A NONMONOTONE LEARNING RATE STRATEGY FOR SGD TRAINING OF DEEP NEURAL NETWORKS

*Nitish Shirish Keskar**

Northwestern University
Evanston, IL

George Saon

IBM T.J. Watson Research Center
Yorktown Heights, NY

ABSTRACT

The algorithm of choice for cross-entropy training of deep neural network (DNN) acoustic models is mini-batch stochastic gradient descent (SGD). One of the important decisions for this algorithm is the learning rate strategy (also called stepsize selection). We investigate several existing schemes and propose a new learning rate strategy which is inspired by nonmonotone linesearch techniques in nonlinear optimization and the NewBob algorithm. This strategy was found to be relatively insensitive to poorly tuned parameters and resulted in lower word error rates compared to Newbob on two different LVCSR tasks (English broadcast news transcription 50 hours and Switchboard telephone conversations 300 hours). Further, we discuss some justifications for the method by briefly linking it to results in optimization theory.

***Index Terms*—** deep learning, speech recognition, stepsize, learning rate, nonmonotonicity

1. INTRODUCTION

Deep Neural Networks (DNNs) have become the model of choice for acoustic modeling in automatic speech recognition systems [1]. The problem of training a DNN translates to that of mathematical optimization with a non-convex objective function. These problems are categorically difficult to solve and require efficient starting points and search directions. For solving such problems, there are two major steps in the iterative process: computation of search direction and decision about learning rate [2]. Depending on the algorithm, the search directions are either sub-sampled gradients or linear transformations of sub-sampled gradients through Hessian-Free or Quasi-Newton schemes. The choice of learning rate, while an important decision, is often rather arbitrary. Common choices are, either a fixed value, a pre-determined schedule, or an adaptive scheme based on curvature or gradient information. The motivation behind the latter mechanisms is to allow aggressive optimization in the beginning by allowing larger steps (and hence, a larger learning rate) but cutting back later in the optimization during the "fine-tuning" phase. Adaptive schemes have been preferred to pre-determined schedules primarily because of the lack of tuning involved and the fact that the decrease in steps is driven by the curvature estimates rather than trial-and-error [3, 4]. The problem of choosing learning rates has been investigated by the optimization community with many options available which ensure global convergence under regularity conditions. Examples include Armijo-like and Wolfe linesearch. However, all of the said choices face two major flaws: they are restrictive in allowing the learning rates to stay higher and they completely ignore the cross-validation (CV) set during training. Through the use of adaptive schemes or schedules, the learning rate is gradually reduced however, this reduction might be too restrictive and the algorithm may stagnate prematurely. Also, while the goal of the optimization algorithm is to best fit the training data, it is not entirely in line with the overarching goal of having the best model to make predictions on unseen data. Indeed, the experiments with linesearch mechanisms popular in the optimization community suggest that driving the error on the training set to the absolute minimum does result in an overall worse model in terms of testing error. Thus, it is necessary to make step acceptance decisions on the basis of how well the model behaves on unseen data rather than how well it fits the training data. To do this, it becomes necessary to use the heldout dataset during the training phase as opposed to only using it once the model has been trained. We discuss some statistical repercussions of this choice. This strategy of using the CV set during training is not new and has been proposed earlier under the name NewBob [5] and has been redesigned in a recent paper to make it less restrictive [6]. We propose a nonmonotone variant of this strategy which brings forth two significant improvements: firstly, it allows the learning rate to stay larger for a longer period of time thus allowing rapid progress when the stochastic gradients can be trusted and secondly, the algorithm becomes less sensitive to the choice of the initial parameters. In summary, the proposed algorithm is more permissive and yet, more robust when used as the learning rate strategy for minibatch stochastic gradient descent.

2. NONMONOTONE LEARNING RATE STRATEGY

In this section, we describe our proposed nonmonotone learning rate strategy, its motivation and theoretical properties.

2.1. General setting and notation

In order to effectively build connections with the literature in the field of nonlinear optimization, let us abstract the problem of training neural networks into that of non-convex optimization. We are interested in optimizing the function $f(x)$ which is the average of M loss functions each corresponding to a unique datapoint. Thus, we are interested in finding the vector $x^\star$ in the search space $\mathbb{R}^N$ such that

$$x^\star = \arg\min_x f(x),\ f(x) = \frac{1}{M}\sum_{i=0}^{M} f_i(x)$$

The standard minibatch SGD used for solving this problem starts off with an initial estimate of the solution x_0 and follows the iterative scheme:

*Work was performed at the IBM T.J. Watson Research Center, Yorktown Heights, NY. This author was partially supported by the National Science Foundation grant DMS-1216920.

$$x_{k+1} = x_k - \eta_k \hat{\nabla} f(x_k)$$

where $\hat{\nabla} f(x)$ is the minibatch stochastic gradient computed as $\hat{\nabla} f(x) = \frac{1}{m} \sum_{i \in \mathcal{I}} \nabla f_i(x)$ with $\mathcal{I}$ being an index set of cardinality m. For the purposes of this study, we use a fixed batch size which attempts to compromise between the reliability of the gradient and the computational effort. In the following subsections, we briefly survey the techniques used to determine η_k for effective convergence, discuss their drawbacks and then propose a new strategy.

2.2. Prior work

In this subsection, we discuss some of the techniques that have been proposed in literature to determine η_k. This has been a fairly active topic of research and the discussion is far from exhaustive. We classify the various schemes under broad categories and discuss two competitive strategies in greater detail. Figure 1 aims to classify some of the popular learning rate strategies on the basis of the fundamental idea.

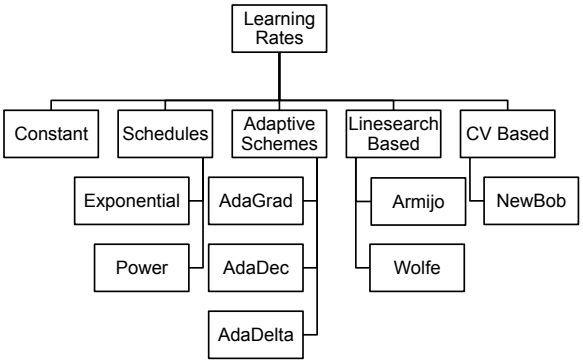

Fig. 1. Classification of popular learning rate strategies.

2.2.1. Adaptive schemes

These schemes are based on reducing the stepsizes on the basis of accumulated gradient information. Examples include [3, 4, 7].These schemes allow for each weight in the DNN to have its own individual learning rate. For instance, given any unit i, AdaGrad uses the stepsize of the form

$$\eta_{ik} = \frac{\eta_{i0}}{\sqrt{\sum_{j=0}^{k-1} \hat{\nabla}_i^2 f(x_j)}}$$

where we slightly abuse the notation $\hat{\nabla}_i f(x_j)$ to mean the i^{th} component of the stochastic gradient computed at iteration j. These methods are documented to require much less tuning but we lose control over the decay rate of the stepsizes and adversarial gradient estimates can lead to steps reducing much faster than necessary. This fast stepsize reduction was found to be beneficial in the context of asynchronous SGD such as the DistBelief framework [8].

2.2.2. NewBob scheme

This scheme starts with a relatively high stepsize value and uses the cross-validation set to make judgments about reducing the stepsizes. What differentiates this mechanism from others is that is uses the CV set during training. The specific algorithm used in QuickNet [9] starts with a reasonably high learning rate and keeps that for subsequent epochs until the error on the CV set decreases by less than 0.5% over the previous epoch. After that, the learning rate is halved before each epoch to home-in with increasing precision on the local optimum. Recent work by Wiesler et al. [6] uses a representative chunk of the training dataset instead of the cross-validation set and attempts to make this less restrictive by reducing the stepsize for the current epoch only if there isn't sufficient improvement over the previous one.

2.2.3. Linesearch schemes

Although very popular in nonlinear optimization [2], they are not particularly well suited for DNN training because of expensive function evaluations and noise in the stochastic gradient estimates. Arguably the most popular strategy of backtracking attempts to find $\eta_k \in \{2^0, 2^{-1}, 2^{-2}, \cdots\}$ such that $f(x_{k+1}) < f(x_k) - \epsilon \cdot \eta_k \cdot \|\nabla f(x_k)\|^2$. This strategy is ill-suited because the sufficiency requirement cannot be well estimated using noisy gradients and the repeated backtracking would involve multiple forward passes. Predetermined schedules (including constant stepsizes) can be tuned to perform well but the tuning process can be involved and the same strategy does not generalize to other datasets.

2.3. Proposed strategy

In this subsection, we describe the proposed learning rate and its advantages and theoretical properties. We start off by motivating the idea and discussing its provenance. The problem of finding the unconstrained minima of a function is quite well-studied in the optimization community. As mentioned in the previous sections, the popular linesearch algorithms used to solve these problems usually consist of 2 major steps: step computation and step acceptance. These two steps, in conjunction, lead to globally convergent algorithms. However, because function and gradient evaluation in the context of DNN training is expensive, the schemes are adapted to function well in DNN training. An adaptation of the popular backtracking idea is that the current step is cut in half and the optimization continues irrespective of whether the reduced step is satisfactory. This forms the basis of the diminishing stepsize scheme for SGD [10]. When the function being evaluated is using the CV set, it has been known to perform extremely well [6]. However, it has been known in the optimization community that enforcing monotonicity on function decrease can greatly hamper convergence speeds in the intermediate stages of optimization and more so when there is no resetting of the learning rate [11, 12]. Thus, we propose a non-monotone variant of this scheme where we allow more freedom for the steps and require functional decrease every J iterations where J is user-defined. Further, we keep memory of the best known iterate which is used as part of the algorithm to safeguard against adversarial steps.

2.3.1. Algorithm

In this subsection, we describe the algorithm in detail.

Algorithm 1 Minibatch SGD with non-monotone step selection

Given $x_0 \in \mathbb{R}^N$, $\eta_0, \epsilon > 0$ and $m, J, H \in N$
$x_{best} = x_0$ and $\Phi_{best} = f(x_0)$
$t = 0$ and $\Phi_0 = f(x_0)$
for $k = 0, 1, 2, \cdots$ **do**
  Compute $\hat{\nabla} f(x_k) = \frac{1}{m} \sum_{i \in \mathcal{I}_k} \nabla f_i(x_k)$
  $x_{k+1} = x_k - \eta_k \hat{\nabla} f(x_k)$
  if $k \bmod H = 0$ **then**
    $\hat{\Phi} = \max(\Phi_{t-1}, \Phi_{t-2} \cdots, \Phi_{t-J})$
    if $f(x_{k+1}) > \hat{\Phi} - \epsilon \cdot \eta_k \cdot \|\hat{\nabla} f(x_k)\|^2$ **then**
      $\eta_{k+1} = 0.5 \cdot \eta_k$
      if $f(x_{k+1}) > \hat{\Phi}$ **then**
        $x_{k+1} = x_{best}$
      end if
    else
      $\eta_{k+1} = \eta_k$
      if $f(x_{k+1}) \leq \Phi_{best}$ **then**
        $x_{best} = x_{k+1}$ and $\Phi_{best} = f(x_{k+1})$
      end if
    end if
    $t = t + 1$
    $\Phi_t = f(x_{k+1})$
  end if
end for

The version of the algorithm described uses an Armijo-like condition on the training dataset for sufficient decrease. However, for practical implementations, this can be replaced by some other acceptable sufficiency condition on the cross-validation set such as the NewBob strategy $1.01 \cdot f(x_{k+1}) > \hat{\Phi}$ which is what we ended up using in our experiments.

While this change over the existing framework may appear to be trivial and incremental, in reality, it has significant repercussions on the convergence of the algorithm. Firstly, the nonmonotone requirement allows the stepsize to remain high for a longer duration but is restrictive when sufficient progress is not being made. This is especially useful when the initial stepsize is underestimated. Further, it has been known that nonmonotone strategies are especially powerful in the intermediate stages of optimization [11]; which in our case, is an important chunk given that the optimization is never continued till the gradients drop to 0. The safeguards allow for incremental progress and prevent situations where the objective continues to get worse for an extended duration of the optimization. The idea for safeguards is rooted in the validity of the Taylor model around the iterate. In case of insufficient progress, by reverting back to best known iterate and cutting back steps, we improve the accuracy in the Taylor model thus increasing the possibility of making progress. If the dataset to determine sufficiency is chosen to be the CV set, similar to the NewBob strategy, the model does return iterates which have better generalization and testing error.

2.3.2. Theoretical justifications

In this section, we attempt to justify the method theoretically. We summarize prior work in cyclic gradient methods and nonmonotone linesearches and their relation with the proposed scheme. Conceptually, our method can be thought of as a cyclic perturbed gradient method with a nonmonotone step acceptance strategy. The literature in nonmonotone step acceptance is quite rich. It is known that under regularity conditions, nonmonotone Armijo-like strategies converge [11, 12]. Further, we know that cyclic versions of these methods i.e., when the step is kept constant for a duration of time before being updated, also converge; for instance, see [13] for cyclic nonmonotone strategy with heuristic initialization. This would indeed be the case if each gradient in our algorithm was using the entire batch instead of stochastic. Because of the fact that batch gradient computations are often expensive and unnecessary; and that backtracking, owing to function evaluations, can be expensive, the modified algorithm retains the speed of SGD at the expense of less informative theory.

Further, since we are using mini-batch gradient estimates as opposed to batch estimates, the directions that we search along can be thought of as perturbed gradient directions. This line of thought of characterizing the Incremental Gradient Algorithm (IGA), an algorithm similar to SGD, as a perturbed gradient has been used to demonstrate convergence of incremental gradient methods. In [14], under conditions of strong growth, the author proves convergence of IGA with a small enough stepsize. The strong growth condition requires boundedness of each stochastic gradient relative to the batch gradient. Mathematically, we require that there exist a $B \in \mathbb{R}$ such that for all x, $\max_i\{\nabla_i f(x)\} \leq B\|\nabla f(x)\|$. Under more binding assumptions, linear rate of convergence can be proved for such methods [15]. Recently, it was proved that under the strong growth assumption, SGD would also be globally convergent for constant steps small enough and the rate is sublinear [16].

Conditions for global convergence of SGD usually require a stepsize which decays at a rate which is "not summable but square summable" [17, 18]. Adaptive schemes like the one we propose do not fall into this category, making convergence proofs more involved. However, if we start with a stepsize small enough, on the basis of the aforementioned results, we can expect that the gradient estimates will be descent directions and we will make sufficient progress leading to convergence. Further justification can be made by comparing our algorithm and a similar approach discussed in [19]. In that paper, the authors prove global convergence of SGD for nonconvex problems while using a cyclic choice of a heuristic stepsize. While a complete proof for our algorithm, combining the ideas discussed above, would be informative; such an analysis would be quite involved and left as topic of future research.

3. EXPERIMENTAL RESULTS

We ascertain the viability of the proposed method on two speech recognition datasets: English broadcast news transcription 50 hours (EBN50) and Switchboard conversational telephone speech 300 hours (SWB300). For both tasks, we compare cross-entropy trained DNN acoustic models in a hybrid decoding setup using either the NewBob step selection strategy or the proposed method.

3.1. Frontend processing

Feature extraction is similar for both tasks and consists in extracting 13-dimensional VTL-warped PLP cepstra every 10ms which are mean and variance normalized for every speaker. Every 9 consecutive frames are spliced together and projected to 40 dimensions using LDA followed by one global semi-tied covariance transform. The LDA/STC features are further adapted with one feature-space MLLR transform per speaker and both training and test time. The input to the DNNs is formed by consecutive FMLLR feature vectors within a temporal window of ± 4 frames for EBN50 and ± 5 frames for SWB300, respectively. For Switchboard, we also add i-vector based speaker adaptation [20] by appending a 100-dimensional i-vector to every stack of 11 FMLLR frames resulting in an input dimension of 540 (versus 360 for EBN50).

3.2. Network topology and CE training

The DNNs have 6 hidden layers with sigmoid activation functions and a softmax output layer whose units correspond to the leaves of a phonetic decision tree with pentaphone crossword context. For EBN50, each hidden layer has 1024 units except the last one which has 512 and the output layer contains 5000 units. For SWB300, the hidden layers are of size 2048 except the last one which has 256 units and the output layer is comprised of 9300 neurons. The baseline nets are trained with 20 passes of cross-entropy SGD on frame-randomized minibatches of size 256 for EBN50 and 250 for SWB300. All DNNs are initialized with layerwise discriminative pretraining [21] by running one cross-entropy sweep over the training data.

3.3. English BN 50 hours results

in Figure 2 we show the phone frame error rates on the held-out data for various SGD runs using the NewBob strategy (baseline) and the proposed non-monotone scheme. The experimental settings for the non-monotone algorithm are as follows. We used a history of 5 previous objective function values ($J = 5$ from Algorithm 1) and we compute the cross-entropy loss twice per epoch (i.e. $H = K/2$ where K is the number of training minibatches). While not reported here, we have also experimented with varying J from 3 to 7 while keeping H fixed and observed a similar performance.

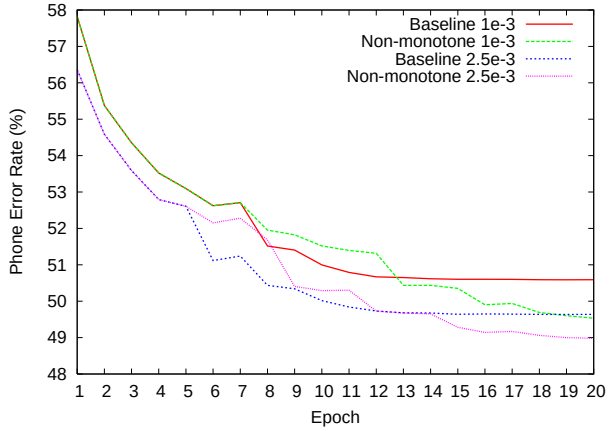

Fig. 2. Phone frame error rates on heldout data for EBN50.

As can be seen, the non-monotone strategy achieves lower phone error rates at the end of the optimization because it keeps a higher stepsize longer compared to the baseline whereas the latter is too aggressive in the stepsize reduction. As a consequence, the baseline algorithm converges prematurely to a suboptimal phone error rate. This is also reflected in Table 1 where we compare the word error rates of the two algorithms for different initial stepsizes. The WERs were computed on the DEV'04f testset using an 84K word vocabulary with 90K pronunciations and a 4-gram LM with 3.3M n-grams.

We observe that the non-monotone strategy outperforms the baseline NewBob algorithm for all initial stepsizes although the gain in performance is reduced if the initial stepsize is carefully tuned.

3.4. Switchboard 300 hours results

Similar to the EBN50 setup, in Table 2 we compare the word error rates obtained for the SGD runs with different initial stepsizes. The

Initial stepsize	Baseline	Non-monotone
1.0e-3	18.2	17.6
2.5e-3	17.7	17.1
5.0e-3	17.0	16.7
1.0e-2	17.4	16.9

Table 1. Word error rates on the DEV'04f testset for EBN50.

WERs were computed on the Hub5'00 testset using a 30.5K word vocabulary with 32.8K pronunciations and a 4-gram LM with 4M n-grams. Here, the results are mixed: while our method improves for a poor initial stepsize selection, it does not result in gains over NewBob if the initial learning rate is adequate.

Initial stepsize	Baseline	Non-monotone
1.0e-3	13.5	13.1
2.5e-3	13.2	13.2
5.0e-3	13.2	13.3
1.0e-2	13.5	13.3

Table 2. Word error rates on the Hub5'00 testset for SWB300.

4. DISCUSSION

In this paper, we proposed a new learning rate strategy which builds on the ideas of nonmonotone search and using a cross-validation set during training. We justified the algorithm theoretically and demonstrated good performance on two speech recognition tasks. Nonmonotone search is popular in nonlinear optimization for use when the method is expected to do well in general but cannot make sufficient progress on each iteration. This is precisely the situation in DNN training; the minibatch gradients and an appropriately chosen constant stepsize are expected to do well in general, but sufficient progress after each epoch cannot be guaranteed. By using a few more function evaluations, we allow for a more rapid decrease in objective while retaining the robustness of other competitive methods. Further, by reverting back to the best known iterate in case of insufficient progress, we allow for a more controlled decrease by preventing the objective functions from increasing for an extended duration. The idea of nonmonotonicity does not depend on the dataset being used for determining sufficiency; by using the cross-validation set however, we are able to achieve better generalization. It is known that using solely the training error is suboptimal because the model adapts to the idiosyncrasies of the training dataset leading to optimistic biases and inferior generalization. Cross-validation allows for better model selection by corroborating the viability of the model against a new dataset giving an estimate for the generalization error. However, a subtle point about cross-validation is that the purpose of the dataset is to estimate the generalization of the method used to generate the model as opposed to the model itself. Thus, using the cross-validation set during training might lead to optimistic generalization errors [22]. Yet, this procedure simplifies the process to improve generalization and leads to better training outcomes in our experiments. While the learning rate strategy is described in the context of minibatch SGD, the proposed method applies more generally. Preliminary experiments with stochastic quasi-Newton methods suggest potential improvements over other strategies. A detailed study of these improvements and a more complete convergence proof forms the basis of future work in this domain.

5. REFERENCES

[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. 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] J. Nocedal and S. Wright, "Numerical optimization, second edition," *Numerical optimization*, pp. 497–528, 2006.

[3] A. Senior, G. Heigold, M. Ranzato, and K. Yang, "An empirical study of learning rates in deep neural networks for speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6724–6728.

[4] M. Zeiler, "Adadelta: An adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[5] S. Renals and N. Morgan and H. Bourlard and M. Cohen and H. Franco and C. Wooters and P. Kohn, "Connectionist speech recognition: status and prospects," Tech. Rep. TR-01-070, ICSI, 1991.

[6] S. Wiesler, A. Richard, R. Schluter, and H. Ney, "Mean-normalized stochastic gradient for large-scale deep learning," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 180–184.

[7] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *The Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[8] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. Le, et al., "Large scale distributed deep networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1223–1231.

[9] "Quicknet," http://www1.icsi.berkeley.edu/Speech/qn.html, Accessed: 2014-09-30.

[10] Y. LeCun, L. Bottou, G. Orr, and K. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*, pp. 9–48. Springer, 2012.

[11] L. Grippo, F. Lampariello, and S. Lucidi, "A nonmonotone line search technique for newton's method," *SIAM Journal on Numerical Analysis*, vol. 23, no. 4, pp. 707–716, 1986.

[12] H. Zhang and W. Hager, "A nonmonotone line search technique and its application to unconstrained optimization," *SIAM Journal on Optimization*, vol. 14, no. 4, pp. 1043–1056, 2004.

[13] Y. Dai, W. Hager, K. Schittkowski, and H. Zhang, "The cyclic barzilai—borwein method for unconstrained optimization," *IMA Journal of Numerical Analysis*, vol. 26, no. 3, pp. 604–627, 2006.

[14] M. Solodov, "Incremental gradient algorithms with stepsizes bounded away from zero," *Computational Optimization and Applications*, vol. 11, no. 1, pp. 23–35, 1998.

[15] P. Tseng, "An incremental gradient (-projection) method with momentum term and adaptive stepsize rule," *SIAM Journal on Optimization*, vol. 8, no. 2, pp. 506–531, 1998.

[16] M. Schmidt and N. LeRoux, "Fast convergence of stochastic gradient descent under a strong growth condition," *arXiv preprint arXiv:1308.6370*, 2013.

[17] A Shapiro and Y Wardi, "Convergence analysis of gradient descent stochastic algorithms," *Journal of optimization theory and applications*, vol. 91, no. 2, pp. 439–454, 1996.

[18] Saeed Ghadimi and Guanghui Lan, "Stochastic first-and zeroth-order methods for nonconvex stochastic programming," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2341–2368, 2013.

[19] Xiao Wang, Shiqian Ma, and Wei Liu, "Stochastic quasi-Newton methods for nonconvex stochastic optimization," 2014.

[20] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Proceedings of ASRU*, 2013.

[21] F. Seide, G. Li, X. Chien, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proceedings of ASRU*, 2011.

[22] G. Cawley and N. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," *The Journal of Machine Learning Research*, vol. 11, pp. 2079–2107, 2010.