# On the Locality of Neural Meta-Representations

Luís F. Simões VU University Amsterdam Iuis.simoes@vu.nl A. E. Eiben VU University Amsterdam a.e.eiben@vu.nl

## ABSTRACT

We consider the usage of artificial neural networks for representing genotype-phenotype maps, from and into continuous decision variable domains. Through such an approach, genetic representations become explicitly controllable entities, amenable to adaptation. With a view towards understanding the kinds of space transformations neural networks are able to express, we investigate here the typical representation locality given by arbitrary neuro-encoded genotypephenotype maps. We consistently find high locality space transformations being carried out, across all tested feedforward neural network architectures, in 5, 10 and 30 dimensional spaces.

#### **Categories and Subject Descriptors**

I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search—*Heuristic methods* 

#### Keywords

Representations; Genotype-Phenotype map; Neural networks

#### 1. INTRODUCTION

Adjusting the parameters and/or components of evolutionary algorithms (EA) during a run offers several advantages [1]. Over the last two decades, we have witnessed a plethora of methods being introduced for automatically adapting all sorts of EA parameters and components, with one notable exception: the genotype-phenotype (G-P) mapping, a.k.a. the representation. Few papers ever addressed its adaptation.

We propose neural networks (NN) as a generic framework for representing representations (i.e. NNs as a metarepresentation). Following such an approach, changes to the architecture and/or weights of a NN providing a problem's genotype-phenotype mapping will effectively change the representation space explored by evolution. Feasible ap-

*GECCO'14*, July 12–16, 2014, Vancouver, BC, Canada. ACM 978-1-4503-2881-4/14/07.

proaches for implementing such representation learning include co-evolution, self-adaptation, and meta-evolution.

In this paper, we formally introduce the approach of using NNs as genotype-phenotype maps, and then follow with an investigation into the locality of representations defined though these means.

## 2. NEURAL NETWORKS AS GENOTYPE-PHENOTYPE MAPS

Neural networks form the basis of our approach. Being proven global function approximators (given a sufficiently large number of hidden layer neurons), feedforward neural networks are in principle capable of expressing any possible G-P map. Also, the community has ample experience with neuroevolution methods, which can be deployed for the effective learning of representations.

In general, evolutionary search is guided by a fitness function f that can be decomposed into a genotype-phenotype mapping  $f_g : \Phi_g \to \Phi_p$  and a phenotype-fitness mapping  $f_p : \Phi_p \to \mathbb{R}$ , cf. [2, Sec. 2.1.2]. When considering an EA that searches in a continuous genotypic space, for solutions that decode into continuous phenotypes, we then have that  $\Phi_g \subset \mathbb{R}^m$ , and  $\Phi_p \subset \mathbb{R}^n$ , where m and n stand, respectively, for the dimensionalities of the considered genotypic and phenotypic spaces. A genotype-phenotype map is then a transformation  $f_g : \mathbb{R}^m \to \mathbb{R}^n$ .

Let  $\mathcal{N}$  be a fully connected, feedforward neural network with l layers and  $d^k$  neurons on its k-th layer (k = 1..l). If  $\mathcal{L}^k$  is the vector representing the states of the  $d^k$  neurons in its k-th layer, then the network's output can be determined through

$$\mathcal{L}_{i}^{k} = \sigma(b_{i}^{k} + \sum_{j=1}^{d^{k-1}} w_{ij}^{k} \mathcal{L}_{j}^{k-1}), i = 1..d^{k}$$

where  $b^k$  represents the biases for neurons in the k-th layer, and  $w_i^k$  the weights given to signals neuron  $\mathcal{L}_i^k$  gets from neurons in the preceding layer. A sigmoidal activation function  $\sigma(y) = 1/(1 + e^{-y})$  is used throughout this paper. The network's output,  $\mathcal{L}^l$ , is then uniquely determined through b, w, and  $\mathcal{L}^1$ , the input vector fed to its input layer.

Without loss of generality, in the sequel we assume that the given phenotype space is an n dimensional hypercube. (If needed, the interval [0, 1] can be mapped with a trivial linear transformation to the actual user specified lower and upper bounds for each variable under optimization.) Using a neural network as a G-P map, we then obtain a setup where the number of output neurons  $d^l = n$  and the mapping itself is  $f_g : [0, 1]^{d^1} \to [0, 1]^{d^l}$ . To specify a given G-P mapping

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s). Copyright is held by the author/owner(s).

network we will use the notation  $\mathcal{N}(m_p, m_a)$ , where  $m_p$  and  $m_a$  are the map parameters<sup>1</sup> and map arguments, defined as follows. The vector  $m_p \in [-1, 1]^d$  contains the definition of all weights and biases in the network, while the vector  $m_a$  designates the input vector fed into the network. With this notation, we obtain a formal framework where genotypes are map arguments to the neural net and the representation is  $f_g = \mathcal{N}(m_p, .)$ . Given a genotype  $x^g \in [0, 1]^{d^1}$ , the corresponding phenotype is  $f_g(x^g) = \mathcal{N}(m_p, x^g) \in [0, 1]^{d^1}$ .

As shorthand for a considered NN architecture, we will use notation such as 30-5-10, to indicate a fully connected feeedforward neural network with l = 3 layers, having  $d^1 =$ 30 neurons in its input layer,  $d^2 = 5$  neurons in the hidden layer, and  $d^3 = 10$  neurons in the output layer ( $\Phi_g =$  $[0, 1]^{30}, \Phi_p \subset \mathbb{R}^{10}$ ).

#### 3. LOCALITY

In the sizing of the neural network employed to represent a G-P map, we are given a trade-off between its capacity to define arbitrary space transformations, and our capacity to effectively train it. To better understand which kinds of mappings might be defined in practice through this approach, we focus on one often-studied representation property, *locality*, and address the following research question:

• what is the locality of representations given by small to medium sized neural networks?

A representation's locality [2, Sec. 3.3] describes how well neighboring genotypes correspond to neighboring phenotypes. In a representation with perfect (high) locality, all neighboring genotypes correspond to neighboring phenotypes. Theoretical and experimental evidence [2] support the view that high locality representations are important for efficient evolutionary search, as they do not modify the complexity of the problems they are used for.

#### Map characterization.

The G-P map design space is explored by randomly sampling (with uniform probability) NN weights and biases, in the range [-1, 1], thus providing the definition of map parameters,  $m_p$ . We follow by generating a large number of map arguments,  $m_a$  (10000, to be precise), in the range [0, 1], according to a quasi-random distribution. Sobol sequences are used to sample the genotype space  $(m_a \in \Phi_g)$ , so as to obtain a more evenly spread coverage. The  $m_a$  scattered in the genotype space are subsequently mapped into the phenotype space. The Euclidean metric is used to measure distances between points within the genotype space, as well as within the phenotype space.

We characterize the locality of representations definable by a given neural network architecture, by randomly sampling the space of possible network configurations  $(m_p)$  in that architecture. A sample of 1000 points is taken, out of the 10000  $m_a$  given by the Sobol sequence (mentioned above), and a mutated version generated. A mutation is always a random point along the surface of the hypersphere centered on the original genotype, and having a radius equal to 1% the maximum possible distance in the  $\Phi_g$  hypercube. The mutated  $m_a$  is mapped into the phenotype space, and



Figure 1: Locality of representations expressible by different sized neural networks. Shown: empirical cumulative distribution functions of distances in phenotype space between pairs of neighboring genotypes.

its distance there to the original point's phenotype measured. Given we wish to consider phenotype spaces having distinct numbers of dimensions, and importantly, given the fact that each different G-P map encodes a different subset of the phenotype space, it becomes important to normalize phenotypic distances, in a way that makes then comparable. To that end, we identify the hyperrectangle that encloses all the phenotypes identified in the initial scatter of 10000 points, and use the maximum possible distance value there to normalize phenotype distances.

#### Results.

Figure 1 characterizes the locality of representations definable by different NN architectures. Each of the shown distributions was obtained by analyzing 1000 randomly generated G-P maps having that architecture, and thus represents a total of  $10^6$  measured phenotype distances. We consistently observe high locality representations resulting from *all* studied NN architectures: a mutation step of 1% the maximum possible distance in genotype space is in all cases expected to take us across a distance in phenotype space of at most ~ 1% the maximum possible distance among phenotypes representable by the considered G-P map.

#### 4. CONCLUSION

We investigated the usage of neural networks as a metarepresentation, suited to the encoding of genotype-phenotype maps for arbitrary pairings of fitness landscapes and metaheuristics that are to search on them.

Small to moderately sized feedforward neural networks were found to define, on average, high locality representations (where structure of the phenotypic fitness landscape is locally preserved in the genotype space).

## 5. ACKNOWLEDGMENTS

Luís F. Simões was supported by FCT (Ministério da Ciência e Tecnologia) fellowship SFRH/BD/84381/2012.

#### 6. **REFERENCES**

- A. E. Eiben, R. Hinterding, and Z. Michalewicz. Parameter Control in Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2):124–141, 1999.
- F. Rothlauf. Representations for Genetic and Evolutionary Algorithms. Springer, Berlin, Heidelberg, 2nd edition, Nov. 2006.

<sup>&</sup>lt;sup>1</sup>"Map parameters" named by analogy with the strategy parameters (e.g., standard deviations of a Gaussian mutation) traditionally used in Evolution Strategies.