BIGRAM BASED WORD SENSE DISAMBIGUATION USING NEURAL NETWORK

Tamilselvi P[1], Srivatsa S.K[2]

[1]Research Scholar, Sathyabama Universtiy, Chennai, TN, India
[2]St. Joseph College of Engineering, Chennai, TN, India

Abstract

This paper presents a method to solve word sense ambiguity using neural network. Most of the previous word sense disambiguation approaches were based on neural networks, having limitations due to their huge feature set size. Here, bigram method is adopted in two ways: post-bigram (l_1w) and pre-bigram (wr_1). Two bigram features are treated as input for the networks, each defined with one hidden layer with hidden neurons ranging from two to twenty. The input model is extracted from the sentences in Brown corpus. In this, the performance of the networks are compared using mean squared error values. Among all networks, trainable cascade forward back propagation network gives 71.3% of accuracy with pre-bigram.

Keywords: Neural Network, Word sense disambiguation, post bigram, pre bigram.

I. INTRODUCTION

One major problem of Natural language processing (NLP) is figuring out what a word means when it is used in a particular context. The different meanings of a word are listed as its various senses in a dictionary. The task of word sense disambiguation is to identify the correct sense of a word in context. Improvement in the accuracy of identifying the correct word sense will result in better machine translation systems, information retrieval systems, etc. Many research on word sense disambiguation has made it known that several information can contribute to solve the lexical ambiguity. These includes surrounding words (an unordered set of words around the target word), local collocations (a short sequence of words near a target word, taking word order into account), syntactic relations, part-of-speech (POS), morphological forms, etc (Ng and Zelle, 1997).

With rising of corpus linguistics, the machine learning methods based on statistics are booming (Yarowsky, 1992). Disambiguation approach can be classified into supervised and unsupervised based on the sentences in the Corpus which are sense labeled or not. Supervised learning methods have good learning ability and can get better accuracy in word sense disambiguation experiments (Schutze, 1998). In general, data sparseness is a common problem in supervised learning. This can be overcome by data smoothing which is a time consuming task. Unsupervised word sense disambiguation never depends on tagged corpus and could realize the training of large real corpus coming from all kinds of field. This method may overcome data sparseness problem to some extent. It is clear that the two kinds of disambiguation methods have their own advantages and disadvantages, and can't supersede each other.

(Pedersen, 2001) experimented the use of bigrams for WSD with a decision tree and naive Bayes classifier. He tested different bigrams that occur close to the ambiguous words (within approximately 50 words to the left or right of the ambiguous word) as possible disambiguation features. He applied statistical method to disambiguate texts using decision tree with bigram concept.

Some researchers use neural networks in their word sense disambiguation systems since it is strong in classification. Concept co-occurrence information was adopted as input features to disambiguate only nouns using multilayered feed forward neural network. (You-Jin chung, Sin-Jae Kang, Kyong-Hi Moon and Jong-Hyeok Lee, 2002).

(Zhimao Lu, Ting Liu and Sheng Li, 2004) extracted mutual Information (MI) of the words as input vectors for back-propagation neural network. The network is tested with maximum feature sets varying from ten words from left and ten from right with respect to ambiguous word. When the number of features increases, the sparseness is unavoidable. Smoothing is really required to overcome the above problem and to improve the performance.

We propose a bigram based sense disambiguation method using different neural networks. Only two features, the ambiguous word and immediate left word (post bigram) or immediate right word (pre-bigram) are taken for disambiguation process. We proceed this paper in such a way to describe the pre disambiguation process in the next section, disambiguation process in section 3, results discussion in section 4, way of improving the accuracy in section 5 and finally, conclusion in section 6.

II. PRE DISAMBIGUATION PROCESS

A. Tokenization

Tokenization is the process of breaking up the sequence of characters in a text by locating the word boundaries, the point where one word ends and another begins (Palmer 2000). This was not seen to be a serious problem for researchers working on English and similar languages, where word boundaries generally coincide with space characters.

B. Compound Words Separation

It is the process of breaking up the compound words defined with '–' into individual words. It is always better to have individual words for disambiguation process. Individual words always dominate in contributing their features for disambiguation and also rectify the data sparseness to some extent.

C. Morphological Formation

Different forms of a word with suffix attached tags such as 's', 'ed', 'ing', 'ful' etc, are brought into their original form by removing the tags. If this process is not done, feature vectors are defined individually for all different form of a word, which leads to memory wastage that will ultimately slow down the process.

D. PoS-Tagging

In PoS-tagging, each word must be assigned its correct Part-of-Speech, such as noun, verb, adjective or adverb. The number of tags used by different systems varies a lot. Some systems use fewer than 20 tags, while others use over 400. Many systems for Part-of-Speech tagging have learnt statistical models from a training corpus. An early example was CLAWS, the Constituent Likelihood Automatic Word-tagging System (Leech at al, Atwell 1983), which learnt a "Constituent Likelihood" model from the tagged Brown Corpus. In this research, PoS is assigned using Hidden Markov Model (P. Tamilselvi, S.K.Srivatsa, 2010). Probability of next word is extracted from word transition probability matrix and its PoS is extracted from emission matrix, and finally, entire terms (words) in the sequence (sentence) is assigned with its PoS tag. More than 97% accuracy is reflected on short sentences and above 85% accuracy for long sentences, having more than 20 words.

III. DISAMBIGUATION PROCESS

The disambiguation task is achieved through four steps: collecting ambiguous words from preprocessed input sentence, feature extraction for ambiguous words, selecting data from lexical library and training all neural networks, and finally, extracting the sense from neural network. Disambiguation process is shown in Fig-1. Generally, a sentence may have more than one ambiguous word. WordNet (Fellbaum, 1998) is used here to separate the ambiguous words from the input sentence.

A. Lexical Library Construction

Sources of sentences are taken from Brown Corpus. Brown corpus sentences are refined in two forms: [word PoS] or [**word PoS sense sense-tag**]. Words are separated by 'l' in the refined sentences. PoS of DT, CC, CD, PRP, PRP$ etc are stored as first form and PoS of NN, VB, RB, JJ etc as second form (P. Tamilselvi, S.K. Srivatsa, 2010). An example of refined form of a sentence is given in Fig-2. Collections of refined Brown Corpus sentences are referred as lexical library, having more than 5000 sentences. To vectorise the Parts of speech features in Brown Corpus, they are categorized into 17 groups having constant values from .1 to .17. Ambiguous words with sense value, sense-tag and their bigram PoS information are stored as decentralized lexical forms (P. Tamilselvi, S.K. Srivatsa, 2009) in 26 separate files ('a' to 'z'), having 74530 ambiguity words with a maximum of 8726 words in 's' and a minimum of 56 words in 'x'.

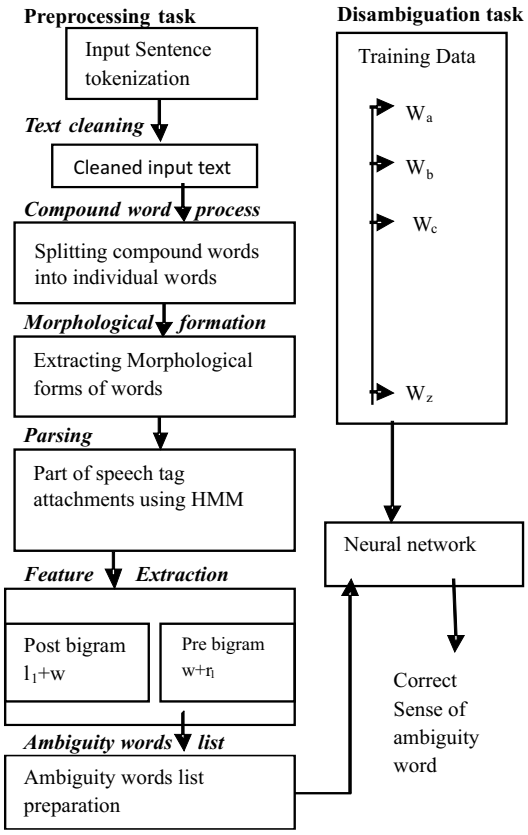

Fig. 1. Disambiguation Process

It PRP | recommend VB 1 2:32:01:: | that IN | Fulton NNP 1 1:03:00:: | legislator NN 1 1:18:00:: | act VB 1 2:41:00:: | to TO | have VB 5 2:30:00:: | these DT | law NN 2 1:10:00:: | study VB 1 2:31:02:: | and CC | revise VB 1 2:32:00:: | to TO | the DT | end NN 4 1:09:02:: | of IN | modernize VB 1 2:30:00:: | and CC | improve VB 2 2:30:00:: | them PRP |

Fig. 2. Example of Refined Brown Corpus Sentence

Decentralized training data for the neural network is represented as in Fig-3. Feature vectors are as matrix having C4 & C5 columns for post bigram and C5 & C6 for pre bigram. Target output is C3 with C2 columns.

C1	C2	C3	C4	C5	C6
Ambiguity Word	Sense	Sense-tag	Left word POS constant value	Ambiguity word POS constant value	Right word POS Constant value

Fig. 3. Format of Decentralized Text

B. Different Neural Networks

Five different neural networks, namely, feed-forward back propagation network (M1), Elman back propagation network (M2), trainable cascade forward back propagation network (M3), pattern recognition network (M4) and feed forward back propagation network with feedback from output to input (M5) are taken for process.

Almost all networks are designed to have a single hidden layer having neurons ranging from two to twenty. Tangent Sigmoid transfer function is applied in hidden layer and liner transfer function is used in output layer. Levenberg-Marquardt back propagation function is used for training. Gradient decent with momentum weight and bias learning function is used for learning. To measure the performance, mean squared error function (***mse)*** is used. The networks are adopted and trained by changing the weights repeatedly for producing better result.

IV. RESULTS

More than 500 sentences from Brown Corpus which are not a part of decentralized lexical library are taken and tested with all networks. Disambiguation accuracy is given in table-1. The performance is measured by '***mse***' in all five networks, given in table-2 (post bigram) and table-3 (pre bigram). From the tables, it is clear that, the trainable cascade forward back propagation network (M3) is having least average ***mse*** values in both post bigram and pre bigram methods. Disambiguation performance can also be viewed as a chart for each word after training session.

A sample of performance chart is shown in Fig-4. Training sessions of the five networks are given in Fig-5. Sense disambiguation performance is also calculated using the basic MATLAB command 'tic' & 'toc' (stop watch timer) for all types of networks, shown in chart-1 (post bigram) and chart-2 (pre bigram). Disambiguation accuracy is good in M3 (71.3%) than M2 (70.6%) network, shown in table-1, even though, the minimum average processing time taken by M1 is lesser (3 seconds for M1 and 5 seconds M3).

Table 1. Disambiguation Accuracy

% of Disambiguation Accuracy		
Network Type	Post bigram	Pre Bigram
M1	69.1	70.6
M2	68.5	63.6
M3	70.6	71.3
M4	68.5	65.7
M5	58	65

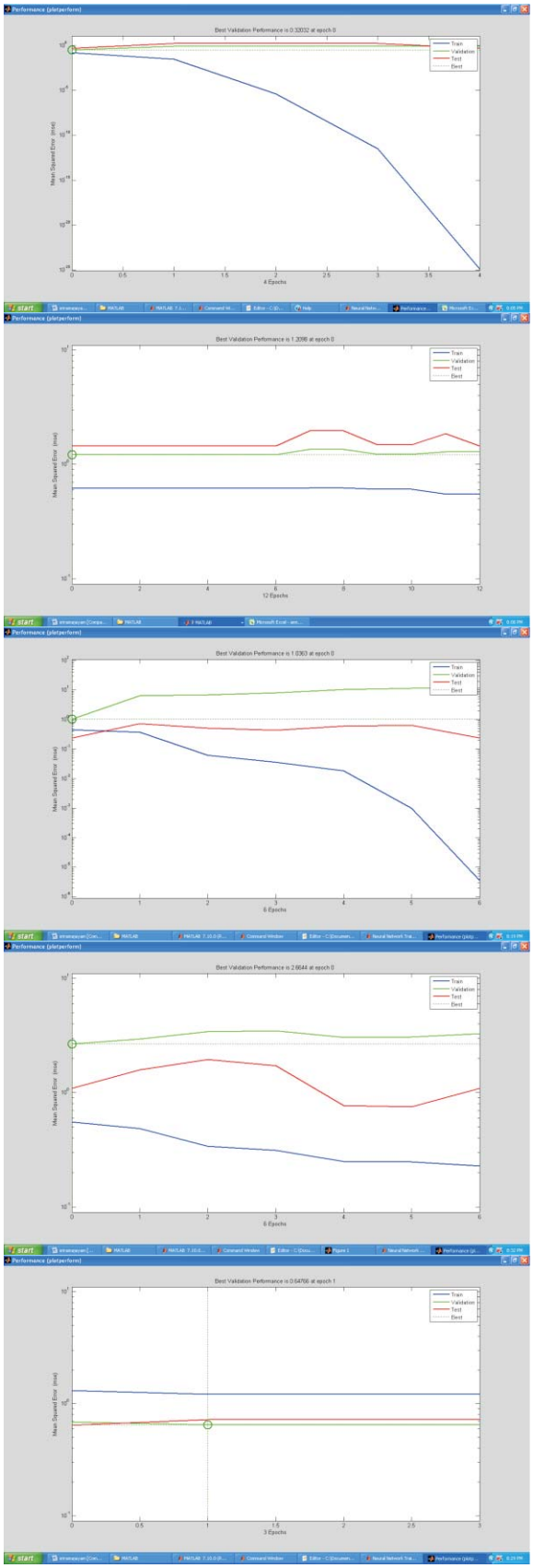

Fig. 4. Performance Chart

V. FUTURE ENHANCEMENT

Other disambiguation methods such as statistic methods, decision trees etc produce better accuracy. Even by using Neural Networks with enormous number of features, accuracy measured from 33.93% to 97.40% for words with more than two senses and 75% of accuracy for words with two senses (A. Azzini, C. da Costa Pereira, M. Dragoni and A. G. B. Tettamanzi, 2008). We have used networks with only two input neurons, means only two text features, for disambiguation. The percentage of accuracy varied from 50% to 86%, and on an average 71.3%. Accuracy level can be increased by raising the feature size as four (left two and right two words) or six(left three and right three) etc.

VI. CONCLUSION

We used different neural networks to disambiguate all ambiguous words in a sentence. Networks use the decentralized data set to make the disambiguation with only two features referred as post bigram and pre bigram. Among the networks, trainable cascade forward back propagation network produced least ***mse*** value 0.2483 with 6 neurons in post bigram and .144 with 8 neurons in pre bigram. And the network M3 achieved the average accuracy of 71.3%. We recommend the network M3 with 8 neurons as the best network architecture among networks we tested.

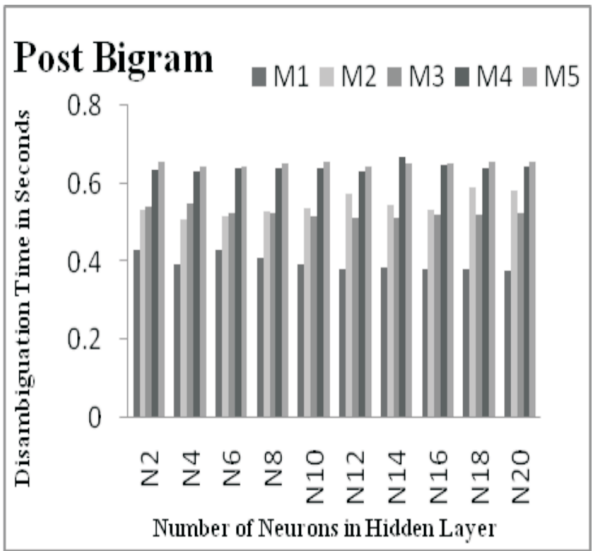

Fig. 5. Average disambiguation Time

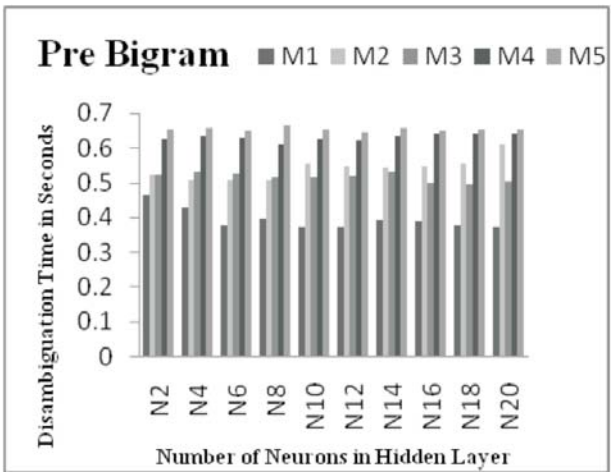

Fig. 6. Average disambiguation Time

Table 2: Post bigram performance Analysis

Neural Network Model	mse values [Post bigram]										
	N2	N4	N6	N8	N10	N12	N14	N16	N18	N20	Average
M1	0.5651	0.6719	0.6188	0.5583	0.5643	0.5612	**0.4796**	0.6615	**0.5553**	0.5934	0.5830
M2	0.6536	0.563	0.5912	0.5571	0.5799	0.563	0.5830	0.5907	0.6126	0.5909	0.5885
M3	**0.4020**	**0.2491**	**0.2483**	**0.3400**	**0.4221**	**0.3072**	0.5433	0.5576	0.5804	**0.5860**	**0.4296**
M4	0.6037	0.6686	0.5550	0.4308	0.5681	0.3801	0.6943	**0.5385**	0.7054	0.6014	0.5746
M5	0.9966	0.9966	0.9966	0.9966	0.9966	0.9966	0.9966	0.9966	0.9966	0.9966	0.9966

Table 3. Pre bigram performance Analysis

Neural Network Model	mse values [Pre bigram]										
	N2	N4	N6	N8	N10	N12	N14	N16	N18	N20	Average
M1	**0.2748**	0.6422	0.5865	0.3625	**0.339**	0.2871	0.5627	0.5343	0.5368	0.608	0.4734
M2	0.2978	0.5549	0.6257	0.6073	0.6789	0.5964	0.5451	0.562	0.5995	0.583	0.5651
M3	0.4027	**0.2551**	**0.3052**	**0.144**	0.5941	0.5859	**0.5336**	**0.5303**	0.6674	0.5412	**0.4559**
M4	0.6105	0.4728	0.531	0.4649	0.4256	**0.2865**	0.5789	0.5418	**0.5244**	**0.5411**	0.4977
M5	0.876	0.876	0.876	0.876	0.876	0.876	0.876	0.876	0.876	0.876	0.876

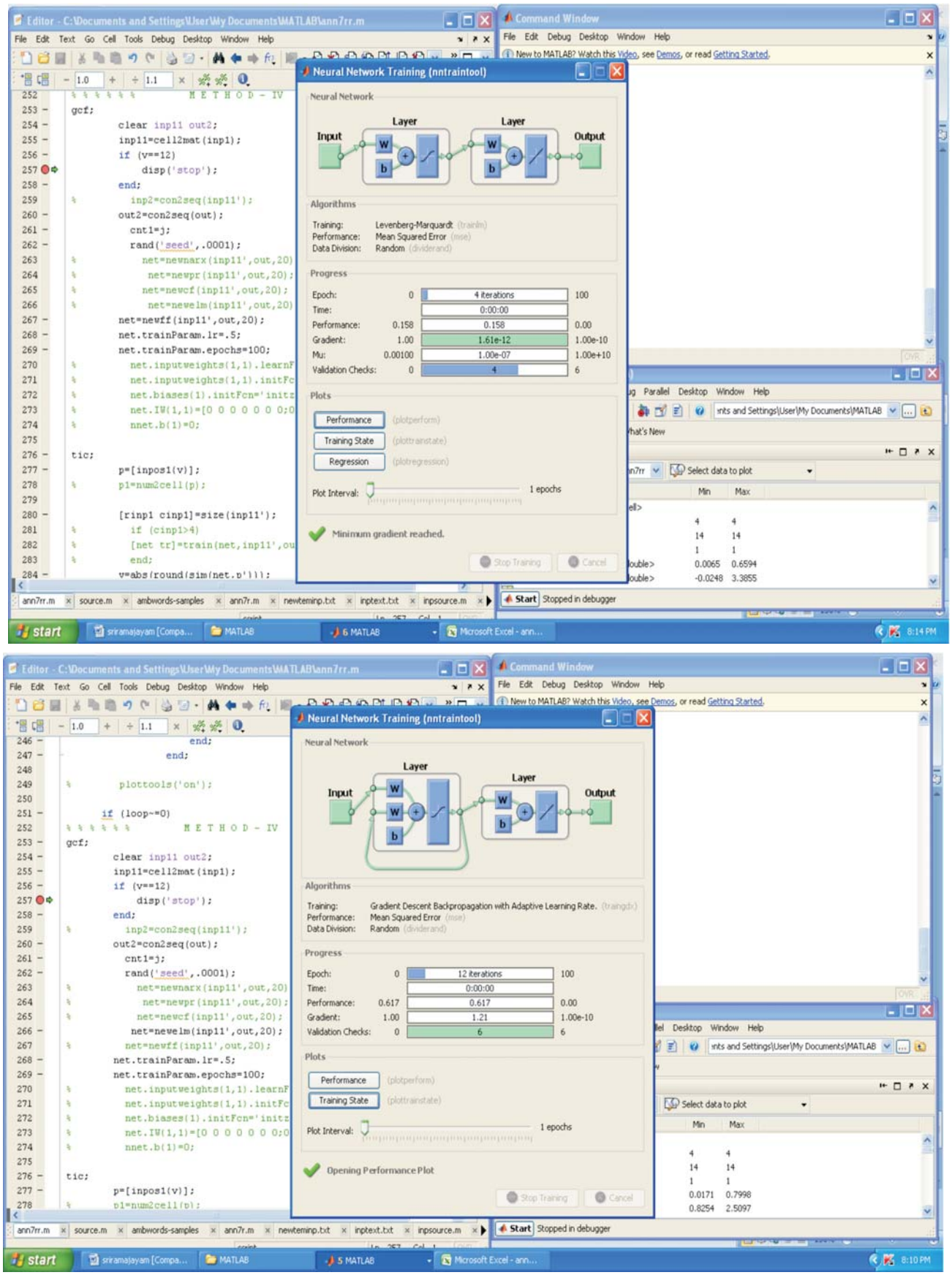

Fig. 7. Neural Network training

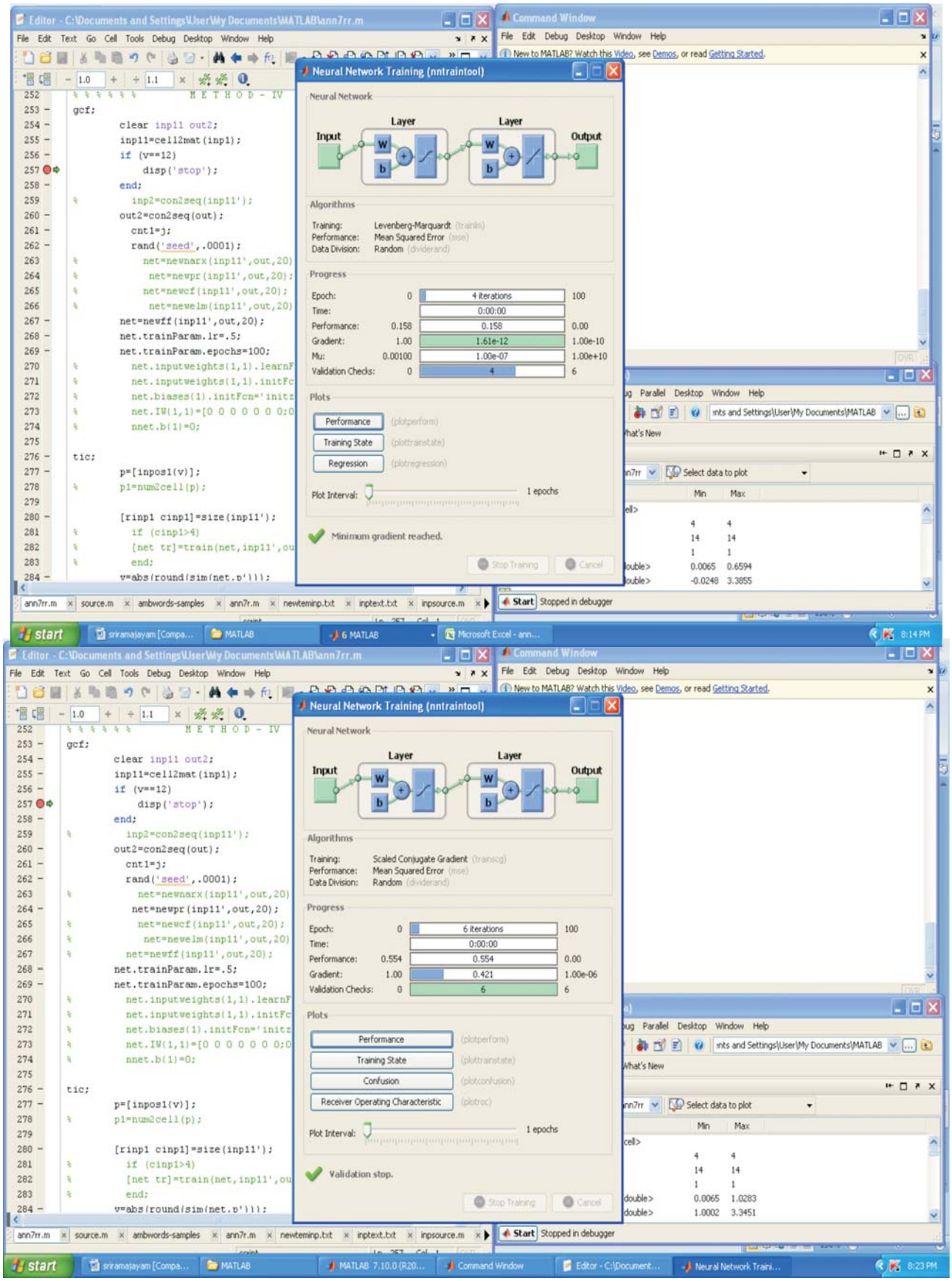

Fig. 8. Neural Network training

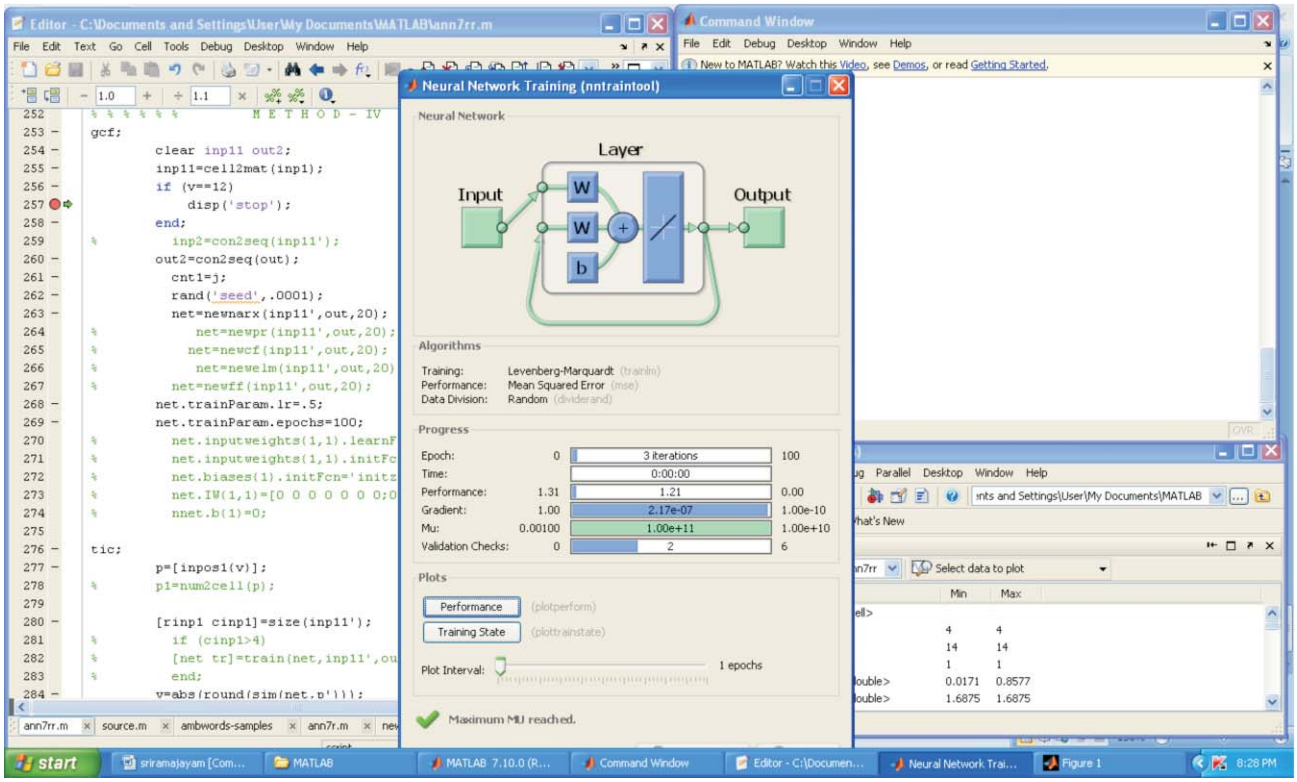

Fig. 9. Neural Network training

REFERENCES

[1] A. Azzini, C. da Costa Pereira, M. Dragoni, and A. Tettamanzi. Evolving Neural Networks for Word Sense Disambiguation, HIS'08, pages 332-337, LNCS, Springer, Septermber 2008.

[2] Atwell, E, 1988. *Transforming a Parsed Corpus into a Corpus Parser* in Kyto, M, Ihalainen, O &Risanen, M (editors), Corpus Linguistics, Hard and Soft: Proceedings of the ICAME 8th International Conference on English Language Research on Computerised Corpora, pp 61-70, Rodopi.

[3] Hinrich Schutze, Automatic word sense discrimination, Computation Linguistics, 1998, 24(1), pp.97-124.

[4] Leech, G, Garside, R & Atwell, E, 1983. The Automatic Grammatical Tagging of the LOB Corpus ICAME Journal of the International Computer Archive of Modern English Vol.7

[5] Ng. H.T. and Zelle J Corpus-Based Approaches to Semantic Interpretation in Natural Language Processing. AI Magazine, 1997, 18(4), PP. 45-64.

[6] Palmer, D.D. (2000). Tokenisation and sentence segmentation. In Dale, R.Somers, H. L., and Moisl, H. (Eds.), *Handbook of Natural Language Processing*. Marcel Dekker, Inc., New York, NY, USA.

[7] T. Pedersen, A decision tree of bigrams is an accurate predictor of word senses, in: Presented at Second Annual Meeting of the North American Chapter of the Association for Computational Linguistics, 2001.

[8] P.Tamilselvi, S.K.Srivatsa, Part-Of-Speech Tag Assignment Using Hidden Markov Model, International Journal of Highly reliable Electronic System, Vol-3, No-2, 2010.

[9] Yarowsky. D, Word sense disambiguation using statistical models of Roget's categories trained on large corpora, In: Zampolli, A., ed. Computatuion Linguistic'92. Nantas: Association for computational Linguistis, 1992, 454-460.

[10] You-Jin Chung, Sin-Jae Kang, Kyong-Hi Moon and Jong-Hyeok Lee, "Word Sense Disambiguation in a Korean-to-Japanese MT System Using Neural Networks", COLING 2002 Workshop on Machine Translation in Asia, pp. 74-80, 2002.

[11] Zhimao Lu, Ting Liu, and Sheng Li. Combining neural networks and statistics for chinese word sense

disambiguation. In Oliver Streiter and Qin Lu, editors, *ACL SIGHAN Workshop 2004*, pages 49-56.

[12] P.Tamilselvi, S.K.Srivatsa, Decentralized E-Dictionary (DED) for NLP task, Proceedings of ICMCS International conference on Mathematics and computer Science, India, 2009.

[13] P.Tamilselvi, S.K.Srivatsa, A Study on Lexicographical Information using open source lexical databases, Proceeding of NCRTCSE National conference on Recent Trends in Computer Science and Engineering, 2010.

[14] C. Fellbaum. WordNet: An Electronic Lexical Database. MIT Press, Cambridge, Massachusetts, 1998.

T. Tamilselvi obtained her MCA in 1995 and M.Phil (Computer Science) in 2003. Her research interest includes natural language processing, case based resoning, data decentralization etc. She published more than 10 papers in conferences and journals.