

Resolving Hidden Representations

Cheng-Yuan Liou and Wei-Chen Cheng
National Taiwan University, Taipei, Taiwan(R.O.C.)
E-mail: cyliou@csie.ntu.edu.tw

Key Idea: Traditional back-propagation training minimizes the difference between the output and the desired output. This paper presents a novel technique to separate the pattern representation in each hidden layer to facilitate many classification tasks. This technique requires that all patterns in the same class will have near representations and the patterns in different classes will have distant representations. This requirement is applied to any two data patterns to train a selected hidden layer of the multilayer perceptrons(MLP) or the recurrent neural networks(RNN). The MLP can be trained layer by layer feedforwardly to accomplish resolved representations. In RNN[3] case, the output of the hidden layer is copied to a context layer which represents the state in sequence processing. We can apply the same technique to control states' representation.

Methods: We combine the idea of SIR [1] and back-propagation [2] and define an energy function,

$$E = \gamma E^{BP} + (1 - \gamma) E^{SIR(m)}, \quad (1)$$

where m denotes a specific hidden layer. The weights are updated by propagating the error backward layer by layer and $E^{SIR(m)}$ only affect the output of the specific hidden layer.

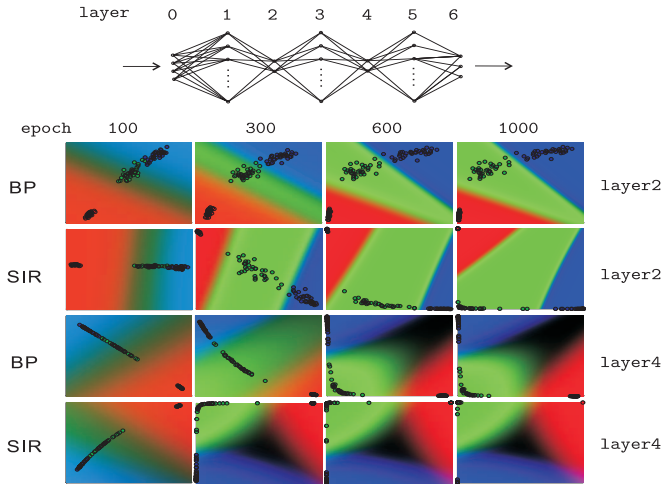


Fig. 1: The representation of hidden layer. BP means back-propagation. SIR means the model trained by the proposed energy function.

Major Results: The main result of this idea is shown in figure 1. We use fisheriris data set which has 150 data entries of four dimensions. There are three classes in the dataset. Figure 1 shows the representation of the hidden layer during the training. There are 2 hidden neurons at layer 2 and 4. Circles in figure 1 represent the output position of data in the hidden layer. The background color represents the corresponding output of the network, that is

to say, how the space is divided. The hidden neuron at layer 4 uses SIR for 1000 epochs, the patterns has output close to ± 1 at this layer.

For RNN, Using trained RNN to recognize finite state machine has been proposed. We use the proposed method to train a RNN learning the grammar in [4] and we retrieve the automata shown in figure 2.

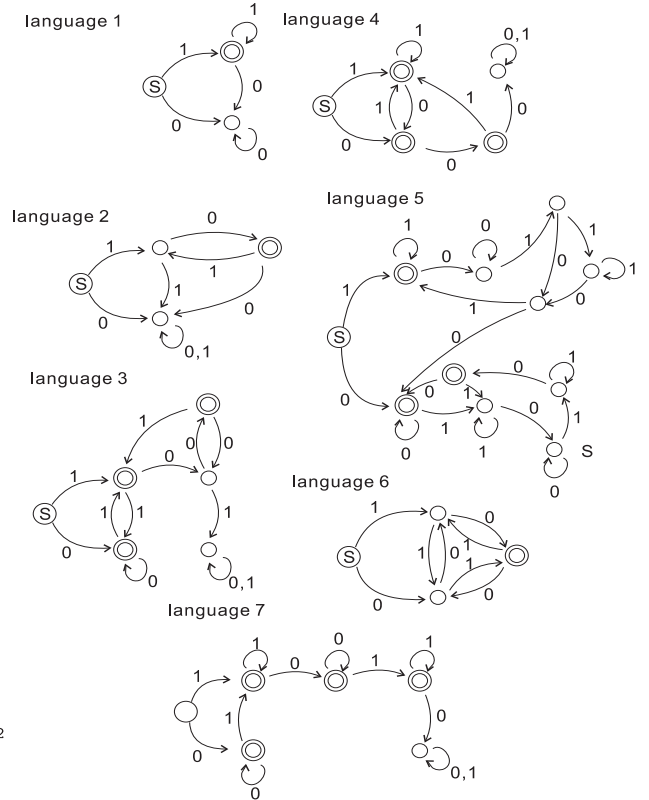


Fig. 2: FSMs learned by RNN with SIR.

References

- [1] Liou, C.Y., Chen, H.T., and Huang, J.C.: Separation of Internal Representations of the Hidden Layer. Proceedings of the International Computer Symposium, ICS, Workshop on Artificial Intelligence, Chiayi, Taiwan (2000) 26-34
- [2] Rumelhart, D.E., Hinton, G.E., and Williams, R.J.: Learning Internal Representations by Error Propagation. In D.E. Rumelhart and J.L. McClelland (Eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Cambridge, MA: MIT Press. 1 (1986) 318-362
- [3] Elman, J.L.: Finding Structure in Time. Cognitive Science 14 (1990) 179-211
- [4] Tomita, M.: Dynamics Construction of Finite-State Automata from Examples Using Hill-Climing. in Proceedings of the Fourth Annual Conference of the Cognitive Science Society (1982) 105-108.