On consensum

Giulio Galvan

26 settembre 2015

Algorithm 1 shows a Gauss-Seidel like decomposition method which optimizes a function $f(\cdot)$ w.r.t a block of variables i in a iteratively fashion. It can be shown that the algorithm converges if $\mathcal{L}_0$ is compact.

Algorithm 1: Gauss-Seidel like decomposition method

Data:
$x^0 \in \mathbb{R}^m$: candidate solution
$\lambda_i > 0, \quad i = 1, \ldots, m$
$k \leftarrow 0$
while *stop criterion* **do**
  for $i = 1, ..., m$ **do**
    $d_i^k \leftarrow -\nabla_i f(x^k)$
    compute α_i^k with line search
    $x_i^{k+1} = x_i^k + \alpha_i^k d_i^k$
  end
  $k \leftarrow k + 1$
end

In the case fo RNNs we would, ideally, in a consesum inspired way, solve

$$\begin{aligned} \min_{W, W^i} \quad & f(W, W^i, i = 1 \ldots, T) \\ \text{subject to} \quad & W_i = W, \quad i = 1, \ldots, T \end{aligned},$$

considering only W^{rec} for easiness. The Lagrangian can then be written as:

$$\min_{\tilde{W}, W^i} g(\cdot) = f(\tilde{W}, W^i, i = 1 \ldots, T) + \lambda(W^i - \tilde{W}) \tag{1}$$

The blocks of variables are, hence, $W^i, i = 1, ..., T$ where T is the length of the sequence, and $\tilde{W}$ the master variable. However we cannot have T of such matrix in memory so we have to devise a modification of the algorithm like I tried to do in Algorithm 2

The idea is not to store all the matrices, computing gradients for each block separately but using an iteratively update rule, like is done at line 7.

Algorithm 2: RNN consensum-decomposition method

Data:

$W^i = \tilde{W} = W^0, \quad i = 1, \ldots, T$: candidate solution

$k \leftarrow 0$

while *stop criterion* **do**

    for $i = 1, ..., m$ **do**

        $d_i^k \leftarrow -\nabla_i g(x^k)$

        compute α_i^k with line search

        $\tilde{W} = W_i^k + \alpha_i^k d_i^k$

        $W_j^{k+1} = \tilde{W}, \quad j = 1, \ldots, i$

    end

    $k \leftarrow k + 1$

end