# Disentangling Cognitive and Constructivist Aspects of Hierarchies

Stefano Bennati
*SOMS Group, ETH Zurich, Clausiusstraße 50, Zürich, Switzerland*

## 1 RESEARCH PROBLEM

One of the most puzzling problems in the social sciences is the emergence of social institutions. The field of sociology is trying to understand why our society is the way we know it and whether an alternative, possibly better, society would be possible.

One of the fundamental questions is the emergence of hierarchies. On the one hand the cognitive approach suggests that hierarchies are encoded in human nature, therefore are the most natural form of organization; on the other hand the constructivist approach sees hierarchies as a product of interactions between individuals that emerges independently of individual preferences.

We will investigate under which conditions hierarchies emerge from a cognitive factor, a constructivist factor or a combination of both.

We will study this question both on the analytic level, with the help of Agent-Based Modeling, and on the empirical level by running sociological experiments in our laboratory.

## 2 OUTLINE OF OBJECTIVES

The main objective is to understand what are the processes that favor hierarchies in our society. Quoting Herbert Simon (Simon, 1977):

> It is a commonplace observation that nature loves hierarchies.

One could think that we are naturally biased towards preferring hierarchies because of physiological factors such as the structure of the brain. We know that some regions of the brain are specialized for a specific function (e.g. speech processing, vision, etc.) (Lashley, 1929) but it is not clear how they are organized and interact to produce the Mind. If there were evidence for a hierarchical structure of the brain, for example a "controller" zone that coordinates the functioning of the other zones, one could speculate that preference for hierarchies emerges from human nature (Damasio, 1999).

Neural networks are a sophisticated computational model inspired by the brain so they are a good place to start looking for evidence of hierarchical organization. Once we get results on this fundamental puzzle, we can extend our findings from the neural network to society and produce hypotheses to be analyzed by means of agent based simulations and tested in a laboratory setting.

## 3 STATE OF THE ART

Despite the intensive research in this direction, understanding the emergence of hierarchies and the reasons for their diffusion is still an open problem: research shows that hierarchies help to solve the coordination problem, collaboration problem or both (Halevy et al., 2011). Traditional mathematical models of society describe it in a top-down fashion, although helping to understand why hierarchies are useful they cannot explain why hierarchies are there in the first place, why would individuals voluntarily delegate their decisions to a leader.

In contrast the bottom-up approach of Agent-Based Models (Axelrod, 1980) tackles the problem from the individual's perspective and tries to explain social institutions as an emergent phenomenon from interactions between individuals. Bottom-up analytical models have been used to study the conditions that favor hierarchy over egalitarian society (Gould, 2002). The limitation of this research is in the assumptions: agents are either assumed to be specimens of the perfect individualist "Homo Economicus" or other-regarding and empathetic "Homo Socialis".

These assumptions are usually arbitrary so the models are not able to explain why would people develop other regarding preferences in the first place. Quoting Axelrod (Axelrod, 1997):

> But if the goal is to deepen our understanding of some fundamental process then simplicity of the assumptions is important and realistic representation of all the details of a particular setting is not.

Our model will keep the bottom-up approach - without which emergent phenomena cannot be understood - and take advantage of the power of agent based modeling but it will differ from previous attempts e.g. (Helbing and Yu, 2009) by taking into consideration not only interactions between arbitrary agents, but also evolutionary, cognitive and social processes that shape the agents. We will model our agents by means of neural networks (NN), a powerful Machine Learning tool, inspired by biological nerve systems. Neural networks are Turing complete (Siegelmann and Sontag, 1991) (Siegelmann and Sontag, 1995) so they can learn any function starting from a random - unbiased - initial configuration. NNs are composed by interconnected neurons that implement a function and respond to inputs according to it, while weights of connections between neurons define how the network responds to inputs.

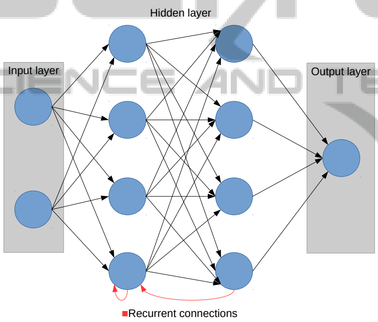


Figure 1: An example of neural network with two inputs and one output. This is a recurrent neural network because it has recurrent connections.

NNs can be trained to perform any task by presenting them examples and letting them adjust the connection weights; the difference between produced and expected output is propagated through the network and weights are adjusted in order to minimize it. The weights of a trained network are the product of simple learning rules applied on a series of input patterns, so they are an "emergent feature" of the trained network (Pessa, 2009). The use of NNs in simulations of society is not unprecedented: i.e. Duong et al. used Feed-forward NNs to study emergence of stereotypes (Duong and Reilly, 1995).

Feed-forward NNs are the simplest type of network, where inter-neuron connection cannot form directed cycles: information flows linearly from the input to the output, self- and backward connections between neurons are not allowed. In our simulation we will use Recurrent NNs instead, which allow for directed cycles and more sophisticated training mechanisms. The advantage of Recurrent NNs is that they can have memory, therefore can solve problems that Feed-Forward NNs cannot solve (Schmidhuber, 1990).

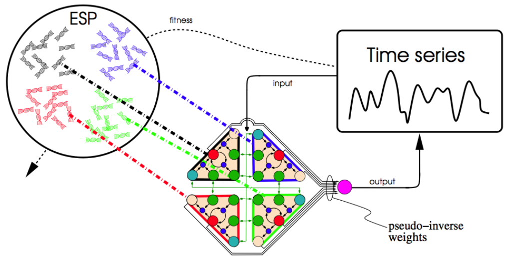


Figure 2: Network divided into subpopulation. Each subpopulation co-evolves by Enforced SubPopulation (ESP) to specialize in a specific sub-function. Courtesy of (Schmidhuber et al., 2007).

In particular we will use the Evolino framework (Schmidhuber et al., 2007) to train our agents. The idea behind Evolino is that most problems can be decomposed into a linear part and few non-linear components. Optimal linear methods such as linear regression are not able to deal with non-linearities but perform much better on the linear part than suboptimal non-linear methods, Evolino solves this issue by combining analytical linear methods and neuroevolution: the Enforced SubPopulation (ESP) neuroevolution method (Figure 2) coevolves subpopulations of neurons to accelerate the specialization of neurons into sub-functions. Subdividing neurons allows for specialization because intra-population mating is not allowed so each subpopulation evolves independently. Populations of neurons specializing in subtasks is something that is also found in nature, in fact our cortex is divided in spatially-contiguous regions that perform a specific task, such as vision or language processing (Lashley, 1929). This specialization has an interesting parallel with the concept of "Division of Labour" proposed by Adam Smith: productivity is increased if cooperating individuals specialize each in a specific task.

Recent research showed that hierarchical neural network perform better than an equivalent fully connected network on a sequence of movement tasks as different timescales emerge in the different parts of the hierarchical networks, suggesting that the upper level exerts open-loop top-down control on the lower level (Paine and Tani, 2005). Moreover they find that the same timescales emerge in the fully connected network as well, although the topology of the network does not evolve to replicate the hierarchical structure, leading to a lower performance; they attribute this failure to the excessive size of the weight space.

# 4 METHODOLOGY

The incremental structure and the long term goals of the project allow for breaking up work in subprojects that build up on one another's results. The first subproject describes the near-term goal of finding under which conditions neural networks self-organize in a hierarchical fashion, according to what the cognitive approach claims. Subprojects two and three define the medium- and long-term goals that extend the scope of the project to societies of agents. The former will allow us to study the impact of the constructivist factor on the creation of hierarchies, the latter is a combination of the two previous experiments that will help us finding how the two factors coexist.

## 4.1 Subproject 1

Starting from the results in (Paine and Tani, 2005) we want to study whether hierarchies are an emergent properties of networks. The factor that in the original paper didn't allow a hierarchical topology to emerge was the weight space, to keep it limited we would have to decrease the number of nodes in the networks, although having only few neurons has a negative impact on performance. A way to keep the number of connections low and allow for good performance is to abstract the concept of network and create a meta neural network (MetaNN), a fully connected NN where nodes are networks instead of neurons: each node is a fully connected neural network and between each pair of nodes there is a bottleneck, one input and one output connection, exactly as in the original paper.

## 4.2 Experiment

To understand the organization of NNs we need to have access to the structure of the network (i.e. the connection weights). The main disadvantages of NNs is that it is very difficult to understand what the network is actually doing because of the unpredictable modifications of weights and of the high number of connections between the neurons. For this reasons NNs are usually trained on a specific task and treated as black boxes.

To walk around this problem we will generate a meta neural network: a NN whose nodes are themselves full fledged NNs, this means that the function that nodes implement is not a static step function but changes over time. The drawback is that nodes have to be trained one by one, refer to section 4.2.2 for details on the training. Like in a normal NN the inter-node connection weights represent interaction between nodes, so a MetaNN can be trained and tested exactly like a normal NN. The advantage of this approach over a single network with the same total amount of neurons is that it can be inspected much easier: we will consider only inter-node connections which are in the order of the number of nodes, instead of being in the order of neurons, and derive conclusion on the organization of the network from their values. By doing so we assume that each node specializes in a subfunction, in agreement with the organization of the brain, and they somehow need to coordinate and merge their outputs together.

### 4.2.1 Task

Evolino divides the NN in subpopulations to accelerate specialization in subproblems and increase efficiency, the same is true for the brain. For this reason we expect the MetaNN to do the same: each node would specialize in a specific subtask. Subtasks can be more or less self contained and independent, depending on the task.

To make our analysis easier, we will ask the NN to solve a classification problem and perform different independent subtasks: algebraic operations of two numbers, driven by an input value that specifies what operation should be executed. We would expect each NN to specialize in one of the subtasks and one doing the classification. This will speak for a hierarchical organization with a master node doing classification and asking another node to perform the operation it is specialized in. For a different task where subproblems are not so well defined, we might expect the network to process in a more distributed way, without hierarchies.

We can infer what type of organization was learned by the network by looking at the interconnection weights:

- If only one node has a positive weight with the input but weigh 0 to the output, we can assume that it is doing the classification and instructing the other networks.
- If interconnection weights are 0 and every node has positive weights for inputs and outputs, we can assume that each node is computing the full problem by itself.
- If interconnection, as well as input and output weights are positive we can assume that the computation is done in a collaborative way.

In certain circumstances it could be useful, for the sake of understanding the contribution of each component, to disassemble the MetaNN and test each node singularly on a set of inputs to infer what function it is implementing.

#### 4.2.2 Design

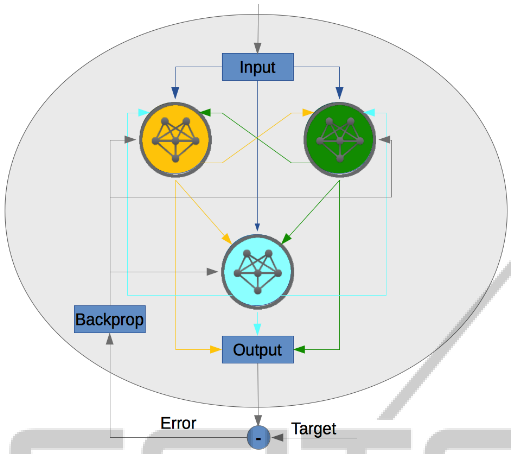


Figure 3: Structure of a meta neural network. Each node implements a neural network.

To avoid biasing the network towards hierarchy we will design it to have a symmetrical structure where nodes have all the same characteristics, we will introduce an input layer that forwards all inputs to each node and an output layer that averages the output of every node, weighted on the connection strength of the output link.

To allow for hierarchies to develop our MetaNN should have at least one node per operation plus one node for the classification, a smaller network size might not allow this specialization to happen. The hidden layer is composed by the nodes, each of which is a NN, and is fully connected: each node in the hidden layer has inputs coming from all other nodes and its output is fed into all other nodes. The size of the nodes should not have a big impact on the quality of the results, having bigger nodes might improve the quality of the output and the convergence time. Connections weights are initially random and are adapted through backpropagation.

The error is fed back to the backpropagation system of the MetaNN as well as to the backpropagation system of each node. Errors propagated to the MetaNN and to the nodes do not necessarily have to be the same, each node could have an error being the linear combination of the global error and the node's local error, computed with a special function that rewards positively the node if it computes one subtask.

Backpropagating the global error to the single neurons could be important to avoid oscillatory output, due to maximizing two independent functions at the local and global level. On the other hand the local error is useful to help nodes differentiate their output and specialize in a subtask. We will try different combinations of the two to find out what influence both errors have on the performance of the node and of the network.

We also believe that different timescales are a common denominator of many emergent systems in nature (Lemke, 2000) and they will help our network show emergent behavior. We will implement different time scales for training the nodes: for a certain number of iterations the nodes will be trained over their local error, based on their own output, in this phase each node will train towards the common goal. The second phase consists in training each node on the global error, based on the MetaNN's output that combines all outputs produced by the nodes. In this phase the nodes will be trained to work as a team, incentivizing specialization.

We expect that alternating local and global error will speed up convergence and help the network get out of local minima by cyclically reducing the distance between the state of each node, but at the same time fostering diversity and specialization.

We will try different combination of global and local error and values of timescale and study for which values the convergence time and error of the network are optimal.

## 4.3 Subproject 2

Typical simulations of society are designed as many agents that are able to interact with the environment and with their neighboring agents. This configuration can be replicated in a NN by assuming that nodes are agents: nodes receive input from outside the network - the environment - and produce output on which they are evaluated. They also interact with other nodes through inter-node connections whose weights determine the strength of the interaction - connections towards non neighboring nodes will have null weight.

We will modify the setting described in section 4.1 and adapt it to a bigger simulation where several agents are interacting with the environment and with other agents, such a design could be easily implemented in our framework by creating a new layer, a Meta-MetaNN, where nodes (agents) are MetaNNs. Typical settings allow agents to relocate and obtain a new set of neighbors, in our case that translates into changing the connection weights. We will also vary the environment configuration and payoffs to study the dynamics of the system and the emergence of social institutions.

### 4.4 Subproject 3

We will study more in detail the influence of social interactions in the emergence of hierarchies by creating isolated societies and varying the number and strength of the connections between them. We are planning on running a distributed simulation in our cluster to support a higher number of agents. We hope to find correlation between the interconnectivity of population and changes in their organization. From these findings we will formulate hypotheses and design a laboratory experiment to test them, one possible domain of application could be opinion dynamics and voting. This simulation will be the backbone for a series of future studies: i.e. one interesting issue that we want to investigate is how agent complexity influences model predictions. Previous studies (Axtell et al., 2000) shown the emergence of stereotypes by mean of a simple Agent-Based simulation. We will replicate their simulation with our framework and test whether above a certain threshold of complexity this result is not valid anymore.

## 5 EXPECTED OUTCOME

With this simulation we expect to find evidence for both a hierarchical and a distributed organization of the network. We expect not to find evidence for a non collaborative organization.

We will identify what are the situations and tasks that favor one or the other organization and look for parallels in the society.

We will implement a high scale simulation of society based on MetaNNs and use it to study emergent properties of the networks, we expect to find the same properties also at the level of society.

## REFERENCES

Axelrod, R. (1980). Effective choice in the prisoner's dilemma. *Journal of conflict resolution*, 24(1):3–25.

Axelrod, R. M. (1997). *The complexity of cooperation: Agent-based models of competition and collaboration*. Princeton University Press.

Axtell, R., Epstein, J. M., and Young, H. P. (2000). The emergence of classes in a multi-agent bargaining model.

Damasio, A. R. (1999). How the brain creates the mind. *SCIENTIFIC AMERICAN-AMERICAN EDITION-*, 281:112–117.

Duong, D. V. and Reilly, K. D. (1995). A system of iac neural networks as the basis for self-organization in a sociological dynamical system simulation. *Behavioral science*, 40(4):275–303.

Gould, R. V. (2002). The origins of status hierarchies: A formal theory and empirical test1. *American journal of sociology*, 107(5):1143–1178.

Halevy, N., Chou, E. Y., and Galinsky, A. D. (2011). A functional model of hierarchy why, how, and when vertical differentiation enhances group performance. *Organizational Psychology Review*, 1(1):32–52.

Helbing, D. and Yu, W. (2009). The outbreak of cooperation among success-driven individuals under noisy conditions. *Proceedings of the National Academy of Sciences*, 106(10):3680–3685.

Lashley, K. S. (1929). Brain mechanisms and intelligence: A quantitative study of injuries to the brain.

Lemke, J. L. (2000). Across the scales of time: Artifacts, activities, and meanings in ecosocial systems. *Mind, culture, and activity*, 7(4):273–290.

Paine, R. W. and Tani, J. (2005). How hierarchical control self-organizes in artificial adaptive systems. *Adaptive Behavior*, 13(3):211–225.

Pessa, E. (2009). Self-organization and emergence in neural networks. *Electronic Journal of Theoretical Physics*, 6(20):269–306.

Schmidhuber, J. (1990). *Dynamische neuronale Netze und das fundamentale raumzeitliche Lernproblem*.

Schmidhuber, J., Wierstra, D., Gagliolo, M., and Gomez, F. (2007). Training recurrent networks by evolino. *Neural computation*, 19(3):757–779.

Siegelmann, H. T. and Sontag, E. D. (1991). Turing computability with neural nets. *Applied Mathematics Letters*, 4(6):77–80.

Siegelmann, H. T. and Sontag, E. D. (1995). On the computational power of neural nets. *Journal of computer and system sciences*, 50(1):132–150.

Simon, H. A. (1977). The organization of complex systems. In *Models of Discovery*, pages 245–261. Springer.

## STAGE OF THE RESEARCH

Research is still at an early stage, we are developing the simulation environment, running the first simulation and collecting the first data