On Engineering Design by Finite Element Analysis and Deep Reinforcement Learning

Georgy Tskhondiya at Gigala (www.facebook.com/GigaTsk/)

Abstract

Recently, deep reinforcement learning has had great success in artificial intelligence applications. Among them, beating the champion of the game of Go in 2016, mastering many Atari games and optimizing the work of data centers. In this paper, I have combined deep reinforcement learning and finite element analysis for the purpose of engineering design. General approach on how this combining can be accomplished as well as specific problems' solutions for simple structures are presented.

1 Introduction

Artificial intelligence in general and deep reinforcement learning in particular are powerful approaches in solving many nowadays problems in information technology, business, healthcare, and engineering. There is a myriad of applications for AI technologies that one can implement to make life easier. Engineering design is no exception. Designing a structure or the part of machinery is a very tiring process. One needs to make a lot of changes before resulting in the final design that satisfies structural loads. But this iterative process can be automated.

A typical approach to engineering design is finite element analysis. A number of authors have tried to combine finite element analysis and machine learning [1-3]. For example, [2] have used deep-autoencoder to approximate the large deformations on a non-linear, muscle actuated beam. In [3], machine learning was used to predict the deformation of the breast tissues during the compression. However, little attention has been paid to using reinforcement leaning in assisting engineering design.

In this work, my approach was to combine finite element analysis and deep reinforcement leaning to assist an engineer in her design process. General description on how this combination can be implemented as well as a designing process for three simple structures is presented. Structures are: (a) 1D bar, (b) bridge-like 2D truss, and (c) spool-like 2D frame. These structures can be found in most of real-world structural problems. For examples, spools are used to connect the subsea pipeline with a fixed riser nearby the offshore platform (Figure 1).

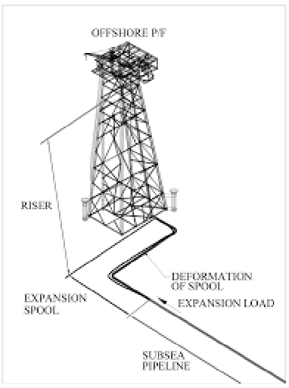

Figure 1. Spool connecting an offshore platform to the pipeline

2 Model

In this section, a way of combining finite element and reinforcement learning models is described.

2.1 **Finite Elements**, [4]

The three models considered in this work include the following types of finite elements:

(a) The linear bar element is a one-dimensional finite element where the local and global coordinates coincide. It is characterized by linear shape functions and is identical to the spring element except that the stiffness of the bar is not given directly. The linear element has modulus of elasticity *E*, cross-sectional area *A*, and length *L*.
(b) The plane truss element is a two-dimensional finite element with both local and global coordinates. It is characterized by linear shape functions. The plane truss element has modulus of elasticity *E*, cross-sectional area *A*, and length *L*. Each plane truss element has two nodes and is inclined with an angle ϑmeasured counterclockwise from the positive global *X* axis.
(c) The plane frame element is a two-dimensional finite element with both local and global coordinates. The plane frame element has modulus of elasticity *E*, moment of inertia *I*, cross-sectional area *A*, and length *L*. Each plane frame element has two nodes and is inclined with an angle ϑmeasured counterclockwise from the positive.

2.2 Reinforcement Learning

Reinforcement learning can be understood using the concepts of agents, environments, states, observations, actions and rewards. A reinforcement learning agent interacts with its environment in discrete time steps. At each time step, the agent receives an observation, which typically includes the reward. It then chooses an action, from the set of available actions, which is subsequently sent to the environment. The environment moves to a new state and the reward associated with transition is determined. The goal of a reinforcement learning agent is to collect as much reward as possible. The advantage of reinforcement learning is that one does not have to provide labeled data for training. Reinforcing learning system learns by maximizing rewards with no supervision. In this work, agent uses neural network policy gradient algorithm. The algorithm optimizes the parameters of a policy by following the gradients toward higher rewards, [5].

2.3 Finite Element Environment to Reinforcement Learning Agent Interaction

The finite element model represents an environment to which an agent applies actions and from which it gets observations and rewards (Figure 2). An agent uses neural network to decide on its actions. Actions change geometry of the structure, new geometry then subjected to FEA. Finite element analysis produces the state, which then is fed to neural network. And the process repeats itself. The agent gets rewards if it meets an optimization objective of minimizing target value (usually some nodal displacement in the structure) in each of the learning iterations. The end result of the modeling (after inference stage) is an optimized design of a structure. The inference stage is a usual predict-function for a neural network where an agent makes actions of altering the geometry based on observations only.

An engineer interacting with the presented system provides initial design of a structure, applies loads, sets allowed alternation of the geometry, and specifies an optimization objective (usually minimize nodal displacement or internal force values). The rest the system does automatically.

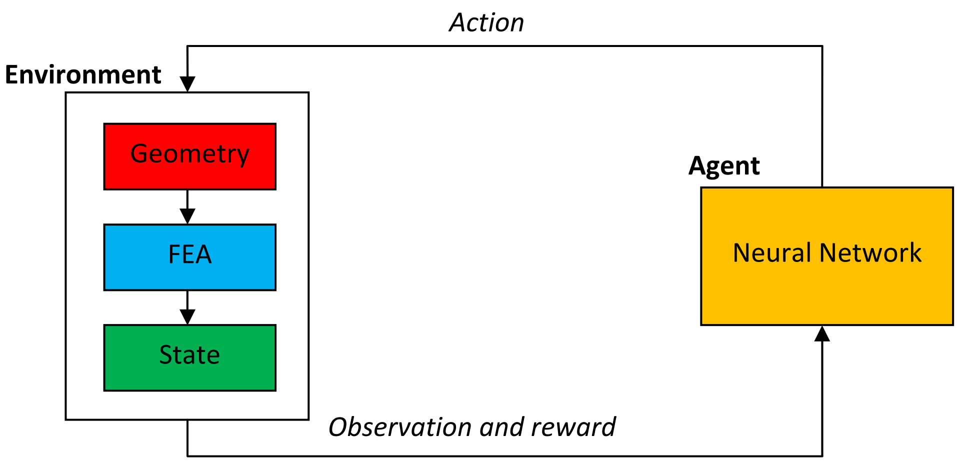

Figure 2. Reinforcement learning agent to finite element environment interaction

2.4 Implementation Details

(a) One-dimensional bar was subjected to a load at its right tip, the left tip was fixed. The environment state (modulus of elasticity E, cross-sectional area A, and tip nodal displacement) was fed to an agent (neural network) which produced actions of altering modulus of elasticity or cross-sectional area or both. The optimization objective was to minimize tip displacement. After training, 50 inference steps were used to obtain optimal modulus of elasticity and cross-sectional area. The modeling has shown that the agent learned how to meet an objective in each inference step. Tip nodal displacement reduced from 2.6 down to 1.6.

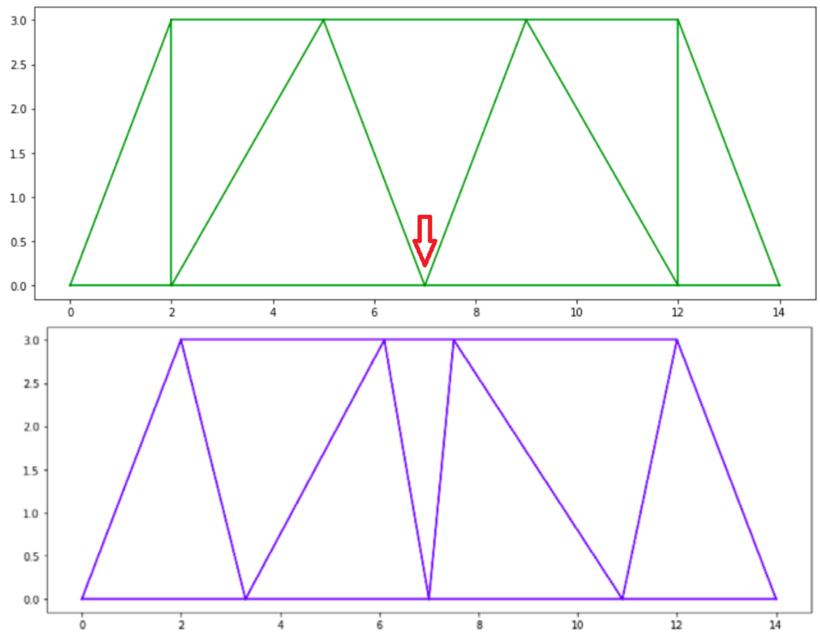

Figure 3. Initial (top) and final (bottom) designs of a bridge-like structure by reinforcement learning and finite element analysis

(b) Two-dimensional bridge-like plane truss fixed at its leftmost bottom and rightmost bottom nodes was loaded in the middle node in downward vertical direction. The environment state (geometry and middle node vertical displacement) was fed to an agent (neural network) which produced actions of altering geometry of a "bridge" in a certain way. The optimization objective was to minimize the middle node displacement. After training, 50 inference steps were used to obtain optimal geometry of a bridge. The modeling shows that the agent learned how to meet an objective and produced almost symmetrical final design of a bridge (Figure 3). Middle node vertical displacement reduced from -0.8888E-03 down to -0.8384E-03.

(c) Two-dimensional spool-like plane frame was loaded at its rightmost node in compressing horizontal direction, leftmost node was completely fixed. The environment state (geometry and maximum nodal displacement) was fed to an agent (neural network) which produced actions of altering geometry of a "spool" in a certain way. The optimization objective was to minimize the maximum nodal displacement. After training, 50 inference steps were used to obtain optimal geometry of a spool. The modeling has shown that the agent learned how to meet the objective, (Figure 4). Maximum nodal displacement reduced from 1.38 down to 0.18.

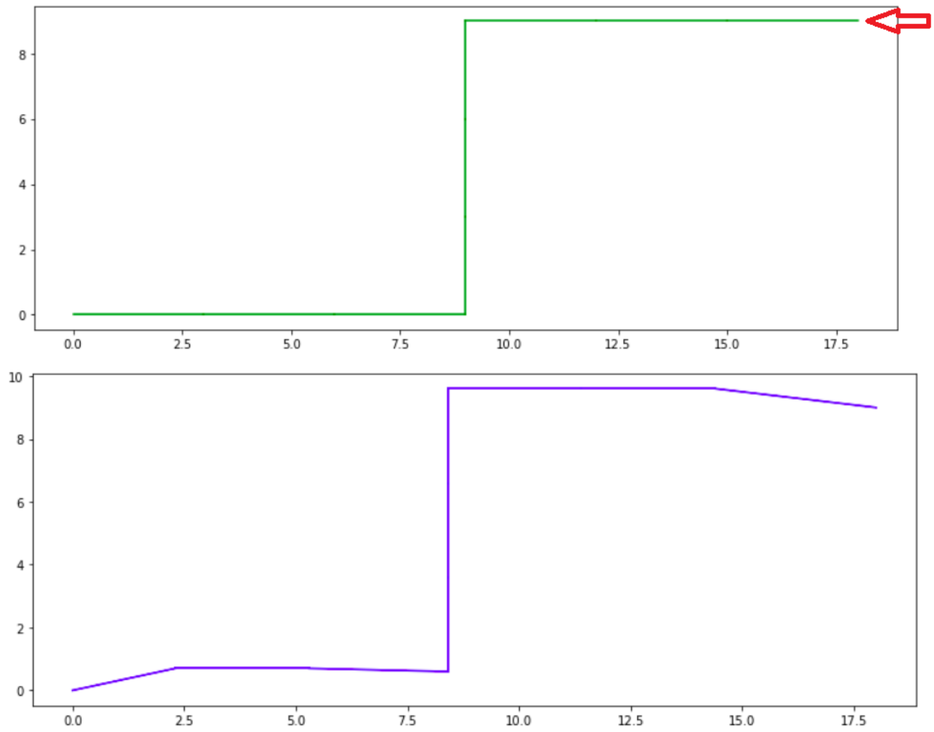

Figure 4.Initial (top) and final (bottom) designs of a spool-like structure by reinforcement learning and finite element analysis

The code used in this paper can be found at [6].

3 Discussion

The results in this paper show that deep reinforcement learning in combination with finite element analysis can be used as automatic iterative process of engineering design. An engineer provides initial geometry of a structure, sets loads and allowed actions to alter the geometry, specifies an optimization objective (e.g. minimize internal force value), and starts training of the model. After training, in inference stage, the engineer gets her final design. Thus, combination of finite element analysis and reinforcement learning makes engineering design automated. In the future, the role of artificial intelligence in assisting engineering design will grow substantially. Today, most companies which develop FE packages already incorporate machine learning in their products in some form or another. In this regard, more research focus should be put on this issue.

Acknowledgments

The author is thankful to Archil Tskhondiya, Yuriy Ermolov and Olivier Carloni for reviewing this paper and providing valuable comments.

References

[1] Machine Learning and Finite Element Method for Physical Systems Modeling. O.Kononenko, I.Kononenko, arXiv.org

[2] Towards Finite-Element Simulation Using Deep Learning. Francois Roewer-Despres, Najeeb Khan, Ian Stavness, CMBBE 2018

[3] A finite element-based machine learning approach for modeling the mechanical behavior of the breast tissues under compression in real-time. Martinez-Martinez F, et al. Comput Biot Med, 2017 Nov 1; 90:116-124

[4] MATLAB Guide to Finite Elements. An Interactive Approach, Peter I. Kattan, 2nd edition

[5] Hands-On Machine Learning with Scikit-Learn and TensorFlow, Aurelien Geron

[6] https://github.com/gigatskhondia/Reinforcement_Learning_and_FEA