DOI: 10.15439/2017F552
ISSN 2300-5963 ACSIS, Vol. 12

Direct Potentiality Assimilation for Improving Multi-Layered Neural Networks

Ryotaro Kamimura, IT Education Center, Tokai Univeristy
4-1-1 Kitakaname, Hiratsuka, Kanagawa 259-1292, Japan
ryo@keyaki.cc.u-tokai.ac.jp

Abstract—**The present paper aims to propose a new potential learning method to overcome the problem of collective interpretation for interpreting multi-layered neural networks. The potential learning has been introduced to detect important components of neural networks and to train them, taking into account the importance of components. Recently, it has been applied to multi-layered neural networks and then the interpretation of input neurons or variables can be possible by collectively treating intermediate layers. However, the collective interpretation for multi-layered neural networks tends to be instable, because the potentialities computed in the pre-training become different from those in the main training. To overcome this problem, we introduce the potential learning with direct potential assimilation. The direct potential assimilation means that the potentiality assimilation is not applied in the phase of pre-training but it is applied directly to training multi-layered neural networks. The new method was applied to the student evaluation data set. Then, it was observed that the selectivity of connection weights could be increased. Then, the input-output potentiality was quite similar to the regression coefficients of logistic regression analysis. Finally, the new method could extract more explicitly input-output relations than the regression coefficients by the logistic regression analysis, while improving generalization performance.**

I. Introduction

A. Problem of Collective Interpretation

NEURAL networks have been well known for their inability to interpret final results [1], [2], [3], [4], [5]. Thus, compared with conventional logistic analysis, the neural networks have not been necessarily used in many practical problems. This hard interpretation has become much more serious for multi-layered neural networks. Some results on interpretation were reported [6], [7], but the majority were heavily based on the characteristics of input patterns. For example, when the inputs are images, they can be easily interpreted intuitively by the conventional visualization methods. Particularly, in multi-layered neural networks, it has been difficult to interpret the intermediate layers.

For interpretation, we have so far introduced potential learning [8], [9], [10] where the importance of neural components is determined before learning, and they are assimilated in connection weights. Potential learning has been developed to simplify the computational procedures of information-theoretic methods [11], [12], [13], [14]. The potential learning has been recently extended to multi-layered neural networks. As above mentioned, for the multi-layered neural networks, the problem becomes more serious, particularly, for interpretation. Even in the case of single-layered neural networks, the interpretation is not so easy that we need very special types of procedures for interpretation. In multi-layered neural networks, the complicated behaviors of many intermediate layers cannot be easily interpreted. To simplify the interpretation of multi-layered neural network, we focus on relations between inputs and outputs by treating collectively all intermediate layers. This is because in many applications, we must examine how input variables (neurons) are related to the corresponding outputs [15], [16], [17]. Thus, we try to estimate how input neurons have influences on outputs by considering all intermediate layers.

However, the problem of this collective interpretation is that the interpretation has been unstable because of unstable potentialities. The instability of final results is due to the fact that the connection weights, obtained in the pre-training, can be changed in the fine-turning or main-training. Thus, even if the potentiality of neural components is rigorously computed, it can be of no use in main-training. For this problem, we have introduced direct potentiality assimilation where the potential learning focuses not on pre-training but on main-training. In our new method, the roles of pre-training are reduced as much as possible.

B. Direct Potentiality Assimilation

The instability problem, inherent to the potentiality learning or assimilation, can be solved by transferring the potentiality assimilation from the pre-training to the main-training. Since the method directly apply the potentiality to the main-training, it is called “direct potentiality assimilation”. In the ordinary deep learning, the un-supervised or semi-unsupervised learning such as auto-encoders is used for the pre-training. In the pre-training using the auto-encoders, the potentiality must be assimilated by repeating the assimilation processes, because the effect of potentiality tends to disappear. Then, we have the fine-tuning or main-training with connection weights by the pre-training. The problem is that the information on input patterns tends to disappear in the time of pre-training, because of the repeated assimilation. This means that the original information on inputs tends to disappear in the pre-training, and thus connection weights, transfered to the main-training, happen to have little information on input patterns, leading to the instability of learning and interpretation. To overcome this problem, we transfer the process of assimilation to the main training. Then, in the pre-training, no regularization can be implemented and we try to obtain the overall or rough information on input patterns. In the main-training, the important connection weights in terms of potentiality is extracted and assimilated fully.

C. Paper Organization

In Section 2, we first explain conceptually direct and indirect potential learning and then how to compute the potentiality and how to assimilate the potentiality into connection weights. For the collective interpretation, we present how to deal collectively with intermediate layers by considering only positive weights. In Section 3, the student evaluation data set was used where we try to show that the selectivity could be improved, with better generalization performance. In addition, the collective weights were found to be very similar to those by the logistic regression analysis. Finally, the method could successfully extract the clearer roles of input neurons or variables.

II. Theory and Computational Methods

A. Direct and Indirect Potential Learning

In the previous models, we applied the potential learning to multi-layered neural networks indirectly. This means that the potential learning was applied to the pre-training phase. The problem of this indirect method is that the information on input patterns tends to disappear in the phase of pre-training. The multi-layered neural networks themselves tend to lose the original information when going through many different layers, as pointed out and well-known in the field of information theory [18], [19], [20]. In addition, the weight decay and sparse constraints [21], [22], [23], [24], usually used in the pre-training, naturally tend to lose the original information, because those methods try to simplify the complexity of networks by decreasing the supposed redundant information. The present method tries to keep the original information by reducing the roles of pre-training as much as possible. All important precedences of potentiality assimilation are implemented in the main-training. As several reports stated, deep neural networks could produce better results without pre-training [25]. Our method to focus on the main learning is quite well suited for this situation.

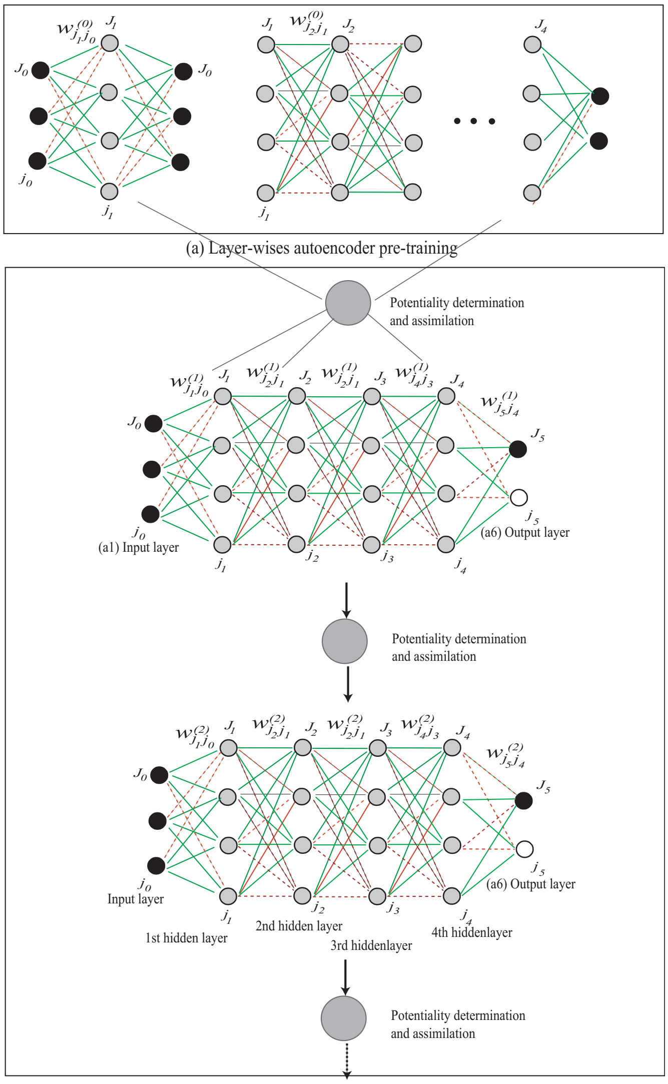
B. Direct Potentiality Assimilation

In Figure 1, a neural network architecture with four hidden layers is shown in which the connection weights from the input to the first hidden layer for the pre-training are represented by $w^{(0)}_{j_1 j_0}$ with J_1 and J_0 neurons in the pre-training. Then, the positive weights are computed by

$$u^{(0)}_{j_1 j_0} = \max\left(w^{(0)}_{j_1 j_0}, 0\right). \tag{1}$$

By normalizing these weights, we have the potentiality

$${}^r\phi^{(0)}_{j_1 j_0} = \left(\frac{u^{(0)}_{j_1 j_0}}{u^{(0)}_{\max}}\right)^r, \tag{2}$$



(a) Layer-wises autoencoder pre-training

(b) Main-training

Fig. 1. Network architecture with four hidden layers by the direct potentiality assimilation.

where $u^{(0)}_{\max}$ denotes the maximum positive weight and r denotes the potential parameter. Using this potentiality, the selective potentiality can be defined by

$$\phi^r_{10} = \frac{J_1 J_0 - \sum_{j_1=1}^{J_1} \sum_{j_0=1}^{J_0} {}^r\phi^{(0)}_{j_1 j_0}}{J_1 J_0 - 1}. \tag{3}$$

When all connection weighs become zero, the selective potentiality is also zero. This selective potentiality increases when the number of strong connection weights decreases. In the end, potentiality reaches its maximum of one when only one weight is the strongest, while all the others are forced to be zero.

This potentiality is assimilated in the main training as

$$w^{(1)}_{j_1 j_0} = {}^r\phi_{j_1 j_0} w^{(0)}_{j_1 j_0}. \tag{4}$$

In the same way, for the second step, we have

$$w^{(2)}_{j_1 j_0} = {}^r\phi^{(1)}_{j_1 j_0} w^{(1)}_{j_1 j_0}, \tag{5}$$

where ${}^{r}\phi^{(1)}_{j_1 j_0}$ denotes the potentiality at the first step of learning.

The Average potentiality is the average of all potentialities, in this case, five different potentialities for five layers,

$$\phi^{r}_{avg} = \frac{1}{5}\sum_{k=1}^{5}\phi^{r}_{j_k,j_{k-1}}. \qquad (6)$$

C. Collective Interpretation

We focus on the interpretation of input neurons or variables. Since it is impossible to interpret all the connection weights of all intermediate layers, we try to treat them collectively. Thus, the potentiality of the input-output connection weights is computed by summing all weights in the intermediate layers. The collective weights from the input to the output layer are computed by

$$u_{j_5 j_0} = \sum_{j_4=1}^{J_4}\sum_{j_3=1}^{J_3}\sum_{j_2=1}^{J_2}\sum_{j_1=1}^{J_1} w_{j_5 j_4} w_{j_4 j_3} w_{j_3 j_2} w_{j_2 j_1} w_{j_1 j_0}. \qquad (7)$$

We use here raw connection weights to see detailed characteristics. However, since connection weights are forced to be positive, the final collective weights are not so different from those by the positive weights.

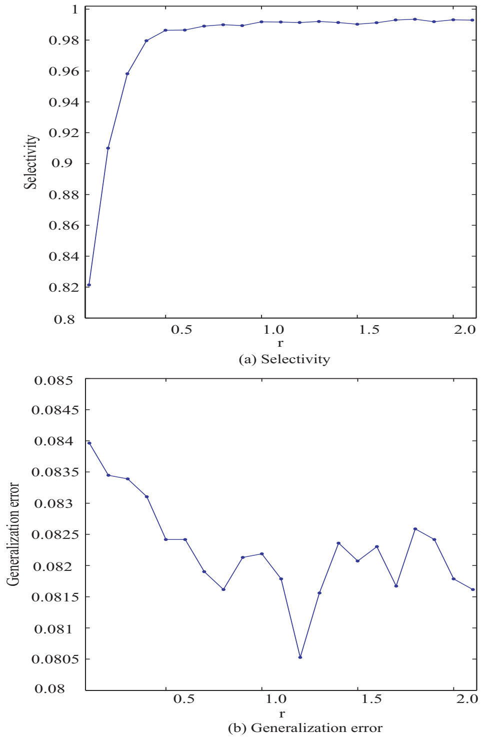
III. Results and Discussion

A. Student Evaluation Data Set

1) Experimental Outline: The data set was composed of 5,820 class evaluation scores by the students from the machine learning database [26]. Of total 33 variables, 28 variables were extracted on the evaluation questions. Then, the variable No.9, related to the class satisfaction[1] was used for the targets representing the class satisfaction. The 70 percent of the data set was for training and the remaining one for evaluation. We used the Matlab neural network package with all default parameter values, because we focused on the easy reproduction of the present results.

2) Selectivity and Generalization: Figure 2 shows the average selectivity (a) and generalization errors (b). As can be seen in the figures, the selectivity increased gradually in Figure 2(a), and correspondingly, the generalization errors decreased to the minimum point when the parameter r was increased from 0 to 1.1. Then, the generalization errors did not decrease but fluctuated. These results show that the selectivity can be used to increase generalization performance by choosing appropriately the parameter r.

3) Comparison of Connection Weights: Figure 3 shows connection weights when the parameter r was zero. Connection weights were almost random and it was impossible to detect any regularity over connection weights. Figure 4 shows connection weights when the parameter r was 1.1, producing the best generalization performance. Though some minor negative connection weights were seen in the weights to the first hidden layer in Figure 4(a). The majority of weights were positive and they remained to be strong for all layers.



(a) Selectivity

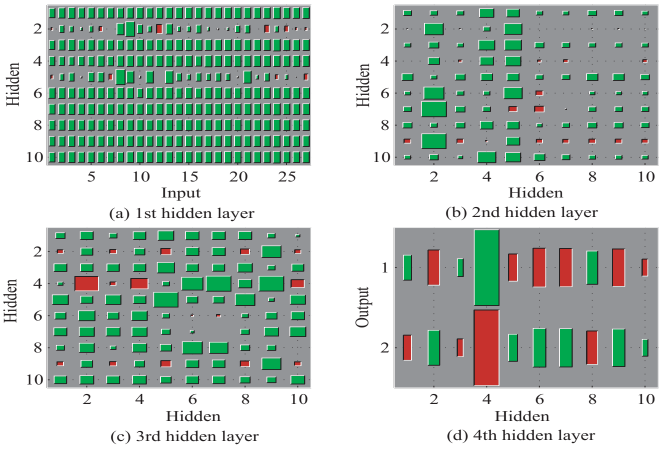
(b) Generalization error

Fig. 2. Selectivity (a) and generalization errors (b) for the student evaluation data set.

Let us examine why connection weights with r=1.1 in Figure 4 produced better generalization performance. In Figure 4, the vertical lines and horizontal lines were added. The horizontal lines represent that connection weights are connected with the subsequent connection weights. On the other hand, the vertical lines show that the corresponding weights are connected with ones located in the former layer. Connection weights to the fifth hidden neuron are strong in Figure 4(a) and they are connected with the third hidden neurons in the second hidden layer in Figure 4(b). Then, these neurons were connected with the fourth hidden neurons in Figure 4(c). Finally, the connection weights are connected with connection weights into the first output neuron in Figure 4(d). Thus, those connection weights make it possible to transmit information on original input patterns to the output layer.

4) Interpreting Input Selective Potentiality: Figure 5(a) shows the collective weights when the parameter was 1.1, giving the best generalization performance. As can be seen in the figures, the ninth input neuron took the highest weight value. When the class expectation is met by students, they tend to be satisfied with the class. On the other hand, Figure 5(b) show the regression coefficients by the logistic regression analysis. We can see the same tendency that the ninth variable had the largest value. However, some other variables had relatively larger values, for example, the variable No.16. These result show that the direct potentiality assimilation can extract clearer characteristics than logistic regression analysis. This is

[1] Actually, the variable No.9 means that the students enjoyed the class very much.



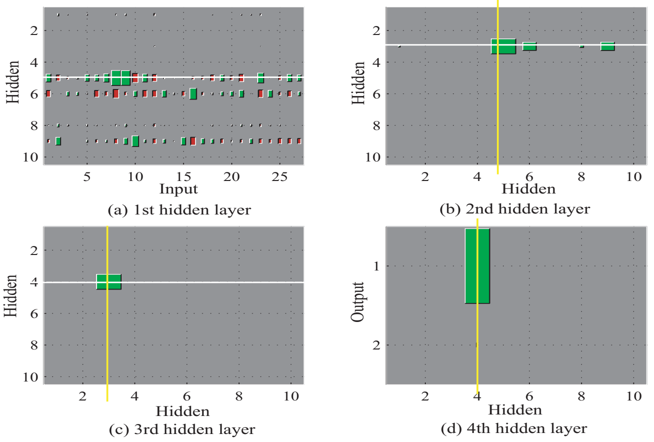
(a) 1st hidden layer

(b) 2nd hidden layer

(c) 3rd hidden layer

(d) 4th hidden layer

Fig. 3. Connection weights from the first (a) to output (d) layer when the parameter r was 0 for the student evaluation data set.



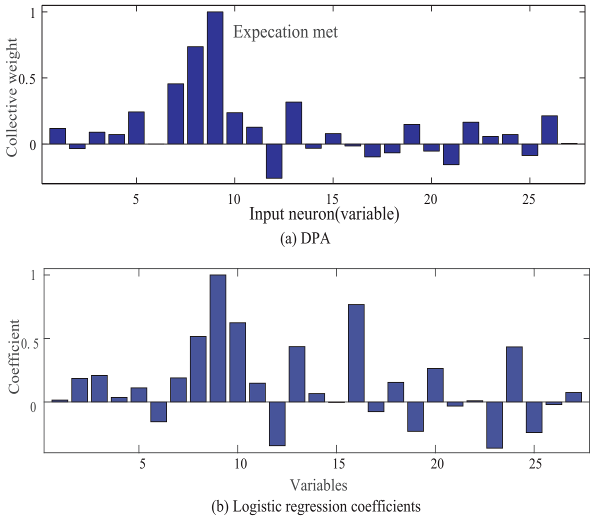
(a) 1st hidden layer

(b) 2nd hidden layer

(c) 3rd hidden layer

(d) 4th hidden layer

Fig. 4. Connection weights from the first (a) to output (d) layer with r=1.1 for the student evaluation data set.



(a) DPA

(b) Logistic regression coefficients

Fig. 5. Collective weights (a) and regression coefficients (b) by the logistic regression analysis.

due to the selective effect of potentiality assimilation.

5) Generalization Comparison: Table I shows the generalization performance by the logistic analysis, deep learning and direct potentiality assimilation method. The lowest errors were 0.0805, 0.0710 and 0.0899 in terms of average, minimum and maximum values when the parameter r was 1.1. The second best average error of 0.0868 was obtained by the logistic regression analysis. Then, the worst error of 0.0940 was by the deep learning with three hidden layers. This means that it became impossible to learn input patterns by the ordinary deep learning with auto-encoders. The unsupervised learning such as auto-encoders tends to lose information content gradually when the layer becomes higher.

IV. Conclusion

In the present paper, we proposed a new potential leaning method in which the potentiality assimilation is transferred from the pre-training to the main-training. The potential learning has been originally developed to simplify complicated information-theoretic methods [11], [14]. Because of the complexity in computing entropy or mutual information, the methods have not been fully explored in the neural networks. In this context, the potential learning has been introduced to simplify the computational procedures of information maximization [8], [27], [9], [10]. First, the potentiality of some components is determined and then this potentiality is assimilated. Usually, a smaller number of components with higher potentiality is extracted. The potential learning has been applied to single-layered neural networks as well as multi-layered neural networks. To train the deep neural networks, the pre-training has been believed to have much importance. In multi-layered neural networks, the un-supervised pre-training is usually used to solve the vanishing information problem, inherent to the gradient descent. In addition, several regularization terms such as weight decay and sparsity constraints are implemented. These methods such as un-supervised pre-training with the regularization terms tend naturally to reduce information content on original input patterns. Actually, it is difficult to control information in the pre-training for the benefit of the subsequent main training. To solve this problem, though the pre-training is necessary in training multi-layered neural networks, the roles of the pre-training should be minimized. We think that the main role of pre-training is to give the overall or rough information content to be used in the main-training.

The method was applied to the student evaluation data set. Then, it could be observed that generalization performance could be improved. The final collective weights were very similar to those by the regression coefficients by the logistic regression analysis. This means that the present method extracted the same characteristics by the logistic regression analysis, taking into account some additional features which the conventional logistic analysis could not deal with.

The problem is that the potentiality was applied independently in all layers. This means that when the parameter was increased, and the effect of potentiality is more apparent, the potentiality tends to be assimilated independently in each layer. Finally, the layers tend to be dis-connected with each other.

TABLE I
SUMMARY OF EXPERIMENTAL RESULTS ON GENERALIZATION PERFORMANCE FOR THE STUDENT DATA SET.

Method	Layers	r	Avg	Std	Min	Max
Logistic			0.0868	0.0065	0.0733	0.0956
Deep	3		0.0940	0.0088	0.0762	0.1031
DPA		1.1	**0.0805**	0.0065	**0.0710**	**0.0899**

Then, it can be considered that the original information content in input patterns cannot be transmitted through layer. Thus, the information on input patterns tends to be lost gradually in the course of learning. To solve this problem, the present method should be formulated, taking into account the connectivity between neurons and layers. Though some problems should be solved for the practical data sets, the method is simple enough to be implemented in large-scale networks.

REFERENCES

[1] R. Andrews, J. Diederich, and A. B. Tickle, “Survey and critique of techniques for extracting rules from trained artificial neural networks,” *Knowledge-based systems*, vol. 8, no. 6, pp. 373–389, 1995.

[2] J. M. Benítez, J. L. Castro, and I. Requena, “Are artificial neural networks black boxes?,” *IEEE Transactions on neural networks*, vol. 8, no. 5, pp. 1156–1164, 1997.

[3] M. Ishikawa, “Rule extraction by successive regularization,” *Neural Networks*, vol. 13, no. 10, pp. 1171–1183, 2000.

[4] T. Q. Huynh and J. A. Reggia, “Guiding hidden layer representations for improved rule extraction from neural networks,” *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 264–275, 2011.

[5] B. Mak and T. Munakata, “Rule extraction from expert heuristics: a comparative study of rough sets with neural network and ID3,” *European journal of operational research*, vol. 136, pp. 212–229, 2002.

[6] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, “Understanding neural networks through deep visualization,” *arXiv preprint arXiv:1506.06579*, 2015.

[7] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, “Visualizing higher-layer features of a deep network,” *University of Montreal*, vol. 1341, 2009.

[8] R. Kamimura, “Simple and stable internal representation by potential mutual information maximization,” in *International Conference on Engineering Applications of Neural Networks*, pp. 309–316, Springer, 2016.

[9] R. Kamimura, “Collective interpretation and potential joint information maximization,” in *Intelligent Information Processing VIII: 9th IFIP TC 12 International Conference, IIP 2016, Melbourne, VIC, Australia, November 18-21, 2016, Proceedings*, pp. 12–21, Springer, 2016.

[10] R. Kamimura, “Repeated potentiality assimilation: Simplifying learning procedures by positive, independent and indirect operation for improving generalization and interpretation (in press),” in *Proc. of IJCNN-2016*, (Vancouver), 2016.

[11] R. Linsker, “Self-organization in a perceptual network,” *Computer*, vol. 21, no. 3, pp. 105–117, 1988.

[12] R. Linsker, “How to generate ordered maps by maximizing the mutual information between input and output signals,” *Neural computation*, vol. 1, no. 3, pp. 402–411, 1989.

[13] R. Linsker, “Local synaptic learning rules suffice to maximize mutual information in a linear network,” *Neural Computation*, vol. 4, no. 5, pp. 691–702, 1992.

[14] R. Linsker, “Improved local learning rule for information maximization and related applications,” *Neural networks*, vol. 18, no. 3, pp. 261–265, 2005.

[15] G. Castellano and A. M. Fanelli, “Variable selection using neural-network models,” *Neurocomputing*, vol. 31, pp. 1–13, 1999.

[16] G. G. Oliveira, O. C. Pedrollo, and N. M. Castro, “Simplifying artificial neural network models of river basin behaviour by an automated procedure for input variable selection,” *Engineering Applications of Artificial Intelligence*, vol. 40, pp. 47–61, 2015.

[17] J. D. Olden, M. K. Joy, and R. G. Death, “An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data,” *Ecological Modelling*, vol. 178, no. 3, pp. 389–397, 2004.

[18] C. E. Shannon, “A mathematical theory of communication,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3–55, 2001.

[19] C. E. Shannon, “Prediction and entropy of printed english,” *Bell system technical journal*, vol. 30, no. 1, pp. 50–64, 1951.

[20] N. Abramson, “Information theory and coding,” 1963.

[21] G. Hinton, “A practical guide to training restricted boltzmann machines,” *Momentum*, vol. 9, no. 1, p. 926, 2010.

[22] J. Kim, V. D. Calhoun, E. Shim, and J.-H. Lee, “Deep neural network with weight sparsity control and pre-training extracts hierarchical features and enhances classification performance: Evidence from whole-brain resting-state functional connectivity patterns of schizophrenia,” *NeuroImage*, vol. 124, pp. 127–146, 2016.

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[24] T. Xiao, H. Li, W. Ouyang, and X. Wang, “Learning deep feature representations with domain guided dropout for person re-identification,” *arXiv preprint arXiv:1604.07528*, 2016.

[25] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Networks*, vol. 61, pp. 85–117, 2015.

[26] K. Bache and M. Lichman, “UCI machine learning repository,” 2013.

[27] R. Kamimura, “Self-organizing selective potentiality learning to detect important input neurons,” in *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*, pp. 1619–1626, IEEE, 2015.

[28] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, “Deep, big, simple neural nets for handwritten digit recognition,” *Neural computation*, vol. 22, no. 12, pp. 3207–3220, 2010.