International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 3 Issue: 9 5421 - 5424

Conversion of NNLM to Back-off language model in ASR

Mr.Pappu kumar
Dept. of Electronics and telecommunication
JSPM's ICOER
Pune, India
Pappukumar1991@gmail.com

Dr.S.L.lahudkar
Dept. of Electronics and telecommunication
JSPM's ICOER
Pune, India
Swapnillahudkar@gmail.com

***Abstract*—**In daily life, automatic speech recognition is one of the aspect which is widely used for security system. To convert speech into text using neural network, Language model is one of the block on which efficiency of speech recognition depends. In this paper we developed an algorithm to convert Neural Network Language model (NNLM) to Back-off language model for more efficient decoding. For large vocabulary system this conversion gives more efficient result. Efficiency of language model depends on perplexity and Word Error Rate (WER)

Keywords-*NNLM,n-gram,back-off model,perplexity,WER*

*****

I. INTRODUCTION

In everyday people need key access for different security application. Speech recognition is one of the key contexts for security purpose. In preceding few decades speech recognition becomes very beneficial for security purpose. There is various method or technique of speech recognition but decoding with neural network become popular in last decade. Language modeling is very chief aspect developed in field of speech recognition.. State of the art decoders for automatic speech recognition generally utilize back-off n-gram language models in decoding . a back off n-gram language model P(ώ|h) takes form

$$P(ώ|h)=\begin{cases} P(ώ|h) & \text{if } hώ \in P \\ \alpha h P(ώ|h') & \text{if } hώ \notin P \end{cases}$$

Where w is the current word; h is the history or last n-1 words; h' is the truncated history obtained by dropping the first word in h ; and α(h) is a back-off weight that enforces normalization. The set P contains the n-grams for which we keep explicit probability estimates $P(ώ|h)$; all other n-gram probabilities are computed by backing off to a lower order estimate. The distribution $P(ώ|h')$ as well as lower-order distributions are represented in similar fashion. Since the majority of probabilities are evaluated using back-off estimates, the model can be represented with a modest number of parameters $P(ώ|h)$. Back-off language models have been expansively studied, and very efficient decoding systems have been developed.

Generally, language models are a precarious factor in many speech and language processing technologies, like speech recognition and accepting, voice searching, informal interaction and machine conversion. In last few decades, several advanced language modeling ideas have been proposed. Some of the methods have engrossed on incorporating linguistic information such as composition and semantics whereas others have focused on crucial modeling and parameter estimation performances. While incredible growth has been made in language modeling, n-grams are still very much the state-of-the-art due to the ease of the model and worthy performance they can achieve. Language models play a significant role in large vocabulary speech recognition and statistical machine translation systems.

Recently, a novel approach has been developed that carry out the estimation task in a continuous space model. The basic concept is to project the word indices onto a continuous space and to use a probability estimator operating on this space. The resulting probability functions are plane functions of the word representation, better generalization to unidentified n-grams can be predictable.

Type Therefore, we propose a hierarchical implementation: starting from a lower order NNLM, we grow back-off models of successively higher order using higher-order NNLMs. At each level, n-gram histories are restricted to those retained in the lower-order language model, thereby making the overall pruning computation manageable. Note that this is similar to algorithms for "growing" variable length n -gram models [10]. Neural network LMs were introduced in [4], [5] as a means to improve discrete models. Standard n-gram back-off LMs rely on a discrete representation of the vocabulary, where each word is associated with a discrete index. In contrast, NNLMs are based on the idea of a continuous word representation, where each word is associated with a real-valued feature vector. In this continuous space, distributionally similar words are neighbors. Thus n-gram distributions are expressed as a smooth function of the word representation, and can take into the account underlying similarities between words.

II. LITRETURE REVIEW

Several approaches for building varigram models have been presented.

Ebru Arisoy, Stanley F. Chen, Bhuvana Ramabhadran and Abhinav Sethy entitled "Converting Neural Network Language Models into Back-off Language Models for Efficient Decoding in Automatic Speech Recognition"[1] gives the result conversion of back-off language model up-to 4-gram.

We already mentioned a recent one [2]. Naturally, also various methods for clustering the model units have been developed.

During the last years there has been growing interest in using neural networks for language modeling. In contrast to the well known back-off n-gram language models[],the neural network approach attempts to overcome the data sparseness problem by performing the estimation in a continuous space. This type of language model was mostly used for tasks for

which only a very limited amount of in-domain training data is available. In this paper they present new algorithms to train a neural network language model on very large text corpora. This makes possible the use of the approach in domains where several hundreds of millions words of texts are available. The neural network language model is evaluated in a state-of-the-art real-time continuous speech recognizer for French Broadcast News. Word error reductions of 0.5% absolute are reported using only a very limited amount of additional processing time.

Holger Schwenk [4] entitled "Continuous space language models" describes the usage of a neural network language model for huge vocabulary continuous speech recognition. The underlying idea of this approach is to attack the data sparseness problem by performing the language model probability estimation in a continuous space. Highly efficient learning algorithms are described that enable the use of training corpora of several hundred million words. It is also shown that this approach can be incorporated into a large vocabulary continuous speech recognizer using a lattice rescoring framework at a very low additional processing time. The neural network language model was thoroughly evaluated in a state-of-the-art large vocabulary continuous speech recognizer for several international benchmark tasks, in particular the NIST evaluations on broadcast news and conversational speech recognition. The new method is compared to four-gram back-off language models trained with modified Kneser–Ney smoothing which has often been reported to be the best known smoothing method. Usually the neural network language model is interpolated with the back-off language model. In that way, consistent word error rate reductions for all considered tasks and languages were achieved, ranging from 0.4% to almost 1% absolute.

T. Mikolov, S. Kombrink, L. Burget, J. Cernocky, and S. Khudanpur[7] entitled "Extensions of recurrent neural network language model," compared Recurrent vs feedforward neural networks. Recurrent networks have possibility to form short term memory, so they can better deal with position invariance; feedforward networks cannot do that.Also, recurrent networks can learn to compress whole history in low dimensional space, while feedforward networks compress (project) just single word. In recurrent networks, history is represented by neurons with recurrent connections history length is unlimited. In feedforward networks, history is represented by context of N- 1 words.it is limited in the same way as in N-gram backoff models.

III. NEURAL NETWORK LANGUAGE MODEL

In neural network language models [2], [3], [4], a neural network is used to estimate language model probabilities. The input to the neural network is the words in the history and the output is a probability distribution over the predicted word. Input words are projected into a continuous multi-dimensional feature space, after which n-gram probabilities can be computed in a straight- forward fashion from the network. The expectation is that words that are semantically or grammatically related will be mapped to similar locations in the continuous space, allowing NNLMs to generalize well to unseen n-grams.

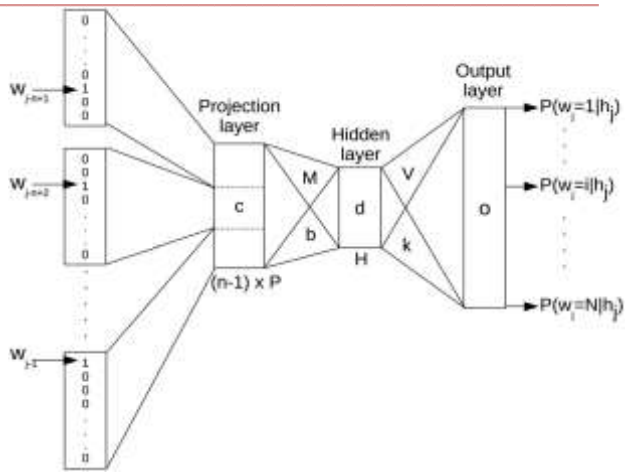


Figure:Architectur of Neural network language model

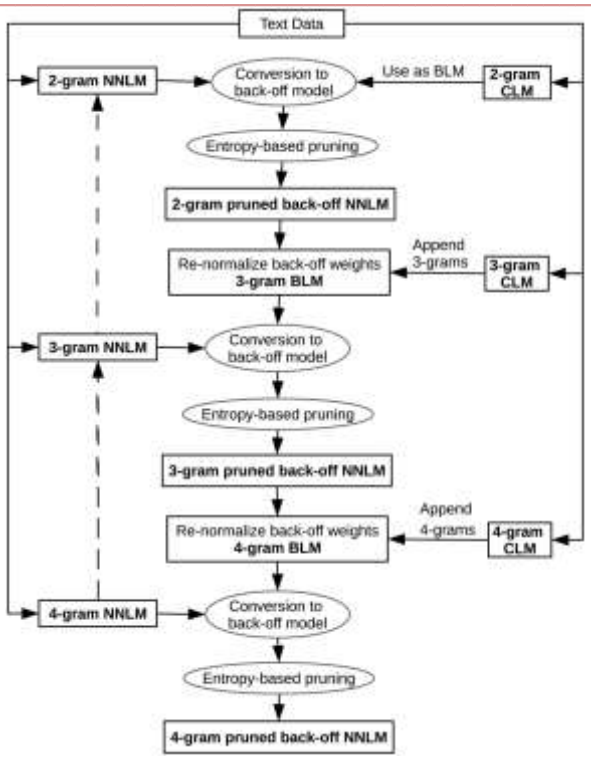
A typical NNLM consists of input, projection, hidden and output layers. Fig. 4.4 shows the architecture of a feed-forward neural network language model, following the notation given in [4]. Each word in the input vocabulary (containing words) is represented by an n-dimensional sparse vector where the entry corresponding to the index of that word is 1 and the rest of the entries are 0. The discrete representations of each of the n-1 previous words in the history are concatenated to form the input to the network. Each history word is then mapped to its continuous space representation (of dimension P) using a shared linear projection, an N xP matrix where the ith row of the matrix contains the continuous space feature representation of the ith word in the vocabulary.Equations.

IV. NNLM INTO A BACK-OFF LANGUAGE MODEL

For a language model to be expressed compactly as a back-off model, the probabilities of most n-grams must be proportional to their back-off probabilities (corresponding to the term α(h)P(w|h') in Eq. 1. This property does not hold for NNLMs, so to represent NNLM probabilities exactly over the output vocabulary requires |V|n-1 x |Vo| parameters in general, where V is the complete vocabulary. However, if the background language model in Eq. 2 is a back-off language model, we can take advantage of its structure to represent the overall NNLM as a back-off model. Substituting Eq. 1 for PBLM(w|h) in Eq. 1, we obtain

$$P(w|h)=\begin{cases}\beta(h)PNNLM(w|h) & if\ w \in Vo \\ PBLM(w|h) & if\ w \notin Vo \wedge hw \in PBLM \\ \alpha(h)Pblm(w|h') & if\ w \notin Vo \wedge hw \notin PBLM\end{cases}$$

While we can represent the overall NNLM as a back-off model exactly, it is prohibitively large as noted above. The technique of pruning can be used to reduce the set of n-grams P for which we explicitly store probabilities P(w|h) . In this paper, we use entropy-based pruning [9], the most common method for pruning back-off language models.



A naive implementation for converting a NNLM into a back-off model is to build the entire unpruned n-gram model Eq. 4 before performing pruning. However, this is impractical for vocabularies of any reasonable size; e.g., if |V|, |Vo|>10 K words, an unpruned 4-gram model contains at least 10^6 n - grams. To make pruning tractable, we propose a hierarchical algorithm where we build pruned models of increasingly higher order, and use the pruned lower-order models to constrain which n-grams are considered in the next higher-order model. In particular, given a pruned (m-1)-gram model, we only consider m-grams of the form for that belong to the lower-order model. All other m-grams are automatically pruned, or more accurately, never added to the model in the first place. Given this restriction, we need only consider K x |Vo| m-grams for pruning at a given level if there are items in the pruned lower-order model.

We outline the complete process for converting a 4-gram feedforward NNLM into a 4-gram back-off language model in Fig. 2. First, 2-gram, 3-gram and 4-gram NNLMs and conventional language models (CLMs) are built from the training data. Alternatively, 2-gram and 3-gram NNLM probabilities are computed from the 4-gram NNLM instead of training these models (shown with the dashed lines in Fig. 2). Our hierarchical implementation starts from 2-grams. We use Eq. 4 to combine the 2-gram NNLM with the background 2-gram CLM, giving us a 2-gram back-off NNLM. This model is quite large, containing |V| x |Vo| 2-grams, not including 2-grams coming from the background CLM. We apply entropy-based pruning, producing a 2-gram pruned back-off NNLM. The size of this model is determined by a pruning threshold. To create the 3-gram background language model, we append the 3-grams from the 3-gram CLM to the 2-gram pruned back-off NNLM and recompute the α(h) 's to renormalize the model. Then, we repeat this procedure until the highest-order pruned back-off NNLM is obtained. When creating the initial back-off model for 3-grams and above, we add only n-grams from the NNLM that are extensions of n-grams found in the lower-order pruned back-off NNLM. Note that the proposed hierarchical approach lets us use lower-order NNLMs for backing off and same-order conventional language models for smoothing zero probability events.

IV. RESULT

In this section, the experimental results are given only for the approach where lower-order NNLM probabilities are extracted from lower-order NNLMs trained on the text data. The alternative approach where lower-order NNLM probabilities are computed from the 4-gram NNLM by only setting the activations for lower-order histories yields similar results. We first investigate the effect on perplexity of converting a NNLM into a back-off language model. Fig. 8.1 shows the held-out set perplexity for NNLMs of various order (up to 6-gram, solid lines) as well as for the corresponding back-off language models (up to 4-gram, dashed lines).

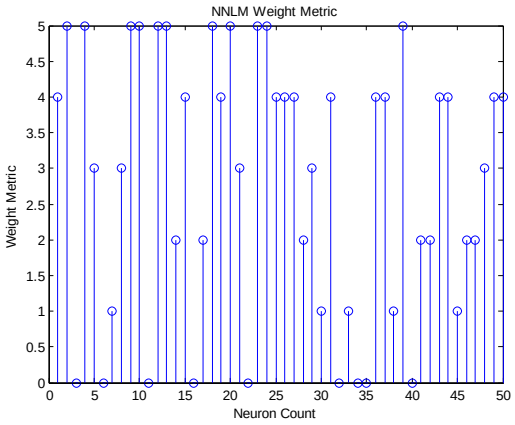


Figure: Plot of neuron count Vs weight metric

N-grams	Entropy	Wieght Error
1	10	220.2647
2	10	2202647
3	4	0.54598
4	2	0.073891
5	1	0.027183
6	1	0.02783

Table: Entropy and Wieght Error for each N-gram

Above result is calculated for 72M neuron points. Here also as value of n-gram increses WER decreases. All models are trained on the smaller 72M word training set. We do the conversion for only up to 6-grams since this is the highest-order model we use in first-pass decoding. In addition, we show results for before and after these models are interpolated with the baseline 4-gram language model. Before interpolation with the baseline, the 4-gram NNLM yields a better perplexity than the baseline, while the back-off NNLM with 202M n-grams is a little worse and the back-off NNLM. After interpolation, the

perplexities of both back-off NNLMs are significantly better than the baseline, with the original NNLM achieving lower perplexities then its back-off counterparts.

REFERENCES

[1] E. Arisoy, S. F. Chen, B. Ramabhadran, and A. Sethy, "Converting neural network language models into back-off language models for ef- ficient decoding in automatic speech recognition," in Proc. ICASSP, 2013, pp. 8242–8246.
[2] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural proba- bilistic language model," J. Mach. Learn. Res., vol. 3, pp. 1137–1155, 2003.
[3] H. Schwenk and J.-L. Gauvain, "Training neural network language models on very large corpora," in Proc. HLT-EMNLP, 2005, pp. 201–208.
[4] H. Schwenk, "Continuous space language models," Comput. Speech Lang., vol. 21, no. 3, pp. 492–518, Jul. 2007.
[5] H.-K. J. Kuo, E. Arisoy, A. Emami, and P. Vozila, "Large scale hierarchical neural network language models," in Proc. Interspeech, Port- land, OR, USA, 2012.
[6] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, and S. Khudanpur, "Recurrent neural network based language model," in Proc. Inter-speech, 2010, pp. 1045–1048.
[7] T. Mikolov, S. Kombrink, L. Burget, J. Cernocky, and S. Khudanpur, "Extensions of recurrent neural network language model," in Proc. ICASSP, 2011, pp. 5528–5531.
[8] H. Schwenk and J.-L. Gauvain, "Connectionist language modeling for large vocabulary continuous speech recognition," in Proc. ICASSP, Or- lando, FL, USA, 2002, pp. 765–768.
[9] A. Stolcke, "Entropy-based pruning of backoff language models," in Proc. DARPA Broadcast News Transcription and Understanding Work- shop, Lansdowne, VA, USA, 1998, pp. 270–274.
[10] M. Siu and M. Ostendorf, "Variable n-grams and extensions for con- versational speech language modeling," IEEE Trans. Acoust., Speech, Signal Process., vol. 8, no. 1, pp. 63–75, Jan. 2000.
[11] V. Siivola and B. Pellom, "Growing an n-gram model," in Proc. Inter- speech, 2005, pp. 1309–1312.
[12] V. Siivola, T. Hirsimaki, and S. Virpioja, "On growing and pruning Kneser-Ney smoothed n-gram models," IEEE Trans. Audio, Speech, Lang. Process., vol. 15, no. 5, pp. 1617–1624, Jul. 2007.
[13] S. Virpioja and M. Kurimo, "Compact n-gram models by incremental growing and clustering of histories," in Proc. Interspeech—ICSLP, Pittsburgh, PA, USA, 2006, pp. 1037–1040.
[14] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," Comput. Speech Lang., vol. 13, no. 4, pp. 359–394, 1999.
[15] P. F. Brown, V. J. Della Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer, "Class-based n-gram models of natural language," Comput. Linguist., vol. 18, no. 4, pp. 467–479, 1992.
[16] E. Arisoy, T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Deep neural network language models," in Proc. NAACL-HLT Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Lang. Model. for HLT, Montreal, QC, Canada, Jun. 2012, pp. 20–28.
[17] J. Goodman, "Classes for fast maximum entropy training," in Proc. ICASSP, 2001, pp. 561–564.
[18] A. Emami, "A neural syntactic language model," Ph.D. dissertation, Johns Hopkins Univ., Baltimore, MD, USA, 2006.
[19] G. Zweig and K. Makarychev, "Speed regularization and optimality in word classing," in Proc. ICASSP, Vancouver, BC, Canada, 2013, pp. 8237–8241.
[20] F. Morin and Y. Bengio, "Hierarchical probabilistic neural network language model," in Proc. AISTATS, 2005, pp. 246–252.
[21] A. Mnih and G. Hinton, "A scalable hierarchical distributed language model," in Proc. NIPS, 2008, pp. 1081–1088.
[22] H.-S. Le, I. Oparin, A. Allauzen, J.-L. Gauvain, and F. Yvon, "Structured output layer neural network language models for speech recognition," IEEE Trans. Audio, Speech, Lang. Process., vol. 21, no. 1, pp. 197–206, Jan. 2013.
[23] Y. Bengio and J.-S. Senecal, "Quick training of probabilistic neural nets by importance sampling," in Proc. AISTATS, 2003.
[24] A. Mnih and Y. W. Teh, "A fast and simple algorithm for training neural probabilistic language models," in Proc. ICML, Edinburgh, U.K., 2012.
[25] H.-K. J. Kuo, L. Mangu, A. Emami, I. Zitouni, and Y.-S. Lee, "Syntactic features for arabic speech recognition," in Proc. ASRU, Merano, Italy, 2009, pp. 327–332.
[26] T. Mikolov, A. Deoras, D. Povey, L. Burget, and J. Cernocky, "Strategies for training large scale neural network language models," in Proc. ASRU, 2011, pp. 196–201.
[27] A. Deoras, T. Mikolov, S. Kombrink, M. Karafiat, and S. Khudanpur, "Variational approximation of long-span language models for LVCSR," in Proc. ICASSP, Prague, Czech Republic, 2012, pp. 5532–5535.
[28] A. Deoras, T. Mikolov, S. Kombrink, and K. Church, "Approximate inference: A sampling based modeling technique to capture complex dependencies in a language model," Speech Commun., vol. 55, no. 1, pp. 162–177, Jan. 2013.
[29] G. Lecorve´ and P. Motlicek, "Conversion of recurrent neural network language models to weighted finite state transducers for automatic speech recognition," in Proc. Interspeech, Portland, OR, USA, 2012.
[30] W. Wang, A. Stolcke, and M. P. Harper, "The use of a linguistically motivated language model in conversational speech recognition," in Proc. ICASSP, 2004, pp. 261–264.
[31] S. F. Chen, B. Kingsbury, L. Mangu, D. Povey, G. Saon, H. Soltau, and G. Zweig, "Advances in speech transcription at IBM under the DARPA EARS program," IEEE Trans. Audio, Speech, Lang. Process., vol. 14, no. 5, pp. 1596–1608, Sep. 2006.