Attention in machine learning: how to explain the empirical formula

Sobita Alam, Arman Hossain, Samin Islam, Arin Rahman, Olga Kosheleva, and Vladik Kreinovich

Abstract One of the most important techniques in deep learning applications is the attention technique. In this paper, we provide a theoretical explanation for the main empirical formula of attention.

1 Formulation of the problem

Why machine learning and why attention. In the last decades, a significant progress has been achieved in applying neural-network-based techniques (see, e.g., [2]) to numerous application areas – in particular, to control of technological processes. In particular, lately, this has been one of the main direction of Professor Yusupbekov's research [6, 7, 8, 9, 10, 11].

Several major ideas has led to the current success of neural networks. One of these idea is the idea of using attention techniques; see, e.g. [4]. In this paper. we provide a theoretical explanation for the empirical formulas underlying the success of attention techniques.

Classification: one of main applications of machine learning. In many practical situations, we want to classify objects into classes. For example, we want to classify pictures of pets into picture of cats and picture of dogs.

Sobita Alam, Arman Hossain, Samin Islam, Arin Rahman, and Vladik Kreinovich
Department of Computer Science, University of Texas at El Paso, 500 W. University
El Paso, Texas 79968, USA
e-mail: salam@miners.utep.edu, arahman6@miners.utep.edu, sislam3@miners.utep.edu,
ahossain4@miners.utep.edu, 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

In a computer, each object i is described by a vector $x_i = (x_{i,1}, \ldots, x_{i,N})$ consisting of this object's numerical characteristics. For example, a picture can be described by intensities of different colors at different pixels.

Based to the given object – i.e., based on the given vector x_i describing this object – we need to classify this object. Machine learning approach to the classification problem is that:

- we train the machine learning tool – e.g., a neural network – on several examples of objects for which classification is known, and
- we hope that after training, this tool will be able to correctly classify all objects.

One of main difficulties. One of the difficulties is that objects within some classes are very different. For example, dogs can be large and small, of different breeds, etc.

Attention is a way to overcome this difficulty. To make classification task easier, it is desirable: to replace each specific vector x_i with a weighted average

$$y_i = \sum_j w_{ij} \cdot x_j$$

of all the objects x_j which are similar to x_i. This way, the role of individual characteristics – that distinguish objects within the same class – will diminish, and the classification task will become easier.

Attention is a technical term for implementing this natural idea.

Comment. Recent research [5] has shown that a similar mechanism is also present in our brains, when we ourselves learn:

- signals from similar objects get routed to neighboring neurons, and
- neighboring neurons influence each other, thus "averaging" the effect of similar objects.

How to describe similarity. To implement the above idea, we need to describe, in precise terms, what it means for objects to be similar.

A natural way to describe similarity between the objects x_i and x_j is to use the usual Euclidean metric

$$d(a,b) = \sqrt{\sum_k (a_k - b_k)^2}.$$

The smaller this distance, the more similar the two objects – and thus, larger should be the weight. So, we must have $w_{ij} \sim f(d(x_i, x_j))$ for some decreasing function $f(z)$.

The sum of the weights should be equal to 1, so we must have

$$w_{ij} = \frac{f(d(x_i, x_j)}{\sum_\ell f(d(x_i, x_\ell))}. \tag{1}$$

Let us simplify this expression. The expression (1) can be simplified if we take into account that overall, the values x_{ij} are reasonably random. In this case, for large N, the arithmetic average of the values $x_{i,k}^2$ is close to its limit value m – i.e., to the mathematical expectation of this random quantity; see, e.g., [3]:

$$\frac{x_{i,1}^2+\ldots+x_{i,N}^2}{N} \approx m \stackrel{\text{def}}{=} E[x_{i,j}^2].$$

Thus, we have

$$x_i^2 = x_{i,1}^2+\ldots+x_{i,N}^2 \approx C \stackrel{\text{def}}{=} N\cdot m.$$

It is easy to check that $d^2(x_i,x_j) = x_i^2+x_j^2-2x_i\cdot x_j \approx 2C-2x_i\cdot x_j$, where $x_i\cdot x_j$ denotes the usual scalar (dot) product of the two vectors:

$$x_i\cdot x_j \stackrel{\text{def}}{=} x_{i,1}\cdot x_{j,1}+\ldots+x_{i,N}\cdot x_{j.N}.$$

So, a decreasing function of $d(x_i,x_j)$ can be described as an increasing function of the dot product $x_i\cdot x_j$:

$$f(d(x_i,x_j)) = F(x_i\cdot x_j),$$

where we denoted

$$F(z) \stackrel{\text{def}}{=} f\left(\sqrt{2C-2z}\right).$$

Thus, we arrive at the following formula:

$$w_{ij} = \frac{F(x_i\cdot x_j)}{\sum_\ell F(x_i\cdot x_\ell)}, \tag{2}$$

for some increasing function $F(z)$.

Which function $F(z)$ should we use? Empirical evidence shows that out of all increasing functions $F(z)$, functions $F(z)=\exp(\alpha\cdot z)$ work the best.

A natural question: How can we explain this empirical fact?

What we do in this paper. In this paper, we provide a theoretical explanation for this empirical fact.

2 Our explanation

The main idea behind our explanation. Our explanation is based on the fact that the values $x_{i,j}$ come from measurements, and measurements are never absolutely accurat: there is always some noise affecting the measurement results. So, a natural requirement is that the resulting values y_i should be affected by the noise as little as possible.

Let us formulate this requirement in precise terms. What if we replace the original values $x_{i,j}$ with noisy values $\widetilde{x}_{i,k} = x_{i,k} + n_{i,k}$ for some noise $n_{i,k}$ with 0 mean? Then the dot product $\widetilde{x}_i \cdot \widetilde{x}_j$ becomes $x_i \cdot x_j + x_i \cdot n_i + n_i \cdot x_j + n_i \cdot n_j$.

The expected value of terms $x_i \cdot n_j$ is 0, so the only non-zero addition to the dot product is the term $E[n_i \cdot n_j]$. Let us estimate this expected value.

If the noise is *local* – i.e., if noises corresponding to two objects x_i and x_j are independent – then the expected value of the noises' product is equal to the product of their expected values, i.e., to 0:

$$E[n_i \cdot n_j] = E[n_{i,1} \cdot n_{j,1} + \ldots + n_{i,N} \cdot n_{j,N}] = E[n_{i,1} \cdot n_{j,1}] + \ldots + E[n_{i,N} \cdot n_{j,N}] =$$

$$E[n_{i,1}] \cdot E[n_{j,1}] + \ldots + E[n_{i,N}] \cdot E[n_{j,N}] = 0.$$

However, other, the noise has a *global* component with mean square value M, i.e., a component that affects all the measurements. In this case, $E[n_i \cdot n_j] = M$. Thus, due to the noise, all dot products are increased by the same constant M.

So, the above requirement takes the following form: we want to find the function $F(v)$ for which adding a constant M to all the dot products would not change the weights.

Which functions $F(z)$ satisfy this requirement? For two objects, the above requirement means that for all a, b, and M we should have:

$$\frac{F(a+M)}{F(a+M)+F(b+M)} = \frac{F(a)}{F(a)+F(b)}. \tag{3}$$

If we apply $1/z$ to both sides of this equality and subtract 1 from both sides, we get

$$\frac{F(b+M)}{F(a+M)} = \frac{F(b)}{F(a)}. \tag{4}$$

Multiplying both sides by

$$\frac{F(a+M)}{F(b)},$$

we get

$$\frac{F(b+M)}{F(b)} = \frac{F(a+M)}{F(a)}. \tag{5}$$

This equality holds for all a and b. So, the ratio

$$\frac{F(a+M)}{F(a)}$$

does not depend on a, it only depends on M:

$$\frac{F(a+M)}{F(a)} = g(M) \tag{6}$$

for some function $g(M)$. Thus,

$$F(a+M) = g(M) \cdot F(a). \tag{7}$$

It is known (see, e.g., [1]; see also a comment below) that the only increasing solution to this functional equation is

$$F(a) = c \cdot \exp(\alpha \cdot a). \tag{8}$$

So, we get the desired explanation. From the viewpoint of the weights $w_{i,j}$ – as described by the formula (2) – the use of the function (8) is equivalent to using the function

$$F(a) = \exp(\alpha \cdot a). \tag{9}$$

This is exactly what we needed to explain.

Comment: how to solve the above functional equation. To solve the above functional equation, let us differentiate both sides by M and take $M = 0$. Then, we get

$$F'(a) = g'(0) \cdot F(a),$$

with $\alpha \stackrel{\text{def}}{=} g'(0)$, i.e.,

$$\frac{dF}{da} = \alpha \cdot F. \tag{10}$$

Dividing both sides by F and multiplying both sides by da, we get

$$\frac{dF}{F} = \alpha \cdot da. \tag{11}$$

We can now integrate both sides, and get

$$\ln(F) = \alpha \cdot a + \text{const}. \tag{12}$$

Now, we can apply the function $\exp(z)$ to both sides, and get

$$F(a) = \text{const} \cdot \exp(\alpha \cdot a).$$

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. J. Aczél and J. Dhombres, *Functional Equations in Several Variables*, Cambridge University Press, 2008.
2. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, Massachusetts, 2016.
3. D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman and Hall/CRC, Boca Raton, Florida, 2011.
4. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need", *Proceedings of the 31st Conference on Neural Information Processing Systems NIPS 2017*, Long Beach, California, USA, December 4–9, 2017, pp. 6000–6010.
5. W. J. Wright, N. G. Hendrick, and T. Komiyama, "Distinct symaptic plasticity rules operate across dendritic compartments in vivo during learning", *Science*, 2025, Vol. 388, pp. 322–328.
6. N. R. Yusupbekov, Sh. M. Gulyamov, and M. Yu. Doshchanova, "Neural identification of a dynamic model of a technological process", *Proceedings of the 2019 International Conference on Information Science and Communications Technologies ICISCT 2019*, Tashkent, Uzbekistan, November 4–6, 2019.
7. N. R. Yusupbekov, H. Z. Igamberdiev, and U. F. Mamirov, "Adaptive control system with a multilayer neural network under parametric uncertainty condition", *Proceedings of the 8th International Conference on Fuzzy Systems, Soft Computing and Intelligent Technologies FSSCIT 2020*, Smolensk, Russia, June 29 – July 1, 2020; *CEUR Workshop Proceedings*, 2020, Vol. 2782, pp. 228–234.
8. N. R. Yusupbekov, H. Z. Igamberdiev, O. O. Zaripov, and U. F. Mamirov, "Stable iterative neural network training algorithms based on the extreme method", In: R. A. Aliev, J. Kacprzyk, W. Pedrycz, M. Jamshidi, M. Babanli, and F. M. Sadikoglu (eds.), *Proceedings of the International Conference on Theory and Applications of Fuzzy Systems and Soft Computing ICAFS 2020*, Budva, Montenegro, August 27–28, 2020, Springer, Cham, Switzerland, 2020, pp. 246–253.
9. N. Yusupbekov, D. Mukhitdinov, Y. Kadirov, O. Sattarov, and A. Samadov, "Control of nonstandard dynamic objects with the method of adaptation according to the misalignment based on neural networks", *International Journal of Emerging Trends in Engineering Research*, 2020, Vol. 8, No. 9, pp. 5273–5278.
10. N. R. Yusupbekov, D. P. Mukhitdinov, and O. U. Sattarov, "Neural Network Model for Adaptive Control of Nonlinear Dynamic Object", In: R. A. Aliev, N. R. Yusupbekov, J. Kacprzyk, W. Pedrycz, and F. M. Sadikoglu (eds.), *Proceedings of the 11th World Conference "Intelligent System for Industrial Automation" WCIS 2020*, Tashkent, Uzbekistan, November 26–28, 2020, Springer, Cham, Switzerland, 2020, pp. 229–236.
11. N. R. Yusupbekov, D. P. Mukhitdinov, O. U. Sattarov, and S. B. Boybutaev, "Construction of a neural network using an approach to a genetic algorithm", *International Journal of Advanced Research in Science, Engineering and Technology*, 2019, Vol. 6, No. 6, pp. 9837–9841.