Improved Neural Network Based Language Modelling and Adaptation

J. Park, X. Liu, M.J.F. Gales and P.C. Woodland

Department of Engineering, University of Cambridge,
Trumpington St., Cambridge, CB2 1PZ, U.K.
{jhp33, xl207, mjfg, pcw}@eng.cam.ac.uk

Abstract

Neural network language models (NNLM) have become an increasingly popular choice for large vocabulary continuous speech recognition (LVCSR) tasks, due to their inherent generalisation and discriminative power. This paper present two techniques to improve performance of standard NNLMs. First, the form of NNLM is modelled by introduction an additional output layer node to model the probability mass of *out-of-shortlist* (OOS) words. An associated probability normalisation scheme is explicitly derived. Second, a novel NNLM adaptation method using a cascaded network is proposed. Consistent WER reductions were obtained on a state-of-the-art Arabic LVCSR task over conventional NNLMs. Further performance gains were also observed after NNLM adaptation.

Index Terms: Neural Network Language Model, Language Model Adaptation

1. Introduction

Statistical language models (LM) play an important role in state-of-the-art large vocabulary continuous speech recognition (LVCSR) systems. Back-off n-gram and class-based LMs [1, 2] are the dominant language models used in LVCSR systems. However, when only limited amounts of text data is available in training and adaptation, the generalization ability of these discrete, non-parametric models remain limited. To handle this data sparsity problem, a range of language modelling techniques based on a continuous vector space representation of word sequences have been proposed [3, 4, 5]. Among these one of the most successful schemes is the neural network LM (NNLM) [6, 7]. Due to their inherently strong generalisation and discriminative power, they have become an increasingly popular choice for LVCSR tasks [8, 9].

To reduce computational cost, existing forms of NNLMs only model the probabilities of a small and more frequent subset of the whole vocabulary, commonly referred to as the *shortlist*. The associated network's output layer contains only nodes for in-shortlist words. Two issues arise when adopting this conventional form of NNLM architecture. First, NNLM parameters are trained only using the statistics of in-shortlist words thus introduces an undue bias to them. This may poorly represent the properties of complete word sequences found in the training data as the n-gram sequences have "gaps" in them. Secondly, as there is no explicit modelling of probabilities of *out-of-shortlist* (OOS) words in the output layer, statistics associated with them will also be discarded in network optimisation. In order to address these issues, a modified form of NNLM architecture with an additional OOS node in the output layer is proposed in this paper. In addition to in-shortlist words, this NNLM models the probability mass of OOS words following arbitrary contexts. It ensures that the LM probabilities of in-shortlist words are smoothed by the OOS probability mass in training. As all training data is used and there is only a minimum increase of weight parameters associated with the additional output node, a more robust NNLM parameter estimation may be also obtained. An appropriate scheme to redistribute the OOS probability mass is also required for this modified NNLM.

In order to improve robustness to varying styles or tasks, unsupervised language model adaptation to a particular broadcast show or conversation, for example, may be used. Due to the previously mentioned data sparsity issue, directly adapting n-gram probabilities is impractical on limited amounts of data. To cope with this problem, an alternative and more general approach based on weighted finite state transducers (WFSTs) was investigated for LM combination and adaptation [15]. Continuous space language modelling techniques may be considered for their stronger generalisation ability [10, 11]. In this paper an NNLM adaptation scheme by cascading an additional layer between the projection and hidden layer is proposed. This scheme provides a direct adaptation of NNLMs via a non-linear, discriminative transformation to a new domain.

The rest of the paper is organized as follows. The conventional form of NNLMs is reviewed in section 2. An improved NNLM architecture with an additional OOS output node is also presented, together with probability normalization schemes for both forms of models given in section 2.2. A cascaded network based NNLM adaptation scheme is proposed in section 3. In section 4 various NNLMs are evaluated on a state-of-the-art Arabic broadcast transcription task.

2. Neural Network Language Model

The standard NNLM projects a set of contexts $h_t = w_{t-1} \ldots w_{t-n+1}$ onto a continuous vector space, then calculates the LM probability for each word given a history, $P(w_t = i|h_t)$. It is possible to represent a full context span probability distribution for all words following any history of $n-1$ words without back-off to lower order distributions as in n-gram LMs.

2.1. Architecture of network with OOS node

The architecture of conventional NNLMs is based on a fully-connected multi-layer perceptron (MLP) structure. The inputs to the network are the indices of the $n-1$ history words in the input vocabulary V_{in}. Between input and output layers, there are two hidden layers for projection and achieving non-linear probability estimation. For a N word input vocabulary, V_{in}, and P

This work was in part supported by DARPA under the GALE programme via a subcontract to BBN Technologies. The paper does not necessarily reflect the position or the policy of the US Government and no official endorsement should be inferred.

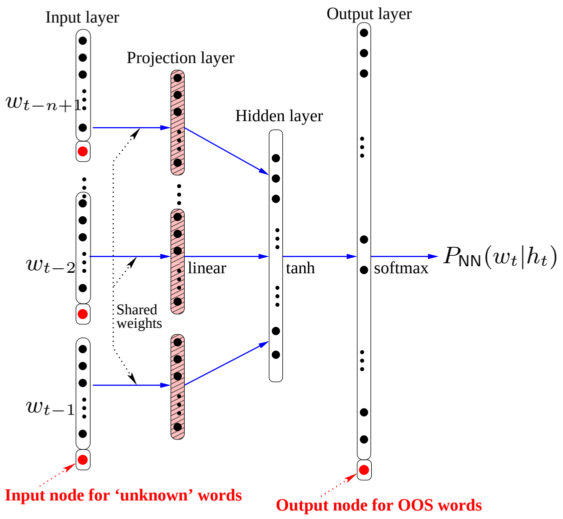

Figure 1: *Architecture of NNLM with OOS output node.*

dimensional continuous projection space, the input to the neural network are coded as the 1-of-k coding. This simplifies the projection layer calculation as one only needs to copy the i^{th} row of the $N \times P$ dimensional projection matrix. The projection matrix is shared for different word positions in the history context. The outputs of an NNLM are the posterior probabilities of all words in the output shortlist vocabulary, V_{sl}, following a given history.

As discussed in section 1, in conventional NNLMs the output layer contains only nodes for in-shortlist words. This form of architecture has undesirable impacts on the training of NNLMs. In order to handle these issues, an improved form of NNLM architecture with an additional OOS node in the output layer is proposed. This is shown in figure 1. In addition to in-shortlist words, this form of NNLM also explicit models the probability mass of OOS words following arbitrary contexts, This ensures the probabilities of in-shortlist words can be automatically smoothed by the OOS probability mass during neural network training. As no training data is discarded, a more robust NNLM weight normalisation can also be obtained.

Both types of NNLMs, without or with an OOS output layer node, can be trained using an error back-propagation method to minimise the training data perplexity (PP) until convergence. The error criterion used is cross-entropy with a weight decay regularisation in order to reduce over-fitting. To further speed-up the training procedure, stochastic back-propagation with a bunch mode weights update can be used [12], A modified version of the ICSI QuickNet[1] software suite for network training was used.

2.2. Use of NNLM probabilities

In state-of-the-art LVCSR systems, NNLMs are often linearly interpolated with n-gram LMs to obtain both a good coverage of contexts and strong generalisation ability [8, 9], as considered in this paper. Let $P(w_t|h_t)$ denote the interpolated LM probability for word w_t following history h_t. This is given by

$$P(w_t|h_t) \quad = \quad \lambda P_{\mathsf{NG}}(w_t|h_t) + (1-\lambda)\tilde{P}_{\mathsf{NN}}(w_t|h_t) \tag{1}$$

where λ is the interpolation weight assigned to n-gram distribution $P_{\mathsf{NG}}(\cdot)$. As discussed, conventional forms of NNLMs without an OOS output node only model probabilities of more frequent, in-shortlist words $w \in V_{\mathsf{sl}}$. As zero probability mass is reserved for other OOS words in the complete vocabulary, a direct interpolation between NNLMs and full vocabulary based n-gram LMs will also retain such an undue bias to in-shortlist words. To handle this issue, NNLM probabilities $P_{\mathsf{NN}}(\cdot)$ for in-shortlist words can be normalised using n-gram LM statistics as

$$\tilde{P}_{\mathsf{NN}}(w_t|h_t) = \begin{cases} P_{\mathsf{NN}}(w_t|h_t)\alpha_{\mathsf{S}}(h_t) & w_t \in V_{\mathsf{sl}} \\ P_{\mathsf{NG}}(w_t|h_t) & \text{otherwise} \end{cases} \tag{2}$$

$$\alpha_{\mathsf{S}}(h_t) = \sum_{\tilde{w}_t \in V_{\mathsf{sl}}} P_{\mathsf{NG}}(\tilde{w}_t|h_t) \tag{3}$$

so that the same amount of probability mass is assigned to OOS words as n-gram LMs [7, 9].

For LVCSR systems using very large sized n-gram LMs containing, for example, billions of n-gram entries [8], the above full normalisation scheme can be very expensive in decoding time. A more efficient alternative is to use the biased NNLM probabilities directly [9], so-called *znorm*. This is given by

$$\tilde{P}_{\mathsf{NN}}(w_t|h_t) = \begin{cases} P_{\mathsf{NN}}(w_t|h_t) & w_t \in V_{\mathsf{sl}} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Both the full normalization in Eq.(2) and the *znorm* scheme guarantee a sum-to-one constraint for NNLM probabilities. They were also found to give equivalent error rate performance [9].

When using the modified NNLM architecture with an OOS output node presented in section 2.1, an alternative probability normalisation scheme is required. As the underlying NNLM now models all words in the complete vocabulary, the probability mass computed from the additional OOS output layer node must be re-distributed among all OOS words. Again this can be achieved using n-gram LM statistics $P_{\mathsf{NG}}(\cdot)$ as,

$$\tilde{P}_{\mathsf{NN}}(w_t|h_t) = \begin{cases} P_{\mathsf{NN}}(w_t|h_t) & w_t \in V_{\mathsf{sl}} \\ \beta_{\mathsf{S}}(w_t|h_t)P_{\mathsf{NN}}(w_{\mathsf{oos}}|h_t) & \text{otherwise} \end{cases} \tag{5}$$

$$\beta_{\mathsf{S}}(w_t|h_t) = \frac{P_{\mathsf{NG}}(w_t|h_t)}{\sum_{\tilde{w}_t \notin V_{\mathsf{sl}}} P_{\mathsf{NG}}(\tilde{w}_t|h_t)} \tag{6}$$

so that an informative NNLM distribution for OOS words can be used during interpolation with n-gram LMs, rather than back-off to n-gram probabilities as the standard NNLMs of equation (2).

Similar to the full normalization for standard NNLMs in equation (2), the above normalization can also be very expensive for LVCSR tasks. To improve efficiency, it may be assumed that the OOS probability mass assigned by the NNLM and n-gram LM are equal, Thus an approximated normalization is

$$\tilde{P}_{\mathsf{NN}}(w_t|h_t) = \begin{cases} P_{\mathsf{NN}}(w_t|h_t) & w_t \in V_{\mathsf{sl}} \\ P_{\mathsf{NG}}(w_t|h_t) & \text{otherwise} \end{cases} \tag{7}$$

The above no longer guarantees a sum-to-one constraint for the NNLM probabilities. However, it is hoped that the unbiased, more robust NNLM probability estimation for in-shortlist words can still improve recognition performance. Hence, in this paper the full normalisation in equations (5)-(6) are only used for perplexity calculation. The approximated normalization is used in all decoding experiments for error rate evaluation.

3. Neural Network LM Adaptation

In order to improve robustness to varying styles or tasks, unsupervised LM adaptation to a particular domain or task may be used. As discussed in section 2, the use of continuous space

[1] http://www.icsi.berkeley.edu/Speech/qn.html

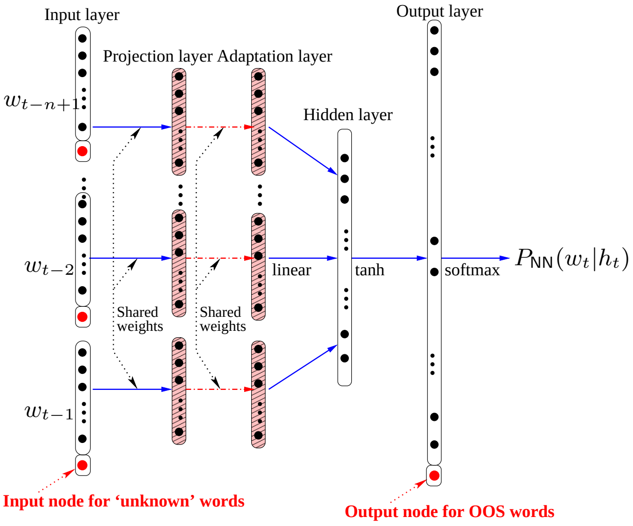

Figure 2: *Architecture of NNLM with adaptation layer.*

representation of words in NNLMs lead to an stronger generalisation ability. This distinct advantage can also be exploited when adapting NNLMs to a given target domain using limited amounts of supervision data.

The NNLM adaptation method considered in this paper involves cascading an additional network layer which is called adaptation layer to an unadapted NNLM. The architecture of an adapted NNLM is illustrated in figure 2. This configuration is conceptually very similar to the Linear Input Network (LIN) based MLP acoustic feature adaptation described in [13].

The precise location to introduce the adaptation layer is determined by two factors. First, as very limited amounts of data, for example, only a few hundred words per audio snippet, are available, a compact structure of the adaptation layer should be used. As discussed in section 2, the number of input and output layer nodes are often very large, for example, tens of thousands of words, for the Arabic speech recognition task considered in this paper. In contrast, much fewer nodes, in the range of a few hundreds, are often used for projection and hidden layers [7, 9]. Second, non-linear activation functions are used in hidden and output layers of standard NNLMs. It is also preferable to retain the same discriminative power and non-linearity during NNLM adaptation. Due to these reasons, the proposed adaptation layer is cascaded between projection and hidden layers. It acts as a linear input transformation to the hidden layer of a task independent NNLM. Using this form of network architecture NNLM probabilities can be adapted to the target domain via a non-linear and discriminative mapping function. During adaptation, only the part of network connecting the adaptation layer and projection layer are updated (shown as dashed lines in Figure 2), while other parameters (shown as solid lines in Figure 2) are fixed.

Another issue with NNLM adaptation is the choice of supervision. The 1-best outputs generated by an unadapted baseline NNLM may be used. In order to improve robustness in adaptation, it is also possible to use confidence measure based soft-target supervision, for example, outputs generated using confusion network (CN) decoding. However, in practice this was found to have minimal impact on performance for the adaptation scheme considered in this paper. Hence, 1-best hypotheses of unadapted baseline NNLMs are used as supervision in all NNLM adaptation experiments of the following section.

4. Experiments

The CU-HTK Arabic LVCSR system was used to evaluate performance of various NNLMs and the adaptation scheme presented in this paper. It was trained on on 1500 hours of broadcast speech data. A total of 1.2G words from 22 text sources were used in LM training. A 350k word list was used. The system uses a multi-pass recognition and system combination framework described in [14]. In the initial "P2" lattice generation stage, an interpolated 4-gram graphemic baseline LM and adapted gender dependent cross-word triphone MPE acoustic models with HLDA projected PLP features were used in decoding. The resulting lattices were then used in a "P3" acoustic re-adaptation and lattice rescoring stage, where two PLP feature based and two PLP+MLP frontend based acoustic models were used [13], before a final CNC combination. All NNLMs were trained using the 15M words of acoustic transcriptions only. The size of NNLM input and output vocabularies are 100k and 20k words respectively. OOS rates are 11.1% for training tokens and 10.6% for development sets. A weight of 0.5 was tuned for a linear interpolation between n-gram and NNLMs and fixed in all experiments. Three GALE Arabic speech development sets: 3 hour dev08, 3 hour eval07, and 3 hour dev09 were used.

4.1. Perplexity results

Perplexity performance of a standard n-gram LM (NG) and several NNLMs: without OOS output node (NN), without OOS output node with *znorm* (NN.znorm), with OOS output node (NN.OOS) and adapted with OOS configurations (NN.OOS.adapt) are shown in the first section of table 1. PP reductions of 22 to 61 points (5.4%-10.3% rel.) were obtained over the n-gram baseline using the conventional NNLM architecture without an OOS output layer node. The unnormalized "NNLM.znorm" model gave slightly lower PP scores than the fully normalised NNLM, due to its bias to in-shortlist words. Further PP reductions of 18 to 32 points were obtained consistently across all three sets, when using the proposed NNLM architecture by explicitly adding an OOS node to the output layer. As discussed in section 2, these PP gains are expected because the use of OOS output layer node more closely represents the nature of complete word sequences found in the training data. It ensures all words were used in NNLM training for more robust weight parameter estimation, compared with discarding all contexts predicting OOS words as in conventional NNLMs. This improved NNLM architecture can also provide an informative distribution for OOS words after appropriately redistributing their probability mass using n-gram statistics as described in section 2.2. Adapted PP performance of this NNLM in both unsupervised and supervised mode are shown in the last two lines table 1. Additional PP reduction of 18 to 21 points were obtained over the unadapted baseline using the proposed network cascading scheme presented in section 3. The adapted PP performance in supervised mode are also shown in the bottom line of the table with a † to serve as a PP lower bound. As expected, some further, but not large, PP reductions between 11 and 21 points were obtained over unsupervised adaptation.

4.2. System evaluation

Now it's interesting to examine whether the PP improvements in table 1 can be transformed into error rate reductions. A first set of experiments were conducted using the P2 lattice rescoring setup with various NNLMs linearly interpolated with the baseline 4-gram LM. A weighted finite state transducer based on-the-

LM	Perplexity		
	eval07	dev08	dev09
NG	495	404	591
NG + NN	467	382	530
NG + NN.znorm	461	377	521
NG + NN.OOS	435	364	479
NG + NN.OOS.adapt	411	346	452
NG + NN.OOS.adapt†	394	325	441

Table 1: PP performance of NNLMs.(† for supervised adaptation)

LM	WER,%		
	eval07	dev08	dev09
NG	15.1	15.6	19.4
NG + NN.znorm	14.7	15.4	18.6
NG + NN.OOS	14.5	15.2	18.5
NG + NN.OOS.adapt	14.5	15.1	18.5
NG + NN.OOS.adapt†	14.1	14.8	18.1

Table 2: P2 WER performance of NNLMs.(† for supervised adaptation)

fly lattice expansion algorithm [15] was used in lattice rescoring. As discussed in section 2.2, the expected impact on error rate is small and high computational cost required, the NNLM probability normalisation schemes given in equations (2) and (5) were not used in these experiments. Other *znorm* and approximated NNLM probabilities of equation (4) and (7) were used.

The first three lines of table 2 show WER performance of the baseline 4-gram back-off LM and two unadapted NNLMs. Consistent with the trends of perplexity reduction observed in table 1, the use of additional OOS output node outperformed the conventional NNLM architecture on all three test sets by 0.1%-0.2% in WER. These results confirm that the modified NNLM architecture gave a better probability estimation for in-shortlist words, as both forms used the normalised in-shortlist NNLM scores of equation (4) and (7) during decoding. The total error rate gains using the NNLM with OOS over the baseline 4-gram LM were 0.4%-0.9% absolute (2.5%–4.6% rel.) across three test sets, all being statistically significant. NNLM adaptation gave a small WER reduction of 0.1% on dev08 and dev09. The performance of supervised NNLM adaptation is also shown in the last line of the table. This system serves as an upper bound for WER gains using the NNLM adaptation method of this paper. It outperformed unsupervised NNLM adaptation by only 0.3%-0.4% absolute. This gap is much smaller than directly retraining the n-gram LM using the test data reference. This confirms NNLMs are less sensitive to the supervision quality than n-gram LMs.

Table 2 shows the performance of the improved and adapted NNLMs at the P2 stage. It's interesting to examine whether the WER gains can be maintained at the P3 stage where 4 re-adapted acoustic models (graphemic/phonetic PLP models G3p/V3p, and graphemic/phonetic PLP+MLP models G3m/V3m) are used to rescoring P2 lattices generated by various LMs of table 2. These WER results are shown in table 3, together with the CNC combined performance in the bottom section of the table. WER gains with the OOS NNLM architecture are maintained for all 4 different branches over n-gram LM baseline. For example, on the best single branch using the "V3m" system, consistent WER reductions of 0.1%-0.6% absolute were obtained over the n-gram LM baseline. Further improvements of 0.1%-0.2% were observed on eval07 and dev09 after NNLM adaptation. Similar trends can also be found in the CNC combined performance. The final gains in CNC combination from the NNLM were 0.2%-0.4% over the n-gram baseline, and further increased to 0.3%-0.6% after NNLM adaptation. These final error reductions to the standard n-gram baseline are significant at the 95% confidence level using the NIST MAPSSWE test.

System	LM	WER,%		
		eval07	dev08	dev09
G3p	NG	14.4	15.0	18.9
	NG + NN.OOS	13.7	14.5	18.4
	NG + NN.OOS.adapt	13.7	14.6	18.2
G3m	NG	13.2	14.4	17.4
	NG + NN.OOS	12.9	13.8	16.8
	NG + NN.OOS.adapt	12.7	13.7	16.7
V3p	NG	13.0	13.9	17.5
	NG + NN.OOS	12.6	13.4	16.9
	NG + NN.OOS.adapt	12.6	13.4	16.7
V3m	NG	12.4	13.5	17.0
	NG + NN.OOS	12.3	13.1	16.4
	NG + NN.OOS.adapt	12.2	13.1	16.2
CNC	NG	11.6	12.5	15.7
	NG + NN.OOS	11.4	12.2	15.3
	NG + NN.OOS.adapt	11.3	12.2	15.1

Table 3: P3 lattice rescoring performance of NNLMs.

5. Conclusion

This paper investigated an improved NNLM architecture and a NNLM adaptation method using a cascaded network. Consistent WER reductions obtained on a state-of-the-art LVCSR task suggest the improved network architecture and proposed adaptation scheme are useful. Future research will focus on more complex forms of modelling of OOS words in NNLMs and improving robustness for NNLM adaptation.

6. References

[1] S.M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer", *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. 35, No. 3. 1987.

[2] P.F. Brown et al., "Class-based n-gram models of natural language", *Computational Linguistic*, Vol. 18, No. 4, 1992, pp. 467–479.

[3] A. Paccanaro and G.E. Hinton, "Extracting distributed representations of concepts and relations from positive and negative propositions", In *Proc. IJCNN2000*.

[4] D. Mrva and P.C. Woodland, "A PLSA-based language model for conversational telephone speech", In *Proc. Interspeech2004*.

[5] T. Brants, "Test data likelihood for PLSA models", *Information Retrieval*, Vol. 8, No. 2, 2005, pp. 181–196.

[6] Y. Bengio et al., "A neural probabilistic language model", *Journal of Machine Learning Research*, Vol. 3, No. 2, 2001, pp. 1137–1155.

[7] H. Schwenk and J-L. Gauvain, "Training neural network language models on very large corpora", In *Proc. HLT/EMNLP2005*.

[8] G. Saon et al., "The IBM 2008 GALE Arabic speech transcription system", In *Proc. ICASSP2010*.

[9] A. Emami and L. Mangu, "Empirical study of neural network language models of Arabic speech recognition", In *Proc. ASRU2007*.

[10] M. Afify et al., "Gaussian mixture language models for speech recognition", In *Proc. ICASSP2007*.

[11] R. Sarikaya et al., "Tied-mixture language modeling", In *Proc. HLT/NAACL2009*.

[12] H. Schwenk, "Continuous space language models", *Computer Speech and Language*, Vol. 21, 2007, pp. 492–518.

[13] J. Park, F. Diehl, M.J.F. Gales, M. Tomalin and P.C. Woodland, "Training and adapting MLP features for Arabic speech recognition", In *Proc. ICASSP2009*.

[14] M. Tomalin, F. Diehl, M.J.F. Gales, J. Park and P.C. Woodland, "Recent improvements to the Cambridge Arabic Speech-to-Text systems", In *Proc. ICASSP2010*.

[15] X. Liu, M.J.F. Gales, J.L. Hieronymus and P.C. Woodland, "Language model combination and adaptation using weighted finite state transducers", in *Proc. ICASSP2010*.