Research article Available online www.ijsrr.org ISSN: 2279-0543

International Journal of Scientific Research and Reviews

Language Modeling Through Neural Networks to Increase Performance of Speech Recognition System

Mutcha Srinivasa Rao*

Sr. Technical Officer & Project Leader. AAI, C-DAC
NSG IT Park, Aundh, Pune, Maharashtra, India – 411 007

ABSTRACT:

Speech is the spoken version of natural language. Speech recognition in essence is a language dependent process. Hence linguistic knowledge has to be modeled in a form that can be employed by a general purpose automatic speech recognition system. Lexical Knowledge can be modeled using a pronunciation dictionary. Syntactic and Semantic knowledge can be represented in the form a word net for task specific speech recognition. Statistical grammars have been used for large vocabulary speech recognition. Currently, N-gram models are the most common and widely used models for statistical language modeling. This paper deals with building acoustic and language models using artificial neural networks to learn the language model.

Language model estimates the probability distributions of various linguistic units or their composites. Availability of large amount of training data (i.e., text) has led to improved quality of SLMs. This, in turn, has increased performance of ASR systems despite the fact that language models hardly take note of the fact that what is being modeled is language.

For a given speech signal, the goal of speech recognition is to generate the optimal word sequence subject to linguistic constraints. A sentence is composed of linguistic units such as words, syllables, phonemes. In speech recognition, a sentence model is assumed to be a sequence of models of such smaller units. The acoustic evidence provided by the acoustic models of such units is combined with the rules of constructing valid and meaningful sentences in the language to hypothesize the spoken sentence.

KEYWORDS: Language modeling, ASR, Speech Optimization, neural networks, Language modeling using neural networks.

Corresponding Author:-

Srinivasa Rao Mutcha
Sr. Technical Officer & Project Leader
Applied Artificial Intelligence Group (AAI), Centre for Development of Advanced Computing (C-DAC), NSG IT Park, Aundh, Pune, Maharashtra, India – 411 007
E-Mail: srinivas.mutcha@gmail.com, srinivasam@cdac.in

INTRODUCTION:

Language models are widely used in speech recognition, text classification, optical character recognition, etc. Artificial neural networks (NN) are also a powerful technique that is widely used in various fields of computer science. Though there are some works on connectionist natural language processing, a strange phenomenon is that in spite of the popularity of artificial neural networks, it is hard to find any work on language modeling using NN in the literature. There might be two reasons for the lack of work on using NN for language modeling. The first is that it is reasonable for one to think that the standard statistical method is more suitable for this problem. The second is that the size of the neural network needed for this problem is too huge and the training would be too slow to be tolerable.

BASICS OF LANGUAGE MODELING

Language model is used to assign a probability $P(\mathbf{W})$ to every possible word sequence **W**. Using Bayes' rule of probability,

$$P(W) = \prod_{i=1}^{n} P(Wt \mid ht)$$

Where $\mathbf{h}_t$ denote the history of word W_t, W_1, W_{t-1}. So the task of language model is to estimate the probability of a word given its history. Because there are a huge number of different histories, it is impractical to specify all $\mathbf{P(W_t \mid h_t)}$ completely. If one can map the histories into some number of equivalence classes, and let F be this mapping. Then the conditional probability can be approximated by $\mathbf{P(W_t \mid h_t) = P(W_t \mid Ø(h_t))}$. A commonly used equivalence classification is to consider the histories that end with same n-1 words as one equivalent class. For n=2, a bigram language model:

$P(W) = \prod_{i=1}^{n} P(Wt \mid Wt-1)$ is obtained.

NEURAL NETWORKS APPROACH TO LANGUAGE MODELING

The following figure depicts a Neural Network Architecture used for training the LM into the system.

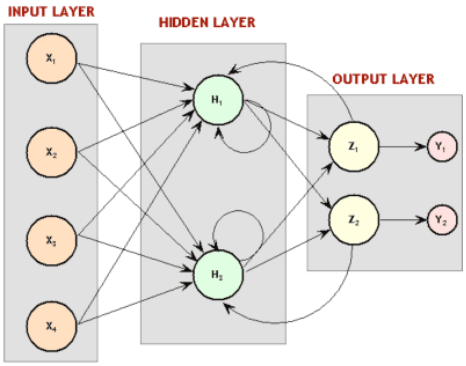

Fig. 1 Neural Network architecture for Language Model Training

Input and Output Encoding

The network considered, has V input units and V output units, where V is the vocabulary size. The i^{th} input unit is 1 if the current word is w_i. The value of i^{th} output unit represents the probability of w_i *being* the next word.

Error Function

Because the goal is to minimize the perplexity, so we use the logarithm of perplexity as error function, is used. This is same as the negative log likelihood.

$E = \sum_t Log\ Ot.Wt$

When training the neural network, the need is to minimize the above error function. It can be proven that using this error function the value of i^{th} output will converge to $\boldsymbol{P(W_i \mid W_j)}$ if the input to the network is word j w . So upon convergence, the network will be equivalent to a bigram language model without any smoothing. Without smoothing, the performance on test data will be very poor, so methods of preventing over-fitting will be very important to us. In this paper, early-stopping is used to prevent over-fitting, i.e. stop the training when the network achieves best performance on holdout data.

Activation Function

The sigmoid activation function is used for the output units, which guarantee the sum of the outputs is 1. Let the net input to each output unit be Net_i, then the sigmoid output O_i is:

$$\mathbf{O_i} = \frac{e^{Neti}}{\sum e^{Netj}}$$

where the net input is $Net_i = W_{ij} + \sum Wij\ Xij$ and X_{ij} is the j^{th} input to this unit.

Network Structure

The network considered, is a single layer network. The input units and output units are fully connected, so V' $(V + 1)$ weights are obtained (including bias weight).

An Issue On Computation Cost

A major problem for training such a neural network to learn language model is that it is very computationally expensive. So the aim is to try all possible ways to reduce the computational cost.

One important characteristic for the neural network is the sparsity of its inputs (i.e. most of them are zero). Using this fact, and notice the formula for updating weights in back-propagation algorithm:

$$\Delta W_{ij} = \eta \delta_i X_{ij}$$

So the weight will not be changed if the corresponding input value is zero. Thus a vast amount of computation is saved upon updating those weights with non-zero input value.

Training Method

Back-propagation algorithm is used to train the network. The initial weights are set to zero (This is equivalent to a uniform distribution). Batch training (i.e. update weights after a whole epoch) is employed. As mentioned earlier, it is particularly important to prevent over-fitting in this problem. A small constant learning rate is used along with early stopping to prevent over-fitting. The training is stopped when the perplexity on the holdout set reaches the lowest point.

Consider the following simple network:

- There is no hidden unit in the network.
- There is no bias weight.
- The output is the linear summation of input units.
- Use batch updating and the learning rate are small enough.
- The target value of j^{th} output unit is set to 1 and the other output units are set to '0' when the network is presented with a word pair (i, j).
- Use squared error as error function.

SUMMARY

Most natural language processing system accepts deterministic text. In contrast, speech recognition systems yield probabilistic output of word sequences. A network of word hypotheses is generally formed using word continuity constraint. Every node is associated with the strength of acoustic evidence in the form of likelihood. Now, sophisticated language models can be employed to hypothesize the most likely word sequence. Thus, language processors for speech recognition should be empowered with suitable strategies for utilizing acoustic evidence. In addition, they should take into account ill-formed phrases, hesitations, false starts, repetitions, incompletely spoken dialogue; the language processor can utilize information gleaned from previous utterances of the user and the current state of the dialogue model. In fact, unlike the prevalent bottom-up approach of speech recognition,

dialogue models should provide anticipatory information to language as well as acoustic processor so that they can adapt models to the current dialogue state. The adaptation of the acoustic processor can be in the form of dialogue state-specific lexicon; the language processor may dynamically adjust the probabilities of language model so as to suit the current state of dialogue.

REFERENCES:

[1] Lawrence Rabiner and Juang BH. Fundamentals of Speech Recognition. Prentice-Hall International, Inc., New Jersey, 1993.

[2] Albesano D, Gemello R and Mana F: Hybrid HMM-NN for speech recognition and prior class probabilities, 9th International Conference on Neural Information Processing (ICONIP) 2002.

[3] Kuan-Yu Chen and Berlin Chen. Relevance language modeling for speech recognition, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 10.1109/ICASSP.2011;5947621: 2011; 5568 – 5571.

[4] Kuo, Hong-Kwang Jeff, Fosler-Lussier, Eric, Jiang Hui, Lee, Chin-Hui: Discriminative training of language models for speech recognition, IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 1:10.1109/ICASSP.2002.5743720: 2002;I-325 - I-328

[5] Korkmazsky Jojic F and Shevade B. Boosting of Speech Recognition Performance by Language Model Adaptation. IEEE International conference on Aerospace Conference. 10.1109/AERO.2007.352980: 2007; 1 – 10.

[6] Bellegarda JR. A multispan language modeling framework for large vocabulary speech recognition. IEEE Transactions on Speech and Audio Processing, Digital Object Identifier:10.1109/89.709671, 1998;6(5):456 – 467

[7] Schwenk H. Trends and challenges in language modeling for speech recognition and machine translation, IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU) 2009.

[8] Christopher M. Bishop. Neural networks for Pattern Recognition, Oxford University Press, 2004.

[9] Moore RK and Peeling SM. Minimally distinct word pair discrimination using a back-propagation network, Computer Speech and Language, 1989; 3(2): 119- 131.

[10] Elman JL. Finding structure in time, CRL-TR-8801, University of California, San Diego, 1988.

[11] Florez Choque O, Cuadros Vargas E. Improving Human Computer Interaction through Spoken Natural Language, IEEE Symposium on Computational Intelligence in Image and Signal Processing (CIISP) 2007.