Automated Optimization Of Decoder Hyper-Parameters for online LVCSR

Akshay Chandrashekaran[1], Ian Lane[1,2]
[1]Electrical and Computer Engineering, Carnegie Mellon University
[2]Language Technologies Institute, Carnegie Mellon University
akshayc@cmu.edu, lane@cmu.edu

Carnegie Mellon University

Introduction

- ASR decoding on servers should take as less computational resources as possible.
- More users can then be served by allowing more instances to run in parallel.
- Given:
  - DNN-HMM Acoustic Model and 3 and 4-gram Language Model
  - Validation Dataset
  - Online hybrid lattice-based rescoring decoder
- Objectives:
  - Minimize word error rate (WER), real-time factor (RTF) and memory footprint for the decoder.
  - Minimize WER given a hard constraint of 0.1 on RTF.

Approach

- Scalarize the objectives and minimize the scalarization. Techniques Compared:
  - Manual Optimization
  - Random Sampling
  - Genetic Algorithm
  - Tree of Parzen Estimators Approach
  - Gaussian Process based Bayesian Optimization
- Formulate RTF as a constraint and minimize WER with RTF constraint. Techniques Compared:
  - Manual Optimization
  - Constrained Random Sampling (CRS)
  - Constrained Bayesian Optimization with Gaussian Processes (CBO)

Decoder

- Online Hybrid Lattice based rescoring decoder.
  - AM likelihood computation on GPU
  - Decoder graph traversal and rescoring on CPU.
  - Rescoring done with a const-arpa language model.

Hyper-parameter	Range/Values
Acoustic Scale	0.05 - 0.3
Decoder Beam	10.0 - 18.0
Maximum number of active states	3000:500:8000
Minimum number of active states	50:50:300
Lattice Pruning beam	4.0 - 10.0
Lattice Pruning interval	5:5:50 frames

Table 1: Ranges for the decoder hyper-parameters used in all optimization techniques

Scalarization

- Augmented Tchebyscheff Function (ATF)
  - Combines multiple objectives into a single value using a scalarizing vector.
  - $ATF(x) = \max_j (w_j \hat{f}_j(x)) + \rho \sum_{k=1}^{M} w_k \hat{f}_k(x)$ where $w_i \geq 0\ \forall i$ and $\sum_{i=1}^{M} w_i = 1$
  - Selected weights: WER: 0.8, RTF: 0.1, Memory: 0.1

Optimization Techniques Setup

- All Optimizations run for 25 iterations
- Manual Optimization
  - Best acoustic scale found by using the language model reweighting giving best performance at full beam settings
  - Decoder beam search done till a point where WER degraded by 10%.
  - Lattice beam search done till WER degraded by an additional 10%.
  - Maximum number of active states searched up to point that there was no degradation in performance.
- Genetic Algorithm
  - 5 iterations with population of 5
  - Mutation Probability:0.02, Crossover Probability: 0.95

Experimental Setup

- Acoustic Model
  - Feed-Forward Fully connected DNN Architecture
  - Trained using LibriSpeech Corpus
  - Input Feature frame: Log of Filterbank with 23 filterbanks
  - Input to DNN: Current frame is spliced with 5 frames from past and future
  - 5 Hidden layers
  - 2048 Neurons per hidden layer with ReLU activation
  - 5683 Output Acoustic States using a SoftMax Layer
- Language Model
  - Weak LM: Bigram LM
    - 322k unigrams
    - 67M bigrams
  - Const ARPA LM:
    - 322k unigrams
    - 67M bigrams
    - 72M trigrams
    - 51M 4-grams
- Evaluation Dataset:
  - Provided by LGE
  - 6000 Utterances, 5.2 hours
  - Recorded from cell-phones
  - SMS transcriptions and commands
  - All audio for training and evaluation sampled at 16kHz

Results: Approach 1

Technique	WER (%)	RTF	Memory (GB)	ATF
Manual	15.39	0.0969	44.28	0.1340
Random Sampling (average)	14.04	0.4122	44.33	0.1236
Genetic Algorithm	14.02	0.3562	44.34	0.1234
Tree of Parzen Estimators	14.52	0.2475	44.29	0.1273
Bayesian Optimization	**13.85**	**0.8453**	**44.43**	**0.1226**

Table 2: Results of the single-objective hyper-parameter optimization techniques with ATF as the objective

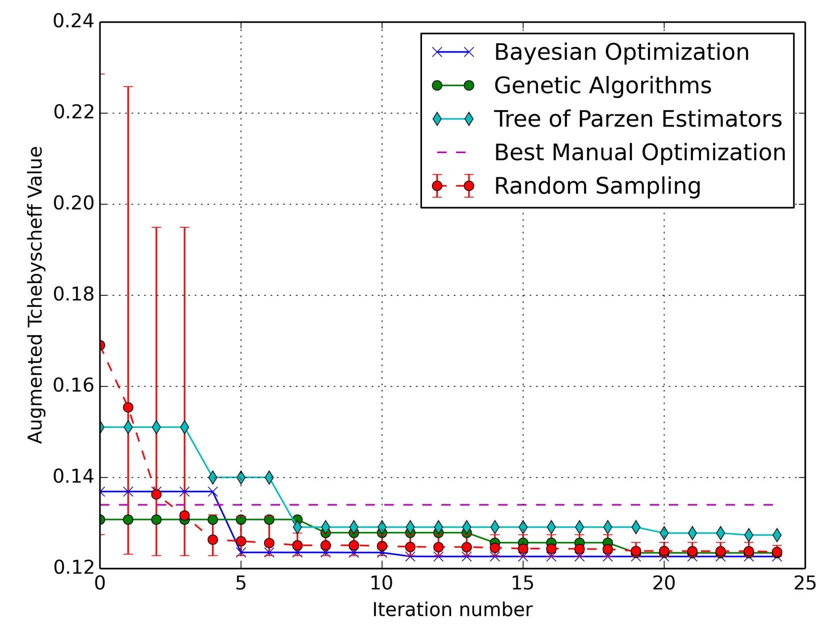

Figure 1: ATF of best model at a given iteration for different optimization techniques

Random Sampling are averaged over 12 independent runs.

Toolkits Used

Toolkit	Optimization Techniques
Spearmint	Bayesian Optimization, Constrained Bayesian Optimization
HyperOpt	Tree of Parzen Estimators
PyGMO	Genetic Algorithms

Acknowledgements

This work was supported in part under research grant 33912.1.1011568 from LGE Electronics.

Results: Approach 2

Technique	WER(%)	RTF
Manual	15.39	0.0969
Constrained Random Sampling (average)	15.70	0.0841
Constrained Bayesian Optimization	**14.99**	**0.0907**

Table 3: Results of constrained hyper-parameter optimization techniques with WER as objective and RTF constraint of 0.1

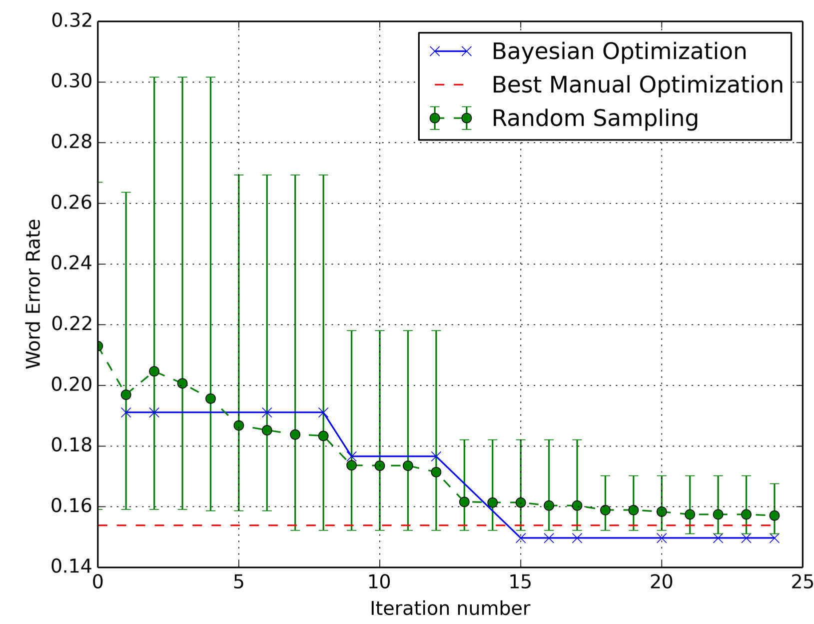

Figure 2: WER of best model satisfying constraints for different constrained optimization techniques

Constrained Random Sampling are averaged over 12 independent runs.

Discussion

- For Approach 1:
  - Bayesian Optimization is best (8.5% better ATF than manual optimization)
  - Automated Techniques outperform Manual Optimization within 8 iterations
  - Memory Usage as a optimization metric was largely uninformative.
- For Approach 2:
  - Constrained Bayesian Optimization performed best (2.7% better WER than manual)
  - Optimum achieved with 40% fewer iterations.
  - Constrained Random Sampling performed poorer than manual optimization on average

Conclusion

- Usage of automated optimization strategies outperformed manual optimization for same number of iterations.
- For small number of iterations, evolutionary algorithms are no better than random sampling.
- Bayesian Optimization outperformed other optimization strategies for ATF objective.
- Constrained Bayesian Optimization is a generalized approach and should be considered as a formal outer loop to the training and decoding procedure.
- Scalarized optimization requires careful tuning of objective weights, making constrained optimization a more attractive choice.

References

[1] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012.

[2] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, pp. 2951–2959, 2012.

[3] J. Bergstra, D. Yamins, and D. D. Cox, "Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms," Citeseer, 2013.

[4] K.-F. Man, K. S. Tang, and S. Kwong, *Genetic algorithms: concepts and designs*. Springer Science & Business Media, 2012.

[5] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584, IEEE Signal Processing Society, 2011.

[6] M. A. Gelbart, J. Snoek, and R. P. Adams, "Bayesian optimization with unknown constraints," *UAI*, 2014.