Self-Competitive Simplification: Competition between Forward and Backward Simplification in Multi-Layered Neural Networks

Ryotaro Kamimura
Tokai University
Kitakaname, Hiratsuka, Kanagawa 259-1292, Japan
e-mail: ryotarokami@gmail.com

***Abstract*—The present paper aims to demonstrate that strong forces of simplification exist within neural networks. These forces compete with one another to make the simplification process as effective as possible. As a first approximation, we consider four types of simplification forces: forward, backward, collective, and individual simplification. The winners in the competition among these forces can efficiently simplify the network configuration, whereas the losers can eventually be utilized to introduce a certain level of complexity, which is required in actual learning. The proposed method was applied to a simple artificial data set containing both linear and non-linear inputs. When backward and forward simplification were forced to compete, forward simplification became more efficient in achieving simplification, while backward simplification played a complementary role by introducing additional complexity for improved generalization. These results suggest that, at a deeper level, multiple simplification forces coexist within neural networks and compete with one another to achieve efficient simplification.**

Keywords-competition; simplification; complexity; forward; backward; collective; individual; cost.

I. Introduction

This section explains the concept of self-competitive simplification, along with a brief introduction to the actual computational procedures.

A. Self-Learning

The present paper aims to demonstrate that neural networks have a strong intrinsic tendency toward simplification, which should ideally be observed even in the absence of external information. Previous studies on neural networks have primarily focused on representing input patterns as faithfully as possible, including efforts to obtain representations that lead to improved generalization. In contrast, the present study emphasizes the necessity of examining neural networks not only in relation to input patterns but also in terms of their internal structure and dynamics. This perspective suggests that we should explore the inner workings of neural networks, which are assumed to operate even without external information. This internal perspective is referred to as “self-learning,” in which a network is ideally configured without external information, or at least self-organizes when triggered by only a very small amount of external input. When attention is shifted from external information to internal information, it becomes easier to observe how a network operates and, in particular, to identify the fundamental limitations of neural networks.

B. Self-Competitive Simplification for Complexity

At first glance, the simplification principle underlying self-learning appears to contradict the complexity required in network configurations during actual learning. To address this apparent contradiction, we hypothesize that such complexity arises from the coexistence of many different types of simplification procedures within a network. The simplification principle attempts to achieve simplification through all possible means, thereby producing various types of components and computational procedures that are suitable for simplification. These procedures compete with each other to make the network configuration as simple as possible. Some components or procedures win this competition, while others lose. Competition naturally produces losers with respect to simplification; however, from the perspective of actual learning, which requires complexity, these losing components may acquire external information and thereby increase complexity. Severe competition among multiple simplification procedures eventually creates room for the introduction of complexity. In this sense, losers in self-competition can become winners in acquiring external information during actual learning. It is important to note that, beneath the apparently complex network configurations observed at the surface level, there exists a simpler configuration at a deeper level, formed as a result of intense competition among diverse simplification procedures.

C. Self-Competitive Procedures

The competition among simplification procedures described above is referred to as “self-competition.” This means that competition occurs not between input patterns but among components and learning procedures within the network itself. As a first approximation, we consider four competitive procedures: forward, backward, collective, and individual. In practical situations, such competition is assumed to be triggered by a small amount of external input information. First, competition is assumed to occur between forward and backward information processing. Information is transmitted from input to output in a forward manner, while it can also be propagated backward from output to input. These two modes of information flow compete with each other to facilitate both simplification and error minimization during learning.

In addition, components can be treated either collectively or individually. For example, a set of connection weights or neurons can be treated as a group, while each neuron or weight can also be considered individually. Collective and individual procedures therefore compete with each other to achieve a simplified network configuration. Within this framework, all components and procedures compete, and those that win the competition are utilized more explicitly for simplification. Conversely, procedures that lose the competition for simplification may still be effective in reducing training error, even if they do not directly contribute to simplifying the network configuration.

D. Paper Structure

The rest of the paper is structured as follows. In Section II, we discuss related work on competitive learning. In Section III, we explain how to compute total and forward simplification, as well as the contradiction between collective and individual potentiality. In Section IV, we apply the method to a controlled data set containing both linear and nonlinear relationships. The results show that forward simplification wins the competition and produces efficient simplification. Backward simplification loses the competition but plays a role in introducing the complexity necessary for learning. The combination of forward and backward simplification can be used to achieve simplification while retaining sufficient complexity for effective learning.

II. Related Work

Related to conventional and well-established competitive learning, the method presented here introduces a new concept of competition, which ideally minimizes the influence of external information.

Competitive learning has been regarded as one of the fundamental learning paradigms since the early days of neural network research [1]–[3]. Numerous studies have aimed to extract representative features from input patterns [4]–[9]. Competitive learning has also been implemented as a core mechanism in Self-Organizing Maps (SOMs), a major unsupervised learning framework [10], [11].

In conventional competitive learning, a neuron that best represents the input patterns, often referred to as the best matching neuron, is selected. Such competition focuses on determining which neuron most closely matches the input, and thus which input features should be represented. In this sense, competitive learning, including SOMs, is guided by a simplification principle that aims to represent input patterns as economically as possible within a restricted network structure.

In contrast, we propose that competition should occur not only among input patterns but also among components and learning procedures within the network itself. This implies that competition can exist even without input patterns, at least ideally. We therefore refer to this framework as "self-competition," distinguishing it from conventional competitive learning. In self-competition, components compete with one another to achieve simplification from the very beginning of network formation, either without input patterns or with minimal external information.

At first glance, neural networks may appear to exhibit no explicit competition in their initial configurations. We hypothesize that what is observed is a collection of networks with many peripheral components and procedures, behind which there exists a simpler network in which all components and learning procedures compete to simplify the network configuration as much as possible. To reveal this internal structure, it is necessary to artificially simplify superficial and complex network configurations.

III. Theory and Computational Methods

In this section, after explaining the principles of simplification and competition, we introduce total and forward simplification.

A. Simplification Principle and Competition

The present paper introduces competition among components of neural networks, such as weights and neurons. In addition, computational procedures—including forward, backward, collective, and individual operations—are assumed to compete with each other in order to achieve maximum simplification.

We begin by explaining forward and backward simplification. Figure 1 illustrates the processes of forward and backward simplification. Figure 1(b) shows one of the final configurations obtained through forward simplification. In forward simplification, information entering a neuron is distributed, or de-compressed, to all neurons in the subsequent layer, and this process continues through all layers. Figure 1(d) illustrates the process of backward simplification, in which information in a layer is distributed to all neurons in the corresponding lower layer. We assume that competition between forward and backward simplification occurs, as shown in Figure 1(c).

Collective and individual competition are then applied either to a set of connection weights or to individual connection weights. In forward simplification, as shown in Figure 1(b), a neuron is regarded as being composed of a set of connection weights from all neurons in the lower layer. These neurons, represented as a set of connection weights, should be used as equally as possible, which is referred to as "collective" competition or simplification. In contrast, individual connection weights from neurons in the lower layer to a neuron in the subsequent layer compete with each other, which is referred to as "individual simplification." Backward simplification operates in the same manner as forward simplification.

B. Total Simplification

For simplicity, we consider only a single layer, because the same formulation can be applied to all other layers. The individual potentiality of a connection weight from the jth neuron to the kth neuron is computed as

$$u_{jk} = \mid w_{jk} \mid . \tag{1}$$

As mentioned above, all final values defined in this section can be obtained by averaging over all layers, including the input and output layers. In addition, for simplicity, it is assumed that the strength of all weights is greater than zero.

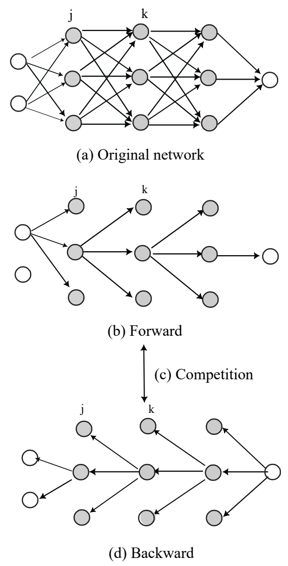

Figure 1. An original network (a), final configurations obtained by forward simplification (b) and backward simplification (d), and their competition (c).

The primary objective of simplification is to reduce the total cost potentiality of a network. The cost, or cost potentiality, is defined as the sum of individual potentialities:

$$C = \frac{1}{n_j n_k} \sum_{jk} u_{jk}, \tag{2}$$

where n_j and n_k denote the numbers of neurons in the corresponding layers.

Simplification aims to reduce this cost potentiality, which can be achieved in various ways.

C. Forward Simplification

We now formulate forward simplification. Collective simplification aims to simplify a group of connections from one layer to the subsequent layer. The potentiality of the kth neuron is defined as the sum of the absolute strengths of the incoming weights:

$$u_k = \sum_j u_{jk}. \tag{3}$$

The potentiality of the kth group is then computed as

$$h_k = \frac{u_k}{\max_{k'} u_{k'}}. \tag{4}$$

The collective potentiality is defined by

$$H = \frac{1}{n_k} \sum_k h_k. \tag{5}$$

Individual potentiality is defined for each connection weight. By normalizing the absolute weight by its maximum value, the relative individual potentiality is given by

$$g_{jk} = \frac{u_{jk}}{\max_{j'} u_{j'k}}. \tag{6}$$

Summing over all layers yields the individual potentiality:

$$G = \frac{1}{n_j n_k} \sum_{jk} \frac{u_{jk}}{\max_{j'} u_{j'k}}. \tag{7}$$

We define the contradiction between collective and individual potentialities as

$$D = H - G. \tag{8}$$

As this contradiction increases, collective potentiality increases while individual potentiality decreases, indicating a growing contradiction between connection weights and neurons. This quantity is not guaranteed to be positive; when individual potentiality increases and collective potentiality decreases, the contradiction can become negative.

This contradiction can be further extended by introducing the cost potentiality. The objective is to increase the contradiction D while minimizing the associated cost, leading to the contradiction ratio

$$R = \frac{D}{C}. \tag{9}$$

An increase in this ratio indicates a stronger contradiction between collective and individual potentiality accompanied by a reduction in total cost.

IV. Results and Discussion

After briefly explaining the overall experimental procedures, we present a numerical summary, followed by the results on collective and individual potentiality, cost and generalization, weights for all layers, and layer potentiality.

A. Experimental Outline

For clearly and simply demonstrating the performance of the simplification method, an artificial data set was used, in which both linear and non-linear relationships were implemented. The number of input patterns was 1000, and the number of inputs was seven. Among these inputs, the first four were linearly related to the targets, whereas the remaining three were non-linearly related to the targets. The non-linear inputs were generated by applying the square, sine, and logarithmic functions to the original inputs. The number of hidden layers was ten, and all parameter values, except for the number of learning steps (epochs), were set to the default values in the PyTorch package to facilitate easy reproduction of the results.

Figure 2 shows an outline of our experiments. As shown in Figure 2(a), initially, one step of collective simplification and four steps of individual simplification were applied to both forward and backward simplification. As a result, the final network was compressed into a network without hidden layers, as shown in Figure 2(b). This compressed network was compared with a prototype network in terms of correlation coefficients, which were computed using the data sets shown in Figure 2(d) and (e). When both backward and forward methods were used, backward simplification was first applied for 1100 learning steps, followed by forward simplification. This setting was chosen to improve generalization performance.

The main findings can be summarized as follows:

- The results show that forward contradiction maximization tended to win the competition over backward contradiction maximization.

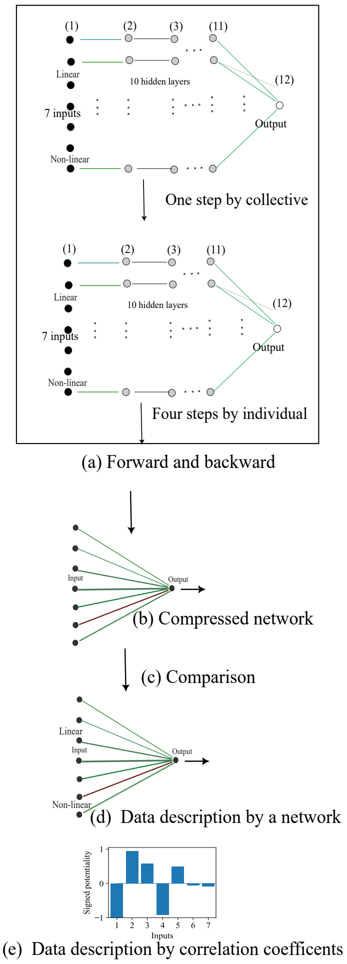

Figure 2. Forward and backward simplification (a), compressed network (b), comparison between compressed and prototype networks (c), data description by a network (d), and data description by correlation coefficients (e).

- Forward competition attempted to de-compress input information. Although input information tended to be compressed in the hidden layers, it could be de-compressed and fully transferred to all neurons in the subsequent layers.
- Backward competition focused on the output layer and target information. By losing the competition, backward simplification could be used to add an appropriate level of complexity, leading to improved generalization.
- Ultimately, the combination of forward and backward simplification could be used to simplify compressed networks by reducing the number of important inputs.

B. Numerical Summary

In the first place, we explain a summary of our experimental results, particularly from the viewpoint of generalization performance, since generalization has been one of the major indices of improved performance. However, it should be noted that this paper does not aim to explain the mechanism of improved generalization itself, but rather to clarify how different simplification procedures compete with each other during the simplification process.

The results show that forward simplification produced the most efficient simplification. In this case, the collective potentiality was the largest (0.737), the individual potentiality was the smallest (0.277), and correspondingly the largest difference, or contradiction, between collective and individual potentiality (0.460) was obtained. In addition, the ratio of this difference to the cost was the highest (2.931). This indicates that the forward type of simplification could most effectively simplify the network configuration in terms of both collective and individual simplification. When the two types of simplification forces were combined, the best generalization performance was obtained (0.861). In this combination, the difference or contradiction between collective and individual potentiality became negative (-0.078) in the backward simplification, indicating that the individual simplification force was stronger than the collective simplification force. At the same time, in this combination, the collective potentiality produced by the forward method was the third largest (0.688), the individual potentiality was the second smallest (0.279), and the resulting difference was the second largest (0.409). In addition, the ratio of this difference to the cost was the second largest (2.863). This suggests that, through the combination of forward and backward simplification, forward simplification was slightly weakened due to competition between the two simplification processes. This weakening could increase complexity and thereby improve generalization.

TABLE I. SUMMARY OF RESULTS, BASED ON MAXIMUM TESTING ACCURACY.

	Step	Coll	Indi	C-I	Cost	C-I/Cost	Accu
Conv	535	0.703	0.502	0.201	0.228	0.882	0.763
For	3316	**0.737**	**0.277**	**0.460**	0.157	**2.931**	0.839
Back	3241	0.547	0.363	0.184	0.174	1.057	0.851
FB(F)	**3632**	0.688	0.279	0.409	**0.143**	2.863	**0.861**
FB(B)		0.313	0.391	-0.078	**0.143**	-0.548	

C. Collective and Individual Potentiality

The results show that the forward potentiality tended to win the competition over the backward potentiality.

Figure 3 shows the collective (left), individual (middle), and the difference between them (right) as functions of the number of learning steps (epochs). When the conventional method shown in Figure 3(a) was used, the collective potentiality, the individual potentiality, and their difference or contradiction remained unchanged throughout the entire learning process. Naturally, the conventional method did not achieve simplification. Figure 3(b) shows the results obtained using the forward collective and individual method. The collective potentiality (left) remained higher throughout all learning steps, while the individual potentiality (middle) decreased clearly. As a result, the difference between them (right) increased and remained relatively high. Figure 3(c) shows the results obtained using the backward collective and individual method. The collective potentiality (left) was high at the beginning, but it eventually decreased. The individual potentiality (middle) decreased gradually. Consequently, the corresponding difference between them (right) increased slightly at the beginning and then tended to decrease slightly. This indicates that, with the backward method, the collective potentiality could not be increased, which caused a reduction in the difference between collective and individual potentiality.

Figure 3(d) and (e) show the results obtained when the backward method was used during the first 1100 learning steps and the forward method was applied during the remaining learning steps. The final results were then computed in terms of potentialities using the forward method (d) and the backward method (e). Figure 3(d) shows the results obtained by the forward and backward method, where the potentialities were computed using the forward method. The collective potentiality remained higher even in the later stages of learning. The individual potentiality decreased fully, in the same manner as when only the forward method was used, as shown in Figure 3(b). The contradiction initially decreased due to the application of backward simplification, and then increased rapidly at the end as a result of forward simplification. In terms of backward potentialities shown in Figure 3(e), the collective potentiality (left) increased slightly during the backward computation period, namely up to 1100 learning steps. It then decreased considerably toward the end. The individual potentiality (middle) remained almost constant throughout the entire learning process. As a result, the difference between them increased at the beginning but decreased substantially at the end. This indicates that competition for simplification favored the forward method. In particular, the backward collective potentiality decreased considerably, implying that neurons were not used equally in terms of backward potentiality.

The combined use of forward and backward methods shows that the forward method won the competition for simplification. In this case, all neurons in the subsequent layers tended to be used equally, and the connection weights to those neurons were used locally. This implies that input information tended to be compressed within a layer, but could be de-compressed in the subsequent layer.

D. Cost and Generalization

Figure 4 shows the cost (left), the ratio of the difference between collective and individual potentiality to the cost (middle), and the generalization accuracy (right). Forward simplification considerably increased the contradiction ratio, whereas backward simplification could not increase it. When both methods were combined, this tendency was maintained in terms of forward simplification, but it was weakened in terms of backward simplification. This indicates that forward simplification won the competition over backward simplification.

Figure 4(a) shows the results obtained using the conventional method without potentiality control. The cost (left) increased rapidly, while the ratio remained constant. The generalization accuracy was relatively low and remained almost constant throughout the entire learning process. Figure 4(b) shows the results obtained using forward potentiality control. The cost (left) was kept small throughout all learning steps, and the ratio of the difference to the cost increased and approached 3. In addition, the generalization accuracy (right) increased gradually toward the end of learning. Figure 4(c) shows the results obtained using the backward method. The

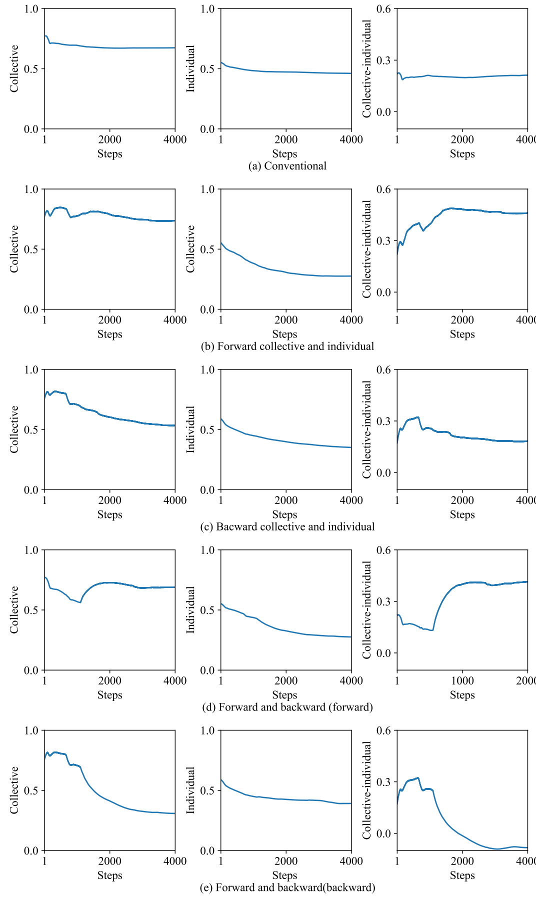

Figure 3. Collective (left), individual (middle) potentiality, and difference or contradiction between collective and individual (right) as a function of the number of learning steps (epochs) by the conventional method (a), by using the forward (b), by the backward (c) and by the forward and backward simplification with forward potentiality values (d) and backward potentiality values (e).

cost (left) was also relatively small, and the ratio (middle) initially increased but gradually decreased. The generalization accuracy (right) continued to increase until the end of learning and was slightly higher than that obtained by the forward method shown in Figure 4(b). Figure 4(d) shows the results obtained using the forward and backward method, where the potentialities were computed using the forward approach. The cost (left) was smaller than that of the conventional method, and the ratio (middle) increased rapidly and approached 3. The generalization accuracy (right) increased rapidly and reached the highest value, as summarized in Table I. Figure 4(e) shows the results obtained using the forward and backward method, where all potentialities were computed using the backward approach. One of the major findings is that the ratio of the difference to the cost (middle) decreased substantially. As explained in Figure 3(d), the backward collective potentiality was reduced. Nevertheless, the generalization performance increased gradually as the number of learning steps increased. The results demonstrate that the forward method could sufficiently increase the ratio of contradiction to the cost while improving generalization. In contrast, the backward method could not increase this ratio, even though the cost

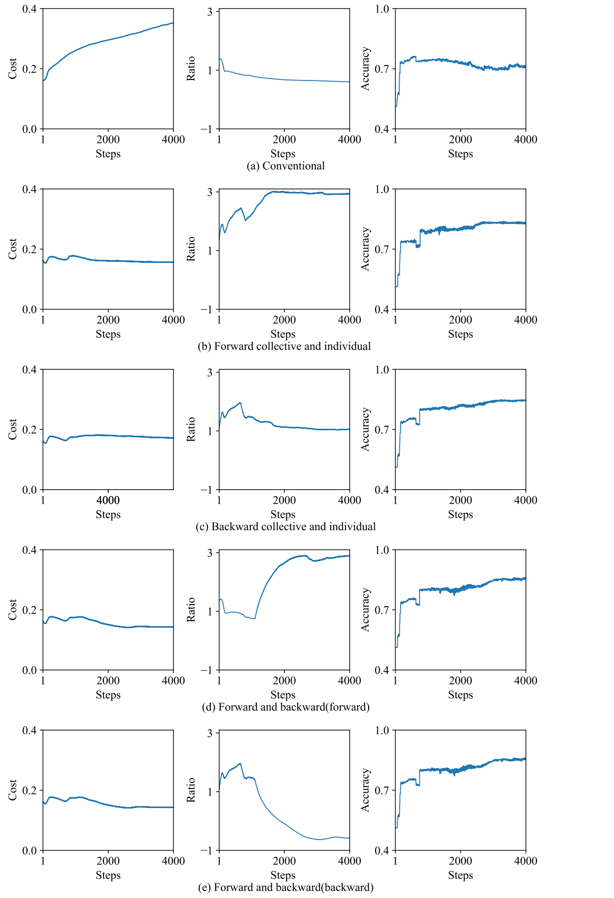

Figure 4. Cost (left), ratio of contradiction potentiality to cost (middle), and generalization accuracy (right) as a function of the number of learning steps by the conventional method (a), by using the forward collective and individual (b), by the backward collective and individual (c) and by the forward and backward with values, computed by the forward way (d) and the backward way (e).

was sufficiently small. This indicates that the forward method won the competition in simplification, whereas the backward method lost the competition, despite achieving relatively high generalization performance. This further suggests that simplification was mainly achieved by the forward method, while the backward method applied at the early stage of learning contributed to improved generalization.

E. Weights for All Layers

Figure 5 shows the connection weights after learning was completed using four different methods. Forward simplification produced more explicit patterns in the weights. When forward and backward simplification were combined, the resulting weights exhibited mixed properties, combining characteristics of both forward and backward simplification.

Figure 5(a) shows the results obtained using the conventional method. As can be seen in the figure, the weights appear to be randomly distributed, and only in the higher layers some regularities can be observed. Figure 5(b) shows the results obtained using the forward method. The weights were compressed in the majority of layers, whereas in the higher layers, compression and de-compression were mixed. Figure 5(c) shows the results obtained using the backward method. The weights became sparser than those obtained using the conventional method, but they were more randomly distributed than those obtained using the forward method in Figure 5(b). Figure 5(d) shows the results obtained using the combined forward and backward method. As expected, mixed weight patterns were obtained, in which the characteristics of forward and backward simplification were combined. Overall, the weights became slightly more randomized than those obtained using the forward method alone.

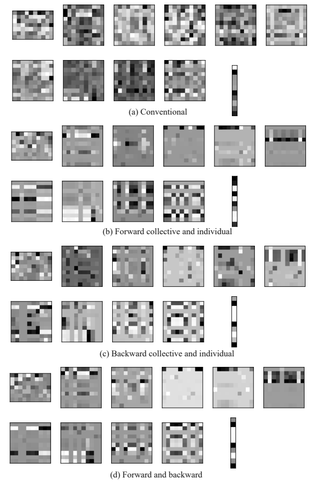

Figure 5. Connection weights across all layers obtained using the conventional method (a), the forward method only (b), the backward method only (c), and the combined forward and backward method (d).

By using both forward and backward simplification, the weights produced by forward simplification became slightly less explicit, introducing additional complexity into the connection weights. This added complexity may be one of the main factors contributing to improved generalization.

F. Layer Potentiality

Figure 6 shows the layer potentiality, computed by summing all individual potentialities within each layer. Forward simplification exhibited a clearer pattern, in which the layer potentialities decreased initially and then increased toward the end. When both forward and backward simplification were applied, this tendency was slightly attenuated.

Figure 6(a) shows the results obtained using the conventional method. The layer potentiality initially exhibited a nearly uniform distribution. Gradually, the layer potentiality of the output layer became the largest, indicating that the output layer played the most important role in learning. Figure 6(b) shows the results obtained using the forward method with collective and individual simplification. The layer potentiality was relatively large at the beginning, then became smallest in the middle layers, and finally became largest at the output layer. Figure 6(c) shows the results obtained using the backward method. The layer potentialities were almost the same as those obtained using the forward method in Figure 6(b), but their overall magnitudes were weaker. Figure 6(d) shows the results obtained when the forward and backward methods were combined. The strength of the layer potentialities reflected

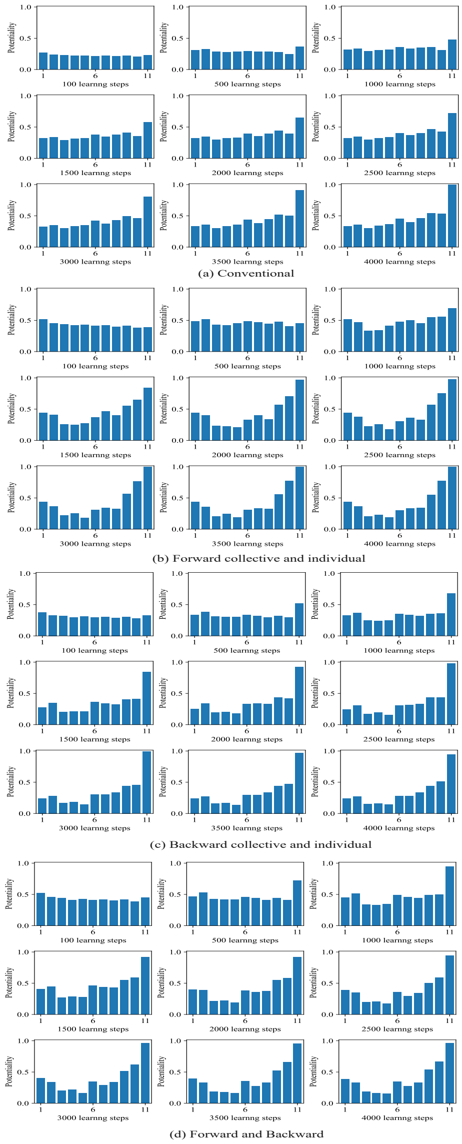

Figure 6. Layer potentialities across all layers, defined as the sum of individual potentialities within each layer, obtained using the conventional method (a), the forward collective and individual method (b), the backward collective and individual method (c), and the combined forward and backward method (d).

a mixture of the characteristics observed in the forward and backward methods.

When both forward and backward simplification were used, the layer potentialities became slightly weaker than those obtained using only the forward method. This is one of the main reasons for the improved generalization, because additional complexity was introduced through the effect of backward simplification.

V. Conclusion and Future Work

The present paper aimed to demonstrate the existence of a simplification principle in which multiple simplification procedures compete with each other. In this study, forward and backward simplification competed with one another, and both collective and individual simplification were incorporated into the learning process. The results show that forward simplification, which aims to simplify the network configuration from the viewpoint of input information, won the competition. As a result, input information was compressed and subsequently de-compressed across the layers. The backward simplification, which lost the competition, could instead be utilized to introduce an appropriate level of complexity, leading to improved generalization. Competition among components and learning procedures produces winning mechanisms through which simplification can be further deepened. Conversely, the losing mechanisms can play a complementary role by adding complexity, for example, to enhance generalization performance.

The present paper is a preliminary study on the competition between forward and backward simplification. Further investigation is required to examine the competitive effects of forward and backward simplification by more carefully controlling the associated simplification parameters. In addition, larger and more practical data sets should be employed to more accurately evaluate the performance of the proposed method.

References

[1] D. E. Rumelhart and D. Zipser, "Feature discovery by competitive learning," *Cognitive Science*, vol. 9, pp. 75–112, 1985.

[2] C. S. Choy and W. Siu, "A class of competitive learning models which avoids neuron underutilization problem," *IEEE Transactions on Neural Networks*, vol. 9, no. 6, pp. 1258–1269, 1998.

[3] A. Banerjee and J. Ghosh, "Frequency-sensitive competitive learning for scalable balanced clustering on high-dimensional hyperspheres," *Neural Networks, IEEE Transactions on*, vol. 15, no. 3, pp. 702–719, 2004.

[4] D. Krotov and J. J. Hopfield, "Unsupervised learning by competing hidden units," *Proceedings of the National Academy of Sciences*, vol. 116, no. 16, pp. 7723–7731, 2019.

[5] Y. Lu, Y.-M. Cheung, and Y. Y. Tang, "Self-adaptive multiprototype-based competitive learning approach: A k-means-type algorithm for imbalanced data clustering," *IEEE Transactions on Cybernetics*, vol. 51, no. 3, pp. 1598–1612, 2019.

[6] T. Shinozaki, "Biologically motivated learning method for deep neural networks using hierarchical competitive learning," *Neural Networks*, vol. 144, pp. 271–278, 2021.

[7] M. G. Mahdy, A. R. Abas, and T. M. Mahmoud, "Multi-phase adaptive competitive learning neural network for clustering big datasets," in *The International Conference on Artificial Intelligence and Computer Vision*, Springer, 2021, pp. 731–741.

[8] G. Lagani, F. Falchi, C. Gennaro, and G. Amato, "Training convolutional neural networks with competitive hebbian learning approaches," in *International Conference on Machine Learning, Optimization, and Data Science*, Springer, 2021, pp. 25–40.

[9] P. Li, S. Tu, and L. Xu, "Deep rival penalized competitive learning for low-resolution face recognition," *Neural Networks*, vol. 148, pp. 183–193, 2022.

[10] T. Kohonen, "The self-organizing maps," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[11] Y. Xu, L. Xu, and T. W. Chow, "Pposom: A new variant of polsom by using probabilistic assignment for multidimensional data visualization," *Neurocomputing*, vol. 74, no. 11, pp. 2018–2027, 2011.