

A New Heuristic Global Algorithm for Automatic Speech Recognition Using Recurrent Neural Networks

M. Bianchini*, S. Fanelli**, M. Gori*, M. Maggini*, and M. Protasi**

* *Dipartimento di Sistemi e Informatica, Università di Firenze*

** *Dipartimento di Matematica, Università di Roma “Tor Vergata”*

1 Introduction

Recurrent Neural Networks (RNN) are an efficient tool for the solution of problems of Automatic Speech Recognition. In [1] was described a new approach for Isolated Word Recognition (IWR) problems founded on the application of the Reduced Gradient (RG) algorithm in a RNN using locally recurrent connections [2]. The preliminary investigation performed in [1] showed that RG is a highly convenient local optimisation algorithm in comparison with classical Back Propagation Through Time (BPTT) methods [3]. The implementation of RG guarantees the convergence to a Kuhn-Tucker (KT) point which, however, is in general neither the global minimum nor a satisfactory suboptimal solution of the problem.

Since a KT point is an attractor for any local optimisation algorithm, it is necessary to implement periodically a procedure to jump out of the neighbourhood of the undesirable solutions. In this paper we have experimented a new heuristic method based on a combined use of RG and a deterministic global algorithm for unconstrained optimisation, acronymed TRUST, recently proposed in [4, 5].

The first numerical experiences of this method, named here RGTRUST1, have evidenced promising performances. As a matter of fact, RGTRUST1 is able both to improve the computational efficiency obtained by a standard BPTT algorithm, and to discover *new better solutions*, associated to different KT points. More precisely, RGTRUST1 overcomes the typical disadvantages of the local optimisation procedures, which are easily trapped into the basin of attraction of the nearest KT point, computed by the algorithm.

2 Reduced Gradient Algorithm for IWR Problems

In [6] was presented a new model for IWR by using a RNN with a K-L (a priori Knowledge and Learning) architecture. This approach is based upon an explicit knowledge of the IW by a Finite State Automaton (FSA); consequently, the learning algorithm is performed with a direct translation of the FSA into the

neural connections.

Since it can be proved (see e.g. [6] and [1], sect. II) that the automaton rules are realised in terms of linear inequalities on the weights, the learning process can be modelised as a Non-Linear Programming Problem with Linear Constraints (NLPLC).

Let N be the set of neurons of the RNN. Moreover, let U denote the set of external network inputs and let V denote the set of output neurons. The activation functions $a_i(t)$ and the corresponding output values $x_i(t)$ are defined for each neuron $i \in N$ as follows:

$$\begin{aligned} a_i(t) &= \sum_{j \in N} w_{ij} x_j(t-1) + \sum_{k \in U} w_{ik} u_k(t), \\ x_i(t) &= f[a_i(t)] = \tanh\left(\frac{a_i(t)}{2}\right) \end{aligned}$$

where, $\forall i, j \in N, \forall k \in U$, w_{ij} indicates the weight on the connection (j, i) and $u_k(t)$ is the k -th external input.

The weights' optimisation is performed by solving the following problem (see [1] for more details):

$$\begin{cases} \min E \\ \hat{x}_{i,r}^b(t) I_{i,r}(t) - \sigma_{i,r} I_i^* > 0, \quad i = 1, \dots, n, r \in \mathfrak{R} \end{cases} \quad (1)$$

where, $\forall i = 1, \dots, n, \forall r \in \mathfrak{R}$

$$\begin{aligned} E &= \frac{1}{2} \sum_{p \in P} \sum_{i \in V} [d_{i,p} - x_i(t)]^2, \\ I_{i,r}(t) &= \sum_{j \in N} w_{ij} (1 - \delta_{i,j}) x_{j,r}^b(t) + \sum_{j \in U} w_{ij} u_{j,r}^b(t), \\ I_i^* &= \sqrt{w_{ii}(w_{ii} - 2)} + \log\left(w_{ii} - 1 - \sqrt{w_{ii}(w_{ii} - 2)}\right), \\ \hat{x}_{i,r}^b(t) &= \text{sign}[\hat{x}_{i,r}(t)], \\ \hat{x}_{i,r}(t) &= \text{the next state after } x_{i,r}(t), \\ \sigma_{i,r} &= \begin{cases} +1 & \text{if a boolean state switching is required} \\ -1 & \text{otherwise} \end{cases} \end{aligned}$$

being P the set of patterns, $d_{i,p}$ the desired target of output unit i for the pattern p , and R the set of rules associated to the FSA.

By substituting to each variable w_{ij} the pairing (y_{ij}, z_{ij}) such that:

$$\begin{aligned} w_{ij} &= y_{ij} - z_{ij}, \\ y_{ij} &\geq 0, \\ z_{ij} &\geq 0, \end{aligned} \quad (2)$$

the minimisation problem (1) becomes a NLPLC.

Classical procedures to solve NLPLC in the *convex case* are the Reduced Gradient (RG) method, due to Wolfe, and the Projected Gradient algorithm, due to Rosen [7]. RG is highly recommended for its major efficiency from an operational point of view.

In this work, we have experimented an extension to the *non-convex case* of the original RG-algorithm, which is particularly suited for the IWR problem. The classical RG method is implemented to solve the problem

$$\begin{cases} \min F(\underline{x}) \\ A\underline{x} = \underline{b} \\ \underline{x} \geq 0 \end{cases} \quad (3)$$

The fundamental idea of the RG algorithm is based on the search of a pair $(\underline{x}^*, \underline{\lambda}^*)$, named KT point, that satisfies the following conditions:

$$\begin{cases} \nabla F(\underline{x}^*) + A'\underline{\lambda}^* \geq 0 \\ (\nabla F(\underline{x}^*) + A'\underline{\lambda}^*, \underline{x}^*) = 0 \\ A\underline{x}^* = \underline{b} \\ \underline{x}^* \geq 0 \end{cases} \quad (4)$$

If F is convex, a point \underline{x}^* satisfying (4) is an optimal solution of (3). If F is not convex, as in the case of the error function E , \underline{x}^* is in general only a suboptimal solution. In the latter case, it is not known in the literature whether the KT point \underline{x}^* is a constrained local minimum or not (see [1] for a detailed description of the RG algorithm).

In order to obtain an effective implementation of the RG algorithm, the choice of a good starting solution $\underline{x}^{(0)}$ is a crucial point. For this purpose in [8] was suggested an efficient procedure to determine a feasible solution of (1), named K-algorithm (see [8], sect. III).

Let M and \underline{t} denote respectively the matrix and the vector such that the linear constraints of problem (1) can be written in the form $M\underline{w} \leq \underline{t}$. By using (2) and suitable slack variables v_{ij} the optimisation problem can be written in the following form:

$$\begin{cases} \min E(\underline{y}, \underline{z}, \underline{v}) \\ \hat{M} \cdot \begin{bmatrix} \underline{y} \\ \underline{z} \\ \underline{v} \end{bmatrix} = \underline{t} \end{cases} \quad (5)$$

where $\hat{M} = [M, -M, I]$. By applying the RG algorithm we are able to find a KT point for problem (5).

Three different situations may occur:

- The KT point is either the optimal solution of the problem (5) or a constrained local minimum accomplishing a satisfactory value of the error function.
- The KT point is a constrained local minimum but the corresponding solution is unsuitable from an operational point of view.
- The KT point is *not* a constrained local minimum (f.i. a saddle point).

In the former case we do not need further investigations, while in the latter ones it is necessary to escape from the neighbourhood of the KT point before continuing the optimisation process. For such purpose, in this work we have implemented a procedure directly derived by the theory of terminal repellers.

3 Terminal Repellers for Constrained Optimisation Problems

A new deterministic algorithm for unconstrained global optimisation, named *TRUST*, was introduced in [5]. This method formulates the problem of learning as the solution of a dynamical system incorporating *terminal repellers* [5] together with a special sub-energy tunneling function.

A terminal repeller is an unstable equilibrium point x violating the Lipschitz condition such that any transient starting at x_0 infinitesimally close to x will escape x in a finite time. The TRUST algorithm is based on the application of gradient descent to a combined function of the form:

$$C(W, W^*) = C_{sub}(W, W^*) + C_{rep}(W, W^*).$$

W^* is a starting solution achieving a “subenergy limit”, $C_{sub}(W, W^*)$ is a non linear but monotonic transformation of the error function $E(W)$, preserving all properties relevant for optimisation, and $C_{rep}(W, W^*)$ is a “repeller energy term”, containing a parameter ξ (the *power of the repeller*).

The optimisation process switches between a “tunneling phase” and a gradient descent phase, depending on the relative values of $E(W^*)$ and $E(W)$ in the neighbourhood of W^* .

Unfortunately, using this interesting and ingenious procedure, the convergence to a global minimum is guaranteed *only for functions of one variable* and the efficiency in the multi-dimensional case may be very low in general.

However, in this paper we show that, for a suitable choice of the power of the repeller ξ , TRUST can be efficiently used in conjunction with the RG method in order to escape from the basin of attraction of the KT point. More precisely, during the learning process performed by RG, the implementation of TRUST takes place whenever a suitable set of “local stopping criteria” is satisfied. The latter set of conditions has the aim of establishing the *right point* of restarting the error dynamics, typically in a small neighbourhood of the KT point. TRUST is utilised just for a single iteration in order to investigate the possibility of descending into a lower valley, by satisfying the linear constraints; RG is then implemented again for the next iterations, performing a new phase of local optimisation. For this reason the whole procedure is named RGTRUST1.

4 Numerical Results

In [6, 8] was implemented a skilled BPTT algorithm, named TRENNS, for the training of RNN with a K-L architecture. This justifies the choice of TRENNS as a comparison tool in IWR problems. General BPTT packages, like MUME [9], are not competitive since they are suitable only for fully connected RNN.

Two examples are reported here in order to show the performances of the new proposed algorithm. We first notice that the efficiency of RGTRUST1, for any local optimisation phase, is outstanding. Moreover, as clearly confirmed by

References

- [1] S. Fanelli, A. Manzo, and M. Protasi, "On the Application of the Reduced Gradient Algorithm in a Recurrent Neural Network Environment for Automatic Speech Recognition," in *Irish Neural Networks Conference '94*, (University College, Dublin), pp. 55–60, September, 12–13, 1994.
- [2] A. D. Back and E. A. Wan (organisers), "Neural Network Architectures with Time Delay Connections for Non Linear Signal Processing: Theory and Applications," in *NIPS*94 Workshop*, 1994.
- [3] R. Williams and D. Zipser, "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks," *Neural Computation*, vol. 1, pp. 270–280, 1989.
- [4] J. Barhen, J. W. Burdick, and B. C. Cetin, "Global Descent Replaces Gradient Descent to Avoid Local Minima Problem in Learning with Artificial Neural Networks," in *ICNN93* (icnn, ed.), vol. 2, (San Francisco (CA)), pp. 836–842, March-April 1993.
- [5] J. Barhen, J. W. Burdick, and B. C. Cetin, "Terminal Repeller Unconstrained Subenergy Tunneling (TRUST) for Fast Global Optimization," *Journal of Optimization Theory and Applications*, vol. 77, no. 1, pp. 97–126, 1993.
- [6] P. Frasconi, M. Gori, M. Maggini, and G. Soda, "Unified Integration of Explicit Rules and Learning by Example in Recurrent Networks," *IEEE Transactions on Knowledge and Data Engineering*, 1993. (in press).
- [7] M. Minoux, *Mathematical Programming: Theory and Algorithms*. Wiley and Sons.
- [8] P. Frasconi, M. Gori, M. Maggini, and G. Soda, "A Unified Approach for Integrating Explicit Knowledge and Learning by Examples in Recurrent Networks," in *International Joint Conference on Neural Networks*, (Seattle WA), pp. 811–816, 1991.
- [9] "Multi Module Computing Environment (MUME)," (NSW, 2006 AUSTRALIA), SEDAL, Sidney University of Electrical Engineering.