

GENERAL REGRESSION NEURAL NETWORK BASED POS TAGGING FOR NEPALI TEXT

ArchitYajnik

Department of Mathematics, Sikkim Manipal University, Sikkim, India

ABSTRACT

This article presents Part of Speech tagging for Nepali text using General Regression Neural Network (GRNN). The corpus is divided into two parts viz. training and testing. The network is trained and validated on both training and testing data. It is observed that 96.13% words are correctly being tagged on training set whereas 74.38% words are tagged correctly on testing data set using GRNN. The result is compared with the traditional Viterbi algorithm based on Hidden Markov Model. Viterbi algorithm yields 97.2% and 40% classification accuracies on training and testing data sets respectively. GRNN based POS Tagger is more consistent than the traditional Viterbi decoding technique.

KEYWORDS

General Regression Neural Networks, Viterbi algorithm, POS tagging

1. INTRODUCTION

Artificial neural networks plays a vital role in various fields like medical imaging, image recognition is covered in [1, 2, 3]and since last one decade it becomes popular in the field of Computational linguistics also. Due to the computational complexities sometimes it is not preferred for the big data analysis. General Regression Neural Network which is based on Probabilistic neural networks is one type of supervised neural network is computationally less expensive as compared to standard algorithms viz. Back propagation, Radial basis function, support vector machine etc is exhibited in [4]. That is the reason GRNN is considered for the Past of speech Tagging experiment for Nepali text in this article.

Several statistical based methods have been implemented for POS tagging [5]as far as Indian languages are concern. Nepali is widely spoken languages in Sikkim and neighbouring countries like Nepal , Bhutan etc. The use of ANN architecture is seldom for tagging[6]. To develop a parser and Morphological analyser for the natural languages POS tagging plays a pivotal role.

This article presents a neural network architecture based on the Statistical learning theory described in [4]. This neural network is usually much faster to train than the traditional multilayer perceptron network. This article is divided into five sections. After this introduction, the second section presents the General Regression Neural Network from the Mathematical point of view

while the experimental set up of GRNN architecture is discussed in the third section. In the fourth section, the result analysis of POS Tagging using GRNN and Viterbi algorithm is presented for Nepali text followed by Conclusion in fifth section and references.

2. GENERAL REGRESSION NEURAL NETWORKS

The detailed information about Probabilistic and General Regression Neural Networks is available in [4]. GRNN can briefly be introduced for the training set, $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$. To estimate the joint probability distribution for vectors \mathbf{x} and y say $f_{\mathbf{x},y}(\mathbf{x}, y)$ and therefore $f_{\mathbf{x}}(\mathbf{x})$, we may use a nonparametric estimator known as the Parzen – Rosenblatt density estimator. Basic to the formulation of this estimator is a kernel, denoted by $K(x)$, which has properties similar to those associated with a probability density function:

Assuming that x_1, x_2, \dots, x_N are independent vectors and identically distributed (each of the random variables has the same probability distribution as the others), we may formally define the Parzen – Rosenblatt density estimate of $f_{\mathbf{x}}(\mathbf{x})$ as

$$\hat{f}_{\mathbf{x}}(\mathbf{x}) = \frac{1}{Nh^{m_0}} \sum_{i=1}^N K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) \text{ for } \mathbf{x} \in R^{m_0} \quad (1)$$

where the smoothing parameter h is a positive number called bandwidth or simply width; h controls the size of the kernel. Applying the same estimator on $f_{\mathbf{x},y}(\mathbf{x}, y)$, the approximated value for the given vector \mathbf{x} is given by

$$F(\mathbf{x}) = \hat{f}(\underline{\mathbf{x}}) = \frac{\sum_{i=1}^N y_i K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right)}{\sum_{i=1}^N K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right)}$$

If we take Gaussian kernel i.e. $K(\mathbf{x}) = e^{-\mathbf{x}^2}$, we obtain,

$$\hat{f}(\mathbf{x}) = \frac{\sum_{i=1}^N y_i \exp\left(-\frac{D_i^2}{2\sigma^2}\right)}{\sum_{i=1}^N \exp\left(-\frac{D_i^2}{2\sigma^2}\right)} \quad (2)$$

where $D_i^2 = (\mathbf{x} - \mathbf{x}_i)^T (\mathbf{x} - \mathbf{x}_i)$ and σ is the standard deviation. $\hat{f}(\mathbf{x})$ can be visualized as a weighted average of all observed values y_i , where each observed value is weighted exponentially according to its Euclidean distance from x . The theory of General Regression Neural Networks discussed above is pertaining to only neuron in the output layer. The same technique can be

applied for the multiple neurons in the output layer also. Therefore the technique can be generalized as shown below:

Let w_{ij} be the target output corresponding to input training vector x_i and j^{th} output node out of the total p . Again let C_i be the centres chosen from the random vector x . Then

$$y_i = \frac{\sum_{i=1}^n w_{ij} h_i}{\sum_{i=1}^n h_i} \quad (3)$$

Here n be the number of patterns in the training set. The estimate y_j can be visualized as a weighted average of all the observed values, w_{ij} , where each observed value is weighted exponentially according to its Euclidean distance from input vector x and n is the number of patterns available in the input space.

$$\text{with } h_i = h_i(\sigma, C_i) = \exp\left(-\frac{D_i^2}{2\sigma^2}\right) \quad (4)$$

$$\text{where, } D_i^2 = (\mathbf{x} - \mathbf{C}_i)^T (\mathbf{x} - \mathbf{C}_i)$$

3. EXPERIMENTAL PROCEDURE AND RESULT

The survey of Part of Speech Tagging for Indian languages is covered by Antony P J (2011) in [7]. The details of the tags used for the experiment is available in [8, 9]. The total of 7873 Nepali words along with their corresponding text are collected. Out of which 5373 samples are used for training and the remaining 2500 samples for testing. The database is distributed in to $n = 41$ tags. Network architecture consists of $41 \times 3 = 123$ input neurons, 5373 hidden neurons which plays a role of centres $C_i (i = 1, 2, \dots, 5373)$ shown in (4) and 41 neurons in output layer.

3.1 Feature Extraction and input neurons

Transition (T) $_{n \times n}$ and Emission probability matrices (E) $_{n \times m}$ are constructed for both the sets viz. training and testing. Transition matrix demonstrates the probability of occurrence of one tag (state) after another tag (state) hence becomes a square matrix 41×41 . Whereas the emission matrix is the matrix of probability distribution of each Nepali word is allotted the respective tag hence it is of the size $n \times m$ (number of Nepali words). In order to fetch the features for i^{th} word say x_i , the i^{th} row, i^{th} column of the transition matrix and i^{th} row of the emission matrix are combined hence becomes $41 \times 3 = 123$ features for each word. Therefore the GRNN architecture consists of 123 input neurons.

3.2 Hidden Neurons

All the patterns (or Nepali words) are used as a centre. Euclidean distance is calculated between patterns and centres. Training set consists of 5373 words hence the same number of hidden neurons are incorporated in GRNN architecture

3.3 Output Neurons

As there are 41 tags, 41 output neurons constitute the output layer of the network. For instance if the word belongs to NN (common noun) category which is the first tag of the tag set then the first neuron has a value 1 and all others are 0.

4. RESULT ANALYSIS

As the General Regression Neural Network follows supervised learning, the network is assigned 123 features as an input layer and 41 neurons as an output layer for each word. The network is trained using 5373 patterns (Nepali words) and corresponding tags. The network took 19 minutes to get trained. In the first phase, the same training set is validated, 4451 words out of 5373 are observed to be correct. In 715 words are tagged in the same group where they belong for example the word “गरिनेछ” has the actual tag “VBF (Finite Verb)” but it is assigned “VBX (Auxiliary Verb)” hence all together, the network has achieved 96.13% accuracy. In the second phase the network is tested on the words does not belong to the training set which contains 2500 Nepali words. The network has achieved 63.88% and 10.4% for correct identification and Group identification accuracy respectively, hence it becomes 74.28% total accuracy.

The same sets are tested using the traditional statistical technique Viterbi. The Viterbi decoding algorithm is applied for POS tagging for several natural languages[10]. The result obtained is depicted in table 1. The table emphasis that Viterbi algorithm gives very poor performance (40%) in identifying the words which do not belong to the training set by which the transition and emission matrices are constructed.

Table: 1 (Result using GRNN)

No	Technique	Validation set	Accuracy (%)	Group Identification accuracy (%)	Total Accuracy (%)
1	GRNN	Training set (5373)	82.84	13.29	96.13
2	GRNN	Testing set (2500)	63.88	10.4	74.28
3	Viterbi	Training set (5373)	93	4.2	97.2
4	Viterbi	Testing set (2500)	37	3	40

Percentage wise analysis is depicted in Fig. 1. The information shown in horizontal axis indicates that out of 320 total tags of NNP (Proper Noun), 200 are classified correctly and remaining 120 are confused with NN (Common Noun) as far as the first column is concern. Table 2 demonstrates the tags identified incorrectly and got confused with other tags. Proper Noun (NNP) is confused in 120 cases with the common noun (NN) because the probability of occurrence of NN after NNP is 0.62 while the reverse case has the probability of occurrence 0. That is the reason NNP is confused with NN but NN is never got confused with NNP even though both the tags belong to Noun group only as mentioned in table 1.2. The whole experiment is carried out in Java.

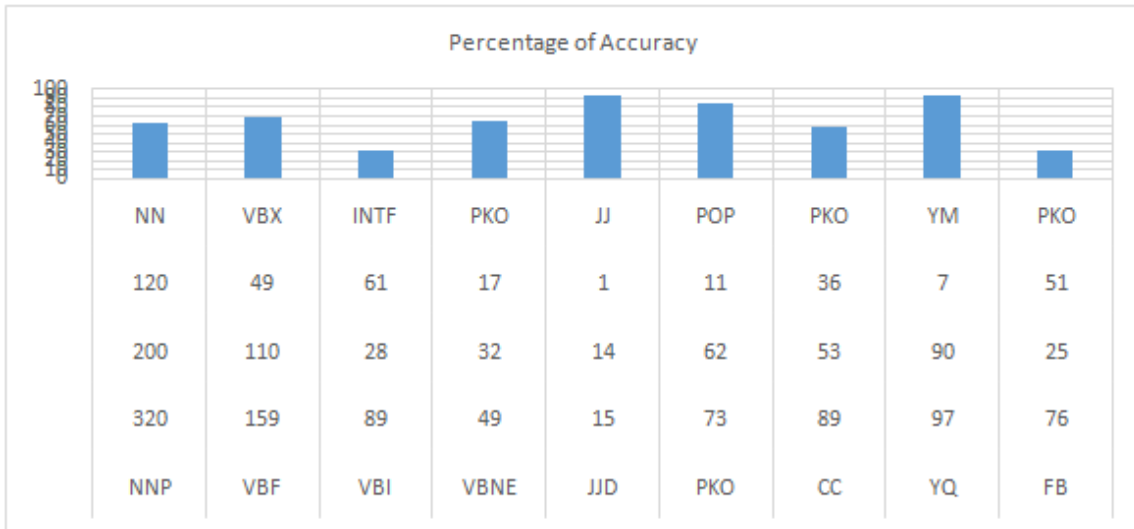


Figure. 1 Error Analysis using GRNN

Table 2 Confusion matrix

Tag	Frequency	Correct	Confusion	percentage	Group
NNP	320	200	NN	62.5	same
VBF	159	110	VBX	69.18	same
VBI	89	28	INTF	31.46	different
VBNE	49	32	PKO	65.31	different
JJD	15	14	JJ	93.33	same
PKO	73	62	POP	84.93	same
CC	89	53	PKO	59.55	different
YQ	97	90	YM	92.78	same
FB	76	25	PKO	32.89	different

5. CONCLUSIONS

In this article, GRNN based POS tagging approach is introduced for Nepali Text. Two techniques are employed viz. GRNN and Viterbi algorithm for this purpose. Section 4 demonstrates the outcome of the experiment on two data sets viz. training (5373 words) and testing (2500 words). Transition and emission probability matrices are constructed for both the techniques. Features are extracted from these matrices and used as an Input layer for GRNN as shown in section 3. Fully connected i.e. all the patterns (5373) are taken as centres GRNN architecture is trained using training set and outputs are validated on both training and testing sets. The result is compared with the traditional statistics based Viterbi algorithm. Table 1.1 shows that both the approaches yields satisfactory accuracy more than 96% as far as the training samples are concern but Viterbi fails completely (with 40% accuracy) while validated on testing set of 2500 patterns. On the other hand GRNN exhibits 74.28% accuracy. The confusion matrix (table 1.2) is generated on the output of GRNN on testing set. The accuracy may further be improved by collecting uniformly distributed data set.

ACKNOWLEDGEMENTS

The author acknowledges Department of Science and Technology, Government of India for financial support vide Reference no SR/CSRI/28/2015 under Cognitive Science Research Initiative (CSRI) to carry out this work.

REFERENCES

- [1] Richard O Duda and Peter E Hart, "Pattern Classification", 2006, Wiley-Interscience, New York, USA.
- [2] S. Rama Mohan, ArchitYajnik: "Gujarati Numeral Recognition Using Wavelets and Neural Network" Proceedings of Indian International Conference on Artificial Intelligence 2005, pp. 397-406.
- [3] ArchitYajnik, S. Rama Mohan, "Identification of Gujarati characters using wavelets and neural networks" Artificial Intelligence and Soft Computing 2006, ACTA Press, pp. 150-155.
- [4] Simon Haykin, "Neural Networks A Comprehensive Foundation" Second Edition, Prentice Hall International, Inc., New Jersey, 1999.
- [5] Prajadip Sinha et al. 2015. Enhancing the Performance of Part of Speech tagging of Nepali language through Hybrid approach, 5(5) International Journal of Emerging Technology and Advanced Engineering.
- [6] Tej Bahadur Shai et al. 2013. Support Vector Machines based Part of Speech Tagging for Nepali Text, Vol: 70-No. 24 International Journal of Computer Applications.
- [7] Antony P J et al. 2011. Parts of Speech Tagging for Indian Languages: A Literature Survey, International Journal of Computer Applications (0975-8887), 34(8).
- [8] <http://www.lancaster.ac.uk/staff/hardiea/nepali/postag.php>
- [9] <http://www.pan110n.net/english/Outputs%20Phase%20202/CCs/Nepal/MPP/Papers/2008/Report%20on%20Nepali%20Computational%20Grammar.pdf>
- [10] ArchitYajnik, "Part of Speech Tagging Using Statistical Approach for Nepali Text", International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:11, No:1, 2017, pp. 76-79.