Topology Optimization Process

Alexander Kolomiytsev, Dmitriy Topchiy
Lada Karimullina, Yekaterina Smolenkova

https://github.com/katerina2901/NLA_Topology_team

Skolkovo Institute of Science and Technology, Skolkovo, Russia

Numerical linear Algebra, 2023

21.12.2023

Skoltech
Skolkovo Institute of Science and Technology

Topology optimization: problem statement

In the current project we want to solve the topology optimization problem for mechanical structures.

$$min_x f(x) = \Sigma_{i=1}^{n} E_i(x_i) q_i^T K_e q_i,$$
$$s.t.\ \ g(x) = \frac{v_e}{v_0} \Sigma_{i=1}^{n} x_i - v_{lim} \leq 0,$$
$$Kq = r,$$
$$0 \leq x_{i_{lower}} \leq x_i \leq 1, i = 1, 2, ..., n.$$

f(x)- compliance or strain energy , Ei - Young's modulus, xi - finite element, qi - element displacement vector , Ke - element stiffness matrix for an element with unit Young's modulus, K - is the global stiffness matrix, r - global force vector, q - global displacement vector, v(x) - final area x 1 in the 2D domain occupying the design domain, v0 - total area of the design domain, vlim - prescribed limit on the final volume fraction. Finite elements are assumed to be equally size, n - number of degrees of freedom

Topology optimization: problem statement with SIMP algorithm

Discretise a 2D domain into X-by-Y mesh of finite elements, and knowing that each element has two possible values (0 and 1), we have 2X ×Y possible permutations of the domain. The above mentioned problem can be solved using the SIMP method. It allows to achieve the non-binary solutions by choosing Young's modulus of a simple but very efficient form:

$$E_j(x_j) = E_{min} + x_j^p (E_0 - E_{min}),$$

where p is some form of penalty that will drive the solution to discrete solid-void-values. Then discrete design variables is replaced with continuous design variables which could be interpreted as the density of the material.

Postprocessing

Expressions for the plane strain or plane stress material law

$$\sigma = D\epsilon, \sigma = [\sigma_x \sigma_y \tau_{xy}]^T, \epsilon = [\epsilon_x \epsilon_y \gamma_{xy}]^T$$

The relationship between the stress and strain components reduced to only account for the strains that are not necessarily equal to zero:

$$D = \frac{E}{(1+v)(1-2v)} \begin{bmatrix} 1 & v & 0 \\ v & 1 & 0 \\ 1 & v & \frac{(1-2v)}{2} \end{bmatrix}$$

Postprocessing

Kinematic relationship

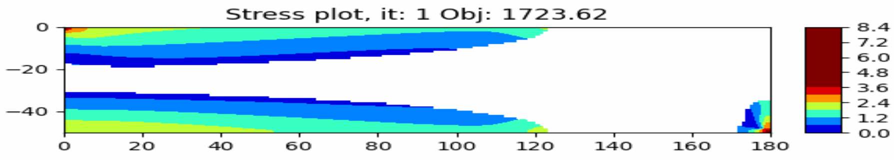
$$\epsilon = \Delta N u \equiv B u$$

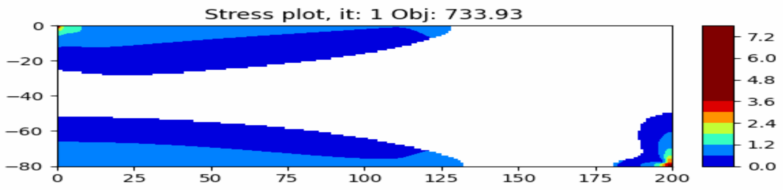
B-matrix contains the derivative of the shape functions

$$B = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{x} & \frac{\partial}{\partial y} \end{bmatrix} \left[\begin{array}{cc|cc|cc|cc} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{array}\right]$$

where N_i, i=1,...4, are the shape functions associated with the four nodes of the element.

Postprocessing

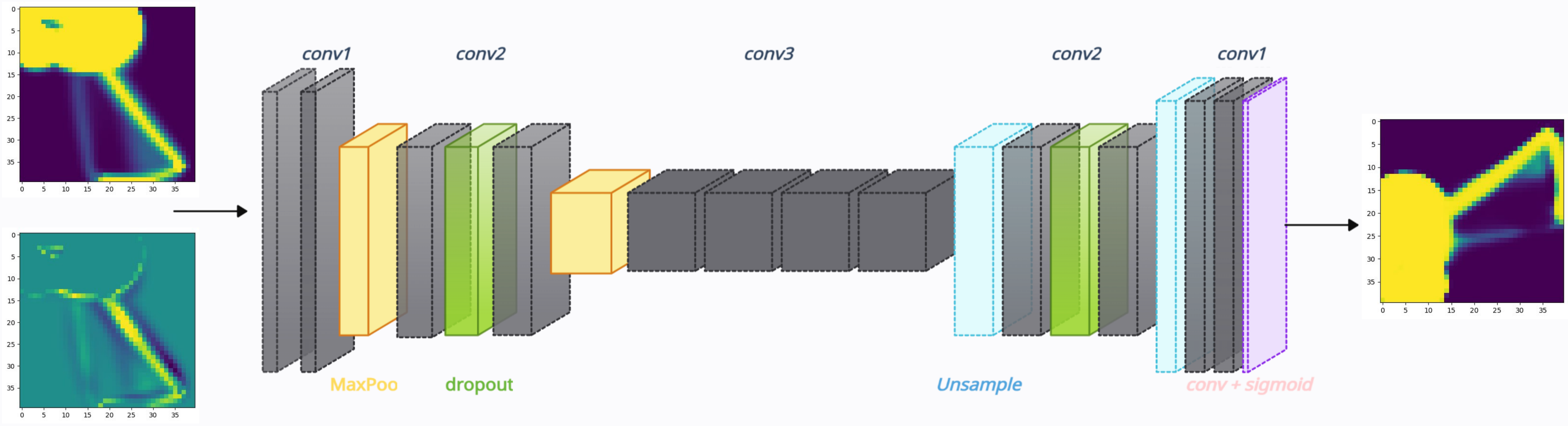




Optimization with greed 180x50, method: SIMP only

Optimization with greed 200x80, method: SIMP only

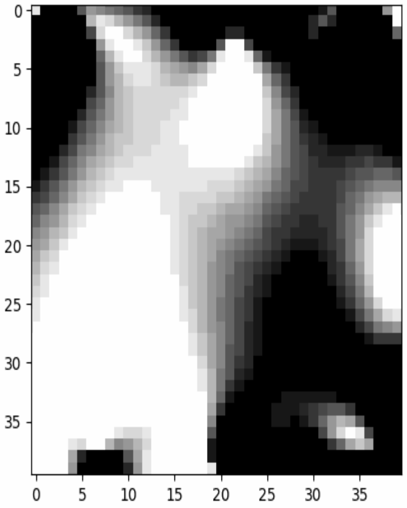
Neural Network for Topology Optimization

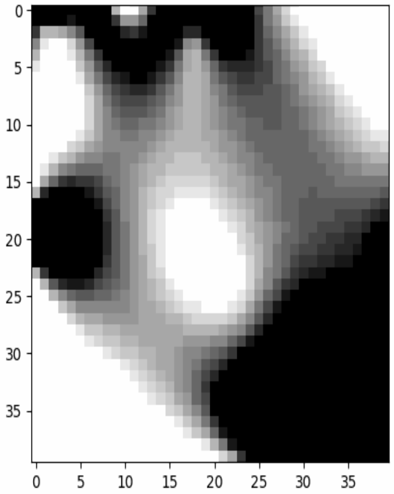


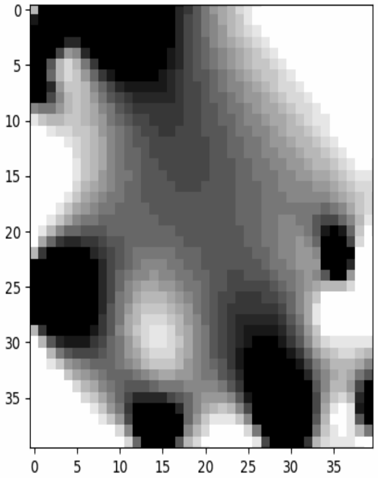
Unet

Dataset

- 10,000 objects.
- tensor of shape $100 \times 40 \times 40$







Neural Network for Topology Optimization

- Input: Two grayscale images or a two-channel image:
  - Density Distribution (Xn): Density distribution inside the design domain after the last topology optimization solver iteration.
  - Density Update (δX): Difference between the densities after the nth and (n-1)th iteration.
- Output: Grayscale image representing the predicted final structure, with the same resolution as the input.
- Batch size: 64
- Training epochs: 30

For training the network we used the objective function of the following form:

$$L = L_{conf}(X_{true}, X_{pred}) + \beta L_{vol}(X_{true}, X_{pred})$$

where the confidence loss is a binary cross-entropy:

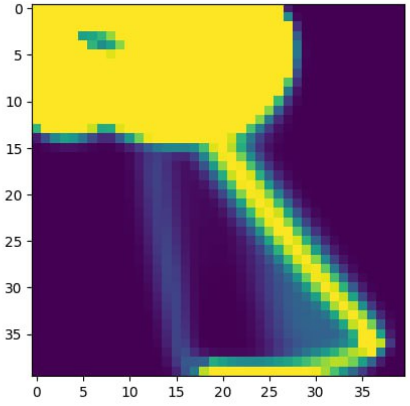
$$L_{conf}(X_{true}, X_{pred}) = -\frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}[X_{true}^{ij}\log(X_{pred}^{ij}) + (1 - X_{true}^{ij})\log(1 - X_{pred}^{ij})]$$

where N × M is the resolution of the image. The Lvol represents the volume fraction constraint:

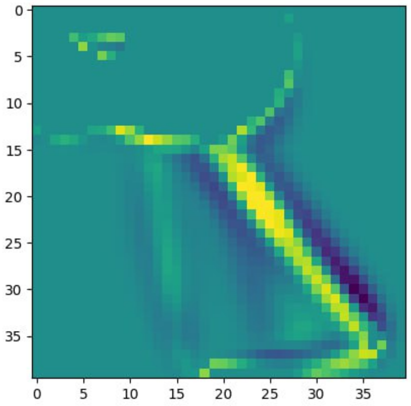
$$L_{vol}(X_{true}, X_{pred}) = (\bar{X}_{pred} - \bar{X}_{true})^2$$

Neural Network for Topology Optimization

Input

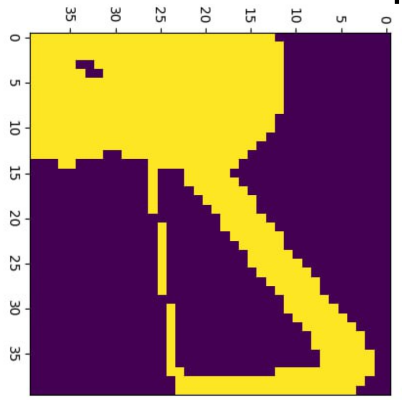


Density distribution Xn inside the design domain, obtained after the last performed iteration of the topology optimization solver.

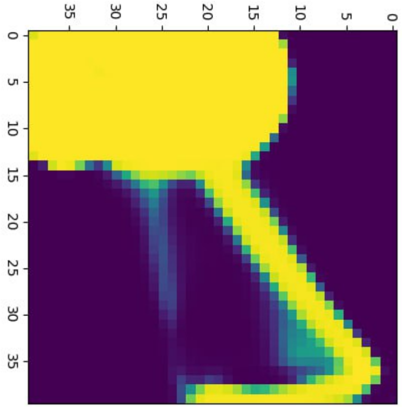


Last performed update (gradient) of the densities $\delta X = X_n - X_{n-1}$, which is the difference between the densities after the nth iteration and the (n-1)th iteration.

Output



y_true



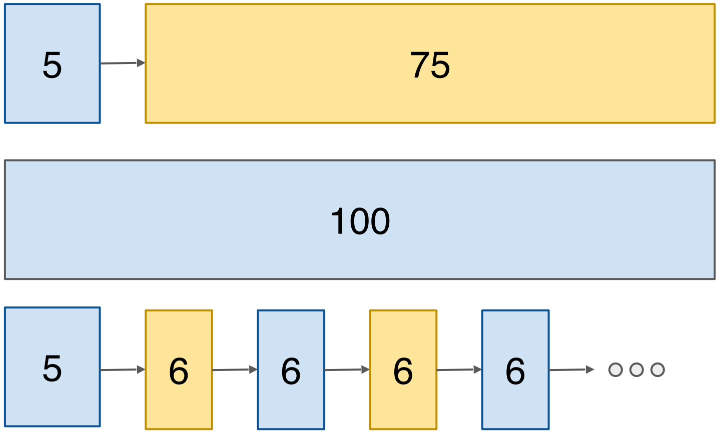
y_pred

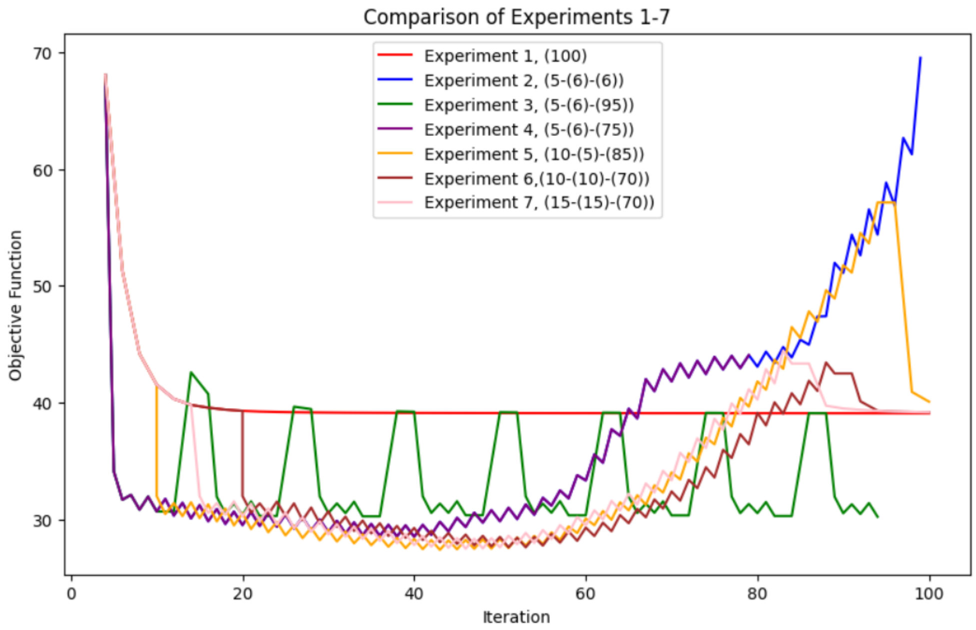
Journal, planning experiments

SIMP OPTIMIZER ITERATIONS

NN OPTIMIZER ITERATIONS

Experiments:

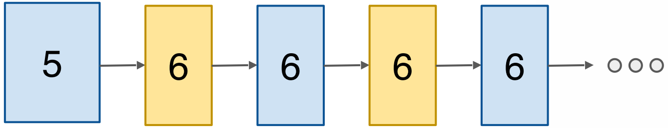


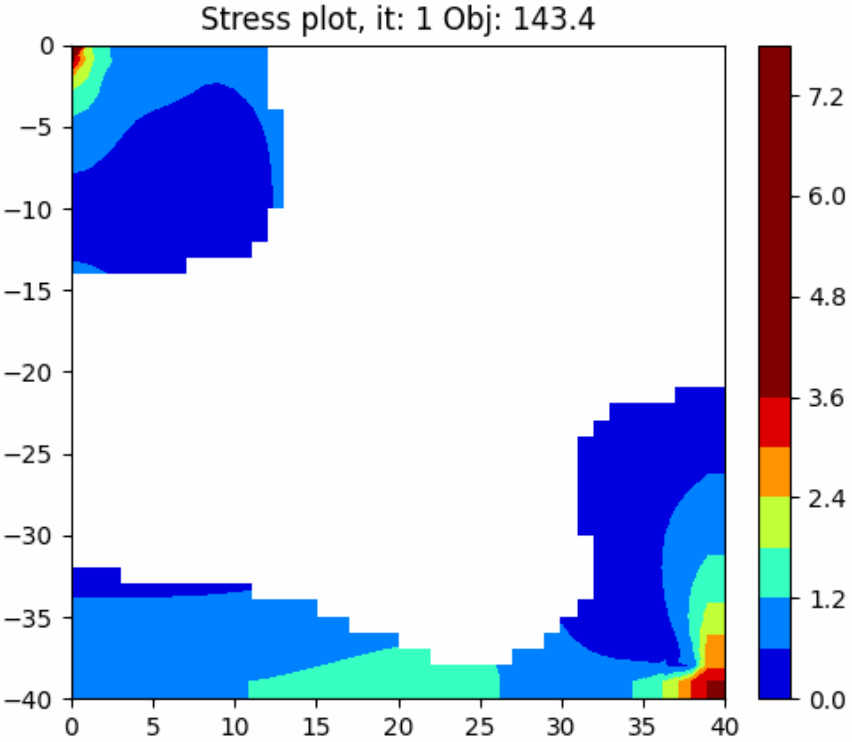


Iteration convergence

Journal, planning experiments

Experiment 2: 5-(6)-(6)





Our team

- Alexander Kolomiytsev - programming classes for SIMP-optimization, optimization with NN and Experiment journaling. Programming postprocessing, presentation
- Dmitriy Topchiy - neural network model with PyTorch, presentation
- Lada Kalimullina - research of topology optimization methods, presentation, comparing time of optimization for different ways to solve linear system
- Yekaterina Smolenkova - analyze and visualization of input data, presentation

Result

- TopoUnet is good to predict the next step, however using NN several times in a row accumulates the error
- Dataset is implemented only for 40*40 because of this the problems with a large grid cannot be solved
- The use of LA special solvers like LUsparse, makes sense only on large meshes. Iterative solvers are needed for 3D cases
- NN is slower than SIMP