

Universität Hamburg
Department Informatik
Knowledge Technology, WTM

Effects of encoding on the general learning ability of artificial neural networks

Seminar Paper

Bio-inspired Artificial Intelligence

Jan Fabian Schmid

Matr.Nr. 6440383

2schmid@informatik.uni-hamburg.de

27.01.2016

Abstract

Synaptic plasticity is an important aspect of the development of artificial neural networks (ANNs) with the capabilities of nature like animal nervous systems. The ability of ANNs to deal with a larger scope of problems depends strongly on their synaptic plasticity. Tonelli and Mouret are using an approach to develop ANNs with evolutionary algorithms and different genetic encodings. They describe synaptic plasticity as the appearance of regularity in the ANN and successfully identify a dependency of the regularity of ANNs and their general learning ability. The approach and its results seem promising, but the measurement of regularity is problematic and doesn't allow doubtless conclusions.

Contents

1	Introduction	2
1.1	Hypothesis	3
2	Background Information	3
2.1	Regularity	4
2.2	From genes to nervous systems	4
2.3	Skinner box	7
3	Approach description	7
4	Approach analysis	10
4.1	Discussion	11
5	Conclusion	13
	Bibliography	15

1 Introduction

This seminar paper is supposed to give an in-depth analysis of the paper of Tonelli and Mouret [3] with focus on the effects of encoding on the general learning ability of artificial neural networks.

One of the more prominent topics in bio-inspired artificial intelligence is the modelling of animal nervous systems using artificial neural networks (ANNs) (cf. [3] p.1).

ANNs are not readily useful to model animal nervous systems, instead they can be used for function approximation in a variety of application fields. In contrast to ANNs nature-like neural networks should be larger, more organized and more plastic (cf. [3] p.1). The common approach to develop such networks, are evolutionary algorithms, not least because the evolution was responsible for developing the natural example.

Two aspects of creating nature-like ANNs, that research mainly focuses on (cf. [3] p.1), are:



- **Genetic encoding:** To use evolutionary algorithms for the development of neural networks with specific behaviours, it is necessary to present the structure in some kind of encoding that can be handled by them. As in the genotype of animals, it is not desired to encode each neuron specifically, but rather to define rules for the network structure, that then lead to regular patterns (cf. [3] p.1).



Algorithms that use a mapping of genotype to phenotype are called *artificial developmental systems*.

- **Synaptic plasticity:** The ability of an artificial neural network to change repeatedly during lifetime is called synaptic plasticity. A neural network with good plasticity is supposed to be able to adapt to different problems through a learning process, ideally even to situations that were not considered during the development of the network.

For the usage of evolutionary algorithms alongside a proper genetic encoding the fitness function has to be defined. To this fitness function individuals that are newly developed during the application of the evolutionary algorithm are adapted. If it is desired to obtain plastic ANNs the fitness function could check for fitness on many different test cases to ensure the general learning ability. It turns out however that this approach can hardly be feasible for more complex environments as the number of required test cases grows exponentially (cf. [3] p.2).



In the studied paper [3] it is therefore suggested that genetic encoding and synaptic plasticity should be examined simultaneously (cf. [3] p. 1).

1.1 Hypothesis

Tonelli and Mouret say that the use of artificial developmental systems leads to more regular network structures since it is easier to describe a more regular structure with repetitions and symmetries than to describe an irregular structure with seemingly no relations between the individual network parts using a general rule set like genetic encoding does.(cf. [3] p.2) Therefore their main hypothesis is: „We here propose that this bias towards regularity is critical to evolve plastic neural networks that can learn in a large variety of situations“ ([3] p.2).

To visualize this idea they present the illustration in figure 1.

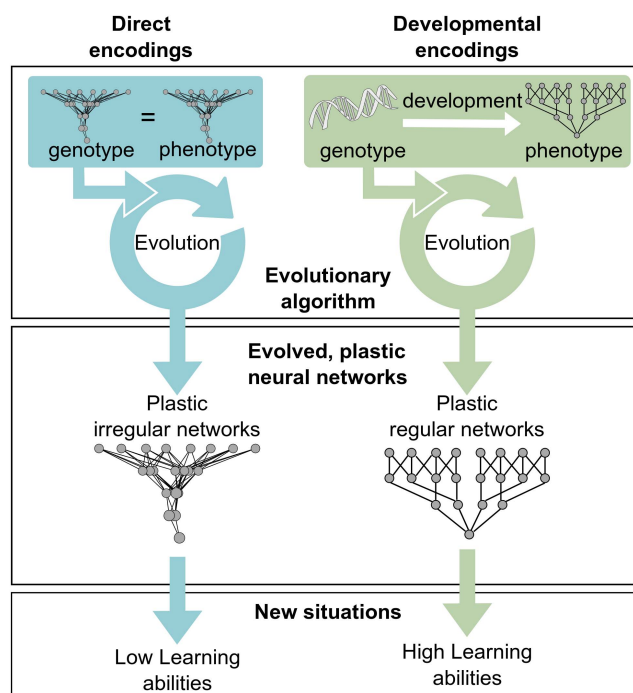


Figure 1: The hypothesis of Tonelli and Mouret is that the usage of developmental encodings during application of evolutionary algorithms leads to more regular neural networks with better learning abilities. ¹

The better general learning ability of more regular ANNs is supposed to stem from higher redundancies of network parts as the ANN can't be as specialized to the specific test cases used by the fitness function as a direct encoded network would be.



2 Background Information

This chapter presents some concepts that are necessary to understand the following experiments. First of all in chapter 2.1 a measurement for regularity is defined to

¹Source: [3] p.3

study ANNs in respect to that regard. Then in chapter 2.2 the two classes of possible genetic encodings for ANNs and three representatives of these are described. And at last in chapter 2.3 the Skinner box is introduced.

2.1 Regularity

Regularity measures the compressibility of a network. It is a characterization of the phenotype, not the genotype, of the ANN. Tonelli and Mouret compute the regularity by calculating the number of symmetry axes (cf. [3] p. 4). If two parts of the network can be interchanged without altering the graph they are called symmetric. It is only necessary to describe one of these parts and reference it for each additional occurrence, therefore the description of a network with more symmetries is more compressible.

These symmetries are graph automorphisms and can be easily calculated for **most** graphs (not all graphs since it is actually an NP-complete problem).

A graph automorphism is a permutation σ of the vertex set V of the graph $G = (V, E)$ that maps an edge $(\sigma(u), \sigma(v)) \in E$ to the vertices u and v iff² they formed an edge (u, v) already in G .

A permutation of a set is a particular ordering of all its members. For example all permutations of the set $\{1, 2, 3\}$ are $(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1)$. If the vertices of G are fixed in a particular ordering, the graph automorphism can be written as mapping from the previous ordering to the new one like this: $\begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}$ where the original ordering was $(1, 2, 3)$ and the automorphism mapped the vertices to $(2, 1, 3)$.

Since an automorphism of the graph to itself always exists, the number of symmetries is the number of automorphisms minus one.

In figure 2 possible graph automorphisms on an equilateral triangle are shown. These graph automorphisms are the following:

- Identity: $\begin{pmatrix} a & b & c \\ a & b & c \end{pmatrix}$
- Reflection on the perpendicular between b and c : $\begin{pmatrix} a & b & c \\ a & c & b \end{pmatrix}$
- 120° Rotation clockwise: $\begin{pmatrix} a & b & c \\ b & c & a \end{pmatrix}$

2.2 From genes to nervous systems

It is necessary to encode the structure of an ANN to a representation that can be modified easily if an evolutionary algorithm is to be used for its development.

²if and only if

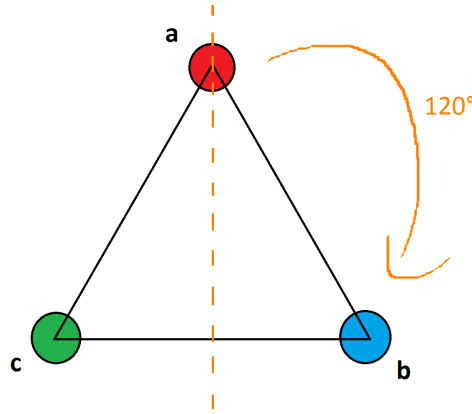


Figure 2: Equilateral triangle with symetrie axis through a and rotation of 120° clockwise drawn in

Encodings can be divided into two groups:


- **Developmental encodings** use a genotype to phenotype mapping like DNA does. The code defines construction rules which if followed lead to a certain network structure. Small changes in the code can have huge impacts on the resulting phenotype allowing a fast exploration of parameter space, but the fine tuning becomes difficult. Often similar and symmetric network structure parts arise in the phenotype since several neurons and connections are affected by the same rules.
- For **direct encodings** the mapping from genotype to phenotype is immediate as the parameters to each neuron and each connection can be directly derived from the code. In the developmental encoding this is not possible since they depend on their relation to different network parts by the defined rules.

In the studied paper three different encodings are compared with each other, two of them are developmental encodings (HNN and map-based) and one direct encoding.




2.2.1 Direct encoding

The ANN is described by a directed graph. The networks of the first generation of the evolutionary algorithm are initiated as a simple feed-forward network without hidden layers and randomly initialized weights.

Seven different mutation operators are modifying the network during evolution while no crossover is implemented in the version used by Tonelli and Mouret (cf. [3] p. 9-10).  mutation happens with a specific probability to the operation:

- A connection is randomly added
- Randomly choose a connection and remove it

- Randomly choose a connection and change its target or source
- Create a new neuron by randomly choosing a connection and creating a new target neuron with the same connection weight
- Randomly choose a neuron and delete it and its connections
- Randomly choose a connection and modify its weight
- Activation function of a random neuron is changed 

2.2.2 map-based encoding

The map-based encoding develops a network structure in a similar way as the direct encoding does, but instead of directly setting neurons and their connections, nodes and edges are used. Nodes and Edges of the developed neural net are however encoded in a specific, more complex way. Because of this part of the encoding it doesn't behave as a direct encoding, but as a developmental encoding.

Each node of the network represents a whole map of neurons. A map of neurons is an array of $N \times M$ identical neurons and is defined by a set of parameters (cf. [2] p. 3). In the map-based encoding in the studied paper each parameter value is a real number between 0 and 1 and is mutated during the evolutionary process. The meaning of a parameter 'flips' by exceeding a threshold value (for example: a parameter value between 0 and 0.4 might mean 'a' and a value between 0.4 and 1 might mean 'b'). One parameter of the neuron map for example decides if it represents only a single neuron or a set array size of neurons.

The edges between maps are described as well with a number of parameters. In this case the type of connection and the synaptic weight are set by parameters. One parameter decides if each neuron of a map is only connected to one neuron of the map on the other side of the edge or if each neuron is connected to each other neuron. Another parameter implies if the connection weights have positive or negative sign (excitatory or inhibitory). And the last parameter sets the weight value.

2.2.3 HNN encoding

The Hyper Neural Network (HNN) encoding used by Tonelli and Mouret is a modification of the complex HyperNEAT encoding. The HyperNEAT encoding is used by other researchers as well to study the effect of developmental encodings on plasticity in ANNs [1] [4].

When using this encoding the network structure has to be specified beforehand (in this case 9 input neurons, 5 hidden neurons and 4 output neurons). During the evolutionary development of the network two additional included networks are formed. One of them answers the question to each pair of neurons, if they are connected or not and if so which connection weight the edge has. The second inherent network defines the parameters of each neuron. These two networks are developed using the direct encoding. Through the interaction of the two inherent

networks with the given main-network structure the HNN encoding meets the requirements of a developmental encoding.

2.3 Skinner box

A Skinner box is a typical experiment to test the learning ability of an animal. The standard configuration of a Skinner box can be seen in figure 3, it can be described with input and output items. The test animal has to associate the inputs with the appropriate output. As stimuli input, lights or sounds from a loudspeaker might be used. The test animal is then supposed to press the associated response lever (as output) according to the stimulus. The association patterns are set arbitrarily by the experimenter. For example the test animal might be supposed to push the lever if the green light appears, but not if the red light shines. As additional input the food dispenser and electrified grid can then be used to support the 'right' behaviour and/or punish the 'wrong' behaviour.

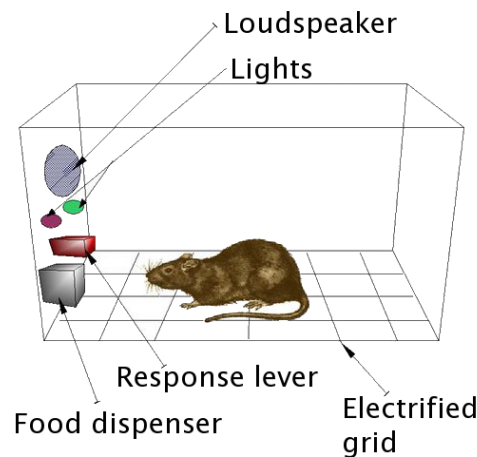


Figure 3: Model of a Skinner box ³

A test animal (or different test subject) with good general learning ability, should be able to identify the supported input to output associations independent of the specific used pairing and act accordingly.



3 Approach description


To test their hypothesis that developmental encodings lead to ANNs with better general learning ability, Tonelli and Mouret used the following experimental setup:

Developed ANNs are trained and tested in a simulated Skinner box with four stimuli inputs (1,2,3,4) and also four possible output actions (A,B,C,D). Only one

³Source: Author: Andreas1, URL: https://commons.wikimedia.org/wiki/File:Skinner_box_scheme_01.png#

stimulus is presented at once and is associated with one desired output action. For further consideration four terms are defined by Tonelli and Mouret:

-  **association** (I,O) is a pair of input stimulus and output action, where the input I should lead to the output O (e.g. (1,A)).
- The **association set** is a set with one association to each input stimulus (e.g. $\{(1, A), (2, B), (3, C), (4, D)\}$)
- The **global training set** is the set of all possible association sets (with four different stimuli and four possible actions there are $4^4 = 256$ different possibilities).
- The **evolutionary training set** is a randomly chosen subset of the global training set, that is used during a particular experiment to evaluate the general learning ability of the evolved ANNs.
- With the **General Learning Ability** (GLA) score the performance of an evolved ANN on all association sets, that were not used during the evolutionary process. 

The ANNs that are developed have ten input and four output nodes. Four  input nodes for the four possible input stimuli, four for a copy of the last output of the ANN and one each for reward and punishment. The figure 4 shows an illustration of the network structure.

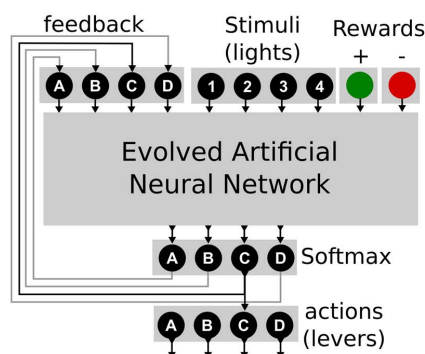


Figure 4: Model of a simulated Skinner box with an ANN as test subject.⁴

To calculate the answer of the ANN to an input the softmax of the output values from the network is taken. The *softmax* in contrast to the simple *max* function does not just take the output node with the highest value as answer of the ANN, but assigns a probability to be taken proportional to the quantity of a value (higher value leads to higher likelihood to be chosen as final answer).

An example of the first steps to the calculation of the fitness of an ANN in respect to the association (3,B) is presented in figure 5

⁴Source: [3] p.3

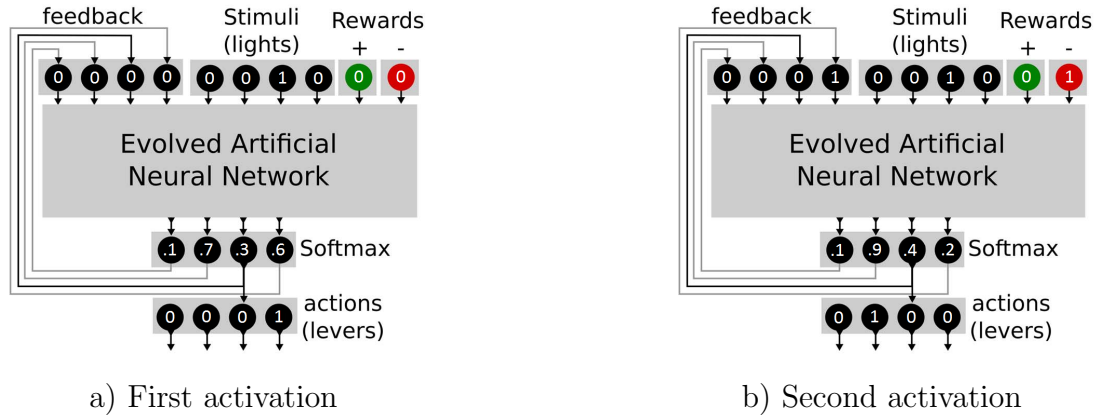


Figure 5: Let the currently considered association be (3,B).

a) In the beginning the feedback- and reward-input are set to 0, the network calculates its answer to the stimulus and with the softmax function one particular action is chosen. By chance not the ANN output with the highest value is chosen by the softmax function.

b) after the initial answer to the stimulus got derived it is together with positive or negative reward given as additional feedback.⁵

The learning process of an ANN to an association set is not done with classical backpropagation and corresponding weight changes, but by using modulatory neurons. Each neuron in the ANN can be a standard or a modulatory one. Modulatory neurons change the connection weights of standard neurons using Hebb's rule ("Cells that fire together, wire together") and induce thereby plasticity to the ANN (cf. [3] p.2).



A run of the experiment

To start a new experiment a subset of the global training set is randomly chosen as evolutionary training set. An evolutionary algorithm is then used to create ANNs with good performance on the training set for each of the three different genetic encodings studied by Tonelli and Mouret: direct, map-based and HNN encoding. As the name suggests direct encoding is a representative of the direct encodings introduced in chapter 2.2, whereas map-based and HNN encoding are among the developmental encodings. The 'Non-dominated Sorting Genetic Algorithm-II' (NSGA-II) a multi-objective optimization algorithm that utilizes evolutionary methods, is the selected evolutionary algorithm by Tonelli and Mouret for their experiments.

It is necessary to define a fitness function to evaluate the performance of individuals during the evolutionary process. The fitness function used by Tonelli and Mouret does the following:

First an association set of the evolutionary training set is chosen. The **modulated**

⁵Source: Based on [3] p.3

connection weights in the ANN are randomly initiated, thereby the ANN forgets previous knowledge. The ANN is then supposed to adapt to each association of the association set, therefore the stimulus is initially presented without feedback or reward input, after the first action to that stimulus was selected the feedback and reward inputs are set for additional iterations while still showing the same stimulus. This procedure is done for each association of each association set of the training set. The fitness score is then the mean value of the number of positive rewards (correct associated actions) at the end of each procedure.

The application of the evolutionary algorithm generates a set of ANNs with decent fitness score on the evolutionary training set. For the ANN with highest score the GLA is calculated. Therefore the fitness function is computed on the association sets that weren't in the training set.

Finally the regularity of the obtained ANN is computed. As described in chapter 2.1 this can be done by evaluating the number of graph automorphisms in the phenotype of the ANN. The comparison of GLA with the number of automorphism can then be used by Tonelli and Mouret to study the relation between both values and thereby show proof or disproof to their hypothesis.

One experiment looks then like the following:

1. Choose a genetic encoding (direct, map-based, HNN)
2. Select random evolutionary training set
3. Develop ANNs with an evolutionary algorithm, which are performing well on the training set
4. Calculate the GLA of the ANN with the best fitness score, computing the fitness function on the unseen association sets of the global training set
5. Evaluate the regularity of the ANN by counting the graph automorphisms

4 Approach analysis

According to the hypothesis of Tonelli and Mouret about regularity inducing a better general learning ability in ANNs, the GLA should be positively correlated to the number of graph automorphisms. Also the usage of developmental encodings should lead to more regular networks and therefore ANNs with better general learning ability. Since plasticity in the ANNs can only be achieved by modulatory neurons it can be expected that the occurrence of at least one modulatory neuron is necessary for good GLA scores.

Indeed they come to the conclusion that well suited ANNs to the problem of the general learning task heavily depend on the usage of modulatory neurons (cf. [3] p.3-4). In fact they summarize: "[...] and we never found any solution with less than two hidden neurons (one of them being modulatory)." ([3] p.3)
A modulatory neuron has to utilize the rewards and feedback input values to

change connection weights if the reward signal corresponds to a punishment and the present weights if the previous action was correct.

Some of the results actually suggest that the correlation of the GLA score and the number of graph automorphisms isn't as strong as expected, since ANNs can be found, that achieve high GLA scores with only a few automorphisms in the network and some ANNs with many automorphisms only reach low GLA scores. Both problems can be explained without impairment of the hypothesis:

- The first case appears when direct encoding is used (cf. [3] p.5), in this encoding no mechanism is available to copy network subparts as it is in the developmental encodings, therefore the regularity might not be measured correctly with automorphisms because slight differences in otherwise very similar network parts mean that no suitable mapping of vertices for automorphisms are available. Direct encoding should still tend to produce regular ANNs, but the parallel network parts are developed step by step in the evolutionary algorithm, it is then unlikely that these actually quite similar network parts can be described with an automorphism. In the example in chapter 2.1 of an equilateral triangle this would mean that a small displacement of a node to one direction prohibits the possibility to use the mentioned reflections and rotations for a uniform mapping.
- The second case is about ANNs with many automorphisms but low GLA score. These are actually no counter proof to the hypothesis of Tonelli and Mouret since their proposal states that plasticity requires high regularity, but not that high regularity implies plasticity. ANNs can have high regularity but completely insufficient structure to solve the given task (cf. [3] p.5).

The figure 6 summarizes the results of the experiments. Plotted are data of ANNs with perfect fitness function score on their evolutionary training set. For A and B the X-Axis shows the minimum number of graph automorphisms of the ANNs, where a network with n automorphisms is included in the columns from 1 to n. In A the performance and in B the distribution of ANNs with more or less automorphisms is shown on the Y-Axis.

Two distinct observations can be made from the two graphs in figure 6:

- **Graph A:** The number of graph automorphisms in an ANN is positively correlated to its GLA score.
- **Graph B:** The developmental encodings (map-based and HNN) produce ANNs with more graph automorphisms than the direct encoding.

4.1 Discussion

Tonelli and Mouret conclude from shown results in figure 6: "[...] using a developmental encoding improves the learning abilities of evolved, plastic neural networks." [3] p.5

⁶Source: [3] p.7

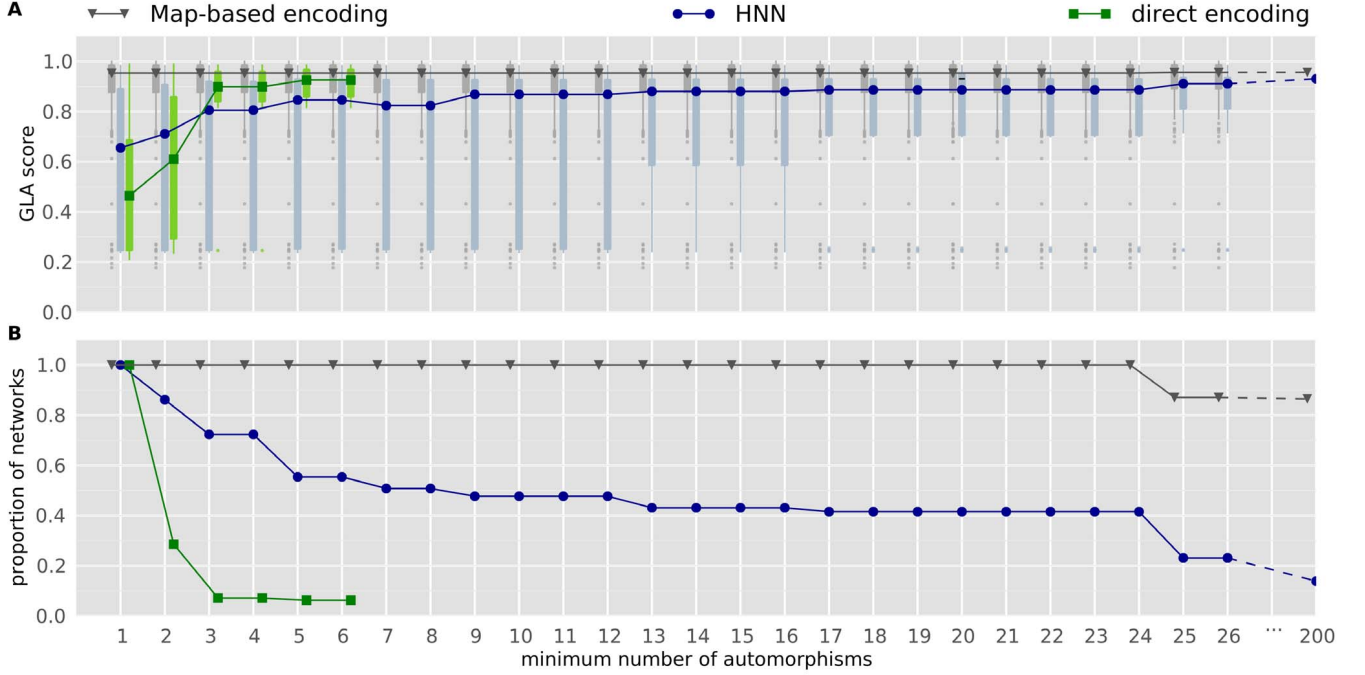


Figure 6: Results of the experiments. **a)** GLA score of ANNs is plotted in respect to the number of automorphisms they at least have for each encoding. **b)** The proportion of ANNs that have at least a specific number of automorphisms for each encoding.⁶

This is well justified, even though for ANNs with at least 3 graph automorphisms the ones with direct encoding achieve higher GLA scores than those with HNN encoding. This can be explained with the previously described problem of measuring regularity in directly encoded ANNs with graph automorphisms. In the rare cases where 3 or more automorphisms occur in ANNs with direct encoding their regularity is probably way higher than it seems.

The main hypothesis of the paper is therefore confirmed by the experimental data. The usage of developmental encodings did lead to ANNs with higher GLA score and more graph automorphisms than direct encoding. With one caveat in the relation of graph automorphisms and regularity. Since graph automorphisms represent symmetries in the graph the appearance of many graph automorphisms does imply high regularity of the network. The absence of them however doesn't imply irregularity, this is, as already described, because if little variations in otherwise similar network parts appear the measurement with automorphisms doesn't give the ANN credit for being regular in that regard.

To make secured conclusions about the superiority of developmental encodings over direct encodings in the task of producing regular networks, another measurement would be necessary.

Tonelli and Mouret write further: "To achieve this flexibility, they have to possess connections that were not directly selected during evolution." ([3] p.7)

This conclusion makes sense and should hold true, but there is actually no experimental data shown to back it up. In fact they themselves stated, that solutions with only two hidden neurons were found (cf. [3] p.3). With a measurement for the proportion of actually used connections in the ANN during the evaluation of its fitness, the assertion could be shown.

Since higher flexibility of the ANN should imply a higher amount of 'useless' network parts, a trade-off between flexibility and trainability of ANNs might exist (cf. [3] p.7). The used experimental setup however considered only a rather simple task (four to four association) to solve for ANNs (cf. [3] p.8). Therefore the shown dominance of ANNs with many graph automorphisms might be impaired for a more complex problem. A second experiment with higher demands on the ANNs could have answered this question.

A Problem of the used experiment is that the complexity of the networks weren't penalized. With a cost for connections an additional goal which requires to minimize the sum of connection costs could have been introduced (cf. [3] p.7). With this implemented it would have been possible to make statements about the efficiency of ANNs with high regularity (and high general learning ability) compared to ANNs evolved with direct encoding (and maybe same GLA score). Thereby the actual usefulness of regularity for animal nervous systems could be assessed.

5 Conclusion

Tonelli and Mouret were able to prove their hypothesis that developmental encodings lead to plastic ANNs with good general learning ability. They used an experimental setup in which ANNs were put into a simulated Skinner-box. An evolutionary algorithm was then utilized to find representative ANNs for three different genetic encodings that solve the task in a satisfying manner. Two genetic encodings were put in the category of developmental encodings, which exploit a more indirect way to describe network structures with high impact of little changes, and the last encoding was a direct encoding, where position and parametrization for each node and edge of the network is immediately accessible through the genetic code. Through a calculation of regularity by counting the number of graph automorphisms in the ANNs it was possible to evaluate the correlation of regularity, general learning ability and genetic encoding.

With the results of their results the main hypothesis was proved quite well, but some questions stay unanswered and could be topic of further research. Mainly the measurement of regularity could be refined, maybe the specification of multiple metrics focusing on different aspects of regularity would shed some light on the evolved regularity in direct encoded networks.

Furthermore it might be interesting to study the source of plasticity in ANNs from the structure of their networks in comparison with not so plastic ANNs.

To verify the attained results from Tonelli and Mouret a similar series of experi-

ments with a more complex problem to solve for ANNs and maybe with a penalization of big networks would be desirable.



References

- [1] Jeff Clune, Benjamin E Beckmann, Philip K McKinley, and Charles Ofria. Investigating whether hyperneat produces modular neural networks. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 635–642. ACM, 2010.
- [2] Paul Tonelli and Jean-Baptiste Mouret. Using a map-based encoding to evolve plastic neural networks. In *Evolving and Adaptive Intelligent Systems (EAIS), 2011 IEEE Workshop on*, pages 9–16. IEEE, 2011.
- [3] Paul Tonelli and Jean-Baptiste Mouret. On the Relationships between Generative Encodings, Regularity, and Learning Abilities when Evolving Plastic Artificial Neural Networks. *PLoS ONE*, 8(11):e79138+, November 2013.
- [4] Phillip Verbancsics and Kenneth O Stanley. Constraining connectivity to encourage modularity in hyperneat. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 1483–1490. ACM, 2011.