REDUCTION OF ACOUSTIC MODEL TRAINING TIME AND REQUIRED DATA PASSES VIA STOCHASTIC APPROACHES TO MAXIMUM LIKELIHOOD AND DISCRIMINATIVE TRAINING

Petr Novák, Roman Otec

IBM Czech Republic
V Parku 4, 148 00 Praha, Czech Republic
petr.novak3@cz.ibm.com
roman.otec@cz.ibm.com

Antonio Lee, Vaibhava Goel

IBM Thomas J. Watson Research Center
Yorktown Heights, NY 10598, USA
arl@us.ibm.com
vgoel@us.ibm.com

ABSTRACT

The recent boom in use of speech recognition technology has made the access to potentially large amounts of training data easier. This, however, also constitutes a challenge in processing such large, continuously growing amount of information. Here we present a stochastic modification of traditional iterative training approach which leads to the same or even better accuracy of acoustic models and reduces the cost of processing large data sets. The algorithm relies on model updates from statistics collected on randomly selected subsets of training data. The approach is demonstrated on maximum likelihood (ML) training and on discriminative training (DT) with minimum phone error (MPE) objective function both in the feature and the model space. Based on our experiments on 30 thousand hours of mobile data, the number of data passes can be reduced to 1/5 of the original for ML training and to 1/10 for model space DT training.

Index Terms— Acoustic modeling, Speech recognition, Stochastic training, Discriminative training

1. INTRODUCTION

Building acoustic models from large data sets has been shown to benefit the accuracy of speech recognition systems. The recent boom in use of speech recognition technology has made the access to very large quantities (several thousands of hours of audio) of training data easier, but also constitutes a challenge in processing large and continuously growing amount of information. While the recent advances in using Deep Neural Networks (DNNs) [1] have taken some of the spotlight away from training traditional Gaussian Mixture Models (GMM) for Hidden Markov Model (HMM) states, the speed of training such systems still has an important role in building speech systems. For example, in one commonly practiced acoustic modeling approach, DNNs are trained to generate so-called bottleneck features which are then used in GMM HMM training [1]. Therefore, any advances in speeding up GMM training will still have a significant impact in building speech systems.

A traditional training procedure for a GMM-based acoustic model encompasses finding optimal parameters of the GMM for each HMM state. This can be done in several steps. The reference training used in this paper would first set Maximum Likelihood (ML) as a training criterion and iteratively look for an optimal solution for each of the HMM states [2]. The training procedure continues with discriminative training (DT) in the feature space followed by DT in the model domain. In both cases a Minimum Phone Error (MPE) objective function is used as a training criterion for larger training data sets [3]. Both the ML and the DT procedures require multiple passes through the entire training data set, which takes several days to complete.

As the amount of available training data has grown dramatically in the recent years, the processing times for the traditional training approach have become prohibitively large. The need has shifted from developing algorithms capable of training acoustic models from a small available training set to exploiting large amounts of available data.

In this paper we present a stochastic modification of traditional iterative training approach, which leads to the same or sometimes even better accuracy of acoustic models while reducing the cost of processing large data sets by requiring fewer passes through the training data. Our goal is to optimize the model parameters just on a subset of available data while making the most of existing state-of-the-art training algorithms. Therefore the approach is demonstrated on ML training and on DT with MPE objective function both in the feature and the model space.

1.1. Relationship to Prior Work

Variants of expectation-maximization (EM) algorithm such as incremental EM [4] have been used to achieve faster convergence of ML training. Similar efforts have been done in DT, however mainly with Minimum Classification Error as an ob-

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

jective function, not MPE we refer to. The work focuses on better generalization of models [5, 6, 7] potentially allowing training on a subset of training data or on faster convergence of the training algorithms employing various online and batch probabilistic techniques [8, 9, 10].

2. ALGORITHMS

2.1. Maximum Likelihood (ML) Training

The goal of the Maximum Likelihood Training is to find the model parameters which maximize the acoustic data likelihood given the reference word strings [2]. Baum-Welch re-estimation algorithm uses Estimation-Maximization (EM) algorithm to find the optimum of an auxiliary function w.r.t. model parameters. Note that the convergence points of the EM procedure correspond to local maxima of the likelihood function, not the global maximum [4]. With the stochastic approach we hope to steer along the likelihood surface so that frequent updates of parameters using random subsets of training data will make the algorithm converge faster with respect to the number of full data passes (FDP). Additionally, we hope to make the algorithm more robust against getting stuck in a local optimum with a consequent gain in model accuracy.

The BW algorithm runs two steps per EM iteration. First, statistics needed for parameter estimation are collected on the provided data set. Next, model parameters are updated [2]. The traditional implementation uses the entire training set to collect the statistics and the model parameters would be updated from statistics collected in the last iteration only [2].

2.2. Feature and Model Space Discriminative Training (DT)

The discriminative training investigated here uses MPE criteria as an objective function both in feature space [3] and model space [11]. The optimization is done with the Extended Baum-Welch re-estimation where two sets of statistics are accumulated [12]; the so-called numerator statistics using the reference word strings and denominator statistics from competing hypotheses.

2.3. Stochastic Modifications of ML and DT Training

Our stochastic training approach follows a simple modification of the traditional training method: rather than gathering statistics on entire training data set and then carrying out model parameter update, we gather statistics on a randomly selected subset of the training data and update model parameters. Our procedure is parametrized by using the following training criteria: (1) how much data was selected into a training subset, (2) whether single or multiple iterations should be carried out on selected subset before moving on to next subset, (3) what method was used for data selection, (4) whether statistics collected in previous iterations should be used for model parameters update, and if yes, then (5) what kind of smoothing if any should be applied.

Pseudo-code for traditional approach to the training is shown in Algorithm 1, while a stochastic variant reflecting the above set parametrization is illustrated in Algorithm 2.

initialize λ
$\{w_1, w_2, \ldots, w_{bNum}\} \leftarrow \{1/bNum, \ldots, 1/bNum\}$

for $i \leftarrow 1, 2, \ldots, tradIter$ **do**
  $\{sts_1, sts_2, \ldots, sts_{bNum}\} \leftarrow \text{ACCU}(tradData)$
  $sts \leftarrow \text{MERGE}(sts_1, sts_2, \ldots, sts_{bNum}, w_1, w_2, \ldots, w_{bNum})$
  $\lambda \leftarrow \text{UPDATE}(\lambda, sts)$
end for

Algorithm 1: Traditional Training. λ represents model parameters, $tradIter$ is the number of traditional training iterations for given task, $tradData$ is the total amount of training data, $bNum$ is the number of unique batches (i.e. training data subsets) to be used in the training.

initialize λ
$stochIter \leftarrow tradIter * tradData/RBS$

for $n \leftarrow 1, 2, \ldots, bNum$ **do** ▷ Batch generation
  $b_n \leftarrow \text{SAMPLE}(tradData, RBS, level)$
end for

for $i \leftarrow 1, 2, \ldots, stochIter$ **do** ▷ Stoch training
  $b_i \leftarrow itr2batch(i)$
  $sts_{id} \leftarrow \text{ACCU}(b_i)$
  $sts \leftarrow \text{MERGE}(sts_1, sts_2 \ldots, sts_{id}, \ldots, sts_{bNum}, w_1, w_2, \ldots, w_{bNum})$
  $\lambda \leftarrow \text{UPDATE}(\lambda, sts)$
end for

Algorithm 2: Proposed stochastic approach to the training. Here λ represents model parameters, $tradIter$ is the number of traditional training iterations for given task, $tradData$ is the total amount of training data, RBS is the requested batch size, and $bNum$ is the number of unique batches (i.e. training data subsets) to be used in the training.

For example, one can decide that the training data set of $tradData = 30000$ (hours) should be sampled to generate $bNum = 60$ data subsets of $RBS = 500$ (hours) each. The order in which the batches are processed is then predefined in $itr2batch$ which corresponds to training criterion (2) and keeps record of which batch gets processed at which

algorithm iteration. In Algorithm 2 SAMPLE is an implementation of criterion (3) and generates subsets of training data by randomly sampling the full set at the speaker or the utterance level to produce a data batch of requested size. MERGE merges statistics sts_i with respective weights w_i implementing training criteria (4) and (5), and finally ACCU stands for parallelized statistics accumulation (identical to respective traditional method) on given data, and UPDATE updates model parameters λ using provided statistics sts.

3. EXPERIMENTS

3.1. Setup

Two sets of training data were used for presented experiments, one set with approximately 3000 hours and a second one with 30000 hours of US English search and messaging data captured from mobile phones. The test data was obtained from similar environment and topic domains as the training set. The reason for choosing two different sized training sets was to compare the effectiveness of proposed training strategies on both small and large training sets.

3.2. Results

Table 1 below shows the reduction in FDP and training time for the traditional and the most beneficial of various stochastic training approaches. Please note the reduction in training time is not always equal to reduction in necessary FDPs as the stochastic variant of the training requires more computational overhead.

Table 1: Reduction in full data passes (FDP) and training time (TT) for stochastic variants of respective trainings with no effect on model performance.

Stochastic vs Traditional	FDP Reduction	TT Reduction
ML 3k	67%	40%
ML 30k	77%	70%
FMPE 3k	43%	35%
FMPE 30k	58%	50%
MPE 3k	50%	46%
MPE 30k	92%	92%

3.2.1. ML

Experiments were performed with both 3K and 30K data. We were able to achieve a reduction in total number of FDPs with a concomitant decrease in training time for both train sets. With 3K hours, the amount of FDPs was reduced from 15 to 5 which corresponded to a real-time reduction of training time of about 40%. For the 30K hour case, the FDPs decreased from 15 to 3-4, which reduces the time to train to 1/3 of the original. Figure 1 shows results for different batch sizes.

The stochastic training setup that worked best for ML (cf. Section 2.3):

1. Batch size (RBS) = 500 hours.
2. Each iteration collects statistics on a fresh batch, ie $itr2batch = \{0, 1, 2, 3, \ldots\}$.
3. Random data sampling at the speaker $level$.
4. Process accumulated statistics as in traditional algorithm, ie $w_{n=itr2batch(i)} = 1$ and $w_{n\neq itr2batch(i)} = 0$.

Interestingly, if the batch size is set smaller, to about 50 hours, the setup for training criteria (4) and (5) was no longer valid and the training benefited from reusing the statistics accumulated in previous iterations up to the size of the original training data (3k trainings, data not shown). The smoothing weights w_i were chosen to be proportional to the batch size.

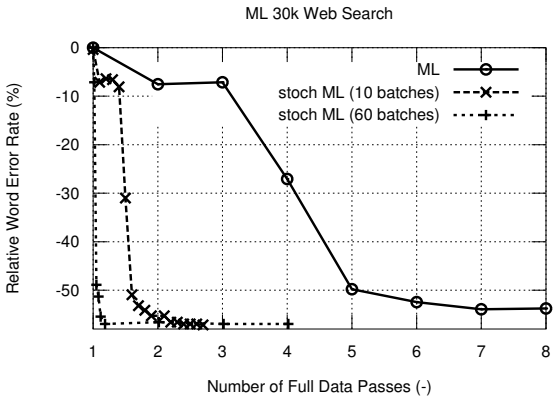

Fig. 1: The number of stochastic batches can lead to different performance in terms of full data passes vs. error rate. Please note the visible stagnation in likelihood improvement at the beginning of a training is due to GMM splitting, an operation not related to the topic of this paper.

3.2.2. fMPE

As shown in Table 1, fMPE training also benefited from using stochastic training techniques. Same accuracy can be achieved using about 58% less number of full data passes for the 30K hour case and about 43% less for 3K hour training. Figure 2 illustrates improved accuracy of traditionally vs. stochastically trained models for tested number of full data passes.

The setup that worked best for fMPE (30K):

1. Batch size (RBS) = 15k hours.
2. Use identical batch for 4 consecutive iterations, ie $itr2batch = \{0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, \ldots\}$.

3. Random data sampling at the speaker $level$.

4. Process accumulated statistics as in traditional algorithm, ie $w_{n=itr2batch(i)} = 1$ and $w_{n\neq itr2batch(i)} = 0$.

5. Do not transform the features when switching to a new batch, i.e. run the algorithm as if it was the first traditional iteration [3].

3.2.3. MPE

The story for model space training is similar to ML and fMPE, with an additional (if unexpected) result. We see that for MPE, the accuracy of the traditional model is reached very quickly, in about a quarter of an FDP for the 30k hour data, leading to training times reduction of 92%. For 3k hours, the reduction of necessary FDPs is 50% with corresponding reduction of 46% in training time.

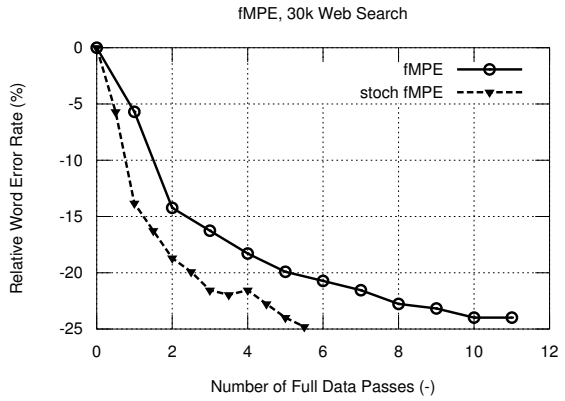

Fig. 2: Relative word error rate of traditionally trained and stochastically trained fMPE models. The number of iterations is limited to 11 iterations in our production training due to the cost of the fMPE procedure.

The stochastic training setup that worked best for MPE (30K) was:

1. Batch size (RBS) = 15k hours.

2. Each iteration collects statistics on a fresh batch, ie $itr2batch = \{0, 1, 2, 3, \ldots\}$.

3. Random data sampling at the speaker $level$.

4. Process accumulated statistics as in traditional algorithm, ie $w_{n=itr2batch(i)} = 1$ and $w_{n\neq itr2batch(i)} = 0$.

In addition, we also observed that if we continue stochastic training past the accuracy equivalency point, we can achieve small improvements in accuracy. We were skeptical at this discovery since we have seen in the past that this could be attributed to a speed versus accuracy shift (improved performance of the model in terms of accuracy vs. associated increase in run time of the decoding process). In order to ascertain if this was the case, a series of experiments with an FST decoder was carried out which allowed for various beam settings in order to measure accuracy vs. real time factor (RTF) characteristics of tested model for given task and decoding setup. The result of this experiment is shown in Figure 3 for a combination of fMPE and MPE trainings. In this plot, we see that, for all levels of RTF, the models trained from stochastic MPE and fMPE are always more accurate than the one trained with traditional training.

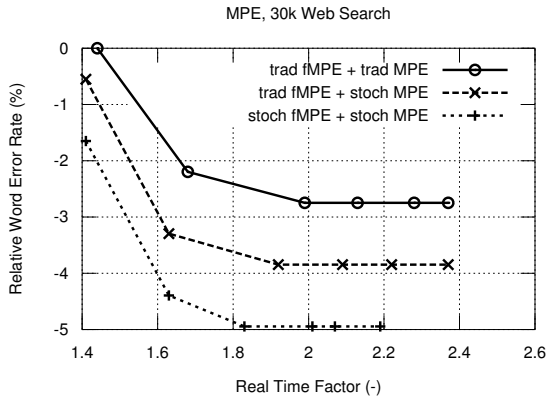

Fig. 3: Real time factor vs. Relative word error rate characteristics of a model trained with traditional MPE algorithm on top of traditionally trained fMPE model, an MPE on top of fMPE mode, both stochastically trained.

4. CONCLUDING REMARKS

The main motivation for this work was to improve the speed of training GMM HMM acoustic models, especially with large amounts of training data. Experiments have shown that proposed techniques work well when training with larger amounts of data. We typically see a decrease in passes through the training data of 30-50% while maintaining the same model accuracy. As expected, the impact of stochastic approach increases with larger training data amounts. For 30k hr training data, we saw decreases in training time of 70% for ML training, 50% for fMPE and even 91% for MPE. Moreover we observed that we could also achieve improved accuracy if we let the training continue to the same levels (in time) as traditional training.

5. ACKNOWLEDGEMENTS

The authors would like to thank Peder Olsen and Etienne Marcheret for useful discussions related to stochastic training techniques.

6. REFERENCES

[1] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[2] Mark Gales and Steve Young, "The application of hidden markov models in speech recognition," *Foundations and Trends in Signal Processing*, vol. 1, no. 3, pp. 195–304, 2008.

[3] Daniel Povey, Brian Kingsbury, Lidia Mangu, George Saon, Hagen Soltau, and Geoffrey Zweig, "fmpe: Discriminatively trained features for speech recognition," in *Proc. ICASSP*. Philadelphia, 2005, vol. 1, pp. 961–964.

[4] Radford M Neal and Geoffrey E Hinton, "A view of the em algorithm that justifies incremental, sparse, and other variants," in *Learning in graphical models*, pp. 355–368. Springer, 1998.

[5] Dong Yu, Li Deng, Xiaodong He, and Alex Acero, "Large-margin minimum classification error training: A theoretical risk minimization perspective," *Computer Speech & Language*, vol. 22, no. 4, pp. 415–429, 2008.

[6] Dong Yu, Li Deng, Xiaodong He, and Alex Acero, "Use of incrementally regulated discriminative margins in mce training for speech recognition.," in *INTERSPEECH*, 2006.

[7] Chaojun Liu, Hui Jiang, and Luca Rigazio, "Recent improvement on maximum relative margin estimation of hmms for speech recognition," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*. IEEE, 2006, vol. 1, pp. I–I.

[8] Jonathan Le Roux and Erik McDermott, "Optimization methods for discriminative training.," in *Interspeech*, 2005, pp. 3341–3344.

[9] Ralf Schlüter, Wolfgang Macherey, Boris Müller, and Hermann Ney, "Comparison of discriminative training criteria and optimization methods for speech recognition," *Speech Communication*, vol. 34, no. 3, pp. 287–310, 2001.

[10] Xiaodong He, Li Deng, and Wu Chou, "A novel learning method for hidden markov models in speech and audio processing," in *Multimedia Signal Processing, 2006 IEEE 8th Workshop on*. IEEE, 2006, pp. 80–85.

[11] Daniel Povey, Dimitri Kanevsky, Brian Kingsbury, Bhuvana Ramabhadran, George Saon, and Karthik Visweswariah, "Boosted mmi for model and feature-space discriminative training," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 4057–4060.

[12] PS Gopalakrishnan, Dimitri Kanevsky, Arthur Nadas, and David Nahamoo, "A generalization of the baum algorithm to rational objective functions," in *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*. IEEE, 1989, pp. 631–634.