Are Neural Networks Imitations of Mind?
Gaetano Licata*
Gaetano Licata, Chair of Logic and Philosophy of Science, Dipartimento di Scienze Umanistiche, University of Palermo, Italy

Abstract
Artificial neural networks are often understood as a good way to imitate mind through the web structure of neurons in brain, but the very high complexity of human brain prevents to consider neural networks as good models for human mind; anyway neural networks are good devices for computation in parallel. The difference between feed-forward and feedback neural networks is introduced; the Hopfield network and the multi-layers Perceptron are discussed. In a very weak isomorphism (not similitude) between brain and neural networks, an artificial form of short term memory and of acknowledgement, in Elman neural networks, is proposed.

Keywords: Artificial neural networks; Feed-forward networks; Recurrent networks; Brain; Mind

Introduction
Nowadays we are out of the illusion that computers can be good models for human mind. The human mind is the result of the biophysical structure of a nervous system in a body which evolved to survive in the environment, in communication with other individuals of same species and in relationship with other species of the ecosystem: its power is due to a very long and hard evolution and we are not able to understand its complexity [1].

(A)The goal that A.I. should attain is the emulation, through a computer, of some processes of mind in relationship with the environment (the world and other individuals).

With respect to this objective I want to underline two obstacles in neural networks strategy: 1) and 2).

1) Neural networks are a strategy to emulate directly the behavior of brain and not the behavior of mind. Thus an important problem that neural network strategy misses is the gap between brain and mind. This is the problem of the translation of states of neuronal activation in concrete mental activity. The mind/brain translation problem will not be overcome until we will not have a clear theory about thought, consciousness, perception and action as cerebral phenomena. Moreover, if this theory wants to be useful to neural network strategy it must be conceived following the neural network philosophy and language. A theory who speaks the language of neural networks should consider thought (i.e. mental representations, planning, consciousness, memory and so on), perception and action not as “states” but as fluxes of states which go through the network (ordered and structured sets of states which go through the network). About these fluxes that we, as thinking brains, perceive in ourselves, we have unclear ideas on their beginning, on their developing and on their ending, but we know that perception can generate them.

2) Artificial neural networks are very poor imitations of brain. Human brain is a “network” of 100 milliard of neurons in which each neuron is connected to many thousands of other neurons, so, in a brain; there are millions of milliards of connections. There are many kinds of structure of neural networks, but the architecture of the most common neural networks consists in a simple three layers structure of artificial neurons, like the three layers “perceptron” of Figure 1, that henceforth I will call TLP.

Discussion
Neural networks can be feed-forward or feedback networks. In feed-forward neural networks like TLP the information propagates in only one direction, from input layer to output layer through the hidden layer (that can be more than one), and there are no cycles. Each unit is connected with every unit of the following layer, there are no connections between units of the same layer or with a unit of previous layer, and there are no connections which jump one (or more) layer(s). A feed-forward network simply calculates a function of input values which depend on the distribution of weights (w) of the incoming connections and on the activation function of the outgoing connection. It has not any internal state different with respect to the weights of connections.

In feedback networks (also called ‘recurrent networks’) the connections are arbitrary. The Hopfield network (Figure 2) is a fully connected graph, typically represented as a matrix of weights; it has bi-directional connections and symmetrical weights [2]. There is no input or output specific layers, all neurons are input and output units; activation levels are only +1 or -1. These kinds of network, with very high redundancy of connections, produce associative memory and permit the recovery of missing information.

Sometimes human brain behaves as a feed-forward network with layers, but it has also many connections that lead information backward to neurons of “preceding layer”, i.e. the brain is a feedback network in which can be many cycles of neurons. Given that sometimes the activation goes back to neurons which have caused it, feedback networks (and the brain) have an internal status memorized as activation levels of units. In recurrent networks the computation has much less order with respect to feed-forward networks. Artificial feedback networks can become unstable, chaotic or can fluctuate and it can be very hard to obtain a stable output from a given input; so it is a mystery how our brain, as feedback network, is able to produce its (so good) computation.

The learning process, in a neural network, is commonly understood

*Corresponding author: Gaetano Licata, Chair of Logic and Philosophy of Science, Dipartimento di Scienze Umanistiche, University of Palermo, Viale delle Scienze Ed. 12, 90128, Palermo, Italy, Tel: 339-456-8136. E-mail: ninnalicata@yahoo.it
Received February 23, 2015; Accepted March 11, 2015; Published March 13, 2015

Copyright: © 2015 Licata G. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

ISSN: 0974-7230 JCSB, an open access journal
Volume 8(3) 124-126 (2015) - 124
similar to those of learning process. The "training set" to administrate able to "make previsions", i.e. to give outputs on the basis of inputs and the output variables, so the network, after the learning process, is network learns the unknown relationship between the input variables of technological applications. In the case of "supervised learning" the algorithm [5], which has today a lot back propagation learning" with theory speaks about artificial learning.

A classical learning process for networks like TLP is the "supervised learning" with back propagation algorithm [5], which has today a lot of technological applications. In the case of "supervised learning" the network learns the unknown relationship between the input variables and the output variables, so the network, after the learning process, is able to "make previsions", i.e. to give outputs on the basis of inputs similar to those of learning process. The "training set" to administrate to the network contains typical examples of inputs with relative outputs (in ordered pairs); when all the training set is administrated to the network, the network will be able to associate, to a new input, the desired output with an error that the network can correct. The error in the output can be corrected through the comparison with the "expected output" (supervision). Given that the output is expressed as neural activation of output layer, the error is a difference of activation between the output proposed by the network and the "expected output". To delete this difference, according to the strategy of Rumelhart et al. [5], the information is propagated back from output layer to hidden layer (or to hidden layers), until the input layer. Step by step, backwards, the network modifies the weights and the activation functions which bind the units, to minimize the difference between the resulted activation of output layer and the activation desired in output. The network, in this kind of supervised learning, has power of generalization: it is able to deal with unknown cases knowing similar cases, as in natural logical induction.

As it is known the weight is the scalar parameter of synaptic "strength" of incoming connections: e.g., in Figure 1, the first arrow, from the first neuron on the left of input layer, to the first neuron on the left of hidden layer. An important incoming connection will have a high weight, while a less important connection will have a low weight. Thus the learning process is the creation of selected connections, between the units, from the input to the output: this is considered a similarity between this kind of neural networks and the brain. Supposing to draw, in TLP, some arrows fatter than others, we could have a representation, in a very smaller and simpler scale, of how the synapses of biological brains are reinforced or weakened by learning and memory. Is it the matter really so simple?

Although TLP is a neural network and it always works in parallel as a nonlinear function, its simplicity permits a linear explication of its state between a complete computation ("epoch") and another. Let us call the neurons of input layer, from left to right, I1, I2, I3; the neurons of hidden layers H1, H2, H3, H4; and the output units O1 and O2. Calling Φ the outgoing activation function of the units and Σ the weighted sum of incoming connections, we can write that, after the exposition of the network to an input, TLP will give the output O1 \& O2 having the state S:

\[ S = \Phi_O(\Sigma_1) + \Phi_H(\Sigma_2) + \Phi_H(\Sigma_3) + \Phi_H(\Sigma_4) \]

In this conjunction the difference between connections is given by the context of functions, e.g. the connection I1, in the context of \( \Phi_H(\Sigma_1) \), will have a different weight with respect to the I1 contained in \( \Phi_H(\Sigma_1) \); so the whole function \( \Phi_H(\Sigma_1) \), in the context of \( \Phi_H(\Sigma_1) \), will have a different value with respect to \( \Phi_H(\Sigma_1) \) contained in \( \Phi_H(\Sigma_1) \). This way to explicate linearly the state of TLP shows its computational order, the internal relationships between the connections and their values, and contains the idea that the activation function (Φ) of a neuron is like a point of view on the weighted sum (Σ) of its incoming connections.

**Conclusion**

To conclude this discussion about feedback and feed-forward networks, I want to introduce a hybrid class of neural networks which has interesting properties: the Elman networks (Figure 3). In 1990 Elman [6] proposed a recurrent form of feed-forward networks similar
to TLP creating bidirectional connections between the hidden layer and a layer which is "contextual" to input layer. The "contextual" layer has the same number of units of hidden layer and to its neurons is assigned 1 as constant weight. The contextual layer has the function to register the state of the hidden layer during the computation. Therefore the function learnt by the network will be based on new inputs and on the state registered in the contextual layer, so the network could learn which is the state to remember. The computational order of usual feed-forward networks is respected by the Elman networks, and a certain kind of artificial "short term memory" is realized by the specific and controlled feedback cycle. Moreover, given that the copy layer is like an alternative input layer, the back-propagation algorithm can be employed in this kind of networks, and this is a great advantage because the back-propagation algorithm is a good learning technique and it cannot be usually employed in recurrent networks [7].

Suppose that a self-driving car, equipped with an Elman network, receives as inputs not only the current perception but also recently past states, registered in the contextual cluster; it is possible to hypothesize that a simple "comparison mechanism" allows the network to identify states, registered in the contextual layer, so the network could learn which is the state to remember. The computational order of usual feed-forward networks is respected by the Elman networks, and a certain kind of artificial "short term memory" is realized by the specific and controlled feedback cycle. Moreover, given that the copy layer is like an alternative input layer, the back-propagation algorithm can be employed in this kind of networks, and this is a great advantage because the back-propagation algorithm is a good learning technique and it cannot be usually employed in recurrent networks [7].

**References**