Causally interpretable multi-step time series forecasting: A new machine learning approach using simulated differential equations.

By: William Schoenberg (University of Bergen, Norway)

Abstract
This work represents a new approach which generates then analyzes a highly non-linear complex system of differential equations to do interpretable time series forecasting at a high level of accuracy. This approach provides insight and understanding into the mechanisms responsible for generating past and future behavior. Core to this method is the construction of a highly non-linear complex system of differential equations that is then analyzed to determine the origins of behavior. This paper demonstrates the technique on Mass and Senge’s two state Inventory Workforce model (1975) and then explores its application to the real world problem of organogenesis in mice. The organogenesis application consists of a fourteen-state system where the generated set of equations reproduces observed behavior with a high level of accuracy ($0.880 r^2$) and when analyzed produces an interpretable and causally plausible explanation for the observed behavior.

Introduction:
Accurate time series forecasting is very important to a variety of scientific fields, engineering disciplines, and socially constructed systems including businesses, and governments (Palit & Popovic, 2006). Past efforts on this problem have focused on developing more accurate methods or models useful for predicting time series data, starting with linear statistical models and evolving into non-linear models and ultimately machine learning techniques (Bontempi, et al, 2012). The major emphasis of much of this research has been on improving prediction accuracy and applicability, and only more recently has interpretability of the underlying mathematical structure responsible for generating the predictions become of significant importance to the end-user of the generated models (Herrera et. al, 2007, Taieb et. al, 2012, Xing et.al, 2014, Doshi-Velez & Kim, 2017, Nemat et. al, 2018).

Early approaches in time series forecasting were characterized by linear statistical methods such as ARIMA models. By the late 1970s it became well understood that linear approaches were not up to the task for ‘real-world’ applications (De Gooijer & Hyndman, 2006). This led to the development of several generally useful analytical non-linear techniques including: the bilinear model, the threshold autoregressive model and the autoregressive conditional heteroscedastic (ARCH) model (De Gooijer & Kumar, 1992, De Gooijer & Hyndmanm, 2006).

More recent approaches to time series forecasting rely on machine learning in the form of Artificial Neural Networks (ANNs). ANNs of this type are commonly referred to as black-box, data-driven models, which by their very nature do not present an interpretation of causal structure (Bontempi et. al, 2012). Significant research exists in the field demonstrating the utility of ANNs to perform time series forecasting, outperforming both the classical linear statistical methods and even the analytical non-linear techniques (Werbos, 1974, Lapedes & Farber, 1987, Werbos, 1988). Basic feedforward ANNs are composed of layers of nodes all
taking input from each other have been developed and employed since the 1940s as a way of generating predictive models using neurological signal transmission as an inspiration (McCulloch and Pitts, 1943, Hebb, 1949). These techniques have evolved over time, by way of more sophisticated training and optimization techniques, as well as by generating more complex networks known including recursive neural nets (Schmidhuber, 2015). The basis for the structure of the nodes in most classes of ANNs is the perceptron which is described as an algorithm for pattern recognition (Rosenblatt, 1958). RNNs are structured in such a way that a particular perceptron will take input from other perceptrons and itself (Goller & Kuchler, 1996).

Interpretability of the underlying mathematical models producing the predictions is a critically important step in the process of validating a generated model for an application. Interpretability in the sense discussed in this paper allows the end-user to answer the question: ‘why does my model give me this result?’ This sets the stage not only for the process of model improvement based upon the answer to that question, but also begins the process of understanding complex systems by using mathematics to represent the system under study (Senge & Forrester, 1980, Forrester, 1994).

This work represents a new approach which generates then analyzes a highly non-linear complex system of differential equations to do interpretable time series forecasting with a high level of accuracy. This approach provides insight and understanding into the mechanisms responsible for generating past and future behavior.

The method
Core to this method is the construction of a highly non-linear complex system of differential equations evaluated over a time dimension. This system of equations is analyzed to determine the origins of behavior. In the generated system of differential equations, the state variables constitute the memory of that system and their relationships (through perceptron firing) are the source from which behavior originates. First the method constructs the system of differential equation which links the state variables together. Second via a process of optimization the system of equations is parameterized to reproduce the measured historical states the system exhibited. Third, the parameterized model is studied and the mathematical relationships between the state variables are objectively measured which clarifies the origins of behavior within the mathematical system. Finally, this explanation is then extracted and validated by subject matter experts, turning the fruits of the method from a black box data-driven model into a transparent and analyzable structure-driven model.

The first step in the method is to generate the system of differential equations to be parameterized. To do that the state variables in the system are enumerated and initialized from historical data. Next the derivative of each state variable is set up to be a function of every other state variable in the system as well as itself. It is these relationships in the derivative functions that represent the opportunity for the learning that will take place during the model parameterization process. The process of constructing the derivatives for each state variable offers the end-user the ability to add a-priori knowledge to the system by specifying which causal relationships are known not to exist, and therefore can be excluded from the
learning process. This is done by eliminating the relationship between any directed pair of source and target state variables.

The relationship between each directed pair of state variables takes the form of equation (1) which is based on a modified perceptron formulation. The perceptron was modified to more easily produce non-linear behavior without having to add additional hidden layers.

\[
Im_{s1s2} = f \left( \left( f \left( \frac{s1}{m_{s1}} \right) + p_{s1s2} \right) * w_{s1s2} \right)
\]

(1)

Equation (1) represents the impact \((Im)\) that state variable 1 \((s1)\) has on the derivative of state variable 2 \((s2)\). \(Im_{s1s2}\) is the term which represents \(s1\) in the function which makes up the derivative of \(s2\). \(f\) is a non-linear activation function which is typically the tanh function which contributes to the ability of this method to reproduce highly non-linear behavior. \(m_{s1}\) is the magnitude of \(s1\) which is typically set as the initial value of \(s1\). The magnitude is used to normalize the state variable to an approximately 1 based scale so that the normalized values of all state variables are directly relatable to each other, reducing the space the optimizer must search during parameterization. \(p_{s1s2}\) is a linear parameter applied to the normalized value of \(s1\) before being multiplied by a non-linear weight \((w_{s1s2})\). This form for the impact of \(s1\) on \(s2\) gives the optimizer two parameters to adjust in order to model the behavior of the relationship between \(s1\) and \(s2\). Only with both of these parameters can the optimizer search through a sufficiently dimensioned space to find a reasonable approximation for the true causal impact of \(s1\) on \(s2\).

The derivative functions for each state variable can take one of two forms (equations 2, 3) each using the terms shown in equation (1). These two forms are combined via an IF statement with a state variable specific switch parameter that is turned on or off by the optimizer to move between the two forms. The magnitude \((m)\) of the state variable is included in these equations to rescale the normalized values back to the appropriate scales for the actual derivative.

\[
\frac{ds2}{dt} = Im_{s1s2} * Im_{s2s2} \ldots * Im_{sns2} * m_{s2}
\]

(2)

\[
\frac{ds2}{dt} = \left( Im_{s1s2} + Im_{s2s2} \ldots + Im_{sns2} \right) / t_{s2} * m_{s2}
\]

(3)

Equation (2) is a pure nonlinear form required to generate complex dynamic behavior such as the logistic curve observed in the bass diffusion model (Bass, 1969). Equation 3 represents either the linear combination of state variables when \(t_s = 1\) or an exponential smooth formulation in all other cases. The term \(t_s\) is conditionally optimized only when this form is active and represents the smoothing time for that state variable. These two forms are required
to provide a wide enough space for the optimizer to search for a reasonable approximation of the true impact of the state variables, 1 through n, on the state variable \( s_2 \) being calculated.

The second step is model parameterization or training, where an optimization process is run defining as a payoff function the minimization of the mean squared error between the scaled values of all calibration data for each state variable. The dimensionality of the parameter space to search is \( 2n^2 + 2n \) where \( n \) represents the number of state variables assuming a fully connected network with no a-priori information. In the examples below either Euler or Runge-Kutta 4 integration was used to solve the generated differential equation over the chosen time period. The optimization algorithm used in all cases was Powell’s BOBYQA (bound optimization by quadratic approximation) as implemented in the public open source project DLib version 19.7.

The third step, once the parameterized model has been generated, is to do model analysis using the link score metric from the loops that matter method (Schoenberg et. al, 2019). The link score metric reports the contribution at a point in time that an independent variable \( x \) has on a dependent variable \( z \). For this approach the link score is used to measure the impact that one state variable has on another so that the causal relationship between state variables can be quantified and understood. Equation (4) is reproduced from Schoenberg et. al. below. Link scores are computed using the solution interval employed to numerically solve the differential equations.

The link score for the link \( x \to z \) is:

\[
LS(x \to z) = \begin{cases} \\
\left( \frac{\Delta x}{\Delta z} \right) \cdot \text{sign} \left( \frac{\Delta z}{\Delta x} \right), & \text{if } \Delta z \neq 0 \text{ and } \Delta x \neq 0 \\
0, & \text{if } \Delta z = 0 \text{ or } \Delta x = 0
\end{cases}
\]

(4)

Where \( \Delta z \) is the change in \( z \) from the previous time to the current time, \( \Delta x \) is the change in \( x \), and \( \Delta x \cdot \Delta z \) is the amount \( z \) would have changed, conditionally, if only \( x \) had changed (all other dependencies held constant). The first term in this equation represents the magnitude of the contribution, the second the polarity.

The magnitude of the effect (force is a good analogy) that \( x \) has on \( z \) is relative to all of the effects on \( z \). This is a dimensionless quantity, and if all of the effects are in the same direction, it is the fraction of the change in \( z \) that originates in a change in \( x \). If the formulation of \( z \) is linear, then the values are restricted to the range \([0,1]\). When there are negative and positive effects, these numbers may be very large in magnitude, but this does not harm the overall analysis of the system of differential equations (Schoenberg, et. al, 2019). The absolute value is used because the change in \( z \) could be in either direction due to the effects from other variables, regardless of the magnitude of the effect that \( x \) has, implying that the polarity can and would be wrong.
The polarity of a link is defined as the sign of the partial difference at time t. This formulation is the same as the one used in Richardson (1995), though the formulation there was as a partial derivative, not difference. The polarity numerator is the same as it is for the magnitude, but the denominator is the change in x. When x does not change, the score is definitionally 0, though the magnitude would be 0 in any case.

Link scores can be multiplied together following the chain rule of partial differentiation allowing the calculation of the system of equations to proceed at any desired aggregation level (that is with more or fewer intermediate algebraic computations). Typically for systems generated using this technique link scores are calculated along the pathway of the relationship from one state variable directly to the derivative of another state variable. It is possible to calculate link scores which pass through other state variables back to the original state variable forming complete feedback loops. This is valuable for models with known or physically bound structure, but of less value for all but the simplest of generated models. This is because this method of generating differential equations produces a number of feedback loops which is the factorial of the number of state variables, numbers which quickly overwhelm computational and analytic evaluations.

When following the standard practice of calculating link scores which measure the impact of one state variable (source) on the derivative of another (target), the generated link scores are then normalized across all of the relationships which have the same target, at each point in time. This produces a signed percentage score which describes specifically and objectively what percentage of the behavior of the target state variable is being ascribed to the source state variable, including what the polarity of their relationship is at that specific point in time.

Communicating the results of this analysis is typically done by generating a network diagram using a force directed graph which plots the strength of each relationship over time in stepwise increments. Each link in the network diagram takes on a thickness which is directly proportional to the magnitude of its link score. This produces a living diagram of the importance of relationships inside the system of differential equations responsible for doing time series forecasting.

**Recasting a known theoretical model**

Before studying a real-world application of the approach, let’s look at how the process works in the context of a well understood theoretical model. This allows for an evaluation of the approach on a known system so that the validity of the generated causal explanation can be objectively ascertained. The theoretical model for this example is Mass & Senge’s 1975 two state oscillator commonly known as the Inventory Workforce model whose actual system of equations is replicated below in equation (5) and was simulated using an Euler algorithm with a dt of 1/4.

\[
\begin{align*}
\frac{d}{dt} \text{Inventory} &= (\text{Workers} \times \text{productivity} - \text{demand}) \\
\frac{d}{dt} \text{Workers} &= ((\text{target_production/} \text{productivity} - \text{Workers})/t1)
\end{align*}
\]
$$\text{target\_production} = \text{demand} + (\text{target\_inventory} - \text{Inventory})/t2$$
$$\text{target\_inventory} = 100$$
$$\text{productivity} = 2$$
$$t1 = 6$$
$$t2 = 1$$

(5)

For this case the generated system of equations appears in equation (6), and its parameterization in Table 1.

$$\frac{d}{dt} \text{Inventory} = (\text{producing} - \text{demand})$$
$$\frac{d}{dt} \text{Workers} = (\text{hiring\_or\_firing})$$
$$\text{producing} = (\text{WIf} - \text{IIf})/\text{tp}\*\text{Im}$$
$$\text{hiring\_or\_firing} = (\text{IWf} - \text{WWf})/\text{th}\*\text{Wm}$$

(demand and initial Inventory and Workers are known)

$$\text{Ir} = \tanh(\text{Inventory}/\text{Im})$$
$$\text{IIf} = \tanh((\text{Ir}+\text{IIfp})*\text{IIw})$$
$$\text{IWf} = \tanh((\text{Ir}+\text{IWp})*\text{IWw})$$
$$\text{Wr} = \tanh(\text{Workers}/\text{Wm})$$
$$\text{WIf} = \tanh((\text{Wf}+\text{Wf})*\text{Wf})$$
$$\text{WWf} = \tanh((\text{Wf}+\text{Wf})*\text{WWf})$$

(6)

The system of equations (6) represents the following a-priori knowledge of the system. First, that there are two state variables, Inventory and Workforce. Second, that the starting values for Inventory and Workforce are 100 and 10 respectively and finally that there is step in demand from 20 units to 25 units at time 2. The key result here is that the false feedback relationship (IIf) between Inventory and its own derivative in equation (6) which does not appear in the ground truth system, equation (5), is severely weakened by the parameterization process setting a near 0 weight on that relationship as shown in Table 1 and seen in Figure 2 where the direct inventory to inventory relationship is generally not dominant determining less than 10% of the generated system’s behavior over the entire 80 time unit period.

Table 1: Parameterization of Inventory Workforce system of equations

| Symbol | Parameter Name               | Initial Value | Discovered Value | Optimization Range |
|--------|------------------------------|---------------|------------------|--------------------|
| IWp    | inventory to workers param   | 0             | -0.783           | -10, 10            |
| IWw    | inventory to workers weight  | 1             | -4.42            | -10, 10            |
| WWp    | workers to workers param     | 0             | -0.759           | -10, 10            |
| WWw    | workers to workers weight    | 1             | 1.10             | -10, 10            |
| th     | workers delay                | 1             | 2.57             | 1, 10              |
| -      | hiring or firing switch      | 1             | 1                | 0, 1               |
| th     | workers delay                | 1             | 2.57             | 1, 10              |
Figure 1: The simulated results of the parameterized generated system of equations for Inventory with an $r^2$ value of 0.976.
The standard analysis of the ground truth Inventory Workforce model in the literature states that the dominant relationship controlling the behavior of this system is the major loop from inventory back to itself via workforce (Sterman, 2000). The results in Figure 2 show the loop scores (link scores multiplied through the three feedback loops in the system) demonstrating that the generated system’s behavior is dominated by the same major loop as seen in the ground truth system. This means the two systems of equations are analogous, producing similar behavior for similar reasons. The same result could also be ascertained in this simple system by studying the generated weights which show the strongest weight is on the relationship between inventory and workforce and therefore any feedback loop involving that relationship must be dominant (Table 1).

**Application to an unknown system, mouse organogenesis**

Knowing now that it is possible to reproduce complex behavior by generating a causally accurate system of differential equations lets apply the same process to a system with unknown causal interactions in order to evaluate and discuss the merits of the approach.

The system under study in this section is a mouse embryo, from embryonic day 6.5 to 8.5. During this time period in mouse embryonic development the cells of the embryo mature from generally undifferentiated pluripotent cells into a gastrula composed of a variety of proto-organs (Tam & Behringer, 1997). The goal with the development of the system in this section is to understand at a genetic level the relationships between the various biological processes occurring in the mouse embryo and how those interactions give rise to the development of proto-organs from undifferentiated cells. The data were collected and organized by the
Marioni Lab and published as a part of Pijuan-Sala et al., 2019. The researchers sampled 116,312 cells from a series of mouse embryos taking readings every 6 hours across the two embryonic days. Then for each cell sample a full genetic profile was performed measuring the expression level for 29,452 genes.

With over 17 million gene expression level samples in the raw dataset, aggregation needed to be performed in order for a hypothesis system to be generated. The basis for the aggregation is the Mouse Gene Ontology slim (Mouse GO slim) produced by the Mouse Genome Informatics (MGI) project (Bult et al., 2019). A GO slim is a subset of the network of gene ontology terms which group genes by three major aspects. The aspect of interest to this work is the biological processes the gene is involved with. The Mouse GO slim represents the sum of current day knowledge about the biological processes, molecular functions and locations of all genes studied by biologists around the world. The Mouse GO slim allows for the categorization of each gene by its biological process into one or many of the 14 major biological processes (Table 2) which the MGI has selected as being key to mouse function.

The aggregation process was completed resulting in a dataset containing 14 state variables which are the average gene expression level for all genes categorized by the Mouse GO slim into each of the 14 biological processes. The dataset contained 9 data points for each state variable. The generated system of equations used to reproduce that dataset appear as equation (7) and were solved using an RK4 integration method with a dt of 1/64 days.

\[
\frac{d}{dt} \text{Measure} = \sum \left( \frac{\tanh \left( \left( \overrightarrow{Mr} + \overrightarrow{p} \right) \cdot \overrightarrow{w} \right)}{\overrightarrow{t}} \right) \cdot \overrightarrow{m}
\]

\[
\overrightarrow{Mr} = \tanh \left( \frac{\text{Measure}}{\overrightarrow{m}} \right)
\]

Equation (7) is a result of the direct application of the method. \( \text{Measure} \) represents a vector of the 14 state variables. \( \overrightarrow{Mr} \) is the vector of reduced normalized measures after each has been divided its magnitude (\( \overrightarrow{m} \)) and passed through the activation function (\( \tanh \)). \( \overrightarrow{p} \) is the vector of linear transformations and \( \overrightarrow{w} \) is a vector of the non-linear transformations. Finally, \( \overrightarrow{t} \) is the vector of the smoothing times for each of the state variables.

The multiplicative form for the derivative equation was not considered during the final system parameterization and does not appear in equation (7) because during early tests when it was included, the generated causal explanations which used this form were invalid when assessed based on biological dependence. Eliminating the multiplicative form from the search space decreased search time without decreasing system fit and significantly increased the utility of the causal explanation.
Table 2 demonstrates the ability of the new method to accurately explain the variance in the observed (aggregated) data with an $r^2$ value of 0.880 averaged across all 14 state variables in the parameterized system.

Figures 3 shows the relationships that are in the 75% and greater quartile of all explanatory relationships averaged over the entire two-day time period. Each relationship in Figure 4 explains at least 6.99% of the cumulative behavior of the target state variable by the source over this time period.

Examining the relationships between the state variables in the system over time yields an interesting insight into the periodic nature of this system. As shown in Figure 5 there are consistent shifts between two major system configurations for explaining the sources of generated behavior. The first major configuration is when the genes responsible for the protein metabolic process are clearly most important in explaining the behavior of the other biological processes. These states are always followed by more dispersed explanations which are characterized by a strong importance of the genes responsible for the nucleic acid templated transcription process. There are 7 transitions between these two general system configurations throughout the two-day period studied. These transitions correlate strongly with 7 local minima and maxima observed in the generated output for each of the 14 state variables.

### Table 2: $r^2$ values by biological processes

| Parameter Name                          | $r^2$  |
|-----------------------------------------|--------|
| Cellular Component Organization         | 0.974  |
| Nucleic Acid Templated Transcription    | 0.896  |
| System Development                      | 0.894  |
| Cell Differentiation                    | 0.956  |
| Response to Stimulus                    | 0.987  |
| Signaling                               | 0.832  |
| Cell Population Proliferation           | 0.859  |
| Protein Metabolic Process               | 0.892  |
| Establishment of Localization           | 0.898  |
| Lipid Metabolic Process                 | 0.711  |
| Immune System Process                   | 0.788  |
| Cell Death                              | 0.969  |
| Carbohydrate Derivative Metabolic Process| 0.897  |
| Homeostatic Process                     | 0.773  |
| **System Average**                      | **0.880** |
Figure 4: Diagram which represents the cumulative impact of each of the 14 state variables on each other. Only connections in the fourth quartile of strength are shown (6.99% or greater). The thicker the arrow, the stronger the impact over the two-day time period.

Figure 5: The two characteristic system configurations for explaining the sources of generated behavior. The diagram on the left shows the dominating importance of the protein metabolic process (teal) to explaining system behavior. The second shows a more dispersed explanation for behavior, but with a strong influence from the nucleic templated transcription process (orange) as being important in these time periods.

Conclusions
This work represents a new approach to multi-step time series forecasting which generates causally interpretable explanations that can be validated and used to improve the utility of the generated forecasting model. It’s correctness and utility have been demonstrated in both theoretical known systems and unknown complex real world systems. The technique presented here is the first step along what promises to be an interesting research pathway which may provide understanding about some of the most difficult and complex problems faced in biology and other areas of dynamic complexity.

The real benefit of this approach is not in its ability to predict future values, but rather the its ability to generate well-formed models that have interpretable causal structures so that insight may be gained into the mechanisms uncovered by the algorithm that produce the target behavior. The patterns of shifting relationships in these models should help to shed light on why observed phenomena arise and allow us to turn black box machine learning techniques into transparent and understandable system representations.

The next steps in this research are to expand its scope to larger more complex systems and utilize more powerful optimization techniques, especially those which can be run on parallel processors. By increasing the power of the optimizer, it should be possible to generate larger, more complex systems for which there is no standard aggregation model to apply to the input data. Other avenues of investigation include research into the utility of adding additional hidden layers of modified perceptrons between state variables in order to produce more nuanced relationships.

References
1. Bass, F. M. (1969). A new product growth for model consumer durables. Management science, 15(5), 215-227.
2. Bontempi, G., Taieb, S. B., & Le Borgne, Y. A. (2012, July). Machine learning strategies for time series forecasting. In European business intelligence summer school (pp. 62-77). Springer, Berlin, Heidelberg.
3. Bult CJ, Blake JA, Smith CL, Kadin JA, Richardson JE, the Mouse Genome Database Group. 2019. Mouse Genome Database (MGD) 2019. Nucleic Acids Res. 2019 Jan. 8;47 (D1): D801–D806.
4. De Gooijer, J. G., & Hyndman, R. J. (2006). 25 years of time series forecasting. International journal of forecasting, 22(3), 443-473.
5. De Gooijer, J. G., & Kumar, K. (1992). Some recent developments in non-linear time series modelling, testing, and forecasting. International Journal of Forecasting, 8(2), 135-156.
6. DLIB [Computer software]. (2017). Retrieved from https://github.com/davisking/dlib
7. Forrester, J. W. (1994). System dynamics, systems thinking, and soft OR. System dynamics review, 10(2-3), 245-256.
8. Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608.
9. Goller, C., & Kuchler, A. (1996, June). Learning task-dependent distributed representations by backpropagation through structure. In *Proceedings of International Conference on Neural Networks (ICNN’96)* (Vol. 1, pp. 347-352). IEEE.
10. Herrera, L. J., Pomares, H., Rojas, I., Guillén, A., Prieto, A., & Valenzuela, O. (2007). Recursive prediction for long term time series forecasting using advanced models. *Neurocomputing, 70*(16-18), 2870-2880.
11. Lapedes, A., & Farber, R. (1987). *Nonlinear signal processing using neural networks: Prediction and system modelling* (No. LA-UR-87-2662; CONF-8706130-4).
12. Mass, N. J., & Senge, P. M. (1975). Understanding oscillations in simple systems. *System Dynamics Group Working Paper D-2045-2, MIT.*
13. McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics, 5*(4), 115-133.
14. Nemati, S., Holder, A., Razmi, F., Stanley, M. D., Clifford, G. D., & Buchman, T. G. (2018). An Interpretable Machine Learning Model for Accurate Prediction of Sepsis in the ICU. *Critical care medicine, 46*(4), 547-553.
15. Palit, A. K., & Popovic, D. (2006). *Computational intelligence in time series forecasting: theory and engineering applications.* Springer Science & Business Media.
16. Pijuan-Sala, B., Griffiths, J. A., Guibentif, C., Hiscock, T. W., Jawaid, W., Calero-Nieto, F. J., ... & Reik, W. (2019). A single-cell molecular map of mouse gastrulation and early organogenesis. *Nature, 566*(7745), 490.
17. Richardson GP. 1995. Loop polarity, loop dominance, and the concept of dominant polarity. *System Dynamics Review* 11(1): 67–88.
18. Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review, 65*(6), 386.
19. Socher, R., Lin, C. C., Manning, C., & Ng, A. Y. (2011). Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (pp. 129-136).
20. Senge, P. M., & Forrester, J. W. (1980). Tests for building confidence in system dynamics models. *System dynamics, TIMS studies in management sciences, 14,* 209-228.
21. Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks, 61,* 85-117.
22. Schoenberg, William, Davidsen, Pål, Eberlein, Robert, (2019). Understanding model behavior using loops that matter. *arXiv preprint arXiv: submit/2818896.*
23. Sterman, J. (2000). *Business dynamics.* Irwin/McGraw-Hill c2000.
24. Taieb, S. B., Bontempi, G., Atiya, A. F., & Sorjamaa, A. (2012). A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert systems with applications, 39*(8), 7067-7083.
25. Tam, P. P., & Behringer, R. R. (1997). Mouse gastrulation: the formation of a mammalian body plan. *Mechanisms of development, 68*(1-2), 3-25.
26. Werbos, P. (1974). Beyond Regression:" New Tools for Prediction and Analysis in the Behavioral Sciences. *Ph. D. dissertation, Harvard University.*
27. Werbos, P. J. (1988). Generalization of backpropagation with application to a recurrent gas market model. *Neural networks, 1*(4), 339-356.
28. Xing, Z., Pei, J., Yu, P. S., & Wang, K. (2011, April). Extracting interpretable features for early classification on time series. In *Proceedings of the 2011 SIAM International Conference on Data Mining* (pp. 247-258). Society for Industrial and Applied Mathematics.