Multilevel Optimization Methods for the Training of Artificial Neural Networks

E. Ricciettti (LIP-ENS, Lyon)

Joint work with: H. Calandra (Total)
S. Gratton (IRIT-INP, Toulouse)
X. Vasseur (ISAE-SUPAERO, Toulouse)

CSE 2021

1 - 5 March 2021

Context

We consider large-scale nonlinear unconstrained optimization problems:

$$\min_{x} f(x), \quad f : \mathbb{R}^n \to \mathbb{R}$$

Context

We consider large-scale nonlinear unconstrained optimization problems:

$$\min_x f(x), \quad f : \mathbb{R}^n \to \mathbb{R}$$

Classical iterative second-order optimization methods:

$$f(x_k + s) \simeq T_2(x_k, s)$$

with $T_2(x_k, s)$ Taylor model of order 2:

$$T_2(x_k, s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T \nabla^2 f(x_k) s$$

Context

We consider large-scale nonlinear unconstrained optimization problems:

$$\min_x f(x), \quad f : \mathbb{R}^n \to \mathbb{R}$$

Classical iterative second-order optimization methods:

$$f(x_k + s) \simeq T_2(x_k, s)$$

with $T_2(x_k, s)$ Taylor model of order 2:

$$T_2(x_k, s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T \nabla^2 f(x_k) s$$

At each iteration we compute a step s_k to update the iterate:

$$\min_s m_k(x_k, s) = T_2(x_k, s) + r(\lambda_k), \qquad \lambda_k > 0$$

$r(\lambda_k)$ regularization term, $x_{k+1} = x_k + s_k$.

Examples

- Trust region method (TR):

$$m_k(x_k, s) = T_2(x_k, s) + \frac{\lambda_k}{2} \|s\|^2$$

- Adaptive Cubic Regularization method (ARC):

$$m_k(x_k, s) = T_2(x_k, s) + \frac{\lambda_k}{3} \|s\|^3$$

Examples

- Trust region method (TR):

$$m_k(x_k, s) = T_2(x_k, s) + \frac{\lambda_k}{2}\|s\|^2$$

- Adaptive Cubic Regularization method (ARC):

$$m_k(x_k, s) = T_2(x_k, s) + \frac{\lambda_k}{3}\|s\|^3$$

- Extension to higher-order models ($q > 2$):

$$m_{q,k}(x_k, s) = T_q(x_k, s) + \frac{\lambda_k}{q+1}\|s\|^{q+1},$$

Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models, E. G. Birgin, J. L. Gardenghi, J. M. Martínez, S. A. Santos and Ph. L. Toint, 2017

Bottleneck: Subproblem solution

Solving

$$\min_s T_q(x_k, s) + \frac{\lambda_k}{q+1}\|s\|^{q+1}$$

represents greatest cost per iteration, which depends on the size of the problem.

$$\Downarrow$$

Multilevel methods!

We propose a family of scalable multilevel methods using high-order models.

Hierarchy of problems

- $\{f^l(x^l)\}$, $f^l : \mathbb{R}^{n_l} \to \mathbb{R}$
- $n_l < n_{l+1}$
- f^l is cheaper to optimize compared to f^{l+1}

Outline

- Part I: multilevel extension of iterative high-order optimization methods
    - global convergence
    - worst-case complexity
    - local convergence rate

Outline

- Part I: multilevel extension of iterative high-order optimization methods
  - global convergence
  - worst-case complexity
  - local convergence rate
- Part II: use of the multilevel methods for the training of artificial neural network
  - multilevel methods in the literature used just for problems with a geometrical structure

Part I

Multilevel extension of iterative high-order optimization methods

One level strategy

At level $l = l_{\max}$, let x_k^l be the current approximation. We look for a correction s_k^l to define the new approximation $x_{k+1}^l = x_k^l + s_k^l$.

x_k^l

One level strategy

At level $l = l_{\max}$, let x_k^l be the current approximation. We look for a correction s_k^l to define the new approximation $x_{k+1}^l = x_k^l + s_k^l$.

$$x_k^l \xrightarrow{\qquad T_q^l \qquad} x_{k+1}^l = x_k^l + s_k^l$$

Multilevel strategy

Two choices:

1. minimize regularized Taylor model, get s_k^l,
2. choose lower level model μ^{l-1} (built from f^{l-1}):

$$x_k^l \xrightarrow{\qquad T_q^l \qquad} x_{k+1}^l = x_k^l + s_k^l$$

Multilevel strategy

Two choices:

1. minimize regularized Taylor model, get s_k^l,
2. choose lower level model μ^{l-1} (built from f^{l-1}):

Multilevel strategy

Two choices:

1. minimize regularized Taylor model, get s_k^l,
2. choose lower level model μ^{l-1} (built from f^{l-1}):

$$x_k^l$$

Multilevel strategy

Two choices:

1. minimize regularized Taylor model, get s_k^l,
2. choose lower level model μ^{l-1} (built from f^{l-1}):

$$\begin{array}{c} x_k^l \\ \Big\downarrow R^l \\ R^l x_k^l := x_{0,k}^{l-1} \end{array}$$

Multilevel strategy

Two choices:

1. minimize regularized Taylor model, get s_k^l,
2. choose lower level model μ^{l-1} (built from f^{l-1}):

$$\begin{array}{ccc} x_k^l & & \\ \Big\downarrow{\scriptstyle R^l} & & \\ R^l x_k^l := x_{0,k}^{l-1} & \xrightarrow{\quad \mu^{l-1} \quad} & x_{*,k}^{l-1} \end{array}$$

Multilevel strategy

Two choices:

1. minimize regularized Taylor model, get s_k^l,
2. choose lower level model μ^{l-1} (built from f^{l-1}):

$$
\begin{tikzcd}
x_k^l \arrow[d, "R^l"'] & & x_{k+1}^l = x_k^l + s_k^l \\
R^l x_k^l := x_{0,k}^{l-1} \arrow[rr, "\mu^{l-1}"] & & x_{*,k}^{l-1} \arrow[u, "s_k^l = P^l(x_{*,k}^{l-1} - x_{0,k}^{l-1})"']
\end{tikzcd}
$$

Multilevel strategy

Two choices:

1. minimize regularized Taylor model, get s_k^l,
2. choose lower level model μ^{l-1} (built from f^{l-1}):

$$
\begin{tikzcd}
x_k^l \arrow[d, "R^l"'] & & x_{k+1}^l = x_k^l + s_k^l \\
R^l x_k^l := x_{0,k}^{l-1} \arrow[rr, "\mu^{l-1}"] & & x_{*,k}^{l-1} \arrow[u, "s_k^l = P^l(x_{*,k}^{l-1} - x_{0,k}^{l-1})"']
\end{tikzcd}
$$

- The lower level model is cheaper to optimize.
- The procedure is recursive: more levels can be used.

Theoretical results

Multilevel q-th order method

For a multilevel method of order q, we have proved its:

- Global convergence: $\lim_{k\to\infty} \nabla f(x_k) = 0$
- Complexity: $\|\nabla f(x_k)\| \leq \epsilon$ in at most $O(\epsilon^{-\frac{(q+1)}{q}})$ iterations
- Local convergence: order of convergence q, i.e., $\exists c > 0$ such that $\lim_{k\to\infty} \frac{\|x_{k+1}-x_*\|}{\|x_k-x_*\|^q} \leq c$ $\rightarrow$ NEW!

Numerical example: solution of PDEs (I)

$$\begin{cases} -\Delta u(z) + e^{u(z)} = g(z) & \text{in } \Omega \subset \mathbb{R}^d, \\ u(z) = 0 & \text{on } \partial\Omega, \end{cases}$$

The following nonlinear minimization problem is then solved:

$$\min_{u \in \mathbb{R}^{n^d}} \frac{1}{2} u^T A u + \| e^{u/2} \|^2 - g^T u,$$

which is equivalent to the system $Au + e^u = g$.

- Coarse approximations: coarser discretization of the problem (2^d times lower dimension).

4 levels methods of order $q = 2, 3$

		$n = 1024$		$n = 4096$	
	$d = 2,\ q = 2$	AR2	MAR2	AR2	MAR2
$\bar{u}_1$	it_T/it_f	11/11	7/2	23/23	15/4
	save		**2.2**		**4.1**
$\bar{u}_2$	it_T/it_f	27/27	13/4	56/56	22/6
	save		**3.9**		**6.1**

4 levels methods of order $q = 2, 3$

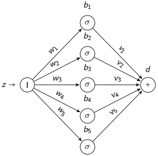
		$n = 1024$		$n = 4096$	
	$d = 2,\ q = 2$	AR2	MAR2	AR2	MAR2
$\bar{u}_1$	it_T / it_f	11/11	7/2	23/23	15/4
	save		**2.2**		**4.1**
$\bar{u}_2$	it_T / it_f	27/27	13/4	56/56	22/6
	save		**3.9**		**6.1**

		$n = 256$		$n = 512$	
	$d = 1,\ q = 3$	AR3	MAR3	AR3	MAR3
$\bar{u}_1$	it_T / it_f	7/7	9/2	18/18	15/2
	save		**2.5**		**4.3**
$\bar{u}_2$	it_T / it_f	23/23	14/1	34/34	20/5
	save		**4.1**		**4.4**

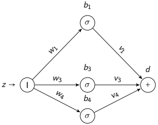
Part II

Use of the multilevel methods for the training of artificial neural networks

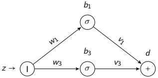
How to exploit multilevel method for training of ANNs?



$$R_1 \Downarrow P_1 \Uparrow$$



$$R_2 \Downarrow P_2 \Uparrow$$



Large-scale problem

How to build the hierarchy of problems? The variables to be optimized are the network's weights:
NO evident geometrical structure to exploit!

Algebraic multigrid (AMG)

Ruge and Stueben C/F splitting for $Ax = b$

- Two variables i, j are said to be *coupled* if $a_{i,j} \neq 0$.
- We say that a variable i is strongly coupled to another variable j, if $-a_{i,j} \geq \epsilon \max_{a_{i,k}<0} |a_{i,k}|$ for a fixed $0 < \epsilon < 1$, usually $\epsilon = 0.25$.

Prolongation-Restriction operators

$P = [I; \Delta]$, $R = P^T$, automatically built.

Algebraic multigrid (AMG)

Ruge and Stueben C/F splitting for $Ax = b$

- Two variables i, j are said to be *coupled* if $a_{i,j} \neq 0$.
- We say that a variable i is strongly coupled to another variable j, if $-a_{i,j} \geq \epsilon \max_{a_{i,k}<0} |a_{i,k}|$ for a fixed $0 < \epsilon < 1$, usually $\epsilon = 0.25$.

Prolongation-Restriction operators

$P = [I; \Delta]$, $R = P^T$, automatically built.

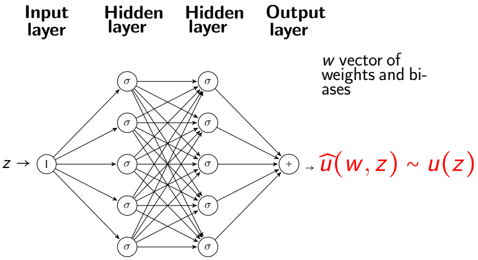
Which matrix to use?

Second order method:

$$T_2(x_k, s) = f(x_k) + s^T \nabla f(x_k) + \frac{1}{2} s^T \nabla^2 f(x_k) s$$

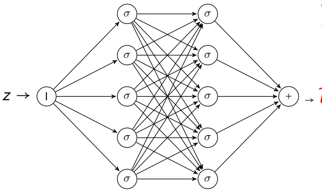
Numerical example: solution of PDEs (II)

1D case: $D(z, u(z)) = g(z),\ z \in (a, b) \quad u(a) = A,\ u(b) = B$



Numerical example: solution of PDEs (II)

1D case: $D(z,u(z)) = g(z),\ z \in (a,b) \quad u(a) = A,\ u(b) = B$



Training problem: find the network weights w by minimizing

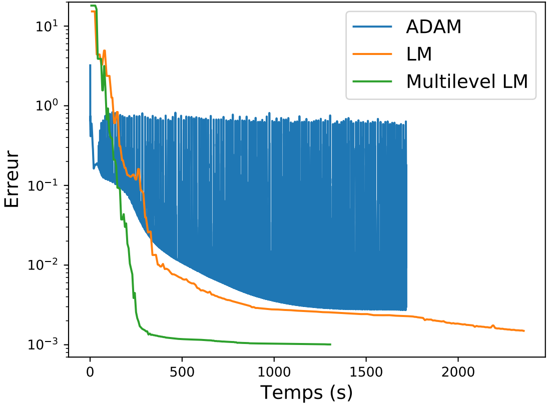
$$\min_w \frac{1}{2T}\sum_{t=1}^{T}\Big(\underbrace{D(z,\widehat{u}(w,z_t)) - g(z_t)}_{\text{Equation residual}}\Big)^2 + \lambda_p\Big(\underbrace{(\widehat{u}(w,a)-A)^2 + (\widehat{u}(w,b)-B)^2}_{\text{Boundary conditions}}\Big)$$

Least-squares problem → multilevel Levenberg-Marquardt method

Solution of PDEs: Numerical example

Poisson's equation
(2D, $n = 4096$)

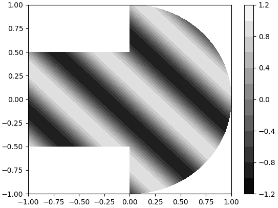
Method	ADAM	1 level	2 levels
Iterations	10000	200	200

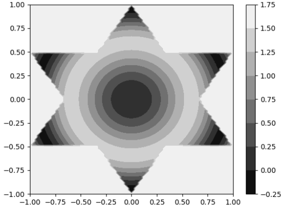


Numerical results on difficult domains ($n = 4096$)

Left: $-\Delta u + \nu^2 u = g_1,\ u(x,y) = \sin(\nu(x+y))\ \nu = 3$

Right: $-\Delta u + \nu u^2 = g_1,\ u(x,y) = (x^2+y^2) + \sin(\nu(x^2+y^2)),\ \nu = \frac{1}{2}$





	iter	RMSE	savings			iter	RMSE	savings		
			min	avg	max			min	avg	max
1 level	395	10^{-4}				1408	10^{-3}			
2 levels	110	10^{-4}	1.3	5.6	10.0	1301	10^{-3}	1.2	1.9	2.4

Conclusions and perspectives

- Theoretical contribution: We have presented a class of multilevel high-order methods for optimization and proved their global and local convergence and complexity.
- Practical contribution: We have got further insight on the methods proposing a AMG strategy to build coarse representations of the problem to use some methods in the family for the training of artificial neural networks.
- Perspective: Hessian-free method. Make it a competitive training method: the method needs to compute and store the Hessian matrix (for step computation and to build transfer operators): still too expensive for very large-scale problems.

Thank you for your attention!

On a multilevel Levenberg-Marquardt method for the training of artificial neural networks and its application to the solution of partial differential equations, H. Calandra, S. Gratton, E. Riccietti X. Vasseur, SIOPT, 2021.

On high-order multilevel optimization strategies, H. Calandra, S. Gratton, E. Riccietti X. Vasseur, OMS, 2020.

On iterative solution of the extended normal equations, H. Calandra, S. Gratton, E. Riccietti X. Vasseur, SIMAX, 2020.