INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Training Context-Dependent DNN Acoustic Models using Probabilistic Sampling

Tamás Grósz[1], *Gábor Gosztolya*[1,2], *László Tóth*[2]

[1]Institute of Informatics, University of Szeged, Hungary
[2]MTA-SZTE Research Group on Artificial Intelligence, Szeged, Hungary
{ groszt, ggabor, tothl } @ inf.u-szeged.hu

Abstract

In current HMM/DNN speech recognition systems, the purpose of the DNN component is to estimate the posterior probabilities of tied triphone states. In most cases the distribution of these states is uneven, meaning that we have a markedly different number of training samples for the various states. This imbalance of the training data is a source of suboptimality for most machine learning algorithms, and DNNs are no exception. A straightforward solution is to re-sample the data, either by upsampling the rarer classes or by dowsampling the more common classes. Here, we experiment with the so-called probabilistic sampling method that applies downsampling and upsampling at the same time. For this, it defines a new class distribution for the training data, which is a linear combination of the original and the uniform class distributions. As an extension to previous studies, we propose a new method to re-estimate the class priors, which is required to remedy the mismatch between the training and the test data distributions introduced by re-sampling. Using probabilistic sampling and the proposed modification we report 5% and 6% relative error rate reductions on the TED-LIUM and on the AMI corpora, respectively.
Index Terms: speech recognition, deep neural networks, probabilistic sampling

1. Introduction

The imbalance in the class distribution poses a significant challenge to most machine learning algorithms [1], and Deep Neural Networks (DNNs) are no exception. It is known that neural networks are inclined to become biased towards classes with more training examples, underestimating the posterior probabilities of the rarer classes [2]. Class imbalance is a typical problem in detection tasks, where usually only a small percentage of the training samples belong to the positive class [3]. The situation is even more difficult when the total amount of training data is already very low in itself.

Here, we examine the effect of class imbalance on the training of DNN acoustic models. At first glance, class imbalance is not an issue in speech recognition, as the frequency of the phones is quite balanced, and we have tremendous amounts of training data compared to some other machine learning tasks. However, we typically use context dependent (CD) phone models, and we increase the number of tied states when the size of the training corpus increases. We will show that the distribution of these CD target labels is far from uniform, meaning that many of the training samples belong to only a few classes, while many of the CD state classes are represented by just a few examples. While one would think that this causes problems only in low-resource scenarios, our experiments will show that the technique we propose may significantly improve the recognition results even in the case of fair-sized corpora.

The problem of class imbalance is typically tackled by applying re-sampling algorithms to the training data. In the simplest approach, the class-balance of the data is achieved by either reducing the number of the examples of the most common classes (*downsampling*) [4] or by presenting the rare examples more frequently (*upsampling*). Here, we utilize a more sophisticated algorithm called probabilistic sampling [5]. Probabilistic sampling offers a solution to apply downsampling and upsampling at the same time by applying a two-step sampling process. For this, we define a new probability distribution over the classes, which determines how frequently the classes are chosen during re-sampling. The first step of the sampling process chooses a class based on this distribution. For the second step, a sample from the training vectors of this class is selected following a uniform distribution. A simple solution to create a probability distribution over the classes is to take the linear combination of the original class distribution and the uniform distribution. This will result in a re-sampling process that has one free parameter, the weight λ of this linear interpolation. With $\lambda = 0$, we retain the original class distribution, while $\lambda = 1$ results in a uniform class sampling.

Tóth and Kocsor applied the probabilistic sampling method to a very small speech recognition task in 2005 in the framework of HMM/ANN hybrids, and they reported improvements in the results [6]. As they worked only with monophone class labels, the main problem they tried to handle by probabilistic sampling was data scarcity. In 2015, Song et al. applied probabilistic sampling in the training of DNN acoustic models with context-dependent targets, and they obtained a significant reduction in the word error rate [7]. However, they performed their experiments on a low-resource task, using a corpus of only 4.5 hours of speech. When discussing re-sampling methods in the framework of speech recognition, we should also mention the in-depth study of García-Moral et al., who applied a simple downsampling approach by discarding examples belonging to the more common classes. Although this made the ANN training process much faster, they experienced a slight drop in the accuracy scores [4]. Lastly, we should mention that in the past few years we successfully used probabilistic sampling in detection-oriented paralinguistic tasks such as detecting the intensity of cognitive and physical load [3].

The classic mathematical formulation of HMM/ANN hybrids states that the neural network outputs estimate the posterior distribution of the training labels, which can be incorporated in the HMM framework after a division by the class priors [8]. When probabilistic sampling is applied with uniform class sampling, Tóth and Kocsor [6] proved that there was no need to divide by the priors, as the network will approximate the class-conditional probabilities within a scaling factor. Unfortunately, neither the authors of [6] nor [7] addressed the

http://dx.doi.org/10.21437/Interspeech.2017-338

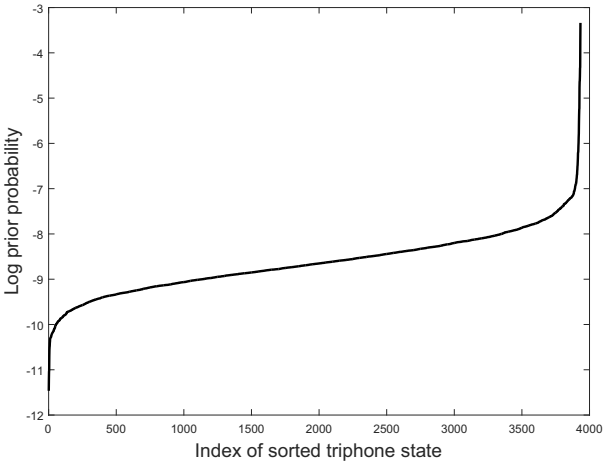

Figure 1: *The distribution of tied CD states on a logarithmic scale in descending order (TED-LIUM corpus, Kaldi recipe)*

problem of intermediate distributions; that is, when the interpolation factor λ is between 0 and 1. García-Moral emphasizes that in such cases the posterior estimates require a proper scaling [4] after re-sampling the training data. To achieve this, here we propose to re-estimate the priors from the re-sampled training data, and divide the DNN outputs by these adjusted priors. Besides examining the effect of scaling by the various estimates of the class priors, we also compare two different strategies for the random selection of the samples within a given class. Our experiments show that with the proposed minor modifications probabilistic sampling can be used to improve the results of training CD DNN acoustic models, even in cases where large amounts of data are available. In the experiments we evaluated our method on the publicly available TED-LIUM corpus (release 1), which contains 118 hours of training data [9], and the public AMI corpus, which has a training set of 100 hours [10]. We report relative word error rate reductions between 5% and 6% on these corpora.

2. Probabilistic sampling

The class distribution of CD state labels is a heavy tailed distribution, meaning that the number of examples for each state differs significantly. Fig. 1 shows the empirical distribution of the CD states on a logarithmic scale for the TED-LIUM corpus (the CD states were obtained using the Kaldi recipe [11]). As can be seen, a subset of the classes is significantly over- and under-represented, which might bias the DNN to favour certain classes and neglect some others. As a result, it outputs imprecise posterior estimates for these classes, which usually leads to a higher word error rate (WER).

One possible way to avoid this is to artificially balance the class distribution by re-sampling the training set. Usually, we have no way of generating additional samples from a rare class, so balancing can be achieved by either reducing the number of examples belonging to the most common classes (downsampling) or by presenting the rare examples more frequently (upsampling).

Probabilistic sampling offers a third option by combining the two previous sampling approaches [5]. It applies a simple two-step sampling scheme; namely, first we select a class, then we pick a training sample belonging to this class. The first step requires us to assign a probability to each class, which determines how frequently it is selected. Here, we will use the following formula to define the sampling probability of the classes:

$$P(c_k) = \lambda \frac{1}{K} + (1 - \lambda) Prior(c_k), \qquad (1)$$

where $Prior(c_k)$ is the prior probability of class c_k, K is the number of classes and $\lambda \in [0, 1]$ is a parameter. For $\lambda = 1$, we get a uniform distribution over the classes; and with $\lambda = 0$ we retain the original class distribution. Using intermediate λ values leads to a linear combination of these two distributions.

2.1. Selecting samples within the classes

Having chosen a class based on Eq. (1), we need to select a sample belonging to that class. During re-sampling our main goal is to modify the class distribution of the training data and leave the distribution of the training examples belonging to the same class unchanged (uniform). The simplest way to do this is to pick a random training vector within the class. However, as we perform only a few iterations through the training data, this strategy could have an undesired side effect that it could change the distribution of the examples within the same class. The reason for this is that for some classes the re-sampling method presents the training vectors to the DNN unevenly, meaning that some examples might not be selected at all during the whole training process. We propose a very simple solution to remedy the problem. First, we define a random ordering of the examples belonging to the given class. Then, during training, we always select the next sample with this ordering. This strategy ensures that the examples of the given class are presented evenly.

2.2. Adjusting the prior probability estimates

The standard practice for HMM/ANN hybrids is to divide the outputs of the DNN acoustic model by the class priors, in order to convert the posterior estimates to likelihood estimates. When applying probabilistic sampling, in theory, the division by the priors is required when $\lambda = 0$ (there is no re-sampling), and there is no need to divide with the priors when $\lambda = 1$ (uniform class sampling). The important theoretical question is what to do in the intermediate cases ($0 < \lambda < 1$). Lacking theoretical results, Tóth and Song performed their evaluations by dividing the posterior estimates by the class priors or by using the neural network outputs directly, and found the optimal λ value experimentally [6, 7]. Here, we argue that the re-sampling of the training database requires us to properly adjust the prior probabilities. The reason is that by balancing the data we create a mismatch between the distribution of the training and the test sets. A simple and intuitive solution for the adjustment is to use the class selection probabilities from Eq. (1) as class prior estimates. This way, we can ensure that the adjusted priors estimate the class distribution of the re-sampled training data. In our experiments we evaluate our models with both the original and the adjusted prior estimates to empirically justify the significance of this adjustment.

3. Experimental Setup

Two large English speech databases were used to train the DNNs, namely the TED-LIUM and AMI corpus. The TED-LIUM corpus [9] is composed of a total of 774 TED talks, containing 118 hours of speech overall: 82 hours of male and 36 hours of female speech. All recordings and their closed captions in this corpus were extracted from the TED website. The

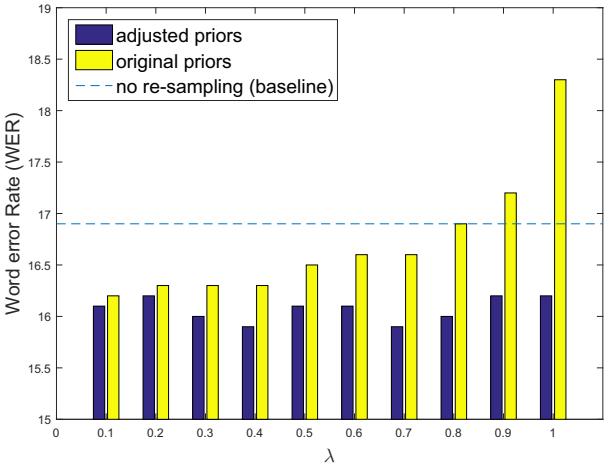

Figure 2: *Word error rates got for the development set of the TED-LIUM corpus using a 3-gram language model and probabilistic sampling.*

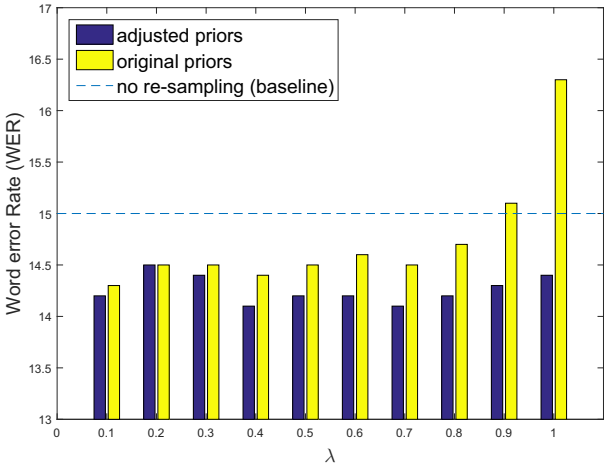

Figure 3: *Word error rates got for the test set of the TED-LIUM corpus using a 3-gram language model and probabilistic sampling.*

training data was automatically transcribed and only the development and test sets were transcribed manually (for more details, see [9]). As training targets we used 3933 CD labels, and the class distribution can be seen in Fig. 1. We evaluated the trained DNN-based acoustic models using a 3-gram and a 4-gram language model as well.

AMI is a multi-modal corpus, which contains recordings of meetings [10]. All participants of the meetings speak in English, but only some of them are native English speakers, which leads to a high degree of variability in speech patterns. Here we used only the audio part of the corpus, specifically the recordings captured with the independent headset microphone (IHM). Following the Kaldi [11] recipe, the DNNs predicted the posterior scores of 3973 CD states, which had a similar class distribution to that of the TED-LIUM corpus.

The acoustic model in our experiments was a DNN with 5 hidden layers, each containing 1000 rectified neurons [12], while we applied the softmax activation function in the output layer. The DNNs were trained using frame aligned labels and no sequence training was applied. The main advantage of Deep Rectifier Networks is that they can be trained without any tedious pre-training (e.g. [13, 14]). As input we used the 40 dimensional fMLLR features extracted by following the Kaldi recipe and the DNNs were trained on 11 neighbouring frames. To train the DNNs we used our own deep learning framework [15], while the decoding and evaluation was performed with Kaldi.

To test the effectiveness of the probabilistic sampling method, we tested λ values between 0.1 and 1.0 with a step size of 0.1. For each training iteration, we re-sampled the same amount of training vectors as that in the original data. All DNN models were evaluated with the division by the original and the adjusted priors to see the effectiveness of the adjustment.

4. Results

First, we compared the two sample selection approaches described in Section 2.1. We found that selecting training vectors within the classes with uniform sampling led to suboptimal models for some rare triphones. In our preliminary experiments we observed that this strategy led to a 1% increase in the frame error rates compared to that for the other selection method, and also resulted in a higher WER. As the selection method that uses a random ordering performed consistently better, we decided to apply it in all our experiments.

Table 1: *Best word error rates got with and without probabilistic sampling and dividing by the original and the adjusted priors.*

LM	Method	Dev WER		Test WER	
		original priors	adjusted priors	original priors	adjusted priors
3-gram	baseline	16.9	–	15.0	–
	$\lambda = 0.4$	16.3	15.9	14.4	14.1
4-gram	baseline	15.2	–	13.7	–
	$\lambda = 0.4$	14.7	14.4	13.0	12.9

4.1. TED

Fig. 2 and 3 shows the results we got with probabilistic sampling on the TED-LIUM corpus. Clearly, dividing the DNN outputs by the original priors gives worse results as λ increases, and we found that small λ values (here 0.1) work best. For small λ values, i.e. when the original distribution remains dominant in the class distribution of the new training data, both prior estimation strategies performed similarly, but as we increase λ above 0.5, the mismatch between the training and test sets caused a significant drop in recognition accuracy (even below the baseline). When we used the adjusted priors, the models became more robust and we got better results than the baseline for all λ values. The best result on the development set was attained using the adjusted priors and $\lambda = 0.4$; this network achieved a 14.1% WER on the test set, which means a 6% relative error reduction compared to the baseline.

Table 1 summarizes the best results on the TED-LIUM database. As can be seen, probabilistic sampling always yielded better results and with the prior adjustment we managed to improve the performance further. Using the 4-gram language model produced similar results to those achieved with the 3-gram model. The optimal value for the re-sampling parameter was 0.4 similar to when the 3-gram language model was used.

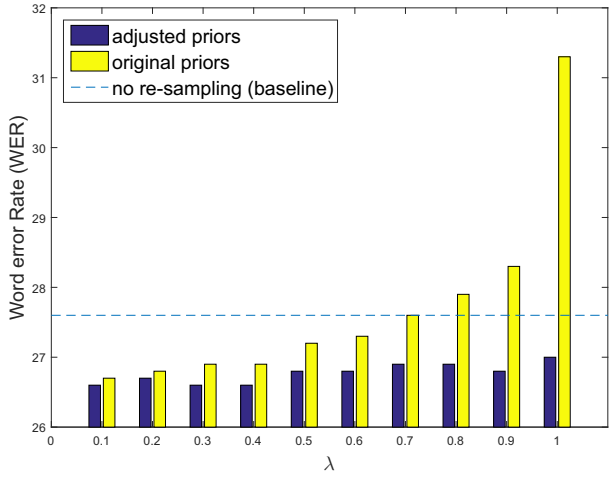

Figure 4: *Word error rates got for the development set of the AMI corpus using probabilistic sampling.*

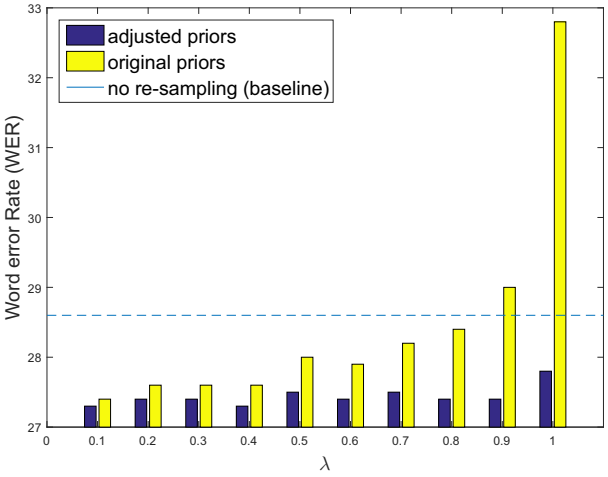

Figure 5: *Word error rates got for the test set of the AMI corpus using probabilistic sampling.*

4.2. AMI

On the AMI corpus the results follow a similiar trend; the best results were achieved with the adjusted priors, and the division by the original priors resulted in a declining recognition accuracy for increasing λ. All DNNs trained with $\lambda \leq 0.7$ performed better than the baseline model both on the development and the test sets. The optimal value of λ was 0.1 when we divided by the original prior (26.7% WER on the development set and 27.4% on the test) and 0.1 or 0.4 when the adjusted priors were used. Both DNN achieved a WER of 26.6% on the development and 27.3% on the test set. On the test set the best WER was 27.3%, which is significantly better than the baseline (28.6%), giving approximately 5% relative error reduction. We would like to note, that using uniform re-sampling with the original priors resulted in recognition results far below the baseline.

4.3. Discussion

To get an insight on why probabilistic sampling helps, we performed an analysis to see how the accuracy of CD state classification varies as a function of state frequency. Fig. 6 shows the average frame level accuracy scores of the sorted CD states, comparing the baseline method with the best model trained with re-sampling. The first thing to notice is that probabilistic sampling significantly improves the accuracy scores of the rare states (Index $\leq$ 1000), and even the common states are recognized more frequently. The downside of this improvement is the lower accuracy of those classes that have the most training data. Interestingly, the accuracy of classes having an average amount of training data (middle section in the figure) also increased with probabilistic sampling; the reason behind this could be that they were less likely confused with the more frequent states.

As we saw, dividing the DNN outputs by the adjusted priors stabilized the results: for almost all λ values we got similar WE scores. If the original priors are used then a declining trend is present as we move farther from the original distribution. The stability of this adjustment could be explained by the fact that it reduces the mismatch between the training and test data introduced by the re-sampling.

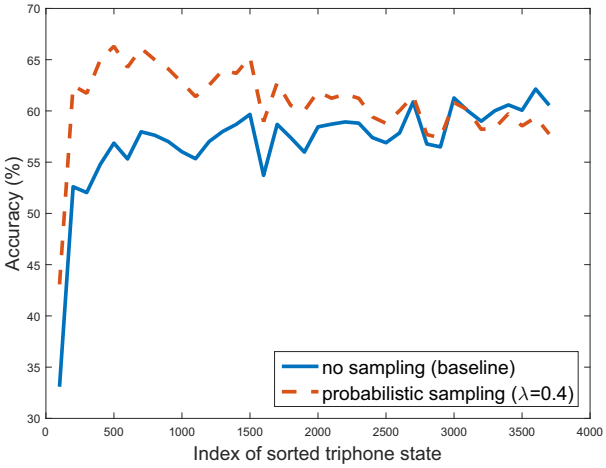

Figure 6: *Averaged accuracies of sorted CD states on the TED-LIUM development set with and without re-sampling.*

5. Conclusions

Here, we showed that CD DNN training can be improved by probabilistic sampling. We also proposed a new method to re-estimate the class priors when using this sampling algorithm. Our results showed that this re-estimation is essential to remedy the mismatch between the training and the test data distributions introduced by the re-sampling step. These adjusted priors made the re-sampling method more robust, and the recognition results varied only slightly as the class distribution was shifted towards uniform distribution. Our experiments showed that by using this modification, the recognition results improved significantly (between 5% and 6% relative error reduction) on two fair-sized corpora (TED-LIUM and AMI).

6. Acknowledgements

Tamás Grósz was supported by the ÚNKP-16-3 New National Excellence Programme of the Ministry of Human Capacities.

7. References

[1] G. M. Weiss and F. Provost, "The effect of class distribution on classifier learning: an empirical study," *Technical Report, Rutgers Univ.*, 2001.

[2] K. Andric and D. Kalpic, "The effect of class distribution on classification algorithms in credit risk assessment," in *Proceedings of MIPRO*. IEEE, 2016, pp. 1241–1247.

[3] G. Gosztolya, T. Grósz, R. Busa-Fekete, and L. Tóth, "Detecting the intensity of cognitive and physical load using AdaBoost and Deep Rectifier Neural Networks," in *Proceedings of Interspeech*, Singapore, Sep 2014, pp. 452–456.

[4] A. I. García-Moral, R. Solera-Ureña, C. Peláez-Moreno, and F. Díaz-de-María, "Data balancing for efficient training of hybrid ANN/HMM automatic speech recognition systems," *IEEE Trans. Audio, Speech & Language Processing*, vol. 19, no. 3, pp. 468–481, 2011.

[5] S. Lawrence, I. Burns, A. Back, A. Tsoi, and C. Giles, "Chapter 14: Neural network classification and prior class probabilities," in *Neural Networks: Tricks of the Trade*. Springer, 1998, pp. 299–313.

[6] L. Tóth and A. Kocsor, "Training HMM/ANN hybrid speech recognizers by probabilistic sampling," in *Proceedings of ICANN*, 2005, pp. 597–603.

[7] M. Song, Q. Zhang, J. Pan, and Y. Yan, "Improving HMM/DNN in asr of under-resourced languages using probabilistic sampling," in *Proceedings of ChinaSIP*, 2015.

[8] H. Bourlard and N. Morgan, *Connectionist Speech Recognition – A Hybrid Approach*. Kluwer Academic, 1994.

[9] A. Rousseau, P. Delglise, and Y. Estve, "TED-LIUM: an automatic speech recognition dedicated corpus," in *Proceedings of LREC*, 2012, pp. 125–129.

[10] J. Carletta, "Unleashing the killer corpus: experiences in creating the multi-everything AMI Meeting Corpus," *Language Resources and Evaluation*, vol. 41, no. 2, pp. 181–190, 2007.

[11] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *Proceedings of ASRU*. IEEE Signal Processing Society, 2011.

[12] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier networks," in *Proceedings of AISTATS*, 2011, pp. 315–323.

[13] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. ASRU*, 2011, pp. 24–29.

[14] N. Jaitly, P. Nguyen, A. Senior, and V. Vanhoucke, "Application of pretrained deep neural networks to large vocabulary conversational speech recognition," Dept. Comp. Sci., University of Toronto, Tech. Rep., 2012.

[15] T. Grósz and L. Tóth, "A comparison of Deep Neural Network training methods for Large Vocabulary Speech Recognition," in *Proceedings of TSD*, Pilsen, Czech Republic, 2013, pp. 36–43.