldots + p_{i_k}}$.
- These are two estimates for the same quantity, so they should coincide.

11. 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: $p_0 > 0$.
- It turns out that these four requirements determine the following softmax-type form of the probabilities.

12. Proposition

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

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

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

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

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

13. Proof

- It is easy to show that the above formulas are 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 desired form.
- 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 two expressions takes, for $i = 0$, the following form:

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

14. Proof (cont-d)

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

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

- Here we denoted $A_i(x_i) \stackrel{\text{def}}{=} \dfrac{f_{1,i}(x_i)}{f_{1,0}(x_i)}$.
- Thus, for all $i > 0$, we have

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

- Let us 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 $f(x)$.

15. Proof (cont-d)

- Then, the above formula takes a simplified form:

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

- Since the sum of all these probabilities is 1, we conclude that:

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

- So, $f_{n,0}(x_1, \ldots, x_n) \cdot (1 + f(x_1) + \ldots + f(x_n)) = 1$.
- Thus, for $f_{n,0}(x_1, \ldots, x_n)$, we have exactly the desired expression.
- If we substitute this expression into the formula for $f_{n,i}(x_1, \ldots, x_n)$, then for $f_{n,i}(x_1, \ldots, x_n)$, we also get exactly the desired formula.
- The proposition is proven.

16. Alternative approach: let us use Bayes formula

- Alternatively, let us use the usual way to update probabilities – the Bayes formula.
- 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 $p(E \mid H_i)$,
  - the probability with which the outcome E happens if this hypothesis is true.

17. Alternative approach: let us use Bayes formula (cont-d)

- 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:

$$p_i = \frac{p(E_0 \mid H_i) \cdot p_0(H_i)}{\sum\limits_j p(E_0 \mid H_i) \cdot p_0(H_i)}.$$

18. 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 $p_0(H_i)$?

19. Let us apply the Bayes formula to our case (cont-d)

- 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*.
- 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

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

- In this case, for each hypothesis H_i, $1 \le i \le n$, an outcome E_0 is characterized:
  - by the value x_i
  - that is generated by the part of the neural network that corresponds to the i-th class.

20. Let us apply the Bayes formula to our case (cont-d)

- We do not know how the probability $p(E_0 \,|\, H_i)$ depends on the value x_i.
- However, 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 $p(E_0 \,|\, H_i) = F_1(x_i)$.
- What about the hypothesis H_0 that the object does not belong to any of the given classes?

21. Let us apply the Bayes formula to our case (cont-d)

- 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 $p(E_0 \mid H_0) = c$ 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
$$p_0 = \frac{c}{c + \sum\limits_{j=1}^{n} F_1(x_j) \cdot p_0(H_1)} \text{ and } p_i = \frac{F_1(x_i) \cdot p_0(H_1)}{c + \sum\limits_{j=1}^{n} F_1(x_j) \cdot p_0(H_1)}.$$
- If we divide both the numerator and the denominator of this formula by c, then we get the following expressions:
$$p_0 = \frac{1}{1 + \sum\limits_{j=1}^{n} f(x_j)} \text{ and } p_i = \frac{f(x_i)}{1 + \sum\limits_{j=1}^{n} f(x_j)}.$$

22. Let us apply the Bayes formula to our case (cont-d)

- Here, we denoted $f(x) \stackrel{\text{def}}{=} \dfrac{F_1(x) \cdot p_0(H_1)}{p_0(H_0)}$.
- This is exactly the formulas that we wanted to derive.

23. Comment

- In our derivation, we assumed that we have no information about the corresponding probabilities.
- 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 $p(x_i \mid H_i)$ can be determined, crudely speaking, as the proportion:
  - among all objects of the class i,
  - of the proportion of objects for which the i-th neural sub-network returns the value x_i.

24. Comment (cont-d)

- 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 $p([x_i, x_i + \varepsilon] \mid H_i)$:
  - among all the objects of class i,
  - the ones for which the i-th neural sub-network returns a value from the interval $[x_i, x_i + \varepsilon]$.
- Then we divide this proportion by the width ε of this interval:

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

- In this case, the Bayes formula enables us to use this additional information about the situation.

25. Comment (cont-d)

- Thus, this formula will give us more accurate estimates of the desired probabilities p_i than the softmax.
- Reason for this: softmax does not use this information.

26. 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.
- The corresponding probability can be obtained, from our formula as the conditional probability

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

27. Proposition

For every permutation-invariant consistent non-trivial probability formula, 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 $c > 0$.*

28. Comment

- Let us divide both numerator and denominator of the corresponding expression by c, and denote $C \stackrel{\text{def}}{=} 1/c$.
- Then, we conclude that in general, the probability formula that generalizes softmax has the following form:

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

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

- 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.

29. Proof

- Let us assume that the two estimates for probabilities p_i are always equal.
- Then, for each of the two formulas, we will get the exact same value of the ratio p_i/p_j.
- So, by equating the two resulting expressions for p_i/p_j, we get the following equality:
$$\frac{f(x_i)}{f(x_j)} = \frac{\exp(\alpha \cdot x_i)}{\exp(\alpha \cdot x_j)}.$$
- 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:
$$\frac{f(x_i)}{\exp(\alpha \cdot x_i)} = \frac{f(x_j)}{\exp(\alpha \cdot x_j)}.$$
- This is true for all possible values x_i and x_j.

30. Proof (cont-d)

- Thus, the ratio $\dfrac{f(x)}{\exp(\alpha \cdot x)}$ 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.

31. Acknowledgments

This work was supported in part by:

- Deutsche Forschungsgemeinschaft Focus Program SPP 100+ 2388, Grant Nr. 501624329;
- National Science Foundation grants 1623190, HRD-1834620, HRD-2034030, and EAR-2225395;
- AT&T Fellowship in Information Technology;
- program of the development of the Scientific-Educational Mathematical Center of Volga Federal District No. 075-02-2020-1478, and
- a grant from the Hungarian National Research, Development and Innovation Office (NRDI).