CHAPTER V

CONCLUSION AND DISCUSSION

This work presents the Data-throwaway Learning for Streaming Chunk DLSC data classification based on Versatile Elliptic Basis Function Neural Network (VEBFNN). In this study, each chunk is feed to the network as class-wise manner. One important aspect of the proposed learning algorithm is based on one-pass-thrown-away concept. Once the data in any class are learned, they are thrown away and never learned again. By learning each class at a time, the time and space complexities can be easily managed. The proposed method is also capable of learning new information while retaining old knowledge which is opposite to *stability-plasticity dilemma*. In this research, complete training data and streaming training data chunk scenarios are investigated. For the first case, the performance of DLSC method was compared with both of batch and incremental learning algorithms, namely, MLP, standard RBF SVM for batch learning and VEBF by one datum, ILVQ and ASC for incremental learning. For the second case, the performance of DLSC method was compared with four incremental learning algorithms. The performance of each method is measured in terms of classification accuracy (%), number of hidden neurons or prototypes, and the learning time.

From the experimental results for complete training data, the results were better than those of incremental learning and RBF methods for all multi-class data set. But for 2-class data sets, the results of DLSC are better than SVM method for four 2-class data sets. There are only three data sets for which the accuracy of DLSC is slightly less than that of MLP method. In addition, the number of hidden neurons used by DLSC method is less than batch and incremental learning algorithms for most data sets. The number of hidden neurons of CIL method is more than that of VEBF for Protein Interaction because of the compromise between the accuracy and the number of hidden neurons. For a large size of MiniBooNE, the number of hidden neurons is much less than those of the other three incremental methods, i.e. VEBF, ILVQ, and ASC. The speed of computational time of CIL is the fastest in almost all data sets except for the data sets with a large number of attributes. Similar to other approaches, the performance of the proposed method depends on the initial width parameter. When concerning the input sequence, the sequence of class labels fed to the network does not degrade the performance of the method because the learning process focuses one class at a time and the distribution of data within the same class does not interfere with the distribution of the other classes. Furthermore, the accuracy and the number of hidden neurons are slightly affected by the center vector selection. The accuracy and the number of hidden neurons are quite sensitive when setting the δ to a small value.

From the experimental results for streaming training data chunk, they show that the results of DLSC are better than those of incremental learning methods for most data sets. There are only two data sets for which the accuracy of DLSC is slightly less than that of RIL method. In addition, the number of hidden neurons of DLSC method is less than those of VEBF, ILVQ and CILDA methods for all data sets. For RIL, the number of hidden neurons is determined by the number of class labels. The number of hidden neurons of DLSC method is more than that of VEBF for Forest Cover Type because of the compromise between the accuracy and the number of hidden neurons. For the learning time, the taken learning time of CILDA is fastest for all data set but it took so long time in assigning a class label for a new sample. The learning time of DLSC is the second learning time for nine data sets as shown in underline except for Liver and Forest Cover Type. The learning time of DLSC is slower slightly than the time of RIL method. For Forest Cover Type, since the initial width vector of DLSC is quite small, the taken time of DLSC is quite long. However, it is the tradeoff between the learning time and the accuracy for Forest Cover Type. The proposed method is suitable for coping with big data problem and handling streaming data as well.