Automatic Parameter Tuning and Extended Training Material: Recent Advances in the Fraunhofer Speech Recognition System

Jochen Schwenninger, Daniel Stein, Michael Stadtschnitzer

Fraunhofer IAIS
Schloss Birlinghoven
53757 Sankt Augustin, Germany
name.surname@iais.fraunhofer.de

Abstract: Building the acoustic and language models on a larger amount of training data is a well-known method for robustifying automatic speech recognition approaches. The adaption of the decoder settings afterwards, however, is often only marginally addressed (e.g. being manually set or using default values provided by a toolkit). Without proper adaption, these settings are most often sub-optimal and lead to degraded performance without unlocking the full potential of the speech recognizer. Ideally, the decoder settings should be optimized after each modification of the language model and/or the acoustic model of the speech recognition system, a task that is typically too tedious for manual work. In this paper, we employ an automatic optimization technique on the Fraunhofer IAIS speech recognition setup as a subsequent step to training data increase. We will present the improvements of the expanded training data for the acoustic models and the optimization of the decoder settings on the German Difficult Speech Corpus.

Index Terms: simultaneous perturbation stochastic approximation, free decoding parameters, word error rate optimization

1 Introduction

The optimization of the acoustic model and accordingly the language model in large vocabulary continuous speech recognition (LVCSR) are well-established tasks. A common way to achieve lower word error rates (WER) is, amongst others, the reduction of perplexity of the language model on a withheld development set (cf. [KP02]). The speech decoding process, however, requires a large number of parameters to be set and adapted to the given task. While some parameters directly weight the models, others affect the size of the search space, where it is even harder to foresee the effect on the hypothesis quality.

Often, speech decoder parameters are either set empirically using cross-validation on a test set, which is a rather tedious task, or the default values of toolkits are retained. However, to guarantee optimality, they should be adapted to new domains, whenever the training material changes or when speech recognizer system changes occur.

In this paper, we significantly increase the size of the training data of the LVCSR system described in [SSE08] and employ Simultaneous Perturbation Stochastic Approximation (SPSA) [Spa92] for the optimization of the free decoding parameters. We show that both approaches, individually and together as well, lead to increased performance in terms of WER.

We offer our results on the German Difficult Speech Corpus (DiSCo) [BSB+10] as well as on data from the LinkedTV[1] project provided by the Rundfunk Berlin Brandenburg (RBB).

2 Related Work

Speech decoders typically provide wide ranges and default values, which often appear to be arbitrarily set, for their parameters (e.g., in HTK [YEG+06], Julius [LKS01] or Kaldi [PGB+11]). The Sphinx [WLK+04] Wiki [2] offers detailed ways on how to improve the hypothesis quality or decoding speed, but once more these methods have to be adapted manually to the task, usually by some variant of grid search.

Recently, several groups approached the task of optimizing these parameters automatically. While some optimize the WER or similar metrics directly (e.g. Korosi and Kacur using evolutional algorithms [KK07], or Mak and To employing large-margin iterative linear programming [MK09]), other also take the decoding speed in the form of real-time factor (RTF) into account. El Hannani and Hain [EHH10] trace the curve of all possible WER for a given RTF and therefore are able to use the best configuration for any speed constraints. Bulyko [Bul10] instead uses an approach very similar to the Kiefer-Wolfowitz finite-difference stochastic approximation (FDSA) [KW52], where the influence of every parameter on WER and RTF is evaluated seperately for a given configuration. This requires sampling the decoding performance $2kp$ times, with k being the number of iterations and p the number of free parameters.

In the field of machine translation (MT), the parameters of recent decoders (e.g., [KHB+07, VSHN12]) are typically optimized either with the Downhill Simplex Method [NM65] or with Och's Minimum Error Rate Training [Och03]. SPSA has been employed for MT as well and has been shown to converge faster than downhill simplex, while maintaining comparable hypothesis quality results [LB06].

SPSA has already been applied to various tasks other than natural language processing, such as statistical simulations, traffic control, as well as signal and image processing [Spa98b].

[1] `http://www.linkedtv.eu`

[2] `http://cmusphinx.sourceforge.net/wiki/sphinx4:largevocabularyperformanceoptimization`, accessed: 30.1.2013

3 Simultaneous Perturbation Stochastic Approximation

For the optimization of a p-tuple of free parameters θ, we employ the SPSA algorithm [Spa92], which minimized a loss function $L(\cdot)$ as follows:

Let $\hat{\theta}_k$ denote the estimate for θ in the k-th iteration. Then, for a gain sequence denoted as a_k, and an estimate of the gradient of the loss function at a certain position denoted as $\hat{g}_k(\cdot)$, the algorithm has the form

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \hat{g}\left(\hat{\theta}_k\right) \tag{1}$$

In order to estimate $\hat{g}_k(\cdot)$, we perturbate each $\hat{\theta}_k$ with a vector of length p containing mutually independent, zero-mean random variables Δ_k, multiplied by a positive scalar c_k, to obtain two new parameter tuples:

$$\hat{\theta}_k^+ = \hat{\theta}_k + c_k \Delta_k \tag{2}$$

$$\hat{\theta}_k^- = \hat{\theta}_k - c_k \Delta_k \tag{3}$$

For a loss function $L(\cdot)$, we then estimate $\hat{g}\left(\hat{\theta}_k\right)$ as:

$$\hat{g}\left(\hat{\theta}_k\right) = \frac{L\left(\hat{\theta}_k^+\right) - L\left(\hat{\theta}_k^-\right)}{2c_k} \begin{bmatrix} \Delta_{k1}^{-1} \\ \vdots \\ \Delta_{kp}^{-1} \end{bmatrix} \tag{4}$$

We follow the implementation suggestions in [Spa98a] and use a ± 1 Bernoulli distribution for Δ_k, and further set:

$$a_k = \frac{a}{(A+k+1)^\alpha} \quad \text{with } a = 2,\ A = 8,\ \alpha = 0.602 \tag{5}$$

$$c_k = \frac{c}{(k+1)^\gamma} \quad \text{with } c = 0.25,\ \gamma = 0.101 \tag{6}$$

Using these gain sequences a_k and c_k, SPSA normally converges in a similar number of iterations as the classical Kiefer-Wolfowitz FDSA, but requires p times fewer measurements of the loss function, as the decoding performance has to be evaluated only 2 times in each iteration. α and γ are set to the lowest values satisfying the theoretical conditions for convergence, leading to larger step sizes in the iteration process and therefore faster performance.

Since the measurements of $L(\theta)$ do not contain noise, we choose c to be a small positive number as proposed in [Spa98a]. Likewise, we set $A = 8$ to approximately 10% of the expected iterations and finally $a = 2$ so that the steps in the update have a reasonable size for the first iterations. During our experiments, we did not experience a high sensitivity of SPSA for any of its hyperparameters, only the required number of iterations to reach a stable result changes.

label	description
⟨int⟩	If an utterance contains clearly audible background noises, it is tagged with ⟨int⟩. The type and volume of noise was not differentiated in this annotation sequence.
⟨spk⟩	This tag denotes various speaker noises, such as breathing, throat clearing or coughing.
⟨fil⟩	All kinds of hesitations are labeled with this tag.
⟨spk_change⟩	If the speaker changes during the utterance, ⟨spk_change⟩ is inserted at the corresponding position. Using this annotation, speaker turns can be inferred and then used for speaker-adaptive training schemes in later steps.
⟨overlap⟩	If more than one speaker is talking at the same time, the utterance is marked with this tag.
⟨foreign⟩	One or more foreign words, sometimes proper names but most of the time original material with a voice-over.
⟨mispron⟩WORD⟨mispron⟩	Clearly mispronounced words are enclosed in this tag.
⟨reject⟩	If a whole utterance can not be transcribed, it is marked with this tag.
**	If one or more words are unintelligible (e.g. due to background noise), they are transcribed with **.
=	Word fragments are transcribed and end with =, marking them as incomplete.

Table 1: All labels used for the annotation of the new training corpus

4 Setup and Training/Testing Material

The basic architecture and training workflow of our ASR system has been described in [SSE08].

Recently we collected and manually transcribed a huge new training corpus of broadcast video material, with a volume of approx. 400 h and containing roughly 225 h of clean speech. This corpus allows us to train much more accurate acoustic models, which will be used for ASR in both the LinkedTV project as well as for the AXES project. As the effort for acquiring new training data is still ongoing, the final size of the corpus will eventually reach 900 h, making this one of the largest corpora of German TV and radio broadcast material known to us.

The new corpus is segmented into utterances with a mean duration of 10 seconds and is transcribed manually on word level. The recorded data covers a broad selection of news, interviews, talk shows and documentaries, both from television and radio content across several stations. In addition to the verbatim transcript, the tags in table 1 were used to further describe the recordings.

The audio is recorded and stored in 16-bit PCM waveform files, with 16 kHz sampling

frequency and a single mono channel. Utterances containing one or more annotations with ⟨int⟩, ⟨overlap⟩, ⟨foreign⟩, ⟨mispron⟩, ⟨reject⟩, ⟨**⟩ or ⟨=⟩ were excluded for the following experiments from the clean speech training data.

In order to use the additional data, the training setup using HTK [YEG+06] was heavily modified. The initial configuration with approx. 105 h training data used tied crossword triphones using 32 mixtures per state in 3-state HMMs, leading to 240k Gaussians in 7.5k states. Since simply increasing the number of mixtures led to unsatisfactory results, the state tying configuration was adapted. Using fewer states and therefore more training data per state solved the problem of overfitting infrequent states and leads to significant improvements in WER, especially in acoustically challenging situations like spontaneous speech. The new baseline model uses 575k Gaussians for 6k states and generalizes well to previously unseen material, as can be seen from Table 5, still using the old decoding parameters. In order to use this new model optimally, we revisit setting these parameters using SPSA in the following section.

For developing and optimizing the free parameters, we use a corpus from German broadcast shows, which contains a mix of planned (i.e., read news) and spontaneous (i.e., interviews) speech, for a total of 2,348 utterances (33,744 words).

For evaluation, we make use of clean speech segments of the DiSCo corpus as described in [BSB+10], and use "planned clean speech" (0:55h, 1,364 utterances) as well as "spontaneous clean speech" (1:55h, 2,861 utterances).

Additionally, we test the decoding performance on content from the RBB provided to the LinkedTV project, again separated into a planned set (1:08h, 787 utterances) and a spontaneous set (0:44h, 596 utterances). While during the annotation of the DiSCo corpus a strong emphasis was put on selecting only segments with clean acoustics (i.e. no background music, high quality) and without dialectal speech, this was not feasible for the newer RBB corpus. Therefore, especially the spontaneous set contains utterances from street interviews, partly with strong Berlin dialect (e.g. "dit Jahr war ja nich berauschend bei Hertha wa" instead of "das Jahr war nicht sehr berauschend bei Hertha").

5 Parameter Tuning with SPSA

For optimization, we chose to optimize both parameters that primarily affect the search space as well as those that affect the internal weighting/penalty of the underlying models. Table 3 lists the Julius parameters, the ranges that we allow for as well as the starting values for optimization. Internally, we map these ranges to $[-15 \cdots +15]$ for the SPSA iterations. If the parameters are integers, we store them as floats internally but truncate them for each evaluation of the loss function.

We optimized the parameters on the WER, i.e., the number of substitutions, insertions and deletion errors divided by the reference length, using it directly as the loss function in SPSA.

The results on the development set are shown in Figure 1. In total, the hypothesis qual-

name	start	min	max
(2) LM weight	10.0	0.0	20.0
(2) ins. penalty	-7.0/10.0	-20.0	20.0
(2) beam width	250/1,500	700/20	3,000/1,000
score envelope	80.0	50.0	150.0
stack size	10,000	500	20,000
#expanded hyp.	20,000	2,000	20,000
#sentence hyp.	10	5	1,000

Table 2: Free parameters of the decoding process. Some parameters are given individually to the 1^{st} pass or 2^{nd} pass of the Julius decoder, and are marked with (2). Continuous parameters are marked by a trailing ".0".

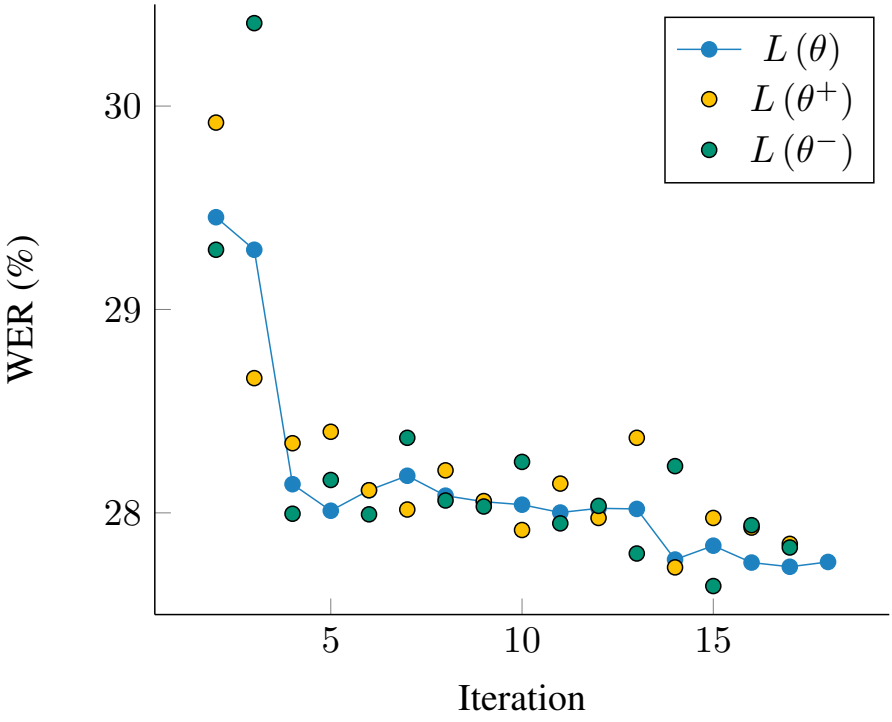
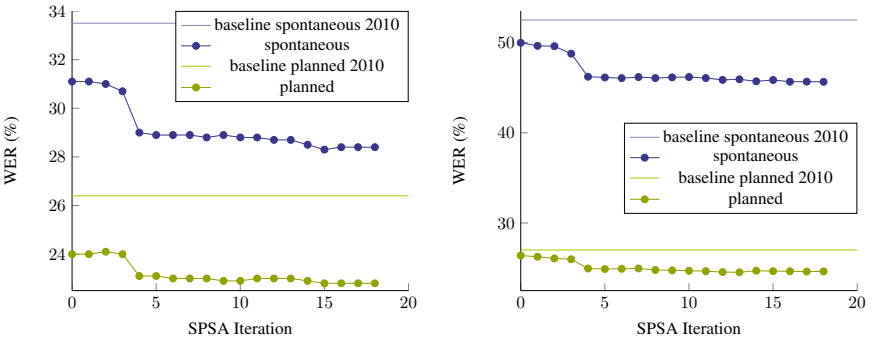


Figure 1: Example run of SPSA and its word error rate progression on the development corpus. Please note the offset in y-direction.

parameter set	WER dev	WER DiSCo planned	WER DiSCo spontaneous	WER LinkedTV planned	WER LinkedTV spontaneous
baseline cf. [BSB+10]	30.2	26.4	33.5	27.0	52.5
improved baseline	29.6	24.0	31.1	26.4	50.0
SPSA 1st run	27.7	22.8	28.4	24.6	45.7
SPSA 2nd run	27.7	22.6	28.4	24.5	45.6

Table 3: WER results on all corpora, for the SPSA iterations and their respective loss functions. Each optimization on a given loss function has been executed two times from scratch with 18 iterations to check for convergence.



(a) WER progression on DiSCo corpora.

(b) WER progression on LinkedTV corpora.

Figure 2: WER results on planned and spontaneous data, showing the first run of SPSA with 18 iterations. Iteration 0 denotes the improved acoustic baseline. Please note the different scaling and offset in y-direction for the two graphs.

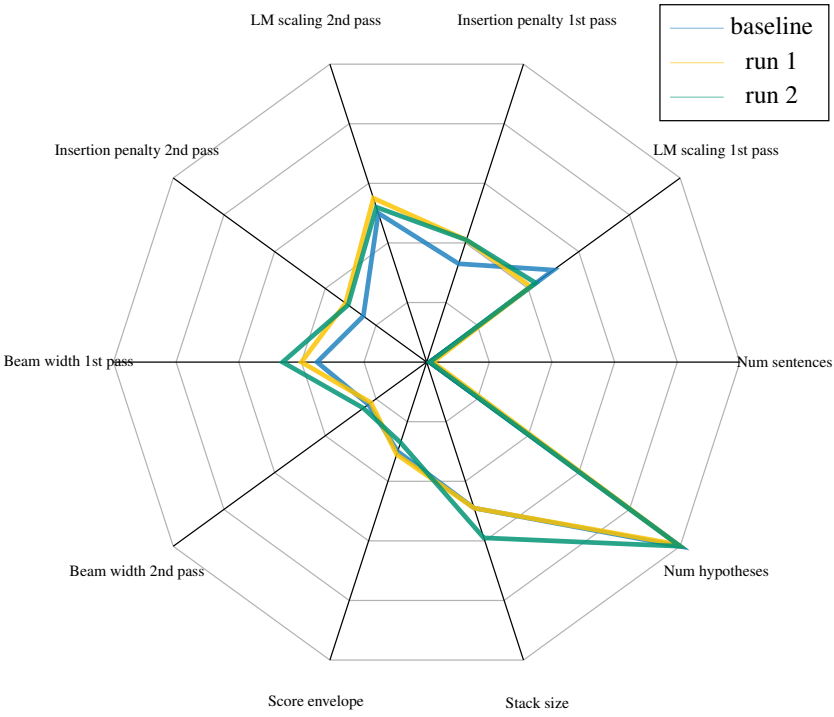


Figure 3: Radar plot of the resulting configurations, with -15.0 mapped to the center and +15.0 to the outmost circle

ity improved by 1.9% WER absolute (6.4% rel.) during 18 iterations. In a second run (s. Table 5), the improvement was similar and converged after 10 iteration runs already. The results on the test sets are presented in Figure 2. It can be seen that the optimization generalizes nicely on all 4 test corpora: 1.2% WER absolute improvement on the planned speech task, and 2.7% WER absolute improvement on the spontaneous speech task for the DiSCo datasets (s. Fig.2(a)), over a strong baseline surpassing the results given in the original corpus paper [BSB+10]. The same is true for the LinkedTV data, with even higher improvements of 1.9% and 4.4% WER respectively. While the resulting error rates for the two "planned" corpora are comparable, the LinkedTV spontaneous dataset performs a lot worse. This can likely be explained by the fact that LinkedTV spontaneous contains partly dialectal speech, while DiSCo spontaneous is purely standard German.

The spider graph in Figure 3 depicts the parameters (normalized to ± 15.0) of the baseline (red) and the optimized values after two SPSA runs (blue and cyan, respectively). It can be seen that both runs lead to similar configurations, with the largest changes in the insertion penalties and the scaling of the language model in the 1^{st} pass. The number of hypotheses produced is always at the highest value, so that it seems to make sense to increase it whenever there are no memory concerns.

6 Conclusion

In this paper, we presented an improved acoustic model for German ASR and have shown that SPSA is an efficient means to optimize free parameters of an ASR decoder. Using the WER as loss function directly, the optimization converges after less than 20 iterations, leading to an improvement of about 4% absolute above the original baseline and about 2% absolute above the new acoustic models.

The resulting settings generalize well to different datasets, i.e., both planned and spontaneous speech. In future experiments, we plan to investigate whether the real-time factor of the decoder can be thus optimized as well. Also, we want to extend SPSA with an estimate of the second derivative in order to adjust the step size for each parameter given its estimated influence on the overall loss function. Additionally we want to investigate how SPSA reacts to non-clean speech material as development data, since parameters like insertion penalty might be used as an effective means to set noise thresholds.

7 Acknowledgements

This work has been partly funded by the European Community's Seventh Framework Programme (FP7-ICT) under grant agreement n° 287911 LinkedTV and n° 269980 AXES.

References

[BSB+10] Doris Baum, Daniel Schneider, Rolf Bardeli, Jochen Schwenninger, Barbara Samlowski, Thomas Winkler, and Joachim Köhler. DiSCo — A German Evaluation Corpus for Challenging Problems in the Broadcast Domain. In *Proc. Seventh conference on International Language Resources and Evaluation (LREC)*, Valletta, Malta, may 2010.

[Bul10] Ivan Bulyko. Speech recognizer optimization under speed constraints. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[EHH10] Asmaa El Hannani and Thomas Hain. Automatic optimization of speech decoder parameters. *Signal Processing Letters, IEEE*, 17(1):95–98, 2010.

[KHB+07] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Annual meeting-association for computational linguistics*, volume 45, page 2, 2007.

[KK07] Juraj Kacur and Jan Korosi. An Accuracy Optimization of a Dialog Asr System Utilizing Evolutional Strategies. pages 180–184, Istanbul, 2007.

[KP02] Dietrich Klakow and Jochen Peters. Testing the correlation of word error rate and perplexity. *Speech Communication*, 38(1–2):19–28, 2002.

[KW52] J. Kiefer and J. Wolfowitz. Stochastic Estimation of a Regesssion Function. *Ann. Math. Stat.*, (23):462–466, 1952.

[LB06] P. Lambert and R.E. Banchs. Tuning machine translation parameters with SPSA. In *Proc. IWSLT*, pages 190–196, 2006.

[LKS01] A. Lee, T. Kawahara, and K. Shikano. Julius – an Open Source Real-Time Large Vocabulary Recognition Engine. In *Proceedings of Eurospeech*, pages 1691–1694, Aalborg, Denmark, 2001.

[MK09] Brian Mak and Tom Ko. Automatic estimation of decoding parameters using large-margin iterative linear programming. In *Proc. of Interspeech*, pages 1219–1222, 2009.

[NM65] J.A. Nelder and R. Mead. The Downhill Simplex Method. *Computer Journal*, 7:308, 1965.

[Och03] F.J. Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics, 2003.

[PGB+11] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. The Kaldi Speech Recognition Toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, December 2011. IEEE Catalog No.: CFP11SRW-USB.

[Spa92] James C. Spall. Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation. *IEEE Transactions on Automatic Control*, 37:3, March 1992.

[Spa98a] James C. Spall. Implementation of the Simultaneous Perturbation Algorithm for Stochastic Optimization. *IEEE Transactions on Aerospace and Electronic Systems*, 34:3, July 1998.

[Spa98b] J.C. Spall. An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins APL Technical Digest*, 19(4):482–492, 1998.

[SSE08] D. Schneider, J. Schon, and S. Eickeler. Towards Large Scale Vocabulary Independent Spoken Term Detection: Advances in the Fraunhofer IAIS Audiomining System. In *Proc. Association for Computing Machinery's Special Interest Group Information Retrieval (ACM SIGIR)*, Singapore, 2008.

[VSHN12] David Vilar, Daniel Stein, Matthias Huck, and Hermann Ney. Jane: an advanced freely available hierarchical machine translation toolkit. *Machine Translation*, 26(3):197–216, September 2012.

[WLK+04] Willie Walker, Paul Lamere, Philip Kwok, Bhiksha Raj, Rita Singh, Evandro Gouvea, Peter Wolf, and Joe Woelfel. Sphinx-4: A Flexible Open Source Framework for Speech Recognition. Technical report, Sun Microsystems Inc., 2004.

[YEG+06] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland. *The HTK Book, version 3.4*. Cambridge University Engineering Department, Cambridge, UK, 2006.