Neutralized Synchronic and Diachronic Potentiality for Interpreting Multi-Layered Neural Networks

Ryotaro Kamimura
Tokai University
Kitakaname, Hiratsuka, Kanagawa, Japan
e-mail: ryotarokami@gmail.com

***Abstract*—The present paper aims to propose a new method, called "di-synchronic potentiality," to unify or neutralize diachronic and synchronic potentialities in order to seek for the simplest form of a network, referred to as the prototype. This method is necessary because the prototype is deeply hidden within surface networks, making it challenging to detect. The synchronic potentiality is measured by the complexity of connection weights at a specific learning time, while the diachronic potentiality is time-dependent. These potentialities tend to decrease as a natural property of learning. While this reduction is effective in eliminating unnecessary weights, it may also lead to the eventual elimination of important weights. The di-synchronic potentiality aims to unify or neutralize the reduction forces of diachronic and synchronic potentialities to increase synchronic potentiality. With this method, the synchronic potentiality does not necessarily increase, but at the very least, the reductive force is weakened or neutralized for solving the collision between the two types of reduction forces. The method was applied to artificial data simulating the bankruptcy of companies, with both linear and nonlinear relations between inputs and targets. The results confirmed that there were strong and repeated forces striving to obtain the simplest form of potentiality. Ultimately, the method successfully produced representations with improved generalization, while simultaneously achieving the simplest relations between inputs and outputs for easier interpretation. From these results, we can conclude that detecting the simplest prototype can eventually lead to discovering more complex yet interpretable relations between inputs and outputs.**

Keywords-diachronic; synchronic; di-synchronic; interpretation; potentiality; prototype

I. Introduction

A. Simplification Forces

The present paper aims to demonstrate the existence of simplification forces in neural networks. These forces can be observed through the simplest network, called the "prototype network," which is deeply hidden within a vast number of different surface networks. Naturally, as the complexity of neural networks and input patterns increases, this simplification hypothesis might seem irrelevant for practical purposes. However, neural networks attempt to replicate only a small fraction of cognitive functions. Compared with the human brain, the primary learning mechanisms in neural networks are expected to be simpler and more straightforward than we might imagine. This implies that even as the complexity of modern neural networks and their input patterns appears to grow, the fundamental inference mechanism remains inherently simple.

Furthermore, the existence of simplification forces is practically useful for achieving simplified interpretations. Neural networks have been applied across many fields, but a major challenge is understanding the inference mechanisms behind these complex systems. If simplification forces exist, the key task in interpreting neural networks becomes understanding the simplified networks resulting from these forces. Naturally, more complex features beyond the simplified ones may be necessary for addressing practical problems. However, if we assume that even in complex networks, much simpler prototype networks operate at the deepest level, interpretation becomes significantly easier. This suggests that by understanding and referencing the simplest network, we can better grasp why seemingly complex features are required in actual processing.

B. Neutralized Potentiality Reduction

To simplify multi-layered neural networks, we have proposed the potentiality method to control the number of absolute connection weights. The potentiality is similar to conventional entropy, and as is well known, learning can be represented by entropy reduction [1]. Following this entropy hypothesis, we can assume that learning is a process of reducing the potentiality of networks. One of the major problems is that potentiality reduction focuses solely on decreasing the strength of connection weights, which does not necessarily eliminate unimportant weights. Instead, it may also reduce important weights. Thus, it is crucial to control this reduction process to preserve important information.

To address the issue of exclusive potentiality reduction, we introduce two types of potentiality: "diachronic" and "synchronic" potentiality, and propose a method to neutralize or unify them to weaken the force of potentiality reduction. The synchronic potentiality is equivalent to the conventional potentiality or entropy defined without considering time-dependent changes during the learning process. In contrast, diachronic potentiality represents the time-dependent learning process. Like synchronic potentiality, diachronic potentiality also aims to reduce the corresponding potentiality. To weaken the forces of synchronic and diachronic potentiality reduction, we embed the diachronic potentiality into the synchronic potentiality. This approach is referred to as diachronized synchronic potentiality or "di-synchronic potentiality." The control of di-synchronic potentiality seeks to simultaneously reduce information and maximize it, or at least weaken the force of potentiality reduction through the interplay between these two

 ISBN: 978-1-68558-260-9

types of potentiality reduction. Similar to the hypothesis that important information should be maximized while unnecessary information should be minimized, this di-synchronic potentiality control incorporates the effect of potentiality maximization into the process of potentiality minimization.

C. Detecting the Prototype Network

This di-synchronic potentiality control can be used to detect the simplest form of a network, or the prototype network, deeply hidden within seemingly complicated surface neural networks. In this paper, we do not extract the prototype network directly from input patterns; rather, the prototype is expected to be self-organized if all necessary network components are provided before learning. We assume that behind any multi-layered neural network, there exists the simplest network without hidden layers, where all components are linearly and separately connected. These properties are not derived from receiving input patterns but are ideally self-organized, independently of any input patterns. However, in practice, it is impossible to achieve full self-organization automatically. Thus, it is more accurate and moderate to state that the prototype can only be self-organized by processing some input patterns initially. The prototype is self-organized by incorporating information necessary to conform to the norms inherent to the prototype. The di-synchronic potentiality is introduced to extract this prototype, which is deeply hidden within seemingly complex network configurations and input patterns. We aim to control the potentiality until we uncover the simplest form of the prototype, even at significant cost.

D. Main Contributions

Then, the outline and the main contributions of this paper can be summarized as follows:

- The present paper aims to clarify the simplification forces hidden in multi-layered neural networks, which are assumed to be represented in terms of the simplest prototype networks.
- These prototypes are presumed to be deeply hidden within surface neural networks, and it is necessary to uncover them using specific methods. We have proposed the potentiality method to reduce and simplify the configuration of connection weights. However, it is challenging to preserve important connection weights through potentiality reduction alone. In this context, we propose an additional potentiality method to increase the potentiality or at least to weaken the reduction forces, thereby controlling the process of potentiality reduction.
- The new method is realized by combining the conventional synchronic potentiality with diachronic potentiality, resulting in a new type of potentiality called “di-synchronic potentiality.” This di-synchronic potentiality is achieved by deploying synchronic potentiality over diachronic processes or learning processes. While this potentiality does not necessarily increase the synchronic potentiality, it can at least weaken the reduction forces of synchronic potentiality.

E. Paper Organization

In Section 2, we summarize several related works, such as internal, prototype, interpretable, distilling, mutual information simplification. In Section 3, we present how to compute the synchronic and diachronic potentiality, and the di-synchronic potentiality. In Section 4, we applied the method to an artificial data set designed to simulate a business data set, incorporating both linear and non-linear relations between inputs and targets. The results showed that the reduction of synchronic potentiality was weakened, preventing the network from excessively reducing the potentiality. The di-synchronic potentiality repeatedly and intensely sought to extract the prototype during the learning process. Connection weights became closer to those of the assumed prototype, and improved generalization was observed. Ultimately, the results confirmed that as interpretation became easier, generalization performance also improved.

II. Related Work

A. Internal and External Simplification

As mentioned in the introductory section, one of the major problems in neural networks is the inability to interpret the inference mechanism. To address this issue, many different types of interpretation methods have been developed, all of which are directly related to simplifying network configurations and input patterns. To explain the main characteristics of our method in comparison to conventional simplification methods, we introduce two types of simplification: external and internal simplification. External simplification aims to simplify the network configuration depending on inputs and, more directly, to simplify input patterns themselves. In contrast, internal simplification focuses not on input patterns but on the network itself. Ideally, the network should be self-organized without external stimuli. However, self-organization without any external stimuli is impossible, so this simplification is moderated to allow networks to self-organize with some initial external stimuli. We explain four simplification procedures in comparison to our method, namely, prototype simplification, interpretable simplification, distilling simplification, and mutual information-theoretic simplification.

B. Prototype Simplification

The prototype approach has been one of the major simplification procedures in neural networks. Here, we use the prototype to describe the simplest form achievable through the simplification forces inherent in the network. Prototype-based approaches have been studied extensively since the early stages of learning, under names, such as vector quantization, competitive learning, and self-organizing maps [2]. These methods aim to identify a small number of representative vectors for many input patterns. In recent deep learning research, prototype learning has gained attention [3], as the volume of input patterns to be processed has grown larger and increasingly heterogeneous. Simplifying these complex and heterogeneous patterns has become urgent, giving rise to methods like one-shot and few-shot learning to represent many inputs using a

few representative patterns. Similarly, zero-shot learning [4] has been developed to incorporate the abstract and semantic properties of input patterns. These methods exemplify external simplification, representing a large number of input patterns with a smaller set of representative inputs and more abstract features. In contrast, our approach focuses on the network itself, aiming to determine which network configuration should be organized when all network components are given before learning.

C. Interpretable Simplification

Second, most interpretation methods can be classified as external simplification. Among these, one prevalent approach is localized interpretation, which focuses on specific instances rather than the overall inference mechanism. Unlike global interpretation methods [5]–[7], localized interpretation has proven effective in practical applications due to its simplicity and the urgent need for explainability. Linear and local simplifications have been widely used in interpretation methods, replacing complex non-linear models with corresponding linear models [8]–[10]. These simplified local models aim to interpret surface networks, which are diverse and heterogeneous. However, our approach seeks to uncover the prototype hidden deeply within surface models. Surface models are produced through transformational rules and may be too complex to understand, but the underlying prototype is expected to be much easier to interpret.

D. Distilling Simplification

Third, network compression simplifies neural networks by replacing complex networks with smaller ones [11]–[13]. This type of simplification has become increasingly necessary as networks grow larger to process more input patterns. However, interpreting how all components in such networks operate to produce outputs is a significant challenge. Most compression methods represent typical external simplification, replacing the original network with smaller, unrelated networks. These methods attempt to interpret the smaller "student" networks under the assumption that their inference mechanism is similar to the original network's. However, this assumption may not always hold true. In contrast, the internal simplification we propose compresses the original multi-layered neural network directly, retaining the original network's information. Thus, interpreting and explaining the compressed networks is more directly connected to the original multi-layered networks.

E. Mutual Information-Theoretic Simplification

Fourth, information-theoretic methods also relate to network simplification, albeit in more abstract ways. Multiple network configurations can be represented by more simple and abstract information content, and the objective of learning can be considered as necessary information acquisition. In particular, mutual information has played a significant role in neural networks. Notable examples include the well-known maximum mutual information preservation [14], which aims to retain as much relevant information as possible, and the information bottleneck method [15], which seeks to maximize necessary information while minimizing unnecessary information. Despite their utility, mutual information-based methods face challenges, such as computational complexity and ambiguous interpretations of the results. These issues arise because mutual information involves contradictory operations: entropy maximization and conditional entropy minimization. Recent information bottleneck methods introduce additional mutual information computations to balance compression and relevant information preservation [16], but computational complexity remains a challenge.

Our bi-synchronic potentiality method shares the goal of acquiring simplified, necessary information but differs in three key ways. First, it focuses on information necessary for self-configuring neural networks with minimal external input, deepening the level of simplification compared to conventional methods. Second, it embeds diachronic potentiality reduction into synchronic potentiality reduction, neutralizing and weakening potentiality reduction forces. Third, it aims to make network interpretation easier by transitioning from surface-level to prototype-level interpretation. Fundamentally, our method seeks to examine the information required to configure networks when all resources are provided before learning begins. Unlike conventional information-theoretic methods that optimize configurations to represent input patterns efficiently, our approach tries to create a framework to prepare for potential input patterns.

III. Theory and Computational Methods

A. Synchronic and Diachronic Potentiality

The prototype represents the state of network configuration in terms of connection weights. So far, we have defined the potentiality without considering time-dependent factors. This paper introduces a time-dependent potentiality, termed "diachronic potentiality," while the original potentiality, which does not account for time-dependent factors, is referred to as synchronic potentiality. The potentiality function was developed to simplify the conventional entropy function. By omitting the logarithmic function, potentiality becomes computationally efficient and easier to interpret. One major issue is that potentiality is designed to decrease its strength. Minimizing synchronic potentiality is expected to reduce redundant information and highlight important information. However, this reduction does not necessarily extract important information, as it focuses solely on reducing strength without adequately considering the properties of network components.

To address this issue, we introduce a factor that augments potentiality or at least weakens potentiality reduction to enhance the extraction of important information. For this purpose, we define diachronic potentiality to capture the time-dependent properties of learning. Diachronic potentiality characterizes the entire learning process and naturally tends to decrease as learning progresses, similar to synchronic potentiality. By embedding diachronic potentiality into synchronic potentiality, the reduction effect can be weakened through a process called neutralization. This combined potentiality

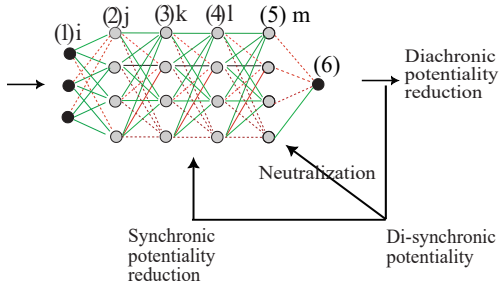


Figure 1. Synchronic, diachronic and di-synchronic potentiality by neutralization.

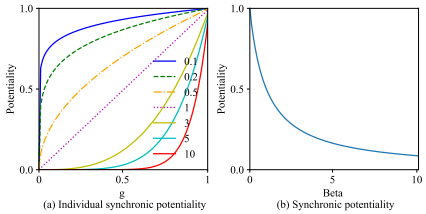


Figure 2. Individual synchronic potentiality (a) and synchronic potentiality as a function of the parameter β_{syn} (b). The potentialities are normalized for easy comparison.

is referred to as "di-synchronic potentiality," reflecting the integration of diachronic and synchronic aspects of potentiality reduction. Figure 1 illustrates the concept of this neutralization. In the figure, diachronic potentiality operates throughout all learning steps, while synchronic potentiality is effective only at specific steps of the overall learning process. Both types of potentiality aim to reduce their respective strengths. However, through neutralization, achieved by embedding diachronic potentiality into synchronic potentiality, the two can be unified. The concept of neutralization signifies an effort to increase synchronic potentiality or, when this is not feasible, to mitigate the strength of synchronic potentiality reduction.

B. Synchronic Potentiality Reduction

Let us begin with the definition of synchronic potentiality, as all other types of potentiality are based on this concept. The synchronic potentiality is defined using the absolute values of connection weights. For simplicity, we consider only one hidden layer, from the nth layer to the $n+1$th layer, denoted as $(n, n+1)$, as shown in Figure 1. The individual synchronic potentiality for $(n, n+1)$ is computed as:

$$u_{jk}^{(n,n+1)} = \mid w_{jk}^{(n,n+1)} \mid, \tag{1}$$

where all weights are assumed to be greater than zero for simplicity. By normalizing with the maximum value, we define the relative synchronic potentiality as:

$$g_{jk}^{(n,n+1)} = \gamma_{syn} \left[\frac{u_{jk}^{(n,n+1)}}{\max_{j'k'} u_{j'k'}^{(n,n+1)}} \right]^{\beta_{syn}}, \tag{2}$$

where the maximum operation is taken over all connection weights in the layer, and β_{syn} is a parameter controlling the strength of the potentiality. The parameter γ_{syn} is introduced for computational purposes, as this potentiality may excessively reduce the strength of connection weights. By summing all the relative potentialities, we obtain the final form of synchronic potentiality:

$$G^{(n,n+1)} = \gamma_{syn} \sum_{jk} \left[\frac{u_{jk}^{(n,n+1)}}{\max_{j'k'} u_{j'k'}^{(n,n+1)}} \right]^{\beta_{syn}}. \tag{3}$$

Figure 2(a) illustrates synchronic potentiality as a function of the strength of individual synchronic potentiality. As the parameter β_{syn} increases, the majority of connection weights are forced to decrease, thereby reducing the potentiality. Figure 2(b) shows synchronic potentiality as a function of the parameter β_{syn}. As explained above, synchronic potentiality decreases gradually with increasing parameter strength.

New weights at the $(t+1)$th learning step are obtained by multiplying the weights at the tth step by the corresponding potentiality:

$$w_{jk}^{(n,n+1)}(t+1) = \gamma_{syn} \left[\frac{u_{jk}^{(n,n+1)}}{\max_{j'k'} u_{j'k'}^{(n,n+1)}} \right]^{\beta_{syn}} w_{jk}^{(n,n+1)}(t). \tag{4}$$

In this learning rule, the individual potentiality is multiplied by the corresponding weight. The weights are incorporated to facilitate learning, as the strength of connection weights must be considered for effective learning. Learning can theoretically proceed with only the potentiality term, without actual weights, although the parameter γ_{syn} would need careful tuning for stable learning. In this paper, we adopt this formulation for computational convenience.

C. Diachronic Potentiality Reduction

The diachronic potentiality is defined by considering the entire sequence of learning steps. The individual diachronic potentiality at the tth learning step (epoch) is defined solely based on the time step t:

$$v_t = t. \tag{5}$$

The relative individual diachronic potentiality is then obtained by normalizing the potentiality with its maximum value:

$$h_t = \gamma_{dch} \left[\frac{v_t}{\max_{t'} v_{t'}} \right]^{\beta_{dch}}, \tag{6}$$

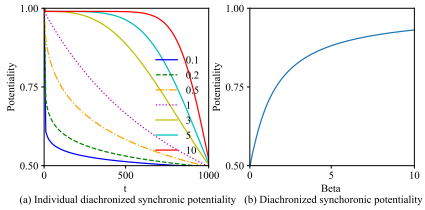
where γ_{dch} is a scaling parameter, and β_{dch} controls the strength of the diachronic potentiality. The total diachronic potentiality is defined as:

$$H = \gamma_{dch} \sum_{t} \left[\frac{v_t}{\max_{t'} v_{t'}} \right]^{\beta_{dch}}. \tag{7}$$

The final form of the weight update rule is given by:

$$w_{jk}^{(n,n+1)}(t+1) = \gamma_{dch} \left[\frac{v_t}{\max_{t'} v_{t'}} \right]^{\beta_{dch}} w_{jk}^{(n,n+1)}(t). \tag{8}$$

This learning equation indicates that the strength of the weights is reduced progressively over the course of learning.



(a) Individual diachronized synchronic potentiality (b) Diachronized synchoronic potentiality

Figure 3. Individual di-synchronic potentiality (a) and di-synchronic potentiality as a function of the parameter β_{dia} (b).

The effect of the diachronic potentiality increases as the learning step progresses. As the diachronic parameter β_{dch} increases, the diachronic potentiality is forced to become smaller, signifying the application of stronger potentiality reduction. The diachronic potentiality is defined in relation to the corresponding synchronic potentiality, and it tends to decrease as the parameter β_{dch} increases, as illustrated in Figure 2.

D. Di-Synchronic Potentiality Augmentation

We aim to neutralize the synchronic and diachronic potentiality into a new form, called the "di-synchronic potentiality." Specifically, we attempt to reduce the effects of synchronic and diachronic potentiality reduction to a certain extent. As mentioned above, this process is referred to as neutralization, where we balance the forces of potentiality reduction. One possible approach is to replace the synchronic parameter β_{syn} with the diachronic individual potentiality:

$$\text{neutral}_{jk}^{(n,n+1)}(t) = \gamma_{syn} \left[\frac{u_{jk}^{(n,n+1)}}{\max_{j'k'} u_{j'k'}^{(n,n+1)}} \right]^{h_t}, \quad (9)$$

where h_t is the individual relative diachronic potentiality. This embedding is shown to be effective for neutralization.

Figure 3(a) illustrates the individual di-synchronic potentiality as a function of the number of learning steps for different values of the parameter β_{dia}. As shown in the figure, as the strength of the parameter increases, the strength of the individual potentiality also increases. Figure 3(b) presents the di-synchronic potentiality as a function of the parameter β_{dia}. It can be observed that the di-synchronic potentiality gradually increases as the parameter increases. This implies that while the diachronic potentiality decreases as the diachronic parameter increases, the di-synchronic potentiality can still be enhanced.

The learning procedure is finally expressed as:

$$w_{jk}^{(n,n+1)}(t+1) = \text{neutral}_{jk}^{(n,n+1)}(t)\; w_{jk}^{(n,n+1)}(t). \quad (10)$$

By employing this procedure, we aim to augment the synchronic potentiality while neutralizing the effects of potentiality reduction.

Lastly, it is important to note that several additional potentialities are required to fully describe the learning process. Due to page limitations, explanations are provided as needed.

IV. Results and Discussion

A. Experiment Outline

The data set was created by imitating an actual business data set used to estimate the bankruptcy of companies [17]. The objective of the experiment is not to improve generalization but to interpret how bankruptcy occurs and what the major causes of bankruptcy are. In more technical terms, we aim to understand the relationships between inputs and outputs. In the actual data set, linear and non-linear relationships are naturally mixed, making it seemingly impossible to explicitly understand the relationships between inputs and outputs. To address this situation, we created a data set with both linear and non-linear relationships between inputs and targets artificially. This artificial data set allows us to explicitly understand how neural networks respond to specific inputs to produce outputs.

The number of input variables was seven. Of these, input No.6 (financial stability) and input No.7 (market sentiment) were created using exponential functions to stress the lower or higher values. Input No.5 was also created non-linearly using the sine function to represent large variations. Figure 4 shows the actual normalized correlation coefficients between inputs and outputs. Inputs No.1 to No.4 were created linearly, while the remaining inputs were created non-linearly, as described above. For example, input No.7 showed the lowest correlation coefficient among the inputs but was related non-linearly to the target. The problem is to examine whether the potentiality method can distinguish between these two types of inputs.

The number of input patterns was 1000, and the number of hidden layers was ten, with ten neurons in each hidden layer. We used the PyTorch program package, where almost all parameter values were set to default values to ensure easy reproduction of the results presented here. The experiment was designed to make the neural networks as close as possible to the prototype network, which is assumed to be hidden within the surface networks. The prototype network is the simplest form, and the connection weights in this paper were computed using the correlation coefficient between inputs and targets of the training data set. Next, we attempted to use the di-synchronic potentiality to neutralize the effect of synchronic potentiality reduction. The final multi-layered neural network was compressed into the simplest network without hidden layers. Compression was conducted layer by layer. We then compared the estimated and compressed networks with the assumed simplest network. Our objective is to demonstrate that the di-synchronic potentiality could be used to make the actual neural networks as close as possible to the prototype network.

Now, the main findings can be summarized as follows:

- Our method was able to weaken the forces of synchronic potentiality reduction. This means that the di-synchronic potentiality was effective in controlling the process of synchronic potentiality reduction.
- The results confirmed that the di-synchronic potentiality could regulate the synchronic potentiality to increase gener-

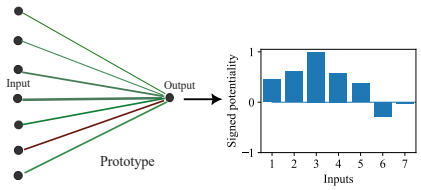


Figure 4. Supposed prototype (left) and the strength of connection weights (right).

alization performance by increasing the diachronic parameter β_{dia}.

- In addition, the di-synchronic potentiality repeatedly attempted to increase the ratio of potentiality, measuring similarity to the supposed prototype, even in the later stages of learning. This indicates that the potentiality could force networks to become closer to the prototype, while maintaining improved generalization.
- The experimental results confirmed that the di-synchronic potentiality could simplify a ten-layered neural network into the simplest prototype network with better generalization. We were able to simplify the internal representations created by the neural network, while still preserving improved generalization.

B. Potentiality and Generalization

The results confirmed that the di-synchronic potentiality, controlled by the diachronic parameter β_{dia}, could increase the synchronic potentiality, though moderately. This potentiality augmentation was associated with improved generalization.

Figure 5 shows the synchronic potentiality (left), entropy (middle), and testing and validation accuracy (right) as a function of the number of steps. When the linear activation function was used in Figure 5(a), both the potentiality and entropy took relatively higher values, but they remained almost unchanged, as the potentiality control was not introduced. When the parameter β_{dia} increased from 0 (b) to 1.2 (f), the synchronic potentiality (left) tended to have higher values. Since the differences were relatively small, we plotted these potentialities in one graph in Figure 6. It is clear that the values of synchronic potentiality increased gradually as the parameter β_{dia} increased. This indicates that, although the synchronic potentiality tended to decrease, a force to increase it could be observed as the parameter β_{dia} increased. In addition, the entropy (middle) exhibited this trend more clearly. When the parameter β_{dia} was small, the entropy remained higher in the early stages of learning and then decreased considerably. As the parameter increased further, the entropy tended to stay at higher values throughout the learning process.

The right figures in Figure 5 show testing and validation accuracy as a function of the number of steps. Generalization accuracy initially increased and then decreased rapidly as the parameter β_{dia} increased from 0 (b) to 0.4 (d). When the parameter was relatively small, the effect of synchronic potentiality tended to be stronger. As the parameter increased further, generalization accuracy gradually stabilized and began

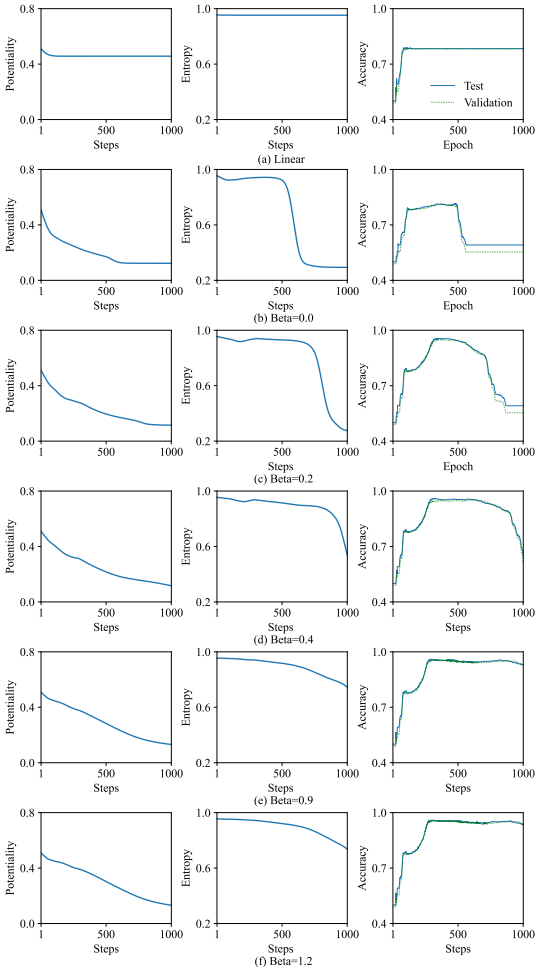


Figure 5. Synchronic potentiality (left), entropy (middle), and testing and validation accuracy (right) as a function of the number of steps (epochs), when the parameter β_{dia} increased from 0 (b) to 1.2 (f). In addition, results with the linear activation function was added (a).

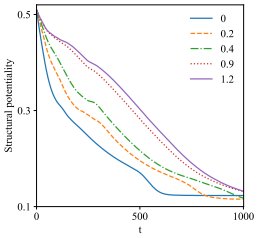


Figure 6. Synchronic potentialities as a function of the number of steps, when the parameter β_{dia} increased from 0 to 1.2.

to show relatively higher values. Finally, when the parameter reached 1.2 (f), the generalization accuracy was the highest.

The di-synchronic potentiality was shown to be effective in weakening the synchronic potentiality reduction, and this effect was clearly related to improved generalization.

C. Ratio Potentiality

Then, we examined to what extent the final networks could be close to the supposed prototype network. We used the ratio potentiality, representing the ratio of estimated potentiality to the supposed potentiality. As the ratio potentiality increases,

the estimated prototype becomes closer to the supposed prototype. The results confirmed that the ratio potentiality showed higher values initially, and in particular, as the diachronic parameter increased, the force to achieve higher ratio values became stronger. Eventually, when the diachronic parameter increased to 1.2, which produced the best generalization, we could extract relatively higher ratio values three times during the learning process.

Figure 7 shows the ratio potentiality (left), divergence (middle), and correlation coefficients between the supposed and estimated prototypes (right). First, we observed that the correlation coefficients between the estimated and supposed prototypes were much higher, indicating close relations between the two prototypes. The ratio potentiality (left) remained almost flat throughout the entire learning process with the linear activation function (a). When the parameter β_{dia} was zero, the ratio potentiality tended to have higher values in the initial stages of learning, and then decreased. When the parameter β_{dia} increased to 0.2 (c) and 0.4 (d), only one peak appeared at the beginning of the learning process. When the parameter increased to 0.9 (e), two peaks in the ratio potentiality were observed. Finally, when the parameter reached 1.2 (f), three peaks were evident. This indicates that the di-synchronic potentiality forced the networks to resemble the prototype as much as possible. Comparing the results with those obtained from the divergence (middle) and correlation coefficients (right), the ratio potentiality more explicitly demonstrated the tendency to acquire the prototype.

The results confirmed that simplification forces were present in the neural networks, as evidenced by the higher ratio potentiality. Additionally, we observed many peaks in the ratio potentialities, meaning that these simplification forces were very strong throughout the entire learning process.

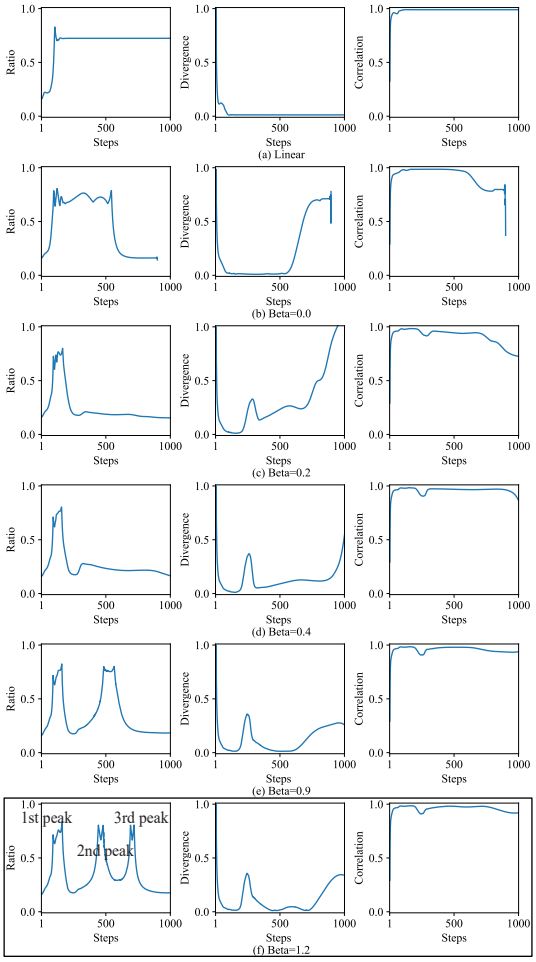


Figure 7. Ratio potentialities (left) and divergence (middle), and correlation coefficient (right), when the linear activation function was used (a), and the parameter β_{dia} increased from 0 (b) to 1.2 (f).

D. Interpretation of Rotating Individual Ratio Potentialities

1) Signed Individual Potentialities: The results confirmed that the signed individual potentialities, corresponding to the actual connection weights, were close to the prototype. Then, when we examined the ratio potentialities, the three ratio peaks exhibited higher ratio potentialities, indicating that the networks were close to the supposed prototype three times during the entire learning process.

Figure 8 (a) shows signed individual potentialities, namely, actual and normalized connection weights, when the parameter β_{dia} increased to 1.2, resulting in the best generalization. Comparing with the supposed prototype in Figure 4, the potentialities were quite close to the supposed prototype. However, gradually, the final input No.7 tended to become stronger. Figure 8 (b) shows individual ratio potentialities. For the three peaks, the potentialities were considerably high, indicating that the final connection weights were close to the prototype. However, in other cases, only input No.7 had overwhelmingly large values. The input No.7 was created with non-linear relations between inputs and outputs.

The results confirmed that there were strong simplification forces in the neural networks. By examining these individual potentialities, we observed that improved generalization could be obtained not only by detecting non-linear relations between inputs and outputs but also by clearly detecting linear relations.

2) Interpretation by Rotation of Individual Ratio Potentialities: We present the results by rotating the potentialities as the parameter β_{rat} of the ratio potentiality increases. As the parameter increased, the number of stronger potentialities became smaller. For the three peaks, the linear input No.4 and the non-linear inputs No.6 and No.7 were used to infer the outputs. The combination of linear and non-linear relations could be used to improve generalization.

For easier understanding of the ratio potentialities, we rotated the potentialities by changing the parameter β_{rat} for the ratio potentialities in Figure 9. When the parameter was 0.1 (a), almost all ratio potentialities became higher. As the parameter increased from 1 (b) to 5 (d), a progressively smaller number of potentialities remained relatively stronger. For the three peaks, input No.4, No.6, and No.7 became relatively higher and more important. Input No.4 was created with a linear relation, while the other two inputs were created with non-linear relations. Thus, the combination of linear and non-

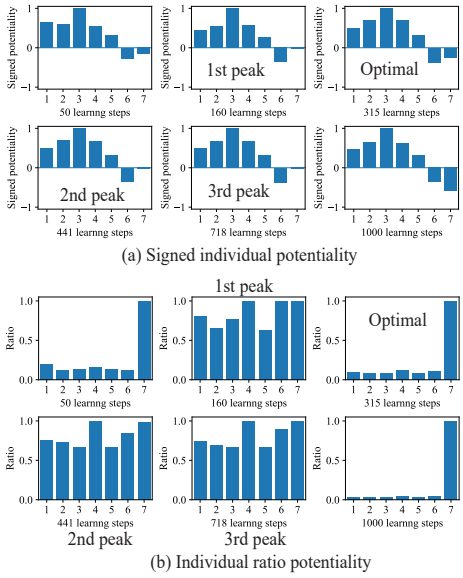


Figure 8. Individual ratio potentiates, when the parameter β was 1.2 with the best generalization.

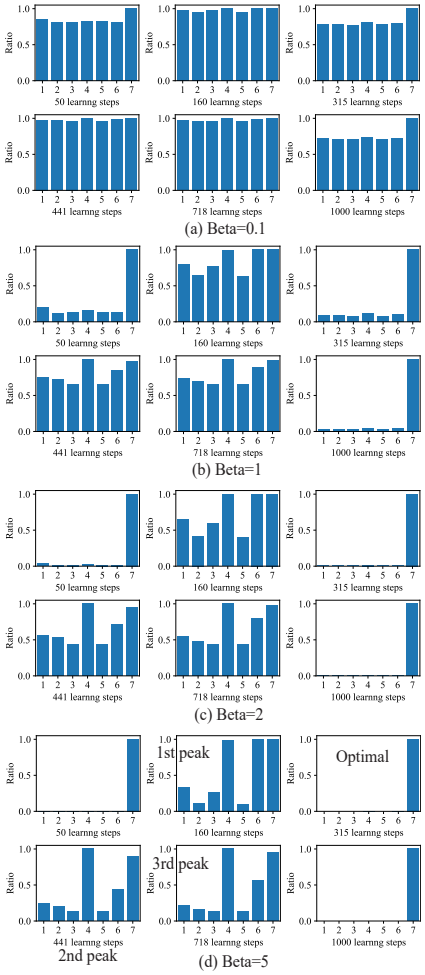


Figure 9. Rotating the ratio potentiality, when the parameter β_{rat} was increased from 0.1 (a) to 5 (d).

linear relations should be used to improve generalization. However, for the three peaks, the ratio potentiality took very high values, and the relations between inputs and outputs could be interpreted easily by examining the linear correlation coefficients between inputs and outputs.

The results confirmed that the networks could show improved generalization based exclusively on the non-linear relation between inputs and outputs, but they could also produce approximately the same results with both linear and non-linear relations.

E. Prototype-based Interpretation

Finally, we should again examine the highest peaks in the ratio potentiality. The results confirmed that the networks could be close to the linear prototype, achieving improved generalization almost equivalent to the highest value by focusing on the non-linear relations. This means that the bi-synchronic potentiality could produce simplified and linear connection weights for easier interpretation with better generalization.

Table I shows the summary of the three peaks of ratio potentiality when the parameter was 1.2, with the best generalization. As shown in the table, the highest testing accuracy was 0.959 with 314 learning steps. However, the ratio potentiality was very low (0.224), and the divergence was larger (0.108). As explained in the previous section, this highest peak was obtained by considering exclusively input No.7 with non-linear relations.

For the first peak of ratio potentiality, with 159 learning steps, the ratio potentiality became higher (0.836), and the divergence was smaller (0.014). However, the testing accuracy was the lowest (0.783). This suggests that it seems impossible to increase generalization with the simple linear relations in the prototype network. However, for the second peak, the ratio potentiality was also quite large (0.805), and the divergence was the smallest (0.011). In particular, the testing accuracy was 0.953, which was quite large, though smaller than that of the optimal case. For the third peak, with 717 steps, the ratio potentiality was still high (0.806), and the divergence was the second lowest (0.012). The generalization accuracy was 0.939, which was smaller than that of the optimal case, but sufficiently large.

This shows that the networks could be similar to the prototype network, whose simple relations between components make it possible to interpret the final results more easily, while still maintaining improved generalization.

TABLE I. TESTING ACCURACY, WHEN THE RATIO POTENTIALITY TOOK THE PEAKS WITH THE OPTIMAL CASE AND WITH THE BEST GENERALIZATION PERFORMANCE.

Peak	Step	Ratio Potent	Divergence	Testing
1st	159	0.836	0.014	0.783
2nd	440	0.805	0.011	0.953
3rd	717	0.806	0.012	0.939
Optimal	314	0.224	0.108	0.959

V. Conclusion

The present paper aimed to show that there are strong simplification forces in neural networks. The simplest form is realized in terms of the prototype. However, the prototype is almost always deeply hidden behind seemingly complicated surface networks, and we need to develop a method to search for the hidden prototypes at any cost. To obtain the prototype network, we need to increase important information while simultaneously decreasing unnecessary information. Our method of potentiality, which is closely related to informational entropy, is effective at reducing potentiality or entropy, but it is very challenging to augment it in the presence of strong forces of potentiality reduction. To address this problem, we introduce two types of potentiality: synchronic and diachronic potentiality. Both of these potentialities tend to reduce the corresponding strength. However, by combining and neutralizing the synchronic potentiality with the diachronic potentiality, or by using di-synchronic potentiality, it becomes possible to augment synchronic potentiality in a diachronic way, or at least weaken its tendency to reduce. We applied this method to an artificial data set in which linear and non-linear relations were explicitly included. The results showed that di-synchronic potentiality was effective in weakening synchronic potentiality reduction during the learning process. With this method, the networks tried to produce the simplest networks, close to the supposed prototype networks, with improved generalization. This implies that we could generate networks whose generalization performance was better while simultaneously making the relations between inputs and outputs easier to understand in a linear way.

For future work, we should mention three points. First, the relationship between diachronic and synchronic potentiality has not been fully analyzed at this stage. It is necessary to examine more carefully how diachronic and synchronic potentiality should be interrelated. Second, our method has been inspired by the information-theoretic approach to neural networks, such as mutual information and the information bottleneck. We need to examine more carefully the relations between our method and these more mathematically-oriented approaches. Finally, the data set used in this paper was artificially created to examine how well our method can distinguish between linear and non-linear relations. It is naturally necessary for this method to be applied to larger-scale and more practical data sets.

References

[1] S. Watanabe, *Knowing and guessing: A quantitative study of inference and information*. New York: John Wiley and Sons Inc, 1969.

[2] P. Schneider, M. Biehl, and B. Hammer, "Adaptive relevance matrices in learning vector quantization," *Neural computation*, vol. 21, no. 12, pp. 3532–3561, 2009.

[3] M. Biehl, B. Hammer, and T. Villmann, "Prototype-based models in machine learning," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 7, no. 2, pp. 92–111, 2016.

[4] F. Pourpanah *et al.*, "A review of generalized zero-shot learning methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 4, pp. 4051–4070, 2022.

[5] D. Alvarez Melis and T. Jaakkola, "Towards robust interpretability with self-explaining neural networks," *Advances in Neural Information Processing Systems*, vol. 31, pp. 7775–7784, 2018.

[6] C. Yang, A. Rangarajan, and S. Ranka, "Global model interpretation via recursive partitioning," in *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, IEEE, 2018, pp. 1563–1570.

[7] S. M. Lundberg *et al.*, "From local explanations to global understanding with explainable ai for trees," *Nature machine intelligence*, vol. 2, no. 1, pp. 2522–5839, 2020.

[8] G. Alain, "Understanding intermediate layers using linear classifier probes," *arXiv preprint arXiv:1610.01644*, 2016.

[9] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, ACM, 2016, pp. 1135–1144.

[10] D. Garreau and U. Luxburg, "Explaining the explainer: A first theoretical analysis of lime," in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2020, pp. 1287–1296.

[11] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *stat*, vol. 1050, p. 9, 2015.

[12] P. Luo, Z. Zhu, Z. Liu, X. Wang, and X. Tang, "Face model compression by distilling knowledge from neurons," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[13] R. Mishra, H. P. Gupta, and T. Dutta, "A survey on deep neural network compression: Challenges, overview, and solutions," *arXiv preprint arXiv:2010.03954*, 2020.

[14] R. Linsker, "Perceptual neural organization: Some approaches based on network models and information theory," *Annual review of Neuroscience*, vol. 13, no. 1, pp. 257–281, 1990.

[15] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *2015 ieee information theory workshop (itw)*, IEEE, 2015, pp. 1–5.

[16] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," *arXiv preprint arXiv:1612.00410*, 2016.

[17] K. Shimizu, *Multivariate analysis (in Japanese)*. Nikkan Kogyo Shinbun, 2009.