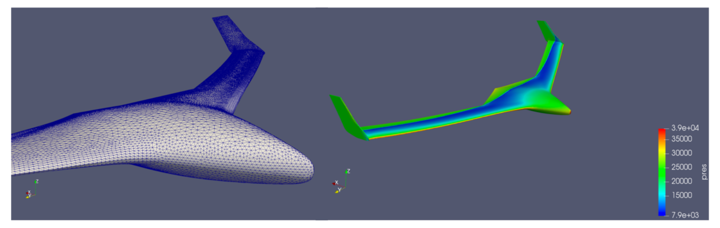
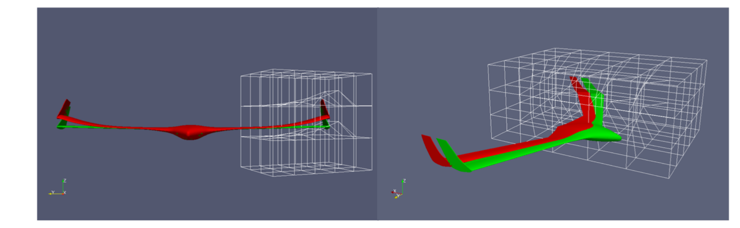
Computing Nonlinear Active Subspaces for Highly Parameterized Optimization Problems using Autoencoders





Gabriele Boncoraglio, gbonco@stanford.edu

Overview

A novel **dimensionality reduction approach** related to solving **highly parameterized optimization problems** is introduced specifically for problems involving running expensive computer simulations such as aircraft design.

Issue: A method used for dimensionality reduction is Active Subspace (AS); it consists in a **linear approximation of the subspace** and it is not always feasible for nonlinear problems.

Objective: Building an autoencoder using Neural Networks for computing a **nonlinear approximation of the subspace** and solving the optimization problem in the subspace.

Results: the proposed approach is applied to a series of optimization problems and the results show that the autoencoder can efficiently represent a nonlinear manifold reducing the dimensionality of the problem and solve the optimization problem.

Background & Methodology

Assuming the optimization problem has the following form:

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^{N_x}}{\text{maximize}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & z(\mathbf{x}) \le 0 \\ & \mathbf{x}^{ub} \le \mathbf{x} \le \mathbf{x}^{ub} \end{aligned}$$

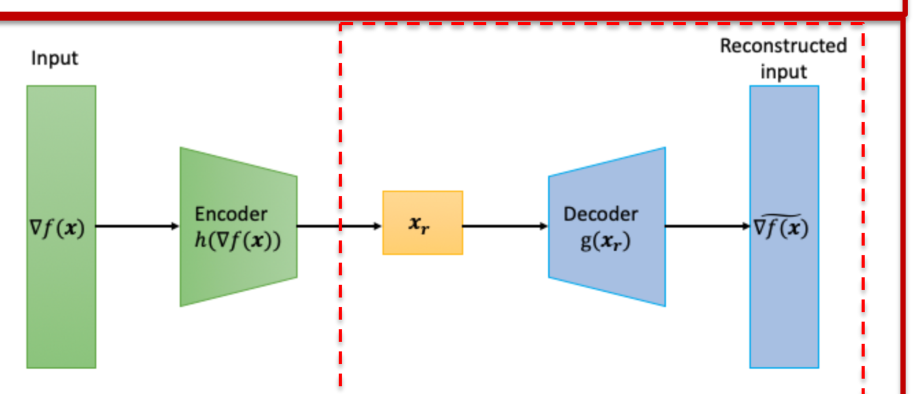
The method of active subspace compresses with SVD the gradients of the objective function f(x) to learn a linear low-dimensional representation of the input parameters: $x \approx U_x x_r$

$$[\nabla f(\mathbf{x}_1), \cdots, \nabla f(\mathbf{x}_m)] = [\mathbf{U}_x \ \ \mathbf{U}_{tr}] \begin{bmatrix} \mathbf{\Sigma}_x & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{tr} \end{bmatrix} \begin{bmatrix} \mathbf{V}_x \\ \mathbf{V}_{tr} \end{bmatrix}$$

Using linear Active subspace, the following problem is solved:

$$\begin{aligned} \underset{\mathbf{x}_r\in\mathbb{R}^{n_x}}{\text{maximize}} \quad & f(\mathbf{U}_x\mathbf{x}_r) \\ \text{subject to} \quad & z(\mathbf{U}_x\mathbf{x}_r) \le 0 \\ & \mathbf{x}^{ub} \le \mathbf{U}_x\mathbf{x}_r \le \mathbf{x}^{ub} \end{aligned}$$

The autoencoder learn a nonlinear low-dimensional representation of the input parameters: $x \approx g(x_r)$

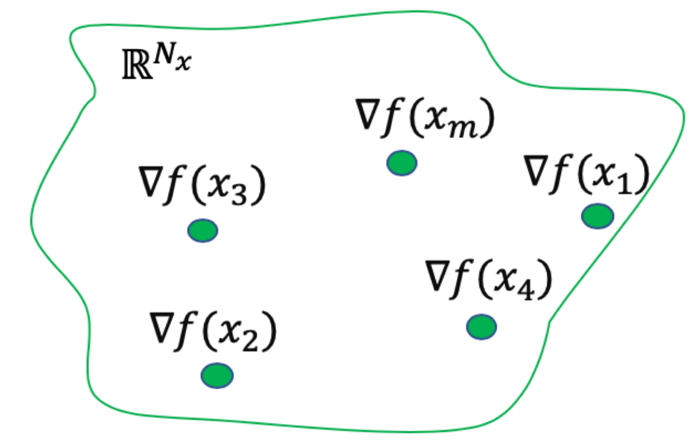


Using nonlinear Active subspace, the following problem is solved:

$$\begin{aligned} \underset{\mathbf{x}_r\in\mathbb{R}^{n_x}}{\text{maximize}} \quad & f(g(\mathbf{x}_r)) \\ \text{subject to} \quad & z(g(\mathbf{x}_r)) \le 0 \\ & \mathbf{x}^{ub} \le g(\mathbf{x}_r) \le \mathbf{x}^{ub} \end{aligned}$$

Data & Features

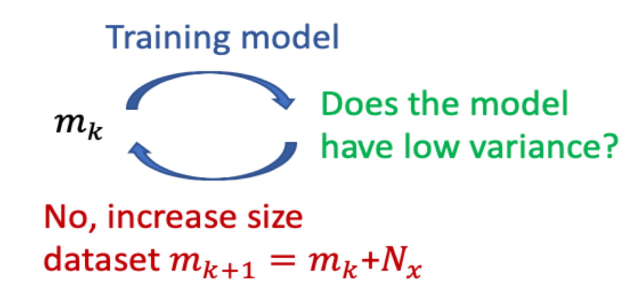
The dataset for training the autoencoder contains gradients ∇f. The data is generated using computer simulations: For a randomly sampled vector of parameters x_i the gradient $\nabla f(x_i)$ is computed.



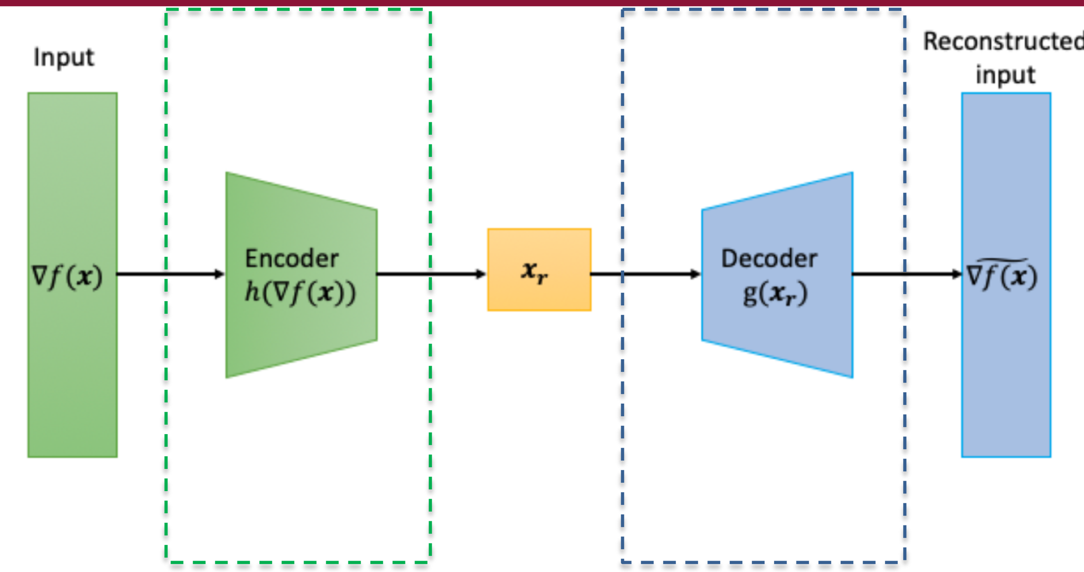
The dataset is the matrix M, (#rows= N_x and #cols = m)

$$\mathbf{M} = [\nabla f(\mathbf{x}_1), \cdots, \nabla f(\mathbf{x}_m)]$$

The size of the dataset, m, is chosen using an iterative process



Model & Model Selection



$$\mathbf{x}_{r_i} = h(\nabla f(\mathbf{x}_i), \mathbf{\Theta}_e) \qquad \widetilde{\nabla f(\mathbf{x}_i)} = g(\mathbf{x}_{r_i}, \mathbf{\Theta}_d)$$

Loss has the following definition:

$$RE(\nabla f(\mathbf{x}_i), \mathbf{\Theta}_e, \mathbf{\Theta}_d) = \sum_{j=1}^{N_x} \left(\frac{\partial f(\mathbf{x}_i)}{\partial x^j} - (g(\mathbf{x}_{r_i}(\mathbf{\Theta}_e)), \mathbf{\Theta}_d)^j \right)^2$$

$$Loss = \frac{1}{m} \sum_{i=1}^{m} RE(\nabla f(\mathbf{x}_i), \mathbf{\Theta}_e, \mathbf{\Theta}_d)$$

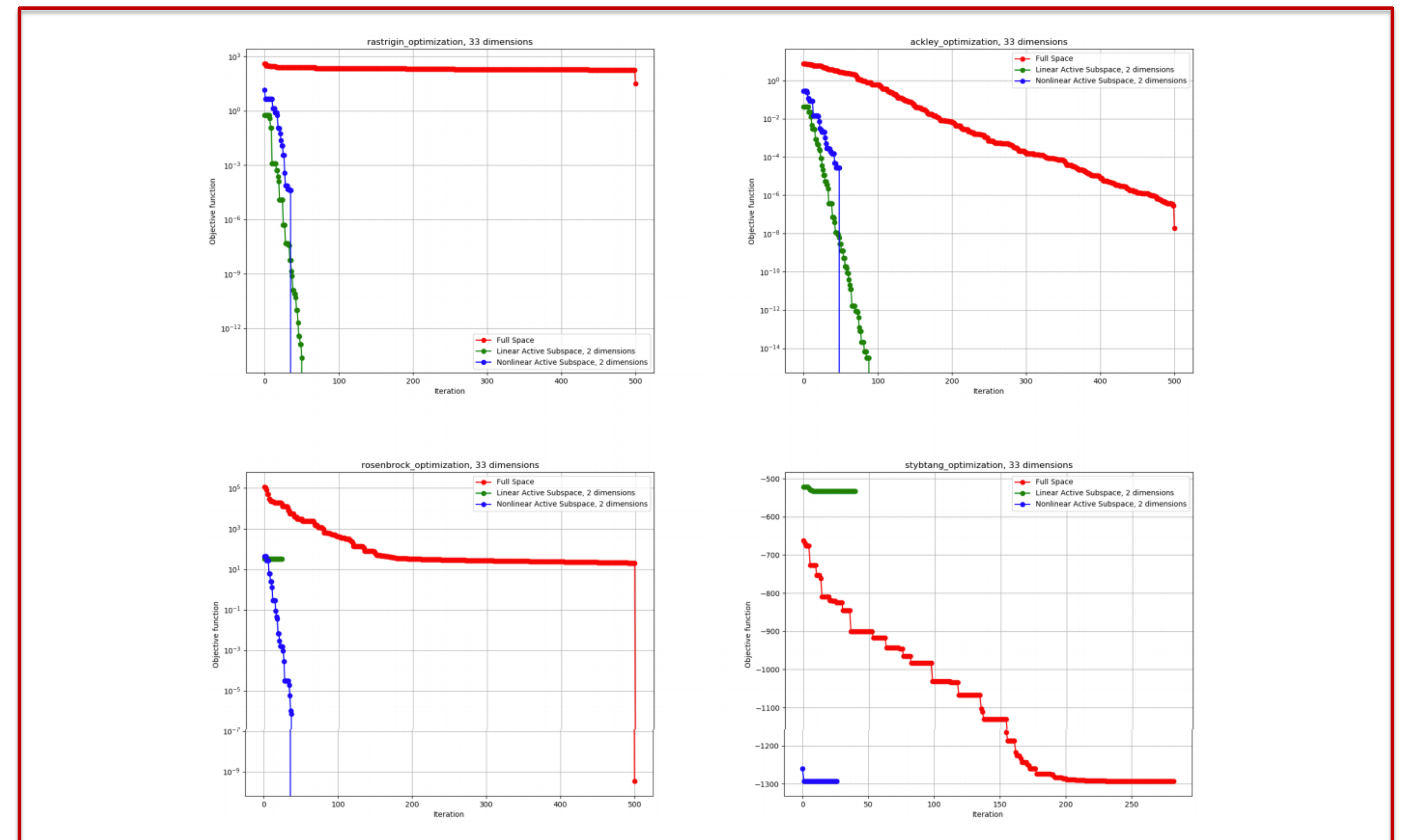
Three models considered. A grid search for selecting the model with the lowest dev set loss is run after training with the mAEWing dataset

model	F.C. autoenc.	Conv autoenc.	Conv-FC autoe.
Dev set Loss	36.6	21.8	36.7

The optimal architecture has 3 layers: the first layer has 33 filters, the second layer has 23 filters and the third layer has 3 filters (equal to n_x)

Experiments & Results

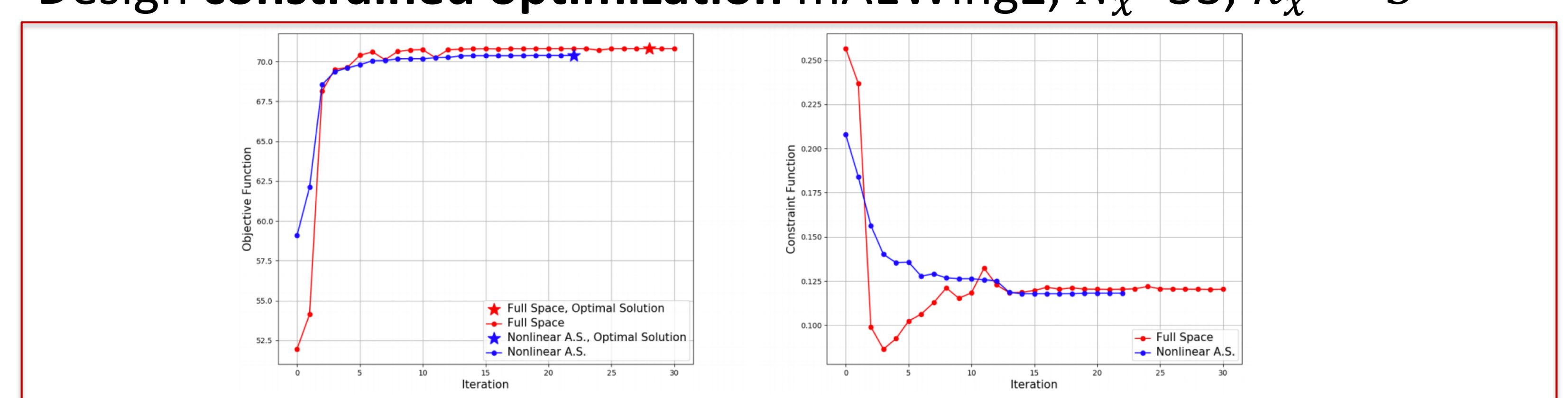
Unconstrained optimization, N_x=33, $n_x = 2$



Function	Computing gradients [s]	Training, linear active subspace [s]	Training, nonlinear active subspace [s]
Rastrigin	0.284	0.004	3.018
Ackley	0.070	0.004	3.014
Rosenbrock	0.052	0.004	3.581
Styb.-Tang	0.193	0.004	3.143

Function	Full space [s]	Found sol?	linear A.S. [s]	Found sol?	nonlinear A.S. [s]	Found sol?
Rastrigin	53.9	NO	0.5	YES	2.7	YES
Ackley	33.8	YES	0.4	YES	2.5	YES
Rosenbrock	31.6	YES	0.1	NO	1.8	YES
Styb.-Tang	26.3	YES	0.2	NO	1.5	YES

Design **constrained optimization** mAEWing2, N_x=33, $n_x = 5$



Problem	Computing gradients [hours]	Training, nonlinear active subspace [hours]
mAEWing2	41.25	0.0023

Function	Full space [s]	Found sol?	nonlinear A.S. [s]	Found sol?
mAEWing2	27.4	YES	20.52	YES

Conclusions and Future Work

- Using a nonlinear manifold for representing a subspace has advantages especially when a linear active subspace fails to capture nonlinearities in the problem
- The linear active subspace is a particular case of the nonlinear active subspace obtained with the autoencoder
- In the future, I would like to study how to make the training of the autoencoder cheaper and to study the performances of this methodology in other additional problems.