ANN Based POS Tagging For Nepali Text

Archit Yajnik

Department of Mathematics, Sikkim Manipal University, Sikkim, India

ABSTRACT

The article presents Part of Speech Tagging for Nepali Text using three techniques of Artificial Neural networks. The novel algorithm for POS tagging is introduced .Features are extracted from the marginal probability of Hidden Markov Model. The extracted features are supplied to 3 different ANN architectures viz. Radial Basis Function (RBF) network, General Regression Neural Networks (GRNN) and Feed forward Neural network as an input vector for each word. Two different Annotated Tagged sets are constructed for training and testing purpose. Results are compared using all the 3 techniques and applied on both the sets. GRNN based POS tagging technique is found better as it produces 100% and 98.32% accuracies for both training and testing sets respectively.

KEY WORDS

Radial Basis Function, General Regression Neural Networks, Feed forward neural network, Hidden Markov Model, POS Tagging

1. INTRODUCTION

Natural Language Processing (NLP) is a diversified field of computational linguistics which in high demand for the researchersworld wide due to its large number of applications like language translation, Parsing, POS tagging etc. Among those POS tagging is one of the core part of NLP which is being used in other applications of computational linguistics. There are various techniques available in the literature for POS tagging like HMM based Viterbi algorithm, SVM etc.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. Following sentence illustrates annotated tagged Nepali sentence

मेरो PP नव NN अर्चित NNP हो VBX | YF

(My Name is Archit)

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

DOI: 10.5121/ijnlc.2018.7302

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 pivotalrole.

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 briefs the traditional ANN architectures viz Feed forward MLP, Radial basis function networks (RBF), General Regression Neural Network (GRNN). Experimental set up is highlighted in the third section followed by Result and discussion in the 4th section. The article is concluded in the fifth section.

2. ANN ARCHITECTURES

Single layer feedforward Multilayer Perceptron is based on backpropagation algorithm using Sigmoidal transfer function is successfully employed in various multiclass classification problems. Due to the existence of Centres, the kernel based technique RBF is widely used for the classification. The networks can be trained for centres as well as synaptic weights. The detail information about these networks is available in [4].
GRNN is also a kernel based technique in which Mahalanobis distance is calculated of each pattern with the corresponding centre. 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 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 $x_1, x_2, \ldots, x_N$ 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

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

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 x is given by

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

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 Ci 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

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

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

3. EXPERIMENTAL PROCEDURE

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]. Out of which 42100 samples (patterns) 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 x 3 = 123 input neurons, 42100 hidden neurons which plays a role of centres *Ci* (i = 1, 2, …,42100) shown in (4) and 41 neurons in output layer.

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 x 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 x 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 x 3 = 123 features for each word. Therefore the ANN architectures consists of 123 input 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 ANN architectures.

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.

3.1 FEED FORWARD NEURAL NETWORK

The MLP network is trained with 123 input neurons, 30 hidden neurons and 41 output neurons using sigmoidal transfer function till the error goal of computed output and original output reachesuptoExp(-16). The code is implemented in Matlab. The network is trained up to 800 epochs.

3.2 RADIAL BASIS FUNCTION NETWORK

The RBF network is trained with 123 input neurons, hidden neurons are same as the number of patterns and 41 output neurons using sigmoidal transfer function till the error goal of computed output and original output reachesuptoExp(-16). The code is implemented in Matlab. The network is trained up to 800 epochs.

3.3 GENERAL REGRESSION NETWORK

The synaptic weights are computed for GRN network as shown in section 2. There are 123 input neurons, hidden neurons are same as the number of patterns and 41 output neurons using Gaussian transfer function The code is implemented Java.

4 RESULT ANALYSIS

The training database contains 42100 words whereas testing set consists of 6000 words.Both database does not contain the words with multiple tags i.e. same nepali word with more than one different tags. The performance of all the three networks are depicted in table 1. Experiments demonstrate that all the architectures are better able to identify the suitable tags as far as the training set is concern but in the case of testing set except GRNN none other network performs efficiently. Table 1 exhibits the fact that 5899 out of 6000 testing samples are tagged properly where in other cases tagging accuracy is extremely poor.

Table: 1 Performance analysis of ANN techniques

Sr No	Technique	Accuracy (%) in train set	Accuracy (%) in test set	Number of correctly tagged words in test set
1	RBF	100	25	1500
2	GRNN	100	98.32	5899
3	Feed forward Neural Network	99.76	26.65	1599

As demonstrated the output of GRNN is portrayed on the test input shown below. Only the first word **नानी**(a child) is erroneously tagged PP (pronoun) but its actual tag is NN (common noun). It has happened because the frequency of the word **नानी**is 1 in the training set and the frequency of PP is highest that is 37 among all the tags.

Test input

नानी ले केरा खान्छ | मेरो नव अर्चित हो | श्री क्यानेपा ले उहाँ ले कारबाही लाई स्वीकार गरेको पुष्टि गर्नुभयो तर थप टिप्पणी गर्न अस्वीकार गर्नुभयो । मेरो नव केरा हो | पश्चिम जर्मनी का व्यापारी हरू ले बिहीबार को सरकारी बोन्ड निष्कासन को अगाडि वित्तीय बोन्ड मा कम चासो रहेको बताए । श्री क्यानेपा ले केरा खान्छ |

Tagged Output using GRNN

नानी **PP** ले **PLE** केरा **NN** खान्छ **VBF** | **YF** मेरो **PP** नव **NN** अर्चित **NNP** हो **VBX** | **YF** श्री **NN** क्यानेपा **NNP** ले **PLE** उहाँ **PP** ले **PLE** कारबाही **NN** लाई **PLAI** स्वीकार **NN** गरेको **VBKO** पुष्टि **JJ** गर्नुभयो **VBX** तर **CC** थप **JJ** टिप्पणी **NN** गर्न **VBI** अस्वीकार **NN** गर्नुभयो **VBX** । **YF** मेरो **PP** नव **NN** केरा **NN** हो **VBX** | **YF** पश्चिम **JJ** जर्मनी **NNP** का **PKO** व्यापारी **NN** हरू **HRU** ले **PLE** बिहीबार **NNP** को **PKO** सरकारी **JJ** बोन्ड **NN** निष्कासन **NN** को **PKO** अगाडि **PP** वित्तीय **JJ** बोन्ड **NN** मा **POP** कम **JJ** चासो **NN** रहेको **VBKO** बताए **VBF** । **YF** श्री **NN** क्यानेपा **NNP** ले **PLE** केरा **NN** खान्छ **VBF** | **YF**

5 CONCLUSIONS

GRNN achieves better output accuracy (98.32%) in testing set as compared to other two ANN techniques viz RBF and Feed Forward Neural Network as depicted in table 1. Words of testing set with less frequency of occurrence may be 1 or 2 in training set are failed to get an appropriate tag in the case of GRNN architecture. It concludes that GRNN technique which does not require any kind of training and yields the output only based on the distance provides much better accuracy as compared to the other ANN architectures which require training.

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.panl10n.net /english/ Outputs%20 Phase %202/CCs/Nepal/MPP/ Papers/2008/ Report %20 on %20 Nepali %20 Computational %20 Grammar.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.