The Golden Ratio in Machine Learning

Stefan Jaeger
National Library of Medicine
National Institutes of Health
Bethesda, MD 20894, USA
stefan.jaeger@nih.gov

***Abstract*—Gradient descent has been a central training principle for artificial neural networks from the early beginnings to today's deep learning networks. The most common implementation is the backpropagation algorithm for training feed-forward neural networks in a supervised fashion. A drawback of backpropagation has been the search required to find optimal values of two important training parameters, learning rate and momentum weight. The learning rate specifies the step size towards a minimum of the loss function when following the gradient, while the momentum weight considers previous weight changes when updating current weights. Using both parameters in conjunction with each other generally improves training, although their specific values do not follow immediately from standard backpropagation theory. This paper proposes a new information-theoretical loss function based on cross-entropy for which it derives a specific learning rate and momentum weight. Many training procedures based on backpropagation use cross-entropy directly as their loss function. Instead, this paper investigates a dual process model with two processes, in which one process minimizes the Kullback-Leibler divergence while its dual counterpart minimizes the Shannon entropy. The golden ratio plays an important role here, allowing to derive theoretical values for the learning rate and momentum weight, matching closely the values traditionally used in the literature, which are determined empirically. To validate this information-theoretical approach further, classification results for a handwritten digit recognition task are presented, showing that the proposed loss function, in conjunction with the derived learning rate and momentum weight, works in practice.**

***Index Terms*—machine learning, pattern recognition, neural network theory, optimization, golden ratio**

I. Introduction

Modern neural networks are typically trained on big data in a supervised fashion. This is commonly achieved by minimizing a loss function that measures the distance between network output and teaching input in several iterations so that the network output approaches the training input. A widely used method to do this is backpropagation, which descends along the gradient of the loss function by propagating weight changes through the network [1]–[3]. While backpropagation has been applied very successfully, there are still several open questions requiring a definite answer. In particular, the type of loss function to be used and the optimal way of following the gradient are still open problems. A variety of loss functions has been used in the literature to judge the quality of a network output. Similarly, many optimization strategies have been investigated to let gradient descent converge faster or avoid local minima, including stochastic gradient descent, second order methods, and others [2], [4]. However, most design decisions are still largely based on empirical evidence and experience.

This paper presents a theoretical study of one of the most popular loss functions, cross-entropy. Technically, the paper describes ideas developed in [5], [6]. It discusses the theoretical ramifications when cross-entropy is minimized by two dual processes, which minimize the Kullback–Leibler divergence and the Shannon entropy, respectively. The advantage of this approach is that it leads to a loss function and weight space for which learning rate and momentum weight can be derived theoretically. These regularization parameters control weight updates during backpropagation and thus have a strong effect on gradient descent. Although rules of thumb and effective heuristics exist for choosing these parameters, including dynamic parameter updates during backpropagation, they have eluded a conclusive theoretical analysis so far. The theoretical values derived in the following are similar to empirical values in the literature. Another feature of the dual process model proposed in this paper is that it includes the golden ratio, which is a novel concept in machine learning.

The paper is structured as follows: Section II motivates the paper, discussing intrinsic uncertainty in human perception. Section III recalls the well-known definitions of cross-entropy, Shannon entropy, and Kullback–Leibler divergence. Then, Section IV lists the mathematical features of the golden ratio, before Section V formalizes the intrinsic uncertainty between observed and true probabilities. Based on these results, Section VI introduces a new loss function. Next, Section VII derives the two regularization parameters, learning rate and momentum weight. Finally, Section VIII shows a practical application of the theoretical framework to deep learning for handwritten digit recognition. The paper concludes with a discussion and a summary of the main results.

II. Motivation and Approach

The main motivation of the approach developed in this paper lies in the observation that human perception is fraught with intrinsic uncertainty. A good example is Rubin's Vase, an ambiguous visual perception made popular by the Danish psychologist Edgar Rubin around 1915, which is shown in Figure 1 [7]. This figure could be interpreted either as a vase, or as two faces looking at each other, depending on what is considered background and foreground. One interpretation is

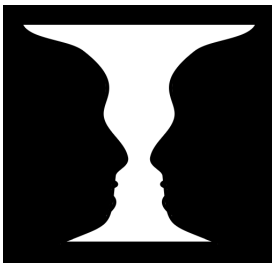

Fig. 1. Rubin's Vase ([7])

the complement of the respective other. To develop this idea further in a more formal approach, let $\mathcal{X}$ be a discrete random variable with two possible values, $\mathcal{X} = \{x, \neg x\}$. Furthermore, let $\mathrm{P}(\mathcal{X})$ be a probability mass function on $\mathcal{X}$ that assigns probability values as follows: $\mathrm{P}(x) = p$, and $\mathrm{P}(\neg x) = 1 - p$. Assuming that $\mathcal{X}$ underlies the intrinsic uncertainty of our perception, viewers do not know whether they observe x or $\neg x$, or in information-theoretical terms, they do not know whether the information content of their observation is $-\ln(p)$ or $-\ln(1-p)$. Assuming, without loss of generality, that p is the true probability, viewers do not know whether the information they can expect is $-p \cdot \ln(p)$ or $-p \cdot \ln(1-p)$. Only if both terms are equal, which is the case for $p = 0.5$ when $p \in]0\,;1[$, is there no uncertainty between them:

$$-p \cdot \ln(p) = -p \cdot \ln(1-p), \tag{1}$$

or equivalently:

$$0 = -p \cdot \ln\left(\frac{1-p}{p}\right) \tag{2}$$

In (2), the intrinsic uncertainty shows as follows: If an observer knows the value of p, the observer does not know whether the fraction $(1-p)/p$ or its inverse needs to be used as argument to the logarithmic term, and for that matter, does not know the sign of the difference. Conversely, if the observer knows the sign of the difference, the observer cannot know the value of p, which could be either p or $1-p$.

Equation (2) also shows the basic equation structure of the Kullback–Leibler divergence, which will be discussed in the next section and Section V.

III. Cross Entropy

Cross-entropy is a common loss function used for training of artificial neural networks [8], [9]. It quantifies the difference between two probability distributions, say p and q, [10]. In communication theory, it measures the average number of bits needed to encode data coming from a source with distribution p, when the model (encoding) is optimized for an estimated probability distribution q, rather than the true distribution p.

Mathematically, for discrete probability distributions p and q, defined on the same probability space $\mathcal{X}$, the cross-entropy $H(p,q)$ is computed as follows:

$$H(p,q) = -\sum_{x \in \mathcal{X}} p(x) \cdot \ln\Big(q(x)\Big) \tag{3}$$

The cross-entropy $H(p,q)$ can also be expressed as the sum of the Kullback–Leibler divergence $D(p,q)$ from p to q and the Shannon entropy $H(p)$:

$$H(p,q) = D(p,q) + H(p) \tag{4}$$

The next subsections will briefly write out the definitions for these two information measures, including the special case of a two-valued random variable $\mathcal{X}$ with outcome probabilities p and $1-p$.

A. Kullback–Leibler divergence

The Kullback–Leibler divergence $D_{\mathrm{KL}}(p,q)$ describes the difference between a probability distribution p, say a measured observation, and a second probability distribution, q, serving as a reference or model distribution [11]. The Kullback–Leibler divergence can then be interpreted as the average difference of the number of bits required for encoding samples of p using the optimal encoding given by q (rather than the optimal coding for p).

For the two distributions, p and q, the Kullback–Leibler divergence $D_{\mathrm{KL}}(p,q)$ is then defined as follows

$$D_{\mathrm{KL}}(p,q) = -\sum_{x \in \mathcal{X}} p(x) \cdot \ln\left(\frac{q(x)}{p(x)}\right) \tag{5}$$

for a probability space $\mathcal{X}$.

In the specific case of a two-valued random variable $\mathcal{X}$, the Kullback–Leibler divergence $D_{\mathrm{KL}}(p, 1-p)$ therefore computes as follows

$$D_{\mathrm{KL}}(p, 1-p) = -p \cdot \ln\left(\frac{1-p}{p}\right) - (1-p) \cdot \ln\left(\frac{p}{1-p}\right) \tag{6}$$

For this specific case, the divergence $D(p) = D_{\mathrm{KL}}(p, 1-p)$, assumes its minimum of zero when both distributions p and $1-p$ are identical, and thus when both outcomes of the random variable have a probability of 0.5.

B. Shannon entropy

The Shannon entropy, or simply entropy, of a random variable $\mathcal{X}$ is the average information conveyed by its possible outcomes [12]. In mathematical terms, the entropy $H(\mathcal{X})$ of $\mathcal{X}$ can be computed as follows:

$$H(\mathcal{X}) = -\sum_{x \in \mathcal{X}} p(x) \cdot \ln\Big(p(x)\Big), \tag{7}$$

where $p(x)$ is a probability distribution defined on all outcomes x of $\mathcal{X}$.

For a random variable $\mathcal{X}$ with two possible outcomes, with probabilities p and $1-p$, the entropy thus computes as follows:

$$H(\mathcal{X}) = -p \cdot \ln(p) - (1-p) \cdot \ln(1-p) \tag{8}$$

In this case, contrary to the Kullback–Leibler divergence, the entropy assumes its maximum (not its minimum) when the outcome and its complement have the same probability, namely $p = 0.5$.

IV. Golden Ratio

This section highlights a connection between (2) and the golden ratio [13]. In (2), the argument to the logarithm, $(1-p)/p$, can be regarded as the perceived (or measured) probability, whereas the multiplier p is the true probability. If an observer wants to measure the true probability, both terms need to be equal, meaning

$$p = \frac{1-p}{p} \tag{9}$$

This holds true if p is the golden ratio, as discussed next.

Equation (9) is equivalent to the following quadratic equation:

$$p^2 + p - 1 = 0, \tag{10}$$

which has two irrational solutions p_1 and p_2:

$$p_1 = \frac{\sqrt{5}-1}{2} \approx 0.618, \tag{11}$$

and

$$p_2 = \frac{-\sqrt{5}-1}{2} \approx -1.618 \tag{12}$$

An important feature of both solutions is that their complement, $1-p$, equals their square,

$$1 - p = p^2 \tag{13}$$

Alternatively, the golden ratio can be derived from another quadratic equation, which may be more commonly found in textbooks, and which results from replacing p by $-p$ in (10):

$$p^2 - p - 1 = 0 \tag{14}$$

This equation also has two irrational solutions, which are the negatives of p_1 and p_2:

$$-p_1 \approx -0.618 \quad \text{and} -p_2 \approx 1.618 \tag{15}$$

However, for both solutions of the second quadratic equation, (14), the complement $1-p$ is the negative reciprocal:

$$1 - p = -\frac{1}{p} \tag{16}$$

The sum of both solutions for (10) and (14) is either minus one or one, respectively:

$$p_1 + p_2 = -1 \quad \text{and} \qquad -p_1 - p_2 = 1 \tag{17}$$

The literature sometimes uses the letter φ for the golden ratio, and typically defines the golden ratio as a single value, often with $\varphi \approx 1.618$ and discarding negative values [13], [14]. In this paper, all four solutions to the quadratic equations above will be referred to as the golden ratio.

V. Intrinsic Uncertainty

For (2), the previous section showed that the measured probability equals the true probability in the golden ratio. This section develops (2) in a way that allows computing all possible measurements in a systematic way.

By coupling the measured probability with the true probability, according to the relationship in (9), and using the letter E (Energy) to denote the information difference, (2) can be developed as follows:

$$E = -p \cdot \ln\left(\frac{1-p}{p}\right) \tag{18}$$

$$\Leftrightarrow -p^2 \cdot \ln\Big(1-p\Big) \tag{19}$$

$$\Leftrightarrow -p \cdot \ln\Big(1-p^2\Big) \tag{20}$$

$$\Leftrightarrow -p \cdot \ln\Big(\sqrt{1-p^2}\Big) \cdot 2 \tag{21}$$

$$\Leftrightarrow -\sin(\phi) \cdot \ln\big(\cos(\phi)\big) \cdot 2, \tag{22}$$

where the last expression holds for an angle $\phi \in \left[0\,;\frac{\pi}{2}\right]$. Varying ϕ in this range will produce all possible measurements, which are points on the unit circle. Using this scheme, the measured probability equals the true probability for $\phi = \pi/4$, and a measured probability of $\sin(\pi/4) = \cos(\pi/4) = 1/\sqrt{2}$. The letter E for energy is used in (18) to emphasize that information is related to energy in the physical sense and also to emphasize that this energy, or information, can be absorbed or released, depending on the sign.

In (22), the energy E attains its minimum of zero for $\phi = 0$, when $\sin(\phi) = 0$ and $\cos(\phi) = 1$. On the other hand, E reaches its maximum, infinity, for $\phi = \pi/2$, when $\sin(\phi) = 1$ and $\cos(\phi) = 0$.

Now, a dual energy for a second observer can be computed by swapping sine and cosine in (22), which amounts to using the main diagonal as a mirror axis. This dual energy will reach its maximum, when the original energy reaches its minimum; and vice versa, it will be minimum when the original energy is maximum. Most importantly, it is possible to establish a formal analogy to the intrinsic uncertainty in observations, as motivated in Section II. For this, let there be two dual and intertwined processes based on the dual energies above. While one process considers all energies for $\phi > \pi/4$ as released, and all energies for $\phi < \pi/4$ as absorbed; its dual counterpart considers energies for $\phi > \pi/4$ as absorbed, and all energies for $\phi < \pi/4$ as released. The intrinsic uncertainty then shows as follows: If one process knows whether energy is released or absorbed, it does not know the magnitude of the energy. Conversely, if a process knows the magnitude of the energy, it does not know whether the energy is released or absorbed. Only when both processes work hand in hand, synchronously, can there be knowledge about the direction and magnitude of energy. However, an observer can truly measure only one or the other, which shows a connection to Heisenberg's uncertainty principle in physics [15], [16].

VI. Loss function

This section will develop a loss function based on the theoretical results derived above. Section V showed that for an angle of $\phi = \pi/4$, one of the two dual intertwined processes will know the quantity of the input, but not the sign. Typically, for a machine learning task, the goal is to match the magnitude of the teaching input with the output of a classifier, say a neural network. The sign of the teaching input is ignored, or tacitly assumed to be positive. As discussed above, there is no way of knowing both the magnitude and the sign of the teaching input. Therefore, assuming a binary teaching input t, let the difference function $d(y,t)$ between t and network output y be defined as follows:

$$d(y,t) = (y - t + 1) \cdot \frac{\pi}{4} \tag{23}$$

This function will output values in the interval $\left[0; \frac{\pi}{2}\right]$, and will be equal to one of the outer boundaries of the interval when the difference between network output and teaching input is maximum, with $|y - t| = 1$. In case the network output is identical to the teaching input, $d(y,t)$ will be in the middle of the interval, $d(y,t) = \pi/4$. The angle defined by $d(y,t)$ can now be used as ϕ, and its cosine as the observed input.

Based on the distance function in (23), the combined energy E^* for both dual energies can be computed as follows:

$$E^*(d) = -\sin(d) \cdot \ln\big(\cos(d)\big) - \cos(d) \cdot \ln\big(\sin(d)\big) \tag{24}$$

The energy in (24) assumes its minimum for an angle of $45°$, or $d = \pi/4$.

Based on the combined energy in (24), the loss function for training is developed as follows: First, resolving (18) for the true probability p leads to the following sigmoidal expression for p (see [17] for the role of the sigmoid function in natural neural networks):

$$p = \frac{1}{1 + \exp\left(-E/p\right)} \tag{25}$$

Then, using $E^*(d) - E^*(\pi/4)$ as an information estimate of E in (25), and assuming the equilibrium value $1/\sqrt{2}$ for p, with $\phi = \pi/4$ in (22), produces the following loss function:

$$L(d) = \frac{\sqrt{2}}{1 + \exp\left(-(E^*(d) - E^*(\pi/4))/2\right)} \tag{26}$$

The loss function in (26) reaches its minimum of $1/\sqrt{2}$ when the network output equals the teaching input, and when $y - t = 0$, with $d = \pi/4$.

The derivative of the loss function in (26) with respect to the prediction Y, dL/dY, can be computed by applying the chain rule [5]. This derivative can then be used for descending the gradient in the traditional backpropagation process. The next section introduces the corresponding learning rate and momentum weight to be used in combination with the loss function in (26)) and its derivative.

VII. Regularization

A training method based on backpropagation adapts the network weights in a way that minimizes the loss, meaning the difference between network output and teaching input [3]. Using gradient descent, training implies computing the gradient of a loss function L, such as the loss given by (26), with respect to each network weight. A backpropagation method accomplishes this for one network layer at a time, iteratively, propagating the gradient back from the output layer to the input layer. To move along the gradient towards the minimum of the loss function, a delta is added to each weight, which has the following form, when adding also a momentum term:

$$\Delta w_{ij}(t) = -\eta \frac{\partial L}{\partial w_{ij}(t)} + \alpha \cdot \Delta w_{ij}(t-1) \tag{27}$$

In (27), $\Delta w_{ij}(t)$ denotes the delta added to each weight w_{ij} between a node i and a node j in the network, at training iteration (or time) t. The term $\partial L/\partial w_{ij}(t)$ is the partial derivative of the loss function with respect to w_{ij}, at time t, which is multiplied with the learning rate η. The sign of $\Delta w_{ij}(t)$ is negative so that the loss function approaches its minimum. In practice, a momentum term describing the weight change at time $t-1$, $\Delta w_{ij}(t-1)$, is commonly added. This term is typically multiplied by a weighting factor α, as seen in (27). The general conception is that the momentum term improves stochastic gradient descent by dampening oscillations. However, according to the dual process model developed here, the actual reason for the performance improvement brought about by the momentum term lies in the gradient of the dual process.

As of yet, a conclusive theory for the optimal values of the learning rate η and the momentum weight α has been lacking, although second order methods have been tried [2], [4], [18]; see also [19]–[22] as examples of adaptive methods proposed. Both parameters are often determined heuristically, either through empirical experiments or through systematic search [23]. Training results can be very sensitive to the value of the learning rate. For example, a small learning rate may produce a slow convergence, whereas a larger learning rate may result in the search passing over the minimum loss. Negotiating this delicate trade-off in the regularization of the training process can be time-consuming in practical applications. Literature seems to prefer initial learning rates around 0.01 or smaller, although reported values differ by several orders of magnitude. For the momentum weight, higher initial values around 0.9 are more common [9], [24]–[26].

The proposed dual process model allows deriving theoretical values for both regularization parameters, learning rate η and momentum weight α. The information loss function in (24), and the loss function in (26) assume their minima for an angle of $45°$, or $d = \pi/4$, in the state of equilibrium when both dual energies are equal. This corresponds to $p = 0.5$ in (18). Minimizing the loss function in (26) therefore minimizes the Kullback–Leibler divergence and maximizes the entropy.

On the other hand, the dual process minimizing the negative energies does the opposite. It maximizes the Kullback–Leibler

divergence and minimizes the entropy. A gradient in the dual process model is therefore a composite of the gradients of both processes, involving the gradient of one process and the negative gradient of its dual process. Each summand in the weight adjustment defined by (27), namely the partial derivative $\partial L/\partial w_{ij}(t)$ and the momentum term $\Delta w_{ij}(t-1)$, corresponds to a gradient of one of the dual processes.

The momentum weight α follows from results above. The probability p in (18) can be considered as a gradient of a linear function with information input, $-\ln(x)$, and information output, E. The dual process, with input and output reversed, has a similar gradient. Because both dual processes are intertwined, it is fair to say that the dual process happens at time $t-1$, and that the current process at time t observes the output of its dual counterpart. Therefore, the multiplier p in (18) represents the gradient from the previous iteration. This gradient, and thus the delta at $t-1$, $\Delta w_{ij}(t-1)$, needs to be multiplied by a constant to obtain the golden ratio for the state of equilibrium, in (18), for which the measured probability equals the true probability. This regularization can be computed as follows:

$$\alpha = \sqrt{2} \cdot p_1 \approx 0.874, \tag{28}$$

where p_1 is the value of the golden ratio in (11), which provides the value for the momentum weight $\alpha \approx 0.874$.

The learning rate η can be derived from the momentum weight α by converting the latter to the corresponding value of the dual process. For $\phi = \pi/2$, in (22), $\sin(\phi)$ becomes one. Therefore, after multiplying with the momentum weight, the multiplier, or true probability, will be α. The true probability for the dual process is then given by the complement of α: $1 - \alpha$. To make probabilities consistent with each other, this measured probability needs to be squared according to (13) in order to compute its complement. Taking the complement twice can be understood as looking at the same process from a dual point of view. Applying these steps to the momentum weight α results in the following expression for the learning rate η:

$$\eta = (1 - \alpha)^2 \approx 0.016 \tag{29}$$

This provides the value for the second regularization term, learning rate η, with $\eta \approx 0.016$ [6].

VIII. Experimental Evaluation

To show that the proposed loss function in (26) works in conjunction with the derived learning rate η and momentum weight α, a practical experiment is performed on public data. For handwritten digit classification, a deep learning network is trained on a dataset containing 10,000 handwritten, artificially rotated digits, and evaluated by averaging ten runs for each fold in 10-fold cross-validation [27]. Each digit is a 28-by-28 gray-scale image, with a corresponding label denoting which digit the image represents (MNIST database [28]). Figure 2 shows the network architecture used in the experiment, with a sample input digit 3 and a correct output result [9]. The first layer is the image input layer, with a size of 28-by-28, followed by a convolution layer with 20 5-by-5 filters. The next layers are a batch normalization layer, a ReLU layer, and a fully-connected layer with an output size of 10. Finally, a softmax layer and a classification layer are the last two layers of the network, with the latter computing the proposed loss function in (26). For training, the learning rate given by (29) and the momentum weight given by (28) are used.

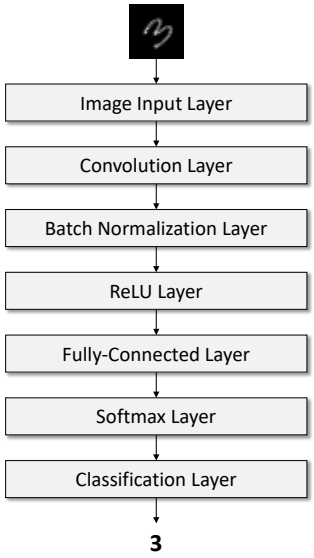

Fig. 2. Network architecture

Table I shows the classification results for training with the common loss function defined by the sum of squares, and with the proposed loss function defined by (26). All results have been achieved after 30 training epochs, using ten-fold cross-validation. The results show that training with SSE loss benefits significantly from using a momentum term, which increases the accuracy from 77.4% to 98.9%. The proposed loss function in (26) with momentum performs best, with an accuracy of 99.4%. It is also worth noting that the standard deviation improves by an order of magnitude each time, decreasing from 10 for SSE without momentum to only 0.06 for the proposed loss function, learning rate, and momentum weight.

TABLE I
Experimental results with 10-fold cross-validation

Loss	avg. accuracy (%)	std
SSE without momentum	77.4	10
SSE with momentum	98.9	1
(26) with momentum	99.4	0.06

IX. Discussion

The previous sections have shown how the regularization parameters of the delta learning rule given by (27), which are learning rate η and momentum weight α, can be derived from a dual process model. Specifically, according to the

proposed theoretical framework, the delta rule combines the gradients of two dual processes. This goes beyond the traditional understanding according to which the momentum term produces a more stable gradient descent by smoothing weight changes over several iterations. While one process minimizes the Kullback–Leibler divergence ($p = 0.5$) and maximizes the Shannon entropy ($p = 0.5$), its dual counterpart does the opposite, by maximizing the Kullback–Leibler divergence and minimizing the entropy.

The process that minimizes the Kullback–Leibler divergence ensures that the output equals the training input, while its dual counterpart that minimizes the Shannon entropy ensures that there is no uncertainty in the output. However, only both processes taken together can minimize the cross entropy. Each process alone has limitations. The process minimizing the Kullback–Leibler divergence may know that the output equals the training input, but it cannot know with absolute certainty whether the output should be zero or one. This is why the loss function given by (26) is defined in such a way that it assumes its minimum for an angle of of $45°$. On the other hand, the process minimizing the Shannon entropy may know that the output is zero, for example, but it cannot know if this is equal to the intended teaching input.

This reveals an inherent problem of the teaching input that has largely gone unnoticed so far in the literature. It is only possible to know either the teaching input signal, zero or one, or the actual teaching input, which could be identical to the teaching input signal or could equally well be its complement. It is only possible to know one or the other, but not both, which is reminiscent of Heisenberg's Uncertainty principle.

Under these theoretical considerations, the gradient adjustment by means of the delta learning rule, as defined by (27), becomes a composite of two gradient adjustments. On the one hand, the gradient is followed to minimize the Kullback–Leibler divergence. On the other hand, the reversed gradient of the dual process maximizing the Kullback–Leibler divergence is followed to minimize the entropy. After a successful training, both processes together have minimized the cross-entropy. However, their knowledge is distributed among them. While one process has learned to mimic the teaching input, the dual process has learned whether the teaching signal needs to be taken at face value or if it needs to be flipped.

The theoretical results in this paper confirm that cross-entropy is a profound loss function. However, rather than using cross-entropy directly as a loss function, it may be more appropriate to use it indirectly, via the sum of Kullback–Leibler divergence and Shannon entropy, following the dual process model. This can be achieved by using the loss function defined by (26), and applying the delta learning rule with momentum, as given by (27), with the specific values for learning rate η and momentum weight α derived in Section VII.

X. Conclusion

This paper presents a theoretical analysis for minimizing cross-entropy. The main result is a model comprising two dual processes, with one process minimizing the Kullback–Leibler divergence and the other process minimizing the Shannon entropy. The golden ratio plays a major role in this model, and is a novel concept in machine learning. Specific values for learning rate and momentum weight follow from the model. The order of magnitude of these values is very similar to empirical values often used in the literature. Choosing these values for both regularization parameters improves the performance of gradient descent. The proposed theoretical framework could therefore be a step toward a better understanding of gradient descent and rendering an expensive hyperparameter grid search redundant in the future.

Acknowledgment

This work was supported by the Intramural Research Program of the National Library of Medicine, National Institutes of Health.

References

[1] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[2] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 437–478.

[3] Y. LeCun, L. Bottou, G. Orr, and K. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[4] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International Conference on Machine Learning*, 2013, pp. 1139–1147.

[5] S. Jaeger, "The golden ratio of learning and momentum," arXiv:2006.04751 [cs.LG], June 2020. [Online]. Available: https://arxiv.org/abs/2006.04751

[6] ——, "A dual process model for optimizing cross entropy in neural networks," arXiv:2104.13277v1 [cs.LG], April 2021. [Online]. Available: https://arxiv.org/abs/2104.13277

[7] E. Rubin, *Rubin Vase*, Wikimedia Commons (last accessed March 4, 2022, CC BY-SA 3.0), 1915. [Online]. Available: https://commons.wikimedia.org/wiki/File:Facevase.png

[8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[9] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[10] R. Y. Rubinstein and D. Kroese, *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2013.

[11] S. Kullback and R. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[12] C. Shannon, "A mathematical theory of communication," *Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[13] M. Livio, *The Golden Ratio*. Random House, Inc., 2002.

[14] H. Huntley, *The Divine Proportion*. Dover Publications, 1970.

[15] W. Heisenberg, "Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik," in *Original Scientific Papers - Wissenschaftliche Originalarbeiten*. Springer, 1985, pp. 478–504.

[16] J. Hilgevoord and J. Uffink, "The uncertainty principle," *Stanford Encyclopedia of Philosophy*, 2001.

[17] A. Hodgkin and A. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *Bulletin of mathematical biology*, vol. 52, no. 1-2, pp. 25–71, 1990.

[18] J. Spall, "Adaptive stochastic approximation by the simultaneous perturbation method," *IEEE transactions on automatic control*, vol. 45, no. 10, pp. 1839–1853, 2000.

[19] R. Jacobs, "Increased rates of convergence through learning rate adaptation," *Neural networks*, vol. 1, no. 4, pp. 295–307, 1988.

[20] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization." *Journal of machine learning research*, vol. 12, no. 7, 2011.

[22] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[23] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization." *Journal of machine learning research*, vol. 13, no. 2, 2012.

[24] H. Li, P. Chaudhari, H. Yang, M. Lam, A. Ravichandran, R. Bhotika, and S. Soatto, "Rethinking the hyperparameters for fine-tuning," *arXiv preprint arXiv:2002.11770*, 2020.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[27] MathWorks, *Data Sets for Deep Learning*, accessed March 4, 2022. [Online]. Available: https://www.mathworks.com/help/deeplearning/ug/data-sets-for-deep-learning.html

[28] Y. LeCun, C. Cortes, and C. Burges, *The MNIST Database*, accessed March 4, 2022. [Online]. Available: http://yann.lecun.com/exdb/mnist/