Linear Neural Networks Revisited: From PageRank to Family Happiness

Vladik Kreinovich
Department of Computer Science
University of Texas at El Paso
500 W. University
El Paso, TX 79968, USA
{paulo,vladik}@utep.edu

Abstract

The study of Artificial Neural Networks started with the analysis of linear neurons. It was then discovered that networks consisting only of linear neurons cannot describe non-linear phenomena. As a result, most currently used neural networks consist of non-linear neurons. In this paper, we show that in many cases, linear neurons can still be successfully applied. This idea is illustrated by two examples: the PageRank algorithm underlying the successful Google search engine and the analysis of family happiness.

1 Linear Neural Networks: A Brief Reminder

Neural networks. A general neural network consists of several *neurons* exchanging signals. At each moment of time, for each neuron, we need finitely many numerical parameters to describe the current state of this neuron and the signals generated by this neuron. The state of the neuron at the next moment of time and the signals generated by the neuron at the next moment of time are determined by its current state and by the signals obtained from other neurons.

Non-linear and linear neural networks. In general, the dependence of the neuron's next state and/or next signal on its previous state and previous signals is non-linear; see, e.g., [5]. However, original neural networks started with linear neurons, and, as we will argue, there are still cases when linear neurons work well.

What we do in this paper. In this paper, we show that there are indeed many useful applications of linear neural networks.

2 Linear Neural Networks: A Precise Description

Description. The state of each neuron i at each moment of time t can be described by the values $w_{i,1}(t), \ldots, w_{i,n_i}(t)$ of the corresponding parameters. Since we consider linear neural networks, the transformation to the next moment of time is linear in each of these parameters:

$$w_{i,j}(t+1) = a_{i,j} + \sum_{k,l} a_{i,j,k,l} \cdot w_{k,l}(t).$$

Simplification: step one. In neural networks, to simplify the description, it is usually convenient to add a fictitious parameter w_0 whose value is 1; then, the above formula takes the simplified form

$$w_{i,j}(t+1) = \sum_{k,l} a_{i,j,k,l} \cdot w_{k,l}(t),$$

where now the sum includes the case when $(k,l) = 0$; for this case, we take $a_{i,j,0} = a_{i,j}$.

Simplification: step two. This formula can be further simplified if we ignore the relation between parameters and neurons and simply consider all parameters of all the neurons. In precise terms, we take A to be an index that goes over all pairs (i,j). In this case, we have

$$w_A(t+1) = \sum_B a_{A,B} \cdot w_B(t).$$

Asymptotic behavior is what we are interested in. In a general neural network, the dynamical process is important, we what is even more important is what happens after all these changes are done. In other words, the learning process is important but the main objective is the result of this process. From this viewpoint, we are interested in the asymptotic behavior of the above dynamical system.

How to describe asymptotic behavior: enter eigenvectors and eigenvalues. The behavior of a general linear dynamical system is easier to describe if we use the basis consisting of the eigenvectors of the matrix a_{AB}. In this basis, the matrix becomes diagonal, with the eigenvalues λ_A on the diagonal, so each component gets transformed as $w_A(t+1) = \lambda_A \cdot w_A(t)$. Thus, we get $w_A(t) = \lambda_A^t \cdot w_A(0)$.

In the general case, all initial values $w_A(0)$ are non-zeros. In this case, for large t, coefficients $w_A(t)$ corresponding to the eigenvalue with the largest $|\lambda_A|$ become thus much larger than all other coefficients. Hence, for large t, the state

vector becomes proportional to the eigenvector corresponding to this largest eigenvalue.

So, asymptotically, a linear neural network transforms the original state into the eigenvector corresponding to the largest eigenvalue (largest by the absolute value).

Comment. The above ideas can be viewed as a particular case of the general ideas of neural-network-related *dynamics logic*; see, e.g., [15, 16, 17].

3 Linear Neural Networks: From the General Case to the Simplest Case

Simplest case: we only know connections, not weights. In general, to describe a linear neural network, we need to describe real values a_{AB} corresponding to different pairs of nodes. In geometric terms, $a_{AB} \neq 0$ means that there is a connection from node B to node A. Thus, to describe a linear neural network, we need to describe:

- all the connections between the nodes, and
- for connected pairs of nodes, the strength a_{AB} of this connection.

The simplest case is when we do not know the strengths, we only know which neurons are connected to each other.

Let us see how in this case, we can naturally define the strengths.

Limitations on the values w_A of the state variables. In general, as we have mentioned, the values $w_A(t)$ of the state variables grow exponentially with time t. To restrict this growth, we can put a restriction on these values. The simplest limitations are linear ones, i.e., limitations of the type

$$\sum_A c_A \cdot w_A = \text{const}.$$

In the generic case, the constant is not 0. By dividing all the coefficients c_A by this constant, we get a simplified relation $\sum\limits_A c_A \cdot w_A = 1$.

A priori, we have no reasons to prefer one state variable over another. Thus, it makes sense to require that the weights c_A corresponding to all the state variables are the same, i.e., that $c_A = c$ for all A and for some constant c. Thus, we conclude that $\sum\limits_A c \cdot w_A = 1$.

We can simplify this limitation even further if we change the unit in which we measure the values of the variables w_A to a new measuring unit which is c times smaller, so that 1 old unit is equivalent to c new units. With this new choice of a unit, the value w_A in the old units becomes $c \cdot w_A$ in the new units.

Thus, if we describe the values w_A in the new units, the above limitation takes a simplified form

$$\sum_A w_A = 1.$$

Resulting limitations on dynamics. The above limitation on the states means that we have to restrict dynamics as well: to make sure that if the above restriction holds at some moment of time, then it will also hold in the next moment of time. In precise terms, if $\sum\limits_A w_A = 1$ and $w'_A = \sum\limits_B a_{AB} \cdot w_B$, then we should have $\sum\limits_A w'_A = 1$.

Let us describe this relation in terms of the coefficients a_{AB}. By substituting the expression for w'_A into the desired formula $\sum\limits_A w'_A = 1$, we can reformulate the above condition as follows: if $\sum\limits_A w_A = 1$, then $\sum\limits_A \sum\limits_B a_{AB} \cdot w_B = 1$, i.e., in other words, $\sum\limits_A c_A \cdot w_A = 1$, where $c_A \stackrel{\text{def}}{=} \sum\limits_B a_{BA}$.

In geometric terms, both the original equation $\sum\limits_A w_A = 1$ and the new equation $\sum\limits_A c_A \cdot w_A = 1$ describe planes, and the above if-then condition means the first plane is a subset of the second plane. This can only happen when the two planes coincide, and this, in turn, means that the corresponding coefficients must coincide, i.e., that $\sum a_{AB} = 1$ for all A.

Resulting formulas for the weights. For each A which is connected to some variables B, we need to assign weights a_{BA} to all such connections. Since there is no reason to distinguish different B from which there is a connection from A, it makes sense to make all these weights equal to each other. Due to the condition $\sum\limits_B a_{BA} = 1$, this means that $a_{BA} = \dfrac{1}{n_A}$, where n_A denotes the total number of variables B for which there is a connection from A to B.

For some variables A, there is no connection to any B's. In this case, similarly, there is no reason to prefer one of the B's, so it makes sense to assign equal weights to all the corresponding variables a_{BA}. Thus, for such variables A, we get $a_{BA} = \dfrac{1}{n}$, where n is the total number of the variables.

A problem. Our main interest is in eigenvectors. The problem is that for the above matrix a_{AB}, there are many different eigenvectors. For example, if we have a group of variables which are only connected to each other, then there is an eigenvector in which $w_A \neq 0$ only for these variables A. To get this eigenvector, we can start, e.g., with equal weights assigned to all these variables. Since these variables are only connected to each other, we will never get $w_A \neq 0$ for any variable A outside the group.

In general, the abundance of eigenvectors is not a big problem since we distinguish eigenvectors by their eigenvalues – and select only the one with the

largest eigenvalue. However, in our case, due to the restriction $\sum_A w_A(t) = 1$, all eigenvalues are equal to 1.

So, to restrict the number of eigenvectors, it is desirable to modify the original matrix a_{AB}.

How to modify the original matrix. The above problem occurs when some variables are not connected with others. So, to resolve this problem, a natural idea is to add small connections $\Delta a_{AB} \neq 0$ between every two nodes. Again, there is no reason to prefer one pair of variables to any other pair, so it makes sense to make all these values equal to each other: $\Delta a_{AB} = \delta \neq 0$ for all A and B.

If we simply add these values $\Delta a_{AB} = \delta$ to all the entries of the original matrix a_{AB}, we will violate the condition $\sum_B a_{BA} = 1$, since for the matrix $a'_{AB} = a_{AB} + \Delta a_{AB}$, we will have

$$\sum_B a'_{BA} = \sum_B a_{BA} + \sum_B \Delta a_{BA} = 1 + \varepsilon,$$

where we denoted $\varepsilon \stackrel{\text{def}}{=} n \cdot \delta$. To preserve this condition, we should therefore multiply all the entries on the original matrix a_{AB} by $1 - \varepsilon$. This multiplication does not change the eigenvectors and the order of eigenvalues of the original matrix, and for the new matrix $a''_{AB} = (1 - \varepsilon) \cdot a_{AB} + \Delta_{AB}$, we indeed have

$$\sum_B a''_{BA} = (1 - \varepsilon) \cdot \sum_B a_{BA} + \sum_B \Delta a_{BA} = (1 - \varepsilon) + \varepsilon = 1.$$

One can check that for nodes A that have no outgoing connections, the new formula leads to exactly the same equal values a''_{BA} as the formula for a_{BA} – which makes perfect sense, since for this node A, we did not introduce any differentiation between different nodes B. Thus, we arrive at the following natural matrix.

Final formula for the matrix a_{BA}. We start with a small number ε and a directed graph in which vertices correspond to variables A and there is an edge from A to B if and only if the variable A can influence the variable B.

When a node A cannot influence any variable B, i.e., when there is no edge coming out of A in our graph, then we take $a_{BA} = \dfrac{1}{n}$, where n is the total number of nodes.

For every other node A, we take:

- $a_{BA} = \dfrac{\varepsilon}{n}$ for all B for which there is no connection from A, and
- $a_{BA} = (1 - \varepsilon) \cdot \dfrac{1}{n_A} + \dfrac{\varepsilon}{n}$ for every node B for which there is an edge from A to B, where n_A is the number of such nodes B.

4 First Application: PageRank Algorithm as an Example of a Linear Neural Network

PageRank algorithm: brief reminder. Web search engines return all the webpages that contain the desired phrase or keywords. Often, there are thousands and millions of such webpages. So, to make search results meaningful, search engines sort the webpages and only return the top ranking ones.

The most successful and the most widely used search engine – Google – uses a special PageRank algorithm for sorting. This algorithm is based on the following idea. Let us assume that a user starts with a randomly selected page. If this page has no links on it, the user picks another random page – with equal probability of selecting each of the webpages. If there are links, the user either goes to one of these links (with equal probability), or, with a certain probability $\varepsilon > 0$, starts the process anew.

As a result:

- reputable pages, i.e., pages to which many pages link will be visited more frequently, while
- obscure pages, i.e., pages to which no pages link will be visited much less frequently.

The probability of visiting a page in the above process is then used as a rank of this page; see, e.g., [6, 13] and references therein.

PageRank and linear neural networks. One can represent the world wide web as a directed graph, in which webpages are nodes and an edge from page A to page B means that there is a link to page B on page A. From the above description of the PageRank algorithm, one can see that the probabilities a_{BA} of going from A to B are described exactly by our linear neural network formulas, so the final probabilities – ranks of different pages – form an eigenvector of the corresponding matrix.

Thus, PageRank can be viewed as a natural application of linear neural networks.

PageRank and linear neural networks beyond web search. Similar ideas and formulas have been used beyond web search, e.g., to describe the degree of trust of different participants in a peer-to-peer network; see, e.g., [11].

This idea can also be applied to other situations if instead of the full graph of all the connections, we apply this algorithm to the appropriate subgraph. In some cases, this restriction leads to heterogenous networks (see, e.g., [25]); examples will be given below.

Let us list several possible applications of this type. We will see that the result depends on how much information we keep in the subgraph.

First example: ranking papers based on citations. A paper which is cited more is, in general, considered to be more valuable. This is why often, papers are ranked by the number of citations. However, this simplified ranking is – well, simplified: a citation is a more important (more cited) paper should be counted for more than a citation in an obscure paper – a paper that no one cites.

A natural way to assign rank to papers based on citations is to form a directed graph in which nodes are papers and there is a link from A to B if the paper A cites the paper B. Then, we can use the PageRank algorithm – i.e., in effect, linear neural networks – to provide a good ranking of papers by their importance; this idea has been proposed already by the authors of the PageRank algorithm.

Second example: ranking of authors. To nodes describing papers, we can add nodes describing authors, with a link from each author to each of his or her papers, and with links from each paper to each of its authors.

Computational comment. This is an example of a heterogenous graph: we clearly have nodes of two different types – papers and authors.

We can also consider a simplified (and homogeneous) version of this graph, in which there are only author nodes, and each author links to co-authors; it will provide the ranking by intensity of collaborations.

Practical comment. Because of the inter-disciplinary character of modern research, it makes sense, when ranking authors from a certain area (e.g., from geosciences), to include papers from other areas in this graph as well – this way, if a computer science paper is citing a geoscience paper, we give this citation more credit if this computer science paper is itself more important (i.e., in this description, more cited).

Third example: ranking journals and conferences. To do that, we add, to the graph describing authors and papers, additional nodes describing journals and conferences. Then, we have a link from each paper to a journal or conference where it appeared, and we have links from each journal or conference page to all the papers that have been published there.

5 Second Application: Family Dynamics

From the simplest case to the general case. In the previous two sections, we only considered the simplest case when we only know which nodes can influence other nodes, but we do not have specific information about the degree to which they can influence each other. In many practical cases, however, we know these degrees as well. Let us describe a natural example of such a case.

Description of the example. A natural example of dependence is between people. Each person's utility (i.e., degree of happiness, degree of satisfaction) is, in general, determined not only by the objective factors – like what this person gets and what others get – but also by the degree of happiness of other people.

Normally, this dependence is positive, i.e., we feel happier if other people are happy. However, negative emotions such as jealousy are also common, when someone else's happiness makes a person not happy.

The idea that a utility of a person depends on utilities of others was first described in [20, 21]. It was further developed by another future Nobelist Gary Becker; see, e.g., [1]; see also [3, 7, 10, 14, 26].

Interdependence of utilities: general description. In general, the utility u_i of i-th person under interdependence can be described as $u_i = f_i(u_i^{(0)}, u_j)$, where $u_i^{(0)}$ is the utility that does not take interdependence into account, and u_j are utilities of other people.

The effects of interdependence can be illustrated on the example of linear approximation, when we approximate the dependence by the first (linear) terms in its expansion into Taylor series, i.e., when the utility u_i of i-th person is equal to

$$u_i = u_i^{(0)} + \sum_{j \neq i} a_{ij} \cdot u_j,$$

where the interdependence is described by the coefficients a_{ij} – i.e., in effect, as a limit state of a linear neural network; the ideas of using neural networks to describe social interactions can be also found in [18].

Paradoxes of love. This simple and seemingly natural model leads to interesting and somewhat paradoxical conclusions; see, e.g., [2, 4, 12, 14].

For example, mutual affection between persons P_1 and P_2 means that $a_{12} > 0$ and $a_{21} > 0$. In particular, selfless love, when someone else's happiness means more than one's own, corresponds to $a_{12} > 1$.

In general, for two persons, we thus have

$$u_1 = u_1^{(0)} + a_{12} \cdot u_2; \quad u_2 = u_2^{(0)} + a_{21} \cdot u_1.$$

Once we know the original utility values $u_1^{(0)}$ and $u_2^{(0)}$, we can solve this system of linear equations and find the resulting values of utility:

$$u_1 = \frac{u_1^{(0)} + a_{12} \cdot u_2^{(0)}}{1 - a_{12} \cdot a_{21}}; \quad u_2 = \frac{u_2^{(0)} + a_{21} \cdot u_1^{(0)}}{1 - a_{12} \cdot a_{21}}.$$

As a result, when two people are deeply in love with each other ($a_{12} > 1$ and $a_{21} > 1$), then positive original pleasures $u_i^{(0)} > 0$ lead to $u_i < 0$ – i.e., to unhappiness. This phenomenon may be one of the reasons why people in love often experience deep negative emotions.

From this viewpoint, a situation when one person loves deeply and another rather allows him- or herself to be loved may lead to more happiness than

mutual passionate love. The fact that the coefficients a_{ij}, in general, change with time, explains a frequent family dynamics, when a passionate happy marriage surprisingly drifts into negative emotions.

A similar negative consequence of love can also happen in situations like selfless Mother's love when $a_{12} > 0$ may be not so large but a_{21} is so large that $a_{12} \cdot a_{21} > 1$.

There are also interesting consequences when we try to generalize these results to more than 2 persons. For example, we can define an ideal love, when each person treats other's emotions almost the same way as one's own, i.e., $a_{12} = a = 1 - \varepsilon$ for a small $\varepsilon > 0$. For two people, from $u_i^{(0)} > 0$, we get $u_i > 0$ – i.e., we can still have happiness. However, if we have three or more people in the state of mutual affection, i.e., if $u_i = u_i^{(0)} + a \cdot \sum\limits_{j \neq i} u_j$, then in case when everything is fine – e.g., $u_i^{(0)} = u^{(0)} > 0$ – we have

$$u_i \cdot (1 - a \cdot (n-1)) = u_i \cdot (2 - \varepsilon - (1-\varepsilon) \cdot n) = u^{(0)},$$

hence

$$u_i = \frac{u^{(0)}}{2 - \varepsilon - (1-\varepsilon) \cdot n} < 0,$$

i.e., we have unhappiness. This may be the reason why 2-person families are the main form – or, in other words, if two people care about the same person (e.g., his mother and his wife), all there of them are happier if there is some negative feeling (e.g., jealousy) between them.

Comment. It is important to distinguish between emotional interdependence in which one's utility is determined by the utility of other people, and "objective" altruism, in which one's utility depends on the material gain of other people – but not on their subjective utility values, i.e., in which (in the linearized case)

$$u_i = u_i^{(0)} + \sum_j a_{ij} \cdot u_j^{(0)}.$$

In this approach, when we care about others' well-being and not their emotions, no paradoxes arise, and any degree of altruism only improves the situation; see, e.g, [8, 9, 19].

This objective approach to interdependence was proposed and actively used by yet another Nobel Prize winner: Amartya K. Sen; see, e.g., [22, 23, 24].

Acknowledgments

This work was supported in part by the National Science Foundation grants HRD-0734825 and DUE-0926721, by Grant 1 T36 GM078000-01 from the National Institutes of Health, by Grant MSM 6198898701 from MŠMT of Czech Republic, and by Grant 5015 "Application of fuzzy logic with operators in the

knowledge based systems" from the Science and Technology Centre in Ukraine (STCU), funded by European Union.

The author is thankful to Jiawei Han, Leonid Perlovsky, and Paulo Pinheiro da Silva for valuable discussions.

References

[1] G. S. Becker, *A Treatise on the Family*, Harvard University Press, Cambridge, Massachusetts, 1991.

[2] T. Bergstrom, "Love and spaghetti, the opportunity cost of virtue", *Journal of Economic Perspectives*, 1989, Vol. 3, No., pp. 165–173.

[3] B. D. Bernheim and O. Stark, "Altruism within the family reconsidered: do nice guys finish last?", *American Economic Review*, 1988, Vol. 78, No. 5, pp. 1034–1045.

[4] T. Bergstron, *Systems of benevolent utility interdependence*, University of Michigan, Technical Report, 1991.

[5] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, 2007.

[6] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine", *Comput. Netw. ISDN Syst.*, 1998, Vol. 30, No. 1–7, pp. 107–117.

[7] D. D. Friedman, *Price Theory*, South-Western Publ., Cincinnati, Ohio, 1986.

[8] J. S. Harsanyi, *Rational behavior and bargaining equilibrium in games and social situations*, Cambridge University Press, New York, 1977.

[9] J. S. Harsanyi, "Morality and the theory of rational behavior", In: A. Sen and B. Williams, *Utilitarianism and Beyond*, Cambridge University Press, Cambridge, UK, 1982, pp. 39–62.

[10] H. Hori and S. Kanaya, "Utility functionals with nonpaternalistic intergerenational altruism", *Journal of Economic Theory*, 1989, Vol. 49, pp. 241–265.

[11] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks", *Proceedings of the 12th International World Wide Web Conference WWW'2003*, Budapest, Hungary, May 20–24, 2003.

[12] V. Kreinovich, *Paradoxes of Love: Game-Theoretic Explanation*, University of Texas at El Paso, Department of Computer Science, Technical Report UTEP-CS-90-16, July 1990.

[13] A. N. Langville, C. D. Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings*, Princeton University Press, Princeton, New Jersey, 2006.

[14] H. T. Nguyen, O. Kosheleva, and V. Kreinovich, "Decision Making Beyond Arrow's 'Impossibility Theorem', With the Analysis of Effects of Collusion and Mutual Attraction", *International Journal of Intelligent Systems*, 2009, Vol. 24, No. 1, pp. 27–47.

[15] L. I. Perlovsky, "Neural Network with Fuzzy Dynamic Logic", In: *Proceedings of the In- ternational IEEE and INNS Joint Conference on Neural Networks IJCNN'05*, Montreal, Quebec, Canada, 2005.

[16] L. I. Perlovsky, "Fuzzy Dynamic Logic", *New Math. and Natural Computation*, 2006, Vol. 2, No. 1, pp. 43–55.

[17] L. I. Perlovsky, "Neural Networks, Fuzzy Models and Dynamic Logic", In: R. Köhler and A. Mehler, eds., *Aspects of Automatic Text Analysis: Festschrift in Honor of Burghard Rieger*, Springer, Germany, 2007, pp. 363–386.

[18] L. Perlovsky, "Evolution of languages, consciousness, and culture", *IEEE Computational Intelligence Magazine*, 2009, Vol. 2, No. 3, pp. 25–39.

[19] L. Putterman,*Peasants, collectives, and choice: economic theory and Tanzania's villages*, JAI Press, Greenwich, Connecticut, 1986.

[20] A. Rapoport, "Some game theoretic aspects of parasitism and symbiosis", *Bull. Math. Biophysics*, 1956, Vol. 18.

[21] A. Rapoport, *Strategy and Conscience*, New York, 1964.

[22] A. K. Sen, "Labor allocation in a cooperative enterprise", *Review of Economic Studies*, 1966, Vol. 33, No. 4, pp. 361–371.

[23] A. K. Sen, *Collective Choice and Socieal Welfare*, Holden-Day, San Francisco, California, 1970.

[24] A. K. Sen, *Resources, Values, and Development*, Hardard University Press, Harvard, Massachusetts, 1984.

[25] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu, "RankClus: Integrating Clustering with Ranking for Heterogeneous Information Network Analysis", *Proceedings of the European Conference on Data Base Theory EDBT'2009*, Saint Petersburg, Russia, March 24–26, 2009.

[26] F. J. Tipler, *The Physics of Immortality*, Doubleday, New York, 1994.