Autoencoder-based generation of individuals in population-based metaheuristics

Bastien Pasdeloup, Maryam Karimi-Mamaghan, Mehrdad Mohammadi, Patrick Meyer

Autoencoder-based generation of individuals in population-based metaheuristics

Bastien Pasdeloup, Maryam Karimi-Mamaghan, Mehrdad Mohammadi, Patrick Meyer

IMT Atlantique, Lab-STICC, UMR CNRS 6285, Brest F-29238, France
{ bastien.pasdeloup, maryam.karimi, mehrdad.mohammadi, patrick.meyer }
@imt-atlantique.fr

Keywords: *Metaheuristics, autoencoders, traveling salesman problem, optimization.*

1 Introduction

Autoencoders [1] are a particular class of neural network architectures that work in an unsupervised way. Contrary to more common architectures that require a lot of labeled data for tasks such as classification, autoencoders are trained so that the produced output is as close as possible to the given input. The model of an autoencoder can be split in two parts : first, encoding layers transform the input data into a lower-dimensional vector. Then, decoding layers try to retrieve the initial data from that vector. In the process, the network learns an efficient representation of the distribution of the training set, so called the latent space. This particular space can be seen as a space of features, which is learned directly from the data. The latent space has the property to allow arithmetics on the data, such as interpolation. As an example, one can compute the latent representation of two images of handwritten digits [2], compute their average in that space, and decode the obtained vector to generate an unseen third digit that mixes characteristics of the two inputs.

Population-based metaheuristics [3] such as genetic algorithms aim at approximating the optimum to a complex problem by maintaining a population of candidate solutions, some of which are selected for reproduction to produce offsprings for the next generation. Reproduction is generally made through crossovers and mutations, that allow production of children that draw characteristics from both parents, while keeping some undeterministic aspects.

In this work, we explore a new way of breeding individuals in a population-based metaheuristic, by exploiting the ability of autoencoders to interpolate representation of candidate solutions in a learned latent space. In the remaining of this document, we present in more details our idea and show its application to the traveling salesman problem.

2 Proposed approach

In this work, we propose to consider the traveling salesman problem (TSP). Let us name the cities with integers in $[\![1, N]\!]$, and let $\mathbf{W} \in \mathbb{R}^{N \times N}$ be the weights matrix of a complete graph $\mathcal{G}$ such that $\mathbf{W}[i, j]$ is the Euclidean distance between cities i and j. A candidate solution to the TSP is a Hamiltonian cycle on $\mathcal{G}$. We choose to describe it as a matrix $\mathbf{S}$ in which $\mathbf{S}[i, j] = 1$ if i and j are adjacent cities in that cycle, and 0 otherwise. The fitness score of that solution is given by $\mathrm{f}(\mathbf{S}) = \frac{1}{2\|\mathbf{W} \odot \mathbf{S}\|_{1,1}}$, where $\odot$ denotes the Hadamard (elementwise) product. Finally, let $\mathbf{s} = \mathrm{vec}(\mathbf{S})$ be the vectorization of $\mathbf{S}$, and $\mathbf{S} = \mathrm{vec}^{-1}(\mathbf{s})$ the inverse operation.

We first train an autoencoder to learn a latent space that describes properly the candidate solutions. We design a simple model, composed of an input layer of size N^2, a dense layer (associated with the latent representation) of size L, and a dense output layer of size N^2. We then train this network by feeding it T randomly generated vectorized candidate solutions.

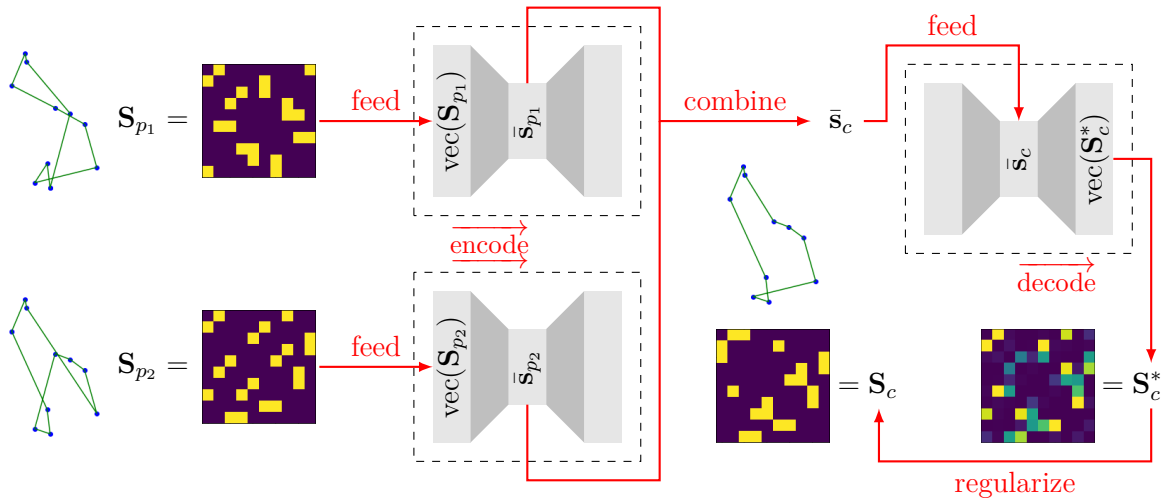

FIG. 1 – Schematic view of the proposed approach to produce a new candidate solution.

Once the network is trained, we randomly generate an initial population $\mathcal{P} = \{\mathbf{S}_1, \dots, \mathbf{S}_S\}$ of S candidate solutions, and apply a genetic algorithm as follows (also, see Figure 1) :

1. We initialize a new population $\mathcal{P}'$ with the H solutions in $\mathcal{P}$ with highest fitness ;
2. We draw at random P candidate solutions $(\mathbf{S}_{p_1}, \dots, \mathbf{S}_{p_P})$ from $\mathcal{P}$ to act as parents, with probability $\frac{\mathrm{f}(\mathbf{S}_{p_i})}{\sum_{j=1}^{P} \mathrm{f}(\mathbf{S}_{p_j})}$ for parent $\mathbf{S}_{p_i}$. We feed the vectorizations $(\mathrm{vec}(\mathbf{S}_{p_1}), \dots, \mathrm{vec}(\mathbf{S}_{p_P}))$ of these candidate solutions to the encoding part of the autoencoder, and obtain a representation in the latent space for each of them, which we note $(\bar{\mathbf{s}}_{p_1}, \dots, \bar{\mathbf{s}}_{p_P})$;
3. We generate the latent representation $\bar{\mathbf{s}}_c$ of a new child solution as a combination of those of the parents, weighted by their fitness :

$$\bar{\mathbf{s}}_c = \frac{1}{\sum_{i=1}^{P} \mathrm{f}(\mathbf{S}_{p_i})} \sum_{i=1}^{P} \mathrm{f}(\mathbf{S}_{p_i}) \bar{\mathbf{s}}_{p_i} \ ;$$

4. We feed $\bar{\mathbf{s}}_c$ to the decoding part of the autoencoder to obtain a decoded vector $\mathbf{s}_c^*$, which in general does not describe a Hamiltonian cycle. We thus regularize $\mathbf{S}_c^* = \mathrm{vec}^{-1}(\mathbf{s}_c^*)$ by following a path in $\mathbf{S}_c^*$ that, from city i, greedily goes to the next unvisited city j with maximum $\mathbf{S}_c^*[i, j]$, until a cycle is formed. This produces the child candidate solution $\mathbf{S}_c$;
5. If $\forall i \in [\![1, P]\!] : \mathrm{f}(\mathbf{S}_c) > \mathrm{f}(\mathbf{S}_{p_i})$, we append $\mathbf{S}_c$ to $\mathcal{P}'$. Then, if $\mathcal{P}'$ has S elements, $\mathcal{P} \leftarrow \mathcal{P}'$ and we start again from 1 until a stopping criterion is met. Otherwise, we loop to 2.

One advantage of the proposed method is that it removes the necessity for expert-designed breeding rules, and is quite generic. Additionally, it is worth noticing that in general, the output of the autoencoder is not perfectly equal to the fed input. This leads to small errors in the latent representation, and has a similar impact as mutations in classical genetic algorithm settings. Some encouraging experiments have been made and will be developed in more details in the presentation, along with comparisons with existing methods.

Références

[1] Geoffrey E Hinton and Richard S Zemel. *Autoencoders, minimum description length and Helmholtz free energy.* Advances in neural information processing systems, 3–10, 1994.

[2] Yoshua Bengio, Grégoire Mesnil, Yann Dauphin and Salah Rifai. *Better mixing via deep representations.* International conference on machine learning, 552–560, 2013.

[3] Christian Blum and Andrea Roli. *Metaheuristics in combinatorial optimization : Overview and conceptual comparison.* ACM Computing Surveys (CSUR), 35(3), 268–308, 2003.