ON DESCENT SPECTRAL CG ALGORITHM FOR TRAINING RECURRENT NEURAL NETWORKS

I.E. Livieris\textsuperscript{1,†}, D.G. Sotiropoulos\textsuperscript{2}, P. Pintelas\textsuperscript{1,3}

No. TR09-04

\textsuperscript{1}University of Patras, Department of Mathematics, GR-265 04, Patras, Hellas. \textit{e-mail: livieris|pintelas@math.upatras.gr}
\textsuperscript{2}Ionian University, Department of Informatics, Curfu, Hellas. \textit{e-mail: dgs@ionio.gr}
\textsuperscript{3}Educational Software Development Laboratory, University of Patras, GR-265 04 Patras, Hellas.

† corresponding author
Abstract. In this paper, we evaluate the performance of a new class of conjugate gradient methods for training recurrent neural networks which ensure the sufficient descent property. The presented methods preserve the advantages of classical conjugate gradient methods and simultaneously avoid the usually inefficient restarts. Simulation results are also presented using three different recurrent neural network architectures in a variety of benchmarks.

1 Introduction

Recurrent neural networks (RNN) constitute an elegant way to provide better modeling accuracy compared to classical feedforward neural networks. Their main advantages is that they have the ability to deal with time varying input and output through their own natural temporal operation [9]. It is well known that recurrent neural networks can be trained in a similar way with the feedforward neural networks. The training process can be formulated as the minimization of an error function $E(w)$ that depends on the connection weights $w$ of the network. A traditional way to solve this problem is by an iterative gradient-based training algorithm which generates a sequence of weights $\{w\}_{k=0}^{\infty}$ using the recurrence

$$w_{k+1} = w_k + \eta_k d_k, \quad k = 0, 1, \ldots$$

(1)

where $k$ is the current iteration usually called epoch, $w_0 \in \mathbb{R}^n$ is a given starting point, $\eta_k > 0$ is the learning rate and $d_k$ is a descent search direction. In general, conjugate gradient methods constitute an elegant choice for efficiently training large neural networks due to their simplicity and their very low memory requirements since they do not require the evaluation of the Hessian matrix neither the impractical storage of an approximation of it.

Despite the theoretical and practical advantages of conjugate gradient methods, the main drawback of these methods is the use of restarting procedures in order to guarantee convergence [15]. Nevertheless, there is also a worry with restart algorithms that restarts may be triggered too often, abandoning the second-order derivative information; thus degrading the overall efficiency of the method [12]. Recently, Yu and Guan [19] proposed a new class of conjugate gradient methods which guarantee sufficient descent independent of the accuracy of the line search, avoiding thereby the usually inefficient restarting procedures.

In this work, we study the performance of this class of methods and we propose a new conjugate gradient algorithm for training recurrent neural networks. The proposed algorithm preserves the advantages of classical conjugate gradient methods and avoids of the inefficient restarts.

The remainder of this paper is organized as follows. Section 2 presents a brief summary of conjugate gradient methods and in Section 3 are presented the proposed conjugate gradient method and two truncated schemes for neural network training. Section 4, reports our experimental results and in Section 5 are presented our concluding remarks and our proposals for future research.

Notations. Throughout this paper $\| \cdot \|$ denotes the Euclidean norm, $\langle \cdot \rangle$ stands for the inner product and the gradient of the error function is indicated by $\nabla E(w^k) = g_k$.

2 Conjugate Gradient Methods

In conjugate gradient methods the basic idea for determining the search direction in Eq. (1) is the linear combination of the negative gradient vector at the current iteration with the
previous search direction, namely

\[
d_k = \begin{cases} 
  -g_k, & \text{if } k = 0; \\
  -g_k + \beta_k d_{k-1}, & \text{otherwise.}
\end{cases}
\]

In the literature, there have been proposed several choices for defining the scalar parameter \(\beta_k\) which give rise to distinct conjugate gradient methods. The most famous ones were proposed by Fletcher-Reeves [5] and Polak-Ribiére [14]. However, probably the most efficient choice was introduced by Perry [13], namely

\[
\beta_k^P = \frac{g_k^T (y_{k-1} - s_{k-1})}{y_{k-1}^T d_{k-1}}
\]

where \(s_{k-1} = w_k - w_{k-1}\) and \(y_{k-1} = g_k - g_{k-1}\). The convergence analysis are usually based on mild conditions which refer to the Lipschitz and boundedness assumptions and is closely connected with the sufficient descent property

\[
g_k^T d_k \leq -c \|g_k\|^2
\]

where \(c\) is a fixed constant. Hager and Zhang [8] have presented an excellent survey on different versions of nonlinear conjugate gradient methods, with special attention to their convergence properties.

Birgin and Martínez [2] incorporated the spectral gradient [17] in the conjugate gradient framework as follows

\[
d_k = \begin{cases} 
  -g_k, & \text{if } k = 0; \\
  -\frac{1}{\delta_k} g_k + \beta_k d_{k-1}, & \text{otherwise,}
\end{cases}
\]

where \(\delta_k\) is a scalar parameter usually defined as

\[
\delta_k = \frac{\langle y_{k-1}, s_{k-1} \rangle}{\langle s_{k-1}, s_{k-1} \rangle}
\]

which lies between the largest and the smallest eigenvalue of the average Hessian. The motivation for this selection constitutes in providing a two point approximation to the secant equation underlying Quasi-Newton methods [1]. Using a geometric interpretation for quadratic function minimization Birgin and Martínez suggested the following expression for defining the update parameter \(\beta_k\) in Eq. (5)

\[
\beta_k^{SP} = \frac{g_k^T (y_{k-1} - \delta_k s_{k-1})}{\delta_k y_{k-1}^T d_{k-1}}
\]

Clearly, in case \(\delta_k = 1\) this formula is reduced to the classical formula Eq. (3), introduced by Perry [13].

Unfortunately even with exact line search conjugate gradient methods, cannot guarantee to generate descent directions. Therefore, the use of a restarts are employed in order to guarantee convergence. A more sophisticated restarting criterion has been introduced by
Birgin and Martínez [2] which consists of testing if the angle between the current direction and the gradient is not acute enough, namely

\[ d_k^T g_k \leq 10^{-3} \|d_k\|_2 \|g_k\|_2 \]

In the literature there have been proposed several restarting criteria [3, 15, 18] with various performances.

### 3 Descent Conjugate Gradient Methods

In more recent works, Yu and Guan [19] proposed a new class of conjugate gradient methods which can guarantee the sufficient descent property (4) independent of the accuracy of the line search where the scalar parameter \( \beta_k \) is defined as

\[ \beta_k^{DP} = \beta_k^p - \frac{C\|y_{k-1} - s_{k-1}\|^2}{(y_{k-1}^T d_{k-1})^2} g_k^T d_{k-1} \]  

(8)

Parameter \( C \) essentially controls the relative weight between conjugacy and descent and in case \( C \geq 1/4 \) then all generated directions are descent directions.

Along this line, Yu et al. [20] motivated by the previous works [2, 7, 19], embedded the spectral gradient [1] in the descent conjugate methods framework

\[ \beta_k^{DSP} = \beta_k^p - \frac{C\|y_{k-1} - s_{k-1}\|^2}{\delta_k (y_{k-1}^T d_{k-1})^2} g_k^T d_{k-1} \]  

(9)

Obviously, if \( \delta_k = 1 \) then the above formula will reduce to the corresponding descent conjugate gradient formula Eq. (8).

In order to ensure global convergence of the conjugate gradient methods for general nonconvex functions, it has been proposed to truncate \( \beta_k^{DSP} \) by restricting the lower bound of the update parameter of being nonnegative [6, 16]. However by restricting \( \beta_k^{DSP} \) to be nonnegative, results the iterates may differ significantly from those of (2)-(9) and the convergence speed may be reduced. Thus, Yu et al. [20] proposed two modified schemes in which the lower bound on \( \beta_k \) is dynamically adjusted, in order to make the lower bound small as the iterates converge. In particular, they proposed two variants for selecting the truncated update parameter \( \beta_k^{DSP+} \). The first variant of the truncated scheme uses the update parameter

\[ \beta_k^{DSP+1} = \max \left\{ \beta_k^{DSP}, \frac{C\|y_{k-1} - s_{k-1}\|^2}{\delta_k (y_{k-1}^T d_{k-1})^2} |g_k^T s_{k-1}| \right\} \]  

(10)

while the second one is based on a scheme that was first introduced by Hager and Zhang [7], namely

\[ \beta_k^{DSP+2} = \max \left\{ \beta_k^{DSP}, \frac{1}{\delta_k \|d_{k-1}\| \min \{\xi, \|g_{k-1}\|\}} \right\} \]  

(11)
where $\xi$ is some positive constant. Notice that when $\|g_{k-1}\|$ tends to zero as $k$ grows, $\xi_k$ tend to $-\infty$ when $\|d_{k-1}\|$ is bounded [7]. In that case $\beta_k^{DSP_2^+}$ will reduce to the descent spectral Perry formula Eq. (9).

At this point, we summarize the new training algorithm.

**Descent Spectral Conjugate Gradient Training Algorithm**

(1.) Initiate $w_0$, $0 < \sigma_1 < \sigma_2 < 1$ and $E_G$; set $k = 0$.

(2.) If $(E(w_k) < E_G)$ terminate.

(3.) Compute the descent direction

$$d_k = \begin{cases} -\nabla E(w_0), & \text{if } k = 0; \\ -\frac{1}{\delta_k} \nabla E(w_k) + \beta_k d_{k-1}, & \text{otherwise} \end{cases}$$

where the update parameter is defined by one of (9), (10) or (11).

(4.) Find a step length $\eta_k$ using the following line search. For $0 < \sigma_1 < \sigma_2 < 1$ at each iteration, choose the step length satisfying the strong Wolfe line search conditions

$$E(w_k + \eta_k d_k) - E(w_k) \leq \sigma_1 \eta_k \langle \nabla E(w_k) d_k \rangle$$

$$|\langle \nabla E(w_k + \eta_k d_k), d_k \rangle| \leq -\sigma_2 \langle \nabla E(w_k), d_k \rangle$$

(5.) Update the weights

$$w_{k+1} = w_k + \eta_k d_k$$

(6.) Set $k = k + 1$ and goto Step 2.

**4 Experimental Results**

In this section we will present experimental results in order to evaluate the performance of the proposed conjugate gradient algorithm in three famous benchmarks acquired by the UCI Repository [11]. Subsequently, we briefly describe each problem and present the performance comparison between: the descent spectral Perry’s conjugate gradient (DSP) with the spectral Perry’s conjugate gradient (SP) of Birgin and Martínez [2]. Moreover we test the numerical performance of two different truncation strategies for the update parameter $\beta_k$. So we evaluate the two versions of the truncate spectral Perry conjugate gradient ($DSP_2^+$): the first one ($DSP_2^+$) uses the update parameter $\beta_k^{DSP_2^+}$ defined in Eq. (10) and the other one ($DSP_2^+$) uses the update parameter $\beta_k^{DSP_2^+}$ defined in Eq. (11). We have chosen three classical first-order recurrent neural architectures:
• Feedforward Time Delay (FFTD),

• Nonlinear Autoregressive Network with Exogenous Inputs (NARX)

• Elman’s Layered Recurrent Networks (ELRN)

For all algorithms the heuristic parameters were set as \( \sigma_1 = 10^{-4}, \sigma_2 = 0.5 \) and \( \xi = 0.01 \) for all experiments [19]. All networks have received the same sequence of input patterns and the initial weights were initiated using the Nguyen-Widrow method. All algorithms were evaluated using the performance profiles proposed by Dolan and Morè [4] for measuring their efficiency and robustness in terms of computational cost (i.e. function/gradient evaluations). The implementation has been carried out using Matlab version 7, based on the SCG code of Birgin and Martínez [2].

4.1 SPECT Heart Problem

This data set contains data instances derived from cardiac Single Proton Emission Computed Tomography (SPECT) images from the University of Colorado [11]. This is also a binary classification task, where patients’ heart images are classified as normal or abnormal. The network architectures for this medical classification problem constitute of 1 hidden layer with 6 neurons and an output layer of 1 neuron. The termination criterion is set to \( E_G \leq 0.1 \) within the limit of 1000 epochs.

![Figure 1: Log_{10} scaled performance profiles for the spect heart problem.](image)

Figure 1 presents the performance profiles for each recurrent neural network architecture for the spect heart problem. DSP is much more robust and efficient the SP since the curve of the former lies above the curve of the latter. Furthermore, \( \text{DSP}^+_1 \) and \( \text{DSP}^+_2 \) exhibit the highest probability of being the optimal solver with the latter exhibiting slightly better performance.
4.2 Iris Problem

This benchmark is perhaps the most best known to be found in the pattern recognition literature [11]. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. The network architectures constitute of 1 hidden layer with 7 neurons and an output layer of 3 neurons. The termination criterion is set to $E_G \leq 0.01$ within the limit of 1000 epochs and all networks were tested using 10-fold cross-validation.

![Performance profiles for FFTD, ELRN, and NARX](image)

Figure 2: Log$_{10}$ scaled performance profiles for the iris problem.

In Figure 2 are presented the performance profiles for the iris problem investigating the efficiency and robustness of each training method. DSP$_1^+$ and DSP$_2^+$ exhibit the best performance regarding all recurrent neural network architectures. Moreover, the interpretation in Figure 2 implies that the sufficient descent property led to a significant improvement of DSP, DSP$_1^+$ and DSP$_2^+$ over the SP method since the curves of the former lie above the curve of the latter.

4.3 Escherichia coli Problem

This problem is based on a drastically imbalanced data set of 336 patterns and concerns the classification of the E. coli protein localization patterns into eight localization sites. E. coli, being a prokaryotic gram-negative bacterium, is an important component of the biosphere. Three major and distinctive types of proteins are characterized in E. coli: enzymes, transporters and regulators. The largest number of genes encodes enzymes (34%) (this should include all the cytoplasm proteins) followed by the genes for transport functions and the genes for regulatory process (11.5%) [10]. The network architectures constitute of 1 hidden layer with 16 neurons and an output layer of 8 neurons. The stopping criterion is set to $E_G \leq 0.02$ within the limit of 2000 epochs. In all simulations the neural networks were tested using 4-fold cross-validation.

Figure 3 illustrates the performance profiles for the ecoli problem investigating the performance of each training method. DSP is much more efficient and robust than SP regarding all recurrent neural network architectures. Furthermore, DSP$_1^+$ and DSP$_2^+$ perform similarly
and exhibit the best performance since they present the highest probability of being the optimal solver.

5 Conclusions

In this work, we evaluate the performance of the proposed conjugate gradient algorithm for training recurrent neural networks. From the rigorous analysis of the simulation results is strongly demonstrated the sufficient descent property led to a significant improvement of the DSP method over SP method which provides faster, more stable and reliable convergence.

Our future work is concentrated on performing a large scale study on several training methods for large neural networks and investigate the effect of incorporating a nonmonotone strategy in our framework.

References


