Acoustic Model Optimization Based On Evolutionary Stochastic Gradient Descent with Anchors for Automatic Speech Recognition

Xiaodong Cui, Michael Picheny

IBM Research AI
IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA
{cuix,picheny}@us.ibm.com

Abstract

Evolutionary stochastic gradient descent (ESGD) was proposed as a population-based approach that combines the merits of gradient-aware and gradient-free optimization algorithms for superior overall optimization performance. In this paper we investigate a variant of ESGD for optimization of acoustic models for automatic speech recognition (ASR). In this variant, we assume the existence of a well-trained acoustic model and use it as an anchor in the parent population whose good “gene” will propagate in the evolution to the offsprings. We propose an ESGD algorithm leveraging the anchor models such that it guarantees the best fitness of the population will never degrade from the anchor model. Experiments on 50-hour Broadcast News (BN50) and 300-hour Switchboard (SWB300) show that the ESGD with anchors can further improve the loss and ASR performance over the existing well-trained acoustic models.

Index Terms: automatic speech recognition, evolutionary SGD, population-based optimization, gradient-free optimization, evolutionary strategy

1. Introduction

Evolutionary stochastic gradient descent (ESGD) was proposed in [1] for optimization of deep neural networks (DNNs). It is a population-based [2] approach that integrates gradient-aware SGD and gradient-free evolutionary strategy (ES) [3][4] in one framework to take advantage of the merits of both families of algorithms to deal with complicated optimization landscapes of DNNs. In the meantime, a variety of SGD algorithms with various hyper-parameters are also taken into account in a co-evolution [5][6] setting of the population to leverage their complementariness. ESGD has shown its effectiveness in various domains such as speech, vision and language modeling.

The work to be reported in this paper is an extension of [1]. In [1], ESGD, given a fixed network architecture, starts optimization from scratch. In this paper, we investigate one variation of ESGD in which we assume we already have some well-trained model in place to begin with. This is a fairly common scenario in acoustic modeling for ASR. For instance, we want to improvement performance over an existing baseline model in terms of fitness. Therefore, the proposed ESGD with anchors will guarantee a monotonic non-increase in loss function starting from the existing model.

We briefly investigated a prototype of ESGD using anchors in language modeling in [1] where the inserted anchor is never changed. In this work, we comprehensively investigate the anchor strategy on acoustic models. The contributions of this work include: (1) Design an ESGD algorithm with anchors to leverage the existing baseline models and switch anchors when better candidates surface; (2) Introduce an iterative process to apply ESGD with anchors to monotonically improve the best fitness of the population; (3) Conducted experiments on 50-hour Broadcast News (BN50) and 300-hour Switchboard (SWB300) to show the effectiveness of the proposed ESGD with anchor algorithm.

The remainder of the paper is organized as follows. Section 2 formulates the ESGD optimization problem mathematically. Its implementation with anchor models will be given in Section 3 along with the proof of its monotonic non-increasing best fitness of population. Section 4 presents the experimental results on BN50 and SWB300 datasets followed by a discussion in Section 5. Finally, Section 6 concludes the paper with a summary.

2. Mathematical Formulation

Define the loss function

\[ l_i(\theta) \triangleq \ell(h(x_i; \theta), y_i) \]

where \( h \) is the function to be learned with parameter \( \theta \) which maps the input space \( \mathcal{X} \subseteq \mathbb{R}^{d_x} \) to the output space \( \mathcal{Y} \subseteq \mathbb{R}^{d_y} \) and \( \{(x_i, y_i)\}_{i=1}^n \in \mathcal{X} \times \mathcal{Y} \). We further assume \( \theta \) follows distribution \( p(\theta) \), data follows distribution \( p(x, y) \) and consider the expected empirical risk over \( p(\theta) \) and \( p(x, y) \)

\[ J = \mathbb{E}_\theta[\mathbb{E}_x[\ell_i(\theta)]]. \]

In ESGD [1], \( \omega \sim \text{Uniform}\{1, \cdots, n\} \) for SGD and a population of \( \mu \) candidate solutions, \( \{\theta_j\}_{j=1}^\mu \), is drawn to give the following average empirical risk of the population

\[ J_\mu = \frac{1}{\mu} \sum_{j=1}^\mu \left( \frac{1}{n} \sum_{i=1}^n \ell_i(\theta_j) \right) \]

Eq.3 is the objective function under ESGD.

Following [1], to avoid cluttered notation, we define the fitness function to be the empirical risk function, \( \frac{1}{n} \sum_{i=1}^n \ell_i(\theta_j) \), which is to be minimized. We define the \( m \)-elitist fitness: Let \( \Psi_m = \{\theta_1, \cdots, \theta_m\} \) be a population with \( \mu \) individuals and let
be a fitness function associated with each individual in the population. Rank the individuals in the ascending order
\[ f(\theta_1, \mu) \leq f(\theta_2, \mu) \leq \cdots \leq f(\theta_{m}, \mu) \] (4)
where \( \theta_{k, \mu} \) denotes the \( k \)-th best individual of the population [7]. We are interested in the fitness of the first \( m \)-best individuals (\( 1 \leq m \leq \mu \)). When \( m = 1 \), it amounts to the best fitness of the population.

Each individual in the population comes from a different optimizer with different hyper-parameters in each generation. After the gradient-based optimization, the individuals in the population will interact with each other in a gradient-free fashion to hopefully produce offsprings with better quality. In this way, the population-based ESGD not only can leverage the complementariness of various optimizers but also the complementariness of gradient-aware and gradient-free algorithms. Under a strategy of model back-off and elitist selection, ESGD guarantees that the best fitness in the population will never degrade ([1], Corollary 1).

3. ESGD with Anchors

Suppose we have a well-trained model in hand and want to further improve it without changing its architecture. Conventionally, we would pick an optimizer and its hyper-parameters and optimize the model until certain conditions (e.g., no improvement on the validation loss) met. This process is usually repeated multiple times and the best-performance model is chosen. ESGD can help with this situation with a population of optimizers and let them interact in the evolution stage. Previously in [1], ESGD starts with models with randomized weights without assuming the existence of any models. A baseline model reflects our prior knowledge in the model space, of which we can make use to also help ESGD. With this consideration, we propose the following ESGD algorithm with anchors (Algorithm 1). The motivation behind this algorithm is that we always insert the best updated model, starting from the provided baseline model, into the population in the evolution as an anchor to make sure its “genes” can spread out in future generations to not only accelerate the evolution process but also strive for better quality offspring.

In Algorithm 1, the initial population is created by inserting the existing DNN model \( \theta_{(0)}^{(0)} \) (the initial an anchor) into a family of otherwise randomly initialized networks. In the SGD step, the non-anchor models are optimized by a randomly selected SGD optimizer with randomly selected hyper-parameters for \( K_{s} \) steps while the anchor model stays unchanged. The SGD optimization is implemented with model back-off. If the new fitness gets degraded, the individual will roll back to the previous step. In the evolution step, model encoding is first conducted where the weights of the models are vectorized into a real-valued vector. Then offsprings are produced following the \((\mu/\rho+\lambda)\)-ES [3] rule:
\[ \theta^{(k)}_i = \frac{1}{\rho} \sum_{j=1}^{\rho} \theta^{(k)}_j + \epsilon^{(k)}_i \] (5)
where \( \rho \) parents are randomly selected in proportional to their fitness and \( \epsilon^{(k)}_i \sim N(0, \sigma^2_i) \) is Gaussian noise. Eq.5 amounts to a model average plus a perturbation. The probability of the anchor model showing up in Eq.5 is \( \rho \). Therefore, about \( \rho \) of the offsprings have the “genes” from the anchor. After the offspring population is generated, their fitness are evaluated. If the best fitness is better than that of anchor, the model with the best fitness with switch with the anchor to become the new anchor for the next round. The top \( m \) individuals are first selected and the rest \( \mu-m-1 \) individuals are then randomly picked from the non-\( m \)-elitist candidates. Finally, the updated anchor is inserted and new population is formed.

It can be shown the design of Algorithm 1 guarantees a non-degraded fitness from the initial anchor model. In fact, since the SGD step uses model back-off, the fitness of each individual in the population will not deteriorate. Furthermore, after the evolution step, the \( m \)-elitist strategy guarantees that individuals of
the top m fitness in the combined population of parent (excluding the anchor parent) and offspring population will enter the next generation. The best individual with the top-most fitness selected this way is then compared with the fitness of the anchor and whichever the better will become the next anchor. Therefore, the anchor in the next generation will be no worse than the anchor of the previous generation.

4. Experiments

Experiments are conducted on two datasets: BN50 and SWB300. The 300-hour data in BN50 consists of a 45-hour training set and a 5-hour validation set. The test set comprises 3 hours of audio. The acoustic models are fully-connected feed-forward network with 6 hidden layers and one softmax output layer with 5,000 units. There are 1,024 units in the first 5 hidden layers and 512 units in the last hidden layer. ReLU is used for the bottom 3 hidden layers and sigmoid for the rest. The input to the network is 9 frames of 40-dim features derived from speaker-adapted PLP [8] features.

The 300-hour data in SWB300 is split into 295 hours of training data and 5 hours of validation data. The test set is the Hub5 2000 evaluation set, composed of two parts: 2.1 hours of switchboard (SWB) data and 1.6 hours of call-home (CH) data. The acoustic models are bi-directional LSTM [9] with 4 layers. Each layer contains 1,024 cells with 512 in each direction. On top of the LSTM layers, there is a linear bottleneck layer with 256 hidden units followed by a softmax output layer with 32,000 units. The LSTMs are unrolled 21 frames. The input to the network is 140 in dimension which comprises 40-dim speaker-adapted PLP features after LDA and 100-dim i-vectors [10].

The cross-entropy (CE) is chosen as the fitness function which is measured on the validation set. We consider various SGD variants and ADAM [11] for the pool of optimizers. All optimization and fitness evaluation are carried out in parallel on multiple GPUs. We re-establish the single SGD baseline used in [1] as the references. The single baseline is trained using SGD with a batch size 128 without momentum for 20 epochs. The initial learning rate is 0.001 for BN50 and 0.025 for SWB300. The learning rate is annealed by 2x every time the loss on the validation set of the current epoch is worse than the previous epoch.

The size of the parent population is 100 and the offspring population 400. In ESGD, the hyper-parameters of optimizers which are to be chosen to create the parent population include batch size, learning rate, momentum and nesterov acceleration. Specifically, given a chosen optimizer, if applicable, there is a 80% chance using the momentum and 20% not using it. When using the momentum, there is a 50% chance using the nesterov acceleration. The momentum is randomly selected from [0.1, 0.9]. The batch size is uniformly chosen from [64, 128, 256, 512]. The learning rate is also randomly selected from a range $[a_0, b_0]$ depending on the generation $k$. The upper and lower bounds of the range are annealed over generations starting from the initial range $[a_0, b_0]$. For ADAM, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ are fixed and only the learning rate is randomized. The m-贪婪 strategy is applied to 60% of the population. The probability of mating of the anchor model is 0.25. Table 1 summarizes the hyper-parameter settings of the proposed ESGD with anchors for the two datasets.

Table 2 shows the single SGD baseline (loss=2.082, WER=17.4%, first row) and ESGD baseline (loss=1.916, WER=16.4%, third row) on BN50 respectively. In the first experiments, we use the SGD baseline and ESGD baseline as the initial anchors. The evolution of the best fitness (minimal loss) of the population over generations is shown in the left and right panels of Fig.1, respectively. The drop of loss is monotonic. On the left panel, it takes 5 generations to break the SGD baseline while on the right it takes 8 generations. After 20 generations, ESGD using the SGD baseline as the initial anchor has the loss of 1.935 and WER 16.5, which improves significantly over the baseline model. Note that the SGD baseline is already a decent model trained under well-tuned training recipe. On the other hand, ESGD using the ESGD baseline as the initial anchor has the loss of 1.899 and WER 16.5, also improves from the ESGD baseline. The ESGD baseline is a very strong baseline. Using anchors is able to further improve it. These results are summarized in Table 2.
5. Discussion

Parallel computing is a necessity for ESGD which is a powerful approach when there is strong computational power in hand. The reported experiments are carried out in a distributed manner where SGD and fitness evaluation are conducted on multiple GPUs in parallel, the number of which is roughly the number of individuals in the parent population (~100). Under this condition, the wall clock time of ESGD is about the same as that of an end-to-end vanilla SGD run.

Table 2: CE loss and WER of ESGD with anchors on BN50

<table>
<thead>
<tr>
<th>model</th>
<th>loss</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single SGD baseline</td>
<td>2.082</td>
<td>17.4</td>
</tr>
<tr>
<td>ESGD with anchor (SGD baseline)</td>
<td>1.935</td>
<td>16.5</td>
</tr>
<tr>
<td>ESGD baseline</td>
<td>1.916</td>
<td>16.4</td>
</tr>
<tr>
<td>ESGD with anchor (ESGD baseline)</td>
<td>1.899</td>
<td>16.3</td>
</tr>
<tr>
<td>ESGD with anchor (iterated anchor)</td>
<td>1.882</td>
<td>16.2</td>
</tr>
</tbody>
</table>

The reported experiments are carried out in a distributed manner where SGD and fitness evaluation are conducted on multiple GPUs in parallel, the number of which is roughly the number of individuals in the parent population (~100). Under this condition, the wall clock time of ESGD is about the same as that of an end-to-end vanilla SGD run.

A healthy population diversity is crucial for good performance of ESGD. It can prevent pre-mature convergence in ES and give good chances to produce better offsprings. We ensure this by using complementary optimizers and employing very short SGD and ES updates in each ESGD generation ($K_s = K_u = 1$). Moreover, the initial parent population is created using models with randomized weights. This seems to be hurtful in the sense of fitness but in the long run it helps to establish the diversity in the population and turns out to be better than just slightly perturb the initial anchor model to create the population.

Anchor models can help to accelerate the evolution process as the good “genes” of an anchor can spread out (with probability) to the next generations until it is replaced by another anchor with better “genes”. This is the motivation behind the design of the proposed algorithm. With model backoff, elitist and anchor switching, it is guaranteed that the fitness of the best model of the population will never degrade. This has been demonstrated in Figs.1 and 2. In the worst case, after the evolution no better model is found and the initial anchor survives as the best model.

The objective to optimize under ESGD is fitness which is CE loss in this work. In essence, there is no difference from other deep learning optimization problems except that it is population based and makes use of complementary optimizers and gradient-aware/gradient-free algorithms. From this perspective, it also has to cope with the training vs. generalization issue. We evaluate the fitness on the validation set, which is one way to help generalization. But observations are also made that there are cases where better fitness may not lead to better generalization (e.g. a narrow energy well in the landscape can give a low loss but may not generalize well [12][13]). Furthermore, although CE is correlated with WER, they are not exactly aligned. A better CE loss may not give rise to better WER. Looking forward, we hope to explore alternative fitness functions such as sequence training criteria sMBR [14] and MPE [15] which are better aligned with WER, the ultimate objective of ASR. Regularized fitness functions which can help generalization are another direction worth exploring.

Table 3 presents the experimental results of using ESGD with anchors on SWB300. The first two rows of the table provide the SGD and ESGD baselines, respectively, on SWB and CH test sets. When using ESGD baseline as the initial anchor, the CE loss improves from 1.551 to 1.536 after 20 generations. It slightly improves WER on CH (18.2% → 18.1%) but the WER on SWB stays the same (10.0% → 10.0%). If we start the ESGD all over with this best model, after another 40 generations with iterated anchors, the loss drops to 1.512 and the WER on CH further reduces to 17.7% and the WER on SWB reduces to 9.9%.

Table 3: CE loss and WER of ESGD with anchors on SWB300

<table>
<thead>
<tr>
<th>model</th>
<th>loss</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single SGD baseline</td>
<td>1.648</td>
<td>10.4</td>
</tr>
<tr>
<td>ESGD baseline</td>
<td>1.551</td>
<td>10.0</td>
</tr>
<tr>
<td>ESGD with anchor (ESGD baseline)</td>
<td>1.536</td>
<td>10.0</td>
</tr>
<tr>
<td>ESGD with anchor (iterated anchor)</td>
<td>1.512</td>
<td>9.9</td>
</tr>
</tbody>
</table>

6. Summary

In this paper, we investigated a population-based ESGD algorithm assuming some well-trained model exists. We use this model as an anchor in the population to accelerate the evolution and improve the quality of offsprings. We introduced anchor switching in the population and also an iterative way of applying ESGD with anchors to monotonically improve the best fitness of the population. We carried out experiments on BN50 and SWB300 and demonstrated the monotonic decrease of the CE loss. In some cases, it will also help improve the WERs over strong ESGD baselines.

7. References


