THE LABOUR-ORIENTED, COLLECTIVE INTELLIGENCE OF OURS: PENN TABLES 9.1 SEEN THROUGH THE EYES OF A NEURAL NETWORK

Manuscript, total wordcount: 9698

Author: *Krzysztof Wasniewski*

Affiliation: *The Andrzej-Frycz Modrzewski Krakow University, Faculty of Management, Krakow, Poland*

Mailing address: *Andrzej Frycz Modrzewski Krakow University, Gustawa Herlinga-Grudzińskiego 1, 30-705 Kraków, Poland*

Email: *kwasniewski@afm.edu.pl*

Phone : +48 601 48 90 51

Abstract

This article studies the surprising phenomenon, signalled by the World Bank, of digital technologies contributing to an increase in available jobs rather than their elimination. A method of studying social systems is introduced, as instances of collective intelligence, where quantitative, macroeconomic variables, in a given place at a given time can be treated as numerical expression of cumulative outcomes in processes of social change, made of collective decisions and collective action, having taken place up to the moment of observation. In its empirical part, the article presents an attempt at using that method to deconstruct collective intelligence expressed in Penn Tables 9.1. Empirical findings thus obtained substantiate the hypothesis that economic systems optimize themselves, as collective intelligence, so as to adapt real output and price levels to the average workload per person, recurrent in the given national economy.

JEL: E01, E17, J01, J11

Keywords: collective intelligence, artificial intelligence, labour market,

Introduction

The latest **World Development Report (WDR)** by World Bank (World Bank 2019) brings interesting observations as for the impact of technological change on the labour market. Apparently, at least so far, the fears of robots and digital technologies sweeping millions of jobs out of existence are unfounded. WDR 2019 brings evidence that not only don't those new technologies destroy jobs, but they also create whole new job categories and new sub-markets for human labour. Still, developing human capital is crucial: new jobs can appear, accompanying new technologies, when people acquire new skills. The most likely scenario for the global labour market is that, whilst new technologies do can create new jobs, people will have to adapt deeply in their lifestyles, including education, and cross-jobs mobility. The already proverbial 'uberization' of the labour market is a fact.

The impact of digital technologies upon the labour market can be expressed in an otherwise simple way, as correlation with two coefficients, namely that of number of hours worked per person per year, as well as the share of labour in the national income. The greater those coefficients, in the presence of technological change, the more 'job-making' this change is, and vice versa. Whilst unemployment rates obey pretty clear rules, the coefficient of hours worked per person per year is much more idiosyncratic across countries, both in terms of weeks worked, and that of hours worked per week (Fuchs-Schündeln et al. 2017). The coefficient of hours worked per person per year is correlated with technological change. More specifically, it seems being associated with the ratio of substitution between capital and labour. That ratio seems to be growing over time, and we seem to be facing a paradox: as technological change becomes faster, people work more per year. In general, between 1948 and 2009, the response of hours worked to technology shock has been varying in its shape, every 30 years, as if in a Kondratieff cycle (Cantore et al. 2017).

The coefficient of hours worked per person per year is informative both about the technologies used by people working, and about the workstyles built around those technologies. Interestingly, that coefficient seems to be inversely correlated with income, still, the correlation is rather international than intranational. When developing countries are compared with the developed ones, the average adult in the former category works about 50% more per week than the average adult in the latter group. However, inside countries, the same type of correlation reverts, and one sees richer people work more than the relatively poorer ones (Bick et al. 2018).

Somewhat in the background of research on the strictly speaking coefficient of hours worked, another question arises: do we adapt our balance between workload and leisure to our expected consumption, or maybe we do something else, i.e. form a balance between consumption and savings on the grounds of what the local labour market imposes on us? Interestingly, both approaches may be quantitatively robust (Koo et al. 2013). Research focused on the impact of institutional changes, e.g. new environmental regulations, upon the labour market suggest that the latter is much more adaptable than in was assumed in the past, and has the capacity to offset the disappearance of jobs in some sectors by the creation of jobs in other sectors (Hafstead & Williams 2018).

Since David Ricardo, all the way through the works of Karl Marks, John Maynard Keynes, and those of Kuznets, economic sciences seem to be treating the labour market as easily transformable in response to an otherwise exogenous technological change. It is the assumption that technological change brings greater a productivity, and technology has the capacity to bend social structures. In this view, work means executing instructions coming from the management of business structures. In other words, human labour is supposed to be subservient and executive in relation to technological change. Still, the interaction between technology and society seems to be mutual, rather than unidirectional (Mumford 1964, McKenzie 1984, Kline and Pinch 1996; David 1990, Vincenti 1994). The relation between technological change and the labour market can be restated in the opposite direction. There is a body of literature, which perceives society as an organism, and social change is seen as complex metabolic adaptation of that organism. This channel of research is applied, for example, in order to apprehend energy efficiency of national economies. The so-called MuSIASEM model is an example of that approach, claiming that complex economic and technological change, including transformations in the labour market, can be seen as a collectively intelligent change towards optimal use of energy (see for example: Andreoni 2017; Velasco-Fernández et al 2018). Work can be seen as fundamental human activity, crucial for the management of energy in human societies. The amount of work we perform creates the need for a certain caloric intake, in the form of food, which, in turn, shapes the economic system around, so as to produce that food. This is a looped adaptation, as, on the long run, the system supposed to feed humans at work relies on this very work.

What if economic systems, inclusive of their technological change, optimized themselves so as to satisfy a certain workstyle? The thought seems incongruous, and yet Adam Smith noticed that division of labour, hence the way we work, shapes the way we structure our society. Can we hypothesise that technological change we are witnessing is, most of all, a collectively intelligent adaptation in the view of making a growing mass of humans work in ways they collectively like working? That would revert the Marxist logic, still, the report by World Bank, cited in the beginning of the article, allows such an intellectual adventure. On the path to clarify the concept, it is useful to define the meaning of collective intelligence.

Collective intelligence, as a theoretical construct, might be an interesting, and developmental compromise between agent-based theories in economics, and models of general equilibrium. Some literature suggests the need of such a synthesis (see for example: Babatunde et al. 2017). Yet, collective intelligence is a hypothesis rather than a theory. On the one hand, since Emile Durkheim, social sciences acknowledge the potent impact of cultural constructs upon human behaviour. Yet, defining intelligence in itself is tricky. Lavniczak and Di Stefano (2010) claim that intelligence requires the existence of a cognitive agent, which, in turn, should be autonomous, i.e. able to interact with its environment and with other agents. This autonomy translates into 5 specific cognitive functions: a) perceiving information in the environment and provided by other agents b) reasoning about this information using existing knowledge c) judging the obtained information using existing knowledge d) responding to other cognitive agents or to the external environment, as it may be required and e) learning, i.e. changing (and, hopefully augmenting) the existing knowledge if the newly acquired information allows it. Among those six characteristics (i.e. one general and five specific ones), when studying collective intelligence in human societies, we can unequivocally tick just the first box, and the last one, i.e. perception, and the capacity to learn. When we study collective intelligence possibly represented by quantitative variables, aggregated into databases, those sets of numbers are supposed to be representative for a complete society, global, continental or national. How can it be autonomous? Completely, on the other hand, and not at all, on the other hand. Do we communicate with other civilisations? Probably not yet, although we communicate with the natural environment. As for collective reasoning, judgment and response, these are arguably black boxes. Manifestations of social behaviour, such as, for example, the average wage, change from year to year, which suggests that some people do something in response to something else, and those people belong to a collective intelligence. Doing something meaningful implies, most likely, some reasoning and some judgment, and yet we do not have a clear map of these functions at the collective level. We suppose it happens, but we don't know how.

If we approach collective intelligence as entanglement of individual nervous systems, assuming the existence of autonomous cognitive agents is questionable. There are strong theoretical and empirical grounds for treating any neural function in a human brain as a network occurrence, where the construct of autonomous cognitive agent is a convenient illusion rather than a true representation of reality. The so-called 'Bignetti model' is one of the best-known approaches in this theoretical stream (Bignetti 2014; Bignetti 2018; Bignetti et al. 2017): whatever happens in observable human behaviour, individual or collective, has already happened neurologically beforehand.

This cursory review of the issue suggests an interesting question: how strong the assumptions formulated for studying collective intelligence should be? Two opposite, theoretical paths open up. On the one hand, we can go towards as weak assumptions as possible, in order to make the corresponding research immune to false ideas. The so-called 'swarm theory' is a good example (Stradner et al. 2013). Collective intelligence occurs even in animals as simple neurologically as bees, or even as the Toxo parasite; claiming autonomous cognitive functions in them is truly a strong assumption. Still, they manifest collective intelligence by shifting between different levels of coordination: from deterministically coordinated action (behaviour A in one specimen always provokes behaviour B in another specimen), through correlated action (behaviour A is significantly correlated with behaviour B), all the way up to randomly coupled action (behaviour A provokes a loose, apparently random range of responses in other specimens). The swarm theory claims that social learning is possible precisely by shifting between those different levels of coordination, and demonstrable shift of this kind is enough to assume that a society is collectively intelligent.

On the opposite end of the theoretical spectrum, we have approaches such as that presented by Doya & Taniguchi (2019). They claim that intelligence, whether individual or collective, is endowed with many layers. The capacity to reproduce something complex is one layer, arguably the crudest one. Developmental intelligence means the capacity to create new symbolic models of reality and figure out something practical on that basis. If we produce many different versions of empirical reality, the most functional one (i.e. the one that brings the best immediate reward) reflects the most basic level of collective intelligence. More abstract levels of intelligence can emerge, as we solve more and more complex problems through theory. Quantum physics are an excellent example: very abstract, symbolic representations of reality serve to solve very practical issues, such as the best way to make good steel. The distinction introduced by Doya and Taniguchi suggests two distinct levels of abstraction in studying the human collective intelligence at work. On the one hand, we can adopt the reductionist approach and test human societies for the presence of recurrent, swarm-like patterns of shifting coherence in social coordination. Down this avenue, we hypothesise the working of collective intelligence understood crudely and elementarily. On the other hand, elaborate cultural patterns, representative for developmental, abstract intelligence, can be subject to scrutiny. Empirical research presented further below in this article aims at verifying the **hypothesis that economic systems optimize themselves, as collective intelligence, so as to adapt real output and price levels to the average workload per person, recurrent in the given national economy**.

The method and the dataset

Studying collective intelligence requires analytical tools which allow to represent it. Artificial intelligence, i.e. neural networks, can be used to that purpose. Artificial intelligence can be approached as the general science of intelligence, as it allows testing abstract concepts we make as regards our own mind (Sloman 1993). AI is what we make it: any logical structure used in neural networks is a projection of the programmers' intuition as for how an intelligent structure should work. Quantitative variables regarding social phenomena can be considered as informative about the cumulative outcomes of past collective decisions in the given society, and it is possible to hypothesize collective intelligence in these decisions. Observing the way that a neural network processes empirical data can provide insights as for how the given society is collectively intelligent. The very same artificial intelligence, which enters into so surprising an interaction with human work, can be used to study societies in that interaction.

There are several examples in recent literature of neural networks being used as solvers in econometric models, i.e. as an alternative to stochastic regression. Accurate prediction in financial markets is a good example (see for example: Lee 2019). In this field of application, neural networks achieve good accuracy of prediction as, instead of averaging local errors of estimation into one 'big' standard error, they process each local error separately in a sequence of recurrent trials. Still, another approach is possible: neural networks can be used in order to simulate and observe how an intelligent structure works. Even a beginner's experience with neural networks brings that interesting, and somewhat stunning observation, that rational functions denominated over the expression '$1 + e^{-b*h}$', where 'h' is a complex vector of input variables, and 'b' is a parameter, have the peculiar capacity to optimize virtually everything, as long as the input data is standardized, so as not to exceed 1, and sufficiently high a number of trials is allowed. This strange capacity of some functions to optimize any desired outcome is so pronounced, that deep learning involves algorithms, such as the Gaussian mixture or the random forest, serving to slow down learning and prevent hasty conclusions (see for example: Zhang & Cheng 2003; Sinha et al. 2019). Apparently, even relatively simple specimens of neural networks can shed a completely new light on the ways that human societies work. In classical econometric and sociometric models, we tend to fall into the illusion of simultaneity, i.e. that everything works in the same time and more or less at the same pace. Neural networks are based on sequence and order in experimentation and thus force the researcher to think in the same terms, i.e. in terms of a sequence of events, endowed with hysteresis (dos Santos 2017). As for strictly spoken macroeconomics, the usage of neural networks to simulate economic change can be seen as one step further beyond the application of Extended Kalman Filter (see for example: Munguia et al. 2019), i.e. as a method to include noise and error as explicit, valuable information in empirical research.

The same capacity of neural networks to optimize on virtually any set of properly standardized numerical values leads to re-asking the question about the meaning of the numbers we use. It is a commonly known issue - as regards quantitative research in social sciences – that large databases are arranged very largely so as to assure internal coherence, even at the expense of accuracy in the measurement of real-life phenomena. Quantitative variables and their large collections in the form of databases are ontologies, and, as any ontology, they contain a large part of arbitrary (Munir, Anjum

2018). The tail side of that ontological coin is that a given ontology, i.e. a given methodology for defining quantitative variables, creates a controlled experimental environment for using a neural network as representation of collective intelligence. It is, in fact, the classical Laplacian approach: when two sets of empirical data have the same logical structure, i.e. they are both based on the same ontology, they can be considered as vectors subject to comparison for mutual similarity, and measures such as the Euclidean distance between them are informative about that similarity. In the here-presented research, the database known as **Penn Tables 9.1** (Feenstra et al. 2015) **has been used to create a set of 41 neural networks, which, in turn, has been an artificial, experimental environment for testing the hypothesis formulated in the introduction.** The use of Penn Tables 9.1. is largely based on observations made as regards its earlier versions, e.g. Penn Tables 8.0: this particular database seems to be truly robust as for measuring human capital and variables pertinent to the labour market (Fang 2016).

The logical structure of the neural networks used for that research is essentially that of a multi-layer perceptron (MLP), based on the propagation of local errors. Empty cells in raw data are treated, in such a logical structure, as null observations (i.e. zeros), and their presence in the source empirical data makes the neural network rely too much on its own estimations of the outcome variable. Thus, all the incomplete 'country-year' observations from the original Penn Tables 9.1 dataset, i.e. observations containing empty cells, have been removed. The resulting empirical set, further called **'No-Empty-Cells' set**, consists of ***n = 41*** variables and ***m = 3008*** 'country-year' observations. Meta-variables, pertaining to the quality of national statistics, have been removed. The detailed composition of this set, regarding countries and years, is provided in the Appendix, at the end of this article. Table 1, as well as Figures I and II, to be found in Appendix 1, too, give a general view of empirical correspondence between the original set of 12 000 observations, as published in Penn Tables 9.1, and the 'No-Empty-Cells' set. Whilst seeming pertinent as for the core issue of this article, i.e. the labour market, the 'No-Empty-Cells' set is representative for relatively big economies, both GDP-wise and population-wise, with a lower than average propensity to consume, and slightly faster inflation. It is also to keep in mind that timewise, this set is slightly sloping towards relatively recent years, and, for example, does not cover the period 1950 ÷ 1953, as reported in Penn Tables 9.1.

The method of simulating collective intelligence, used in this article, is reductionist, with weak assumptions, generally based on the 'swarm theory'. The neural network, besides feeding forward the strictly spoken error of estimation in the output variable, feeds a fitness function of the dataset (de Vincenzo et al. 2018). If a set of empirical data, pertinent to social phenomena, and treated by a neural network, displays visible changes in the value of its general fitness function, it can be assumed that the data in question represents collective intelligence. The so-far experience of social scientists with neural networks as predictive tools indicates that complexity of the network is not necessarily a straightforward correlate of its cognitive value (Olawoyin & Chen 2018). Apparently, keeping the network simple enough to be able to explain each of its functions is important for interpreting the results it yields.

The ***m = 3008*** observations in the 'No-Empty-Cells' set are assumed to be ***m = 3008*** phenomenal occurrences, which the neural network uses as learning material. Each "country – year" observation is a phenomenal occurrence ***o(j)***. The 'No-Empty-Cells' set is subject to standardization over local maxima in individual variables. Further, for the sake of presentational convenience, the 'No-Empty-Cells' is designated as source data, or ***X*** set, whilst the set of standardized values is called ***Z***. ***Z*** is structured into ***m*** phenomenal occurrences just as ***X*** is. Thus, $z_1(1)$ is the standardized value of variable x_1 in the phenomenal occurrence ***1*** etc. In the method such as it is presented here, it is assumed that the order of phenomenal occurrences is fixed. It corresponds to the assumption that a society, as collective intelligence, deals with an already given order of events to learn from.

The perceptron starts learning with the first phenomenal occurrence ***o(1)*** in the set ***Z***. Two operations are being carried out: neural activation and observation of the fitness function. $V[z_i(j)]$ stands for local value of the fitness function ***V*** in variable x_i (or ***y***), in the phenomenal occurrence ***j***; $V[z_i(j)]$ is calculated as the mean local Euclidean distance of $z_i(j)$ from other variables in the given ***j***-th phenomenal occurrence, as in equation (1) below.

$$V[z_i(j)] = \frac{\sum_{i=1}^{n}\sum_{k=1}^{n-1}\sqrt{[z_i(j)-z_k(j)]^2}}{n+1} \qquad (1)$$

$V[Z(j)]$ is the general fitness function of the set ***Z*** in the ***j***-th phenomenal occurrence, calculated as the mean value of $V[z_i(j)]$, as in equation (2).

$$V[Z(j)] = \frac{\sum_{i=1}^{n} V[z_i(j)]}{n+1} \qquad (2)$$

Note that fitness functions can include various detailed assumptions. The here presented one is probably the simplest measure of coherence between variables. Additional assumptions can cover, for example, arbitrary or non-arbitrary weighing of individual Euclidean distances etc.

Neural activation occurs by feeding data ***Z(1***) into the neural activation function ***g(1)***, used in the perceptron for estimating the value of output variable ***y*** in the phenomenal occurrence ***1***. In the original application of this method, the author used the function of hyperbolic tangent, or $g(j) = \frac{e^{2h}-1}{e^{2h}+1}$. Note that other activation functions (i.e. other than hyperbolic tangent) can be used; the key is the meaningfulness of results yielded by the perceptron. The variable ***h*** in ***Z(1)*** is calculated as $h = \sum_{i=1}^{n} z_i * w$, where ***w*** is a random weight ***0 > w > 1***.

For each consecutive phenomenal occurrence used to teach the perceptron, local error of estimation is computed $e_y(j)$, as in equation (3). The factor ***g'(j)*** is the first derivative of the activation function ***g(j)***. The idea behind adding it to the estimation of local error is that errors matter by their sheer magnitude as well as by the relative steepness of the neural activation function in the given phenomenal occurrence ***o(j)***.

$$e_y(j) = [g(j) - y] * g'(j) \quad (3)$$

The first round of learning with data, i.e. the perception, processing, estimation of coherence and that of accuracy yield two values: the vector of variable-specific local fitness functions ***V[Z(1)]***, and the local error ***e(1)***. The perceptron learns on its capacity to estimate the output variable 'y', and on the mutual coherence (Euclidean distance) of input variables. The underlying theoretical assumption is that collective intelligence attempts to achieve some outcomes and evaluates its own capacity to do it (that's why governments fall when they fail on key economic promises, for example), as well as its own coherence. The last assumption means that any culture learns and optimizes within a repertoire of moves coherent with the given set of social norms, e.g. changes in the healthcare system usually mean incremental change in public spending rather that brutal swing from 100% public funding to 100% private.

In the second round of learning, as well as in any consecutive one, thus when processing data ***Z(2) ≥ Z(j) ≥ Z(m)***, the logical structure changes slightly. The parameter h of the neural activation function

g(j) incorporates both the error generated in the previous round of learning, and the values of local fitness functions in the same preceding round, as in equation (4). Besides incorporating the lesson from previous rounds, the perceptron keeps experimenting with random weights $w(z_i)$, which, in turn, reflects the innovative component of collective intelligence.

$$h(2 \leq j \leq m) = \sum_{i=1}^{n}[z_i + e_y(j-1)] * w * V[z_i(j-1)] \qquad (4)$$

The procedure proposed in this method includes a formal check of intelligence, which, by the author's experience, is really a formality. The series of ***m*** general fitness functions ***V[Z(j)]***, as well as the series of ***m*** errors ***e(j)*** must both be non-monotonous. In other words, there must be demonstrable adjustment.

The set ***X*** of empirical observations, structured into ***m*** phenomenal occurrences and ***n*** variables, treated with the neural network, generates a transformed set ***S*** of values. Technically, occurrence 1 in the set ***S*** is identical to that in the set ***X***, and the remaining '***m – 1***' occurrences are different. With the same logical structure of the perceptron, the set ***X*** can generate as many different sets ***S***, as there are variables, thus ***n = 41***. In each mutation of the set ***X***, another variable from among ***n = 41*** is taken as desired output of the neural network. The set ***X*** can therefore generate ***n = 41*** alternative sets ***S***. Note that the exact empirical look of each set S_i is probabilistic: each time we run the perceptron over ***m*** phenomenal occurrences, it produces a slightly different set S_i of values.

Each set S_i can be compared as for its similarity to the original set ***X***. Many possible procedures of comparison are possible, and once more, this method goes for simplicity, and consistently with the logical structure of the perceptron, Euclidean distance has been chosen to gauge similarity.

In the set ***X***, each variable x_i yields a mean value $avg(X;x_i)$ over ***m*** phenomenal occurrences. In the same manner, each variable x_i in each set S_i is endowed with a mean value $avg(S_i;x_i)$. Both the mean values value $avg(X;x_i)$ and the value $avg(S_i;x_i)$ are vectors, characteristic for their respective sets. Each set S_i can be compared to the set ***X*** as for the Euclidean distance $E(X;S_i)$ between these vectors, as in equation (5). Among the ***n*** sets S_i generated from the set ***X***, the set S_i endowed with the smaller Euclidean distance $min[E(X; S_i)]$ is the most similar to ***X***. Consequently, we can assume that the output variable of this specific set S_i is the most likely value optimized by the society represented in the original empirical set ***X***. Other sets Si, endowed with higher ***E(X;S_i)*** are, respectively, less and less likely representations of values pursued by the society studied.

$$E(X;S_i) = \sqrt{\sum_{i=1}^{n}\left[\frac{avg(S_i;x_i)}{avg(X;x_i)} - 1\right]^2} \qquad (5)$$

It is arguable to what extent the neural network outlined in equations (1) – (4) is a recurrent neural network (RNN). On the one hand, phenomenal occurrences fed into it are ordered over time, as we have 'country-year' pooled observations in the dataset. That would fit the general definition of an RNN as a network learning from events in a time sequence (see for example: Alpay et al. 2019). However, the temporal order of phenomenal occurrences in the dataset is purposefully mixed, e.g. the last, 3008th occurrence in the set is Barbados in 1996, whilst Sweden in 2016 comes as the 1736th occurrence. According to the author's best understanding of the matter, the perceptron used is a non-recurrent one, as time is just an arbitrary dimension in this case.

Results and conclusions

Table 2, and figures III ÷ VII, in Appendix 1, document the results of simulation run with the above-described method. Table 2 presents the normalized Euclidean distances ***E(X;S_i)*** of the ***n = 41*** perceptrons from the original set ***X***. Figure III introduces an intuitive visualisation of those distances, after inversion, thus '***1/ E(X;S_i)'***, so as to show the hierarchy of relative importance among the ***n = 41*** variables apprehended as alternative values for optimization in the experiment. Besides this general presentation of results, the reader can consult detailed results in two Excel workbooks, made available in the archives of the author's blog. The workbook entitled 'Comparison of Perceptrons'[1] presents detailed calculations of Euclidean distances as in equation (5), which can be interesting to the extent that some among the 41 sets S_i display apparently absurd mean values, e.g. negative aggregate real output. The author attempts to give as exhaustive an interpretation of those results, yet the reader is

[1] https://discoversocialsciences.com/wp-content/uploads/2019/10/PWT-9_1-comparison-of-perceptrons_for-upload.xlsx

welcome to study them directly. On the other hand, the workbook entitled 'PWT 9_1 Perceptron pegged AVH'[2] introduces the database used for this empirical research, as well as the logical structure of the neural network, according to equations (1) – (4), in the form of an Excel spreadsheet, in the specific version oriented on optimizing the number of hours worked (AVH) as output variable. Appendix 2, further after Appendix 1, presents in detail the logical structure of the neural network in the Excel form, using the same version, optimizing the variable AVH, as working example.

Probably the most striking observation about the results obtained is that only a few variables in the dataset yield meaningful results (i.e. results with non-negative mean aggregates), as collective outcomes (output variables) of the neural network formalized in equations (1) – (4). These variables are: RGDPE, AVH, HC, LABSH, CSH_C, PL_X, PL_M. All the other variables, used as output ones, yield absurd results with negative mean aggregates. The collective intelligence represented in equations (1) – (4) is able to optimize itself in elementary accord with real-life data only in those few cases. Of course, much can be studied more in depth about the logical structure of the neural network itself, still the basic observation remains: the neural network used here is a standardized experimental environment and this environment yields different results according to the variable optimized as the desired outcome. Among those different results some are impossible: there is no possible social reality with negative economic aggregates. According to the methodology used the here-presented research, it is to conclude that these particular outcomes, i.e. 34 from among 41 variables tested, cannot possibly be collectively desired outcomes for the society represented in the source dataset X. This observation strongly corroborates conclusions the application of Extended Kalman Filter (Munguia et al. 2019, op. cit.), namely that macroeconomic variables vary substantially as for their predictive value. Some of them have much more meaning than others.

Interestingly, when oriented on optimizing the above-mentioned 7 variables, the neural network produces lower mean aggregates than those in the original dataset ***X***, the one labelled earlier as 'No-Empty-Cells' one, selected out of Penn Tables 9.1. It is as if the collective intelligence represented by the perceptron used was steering the data towards a greater similarity with the full information available

[2] https://discoversocialsciences.com/wp-content/uploads/2019/10/PWT-9_1-Perceptron-pegged-AVH.xlsx

in PWT 9.1. Among the 7 variables that yield plausibly acceptable sets S_i, LABSH and AVH clearly outrun the five remaining ones. The working hypothesis of this article, namely that **economic systems optimize themselves, as collective intelligence, so as to adapt real output and price levels to the average workload practiced in the given national economy**, seems to be substantiated by these results.

In the method used, the process of learning observable in collective intelligence is just as interesting to follow, as the final results it yields. Two metrics of the neural network, namely the error of estimation ***e*** as in equation (3), and the overall fitness function ***V[Z(j)]*** as in equation (2), inform about the essential dynamics of learning. Figures IV ÷ VII document the unfolding of two key operational values of the perceptrons created, thus the local error and the overall fitness function ***V[Z(j)]***, for two specific cases: the best-fitting-to-reality perceptron optimizing the share of labour in the national income (LABSH), and one of the worst fitting perceptrons, namely that oriented on optimizing the price level in investment (PL_I). Those figures serve as formal proof that the neural network used is, indeed, an intelligent structure, as it is able to adjust itself to local error, and to shift between different levels of internal coherence. The fitness function ***V[Z(j)]*** unfolds along very similar paths in both cases, and its observable variation is, by the way, a formal proof of collective intelligence in the neural network used. On the other hand, the graph of error is different in these two versions of the perceptron. When optimizing LABSH, the neural network recurrently shifts between widely differing values of the local error. It is a non-monotonous process of learning, oscillating between quick accumulation of new experience (wide margin of error), and utilisation thereof (shrinking magnitude of error). When prices PL_I are being optimized as output variable, the error seems pretty monotonous, as if the intelligent structure at work was constantly missing the target values by more or less the same magnitude, i.e. as if it was not learning anything.

Affiliation*: The Andrzej-Frycz Modrzewski Krakow University, Faculty of Management, Krakow, Poland*

References

1. Alpay, T., Abawi, F., & Wermter, S. (2019). Preserving activations in recurrent neural networks based on surprisal. Neurocomputing, 342, 75-82
2. Andreoni, V. (2017). Energy Metabolism of 28 World Countries: A Multi-scale Integrated Analysis. Ecological Economics, 142, 56-69
3. Babatunde, K. A., Begum, R. A., & Said, F. F. (2017). Application of computable general equilibrium (CGE) to climate change mitigation policy: a systematic review. Renewable and Sustainable Energy Reviews, 78, 61-71.
4. Bick, A., Fuchs-Schündeln, N., & Lagakos, D. (2018). How do hours worked vary with income? Cross-country evidence and implications. American Economic Review, 108(1), 170-99.
5. Bignetti, E. (2014). The functional role of free-will illusion in cognition:"The Bignetti Model". Cognitive Systems Research, 31, 45-60.
6. Bignetti, E. (2018). New Insights into "The Bignetti Model" from Classic and Quantum Mechanics Perspectives. Perspective, 4(1), 24.
7. Bignetti, E., Martuzzi, F., & Tartabini, A. (2017). A Psychophysical Approach to Test:"The Bignetti Model". Psychol Cogn Sci Open J, 3(1), 24-35.
8. Cantore, Cristiano and Ferroni, Filippo and Leon-Ledesma, Miguel A. (2017) The Dynamics of Hours Worked and Technology. Journal of Economic Dynamics and Control, 82 . pp. 67-82. ISSN 0165-1889, DOI: https://doi.org/10.1016/j.jedc.2017.05.009
9. David, P. A. (1990). The dynamo and the computer: an historical perspective on the modern productivity paradox. The American Economic Review, 80(2), 355-361.
10. De Vincenzo, I., Massari, G. F., Giannoccaro, I., Carbone, G., & Grigolini, P. (2018). Mimicking the collective intelligence of human groups as an optimization tool for complex problems. Chaos, Solitons & Fractals, 110, 259-266
11. dos Santos, M. M. (2017). Holism, collective intelligence, climate change and sustainable cities. Procedia Computer Science, 109, 763-770
12. Doya, K., & Taniguchi, T. (2019). Toward evolutionary and developmental intelligence. Current Opinion in Behavioral Sciences, 29, 91-96.
13. Fang, Z. (2016). Data on examining the role of human capital in the energy-growth nexus across countries. Data in brief, 9, 540-542
14. Feenstra, Robert C., Robert Inklaar and Marcel P. Timmer (2015), "The Next Generation of the Penn World Table" American Economic Review, 105(10), 3150-3182, available for download at www.ggdc.net/pwt
15. Fuchs-Schündeln, N., Bick, A., & Brüggemann, B. (2017). Hours Worked in Europe and the US: New Data, New Answers. In Annual Conference 2017 (Vienna): Alternative Structures for Money and Banking (No. 168232). Verein für Socialpolitik/German Economic Association, https://www.econstor.eu/bitstream/10419/147865/1/dp10179.pdf
16. Hafstead, M. A., & Williams III, R. C. (2018). Unemployment and environmental regulation in general equilibrium. Journal of Public Economics, 160, 50-65
17. https://discoversocialsciences.com/wp-content/uploads/2019/10/PWT-9_1-comparison-of-perceptrons_for-upload.xlsx
18. https://discoversocialsciences.com/wp-content/uploads/2019/10/PWT-9_1-Perceptron-pegged-AVH.xlsx
19. Kline, R., Pinch, T., 1996, Users as Agents of Technological Change : The Social Construction of the Automobile in the Rural United States, Technology and Culture, vol. 37, no. 4 (Oct. 1996), pp. 763 - 795
20. Koo, J. L., Koo, B. L., & Shin, Y. H. (2013). An optimal investment, consumption, leisure, and voluntary retirement problem with Cobb–Douglas utility: dynamic programming approaches. Applied Mathematics Letters, 26(4), 481-486.
21. Lawniczak, A. T., & Di Stefano, B. N. (2010). Computational intelligence based architecture for cognitive agents. Procedia Computer Science, 1(1), 2227-2235.
22. Lee, R. S. (2019). COSMOS trader–Chaotic Neuro-oscillatory multiagent financial prediction and trading system. The Journal of Finance and Data Science, 5(2), 61-82.

23. MacKenzie, D., 1984, Marx and the Machine, Technology and Culture, Vol. 25, No. 3. (Jul., 1984), pp. 473-502.
24. Mumford, L., 1964, Authoritarian and Democratic Technics, Technology and Culture, Vol. 5, No. 1 (Winter, 1964), pp. 1-8, Published by: The Johns Hopkins University Press on behalf of the Society for the History of Technology
25. Munguía, R., Davalos, J., & Urzua, S. (2019). Estimation of the Solow-Cobb-Douglas economic growth model with a Kalman filter: An observability-based approach. Heliyon, 5(6), e01959.
26. Munir, K., & Anjum, M. S. (2018). The use of ontologies for effective knowledge modelling and information retrieval. Applied Computing and Informatics, 14(2), 116-126.
27. Olawoyin, A., & Chen, Y. (2018). Predicting the Future with Artificial Neural Network. Procedia Computer Science, 140, 383-392
28. Selten, R. (1990). Bounded rationality. Journal of Institutional and Theoretical Economics (JITE)/Zeitschrift für die gesamte Staatswissenschaft, 146(4), 649-658
29. Sinha, P., Gaughan, A. E., Stevens, F. R., Nieves, J. J., Sorichetta, A., & Tatem, A. J. (2019). Assessing the spatial sensitivity of a random forest model: Application in gridded population modeling. Computers, Environment and Urban Systems, 75, 132-145.
30. Sloman, A. (1993). Prospects for AI as the general science of intelligence. Prospects for artificial intelligence, 1-10.
31. Stradner, J., Thenius, R., Zahadat, P., Hamann, H., Crailsheim, K., & Schmickl, T. (2013). Algorithmic requirements for swarm intelligence in differently coupled collective systems. Chaos, Solitons & Fractals, 50, 100-114
32. Velasco-Fernández, R., Giampietro, M., & Bukkens, S. G. (2018). Analyzing the energy performance of manufacturing across levels using the end-use matrix. Energy, 161, 559-572
33. Vincenti, W.G., 1994, The Retractable Airplane Landing Gear and the Northrop "Anomaly": Variation-Selection and the Shaping of Technology, Technology and Culture, Vol. 35, No. 1 (Jan., 1994), pp. 1-33
34. World Bank. 2019. World Development Report 2019: The Changing Nature of Work. Washington, DC: World Bank. doi:10.1596/978-1-4648-1328-3. License: Creative Commons Attribution CC BY 3.0 IGO
35. Zhang, M. H., & Cheng, Q. S. (2003). Gaussian mixture modelling to detect random walks in capital markets. Mathematical and computer modelling, 38(5-6), 503-508.

Appendix 1 – Dataset and results

Geographical scope of the 'No-Empty-Cells' database selected from PWT 9.1 for being processed with the neural network (hyphenated values are the respective numbers of annual observations for each given country): Argentina – 64; Australia – 64; Austria – 64; Barbados – 11; Belgium – 64; Brazil – 64; Bulgaria – 23; Canada – 64; Chile – 63; China – 48; China, Hong Kong SAR – 54; Colombia – 64; Costa Rica – 31; Croatia – 23; Cyprus - 23; Czech Republic – 24; Denmark – 63; Ecuador – 23; Estonia – 23; Finland – 64; France – 64; Germany – 64; Greece – 63; Hungary – 38; Iceland – 54; India – 48; Indonesia – 48; Ireland – 64; Israel – 37; Italy – 64; Jamaica- 17; Japan – 64; Latvia – 23; Lithuania – 23; Luxembourg – 48; Malaysia – 48; Malta – 24; Mexico – 64; Netherlands – 64; New Zealand – 48; Nigeria – 8; Norway – 63; Peru – 64; Philippines – 48; Poland – 25; Portugal – 64; Republic of Korea - 61; Romania – 23; Russian Federation – 24; Singapore – 54; Slovakia – 24; Slovenia – 23; South Africa – 17; Spain 64; Sri Lanka – 48; Sweden – 64; Switzerland – 64; Taiwan – 63; Thailand – 48; Trinidad and Tobago – 12; Turkey – 48; United Kingdom – 64; United States - 64; Uruguay – 28; Venezuela (Bolivarian Republic of) - 53

Temporal scope of the 'No-Empty-Cells' database selected from PWT 9.1 for being processed with the neural network (hyphenated values are the respective numbers of national observations for each given year): 1954 – 25; 1955 – 28; 1956 – 28; 1957 ÷ 1963 – 29 each; 1964 ÷ 1969 – 32 each; 1970 ÷ 1979 – 42 each; 1980 – 43; 1981 ÷ 1985 – 44 each; 1986 – 45; 1987 ÷ 1989 - 46 each; 1990- 47; 1991 ÷ 1992 – 49 each; 1993 – 50; 1994 – 54; 1995 ÷ 2000 – 63 each; 2001 – 64; 2002- 63; 2003 ÷ 2005 – 61 each; 2006 ÷ 2009 and 2013 – 60 each; 2010 ÷ 2012 and 2014 ÷ 2017 – 61 each;

Table 1 Comparison of mean values in variables studied, full Penn Tables 9.1 database vs. the 'No-Empty-Cells' set selected for being processed with the neural network

Acronym for the variable	**Desciption of the variable**	**Mean value in the original PWT 9_1 set**	**Mean value in the 'No empty cells set'**	**Normalized Euclidean distance**
rgdpe	Expenditure-side real GDP at chained PPPs (in mil. 2011US$)	272 056,94	778 348,68	1,86
rgdpo	Output-side real GDP at chained PPPs (in mil. 2011US$)	269 192,84	766 791,64	1,85
pop	Population (in millions)	30,74	68,16	1,22
emp	Number of persons engaged (in millions)	14,80	30,97	1,09
avh	Average annual hours worked by persons engaged	1 984,10	1 952,22	0,02
hc	Human capital index, based on years of schooling and returns to education; see Human capital in PWT9.	2,06	2,63	0,28
ccon	Real consumption of households and government, at current PPPs (in mil. 2011US$)	198 499,76	564 898,51	1,85

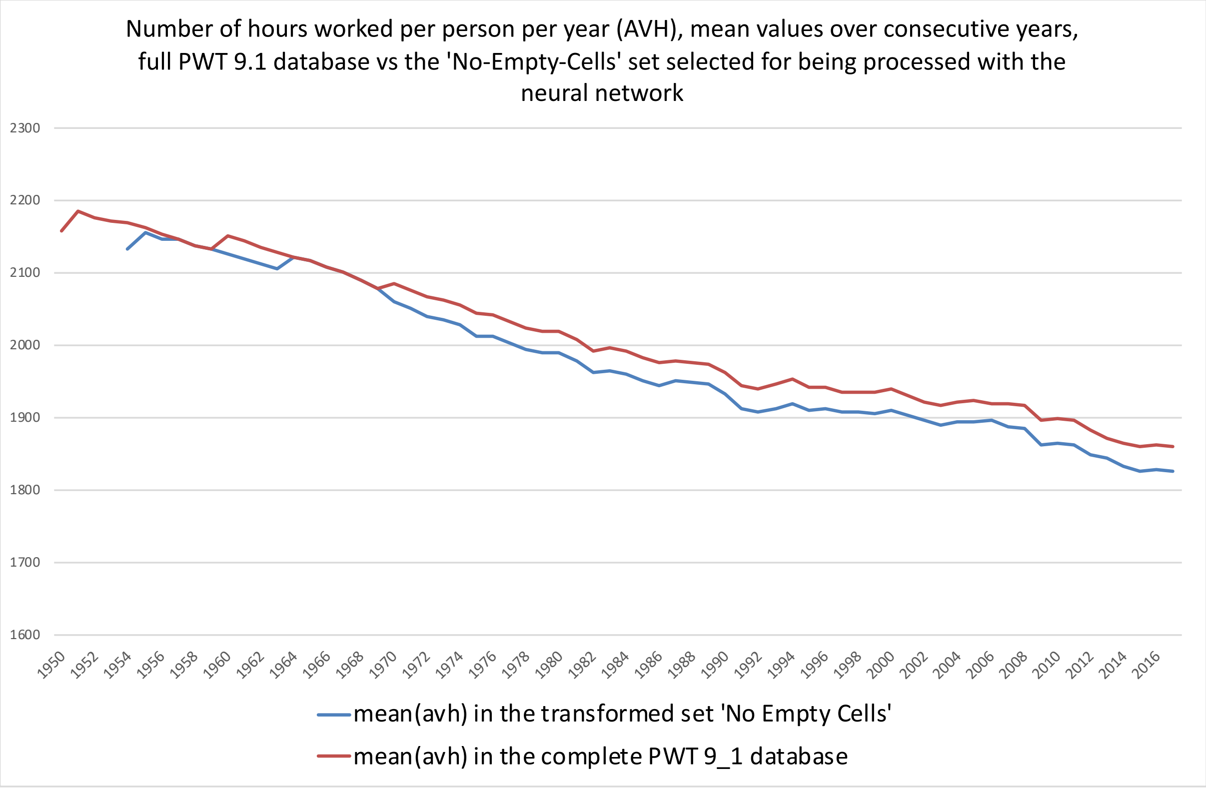
cda	Real domestic absorption, (real consumption plus investment), at current PPPs (in mil. 2011US$)	268 658,05	770 815,26	1,87
cgdpe	Expenditure-side real GDP at current PPPs (in mil. 2011US$)	269 708,84	772 380,20	1,86
cgdpo	Output-side real GDP at current PPPs (in mil. 2011US$)	269 769,30	769 436,58	1,85
cn	Capital stock at current PPPs (in mil. 2011US$)	908 755,60	2 668 041,35	1,94
ck	Capital services levels at current PPPs (USA=1)	0,03	0,07	1,14
ctfp	TFP level at current PPPs (USA=1)	0,71	0,75	0,06
cwtfp	Welfare-relevant TFP levels at current PPPs (USA=1)	0,70	0,74	0,06
rgdpna	Real GDP at constant 2011 national prices (in mil. 2011US$)	297 194,26	830 168,48	1,79
rconna	Real consumption at constant 2011 national prices (in mil. 2011US$)	215 900,89	610 151,00	1,83
rdana	Real domestic absorption at constant 2011 national prices (in mil. 2011US$)	289 386,29	824 465,56	1,85

rnna	Capital stock at constant 2011 national prices (in mil. 2011US$)	1 128 532,98	3 262 012,07	1,89
rkna	Capital services at constant 2011 national prices (2011=1)	0,53	0,55	0,04
rtfpna	TFP at constant national prices (2011=1)	0,99	0,92	0,07
rwtfpna	Welfare-relevant TFP at constant national prices (2011=1)	0,97	0,93	0,04
labsh	Share of labour compensation in GDP at current national prices	0,53	0,56	0,04
irr	Real internal rate of return	0,14	0,11	0,20
delta	Average depreciation rate of the capital stock	0,04	0,04	0,09
xr	Exchange rate, national currency/USD (market+estimated)	244,63	132,26	0,46
pl_con	Price level of CCON (PPP/XR), price level of USA GDPo in 2011=1	0,38	0,51	0,36
pl_da	Price level of CDA (PPP/XR), price level of USA GDPo in 2011=1	0,38	0,50	0,33
pl_gdpo	Price level of CGDPo (PPP/XR), price level of USA GDPo in 2011=1	0,40	0,51	0,30

csh_c	Share of household consumption at current PPPs	0,64	0,59	0,08
csh_i	Share of gross capital formation at current PPPs	0,22	0,26	0,16
csh_g	Share of government consumption at current PPPs	0,19	0,17	0,15
csh_x	Share of merchandise exports at current PPPs	0,23	0,30	0,33
csh_m	Share of merchandise imports at current PPPs	(0,31)	(0,34)	0,10
csh_r	Share of residual trade and GDP statistical discrepancy at current PPPs	0,02	0,02	0,04
pl_c	Price level of household consumption, price level of USA GDPo in 2011=1	0,39	0,51	0,30
pl_i	Price level of capital formation, price level of USA GDPo in 2011=1	0,49	0,49	0,01
pl_g	Price level of government consumption, price level of USA GDPo in 2011=1	0,37	0,55	0,49
pl_x	Price level of exports, price level of USA GDPo in 2011=1	0,44	0,47	0,07

pl_m	Price level of imports, price level of USA GDPo in 2011=1	0,43	0,45	0,05
pl_n	Price level of the capital stock, price level of USA in 2011=1	0,47	0,50	0,06
pl_k	Price level of the capital services, price level of USA=1	1,40	1,06	0,25

Figure I



Number of hours worked per person per year (AVH), mean values over consecutive years, full PWT 9.1 database vs the 'No-Empty-Cells' set selected for being processed with the neural network
2300
2200
2100
2000
1900
1800
1700
1600
1950
1952
1954
1956
1958
1960
1962
1964
1966
1968
1970
1972
1974
1976
1978
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
2004
2006
2008
2010
2012
2014
2016
mean(avh) in the transformed set 'No Empty Cells'
mean(avh) in the complete PWT 9_1 database

Figure II

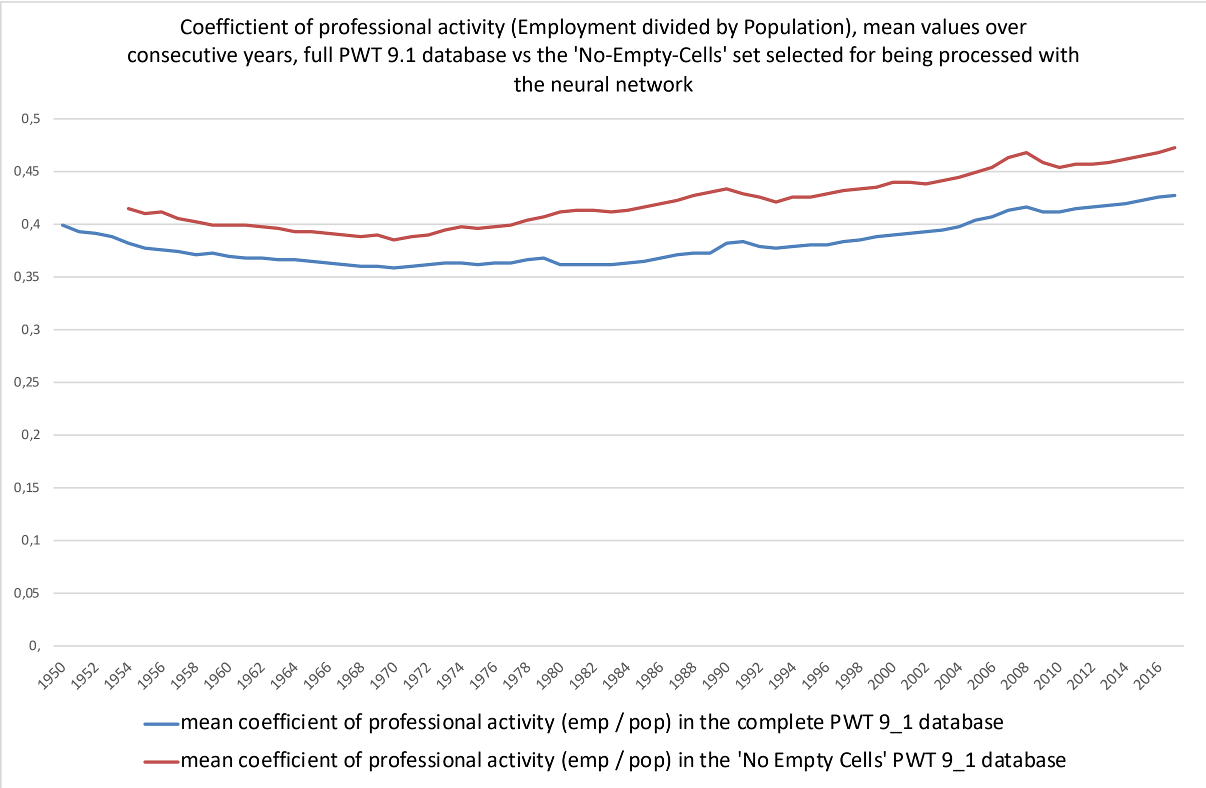

Coeffictient of professional activity (Employment divided by Population), mean values over consecutive years, full PWT 9.1 database vs the 'No-Empty-Cells' set selected for being processed with the neural network
0,5
0,45
0,4
0,35
0,3
0,25
0,2
0,15
0,1
0,05
0,
1950
1952
1954
1956
1958
1960
1962
1964
1966
1968
1970
1972
1974
1976
1978
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
2004
2006
2008
2010
2012
2014
2016
mean coefficient of professional activity (emp / pop) in the complete PWT 9_1 database
mean coefficient of professional activity (emp / pop) in the 'No Empty Cells' PWT 9_1 database

Table 2

Output variable	Euclidean distance E(X; Si)	Output variable	Euclidean distance E(X; Si)
labsh	0,10289643	**xr**	1,16419328
avh	0,13213112	**irr**	1,18830309
pl_x	0,25313668	**ctfp**	1,20714548
hc	0,28314643	**csh_r**	1,27066942
csh_c	0,30163182	**csh_m**	1,33272667
pl_m	0,40636264	**ccon**	1,34108299
rgdpe	0,53029917	**emp**	1,34595378
delta	0,70944368	**ck**	1,34647368
rwtfpna	0,81232978	**csh_x**	1,35027583
rkna	0,85973815	**rnna**	1,35078629
pl_c	0,9952317	**pop**	1,35133374
pl_con	0,99815527	**cn**	1,35752019
pl_g	1,01978919	**rdana**	1,36035624
pl_da	1,02886881	**rconna**	1,36354026
csh_i	1,04481571	**cgdpo**	1,37842923
rtfpna	1,06257424	**cda**	1,37887019
csh_g	1,07277442	**cgdpe**	1,38814459
pl_n	1,14136608	**pl_i**	1,4040277
cwtfp	1,14762207	**rgdpna**	1,40404285

pl_gdpo	1,14945035	**rgdpo**	1,41830252
pl_gdpo	1,14945035	**pl_k**	1,42108981

Figure III

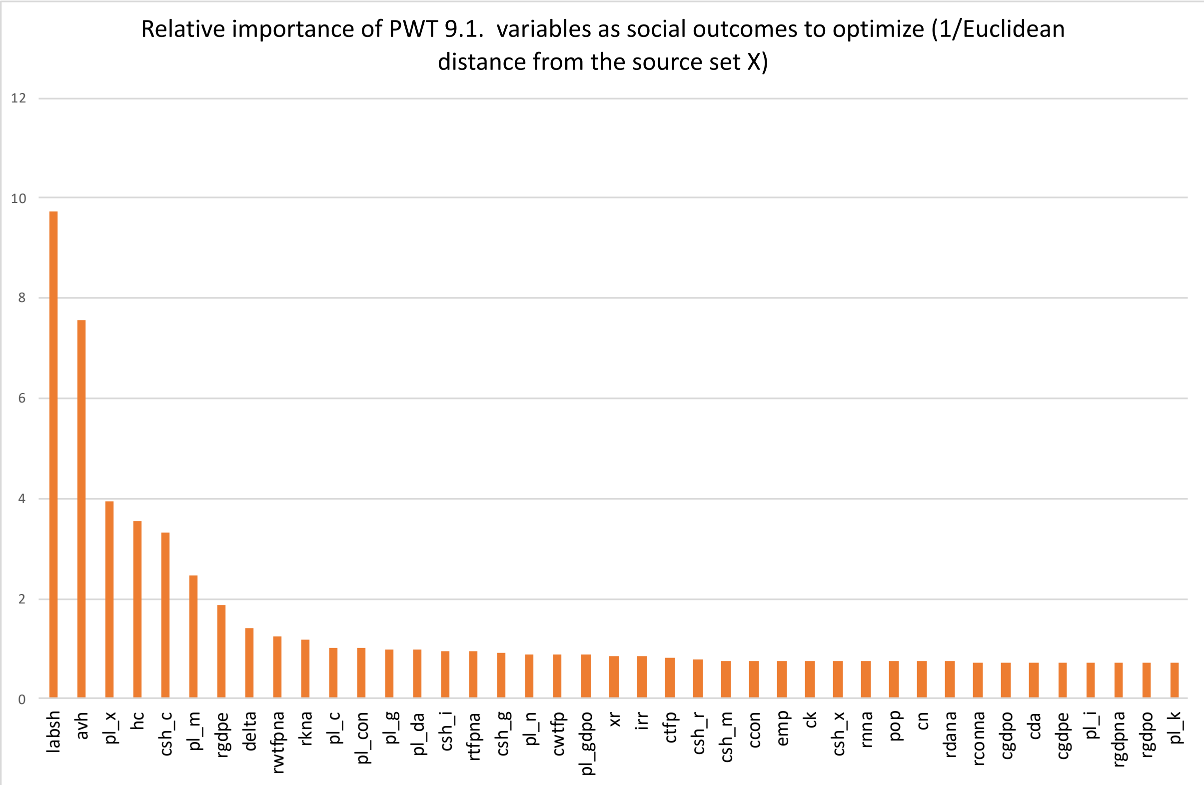

Relative importance of PWT 9.1. variables as social outcomes to optimize (1/Euclidean distance from the source set X)
12
10
8
6
4
2
0
labsh
avh
pl_x
hc
csh_c
pl_m
rgdpe
delta
rwtfpna
rkna
pl_c
pl_con
pl_g
pl_da
csh_i
rtfpna
csh_g
pl_n
cwtfp
pl_gdpo
xr
irr
ctfp
csh_r
csh_m
ccon
emp
ck
csh_x
rnna
pop
cn
rdana
rconna
cgdpo
cda
cgdpe
pl_i
rgdpna
rgdpo
pl_k

Figure IV

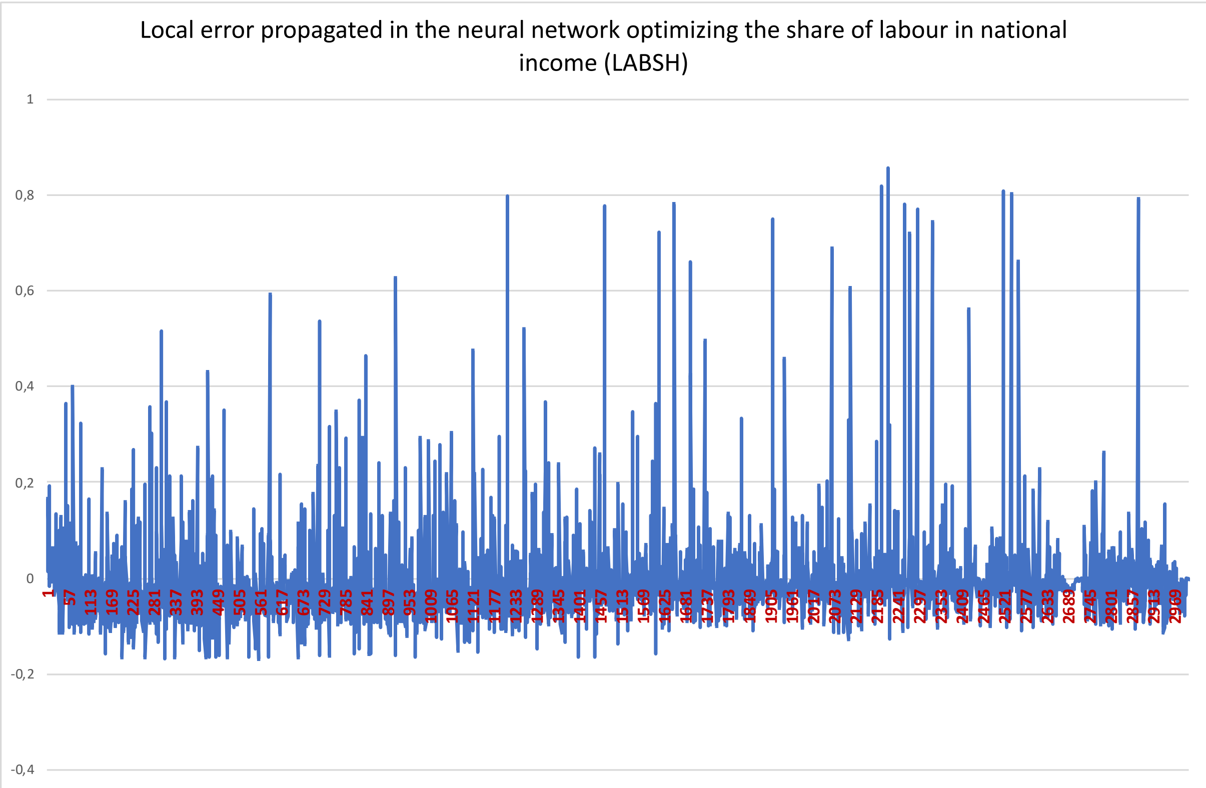

Local error propagated in the neural network optimizing the share of labour in national income (LABSH)
1
0,8
0,6
0,4
0,2
0
-0,2
-0,4

Figure V

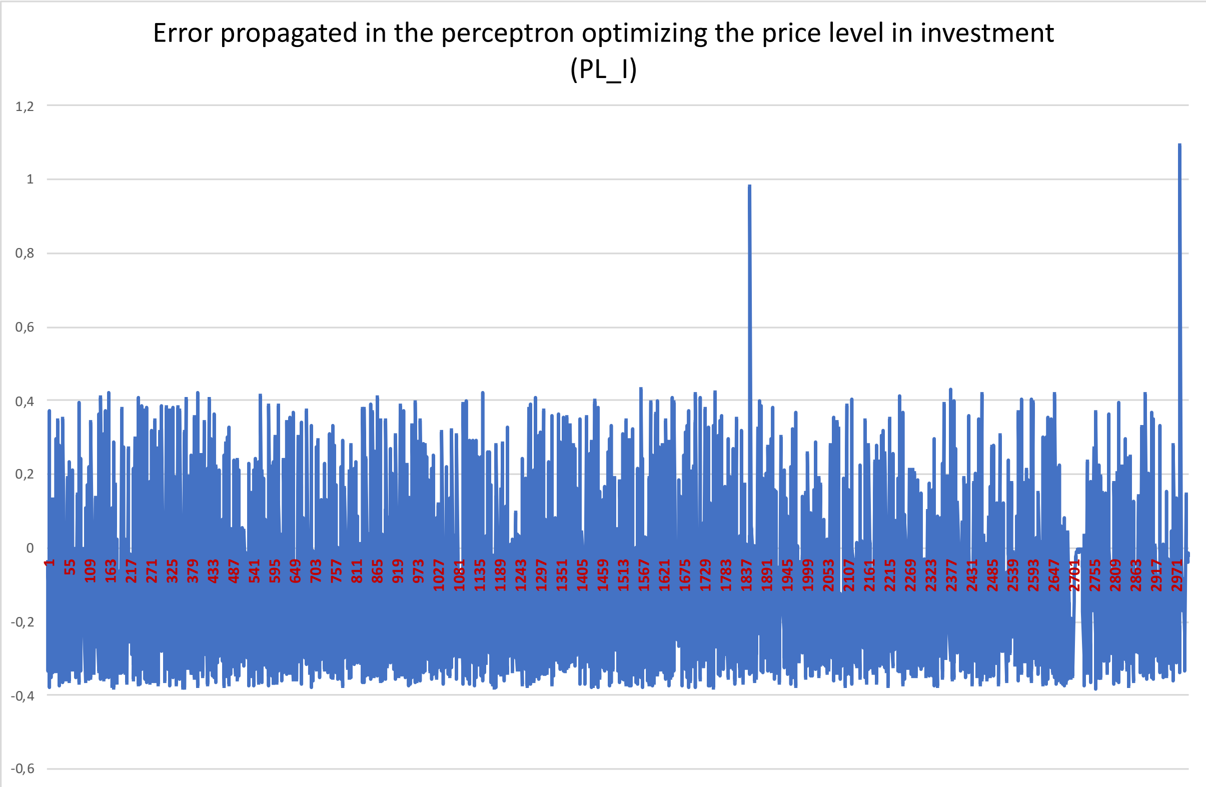

Error propagated in the perceptron optimizing the price level in investment (PL_I)
1,2
1
0,8
0,6
0,4
0,2
0
-0,2
-0,4
-0,6

Figure VI

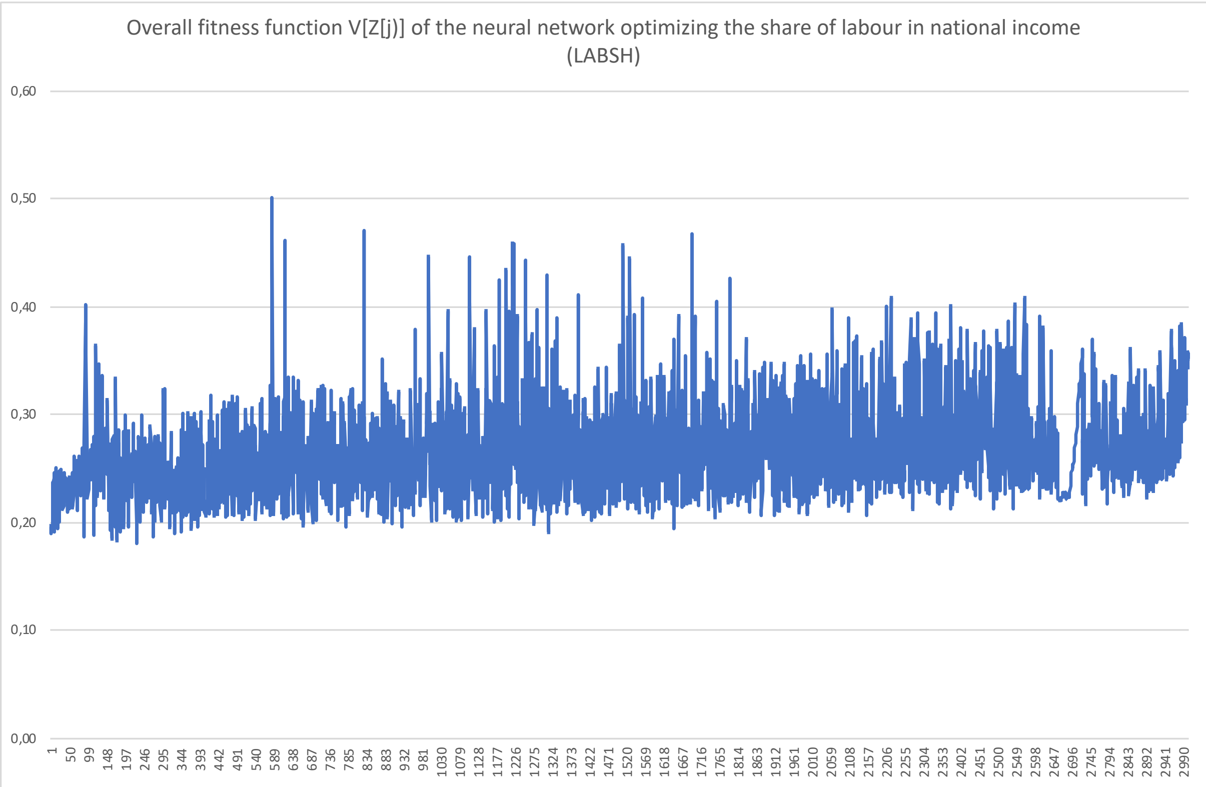
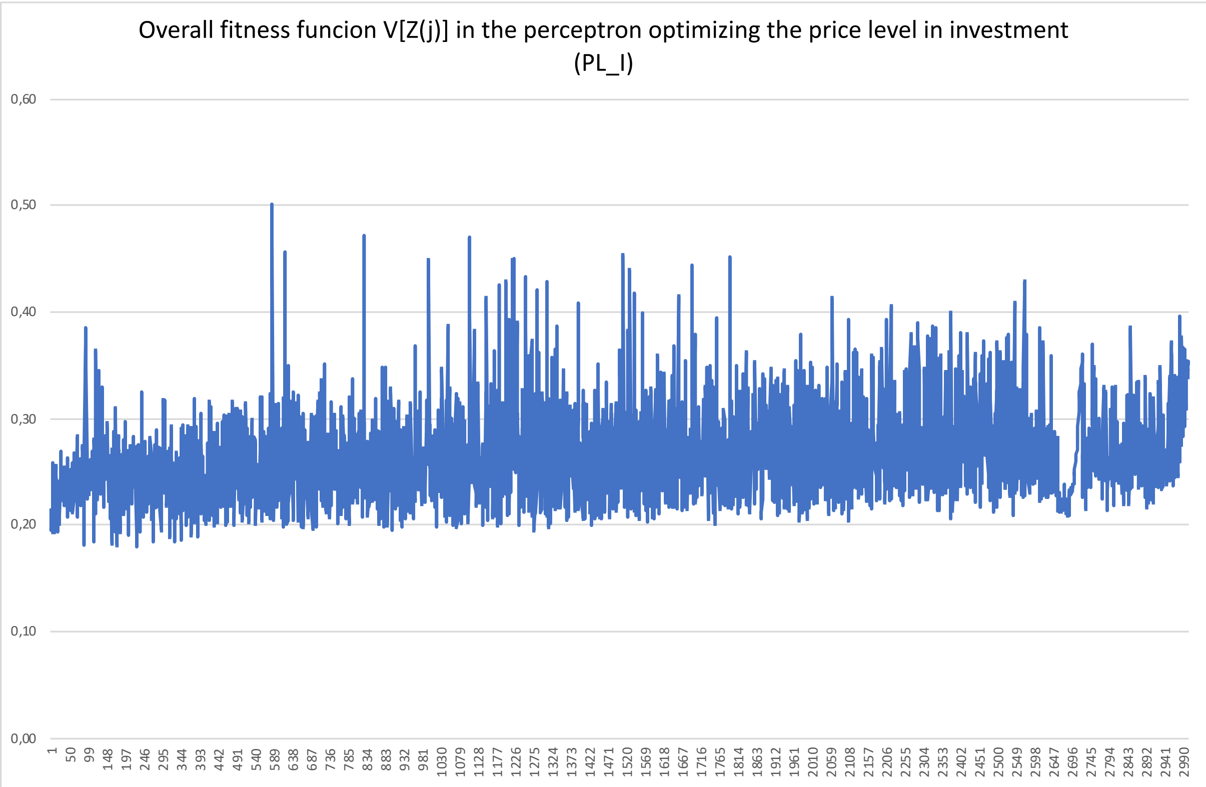

Overall fitness function V[Z[j)] of the neural network optimizing the share of labour in national income (LABSH)
0,60
0,50
0,40
0,30
0,20
0,10
0,00
1
50
99
148
197
246
295
344
393
442
491
540
589
638
687
736
785
834
883
932
981
1030
1079
1128
1177
1226
1275
1324
1373
1422
1471
1520
1569
1618
1667
1716
1765
1814
1863
1912
1961
2010
2059
2108
2157
2206
2255
2304
2353
2402
2451
2500
2549
2598
2647
2696
2745
2794
2843
2892
2941
2990

Figure VII



Overall fitness funcion V[Z(j)] in the perceptron optimizing the price level in investment (PL_I)
0,60
0,50
0,40
0,30
0,20
0,10
0,00
1
50
99
148
197
246
295
344
393
442
491
540
589
638
687
736
785
834
883
932
981
1030
1079
1128
1177
1226
1275
1324
1373
1422
1471
1520
1569
1618
1667
1716
1765
1814
1863
1912
1961
2010
2059
2108
2157
2206
2255
2304
2353
2402
2451
2500
2549
2598
2647
2696
2745
2794
2843
2892
2941
2990

Appendix 2 – Methodological note: The neural network in the form of an Excel spreadsheet

This note describes the exact structure of the Excel workbook that contain both the source data used in the empirical research, and the multi-layer perceptron used to treat that data. The interest of using Excel is strictly scientific: whilst being a lot slower than algorithm written in languages like Python, especially with large datasets, an Excel spreadsheet allows observing hands-on, one round after another, the learning process of the neural network. Besides, the spreadsheet, such as discussed here, allows graphical insight into the learning process, due to the flat, two-dimensional structure of the tables, as well as via self-updating graphs.

The Excel spreadsheet adjacent to this methodological paper provide empirical data used in the manuscript "*The labour-oriented, collective intelligence of ours: Penn Tables 9.1 seen through the eyes of a neural network*". According to the above-presented methodology, 41 Excel files have been created, to emulate the 41 sets S_i, oriented on optimizing, respectively, the 41 variables taken into consideration. This methodological note describes the logical structure of the neural network pegged on optimizing the variable '**avh**', or the average annual hours worked by persons engaged, as this specific perceptron achieved one of the greatest Euclidean similarity to the source X set. The corresponding Excel workbook is made available to the reader. Once the reader grasps the general logic,

The general structure of the spreadsheet covers 4 tables and 2 graphs. The tables are: **Source Data, Standardized, Perceptron, Notes**. Graphs show the distribution of local error ***e(j)*** over the given series of phenomenal occurrences, and the distribution of general fitness function ***V[Z(j)]***. The '**Source Data**' table contains the set ***X*** of source empirical data. In this empirical case, the set X contains 41 variables. As the digital vehicle is Excel, empty cells in the '**Source Data**' table are to avoid. Thus, the set ***X*** presented in the Excel file adjacent to this paper is filtered for records containing empty observations. In the very bottom of the 'Source Data' table the reader can find two lines, marked:

Maximum ($3010), and Average ($3011). These are useful, in the first place, for the standardization of source data, and for further assessment of Euclidean distance between the source set ***X***, and the derived sets S_i.

The table labelled '***Standardized***' contains the set ***Z*** of data standardized for being processed with the neural network. The method of standardization used in the Excel file linked to in the 'Resource availability' section standardizes the source data in the simplest possible way, i.e. linearly over local maxima. The cell C2 in the 'Standardized' table contains the formula: **='Source data'!C2/'Source data'!C$3010**, where the address '**Source data'!C2** corresponds to the source value $x_i(1)$ – variable 'RGDPE', phenomenal occurrence 1, which is Republic of Korea in 1957, in the 'Source data' table, and **'Source data'!C$3010** is the local maximum of the same variable. All the other cells in the 'Standardized' table reproduce the same logic.

The '**Perceptron**' table contains the logical structure of the neural network strictly spoken. Note that lines in this table are moved one step down, as compared to lines in tables 'Source Data', and 'Standardized'. Line 4 in 'Perceptron' is functionally connected to line 3 in 'Source Data' and 'Standardized' and so forth. **The first two columns, A and B**, reproduce the labelling of phenomenal occurrences ***o(j)*** from the 'Source Data' table. In this specific Excel file, they name the 'country – year' observations for research on energy efficiency. This labelling of lines can be important as far as the user is interested in the exact order of phenomenal occurrences, as well as in observing the behaviour of the neural network in a precise point $x_i[o(j)]$.

The logical structure of the neural network is reflected in the functional assignment of columns. In other words, the table 'Perceptron' is divided into sections of columns, each bearing a different label, and each corresponding to a different neuron (or layer of neurons) in the perceptron. **Columns C ÷ AQ of the 'Perceptron'** table are labelled '**Initial values**'. Column **G** contains the output variable '**AVH'**, whilst columns **C ÷ F** and **H ÷ AQ** cover the input variables. Consequently, the cell G3 contains the formula: =Standardized!G2, cell G4 refers "=Standardized!G3" etc. The first line in column D spells "=Standardized!D3", but in the next lines, this precise input variable – RGDPO – is being fed with error ***e(j-1)***, from the preceding round of learning, i.e. the preceding phenomenal occurrence ***o(j-1)***. Error is fed from **column CO**, whose exact logical connection is described further in this section. Cell D4 refers

"=Standardized!D3+$CO3", cell D5 is "=Standardized!D4+$CO4" etc. The remaining columns of the {**C ÷ F; H ÷ A**} section follow the same logic: in lines 4 ÷ 3008 they sum standardized values z_i from the 'Standardized' table with the error ***e(j-1)*** from the column AI.

The first line of these values (line #3 in this precise Excel file) contains the standardized values in the first phenomenal occurrence ***o(1)*** in the set ***Z***. Cells D3 ÷ N3 (column-wise), reproduce the same connection to the 'Standardized' table. Further lines in columns C ÷ N are split in two categories: standardized output variable in column C, and standardized values of input variables, fed with error ***e(j)***, in columns ***D*** ÷ ***N***.

The next group of columns in the '**Perceptron**' table, which the reader should get interested in, are **columns CP ÷ ED**. Whilst being pushed to the very right end of the table, they become important quite early, as they contain local fitness functions ***V[z_i(j)],*** conformingly to equation (1). As the reader will study them, please bear in mind that the final denominator, '/41' in this case, depends on the number of variables in the perceptron. **Column EE** produces the general fitness function ***V[Z(j)]*** as in equation (2), simply by drawing an average from columns **CP ÷ ED**, e.g. **cell EE3**: = AVERAGE(CP3:ED3), cell EE4 = AVERAGE(CP4 :ED4) etc.

Now, as the reader has got at least somehow familiar with the way of handling the fitness function, or equations (1) and (2), in the '**Perceptron**' table of the spreadsheet, it is time to move back to the left side, and to see the way of generating the neural activation function. The first step is to generate the vector '***h***', the exponent of activation function, which, as the reader probably remembers, is generated differently in the first phenomenal occurrence on the one hand, from those generated in further ones. Vector ***h*** is computed in **columns AR ÷ CG** of the table. The general philosophy of structuring this table is to make the process rather piecemeal, in order to be able to follow small local changes. Hence, **columns AR ÷ CF**, grouped in the section labelled '*Values weighed with random and V'*, contain local components of the vector h, corresponding to individual variables. Column CG of the table sums up all the local components of the vector *h*, and here comes a little trick used by the author. The 'Perceptron' table should be mutable into as many versions as there are variables in the set ***X***, in order to simulate the way this intelligent structure optimizes each of those variables, alternatively, as the output variable. Computation of the vector ***h*** is part of such mutation. Thus, **cell CG3** in column CG is formulated as: ***=SUM(AR3:CF3)-AV3***. In this specific version of the spreadsheet, the perceptron optimizes hours worked, or AVH, as output variable. The component weighed version of this variable is to find in column AV, and, consequently, value in this column is to eliminate from the computation of the vector h: this is the '-AV3' component in the formula to find in cell CG3.

In **the first row of the perceptron**, each of the component values for computing the vector $\boldsymbol{h}$ is formulated as: **=RAND()*{address of the cell with standardized values in the "Initial values" section in columns C ÷ AQ}**. The 'RAND()' factor is, of course, the generator of pseudo-random values, which corresponds to the random weight '$\boldsymbol{w}$' in the theoretical structure of the neural network.

In each further row of the perceptron, thus in phenomenal occurrences $2 \leq j \leq 3008$ in this precise case, cells in columns AR ÷ CF are fed with local fitness functions of the corresponding variables, observed the previous phenomenal occurrence. Thus, **cell AR4** spells: =RAND()*C3*CP3, **cell AS4** is: =RAND()*D4*AK3 etc. References 'AJ3', and 'AK3' connect to cells with local fitness functions from the preceding round.

Columns CH ÷ CM in the 'Perceptron' table generate the **functions of neural activation**. The plural 'functions' refers to the fact that in this precise structure of the spreadsheet, two classical activation functions have been dropped into the toolbox: the sigmoid in **columns CH ÷ CJ**, and the hyperbolic tangent in **columns CK ÷ CM**. The theoretical structure of the neural network, as described in the preceding section of this paper, uses the hyperbolic tangent, still the sigmoid can be useful in some simulations. In the author's experience, the sigmoid function, expressed as $g(j) = \frac{1}{1+e^{-h}}$, where '$\boldsymbol{h}$' is the same vector $\boldsymbol{h}$ as discussed previously, has the capacity to level out sudden surges in the input data. Speaking in metaphorically psychological terms, the sigmoid calms down the reaction to surprising states of nature, whilst the hyperbolic tangent corresponds to a much less filtered neural reaction. That 'calming' property of the sigmoid makes it learn faster, as it drives the residual error asymptotically down to zero at a noticeably faster pace. Still, in neural networks it is just as in real life: faster learning does not necessarily mean more accurate a learning. The sigmoid is relatively more prone to the so-called 'overfitting', or too hasty a conclusion from the data at hand. As the overall logical structure is applied to social systems, the hyperbolic tangent corresponds to weak (or even non-existent) assumptions as for counter-cyclical economic policies. On the other hand, the sigmoid can be useful in simulating social systems with some economic filters on, such as significant automatic stabilizers in fiscal policy etc.

Each of the two neural functions in the 'Perceptron' table is spread into three distinct columns: the first column computes the value of the activation function ***g(j)*** as such, the second one computes its first derivative ***g'(j)***, and the third one computes the local error ***e(j)*** as in equation (3). The 'Perceptron' table contains additional features that allow steering the process of feeding the error e(j) forward into consecutive phenomenal occurrences. These features are to find in **columns CN and CO**. As it was stated previously, error is fed from cells in **column CM**, and yet this column can be used as an autonomous layer in the neural network. In the basic version of the perceptron, such as outlined in the theoretical section of this paper, the error to feed forward is that generated by the hyperbolic tangent, thus by cells in columns **CK ÷ CM**. Thus, in this baseline case, **cell CO3** spells simply '=CM3' etc. Still, mutations are possible: errors from the two neural functions – the sigmoid and the hyperbolic tangent – can be put in competition or in congruence, by averaging them, weighted-averaging etc. In such cases, cells in column CO can take on formulas like '=CJ#*0,5+CM#*0,5'.

Column CN contains something that the author designates as '**coefficient of memory**'. This is an instrumental extension out of the theoretical model presented in the previous section. It is simply the ordinal number of the given phenomenal occurrence, and it refers theoretically to the theory of social games with imperfect recall such as discussed by Reinhard Selten (Selten 1990). The baseline assumption of the here-presented method is that each consecutive phenomenal occurrence in the neural network incorporates all the previous learning, on the preceding errors and observed fitness functions. Once again, theoretical branching is possible. Societies can be forgetful, which means that new learning is more important than old learning. Analytically, it means that error ***e(j)*** to propagate from cells in column CO can be divided by '*1/{coefficient of memory in column CN}*. Conversely, societies can be conservative, which implies lesser importance attached to new learning, as opposed to old learning. In such case, error ***e(j)*** gets multiplied by '*1/{coefficient of memory in column CN}*.

After the table ‘Perceptron’ in the spreadsheet, the reader will find **the table** ‘**Notes**’, which is exactly what it states: a notebook. In its baseline look, **columns B ÷ AP** contain the average values from **columns C ÷ AQ in the table ‘Perceptron’**. In the case of the specific dataset that serves to exemplify the logical structure at hand, row #4 in ‘Notes’ repeats, for the sake of convenience, the mean values of each variable studied, as in the table ‘Source Data’. Thus, **cell B4** in the table ‘Notes’ refers: **='Source Data'!C3011**, **cell C4** spells: **='Source Data'!C3011** etc. In row #5, the spreadsheet calculates the mean values of each variable as transformed by the specific version of the perceptron, thus as means of the set S_1, according to equation (5). Cell B5, for example, contains the formula “=Perceptron!C3011*'Source Data'!C3010”, where the factor “Perceptron!C3011” is the mean value of variable RGDPE, as transformed in the table ‘Perceptron’, and the factor “'Source Data'!C3010” is the maximum of that variable in the source set X. This is reverse-standardization (destandardization). De-standardized means from row #5 can be directly fed into equation (5), in order to compute the corresponding Euclidean similarity with the original set ***X***.

The last issue to discuss as regards the Excel spreadsheet used to emulate a neural network is the way of building its many mutations, or multiple sets S_i, defined by the exact variable from the set X pegged as the output one. The output variable has three main characteristics in the ‘**Perceptron**’ table. First of all, in the section ‘Initial values’, i.e. in columns **C ÷ AQ** of ‘Perceptron’, the output variable is not fed with error ***e(j-1)***. Thus, each line in the column corresponding to the output variable simply refers to its sibling in the ‘Standardized’ table, whilst input variables in the remaining columns in the section **C ÷ AQ** are fed with the local error. Second of all, the output variable is subtracted from the sum total of weighted values, calculated individually in columns AR ÷ CF, and then summed in column CG. Thus, the output variable is the ‘-{cell reference}’ component in the formula to be found in the cells of column CG. In the third place, the output variable serves to calculate the local error ***e(j)*** in columns CJ and CM of the ‘Perceptron’ table, respectively for the sigmoid function, and the hyperbolic tangent. Modification of these three features in the ‘Perceptron’ table reorients the neural network on optimizing another variable as the output one, and thus creates an alternative set S_i.